NetScaler VPX 13.1

Contents

Support matrix and usage guidelines

This document lists the different hypervisors and features supported on a NetScaler VPX instance. The document also describes their usage guidelines and known limitations.

VPX instance on XenServer or Citrix Hypervisor

Citrix Hypervisor version	SysID	Performance range
8.2 supported 13.0 64.x onwards, 8.0, 7.6, 7.1	450000	10 Mbps to 40 Gbps

VPX instance on VMware ESXi hypervisor

ESXi version	ESXi release date (YYYY/MM/DD)	ESXi build number	NetScaler VPX version	Performance range
ESXi 8.0 update 3e	2025/04/10	24674464	13.1-58.x and higher builds	10 Mbps to 100 Gbps
ESXi 8.0 update 3d	2025/03/04	24585383	13.1-56.x and higher builds	
ESXi 8.0 update 3c	2025/01/23	24414501	13.1-55.x and higher builds	
ESXi 8.0 update 3b	2024/09/17	24280767	13.1-53.x and higher builds	
ESXi 8.0 update 3	2024/06/25	24022510	13.1-53.x and higher builds	
ESXi 8.0 update 2c	2024/05/21	23825572	13.1-53.x and higher builds	
ESXi 8.0 update 2b	2024/02/29	23305546	13.1-49.15, and 13.1-52.x and higher builds	
ESXi 8.0 update 2	2023/09/21	22380479	13.1-52.x and higher builds	
ESXi 8.0 update 1	2023/04/18	21495797	13.1-45.x and higher builds	
ESXi 8.0c	2023/03/30	21493926	13.1-45.x and higher builds	

ESXi version	ESXi release date (YYYY/MM/DD)	ESXi build number	NetScaler VPX version	Performance range
ESXi 8.0	2022/10/11	20513097	13.1-42.x and higher builds	
ESXi 7.0 update 3s	2025/03/04	24585291	13.1-55.x and higher builds	
ESXi 7.0 update 3r	2024/12/12	24411414	13.1-55.x and higher builds	
ESXi 7.0 update 3q	2024/05/21	23794027	13.1-53.x and higher builds	
ESXi 7.0 update 3p	2024/03/05	23307199	13.1-52.x and higher builds	
ESXi 7.0 update 3o	2023/09/28	22348816	13.1-51.x and higher builds	
ESXi 7.0 update 3n	2023/07/06	21930508	13.1-49.x and higher builds	
ESXi 7.0 update 3m	2023/05/03	21686933	13.1-48.x and higher builds	
ESXi 7.0 update 3i	2022/12/08	20842708	13.1-37.x and higher builds	
ESXi 7.0 update 3f	2022/07/12	20036589	13.1-33.x and higher builds	
ESXi 7.0 update 3d	2022/03/29	19482537	13.1-27.x and higher builds	
ESXi 7.0 update 3c	2022/01/27	19193900	13.1-21.x and higher builds	
ESX 7.0 update 2d	2021/09/14	18538813	13.1-9.x and higher builds	
ESX 7.0 update 2a	2021/04/29	17867351	13.1-4.x and higher builds	

> **Note:**
>
> Each ESXi patch support is validated on the NetScaler VPX version specified in the preceding table and is applicable for all the higher builds of NetScaler VPX 13.1 version.

VPX instance on Microsoft Hyper-V

Hyper-V version	SysID	Performance range
2016, 2019	450020	10 Mbps to 3 Gbps

VPX instance on Nutanix AHV

NetScaler VPX is supported on Nutanix AHV through the Citrix Ready partnership. Citrix Ready is a technology partner program that helps software and hardware vendors develop and integrate their products with NetScaler technology for digital workspace, networking, and analytics.

For more information on a step-by-step method to deploy a NetScaler VPX instance on Nutanix AHV, see Deploying a NetScaler VPX on Nutanix AHV.

Third-party support:

If you experience any issues with a particular third-party (Nutanix AHV) integration on a NetScaler environment, open a support incident directly with the third-party partner (Nutanix).

If the partner determines that the issue appears to be with NetScaler, the partner can approach NetScaler support for further assistance. A dedicated technical resource from partners works with the NetScaler support until the issue is resolved.

VPX instance on generic KVM

Generic KVM version	SysID	Performance range
RHEL 7.6, RHEL 8.0, RHEL 9.3 Ubuntu 16.04, Ubuntu 18.04, Ubuntu 22.04	450070	10 Mbps to 100 Gbps

Points to note:

Consider the following points while using KVM hypervisors.

- The VPX instance is qualified for hypervisor release versions mentioned in table 1–4, and not for patch releases within a version. However, the VPX instance is expected to work seamlessly with patch releases of a supported version. If it does not, log a support case for troubleshooting and debugging.
- Before using RHEL 7.6, complete the following steps on the KVM host:
    1. Edit /etc/default/grub and append `"kvm_intel.preemption_timer=0"` to `GRUB_CMDLINE_LINUX` variable.

2. Regenerate grub.cfg with the command `"# grub2-mkconfig -o /boot/grub2/grub.cfg"`.
3. Restart the host machine.

- Before using Ubuntu 18.04, complete the following steps on the KVM host:
  1. Edit /etc/default/grub and append `"kvm_intel.preemption_timer=0"` to `GRUB_CMDLINE_LINUX` variable.
  2. Regenerate grub.cfg with the command `"# grub-mkconfig -o /boot/grub/grub.cfg “`.
  3. Restart the host machine.

VPX instance on public clouds

Public cloud	SysID	Performance range
AWS	450040	10 Mbps to 30 Gbps
Azure	450020	10 Mbps to 10 Gbps
GCP	450070	10 Mbps to 10 Gbps

VPX features supported on hypervisors

Hypervisors → Features ↓	VPX on XenServer		VPX on VMware ESX				VPX on Microsoft Hyper-V	VPX on generic KVM		
Interfaces →	**PV**	**SR-IOV**	**PV**	**SR-IOV**	**Emulated**	**PCI Passthrough**	**PV**	**PV**	**SR-IOV**	**PCI Passthrough**
Multi-PE Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Clustering Support	Yes	Yes[1]	Yes	Yes[1]	Yes	Yes	Yes	Yes	Yes[1]	Yes

Hypervisors → Features ↓	VPX on XenServer		VPX on VMware ESX				VPX on Microsoft Hyper-V	VPX on generic KVM		
VLAN Tagging	Yes	Yes	Yes	Yes	Yes	Yes	Yes(only on 2012R2)	Yes	Yes	Yes
Detecting Link Events/HA-Mon	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	No[2]	Yes[3]	Yes[3]
Interface Parameter Configuration	No	No	No	No	No	Yes	No	No	No	Yes
Static LA	Yes[2]	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	Yes[2]	Yes[2]	Yes[3]	Yes[3]
LACP	No	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]
Static CLAG	No	No	No	No	No	No	No	No	No	No
LACP CLAG	No	No	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]
Hot-plug	No	No	No	No	No	No	No	No	No	No

VPX features supported on public clouds

Public clouds → Features ↓	VPX on AWS	VPX on Azure	VPX on GCP
Multi-PE Support	Yes	Yes	Yes
Clustering Support	No	No	No
VLAN Tagging	No	No	No

Public clouds → Features ↓	VPX on AWS	VPX on Azure	VPX on GCP
Detecting Link Events/HAMon	No[2]	No[2]	No[2]
Interface Parameter Configuration	No	No	No
Static LA	No	No	No
LACP	No	No	No
Static CLAG	No	No	No
LACP CLAG	No	No	No
Hot-plug	Yes	No	No

The superscript numbers (1, 2, 3) used in the two preceding tables refers to the following points with respective numbering:

1. Clustering support is available on SRIOV for client-facing and server-facing interfaces, and not for the backplane.
2. Interface DOWN events are not recorded in NetScaler VPX instances.
3. For Static LA, traffic might still be sent on the interface whose physical status is DOWN.

The following points apply to the respective features captured in the two preceding tables:

- For LACP, the peer device knows the interface DOWN event based on the LACP timeout mechanism.
  - Short timeout: 3 seconds
  - Long timeout: 90 seconds
- For LACP, do not share interfaces across VMs.
- For Dynamic routing, convergence time depends on the Routing Protocol since link events aren't detected.
- Monitored static Route functionality fails if you do not bind monitors to static routes because the Route state depends on the VLAN status. The VLAN status depends on the link status.
- Partial failure detection does not happen in high availability if there's link failure. High availability-split brain condition might happen if there's link failure.

  - When any link event (disable, enable, reset) is generated from a VPX instance, the physical status of the link does not change. For static LA, any traffic initiated by the peer gets dropped on the instance.
  - For the VLAN tagging feature to work on the VMware ESX, set the port group's VLAN ID to 1–4095 on the vSwitch of the VMware ESX server.

- Hot-plug is not supported on VPX instances with ENA interfaces, and the behavior of the instances can be unpredictable if hot-plugging is attempted. Hot adding is supported only for PV and SRIOV interfaces with NetScaler on AWS.
- Hot removing either through the AWS Web console or AWS CLI interface is not supported with the PV, SRIOV, and ENA interfaces for NetScaler. The behavior of the instances can be unpredictable if hot-removal is attempted.

Supported browsers

For information on supported browsers for accessing NetScaler GUI versions 14.1 and 13.1, see Compatible browsers.

Supported processors for NetScaler VPX

Platforms	Intel Processor	AMD Processor
Citrix Hypervisor	Yes	No
ESXi Hypervisor	Yes	Yes
Hyper-V	Yes	No
KVM	Yes	No
AWS	Yes	Yes
Azure	Yes	Yes
GCP	Yes	Yes

Supported NICs for NetScaler VPX

The following table lists the NICs supported on a VPX platform or cloud.

NICs → Platforms ↓	Mellanox CX-3	Mellanox CX-4	Mellanox CX-5	Intel 82599 SRIOV VF	Intel X710/X722/XL710 SRIOV VF	Intel X710/XL710/XXV710 PCI-Passthrough Mode
Citrix Hypervisor	NA	NA	NA	Yes	Yes	No
ESXi Hypervisor	No	Yes	No	Yes	No	Yes
Hyper-V	NA	NA	NA	No	No	No
KVM	No	Yes	Yes	Yes	Yes	No
AWS	NA	NA	NA	Yes	NA	NA
Azure	Yes	Yes	Yes	NA	NA	NA
GCP	NA	NA	NA	NA	NA	NA

Usage guidelines

Follow these usage guidelines:

- We recommend you to deploy a VPX instance on local disks of the server or SAN-based storage volumes.

See the **VMware ESXi CPU Considerations** section in the Performance Best Practices for VMware vSphere 6.5 document. Here's an extract:

- It isn't recommended that virtual machines with high CPU/Memory demand sit on a Host or Cluster that is overcommitted.
- In most environments, ESXi allows significant levels of CPU overcommitment without impacting virtual machine performance. On a host, you can run more vCPUs than the total number of physical processor cores in that host.
- If an ESXi host becomes CPU saturated, that is, the virtual machines and other loads on the host demand all the CPU resources the host has, latency-sensitive workloads might not perform well. In this case you might want to reduce the CPU load, for example by powering off some virtual machines or migrating them to a different host (or allowing DRS to migrate them automatically).
- Citrix recommends the latest hardware compatibility version to avail the latest feature sets of the ESXi hypervisor for the virtual machine. For more information about the hardware and ESXi version compatibility, see the VMware documentation.
- The NetScaler VPX is a latency-sensitive, high-performance virtual appliance. To deliver its expected performance, the appliance requires vCPU reservation, memory reservation, vCPU pin-

ning on the host. Also, hyper threading must be disabled on the host. If the host does not meet these requirements, issues such as high-availability failover, CPU spike within the VPX instance, sluggishness in accessing the VPX CLI, pit boss daemon crash, packet drops, and low throughput occur.

A hypervisor is considered over-provisioned if one of the following two conditions is met:

- The total number of virtual cores (vCPU) provisioned on the host is greater than the total number of physical cores (pCPUs).
- The total number of provisioned VMs consume more vCPUs than the total number of pCPUs.

  If an instance is over-provisioned, the hypervisor might not guarantee the resources reserved (such as CPU, memory, and others) for the instance due to hypervisor scheduling over-heads, bugs, or limitations with the hypervisor. This behavior can cause lack of CPU resource for NetScaler and might lead to the issues mentioned in the first point under **Usage guidelines**. As administrators, you're recommended to reduce the tenancy on the host so that the total number of vCPUs provisioned on the host is lesser or equal to the total number of pCPUs.

  Example

  For ESX hypervisor, if the `%RDY%` parameter of a VPX vCPU is greater than 0 in the `esxtop` command output, the ESX host is said to be having scheduling overheads, which can cause latency related issues for the VPX instance.

  In such a situation, reduce the tenancy on the host so that `%RDY%` returns to 0 always. Alternatively, contact the hypervisor vendor to triage the reason for not honoring the resource reservation done.

- Hot adding is supported only for PV and SRIOV interfaces with NetScaler on AWS. VPX instances with ENA interfaces do not support hot-plug, and the behavior of the instances can be unpredictable if hot-plugging is attempted.
- Hot removing either through the AWS Web console or AWS CLI interface is not supported with the PV, SRIOV, and ENA interfaces for NetScaler. The behavior of the instances can be unpredictable if hot-removal is attempted.

Commands to control the packet engine CPU usage

You can use two commands (`set ns vpxparam` and `show ns vpxparam`) to control the packet engine (non-management) CPU usage behavior of VPX instances in hypervisor and cloud environments:

- `set ns vpxparam [-cpuyield (YES | NO | DEFAULT)] [-masterclockcpu1 (YES | NO)]`

Allow each VM to use CPU resources that have been allocated to another VM but are not being used.

`Set ns vpxparam` parameters:

-cpuyield: Release or do not release of allocated but unused CPU resources.

- **YES**: Allow allocated but unused CPU resources to be used by another VM.
- **NO**: Reserve all CPU resources for the VM to which they have been allocated. This option shows a higher percentage in hypervisor and cloud environments for VPX CPU usage.
- **DEFAULT**: No.

> **Note:**
>
> On all the NetScaler VPX platforms, the vCPU usage on the host system is 100 percent. Type the `set ns vpxparam –cpuyield YES` command to override this usage.

If you want to set the cluster nodes to "yield", you must perform the following extra configurations on CCO:

- If a cluster is formed, all the nodes come up with "yield=DEFAULT".
- If a cluster is formed using the nodes that are already set to "yield=YES", then the nodes are added to the cluster using the "DEFAULT"yield.

> **Note:**
>
> If you want to set the cluster nodes to "yield=YES", you can configure only after forming the cluster but not before the cluster is formed.

-masterclockcpu1: You can move the main clock source from CPU0 (management CPU) to CPU1. This parameter has the following options:

- **YES**: Allow the VM to move the main clock source from CPU0 to CPU1.
- **NO**: VM uses CPU0 for the main clock source. By default, CPU0 is the main clock source.

* `show ns vpxparam`

  Display the current `vpxparam` settings.

Other References

- For Citrix Ready products, visit Citrix Ready Marketplace.
- For Citrix Ready product support, see the Citrix Ready partners page.
- For VMware ESX hardware versions, see Upgrading VMware Tools.

Optimize NetScaler VPX performance on VMware ESX, Linux KVM, and Citrix Hypervisors

The NetScaler VPX performance greatly varies depending on the hypervisor, allocated system resources, and the host configurations. To achieve the desired performance, first follow the recommendations in the VPX data sheet, and then further optimize it using the best practices provided in this document.

NetScaler VPX instance on VMware ESX hypervisors

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of NetScaler VPX instance on VMware ESX hypervisors.

- Recommended configuration on ESX hosts
- NetScaler VPX with E1000 network interfaces
- NetScaler VPX with VMXNET3 network interfaces
- NetScaler VPX with SR-IOV and PCI passthrough network interfaces

Recommended configuration on ESX hosts

To achieve high performance for VPX with E1000, VMXNET3, SR-IOV, and PCI passthrough network interfaces, follow these recommendations:

- The total number of virtual CPUs (vCPUs) provisioned on the ESX host must be less than or equal to the total number of physical CPUs (pCPUs) on the ESX host.
- Non-uniform Memory Access (NUMA) affinity and CPU affinity must be set for the ESX host to achieve good results.

  –To find the NUMA affinity of a Vmnic, log in to the host locally or remotely, and type:

  ```
  #vsish -e get /net/pNics/vmnic7/properties | grep NUMA
  Device NUMA Node: 0
  ```

  - To set NUMA and vCPU affinity for a VM, see VMware documentation.

NetScaler VPX with E1000 network interfaces

Perform the following settings on the VMware ESX host:

- On the VMware ESX host, create two vNICs from one pNIC vSwitch. Multiple vNICs create multiple Rx threads in the ESX host. This increases the Rx throughput of the pNIC interface.

- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase vNIC transmit (Tx) throughput, use a separate Tx thread in the ESX host per vNIC. Use the following ESX command:
  - For ESX version 5.5:

    ```
    esxcli system settings advanced set –o /Net/NetTxWorldlet –
        i
    ```

  - For ESX version 6.0 onwards:

    ```
    esxcli system settings advanced set -o /Net/NetVMTxType –i 1
    ```

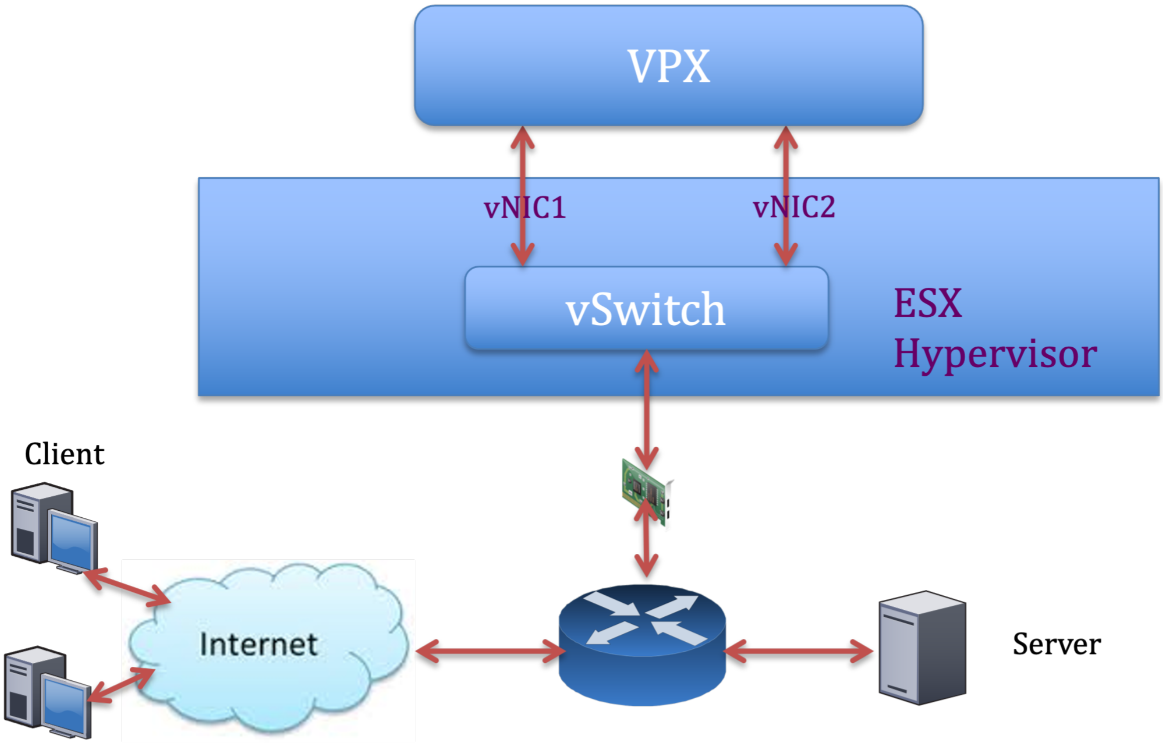
- To further increase the vNIC Tx throughput, use a separate Tx completion thread and Rx threads per device (NIC) queue. Use the following ESX command:

  ```
  esxcli system settings advanced set -o /Net/
      NetNetqRxQueueFeatPairEnable -i 0
  ```

> **Note:**
>
> Make sure that you reboot the VMware ESX host to apply the updated settings.

Two vNICs per pNIC deployment

The following is a sample topology and configuration commands for the **Two vNICs per pNIC** model of deployment that delivers better network performance.



NetScaler VPX sample configuration:

To achieve the deployment shown in the preceding sample topology, perform the following configuration on the NetScaler VPX instance:

- On the client side, bind the SNIP (1.1.1.2) to network interface 1/1 and enable the VLAN tag mode.

```
bind vlan 2 -ifnum 1/1 – tagged
bind vlan 2 -IPAddress 1.1.1.2 255.255.255.0
```

- On the server side, bind the SNIP (2.2.2.2) to network interface 1/1 and enable the VLAN tag mode.

```
bind vlan 3 -ifnum 1/2 – tagged
bind vlan 3 -IPAddress 2.2.2.2 255.255.255.0
```

- Add an HTTP virtual server (1.1.1.100) and bind it to a service (2.2.2.100).

```
add lb vserver v1 HTTP 1.1.1.100 80 -persistenceType NONE -
    Listenpolicy None -cltTimeout 180
add service s1 2.2.2.100 HTTP 80 -gslb NONE -maxClient 0 -maxReq
    0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
    180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
bind lb vserver v1 s1
```


> **Note:**
>
> Make sure that you include the following two entries in the route table:
>
> - 1.1.1.0/24 subnet with gateway pointing to SNIP 1.1.1.2
> - 2.2.2.0/24 subnet with gateway pointing to SNIP 2.2.2.2

NetScaler VPX with VMXNET3 network interfaces

To achieve high performance for VPX with VMXNET3 network interfaces, do the following settings on the VMware ESX host:

- Create two vNICs from one pNIC vSwitch. Multiple vNICs create multiple Rx threads in the ESX host. This increases the Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase vNIC transmit (Tx) throughput, use a separate Tx thread in the ESX host per vNIC. Use the following ESX commands:
  - For ESX version 5.5:

```
esxcli system settings advanced set –o /Net/NetTxWorldlet –i
```

  - For ESX version 6.0 onwards:

```
 esxcli system settings advanced set -o /Net/NetVMTxType –i 1
```

On the VMware ESX host, perform the following configuration:

- On the VMware ESX host, create two vNICs from 1 pNIC vSwitch. Multiple vNICs create multiple Tx and Rx threads in the ESX host. This increases the Tx and Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase Tx throughput of a vNIC, use a separate Tx completion thread and Rx threads per device (NIC) queue. Use the following command:

```
esxcli system settings advanced set -o /Net/
    NetNetqRxQueueFeatPairEnable -i 0
```

- Configure a VM to use one transmit thread per vNIC, by adding the following setting to the VM's configuration:

```
ethernetX.ctxPerDev = "1"
```

- Configure a VM to use up to 8 transmit thread per vNIC, by adding the following setting to the VM's configuration:


```
ethernetX.ctxPerDev = "3"
```

> **Note:**
>
> Increasing the transmit threads per vNIC requires more CPU resources (up to 8) on the ESX host. Ensure that sufficient CPU resources are available before making the preceding settings.

> **Note:**
>
> Make sure that you reboot the VMware ESX host to apply the updated settings.

You can configure VMXNET3 as a **Two vNICs per pNIC** deployment. For more information, see Two vNICs per pNIC deployment.

Configure multi-queue and RSS support on VMware ESX for VMXNET3 devices By default, the VMXNET3 device supports only 8 Rx and Tx queues. When the number of vCPUs on the VPX goes beyond 8, the number of Rx and Tx queues configured for a VMXNET3 interface switches to 1 by default. You can configure up to 19 Rx and Tx queues for VMXNET3 devices by changing certain configurations on ESX. This option increases the performance and uniform distribution of packets across the vCPUs of the VPX instance.

> **Note:**
>
> Starting from NetScaler release 13.1 build 48.x, the NetScaler VPX supports up to 19 Rx and Tx queues on ESX for VMXNET3 devices.

Prerequisites:

To configure up to 19 Rx and Tx queues on ESX for VMXNET3 devices, make sure that the following prerequisites are met:

- NetScaler VPX version is 13.1 build 48.X and later.
- NetScaler VPX is configured with a virtual machine of hardware version 17 and later, which is supported by VMware ESX 7.0 and later.

Configure VMXNET3 interfaces to support more than 8 Rx and Tx queues:

1. Open the virtual machine configuration file (.vmx) file.
2. Specify the number of Rx and TX queues by configuring the `ethernetX.maxTxQueues` and `ethernetX.maxRxQueues` values (where X is the number of the virtual NICs to configure). The maximum number of queues configured must not be greater than the number of vCPUs in the virtual machine.

> **Note:**
>
> Increasing the number of queues also increases the processor overhead on the ESX host. Therefore, ensure that sufficient CPU resources are available in the ESX host before increasing the queues. You can increase the maximum number of queues supported, in scenarios, where the number of queues are identified as a bottleneck for performance. In these situations, we recommend increasing the number of queues gradually. For example, from 8 to 12, then to 16, then to 20, and so on. Evaluate the performance at each setting, rather than increasing directly to the maximum limit.

NetScaler VPX with SR-IOV and PCI passthrough network interfaces

To achieve high performance for VPX with SR-IOV and PCI passthrough network interfaces, see Recommended configuration on ESX hosts.

NetScaler VPX instance on Linux-KVM platform

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of NetScaler VPX instance on Linux-KVM platform.

- Performance settings for KVM
- NetScaler VPX with PV network interfaces
- NetScaler VPX with SR-IOV and Fortville PCIe passthrough network interfaces

Performance settings for KVM

Perform the following settings on the KVM host:

Find the NUMA domain of the NIC using the `lstopo` command:

Make sure that memory for the VPX and the CPU is pinned to the same location.
In the following output, the 10G NIC "ens2"is tied to NUMA domain #1.


```
[root@localhost ~]# lstopo-no-graphics
Machine (128GB)
  NUMANode L#0 (P#0 64GB)
    Socket L#0 + L3 L#0 (20MB)
      L2 L#0 (256KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 + PU L#0 (P#0)
      L2 L#1 (256KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1 + PU L#1 (P#1)
      L2 L#2 (256KB) + L1d L#2 (32KB) + L1i L#2 (32KB) + Core L#2 + PU L#2 (P#2)
      L2 L#3 (256KB) + L1d L#3 (32KB) + L1i L#3 (32KB) + Core L#3 + PU L#3 (P#3)
      L2 L#4 (256KB) + L1d L#4 (32KB) + L1i L#4 (32KB) + Core L#4 + PU L#4 (P#4)
      L2 L#5 (256KB) + L1d L#5 (32KB) + L1i L#5 (32KB) + Core L#5 + PU L#5 (P#5)
      L2 L#6 (256KB) + L1d L#6 (32KB) + L1i L#6 (32KB) + Core L#6 + PU L#6 (P#6)
      L2 L#7 (256KB) + L1d L#7 (32KB) + L1i L#7 (32KB) + Core L#7 + PU L#7 (P#7)
    HostBridge L#0
      PCIBridge
        PCI 8086:1521
          Net L#0 "eno1"
        PCI 8086:1521
          Net L#1 "eno2"
      PCIBridge
        PCI 8086:1584
          Net L#2 "ens3"
      PCIBridge
        PCI 8086:1584
          Net L#3 "ens4"
      PCI 8086:8d62
        Block L#4 "sda"
        Block L#5 "sdb"
      PCIBridge
        PCIBridge
          PCI 1a03:2000
            GPU L#6 "card0"
            GPU L#7 "controlD64"
      PCI 8086:8d02
  NUMANode L#1 (P#1 64GB)
    Socket L#1 + L3 L#1 (20MB)
      L2 L#8 (256KB) + L1d L#8 (32KB) + L1i L#8 (32KB) + Core L#8 + PU L#8 (P#8)
      L2 L#9 (256KB) + L1d L#9 (32KB) + L1i L#9 (32KB) + Core L#9 + PU L#9 (P#9)
      L2 L#10 (256KB) + L1d L#10 (32KB) + L1i L#10 (32KB) + Core L#10 + PU L#10 (P#10)
      L2 L#11 (256KB) + L1d L#11 (32KB) + L1i L#11 (32KB) + Core L#11 + PU L#11 (P#11)
      L2 L#12 (256KB) + L1d L#12 (32KB) + L1i L#12 (32KB) + Core L#12 + PU L#12 (P#12)
      L2 L#13 (256KB) + L1d L#13 (32KB) + L1i L#13 (32KB) + Core L#13 + PU L#13 (P#13)
      L2 L#14 (256KB) + L1d L#14 (32KB) + L1i L#14 (32KB) + Core L#14 + PU L#14 (P#14)
      L2 L#15 (256KB) + L1d L#15 (32KB) + L1i L#15 (32KB) + Core L#15 + PU L#15 (P#15)
    HostBridge L#6
      PCIBridge
        PCI 8086:1584
          Net L#8 "ens2"
      PCIBridge
        PCI 8086:10fb
          Net L#9 "ens1f0"
        PCI 8086:10fb
          Net L#10 "ens1f1"
        PCI ffff:ffff
          Net L#11 "enp131s16"
[root@localhost ~]#  modprobe kvm-intel acpienv=N
```

Allocate the VPX memory from the NUMA domain.

The `numactl` command indicates the NUMA domain from which the memory is allocated. In the following output, around 10 GB RAM is allocated from NUMA node #0.

```
[root@localhost ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 65429 MB
node 0 free: 55854 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 65536 MB
node 1 free: 52388 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
[root@localhost ~]#
```

To change the NUMA node mapping, follow these steps.

1. Edit the .xml of the VPX on the host.

   ```
   /etc/libvirt/qemu/<VPX_name>.xml
   ```


2. Add the following tag:

```
<numatune>
<memory mode="strict" nodeset="1"/>    ⊠ This is the NUMA domain
    name
</numatune>
```

3. Shut down the VPX.

4. Run the following command:

```
virsh define /etc/libvirt/qemu/<VPX_name>.xml
```

This command updates the configuration information for the VM with the NUMA node mappings.

5. Power on the VPX. Then check the `numactl –hardware` command output on the host to see the updated memory allocations for the VPX.

```
[root@localhost ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 65429 MB
node 0 free: 65429 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 65536 MB
node 1 free: 55854 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
[root@localhost ~]#
```

Pin vCPUs of VPX to physical cores.

- To view the vCPU to pCPU mappings of a VPX, type the following command

```
virsh vcpupin <VPX name>
```

```
root@localhost qemu]#  virsh vcpupin NS-VPX-DVR
PU: CPU Affinity
----------------------------------
  0: 8
  1: 9
  2: 10
  3: 11
```

The vCPUs 0–4 are mapped to physical cores 8–11.

- To view the current pCPU usage, type the following command:

```
mpstat -P ALL 5
```



```
[root@localhost qemu]# mpstat -P ALL 5
Linux 3.10.0-123.el7.x86_64 (localhost.localdomain)     05/17/2016      _x86_64_        (16 CPU)

02:26:20 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
02:26:25 PM  all    0.24    0.00    1.67    0.00    0.00    0.00    0.00   17.32    0.00   80.78
02:26:25 PM    0    0.20    0.00    1.00    0.00    0.00    0.00    0.00    0.00    0.00   98.80
02:26:25 PM    1    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    2    0.20    0.00    0.40    0.00    0.00    0.00    0.00    0.00    0.00   99.40
02:26:25 PM    3    0.00    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.80
02:26:25 PM    4    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    5    0.60    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.20
02:26:25 PM    6    0.40    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    7    1.62    0.00    1.42    0.00    0.00    0.00    0.00    0.00    0.00   96.96
02:26:25 PM    8    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM    9    0.00    0.00    7.60    0.00    0.00    0.00    0.00   92.40    0.00    0.00
02:26:25 PM   10    0.20    0.00    7.00    0.00    0.00    0.00    0.00   92.80    0.00    0.00
02:26:25 PM   11    0.00    0.00    8.60    0.00    0.00    0.00    0.00   91.40    0.00    0.00
02:26:25 PM   12    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   13    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   14    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   15    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
```

In this output, 8 is management CPU, and 9–11 are packet engines.

- To change the vCPU to pCPU pinning, there are two options.
  - Change it at runtime after the VPX boots up using the following command:

    ```
    virsh vcpupin <VPX name> <vCPU id>  <pCPU number>
    virsh vcpupin NetScaler-VPX-XML 0 8
    virsh vcpupin NetScaler-VPX-XML 1 9
    virsh vcpupin NetScaler-VPX-XML 2 10
    virsh vcpupin NetScaler-VPX-XML 3 11
    ```

  - To make static changes to the VPX, edit the `.xml` file as before with the following tags:

    1. Edit the .xml file of the VPX on the host

       ```
       /etc/libvirt/qemu/<VPX_name>.xml
       ```

    2. Add the following tag:

       ```
       <vcpu placement='static' cpuset='8-11'>4</vcpu>
           <cputune>
               <vcpupin vcpu='0' cpuset='8'/>
               <vcpupin vcpu='1' cpuset='9'/>
               <vcpupin vcpu='2' cpuset='10'/>
               <vcpupin vcpu='3' cpuset='11'/>
           </cputune>
       ```

    3. Shut down the VPX.

    4. Update the configuration information for the VM with the NUMA node mappings using the following command:

       ```
       virsh define /etc/libvirt/qemu/ <VPX_name>.xml
       ```

5. Power on the VPX. Then check the `virsh vcpupin <VPX name>` command output on the host to see the updated CPU pinning.

Eliminate host interrupt overhead.

- Detect VM_EXITS using the `kvm_stat` command.

  At the hypervisor level, host interrupts are mapped to the same pCPUs on which the vCPUs of the VPX are pinned. This might cause vCPUs on the VPX to get kicked out periodically.

  To find the VM exits done by VMs running the host, use the `kvm_stat` command.

```
[root@localhost ~]# kvm_stat -1  | grep EXTERNAL
kvm_exit(EXTERNAL_INTERRUPT)  1728349 27738
[root@localhost ~]#
```

  A higher value in the order of 1+M indicates an issue.

  If a single VM is present, the expected value is 30–100 K. Anything more than that can indicate that there are one or more host interrupt vectors mapped to the same pCPU.

- Detect host interrupts and migrate host interrupts.

  When you run the `concatenate` command for the "/proc/interrupts"file, it displays all the host interrupt mappings. If one or more active IRQs map to the same pCPU, its corresponding counter increments.

  Move any interrupts that overlap with your NetScaler VPX's pCPUs to unused pCPUs:

```
echo 0000000f > /proc/irq/55/smp_affinity
0000000f - - > it is a bitmap, LSBs indicates that IRQ 55 can
    only be scheduled on pCPUs 0 – 3
```

- Disable IRQ balance.

  Disable IRQ balance daemon, so that no rescheduling happens on the fly.

```
service irqbalance stop
service irqbalance show - To check the status
service irqbalance start - Enable if needed
```

  Make sure you run the `kvm_stat` command to ensure that there are not many counters.

NetScaler VPX with PV network interfaces

You can configure para-virtualization (PV), SR-IOV, and PCIe passthrough network interfaces as a **Two vNICs per pNIC** deployment. For more information, see Two vNICs per pNIC deployment.

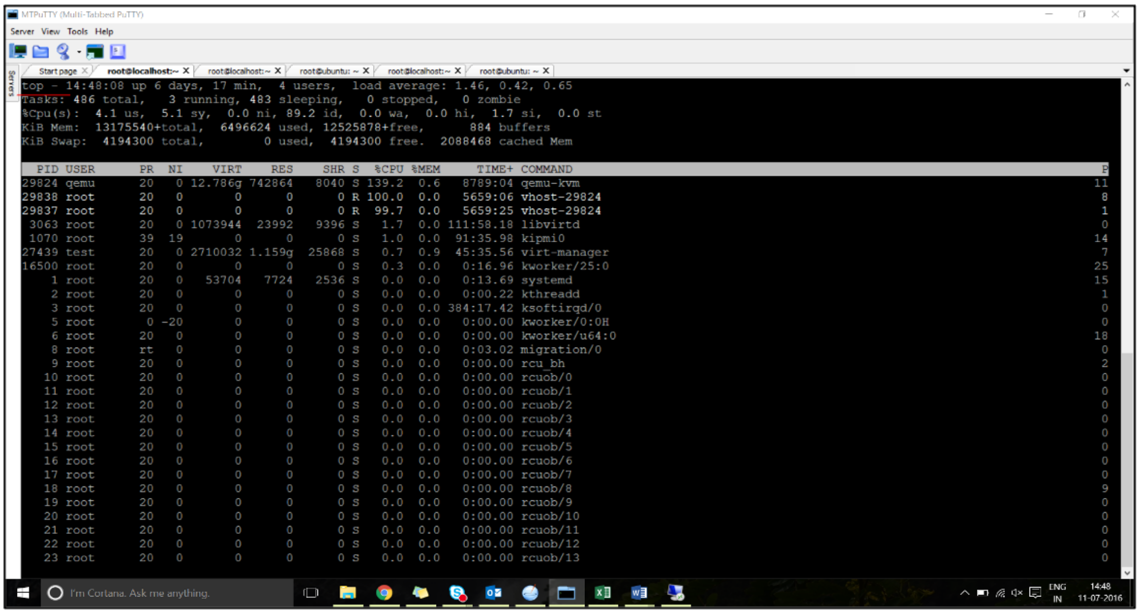
For optimal performance of PV (virtio) interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot/NIC is tied to.

- The Memory and vCPU for the VPX must be pinned to the same NUMA domain.
- Vhost thread must be bound to the CPUs in the same NUMA domain.

Bind the virtual host threads to the corresponding CPUs:

1. Once the traffic is started, run the `top` command on the host.



2. Identify the virtual host process (named as `vhost-<pid-of-qemu>`) affinity.

3. Bind the vHost processes to the physical cores in the NUMA domain identified earlier using the following command:

   ```
   taskset –pc <core-id> <process-id>
   ```

   Example:

   ```
   taskset –pc 12 29838
   ```

4. The processor cores corresponding to the NUMA domain can be identified with the following command:

   ```
   [root@localhost ~]# virsh capabilities | grep cpu
   <cpu>
       </cpu>
           <cpus num='8'>
               <cpu id='0' socket_id='0' core_id='0' siblings='0'/>
               <cpu id='1' socket_id='0' core_id='1' siblings='1'/>
               <cpu id='2' socket_id='0' core_id='2' siblings='2'/>
               <cpu id='3' socket_id='0' core_id='3' siblings='3'/>
               <cpu id='4' socket_id='0' core_id='4' siblings='4'/>
               <cpu id='5' socket_id='0' core_id='5' siblings='5'/>
               <cpu id='6' socket_id='0' core_id='6' siblings='6'/>
               <cpu id='7' socket_id='0' core_id='7' siblings='7'/>
   ```



```
        </cpus>

        <cpus num='8'>
        <cpu id='8' socket_id='1' core_id='0' siblings='8'/>
        <cpu id='9' socket_id='1' core_id='1' siblings='9'/>
        <cpu id='10' socket_id='1' core_id='2' siblings='10'/>
        <cpu id='11' socket_id='1' core_id='3' siblings='11'/>
        <cpu id='12' socket_id='1' core_id='4' siblings='12'/>
        <cpu id='13' socket_id='1' core_id='5' siblings='13'/>
        <cpu id='14' socket_id='1' core_id='6' siblings='14'/>
        <cpu id='15' socket_id='1' core_id='7' siblings='15'/>
        </cpus>

    <cpuselection/>
    <cpuselection/>
```

Bind the QEMU process to the corresponding physical core:

1. Identify the physical cores on which the QEMU process is running. For more information, see the preceding output.
2. Bind the QEMU process to the same physical cores to which you bind the vCPUs, using the following command:

```
taskset –pc 8-11 29824
```

NetScaler VPX with SR-IOV and Fortville PCIe passthrough network interfaces

For optimal performance of the SR-IOV and Fortville PCIe passthrough network interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot/NIC is tied to.
- The Memory and vCPU for the VPX must be pinned to the same NUMA domain.

Sample VPX XML file for vCPU and memory pinning for Linux KVM:

```
    <domain type='kvm'>
        <name>NetScaler-VPX</name>
        <uuid>138f7782-1cd3-484b-8b6d-7604f35b14f4</uuid>
        <memory unit='KiB'>8097152</memory>
        <currentMemory unit='KiB'>8097152</currentMemory>
        <vcpu placement='static'>4</vcpu>

    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
    </cputune>

```



```
    <numatune>
    <memory mode='strict' nodeset='1'/>
    </numatune>

    </domain>
```

NetScaler VPX instance on Citrix Hypervisors

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of NetScaler VPX instance on Citrix Hypervisors.

- Performance settings for Citrix Hypervisors
- NetScaler VPX with SR-IOV network interfaces
- NetScaler VPX with para-virtualized interfaces

Performance settings for Citrix Hypervisors

Find the NUMA domain of the NIC using the "xl"command:

```
xl info -n
```

Pin vCPUs of VPX to physical cores.

```
xl vcpu-pin <Netsclaer VM Name>  <vCPU id>  <physical CPU id>
```

Check binding of vCPUs.

```
xl vcpu-list
```

Allocate more than 8 vCPUs to NetScaler VMs.

For configuring more than 8 vCPUs, run the following commands from the Citrix Hypervisor console:

```
xe vm-param-set uuid=your_vms_uuid VCPUs-max=16
xe vm-param-set uuid=your_vms_uuid VCPUs-at-startup=16
```

NetScaler VPX with SR-IOV network interfaces

For optimal performance of the SR-IOV network interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot or NIC is tied to.
- Pin the Memory and vCPU for the VPX to the same NUMA domain.
- Bind the Domain-0 vCPU to the remaining CPU.

NetScaler VPX with para-virtualized interfaces

For optimal performance, two vNICs per pNIC and one vNIC per pNIC configurations are advised, as in other PV environments.

To achieve optimal performance of para-virtualized (netfront) interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot or NIC is tied to.
- Pin the memory and vCPU for the VPX to the same NUMA domain.
- Bind the Domain-0 vCPU to the remaining CPU of the same NUMA domain.
- Pin host Rx/Tx threads of vNIC to Domain-0 vCPUs.

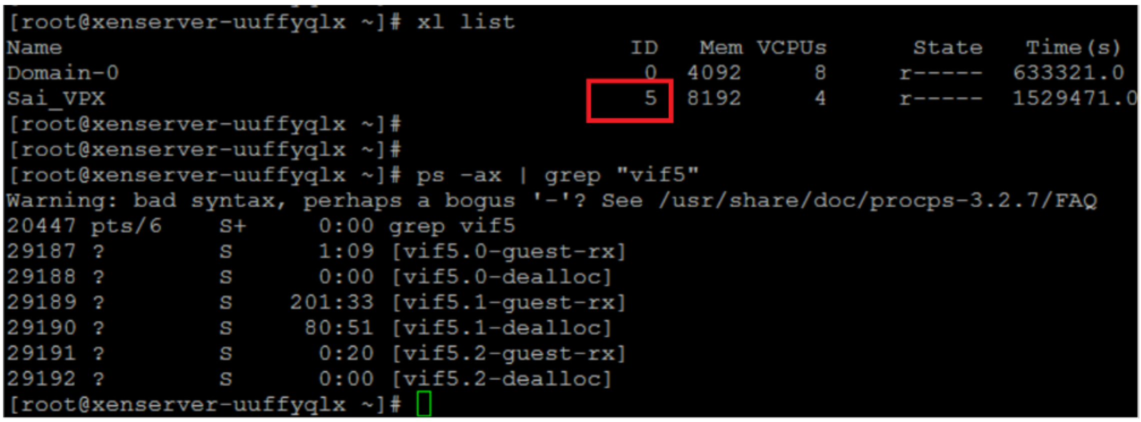
Pin host threads to Domain-0 vCPUs:

1. Find Xen-ID of the VPX by using the `xl list` command on the Citrix Hypervisor host shell.
2. Identify host threads by using the following command:

   ```
   ps -ax | grep vif <Xen-ID>
   ```

   In the following example, these values indicate:

   - **vif5.0** - The threads for first interface allocated to VPX in XenCenter (management interface).
   - **vif5.1** - The threads for second interface assigned to VPX and so on.

   ```
   [root@xenserver-uuffyqlx ~]# xl list
   Name                                        ID   Mem VCPUs      State   Time(s)
   Domain-0                                     0  4092     8     r-----  633321.0
   Sai_VPX                                      5  8192     4     r-----  1529471.0
   [root@xenserver-uuffyqlx ~]#
   [root@xenserver-uuffyqlx ~]#
   [root@xenserver-uuffyqlx ~]# ps -ax | grep "vif5"
   Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
   20447 pts/6    S+     0:00 grep vif5
   29187 ?        S      1:09 [vif5.0-guest-rx]
   29188 ?        S      0:00 [vif5.0-dealloc]
   29189 ?        S    201:33 [vif5.1-guest-rx]
   29190 ?        S     80:51 [vif5.1-dealloc]
   29191 ?        S      0:20 [vif5.2-guest-rx]
   29192 ?        S      0:00 [vif5.2-dealloc]
   [root@xenserver-uuffyqlx ~]#
   ```

3. Pin the threads to Domain-0 vCPUs using the following command:

   ```
   taskset –pc <core-id> <process-id>
   ```

   Example:

   ```
   taskset -pc 1 29189
   ```

Support for increasing NetScaler VPX disk space

NetScaler VPX supports a default disk space of 20 GB. If you encounter disk size constraints for various reasons, the following options are available to increase VPX disk space:

- Manually increase the primary disk size
- Dynamically increase the primary disk size
- Add a secondary disk

> **Note:**
>
> The ability to increase NetScaler VPX disk space is available for both VPX on-premises and VPX cloud deployments.

Manually increase the primary disk size on NetScaler VPX

Follow these steps to manually increase the VPX primary disk size using a Hypervisor or Cloud platform:

1. Shut down the VM.
2. Extend the default disk size from 20 GB to a higher value, such as 30 GB or 40 GB. For Azure, extend the default disk size from 32 GB to 64 GB.
3. Power on the VM and enter the boot prompt.
4. Log into single user mode using the `boot -s` command.
5. Verify the disk space. You can check the newly allocated disk space using `gpart show` command.
6. Note the partition name. In the following example, the VM partition is da0.
7. Resize the disk partition using the `gpart resize` command.

   Example:
   Let's resize the da0 MBR partition to include 10 GB free space by running the following command.

   ```
   gpart resize -i 1 da0
   ```

8. Merge the free space to the last partition.

   Example:

   ```
   gpart resize -i 5 da0s1
   ```

9. Extend the filesystem to include newly allocated free space using the "growfs"command.

   Example:


```
growfs /dev/da0s1e
```

10. Reboot the VM and verify the increased disk space using the "df -h"command on shell prompt.

Dynamically increase the primary disk size on NetScaler VPX

Starting from NetScaler release 14.1 build 21.x, administrators can dynamically increase the primary disk size on NetScaler VPX from 20 GB up to 1 TB at a time. For each subsequent increase, you can again extend up to 1 TB. Ensure you shut down the VM each time you increase the primary disk size. This allows the system to properly recognize the new disk size, update the partition table, and maintain system stability. To increase the disk space, extend the primary disk size by at least 1 GB in the respective cloud or hypervisor UI.

> **Note:**
>
> You can only increase the size of the disks. Once the new size is allocated, you cannot decrease it later. Therefore, increase the disk size only if it is essential.

Add a secondary disk

Starting from NetScaler release 13.1 build 21.x, you can increase disk space on the NetScaler VPX instance by adding a secondary disk. When you attach the secondary disk, the `/var/crash` directory is automatically mounted on this disk. The secondary disk is used for storing core files and logs. Existing directories for core files and log files continue to function as before.

> **Note:**
>
> Take an external backup before downgrading the NetScaler appliance to avoid loss of data.

For information on how to attach a new hard disk drive (HDD) to a NetScaler VPX instance on a cloud, see the following:

- Azure documentation

  > **Note:**
  >
  > To attach a secondary disk on VPX instances deployed on Azure, ensure that the Azure VM sizes have a local temporary disk. For more information, see Azure VM sizes with no local temporary disk.

- AWS documentation
- GCP documentation

> **Warning:**
>
> After you add a new HDD to VPX, some of the scripts that work on files, which are moved to the new HDD might fail under the following conditions:
>
> If you use the "link"shell command to create hard links to the files, which were moved to a new HDD.
>
> Replace all such commands with "ln -s"to use a symbolic link. Also, modify the failing scripts accordingly.

Apply NetScaler VPX configurations at the first boot of the NetScaler appliance in cloud

You can apply the NetScaler VPX configurations during the first boot of the NetScaler appliance in a cloud environment. This stage is addressed as the **preboot** stage in this document. Therefore in certain cases like ADC pooled licensing, a specific VPX instance is brought up in much lesser time. This feature is available in Microsoft Azure, Google Cloud platform, and AWS clouds.

What is user data

When you provision a VPX instance in a cloud environment, you have the option of passing user data to the instance. The user data allows you to perform common automated configuration tasks, customize the startup behaviors of instances, and run scripts after the instance starts. At the first boot, the NetScaler VPX instance performs the following tasks:

- Reads the user data.
- Interprets the configuration provided in user data.
- Applies the newly added configuration as it boots up.

How to provide preboot user data in cloud instance

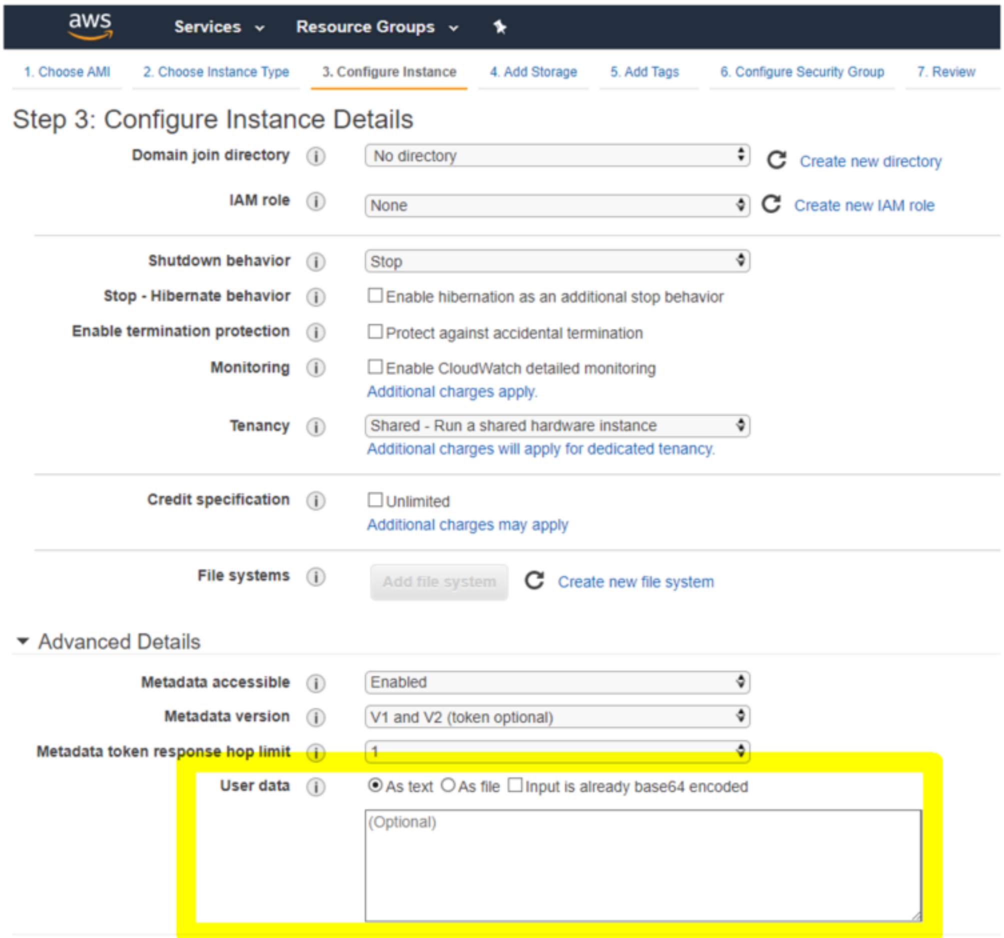
You can provide preboot user data to the cloud instance in XML format. Different clouds have different interfaces for providing user data.

Provide preboot user data using the AWS console

When you provision a NetScaler VPX instance using the AWS console, navigate to **Configure Instance Details > Advanced Details**, and provide the preboot user data configuration in the **User data** field.

For detailed instructions on each of the steps, see Deploy a NetScaler VPX instance on AWS by using the AWS web console.

For more information, see AWS documentation on Launching an instance.



> **Note:**
>
> AWS IMDSv2 only mode for the preboot user data feature is supported from NetScaler VPX release 13.1.48.x and later releases.

Provide preboot user data using AWS CLI

Type the following command in the AWS CLI:

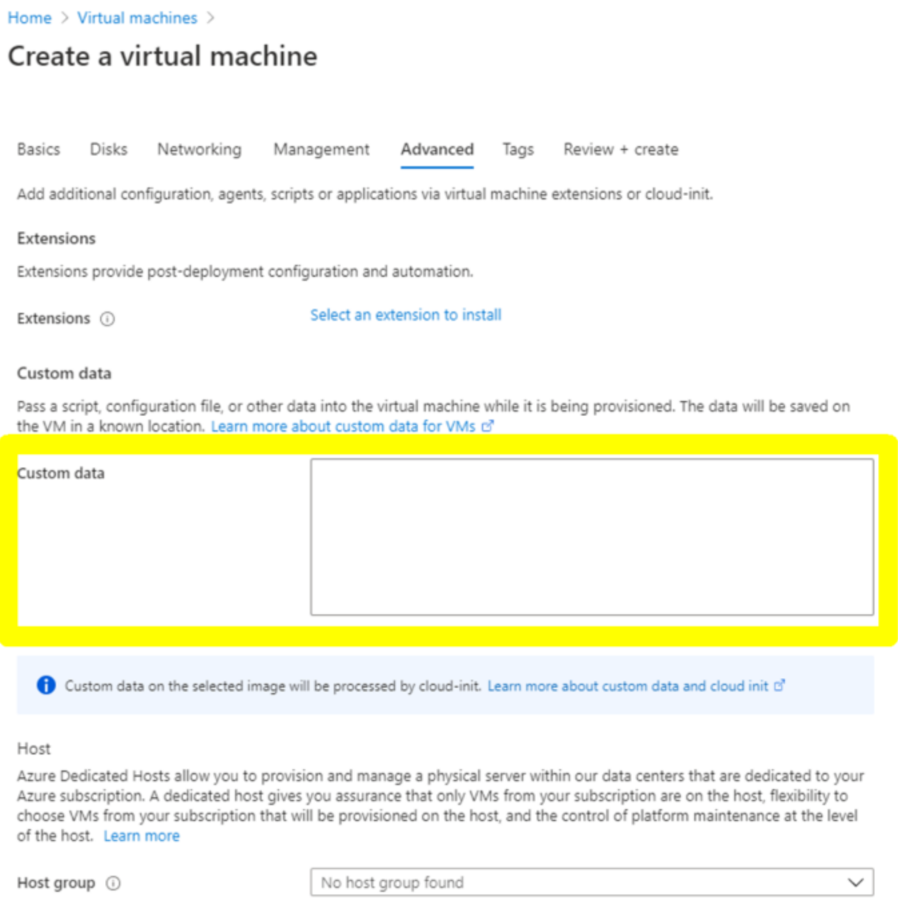
```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --user-data file://my_script.txt
```


For more information, see AWS documentation on Running instances.

For more information, see AWS documentation on Using instance user data

Provide preboot user data using the Azure console

When you provision a NetScaler VPX instance using Azure console, navigate to **Create a virtual machine > Advanced** tab. In the **Custom data** field, provide preboot user data configuration.



Provide preboot user data using the Azure CLI

Type the following command in the Azure CLI:

```
az vm create \
  --resource-group myResourceGroup \
  --name MyVm \
  --image debian \
  --custom-data MyCloudInitScript.txt \
```

Example:


```
az vm create --resource-group MyResourceGroup -name MyVm --image debian
    --custom-data MyCloudInitScript.txt
```

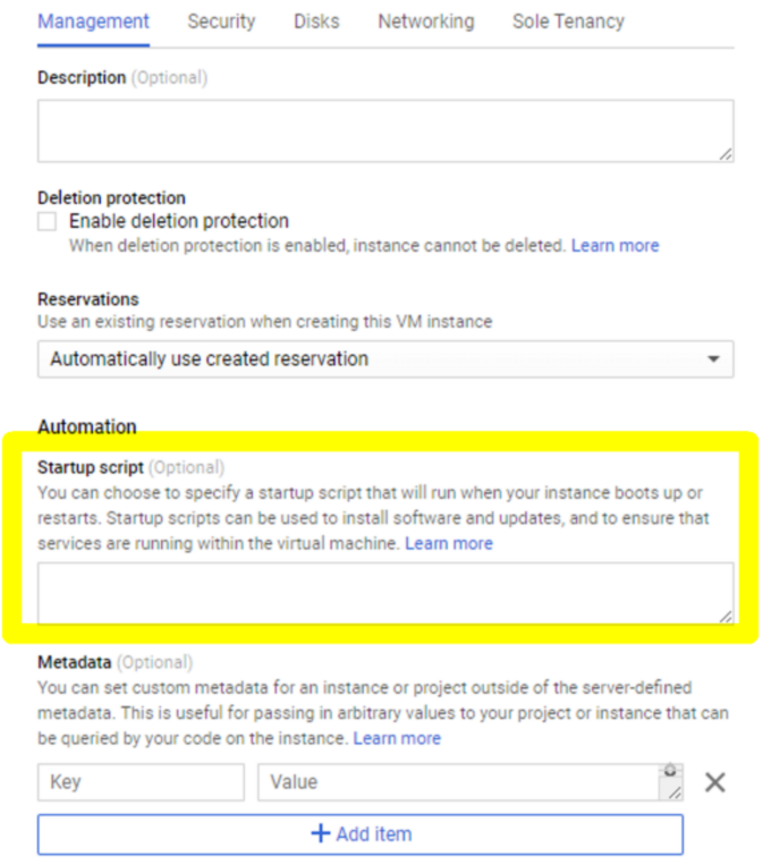
You can pass your custom data or preboot configuration as a file to "–custom-data"parameter. In this example, the file name is **MyCloudInitScript.txt**.

For more information, see Azure CLI documentation.

Provide preboot user data using the GCP console

When you provision a NetScaler VPX instance using GCP console, fill in the properties of instance. Expand **Management, security, disks, networking, sole tenancy**. Navigate to the **Management** tab. In the **Automation** section, provide preboot user data configuration in the **Startup Script** field.

For detailed information on creating the VPX instance using GCP, see Deploy a NetScaler VPX instance on Google Cloud Platform.



Provide preboot user data using the gcloud CLI

Type the following command in the GCP CLI:

```
gcloud compute instances create INSTANCE_NAMES --metadata-from-file=
    startup-script=LOCAL_FILE_PATH
```

metadata-from-file - Reads the value or user data from a file stored at the .

For more information, see gcloud CLI documentation

Preboot user data format

The preboot user data must be provided to the cloud instance in XML format. The NetScaler preboot user data that you provide through the cloud infrastructure during boot can comprise the following four sections:

- NetScaler configuration represented with the `<NS-CONFIG>` tag.
- Custom bootstrapping the NetScaler represented with the `<NS-BOOTSTRAP>` tag.
- Storing user-scripts in NetScaler represented with the `<NS-SCRIPTS>` tag.
- Pooled licensing configuration represented with the `<NS-LICENSE-CONFIG>` tag.

You can provide the preceding four sections in any order within the ADC preboot configuration. Ensure to strictly follow the formatting shown in the following sections while providing the preboot user data.

> **Note:**
>
> The entire preboot user data configuration must be enclosed in the `<NS-PRE-BOOT-CONFIG>` tag as shown in the following examples.

Example 1:

```
<NS-PRE-BOOT-CONFIG>
     <NS-CONFIG>          </NS-CONFIG>
     <NS-BOOTSTRAP>       </NS-BOOTSTRAP>
     <NS-SCRIPTS>         </NS-SCRIPTS>
     <NS-LICENSE-CONFIG>  </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
```

Example 2:

```
<NS-PRE-BOOT-CONFIG>
    <NS-LICENSE-CONFIG> </NS-LICENSE-CONFIG>
    <NS-SCRIPTS>        </NS-SCRIPTS>
    <NS-BOOTSTRAP>      </NS-BOOTSTRAP>
    <NS-CONFIG>         </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
```

Use the <NS-CONFIG> tag to provide the specific NetScaler VPX configurations that needs to be applied to the VPX instance at the preboot stage.

> **Note:**
>
> The <NS-CONFIG> section must have valid ADC CLI commands. The CLIs are not verified for the syntactic errors or format.

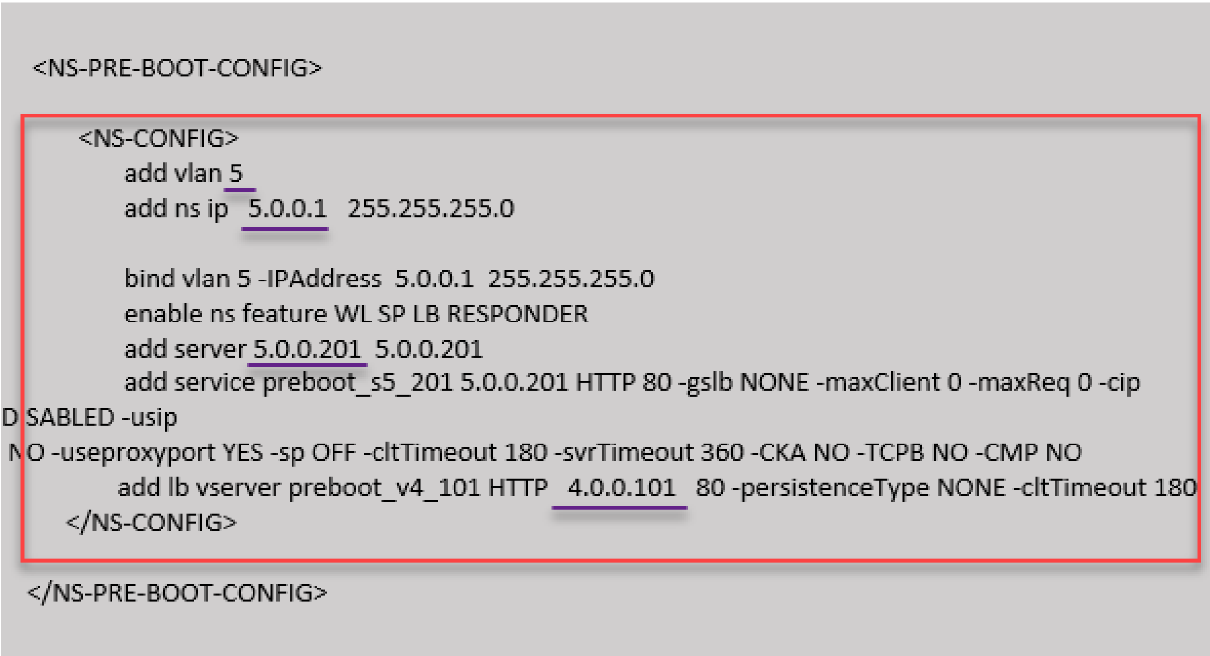
NetScaler configurations

Use the <NS-CONFIG> tag to provide the specific NetScaler VPX configurations that needs to be applied to the VPX instance at the preboot stage.

> **Note:**
>
> The <NS-CONFIG> section must have valid ADC CLI commands. The CLIs are not verified for the syntactic errors or format.

Example:

In the following example, the <NS-CONFIG> section has the details of the configurations. A VLAN of ID '5'is configured and bound to the SNIP (5.0.0.1). A load balancing virtual server (4.0.0.101) is also configured.



```
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add vlan 5
        add ns ip  5.0.0.1  255.255.255.0

        bind vlan 5 -IPAddress  5.0.0.1  255.255.255.0
        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201  5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip
DISABLED -usip
NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
        add lb vserver preboot_v4_101 HTTP  4.0.0.101  80 -persistenceType NONE -cltTimeout 180
    </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
```

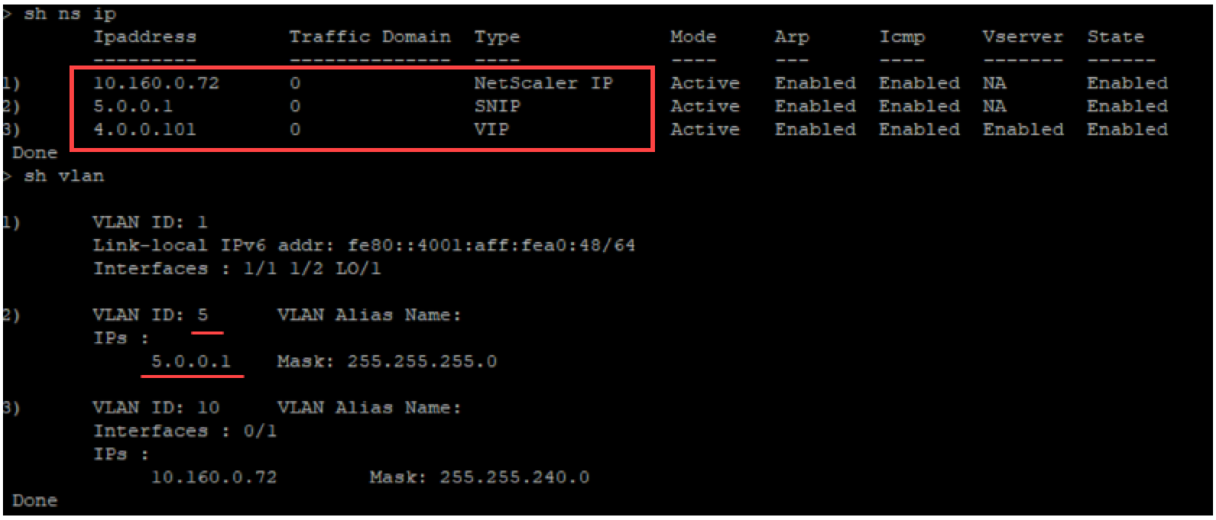
You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
        <NS-CONFIG>
```



```
            add vlan 5
            add ns ip 5.0.0.1 255.255.255.0
            bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
            enable ns feature WL SP LB RESPONDER
            add server 5.0.0.201 5.0.0.201
            add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
                maxClient 0 -maxReq 0 -cip DISABLED -usip
NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -
    TCPB NO -CMP NO
            add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
                persistenceType NONE -cltTimeout 180
      </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
```

The NetScaler VPX instance comes up with the configuration applied in the <NS-CONFIG> section as shown in the following illustrations.




```
> sh server
1)      Name:           5.0.0.201       State:ENABLED
        IPAddress:          5.0.0.201
2)      Name:   169.254.169.254         State:ENABLED
        IPAddress: 169.254.169.254
 Done
> stat service

Service(s) Summary
                          IP  port          Type         State     Req/s
preb...5_201       5.0.0.201    80          HTTP          DOWN       0/s
gcpl...vice0 169.254.169.254    53           DNS            UP       0/s
 Done
> sh service preboot_s5_201
        preboot_s5_201 (5.0.0.201:80) - HTTP
        State: DOWN
        Last state change was at Tue Dec 29 07:18:28 2020
        Time since last state change: 0 days, 00:05:02.820
        Server Name: 5.0.0.201
        Server ID : None        Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Monitoring Owner: 0
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
        Monitor Connection Close : NONE
        Appflow logging: ENABLED
        Process Local: DISABLED
```

User scripts

Use the `<NS-SCRIPTS>` tag to provide any script that must be stored and ran in NetScaler VPX instance.

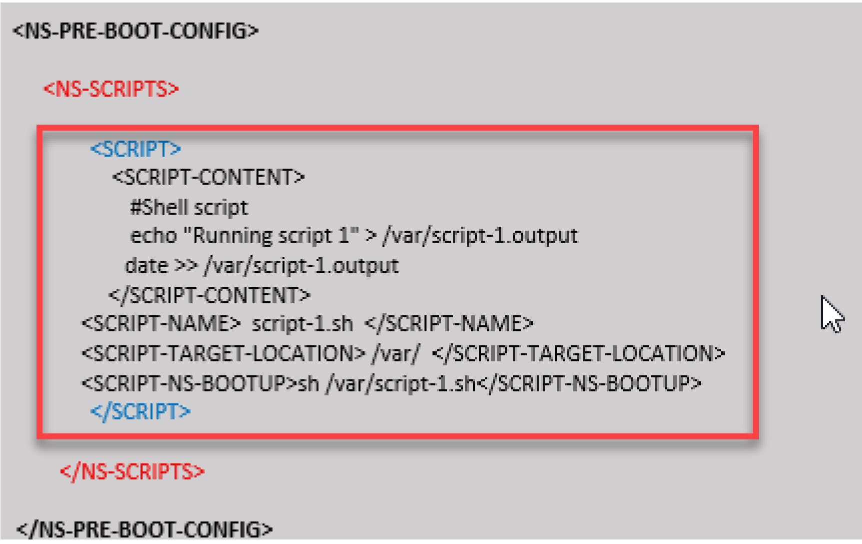
You can include many scripts within the `<NS-SCRIPTS>` tag. Each script must be included within the `<SCRIPT>` tag.
Each `<SCRIPT>` section corresponds to one script and contains all the details of the script using the following sub tags.

- **`<SCRIPT-NAME>`:** Indicates the name of the script file that must be stored.
- **`<SCRIPT-CONTENT>`:** Indicates the content of the file that must be stored.
- **`<SCRIPT-TARGET-LOCATION>`:** Indicates the designated target location where this file must be stored. If the target location is not provided, by default, the file, or script is saved in the "/nsconfig" directory.
- **`<SCRIPT-NS-BOOTUP>`:** Specify the commands that you use to run the script.

  - If you use the <SCRIPT-NS-BOOTUP> section, the commands provided in the section are stored in "/nsconfig/nsafter.sh", and the commands are run after the packet engine boots up as part of "nsafter.sh"execution.
  - If you do not use the <SCRIPT-NS-BOOTUP> section, the script file is stored in the target location that you specify.

Example 1:

In this example, the <NS-SCRIPTS> tag contains details of only one script: script-1.sh. The "script-1.sh"script is saved at the "/var"directory. The script is populated with the specified contents, and is run with the "sh /var/script-1.sh"command after packet engine boots up.



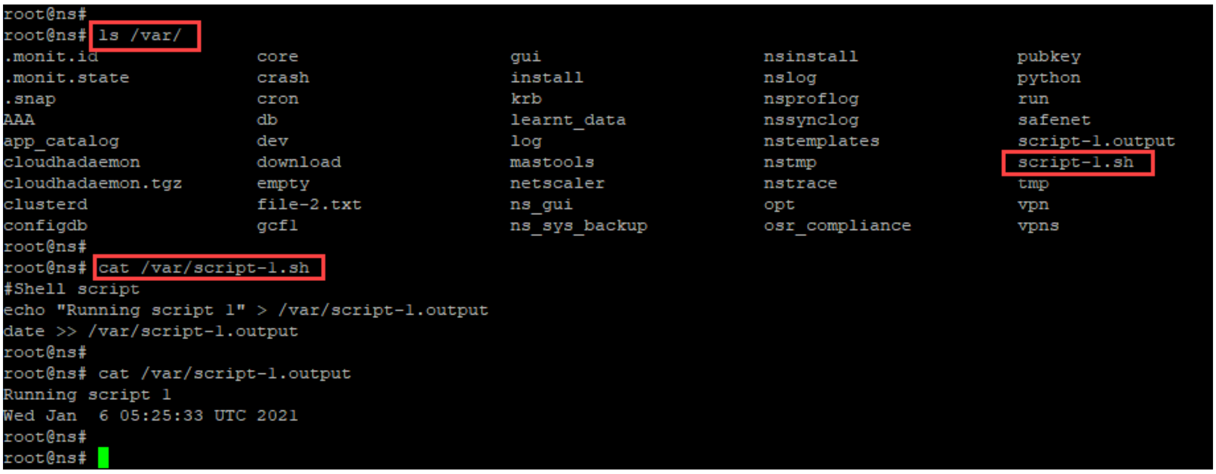
```
<NS-PRE-BOOT-CONFIG>
  <NS-SCRIPTS>
    <SCRIPT>
      <SCRIPT-CONTENT>
        #Shell script
        echo "Running script 1" > /var/script-1.output
        date >> /var/script-1.output
      </SCRIPT-CONTENT>
    <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
    <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION>
    <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
    </SCRIPT>
  </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```

You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
    <SCRIPT>
            <SCRIPT-CONTENT>
                #Shell script
                echo "Running script 1" > /var/script-1.output
                date >> /var/script-1.output
            </SCRIPT-CONTENT>

                <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
                <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION>
                <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```

In the following snapshot, you can verify that "script-1.sh"script is saved in the "/var/"directory. The "Script-1.sh"script is run, and the output file is created appropriately.

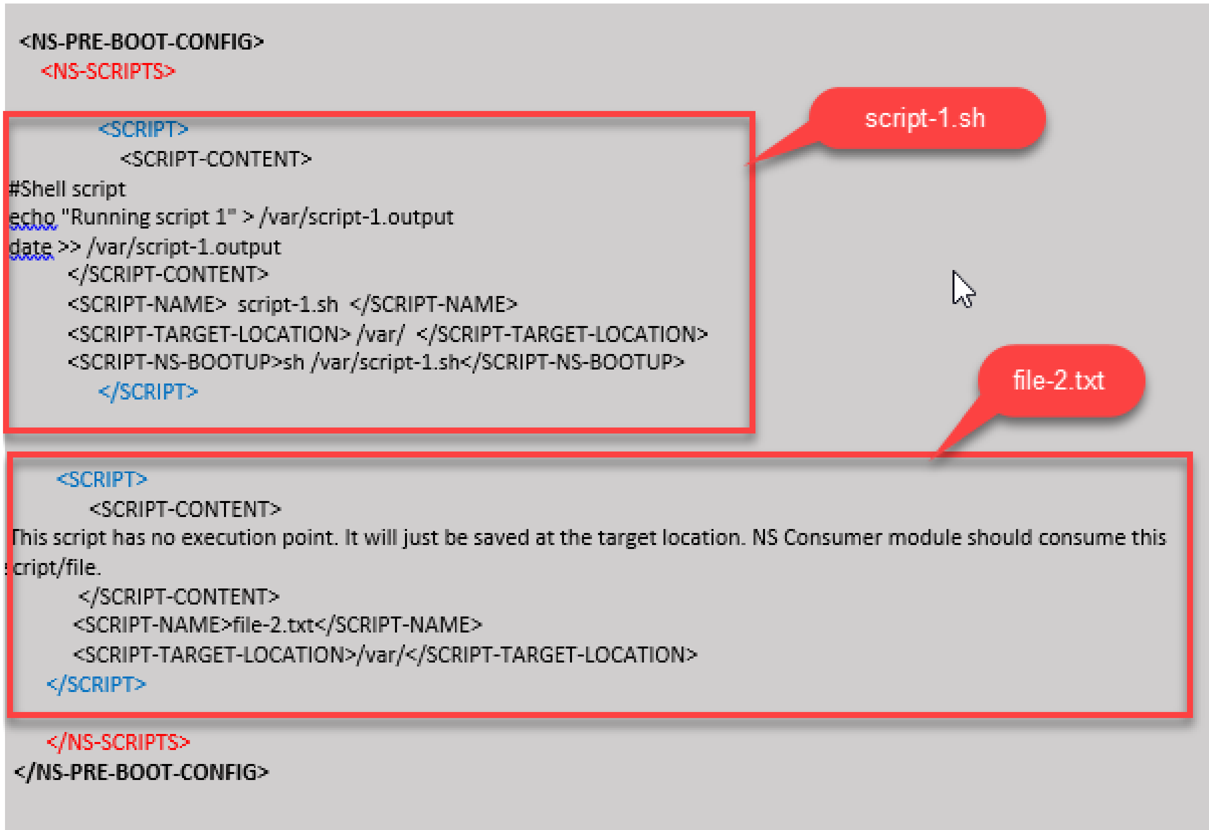


```
root@ns#
root@ns# ls /var/
.monit.id          core          gui           nsinstall      pubkey
.monit.state       crash         install       nslog          python
.snap              cron          krb           nsproflog      run
AAA                db            learnt_data   nssynclog      safenet
app_catalog        dev           log           nstemplates    script-1.output
cloudhadaemon      download      mastools      nstmp          script-1.sh
cloudhadaemon.tgz  empty         netscaler     nstrace        tmp
clusterd           file-2.txt    ns_gui        opt            vpn
configdb           gcfl          ns_sys_backup osr_compliance vpns
root@ns#
root@ns# cat /var/script-1.sh
#Shell script
echo "Running script 1" > /var/script-1.output
date >> /var/script-1.output
root@ns#
root@ns# cat /var/script-1.output
Running script 1
Wed Jan  6 05:25:33 UTC 2021
root@ns#
root@ns#
```

Example 2:

In the following example, the <NS-SCRIPTS> tag contains details of two scripts.

- The first script is saved as "script-1.sh"at the "/var"directory. The script is populated with the specified contents, and is run with command "sh /var/script-1.sh"after packet engine boots up.
- The second script is saved as "file-2.txt"at the "/var"directory. This file is populated with the specified contents. But it is not run because the bootup execution command <SCRIPT-NS-BOOTUP> is not provided.




```
<NS-PRE-BOOT-CONFIG>
  <NS-SCRIPTS>

        <SCRIPT>
          <SCRIPT-CONTENT>
#Shell script
echo "Running script 1" > /var/script-1.output
date >> /var/script-1.output
      </SCRIPT-CONTENT>
      <SCRIPT-NAME>  script-1.sh  </SCRIPT-NAME>
      <SCRIPT-TARGET-LOCATION> /var/  </SCRIPT-TARGET-LOCATION>
      <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
        </SCRIPT>

     <SCRIPT>
        <SCRIPT-CONTENT>
This script has no execution point. It will just be saved at the target location. NS Consumer module should consume this script/file.
       </SCRIPT-CONTENT>
      <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
      <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
    </SCRIPT>

   </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```

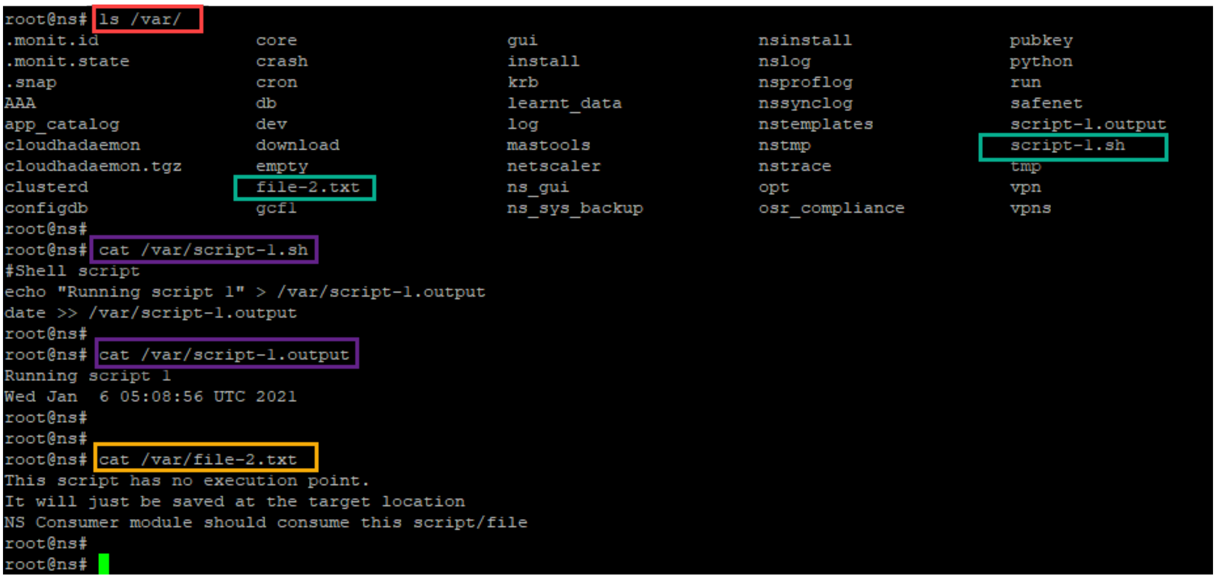

You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
        <SCRIPT>
            <SCRIPT-CONTENT>
               #Shell script
               echo "Running script 1" > /var/script-1.output
               date >> /var/script-1.output
            </SCRIPT-CONTENT>

            <SCRIPT-NAME>  script-1.sh  </SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION> /var/  </SCRIPT-TARGET-LOCATION>
            <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
            </SCRIPT>

        <SCRIPT>
            <SCRIPT-CONTENT>
                This script has no execution point.
                It will just be saved at the target location
                NS Consumer module should consume this script/file
            </SCRIPT-CONTENT>
            <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```

In the following snapshot, you can verify that script-1.sh and file-2.txt are created in the "/var/" directory. The Script-1.sh is run, and the output file is created appropriately.



Licensing

Use the <NS-LICENSE-CONFIG> tag to apply NetScaler pooled licensing while booting up the VPX instance. Use the <LICENSE-COMMANDS> tag within <NS-LICENSE-CONFIG> section to provide the pooled license commands. These commands must be syntactically valid.

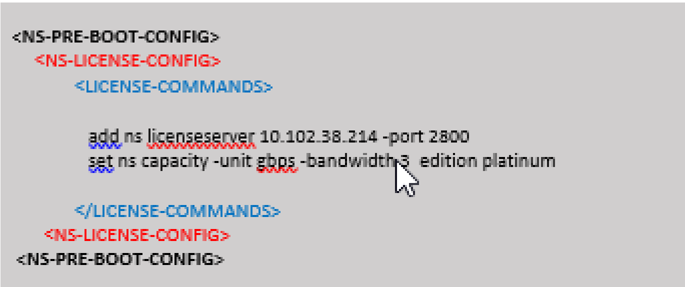
You can specify the pooled licensing details such as, license type, capacity, and license server in the <LICENSE-COMMANDS> section using the standard pooled licensing commands. For more information, see Configure NetScaler pooled capacity licensing.

After applying the <NS-LICENSE-CONFIG>, the VPX comes up with the requested edition upon boot, and VPX tries to check out the configured licenses from the license server.

- If the license checkout is successful, the configured bandwidth is applied to VPX.
- If the license checkout fails, the license is not retrieved from license server within 10–12 minutes approximately. As a result, the system reboots and enters an unlicensed state.

Example:

In the following example, after applying the <NS-LICENSE-CONFIG>, the VPX comes up with the Premium edition upon boot, and VPX tries to check out the configured licenses from the license server (10.102.38.214).



```
<NS-PRE-BOOT-CONFIG>
  <NS-LICENSE-CONFIG>
      <LICENSE-COMMANDS>

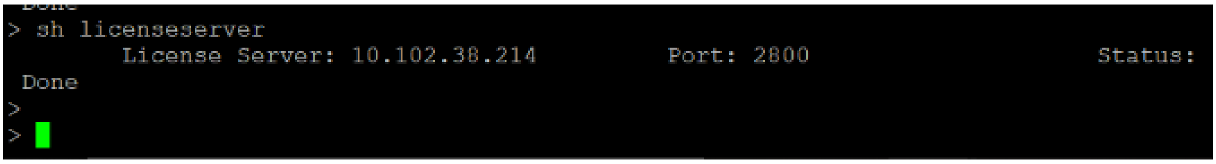
        add ns licenseserver 10.102.38.214 -port 2800
        set ns capacity -unit gbps -bandwidth 3  edition platinum

      </LICENSE-COMMANDS>
   <NS-LICENSE-CONFIG>
<NS-PRE-BOOT-CONFIG>
```

You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
   <NS-LICENSE-CONFIG>
        <LICENSE-COMMANDS>
            add ns licenseserver 10.102.38.214 -port 2800
            set ns capacity -unit gbps -bandwidth 3  edition platinum
        </LICENSE-COMMANDS>
    </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
```

As shown in the following illustration, you can run the "show license server" command, and verify that the license server (10.102.38.214) is added to the VPX.



Bootstrapping

Use the <NS-BOOTSTRAP> tag to provide the custom bootstrapping information. You can use the <SKIP-DEFAULT-BOOTSTRAP> and <NEW-BOOTSTRAP-SEQUENCE> tags within the <NS-BOOTSTRAP> section. This section informs NetScaler appliance whether to avoid the default bootstrap or not. If the default bootstrapping is avoided, this section provides you an option to provide a new bootstrapping sequence.

Default bootstrap configuration

The default bootstrap configuration in NetScaler appliance follows these interface assignments:

- **Eth0** - Management interface with a certain NSIP address.
- **Eth1** - Client-facing interface with a certain VIP address.
- **Eth2** - Server-facing interface with a certain SNIP address.

Customize bootstrap configuration

You can skip the default bootstrap sequence and provide a new bootstrap sequence for the NetScaler VPX instance. Use the <NS-BOOTSTRAP> tag to provide the custom bootstrapping information. For example, you can change the default bootstrapping, where the Management interface (NSIP), Client-facing interface (VIP), and server-facing interface (SNIP) are always provided in certain order.

The following table indicates the bootstrapping behavior with the different values that are allowed for <SKIP-DEFAULT-BOOTSTRAP> and <NEW-BOOTSTRAP-SEQUENCE> tags.

SKIP-DEFAULT-BOOTSTRAP	NEW-BOOTSTRAP-SEQUENCE	Bootstrap behavior
YES	YES	The default bootstrapping behavior is skipped, and a new custom bootstrap sequence provided in the <NS-BOOTSTRAP> section is run.

SKIP-DEFAULT-BOOTSTRAP	NEW-BOOTSTRAP-SEQUENCE	Bootstrap behavior
YES	NO	The default bootstrapping behavior is skipped. The bootstrap commands provided in the <NS-CONFIG> section is run.

You can customize the bootstrap configuration by the following three methods:

- Provide only the interface details
- Provide the interface details along with IP addresses and subnet mask
- Provide bootstrap related commands in the <NS-CONFIG> section

Method 1: Custom bootstrap by specifying only the interface details

You specify the management, client-facing and server-facing interfaces but not their IP addresses and subnet masks. The IP addresses and subnet masks are populated by querying the cloud infrastructure.

Custom bootstrap example for AWS

You provide the custom bootstrap sequence as shown in the following example. For more information, see How to provide preboot user data in cloud instance. Eth1 interface is assigned as the management interface (NSIP), Eth0 interface as the client interface (VIP), and Eth2 interface as the server interface (SNIP). The <NS-BOOTSTRAP> section contains only the interface details and not the details of IP addresses and subnet masks.


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

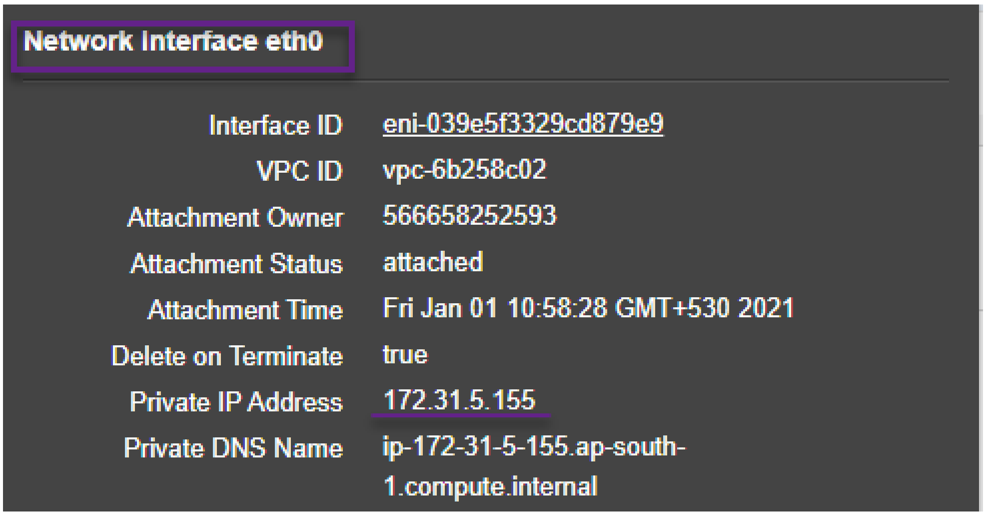
    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

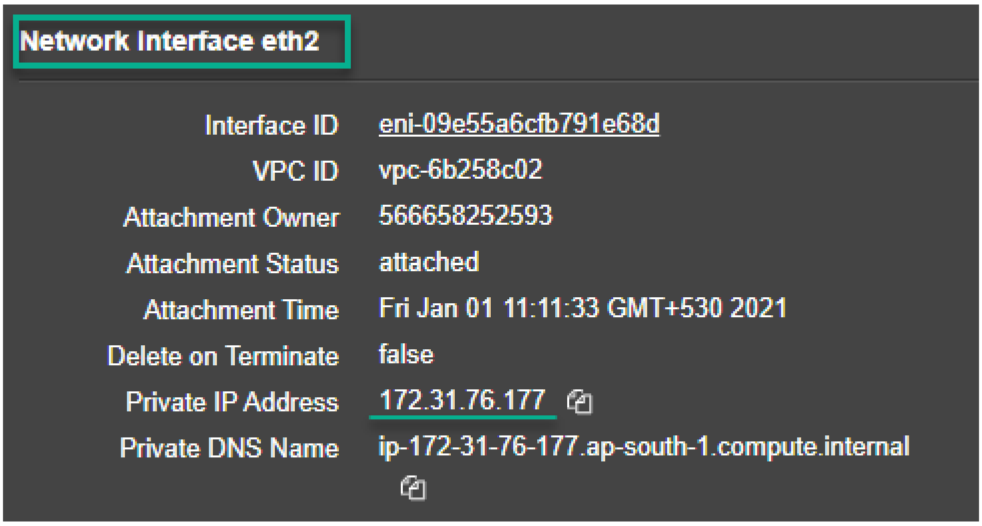
After the VM instance is created, in the AWS portal, you can verify the network interface properties as follows:

1. Navigate to the **AWS Portal > EC2 instances**, and select the instance that you have created by providing the custom bootstrap information.
2. In the **Description** tab, you can verify the properties of each network interface as shown in the following illustrations.

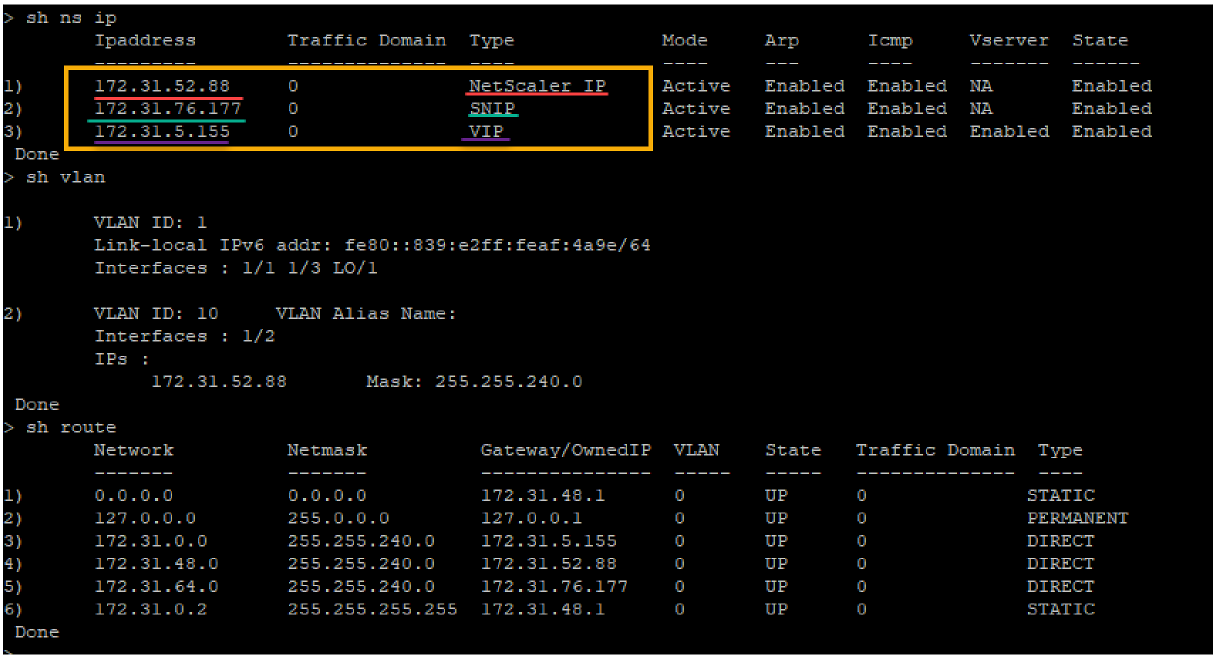
Network Interface eth1

Interface ID	eni-021961099be6815eb
VPC ID	vpc-6b258c02
Attachment Owner	566658252593
Attachment Status	attached
Attachment Time	Fri Jan 01 11:11:23 GMT+530 2021
Delete on Terminate	false
Private IP Address	172.31.52.88
Private DNS Name	ip-172-31-52-88.ap-south-1.compute.internal





You can run the `show nsip` command in **ADC CLI**, and verify the network interfaces applied to the NetScaler VPX instance during the first boot of the ADC appliance.



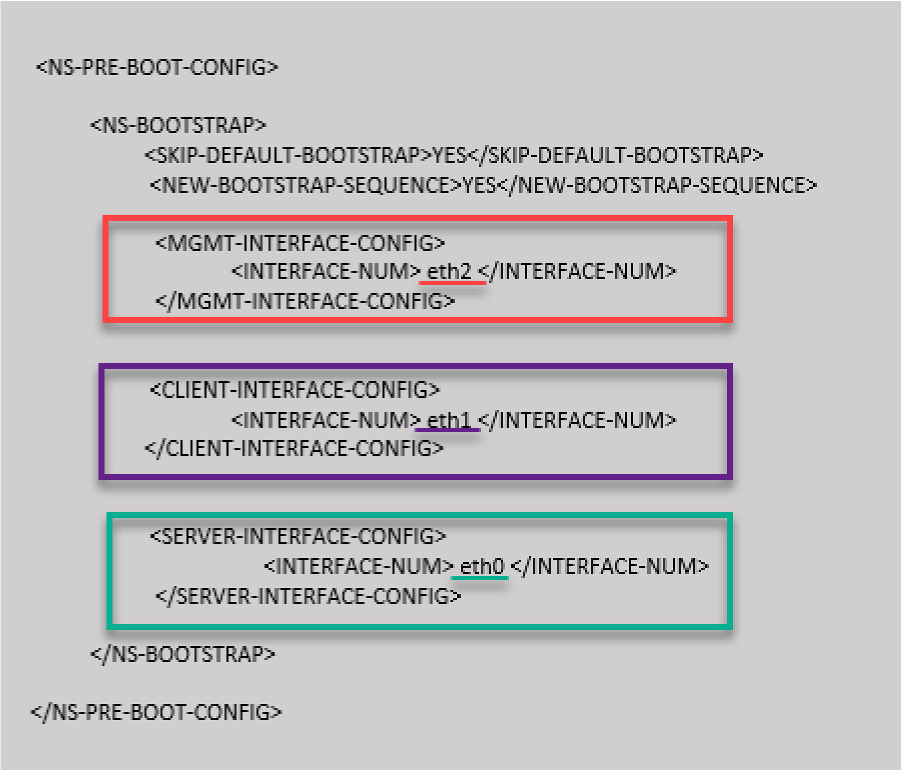
```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10      VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88        Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  ----    -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0              STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0              PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0              DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0              DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0              DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0              STATIC
 Done
```

Custom bootstrap example for Azure

You provide the custom bootstrap sequence as shown in the following example. For more information, see How to provide preboot user data in cloud instance. Eth2 interface is assigned as the management interface (NSIP), Eth1 interface as the client interface (VIP), and Eth0 interface as the server interface (SNIP). The `<NS-BOOTSTRAP>` section contains only the interface details and not the details of IP addresses and subnet masks.



```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

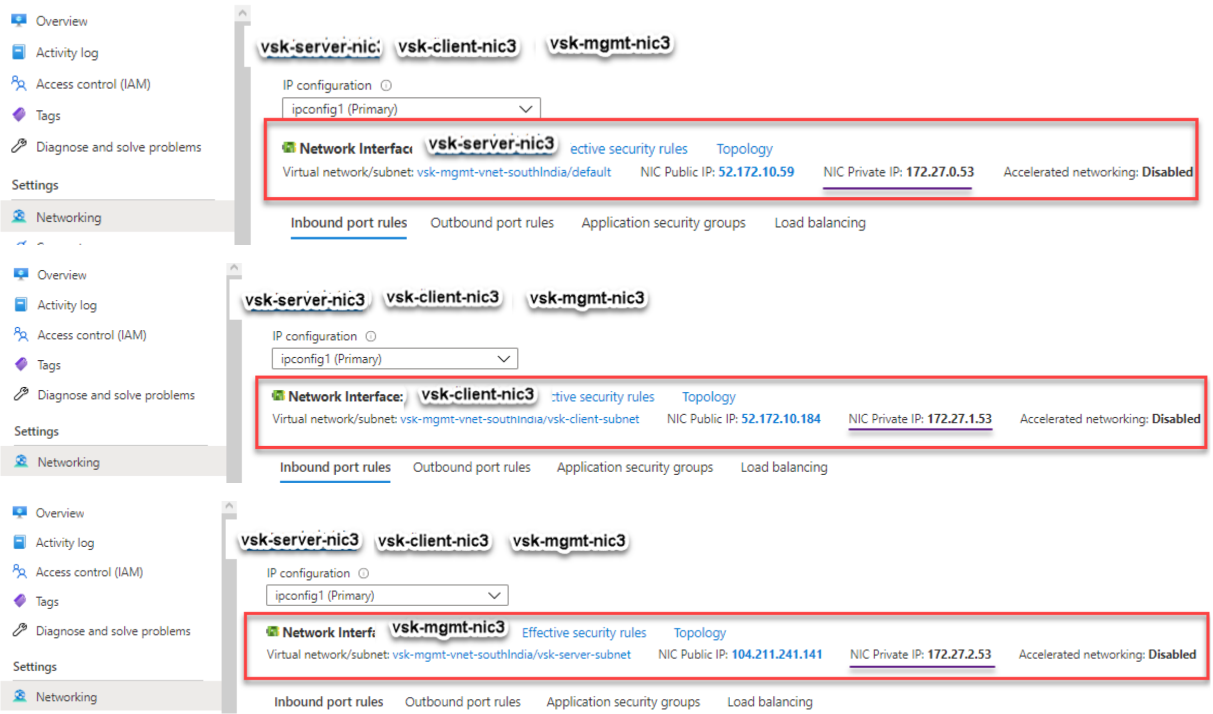
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

You can see that the NetScaler VPX instance is created with three network interfaces. Navigate to the **Azure portal > VM instance > Networking**, and verify the networking properties of the three NICs as shown in the following illustrations.



You can run the "show nsip"command in the ADC CLI, and verify that the new bootstrap sequence specified in the <NS-BOOTSTRAP> section is applied. You can run the "show route"command to verify the subnet mask.

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN     State    Traffic Domain  Type
        -------          -------          ---------------  -----    -----    --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0        UP       0              STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0        UP       0              PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0        UP       0              DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0        UP       0              DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0        UP       0              DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0        UP       0              STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0        UP       0              STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0        UP       0              STATIC
 Done
>
```

Custom bootstrap examples for GCP

You provide the custom bootstrap sequence as shown in the following example. For more information, see How to provide preboot user data in cloud instance. Eth1 interface is assigned as the management interface (NSIP), Eth0 interface as the client interface (VIP), and Eth2 interface as the server interface (SNIP). The <NS-BOOTSTRAP> section contains only the interface details and not the details of IP addresses and subnet masks.

```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

After the VM instance is created in the GCP portal, you can verify the network interface properties as follows:

1. Select the instance that you have created by providing the custom bootstrap information.
2. Navigate to the Network interface properties and verify the NIC details as follows:

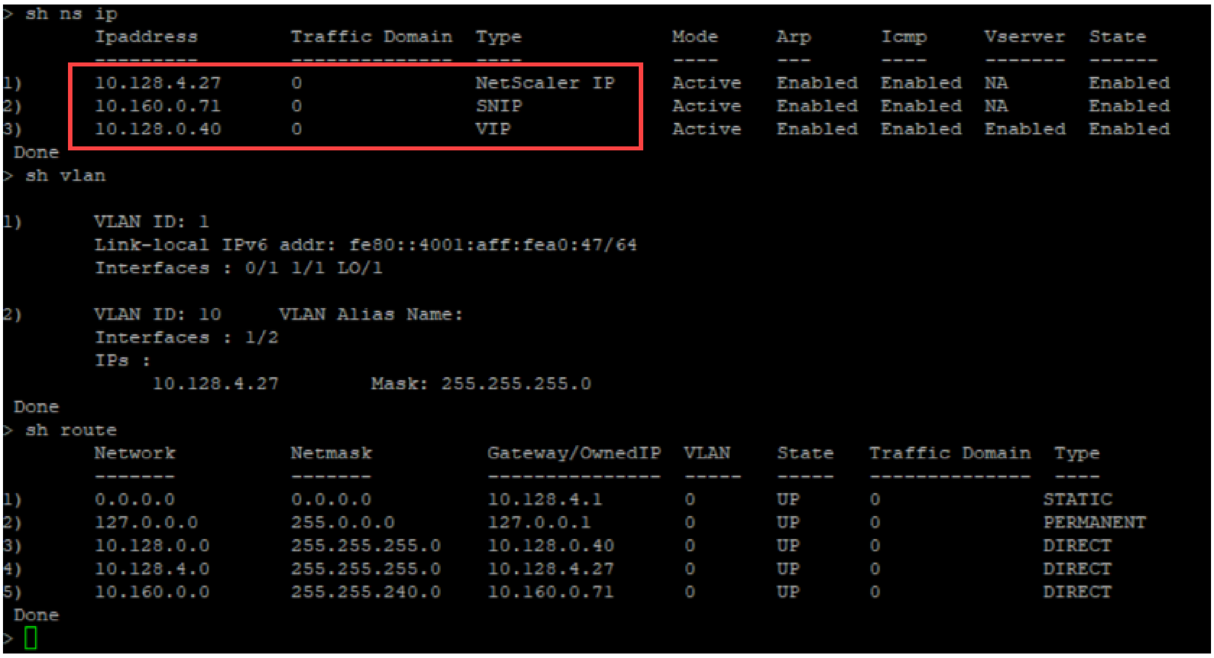
Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	10.160.0.71	–	35.244.56.180 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	10.128.0.40	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	10.128.4.27	–	34.93.241.147 (ephemeral)	Premium		View details

Public DNS PTR Record

None

You can run the `show nsip` command in **ADC CLI**, and verify the network interfaces applied to the NetScaler VPX instance during the first boot of the ADC appliance.



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.4.27      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.160.0.71      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.128.0.40      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:47/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10      VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             10.128.4.27        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  ----    -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.4.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.40      0       UP      0               DIRECT
4)      10.128.4.0       255.255.255.0    10.128.4.27      0       UP      0               DIRECT
5)      10.160.0.0       255.255.240.0    10.160.0.71      0       UP      0               DIRECT
 Done
>
```

Method 2: Custom bootstrap by specifying the interfaces, IP addresses, and subnet masks

You specify the management, client-facing and server-facing interfaces along with their IP addresses and subnet mask.

Custom bootstrap examples for AWS

In the following example, you skip the default bootstrap and run a new bootstrap sequence for the NetScaler appliance. For the new bootstrap sequence, you specify the following details:

- **Management interface:** Interface - Eth1, NSIP - 172.31.52.88, and subnet mask - 255.255.240.0
- **Client facing interface:** Interface - Eth0, VIP - 172.31.5.155, and subnet mask - 255.255.240.0.
- **Server facing interface:** Interface - Eth2, SNIP - 172.31.76.177, and subnet mask - 255.255.240.0.

```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth1 </INTERFACE-NUM>
                <IP> 172.31.52.88 </IP>
                <SUBNET-MASK> 255.255.240.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>

        <CLIENT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 172.31.5.155 </IP>
                <SUBNET-MASK> 255.255.240.0 </SUBNET-MASK>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 172.31.76.177 </IP>
            <SUBNET-MASK> 255.255.240.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence specified in the `<NS-BOOTSTRAP>` section is applied. You can run the "show route"command to verify the subnet mask.

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0              STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0              PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0              DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0              DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0              DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0              STATIC
 Done
>
```

Custom bootstrap example for Azure

In the following example, a new bootstrap sequence for ADC is mentioned and default bootstrap is skipped. You provide the interface details along with the IP addresses and subnet masks as follows:

- Management interface (eth2), NSIP (172.27.2.53), and subnet mask (255.255.255.0)
- Client facing interface (eth1), VIP (172.27.1.53), and subnet mask (255.255.255.0)
- Server facing interface (eth0), SNIP (172.27.0.53), and subnet mask (255.255.255.0)


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 172.27.2.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>

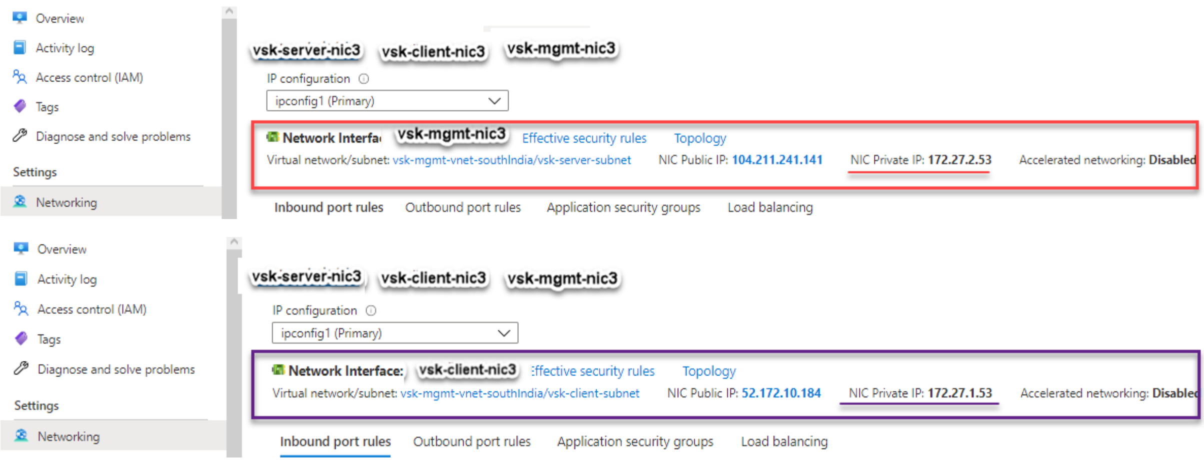
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 172.27.1.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
            </CLIENT-INTERFACE-CONFIG>

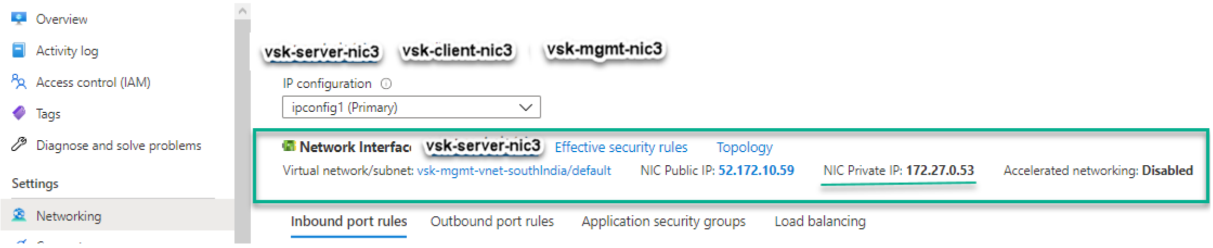
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 172.27.0.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

You can see that the NetScaler VPX instance is created with three network interfaces. Navigate to the **Azure portal > VM instance > Networking**, and verify the networking properties of the three NICs as shown in the following illustrations.





You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence specified in the <`NS-BOOTSTRAP`> section is applied. You can run the "show route"command to verify the subnet mask.

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

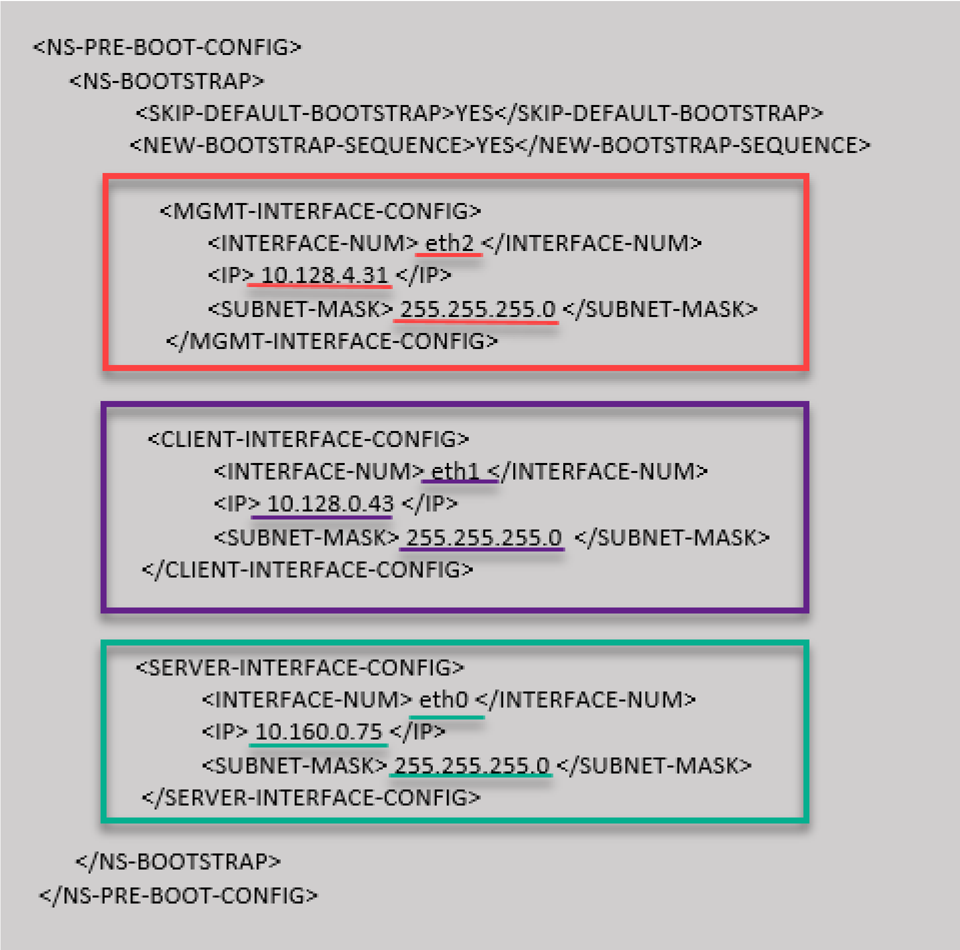
1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN   State  Traffic Domain  Type
        -------          -------          ---------------  -----  -----  --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0      UP     0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0      UP     0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0      UP     0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0      UP     0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0      UP     0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0      UP     0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0      UP     0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0      UP     0               STATIC
 Done
```

Custom bootstrap example for GCP

In the following example, a new bootstrap sequence for ADC is mentioned and default bootstrap is skipped. You provide the interface details along with the IP addresses and subnet masks as follows:

- Management interface (eth2), NSIP (10.128.4.31), and subnet mask (255.255.255.0)
- Client facing interface (eth1), VIP (10.128.0.43), and subnet mask (255.255.255.0)
- Server facing interface (eth0), SNIP (10.160.0.75), and subnet mask (255.255.255.0)



```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 10.128.4.31 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>

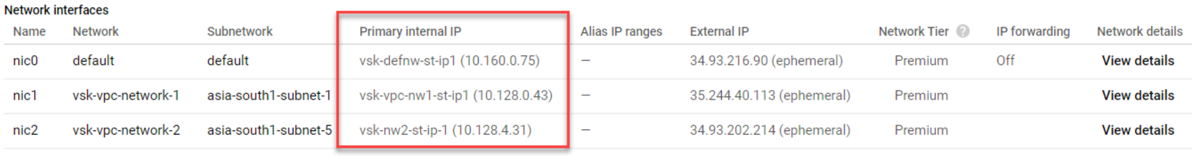
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 10.128.0.43 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 10.160.0.75 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

After the VM instance is created in the GCP portal with the custom bootstrap, you can verify the network interface properties as follows:

1. Select the instance that you have created by providing the custom bootstrap information.
2. Navigate to the Network interface properties and verify the NIC details as follows.



Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	vsk-defnw-st-ip1 (10.160.0.75)	–	34.93.216.90 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	vsk-vpc-nw1-st-ip1 (10.128.0.43)	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	vsk-nw2-st-ip-1 (10.128.4.31)	–	34.93.202.214 (ephemeral)	Premium		View details

You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence specified in the `<NS-BOOTSTRAP>` section is applied. You can run the "show route"command to verify the subnet mask.


```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.4.31      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.160.0.75      0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      10.128.0.43      0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:4b/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             10.128.4.31        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN     State    Traffic Domain  Type
        -------          -------          ---------------  -----    -----    --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.4.1       0        UP       0              STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0        UP       0              PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.43      0        UP       0              DIRECT
4)      10.128.4.0       255.255.255.0    10.128.4.31      0        UP       0              DIRECT
5)      10.160.0.0       255.255.255.0    10.160.0.75      0        UP       0              DIRECT
 Done
>
```

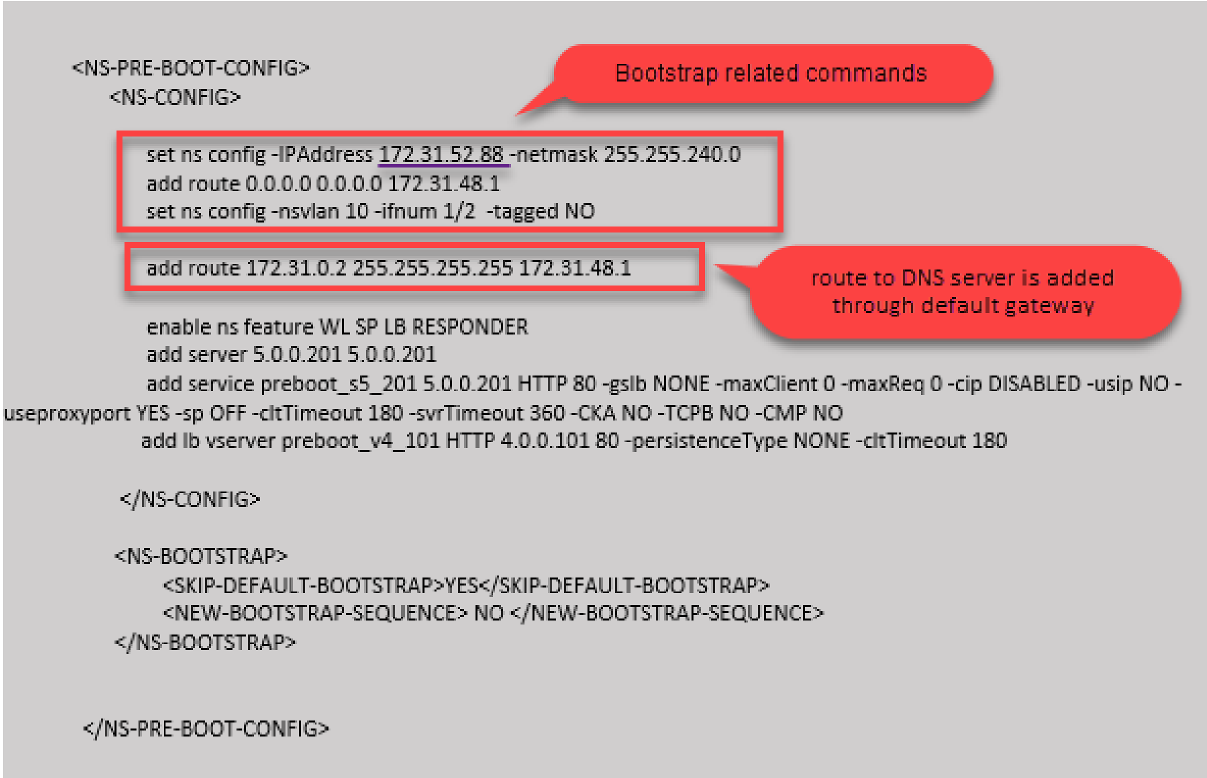
Method 3: Custom bootstrap by providing bootstrap related commands in the `<NS-CONFIG>` section

You can provide the bootstrap related commands in the <`NS-CONFIG`> section. In the <`NS-BOOTSTRAP`> section, you must specify the <`NEW-BOOTSTRAP-SEQUENCE`> as "No"to run the bootstrapping commands in the <`NS-CONFIG`> section. You must also provide the commands to assign NSIP, default route, and NSVLAN. In addition, provide the commands relevant for the cloud that you use.

Before providing a custom bootstrap, ensure that your cloud infrastructure supports a particular interface configuration.

Custom bootstrap example for AWS

In this example, bootstrap related commands are provided in the <`NS-CONFIG`> section. The <`NS-BOOTSTRAP`> section indicates that the default bootstrapping is skipped, and the custom bootstrap information provided in the <`NS-CONFIG`> section is run. You must also provide the commands to create NSIP, add default route, and add NSVLAN.



You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>

        set ns config -IPAddress 172.31.52.88 -netmask 255.255.240.0
        add route 0.0.0.0 0.0.0.0 172.31.48.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add route 172.31.0.2 255.255.255.255 172.31.48.1

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -persistenceType NONE -cltTimeout 180

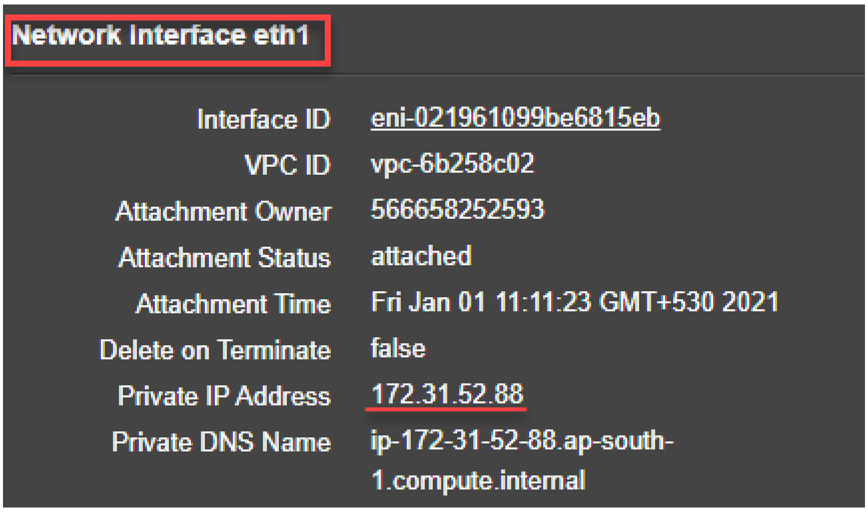
    </NS-CONFIG>

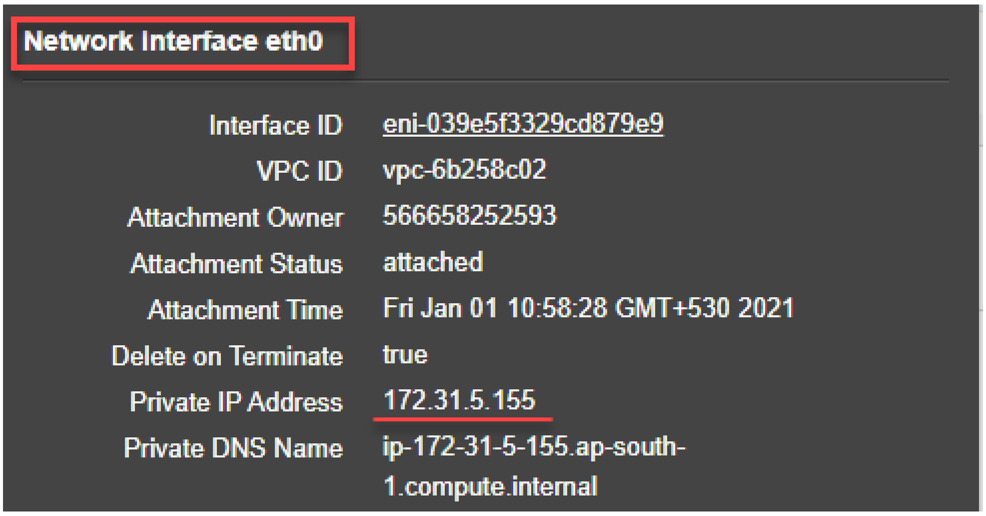
    <NS-BOOTSTRAP>
     <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
     <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
    </NS-BOOTSTRAP>

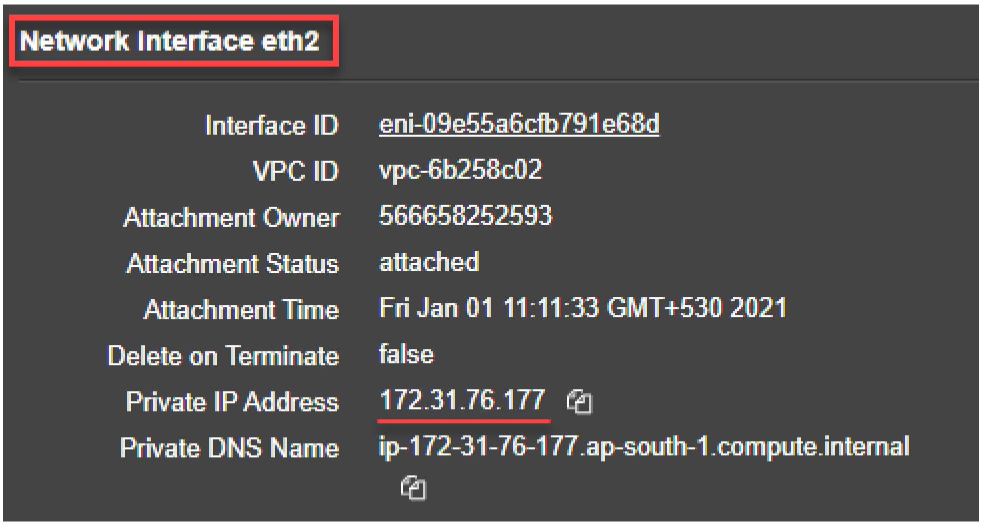

</NS-PRE-BOOT-CONFIG>
```


After the VM instance is created, in the AWS portal, you can verify the network interface properties as follows:

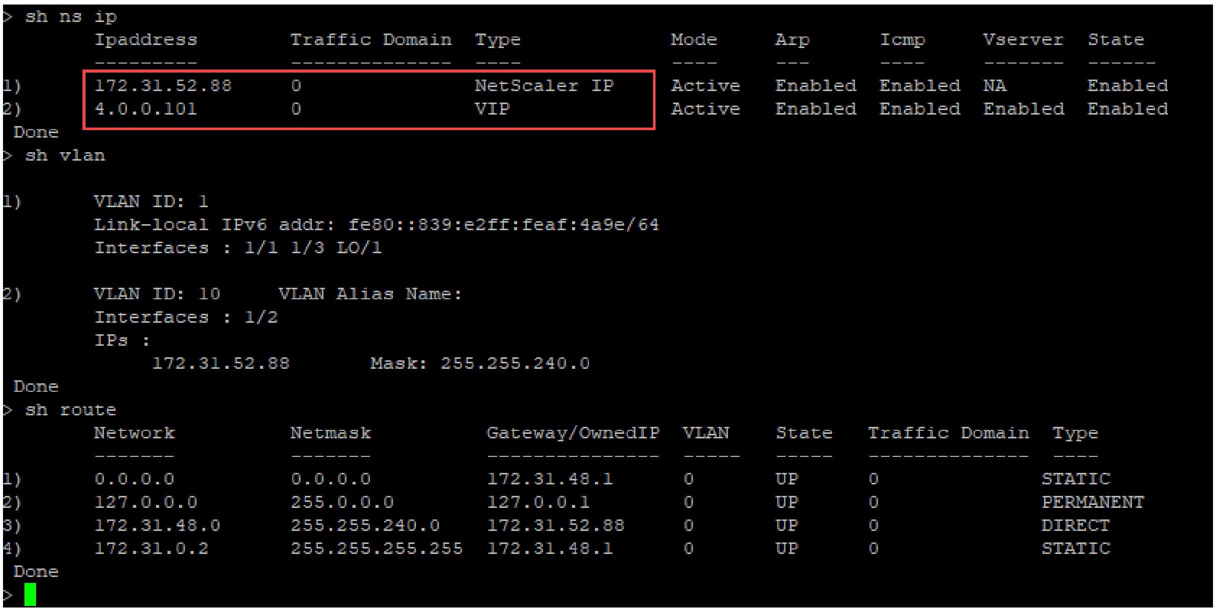
1. Navigate to the **AWS Portal > EC2 instances**, and select the instance that you have created by providing the custom bootstrap information.
2. In the **Description** tab, you can verify the properties of each network interface as shown in the following illustrations.







You can run the `show nsip` command in **ADC CLI**, and verify the network interfaces applied to the NetScaler VPX instance during the first boot of the ADC appliance.



```
> sh ns ip
        Ipaddress          Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------          --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88       0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101          0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

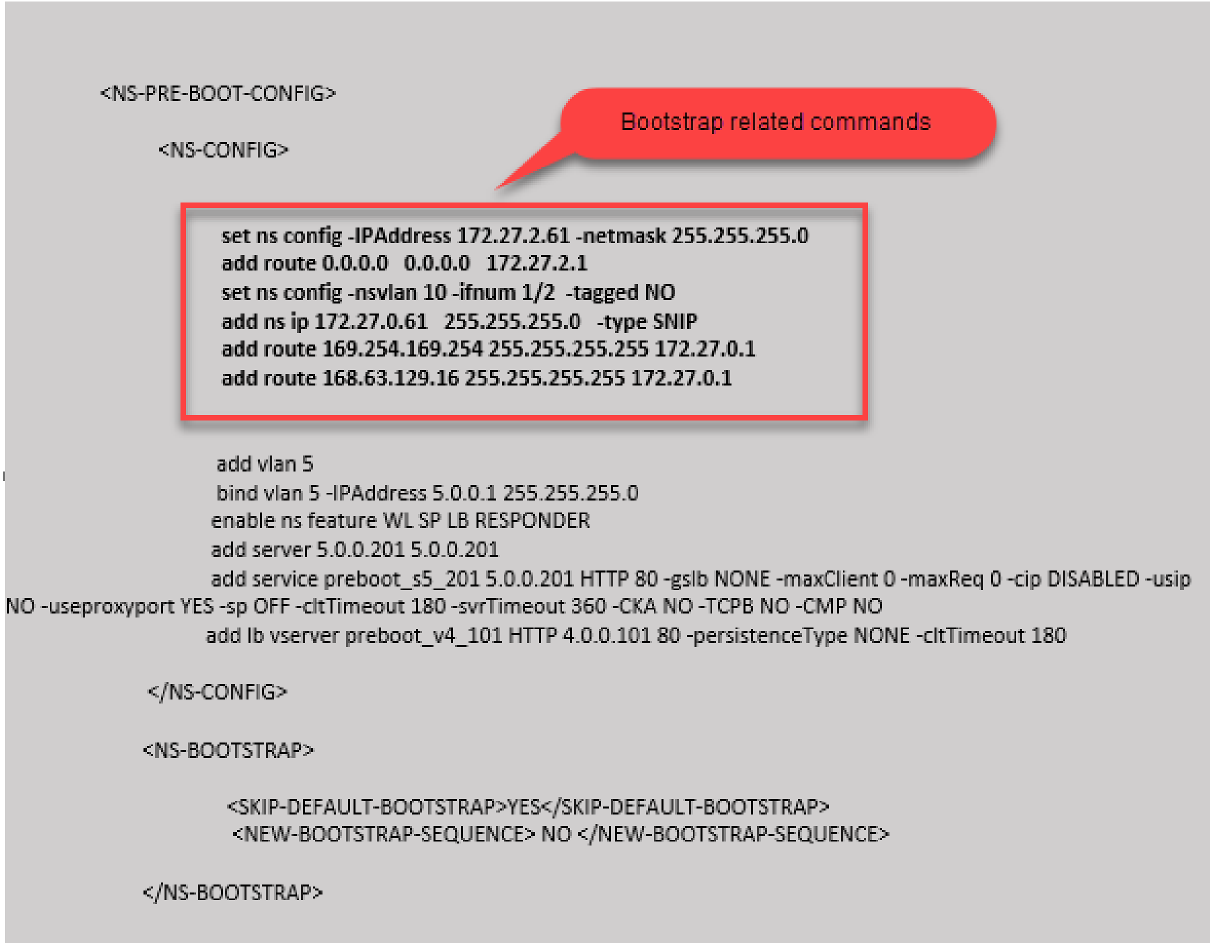
2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN   State   Traffic Domain  Type
        -------          -------          ---------------  -----  -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0      UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0      UP      0               PERMANENT
3)      172.31.48.0      255.255.240.0    172.31.52.88     0      UP      0               DIRECT
4)      172.31.0.2       255.255.255.255  172.31.48.1      0      UP      0               STATIC
 Done
>
```

Custom bootstrap example for Azure

In this example, bootstrap related commands are provided in the <`NS-CONFIG`> section. The <`NS-BOOTSTRAP`> section indicates that the default bootstrapping is skipped, and the custom bootstrap information provided in the <`NS-CONFIG`> section is run.

> **Note:**
>
> For Azure cloud, Instance Metadata Server (IMDS) and DNS servers are accessible only through primary interface (Eth0). Therefore, if Eth0 interface is not used as management interface (NSIP),

> Eth0 interface must at least be configured as SNIP for IMDS or DNS access to work. The route to IMDS endpoint (169.254.169.254) and DNS endpoint (168.63.129.16) through Eth0's gateway must also be added.




```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>

          set ns config -IPAddress 172.27.2.61 -netmask 255.255.255.0
          add route 0.0.0.0   0.0.0.0   172.27.2.1
          set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
          add ns ip 172.27.0.61   255.255.255.0   -type SNIP
          add route 169.254.169.254 255.255.255.255 172.27.0.1
          add route 168.63.129.16 255.255.255.255 172.27.0.1

          add vlan 5
          bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
          enable ns feature WL SP LB RESPONDER
          add server 5.0.0.201 5.0.0.201
          add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
```



```
                    -CMP NO
            add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
                persistenceType NONE -cltTimeout 180

        </NS-CONFIG>

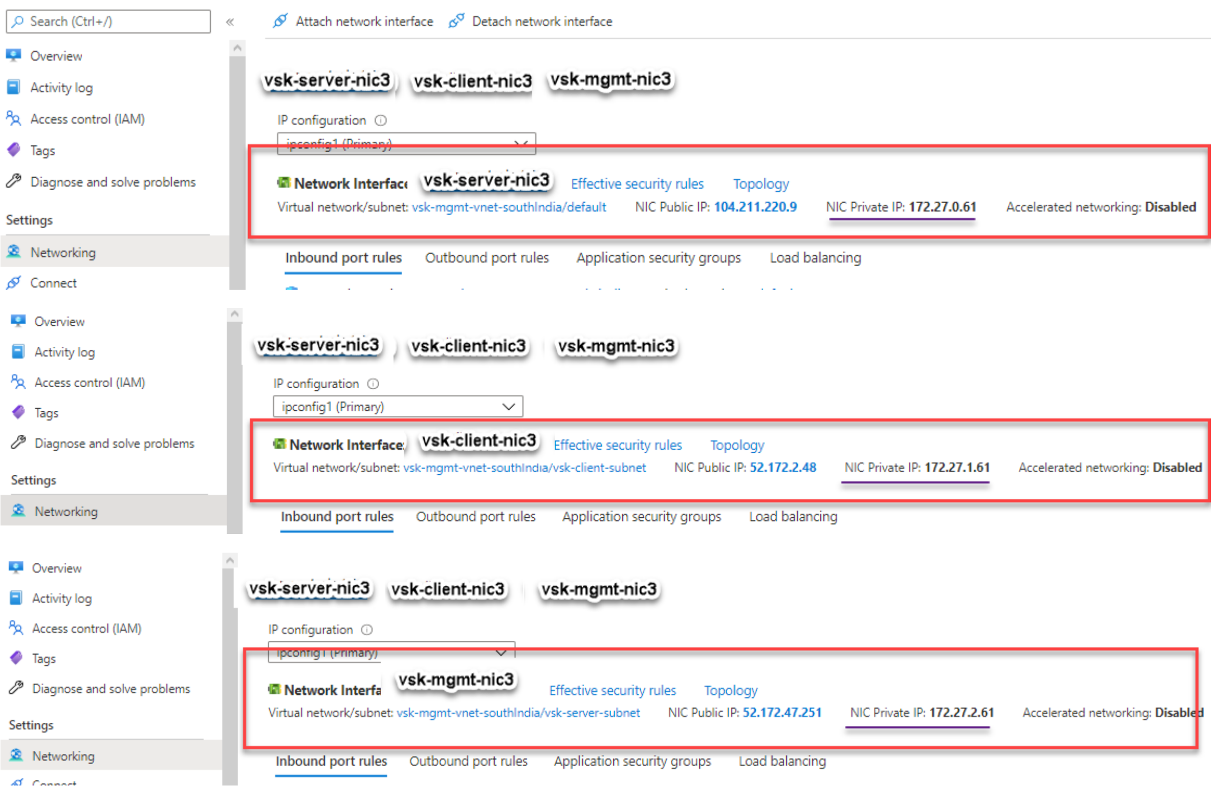
        <NS-BOOTSTRAP>

        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>

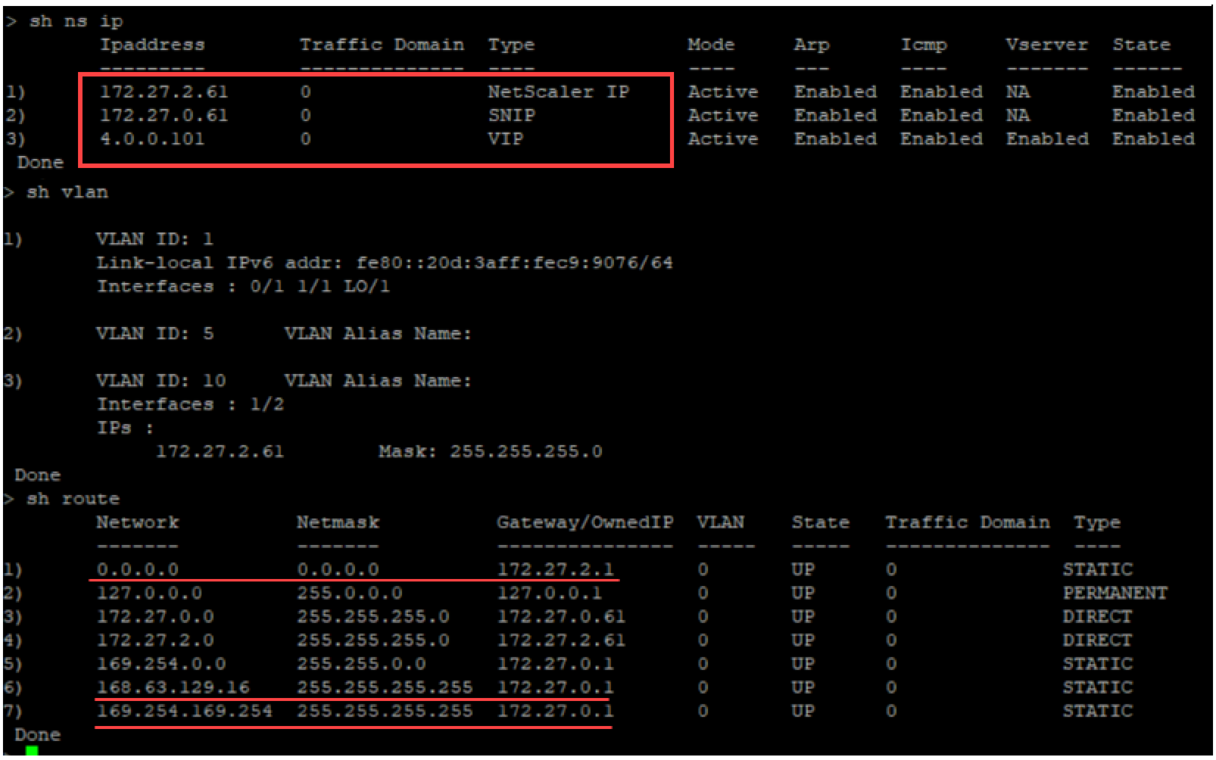
        </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

You can see that the NetScaler VPX instance is created with three network interfaces. Navigate to the **Azure portal > VM instance > Networking**, and verify the networking properties of the three NICs as shown in the following illustrations.



You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence specified in the `<NS-BOOTSTRAP>` section is applied. You can run the "show route"command to verify the subnet mask.



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.61      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.61      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

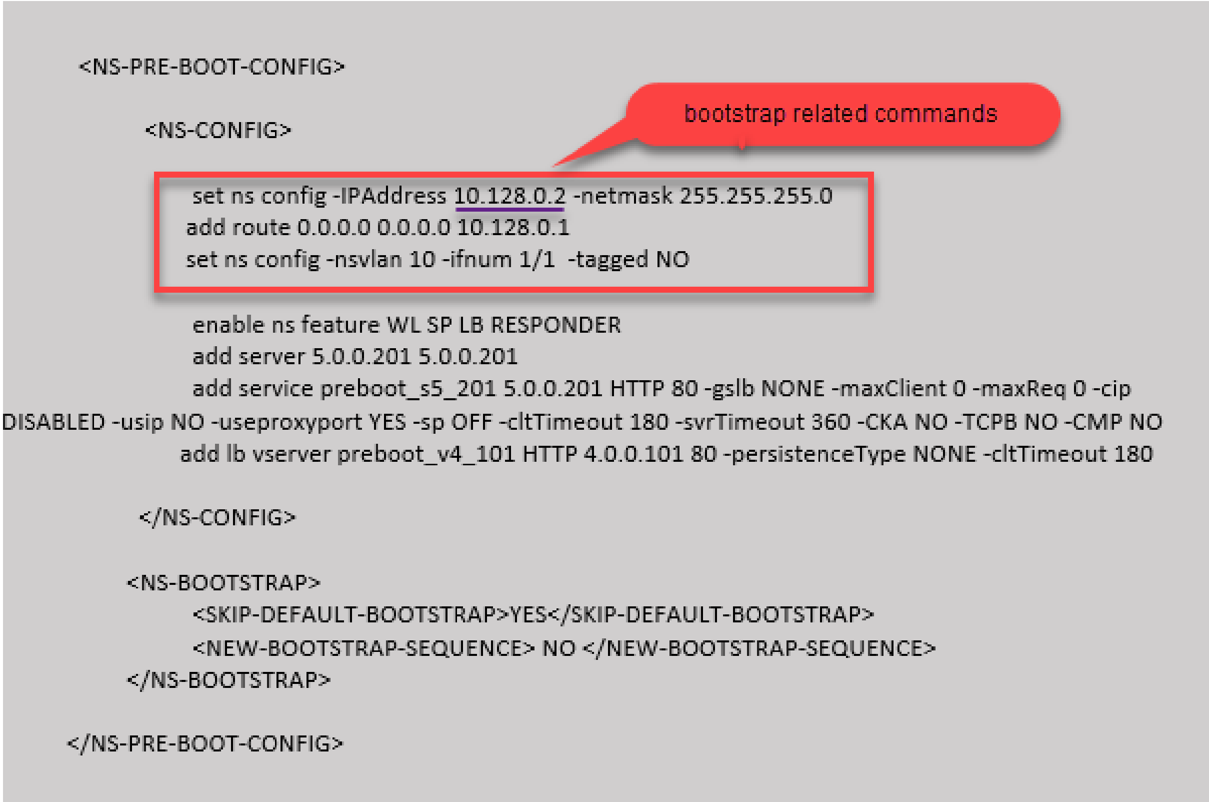
1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:9076/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 5       VLAN Alias Name:

3)      VLAN ID: 10      VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.61        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN   State  Traffic Domain  Type
        -------          -------          ---------------  -----  -----  --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0      UP     0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0      UP     0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.61      0      UP     0               DIRECT
4)      172.27.2.0       255.255.255.0    172.27.2.61      0      UP     0               DIRECT
5)      169.254.0.0      255.255.0.0      172.27.0.1       0      UP     0               STATIC
6)      168.63.129.16    255.255.255.255  172.27.0.1       0      UP     0               STATIC
7)      169.254.169.254  255.255.255.255  172.27.0.1       0      UP     0               STATIC
 Done
```

Custom bootstrap example for GCP

In this example, bootstrap related commands are provided in the <`NS-CONFIG`> section. The <`NS-BOOTSTRAP`> section indicates that the default bootstrapping is skipped, and the custom bootstrap information provided in the <`NS-CONFIG`> section is applied.



bootstrap related commands

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>

        set ns config -IPAddress 10.128.0.2 -netmask 255.255.255.0
        add route 0.0.0.0 0.0.0.0 10.128.0.1
        set ns config -nsvlan 10 -ifnum 1/1  -tagged NO

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip
DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>

        set ns config -IPAddress 10.128.0.2 -netmask 255.255.255.0
        add route 0.0.0.0 0.0.0.0 10.128.0.1
        set ns config -nsvlan 10 -ifnum 1/1  -tagged NO

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```


After the VM instance is created in the GCP portal with the custom bootstrap, you can verify the network interface properties as follows:

1. Select the instance that you have created by providing the custom bootstrap information.
2. Navigate to the Network interface properties and verify the NIC details as shown in the illustration.

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP
nic0	default	default	10.160.0.74	–	34.93.9.79 (ephemeral)
nic1	vsk-vpc-network-1	asia-south1-subnet-1	asia-south1-subnet1-10-128-0-2 (10.128.0.2)	–	34.93.245.110 (ephemeral)
nic2	vsk-vpc-network-2	asia-south1-subnet-5	10.128.4.30	–	34.93.146.248 (ephemeral)

You can run the `show nsip` command in **ADC CLI**, and verify that the configurations provided in the preceding `<NS-CONFIG>` section are applied at the first boot of the ADC appliance.

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.0.2       0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:4a/64
        Interfaces : 0/1 1/2 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/1
        IPs :
             10.128.0.2         Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  ----    -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.0.1       0       UP      0              STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0              PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.2       0       UP      0              DIRECT
 Done
```

Impact of attaching and detaching NICs in AWS and Azure

AWS and Azure provide the option to attach a network interface to an instance, and detach a network interface from an instance. Attaching or detaching interfaces might alter interface positions. Hence, Citrix recommends you to refrain from detaching interfaces from the NetScaler VPX instance. If you detach or attach an interface when custom bootstrapping is configured, NetScaler VPX instance reassigns the primary IP of the newly available interface in the management interface's position as NSIP. If no further interfaces are available after the one you detached, then the first interface is made the management interface for the NetScaler VPX instance.

For example, a NetScaler VPX instance is brought up with 3 interfaces: Eth0 (SNIP), Eth1 (NSIP), and Eth2 (VIP). If you detach Eth1 interface from the instance, which is a management interface, ADC configures the next available interface (Eth2) as the management interface. Thereby, the NetScaler VPX instance is still accessed through the primary IP of Eth2 interface. If Eth2 is also not available, then the remaining interface (Eth0) is made the management interface. Therefore, the access to NetScaler VPX instance continues to exist.

Let's consider a different assignment of interfaces as follows: Eth0 (SNIP), Eth1 (VIP), and Eth2 (NSIP). If you detach Eth2 (NSIP), because no new interface is available after Eth2, the first interface (Eth0) is made the management interface.

Improve SSL-TPS performance on public cloud platforms

You can get better SSL-TPS performance on AWS and GCP clouds by distributing the packet engine (PE) weights equally. Enabling this feature might result in a slight drop in HTTP throughput by around 10–12 %.

On AWS and GCP clouds, the NetScaler VPX instances with 10–16 vCPUs do not show any performance improvement because the PE weights are equally distributed by default.

> **Note:**
>
> In the Azure cloud, the PE weights are equally distributed by default. This feature does not improve any performance for the Azure instances.

Configure PE mode by using the NetScaler CLI

After setting the PE mode, you must reboot the system for the configuration changes to take effect.

At the command prompt, type:

```
set cpuparam pemode [CPUBOUND | Default]
```

When the PE mode is set to CPUBOUND, the PE weights are equally distributed.
When the PE mode is set to DEFAULT, the PE weights are set to default values.

> **Note:**
>
> This command is node specific. In a high availability or a cluster setup, you must run the command on each node. If you run the command on CLIP, the following error occurs: `Operation not permitted on CLIP`

To show the state of the PE mode that is configured, run the following command:

```
show cpuparam
```

Example:


```
> show cpuparam
    Pemode:   CPUBOUND
  Done
```

Apply PE mode configuration at the first boot of the NetScaler appliance in the cloud

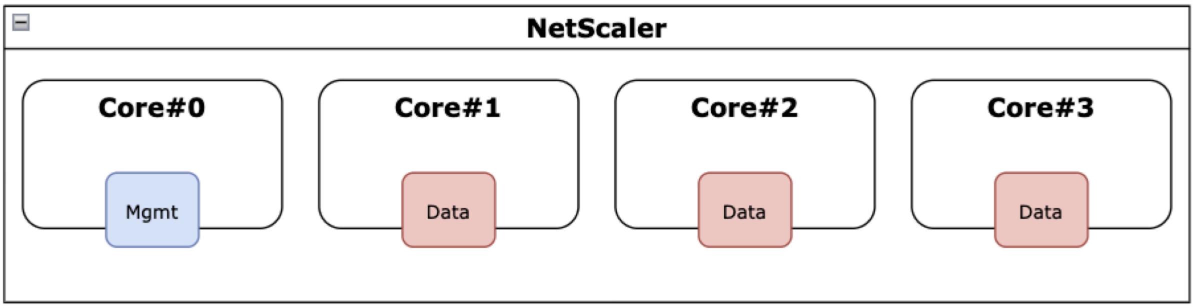
To apply the PE mode configuration at the first boot of the NetScaler appliance in the cloud, you must create a `/nsconfig/.cpubound.conf` file using the custom script. For more information, see Apply NetScaler VPX configurations at the first boot of the NetScaler appliance in cloud.

Configure simultaneous multithreading for NetScaler VPX on public clouds

NetScaler uses different dedicated cores for its management and its data plane functions. One core is typically assigned to management plane functions. The rest of the available cores are assigned to data plane functions.

The following image shows a simplified illustration of a 4 core NetScaler VPX.

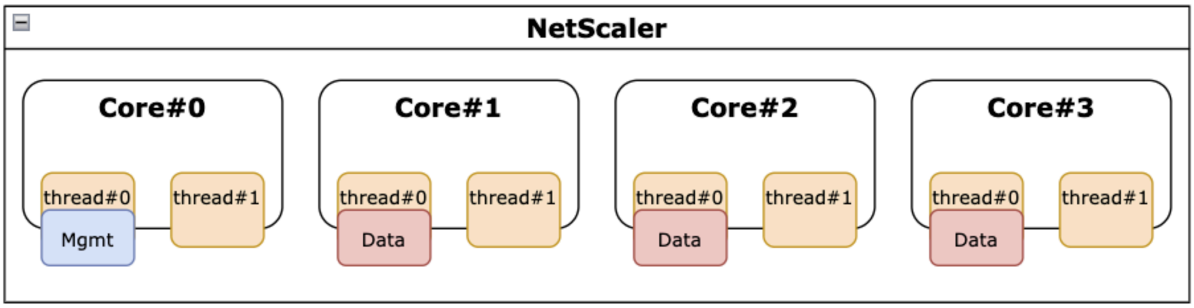
Figure 1. NetScaler management and data plane workload on a 4 core system



While the preceding image shows the distribution of NetScaler functions across available cores, it's not necessarily an accurate depiction of the underlying hardware. Most modern x86 CPUs provide two logical cores per physical core, through features commercially known as Intel Hyperthreading (HT) or AMD simultaneous multithreading (SMT).

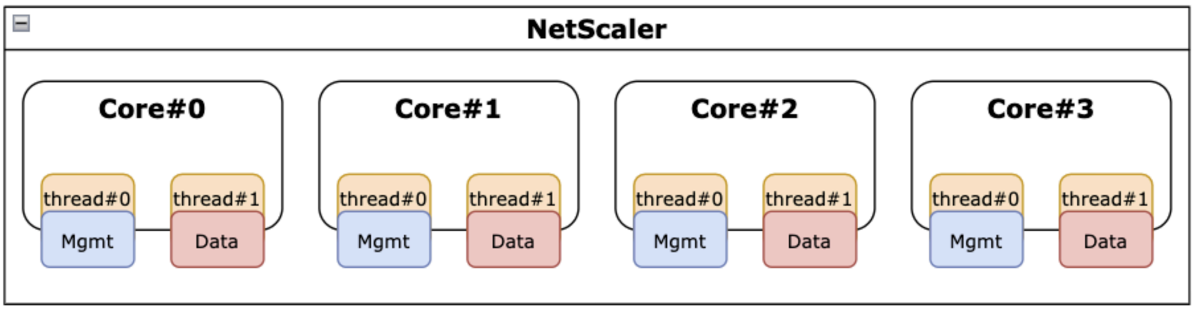
The following image shows NetScaler VPX running on a modern CPU with SMT disabled. Each CPU core is split into two or more logical CPUs, commonly referred to as threads. Each thread has its own set of replicated resources, a portion of partitioned resources, and competes for shared resources with its sibling threads.

Figure 2. NetScaler management and data plane workload on a 4 core/8 thread system with SMT disabled



The following image shows NetScaler VPX running on a modern CPU with SMT enabled.

Figure 3. NetScaler management and data plane workload on a 4 core system with SMT enabled



Enabling SMT improves NetScaler performance by:

- Running data plane functions on all physical cores.
- Moving the management plane functions to the sibling thread.
- Introducing a flexible resource limit mechanism to prevent management plane functions from compromising the performance of data plane functions.

SMT support matrix

The VPX platforms, cloud instance types, and NetScaler versions that support SMT are listed in the following table.

VPX platform	Instance types	NetScaler VPX version
AWS	M5, m5n, c5, c5n	13.1-48.x and later

> **Note:**
>
> By enabling the SMT feature, NetScaler VPX performance is boosted on the supported types.

Limitations

The SMT feature effectively doubles the vCPUs available to a NetScaler appliance. The licensing limits must be considered to allow NetScaler appliance to use them.

For example, consider NetScaler VPX illustrated in Figure 3. If a throughput-based licensing is used, a 10 Gbps or above license is required with the SMT feature to enable 8 vCPUs. Previously, a 1 Gbps license was sufficient for enabling 4 vCPUs. If a vCPU licensing is used, NetScaler VPX must be configured to check out licenses for double the count of vCPUs for proper operation. Contact NetScaler technical support for further guidance on this topic.

Configure SMT

Before enabling the SMT feature, ensure that your platform supports this feature. See the support matrix table in the previous section.

To enable the SMT feature, follow these steps:

1. Create an empty file named `.smt_handling` under the "/nsconfig" directory.
2. Save the current configuration.
3. Reboot NetScaler VPX instance.

```
nscli> shell touch /nsconfig/.smt_handling
  Done
nscli> reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:Y
Done
```

4. After rebooting, NetScaler indicates that the feature is both available and enabled.

```
smt_handling is set to “1”

> shell sysctl -a | grep smt_handling
netscaler.smt_handling_platform: 1
netscaler.smt_handling: 1
```

To disable the SMT feature, follow these steps:

1. Remove the `.smt_handling` file.
2. Reboot NetScaler VPX instance.

```
shell rm -f /nsconfig/.smt_handling
  Done

reboot

```

```
Are you sure you want to restart NetScaler (Y/N)? [N]:Y
Done
```

3. After rebooting, NetScaler indicates that the feature is available but disabled.

```
> shell sysctl -a | grep smt_handling
netscaler.smt_handling_platform: 1
netscaler.smt_handling: 0
```

Troubleshooting

Run the `sysctl` shell command to verify the status of the SMT feature.

```
```
> shell sysctl -a | grep smt_handling
>
```
```

The command can return any of the following outputs.

- The SMT feature is missing.

  The `sysctl` command returns no output.

- The SMT feature is not supported.

  The SMT feature isn't supported for any of the following reasons:

  - Your NetScaler VPX is older than 13.1-48.x or 14.1-12.x.
  - Your cloud does not support SMT.
  - Your VM instance type doesn't support SMT, for example, the vCPU count is more than 8.

  ```
  > shell sysctl -a | grep smt_handling
  netscaler.smt_handling_platform: 0  (indicates not supported)
  netscaler.smt_handling: 0  (indicates not enabled)
  ```

- The SMT feature is supported but not enabled.

```
    > shell sysctl -a | grep smt_handling
    netscaler.smt_handling_platform: 1 (available)
    netscaler.smt_handling: 0 (not enabled)
```

Install a NetScaler VPX instance on a bare metal server

A bare metal is a fully dedicated physical server that delivers physical isolation, fully integrated into the cloud environment. It is also known as a single-tenant server. A single tenancy allows you to avoid the noisy neighbor effect. With bare metal, you do not witness the noisy neighbor effect because you are the sole user.

A bare metal server installed with a hypervisor provides you a management suite to create virtual machines on the server. The hypervisor does not run applications natively. Its purpose is to virtualize your workloads into separate virtual machines to gain the flexibility and reliability of virtualization.

Prerequisites for installing NetScaler VPX instance on bare metal servers

A bare metal server must be obtained from a cloud vendor that meets all the system requirements for the respective hypervisor.

Install the NetScaler VPX instance on bare metal servers

To install NetScaler VPX instances on a bare metal server, you must first obtain a bare metal server with adequate system resources from a cloud vendor. On that bare metal server, any of the supported hypervisors such as Linux KVM, VMware ESX, Citrix Hypervisor, or Microsoft Hyper-V must be installed and configured before deploying the NetScaler VPX instance.

For more information on the list of different hypervisors and features supported on a NetScaler VPX instance, see Support matrix and usage guidelines.

For more information on installing NetScaler VPX instances on different hypervisors, see the respective documentation.

- **Citrix Hypervisor:** See Install a NetScaler VPX instance on Citrix Hypervisor.
- **VMware ESX:** See Install a NetScaler VPX instance on VMware ESX.
- **Microsoft Hyper-V:** See Install a NetScaler VPX instance on Microsoft Hyper-V server.
- **Linux KVM platform:** See Install a NetScaler VPX instance on Linux-KVM platform.

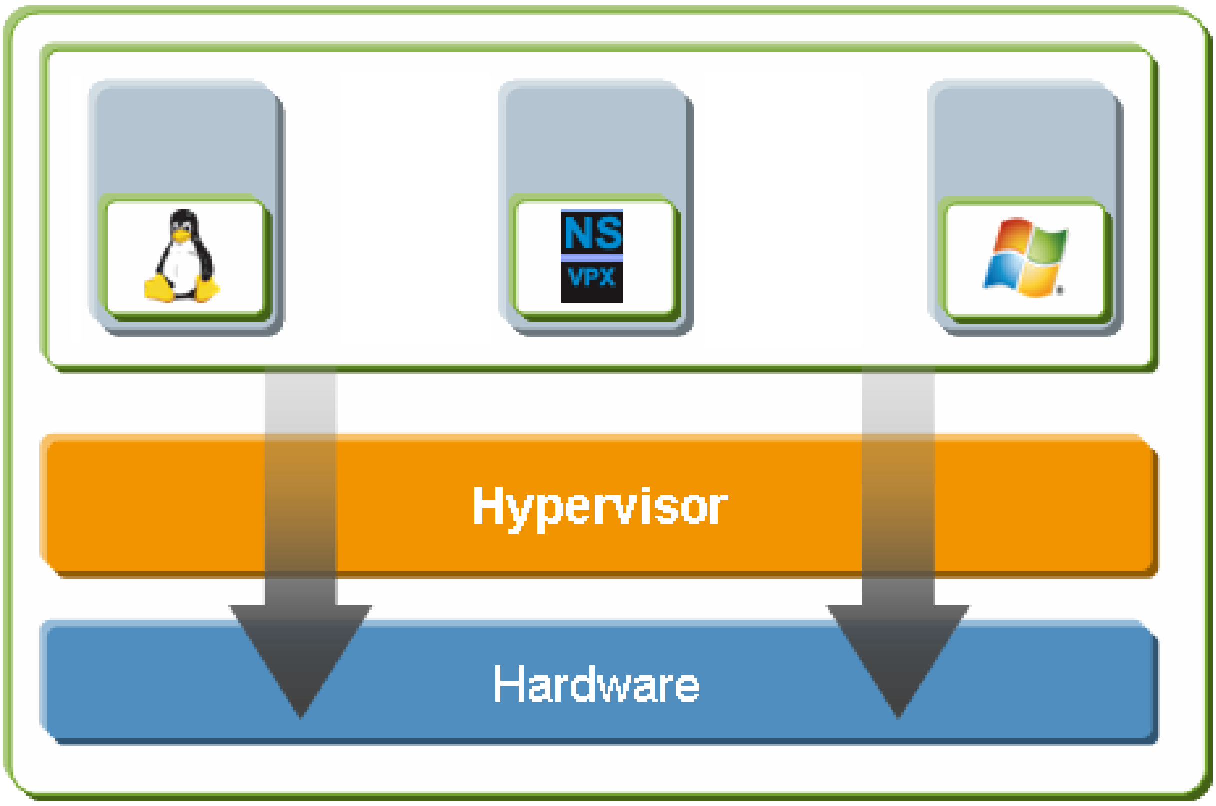
Install a NetScaler VPX instance on Citrix Hypervisor/XenServer

To install VPX instances on the Citrix Hypervisor/XenServer, you must first install the Hypervisor on a machine with adequate system resources. To perform the NetScaler VPX instance installation, you use Citrix XenCenter, which must be installed on a remote machine that can connect to the Hypervisor host through the network.

For more information about Hypervisor, see Citrix Hypervisor documentation.

The following figure shows the bare-metal solution architecture of NetScaler VPX instance on Hypervisor.

Figure. A NetScaler VPX instance on Citrix Hypervisor/XenServer



Prerequisites for installing a NetScaler VPX instance on Hypervisor

Before you begin installing a virtual appliance, do the following:

- Install Hypervisor version 6.0 or later on hardware that meets the minimum requirements.
- Install XenCenter on a management workstation that meets the minimum system requirements.
- Obtain virtual appliance license files. For more information about virtual appliance licenses, see the NetScaler Licensing Guide.

Hypervisor hardware requirements

The following table describes the minimum hardware requirements for a Hypervisor platform running a NetScaler VPX instance.

Table 1. Minimum system requirements for Hypervisor running a nCore VPX instance

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT) enabled. To run the NetScaler VPX instance, hardware support for virtualization must be enabled on the Hypervisor host. Make sure that the BIOS option for virtualization support is not disabled. For more details, see BIOS documentation.
RAM	3 GB
Disk space	Locally attached storage (PATA, SATA, SCSI) with 40 GB of disk space. **Note:** Hypervisor installation creates a 4 GB partition for the Hypervisor host control domain. The remaining space is available for NetScaler VPX instance and other virtual machines.
NIC	One 1 Gbps NIC; recommended: two 1 Gbps NICs

For more information, see the XenServer documentation.

The following table lists the virtual computing resources that Hypervisor must provide for each nCore VPX virtual appliance.

Table 2. Minimum virtual computing resources required for running a nCore VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	2

> **Note:**
>
> For production use of NetScaler VPX instance, Citrix recommends that CPU priority (in virtual machine properties) must be set to the highest level, to improve scheduling behavior and network latency.

XenCenter system requirements

XenCenter is a Windows client application. It cannot run on the same machine as the Hypervisor host. For more information about minimum system requirements and installing XenCenter, see the following Hypervisor documents:

- System requirements
- Install

Install NetScaler VPX instances on Hypervisor by using XenCenter

After you have installed and configured Hypervisor and XenCenter, you can use XenCenter to install virtual appliances on Hypervisor. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running Hypervisor.

To install NetScaler VPX instances on Hypervisor by using XenCenter, follow these steps:

1. Start **XenCenter** on your workstation.
2. On the **Server** menu, click **Add**.
3. In the **Add New Server** dialog box, in the host name text box, type the IP address or DNS name of the Hypervisor that you want to connect to.
4. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click **Connect**. The Hypervisor name appears in the navigation pane with a green circle, which indicates that the Hypervisor is connected.
5. In the navigation pane, click the name of the Hypervisor on which you want to install the NetScaler VPX instance.
6. On the **VM** menu, click **Import**.
7. In the **Import** dialog box, in the Import file name, browse to the location at which you saved the NetScaler VPX instance `.xva` image file. Make sure that the Exported VM option is selected, and then click **Next**.
8. Select the Hypervisor on which you want to install the virtual appliance, and then click **Next**.
9. Select the local storage repository in which to store the virtual appliance, and then click Import to begin the import process.
10. You can add, modify, or delete the virtual network interfaces as required. When finished, click **Next**.
11. Click **Finish** to complete the import process.

    > **Note:**
    >
    > To view the status of the import process, click the **Log** tab.

12. If you want to install another virtual appliance, repeat steps 5 through 11.

> **Note:**
>
> After the initial configuration of the VPX instance, if you want to upgrade the appliance to the latest software release, see Upgrading or Downgrading the System Software.

Configure VPX instances to use single root I/O virtualization (SR-IOV) network interfaces

After you have installed and configured a NetScaler VPX instance on Citrix Hypervisor, you can configure the virtual appliance to use SR-IOV network interfaces.

The following NICs are supported:

- Intel 82599 10G
- Intel X710 10G
- Intel XL710 40G

Limitations

Citrix Hypervisor does not support some features on SR-IOV interfaces. The limitations with Intel 82599, Intel X710, and Intel XL710 NICs are listed in the following sections.

Limitations for Intel 82599 NIC

Intel 82599 NIC does not support the following features:

- L2 mode switching
- Clustering
- Admin partitioning [Shared VLAN mode]
- High Availability [Active - Active mode]
- Jumbo frames
- IPv6 protocol in Cluster environment

Limitations for Intel X710 10G and Intel XL710 40G NICs

Intel X710 10G and Intel XL710 40G NICs have the following limitations:

- L2 mode switching is not supported.
- Admin partitioning (shared VLAN mode) is not supported.

- In a cluster, Jumbo frames are not supported when the XL710 NIC is used as a data interface.
- Interface list reorders when interfaces are disconnected and reconnected.
- Interface parameter configurations such as speed, duplex, and auto negotiations are not supported.
- For both Intel X710 10G and Intel XL710 40G NICs, the interface comes up as 40/x interface.
- Up to only 16 Intel X710/XL710 SR-IOV interfaces can be supported on a VPX instance.

> **Note:**
>
> For Intel X710 10G and Intel XL710 40G NICs to support IPv6, enable trust mode on the Virtual Functions (VFs) by typing the following command on the Citrix Hypervisor host:
>
> `# ip link set <PNIC> <VF> trust on`
>
> **Example:**
>
> `# ip link set ens785f1 vf 0 trust on`

Prerequisites for Intel 82599 NIC

On the Citrix Hypervisor host, ensure that you:

- Add the Intel 82599 NIC (NIC) to the host.
- Block list the `ixgbevf` driver by adding the following entry to the **/etc/modprobe.d/blacklist.conf** file:

  blacklist ixgbevf

- Enable SR-IOV Virtual Functions (VFs) by adding the following entry to the **/etc/modprobe.d/ixgbe** file:

  options ixgbe max_vfs=<*number_of_VFs*>

  where <*number_VFs*> is the number of SR-IOV VFs that you want to create.

- Verify that SR-IOV is enabled in BIOS.

> **Note:**
>
> IXGBE driver version 3.22.3 is recommended.

Assign Intel 82599 SR-IOV VFs to the NetScaler VPX instance by using the Citrix Hypervisor host

To assign an Intel 82599 SR-IOV VFs to NetScaler VPX instance, follow these steps:

1. On the Citrix Hypervisor host, use the following command to assign the SR-IOV VFs to the NetScaler VPX instance:

   xe host-call-plugin plugin=iovirt host-uuid=*<Xen host UUID>* **fn=assign_free_vf args:uuid**=*<NetScaler VM UUID>* **args:ethdev**=*<interface name>* **args:mac=***<Mac addr>*

   Where:

   - *<Xen host UUID>* is the UUID of the Citrix Hypervisor host.
   - *<NetScaler VM UUID>* is the UUID of the NetScaler VPX instance.
   - *<interface name>* is the interface for the SR-IOV VFs.
   - *<MAC address* > is the MAC address of the SR-IOV VF.

   > **Note:**
   >
   > Specify the MAC address that you want use in the args:Mac= parameter, if not specified, the `iovirt` script randomly generates and assigns a MAC address. Also, if you want to use the SR-IOV VFs in Link Aggregation mode, make sure that you specify the MAC address as 00:00:00:00:00:00.

2. Boot the NetScaler VPX instance.

Unassign Intel 82599 SR-IOV VFs to the NetScaler VPX instance by using the Citrix Hypervisor host

If you have assigned an incorrect SR-IOV VFs or if you want to modify an assigned SR-IOV VFs, you need to unassign and reassign the SR-IOV VFs to the NetScaler VPX instance.

To unassign SR-IOV network interface assigned to a NetScaler VPX instance, follow these steps:

1. On the Citrix Hypervisor host, use the following command to assign the SR-IOV VFs to the NetScaler VPX instance and reboot the NetScaler VPX instance:

   xe host-call-plugin plugin=iovirt **host-uuid**=*<Xen_host_UUID>* **fn**=unassign_all **args:uuid**=*<Netscaler_VM_U*

   Where:

   - *<Xen_host_UUID>* - The UUID of the Citrix Hypervisor host.
   - *<Netscaler_VM_UUID>* - The UUID of the NetScaler VPX instance

2. Boot the NetScaler VPX instance.

Assign Intel X710/XL710 SR-IOV VFs to the NetScaler VPX instance by using the Citrix Hypervisor host

To assign an Intel X710/XL710 SR-IOV VF to the NetScaler VPX instance, follow these steps:

1. Run the following command on the Citrix Hypervisor host to create a network.

   ```
   xe network-create name-label=<network-name>
   ```

   Example:

   ```
   xe network-create name-label=SR-IOV-NIC-18 8ee59b73-7319-6998-cd69
      -b9fa3e8d7503
   ```

2. Determine the PIF Universal Unique Identifier (UUID) of the NIC on which the SR-IOV network is to be configured.

   ```
   xe pif-list

               uuid ( RO)   : e2874343-f1de-1fa7-8fef-98547c348783
             device ( RO): eth18
   currently-attached ( RO): true
               VLAN ( RO): -1
         network-uuid ( RO): f865bd85-44dd-b865-ab65-dcd6ae28c16e
   ```

3. Configure the network as an SR-IOV network. The following command also returns the UUID of the newly created SR-IOV network:

   ```
   xe network-sriov-create network-uuid=<network-uuid> pif-uuid=<
      physical-pif-uuid>
   ```

   Example:

   ```
   xe network-sriov-create network-uuid=8ee59b73-7319-6998-cd69-
      b9fa3e8d7503 pif-uuid=e2874343-f1de-1fa7-8fef-98547
      c3487831629b44f-832a-084e-d67d-5d6d314d5e0f
   ```

   To get more information on the SR-IOV network parameters, run the following command:

   ```
   [root@citrix-XS82-TOPO ~]# xe network-sriov-param-list uuid=1629
      b44f-832a-084e-d67d-5d6d314d5e0f

               uuid ( RO): 1629b44f-832a-084e-d67d-5d6d314d5e0f
         physical-PIF ( RO): e2874343-f1de-1fa7-8fef-98547c348783
          logical-PIF ( RO): 85d52771-5814-c62d-45fa-f37b536144ff
      requires-reboot ( RO): false
   remaining-capacity ( RO): 32
   ```

4. Create a virtual interface (VIF) and attach it to the target VM.

```
xe vif-create device=0 mac=b2:61:fc:ae:00:1d network-uuid=8ee59b73
    -7319-6998-cd69-b9fa3e8d7503 vm-uuid=b507e8a6-f5ca-18eb-561d
    -308218a9dd68
3e1e2e58-b2ad-6dc0-61d4-1d149c9c6466
```

> **Note:**
>
> The NIC index number of the VM must start with 0.

Use the following command to find the VM UUID:

```
[root@citrix-XS82-TOPO ~]# xe vm-list
uuid ( RO): b507e8a6-f5ca-18eb-561d-308218a9dd68
 name-label ( RW): sai-vpx-1
power-state ( RO): halted
```

Remove Intel X710/XL710 SR-IOV VFs from the NetScaler instance by using the Citrix Hypervisor host

To remove an Intel X710/XL710 SR-IOV VF from a NetScaler VPX instance, follow these steps:

1. Copy the UUID for the VIF that you want to destroy.
2. Run the following command on the Citrix Hypervisor host to destroy the VIF.

```
xe vif-destroy uuid=<vif-uuid>
```

Example:

```
[root@citrix-XS82-TOPO ~]# xe vif-destroy uuid=3e1e2e58-b2ad-6dc0
    -61d4-1d149c9c6466
```

Configure link aggregation on the SR-IOV interface

To use the SR-IOV virtual functions (VFs) in link aggregation mode, you need to disable spoof checking for virtual functions that you have created.

On the Citrix Hypervisor host, use the following command to disable spoof checking:

ip link set *<interface_name>* **vf** *<VF_id>* **spoofchk** off

Where:

- <interface_name> is the interface name.
- <VF_id> is the virtual function ID.

After disabling spoof checking for all the virtual function that you have created, restart the NetScaler VPX instance, and configure link aggregation. For instructions, see Configure link aggregation.

> **Important:**
>
> While you are assigning the SR-IOV VFs to the NetScaler VPX instance, make sure that you specify MAC address 00:00:00:00:00:00 for the VFs.

Configure VLAN on the SR-IOV interface

You can configure VLAN on the SR-IOV virtual functions. For instructions, see Configuring a VLAN.

> **Important:**
>
> Make sure that the Citrix Hypervisor host does not contain VLAN settings for the VF interface.

Other references

SR-IOV enabled NICs

Add an SR-IOV Network

Install a NetScaler VPX instance on VMware ESX

Before installing NetScaler VPX instances on VMware ESX, make sure the VMware ESX Server is installed on a machine with adequate system resources. To install a NetScaler VPX instance on VMware ESXi, you use the VMware vSphere client. The client or tool must be installed on a remote machine that can connect to VMware ESX through the network.

This section includes the following topics:

- Prerequisites
- Installing a NetScaler VPX instance on VMware ESX

> **Important:**
>
> You can't install standard VMware Tools or upgrade the VMware Tools version available on a NetScaler VPX instance. VMware Tools for a NetScaler VPX instance are delivered as part of the NetScaler software release.

Prerequisites

Before you begin installing a virtual appliance, do the following:

- Install VMware ESX on hardware that meets the minimum requirements.

- Install VMware Client on a management workstation that meets the minimum system requirements.
- Download the NetScaler VPX appliance setup files.
- Create a virtual switch and attach the physical NIC to the virtual switch.
- Add port group and attach to the virtual switch.
- Attach the port group to the VM.
- Obtain VPX license files. For more information about NetScaler VPX instance licenses, see Licensing overview.

VMware ESX hardware requirements

The following table describes the minimum system requirements for VMware ESX servers running NetScaler VPX nCore virtual appliance.

Table 1. Minimum system requirements for a VMware ESX server running a NetScaler VPX instance

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT) enabled. To run a NetScaler VPX instance, hardware support for virtualization must be enabled on the VMware ESX host. Make sure that the BIOS option for virtualization support isn't disabled. For more information, see your BIOS documentation. From the NetScaler 13.1 release onwards, the NetScaler VPX instance on VMware ESXi hypervisor supports AMD processors.
RAM	2 GB VPX. For critical deployments, we do not recommend 2 GB RAM for VPX because the system operates in a memory-constrained environment. This might lead to scale, performance, or stability related issues. Recommended is 4 GB RAM or 8 GB RAM.
Disk space	20 GB more than the minimum server requirements from VMware for setting up ESXi. See VMware documentation for minimum server requirements.
Network	One 1-Gbps NIC (NIC); Two 1-Gbps NICs recommended

For information about installing VMware ESX, see http://www.vmware.com/.

For the SR-IOV network interface or PCI passthrough support, ensure that the following processors and settings are enabled:

- Intel processors supporting Intel-VT
- AMD processors supporting AMD-V
- I/O Memory Management Unit (IOMMU) or SR-IOV is enabled in BIOS

The following NICs are supported in SR-IOV mode:

- Mellanox ConnectX-4 NIC, starting from NetScaler release 13.1-42.x onwards
- Intel 82599 NIC

The following table lists the virtual computing resources that the VMware ESX server must provide for each VPX nCore virtual appliance.

Table 2. Minimum virtual computing resources required for running a NetScaler VPX instance

Component	Requirement
Memory	4 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	In ESX, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded to version 7 or higher.
Disk space	20 GB

> **Note:**
>
> This is in addition to any disk requirements for the hypervisor.

For production use of VPX virtual appliance, the full memory allocation must be reserved. CPU cycles (in MHz) equal to at least the speed of one CPU core of the ESX must be reserved.

VMware vSphere client system requirements

VMware vSphere is a client application that can run on Windows and Linux operating systems. It can't run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 3. Minimum system requirements for VMware vSphere client installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "vSphere Compatibility Matrixes"PDF file at http://kb.vmware.com/.
CPU	750 MHz; 1 gigahertz (GHz) or faster recommended
RAM	1 GB. 2 GB recommended
NIC (NIC)	100 Mbps or faster NIC

OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. It can't run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 4. Minimum system requirements for OVF tool installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended
RAM	1 GB Minimum, 2 GB recommended
NIC (NIC)	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/.

Downloading the NetScaler VPX setup files

The NetScaler VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log on. If you do not have a Citrix account, access the home page at http://www.citrix.com, click the **New Users link**, and follow the instructions to create a Citrix account.

Once logged on, navigate the following path from the Citrix home page:

Citrix.com > **Downloads > NetScaler > Virtual Appliances**.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-13.0-71.44_nc_64-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-13.0-71.44_nc_64.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-13.0-71.44_nc_64.mf)

Install a NetScaler VPX instance on VMware ESX

After you have installed and configured VMware ESX, you can use the VMware vSphere client to install virtual appliances on the VMware ESX server. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running VMware ESX.

To install NetScaler VPX instances on VMware ESX by using VMware vSphere Client, follow these steps:

1. Start the VMware vSphere client on your workstation.
2. In the **IP address / Name** text box, type the IP address of the VMware ESX server that you want to connect to.
3. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click Login.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file**, browse to the location at which you saved the NetScaler VPX instance setup files, select the .ovf file, and click **Next**.
6. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the ESX host. Click **Next** to start installing a virtual appliance on VMware ESX. When installation is complete, a pop-up window informs you of the successful installation.
7. You are now ready to start the NetScaler VPX instance. In the navigation pane, select the NetScaler VPX instance that you have installed, and from the right-click menu, select **Power On**.
8. After the VM is booted, from the console, configure the NetScaler IP, Netmask, and Gateway addresses. When you complete the configuration, select the **Save and Quit** option in the console.
9. To install another virtual appliance, repeat from Step 6 through Step 8.

> **Note:**
>
> By default, the NetScaler VPX instance uses E1000 network interfaces.
>
> After the installation, you can use the vSphere client or vSphere Web Client to manage virtual appliances on VMware ESX.

> To enable VLAN tagging on VMware ESX, configure the port group's VLAN ID to All (4095) on the vSwitch. For detailed instructions on setting a VLAN ID on the vSwitch, refer to the VMware documentation.

Migrate a NetScaler VPX instance by using VMware vMotion

You can migrate a NetScaler VPX instance by using VMware vSphere vMotion.

Follow these usage guidelines:

- VMware does not support the vMotion feature on virtual machines configured with PCI Passthrough and SR-IOV interfaces.
- Supported interfaces are E1000 and VMXNET3. To use vMotion on your VPX instance, ensure that the instance is configured with a supported interface.
- For more information about how to migrate an instance by using VMware vMotion, see the VMware documentation.

Configure a NetScaler VPX instance to use VMXNET3 network interface

After you have installed and configured the NetScaler VPX instance on the VMware ESX, you can use the VMware vSphere web client to configure the virtual appliance to use VMXNET3 network interfaces.

To configure NetScaler VPX instances to use VMXNET3 network interfaces by using the VMware vSphere Web Client:

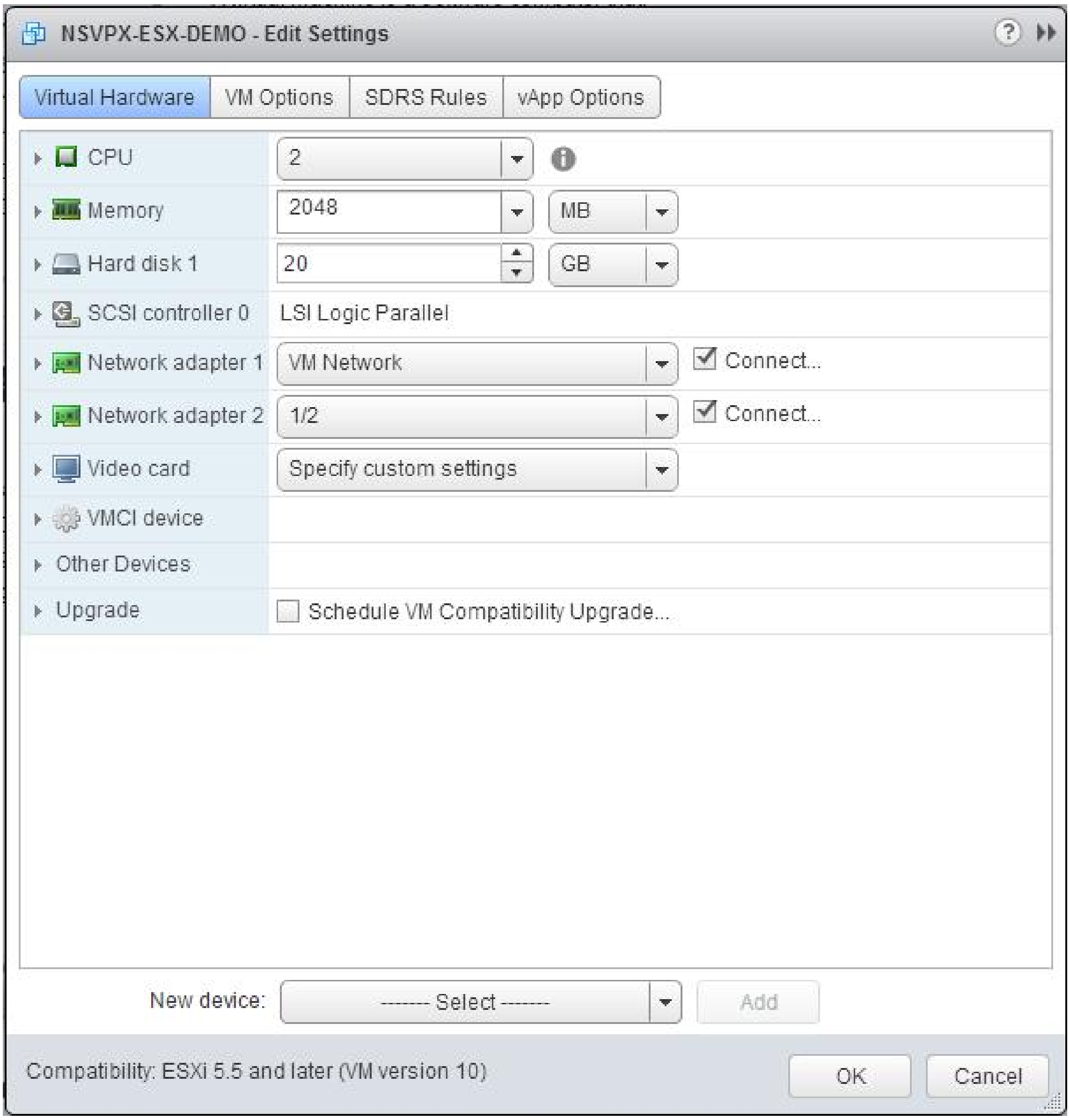
1. In the vSphere Web Client, select Hosts and Clusters.
2. Upgrade the Compatibility setting of the NetScaler VPX instance to ESX, as follows:

   a. Power off the NetScaler VPX instance.

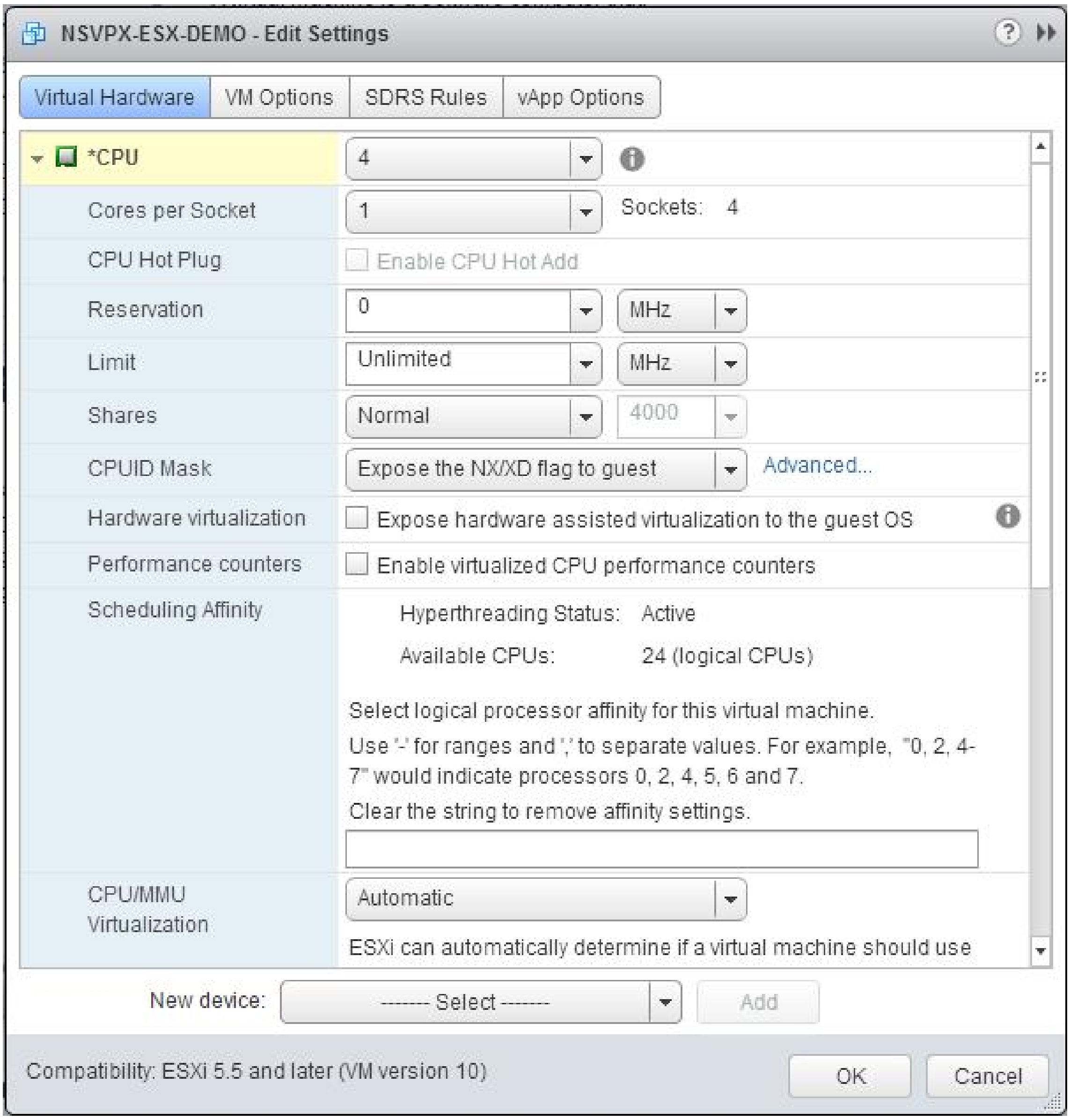
   b. Right-click the NetScaler VPX instance and select Compatibility > Upgrade VM Compatibility.

   c. In the Configure VM Compatibility dialog box, select ESXi 5.5 and later from the Compatible with drop-down list and click OK.

3. Right-click on the NetScaler VPX instance and click Edit Settings.



4. In the <virtual_appliance> - Edit Settings dialog box, click the CPU section.



5. In the CPU section, update the following:

    - Number of CPUs
    - Number of Sockets
    - Reservations
    - Limit
    - Shares

    Set the values as follows:

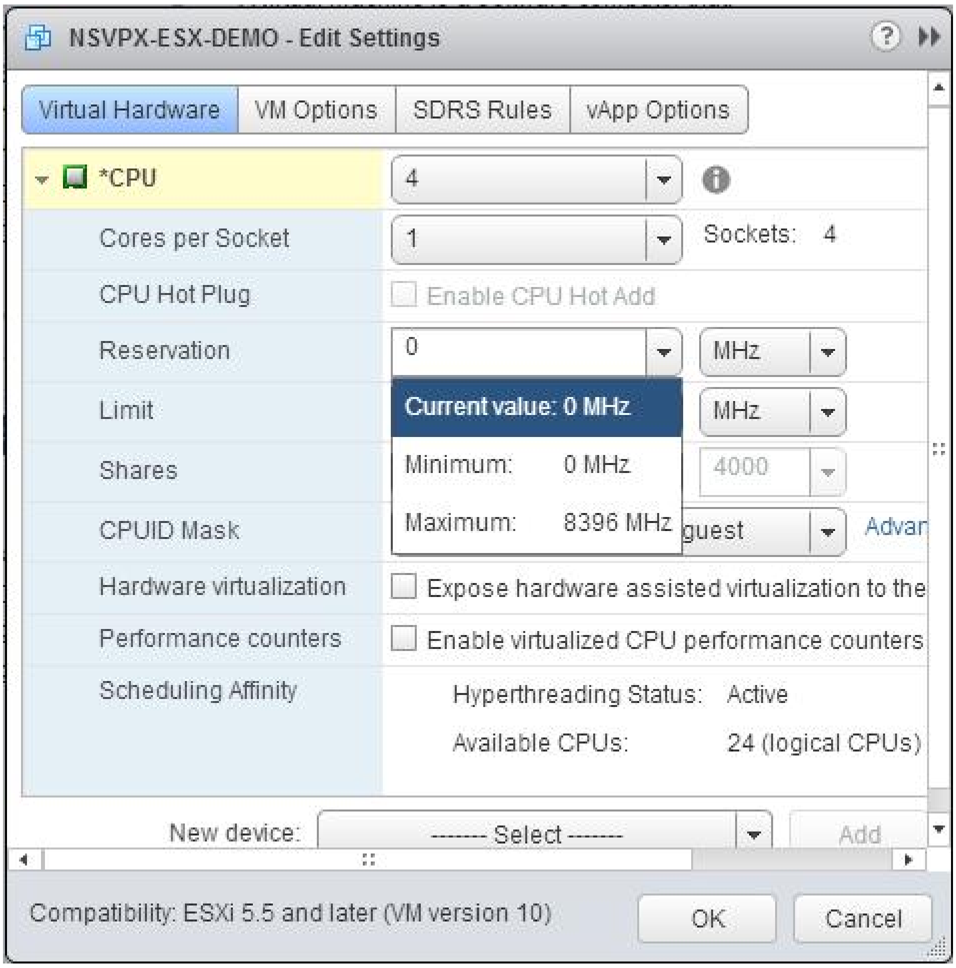
    a. In the CPU drop-down list, select the number of CPUs to assign to the virtual appliance.

    b. In the Cores per Socket drop-down list, select the number of sockets.

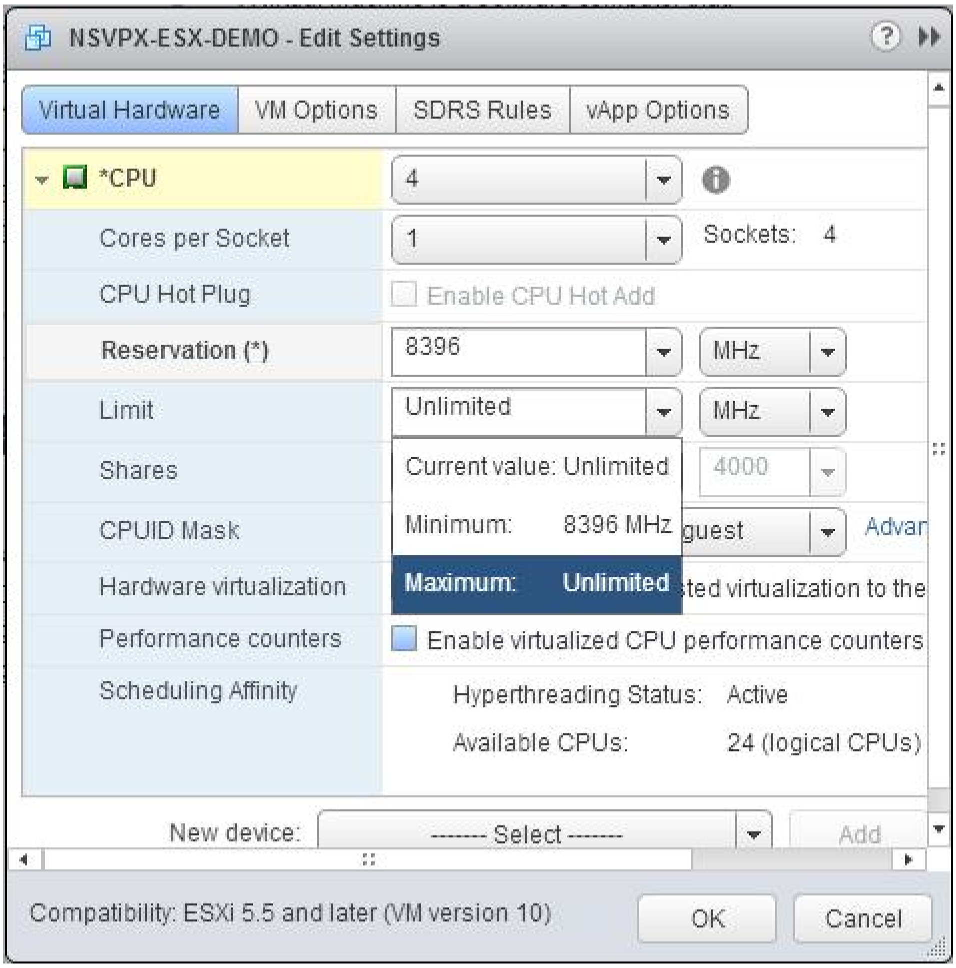
    c. (Optional) In the CPU Hot Plug field, select or unselect the Enable CPU Hot Add check box.

> **Note:**
>
> Citrix recommends accepting the default (disabled).

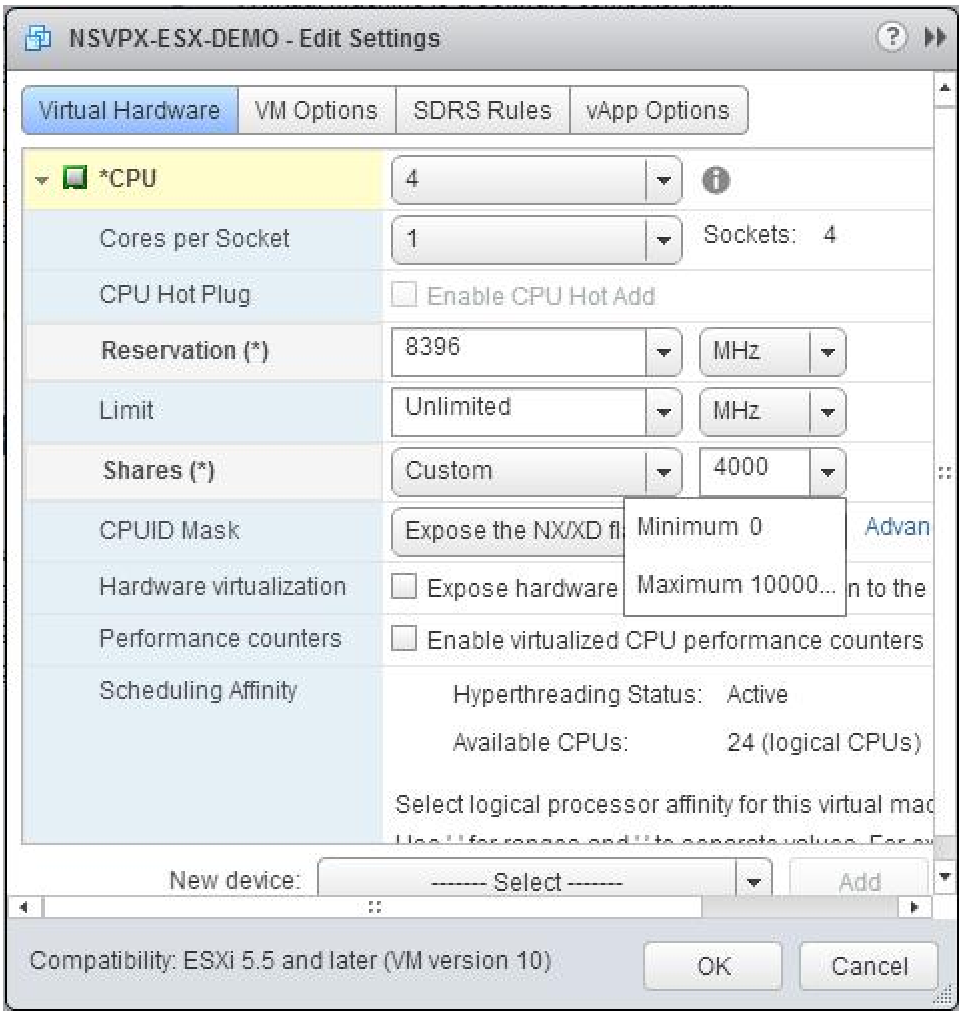
d. In the Reservation drop-down list, select the number that is shown as the maximum value.



e. In the Limit drop-down list, select the number that is shown as the maximum value.



f. In the Shares drop-down lists, select Custom and the number that is shown as the maximum value.



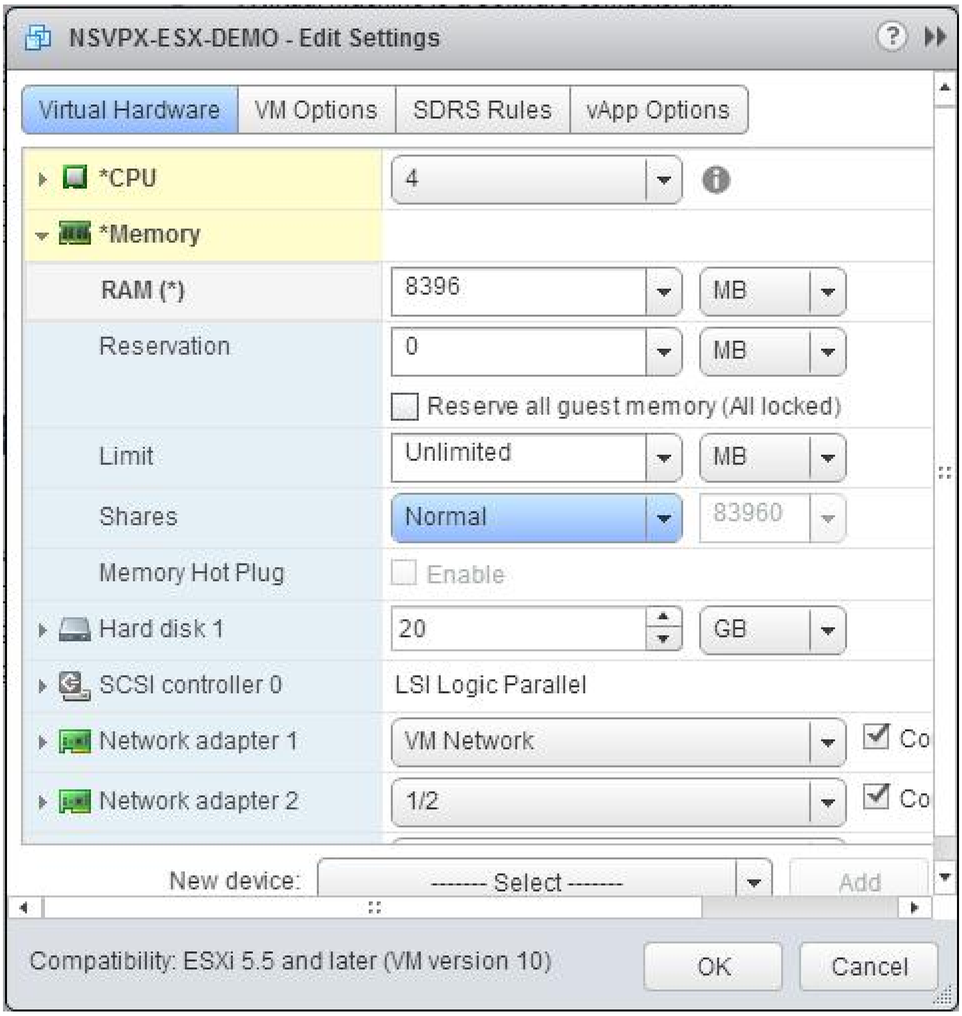
6. In the Memory section, update the following:

    - Size of RAM
    - Reservations
    - Limit
    - Shares

    Set the values as follows:

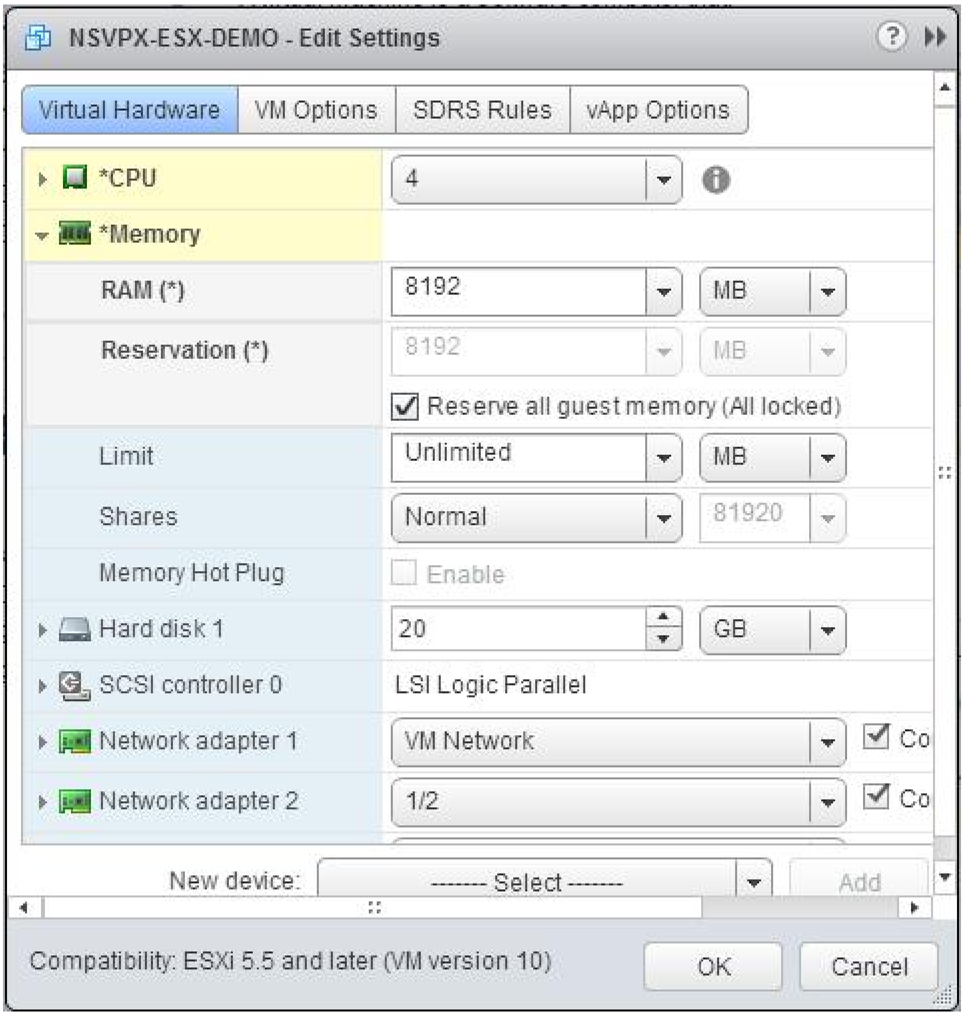
    a. In the **RAM** drop-down list, select the size of the RAM. It must be the number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the RAM must be 4 x 2 GB = 8 GB.

    > **Note:**
    >
    > For an Advanced or Premium edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

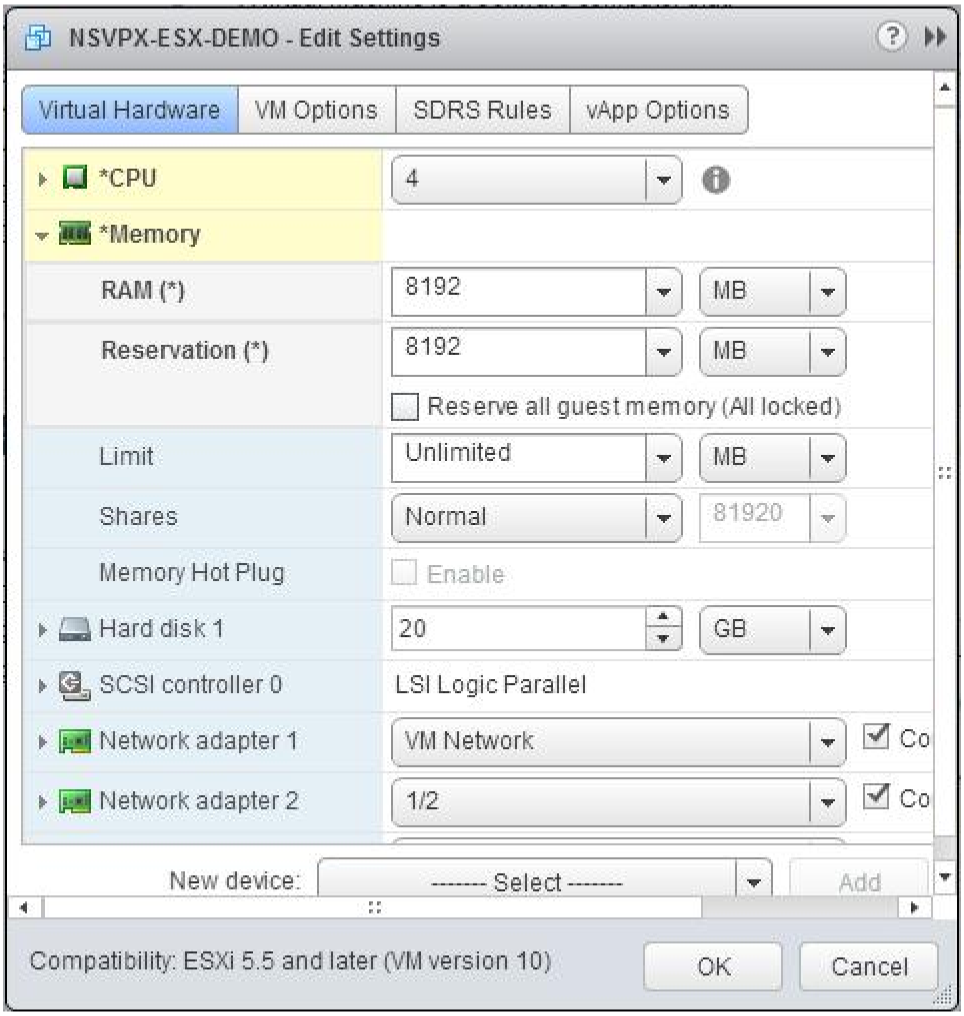


b. In the Reservation drop-down list, enter the value for the memory reservation, and select the Reserve all guest memory (All locked) check box. The memory reservation must be the number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation must be 4 x 2 GB = 8 GB.

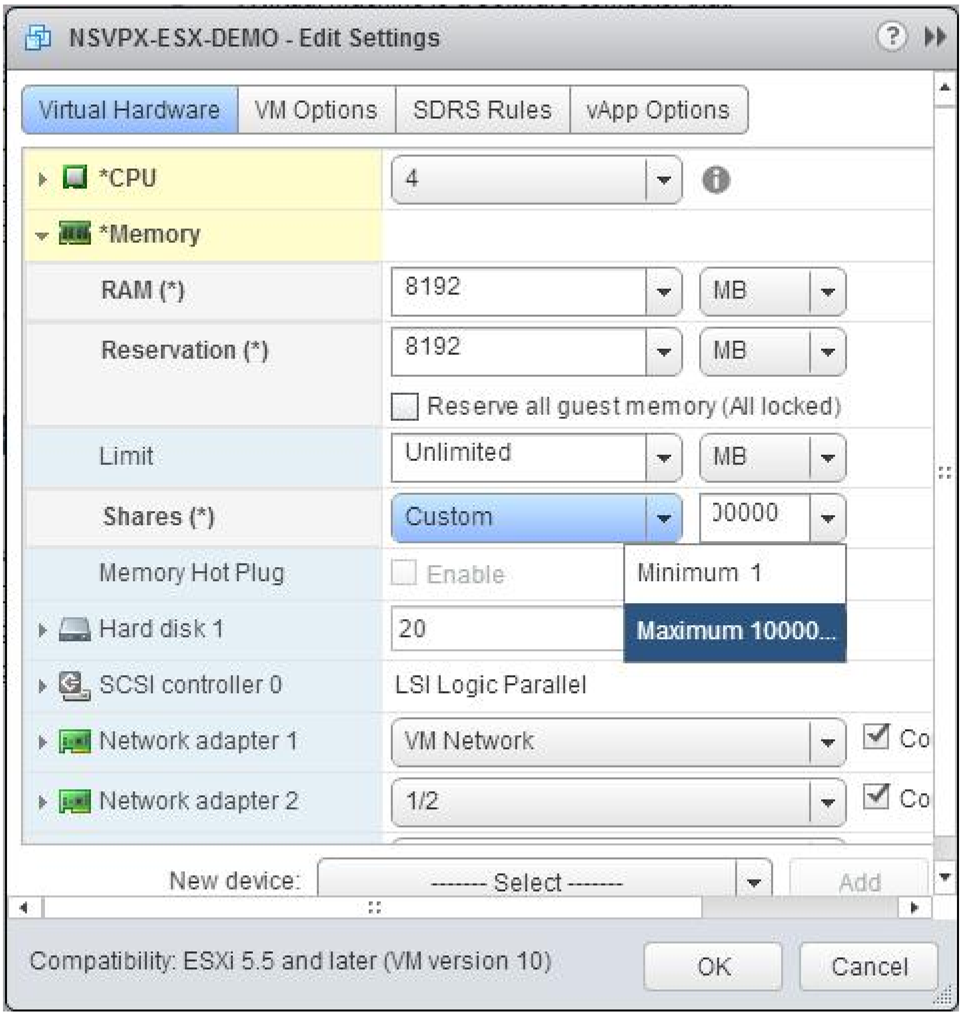
> **Note:**
>
> For an Advanced or Premium edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.



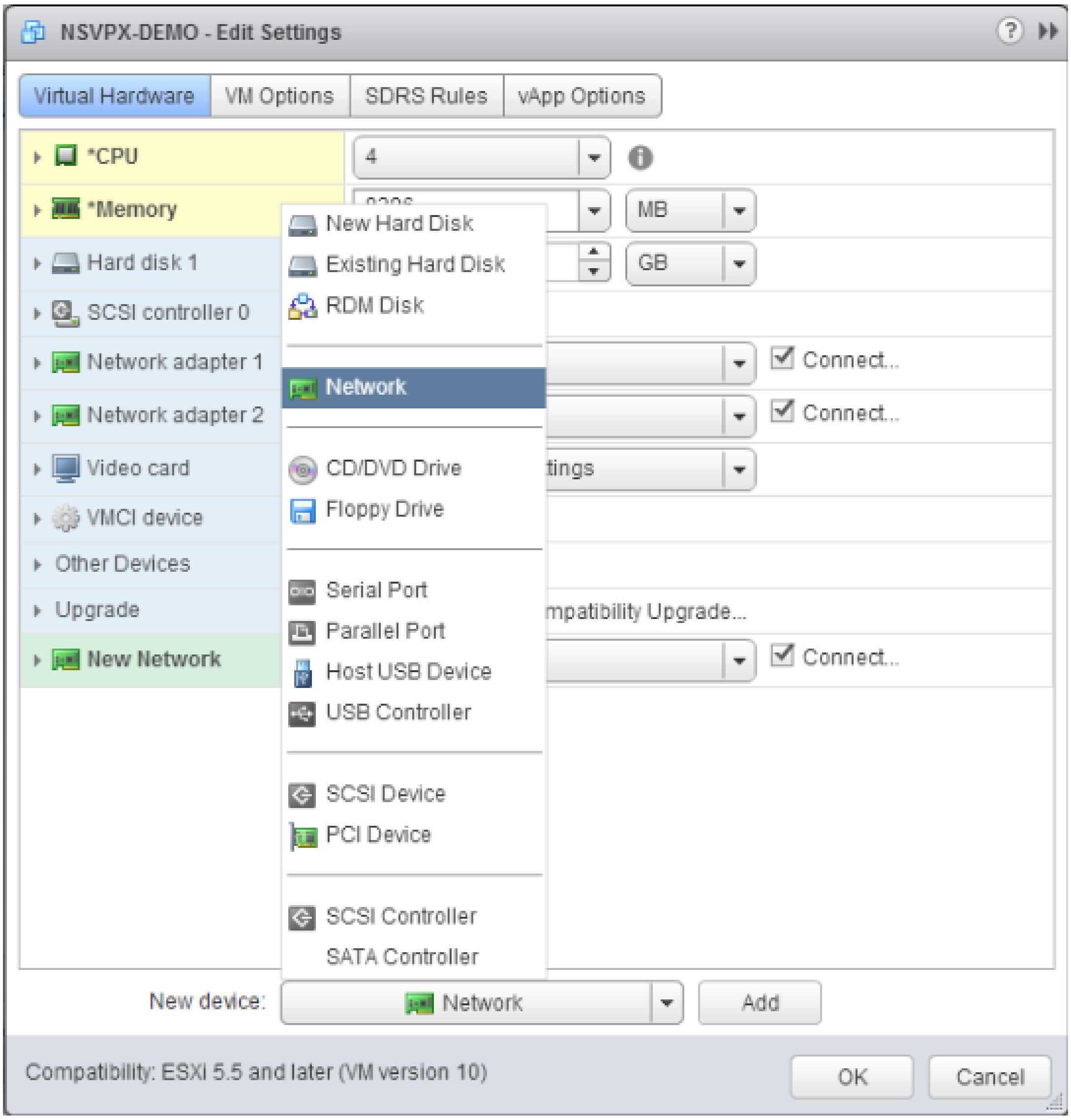
c. In the Limit drop-down list, select the number that is shown as the maximum value.



d. In the Shares drop-down lists, select Custom and the number that is shown as the maximum value.



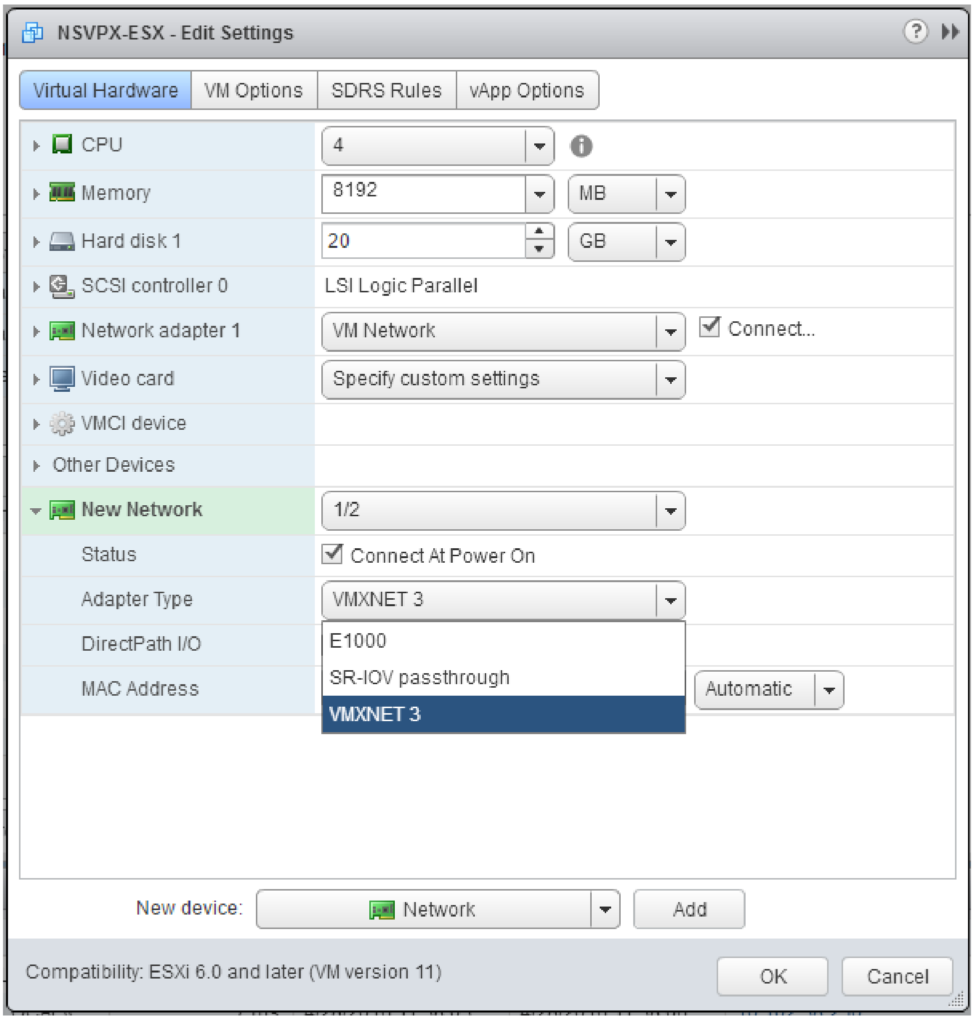
7. Add a VMXNET3 network interface. From the New device drop-down list, select Network and click Add.



8. In the New Network section, from the drop-down list, select the network interface, and do the following:

   a. In the Adapter Type drop-down list, select VMXNET3.

   > **Important:**
   >
   > The default E1000 network interface and VMXNET3 cannot coexist, make sure that you remove the E1000 network interface and use VMXNET3 (0/1) as the management interface.



9. Click **OK**.
10. Power on the NetScaler VPX instance.
11. Once the NetScaler VPX instance powers on, you can use the following command to verify the configuration:

    > show interface summary

The output must show all the interfaces that you configured:

```
> show interface summary
--------------------------------------------------------------------------------
     Interface  MTU        MAC                    Suffix
--------------------------------------------------------------------------------
1     0/1       1500       00:0c:29:89:1d:0e     NetScaler Vir...rface,
    VMXNET3
2     1/1       9000       00:0c:29:89:1d:18     NetScaler Vir...rface,
    VMXNET3
```



```
3    1/2      9000      00:0c:29:89:1d:22    NetScaler Vir...rface, VMXNET3
4    LO/1     9000      00:0c:29:89:1d:0e    Netscaler Loopback interface
```

> **Note:**
>
> After you add a VMXNET3 interface and restart the NetScaler VPX appliance, the VMware ESX hypervisor might change the order in which the NIC is presented to the VPX appliance. So, network adapter 1 might not always remain 0/1, resulting in loss of management connectivity to the VPX appliance. To avoid this issue, change the virtual network of the network adapter accordingly.
>
> This is a VMware ESX hypervisor limitation.

Configure a NetScaler VPX instance to use SR-IOV network interface

After you have installed and configured the NetScaler VPX instance on VMware ESX, you can use the VMware vSphere web client to configure the virtual appliance to use single root I/O v virtualization (SR-IOV) network interfaces.

Limitations

A NetScaler VPX configured with SR-IOV network interface has the following limitations:

- The following features are not supported on SR-IOV interfaces using the Intel 82599 10G NIC on ESX VPX:
  - L2 mode switching
  - Static Link Aggregation and LACP
  - Clustering
  - Admin partitioning [Shared VLAN mode]
  - High Availability [Active - Active mode]
  - Jumbo frames
  - IPv6
- The following features are not supported on the SR-IOV interface with an Intel 82599 10G NIC on KVM VPX:
  - Static Link Aggregation and LACP
  - L2 mode switching
  - Clustering
  - Admin partitioning [Shared VLAN mode]

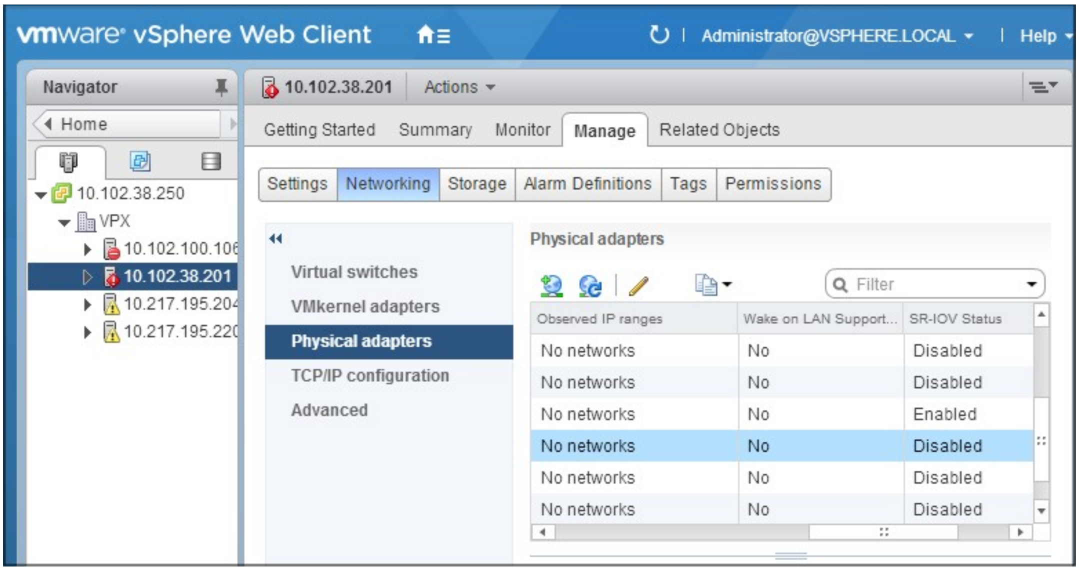
- High Availability [Active –Active mode]
- Jumbo frames
- IPv6
- VLAN configuration on Hypervisor for SR-IOV VF interface through `ip link` command is not supported

Prerequisite

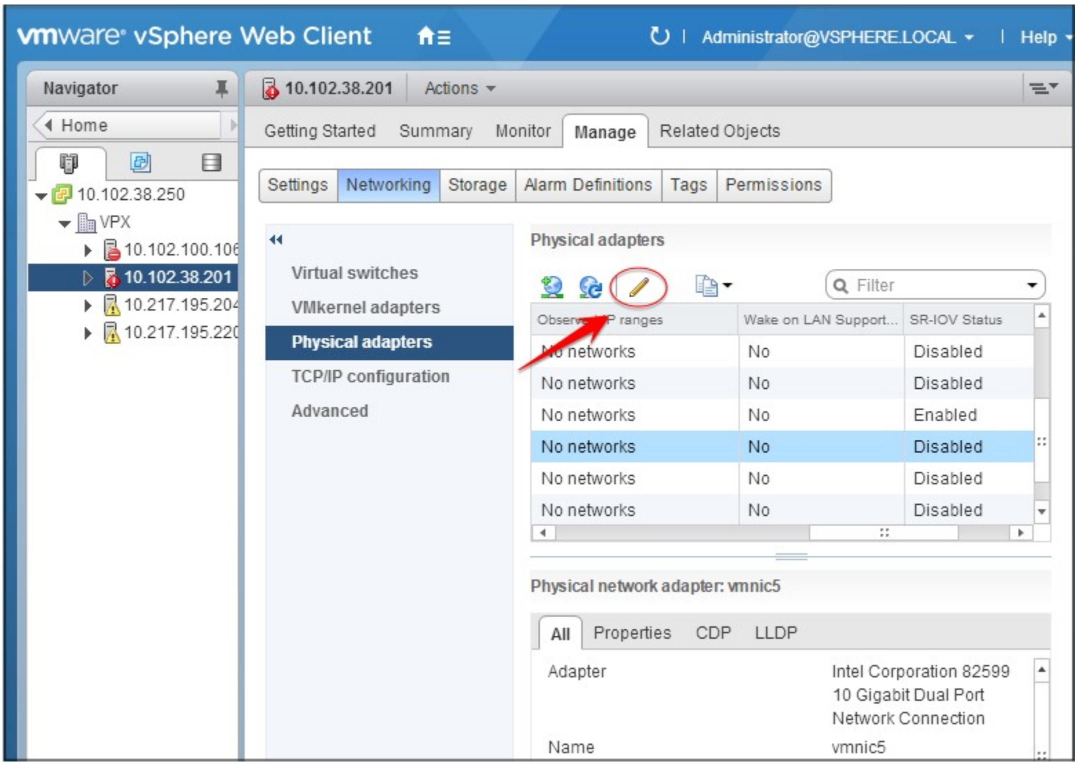
- Make sure that you add any of the following NICs to the ESX host:
  - Intel 82599 NIC, IXGBE driver version 3.7.13.7.14iov or later is recommended.
  - Mellanox ConnectX-4 NIC
- Enable SR-IOV on the host physical adapter.

  Follow this procedure to enable SR-IOV on the host physical adapter:

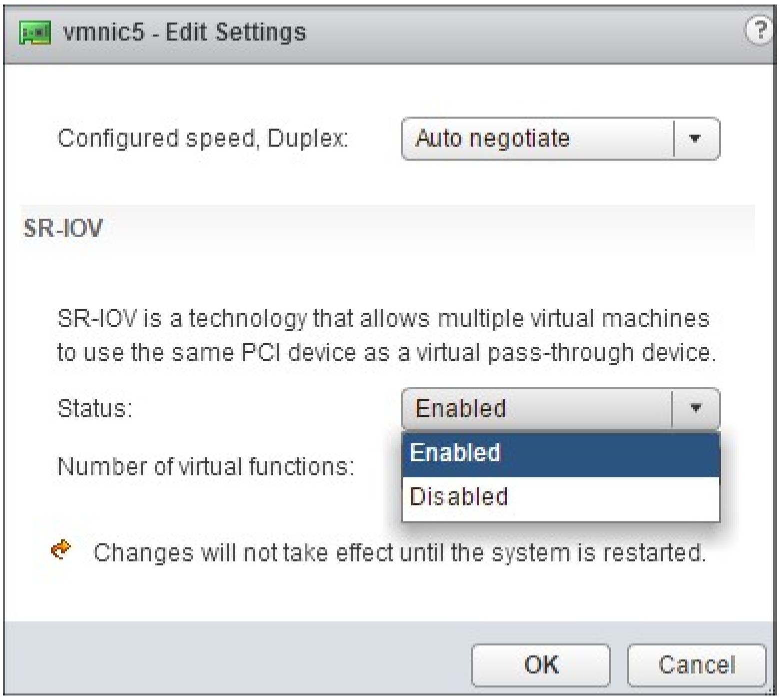
  1. In the vSphere Web Client, navigate to the Host.
  2. On the **Manage > Networking** tab, select **Physical adapters**. The SR-IOV Status field shows whether a physical adapter supports SR-IOV.

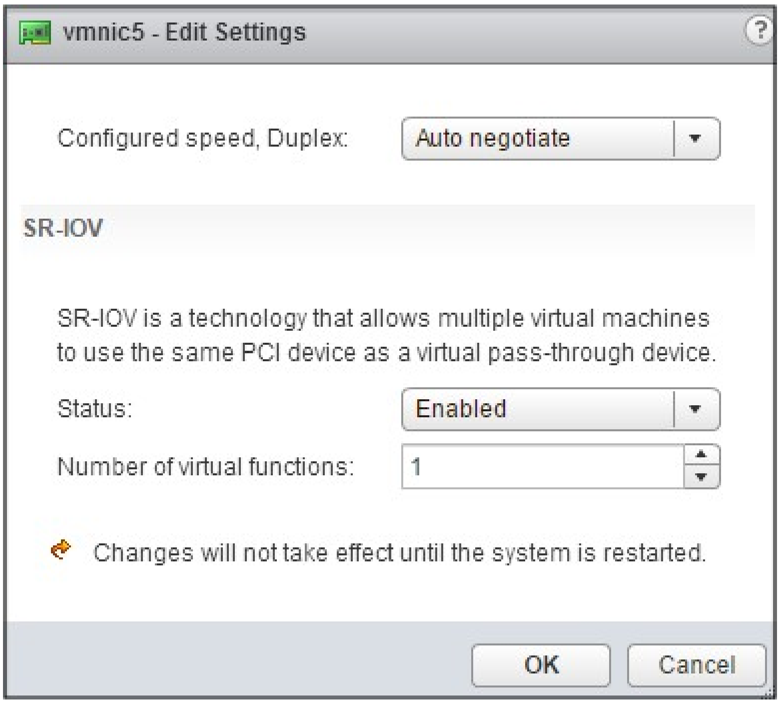
  3. Select the physical adapter, and then click the pencil icon to open the **Edit Settings** dialog box.



4. Under SR-IOV, select **Enabled** from the **Status** drop-down list.



5. In the **Number of virtual functions** field, enter the number of virtual functions that you want to configure for the adapter.



6. Click **OK**.
7. Restart the host.

- Create a Distributed Virtual Switch (DVS) and `Portgroups`. For instructions, see the VMware Documentation.

> **Note:**
>
> Citrix has qualified the SR-IOV configuration on DVS and `Portgroups` only.

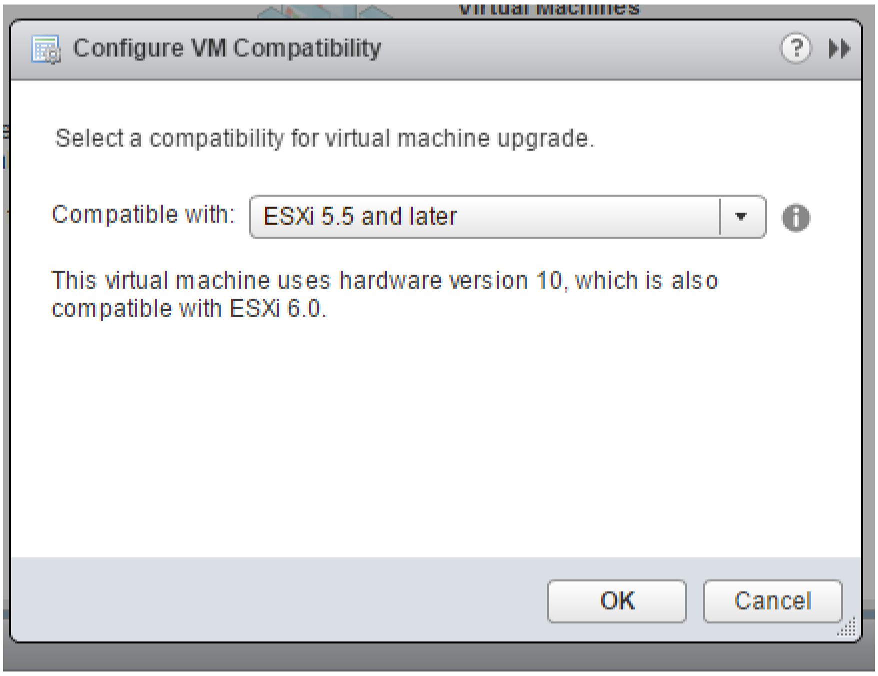
To configure NetScaler VPX instances to use SR-IOV network interface by using VMware vSphere Web Client:

1. In the vSphere Web Client, select **Hosts and Clusters**.
2. Upgrade the Compatibility setting of the NetScaler VPX instance to ESX 5.5 or later, as follows:

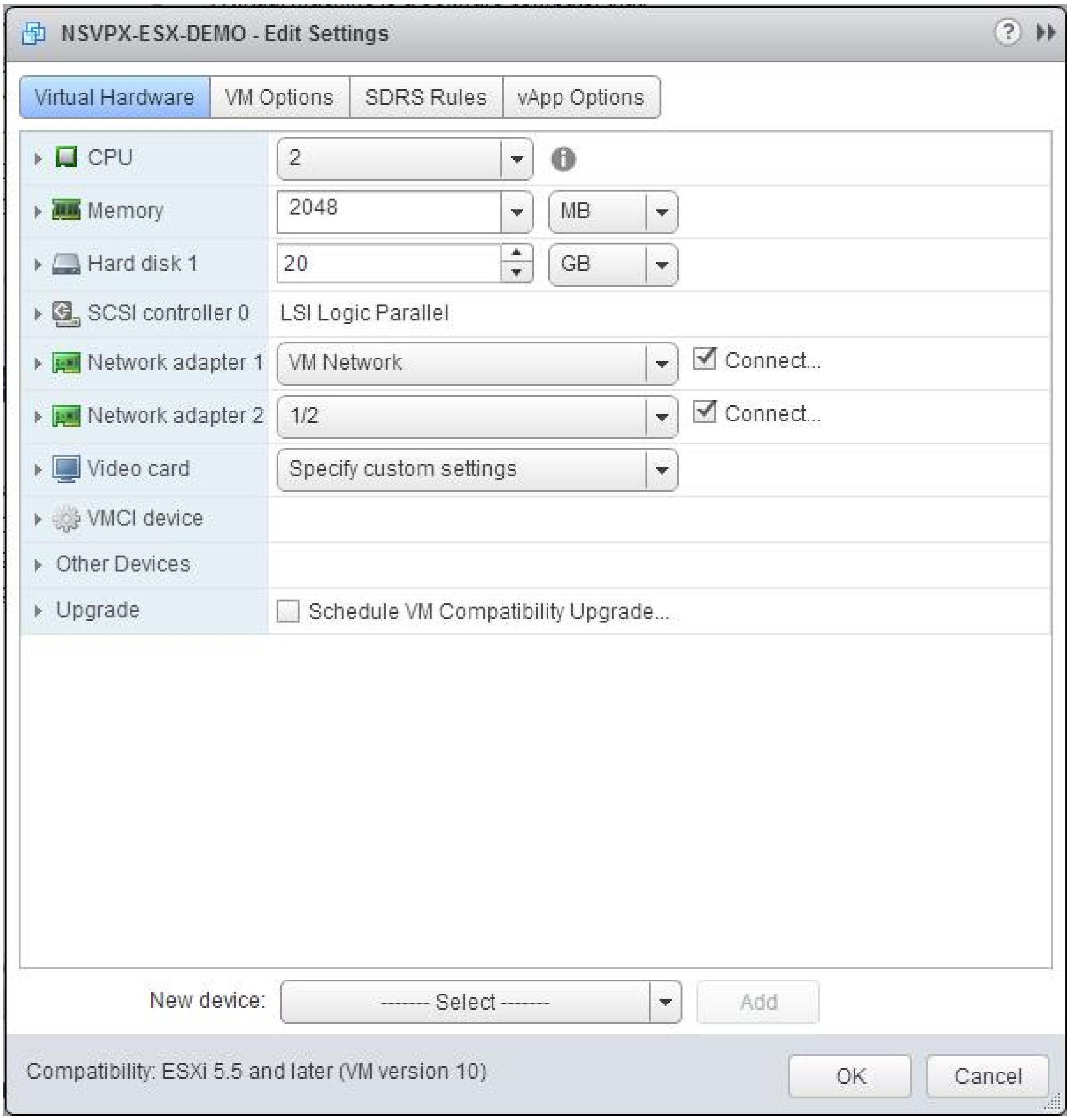
   a. Power off the NetScaler VPX instance.

   b. Right-click the NetScaler VPX instance and select **Compatibility > Upgrade VM Compatibility**.

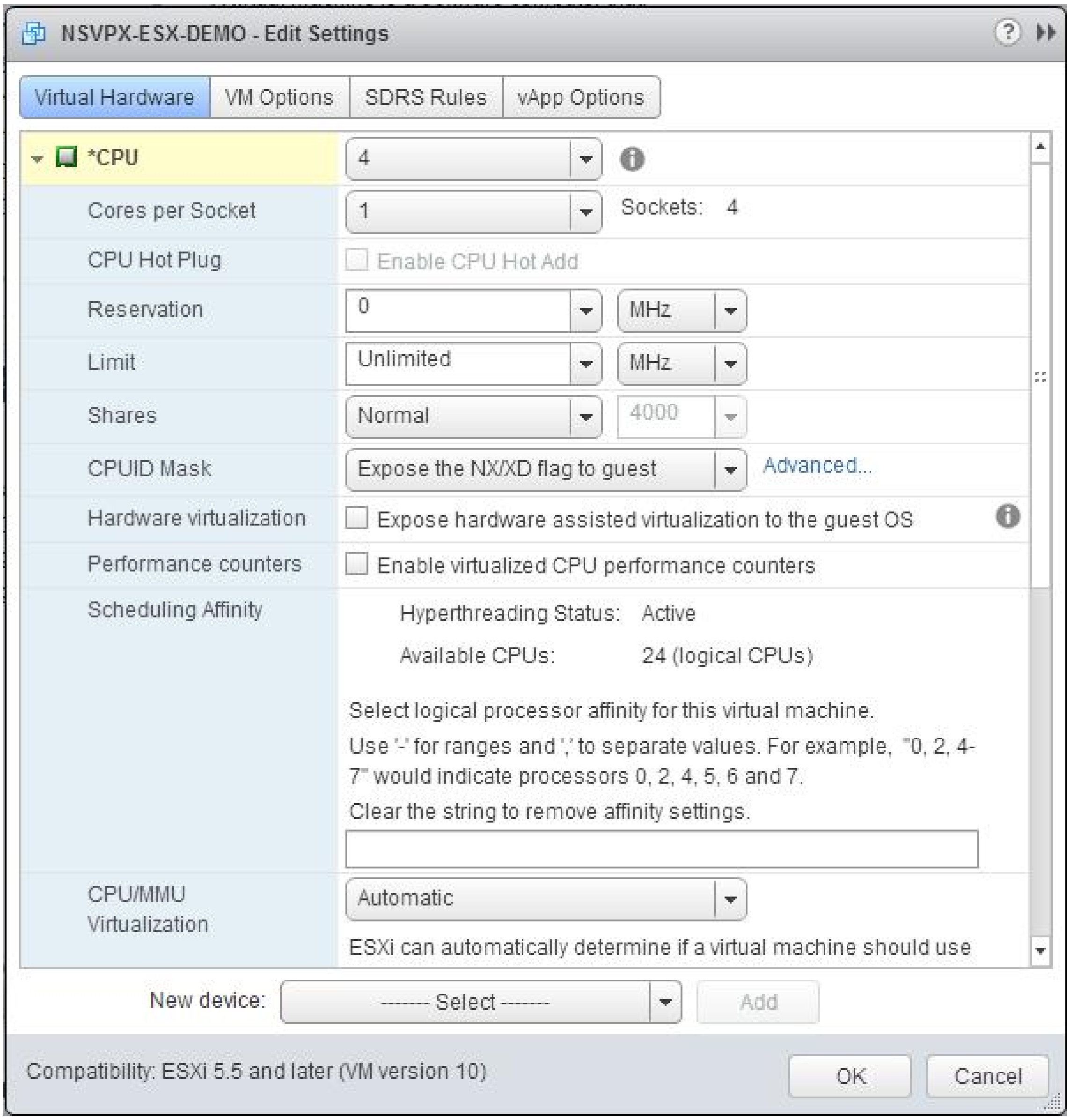
   c. In the **Configure VM Compatibility** dialog box, select **ESXi 5.5 and later** from the **Compatible with** drop-down list and click **OK**.



3. Right-click on the NetScaler VPX instance and click **Edit Settings**.



4. In the **<virtual_appliance> - Edit Settings** dialog box, click the **CPU** section.



5. In the **CPU** section, update the following settings:

   - Number of CPUs
   - Number of Sockets
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

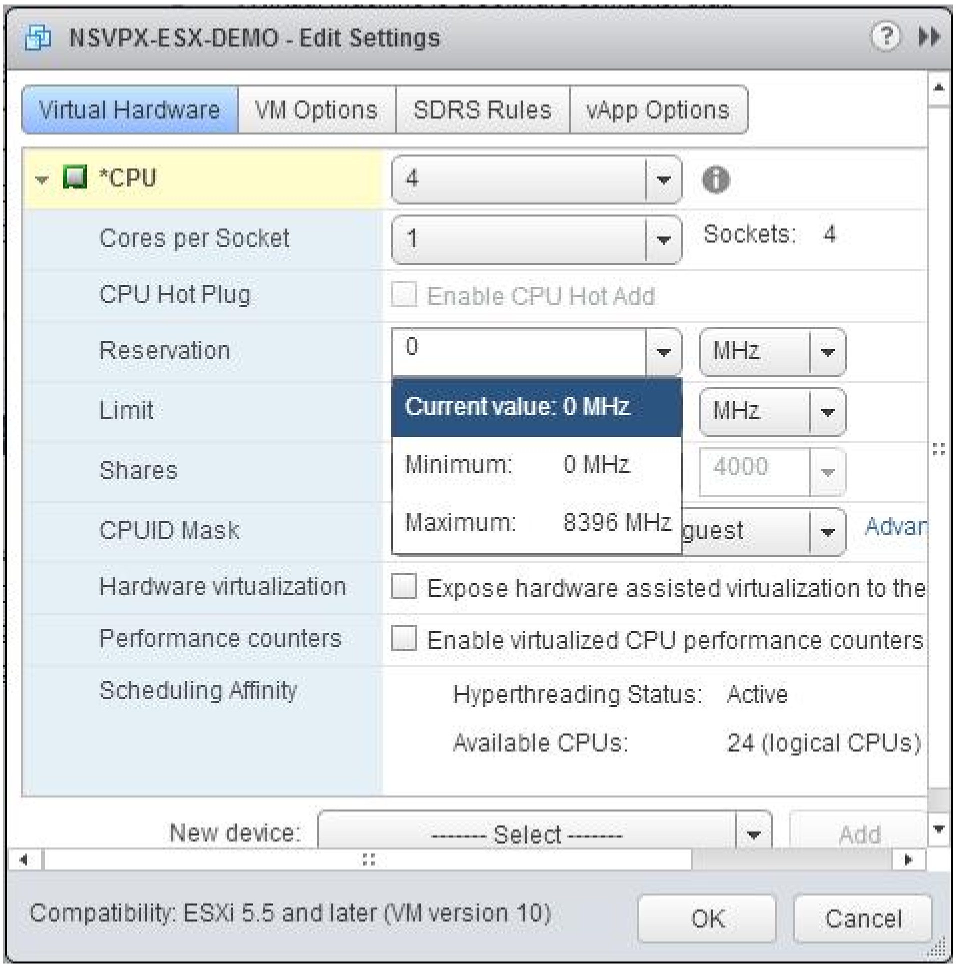
   a. In the **CPU** drop-down list, select the number of CPUs to assign to the virtual appliance.

   b. In the **Cores per Socket** drop-down list, select the number of sockets.

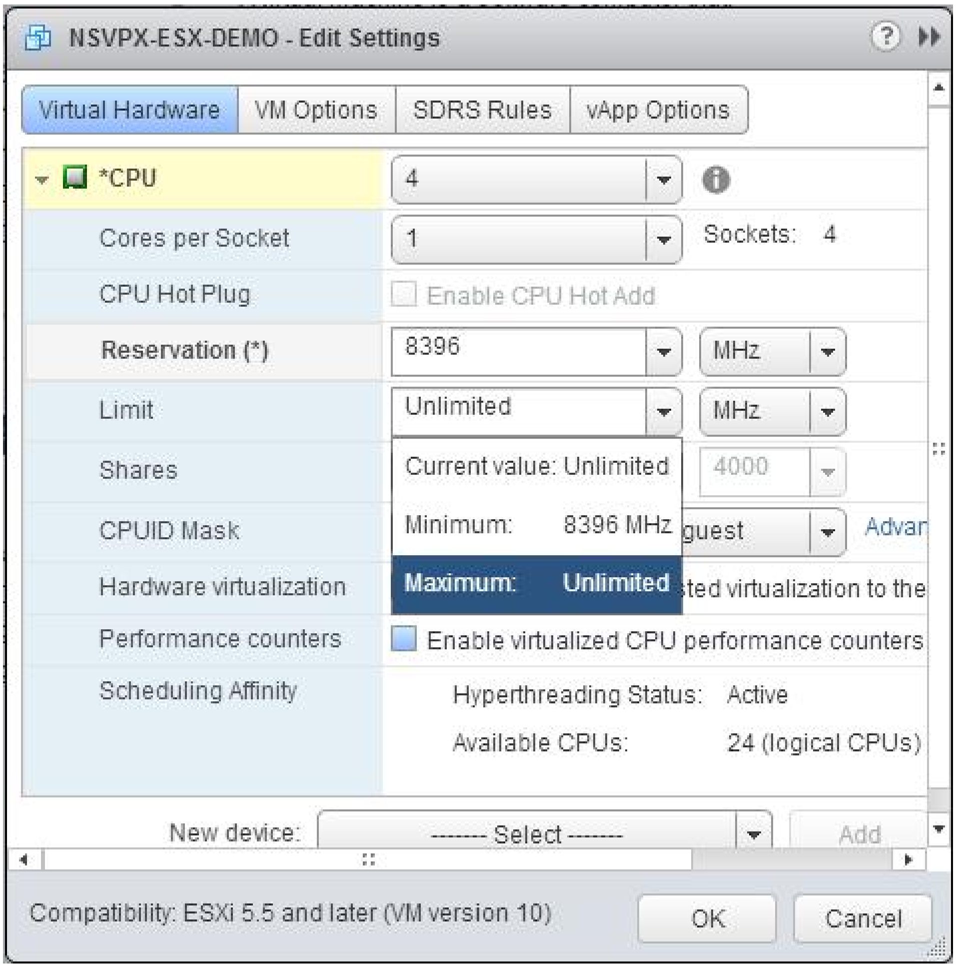
   c. (Optional) In the **CPU Hot Plug** field, select or clear the **Enable CPU Hot Add** check box.

> **Note:**
>
> Citrix recommends accepting the default (disabled).

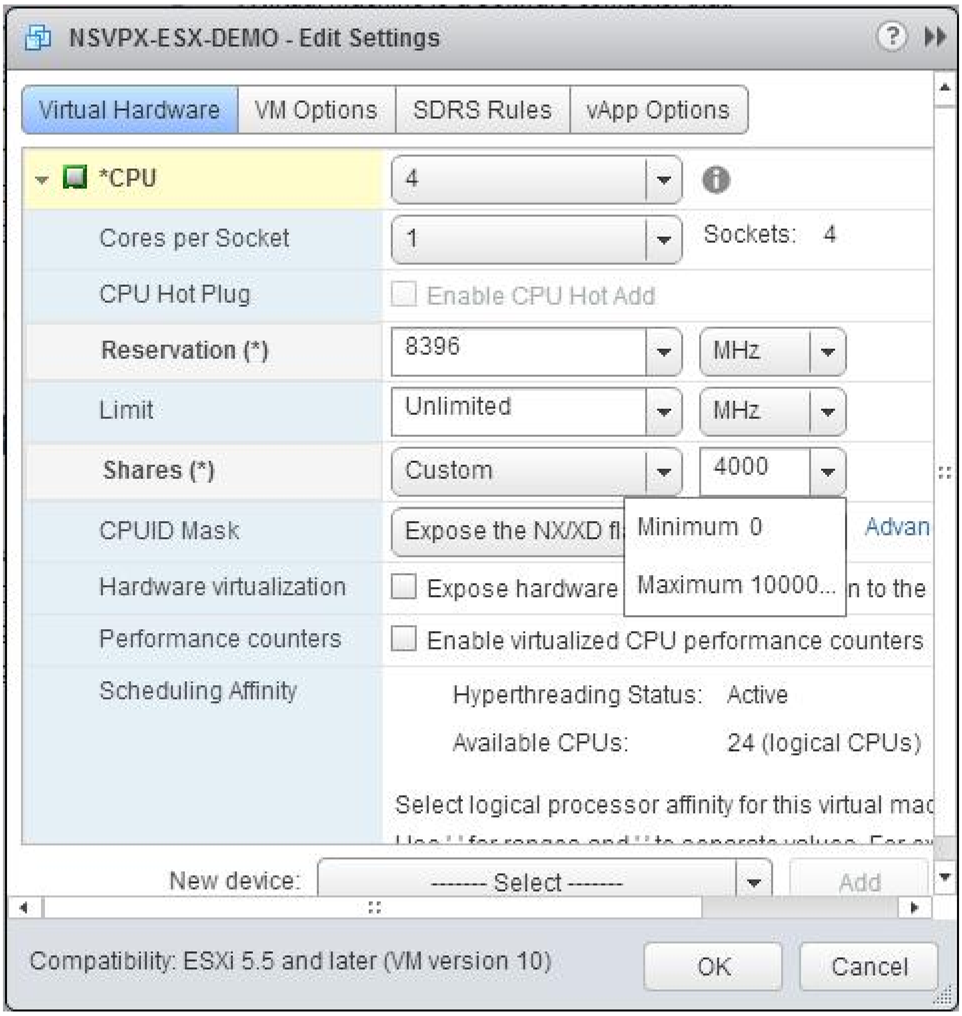
d. In the **Reservation** drop-down list, select the number that is shown as the maximum value.



e. In the **Limit** drop-down list, select the number that is shown as the maximum value.



f. In the **Shares** drop-down lists, select **Custom** and the number that is shown as the maximum value.



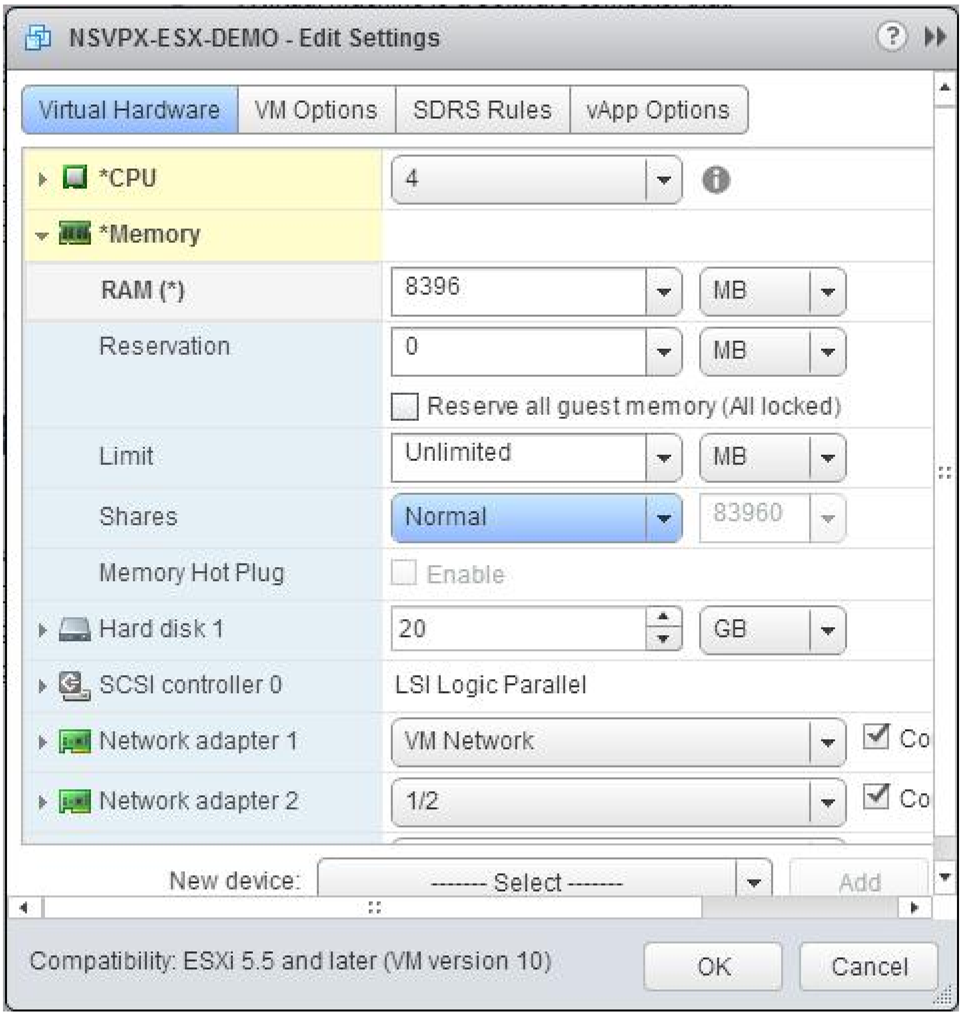
6. In the **Memory** section, update the following settings:

   - Size of RAM
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

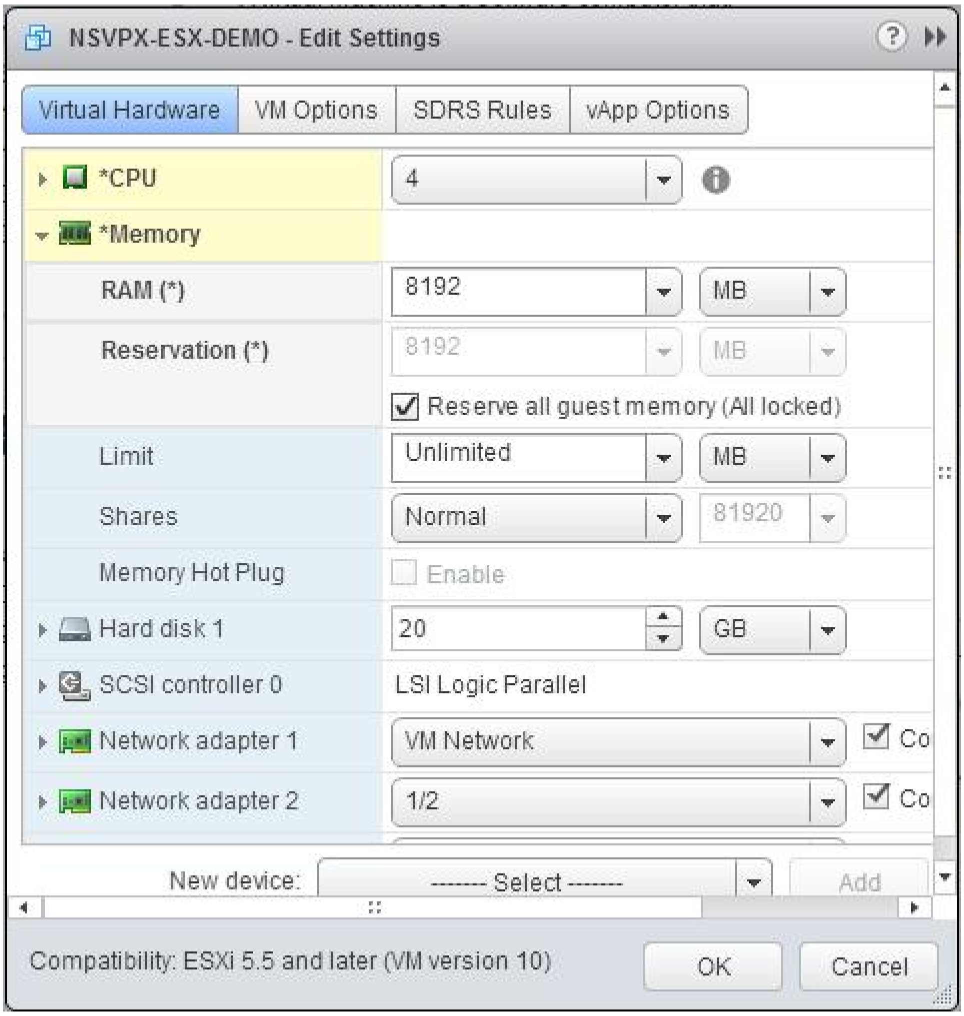
   a. In the **RAM** drop-down list, select the size of the RAM. It must be the number of vCPUs x 2 GB. For example, if the number of vCPU is 4 then RAM = 4 x 2 GB = 8 GB.

   > **Note:**
   >
   > For Advanced or Premium edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

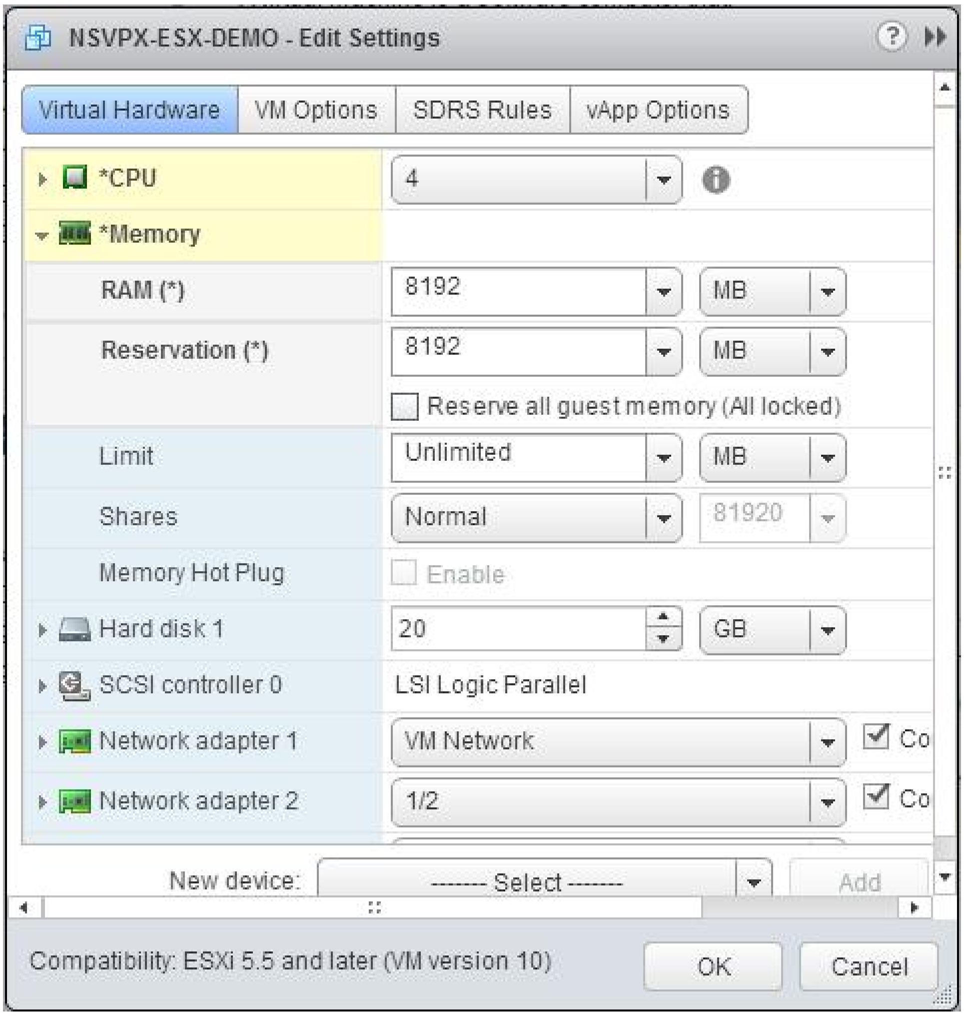


b. In the **Reservation** drop-down list, enter the value for the memory reservation, and select the **Reserve all guest memory (All locked)** check box. The memory reservation must be number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation must be 4 x 2 GB = 8 GB.

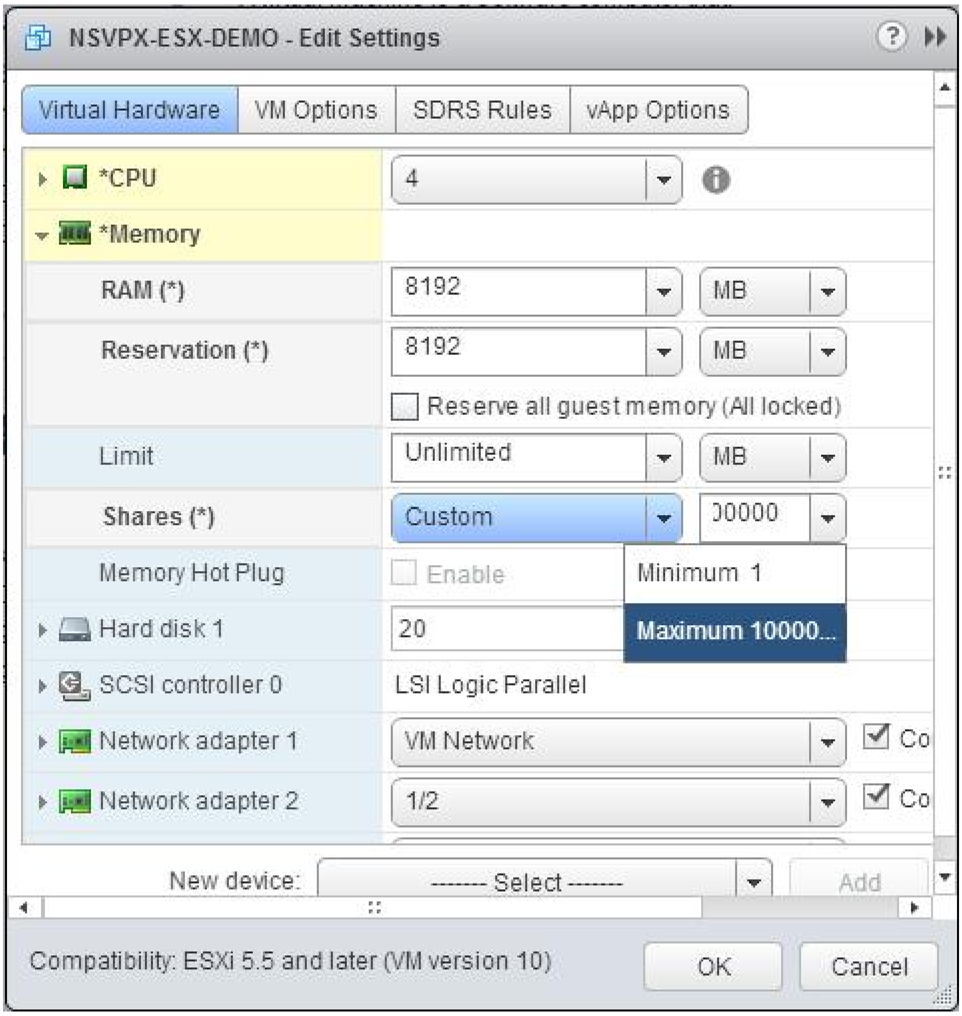
> **Note:**
>
> For Advanced or Premium edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.



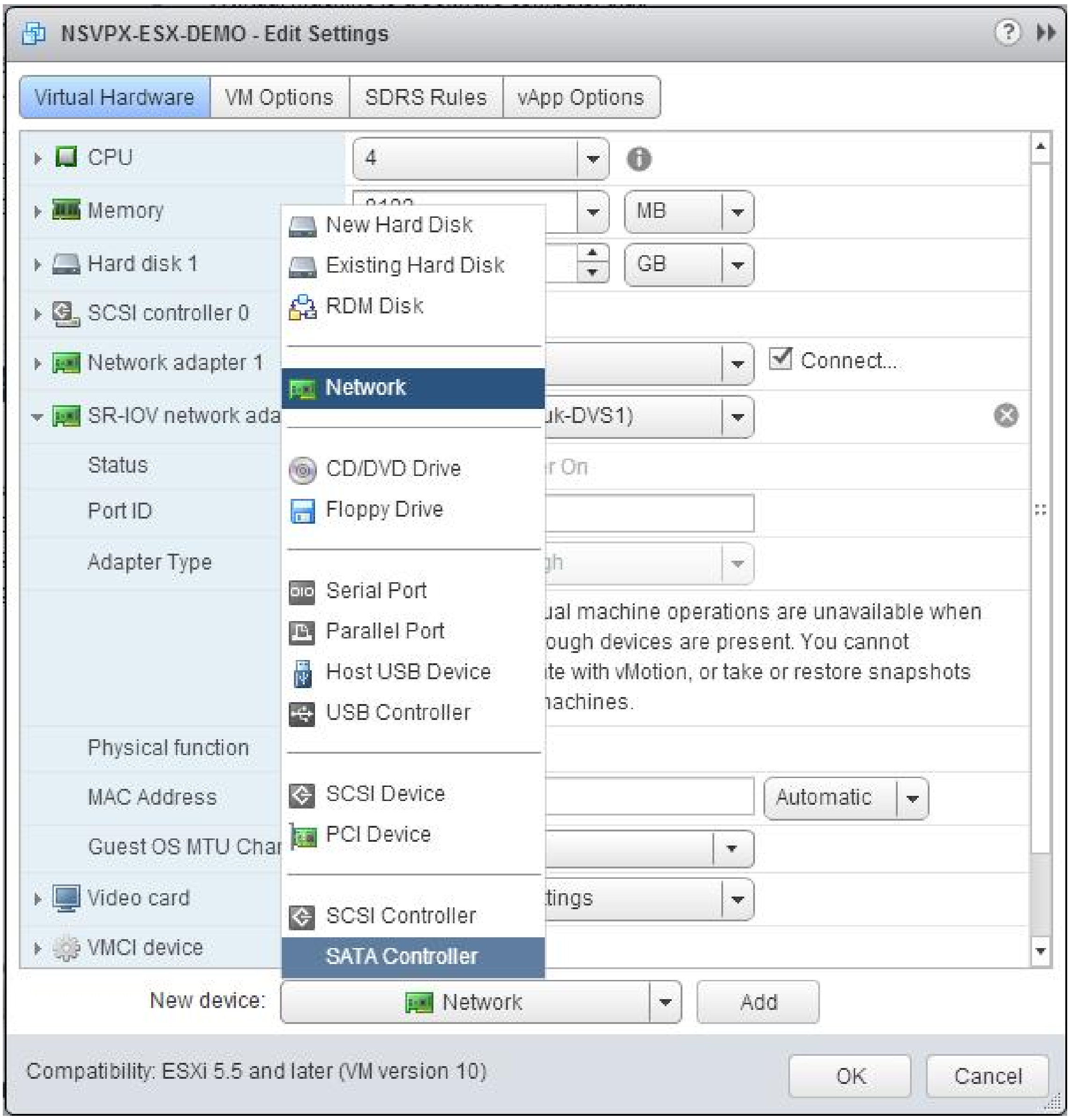
c. In the **Limit** drop-down list, select the number that is shown as the maximum value.



d. In the **Shares** drop-down lists, select **Custom**, and select the number that is shown as the maximum value.

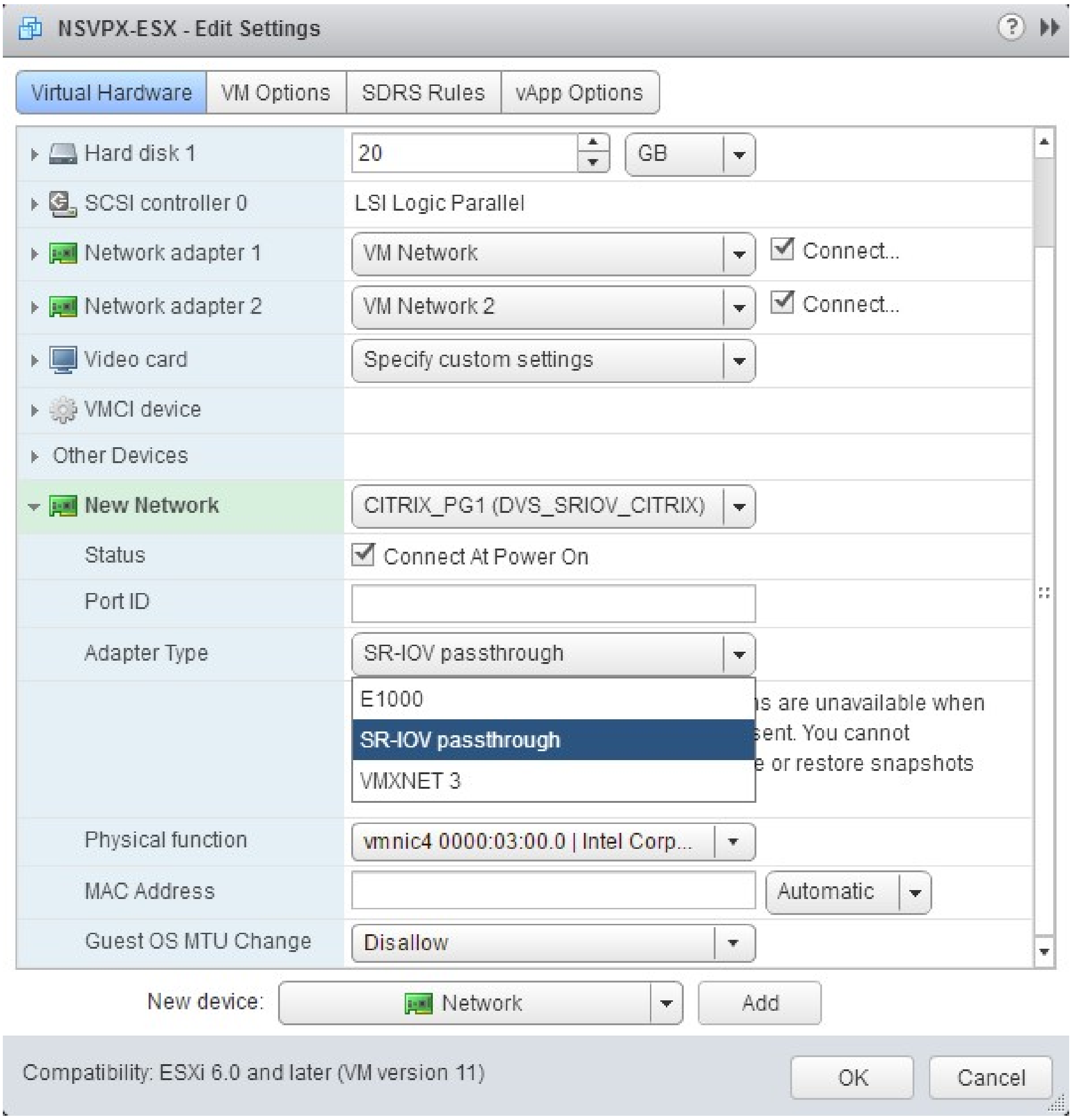


7. Add an SR-IOV network interface. From the **New device** drop-down list, select **Network** and click **Add**.

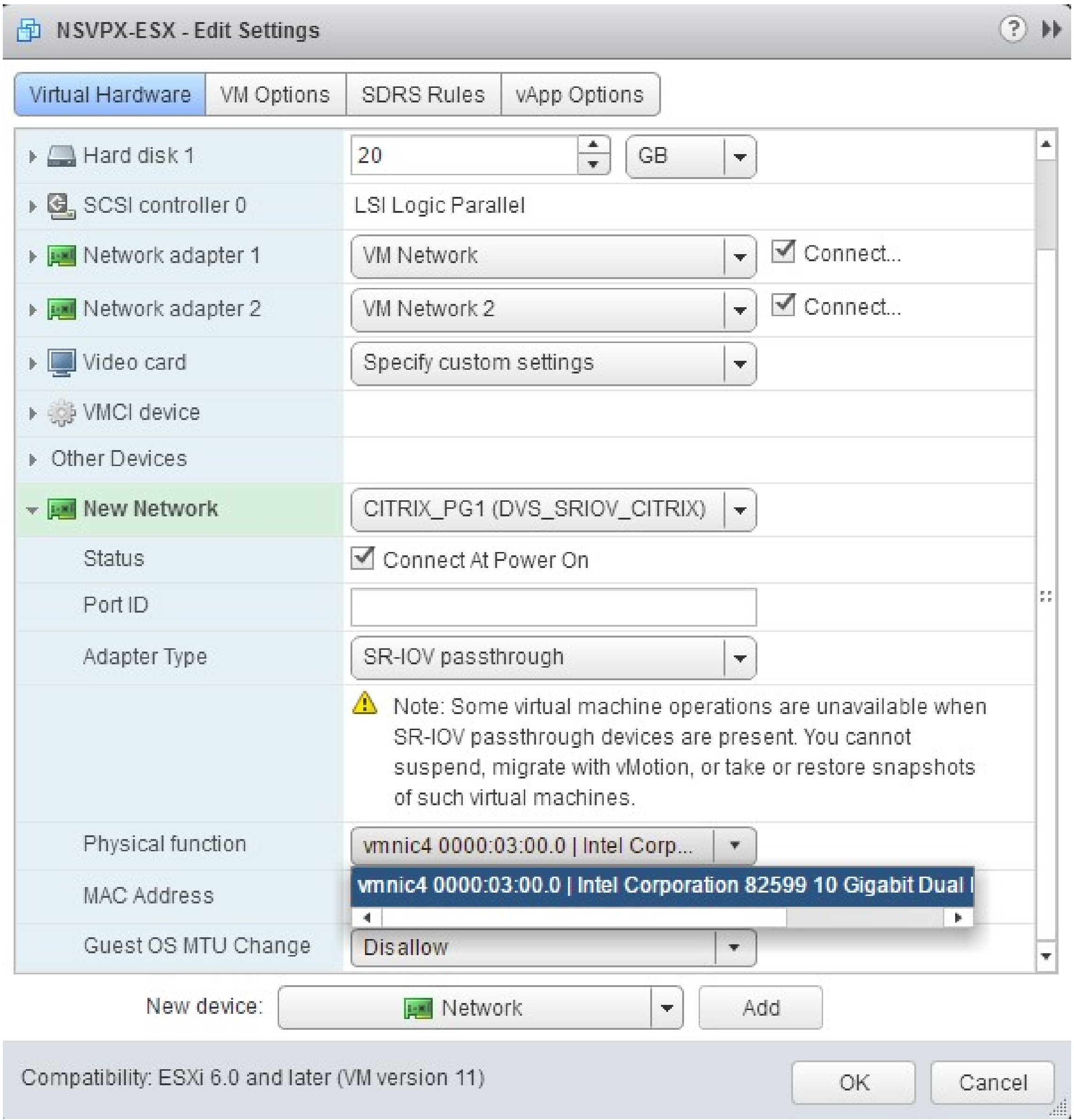


8. In the **New Network** section. From the drop-down list, select the `Portgroup` that you created, and do the following:

   a. In the **Adapter Type** drop-down list, select **SR-IOV passthrough**.

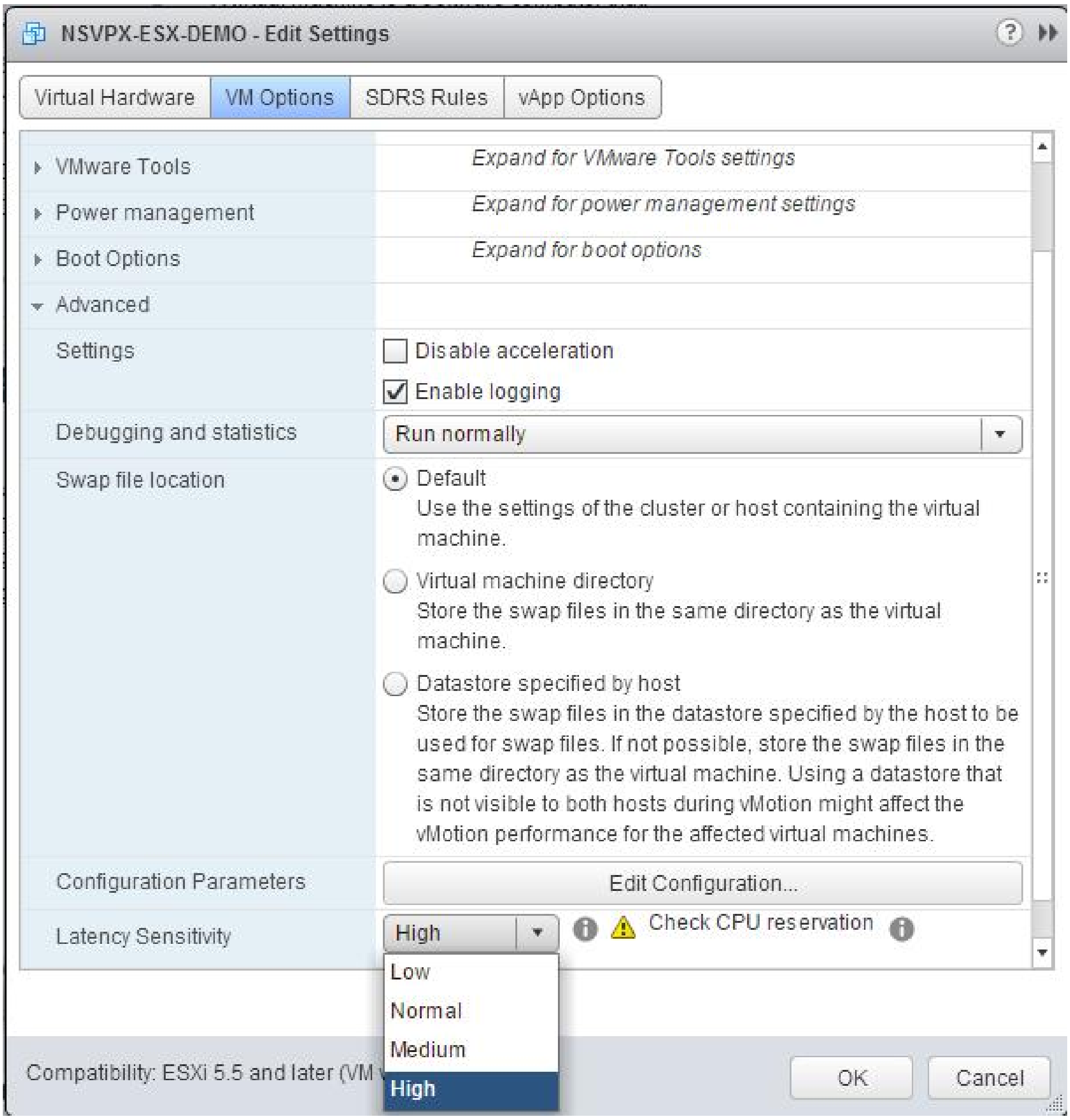


b. In the **Physical function** drop-down list, select the physical adapter mapped with the `Portgroup`.



c. In the **Guest OS MTU Change** drop-down list, select **Disallow**.

9. In the **<virtual_appliance> - Edit Settings** dialog box, click the **VM Options** tab.
10. On the **VM Options** tab, select the **Advanced** section. From the **Latency Sensitivity** drop-down list, select **High**.



11. Click **OK**.
12. Power on the NetScaler VPX instance.
13. Once the NetScaler VPX instance powers on, you can use the following command to verify the configuration:

    ```
    show interface summary
    ```

The output must show all the interfaces that you configured:

```
> show interface summary
--------------------------------------------------------------------------------------

        Interface   MTU        MAC                      Suffix
```

```
-----------------------------------------------------------------------------
1      0/1        1500       00:0c:29:1b:81:0b    NetScaler Virtual
    Interface
2      10/1       1500       00:50:56:9f:0c:6f    Intel 82599 10G VF
    Interface
3      10/2       1500       00:50:56:9f:5c:1e    Intel 82599 10G VF
    Interface
4      10/3       1500       00:50:56:9f:02:1b    Intel 82599 10G VF
    Interface
5      10/4       1500       00:50:56:9f:5a:1d    Intel 82599 10G VF
    Interface
6      10/5       1500       00:50:56:9f:4e:0b    Intel 82599 10G VF
    Interface
7      LO/1       1500       00:0c:29:1b:81:0b    Netscaler Loopback
    interface
 Done
> show inter 10/1
1)       Interface 10/1 (Intel 82599 10G VF Interface) #1
         flags=0xe460 <ENABLED, UP, UP, HAMON, 802.1q>
         MTU=1500, native vlan=55, MAC=00:50:56:9f:0c:6f, uptime 0
             h21m53s
         Actual: media FIBER, speed 10000, duplex FULL, fctl NONE,
             throughput 10000
         LLDP Mode: NONE,                 LR Priority: 1024

         RX: Pkts(838020742) Bytes(860888485431) Errs(0) Drops(2527)
             Stalls(0)
         TX: Pkts(838149954) Bytes(860895860507) Errs(0) Drops(0) Stalls
             (0)
         NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
         Bandwidth thresholds are not set.
 Done
```

Migrating the NetScaler VPX from E1000 to SR-IOV or VMXNET3 Network Interfaces

You can configure your exiting NetScaler VPX instances that use E1000 network interfaces to use SR-IOV or VMXNET3 network interfaces.

To configure an existing NetScaler VPX instance to use SR-IOV network interfaces, see Configure a NetScaler VPX instance to use SR-IOV network interface.

To configure an existing NetScaler VPX instance to use VMXNET3 network interfaces, see Configure a NetScaler VPX instance to use VMXNET3 network interface.

Configure a NetScaler VPX instance to use PCI passthrough network interface

Overview

After you have installed and configured a NetScaler VPX instance on VMware ESX Server, you can use the vSphere Web Client to configure the virtual appliance to use PCI passthrough network interfaces.

The PCI passthrough feature allows a guest virtual machine to directly access physical PCI and PCIe devices connected to a host.

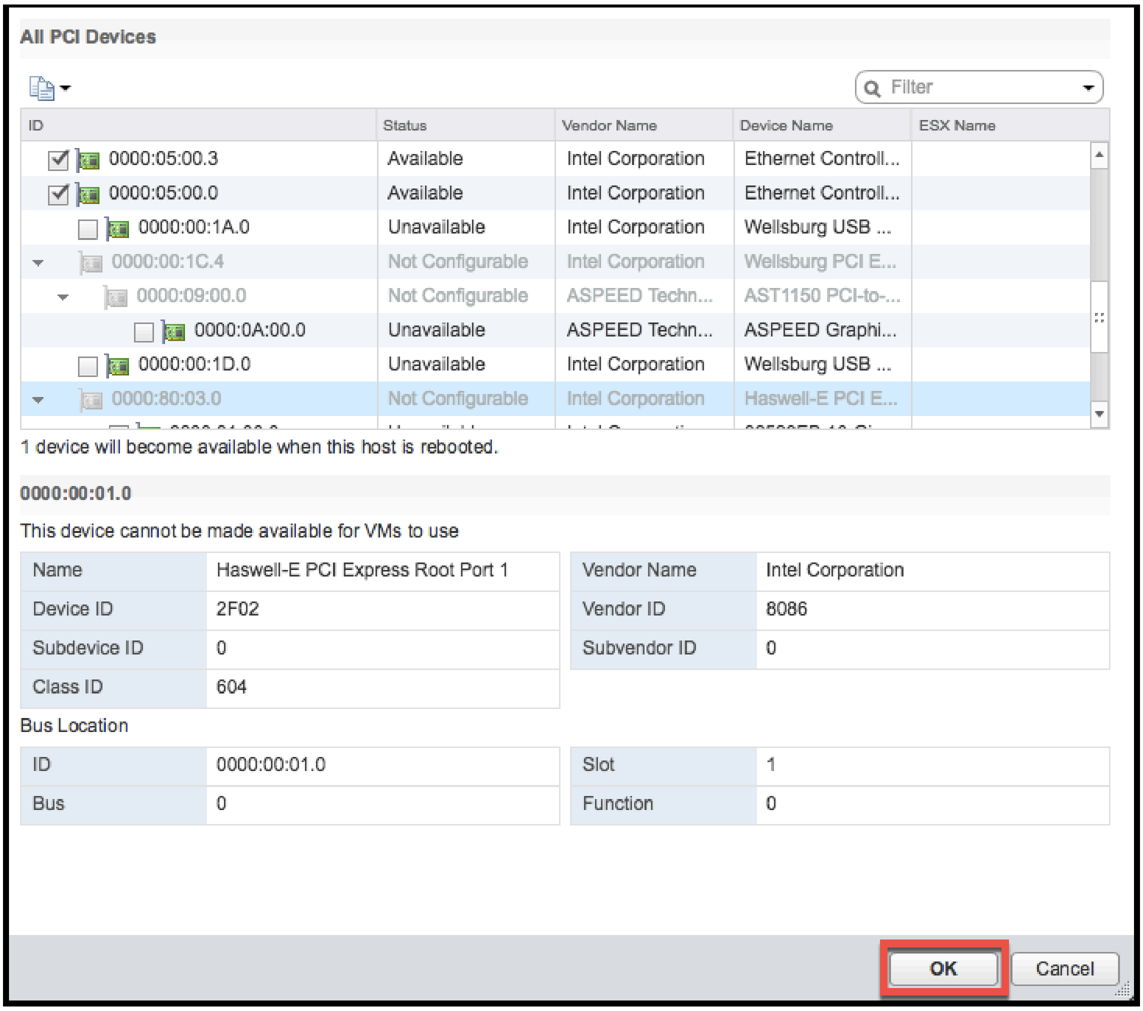
Prerequisites

- The firmware version of the Intel XL710 NIC on the host is 5.04.
- A PCI passthrough device connected to and configured on the host
- Supported NICs:
  - Intel X710 10G NIC
  - Intel XL710 Dual Port 40G NIC
  - Intel XL710 Single Port 40G NIC

Configure passthrough devices on a host

Before configuring a passthrough PCI device on a virtual machine, you must configure it on the host machine. Follow these steps to configure passthrough devices on a host.

1. Select the host from the Navigator panel of the vSphere Web Client.
2. Click **Manage** > **Settings** > **PCI Devices**. All available passthrough devices are displayed.
3. Right-click the device that you want to configure and click **Edit**.
4. The **Edit PCI Device Availability** window appears.
5. Select the devices to be used for passthrough and click **OK**.

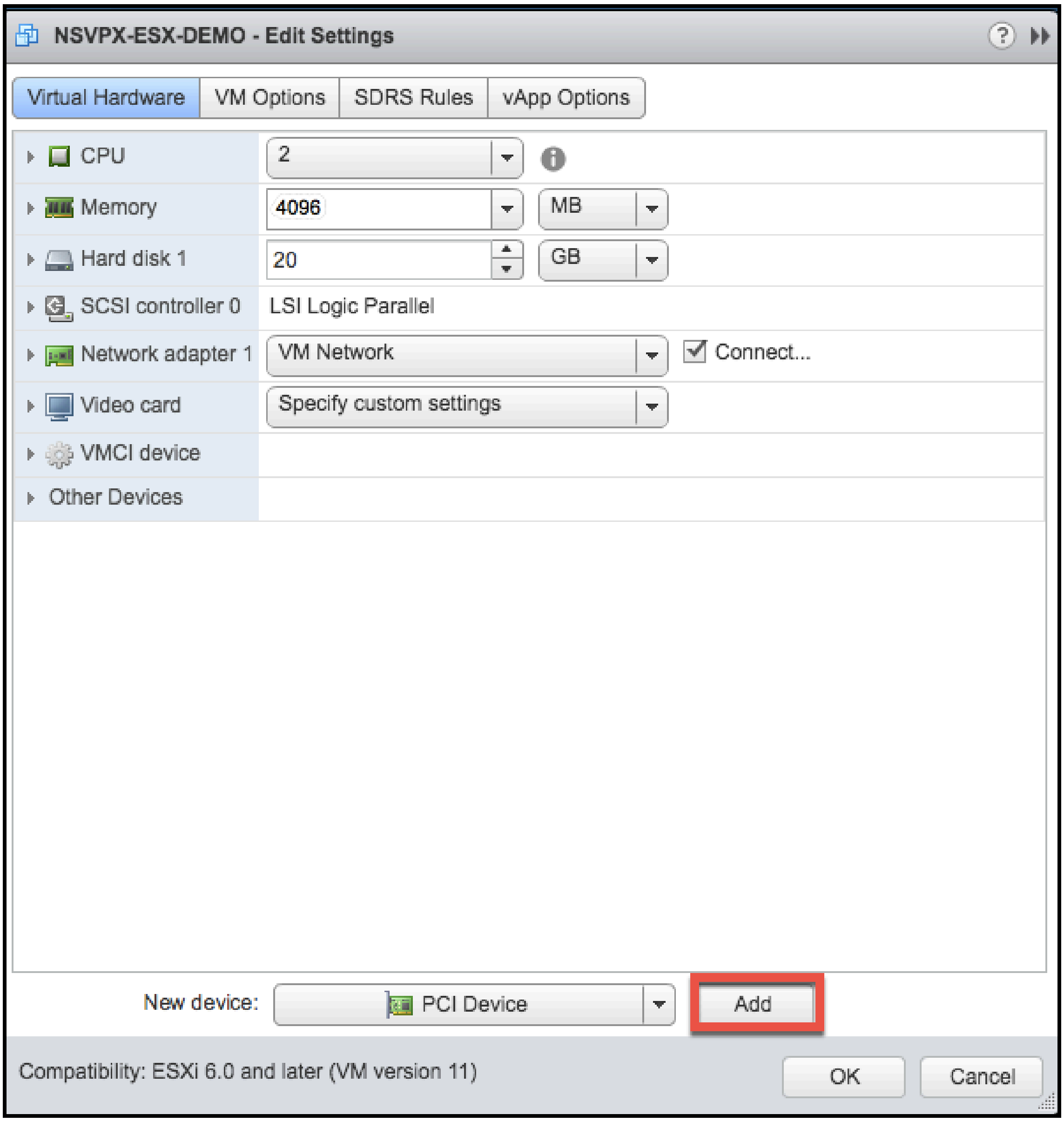


6. Restart the host machine.

Configure passthrough devices on a NetScaler VPX instance

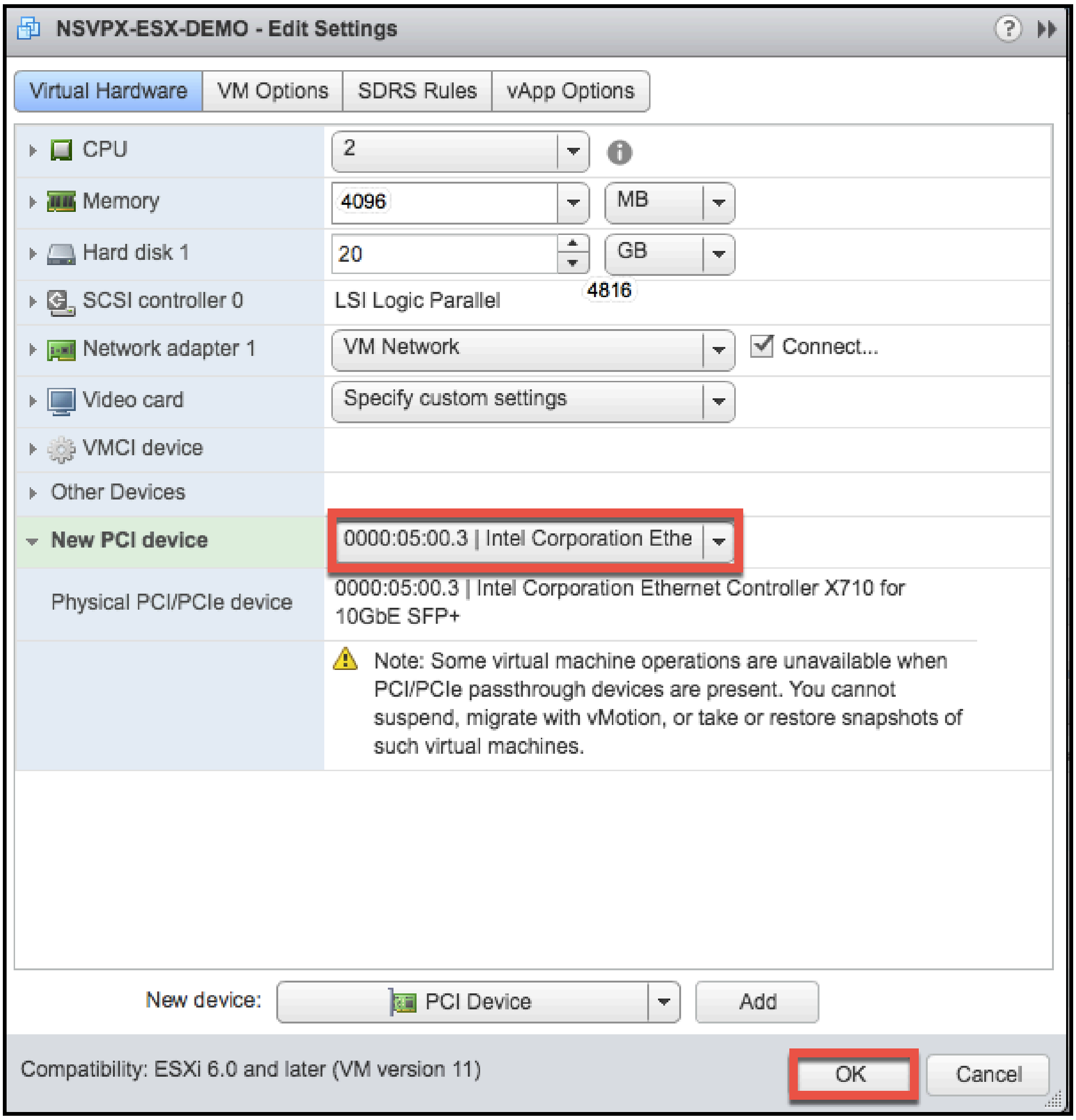
Follow these steps to configure a passthrough PCI device on a NetScaler VPX instance.

1. Power off the virtual machine.
2. Right-click the virtual machine and select **Edit Settings**.
3. On the **Virtual Hardware** tab, select **PCI Device** from the **New Device** drop-down menu, and click **Add**.



4. Expand **New PCI device** and select the passthrough device to connect to the virtual machine from the drop-down list and click **OK**.

> **Note:**
>
> VMXNET3 network interface and PCI Passthrough Network Interface cannot coexist.



5. Power on the guest virtual machine.

You have completed the steps to configuring NetScaler VPX to use PCI passthrough network interfaces.

Apply NetScaler VPX configurations at the first boot of the NetScaler appliance on VMware ESX hypervisor

You can apply the NetScaler VPX configurations during the first boot of the NetScaler appliance on the VMware ESX hypervisor. Therefore in certain cases, a specific setup or VPX instance is brought up in lesser time.

For more information on Preboot user data and its format, see Apply NetScaler VPX configurations at the first boot of the NetScaler appliance in cloud.

> **Note:**
>
> To bootstrap using preboot user data in ESX, default gateway config must be passed in `<NS-CONFIG>` section. For more information on the content of the `<NS-CONFIG>` tag, see Sample-`<NS-CONFIG>`-section.

Sample <NS-CONFIG> section:

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

How to provide preboot user data on ESX hypervisor

You can provide preboot user data on ESX hypervisor from web client or vSphere client in the following two ways:

- Using CD/DVD ISO
- Using OVF Property

Provide user data using CD/DVD ISO

You can use the VMware vSphere client to inject user data into the VM as an ISO image using the CD/DVD drive.

Follow these steps to provide user data using the CD/DVD ISO:

1. Create a file with file name `userdata` that has the preboot user data content. For more information on the content of the `<NS-CONFIG>` tag, see Sample `<NS-CONFIG>` section.

> **Note:**
>
> File name must be strictly used as `userdata`.

2. Store the `userdata` file in a folder, and build an ISO image using the folder.

   You can build an ISO image with `userdata` file by the following two methods:

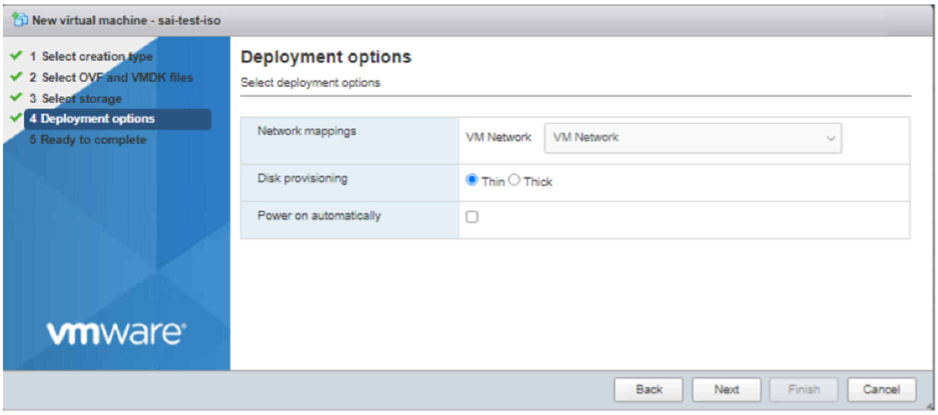
   - Using any image processing tool such as PowerISO.
   - Using `mkisofs` command in Linux.

   The following sample configuration shows how to generate an ISO image using the `mkisofs` command in Linux.

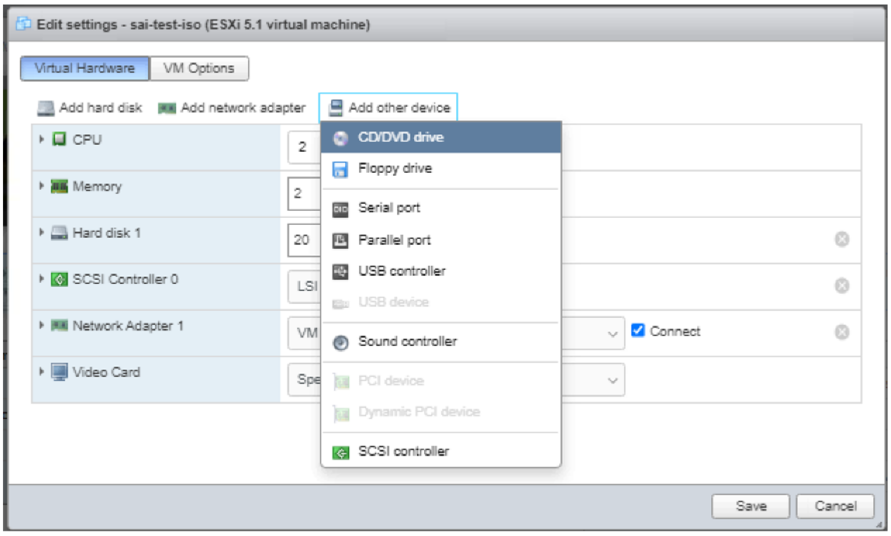
```
root@ubuntu:~/sai/14jul2021# ls -l total 4
drwxr-xr-x 2 root root 4096 Jul 14 12:32 esx_preboot_userdata
root@ubuntu:~/sai/14jul2021#
root@ubuntu:~/sai/14jul2021# ls -l esx_preboot_userdata/total 4
-rw-r--r-- 1 root root 3016 Jul 14 12:32 userdata
root@ubuntu:~/sai/14jul2021# mkisofs -o esx_preboot_userdata.iso
    ./esx_preboot_userdata
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)
root@ubuntu:~/sai/14jul2021# ls -lh
total 356K
drwxr-xr-x 2 root root 4.0K Jul 14 12:32 esx_preboot_userdata
-rw-r--r-- 1 root root 352K Jul 14 12:34 esx_preboot_userdata.iso

root@ubuntu:~/sai# ls preboot_userdata_155_193 userdata
root@ubuntu:~/sai# mkisofs -o preboot_userdata_155_193.iso ./
    preboot_userdata_155_193
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)
```

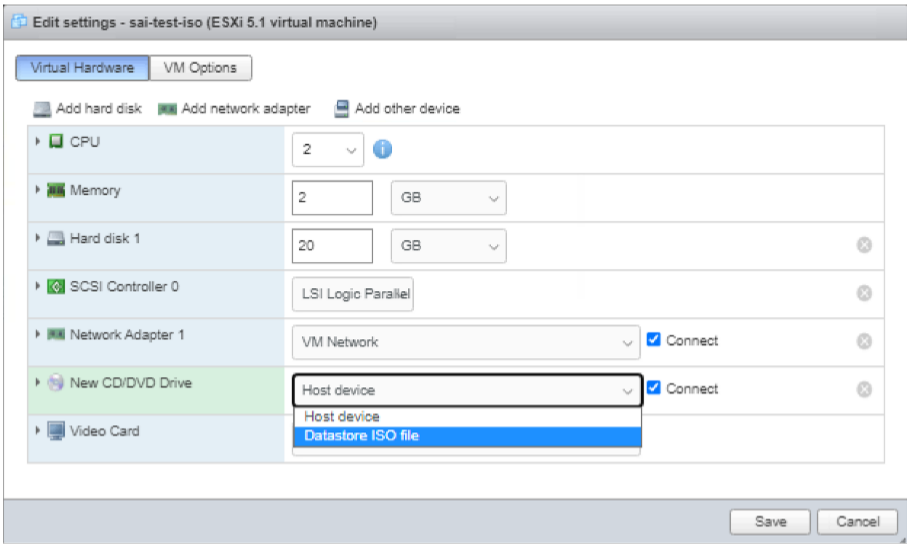
3. Provision the NetScaler VPX instance using standard deployment process to create the VM. But do not power on the VM automatically.



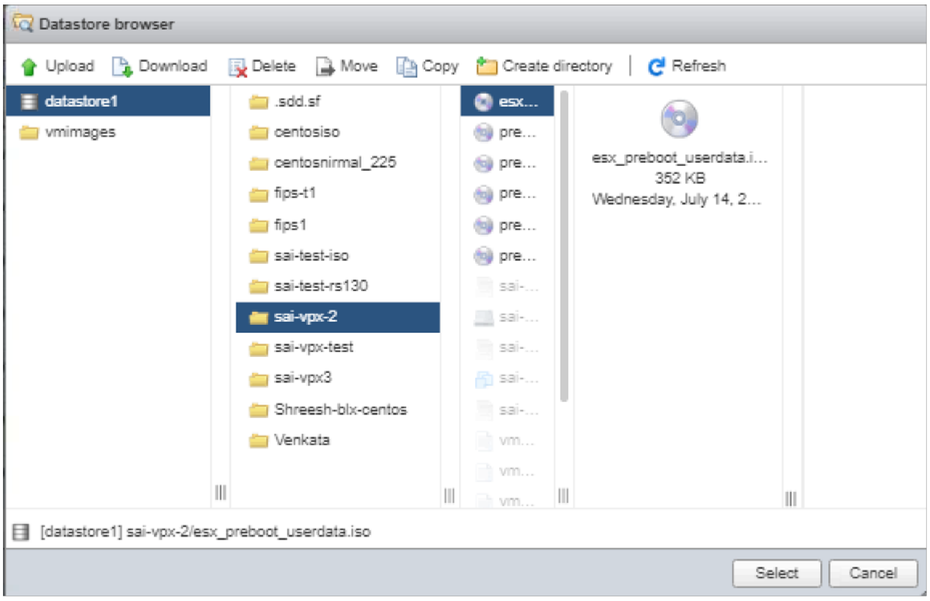
4. After the VM is successfully created, attach the ISO file as CD/DVD drive to the VM.



5. Navigate to **New CD/DVD Drive** and choose **Datastore ISO file** from the drop-down menu.



6. Select a Datastore in the vSphere Client.



7. Power on the VM.

Provide user data using the OVF property from the ESX web client

Follow these steps to provide user data using the OVF property.

1. Create a file with user data content.

```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

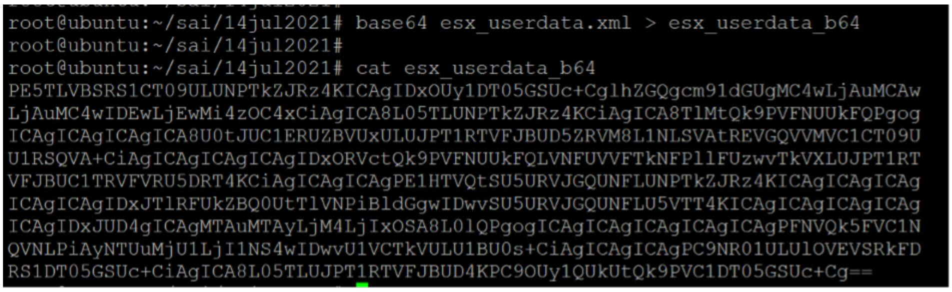
2. Encode the user data content with Base64 encoding. You can perform the Base64 encoding using the following two methods:

   - In Linux, use the following command:

```
base64 <userdata-filename>  >  <outuput-file>
```


Example:

```
base64 esx_userdata.xml  > esx_userdata_b64
```



- Use online tools to encode user data content, for example, Base64 Encode and Decode.

3. Include a **Product** section in the OVF template of a NetScaler VPX instance on ESX hypervisor.

   Sample Product section:

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>

  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
      userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
```

4. Provide the base64 encoded user data as the `ovf:value` for `guestinfo.userdata` property in the Product section.

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
      userConfigurable="true"
    ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
        CglhZGQgcm91dGUgMC4wLjAuMCAw
    LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQ
```

```
    ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C
    U1RSQVA+
        CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
    VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg
    ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAg
    ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgICAgPFNNVQk5F
    QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
        CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
    RS1DT05GSUc+
        CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
        =="&gt;

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
```

5. Use the modified OVF template with Product section for the VM deployment.

```
Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
 Done
> sh ns ver
        NetScaler NS13.0: Build 83.9005.nc, Date: Jul 13 2021, 02:56:05   (64-bit)
 Done
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  S
tate
        ---------        --------------  ----             ----     ---      ----     -------  -
-----
1)      10.102.38.219    0               NetScaler IP     Active   Enabled  Enabled  NA       E
nabled
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN   State   Traffic Domain  Type
        -------          -------          ---------------  ----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.102.38.1      0      UP      0              STATI
C
2)      127.0.0.0        255.0.0.0        127.0.0.1        0      UP      0              PERMA
NENT
3)      10.102.38.0      255.255.255.0    10.102.38.219    0      UP      0              DIREC
T
 Done
```

Provide user data using OVF property from the ESX vSphere client

Follow these steps to provide user data using the OVF property from the ESX vSphere client.

1. Create a file with user data content.

```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

2. Encode the user data content with Base64 encoding. You can perform the Base64 encoding using the following two methods:

    - In Linux, use the following command:

    ```
    base64 <userdata-filename>  >   <outuput-file>
    ```

    Example:

    ```
    base64 esx_userdata.xml  > esx_userdata_b64
    ```

    ```
    root@ubuntu:~/sai/14jul2021# base64 esx_userdata.xml > esx_userdata_b64
    root@ubuntu:~/sai/14jul2021#
    root@ubuntu:~/sai/14jul2021# cat esx_userdata_b64
    ```

    ```
    PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+CglhZGQgcm91dGUgMC4wLjAuMCAw
    LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQPgog
    ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1CT09U
    U1RSQVA+CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
    VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAgICAg
    ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAgICAg
    ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5FVC1N
    QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
    RS1DT05GSUc+CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg==
    ```


    - Use online tools to encode user data content, for example, Base64 Encode and Decode.

3. Include a **Product** section in the OVF template of a NetScaler VPX instance on ESX hypervisor.

    Sample Product section:

    ```
    <ProductSection>

      <Info>Information about the installed software</Info>
      <Product>NSVPX-VSK Template</Product>
      <Vendor>Citrix</Vendor>
      <VendorUrl>www.citrix.com</VendorUrl>
      <Category> Preboot Userdata </Category>

    ```



```
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
      userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
```

4. Provide the base64 encoded user data as the `ovf:value` for `guestinfo.userdata` property in the Product section.

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.Citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
      userConfigurable="true"
    ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
        CglhZGQgcm91dGUgMC4wLjAuMCAw
    LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQ

    ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C

    U1RSQVA+
        CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT

    VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg

    ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAg

    ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNNVQk5F

    QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
        CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
    RS1DT05GSUc+
        CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
        ==">

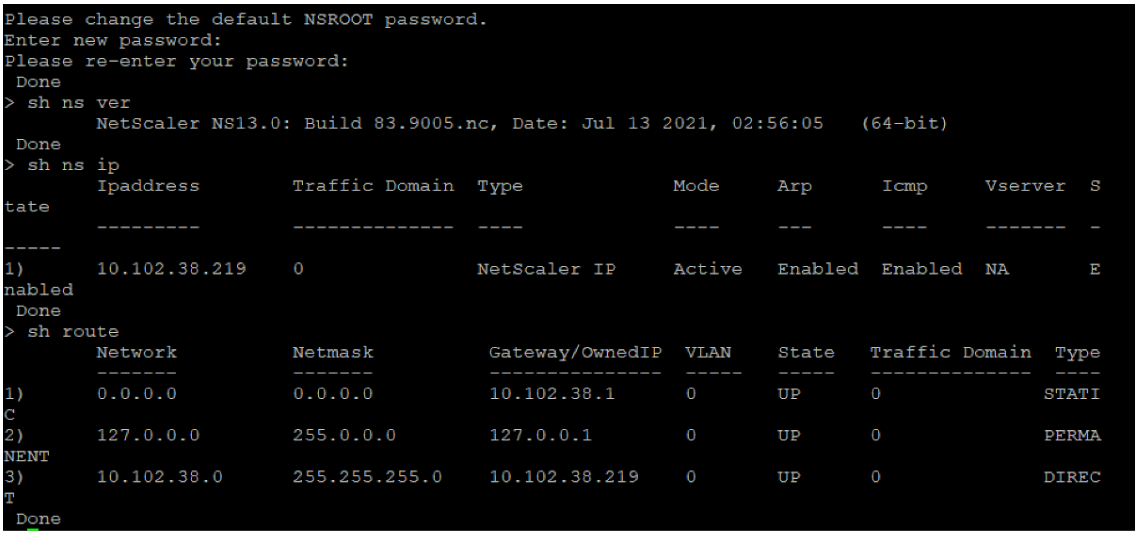
    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
```

5. Add the property `ovf:transport="com.vmware.guestInfo"` to VirtualHardwareSection as follows:

```
<VirtualHardwareSection  ovf:transport="com.vmware.guestInfo">
```


6. Use the modified OVF template with Product section for the VM deployment.



```
Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
 Done
> sh ns ver
        NetScaler NS13.0: Build 83.9005.nc, Date: Jul 13 2021, 02:56:05   (64-bit)
 Done
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  S
tate
        ---------        --------------  ----             ----     ---      ----     -------  -
-----
1)      10.102.38.219    0               NetScaler IP     Active   Enabled  Enabled  NA       E
nabled
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State    Traffic Domain  Type
        -------          -------          ---------------  -----   -----    --------------  ----
1)      0.0.0.0          0.0.0.0          10.102.38.1      0       UP       0              STATI
C
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP       0              PERMA
NENT
3)      10.102.38.0      255.255.255.0    10.102.38.219    0       UP       0              DIREC
T
 Done
```

Install a NetScaler VPX instance on VMware cloud on AWS

The VMware Cloud (VMC) on AWS enables you to create cloud software-defined data centers (SDDC) on AWS with the desired number of ESX hosts. The VMC on AWS supports NetScaler VPX deployments. VMC provides a user interface same as on-prem vCenter. It functions identical to the ESX-based NetScaler VPX deployments.

Prerequisites

Before you begin installing a virtual appliance, do the following:

- One VMware SDDC must be present with at least one host.
- Download the NetScaler VPX appliance setup files.
- Create appropriate network segments on VMware SDDC to which the virtual machines connect.
- Obtain VPX license files. For more information about NetScaler VPX instance licenses, see the *NetScaler VPX Licensing Guide* at </en-us/licensing/licensing-guide-for-netscaler>.

VMware cloud hardware requirements

The following table lists the virtual computing resources that the VMware SDDC must provide for each VPX nCore virtual appliance.

Table 1. Minimum virtual computing resources required for running a NetScaler VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	In VMware SDDC, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded to version 7 or higher.
Disk space	20 GB

> **Note:**
>
> This is in addition to any disk requirements for the hypervisor.

For production use of the VPX virtual appliance, the full memory allocation must be reserved.

OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. The following table describes the minimum system requirements.

Table 2. Minimum system requirements for OVF tool installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended
RAM	1 GB Minimum, 2 GB recommended
NIC	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/.

Downloading the NetScaler VPX setup files

The NetScaler VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log on. If you do not have a Citrix account, access the home page at http://www.citrix.com. Click the **New Users link**, and follow the instructions to create a new Citrix account.

Once logged on, navigate the following path from the Citrix home page:

Citrix.com > **Downloads > NetScaler > Virtual Appliances**.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-13.0-79.64.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-13.0-79.64.mf)

Install a NetScaler VPX instance on VMware cloud

After you have installed and configured VMware SDDC, you can use the SDDC to install virtual appliances on the VMware cloud. The number of virtual appliances that you can install depends on the amount of memory available on the SDDC.

To install NetScaler VPX instances on VMware cloud, follow these steps:

1. Open VMware SDDC on your workstation.
2. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click Login.
3. On the **File** menu, click **Deploy OVF Template**.
4. In the **Deploy OVF Template** dialog box, in **Deploy from file**, browse to the location at which you saved the NetScaler VPX instance setup files, select the .ovf file, and click **Next**.

   > **Note:**
   >
   > By default, the NetScaler VPX instance uses E1000 network interfaces. To deploy ADC with the VMXNET3 interface, modify the OVF to use VMXNET3 interface instead of E1000.

5. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the VMware SDDC. Click **Next** to start installing a virtual appliance on VMware SDDC.
6. You are now ready to start the NetScaler VPX instance. In the navigation pane, select the NetScaler VPX instance that you have installed and, from the right-click menu, select **Power On**. Click the **Console** tab to emulate a console port.
7. If you want to install another virtual appliance, repeat from Step 6.

8. Specify the management IP address from the same segment that is selected to be the management network. The same subnet is used for the Gateway.
9. The VMware SDDC requires that NAT and firewall rules are created explicitly for all private IP addresses belonging to network segments.

Install a NetScaler VPX instance on Microsoft Hyper-V server

To install NetScaler VPX instances on Microsoft Windows Server, you must first install Windows Server with the Hyper-V role enabled, on a machine with adequate system resources. While installing the Hyper-V role, be sure to specify the NICs on the server that Hyper-V uses to create the virtual networks. You can reserve some NICs for the host. Use Hyper-V Manager to perform the NetScaler VPX instance installation.

The NetScaler VPX instance for Hyper-V is delivered in virtual hard disk (VHD) format. It includes the default configuration for elements such as CPU, network interfaces, and hard-disk size and format. After you install NetScaler VPX instance, you can configure the network adapters on a virtual appliance, add virtual NICs, and then assign the NetScaler IP address, subnet mask, and gateway, and complete the basic configuration of the virtual appliance.

After the initial configuration of the VPX instance, if you want to upgrade the appliance to the latest software release, see Upgrade a NetScaler VPX standalone appliance

> **Note:**
>
> Intermediate System-to-Intermediate System (ISIS) protocol is not supported on the NetScaler VPX virtual appliance hosted on the HyperV-2012 platform.

Prerequisites for installing NetScaler VPX instance on Microsoft servers

Before you begin installing a virtual appliance, do the following:

- Enable the Hyper-V role on Windows Servers. For more information, see http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspx.
- Download the virtual appliance setup files.
- Get NetScaler VPX instance license files. For more information about NetScaler VPX instance licenses, see the *NetScaler VPX Licensing Guide* at https://support.citrix.com/s/article/CTX255959-how-to-allocate-and-install-citrix-netscaler-vpx-licenses?language=en_US.

Microsoft server hardware requirements

The following table describes the minimum system requirements for Microsoft servers.

Table 1. Minimum system requirements for Microsoft servers

Component	Requirement
CPU	1.4 GHz 64-bit processor
RAM	8 GB
Disk Space	32 GB or greater

The following table lists the virtual computing resources for each
NetScaler VPX instance.

Table 2. Minimum virtual computing resources required for running a NetScaler VPX instance

Component	Requirement
RAM	4 GB
Virtual CPU	2
Disk Space	20 GB
Virtual Network Interfaces	1

Download the NetScaler VPX setup files

The NetScaler VPX instance for Hyper-V is delivered in virtual hard disk (VHD) format. You can download the files from the Citrix website. You need a Citrix account to log in. If you do not have a Citrix account, access the home page at http://www.citrix.com, click **Sign In > My account > Create Citrix Account**, and follow the instructions to create a Citrix account.

To download the NetScaler VPX instance setup files, follow these steps:

1. In a web browser, go to http://www.citrix.com/.
2. Sign in with your user name and password.
3. Click **Downloads**.
4. In **Select a Product** drop-down menu, select **NetScaler (NetScaler ADC)**.
5. Under **NetScaler Release X.X > Virtual Appliances**, click **NetScaler VPX Release X.X**
6. Download the compressed file to your server.

Install the NetScaler VPX instance on Microsoft servers

After you have enabled the Hyper-V role on Microsoft Server and extracted the virtual appliance files, you can use Hyper-V Manager to install NetScaler VPX instance. After you import the virtual machine, you need to configure the virtual NICs by associating them to the virtual networks created by Hyper-V.

You can configure a maximum of eight virtual NICs. Even if the physical NIC is DOWN, the virtual appliance assumes that the virtual NIC is UP, because it can still communicate with the other virtual appliances on the same host (server).

> **Note:**
>
> You cannot change any settings while the virtual appliance is running. Shut down the virtual appliance and then make changes.

To install NetScaler VPX instance on Microsoft Server by using Hyper-V Manager:

1. To start Hyper-V Manager, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In the navigation pane, under **Hyper-V Manage**r, select the server on which you want to install NetScaler VPX instance.
3. On the **Action** menu, click **Import Virtual Machin**e.
4. In the **Import Virtual Machine** dialog box, in **Location**, specify the path of the folder that contains the NetScaler VPX instance software files, and then select **Copy the virtual machine (create a new unique ID)**. This folder is the parent folder that contains the Snapshots, Virtual Hard Disks, and Virtual Machines folders.

> **Note:**
>
> If you received a compressed file, make sure that you extract the files into a folder before you specify the path to the folder.

1. Click **Import**.
2. Verify that the virtual appliance that you imported is listed under **Virtual Machines**.
3. To install another virtual appliance, repeat steps **2** through **6**.

> **Important:**
>
> Make sure that you extract the files to a different folder in step **4**.

Auto-provision a NetScaler VPX instance on Hyper-V

Auto-provisioning of NetScaler VPX instance is optional. If auto-provisioning is not done, the virtual appliance provides an option to configure the IP address and so on.

To auto-provision NetScaler VPX instance on Hyper-V, follow these steps.

1. Create an ISO9660 compliant ISO image using the xml file as depicted in the example. Make sure that the name of the xml file is **userdata**.

   You can create an ISO file from XML file using:

   • Any image processing tool such as PowerISO.

   • `mkisofs` command in Linux.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe=`"http://schemas.dmtf.org/ovf/environment/1`"
xmlns:xsi=`"http://www.w3.org/2001/XMLSchema-instance`"
oe:id=""
xmlns=`"http://schemas.dmtf.org/ovf/environment/1`">
<PlatformSection>
<Kind>HYPER-V</Kind>
<Version>2013.1</Version>
<Vendor>CITRIX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0" />
<Property oe:key="com.citrix.netscaler.platform" oe:value="NS1000V"/>
<Property oe:key="com.citrix.netscaler.orch\_env" oe:value="cisco-orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.102.100.122"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="255.255.255.128"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.102.100.67"/></PropertySection>
```



```
</Environment>
```

2. Copy the ISO image to hyper-v server.
3. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**. You can also select the virtual appliance and then right click and select **Settings**. The **Settings** window for the selected virtual appliance is displayed.
4. In the **Settings** window, under the hardware section, click **IDE Controller**.
5. In the right window pane, select **DVD Drive** and click **Add**. The DVD Drive is added under the **IDE Controller** section in the left window pane.
6. Select the **DVD Drive** added in step 5. In the right window pane, select the **Image file radio** button and click **Browse** and select the ISO image that you copied on Hyper-V server, in step 2.
7. Click **Apply**.

> **Note:**
>
> The virtual appliance instance comes up in the default IP address, when:
>
> - The DVD drive is attached and the ISO file is not provided.
> - The ISO file does not include the user data file.
> - The user data file name or format is not correct.

To configure virtual NICs on the NetScaler VPX instance, follow these steps:

1. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**.
2. In the **Settings for <virtual appliance name>** dialog box, click **Add Hardware** in the left pane.
3. In the right pane, from the list of devices, select **Network Adapter**.
4. Click **Add**.
5. Verify that **Network Adapter (not connected)** appears in the left pane.
6. Select the network adapter in the left pane.
7. In the right pane, from the **Network** menu, select the virtual network to connect the adapter to.
8. To select the virtual network for other network adapters that you want to use, repeat steps **6** and **7**.
9. Click **Apply**, and then click **OK**.

To configure the NetScaler VPX instance:

1. Right-click the virtual appliance that you previously installed, and then select **Start**.
2. Access the console by double-clicking the virtual appliance.
3. Type the NetScaler IP address, subnet mask, and gateway for your virtual appliance.

You have completed the basic configuration of your virtual appliance. Type the IP address in a Web browser to access the virtual appliance.

> **Note:**
>
> You can also use virtual machine (VM) template to provision NetScaler VPX instance using SCVMM.
>
> If you use Microsoft Hyper-V NIC teaming solution with NetScaler VPX instances, see article CTX224494 for more information.

Install a NetScaler VPX instance on Linux-KVM platform

To set up a NetScaler VPX for the Linux-KVM platform, you can use the graphical Virtual Machine Manager (Virtual Manager) application. If you prefer the Linux-KVM command line, you can use the `virsh` program.

The host Linux operating system must be installed on suitable hardware by using virtualization tools such as KVM Module and QEMU. The number of virtual machines (VMs) that can be deployed on the hypervisor depends on the application requirement and the chosen hardware.

After you provision a NetScaler VPX instance, you can add more interfaces.

Limitations and usage guidelines

General recommendations

To avoid unpredictable behavior, apply the following recommendations:

- Do not change the MTU of the VNet interface associated with the VPX VM. Shut down the VPX VM before modifying any configuration parameters, such as Interface modes or CPU.
- Do not force a shutdown of the VPX VM. That is, do not use the **Force off** command.
- Any configurations done on the host Linux might or might not be persistent, depending on your Linux distribution settings. You can choose to make these configurations persistent to ensure consistent behavior across reboots of host Linux operating system.
- The NetScaler package has to be unique for each of the NetScaler VPX instance provisioned.

Limitations

- Live migration of a VPX instance that runs on KVM is not supported.

Prerequisites for installing a NetScaler VPX instance on Linux-KVM platform

Check the minimum system requirements for a Linux-KVM server running on a NetScaler VPX instance.

CPU requirement:

- 64-bit x86 processors with the hardware virtualization feature included in Intel VT-X processors.

To test whether your CPU supports the Linux host, enter the following command at the host Linux shell prompt:

```
*.egrep '^flags.*(vmx|svm)' /proc/cpuinfo*
```

If the **BIOS** settings for the preceding extension are disabled, you must enable them in the BIOS.

- Provide at least 2 CPU cores to Host Linux.
- There is no specific recommendation for processor speed, but higher the speed, the better the performance of the VM application.

Memory (RAM) requirement:

Minimum 4 GB for the host Linux kernel. Add more memory as required by the VMs.

Hard disk requirement:

Calculate the space for Host Linux kernel and VM requirements. A single NetScaler VPX VM requires 20 GB of disk space.

Software requirements

The Host kernel used must be a 64-bit Linux kernel, release 2.6.20 or later, with all virtualization tools. Citrix recommends newer kernels, such as 3.6.11-4 and later.

Many Linux distributions such as Red Hat, CentOS, and Fedora, have tested kernel versions and associated virtualization tools.

Guest VM hardware requirements

NetScaler VPX supports IDE and virtIO hard disk type. The Hard Disk Type has been configured in the XML file, which is a part of the NetScaler package.

Networking requirements

NetScaler VPX supports virtIO para-virtualized, SR-IOV, and PCI Passthrough network interfaces.

For more information about the supported network interfaces, see:

- Provision the NetScaler VPX instance by using the Virtual Machine Manager
- Configure a NetScaler VPX instance to use SR-IOV network interfaces
- Configure a NetScaler VPX instance to use PCI passthrough network interfaces

Source Interface and Modes

The source device type can be either Bridge or MacVTap. In MacVTap, four modes are possible - VEPA, Bridge, Private, and Pass-through. Check the types of interfaces that you can use and the supported traffic types, as per the following:

Bridge:

- Linux Bridge.
- `Ebtables` and `iptables` settings on host Linux might filter the traffic on the bridge if you do not choose the correct setting or disable `IPtable` services.

MacVTap (VEPA mode):

- Better performance than a bridge.
- Interfaces from the same lower device can be shared across the VMs.
- Inter-VM communication using the same
- lower device is possible only if the upstream or downstream switch supports VEPA mode.

MacVTap (private mode):

- Better performance than a bridge.
- Interfaces from the same lower device can be shared across the VMs.
- Inter-VM communication using the same lower device is not possible.

MacVTap (bridge mode):

- Better as compared to bridge.
- Interfaces out of the same lower device can be shared across the VMs.
- Inter-VM communication using the same lower device is possible, if the lower device link is UP.

MacVTap (Pass-through mode):

- Better as compared to bridge.
- Interfaces out of the same lower device cannot be shared across the VMs.
- Only one VM can use the lower device.

> **Note:**
>
> For best performance by the VPX instance, ensure that the `gro` and `lro` capabilities are switched off on the source interfaces.

Properties of source interfaces

Make sure that you switch off the generic-receive-offload (`gro`) and large-receive-offload (`lro`) capabilities of the source interfaces. To switch off the `gro` and `lro` capabilities, run the following commands at the host Linux shell prompt.

```
ethtool -K eth6 gro off
ethool -K eth6 lro off
```

Example:

```
[root@localhost ~]# ethtool -K eth6

                Offload parameters for eth6:

                                rx-checksumming: on

                                tx-checksumming: on

                scatter-gather: on

                tcp-segmentation-offload: on

                udp-fragmentation-offload: off

                generic-segmentation-offload: on

                generic-receive-offload: off

                large-receive-offload: off

                rx-vlan-offload: on

                tx-vlan-offload: on

                ntuple-filters: off

                receive-hashing: on

[root@localhost ~]#
```

Example:

If the host Linux bridge is used as a source device, as in the following example, and `lro` capabilities must be switched off on the VNet interfaces, which are the virtual interfaces connecting the host to the guest VMs.

```
[root@localhost ~]# brctl show eth6_br

bridge name     bridge  id              STP enabled interfaces

eth6_br         8000.00e0ed1861ae         no          eth6

                                                      vnet0

                                                      vnet2

[root@localhost ~]#
```

In the preceding example, the two virtual interfaces are derived from the eth6_br and are represented as vnet0 and vnet2. Run the following commands to switch off `gro` and `lro` capabilities on these interfaces.

```
ethtool -K vnet0 gro off
        ethtool -K vnet2 gro off
        ethtool -K vnet0 lro off
                 ethtool -K vnet2 lro off
```

Promiscuous mode

The promiscuous mode must be enabled for the following features to work:

- L2 mode
- Multicast traffic processing
- Broadcast
- IPV6 traffic
- virtual MAC
- Dynamic routing

Use the following command to enable the promiscuous mode.

```
[root@localhost ~]# ifconfig eth6 promisc
[root@localhost ~]# ifconfig eth6
eth6       Link encap:Ethernet  HWaddr 78:2b:cb:51:54:a3
            inet6 addr: fe80::7a2b:cbff:fe51:54a3/64 Scope:Link
            UP BROADCAST RUNNING PROMISC MULTICAST  MTU:9000  Metric:1
            RX packets:142961 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2895843 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:14330008 (14.3 MB)  TX bytes:1019416071 (1.0 GB)

[root@localhost ~]#
```


Module required

For better network performance, make sure the vhost_net module is present in the Linux host. To check the existence of vhost_net module, run the following command on the Linux host:

```
lsmod | grep "vhost\_net"
```

If vhost_net is not yet running, enter the following command to run it:

```
modprobe vhost\_net
```

Provision the NetScaler VPX instance by using OpenStack

You can provision a NetScaler VPX instance in an OpenStack environment either by using the **Nova boot** command (OpenStack CLI) or Horizon (OpenStack dashboard) .

Provisioning a VPX instance, optionally involves using data from the config drive. Config drive is a special configuration drive that attaches to the instance as a CD-ROM device when it boots. This configuration drive can be used to pass networking configuration such as management IP address, network mask, default gateway, and to inject customer scripts.

In a NetScaler appliance, the default authentication mechanism is password based. Now, the SSH key-pair authentication mechanism is supported for NetScaler VPX instances on the OpenStack environment.

The key-pair (public key and private key) is generated before using the Public Key Cryptography mechanism. You can use different mechanisms, such as Horizon, Puttygen.exe for Windows, and `ssh-keygen` for the Linux environment, to generate the key pair. Refer to online documentation of respective mechanisms for more information about generating key pair.

Once a key pair is available, copy the private key to a secure location to which authorized persons have access. In OpenStack, public key can be deployed on a VPX instance by using the Horizon or Nova boot command. When a VPX instance is provisioned by using OpenStack, it first detects that the instance is booting in an OpenStack environment by reading a specific BIOS string. This string is "OpenStack Foundation"and for Red Hat Linux distributions it is stored in /etc/nova/release. This is a standard mechanism that is available in all OpenStack implementations based on the KVM hypervisor platform. The drive must have a specific OpenStack label.

If the config drive is detected, the instance attempts to read the network configuration, custom scripts, and SSH key pair if provided.

User data file

The NetScaler VPX instance uses a customized OVF file, also known as the user data file, to inject network configuration, custom scripts. This file is provided as part of config drive. Here is an example of a customized OVF file.

````
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1"
xmlns:cs="http://schemas.citrix.com/openstack">
<PlatformSection>
<Kind></Kind>
<Version>2016.1</Version>
<Vendor>VPX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
<Property oe:key="com.citrix.netscaler.orch_env" oe:value="openstack-orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="255.255.255.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1"/>
</PropertySection>
 <cs:ScriptSection>
   <cs:Version>1.0</cs:Version>
     <ScriptSettingSection xmlns="http://schemas.citrix.com/openstack" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
         <Scripts>
           <Script>
                 <Type>shell</Type>
                 <Parameter>X Y</Parameter>
                <Parameter>Z</Parameter>
                <BootScript>before</BootScript>
                  <Text>
                        #!/bin/bash
                        echo "Hi, how are you" $1 $2 >> /var/sample.txt
                  </Text>
           </Script>
           <Script>
                 <Type>python</Type>
                 <BootScript>after</BootScript>
                  <Text>
                       #!/bin/python
 print("Hello");
                  </Text>
````
````

````
                </Script>
        <Script>
                        <Type>perl</Type>
                        <BootScript>before</BootScript>
                         <Text>
                              !/usr/bin/perl
    my $name = "VPX";
    print "Hello, World $name !\n" ;
                         </Text>
                </Script>
                <Script>
                       <Type>nscli</Type>
                       <BootScript>after</BootScript>
                       <Text>
                        add vlan 33
    bind vlan 33 -ifnum 1/2
                       </Text>
                </Script>
              </Scripts>
          </ScriptSettingSection>
    </cs:ScriptSection>
  </Environment>
  ```
````

In the OVF file preceding "PropertySection"is used for NetScaler networking configuration while <cs:ScriptSection> is used to enclose all scripts. <Scripts></Scripts> tags are used to bundle all scripts together. Each script is defined in between <Script> </Script> tags. Each script tag has following fields/tags:

a) <Type>: Specifies value for script type. Possible values: Shell/Perl/Python/NSLCI (for NetScaler CLI scripts)

b) <Parameter>: Provides parameters to the script. Each script can have multiple <Parameter> tags.

c) <BootScript>: Specifies script execution point. Possible values for this tag: before/after. "before" specifies script is run before PE comes up. "after"specifies that the script will be run after PE comes up.

d) <Text>: Pastes content of a script.

> **Note:**
>
> Currently the VPX instance does not take care of sanitization of scripts. As an administrator, you must check the validity of the script.
>
> Not all sections need to be present. Use an empty "PropertySection"to only define scripts to run on first boot or an empty <cs:ScriptSection> to only define networking configuration.
>
> After the required sections of the OVF file (user data file) are populated, use that file to provision the VPX instance.


````

### Network configuration

As part of the network configuration, the VPX instance reads:

- Management IP address
- Network mask
- Default gateway

After the parameters are successfully read, they are populated in the NetScaler configuration, to allow managing the instance remotely. If the parameters are not read successfully or the config drive is not available, the instance transitions to the default behavior, which is:

- The instance attempts to retrieve the IP address information from DHCP.
- If DHCP fails or times-out, the instance comes up with the default network configuration (192.168.100.1/16).

### Customer script

The VPX instance allows to run a custom script during initial provisioning. The appliance supports script of type Shell, Perl, Python, and NetScaler CLI commands.

### SSH key pair authentication

The VPX instance copies public key, available within the configuration drive as part of instance meta data, into its "authorized_keys"file. This allows the user to access the instance with private key.

> **Note:**
>
> When an SSH key is provided, the default credentials (nsroot/nsroot) no longer work, if password-based access is needed, log on with the respective SSH private key and manually set a password.

### Before you begin

Before you provision a VPX instance on OpenStack environment, extract the `.qcow2` file from the .tgz file and build

An OpenStack image from the qcow2 image. Follow these steps:

1. Extract the `.qcow2` file from the `.tqz` file by typing the following command

```
tar xvzf <TAR file>
tar xvzf <NSVPX-KVM-12.0-26.2_nc.tgz>
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```

2. Build an OpenStack image using the `.qcoz2` file extracted in step 1 by typing the following command.

```
openstack image create --container-format bare --property hw_disk_bus=ide --disk-format qcow2 --file <path to qcow2 file> --public <name of the OpenStack image>

glance image-create --name="NS-VPX-12-0-26-2" --property hw_disk_bus=ide --ispublic=
true --container-format=bare --disk-format=qcow2< NSVPX-KVM-12.0-26.2_nc.qcow2
```

**Figure 1**: The following illustration provides a sample output for the glance image-create command.

```
+------------------+----------------------------------------------------+
| Field            | Value                                              |
+------------------+----------------------------------------------------+
| checksum         | 154ade3fc7dca7d1706b1d03d7d97552                   |
| container_format | bare                                               |
| created_at       | 2017-03-13T08:52:31Z                               |
| disk_format      | qcow2                                              |
| file             | /v2/images/322c1e0f-cce8-4b7b-b53e-bd8152c388ed/file |
| id               | 322c1e0f-cce8-4b7b-b53e-bd8152c388ed               |
| min_disk         | 0                                                  |
| min_ram          | 0                                                  |
| name             | VPX-KVM-12.0-26.2
| owner            | 58d17d81df5d4406afbb4fdab3a58d79                   |
| properties       | hw_disk_bus='ide'                                  |
| protected        | False                                              |
| schema           | /v2/schemas/image                                  |
| size             | 784338944                                          |
| status           | active                                             |
| updated_at       | 2017-03-13T08:52:43Z                               |
| virtual_size     | None                                               |
| visibility       | public                                             |
+------------------+----------------------------------------------------+
```

### Provisioning the VPX instance

You can provision a VPX instance in two ways by using one of the options:

- Horizon (OpenStack dashboard)
- Nova boot command (OpenStack CLI)

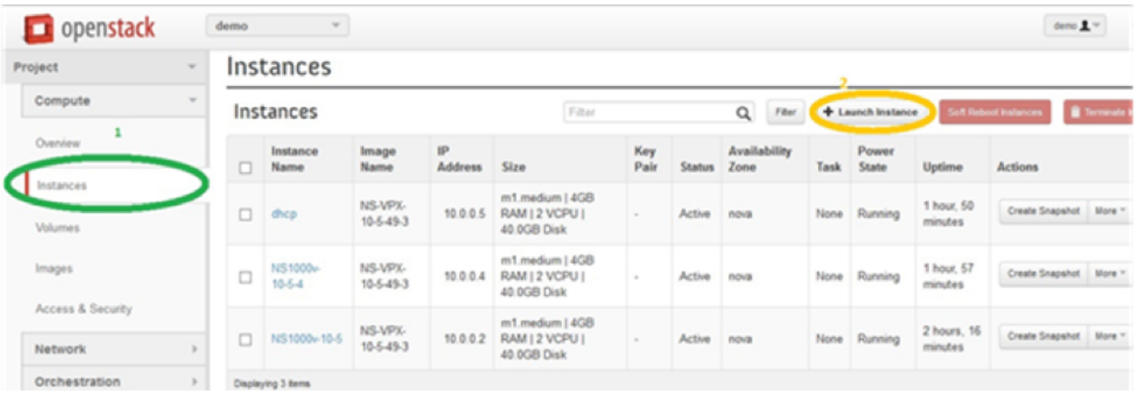
#### Provision a VPX instance by using the OpenStack dashboard

Follow these steps to provision the VPX instance by using Horizon:

1. Log on to the OpenStack dashboard.
2. In the Project panel on the left hand side of the dashboard, select **Instances**.

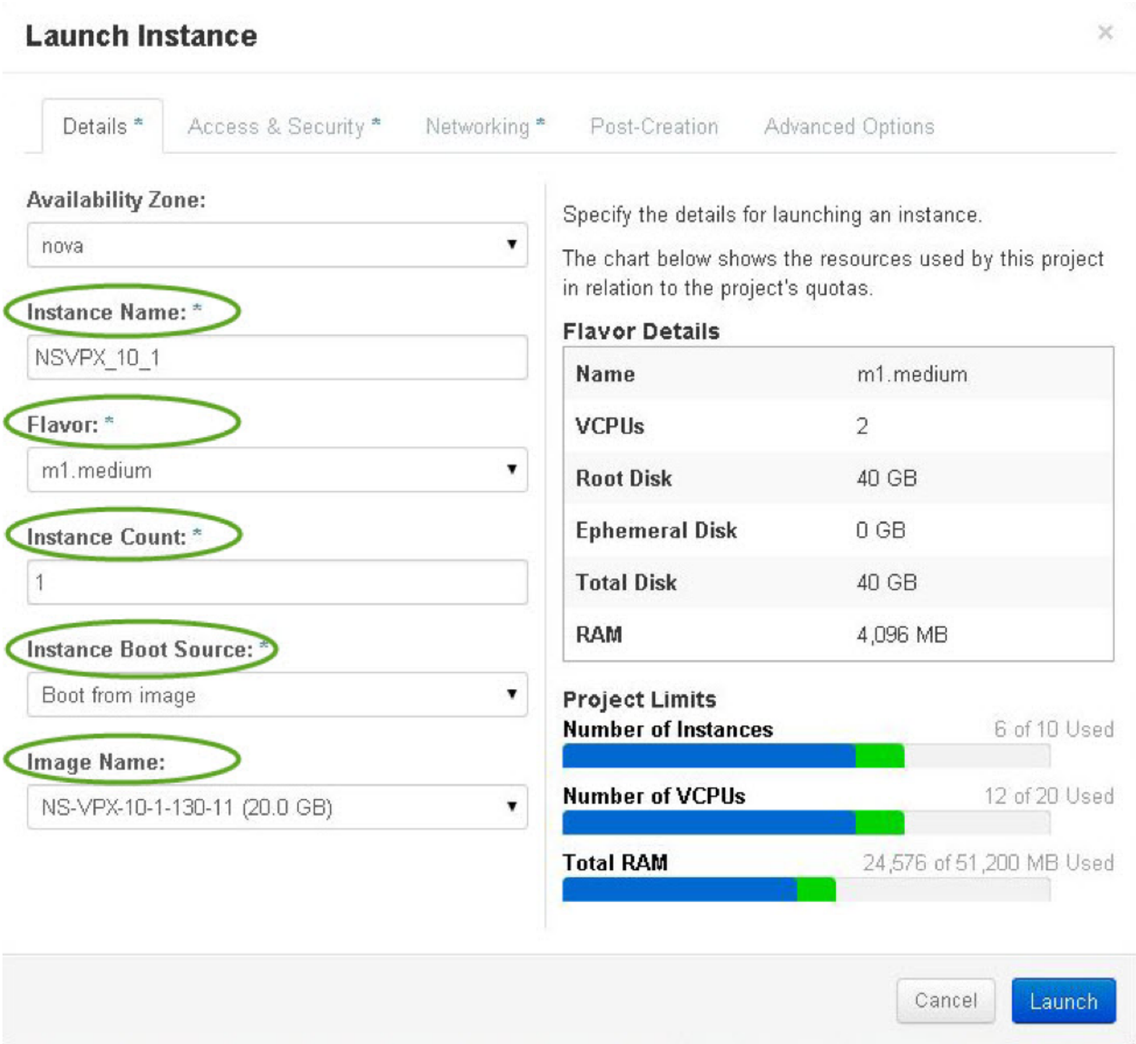
3. In the Instances panel, click **Launch Instance** to open the Instance Launching wizard.




4. In the Launch Instance wizard, fill in the details, like:

    a) Instance Name
    b) Instance Flavor
    c) Instance Count
    d) Instance Boot Source
    e) Image Name




5. Deploy a new key pair or an existing key pair through Horizon by completing the following steps:

   a) If you don't have an existing key pair, create the key by using any existing mechanisms. If you've an existing key, skip this step.

   b) Copy the content of public key.

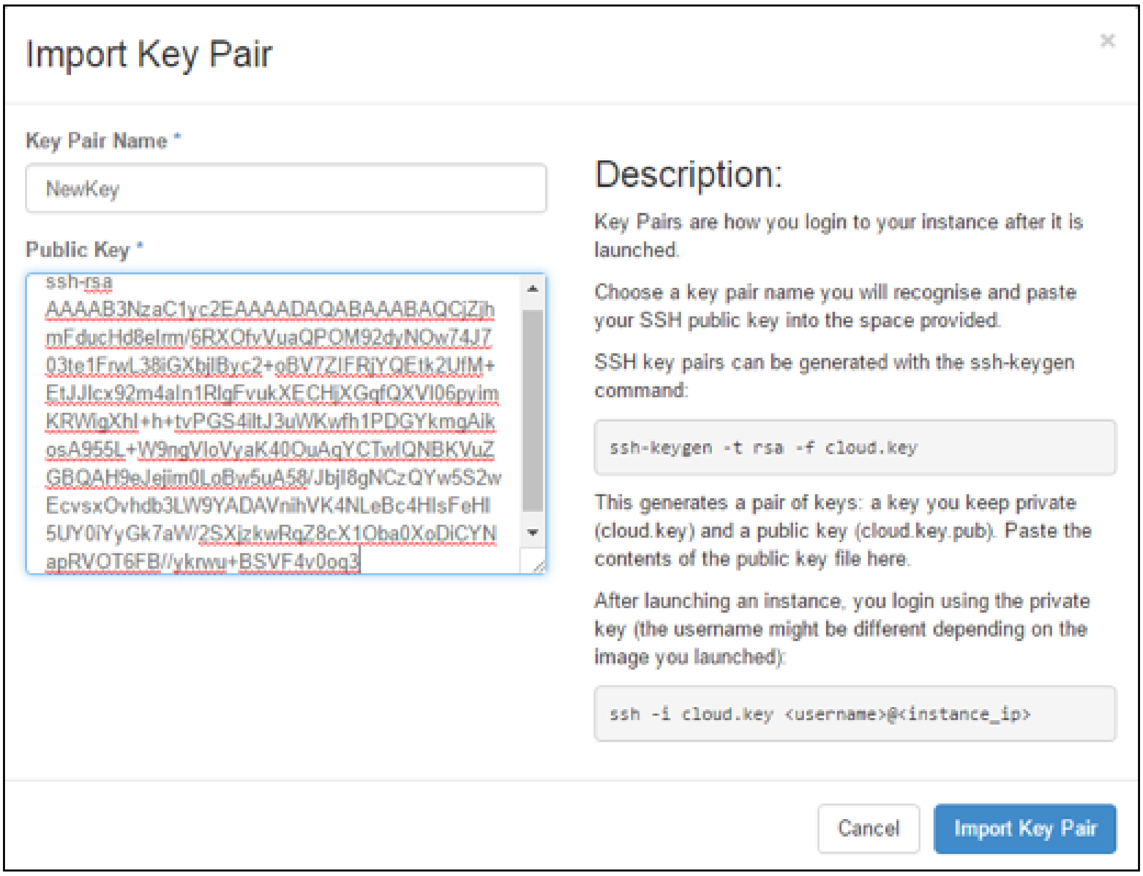
   c) Go to **Horizon > Instances > Create New Instances**.

   d) Click **Access & Security**.

   e) Click the + sign next to the **Key Pair** drop-down menu and provide values for shown parameters.

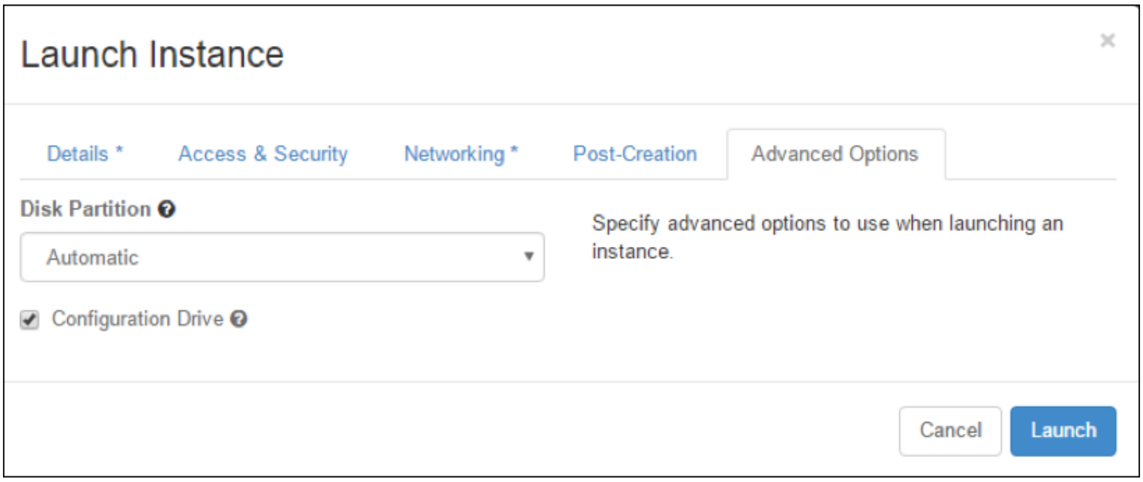
   f) Paste public key content in *Public key* box, give a name to the key and click **Import Key Pair**.




6. Click the **Post Creation** tab in the wizard. In Customization Script, add the content of the user data file. The user data file contains the IP address, Netmask and Gateway details, and customer scripts of the VPX instance.

7. After a key pair is selected or imported, check config-drive option and click **Launch**.



#### Provision the VPX instance by using OpenStack CLI

Follow these steps to provision a VPX instance by using OpenStack CLI.

1. To create an image from qcow2, type the following command:

   `openstack image create --container-format bare --property hw_disk_bus =ide --diskformat qcow2 --file NSVPX-OpenStack.qcow2 --public VPX -ToT-Image`

2. To select an image for creating an instance, type the following command:

   `openstack image list | more`

3. To create an instance of a particular flavor, type the following command to choose a flavor ID/-Name of from a list:

   `openstack flavor list`

4. To attach a NIC to a particular network, type the following command to choose a network ID from a network list:

   `openstack network list`

5. To create an instance, type the following command:

   ```
   openstack server create --flavor FLAVOR_ID --image IMAGE_ID --key-
       name KEY_NAME
   --user-data USER_DATA_FILE_PATH --config-drive True --nic net-id=
       net-uuid
   INSTANCE_NAME
   openstack server create --image VPX-ToT-Image --flavor m1.medium
       --user-data
   ovf.xml --config-drive True --nic net-id=2734911b-ee2b-48d0-a1b6-3
       efd44b761b9
   VPX-ToT
   ```

**Figure 2**: The following illustration provides a sample output.

```
+--------------------------------------+--------------------------------------------------------------------+
| Field                                | Value                                                              |
+--------------------------------------+--------------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                             |
| OS-EXT-AZ:availability_zone          |                                                                    |
| OS-EXT-SRV-ATTR:host                 | None                                                               |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | None                                                               |
| OS-EXT-SRV-ATTR:instance_name        | instance-000001c2                                                  |
| OS-EXT-STS:power_state               | 0                                                                  |
| OS-EXT-STS:task_state                | scheduling                                                         |
| OS-EXT-STS:vm_state                  | building                                                           |
| OS-SRV-USG:launched_at               | None                                                               |
| OS-SRV-USG:terminated_at             | None                                                               |
| accessIPv4                           |                                                                    |
| accessIPv6                           |                                                                    |
| addresses                            |                                                                    |
| adminPass                            | pFVvMtq7N8Z6                                                       |
| config_drive                         | True                                                               |
| created                              | 2017-03-13T10:32:59Z                                               |
| flavor                               | m1.medium (3)                                                      |
| hostId                               |                                                                    |
| id                                   | a1fe991e-3604-43a0-9dd6-59fa0f3749df                               |
| image                                | VPX-ToT-Image (f0c2f9d1-08f2-4b2e-9943-2ee6bc2edbc7)               |
| key_name                             | None                                                               |
| name                                 |VPX-ToT                                                             |
| os-extended-volumes:volumes_attached | []                                                                 |
| progress                             | 0                                                                  |
| project_id                           | 58d17d81df5d4406afbb4fdab3a58d79                                   |
| properties                           |                                                                    |
| security_groups                      | [{u'name': u'default'}]                                            |
| status                               | BUILD                                                              |
| updated                              | 2017-03-13T10:33:00Z                                               |
| user_id                              | a6347b33916b4eb1b1f76360a9c8f935                                   |
+--------------------------------------+--------------------------------------------------------------------+
```

## Provision the NetScaler VPX instance by using the Virtual Machine Manager

The Virtual Machine Manager is a desktop tool for managing VM guests. It enables you to create new VM guests and various types of storage, and manage virtual networks. You can access the graphical console of VM guests with the built-in VNC viewer and view performance statistics, either locally or remotely.

After installing your preferred Linux distribution, with KVM virtualization enabled, you can proceed with provisioning virtual machines.

While using the Virtual Machine Manager to provision a NetScaler VPX instance, you have two options:

- Enter the IP address, gateway, and netmask manually
- Assign the IP address, gateway, and netmask automatically (auto-provisioning)

You can use two kinds of images to provision a NetScaler VPX instance:

- RAW
- QCOW2

You can convert a NetScaler VPX RAW image to a QCOW2 image and provision the NetScaler VPX instance. To convert the RAW image to a QCOW2 image, type the following command:

```
qemu-img convert -O qcow2 original-image.raw image-converted.qcow2
```

**For example:**

```
qemu-img convert -O qcow2 NSVPX-KVM-11.1-12.5_nc.raw NSVPX-KVM-11.1-12.5
_nc.qcow2
```

A typical NetScaler VPX deployment on KVM includes the following steps:

- Checking prerequisites for auto-provisioning a NetScaler VPX instance
- Provisioning the NetScaler VPX instance by using a RAW image
- Provisioning the NetScaler VPX instance by using a QCOW2 image
- Adding more interfaces to a VPX instance by using a virtual machine manager

### Check prerequisites for auto-provisioning a NetScaler VPX instance

Auto-provisioning is an optional feature, and it involves using data from the CDROM drive. If this feature is enabled, you need not enter the management IP address, network mask, and default gateway of the NetScaler VPX instance during initial setup.

You need to complete the following tasks before you can auto-provision a VPX instance:

1. Create a customized Open Virtualization Format (OVF) XML file or user data file.
2. Convert the OVF file into an ISO image by using an online application (for example PowerISO).
3. Mount the ISO image on the KVM host by using any secure copy (SCP)-based tools.

**Sample OVF XML file**:

Here's is an example of the contents an OVF XML file, which you can use as a sample to create your file.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Environment xmlns:oe="`http://schemas.dmtf.org/ovf/environment/1"`

xmlns:xsi="`http://www.w3.org/2001/XMLSchema-instance"`

oe:id=""

xmlns="`http://schemas.dmtf.org/ovf/environment/1"`

xmlns:cs="`http://schemas.citrix.com/openstack">`

<PlatformSection>

<Kind></Kind>

<Version>2016.1</Version>

```

```
<Vendor>VPX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
<Property oe:key="com.citrix.netscaler.orch\_env" oe:value="KVM"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="255.255.255.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1"/>
</PropertySection>
</Environment>
```

In the OVF XML file preceding, "PropertySection"is used for NetScaler networking configuration. When you create the file, specify values for the parameters that are highlighted at the end of the example:

- Management IP address
- Netmask
- Gateway

> **Important:**
>
> If the OVF file is not properly XML formatted, the VPX instance is assigned the default network configuration, not the values specified in the file.

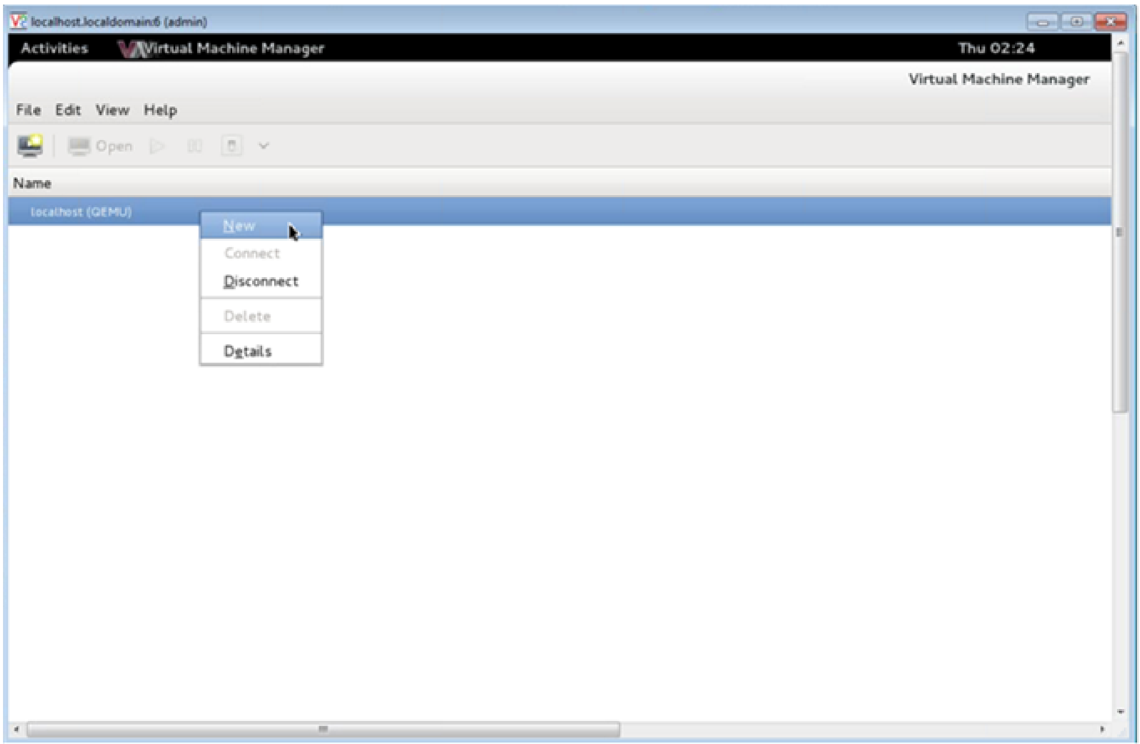
### Provision the NetScaler VPX instance by using a RAW image

The Virtual Machine Manager enables you to provision a NetScaler VPX instancy by using a RAW image.

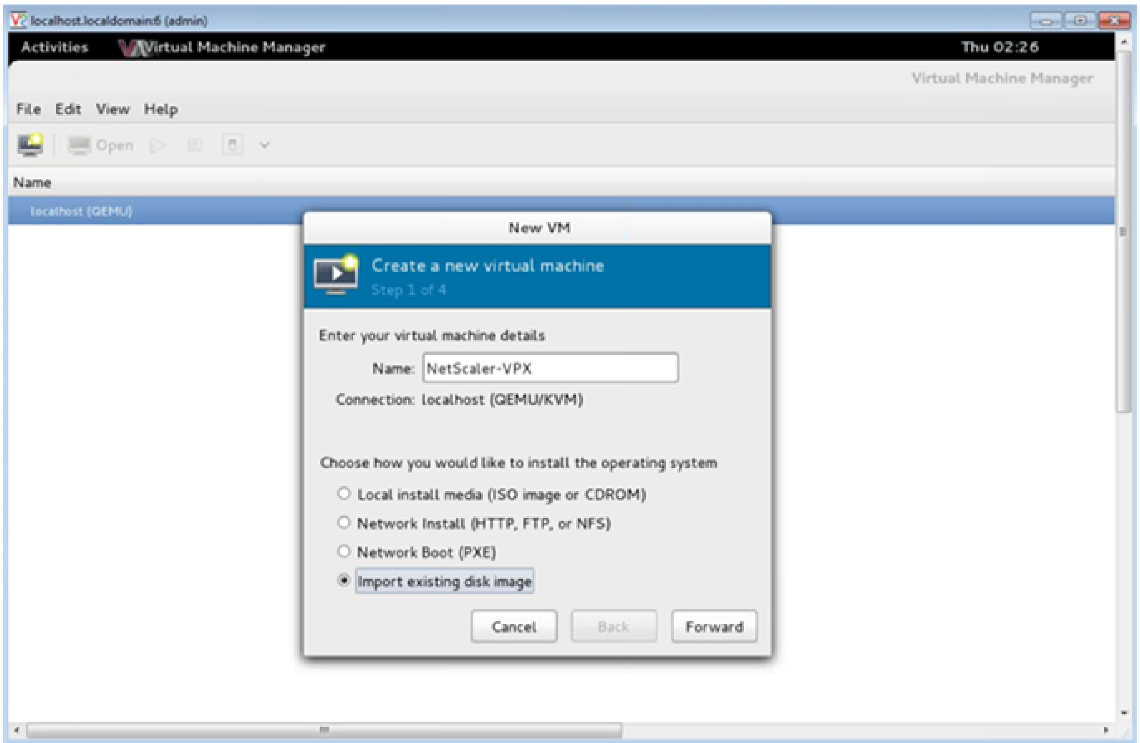
To provision a NetScaler VPX instance by using the Virtual Machine Manager, follow these steps:

1. Open the Virtual Machine Manager (**Application** > **System Tools** > **Virtual Machine Manager**) and enter the logon credentials in the **Authenticate** window.
2. Click the  icon or right-click **localhost (QEMU)** to create a new NetScaler VPX instance.

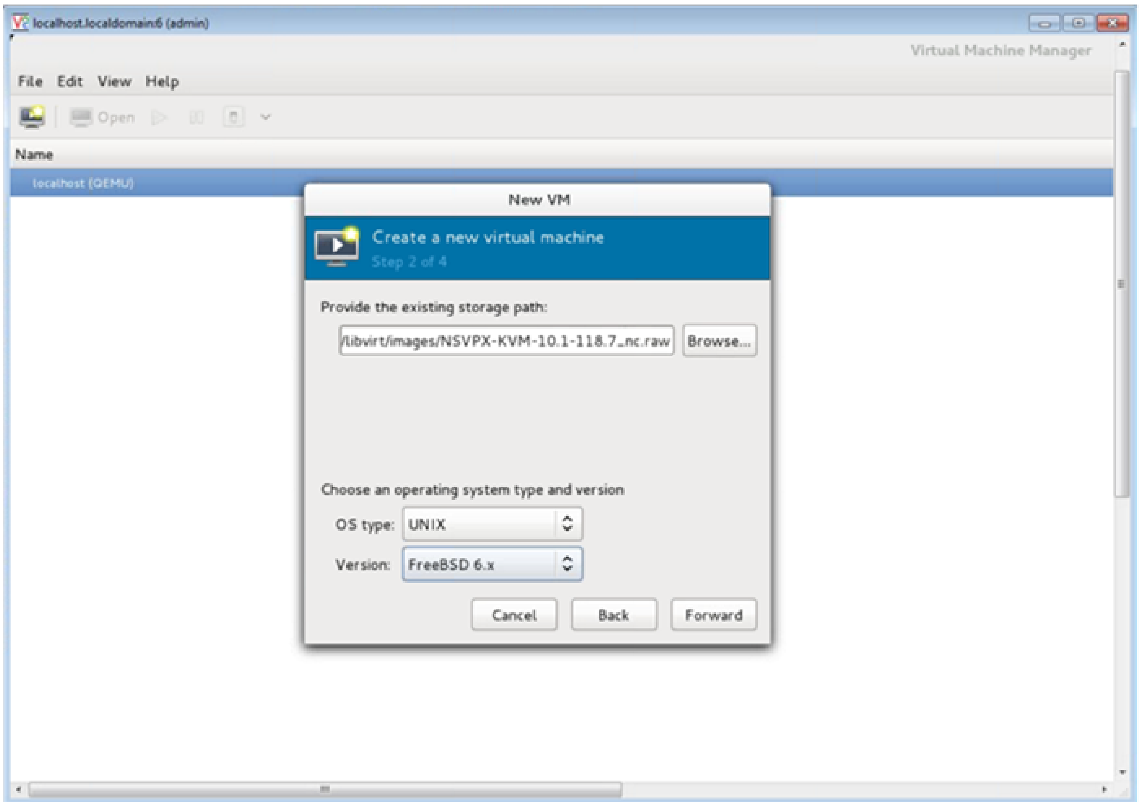
 




3. In the **Name** text box, enter a name for the new VM (for example, NetScaler-VPX).
4. In the **New VM** window, under "Choose how you would like to install the operating system," select **Import existing disk image**, and then and click **Forward**.



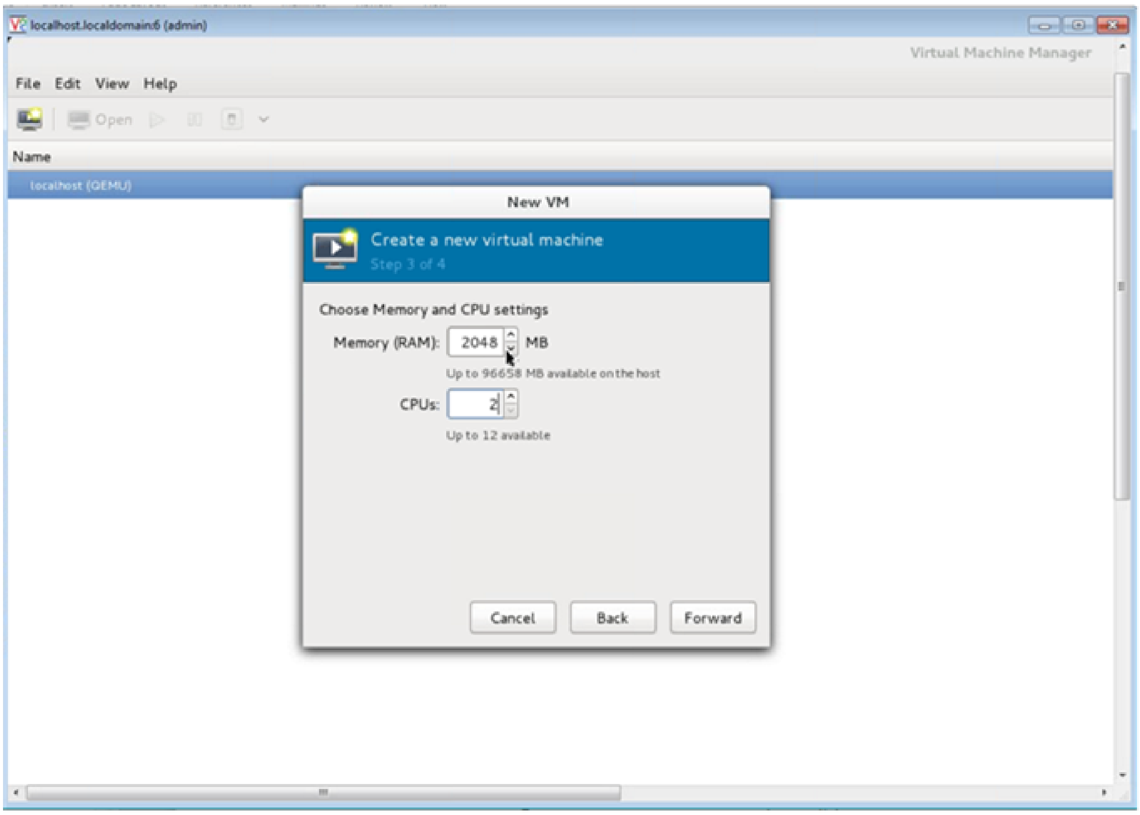
5. In the **Provide the existing storage path** field, navigate the path to the image. Choose the OS type as UNIX and Version as FreeBSD 6.x. Then, click **Forward**.




6. Under **Choose Memory and CPU** settings select the following settings, and then click **Forward**:

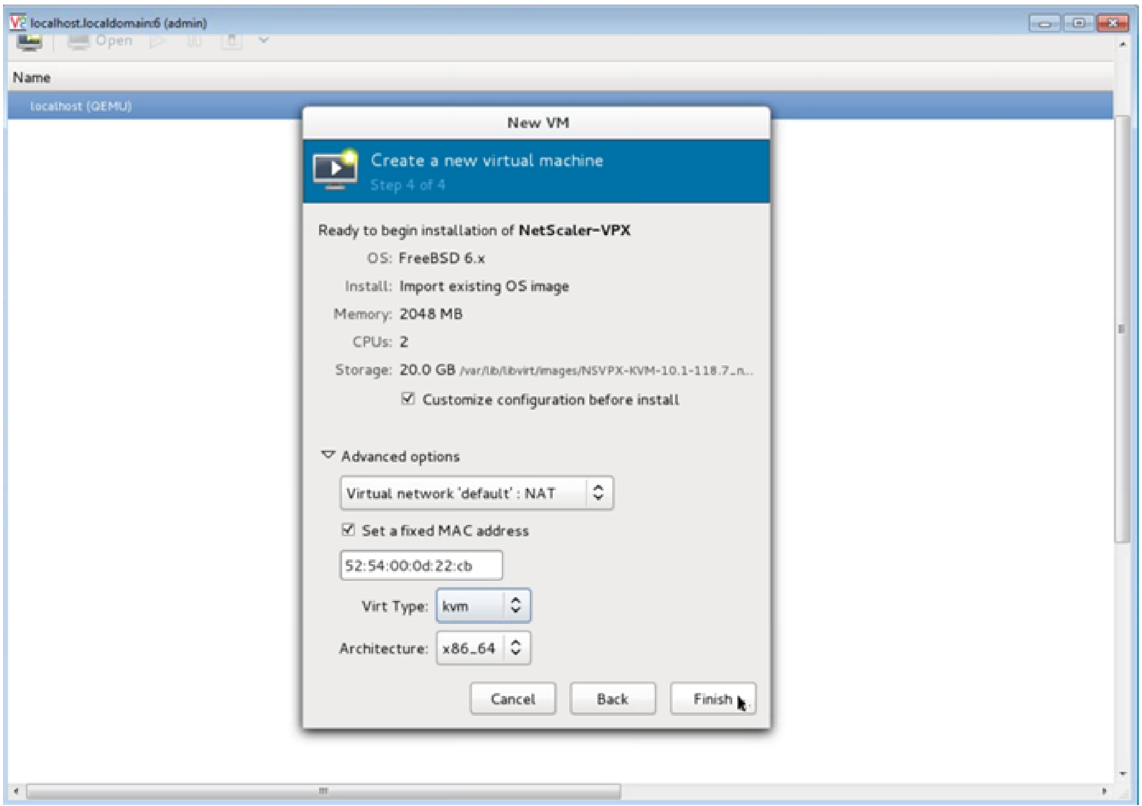
   - Memory (RAM)—2048 MB
   - CPUs—2




7. Select the **Customize configuration before install** check box. Optionally, under **Advanced options** you can customize the MAC address. Make sure the **Virt Type** selected is KVM and the Architecture selected is x86_64. Click **Finish**.

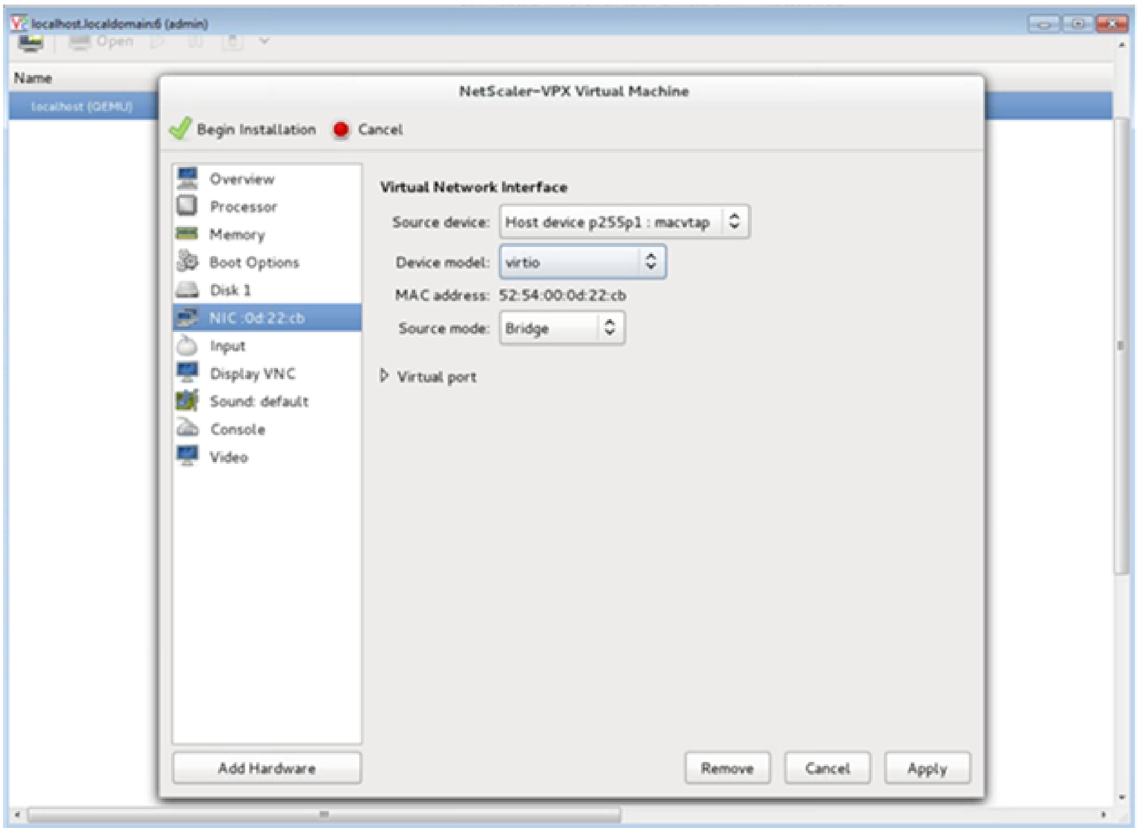
 




8. Select a NIC and provide the following configuration:

   - Source device—`ethX macvtap` or Bridge
   - Device model—`virtio`
   - Source mode—Bridge




9. Click **Apply**.
10. If you want to auto-provision the VPX instance, see the section **Enabling Auto-Provisioning by Attaching a CDROM Drive** in this document. Otherwise, click **Begin Installation**. After you have provisioned the NetScaler VPX on KVM, you can add more interfaces.

### Provision the NetScaler VPX instance by using a QCOW2 image

Using the Virtual Machine Manager, you can provision the NetScaler VPX instance by using a QCOW2 image.

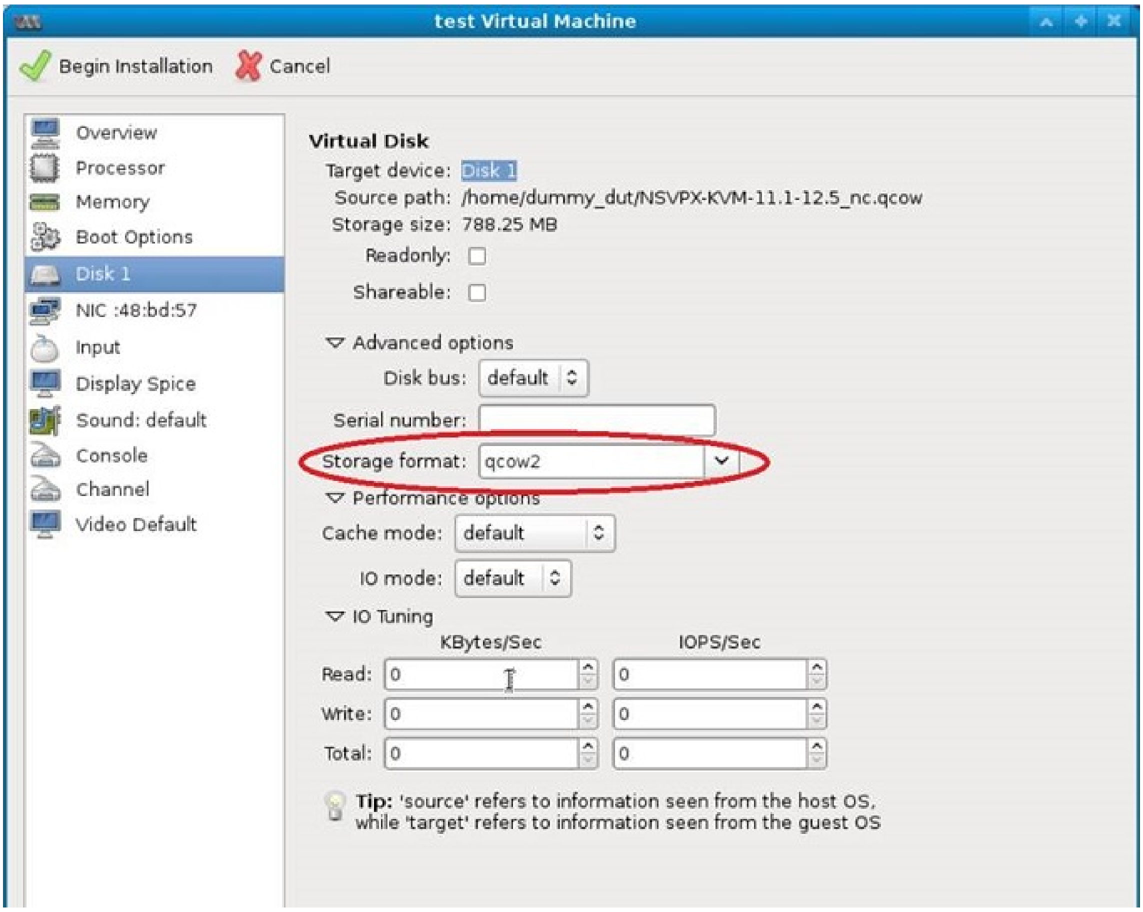
To provision a NetScaler VPX instance by using a QCOW2 image, follow these steps:

1. Follow **step 1** to **step 8** in Provision the NetScaler VPX instance by using a RAW image.

   > **Note:**
   >
   > Ensure that you select **qcow2** image in **step 5**.

2. Select **Disk 1** and click **Advanced options**.
3. Select **qcow2** from the Storage format drop-down list.

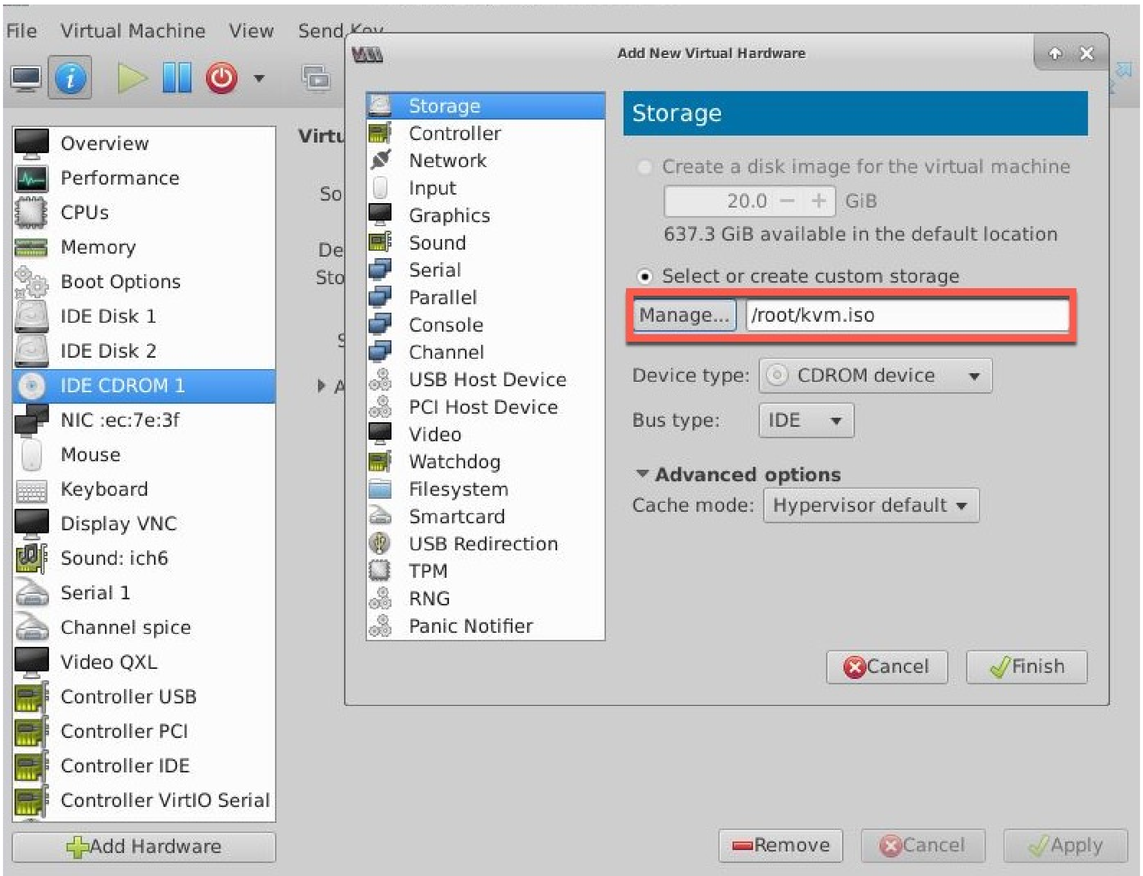



4. Click **Apply**, and then click **Begin Installation**. After you have provisioned the NetScaler VPX on KVM, you can add more interfaces.

## Enable auto-provisioning by attaching a CDROM drive

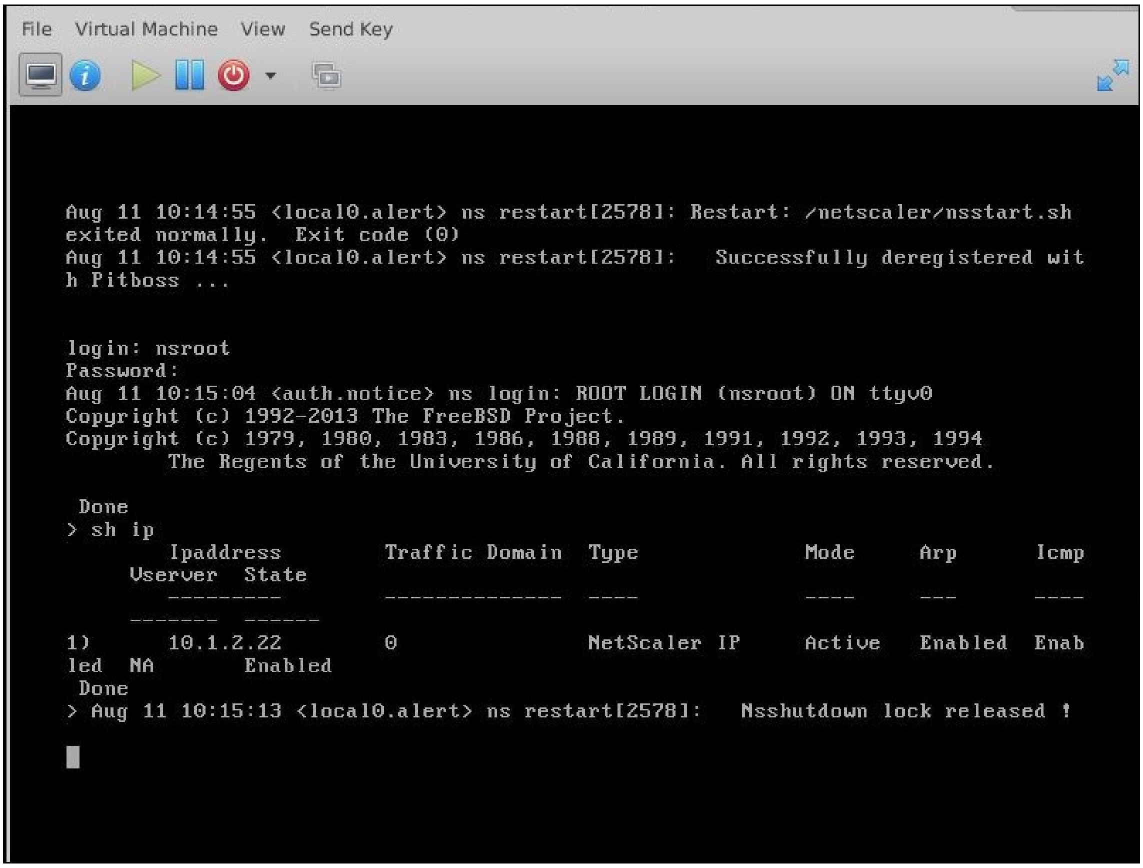
1. Click Add **Hardware > Storage > Device type > CDROM device**.
2. Click **Manage** and select the correct ISO file that you mounted in the "Prerequisites for Auto-Provisioning a NetScaler VPX Instance" section, and click **Finish**. A new CDROM under Resources on your NetScaler VPX instance is created.




3. Power on the VPX instance, and it auto-provisions with the network configuration provided in the OVF file, as shown in the example screen capture.

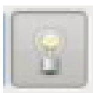



4. If auto-provision fails, the instance comes up with the default IP address (192.168.100.1). In that case, you must complete the initial configuration manually. For more information, see Configure the ADC for the first time.

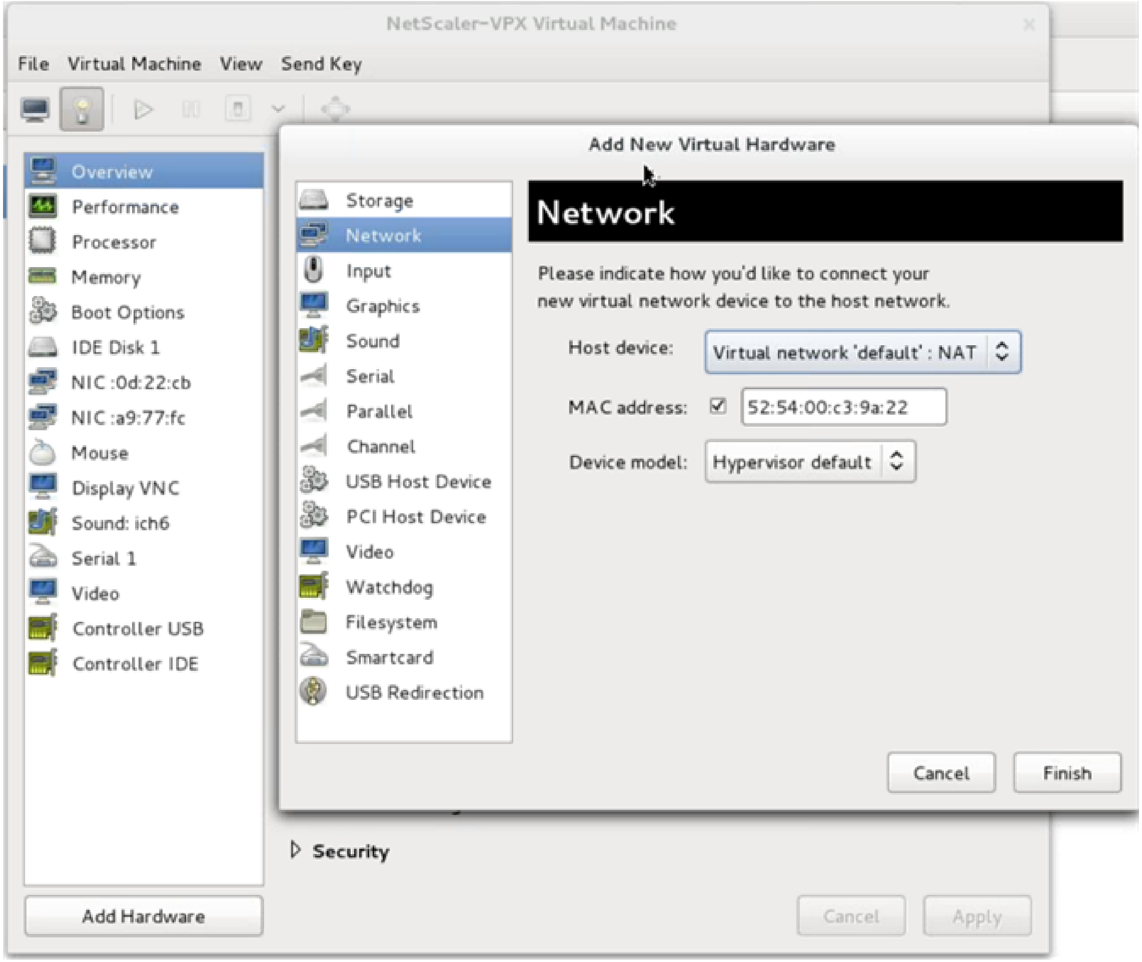
## Add more interfaces to the NetScaler VPX instance by using the Virtual Machine Manager

After you have provisioned the NetScaler VPX instance on KVM, you can add more interfaces.

To add more interfaces, follow these steps.

1. Shut down the NetScaler VPX instance running on the KVM.
2. Right-click the VPX instance and choose **Open** from the pop-up menu.
3. Click the  icon in the header to view the virtual hardware details.
4. Click **Add Hardware**. In the **Add New Virtual Hardware window**, select **Network** from the navigation menu.




5. In **Host Device** field, select the physical interface type. The host device type can be either Bridge or MacVTap. In case of MacVTap, four modes possible are VEPA, Bridge, Private, and Pass-through.

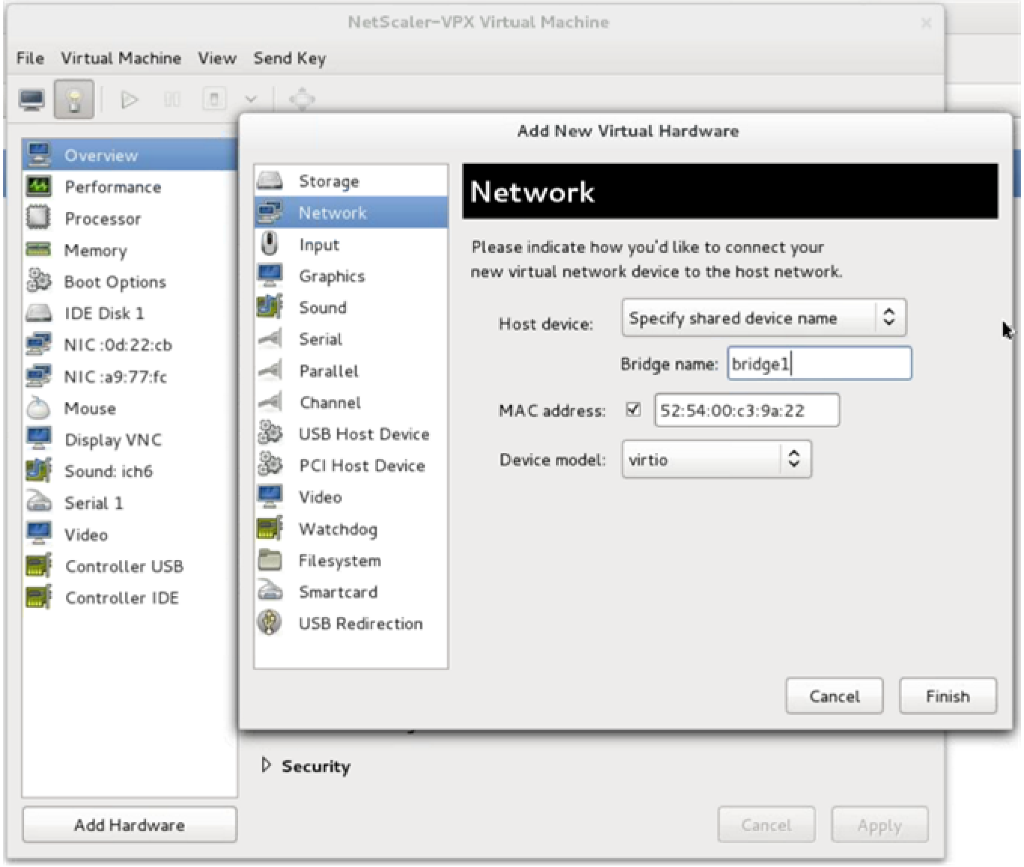
   a) For Bridge

      i. Host device—Select the "Specify shared device name"option.

      ii. Provide the Bridge name that is configured in the KVM host.

> **Note:**
>
> Make sure that you have configured a Linux bridge in the KVM host, bound the physical interface to the bridge, and put the bridge in the UP state.




iii. Device model—`virtio`.

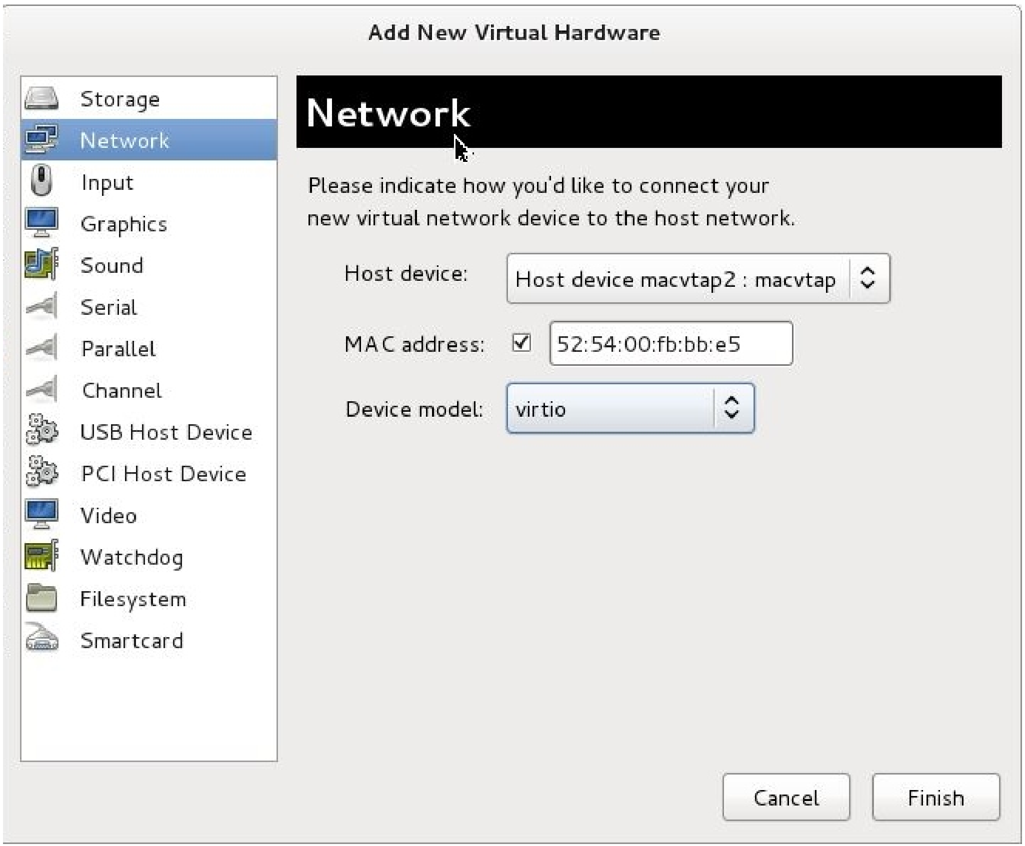
iv. Click **Finish**.

b) For MacVTap

i. Host device—Select the physical interface from the menu.

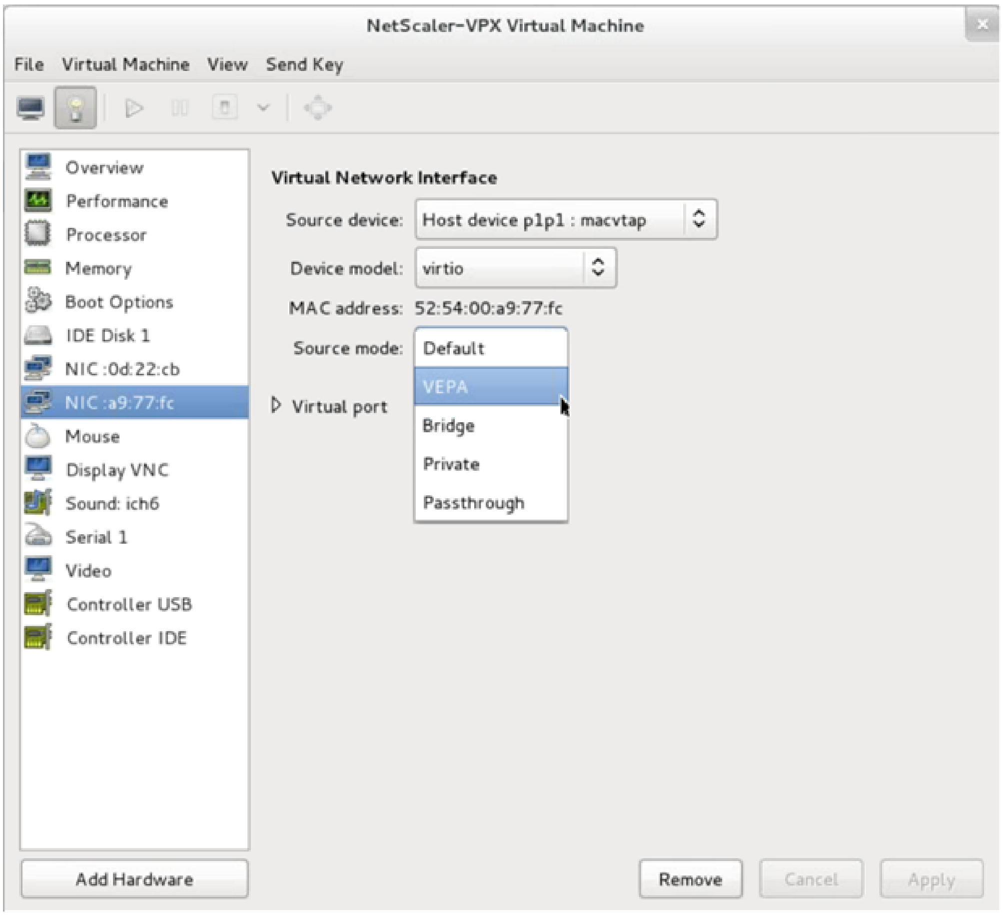
ii. Device model—`virtio`.




iii. Click **Finish**. You can view the newly added NIC in the navigation pane.




iv. Select the newly added NIC and select the Source mode for this NIC. The available modes are VEPA, Bridge, Private, and Passthrough. For more details on the interface and modes, see Source Interface and Modes.

v. Click **Apply**.

6. If you want to auto-provision the VPX instance, see the section "Adding a Config Drive to Enable Auto-Provisioning"in this document. Otherwise, power on the VPX instance to complete the initial configuration manually.

> **Important:**
>
> Interface parameter configurations such as speed, duplex, and autonegotiation are not supported.

## Configure a NetScaler VPX instance to use SR-IOV network interfaces

You can configure a NetScaler VPX instance running on Linux-KVM platform using single root I/O virtualization (SR-IOV) with the following NICs:

- Intel 82599 10G
- Intel X710 10G
- Intel XL710 40G
- Intel X722 10G

This section describes how to:

- Configure a NetScaler VPX Instance to Use SR-IOV Network Interface
- Configure Static LA/LACP on the SR-IOV Interface
- Configure VLAN on the SR-IOV Interface

## Limitations

Keep the limitations in mind while using Intel 82599, X710, XL710, and X722 NICs. The following features not supported.

**Limitations for Intel 82599 NIC**:

- L2 mode switching.
- Admin partitioning (shared VLAN mode).
- High availability (active-active mode).
- Jumbo frames.
- IPv6: You can configure only up to 30 unique IPv6 addresses in a VPX instance if you've at least one SR-IOV interface.
- VLAN configuration on Hypervisor for SRIOV VF interface through `ip link` command is not supported.
- Interface parameter configurations such as speed, duplex, and autonegotiations are not supported.

**Limitations for Intel X710 10G, Intel XL710 40G, and Intel X722 10G NICs**:

- L2 mode switching.
- Admin partitioning (shared VLAN mode).
- In a cluster, Jumbo frames are not supported when the XL710 NIC is used as a data interface.
- Interface list reorders when interfaces are disconnected and reconnected.
- Interface parameter configurations such as speed, duplex, and auto negotiations are not supported.
- Interface name is 40/X for Intel X710 10G, Intel XL710 40G, and Intel X722 10G NICs
- Up to 16 Intel XL710/X710/X722 SRIOV or PCI passthrough interfaces can be supported on a VPX instance.

> **Note:**
>
> For Intel X710 10G, Intel XL710 40G, and Intel X722 10G NICs to support IPv6, you need to enable trust mode on the Virtual Functions (VFs) by typing the following command on the KVM host:
>
> ```
> # ip link set <PNIC> <VF> trust on
> ```
>
> Example:
>
> ```
> # ip link set ens785f1 vf 0 trust on
> ```

### Prerequisites

Before you configure a NetScaler VPX instance to use SR-IOV network interfaces, complete the following prerequisite tasks. See the NIC column for details about how to complete the corresponding tasks.

| Task | Intel 82599 NIC | Intel X710, XL710, and X722 NICs |
|---|---|---|
| 1. Add the NIC to the KVM host. | - | - |
| 1. Download and install the latest Intel driver. | IXGBE driver | I40E driver |
| 1. Block list the driver on the KVM host. | Add the following entry in the /etc/modprobe.d/blacklist.conf file: `blacklist ixgbevf`. Use IXGBE driver version 4.3.15 (recommended). | Add the following entry in the /etc/modprobe.d/blacklist.conf file: `blacklist i40evf`. Use i40e driver version 2.0.26 (recommended). |

| Task | Intel 82599 NIC | Intel X710, XL710, and X722 NICs |
|---|---|---|
| 4.Enable SR-IOV Virtual Functions (VFs) on the KVM host. In both the commands in the next two columns: `number_of_VFs` = the number of Virtual VFs that you want to create. `device_name` = the interface name.<br><br>1. Make the VFs persistent by adding the commands that you used to create VFs, to the rc.local file. | If you are using earlier version of kernel 3.8, then add the following entry to the /etc/modprobe.d/ixgbe file and restart the KVM host: `options ixgbe max_vfs` =<`number_of_VFs`>. If you are using kernel 3.8 version or later, create VFs using the following command: `echo <number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`. See example in figure 1.<br><br>See example in figure 3. | If you are using earlier version of kernel 3.8, then add the following entry to the /etc/modprobe.d/i40e.conf file and restart the KVM host: `options i40e max_vfs` =<`number_of_VFs`>. If you are using kernel 3.8 version or later, create VFs using the following command: `echo<number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`. See example in figure 2.<br><br>See example in figure 3. |

> **Important:**
>
> When you create the SR-IOV VFs, ensure that you do not assign MAC addresses to the VFs.

**Figure 1**: Enable SR-IOV VFs on the KVM host for Intel 82599 10G NIC.

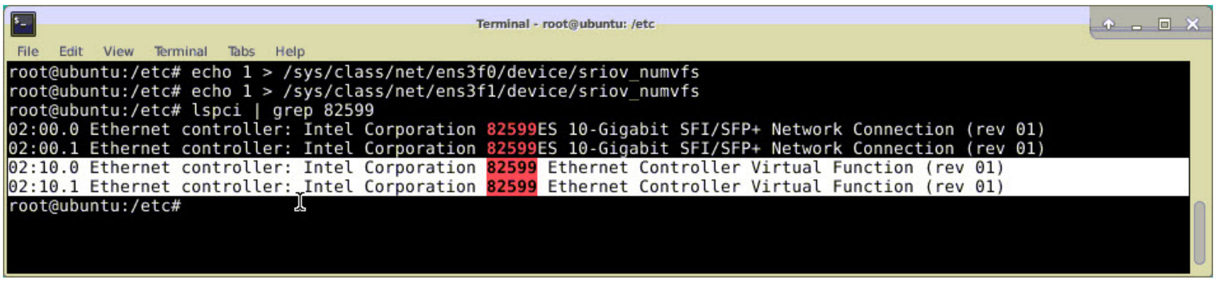



**Figure 2:** Enable SR-IOV VFs on the KVM host for Intel X710 10G and XL710 40G NICs.

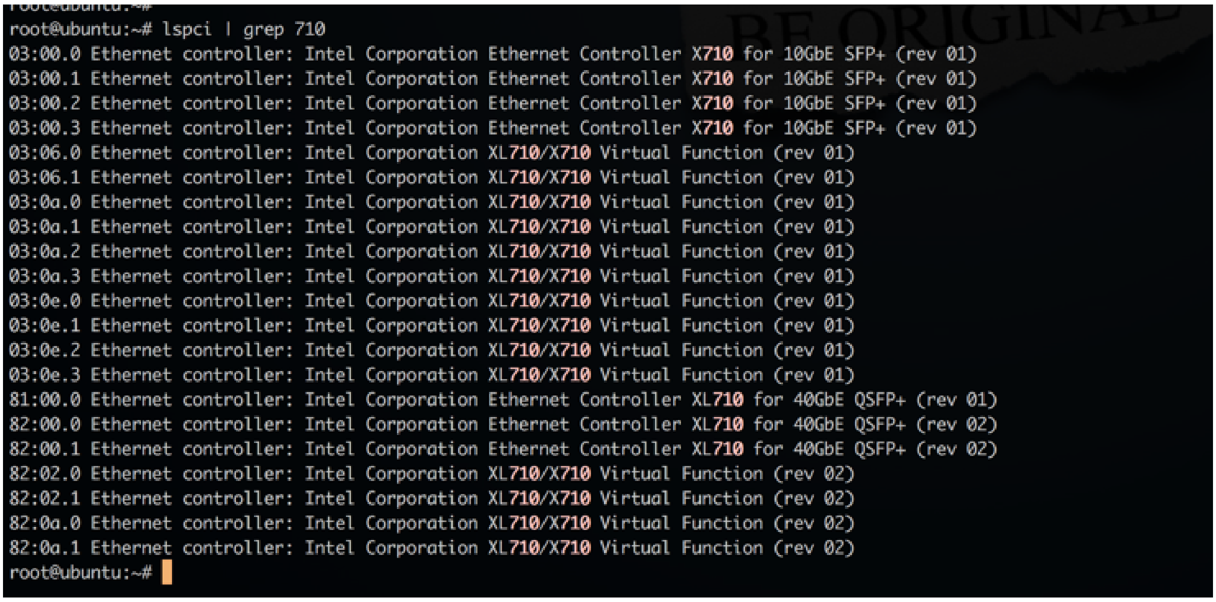
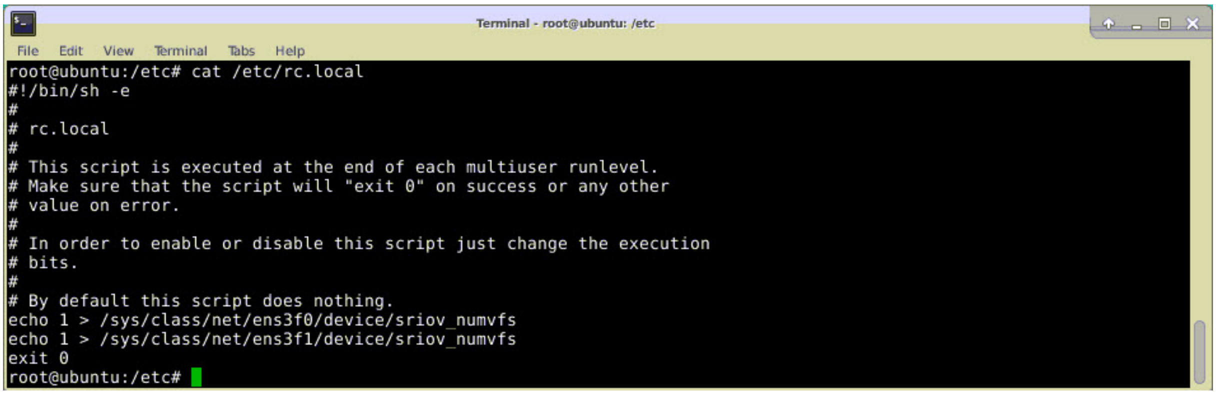



**Figure 3:** Enable SR-IOV VFs on the KVM host for Intel X722 10G NIC.


**Figure 4**: Make the VFs persistent.

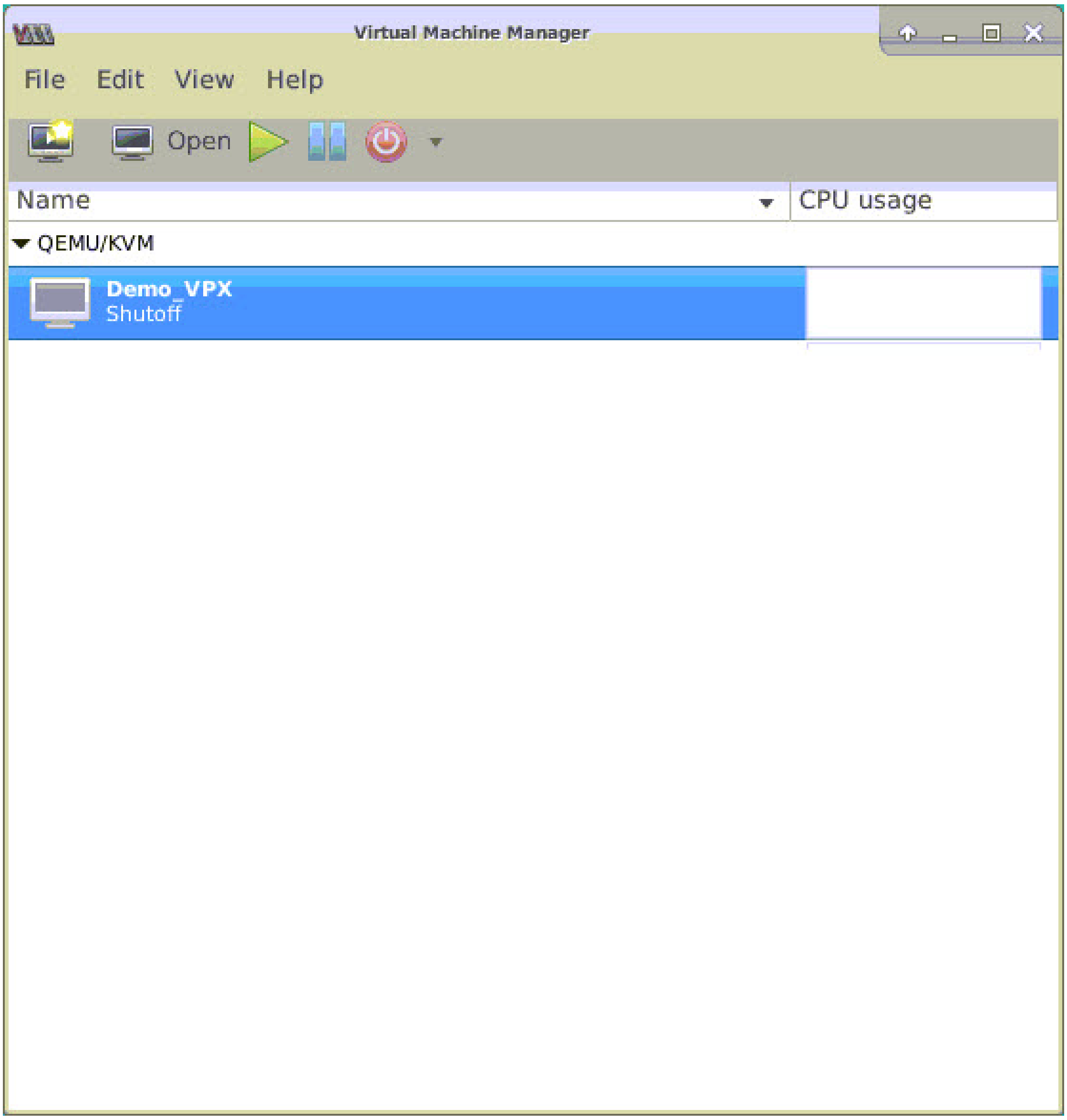



### Configure a NetScaler VPX instance to use SR-IOV network interface

To configure the NetScaler VPX instance to use SR-IOV network interface by using Virtual Machine Manager, complete these steps:

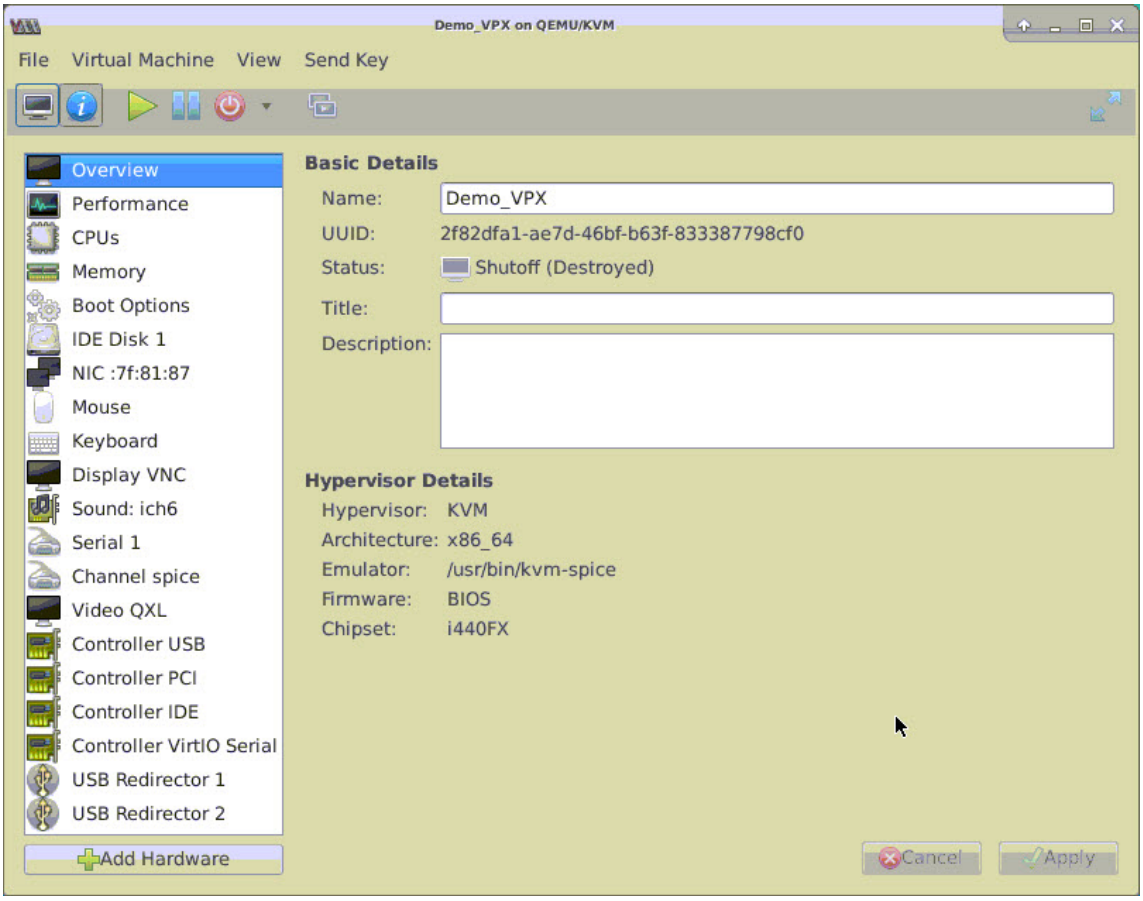
1. Power off the NetScaler VPX instance.
2. Select the NetScaler VPX instance and then select Open.

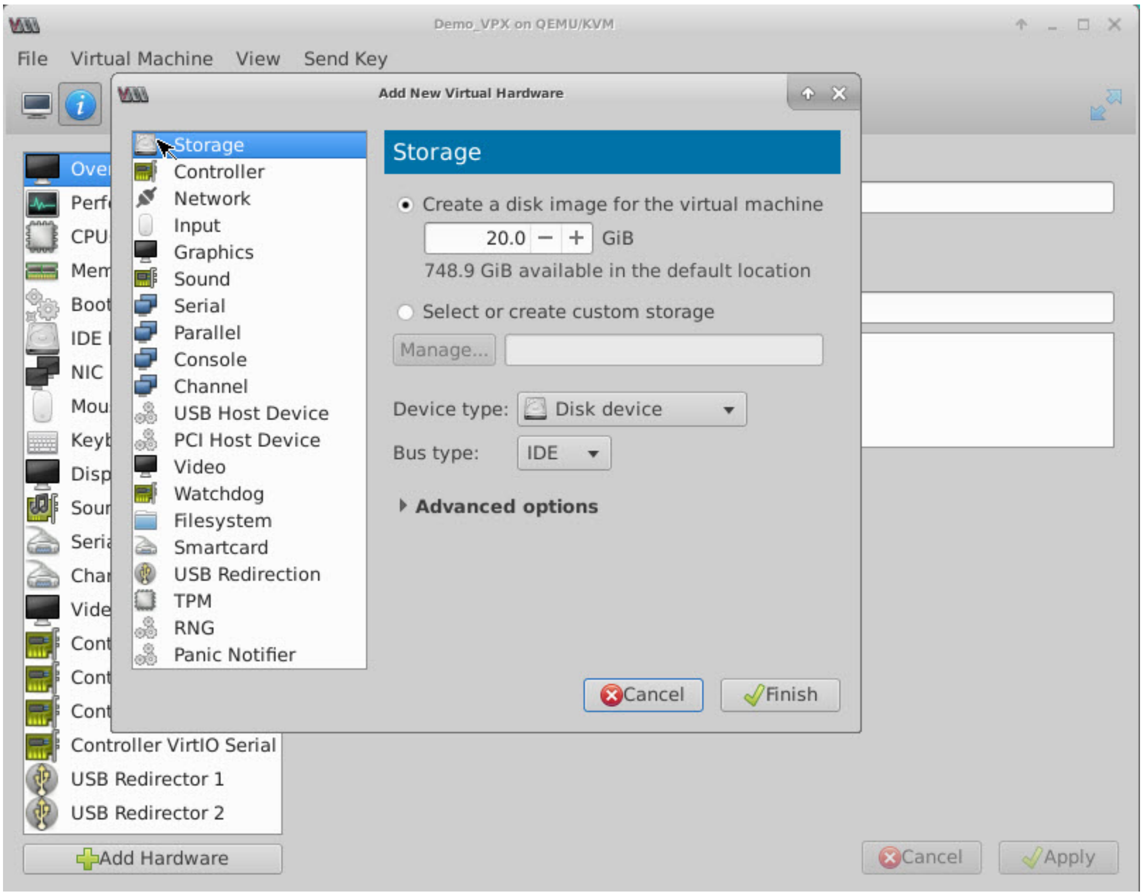



3. In the <virtual machine on KVM> window, select the **i** icon.




4. Select **Add Hardware**.




5. In the **Add New Virtual Hardware** dialog box, do the following:

   a) Select PCI Host Device.
   b) In the Host Device section, select the VF you have created and click Finish.

**Figure 4**: VF for Intel 82599 10G NIC

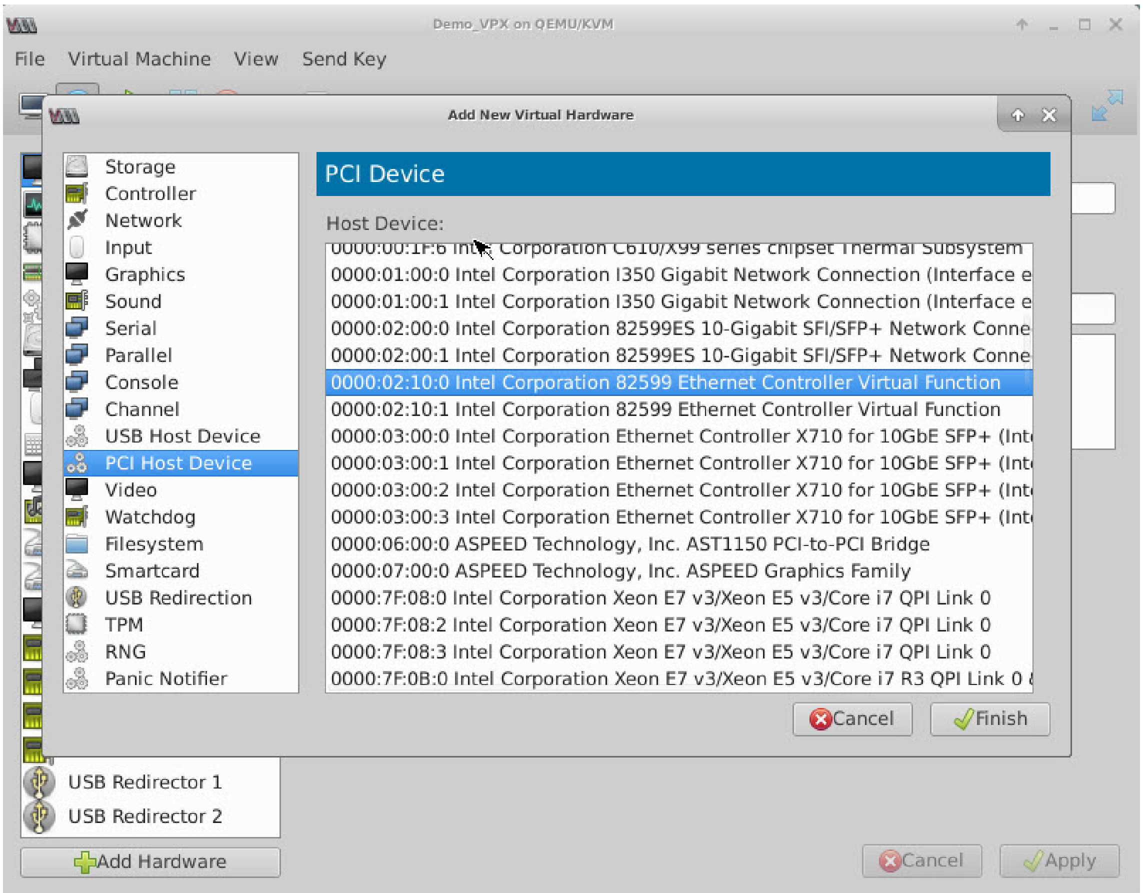



**Figure 5**: VF for Intel XL710 40G NIC

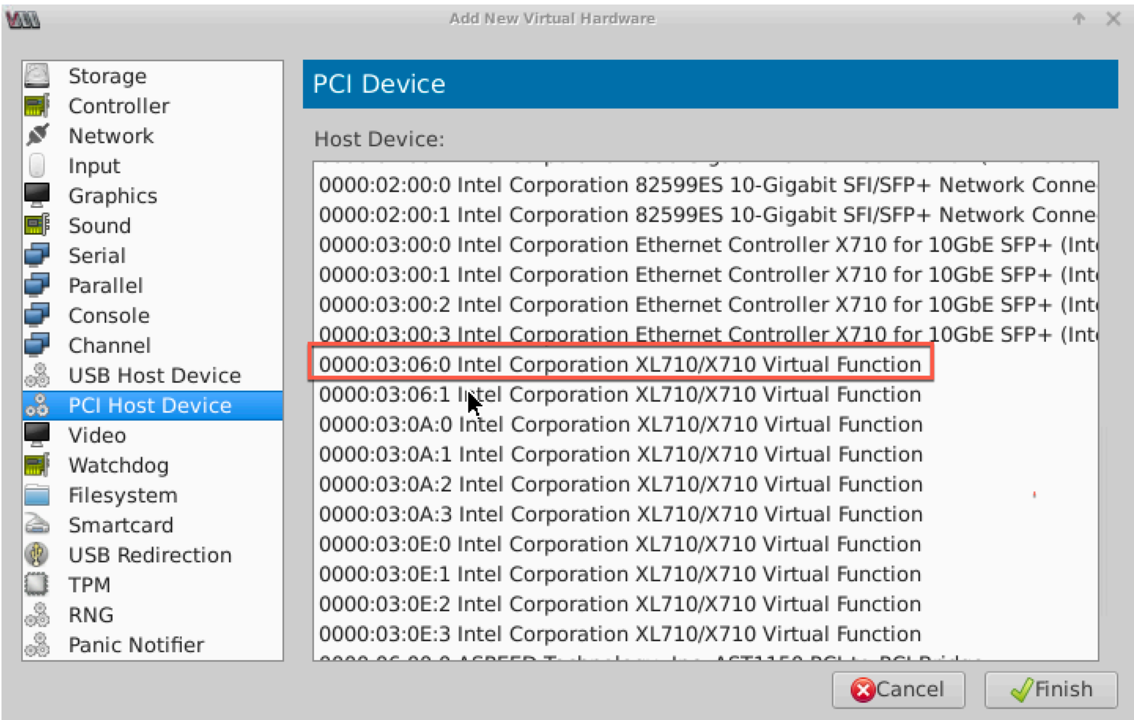
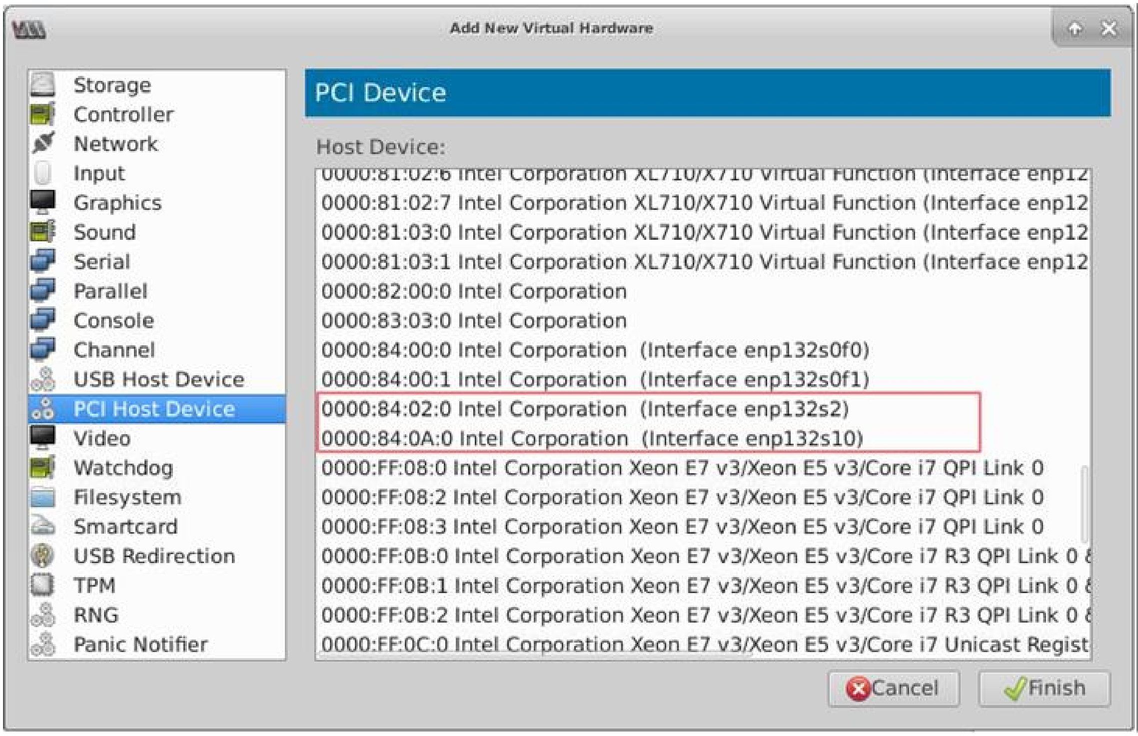


**Figure 6**: VF for Intel X722 10G NIC




6. Repeat Step 4 and 5 to add the VFs that you have created.
7. Power on the NetScaler VPX instance.
8. After the NetScaler VPX instance powers on, use the following command to verify the configuration:

   ```
   show interface summary
   ```

   The output shows all the interfaces that you configured.

   **Figure 6**: output summary for Intel 82599 NIC.

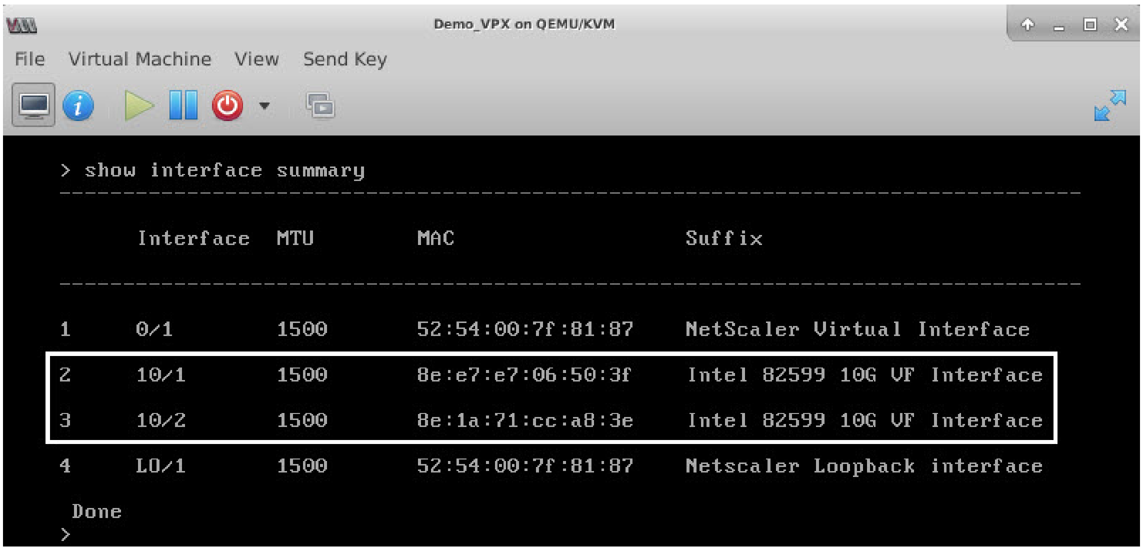
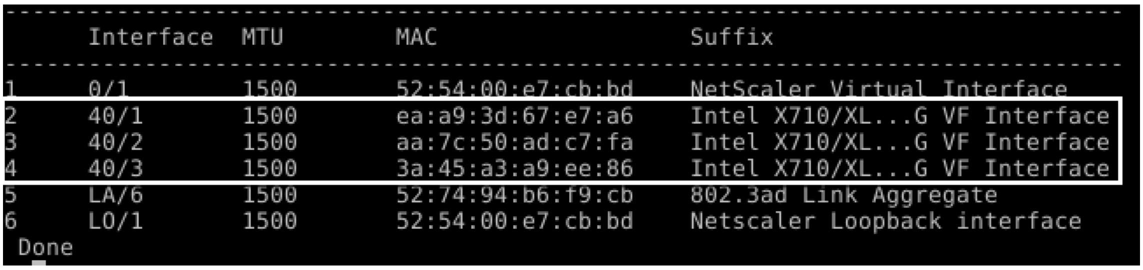



**Figure 7**. Output summary for Intel X710 and XL710 NICs.




### Configure static LA/LACP on the SR-IOV interface

> **Important:**
>
> When you are creating the SR-IOV VFs, ensure that you do not assign MAC addresses to the VFs.

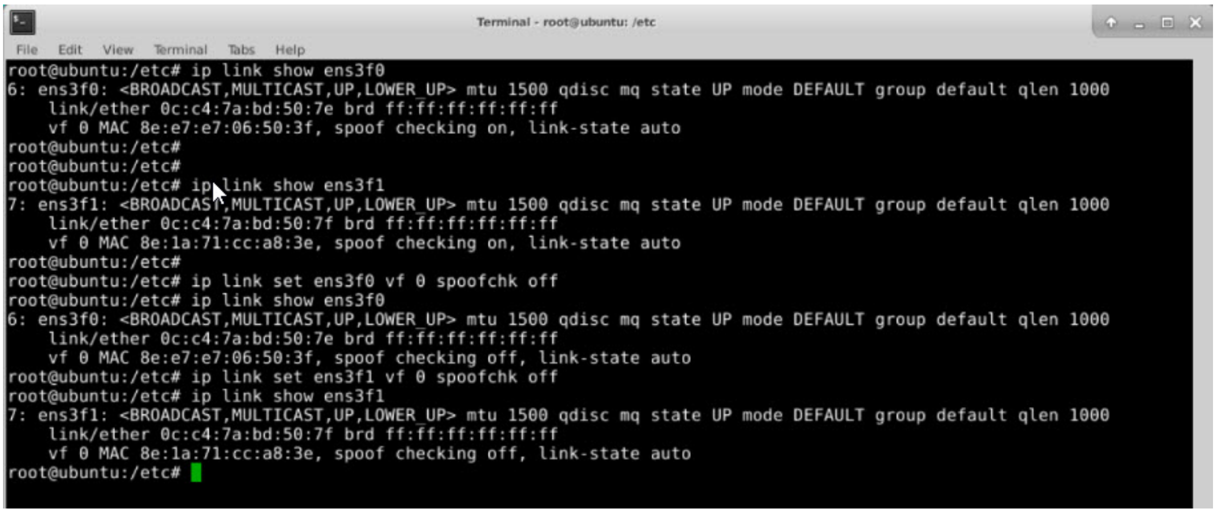
To use the SR-IOV VFs in link aggregation mode, disable spoof checking for VFs that you have created. On the KVM host, use the following command to disable spoof checking:

*ip link set \<interface_name\> vf \<VF_id\> spoofchk off*

Where:

- Interface_name –is the interface name.
- VF_id –is the Virtual Function id.

**Example**:




After you disable spoof checking for all the VFs that you have created. Restart the NetScaler VPX instance and configure link aggregation. For detailed instructions, see Configuring Link Aggregation.

#### Configuring VLAN on the SR-IOV Interface

You can configure VLAN on SR-IOV VFs. For detailed instructions, see Configuring a VLAN.

> **Important:**
>
> Ensure that the KVM host does not contain VLAN settings for the VF interface.

## Configure a NetScaler VPX instance to use PCI passthrough network interfaces

After you have installed and configured a NetScaler VPX instance on the Linux-KVM platform, you can use the Virtual Machine Manager to configure the virtual appliance to use PCI passthrough network interfaces.

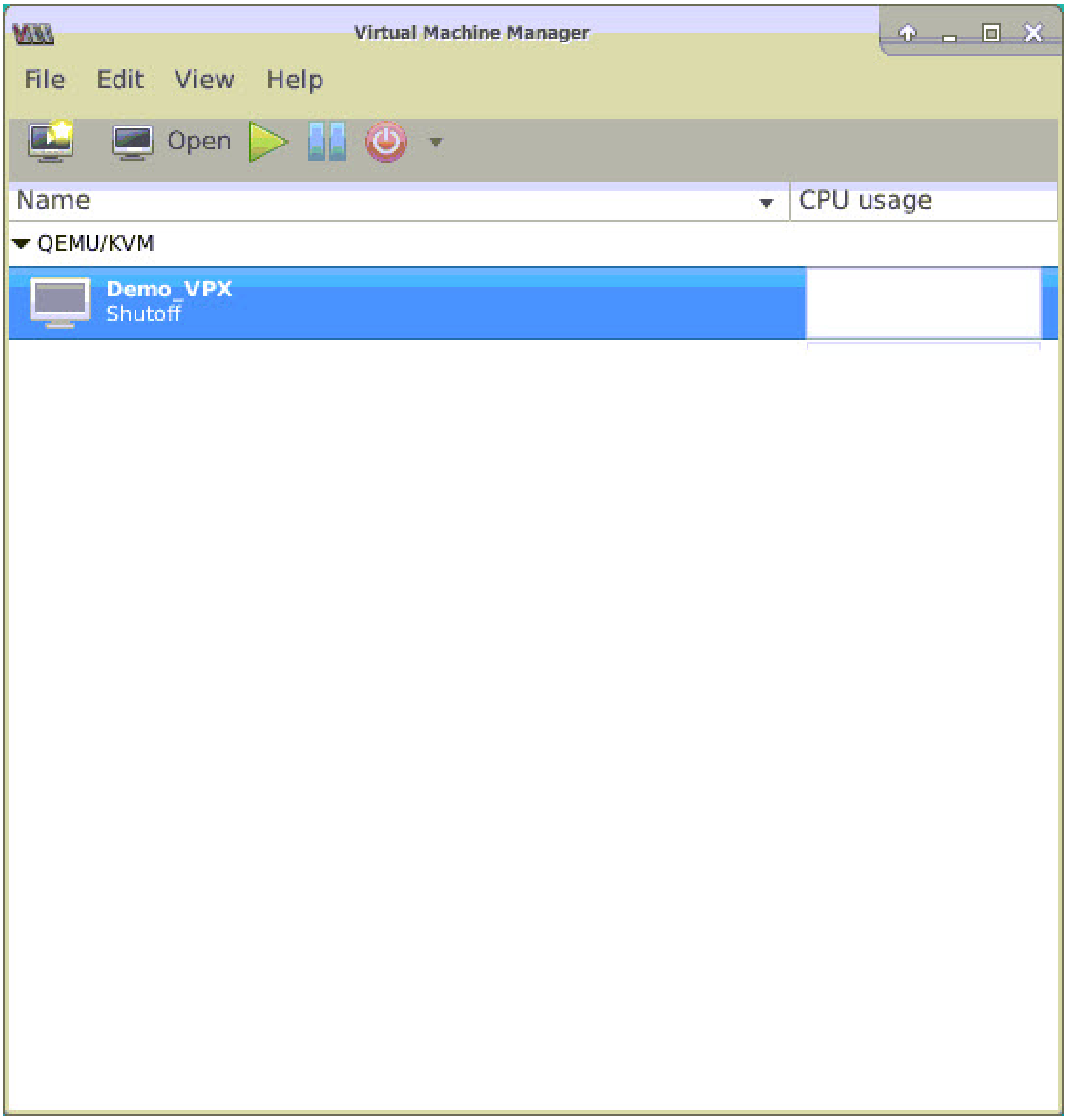
#### Prerequisites

- The firmware version of the Intel XL710 NIC (NIC) on the KVM Host is 5.04.
- The KVM Host supports input–output memory management unit (IOMMU) and Intel VT-d, and they are enabled in the BIOS of the KVM Host. On the KVM Host, to enable IOMMU, add the following entry to the **/boot/grub2/grub.cfg** file: **intel_iommu=1**
- Run the following command and reboot the KVM Host: **Grub2-mkconfig –o /boot/grub2/grub.cfg**

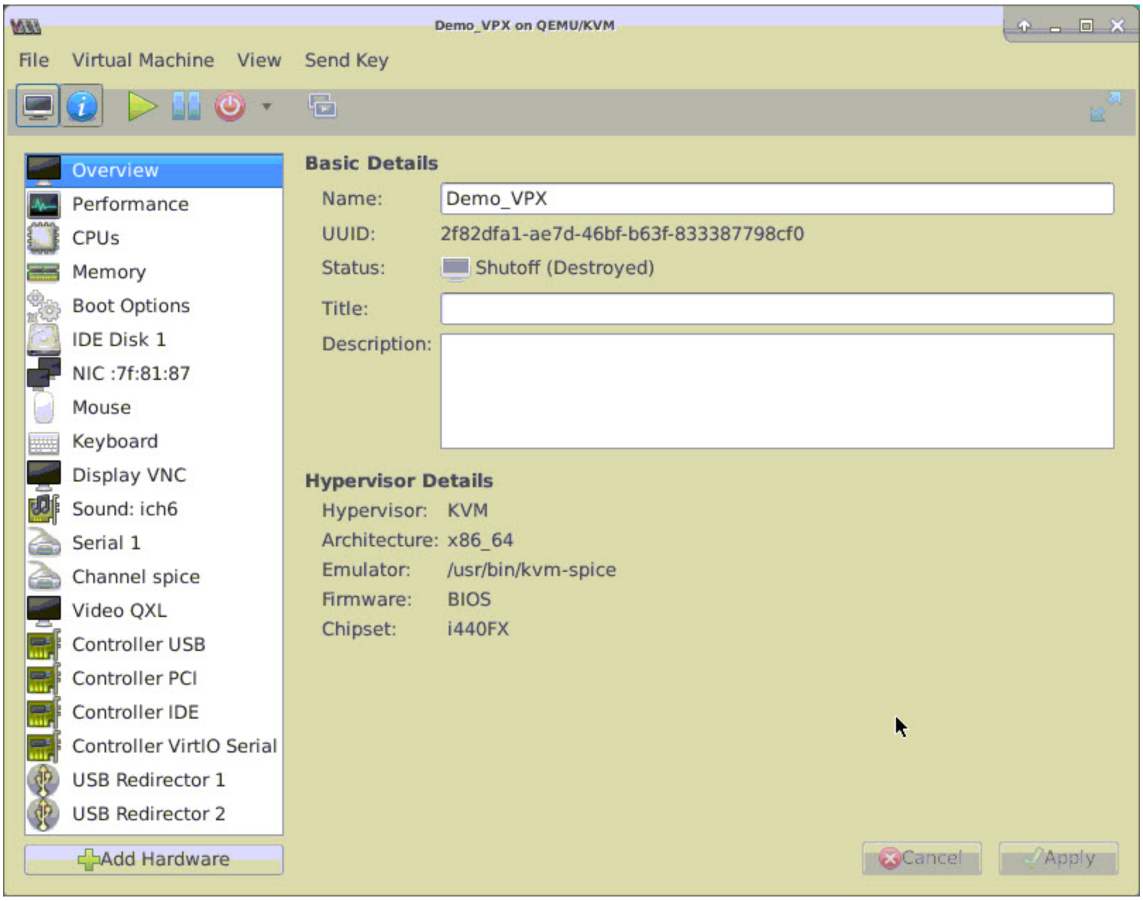
 

**To configure NetScaler VPX instances to use PCI passthrough network interfaces by using the Virtual Machine Manager:**

1. Power off the NetScaler VPX instance.
2. Select the NetScaler VPX instance and click **Open**.




3. In the **virtual_machine on KVM>** window, click the **i** icon.



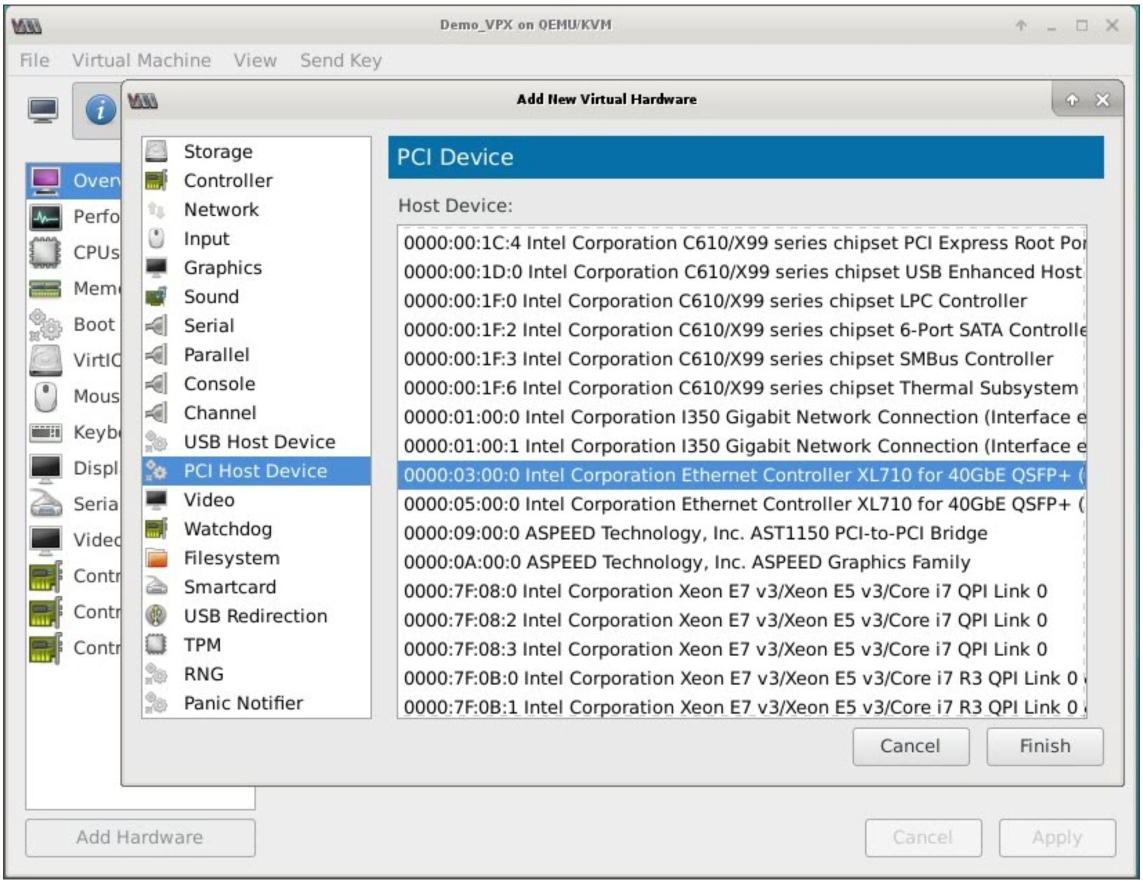

4. Click **Add Hardware**.
5. In the **Add New Virtual Hardware** dialog box, do the following:

   a. Select **PCI Host Device**.

   b. In the **Host Device** section, select the Intel XL710 physical function.

   c. Click **Finish**.



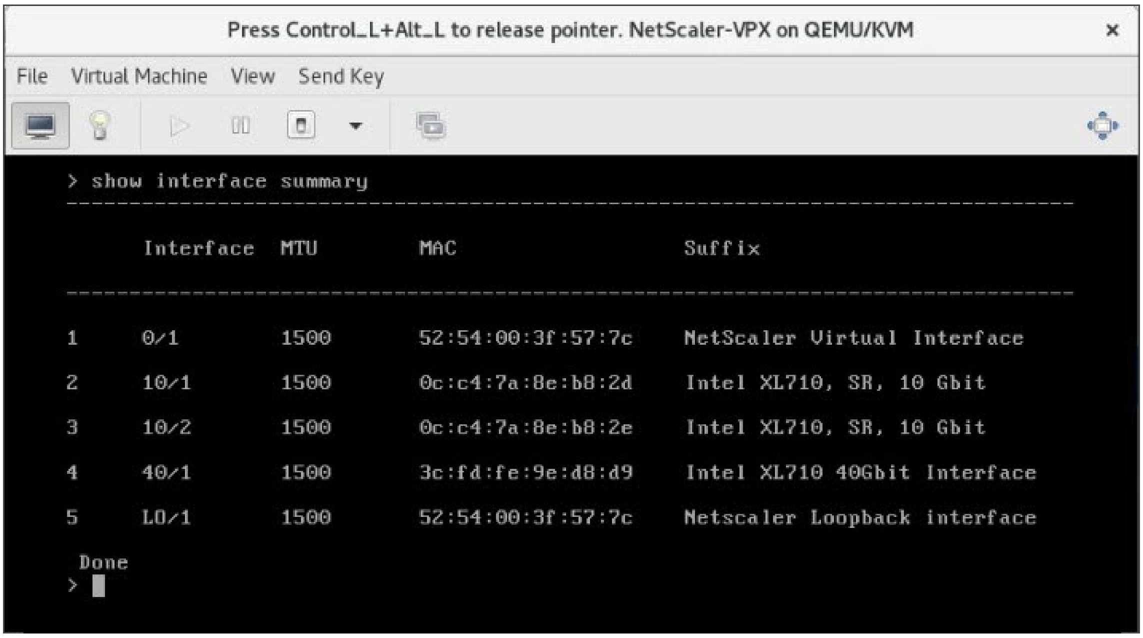

6. Repeat steps **4** and **5** to add any additional Intel XL710 physical functions.
7. Power on the NetScaler VPX instance.
8. Once the NetScaler VPX instance powers on, you can use the following command to verify the configuration:

   COMMAND

   ```
   > show interface summary
   ```

   The output must show all the interfaces that you configured:




## Provision the NetScaler VPX instance by using the virsh program

The `virsh` program is a command line tool for managing VM Guests. Its functionality is similar to that of Virtual Machine Manager. It enables you to change a VM Guest's status (start, stop, pause, and so on), to set up new Guests and devices, and to edit existing configurations. The `virsh` program is also useful for scripting VM Guest management operations.

To provision NetScaler VPX by using the `virsh` program, follow these steps:

1. Use the tar command to untar the NetScaler VPX package. The NSVPX-KVM-*_nc.tgz package contains the following components:

   - The Domain XML file specifying VPX attributes [NSVPX-KVM-*_nc.xml]
   - Check sum of NS-VM Disk Image [Checksum.txt]
   - NS-VM Disk Image [NSVPX-KVM-*_nc.raw]

   Example:

   ```
   tar -xvzf NSVPX-KVM-10.1-117_nc.tgz
   NSVPX-KVM-10.1-117_nc.xml
   NSVPX-KVM-10.1-117_nc.raw
   checksum.txt
   ```

2. Copy the `NSVPX-KVM-\*\\_nc.xml XML` file to a file named `\<DomainName\>-NSVPX-KVM-\*\\_nc.xml`. The <DomainName> is also the name of the virtual machine. Example:

   ```
   cp NSVPX-KVM-10.1-117_nc.xml NetScaler-VPX-NSVPX-KVM-10.1-117_nc.
      xml
   ```

3. Edit the `\<DomainName\>-NSVPX-KVM-\*\\_nc.xml` file to specify the following parameters:

   - name—Specify the name.
   - Mac—Specify the MAC address.

     > **Note:**
     >
     > The domain name and the MAC address have to be unique.

   - source file—Specify the absolute disk-image source path. The file path has to be absolute. You can specify the path of the RAW image file or a QCOW2 image file.

     If you want to specify a RAW image file, specify the disk image source path as shown in the following example:

     Example:

     ```
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.raw'/>
     ```

     Specify the absolute QCOW2 disk-image source path and define the driver type as **qcow2**, as shown in the following example:

     Example:

     ```
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <driver name ='qemu' type='qcow2'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.qcow'/>*
     ```

4. Edit the `\<DomainName\>-NSVPX-KVM-\*\\_nc.xml` file to configure the networking details:

   - source dev—specify the interface.
   - mode—specify the mode. The default interface is **Macvtap Bridge**.

   Example: Mode: MacVTap Bridge Set target interface as `ethx` and mode as bridge Model type as `virtio`

   ```
   <interface type='direct'>
       <mac address='52:54:00:29:74:b3'/>
       <source dev='eth0' mode='bridge'/>
       <target dev='macvtap0'/>
       <model type='virtio'/>
       <alias name='net0'/>
       <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
          function='0x0'/>
     </interface>
   ```

Here, eth0 is the physical interface attached to the VM.

5. Define the VM attributes in the `\<DomainName\>-NSVPX-KVM-\*\\_nc.xml` file by using the following command:

```
virsh define \<DomainName\>-NSVPX-KVM-\*\_nc.xml
```

Example:

```
virsh define NS-VPX-NSVPX-KVM-10.1-117_nc.xml
```

6. Start the VM by entering the following command:

```
virsh start \[\<DomainName\> | \<DomainUUID\>\]
```

Example:

```
virsh start NetScaler-VPX
```

7. Connect the Guest VM through the console:

```
virsh console \[\<DomainName\> | \<DomainUUID\> |\<DomainID\> \]
```

Example:

```
virsh console NetScaler-VPX
```

### Add more interfaces to NetScaler VPX instance using `virsh` program

After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces.

To add more interfaces, follow these steps:

1. Shut down the NetScaler VPX instance running on the KVM.
2. Edit the `\<DomainName\>-NSVPX-KVM-\*\\_nc.xml` file using the command:

```
virsh edit \[\<DomainName\> | \<DomainUUID\>\]
```

3. In the `\<DomainName\>-NSVPX-KVM-\*\\_nc.xml` file, append the following parameters:

    a) **For MacVTap**

    - Interface type—Specify the interface type as 'direct'.
    - MAC address—Specify the MAC address and make sure the MAC address is unique across the interfaces.
    - source dev—Specify the interface name.

- mode—Specify the mode. The modes supported are - Bridge, VEPA, Private, and Pass-through
- model type—Specify the model type as `virtio`

Example:

Mode: MacVTap Pass-through

Set target interface as
`ethx`, Mode as
bridge, and model type as
`virtio`

```
<interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth1' mode='passthrough'/>
      <model type='virtio'/>
 </interface>
```

Here eth1 is the physical interface attached to the VM.

b) **For Bridge Mode**

> **Note:**
>
> Make sure that you have configured a Linux bridge in the KVM host, bound the physical interface to the bridge, and put the bridge in the UP state.

- Interface type—Specify the interface type as 'bridge'.
- MAC address—Specify the MAC address and make sure the MAC address is unique across the interfaces.
- source bridge—Specify the bridge name.
- model type—Specify the model type as `virtio`

Example: Bridge Mode

```
<interface type='bridge'>
      <mac address='52:54:00:2d:43:a4'/>
      <source bridge='br0'/>
      <model type='virtio'/>
 </interface>
```

## Manage the NetScaler VPX guest VMs

You can use the Virtual Machine Manager and the `virsh` program to perform management tasks such as starting or stopping a VM Guest, setting up new guests and devices, editing existing configurations, and connecting to the graphical console through Virtual Network Computing (VNC).

### Manage the VPX guest VMs by using Virtual Machine Manager

- List the VM guests

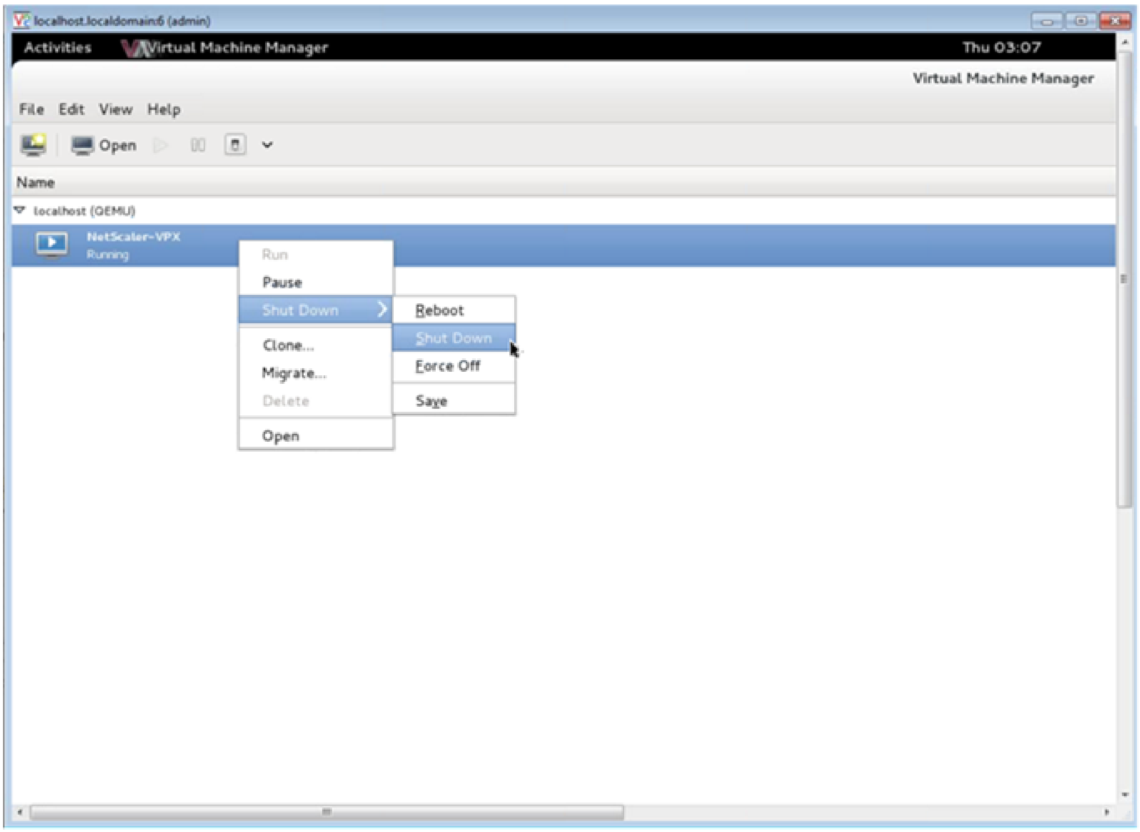
  The main Window of the Virtual Machine Manager displays a list of all the VM Guests for each VM host server it is connected to. Each VM Guest entry contains the virtual machine's name, along with its status (Running, Paused, or Shutoff) displayed as in the icon.

- Open a graphical console

  Opening a Graphical Console to a VM Guest enables you to interact with the machine like you would with a physical host through a VNC connection. To open the graphical console in the Virtual Machine Manager, right-click the VM Guest entry and select the Open option from the pop-up menu.

- Start and shut down a guest

  You can start or stop a VM Guest from the Virtual Machine Manager. To change the state of the VM, right-click the VM Guest entry and select Run or one of the Shut Down options from the pop-up menu.




- Reboot a guest

  You can reboot a VM Guest from the Virtual Machine Manager. To reboot the VM, right-click the VM Guest entry, and then select Shut Down > Reboot from the pop-up menu.

- Delete a guest

  Deleting a VM Guest removes its XML configuration by default. You can also delete a guest's storage files. Doing so completely erases the guest.

  1. In the Virtual Machine Manager, right-click the VM Guest entry.
  2. Select Delete from the pop-up menu. A confirmation window opens.

     > **Note:**
     >
     > The Delete option is enabled only when the VM Guest is shut down.

  3. Click **Delete**.
  4. To completely erase the guest, delete the associated .raw file by selecting the Delete Associated Storage Files check box.

### Manage the NetScaler VPX guest VMs using the `virsh` program

- List the VM Guests and their current states.

  To use `virsh` to display information about the Guests

  ```
  virsh list --all
  ```

  The command output displays all domains with their states. Example output:

  ```
  Id Name                     State
  ----------------------------------
    0 Domain-0               running
    1 Domain-1               paused
    2 Domain-2               inactive
    3 Domain-3               crashed
  ```

- Open a `virsh` console.

  Connect the Guest VM through the console

  ```
  virsh console [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  Example:

  ```
  virsh console NetScaler-VPX
  ```

- Start and shut down a guest.

  Guests can be started using the DomainName or Domain-UUID.

  ```
  virsh start [<DomainName> | <DomainUUID>]
  ```

  Example:

```
virsh start NetScaler-VPX
```

**To shut down a guest**:

```
virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
```

Example:

```
virsh shutdown NetScaler-VPX
```

- Reboot a guest

```
virsh reboot [<DomainID> | <DomainName> | <DomainUUID>]
```

Example:

```
virsh reboot NetScaler-VPX
```

Delete a guest

To delete a Guest VM you must shut down the Guest and undefine the <DomainName>-NSVPX-KVM-*_nc.xml before you run the delete command.

```
virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
virsh undefine [<DomainName> | <DomainUUID>]
```

Example:

```
virsh shutdown NetScaler-VPX
virsh undefine NetScaler-VPX
```

> **Note:**
>
> The delete command doesn't remove disk image file which must be removed manually.

## Provision the NetScaler VPX instance with SR-IOV, on OpenStack

You can deploy high-performance NetScaler VPX instances that use single-root I/O virtualization (SR-IOV) technology, on OpenStack.

You can deploy a NetScaler VPX instance that uses SR-IOV technology, on OpenStack, in three steps:

- Enable SR-IOV Virtual Functions (VFs) on the host.
- Configure and make the VFs available to OpenStack.
- Provision the NetScaler VPX on OpenStack.

### Prerequisites

Ensure that you:

- Add the Intel 82599 NIC (NIC) to the host.
- Download and Install the latest IXGBE driver from Intel.
- Block list the IXGBEVF driver on the host. Add the following entry in the /etc/modprobe.d/blacklist.conf file: Block list `ixgbevf`

> **Note:**
>
> The `ixgbe` driver version must be minimum 5.0.4.

### Enable SR-IOV VFs on the host

Do one of the following steps to enable SR-IOV VFs:

- If you are using a kernel version earlier than 3.8, add the following entry to the /etc/modprobe.d/ixgbe file and restart the host: options ixgbe max_vfs=<number_of_VFs>
- If you are using kernel 3.8 version or later, create VFs by using the following command:

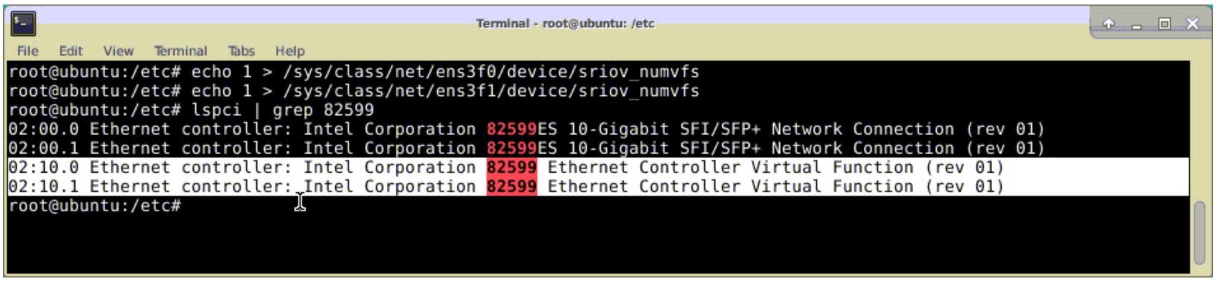
```
echo <number_of_VFs> > /sys/class/net/<device_name>/device/
    sriov_numvfs
```

Where:

- number_of_VFs is the number of Virtual Functions that you want to create.
- device_name is the interface name.

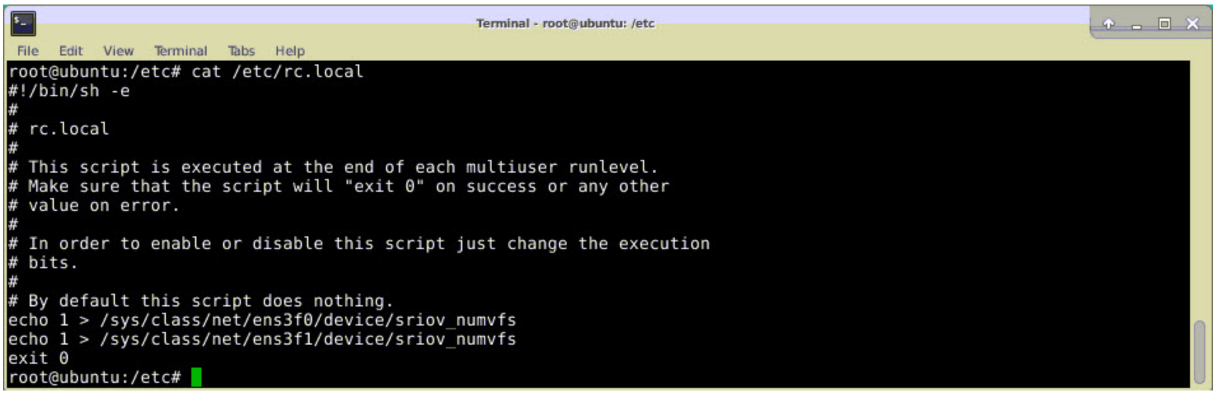
> **Important:**
>
> While you are creating the SR-IOV VFs, make sure that you do not assign MAC addresses to the VFs.

Here is an example of four VFs being created.




Make the VFs persistent, add the commands that you used to created VFs to the **rc.local** file. Here is an example showing content of rc.local file.




For more information, see this Intel SR-IOV Configuration Guide.

### Configure and make the VFs available to OpenStack

Follow the steps given at the link below to configure SR-IOV on OpenStack: https://wiki.openstack.org/wiki/SR-IOV-Passthrough-For-Networking.

### Provision the NetScaler VPX instance on OpenStack

You can provision a NetScaler VPX instance in an OpenStack environment by using the OpenStack CLI.

Provisioning a VPX instance, optionally involves using data from the config drive. The config drive is a special configuration drive that attaches to the instance when it boots. This configuration drive can be used to pass networking configuration information such as management IP address, network mask, and default gateway and so on to the instance before you configure the network settings for the instance.

When OpenStack provisions a VPX instance, it first detects that the instance is booting in an OpenStack environment, by reading a specific BIOS string (OpenStack Foundation) that indicates OpenStack. For Red Hat Linux distributions, the string is stored in /etc/nova/release. This is a standard mechanism that is available in all OpenStack implementations based on KVM hyper-visor platform. The drive must have a specific OpenStack label. If the config drive is detected, the instance attempts to read the following information from the file name specified in the `nova` boot command. In the procedures below, the file is called "userdata.txt."

- Management IP address
- Network mask
- Default gateway

Once the parameters are successfully read, they are populated in the NetScaler stack. This helps in managing the instance remotely. If the parameters are not read successfully or the config drive is not available, the instance transitions to the default behavior, which is:

- The instance attempts to retrieve the IP address information from DHCP.
- If DHCP fails or times-out, the instance comes up with default network configuration (192.168.100.1/16).

#### Provision the NetScaler VPX instance on OpenStack through CLI

You can provision a VPX instance in an OpenStack environment by using the OpenStack CLI. Here's the summary of the steps to provision a NetScaler VPX instance on OpenStack:

1. Extracting the `.qcow2` file from the .tgz file
2. Building an OpenStack image from the qcow2 image
3. Provisioning a VPX instance

To provision a VPX instance in an OpenStack environment, do the following steps.

1. Extract the. `qcow2` file from the `.tqz` file by typing the command:

```
tar xvzf <TAR file>
tar xvzf NSVPX-KVM-12.0-26.2_nc.tgz
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```

2. Build an OpenStack image using the `.qcoz2` file extracted in step 1 by typing the following command:

```
glance image-create --name="<name of the OpenStack image>" --
    property hw_disk_bus=ide --is-public=true --container-format=
    bare --disk-format=qcow2< <name of the qcow2 file>

glance image-create --name="NS-VPX-12-0-26-2" --property
    hw_disk_bus=ide --is-public= true --container-format=bare --
    disk-format=qcow2< NSVPX-KVM-12.0-26.2_nc.qcow2
```

The following illustration provides a sample output for the glance image-create command.

```
+------------------+--------------------------------------
| Property         | Value
+------------------+--------------------------------------
| checksum         | 735dae4ea6e46e39ed3f0acfba02e755
| container_format | bare
| created_at       | 2017-02-16T10:03:29Z
| disk_format      | qcow2
| hw_disk_bus      | ide
| id               | aeaa13e9-b49b-411c-ab54-c61820a8e2f3
| min_disk         | 0
| min_ram          | 0
| name             | NSVPX-KVM-12.0-26.2
| owner            | 06c41a73b32f4b48af55359fd7d3502c
| protected        | False
| size             | 717946880
| status           | active
| tags             | []
| updated_at       | 2017-02-16T10:03:38Z
| virtual_size     | None
| visibility       | private
+------------------+--------------------------------------
```

3. After an OpenStack image is created, provision the NetScaler VPX instance.

```
nova boot --image NSVPX-KVM-12.0-26.2 --config-drive=true --
    userdata
./userdata.txt --flavor m1. medium --nic net-id=3b258725-eaae-
455e-a5de-371d6d1f349f --nic port-id=218ba819-9f55-4991-adb6-
02086a6bdee2 NSVPX-10
```

In the preceding command, userdata.txt is the file which contains the details like, IP address, netmask, and default gateway for the VPX instance. The user data file is a user customizable file. NSVPX-KVM-12.0-26.2 is the name of the virtual appliance that you want to provision. –NIC port-id=218ba819-9f55-4991-adb6-02086a6bdee2 is the OpenStack VF.

The following illustration gives a sample output of the `nova` boot command.

```
+--------------------------------------+---------------------------------------------------------------+
| Property                             | Value                                                         |
+--------------------------------------+---------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                        |
| OS-EXT-AZ:availability_zone          |                                                               |
| OS-EXT-SRV-ATTR:host                 | -                                                             |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -                                                             |
| OS-EXT-SRV-ATTR:instance_name        | instance-0000003c                                             |
| OS-EXT-STS:power_state               | 0                                                             |
| OS-EXT-STS:task_state                | scheduling                                                    |
| OS-EXT-STS:vm_state                  | building                                                      |
| OS-SRV-USG:launched_at               | -                                                             |
| OS-SRV-USG:terminated_at             | -                                                             |
| accessIPv4                           |                                                               |
| accessIPv6                           |                                                               |
| adminPass                            | 43EjPdM5shLz                                                  |
| config_drive                         | True                                                          |
| created                              | 2017-02-20T11:53:37Z                                          |
| flavor                               | m1.medium (3)                                                 |
| hostId                               |                                                               |
| id                                   | 6b9f6968-aab9-463c-b619-d58c73db3187                          |
| image                                | NSVPX-KVM-12.0-26.2 (a5478b8a-8435-48d1-b4a0-1494e2c8f8b1)    |
| key_name                             | -                                                             |
| metadata                             | {}                                                            |
| name                                 | NSVPX-10                                                      |
| os-extended-volumes:volumes_attached | []                                                            |
| progress                             | 0                                                             |
| security_groups                      | default                                                       |
| status                               | BUILD                                                         |
| tenant_id                            | 06c41a73b32f4b48af55359fd7d3502c                              |
| updated                              | 2017-02-20T11:53:38Z                                          |
| user_id                              | 418524f7101b4f0389ecbb36da9916b5                              |
+--------------------------------------+---------------------------------------------------------------+
```

The following illustration shows a sample of the userdata.txt file. The values within the <PropertySection></PropertySection> tags are the values which are user configurable and holds the information like, IP address, netmask, and default gateway.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1">
<PlatformSection>
<Kind>NOVA</Kind>
<Version>2013.1</Version>
<Vendor>Openstack</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
   />
<Property oe:key="com.citrix.netscaler.platform" oe:value="vpx"/>
citrix.com 4
<Property oe:key="com.citrix.netscaler.orch_env"
oe:value="openstack-orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip"
oe:value="10.1.0.100"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask"
oe:value="255.255.0.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway"
```

```
oe:value="10.1.0.1"/>
</PropertySection>
</Environment>
```

**Additional supported Configurations: Creating and Deleting VLANs on SR-IOV VFs from the Host**

Type the following command to create a VLAN on the SR-IOV VF:

`ip link show enp8s0f0 vf 6 vlan 10`

In the preceding command “enp8s0f0”is the name of the physical function.

**Example**: VLAN 10, created on vf 6

```
4: enp8s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT qlen 1000
    link/ether 00:1b:21:7b:d7:88 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
    vf 1 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
    vf 2 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
    vf 3 MAC fa:16:3e:1e:0b:ee, spoof checking on, link-state auto, trust off
    vf 4 MAC fa:16:3e:0d:05:62, spoof checking on, link-state auto, trust off
    vf 5 MAC 5e:46:0d:79:de:f8, spoof checking on, link-state auto, trust off
    vf 6 MAC fa:16:3e:db:ea:b3, vlan 10, spoof checking on, link-state auto, trust off
    vf 7 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
```

Type the following command to delete a VLAN on the SR-IOV VF:

`ip link show enp8s0f0 vf 6 vlan 0`

**Example**: VLAN 10, removed from vf 6

```
[root@localhost ~]# ip link show enp8s0f0
4: enp8s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT qlen 1000
    link/ether 00:1b:21:7b:d7:88 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
    vf 1 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
    vf 2 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
    vf 3 MAC fa:16:3e:1e:0b:ee, spoof checking on, link-state auto, trust off
    vf 4 MAC fa:16:3e:0d:05:62, spoof checking on, link-state auto, trust off
    vf 5 MAC 5e:46:0d:79:de:f8, spoof checking on, link-state auto, trust off
    vf 6 MAC fa:16:3e:db:ea:b3, spoof checking on, link-state auto, trust off
    vf 7 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off
```

These steps complete the procedure for deploying a NetScaler VPX instance that uses SRIOV technology, on OpenStack.

## Configure a NetScaler VPX instance on KVM to use OVS DPDK-based host interfaces

You can configure a NetScaler VPX instance running on KVM (Fedora and RHOS) to use Open vSwitch (OVS) with Data Plane Development Kit (DPDK) for better network performance. This document describes how to configure the NetScaler VPX instance to operate on the `vhost-user` ports exposed by OVS-DPDK on the KVM host.

OVS is a multilayer virtual switch licensed under the open-source Apache 2.0 license. DPDK is a set of libraries and drivers for fast packet processing.

The following Fedora, RHOS, OVS, and DPDK versions are qualified for configuring a NetScaler VPX instance:

| Fedora | RHOS |
| --- | --- |
| Fedora 25 | RHOS 7.4 |
| OVS 2.7.0 | OVS 2.6.1 |
| DPDK 16.11.12 | DPDK 16.11.12 |

### Prerequisites

Before you install DPDK, make sure the host has 1 GB huge pages.

For more information, see this DPDK system requirements documentation. Here is a summary of the steps required to configure a NetScaler VPX instance on KVM to use OVS DPDK-based host interfaces:

- Install DPDK.
- Build and Install OVS.
- Create an OVS bridge.
- Attach a physical interface to the OVS bridge.
- Attach `vhost-user` ports to the OVS data path.
- Provision a KVM-VPX with OVS-DPDK based `vhost-user` ports.

### Install DPDK

To install DPDK, follow the instruction given at this Open vSwitch with DPDK document.

### Build and install OVS

Download OVS from the OVS download page. Next, build, and install OVS by using a DPDK datapath. Follow the instructions given in the Installing Open vSwitch document.

For more detailed information, DPDK Getting Started Guide for Linux.

### Create an OVS bridge

Depending on your need, type the Fedora or RHOS command to create an OVS bridge:

**Fedora command**:

```
> $OVS_DIR/utilities/ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0
    datapath_type=netdev
```

**RHOS command**:

```
ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0 datapath_type=netdev
```

### Attach the physical interface to the OVS bridge

Bind the ports to DPDK and then attach them to the OVS bridge by typing the following Fedora or RHOS commands:

**Fedora command**:

```
> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface
    dpdk0 type=dpdk  options:dpdk-devargs=0000:03:00.0

> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface
    dpdk1 type=dpdk  options:dpdk-devargs=0000:03:00.1
```

**RHOS command**:

```
ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface dpdk0 type=dpdk
    options:dpdk-devargs=0000:03:00.0


ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface dpdk1 type=dpdk
    options:dpdk-devargs=0000:03:00.1
```

The `dpdk-devargs` shown as part of the options specifies the PCI BDF of the respective physical NIC.

### Attach `vhost-user` ports to the OVS data path

Type the following Fedora or RHOS commands to attach `vhost-user` ports to the OVS data path:

**Fedora command**:

```
> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user1 -- set Interface vhost-user1 type=dpdkvhostuser -- set Interface vhost-user1  mtu_request=9000

> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user2 -- set Interface vhost-user2 type=dpdkvhostuser -- set Interface vhost-user2  mtu_request=9000

chmod g+w  /usr/local/var/run/openvswitch/vhost*
```

**RHOS command**:

```
ovs-vsctl add-port ovs-br0 vhost-user1 -- set Interface vhost-user1 type=dpdkvhostuser -- set Interface vhost-user1  mtu_request=9000

ovs-vsctl add-port ovs-br0 vhost-user2 -- set Interface vhost-user2 type=dpdkvhostuser -- set Interface vhost-user2  mtu_request=9000

chmod g+w /var/run/openvswitch/vhost*
```

### Provision a KVM-VPX with OVS-DPDK-based `vhost-user` ports

You can provision a VPX instance on Fedora KVM with OVS-DPDK-based `vhost-user` ports only from the CLI by using the following QEMU commands:
**Fedora command**:

```
qemu-system-x86_64 -name KVM-VPX -cpu host -enable-kvm -m 4096M \

-object memory-backend-file,id=mem,size=4096M,mem-path=/dev/hugepages,share=on -numa node,memdev=mem \

-mem-prealloc -smp sockets=1,cores=2 -drive file=<absolute-path-to-disc-image-file>,if=none,id=drive-ide0-0-0,format=<disc-image-format> \

-device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-0,id=ide0-0-0,bootindex=1 \

-netdev type=tap,id=hostnet0,script=no,downscript=no,vhost=on \

-device virtio-net-pci,netdev=hostnet0,id=net0,mac=52:54:00:3c:d1:ae,bus=pci.0,addr=0x3 \

-chardev socket,id=char0,path=</usr/local/var/run/openvswitch/vhost-user1> \
```

```
-netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce -device virtio-net-pci,mac=00:00:00:00:00:01,netdev=mynet1,mrg_rxbuf=on \
-chardev socket,id=char1,path=</usr/local/var/run/openvswitch/vhost-user2> \
-netdev type=vhost-user,id=mynet2,chardev=char1,vhostforce -device virtio-net
pci,mac=00:00:00:00:00:02,netdev=mynet2,mrg_rxbuf=on \
--nographic
```

For RHOS, use the following sample XML file to provision the NetScaler VPX instance, by using `virsh`
.

```
<domain type='kvm'>
  <name>dpdk-vpx1</name>
  <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>
  <memory unit='KiB'>16777216</memory>
  <currentMemory unit='KiB'>16777216</currentMemory>
  <memoryBacking>
    <hugepages>
      <page size='1048576' unit='KiB'/>
    </hugepages>
  </memoryBacking>
  <vcpu placement='static'>6</vcpu>
  <cputune>
    <shares>4096</shares>
    <vcpupin vcpu='0' cpuset='0'/>
    <vcpupin vcpu='1' cpuset='2'/>
    <vcpupin vcpu='2' cpuset='4'/>
    <vcpupin vcpu='3' cpuset='6'/>
    <emulatorpin cpuset='0,2,4,6'/>
```

```
  </cputune>
  <numatune>
    <memory mode='strict' nodeset='0'/>
  </numatune>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86\_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <cpu mode='custom' match='minimum' check='full'>
    <model fallback='allow'>Haswell-noTSX</model>
    <vendor>Intel</vendor>
    <topology sockets='1' cores='6' threads='1'/>
    <feature policy='require' name='ss'/>
    <feature policy='require' name='pcid'/>
    <feature policy='require' name='hypervisor'/>
    <feature policy='require' name='arat'/>
<domain type='kvm'>
  <name>dpdk-vpx1</name>
  <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>
  <memory unit='KiB'>16777216</memory>
```

```
  <currentMemory unit='KiB'>16777216</currentMemory>
  <memoryBacking>
    <hugepages>
      <page size='1048576' unit='KiB'/>
    </hugepages>
  </memoryBacking>
  <vcpu placement='static'>6</vcpu>
  <cputune>
    <shares>4096</shares>
    <vcpupin vcpu='0' cpuset='0'/>
    <vcpupin vcpu='1' cpuset='2'/>
    <vcpupin vcpu='2' cpuset='4'/>
    <vcpupin vcpu='3' cpuset='6'/>
    <emulatorpin cpuset='0,2,4,6'/>
  </cputune>
  <numatune>
    <memory mode='strict' nodeset='0'/>
  </numatune>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86\_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
```

```
    <acpi/>
    <apic/>
  </features>
  <cpu mode='custom' match='minimum' check='full'>
    <model fallback='allow'>Haswell-noTSX</model>
    <vendor>Intel</vendor>
    <topology sockets='1' cores='6' threads='1'/>
    <feature policy='require' name='ss'/>
    <feature policy='require' name='pcid'/>
    <feature policy='require' name='hypervisor'/>
    <feature policy='require' name='arat'/>
    <feature policy='require' name='tsc\_adjust'/>
    <feature policy='require' name='xsaveopt'/>
    <feature policy='require' name='pdpe1gb'/>
    <numa>
      <cell id='0' cpus='0-5' memory='16777216' unit='KiB' memAccess='
         shared'/>
    </numa>
  </cpu>
  <clock offset='utc'/>
  <on\_poweroff>destroy</on\_poweroff>
  <on\_reboot>restart</on\_reboot>
  <on\_crash>destroy</on\_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='none'/>
```

```
      <source file='/home/NSVPX-KVM-12.0-52.18\_nc.qcow2'/>
      <target dev='vda' bus='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
    </disk>
    <controller type='ide' index='0'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <controller type='usb' index='0' model='piix3-uhci'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
    </controller>
    <controller type='pci' index='0' model='pci-root'/>
    <interface type='direct'>
      <mac address='52:54:00:bb:ac:05'/>
      <source dev='enp129s0f0' mode='bridge'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <interface type='vhostuser'>
      <mac address='52:54:00:55:55:56'/>
      <source type='unix' path='/var/run/openvswitch/vhost-user1' mode='client'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </interface>
    <interface type='vhostuser'>
```

```
      <mac address='52:54:00:2a:32:64'/>
      <source type='unix' path='/var/run/openvswitch/vhost-user2' mode='client'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </interface>
    <interface type='vhostuser'>
      <mac address='52:54:00:2a:32:74'/>
      <source type='unix' path='/var/run/openvswitch/vhost-user3' mode='client'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </interface>
    <interface type='vhostuser'>
      <mac address='52:54:00:2a:32:84'/>
      <source type='unix' path='/var/run/openvswitch/vhost-user4' mode='client'/>
     <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0'/>
    </interface>
    <serial type='pty'>
      <target port='0'/>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
```

```
    <input type='mouse' bus='ps2'/>
    <input type='keyboard' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes'>
      <listen type='address'/>
    </graphics>
    <video>
      <model type='cirrus' vram='16384' heads='1' primary='yes'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02'
          function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08'
          function='0x0'/>
    </memballoon>
  </devices>
</domain
```

### Points to note

In the XML file, the `hugepage` size must be 1 GB, as shown in the sample file.

```
<memoryBacking>
    <hugepages>
      <page size='1048576' unit='KiB'/>
    </hugepages>
```

Also, in the sample file vhost-user1 is the `vhost` user port bound to ovs-br0.

```
<interface type='vhostuser'>
      <mac address='52:54:00:55:55:56'/>
      <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
          'client'/>
```

```
        <model type='virtio'/>

        <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
           function='0x0'/>

    </interface>
```

To bring up the NetScaler VPX instance, start using the `virsh` command.

## Apply NetScaler VPX configurations at the first boot of the NetScaler appliance on the KVM hypervisor

You can apply the NetScaler VPX configurations on the KVM hypervisor during the first boot of the NetScaler appliance. Therefore, a customer setup on a VPX instance can be configured in lesser time.

For more information about Preboot user data and its format, see Apply NetScaler VPX configurations at the first boot of the NetScaler appliance in cloud.

> **Note:**
>
> To bootstrap using preboot user data in KVM hypervisor, the default gateway configuration must be passed in `<NS-CONFIG>` section. For more information on the content of the `<NS-CONFIG>` tag, see the following Sample `<NS-CONFIG>` section.

**Sample <NS-CONFIG> section:**

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

### How to provide preboot user data on KVM hypervisor

You can provide preboot user data on the KVM hypervisor through an ISO file, which is attached using a CDROM device.

### Provide user data using a CDROM ISO file

You can use Virtual Machine Manager (VMM) to inject user data into the Virtual Machine (VM) as an ISO image using the CDROM device. KVM supports CD-ROMs in VM Guest either by directly accessing a physical drive on the VM host server or by accessing ISO images.

The following steps enable you to provide user data using the CDROM ISO file:

1. Create a file with file name `userdata` that contains the preboot user data content.

   > **Note:**
   >
   > File name must be strictly used as `userdata`.

2. Store the `userdata` file in a folder, and build an ISO image using the folder.

   You can build an ISO image with `userdata` file by the following two methods:

   - Using any image processing tool such as PowerISO.
   - Using `mkisofs` command in Linux.

   The following sample configuration shows how to generate an ISO image using the `mkisofs` command in Linux.

   ```
   root@ubuntu:~/sai/19oct# ls -lh
   total 4.0K
   -rw-r--r-- 1 root root 1.1K Oct 19 16:25 userdata
   root@ubuntu:~/sai/19oct#
   root@ubuntu:~/sai/19oct# mkisofs -o kvm-userdata.iso userdata
   I: -input-charset not specified, using utf-8 (detected in locale
       settings)
   Total translation table size: 0
   Total rockridge attributes bytes: 0
   Total directory bytes: 0
   Path table size(bytes): 10
   Max brk space used 0
   175 extents written (0 MB)
   root@ubuntu:~/sai/19oct#
   root@ubuntu:~/sai/19oct# ls -lh
   total 356K
   -rw-r--r-- 1 root root 350K Oct 19 16:25 kvm-userdata.iso
   -rw-r--r-- 1 root root 1.1K Oct 19 16:25 userdata
   ```

3. Provision the NetScaler VPX instance using the standard deployment process to create the VM. But do not power on the VM automatically.
4. Add a CD-ROM device with Virtual Machine Manager using the following steps:
    a) Double-click a VM Guest entry in the Virtual Machine Manager to open its console, and switch to the Details view with **View > Details**.
    b) Click **Add Hardware > Storage > Device type > CDROM device**.
    c) Click **Manage** and select the correct ISO file, and click **Finish**. A new CDROM under **Resources** on your NetScaler VPX instance is created.
5. Power on the VM.

## NetScaler VPX on AWS

You can launch a NetScaler VPX instance on Amazon Web Services (AWS). The NetScaler VPX appliance is available as an Amazon Machine Image (AMI) in AWS marketplace. A NetScaler VPX instance on AWS enables you to use AWS cloud computing capabilities and use NetScaler load balancing and traffic management features for their business needs. The VPX instance supports all the traffic management features of a physical NetScaler appliance, and it can be deployed as standalone instances or in HA pairs. For more information on VPX features, see the VPX data sheet.

### Getting started

Before you get started with your VPX deployment, you must be familiar with the following information:

- AWS terminology
- AWS-VPX support matrix
- Limitations and usage guidelines
- Prerequisites
- How a NetScaler VPX instance on AWS works

### Deploy a NetScaler VPX instance on AWS

In AWS, the following deployment types are supported for VPX instances:

- Standalone
- High availability (Active-Passive)
    - High availability within same zone

  - High availability across different zones using Elastic IP
  - High availability across different zones using Private IP

- Active-Active GSLB
- Autoscaling (Active-Active) using ADM

### Hybrid Deployments

- Deploy NetScaler in AWS Outpost
- Deploy NetScaler in VMC in AWS

### Licensing

A NetScaler VPX instance on AWS requires a license. The licensing option available for NetScaler VPX instances running on AWS is Bring Your Own License (BYOL).

### Automation

- NetScaler ADM: Smart Deployment
- GitHub CFTs: NetScaler templates and scripts for AWS deployment
- GitHub Ansible: NetScaler templates and scripts for AWS deployment
- GitHub Terraform: NetScaler templates and scripts for AWS deployment
- AWS Pattern Library (PL): NetScaler VPX

### Blogs

- How NetScaler on AWS Helps Customers Deliver Applications Securely
- Application delivery in hybrid cloud with NetScaler and AWS
- Citrix is an AWS Networking Competency Partner
- NetScaler: Always ready for public clouds
- Scale out or scale in with ease in public clouds through NetScaler
- Citrix expands ADC deployment choice with AWS Outposts
- Using NetScaler with Amazon VPC ingress routing
- Citrix delivers choice, performance, and simplified deployment in AWS

- The security of NetScaler Web App Firewall –now on the AWS Marketplace
- How Aria Systems uses NetScaler Web App Firewall on AWS

### Videos

- Simplifying public cloud NetScaler deployments through ADM
- Provisioning and configuring NetScaler VPX in AWS using ready-to-use terraform scripts
- Deploy NetScaler HA in AWS using CloudFormation Template
- Deploy NetScaler HA across Availability Zones using AWS QuickStart
- NetScaler Autoscale using ADM

### Customer case studies

- Technology Solution - Xenit AB
- Discover the NetScaler and AWS advantage

### Solutions

- Deploy digital advertising platform on AWS with NetScaler
- Enhancing Clickstream analytics in AWS using NetScaler

### Support

- Open a Support case
- For NetScaler subscription offering, see Troubleshoot a VPX instance on AWS. To file a support case, find your AWS account number and support PIN code, and call NetScaler support.
- For NetScaler Customer Licensed offering or BYOL, ensure that you have the valid support and maintenance agreement. If you do not have an agreement, contact your NetScaler representative.

### Additional References

- AWS On-Demand Webinar - NetScaler on AWS
- NetScaler VPX data sheet
- NetScaler in AWS Marketplace

- NetScaler is part of AWS networking partner solutions (load balancers)
- AWS FAQs

## AWS terminology

This section describes the list of commonly used AWS terms and phrases. For more information, see AWS Glossary.

| Term | Definition |
| --- | --- |
| Amazon Machine Image (AMI) | A machine image, which provides the information required to launch an instance, which is a virtual server in the cloud. |
| Elastic Block Store | Provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. |
| Simple Storage Service (S3) | Storage for the Internet. It is designed to make web-scale computing easier for developers. |
| Elastic Compute Cloud (EC2) | A web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. |
| Elastic Load Balancing (ELB) | Distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones. This increases the fault tolerance of your applications. |
| Elastic network interface (ENI) | A virtual network interface that you can attach to an instance in a Virtual Private Cloud (VPC). |
| Elastic IP (EIP) address | A static, public IPv4 address that you have allocated in Amazon EC2 or Amazon VPC and then attached to an instance. Elastic IP addresses are associated with your account, not a specific instance. They are elastic because you can easily allocate, attach, detach, and free them as your needs change. |

| Term | Definition |
| --- | --- |
| Instance type | Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. |
| Identity and Access Management (IAM) | An AWS identity with permission policies that determine what the identity can and cannot do in AWS. You can use an IAM role to enable applications running on an EC2 instance to securely access your AWS resources. IAM role is required for deploying VPX instances in a high-availability setup. |
| Internet Gateway | Connects a network to the Internet. You can route traffic for IP addresses outside your VPC to the Internet gateway. |
| Key pair | A set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key. |
| Route tables | A set of routing rules that controls the traffic leaving any subnet that is associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time. |
| Security groups | A named set of allowed inbound network connections for an instance. |
| Subnets | A segment of the IP address range of a VPC that EC2 instances can be attached to. You can create subnets to group instances according to security and operational needs. |
| Virtual Private Cloud (VPC) | A web service for provisioning a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. |

| Term | Definition |
|---|---|
| Auto Scaling | A web service to launch or terminate Amazon EC2 instances automatically based on user-defined policies, schedules, and health checks. |
| CloudFormation | A service for writing or changing templates that create and delete related AWS resources together as a unit. |

## AWS-VPX support matrix

The following tables list the supported VPX model and AWS regions, instance types, and services.

Table 1: Supported VPX models on AWS

| Supported VPX model |
|---|
| NetScaler VPX Advanced - 200 Mbps |
| NetScaler VPX Premium - 1 Gbps |
| NetScaler VPX Premium - 5 Gbps |
| NetScaler VPX - Customer Licensed |
| NetScaler VPX FIPS - Customer Licensed |
| NetScaler VPX FIPS ENA - Customer Licensed |

Table: 2 Supported AWS regions

| Supported AWS regions |
|---|
| US West (Oregon) |
| US West (N. California) |
| US East (Ohio) |
| US East (N. Virginia) |
| Asia Pacific (Mumbai) |
| Asia Pacific (Melbourne) |

| Supported AWS regions |
| --- |
| Asia Pacific (Seoul) |
| Asia Pacific (Singapore) |
| Asia Pacific (Sydney) |
| Asia Pacific (Tokyo) |
| Asia Pacific (Hong Kong) |
| Asia Pacific (Osaka) |
| Asia Pacific (Jakarta) |
| Asia Pacific (Hyderabad) |
| Canada (Central) |
| EU (Frankfurt) |
| EU (Ireland) |
| EU (London) |
| EU (Paris) |
| EU (Milan) |
| South America (São Paulo) |
| AWS GovCloud (US-East) |
| AWS GovCloud (US-West) |
| AWS Top Secret (C2S) |
| Middle East (Bahrain) |
| Middle East (UAE) |
| Africa (Cape Town) |
| C2S |

> **Note:**
>
> For AWS Hong Kong region, NetScaler VPX support is available only with BYOL licenses.

Table 3: Supported AWS instance types

| Supported AWS instance types |
| --- |
| c4.large, c4.xlarge, c4.2xlarge, c4.4xlarge, c4.8xlarge |
| c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge, c5.9xlarge, c5.18xlarge, c5.24xlarge |
| c5n.large, c5n.xlarge, c5n.2xlarge, c5n.4xlarge, c5n.9xlarge, c5n.18xlarge |
| c6in.large, c6in.xlarge, c6in.2xlarge, c6in.4xlarge, c6in.8xlarge, c6in.12xlarge, c6in.16xlarge, c6in.24xlarge, c6in.32xlarge |
| d2.xlarge, d2.2xlarge, d2.4xlarge, d2.8xlarge |
| m3.large, m3.xlarge, m3.2xlarge |
| m4.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m4.10xlarge, m4.16xlarge |
| m5.large, m5.xlarge, m5.2xlarge, m5.4xlarge, m5.8xlarge, m5.12xlarge, m5.16xlarge, m5.24xlarge |
| m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge |
| m5n.large, m5n.xlarge, m5n.2xlarge, m5n.4xlarge, m5n.8xlarge, m5n.12xlarge, m5n.16xlarge, m5n.24xlarge |
| m6i.large, m6i.xlarge, m6i.2xlarge, m6i.4xlarge, m6i.8xlarge, m6i.12xlarge, m6i.16xlarge, m6i.24xlarge, m6i.32xlarge |
| r7iz.large, r7iz.xlarge, r7iz.2xlarge, r7iz.4xlarge, r7iz.8xlarge, r7iz.12xlarge, r7iz.16xlarge, r7iz.32xlarge |
| t2.medium, t2.large, t2.xlarge, t2.2xlarge |
| t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge |

> **Note:**
>
> NetScaler VPX provisioned on AWS m6i and r7iz instance types do not support the ENA low latency queue (LLQ) feature.

Table 4: Supported AWS Services

| Supported AWS services |
| --- |
| **EC2:** Launches ADC instances. |
| **Lambda:** Invokes NetScaler VPX NITRO APIs during provisioning of NetScaler VPX instances from CFT. |
| **VPC and VPC ingress routing:** VPC creates isolated networks in which ADC can be launched. VPC ingress routing |
| **Route53:** Distributes traffic across all the NetScaler VPX nodes in the NetScaler Autoscale solution. |
| **ELB:** Distributes traffic across all the NetScaler VPX nodes in the NetScaler Autoscale solution. |

| Supported AWS services |
| --- |
| **Cloudwatch:** Monitors performance and system parameters for NetScaler VPX instance. |
| **AWS Autoscaling:** Used for back-end server autoscaling. |
| **Cloud formation:** CloudFormation templates are used to deploy NetScaler VPX instances. |
| **Simple Queue Service (SQS):** Monitors scale up and scale down events in back-end autoscaling. |
| **Simple Notification Service (SNS):** Monitors scale up and scale down events in back-end autoscaling. |
| **Identity and Access Management (IAM):** Provides access to AWS services and resources. |
| **AWS Outposts:** Provisions NetScaler VPX instances in AWS Outposts. |

NetScaler recommends the following AWS instance types:

- M5 and C5n series for marketplace editions or bandwidth-based pool licensing.
- C5n series for vCPU-based pool licensing.

| VPX offering in AWS marketplace | AWS instance recommendation |
| --- | --- |
| VPX 10, VPX 200 | M5.xLarge |
| VPX 1000, VPX 3G, VPX 5G | M5.2xLarge |

NetScaler recommends the following AWS instance types based on throughput.

| VPX with Pooled licensing (Bandwidth licenses) | AWS instance recommendation |
| --- | --- |
| VPX 8G | C5n.4xLarge |
| VPX 10G, VPX 15G, VPX 25G | C5n.9xLarge |

> **Note:**
>
> The VPX 25G offering doesn't give the desired 25G throughput in AWS but can give a higher SSL transactions rate.

To achieve throughput more than 5G, do the following:

- Choose **NetScaler VPX - Customer Licensed (BYOL)** offering in the AWS marketplace.
- Select **Pooled Licensing (Bandwidth licenses)** in NetScaler GUI or CLI.

To determine your instance based on different metrics such as packets per second, SSL transactions rate, reach out to your NetScaler contact for guidance. For vCPU based Pool licensing and sizing guidance, reach out to NetScaler support.

## Limitations and usage guidelines

The following limitations and usage guidelines apply when deploying a NetScaler VPX instance on AWS:

- Before you start, read the AWS terminology section in Deploy a NetScaler VPX instance on AWS.
- The clustering feature is not supported for VPX.
- For the high availability setup to work effectively, associate a dedicated NAT device to management Interface or associate EIP to NSIP. For more information on NAT, in the AWS documentation, see NAT Instances.
- Data traffic and management traffic must be segregated with ENIs belonging to different subnets.
- Only the NSIP address must be present on the management ENI.
- If a NAT instance is used for security instead of assigning an EIP to the NSIP, appropriate VPC level routing changes are required. For instructions on making VPC level routing changes, in the AWS documentation, see Scenario 2: VPC with Public and Private Subnets.
- A VPX instance can be moved from one EC2 instance type to another (for example, from m3.large to an m3.xlarge).
- For storage options for VPX on AWS, Citrix recommends EBS, because it is durable and the data is available even after it is detached from the instance.
- Dynamic addition of ENIs to VPX is not supported. Restart the VPX instance to apply the update. Citrix recommends you to stop the standalone or HA instance, attach the new ENI, and then restart the instance.
- You can assign multiple IP addresses to an ENI. The maximum number of IP addresses per ENI is determined by the EC2 instance type, see the section "IP Addresses Per Network Interface Per Instance Type"in Elastic Network Interfaces. You must allocate the IP addresses in AWS before you assign them to ENIs. For more information, see Elastic Network Interfaces.
- Citrix recommends that you avoid using the enable and disable interface commands on NetScaler VPX interfaces.
- The NetScaler `set ha node \<NODE\\_ID\> -haStatus STAYPRIMARY` and `set ha node \<NODE\\_ID\> -haStatus STAYSECONDARY` commands are disabled by default.
- IPv6 is not supported for VPX.
- Due to AWS limitations, these features are not supported:
  - Gratuitous ARP(GARP)

  - L2 mode
  - Tagged VLAN
  - Dynamic Routing
  - virtual MAC

- For RNAT to work, ensure **Source/Destination Check** is disabled. For more information, see "Changing the Source/Destination Checking"in Elastic Network Interfaces.
- In a NetScaler VPX deployment on AWS, in some AWS regions, the AWS infrastructure might not be able to resolve AWS API calls. This happens if the API calls are issued through a nonmanagement interface on the NetScaler VPX instance.
  As a workaround, restrict the API calls to the management interface only. To do that, create an NSVLAN on the VPX instance and bind the management interface to the NSVLAN by using the appropriate command.
  For example:
  ```
  set ns config -nsvlan <vlan id> -ifnum 1/1 -tagged NO
  save config
  ```
  Restart the VPX instance at the prompt. For more information about configuring `nsvlan`, see Configuring NSVLAN.
- In the AWS console, the vCPU usage shown for a VPX instance under the **Monitoring** tab might be high (up to 100 percent), even when the actual usage is much lower. To see the actual vCPU usage, navigate to **View all CloudWatch metrics**. For more information, see Monitor your instances using Amazon CloudWatch.

## Prerequisites

Before attempting to create a VPX instance in AWS, ensure you have the following:

- **An AWS account**: to launch a NetScaler VPX AMI in an AWS Virtual Private Cloud (VPC). You can create an AWS account for free at [www.aws.amazon.com] (http://www.aws.amazon.com).
- **An AWS Identity and Access Management (IAM) user account**: to securely control access to AWS services and resources for your users. For more information about how to create an IAM user account, see Creating IAM Users (Console). An IAM role is mandatory for both standalone and high availability deployments.

  The IAM role associated with your AWS account must have the following IAM permissions for various scenarios.

  **HA pair with IPv4 addresses in the same AWS zone:**

  ```
  "ec2:DescribeInstances",
  ```

```
"ec2:AssignPrivateIpAddresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
```

**HA pair with IPv6 addresses in the same AWS zone:**

```
"ec2:DescribeInstances",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
```

**HA pair with both IPv4 and IPv6 addresses in the same AWS zone:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
```

**HA pair with elastic IP addresses across different AWS zones:**

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
```

**HA pair with private IP addresses across different AWS zones:**

```
"ec2:DescribeInstances",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
```

**HA pair with both private IP and elastic IP addresses across different AWS zones:**

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
```

```
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
```

**AWS backend autoscaling:**

```
"ec2:DescribeInstances",
"autoscaling:*",
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:ListTopics",
"sns:Subscribe",
"sqs:CreateQueue",
"sqs:ListQueues",
"sqs:DeleteMessage",
"sqs:GetQueueAttributes",
"sqs:SetQueueAttributes",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
"ec2:CreateTags"
```

> **Note:**
>
> - If you use any combination of the preceding features, use the combination of IAM permissions for each of the features.
> - If you use the Citrix CloudFormation template, the IAM role is automatically created. The template does not allow selecting an already created IAM role.
> - When you log on to the VPX instance through the GUI, a prompt to configure the required privileges for the IAM role appears. Ignore the prompt if you've already configured the privileges.

- **AWS CLI**: To use all the functionality provided by the AWS Management Console from your terminal program. For more information, see the AWS CLI user guide. You also need the AWS CLI to change the network interface type to SR-IOV.
- **Elastic Network Adapter (ENA)**: For ENA driver-enabled instance type, for example M5, C5 instances, the firmware version must be 13.0 and above.
- You must configure Instance Metadata Service (IMDS) on the EC2 instance for NetScaler VPX. IMDSv1 and IMDSv2 are two modes available for accessing instance metadata from a running AWS EC2 instance. IMDSv2 is more secure than IMDSv1. You can configure the instance either to use both methods (the default option) or only the IMDSv2 mode (by disabling IMDSv1). Citrix ADC VPX supports IMDSv2 only mode from NetScaler VPX release 13.1.48.x onwards.

## Configure AWS IAM roles on NetScaler VPX instance

Applications that run on an Amazon EC2 instance must include AWS credentials in the AWS API requests. You can store AWS credentials directly within the Amazon EC2 instance and allow applications in that instance to use those credentials. But you then have to manage the credentials and ensure that they securely pass the credentials to each instance and update each Amazon EC2 instance when it's time to rotate the credentials. That's a lot of additional work.

Instead, you can and must use an Identity and Access Management (IAM) role to manage temporary credentials for applications that run on an Amazon EC2 instance. When you use a role, you don't have to distribute long-term credentials (such as a user name and password or access keys) to an Amazon EC2 instance. Instead, the role provides temporary permissions that applications can use when they make calls to other AWS resources. When you launch an Amazon EC2 instance, you specify an IAM role to associate with the instance. Applications that run on the instance can then use the role-supplied temporary credentials to sign API requests.

The IAM role associated with your AWS account must have the following IAM permissions for various scenarios.

**HA pair with IPv4 addresses in the same AWS zone:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
```

**HA pair with IPv6 addresses in the same AWS zone:**

```
"ec2:DescribeInstances",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
```

**HA pair with both IPv4 and IPv6 addresses in the same AWS zone:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
```

**HA pair with elastic IP addresses across different AWS zones:**

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
```

```
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
```

**HA pair with private IP addresses across different AWS zones:**

```
"ec2:DescribeInstances",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
```

**HA pair with both private IP and elastic IP addresses across different AWS zones:**

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
```

**AWS backend autoscaling:**

```
"ec2:DescribeInstances",
"autoscaling:*",
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:ListTopics",
"sns:Subscribe",
"sqs:CreateQueue",
"sqs:ListQueues",
"sqs:DeleteMessage",
"sqs:GetQueueAttributes",
"sqs:SetQueueAttributes",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
```

**Points to note:**

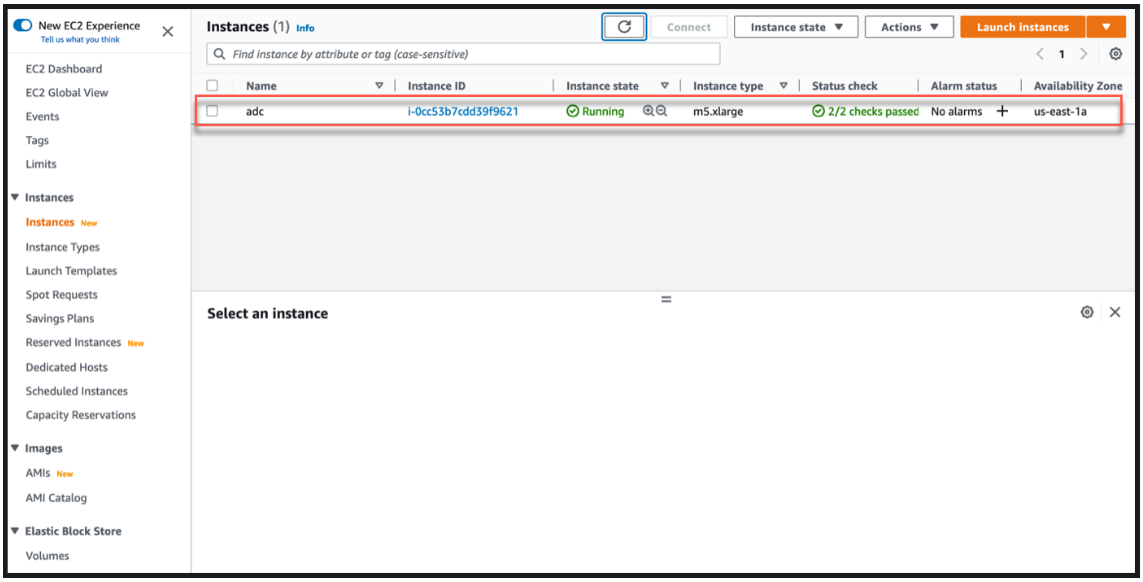
- If you use any combination of the preceding features, use the combination of IAM permissions for each of the features.
- If you use the Citrix CloudFormation template, the IAM role is automatically created. The template does not allow selecting an already created IAM role.
- When you log on to the VPX instance through the GUI, a prompt to configure the required privileges for the IAM role appears. Ignore the prompt if you've already configured the privileges.
- An IAM role is mandatory for both standalone and high availability deployments.

### Create an IAM role

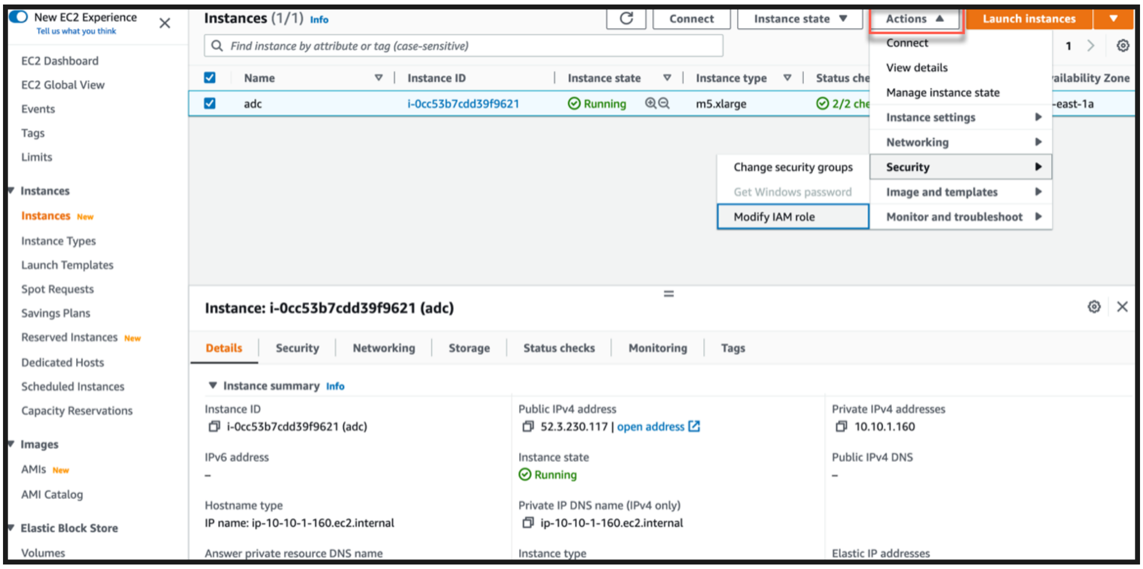
This procedure describes how to create an IAM role for the AWS back-end autoscaling feature.

> **Note:**
>
> You can follow the same procedure to create any IAM roles corresponding to other features.

1. Log on to the AWS Management Console for EC2.
2. Go to EC2 instance page, and select your ADC instance.

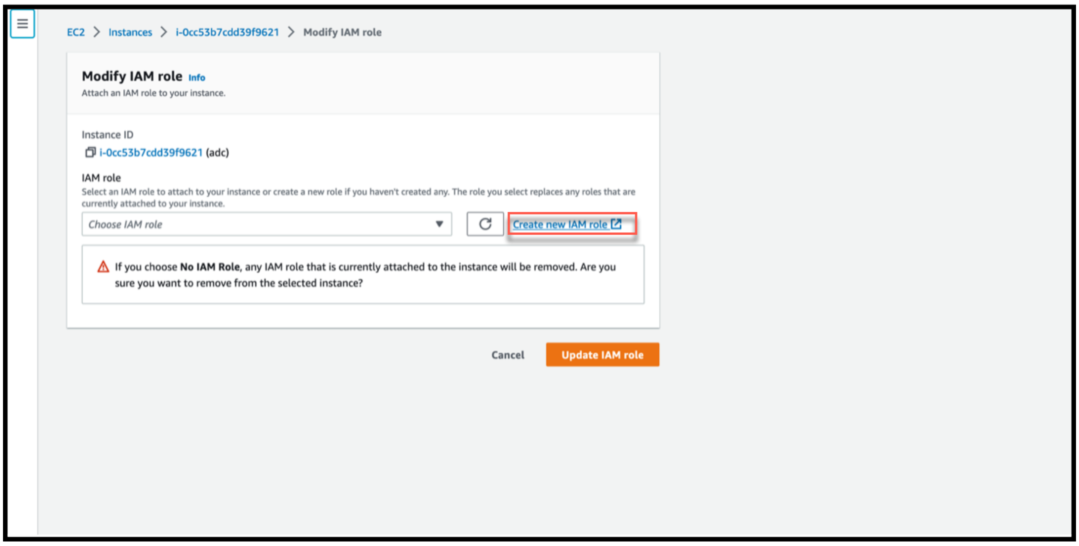



3. Navigate to **Actions > Security > Modify IAM role**.

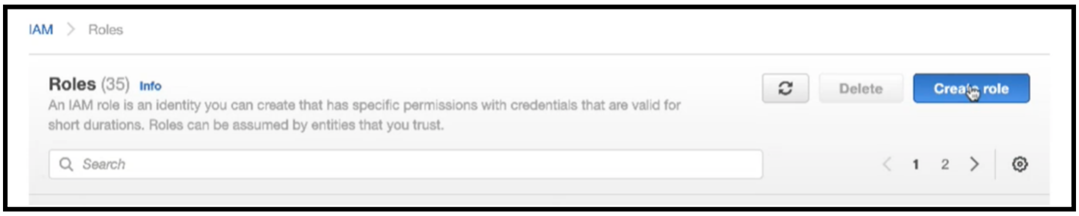



4. In the **Modify IAM role** page, you can either choose an existing IAM role or create a IAM role.
5. To create a IAM role, follow these steps:

a) In the **Modify IAM role** page, click **Create new IAM role**.




b) In the **Roles** page, click **Create role**.




c) Select **AWS service** under **Trusted entity type** and **EC2** under **Common use cases** and then click **Next**.


d) In the **Add permissions** page, click **Create policy**.


e) Click the **JSON** tab to open the JSON editor.


f) In the JSON editor, delete everything and paste the IAM permissions for the feature that you want to use.

For example, paste the following IAM permissions for the AWS back-end autoscaling feature:

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

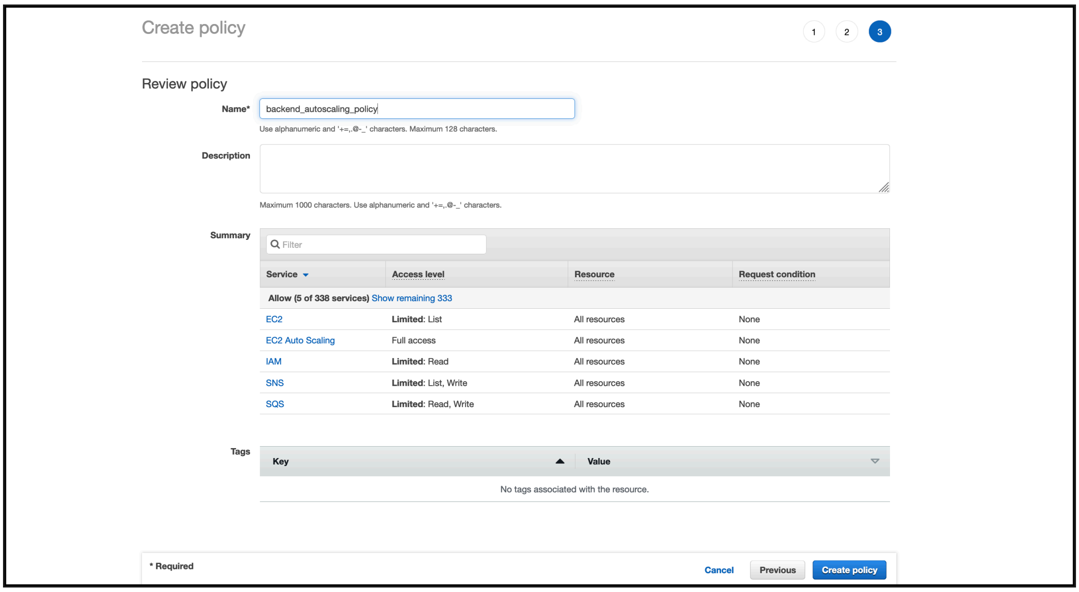
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "autoscaling:*",
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:ListTopics",
                "sns:Subscribe",
                "sqs:CreateQueue",
                "sqs:ListQueues",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:SetQueueAttributes",
                "iam:SimulatePrincipalPolicy",
                "iam:GetRole"
            ],
            "Resource": "*"
         }

    ]
 }
```

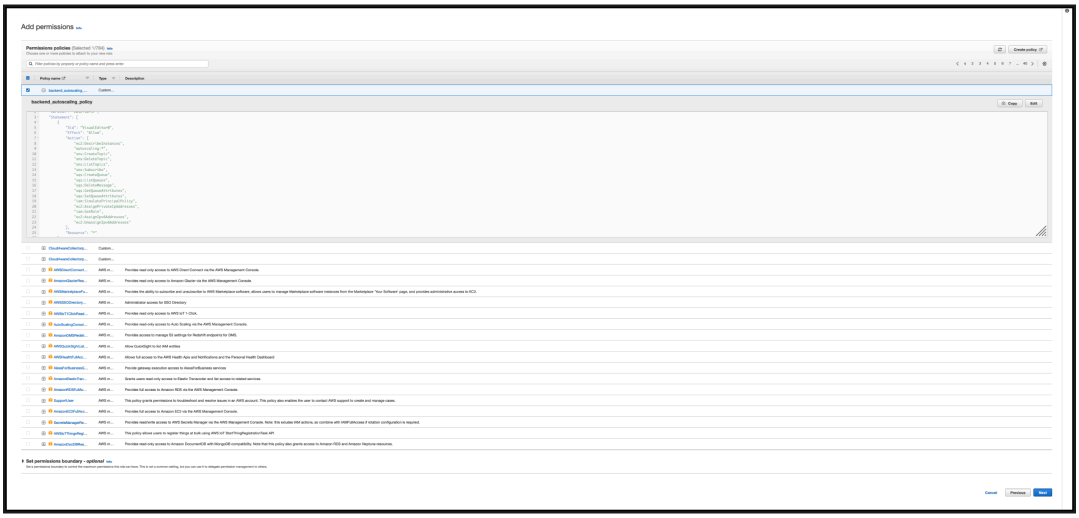
Ensure that the "Version"key-value pair that you provide is the same as the one automatically generated by AWS.

g) Click **Next: Review**.

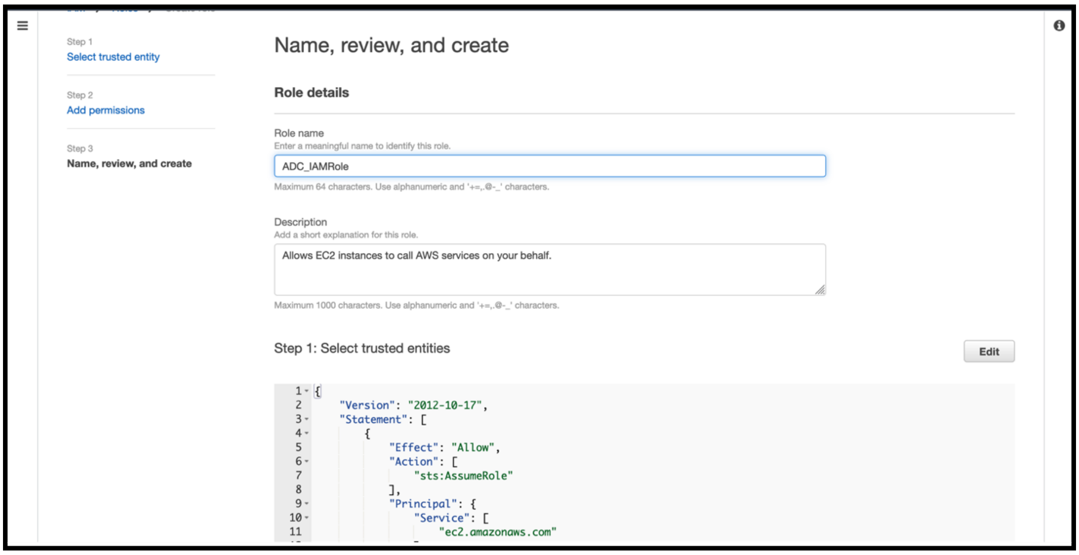

h) In the **Review policy** tab, give a valid name to the policy, and click **Create Policy**.




i) In the **Identity Access Management** page, click the policy name that you created. Expand the policy to check the entire JSON, and click **Next**.

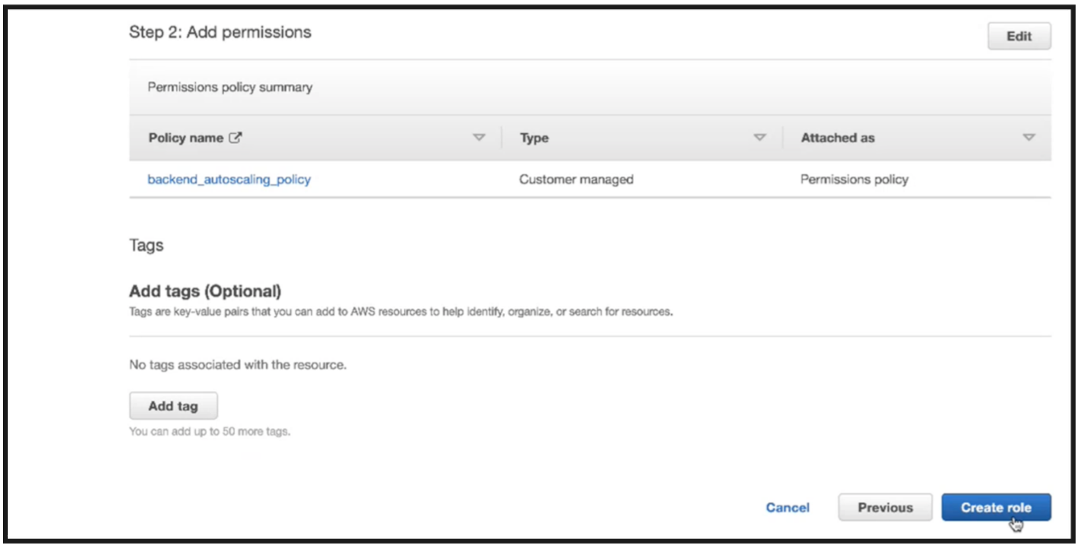



j) In the **Name, review, and create** page, give a valid name to the role.

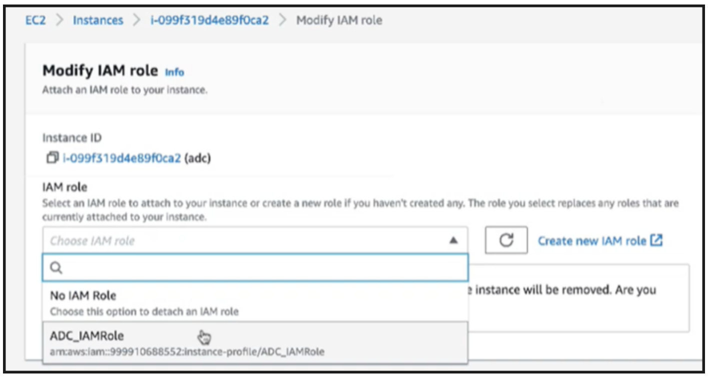



k) Click **Create role**.

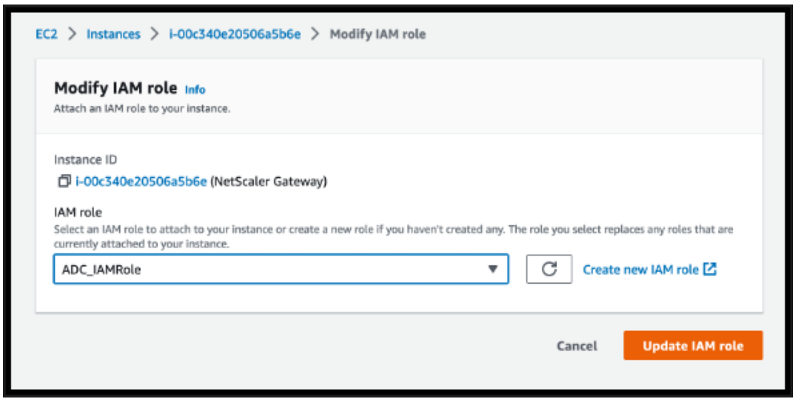
 




6. Repeat steps: 1, 2 and 3. Select the **Refresh** button and select the drop-down menu to see the role that you created.




7. Click **Update IAM role**.

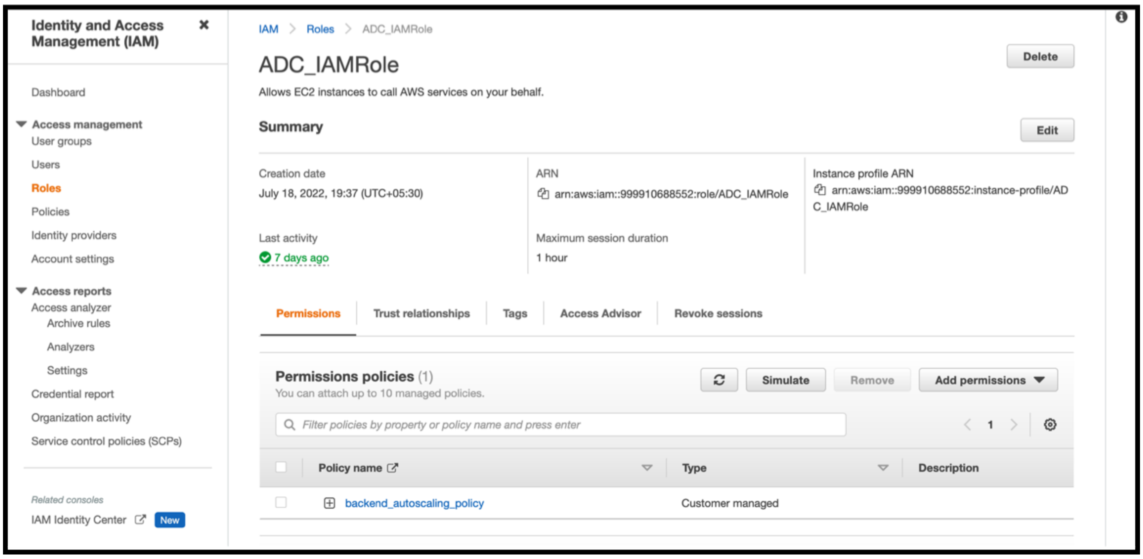


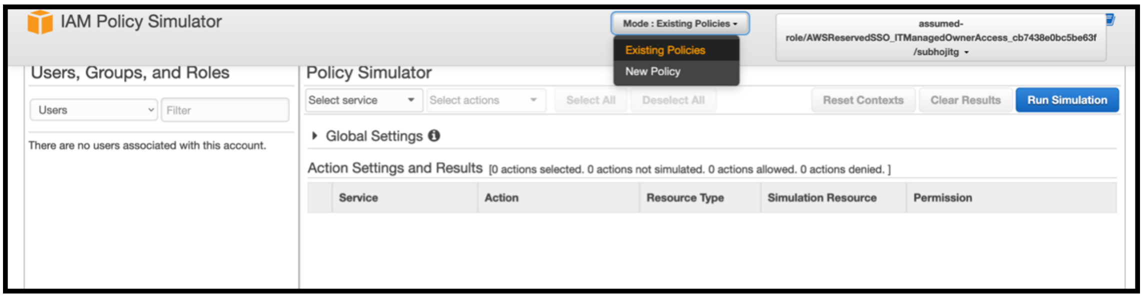
### Test IAM policies with the IAM policy simulator

The IAM policy simulator is a tool that enables you to test the effects of IAM access control policies before committing them into production. It is easier to verify and troubleshoot permissions.

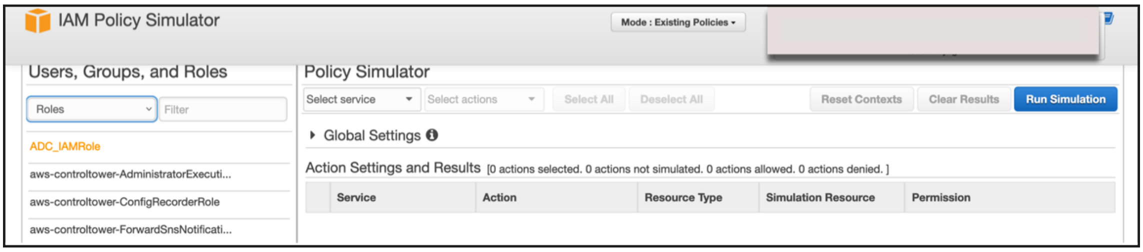
1. In the **IAM** page, select the IAM role that you want to test, and click **Simulate**. In the following example, “ADC_IAMRole”is the IAM role.




2. In the **IAM policy simulator** console, select **Existing Policies** as the **Mode**.

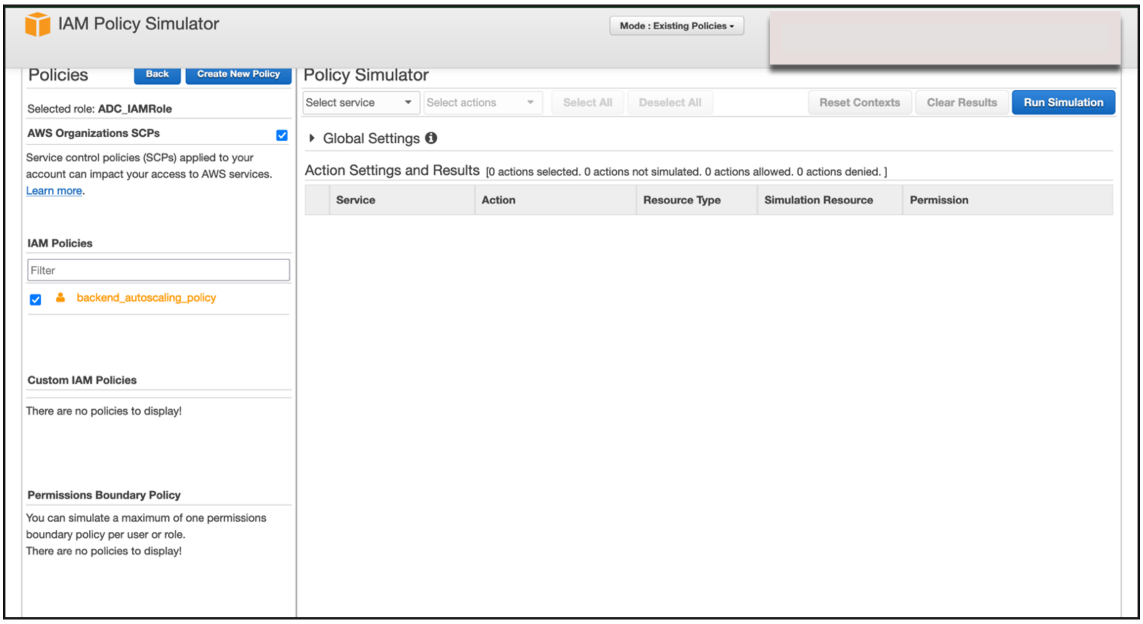



3. In the **Users, Groups and Roles** tab, select **Roles** from the drop-down menu and choose an existing role.

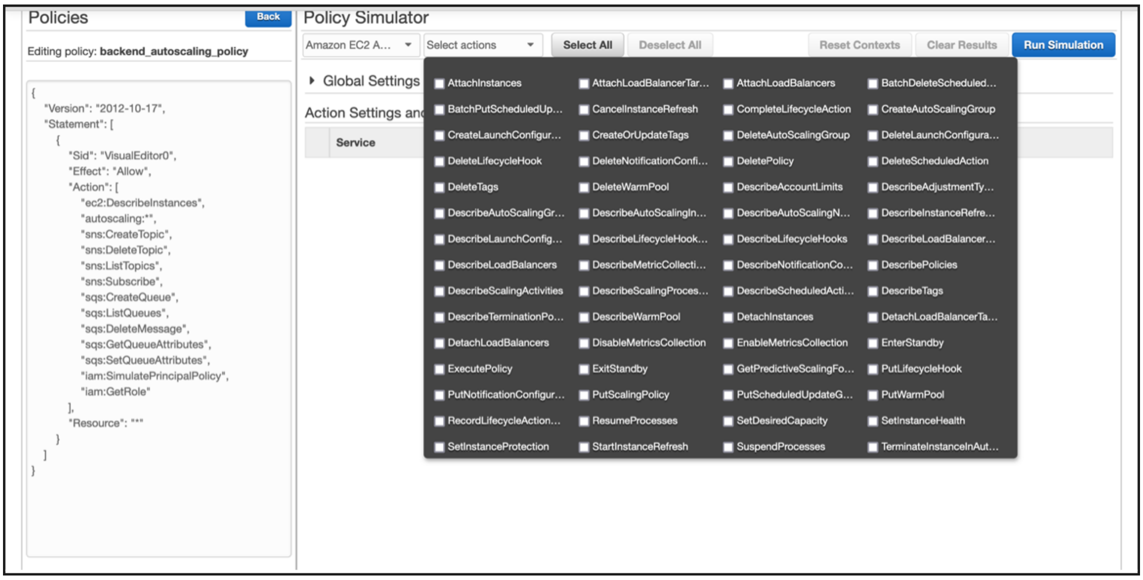



4. After selecting the existing role, select the existing policy under it.

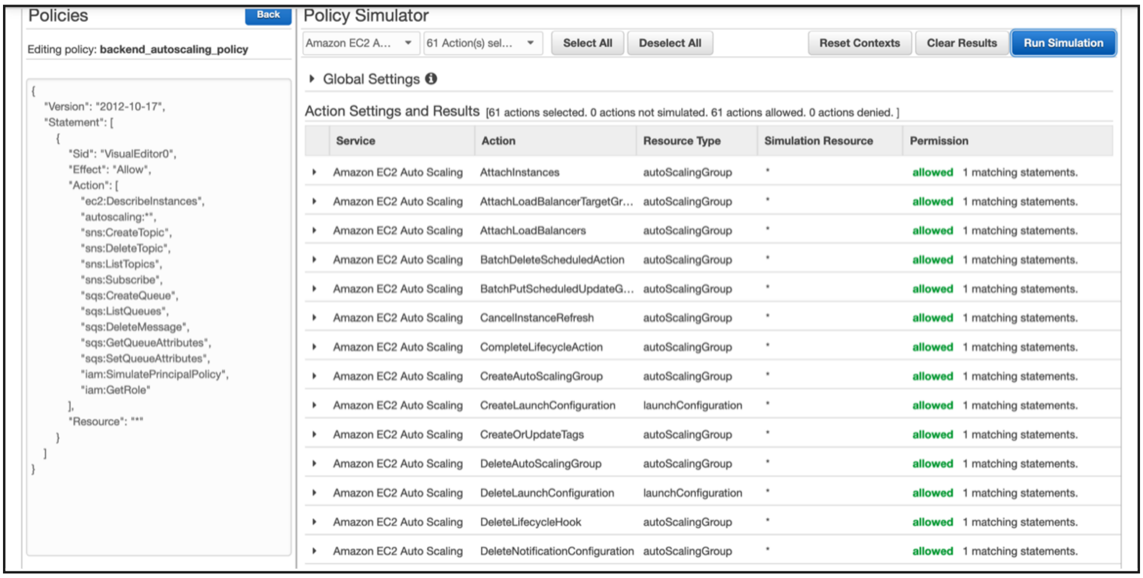



5. After you select the policy, you can see the exact JSON on the left-hand side of the screen. Select the desired actions in the **Select actions** drop-down menu.




6. Click **Run simulation**.




For detailed information, see AWS IAM documentation.

#### Other references

Using an IAM role to grant permissions to applications running on Amazon EC2 instances

## How a NetScaler VPX instance on AWS works

The NetScaler VPX instance is available as an AMI in AWS marketplace, and it can be launched as an EC2 instance within an AWS VPC. The NetScaler VPX AMI instance requires a minimum of 2 virtual CPUs and 2 GB of memory. An EC2 instance launched within an AWS VPC can also provide the multiple interfaces, multiple IP addresses per interface, and public and private IP addresses needed for VPX configuration. Each VPX instance requires at least three IP subnets:

- A management subnet
- A client-facing subnet (VIP)
- A back-end facing subnet (SNIP, MIP, and so on)

Citrix recommends three network interfaces for a standard VPX instance on AWS installation.

AWS currently makes multi-IP functionality available only to instances running within an AWS VPC. A VPX instance in a VPC can be used to load balance servers running in EC2 instances. An Amazon VPC allows you to create and control a virtual networking environment, including your own IP address range, subnets, route tables, and network gateways.

> **Note:**
>
> By default, you can create up to 5 VPC instances per AWS region for each AWS account. You can request higher VPC limits by submitting Amazon's request form http://aws.amazon.com/contact-us/vpc-request.

**Figure 1.** A Sample NetScaler VPX Instance Deployment on AWS Architecture

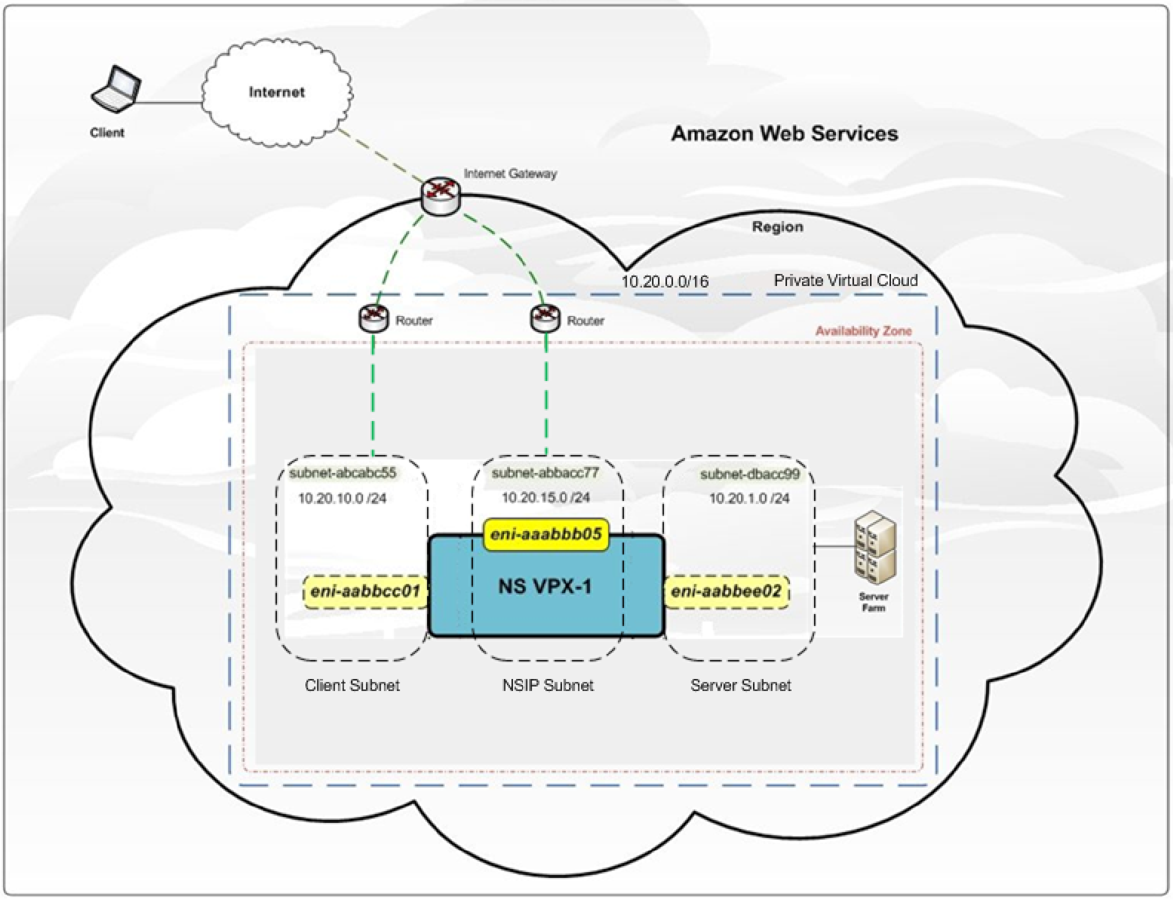



Figure 1 shows a simple topology of an AWS VPC with a NetScaler VPX deployment. The AWS VPC has:

1. A single Internet gateway to route traffic in and out of the VPC.
2. Network connectivity between the Internet gateway and the Internet.
3. Three subnets, one each for management, client, and server.
4. Network connectivity between the Internet gateway and the two subnets (management and client).
5. A standalone NetScaler VPX instance deployed within the VPC. The VPX instance has three ENIs, one attached to each subnet.

# Deploy a NetScaler VPX standalone instance on AWS

You can deploy a NetScaler VPX standalone instance on AWS by using the following options:

- AWS web console
- Citrix-authored CloudFormation template
- AWS CLI

This topic describes the procedure for deploying a NetScaler VPX instance on AWS.

Before you start your deployment, read the following topics:

- Prerequisites
- Limitation and usage guidelines

## Deploy a NetScaler VPX instance on AWS by using the AWS web console

You can deploy a NetScaler VPX instance on AWS through the AWS web console. The deployment process includes the following steps:

1. Create a Key Pair
2. Create a Virtual Private Cloud (VPC)
3. Add more subnets
4. Create security groups and security rules
5. Add route tables
6. Create an internet gateway
7. Create a NetScaler VPX instance
8. Create and attach more network interfaces
9. Attach elastic IPs to the management NIC
10. Connect to the VPX instance

**Step 1: Create a key pair**.

Amazon EC2 uses a key pair to encrypt and decrypt logon information. To log on to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

When you review and launch an instance by using the AWS Launch Instance wizard, you are prompted to use an existing key pair or create a new key pair. More more information about how to create a key pair, see Amazon EC2 Key Pairs.

**Step 2: Create a VPC**.

A NetScaler VPC instance is deployed inside an AWS VPC. A VPC allows you to define the virtual network dedicated to your AWS account. For more information about AWS VPC, see Getting Started With Amazon VPC.

While creating a VPC for your NetScaler VPX instance, keep the following points in mind.

- Use the VPC with a Single Public Subnet Only option to create an AWS VPC in an AWS availability zone.
- Citrix recommends that you create at least **three subnets**, of the following types:
  - One subnet for management traffic. You place the management IP(NSIP) on this subnet. By default elastic network interface (ENI) eth0 is used for management IP.
  - One or more subnets for client-access (user-to-NetScaler VPX) traffic, through which clients connect to one or more virtual IP (VIP) addresses assigned to NetScaler load balancing virtual servers.
  - One or more subnets for the server-access (VPX-to-server) traffic, through which your servers connect to VPX-owned subnet IP (SNIP) addresses. For more information about NetScaler load balancing and virtual servers, virtual IP addresses (VIPs), and subnet IP addresses (SNIPs), see:
  - All subnets must be in the same availability zone.

**Step 3: Add subnets**.

When you used the VPC wizard, only one subnet was created. Depending on your requirement, you might want to create more subnets. For more information about how to create more subnets, see Adding a Subnet to Your VPC.

**Step 4: Create security groups and security rules**.

To control inbound and outbound traffic, create security groups and add rules to the groups. For more information how to create groups and add rules, see Security Groups for Your VPC.

For NetScaler VPX instances, the EC2 wizard gives default security groups, which are generated by AWS Marketplace and is based on recommended settings by Citrix. However, you can create more security groups based on your requirements.

> **Note:**
>
> Port 22, 80, 443 to be opened on the Security group for SSH, HTTP, and HTTPS access respectively.

**Step 5: Add route tables**.

Route table contains a set of rules, called routes, that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table. For more information about how to create a route table, see Route Tables.

**Step 6: Create an internet gateway**.

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

Create an internet gateway for internet traffic. For more information about how to create an Internet Gateway, see the section Attaching an Internet Gateway.

**Step 7: Create a NetScaler VPX instance by using the AWS EC2 service**.

To create a NetScaler VPX instance by using the AWS EC2 service, complete the following steps.

1. From the AWS dashboard, go to **Compute > EC2 > Launch Instance > AWS Marketplace**.

   Before you click **Launch Instance**, ensure your region is correct by checking the note that appears under **Launch Instance**.

2. In the Search AWS Marketplace bar, search with the keyword NetScaler VPX.
3. Select the version you want to deploy and then click **Select**. For the NetScaler VPX version, you have the following options:
   - A licensed version
   - NetScaler VPX Express appliance (This is a free virtual appliance, which is available from NetScaler 12.0 56.20.)
   - Bring your own device

The Launch Instance wizard starts. Follow the wizard to create an instance. The wizard prompts you to:

- Choose Instance Type
- Configure Instance
- Add Storage
- Add Tags
- Configure Security Group
- Review


**Step 8: Create and attach more network interfaces**.

Create two more network interfaces for VIP and SNIP. For more information about how to create more network interfaces, see the Creating a Network Interface section.

After you've created the network interfaces, you must attach them to the VPX instance. Before attaching the interface, shut down the VPX instance, attach the interface, and power on the instance. For more information about how to attach network interfaces, see the Attaching a Network Interface When Launching an Instance section.

**Step 9: Allocate and associate elastic IPs**.

If you assign a public IP address to an EC2 instance, it remains assigned only until the instance is stopped. After that, the address is released back to the pool. When you restart the instance, a new public IP address is assigned.

In contrast, an elastic IP (EIP) address remains assigned until the address is disassociated from an instance.

Allocate and associate an elastic IP for the management NIC. For more information about how to allocate and associate elastic IP addresses, see these topics:

- Allocating an Elastic IP Address
- Associating an Elastic IP Address with a Running Instance

These steps complete the procedure to create a NetScaler VPX instance on AWS. It can take a few minutes for the instance to be ready. Check that your instance has passed its status checks. You can view this information in the **Status Checks** column on the Instances page.

**Step 10: Connect to the VPX instance**.

After you've created the VPX instance, you connect the instance by using the GUI and an SSH client.

- GUI

The following are the default administrator credentials to access a NetScaler VPX instance

User name: `nsroot`

Password: The default password for the ns root account is set to the AWS instance-ID of the NetScaler VPX instance. On your first logon, you are prompted to change the password for security reasons. After changing the password, you must save the configuration. If the configuration is not saved and the instance restarts, you must log on with the default password. Change the password again at the prompt.

- SSH client

From the AWS management console, select the NetScaler VPX instance and click **Connect**. Follow the instructions given on the **Connect to Your Instance** page.

For more information about how to deploy a NetScaler VPX standalone instance on AWS by using the AWS web console, see Scenario: standalone instance

### Configure a NetScaler VPX instance by using the Citrix CloudFormation template

You can use the Citrix-provided CloudFormation template to automate VPX instance launch. The template provides functionality to launch a single NetScaler VPX instance, or to create a high availability environment with a pair of NetScaler VPX instances.

You can launch the template from AWS Marketplace or GitHub.

The CloudFormation template requires an existing VPC environment, and it launches a VPX instance with three elastic network interfaces (ENIs). Before you start the CloudFormation template, ensure that you complete the following requirements:

- An AWS virtual private cloud (VPC)
- Three subnets within the VPC: one for management, one for client traffic, and one for back-end servers
- An EC2 key pair to enable SSH access to the instance
- A security group with UDP 3003, TCP 3009–3010, HTTP, SSH ports open

See the "Deploy a NetScaler VPX Instance on AWS by Using the AWS Web Console"section or AWS documentation for more information about how to complete the prerequisites.

Further, you configure and launch a NetScaler VPX Express standalone instance by using the Citrix CloudFormation template available in GitHub:

https://github.com/citrix/citrix-adc-aws-cloudformation/tree/master/templates/standalone/

An IAM role is not mandatory for a standalone deployment. However, Citrix recommends that you create and attach an IAM role with the required privileges to the instance, for future need. The IAM role ensures that the standalone instance is easily converted to a high availability node with SR-IOV, when required.

For more information about the required privileges, see Configuring NetScaler VPX instances to Use SR-IOV Network Interface.

> **Note:**
>
> If you deploy a NetScaler VPX instance on AWS by using the AWS web console, the CloudWatch service is enabled by default. If you deploy a NetScaler VPX instance by using the Citrix CloudFormation template, the default option is "Yes."If you want to disable the CloudWatch service, select "No."For more information, see Monitor your instances using Amazon CloudWatch

### Configure a NetScaler VPX instance by using the AWS CLI

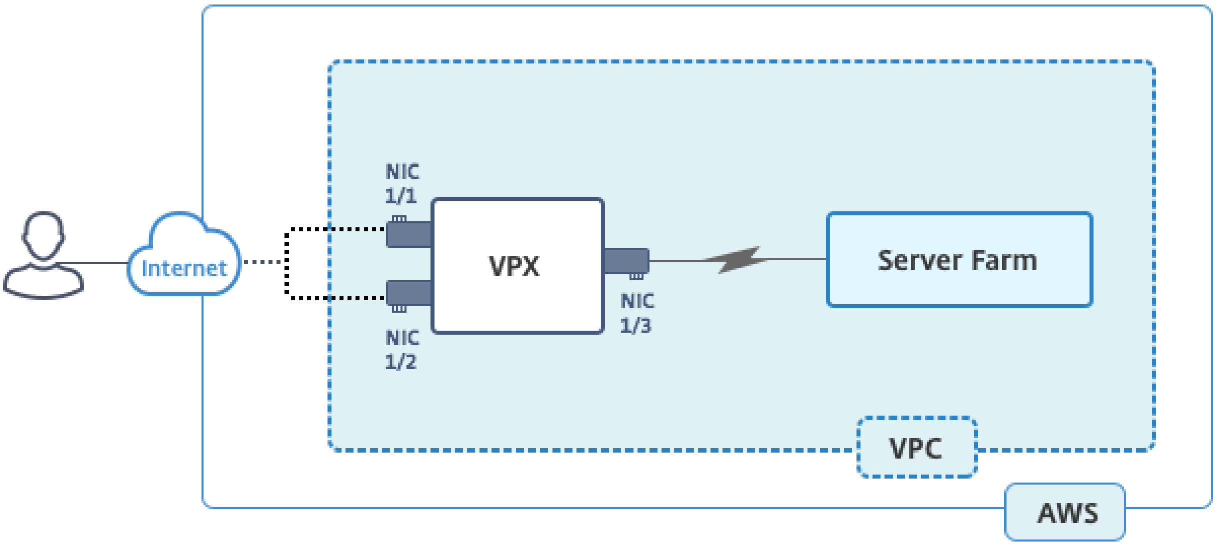
You can use the AWS CLI to launch instances. For more information, see the AWS Command Line Interface Documentation.

## Scenario: standalone instance

This scenario illustrates how to deploy a NetScaler VPX standalone EC2 instance in AWS by using the AWS GUI. Create a standalone VPX instance with three NICs. The instance, which is configured as a load balancing virtual server, communicates with back-end servers (the server farm). For this configuration, set up the required communication routes between the instance and the back-end servers, and between the instance and the external hosts on the public internet.

For more details about the procedure for deploying a VPX instance, see Deploy a NetScaler VPX standalone instance on AWS.




Create three NICs. Each NIC can be configured with a pair of IP addresses (public and private). The NICs serve the following purposes.

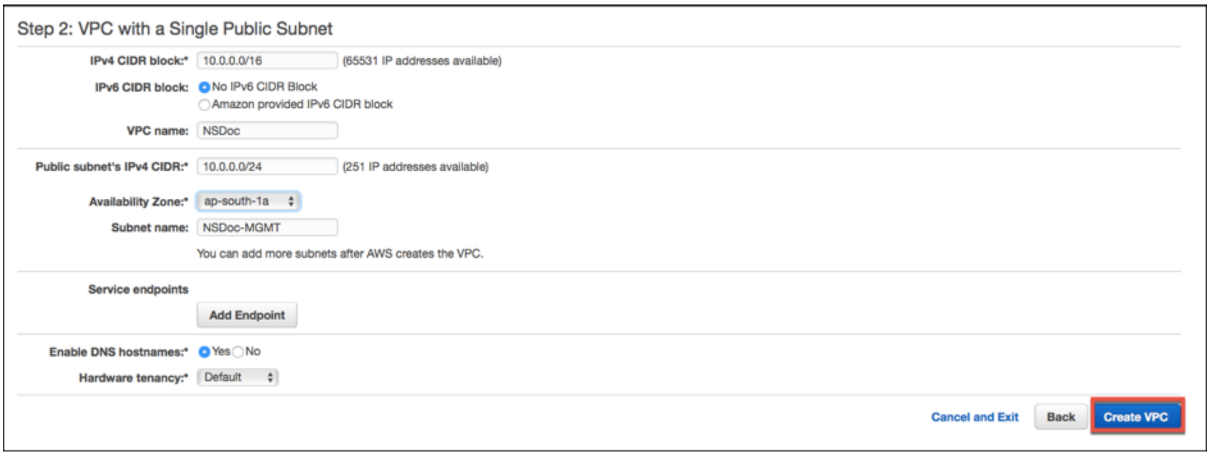
| NIC | Purpose | Associated with |
|---|---|---|
| eth0 | Serves management traffic (NSIP) | A public IP address and a private IP address |
| eth1 | Serves client-side traffic (VIP) | A public IP address and a private IP address |
| eth2 | Communicates with back-end servers (SNIP) | A public IP address (Private IP address not mandatory) |

Step 1: Create a VPC.

1. Log on to the AWS web console and navigate to **Networking & Content Delivery** > **VPC**. Click **Start VPC Wizard**.
2. Select **VPC with a Single Public Subnet** and click **Select**.

3. Set the IP CIDR Block to 10.0.0.0/16, for this scenario.
4. Give a name for the VPC.
5. Set the public subnet to 10.0.0.0/24. (This is the management network).
6. Select an availability zone.
7. Give a name for the subnet.
8. Click Create **VPC**.

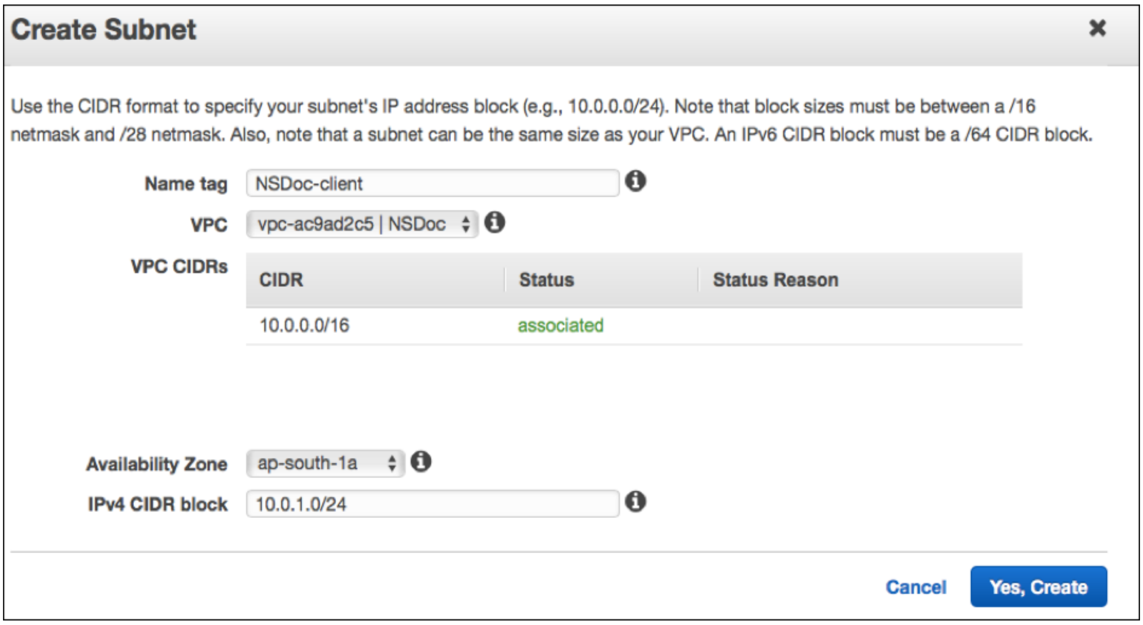



Step 2: Create extra subnets.

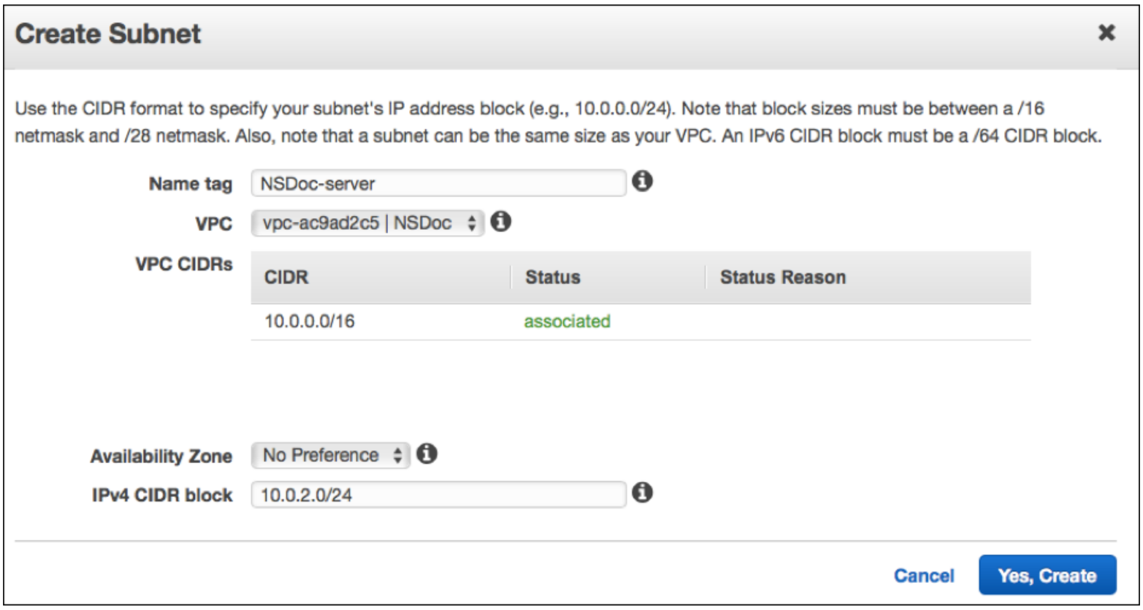
1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Subnets, Create Subnet after you enter the following details.

   - Name tag: Provide a name for your subnet.
   - VPC: Choose the VPC for which you're creating the subnet.
   - Availability Zone: Choose the availability zone in which you created the VPC in step 1.
   - IPv4 CIDR block: Specify an IPv4 CIDR block for your subnet. For this scenario, choose 10.0.1.0/24.



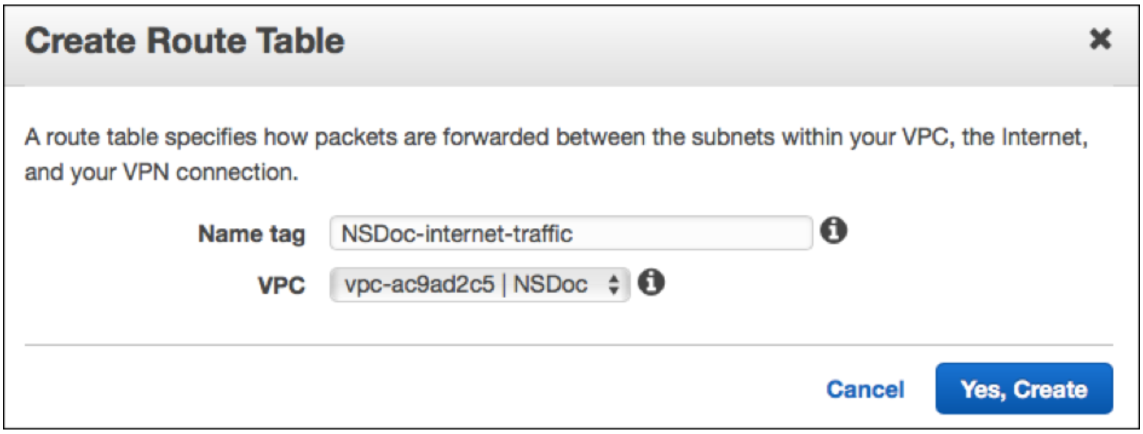

3. Repeat the steps to create one more subnet for back-end servers.




Step 3: Create a route table.

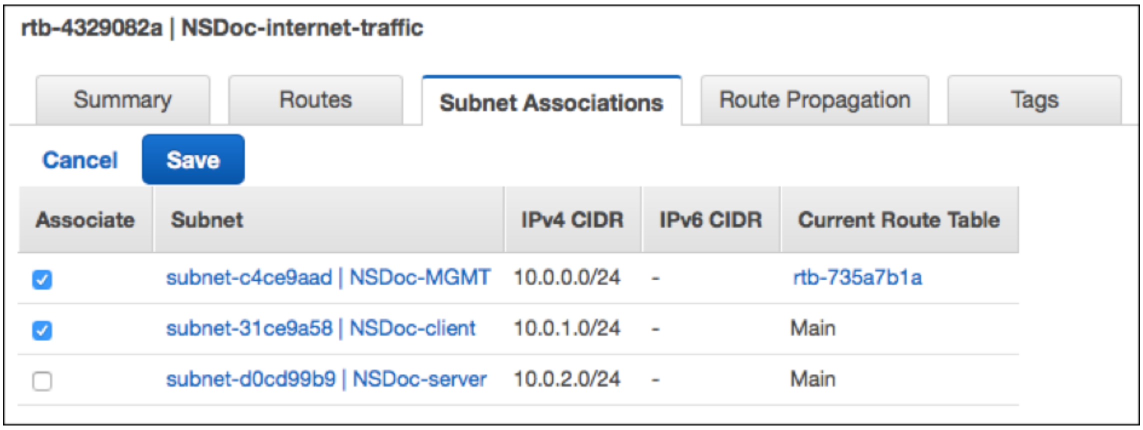
1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route Tables** > **Create Route Table.**
3. In the Create Route Table window, add a name and select the VPC that you created in step 1.
4. Click **Yes, Create**.




The route table is assigned to all the subnets that you created for this VPC, so that routing of traffic from an instance in one subnet can reach an instance in another subnet.

5. Click Subnet Associations, and then click **Edit**.
6. Click the management and client subnet and click **Save**. This creates a route table for internet traffic only.




7. Click **Routes > Edit > Add another route**.
8. In the Destination field add 0.0.0.0/0, and click the Target field to select igw-<xxxx> the Internet Gateway that the VPC Wizard created automatically.
9. Click **Save**.

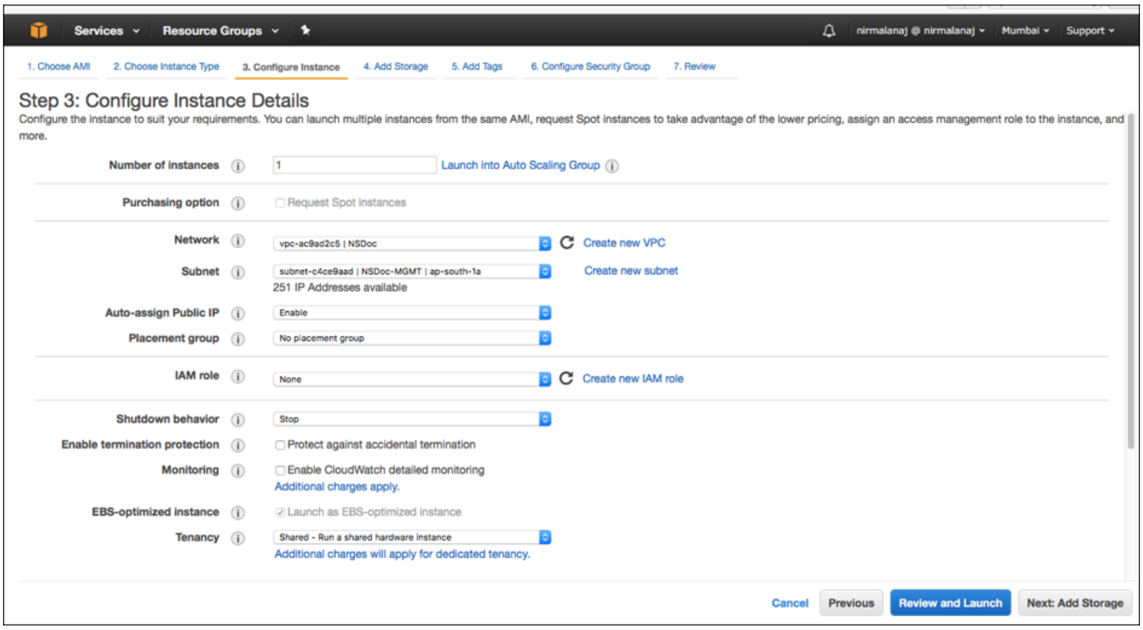

10. Follow the steps to create a route table for server-side traffic.

Step 4: Create a NetScaler VPX instance.

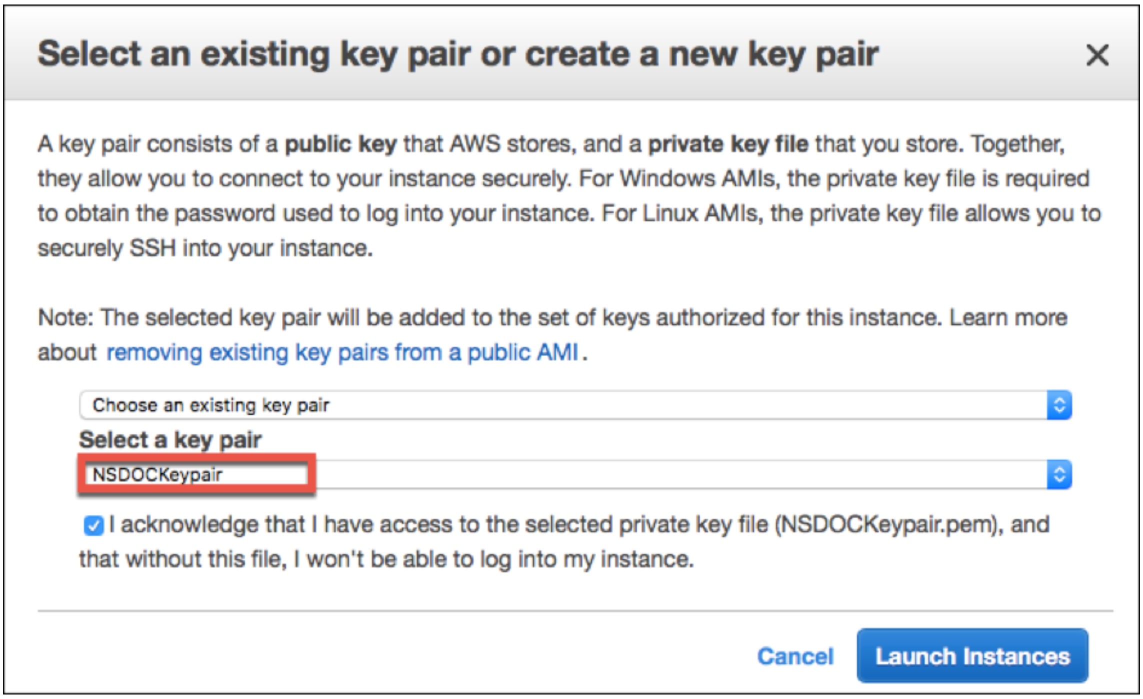
1. Log on the AWS management console and click **EC2** under **Compute**.
2. Click AWS Marketplace. In the Search AWS Marketplace bar, type NetScaler VPX and press Enter. The available NetScaler VPX editions are displayed.
3. Click **Select** to choose the desired NetScaler VPX edition. The EC2 instance wizard starts.
4. In the **Choose Instance Type** page, select **m4. Xlarge** (recommended) and click **Next: Configure Instance Details**.
5. In the Configure Instance Details page, select the following, and then click Next: Add Storage.

    - Number of instances: 1
    - Network: the VPC that created in Step 1
    - Subnet: the management subnet
    - Auto-assign Public IP: Enable




6. In the Add Storage page, select the default option, and click Next: Add Tags.
7. In the Add Tags page, add a name for the instance, and click Next: Configure Security Group.
8. In the Configure Security Group page, select the default option (which is generated by AWS Marketplace and is based on recommended settings by Citrix Systems) and then click **Review and Launch > Launch**.
9. You are prompted to select an existing key pair or create and new key pair. From the Select a key pair drop-down list, select the key pair that you created as a prerequisite (See the Prerequisite section.)
10. Check the box to acknowledge the key pair and click Launch Instances.




Launch Instance Wizard displays the Launch Status, and the instance appears in the list of instances when it is fully launched.

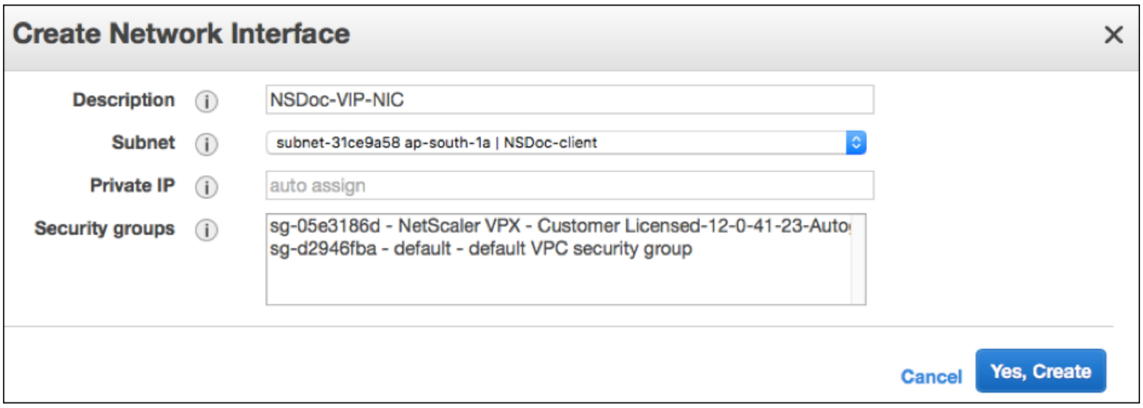
The check instance, go the AWS console click EC2 > Running Instances. Select the instance and add a name. Make sure the Instance State is running and Status Checks is complete.

Step 5: Create and attach more network interfaces.

When you created the VPC, only one network interface associated with it. Now add two more network interfaces to the VPC, for the VIP and SNIP.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Choose Create Network Interface.
4. For Description, enter a descriptive name.
5. For Subnet, select the subnet that you created previously for the VIP.
6. For Private IP, leave the default option.
7. For Security groups, select the group.
8. Click **Yes, Create**.

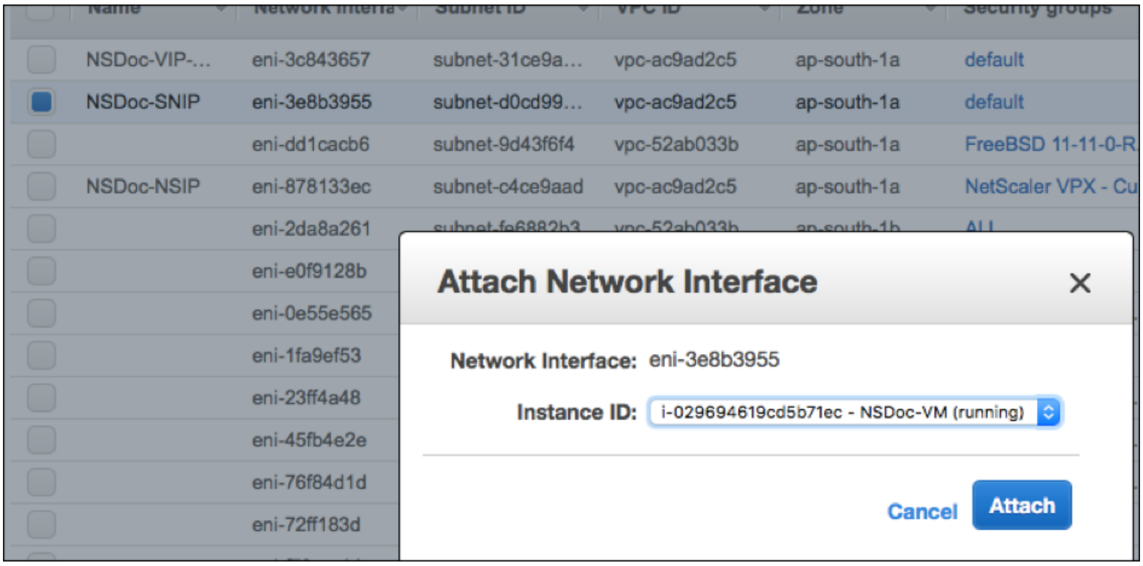
 




9. After the network interface is created, add a name to the interface.
10. Repeat the steps to create a network interface for server-side traffic.

**Attach the network interfaces**:

1. In the navigation pane, choose Network Interfaces.
2. Select the network interface and choose Attach.
3. In the Attach Network Interface dialog box, select the instance and choose Attach.




Step 6: Attach an elastic IP to the NSIP.

1. From the AWS management console, go to **NETWORK & SECURITY > Elastic IPs**.
2. Check for available free EIP to attach. If none, click **Allocate new address**.
3. Select the newly allocated IP address and choose **Actions > Associate address**.
4. Click the **Network interface** radio button.
5. From the Network interface drop-down list, select the management NIC.

6. From the **Private IP** drop-down menu, select the AWS-generated IP address.
7. Select the **Reassociation** check box.
8. Click **Associate**.


**Access the VPX instance**:

After you've configured a standalone NetScaler VPX instance with three NICs, log on to the VPX instance to complete the NetScaler-side configuration. Use of the following options:

- GUI: Type the public IP of the management NIC in the browser. Log on by using `nsroot` as the user name and the instance ID (i-0c1ffe1d987817522) as the password.

> **Note:**
>
> On your first logon, you are prompted to change the password for security reasons. After changing the password, you must save the configuration. If the configuration is not saved and the instance restarts, you must log on with the default password. Change the password again at the prompt and save the configuration.

- SSH: Open an SSH client and type:

```
ssh -i \<location of your private key\> ns root@\<public DNS of the
instance\>
```

To find the public DNS, click the instance, and click **Connect**.

**Related information**:

- To configure the NetScaler-owned IP addresses (NSIP, VIP, and SNIP), see Configuring NetScaler-Owned IP Addresses.
- You've configured a BYOL version of the NetScaler VPX appliance, for more information see the VPX Licensing Guide at https://support.citrix.com/s/article/CTX255959-how-to-allocate-and-install-citrix-netscaler-vpx-licenses?language=en_US

## Download a NetScaler VPX license

After the launch of NetScaler VPX-customer licensed instance from the AWS marketplace, a license is required. For more information on VPX licensing, see Licensing overview.

You have to:

1. Use the licensing portal within the Citrix website to generate a valid license.
2. Upload the license to the instance.

If this is a **paid** marketplace instance, then you do not need to install a license. The correct feature set and performance activate automatically.

If you use a NetScaler VPX instance with a model number higher than VPX 5000, the network throughput might not be the same as specified by the instance's license. However, other features, such as SSL throughput and SSL transactions per second, might improve.

5 Gbps network bandwidth is observed in the `c4.8xlarge` instance type.

### How to migrate the AWS subscription to BYOL

This section describes the procedure to migrate from AWS subscription to Bring your own license (BYOL), and conversely.

Do the following steps to migrate an AWS subscription to BYOL:

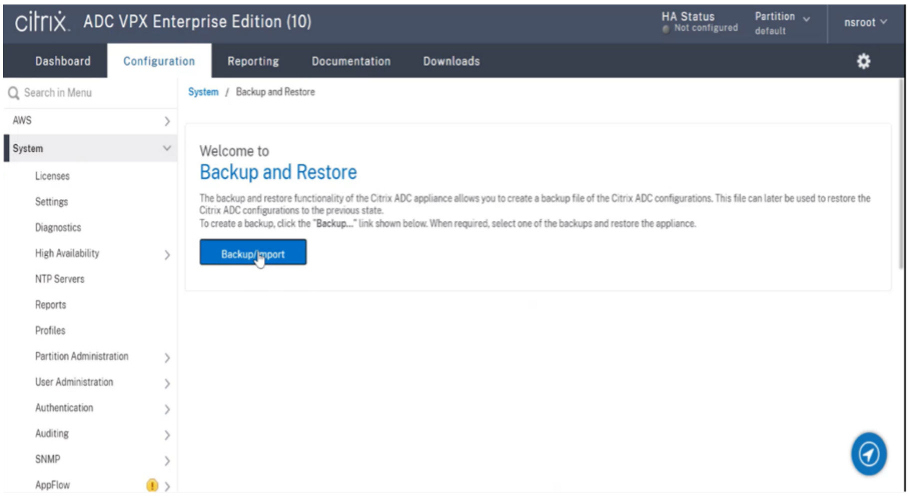
> **Note:**
>
> The **Step 2** and **Step 3** are done on the NetScaler VPX instance, and all other steps are done on the AWS portal.

1. Create a BYOL EC2 instance using NetScaler VPX - Customer Licensed in the same availability zone as the old EC2 instance that has the same security group, IAM role, and subnet. The new EC2 instance must have only one ENI interface.
2. To back up the data on the old EC2 instance using the NetScaler GUI, follow these steps.

    a) Navigate to **System > Backup and Restore**.

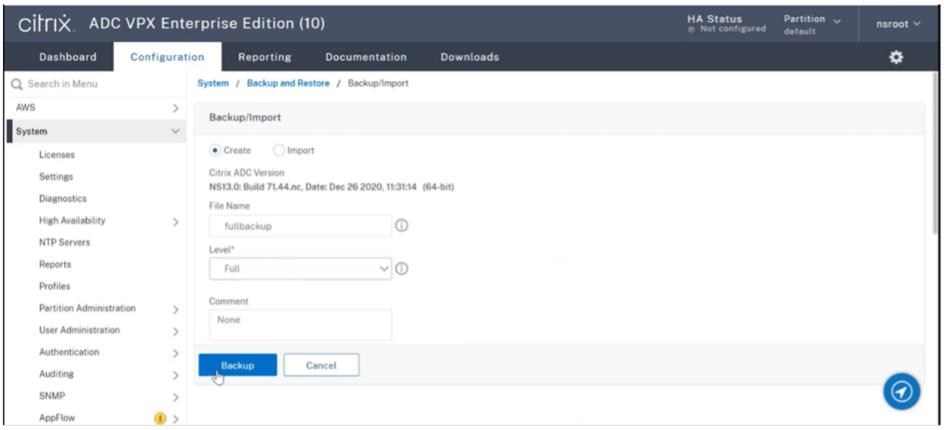
    b) In the **Welcome** page, click **Backup/Import** to start the process.



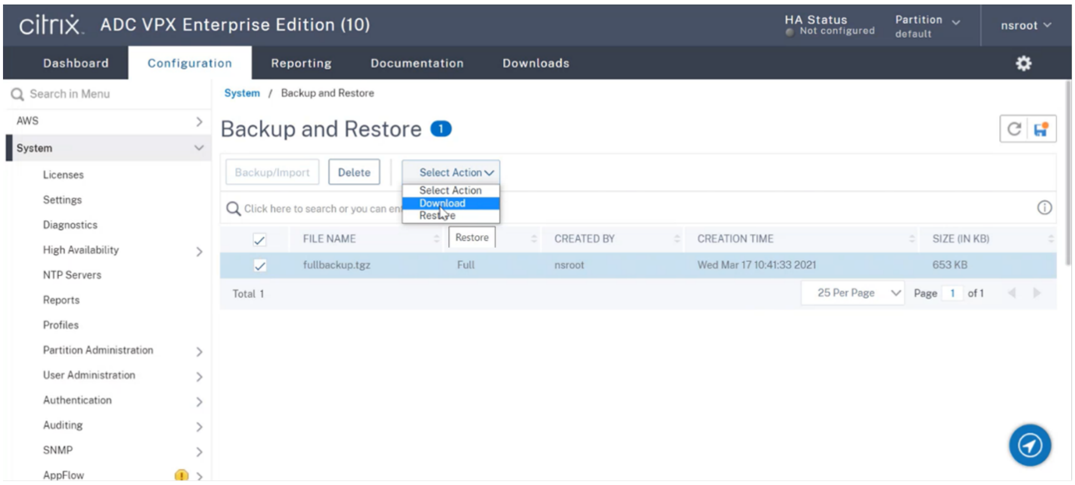

c) In the **Backup/Import** page, fill in the following details:

- **Name** –Name of the backup file.
- **Level** –Select the backup level as **Full**.
- **Comment** –Provide a brief description of the backup.




d) Click **Backup**. Once the backup is complete, you can select the file and download it to your local machine.

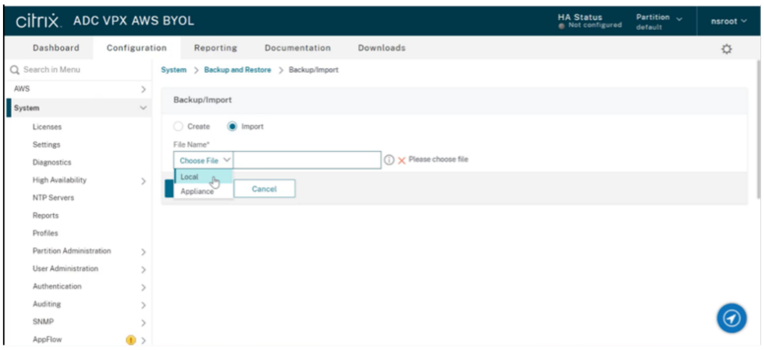



3. To restore the data on the new EC2 instance using the NetScaler GUI, follow these steps:

   a) Navigate to **System > Backup and Restore**.

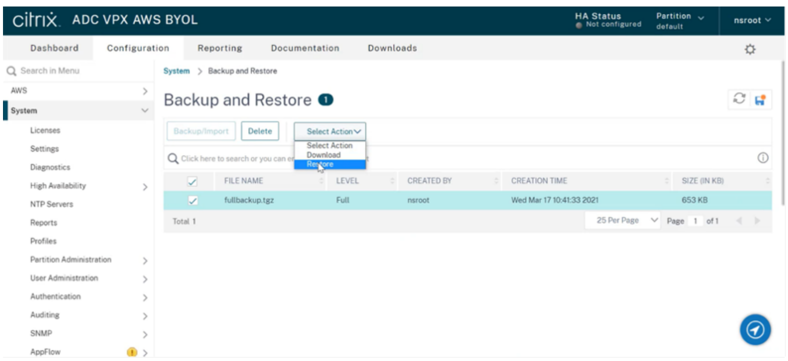
   b) Click **Backup/Import** to start the process.

   c) Select the **Import** option and upload the backup file.



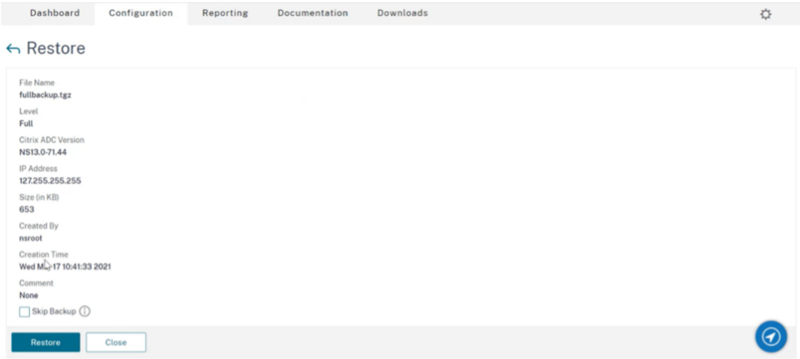

   d) Select the file.

   e) From the **Select Action** drop-down menu, select **Restore**.




   f) On the **Restore** page, verify the file details, and click **Restore**.



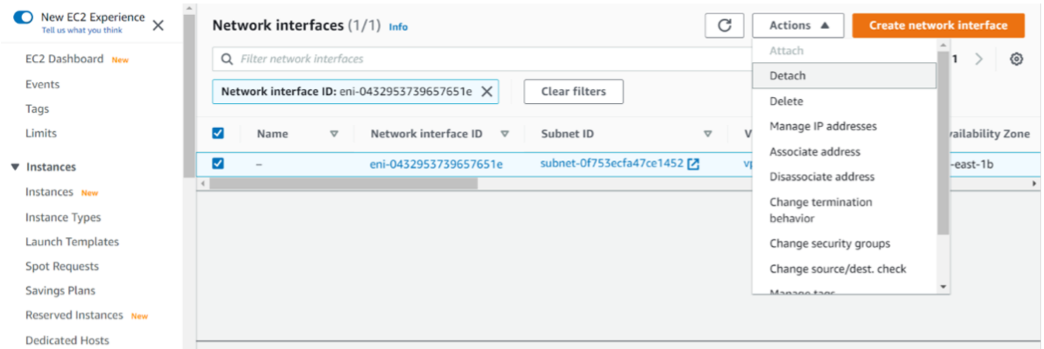

g) After the restore, reboot the EC2 instance.

4. Move all interfaces (except the management interface to which the NSIP address is bound) from the old EC2 instance to the new EC2 instance. To move a network interface from one EC2 instance to another, follow these steps:

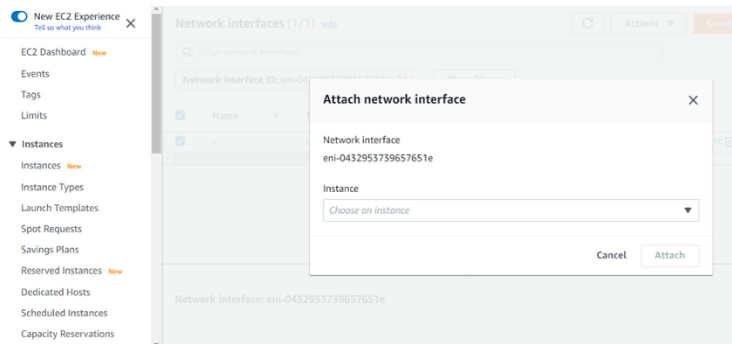
   a) In the **AWS Portal**, stop both the old and new EC2 instances.

   b) Navigate to **Network Interfaces**, and select the network interface attached to the old EC2 instance.

   c) Detach the EC2 instance by clicking **Actions > Detach**.



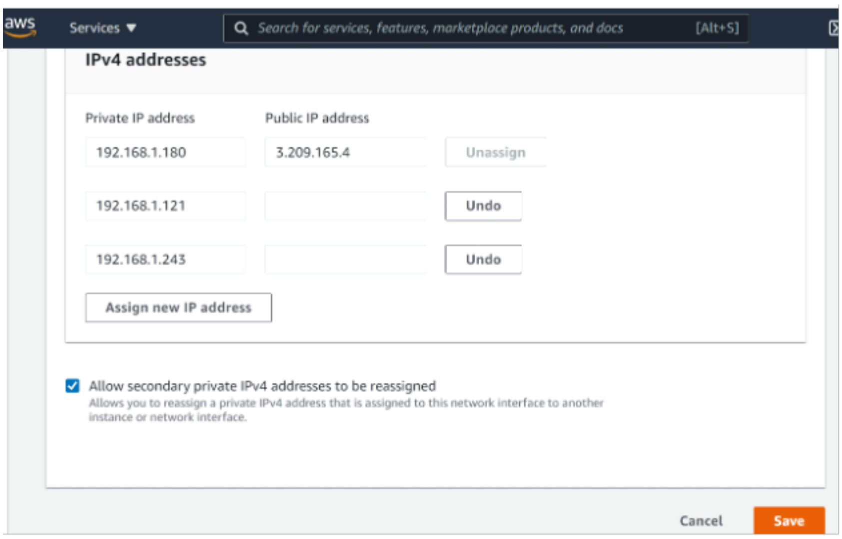

   d) Attach the network interface to the new EC2 instance by clicking **Actions > Attach**. Enter the EC2 instance name to which the network interface must be attached.




   e) Do the **Step 1** to **Step 4** for all other interfaces that are attached. Make sure to follow the sequence and maintain the interface order. That is, first detach interface 2 and attach it, and then detach interface 3 and attach it, and so on.

5. You can't detach the management interface from an old EC2 instance. So, move all the secondary IP addresses (if any) on the management interface (primary network interface) of the old EC2 instance to the new EC2 instance. To move an IP address from one interface to another, follow these steps:

    a) In the **AWS Portal**, make sure that both the old and new EC2 instances are in **Stop** state.
    b) Navigate to **Network Interfaces**, and select the management network interface attached to the old EC2 instance.
    c) Click **Actions > Manage IP Address**, and make note of all the secondary IP addresses assigned (if any).
    d) Navigate to the management network interface or primary interface of the new EC2 instance.
    e) Click **Actions > Manage IP Addresses**.
    f) Under **IPv4 Addresses**, click **Assign new IP address**.
    g) Enter the IP addresses, which are noted in the **Step 3**.
    h) Select **Allow secondary private IP addresses to be reassigned** check box.
    i) Click **Save**.




6. Start the new EC2 instance and verify the configuration. After all the configuration is moved, you can delete or keep the old EC2 instance as per your requirement.
7. If any EIP address is attached to the NSIP address of the old EC2 instance, move the old instance NSIP address to the new instance NSIP address.

8. If you want to revert to the old instance, then follow the same steps in the opposite way between the old and new instance.
9. After you move from subscription instance to BYOL instance, a license is required. To install a license follow these steps:

    - Use the licensing portal in the Citrix website to generate a valid license.
    - Upload the license to the instance.

> **Note:**
>
> When you move BYOL instance to subscription instance (paid marketplace instance), you need not install the license. The correct feature set and performance is automatically activated.

**Limitations**

The management interface can't be moved to the new EC2 instance. So Citrix recommends you manually configure the management interface. For more information, see **Step 5** in the preceding procedure. A new EC2 instance is created with the exact replica of the old EC2 instance but only the NSIP address has a new IP address.

## Load balancing servers in different availability zones

A VPX instance can be used to load balance servers running in the same availability zone, or in:

- A different availability zone (AZ) in the same AWS VPC
- A different AWS region
- AWS EC2 in a VPC

To enable a VPX instance to load balance servers running outside the AWS VPC that the VPX instance is in, configure the instance to use EIPs to route traffic through the Internet gateway, as follows:

1. Configure a SNIP on the NetScaler VPX instance by using the NetScaler CLI or the GUI.
2. Enable traffic to be routed out of the AZ, by creating a public facing subnet for the server-side traffic.
3. Add an Internet gateway route to the routing table, using the AWS GUI console.
4. Associate the routing table you updated with the server-side subnet.
5. Associate an EIP with the server-side private IP address that is mapped to a NetScaler SNIP address.

## How high availability on AWS works

You can configure two NetScaler VPX instances on AWS as a high availability (HA) active-passive pair. When you configure one instance as the primary node and the other as the secondary node, the primary node accepts connections and manages servers. The secondary node monitors the primary. If for any reason, the primary node is unable to accept connections, the secondary node takes over.

In AWS, the following deployment types are supported for VPX instances:

- High availability within same zone
- High availability across different zones

> **Note:**
>
> For high availability to work, ensure that both the NetScaler VPX instances are attached with IAM roles and assigned with the Elastic IP (EIP) address to the NSIP. You need not assign an EIP on NSIP if the NSIP can reach internet through the NAT instance.

### High availability within the same zones

In a high-availability deployment within the same zones, both VPX instances must have similar networking configurations.

Follow these two rules:

Rule 1. Any NIC on one VPX instance must be in the same subnet as the corresponding NIC in the other VPX. Both instances must have:

- Management interface on the same subnet (referred as management subnet)
- Client interface on the same subnet (referred as client subnet)
- Server interface on the same subnet (referred as server subnet)

Rule 2. Sequence of mgmt NIC, client NIC, and server NIC on both instances must be the same.
For example, the following scenario is not supported.

VPX instance 1

NIC 0: management
NIC 1: client
NIC 2: Server

VPX instance 2

NIC 0: management

NIC 1: server

NIC 2: client

In this scenario, NIC 1 of instance 1 is in client subnet while NIC 1 of instance 2 is in server subnet. For HA to work, NIC 1 of both the instances must be either in the client subnet or in the server subnet.

From 13.0 41.xx, high availability can be achieved by migrating secondary private IP addresses attached to the NICs (client and server-side NICs) of the primary HA node to the secondary HA node after failover. In this deployment:

- Both the VPX instances have the same number of NICs and subnet mapping according to NIC enumeration.
- Each VPX NIC has one extra private IP address, except the first NIC - which corresponds to the management IP address. The extra private IP address appears as the primary private IP address in the AWS web console. In our document, we refer to this extra IP address as the dummy IP address).
- The dummy IP addresses must be not configured on the NetScaler instance as VIP and SNIP.
- Other secondary private IP addresses must be created, as required, and configured as VIP and SNIP.
- On failover, the new primary node looks for configured SNIPs and VIPs and moves them from NICs attached to the previous primary to corresponding NICs on the new primary.
- NetScaler instances require IAM permissions for HA to work. Add the following IAM privileges to the IAM policy added to each instance.

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeNetworkInterfaces"
"ec2:AssignPrivateIpAddresses"
```

> **Note:**
>
> `unassignPrivateIpAddress` is not required.

This method is faster than the legacy method. In the older method, HA depends on the migration of AWS elastic network interfaces of the primary node to the secondary node.

For a legacy method, the following policies are required:

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

For more information, see Deploy a high availability pair on AWS.

### High availability across different zones

You can configure two NetScaler VPX instances on two different subnets or two different AWS availability zones, as a high availability active-passive pair in Independent Network Configuration (INC) mode. Upon failover, the EIP (Elastic IP) of the VIP of the primary instance migrates to the secondary, which takes over as the new primary. In the failover process, the AWS API:

- Checks the virtual servers that have `IPSets` attached to them.
- Finds the IP address that has an associated public IP, from the two IP addresses the virtual server is listening on. One that is directly attached to the virtual server, and one that is attached through the IP set.
- Reassociates the public IP (EIP) to the private IP belonging to the new primary VIP.

For HA across different zones, the following policies are required:

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

For more information, see High availability across AWS availability zones.

### Before you start your deployment

Before you start any HA deployment on AWS, read the following document:

- Prerequisites
- Limitations and usage guidelines
- Deploy a NetScaler VPX instance on AWS
- High Availability

### Troubleshooting

To troubleshoot any failure during a HA failover of NetScaler VPX instance on AWS cloud, check the `cloud-ha-daemon.log` file stored in the `/var/log/` location.

## Deploy a VPX HA pair in the same AWS availability zone

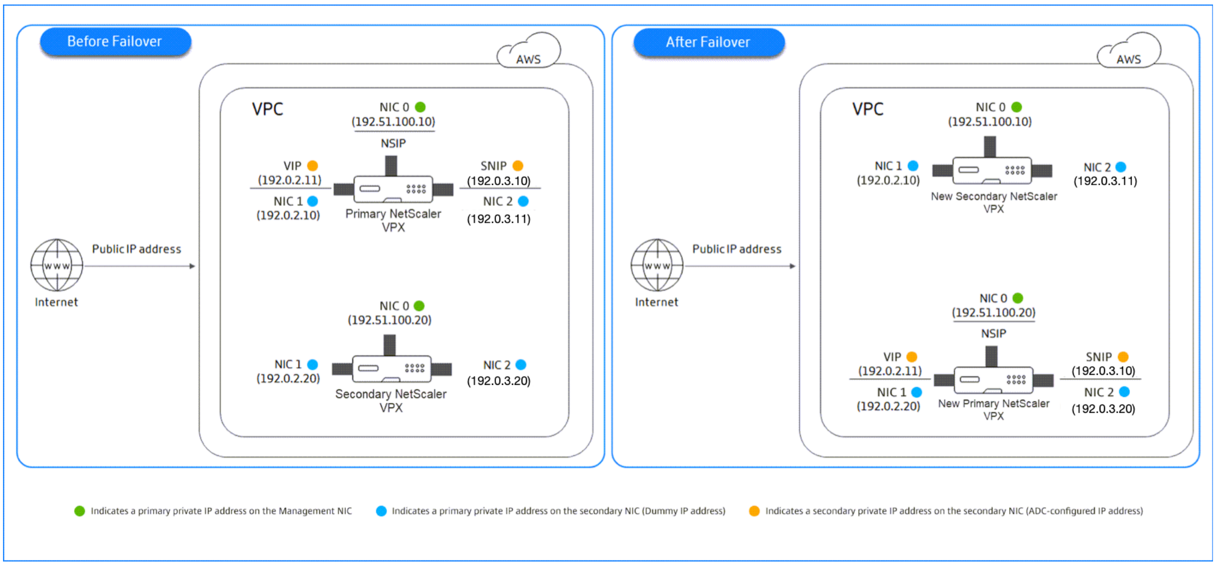
> **Note:**
>
> From NetScaler release 13.1 build 27.x onwards, the VPX HA pair in the same AWS availability zone supports IPv6 addresses.

You can configure two NetScaler VPX instances on AWS as a HA pair, in the same AWS zone where both VPX instances are on the same subnet. HA is achieved by migrating secondary private IP addresses attached to the NICs (client and server-side NICs) of the primary HA node to the secondary HA node after failover. All the Elastic IP addresses associated with the secondary private IP addresses are also migrated.

The NetScaler VPX HA pair supports both IPv4 and IPv6 addresses in the same AWS availability zone.

The following illustration depicts an HA failover scenario by migrating secondary private IP addresses.

Figure 1. A NetScaler VPX HA Pair on AWS, using private IP migration




Before you start your document, read the following docs:

- Prerequisites
- Limitations and usage guidelines
- Deploy a NetScaler VPX instance on AWS
- High Availability

### How to deploy a VPX HA pair in the same zone

Here is the summary of the steps to deploy a VPX HA pair in the same zone:

1. Create two VPX instances on AWS, each with three NICs

2. Assign AWS secondary private IP address to VIP and SNIP of primary node
3. Configure VIP and SNIP on primary node using AWS secondary private IP addresses
4. Configure HA on both nodes

#### Step 1. Create two VPX instances (primary and secondary nodes) by using the same VPC, each with three NICs (Ethernet 0, Ethernet 1, Ethernet 2)

Follow the steps given in Deploy a NetScaler VPX instance on AWS by using the AWS web console.

#### Step 2. On the primary node, assign private IP addresses for Ethernet 1 (client IP or VIP) and Ethernet 2 (back-end server IP or SNIP)

The AWS console automatically assigns primary private IP addresses to the configured NICs. Assign more private IP addresses to VIP and SNIP, known as secondary private IP addresses.

To assign a private IP address to a network interface, follow these steps:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces** and then select the network interface attached to the instance.
3. Choose **Actions > Manage IP Addresses**.
4. Select **IPv4 Addresses** or **IPv6 Addresses** based on your requirement.
5. For IPv4 Addresses:

    a) Choose **Assign new IP**.
    b) Enter a specific IPv4 address that's within the subnet range for the instance, or leave the field blank to let Amazon select an IP address for you.
    c) (Optional) Choose **Allow reassignment** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.

6. For IPv6 Addresses:

    a) Choose **Assign new IP**.
    b) Enter a specific IPv6 address that's within the subnet range for the instance, or leave the field blank to let Amazon select an IP address for you.
    c) (Optional) Choose **Allow reassignment** to allow the primary or secondary private IP address to be reassigned if it is already assigned to another network interface.

7. Choose **Yes > Update**.

Under the **Instance description**, the assigned private IP addresses appear.

> **Note:**
>
> In an IPv4 HA pair deployment, you can assign only the secondary IPv4 addresses on the interface and use them as VIP and SNIP addresses. But in an IPv6 HA pair deployment, you can assign either the primary IPv6 or secondary IPv6 addresses on the interface and use them as VIP and SNIP addresses.

#### Step 3. Configure VIP and SNIP on the primary node, using secondary private IP addresses

Access the primary node using SSH. Open an ssh client and type:

```
ssh -i <location of your private key> nsroot@<public DNS of the
    instance>
```

Next, configure VIP and SNIP.

For VIP, type:

```
add ns ip <IPAddress> <netmask> -type <type>
```

For SNIP, type:

```
add ns ip <IPAddress> <netmask> -type SNIP
```

Type `save config` to save.

To see the configured IP addresses, type the following command:

```
show ns ip
```

For more information, see the following topics:

- Configuring and Managing Virtual IP (VIP) Addresses
- Configuring the NSIP address

#### Step 4: Configure HA on both instances

On the primary node, open a Shell client and type the following command:

```
add ha node <id> <private IP address of the management NIC of the
    secondary node>
```

On the secondary node, type the following command:

```
add ha node <id> < private IP address of the management NIC of the
    primary node >
```

Type `save config` to save the configuration.

To see the configured HA nodes, type `show ha node`.

Upon failover, the secondary private IP addresses configured as VIP and SNIP on the previous primary node are migrated to the new primary node.

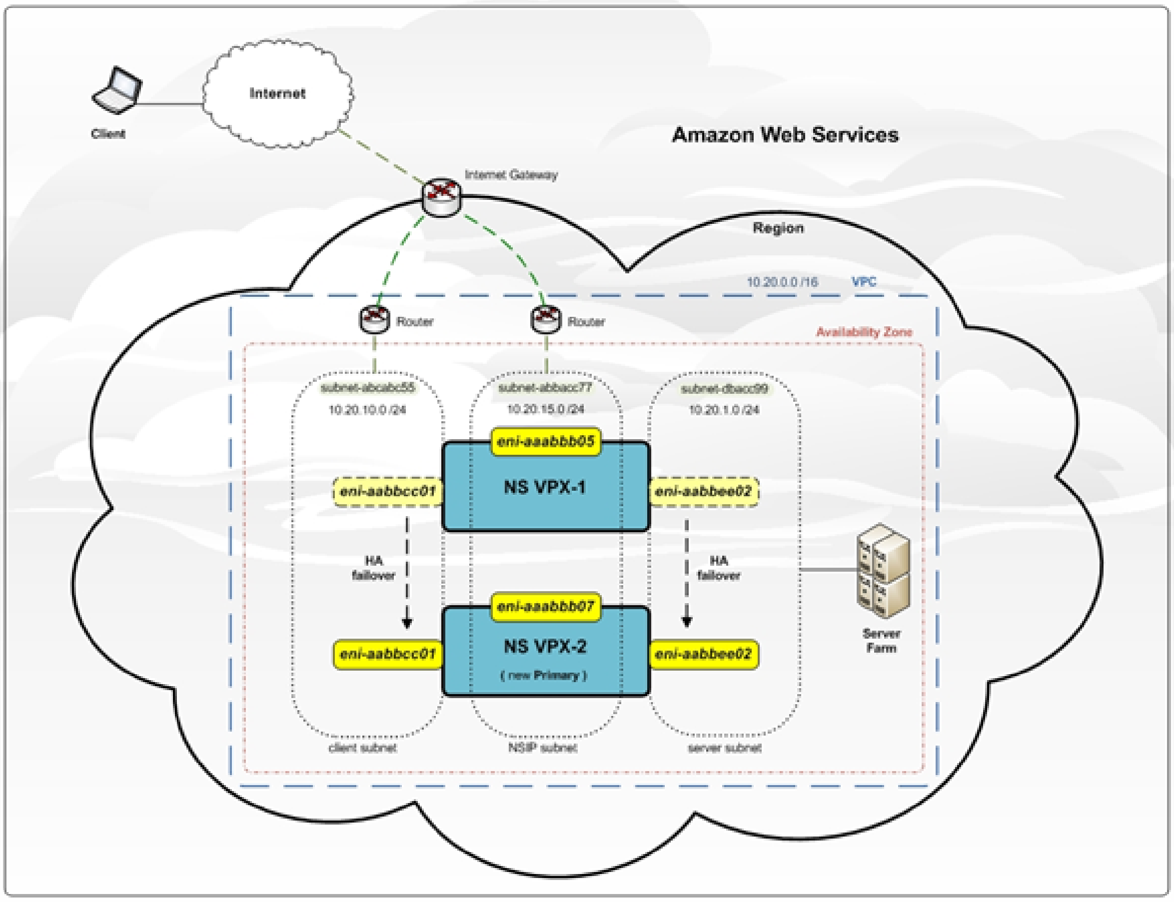
To force a failover on a node, type `force HA`failover.

### Legacy method for deploying a VPX HA pair

Before 13.0 41.x release, HA within the same zone was achieved through AWS elastic network interface (ENI) migration. However, this method is slowly deprecated.

The following figure shows an example of the HA deployment architecture for NetScaler VPX instances on AWS.

Figure 1. A NetScaler VPX HA Pair on AWS, using ENI migration




You can deploy two VPX instances on AWS as an HA pair by using one of the following options:

- Create the instances with IAM Role manually by using the AWS Management Console and then configure HA on them.
- Or automate the high availability deployment by using the Citrix CloudFormation template.

The CloudFormation template significantly decreases the number of steps involved for creating an HA pair, and it automatically creates an IAM Role. This section shows how to deploy a NetScaler VPX HA (active-passive) pair by using the Citrix CloudFormation template.

Keep the following points in mind while deploying two NetScaler VPX instances as an HA pair.

### Points to note

- HA on AWS requires the primary node to have at least two ENIs (one for management and the other for data traffic), and the secondary node to have one management ENI. However, for security purposes, create three ENIs on the primary node, because this setup allows you to segregate the private and public network (recommended).
- The secondary node always has one ENI interface (for management) and the primary node can have up to four ENIs.
- The NSIP addresses for each VPX instance in a high availability pair must be configured on the default ENI of the instance.
- Amazon does not allow any broadcast/multicast packets in AWS. As a result, in a HA setup, data-plane ENIs are migrated from the primary to the secondary VPX instance when the primary VPX instance fails.
- Because the default (management) ENI cannot be moved to another VPX instance, do not use the default ENI for client and server traffic (data-plane traffic).
- The message AWSCONFIG IOCTL NSAPI_HOTPLUG_INTF success output 0 in the /var/log/ns.log indicates that the two data ENIs have successfully attached to the secondary instance (the new primary).
- Failover might take up to 20 seconds due to the AWS detach/attach ENI mechanism.
- Upon failover, the failed instance always restarts.
- The heartbeat packets are received only on the management interface.
- The configuration file of the primary and secondary VPX instances is synchronized, including the `nsroot` password. The `nsroot` password of the secondary node is set to that of the primary node after the HA configuration synchronization.
- To have access to the AWS API servers, either the VPX instance must have a public IP address assigned or routing must be set up correctly at VPC subnet level pointing to the internet gateway of the VPC.
- Nameservers/DNS servers are configured at VPC level using DHCP options.
- The Citrix CloudFormation template does not create an HA setup between different availability zones.

- The Citrix CloudFormation template does not create an INC mode.
- The AWS debug messages are available in the log file, /var/log/ns.log, on the VPX instance.

### Deploy a high availability pair by using the Citrix CloudFormation template

Before starting the CloudFormation template, ensure that you complete the following requirements:

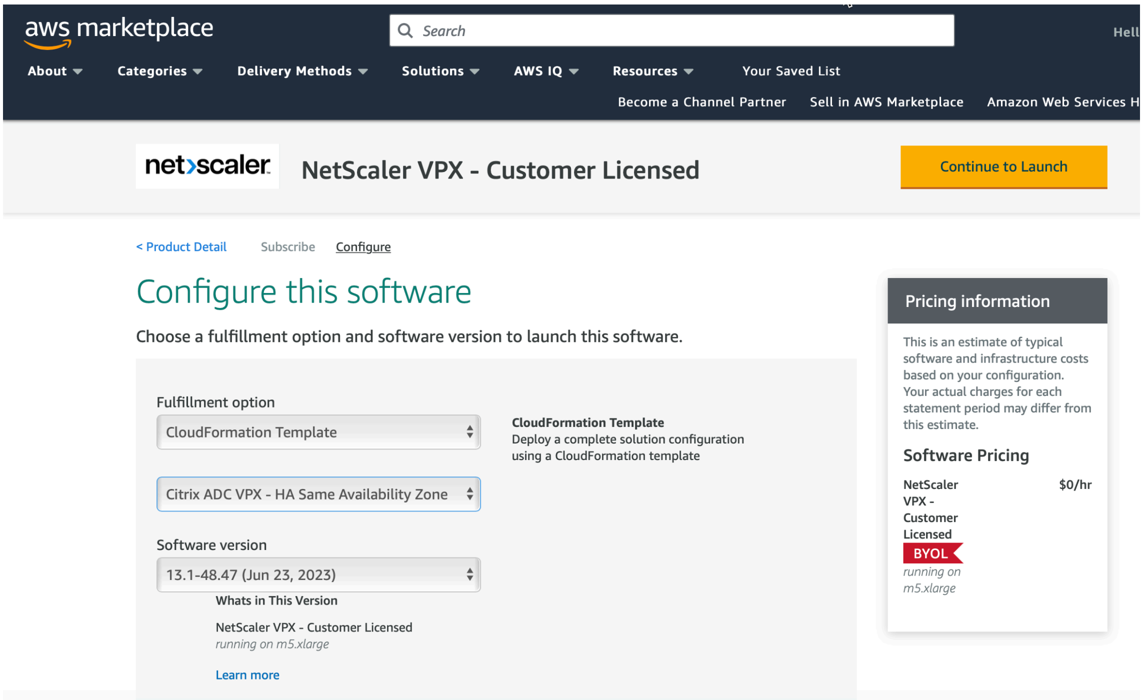
- A VPC
- Three subnets within the VPC
- A security group with UDP 3003, TCP 3009–3010, HTTP, SSH ports open
- A key pair
- Create an internet gateway
- Edit route tables for client and management networks to point to the internet gateway

> **Note**
>
> The Citrix CloudFormation template automatically creates an IAM Role. Existing IAM Roles do not appear in the template.

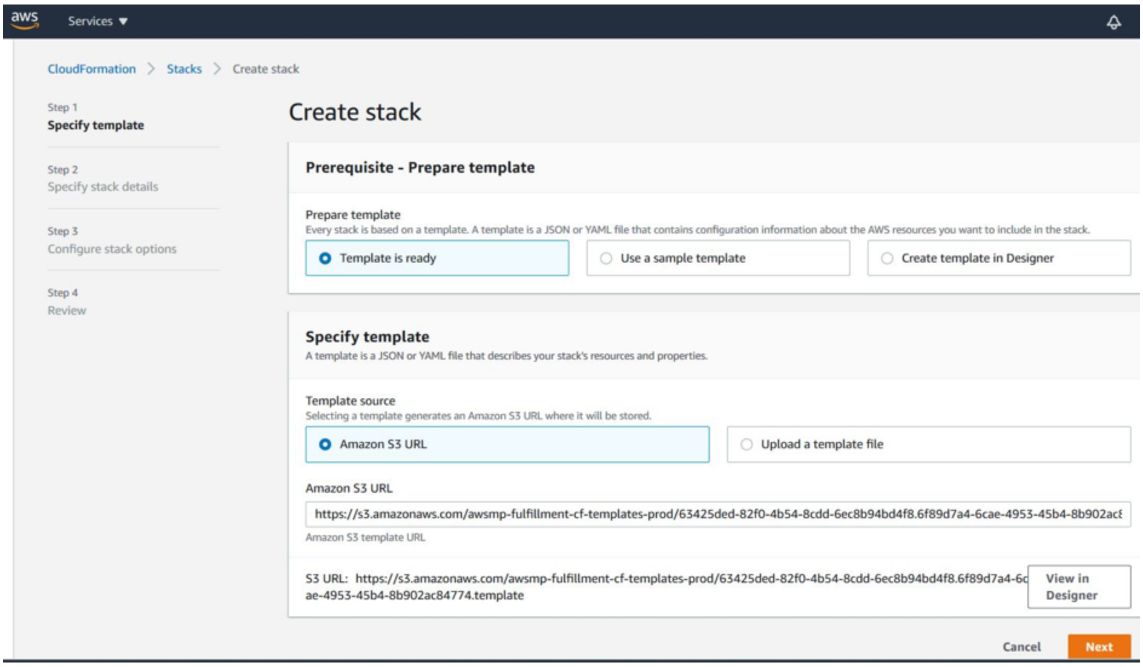
**To launch the Citrix CloudFormation template**:

1. Log on to the AWS marketplace by using your AWS credentials.
2. In the search field, type **NetScaler VPX** to search for the NetScaler AMI, and click **Go**.
3. On the search result page, click the desired NetScaler VPX offering.
4. Click the **Pricing** tab, to go to **Pricing Information**.
5. Select the region and **Fulfillment Option** as **NetScaler VPX –Customer Licensed**.
6. Click **Continue to Subscribe**.
7. Check the details in the **Subscribe** page and click **Continue to Configuration**.
8. Select **Delivery Method** as **CloudFormation Template**.
9. Select the required CloudFormation template.
10. Select **Software Version** and **Region**, and click **Continue to Launch**.




11. Under **Choose Action**, select **Launch CloudFormation**, and click **Launch**.
    The **Create stack** page appears.
12. Click **Next**.

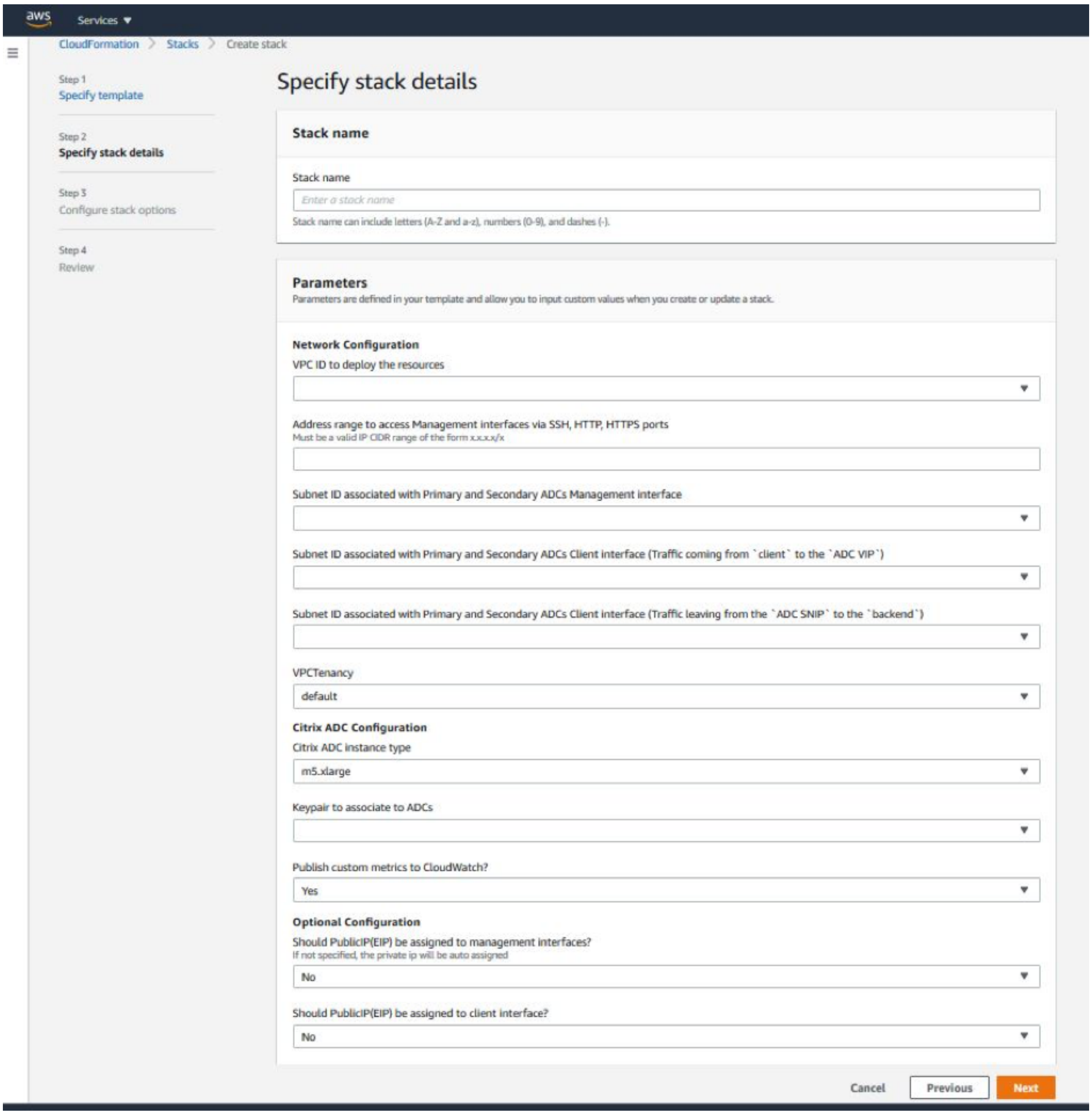



13. The **Specify stack details** page appears. Enter the following details.

    - Type a **Stack name**. The name must be within 25 characters.
    - Under **Network Configuration**, perform the following:

    - Select **Management Subnetwork**, **Client Subnetwork**, and **Server Subnetwork**. Ensure that you select the correct subnetworks you created within the VPC that you selected under VPC ID.
    - Add **Primary Management IP**, **Secondary Management IP**, **Client IP**, and **Server IP**. The IP addresses must belong to the same subnets of the respective subnetworks. Alternatively, you can let the template assign the IP addresses automatically.
    - Select **default** for **VPCTenancy**.
- Under **NetScaler Configuration**, perform the following:
    - Select **m5.xlarge** for **Instance type**.
    - Select the key pair that you've already created from the menu for **Key Pair**.
    - By default, the **Publish custom metrics to CloudWatch?** option is set to **Yes**. If you want to disable this option, select **No**.
      For more information about CloudWatch metrics, see [Monitor your instances using Amazon CloudWatch] (#monitor-your-instances-using-amazon-cloudWatch).
- Under **Optional Configuration**, do the following:
    - By default, the **Should publicIP(EIP) be assigned to management interfaces?** option is set to **No**.
    - By default, the **Should publicIP(EIP) be assigned to client interface?** option is set to **No**.

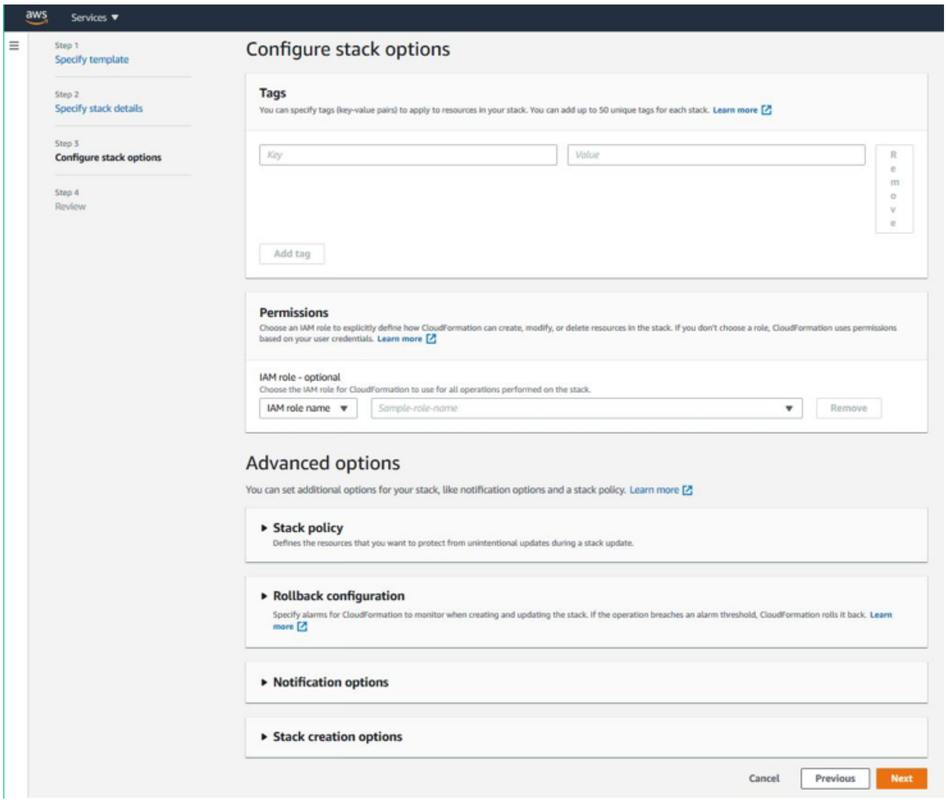
 




14. Click **Next**.

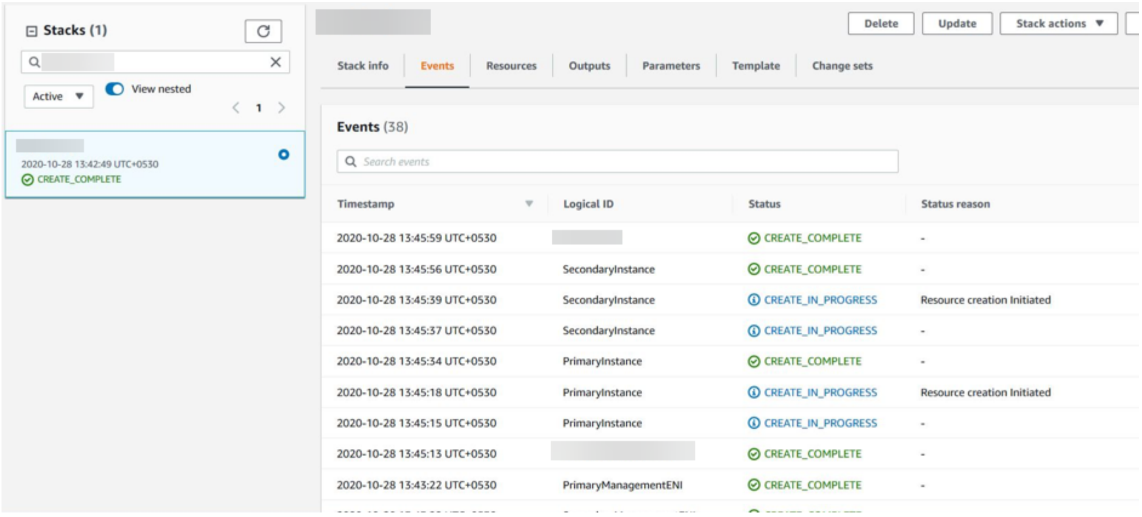
15. The **Configure stack options** page appears. This is an optional page.

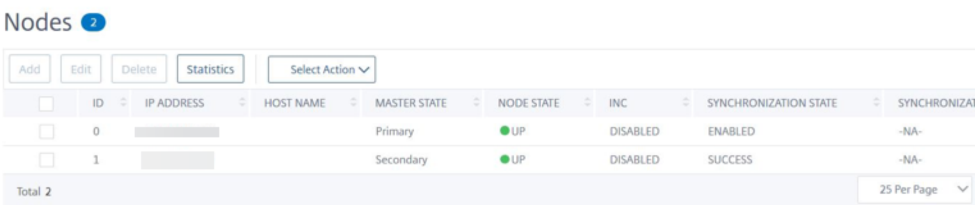



16. Click **Next**.
17. The **Options** page appears. (This is an optional page.). Click **Next**.
18. The **Review** page appears. Take a moment to review the settings and make any changes, if necessary.
19. Select the **I acknowledge that AWS CloudFormation might create IAM resources.** check box, and then click **Create stack**.
20. The **CREATE-IN-PROGRESS** status appears. Wait until the status is **CREATE-COMPLETE**. If the status does not change to **COMPLETE**, check the **Events** tab for the reason of failure, and recreate the instance with proper configurations.




21. After an IAM resource is created, navigate to **EC2 Management Console > Instances**. You find two VPX instances created with IAM role. The primary and secondary nodes are created each with three private IP addresses and three network interfaces.
22. Log on to the primary node with user name `nsroot` and the instance ID as the password. From the GUI, navigate to **System > High Availability > Nodes**. The NetScaler VPX is already configured in HA pair by the CloudFormation template.
23. The NetScaler VPX HA pair appears.




## Monitor your instances using Amazon CloudWatch

You can use the Amazon CloudWatch service to monitor a set of NetScaler VPX metrics such as CPU and memory utilization, and throughput. CloudWatch monitors resources and applications that run on AWS, in real time. You can access the Amazon CloudWatch dashboard by using the AWS Management console. For more information, see Amazon CloudWatch.

### Points to note

- If you deploy a NetScaler VPX instance on AWS by using the AWS web console, the CloudWatch service is enabled by default.

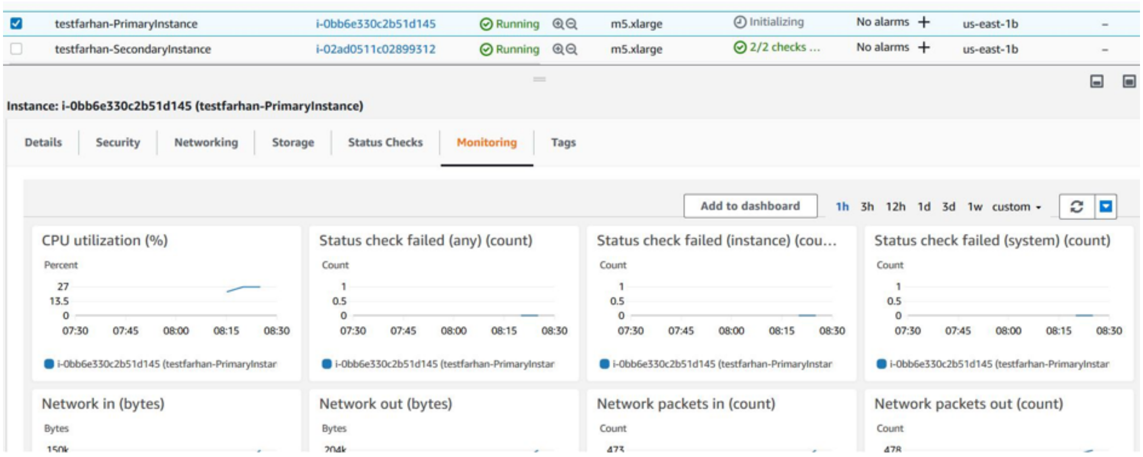
 

- If you deploy a NetScaler VPX instance by using the Citrix CloudFormation template, the default option is "Yes." If you want to disable the CloudWatch service, select "No."
- Metrics are available for CPU (management and packet CPU usage), memory, and throughput (inbound and outbound).

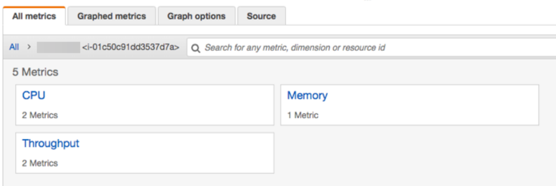
### How to view CloudWatch metrics

To view CloudWatch metrics for your instance, follow these steps:

1. Log on to **AWS Management console > EC2 > Instances**.
2. Select the instance.
3. Click **Monitoring**.
4. Click **View all CloudWatch metrics**.

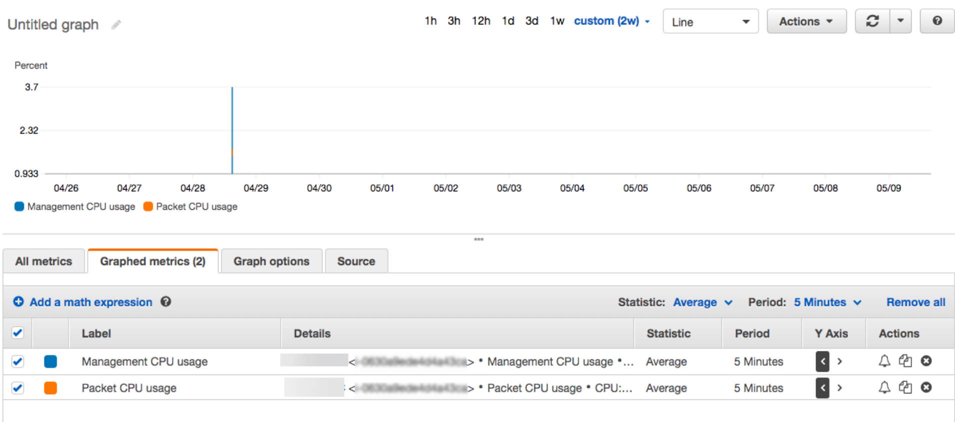



5. Under All metrics, click your instance ID.




6. Click the metrics that you want to view, set the duration (by minutes, hours, days, weeks, months).
7. Click **Graphed metrics** to view the statistics of usage. Use the **Graph options** to customize your graph.

Figure. Graphed metrics for CPU usage




### Configuring SR-IOV on a high availability setup

Support for SR-IOV interfaces in a high availability setup is available from NetScaler release 12.0 57.19 onwards. For more information about how to configure SR-IOV, see Configuring NetScaler VPX instances to Use SR-IOV Network Interface.

### Related resources

How high availability on AWS works

## High availability across different AWS availability zones

You can configure two NetScaler VPX instances on two different subnets or two different AWS availability zones, as a high availability active-passive pair in Independent Network Configuration (INC) mode. If for any reason, the primary node is unable to accept connections, the secondary node takes over.

For more information about high availability, see High availability. For more information about INC, see Configuring high availability nodes in different subnets.

### Points to note

- Read the following documents before you start your deployment:
  - AWS terminology
  - Prerequisites
  - Limitations and usage guidelines
- The VPX high availability pair can either reside in the same availability zone in a different subnet or in two different AWS availability zones.

- Citrix recommends that you use different subnets for management (NSIP), client traffic (VIP), and back-end server (SNIP).
- High availability must be set in Independent Network Configuration (INC) mode for a failover to work.
- The two instances must have port 3003 open for UDP traffic as that is used for heartbeats.
- The management subnets of both the nodes must have access to internet or to AWS API server through internal NAT so that the rest APIs are functional.
- IAM role must have E2 permission for the public IP or elastic IP (EIP) migration and EC2 Route Table permissions for the private IP migration.

You can deploy high availability across AWS availability zones in the following ways:

- Using elastic IP addresses
- Using private IP addresses

### Additional References

For more information on NetScaler Application Delivery Management (ADM) for AWS, see Install the NetScaler ADM agent on AWS.

## Deploy a VPX high-availability pair with elastic IP addresses across different AWS zones

You can configure two NetScaler VPX instances on two different subnets or two different AWS availability zones using elastic IP (EIP) addresses in the INC mode.

For more information about high availability, see High availability. For more information about INC, see Configuring high availability nodes in different subnets.

### How HA with EIP addresses across different AWS zones works

Upon failover, the EIP of the VIP of the primary instance migrates to the secondary, which takes over as the new primary. In the failover process, AWS API:

1. Checks the virtual servers that have `IPSets` attached to them.
2. Finds the IP address that has an associated public IP, from the two IP addresses the virtual server is listening on. One that is directly attached to the virtual server, and the one that is attached through the IP set.
3. Reassociates the public IP (EIP) to the private IP belonging to the new primary VIP.

> **Note:**
>
> To protect your network from attacks such as denial-of-service (DoS), when using an EIP, you can create security groups in AWS to restrict the IP access. For high availability, you can switch from EIP to a private IP movement solution as per your deployments.

### How to deploy a VPX high-availability pair with elastic IP addresses across different AWS zones

The following is the summary of steps for deploying a VPX pair on two different subnets or two different AWS availability zones.

1. Create an Amazon virtual private cloud.
2. Deploy two VPX instances in two different availability zones or in the same zone but in different subnets.
3. Configure high availability

    a) Set up high availability in INC mode in both the instances.
    b) Add an IP set in both the instances.
    c) Bind the IP set in both the instances to the VIP.
    d) Add a virtual server in the primary instance.

For steps 1 and 2, use the AWS console. For steps 3, use the NetScaler VPX GUI or the CLI.

**Step 1.** Create an Amazon virtual private cloud (VPC).

**Step 2.** Deploy two VPX instance in two different availability zones or in the same zone but in different subnets. Attach an EIP to the VIP of the primary VPX.

For more information about how to create a VPC and deploy a VPX instance on AWS, see Deploy a NetScaler VPX standalone instance on AWS and Scenario: standalone instance

**Step 3.** Configure high availability. You can use the NetScaler VPX CLI or the GUI to set up high availability.

#### Configure high availability by using the CLI

1. Set up high availability in INC mode in both the instances.

   **On the primary node:**

   ```
   add ha node 1 <sec_ip> -inc ENABLED
   ```

   **On the secondary node:**

   ```
   add ha node 1 <prim_ip> -inc ENABLED
   ```

<`sec_ip`> refers to the private IP address of the management NIC of the secondary node

<`prim_ip`> refers to the private IP address of the management NIC of the primary node

2. Add the IP set in both the instances.

   Type the following command on both the instances.

   ```
   add ipset <ipsetname>
   ```

3. Bind the IP set to the VIP set on both the instances.

   Type the following command on both the instances:

   ```
   add ns ip <secondary vip> <subnet> -type VIP
   ```

   ```
   bind ipset <ipsetname> <secondary VIP>
   ```

   > **Note:**
   >
   > You can bind the IP set to the primary VIP or to the secondary VIP. However, if you bind the IP set to the primary VIP, use the secondary VIP to add to the virtual server, and conversely.

4. Add a virtual server on the primary instance.

   Type the following command:

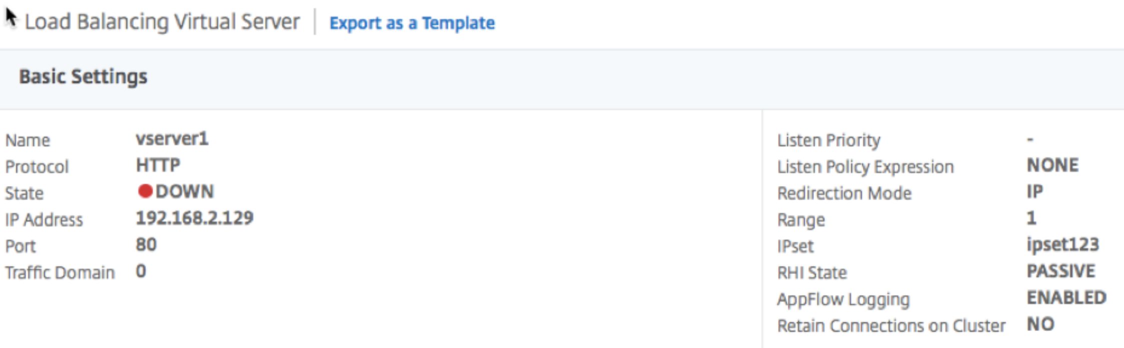
   ```
   add <server_type> vserver <vserver_name> <protocol> <primary_vip>
    <port> -ipset \<ipset_name>
   ```

#### Configure high availability by using the GUI

1. Set up high availability in INC mode on both the instances
2. Log on to the primary node with user name `nsroot` and instance ID as password.
3. From the GUI, go to **Configuration** > **System** > **High Availability**. Click **Add**.
4. At the **Remote Node IP address** field, add the private IP address of the management NIC of the secondary node.
5. Select **Turn on NIC (Independent Network Configuration)** mode on self-node.
6. Under **Remote System Login Credential**, add the user name and password for the secondary node and click **Create**.
7. Repeat the steps in the secondary node.
8. Add IP set and bind IP set to the VIP set on both the instances.
9. From the GUI, navigate to **System** > **Network** > **IPs** > **Add**.
10. Add the required values for IP Address, Netmask, IP Type (virtual IP) and click **Create**.

11. Navigate to **System** > **Network** > **IP Sets** > **Add**. Add an IP set name and click **Insert**.
12. From the IPV4s page, select the virtual IP and click **Insert**. Click **Create** to create the IP set.
13. Add a virtual server in the primary instance

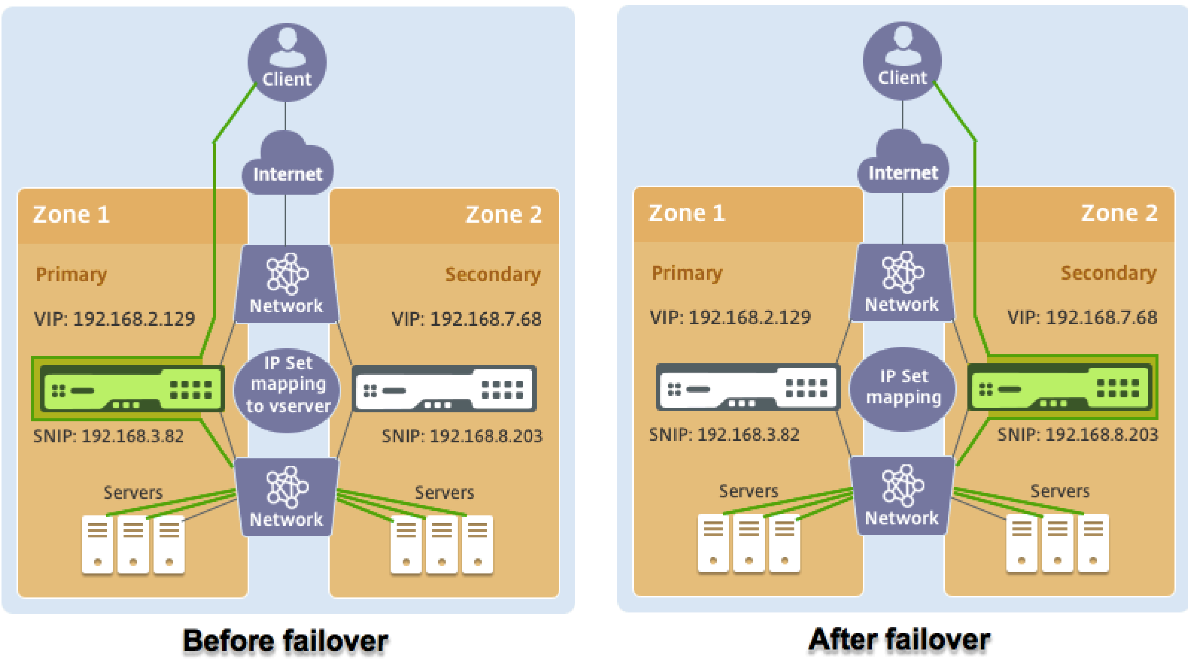
    From the GUI, go to **Configuration** > **Traffic Management** > **Virtual Servers** > **Add**.




## Scenario

In this scenario, a single VPC is created. In that VPC, two VPX instances are created in two availability zones. Each instance has three subnets - one for management, one for client, and one for back-end server. An EIP is attached to the VIP of the primary node.

Diagram: This diagram illustrates the NetScaler VPX high availability setup in INC mode, on AWS




For this scenario, use CLI to configure high availability.

1. Set up high availability in INC mode on both the instances.

Type the following commands on the primary and the secondary nodes.

**On primary:**

`add ha node 1 192.168.6.82 -inc enabled`

Here, 192.168.6.82 refers to the private IP address of the management NIC of the secondary node.

**On secondary:**

`add ha node 1 192.168.1.108 -inc enabled`

Here, 192.168.1.108 refers to the private IP address of the management NIC of the primary node.

2. Add an IP set and bind the IP set to the VIP on both the instances

   **On primary**:

```
add ipset ipset123
add ns ip 192.168.7.68 255.255.255.0 -type VIP
bindipset ipset123 192.168.7.68
```

   **On secondary**:

```
add ipset ipset123
add ns ip 192.168.7.68 255.255.255.0 -type VIP
bind ipset ipset123 192.168.7.68
```

3. Add a virtual server on the primary instance.

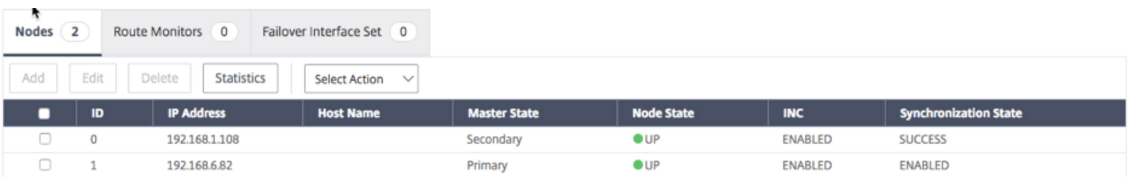
   The following command:

   `add lbvserver vserver1 http 192.168.2.129 80 -ipset ipset123`

4. Save the configuration.


| | ID | IP Address | Host Name | Master State | Node State | INC | Synchronization State |
|---|---|---|---|---|---|---|---|
| | 0 | 192.168.1.108 | | Primary | UP | ENABLED | ENABLED |
| | 1 | 192.168.6.82 | | Secondary | UP | ENABLED | SUCCESS |

5. After a forced failover, the secondary becomes the new primary.




| | ID | IP Address | Host Name | Master State | Node State | INC | Synchronization State |
|---|---|---|---|---|---|---|---|
| | 0 | 192.168.1.108 | | Secondary | UP | ENABLED | SUCCESS |
| | 1 | 192.168.6.82 | | Primary | UP | ENABLED | ENABLED |

## Deploy a VPX high-availability pair with private IP addresses across different AWS zones

You can configure two NetScaler VPX instances on two different subnets or two different AWS availability zones using private IP addresses in the INC mode. This solution can be easily integrated with the existing multizone VPX high-availability pair with elastic IP addresses. Therefore, you can use both the solutions together.

For more information about high availability, see High availability. For more information about INC, see Configuring high availability nodes in different subnets.

> **Note:**
>
> This deployment is supported from NetScaler release 13.0 build 67.39 onwards. This deployment is compatible with AWS Transit Gateway.

### High availability pair with private IP addresses using AWS non-shared VPC

#### Prerequisites

Ensure that the IAM role associated with your AWS account has the following IAM permissions:

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeAddresses",
                "ec2:AssociateAddress",
                "ec2:DisassociateAddress",
                "ec2:DescribeRouteTables",
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "iam:SimulatePrincipalPolicy",
                "iam:GetRole"
            ],
            "Resource": "*",
            "Effect": "Allow"
         }

    ]
 }
```

#### Deploy a VPX HA pair with private IP addresses using AWS non-shared VPC

The following is the summary of steps for deploying a VPX pair on two different subnets or two different AWS availability zones using private IP addresses.

1. Create an Amazon virtual private cloud.
2. Deploy two VPX instances in two different availability zones.
3. Configure high availability

    a) Set up high availability in INC mode in both the instances.
    b) Add the respective route tables in the VPC that points to the client interface.
    c) Add a virtual server in the primary instance.

For steps 1, 2, and 3b, use the AWS console. For step 3a and 3c, use the NetScaler VPX GUI or the CLI.

**Step 1.** Create an Amazon virtual private cloud (VPC).

**Step 2.** Deploy two VPX instance in two different availability zones with the same number of ENI (Network Interface).

For more information about how to create a VPC and deploy a VPX instance on AWS, see Deploy a NetScaler VPX standalone instance on AWS and Scenario: standalone instance

**Step 3.** Configure the ADC VIP addresses by choosing a subnet that does not overlap with the Amazon VPC subnets. If your VPC is 192.168.0.0/16, then to configure ADC VIP addresses, you can choose any subnet from these IP address ranges:

- 0.0.0.0 - 192.167.0.0
- 192.169.0.0 - 254.255.255.0

In this example, the chosen 10.10.10.0/24 subnet and created VIPs in this subnet. You can choose any subnet other than the VPC subnet (192.168.0.0/16).

**Step 4.** Add a route that points to the client interface (VIP) of the primary node from the VPC route table.

From the AWS CLI, type the following command:

```
aws ec2 create-route --route-table-id rtb-2272532 --destination-cidr-
    block 10.10.10.0/24 --gateway-id <eni-client-primary>
```

From the AWS GUI, perform the following steps to add a route:

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Route Tables**, and select the route table.
3. Choose **Actions**, and click **Edit routes**.

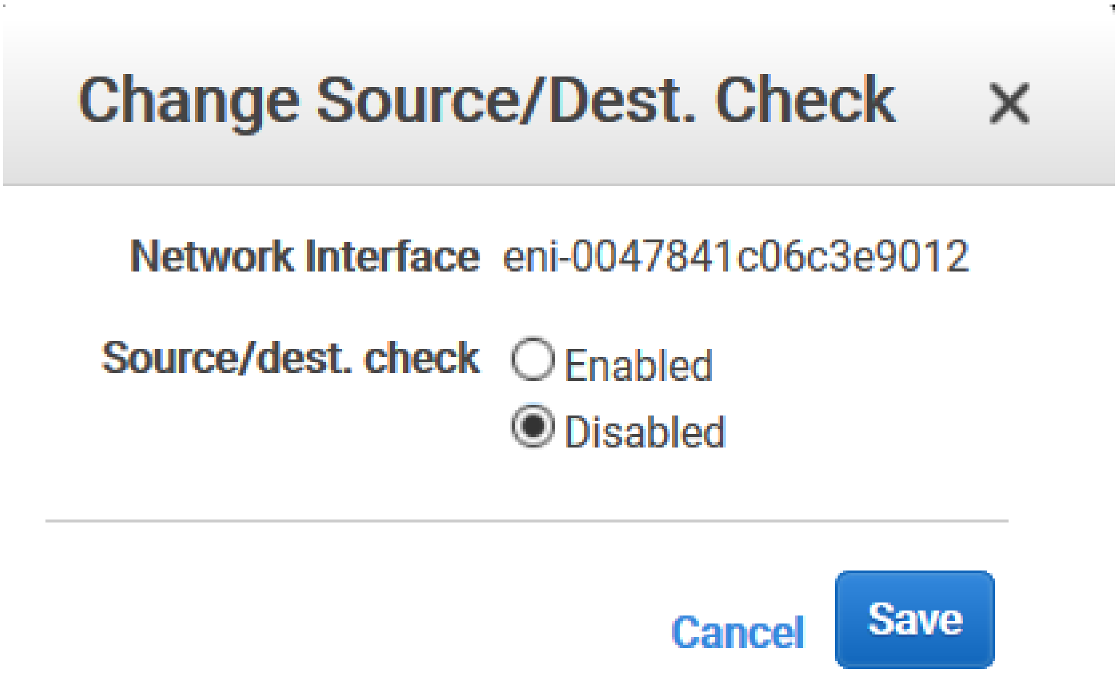
4. To add a route, choose **Add route**. For **Destination**, enter the destination CIDR block, a single IP address, or the ID of a prefix list. For gateway ID, select the ENI of a client interface of the primary node.


> **Note:**
>
> You must disable **Source/Dest Check** on the client ENI of the primary instance.

To disable the source/destination checking for a network interface using the console, perform the following steps:

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface of a primary client interface, and choose **Actions**, and click **Change Source/Dest. Check**.
4. In the dialog box, choose **Disabled**, click **Save**.




**Step 5.** Configure high availability. You can use the NetScaler VPX CLI or the GUI to set up high availability.

#### Configure high availability by using the CLI

1. Set up high availability in INC mode in both the instances.

   On the primary node:

   ```
   add ha node 1 \<sec\_ip\> -inc ENABLED
   ```

   On the secondary node:

   ```
   add ha node 1 \<prim\_ip\> -inc ENABLED
   ```

   <sec_ip> refers to the private IP address of the management NIC of the secondary node.

   <prim_ip> refers to the private IP address of the management NIC of the primary node.

2. Add a virtual server on the primary instance. You must add it from the chosen subnet, for example, 10.10.10.0/24.

   Type the following command:

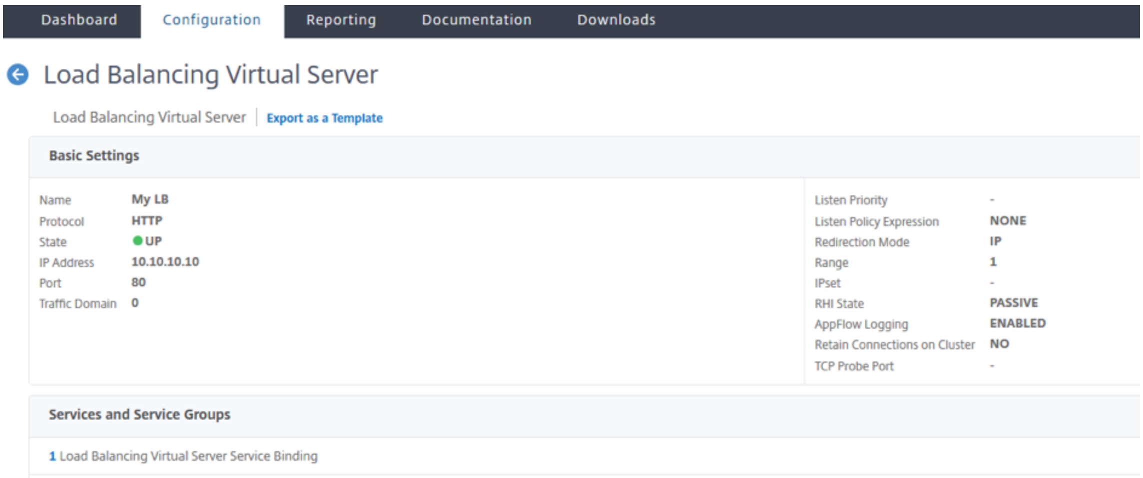
   ```
   add \<server\_type\> vserver \<vserver\_name\> \<protocol\> \<
       primary\_vip\> \<port\>
   ```

#### Configure high availability by using the GUI

1. Set up high availability in INC mode on both the instances
2. Log on to the primary node with user name `nsroot` and instance ID as password.
3. Navigate to **Configuration** > **System** > **High Availability**, and click **Add**.
4. At the **Remote Node IP address** field, add the private IP address of the management NIC of the secondary node.
5. Select **Turn on NIC (Independent Network Configuration)** mode on self-node.
6. Under **Remote System Login Credential**, add the user name and password for the secondary node and click **Create**.
7. Repeat the steps in the secondary node.
8. Add a virtual server in the primary instance

   Navigate to **Configuration** > **Traffic Management** > **Virtual Servers** > **Add**.




### Deploy a VPX HA pair with private IP addresses using AWS shared VPC

In an AWS shared VPC model, the account that owns the VPC (owner) shares one or more subnets with other accounts (participants). Therefore, you have a VPC owner account and a participant account. After a subnet is shared, the participants can view, create, modify, and delete their application resources in the subnets shared with them. Participants cannot view, modify, or delete resources that belong to other participants or the VPC owner.

For information on AWS shared VPC, see AWS documentation.

> **Note:**
>
> The configuration steps for deploying a VPX HA pair with private IP addresses using AWS shared VPC is same as the Deploy a VPX HA pair with private IP addresses using AWS non-shared VPC with the following exception:
>
> - The route tables in the VPC that points to the client interface must be added from the *VPC owner account*.

#### Prerequisites

- Ensure that the IAM role associated with NetScaler VPX instance in the AWS participant account has the following IAM permissions:

```
"Version": "2012-10-17",
    "Statement": [
        {

            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateAddress",
                "iam:GetRole",
                "iam:SimulatePrincipalPolicy",
                "ec2:DescribeInstances",
                "ec2:DescribeAddresses",
                "ec2:ModifyNetworkInterfaceAttribute",
                “ec2:AssociateAddress”,
                "sts:AssumeRole"
        ],
            "Resource": "*"
         }

    ]
 }
```

> **Note:**
>
> The **AssumeRole** allows NetScaler VPX instance to assume the cross-account IAM role, which is created by the VPC owner account.

- Ensure that the VPC owner account provides the following IAM permissions to the participant account using cross-account IAM role:

```
{

    "Version": "2012-10-17",
    "Statement": [
        {
```

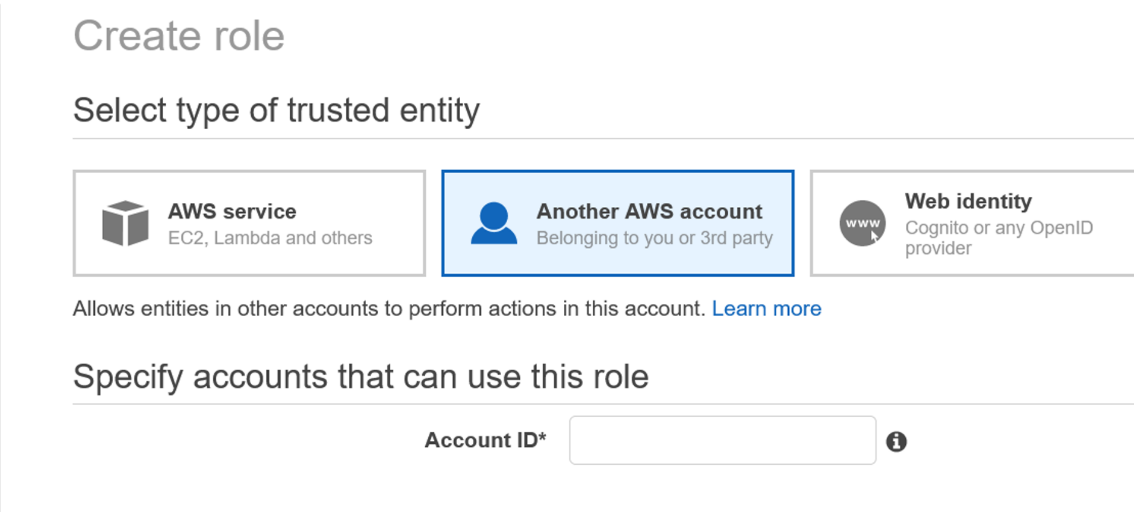
 

```
                "Sid": "VisualEditor0",
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateRoute",
                    "ec2:DeleteRoute",
                    "ec2:DescribeRouteTables"
                ],
                "Resource": "*"
            }

        ]
    }
```

#### Create cross-account IAM role

1. Log in to the AWS web console.
2. In the **IAM** tab, navigate to **Roles** and then choose **Create Role**.
3. Choose **Another AWS account**.




4. Enter the 12-digit account ID number of the participant account that you want to grant administrator access to.

#### Set cross-account IAM role by using the NetScaler CLI

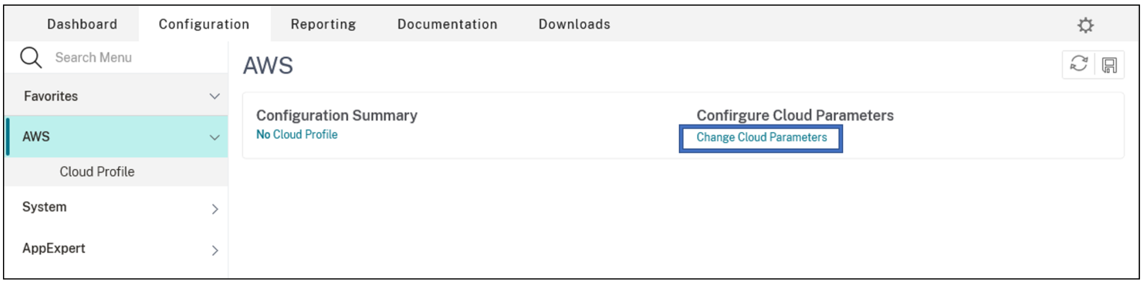
The following command enables NetScaler VPX instance to assume the cross-account IAM role that exists in the VPC owner account.

```
set cloud awsParam -roleARN <string>
```

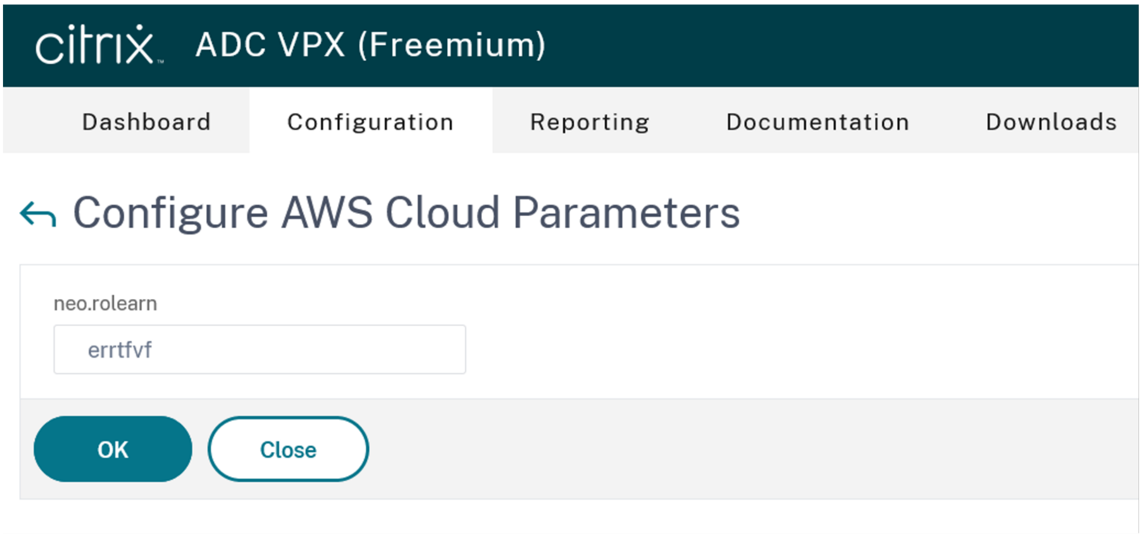
 

#### Set cross-account IAM role by using the NetScaler GUI

1. Sign into NetScaler appliance and navigate to **Configuration > AWS > Change cloud parameters**.




2. In the **Configure AWS Cloud Parameters** page, enter value for the **RoleARN** field.

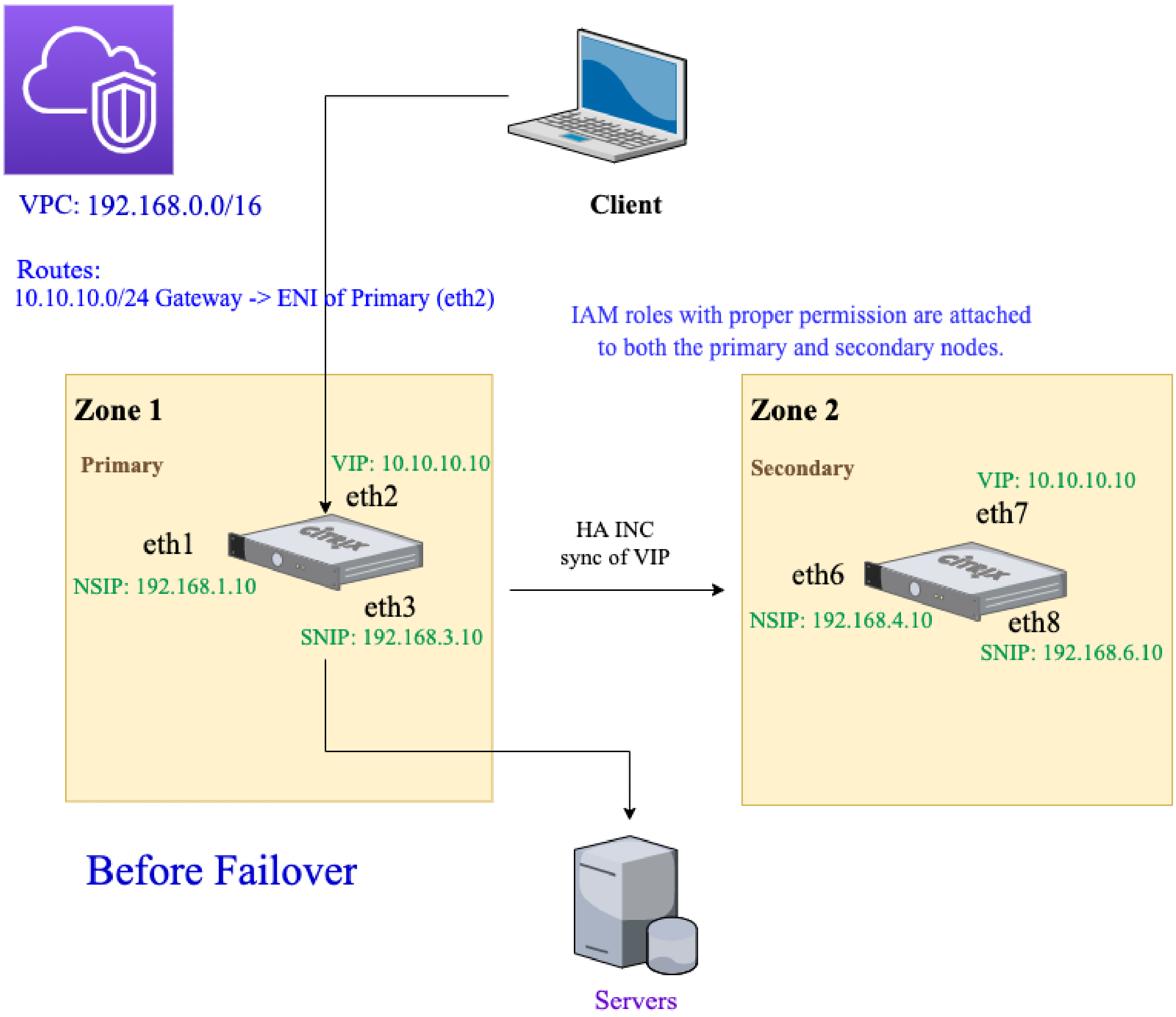



### Scenario

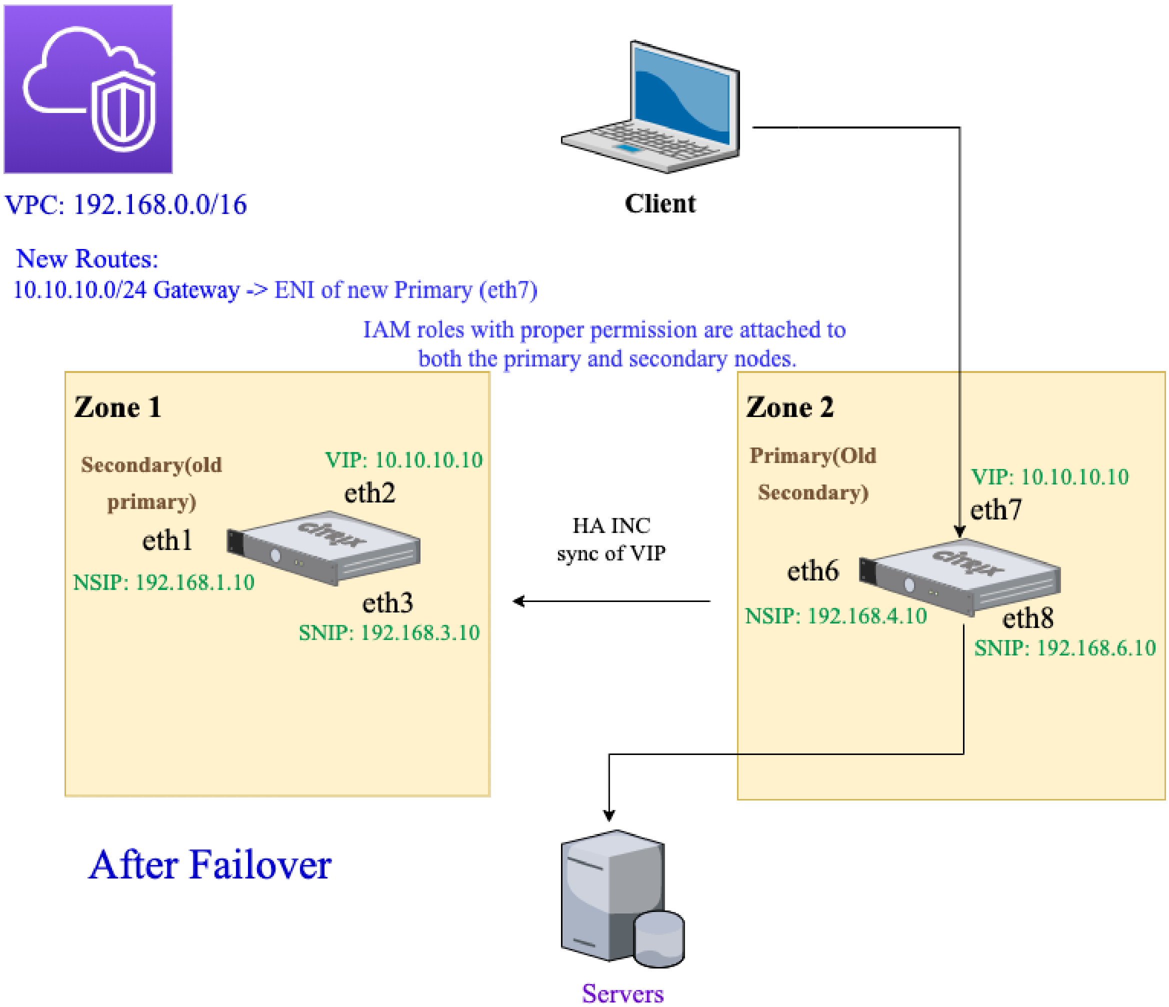
In this scenario, a single VPC is created. In that VPC, two VPX instances are created in two availability zones. Each instance has three subnets - one for management, one for client, and one for back-end server.

The following diagrams illustrate the NetScaler VPX high availability setup in INC mode, on AWS. The custom subnet 10.10.10.10, which is not part of the VPC is used as VIP. Therefore, the 10.10.10.10 subnet can be used across availability zones.


VPC: 192.168.0.0/16
Routes:
10.10.10.0/24 Gateway -> ENI of Primary (eth2)
Client
IAM roles with proper permission are attached
to both the primary and secondary nodes.
Zone 1
Primary
VIP: 10.10.10.10
eth2
eth1
NSIP: 192.168.1.10
eth3
SNIP: 192.168.3.10
HA INC
sync of VIP
Zone 2
Secondary
VIP: 10.10.10.10
eth7
eth6
NSIP: 192.168.4.10
eth8
SNIP: 192.168.6.10
Before Failover
Servers




For this scenario, use CLI to configure high availability.

1. Set up high availability in INC mode on both the instances.

   Type the following commands on the primary and the secondary nodes.

   On the primary node:

   ```
   add ha node 1 192.168.4.10 -inc enabled
   ```

   Here, 192.168.4.10 refers to the private IP address of the management NIC of the secondary node.

   On the secondary node:

   ```
   add ha node 1 192.168.1.10 -inc enabled
   ```

   Here, 192.168.1.10 refers to the private IP address of the management NIC of the primary node.

2. Add a virtual server on the primary instance.

   Type the following command:

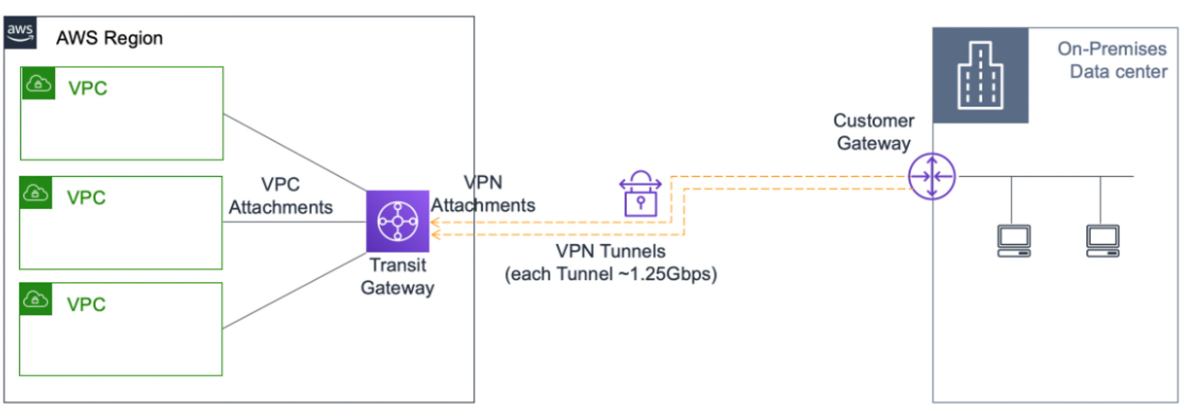
```
add lbvserver vserver1 http 10.10.10.10 80
```

3. Save the configuration.
4. After a forced failover:

   - The secondary instance becomes the new primary instance.
   - The VPC route pointing to the primary ENI migrates to the secondary client ENI.
   - Client traffic resumes to the new primary instance.

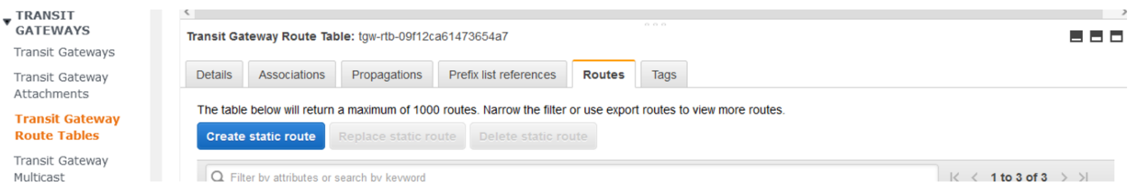
### AWS Transit Gateway configuration for HA private IP solution

You need AWS Transit Gateway to make the private VIP subnet routable within the internal network, across AWS VPCs, regions, and On-premises networks. The VPC must connect to AWS Transit Gateway. A static route for the VIP subnet or IP pool inside the AWS Transit Gateway route table is created and pointed towards the VPC.



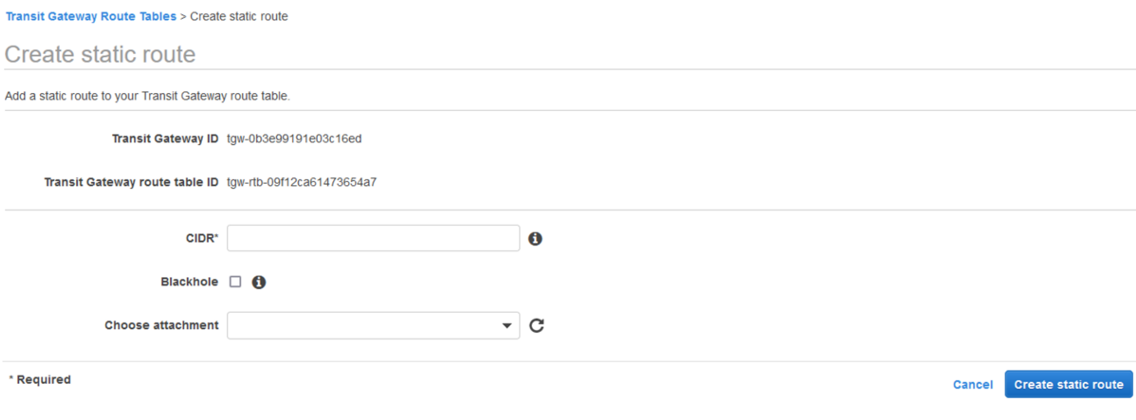

To configure AWS Transit Gateway, follow these steps:

1. Open the [Amazon VPC console](https://console.aws.amazon.com/vpc/).
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Choose the **Routes** tab, and click **Create static route**.




4. Create a static route where CIDR points to your private VIPS subnet and attachment points to the VPC having NetScaler VPX.




5. Click **Create static route**, then choose **Close**.

### Troubleshooting

If you face any issues while configuring HA private IP solution across multizone HA, check the following key points for troubleshooting:

- Both primary and secondary nodes have the same set of IAM permissions.
- INC mode is enabled on both the primary and secondary nodes.
- Both primary and secondary nodes have the same number of interfaces.
- While creating an instance, follow the same order of attaching interfaces on both the nodes. On a primary node, if the client interface is attached first and the server interface is attached second. Then, follow the same order on the secondary node as well. If there is any mismatch, detach and reattach the interfaces in the correct order.
- If traffic does not flow, make sure the "Source/dest. Check"is disabled on the client interface of the primary node for the first time.
- Make sure the cloudhadaemon command (`ps -aux | grep cloudha`) is running in Shell.
- Make sure that the NetScaler firmware version is 13.0 build 70.x or later.
- For issues with the failover process, check the log file available at: /var/log/cloud-ha-daemon.log

## Deploy a NetScaler VPX instance on AWS Outposts

AWS Outposts is a pool of AWS compute and storage capacity deployed at your site. Outposts provides AWS infrastructure and services in your on-premises location. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can use the same NetScaler VPX instances, AWS APIs, tools, and infrastructure across on-premises and the AWS cloud for a consistent hybrid experience.

You can create subnets on your Outposts and specify them when you create AWS resources such as EC2 instances, EBS volumes, ECS clusters, and RDS instances. Instances in the Outposts subnets communicate with other instances in the AWS Region using private IP addresses, all within the same Amazon Virtual Private Cloud (VPC).

For more information, see the AWS Outposts user guide.

### How AWS Outposts works

AWS Outposts is designed to operate with a constant and consistent connection between your Outposts and an AWS Region. To achieve this connection to the Region, and to the local workloads in your on-premises environment, you must connect your Outpost to your on-premises network. Your on-premises network must provide WAN access back to the Region and to the internet. The internet must also provide LAN or WAN access to the local network where your on-premises workloads or applications reside.

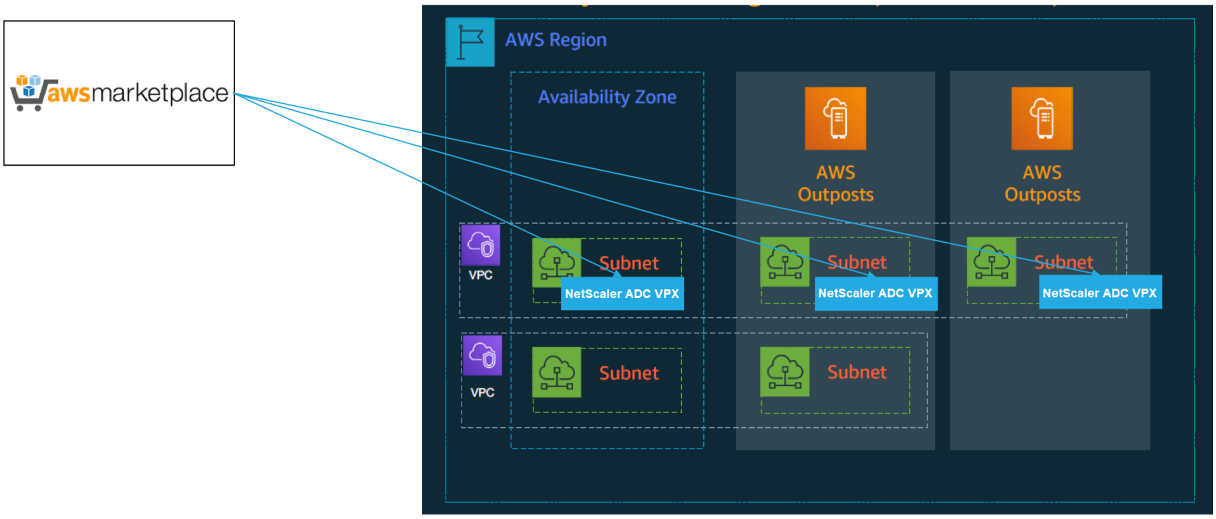
### Prerequisite

- You must install an AWS Outposts at your site.
- The AWS Outposts' compute and storage capacity must be available for use.

For more information on how to place an order for AWS Outposts, see the following AWS documentation:
https://aws.amazon.com/blogs/aws/aws-outposts-now-available-order-your-racks-today/

### Deploy a NetScaler VPX instance on AWS Outposts by using the AWS web console

The following figure depicts a simple deployment of NetScaler VPX instances on the Outposts. The NetScaler AMI present in the AWS Marketplace is also deployed in the Outposts.




Log in to the AWS web console and complete the following steps to deploy NetScaler VPX EC2 instances on your AWS Outposts.

1. Create a key pair.
2. Create a Virtual Private Cloud (VPC).
3. Add more subnets.
4. Create security groups and security rules.
5. Add route tables.
6. Create an internet gateway.
7. Create an NetScaler VPX instance by using the AWS EC2 service.
   From the AWS dashboard, navigate to **Compute > EC2 > Launch Instance > AWS Marketplace**.
8. Create and attach more network interfaces.
9. Attach elastic IPs to the management NIC.
10. Connect to the VPX instance.

For detailed instructions on each of the steps, see Deploy a NetScaler VPX instance on AWS by using the AWS web console.

For high availability within same availability zone deployment, see Deploy a high availability pair on AWS.

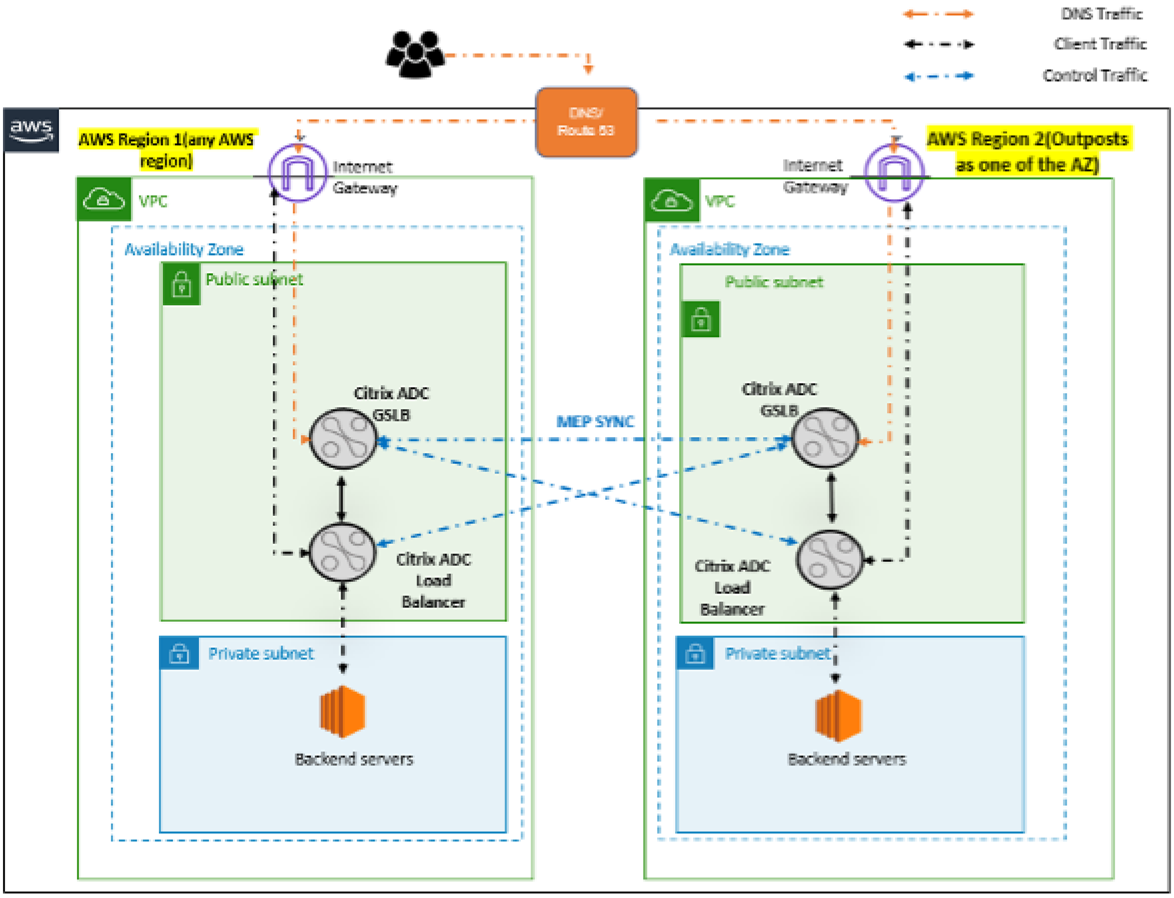
### Deploy a NetScaler VPX instance on hybrid cloud with AWS Outposts

You can deploy a NetScaler VPX instance on hybrid cloud in an AWS environment that contains AWS outposts. You can simplify the app delivery mechanism using the NetScaler global server load balancing (GSLB) solution. The GSLB solution distributes application traffic across multiple data centers in hybrid clouds that are built using AWS regions and AWS Outposts infrastructure.

NetScaler GSLB supports both the active-active and active-passive deployment types to address different use cases. Along with these flexible deployment options and application delivery mechanisms, NetScaler secures the entire network and application portfolio, irrespective of whether applications are deployed natively on AWS Cloud or AWS Outposts.

The following diagram illustrates an application delivery with NetScaler appliance in hybrid cloud with AWS.




In an active-active deployment, the NetScaler steers the traffic globally across a distributed environment. All the sites in the environment exchange metrics about their availability and health of resources through the Metrics Exchange Protocol (MEP). The NetScaler appliance uses this information to load balance traffic across sites, and sends client requests to the most appropriate GSLB site as determined by the defined method (round robin, least connection, and static proximity) specified in the GSLB configuration.

You can use the active-active GSLB deployment to:

- Optimize the resource utilization with all nodes being active.
- Enhance the user experience by steering requests to the site closest to each individual user.
- Migrate applications to the cloud at a user-defined pace.

You can use the active-passive GSLB deployment for:

- Disaster recovery
- Cloud burst

#### References

- Deploy a NetScaler VPX instance on AWS
- Deploy a NetScaler VPX instance on AWS Outposts by using the AWS web console
- Configure GSLB on NetScaler VPX instances

## Protect AWS API Gateway using the NetScaler Web App Firewall

You can deploy a NetScaler appliance in front of your AWS API Gateway and secure the API gateway from external threats. NetScaler Web App Firewall (WAF) can defend your API against OWASP top 10 threats and zero-day attacks. NetScaler Web App Firewall uses a single code base across all ADC form factors. Hence, you can consistently apply and enforce security policies across any environment. NetScaler Web App Firewall is easy to deploy and is available as a single license. The NetScaler Web App Firewall provides you the following features:

- Simplified configuration
- Bot management
- Holistic visibility
- Collate data from multiple sources and display the data in a unified screen

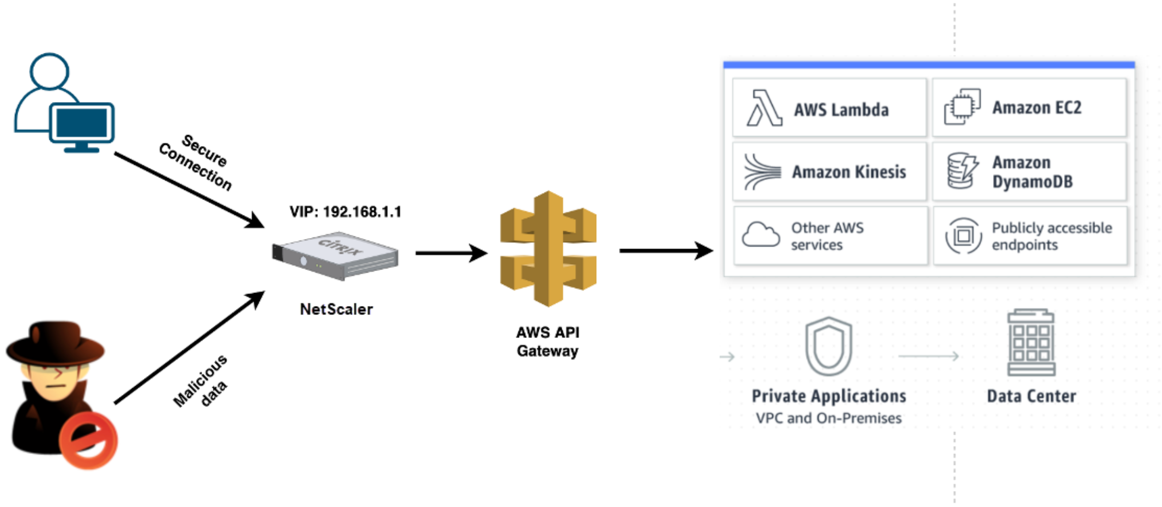
In addition to API gateway protection, you can also use the other NetScaler features. For more information, see NetScaler documentation. Besides to avoid data center failovers and minimizing shutdown time, you can place ADC in high availability within or across availability zones. You can also use or configure clustering with Autoscale feature.

Earlier, AWS API Gateway did not support the protections needed to secure the applications behind it. Without the Web Application Firewall (WAF) protections, APIs were prone to security threats.

### Deploy NetScaler appliance in front of AWS API gateway

In the following example, a NetScaler appliance is deployed in front of the AWS API gateway.




Let's assume there's a genuine API request for AWS Lambda service. This request can be for any of the API services as mentioned in Amazon API Gateway documentation. As shown in the preceding diagram, the traffic flow is as follows:

1. Client sends a request to the AWS Lambda Function (XYZ). This client request is sent to the NetScaler virtual server (192.168.1.1).
2. The virtual server inspects the packet and checks for any malicious content.
3. The NetScaler appliance triggers a Rewrite policy to change the host name and URL in a client request. For example, you want to change `https://restapi.citrix.com/default/LamdaFunctionXYZ` to `https://citrix.execute-api.<region>.amazonaws.com/default/LambdaFunctionXYZ`.
4. The NetScaler appliance forwards this request to the AWS API gateway.
5. The AWS API Gateway further sends the request to the Lambda service and calls the Lambda function "XYZ".
6. At the same time, if an attacker sends an API request with malicious content, the malicious request lands on the NetScaler appliance.
7. The NetScaler appliance inspects the packets and drops the packets based on the configured action.

### Configure NetScaler appliance with WAF enabled

To enable WAF on a NetScaler appliance, do the following steps:

1. Add a content switching or a load balancing virtual server. Let's say the IP address of the virtual server is 192.168.1.1, which resolves to a domain name (restapi.citrix.com).

2. Enable WAF policy on NetScaler virtual server. For more information, see Configuring the Web App Firewall.

3. Enable Rewrite policy to change the domain name. Let's say, you want to change the incoming request to load balancer at "restapi.citrix.com"domain name to be rewritten to the back-end AWS API Gateway at "citrix.execute-api.<region>.amazonaws"domain name.
4. Enable L3 mode on the NetScaler appliance to make it act as a proxy. Use the following command:

```
enable ns mode L3
```

In Step 3 of the preceding example, let's say the website administrator wants the NetScaler appliance to replace the "restapi.citrix.com"domain name with "citrix.execute-api.<region>.amazonaws.com"and the URL with "default/lambda/XYZ".

The following procedure describes how to change the host name and URL in a client request using rewrite feature:

1. Log on to the NetScaler appliance using SSH.
2. Add rewrite actions.

```
add rewrite action rewrite_host_hdr_act replace "HTTP.REQ.HEADER(\"Host\")" "\"citrix.execute-api.<region>.amazonaws.com\""

add rewrite action rewrite_url_act replace HTTP.REQ.URL.PATH_AND_QUERY "\"/default/lambda/XYZ\""
```

3. Add rewrite policies for the rewrite actions.

```
add rewrite policy rewrite_host_hdr_pol "HTTP.REQ.HEADER(\"Host\").CONTAINS(\"restapi.citrix.com\") "rewrite_host_hdr_act

add rewrite policy rewrite_url_pol "HTTP.REQ.HEADER(\"Host\").CONTAINS(\"restapi.citrix.com\") "rewrite_url_act
```

4. Bind the rewrite policies to a virtual server.

```
bind lb vserver LB_API_Gateway -policyName rewrite_host_hdr_pol -priority 10 -gotoPriorityExpression 20 -type REQUEST

bind lb vserver LB_API_Gateway -policyName rewrite_url_pol -priority 20 -gotoPriorityExpression END -type REQUEST
```

For more information, see Configure rewrite to change the host name and URL in client request on NetScaler appliance.

### NetScaler features and capabilities

The NetScaler appliance besides securing the deployment can also enhance the request based on the user requirement. The NetScaler appliance provides the following key features.

- **Load balance the API gateway:** If you have more than one API gateway, you can load balance multiple API gateways using the NetScaler appliance and define the behavior of the API request.
  - Different load balancing methods are available. For example, the Least connection method avoids overloading of API Gateway limit, the Custom load method maintains a specific load on a particular API gateway, and so on. For more information, see Load balancing algorithms.
  - SSL offloading is configured without interrupting the traffic.
  - Use Source IP (USIP) mode is enabled to preserve the client IP address.
  - User-defined SSL settings: You can have your own SSL virtual server with your own-signed certificates and algorithms.
  - Backup virtual server: If the API gateway is not reachable, you can send the request to a backup virtual server for further actions.
  - Many other load balancing features are available. For more information, see Load balance traffic on a NetScaler appliance.
- **Authentication, Authorization and Auditing:** You can define your own authentication methods like LDAP, SAML, RADIUS, and authorize and audit the API requests.
- **Responder:** You can redirect API requests to some other API Gateway during the shutdown time.
- **Rate limiting:** You can configure the rate limiting feature to avoid overloading of an API gateway.
- **Better Availablity:** You can configure a NetScaler appliance in a high availability setup or a cluster setup to give better availability to your AWS API traffics.
- **REST API:** Supports the REST API, which can be used for automating the work in cloud production environments.
- **Monitor data:** Monitors and logs the data for reference.

The NetScaler appliance provides a lot more features, which can be integrated with the AWS API gateway. For more information, see NetScaler documentation.

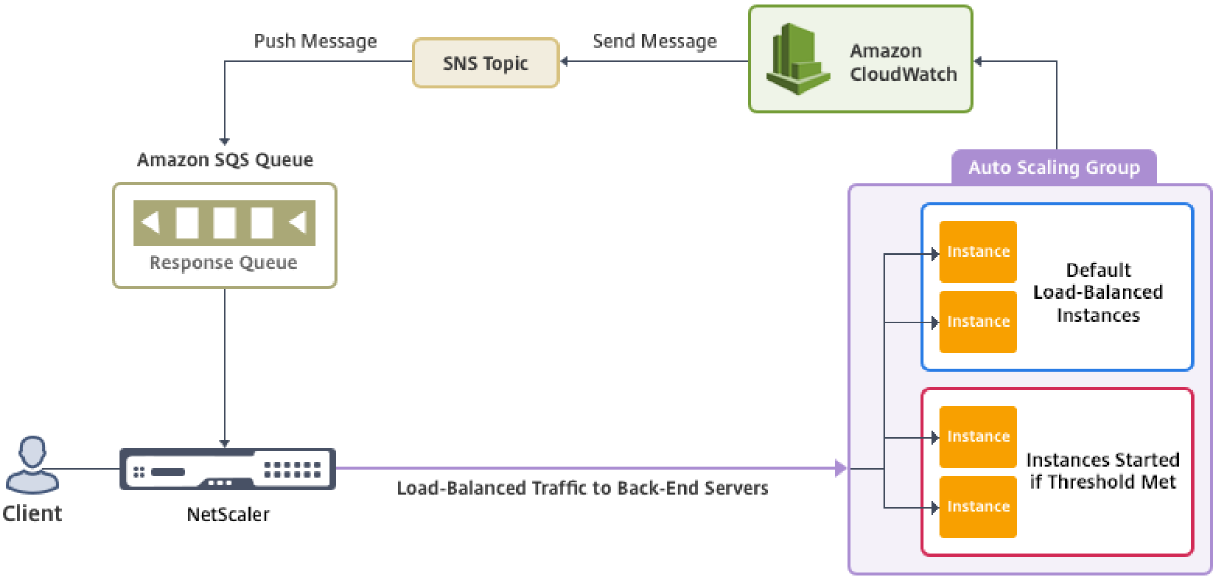
## Add back-end AWS Autoscaling service

Efficient hosting of applications in a cloud involves easy and cost-effective management of resources depending on the application demand. To meet increasing demand, you have to scale network resources upward. Whether demand subsides, you need to scale down to avoid the unnecessary cost of idle resources. To minimize the cost of running the application by deploying only as many instances as are necessary during any given time, you constantly have to monitor traffic, memory and CPU use, and so on. However, monitoring traffic manually is cumbersome. For the application environment to scale up or down dynamically, you must automate the processes of monitoring traffic and of scaling resources up and down whenever necessary.

Integrated with the AWS Auto Scaling service, the NetScaler VPX instance provides the following advantages:

- **Load balance and management**: Auto configures servers to scale up and scale down, depending on demand. The VPX instance auto detects Autoscale groups in the back-end subnet and allows a user to select the Autoscale groups to balance the load. All of this is done by auto configuring the virtual and subnet IP addresses on the VPX instance.
- **High availability**: Detects Autoscale groups that span multiple availability zones and load-balance servers.
- **Better network availability**: The VPX instance supports:
  - Back-end servers on different VPCs, by using VPC peering
  - Back-end servers on same placement groups
  - Back-end servers on different availability zones
- **Graceful connection termination**: Removes Autoscale servers gracefully, avoiding loss of client connections when scale-down activity occurs, by using the Graceful Timeout feature.

**Diagram**: AWS Autoscaling service with a NetScaler VPX Instance




This diagram illustrates how the AWS Autoscaling service is compatible with a NetScaler VPX instance (Load balancing virtual server). For more information, see the following AWS topics.

- Autoscaling groups

- CloudWatch
- Simple Notification Service (SNS)
- Simple Queue Service (Amazon SQS)

### Before you begin

Before you start using Autoscaling with your NetScaler VPX instance, you must complete the following tasks.

1. Read the following topics:
   - Prerequisites
   - Limitation and usage guidelines
2. Create a NetScaler VPX instance on AWS according to your requirement.
   - For more information about how to create a NetScaler VPX standalone instance, see Deploy a NetScaler VPX standalone instance on AWS and Scenario: standalone instance
   - For more information about how to deploy VPX instances in HA mode, see Deploy a high availability pair on AWS.

   > **Note:**
   >
   > Citrix recommends the CloudFormation template for creating NetScaler VPX instances on AWS.
   >
   > Citrix recommends you create three interfaces: one for management (NSIP), one for client-facing LB virtual server (VIP), and one for subnet IP (NSIP).

3. Create an AWS Autoscale group. If you don't have an existing Autoscaling configuration, you must:

   a) Create a Launch Configuration

   b) Create an Autoscaling Group

   c) Verify the Autoscaling Group

   For more information, see http://docs.aws.amazon.com/autoscaling/latest/userguide/GettingStartedTutorial.html.
4. In the AWS Autoscale group, you must specify at least one scale-down policy. The NetScaler VPX instance supports only the Step scaling policy. The Simple scaling policy and Target tracking scaling policy are not supported for Autoscale group.

### Add the AWS Autoscaling service to a NetScaler VPX instance

You can add the Autoscaling service to a VPX instance with a single click by using the GUI. Complete these steps to add the Autoscaling service to the VPX instance:

1. Log on to the VPX instance by using your credentials for `nsroot`.
2. When you log on to the NetScaler VPX instance for the first time, you see the default Cloud Profile page. Select the AWS Autoscaling group from the drop-down menu and click **Create** to create a cloud profile. Click **Skip** if you want to create the cloud profile later.

   **Points to keep in mind while creating a Cloud Profile**: By default the CloudFormation Template creates and attaches the below IAM Role.

```
{


    "Version": "2012-10-17",
    "Statement": \[

     {


            "Action": \[

                "ec2:DescribeInstances",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DetachNetworkInterface",
                "ec2:AttachNetworkInterface",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances",
                "autoscaling:\*",
                "sns:\*",
                "sqs:\*"

             " iam: SimulatePrincipalPolicy "
             " iam: GetRole "

            \],

            "Resource": "\*",
            "Effect": "Allow"

         }


    \]

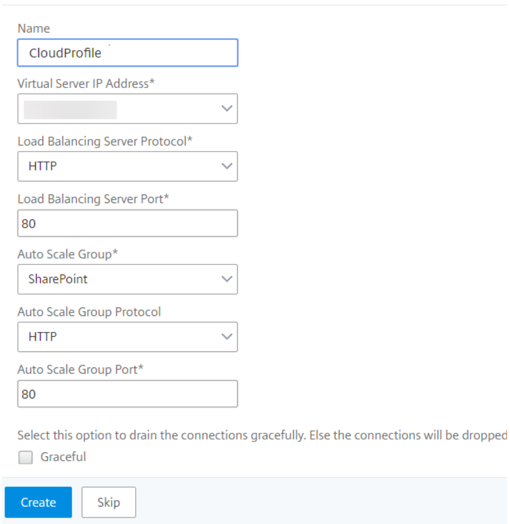
 }
```

   **Ensure** the IAM role of an instance has proper permissions.

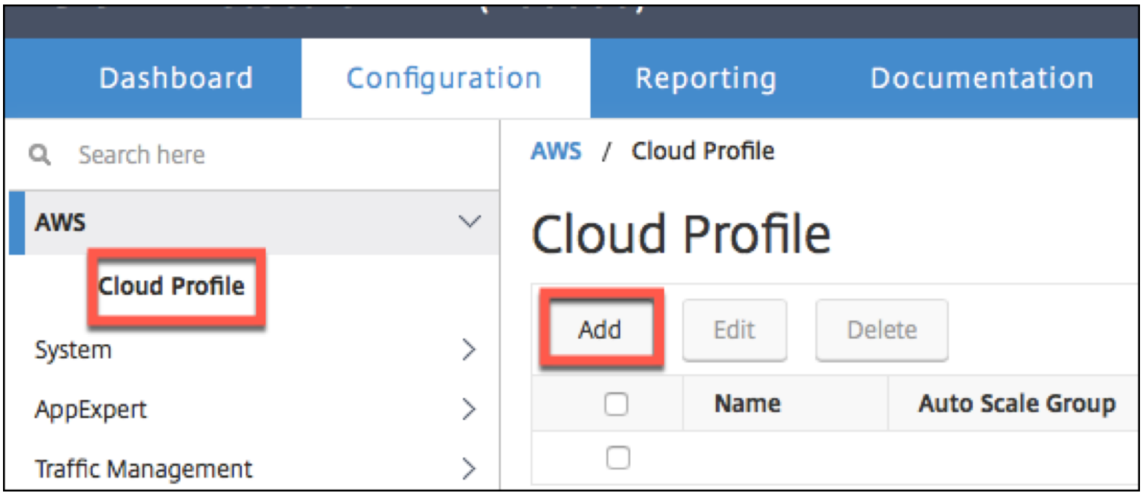
 

- The virtual server IP address is autopopulated from the free IP address available to the VPX instance. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/MultipleIP.html#ManageMultipleIP
- Autoscale group is prepopulated from the Autoscale group configured on your AWS account. http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroup.html.
- While selecting the Autoscaling Group protocol and port, ensure your servers listen on those protocol and ports, and you bind the correct monitor in the service group. By default, the TCP monitor is used.
- For SSL Protocol type Autoscaling, after you create the Cloud Profile the load balance virtual server or service group is down because of a missing certificate. You can bind the certificate to the virtual server or service group manually.
- Select the Graceful Timeout option to remove Autoscale servers gracefully. If this option is not selected the server is the Autoscale group is removed immediately after the load goes down, which might cause service interruption for the existing connected clients. Selecting Graceful and giving a timeout means in the event of scale down. The VPX instance does not remove the server immediately but marks one of the servers for graceful deletion. During this period, the instance does not allow new connections to this server. Existing connection are served until the timeout occurs, and after a timeout, the VPX instance removes the server.

**Figure:** Default Cloud Profile page

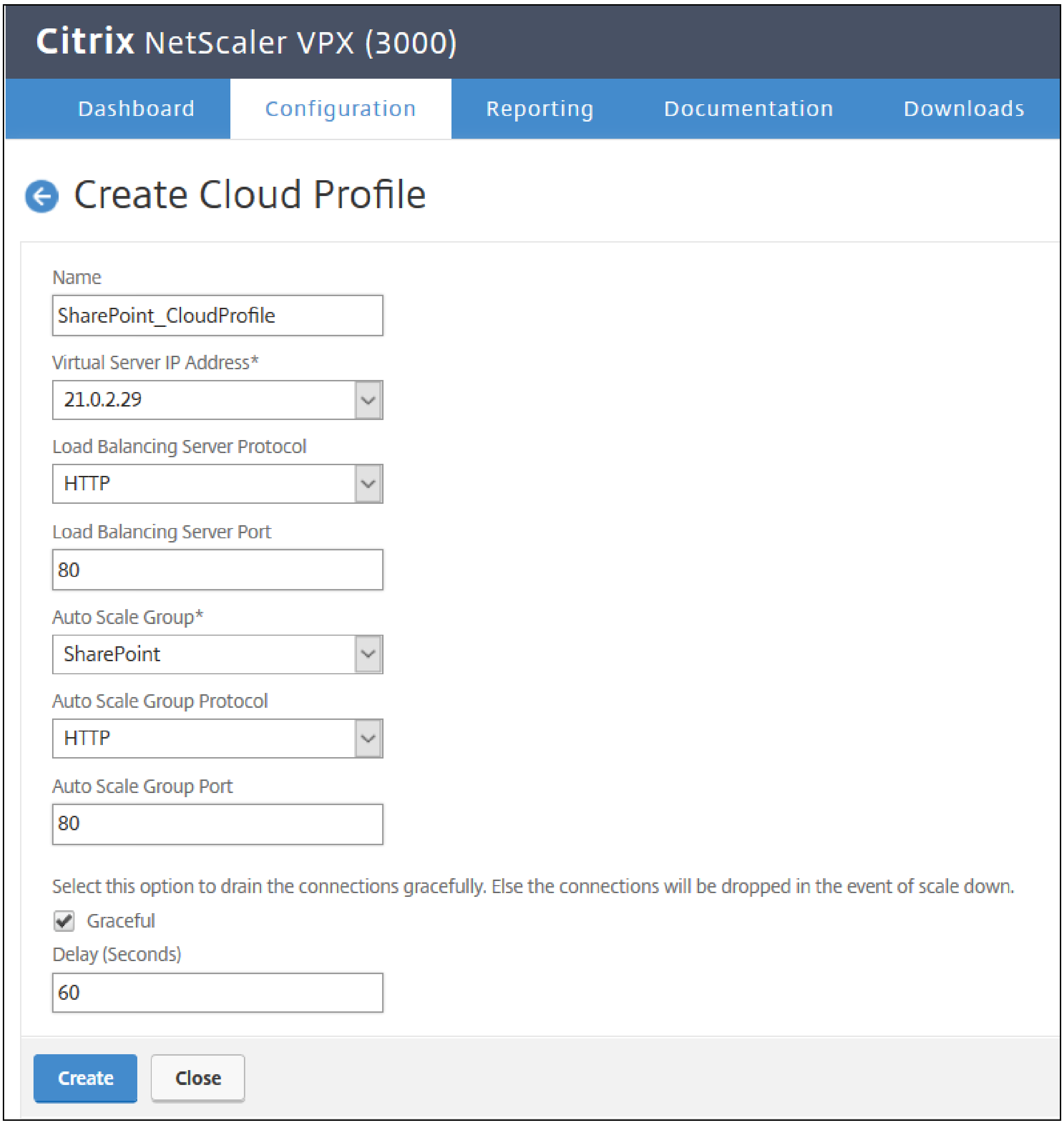



3. After the first time logon if you want to create Cloud Profile, on the GUI go to **System > AWS > Cloud Profile** and click **Add**.




The **Create Cloud Profile** configuration page appears.

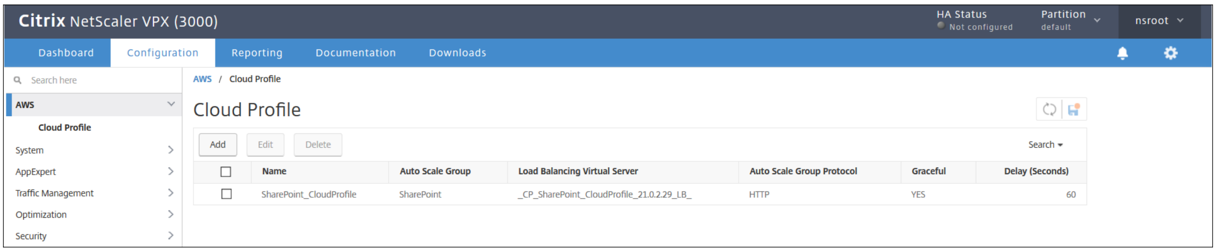
 




Cloud Profile creates a NetScaler load-balancing virtual server and a service group with members as the servers of the Autoscaling group. Your back-end servers must be reachable through the SNIP configured on the VPX instance.

> **Note:**
>
> From NetScaler release 13.1-42.x onwards, you can create different cloud profiles for different services (using different ports) with the same Autoscaling Group (ASG) in AWS. Thus, the NetScaler VPX instance supports multiple services with the same Autoscaling group in public cloud.

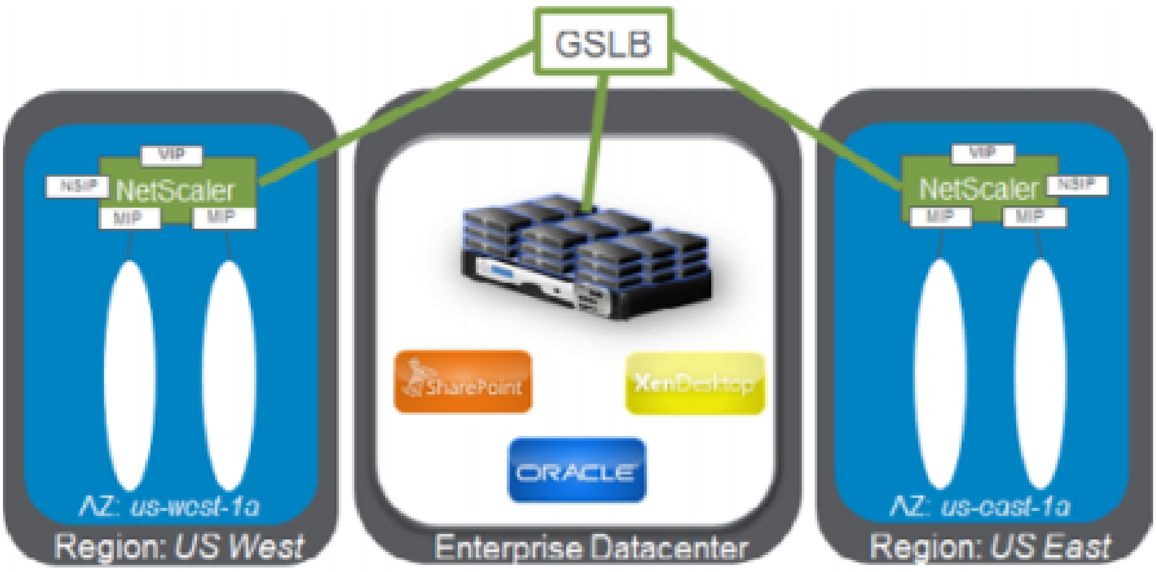
 




> **Note:**
>
> To view Autoscale-related information in the AWS console, go to **EC2** > **Dashboard** > **Auto Scaling** > **Auto Scaling Group**.

## Deploy NetScaler GSLB on AWS

Setting up GSLB for NetScaler on AWS basically consists of configuring NetScaler to load balance traffic to servers located outside the VPC that NetScaler belongs to, such as within another VPC in a different Availability Region or an on-premises data center.




### DBS overview

NetScaler GSLB support using DBS (Domain Based Services) for Cloud load balancers allows for the automatic discovery of dynamic cloud services using a cloud load balancer solution. This configuration allows NetScaler to implement Global Server Load Balancing Domain-Name Based Services (GSLB DBS) in an Active-Active environment. DBS allows the scaling of back-end resources in AWS environments from DNS discovery.

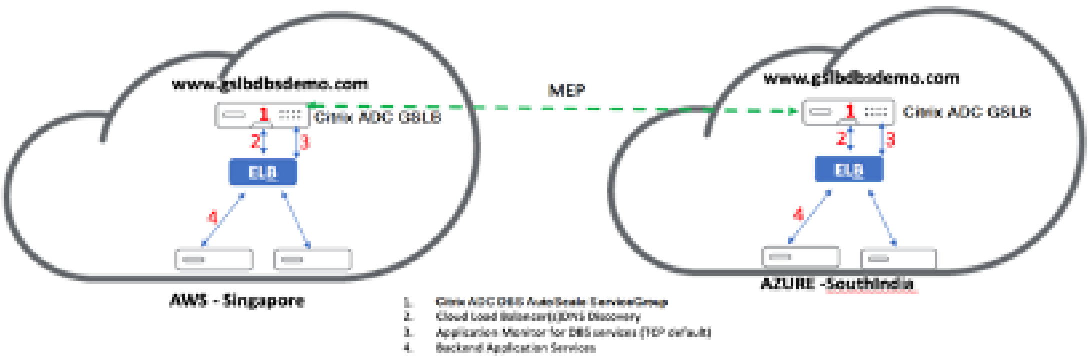
 

This section covers integrations between NetScaler in AWS AutoScaling environments. The final section of the document details the ability to set up a HA pair of NetScaler ADCs that span two different Availability Zones (AZs) specific to an AWS region.

#### Domain-name based services (DBS) with ELB

GSLB DBS utilizes the FQDN of the user Elastic Load Balancer (ELB) to dynamically update the GSLB Service Groups to include the back-end servers that are being created and deleted within AWS. The back-end servers or instances in AWS can be configured to scale based on network demand or CPU utilization. To configure this feature, point NetScaler to the ELB to dynamically route to different servers in AWS without having to manually update NetScaler every time an instance is created and deleted within AWS. NetScaler DBS feature for GSLB Service Groups uses DNS aware service discovery to determine the member service resources of the DBS namespace identified in the AutoScale group.

NetScaler GSLB DBS autoScale components with cloud load balancers:




### Configure AWS components

#### Security groups

> **Note:**
>
> We recommend you to create different security groups for ELB, NetScaler GSLB Instance, and Linux instance, as the set of rules required for each of these entities is different. This example has a consolidated Security Group configuration for brevity.

To ensure the proper configuration of the virtual firewall, see Security Groups for Your VPC.

1. Log in to the user **AWS resource group** and navigate to **EC2 > NETWORK & SECURITY > Security Groups**.
2. Click **Create Security Group** and provide a name and description. This security group encompasses NetScaler and Linux back-end web servers.

3. Add the inbound port rules from the following screenshot.

   > **Note:**
   >
   > Limiting Source IP access is recommended for granular hardening. For more information, see Web Server Rules.

4. Amazon linux back-end web services

   a) Log in to the user **AWS resource group** and navigate to **EC2 > Instances**.

   b) Click **Launch Instance** using the details that follow to configure the **Amazon Linux** instance.

   Enter the details about setting up a Web Server or back-end service on this instance.

5. NetScaler Configuration

   a) Log in to the user **AWS resource group** and navigate to **EC2 > Instances**.

   b) Click **Launch Instance** and use the following details to configure the **Amazon AMI** instance.

6. Elastic IP Configuration

   > **Note:**
   >
   > NetScaler can also be made to run with a single elastic IP if necessary to reduce cost, by not having a public IP for the NSIP. Instead, attach an elastic IP to the SNIP which can cover for management access to the box, in addition to the GSLB site IP and ADNS IP.

   a) Log in to the user **AWS resource group** and navigate to **EC2 > Network & Security > Elastic IPs**.

   b) Click **Allocate new address** to create a Elastic IP address.

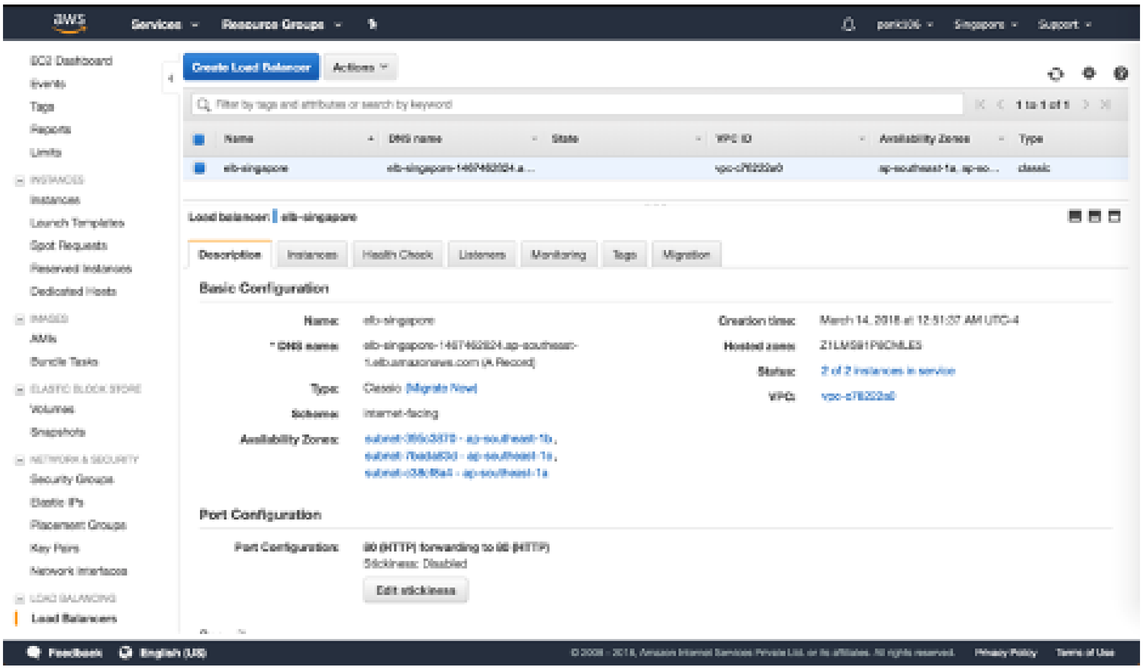
   c) Configure the Elastic IP to point to the user running NetScaler instance within AWS.

   d) Configure a second Elastic IP and again point it to the user running NetScaler instance.

7. Elastic Load Balancer

   a) Log in to the user **AWS resource group** and navigate to **EC2 > Load Balancing > Load Balancers**.




a) Click **Create Load Balancer** to configure a classic load balancer.

The user Elastic Load Balancers allow users to load balance their back-end Amazon Linux instances while also being able to Load Balance other instances that are spun up based on demand.

### Configure global server load balancing domain-name based services

For Traffic Management Configurations, see Configure NetScaler GSLB domain-based service.

### Deployment types

Three-NIC Deployment

- Typical Deployments
  - GSLB StyleBook
  - With ADM
  - With GSLB (Route53 w/domain registration)
  - Licensing - Pooled/Marketplace
- Use Cases
  - Three-NIC Deployments are used to achieve real isolation of data and management traffic.

  - Three-NIC Deployments also improve the scale and performance of the ADC.
  - Three-NIC Deployments are used in network applications where throughput is typically 1 Gbps or higher and a Three-NIC Deployment is recommended.

#### CFT deployment

Customers would deploy using CloudFormation Templates if they are customizing their deployments or they are automating their deployments.

#### Deployment steps

The followiwng are the deployments steps:

1. Three-NIC deployment for GSLB
2. Licensing
3. Deployment options

**Three-NIC deployment for GSLB** NetScaler VPX instance is available as an Amazon Machine Image (AMI) in the AWS marketplace, and it can be launched as an Elastic Compute Cloud (EC2) instance within an AWS VPC. The minimum EC2 instance type allowed as a supported AMI on NetScaler VPX is m4.large. NetScaler VPX AMI instance requires a minimum of 2 virtual CPUs and 2 GB of memory. An EC2 instance launched within an AWS VPC can also provide the multiple interfaces, multiple IP addresses per interface, and public and private IP addresses needed for VPX configuration. Each VPX instance requires at least three IP subnets:

- A management subnet
- A client-facing subnet (VIP)
- A back-end facing subnet (SNIP)

NetScaler recommends three network interfaces for a standard VPX instance on AWS installation.

AWS currently makes multi-IP functionality available only to instances running within an AWS VPC. A VPX instance in a VPC can be used to load balance servers running in EC2 instances. An Amazon VPC allows users to create and control a virtual networking environment, including their own IP address range, subnets, route tables, and network gateways.

> **Note:**
>
> By default, users can create up to 5 VPC instances per AWS region for each AWS account. Users can request higher VPC limits by submitting Amazon's request form here: Amazon VPC Request.

**Licensing** A NetScaler VPX instance on AWS requires a license. The following licensing options are available for NetScaler VPX instances running on AWS:

- Free (unlimited)
- Hourly
- Annual

Bring your own license

Free Trial (all NetScaler VPX-AWS subscription offerings for 21 days free in AWS marketplace).

**Deployment options** Users can deploy a NetScaler VPX standalone instance on AWS.
For more information, see Deploy a NetScaler VPX standalone instance on AWS

### NetScaler global load balancing for hybrid and multi-cloud dployments

NetScaler hybrid and multi-cloud global server load balancing (GSLB) solution enables users to distribute application traffic across multiple data centers in hybrid clouds, multiple clouds, and on-premises deployments. NetScaler hybrid and multi-cloud GSLB solution helps users to manage their load balancing setup in hybrid or multi-cloud environments without altering the existing setup. Also, if users have an on-premises setup, they can test some of their services in the cloud by using NetScaler hybrid and multi-cloud GSLB solution before completely migrating to the cloud. For example, users can route only a small percentage of their traffic to the cloud, and handle most of the traffic on-premises. NetScaler hybrid and multi-cloud GSLB solution also enables users to manage and monitor NetScaler instances across geographic locations from a single, unified console.

A hybrid and multi-cloud architecture can also improve overall enterprise performance by avoiding "vendor lock-in"and using different infrastructure to meet the needs of user partners and customers. With multiple cloud architecture, users can manage their infrastructure costs better as they now have to pay only for what they use. Users can also scale their applications better as they now use the infrastructure on demand. It also provides the ability to quickly switch from one cloud to another to take advantage of the best offerings of each provider.

### NetScaler hybrid and multi-cloud GSLB Solution

NetScaler GSLB nodes handle the DNS name resolution. Any of these GSLB nodes can receive DNS requests from any client location. The GSLB node that receives the DNS request returns the load balancer virtual server IP address as selected by the configured load balancing method. Metrics (site, network, and persistence metrics) are exchanged between the GSLB nodes using the metrics exchange protocol (MEP), which is a proprietary NetScaler protocol. For more information on the MEP protocol, see Configure Metrics Exchange Protocol.

The monitor configured in the GSLB node monitors the health status of the load balancing virtual server in the same data center. In a parent-child topology, metrics between the GSLB and NetScaler nodes are exchanged by using MEP. However, configuring monitor probes between a GSLB and NetScaler LB node is optional in a parent-child topology.

The NetScaler agent enables communication between the NetScaler ADM and the managed instances in the user data center. For more information on NetScaler agents and how to install them, see Getting Started.

> **Note:**
>
> This document makes the following assumptions:
>
> - If users have an existing load balancing setup, it is up and running.
> - A SNIP address or a GSLB site IP address is configured on each of NetScaler GSLB nodes. This IP address is used as the data center source IP address when exchanging metrics with other data centers.
> - An ADNS or ADNS-TCP service is configured on each of NetScaler GSLB instances to receive the DNS traffic.
> - The required firewall and security groups are configured in the cloud service providers.

#### Security groups configuration

Users must set up the required firewall/security groups configuration in the cloud service providers. For more information about AWS security features, see AWS/Documentation/Amazon VPC/User Guide/Security.

Also, on the GSLB node, users must open port 53 for ADNS service/DNS server IP address and port 3009 for GSLB site IP address for MEP traffic exchange. On the load balancing node, users must open the appropriate ports to receive the application traffic. For example, users must open port 80 for receiving HTTP traffic and open port 443 for receiving HTTPS traffic. Open port 443 for NITRO communication between the NetScaler agent and NetScaler ADM.

For the dynamic round trip time GSLB method, users must open port 53 to allow UDP and TCP probes depending on the configured LDNS probe type. The UDP or the TCP probes are initiated using one of the SNIPs and therefore this setting must be done for security groups bound to the server-side subnet.

### Capabilities of NetScaler hybrid and multi-cloud GSLB solution

Some of the capabilities of NetScaler hybrid and multi-cloud GSLB solution are described in this section.

#### Compatibility with other load balancing solutions

NetScaler hybrid and multi-cloud GSLB solution supports various load balancing solutions such as NetScaler load balancer, NGINX, HAProxy, and other third-party load balancers.

> **Note:**
>
> Load balancing solutions other than NetScaler are supported only if proximity-based and non-metric based GSLB methods are used and if parent-child topology is not configured.

#### GSLB methods

NetScaler hybrid and multi-cloud GSLB solution supports the following GSLB methods.

- Metric-based GSLB methods. Metric-based GSLB methods collect metrics from the other NetScaler nodes through the metrics exchange protocol.
  - Least Connection: The client request is routed to the load balancer that has the fewest active connections.
  - Least Bandwidth: The client request is routed to the load balancer that is currently serving the least amount of traffic.
  - Least Packets: The client request is routed to the load balancer that has received the fewest packets in the last 14 seconds.
- Non-metric based GSLB methods
  - Round Robin: The client request is routed to the IP address of the load balancer that is at the top of the list of load balancers. That load balancer then moves to the bottom of the list.
  - Source IP Hash: This method uses the hashed value of the client IP address to select a load balancer.
- Proximity-based GSLB methods
  - Static Proximity: The client request is routed to the load balancer that is closest to the client IP address.
  - Round-Trip Time (RTT): This method uses the RTT value (the time delay in the connection between the client's local DNS server and the data center) to select the IP address of the best performing load balancer.

For more information on the load balancing methods, see Load Balancing Algorithms.

#### GSLB topologies

NetScaler hybrid and multi-cloud GSLB solution supports the active-passive topology and parent-child topology.

- Active-passive topology - Provides disaster recovery and ensures continuous availability of applications by protecting against points of failure. If the primary data center goes down, the passive data center becomes operational. For more information about GSLB active-passive topology, see Configure GSLB for Disaster Recovery.
- Parent-child topology –Can be used if customers are using the metric-based GSLB methods to configure GSLB and LB nodes and if the LB nodes are deployed on a different NetScaler instance. In a parent-child topology, the LB node (child site) must be a NetScaler appliance because the exchange of metrics between the parent and child site is through the metrics exchange protocol (MEP).

For more information about parent-child topology, see Parent-Child Topology Deployment using the MEP Protocol.

#### IPv6 support

NetScaler hybrid and multi-cloud GSLB solution also supports IPv6.

#### Monitoring

NetScaler hybrid and multi-cloud GSLB solution supports built-in monitors with an option to enable the secure connection. However, if LB and GSLB configurations are on the same NetScaler instance or if parent-child topology is used, configuring monitors is optional.

#### Persistence

NetScaler hybrid and multi-cloud GSLB solution supports the following:

- Source IP based persistence sessions, so that multiple requests from the same client are directed to the same service if they arrive within the configured time-out window. If the time-out value expires before the client sends another request, the session is discarded, and the configured load balancing algorithm is used to select a new server for the client's next request.
- Spillover persistence so that the backup virtual server continues to process the requests it receives, even after the load on the primary falls below the threshold. For more information, see Configure Spillover.

- Site persistence so that the GSLB node selects a data center to process a client request and forwards the IP address of the selected data center for all subsequent DNS requests. If the configured persistence applies to a site that is DOWN, the GSLB node uses a GSLB method to select a new site, and the new site becomes persistent for subsequent requests from the client.

### Configuration by using NetScaler ADM styleBooks

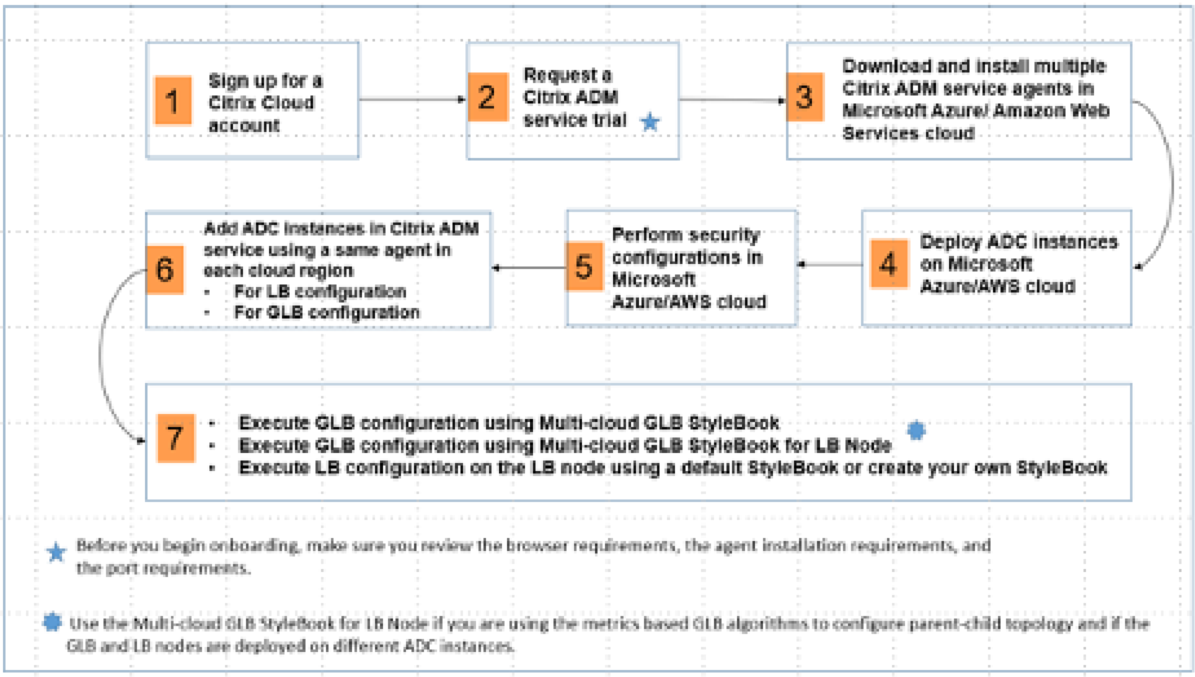
Customers can use the default Multi-cloud GSLB StyleBook on NetScaler ADM to configure NetScaler instances with hybrid and multi-cloud GSLB configurations.

Customers can use the default Multi-cloud GSLB StyleBook for the LB Node StyleBook to configure NetScaler load balancing nodes which are the child sites in a parent-child topology that handle the application traffic. Use this StyleBook only if users want to configure LB nodes in a parent-child topology. However, each LB node must be configured separately using this StyleBook.

### Workflow of NetScaler hybrid and multi-cloud GSLB solution configuration

Customers can use the shipped Multi-cloud GSLB StyleBook on NetScaler ADM to configure NetScaler instances with hybrid and multi-cloud GSLB configurations.

The following diagram shows the workflow for configuring a NetScaler hybrid and multi-cloud GSLB solution. The steps in the workflow diagram are explained in more detail after the diagram.




Perform the following tasks as a cloud administrator:

1. Sign up for a NetScaler Cloud account.

   To start using NetScaler ADM, create a NetScaler Cloud company account or join an existing one that has been created by someone in your company.

2. After users log on to NetScaler Cloud, click **Manage** on the **NetScaler Application Delivery Management** tile to set up the ADM service for the first time.

3. Download and install multiple NetScaler ADM service agents.

   Users must install and configure the NetScaler ADM service agent in their network environment to enable communication between the NetScaler ADM and the managed instances in their data center or cloud. Install an agent in each region, so that they can configure LB and GSLB configurations on the managed instances. The LB and GSLB configurations can share a single agent. For more information on the above three tasks, see Getting Started.

4. Deploy load balancers on Microsoft AWS cloud/on-premises data centers.

   Depending on the type of load balancers that users are deploying on cloud and on-premises, provision them accordingly. For example, users can provision NetScaler VPX instances in an Amazon Web Services (AWS) virtual private cloud and in on-premises data centers. Configure NetScaler instances to function as LB or GSLB nodes in standalone mode, by creating the virtual machines and configuring other resources. For more information on how to deploy NetScaler VPX instances, see the following documents:

   - NetScaler VPX on AWS.
   - Configure a NetScaler VPX Standalone Instance.

5. Perform security configurations.

   Configure network security groups and network ACLs in ARM and in AWS to control inbound and outbound traffic for user instances and subnets.

6. Add NetScaler instances in NetScaler ADM.

   NetScaler instances are network appliances or virtual appliances that users want to discover, manage, and monitor from NetScaler ADM. To manage and monitor these instances, users must add the instances to the service and register both LB (if users are using NetScaler for LB) and GSLB instances. For more information on how to add NetScaler instances in the NetScaler ADM, see Getting Started

7. Implement the GSLB and LB configurations using default NetScaler ADM StyleBooks.

   - Use Multi-cloud GSLB StyleBook to execute the GSLB configuration on the selected GSLB NetScaler instances.

- Implement the load balancing configuration. (Users can skip this step if they already have LB configurations on the managed instances.) Users can configure load balancers on NetScaler instances in one of two ways:
- Manually configure the instances for load balancing the applications. For more information on how to manually configure the instances, see Set up Basic Load Balancing.
- Use StyleBooks. Users can use one of the NetScaler ADM StyleBooks (HTTP/SSL Load Balancing StyleBook or HTTP/SSL Load Balancing (with Monitors) StyleBook) to create the load balancer configuration on the selected NetScaler instance. Users can also create their own StyleBooks. For more information on StyleBooks, see StyleBooks.

8. Use Multi-cloud GSLB StyleBook for LB Node to configure GSLB parent-child topology in any of the following cases:
   - If users are using the metric-based GSLB algorithms (Least Packets, Least Connections, Least Bandwidth) to configure GSLB and LB nodes and if the LB nodes are deployed on a different NetScaler instance.
   - If site persistence is required.

#### Using styleBooks to configure GSLB on NetScaler LB nodes

Customers can use the **Multi-cloud GSLB StyleBook for LB Node** if they are using the metric-based GSLB algorithms (Least Packets, Least Connections, Least Bandwidth) to configure GSLB and LB nodes and if the LB nodes are deployed on a different NetScaler instance.

Users can also use this StyleBook to configure more child sites for an existing parent site. This StyleBook configures one child site at a time. So, create as many configurations (config packs) from this StyleBook as there are child sites. The StyleBook applies the GSLB configuration on the child sites. Users can configure a maximum of 1024 child sites.

> **Note:**
>
> Use Multi-cloud GSLB StyleBook to configure the parent sites.

This StyleBook makes the following assumptions:

- A SNIP address or a GSLB site IP address is configured.
- The required firewall and security groups are configured in the cloud service providers.

#### Configuring a child site in a parent-child topology by using multi-cloud GSLB styleBook for LB node

1. Navigate to **Applications > Configuration > Create New**.

2. Navigate to **Applications** > **Configuration**, and click **Create New**.

   The StyleBook appears as a user interface page on which users can enter the values for all the parameters defined in this StyleBook.

   > **Note:**
   >
   > The terms data center and sites are used interchangeably in this document.

3. Set the following parameters:

   - **Application Name**. Enter the name of the GSLB application deployed on the GSLB sites for which you want to create child sites.
   - **Protocol**. Select the application protocol of the deployed application from the drop-down list box.
   - **LB Health Check** (Optional)
   - **Health Check Type**. From the drop-down list box, select the type of probe used for checking the health of the load balancer VIP address that represents the application on a site.
   - **Secure Mode**. (Optional) Select **Yes** to enable this parameter if SSL based health checks are required.
   - **HTTP Request**. (Optional) If users selected HTTP as the health-check type, enter the full HTTP request used to probe the VIP address.
   - **List of HTTP Status Response Codes**. (Optional) If users selected HTTP as the health check type, enter the list of HTTP status codes expected in responses to HTTP requests when the VIP is healthy.

4. Configuring parent site.

   - Provide the details of the parent site (GSLB node) under which you want to create the child site (LB node).
     - **Site Name**. Enter the name of the parent site.
     - **Site IP Address**. Enter the IP address that the parent site uses as its source IP address when exchanging metrics with other sites. This IP address is assumed to be already configured on the GSLB node in each site.
     - **Site Public IP Address**. (Optional) Enter the Public IP address of the parent site that is used to exchange metrics, if that site's IP address is NAT'ed.

5. Configuring child site.

   - Provide the details of the child site.

    - **Site name**. Enter the name of the site.
    - **Site IP Address**. Enter the IP address of the child site. Here, use the private IP address or SNIP of NetScaler node that is being configured as a child site.
    - **Site Public IP Address**. (Optional) Enter the Public IP address of the child site that is used to exchange metrics, if that site's IP address is NAT'ed.

6. Configuring active GSLB services (optional)

    - Configure active GSLB services only if the LB virtual server IP address is not a public IP address. This section allows users to configure the list of local GSLB services on the sites where the application is deployed.
        - **Service IP**. Enter the IP address of the load balancing virtual server on this site.
        - **Service Public IP Address**. If the virtual IP address is private and has a public IP address NAT'ed to it, specify the public IP address.
        - **Service Port**. Enter the port of the GSLB service on this site.
        - **Site Name**. Enter the name of the site on which the GSLB service is located.

7. Click **Target Instances** and select NetScaler instances configured as GSLB instances on each site on which to deploy the GSLB configuration.
8. Click **Create** to create the LB configuration on the selected NetScaler instance (LB node). Users can also click **Dry Run** to check the objects that would be created in the target instances. The StyleBook configuration that users have created appears in the list of configurations on the Configurations page. Users can examine, update, or remove this configuration by using the NetScaler ADM GUI.

### CloudFormation template deployment

NetScaler VPX is available as Amazon Machine Images (AMI) in the AWS Marketplace. Before using a CloudFormation template to provision a NetScaler VPX in AWS, the AWS user has to accept the terms and subscribe to the AWS Marketplace product. Each edition of NetScaler VPX in the Marketplace requires this step.

Each template in the CloudFormation repository has collocated documentation describing the usage and architecture of the template. The templates attempt to codify recommended deployment architecture of NetScaler VPX, or to introduce the user to NetScaler or to demonstrate a particular feature, edition, or option. Users can reuse, modify, or enhance the templates to suit their particular production and testing needs. Most templates require full EC2 permissions in addition to permissions to create IAM roles.

The CloudFormation templates contain AMI Ids that are specific to a particular release of NetScaler VPX (for example, release 12.0-56.20) and edition (for example, NetScaler VPX Platinum Edition - 10 Mbps) OR NetScaler BYOL. To use a different version / edition of NetScaler VPX with a CloudFormation template requires the user to edit the template and replace the AMI IDs.

The latest NetScaler AWS-AMI-IDs are located here: NetScaler AWS CloudFormation Master.

**CFT three-NIC deployment**

This template deploys a VPC, with 3 subnets (Management, client, server) for 2 Availability Zones. It deploys an Internet Gateway, with a default route on the public subnets. This template also creates a HA pair across Availability Zones with two instances of NetScaler: 3 ENIs associated to 3 VPC subnets (Management, Client, Server) on primary and 3 ENIs associated to 3 VPC subnets (Management, Client, Server) on secondary. All the resource names created by this CFT are prefixed with a tagName of the stack name.

The output of the CloudFormation template includes:

- PrimaryCitrixADCManagementURL - HTTPS URL to the Management GUI of the Primary VPX (uses self-signed cert)
- PrimaryCitrixADCManagementURL2 - HTTP URL to the Management GUI of the Primary VPX
- PrimaryCitrixADCInstanceID - Instance Id of the newly created Primary VPX instance
- PrimaryCitrixADCPublicVIP - Elastic IP address of the Primary VPX instance associated with the VIP
- PrimaryCitrixADCPrivateNSIP - Private IP (NS IP) used for management of the Primary VPX
- PrimaryCitrixADCPublicNSIP - Public IP (NS IP) used for management of the Primary VPX
- PrimaryCitrixADCPrivateVIP - Private IP address of the Primary VPX instance associated with the VIP
- PrimaryCitrixADCSNIP - Private IP address of the Primary VPX instance associated with the SNIP
- SecondaryCitrixADCManagementURL - HTTPS URL to the Management GUI of the Secondary VPX (uses self-signed cert)
- SecondaryCitrixADCManagementURL2 - HTTP URL to the Management GUI of the Secondary VPX
- SecondaryCitrixADCInstanceID - Instance Id of the newly created Secondary VPX instance
- SecondaryCitrixADCPrivateNSIP - Private IP (NS IP) used for management of the Secondary VPX
- SecondaryCitrixADCPublicNSIP - Public IP (NS IP) used for management of the Secondary VPX

- SecondaryCitrixADCPrivateVIP - Private IP address of the Secondary VPX instance associated with the VIP
- SecondaryCitrixADCSNIP - Private IP address of the Secondary VPX instance associated with the SNIP
- SecurityGroup - Security group id that the VPX belongs to

When providing input to the CFT, the * against any parameter in the CFT implies that it is a mandatory field. For example, `VPC ID*` is a mandatory field.

The following prerequisites must be met. The CloudFormation template requires sufficient permissions to create IAM roles, beyond normal EC2 full privileges. The user of this template also needs to accept the terms and subscribe to the AWS Marketplace product before using this CloudFormation template.

The following should also be present:

- Key Pair
- 3 unallocated EIPs
- Primary Management
- Client VIP
- Secondary Management

For more information on provisioning NetScaler VPX instances on AWS, users can visit: Provisioning NetScaler VPX Instances on AWS.

For information on how to configure GSLB using stylebooks visit Using StyleBooks to Configure GSLB

### Prerequisites

Before attempting to create a VPX instance in AWS, users should ensure they have the following:

- An AWS account to launch a NetScaler VPX AMI in an Amazon Web Services (AWS) Virtual Private Cloud (VPC). Users can create an AWS account for free at Amazon.
- An AWS Identity and Access Management (IAM) user account to securely control access to AWS services and resources for users. For more information about how to create an IAM user account, see the topic: Creating IAM Users (Console).

An IAM role is mandatory for both standalone and high availability deployments. The IAM role must have the following privileges:

- ec2:DescribeInstances

- ec2:DescribeNetworkInterfaces
- ec2:DetachNetworkInterface
- ec2:AttachNetworkInterface
- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- ec2:DescribeAddresses
- ec2:AssociateAddress
- ec2:DisassociateAddress
- autoscaling:*
- sns:*
- sqs:*
- iam:SimulatePrincipalPolicy
- iam:GetRole

If the NetScaler CloudFormation template is used, the IAM role is automatically created. The template does not allow selecting an already created IAM role.

> **Note:**
>
> When users log on the VPX instance through the GUI, a prompt to configure the required privileges for IAM role appears. Ignore the prompt if the privileges have already been configured.

- AWS CLI is required to use all the functionality provided by the AWS Management Console from the terminal program. For more information, see What Is the AWS Command Line Interface?. Users also need the AWS CLI to change the network interface type to SR-IOV.

#### GSLB prerequisites

The prerequisites for NetScaler GSLB Service Groups include a functioning AWS environment with the knowledge and ability to configure Security Groups, Linux Web Servers, NetScaler ADCs within AWS, Elastic IPs, and Elastic Load Balancers.

GSLB DBS Service integration requires NetScaler version 12.0.57 for AWS ELB load balancer instances.

For latest update about the current supported VPX models and AWS regions, instance types, and services see VPX-AWS support matrix.

### Other resources

NetScaler ADM GSLB for Hybrid and Multi-Cloud Deployments.

## Configure a NetScaler VPX instance to use SR-IOV network interface

> **Note:**
>
> Support for SR-IOV interfaces in a high availability setup is available from NetScaler release 12.0 57.19 onwards.

After you have created a NetScaler VPX instance on AWS, you can configure the virtual appliance to use SR-IOV network interfaces, by using the AWS CLI.

In all NetScaler VPX models, except NetScaler VPX AWS Marketplace Editions of 3G and 5G, SR-IOV is not enabled in the default configuration of a network interface.

Before you start the configuration, read the following topics:

- Prerequisites
- Limitations and Usage Guidelines

This section includes the following topics:

- Change the Interface Type to SR-IOV
- Configure SR-IOV on a High Availability Setup

### Change the interface type to SR-IOV

You can run the show interface summary command to check the default configuration of a network interface.

**Example 1:** The following CLI screen capture shows the configuration of a network interface where SR-IOV is enabled by default on NetScaler VPX AWS Marketplace Editions of 3G and 5G.

```
> show interface summary
--------------------------------------------------------------------------------
     Interface  MTU       MAC                  Suffix
--------------------------------------------------------------------------------
1     1/1       1500      0a:1e:2e:17:a2:37    Intel 82599 10G VF Interface
2     LO/1      1500      0a:1e:2e:17:a2:37    Netscaler Loopback interface
 Done
```

**Example 2:** The following CLI screen capture shows the default configuration of a network interface where SR-IOV is not enabled.

```
 Done
> sh int s
--------------------------------------------------------------------------------
      Interface  MTU       MAC                Suffix
--------------------------------------------------------------------------------
1     1/1        1500      12:fc:04:c5:d0:12  NetScaler Virtual Interface
2     LO/1       1500      12:fc:04:c5:d0:12  Netscaler Loopback interface
 Done
>
```

For more information about changing the interface type to SR-IOV, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.html

**To change the interface type to SR-IOV**

1. Shut down the NetScaler VPX instance running on AWS.
2. To enable SR-IOV on the network interface, type the following command in the AWS CLI.

   ```
   $ aws ec2 modify-instance-attribute --instance-id \<instance\\_id
   \> --sriov-net-support simple
   ```

3. To check if SR-IOV has been enabled, type the following command in the AWS CLI.

   ```
   $ aws ec2 describe-instance-attribute --instance-id \<instance\\
   _id\> --attribute sriovNetSupport
   ```

   **Example 3:** Network interface type changed to SR-IOV, by using the AWS CLI.

   ```
   aws ec2 modify-instance-attribute --instance-id i-008c1230aaf303bee --sriov-net-support simple
   aws ec2 describe-instance-attribute --instance-id   i-008c1230aaf303bee  --attribute sriovNetSupport
   {
       "InstanceId": "i-008c1230aaf303bee",
       "SriovNetSupport": {
           "Value": "simple"
       }
   }
   ```

   If SR-IOV is not enabled, value for SriovNetSupport is absent.

   **Example 4:** In the following example, SR-IOV support is not enabled.

   ```
   {
       "InstanceId": "i-0c3e84cfa65b04cc8",
       "SriovNetSupport": {}
   }
   ```

4. Power on the VPX instance. To see the changed status of the network interface, type "show interface summary" in the CLI.

**Example 5:** The following screen capture shows the network interfaces with SR-IOV enabled. The interfaces 10/1, 10/2, 10/3 are SR-IOV enabled.

```
> show interface summary
--------------------------------------------------------------------------------
     Interface  MTU        MAC                 Suffix
--------------------------------------------------------------------------------
1    10/1       1500       0a:1e:2e:17:a2:37   Intel 82599 10G VF Interface
2    10/2       1500       0a:df:17:0a:fe:83   Intel 82599 10G VF Interface
3    10/3       1500       0a:de:5d:31:bf:c3   Intel 82599 10G VF Interface
4    LO/1       1500       0a:1e:2e:17:a2:37   Netscaler Loopback interface
 Done
```

These steps complete the procedure to configure VPX instances to use SR-IOV network interfaces.

### Configure SR-IOV on a high availability setup

High availability is supported with SR-IOV interfaces from NetScaler release 12.0 build 57.19 onwards.

If the high availability setup was deployed manually or by using the Citrix CloudFormation template for NetScaler version 12.0 56.20 and lower, the IAM role attached to the high availability setup must have the following privileges:

- ec2:DescribeInstances
- ec2:DescribeNetworkInterfaces
- ec2:DetachNetworkInterface
- ec2:AttachNetworkInterface
- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- autoscaling:*
- sns:*
- sqs:*
- IAM:SimulatePrincipalPolicy
- IAM:GetRole

By default, the Citrix CloudFormation template for NetScaler version 12.0 57.19 automatically adds the required privileges to the IAM role.

> **Note:**
>
> A high availability setup with SR-IOV Interfaces takes around 100 seconds of downtime.

**Related resources**:

For more information about IAM roles, see AWS documentation.

## Configure a NetScaler VPX instance to use Enhanced Networking with AWS ENA

After you have created a NetScaler VPX instance on AWS, you can configure the virtual appliance to use Enhanced Networking with AWS Elastic Network Adapter (ENA), by using AWS CLI.

Coupled with AWS ENA, enhanced networking provides higher bandwidth, higher packet-per-second (PPS) performance, and consistently lower inter-instance latencies.

Before you start the configuration, read the following topics:

- Prerequisites
- Limitations and Usage Guidelines

The following HA configurations are supported for ENA-enabled instances:

- Private IP addresses can be moved within the same availability zone.
- Elastic IP addresses can be moved across availability zones.

## Upgrade a NetScaler VPX instance on AWS

You can upgrade the EC2 instance type, throughput, software edition, and the system software of a NetScaler VPX running on AWS. For certain types of upgrades, Citrix recommends using the High Availability Configuration method to minimize downtime.

> **Note:**
>
> - NetScaler software release 10.1.e-124.1308.e or later for a NetScaler VPX AMI (including both utility license and customer license) does not support the M1 and M2 instance families.
> - Because of changes in VPX instance support, downgrading from 10.1.e-124 or a later release to 10.1.123.x or an earlier release is not supported.
> - Most of the upgrades do not require the launch of a new AMI, and the upgrade can be done on the current NetScaler AMI instance. If you do want to upgrade to a new NetScaler AMI instance, use the high availability configuration method.

### Change the EC2 instance type of a NetScaler VPX instance on AWS

If your NetScaler VPX instances are running release 10.1.e-124.1308.e or later, you can change the EC2 instance type from the AWS console as follows:

1. Stop the VPX instance.

2. Change the EC2 instance type from the AWS console.
3. Start the instance.

You can also use the above procedure to change the EC2 instance type for a release, earlier than 10.1.e-124.1308.e, unless you want to change the instance type to M3. In that case, you must first follow the standard NetScaler upgrade procedure, at, to upgrade the NetScaler software to 10.1.e-124 or a later release, and then follow the above steps.

### Upgrade the throughput or software edition of a NetScaler VPX instance on AWS

To upgrade the software edition (for example, to upgrade from Standard to Premium edition) or throughput (for example, to upgrade from 200 Mbps to 1000mbps), the method depends on the instance's license.

#### Using a customer license (Bring-Your-Own-License)

If you are using a customer license, you can purchase and download the new license from the Citrix website, and then install the license on the VPX instance. For more information about downloading and installing a license from the Citrix website, see the VPX Licensing Guide.

#### Using a utility license (Utility license with hourly fee)

AWS does not support direct upgrades for fee-based instances. To upgrade the software edition or throughput of a fee based NetScaler VPX instance, launch a new AMI with the desired license and capacity and migrate the older instance configuration to the new instance. This can be achieved by using a NetScaler high availability configuration as described in [Upgrade to a new NetScaler AMI instance by using a NetScaler high availability configuration] (#upgrade-to-a-new-citrix-adc-ami-instance-by-using-a-citrix-adc-high-availability-configuration) subsection in this page.

### Upgrade the system software of a NetScaler VPX instance on AWS

If you need to upgrade a VPX instance running 10.1.e-124.1308.e or a later release, follow the standard NetScaler upgrade procedure at Upgrade and downgrade a NetScaler appliance.

If you need to upgrade a VPX instance running a release older than 10.1.e-124.1308.e to 10.1.e-124.1308.e or a later release, first upgrade the system software, and then change the instance type to M3 as follows:

1. Stop the VPX instance.
2. Change the EC2 instance type from the AWS console.
3. Start the instance.

### Upgrade to a new NetScaler AMI instance by using a NetScaler high availability configuration

To use the high availability method of upgrading to a new NetScaler AMI instance, perform the following tasks:

- Create a new instance with the desired EC2 instance type, software edition, throughput, or software release from the AWS marketplace.
- Configure high availability between the old instance (to be upgraded) and the new instance. After high availability is configured between the old and the new instance, configuration from the old instance is synchronized to the new instance.
- Force an HA failover from the old instance to the new instance. As a result, the new instance becomes primary and starts receiving traffic.
- Stop, and reconfigure or remove the old instance from AWS.

#### Prerequisites and points to consider

- Ensure you understand how high availability works between two NetScaler VPX instances on AWS. For more information about high availability configuration between two NetScaler VPX instances on AWS, see Deploy a high availability pair on AWS.
- You must create the new instance in the same availability zone as the old instance, having the exact same security group and subnet.
- High availability setup requires access and secret keys associated with the user's AWS Identity and Access Management (IAM) account for both instances. If the correct key information is not used when creating VPX instances, the HA setup fails. For more information about creating an IAM account for a VPX instance, see Prerequisites.
  - You must use the EC2 console to create the new instance. You cannot use the AWS 1-click launch, because it does not accept the access and secret keys as the input.
  - The new instance must have only one ENI interface.

To upgrade a NetScaler VPX Instance by using a high availability configuration, follow these steps:

1. Configure high availability between the old and the new instance. To configure high availability between two NetScaler VPX instances, at the command prompt of each instance, type:

   - `add ha node <nodeID> <IPaddress of the node to be added>`
   - `save config`

   **Example**:

   At the command prompt of the old instance, type:

```
add ha node 30 192.0.2.30
Done
```

At the command prompt of the new instance, type:

```
add ha node 10 192.0.2.10
Done
```

Note the following:

- In the HA setup, the old instance is the primary node and the new instance is the secondary node.
- The NSIP IP address is not copied from the old instance to the new instance. Therefore, after the upgrade, your new instance has a different management IP address from the previous one.
- The `nsroot` account password of the new instance is set to that of the old instance after HA synchronization.

For more information about high availability configuration between two NetScaler VPX instances on AWS, see Deploy a high availability pair on AWS.

2. Force an HA failover. To force a failover in a high availability configuration, at the command prompt of either of the instances, type:

```
force HA failover
```

As the result of forcing a failover, the ENIs of the old instance are migrated to the new instance and traffic flows through the new instance (the new primary node). The old instance (the new secondary node) restarts.

If the following warning message appears, type N to abort the operation:

```
[WARNING]:Force Failover may cause configuration loss, peer health
    not optimum. Reason(s):
HA version mismatch
HA heartbeats not seen on some interfaces
Please confirm whether you want force-failover (Y/N)?
```

The warning message appears because the system software of the two VPX instances is not HA compatible. As a result, the configuration of the old instance cannot be automatically synced to the new instance during a forced failover.

Following is the workaround for this issue:

a) At the NetScaler shell prompt of the old instance, type the following command to create a backup of the configuration file (ns.conf):

`copy /nsconfig/ns.conf to /nsconfig/ns.conf.bkp`

b) Remove the following line from the backup configuration file (ns.conf.bkp):

- `set ns config -IPAddress <IP> -netmask <MASK>`

For example, `set ns config -IPAddress 192.0.2.10 -netmask 255.255.255.0`

c) Copy the old instance's backup configuration file (ns.conf.bkp) to the /nsconfig directory of the new instance.

d) At the NetScaler shell prompt of the new instance, type the following command to load the old instance's configuration file (ns.conf.bkp) on the new instance:

- `batch -f /nsconfig/ns.conf.bkp`

e) Save the configuration on the new instance.

- `save conifg`

f) At the command prompt of either of the nodes, type the following command to force a failover, and then type Y for the warning message to confirm the force failover operation:

- `force ha failover`

**Example**:

```
        > force ha failover

WARNING]:Force Failover may cause configuration loss, peer health
    not optimum.
        Reason(s):
        HA version mismatch
        HA heartbeats not seen on some interfaces
        Please confirm whether you want force-failover (Y/N)? Y
```

3. Remove the HA configuration, so that the two instances are no longer in an HA configuration. First remove the HA configuration from the secondary node and then remove the HA configuration from the primary node.

   To remove an HA configuration between two NetScaler VPX instances, at the command prompt of each instance, type:

```
    > remove ha node \<nodeID\>
    >  save config
```

   For more information about high availability configuration between two VPX instances on AWS, see Deploy a high availability pair on AWS.

   **Example**:

At the command prompt of the old instance (new secondary node), type:

```
> remove ha node 30
  Done
> save config
  Done
```

At the command prompt of the new instance (new primary node), type:

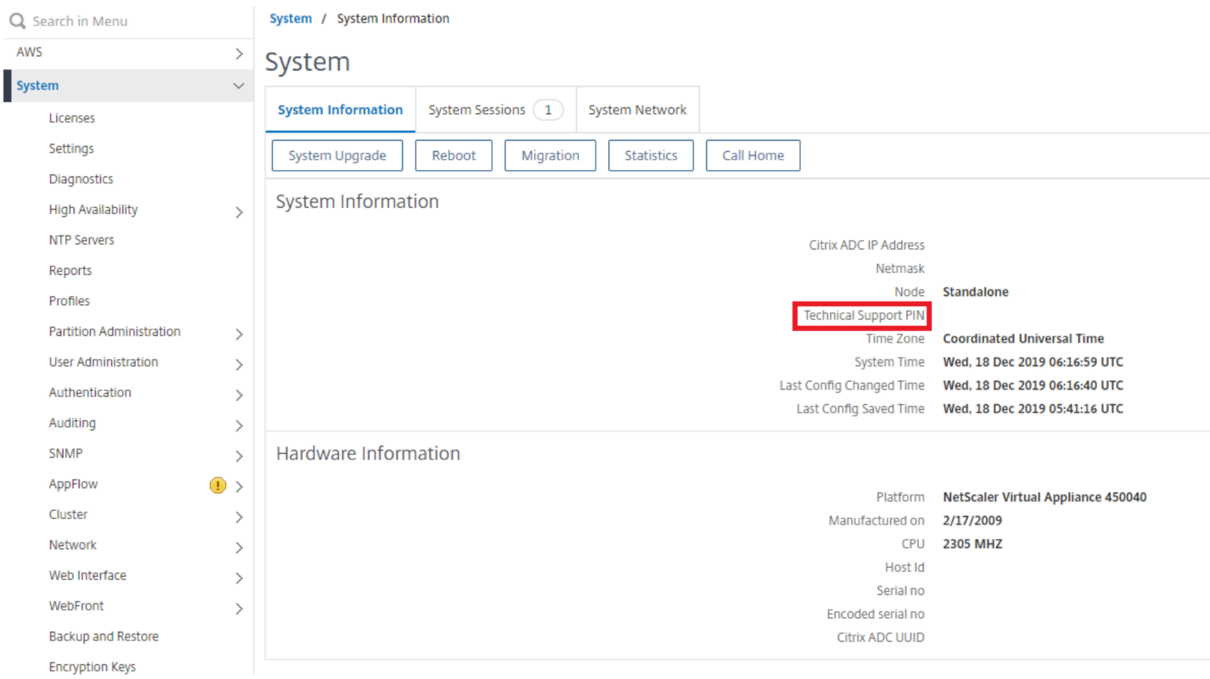
```
> remove ha node 10
  Done
> save config
  Done
```

## Troubleshoot a VPX instance on AWS

Amazon does not provide console access to a NetScaler VPX instance. To troubleshoot, you have to use the AWS GUI to view the activity log. You can debug only if the network is connected. To view an instance's System Log, right-click the instance and select System Log.

NetScaler provides support for AWS Marketplace-licensed NetScaler VPX instances (utility license with hourly fee) on AWS. To file a support case, find your AWS account number and support PIN code, and call NetScaler support. You will also be asked for your name and email address. To find the support PIN, log on to the VPX GUI and navigate to the System page.

Here is an example of a system page showing the support PIN.



## AWS FAQs

- **Does a NetScaler VPX instance support the encrypted volumes in AWS?**

  Encryption and decryption happen at the hypervisor level, and hence it works seamlessly with any instance. For more information about the encrypted volumes see the following AWS document:

  https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html

- **What is the best way to provision NetScaler VPX instance on AWS?**

  You can provision a NetScaler VPX instance on AWS by any of the following ways:

  - AWS CloudFormation Template (CFT) in AWS marketplace
  - NetScaler ADM
  - AWS Quick Starts
  - Citrix AWS CFTs in GitHub
  - Citrix Terraform Scripts in GitHub
  - Citrix Ansible Playbooks in GitHub
  - AWS EC2 launch workflow

  You can choose any of the listed options based on the automation tool that you use.

  For more details about the options, see NetScaler VPX on AWS.

- **How to upgrade NetScaler VPX instance in AWS?**

  To upgrade the NetScaler VPX instance in AWS, you can upgrade the system software or upgrade to a new NetScaler VPX Amazon Machine Image (AMI) by following the procedure at Upgrade a NetScaler VPX instance on AWS.

  The recommended way to upgrade a NetScaler VPX instance is using the ADM service by following the procedure at Use jobs to upgrade NetScaler instances.

- **What is the HA failover time for NetScaler VPX in AWS?**

  - HA failover of NetScaler VPX within the AWS availability zone takes around 3 seconds.
  - HA failover of NetScaler VPX across AWS availability zones takes around 5 seconds.

- **What level of support is provided for NetScaler VPX marketplace subscription customers who provide the technical support PIN?**

  By default, the "Select for Software" service is provided to customers who provide the technical support PIN.

- **In High availability across different zones using Elastic IP deployment, do we need to create Multiple IPSets for each application?**

Yes. If there are multiple applications with multiple VIPs mapped to multiple EIPs then multiple IPSets are required. Therefore during HA failover, all the primary VIP mappings of EIPs are changed to secondary (new primary) VIPs.

- **Why is INC mode enabled in high availability across different zone deployments?**

  HA pairs across availability zones are in different networks. For HA synchronization, network configuration must not be synchronized. This is achieved by enabling INC mode on HA pair.

- **Can HA node in one availability zone communicate with back-end servers in another availability zone, provided those availabilty zones are in same VPC?**

  Yes, subnets in different availability zones of the same VPC are reachable by adding an extra route pointing to the backend-server subnet via SNIP. For example, if the SNIP subnet of ADC in AZ1 is 192.168.3.0/24 and the backend-server subnet in AZ2 is 192.168.6.0/24, then a route must be added in the NetScaler appliance present in AZ1 as 192.168.6.0 255.255.255.0 192.168.3.1.

- **Can High availability across different zones using Elastic IP and High availability across different zones using Private IP deployments work together?**

  Yes, both the configurations can be applied on the same HA Pair.

- **In High availability across different zones using Private IP deployment, if there are multiple subnets with multiple route tables in a VPC, how does a secondary node in HA pair know about the route table to be checked during HA failover?**

  Secondary node is aware of the primary NICs and searches across all the route tables in a VPC.

- **What is the size of the `/var` partition when using the default image for VPX on AWS? How to increase the disk space?**

  The size of the root disk is limited to 20 GB to keep the disk image small.

  If you want to increase the `/var/core/` or the `/var/crash/` directory space, attach an extra disk. To increase the `/var` size, currently, you must attach an extra disk and create a symbolic link to `/var`, after copying the critical contents to the new disk.

- **How many packet engines are activated and allocated to vCPUs?**

  The packet engines (PEs) are limited by the number of licensed vCPUs. The NetScaler daemons are not pinned to any particular vCPU and might run on any of the non-PE vCPUs. According to AWS, the C5.9xlarge is a 36vCPU instance with 72 GB memory. With pooled licensing, the NetScaler VPX instance deploys with the maximum number of PEs. In this case, 19 PEs run on cores 1–19. However, ADC management processes run from CPUs 20–31.

- **How to decide the right AWS instance for ADC?**

  1. Understand your use case and requirements like throughput, PPS, SSL requirement, and average packet size.

2. Choose the right ADC offering and licensing that meets your requirements, such as VPX bandwidth offerings or vCPU based licensing.
3. Based on the chosen offering, decide on the AWS instance.

**Example:**

A 5 Gbps license enables 5 data packet engines. Hence, the vCPU requirement is 6 (5+1 for management). But 6 vCPU instance is not available. So an 8 vCPU is good enough to reach that throughput provided you choose a network that supports 5 Gbps bandwidth. For example, you must choose m5.2xlarge for a 5 Gbps bandwidth license to enable max PE allocation for 5 Gbps license. But if you use vCPU license that is not limited by throughput, you might get 5 Gbps throughput using the m5.xlarge instance itself.

| Instance Size | vCPU | Memory (GiB) | Instance Storage (GiB) | Network Bandwidth (Gbps) | EBS Bandwidth (Mbps) |
|---|---|---|---|---|---|
| m5.large | 2 | 8 | EBS-Only | Up to 10 | Up to 4,750 |
| m5.xlarge | 4 | 16 | EBS-Only | Up to 10 | Up to 4,750 |
| m5.2xlarge | 8 | 32 | EBS-Only | Up to 10 | Up to 4,750 |
| m5.4xlarge | 16 | 64 | EBS-Only | Up to 10 | 4,750 |

- **Is three NICs-three subnets deployment mandatory for ADC in AWS?**

  Three NICs-three subnets is the recommended deployment, where each one for management, client and server network. This deployment gives better traffic isolation and VPX performance. Two NICs-two subnets, and one NIC-one subnet are the other available options. It is not recommended to have multiple NICs sharing a subnet in AWS, such as a two NICs–one subnet deployment. This scenario can cause networking issues like asymmetric routing. For more information, see Best practices for configuring network interfaces in AWS.

- **Why does an ENA driver on AWS always indicate a 1Gbps (1/1) link speed, irrespective of the instance's network capabilities?**

  The reported speed of an AWS Elastic Network Adapter (ENA) is often displayed as 1Gbps (1/1) regardless of the selected instance type. This is because the indicated speed does not directly reflect the actual network performance. Unlike traditional network interfaces, ENA speeds can dynamically scale based on the instance's requirements and workload. The true network performance is primarily determined by the instance type and size. Therefore, the actual network throughput can vary significantly depending on the specific instance type and the current network load.

## Deploy a NetScaler VPX instance on Microsoft Azure

When you deploy a NetScaler VPX instance on Microsoft Azure Resource Manager (ARM), you can use both of the following feature sets to achieve your business needs:

- Azure cloud computing capabilities
- NetScaler load balancing and traffic management features

You can deploy NetScaler VPX instances on ARM either as standalone instances or as high availability pairs in active-standby modes.

You can deploy a NetScaler VPX instance on the Microsoft Azure in two ways:

- Through Azure Marketplace. The NetScaler VPX virtual appliance is available as an image in the Microsoft Azure Marketplace.
- Using the NetScaler Azure Resource Manager (ARM) json template available on GitHub. For more information, see the GitHub repository for NetScaler solution templates.

The Microsoft Azure stack is an integrated platform of hardware and software that delivers the Microsoft Azure public cloud services in a local data center to let organizations construct hybrid clouds. You can now deploy the NetScaler VPX instances on the Microsoft Azure stack.

> **Note:**
>
> Azure restricts access to traffic originating from outside Azure and blocks them. To provide access, enable the service or port by adding an inbound rule in the network security group attached to the NIC of the VM to which a public IP address is attached. For more information, see Azure documentation about Inbound NAT rules.

### Prerequisite

You need some prerequisite knowledge before deploying a NetScaler VPX instance on Azure.

- Familiarity with Azure terminology and network details. For information, see Azure terminology.
- Knowledge of a NetScaler appliance. For detailed information the NetScaler appliance, see NetScaler
- Knowledge of NetScaler networking. See the Networking topic.

### How a NetScaler VPX instance works on Azure

In an on-premises deployment, a NetScaler VPX instance requires at least three IP addresses:

- Management IP address, called NSIP address
- Subnet IP (SNIP) address for communicating with the server farm
- Virtual server IP (VIP) address for accepting client requests

For more information, see Network architecture for NetScaler VPX instances on Microsoft Azure.

> **Note:**
>
> NetScaler VPX instance supports both the Intel and AMD processors. VPX virtual appliances can be deployed on any instance type that has two or more virtualized cores and more than 2 GB memory. For more information on system requirements, see NetScaler VPX data sheet.

In an Azure deployment, you can provision a NetScaler VPX instance on Azure in three ways:

- Multi-NIC multi-IP architecture
- Single NIC multi-IP architecture
- Single NIC single IP

Depending on your needs, you can use any of these supported architecture types.

#### Multi-NIC multi-IP architecture

In this deployment type, you can have more than one network interfaces (NICs) attached to a VPX instance. Any NIC can have one or more IP configurations - static or dynamic public and private IP addresses assigned to it.

For more information, see the following use cases:

- Configure a high-availability setup with multiple IP addresses and NICs
- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands

> **Note:**
>
> To avoid MAC moves and interface mutes on Azure environments, we recommend you to create a VLAN per data interface (without tag) of NetScaler VPX instance and bind the primary IP of NIC in Azure. For more information, see CTX224626 article.

#### Single NIC multi-IP architecture

In this deployment type, one network interface (NIC) associated with multiple IP configurations - static or dynamic public and private IP addresses assigned to it.
For more information, see the following use cases:

- Configure multiple IP addresses for a NetScaler VPX standalone instance
- Configure multiple IP addresses for a NetScaler VPX standalone instance by using PowerShell commands

#### Single NIC single IP

In this deployment type, one network interface (NIC) associated with a single IP address, which is used to perform the functions of NSIP, SNIP, and VIP.

For more information, see Configure a NetScaler VPX standalone instance.

> **Note:**
>
> The single IP mode is available only in Azure deployments. This mode isn't available for a NetScaler VPX instance on your premises, on AWS, or in other types of deployment.

### NetScaler VPX licensing

A NetScaler VPX instance on Azure requires a valid license. The licensing options available for NetScaler VPX instances running on Azure are:

- **Bring your own license (BYOL)**: To use the BYOL option, follow these steps:
  - Use the licensing portal on the NetScaler website to generate a valid license.
  - Upload the generated license to the instance.
- **NetScaler VPX Check-in and Check-out license**: This licensing model allows you to check out a license from a pool of available licenses and check it back in when no longer needed. For more information and detailed instructions, see NetScaler VPX Check-in and Check-out License.

> **Note:**
>
> - Subscription-based licensing is no longer supported for NetScaler VPX instances on Azure.
> - Do a warm restart before making any configuration changes on the NetScaler VPX instance to enable the correct NetScaler VPX license.

### VPX performance and Recommended Azure instance types

For the desired VPX performance, the following Azure instance types are recommended.

| VPX performance | Azure instance types | | |
|---|---|---|---|
| | VPX 1 NIC/2 NIC | VPX 3 NIC | VPX up to 8 NIC |
| Up to 200 Mbps | Standard_D2s_v5 | Standard_D8s_v5 | Standard_D16_v5 |
| Up to 1 Gbps | Standard_D4s_v5 | Standard_D8s_v5 | Standard_D16_v5 |
| Up to 5 Gbps | Standard_D8ds_v5 | Standard_D8ds_v5 | Standard_D16_v5 |
| Up to 10 Gbps | Standard_D8s_v5 | Standard_D8s_v5 | Standard_D16_v5 |

### Points to note

- Azure supports VPX throughput up to 10 Gbps. For more information, see the NetScaler VPX Data Sheet.
- To achieve optimal performance on NetScaler VPX instances with throughput over 1 Gbps, you must enable Azure accelerated networking. It is recommended to use an Azure instance type that supports accelerated networking for this purpose. For more information on configuring Accelerated networking, see Configure a NetScaler VPX instance to use Azure accelerated networking.
- If you expect that you might have to shut down and temporarily deallocate the NetScaler VPX virtual machine at any time, assign a static Internal IP address while creating the virtual machine. If you do not assign a static internal IP address, Azure might assign the virtual machine a different IP address each time it restarts, and the virtual machine might become inaccessible.
- For Citrix Virtual Apps and Desktops deployment, a VPN virtual server on a VPX instance can be configured in the following modes:
  - Basic mode, where the `ICAOnly` VPN virtual server parameter is set to ON. The Basic mode works fully on an unlicensed NetScaler VPX instance.
  - SmartAccess mode, where the `ICAOnly` VPN virtual server parameter is set to OFF. The SmartAccess mode works for only five NetScaler AAA session users on an unlicensed NetScaler VPX instance.

  > **Note:**
  >
  > To configure the SmartControl feature, you must apply a Premium license to the NetScaler VPX instance.

### IPv6 support for NetScaler VPX instance in Azure

From release 13.1-21.x onwards, NetScaler VPX standalone instance supports IPv6 addresses in Azure. You can configure the IPv6 addresses as VIP and SNIP addresses on NetScaler VPX standalone instance in Azure cloud.

For information on how to enable IPv6 on Azure, see the following Azure documentation:

- What is IPv6 for Azure Virtual Network?
- Add IPv6 to an IPv4 application in Azure virtual network - Azure CLI
- Address types

For information on how the NetScaler appliance supports IPv6, see Internet Protocol version 6.

**IPv6 Limitations:**

- IPv6 deployments in NetScaler currently do not support Azure backend autoscaling.
- IPv6 is not supported for NetScaler VPX HA deployment.

### Limitations

Running the NetScaler VPX load-balancing solution on ARM imposes the following limitations:

- The Azure architecture does not accommodate support for the following NetScaler features:
  - Gratuitous ARP (GARP)
  - L2 Mode
  - Tagged VLAN
  - Dynamic Routing
  - virtual MAC
  - USIP
  - Clustering
- When using a NetScaler VPX instance with a throughput exceeding 3 Gbps, the actual network throughput may not align with the throughput specified in the instance's license. However, other features such as SSL throughput and SSL transactions per second might improve.
- The deployment ID that is generated by Azure during virtual machine provisioning isn't visible to the user in ARM. You can't use the deployment ID to deploy NetScaler VPX appliance on ARM.

## Azure terminology

Some of the Azure terms that are used in the NetScaler VPX Azure documentation are listed below.

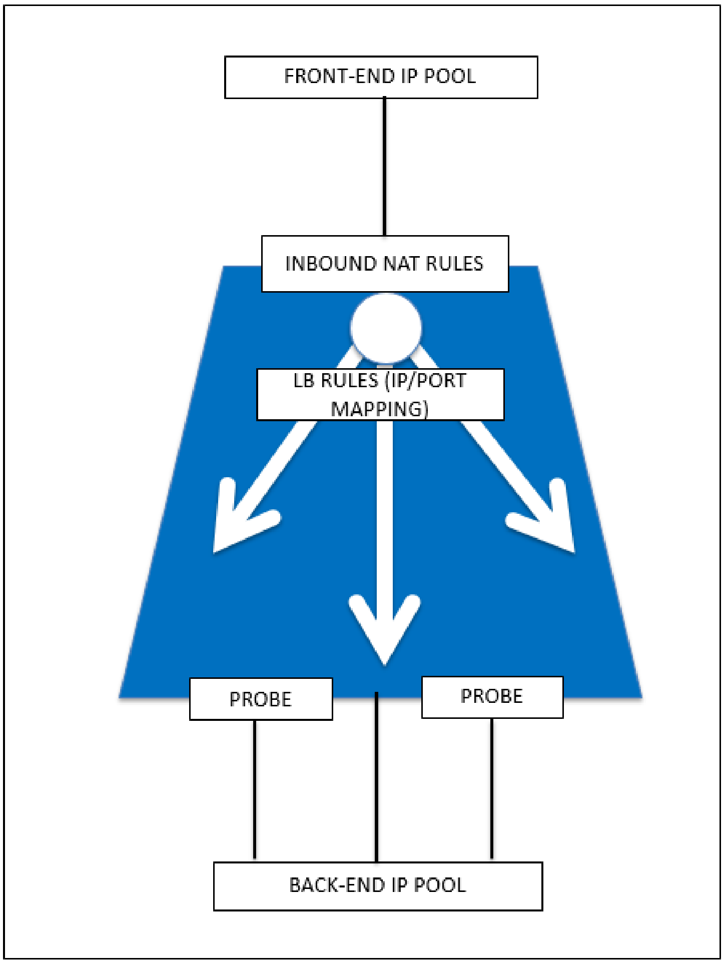
 

1. Azure Load Balancer –Azure load balancer is a resource that distributes incoming traffic among computers in a network. Traffic is distributed among virtual machines defined in a load-balancer set. A load balancer can be external or internet-facing, or it can be internal.
2. Azure Resource Manager (ARM) –ARM is the new management framework for services in Azure. Azure Load Balancer is managed using ARM-based APIs and tools.
3. Back-End Address Pool –These are IP addresses associated with the virtual machine NIC (NIC) to which load will be distributed.
4. BLOB - Binary Large Object –Any binary object like a file or an image that can be stored in Azure storage.
5. Front-End IP Configuration –An Azure Load balancer can include one or more front-end IP addresses, also known as a virtual IPs (VIPs). These IP addresses serve as ingress for the traffic.
6. Instance Level Public IP (ILPIP) –An ILPIP is a public IP address that you can assign directly to your virtual machine or role instance, rather than to the cloud service that your virtual machine or role instance resides in. This does not take the place of the VIP (virtual IP) that is assigned to your cloud service. Rather, it's an extra IP address that you can use to connect directly to your virtual machine or role instance.

   > **Note:**
   >
   > In the past, an ILPIP was referred to as a PIP, which stands for public IP.

7. Inbound NAT Rules –This contains rules mapping a public port on the load balancer to a port for a specific virtual machine in the back end address pool.
8. IP-Config - It can be defined as an IP address pair (public IP and private IP) associated with an individual NIC. In an IP-Config, the public IP address can be NULL. Each NIC can have multiple IP-Config associated with it, which can be up to 255.
9. Load Balancing Rules –A rule property that maps a given front-end IP and port combination to a set of back-end IP addresses and port combination. With a single definition of a load balancer resource, you can define multiple load balancing rules, each rule reflecting a combination of a front end IP and port and back end IP and port associated with virtual machines.



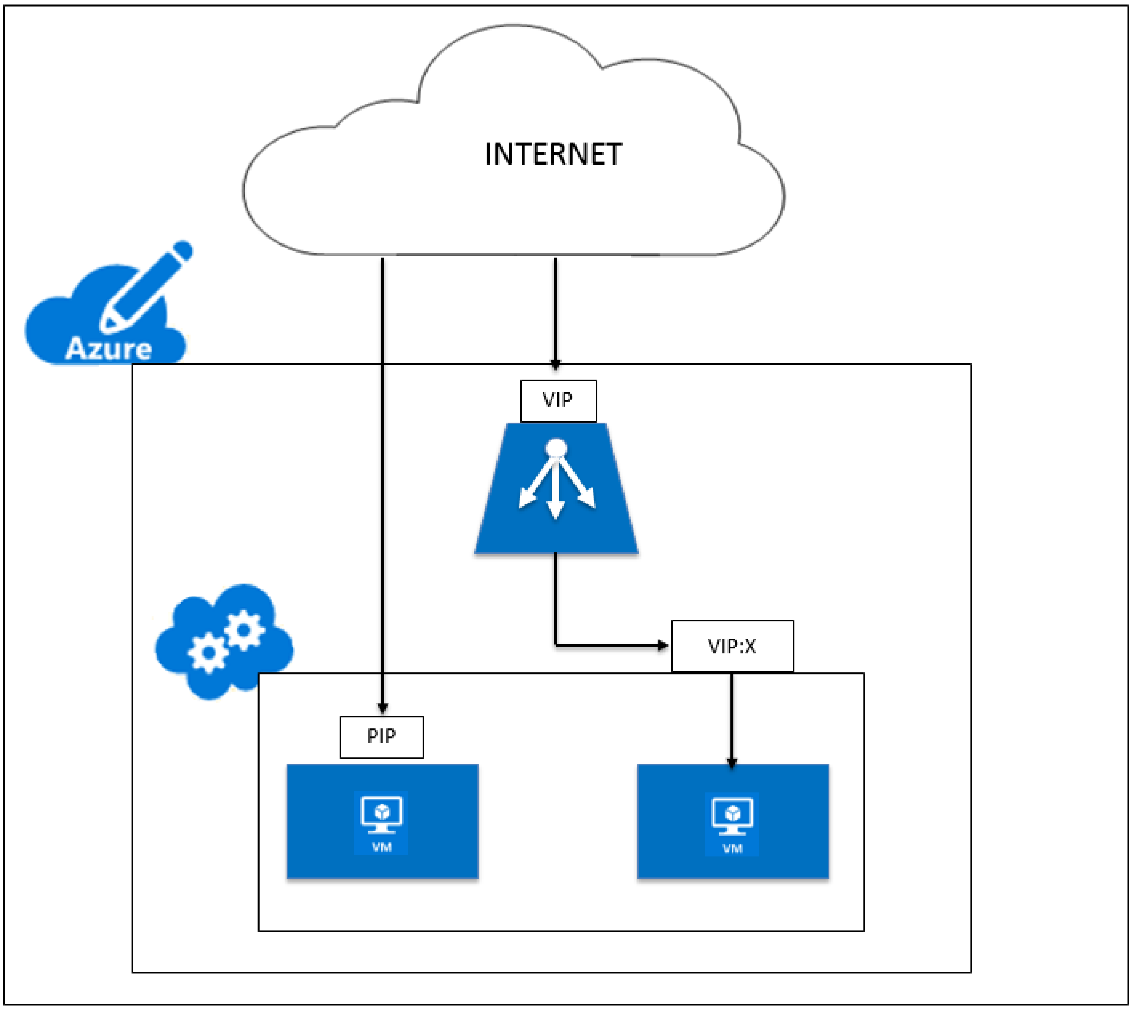

10. Network security group –Contains a list of Access Control List (ACL) rules that allow or deny network traffic to your virtual machine instances in a virtual network. NSGs can be associated with either subnets or individual virtual machine instances within that subnet. When a network security group is associated with a subnet, the ACL rules apply to all the virtual machine instances in that subnet. In addition, traffic to an individual virtual machine can be restricted further by associating a network security group directly to that virtual machine.
11. Private IP addresses –Used for communication within an Azure virtual network, and your on-premises network when you use a VPN gateway to extend your network to Azure. Private IP addresses allow Azure resources to communicate with other resources in a virtual network or an on-premises network through a VPN gateway or ExpressRoute circuit, without using an Internet-reachable IP address. In the Azure Resource Manager deployment model, a private IP address is associated with the following types of Azure resources –virtual machines, internal load balancers (ILBs), and application gateways.
12. Probes –This contains health probes used to check availability of virtual machines instances in the back end address pool. If a particular virtual machine does not respond to health probes for some time, then it is taken out of traffic serving. Probes enable you to keep track of the

health of virtual instances. If a health probe fails, the virtual instance will be taken out of rotation automatically.

13. Public IP Addresses (PIP) –PIP is used for communication with the Internet, including Azure public-facing services and is associated with virtual machines, Internet-facing load balancers, VPN gateways, and application gateways.
14. Region - An area within a geography that does not cross national borders and that contains one or more data centers. Pricing, regional services, and offer types are exposed at the region level. A region is typically paired with another region, which can be up to several hundred miles away, to form a regional pair. Regional pairs can be used as a mechanism for disaster recovery and high availability scenarios. Also referred to generally as location.
15. Resource Group - A container in Resource Manager holds related resources for an application. The resource group can include all of the resources for an application, or only those resources that are logically grouped together
16. Storage Account –An Azure storage account gives you access to the Azure blob, queue, table, and file services in Azure Storage. Your storage account provides the unique namespace for your Azure storage data objects.
17. Virtual Machine –The software implementation of a physical computer that runs an operating system. Multiple virtual machines can run simultaneously on the same hardware. In Azure, virtual machines are available in a variety of sizes.
18. Virtual Network - An Azure virtual network is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can fully control the IP address blocks, DNS settings, security policies, and route tables within this network. You can also further segment your VNet into subnets and launch Azure IaaS virtual machines and cloud services (PaaS role instances). Additionally, you can connect the virtual network to your on-premises network using one of the connectivity options available in Azure. In essence, you can expand your network to Azure, with complete control on IP address blocks with the benefit of enterprise scale Azure provides.




## Network architecture for NetScaler VPX instances on Microsoft Azure

In Azure Resource Manager (ARM), a NetScaler VPX virtual machine (VM) resides in a virtual network. A single network interface can be created in a given subnet of the Virtual Network and can be attached to the VPX instance. You can filter network traffic to and from a VPX instance in an Azure virtual network with a network security group. A network security group contains security rules that allow or deny inbound network traffic to or outbound network traffic from a VPX instance. For more information, see Security groups.

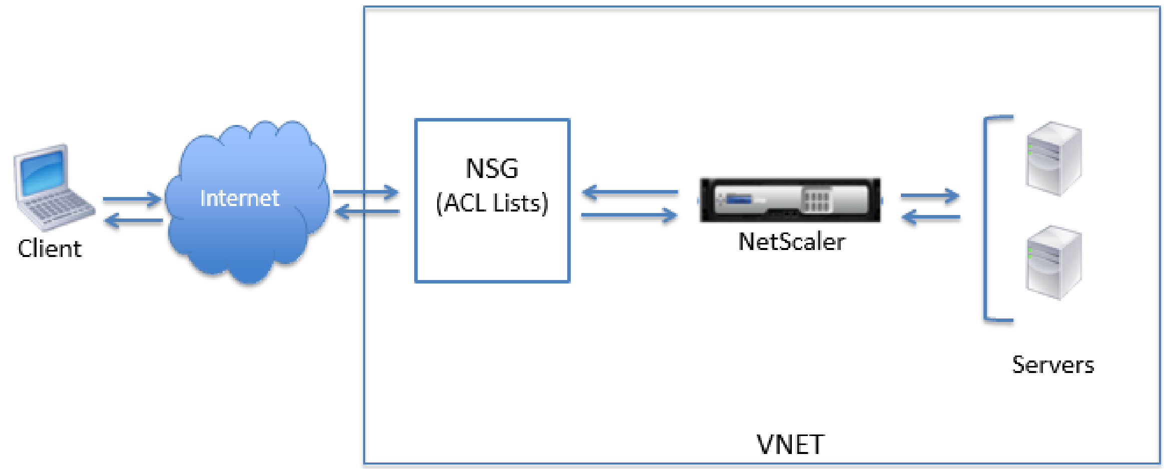
Network security group filters the requests to the NetScaler VPX instance, and the VPX instance sends them to the servers. The response from a server follows the same path in reverse. The Network security group can be configured to filter a single VPX VM, or, with subnets and virtual networks, can filter traffic in deployment of multiple VPX instances.

The NIC contains network configuration details such as the virtual network, subnets, internal IP address, and Public IP address.

While on ARM, it is good to know the following IP addresses that are used to access the VMs deployed with a single NIC and a single IP address:

- Public IP (PIP) address is the internet-facing IP address configured directly on the virtual NIC of the NetScaler VM. This allows you to directly access a VM from the external network.
- NetScaler IP (also known as NSIP) address is the internal IP address configured on the VM. It is non-routable.
- Virtual IP address (VIP) is configured by using the NSIP and a port number. Clients access NetScaler services through the PIP address, and when the request reaches the NIC of the NetScaler VPX VM or the Azure load balancer, the VIP gets translated to internal IP (NSIP) and internal port number.
- Internal IP address is the private internal IP address of the VM from the virtual network's address space pool. This IP address cannot be reached from the external network. This IP address is by default dynamic unless you set it to static. Traffic from the internet is routed to this address according to the rules created on the network security group. The network security group integrates with the NIC to selectively send the right type of traffic to the right port on the NIC, which depends on the services configured on the VM.

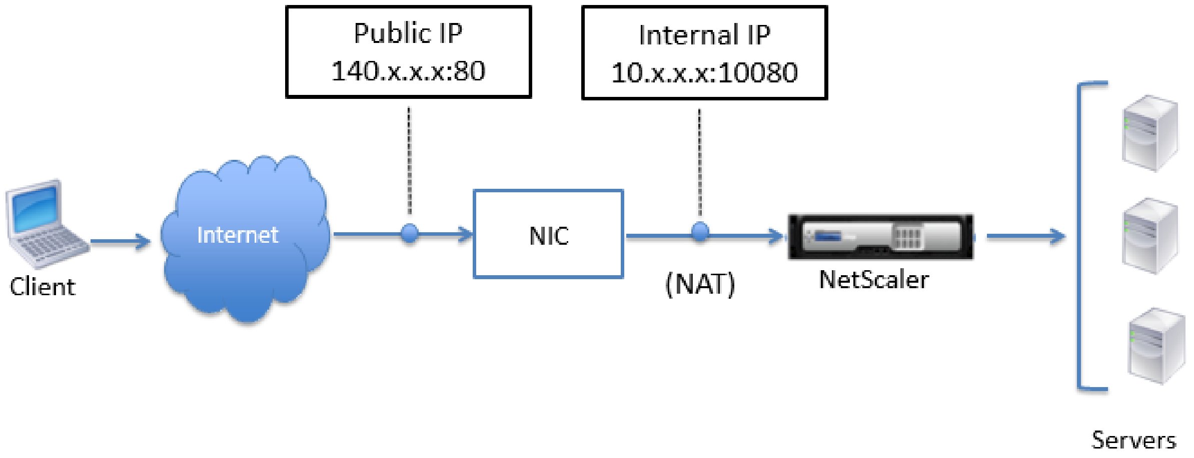
The following figure shows how traffic flows from a client to a server through a NetScaler VPX instance provisioned in ARM.




### Traffic flow through network address translation

You can also request a public IP (PIP) address for your NetScaler VPX instance (instance level). If you use this direct PIP at the VM level, you need not define inbound and outbound rules to intercept the network traffic. The incoming request from the Internet is received on the VM directly. Azure performs network address translation (NAT) and forwards the traffic to the internal IP address of the VPX instance.

The following figure shows how Azure performs network address translation to map the NetScaler internal IP address.




In this example, the Public IP assigned to the network security group is 140.x.x.x and the internal IP address is 10.x.x.x. When the inbound and outbound rules are defined, public HTTP port 80 is defined as the port on which the client requests are received, and a corresponding private port, 10080, is defined as the port on which the NetScaler VPX instance listens. The client request is received on the Public IP address (140.x.x.x). Azure performs network address translation to map the PIP to the internal IP address 10.x.x.x on port 10080, and forwards the client request.

> **Note:**
>
> NetScaler VPX VMs in high availability are controlled by external or internal load balancers that have inbound rules defined on them to control the load balancing traffic. The external traffic is first intercepted by these load balancers and the traffic is diverted according to the load balancing rules configured, which has back-end pools, NAT rules, and health probes defined on the load balancers.

### Port usage guidelines

You can configure more inbound and outbound rules n network security group while creating the NetScaler VPX instance or after the virtual machine is provisioned. Each inbound and outbound rule is associated with a public port and a private port.

Before configuring network security group rules, note the following guidelines regarding the port numbers you can use:

1. The NetScaler VPX instance reserves the following ports. You cannot define these as private ports when using the Public IP address for requests from the internet.

   Ports 21, 22, 80, 443, 8080, 67, 161, 179, 500, 520, 3003, 3008, 3009, 3010, 3011, 4001, 5061, 9000, 7000.

   However, if you want internet-facing services such as the VIP to use a standard port (for example, port 443) you have to create port mapping by using the network security group. The standard port is then mapped to a different port that is configured on the NetScaler for this VIP service.

   For example, a VIP service might be running on port 8443 on the VPX instance but be mapped to public port 443. So, when the user accesses port 443 through the Public IP, the request is directed to private port 8443.

2. Public IP address does not support protocols in which port mapping is opened dynamically, such as passive FTP or ALG.
3. High availability does not work for traffic that uses a public IP address (PIP) associated with a VPX instance, instead of a PIP configured on the Azure load balancer.

> **Note:**
>
> In Azure Resource Manager, a NetScaler VPX instance is associated with two IP addresses - a public IP address (PIP) and an internal IP address. While the external traffic connects to the PIP, the internal IP address or the NSIP is non-routable. To configure VIP in VPX, use the internal IP address and any of the free ports available. Do not use the PIP to configure VIP.

## Configure a NetScaler VPX standalone instance

You can provision a single NetScaler VPX instance in Azure Resource Manager (ARM) portal in a standalone mode by creating the virtual machine and configuring other resources.

### Before you begin

Ensure that you have the following:

- A Microsoft Azure user account
- Access to Microsoft Azure Resource Manager
- Microsoft Azure SDK
- Microsoft Azure PowerShell

On the Microsoft Azure Portal page, log on to the Azure Resource Manager portal by providing your user name and password.

> **Note:**
>
> In ARM portal, clicking an option in one pane opens a new pane to the right. Navigate from one pane to another to configure your device.

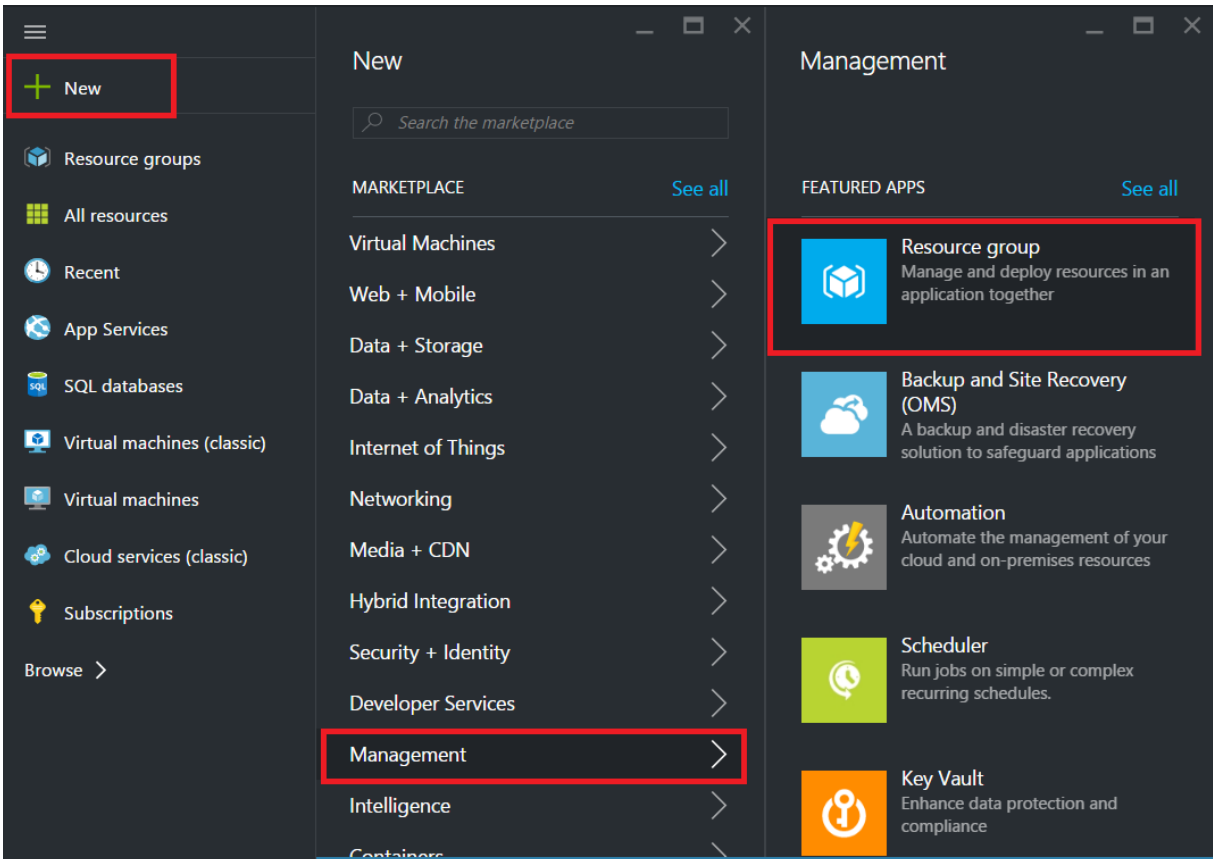
### Summary of configuration steps

1. Configure a resource group
2. Configure a network security group
3. Configure virtual network and its subnets
4. Configure a storage account
5. Configure an availability set
6. Configure a NetScaler VPX instance.

### Configure a resource group

Create a new resource group that is a container for all your resources. Use the resource group to deploy, manage, and monitor your resources as a group.

1. Click **New** > **Management** > **Resource group**.
2. In the **Resource group** pane, enter the following details:
    - Resource group name
    - Resource group location
3. Click **Create**.




#### Configure a network security group

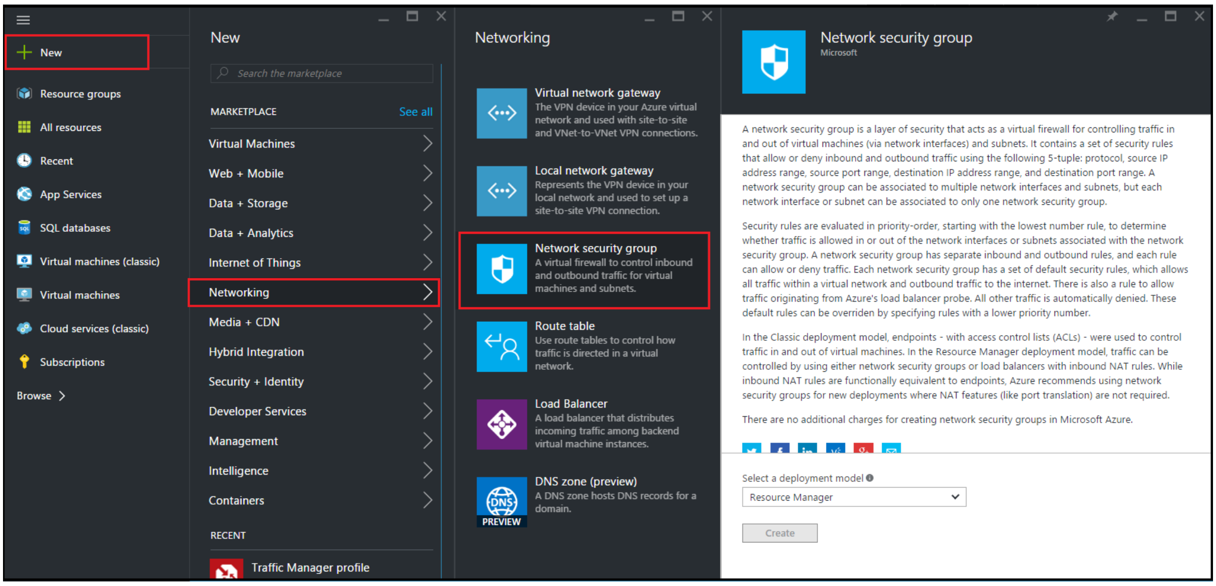
Create a network security group to assign inbound and outbound rules to control the incoming and outgoing traffic within the virtual network. Network security group allows you to define security rules for a single virtual machine and also to define security rules for a virtual network subnet.

1. Click **New** > **Networking** > **Network security group**.
2. In the **Create network security group** pane, enter the following details, and then click **Create**.

    - Name - type a name for the security group
    - Resource group - select the resource group from the drop-down list

> **Note:**
>
> Ensure that you have selected the correct location. The list of resources that appear in the drop-down list is different for different locations.

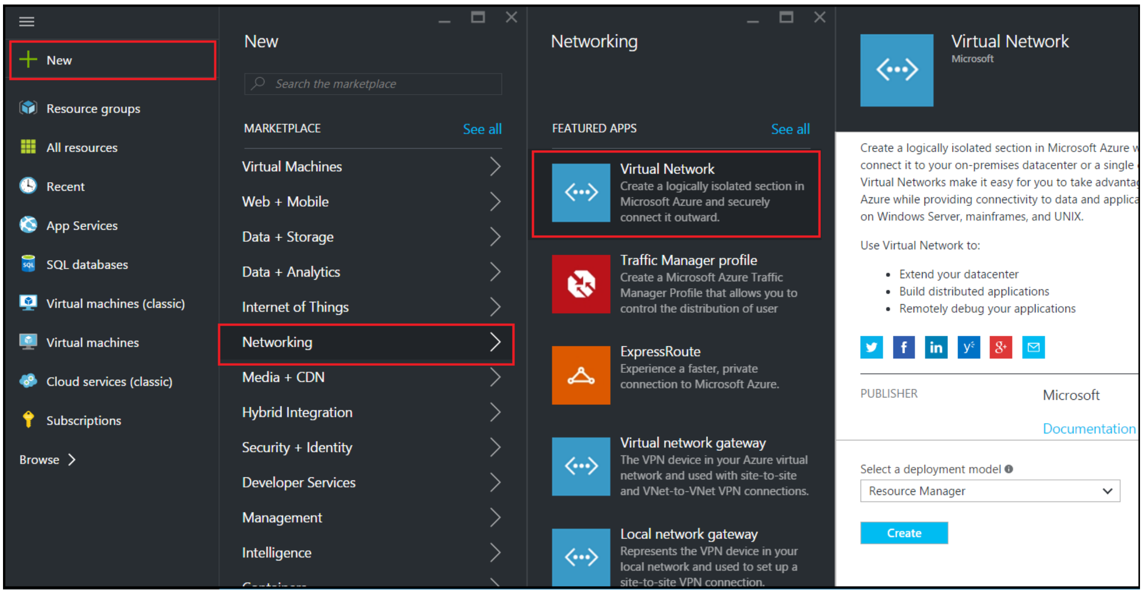



## Configure a virtual network and subnets

Virtual networks in ARM provide a layer of security and isolation to your services. VMs and services that are part of the same virtual network can access each other.

For these steps to create a virtual network and subnets.

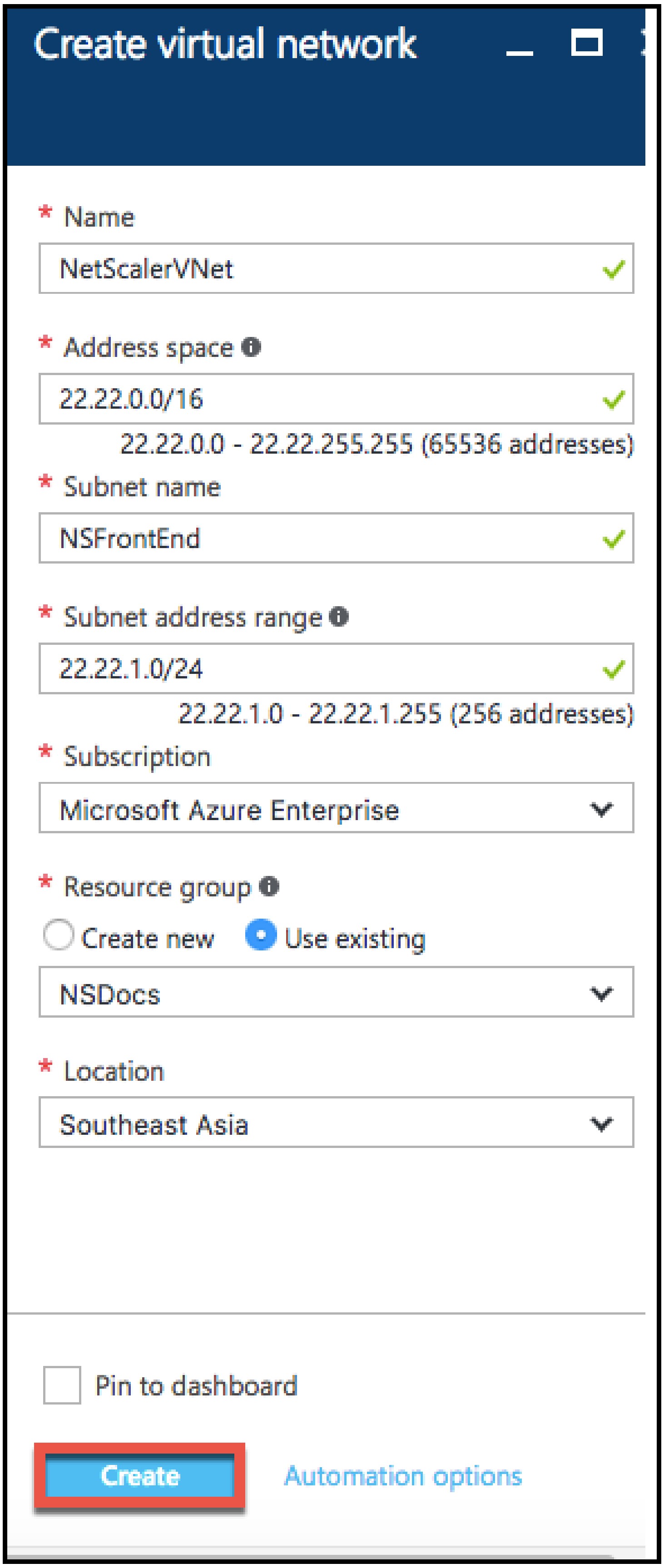
1. Click **New** > **Networking** > **Virtual Network**.
2. In the **Virtual Network** pane, ensure the deployment mode is **Resource Manager** and click **Create**.




3. In the **Create virtual network** pane, enter the following values, and then click **Create**.

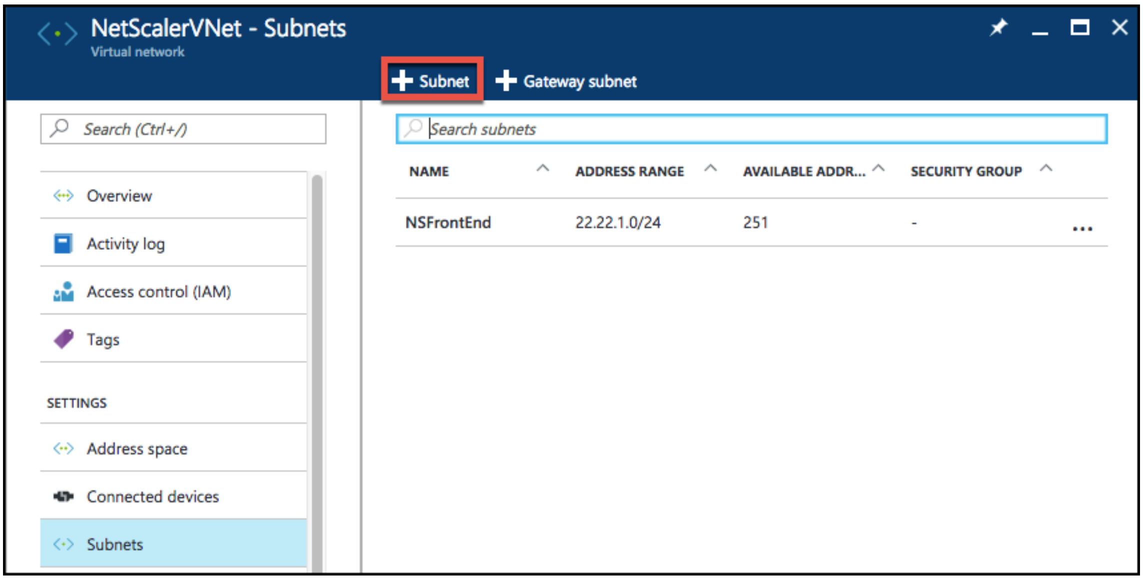
- Name of the virtual network
- Address space - type the reserved IP address block for the virtual network
- Subnet - type the name of the first subnet (you create the second subnet later in this step)
- Subnet address range - type the reserved IP address block of the subnet
- Resource group - select the resource group created earlier from the drop-down list


Create virtual network
* Name
NetScalerVNet
* Address space
22.22.0.0/16
22.22.0.0 - 22.22.255.255 (65536 addresses)
* Subnet name
NSFrontEnd
* Subnet address range
22.22.1.0/24
22.22.1.0 - 22.22.1.255 (256 addresses)
* Subscription
Microsoft Azure Enterprise
* Resource group
Create new
Use existing
NSDocs
* Location
Southeast Asia
Pin to dashboard
Create
Automation options

#### Configure the second subnet

1. Select the newly created virtual network from **All resources** pane and in the **Settings** pane, click **Subnets**.

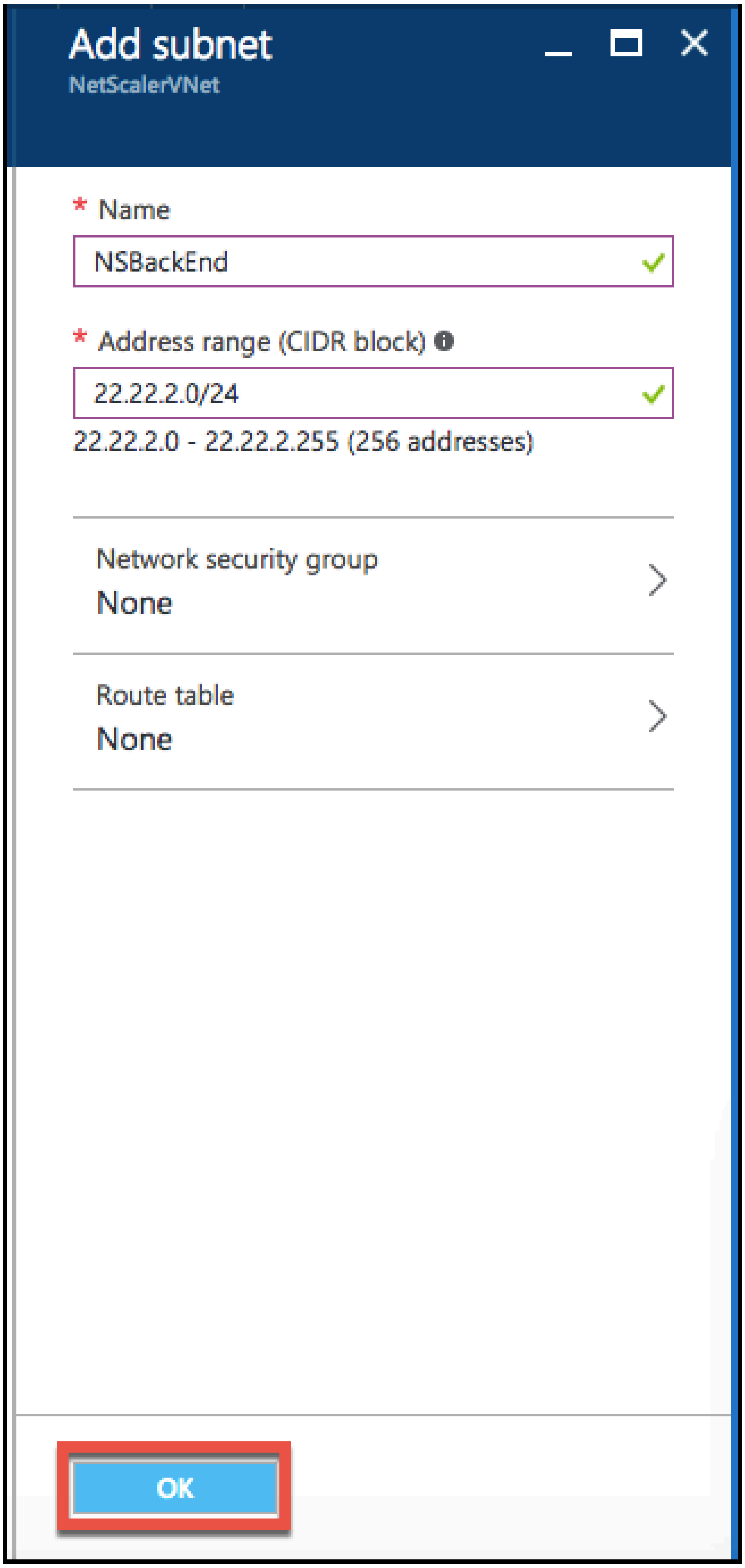



2. Click **+Subnet** and create the second subnet by entering the following details.

   - Name of the second subnet
   - Address range - type the reserved IP address block of the second subnet
   - Network security group - select the network security group from the drop-down list

3. Click **Create**.


Add subnet
NetScalerVNet
* Name
NSBackEnd
* Address range (CIDR block)
22.22.2.0/24
22.22.2.0 - 22.22.2.255 (256 addresses)
Network security group
None
Route table
None
OK

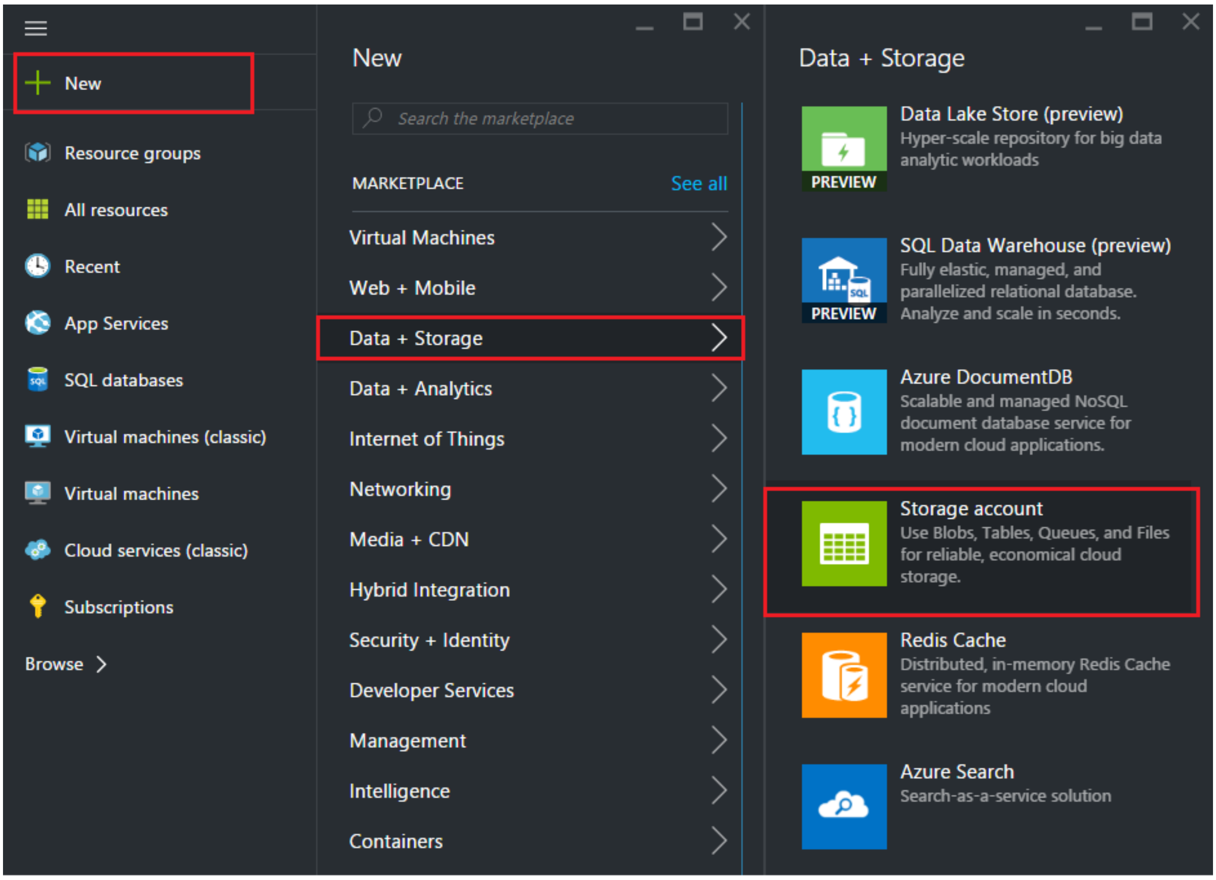
 

### Configure a storage account

The ARM IaaS infrastructure storage includes all services where we can store data in the form of blobs, tables, queues, and files. You can also create applications using these forms of storage data in ARM.

Create a storage account to store all your data.

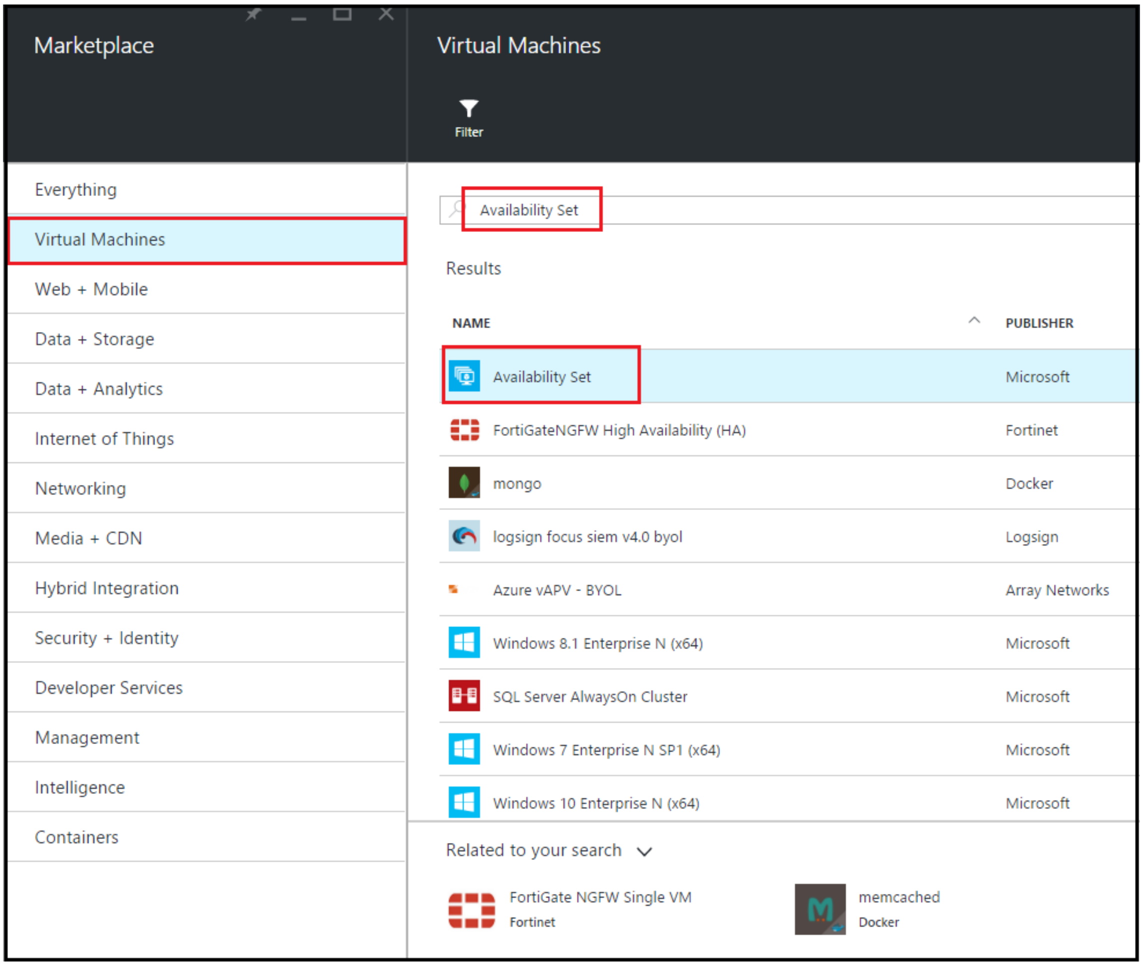
1. Click **+New** > **Data + Storage** > **Storage account**.
2. In the **Create storage account** pane, enter the following details:
   - Name of the account
   - Deployment mode - make sure to select **Resource Manager**
   - Account kind - select **General purpose** from the drop-down list
   - Replication - select **Locally redundant storage** from the drop-down list
   - Resource group - select the newly created resource group from the drop-down list
3. Click **Create**.




### Configure an availability set

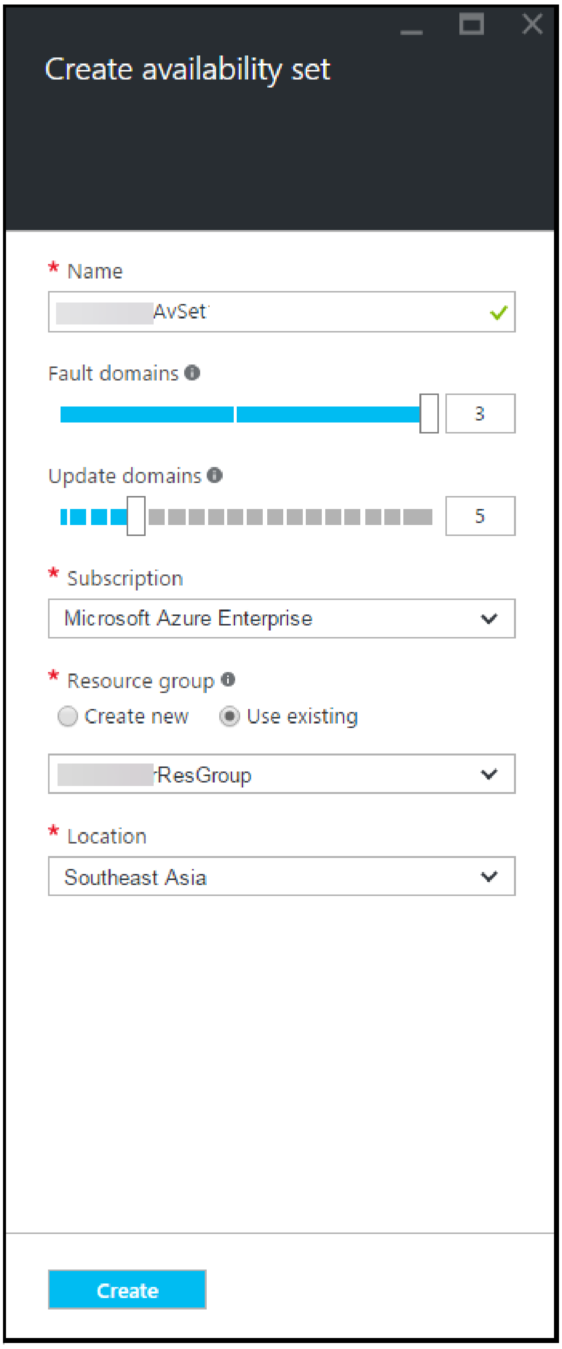
An availability set guarantee that at least one VM is kept up and running in case of planned or unplanned maintenance. Two or more VMs under the same 'availability set' are placed on different fault domains to achieve redundant services.

1. Click **+New**.
2. Click **See all** in the MARKETPLACE pane and click **Virtual Machines**.
3. Search for availability set, and then select **Availability set** entity from the list displayed.




4. Click **Create,** and in the **Create availability set** pane, enter the following details:
   - Name of the set
   - Resource group - select the newly created resource group from the drop-down list
5. Click **Create**.




### Configure a NetScaler VPX instance

Create an instance of NetScaler VPX in the virtual network. Obtain the NetScaler VPX image from the Azure Marketplace, and then use the Azure Resource Manager portal to create a NetScaler VPX instance.

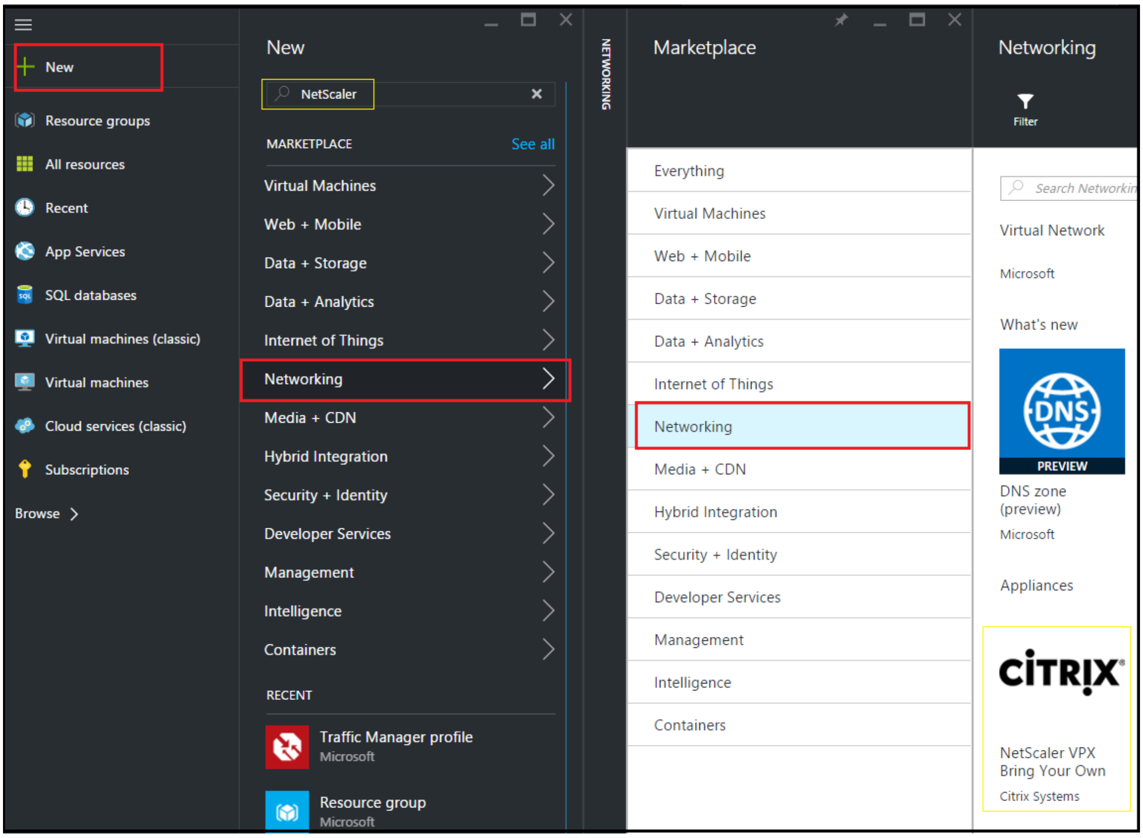
Before you begin creating the NetScaler VPX instance, make sure that you have created a virtual network with required subnets in which the instance resides. You can create virtual networks during VM provisioning, but without the flexibility to create different subnets.

Optionally, configure DNS server and VPN connectivity that allows a virtual machine to access internet resources.

> **Note:**
>
> Citrix recommends that you create resource group, network security group, virtual network, and other entities before you provision the NetScaler VPX VM, so that the network information is available during provisioning.

1. Click **+New** > **Networking**.
2. Click **See All** and in the Networking pane, click **NetScaler 13.0**.
3. Select **NetScaler 13.0 VPX Bring Your Own License** from the list of software plans.

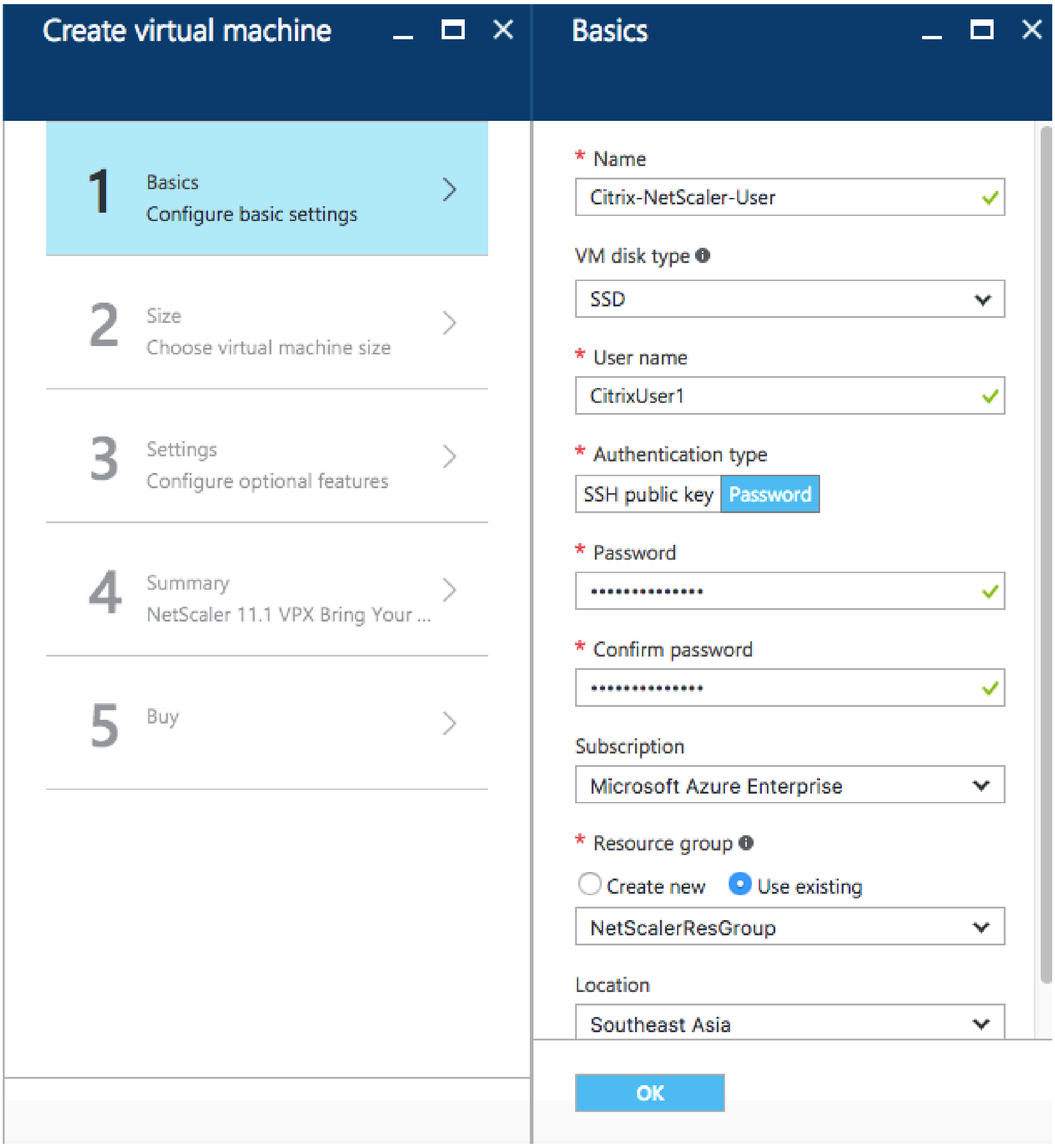
   As a quick way to find any entity on ARM portal, you can also type the name of the entity in the Azure Marketplace search box and press <Enter>. Type NetScaler in the search box to find the NetScaler images.

   > **Note:**
   >
   > Ensure to select the latest image. Your NetScaler image might have the release number in the name.

4. On the **NetScaler VPX Bring Your Own License** page, from the drop-down list, select **Resource Manager** and click **Create**.




5. In the **Create virtual machine** pane, specify the required values in each section to create a virtual machine. Click **OK** in each section to save your configuration.

**Basic**:

- Name - specify a name for the NetScaler VPX instance
- VM disk type - select SSD (default value) or HDD from the drop-down menu
- User name and Password - specify a user name and password to access the resources in the resource group that you have created

- Authentication Type - select SSH Public Key or Password
- Resource group - select the resource group you have created from the drop-down list

You can create a resource group here, but Citrix recommends that you create a resource group from Resource groups in Azure Resource Manager and then select the group from the drop-down list.

> **Note:**
>
> In an Azure stack environment, in addition to the basic parameters, specify the following parameters:
>
> - Azure stack domain
> - Azure stack tenant (Optional)
> - Azure client (Optional)
> - Azure client secret (Optional)

**Size**:

Depending on the VM disk type, SDD, or HDD, you selected in Basic settings, the disk sizes are displayed.

- Select a disk size according to your requirement and click **Select**.

**Settings**:

- Select the default (Standard) disk type
- Storage account - select the storage account
- Virtual network - select the virtual network
- Subnet - set the subnet address
- Public IP address - select the type of IP address assignment
- Network security group - select the security group that you have created. Ensure that inbound and outbound rules are configured in the security group.
- Availability Set - select the availability set from the drop-down menu box

**Summary**:

The configuration settings are validated and the Summary page displays the result of the validation. If the validation fails, the Summary page displays the reason of the failure. Go back to the particular section and make changes as required. If the validation passes, click **OK**.

**Buy**:

Review the offer details and legal terms on the Purchase page and click **Purchase**.

For high availability deployment, create two independent instances of NetScaler VPX in the same availability set and in the same resource group to deploy them in active-standby configuration.

## Configure multiple IP addresses for a NetScaler VPX standalone instance

This section explains how to configure a standalone NetScaler VPX instance with multiple IP addresses, in Azure Resource Manager (ARM). The VPX instance can have one or more NIC attached to it, and each NIC can have one or more static or dynamic public and private IP addresses assigned to it. You can assign multiple IP addresses as NSIP, VIP, SNIP, and so on.

For more information, see the Azure documentation Assign multiple IP addresses to virtual machines using the Azure portal.

If you want to use PowerShell commands, see Configuring multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands.

### Use case

In this use case, a standalone NetScaler VPX appliance is configured with a single NIC that is connected to a virtual network (VNET). The NIC is associated with three IP configurations (ipconfig), each server a different purpose - as shown in the table.

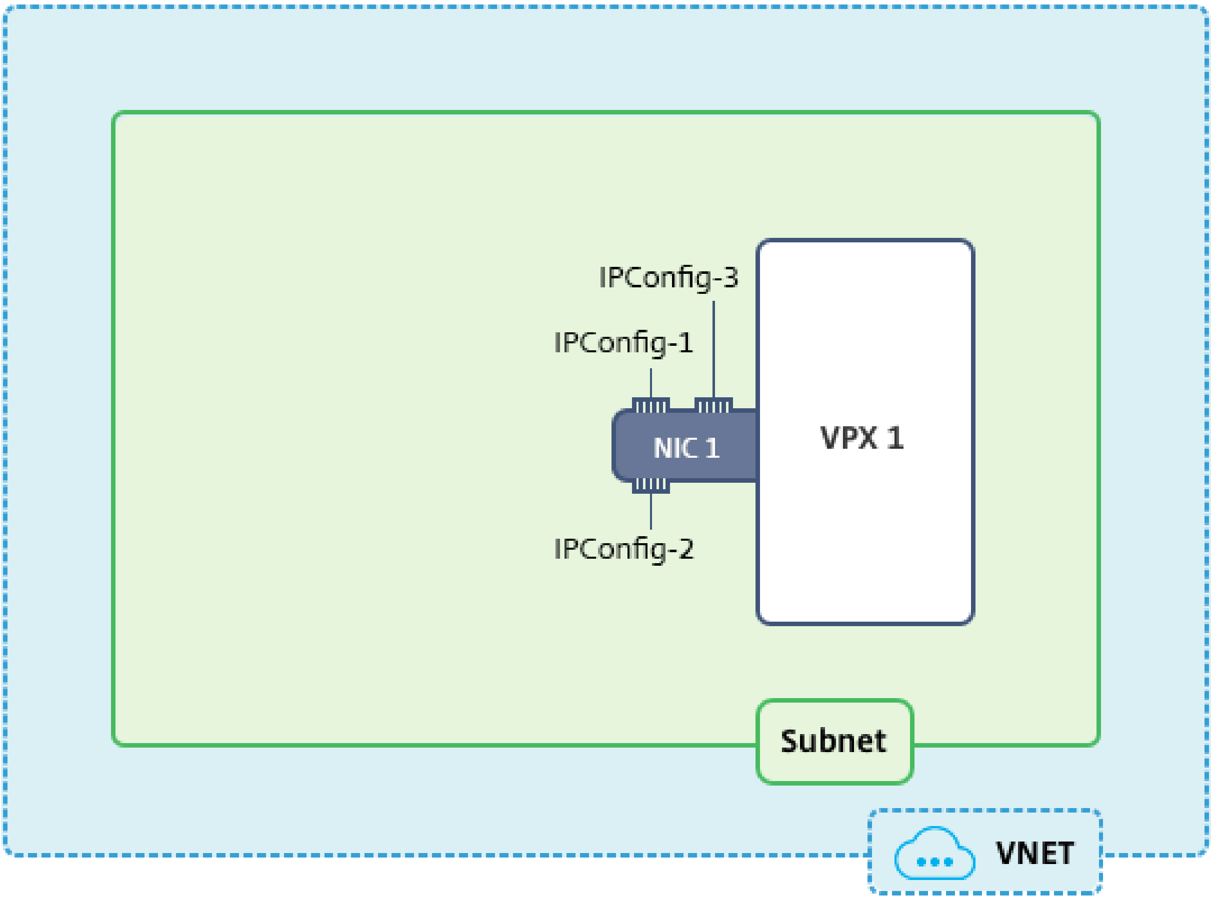
| IP config | Associated with | Purpose |
|---|---|---|
| ipconfig1 | Static public IP address; static private IP address | Serves management traffic |
| ipconfig2 | Static public IP address; static private address | Serves client-side traffic |
| ipconfig3 | Static private IP address | Communicates with back-end servers |

> **Note:**
>
> `IPConfig-3` is not associated with any public IP address.

**Diagram**: Topology

Here is the visual representation of the use case.




> **Note:**
>
> In a multi-NIC, multi-IP Azure NetScaler VPX deployment, the private IP associated with the primary (first) `IPConfig` of the primary (first) NIC is automatically added as the management NSIP of the appliance. The remaining private IP addresses associated with `IPConfigs` need to be added in the VPX instance as a VIP or SNIP by using the `add ns ip` command, according to your requirement.

### Before you begin

Before you begin, create a VPX instance by following the steps given at this link:

Configure a NetScaler VPX standalone instance

For this use case, the NSDoc0330VM VPX instance is created.

**Procedure to configure multiple IP addresses for a NetScaler VPX instance in standalone mode.**

For configuring multiple IP addresses for a NetScaler VPX appliance in standalone mode:

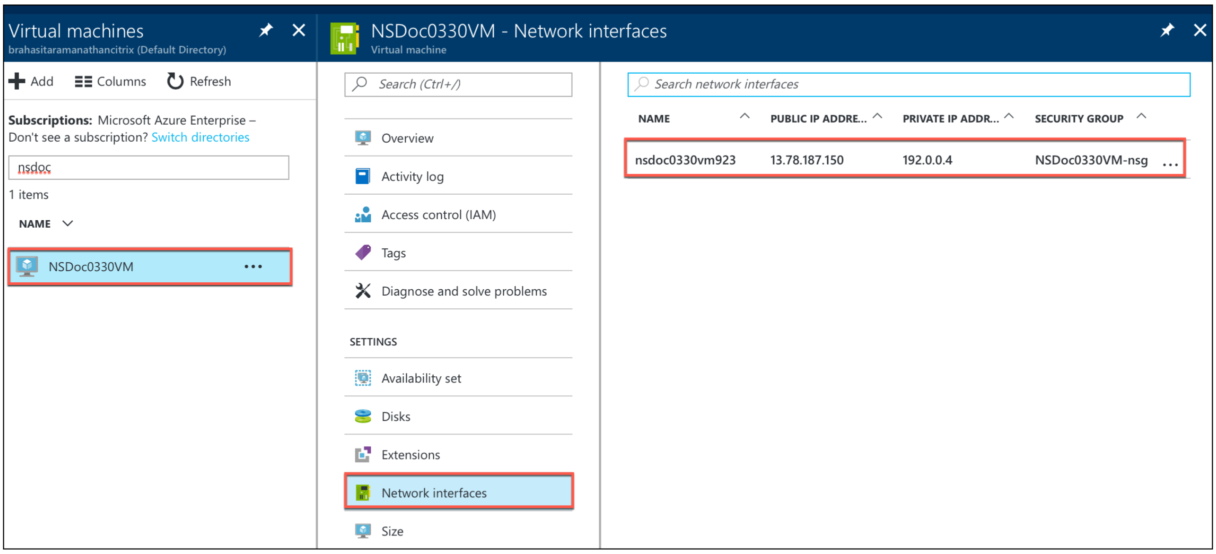
1. Add IP addresses to the VM

2. Configure NetScaler -owned IP addresses

**Step 1**: Add IP addresses to the VM

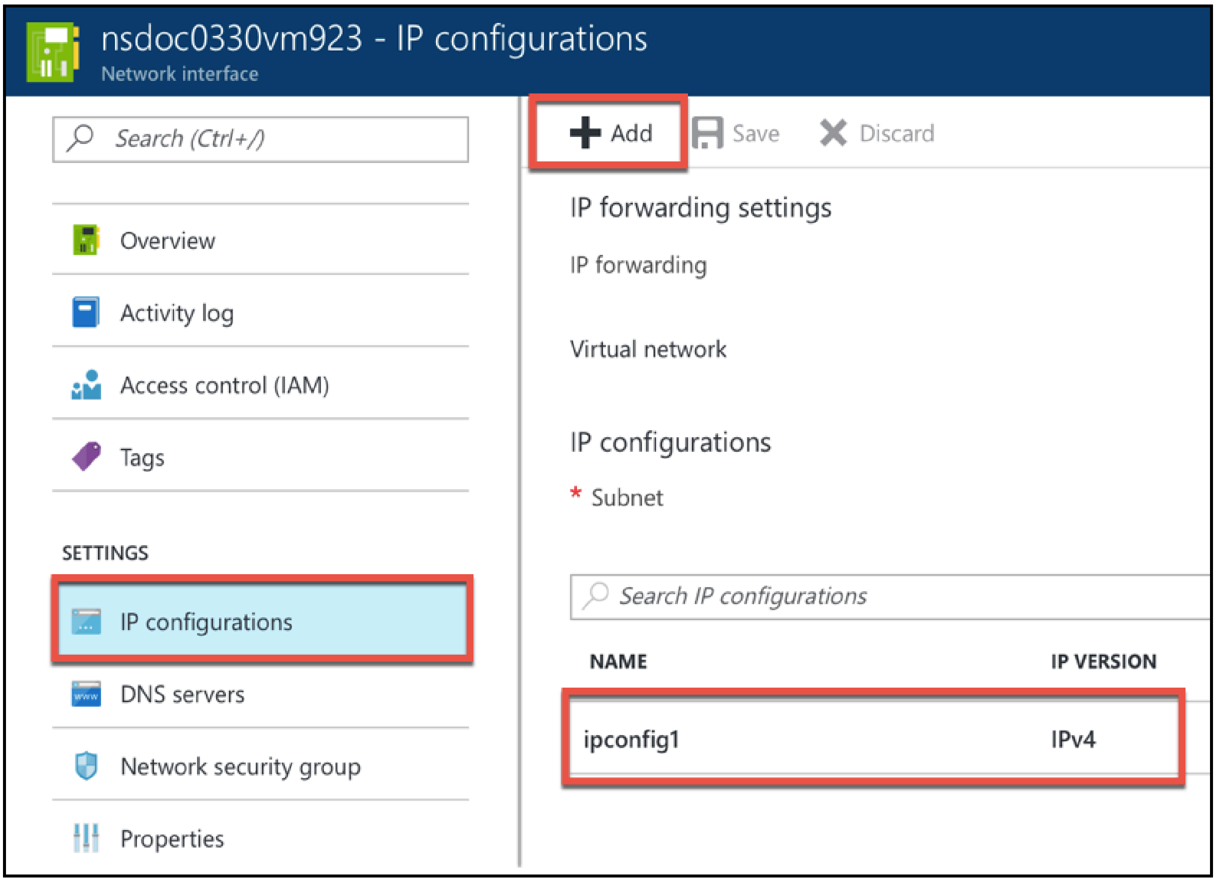
1. In the portal, click **More services > type virtual machines** in the filter box, and then click **Virtual machines**.
2. In the **Virtual machines** blade, click the VM you want to add IP addresses to. Click **Network interfaces** in the virtual machine blade that appears, and then select the network interface.




In the blade that appears for the NIC you selected, click **IP configurations**. The existing IP configuration that was assigned when you created the VM, **ipconfig1**, is displayed. For this use case, make sure the IP addresses associated with ipconfig1 are static. Next, create two more IP configurations: ipconfig2 (VIP) and ipconfig3 (SNIP).

To create more `ipconfigs`, create **Add**.

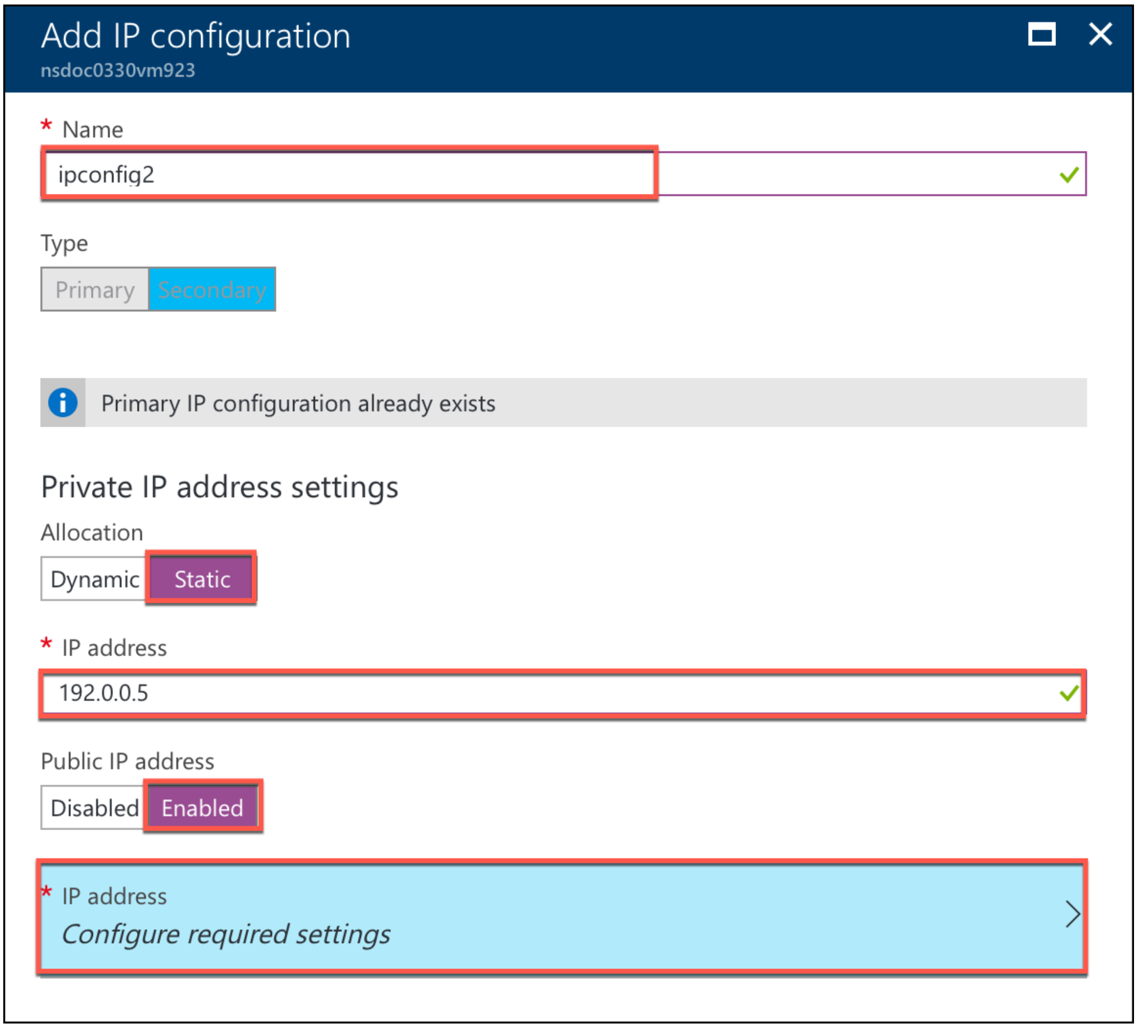
 




In the **Add IP configuration** window, enter a **Name**, specify allocation method as **Static**, enter an IP address (192.0.0.5 for this use case), and enable **Public IP address**.

> **Note:**
>
> Before adding a static private IP address, check for IP address availability and make sure the IP address belongs to the same subnet to which the NIC is attached.

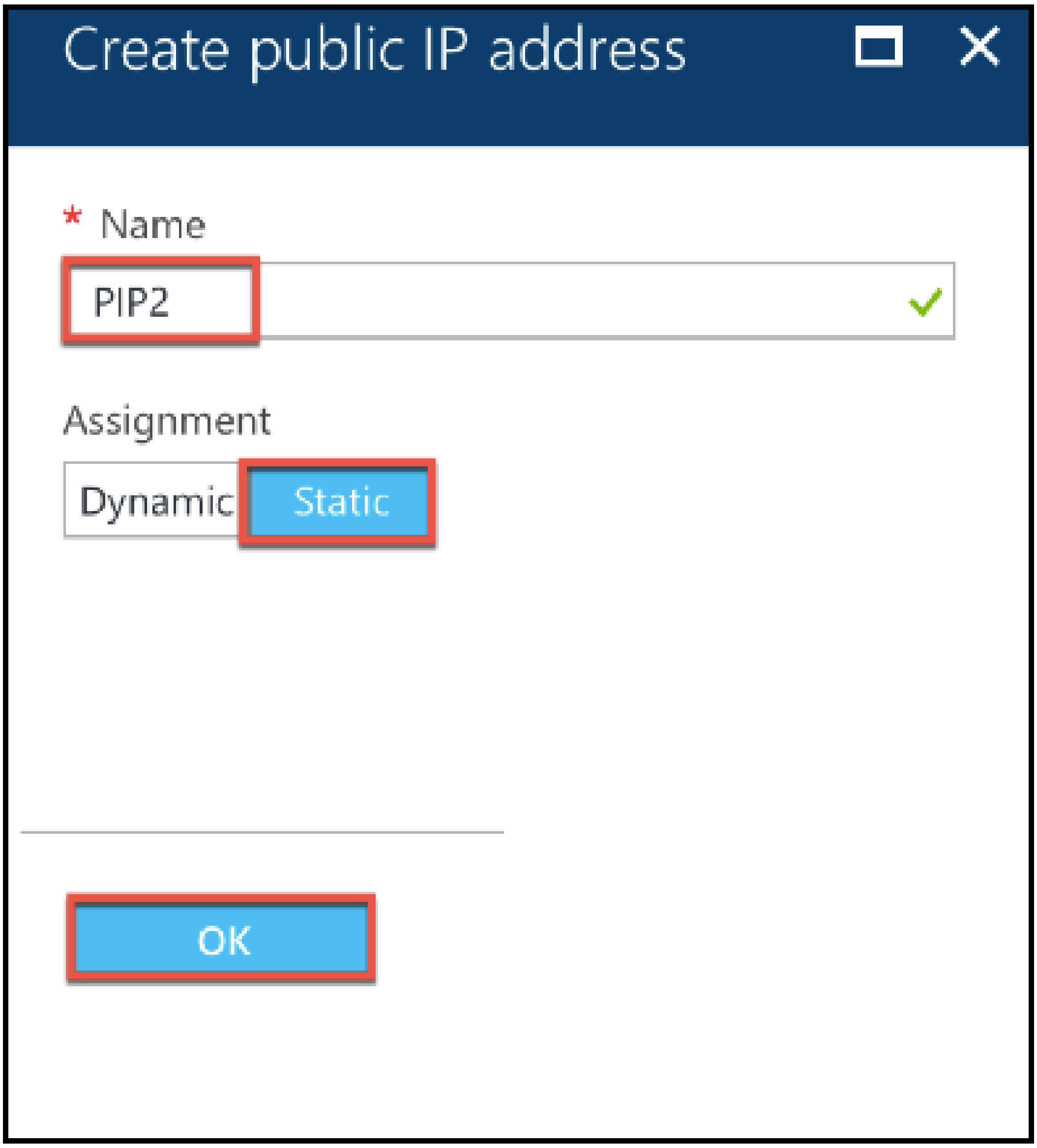



Next, click **Configure required settings** to create a static public IP address for ipconfig2.

By default, public IPs are dynamic. To make sure that the VM always uses the same public IP address, create a static Public IP.

In the Create public IP address blade, add a Name, under Assignment click **Static**. And then click **OK**.




> **Note:**
>
> Even when you set the allocation method to static, you cannot specify the actual IP address assigned to the public IP resource. Instead, it gets allocated from a pool of available IP addresses in the Azure location the resource is created in.

Follow the steps to add one more IP configuration for ipconfig3. Public IP is not mandatory.

Search IP configurations

| NAME | IP VERSION | TYPE | PRIVATE IP ADDRESS | PUBLIC IP ADDRESS |
| --- | --- | --- | --- | --- |
| ipconfig1 | IPv4 | Primary | 192.0.0.4 (Static) | 13.78.187.150 (NSDoc0330VM-ip) |
| ipconfig2 | IPv4 | Secondary | 192.0.0.5 (Static) | 13.78.183.123 (ipconfig2_PIP2) |
| ipconfig3 | IPv4 | Secondary | 192.0.0.6 (Static) | - |

#### Step 2: Configure NetScaler-owned IP addresses

Configure the NetScaler-owned IP addresses by using the GUI or the command `add ns ip`. For more information, see Configuring NetScaler-Owned IP Addresses.

## Configure a high-availability setup with multiple IP addresses and NICs

In a Microsoft Azure deployment, a high-availability configuration of two NetScaler VPX instances is achieved by using the Azure Load Balancer (ALB). This is achieved by configuring a health probe on ALB, which monitors each VPX instance by sending a health probe at every 5 seconds to both primary and secondary instances.

In this setup, only the primary node responds to health probes and the secondary does not. Once the primary sends the response to the health probe, the ALB starts sending the data traffic to the instance. If the primary instance misses two consecutive health probes, ALB does not redirect traffic to that instance. On failover, the new primary starts responding to health probes and the ALB redirects traffic to it. The standard VPX high availability failover time is three seconds. The total failover time that might take for traffic switching can be a maximum of 13 seconds.

You can deploy a pair of NetScaler VPX instances with multiple NICs in an active-passive high availability (HA) setup on Azure. Each NIC can contain multiple IP addresses.

The following options are available for a multi-NIC high availability deployment:

- High availability using Azure availability set
- High availability using Azure availability zones

For more information about Azure Availability Set and Availability Zones, see the Azure documentation Manage the availability of Linux virtual machines.

### High availability using availability set

A high availability setup using a availability set must meet the following requirements:

- An HA Independent Network Configuration (INC) configuration
- The Azure Load Balancer (ALB) in Direct Server Return (DSR) mode

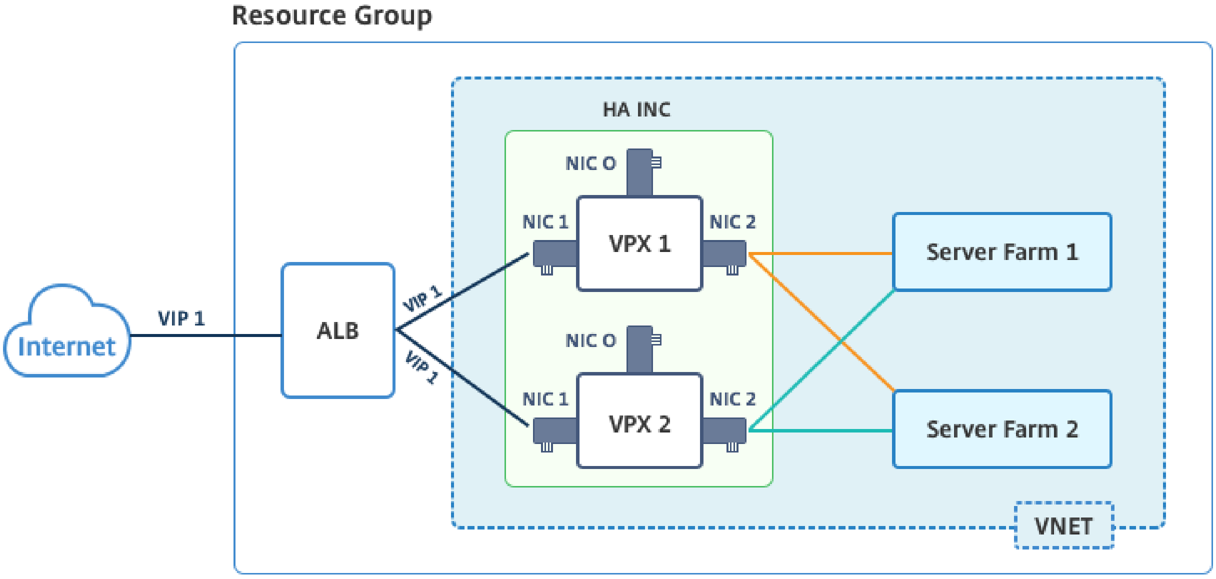
All traffic goes through the primary node. The secondary node remains in standby mode until the primary node fails.

> **Note:**
>
> For a NetScaler VPX high availability deployment on the Azure cloud to work, you need a floating

> public IP (PIP) that can be moved between the two VPX nodes. The Azure Load Balancer (ALB) provides that floating PIP, which is moved to the second node automatically in the event of a failover.

**Diagram**: Example of a high availability deployment architecture, using Azure Availability Set




In an active-passive deployment, the ALB front end public IP (PIP) addresses are added as the VIP addresses in each VPX node. In HA-INC configuration, the VIP addresses are floating and SNIP addresses are instance specific.

You can deploy a VPX pair in active-passive high availability mode in two ways by using:

- **NetScaler VPX standard high availability template**: use this option to configure an HA pair with the default option of three subnets and six NICs.
- **Windows PowerShell commands**: use this option to configure an HA pair according to your subnet and NIC requirements.

This topic describes how to deploy a VPX pair in active-passive HA setup by using the Citrix template. If you want to use PowerShell commands, see Configuring an HA Setup with Multiple IP Addresses and NICs by Using PowerShell Commands.

#### Configure HA-INC nodes by using the NetScaler high availability template

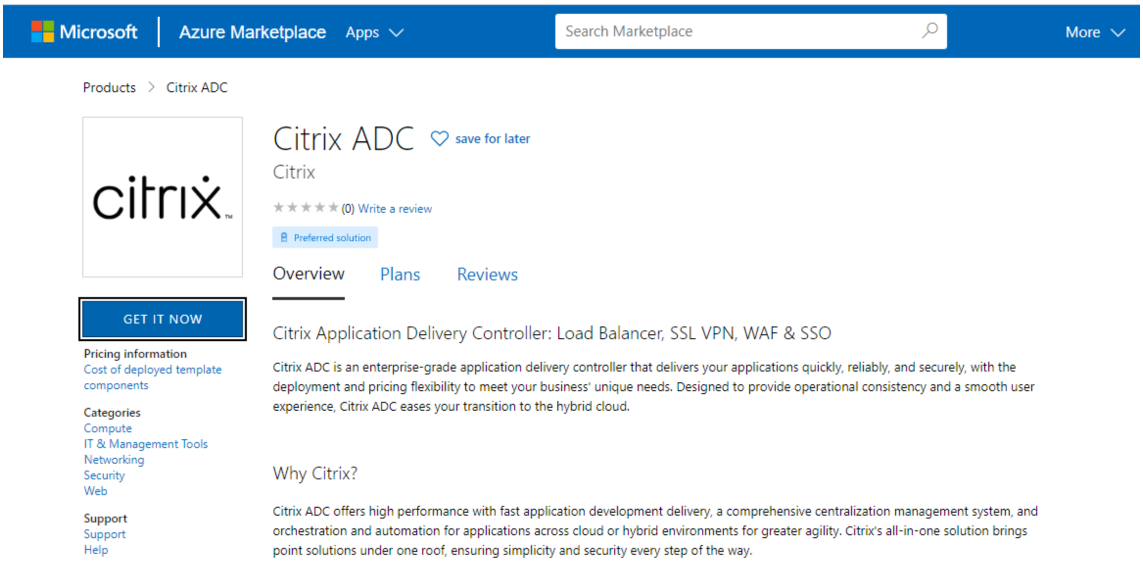
You can quickly and efficiently deploy a pair of VPX instances in HA-INC mode by using the standard template. The template creates two nodes, with three subnets and six NICs. The subnets are for management, client, and server-side traffic, and each subnet has two NICs for both the VPX instances.

You can get the NetScaler HA Pair template at the Azure Marketplace.

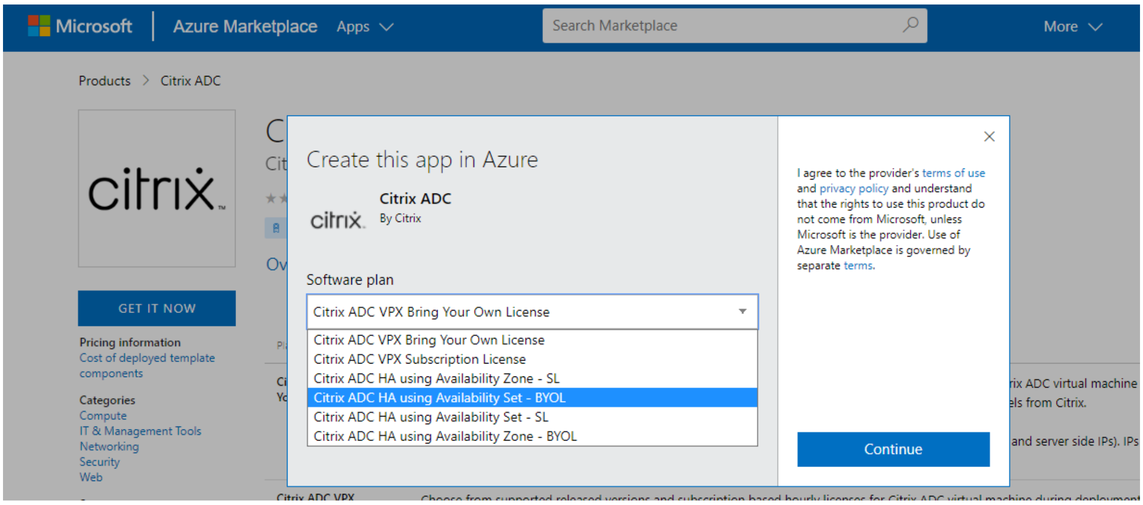
Complete the following steps to launch the template and deploy a high availability VPX pair, by using Azure availability sets.

1. From Azure Marketplace, search **NetScaler**.

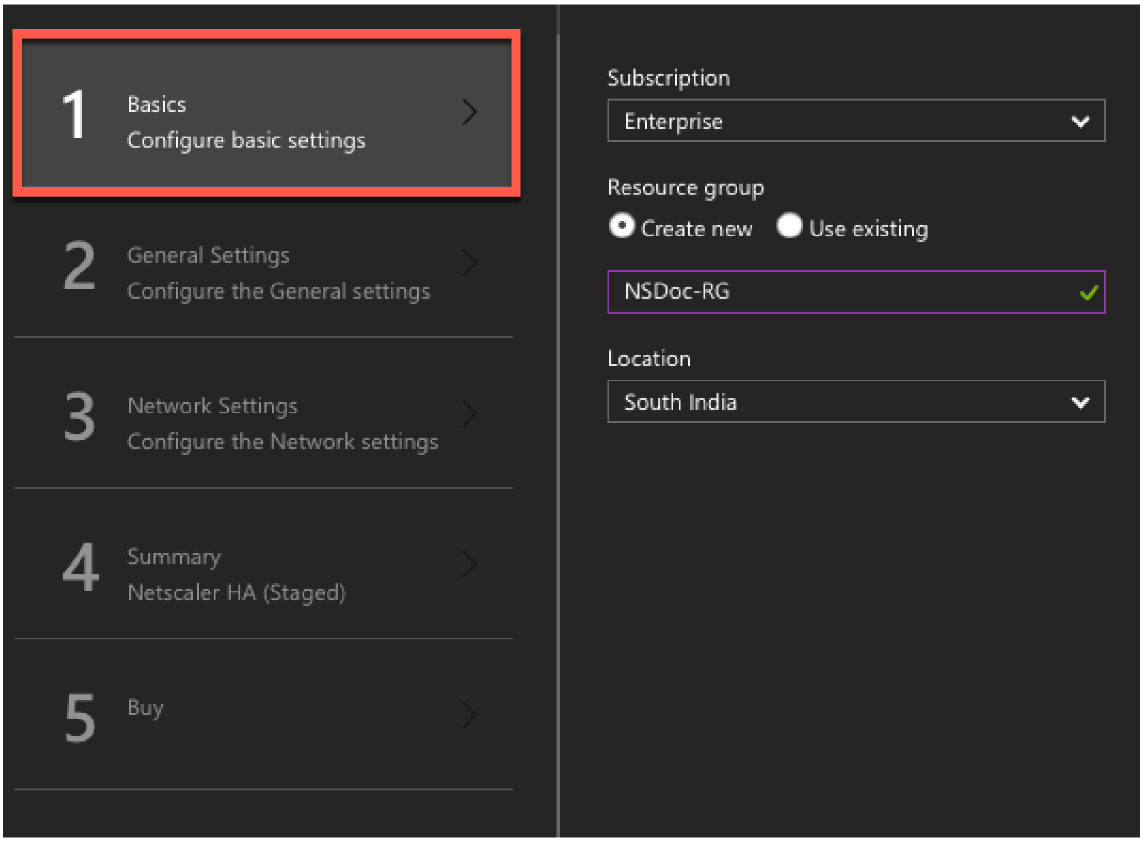



2. Click **GET IT NOW**.

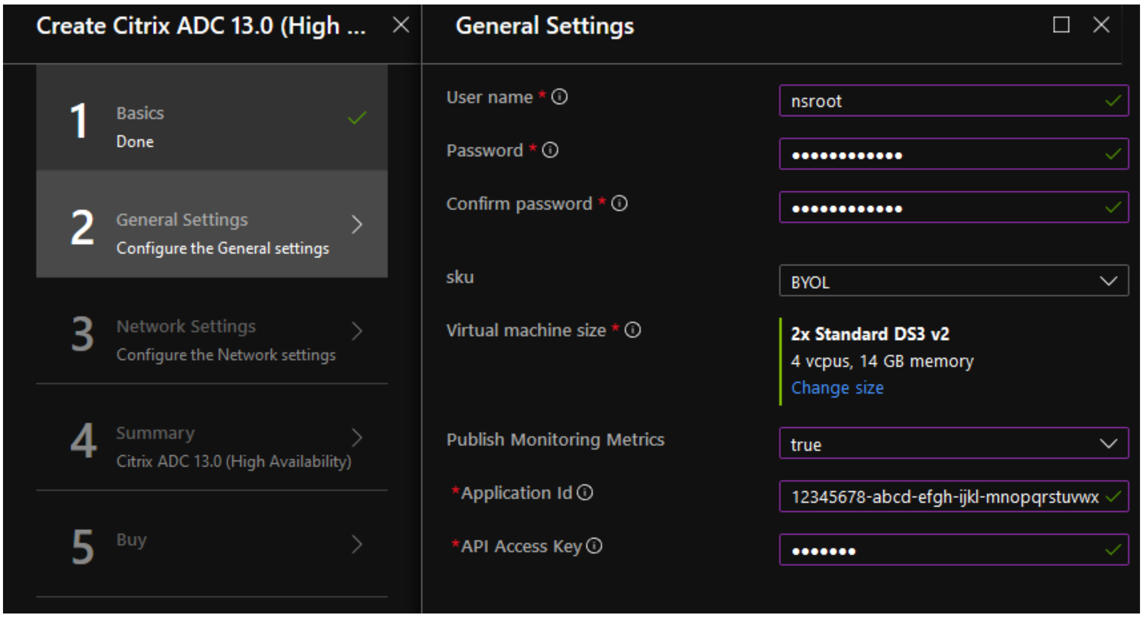
3. Select the required HA deployment along with license, and click **Continue**.




4. The **Basics** page appears. Create a Resource Group and select **OK**.




5. The **General Settings** page appears. Type the details and select **OK**.

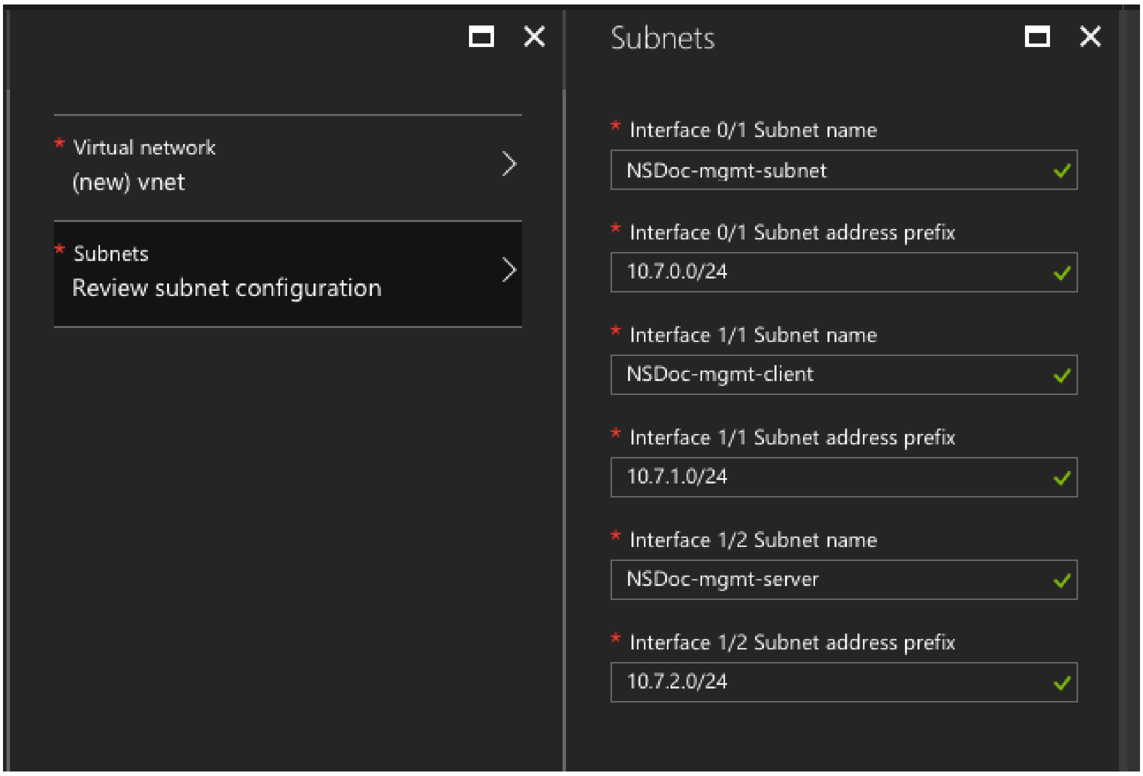



> **Note:**
>
> By default, the **Publishing Monitoring Metrics** option is set to **false**. If you want to enable this option, select **true**.

> Create an Azure Active Directory (ADD) application and service principal that can access resources. Assign contributor role to the newly created AAD application. For more information, see Use portal to create an Azure Active Directory application and service principal that can access resources.

6. The **Network Settings** page appears. Check the VNet and subnet configurations, edit the required settings, and select **OK**.



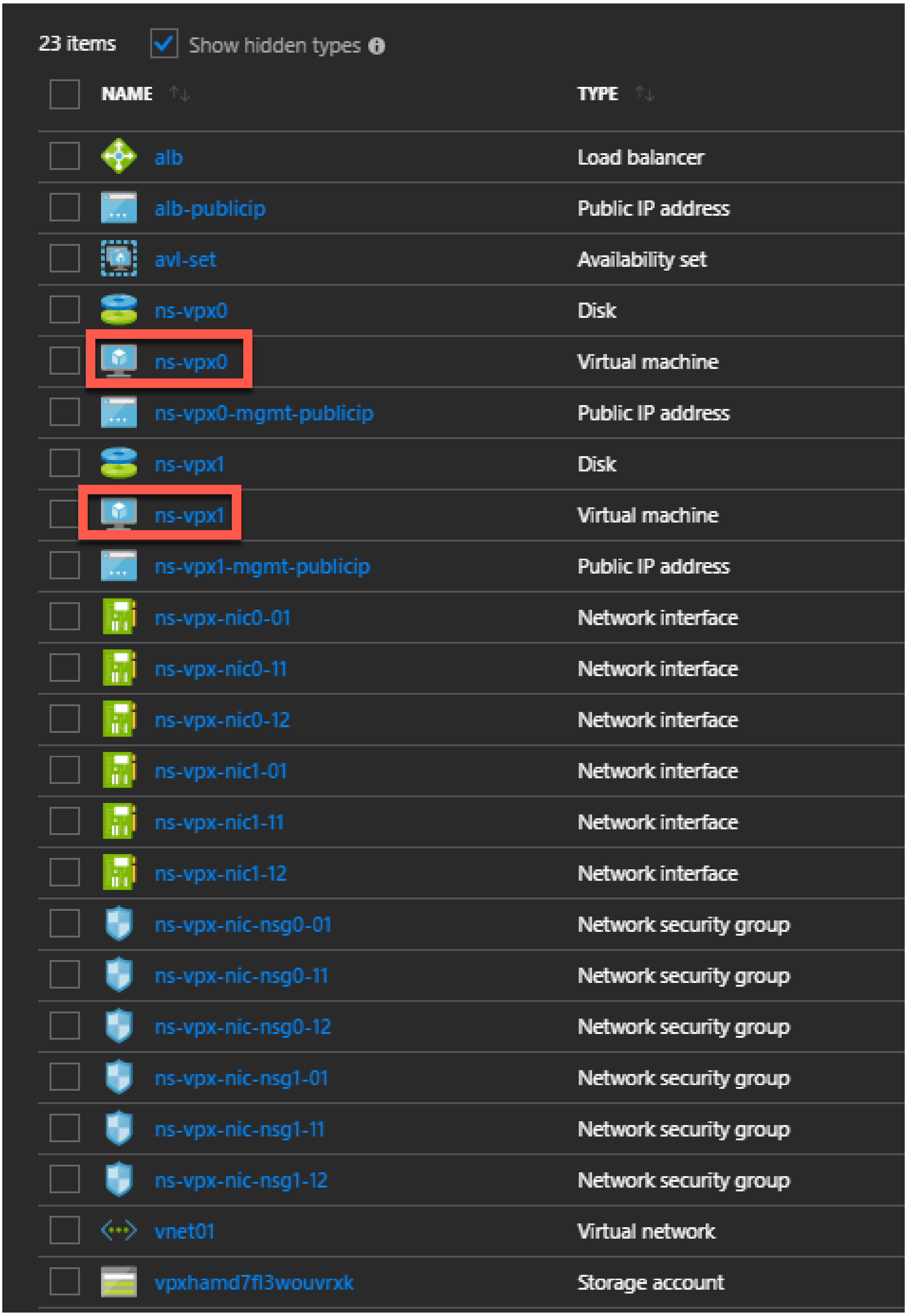

7. The **Summary** page appears. Review the configuration and edit accordingly. Select **OK** to confirm.

8. The **Buy** page appears. Select **Purchase** to complete the deployment.

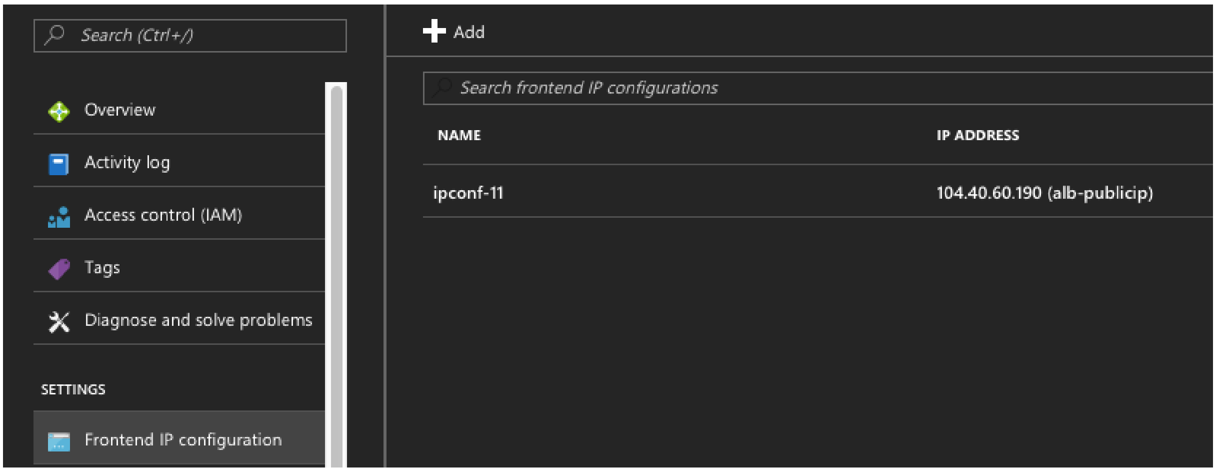
It might take a moment for the Azure Resource Group to be created with the required configurations. After completion, select the **Resource Group** in the Azure portal to see the configuration details, such as LB rules, back-end pools, health probes. The high availability pair appears as ns-vpx0 and ns-vpx1.

If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.


23 items
Show hidden types
NAME
TYPE
alb
Load balancer
alb-publicip
Public IP address
avl-set
Availability set
ns-vpx0
Disk
ns-vpx0
Virtual machine
ns-vpx0-mgmt-publicip
Public IP address
ns-vpx1
Disk
ns-vpx1
Virtual machine
ns-vpx1-mgmt-publicip
Public IP address
ns-vpx-nic0-01
Network interface
ns-vpx-nic0-11
Network interface
ns-vpx-nic0-12
Network interface
ns-vpx-nic1-01
Network interface
ns-vpx-nic1-11
Network interface
ns-vpx-nic1-12
Network interface
ns-vpx-nic-nsg0-01
Network security group
ns-vpx-nic-nsg0-11
Network security group
ns-vpx-nic-nsg0-12
Network security group
ns-vpx-nic-nsg1-01
Network security group
ns-vpx-nic-nsg1-11
Network security group
ns-vpx-nic-nsg1-12
Network security group
vnet01
Virtual network
vpxhamd7fl3wouvrxk
Storage account

Next, you need to configure the load-balancing virtual server with the **ALB's Frontend public IP (PIP) address**, on primary node. To find the ALB PIP, select ALB > **Frontend IP configuration**.




See the **Resources** section for more information about how to configure the load-balancing virtual server.

**Resources**:

The following links provide additional information related to HA deployment and virtual server configuration:

- Configuring high availability nodes in different subnets
- Set up basic load balancing

**Related resources**:

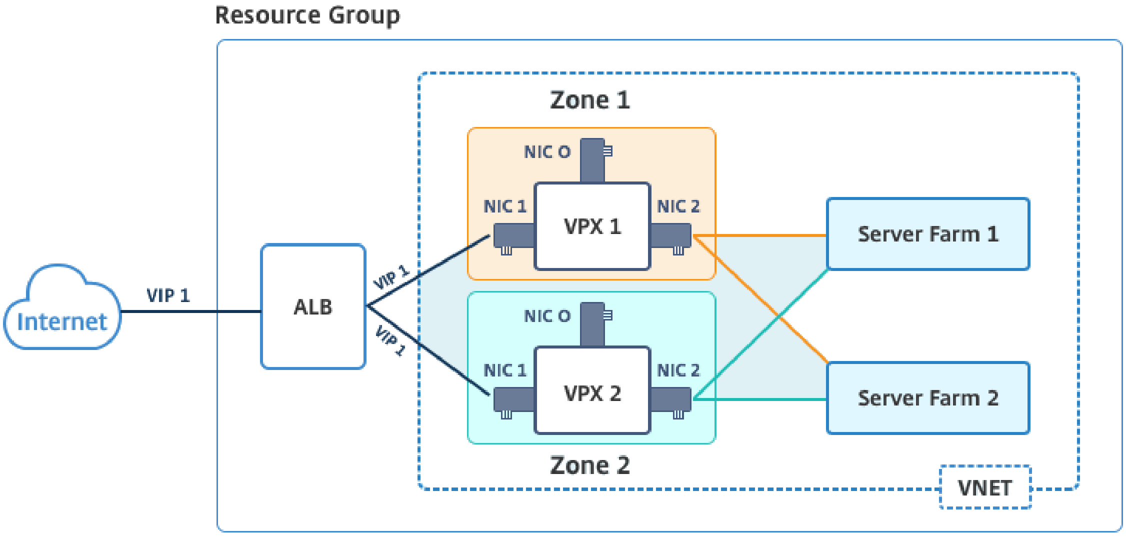
- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands
- Configuring GSLB on Active-Standby HA Deployment on Azure

### High availability using availability zones

Azure Availability Zones are fault-isolated locations within an Azure region, providing redundant power, cooling, and networking and increasing resiliency. Only specific Azure regions support Availability Zones. For more information, see the Azure documentation [What are Availability Zones in Azure.

Diagram: Example of a high availability deployment architecture, using Azure Availability Zones




You can deploy a VPX pair in high availability mode by using the template called "NetScaler 13.0 HA using Availability Zones,"available in Azure Marketplace.

Complete the following steps to launch the template and deploy a high availability VPX pair, by using Azure Availability Zones.

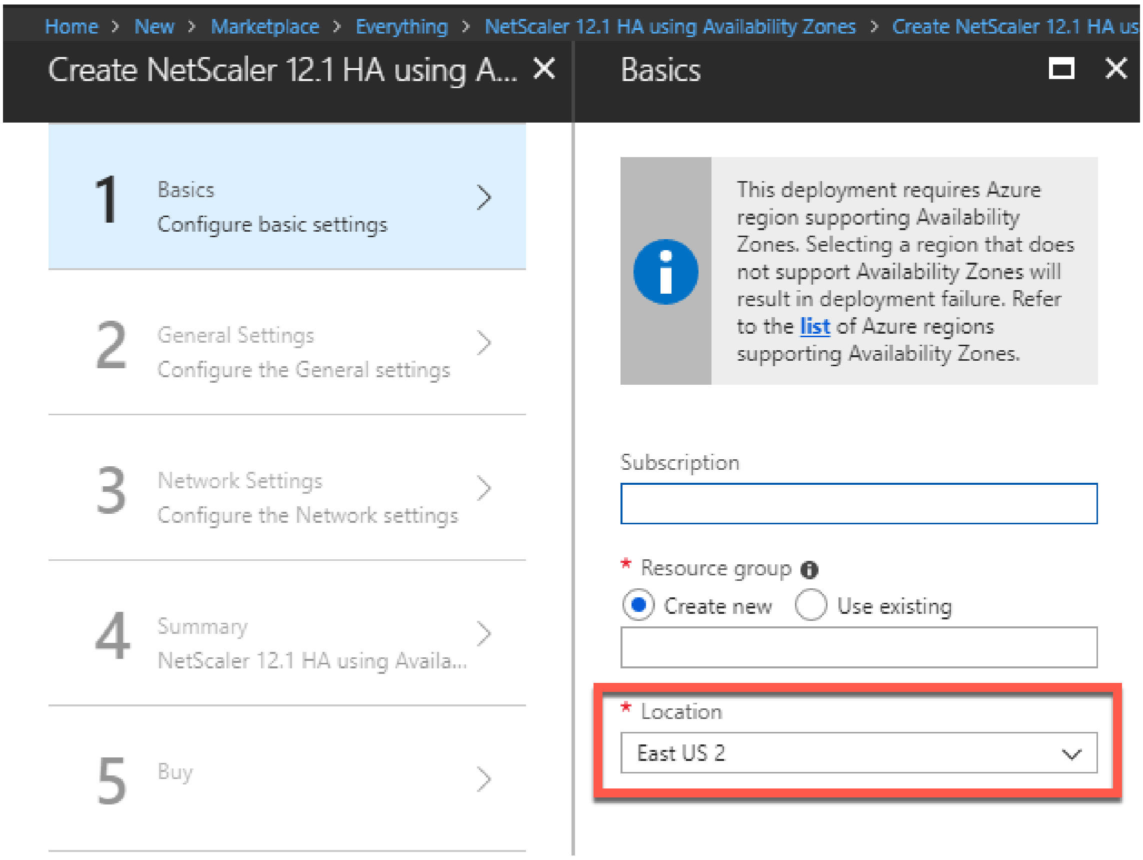
1. From Azure Marketplace, select and initiate the Citrix solution template.

2. Ensure deployment type is Resource Manager and select **Create**.
3. The **Basics** page appears. Enter the details and click **OK**.

   > **Note:**
   >
   > Ensure that you select an Azure region that supports Availability Zones. For more information about regions that support Availability Zones, see Azure documentation What are Availability Zones in Azure?

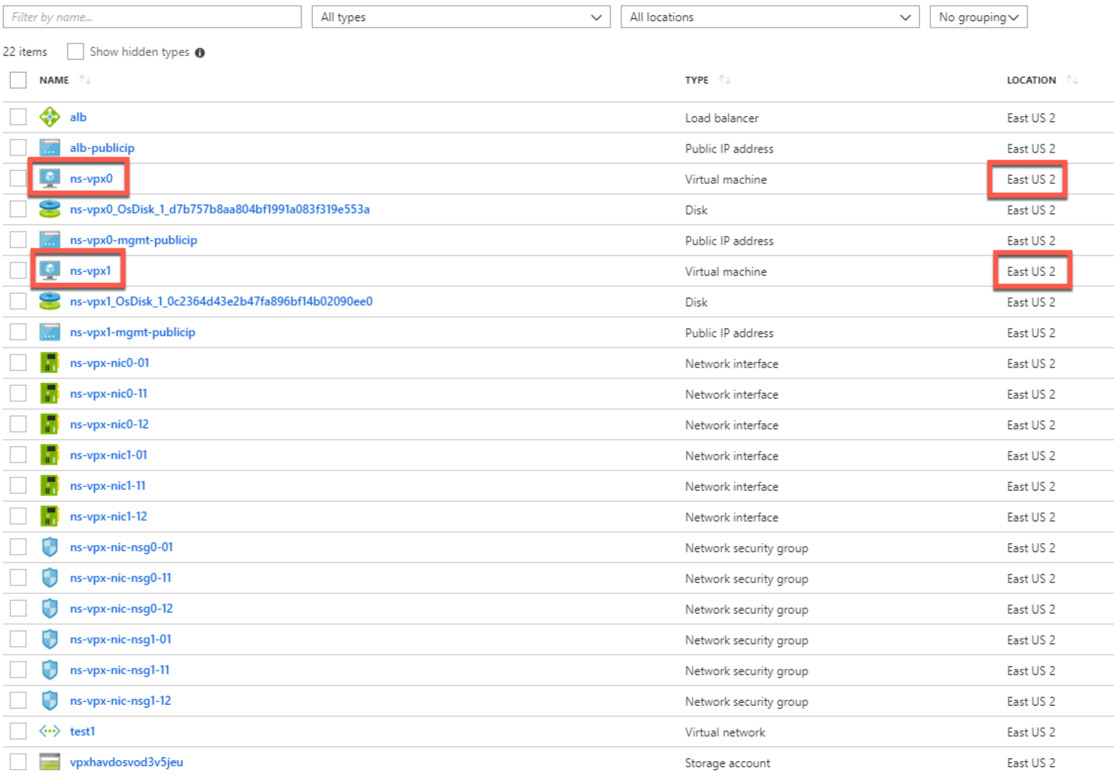
 




4. The **General Settings** page appears. Type the details and select **OK**.
5. The **Network Setting** page appears. Check the VNet and subnet configurations, edit the required settings, and select **OK**.
6. The **Summary** page appears. Review the configuration and edit accordingly. Select **OK** to confirm.
7. The **Buy** page appears. Select **Purchase** to complete the deployment.

   It might take a moment for the Azure Resource Group to be created with the required configurations. After completion, select the **Resource Group** to see the configuration details, such as LB rules, back-end pools, health probes, and so on, in the Azure portal. The high availability pair appears as ns-vpx0 and ns-vpx1. Also, you can see the location under the **Location** column.




If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.

### Monitor your instances using metrics in Azure monitor

You can use metrics in the Azure monitor data platform to monitor a set of NetScaler VPX resources such as CPU, memory utilization, and throughput. Metrics service monitors NetScaler VPX resources that run on Azure, in real time. You can use **Metrics Explorer** to access the collected data. For more information, see Azure Monitor Metrics overview.

#### Points to note

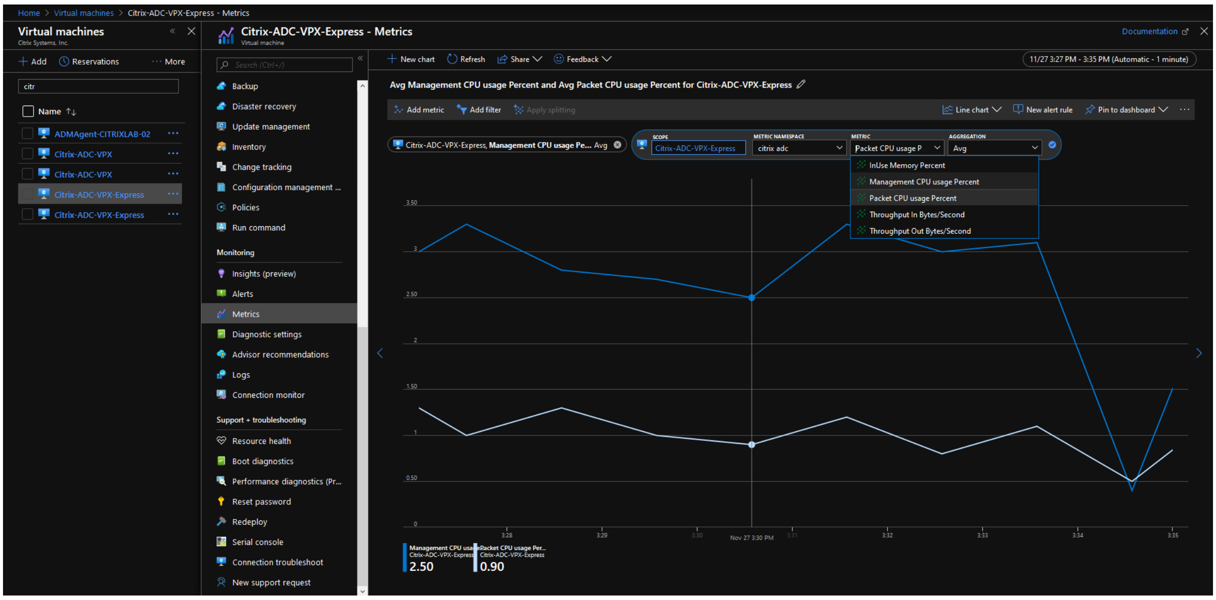
- If you deploy a NetScaler VPX instance on Azure by using the Azure Marketplace offer, Metrics service is disabled by default.
- The Metrics service is not supported in Azure CLI.
- Metrics are available for CPU (management and packet CPU usage), memory, and throughput (inbound and outbound).

### How to view metrics in Azure monitor

To view metrics in the Azure monitor for your instance, perform these steps:

1. Log on to **Azure Portal > Virtual Machines**.
2. Select the virtual machine that is the Primary Node.
3. In the **Monitoring** section, click **Metrics**.
4. From the **Metric Namespace** drop-down menu, click **NetScaler**.
5. Under **All metrics** in **Metrics** drop-down menu, click the metrics you want to view.
6. Click **Add metric** to view another metric on the same chart. Use the Chart options to customize your chart.



## Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands

You can deploy a pair of NetScaler VPX instances with multiple NICs in an active-passive high availability (HA) setup on Azure. Each NIC can contain multiple IP addresses.

An active-passive deployment requires:

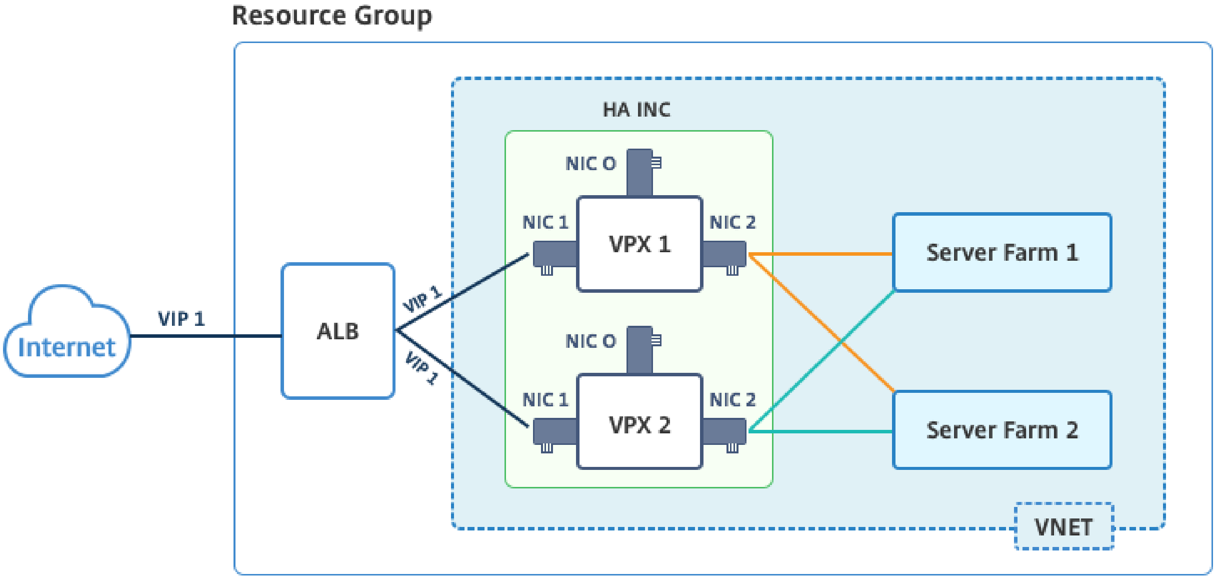
- An HA Independent Network Configuration (INC) configuration
- The Azure Load Balancer (ALB) in Direct Server Return (DSR) mode

All traffic goes through the primary node. The secondary node remains in standby mode until the primary node fails.

> **Note:**
>
> For a NetScaler VPX high availability deployment on an Azure cloud to work, you need a floating public IP (PIP) that can be moved between the two high-availability nodes. The Azure Load Balancer (ALB) provides that floating PIP, which is moved to the second node automatically in the event of a failover.

**Diagram**: Example of an active-passive deployment architecture




In an active-passive deployment, the ALB floating public IP (PIP) addresses are added as the VIP addresses in each VPX node. In HA-INC configuration, the VIP addresses are floating and SNIP addresses are instance specific.

ALB monitors each VPX instance by sending health probe at every 5 seconds and redirects traffic to that instance only that sends health probes response on regular interval. So in an HA setup, the primary node responds to health probes and secondary does not. If the primary instances miss two consecutive health probes, ALB does not redirect traffic to that instance. On failover, the new primary starts responding to health probes and the ALB redirects traffic to it. The standard VPX high availability failover time is three seconds. The total failover time that might take for traffic switching can be maximum of 13 seconds.

You can deploy a VPX pair in active-passive HA setup in two ways by using:

- **NetScaler VPX Standard high availability template**: use this option to configure an HA pair with the default option of three subnets and six NICs.
- **Windows PowerShell commands**: use this option to configure an HA pair according to your subnet and NIC requirements.

This topic describes how to deploy a VPX pair in active-passive HA setup by using PowerShell commands. If you want to use the NetScaler VPX Standard HA template, see Configuring an HA Setup with

Multiple IP Addresses and NICs.

## Configure HA-INC nodes by using PowerShell Commands

### Scenario: HA-INC PowerShell deployment

In this scenario, you deploy a NetScaler VPX pair by using the topology given in the table. Each VPX instance contains three NICs, with each NIC is deployed in a different subnet. Each NIC is assigned an IP configuration.

| ALB | VPX1 | VPX2 |
|---|---|---|
| ALB is associated with public IP 3 (pip3) | Management IP is configured with IPConfig1, which includes one public IP (pip1) and one private IP (12.5.2.24); nic1; Mgmtsubnet=12.5.2.0/24 | Management IP is configured with IPConfig5, which includes one public IP (pip3) and one private IP (12.5.2.26);nic4;Mgmtsubnet=12.5.2.0/24 |
| LB rules and port configured are HTTP (80),SSL (443), health probe (9000) | Client-side IP is configured with IPConfig3, which includes one private IP(12.5.1.27);nic2; FrontEndsubet=12.5.1.0/24 | Client-side IP is configured with IPConfig7, which includes one private IP (12.5.1.28);nic5;FrontEndsubet=12.5.1.0/24 |
| - | Server-side IP is configured with IPConfig4, which includes one private IP(12.5.3.24); nic3;BackendSubnet=12.5.3.0/24 | Server-side IP is configured with IPConfig8, which includes one private IP(12.5.3.28);nic6;BackendSubnet=12.5.3.0/2 |
| - | Rules and ports for NSG are SSH (22),HTTP (80),HTTPS (443) | - |

### Parameter settings

The following parameter settings are used in this scenario:

```
$locName= "South east Asia"

$rgName = "MulitIP-MultiNIC-RG"

$nicName1= "VM1-NIC1"

$nicName2 = "VM1-NIC2"

$nicName3= "VM1-NIC3"
```

```
$nicName4 = "VM2-NIC1"
$nicName5= "VM2-NIC2"
$nicName6 = "VM2-NIC3"
$vNetName = "Azure-MultiIP-ALB-vnet"
$vNetAddressRange= "12.5.0.0/16"
$frontEndSubnetName= "frontEndSubnet"
$frontEndSubnetRange= "12.5.1.0/24"
$mgmtSubnetName= "mgmtSubnet"
$mgmtSubnetRange= "12.5.2.0/24"
$backEndSubnetName = "backEndSubnet"
$backEndSubnetRange = "12.5.3.0/24"
$prmStorageAccountName = "multiipmultinicbstorage"
$avSetName = "multiple-avSet"
$vmSize= "Standard\_DS4\_V2"
$publisher = "Citrix"
$offer = "netscalervpx-120"
$sku = "netscalerbyol"
$version="latest"
$pubIPName1="VPX1MGMT"
$pubIPName2="VPX2MGMT"
$pubIPName3="ALBPIP"
$domName1="vpx1dns"
$domName2="vpx2dns"
$domName3="vpxalbdns"
$vmNamePrefix="VPXMultiIPALB"
$osDiskSuffix1="osmultiipalbdiskdb1"
```

```
$osDiskSuffix2="osmultiipalbdiskdb2"

$lbName= "MultiIPALB"

$frontEndConfigName1= "FrontEndIP"

$backendPoolName1= "BackendPoolHttp"

$lbRuleName1= "LBRuleHttp"

$healthProbeName= "HealthProbe"

$nsgName="NSG-MultiIP-ALB"

$rule1Name="Inbound-HTTP"

$rule2Name="Inbound-HTTPS"

$rule3Name="Inbound-SSH"
```

To complete the deployment, complete the following steps by using PowerShell commands:

1. Create a resource group, storage account, and availability set
2. Create a network security group and add rules
3. Create a virtual network and three subnets
4. Create public IP addresses
5. Create IP configurations for VPX1
6. Create IP configurations for VPX2
7. Create NICs for VPX1
8. Create NICs for VPX2
9. Create VPX1
10. Create VPX2
11. Create ALB

**Create a resource group, storage account, and availability set**.

```
New-AzureRmResourceGroup -Name $rgName -Location $locName


$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $rgName -Type Standard_LRS
     -Location $locName


$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $rgName -Location $locName
```

**Create a network security group and add rules**.

```
$rule1 = New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -
    Description "Allow HTTP" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 101


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 80


$rule2 = New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -
    Description "Allow HTTPS" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 110


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 443


$rule3 = New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -
    Description "Allow SSH" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 120


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 22


$nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
    Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

**Create a virtual network and three subnets**.

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontEndSubnetName -AddressPrefix $frontEndSubnetRange (this
    parameter value should be as per your requirement)


$mgmtSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $mgmtSubnetName
     -AddressPrefix $mgmtSubnetRange


$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backEndSubnetName -AddressPrefix $backEndSubnetRange


$vnet =New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
    $rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
    $frontendSubnet,$backendSubnet, $mgmtSubnet


$subnetName ="frontEndSubnet"


```

```
\$subnet1=\$vnet.Subnets|?{
 \$\_.Name -eq \$subnetName }



$subnetName="backEndSubnet"


\$subnet2=\$vnet.Subnets|?{
 \$\_.Name -eq \$subnetName }



$subnetName="mgmtSubnet"


\$subnet3=\$vnet.Subnets|?{
 \$\_.Name -eq \$subnetName }
```

**Create public IP addresses**.

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $rgName -DomainNameLabel $domName1 -Location $locName -
    AllocationMethod Dynamic

$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $rgName -DomainNameLabel $domName2 -Location $locName -
    AllocationMethod Dynamic

$pip3=New-AzureRmPublicIpAddress -Name $pubIPName3 -ResourceGroupName
    $rgName -DomainNameLabel $domName3 -Location $locName -
    AllocationMethod Dynamic
```

**Create IP configurations for VPX1**.

```
$IpConfigName1 = "IPConfig1"


$IPAddress = "12.5.2.24"


$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip1
    -Primary


$IPConfigName3="IPConfig-3"


$IPAddress="12.5.1.27"


$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
```

```
    Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary


$IPConfigName4 = "IPConfig-4"


$IPAddress = "12.5.3.24"


$IPConfig4 = New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

**Create IP configurations for VPX2**.

```
$IpConfigName5 = "IPConfig5"


$IPAddress="12.5.2.26"


$IPConfig5=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName5 -
    Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip2
    -Primary


$IPConfigName7="IPConfig-7"


$IPAddress="12.5.1.28"


$IPConfig7=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName7 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary


$IPConfigName8="IPConfig-8"


$IPAddress="12.5.3.28"


$IPConfig8=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName8 -
    Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

**Create NICs for VPX1**.

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig1 -
    NetworkSecurityGroupId $nsg.Id


$nic2=New-AzureRmNetworkInterface -Name $nicName2 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig3 -
```

```
NetworkSecurityGroupId $nsg.Id


$nic3=New-AzureRmNetworkInterface -Name $nicName3 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig4 -
    NetworkSecurityGroupId $nsg.Id
```

**Create NICs for VPX2**.

```
$nic4=New-AzureRmNetworkInterface -Name $nicName4 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig5 -
    NetworkSecurityGroupId $nsg.Id


$nic5=New-AzureRmNetworkInterface -Name $nicName5 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig7 -
    NetworkSecurityGroupId $nsg.Id


$nic6=New-AzureRmNetworkInterface -Name $nicName6 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig8 -
    NetworkSecurityGroupId $nsg.Id
```

**Create VPX1**.

This step includes the following substeps:

- Create VM config object
- Set credentials, OS, and image
- Add NICs
- Specify OS disk and create VM

```
$suffixNumber = 1

$vmName=$vmNamePrefix + $suffixNumber

$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id

$cred=Get-Credential -Message "Type the name and password for VPX
     login."

$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred

$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.
    Id -Primary

```

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.
    Id

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.
    Id

$osDiskName=$vmName + "-" + $osDiskSuffix1

$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "
    vhds/" + $osDiskName + ".vhd"

$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -
    VhdUri $osVhdUri -CreateOption fromImage

Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product
    $offer -Name $sku

New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
    $locName
```

**Create VPX2**.

```
```
$suffixNumber=2


$vmName=$vmNamePrefix + $suffixNumber


$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id


$cred=Get-Credential -Message "Type the name and password for VPX login
    ."


$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred


$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version


$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic4.Id -
    Primary


$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic5.Id


$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic6.Id
```
```

```


$osDiskName=$vmName + "-" + $osDiskSuffix2


$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
    + $osDiskName + ".vhd"


$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage


Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku


New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
    $locName
```

To view private and public IP addresses assigned to the NICs, type the following commands:

```
$nic1.IPConfig


$nic2.IPConfig


$nic3.IPConfig


$nic4.IPConfig


$nic5.IPConfig


$nic6.IPConfig
```

**Create Azure load balance (ALB)**.

This step includes the following substeps:

- Create front end IP config
- Create health probe
- Create back end address pool
- Create load-balancing rules (HTTP and SSL)

- Create ALB with front end IP config, back end address pool, and LB rule
- Associate IP config with back end pools

```
$frontEndIP1=New-AzureRmLoadBalancerFrontendIpConfig -Name
$frontEndConfigName1 -PublicIpAddress $pip3
```

```
$healthProbe=New-AzureRmLoadBalancerProbeConfig -Name $healthProbeName
 -Protocol Tcp -Port 9000 –IntervalInSeconds 5 -ProbeCount 2
```

```
$beAddressPool1=New-AzureRmLoadBalancerBackendAddressPoolConfig -
Name $backendPoolName1
```

```
$lbRule1=New-AzureRmLoadBalancerRuleConfig -Name $lbRuleName1
 -FrontendIpConfiguration $frontEndIP1 -BackendAddressPool
$beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort
80 -BackendPort 80 -EnableFloatingIP
```

```
$lb=New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name
$lbName -Location $locName -FrontendIpConfiguration $frontEndIP1
-LoadBalancingRule $lbRule1 -BackendAddressPool $beAddressPool1 -
Probe $healthProbe
```

```
$nic2.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb
.BackendAddressPools[0])
```

```
$nic5.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb
.BackendAddressPools[0])
```

```
$lb=$lb |Set-AzureRmLoadBalancer
```

```
$nic2=$nic2 | Set-AzureRmNetworkInterface
```

```
$nic5=$nic5 | Set-AzureRmNetworkInterface
```

After you've successfully deployed the NetScaler VPX pair, log on to each VPX instance to configure HA-INC, and SNIP and VIP addresses.

1. Type the following command to add HA nodes.

   ```
   add ha node 1 PeerNodeNSIP -inc Enabled
   ```

2. Add private IP addresses of client-side NICs as SNIPs for VPX1 (NIC2) and VPX2 (NIC5)

   ```
   add nsip privateIPofNIC2 255.255.255.0 -type SNIP
   add nsip privateIPofNIC5 255.255.255.0 -type SNIP
   ```

3. Add load-balancing virtual server on the primary node with front-end IP address (public IP) of ALB.

   ```
   add lb virtual server v1 HTTP FrontEndIPofALB 80
   ```

**Related resources**:

Configuring GSLB on Active-Standby HA Deployment on Azure

## Deploy a NetScaler high-availability pair on Azure with ALB in the floating IP-disabled mode

You can deploy a pair of NetScaler VPX instances with multiple NICs in an active-passive high availability (HA) setup on Azure. Each NIC can contain many IP addresses.

An active-passive deployment requires:

- An HA Independent Network Configuration (INC) configuration
- The Azure Load Balancer (ALB) with:
  - Floating IP-enabled mode or Direct Server Return (DSR) mode
  - Floating IP-disabled mode

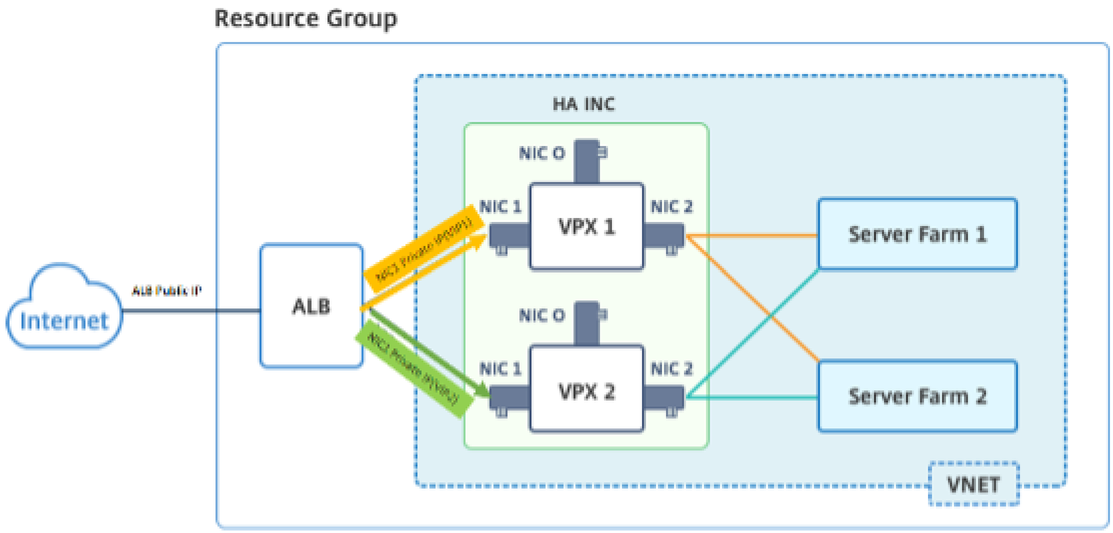
For more information about ALB floating IP options, refer to the Azure documentation.

If you want to deploy a VPX pair in active-passive HA setup on Azure with ALB floating IP enabled, see Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands.

### HA deployment architecture with ALB in the floating IP-disabled mode

In an active-passive deployment, the private IP addresses of the client interface of each instance are added as VIP addresses in each VPX instance. Configure in the HA-INC mode with VIP addresses being shared using IPSet and SNIP addresses being instance specific. All traffic goes through the primary instance. The secondary instance is in standby mode until the primary instance fails.

**Diagram**: Example of an active-passive deployment architecture




### Prerequisites

You must be familiar with the following information before deploying a NetScaler VPX instance on Azure.

- Azure terminology and network details. For more information, see Azure terminology.
- Working of a NetScaler appliance. For more information, see NetScaler documentation.
- NetScaler networking. For more information, see the ADC Networking.
- Azure load balancer and load-balancing rule configuration. For more information, see Azure ALB documentation.

### How to deploy a VPX HA pair on Azure with ALB floating IP disabled

Here's a summary of the HA and ALB deployment steps:

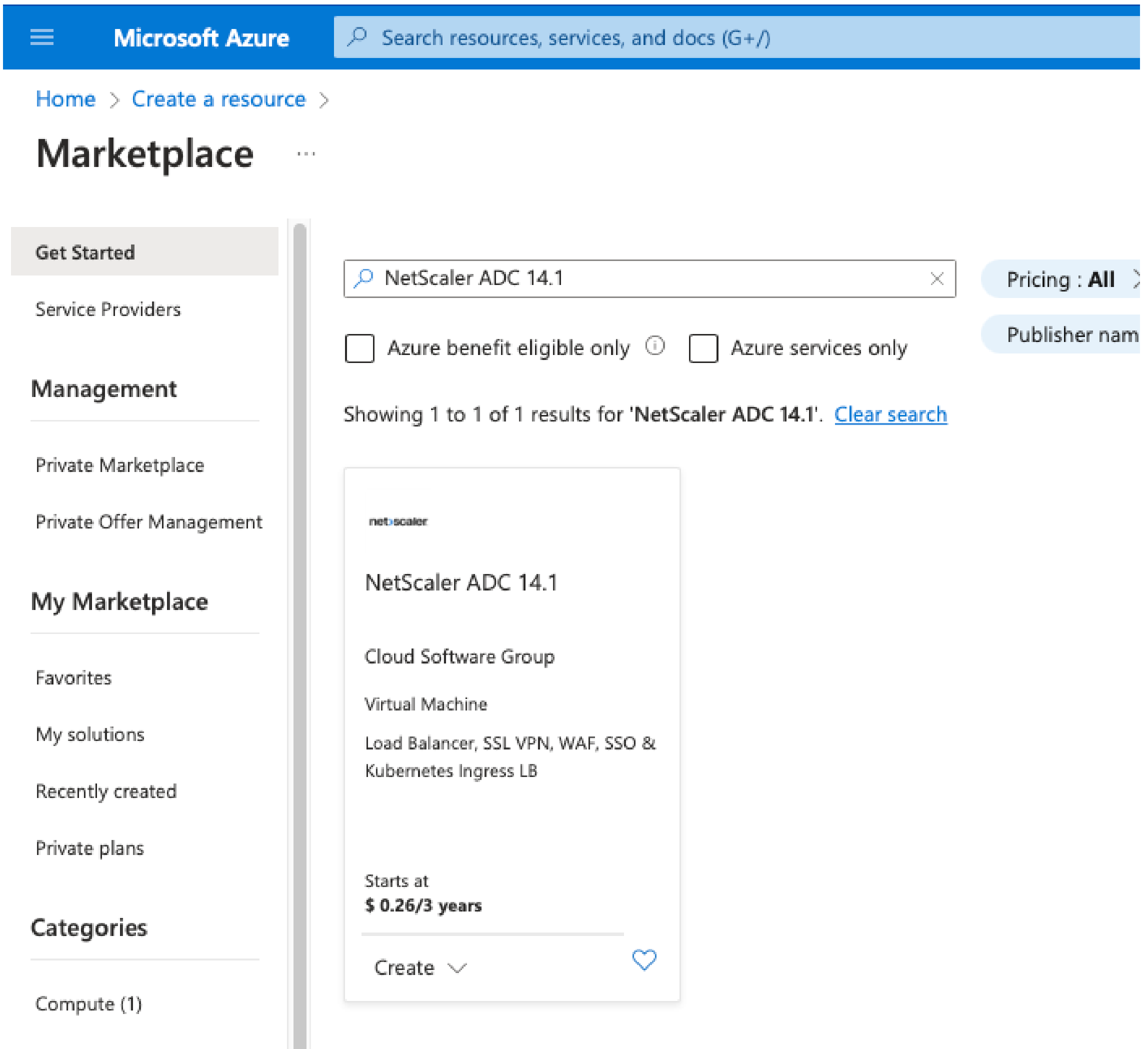
1. Deploy two VPX instances (primary and secondary instances) on Azure.
2. Add client and server NIC on both the instances.
3. Deploy an ALB with load balancing rule whose floating IP mode is disabled.
4. Configure HA settings on both instances by using the NetScaler GUI.

**Step 1. Deploy two VPX instances on Azure.**

Create two VPX instances by following these steps:

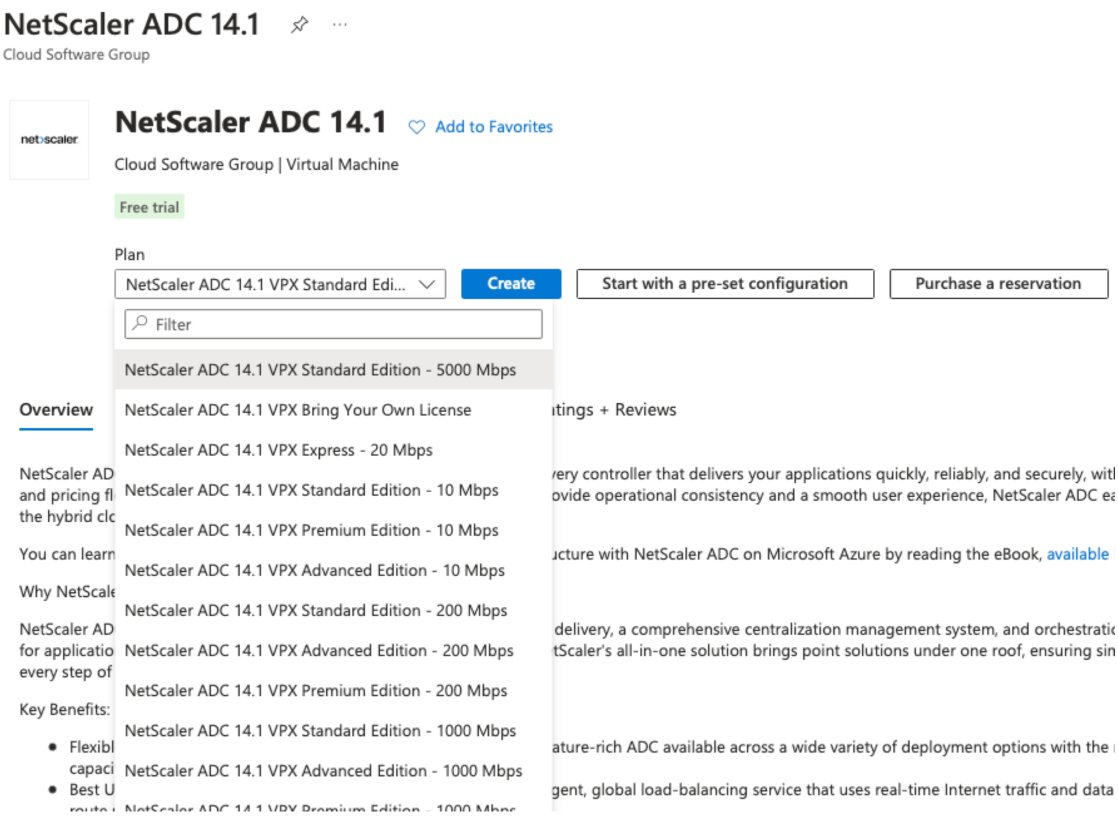
1. Select the NetScaler version from Azure Marketplace (in this example, NetScaler release 13.1 is used).




2. Select the required ADC licensing mode, and click **Create**.

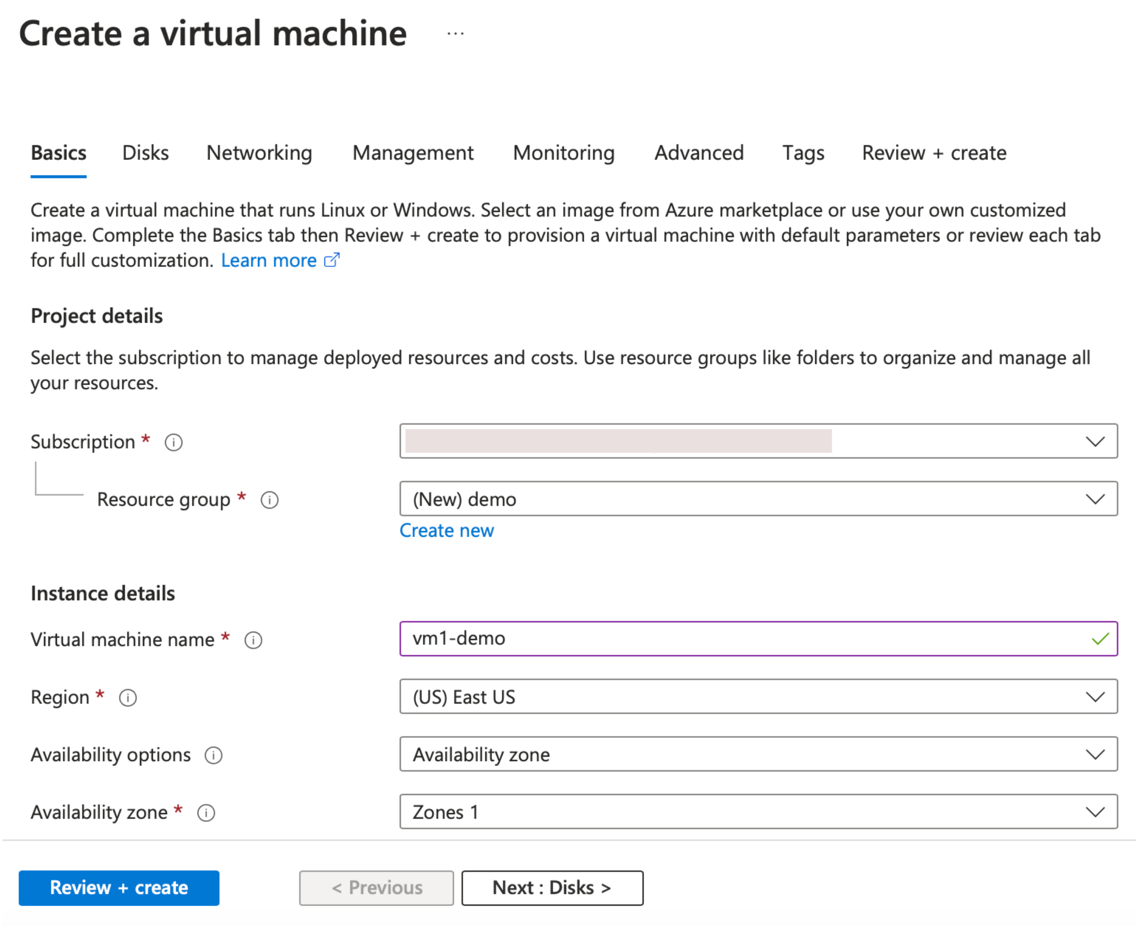
 




The **Create a virtual machine** page opens.

3. Complete the required details in each tab: Basics, Disks, Networking, Management, Monitoring, Advanced, and Tags, for a successful deployment.

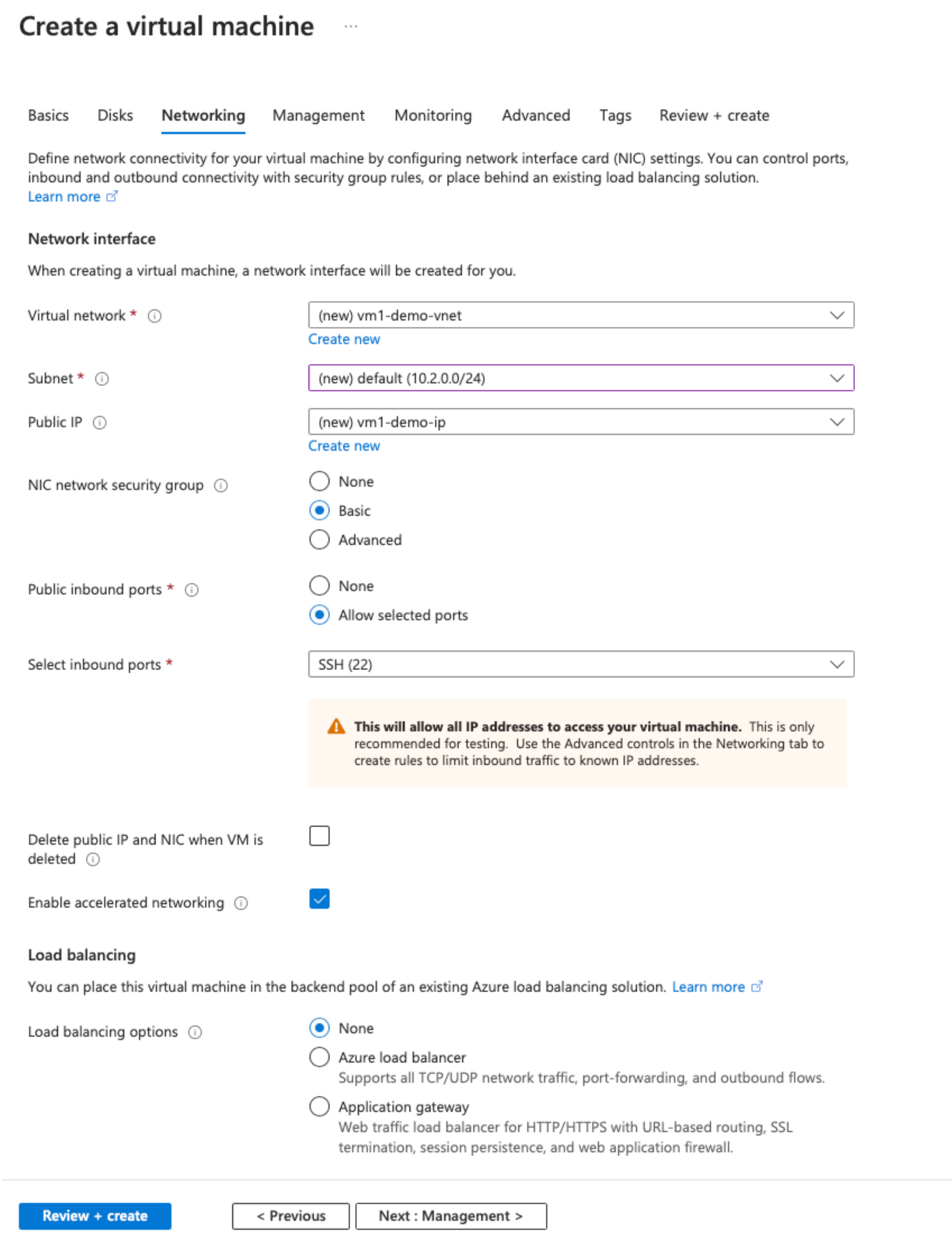



4. In the **Networking** tab, create a new Virtual network with 3 subnets, one each for: management, client, and server NICs. Otherwise, you can also use an existing Virtual network. Management NIC is created during the VM deployment. Client and server NICs are created and attached after the VM is created. For the NIC network security group, you can do one of the following:

   - Select **Advanced** and use an existing network security group that suits your requirements.
   - Select **Basic** and select the required ports.

   > **Note:**
   >
   > You can also change the network security group settings after the VM deployment is completed.

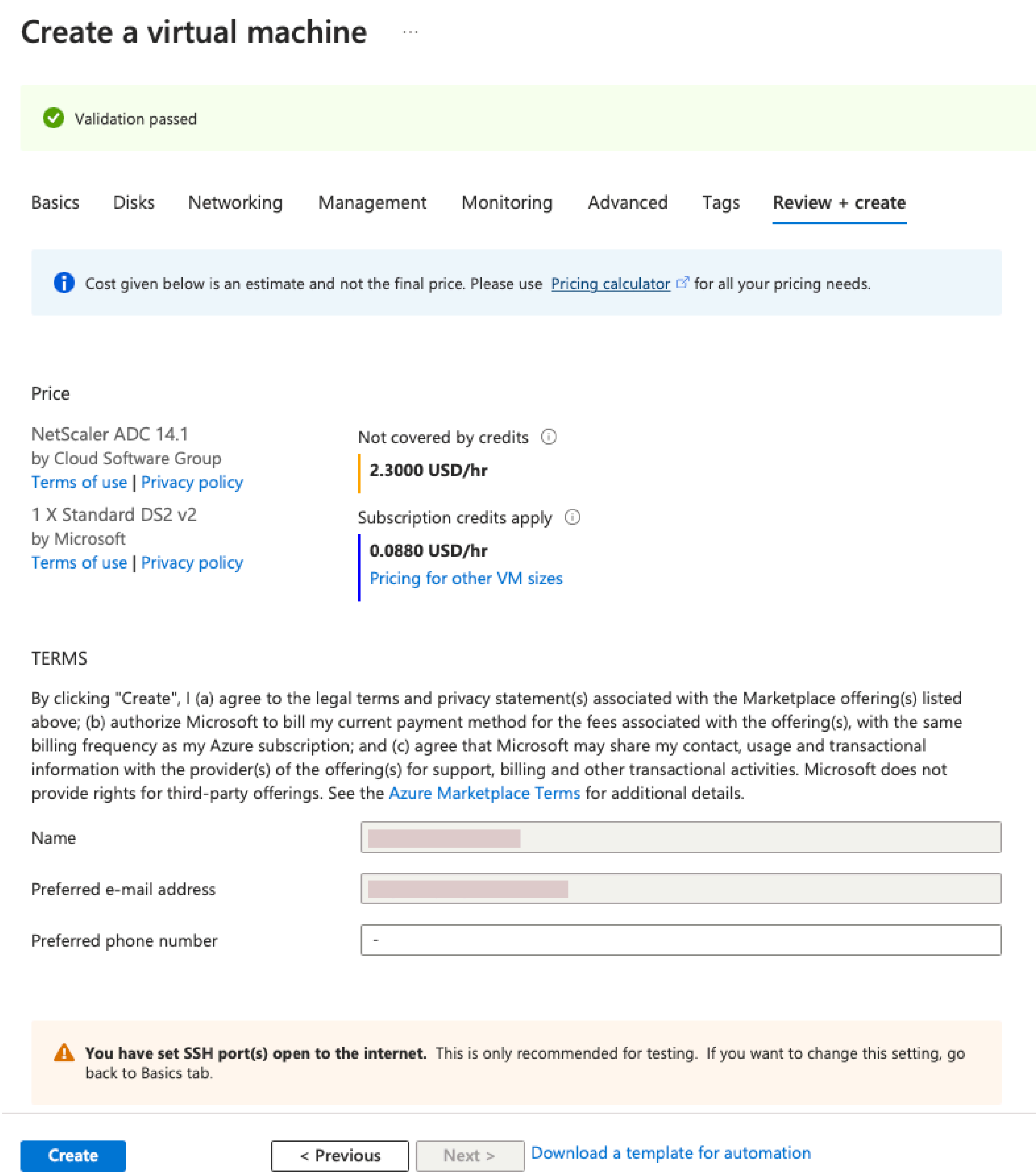
 




5. Click Next: **Review + create >**.

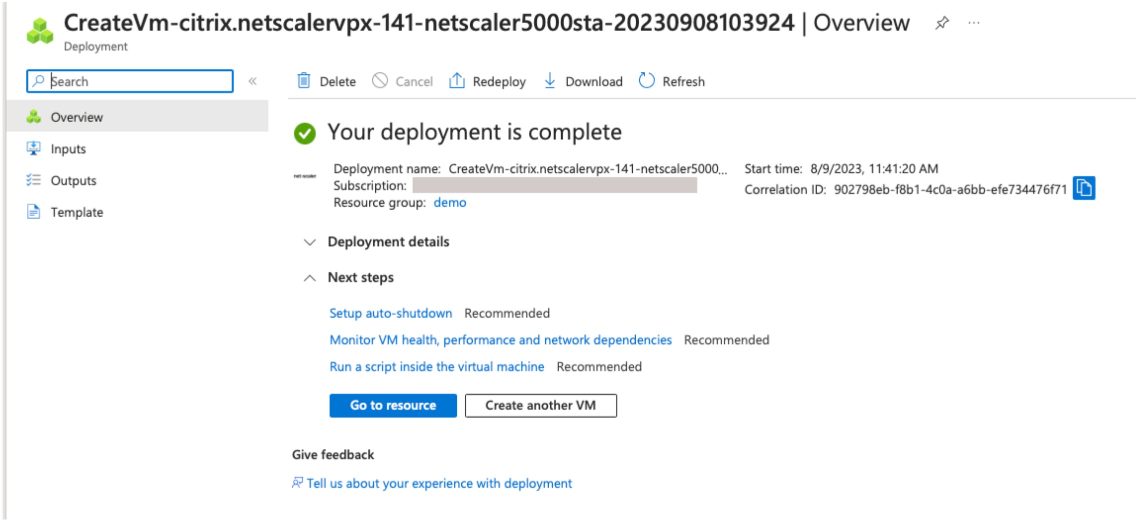
   After the validation is successful, review the basic settings, VM configurations, network and additional settings, and click **Create**.




6. After the deployment is complete, Click **Go to Resource** to see the configuration details.




Similarly, deploy a second NetScaler VPX instance.

**Step 2. Add client and server NICs on both instances.**

> **Note:**
>
> To attach more NICs, you must first stop the VM. In the Azure portal, select the VM that you want to stop. In the **Overview** tab, click **Stop**. Wait for Status to show as **Stopped**.

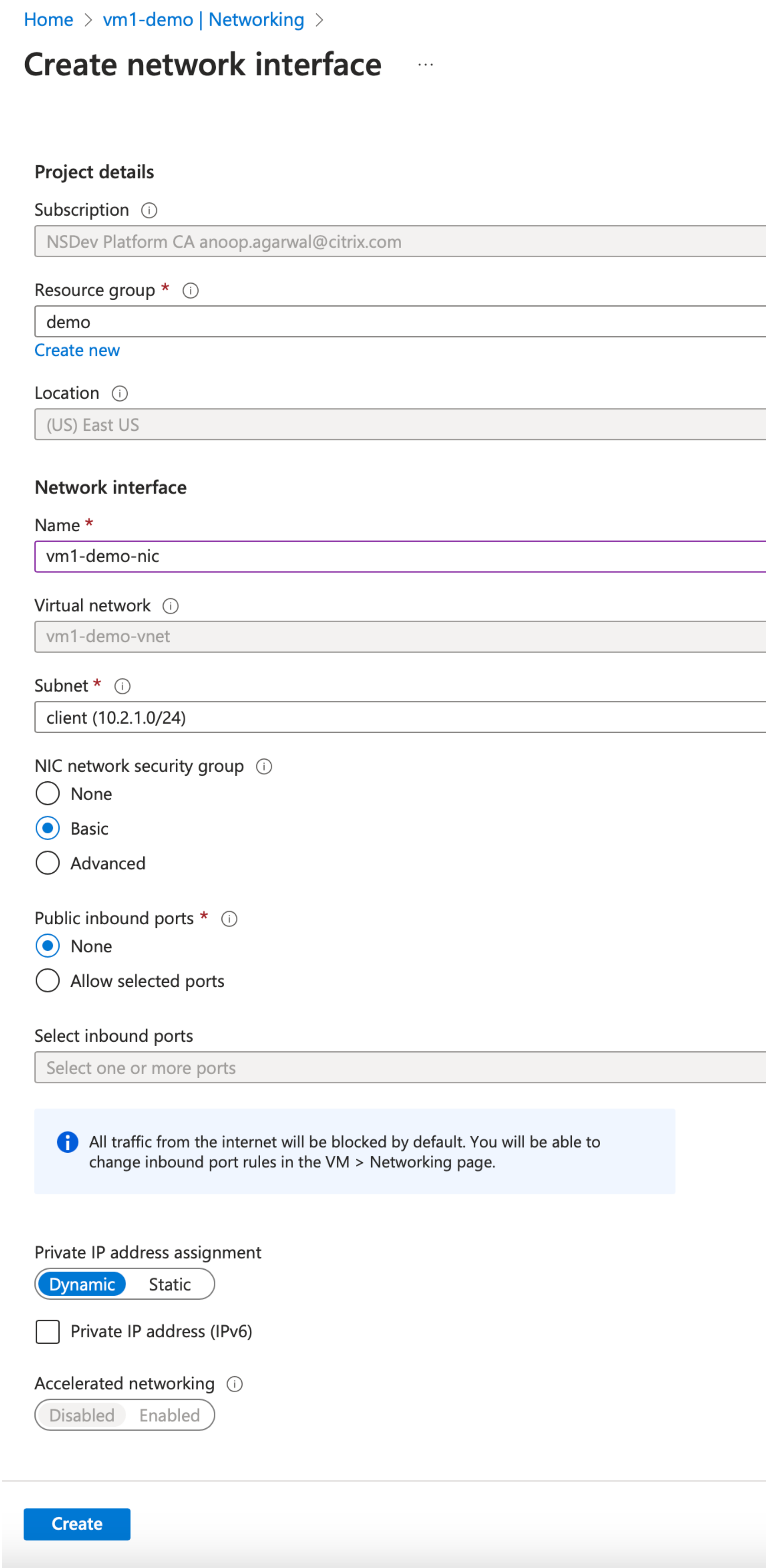
To add a client NIC on the primary instance, follow these steps:

1. Navigate to **Networking > Attach Network Interface**.

   You can select an existing NIC or create and attach a new interface.

2. For the NIC Network Security Group, you can use an existing network security group by selecting **Advanced** or create one by selecting **Basic**.


Home > vm1-demo | Networking >
Create network interface ...
Project details
Subscription
NSDev Platform CA anoop.agarwal@citrix.com
Resource group *
demo
Create new
Location
(US) East US
Network interface
Name *
vm1-demo-nic
Virtual network
vm1-demo-vnet
Subnet *
client (10.2.1.0/24)
NIC network security group
None
Basic
Advanced
Public inbound ports *
None
Allow selected ports
Select inbound ports
Select one or more ports
All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.
Private IP address assignment
Dynamic
Static
Private IP address (IPv6)
Accelerated networking
Disabled
Enabled
Create

To add a server NIC, follow the same steps as for adding a client NIC.

The NetScaler VPX instance has all three NICs (management NIC, client NIC, and server NIC) attached.

Repeat the preceding steps for adding NICs on the secondary instance.

After you create and attach NICs on both the instances, restart both the instances by going to **Overview > Start**.

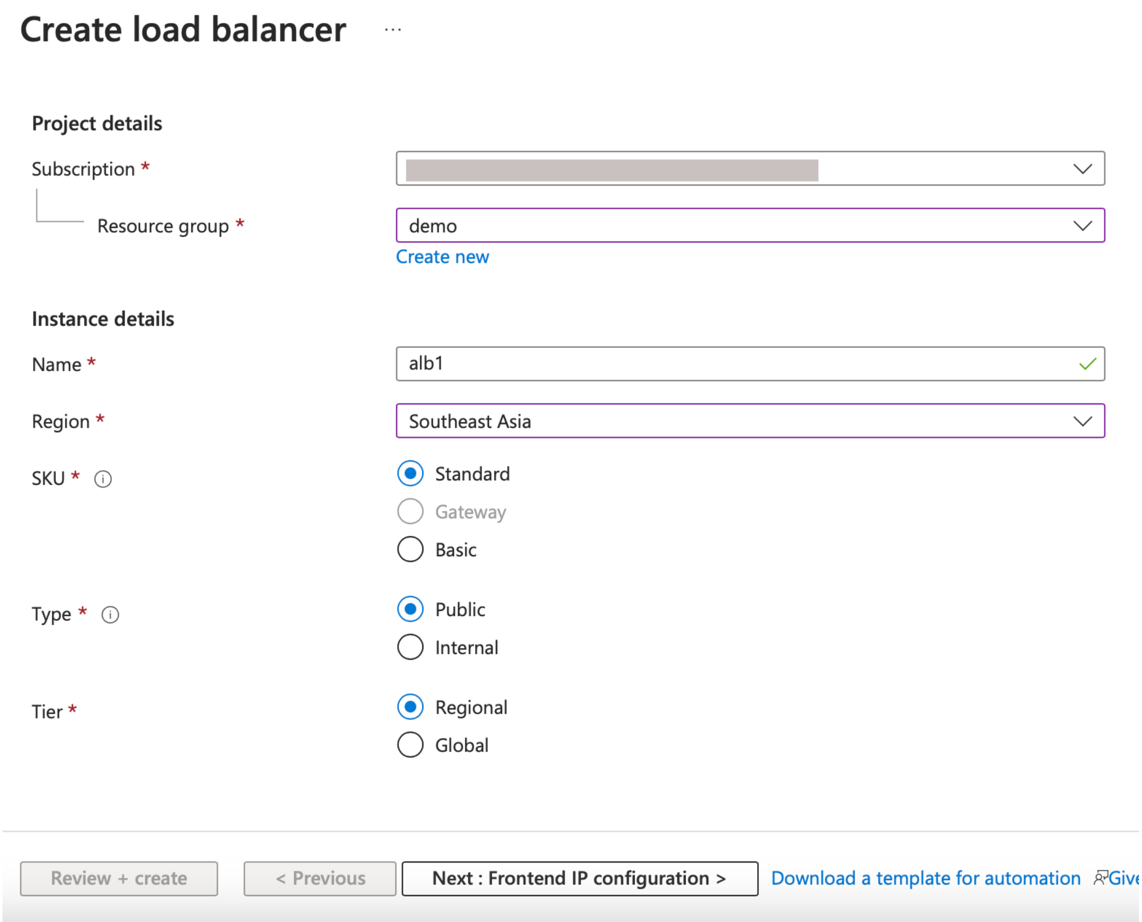
> **Note:**
>
> You must allow traffic through the port in client NIC inbound rule, which is used later to create a load balancing virtual server while configuring the NetScaler VPX instance.

**Step 3. Deploy an ALB with load balancing rule whose floating IP mode is disabled.**

To start the configuration of ALB, follow these steps:

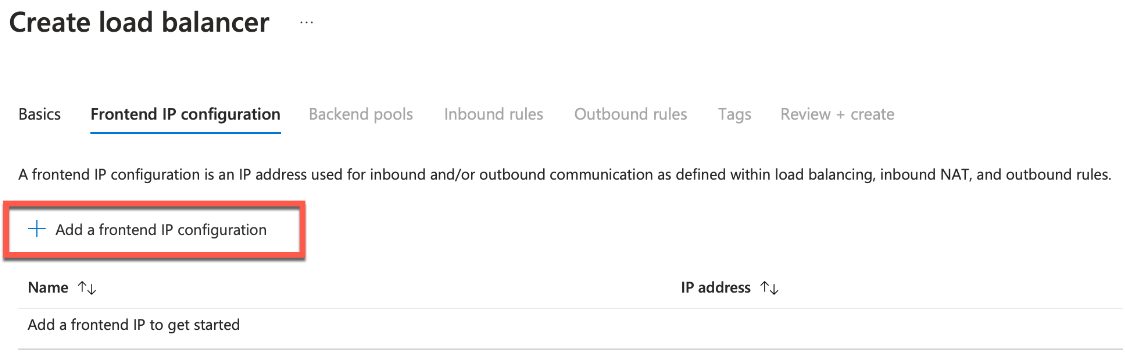
1. Go to the **Load balancers** page and click **Create**.
2. In the **Create load balancer** page, provide the details as required.

   In the following example, we deploy a regional public load balancer of Standard SKU.

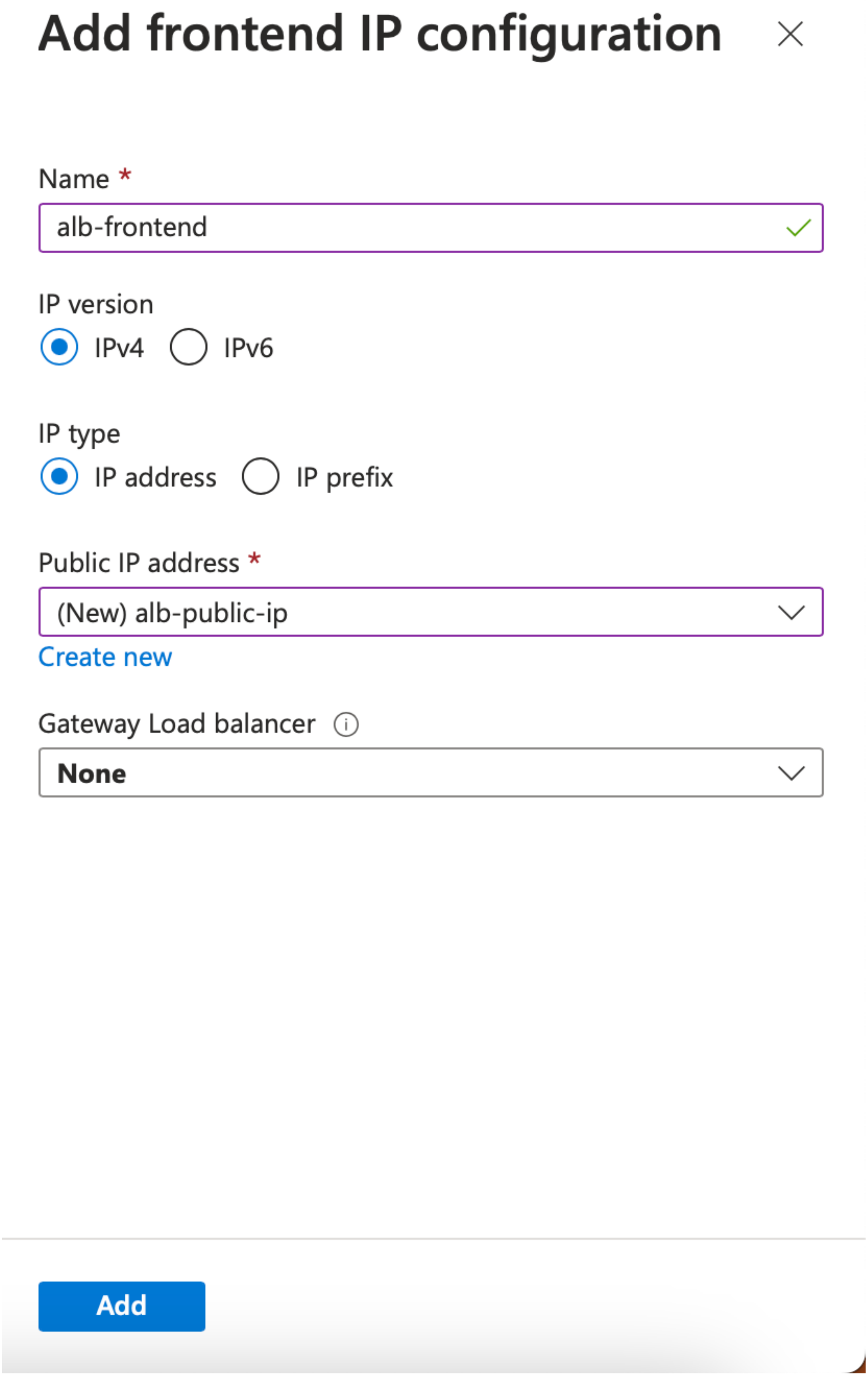


> **Note:**
>
> All public IPs attached to the NetScaler VMs must have the same SKU as that of ALB. For more information about ALB SKUs, see the Azure Load Balancer SKUs'documentation.

3. In the **Frontend IP configuration** tab, either create an IP address or use an existing IP address.

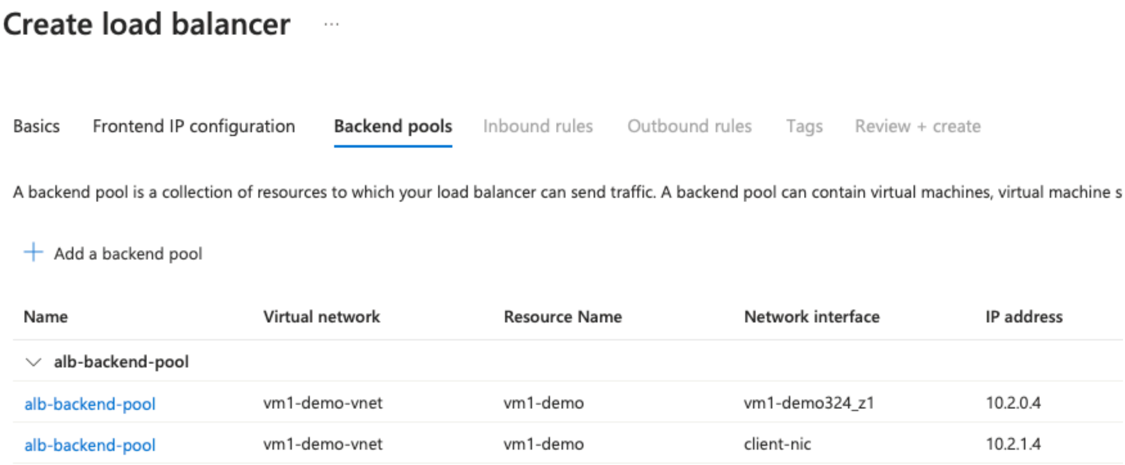


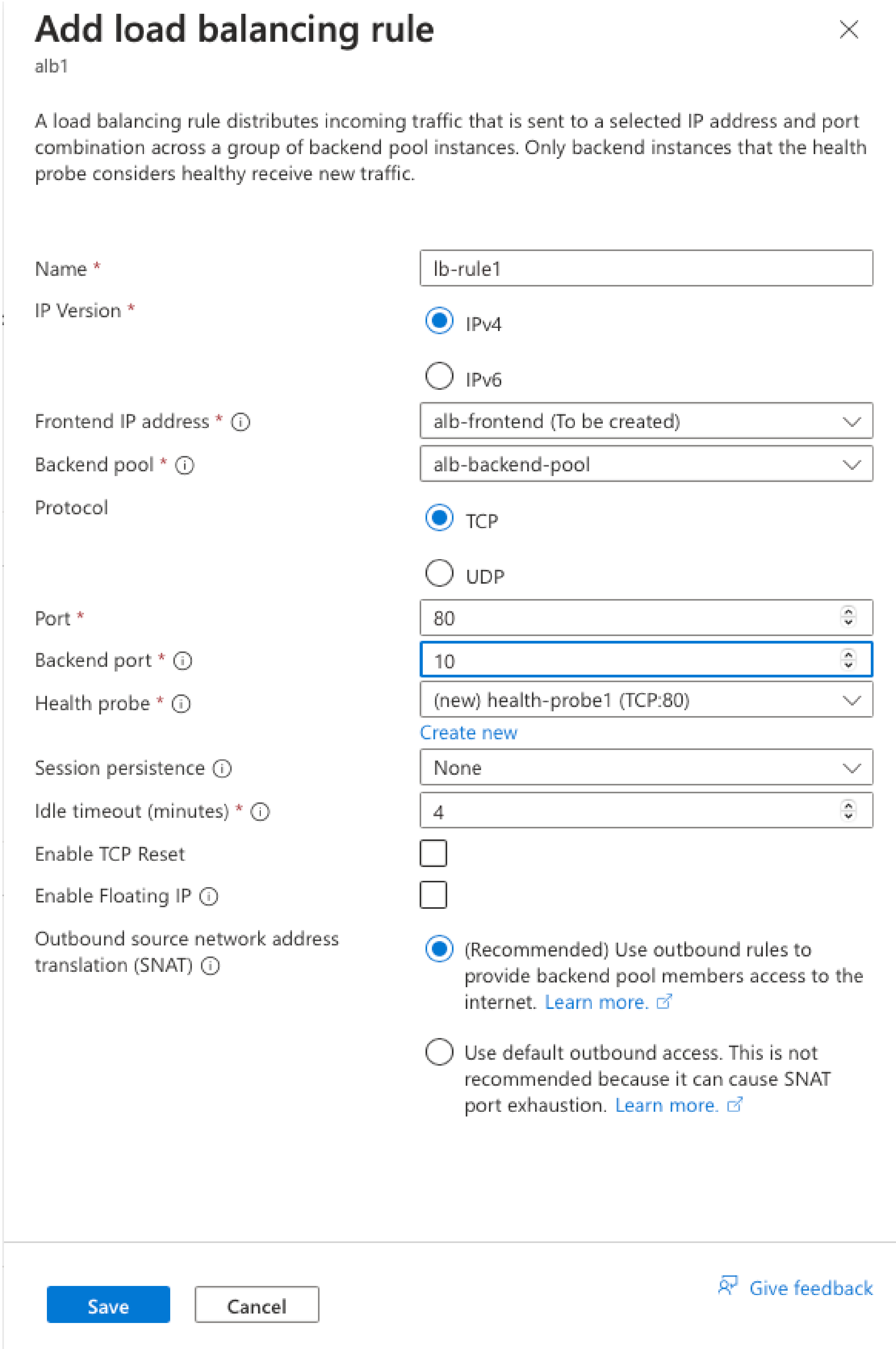

Add frontend IP configuration
Name *
alb-frontend
IP version
IPv4
IPv6
IP type
IP address
IP prefix
Public IP address *
(New) alb-public-ip
Create new
Gateway Load balancer
None
Add

4. In the **Backend pools** tab, select NIC-based backend pool configuration, and add the client NICs of both the NetScaler VMs.

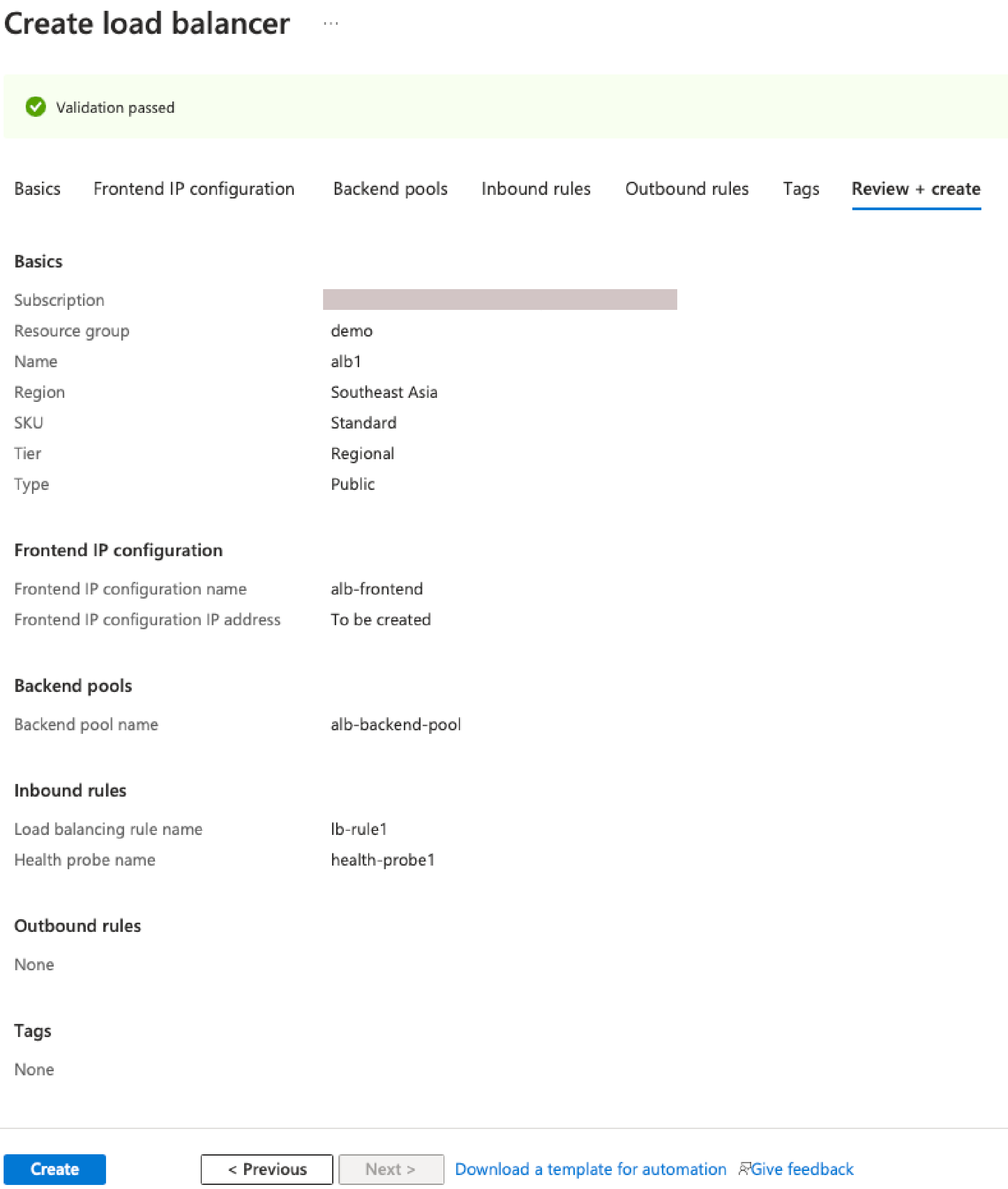



5. In **Inbound rules** tab, click **Add a Load balancing rule**, and provide the frontend IP address and backend pool created in the previous steps. Select the protocol and port based on your requirement. Create or use an existing health probe. The floating IP option must be set as **Disabled**.


Add load balancing rule
alb1
A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend pool instances. Only backend instances that the health probe considers healthy receive new traffic.
Name *
lb-rule1
IP Version *
IPv4
IPv6
Frontend IP address *
alb-frontend (To be created)
Backend pool *
alb-backend-pool
Protocol
TCP
UDP
Port *
80
Backend port *
10
Health probe *
(new) health-probe1 (TCP:80)
Create new
Session persistence
None
Idle timeout (minutes) *
4
Enable TCP Reset
Enable Floating IP
Outbound source network address translation (SNAT)
(Recommended) Use outbound rules to provide backend pool members access to the internet. Learn more.
Use default outbound access. This is not recommended because it can cause SNAT port exhaustion. Learn more.
Save
Cancel
Give feedback

6. Click **Review + Create**. After the validation is passed, click **Create**.




**Step 4. Configure HA settings on both NetScaler VPX instances by using the NetScaler GUI.**

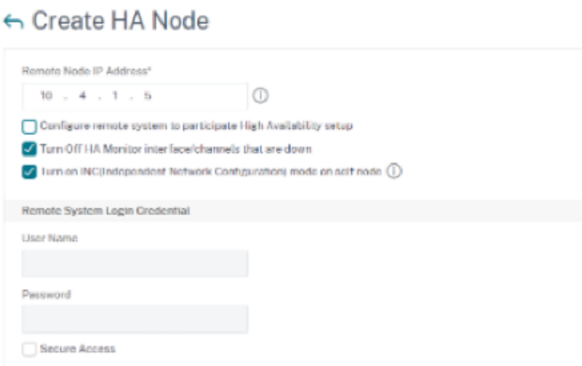
After you have created the NetScaler VPX instances on Azure, you can configure HA by using the NetScaler GUI.

**Step 1. Set up high availability in INC mode on both the instances.**

On the primary instance, do the following steps:

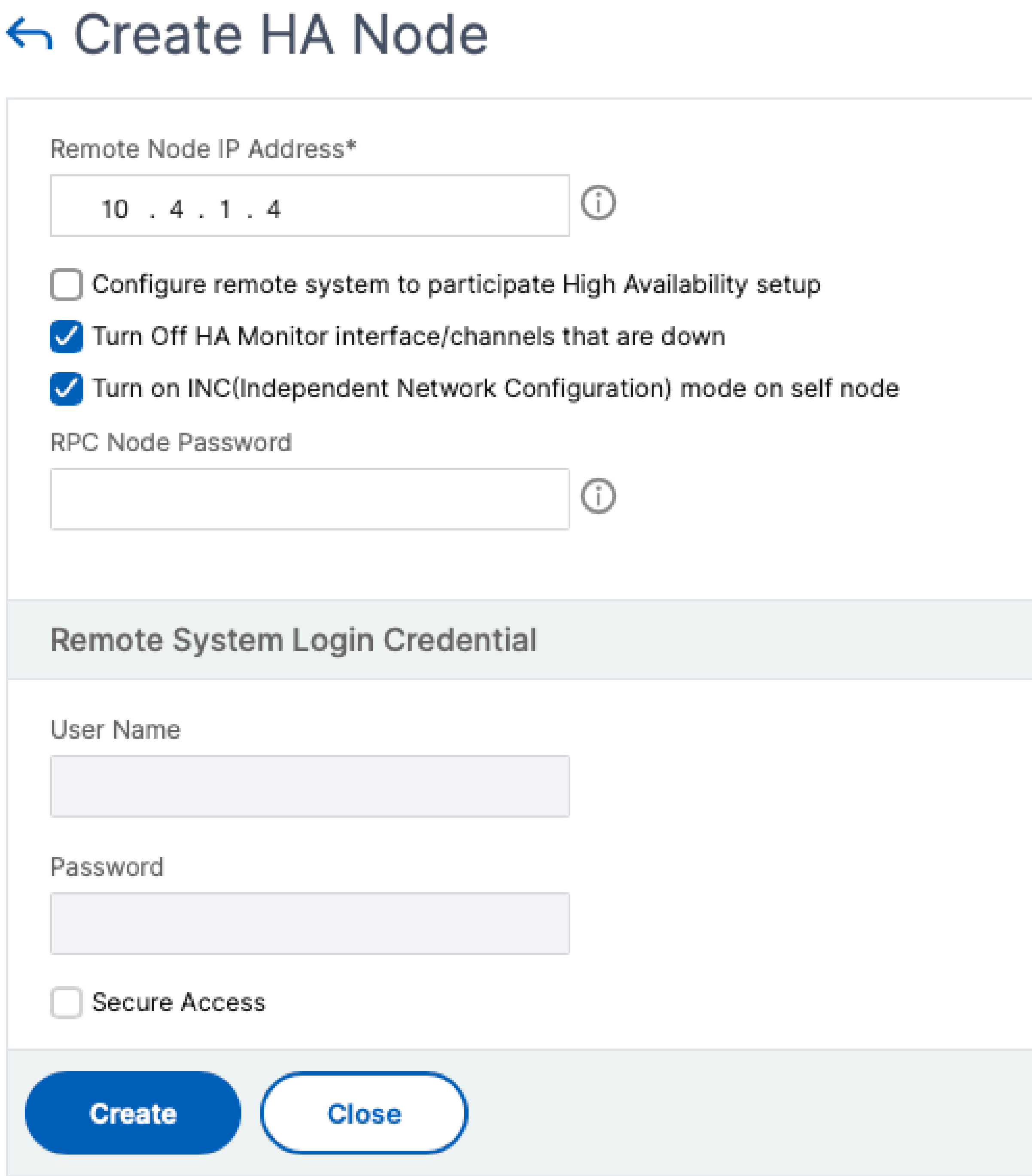
1. Log on to the instance with user name `nsroot` and password provided while deploying the instance.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the secondary instance, for example: 10.4.1.5.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.




On the secondary instance, do the following steps:

1. Log on to the instance with user name `nsroot` and password provided while deploying the instance.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the primary instance, for example: 10.4.1.4.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

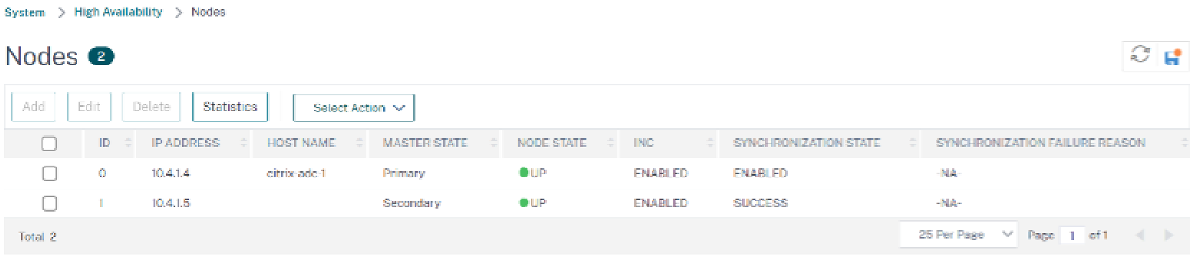
 




Before you proceed further, ensure that the **Synchronization state** of the secondary instance is shown as **SUCCESS** in the **Nodes** page.

> **Note:**
>
> Now the secondary instance has the same log-on credentials as the primary instance.




**Step 2. Add virtual IP address and subnet IP address on both the instances.**

On the primary instance, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. Add a primary VIP address by following these steps:

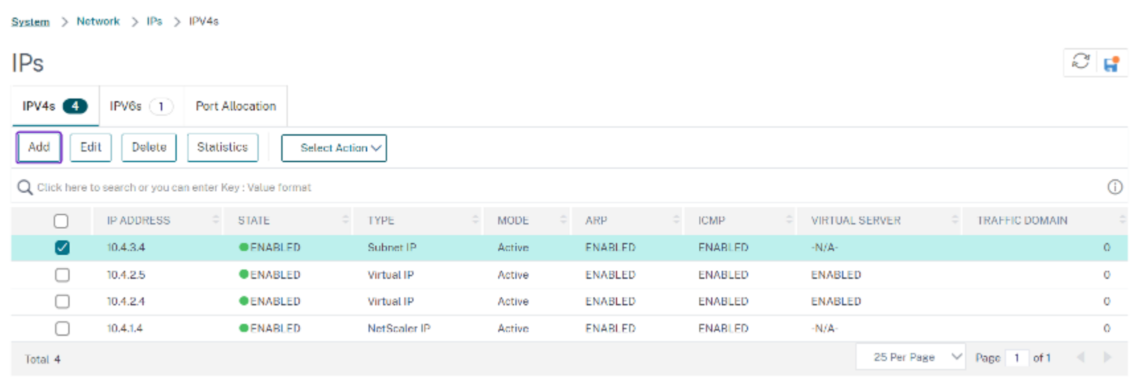
    a) Enter the private IP address of the client NIC of the primary instance and netmask configured for the client subnet in the VM instance.
    b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
    c) Click **Create**.

3. Add a primary SNIP address by following these steps:

    a) Enter the internal IP address of the server NIC of the primary instance, and netmask configured for the server subnet in the primary instance.
    b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
    c) Click **Create**.

4. Add a secondary VIP address by following these steps:

    a) Enter the internal IP address of the client NIC of the secondary instance, and netmask configured for the client subnet in the VM instance.
    b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
    c) Click **Create**.




On the secondary instance, perform the following steps:

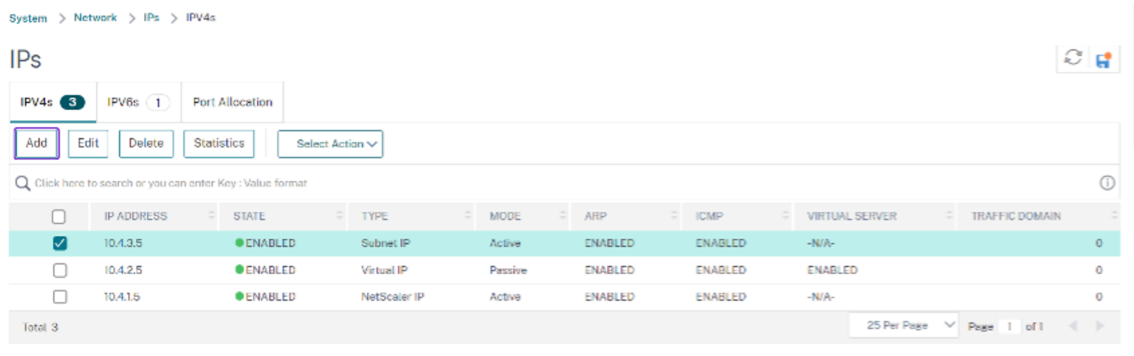
 

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. Add a secondary VIP address by following these steps:

    a) Enter the internal IP address of the client NIC of the secondary instance, and netmask configured for the client subnet in the VM instance.
    b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.

3. Add a secondary SNIP address by following these steps:

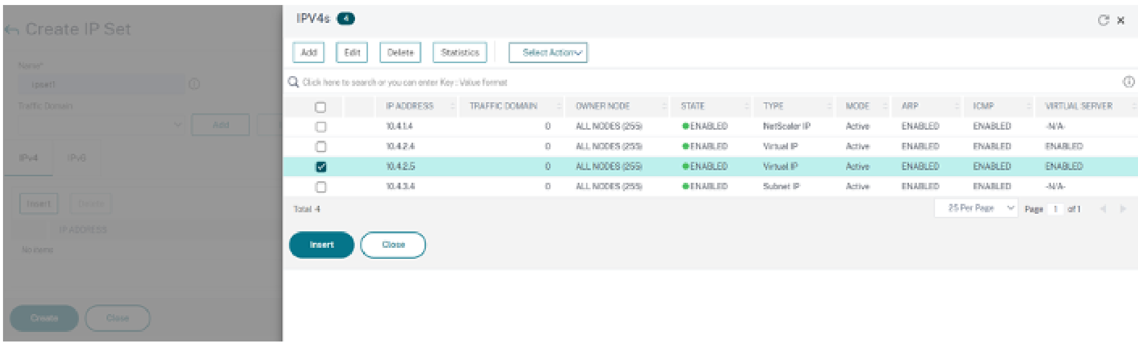
    a) Enter the internal IP address of the server NIC of the secondary instance, and netmask configured for the server subnet in the secondary instance.
    b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
    c) Click **Create**.




**Step 3. Add IP set and bind IP set to the secondary VIP on both the instances.**

On the primary instance, do the following steps:

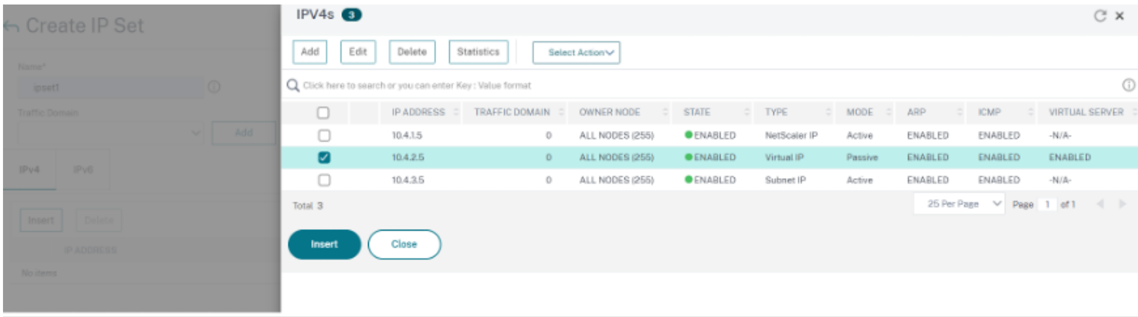
1. Navigate to **System > Network > IP Sets > Add**.
2. Add an IP set name and click **Insert**.
3. From the **IPV4s** page, select the virtual IP (secondary VIP) and click **Insert**.
4. Click **Create** to create the IP set.




On the secondary instance, do the following steps:

1. Navigate to **System > Network > IP Sets > Add**.
2. Add an IP set name and click **Insert**.
3. From the **IPv4s** page, select the virtual IP (secondary VIP) and click **Insert**.
4. Click **Create** to create the IP set.

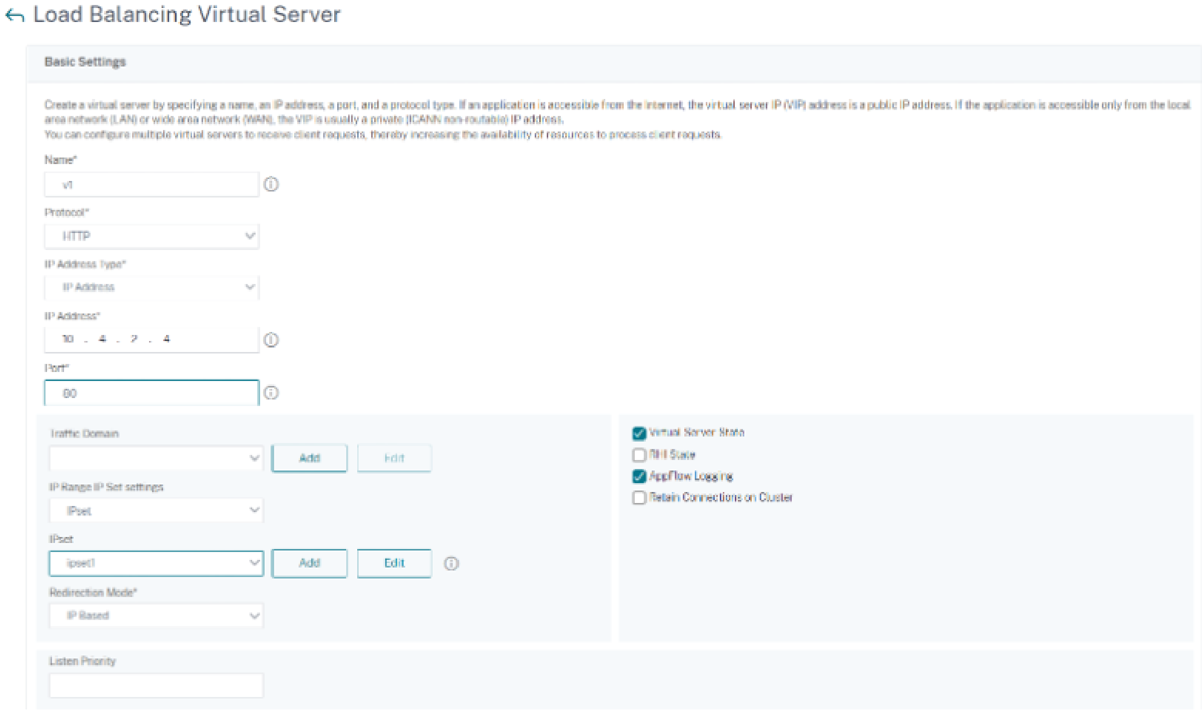



> **Note:**
>
> The IP set name must be the same on both the primary and secondary instances.

**Step 4. Add a load balancing virtual server on the primary instance.**

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.
2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP address (primary VIP), and Port.
3. Click **More**. Navigate to **IP Range IP Set Settings**, select **IPset** from the drop-down menu, and provide the IPset created in **Step 3**.
4. Click **OK** to create the load balancing virtual server.

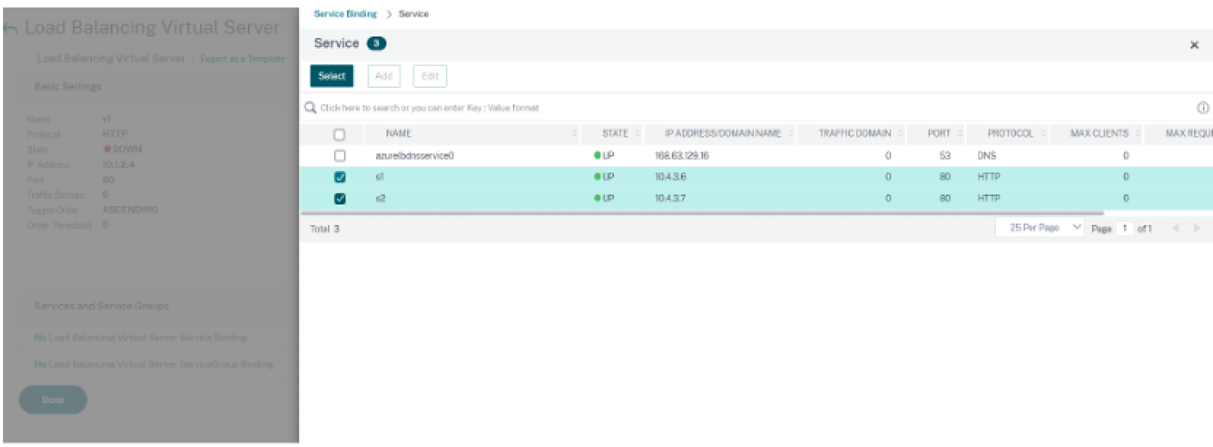



**Step 5. Add a service or service group on the primary instance.**

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

**Step 6. Bind the service or service group to the load balancing virtual server on the primary instance.**

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 4**, and click **Edit**.
3. In the **Service and Service Groups** tab, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in the **Step 5**, and click **Bind**.



**Step 7. Save the configuration.**

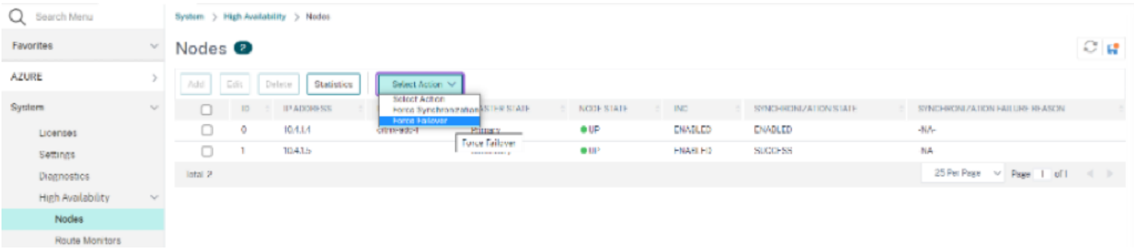
Otherwise, all the configuration is lost after a reboot or if there is an instant restart.

**Step 8. Verify the configuration.**

Make sure that the ALB frontend IP address is reachable after a failover.

1. Copy the ALB frontend IP address.
2. Paste the IP address on browser and make sure that the back-end servers are reachable.
3. On the primary instance, perform failover:

   From NetScaler GUI, navigate to **Configuration > System > High Availability > Action > Force Failover**.

4. Make sure that back-end servers are reachable after failover through ALB frontend IP used earlier.

## Configure a NetScaler VPX instance to use Azure accelerated networking

Accelerated networking enables the single root I/O virtualization (SR-IOV) virtual function (VF) NIC to a virtual machine, which improves the networking performance. You can use this feature with heavy workloads that need to send or receive data at higher throughput with reliable streaming and lower CPU utilization.
When a NIC is enabled with accelerated networking, Azure bundles the NIC's existing para virtualized (PV) interface with an SR-IOV VF interface. The support of SR-IOV VF interface enables and enhances the throughput of the NetScaler VPX instance.

Accelerated networking provides the following benefits:

- Lower latency
- Higher packets per second (pps) performance
- Enhanced throughput
- Reduced jitter
- Decreased CPU utilization

> **Note:**
>
> Azure accelerated networking is supported on NetScaler VPX instances from release 13.0 build 76.29 onwards.

### Prerequisites

- Ensure that your VM size matches the requirements for Azure accelerated networking.
- Stop VMs (individual or in an availability set) before enabling accelerated networking on any NIC.

### Limitations

Accelerated networking can be enabled only on some instance types. For more information, see Supported instance types.

### NICs supported for accelerated networking

Azure provides Mellanox ConnectX3, ConnectX4, and ConnectX5 NICs in the SR-IOV mode for accelerated networking.

When accelerated networking is enabled on a NetScaler VPX interface, Azure bundles either ConnectX3, ConnectX4, or ConnectX5 interface with the existing PV interface of a NetScaler VPX appliance.

> **Note:**
>
> NetScaler VPX supports ConnectX5 NICs from release 13.1 build 37.x onwards.

For more information about enabling accelerated networking before attaching an interface to a VM, see Create a network interface with accelerated networking.

For more information about enabling accelerated networking on an existing interface on a VM, see Enable existing interfaces on a VM.

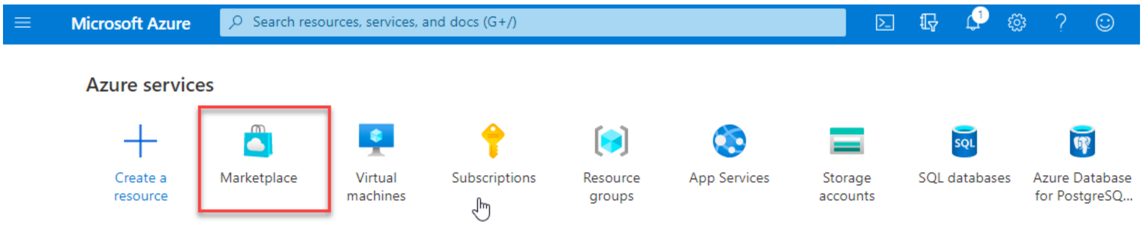
### How to enable accelerated networking on a NetScaler VPX instance using the Azure console

You can enable accelerated networking on a specific interface using the Azure console or the Azure PowerShell.

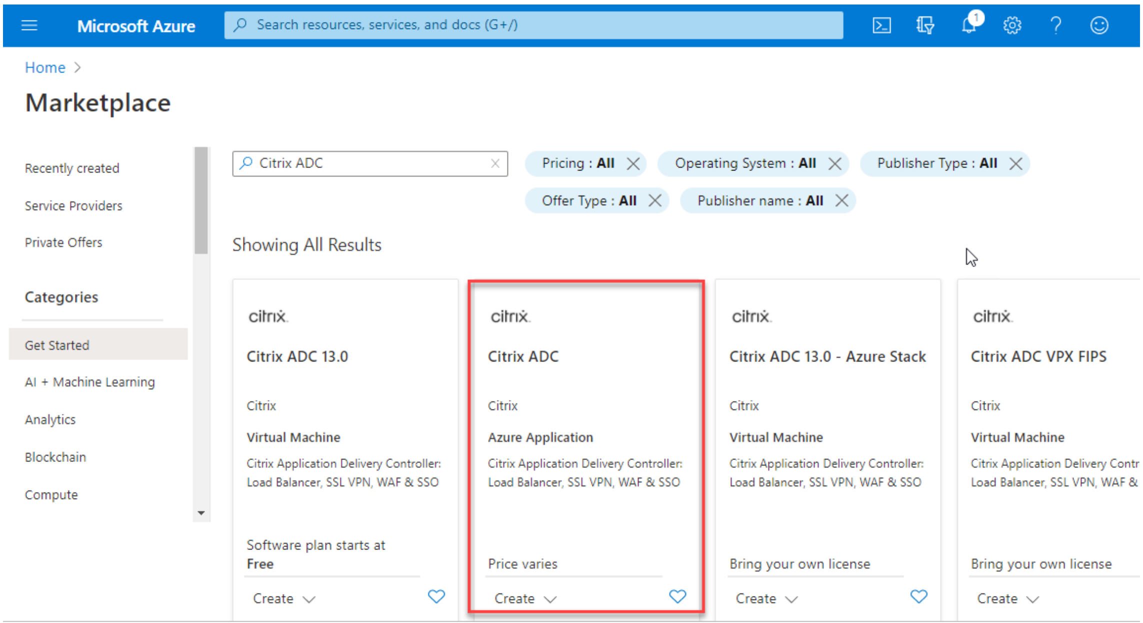
Do the following steps to enable accelerated networking by using Azure availability sets or availability zones.

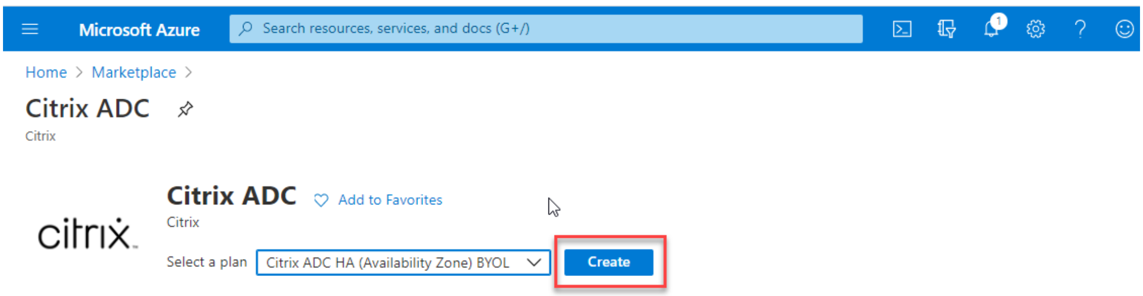
1. Log in to Azure portal, and navigate to **Azure Marketplace**.




2. From the **Azure Marketplace**, search **NetScaler**.



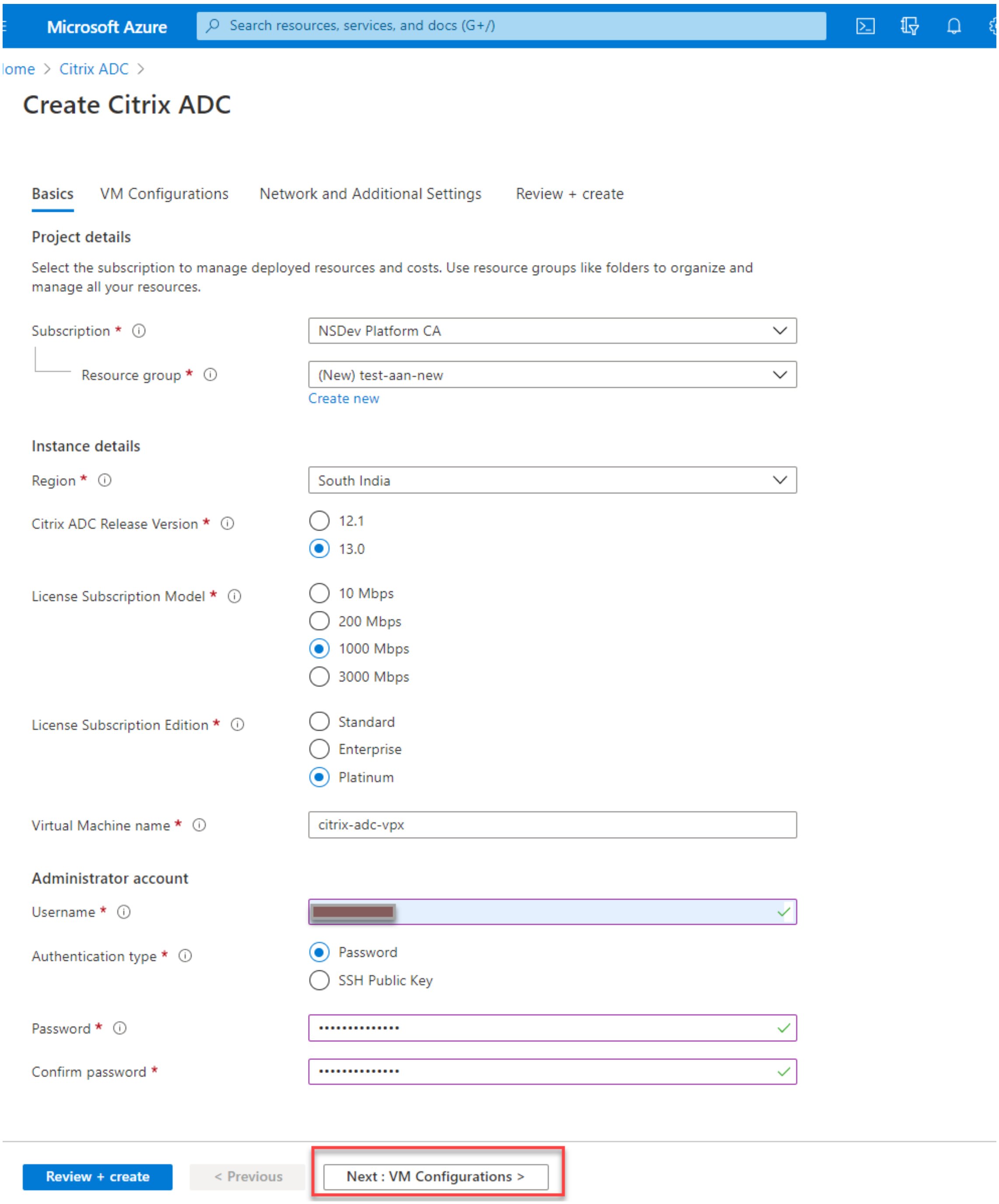

3. Select a non-FIPS NetScaler plan along with license, and click **Create**.




   The **Create NetScaler** page appears.

4. In the **Basics** tab, create a Resource Group. Under the **Parameters** tab, enter details for the Region, Admin user name, Admin Password, license type (VM SKU), and other fields.

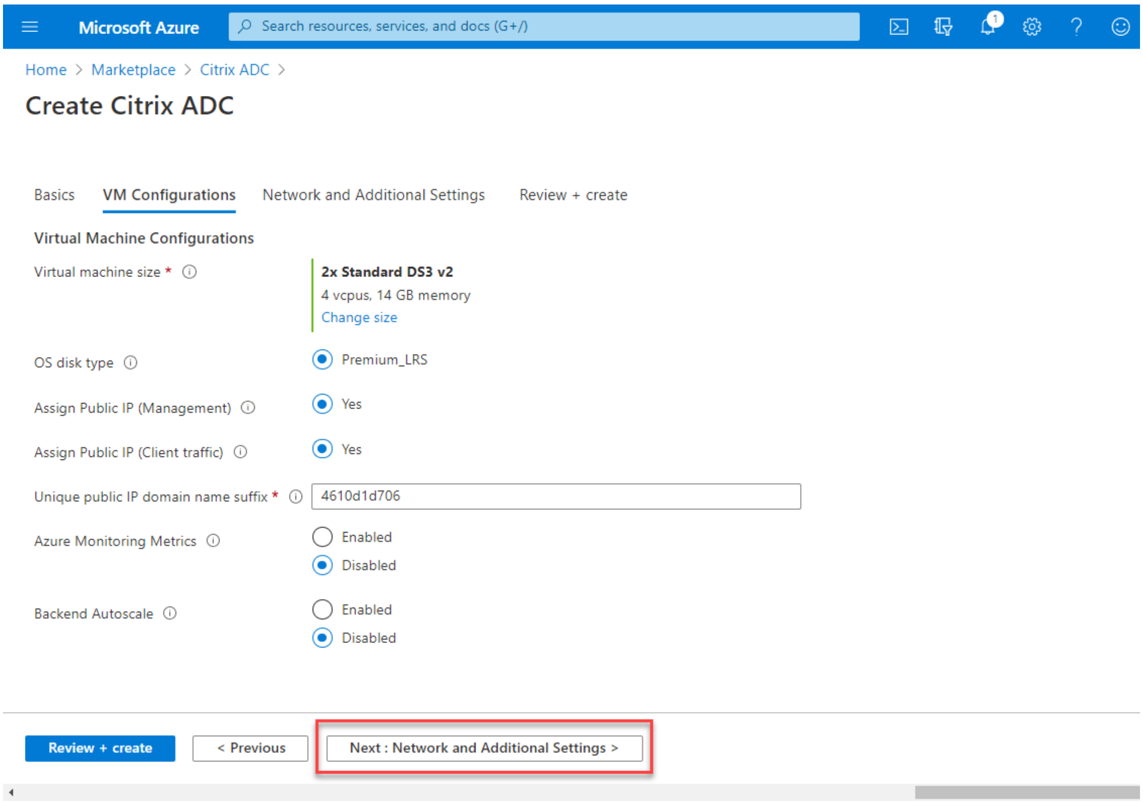



5. Click **Next : VM Configurations >**.

   On the **VM Configurations** page, perform the following:

   a) Configure public IP domain name suffix.
   b) Enable or disable **Azure Monitoring Metrics**.
   c) Enable or disable **Backend Autoscale**.

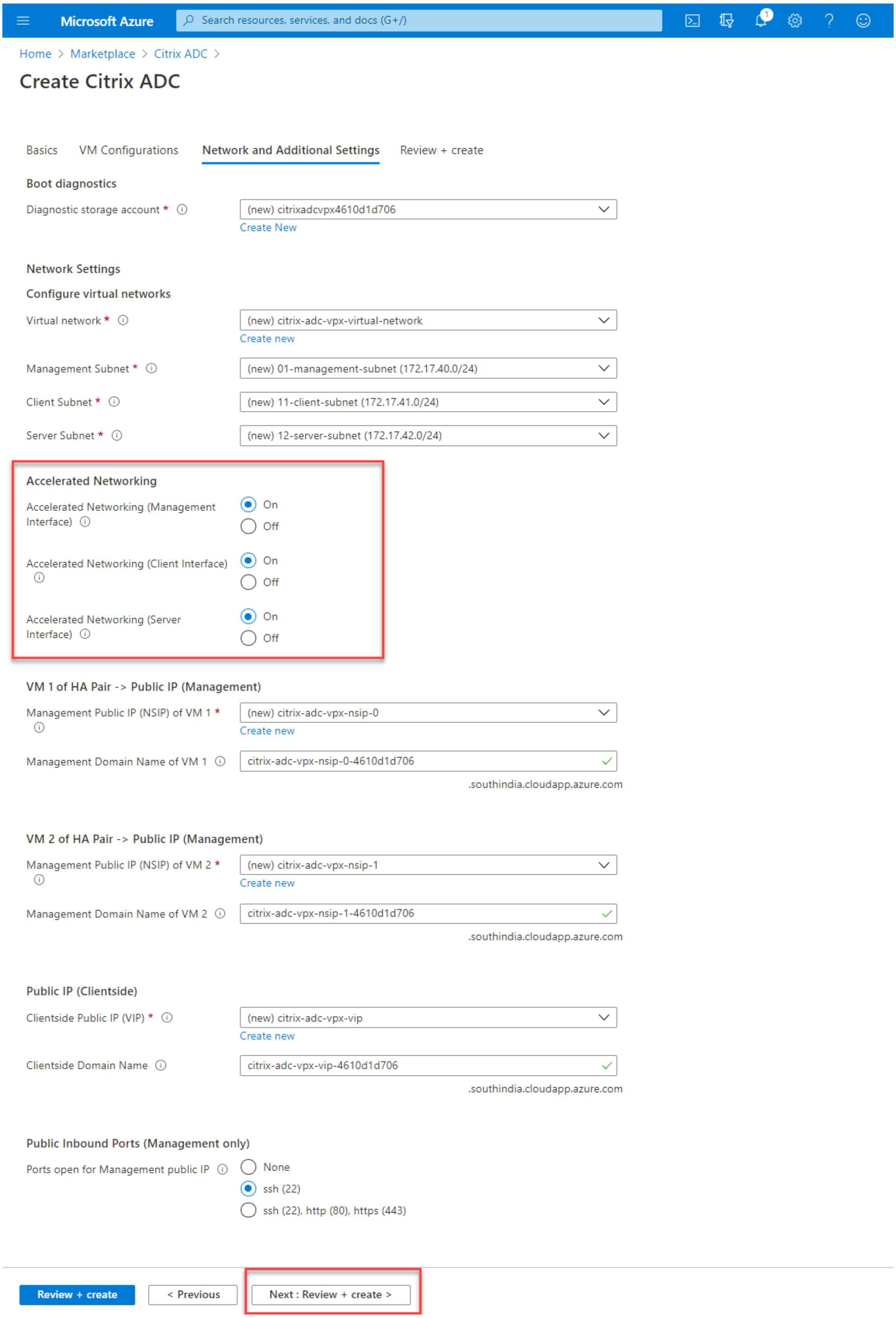



6. Click **Next: Network and Additional settings >**.

   On the **Network and Additional Settings** page, create a Boot diagnostics account and configure the network settings.

   Under the **Accelerated Networking** section, you have the option to enable or disable the accelerated networking separately for the Management interface, Client interface, and Server interface.

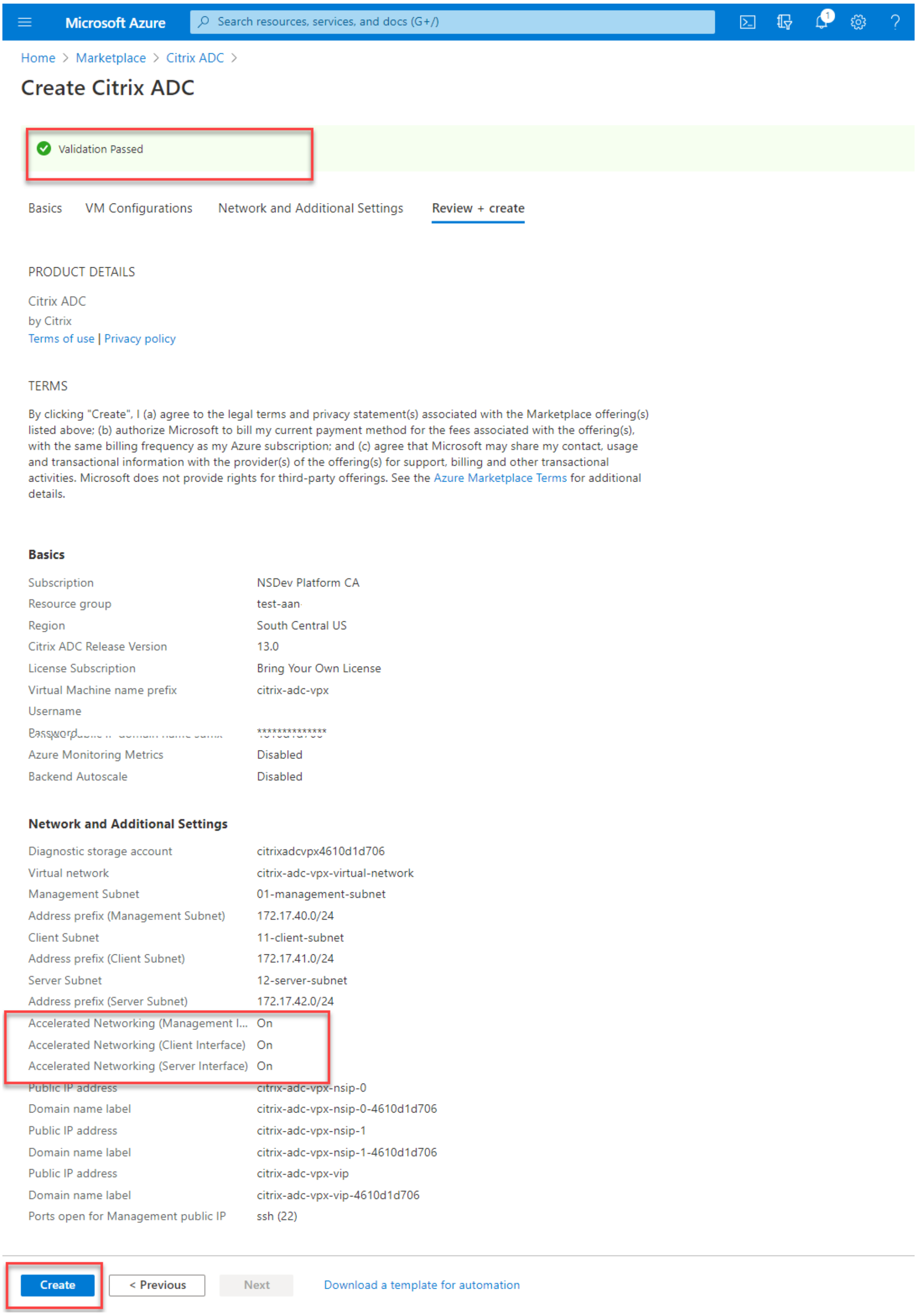
 


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Marketplace > Citrix ADC >
Create Citrix ADC
Basics
VM Configurations
Network and Additional Settings
Review + create
Boot diagnostics
Diagnostic storage account *
(new) citrixadcvpx4610d1d706
Create New
Network Settings
Configure virtual networks
Virtual network *
(new) citrix-adc-vpx-virtual-network
Create new
Management Subnet *
(new) 01-management-subnet (172.17.40.0/24)
Client Subnet *
(new) 11-client-subnet (172.17.41.0/24)
Server Subnet *
(new) 12-server-subnet (172.17.42.0/24)
Accelerated Networking
Accelerated Networking (Management Interface)
On
Off
Accelerated Networking (Client Interface)
On
Off
Accelerated Networking (Server Interface)
On
Off
VM 1 of HA Pair -> Public IP (Management)
Management Public IP (NSIP) of VM 1 *
(new) citrix-adc-vpx-nsip-0
Create new
Management Domain Name of VM 1
citrix-adc-vpx-nsip-0-4610d1d706
.southindia.cloudapp.azure.com
VM 2 of HA Pair -> Public IP (Management)
Management Public IP (NSIP) of VM 2 *
(new) citrix-adc-vpx-nsip-1
Create new
Management Domain Name of VM 2
citrix-adc-vpx-nsip-1-4610d1d706
.southindia.cloudapp.azure.com
Public IP (Clientside)
Clientside Public IP (VIP) *
(new) citrix-adc-vpx-vip
Create new
Clientside Domain Name
citrix-adc-vpx-vip-4610d1d706
.southindia.cloudapp.azure.com
Public Inbound Ports (Management only)
Ports open for Management public IP
None
ssh (22)
ssh (22), http (80), https (443)
Review + create
< Previous
Next : Review + create >

7. Click **Next: Review + create >**.

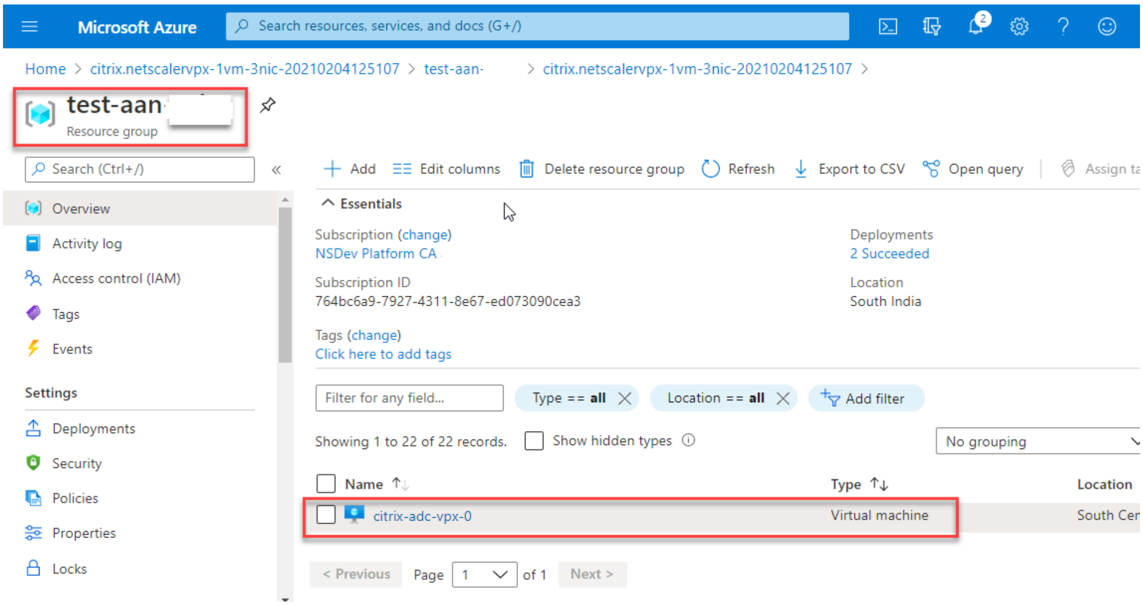
   After the validation is successful, review the basic settings, VM configurations, network and additional settings, and click **Create**. It might take some time for the Azure Resource Group to be created with the required configurations.

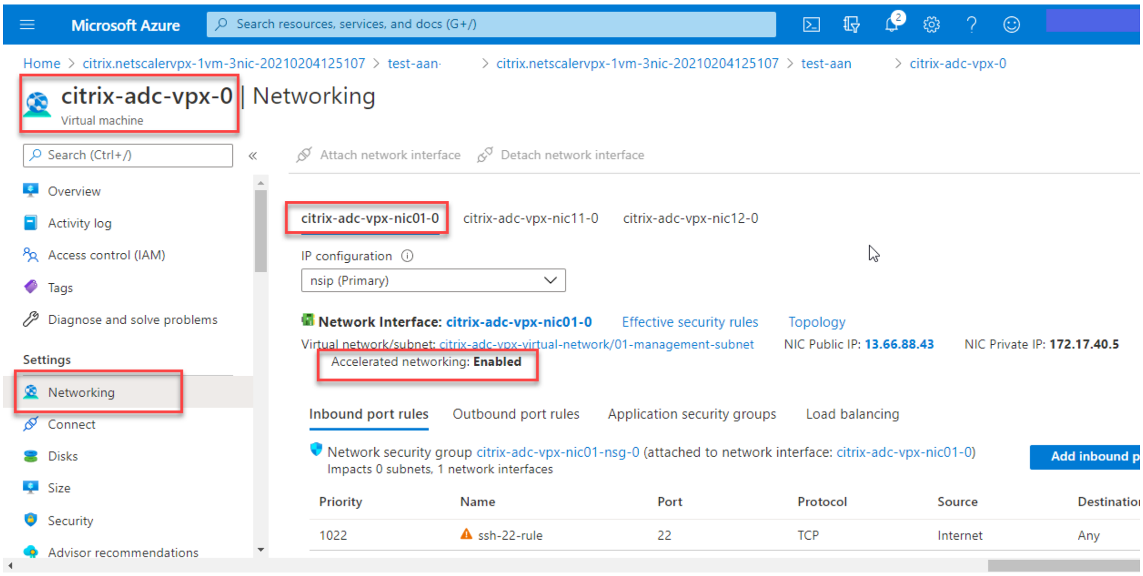



8. After the deployment is complete, select the **Resource Group** to see the configuration details.




9. To verify the Accelerated Networking configurations, select **Virtual machine > Networking**. The Accelerated Networking status is displayed as **Enabled** or **Disabled** for each NIC.




## Enable accelerated networking using Azure PowerShell

If you need to enable accelerated networking after the VM creation, you can do so using Azure PowerShell.

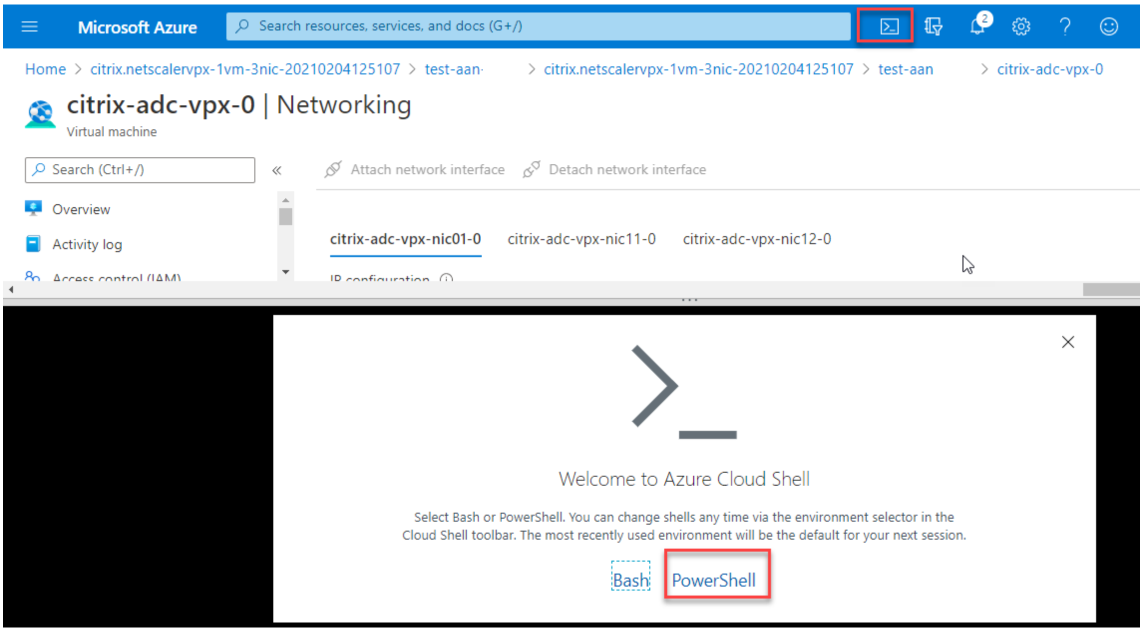
> **Note:**
>
> Ensure to stop the VM before you enable Accelerated Networking using Azure PowerShell.

Perform the following steps to enable accelerated networking by using Azure PowerShell.

1. Navigate to **Azure portal**, click the **PowerShell** icon on the right-hand top corner.

   > **Note:**
   >
   > If you are in the Bash mode, change to the PowerShell mode.




2. At the command prompt, run the following command:

```
az network nic update --name <nic-name> --accelerated-networking [
    true | false] --resource-group <resourcegroup-name>
```

   The accelerated networking parameter accepts either of the following values:

   - **True:** Enables accelerated networking on the specified NIC.
   - **False:** Disables accelerated networking on the specified NIC.

   **To enable accelerated networking on a specific NIC:**

```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-
    networking true --resource-group rsgp1-aan
```

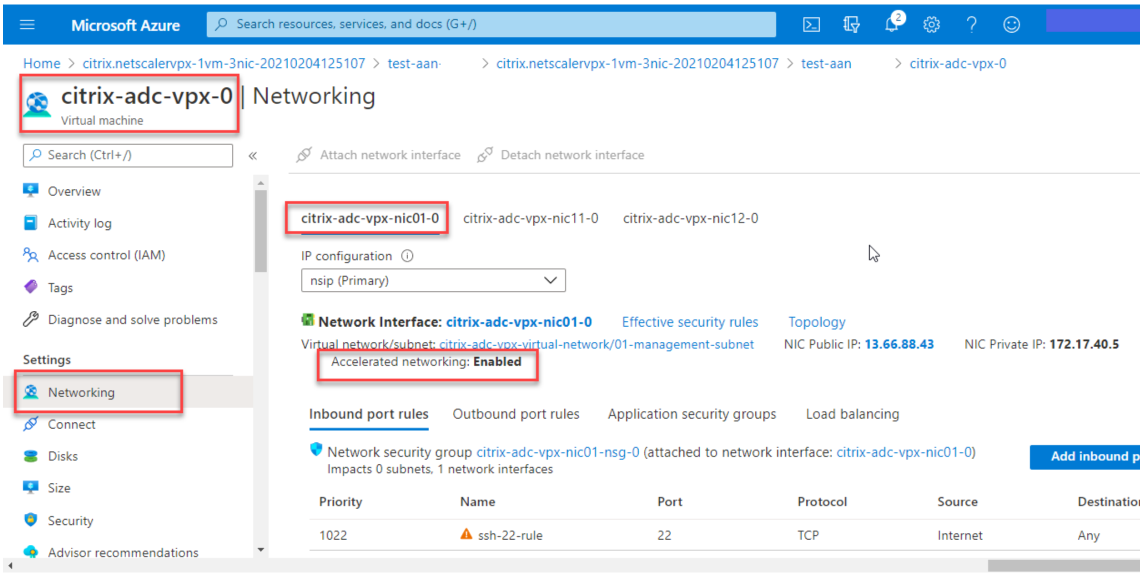
   **To disable accelerated networking on a specific NIC:**

```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-
    networking false --resource-group rsgp1-aan
```

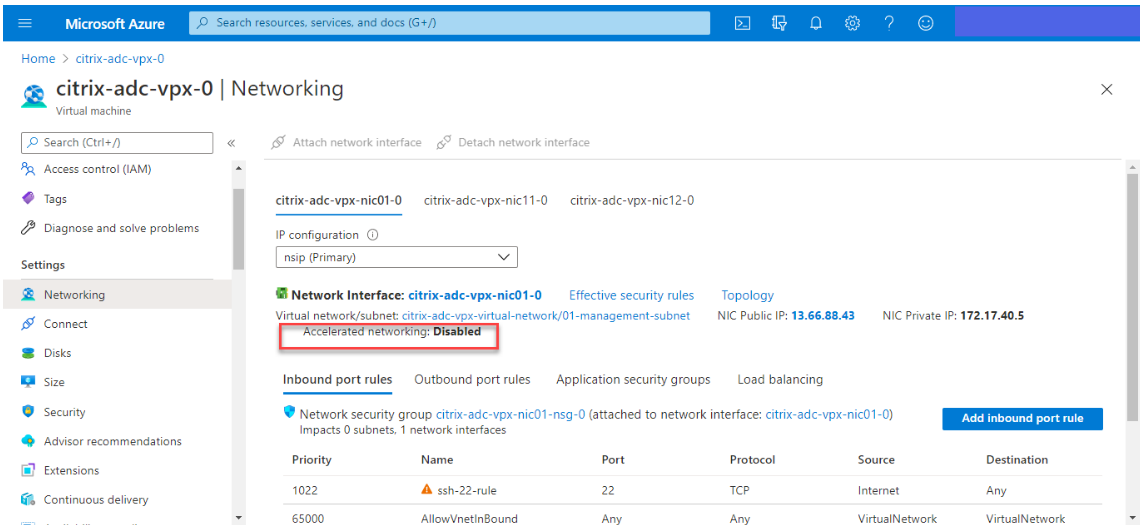
3. To verify that the Accelerated Networking status after the deployment is completed, Navigate to **VM > Networking**.

   In the following example, you can see that Accelerated Networking is **Enabled**.




In the following example, you can see that Accelerated Networking is **Disabled**.




### To verify accelerated networking on an interface by using FreeBSD Shell of NetScaler

You can log in to FreeBSD shell of NetScaler, and run the following commands to verify the accelerated networking status.

**Example for ConnectX3 NIC:**

The following example shows the "ifconfig"command output of the Mellanox ConnectX3 NIC. The "50/n"indicates the VF interfaces of the Mellanox ConnectX3 NICs. 0/1 and 1/1 indicates the PV interfaces of the NetScaler VPX instance. You can observe that both PV interface (1/1) and CX3 VF interface (50/1) have the same MAC addresses (00:22:48:1c:99:3e). This indicates that the two interfaces are bundled together.

```
root@nvr-us-cx3# ifconfig

lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 1500

        options=3<RXCSUM,TXCSUM>

        inet 127.0.0.1 netmask 0xff000000

        inet6 ::1 prefixlen 128

        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1

        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>

0/1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500

        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>

        ether 00:0d:3a:98:71:be

        inet 172.16.27.11 netmask 0xffffff00 broadcast 172.16.27.255

        inet6 fe80::20d:3aff:fe98:71be%0/1 prefixlen 64 autoconf scopeid 0x2

        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>

        media: Ethernet autoselect (10Gbase-T <full-duplex>)

        status: active

1/1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500

        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>

        ether 00:22:48:1c:99:3e

        media: Ethernet autoselect (10Gbase-T <full-duplex>)

        status: active

50/1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500

options=900b8<VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,VLAN_HWFILTER,LINKSTATE>

        ether 00:22:48:1c:99:3e

        media: Ethernet autoselect (<unknown subtype>)

        status: active
```

**Example for ConnectX4 NIC:**

The following example shows the "ifconfig"command output of the Mellanox ConnectX4 NIC. The "100/n"indicates the VF interfaces of the Mellanox ConnectX4 NICs. 0/1, 1/1, and 1/2 indicates the PV interfaces of NetScaler VPX instance.

You can observe that both PV interface (1/1) and CX4 VF interface (100/1) have the same MAC addresses (00:0d:3a:9b:f2:1d). This indicates that the two interfaces are bundled together. Similarly, the PV interface (1/2) and CX4 VF interface (100/2) have the same MAC addresses (00:0d:3a:1e:d2:23).

```
root@SmartNIC-CX4-NS-DUT-NEW1# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
1/1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>

        ether 00:0d:3a:9b:f2:1d

        inet 10.0.1.29 netmask 0xffffff00 broadcast 10.0.1.255
        inet6 fe80::20d:3aff:fe9b:f21d%0/1 prefixlen 64 autoconf scopeid 0x2
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active

1/2: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>

        ether 00:0d:3a:1e:d2:23

        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
100/1: flags=8a03<UP,BROADCAST,ALLMULTI,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 00:0d:3a:9b:f2:1d

        media: Ethernet autoselect <full-duplex,rxpause,txpause> (autoselect
<full-duplex,rxpause>)
        status: active
100/2: flags=8a03<UP,BROADCAST,ALLMULTI,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 00:0d:3a:1e:d2:23

        media: Ethernet autoselect <full-duplex,rxpause,txpause> (autoselect
<full-duplex,rxpause>)
        status: active
```

### To verify accelerated networking on an interface by using ADC CLI

**Example for ConnectX3 NIC:**

The following show interface command output indicates that the PV interface 1/1 is bundled with virtual function 50/1, which is an SR-IOV VF NIC. The MAC addresses of both 1/1 and 50/1 NICs are the same. After accelerated networking is enabled, the data of the 1/1 interface is sent through datapath of the 50/1 interface, which is a ConnectX3 interface. You can see that the "show interface"output of the PV interface (1/1) points to the VF (50/1). Similarly, the "show interface"output of VF interface (50/1) points to the PV interface (1/1).

```
> show interface 1/1

        Interface 1/1 (NetScaler Virtual Interface, SmartNIC, VF 50/1  Datapath 50/1) #1
        flags=0xe060 <ENABLED, UP, UP, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m07s
        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.

Done

> show interface 50/1

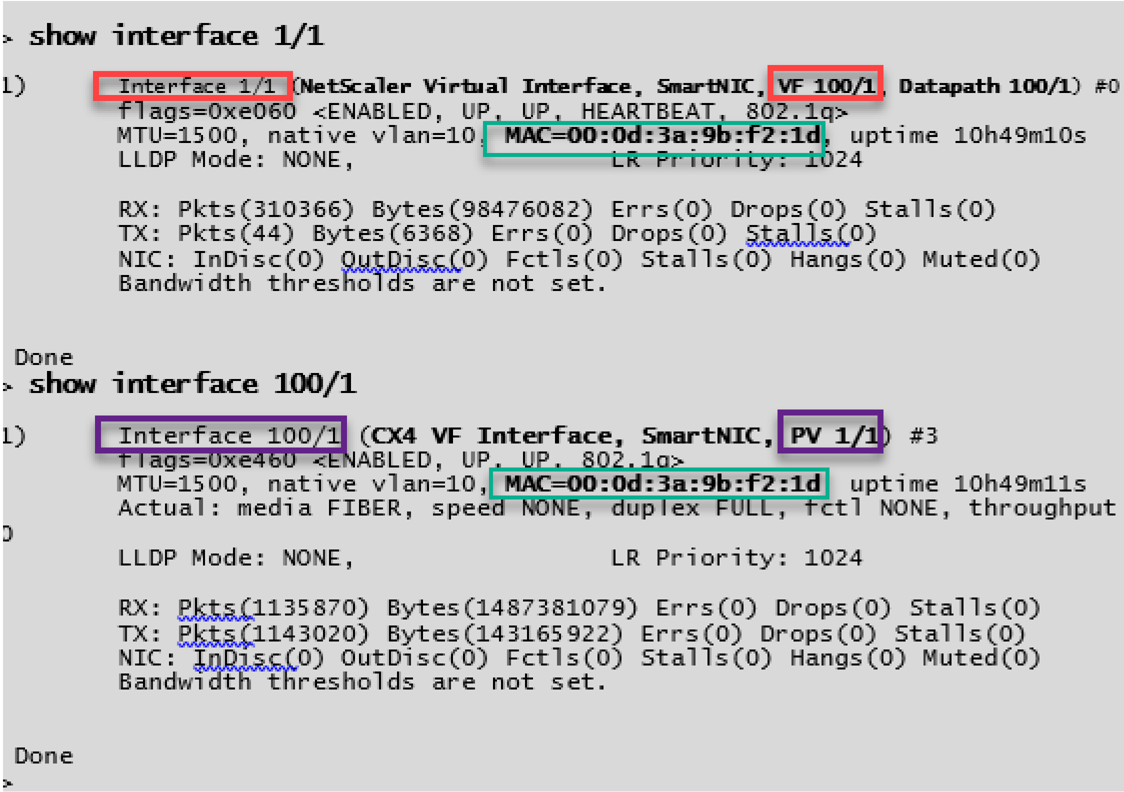
       Interface 50/1 (CX3 VF Interface, SmartNIC, PV 1/1) #2
        flags=0xe460 <ENABLED, UP, UP, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m08s
        Actual: media NONE, speed 50000, duplex FULL, fctl NONE, throughput 50000
        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
```

**Example for ConnectX4 NIC:**

The following show interface command output indicates that the PV interface 1/1 is bundled with virtual function 100/1, which is an SR-IOV VF NIC. The MAC addresses of both 1/1 and 100/1 NICs are the same. After accelerated networking is enabled, the data of 1/1 interface is sent through the data path of 100/1 interface, which is a ConnectX4 interface. You can see that the "show interface" output of PV interface (1/1) points to the VF (100/1). Similarly, the "show interface" output of VF interface (100/1) points to the PV interface (1/1).



```
> show interface 1/1
1)      Interface 1/1 (NetScaler Virtual Interface, SmartNIC, VF 100/1, Datapath 100/1) #0
        flags=0xe060 <ENABLED, UP, UP, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=10, MAC=00:0d:3a:9b:f2:1d, uptime 10h49m10s
        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(310366) Bytes(98476082) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(44) Bytes(6368) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.

 Done
> show interface 100/1
1)      Interface 100/1 (CX4 VF Interface, SmartNIC, PV 1/1) #3
        flags=0xe460 <ENABLED, UP, UP, 802.1q>
        MTU=1500, native vlan=10, MAC=00:0d:3a:9b:f2:1d, uptime 10h49m11s
        Actual: media FIBER, speed NONE, duplex FULL, fctl NONE, throughput
0

        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(1135870) Bytes(1487381079) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(1143020) Bytes(143165922) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.

 Done
>
```

### Points to note in NetScaler

- PV interface is considered as the primary or main interface for all the necessary operations. Configurations must be performed on PV interfaces only.
- All the 'set'operations on a VF interface are blocked except the following:
  - enable interface
  - disable interface
  - reset interface
  - clear stats

  > **Note:**
  >
  > Citrix recommends that you do not perform any operations on the VF interface.

- You can verify the binding of PV interface with VF interface using the `show interface` command.
- From NetScaler release 13.1-33.x, a NetScaler VPX instance can seamlessly handle dynamic NIC removals and reattachment of the removed NICs in Azure accelerated networking. Azure can remove SR-IOV VF NIC of accelerated networking for their host maintenance activities. Whenever a NIC is removed from Azure VM, the NetScaler VPX instance shows the interface status as

"Link Down"and the traffic goes through the virtual interface only. After the removed NIC is reattached, the VPX instances use the reattached SR-IOV VF NIC. This process happens seamlessly and does not require any configuration.

### Configure a VLAN to a PV interface

When a PV interface is bound to a VLAN, the associated accelerated VF interface is also bound to the same VLAN as the PV interface. In this example, the PV interface (1/1) is bound to VLAN (20). The VF interface (100/1) that is bundled with the PV interface (1/1) is also bound to VLAN 20.

**Example:**

1. Create a VLAN.

```
add vlan 20
```

2. Bind a VLAN to the PV interface.

```
bind vlan 20 – ifnum 1/1

show vlan

1)  VLAN ID: 1
    Link-local IPv6 addr: fe80::20d:3aff:fe9b:f21d/64
    Interfaces : LO/1

2)  VLAN ID: 10      VLAN Alias Name:
    Interfaces : 0/1 100/1
    IPs : 10.0.1.29  Mask: 255.255.255.0

3)  VLAN ID: 20      VLAN Alias Name:
    Interfaces :  1/1 100/2
```

> **Note:**
>
> VLAN binding operation is not permitted on an accelerated VF interface.

```
bind vlan 1 -ifnum 100/1
ERROR: Operation not permitted
```

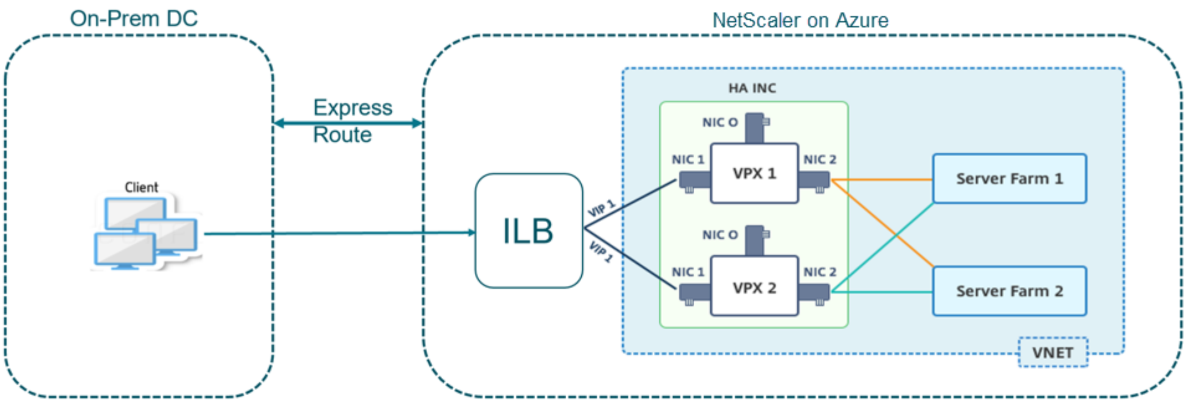
## Configure HA-INC nodes by using the NetScaler high availability template with Azure ILB

You can quickly and efficiently deploy a pair of VPX instances in HA-INC mode by using the standard template for intranet applications. The Azure internal load balancer (ILB) uses an internal or private

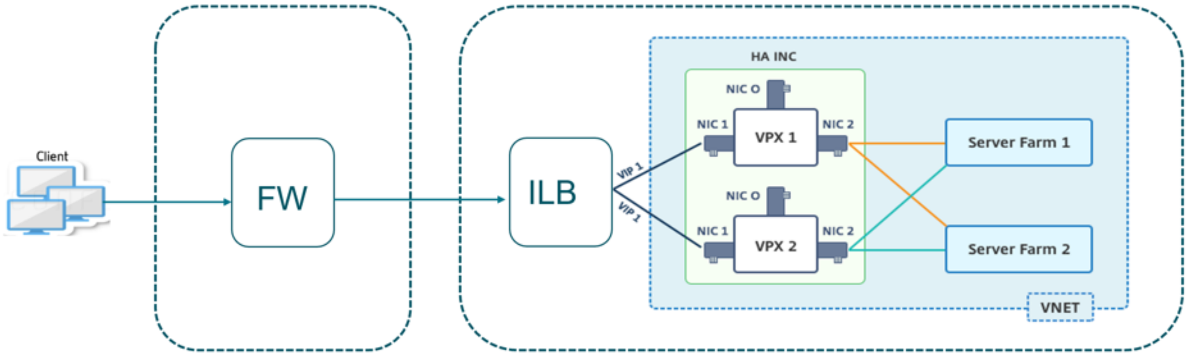
IP address for the front end as shown in Figure 1. The template creates two nodes, with three subnets and six NICs. The subnets are for management, client, and server-side traffic with each subnet belonging to a different NIC on each device.

Figure 1: NetScaler HA pair for clients in an internal network




You can also use this deployment when the NetScaler HA pair is behind a firewall as shown in Figure 2. The public IP address belongs to the firewall and is NAT'd to the front-end IP address of the ILB.

Figure 2: NetScaler HA pair with firewall having public IP address

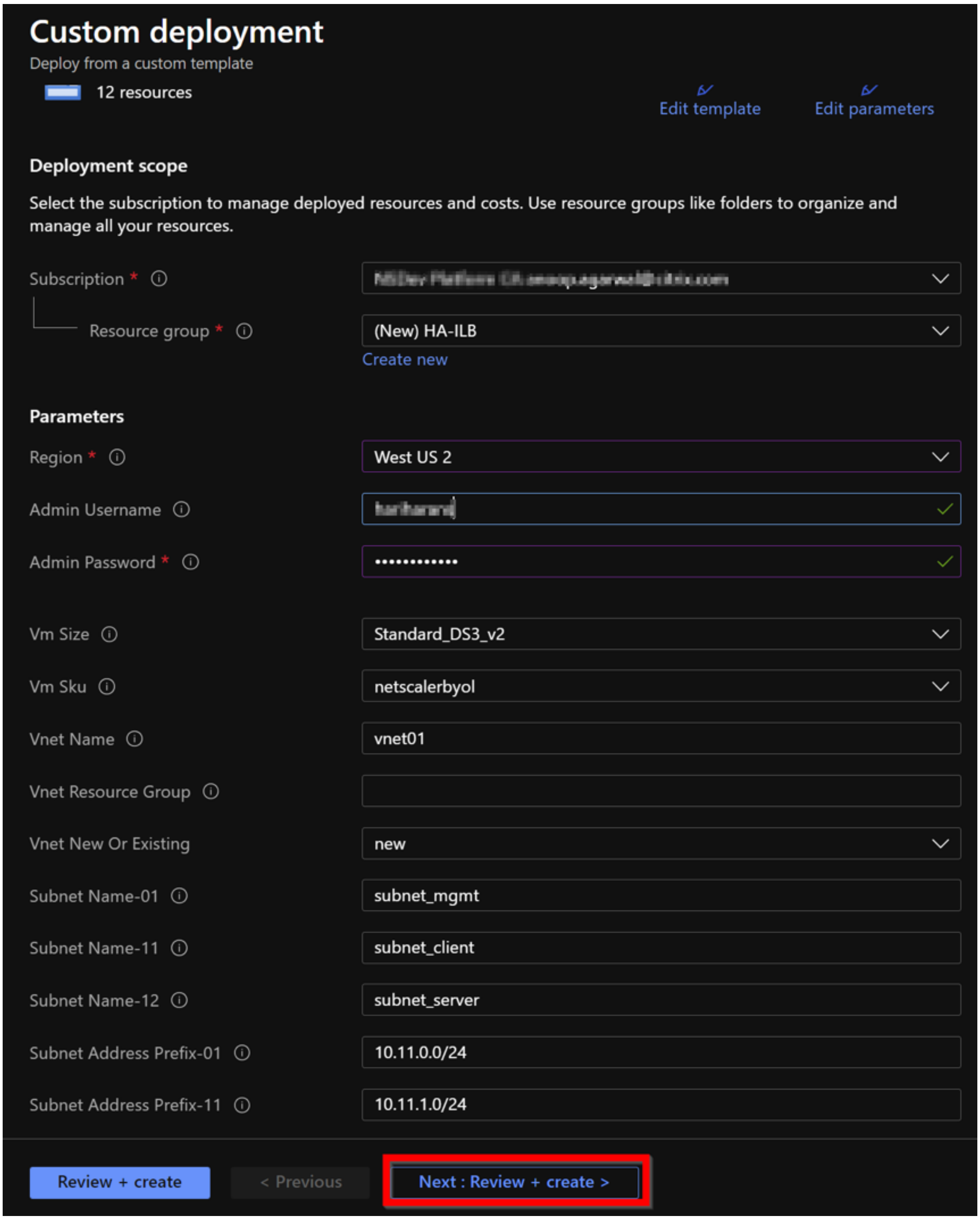



You can get the NetScaler HA pair template for intranet applications at the Azure portal

Complete the following steps to launch the template and deploy a high availability VPX pair by using Azure Availability Sets.

1. From the Azure portal, navigate to the **Custom deployment** page.
2. The **Basics** page appears. Create a Resource Group. Under the **Parameters** tab, enter details for the Region, Admin user name, Admin Password, license type (`VM sku`), and other fields.



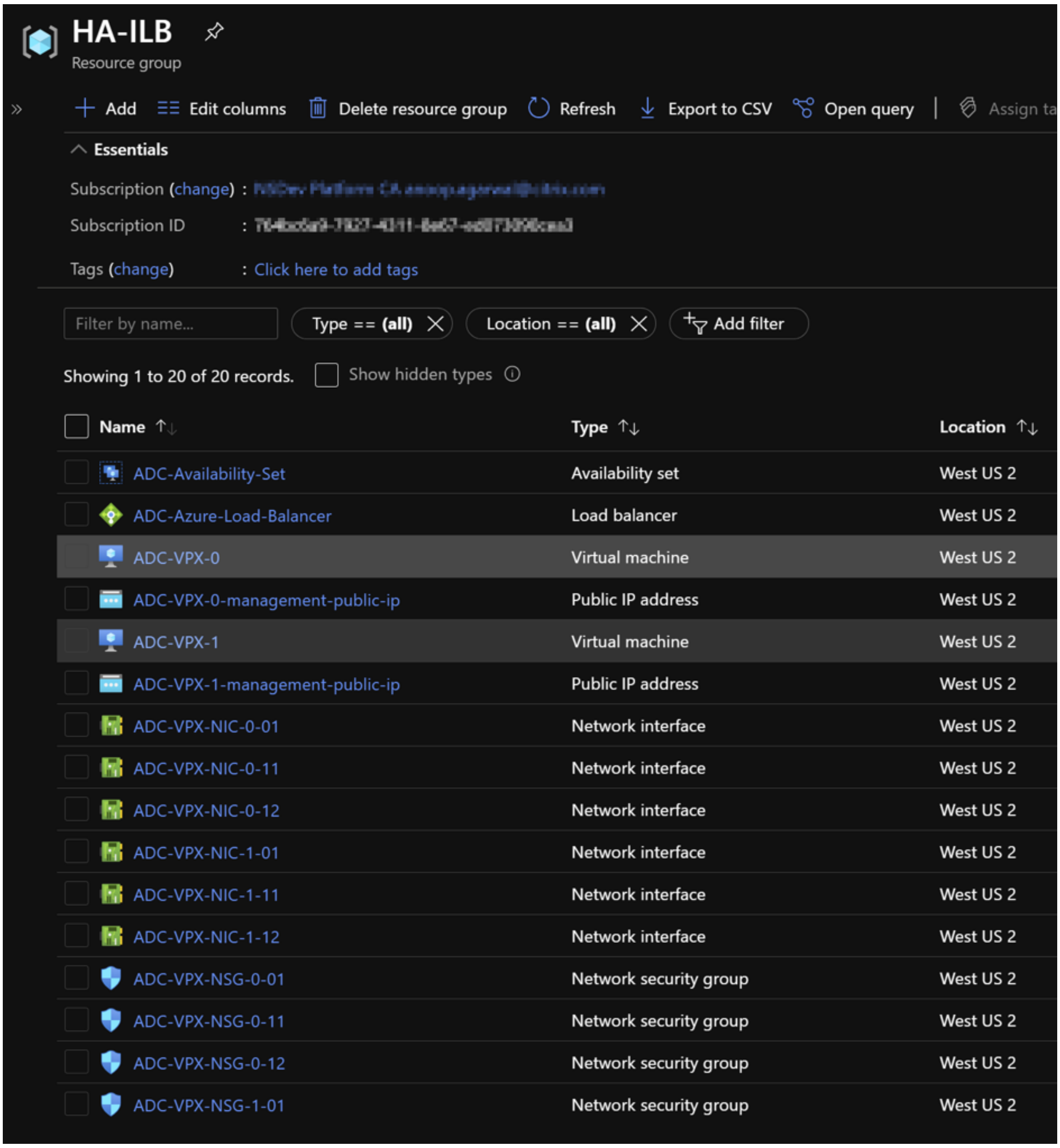

3. Click **Next : Review + create >**.

   It might take a moment for the Azure Resource Group to be created with the required configurations. After completion, select the Resource Group in the Azure portal to see the configuration details, such as LB rules, back-end pools, health probes. The high availability pair appears as ADC-VPX-0 and ADC-VPX-1.

   If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.

Once the required configuration is complete, the following resources are created.




4. Log on to **ADC-VPX-0** and **ADC-VPX-1** nodes to validate the following configuration:

   - NSIP addresses for both nodes must be in the management subnet.
   - On the primary (ADC-VPX-0) and secondary (ADC-VPX-1) nodes, you must see two SNIP addresses. One SNIP (client subnet) is used for responding to ILB probes and the other SNIP (server subnet) is used for back-end server communication.

> **Note:**
>
> In the HA-INC mode, the SNIP address of the ADC-VPX-0 and ADC-VPX-1 VMs are different

> while in the same subnet, unlike with the classic on-premises ADC HA deployment where both are the same.
> To support deployments when the VPX pair SNIP is in different subnets, or anytime the VIP is not in the same subnet as a SNIP, you must either enable Mac-Based Forwarding (MBF), or add a static host route for each VIP to each VPX node.

On the primary node (ADC-VPX-0)

```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.5        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.4        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

```
> sh ha node
1)      Node ID:      0
        IP:        10.11.0.5 (ADC-VPX-0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:20:26 (days:hrs:min:sec)
2)      Node ID:      1
        IP:        10.11.0.4
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

On the secondary node (ADC-VPX-1)

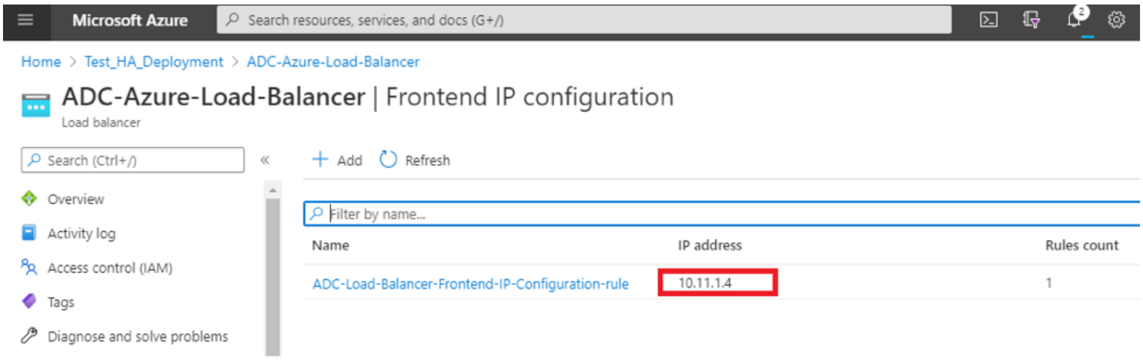
```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.4        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.6        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.4 (ADC-VPX-1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:24:18 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.5
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

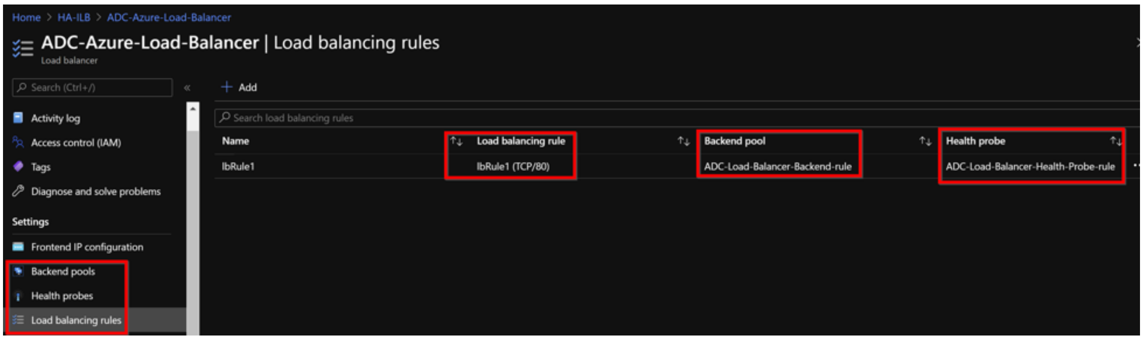
5. After the primary and secondary nodes are UP and the Synchronization status is **SUCCESS**, you must configure the load balancing virtual server or the gateway virtual server on the primary node (ADC-VPX-0) with the private floating IP (FIP) address of the ADC Azure load balancer. For more information, see the Sample configuration section.

6. To find the private IP address of ADC Azure load balancer, navigate to **Azure portal > ADC Azure Load Balancer > Frontend IP configuration**.

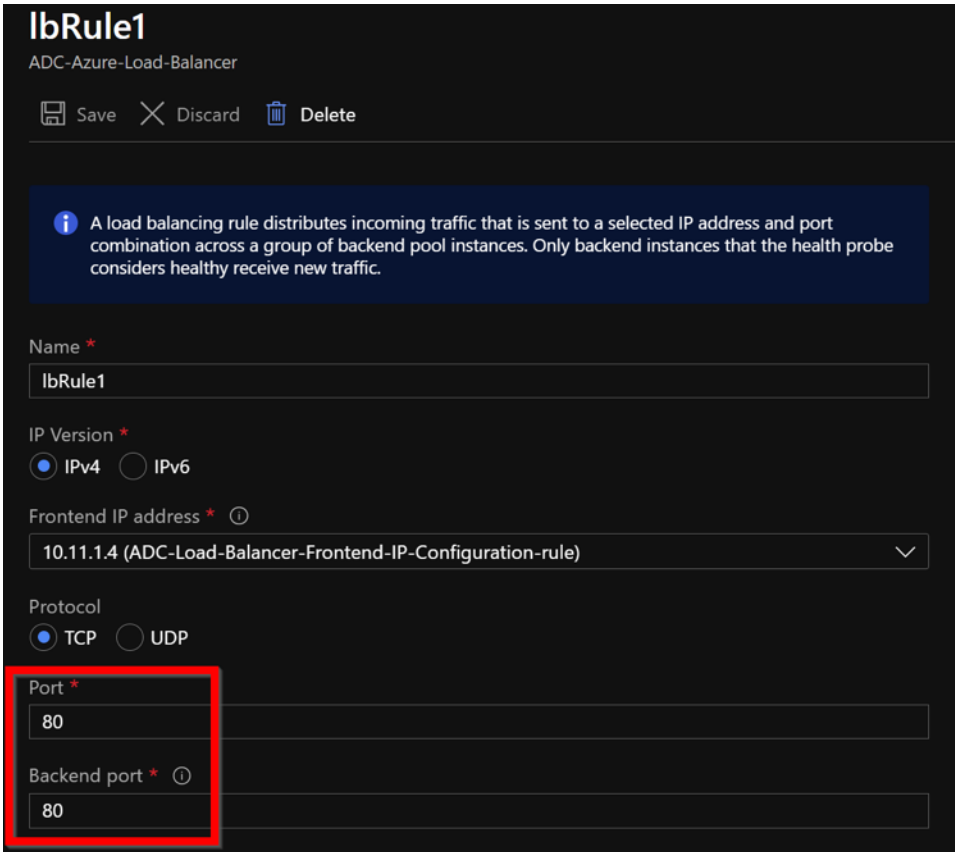
 




7. In the **Azure Load Balancer** configuration page, the ARM template deployment helps create the LB rule, back-end pools, and health probes.



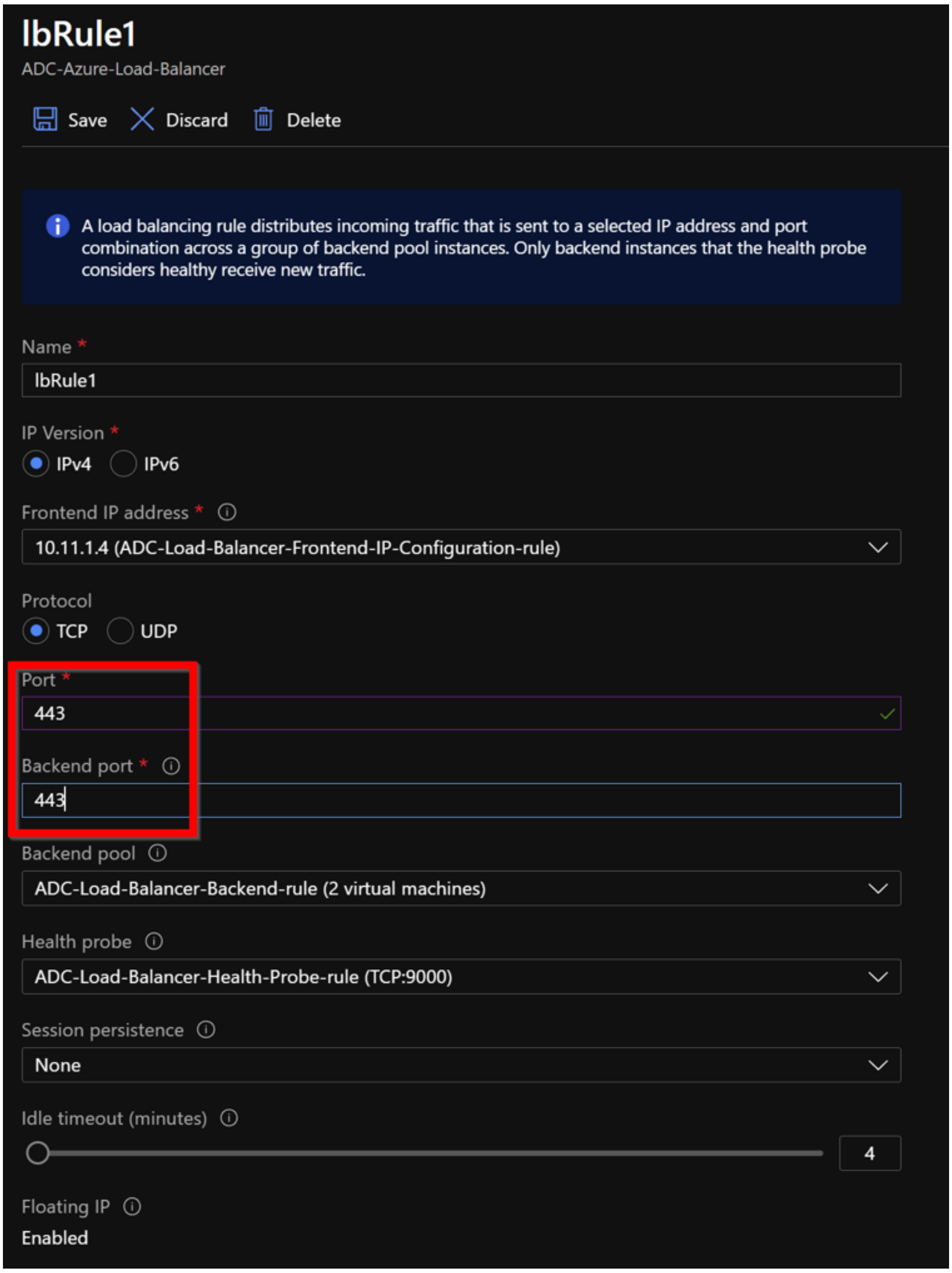

- The LB Rule (LbRule1) uses port 80, by default.




- Edit the rule to use port 443, and save the changes.

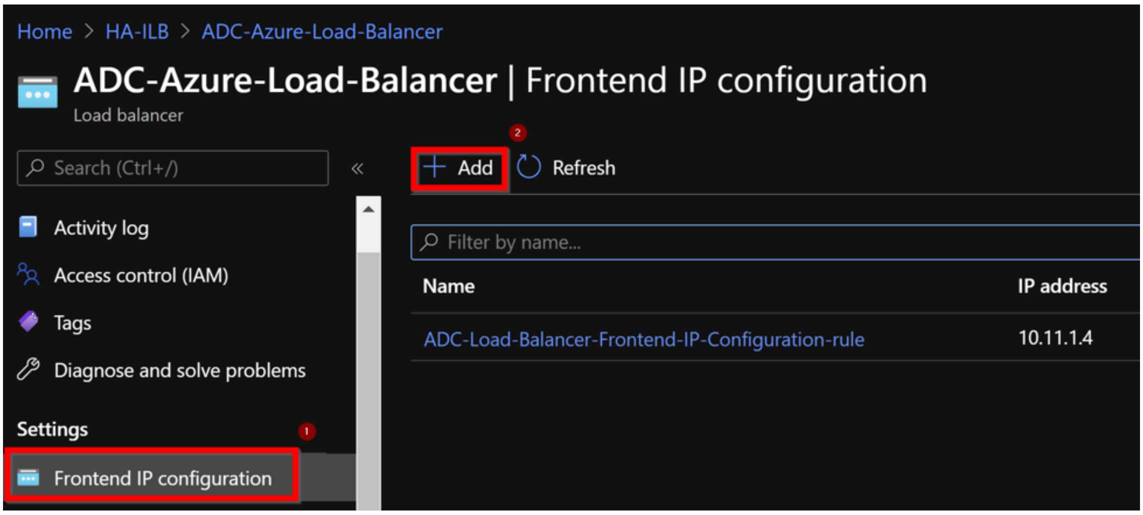
> **Note:**
>
> For enhanced security, Citrix recommends you to use SSL port 443 for LB virtual server or Gateway virtual server.

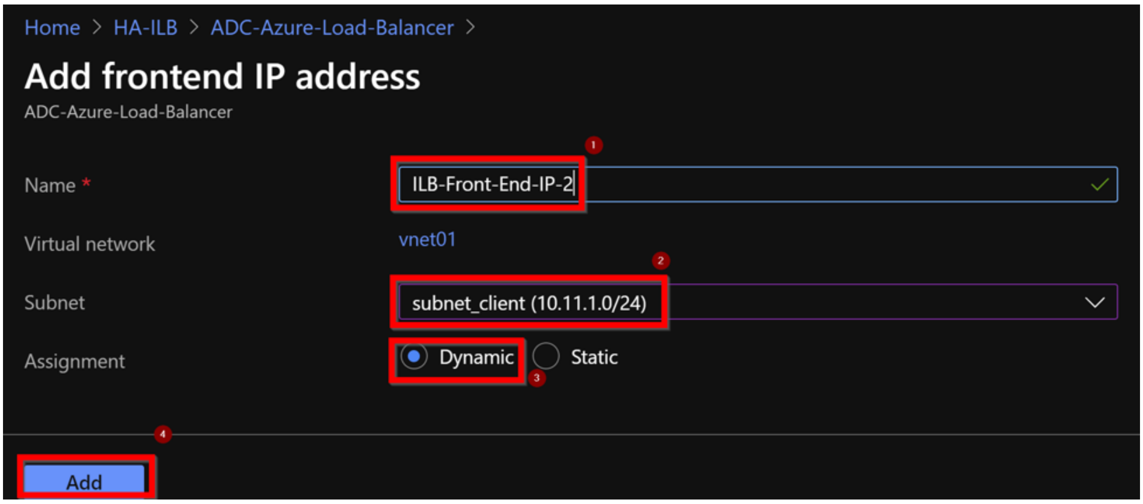



To add more VIP addresses on the ADC, perform the following steps:

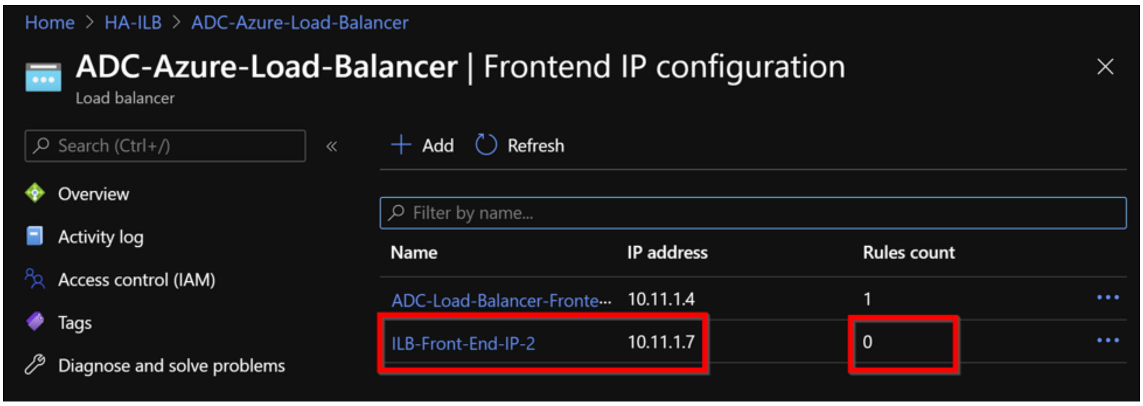
1. Navigate to **Azure Load Balancer > Frontend IP configuration**, and click **Add** to create a new internal load balancer IP address.




2. In the **Add frontend IP address** page, enter a name, choose the client subnet, assign either dynamic or static IP address, and click **Add**.

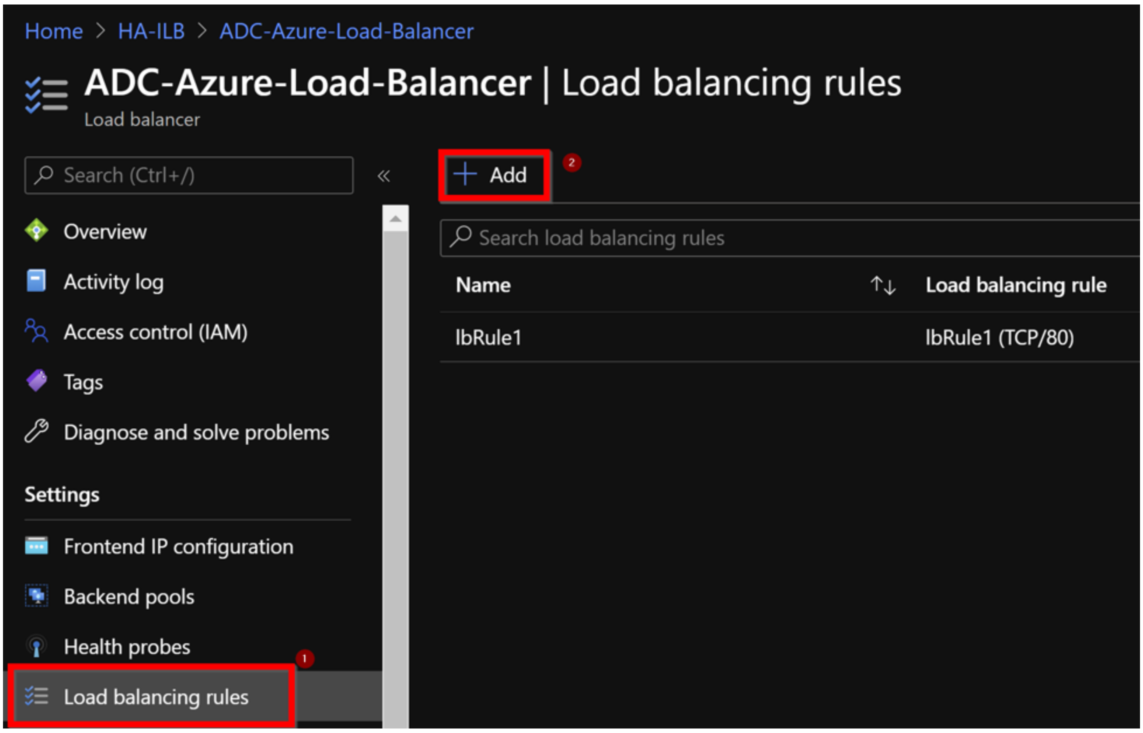



3. The front-end IP address is created but an LB Rule is not associated. Create a new load balancing rule, and associate it with the front-end IP address.



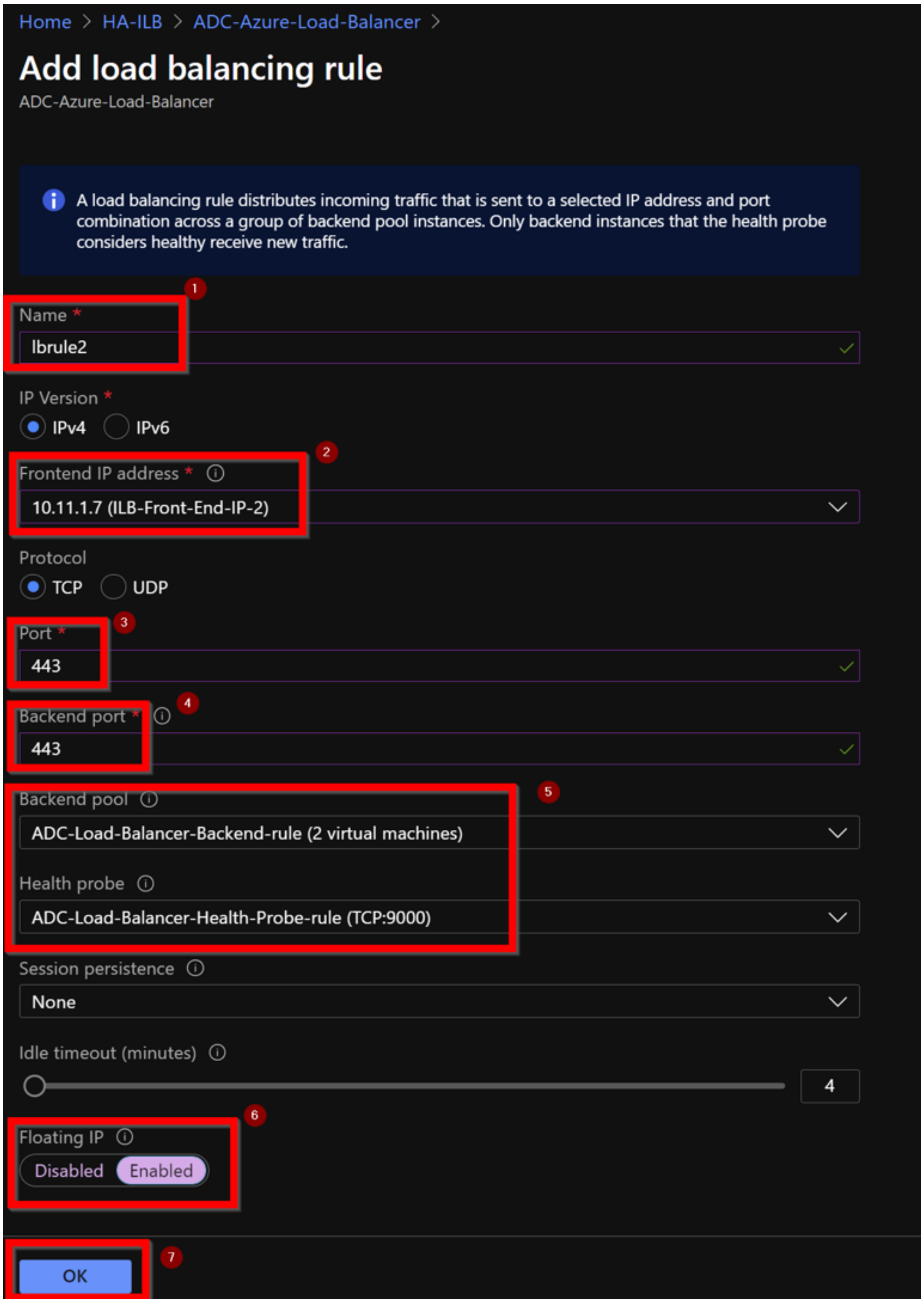

4. In the **Azure Load Balancer** page, select **Load balancing rules**, and then click **Add**.

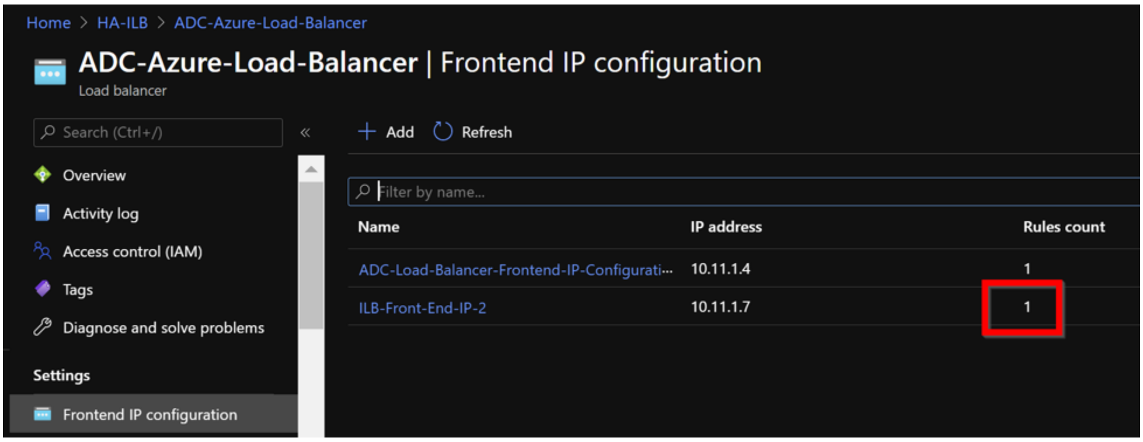



5. Create a new LB Rule by choosing the new front-end IP address and the port. **Floating IP** field must be set to **Enabled**.




6. Now the **Frontend IP configuration** shows the LB rule that is applied.




## Sample configuration

To configure a gateway VPN virtual server and load balancing virtual server, run the following commands on the primary node (ADC-VPX-0). The configuration auto synchronizes to the secondary node (ADC-VPX-1).

Gateway sample configuration

```
enable feature aaa LB SSL SSLVPN
enable ns mode MBF
add vpn vserver vpn_ssl SSL 10.11.1.4 443
add ssl certKey ckp -cert wild-cgwsanity.cer -key wild-cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
```

Load balancing sample configuration

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 10.11.1.7 443
bind ssl vserver lb_vs1 -certkeyName ckp
```

You can now access the load balancing or VPN virtual server using the fully qualified domain name (FQDN) associated with the internal IP address of ILB.

See the **Resources** section for more information about how to configure the load-balancing virtual server.

**Resources**:

The following links provide additional information related to HA deployment and virtual server configuration:

- Configuring high availability nodes in different subnets
- Set up basic load balancing

**Related resources**:

- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands
- Configuring GSLB on Active-Standby HA Deployment on Azure

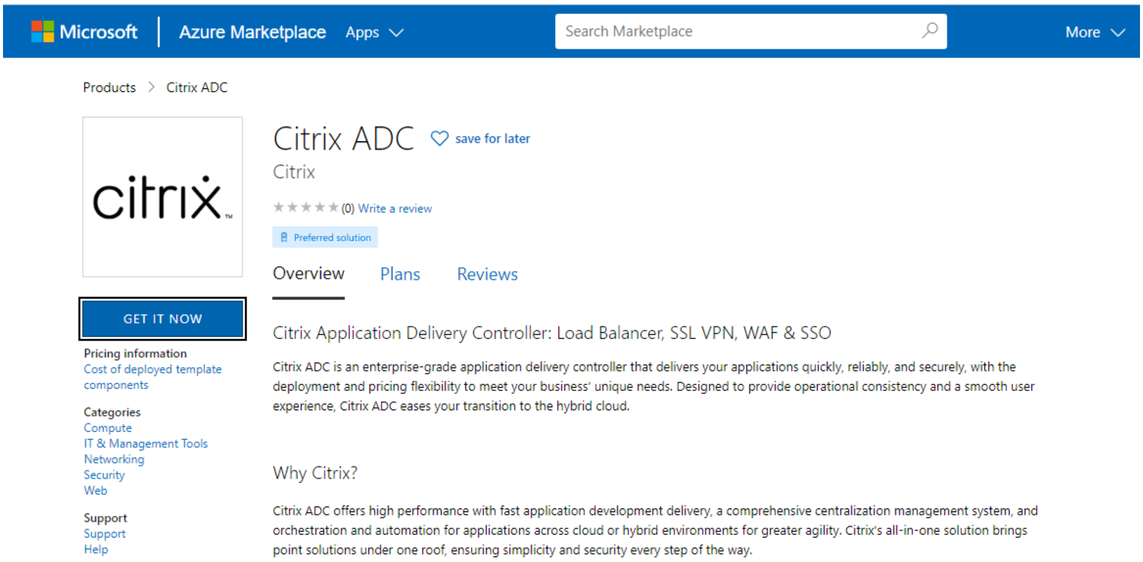
## Configure HA-INC nodes by using the NetScaler high availability template for internet-facing applications

You can quickly and efficiently deploy a pair of VPX instances in HA-INC mode by using the standard template for internet-facing applications. The Azure load balancer (ALB) uses a public IP address for the front end. The template creates two nodes, with three subnets and six NICs. The subnets are for management, client, and server-side traffic. Each subnet has two NICs for both the VPX instances.

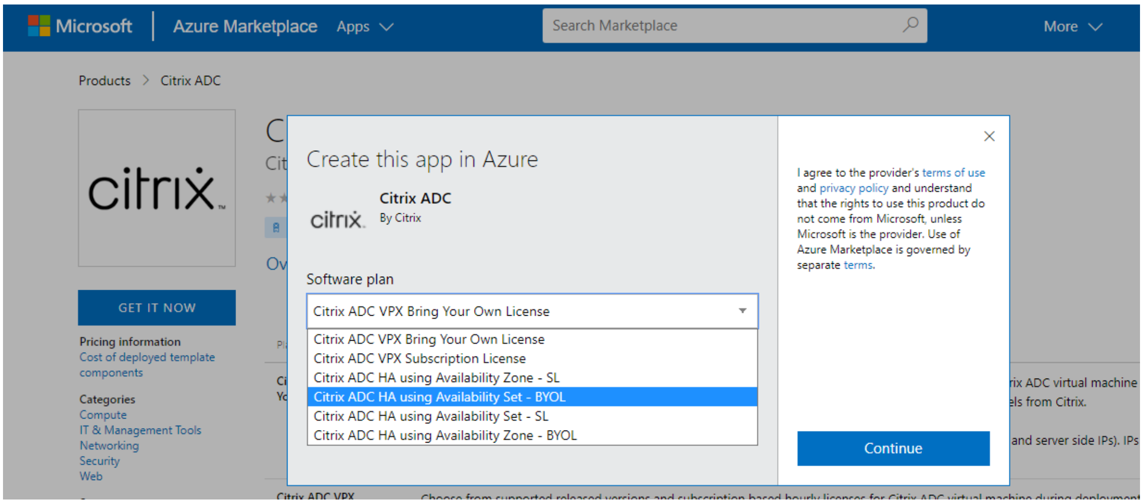
You can get the NetScaler HA pair template for internet-facing applications at the Azure Marketplace.

Complete the following steps to launch the template and deploy a high availability VPX pair by using Azure availability sets or availability zone.

1. From the Azure Marketplace, search **NetScaler**.
2. Click **GET IT NOW**.

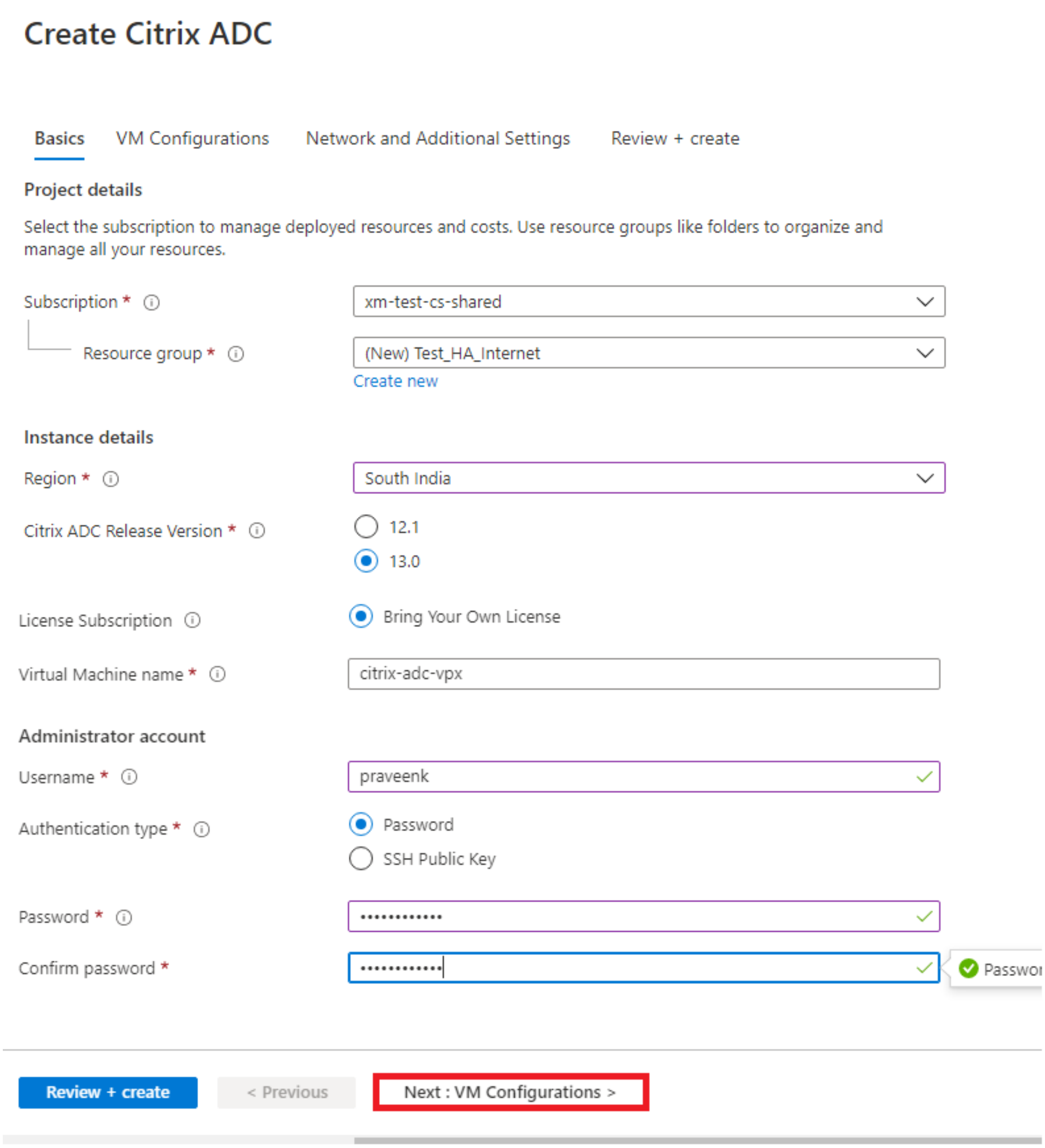



3. Select the required HA deployment along with license, and click **Continue**.

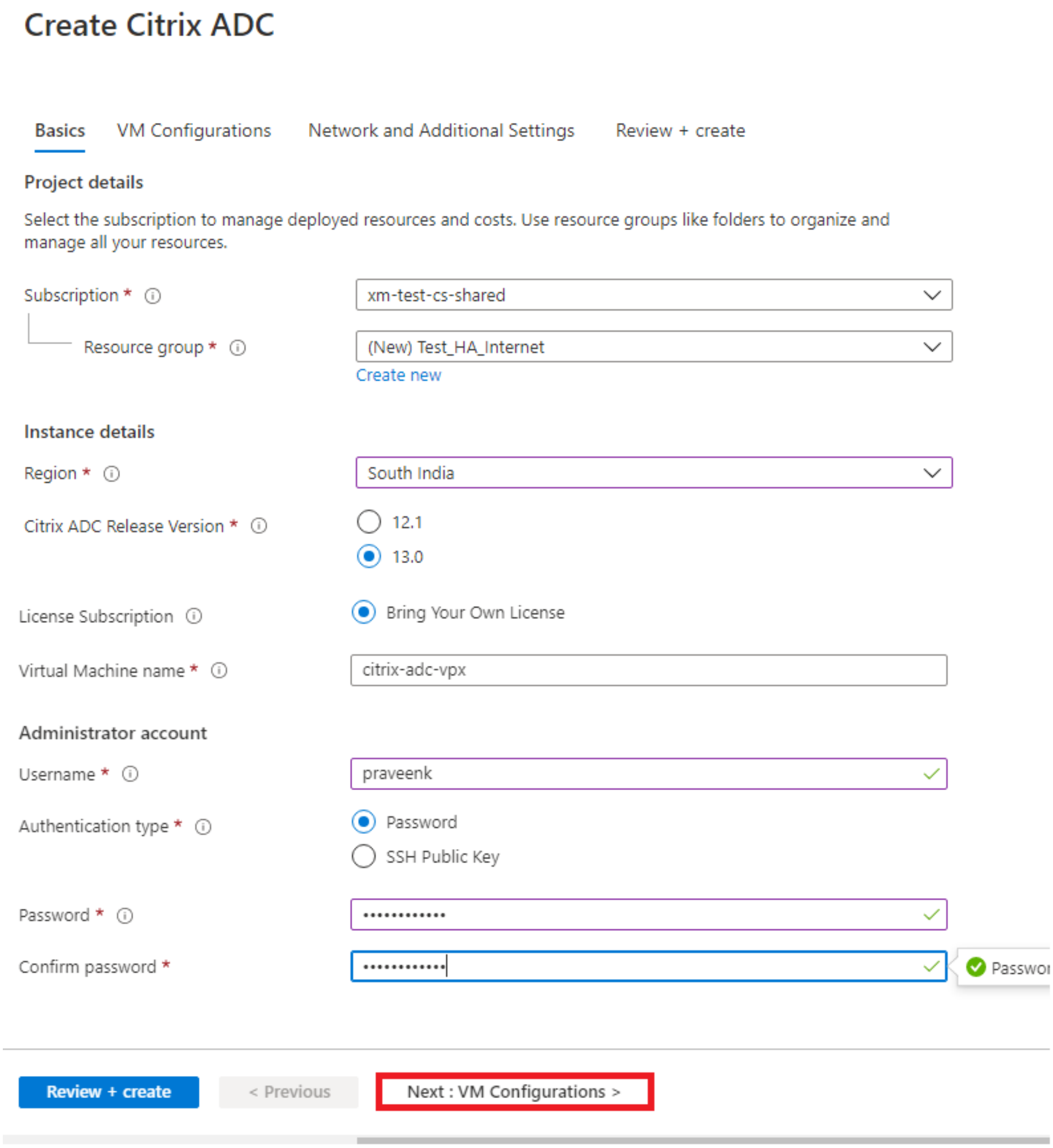



4. The **Basics** page appears. Create a Resource Group. Under the **Parameters** tab, enter details for the Region, Admin user name, Admin Password, license type (VM SKU), and other fields.




5. Click **Next : VM Configurations >**.

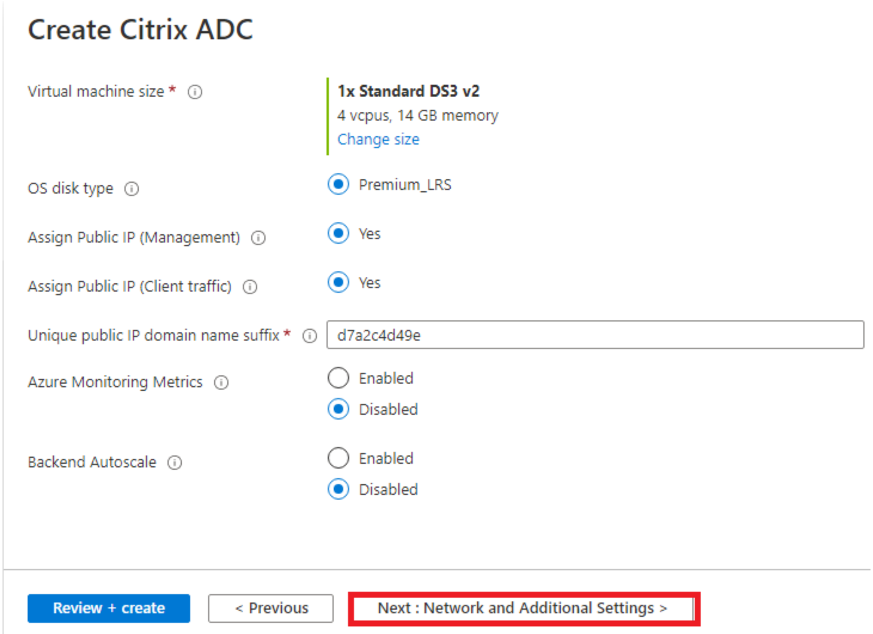



6. On the **VM Configurations** page, perform the following:

   - Configure public IP domain name suffix
   - Enable or disable **Azure Monitoring Metrics**
   - Enable or disable **Backend Autoscale**

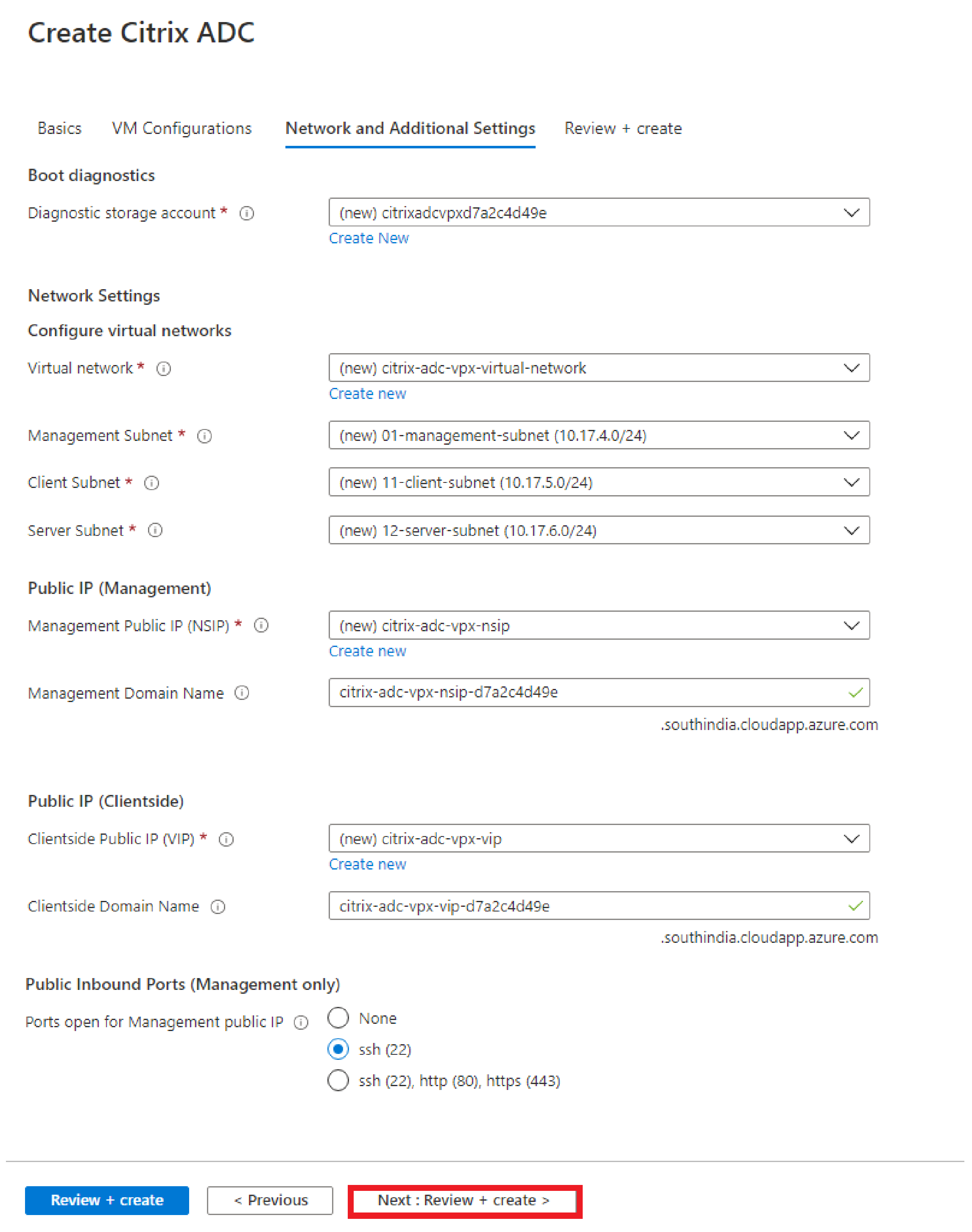
7. Click **Next: Network and Additional settings >**




8. On **Network and Additional Settings** page, create Boot diagnostics account and configure the network settings.

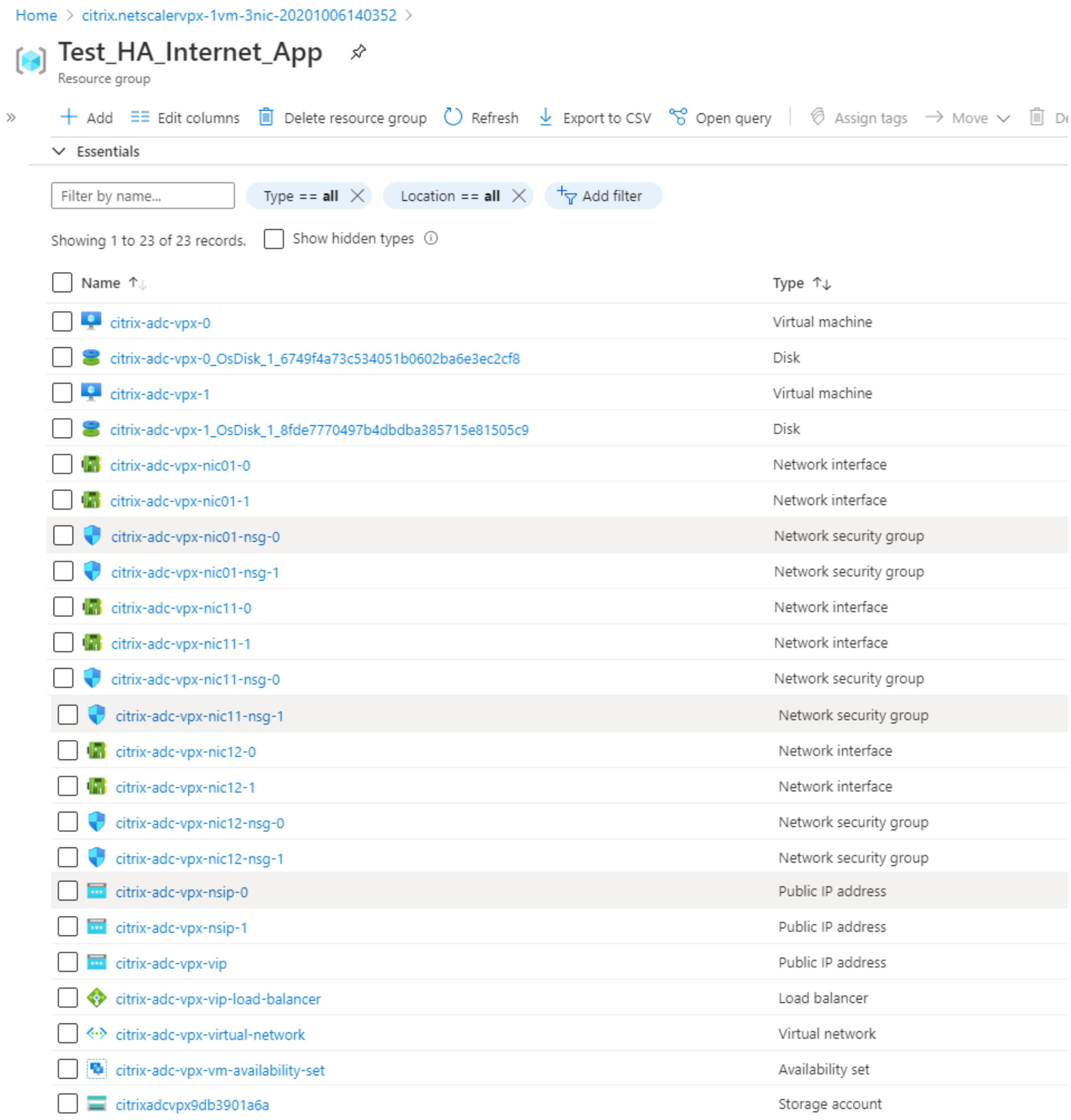



9. Click **Next: Review + create >**.
10. Review the basic settings, VM configuration, network and additional settings, and click **Create**.

    It might take a moment for the Azure Resource Group to be created with the required configurations. After completion, select the Resource Group in the Azure portal to see the configuration details, such as LB rules, back-end pools, and health probes. The high availability pair appears as **citrix-adc-vpx-0** and **citrix-adc-vpx-1**.

If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.

Once the required configuration is complete, the following resources are created.




11. You must log on to **citrix-adc-vpx-0** and **citrix-adc-vpx-1** nodes to validate the following configuration:

    - NSIP addresses for both nodes must be in the management subnet.
    - On the primary (citrix-adc-vpx-0) and secondary (citrix-adc-vpx-1) nodes, you must see

two SNIP addresses. One SNIP (client subnet) is used for responding to the ALB probes and the other SNIP (server subnet) is used for back-end server communication.

> **Note:**
>
> In the HA-INC mode, the SNIP addresses of the citrix-adc-vpx-0 and citrix-adc-vpx-1 VMs are different, unlike with the classic on-premises ADC high availability deployment where both are the same.

On the primary node (citrix-adc-vpx-0)

```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.18.0.4        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.4        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.4 (ns-vpx0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:34:21 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.5
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
```

On the secondary node (citrix-adc-vpx-1)

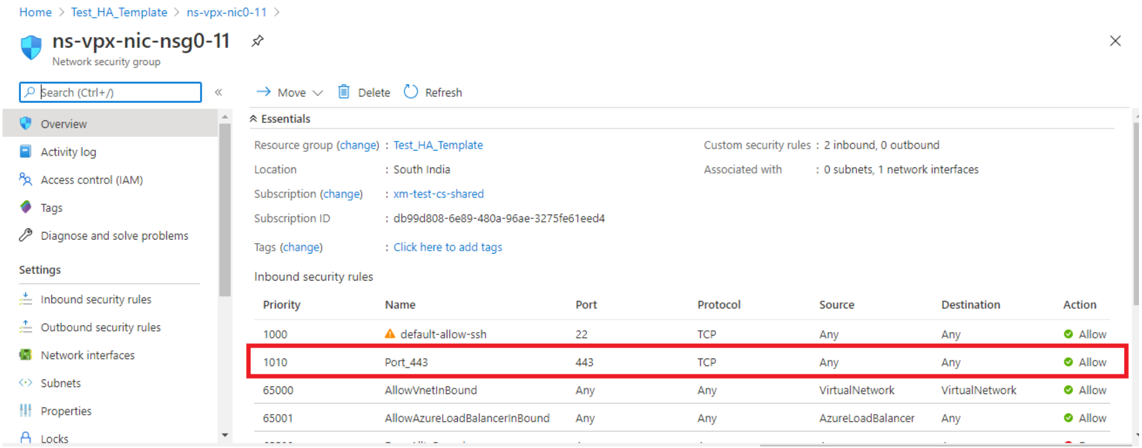
```
> show ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.18.0.5        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.4        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
```

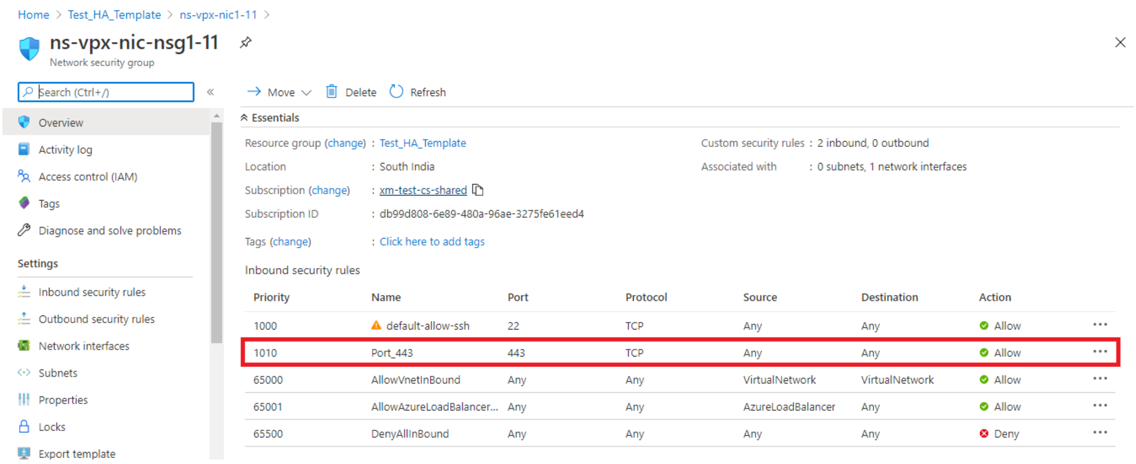
```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.5 (ns-vpx1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:23:51 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.4
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

12. After the primary and secondary nodes are UP and the Synchronization status is **SUCCESS**, you must configure the load balancing virtual server or the gateway virtual server on the primary node (citrix-adc-vpx-0) with the public IP address of the ALB virtual server. For more information, see the Sample configuration section.
13. To find the public IP address of ALB virtual server, navigate to **Azure portal > Azure Load Balancer > Frontend IP configuration**.

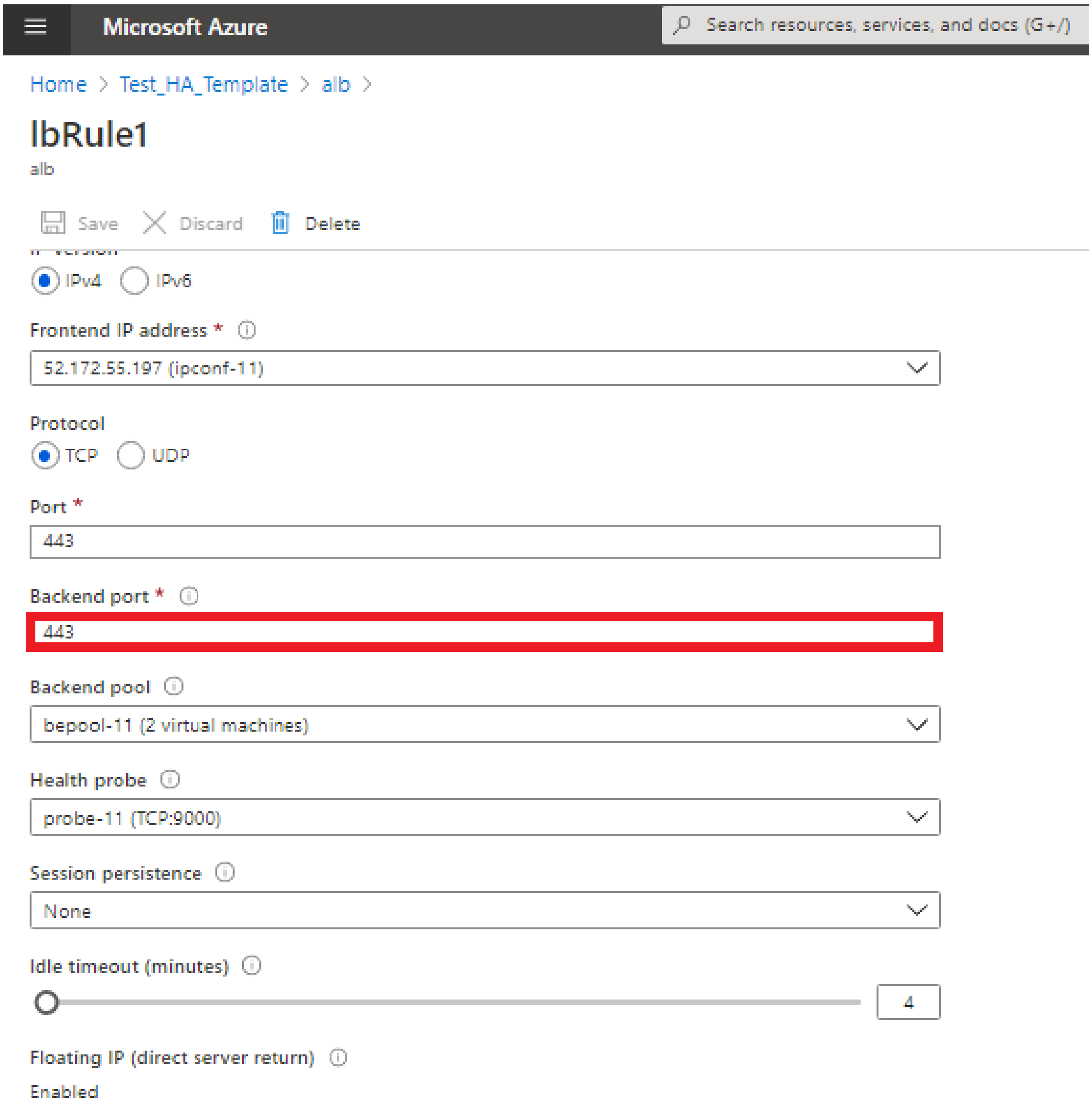
14. Add the inbound security rule for virtual server port 443 on the network security group of both the client interfaces.



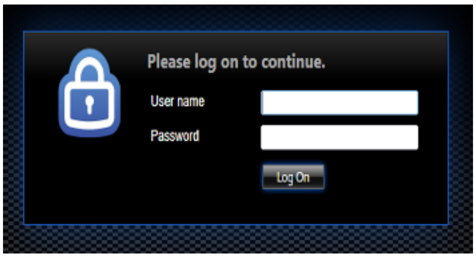


15. Configure the ALB port that you want to access, and create inbound security rule for the specified port. The Backend port is your load balancing virtual server port or the VPN virtual server port.




16. Now, you can access the load balancing virtual server or the VPN virtual server using the fully qualified domain name (FQDN) associated with the ALB public IP address.



### Sample configuration

To configure a gateway VPN virtual server and load balancing virtual server, run the following commands on the primary node (ADC-VPX-0). The configuration auto synchronizes to the secondary node (ADC-VPX-1).

Gateway sample configuration

```
enable feature aaa LB SSL SSLVPN
add ip 52.172.55.197 255.255.255.0 -type VIP
add vpn vserver vpn_ssl SSL 52.172.55.197 443
add ssl certKey ckp -cert cgwsanity.cer -key cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
```

Load balancing sample configuration

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 52.172.55.197 443
bind ssl vserver lb_vs1 -certkeyName ckp
```

You can now access the load balancing or VPN virtual server using the FQDN associated with the public IP address of ALB.

See the **Resources** section for more information about how to configure the load balancing virtual server.

**Resources**:

The following links provide additional information related to HA deployment and virtual server configuration:

- Create virtual servers
- Set up basic load balancing

## Configure a high-availability setup with Azure external and internal load balancers simultaneously

The high availability pair on Azure supports both external and internal load balancers simultaneously.

You have the following two options to configure a high availability pair using both Azure external and internal load balancers:

- Using two LB virtual servers on the NetScaler appliance.
- Using one LB virtual server and an IP set. The single LB virtual server serves traffic to multiple IPs, which are defined by the IPset.

Perform the following steps to configure a high availability pair on Azure using both the external and internal load balancers simultaneously:

For Steps 1 and 2, use the Azure portal. For Steps 3 and 4, use the NetScaler VPX GUI or the CLI.

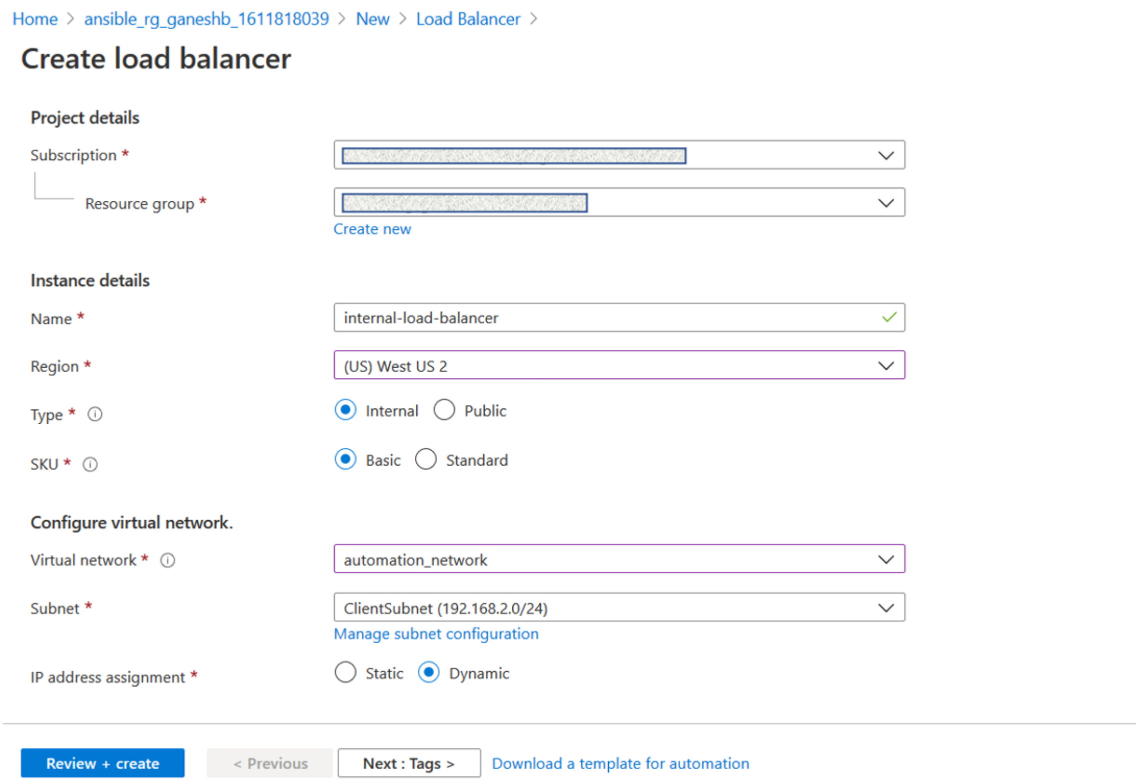
**Step 1.** Configure an Azure load balancer, either an external load balancer or an internal load balancer.

For more information on configuring high-availability setup with Azure external load balancers, see Configure a high-availability setup with multiple IP addresses and NIC.

For more information on configuring high-availability setup with Azure internal load balancers, see Configure HA-INC nodes by using the NetScaler high availability template with Azure ILB.

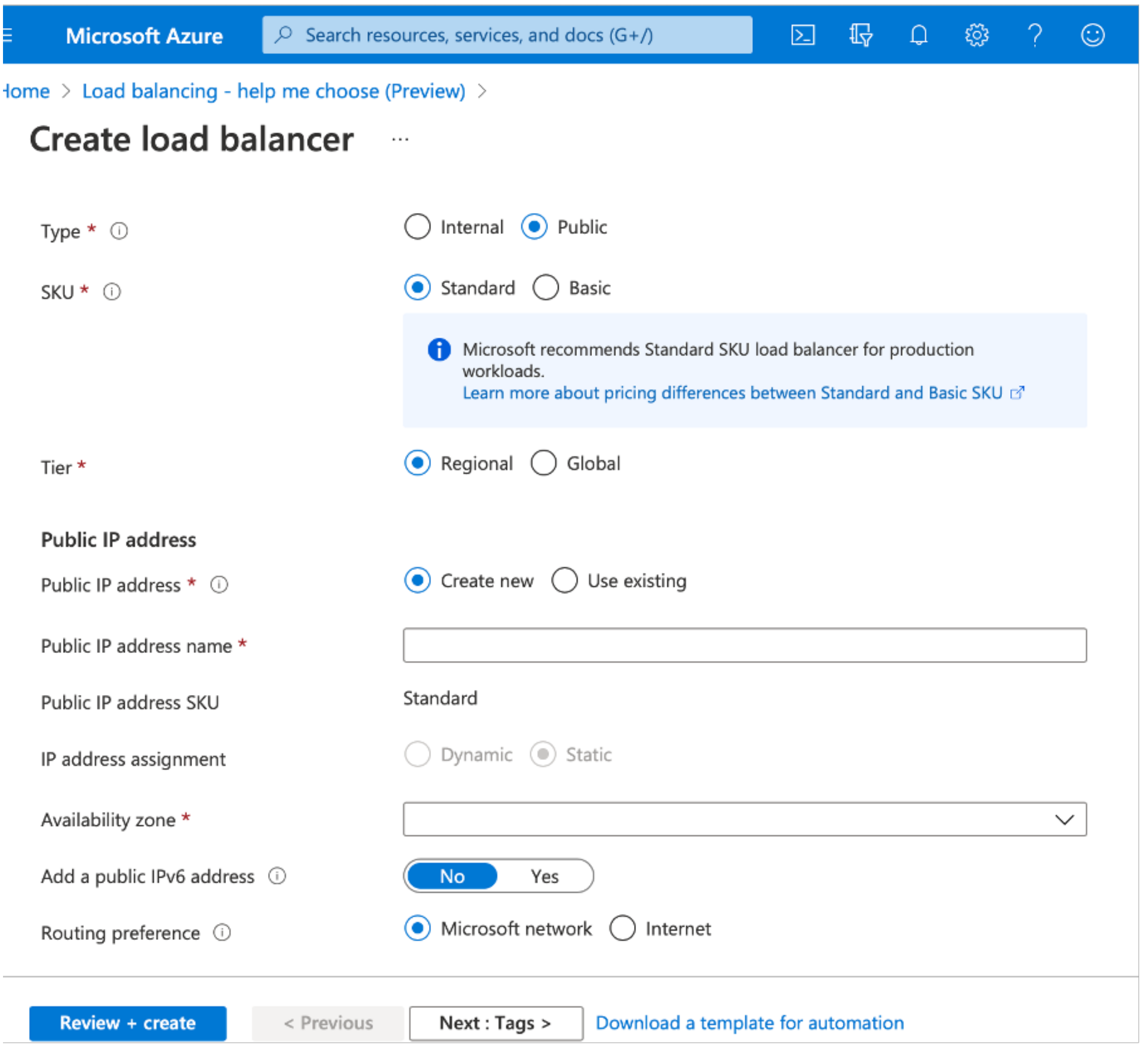
**Step 2.** Create an extra load balancer (ILB) in your resource group. In Step 1, if you have created an external load balancer, you now create an internal load balancer and conversely.

- To create an internal load balancer, choose the load balancer type as **Internal**. For the **Subnet** field, you must choose your NetScaler client subnet. You can choose to provide a static IP address in that subnet, provided there are no conflicts. Otherwise, choose the dynamic IP address.




- To create an external load balancer, choose the load balancer type as **Public** and create the public IP address here.




1. After you have created the Azure Load Balancer, navigate to **Frontend IP configuration** and note down the IP address shown here. You must use this IP address while creating the ADC load balancing virtual server as in Step 3.


2. In the **Azure Load Balancer configuration** page, the ARM template deployment helps create the LB rule, back-end pools, and health probes.
3. Add the high availability pair client NICs to the backend pool for the ILB.
4. Create a health probe (TCP, 9000 port)
5. Create two load balancing rules:
   - One LB rule for HTTP traffic (webapp use case) on port 80. The rule must also use the backend port 80. Select the created backend pool and the health probe. Floating IP must be enabled.
   - Another LB rule for HTTPS or CVAD traffic on port 443. The process is the same as the HTTP traffic.

**Step 3.** On the primary node of NetScaler appliance, create a load balancing virtual server for ILB.

1. Add a load balancing virtual server.

```
add lb vserver <name> <serviceType> [<ILB Frontend IP address>] [<port>]
```

**Example:**

```
add lb vserver vserver_name HTTP 52.172.96.71 80
```

> **Note:**
>
> Use the load balancer frontend IP address, which is associated with the additional Load balancer that you create in Step 2.

2. Bind a service to a load balancing virtual server.

```
bind lb vserver <name> <serviceName>
```

**Example:**

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

For more information, see Set up basic load balancing

**Step 4:** As an alternative to Step 3, you can create a load balancing virtual server for ILB using IPsets.

1. Add an IP address of type virtual server IP (VIP).

   ```
   add nsip <ILB Frontend IP address> -type <type>
   ```

   **Example:**

   ```
   add nsip 52.172.96.71 -type vip
   ```

2. Add an IPset on both primary and secondary nodes.

   ```
   add ipset <name>
   ```

   **Example:**

   ```
   add ipset ipset1
   ```

3. Bind IP addresses to the IP set.

   ```
   bind ipset <name> <ILB Frontend IP address>
   ```

   **Example:**

   ```
   bind ipset ipset1 52.172.96.71
   ```

4. Set the existing LB virtual server to use the IPset.

   ```
   set lb vserver <vserver name> -ipset <ipset name>
   ```

   **Example:**

   ```
   set lb vserver vserver_name -ipset ipset1
   ```

For more information, see Configure a multi-IP virtual server.

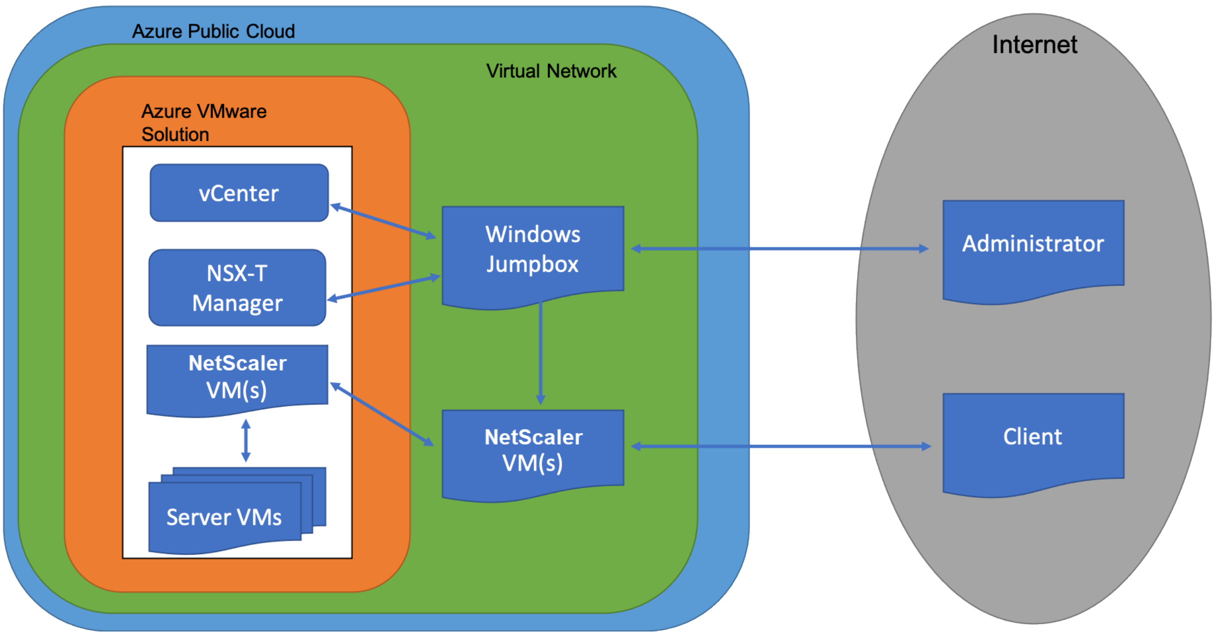
## Install a NetScaler VPX instance on Azure VMware Solution

Azure VMware Solution (AVS) provides you with private clouds that contain vSphere clusters, built from dedicated bare-metal Azure infrastructure. The minimum initial deployment is three hosts, but additional hosts can be added one at a time, up to a maximum of 16 hosts per cluster. All provisioned private clouds have vCenter Server, vSAN, vSphere, and NSX-T.

The VMware Cloud (VMC) on Azure enables you to create cloud software-defined data centers (SDDC) on Azure with the number of ESX hosts that you want. The VMC on Azure supports NetScaler VPX deployments. VMC provides a user interface same as on-prem vCenter. It functions similar to the ESX-based NetScaler VPX deployments.

The following diagram shows the Azure VMware solution on the Azure public cloud that an administrator or a client can access over the internet. An administrator can create, manage, and configure workload or server VMs using Azure VMware solution. The admin can access the AVS's web-based vCenter and NSX-T Manager from a Windows Jumpbox. You can create the NetScaler VPX instances (standalone or high availability pair) and server VMs within Azure VMware Solution using vCenter, and manage the corresponding networking using NSX-T manager. The NetScaler VPX instance on AVS works similar to the on-prem VMware cluster of hosts. AVS is managed from a Windows Jumpbox that is created in the same virtual network.

A client can only access the AVS service by connecting to the VIP of ADC. Another NetScaler VPX instance outside Azure VMware Solution but in the same Azure virtual network helps add the VIP of the NetScaler VPX instance within Azure VMware Solution as a service. As per requirement, you can configure the NetScaler VPX instance to provide service over the internet.




## Prerequisites

Before you begin installing a virtual appliance, do the following:

- For more information on Azure VMware solution and its prerequisites, see Azure VMware Solution documentation.
- For more information on deploying Azure VMware solution, see Deploy an Azure VMware Solution private cloud.
- For more information on creating a Windows Jump box VM to access and manage Azure VMware Solution, see Access an Azure VMware Solution private cloud
- In Windows Jump box VM, download the NetScaler VPX appliance setup files.
- Create appropriate NSX-T network segments on VMware SDDC to which the virtual machines connect. For more information, see Add a network segment in Azure VMware Solution
- Obtain VPX license files.
- Virtual machines (VMs) created or migrated to the Azure VMware Solution private cloud must be attached to a network segment.

### VMware cloud hardware requirements

The following table lists the virtual computing resources that the VMware SDDC must provide for each VPX nCore virtual appliance.

Table 1. Minimum virtual computing resources required for running a NetScaler VPX instance

| Component | Requirement |
| --- | --- |
| Memory | 2 GB |
| Virtual CPU (vCPU) | 2 |
| Virtual network interfaces | In VMware SDDC, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded to version 7 or higher. |
| Disk space | 20 GB |

> **Note:**
>
> This is in addition to any disk requirements for the hypervisor.

For production use of the VPX virtual appliance, the full memory allocation must be reserved.

### OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. The following table describes the system requirements for installing OVF tool.

Table 2. System requirements for OVF tool installation

| Component | Requirement |
| --- | --- |
| Operating system | For detailed requirements from VMware, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/. |
| CPU | 750 MHz minimum, 1 GHz or faster recommended |
| RAM | 1 GB Minimum, 2 GB recommended |
| NIC | 100 Mbps or faster NIC |

For information about installing OVF, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/.

### Downloading the NetScaler VPX setup files

The NetScaler VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log on. If you do not have a Citrix account, access the home page at http://www.citrix.com. Click the **New Users link**, and follow the instructions to create a new Citrix account.

Once logged on, navigate the following path from the Citrix home page:

Citrix.com > **Downloads > NetScaler > Virtual Appliances**.

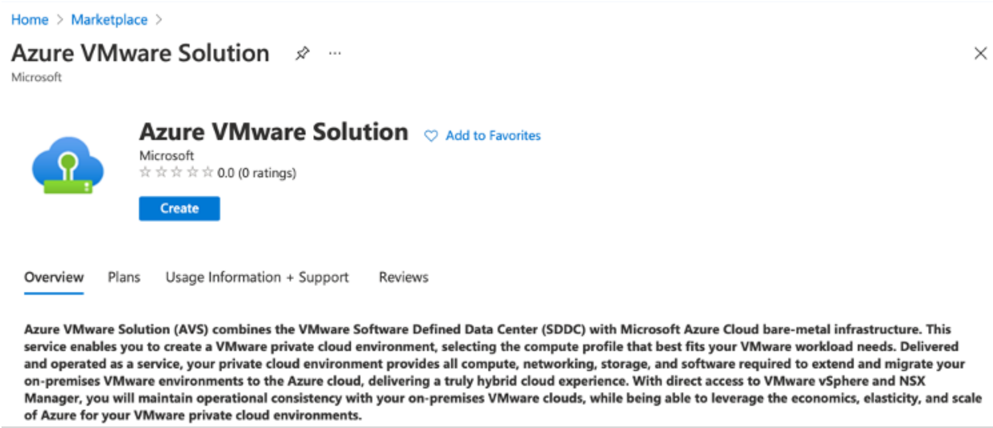
Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-13.0-79.64.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-13.0-79.64.mf)

### Deploy Azure VMware solution

1. Log in to your Microsoft Azure portal, and navigate to **Azure Marketplace**.
2. From the **Azure Marketplace**, search **Azure VMware Solution** and click **Create**.

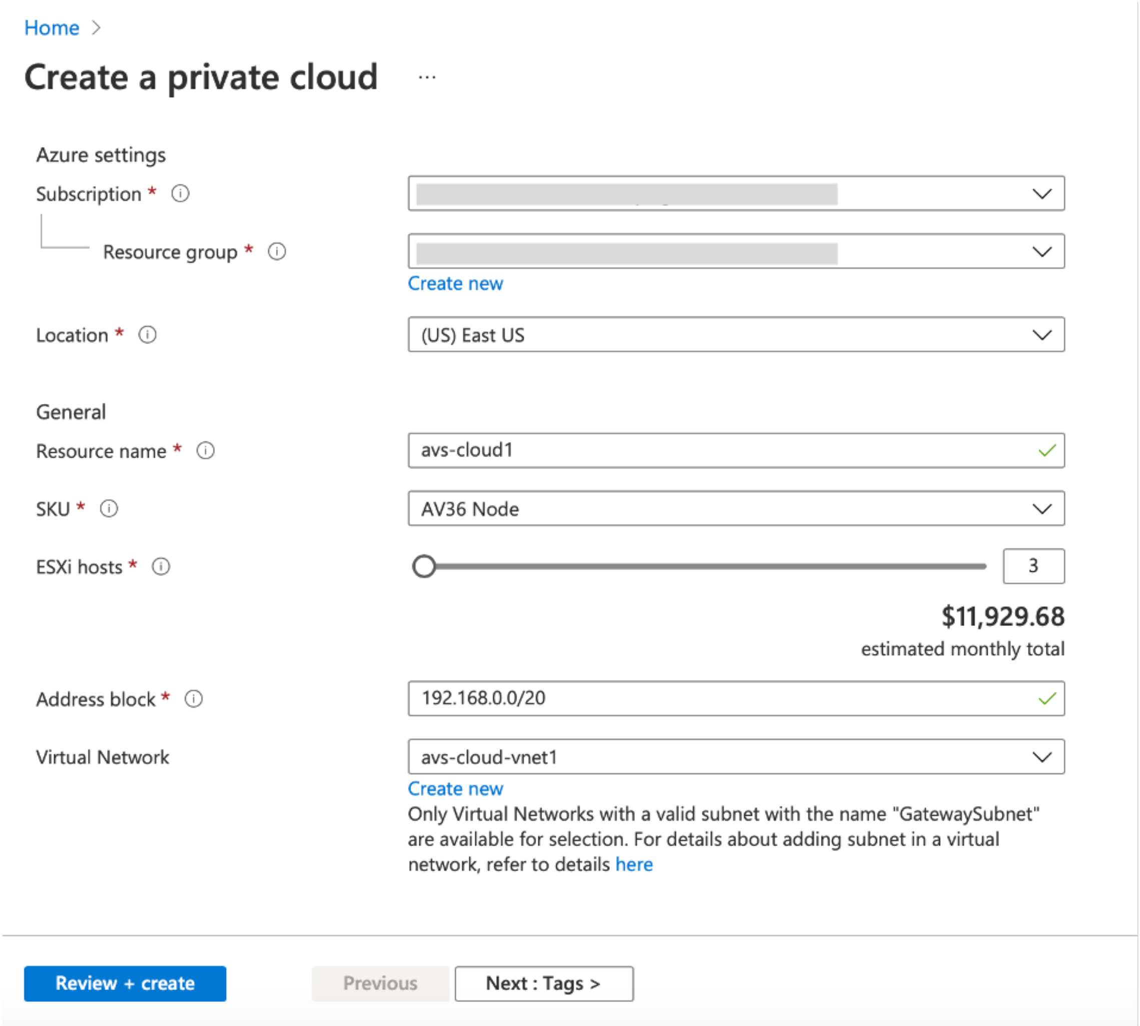
 




3. In the **Create a private cloud** page, enter the following details:

    - Select a minimum of 3 ESXi hosts to create the default cluster of your private cloud.
    - For the **Address block** field, use **/22** address space.
    - For the **Virtual Network**, make sure that the CIDR range doesn't overlap with any of your on-premises or other Azure subnets (virtual networks) or with the gateway subnet.
    - Gateway subnet is used to express route the connection with private cloud.

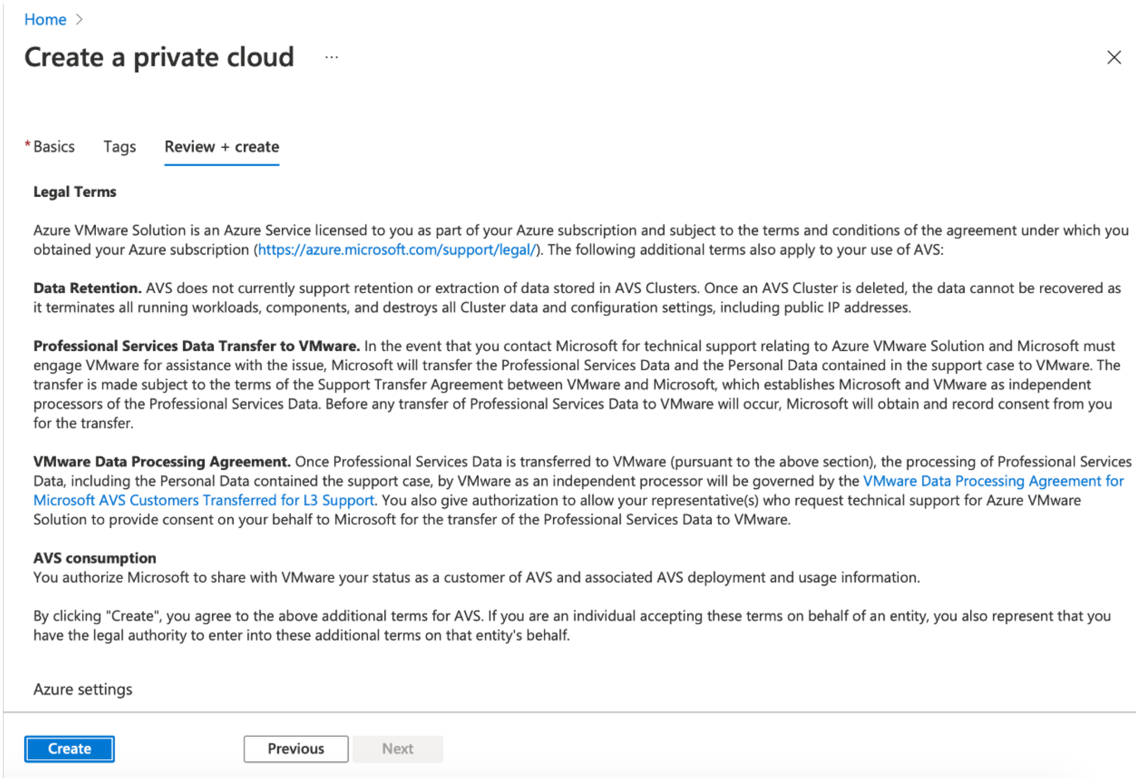
 



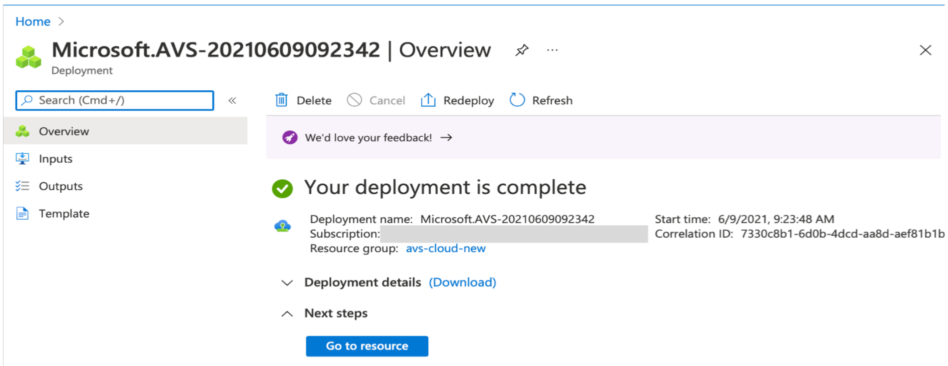

4. Click **Review + Create**.

5. Review the settings. If you must change any settings, click **Previous**.

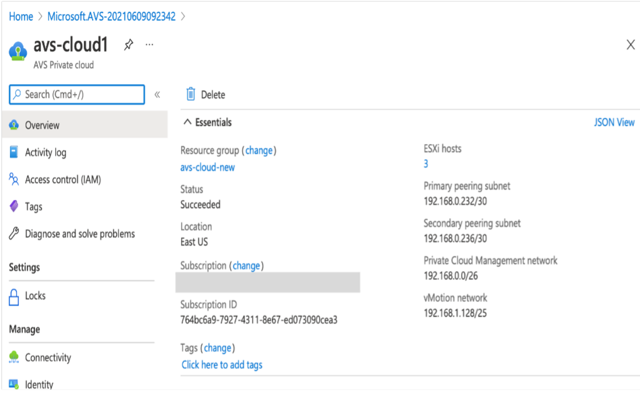



6. Click **Create**. Private cloud provisioning process starts. It can take up to two hours for the private cloud to be provisioned.




7. Click **Go to resource**, to verify the private cloud that is created.



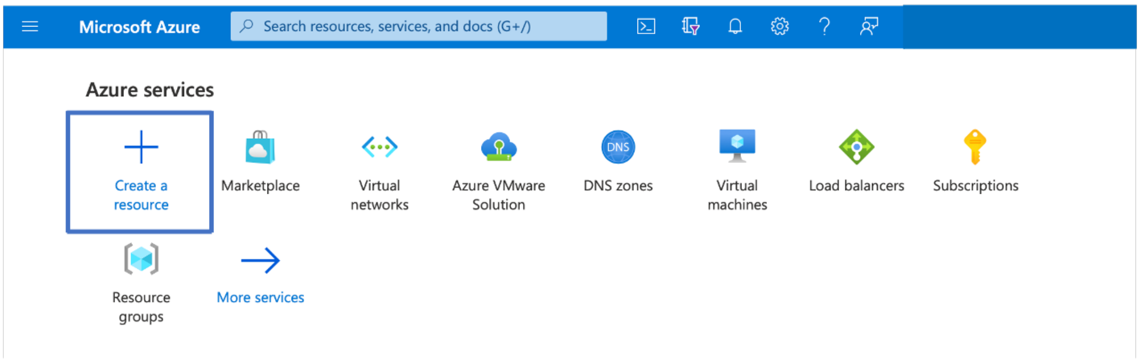

> **Note**
>
> To access this resource, you need a VM in Windows that acts as a Jump box.

## Connect to an Azure virtual machine running Windows

This procedure shows you how to use the Azure portal to deploy a virtual machine (VM) in Azure that runs Windows Server 2019. To see your VM in action, you then RDP to the VM and install the IIS web server.

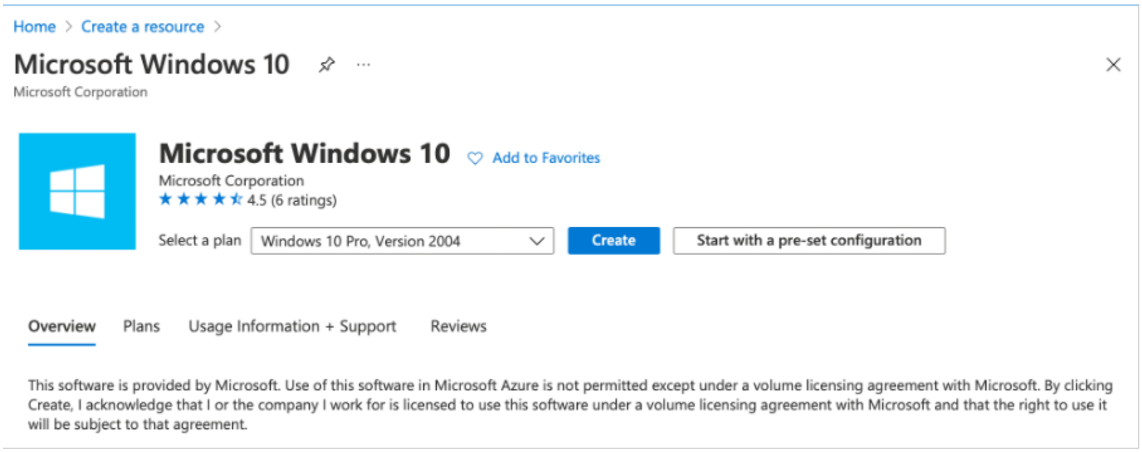
To access the private cloud that you have created, you need to create a Windows Jump box within the same virtual network.

1. Go to the **Azure portal**, and click **Create a Resource**.



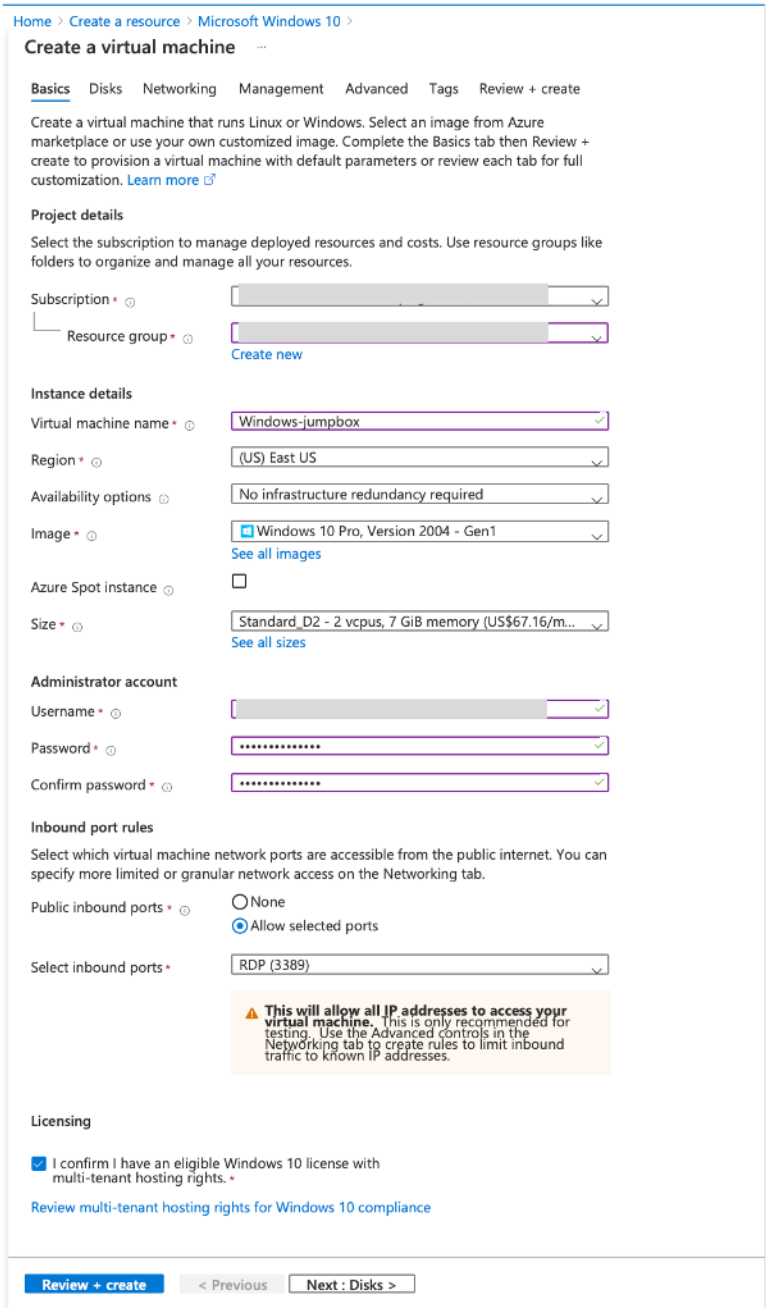

2. Search for **Microsoft Windows 10**, and click **Create**.




3. Create a virtual machine (VM) that runs Windows Server 2019. The **Create a virtual machine** page appears. Enter all the details in **Basics** tab, and select the **Licensing** check box. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.




4. After validation runs, select the **Create** button at the bottom of the page.

5. After the deployment is complete, select **Go to resource**.

6. Go to the Windows VM that you have created. Use the public IP address of the Windows VM and connect using RDP.

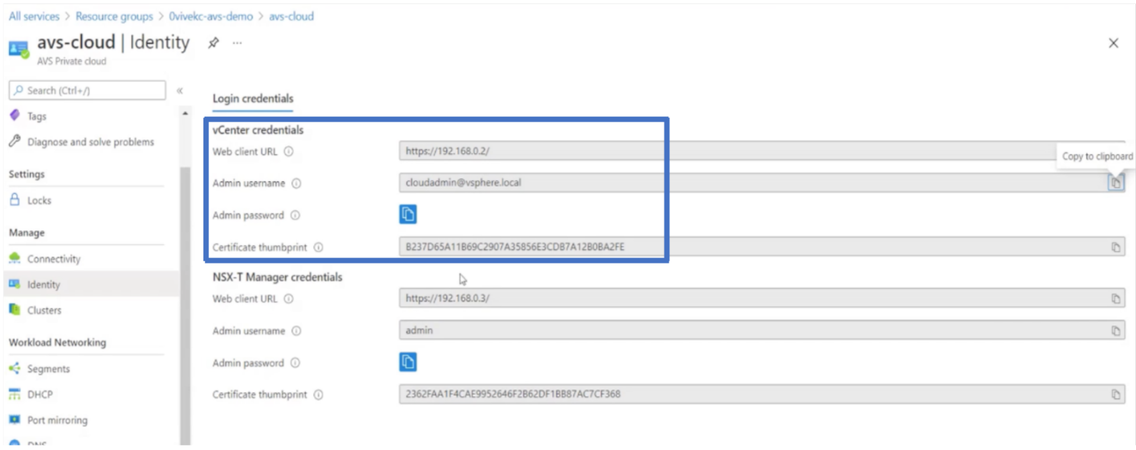
   Use the **Connect** button in the Azure portal to start a Remote Desktop (RDP) session from a Windows desktop. First you connect to the virtual machine, and then you sign on.

   To connect to a Windows VM from a Mac, you must install an RDP client for Mac such as Microsoft

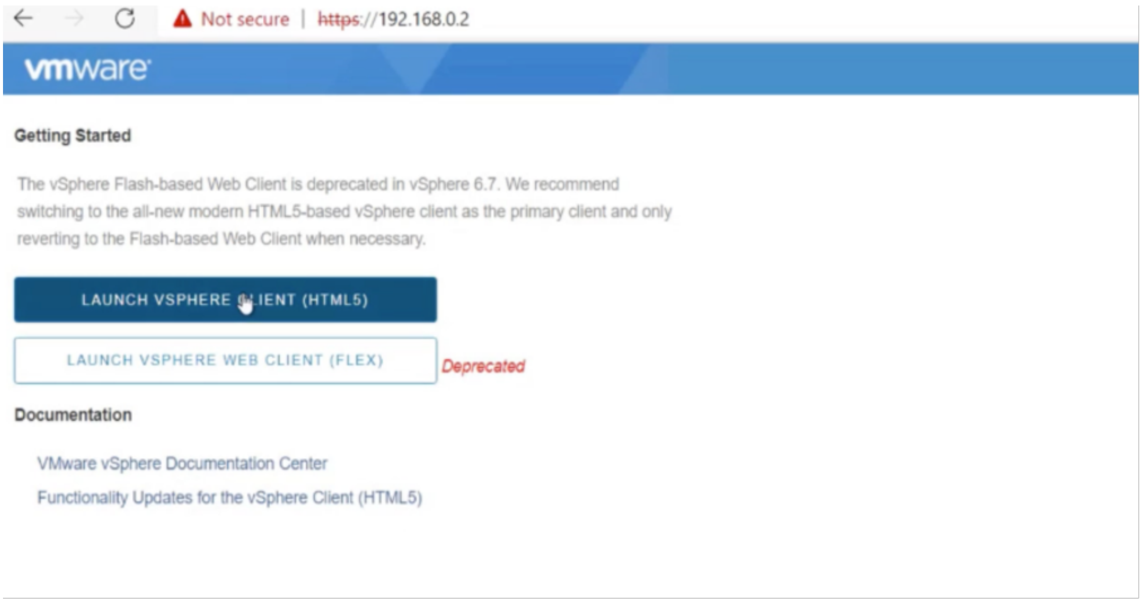
Remote Desktop. For more information, see How to connect and sign on to an Azure virtual machine running Windows.

### Access your Private Cloud vCenter portal

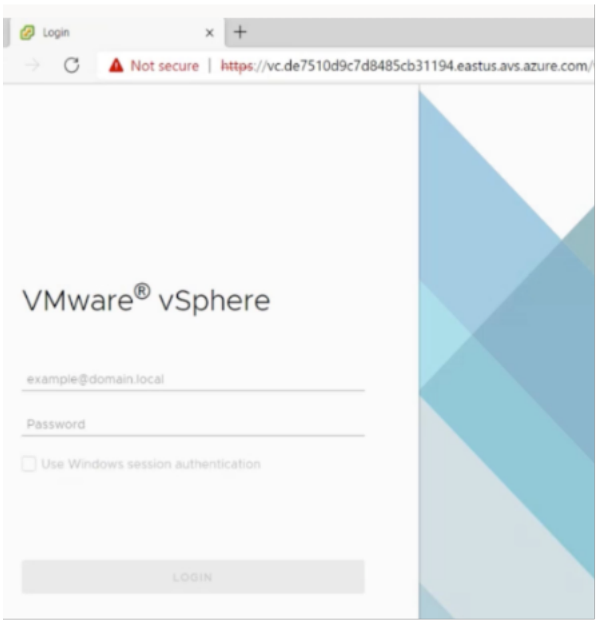
1. In your Azure VMware Solution private cloud, under **Manage**, select **Identity**. Make note of the vCenter credentials.



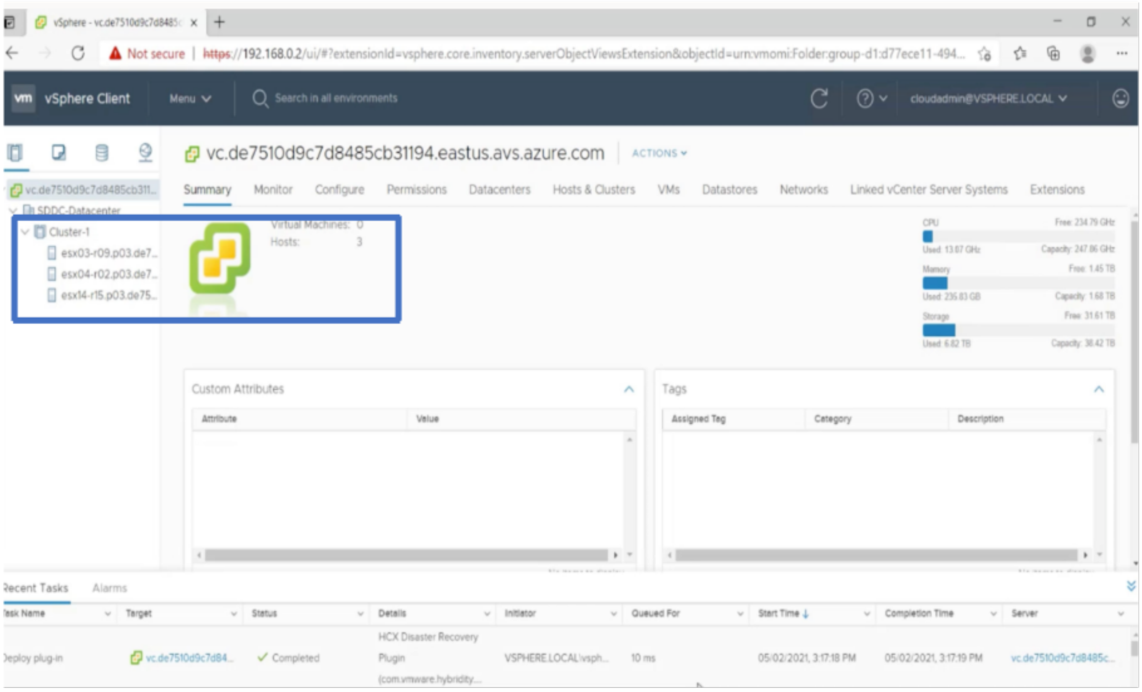

2. Launch the vSphere client by typing the vCenter web client URL.




3. Log in to VMware vSphere using the vCenter credentials of your Azure VMware Solution private cloud.




4. In the vSphere client, you can verify the ESXi hosts that you created in Azure portal.

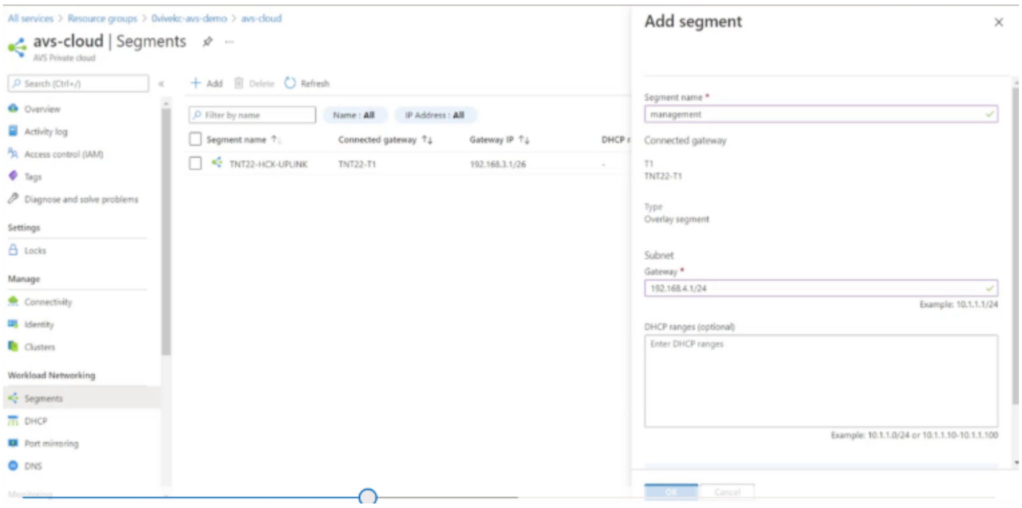



For more information, see Access your Private Cloud vCenter portal.

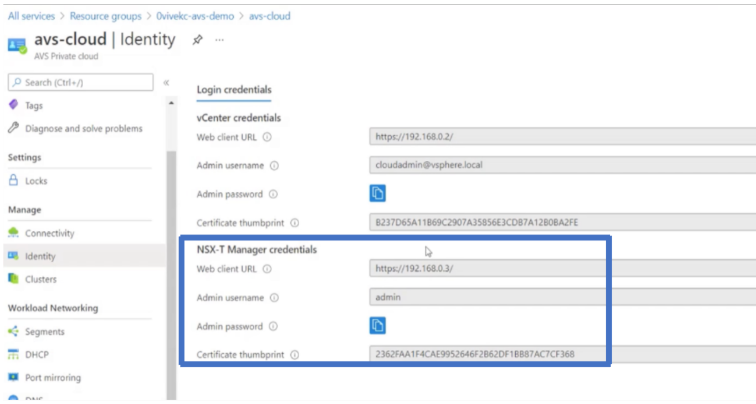
## Create an NSX-T segment in the Azure portal

You can create and configure an NSX-T segment from the Azure VMware Solution console in the Azure portal. These segments are connected to the default Tier-1 gateway, and the workloads on these segments get East-West and North-South connectivity. Once you create the segment, it displays in NSX-T Manager and vCenter.

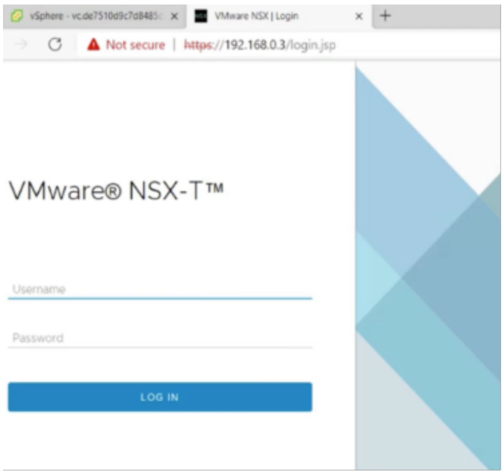
1. In your Azure VMware Solution private cloud, under **Workload Networking**, select **Segments > Add**. Provide the details for the new logical segment and select **OK**. You can create three separate segments for Client, Management, and Server interfaces.



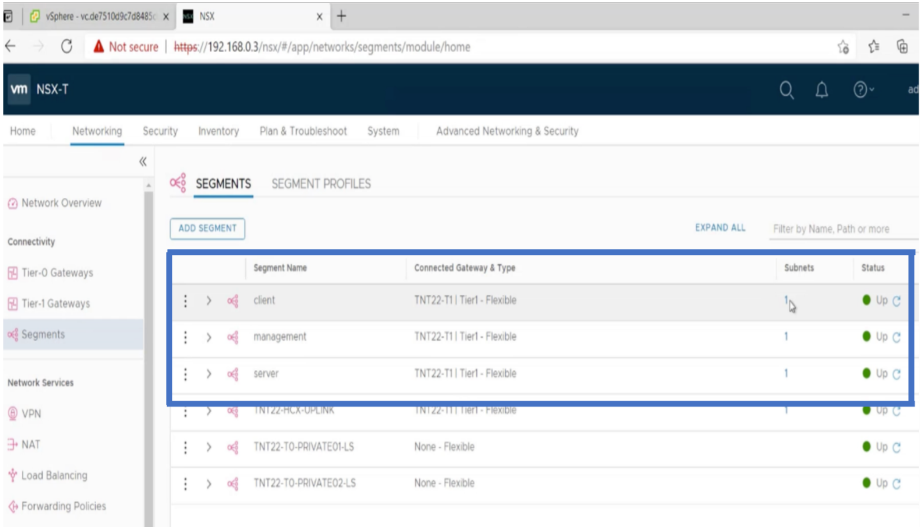

2. In your Azure VMware Solution private cloud, under **Manage**, select **Identity**. Make note of the NSX-T Manager credentials.




3. Launch the VMware NSX-T Manager by typing the NSX-T web client URL.




4. In the NSX-T manager, under **Networking > Segments**, you can see all the segments that you have created. You can also verify the subnets.




For more information, see Create an NSX-T segment in the Azure portal.

## Install a NetScaler VPX instance on VMware cloud

After you have installed and configured VMware Software-Defined Data Center (SDDC), you can use the SDDC to install virtual appliances on the VMware cloud. The number of virtual appliances that you can install depends on the amount of memory available on the SDDC.

To install NetScaler VPX instances on VMware cloud, perform these steps in Windows Jumpbox VM:

1. Download the NetScaler VPX instance setup files for ESXi host from the NetScaler downloads site.

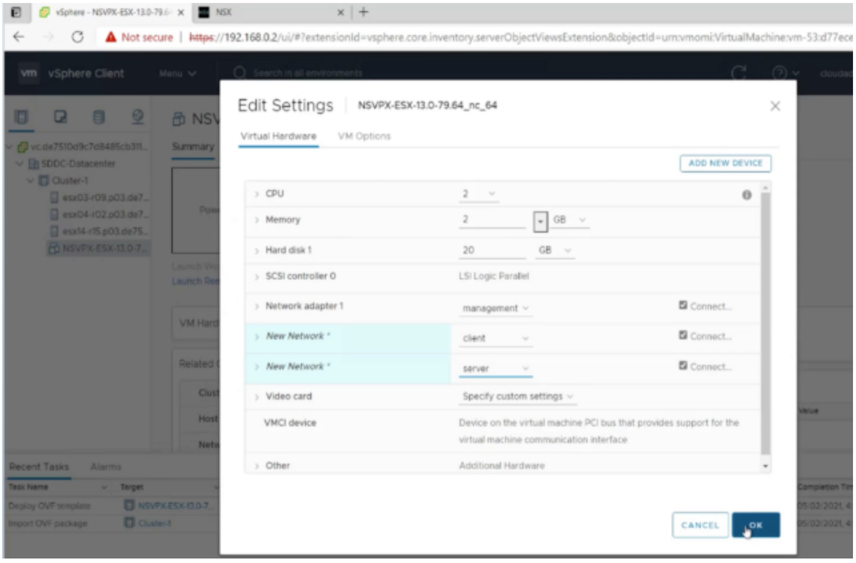
2. Open VMware SDDC in the Windows Jumpbox.

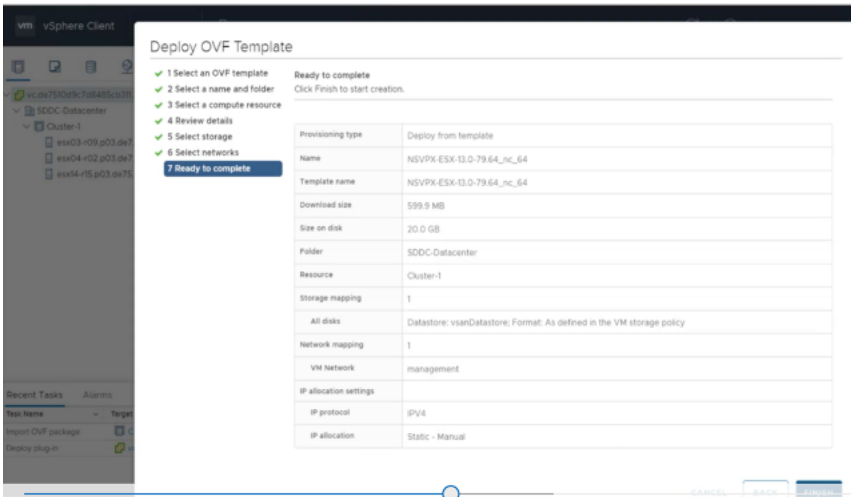
3. In the **User Name** and **Password** fields, type the administrator credentials, and then click **Login**.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file** field, browse to the location at which you saved the NetScaler VPX instance setup files, select the .ovf file, and click **Next**.

   > **Note:**
   >
   > By default, the NetScaler VPX instance uses E1000 network interfaces. To deploy ADC with the VMXNET3 interface, modify the OVF to use VMXNET3 interface instead of E1000. Availability of VMXNET3 interface is limited by Azure infrastructure and might not be available in Azure VMware Solution.

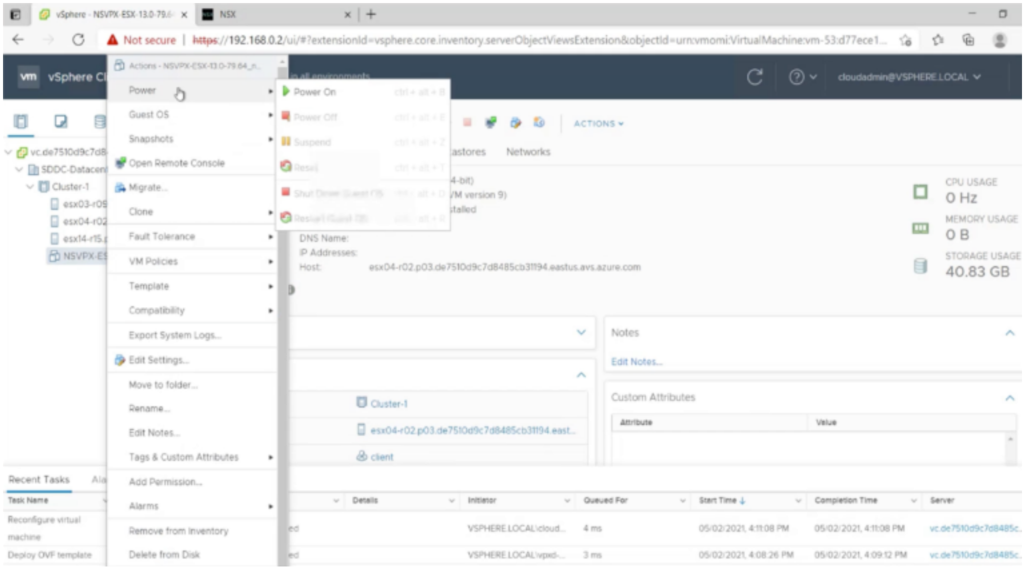
6. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the VMware SDDC. Click **OK**.


7. Click **Finish** to start installing a virtual appliance on VMware SDDC.

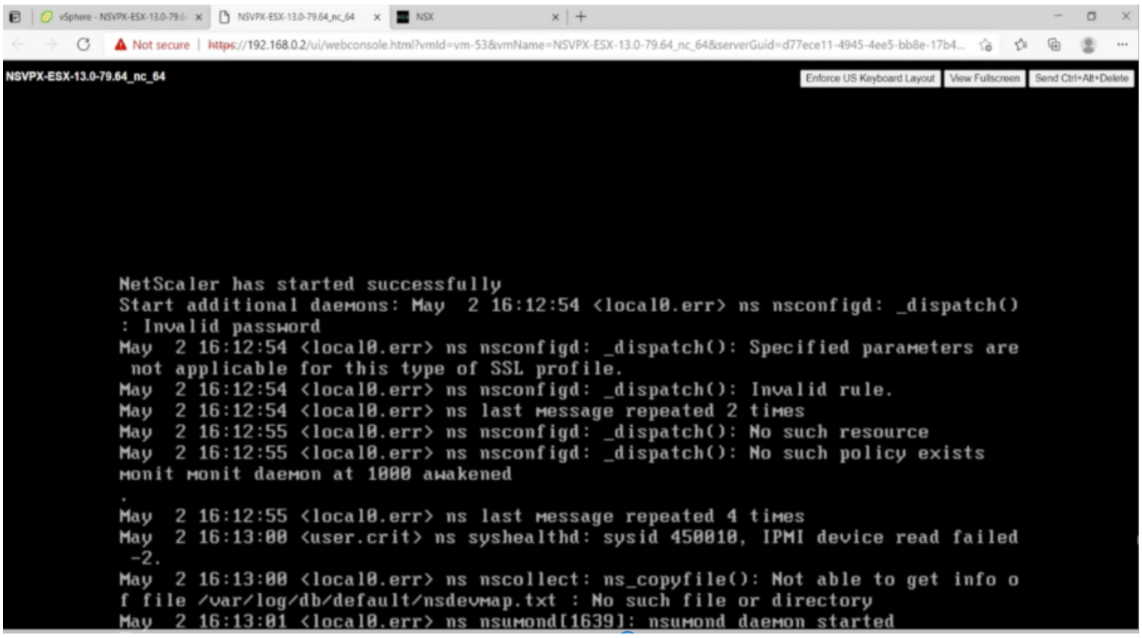



8. You are now ready to start the NetScaler VPX instance. In the navigation pane, select the NetScaler VPX instance that you have installed and, from the right-click menu, select **Power On**. Click the **Console** tab to emulate a console port.




9. You are now connected to the NetScaler VM from the vSphere client.



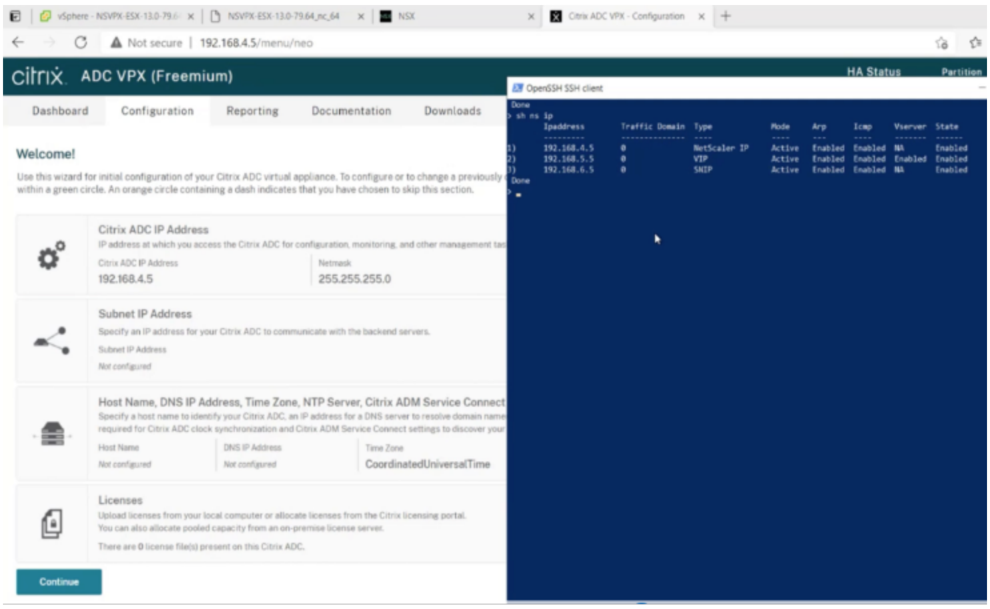

10. To access the NetScaler appliance by using the SSH keys, type the following command in the CLI:

```
ssh nsroot@<management IP address>
```

**Example:**

```
ssh nsroot@192.168.4.5
```

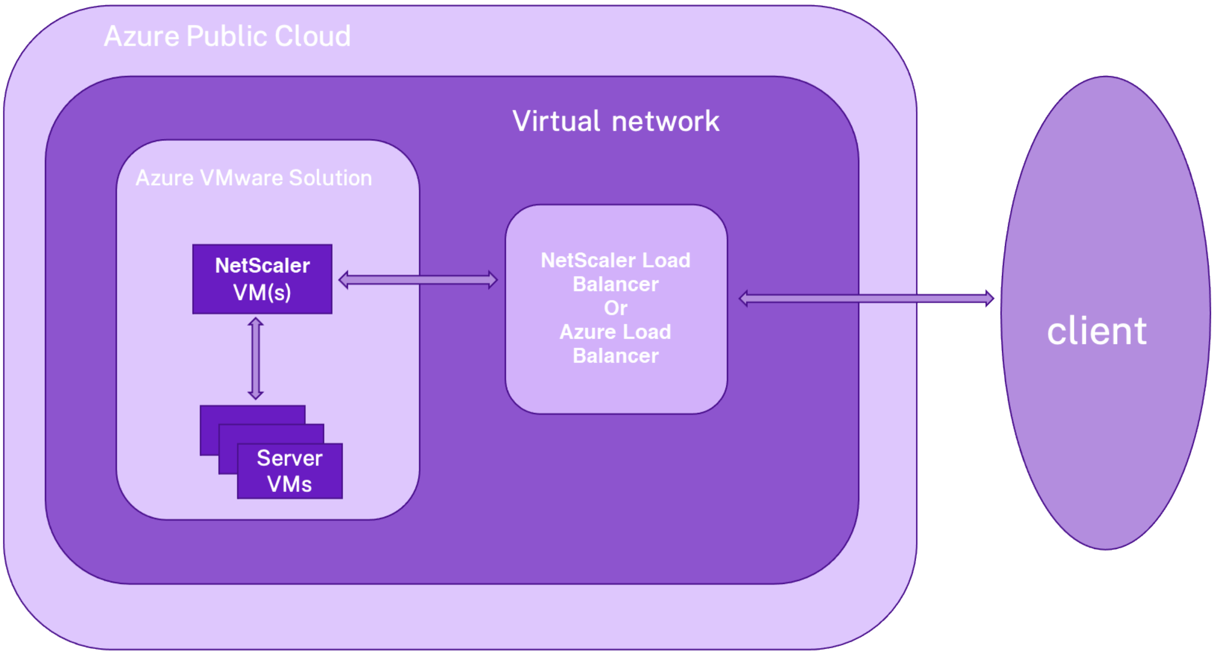
11. You can verify the ADC configuration by using the `show ns ip` command.



## Configure a NetScaler VPX standalone instance on Azure VMware solution

You can configure a NetScaler VPX standalone instance on Azure VMware solution (AVS) for internet facing applications.

The following diagram shows the NetScaler VPX standalone instance on Azure VMware Solution. A client can access the AVS service by connecting to the virtual IP (VIP) address of NetScaler inside the AVS. You can achieve this by provisioning a NetScaler load balancer or the Azure load balancer instance outside AVS but in the same Azure virtual network. Configure the load balancer to access the VIP of the NetScaler VPX instance within AVS service.




### Prerequisites

Before you begin installing a virtual appliance, read the following Azure prerequisites:

- For more information on Azure VMware solution and its prerequisites, see Azure VMware Solution documentation.
- For more information on deploying Azure VMware solution, see Deploy an Azure VMware Solution private cloud.
- For more information on creating a Windows Jump box VM to access and manage Azure VMware Solution, see Access an Azure VMware Solution private cloud.
- In Windows Jump box VM, download the NetScaler VPX appliance setup files.
- Create appropriate NSX-T network segments on VMware SDDC to which the virtual machines connect. For more information, see Add a network segment in Azure VMware Solution

- For more information on how to install a NetScaler VPX instance on VMware cloud, see Install a NetScaler VPX instance on VMware cloud.

### Configure a NetScaler VPX standalone instance on AVS using the NetScaler load balancer

Follow these steps to configure the NetScaler VPX standalone instance on AVS for internet facing applications using the NetScaler load balancer.

1. Deploy a NetScaler VPX instance on the Azure cloud. For more information, see Configure a NetScaler VPX standalone instance.

   > **Note:**
   >
   > Ensure that it is deployed on the same virtual network as the Azure VMware Cloud.

2. Configure the NetScaler VPX instance to access the VIP address of NetScaler VPX deployed on AVS.

   a) Add a load balancing virtual server.

   ```
   add lb vserver <name> <serviceType> [<vip>] [<port>]
   ```

   **Example:**

   ```
   add lb vserver lb1 HTTPS 172.31.0.6 443
   ```

   b) Add a service that connects to the VIP of NetScaler VPX deployed on AVS.

   ```
   add service <name> <ip> <serviceType> <port>
   ```

   **Example:**

   ```
   add service webserver1 192.168.4.10 HTTP 80
   ```

   c) Bind a service to the load balancing virtual server.

   ```
   bind lb vserver <name> <serviceName>
   ```

   **Example:**

   ```
   bind lb vserver lb1 webserver1
   ```

### Configure NetScaler VPX standalone instance on AVS using the Azure load balancer

Follow these steps to configure the NetScaler VPX standalone instance on AVS for internet facing applications using the Azure load balancer.

1. Configure an Azure Load Balancer instance on Azure cloud. For more information, see Azure documentation on creating load balancer.
2. Add the VIP address of the NetScaler VPX instance that is deployed on AVS to the back-end pool.

   The following Azure command adds one back-end IP address into the load balance back-end address pool.

   ```
   az network lb address-pool address add
                                       --resource-group <Azure VMC
                                          Resource Group>
                                       --lb-name <LB Name>
                                       --pool-name <Backend pool name
                                          >
                                       --vnet <Azure VMC Vnet>
                                       --name <IP Address name>
                                       --ip-address <VIP of ADC in
                                          VMC>
   ```

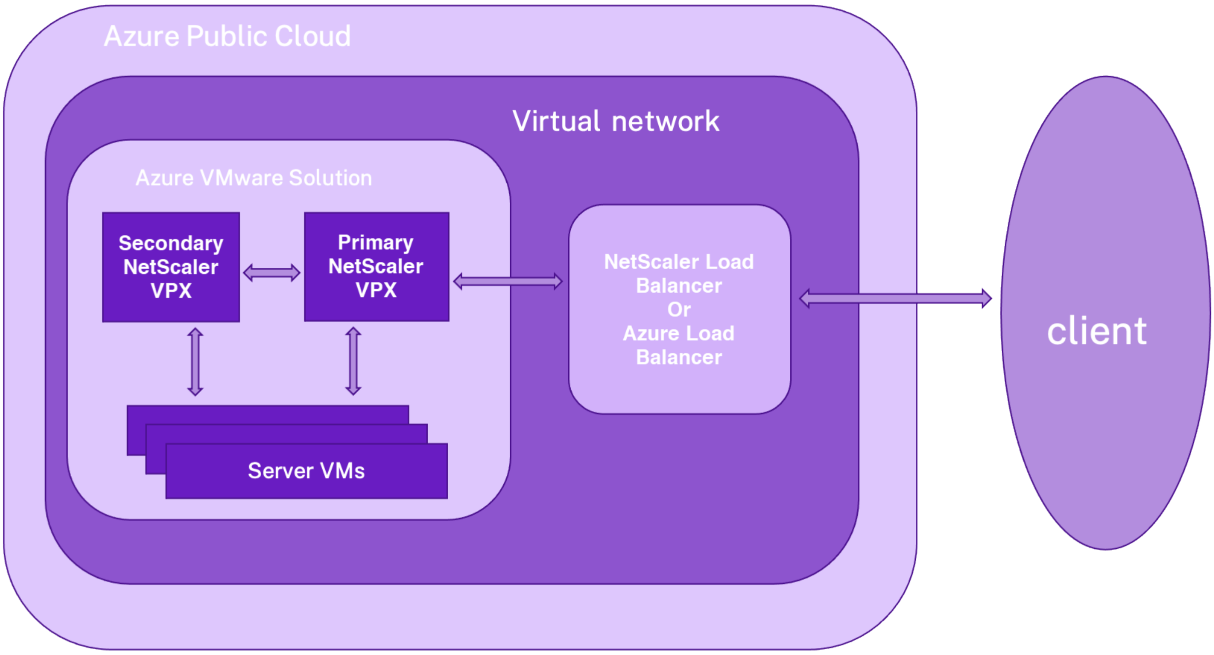
   > **Note:**
   >
   > Ensure that the Azure load balancer is deployed in the same virtual network as the Azure VMware cloud.

## Configure a NetScaler VPX high availability setup on Azure VMware solution

You can configure a NetScaler VPX HA setup on Azure VMware solution (AVS) for internet facing applications.

The following diagram shows the NetScaler VPX HA pair on AVS. A client can access the AVS service by connecting to the VIP of the primary ADC node inside the AVS. You can achieve this by provisioning a NetScaler load balancer or the Azure load balancer instance outside AVS but in the same Azure virtual network. Configure the load balancer to access the VIP of primary ADC node within AVS service.




### Prerequisites

Before you begin installing a virtual appliance, read the following Azure prerequisites:

- For more information on Azure VMware solution and its prerequisites, see Azure VMware Solution documentation.
- For more information on deploying Azure VMware solution, see Deploy an Azure VMware Solution private cloud.
- For more information on creating a Windows Jump box VM to access and manage Azure VMware Solution, see Access an Azure VMware Solution private cloud.
- In Windows Jump box VM, download the NetScaler VPX appliance setup files.
- Create appropriate NSX-T network segments on VMware SDDC to which the virtual machines connect. For more information, see Add a network segment in Azure VMware Solution.

### Configuration steps

Follow these steps to configure the NetScaler VPX high availability setup in AVS for internet facing applications.

1. Create two NetScaler VPX instances on VMware cloud. For more information, see Install a NetScaler VPX instance on VMware cloud.

2. Configure the NetScaler HA setup. For more information, see Configuring high availability.

3. Configure the NetScaler HA setup to be accessible for internet facing applications.

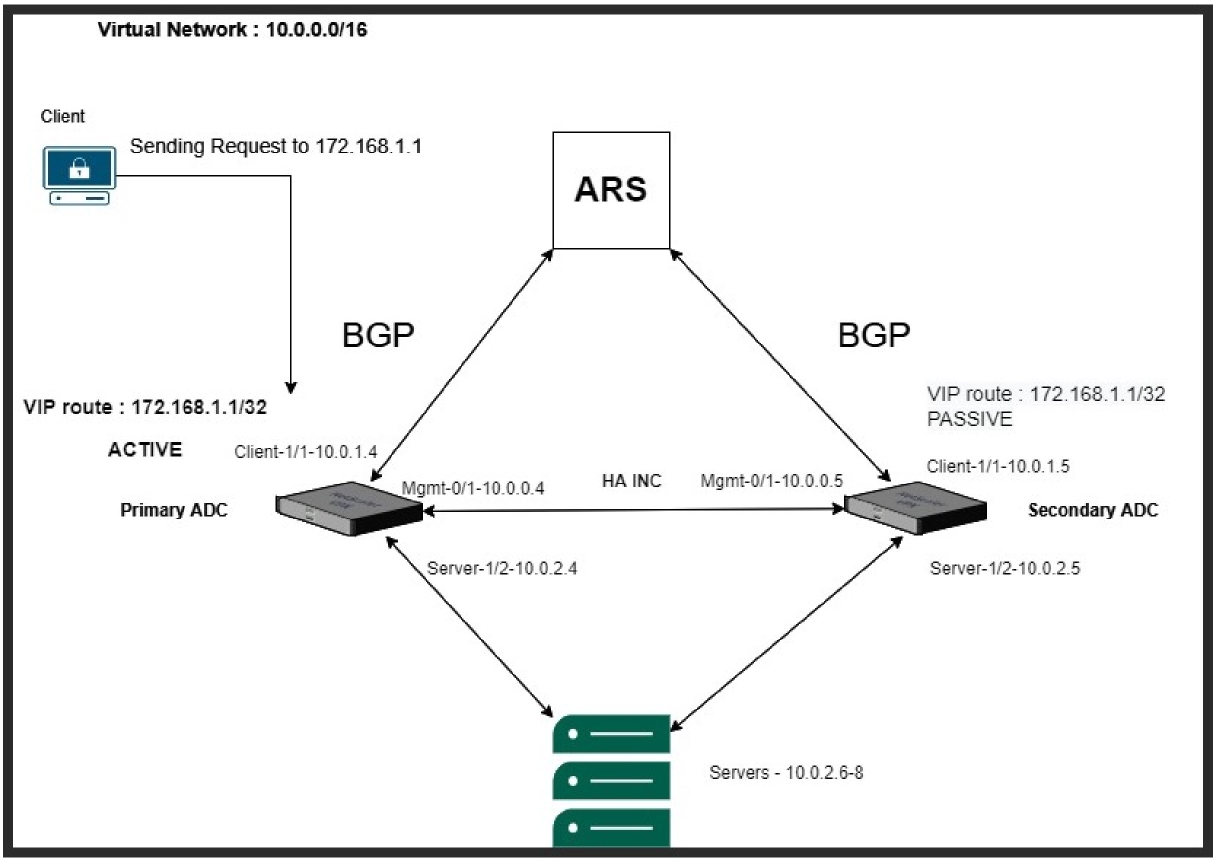
 

- To configure the NetScaler VPX instance using the NetScaler load balancer, see Configure a NetScaler VPX standalone instance on AVS using the NetScaler load balancer.
- To configure the NetScaler VPX instance using the Azure load balancer, see Configure NetScaler VPX standalone instance on AVS using the Azure load balancer.

## Configure Azure route server with NetScaler VPX HA pair

You can configure Azure route server with NetScaler VPX instance to exchange the VIP routes configured with virtual network using the BGP protocol. The NetScaler can be deployed in standalone or in HA-INC mode, and then configured with BGP. This deployment doesn't require an Azure load balancer (ALB) in front of the ADC HA pair.

The following diagram depicts how a VPX HA topology is integrated with the Azure route server. Each of the ADC instances has 3 interfaces: one for management, one for client traffic, and one for server traffic.




The topology diagram uses the following IP addresses.

**Sample IP configuration for primary ADC instance:**

```
NSIP: 10.0.0.4/24
SNIP on 1/1: 10.0.1.4/24
SNIP on 1/2: 10.0.2.4/24
VIP: 172.168.1.1/32
```

**Sample IP configuration for secondary ADC instance:**

```
NSIP: 10.0.0.5/24
SNIP on 1/1: 10.0.1.5/24
SNIP on 1/2: 10.0.2.5/24
VIP: 172.168.1.1/32
```

## Prerequisites

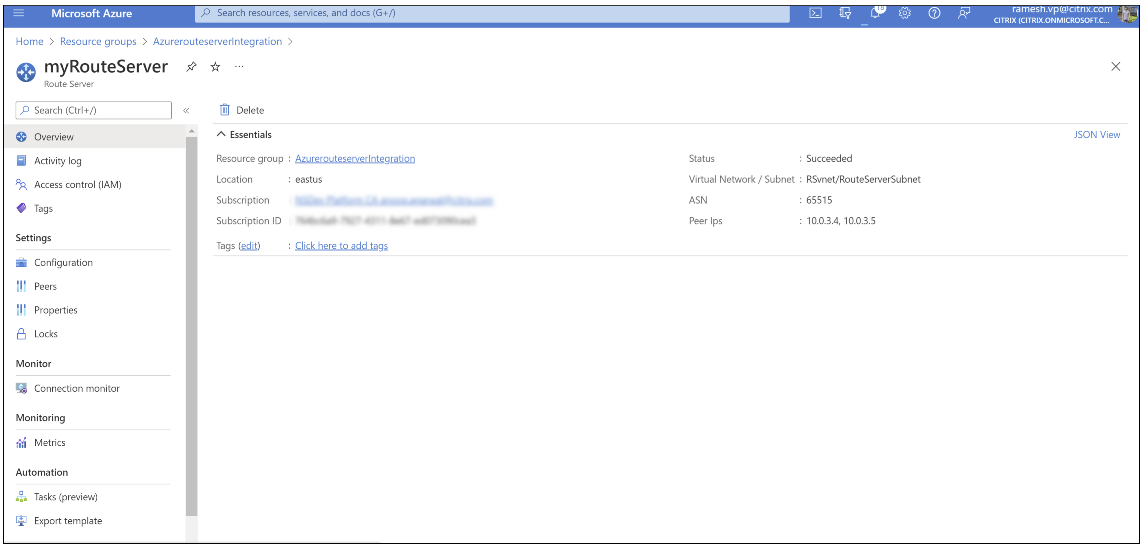
You must be familiar with the following information before deploying a NetScaler VPX instance on Azure.

- Azure terminology and network details. For more information, see Azure terminology.
- Overview of Azure Route Server. For more information, see What is Azure Route Server?.
- Working of a NetScaler appliance. For more information, see NetScaler documentation.
- NetScaler networking. For more information, see the ADC Networking.

## How to configure an Azure route server with NetScaler VPX HA pair

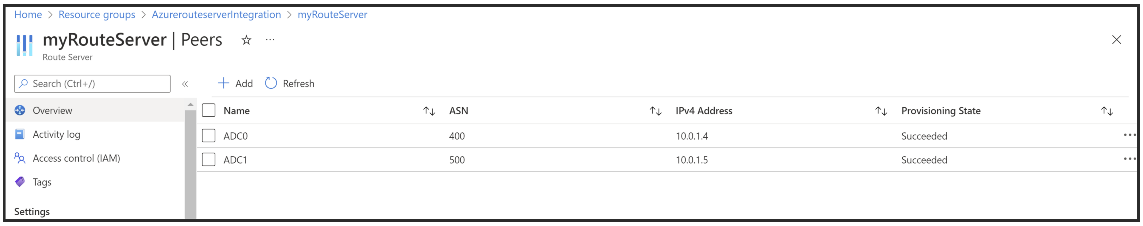
1. Create a route server in the Azure portal. For more information, see Create and configure a Route Server using the Azure portal.

   In the following example, subnet 10.0.3.0/24 is used for deploying Azure server. Once the route server is created, get the route server IP addresses, for example: 10.0.3.4, 10.0.3.5.

2. Set up peering with network virtual appliance (NVA) in the Azure portal. Add your NetScaler VPX instance as the NVA. For more information, see Set up peering with NVA.

   In the following example, the ADC SNIP on 1/1 interfaces: 10.0.1.4 and 10.0.1.5, and the ASN: 400 and 500, are used while adding the peer.




3. Add two NetScaler VPX instances for the HA configuration.

   Complete the following steps:

   a) Deploy two VPX instances (primary and secondary instances) on Azure.
   b) Add client and server NIC on both the instances.
   c) Configure HA settings on both instances by using the NetScaler GUI.

4. Configure dynamic routing in the primary ADC instance.

   **Sample configuration:**

```
enable ns mode L3 MBF USNIP SRADV DRADV  PMTUD
enable ns feature LB BGP
add ns ip 10.0.1.4 255.255.255.0 -vServer DISABLED -dynamicRouting ENABLED
VTYSH
configure terminal
router BGP 400
timers bgp 1 3
neighbor 10.0.3.4 remote-as 65515
neighbor 10.0.3.4 advertisement-interval 3
neighbor 10.0.3.4 fall-over  bfd
neighbor 10.0.3.5 remote-as 65515
neighbor 10.0.3.5 advertisement-interval 3
neighbor 10.0.3.5 fall-over  bfd
address-family ipv4
redistribute kernel
redistribute static
```

5. Configure dynamic routing in the secondary ADC instance.

   **Sample configuration:**

```
enable ns mode L3 MBF USNIP SRADV DRADV  PMTUD
enable ns feature LB BGP
add ns ip 10.0.1.5 255.255.255.0 -vServer DISABLED -dynamicRouting ENABLED
VTYSH
```

```
configure terminal
router BGP 500
timers bgp 1 3
neighbor 10.0.3.4 remote-as 65515
neighbor 10.0.3.4 advertisement-interval 3
neighbor 10.0.3.4 fall-over  bfd
neighbor 10.0.3.5 remote-as 65515
neighbor 10.0.3.5 advertisement-interval 3
neighbor 10.0.3.5 fall-over  bfd
address-family ipv4
redistribute kernel
redistribute static
```

6. Verify the BGP peers established using the BGP commands in the VTY shell interface. For more information, see Verifying the BGP Configuration.

```
show ip bgp neighbors
```

7. Configure LB virtual server in the primary ADC instance.

**Sample configuration:**

```
add ns ip 172.16.1.1 255.255.255.255 -type VIP -hostRoute ENABLED
add lbvserver v1 HTTP 172.16.1.1 80
add service s1 10.0.2.6 HTTP 80
bind lbvserver v1 s1
enable ns feature lb
```

A client in the same virtual network as of the NetScaler VPX instance can now access the LB virtual server. In this case, the NetScaler VPX instance advertises the VIP route to the Azure route server.

## Add Azure Autoscale settings

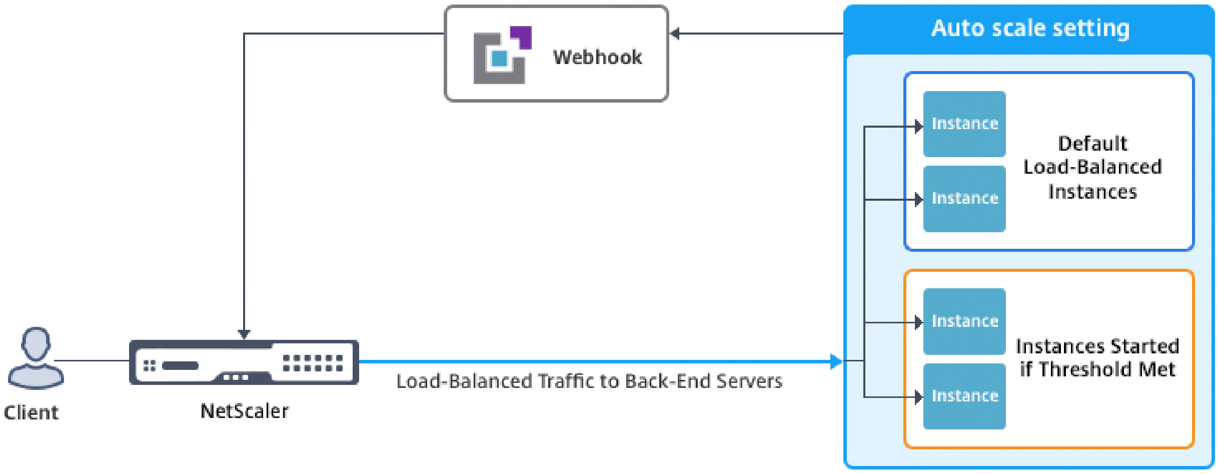
Efficient hosting of applications in a cloud involves easy and cost-effective management of resources depending on the application demand. To meet increasing demand, you have to scale network resources upward. Whether demand subsides, you must scale down to avoid the unnecessary cost of idle resources. To minimize the cost of running the application, you have to constantly monitor traffic, memory and CPU use, and so on. However, monitoring traffic manually is cumbersome. For the application environment to scale up or down dynamically, you must automate the processes of monitoring traffic and of scaling resources up and down whenever necessary.

You can use Autoscale with Azure virtual machine scale sets (VMSS) for VPX multi-IP standalone and high availability deployment on Azure.

Integrated with the Azure VMSS and Autoscale feature, the NetScaler VPX instance provides the following advantages:

- Load balance and management: Auto configures servers to scale up and scale down, depending on demand. The NetScaler VPX instance auto detects the VMSS Autoscale setting in the same virtual network where the VPX instance is deployed, or the peered virtual networks that are in the same Azure subscription. You can select the VMSS Autoscale setting to balance the load. This is done by auto configuring NetScaler virtual IP address and subnet IP address on the VPX instance.
- High availability: Detects Autoscale groups and load balances servers.
- Better network availability: The VPX instance supports back-end servers on different virtual networks (VNets).




For more information, see the following Azure topic

- Virtual Machine Scale Sets Documentation
- Overview of Autoscale in Microsoft Azure Virtual Machines, Cloud Services, and Web Apps

### Before you begin

1. Read Azure-related usage guidelines. For more information, see Deploy a NetScaler VPX instance on Microsoft Azure.
2. Create one or more NetScaler VPX instances with three network interfaces on Azure according to your requirement (standalone or high availability deployment).
3. Open the TCP 9001 port on the network security group of the 0/1 interface of the VPX instance. The VPX instance uses this port to receive the scale-out and scale-in notification.
4. Create an Azure VMSS in the same virtual network, where the NetScaler VPX instance is deployed. If the VMSS and NetScaler VPX instance are deployed in different Azure virtual networks, the following conditions have to be met:

    - Both the virtual networks must be in the same Azure subscription.

- The two virtual networks must be connected using the virtual network peering feature of Azure.

If you don't have an existing VMSS configuration, complete the following tasks:

a) Create a VMSS

b) Enable Autoscale on VMSS

c) Create scale-in and scale-out policy in VMSS Autoscale setting

For more information, see Overview of Autoscale with Azure virtual machine scale sets.

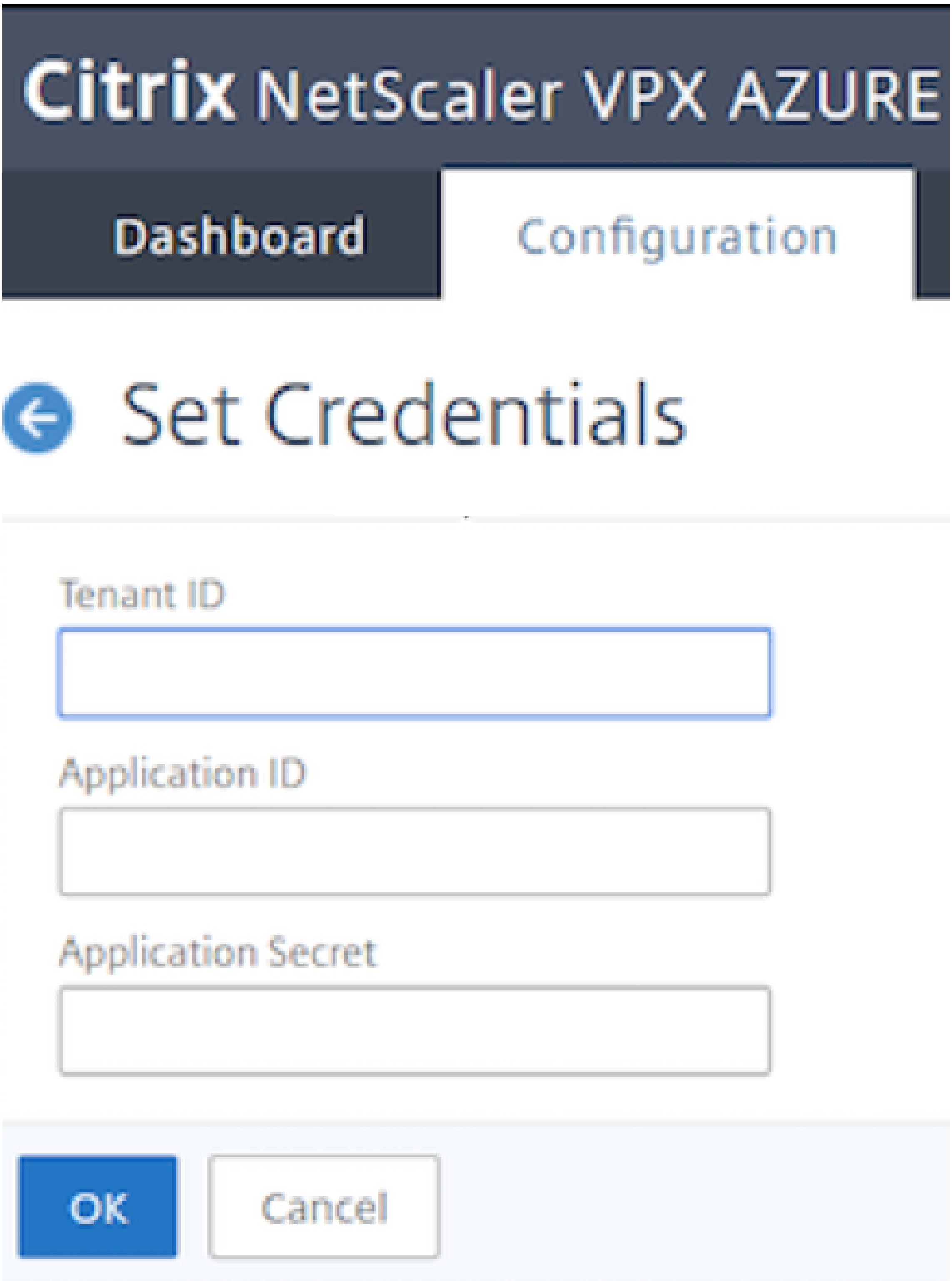
5. Create an Azure Active Directory (ADD) application and service principal that can access resources. Assign contributor role to the newly created AAD application. For more information, see Use portal to create an Azure Active Directory application and service principal that can access resources.

### Add VMSS to a NetScaler VPX instance

You can add the Autoscale setting to a VPX instance with a single click by using the GUI. Complete these steps to add the Autoscale setting to the VPX instance:

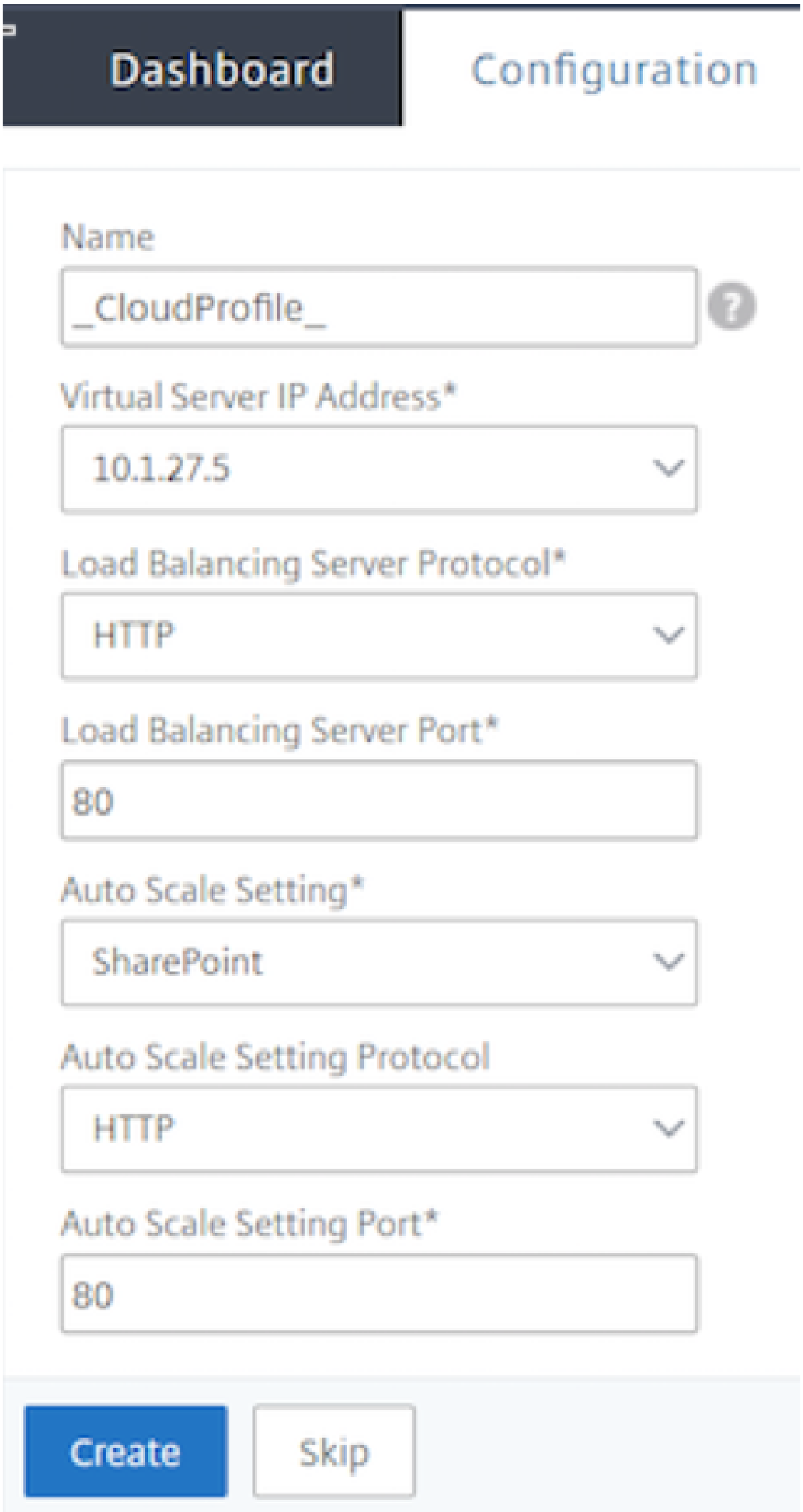
1. Log on to the VPX instance.
2. When you log on to the NetScaler VPX instance for the first time, you see the Set Credentials page. Add the required Azure credentials for the Autoscale feature to work.




3. In the default cloud profile page, enter the details, as shown in the following example, and click Create.

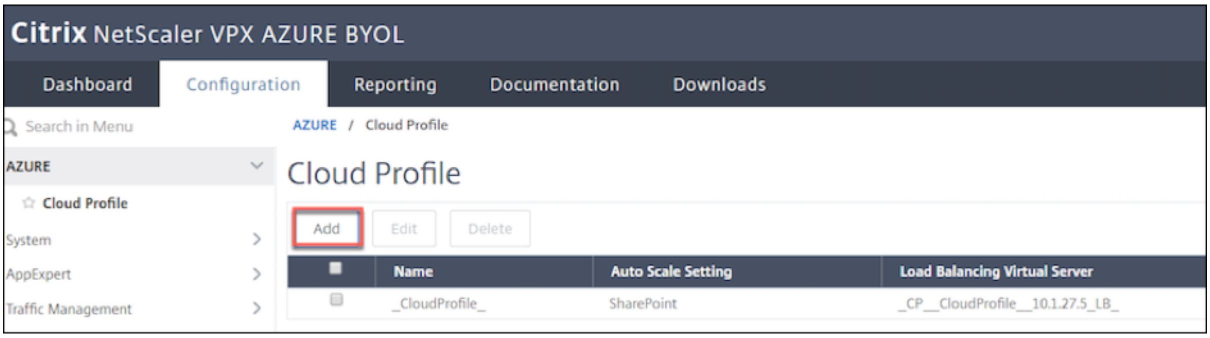

Dashboard
Configuration
Name
_CloudProfile_
Virtual Server IP Address*
10.1.27.5
Load Balancing Server Protocol*
HTTP
Load Balancing Server Port*
80
Auto Scale Setting*
SharePoint
Auto Scale Setting Protocol
HTTP
Auto Scale Setting Port*
80
Create
Skip

#### Points to keep in mind while creating a cloud profile

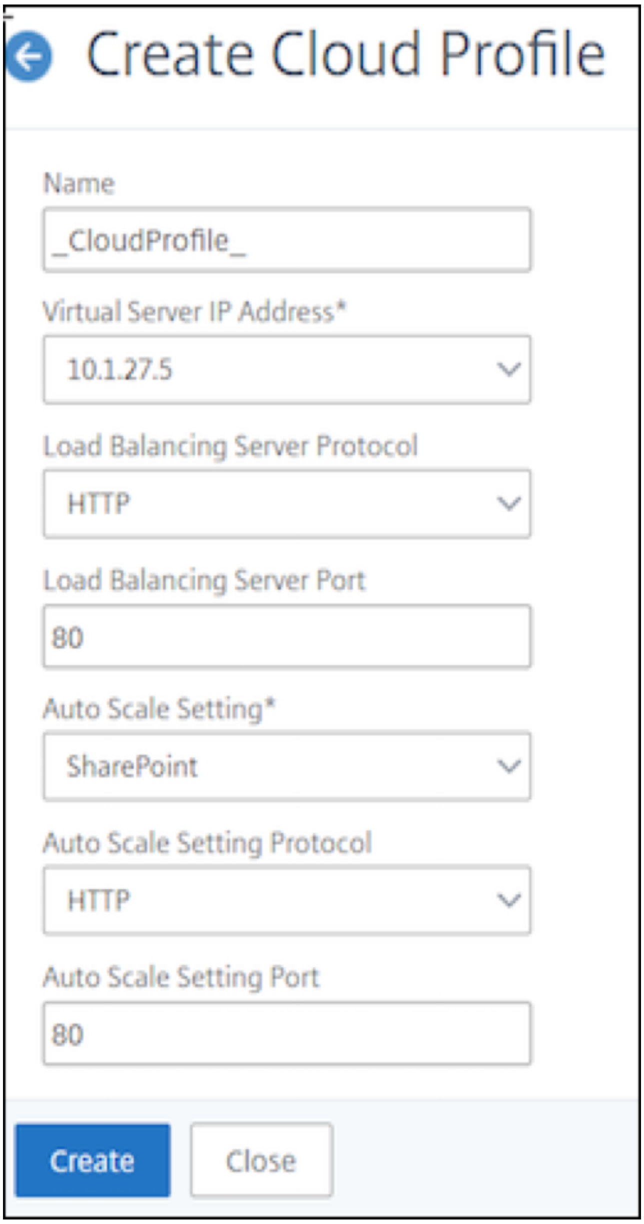
- The virtual server IP address is auto-populated from the free IP address available to the VPX instance. For more information, see Assign multiple IP addresses to virtual machines using the Azure portal.
- Autoscale setting is prepopulated from the VMSS instance that is connected to the NetScaler VPX instance either in the same virtual network or peered virtual networks. For more information, see Overview of Autoscale with Azure virtual machine scale sets.
- While selecting the Auto Scaling Group protocol and port, ensure that your servers listen on the protocols and ports, and you bind the correct monitor in the service group. By default, the TCP monitor is used.
- For SSL Protocol type Autos Scaling, after you create the Cloud Profile, the load balance virtual server or service group is down because of a missing certificate. You can bind the certificate to the virtual server or service group manually.

After the first-time logon, if you want to create a cloud profile, on the GUI go to **System > Azure > Cloud Profile** and click **Add**.




The Create Cloud Profile configuration page appears.




Cloud Profile creates a NetScaler load balancing virtual server and a service group with members (servers) as the servers of the Auto Scaling Group. Your back-end servers must be reachable through the SNIP configured on the VPX instance.

> **Note:**
>
> From NetScaler release 13.1-42.x onwards, you can create different cloud profiles for different services (using different ports) with the same VMSS in Azure. Thus, the NetScaler VPX instance supports multiple services with the same Autoscaling group in public cloud.

To view autoscale-related information in the Azure portal, go to **All service > Virtual machine scale set > Select Virtual machine scale set > Scaling**.

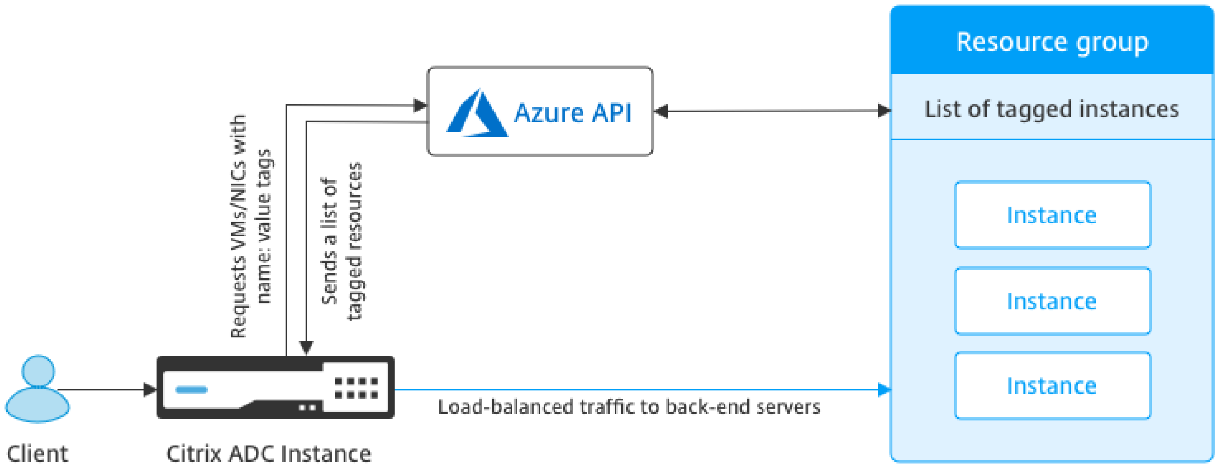
## Azure tags for NetScaler VPX deployment

In the Azure cloud portal, you can tag resources with a name: value pair (such as Dept: Finance) to categorize and view resources across resource groups and, within the portal, across subscriptions. Tagging is helpful when you need to organize resources for billing or management or automation.

### How Azure tag works for VPX deployment

For NetScaler VPX standalone and high-availability instances deployed on Azure Cloud, now you can create load balancing service groups associated with an Azure tag. The VPX instance constantly monitors Azure virtual machines (back-end servers) and network interfaces (NICs), or both, with the respective tag and updates the service group accordingly.

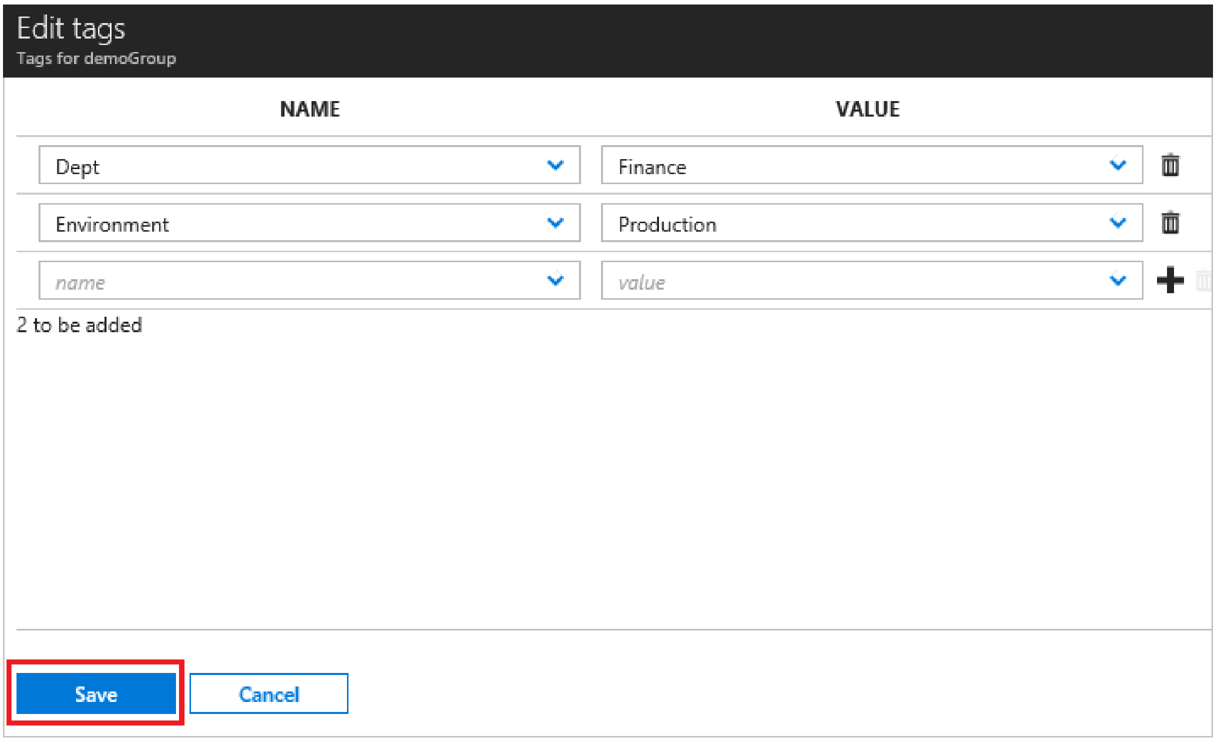
The VPX instance creates the service group that load balances the back-end servers using tags. The instance queries the Azure API for all resources that are tagged with a particular tag name and tag value. Depending on the assigned poll period (by default 60 seconds), the VPX instance periodically polls the Azure API and retrieves the resources available with the tag name and tag values assigned in the VPX GUI. Whenever a VM or NIC with the appropriate tag is added or deleted, the ADC detects the respective change and adds or deletes the VM or NIC IP address from the service group automatically.




### Before you begin

Before creating NetScaler load balancing service groups, add a tag to the servers in Azure. You can assign the tag to either the virtual machine or to NIC.




For more information about adding Azure tags, see Microsoft document Use tags to organize your Azure resources.

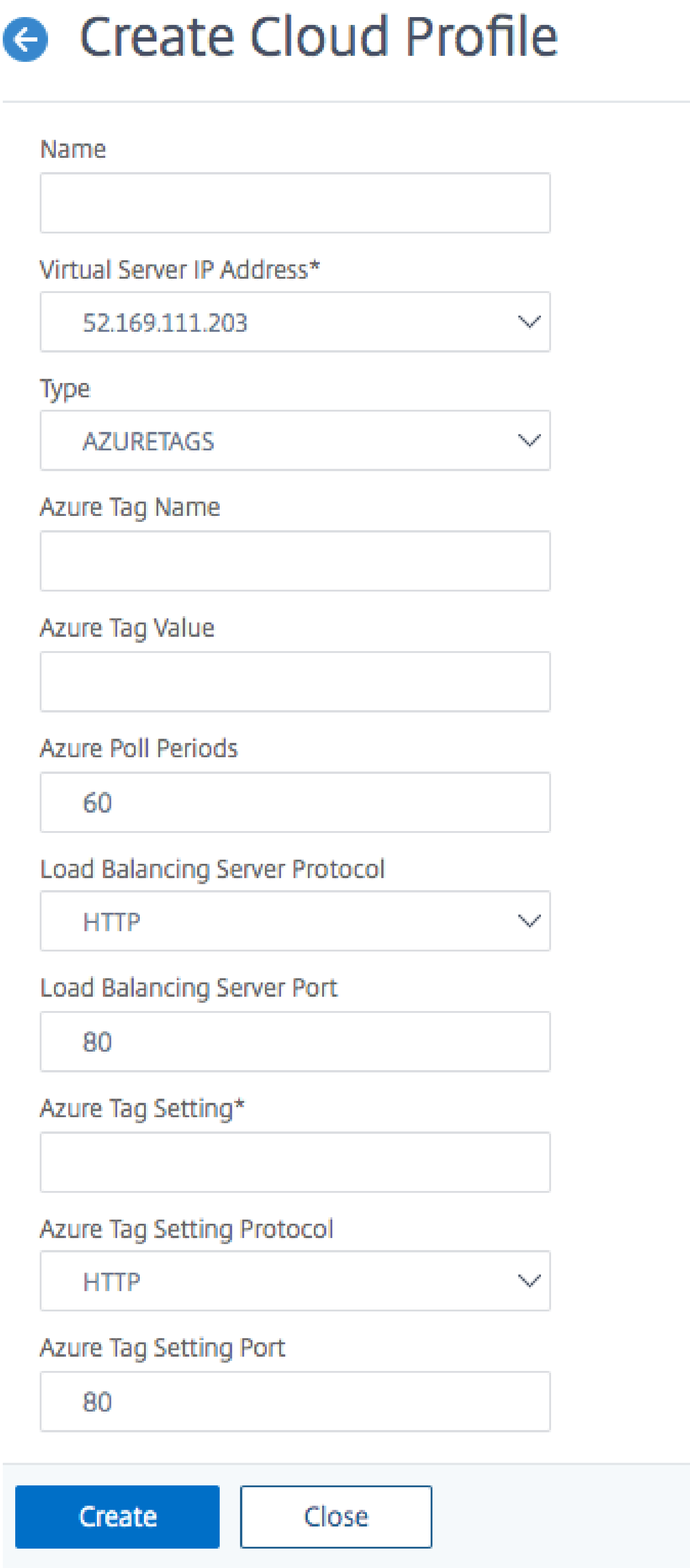
> **Note:**
>
> ADC CLI commands to add Azure tag settings support tag names and tag values that start only with numerals or alphabets and not other keyboard characters.

### How to add Azure tag settings by using VPX GUI

You can add the Azure tag cloud profile to a VPX instance by using the VPX GUI so that the instance can load balance the back-end servers using the specified tag. Follow these steps:

1. From the VPX GUI, go to **Configuration > Azure > Cloud Profile**.
2. Click Add to create a cloud profile. The cloud profile window opens.


Create Cloud Profile
Name
Virtual Server IP Address*
52.169.111.203
Type
AZURETAGS
Azure Tag Name
Azure Tag Value
Azure Poll Periods
60
Load Balancing Server Protocol
HTTP
Load Balancing Server Port
80
Azure Tag Setting*
Azure Tag Setting Protocol
HTTP
Azure Tag Setting Port
80
Create
Close

1. Enter values for the following fields:

    - Name: Add a name for your profile
    - Virtual Server IP Address: The virtual server IP address is auto-populated from the free IP address available to the VPX instance. For more information, see [Assign multiple IP addresses to virtual machines using the Azure portal].
    - Type: From the menu, select AZURETAGS.
    - Azure Tag Name: Enter the name that you have assigned to the VMs or NICs in the Azure portal.
    - Azure Tag Value: Enter the value that you have assigned to the VMs or NICs in Azure portal.
    - Azure Poll Periods: By default the poll period is 60 seconds, which is the minimum value. You can change it according to your requirement.
    - Load Balancing Server Protocol: Select the protocol that your load balancer listens on.
    - Load Balancing Server Port: Select the port that your load balancer listens on.
    - Azure tag setting: The name of the service group that will be created for this cloud profile.
    - Azure Tag Setting Protocol: Select the protocol that your back-end servers listen on.
    - Azure Tag Setting Port: Select the port that your back-end servers listen on.

2. Click **Create**.

A load-balancer virtual server and a service group are created for the tagged VMs or NICs. To see the load balancer virtual server, from the VPX GUI, navigate to **Traffic Management > Load Balancing > Virtual Servers**.

### How to add Azure tag settings by using VPX CLI

Type the following command on NetScaler CLI to create a cloud profile for Azure tags.

```
add cloud profile `<profile name>` -type azuretags -vServerName `<
    vserver name>` -serviceType HTTP -IPAddress `<vserver IP address>` -
    port 80 -serviceGroupName `<service group name>` -
    boundServiceGroupSvcType HTTP -vsvrbindsvcport 80 -azureTagName `<
    Azure tag specified on Azure portal>` -azureTagValue `<Azure value
    specified on the Azure portal>` -azurePollPeriod 60
```

> **Important:**
>
> You must save all configurations; otherwise, the configurations are lost after you restart the instance. Type `save config`.

**Example 1**: Here's a sample command for a cloud profile for HTTP traffic of all Azure VMs/NICs tagged with the "myTagName/myTagValue" pair:

```
add cloud profile MyTagCloudProfile -type azuretags -vServerName
    MyTagVServer -serviceType HTTP -IPAddress 40.115.116.57 -port 80 -
    serviceGroupName MyTagsServiceGroup -boundServiceGroupSvcType HTTP -
    vsvrbindsvcport 80 -azureTagName myTagName -azureTagValue myTagValue
     -azurePollPeriod 60
Done
```

To display the cloud profile, type `show cloudprofile`.

**Example 2**: The following CLI command prints information about the newly added cloud profile in example 1.

```
show cloudprofile
1)     Name:  MyTagCloudProfile Type:  azuretags         VServerName:
    MyTagVServer ServiceType:  HTTP       IPAddress:  52.178.209.133
    Port:  80                 ServiceGroupName:  MyTagsServiceGroup
    BoundServiceGroupSvcType:  HTTP
       Vsvrbindsvcport:  80     AzureTagName:  myTagName AzureTagValue:
           myTagValue AzurePollPeriod:  60     GraceFul:  NO
          Delay:  60
```

To remove a cloud profile, type rm cloud profile <`cloud profile name`>

**Example 3**: The following command removes the cloud profile created in example 1.

```
> rm cloudprofile MyTagCloudProfile
  Done
```

### Troubleshooting

**Issue**: In very rare cases, the "rm cloud profile"CLI command might fail to remove service group and servers associated with the deleted cloud profile. This happens when the command is issued seconds before the poll period of the cloud profile being deleted elapses.

**Solution**: Manually delete the remaining service groups by entering the following CLI command for each of the remaining service groups:

```
#> rm servicegroup <serviceGroupName>
```

Also remove each of the remain servers by entering the following CLI command for each of the remaining servers:

```
#> rm server <name>
```

**Issue**: If you add an Azure tag setting to a VPX instance by using CLI, the rain_tags process continues to run on an HA pair node after a warm reboot.

**Solution**: Manually terminate the process on the secondary node after a warm reboot. From the CLI of the secondary HA node exit to the shell prompt:

```
#> shell
```

Use the following command to kill the rain_tags process:

```
# PID=`ps -aux | grep rain_tags | awk '{
 print $2 }
 '`; kill -9 $PID
```

**Issue**: Back-end servers might not be reachable and reported as DOWN by the VPX instance, in spite of being healthy.

**Solution**: Make sure that the VPX instance can reach the tagged IP address corresponding to the back-end server. For a tagged NIC, this is the NIC IP address; whereas for a tagged VM, this is the VM's primary IP address. If the VM/NIC resides on a different Azure VNet, make sure that VNet peering is enabled.

## Configure GSLB on NetScaler VPX instances

NetScaler appliances configured for global server load balancing (GSLB) provide disaster recovery and continuous availability of applications by protecting against points of failure in a WAN. GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers if there is an outage.

This section describes how to enable GSLB on VPX instances on two sites in a Microsoft Azure environment, by using Windows PowerShell commands.

> **Note:**
>
> For more information about GSLB, see Global Server Load Balancing.

You can configure GSLB on a NetScaler VPX instance on Azure, in two steps:

1. Create a VPX instance with multiple NICs and multiple IP addresses, on each site.
2. Enable GSLB on the VPX instances.

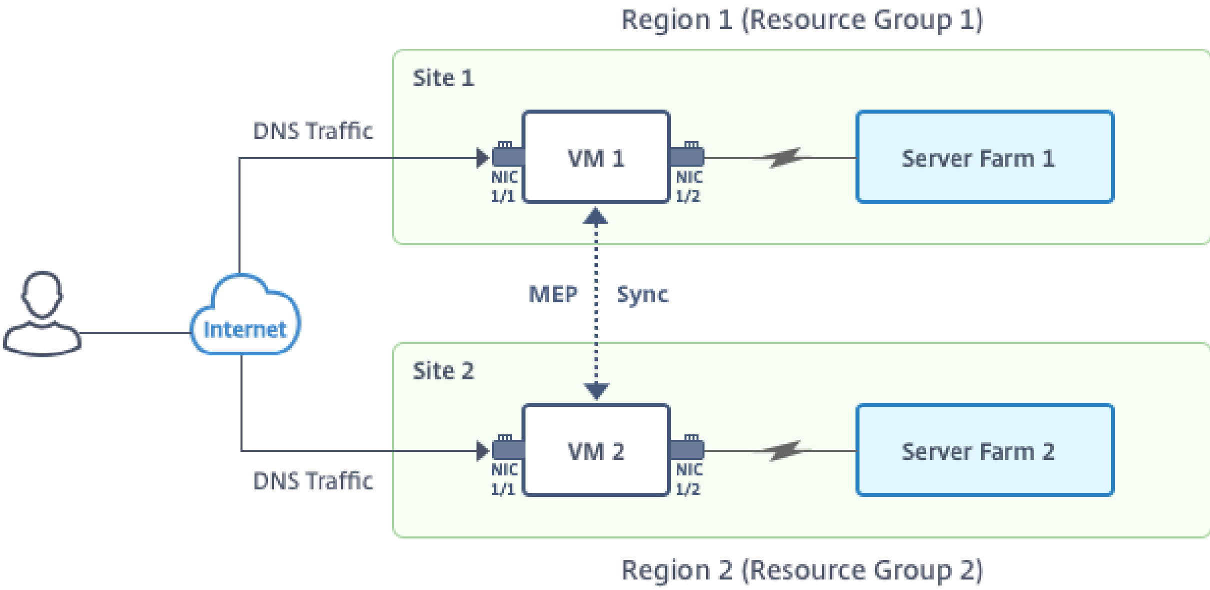
> **Note:**
>
> For more information about configuring multiple NICs and IP addresses see: Configure multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands

### Scenario

This scenario includes two sites - Site 1 and Site 2. Each site has a VM (VM1 and VM2) configured with multiple NICs, multiple IP addresses, and GSLB.

**Figure.** GSLB setup implemented across two sites - Site 1 and Site 2.




In this scenario, each VM has three NICs - NIC 0/1, 1/1, and 1/2. Each NIC can have multiple private and public IP addresses. The NICs are configured for the following purposes.

- NIC 0/1: to serve management traffic
- NIC 1/1: to serve client-side traffic
- NIC 1/2: to communicate with back-end servers

For information about the IP addresses configured on each NIC in this scenario, see the IP configuration details section.

## Parameters

Following are sample parameters settings for this scenario in this document.

```
$location="West Central US"

$vnetName="NSVPX-vnet"

$RGName="multiIP-RG"

$prmStorageAccountName="multiipstorageaccnt"

$avSetName="MultiIP-avset"

$vmSize="Standard\_DS3\_V2"
```

> **Note:**
>
> The minimum requirement for a VPX instance is 2 vCPUs and 2 GB RAM.

```
$publisher="citrix"

$offer="netscalervpx111"

$sku="netscalerbyol"

$version="latest"

$vmNamePrefix="MultiIPVPX"

$nicNamePrefix="MultiipVPX"

$osDiskSuffix="osdiskdb"

$numberOfVMs=1

$ipAddressPrefix="10.0.0."

$ipAddressPrefix1="10.0.1."

$ipAddressPrefix2="10.0.2."

$pubIPName1="MultiIP-pip1"

$pubIPName2="MultiIP-pip2"

$IpConfigName1="IPConfig1"

$IPConfigName2="IPConfig-2"

$IPConfigName3="IPConfig-3"

$IPConfigName4="IPConfig-4"

$frontendSubnetName="default"

$backendSubnetName1="subnet\_1"

$backendSubnetName2="subnet\_2"

$suffixNumber=10
```

### Create a VM

Follow steps 1–10 to create VM1 with multiple NICs and multiple IP addresses, by using PowerShell commands:

1. Create resource group
2. Create storage account
3. Create availability set
4. Create virtual network
5. Create public IP address
6. Create NICs
7. Create VM config object
8. Get credentials and set OS properties for the VM
9. Add NICs
10. Specify OS disk and create VM

After you complete all the steps and commands to create VM1, repeat these steps to create VM2 with parameters specific to it.

#### Create resource group

```
New-AzureRMResourceGroup -Name $RGName -Location $location
```

#### Create storage account

```
$prmStorageAccount=New-AzureRMStorageAccount -Name $prmStorageAccountName -ResourceGroupName $RGName -Type Standard_LRS -Location $location
```

#### Create availability set

```
$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName $RGName -Location $location
```

#### Create virtual network

1. Add subnets.

```
$subnet1=New-AzureRmVirtualNetworkSubnetConfig -Name $frontendSubnetName -AddressPrefix "10.0.0.0/24"
$subnet2=New-AzureRmVirtualNetworkSubnetConfig -Name $backendSubnetName1 -AddressPrefix "10.0.1.0/24"
```

```
$subnet3=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName2 -AddressPrefix "10.0.2.0/24"
```

2. Add virtual network object.

```
$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
     $RGName -Location $location -AddressPrefix 10.0.0.0/16 -Subnet
     $subnet1, $subnet2, $subnet3
```

3. Retrieve subnets.

```
$frontendSubnet=$vnet.Subnets|?{
 $_.Name -eq $frontendSubnetName }

$backendSubnet1=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName1 }

$backendSubnet2=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName2 }
```

#### Create public IP address

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
```

#### Create NICs

Create NIC 0/1

```
$nic1Name=$nicNamePrefix + $suffixNumber + "-Mgmnt"
$ipAddress1=$ipAddressPrefix + $suffixNumber
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    SubnetId $frontendSubnet.Id -PublicIpAddress $pip1 -PrivateIpAddress
     $ipAddress1 -Primary
$nic1=New-AzureRMNetworkInterface -Name $nic1Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig1
```

Create NIC 1/1

```
$nic2Name $nicNamePrefix + $suffixNumber + "-frontend"
$ipAddress2=$ipAddressPrefix1 + ($suffixNumber)
$ipAddress3=$ipAddressPrefix1 + ($suffixNumber + 1)
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    PublicIpAddress $pip2 -SubnetId $backendSubnet1.Id  -
    PrivateIpAddress $ipAddress2  -Primary
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    SubnetId $backendSubnet1.Id  -PrivateIpAddress $ipAddress3
```

```
nic2=New-AzureRMNetworkInterface -Name $nic2Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig2, $IpConfig3
```

Create NIC 1/2

```
$nic3Name=$nicNamePrefix + $suffixNumber + "-backend"
$ipAddress4=$ipAddressPrefix2 + ($suffixNumber)
$IPConfig4=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    SubnetId $backendSubnet2.Id -PrivateIpAddress $ipAddress4 -Primary
$nic3=New-AzureRMNetworkInterface -Name $nic3Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig4
```

#### Create VM config object

```
$vmName=$vmNamePrefix
$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id
```

#### Get credentials and set OS properties

```
$cred=Get-Credential -Message "Type the name and password for VPX login
    ."
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version
```

#### Add NICs

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
    Primary
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.Id
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id
```

#### Specify OS disk and create VM

```
$osDiskName=$vmName + "-" + $osDiskSuffix
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     +$osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $RGName -Location
    $location
```

> **Note:**
>
> Repeat steps 1–10 listed in "Create Multi-NIC VMs by Using PowerShell Commands"to create VM2 with parameters specific to VM2.

### IP configuration details

The following IP addresses are used.

**Table 1**. IP addresses used in VM1

| NIC | Private IP | Public IP (PIP) | Description |
|---|---|---|---|
| 0/1 | 10.0.0.10 | PIP1 | Configured as NSIP (management IP) |
| 1/1 | 10.0.1.10 | PIP2 | Configured as SNIP/GSLB Site IP |
| - | 10.0.1.11 | - | Configured as LB server IP. Public IP is not mandatory |
| 1/2 | 10.0.2.10 | - | Configured as SNIP for sending monitor probes to services; public IP is not mandatory |

**Table 2.** IP addresses used in VM2

| NIC | Internal IP | Public IP (PIP) | Description |
|---|---|---|---|
| 0/1 | 20.0.0.10 | PIP4 | Configured as NSIP (management IP) |
| 1/1 | 20.0.1.10 | PIP5 | Configured as SNIP/GSLB Site IP |
| - | 20.0.1.11 | - | Configured as LB server IP. Public IP is not mandatory |

| NIC | Internal IP | Public IP (PIP) | Description |
|---|---|---|---|
| 1/2 | 20.0.2.10 | - | Configured as SNIP for sending monitor probes to services; public IP is not mandatory |

Here are sample configurations for this scenario, showing the IP addresses and initial LB configurations as created through the NetScaler VPX CLI for VM1 and VM2.

Here's an example configuration on VM1.

```
add ns ip 10.0.1.10 255.255.255.0 -mgmtAccess ENABLED
Add nsip 10.0.2.10 255.255.255.0
add service svc1 10.0.1.10 ADNS 53
add lb vserver v1 HTTP 10.0.1.11 80
add service s1 10.0.2.120 http 80
Add service s2 10.0.2.121 http 80
Bind lb vs v1 s[1-2]
```

Here's an example configuration on VM2.

```
add ns ip 20.0.1.10 255.255.255.0  -mgmtAccess ENABLED
Add nsip 20.0.2.10 255.255.255.0
add service svc1 20.0.1.10 ADNS 53
add lb vserver v1 HTTP 20.0.1.11 80
Add service s1 20.0.2.90 http 80
Add service s2 20.0.2.91 http 80
Bind lb vs v1 s[1-2]
```

### Configure GSLB sites and other settings

Perform the tasks described in the following topic to configure the two GSLB sites and other necessary settings:

Global Server Load Balancing

Here's an example GSLB configuration on VM1 and VM2.

```
enable ns feature LB GSLB
add gslb site site1 10.0.1.10 -publicIP PIP2
add gslb site site2 20.0.1.10 -publicIP PIP5
add gslb service site1_gslb_http_svc1 10.0.1.11 HTTP 80 -publicIP PIP3
   -publicPort 80 -siteName site1
add gslb service site2_gslb_http_svc1 20.0.1.11 HTTP 80 -publicIP PIP6
   -publicPort 80 -siteName site2
add gslb vserver gslb_http_vip1 HTTP
```

```
bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
```

You've configured GSLB on NetScaler VPX instances running on Azure.

### Disaster recovery

Disaster is a sudden disruption of business functions caused by natural calamities or human caused events. Disasters affect data center operations, after which resources and the data lost at the disaster site must be fully rebuilt and restored. The loss of data or downtime in the data center is critical and collapses the business continuity.

One of the challenges that customers face today is deciding where to put their DR site. Businesses are looking for consistency and performance regardless of any underlying infrastructure or network faults.

Possible reasons many organizations are deciding to migrate to the cloud are:

- Having an on-prem data center is very expensive. By using the cloud, the businesses can free up time and resource from expanding their own systems.
- Many of the automated orchrestration enables faster recovery
- Replicate data by providing continuous data protection or continuous snapshots to guard against any outage or attack.
- Support use cases where customers need many different types of compliance and security control which are already present on the public clouds. These make it easier to achieve the compliance they need rather than building their own.

A NetScaler configured for GSLB forwards traffic to the least-loaded or best-performing data center. This configuration, referred to as an active-active setup, not only improves performance, but also provides immediate disaster recovery by routing traffic to other data centers if a data center that is part of the setup goes down. NetScaler thereby saves customers valuable time and money.

#### Multi-NIC Multi-IP (Three-NIC) deployment for disaster recovery

Customers would potentially deploy using three-NIC deployment if they are deploying into a production environment where security, redundancy, availability, capacity, and scalability are critical. With this deployment method, complexity and ease of management are not critical concerns to the users.

#### Single-NIC Multi-IP (One-NIC) deployment for disaster recovery

Customers would potentially deploy using one-NIC deployment if they are deploying into a non-production environment for the following reasons:

- Setting up the environment for testing, or they are staging a new environment before production deployment.
- Deploying directly to the cloud quickly and efficiently.
- While seeking the simplicity of a single subnet configuration.

## Configure GSLB on an active-standby high-availability setup

You can configure global server load balancing (GSLB) on active-standby HA deployment on Azure in three steps:

1. Create a VPX HA pair on each GSLB site. See Configure a high-availability setup with multiple IP addresses and NICs for information about how to create an HA pair.
2. Configure the Azure Load Balancer (ALB) with the front-end IP address and rules to allow GSLB and DNS traffic.

   This step involves the following substeps. See the scenario in this section for the PowerShell commands used to complete these substeps.

   a. Create a front-end `IPconfig` for GSLB site.

   b. Create a back-end address pool with IP address of NIC 1/1 of nodes in HA.

   c. Create load-balancing rules for following:

   ```
   TCP/3009 – gslb communication
   TCP/3008 - gslb communication
   UDP/53 - DNS communication
   ```

   d. Associate back-end address pool with the LB rules created in step c.

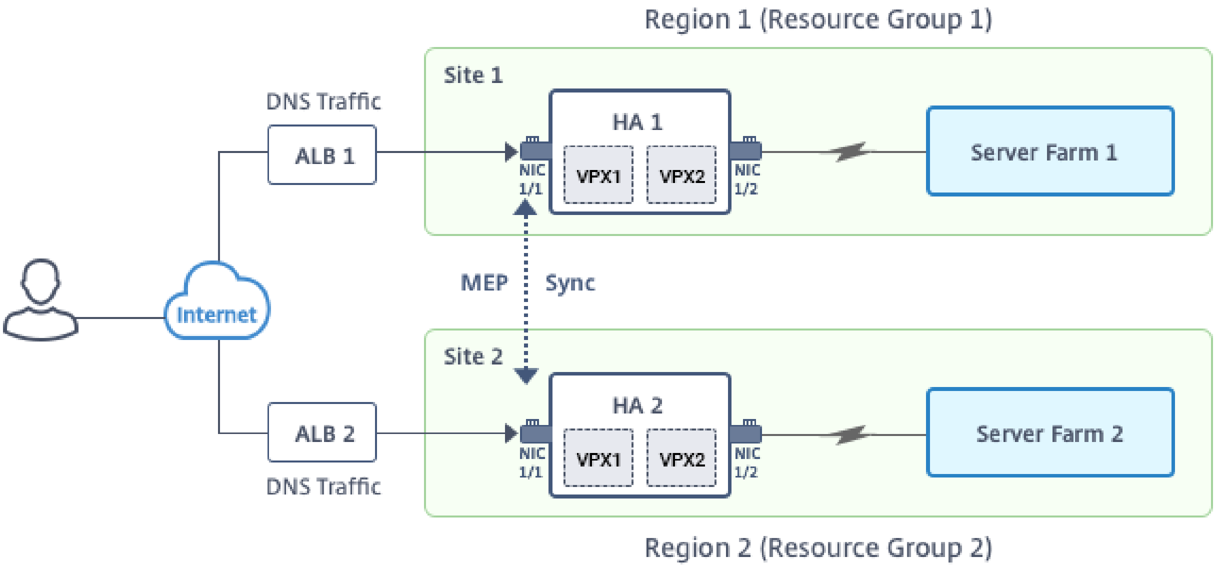
   e. Update the network security group of NIC 1/1 of nodes in both the HA pair to allow the traffic for TCP 3008, TCP 3009 and UDP 53 ports.
3. Enable GSLB on each HA pair.

### Scenario

This scenario includes two sites - Site 1 and Site 2. Each site has an HA pair (HA1 and HA2) configured with multiple NICs, multiple IP addresses, and GSLB.

**Figure:** GLSB on Active-Standy HA Deployment on Azure




In this scenario, each VM has three NICs - NIC 0/1, 1/1, and 1/2. The NICs are configured for the following purposes.

NIC 0/1: to serve management traffic

NIC 1/1: to serve client-side traffic

NIC 1/2: to communicate with back-end servers

### Parameter Settings

Following are sample parameters settings for the ALB. You can use different settings if you want.

```
$locName="South east Asia"

$rgName="MulitIP-MultiNIC-RG"

$pubIPName4="PIPFORGSLB1"

$domName4="vpxgslbdns"

$lbName="MultiIPALB"

$frontEndConfigName2="FrontEndIP2"

$backendPoolName1="BackendPoolHttp"

$lbRuleName2="LBRuleGSLB1"

$lbRuleName3="LBRuleGSLB2"

$lbRuleName4="LBRuleDNS"
```

```
$healthProbeName="HealthProbe"
```

#### Configure ALB with the front-end IP address and rules to allow GSLB and DNS traffic

Step 1. **Create a public IP for GSLB site IP**

```
$pip4=New-AzureRmPublicIpAddress -Name $pubIPName4 -ResourceGroupName
    $rgName -DomainNameLabel $domName4 -Location $locName -
    AllocationMethod Dynamic


Get-AzureRmLoadBalancer -Name \$lbName -ResourceGroupName \$rgName |
    Add-AzureRmLoadBalancerFrontendIpConfig -Name \$frontEndConfigName2
    -PublicIpAddress \$pip4 | Set-AzureRmLoadBalancer
```

Step 2. **Create LB rules and update the existing ALB**.

```
$alb = get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName


$frontendipconfig2=Get-AzureRmLoadBalancerFrontendIpConfig -
    LoadBalancer $alb -Name $frontEndConfigName2


$backendPool=Get-AzureRmLoadBalancerBackendAddressPoolConfig -
    LoadBalancer $alb -Name $backendPoolName1


$healthprobe=Get-AzureRmLoadBalancerProbeConfig -LoadBalancer $alb -
    Name $healthProbeName


\$alb | Add-AzureRmLoadBalancerRuleConfig -Name \$lbRuleName2 -
    BackendAddressPool \$backendPool -FrontendIPConfiguration  \
    $frontendipconfig2 -Protocol \"Tcp\" -FrontendPort 3009 -BackendPort
     3009 -Probe \$healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer


\$alb | Add-AzureRmLoadBalancerRuleConfig -Name \$lbRuleName3 -
    BackendAddressPool \$backendPool -FrontendIPConfiguration  \
    $frontendipconfig2 -Protocol \"Tcp\" -FrontendPort 3008 -BackendPort
     3008 -Probe \$healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer


\$alb | Add-AzureRmLoadBalancerRuleConfig -Name \$lbRuleName4 -
    BackendAddressPool \$backendPool -FrontendIPConfiguration  \
    $frontendipconfig2 -Protocol \"Udp\" -FrontendPort 53 -BackendPort
    53 -Probe \$healthprobe -EnableFloatingIP  | Set-AzureRmLoadBalancer
```

### Enable GSLB on each high availability pair

Now you've two front-end IP addresses for each ALB: ALB 1 and ALB 2. One IP address is for the LB virtual server and the other for the GSLB site IP.

HA 1 has the following front-end IP addresses:

- FrontEndIPofALB1 (for LB virtual server)
- PIPFORGSLB1 (GSLB IP)

HA 2 has the following front-end IP addresses:

- FrontEndIPofALB2 (for LB virtual server)
- PIPFORGSLB2 (GSLB IP)

The following commands are used for this scenario.

```
enable ns feature LB GSLB

add service dnssvc PIPFORGSLB1 ADNS 53

add gslb site site1 PIPFORGSLB1 -publicIP PIPFORGSLB1

add gslb site site2 PIPFORGSLB2 -publicIP PIPFORGSLB2

add gslb service site1_gslb_http_svc1 FrontEndIPofALB1 HTTP 80 -
    publicIP FrontEndIPofALB1 -publicPort 80 -siteName site1

add gslb service site2_gslb_http_svc1 FrontEndIPofALB2 HTTP 80 -
    publicIP FrontEndIPofALB2 -publicPort 80 -siteName site2

add gslb vserver gslb_http_vip1 HTTP

bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
```

**Related resources**:

Configure GSLB on NetScaler VPX instances

Global Server Load Balancing

## Deploy NetScaler GSLB on Azure

With the increasing demand, businesses running an on-prem data center serving regional customers want to scale and deploy across globally using Azure cloud. With NetScaler on the network administrator's side, you can use the GSLB StyleBook to configure applications both on-prem and in the cloud. You can transfer the same configuration to the cloud with NetScaler ADM. You can reach either on-prem or cloud resources depending on proximity with GSLB. This allows you to have a seamless experience no matter where you are in the world.

### DBS overview

NetScaler GSLB supports using Domain-Based Services (DBS) for Cloud load balancers. This allows for the auto-discovery of dynamic cloud services using a cloud load balancer solution. This configuration allows the NetScaler to implement GSLB DBS in an Active-Active environment. DBS allows the scaling of back end resources in Microsoft Azure environments from DNS discovery. This section covers integration between NetScalers in the Azure Autoscale environment.

#### Domain name-based services using Azure load balancer (ALB)

GSLB DBS uses the FQDN of the user ALB to dynamically update the GSLB service groups to include the back-end servers that are being created and deleted within Azure. To configure this feature, the user points the Citrix ADC to their ALB to dynamically route to different servers in Azure. They can do this without having to manually update the Citrix ADC every time an instance is created and deleted within Azure. The Citrix ADC DBS feature for GSLB service groups uses DNS-aware service discovery to determine the member service resources of the DBS namespace identified in the Autoscale group.

The following image depicts the NetScaler GSLB DBS Autoscale components with cloud load balancers:


### Azure GSLB prerequisites

The prerequisites for the NetScaler GSLB service groups include a functioning Microsoft Azure environment, along with the knowledge and ability to configure Linux Web Servers, NetScaler appliances within Azure, public IP addresses, and Azure load balancers (ALB).

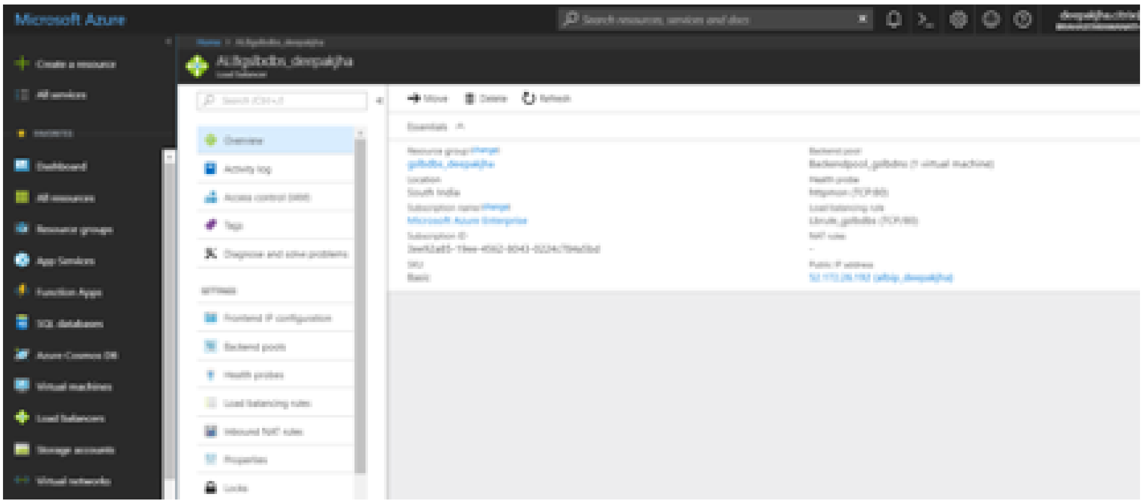
- GSLB DBS Service integration requires NetScaler version 12.0.57 for Microsoft Azure load balancer instances.
- GSLB service group entity: NetScaler version 12.0.57.
- GSLB service group is introduced which supports autoscale using DBS dynamic discovery.
- DBS Feature Components (domain-based service) must be bound to the GSLB service group.

Example:

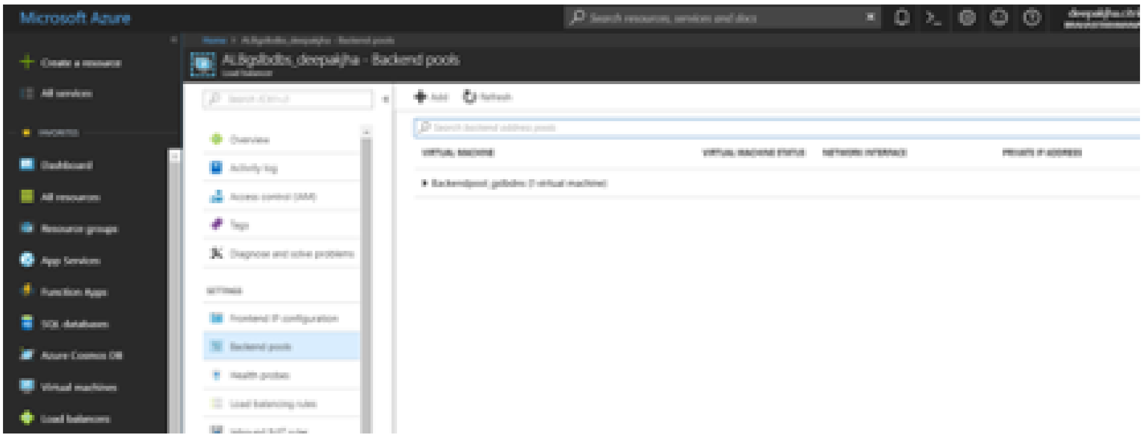
````
```
 > add server sydney_server LB-Sydney-xxxxxxxxxx.australiaeast.cloudapp.azure.com
 > add gslb serviceGroup sydney_sg HTTP -autoscale DNS -siteName sydney
 > bind gslb serviceGroup sydney_sg sydney_server 80
```
````

Configure Azure components

1. Log in to the user Azure Portal and create a new virtual machine from a NetScaler template.
2. Create an Azure load balancer.



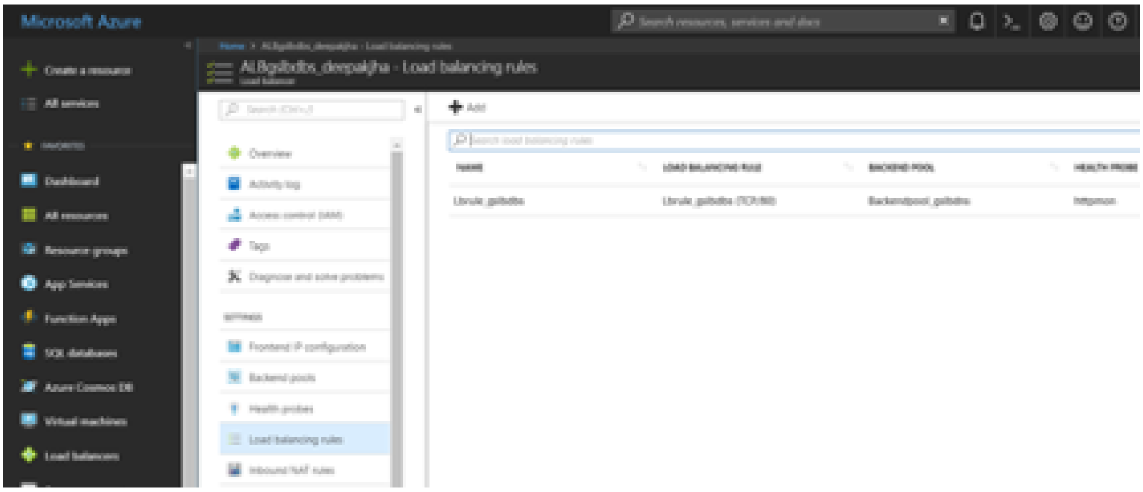
3. Add the created NetScaler back-end pools.



4. Create a health probe for port 80.

   Create a load balancing rule using the front-end IP created from the load balancer.

   - Protocol: TCP
   - Back-end Port: 80
   - Back-end pool: NetScaler created in step 1
   - Health Probe: Created in step 4
   - Session Persistence: None



Configure NetScaler GSLB domain-based service

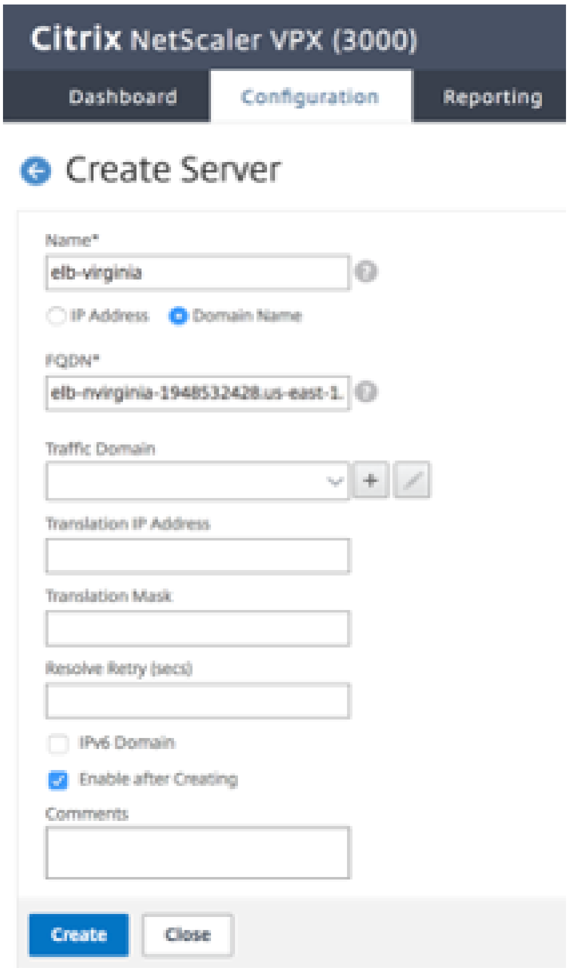
The following configurations summarize what is required to enable domain-based services for autoscaling ADCs in a GSLB enabled environment.

- Traffic management configurations
- GSLB configurations

Traffic management configurations

> **Note:**
>
> It is required to configure the NetScaler with either a nameserver or a DNS virtual server through which the ALB domains are resolved for the DBS service groups. For more information on name servers or DNS virtual servers, see: DNS nameServer.

1. Navigate to **Traffic Management > Load Balancing > Servers**.
2. Click **Add** to create a server, provide a name and FQDN corresponding to the A record (domain name) in Azure for the ALB.



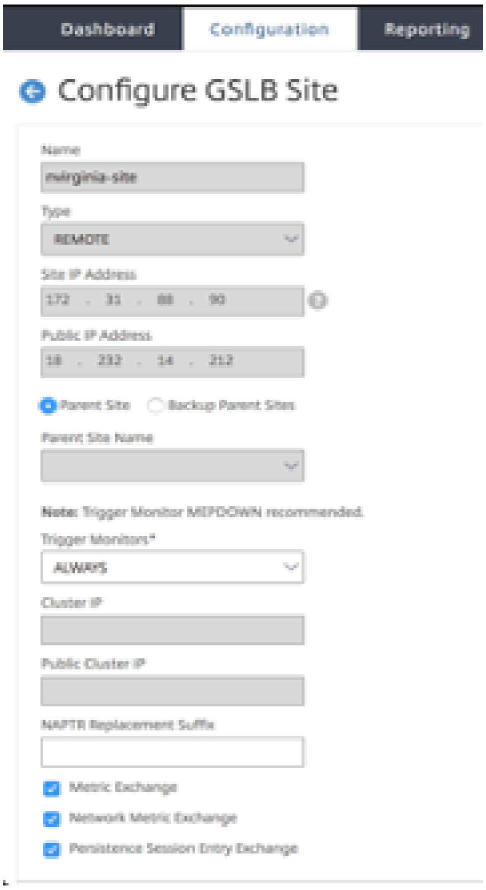
3. Repeat step 2 to add the second ALB from the second resource in Azure.

GSLB configurations

1. Click the **Add** button to configure a GSLB site.
2. Specify the details for configuring the GSLB site

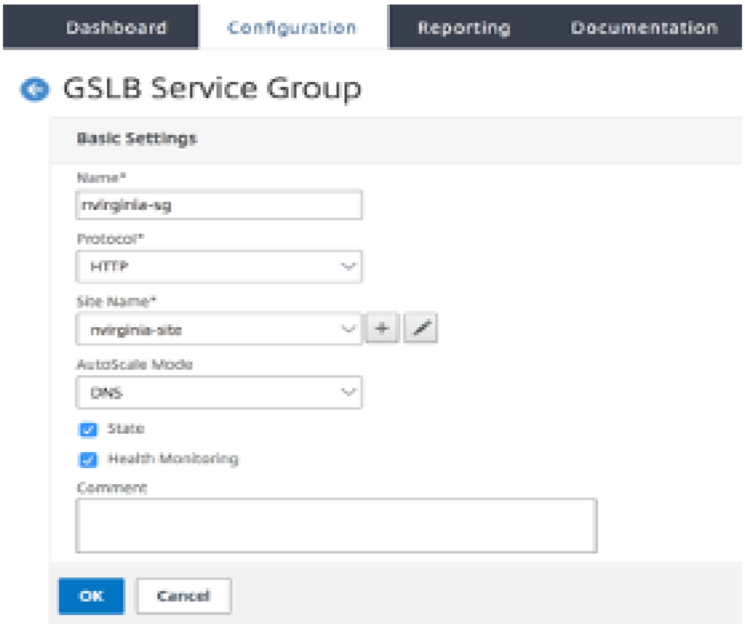
Name the site. Type is configured as remote or local based on which NetScaler you are configuring the site on. The site IP address is the IP address for the GSLB site. The GSLB site uses this IP address to communicate with the other GSLB sites. The public IP address is required when using a cloud service where a particular IP address is hosted on an external firewall or NAT device. Configure the site as a parent site and ensure that the **Trigger Monitors** are set to **ALWAYS**. Also, be sure to check the three boxes at the bottom for **Metric Exchange**, **Network Metric Exchange**, and **Persistence Session Entry Exchange**.

We recommend you set the **Trigger monitor** to **MEPDOWN**. For more information, see Configure a GSLB Service Group.

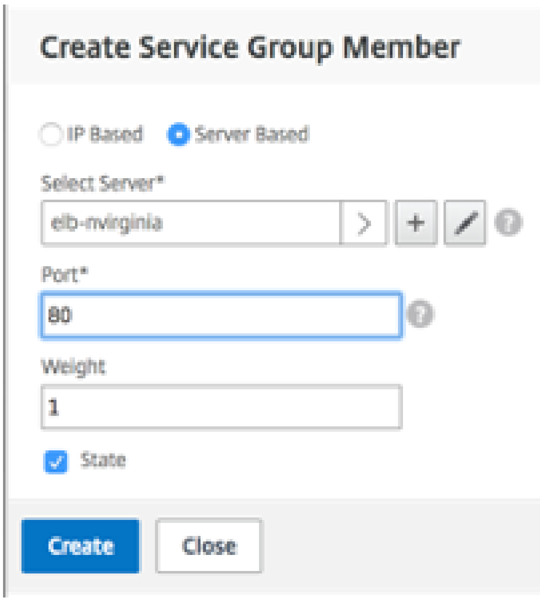


3. Click **Create**.
4. Navigate to **Traffic Management > GSLB > Service Groups**.
5. Click **Add** to add a service group.
6. Specify the details to configure the service group

   Name the Service Group, use the HTTP protocol. Under **Site Name** choose the respective site that you created. Be sure to configure Autoscale Mode as DNS and check off the boxes for State and Health Monitoring. Click **OK** to create the Service Group.



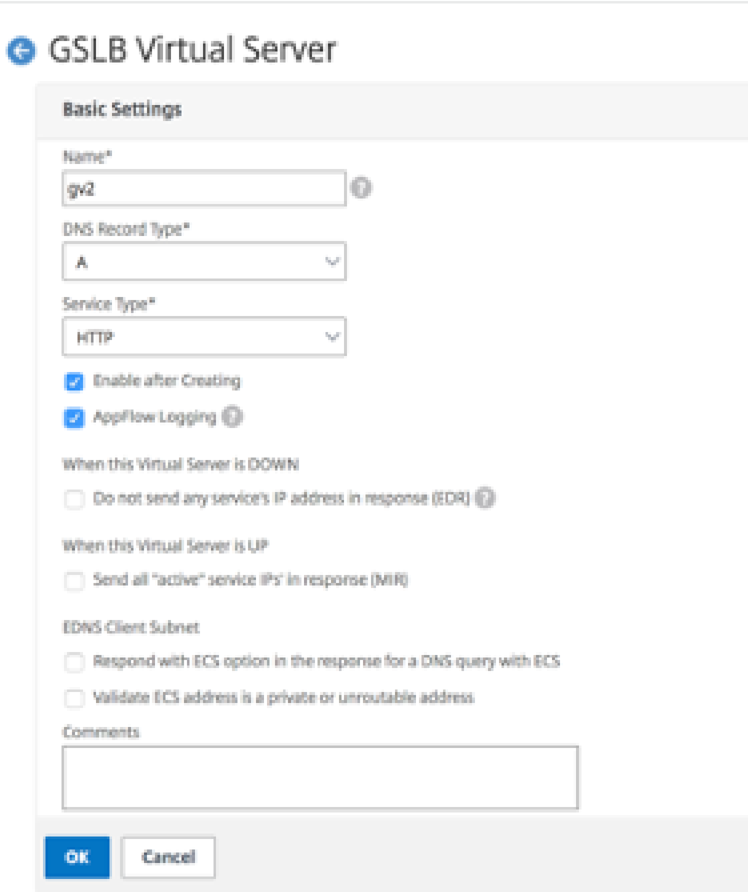
7. Click **Service Group Members** and select **Server Based**. Select the respective ALB that was configured in the start of the run guide. Configure the traffic to go over port 80. Click **Create**.



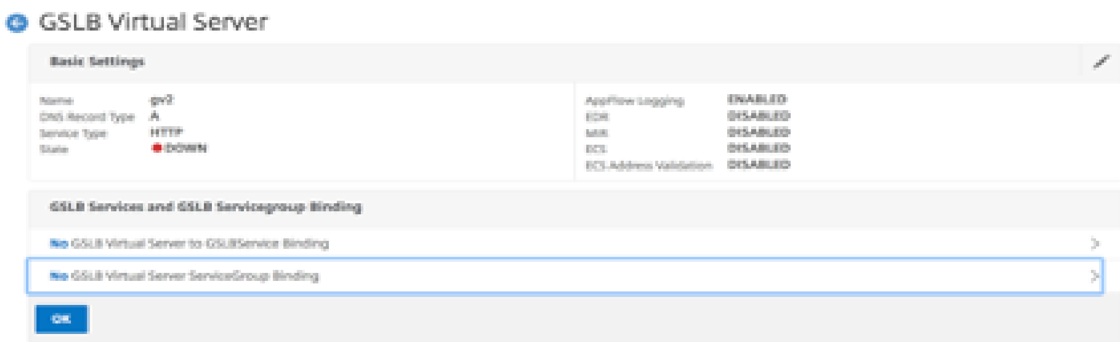
The service group member binding populates with 2 instances that it receives from the ALB.

8. Repeat steps 5 and 6 to configure the service group for the second resource location in Azure. (This can be done from the same NetScaler GUI).
9. To set up a GSLB virtual server. Navigate to **Traffic Management > GSLB > Virtual Servers**.
10. Click **Add** to create the virtual server.
11. Specify the details to configure the GSLB virtual server.

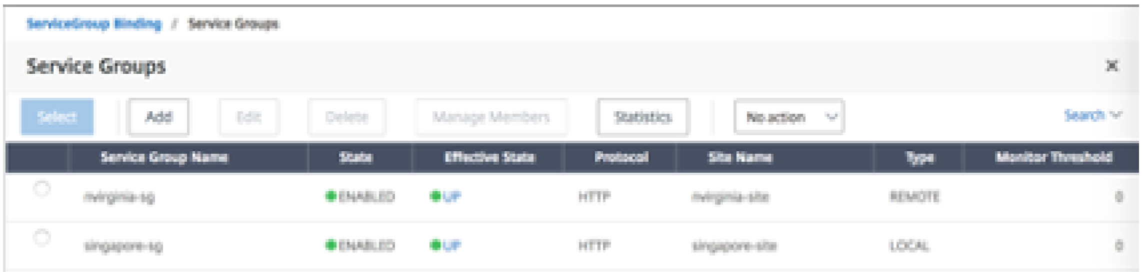
    Name the server, DNS Record Type is set as A, Service Type is set as HTTP, and check the boxes for Enable after Creating and AppFlow Logging. Click **OK** to create the GSLB Virtual Server.



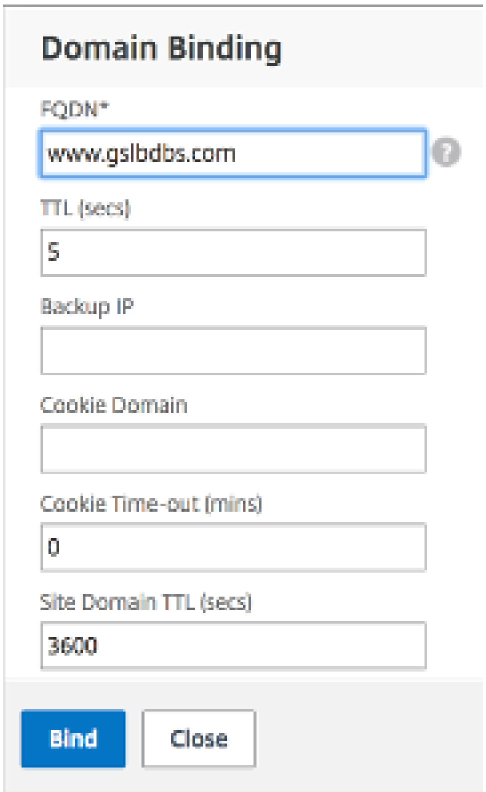
12. Once the GSLB virtual server is created, click **No GSLB Virtual Server ServiceGroup Binding**.



13. Under **ServiceGroup Binding** use **Select Service Group Name** to select and add the service groups that were created in the previous steps.



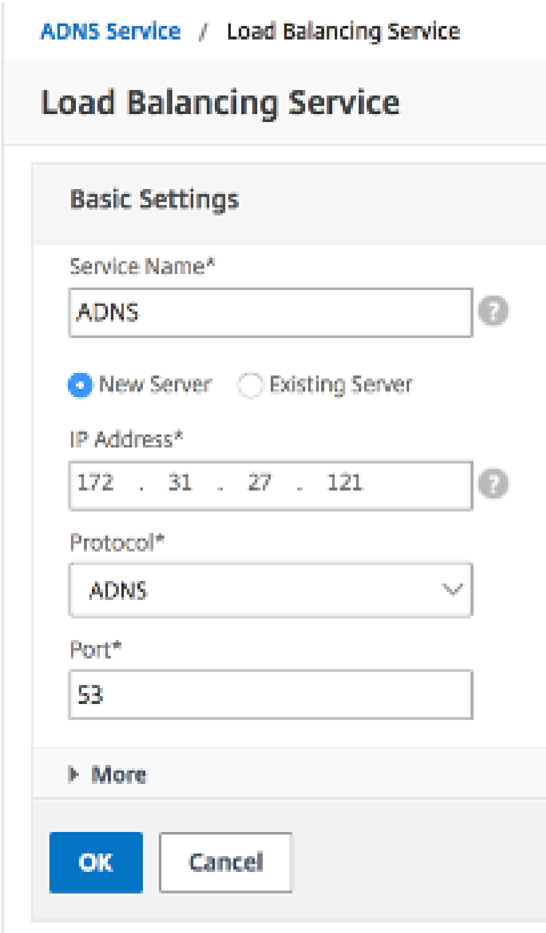
14. Configure the GSLB virtual server domain binding by clicking **No GSLB Virtual Server Domain Binding**. Configure the FQDN and bind. Retain the default setting for other parameters.



15. Configure the ADNS Service by clicking **No Service**.

16. Specify the details to configure load balancing service.

    Add a **Service Name**, click **New Server**, and enter the **IP Address** of the ADNS server. If the user ADNS is already configured, users can select **Existing Server** and then choose the user ADNS from the drop-down menu. Make sure that the protocol is ADNS and the traffic is configured to flow over port 53.



17. Configure the **Method** as **Least Connection** and the Backup Method as **Round Robin**.
18. Click **Done** and verify that the user GSLB virtual server is shown as Up.

Other resources

NetScaler Global Load Balancing for Hybrid and Multi-Cloud Deployments

Deploy NetScaler Web App Firewall on Azure

NetScaler Web App Firewall is an enterprise grade solution offering state of the art protections for modern applications. NetScaler Web App Firewall mitigates threats against public-facing assets, including websites, web applications, and APIs. NetScaler Web App Firewall includes IP reputation-based filtering, Bot mitigation, OWASP Top 10 application threats protections, Layer 7 DDoS protection and more. Also included are options to enforce authentication, strong SSL/TLS ciphers, TLS 1.3, rate limiting and rewrite policies. Using both basic and advanced WAF protections, NetScaler Web App Firewall provides comprehensive protection for your applications with unparalleled ease of use. Getting up and running is a matter of minutes. Further, using an automated learning model, called dynamic profiling, NetScaler Web App Firewall saves users precious time. By automatically learning how a protected application works, NetScaler Web App Firewall adapts to the application even as developers deploy and alter the applications. NetScaler Web App Firewall helps with compliance for all major regulatory standards and bodies, including PCI-DSS, HIPAA, and more. With our CloudFormation templates, it has never been easier to get up and running quickly. With auto scaling, users can rest assured that their applications remain protected even as their traffic scales up.

NetScaler Web App Firewall can be installed as either a Layer 3 network device or a Layer 2 network bridge between customer servers and customer users, usually behind the customer company's router or firewall. For more information, see Introduction to NetScaler Web App Firewall.

NetScaler Web App Firewall deployment strategy

1. Deploy the web application firewall is to evaluate which applications or specific data need maximum security protection, which ones are less vulnerable, and the ones for which security inspection can safely be bypassed. This helps users in coming up with an optimal configuration, and in designing appropriate policies and bind points to segregate the traffic. For example, users might want to configure a policy to bypass security inspection of requests for static web content, such as images, MP3 files, and movies, and configure another policy to apply advanced security checks to requests for dynamic content. Users can use multiple policies and profiles to protect different contents of the same application.
2. To baseline the deployment, create a virtual server and run test traffic through it to get an idea of the rate and amount of traffic flowing through the user system.
3. Deploy the Web Application Firewall. Use NetScaler ADM and the Web Application Firewall StyleBook to configure the Web Application Firewall. See the StyleBook section below in this guide for details.
4. Implement the NetScaler Web App Firewall and OWASP Top Ten.

The three of the Web Application Firewall protections are especially effective against common types of Web attacks, and are therefore more commonly used than any of the others. Thus, they should be implemented in the initial deployment. They are:

- **HTML Cross-Site Scripting**: Examines requests and responses for scripts that attempt to access or modify content on a different website than the one on which the script is located. When this check finds such a script, it either renders the script harmless before forwarding the request or response to its destination, or it blocks the connection.
- **HTML SQL Injection**: Examines requests that contain form field data for attempts to inject SQL commands into a SQL database. When this check detects injected SQL code, it either blocks the request or renders the injected SQL code harmless before forwarding the request to the Web server.

> **Note:**
>
> Ensure that your Web App Firewall is correctly configured for the following conditions to apply in your configuration:
>
> ```
> >-  If users enable the HTML Cross-Site Scripting check or
>     the HTML SQL Injection check (or both).
> >
> >-  User protected websites accept file uploads or contain
>     Web forms that can contain large POST body data.
> ```

For more information about configuring the Web Application Firewall to handle this case, see Configuring the Application Firewall: Configuring the Web App Firewall.

- **Buffer Overflow**: Examines requests to detect attempts to cause a buffer overflow on the Web server.

Configuring the Web Application Firewall

Ensure that the NetScaler Web App Firewall is already enabled and functioning correctly. We recommend that you configure NetScaler Web App Firewall using the Web Application Firewall StyleBook. Most users find it the easiest method to configure the Web Application Firewall, and it is designed to prevent mistakes. Both the GUI and the command-line interface are intended for experienced users, primarily to modify an existing configuration or use advanced options.

SQL injection

The NetScaler Web App Firewall HTML SQL Injection check provides special defenses against the injection of unauthorized SQL code that might break user application security. NetScaler Web App Firewall examines the request payload for injected SQL code in three locations: 1) POST body, 2) headers, and 3) cookies. For more information, see HTML SQL injection check.

Cross-Site scripting

The HTML Cross-Site Scripting (cross-site scripting) check examines both the headers and the POST bodies of user requests for possible cross-site scripting attacks. If it finds a cross-site script, it either modifies (transforms) the request to render the attack harmless, or blocks the request. For more information, see HTML cross-site scripting check.

Buffer overflow check

The Buffer Overflow check detects attempts to cause a buffer overflow on the web server. If the Web Application Firewall detects that the URL, cookies, or header are longer than the configured length, it blocks the request because it can cause a buffer overflow. For more information, see Buffer overflow check.

Virtual patching/signatures

The signatures provide specific, configurable rules to simplify the task of protecting user websites against known attacks. A signature represents a pattern that is a component of a known attack on an operating system, web server, website, XML-based web service, or other resource. A rich set of preconfigured built-in or native rules offers an easy-to-use security solution, applying the power of pattern matching to detect attacks and protect against application vulnerabilities. For more information, see Signatures.

NetScaler Web App Firewall supports both **Auto & Manual** update of signatures.
We also suggest enabling **Auto-update** for signatures to stay up to date.

These signature files are hosted on the AWS Environment and it is important to allow outbound access to NetScaler IP address's from network firewalls to fetch the latest signature files. There is no effect of updating signatures to the NetScaler while processing real-time traffic.

Application security analytics

The **Application Security Dashboard** provides a holistic view of the security status of user applications. For example, it shows key security metrics such as security violations, signature violations, and threat indexes. The application security dashboard also displays attack-related information such as syn attacks, small window attacks, and DNS flood attacks for the discovered NetScaler.

> **Note:**
>
> To view the metrics of the application security dashboard, AppFlow for Security insight should be enabled on the NetScaler instances that users want to monitor.

To view the security metrics of a NetScaler instance on the application security dashboard:

1. Log in to NetScaler ADM using the administrator credentials.
2. Navigate to **Applications > App Security Dashboard**, and select the instance IP address from the Devices list.

Users can further drill down on the discrepancies reported on the Application Security Investigator by clicking the bubbles plotted on the graph.

Centralized learning on ADM

NetScaler Web App Firewall protects user web applications from malicious attacks such as SQL injection and cross-site scripting (XSS). To prevent data breaches and provide the right security protection, users must monitor their traffic for threats and real-time actionable data on attacks. Sometimes, the attacks reported might be false-positives and those need to be provided as an exception.

The centralized learning on NetScaler ADM is a repetitive pattern filter that enables WAF to learn the behavior (the normal activities) of user web applications. Based on monitoring, the engine generates a list of suggested rules or exceptions for each security check applied on the HTTP traffic.

It is much easier to deploy relaxation rules using the learning engine than to manually deploy it as necessary relaxations.

To deploy the learning feature, users must first configure a Web Application Firewall profile (set of security settings) on the user NetScaler. For more information, see Creating Web App Firewall Profiles.

NetScaler ADM generates a list of exceptions (relaxations) for each security check. As an administrator, you can review the list of exceptions in NetScaler ADM and decide to deploy or skip.

Using the WAF learning feature in NetScaler ADM, you can:

- Configure a learning profile with the following security checks.
  - Buffer Overflow
  - HTML Cross-Site Scripting

    > **Note:**
    >
    > The cross-site script limitation of location is only FormField.

- HTML SQL Injection

> **Note:**
>
> For the HTML SQL Injection check, users must configure `set -sqlinjectionTransformSpec` `ON` and `set -sqlinjectiontype sqlspclcharorkeywords` in the NetScaler.

- Check the relaxation rules in NetScaler ADM and decide to take the necessary action (deploy or skip).
- Get the notifications through email, slack, and ServiceNow.
- Use the dashboard to view relaxation details.

To use the WAF learning in NetScaler ADM:

1. Configure the learning profile: Configure the Learning Profile
2. See the relaxation rules: View Relaxation Rules and Idle Rules
3. Use the WAF learning dashboard: View WAF Learning Dashboard

StyleBooks

StyleBooks simplify the task of managing complex NetScaler configurations for user applications. A StyleBook is a template that users can use to create and manage NetScaler configurations. Here, users are primarily concerned with the StyleBook used to deploy the Web Application Firewall. For more information on StyleBooks, see StyleBooks.

Security insight analytics

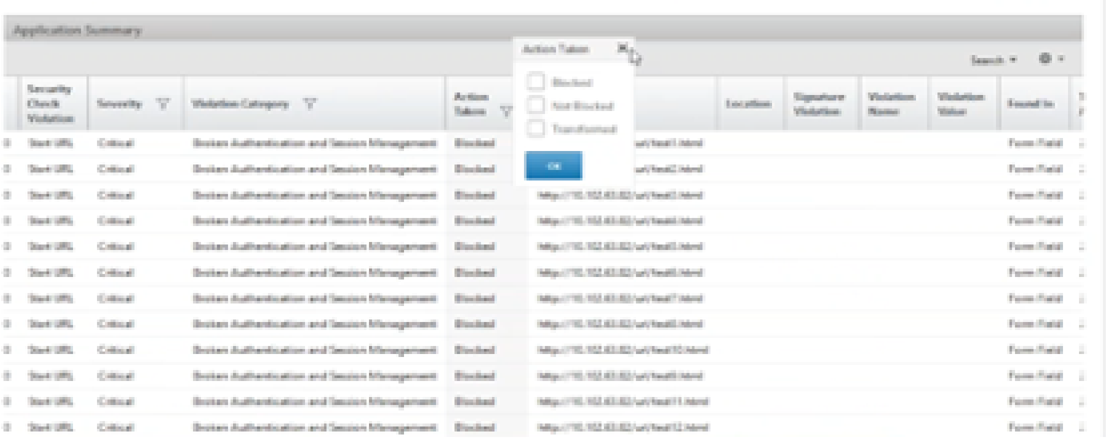
Web and web service applications that are exposed to the Internet have become increasingly vulnerable to attacks. To protect applications from attack, users need visibility into the nature and extent of past, present, and impending threats, real-time actionable data on attacks, and recommendations on countermeasures. Security Insight provides a single-pane solution to help users assess user application security status and take corrective actions to secure user applications.
For more information, see Security Insight.

Obtain detailed information about security breaches

Users might want to view a list of the attacks on an application and gain insights into the type and severity of attacks, actions taken by the ADC instance, resources requested, and the source of the attacks.

For example, users might want to determine how many attacks on Microsoft Lync were blocked, what resources were requested, and the IP addresses of the sources.

On the **Security Insight dashboard**, click **Lync** > **Total Violations**. In the table, click the filter icon in the **Action Taken** column header, and then select **Blocked**.



For information about the resources that were requested, review the **URL** column. For information about the sources of the attacks, review the **Client IP** column.

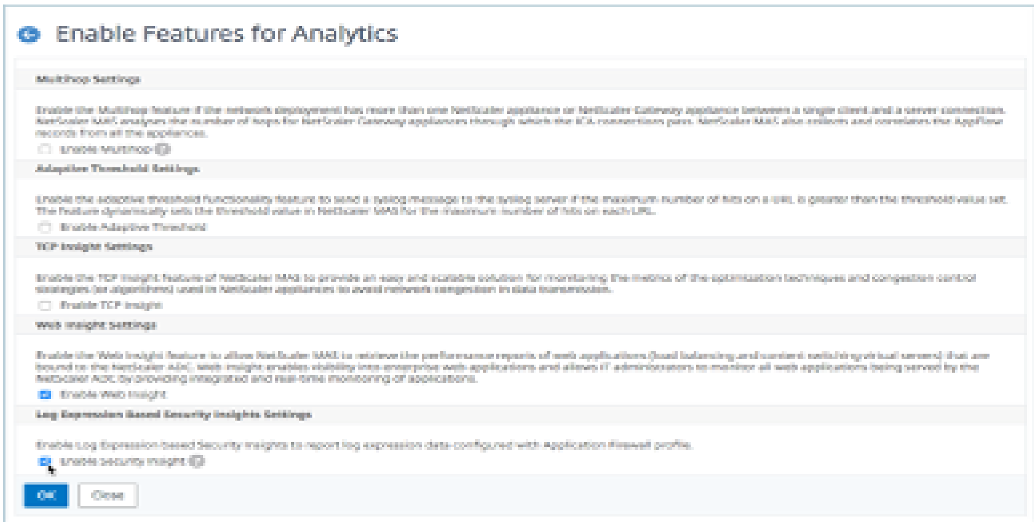
View log expression details

NetScaler use log expressions configured with the Application Firewall profile to take action for the attacks on an application in the user enterprise. In **Security Insight**, users can view the values returned for the log expressions used by the ADC instance. These values include, request header, request body and so on. In addition to the log expression values, users can also view the log expression name and the comment for the log expression defined in the Application Firewall profile that the ADC instance used to take action for the attack.

Prerequisites:

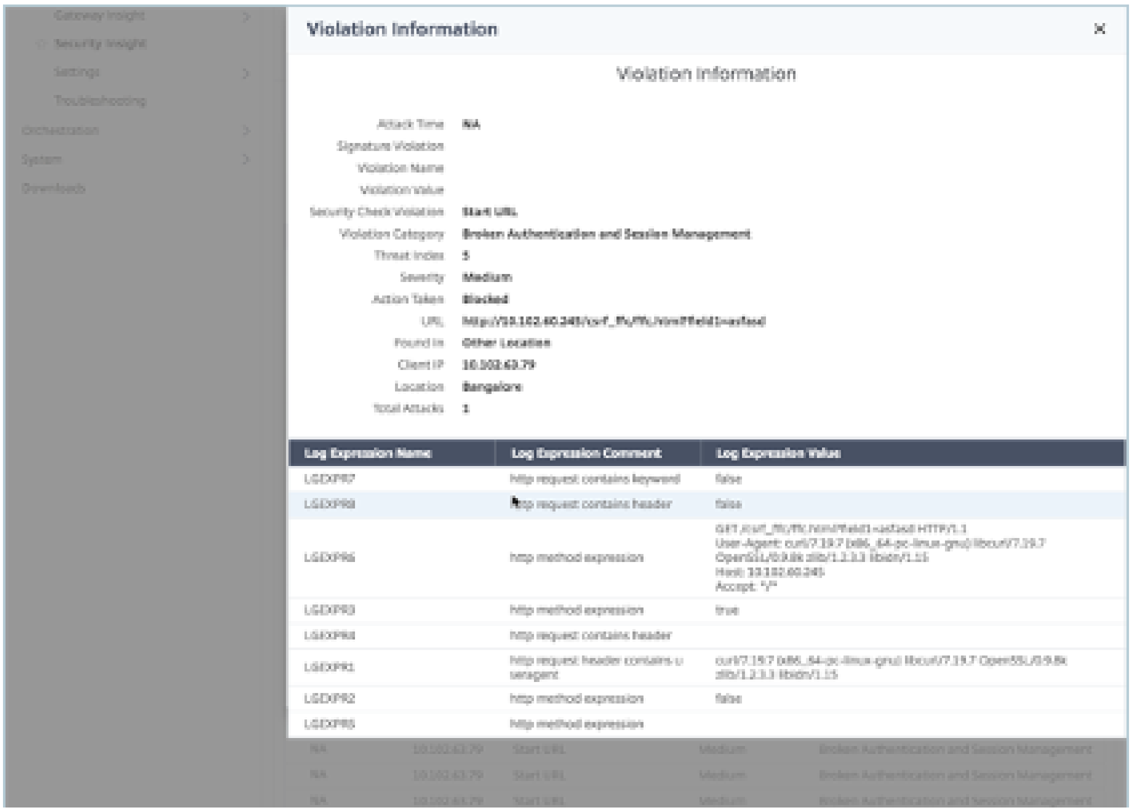
Ensure that users:

- Configure log expressions in the Application Firewall profile. For more information, see Application Firewall.
- Enable log expression-based Security Insights settings in NetScaler ADM. Do the following:
  - Navigate to **Analytics** > **Settings**, and click **Enable Features for Analytics**.
  - In the Enable Features for Analytics page, select **Enable Security Insight** under the **Log Expression Based Security Insight Setting** section and click **OK**.



For example, you might want to view the values of the log expression returned by the ADC instance for the action it took for an attack on Microsoft Lync in the user enterprise.

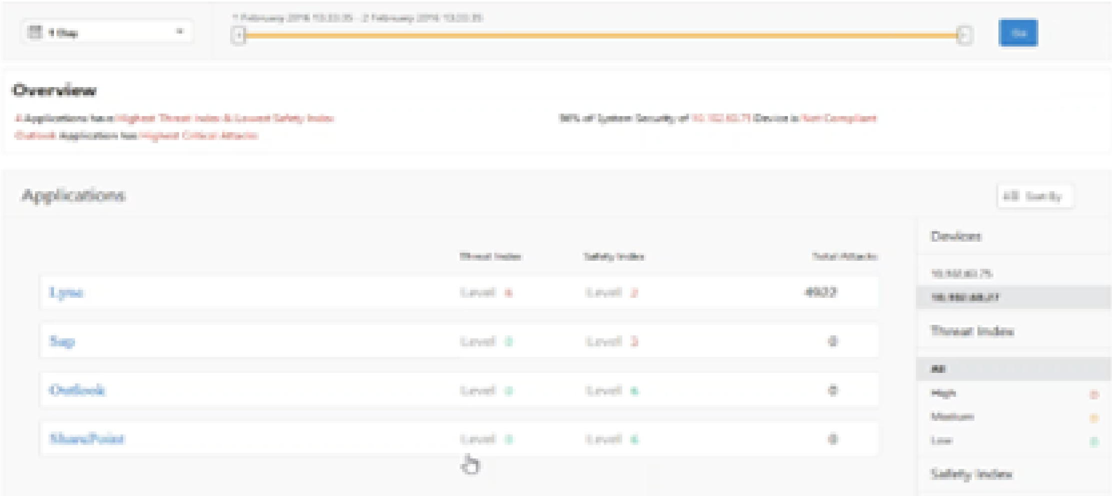
On the **Security Insight dashboard**, navigate to **Lync** > **Total Violations**. In the Application Summary table, click the URL to view the complete details of the violation in the **Violation Information** page including the log expression name, comment, and the values returned by the ADC instance for the action.



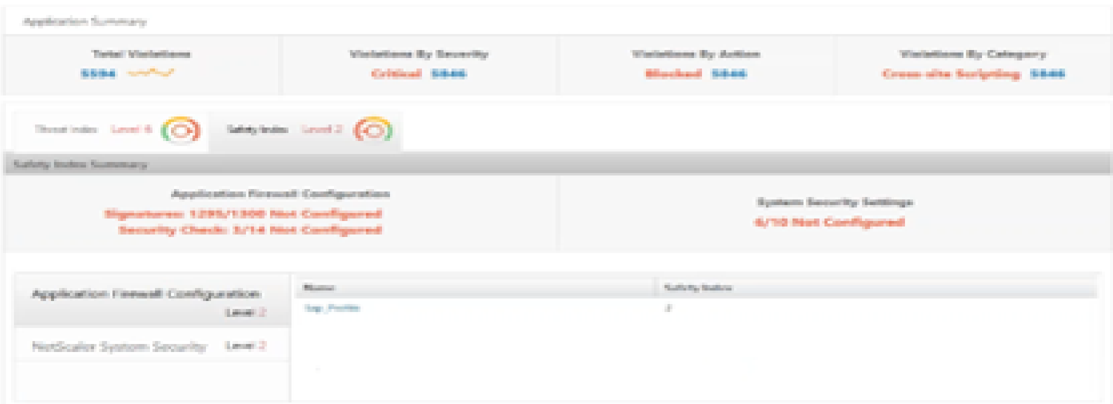
Determine the Safety Index before deploying the Configuration. Security breaches occur after users deploy the security configuration on an ADC instance, but users might want to assess the effectiveness of the security configuration before they deploy it.

For example, users might want to assess the safety index of the configuration for the SAP application on the ADC instance with IP address 10.102.60.27.

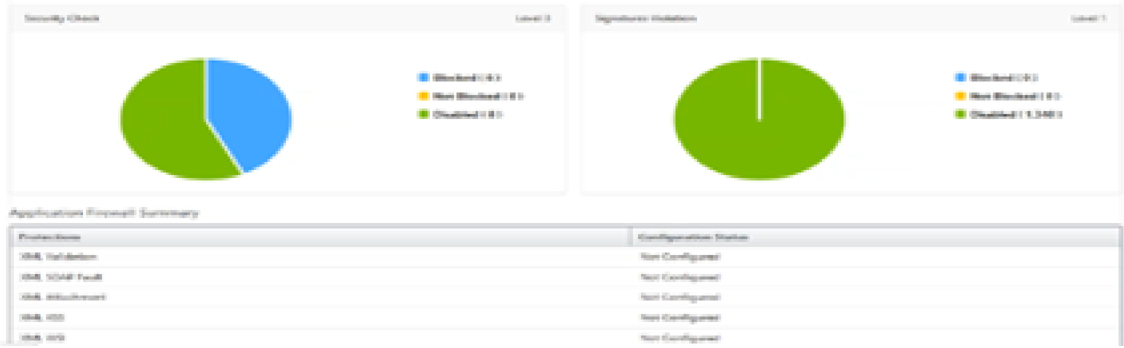
On the **Security Insight dashboard**, under **Devices**, click the IP address of the ADC instance that users configured. Users can see that both the threat index and the total number of attacks are 0. The threat index is a direct reflection of the number and type of attacks on the application. Zero attacks indicate that the application is not under any threat.



Click **Sap** > **Safety Index** > **SAP_Profile** and assess the safety index information that appears.



In the application firewall summary, users can view the configuration status of different protection settings. If a setting is set to log or if a setting is not configured, the application is assigned a lower safety index.



Security violations

Web applications that are exposed to the Internet have become vulnerable to attacks drastically. NetScaler ADM enables you to visualize actionable violation details to protect applications from attacks.

View application security violation details

Web applications that are exposed to the Internet have become drastically more vulnerable to attacks. NetScaler ADM enables users to visualize actionable violation details to protect applications from attacks. Navigate to **Security** > **Security Violations** for a single-pane solution to:

- Access the application security violations based on their categories such as **Network**, **Bot**, and **WAF**
- Take corrective actions to secure the applications

To view the security violations in NetScaler ADM, ensure:

- Users have a premium license for the NetScaler (for WAF and BOT violations).
- Users have applied for a license on the load balancing or content switching virtual servers (for WAF and BOT). For more information, see Manage Licensing on Virtual Servers.
- Users can enable more settings. For more information, see the procedure available at the Setting up section in the NetScaler product documentation: Setting up.

Violation categories

NetScaler ADM enables users to view the violations available in All Violations:

Setting up

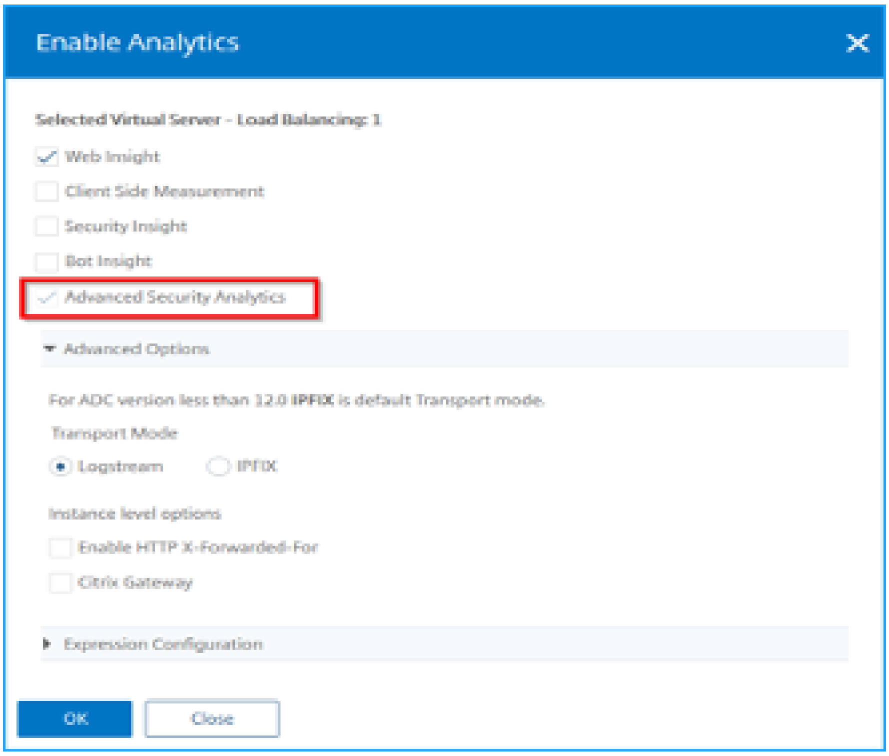
For violations, ensure whether **Metrics Collector** is enabled. By default, **Metrics Collector** is enabled on the NetScaler. For more information, see Configure Intelligent App Analytics.

Enable advanced security analytics

- Navigate to **Networks** > **Instances** > **NetScaler**, and select the instance type. For example, MPX.
- Select the NetScaler instance and from the **Select Action** list, select **Configure Analytics**.
- Select the virtual server and click **Enable Analytics**.
- On the **Enable Analytics** window:
  - Select **Web Insight**. After users select Web Insight, the read-only **Advanced Security Analytics** option is enabled automatically.

    > **Note:**
    >
    > The **Advanced Security Analytics** option is displayed only for premium licensed ADC instances.

  - Select **Logstream** as Transport Mode
  - The Expression is true by default
  - Click **OK**

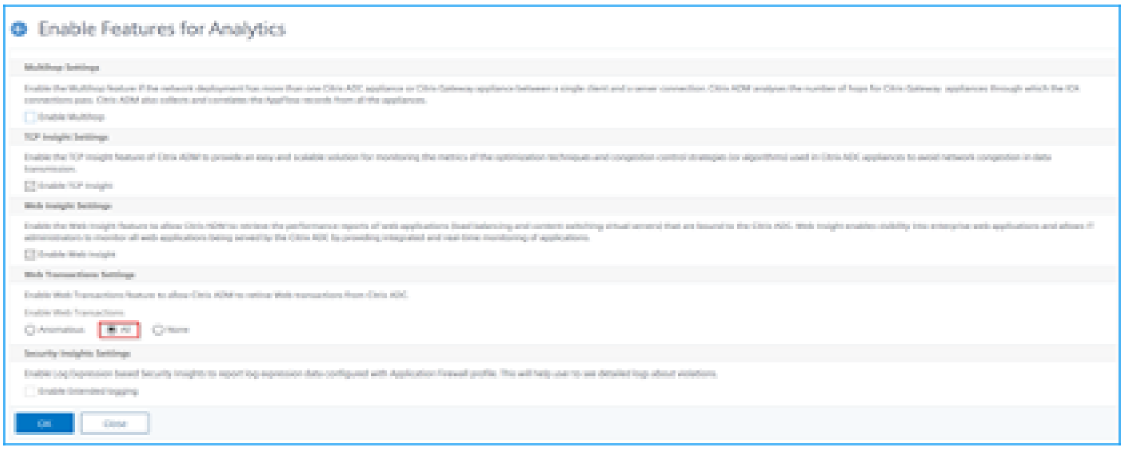


Enable web transaction settings

- Navigate to **Analytics** > **Settings**.

The **Settings** page is displayed.

- Click **Enable Features for Analytics**.
- Under **Web Transaction Settings**, select **All**.



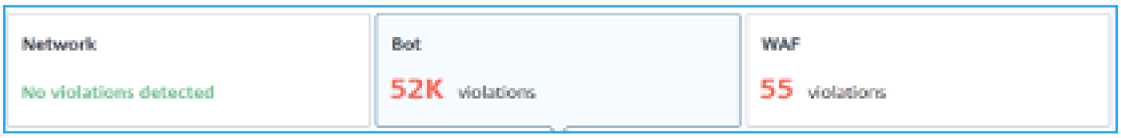
- Click **Ok**.

Security violations dashboard

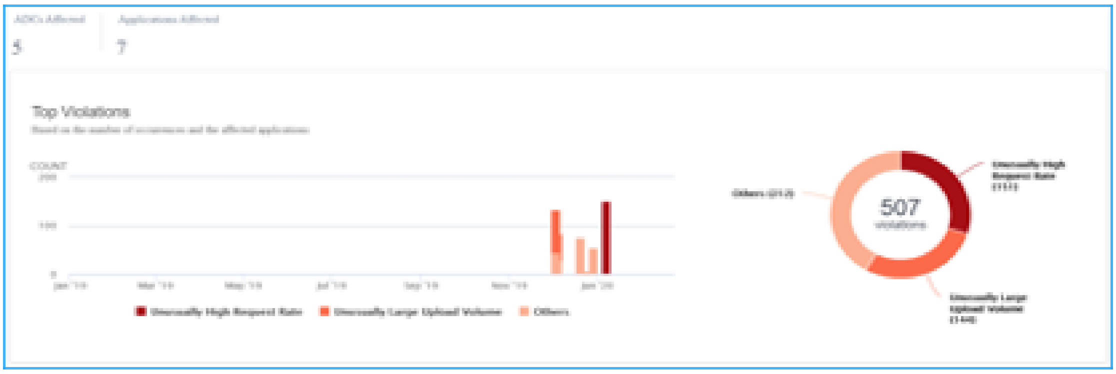
In the security violations dashboard, users can view:

- Total violations occurred across all NetScaler and applications. The total violations are displayed based on the selected time duration.

- Total violations under each category.



- Total ADCs affected, total applications affected, and top violations based on the total occurrences and the affected applications.



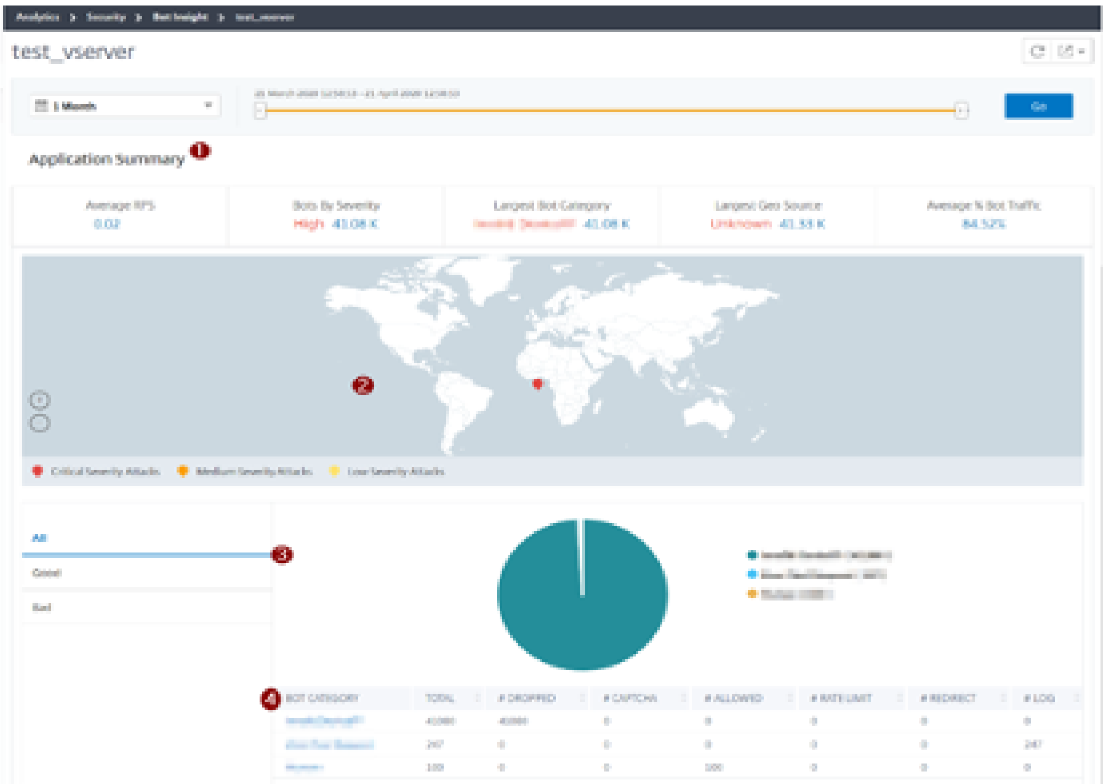
For more information on violations details, see All violations.

Bot insight

Configure BOT insight in NetScaler. For more information, see Bot.

View bots

Click the virtual server to view the **Application Summary**



1. Provides the Application Summary details such as:

   - **Average RPS** –Indicates the average bot transaction requests per second (RPS) received on virtual servers.
   - **Bots by Severity** –Indicates that the highest bot transactions occurred based on the severity. The severity is categorized based on **Critical**, **High**, **Medium**, and **Low**.

   For example, if the virtual servers have 11770 high severity bots and 1550 critical severity bots, then NetScaler ADM displays **Critical 1.55 K** under **Bots by Severity**.

   - **Largest Bot Category** –Indicates that the highest bot attacks occurred based on the bot category.

   For example, if the virtual servers have 8000 block-listed bots, 5000 allow listed bots, and 10000 Rate Limit Exceeded bots, then NetScaler ADM displays **Rate Limit Exceeded 10 K** under **Largest Bot Category**.

   - **Largest Geo Source** –Indicates that the highest bot attacks occurred based on a region.

   For example, if the virtual servers have 5000 bot attacks in Santa Clara, 7000 bot attacks in London, and 9000 bot attacks in Bangalore, then NetScaler ADM displays **Bangalore 9 K** under **Largest Geo Source**.

   - **Average % Bot Traffic** –Indicates the human bot ratio.

2. Displays the severity of the bot attacks based on locations in the map view
3. Displays the types of bot attacks (Good, Bad, and all)
4. Displays the total bot attacks along with the corresponding configured actions. For example, if you have configured:
   - IP address range (192.140.14.9 to 192.140.14.254) as block list bots and selected Drop as an action for these IP address ranges
   - IP range (192.140.15.4 to 192.140.15.254) as block list bots and selected to create a log message as an action for these IP ranges

     In this scenario, NetScaler ADM displays:

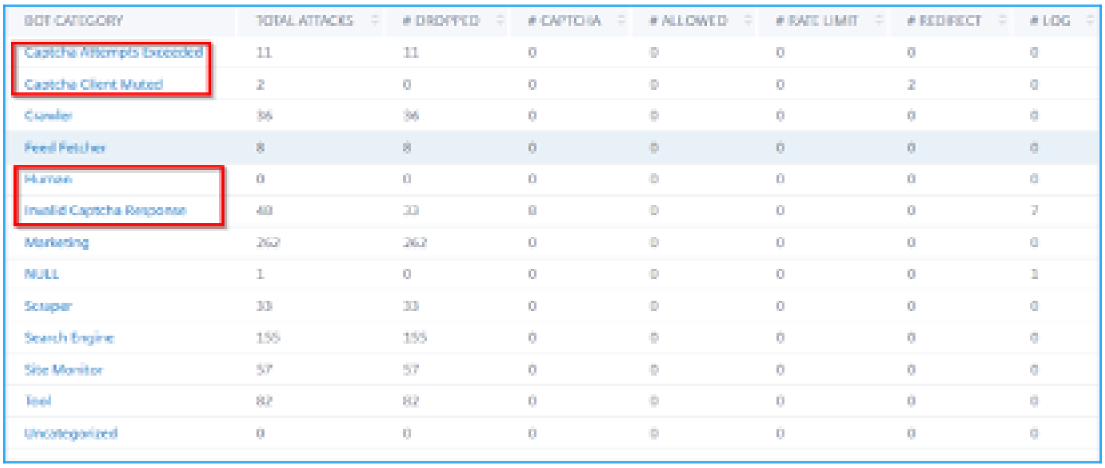
     - Total block listed bots
     - Total bots under **Dropped**
     - Total bots under Log

View CAPTCHA bots

In webpages, CAPTCHAs are designed to identify if the incoming traffic is from a human or an automated bot. To view the CAPTCHA activities in NetScaler ADM, users must configure CAPTCHA as a bot action for IP reputation and device fingerprint detection techniques in a NetScaler ADM instance. For more information, see: Configure Bot Management.

The following are the CAPTCHA activities that NetScaler ADM displays in Bot insight:

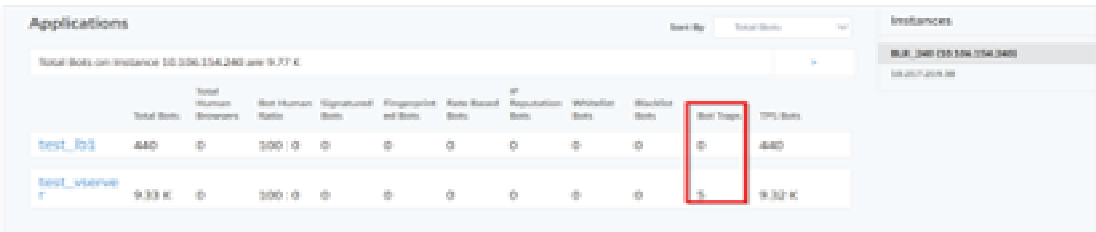
- **Captcha attempts exceeded** –Denotes the maximum number of CAPTCHA attempts made after login failures
- **Captcha client muted** –Denotes the number of client requests that are dropped or redirected because these requests were detected as bad bots earlier with the CAPTCHA challenge
- **Human** –Denotes the captcha entries performed from the human users
- **Invalid captcha response** –Denotes the number of incorrect CAPTCHA responses received from the bot or human, when NetScaler sends a CAPTCHA challenge



BOT CATEGORY	TOTAL ATTACKS	# DROPPED	# CAPTCHA	# ALLOWED	# RATE LIMIT	# REDIRECT	# LOG
Captcha Attempts Exceeded	11	11	0	0	0	0	0
Captcha Client Muted	2	0	0	0	0	2	0
Crawler	36	36	0	0	0	0	0
Feed Fetcher	8	8	0	0	0	0	0
Human	0	0	0	0	0	0	0
Invalid Captcha Response	40	33	0	0	0	0	7
Marketing	262	262	0	0	0	0	0
NULL	1	0	0	0	0	0	1
Scraper	33	33	0	0	0	0	0
Search Engine	155	155	0	0	0	0	0
Site Monitor	57	57	0	0	0	0	0
Tool	82	82	0	0	0	0	0
Uncategorized	0	0	0	0	0	0	0

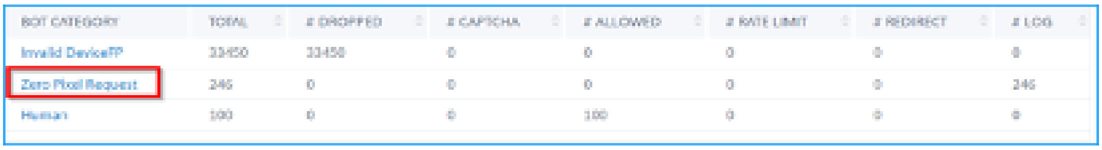
View bot traps

To view bot traps in NetScaler ADM, you must configure the bot trap in NetScaler. For more information, see: Configure Bot Management.



To identify the bot trap, a script is enabled in the webpage and this script is hidden from humans, but not to bots. NetScaler ADM identifies and reports the bot traps, when this script is accessed by the bots.

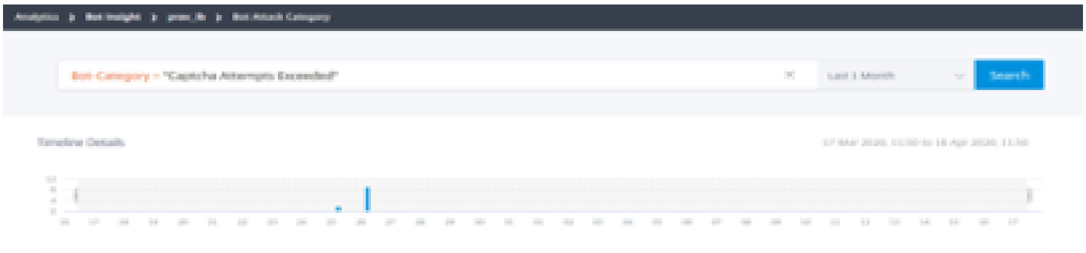
Click the virtual server and select **Zero Pixel Request**



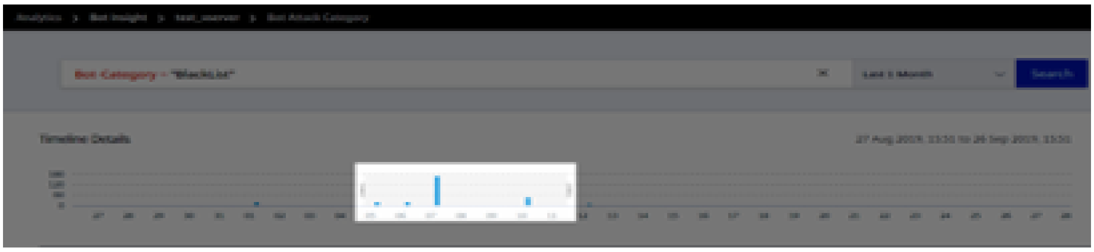
View bot details

For further details, click the bot attack type under **Bot Category**.

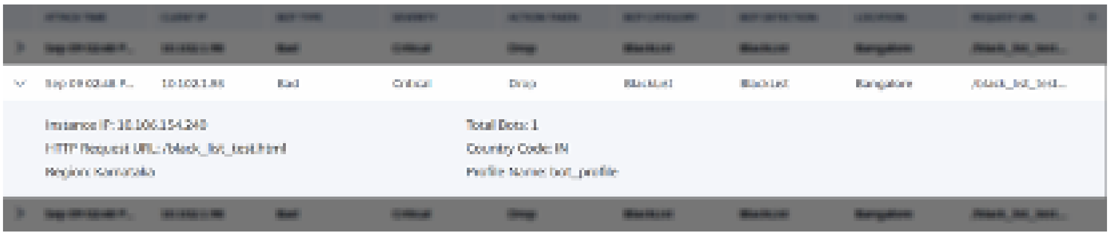
The details such as attack time and total number of bot attacks for the selected captcha category are displayed.



Users can also drag the bar graph to select the specific time range to be displayed with bot attacks.



To get additional information of the bot attack, click to expand.



- **Instance IP** –Indicates the NetScaler instance IP address.
- **Total Bots** –Indicates that the total bot attacks occurred for that particular time.
- **HTTP Request URL** –Indicates the URL that is configured for captcha reporting.
- **Country Code** –Indicates the country where the bot attack occurred.
- **Region** –Indicates the region where the bot attack occurred.
- **Profile Name** –Indicates the profile name that users provided during the configuration.

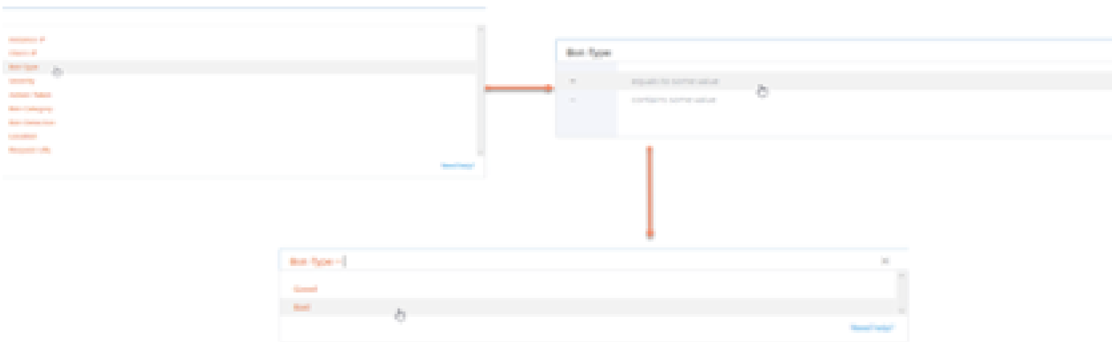
Advanced search

Users can also use the search text box and time duration list, where they can view bot details as per the user requirement. When users click the search box, the search box gives them the following list of search suggestions.

- **Instance IP** –NetScaler instance IP address.
- **Client-IP** –Client IP address.
- **Bot-Type** –Bot type such as Good or Bad.
- **Severity** –Severity of the bot attack.
- **Action-Taken** –Action taken after the bot attack such as Drop, No action, Redirect.
- **Bot-Category** –Category of the bot attack such as block list, allow list, fingerprint. Based on a category, users can associate a bot action to it.
- **Bot-Detection** –Bot detection types (block list, allow list, and so on) that users have configured on NetScaler.
- **Location** –Region/country where the bot attack has occurred
- **Request-URL** –URL that has the possible bot attacks

Users can also use operators in the user search queries to narrow the focus of the user search. For example, if users want to view all bad bots:

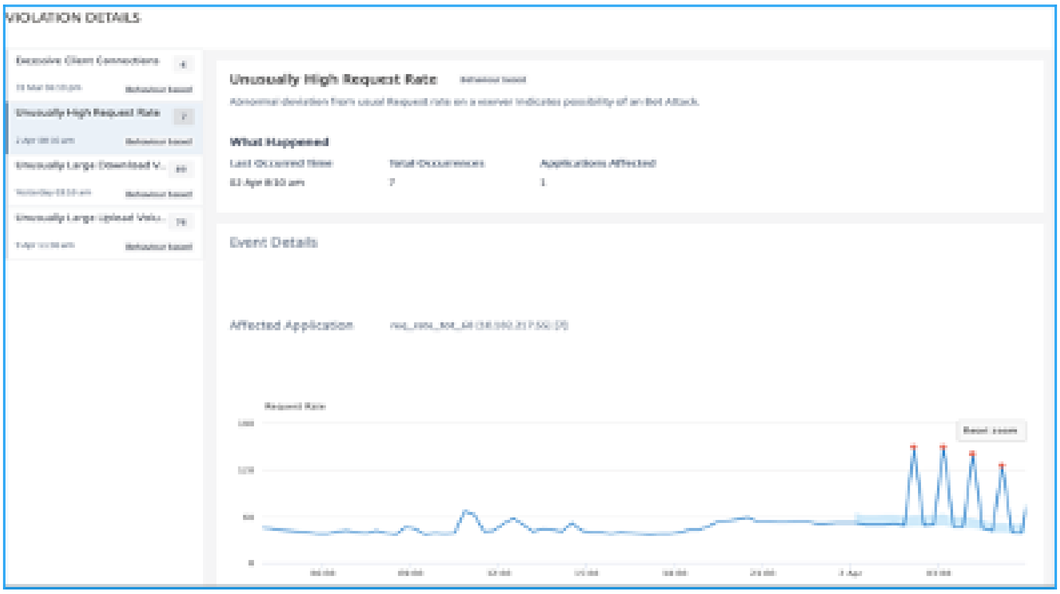
- Click the search box and select **Bot-Type**
- Click the search box again and select the operator **=**
- Click the search box again and select **Bad**
- Click **Search** to display the results



Unusually high request rate

Users can control the incoming and outgoing traffic from or to an application. A bot attack can perform an unusually high request rate. For example, if users configure an application to allow 100 requests/minute and if users observe 350 requests, then it might be a bot attack.

Using the **Unusually High Request Rate** indicator, users can analyze the unusual request rate received to the application.



Under **Event Details**, users can view:

- The affected application. Users can also select the application from the list if two or more applications are affected with violations.
- The graph indicating all violations
- The violation occurrence time
- The detection message for the violation, indicating the total requests received and % of excessive requests received than the expected requests
- The accepted range of expected request rates range from the application

Bot Detection

The NetScaler bot management system uses various techniques to detect the incoming bot traffic. The techniques are used as detection rules to detect the bot type.

Configuring Bot management by using GUI Users can configure NetScaler bot management by first enabling the feature on the appliance. For more information, see Bot Detection.

IP reputation

IP reputation is a tool that identifies IP addresses that send unwanted requests. Using the IP reputation list you can reject requests that are coming from an IP address with a bad reputation.

Configure IP reputation by using GUI This configuration is a prerequisite for the bot IP reputation feature. For more information, see IP Reputation.

Auto update for Bot signatures The bot static signature technique uses a signature lookup table with a list of good bots and bad bots. For more information, see Signature auto update.

NetScaler Web App Firewall and OWASP top ten–2021

The Open Web Application Security Project(OWAP) released the OWASP Top 10 for 2021 for web application security. This list documents the most common web application vulnerabilities and is a great starting point to evaluate web security. This section explains on how to configure the NetScaler Web App Firewall to mitigate these flaws. WAF is available as an integrated module in the NetScaler (Premium Edition) and a complete range of appliances.

The full OWASP Top 10 document is available at OWASP Top Ten.

OWASP Top-10 2021	NetScaler Web App Firewall Features
A1:2021 Broken Access Control	AAA, Authorization security features within AAA module of NetScaler, Form protections, and cookie tampering protections, StartURL, and ClosureURL
A2:2021 - Cryptographic Failures	Credit Card protection, Safe Commerce, Cookie proxying, and Cookie encryption
A3:2021- Injection	Injection attack prevention (SQL or any other custom injections such as OS Command injection, XPath injection, and LDAP injection), auto update signature feature
A5:2021 Security Misconfiguration	This protection including WSI checks, XML message validation & XML SOAP fault filtering check
A6:2021 - Vulnerability and Outdated Components	Vulnerability scan reports, Application Firewall Templates, and Custom Signatures

OWASP Top-10 2021	NetScaler Web App Firewall Features
A7:2021 - Identification and Authentication Failure	AAA, Cookie tampering protection, Cookie proxying, Cookie encryption, CSRF tagging, Use SSL
A8:2021 –Software and Data Integrity Failures	XML Security checks, GWT content type, custom signatures, Xpath for JSON and XML
A9:2021 –Security Logging and Monitoring Failures	User configurable custom logging, Management and Analytics System

A1:2021 Broken Access Control

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and data, such as access other users' accounts, view sensitive files, modify other users'data, change access rights.

NetScaler Web App Firewall protections

- AAA feature that supports authentication, authorization, and auditing for all application traffic allows a site administrator to manage access controls with the ADC appliance.
- The Authorization security feature within the AAA module of the ADC appliance enables the appliance to verify, which content on a protected server it should allow each user to access.
- Form field consistency: If object references are stored as hidden fields in forms, then using form field consistency you can validate that these fields are not tampered on subsequent requests.
- Cookie proxying and cookie consistency: Object references that are stored in cookie values can be validated with these protections.
- Start URL check with URL closure: Allows user access to a predefined allow list of URLs. URL closure builds a list of all URLs seen in valid responses during the user session and automatically allows access to them during that session.

A2:2021 - Cryptographic failures

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such poorly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

NetScaler Web App Firewall protections

- Web Application Firewall protects applications from leaking sensitive data like credit card details.
- Sensitive data can be configured as safe objects in Safe Commerce protection to avoid exposure.
- Any sensitive data in cookies can be protected by Cookie proxying and Cookie encryption.

A3:2021- Injection

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into running unintended commands or accessing data without proper authorization.

XSS flaws occur whenever an application includes untrusted data in a new webpage without proper validation or escaping, or updates an existing webpage with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to run scripts in the victim's browser, which can hijack user sessions, deface websites, or redirect the user to malicious sites.

NetScaler Web App Firewall protections

- SQL injection prevention feature protects against common injection attacks. Custom injection patterns can be uploaded to protect against any type of injection attack, including XPath and LDAP. This is applicable for both HTML and XML payloads.
- The auto update signature feature keeps the injection signatures up to date.
- The field format protection feature allows the administrator to restrict any user parameter to a regular expression. For instance, you can enforce that a zip-code field contains integers only or even 5-digit integers.
- Form field consistency validates each submitted user form against the user session form signature to ensure the validity of all form elements.
- Buffer overflow checks ensure that the URL, headers, and cookies are in the right limits blocking any attempts to inject large scripts or code.
- XSS protection protects against common XSS attacks. Custom XSS patterns can be uploaded to modify the default list of allowed tags and attributes. The ADC WAF uses a white list of allowed HTML attributes and tags to detect XSS attacks. This is applicable for both HTML and XML payloads.
- ADC WAF blocks all the attacks listed in the OWASP XSS Filter Evaluation Cheat Sheet.

- Field format check prevents an attacker from sending inappropriate web form data, which can be a potential XSS attack.
- Form field consistency.

A5:2021 - Security misconfiguration

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or improvised configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

NetScaler Web App Firewall protections

- The PCI-DSS report generated by the Application Firewall, documents the security settings on the Firewall device.
- Reports from the scanning tools are converted to ADC WAF signatures to handle security misconfigurations.
- NetScaler Web App Firewall Web Application Firewall supports Cenzic, IBM AppScan (Enterprise and Standard), Qualys, TrendMicro, WhiteHat, and custom vulnerability scan reports.
- In addition to detecting and blocking common application threats that can be adapted for attacking XML-based applications (that is, cross-site scripting, command injection, and so on).
- NetScaler Web App Firewall Web Application Firewall includes a rich set of XML-specific security protections. These include schema validation to thoroughly verify SOAP messages and XML payloads, and a powerful XML attachment check to block attachments containing malicious executables or viruses.
- Automatic traffic inspection methods block XPath injection attacks on URLs and forms aimed at gaining access.
- NetScaler Web App Firewall Web Application Firewall also thwarts various DoS attacks, including external entity references, recursive expansion, excessive nesting, and malicious messages containing either long or many attributes and elements.

A6:2021 - Vulnerable and outdated components

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

NetScaler Web App Firewall protections

- We recommend having the third-party components up to date.
- Vulnerability scan reports that are converted to ADC signatures can be used to virtually patch these components.
- Application firewall templates that are available for these vulnerable components can be used.
- Custom signatures can be bound with the firewall to protect these components.

A7:2021–Broken authentication

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users'identities temporarily or permanently.

NetScaler Web App Firewall protections

- NetScaler AAA module performs user authentication and provides Single Sign-On functionality to back-end applications. This is integrated into the NetScaler AppExpert policy engine to allow custom policies based on user and group information.
- Using SSL offloading and URL transformation capabilities, the firewall can also help sites to use secure transport layer protocols to prevent stealing of session tokens by network sniffing.
- Cookie proxying and cookie encryption can be employed to completely mitigate cookie stealing.

A8:2021 - Software and data integrity failure

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

NetScaler Web App Firewall protections

- JSON payload inspection with custom signatures.
- XML security: protects against XML denial of service (xDoS), XML SQL and Xpath injection and cross-site scripting, format checks, WS-I basic profile compliance, XML attachments check.
- Field format checks and Cookie Consistency and Field Consistency can be used.

A9:2021 - Security logging and monitoring failures

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show the time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

NetScaler Web App Firewall protections

- When the log action is enabled for security checks or signatures, the resulting log messages provide information about the requests and responses that the application firewall has observed while protecting your websites and applications.
- The application firewall offers the convenience of using the built-in ADC database for identifying the locations corresponding to the IP addresses from which malicious requests are originating.
- Default format (PI) expressions give the flexibility to customize the information included in the logs with the option to add the specific data to capture in the application firewall-generated log messages.
- The application firewall supports CEF logs.

References

- HTML SQL Injection Check
- XML SQL Injection Check
- Using the Command Line to Configure the HTML Cross-Site Scripting Check
- XML Cross-Site Scripting Check
- Using the Command Line to Configure the Buffer Overflow Security Check
- Adding or Removing a Signature Object
- Configuring or Modifying a Signatures Object

- Updating a Signature Object
- Snort Rule Integration
- Bot Detection
- Deploy a NetScaler VPX instance on Microsoft Azure

Configure address pools intranet IP for a NetScaler Gateway appliance

In some situations, users who connect with the NetScaler Gateway Plug-in need a unique IP address for a NetScaler Gateway appliance. When you enable address pools (also known as IP pooling) for a group, the NetScaler Gateway appliance can assign a unique IP address alias to each user. You configure address pools by using intranet IP (IIP) addresses.

You can configure address pools on a NetScaler Gateway appliance deployed on Azure by following this 2-step procedure:

- Registering the private IP addresses that are used in the address pool, in Azure
- Configuring address pools in the NetScaler Gateway appliance

Register a private IP address in the Azure portal

In Azure, you can deploy a NetScaler VPX instance with multiple IP addresses. You can add IP addresses to a VPX instance in two ways:

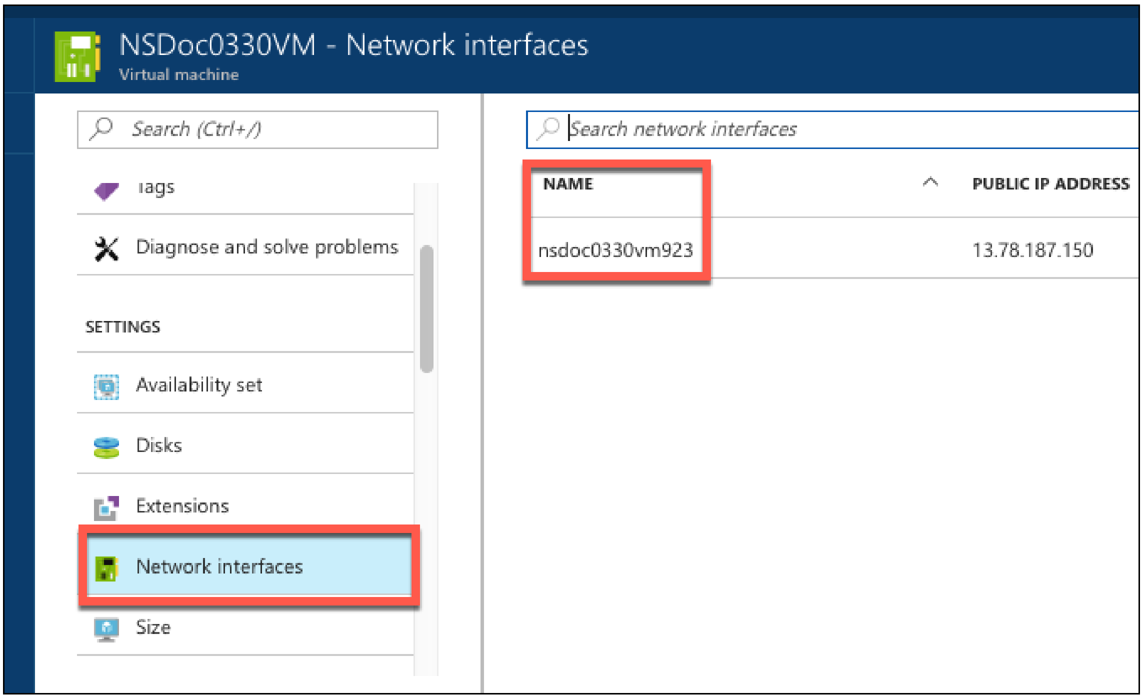
a. While provisioning a VPX instance

For more information about how to add multiple IP addresses while provisioning a VPX instance, see Configure multiple IP addresses for a NetScaler standalone instance. To add IP addresses by using PowerShell commands while provisioning a VPX instance, see Configure multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands.

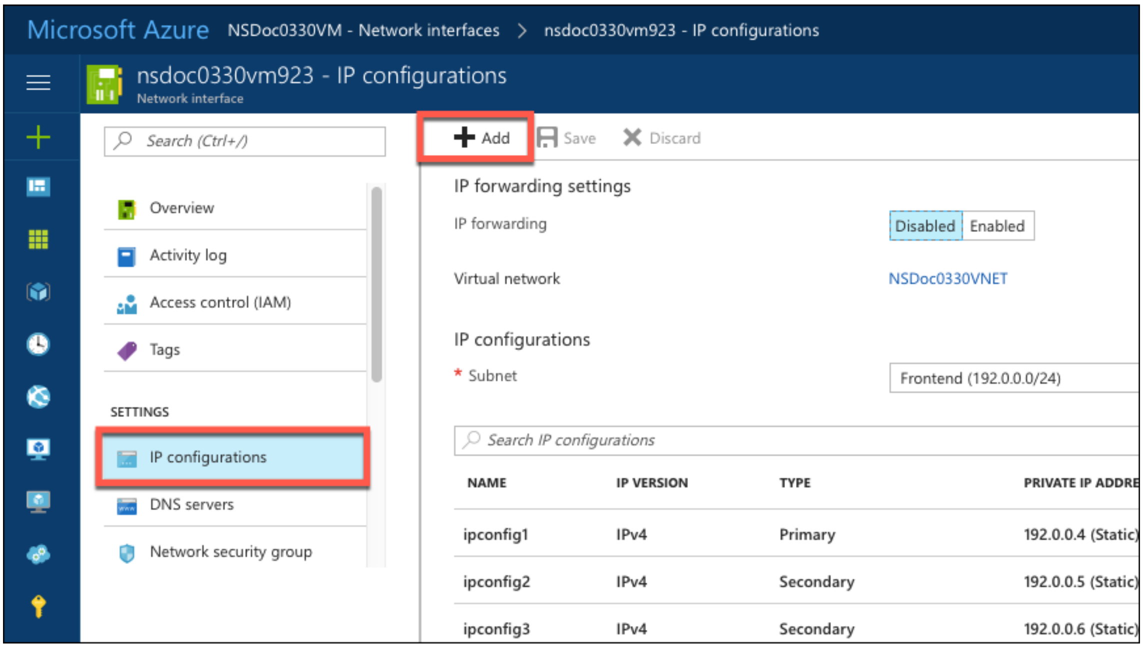
b. After provisioning a VPX instance

After you've provisioned a VPX instance, follow these steps to register a private IP address in the Azure portal, which you configure as an address pool in the NetScaler Gateway appliance.

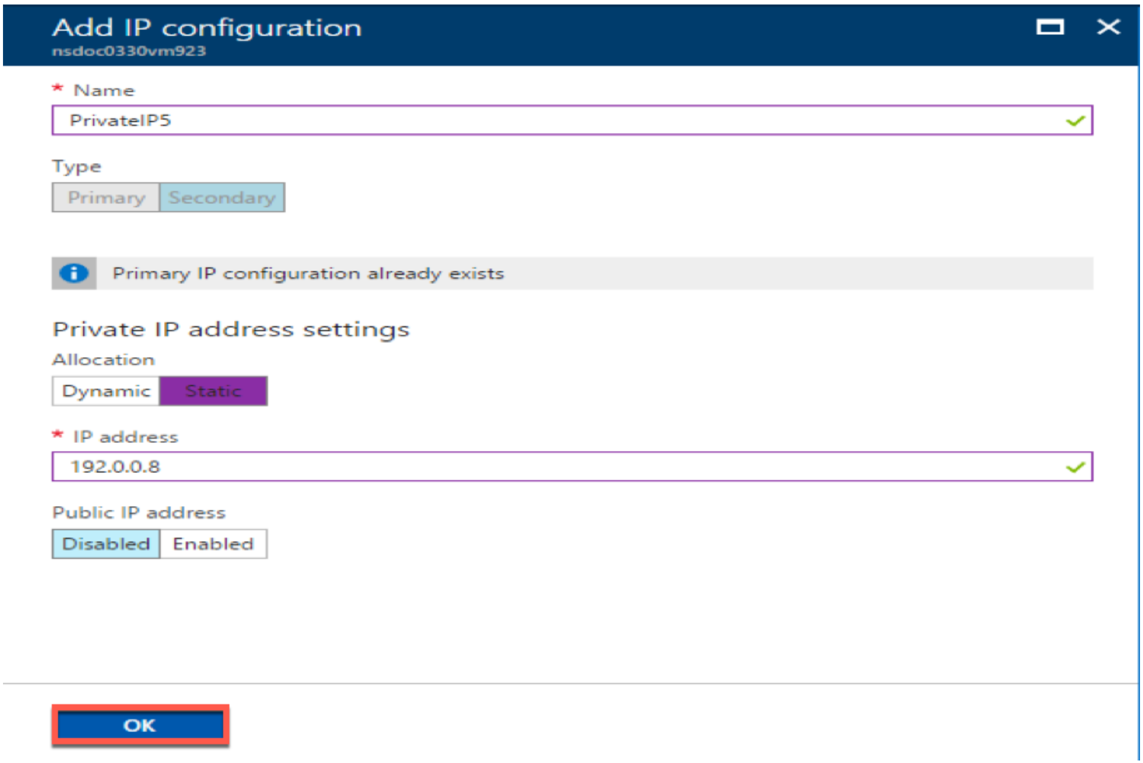
1. From Azure Resource Manager (ARM), go to the already created NetScaler VPX instance > **Network interfaces**. Choose the network interface which is bound to a subnet to which the IIP that you want to register belongs.



2. Click **IP Configurations**, and then click **Add**.



3. Provide the required details as shown in the example below and click **OK**.



Configure address pools in the NetScaler Gateway appliance

For more information about how to configure address pools on the NetScaler Gateway, see Configuring Address Pools.

> **Limitation:**
>
> You cannot bind a range of IIP addresses to users. Every IIP address that is used in an address pool must be registered.

Configure multiple IP addresses for a NetScaler VPX standalone instance by using PowerShell commands

In an Azure environment, a NetScaler VPX virtual appliance can be deployed with multiple NICs. Each NIC can have multiple IP addresses. This section describes how to deploy a NetScaler VPX instance with a single NIC and multiple IP addresses, by using PowerShell commands. You can use the same script for multi-NIC and multi-IP deployment.

> **Note:**
>
> In this document, IP-Config refers to a pair of IP addresses, public IP, and private IP, that is asso-

> ciated with an individual NIC. For more information, see the Azure terminology section.

Use case

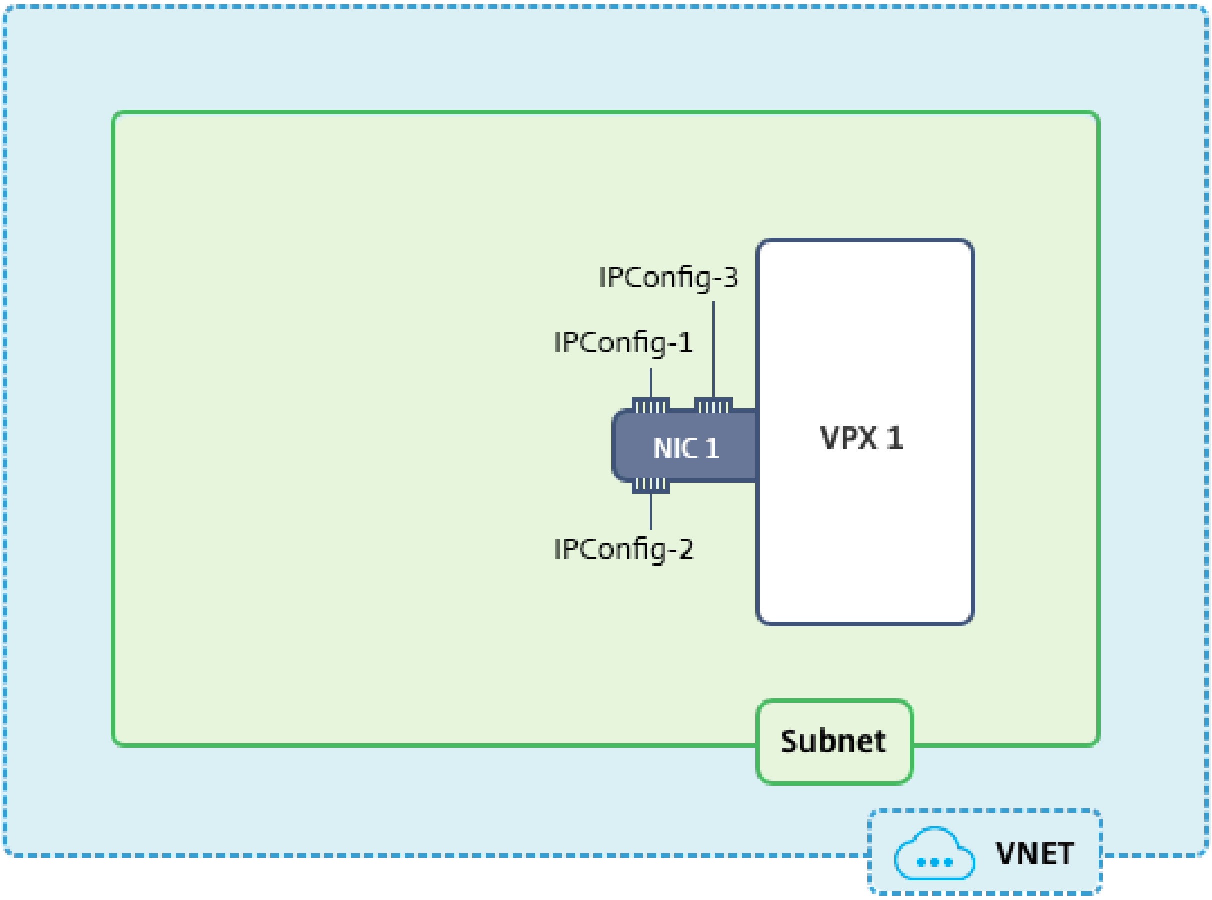
In this use case, a single NIC is connected to a virtual network (VNET). The NIC is associated with three IP configurations, as shown in the following table.

IP Config	Associated with
IPConfig-1	Static public IP address; static private IP address
IPConfig-2	Static public IP address; static private address
IPConfig-3	Static private IP address

> **Note:**
>
> IPConfig-3 is not associated with any public IP address.

Diagram: Topology

Here is the visual representation of the use case.



> **Note:**
>
> In a multi-NIC, multi-IP Azure NetScaler VPX deployment, the private IP address associated with the primary (first) `IPConfig` of the primary (first) NIC is automatically added as the management NSIP address of the appliance. The remaining private IP addresses associated with `IPConfigs` must be added in the VPX instance as VIPs or SNIPs by using the `add ns ip` command, as determined by your requirements.

Here is the summary of the steps required for configuring multiple IP addresses for a NetScaler VPX virtual appliance in standalone mode:

1. Create Resource Group
2. Create Storage Account
3. Create Availability Set
4. Create Network service group
5. Create Virtual Network
6. Create Public IP Address
7. Assign IP Configuration
8. Create NIC
9. Create NetScaler VPX Instance
10. Check NIC Configurations
11. Check VPX-side Configurations

Script

Parameters

Following are sample parameters settings for the use case in this document.

```
$locName="westcentralus"

$rgName="Azure-MultiIP"

$nicName1="VM1-NIC1"

$vNetName="Azure-MultiIP-vnet"

$vNetAddressRange="11.6.0.0/16"

$frontEndSubnetName="frontEndSubnet"

$frontEndSubnetRange="11.6.1.0/24"

$prmStorageAccountName="multiipstorage"

```



```
$avSetName="multiip-avSet"

$vmSize="Standard\_DS4\_V2" (This parameter creates a VM with up to
    four NICs.)
```

> **Note:**
>
> The minimum requirement for a VPX instance is 2 vCPUs and 2 GB RAM.

```
$publisher="Citrix"

$offer="netscalervpx110-6531" (You can use different offers.)

$sku="netscalerbyol" (According to your offer, the SKU can be different
    .)

$version="latest"

$pubIPName1="PIP1"

$pubIPName2="PIP2"

$domName1="multiipvpx1"

$domName2="multiipvpx2"

$vmNamePrefix="VPXMultiIP"

$osDiskSuffix="osmultiipalbdiskdb1"

**Network Security Group (NSG)-related information**:

$nsgName="NSG-MultiIP"

$rule1Name="Inbound-HTTP"

$rule2Name="Inbound-HTTPS"

$rule3Name="Inbound-SSH"

$IpConfigName1="IPConfig1"

$IPConfigName2="IPConfig-2"

$IPConfigName3="IPConfig-3"
```

1. Create Resource Group

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2. Create Storage Account

```
$prmStorageAccount = New-AzureRMStorageAccount -Name $prmStorageAccountName
 -ResourceGroupName $rgName -Type Standard_LRS -Location $locName
```

3. Create Availability Set

```
$avSet = New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
 $rgName -Location $locName
```

4. Create Network Security Group

1. Add rules. You must add a rule to the network security group for any port that serves traffic.

   ```
   $rule1=New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -
   Description "Allow HTTP"-Access Allow -Protocol Tcp -Direction
   Inbound -Priority 101 -SourceAddressPrefix Internet -SourcePortRange
    * -DestinationAddressPrefix * -DestinationPortRange 80
   $rule2=New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -
   Description "Allow HTTPS"-Access Allow -Protocol Tcp -Direction
   Inbound -Priority 110 -SourceAddressPrefix Internet -SourcePortRange
    * -DestinationAddressPrefix * -DestinationPortRange 443
   $rule3=New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name
   -Description "Allow SSH"-Access Allow -Protocol Tcp -Direction
   Inbound -Priority 120 -SourceAddressPrefix Internet -SourcePortRange
    * -DestinationAddressPrefix * -DestinationPortRange 22
   ```

2. Create network security group object.

   ```
   $nsg=New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName
   -Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,
   $rule3
   ```

5. Create Virtual Network

1. Add subnets.

   ```
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
   $frontEndSubnetName -AddressPrefix $frontEndSubnetRange
   ```

2. Add virtual network object.


```
$vnet=New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
 $rgName -Location $locName -AddressPrefix $vNetAddressRange -
Subnet $frontendSubnet
```

3. Retrieve subnets.

```
$subnetName="frontEndSubnet"
$subnet1=$vnet.Subnets|?{ $_.Name -eq $subnetName }
```

6. Create Public IP Address

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
 $rgName -DomainNameLabel $domName1 -Location $locName -AllocationMethod
 Static
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
 $rgName -DomainNameLabel $domName2 -Location $locName -AllocationMethod
 Static
```

> **Note:**
>
> Check availability of domain names before using.
>
> Allocation method for IP addresses can be dynamic or static.

7. Assign IP Configuration

In this use case, consider the following points before assigning IP addresses:

- IPConfig-1 belongs to subnet1 of VPX1.
- IPConfig-2 belongs to subnet 1 of VPX1.
- IPConfig-3 belongs to subnet 1 of VPX1.

> **Note:**
>
> When you assign multiple IP configurations to a NIC, one configuration must be assigned as primary.

```
$IPAddress1="11.6.1.27"
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress1 -PublicIpAddress $pip1
      – Primary
$IPAddress2="11.6.1.28"
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress2 -PublicIpAddress $pip2
$IPAddress3="11.6.1.29"
```



```
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress3 -Primary
```

Use a valid IP address that meets your subnet requirements and check its availability.

8. Create NIC

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
 $rgName -Location $locName -IpConfiguration $IpConfig1,$IpConfig2,
$IPConfig3 -NetworkSecurityGroupId $nsg.Id
```

9. Create NetScaler VPX Instance

1. Initialize variables.

   ```
   $suffixNumber = 1
   $vmName = $vmNamePrefix + $suffixNumber
   ```

2. Create VM config object.

   ```
   $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
   AvailabilitySetId $avSet.Id
   ```

3. Set credentials, OS, and image.

   ```
   $cred=Get-Credential -Message "Type the name and password for VPX
    login."
   $vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
   ComputerName $vmName -Credential $cred
   $vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
   $publisher -Offer $offer -Skus $sku -Version $version
   ```

4. Add NIC.

   ```
   $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.
   Id -Primary
   ```

   > **Note:**
   >
   > In a multi-NIC VPX deployment, one NIC must be primary. So, "-Primary"must be appended while adding that NIC to the VPX instance.

5. Specify OS disk and create VM.

   ```
   $osDiskName=$vmName + "-"+ $osDiskSuffix1
   $osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString()+ "
   vhds/"+ $osDiskName + ".vhd"
   ```



```
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -
VhdUri $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product
$offer -Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
$locName
```

10. Check NIC Configurations

After the VPX instance starts, you can check the IP addresses allocated to `IPConfigs` of the VPX NIC by using the following command.

`$nic.IPConfig`

11. Check VPX-side Configurations

When the NetScaler VPX instance starts, a private IP address associated with primary `IPconfig` of the primary NIC is added as the NSIP address. The remaining private IP addresses must be added as VIP or SNIP addresses, as determined by your requirements. Use the following command.

`add nsip <Private IPAddress><netmask> -type VIP/SNIP`

You've now configured multiple IP addresses for a NetScaler VPX instance in standalone mode.

Additional PowerShell scripts for Azure deployment

This section provides the PowerShell cmdlets with which you can perform the following configurations in Azure PowerShell:

- Provision a NetScaler VPX standalone instance
- Provision a NetScaler VPX pair in a high availability setup with an Azure external load balancer
- Provision a NetScaler VPX pair in a high availability setup with Azure internal load balancer

Also see the following topics for configurations that you can perform by using PowerShell commands:

- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands
- Configure GSLB on NetScaler VPX instances
- Configure GSLB on a NetScaler active-standby high-availability setup
- Configure multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands

Provision a NetScaler VPX standalone instance

1. **Create a resource group**

   The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. The location specified here is the default location for resources in that resource group. Make sure all commands to create a load balancer use the same resource group.

   ```
   $rgName="<resource group name>"
   $locName="<location name, such as West US>
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   For example:

   ```
   $rgName = "ARM-VPX"
   $locName = "West US"
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

2. **Create a storage account**

   Choose a unique name for your storage account that contains only lowercase letters and numbers.

   `$saName="<storage account name>"`

   `$saType="<storage account type>"`, specify one: `Standard_LRS`, `Standard_GRS`, `Standard_RAGRS`, or `Premium_LRS`

   ```
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName
   $rgName -Type $saType -Location $locName
   ```

   For example:

   ```
   $saName="vpxstorage"
   $saType="Standard\_LRS"
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
       -Type $saType -Location $locName
   ```

3. **Create an availability set**

   Availability set helps to keep your virtual machines available during downtime, such as during maintenance. A load balancer configured with an availability set ensures that your application is always available.

   ```
   $avName="<availability set name>"
   ```

   ```
   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
   $rgName -Location $locName
   ```

4. **Create a virtual network**

Add a new virtual network with at least one subnet, if the subnet was not created previously.

```
$FrontendAddressPrefix="10.0.1.0/24"
$BackendAddressPrefix="10.0.2.0/24"
$vnetAddressPrefix="10.0.0.0/16"
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
frontendSubnet -AddressPrefix $FrontendAddressPrefix
$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
backendSubnet -AddressPrefix $BackendAddressPrefix
New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName
$rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
Subnet $frontendSubnet,$backendSubnet
```

For example:

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
   frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
   backendSubnet -AddressPrefix $BackendAddressPrefix

New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName
    -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet
   $frontendSubnet,$backendSubnet
```

5. **Create a NIC**

Create a NIC and associate the NIC with the NetScaler VPX instance. The front end Subnet created in the above procedure is indexed at 0 and the back end Subnet is indexed at 1. Now create NIC in one of the three following ways:

a) *NIC with Public IP address*

```
$nicName="<name of the NIC of the VM>"
```

```
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
 $rgName -Location $locName -AllocationMethod Dynamic
```

```
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
 $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex
].Id -PublicIpAddressId $pip.Id
```

b) *NIC with Public IP and DNS label*

```
$nicName="<name of the NIC of the VM>"
```

```
$domName="<domain name label>"
```

```
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
 $rgName -DomainNameLabel $domName -Location $locName -AllocationMethod
```


`Dynamic`

Before assigning $domName, check it is available or not by using command:

```
Test-AzureRmDnsAvailability -DomainQualifiedName $domName -
Location $locName
```

```
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
 $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex
].Id -PublicIpAddressId $pip.Id
```

For example:

```
$nicName="frontendNIC"

$domName="vpxazure"

$pip = New-AzureRmPublicIpAddress -Name $nicName -
    ResourceGroupName $rgName -DomainNameLabel $domName -Location
    $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -SubnetId $vnet.
    Subnets\[0\].Id -PublicIpAddressId $pip.Id
```

c) *NIC with Dynamic Public Address and Static Private IP address*

Make sure that the private (static) IP address you add to the VM must be the same range as that of the subnet specified.

```
$nicName="<name of the NIC of the VM>"
```

```
$staticIP="<available static IP address on the subnet>"
```

```
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
 $rgName -Location $locName -AllocationMethod Dynamic
```

```
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
 $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex
].Id -PublicIpAddressId $pip.Id -PrivateIpAddress $staticIP
```

6. **Create a virtual object**

```
$vmName="<VM name>"
```

```
$vmSize="<VM size string>"
```

```
$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
 $rgName
```

```
$vm=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
 $avset.Id
```


7. **Get the NetScaler VPX image**

   `$pubName="<Image publisher name>"`

   `$offerName="<Image offer name>"`

   `$skuName="<Image SKU name>"`

   `$cred=Get-Credential -Message "Type the name and password of the local administrator account."`

   Provide your credentials that is used to log in into VPX

   `$vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName $vmName -Credential $cred -Verbose`

   `$vm=Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"`

   `$vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id`

   For example:

   `$pubName="citrix"`

   The following command is used for displaying all offers from Citrix:

```
Get-AzureRMVMImageOffer -Location $locName -Publisher $pubName |
    Select Offer

$offerName="netscalervpx110-6531"
```

   The following command is used to know SKU offered by publisher for specific offer name:

   `Get-AzureRMVMImageSku -Location $locName -Publisher $pubName -Offer $offerName | Select Skus`

8. **Create a virtual machine**

   `$diskName="<name identifier for the disk in Azure storage, such as OSDisk>"`

   For example:

```
$diskName="dynamic"

$pubName="citrix"

$offerName="netscalervpx110-6531"

$skuName="netscalerbyol"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName
```



```
$osDiskUri=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds/"
    + $diskName  + ".vhd"

$vm=Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri
     -CreateOption fromImage
```

When you create VM from Images present in marketplace, use the following command to specify the VM plan:

```
Set-AzureRmVMPlan -VM $vm -Publisher $pubName -Product $offerName
 -Name $skuName
```

```
New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
$vm
```

Provision a NetScaler VPX pair in a high availability setup with an Azure external load balancer

Log on to AzureRmAccount using your Azure user credentials.

1. **Create a resource group**

   The location specified here is the default location for resources in that resource group. Make sure that all commands used to create a load balancer use the same resource group.

   ```
   $rgName="<resource group name>"
   ```

   ```
   $locName="<location name, such as West US>"
   ```

   ```
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   For example:

   ```
   $rgName = "ARM-LB-NS"

   $locName = "West US"

   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

2. **Create a storage account**

   Choose a unique name for your storage account that contains only lowercase letters and numbers.

   ```
   $saName="<storage account name>"
   ```

   $saType="<storage account type>", specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS

`New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName`

For example:

```
$saName="vpxstorage"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
```

3. **Create an availability set**

   A load balancer configured with an availability set ensures that your application is always available.

   `$avName="<availability set name>"`

   `New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName`

4. **Create a virtual network**

   Add a new virtual network with at least one subnet, if the subnet was not created previously.

```
$vnetName = "LBVnet"

$FrontendAddressPrefix="10.0.1.0/24"

$BackendAddressPrefix="10.0.2.0/24"

$vnetAddressPrefix="10.0.0.0/16"

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
   frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
   backendSubnet -AddressPrefix $BackendAddressPrefix

$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
    $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
   Subnet $frontendSubnet,$backendSubnet
```

> **Note:**
>
> Choose the AddressPrefix parameter value as per your requirement.

   Assign front end and back end subnet to the virtual network that you created earlier in this step.

   If the front end subnet is the first element of array VNet, subnetId must be $vnet.Subnets[0].Id.

   If the front end subnet is the second element in the array, the subnetId must be $vnet.Subnets[1].Id, and so on.

5. **Configure front end IP address and create back end address pool**

   Configure a front end IP address for the incoming load balancer network traffic and create a back end address pool to receive the load balanced traffic.

   ```
   $pubName="PublicIp1"

   $publicIP1 = New-AzureRmPublicIpAddress -Name $pubName -
       ResourceGroupName $rgName -Location $locName -AllocationMethod
       Static -DomainNameLabel nsvpx
   ```

   > **Note:**
   >
   > Check for the availability of the value for DomainNameLabel.

   ```
   $FIPName = "ELBFIP"

   $frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name
       $FIPName -PublicIpAddress $publicIP1

   $BEPool = "LB-backend-Pool"

   $beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -
       Name $BEPool
   ```

6. **Create a health probe**

   Create a TCP health probe with port 9000 and interval 5 seconds.

   ```
   $healthProbe = New-AzureRmLoadBalancerProbeConfig -Name
       HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -
       ProbeCount 2
   ```

7. **Create a load balancing rule**

   Create an LB rule for each service that you are load balancing.

   For example:

   You can use the following example to load balance HTTP service.

   ```
   $lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP-LB" -
       FrontendIpConfiguration $frontendIP1 -BackendAddressPool
       $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort
        80 -BackendPort 80
   ```

8. **Create inbound NAT rules**

   Create NAT rules for services that you are not load balancing.

   For example, when creating an SSH access to a NetScaler VPX instance.

> **Note:**
>
> Protocol-FrontEndPort-BackendPort triplet must not be the same for two NAT rules.

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH1     -FrontendIpConfiguration $frontendIP1 -Protocol
    TCP -FrontendPort 22 -BackendPort 22

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -
    FrontendPort 10022 -BackendPort 22
```

9. **Create a load balancer entity**

   Create the load balancer adding all objects (NAT rules, load balancer rules, probe configurations) together.

```
$lbName="ELB"

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name
    $lbName -Location $locName -InboundNatRule $inboundNATRule1,
    $inboundNATRule2 -FrontendIpConfiguration $frontendIP1 -
    LoadBalancingRule $lbrule1 -BackendAddressPool $beAddressPool1
    -Probe $healthProbe
```

10. **Create a NIC**

    Create two NICs and associate each NIC with each VPX instance

    a) NIC1 with VPX1

    For example:

```
$nicName="NIC1"

$lbName="ELB"

$bePoolIndex=0

\* Rule indexes starts from 0.

$natRuleIndex=0

$subnetIndex=0

\* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
    $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -Subnet $vnet.
    Subnets\[$subnetIndex\] -LoadBalancerBackendAddressPool $lb.
```



```
    BackendAddressPools\[$bePoolIndex\] -LoadBalancerInboundNatRule
     $lb.InboundNatRules\[$natRuleIndex\]
```

b) NIC2 with VPX2

For example:

```
$nicName="NIC2"

$lbName="ELB"

$bePoolIndex=0

$natRuleIndex=1

\* Second Inbound NAT (SSH) rule we need to use

`$subnetIndex=0

\* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
    $rgName

$nic2=New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -Subnet $vnet.
    Subnets\[$subnetIndex\] -LoadBalancerBackendAddressPool $lb.
    BackendAddressPools\[$bePoolIndex\] -LoadBalancerInboundNatRule
      $lb.InboundNatRules\[$natRuleIndex\]
```

11. **Create NetScaler VPX instances**

Create two NetScaler VPX instances as part of the same resource group and availability set, and attach it to the external load balancer.

a) NetScaler VPX instance 1

For example:

```
$vmName="VPX1"

$vmSize="Standard\_A3"

$pubName="citrix"

$offerName="netscalervpx110-6531"

$skuName="netscalerbyol"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id
```



```
$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
```

b) NetScaler VPX instance 2

For example:

```
$vmName="VPX2"

$vmSize="Standard\_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id

```



```
$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
    " + $diskName  + ".vhd"

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
     -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
    $vm2
```

12. **Configure the virtual machines**

    When both the NetScaler VPX instances start, then connect to both NetScaler VPX instances using the SSH protocol to configure the virtual machines.

    a) Active-Active: Run the same set of configuration commands on the command line of both the NetScaler VPX instances.

    b) Active-Passive: Run this command on the command line of both the NetScaler VPX instances.

    `add ha node #nodeID <nsip of other NetScaler VPX>`

    In Active-Passive mode, run configuration commands on the primary node only.

Provision a NetScaler VPX pair in a high availability setup with Azure internal load balancer

Log on to AzureRmAccount using your Azure user credentials.

1. **Create a resource group**

   The location specified here is the default location for resources in that resource group. Make sure all commands to create a load balancer use the same resource group.

   `$rgName="\<resource group name\>"`

   `$locName="\<location name, such as West US\>"`

   `New-AzureRmResourceGroup -Name $rgName -Location $locName`

   For example:

```
$rgName = "ARM-LB-NS"

$locName = "West US"
```



```

New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2. **Create a storage account**

   Choose a unique name for your storage account that contains only lowercase letters and numbers.

   ```
   $saName="<storage account name>"
   ```

   $saType="<storage account type>", specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS

   ```
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
   ```

   For example:

   ```
   $saName="vpxstorage"

   $saType="Standard_LRS"

   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
       -Type $saType -Location $locName
   ```

3. **Create an availability set**

   A load balancer configured with an availability set ensures that your application is always available.

   ```
   $avName="<availability set name>"
   ```

   ```
   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName
   ```

4. **Create a virtual network**

   Add a new virtual network with at least one subnet, if the subnet was not created previously.

   ```
   $vnetName = "LBVnet"

   $vnetAddressPrefix="10.0.0.0/16"

   $FrontendAddressPrefix="10.0.1.0/24"

   $BackendAddressPrefix="10.0.2.0/24"

   $vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
        $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
       Subnet $frontendSubnet,$backendSubnet\`

   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       frontendSubnet -AddressPrefix $FrontendAddressPrefix
   ```



```
$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    backendSubnet -AddressPrefix $BackendAddressPrefix
```

> **Note:**
>
> Choose the AddressPrefix parameter value as per your requirement.

Assign front end and back end subnet to the virtual network that you created earlier in this step.

If the front end subnet is the first element of array VNet, subnetId must be $vnet.Subnets[0].Id.

If the front end subnet is the second element in the array, the subnetId must be $vnet.Subnets[1].Id, and so on.

5. **Create a backend address pool**

   `$beaddresspool= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name "LB-backend"`

6. **Create NAT rules**

   Create NAT rules for services that you are not load balancing.

   ```
   $inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
       Name "Inboundnatrule1" -FrontendIpConfiguration $frontendIP -
       Protocol TCP -FrontendPort 3441 -BackendPort 3389

   $inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
       Name "RDP2" -FrontendIpConfiguration $frontendIP -Protocol TCP
       -FrontendPort 3442 -BackendPort 3389
   ```

   Use front end and back end ports as per your requirement.

7. **Create a health probe**

   Create a TCP health probe with port 9000 and interval 5 seconds.

   ```
   $healthProbe = New-AzureRmLoadBalancerProbeConfig -Name "
       HealthProbe"  " -Protocol tcp -Port 9000 -IntervalInSeconds 5 -
       ProbeCount 2
   ```

8. **Create a load balancing rule**

   Create an LB rule for each service that you are load balancing.

   For example:

   You can use the following example to load balance HTTP service.

   ```
   $lbrule = New-AzureRmLoadBalancerRuleConfig -Name "lbrule1" -
       FrontendIpConfiguration $frontendIP -BackendAddressPool
       $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort
       80 -BackendPort 80
   ```


Use front end and back end ports as per your requirement.

9. **Create a load balancer entity**

   Create the load balancer adding all objects (NAT rules, load balancer rules, probe configurations) together.

```
$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgname -Name
    "InternalLB" -Location $locName -FrontendIpConfiguration
    $frontendIP -InboundNatRule $inboundNATRule1,$inboundNatRule2 -
    LoadBalancingRule $lbrule -BackendAddressPool $beAddressPool -
    Probe $healthProbe
```

10. **Create a NIC**

    Create two NICs and associate each NIC with each NetScaler VPX instance

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic1-be -Location $locName -PrivateIpAddress
    10.0.2.6 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools\[0\] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules\[0\]
```

This NIC is for NetScaler VPX 1. The Private IP must be in same subnet as that of subnet added.

```
$backendnic2= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic2-be -Location $locName -PrivateIpAddress
    10.0.2.7 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools\[0\] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules\[1\].
```

This NIC is for NetScaler VPX 2.The parameter `Private IPAddress` can have any private IP as per your requirement.

11. **Create NetScaler VPX instances**

    Create two VPX instances part of the same resource group and availability set, and attach it to the internal load balancer.

    a) NetScaler VPX instance 1

    For example:

```
$vmName="VPX1"

$vmSize="Standard\_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

```



```
$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $backendnic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
```

b) NetScaler VPX instance 2

For example:

```
$vmName="VPX2"

$vmSize="Standard\_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $backendnic2.Id

$diskName="dynamic"
```

```
$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/" + $diskName  + ".vhd"

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm2
```

12. **Configure the virtual machines**

    When both the NetScaler VPX instances start, then connect to both NetScaler VPX instances using the SSH protocol to configure the virtual machines.

    a) Active-Active: Run the same set of configuration commands on the command line of both the NetScaler VPX instances.

    b) Active-Passive: Run this command on the command line of both the NetScaler VPX instances.

    ```
    add ha node #nodeID <nsip of other NetScaler VPX>
    ```

    In Active-Passive mode, run configuration commands on the primary node only.

Azure FAQs

- **Is the upgrade procedure of NetScaler VPX instance installed from Azure Marketplace different from the on-premises upgrade procedure?**

  No. You can upgrade your NetScaler VPX instance in the Microsoft Azure cloud to NetScaler VPX release 11.1 or later, using standard NetScaler VPX upgrade procedures. You can upgrade either using GUI or CLI procedures. For any new installations, use the NetScaler VPX image for Microsoft Azure cloud.

  To download the NetScaler VPX upgrade builds, go to **NetScaler downloads** > **NetScaler Firmware**.

- **How to correct MAC moves and interface mutes observed on NetScaler VPX instances hosted on Azure?**

  In Azure Multi-NIC environment, by default, all data interfaces might show MAC moves and interface mutes. To avoid MAC moves and interface mutes on Azure environments, Citrix recommends you to create a VLAN per data interface (without tag) of the NetScaler VPX instance and bind the primary IP of the NIC in Azure.

For more information, see CTX224626 article.

Deploy a NetScaler VPX instance on the Google Cloud Platform

You can deploy a NetScaler VPX instance on the Google Cloud Platform (GCP). A VPX instance in GCP enables you to take advantage of GCP cloud computing capabilities and use Citrix load balancing and traffic management features for your business needs. You can deploy VPX instances in GCP as stand-alone instances. Both single NIC and multi NIC configurations are supported.

Supported features

All Premium, Advanced, and Standard features are supported on the GCP based on the license/version type used.

Limitation

- IPv6 isn't supported.

Hardware requirements

VPX instance in GCP must have minimum of 2 vCPUs and 4 GB RAM.

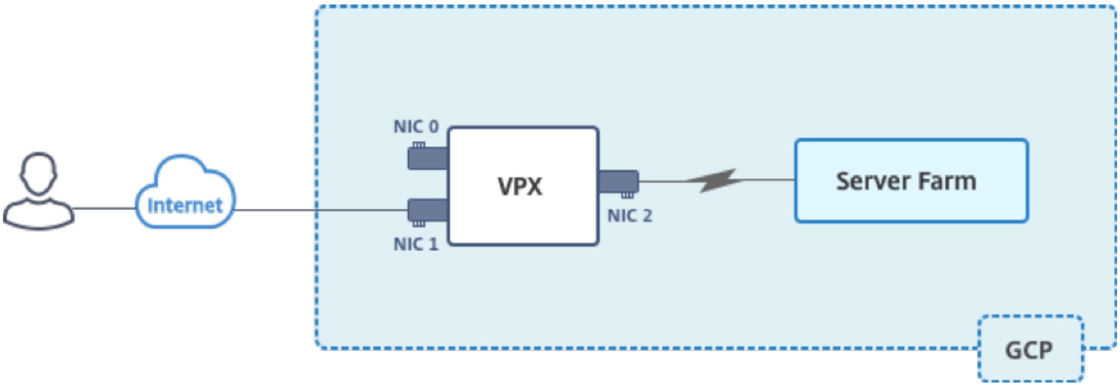
Points to note

Consider the following GCP-specific points before you begin your deployment.

- After creating the instance, you can't add or remove any network interfaces.
- For a multi-NIC deployment, create separate VPC networks for each NIC. One NIC can be associated with only one network.
- For a single-NIC instance, the GCP console creates a network by default.
- Minimum 4 vCPUs are required for an instance with more than two network interfaces.
- If IP forwarding is required, you must enable IP forwarding while creating the instance and configuring the NIC.

Scenario: Deploy a multi-NIC, multi-IP standalone VPX instance

This scenario illustrates how to deploy a NetScaler VPX standalone instance in GCP. In this scenario, you create a standalone VPX instance with many NICs. The instance communicates with back-end servers (the server farm).



Create three NICs to serve the following purposes.

NIC	Purpose	Associated with VPC network
NIC 0	Serves management traffic (NetScaler IP)	Management network
NIC 1	Serves client-side traffic (VIP)	Client network
NIC 2	Communicates with back-end servers (SNIP)	Back-end server network

Set up the required communication routes between the following:

- VPX instance and the back-end servers.
- VPX instance and the external hosts on the public internet.

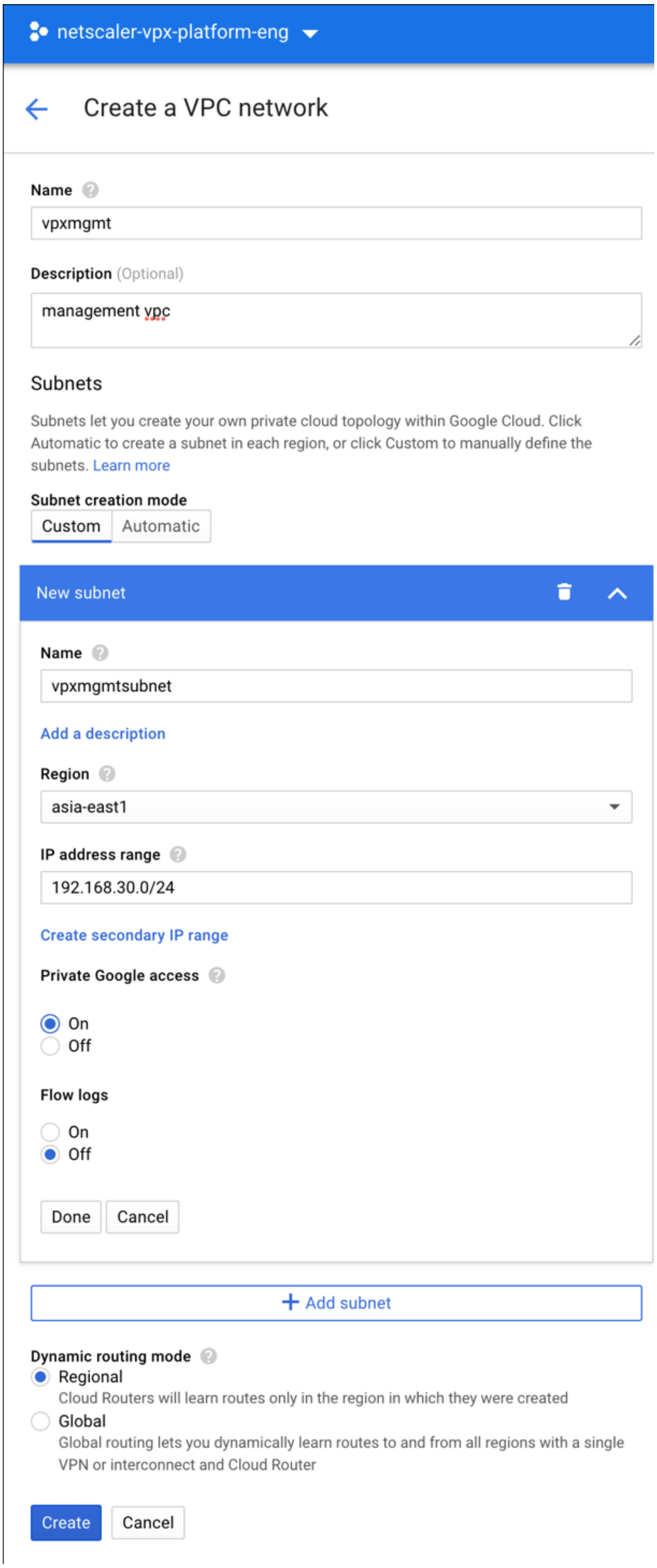
Summary of deployment steps

1. Create three VPC networks for three different NICs.
2. Create firewall rules for ports 22, 80, and 443
3. Create an instance with three NICs

> **Note:**
>
> Create an instance in the same region where you've created the VPC networks.

Step 1. Create VPC networks.

Create three VPC networks that is associated with management NIC, client NIC, and server NIC. To create a VPC network, log on to **Google console > Networking > VPC network > Create VPC Network**. Complete the required fields, as shown in the screen capture, and click **Create**.

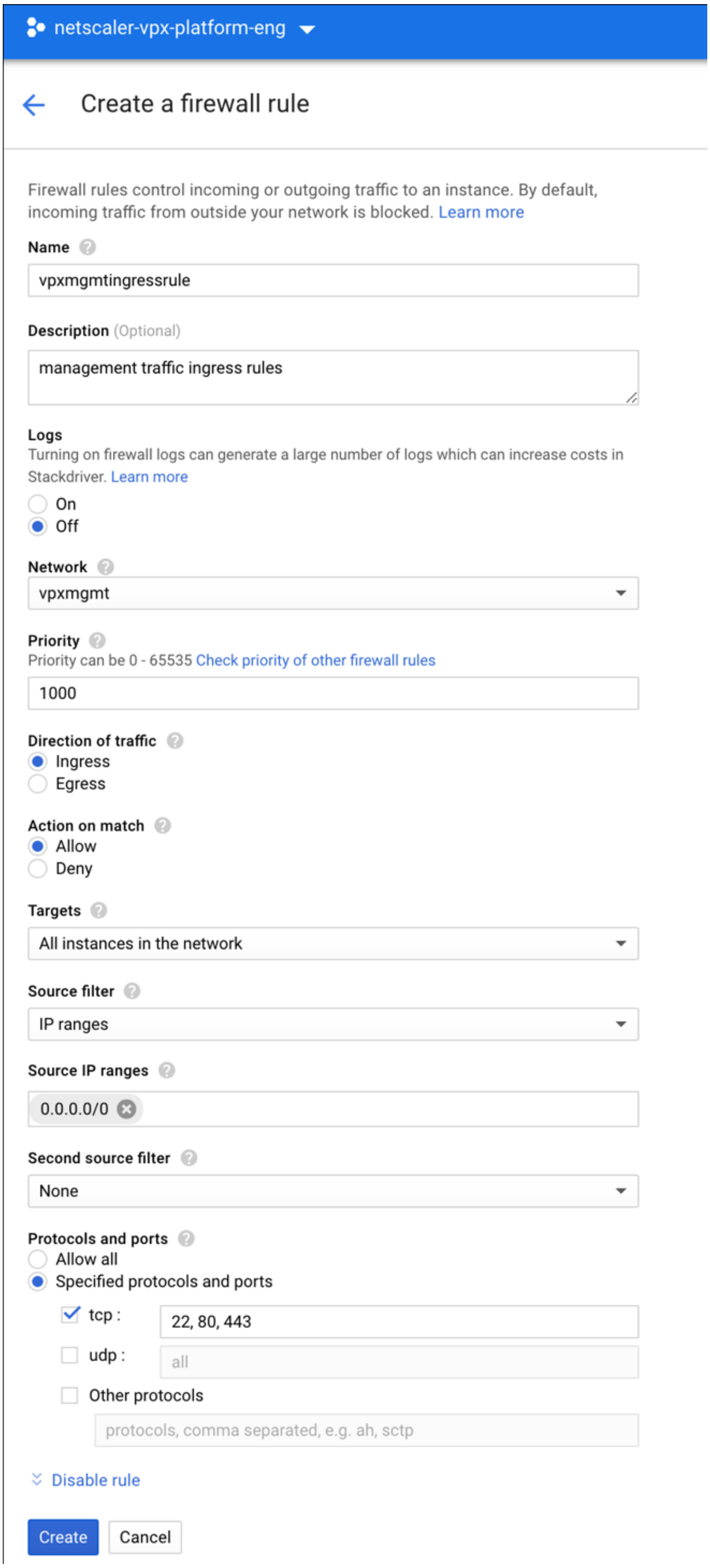

netscaler-vpx-platform-eng
Create a VPC network
Name
vpxmgmt
Description (Optional)
management vpc
Subnets
Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. Learn more
Subnet creation mode
Custom
Automatic
New subnet
Name
vpxmgmtsubnet
Add a description
Region
asia-east1
IP address range
192.168.30.0/24
Create secondary IP range
Private Google access
On
Off
Flow logs
On
Off
Done
Cancel
Add subnet
Dynamic routing mode
Regional
Cloud Routers will learn routes only in the region in which they were created
Global
Global routing lets you dynamically learn routes to and from all regions with a single VPN or interconnect and Cloud Router
Create
Cancel

Similarly, create VPC networks for client and server-side NICs.

> **Note:**
>
> All three VPC networks must be in the same region, which is asia-east1 in this scenario.

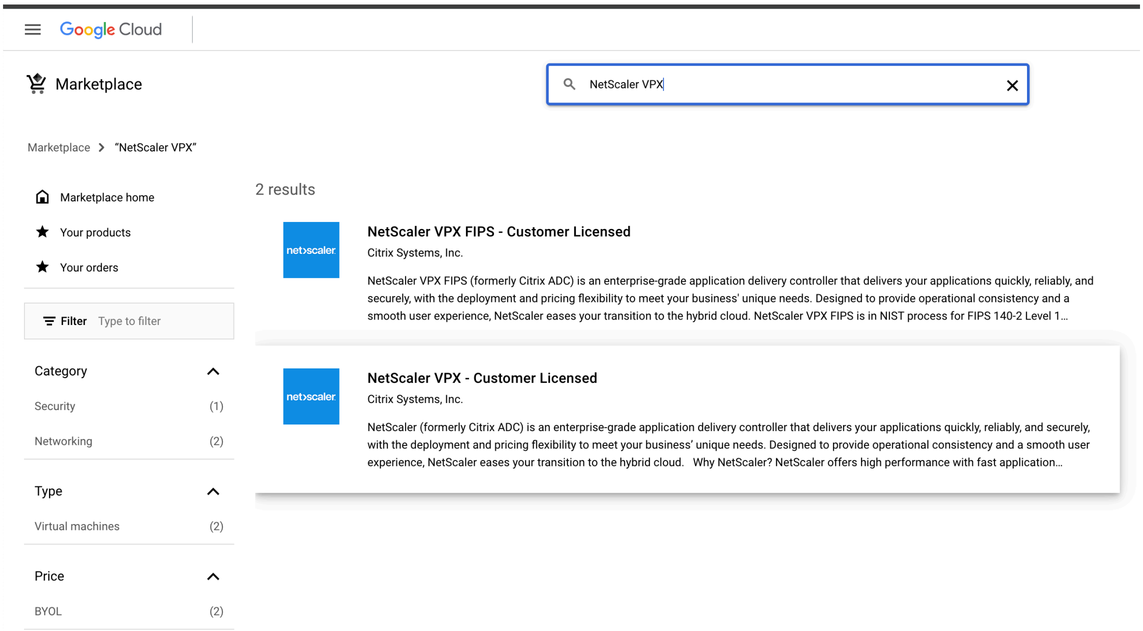
Step 2. Create firewall rules for ports 22, 80, and 443.

Create rules for SSH (port 22), HTTP (port 80), and HTTPS (port 443) for each VPC networks. For more information about firewall rules, see Firewall Rules Overview.

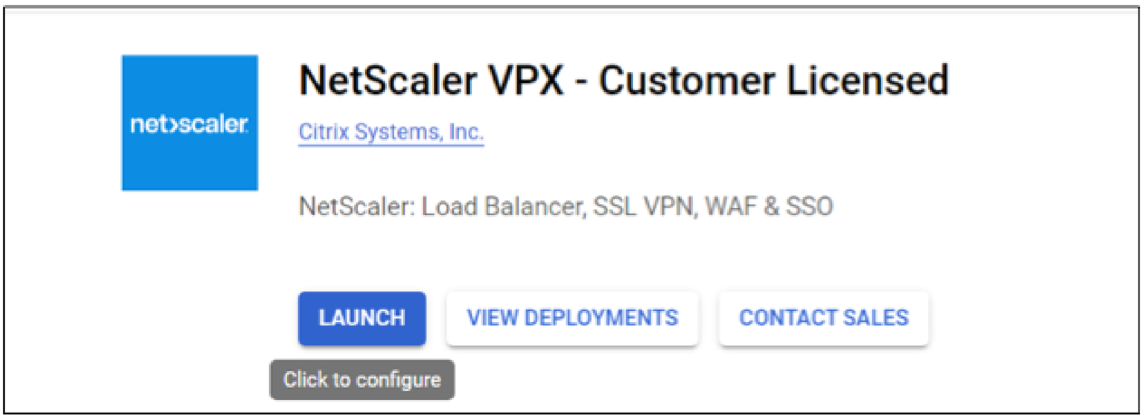

netscaler-vpx-platform-eng
Create a firewall rule
Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more
Name
vpxmgmtingressrule
Description (Optional)
management traffic ingress rules
Logs
Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. Learn more
On
Off
Network
vpxmgmt
Priority
Priority can be 0 - 65535 Check priority of other firewall rules
1000
Direction of traffic
Ingress
Egress
Action on match
Allow
Deny
Targets
All instances in the network
Source filter
IP ranges
Source IP ranges
0.0.0.0/0
Second source filter
None
Protocols and ports
Allow all
Specified protocols and ports
tcp :
22, 80, 443
udp :
all
Other protocols
protocols, comma separated, e.g. ah, sctp
Disable rule
Create
Cancel

Step 3. Create the VPX instance.

1. Log on to the GCP console.
2. Navigate to the GCP Marketplace.
3. Select a subscription based on your requirements.



4. Click **Launch** on the selected subscription.



5. Complete the deployment form and click **Deploy**.

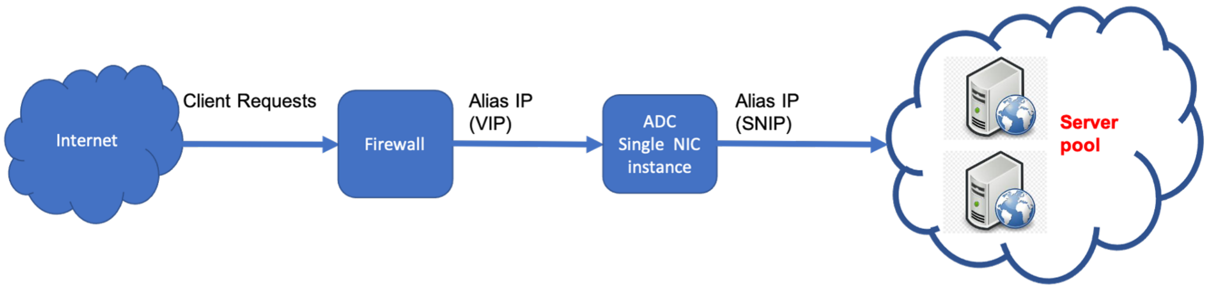
   > **Note:**
   >
   > Use the VPC Networks created in **Step 1**.

6. The deployed instance appears under **Compute Engine > VM instances**.

   Use the GCP SSH or the serial console to configure and manage the VPX instance.

Scenario: Deploy a single-NIC, standalone VPX instance

This scenario illustrates how to deploy a NetScaler VPX standalone instance with a single NIC in GCP. The alias IP addresses are used to achieve this deployment.



Create a single NIC (NIC0) to serve the following purposes:

- Handle management traffic (NetScaler IP) in the management network.
- Handle client-side traffic (VIP) in the client network.
- Communicate with back-end servers (SNIP) in the back-end server network.

Set up the required communication routes between the following:

- Instance and the back-end servers.
- Instance and the external hosts on the public internet.

Summary of deployment steps

1. Create a VPC network for NIC0.
2. Create firewall rules for ports 22, 80, and 443.
3. Create an instance with a single NIC.
4. Add Alias IP addresses to VPX.
5. Add VIP and SNIP on VPX.
6. Add a load balancing virtual server.
7. Add a service or service group on the instance.
8. Bind the service or service group to the load balancing virtual server on the instance.

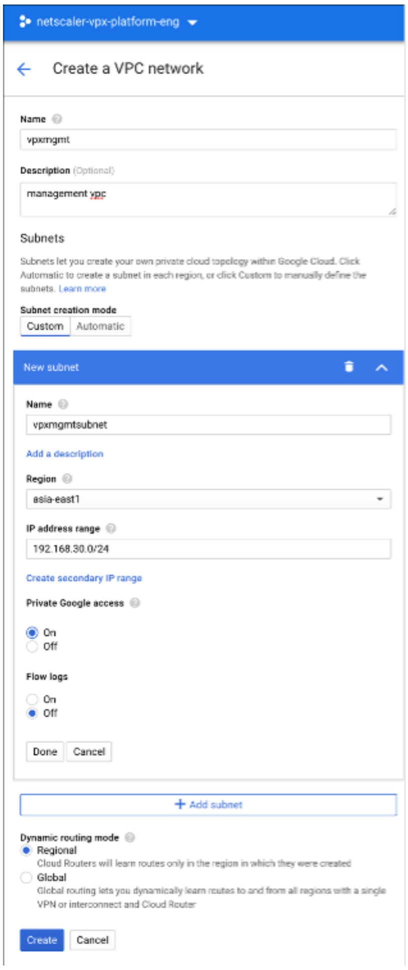
> **Note:**
>
> Create an instance in the same region where you've created the VPC networks.

Step 1. Create one VPC network.

Create one VPC network to associate with NIC0.

To create a VPC network, do these steps:

1. Log on to **GCP console > Networking > VPC network > Create VPC Network**
2. Complete the required fields, and click **Create**.



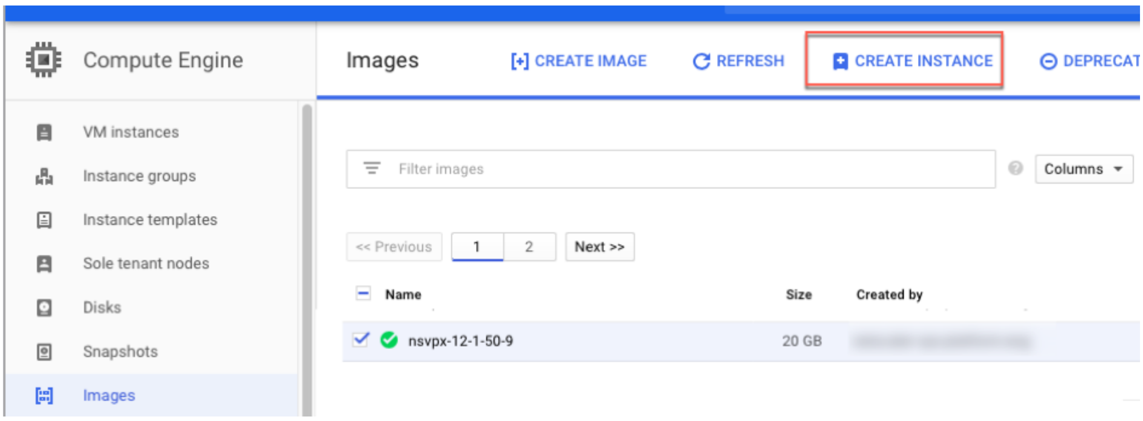
Step 2. Create firewall rules for ports 22, 80, and 443.

Create rules for SSH (port 22), HTTP (port 80), and HTTPS (port 443) for the VPC network. For more information about firewall rules, see Firewall Rules Overview.

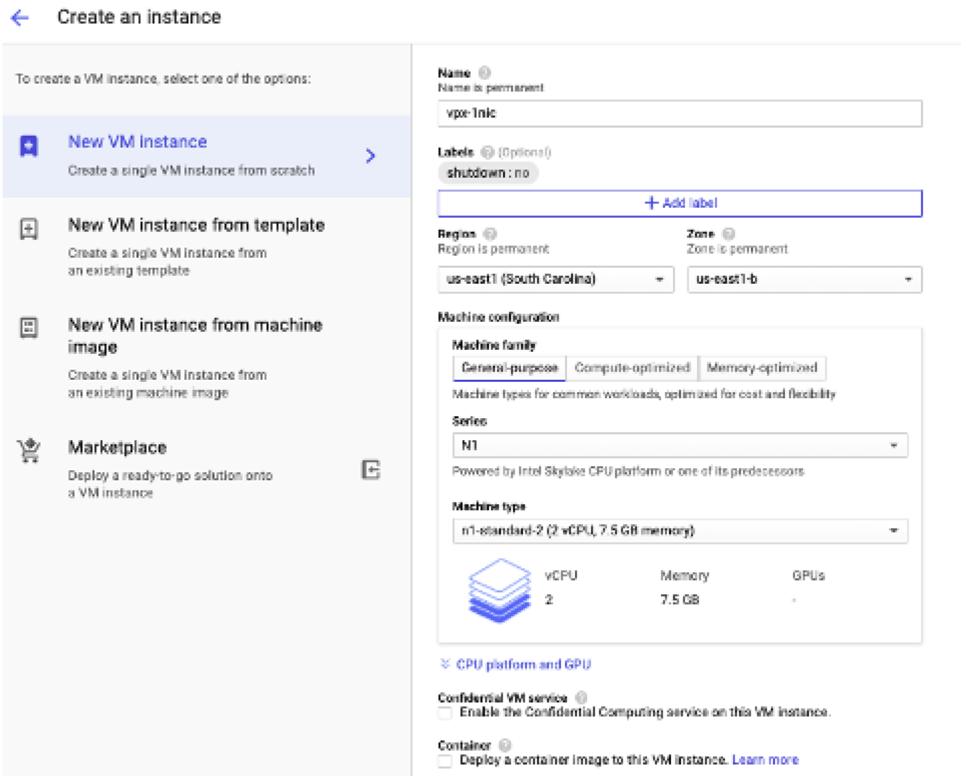
Step 3. Create an instance with single NIC.

To create an instance with single NIC, do these steps:

1. Log on to the **GCP console**.
2. Under **Compute**, hover over **Compute Engine**, and select **Images**.
3. Select the image, and click **Create Instance**.



4. Select an instance type with two vCPUs (minimum requirement for ADC).



5. Click the **Networking** tab from the **Management, security, disks, networking** window.
6. Under **Network interfaces**, click the **Edit** icon to edit the default NIC.
7. In the **Network interfaces** window, under **Network**, select the VPC network that you created.
8. You can create a static external IP address. Under the **External IP addresses**, click **Create IP address**.
9. In the **Reserve a static address** window, add a name and description and click **Reserve**.
10. Click **Create** to create the VPX instance.
    The new instance appears under VM instances.

Step 4. Add alias IP addresses to the VPX instance.

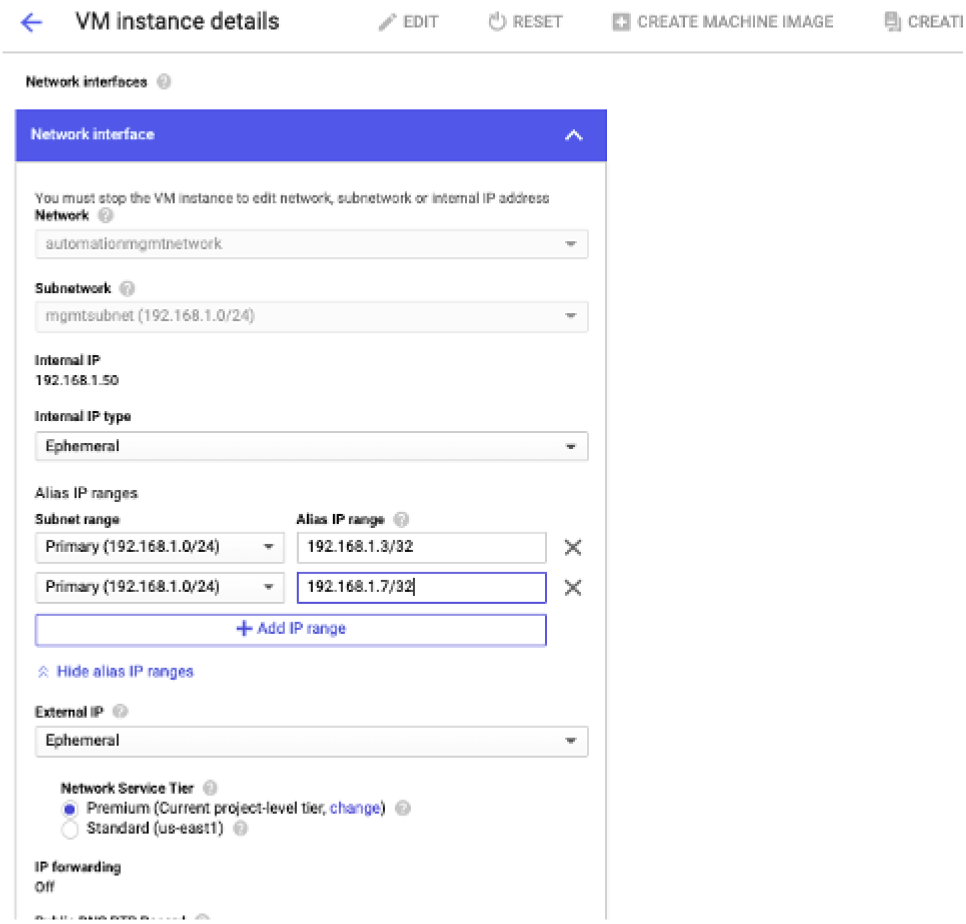
Assign two alias IP addresses to the VPX instance to use as VIP and SNIP addresses.

> **Note:**
>
> Do not use the primary internal IP address of the VPX instance to configure the VIP or SNIP.

To create an alias IP address, perform these steps:

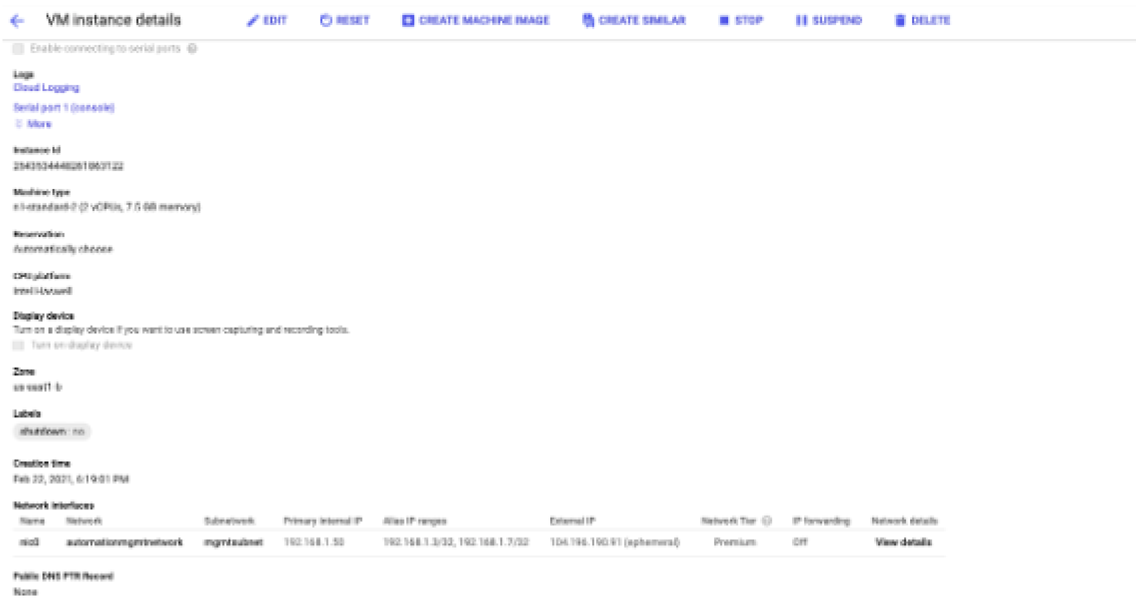
1. Navigate to the VM instance and click **Edit**.
2. In the **Network interface** window, edit the NIC0 interface.

3. In the **Alias IP range** field, enter the alias IP addresses.



4. Click **Done**, and then **Save**.

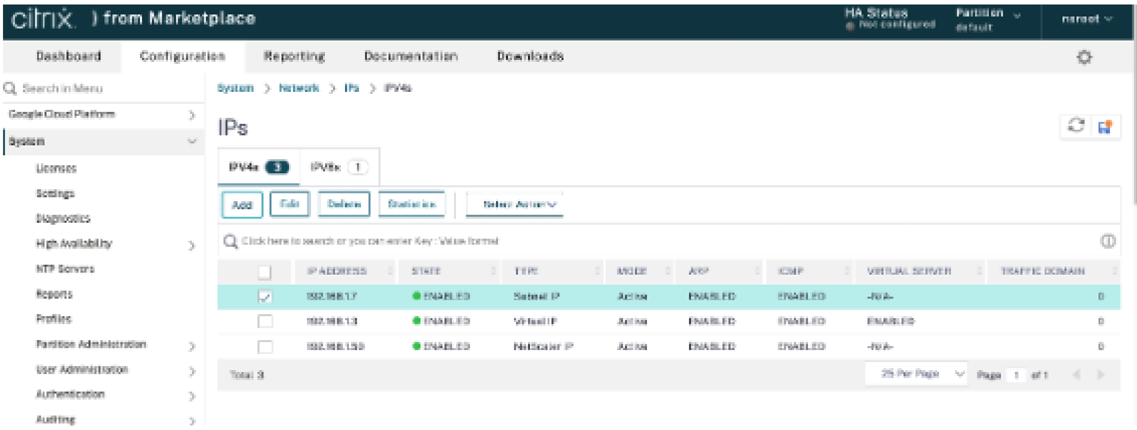
5. Verify the alias IP addresses in the **VM instance details** page.



Step 5. Add VIP and SNIP on the VPX instance.

On the VPX instance, add client alias IP address and server alias IP address.

1. On the NetScaler GUI, navigate to **System > Network > IPs > IPv4s**, and click **Add**.



2. To create a client alias IP (VIP) address:

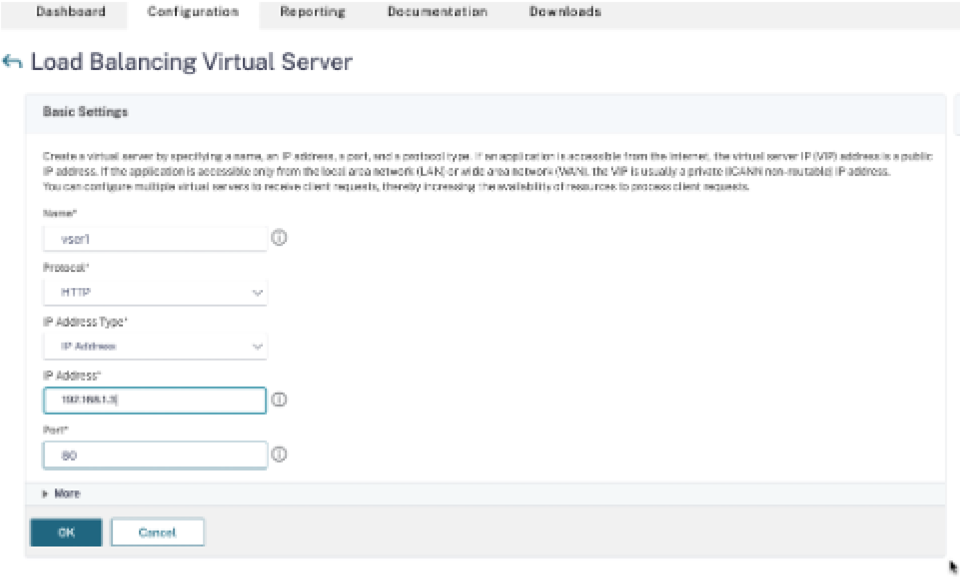
    - Enter the client-alias IP address and netmask configured for the VPC subnet in the VM instance.
    - In the **IP Type** field, select **Virtual IP** from the drop-down menu.
    - Click **Create**.

3. To create a server alias IP (SNIP) address:

    - Enter the server-alias IP address and netmask configured for the VPC subnet in the VM instance.
    - In the **IP Type** field, select **Subnet IP** from the drop-down menu.
    - Click **Create**.

Step 6. Add load balancing virtual server.

1. On the NetScaler GUI, navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**, and click **Add**.
2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP Address (client alias IP), and Port.
3. Click **OK** to create the load balancing virtual server.



Step 7. Add a service or service group on the VPX instance.

1. From the NetScaler GUI, navigate to **Configuration > Traffic Management > Load Balancing > Services**, and click **Add**.
2. Add the required values for Service Name, IP Address, Protocol, and Port, and click **OK**.

Step 8. Bind the service/service group to the Load Balancing Virtual Server on the instance.

1. From the GUI, navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 6**, and click **Edit**.
3. In the **Service and Service Groups** window, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in **Step 7**, and click **Bind**.

Points to note after you've deployed the VPX instance on GCP

- Log on to the VPX with user name `nsroot` and instance ID as password. At the prompt, change the password and save the configuration.
- For collecting a technical support bundle, run the command `shell /netscaler/showtech_cloud.pl` instead of the customary `show techsupport`.
- After deleting a NetScaler VM from GCP console, delete the associated NetScaler internal target instance also. To do so, go to gcloud CLI and type the following command:

```
gcloud compute -q  target-instances delete <instance-name>-
    adcinternal --zone <zone>
```


> **Note:**
>
> `<instance-name>-adcinternal` is the name of the target instance that must be deleted.

NetScaler VPX licensing

A NetScaler VPX instance on GCP requires a valid license. The licensing option available for NetScaler VPX instances running on GCP is as follows:

Bring your own license (BYOL): To use the BYOL option, follow these steps:

- Use the licensing portal on the NetScaler website to generate a valid license.
- Upload the generated license to the instance.
- **NetScaler VPX Check-in and Check-out license**: This licensing model allows you to check out a license from a pool of available licenses and check it back in when no longer needed. For more information and detailed instructions, see NetScaler VPX Check-in and Check-out License.

> **Note:**
>
> Subscription-based licensing is no longer supported for NetScaler VPX instances on GCP.

Supported NetScaler VPX offerings on GCP

The following table lists the supported NetScaler VPX offerings on GCP.

Supported VPX offerings
NetScaler VPX - Customer Licensed
NetScaler VPX FIPS - Customer Licensed

Supported GCP machine type families

Machine type family	Minimum machine type
General Purpose Machines	e2-medium, e2-standard-2, e2-highmem-2, n1-standard-2, n1-highmem-2, n2-standard-2, n2-highmem-2, n2d-standard-2, n2d-highmem-2
Compute-Optimized Machines	c2-standard-4, c2d-standard-2, c2d-highmem-2

GDM templates to deploy a NetScaler VPX instance

You can use a NetScaler VPX Google Deployment Manager (GDM) template to deploy a VPX instance on GCP. For details, see NetScaler GDM Templates.

NetScaler marketplace images

You can use the images in GDM templates to bring up the NetScaler appliance.

The following table lists the images that are available on GCP marketplace.

Release	Image name	Image location
13.1	citrix-adc-vpx-express-13-1-53-17	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-1-53-17
13.1	citrix-adc-vpx-200-enterprise-13-1-53-17	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-1-53-17
13.1	citrix-adc-vpx-1000-platinum-13-1-53-17	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-platinum-13-1-53-17
13.1	citrix-adc-vpx-5000-platinum-13-1-53-17	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-1-53-17
13.1	citrix-adc-vpx-byol-13-1-53-17	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-1-53-17

Resources

- Creating Instances with Multiple Network Interfaces
- Creating and Starting a VM Instance

Related information

- Deploy a VPX high-availability pair on Google Cloud Platform

Deploy a VPX high-availability pair on Google Cloud Platform

You can configure two NetScaler VPX instances on Google Cloud Platform (GCP) as a high availability (HA) active-passive pair. When you configure one instance as the primary node and the other as the secondary node, the primary node accepts connections and manages servers. The secondary node monitors the primary. If for any reason, if the primary node is unable to accept connections, the secondary node takes over.

For more information on HA, see High Availability.

The nodes must be in the same region; however, they can be either in the same zone or different zones. For more information, see Regions and Zones.

Each VPX instance requires at least three IP subnets (Google VPC networks):

- A management subnet
- A client-facing subnet (VIP)
- A back-end facing subnet (SNIP, MIP, and so on)

Citrix recommends three network interfaces for a standard VPX instance.

You can deploy a VPX high-availability pair in the following methods:

- Using external static IP address
- Using private IP address
- Using single nic VMs with private IP address

GDM templates to deploy a VPX high-availability pair on GCP

You can use a NetScaler Google Deployment Manager (GDM) template to deploy a VPX high-availability pair on GCP. For details, see NetScaler GDM Templates.

Forwarding rules support for VPX high-availability pair on GCP

You can deploy a VPX high-availability pair on the GCP using forwarding rules.

For more information on forwarding rules, see Forwarding rules overview.

Prerequisites

- Forwarding rules must be in the same region as the VPX instances.
- Target instances must be in the same zone as the VPX instance.
- Number of target instances for both primary and secondary nodes must match.

Example:

You have a high-availability pair in the `us-east1` region with primary VPX in `us-east1-b` zone and secondary VPX in `us-east1-c` zone. A forwarding rule is configured for the primary VPX with the target instance in `us-east1-b` zone. Configure a target instance for secondary VPX in `us-east1-c` zone to update the forwarding rule on failover.

Limitations

Only forwarding rules that are configured with target instances at the back end are supported in VPX high-availability deployment.

Deploy a VPX high-availability pair with external static IP address on the Google Cloud Platform

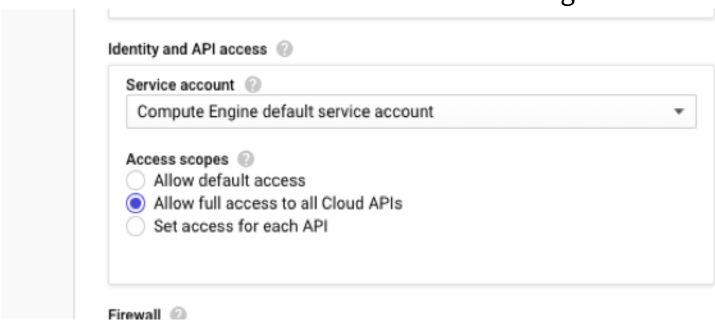
You can deploy a VPX high-availability pair on GCP using an external static IP address. The client IP address of the primary node must be bound to an external static IP address. Upon failover, the external static IP address is moved to the secondary node for traffic to resume.

A static external IP address is an external IP address that is reserved for your project until you decide to release it. If you use an IP address to access a service, you can reserve that IP address so that only your project can use it. For more information, see Reserving a Static External IP Address.

For more information on HA, see High Availability.

Before you start

- Read the Limitation, Hardware requirements, Points to note mentioned in Deploy a NetScaler VPX instance on Google Cloud Platform. This information applies to HA deployments also.
- Enable **Cloud Resource Manager API** for your GCP project.
- Allow full access to all Cloud APIs while creating the instances.



- Ensure that the IAM role associated with your GCP service account has the following IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [

"compute.addresses.use",
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata"
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.instances.get",
"Compute.instances.list",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
```

- If you have configured alias IP addresses on an interface other than the management interface, ensure that your GCP service account has the following additional IAM permissions:

```
"compute.instances.updateNetworkInterface"
```

- If you have configured GCP forwarding rules on the primary node, read the limitations and requirements mentioned in Forwarding rules support for VPX high-availability pair on GCP to update them to new primary on failover.

How to deploy a VPX HA pair on Google Cloud Platform

Here's a summary of the HA deployment steps:

1. Create VPC networks in the same region. For example, Asia-east.
2. Create two VPX instances (primary and secondary nodes) on the same region. They can be in the same zone or different zones. For example Asia east-1a and Asia east-Ib.
3. Configure HA settings on both instances by using the NetScaler GUI or ADC CLI commands.

Step 1. Create VPC networks

Create VPC networks based on your requirements. Citrix recommends you to create three VPC networks for associating with management NIC, client NIC, and server NIC.

To create a VPC network, perform these steps:

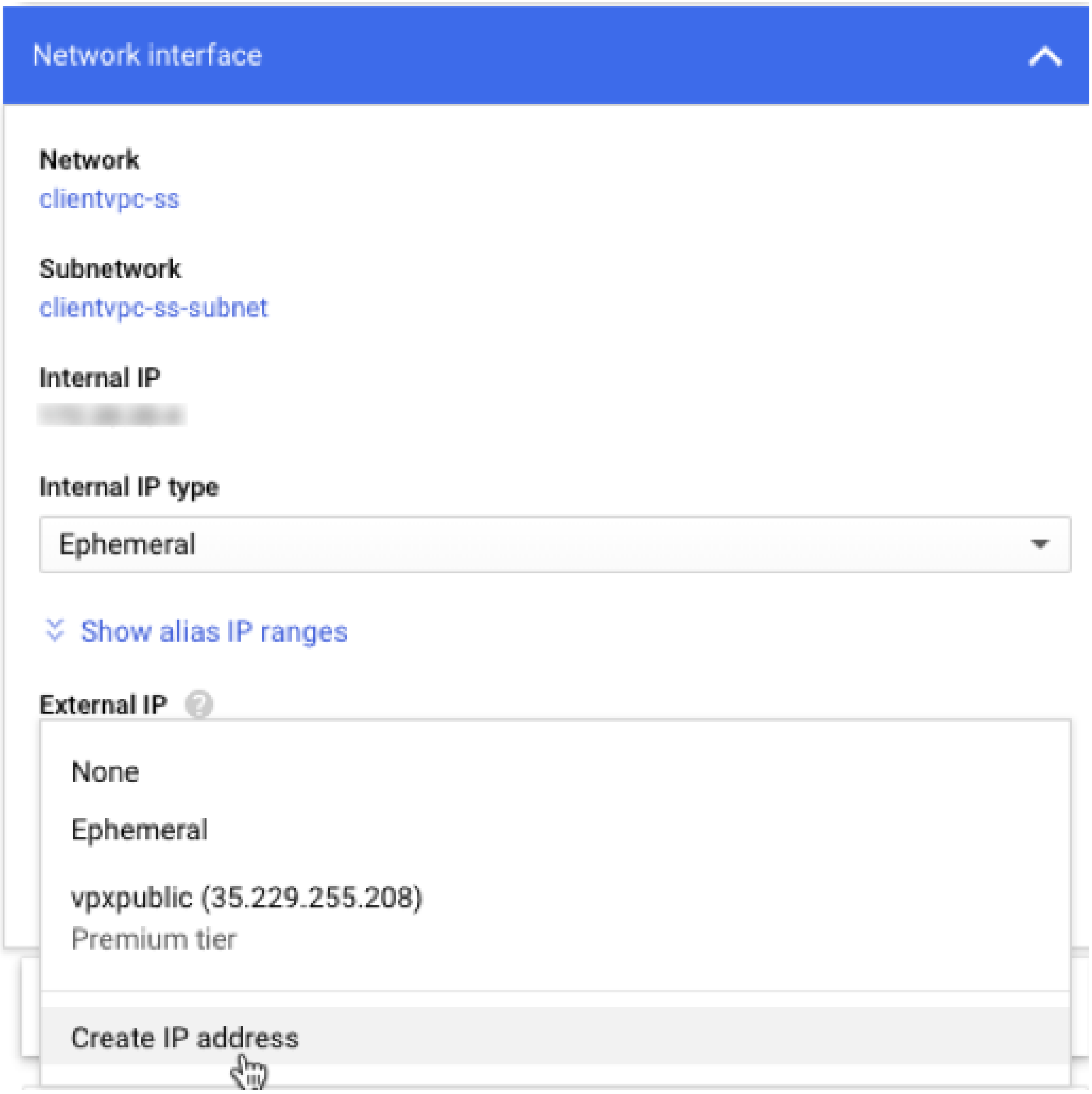
1. Log on the **Google console > Networking > VPC network > Create VPC Network**.
2. Complete the required fields, and click **Create**.

For more information, see the **Create VPC Networks** section in Deploy a NetScaler VPX instance on Google Cloud Platform.

Step 2. Create two VPX instances

Create two VPX instances by following the steps given in Scenario: deploy a multi-NIC, multi-IP standalone VPX instance.

> **Important:**
>
> Assign a static external IP address to client IP address (VIP) of the primary node. You can use an existing reserved IP address or create a new one. To create a static external IP address, navigate to **Network interface > External IP**, click **Create IP address**.



After the failover, when the old primary becomes the new secondary, the static external IP address moves from the old primary and is attached to the new primary. For more information, see the Google cloud document Reserving a Static External IP Address.

After you've configured the VPX instances, you can configure the VIP and SNIP addresses. For more information, see Configuring NetScaler-owned IP addresses.

Step 3. Configure high availability

After you've created the instances on Google Cloud Platform, you can configure HA by using the NetScaler GUI for CLI.

Configure HA by using the GUI **Step 1**. Set up high availability in INC mode on both the instances.

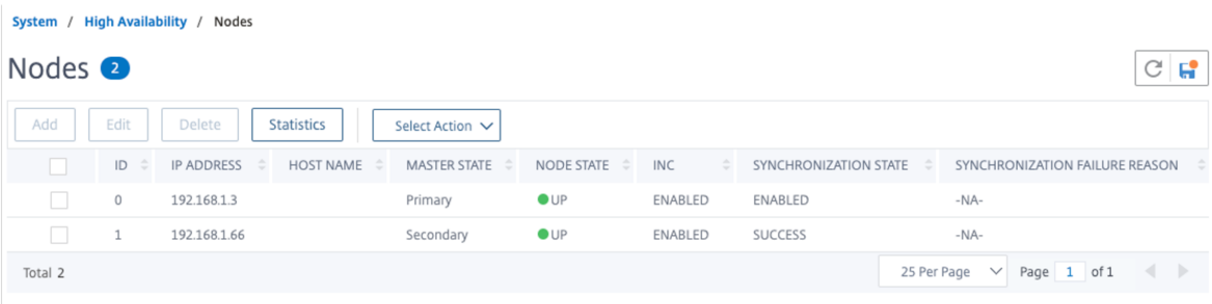
On the **primary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the secondary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

On the **secondary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the primary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

Before you proceed further, ensure that the Synchronization state of the secondary node is shown as **SUCCESS** in the **Nodes** page.



> **Note:**
>
> Now, the secondary node has the same log-on credentials as the primary node.

Step 2. Add Virtual IP address and Subnet IP address on both the nodes.

On the **primary node**, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. Add a primary VIP address by following these steps:

   a) Enter the internal IP address of the client-facing interface of the primary instance and net-mask configured for the client subnet in the VM instance.

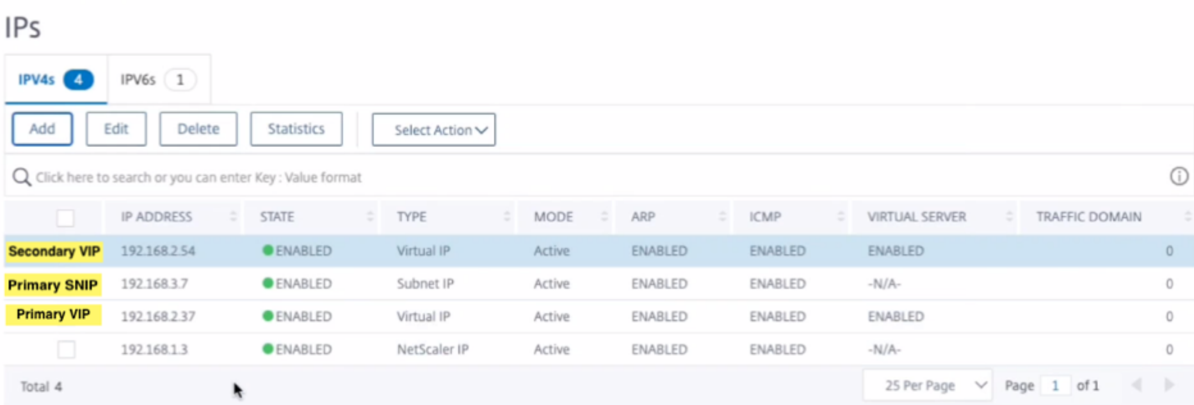
   b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
   c) Click **Create**.

3. Add a primary SNIP address by following these steps:

   a) Enter the internal IP address of the server-facing interface of the primary instance and netmask configured for the server subnet in the primary instance.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.

4. Add a secondary VIP address by following these steps:

   a) Enter the internal IP address of the client-facing interface of the secondary instance and netmask configured for the client subnet in the VM instance.
   b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
   c) Click **Create**.



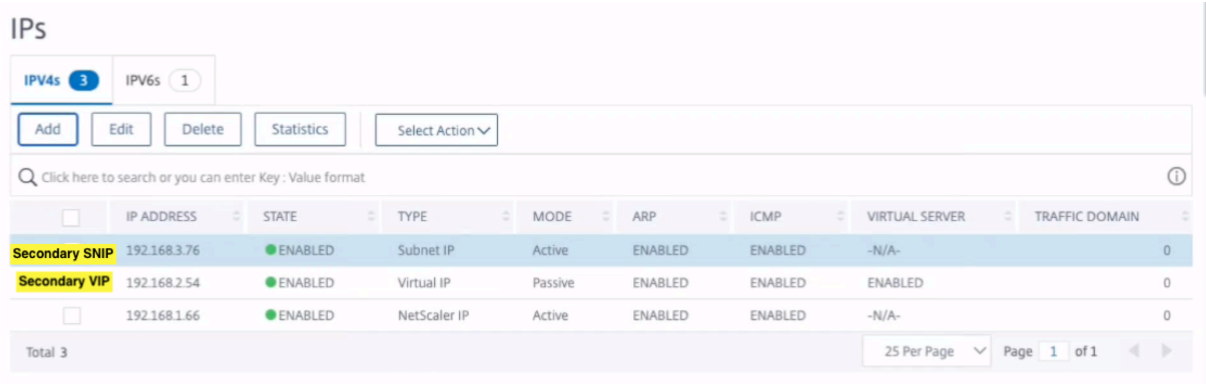
On the **secondary node**, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. Add a secondary VIP address by following these steps:

   a) Enter the internal IP address of the client-facing interface of the secondary instance and netmask configured for the client subnet in the VM instance.
   b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.

3. Add a secondary SNIP address by following these steps:

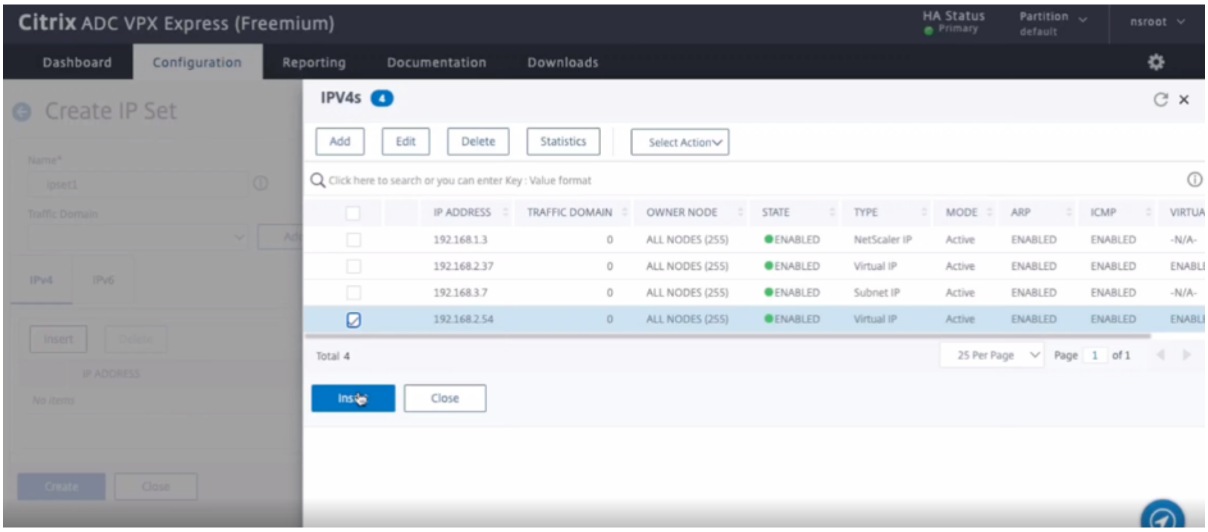
   a) Enter the internal IP address of the server-facing interface of the secondary instance and netmask configured for the server subnet in the secondary instance.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.



Step 3. Add IP set and bind IP set to the secondary VIP on both the instances.

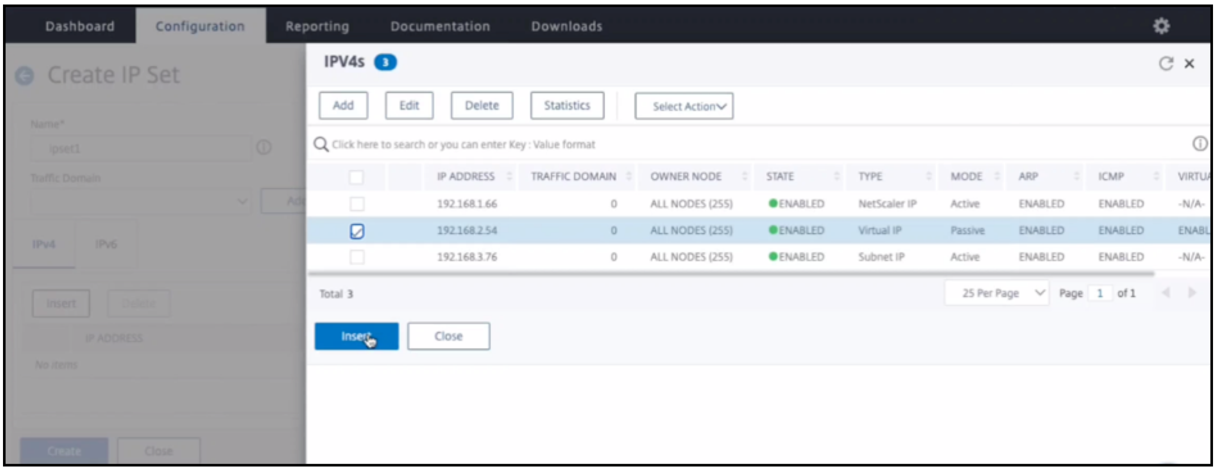
On the **primary node**, perform the following steps:

1. Navigate to **System > Network > IP Sets > Add**.
2. Add an IP set name and click **Insert**.
3. From the **IPV4s** page, select the virtual IP (secondary VIP) and click **Insert**.
4. Click **Create** to create the IP set.



On the **secondary node**, perform the following steps:

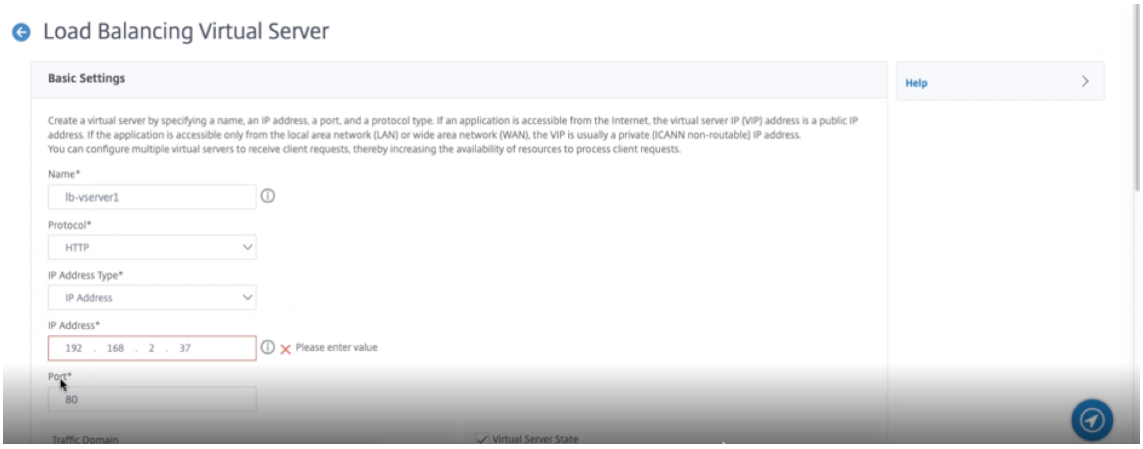
1. Navigate to **System > Network > IP Sets > Add**.
2. Add an IP set name and click **Insert**.
3. From the **IPV4s** page, select the virtual IP (secondary VIP) and click **Insert**.
4. Click **Create** to create the IP set.



> **Note:**
>
> IP set name must be same on both the instances.

Step 4. Add a load balancing virtual server on the primary instance.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.
2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP address (primary VIP), and Port.



3. Click **More**. Navigate to **IP Range IP Set Settings**, select **IPset** from the drop-down menu, and provide the IPset created in **Step 3**.
4. Click **OK** to create the load balancing virtual server.

Step 5. Add a service or service group on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

Step 6. Bind the service or service group to the load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 4**, and click **Edit**.
3. In the **Service and Service Groups** tab, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in the **Step 5**, and click **Bind**.

Save the configuration. After a forced failover, the secondary becomes the new primary. The external static IP of the old primary VIP moves to the new secondary VIP.

Configure high availability using CLI **Step 1**. Set up high availability in INC mode in both the instances.

On the primary node, type the following command.

```
add ha node 1 <sec_ip> -inc ENABLED
```

On the secondary node, type the following command.

```
add ha node 1 <prim_ip> -inc ENABLED
```

`sec_ip` refers to the internal IP address of the management NIC of the secondary node.

`prim_ip` refers to the internal IP address of the management NIC of the primary node.

Step 2. Add Virtual and Subnet IPs on both the nodes.

On the primary node, type the following command.

```
add ns ip <primary_vip> <subnet> -type VIP

add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <primary_snip> <subnet> -type SNIP
```

`primary_vip` refers to the internal IP address of the client-facing interface of the primary instance.

`secondary_vip` refers to the internal IP address of the client-facing interface of the secondary instance.

`primary_snip` refers to the internal IP address of the server-facing interface of the primary instance.

On the secondary node, type the following command.

```
add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <secondary_snip> <subnet> -type SNIP
```


`secondary_vip` refers to the internal IP address of the client-facing interface of the secondary instance.

`secondary_snip` refers to the internal IP address of the server-facing interface of the secondary instance.

Step 3. Add IP set and bind IP set to secondary VIP on both the instances.

On the primary node, type the following command:

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
```

On the secondary node, type the following command:

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
```

> **Note:**
>
> IP set name must be same on both the instances.

Step 4. Add a virtual server on the primary instance.

Type the following command:

```
add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port
    > -ipset <ipset_name>
```

Step 5. Add a service or service group on the primary instance.

Type the following command:

```
add service <service_name> <service_ip_address> <protocol> <port>
```

Step 6. Bind the service/service group to the load balancing virtual server on the primary instance.

Type the following command:

```
bind <server_type> vserver <vserver_name> <service_name>
```

> **Note:**
>
> To save your configuration, type the command `save config`. Otherwise, the configurations are lost after you restart the instances.

Step 7. Verify the configuration.

Ensure that the external IP address attached to the primary client NIC moves to the secondary on a failover.

1. Make a cURL request to the external IP address and make sure that it is reachable.

2. On the primary instance, perform failover:

   From GUI, navigate to **Configuration > System > High Availability > Action > Force Failover**.

   From CLI, type the following command:

   ```
   force ha failover -f
   ```

   On the GCP console, goto the Secondary instance. The external IP address must have moved to the client NIC of secondary after failover.

3. Issue a cURL request to the external IP and ensure it is reachable again.

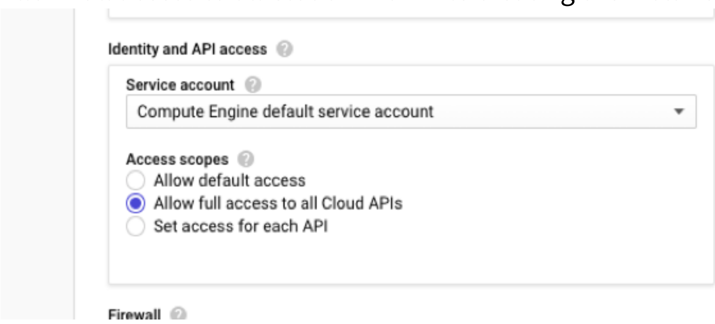
Deploy a single NIC VPX high-availability pair with private IP address on Google Cloud Platform

You can deploy a single NIC VPX high-availability pair on GCP using private IP address. The client IP (VIP) address must be configured as alias IP address on the primary node. Upon failover, the Client IP address is moved to the secondary node, for the traffic to resume. The Subnet IP (SNIPs) addresses for each node must also be configured as an alias IP range.

For more information on high availability, see High Availability.

Before you start

- Read the Limitation, Hardware requirements, Points to note mentioned in Deploy a NetScaler VPX instance on Google Cloud Platform. This information applies to high availability deployments also.
- Enable **Cloud Resource Manager API** for your GCP project.
- Allow full access to all Cloud APIs while creating the instances.



- Ensure that your GCP service account has the following IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.forwardingRules.list",
```



```
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata",
"compute.instances.get",
"compute.instances.list",
"compute.instances.updateNetworkInterface",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
```

- If your VMs do not have internet access, you must enable **Private Google Access** on the VPC subnet.

- If you have configured GCP forwarding rules on the primary node, read the limitations and requirements mentioned in Forwarding rules support for VPX high-availability pair on GCP to update them to new primary on failover.

How to deploy a VPX high availability pair on Google Cloud Platform

Here is a summary of the steps for deploying HA pair with single NIC:

1. Create one VPC network.
2. Create two VPX instances (primary and secondary nodes) in the same region. They can be in the same zone or different zones. For example Asia east-1a and Asia east-Ib.
3. Configure HA settings on both instances by using the NetScaler GUI or ADC CLI commands.

Step 1. Create one VPC network

To create a VPC network, perform these steps:

1. Log on to the **Google console > Networking > VPC network > Create VPC Network**.
2. Complete the required fields, and click **Create**.

For more information, see the **Create VPC Networks** section in Deploy a NetScaler VPX instance on Google Cloud Platform.

Step 2. Create two VPX instances

Create two VPX instances by following the step 1 to step 3 given in Scenario: Deploy a single-NIC, stand-alone VPX instance.

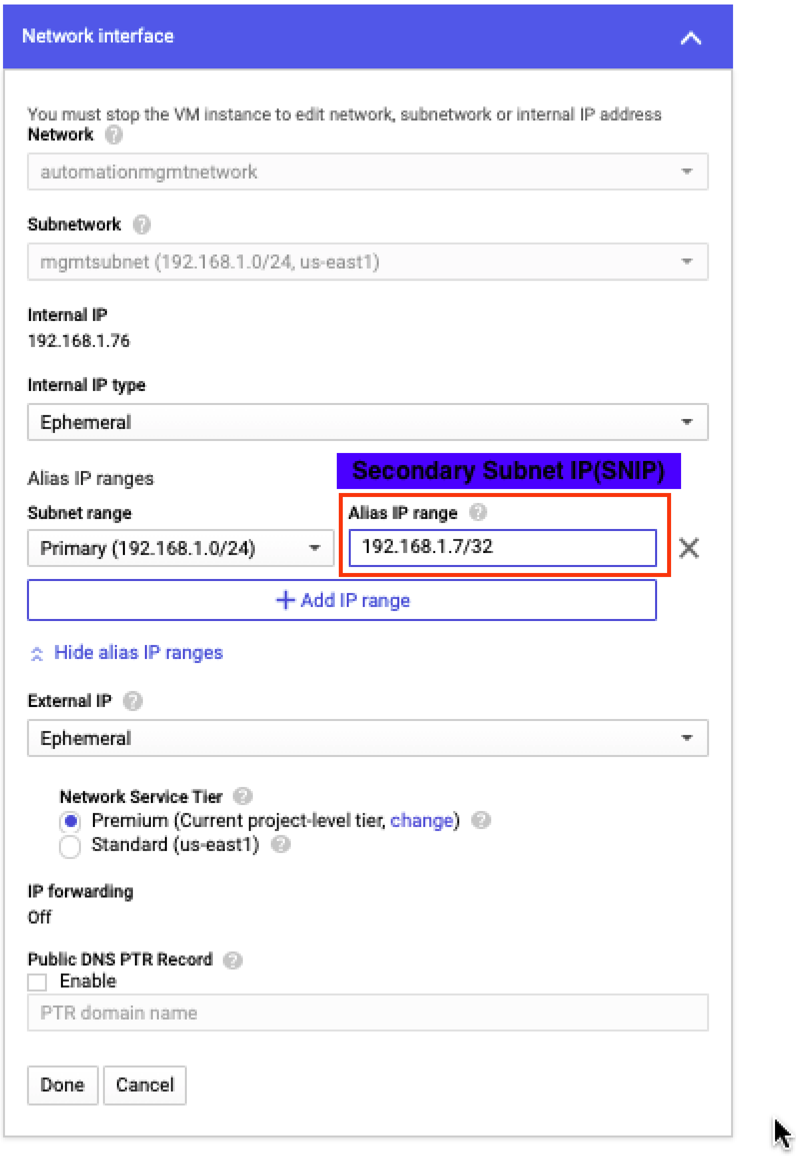
> **Important:**
>
> Assign a client alias IP address only to the primary node and server alias IP addresses to primary and secondary nodes. Do not use the internal IP address of the VPX instance to configure the VIP or SNIP.

To create client and server alias IP addresses, perform these steps on the primary node:

1. Navigate to the VM instance and click **Edit**.
2. In the **Network Interface** window, edit the client (NIC0) interface.
3. In the **Alias IP range** field, enter the client alias IP address.
4. Click **Add IP Range** and enter the server alias IP address.

To create a server alias IP address, perform these steps on the secondary node:

1. Navigate to the VM instance and click **Edit**.
2. In the **Network Interface** window, edit the client (NIC0) interface.
3. In the **Alias IP range** field, enter the server alias IP address.



After the failover, when the old primary becomes the new secondary, the client alias IP address is moved from the old primary and is attached to the new primary.

After you have configured the VPX instances, you can configure the Virtual (VIP) and Subnet IP (SNIP) addresses. For more information, see Configuring NetScaler-owned IP addresses.

Step 3. Configure high availability

After you've created the instances on Google Cloud Platform, you can configure high availability by using the NetScaler GUI or CLI.

Configure high availability by using the GUI

Step 1. Set up high availability in INC Enabled mode on both the nodes.

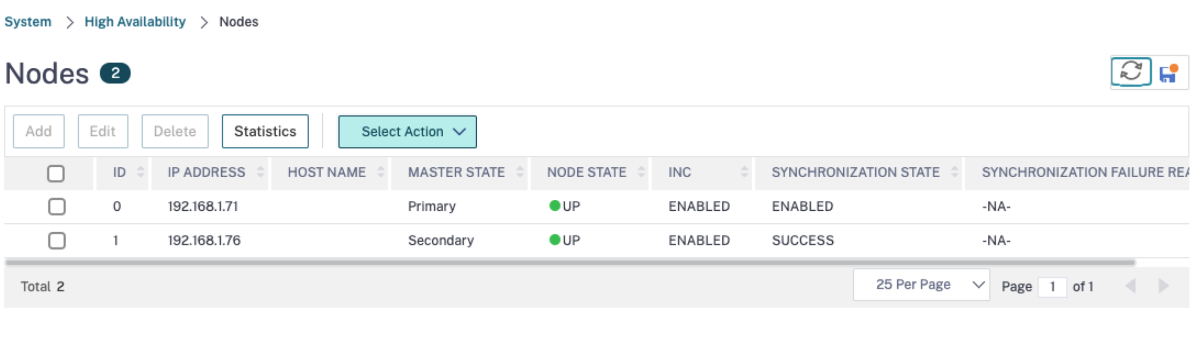
On the **primary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the secondary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

On the **secondary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the primary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

Before you proceed further, ensure that the Synchronization state of the secondary node is shown as **SUCCESS** in the **Nodes** page.



> **Note:**
>
> After the secondary node is synchronized with the primary node, the secondary node has the same log-on credentials as the primary node.

Step 2. Add Virtual IP address and Subnet IP address on both the nodes.

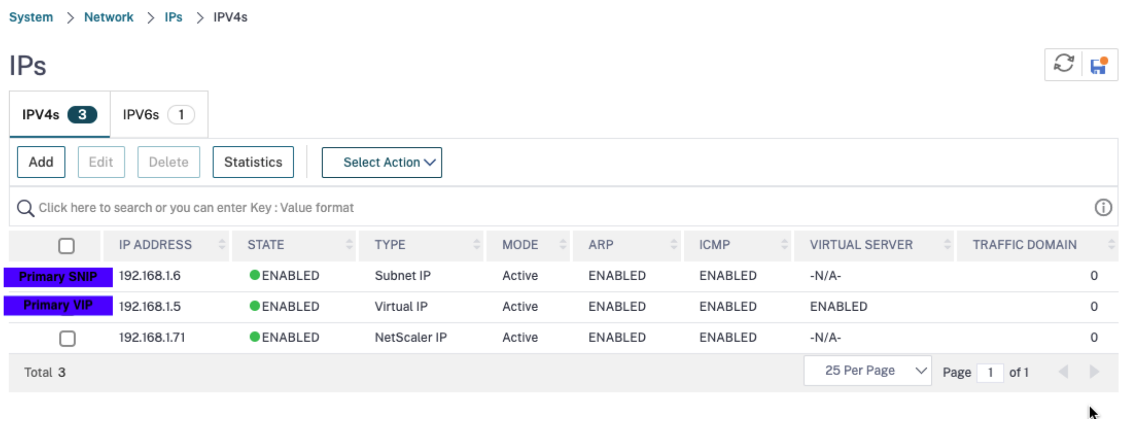
On the primary node, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. To create a client alias IP (VIP) address:

    a) Enter the client alias IP address and netmask configured for the VPC subnet in the primary VM instance.
    b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
    c) Click **Create**.

3. To create a server alias IP (SNIP) address:

    a) Enter the server alias IP address and netmask configured for the VPC subnet in the primary VM instance.
    b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
    c) Click **Create**.



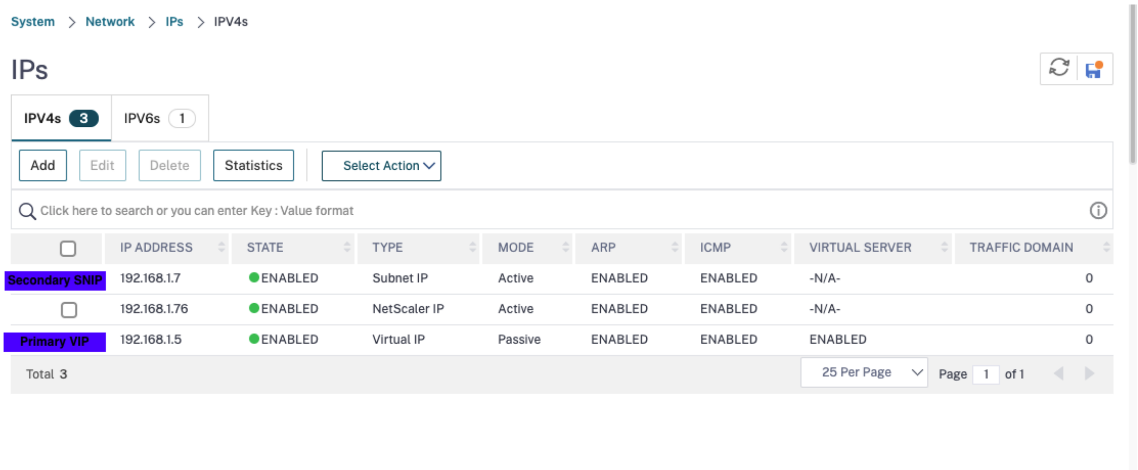
On the secondary node, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. To create a client alias IP (VIP) address:

    a) Enter the client alias IP address and netmask configured for the VPC subnet of the primary VM instance.
    b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
    c) Click **Create**.

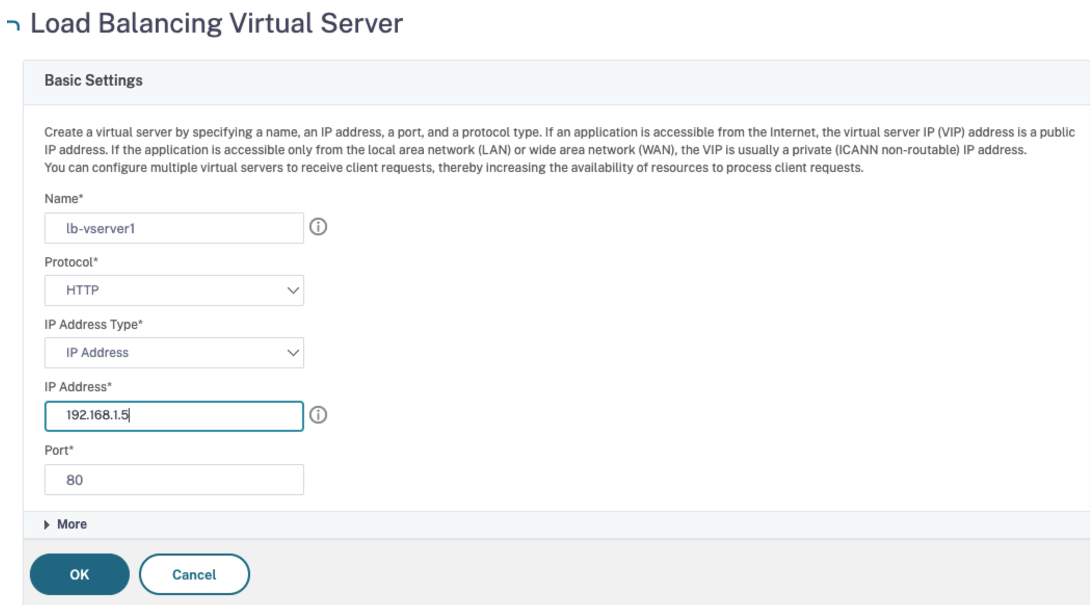
3. To create a server alias IP (SNIP) address:

    a) Enter the server alias IP address and netmask configured for the VPC subnet of the secondary VM instance.
    b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
    c) Click **Create**.



Step 3. Add a load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.
2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP Address (primary client alias IP address) and Port, and click **OK**.



Step 4. Add a service or service group on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

Step 5. Bind the service or service group to the load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 3**, and click **Edit**.

3. In the **Service and Service Groups** tab, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in the **Step 4**, and click **Bind**.

Step 6. Save the configuration.

After a forced failover, the secondary becomes the new primary. The client alias IP (VIP) from the old primary moves to the new primary.

Configure high availability by using the CLI

Step 1. Set up high availability in **INC Enabled** mode in both the instances by using the NetScaler CLI.

On the primary node, type the following command.

```
add ha node 1 <sec_ip> -inc ENABLED
```

On the secondary node, type the following command.

```
add ha node 1 <prim_ip> -inc ENABLED
```

The `sec_ip` refers to the internal IP address of the management NIC of the secondary node.

The `prim_ip` refers to the internal IP address of the management NIC of the primary node.

Step 2. Add VIP and SNIP on both primary and secondary nodes.

Type the following commands on the primary node:

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
```

> **Note:**
>
> Enter the alias IP address and netmask configured for the client subnet in the VM instance.

```
add ns ip <primary_server_alias_ip> <subnet> -type SNIP
```

Type the following commands on the secondary node:

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
```

> **Note:**
>
> Enter the alias IP address and netmask configured for the client subnet in the VM instance.

```
add ns ip <secondary_server_alias_ip> <subnet> -type SNIP
```


> **Note:**
>
> Enter the alias IP address and netmask configured for the server subnet in the VM instance.

Step 3. Add a virtual server on the primary node.

Type the following command:

```
add <server_type> vserver <vserver_name> <protocol> <
    primary_client_alias_ip> <port>
```

Step 4. Add a service or service group on the primary node.

Type the following command:

```
add service <service_name> <service_ip_address> <protocol> <port>
```

Step 5. Bind the service or service group to the load balancing virtual server on the primary node.

Type the following command:

```
bind <server_type> vserver <vserver_name> <service_name>
```

> **Note:**
>
> To save your configuration, type the command `save config`. Otherwise, the configurations are lost after you restart the instances.

Deploy a VPX high-availability pair with private IP address on Google Cloud Platform

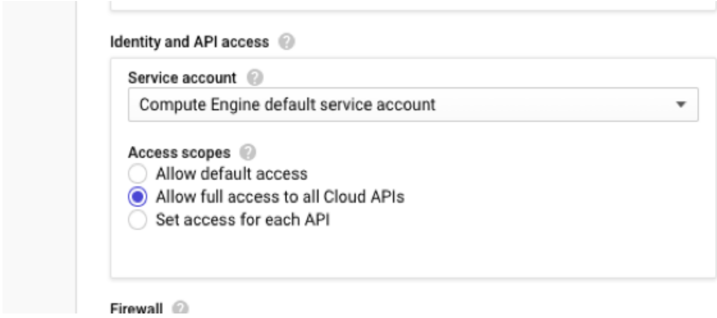
You can deploy a VPX high-availability pair on GCP using private IP address. The client IP (VIP) must be configured as alias IP address on the primary node. Upon failover, the Client IP address is moved to the secondary node, for the traffic to resume.

For more information on high availability, see High Availability.

Before you start

- Read the Limitation, Hardware requirements, Points to note mentioned in Deploy a NetScaler VPX instance on Google Cloud Platform. This information applies to high availability deployments also.
- Enable **Cloud Resource Manager API** for your GCP project.

- Allow full access to all Cloud APIs while creating the instances.



- Ensure that your GCP service account has the following IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata",
"compute.instances.get",
"compute.instances.list",
"compute.instances.updateNetworkInterface",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
```

- If you have configured external IP addresses on an interface other than the management interface, ensure that your GCP service account has the following additional IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.addresses.use"
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
]
```

- If your VMs do not have internet access, you must enable **Private Google Access** on the management subnet.

- If you have configured GCP forwarding rules on the primary node, read the limitations and requirements mentioned in Forwarding rules support for VPX high-availability pair on GCP to update them to new primary on failover.

How to deploy a VPX high availability pair on Google Cloud Platform

Here is a summary of the high availability deployment steps:

1. Create VPC networks in the same region. For example, Asia-east.
2. Create two VPX instances (primary and secondary nodes) on the same region. They can be in the same zone or different zones. For example Asia east-1a and Asia east-Ib.
3. Configure high availability settings on both instances by using the NetScaler GUI or ADC CLI commands.

Step 1. Create VPC networks

Create VPC networks based on your requirements. Citrix recommends you to create three VPC networks for associating with management NIC, client NIC, and server NIC.

To create a VPC network, perform these steps:

1. Log on the **Google console > Networking > VPC network > Create VPC Network**.
2. Complete the required fields, and click **Create**.

For more information, see the **Create VPC Networks** section in Deploy a NetScaler VPX instance on Google Cloud Platform.

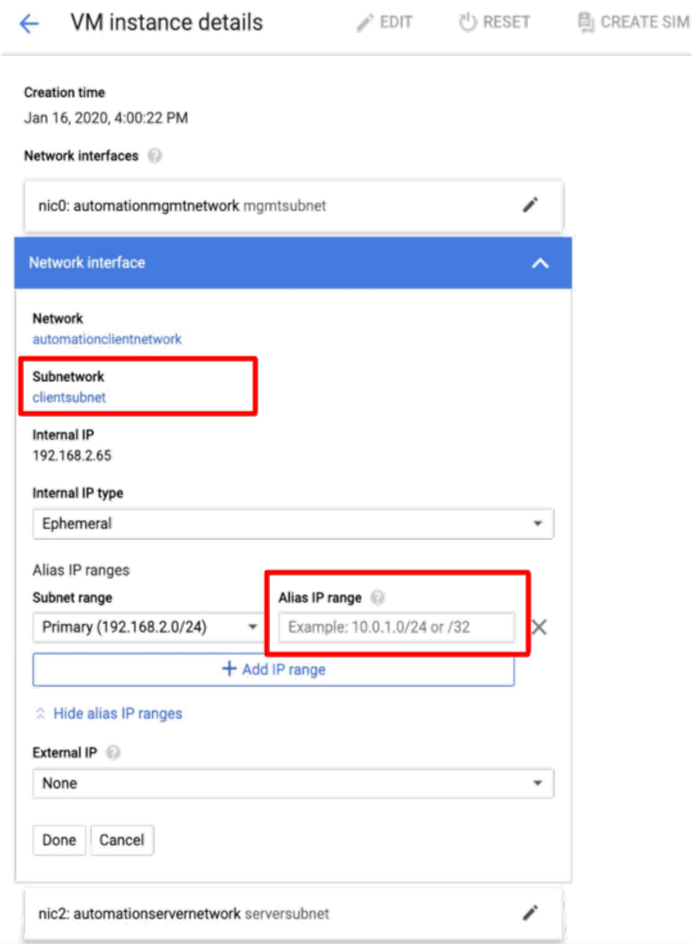
Step 2. Create two VPX instances

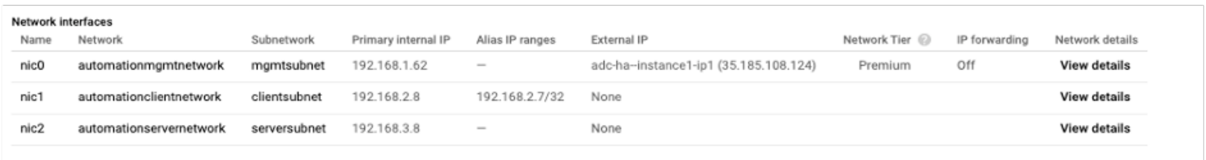
Create two VPX instances by following the steps given in Scenario: deploy a multi-NIC, multi-IP standalone VPX instance.

> **Important:**
>
> Assign a client alias IP address to the primary node. Do not use the internal IP address of the VPX instance to configure the VIP.

To create a client alias IP address, perform these steps:

1. Navigate to the VM instance and click **Edit**.
2. In the **Network Interface** window, edit the client interface.
3. In the **Alias IP range** field, enter the client alias IP address.




Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	automationmgmtnetwork	mgmtsubnet	192.168.1.62	—	adc-ha-instance1-ip1 (35.185.108.124)	Premium	Off	View details
nic1	automationclientnetwork	clientsubnet	192.168.2.8	192.168.2.7/32	None			View details
nic2	automationservernetwork	serversubnet	192.168.3.8	—	None			View details

After the failover, when the old primary becomes the new secondary, the alias IP addresses move from the old primary and is attached to the new primary.

After you have configured the VPX instances, you can configure the Virtual (VIP) and Subnet IP (SNIP) addresses. For more information, see Configuring NetScaler-owned IP addresses.

Step 3. Configure high availability

After you've created the instances on Google Cloud Platform, you can configure high availability by using the NetScaler GUI or CLI.

Configure high availability by using the GUI

Step 1. Set up high availability in INC Enabled mode on both the nodes.

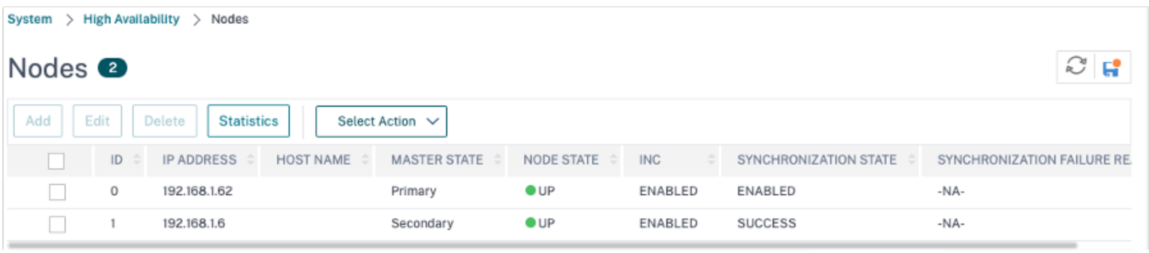
On the **primary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the secondary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

On the **secondary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the primary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

Before you proceed further, ensure that the Synchronization state of the secondary node is shown as **SUCCESS** in the **Nodes** page.



> **Note:**
>
> After the secondary node is synchronized with the primary node, the secondary node has the same log-on credentials as the primary node.

Step 2. Add Virtual IP address and Subnet IP address on both the nodes.

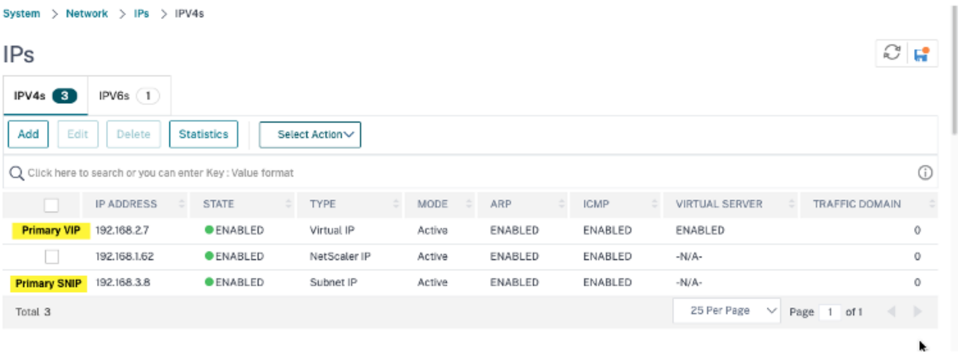
On the primary node, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. To create a client alias IP (VIP) address:

   a) Enter the Alias IP address and netmask configured for the client subnet in the VM instance.
   b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
   c) Click **Create**.

3. To create a server IP (SNIP) address:

   a) Enter the internal IP address of the server-facing interface of the primary instance and netmask configured for the server subnet.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.



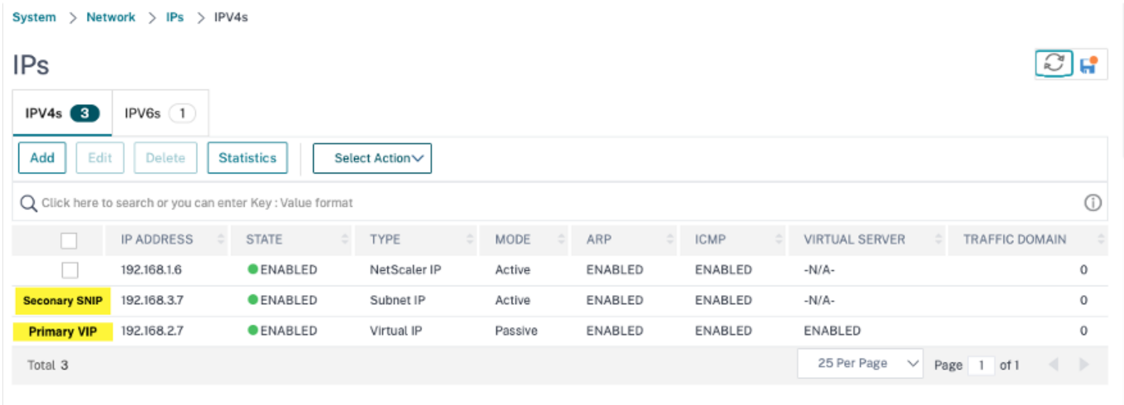
On the secondary node, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. To create a client alias IP (VIP) address:

   a) Enter the Alias IP address and netmask configured for the client subnet on the primary VM instance.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.

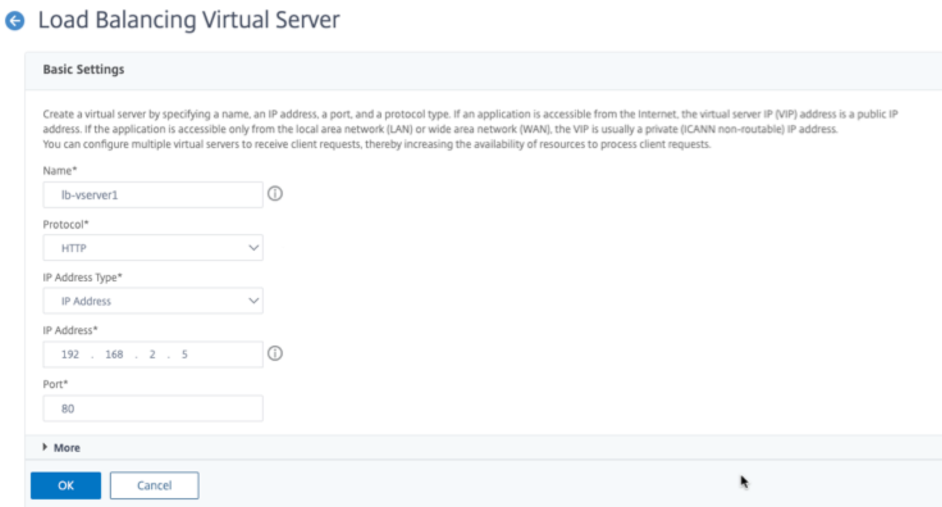
3. To create a server IP (SNIP) address:

   a) Enter the internal IP address of the server-facing interface of the secondary instance and netmask configured for the server subnet.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.



Step 3. Add a load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.
2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP Address (primary client alias IP address) and Port, and click **OK**.



Step 4. Add a service or service group on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

Step 5. Bind the service or service group to the load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 3**, and click **Edit**.
3. In the **Service and Service Groups** tab, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in the **Step 4**, and click **Bind**.

Step 5. Save the configuration.

After a forced failover, the secondary becomes the new primary. The client alias IP (VIP) and the server alias IP (SNIP) from the old primary moves to the new primary.

Configure high availability by using the CLI

Step 1. Set up high availability in **INC Enabled** mode in both the instances by using the NetScaler CLI.

On the primary node, type the following command.

```
add ha node 1 <sec_ip> -inc ENABLED
```

On the secondary node, type the following command.

```
add ha node 1 <prim_ip> -inc ENABLED
```

The `sec_ip` refers to the internal IP address of the management NIC of the secondary node.

The `prim_ip` refers to the internal IP address of the management NIC of the primary node.

Step 2. Add VIP and SNIP on both nodes.

Type the following commands on the primary node:

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
```

> **Note:**
>
> Enter the Alias IP address and netmask configured for the client subnet in the VM instance.

```
add ns ip <primary_snip> <subnet> -type SNIP
```

The `primary_snip` refers to the internal IP address of the server-facing interface of the primary instance.

Type the following commands on the secondary node:

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
```

> **Note:**
>
> Enter the Alias IP address and netmask configured for the client subnet on the primary VM instance.

```
add ns ip <secondary_snip> <subnet> -type SNIP
```

The `secondary_snip` refers to the internal IP address of the server-facing interface of the secondary instance.

> **Note:**
>
> Enter the IP address and netmask configured for the server subnet in the VM instance.

Step 3. Add a virtual server on the primary node.

Type the following command:

```
add <server_type> vserver <vserver_name> <protocol> <
    primary_client_alias_ip> <port>
```

Step 4. Add a service or service group on the primary node.

Type the following command:

```
add service <service_name> <service_ip_address> <protocol> <port>
```

Step 5. Bind the service or service group to the load balancing virtual server on the primary node.

Type the following command:

```
bind <server_type> vserver <vserver_name> <service_name>
```

> **Note:**
>
> To save your configuration, type the command `save config`. Otherwise, the configurations are lost after you restart the instances.

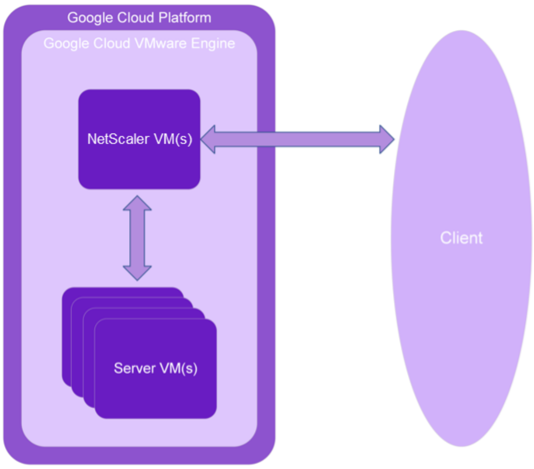
Install a NetScaler VPX instance on Google Cloud VMware Engine

Google Cloud VMware Engine (GCVE) provides you with private clouds that contain vSphere clusters, built from dedicated bare-metal Google Cloud Platform infrastructure. The minimum initial deployment is three hosts, but additional hosts can be added one at a time. All provisioned private clouds have vCenter Server, vSAN, vSphere, and NSX-T.

GCVE enables you to create cloud software-defined data centers (SDDC) on Google Cloud Platform with the desired number of ESX hosts. GCVE supports NetScaler VPX deployments. GCVE provides a user interface same as on-prem vCenter. It functions identical to the ESX-based NetScaler VPX deployments.

The following diagram shows the GCVE on the Google Cloud Platform that an administrator or a client can access over the internet. An administrator can create, manage, and configure workload or server VMs using GCVE. The admin can access the GCVE's web-based vCenter and NSX-T Manager using an OpenVPN connection. You can create the NetScaler VPX instances (standalone or HA pair) and server VMs within GCVE using vCenter, and manage the corresponding networking using NSX-T manager. The

NetScaler VPX instance on GCVE works similar to the On-prem VMware cluster of hosts. GCVE can be managed using OpenVPN connection to the management infrastructure.



Prerequisites

Before you begin installing a virtual appliance, do the following:

- For more information on Google Cloud VMware Engine and its prerequisites, see Google Cloud VMware Engine documentation.
- For more information on deploying Google Cloud VMware Engine, see Deploy a Google Cloud VMware Engine private cloud.
- For more information on connecting to your private cloud using a point-to-site VPN gateway to access and manage Google Cloud VMware Engine, see Access an Google Cloud VMware Engine private cloud.
- On VPN client machine, download the NetScaler VPX appliance setup files.
- Create appropriate NSX-T network segments on VMware SDDC to which the virtual machines connect. For more information, see Add a network segment in Google Cloud VMware Engine.
- Obtain VPX license files. For more information about NetScaler VPX instance licenses, see Licensing overview.
- Virtual machines (VMs) created or migrated to the GCVE private cloud must be attached to a network segment.

VMware cloud hardware requirements

The following table lists the virtual computing resources that the VMware SDDC must provide for each VPX nCore virtual appliance.

Table 1. Minimum virtual computing resources required for running a NetScaler VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	In VMware SDDC, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded to version 7 or higher.
Disk space	20 GB

> **Note:**
>
> This is in addition to any disk requirements for the hypervisor.

For production use of the VPX virtual appliance, the full memory allocation must be reserved.

OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. The following table describes the minimum system requirements for installing OVF tool.

Table 2. Minimum system requirements for OVF tool installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended
RAM	1 GB Minimum, 2 GB recommended
NIC	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide"PDF file at http://kb.vmware.com/.

Downloading the NetScaler VPX setup files

The NetScaler VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log on. If you do not have a Citrix account, access the home page at http://www.citrix.com. Click the **New Users link**, and follow the instructions to create a new Citrix account.

Once logged on, navigate the following path from the Citrix home page:

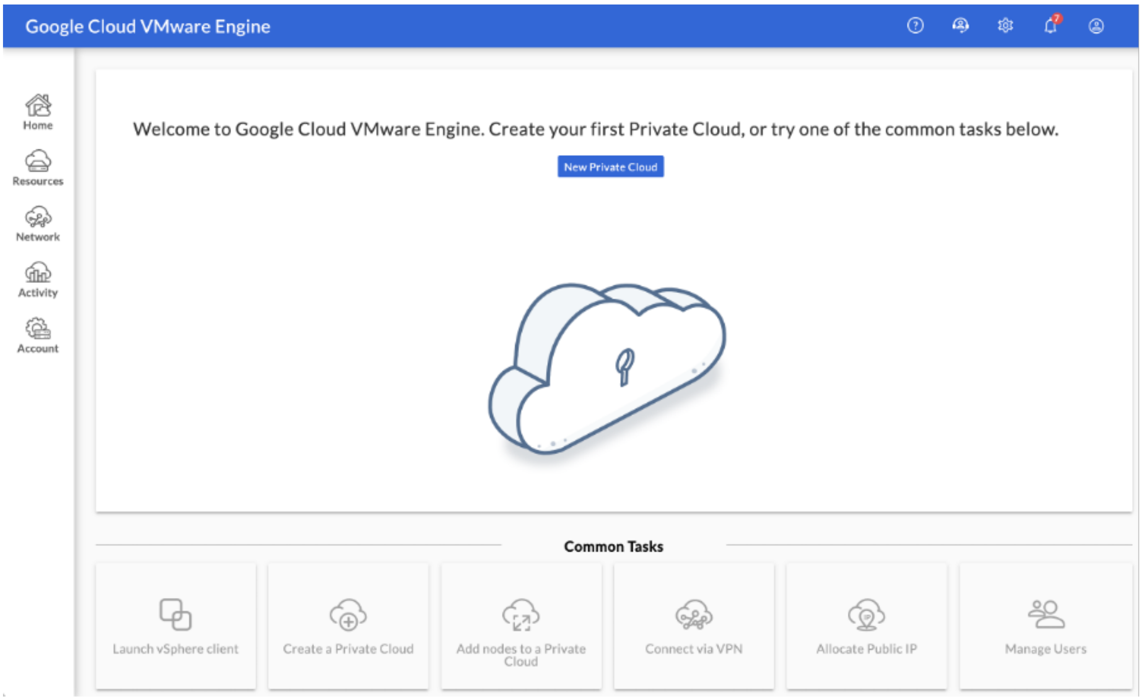
Citrix.com > **Downloads > NetScaler > Virtual Appliances**.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-13.0-79.64.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-13.0-79.64.mf)

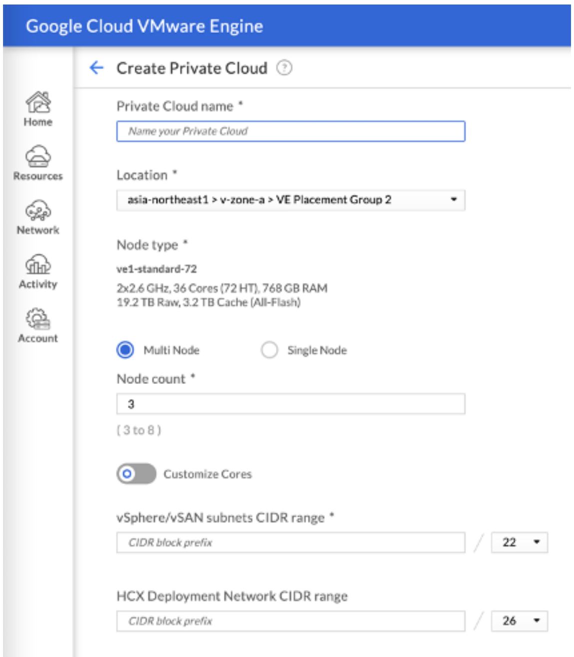
Deploy Google Cloud VMware Engine

1. Log in to your GCVE portal, and navigate to **Home**.

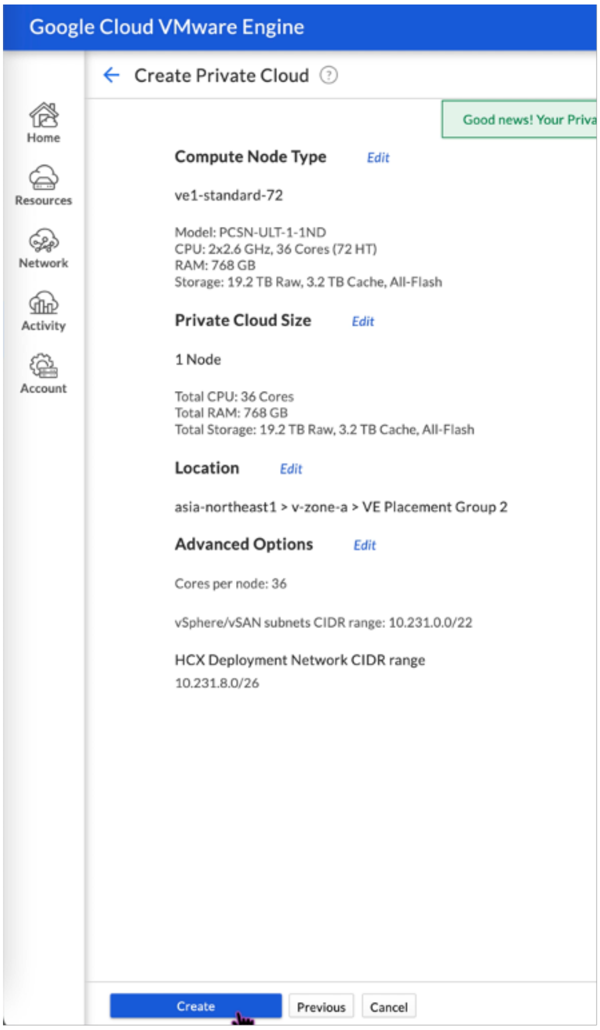


2. In the **New Private Cloud** page, enter the following details:

   - Select a minimum of 3 ESXi hosts to create the default cluster of your private cloud.
   - For the **vSphere/vSan subnet CIDR range** field, use /22 address space.
   - For the **HCX Deployment Network CIDR range** field, use /26 address space.
   - For the virtual network, make sure that the CIDR range doesn't overlap with any of your on-premises or other GCP subnets (virtual networks).

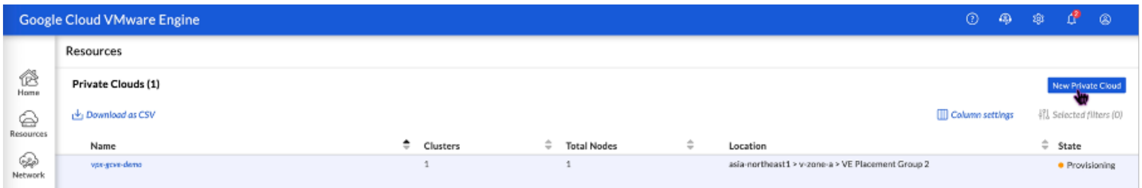


3. Click **Review and Create**.
4. Review the settings. If you need to change any settings, click **Previous**.



5. Click **Create**. Private Cloud provisioning process starts. It can take up to two hours for the Private Cloud to be provisioned.

6. Go to **Resources** to verify the private cloud that is created.

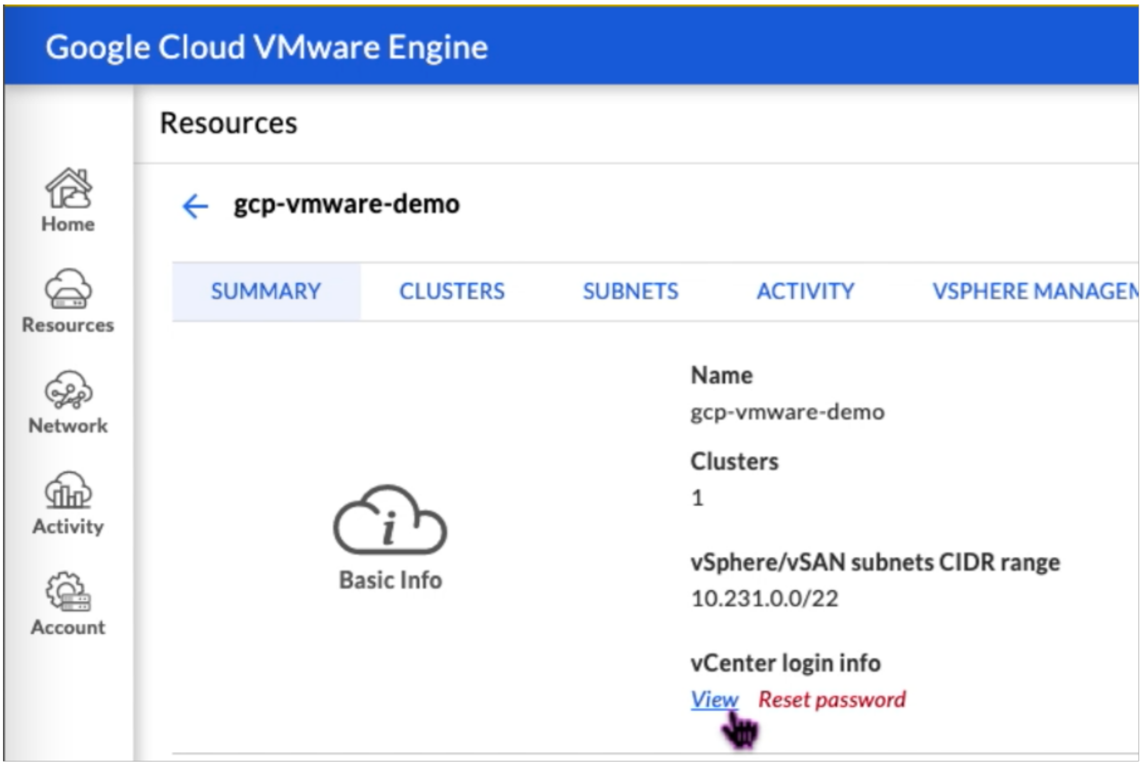


7. To access this resource, you must connect to GCVE using point-to-site VPN. For more information, see the following documentation:

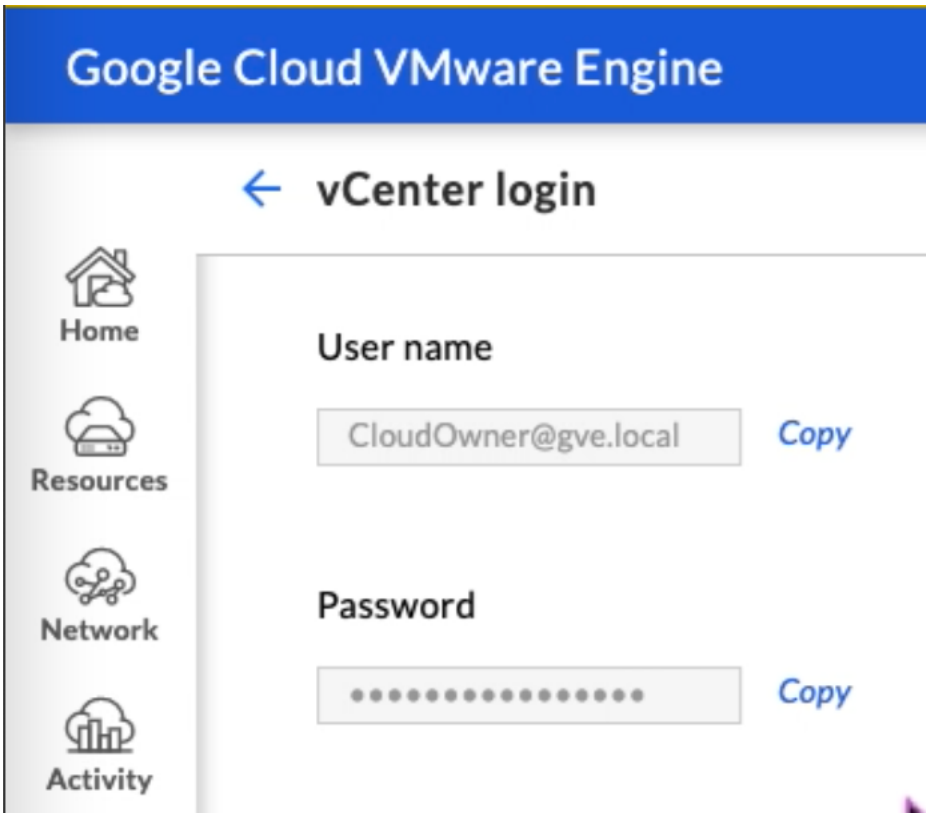
   - VPN gateways
   - Connecting using VPN

Access your Private Cloud vCenter portal

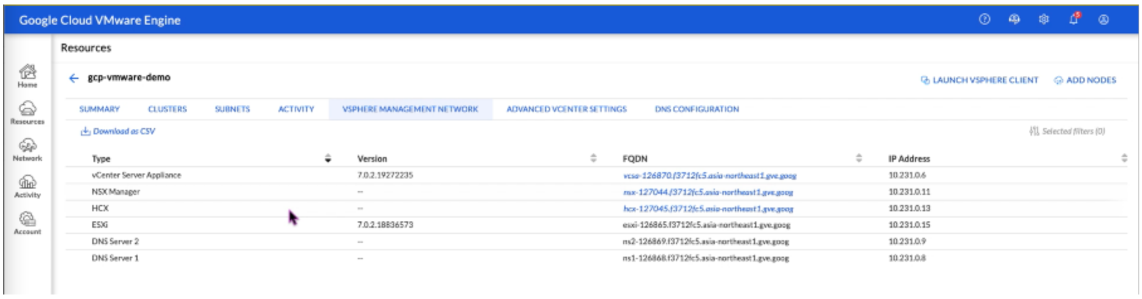
1. Navigate to your Google Cloud VMware Engine private cloud. In the **SUMMARY** tab, under **vCenter Login Info**, click **View**.



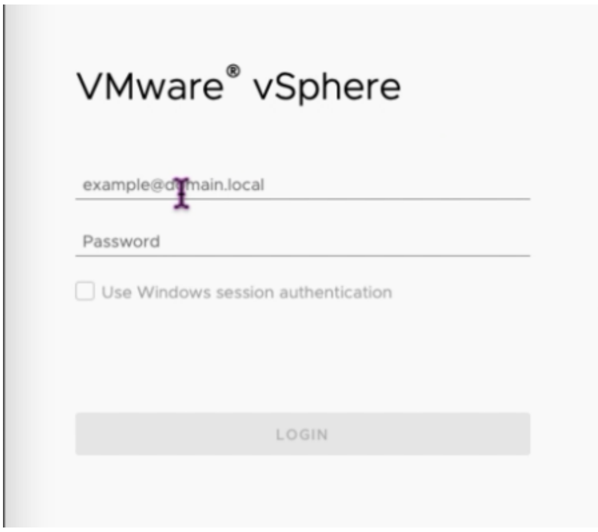
2. Make note of the vCenter credentials.



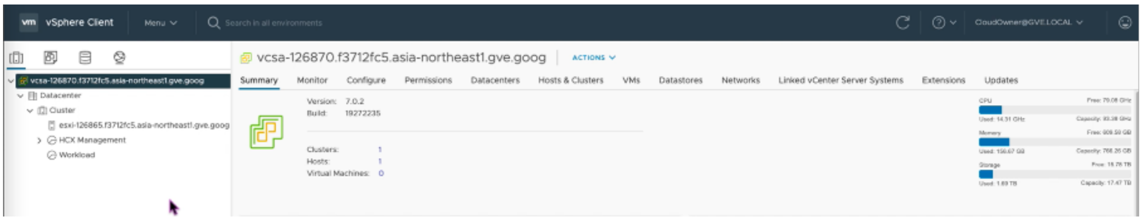
3. Launch the vSphere client by clicking **LAUNCH VSPHERE CLIENT** or navigate to **VSPHERE MANAGEMENT NETWORK** and click the **vCenter Server Appliance** FQDN.



4. Log in to VMware vSphere using vCenter credentials noted in Step 2 of this procedure.



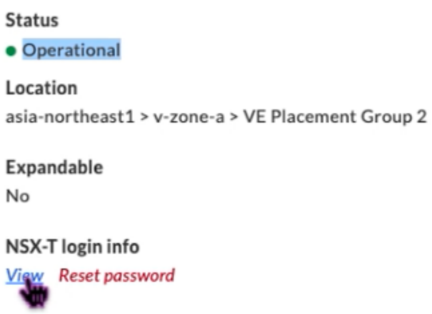
5. In vSphere client, you can verify the ESXi hosts that you created in GCVE portal.



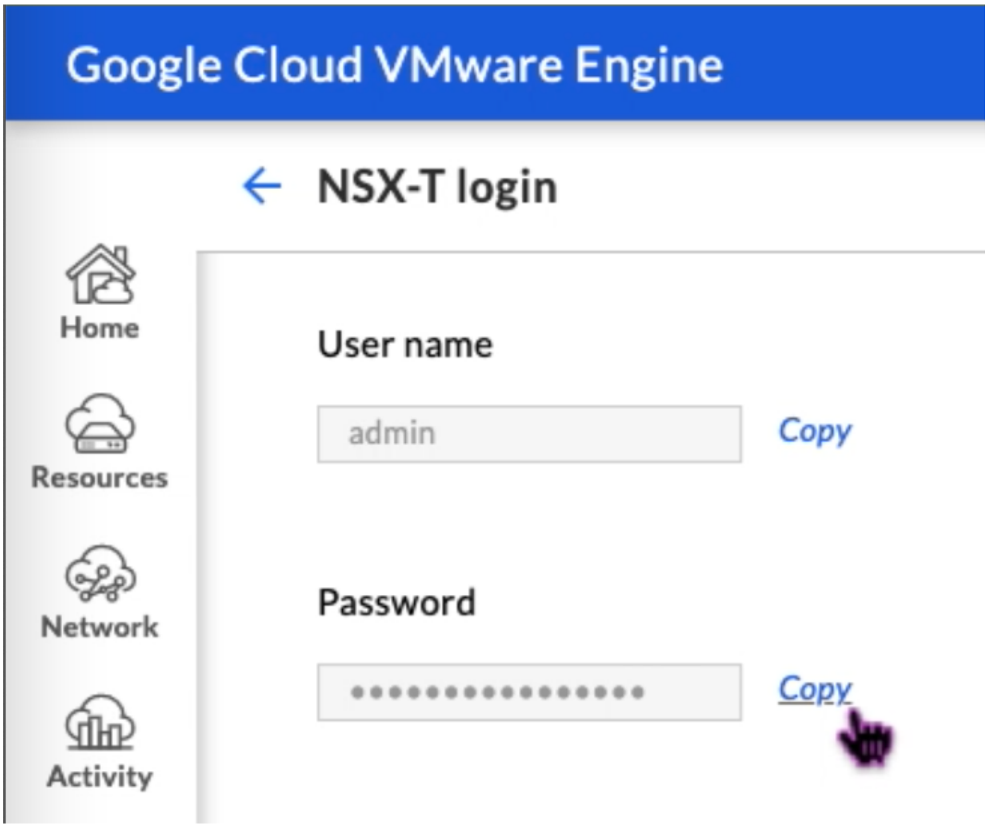
Create an NSX-T segment in the GCVE NSX-T portal

You can create and configure an NSX-T segment from the NSX Manager in the Google Cloud VMware Engine console. These segments are connected to the default Tier-1 gateway, and the workloads on these segments get East-West and North-South connectivity. Once you create the segment, it displays in vCenter.

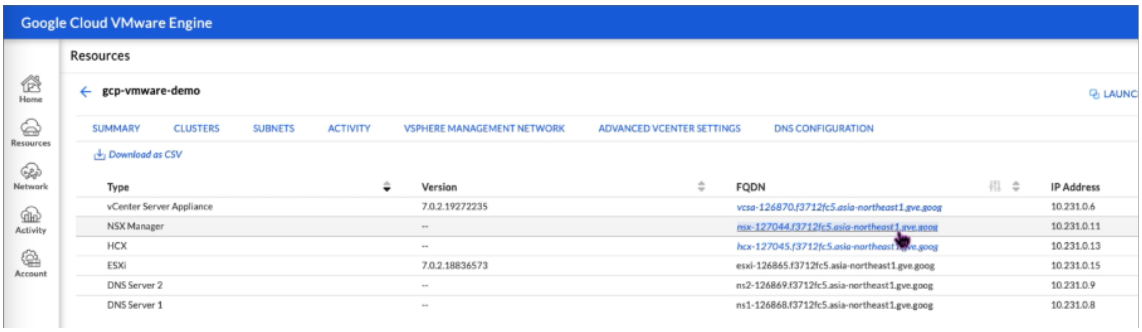
1. In your GCVE private cloud, under **Summary > NSX-T login info**, select **View**.



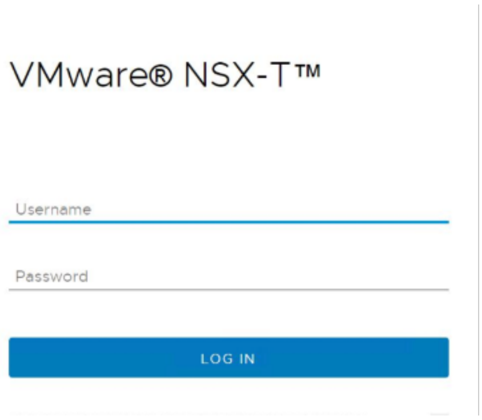
2. Make note of the NSX-T credentials.



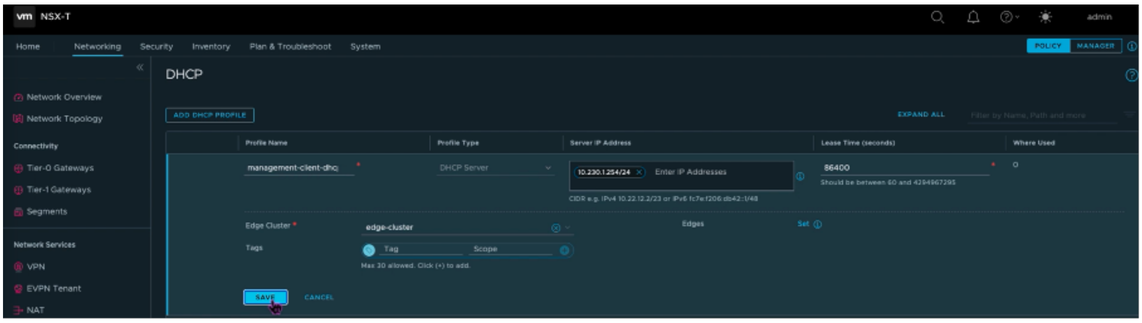
3. Launch the NSX Manager by navigating to **VSPHERE MANAGEMENT NETWORK** and click the **NSX Manager** FQDN.



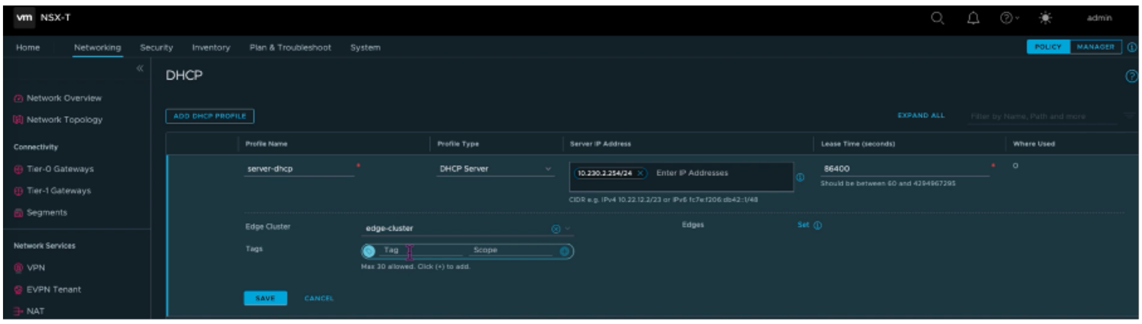
4. Log in to the NSX Manager using the credentials noted in Step 2 of this procedure.



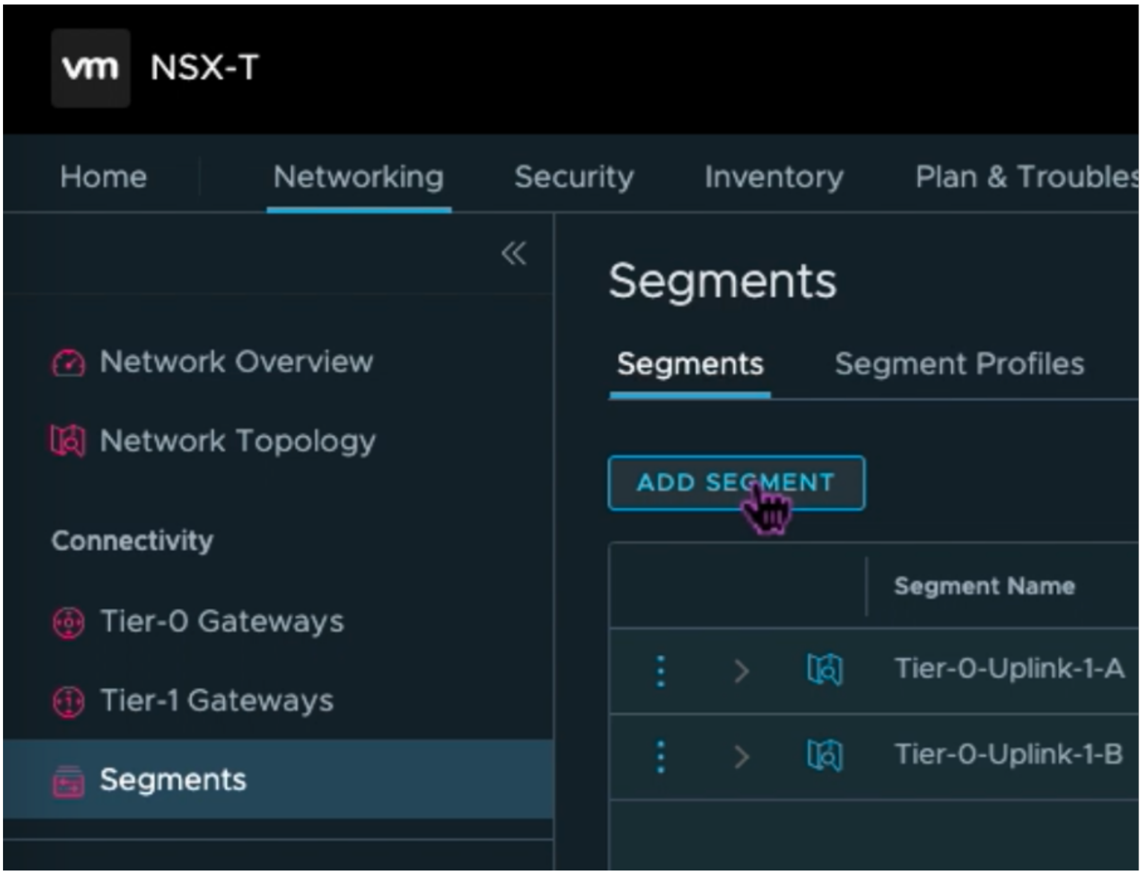
5. Set up DHCP service for the new segments or subnets.
6. Before you can create a subnet, set up a DHCP service.
7. In NSX-T, go to **Networking > DHCP**. The networking dashboard shows that the service creates one tier-0 and one tier-1 gateway.
8. To begin provisioning a DHCP server, click **Add DHCP Profile**.
9. In the DHCP name field, enter a name for the **Client-Management** profile.
10. Select **DHCP server** as the Profile type.
11. In the **Server IP address** column, provide a DHCP service IP address range.
12. Select your **Edge Cluster**.
13. Click **Save** to create the DHCP service.



14. Repeat Steps 6 to 13 for Server DHCP range.



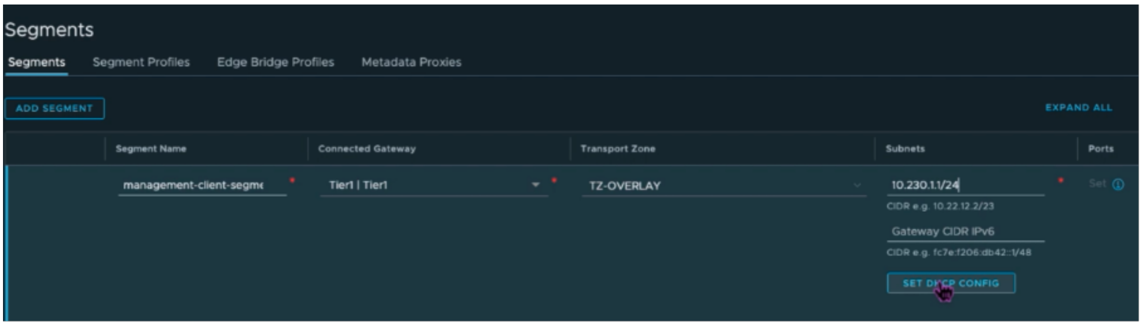
15. Create two separate segments: one for Client and Management interfaces, and another for Server interfaces.
16. In NSX-T, go to **Networking > Segments**.
17. Click **Add Segment**.



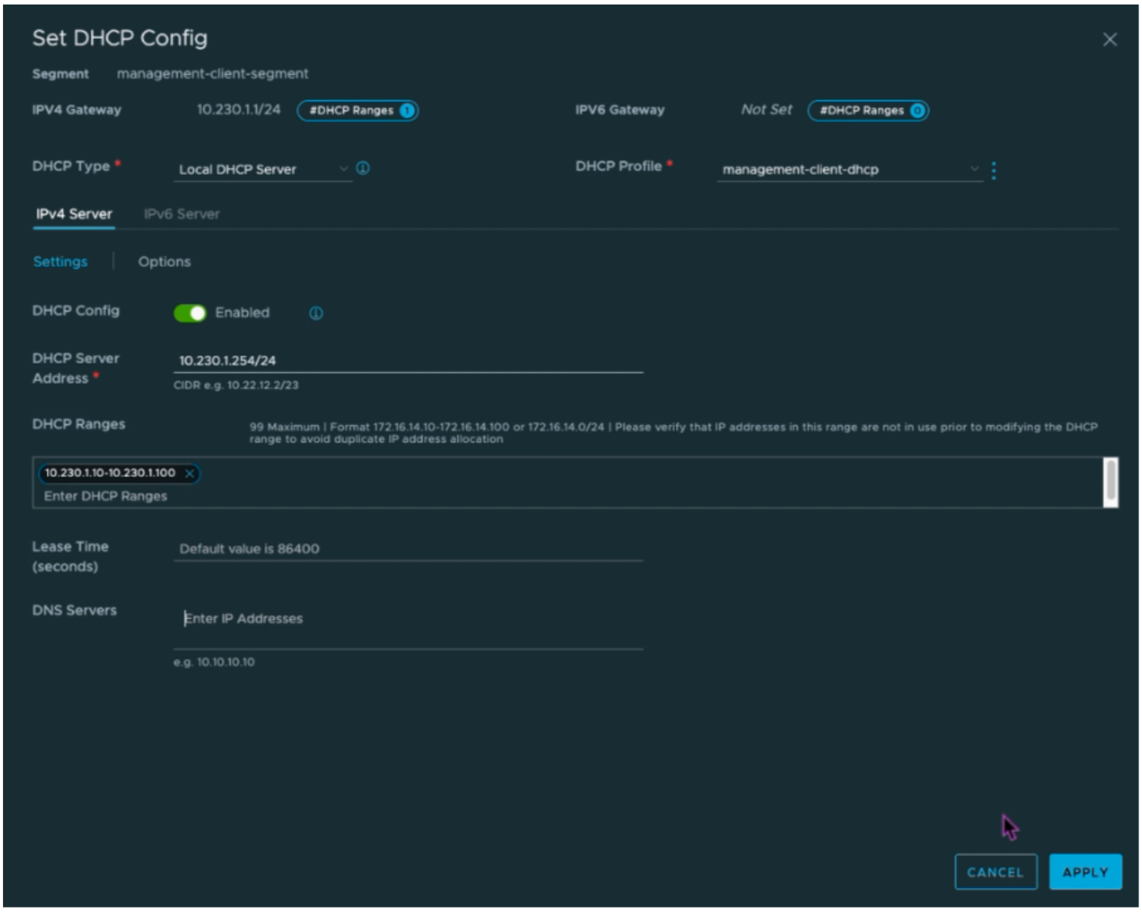
18. In the **Segment Name** field, enter a name for your **Client Management** segment.
19. In the **Connected Gateway** list, select **Tier1** to connect to the tier-1 gateway.

In the **Transport Zone** list, select **TZ-OVERLAY   Overlay**.

20.
21. In the **Subnets** column, enter the subnet range. Specify the subnet range with `.1` as the last octet. For example, 10.12.2.1/24.

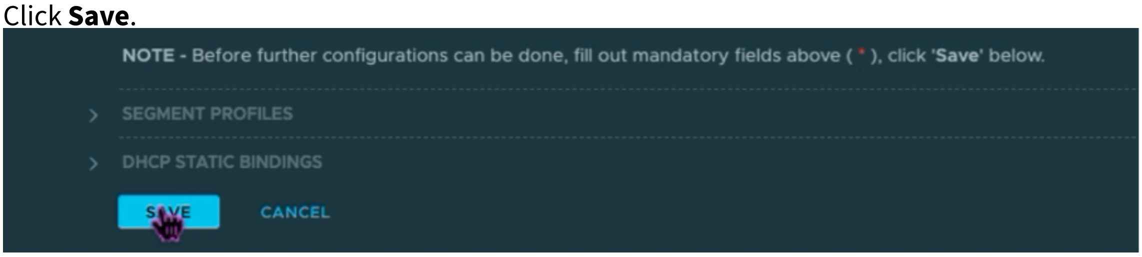


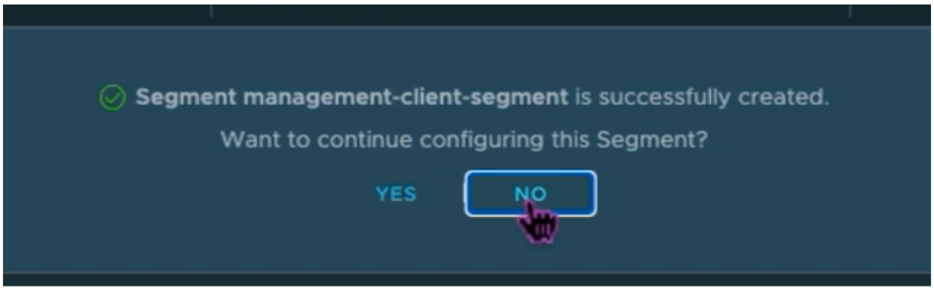
22. Click **Set DHCP Config**, and provide values for the **DHCP Ranges** field.



23. Click **Apply** to save your DHCP configuration.

24. Click **Save**.





25. Repeat Steps 17 to 24 for Server segment as well.
26. You can now select these network segments in vCenter when creating a VM.

For more information, see Creating your first subnet.

Install a NetScaler VPX instance on VMware cloud

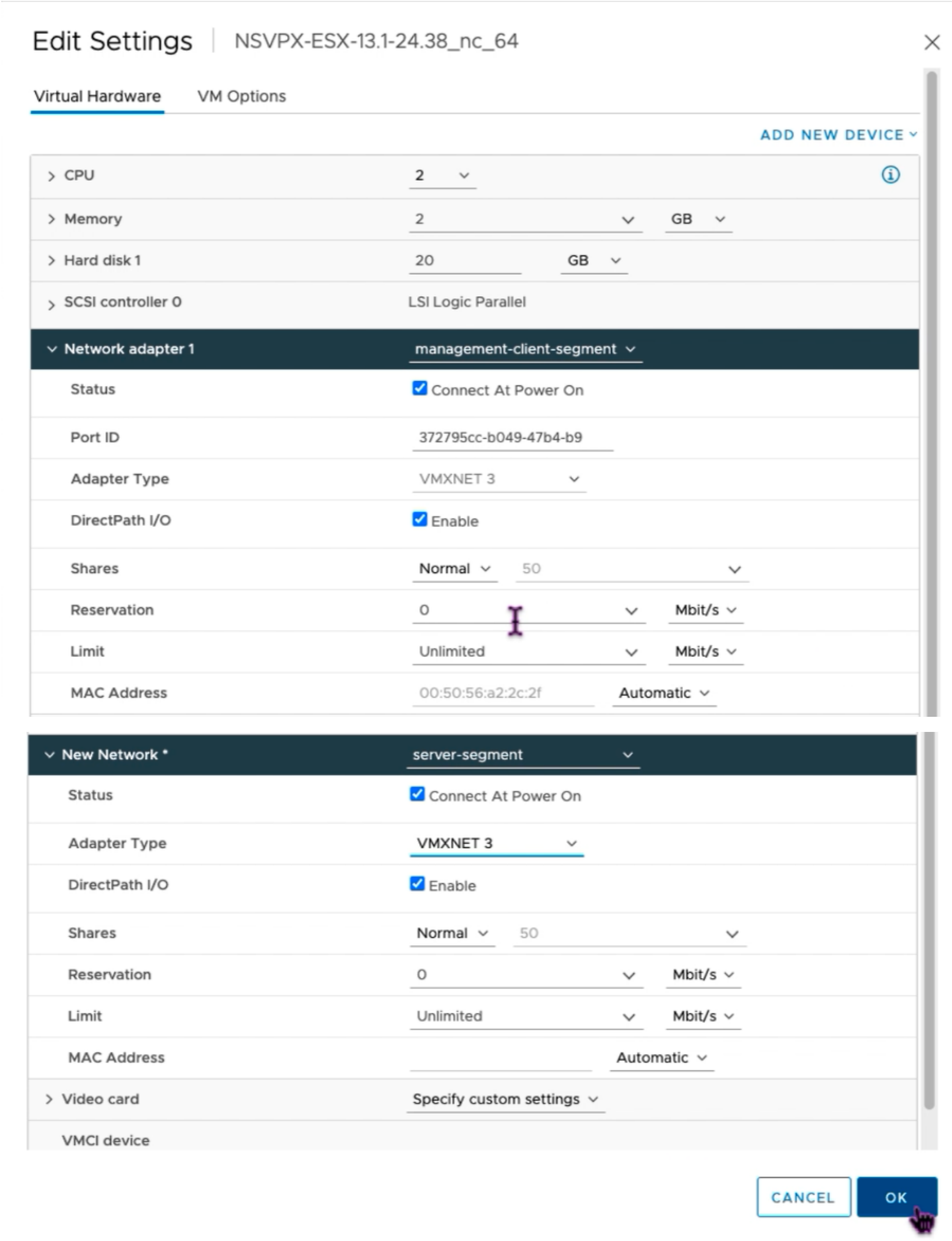
After you have installed and configured Private Cloud on GCVE, you can use the vCenter to install virtual appliances on the VMware Engine. The number of virtual appliances that you can install depends on the amount of resource available on the Private Cloud.

To install NetScaler VPX instances on Private Cloud, perform these steps on a desktop connected to private cloud point-to-site VPN:

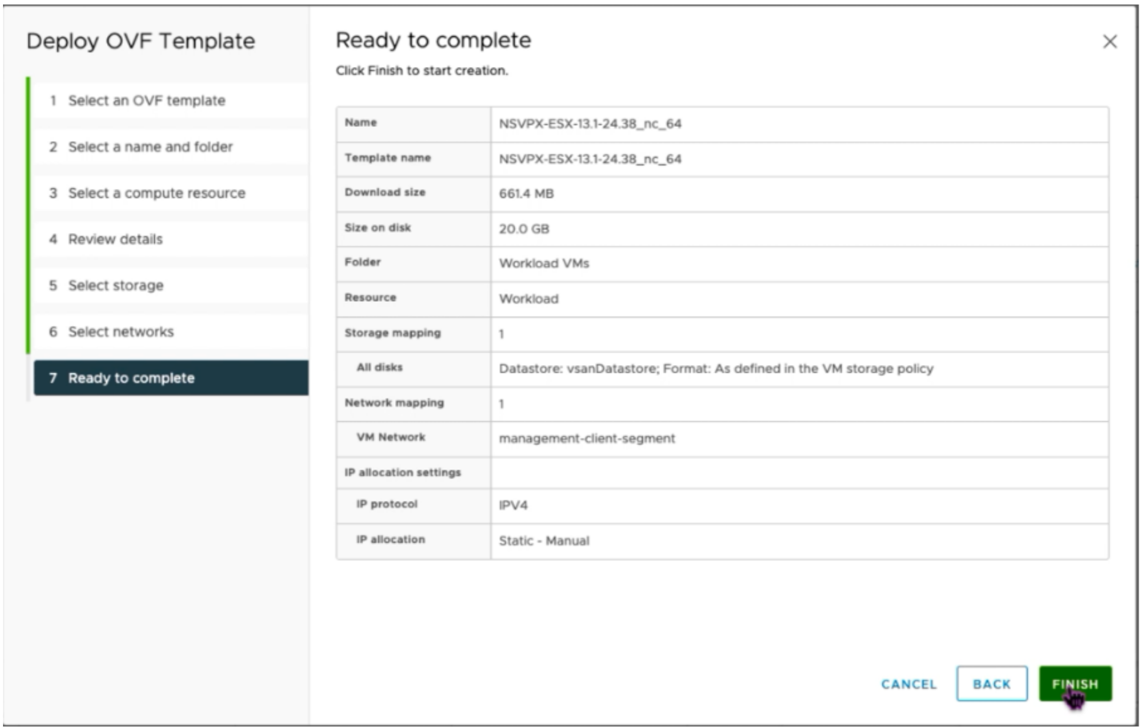
1. Download the NetScaler VPX instance setup files for ESXi host from the NetScaler downloads site.
2. Open VMware vCenter in a browser connected to your private cloud point-to-site VPN.
3. In the **User Name** and **Password** fields, type the administrator credentials, and then click **Login**.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file** field, browse to the location at which you saved the NetScaler VPX instance setup files, select the .ovf file, and click **Next**.

   > **Note:**
   >
   > By default, the NetScaler VPX instance uses E1000 network interfaces. To deploy ADC with the VMXNET3 interface, modify the OVF to use VMXNET3 interface instead of E1000. Availability of VMXNET3 interface is limited by GCP infrastructure and might not be available in Google Cloud VMware Engine.

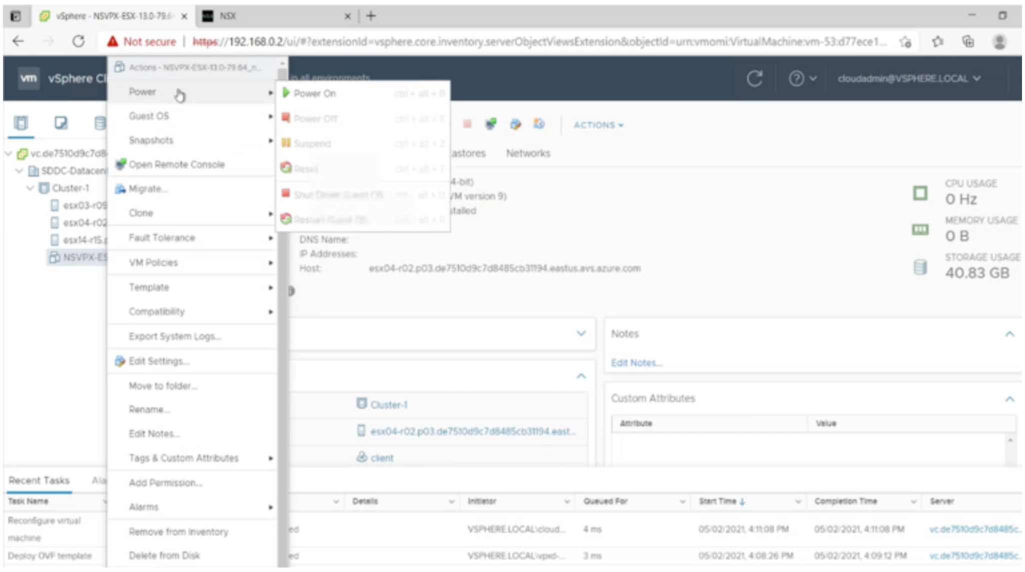
6. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the NSX-T Manager. Click **OK**.



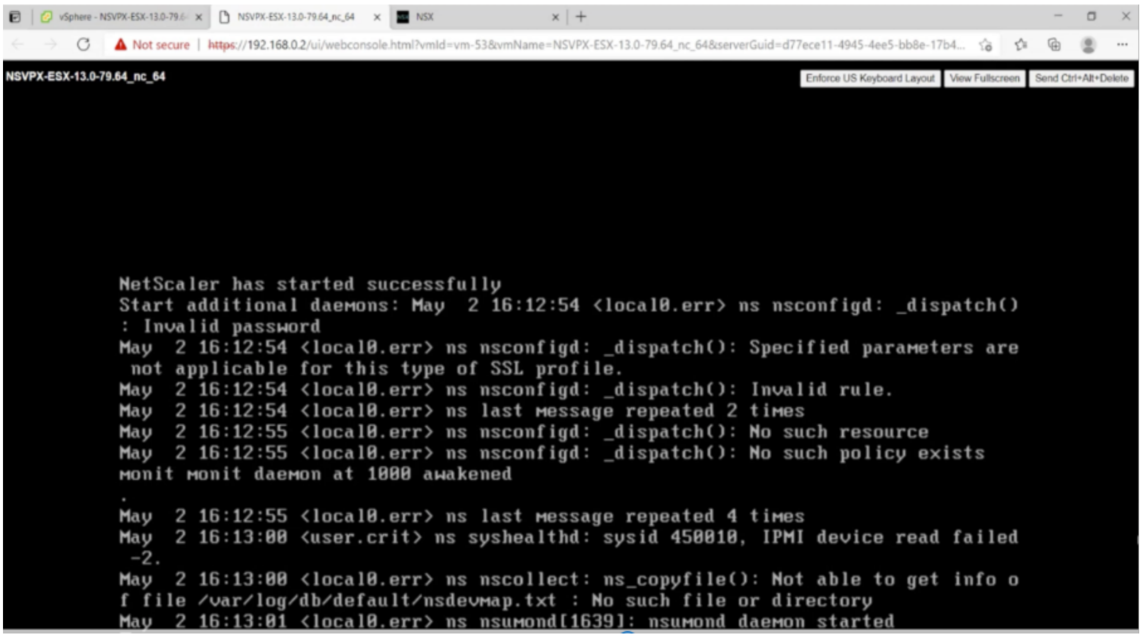
7. Click **Finish** to start installing a virtual appliance on VMware cloud.



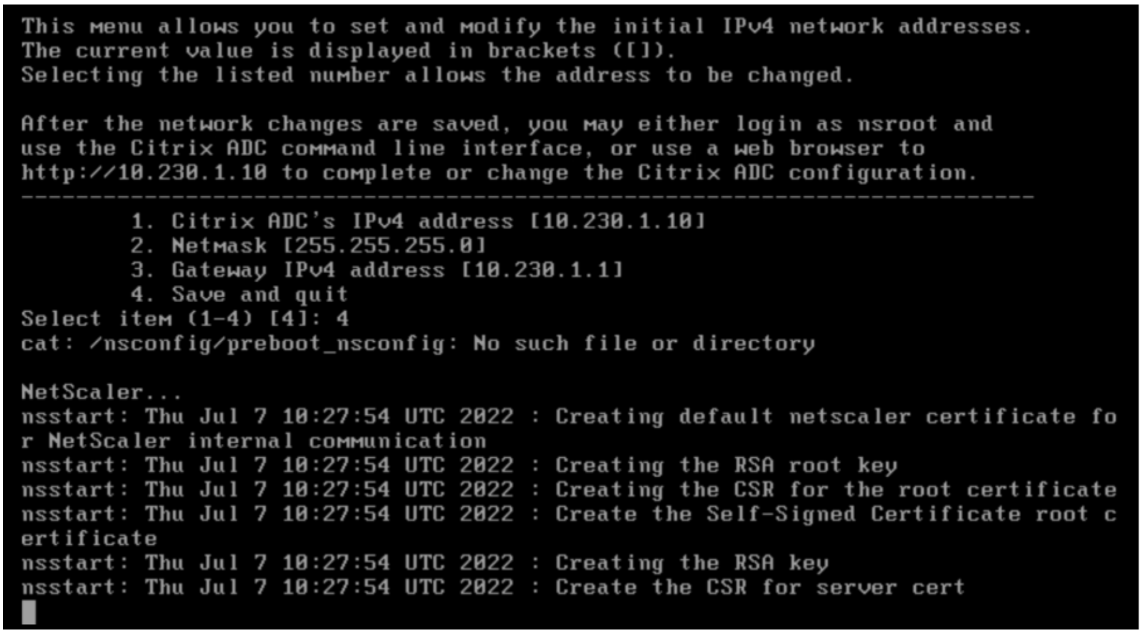
8. You are now ready to start the NetScaler VPX instance. In the navigation pane, select the NetScaler VPX instance that you have installed and, from the right-click menu, select **Power On**. Click the **Launch Web Console** tab to emulate a console port.



9. You are now connected to the NetScaler VM from the vSphere client.



10. On first boot, set the management IP and gateway for the ADC instance.



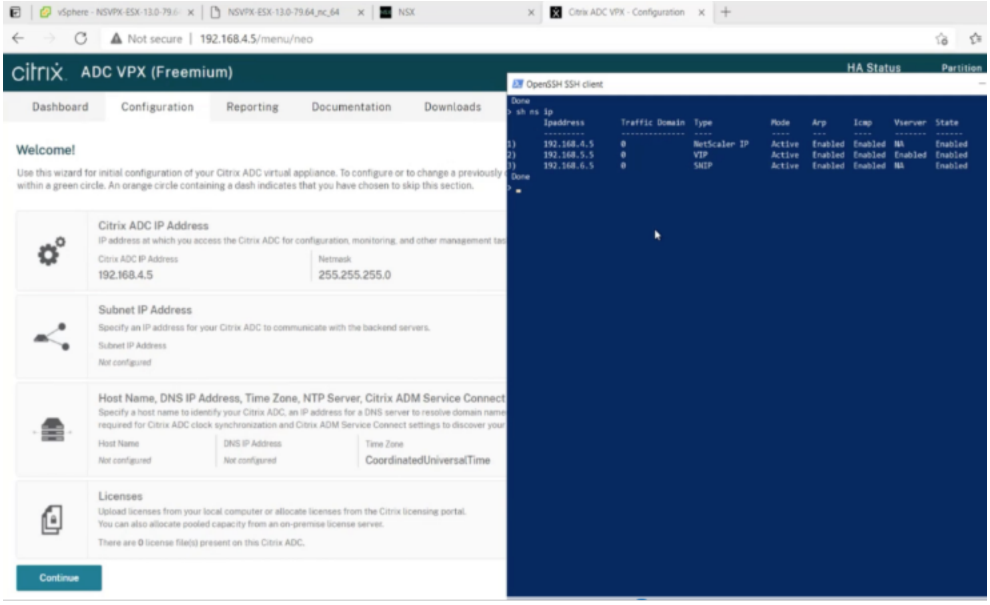
11. To access the NetScaler appliance by using the SSH keys, type the following command in the CLI:

```
ssh nsroot@<management IP address>
```

Example:

```
ssh nsroot@10.230.1.10
```

12. You can verify the ADC configuration by using the `show ns ip` command.

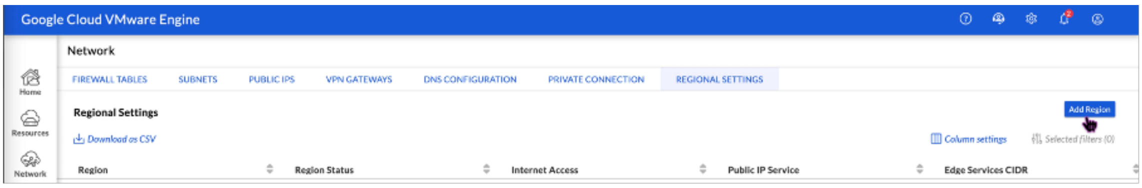


Assign a Public IP address to a NetScaler VPX instance on VMware cloud

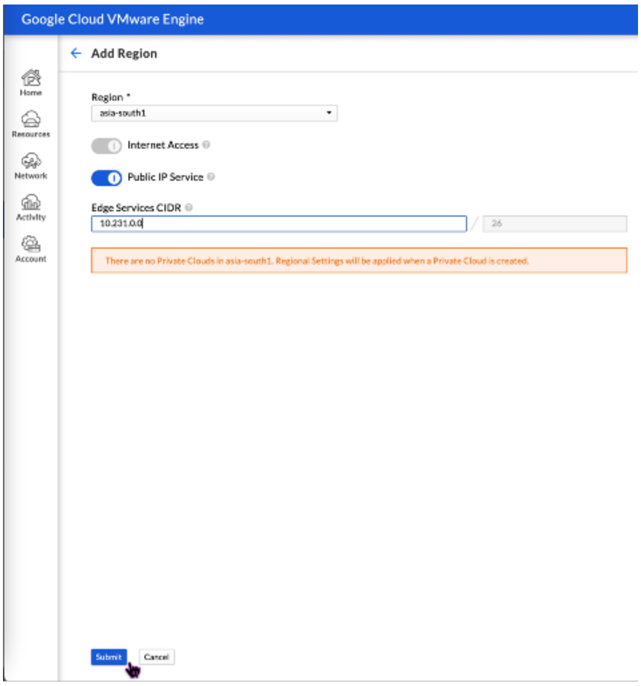
After you have installed and configured NetScaler VPX instance on GCVE, you must assign a public IP address to the Client interface. Before assigning public IP addresses to your VMs, make sure that Public IP service is enabled for your Google Cloud region.

To enable Public IP service for a new region, follow these steps:

1. On GCVE console, navigate to **Network > REGIONAL SETTINGS > Add Region**.



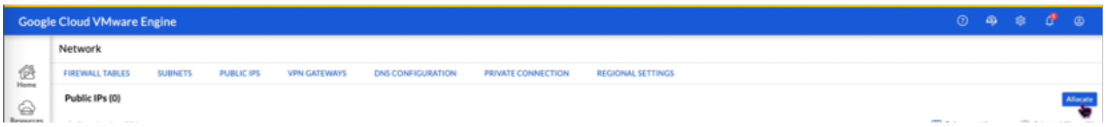
2. Select your region and enable **Internet Access** and **Public IP Service**.
3. Assign an Edge Services CIDR making sure that the CIDR range doesn't overlap with any of your on-premises or other GCP/GCVE subnets (virtual networks).



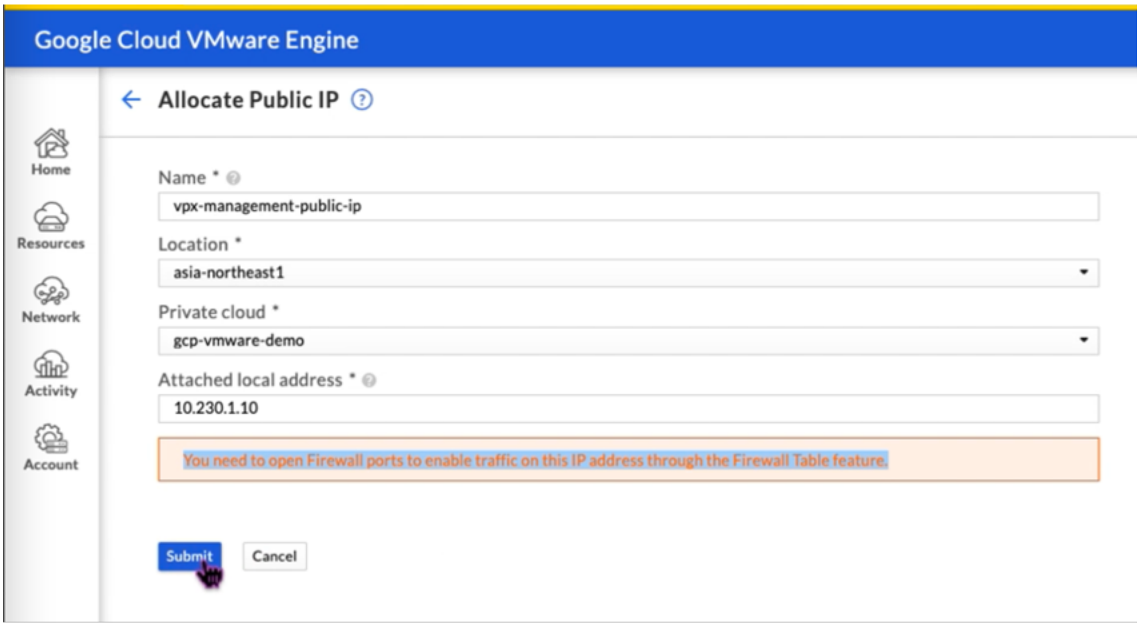
4. Public IP Service will be enabled for the selected region in a few minutes.

To assign public IP to the Client interface on the NetScaler VPX instance on GCVE, perform these steps on GCVE portal:

1. On GCVE console, navigate to **Network > PUBLIC IPS > Allocate**.



2. Enter a name for the public IP. Select your region, and select the private cloud where the IP will be used.
3. Provide the private IP for the interface to which you want the public IP to be mapped. This will be the **private IP** for your **Client** interface.
4. Click **Submit**.



5. Public IP is ready to use in a few minutes.
6. You must add Firewall rules to allow access to the public IP before you can use it. For more information, see Firewall rules.

Add back-end GCP Autoscaling service

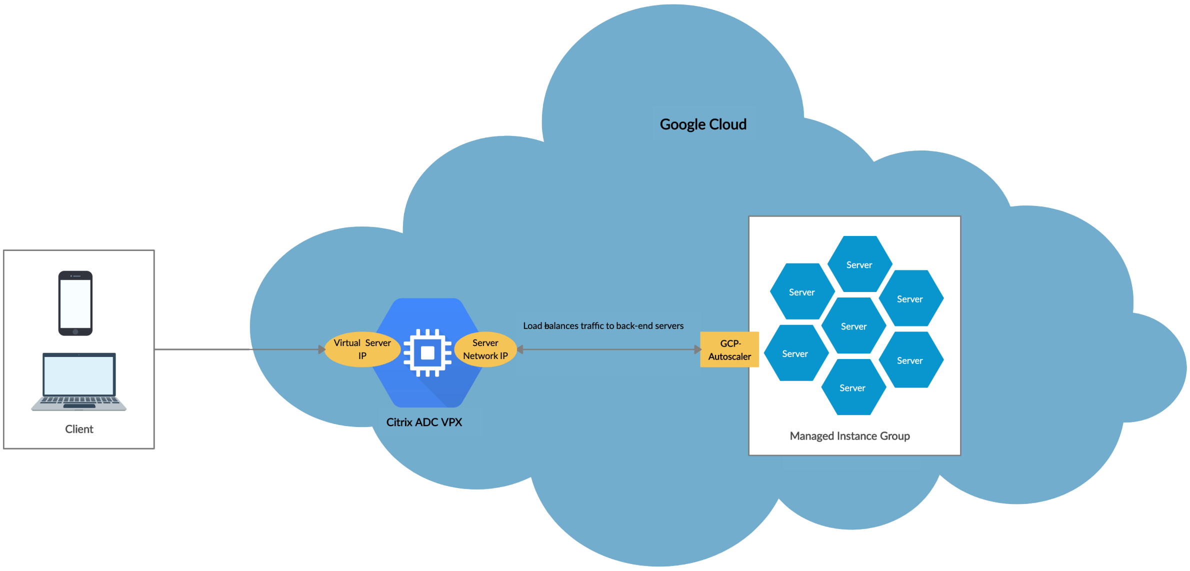
Efficient hosting of applications in a cloud requires easy and cost-effective management of resources, depending on the application demand. To meet the increasing demand, you have to scale network resources upward. When demand subsides, you need to scale down to avoid the unnecessary cost of underutilized resources. To minimize the cost of running the application, you have to constantly monitor traffic, memory and CPU use, and so on. However, monitoring traffic manually is cumbersome. For the application environment to scale up or down dynamically, you must automate the processes of monitoring traffic and of scaling resources up and down whenever necessary.

Integrated with the GCP Autoscaling service, the NetScaler VPX instance provides the following advantages:

- **Load balance and management**: Auto configures servers to scale up and scale down, depending on demand. The VPX instance auto detects managed instance groups in the back-end subnet and allows you to select the managed instance groups to balance the load. The virtual and subnet IP addresses are auto configured on the VPX instance.
- **High availability**: Detects managed instance groups that span multiple zones and load-balance servers.
- **Better network availability**: The VPX instance supports:

- Back-end servers on same placement groups
- Back-end servers on different zones

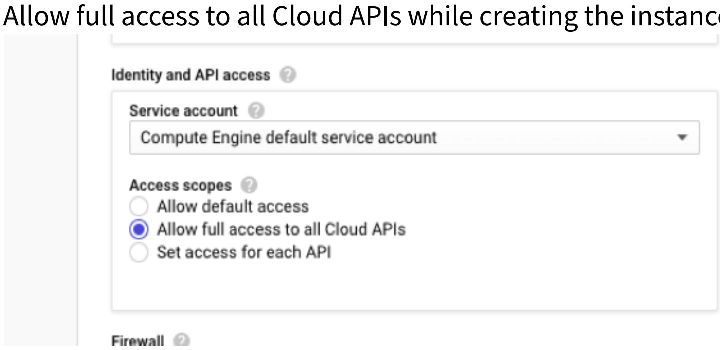
This diagram illustrates how the GCP Autoscaling service works in a NetScaler VPX instance acting as the load balancing virtual server.



Before you begin

Before you start using Autoscaling with your NetScaler VPX instance, you must complete the following tasks.

- Create a NetScaler VPX instance on GCP according to your requirement.
  - For more information about how to create a NetScaler VPX instance, see Deploy a NetScaler VPX instance on the Google Cloud Platform.
  - For more information about how to deploy VPX instances in HA mode, see Deploy a VPX high-availability pair on the Google Cloud Platform.
- Enable **Cloud Resource Manager API** for your GCP project.
- Allow full access to all Cloud APIs while creating the instances.



- Ensure your GCP service account has the following IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [

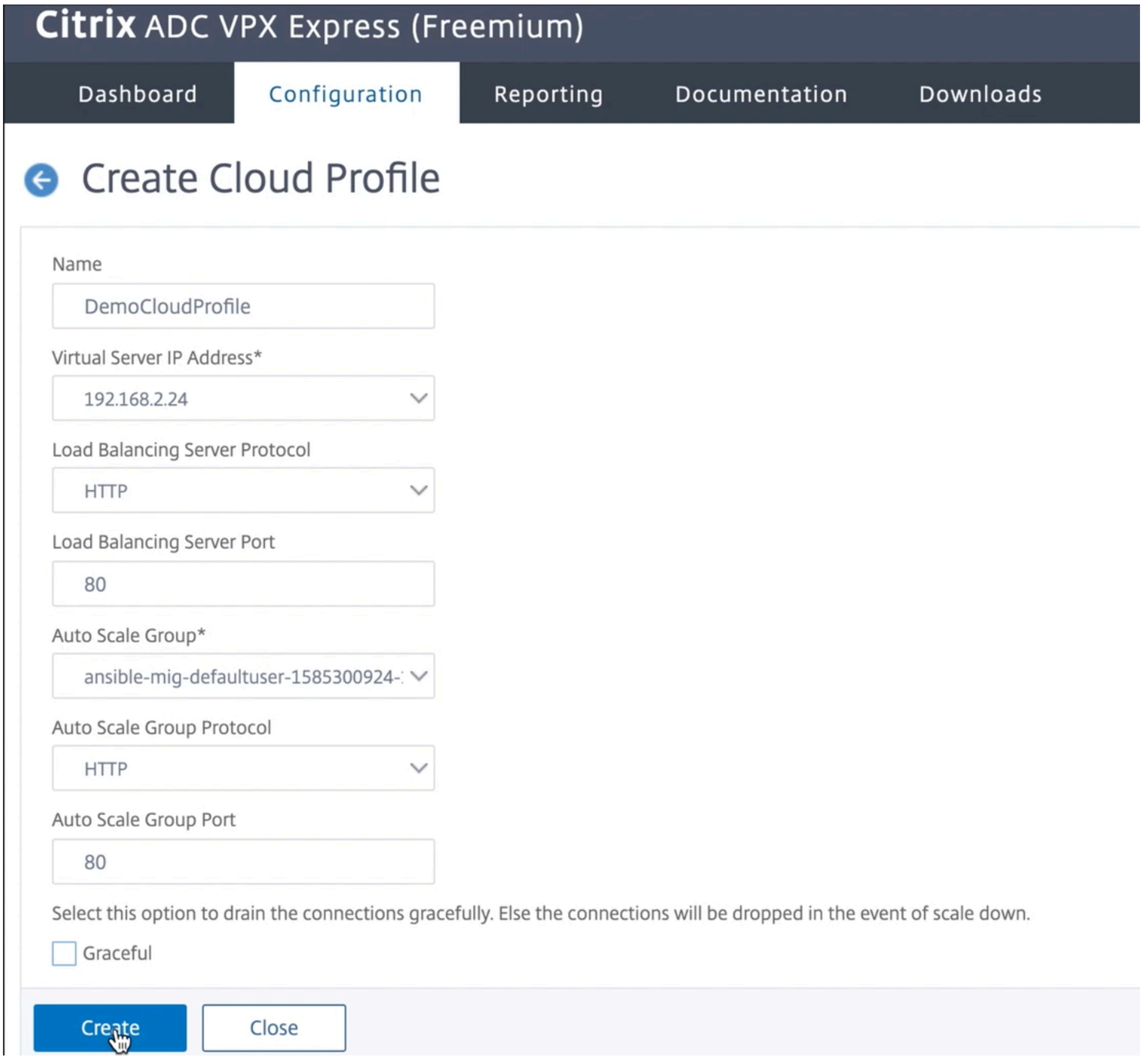
"compute.instances.get",
"compute.zones.list",
"compute.instanceGroupManagers.list",
"compute.instanceGroupManagers.get"
]
```

- To set up Autoscaling, ensure the following are configured:
  - Instance template
  - Managed Instance group
  - Autoscaling policy

Add the GCP Autoscaling service to a NetScaler VPX instance

You can add the Autoscaling service to a VPX instance with a single click by using the GUI. Complete these steps to add the Autoscaling service to the VPX instance:

1. Log on to the VPX instance by using your credentials for `nsroot`.
2. When you log on to the NetScaler VPX instance for the first time, you see the default Cloud Profile page. Select the GCP managed instance group from the drop-down menu and click **Create** to create a cloud profile.

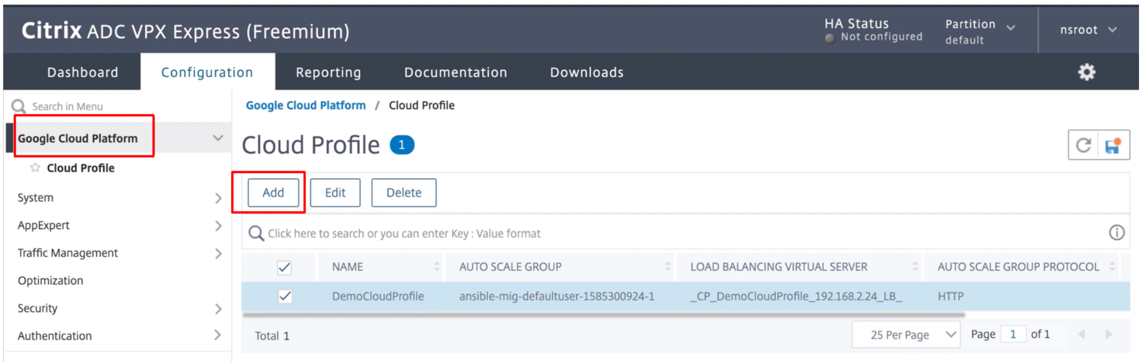


- The **Virtual Server IP Address** field is auto-populated from all the IP addresses associated with the instances.
- The **Autoscale Group** is prepopulated from the managed instance group configured on your GCP account.
- When selecting the **Autoscale Group Protocol** and **Autoscale Group Port**, ensure that your servers listen on the configured protocol and ports. Bind the correct monitor in the service group. By default, the TCP monitor is used.
- Clear the **Graceful** check box because it is not supported.

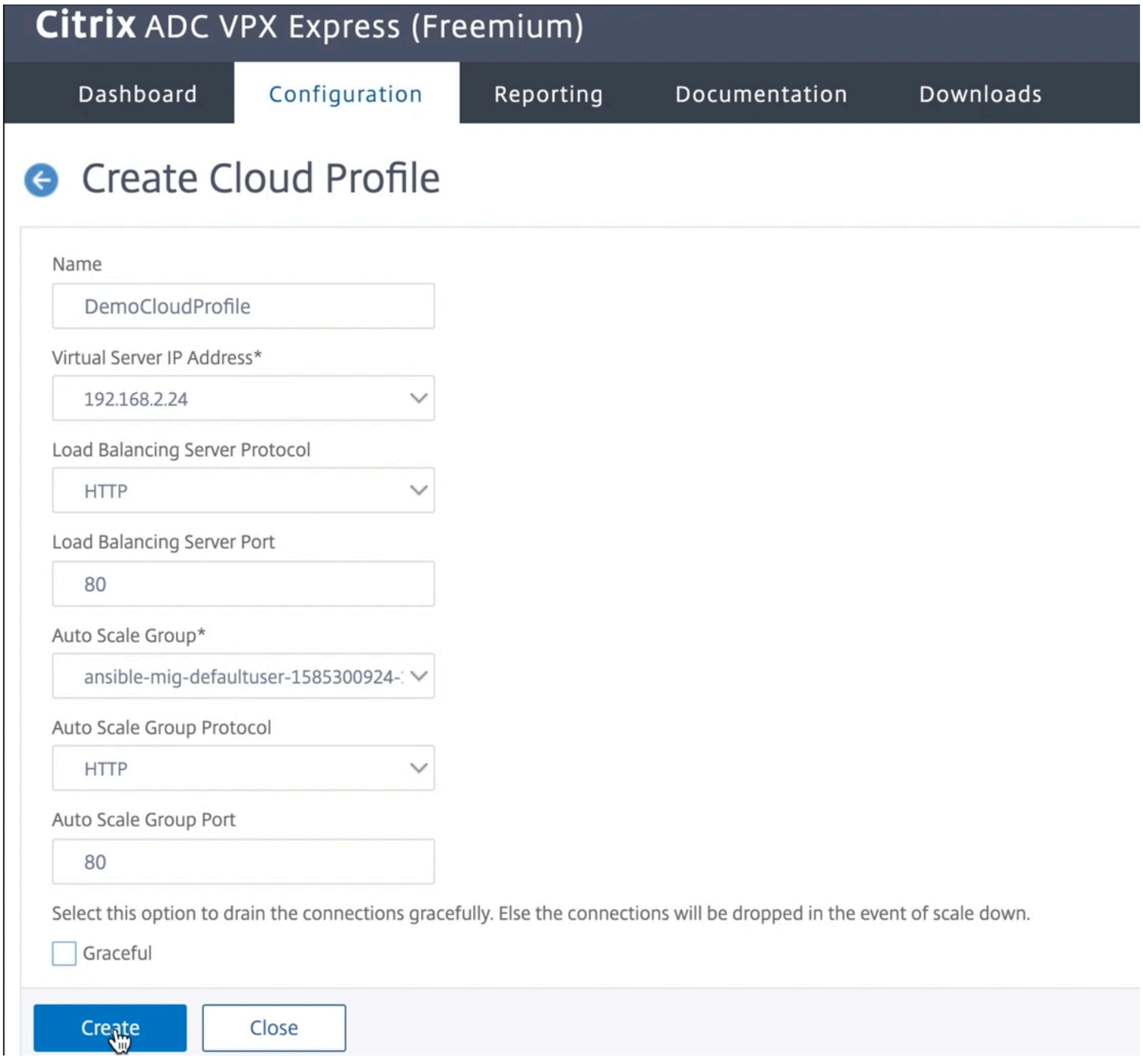
> **Note:**
>
> For SSL Protocol type Autoscaling, after you create the Cloud Profile, the load balance virtual server or service group is down because of a missing certificate. You can bind the certificate to the virtual server or service group manually.

3. After the first time logon if you want to create Cloud Profile, on the GUI go to **System > Google**

Cloud Platform > Cloud Profile and click **Add**.

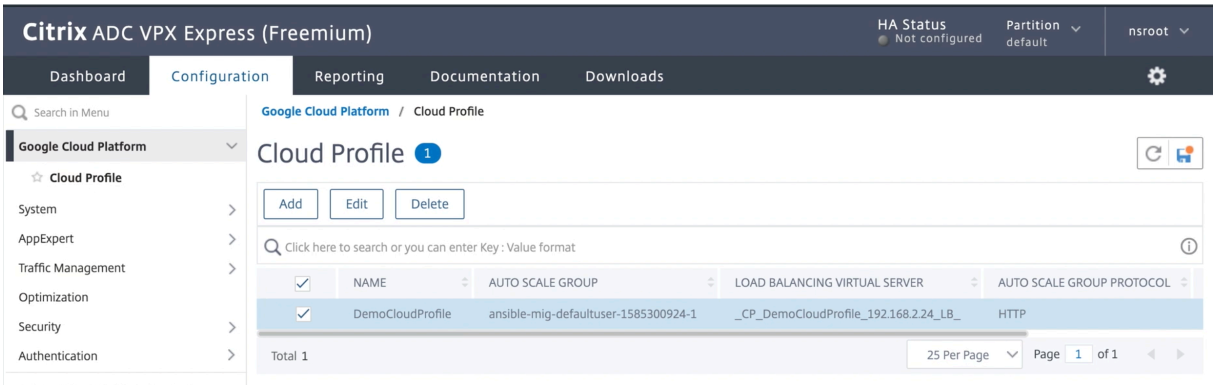


The **Create Cloud Profile** configuration page appears.



Cloud Profile creates a NetScaler load-balancing virtual server and a service group with members as the servers of the managed instance group. Your back-end servers must be reachable through the SNIP configured on the VPX instance.

> **Note:**
>
> From NetScaler release 13.1-42.x onwards, you can create different cloud profiles for different services (using different ports) with the same managed instance group in GCP. Thus, the NetScaler VPX instance supports multiple services with the same Autoscaling group in public cloud.



VIP scaling support for NetScaler VPX instance on GCP

A NetScaler appliance resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers configured on the appliance provide connection points that clients use to access the applications behind the appliance. The number of public virtual IP (VIP) addresses needed for a deployment varies on a case-by-case basis.

The GCP architecture restricts each interface on the instance to be connected to a different VPC. A VPC on GCP is a collection of subnets, and each subnet can span across zones of a region. In addition, GCP imposes the following limitation:

- There is a 1:1 mapping of number of public IP addresses to number of NICs. Only one public IP address can be assigned to a NIC.
- A maximum of only 8 NICs can be attached on a higher capacity instance type.

For example, an n1-standard-2 instance can have only 2 NICs, and the Public VIPs that can be added is limited to 2. For more information, see VPC resource quotas.

To achieve higher scales of public virtual IP addresses on a NetScaler VPX instance, you can configure the VIP addresses as part of the metadata of the instance. The NetScaler VPX instance internally uses forwarding rules provided by the GCP to achieve VIP scaling. The NetScaler VPX instance also provides high availability to the VIPs configured.
After you configure VIP addresses as part of the metadata, you can configure an LB virtual server using the same IP that is used to create the forwarding rules. Thus, we can use forwarding rules to miti-

gate the limitations we have w.r.t scale in using public VIP addresses on an NetScaler VPX instance on GCP.

For more information on forwarding rules, see Forwarding rules overview.

For more information on HA, see High Availability.

Points to note

- Google charges some additional cost for each virtual IP forwarding rule. The actual cost depends on the number of entries created. The associated cost can be found from the Google pricing documents.
- Forwarding rules are applicable only for public VIPs. You can use alias IP addresses when the deployment needs private IP addresses as VIPs.
- You can create forwarding rules only for the protocols, which need the LB virtual server. VIPs can be created, updated, or deleted on the fly. You can also add a new load balancing virtual server with the same VIP address but with a different protocol.

Before you start

- NetScaler VPX instance must be deployed on GCP.
- External IP address must be reserved. For more information, see Reserving a static external IP address.
- Ensure that your GCP service account has the following IAM permissions:

```
REQUIRED_IAM_PERMS = [
"compute.addresses.list",
"compute.addresses.get",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.instances.use",
"compute.subnetworks.use",
"compute.targetInstances.create"
"compute.targetInstances.get"
"compute.targetInstances.use",
]
```

- Enable **Cloud Resource Manager API** for your GCP project.
- If you use VIP scaling on a standalone VPX instance, ensure that your GCP service account has the following IAM permissions:


```
REQUIRED_IAM_PERMS = [
"compute.addresses.list",
"compute.addresses.get",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.instances.use",
"compute.subnetworks.use",
"compute.targetInstances.create",
"compute.targetInstances.list",
"compute.targetInstances.use",
]
```

- If you use VIP scaling in a high availability mode, ensure that your GCP service account has the following IAM permissions:

```
REQUIRED_IAM_PERMS = [
"compute.addresses.get",
"compute.addresses.list",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.use",
"compute.instances.get",
"compute.instances.list",
"compute.instances.setMetadata",
"compute.subnetworks.use",
"compute.targetInstances.create",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.zones.list",
]
```

> **Note:**
>
> In a high availability mode, if your service account does not have owner or editor roles, you must add the **Service Account User role** to your service account.

Configure external IP addresses for VIP scaling on NetScaler VPX instance

1. In the Google Cloud Console, navigate to the **VM Instances** page.
2. Create a new VM instance or use an existing instance.
3. Click the instance name. On the **VM instance details** page, click **Edit**.

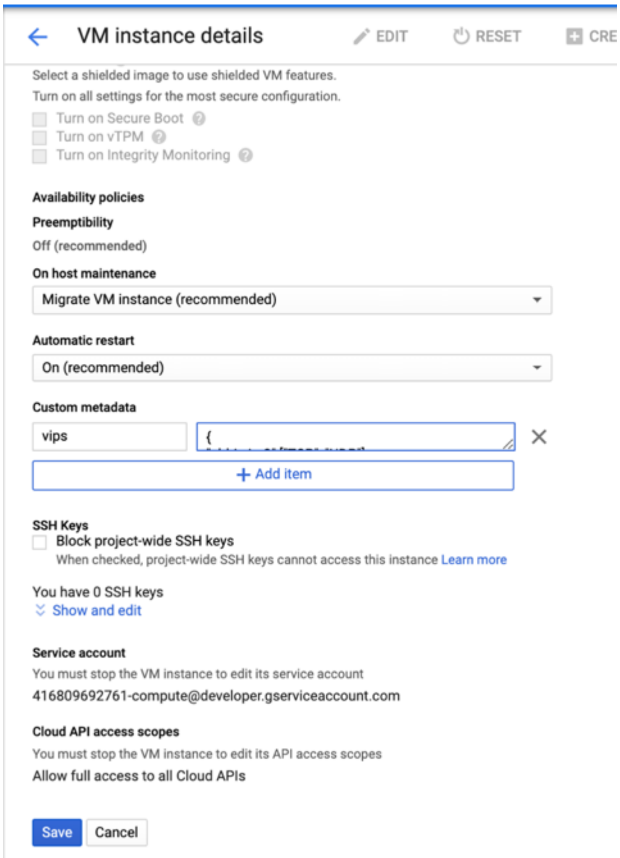
4. Update the **Custom metadata** by entering the following:

   - Key = vips
   - Value = Provide a value in the following JSON format:

     {
     "Name of external reserved IP": [list of protocols],
     }

   GCP supports the following protocols:

   - AH
   - ESP
   - ICMP
   - SCT
   - TCP
   - UDP

   For more information, see Custom metadata.

   Example for Custom metadata:

{
“external-ip1-name”:[“TCP”, “UDP”],
“external-ip2-name”:[“ICMP”, “AH”]
}

In this example, the NetScaler VPX instance internally creates one forwarding rule for each IP, protocol pair. The metadata entries are mapped to the forwarding rules. This example helps you understand how many forwarding rules are created for a metadata entry.

Four forwarding rules are created as follows:

a) external-ip1-name and TCP
b) external-ip1-name and UDP
c) external-ip2-name and ICMP
d) external-ip2-name and AH

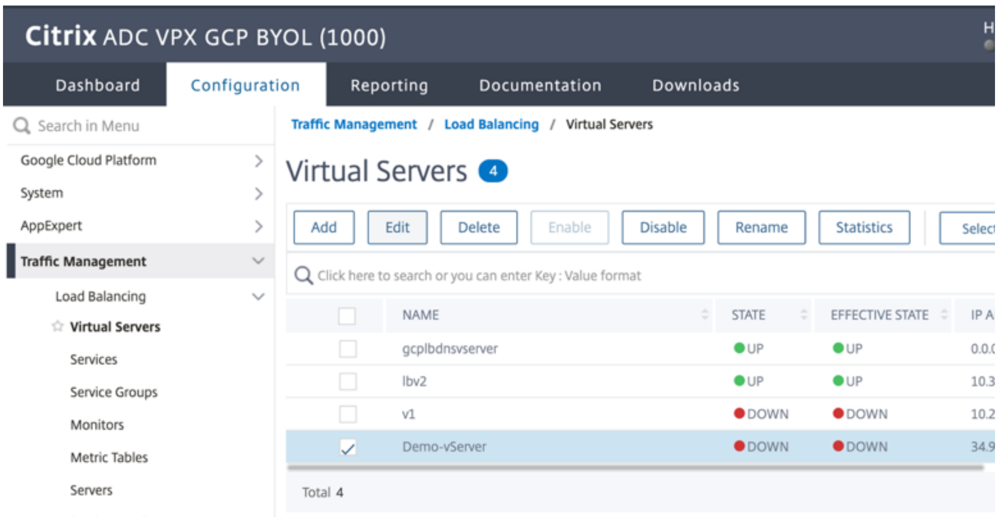
> **Note:**
>
> In HA mode, you must add custom metadata only on the primary instance. On failover, the custom metadata is synchronized to the new primary.

5. Click **Save**.

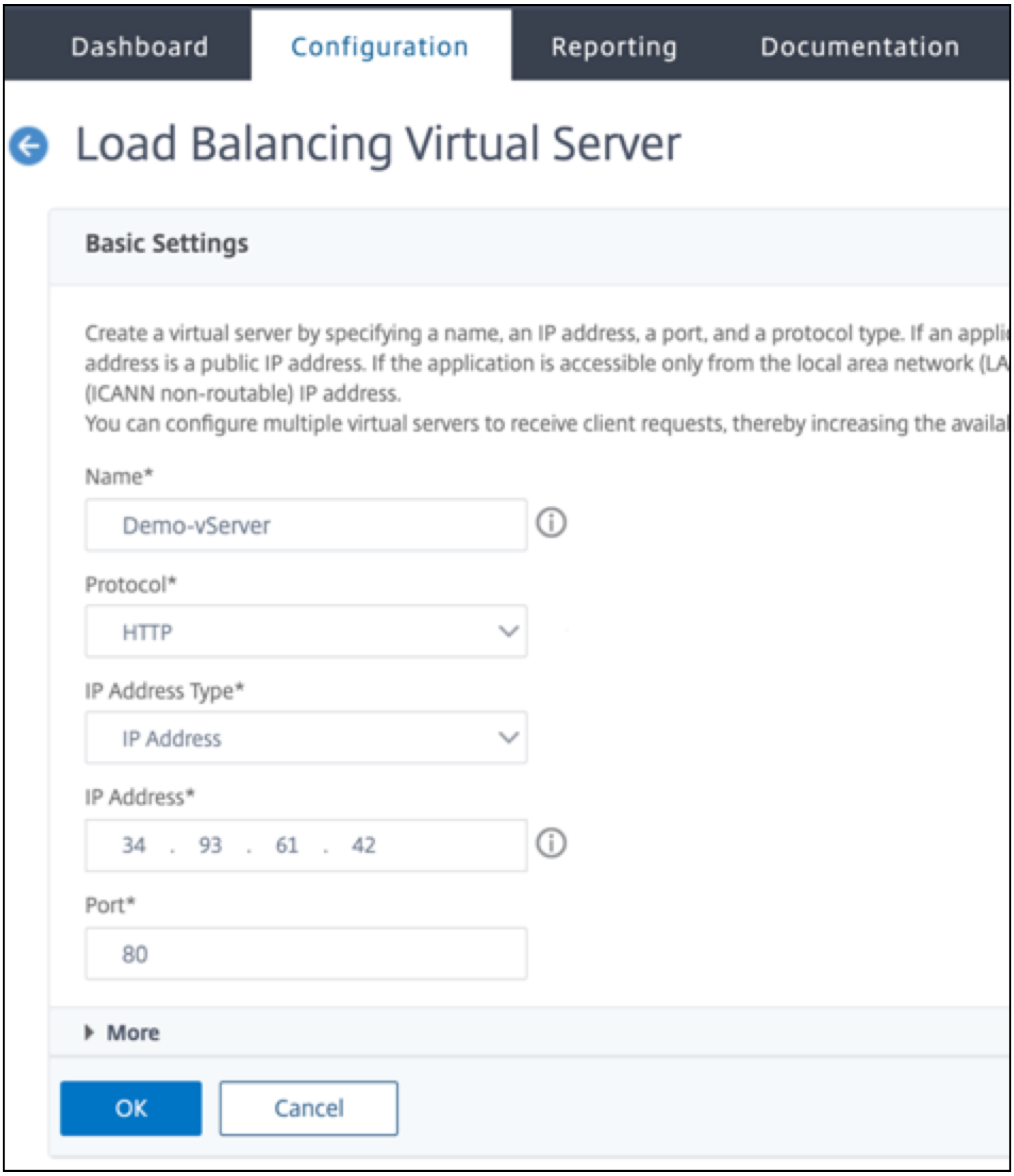
Setting up a load balancing virtual server with external IP address on a NetScaler VPX instance

Step 1. Add a load balancing virtual server.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.

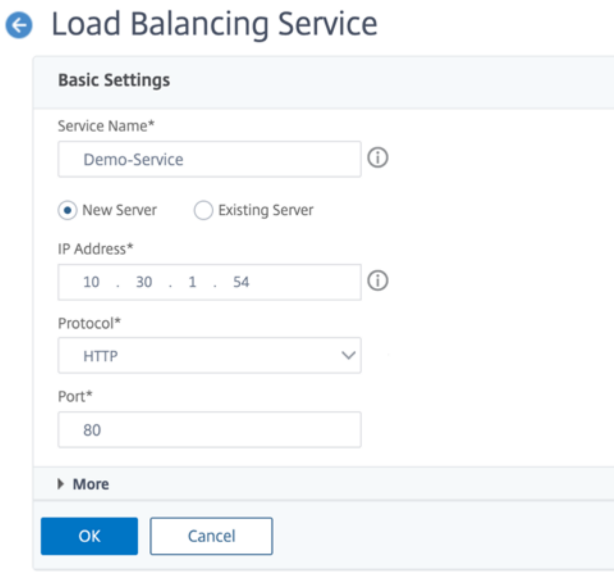


2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP Address (External IP address of the forwarding rule that is added as VIP on ADC) and Port, and click **OK**.



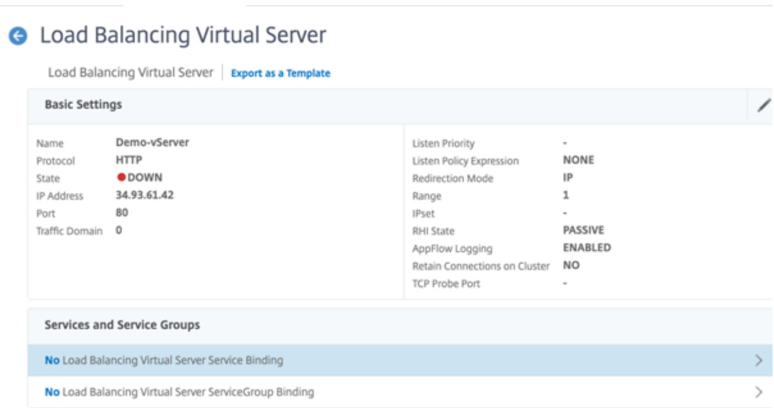
Step 2. Add a service or service group.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

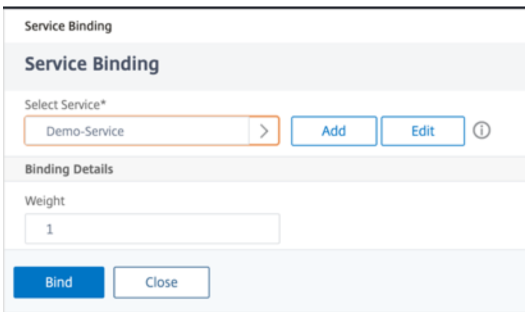


Step 3. Bind the service or service group to the load balancing virtual server.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 1**, and click **Edit**.
3. In the **Service and Service Groups** page, click **No Load Balancing Virtual Server Service Binding**.



4. Select the service configured in the **Step 3**, and click **Bind**.



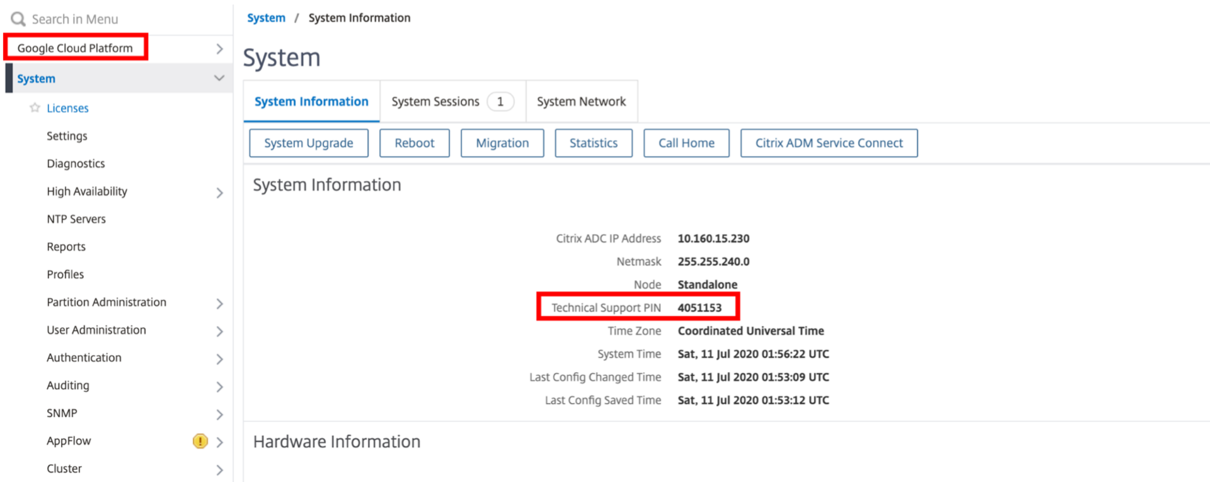
5. Save the configuration.

Troubleshoot a VPX instance on GCP

Google Cloud Platform (GCP) provides console access to a NetScaler VPX instance. You can debug only if the network is connected. To view an instance's System Log, access the console and check **System Log files**.

To file a support case, find your GCP account number and support PIN code, and call NetScaler support. You are asked to provide your name and email address. To find the support PIN, log on to the VPX GUI and navigate to the **System** page.

Here is an example of a system page showing the support PIN.



Jumbo frames on NetScaler VPX instances

NetScaler VPX appliances support receiving and transmitting jumbo frames containing up to 9216 bytes of IP data. Jumbo frames can transfer large files more efficiently than it is possible with the standard IP MTU size of 1500 bytes.

A NetScaler appliance can use jumbo frames in the following deployment scenarios:

- Jumbo to Jumbo. The appliance receives data as jumbo frames and sends it as jumbo frames.
- Non-Jumbo to Jumbo. The appliance receives data as regular frames and sends it as jumbo frames.
- Jumbo to Non-Jumbo. The appliance receives data as jumbo frames and sends it as regular frames.

For more information, see Configuring Jumbo Frames Support on a NetScaler Appliance.

Jumbo frames support is available on NetScaler VPX appliances running on the following virtualization platforms:

- VMware ESX
- Linux-KVM Platform
- Citrix XenServer
- Amazon Web Services (AWS)

Jumbo frames on VPX appliances work similar to jumbo frames on MPX appliances. For more information on Jumbo Frames and its use cases, see Configuring Jumbo Frames on MPX appliances. The use cases of jumbo frames on MPX appliances also apply to VPX appliances.

Configure jumbo frames for a VPX instance running on VMware ESX

Perform the following tasks to configure jumbo frames on a NetScaler VPX appliance running on the VMware ESX server:

1. Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501–9000. Use the CLI or GUI to set the MTU size. The NetScaler VPX appliances running on VMware ESX support receiving and transmitting jumbo frames containing up to only 9000 bytes of IP data.
2. Set the same MTU size on the corresponding physical interfaces of the VMware ESX server by using its management applications. For more information about setting the MTU size on the physical interfaces of VMware ESX, see http://vmware.com/.

Configure jumbo frames for a VPX instance running on Linux-KVM server

Perform the following tasks to configure jumbo frames on a NetScaler VPX appliance running on a Linux-KVM Server:

1. Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501–9216. Use the NetScaler VPX CLI or GUI to set the MTU size.
2. Set the same MTU size on the corresponding physical interfaces of a Linux-KVM Server by using its management applications. For more information about setting the MTU size on the physical interfaces of Linux-KVM, see http://www.linux-kvm.org/.

Configure jumbo frames for a VPX instance running on Citrix XenServer

Perform the following tasks to configure jumbo frames on a NetScaler VPX appliance running on Citrix XenServer:

1. Connect to the XenServer using XenCenter.
2. Shut down all the VPX instances that use the Networks for which the MTU must be changed.
3. On the **Networking** tab, select the network - network 0/1/2.
4. Select **Properties** and edit MTU.

After configuring the jumbo frames on the XenServer, you can configure the jumbo frames on the ADC appliance. For more information, see Configuring Jumbo Frames Support on a NetScaler Appliance.

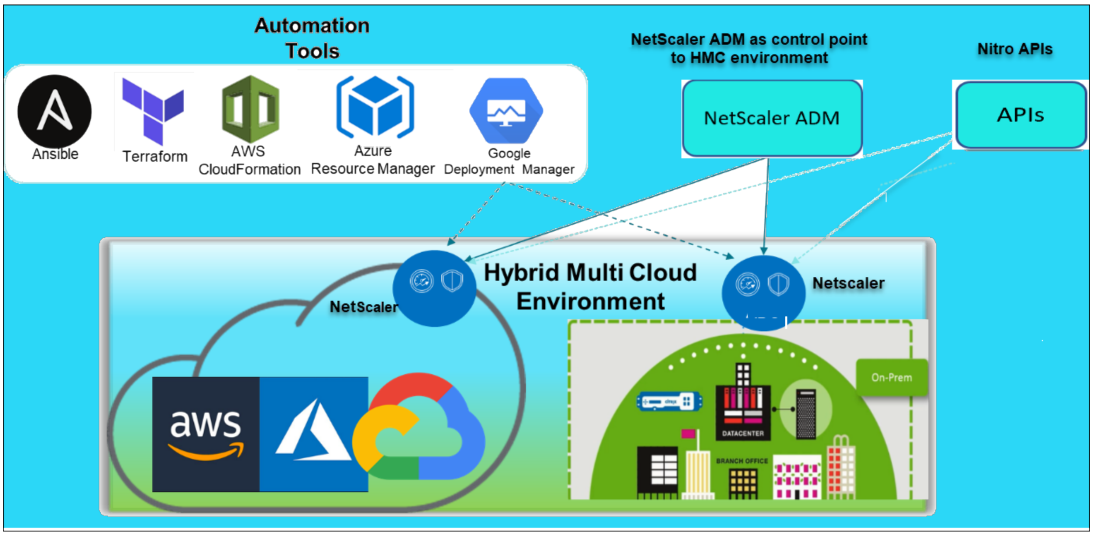
Configure jumbo frames for a VPX instance running on AWS

Host-level configuration is not required for VPX on Azure. To configure Jumbo Frames on VPX, follow the steps given in Configuring Jumbo Frames Support on a NetScaler Appliance.

Automate deployment and configurations of NetScaler

NetScaler provides multiple tools to automate your ADC deployments and configurations. This document provides a brief summary of various automation tools and references to various automation resources that you can use to manage ADC configurations.

The following illustration provides an overview of NetScaler automation in a hybrid multi cloud (HMC) environment.



Automate NetScaler using NetScaler ADM

NetScaler ADM acts as an automation control point to your distributed ADC infrastructure. The NetScaler ADM provides a comprehensive set of automation capabilities from provisioning ADC appliances to upgrading it. The following are the key automation features of ADM:

- Provisioning NetScaler VPX instances on AWS
- Provisioning NetScaler VPX instances on Azure
- StyleBooks
- Configuration jobs
- Configuration audit
- ADC upgrades
- SSL certificate management
- Integrations - GitHub, ServiceNow, Event notifications integrations

NetScaler ADM blogs and videos on automation

- Application migrations using StyleBooks
- Integrate ADC configurations with CI/CD using ADM StyleBooks
- Simplifying public cloud NetScaler deployments through ADM
- 10 ways NetScaler ADM service supports easier NetScaler upgrades

NetScaler ADM also provides APIs for its various capabilities that integrate NetScaler ADM and NetScaler as part of the overall IT automation. For more information, see NetScaler ADM Service APIs.

Automate NetScaler using Terraform

Terraform is a tool that takes infrastructure as code approach to provision and manage cloud, infrastructure, or service. NetScaler terraform resources are available in GitHub for use. Refer GitHub for detailed documentation and usage.

- NetScaler Terraform modules to configure ADC for various use cases such as Load Balancing and GSLB
- Terraform cloud scripts to deploy ADC in AWS
- Terraform cloud scripts to deploy ADC in Azure
- Terraform cloud scripts to deploy ADC in GCP
- Blue-green deployment using NetScaler VPX and Azure pipelines

Blogs and Videos on Terraform for ADC automation

- Automate your NetScaler deployments with Terraform
- Provision and configure ADC in HA setup in AWS using Terraform

Automate NetScaler using Consul-Terraform-Sync

NetScaler Consul-Terraform-Sync (CTS) module empowers application teams to automatically add or remove new instances of services to NetScaler. There is no need to raise manual tickets to IT admins or networking teams to make the necessary ADC configurations changes.

- NetScaler Consul-Terraform-Sync Module for Network Infrastructure Automation
- Citrix-HashiCorp joint webinar: Dynamic Networking with Consul-Terraform-Sync for Terraform Enterprise and NetScaler

Automate NetScaler using Ansible

Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code. NetScaler Ansible modules and sample playbooks can be found in GitHub for use. Refer GitHub for detailed documentation and usage.

- Ansible modules to configure ADC
- ADC Ansible modules documentation/reference guide
- Ansible modules for ADM

Citrix is a certified Ansible Automation Partner. Users having Red Hat Ansible Automation Platform subscription can access NetScaler Collections from Red Hat Automation Hub.

Terraform and Ansible automation blogs

- Citrix named HashiCorp Integration Partner of the Year
- Citrix is now a Certified Red Hat Ansible Automation Platform Partner
- Terraform and Ansible Automation for app delivery and security

Public cloud templates for ADC deployments

Public cloud templates simplify provisioning of your deployments in public clouds. Different NetScaler templates are available for various environments. For usage details, refer to respective GitHub repositories.

AWS CFTs:

- CFTs to provision NetScaler VPX on AWS

Azure Resource Manager (ARM) Templates:

- ARM templates to provision NetScaler VPX on Azure

Google Cloud Deployment Manager (GDM) Templates:

- GDM templates to provision NetScaler VPX on Google

Videos on Templates

- Deploy NetScaler HA in AWS using CloudFormation Template
- Deploy NetScaler HA across Availability Zones using AWS QuickStart
- NetScaler HA deployment in GCP using GDM templates

NITRO APIs

The NetScaler NITRO protocol allows you to programmatically configure and monitor the NetScaler appliance by using Representational State Transfer (REST) interfaces. Therefore, NITRO applications can be developed in any programming language. For applications that must be developed in Java or .NET or Python, NITRO APIs are exposed through relevant libraries that are packaged as separate Software Development Kits (SDKs).

- NITRO API documentation
- Sample ADC use case configuration using NITRO API

FAQs

The following section helps you to categorize the FAQs based on Citrix Application Delivery Controller (ADC) VPX.

- Feature and functionality
- Encryption
- Pricing and packaging
- [NetScaler VPX Express] (#citrix-adc-vpx-express-and-90-day-free-trial)
- Hypervisor
- Capacity planning or sizing
- System requirements
- Other technical FAQs

Feature and functionality

What is NetScaler VPX?

NetScaler VPX is a virtual ADC appliance that can be hosted on a Hypervisor installed on industry standard servers.

Does NetScaler VPX include all the web application optimization functionality as ADC appliances?

Yes. NetScaler VPX includes all load balancing, traffic management, application acceleration, application security (including NetScaler Gateway and Citrix Application Firewall), and offload functionality. For a complete overview of the NetScaler feature and functionality, see Application delivery your way.

Are there any limitations with Citrix Application Firewall when using it on NetScaler VPX?

Citrix Application Firewall on NetScaler VPX provides the same security protections as it does on NetScaler appliances. The performance or throughput of Citrix Application Firewall varies by platform.

Are there any differences between NetScaler Gateway on NetScaler VPX and NetScaler Gateway on NetScaler appliances?

Functionally, they are identical. NetScaler Gateway on NetScaler VPX supports all the NetScaler Gateway features available in NetScaler software release 9.1. However, because NetScaler appliances provide dedicated SSL acceleration hardware, it offers greater SSL VPN scalability than a NetScaler VPX instance.

Other than the obvious difference of being able to run on a Hypervisor, how does NetScaler VPX differ from NetScaler physical appliances?

There are two main areas where customers see differences in behavior. The first is NetScaler VPX cannot offer the same performance as many NetScaler appliances. The second is that while NetScaler appliances incorporate its own L2 networking functionality, NetScaler VPX relies upon the Hypervisor for its L2 networking services. Generally, it does not limit how the NetScaler VPX can be deployed. There can be certain L2 functionality that is configured on a physical NetScaler appliance must be configured on the underlying Hypervisor.

How does NetScaler VPX play a role in the Application Delivery market?

NetScaler VPX changes the game in the application delivery market in the following ways:

- By making a NetScaler appliance even more affordable, NetScaler VPX enables any IT organization to deploy a NetScaler appliance. It is not just for their most mission-critical web applications, but for all of their Web applications.
- NetScaler VPX allows customers to further converge networking and virtualization within their data centers. NetScaler VPX cannot only be used to optimize web applications hosted on virtualized servers. It also enables web application delivery itself to become a virtualized service that can be easily and rapidly deployed anywhere. IT organizations use the standard data center processes for tasks such as provisioning, automation, and charge-back for the web application delivery infrastructure.
- NetScaler VPX opens up new deployment architectures that are not practical if only physical appliances are used. NetScaler VPX and NetScaler MPX appliances can be used basis, tailored to the individual needs of each respective application to handle processor-intensive actions such as compression and application firewall inspection. At the data center edge, NetScaler MPX appliances handle high-volume network-wide tasks such as initial traffic distribution, SSL encryption or decryption, denial of service (DoS) attack prevention, and global load balancing. Pairing high-performance NetScaler MPX appliances with easy-to-deploy NetScaler VPX virtual appliance brings unparalleled flexibility and customization capabilities to modern, large-scale, data center environments while also reducing overall data center costs.

How does NetScaler VPX fit into our Citrix delivery center strategy?

With the availability of NetScaler VPX, the entire Citrix delivery center offering is available as a virtualized offering. The entire Citrix delivery center benefits from the powerful management, provisioning, monitoring, and reporting capabilities available in Citrix XenCenter. This can be deployed rapidly into almost any environment, and managed centrally from anywhere. With one integrated, virtualized application delivery infrastructure, organizations can deliver desktops, client-server applications, and Web applications.

Encryption

Does NetScaler VPX support SSL offload?

Yes. However, NetScaler VPX does all SSL processing in software, so NetScaler VPX does not offer the same SSL performance as NetScaler appliances. NetScaler VPX can support up to 750 new SSL transactions per second.

Does third-party SSL cards installed on the server hosting NetScaler VPX accelerate SSL encryption or decryption?

No. Supporting third-party SSL cards cannot associate the NetScaler VPX to specific hardware implementations. It greatly diminishes an organizations ability to flexibly host NetScaler VPX anywhere within the data center. NetScaler MPX appliances must be used when more SSL throughput than NetScaler VPX provides is required.

Does NetScaler VPX support the same encryption ciphers as physical NetScaler appliances?

VPX supports all encryption ciphers as physical NetScaler appliances, except the ECDSA.

What is the SSL transactions throughput of NetScaler VPX?

See NetScaler VPX data sheet for information on SSL transactions throughput.

Pricing and packaging

How is NetScaler VPX packaged?

NetScaler VPX selection is similar to the selection of NetScaler appliances. First, the customer selects the NetScaler edition based on its functionality requirements. Then, the customer selects the specific NetScaler VPX bandwidth tier based on their throughput requirements. NetScaler VPX is available in Standard, Advanced, and Premium Editions. NetScaler VPX offers from 10 Mbps (VPX 10) to 100 Gbps (VPX 100G). More details can be found in the NetScaler VPX data sheet.

Is NetScaler VPX priced the same for all Hypervisors?

Yes.

Are the same NetScaler SKUs used for VPX on all Hypervisors?

Yes.

Can a NetScaler VPX license be moved from one Hypervisor to another (For example from VMware to Hyper-V)?

Yes. NetScaler VPX licenses are independent of the underlying Hypervisor. If you decide to move the NetScaler VPX virtual machine from one Hypervisor to another, you do not have to get a new license. However, you might need to rehost the existing NetScaler VPX license.

Can NetScaler VPX instances be upgraded?

Yes. Both the throughput limits and NetScaler family edition can be upgraded. Upgrade SKUs for both types of upgrade are available.

If I want to deploy NetScaler VPX in a high availability pair, how many licenses do I need?

As with NetScaler physical appliances, a NetScaler high availability configuration requires two active instances. Therefore, the customer must purchase two licenses.

NetScaler VPX Express and 90-day free trial

Does NetScaler VPX Express include all NetScaler standard functionality? Does it include NetScaler Gateway and load balancing for Citrix Virtual Apps (formerly XenApp) Web Interface and XML broker?

Yes. NetScaler VPX Express includes full NetScaler Standard functionality. Starting from NetScaler release 12.0–56.20, Citrix modified the VPX express behavior.

Does NetScaler VPX Express include all NetScaler standard functionality? Does it include NetScaler Gateway and load balancing for Citrix Virtual Apps Web Interface and XML broker?

Starting from NetScaler release 12.0–56.20, VPX Express offers the NetScaler Standard Edition feature set, except Gateway functionality. Earlier to the 12.0–56.20 release, VPX expresses includes all features in the standard edition.

Does NetScaler VPX Express require a license?

With the new NetScaler VPX Express release (12.0–56.20 and onwards), VPX Express is free and requires no license files to install and comes with no commitment. If you have a VPX Express license already, then the prior VPX Express behavior is preserved. If the VPX Express *license file* is removed and the 12.0–56.20 and onwards release is used, the new VPX express behavior takes effect.

Does the NetScaler VPX Express license expire?

With the new VPX express, no. There is no license and no expiry date. If you have a VPX express license already, the license expires one year after download.

Does NetScaler VPX Express include the five free NetScaler Gateway concurrent licenses?

Yes, if you own a VPX express license.

Is there a limit to how many NetScaler VPX Expresses a customer can download?

Five.

Does NetScaler VPX Express support the same encryption ciphers as NetScaler MPX appliances?

For general availability, all the same strong encryption ciphers supported on NetScaler appliances are available on NetScaler VPX and NetScaler VPX Express. It is subjected to the same import or export regulations.

Can I file technical support cases for NetScaler VPX Express?

No. A retail NetScaler VPX license such as, VPX-10, VPX-200, VPX-1000, VPX- 3000 is required to file technical support cases. However, NetScaler VPX Express users are free to use both the NetScaler VPX Knowledge Center, and request help from the community using the Z discussion forums.

Can NetScaler VPX Express be upgraded to a retail version?

Yes. Simply purchase the retail NetScaler VPX license you need, and then apply the corresponding license to the NetScaler VPX Express instance.

Hypervisor

What VMware versions does NetScaler VPX support?

NetScaler VPX supports both VMware ESX and ESXi for versions 3.5 or later. For more information, see Support matrix and usage guidelines

For VMware, how many virtual network interfaces can you allocate to a VPX?

You can allocate up to 10 virtual network interfaces to a NetScaler VPX.

From vSphere, how can we access the NetScaler VPX command line?

The VMware vSphere client provides built-in access to the NetScaler VPX command line through a console tab. Also, you can use any SSH or Telnet client to access the command line. You can use the NSIP address of the NetScaler VPX in the SSH or Telnet client.

How can you access the NetScaler VPX GUI?

To access the NetScaler VPX GUI, type the NSIP of the NetScaler VPX, for example, `http://NSIP address` in the address field of any browser.

Can two NetScaler VPX instances installed on the same VMware ESX be configured in a high availability setup?

Yes, but it is not recommended. A hardware failure would affect both NetScaler VPX instances.

Can two NetScaler VPX instances running on two different VMware ESX systems be configured in a high availability setup?

Yes. It is recommended in a high availability setup.

For the VMware, are interface related events supported on NetScaler VPX?

No. Interface related events are not supported.

For the VMware, are tagged VLANs supported on NetScaler VPX?

Yes. NetScaler tagged VLANs are supported on NetScaler VPX from release 11.0 and higher. For more information, see NetScaler documentation.

For VMware, are link aggregation and LACP supported on NetScaler VPX?

No. Link Aggregation and LACP are not supported for NetScaler VPX. Link aggregation must be configured at the VMware level.

How do we access NetScaler VPX documentation?

The documentation is available from the NetScaler VPX GUI. After logging in, select the **Documentation** tab.

Capacity planning or sizing

What performance can I expect with NetScaler VPX?

NetScaler VPX offers good performance. See NetScaler VPX data sheet for a specific performance level achievable using NetScaler VPX.

Given that server CPU power varies, how can we estimate the maximum performance of a NetScaler instance?

Using a faster CPU can result in higher performance (up to the maximum allowed by the license), while using a slower CPU can certainly limit the performance.

Are NetScaler VPX bandwidth or throughput limits for inbound only traffic, or both inbound and outbound traffic?

NetScaler VPX bandwidth limits are enforced for traffic inbound to the NetScaler only, regardless of whether the request traffic or response traffic. It indicates that a NetScaler VPX-1000 (for example) can process both 1 Gbps of inbound traffic and 1 Gbps of outbound traffic simultaneously. Inbound and outbound traffic is not the same as request and response traffic. To the NetScaler, both traffic coming from endpoints (request traffic) and traffic coming from origin servers (response traffic) is "inbound" (that is, coming into the NetScaler).

Can multiple instances of NetScaler VPX be run on the same server?

Yes. However, ensure that the physical server has enough CPU and I/O capacity to support the total workload running on the host, or NetScaler VPX performance can be impacted.

If more than one instance of NetScaler VPX is running on a physical server, what is the minimum hardware requirement per NetScaler VPX instance?

Each NetScaler VPX instance must be allocated 2 GB of physical RAM, 20 GB of hard disk space, and 2 vCPUs. For critical deployments, we do not recommend 2 GB RAM for VPX because the system operates in a memory-constrained environment. This might lead to scale, performance, or stability related issues. Recommended is 4 GB RAM or 8 GB RAM.

> **Note:**
>
> The NetScaler VPX is a latency-sensitive, high-performance virtual appliance. To deliver its expected performance, the appliance requires vCPU reservation, memory reservation, vCPU pinning on the host. Also, hyper threading must be disabled on the host. If the host does not meet these requirements, issues such as high-availability failover, CPU spike within the VPX instance, sluggishness in accessing the VPX CLI, pit boss daemon crash, packet drops, and low throughput occur.
>
> Make sure that every VPX instance meets the predefined conditions.

Can I host NetScaler VPX and other applications on the same server?

Yes. For example, NetScaler VPX, Citrix Virtual Apps Web Interface and Citrix Virtual Apps XML Broker can all be virtualized and can run on the same server. For best performance, ensure that the physical host has enough CPU and I/O capacity to support all the running workloads.

Will adding CPU cores to a single NetScaler VPX instance increase the performance of that instance?

Yes, adding CPU cores can improve NetScaler VPX performance, provided the NetScaler VPX instance is licensed for the extra vCPUs. NetScaler VPX can support up to 20 vCPUs (for 41 Gbps - 100 Gbps performance), depending on the configuration and performance tier. More vCPUs can help increase throughput, especially in high-performance scenarios. However, the impact on performance also depends on factors like the network drivers (for example, PCI passthrough or SR-IOV) and the specific workload. For information on number of vCPUs supported for different VPX performance tiers, see NetScaler VPX data sheet.

Why NetScaler VPX looks like consuming more than 90% of the CPU even though it is idle?

It is normal behavior and NetScaler appliances exhibit the same behavior. To see the true extent of NetScaler VPX CPU utilization, use the stat CPU command in the NetScaler CLI, or view NetScaler VPX CPU utilization from the NetScaler GUI. The NetScaler packet processing engine is always "looking for work,"even when there is no work to be done. Therefore, it does everything to take control of the CPU and not release it. On a server installed with NetScaler VPX and nothing else, results in looking like (from the Hypervisor perspective) that NetScaler VPX is consuming the entire CPU. Looking at the CPU utilization from "inside NetScaler"(by using the CLI or the GUI) provides a picture of NetScaler VPX CPU capacity being used.

System requirements

What are the minimum hardware requirements for NetScaler VPX?

The following table explains the minimum hardware requirements for NetScaler VPX.

Type	Requirements
Processor	For the processor requirements of your VPX platform, refer to the Supported processors for NetScaler VPX table.
Memory	Minimum 2 GB. However, 4 GB is recommended.
Disk	Minimum 20 GB hard drive.
Hypervisor	Citrix Hypervisor 5.6 or later, VMware ESX/ESXi 3.5 or later, or Windows Server 2008 R2 with Hyper-V
Network Connectivity	100 Mbps minimum, but 1 Gbps is recommended.
NIC	Use a NIC that is compatible with your hypervisor. For more information, see Supported NICs for NetScaler VPX.

> **Note:**
>
> - For critical deployments, 4 GB memory is preferred for NetScaler VPX. With 2 GB memory, NetScaler VPX operates in a very memory-constrained environment. This might lead to scale, performance, or stability related issues.
> - From NetScaler 13.1 release onwards, the NetScaler VPX instance on VMware ESXi hypervisor supports AMD EPYC processors.

For more information on system requirements, see NetScaler VPX data sheet.

What is Intel VT-x?

These features, sometimes referred to as "hardware assist"or "virtualization assist,"trap sensitive or privileged CPU instructions run by the guest OS out to the Hypervisor. This simplifies hosting guest OSs (BSD for a NetScaler VPX) on the Hypervisor.

How common are VT-x?

Many servers have virtualization assistance features (such as VT-x or AMD-V) disabled by default in the BIOS settings. Before concluding that you cannot run NetScaler VPX, check the BIOS configuration. If virtualization support is disabled, you may need to enable it in the BIOS to ensure your server can properly run virtualized applications like NetScaler VPX.

Is there a hardware compatibility list (HCL) for NetScaler VPX?

As long as the server supports Intel VT-x, NetScaler VPX must run on any server compatible with the underlying Hypervisor. See the Hypervisor HCL for a comprehensive list of supported platforms.

What version of NetScaler OS is NetScaler VPX based on?

NetScaler VPX is based on NetScaler 9.1 or later releases.

Since NetScaler VPX runs on BSD, can it be run natively on a server with BSD Unix installed?

No. NetScaler VPX requires the Hypervisor to run. Detailed Hypervisor supports can be found in NetScaler VPX data sheet.

Other technical FAQs

Does link aggregation on a physical server with multiple NIC's work?

LACP is not supported. For the Citrix Hypervisor, Static link aggregation is supported and has limits of four channels and seven virtual interfaces. For VMware, static link aggregation is not supported within NetScaler VPX, but can be configured at the VMware level.

Is MAC based forwarding (MBF) supported on VPX? Is there any change from the NetScaler appliance implementation?

MBF is supported and it behaves the same way as with the NetScaler appliance. The Hypervisor basically switches all the packets received from NetScaler VPX to the outside and conversely.

How is the NetScaler VPX upgrade process carried out?

Upgrades are performed the same way as for NetScaler appliances: download a kernel file and use install ns or the upgrade utility in the GUI.

How are flash and disk space allocated? Can we change it?

A minimum of 2 GB memory must be allocated to each NetScaler VPX instance. The NetScaler VPX disk image is sized at 20 GB to accommodate serviceability needs, including space for storing up to 4 GB of core dumps, as well as log and trace files. While it would be possible to generate a smaller disk image, there are no plans to do this currently. `/flash` and `/var` are both in the same disk image. They're kept as separate file systems for compatibility purposes.

The following values represent the disk space allocated for specific directories on the NetScaler VPX instance:

- /flash = 965M
- /var = 14G

For detailed memory allocation recommendation, refer to NetScaler VPX data sheet.

Can we add a new hard drive to increase space on NetScaler VPX instance?

Yes. From NetScaler release 13.1 build 21.x onwards, you have the option to increase disk space on the NetScaler VPX instance by adding a second disk. When you attach the second disk, the "/var/crash" directory is automatically mounted on to this disk. The second disk is used for storing core files and logging. Existing directories that are used to store core files and log files continue to work as earlier.

> **Note:**
>
> Take external backup on downgrade of the NetScaler appliance to avoid loss of data.

For information on how to attach a new hard disk drive (HDD) to a NetScaler VPX instance on a cloud, see the following:

- Azure documentation

  > **Note:**
  >
  > To attach a secondary disk on VPX instances deployed on Azure, ensure that the Azure VM sizes have a local temporary disk. For more information, see Azure VM sizes with no local temporary disk.

- AWS documentation
- GCP documentation

> **Warning:**
>
> After you add a new HDD to VPX, some of the scripts that work on files, which are moved to the new HDD might fail under the following condition:

> If you use the "link"shell command to create hard links to the files, which were moved to a new HDD.
>
> All such commands must be replaced by "ln -s"to use a symbolic link. Also, modify the failing scripts accordingly.

What can we expect to regard the NetScaler VPX build numbering and interoperability with other builds?

NetScaler VPX has similar build numbering as the 9.1. Cl (classic) and 9.1. Nc (nCore) release, for instance 9.1_97.3.vpx, 9.1_97.3.nc, and 9.1_97.3.cl.

Can the NetScaler VPX be a part of a high availability setup with a NetScaler appliance?

Not a supported configuration.

Are all the interfaces visible in NetScaler VPX directly related to the number of interfaces on the Hypervisor?

No. You can add up to seven interfaces (10 for VMware) through the NetScaler VPX configuration utility with only one physical NIC on the Hypervisor.

Can Citrix Hypervisor XenMotion or VMware VMotion or Hyper-V live migration be used to move active instances of NetScaler VPX?

NetScaler VPX does not support XenMotion or Hyper-V live migration. VMotion is supported from the NetScaler 12.1 release onwards.