Citrix Intelligent Traffic Management

Contents

What's New

February 15, 2022

New Feature / Enhancement	Version
Alerts - This feature monitors performance issues or anomalies of your configured platforms from an end-user network across the globe.	2022.02.15
Local Persistence - This feature offers the capability for decision stickiness when it is enabled. The requests are identified using the IP subnet mask, the length of which is configurable. For example, when a client repeats a request to the same application within a certain period (Persistence TTL), then the original decision is served back.	2021.12.09
AWS ELB Connector - This new connector pulls `HealthyHostCount`, `UnHealthyHostCount` and `Load Balancer Capacity Units (LCUs)` metrics from AWS ELB via Fusion. It provides customers with an integrated load-balancing experience and visibility into Fusion metrics that are available in their Openmix applications.	2019.08.16
Change Platform Type (Private to Community): This new feature allows customers to change the current settings of their private platform or GSLB to reference the community platform instead. This feature is useful for customers whose private platforms are hosted in a public data center or cloud region.	2019.07.03

New Feature / Enhancement	Version
New Dashboard - The new ITM Dashboard is now operational, information-dense, customizable, and overall more useful than the previous version. In the new Dashboard, you can view Radar Sessions, Radar Performance, Openmix Traffic Management Decisions, and Sonar Monitoring Status charts. You can create multiple dashboards, each tailored to a view that you care about. You can also choose to make the ITM Visualizer or the Dashboard your default landing page.	2019.06.27
Fusion Quarantine: This feature quarantines a customer's failing Fusion data feed, if the feed fails or runs at a polling interval of less than 24 hours. Fusion applies the quarantine logic to stop these failing feeds from running to save resources (CPU/Memory) and avoid impact on other good or valid Fusion data feeds.	2019.06.19
Enable/Disable Platforms for Openmix - A platform can now be enabled or disabled for Openmix by switching the **Openmix Enabled** button on or off in Platform Settings. If a particular platform is disabled for Openmix, that platform is not considered in Openmix decisions.	2019.04.09

New Feature / Enhancement	Version
Platform Geo - This feature allows customers to view and manage the geo location assigned to a platform. By default, there is no **Geo** location assigned to private platforms. When a user creates a private platform and configures a Radar probe, we use the probe URL to geo locate the platform. Alternatively, the user can assign a Geo manually without relying on the Radar URL path. For GSLB and F5 config imports, we geo locate the public IP and use that as the **Geo** of the platform. Community platforms by default inherit the original location of the platform.	2019.04.09
Visualizer: Drill down to the state-level: Active alerts with information about the performance and availability of Clouds, Data Centers, CDNs, and other services. These alerts are measured and viewed at the state level with-in the United States.	2019.04.01
Visualizer: F5 and GSLB imports - F5 and GSLB imports: You can now import a platform via a GSLB or F5 config. The basic site information (IP and name) is imported as ITM platforms. ITM geo-locates the site and allows the platform to be displayed on the Visualizer for performance analysis.	2019.03.29
G-Core Purge Adapter -The G-Core CDN purge adapter is now added to the list of adapters that ITM supports to run purges.	2019.03.29
Radar DSA 3 for all community providers – To continuously improve the Radar community and the accuracy of our benchmarks, we recently released a new Dynamic Content Benchmark. This new benchmark has a dynamic HTML page and a signature with which the measurement can be verified.	2019.03.21

New Feature / Enhancement	Version
Visualizer – The ITM visualizer is an intuitive and intelligent tool that enables you to monitor and analyze the global performance of ISPs and services. The ITM Visualizer UI provides active alerts with information on the performance and availability of Clouds, Data Centers, CDNs, and other services. ITM community measures these alerts across the globe. ITM Radar collects billions of measurements from real users across the globe via the Radar community. It uses a crowd sourcing model to measure these alerts.	2019.03.08
Guided Tours (walk-throughs) for the Visualizer and Openmix are now available in the **ITM Demo Portal**. The Demo Portal can be accessed via the help icon within the ITM portal. At the Bottom right corner of the **Demo Portal** you see an icon that launches the guided tours.	2019.03.08

Third Party Notifications

April 13, 2020

Citrix Intelligent Traffic Management Third Party Notifications (PDF)

Glossary

April 13, 2020

Term	Description
Application	An Openmix application is a specification of load-balancing logic that can be configured within the portal. The application will be processed for each request made to Openmix and a routing decision will be made based on the specified logic. Applications can be used for one or multiple types of content. A customer may have one application for one type of content that has high business value and a different application for content that has a lesser value that must be routed differently. For example, the customer may have one application for content shown to all users which focuses on routing to the fastest provider regardless of cost. The customer may also have another application for rarely shown content that focuses on cost optimization between providers for lower value content. In the above scenario, the customer would have two Openmix applications.
Community Measurements	Community measurements are sourced through a crowd sourcing model providing the customer a view of a vendor performance and availability at a geographic and logical level globally. Community Measurements are available for free to participating community members (installation of the JavaScript tag required). Access to community data for non-contributing (i.e.non JS integrating) organizations is a billed item.
Decision	An Openmix decision is specified as a single request to one of Citrix's load-balancers. For DNS, it is a single DNS request to the DNS load-balancers.For HTTP, it is a GET or HEAD request to the Openmix HTTP endpoint.

Term	Description
Measurement	A Measurement pertains to Radar and the Collection of data from end users on the performance of an application of service. For Community Measurements see Community Measurements.
Platform	A platform is a CDN, Cloud, Data Center, or Other End-Point that the customer wants to either Monitor within Radar or use within the Openmix Application.
Private Measurement	Radar Private Measurements are where measurements or telemetry (in the case of streaming) is fed back about the end-users experience that is not shared with the community. This can apply where a customer is looking to measure: + Their own data center architecture/s + Using their own test-object or page + Using their own contract with a vendor + Audio/Video end-user quality of Experience

Radar Data Definitions

April 13, 2020

Benchmark partners and Radar community members who have deployed the Radar tag can optionally be given access to their Radar measurements. In the case of benchmark partners, we share measurements taken of that partner regardless of the page on which the Radar tag was deployed, or when the measurement was taken. Community members can see all measurements, taken by their web visitors, regardless of which benchmark partner is being measured.

Customer Radar Data Share

Radar Tag deployers can optionally access a subset of the fields we receive from the Radar client when a Radar measurement is taken on their website. User IP addresses are anonymized before reports are generated. For log descriptions, refer to the Netscope (NEM) documentation.

Raw Radar Measurements

Raw Radar Measurements contain a subset of the fields we receive from the Radar client when a Radar measurement is made. User IP addresses are anonymized before reports are generated.

The reports can be made available daily or in real-time that deliver measurement data in under 5 minutes.

The files can be TAB delimited, CSV, or JSON format. For log descriptions and reports, refer to the Netscope documentation.

Autonomous System Numbers

https://s3-eu-west-1.amazonaws.com/community-radar/ref/asns.json.gz

Community (Public) Provider IDs

https://s3-eu-west-1.amazonaws.com/community-radar/ref/providers.json.gz

Probe Types (Measurement Types)

https://s3-eu-west-1.amazonaws.com/community-radar/ref/probetypes.json.gz

---

Response Codes

Code	Module	Description	Value
0	All	Success	Measurement Value
1	Remote Probing	HTTP request timeout	0
2	Remote Probing	RTMP connect failed	0
3	Remote Probing	RTMP stream not found	0
4	Remote Probing	HTTP invalid file	0
5	Navigation Timing	Navigation Timing API not supported	0

Market Codes

Code	Name	ISO Abbreviation
0	Unknown	XX
1	North America	NA
2	Oceania	OC
3	Europe	EU
4	Asia	AS
5	Africa	AF
6	South America	SA

Country Codes

Based on ISO 3166 -1 Alpha 2

https://s3-eu-west-1.amazonaws.com/community-radar/ref/countries.json.gz

Region Codes

There are no ISO standards for regions that we are aware of. Also, our GEO provider provides regions for only a small subset of countries. Per their docs, the goal of "regions" is to subdivide certain countries into areas larger than states. For example "US - Southwest"

To start, we are providing our own numeric "region IDs" and a mapping: https://s3-eu-west-1.amazonaws.com/community-radar/ref/regions.json.gz

> **NOTE:** We reserve the right to change the format of that file. Any code created to load in those mappings must be created with this in mind. Long term there will be an API call to download these mappings.

State Codes

There is an ISO standard for states 3166-2. We are evaluating if this standard meets our needs. So start, we are using our own numeric to string mappings. Similar to region, the format may change https://s3-eu-west-1.amazonaws.com/community-radar/ref/states.json.gz

City Codes

We are using our own numeric to string mappings. Similar to region, the format may change and we may eventually provide these mappings as an API call. https://s3-eu-west-1.amazonaws.com/community-radar/ref/cities.json.gz

Visualizer

February 15, 2022

Introduction

The ITM visualizer is an intuitive and intelligent tool that enables you to monitor and analyze the global performance of ISPs and services. The ITM Visualizer UI provides active alerts with information on the performance and availability of Clouds, Data Centers, CDNs, and other services. ITM community measures these alerts across the globe. ITM Radar collects billions of measurements from real users across the globe via the Radar community. It uses a crowd sourcing model to measure these alerts.

For a new user, the visualizer page opens with all available community alerts on the map. ITM Radar measures performance anomalies and generates alerts across almost every network, and in every location across the globe.

The four tiles over the Visualizer map show the following data.

Active Radar Alerts

Active Radar alerts are current and ongoing.

Radar Alerts

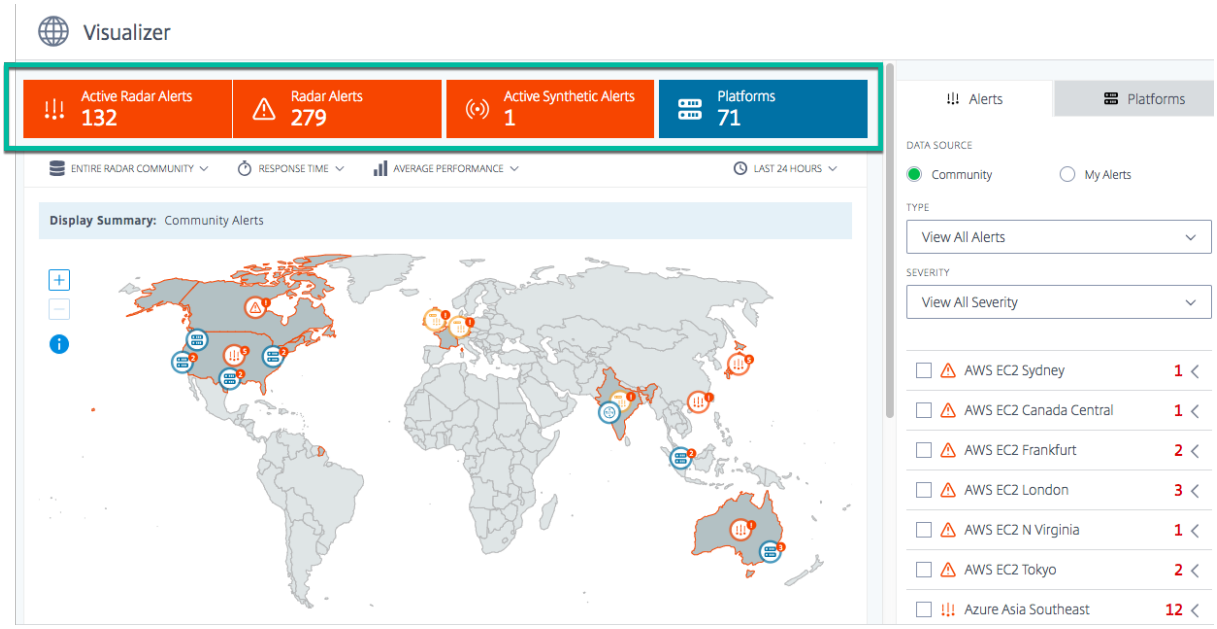
Active Radar alerts are current and ongoing. By default, this tile displays all alerts from the last 24 hours but changes depending on the time period a user selects.

Active Synthetic Alerts

These alerts occur real time. Sonar, our synthetic monitoring system that measures the global availability of a service or data center, generates these alerts.

Platforms

The number of platforms configured in the customer account.



Viewing Options

You can view alerts and platforms on the map using the following criteria:

Entire Radar Community or Only Your Visitors

Choose **Radar Community** to view the performance of platforms across the Radar community. Or alternatively, to view the performance for just your visitors via your private platforms choose **Only Your Visitors**.

Response Time or Availability

Click any platform on the map or in the list to view its performance based on **Availability** or **Response Time**.

Best Performance or Average Performance

Select **Average Performance** or best performance to view the average/best performance you would get for your platforms.

Average Performance is similar to doing a Round robin between your platforms and **Best Performance** is the performance we get by using ITM.

When you choose **Best Performance** you see the performance on the map based on the best performing platform. For example, if you are looking at the performance for a specific country, and you have

two platforms selected, **Best Performance** colors the country map based on which platform had the best performance between the two (highest availability or lowest response time) for that country.

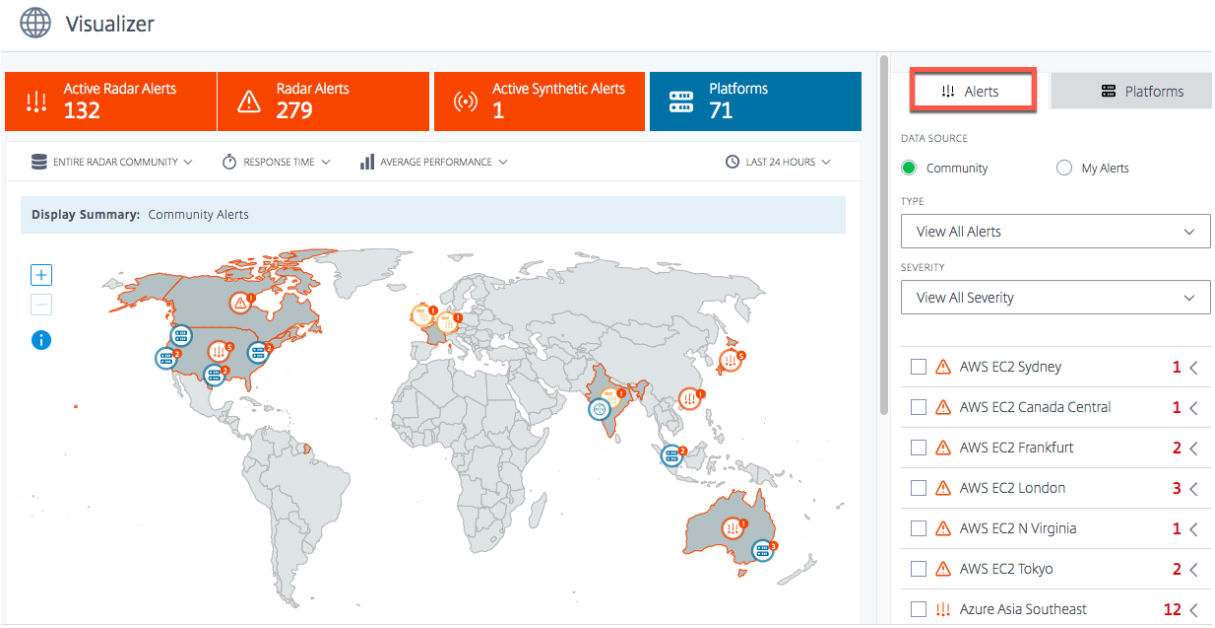
Alternatively, if you choose **Average Performance**, you see the performance on the map based on the average of all the selected platforms. It colors the country map with the average availability (or response time) of the two platforms.

Time Period

Alerts on the map can be generated with a time period of **Last 60 Minutes, Last 24 Hours, Last 48 Hours, Last 7 Days, Last 30 Days, or a custom range**. The default view is the Last 24 Hours. Every time you change the time period it refreshes the data on the map and shows you the triggered alerts for that time period.

Alerts

The **Alerts** tab is the default tab displayed when you land on the visualizer page. The default data source displayed for a new user with no alerts of their own is **Community.** This means that all alerts that you are viewing on the map as a new user are community alerts. Even if you've set up alerts, but don't have any active or ongoing alerts, your view defaults to community alerts. However, if you've set up your alerts, and have active ongoing alerts, then your default view is your own alerts. For more information about Alerts, see Alerts.



Community

Community alerts are performance issues or anomalies as seen by ITM Radar occurring across the ITM community. These alerts are measured via end user networks from across the globe. When you first open the **Visualizer** as a new user, you see all the community alerts on the map. Once you've set up your own alerts you see those alerts instead of the community alerts.

However, if you have private platforms and alerts set up, you see your own alerts as **My Alerts**, the default view.

My Alerts

These alerts are performance issues or anomalies of your private platforms. It uses end user networks across the globe to meassure these alerts.

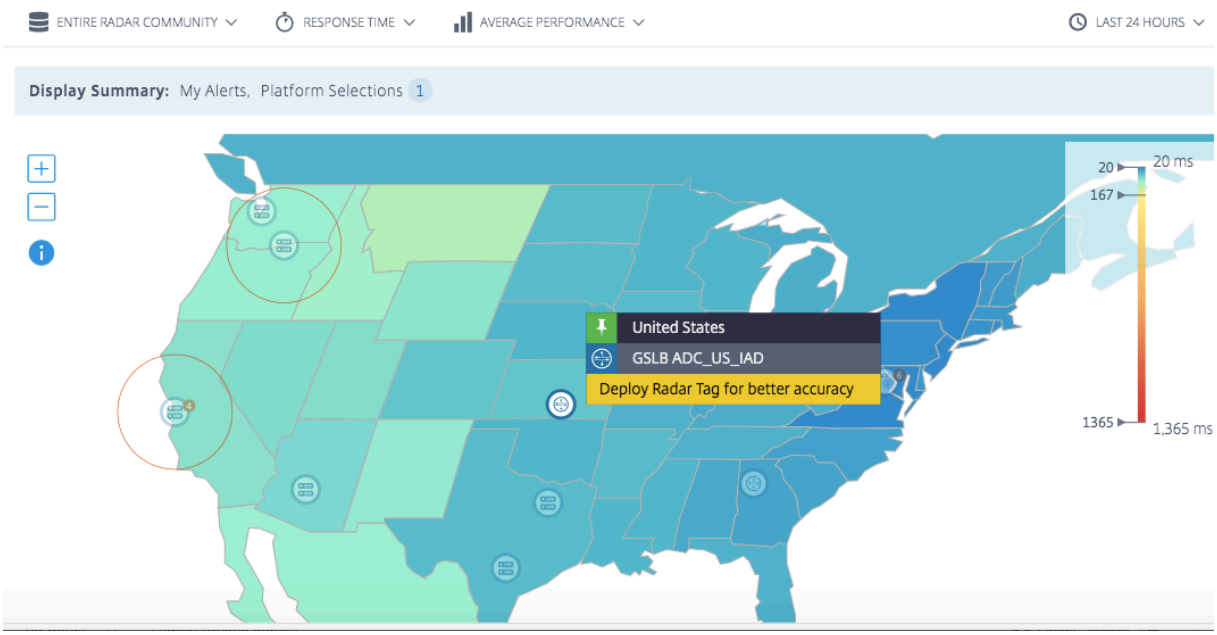
As a new user, if you don't see any alerts, that means you don't have any alerts set up. You can go to the **Alerts** page from the left sidebar to set up alerts for the performance of your platforms. But you would need to first set up your private platforms. To set up platforms you can either go to the **Platforms** page from the left sidebar or do it on the fly via the **Platforms** Tab.

Alert Details

You can hover over the alert on the map to view the country and services for which the alerts are being triggered. For more details on a specific alert,

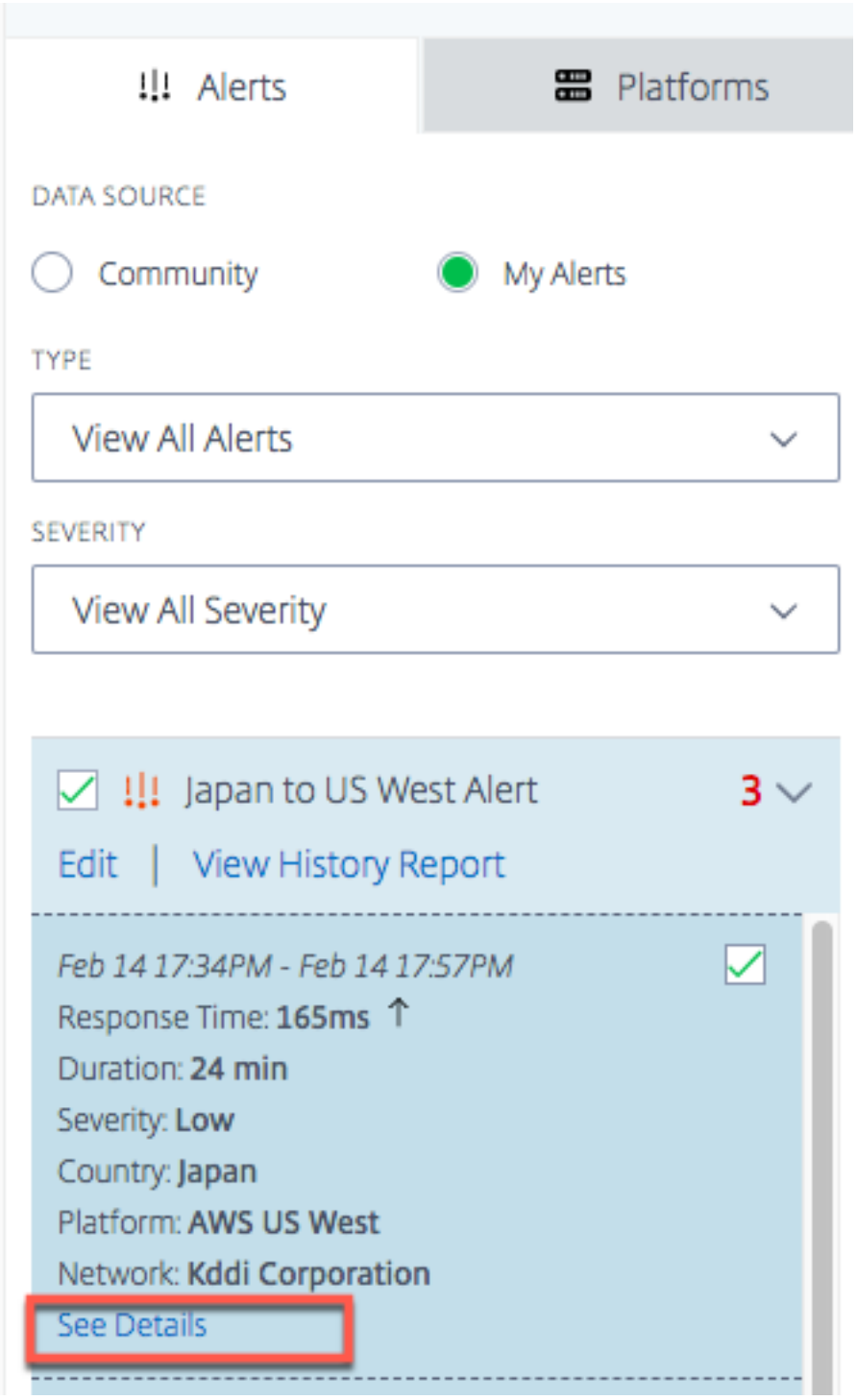
1. Click the alert icon on the map to check the box for the service triggers alert and highlights it on the list.
2. Click the arrow on the right side of the selected platform or service to display the details of the alert including,
    a) **Availability** or **Response Time** of the data source
    b) **Duration** of the alert
    c) **Severity** of the Alert
    d) **Country** of the network from which the issues are measured
    e) Name of the **Platform** for which the alert is triggered.
    f) Name of the **Network** from which the issues are measured.

State-level Alerts: Active alerts with information about the performance and availability of Clouds, Data Centers, CDNs, and other services. These alerts are measured and viewed at the state level within the United States.



To dive deeper into the details of the alert, click **See Details** to go to the **Alerts** page.

> **NOTE:** You can view the **See Details** link only for your own alerts.



Alert Type

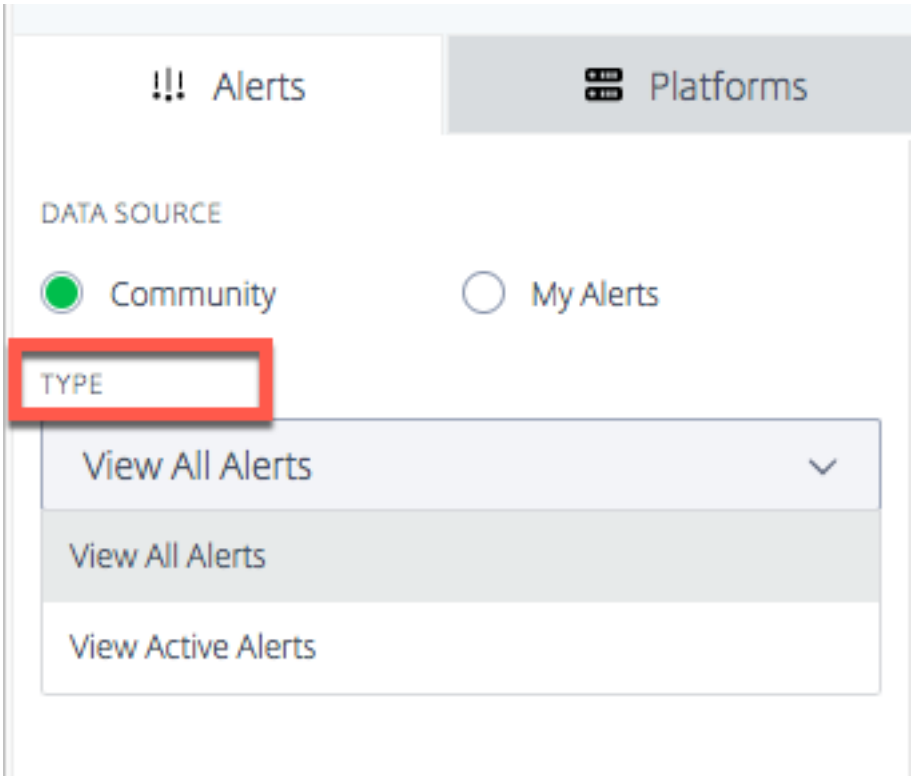
The **Type** menu allows you to view the following type of alerts.

All Alerts

All alerts include active and historical alerts. Historical alerts are alerts that originated later in the selected time period.

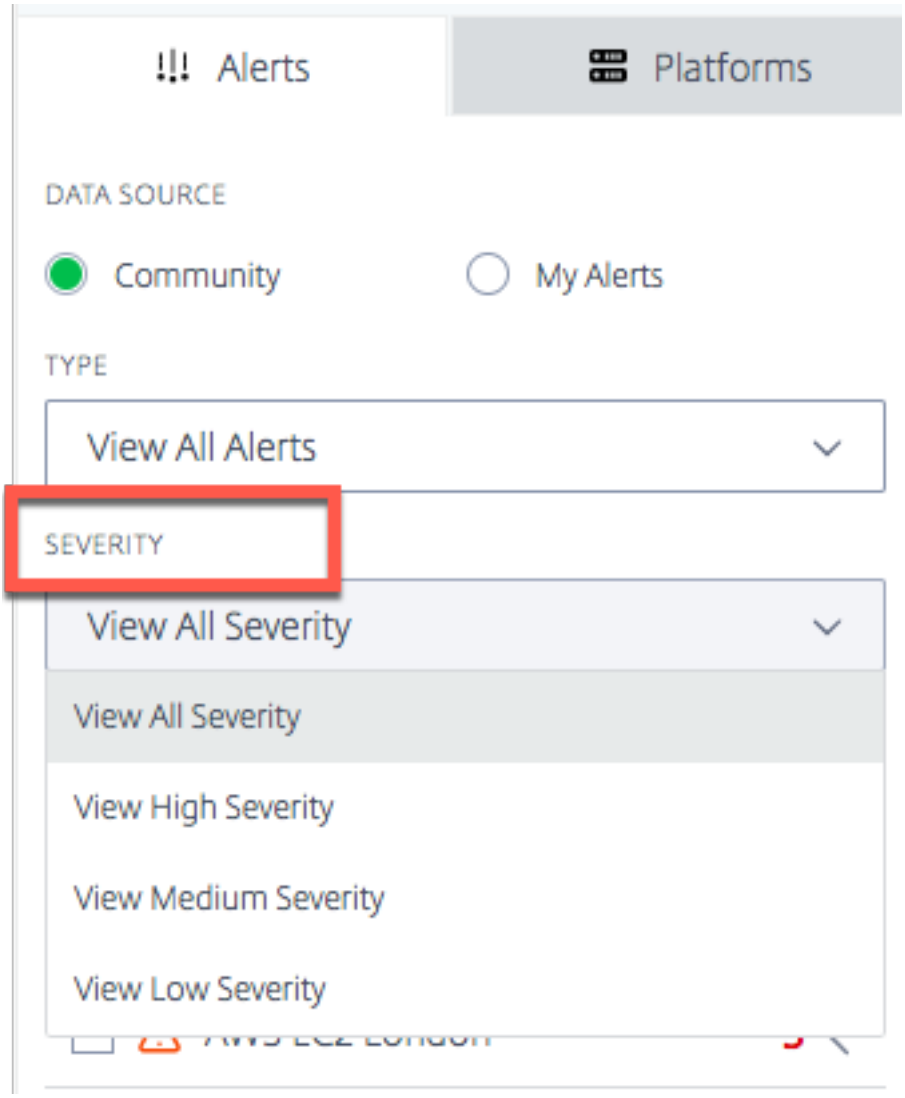
Active Alerts

Active alerts include alerts that are ongoing. They are valid and current of the user specified time period.



Alert Severity

Alerts can be filtered based on **High, Medium**, and **Low** severity. **All Severity** is the default display.



Severity Logic

For Availability:

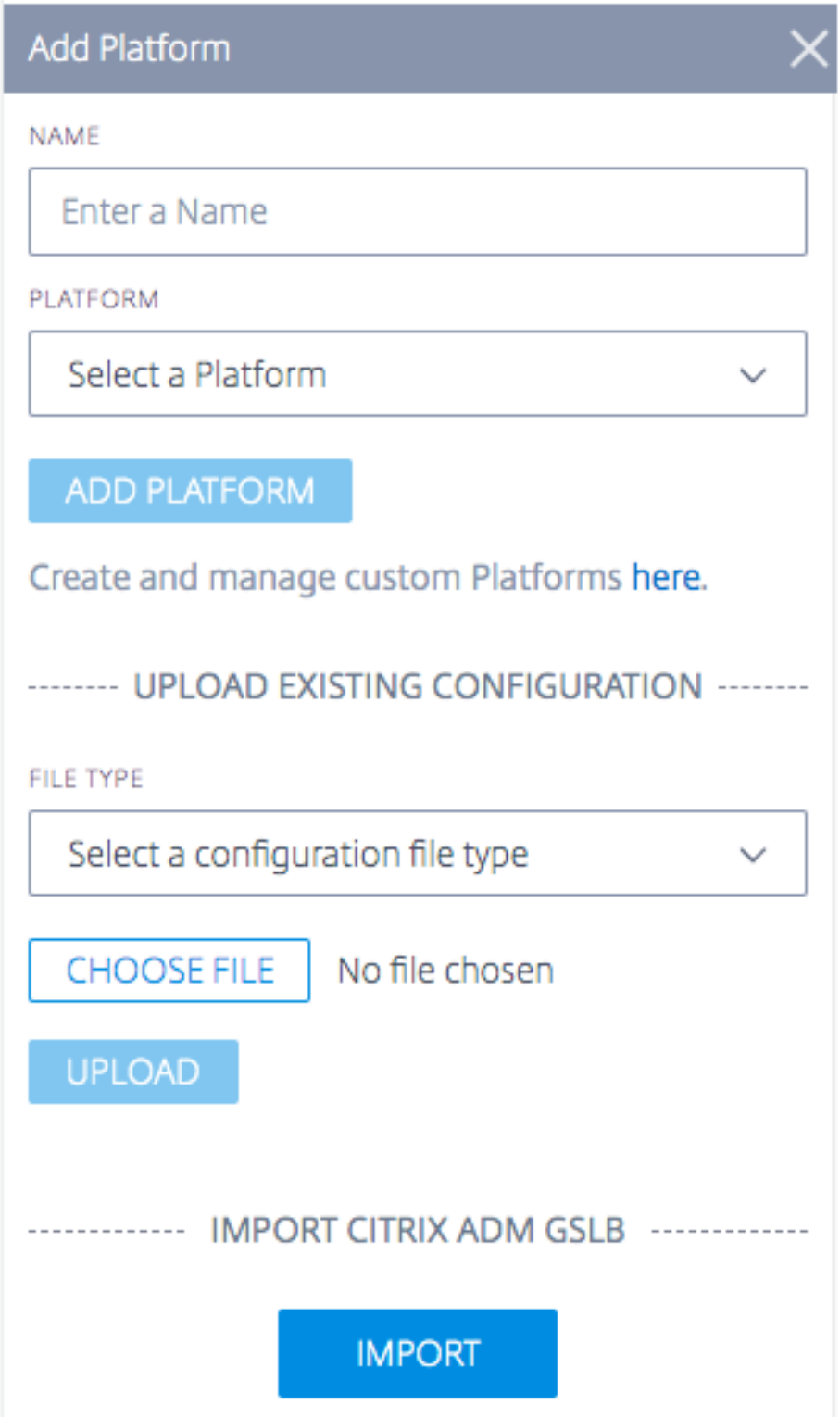
- If more than 50% under threshold -> Severity is **High**
- If more than 25% but less than 50% under threshold -> Severity is **Medium**
- If less than 25% under threshold -> Severity is **Low**

For Response Time:

- If more than 200% over threshold -> Severity is **High**
- If more than 100% over threshold but less than 200% -> Severity is **Medium**
- If less than 100% over threshold -> Severity is **Low**

Platforms

When you select the **Platforms** tab, you see the list of the platforms that you added. However, if you're a new user and haven't set up any platforms yet, you can either add a community platform here on-the-fly, or set up a private platform by clicking the **Create and manage custom Platforms here** link.



Add a Community Platform

1. To add a community platform, click the + icon next to the **Add Platform** bar.

2. Give a name for the platform and select the platform from the list of community platforms in the **Platform** menu.
3. Click **Add Platform**.

Add a Custom/Private Platform

1. To add a private platform, click the + icon next to the **Add Platform** bar.
2. Click the **Create and manage custom Platforms here** link which takes you to the **Platforms** page, where you can add a new private platform. Alternatively, you can go to the **Platforms** page from the left sidebar.

Upload Existing Configuration: Citrix ADC and F5 BIG-IP DNS

This option allows you to choose a Citrix ADC or F5 BIG-IP DNS config file and import the configuration (of your existing platforms) directly. It automatically creates private platforms for your Citrix ADC or F5 BIG-IP DNS configuration.

Import Citrix GSLB from ADM Service

This option enables you to directly import all your GSLBs configured in the ADM Service.

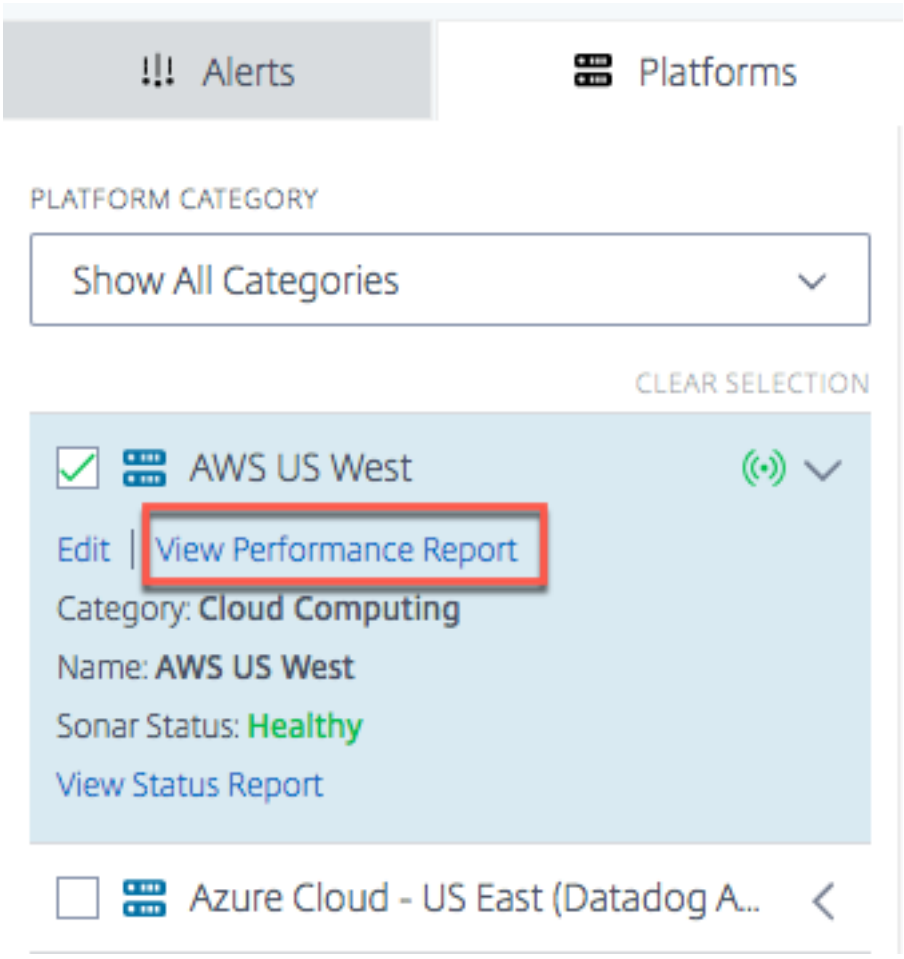
If you are using Citrix Cloud ADM Service, you can import the GSLBs configured there. The basic site information - IP and name are imported as ITM platforms. ITM geo-locate the site and allow the platform to be displayed on the Visualizer for performance analysis.

Performance Report

The Radar Performance Report provides details about specific platforms, alerts triggered, and each network from which it was measured. The report displays Response Time or Availability measurements and the time period for the issue that was measured. It includes all the filters that were applied in the **Visualizer**.

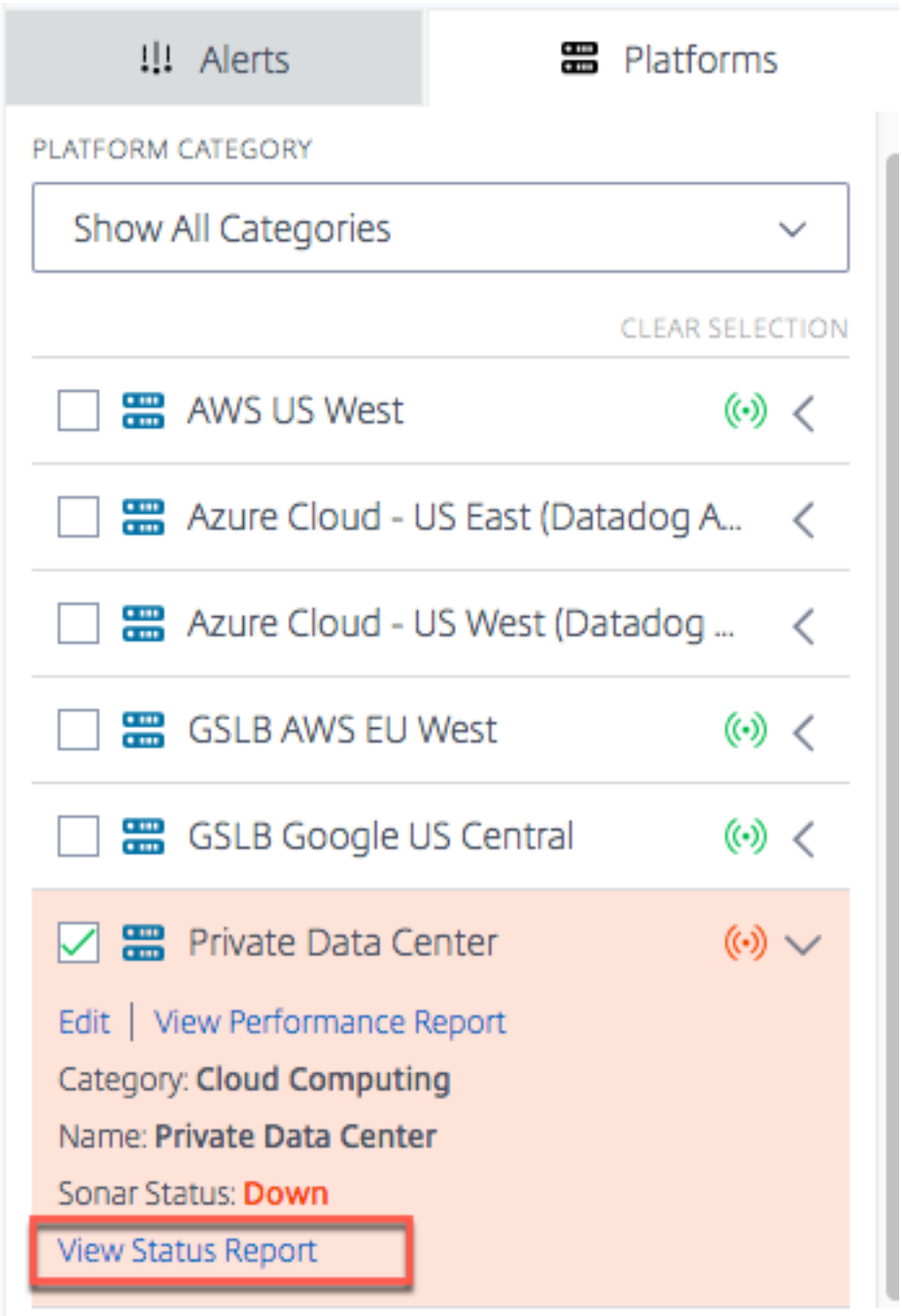
To view the performance details of a specific platform for which the alert was triggered, do the following.

1. Click the platform icon or the alert icon on the map to highlight it and check the box in the list on the right.
2. Click the arrow next to the platform or alert to expand it.
3. Click the **View Performance Report** link to go to the Radar **Performance Report** page.

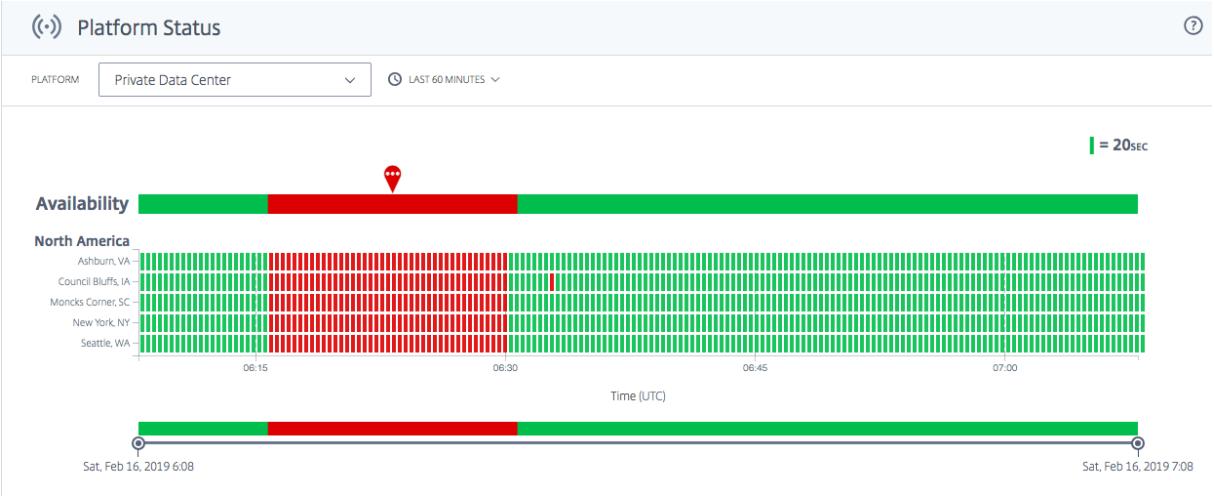


Status Report

For synthetic monitoring alerts, you can see alert details by expanding the platform to view details and then clicking **View Status Report**.



The **View Status Report** link takes you to the Sonar **Platform Status** page and gives you details of the health of your platform based on real-time synthetic monitoring checks.



Radar

December 17, 2020

Overview

Radar forms the backbone of the data collection methodology. Radar uses a JavaScript script embedded within a content page or application provider's pages to collect information about the performance and availability of a data center or delivery platform.

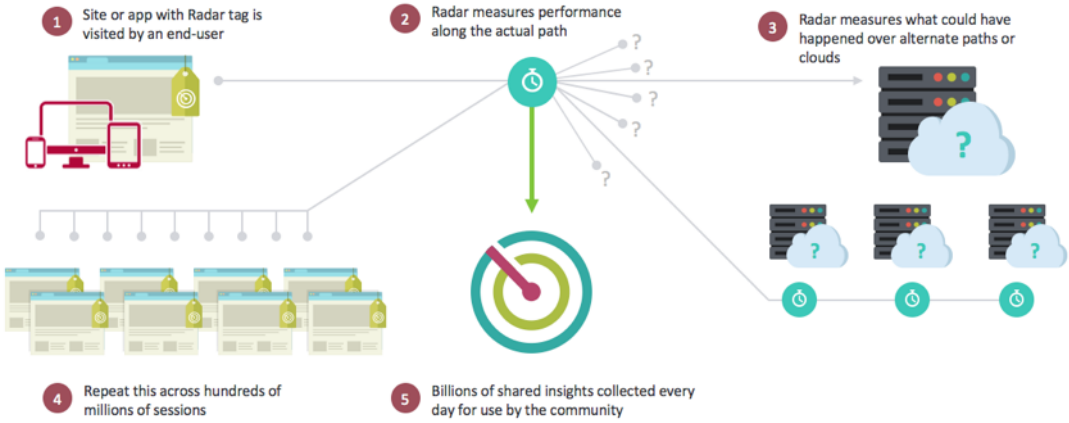
The Radar client is a JavaScript application that runs on customer webpages and inside mobile applications. Its core purpose is to gather network performance data used to drive intelligent routing decisions via Openmix, and provide optional plug-ins to enable other Intelligent Traffic Management services, such as Page Load Time, Page Resource Timing, and Video Playback Metrics.

The Radar client is full-featured, yet lightweight, and unobtrusive. The client waits until after most of the page resources have downloaded before performing the bulk of its work, and all network communication is performed in an asynchronous manner wherever possible. These instructions specify which platform to measure next during the session, picked from among the community platforms and any private platforms specific to that community member. They also indicate the types of measurements to be performed, which may include availability, round trip time, throughput or other metric collection.

To make it as small as possible, the JavaScript is compiled with advanced optimizations using the Google Closure Compiler. Advanced optional features are delivered as plug-ins for customers opting to use them.

Radar Community

Using a unique, community-based approach, Radar brings unrivaled transparency to the global performance and availability of the world's largest public infrastructures, from Cloud Computing and Storage to Content and Application Delivery Networks. Using Radar, customers can quickly find the best—and worst—performing platforms for each one of their visitors.



Radar is the Internet's first cloud monitoring cooperative. Becoming a Community Member means unlimited access to our historical reporting database, including detailed segmentation by provider, country, and network.

Being a Radar community member also provides a rich set of tools for capturing the service levels provided by both internal and external content delivery infrastructures. Unique to Radar is the ability to utilize your website visitors to measure the experience they would receive from platforms not currently used by an enterprise. The same methodology enables objective evaluations of cloud platforms throughout their lifecycle, including on-going evaluation of performance relative to SLAs.

By adding a simple JavaScript tag to your webpage or an SDK to mobile applications, customers can turn each of their visitors into a virtual 'test agent'. Radar triggers device-based measurements by downloading reference objects and comparing internal & external infrastructure, data centers, delivery networks, and cloud platforms as seen by the actual end-users of sites or web applications.

Key Benefits of participating

Radar addresses multiple web delivery challenges through its approach to monitoring and data gathering. Key benefits of participating in the Radar community are:

- Massive testing environment, with end-users in every network in every location (42,000+ recognized networks so far).
- Gain important information on the service providers prior trialing to make a more informed decision.

- Transparency into the performance of current providers and how they behave in geographies where you do and do not have users.
- Focus on the metrics that make a real difference to a web and mobile users (Performance, Availability & QoS).
- Global (190+ countries) unrestricted view of information down to the country, network, region, and state levels.
- Real, unbiased data by using end-users Radar data is “real world” information rather than a synthetic test or best guess.
- All users are not the same: Understand different machines, connections, & devices.
- Visibility into performance of actual pages.

Benchmarks

ITM Radar provides 3 main benchmarks:

- Community Benchmarking
- Private Benchmarking
- Page Load Benchmarking

Community Benchmarking of CDN, Cloud, and Data Centers

Community measurements are sourced through a crowd sourcing model providing the customer a view of a vendor’s performance and availability at a geographic and logical level globally. The community measurements allow comparisons to be made between a vendor’s quality of experience as seen from the end-user and allow a “what-if” analysis in evaluating vendors and suppliers for content and application distribution. By using a crowdsourcing model, ITM customers benefit by gaining a greater level of granularity and quality of data in evaluating and monitoring vendor performance, even in locations where a customer may not have a high density of users, or indeed any users at all.

The measurements themselves use a standard set of objects located across the different Cloud and CDN vendors that end-users download when they execute the Radar JavaScript client, or mobile SDK logic, on a content owner’s site or application.

The following metrics are then reported back to ITM and presented within the Portal or API reporting interfaces:

- Availability—whether the object loads or not.
- Response Time—how long it takes for the server to respond to a subsequent request, once all of the noise of establishing a connection is completed. This is a relatively close approximation of TCP round-trip time (RTT) from the browser to the provider.
- Throughput—this is the data rate of the connection, in kilobits per second, as measured from the retrieval of a 100 KB object.

Private Benchmarking

As part of the Radar Tag deployment, ITM provides the ability for the customer to create their own "benchmark" tests that are measured by the customer's visitors. This can be for Data Centers or their own CDN and Cloud contracts. As with the community benchmark measurements, the same metrics are supplied – Availability, Response Time, and Throughput allowing the customer to effectively evaluate an existing content delivery strategy.

This private information is only available to the customer and is not shared.
Example uses include:

- Their own data center architecture/s
- Using their own test-object or page
- Using their own contract and account with a specific vendor or set of vendors

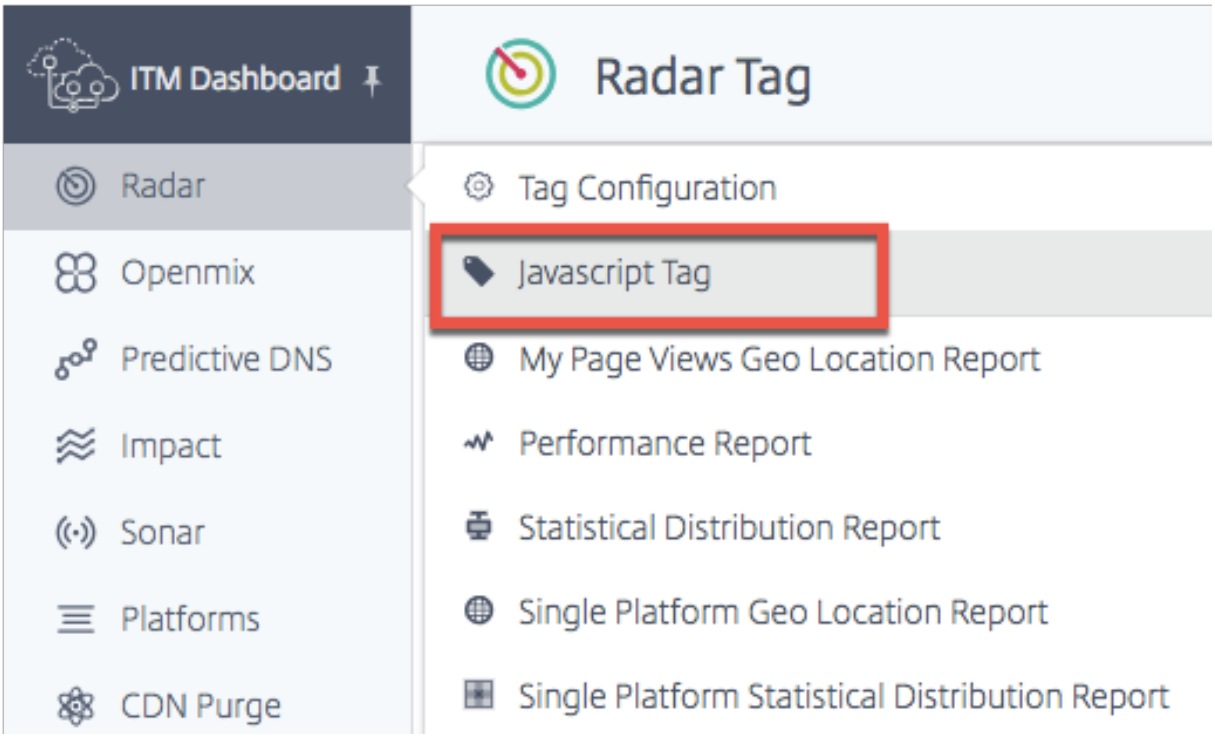
Radar Page Load Benchmarking

Within Radar ITM provides the ability for the customer to see detailed information on the how the pages that the tag is implemented on are being downloaded. ITM provides information that allows you to see the performance actual end-users are experiencing when interacting with your webpages. The data is provided through the Navigation Timing API supported by many of the newer version browsers.

Radar Tag

The Radar tag can be integrated using a JavaScript snippet. To navigate to the **Radar Tag** page, do the following:

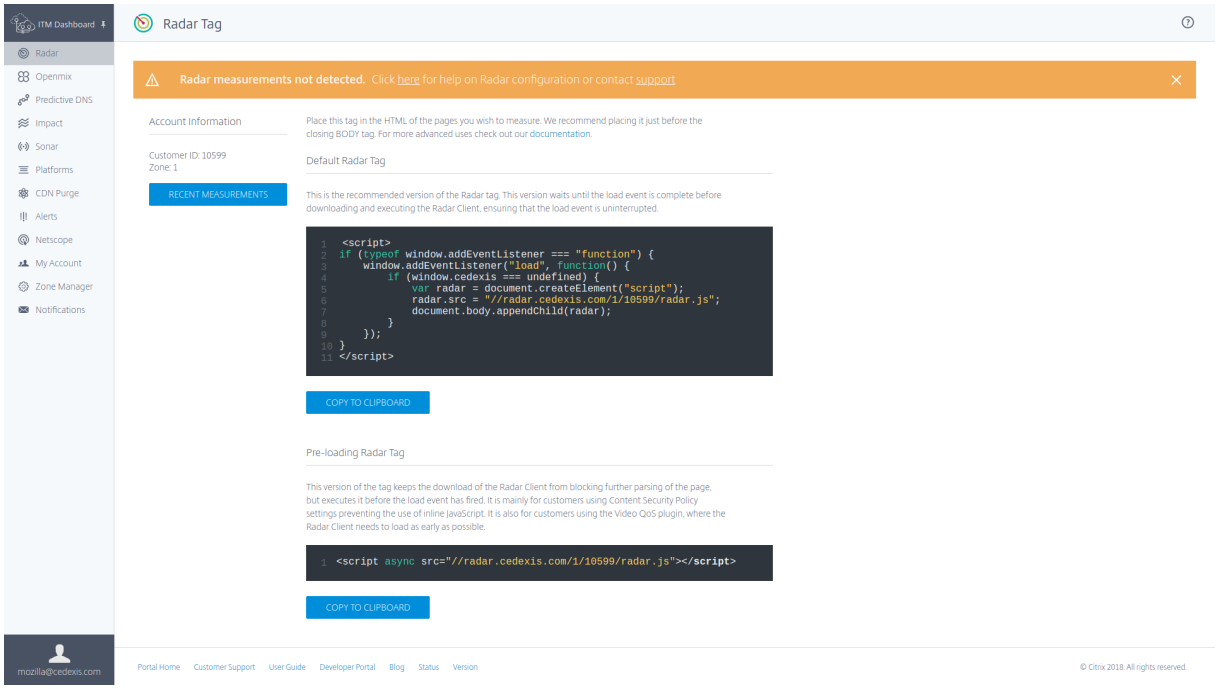
1. Sign in to the Citrix Intelligent Traffic Management Portal.
2. From the left navigation menu, select **Radar > Javascript Tag**.



The **Radar Tag** page opens.

If you haven't configured the Radar tag yet, you see an orange horizontal bar on the top of the screen telling you that Radar measurements were not detected.

This orange bar will also appear if the tag was not configured correctly.



Alternatively, if the Radar Tag is working as expected, you see a green horizontal bar telling you that

Radar measurements were successfully obtained.

On this page you can select the tag version that is applicable to your usage and copy it to the clipboard.

Note: It is important to not change this JavaScript snippet. The code includes important information which if changed can create unexpected or unreliable behavior.

Integrating the Radar Tag

Integrating the Radar tag is relatively simple. All you need to do is add one of the JavaScript snippets below to your site markup. Place it in the HTML of the pages you want to measure. We recommend placing it at the bottom of page before the closing body tag </body>.

Default Radar Tag

This is the recommended version of the Radar tag. This version waits until the load event is complete before downloading and executing the Radar Client, ensuring that the load event is uninterrupted.

```
<script>
if (typeof window.addEventListener === "function") {

    window.addEventListener("load", function() {

        if (window.cedexis === undefined) {

            var radar = document.createElement("script");
            radar.src = "//radar.cedexis.com/1/54621/radar.js"; //
                replace with user specific value
            document.body.appendChild(radar);
         }

     }
 );
 }

</script>
<!--NeedCopy-->
```

This version of the tag keeps the download of the Radar Client from blocking further parsing of the page, but executes it before the load event has fired. It is mainly for customers using Content Security Policy settings preventing the use of inline JavaScript. It is also for customers using the Video QoS plug-in, where the Radar Client needs to load as early as possible.

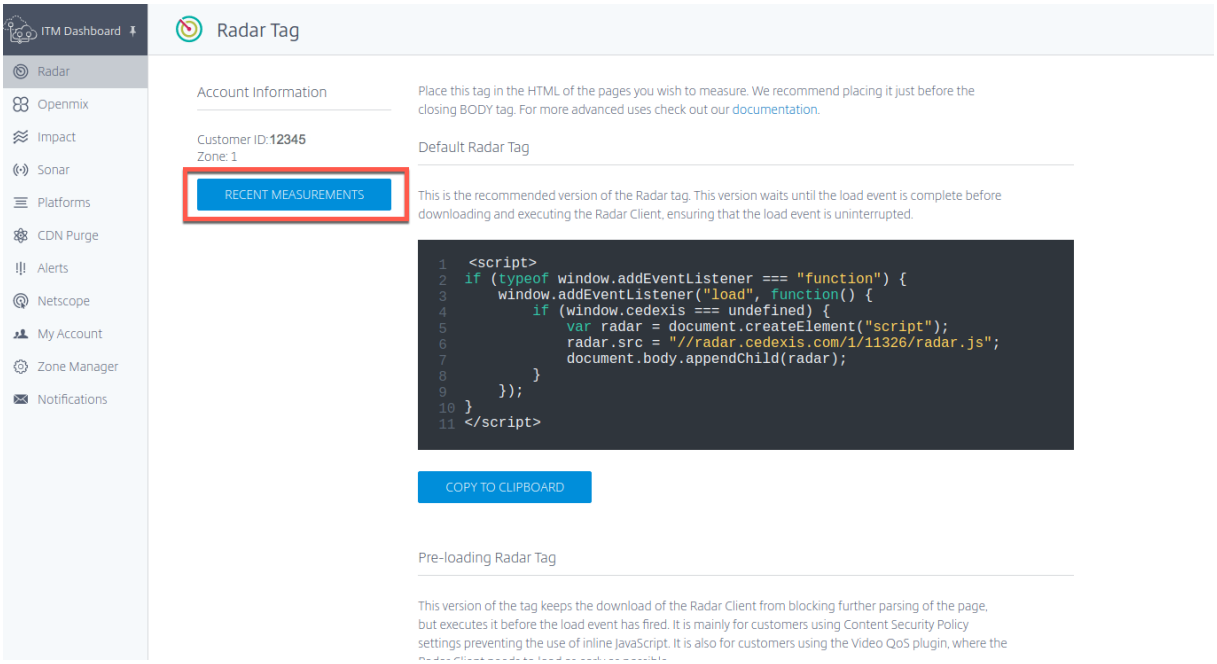
```
<script src="//radar.cedexis.com/1/54621/radar.js" async></script>
```



```
<!--NeedCopy-->
```

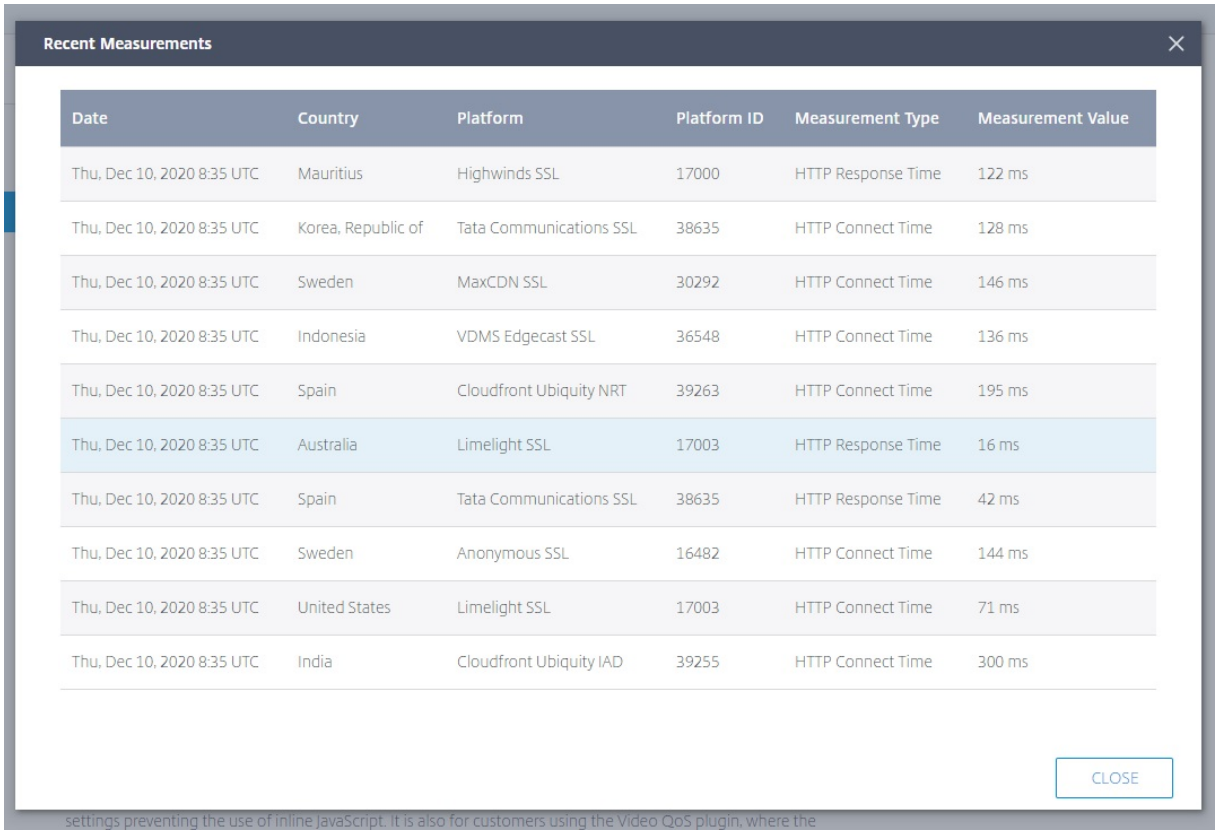
Recent Measurements

The **Recent Measurements** table allows you to view the latest measurements that were taken using Radar.

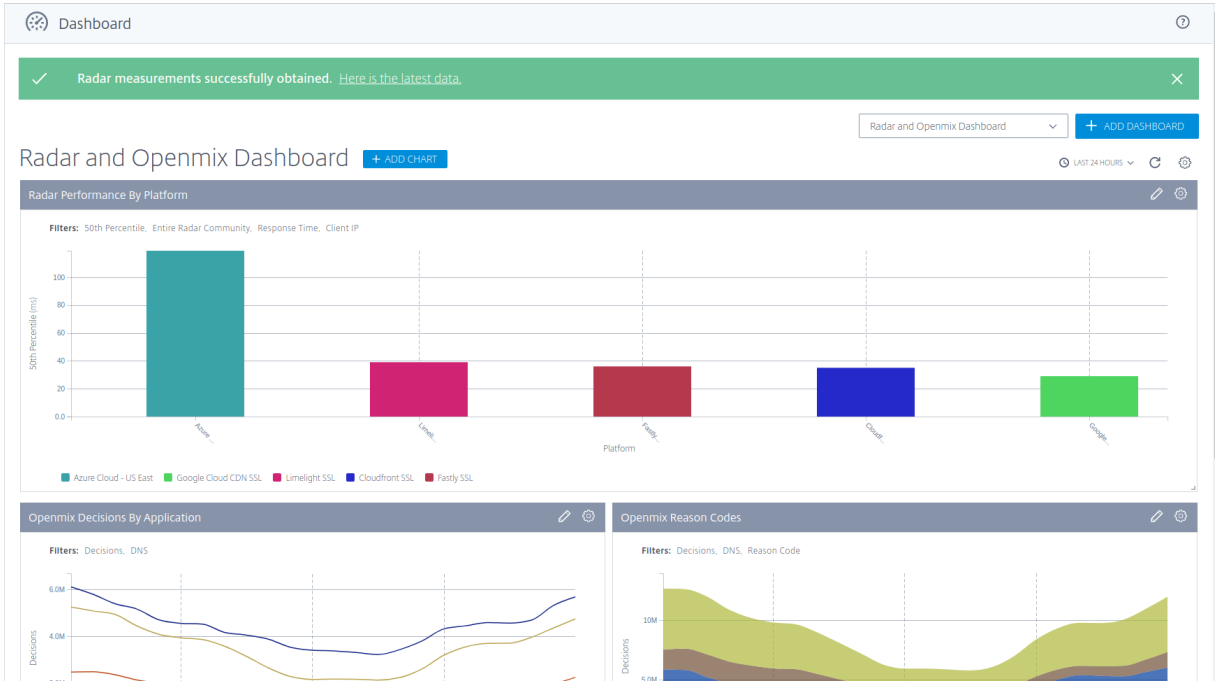


Click the **Recent Measurements** button. It gives you the following information:

- Date and time when the measurement was taken in UTC.
- Country where the measurement was taken.
- The platform that was used for taking the measurement.
- The ID of the platform.
- The type of measurement taken that is, Connect Time (in milliseconds), Response Time (in milliseconds), or Throughput (in Kilobits Per Second)
- The actual value of the measurement in milliseconds (for connect time and response time) or Kilobits Per Second (for throughput).



The Radar measurements bar will also appear in the Radar **Dashboard** page when you first log into the ITM portal.



Integration with Mobile Apps

Integration with mobile apps takes place via wrappers around hidden web views that run the JavaScript client. This ensures that data collected in browsers and mobiles apps is consistent.

Instructions for integrating Radar with iOS app
This following GitHub repository contains the wrapper code and step-by-step instructions for integrating Radar with iOS app:

Radar Runner for iOS

Instructions for integrating Radar with Android
Android Radar is a client library that makes it easy to integrate Radar into Android apps. It can be found here:

AndroidRadar Library

Integration with Citrix ADC

The Radar tag is important because it supplies Openmix with measurements that allow Openmix to make better routing decisions. The more the webpages using the tag, the better the routing decisions.

The following methods enable you to place the Radar JavaScript tag into your webpage using Citrix ADC. You can either use the command line or the Citrix ADC Configuration Utility.

These methods allow you to inject the Radar tag into your responses. To inject the Radar tag, you need to use rewrites. Rewrites are broken down into three steps: creating actions, configuring policies, and binding policies.

Command line Configuration

Command line Configuring Rewrite Action

Template:

```
add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
    pattern <expression> | -search <expression>] [-refineSearch <string
    >] [-comment <string>]
<!--NeedCopy-->
```

Example:

```
add rewrite action radar_tag action insert_after HTTP.RES.BODY(HTTP.RES
    .CONTENT_LENGTH).BEFORE_STR("</body>") '\"<script async src=\\"//
    radar.cedexis.com/1/<customer_id>/radar.js\\"></script>\"'
<!--NeedCopy-->
```


Note: Insert your own Customer ID where it says <customer_id>

Command line configuring Rewrite Policy

Template:

```
add rewrite policy <name> <rule> <action> [<undefAction>] [-comment <
    string>] [-logAction <string>]
<!--NeedCopy-->
```

Example:

```
add rewrite policy radar_tag_policy HTTP.RES.HEADER("Content-Type").
    TO_LOWER.CONTAINS("text/html") radar_tag_action
<!--NeedCopy-->
```

Command line binding Rewrite Policy

Template 1:

```
bind vpn vserver <name> [-policy <string> [-priority <positive_integer
    >] [-secondary] [-groupExtraction] [-gotoPriorityExpression <
    expression>] [-type <type>]] [-intranetApplication <string>] [-
    nextHopServer <string>] [-urlName <string>] [-intranetIP <ip_addr> <
    netmask> ] [-staServer <URL> [-staAddressType ( IPV4 | IPV6 )]] [-
    appController <URL>] [-sharefile <string>]
<!--NeedCopy-->
```

Example 1:

```
bind vpn vserver <name_of_vserver> -policy radar_tag_policy -type
    RESPONSE -priority 10
<!--NeedCopy-->
```

Template 2:

```
bind cs vserver <name> (-lbvserver <string> | -vServer <string> | (-
    policyName <string> [-targetLBVserver <string>] [-priority <
    positive_integer>] [-gotoPriorityExpression <expression>] [-type (
    REQUEST | RESPONSE )] [-invoke (<labelType> <labelName>) ] ) | (-
    domainName <string> [-TTL <secs>] [-backupIP <ip_addr|ipv6_addr|*>]
    [-cookieDomain <string>] [-cookieTimeout <mins>] [-sitedomainTTL <
    secs>]))
<!--NeedCopy-->
```


Example 2:

```
bind cs vserver <name_of_vserver> -policyName radar_tag_policy -type
    RESPONSE -priority 10
<!--NeedCopy-->
```

Template 3:

```
bind lb vserver <name>@ (<serviceName>@ [- weight <positive_integer>])
    | <serviceGroupName>@ | (- policyName <string>@ [-priority <
    positive_integer>] [- gotoPriorityExpression <expression>] [-type (
    REQUEST | RESPONSE )] [-invoke (<labelType> <labelName>) ] )
<!--NeedCopy-->
```

Example 3:

```
bind lb vserver <name_of_vserver> -policyName radar_tag_policy -type
    RESPONSE -priority 10
<!--NeedCopy-->
```

Template 4:

```
bind rewrite global <policyName> <priority> [<gotoPriorityExpression>]
    [-type <type>] [-invoke (<labelType> <labelName>) ]
<!--NeedCopy-->
```

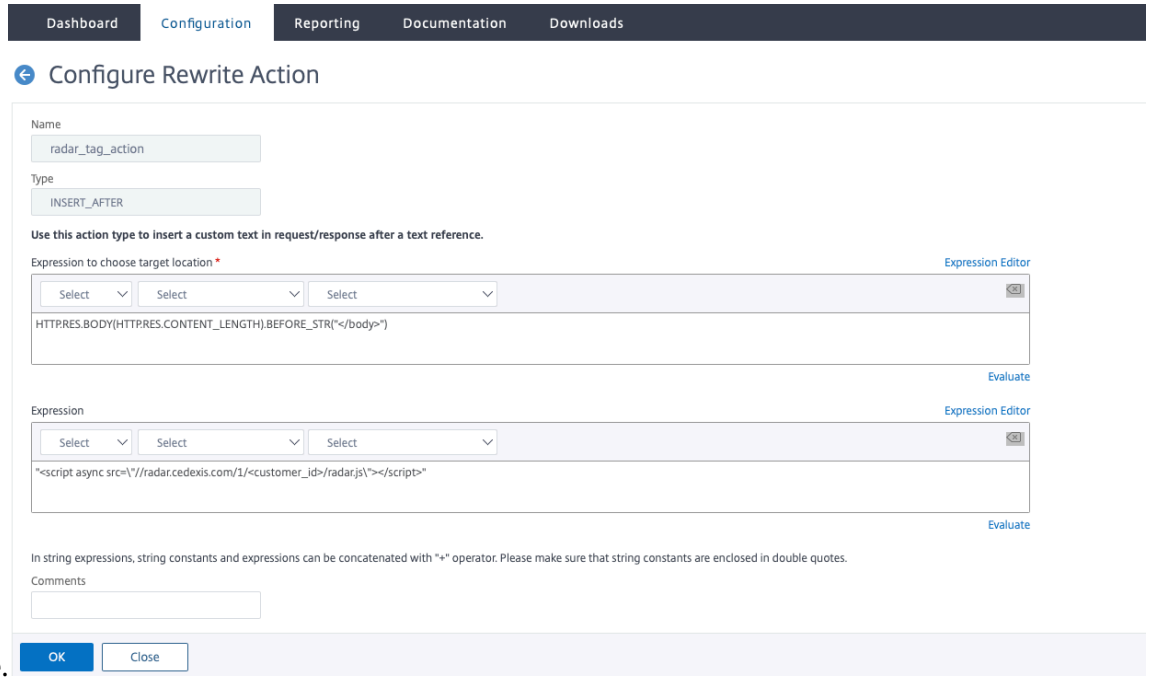
Example 4:

```
bind rewrite global radar_tag_policy 100 -type RES_DEFAULT
<!--NeedCopy-->
```

Configuration of GUI Utility

GUI Rewrite Action

1. From the left navigation menu on the **Citrix ADC Configuration** page, navigate to **AppExpert** -> **Rewrite** -> **Rewrite Actions**
2. Select the **Add** button.
3. In the **Configure Rewrite Action** page, input the expression as shown in the exam-

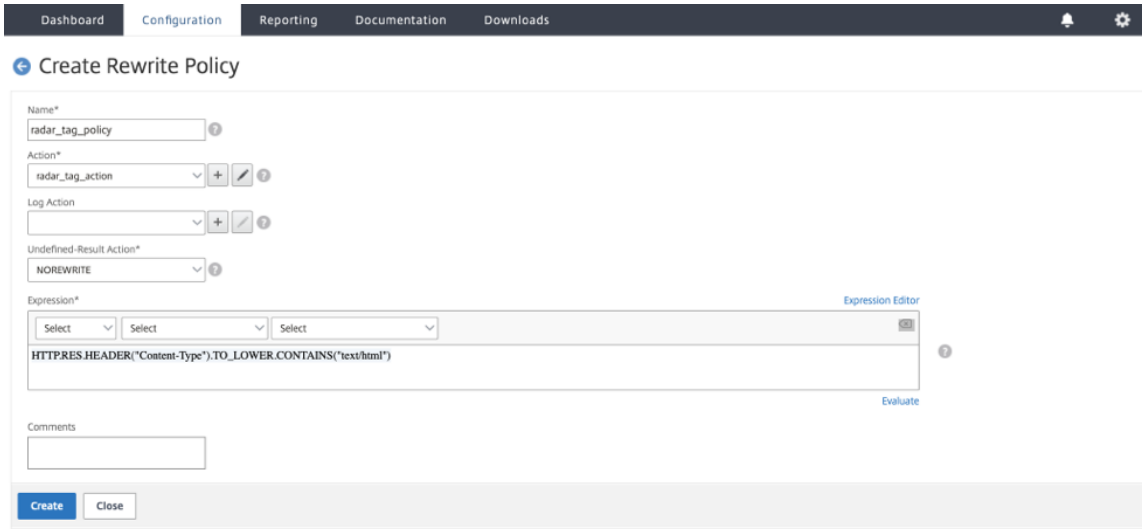


ple.

4. In the Radar script, enter your Customer ID in the space marked `<customer_id>`.
5. Select **OK**. You have completed creating your rewrite action.

GUI Rewrite Policy

1. From the left navigation menu on the **Citrix ADC Configuration** page, go to **AppExpert** -> **Rewrite** -> **Rewrite Policies**
2. Select the **Add** button.
3. On the **Configure Rewrite Policy** page, input the expression as shown in the example.



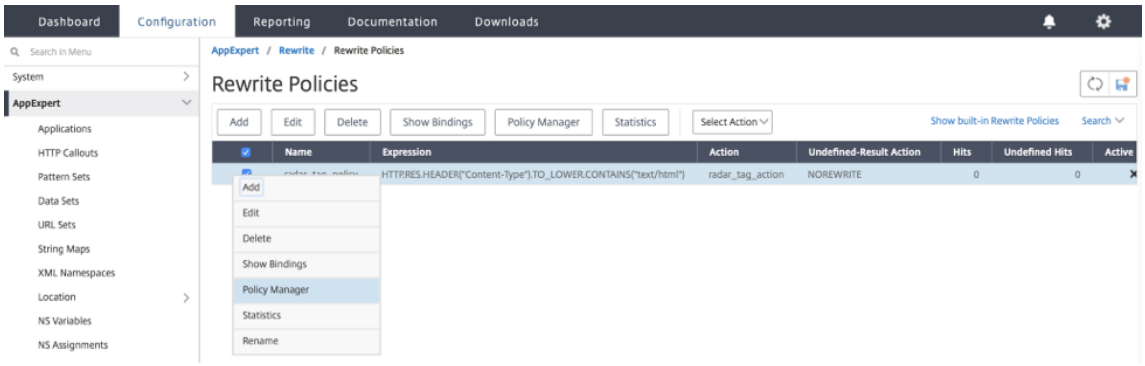
4. Click **Create**.

You have completed the configuration of the Rewrite Policy.

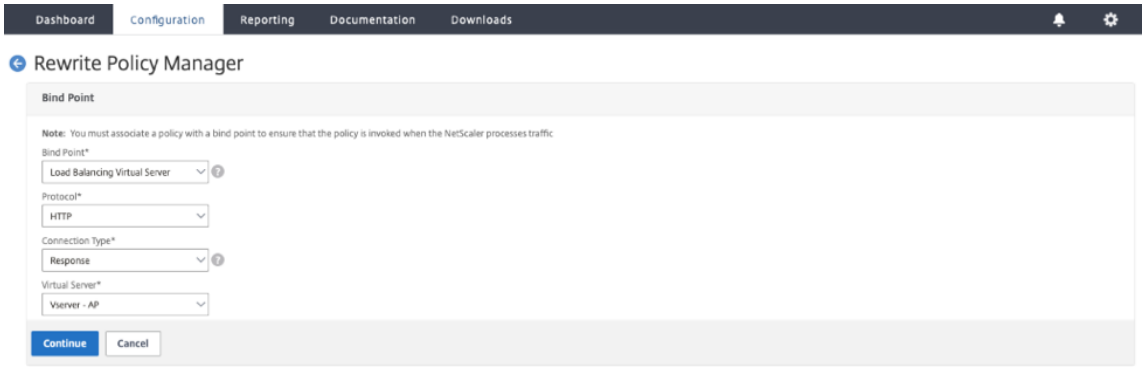
GUI Binding Rewrite Policy

Once you're done configuring your policy, the last step is to bind the policy by using the **Policy Manager**.

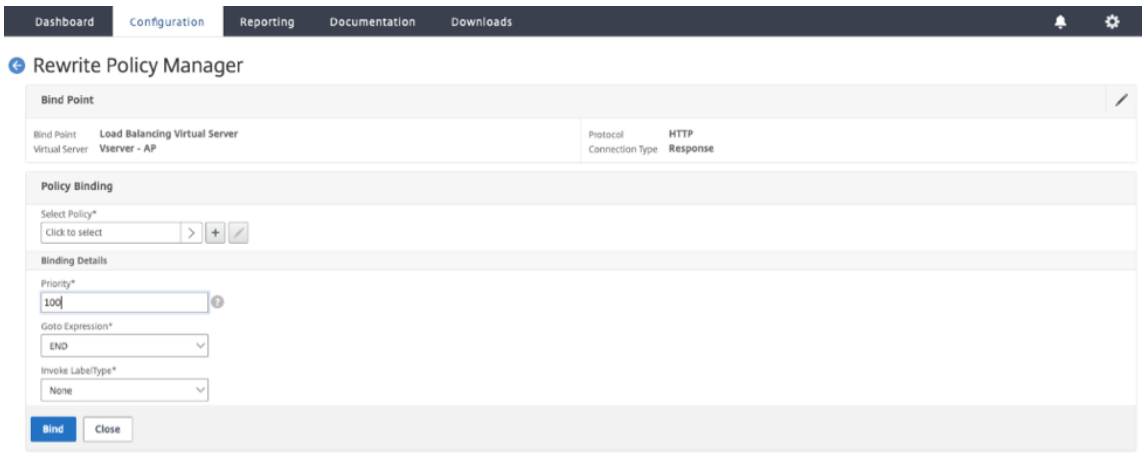
1. Go to the **Rewrite Policies** Page.
2. Select the rewrite policy that you created for the Radar Tag.
3. Go to **Policy Manager**.



4. In the **Policy Manager** page, you can bind the policy by doing the following.
   - For **Bind Point** you have the option to select **Override Global**, **VPN Virtual Server**, **Content Switching Virtual Server**, or **Load Balancing Virtual Server**.
   - For **Protocol** select **HTTP**.
   - For **Connection Type** select **Response**.
   - For **Virtual Server** use your own virtual server name.



   - Click **Continue**.
   - In the next page, select the **Rewrite Policy** that you created earlier.
   - Add **Binding Details**.
   - Click **Bind**.

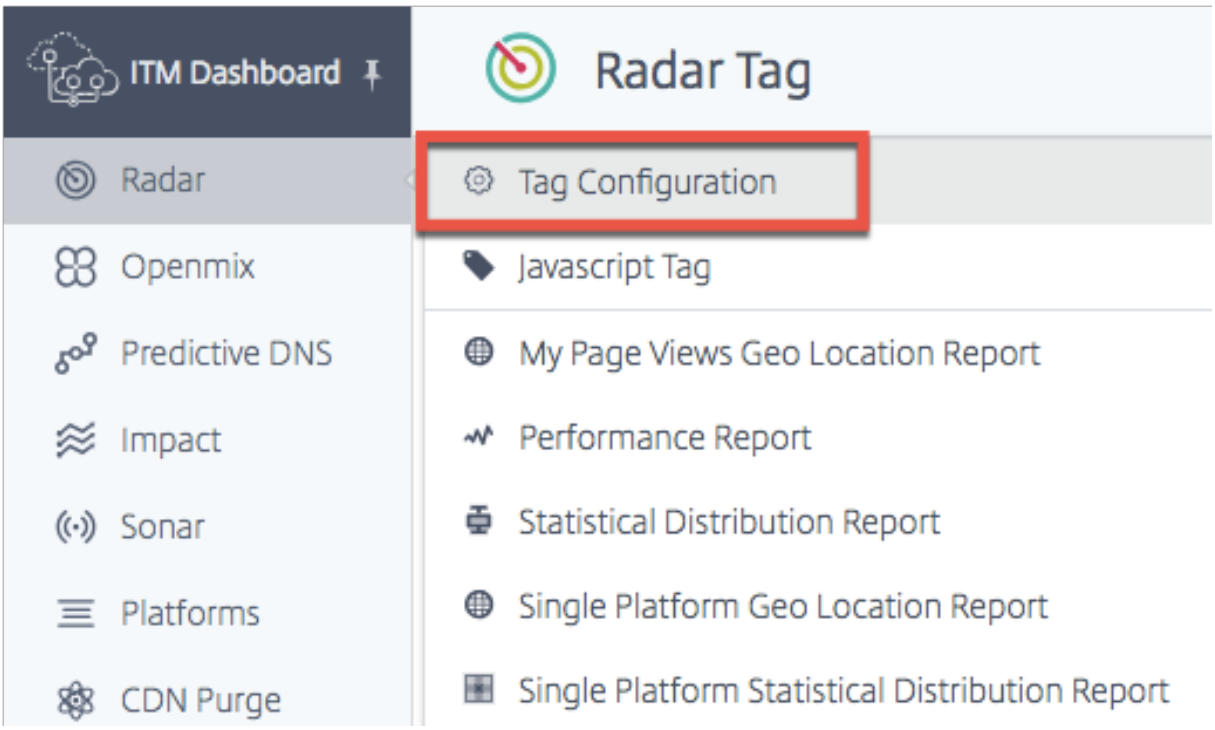


With the above methods you are able to insert the Radar tag into your webpages. However, it must be noted that this is a basic implementation. Further filtering can be done to better control the pages that have the tag implemented.

Radar Tag Configuration

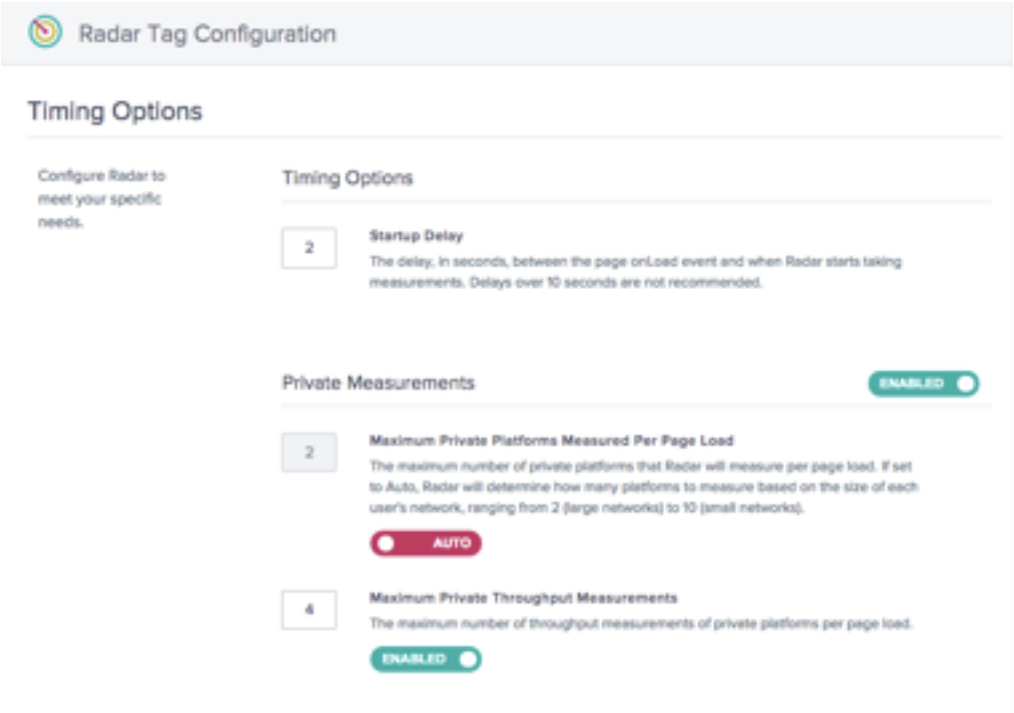
You can configure Radar on the **Radar Tag Configuration** page.

1. Sign in to the Citrix Intelligent Traffic Management Portal.
2. From the left navigation menu, select **Radar > Tag Configuration**.



The Radar Tag Configuration page opens. Here you can set various options to customize Radar measurements. The Radar JavaScript has parameters that you can customize to adjust timing and delay

elements; number of tests completed by end-users for community and private measurements; and time out values to measure availability, and so on



The following table provides information on what the configuration options are and the default settings for each. When making changes, be sure to click **Update Radar Settings** at the bottom of the screen to apply the changes.

Function	Parameter	Description	Default Setting
Timing Options	Startup Delay	The delay, in seconds, between the page onLoad event and when Radar records navigation timing.	2 Seconds

Function	Parameter	Description	Default Setting
	Repeat Delay	The delay, in minutes, between measurement sessions. If the value is greater or equal to 5, the Radar tag will take more measurements after each repeat delay interval. If the value is 0 the Radar Tag will not take any additional measurements.	5 Minutes
Protocol Options	Always Allow Private HTTPS Measurements	Allows Radar client to take HTTPS measurements even from an HTTP website.	Takes measurements of platforms with URL protocols that match the page where the Radar client is running.
	Allow private HTTP measurements on HTTPS connections.	Allows Radar client to take HTTP measurements from an HTTPS website.	Takes measurements of platforms with URL protocols that match the page where the Radar client is running.
Sample Rate	Radar Sample Rate	The percentage of pages where the Radar tag is activated to take measurements.	Disabled
Private Measurements	Maximum Private Measurements per Page Load	The maximum number of private platforms that Radar will measure per page load.**	Auto*

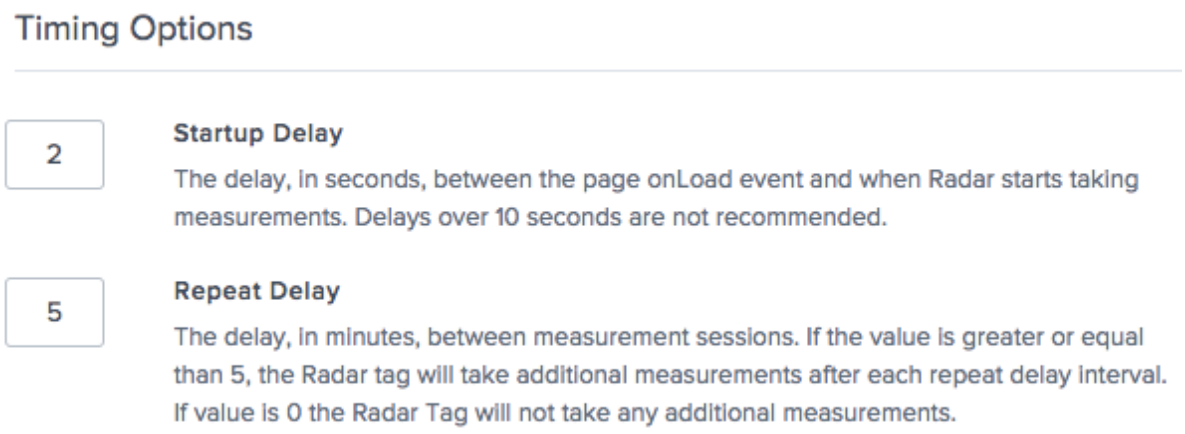
Function	Parameter	Description	Default Setting
	Maximum Private Throughput Measurements	The maximum number of throughput measurements of private platforms per page load.**	4
Community Measurements	Maximum Community Measurements per Page Load	The maximum number of community platforms that Radar will measure per page load.**	Auto*
	Maximum Community Throughput Measurements	The maximum number of throughput measurements of community platforms per page load.**	4

*Auto means that Intelligent Traffic Management determines how many platforms must be measured for a certain session, based on the end user's location. We try to measure more platforms per session for small networks, where data is sparse, rather than from large networks, where it is dense.

**This is the maximum number of measurements attempted per session. For example, Radar can measure 4 private platforms per session, all of them being configured to measure both RTT and throughput. But if Maximum Private Throughput Measurements is set to 2, then the client will stop including the throughput measurements after measuring the first 2 private platforms. For the final two platforms, it will only measure RTT.

Timing options allow you to set the length of time that Radar must wait before starting to take measurements.

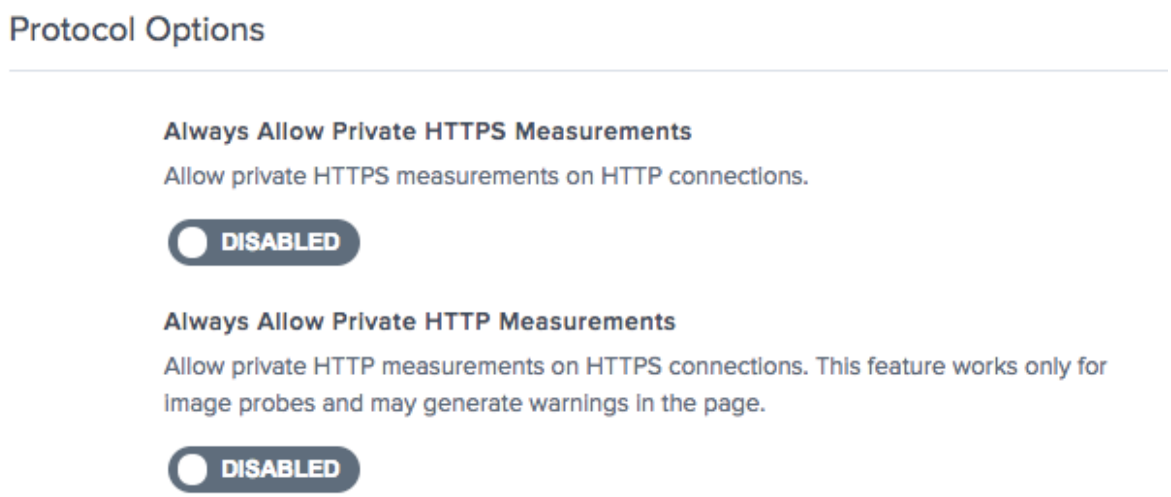
Note: Startup Delay is in seconds, while **Repeat Delay** is in minutes.



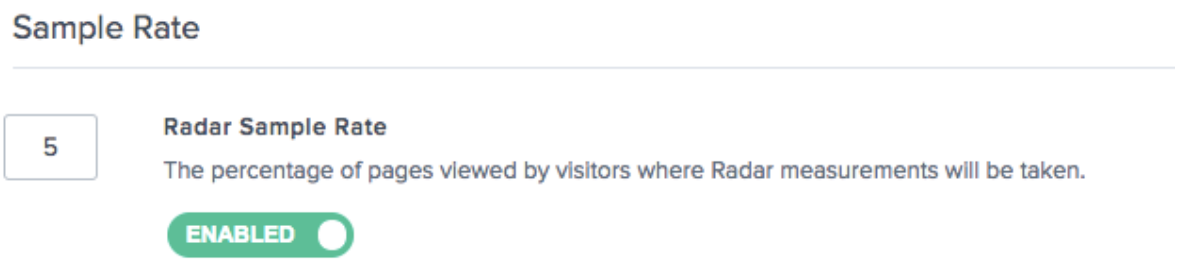
Protocol Options

Normally, the Radar client only measures platforms with URLs whose protocols match that of the page where it is running. These options allow you to override that behavior for private platforms. For example, enabling “Always Allow Private HTTPS Measurements” allows the client to measure `https://myprovider.com/r20.png` from `http://example.com`, while “Always Allow Private HTTP Measurements” allows the client to measure `http://myprovider.com/r20.png` from `https://example.com`.

These options must generally be avoided except for extreme use cases. The best way to ensure that you’re getting adequate private measurement density is to have your platforms configured to measure the platforms and protocols that you actually use in production (and no more), and to have the Radar tag deployed on as many production pages as possible. We sometimes refer to this as “Putting Radar where it’s needed.”

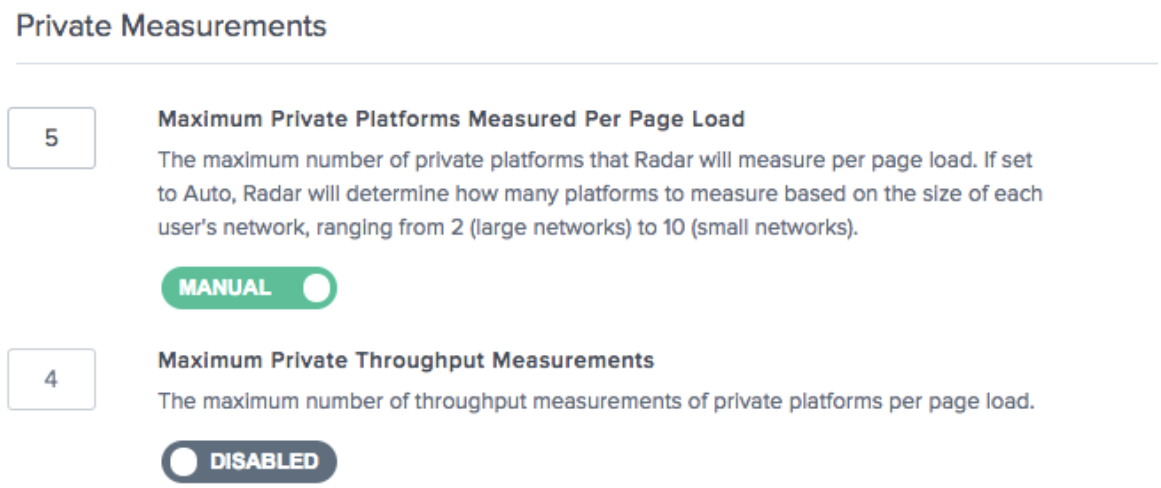


Sample rate enables you to set a percentage of webpages (viewed by users) to collect measurements from. For example, if your website gets a 100,000 page views a day, and you set a 5% sample rate, Radar will only collect measurements from 5% of the 100,000 page views.

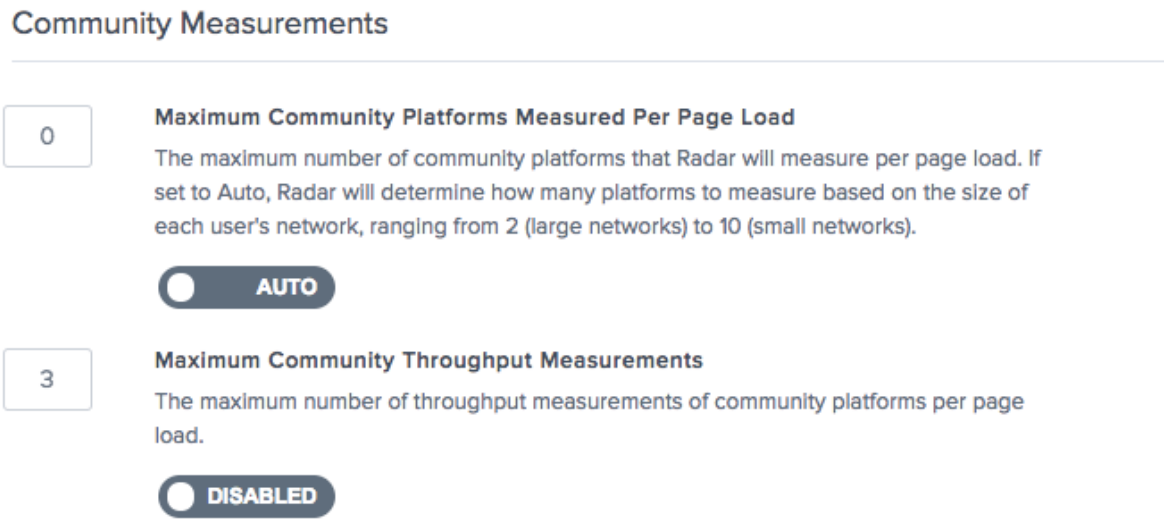


Private measurements

These settings apply to measurements of your private platforms. Private platforms are those that you set up in the **Platforms** section to measure specific CDNs, cloud providers, and other parts of your infrastructure. See Platforms section for more information.



This option allows you configure Radar's behavior when providing information back to the community.



Turn Off Radar Testing

If there is a requirement to quickly turn off Radar measurements must something unexpected occur, you can do so within the Portal to avoid emergency code changes to your site.

On the Radar Tag Configuration page, switch off Private Measurements, Community Measurements or both by clicking the **Enabled** toggle button to Disabled.

Click **Save Radar Configuration** to confirm the changes. The changes may take a minute or two to propagate after which Radar measurements stop.

Radar Client Methodology

A fundamental dimension of client behavior is the **session**. All data that the client sends is associated with a session. Sessions are created by making a call to Citrix servers, known as the initialization request. Sessions expire rather quickly, helping to ensure that only valid Radar data is accepted. Because of this feature, Radar measurements always come in batches associated with their session transaction ID, and we often refer to a "Radar session" to describe the measurements associated with it.

Radar session

A Radar session is the main unit of work that the client performs. It consists of a request to Citrix servers to obtain customer configuration and a set of platforms to measure, followed by requests to measure those platforms, and report the results. These take place in an asynchronous and serialized fashion, so that only one request takes place at a time.A typical session completes in under 10 seconds.

Probe Types

Every report that the client sends has an associated probe type, which tells the system what kind of measurement it is and how to treat it. It also indicates the types of measurements to be performed, which may include availability, round trip time, throughput or other metric collection"

There is an important relationship between availability and performance probing (such as round trip time and throughput). Availability of a particular resource is always measured first in any particular measurement session. Only if the availability measurement succeeds might additional performance measurements of the same resource be taken in that same session."

If a particularly slow network suffers an availability outage, this can result in the aggregate performance of reports that include this network to actually improve. This is only a reporting artifact, as Citrix Intelligent Traffic Management always uses the most granular, network-specific performance data for real-time decisions.

Availability

Availability also known as cold start probes are intended to allow services to warm their caches. Although there is a measurement value associated with this probe. We use the availability probe to determine whether the provider is available.

If a platform is not configured to perform a cold start probe, we use the results of the RTT probe in place of a cold start report to provide availability metrics.

Similarly, for dynamic objects that measure site acceleration services, the client downloads the small test object once and reports the measurement value for both cold start and response time.

Test Object	Definition
Standard	Using Resource Timing timestamps: responseStart - requestStart
Dynamic	Using Resource Timing timestamps: responseEnd - domainLookupStart

RTT

Test Object	Interval	API	Description
Standard	responseStart - requestStart	Resource Timing	The time for a single packet to be returned in response to an HTTP request.
Dynamic	responseEnd - domainLookupStart	Resource Timing	The time for a request to be served, including DNS lookup time, connection time, and response time.

Throughput

Test Object	Interval	API	Description
Standard	File size (kilobytes) * 8 / (responseEnd - requestStart)	Resource Timing	The throughput measured (kilobits per second) for an entire request and response based on a large test object download.
Dynamic	File size (kilobytes) * 8 / (responseEnd - domainLookupStart)	Resource Timing	The throughput measured (kilobits per second) for an entire request and response based on a large test object download. This usually does not include connection time or DNS lookup time in case an RTT test object was already downloaded.

Test Objects

Test objects are files that are hosted on platforms and downloaded by the client to generate measurements. This section describes the different kinds of test objects that the client supports. Not all object types apply to every platform.

Required Header:

The Timing-Allow-Origin response header is required to permit JavaScript access to the low-level timing data supplied by the Resource Timing API. The recommended setting is `Timing-Allow-Origin: *`, which indicates that permission to access the resource's timing data must be granted to JavaScript running on any domain.

Standard

The standard test objects are media, which the client downloads by setting the `src` attribute on an Image object. Once downloaded, the client uses the Resource Timing API to gather performance data.

These test objects must be served with the Timing-Allow-Origin response header. See the **Timing-Allow-Origin Header** section for more information.

Standard Small

The standard small test object is a single pixel image file, used when the client needs to make a lightweight network request.

The standard small test object is used in the following use cases:

- Non-dynamic cold start probes
- Non-dynamic round trip time probes

Standard Large

The standard large test object is a 100KB image file used to measure a platform's throughput.

Large Object Naming: To calculate throughput, the client needs to know the size of the test object.The client determines the file name by looking for KB somewhere in the file name; `r20-100KB.png`, for example. Customers can measure image files of different sizes as long as the name contains the file size in the same manner, for example `myimage-2048kb.jpg`.

Dynamic

Dynamic test objects are used to measure the performance associated with site acceleration services. Each is an HTML file containing JavaScript capable of gathering timestamps from the Navigation Timing API and posting them to the parent page. The client downloads the test object using an iframe and obtains these timestamps, which it uses to calculate measurements.

Security and Validation

The test object is a 40KB object. A new feature of the test object is an HMAC (hash-based message authentication code) that it provides based on query parameters and a secret key that the server has access to. This HMAC is sent back with our measurement, which enables us to validate that the Radar Client was able to access the test object and nothing was cached.

Difference between dynamic and standard test objects:

For standard Radar measurements, we try to isolate only the primary request activity associated with downloading test objects, whereas for site acceleration services our goal is to measure more of the activity. Therefore DNS lookup and connection time are included as well.
Also, dynamic measurements are intended to measure the request performance when hitting the service origin, not just an edge cache.

In the Portal, you can choose this methodology by doing the following:

- From the left navigation menu, go to **Platforms**.
- Click the **Add Platform** icon on the top right corner of the page.
- Go to **Private Platform > Category > Dynamic Content**.
- In the **Radar Test Objects** dialog box, click the **Customize Probes** check box.
- Enter the **Response Time** url and choose **Webpage Dynamic** from the **Object Type** drop-down list.

The dynamic small test object is used to measure availability and round trip time using the same probe for site acceleration services.

iNav

The iNav test object is a static HTML file containing JavaScript able to perform a number of tasks. The client indicates which task it would like performed by including query string parameters in the URL that loads the HTML file in an iframe.
The iNav test object supports the following use cases:
iNav cold start
iNav round trip time

iUNI

The iUNI test object is used to detect the UNI value associated with a set of Radar measurements for a platform (the other method being CORS AJAX which doesn't require a separate test object).

AJAX GET

The AJAX GET methodology can generally be used with any URL that the customer wants to measure, provided that it is served with the **Timing-Allow-Origin** header and an appropriate **Access-Control-Allow-Origin** header.
In the Portal, you can choose this methodology by doing the following:

- From the left navigation menu, go to **Platforms**.
- Click the **Add Platform** icon on the top right corner of the page.
- Go to **Private Platform > Category > Dynamic Content**.
- In the **Radar Test Objects** dialog box, click the **Customize Probes** check box.
- Enter the **Response Time** and choose **AJAX (GET)** from the **Object Type** drop down list.

Timing-Allow-Origin Header

The Timing-Allow-Origin response header is required in order to permit JavaScript access to the low-level timing data supplied by the Resource Timing API.

The recommended setting is `Timing-Allow-Origin: *`, which indicates that permission to access the resource's timing data must be granted to JavaScript running on any domain.

Radar APIs

Radar provides APIs for both operational and data retrieval functions.

- Operations API – Add/Edit/Delete Radar accounts and the control mechanisms for running your account through an API
- Radar Data API – The ITM Radar data API provides aggregates of the Radar public community and private measurement data. Data is updated continuously, and batched approximately every 60 seconds for retrieval by the API. The data API is provided to allow customers to integrate Radar data into their own reports and dashboards. A single call to the API can provide Radar quartile or mean measurement averages for all countries and up 30 ASNs of interest, for each platform.

Radar Reports

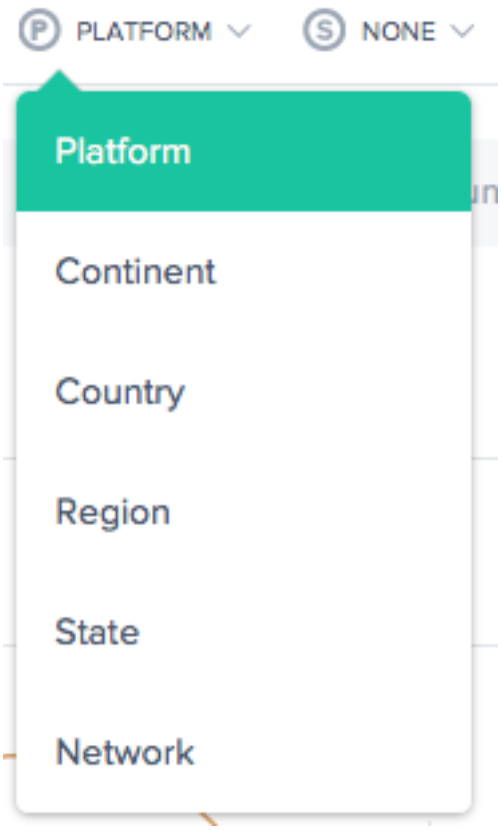
Radar reports provide powerful visibility into the dynamic data collected through the Radar Tag.

Radar members are provided access to a rich data set presented through intuitive interactive charts. The data set collected incorporates both the full public data set of billions of measurements as a context for private data collected from a customer's Radar tag or mobile SDK deployment. Page Load time information is captured with the customer's own tag, providing deep insight into the actual performance experience of your website and mobile application end users.

In addition to performance metrics, Radar reports provide insight into many facets of your end user audience, including: volumes, geographies, user agents, OS types and the timing of their use of your website or mobile application.

Each report is defined below, but here are important aspects of all of the reports:

Primary and Secondary Dimensions

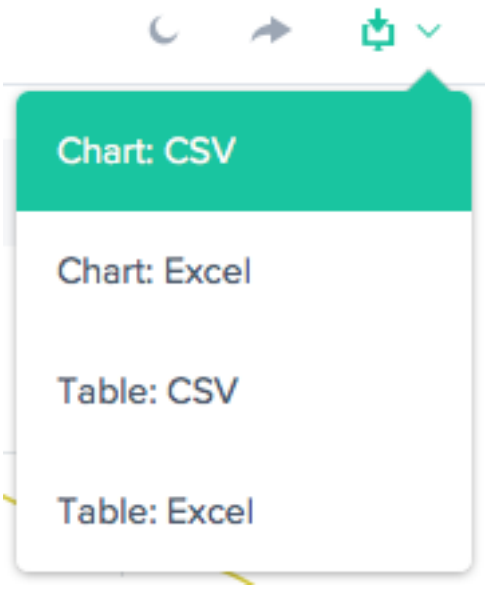


The primary dimension of the chart is selected through a list selection list above the chart. Use this as a powerful pivot on the report.A secondary dimension can be chosen as well to further refine the reporting.

Visualization Background Toggle

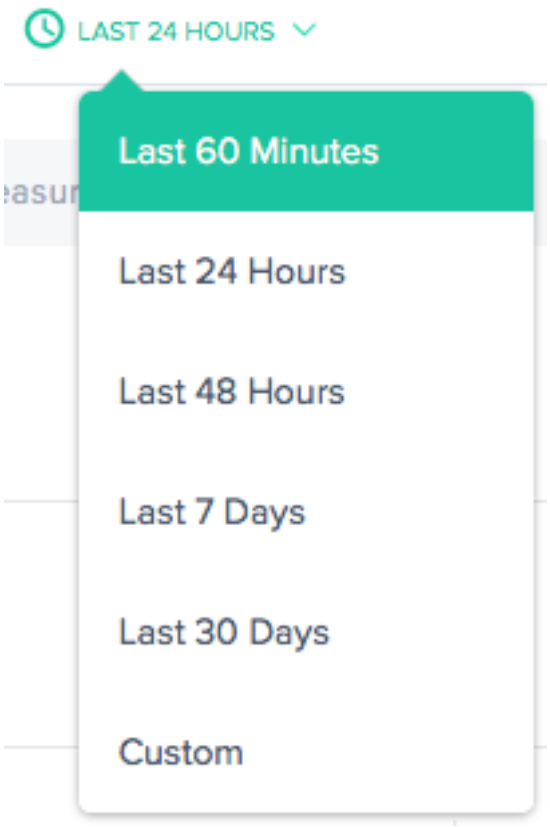
Charts are set to a white background by default. Toggle the background to a dark color for high contrast monitors using the background toggle.

Data Export



In addition, the end-user is able to download the Chart and Table data via the download link at the top of the report.

Filter: Report Time Range



The Radar reports can be generated with a time range of Last 60 Minutes, Last 24 Hours, Last 48 Hours, Last 7 Days, Last 30 Days, or a custom range. The default view is the Last 24 Hours.

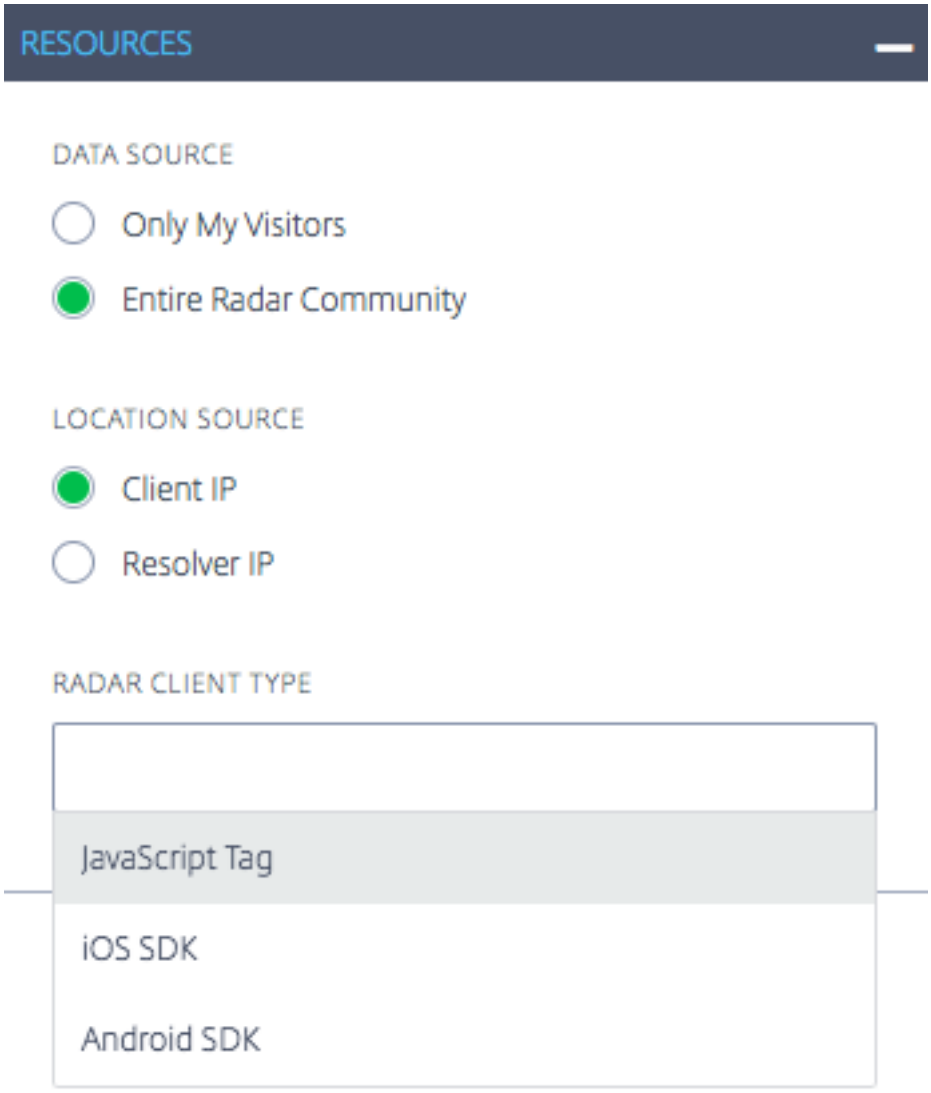
Filter: Platform and Location

The reports vary slightly in terms of which filters are appropriate based on the data. The following are the most common:

- **Platform** – Select one or more platforms (provider) to include.
- **Continent** – Select one or more continents to include.
- **Country** – Select one or more countries to include.
- **Region** – Select one or more geographic regions (where applicable) to include.
- **State** – Select one or more geographic states (where applicable) to include.
- **Network** – Select one or more networks (ASN) to include.

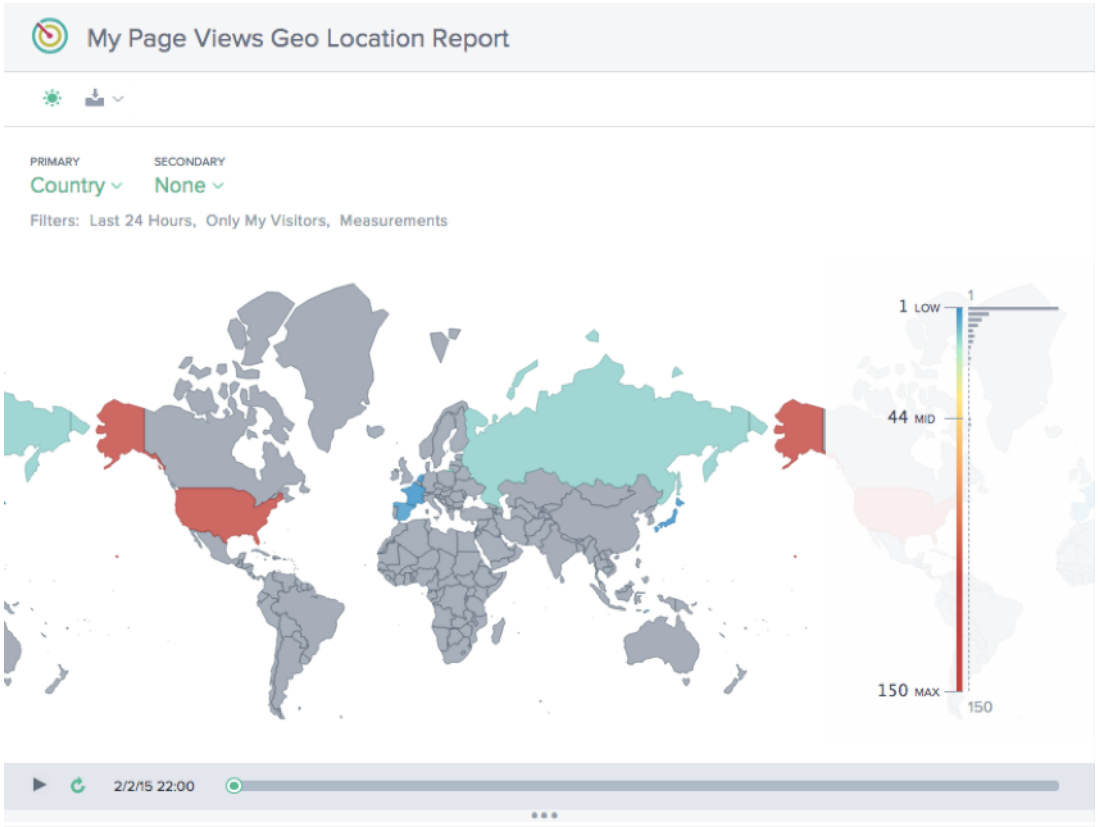
Filter: Resources

- **Data Source** - Include data from the entire Radar Community or from your site visitors only.
- **Location Source** - Select the Client IP or the Resolver IP as your location source.
- **Radar Client Type** - Select the Radar Client Type as a JavaScript Tag, iOS SDK, or Android SDK.



My Page Views Geo Location Report

This report shows the volume of Page Views for each country. This map view can be viewed over time (based on the time range chosen for the report) by selecting the **'Play'** button at the bottom of the chart.

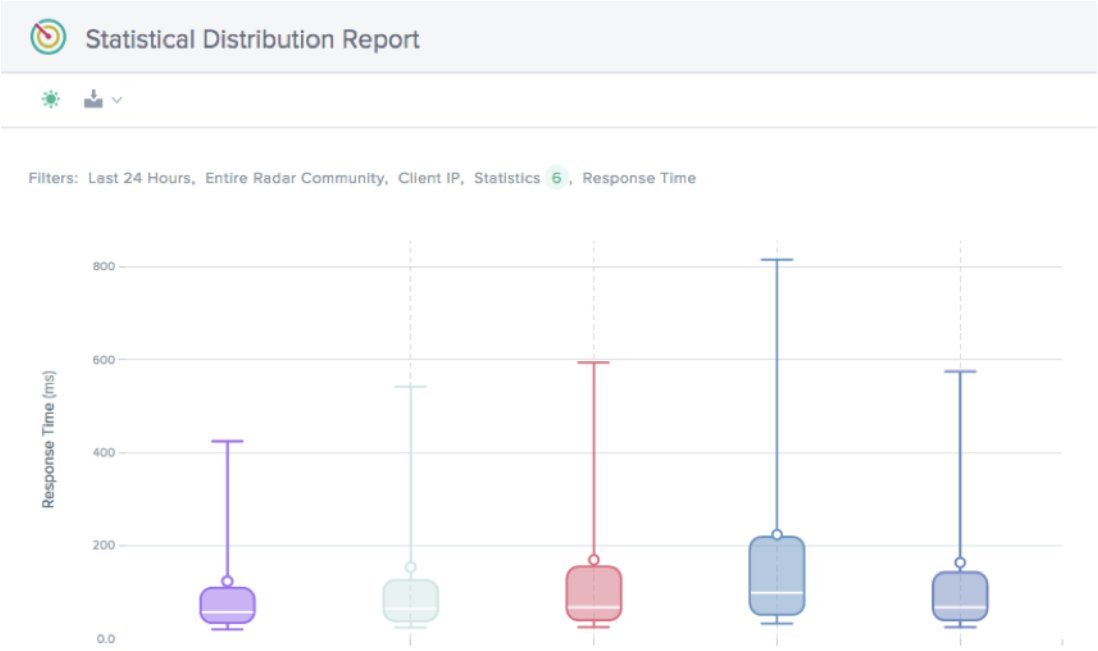


Performance Report

This report shows the trend of performance for each of the Platforms defined.

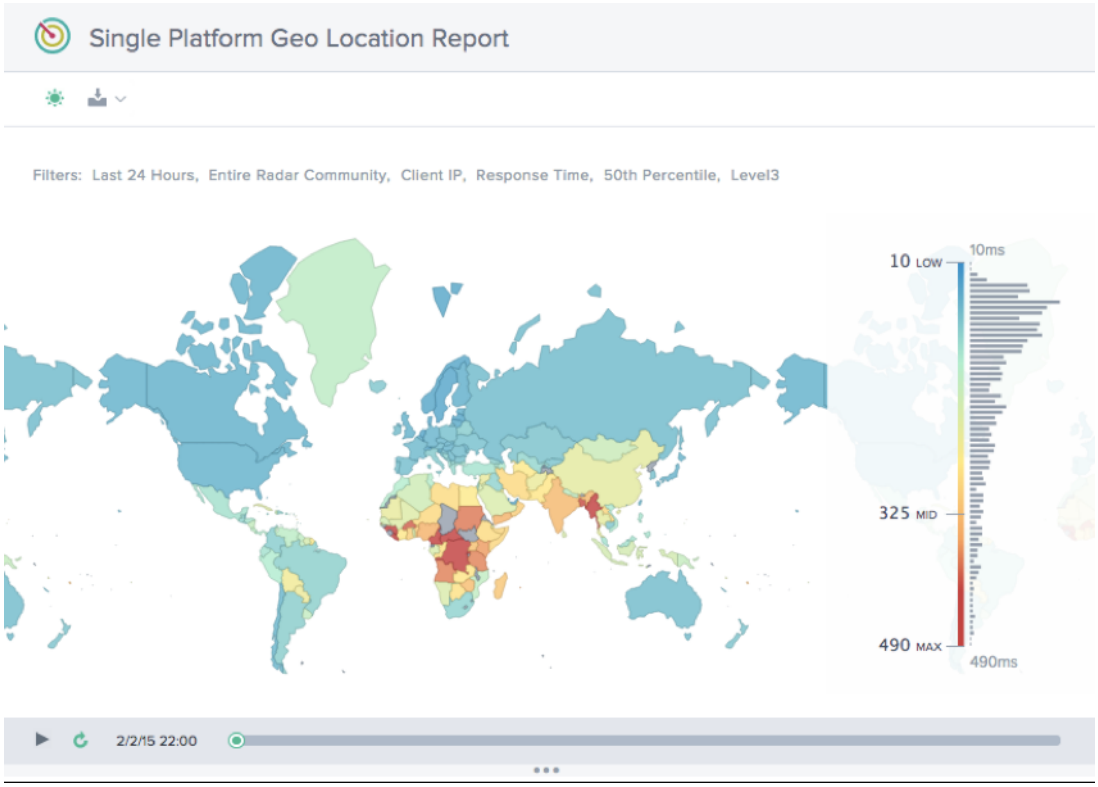
Statistical Distribution Report

This report shows the statistical breakdown for each of the Platforms defined for the account.



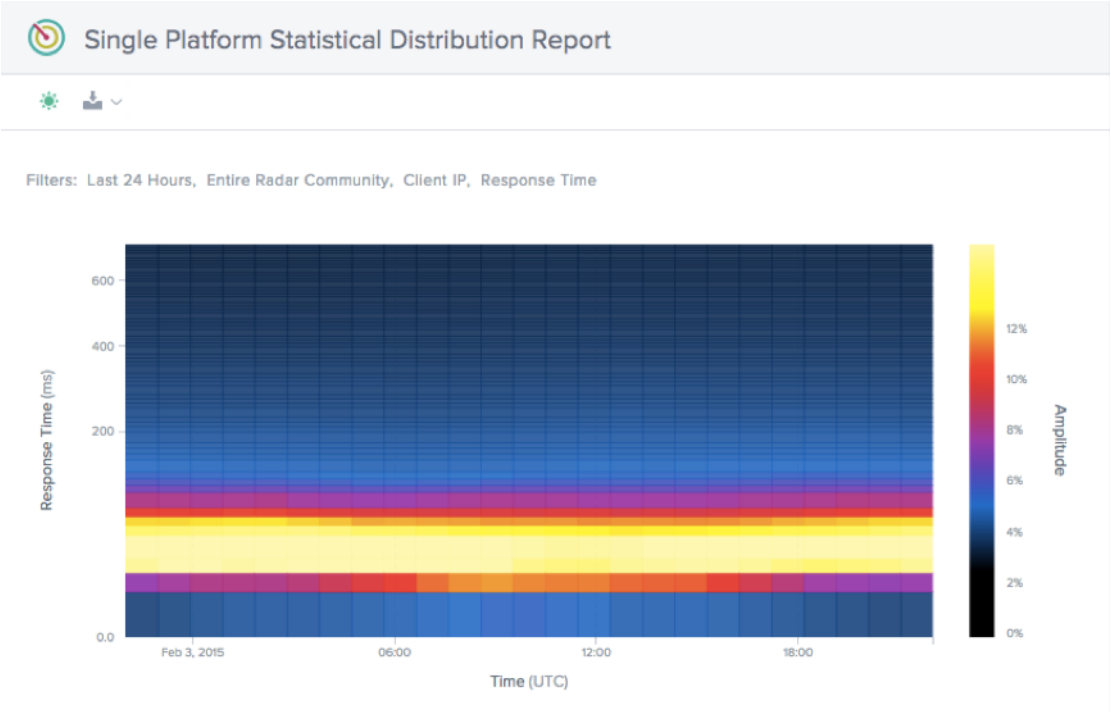
Single Platform Geo Location Report

This report shows the distribution of Radar traffic by country over time for a single platform at a time.



Single Platform Statistical Distribution Report

This report shows the distribution of Radar traffic over time by response time.

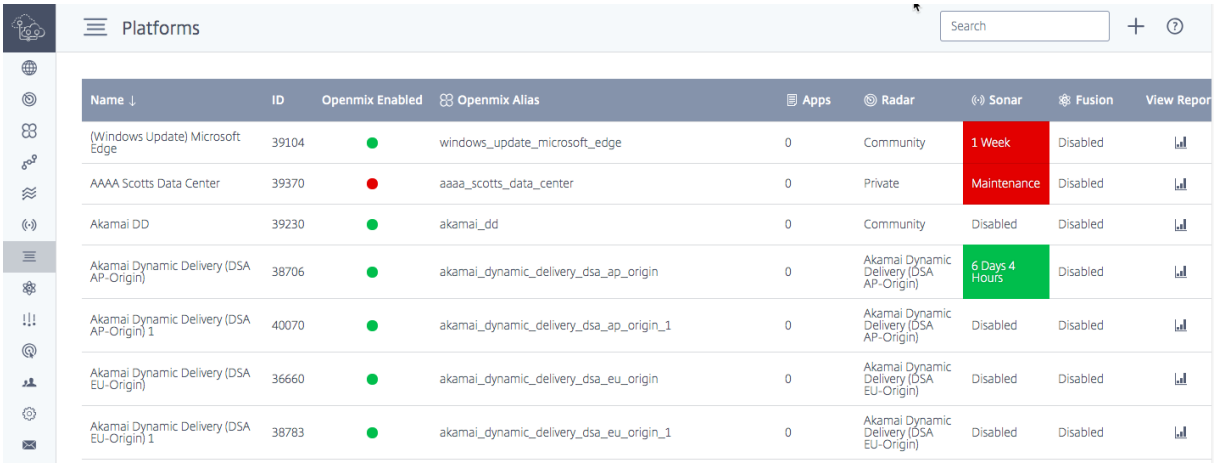


Platforms

December 17, 2020

The **Platforms** page is where the customer specifies the CDNs, clouds, data centers, or other end-points that must be monitored and used with Openmix. A platform must be set up for each routing end-point on which you would like to report. Most often a platform represents a CDN, cloud region, or individual instance, if using Openmix for GSLB.

On clicking this menu item, the customer is presented with the following screen.



Name ↓	ID	Openmix Enabled	Openmix Alias	Apps	Radar	Sonar	Fusion	View Repor
(Windows Update) Microsoft Edge	39104	●	windows_update_microsoft_edge	0	Community	1 Week	Disabled	
AAAA Scotts Data Center	39370	●	aaaa_scotts_data_center	0	Private	Maintenance	Disabled	
Akamai DD	39230	●	akamai_dd	0	Community	Disabled	Disabled	
Akamai Dynamic Delivery (DSA AP-Origin)	38706	●	akamai_dynamic_delivery_dsa_ap_origin	0	Akamai Dynamic Delivery (DSA AP-Origin)	6 Days 4 Hours	Disabled	
Akamai Dynamic Delivery (DSA AP-Origin) 1	40070	●	akamai_dynamic_delivery_dsa_ap_origin_1	0	Akamai Dynamic Delivery (DSA AP-Origin)	Disabled	Disabled	
Akamai Dynamic Delivery (DSA EU-Origin)	36660	●	akamai_dynamic_delivery_dsa_eu_origin	0	Akamai Dynamic Delivery (DSA EU-Origin)	Disabled	Disabled	
Akamai Dynamic Delivery (DSA EU-Origin) 1	38783	●	akamai_dynamic_delivery_dsa_eu_origin_1	0	Akamai Dynamic Delivery (DSA EU-Origin)	Disabled	Disabled	

This screen shows a full list of all the platforms configured for either reporting, radar measurements or the Sonar and Fusion services.

The table shows:

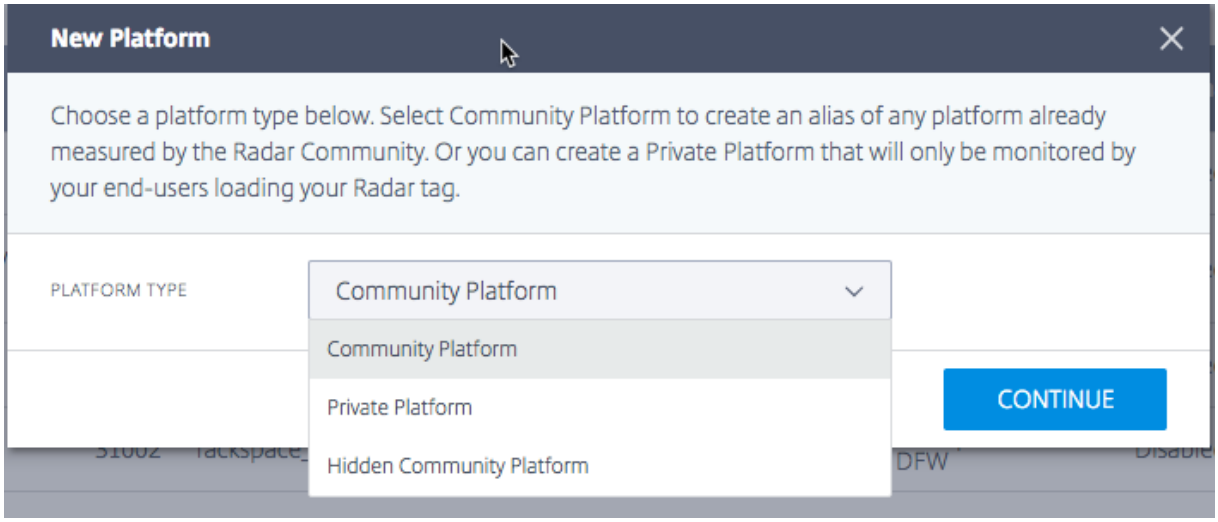
Heading	Description
Name	The user-defined name of the platform.
ID	The ID generated for the platform, useful for API access and reporting.
Openmix Alias	The alias with which to refer to the platform from Openmix applications.
Apps	The number of Openmix applications using the platform.
Radar	Whether the platform is set up to use community or private Radar measurements.
Sonar	Whether Sonar is activated on this platform.
Fusion	Whether Fusion is activated on this platform.

Creating a Platform

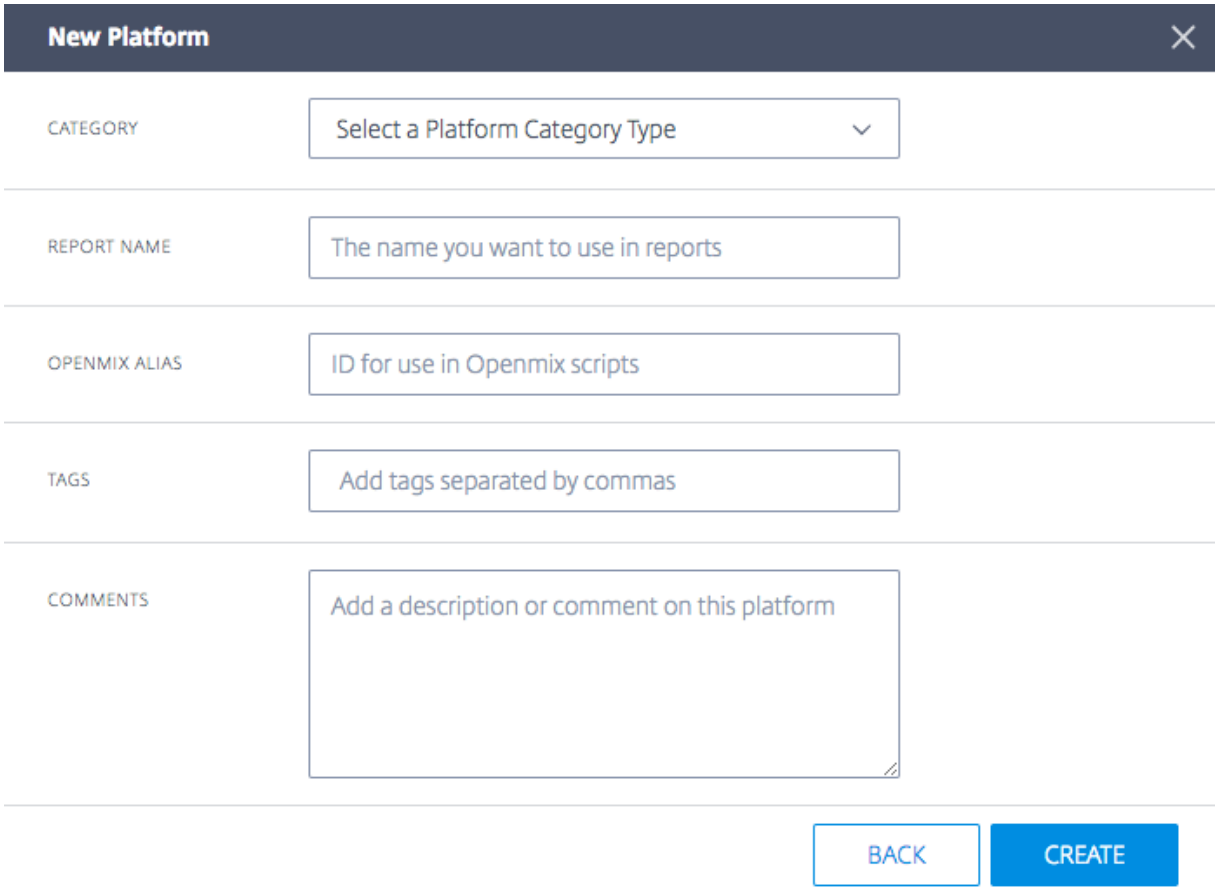
To add a platform, click the **+** button at the top of the Platforms page.

New Platforms

Once you have clicked **Add Platforms** you see the following page where you can select the type of platform you want to configure.



Once you select the **Platform Type**, you can provide a name for the platform that will be used to display information and used within other services that ITM provides such as Openmix.

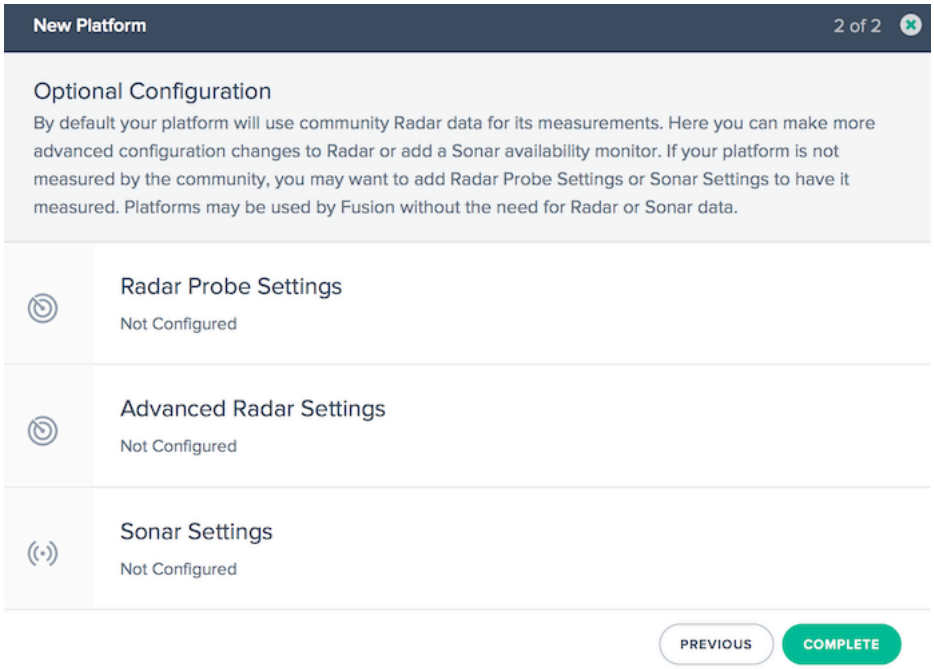


In **Platform settings**, enter the following information:

Input Item	Description
Category	The type of service the platform represents. Platforms are handled differently in Radar and Openmix, depending on the type. The available platform categories are - Cloud Computing, Dynamic Content, Delivery Networks, Cloud Storage, Secure Object Delivery, and Managed DNS. For **Private** platforms, one more category available is **Data Center**. Note: All imported GSLBs are created as data centers.
Platform	Select the Platform you want to test, for example Akamai, Amazon, Azure, and so on
Report Name	Name for the platform used in display and reporting.
Openmix Alias	The alias that Openmix applications use to identify the platform.
Tags	Tags can be assigned to platforms so they can organized as needed.

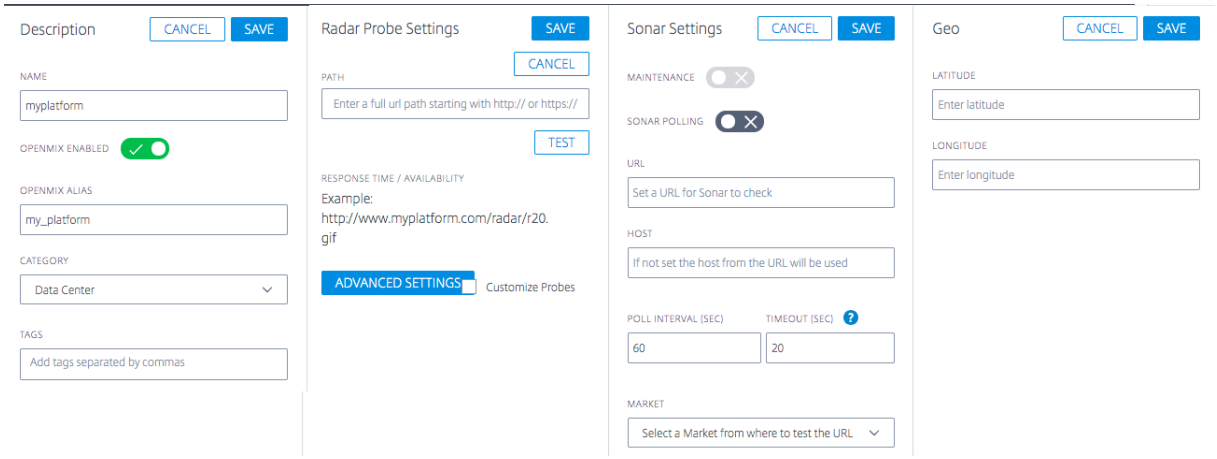
When you select an existing platform the **Report Name and Openmix Alias** fields are filled in. You can leave these fields with the default values or modify them as you prefer.

Click **Next** to continue to the optional configuration. When finished with the optional configuration, click **Complete** to add the platform.



Editing a Platform

Editing a platform is as easy as clicking the platform row in the table and clicking the **Edit** button.



Once you have changed the configuration, simply click **Save**, as you would with a new application and this will bring you back to the platforms screen with your changes saved.

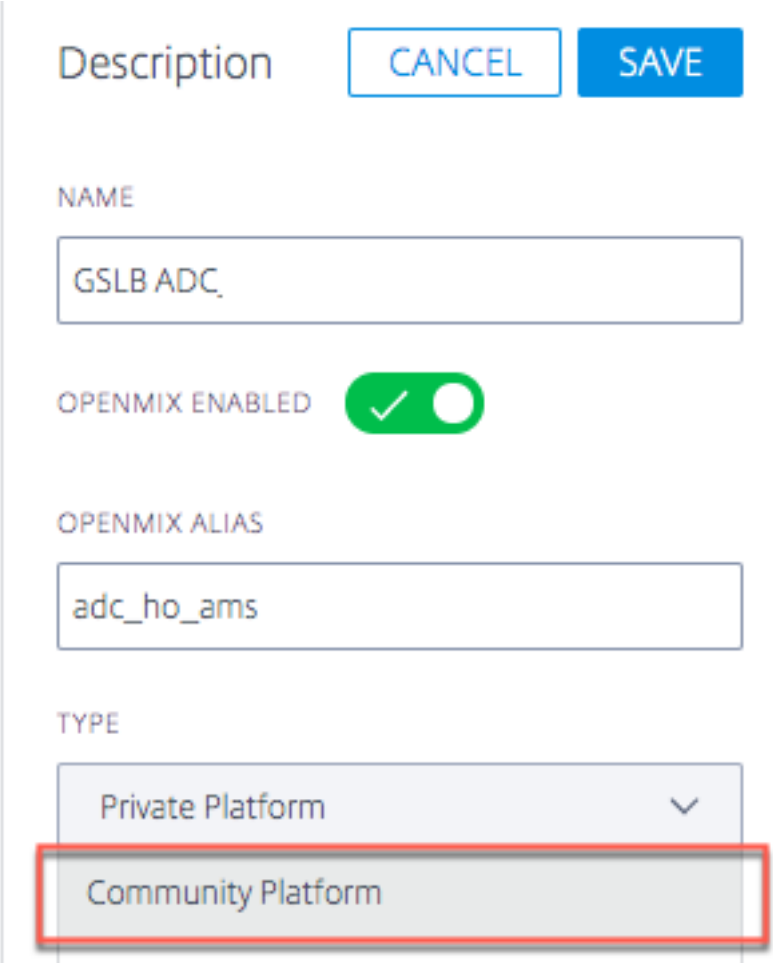
Change Platform Type

This feature is useful for customers whose private platforms are hosted in a public data center or cloud region that is measured by the Radar community (AWS, for example) and want to inherit the Radar data of that community platform. For example, when customers import GSLBs into the ITM portal, they are imported as private data centers but may actually be located in an public cloud region. To

inherit the Radar data of the community platform, customers can change the current settings of the private platform or GSLB to reference the community platform instead.

To change the platform type, such as a GSLB or private data center, to a public community platform (or from community back to private if required), do the following.

1. Click the platform row in the **Platforms** table.
2. In the **Platform Settings** section, click the **Edit** button.
3. Go to **Type**. Select **Community Platform** from the list if you want to change your private platform to a community platform.
4. Go to **Category**. Pick a platform category from the list.
5. Go to **Platform**. Select the platform that you want to change to from the **Platform** dropdown list.
6. Click **Save** on the top right of the **Platform Settings** section. You will see a confirmation message telling you that the Radar probe settings for your private platform will be removed and replaced with the settings of the community platform.
7. Click **Confirm**.



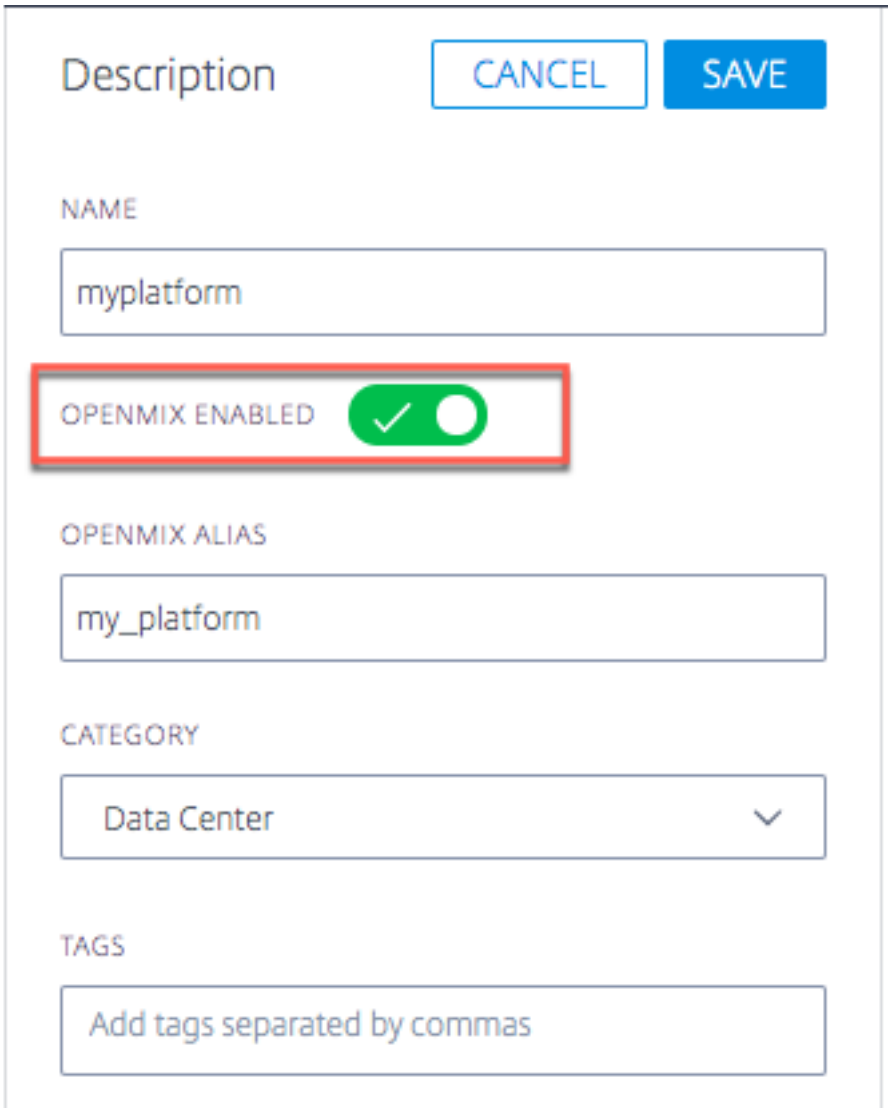
Note: If you decide to change back from community to your private platform, you would need to

reconfigure your Radar probe settings.

Enable Platform for Openmix

A Platform can be enabled or disabled for Openmix by switching the **Openmix Enabled** button on or off in **Platform Settings**.

- Click the **Edit Button** in **Platform Settings**
- Select the button for **Openmix Enabled** to switch it on.



If a particular platform is disabled for Openmix, that platform will no longer be considered in Openmix decisions. This means that a Radar score will not be generated for that particular platform.

In Quickstart apps, the platform (if disabled in the UI) will not show up as an option to be selected.

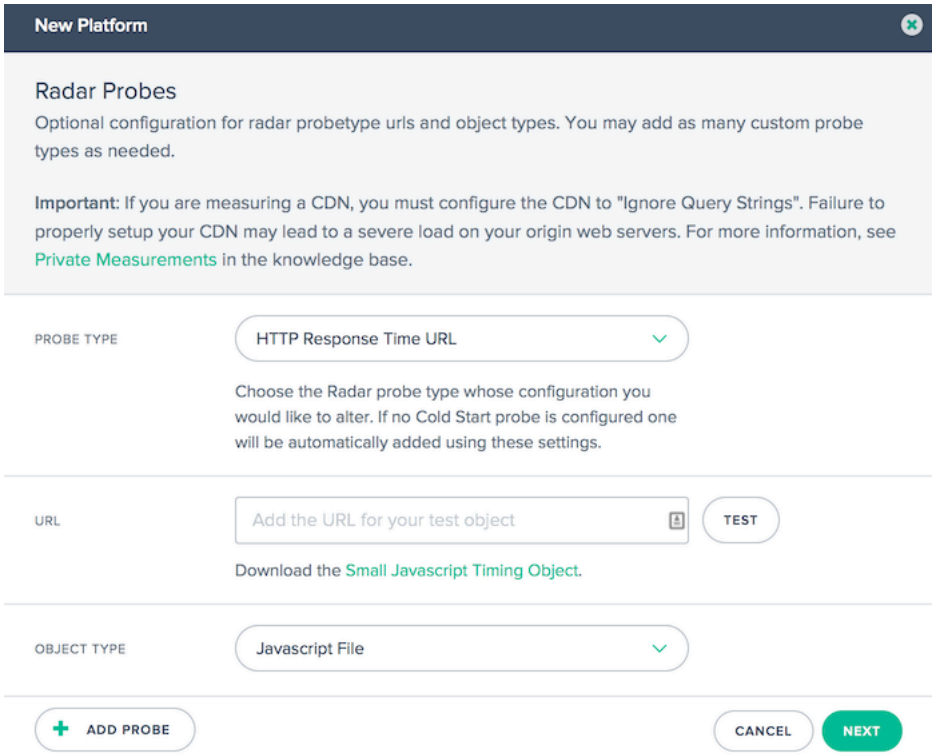
However, for Custom apps, if the platform is hard-coded into the app logic, there is a chance that it may

be picked up (even if that platform is disabled for Openmix in the UI). To avoid this from happening, the custom app must be written in such a way that it always includes a logic to pick up the Radar score. When the platform is disabled for Openmix (in the UI), there will no longer be a Radar score generated for it, and therefore it will be automatically ignored by the app.

This can be used as an operational on/off switch if there is an issue with a particular platform and the customer wants to pull it out of all apps during that issue.

Radar Probes Settings

Radar probes can be specified for each platform. Usually this is only necessary if you are setting up a private platform for Radar monitoring. Public platforms provide data gathered by the community and can be relied on for most uses.



There is a probe for each type of data collected, such as: HTTPS Response Time, HTTP Throughput, HTTPS Cold Start (for availability), and so on Most Radar setups have probes for at least Cold Start and Response Time, with Throughput in some cases.

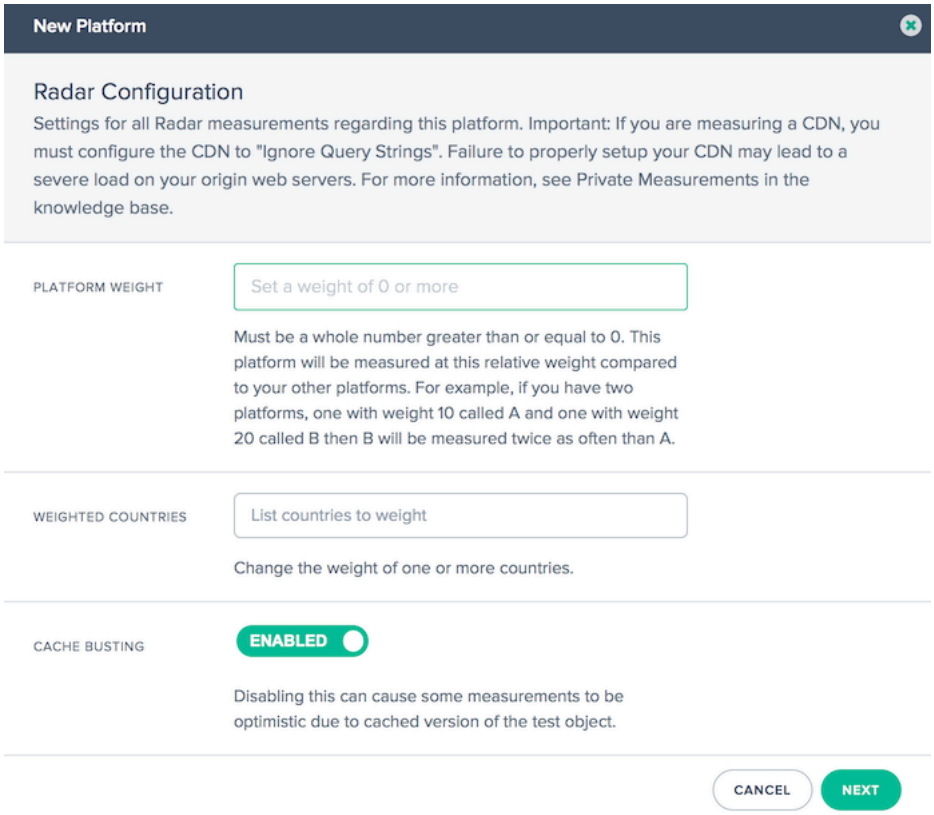
Each probe has the following settings:

Input Item	Description
Probe Type	The value for which the data should be reported. There are separate probes for each protocol (HTTP/HTTPS) and the type of data that will be collected (Cold Start, Round Trip Time, Throughput, and so on).
URL	The URL to the probe object.
Object Type	The type of file that is used to take the measurement. In most cases you want to download the “Timing Object” from the link in the dialog and choose “Image File”. For probes of DSA services, you will normally choose “webpage (Dynamic)”.

Click **Add Probe** in the lower left of the dialog and add information for each probe. Click **Save** after all the probes have been entered.

Advanced Radar Settings

You can control the behavior of the Radar checks for the platform. These should only be changed if you understand the impact on your Openmix application.



The following options are available:

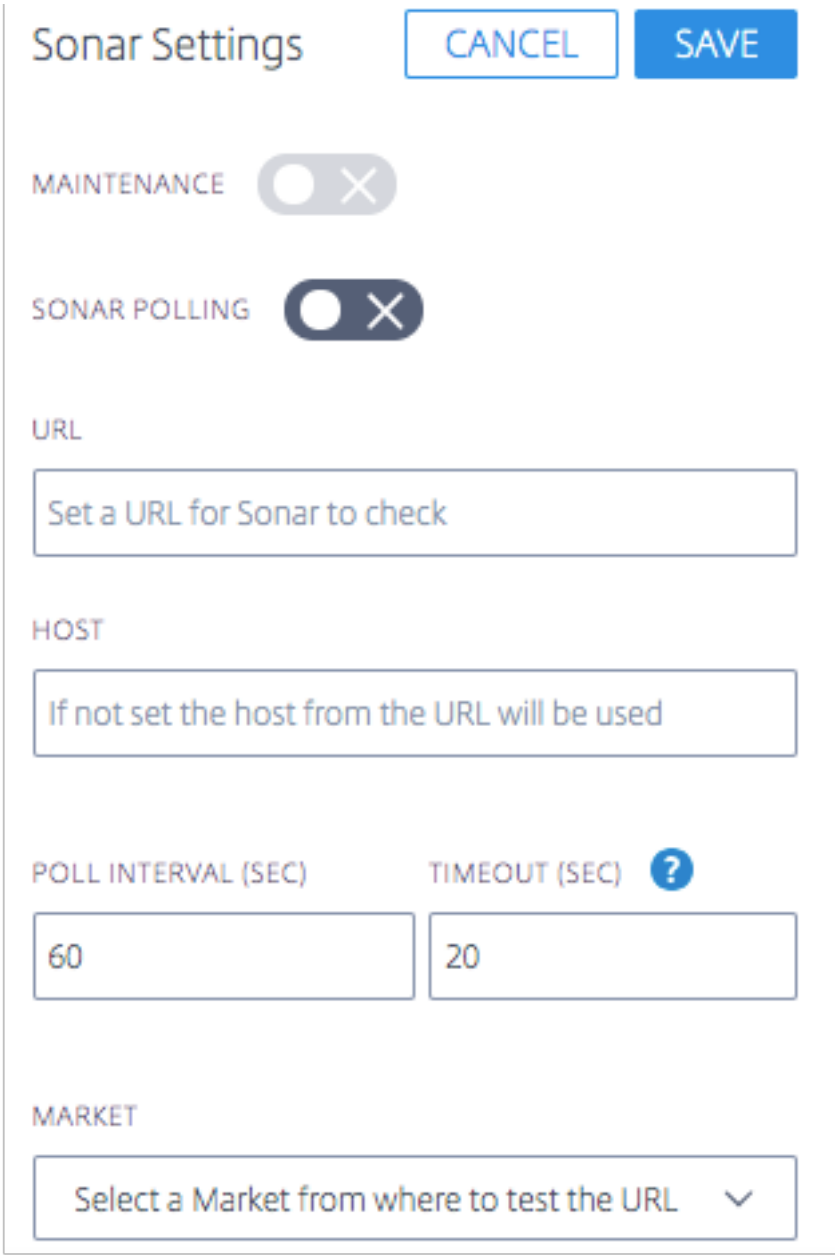
Input Item	Description	Default
Platform Weight	Radar uses a weighting system to help customers prioritize their custom tests, the higher the number the higher priority of this private test. Typically, this is used when you have several custom tests, if you are configuring only one leave it as the default.	10, no weighting
Weighted Countries	You can override the Platform Weight for certain countries by inputting the desired countries. The country is specified using the ISO country codes.	0, no weighting

Input Item	Description	Default
Country Weight	If Weighted Countries are specified, this weight is applied to the countries and will override the Platform Weight. If the weight is set to zero, the platform will not be measured in the specified countries.	
Cache Busting	Disabling this setting can cause some of the measurements to be optimistic due to cached versions of the test object being reported.	Enabled

Sonar Settings

Sonar is a liveness check service that can be used to monitor web-based services for availability. Sonar works by making HTTP or HTTPS requests from multiple points-of-presence around the world to a URL that you specify.

Sonar is enabled in the platform configuration. Please refer to the Sonar user guide for more information.



Platform Geo

The platform **Geo** is a location (latitude and longitude) assigned to a platform. Geo information is what enables you to place platforms accurately on the map in the **Visualizer** tool.

Note: The **Geo** only applies to platforms that have one physical location such as data centers or cloud regions.

For Private Platforms

By default there is no **Geo** location assigned to private platforms. When a user creates a private platform, and configures a **Radar** probe, we use the probe to geo locate it. This means that when you add a URL to the **Radar** settings, we geo locate the IP we get back, and assign that as the **Geo** for the private platform. You can edit this **Geo** if necessary. Alternatively, you can assign a **Geo** manually for your platform without relying on the Radar URL path.

Once the **Geo** is set, it does not reset by itself. Even if you change the **Radar** URL, it does not change the **Geo** of the platform. You need to edit the **Geo** manually to modify it.

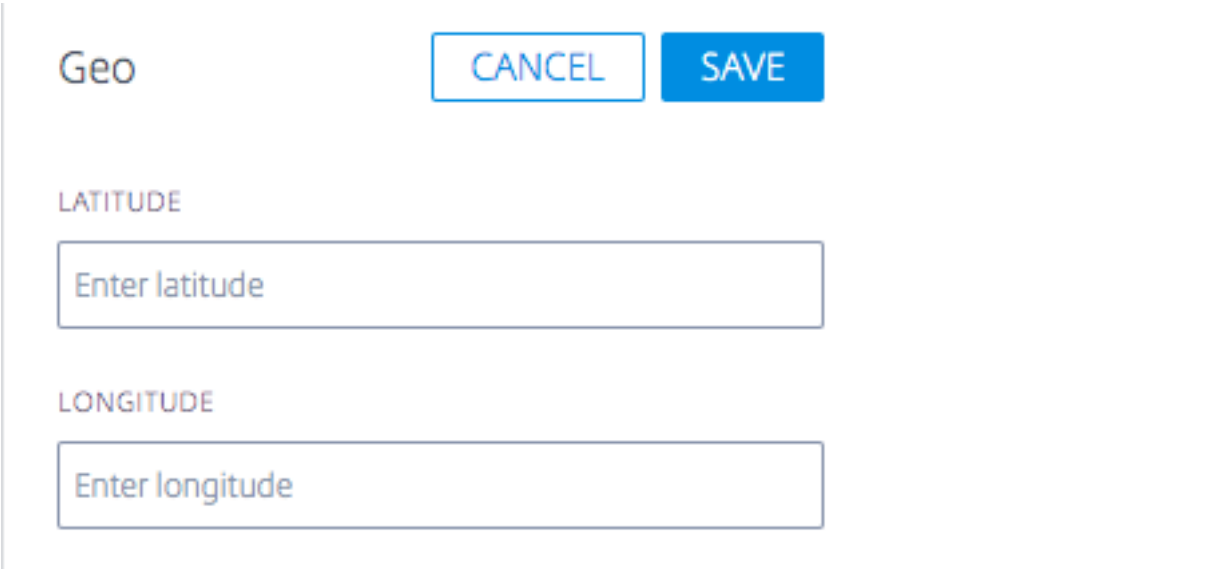
Note: Not all private platforms get a **Geo** value assigned. Geos only apply to platforms that have one physical location.

For Imported Platforms

If you import a platform via a GSLB or F5 config, we geo locate the public IP from that config and use that as the **Geo** of the platform.

For Community Platforms

When a customer adds a community platform to their account, by default this platform inherits the original geo of the **Community Platform**. However, the geo of this platform can be edited by the customer. Normally, a customer must not have to edit it. However, if a customer chooses to edit this **Geo** and enters a new latitude and longitude, then the customer's setting (for the community platform) would override the original **Geo** of the **Community Platform**.



Openmix

August 30, 2022

Overview

Citrix Intelligent Traffic Management (ITM) Openmix provides a revolutionary approach to Global Traffic Management/Global Server Load Balancing (GTM/GSLB). For traditional global traffic management, ITM provides a DNS-based approach to Load-balancing. ITM uses DNS CNAME or records where DNS responses are altered in real-time based on the required business logic. Openmix can be integrated into the video workflow and delivery in multiple ways.

GTM or GSLB tools and services rely on proprietary, inextensible, static rule engines to define, and control a narrow set of fixed policies for failover, round-robin, and geo-targeting. Citrix ITM's mission is to enable next-generation cloud strategies based on real-time data feeds. The Openmix platform provides a highly robust means to ingest real-time data from various sources. It exposes the metadata as environment "variables" which can be evaluated on each request.

Openmix: Key Benefits

- Eliminate single-vendor dependencies and ensure 100% availability
- Control price/performance trade-offs and banish headaches associated with multi-sourcing
- Remove uncertainties of legacy performance tools and offload traffic selectively and strategically
- Apply specific providers to target individual markets

How Openmix Works

Customers log into the Citrix ITM Portal to deploy their first application. A library of sample apps is available to help get started and a step-by-step wizard tool to help create applications with the most common routing logic. ITM Openmix applications can support two protocols for directing traffic: DNS or HTTP.

Application Defined Control

The globally distributed, on-demand, Openmix platform moves GTM/GSLB decision making close to your application audiences. Each host can have its own custom defined Openmix application that considers current metrics and variables that provide the best optimization for any routing request.

Openmix scripts are programmed in JavaScript, a language that is accessible to most web programmers and network administrators. While this script-based approach is where virtually any business

logic can be implemented with minimal coding complexity to use as the basis for truly dynamic traffic management policies. Thanks to the collaborative nature of our customer community, ITM also provides “quick start apps” which are standard applications that don’t require code.

When to Use HTTP or DNS Services

ITM Openmix enables a wide range of content delivery optimization. Which method you use to enable Openmix largely depends on the specifics of your use case. The DNS method is easy to implement, mostly transparent to clients, and usable across a wide variety of content. However, the ability to switch providers is limited by the TTL set on the DNS response and some content cannot be switched to a different provider mid-stream. HTTP provides more integration flexibility and optimization decisions can be made when it is optimal for the client. That more flexibility requires more work to integrate with a CMS or client.

The following table summarizes the customer use case for the DNS and HTTP interfaces.

	Openmix DNS	Openmix Web Services (HTTP)
Typical Use	Webpage Optimization Mobile App Optimization Player or Game Download Initial Video/Game Request Mid-Stream Requests (TTL expiration)	Initial Video Request Initial Game Server Selection Mid-Stream Requests Mid-Play Gaming Client Requests
Radar Tag / SDK & Fusion Data Collection	Cedexis Radar RUM CDN & Cloud Performance Monitoring CDN & Cloud Costs data, 3rd Party Monitoring Metrics: Player, Server or App Health, Synthetic Process Monitoring, etc.	
Client Data Collection	Video Player Performance Metrics	
Cedexis Billing	Per Millions of DNS Queries	Per Millions of HTTP Requests

Openmix: DNS

CNAME Delegation

The easiest integration for ITM customers is to use DNS CNAME delegation. The CNAME delegation works by having the customer point their end-user facing host name (in the following example, `www.acme.com`) at an ITM host name

```
1 www.acme.com  600  IN  CNAME  2-02-123d-000d.cdx.cedexis.net.
```



```
<!--NeedCopy-->
```

On receiving a DNS request from an end-user the ITM system makes a real-time decision. The decision is based on the Radar data, business logic in the application and any third party information. This decision is articulated either as another CNAME record (in our example below acme.cdn1.net) or as an A record such as 111.222.111.222.

By supplying a CNAME record ITM "points" the end-user to the CDN, Cloud, or Data Center of choice. Routes the end-user to use that provider versus another.

```
2-02-123d-000d.cdx.cedexis.net.  19  IN  CNAME acme.cdn1.net.
<!--NeedCopy-->
```

Once the CDN or Cloud CNAME is supplied the end-users machine continues the resolution chain. It requests a CDN name server, until an IP address of the node or server is received. Where upon the process of downloading content begins.
If a record is supplied as part of the logic the end-users machine receives the IP address. It connects directly to the server and initiates the download of content.

```
acme.cdn1.net.  132  IN  A  111.222.222.111
<!--NeedCopy-->
```

Zone Delegation

In addition, Authoritative DNS Zone Delegation is an option to implement Openmix. The customer creates a DNS zone and delegates to a Predictive DNS zone created in the ITM Portal. Create a host name in the delegated zone. Configure it to use an Openmix application or a dynamic Predictive DNS record to generate a response.
The advantage of this option is that there doesn't need to be a CNAME delegation between the host name and the dynamic response from the ITM platform. Using the preceding example, `www.acme.com` the host name is directly resolved to the configured value for the optimal CDN, Cloud, or Data Center.

```
www.acme.com. 19 IN CNAME acme.cdn1.net.
```

A/AAAA records can also be used instead of CNAMEs, and the host name resolves directly to the record of the optimal destination.

```
www.acme.com. 19 IN A 111.222.222.111
```

DNS and Time To Live Implications

Factors such as Time To Live (TTL) values are considered carefully with an appropriate time set for the content and how the decision making must be for users. In most instances ITM recommends a 20 second TTL for page and object content. For video content the ITM consultant works with the customer to find the most appropriate balance based on chunk length and integration method.

Openmix: HTTP

An alternative to DNS is to use the HTTP API. Openmix uses HTTP requests to inform a client such as a video player or CMS on which platform to use at any one point in time.

```
http://hopx.cedexis.com/zones/1/customers/0/apps/1/decision
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Mon, 22 Apr 2015 20:25:24 GMT
< Connection: keep-alive
< Content-Length: 177
<
{

  "providers" : [
    {

    "provider" : "cdn2",
    "host" : "foo.cdn2.net"
     }
 ,
    {

    "provider" : "cdn1",
    "host" : "acme.cdn1.net"
     }

  ]
 }

<!--NeedCopy-->
```

The HTTP Openmix service uses the same application logic as its DNS based counterpart. It also includes some additional extensions, allowing further profiling of a client machine. For example, with HTTP Openmix it is possible to look at the headers for User-Agent String, X-Forwarded-For, and Referer. Supply IP overrides using query string parameters.
As the payload for HTTP Openmix is more extensible than DNS it is also possible to provide the CDN, cloud or server decision selection in different ways. The most common so far has been an ordered list

from most preferred platform to least (as above). A full list allows the decision rank to be supplied to the CMS or Client, yet still allows for internal heuristics to be used in choosing the provider.

CMS Integration

Some customers prefer to handle the provider selection on the server-side rather than implement provider selection in every client. The HTTP API can be used to retrieve an optimization decision from Openmix at request time from the client. It can be used to populate a file that is returned from the CMS to the client.

By default, Openmix HTTP endpoints use the IP of the caller for geo location and decision criteria. If you are calling from a CMS or other system that sits between the end-user client and Openmix, you can specify IP as a parameter to use in the decision.

```
http://hopx.cedexis.com/zones/1/customers/0/apps/1/decision?ip=1.2.3.4
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Mon, 22 Apr 2015 20:25:24 GMT
< Connection: keep-alive
< Content-Length: 177
<
{

  "providers" : [
    {

    "provider" : "cd1",
    "host" : "acme.cdn1.net"
     }
 ,
    {

    "provider" : "cdn2",
    "host" : "foo.cdn2.net"
     }

  ]
 }

<!--NeedCopy-->
```

This method allows you to use a CMS integration to pull decisions from Openmix. You can also get the benefits of geo and ISP route optimization for the end user. The host name returned from Openmix is then packaged into the response, such as a video manifest file, and returned by the CMS to the

client. The client uses the optimized decision without needing any modification to support Openmix optimization.

Openmix Applications

Openmix Quickstart applications are load balancing and traffic management applications. These applications provide real-time traffic routing to the best provider based on a set of rules.

The applications are processed for each request made to Openmix and a routing decision is made based on the specified logic. A customer can have one application for content that has high business value, and a different application for content that has less value. These requests are separately routed.

When you invoke an application, a single request goes to one of Citrix's load-balancers. For DNS, it is a single DNS request to the DNS load-balancers. For HTTP, it is a GET or HEAD request to the Openmix HTTP endpoint.

The following apps are currently available through the Intelligent Traffic Management Portal.

- Static Routing
- Failover
- Round Robin
- Optimal Round Trip Time (ORTT)
- Throughput
- Static Proximity

Openmix Custom JavaScript applications are used by specialized Openmix servers to respond to DNS or HTTP requests based on the logic in the scripts. Deployment of the scripts is done via the customer portal where the app is configured and published. For more information on the ability to create your own JavaScript scripts, refer to the information at our Developer Exchange.

Before you go ahead with setting up the apps, it is important to understand the following concepts:

Availability Threshold

The availability threshold is the minimum availability score that a platform must meet to be considered for routing. The default minimum availability threshold for all applications is 80%. However, you can modify this percentage and set it to a value that is appropriate for your location, network availability, and reliability.

Note: If no platform meets this minimum Availability threshold (the default of 80%, or the value that you set), random routing is done for Round Robin, ORTT, and Throughput applications.

Fallback

The fallback response is returned if the Openmix application does not run successfully for whatever reason. Or if Sonar confirms that there are no available platforms. Therefore, a valid fallback CNAME/A/AAAA record or IP (or path in HTTP) must be specified with which Openmix can respond. This fallback URL or CNAME record can be for a platform that is pre-configured in Openmix.
Fallback sometimes occurs during the following scenarios as well:

- When you switch between versions of your application, you upload and publish a new script. There is a brief millisecond time period of fallback until the new script initializes and the old one is removed.
- If ever there is an overload (which rarely happens), Openmix responds with the fallback CNAME/A/AAAA since the fallback offsets the load on the service.

For fallback, you must enter a valid host name (CNAME/A/AAAA record) or IP address in DNS, and a valid URI (it can be of the form, `scheme:[//host[:port]][/path][?query][##fragment])` in HTTP.

TTL

In Openmix, the DNS Time to Live (TTL) for the application tells resolvers how long they must keep the decision before asking Openmix again.
The TTL is used to Control the volume of traffic that an Openmix app gets. It also controls how sensitive an app must be to changes in the data that it acts upon.
The default TTL is 20 seconds. Although you can modify this value, it is not recommended to do so. If you lower the TTL you get more volume and more real-time DNS queries. It can lead to added costs and lower performance because DNS queries take time on the client. Therefore, it is best not to change the default value of TTL.

> **Note:** The Time to Live applies to Quickstart apps, Custom JS Apps if no TTL is specified in the code, and for all fallback responses

Weights (Used for Round Robin)

You can assign weights for the prioritization and selection of each platform globally and/or by market or country.

For example, say you have three platforms selected for your application - P1, P2, and P3. You give them the weights: 60, 50, and 10 respectively. The Round Robin app converts these values to percentages such as, P1=50%, P2=42%, and P3=8%, that adds up to a 100%. These percentages mean that 50% of the time, users are routed through P1, 42% of the time through P2, and 8% of the time through P3.

The weights you give to the platforms do not have to add up to 100. They can be any integer between 0 and 1,000,000. The weights given to the platforms when converted to percentage (by the app in the

back-end), adds up to a 100%. If all selected platforms are given the same weight, traffic will be evenly distributed across them over time. If you have one platform, then that platform is used 100% of the time, regardless of the weight you give it.

Weights are only used for platforms that are considered to be available according to Radar and Sonar availability checks, depending on the configuration of the application. Unavailable platforms cause the distribution to not match the configured weights. For example, if P1 weighs 100 and P2 weighs 0 but P1 fails the Radar Availability check, then all traffic goes to P2.

Handicap (Used for ORTT and Throughput)

The **Handicap** is a percentage value that can be applied to a platform to modify the radar scores for RTT and throughput that is, artificially increase the response time (in milliseconds) or decrease the throughput (in kbps). Increasing or decreasing these values bring down the performance of the platform such that the likelihood of it being picked becomes low. Handicaps can be added to platforms globally, or separately for specific markets or countries.
In cases where one platform is expensive in a specific market or country and you want to reduce its likelihood of being picked when an equivalent provider is close in terms of performance. You put a Handicap value as a multiplier to increase the value of response time or decrease the value of the throughput. As a result, it lowers the probability of a platform being picked.

Following is roughly how **Handicap** works in the backend:

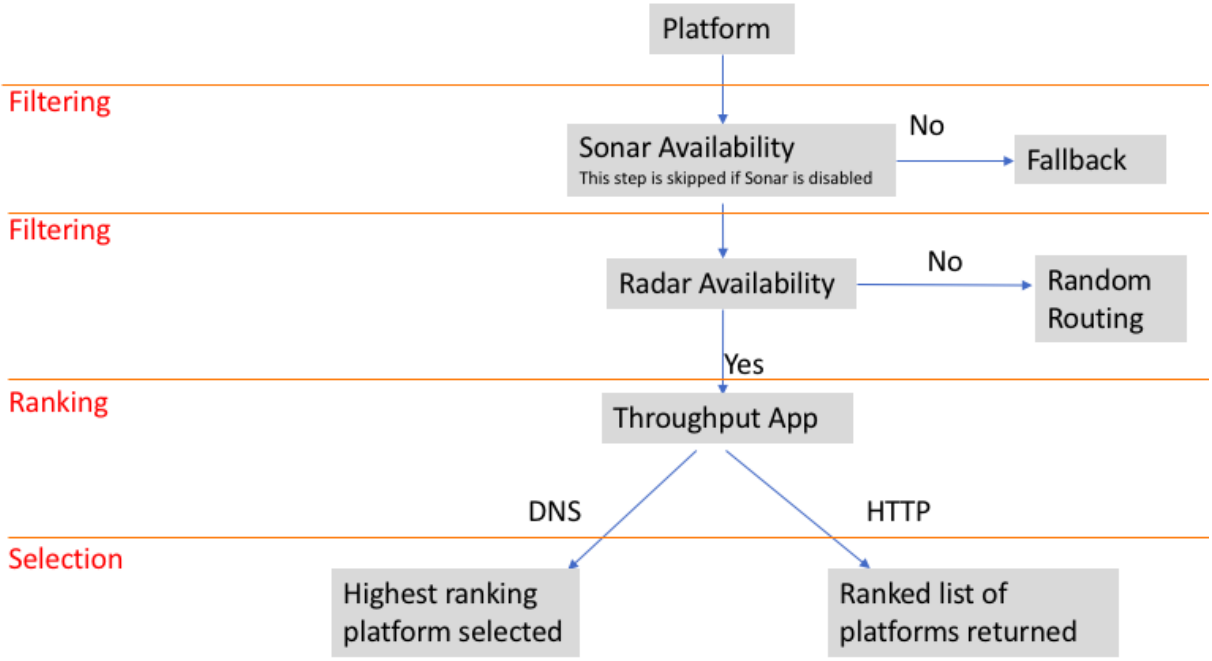
- Platform RTT with Handicap applied = RTT (Round Trip Time in milliseconds) *(1 + Handicap)
  or
- Platform Throughput with Handicap applied = (Throughput in kbps) *(1 – Handicap)

Note: The RTT and Throughput values for the platform are scores from Radar data.
The following table shows how Handicap effects the two platforms - P1 and P2. And how the Handicap decreases the likelihood of P1 being picked.

	P1	P2
RTT without Handicap	50 milliseconds	60 milliseconds
RTT with 50% (0.5) Handicap for P1, and 0% (0) for P2	50(1+0.5)= 75 milliseconds	60(1+0)= 60 milliseconds
Throughput without Handicap	3000 kbps	2800 kbps
Throughput with 50% (0.5) Handicap for P1, and 0% (0) for P2	3000(1-0.5)= 1500 kbps	2800(1- 0)= 2800 kbps

Filtering, Ranking, and Selection Workflow

Sample Flow diagram for Throughput App



Platform Selection Criteria

Openmix Quickstart apps use the following criteria as 1st, 2nd and 3rd level filters to rank and select the best platform.

Filtration Level	Selection Criteria	ORTT	Throughput	Round Robin	Failover	Static Routing	Static Proximity
1st level	Sonar Availability Check (if enabled)	X	X	X	X	X	X
2nd level	Radar Availability Check (if enabled)	X	X	X	X	X	NA

Filtration Level	Selection Criteria	ORTT	Throughput	Round Robin	Failover	Static Routing	Static Proximity
3rd level	Weights (user defined)	NA	NA	X	NA	NA	NA
3rd level	Round Trip Time (in millisec-onds)	X	NA	NA	NA	NA	NA
3rd level	Throughpu (in kbps)	NA	X	NA	NA	NA	NA

Reason Code Reporting

Reason codes provide visibility into why the decision was made and also get to know what part of the app's code is run. During execution, an app can add something to the reason code field at any time. Reason codes mean different things for each Quickstart App. There is some commonality between the reason codes for each app, but it is not comprehensive.

Note: For reason codes to be displayed correctly, they must not exceed the maximum character limit of 200 characters. If this limit is exceeded, the reason code is displayed as **Unknown**. If the user has not added a reason code, it displays **Unknown**.

The following are the reason codes for Quickstart Apps:

Reason Code	Description	Optimal RTT	Round Robin	Static Routing	Throughput	Static Proximity	Failover
Optimal Avail	The best perform-ing provider is available and has been selected.	X	N/A	N/A	X	N/A	X

Reason Code	Description	Optimal RTT	Round Robin	Static Routing	Throughput	Static Proximity	Failover
Optimal Unavail-Radar	The best performing provider is unavailable; another eligible provider has been selected which is available according to radar	X	N/A	N/A	X	N/A	X
Optimal Unavail-Radar+Son	The best performing provider is unavailable due to radar and/or sonar.	X	N/A	N/A	X	N/A	X
All Unavail-Radar	All eligible platforms are unavailable according to radar. Request routed to fallback	X	X	N/A	X	N/A	X

Reason Code	Description	Optimal RTT	Round Robin	Static Routing	Throughput	Static Proximity	Failover
All Unavail-Sonar	All eligible platforms are un-available according to sonar. Request routed to fallback.	X	X	N/A	X	N/A	X
Data Issue	Denotes missing radar measure-ments for one or more plat-forms. The platform is chosen randomly as a result	X	X	N/A	X	N/A	X
Geo Default	The default Geo settings are in effect	X	X	N/A	X	X	X
Geo Override-Country	A country override is in effect for this decision	X	X	N/A	X	X	X

Reason Code	Description	Optimal RTT	Round Robin	Static Routing	Throughput	Static Proximity	Failover
Geo Override-Market	A Market override is in effect for this decision	X	X	N/A	X	X	X
All Avail	All eligible platforms are available via sonar and radar	X	X	N/A	X	N/A	N/A
Proximal Avail	The closest geograph-ically platform is available and has been selected	X	N/A	N/A	N/A	X	N/A
Eligible Unavail-Radar	For Round Robin, the eligible provider is not available according to radar	N/A	X	N/A	N/A	N/A	N/A

Reason Code	Description	Optimal RTT	Round Robin	Static Routing	Throughput	Static Proximity	Failover
Persistent app	The decision served a cached response, no logic executed	X	X	X	X	X	X
Request Geo Un-available	The request's geo cannot be estab-lished. Request routed to fallback	X	N/A	N/A	N/A	X	N/A
All Unavail-Provider	All providers are un-available. Request routed to fallback	X	N/A	N/A	N/A	X	N/A
Unavail-Provider-Dist	No proximity scores have been found for any provider. Request routed to fallback	X	N/A	N/A	N/A	X	N/A

Openmix Quickstart Applications

1. Log in to the Intelligent Traffic Management Portal.
2. From the left navigation menu, navigate to **Openmix > Application Configuration**.
3. If you are configuring your Openmix app for the first time, you see the **Get Started** page, when you click **Openmix > Application Configuration**.
4. To configure a new app, either click the **Get Started** button or the **Add** button on the top right corner of the page. If Openmix apps have been configured previously, you see a list of apps on this page.

The following sections walk you through the process of configuring Openmix apps in the portal.

Static Routing

This type of application does not use any evaluative logic to decide which DNS response must be provided to the end-user. The app always selects a single platform here, specified by the user. Therefore, the app uses only a single DNS CNAME or IP address response. The Static Routing application can be configured through the portal on the **Application Configuration** page.

Note: Before configuring your application, ensure that your platforms are first configured. See Platforms page for platform configuration.

Navigation

1. Navigate to **Openmix > Application Configuration**.
2. Click the **Add** button on the top right

The **Basic Information** dialog box opens.

Basic Information

Follow these steps to enter **Basic Information**:

1. For **Protocol**, select DNS or HTTP from the list.
2. For **Application Type**, select Static Routing. Or if you're configuring another type of app, select it from the list.
3. Give a **Name** to your application (required field); add a **Description** (optional field); and a **Tag** (optional field).
4. Click **Next** for **Configuration**.

Configuration

To configure the app, do the following:

1. Select the associated platform from the **Platform** list. It is the platform that you set up within the Platforms page, representing the CDN, Cloud, or Data Center.
2. Enter a **CNAME/A/AAAA** record (for DNS) or **URL** (for HTTP). The DNS CNAME or HTTP URL for the selected platform must point to a valid IP address or host name.
3. For **CORS**, in an HTTP protocol select None, All, or Custom for CORS. CORS allows you to control access to your site from other sites. You can either completely restrict access to your site from other sites (by clicking **None**), allow access from all other sites (by clicking **All**), or allow access only from specific sites (by clicking **Custom**).
4. Enter a **TTL** (Time-To-Live) for the response. The default is 20 seconds but it can be overridden.
5. Click **Complete**.
6. In the confirmation pop-up click **Done** or **Publish** to see your app listed in the Openmix applications page. If you click **Publish**, your app goes live instantly and have a green status. It means that the application is in production. If you click **Done**, your app is still listed on the applications page but it's unpublished, and the status is red.

Failover

The Failover application supports a simple routing logic where a platform is chosen based on its place in line, and its availability. The customer can create a failover chain that decides which platform to select first, second, and so on. This failover chain can be created to either work globally or for individual markets and countries.

The **Failover** application can be configured within the portal on the **Application Configuration** page.

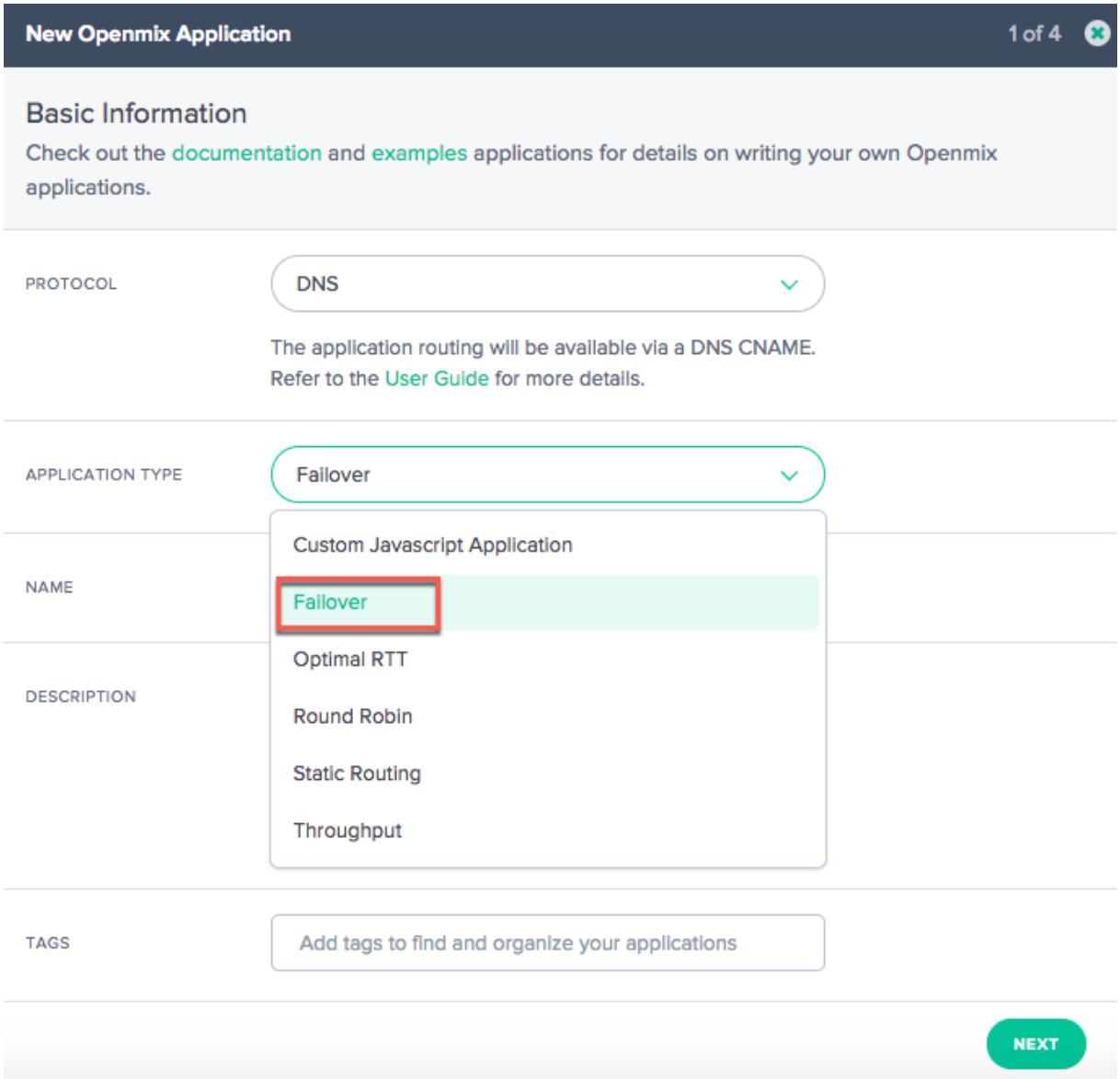
Note: Before configuring your application, ensure that your platforms are configured first. Refer to the Platforms page for platform configuration.

Navigation

1. Log in to the Portal.
2. From the left navigation menu, navigate to **Openmix > Application Configuration**.
3. Click the Add button on the top right to get to the New Openmix Application, **Basic Information** dialog box.

Basic Information

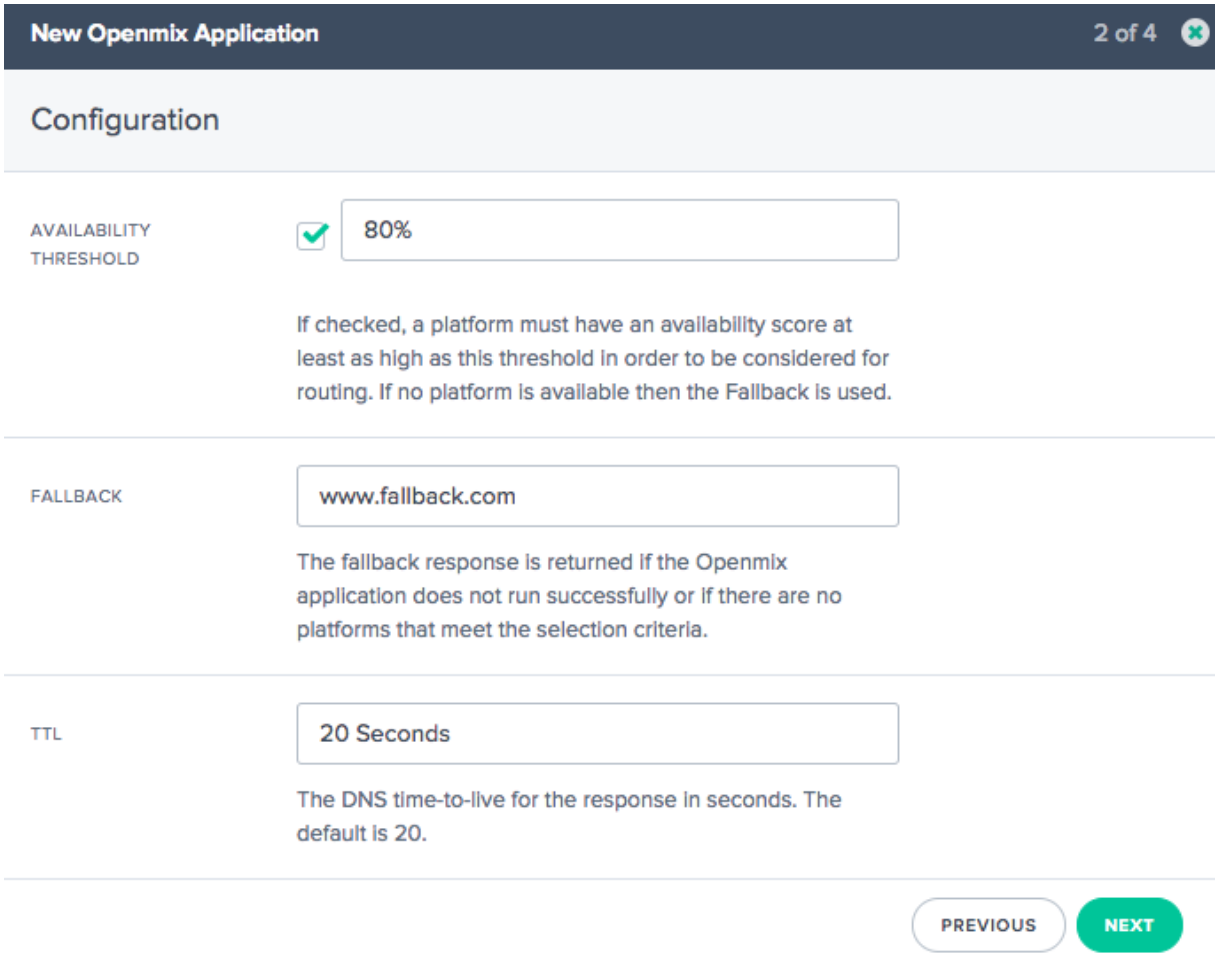
1. Select **DNS** from the **Protocol** list.
2. From the **Application Type** list, select **Failover**.
3. Give a **Name** (required field) to your application; add a **Description** (optional field); and a **Tag** (optional field).
4. When done, click **Next**.



Configuration

1. In the Configuration dialog box, select the **Availability Threshold** check box. The Availability Threshold has a default value of 80%. A platform must have an availability score at least as high as this threshold to be considered for routing.
    - If you want to modify the default availability threshold, just type in a new value to replace the default.
    - If no platform has an availability score that is equal or greater than the specified threshold, then the fallback CNAME or A or AAAA or IP address is used.
    - If the check box is unselected then the platform assumes a zero availability threshold. It means that there is no Radar availability check on this platform.

2. Enter a CNAME/A/AAAA or IP address for **Fallback**. The fallback CNAME/A/AAAA or IP is typically used if the application encounters issues, or errors.
3. Enter a **TTL** (Time-To-Live) for the response. The default is 20 seconds. You can override this value if necessary.



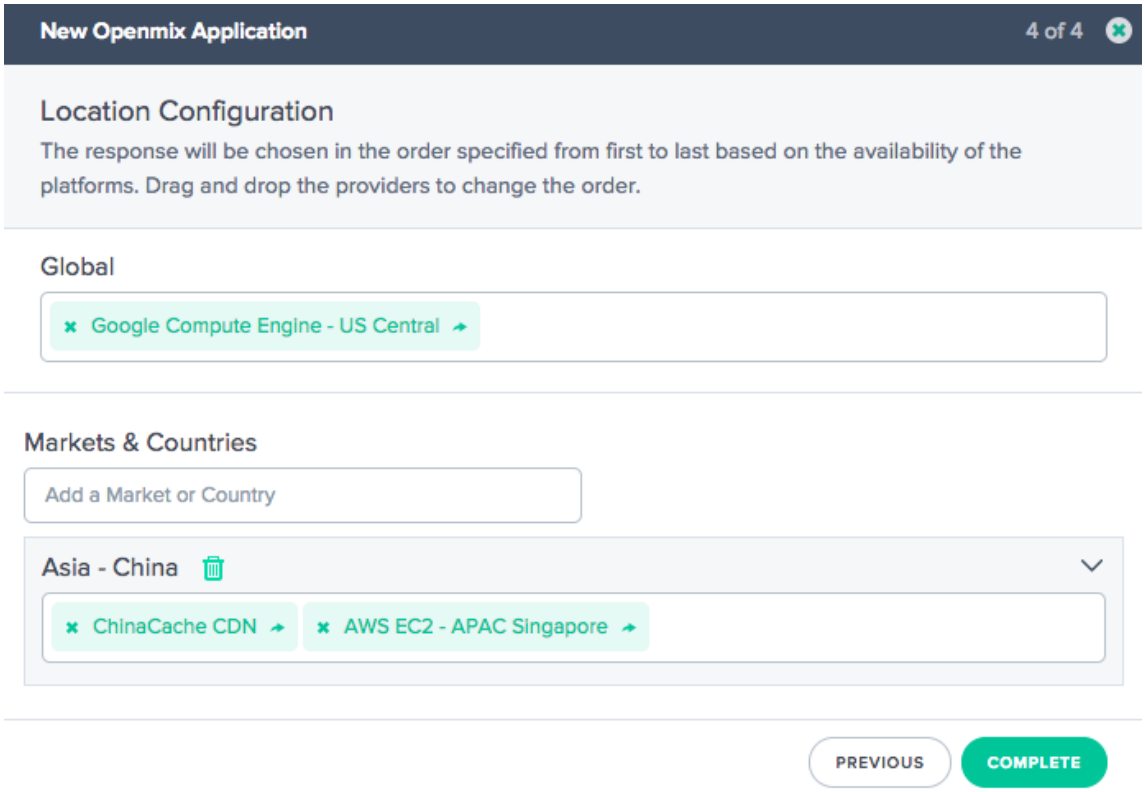
Platform Information

1. In the **Platform Information** dialog box, select a **Platform** from the list.
   - You can select multiple platforms using the **Add Platforms** button. The idea is to select all available platforms applicable for global and geo (markets and countries) routing.
   - The platforms on this list are the ones that you set up in the Platforms page within the portal, representing your CDN, Cloud, or Data Center.
   - All Openmix apps require an associated platform to be set up beforehand. If you don't find a platform in the list, you can set it up in the Platforms page within the portal.
2. Enter the **CNAME/A/AAAA** record for the platform.
3. Ensure that the **Enabled** check box is selected (indicating that the platform is enabled) before moving to the next step.

4. If **Sonar** is configured and you want to use Sonar data to help in the initial decision-making process, make sure to click the **Use Sonar for Platform Availability** check box. **Note**: The Sonar check box appears only if Sonar is enabled for that platform.
5. Click **Next** for **Location Configuration**.

Location Configuration

1. In the **Location Configuration** dialog box, select the required platforms for **Global** routing.
    - **Global** indicates that you are setting up a chain of platforms for global routing.
    - When you click inside the **Global** field, a list displays all the platforms you selected in the **Platform Information** step.
    - Select the required platforms from the list for availability-based global routing.
    - The order in which you place the platform names in this field determine the priority for their selection. For example, if the first platform on your list is unavailable, the second is selected. If none of the platforms on the list are available, then fallback is used.
    - You can drag the platform names to change their order of priority.
2. Click **Markets & Countries** if you would like to set up platforms for local geo routing.
    - When you click inside the **Markets & Countries** field, the list displays all the platforms you selected in the **Platform Information** step.
    - Select platforms for local geo routing, separately for each geo (market/country).
    - The order in which you place the platform names in this field determine the priority for their selection. For example, in China, you want to use the China POP first, and only if that is not available, you would want your Singapore POP to be used, which you would place next in line, and so on.
    - You can drag the platform names to change their order of priority.



3. Click **Complete**, to finish configuring your app.
4. In the confirmation pop-up click **Done** or **Publish** to see your app listed on the **Openmix** page.
    - If you click **Publish**, your app goes live instantly and have a green status. Your application is in production.
    - If you click **Done**, your app is still listed on the Openmix page but it is unpublished, and the status is red.

Round Robin

This application follows a typical Global Server Load-Balancing methodology of Round Robin, where each CNAME alternates being returned to end-users, as DNS requests are made. It uses Sonar data (if Sonar is enabled) and **Platform Availability** threshold to evaluate the best platform for the requesting user then. Each platform is selected based on the Round Robin distribution methodology. For example, if platforms P1, P2 and P3 meet the availability threshold, the first request is routed to P1, second to P2, and third to P3. The fourth request is routed to P1 again, and so on.

To configure a new Round Robin app, click the **Add** button on the top right corner of the Openmix page. The **Basic Information** dialog box opens up.

Navigation

1. Log in to the Portal.
2. From the left navigation menu, navigate to Openmix > Application Configuration.
3. Click the Add button on the top right to get to the New Openmix Application, Basic Information dialog box.

Basic Information

1. In the Basic Information dialog box, select DNS as the Protocol for Round Robin.**Note**: For the Round Robin app, routing is available only via a DNS CNAME.
2. Select the **Application Type** from the list. Give the app a **Name** (required field), a **Description** (optional field), and a **Tag** (optional field).
3. Click **Next** for Configuration.

Configuration

1. The **Availability Threshold** has a default value of 80%. To modify this value, just type in a new value to replace the default.
2. Enter a CNAME/A/AAAA or IP address for Fallback. The fallback CNAME/A/AAAA or IP is typically used if the application encounters issues, or errors.
3. Enter a TTL (Time-To-Live) for the response. The default is 20 seconds but this value can be overridden if necessary.
4. Click **Next** for Platform Information.

Platform Information

1. Select a platform from the **Platform** list. **Note**: All Openmix apps require an associated platform set up beforehand. If you don't find a platform in the list, you can set it up in the Platforms page within the portal.
2. Select more platforms by clicking the **Add Platform** button.
3. Enter a CNAME or an A/AAAA record or IP (in DNS), or URL (in HTTP) for this Platform. It must be a valid URL, host name, or IP address. It can be of the form: `scheme:[//host[:port]][/path][?query][##fragment]`.
4. Ensure that the **Enabled** check box is selected (indicating that the platform is enabled) before moving to the next step.
5. If Sonar is available, and you want to use Sonar data to help in the initial decision-making process, make sure to click the **Use Sonar for Platform Availability** check box.
6. Click **Save** to go to Step 4 to assign appropriate weights for each platform.

Location Configuration

1. Assign **Weights** for the prioritization and selection of each platform globally and/or by market or country.
2. To assign platform weights separately for market or country, enter the name in the Markets & Countries search box and choose from the list.
3. Click **Complete** for your application to be created.
4. In the confirmation pop-up click **Done** or **Publish** to see your app listed on the Openmix page. If you click **Publish**, your app goes live instantly and have a green status. Your application is in production. If you click **Done**, your app is still listed on the Openmix page but it is unpublished, and its status is red.

Optimal Round Trip Time (ORTT) App

The ORTT app uses Radar Response Time, Sonar data, if Sonar is enabled, and the Platform Availability threshold to evaluate the best platform for the user the requesting. The availability threshold is the minimum availability (80% is the default value) that the platform must meet to be picked. In addition, the ORTT app also uses a Handicap value that globally or locally allows customers to influence how to route end-users.

The first three steps – Basic Information, Configuration, and Platform Information, are entered in the same manner as the other apps.

Follow these steps to configure location information and enter values for **Handicap** for each platform, globally, or by location/market.

Location Configuration

1. In the **Location Configuration** dialog box, enter a value for **Handicap** for one or all platforms selected. You can enter a handicap value between 0 and 6000. The use of the handicap is to manually lower the chances of a particular platform picked for routing, when there are better platforms available, in terms of cost or convenience. The more the handicap value, the lesser the chance of the platform being picked. You can deselect a platform if required by turning off the **Platform Selection** button.
2. Click **Markets & Countries** to select a particular market or country from the list and enter **Handicap** values separately for each of the associated platforms.
3. Click **Complete**, to finish configuring your app.
4. In the confirmation pop-up click **Done** or **Publish** to see your app listed on the Openmix applications list page. If you click **Publish**, your app goes live instantly and have a green status. Your application is in production. If you click **Done**, your app is still listed on the Applications page but it is unpublished, and its status is red.

Throughput

The **Throughput** app selects the platform based on Sonar data (if Sonar is enabled), highest throughput (using Radar data), and platform availability threshold (which is 80% by default). In addition, this app allows you to add a Handicap value to decrease the throughput for specific platforms and influence how end-users are routed. This optional Handicap value can be assigned globally and/or locally (for specific markets or countries).

The first three steps – **Basic Information, Configuration, and Platform Information** are entered in the same manner as the other apps. The **Location Configuration** is entered in the same way as in the ORTT app.

When you're done, click **Complete** to return to the Openmix applications list page. Finally, click **Publish** to publish your application when you are ready to go live.

Status of the application

The status of the app shows its current configuration.

- Red stands for unpublished. When you complete the configuration, if you click **Done**, your application is listed in the applications page with a red dot, denoting that it has not been published yet.
- Green stands for published. If you click **Publish** your app goes live instantly, and be denoted with a green dot which means that the application is in production.
- Yellow stands for the latest version that is unpublished. The yellow dot indicates that the application is created and edited, and the last modified settings are not yet published.

Static Proximity

The Static Proximity application responds to the platform that is located close to the latitude and longitude of the requesting user.

> **Note:**
>
> All Openmix apps require a set of associated platforms to be set up beforehand. If you do not find a platform in the list, you can set it up in the Platforms page within the portal.

Navigation

1. Log in to the Intelligent Traffic Management portal.
2. From the left navigation menu, navigate to **Openmix > Application Configuration**.
3. Click the plus button, **Add Openmix App** on the top right.
4. Select **Quickstart App**.

Basic Information

1. In the **Basic Information** dialog box, select **DNS** as the Protocol.
2. Select **Static Proximity** as the Application Type. Give the app a Name (required field), a Description (optional field), and a Tag (optional field).
3. Click **Next** for Configuration.

Configuration

1. If enabled, the **Availability Threshold** has a default value of 80%. Enter a new value to replace the default.
2. Enter a CNAME/A/AAAA or IP address for **Fallback**. The fallback CNAME/A/AAAA or IP is typically used if the application encounters issues, or errors. This field cannot be empty.
3. Enter **TTL (Time-To-Live)** for the response. The default is 20 seconds, but this value can be overridden if necessary.
4. Click **Next** for Persistency Controls.

Persistency Controls

Set up **Local Persistence**. For more information, see Local Persistence. Click **Next** for Platform Information.

Platform Information

Each platform must have its latitude and longitude setup through the **Platforms** page. Aliases for community platforms initially inherit geo information from the community platform, although after creating an alias you can change them. Private platforms need to be set up on creating them or afterwards through their config pane. To see the config pane simply click the Platform entry of the table.

Only platforms that belong to the following categories can have Geo info and be part of an opx app's answer list:

- Cloud Computing
- Cloud Storage
- Data Center

1. Select a platform from the **Platform** list.
2. Enter a CNAME or an A/AAAA record or IP (in DNS), or URL (in HTTP) for the Platform. It must be a valid URL, host name, or IP address. It can be in the form of: scheme:[//host[:port]][/path][?query][#fragment
3. Ensure that the **Enabled** check box is selected indicating that the platform is enabled before moving to the next step.

4. If Sonar is available for this platform and you want to use Sonar data to be considered during DNS resolving, make sure to click the **Use Sonar for Platform Availability** check box.
5. You can add more platforms by clicking the **Add Platform**.
6. Click **Next** for **Location Configuration**.

Location Configuration

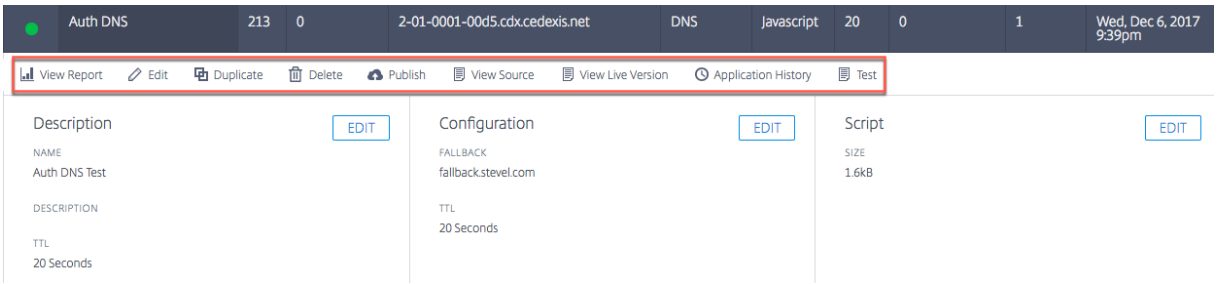
1. In the Global part of the Location Configuration dialog box, you can set up a chain of platforms for global routing. You can enable or disable the selection of each platform globally.
2. In the Markets & Countries, you can create different setups per market or country, effectively having geo fencing rules for them.
3. Click **Complete** to create your application.

In the confirmation pop-up click **Publish, Add another** or **Done**:

- If you click **Publish**, your app goes live instantly and the status is green. This means that the application is in production.
- If you click **Done**, your app is listed on the Openmix page but it is unpublished, and the status is red.
- If you click **Add another**, the status of the app is the same as **Done** but you restart the same process to create a new app.

Managing Quickstart Applications

Us the top tabs within the application manager panel to edit, duplicate, delete, test, view reports, view source and view the application's version history. Click your application in the Openmix applications list page to expand the application manager.



View Report

View Report takes you to the Openmix Decision Reports page where you are able to view the trend of Openmix decisions for each of your applications, platforms, and geographies.

Edit

To edit your Openmix app, simply click the **Edit** icon on the top of the application manager panel. You can also perform individual edits separately for basic information, configuration, platform, or location information by clicking the **Edit** buttons within the panel as shown in the figure. When you finish editing, click **Done**, to list the app with an unpublished status (for more edits later), or click **Publish** to go live instantly.

Duplicate

Click **Duplicate** to replicate the configuration of the current application and save it with a new name.

Delete

Click **Delete**, to remove applications that you no longer need.

Publish

Click **Publish** to directly publish the application from the Openmix application manager. This option is visible only if the app is not yet published.

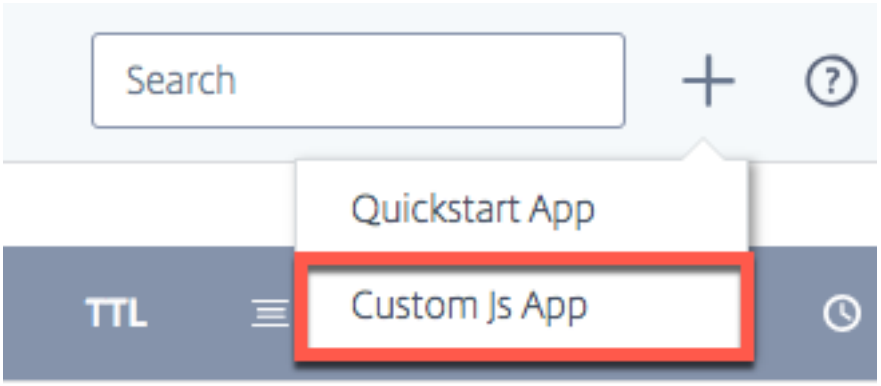
Openmix Custom JavaScript Applications

Openmix JavaScript applications are apps with customizable Java scripts. You can create, configure, test, and publish using the UI in the ITM portal.

Note: This guide does not cover the actual creation of the custom script (syntaxes, variables and so on). For more information on creating custom JavaScript, refer to the Developer Exchange.

Navigation

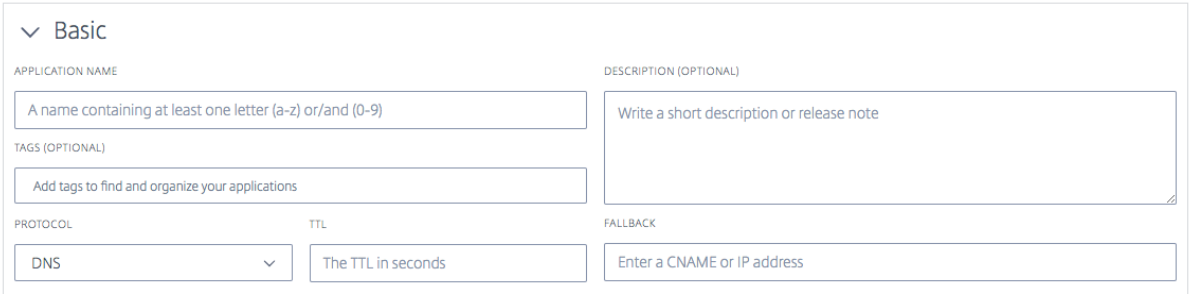
1. Log in to the ITM portal.
2. From the left navigation menu, go to **Openmix**.
3. Choose **Application Configuration**.
4. To configure a new Openmix app, click the add icon on the top right corner.
5. Select **Custom JS App**.
6. The **Openmix Application Configuration** page opens.



Basic Information

1. **Application Name**: Give a name to your app.
2. **Description**: Give the app a description or add a release note here. It's an optional field.
3. **Tags**: Enter an appropriate tag, if necessary. Tags help to identify and organize your app. It's an optional field.
4. **Protocol**: Select DNS or HTTP as the protocol.
   - **DNS**: If you select DNS, a TTL value must be entered.
   - **HTTP**: If you select HTTP, you can enable **Secure Access**.
5. **TTL**: Enter a DNS Time to Live for the application. The recommended value is 20 seconds. Note: This TTL applies if there is no TTL set by the custom JS app or if the response is a fallback value.
6. **Fallback**: Enter a CNAME/A/AAAA or IP address for **Fallback**. The fallback CNAME/A/AAAA or IP is typically used if the application encounters issues, or errors.
7. **Secure Access**: If **Secure Access** is enabled, the HTTP API must require an Oauth access key from the client when being called. Refer to Securing Openmix HTTP API to learn more.

   Note: Enabling secure access, displays a lock icon next to the app name in the list of apps on the Openmix front page.

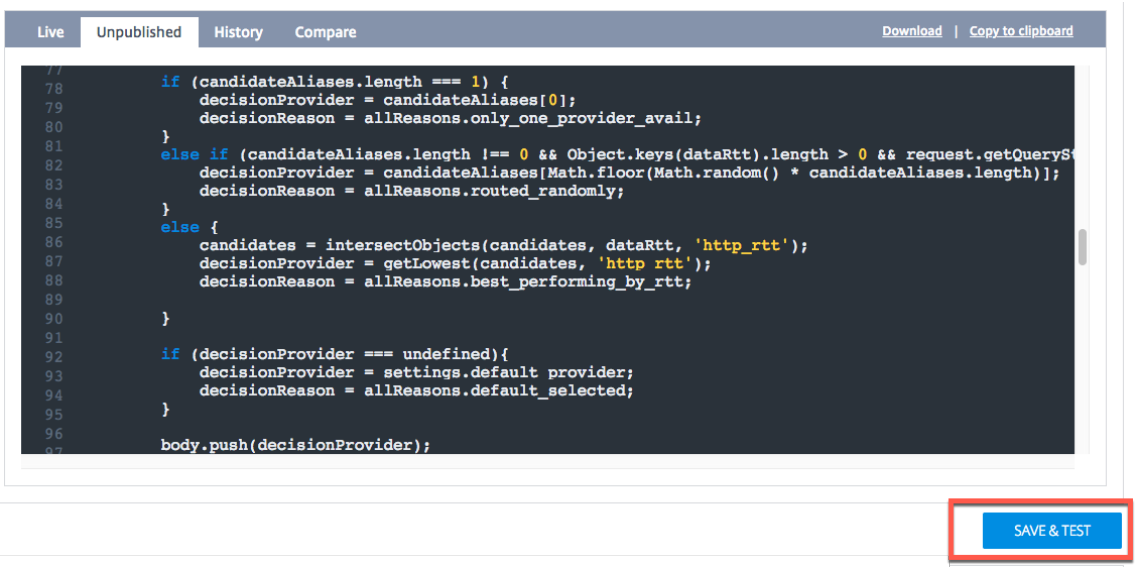


Custom JavaScript

Once you enter the configuration information, you can upload your custom JavaScript.

1. Click the **Choose File** button and select the JavaScript file that you want to upload. You can upload a new file to overwrite an existing one at any time.
2. Click **Save & Test** to save your application.

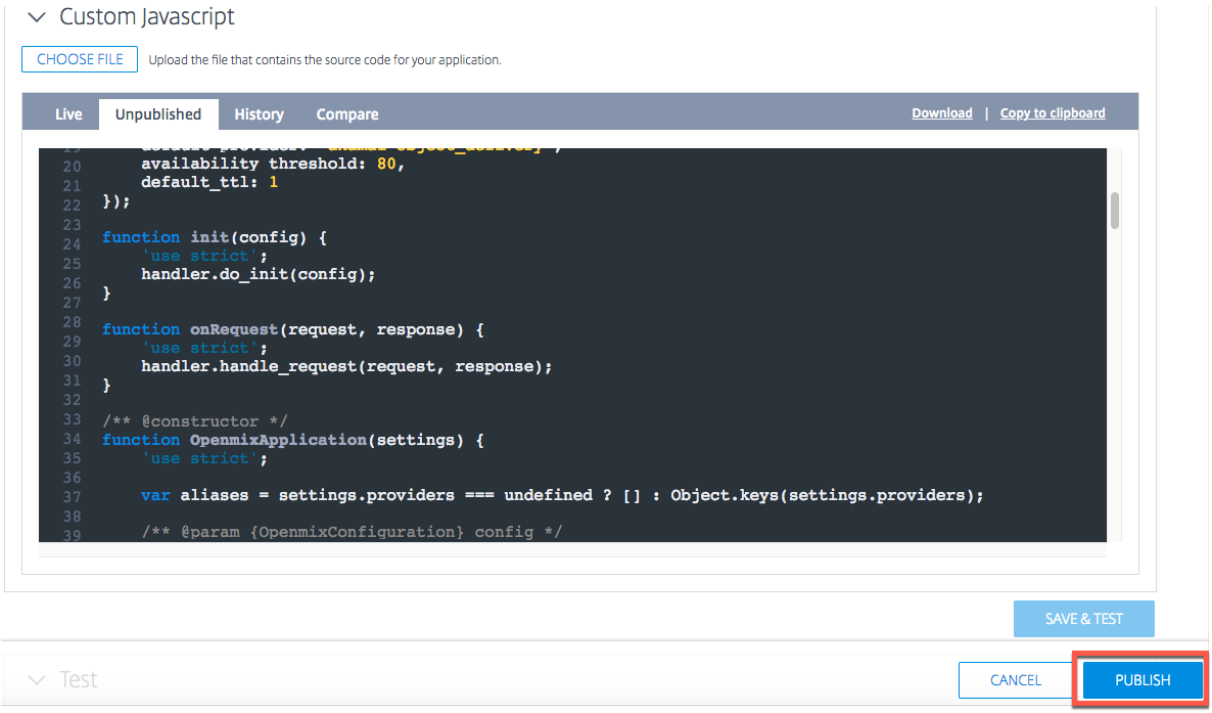
   Note: The application is automatically tested using an application checker when it is uploaded and saved. If there are errors, the application checker shows the error information and the location of the error. For more information about the data available from the application checker, see the Application Verification section.



3. Click **Cancel** to return to the Openmix Applications page or click **Publish** if you are ready for the application to go live.

   Note: If you click **Publish**, your app goes live instantly and have a green status. Yous application is in production.

   If you click **Cancel**, your app is listed on the applications page but is unpublished, and the status is red. To learn more about status refer to the Status of the Application section.



Staged Application Rollout

You can manage the rollout of your application by sending a small percentage of your web traffic through a new version, sometimes called as Canary Deployment. ITM allows you to send a specified percentage of traffic to the new version of an app to ensure that the application logic behaves as expected. You can report on the behavior of the existing and new versions to evaluate the changes to your app in a live environment. This option allows you to fix any issues or anomalies that occur before routing 100% of your web traffic through the newly edited app. After verifying the desired behavior, you can increase the percentage of traffic to the newest version or deploy the application to all users.

To stage the application rollout and release a test version of your newly modified app, do the following:

- Click the app name (in the Openmix applications list page). The application manager panel opens.
- Click the **Edit** icon to edit your app.
- Modify your existing app with all necessary changes.
- Once you're done with the edits, click **Save and Test**.
- Scroll down at the bottom of the page with **Cancel** and **Publish** buttons. Enter the percentage of web traffic (1% to 99%) that you want to flow through this newly modified version.
- Check the box for partial distribution of traffic through this new version of the application. The remaining traffic is sent to the previous live version.
- Click **Publish**. This new test version of the app now appears in the list of apps in the **Openmix Configuration** page with a new **Status** icon. The new **Status** icon signifies that only partial web

traffic is flowing live through this version.

You can modify the traffic flow to the test version and change the percentage of traffic flow to view performance.

```
1 ![Canary](/en-us/citrix-intelligent-traffic-management/media/openmix-
    jsapp-edit-canary.png)
```

To check the performance of your app, go to the Openmix Decision Report. Select **Application** as your primary dimension, and **Version** as your secondary dimension. Then click **Apply Filters** after selecting your application from the list. The chart shows the performance of different versions of your application.

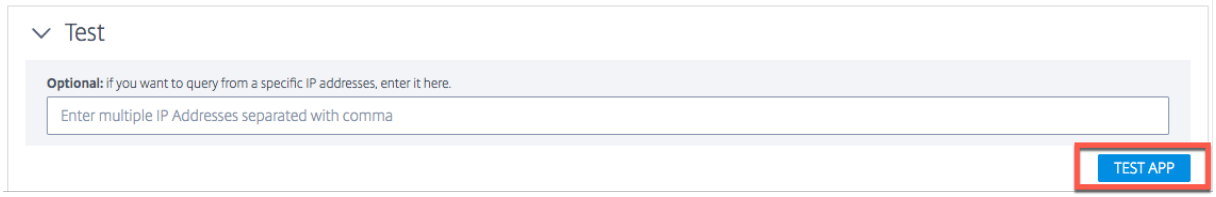
Once you're satisfied with the performance of this version of the app, you can go ahead and route 100% of your web traffic through it by clicking the **Go Live** button.

This version replaces the current live version with the newly edited version.

If you don't want to go live with this version, click **Unpublish**. Your changes are saved and appear as an unpublished app in the list of apps in the **Openmix Configuration** page. Now 100% of your web traffic flows through the current live version of your app.

Test

You can test your JavaScript application using the **Test App** button before or after publishing.

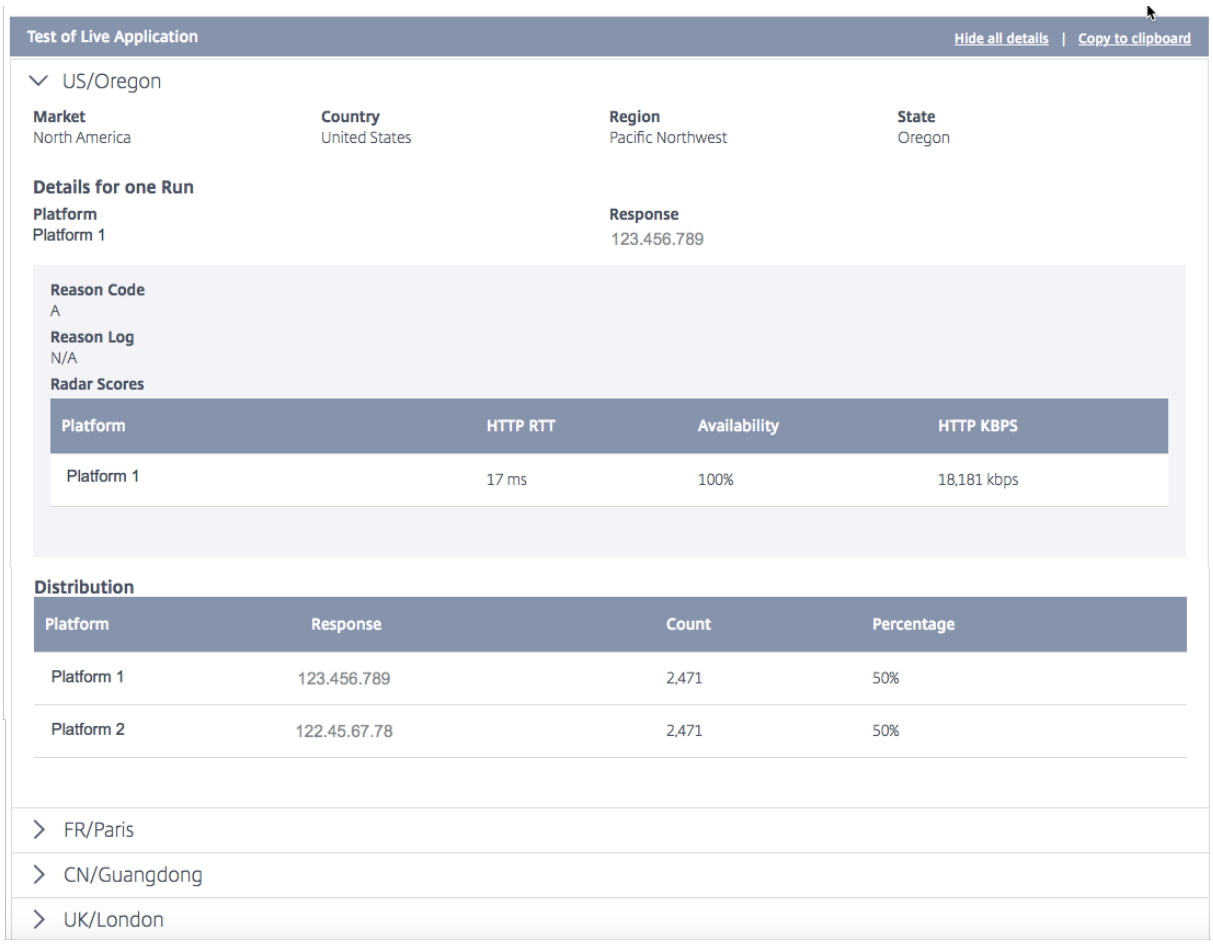


It enables you to view test results across specific sets of markets, countries, regions, and states. You can query the app from specific IP addresses.

Test results include, **Platform** selected by the app, **Response** received, **Reason Code**, **Reason Log**, **Radar Scores**, **Distribution** and so on

This feature also allows you to view the distribution of decisions across different platforms. For example, if two platforms are used for routing, you can view the number of decisions and the response received for each of them.

Click the **Show All Details** link to see the test results of your app.



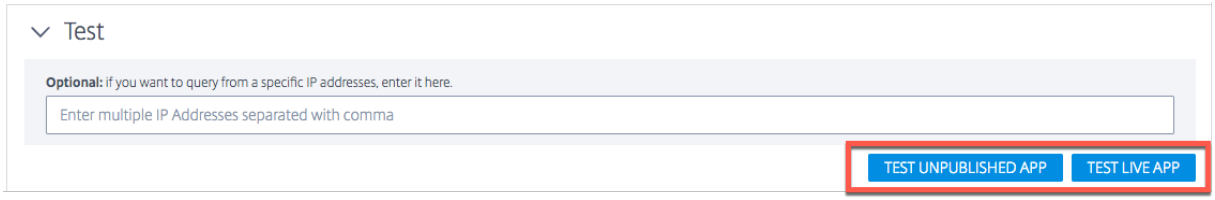
The following values are shown as test results:

Field	Description
Market, Country, Region, and State	The location at which the app was tested.
Platform	The platform selected by the app.
Response	The CNAME or IP address of the platform selected by the app.
Reason Code	Describes the reason behind the decision.
Reason Log	Customer-defined output from the app. Enables customers to log information about the app decisions.
Radar Score	The **Response Time (RTT)**, **Availability**, and **Throughput** measurements recorded for the platform.

Field	Description
Distribution	The distribution of platforms that an app selects for each location that is tested. The **Count** represents the number of times the platform was selected. And the **Percentage** is the percentage of the total count for platform selection.

Note: You can run this test on the live app or the unpublished version, that is, if the app is not yet published.

Once your app is published, you have the option of testing the live app by clicking the **Test Live App** option. If you edit your app or upload a new version, you can test it before publishing by clicking the **Test Unpublished App** button.



Application Verification

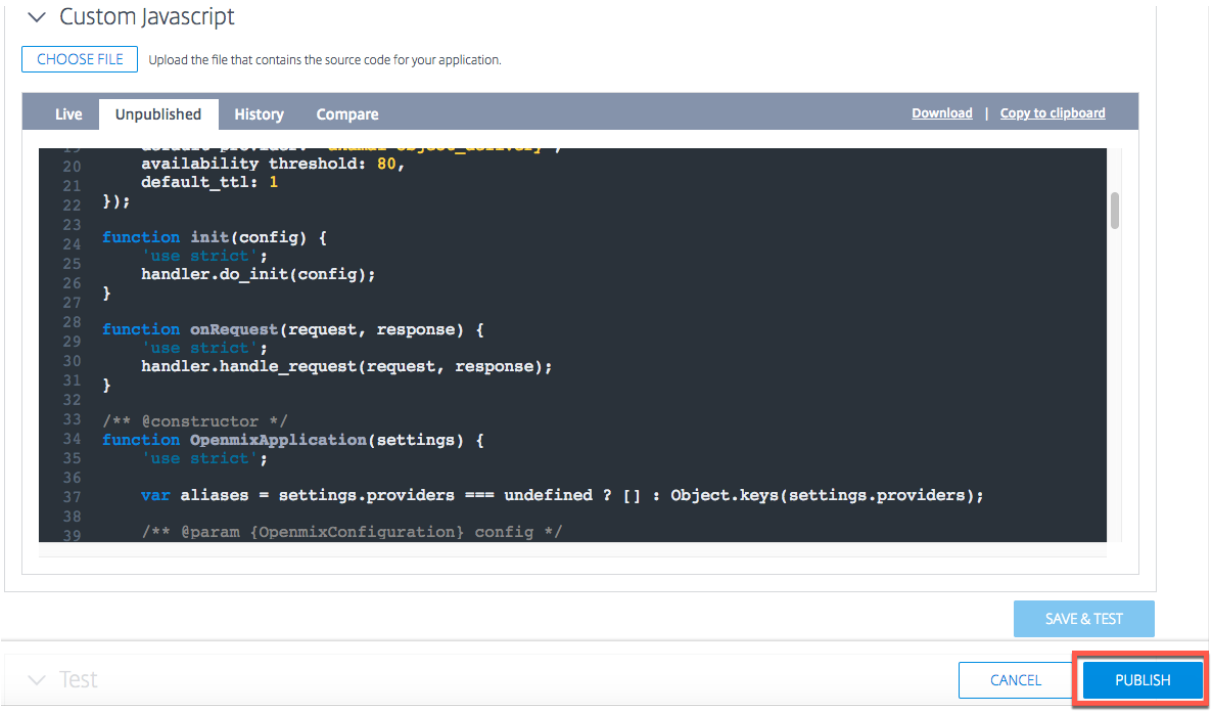
To ensure that custom JavaScript apps behave as expected, run the app through a code and logic verifier when you upload it to the ITM Portal. The application verifier runs the app through a decision server with synthetic traffic to test whether the application compiles and runs successfully.

If the application runs without error, the verifier provides information about the decision distribution and execution characteristics. On the other hand, if the decision server encounters an error while running the app, the verifier provides information about the error. We recommend that the application must be free of errors before publishing.

In errors, you can fix the JavaScript file in your local and reupload it to the Portal by clicking the **Choose File** button.

Publish

To publish your app and have it go live, click the **Publish** button. This option is grayed out if the app is not yet saved or already published. When the app goes live, it appears in the Openmix application manager page with a green status. To learn more about app status refer to the Status of the Application section.

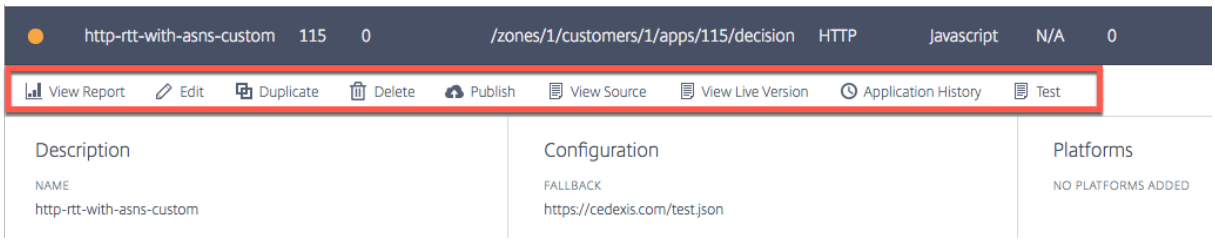


Note: The app is published with errors if necessary.

Managing Custom JavaScript Applications

Use the top tabs within the application manager panel to view reports, edit, duplicate, delete, publish, view source, view live version, view history.

Click your app in the Openmix applications list page to expand the application manager panel.



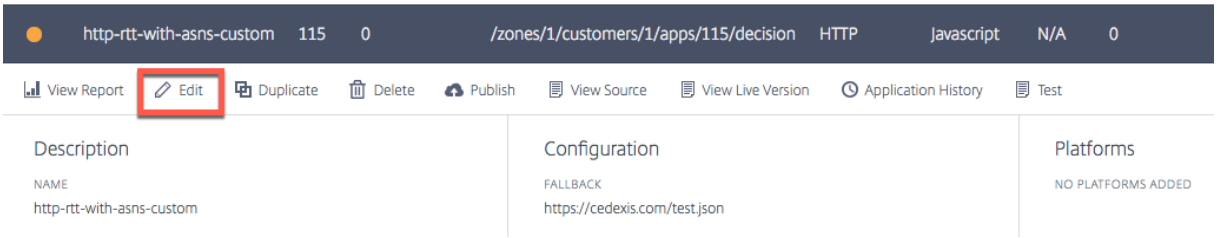
View Report

View Report takes you to the **Openmix Decision Reports** page where you are able to view the trend of Openmix decisions for each of your apps, platforms, and geographies.

Edit

To edit an Openmix custom Javascript app, click the app name (in the Openmix applications list page). The application manager panel opens. Changes and updates can be made to the configuration by

clicking the **Edit** icon.



View Source

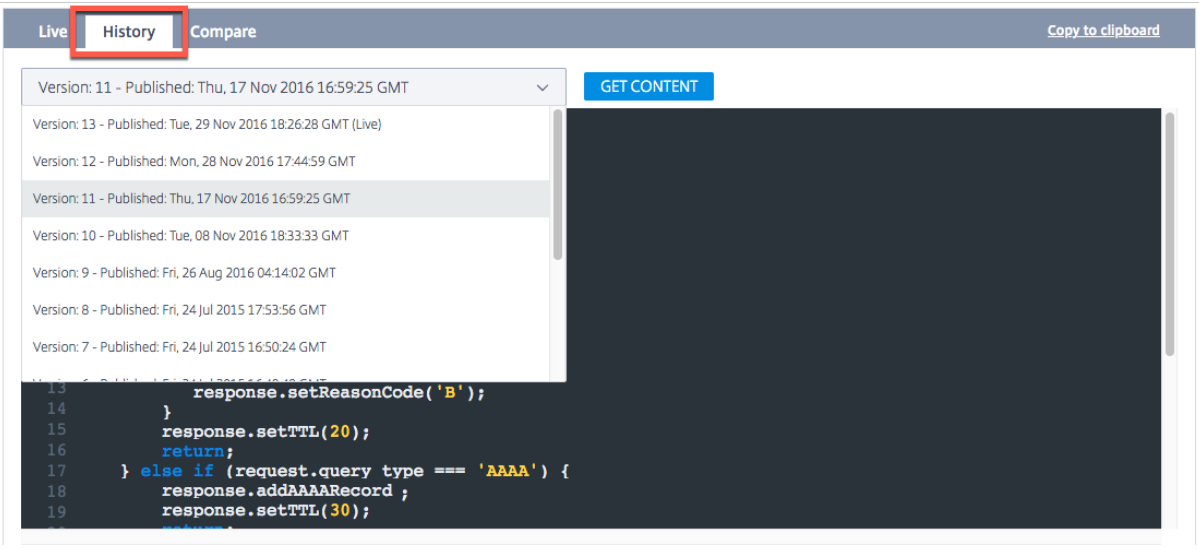
View Source allows you to view the JavaScript source of the app, that is, the latest version of the app whether it was published. This option is only available for custom JavaScript apps.

View Live Version

You can view, copy, and download the latest published version of the app. This option only available for custom JavaScript apps.

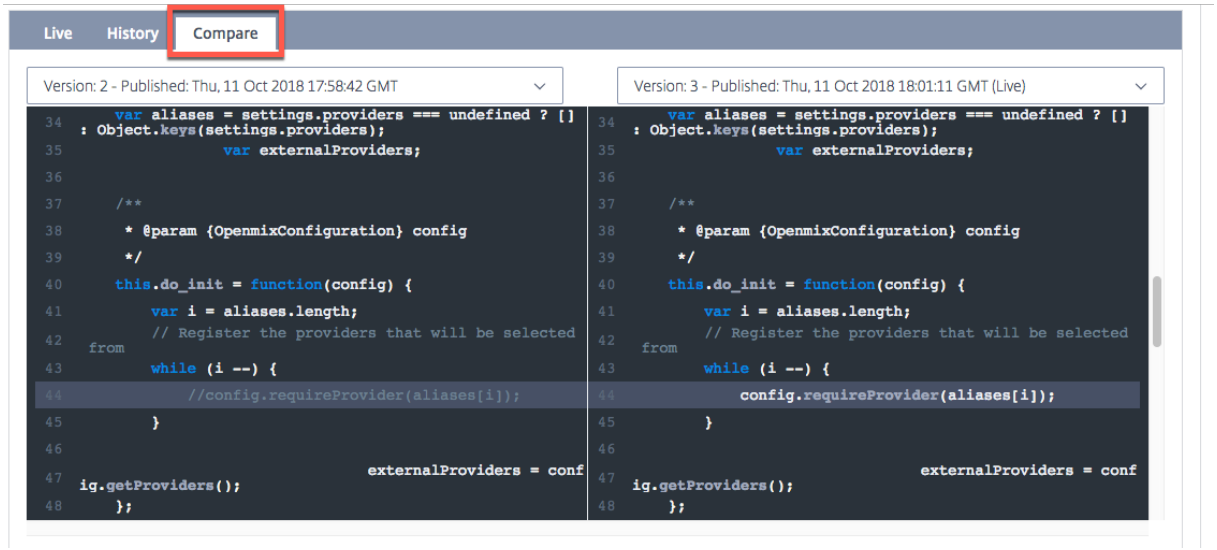
Application History

Application History allows you to view different versions of the app. You can use the **Select a Version** list to switch from a live version to an older version. Click **Get Content** to switch to the older version. This option is only available for custom JavaScript apps.



Compare

The **Compare** feature allows you to compare different versions of your JavaScript file. You can see the differences between the two versions of your app clearly displayed with highlighted lines of script.



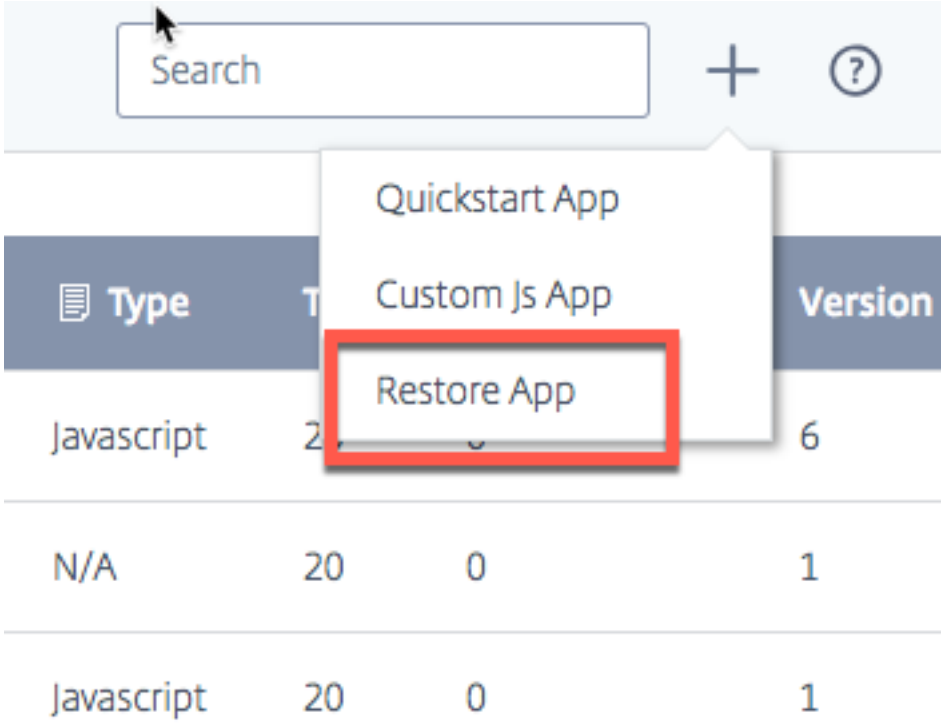
Delete

To delete an Openmix app, click the app name (in the Openmix applications list page). The application manager panel opens. Click the **Delete** icon, and then choose the **Delete** button in the confirmation dialog box. The app disappears from the list.

Restore App

The **Restore App** feature allows you to re-enable an app after it has been deleted.
To restore an app, do the following:

1. Click the **Add +** icon on the top right of the page.
2. Choose **Restore App** from the dropdown menu. The **Restore Application** window opens.



3. Find the app that you want to re-enable from the list and click its corresponding **Restore** button.

The app is put back on the list in the Openmix page with the same status.

Local Persistence

The **Local Persistence** feature offers the capability for decision stickiness when it is enabled for an Openmix application. The requests are identified using the IP subnet mask, the length of which is configurable. For example, when a client repeats a request to the same application within a certain period, then the original decision is served back. It can be an essential feature when a client is required not to bounce between different decisions during a particular session. It is available for both DNS or HTTP Openmix applications.

Due to the underlying natural restrictions of the mechanism, persistence is not guaranteed for 100% of the requests. A best effort approach is applied, instead. Tests have shown the expected persistence accuracy to be in the range of 95-97%.

> **Note:**
>
> To enable the Local Persistence feature for your account, open a support ticket or contact your customer success manager. In addition, a Predictive DNS zone is required, configured with name servers `ns5.cedexis.net` and `ns6.cedexis.net`. Consider the significant amount of time the DNS zone updates might require to propagate across the internet.

Configuration

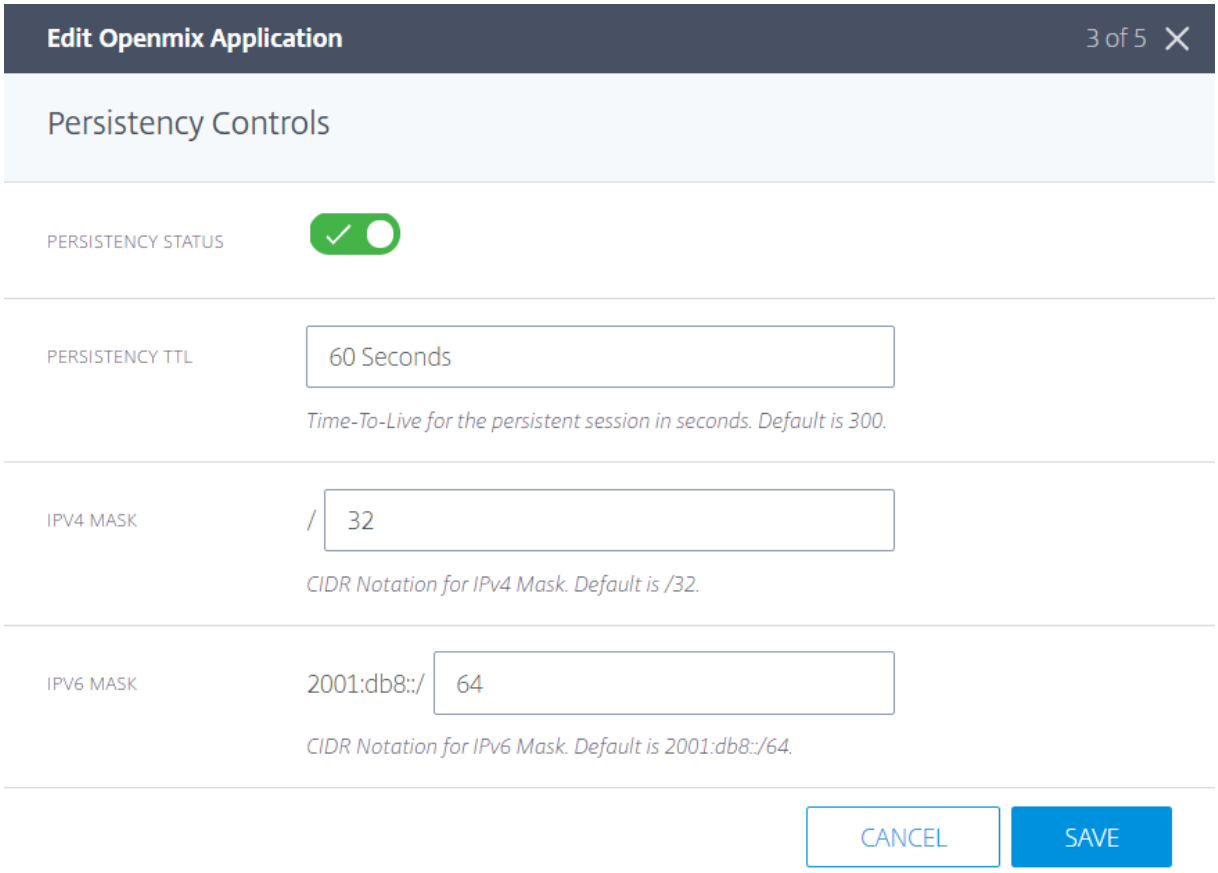
To enable Local Persistence, select **Persistency Controls > Edit**, under the Openmix application options.

The available settings are as follows:

1. In the Configuration dialog box, enter the **Persistency TTL**. The default option is 300 seconds. Values between 60 and 1440 are allowed. After an initial request, the DNS decision served is kept for a maximum of 300 seconds. If another request comes from the same IP subnet range in the system before the expiration, it serves the same decision.
2. Both IPv4 and IPv6 masks are provided for setting the granularity of persistency stickiness. Default is “/32” and “/64”, for IPv4 and IPv6, respectively. Permitted values are:

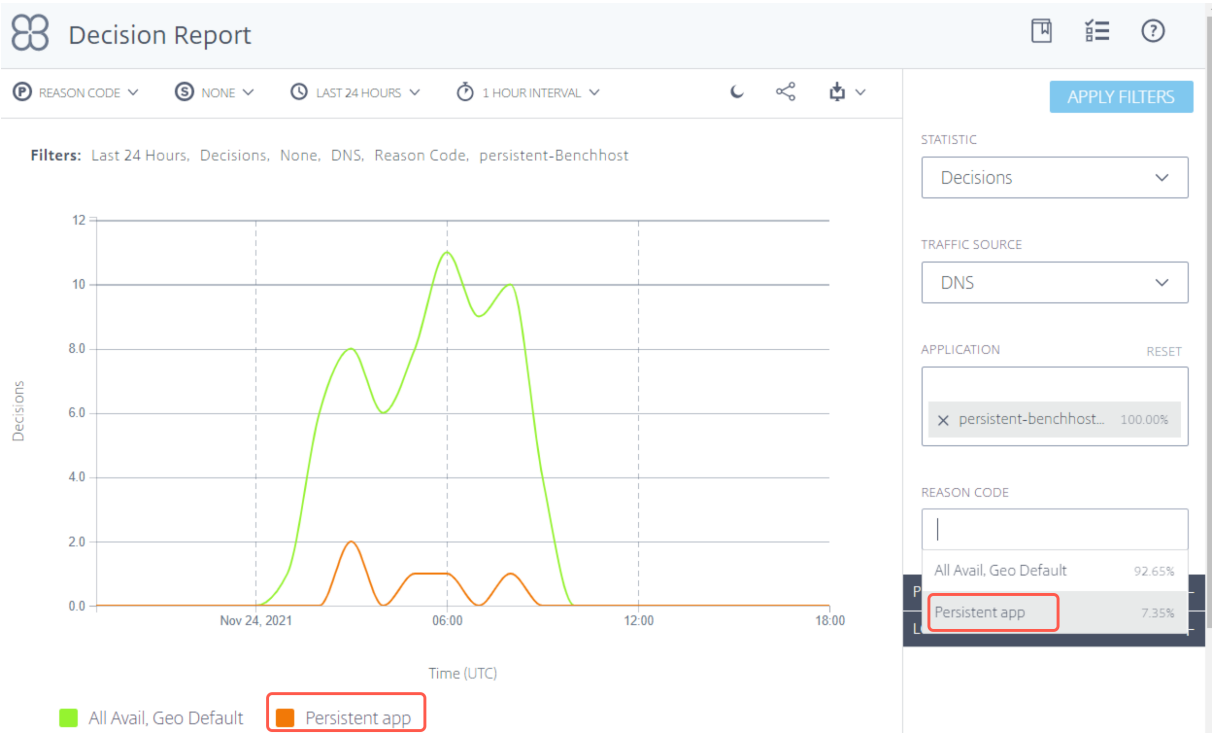
   - /8 up to /32, for IPv4
   - /32 up to /64, for IPv6

   This masking on the client’s IP address determines the persistency key used in the internal data store. For example, if two (or more) client IPs map to the same masked IP address, they are served with the same persistent decision.



The same settings are also available under the predictive application settings.

The Openmix decisions that are provided via the internal data store, are reported with the reason code **Persistent app** in the Decision Report.



Health checks

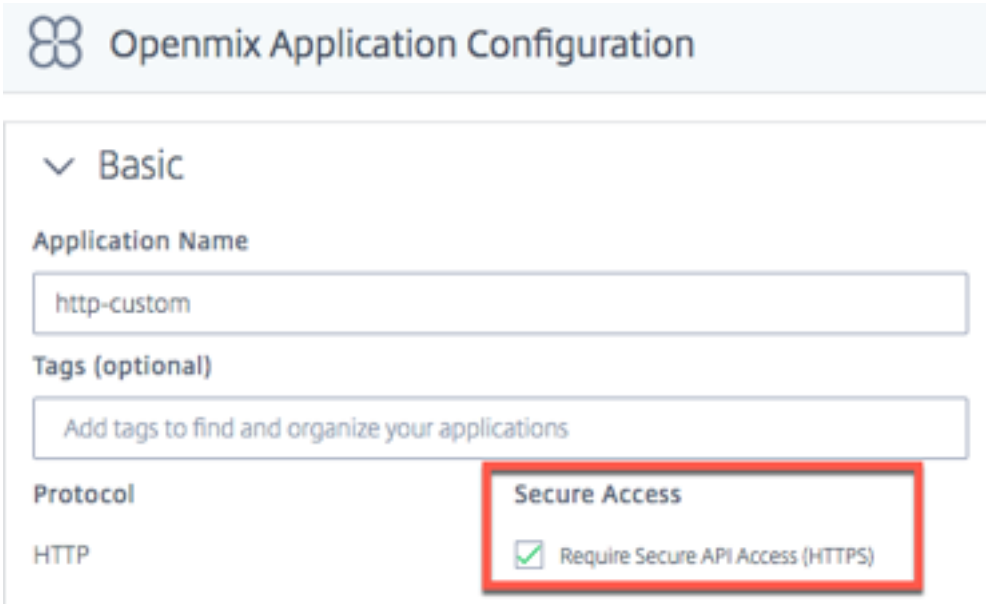
Decisions served from the persistency cache are subject to extra health checks before they are served:

1. If the application is configured with **Sonar Availability Check**, then the Sonar availability health is checked before a cached decision is served. If Sonar reports the platform is “down” then the cached decision is ignored and the OpenMix application is run again.
2. If the application is configured with **Radar Availability Check**, then the Radar availability health is checked before a cached decision is served. If the platform’s availability is less than the configured threshold then the cached decision is ignored.

> **Note:**
>
> For persistence, the maximum threshold for the Radar availability health is set to a fixed 10%.

Securing the Openmix HTTP API

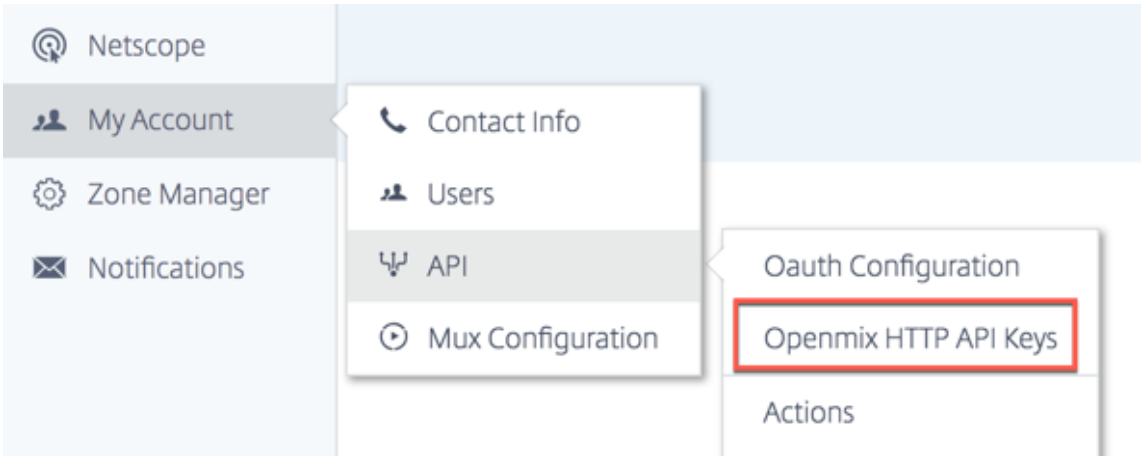
Openmix is available via DNS or an HTTP API for integration into non-DNS workflows. By default, the HTTP API is called over plain HTTP. The API can also be secured via TLS and key authentication. It is done via the UI by checking the box for **Require Secure API Access (HTTPS)**.



Creating API Keys

To enable key authentication, do the following.

1. Select the **Require Secure API Access (HTTPS)** box in the **Openmix Application Configuration** page to turn on secure access for each application.
2. To generate a secure access key, navigate to **My Account** -> **API** -> **Openmix HTTP API Keys**



3. If you're a first-time user, you are prompted to get started by entering your client ID. Enter your **Client ID** in the **New Client** dialog, and click **Complete**.
4. The **Client Secret** key is displayed beside the **Client ID** on the **Openmix HTTP API Authentication Configuration** page.
5. You can now make a request to the Openmix app using basic authentication. Use your **Client ID** as the user name and the **Client Secret** as the password to invoke the app on the browser.

To invoke the app using the command line, use the following cURL command:

```
curl https://hopx.cedexis.com/zones/<zone>/customers/<customer_id>/apps/<app_id>/decision --user <client_key>:<client_secret>
<!--NeedCopy-->
```

Note: Keys you create gives you access to any of your Openmix applications.

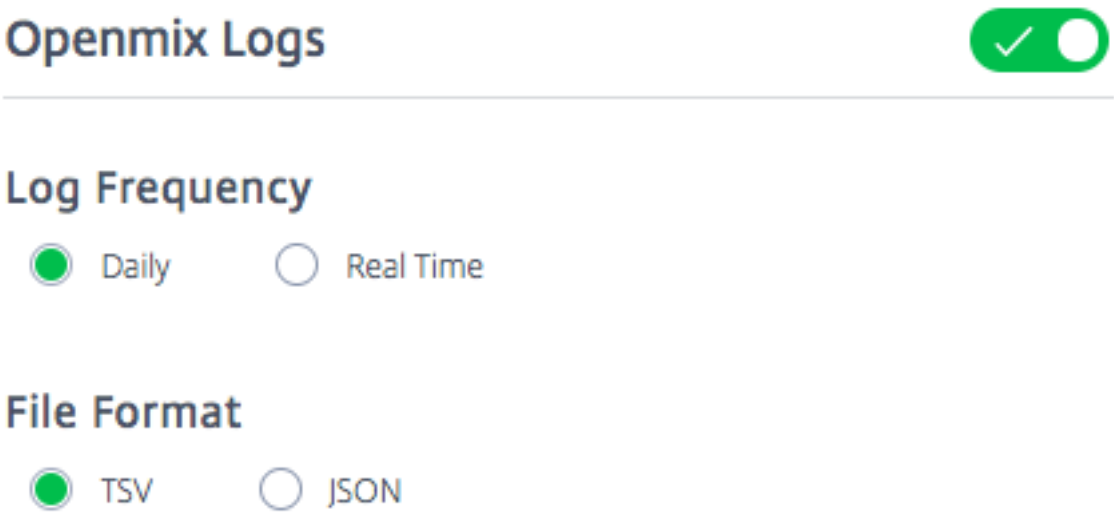
For more information on calling the Openmix HTTP API, refer to the Openmix HTTP API usage documentation.

Deleting API Keys

1. To delete a key, navigate to the **Openmix HTTP API Authentication Configuration** page.
2. Click the **Client ID**.
3. Choose **Delete** in the list. The key is removed from the system. It isn't valid for authentication or secure access to the Openmix application.

Accessing Logs

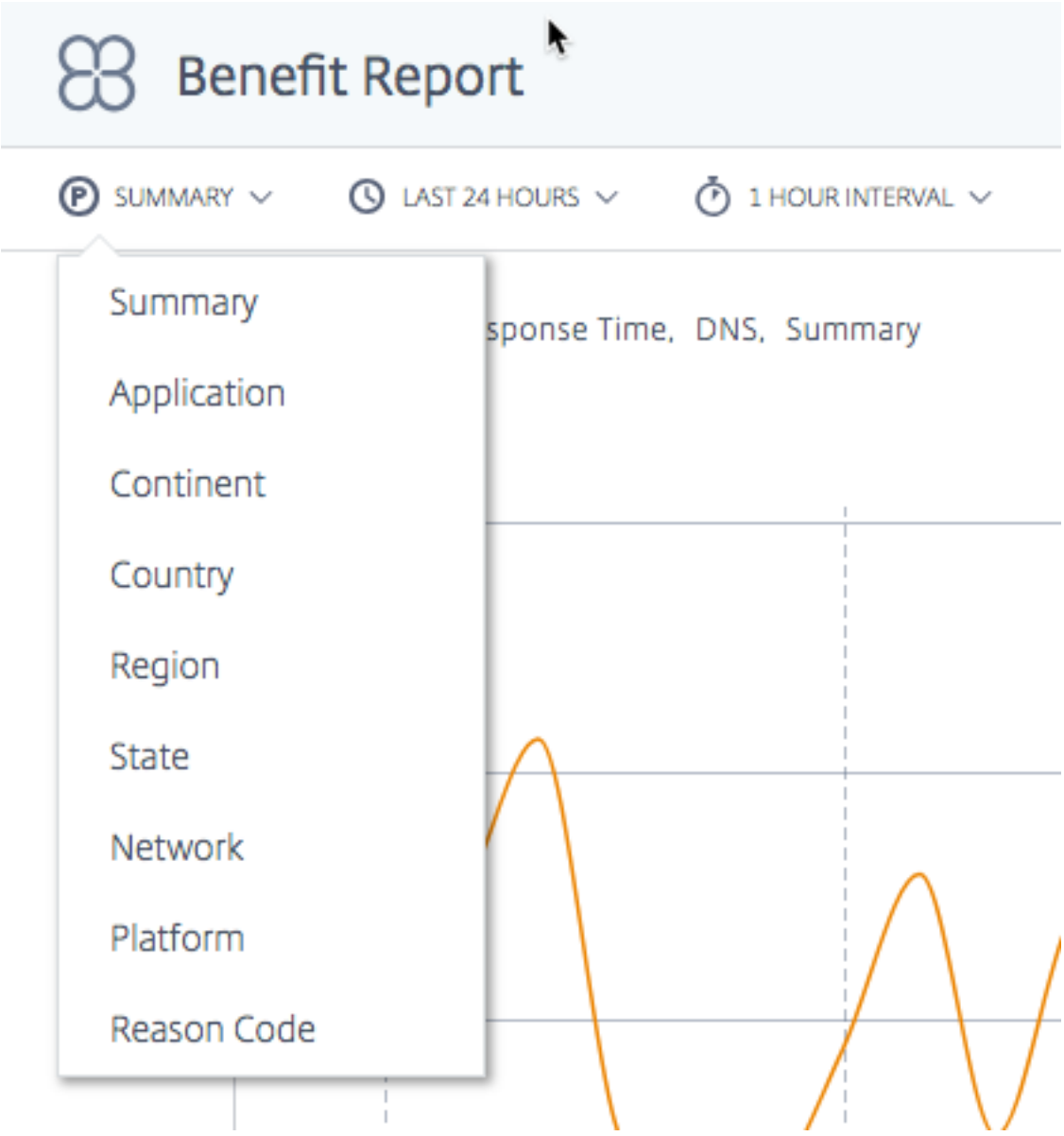
Decisions log made by Openmix can be collected and made available for secure download. These logs can help you analyze the decisions made by your Openmix application and debug request behavior. The logs can be turned on/off and secured at the account level. For details on how to enable and download Openmix logs and see that log descriptions go to Netscope.



Openmix Reports

Openmix reports provide powerful visibility into the Openmix decisions that were made for your DNS or HTTP traffic. Each report is defined in the following section, but here are some important aspects about the reports:

Primary and Secondary Dimensions

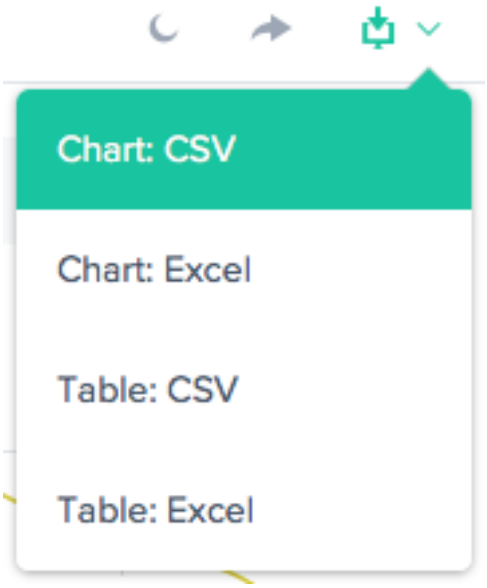


The primary dimension of the chart is selected through a list above the chart. Use this list as a powerful pivot on the report. A secondary dimension can be chosen as well to further refine the reporting.

Visualization Background Toggle

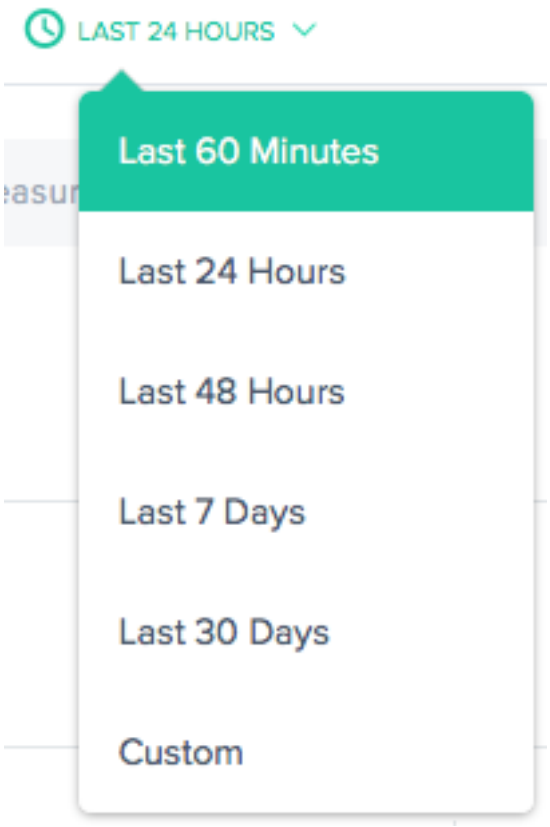
Charts are set to a white background by default. Toggle the background to a dark color for high contrast monitors using the background toggle.

Data Export



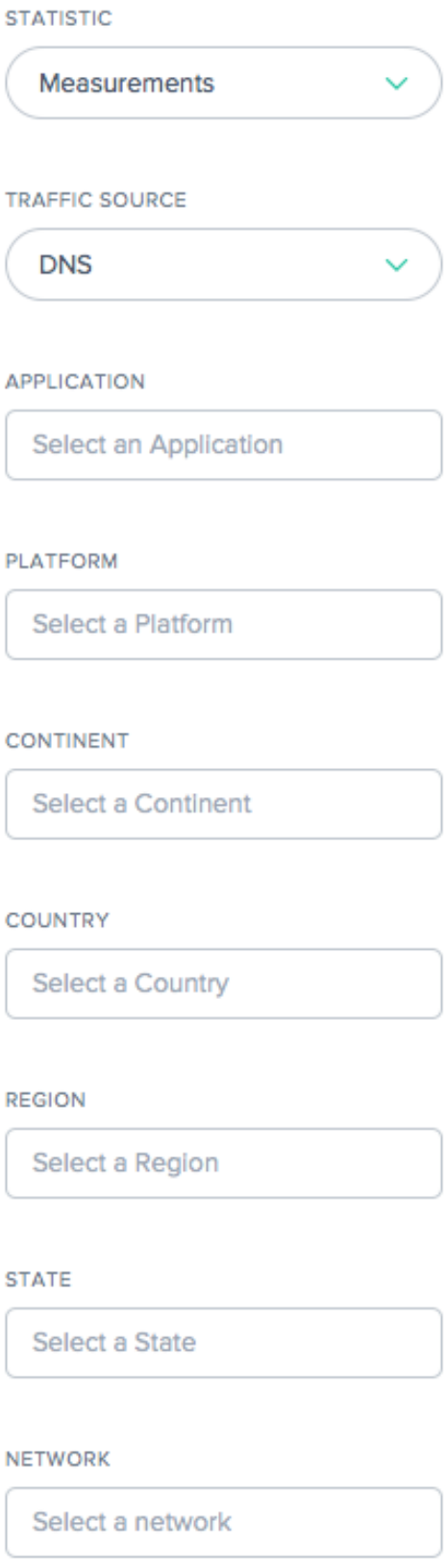
In addition, the end-user is able to download the Chart and Table data via the download link at the top of the report.

Filter: Report Time Range



You can generate a report with a time range of last 60 Minutes, 24 Hours, 48 Hours, 7 Days, 30 Days, or a custom range. The default view is the Last 24 Hours.

Filters: Powerful Drill-down Capabilities



The reports vary slightly in terms of which filters are appropriate based on the data. The following are the most common:

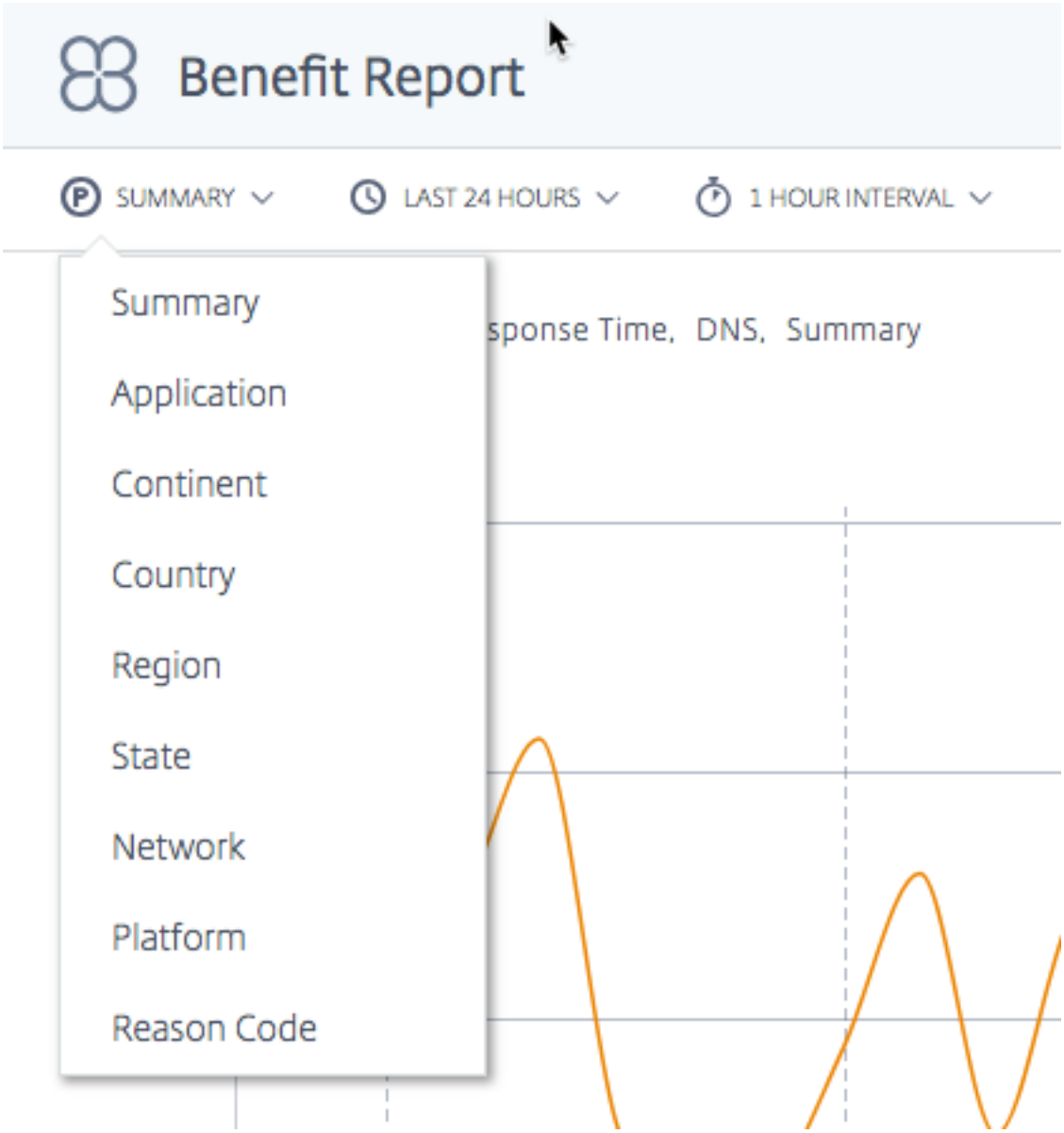
- **Statistic** - Select the value shown in the chart, most often the number of decisions.
- **Traffic Source** – Select the type of traffic to display: DNS or HTTP.
- **Application** – Select one or more Openmix applications to display.
- **Platform** – Select one or more platforms (provider) to include.
- **Continent** – Select one or more continents to include.
- **Country** – Select one or more countries to include.
- **Region** – Select one or more geographic regions (where applicable) to include.
- **State** – Select one or more geographic states (where applicable) to include.
- **Network** – Select one or more networks (ASN) to include.

Benefit Report

The Benefit report gives you the overall improvement in the performance of your application delivery when you use the Intelligent Traffic Management (ITM) service. The benefit is shown as a percentage improvement in response time and throughput. Choose a specific platform from the pool of candidate platforms to generate the report.

Primary Dimensions for the Benefit Report

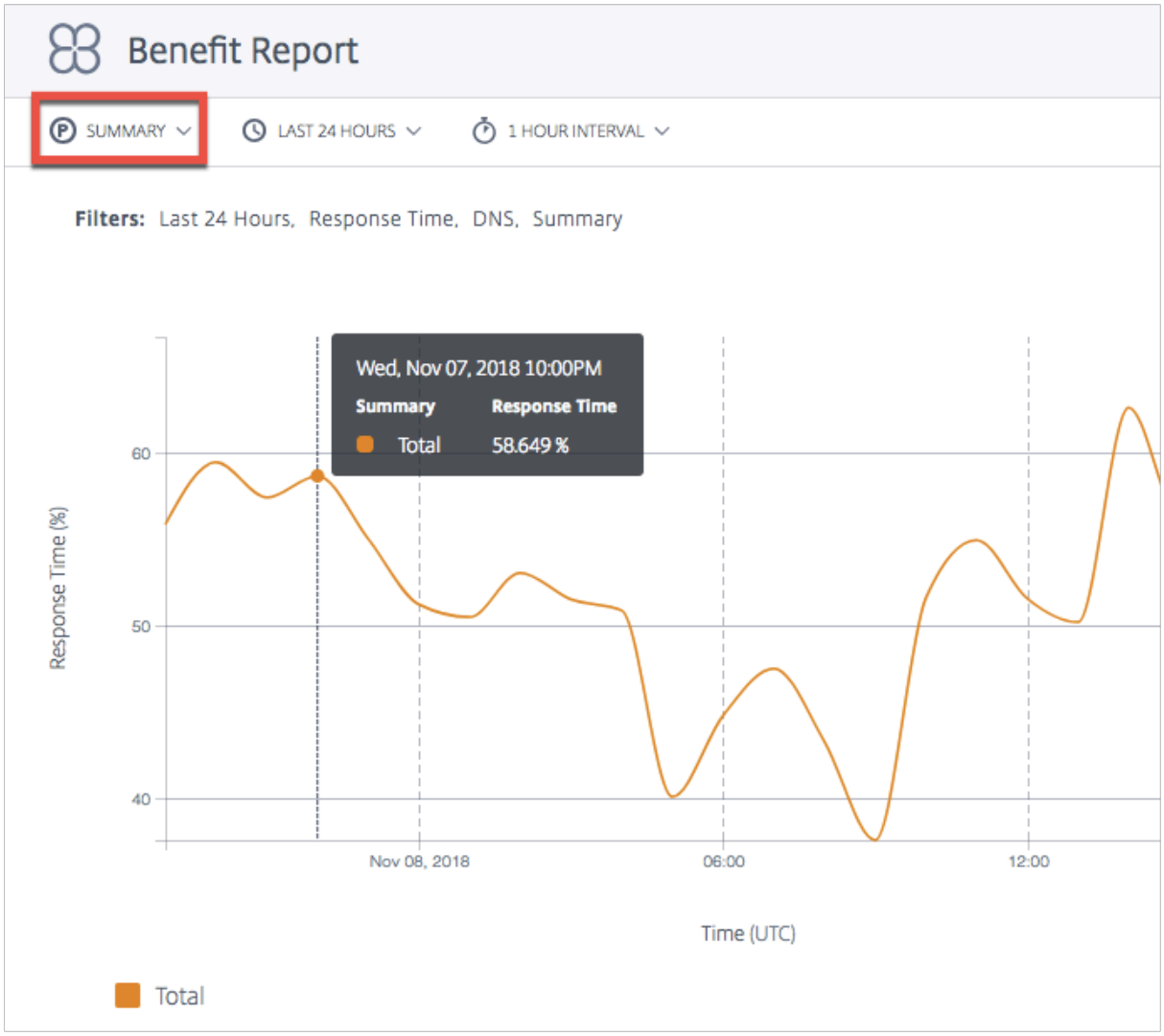
Primary dimensions are independent measures based on which the Benefit Report is displayed. The following sections describe each of these primary dimensions in detail.



Summary

Summary is the default primary dimension. The summary chart shows the average of the total percentage benefit (in terms of response time or throughput) received from all the applications.

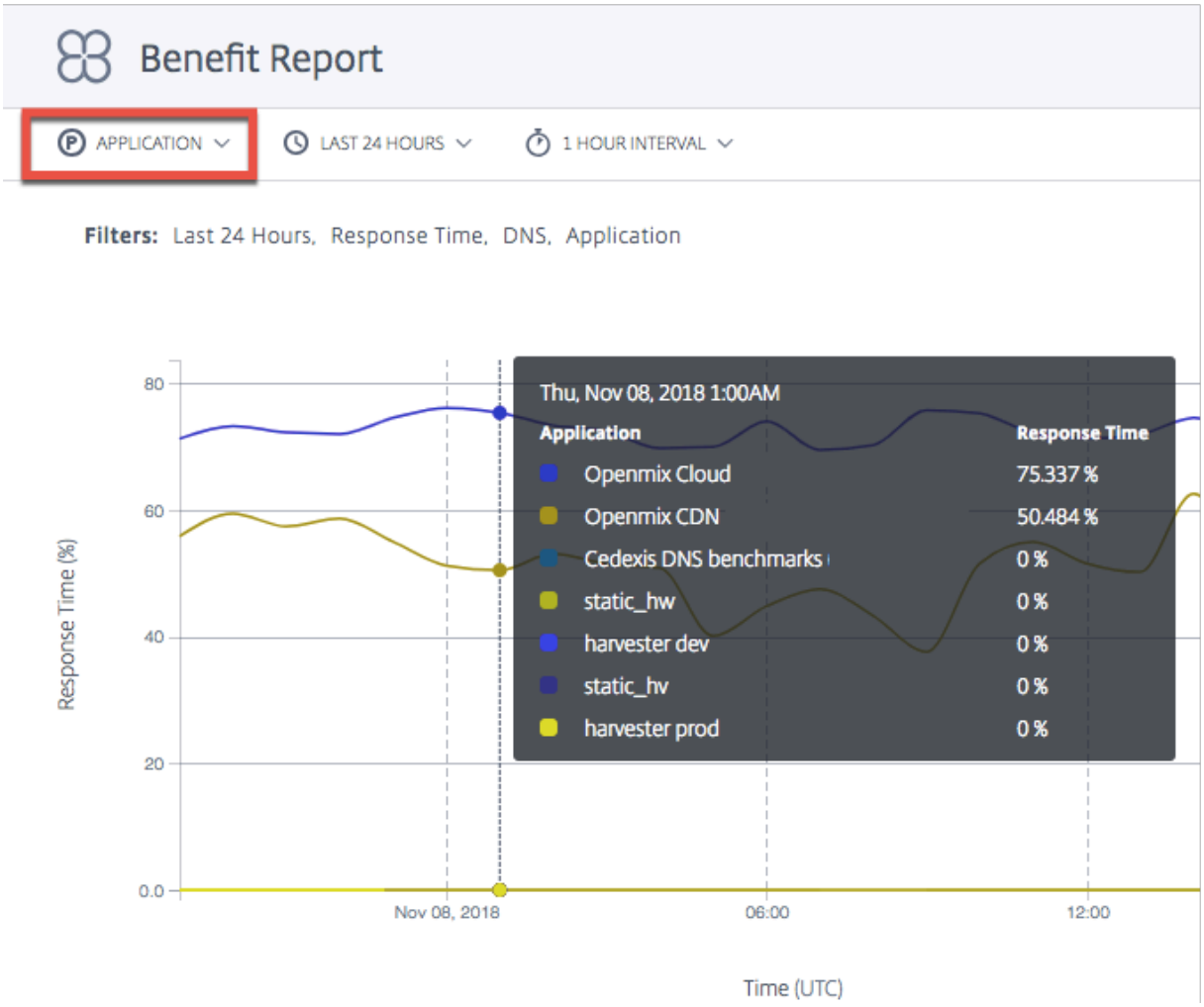
Note: You can switch between the benefit shown in terms of **Response Time** or **Throughput** by using the **Statistic** filter.



Application

When **Application** is chosen as the primary dimension, the chart shows each of the applications and the corresponding performance (in terms of response time or throughput) as a percentage benefit in choosing a certain platform over other candidate platforms.

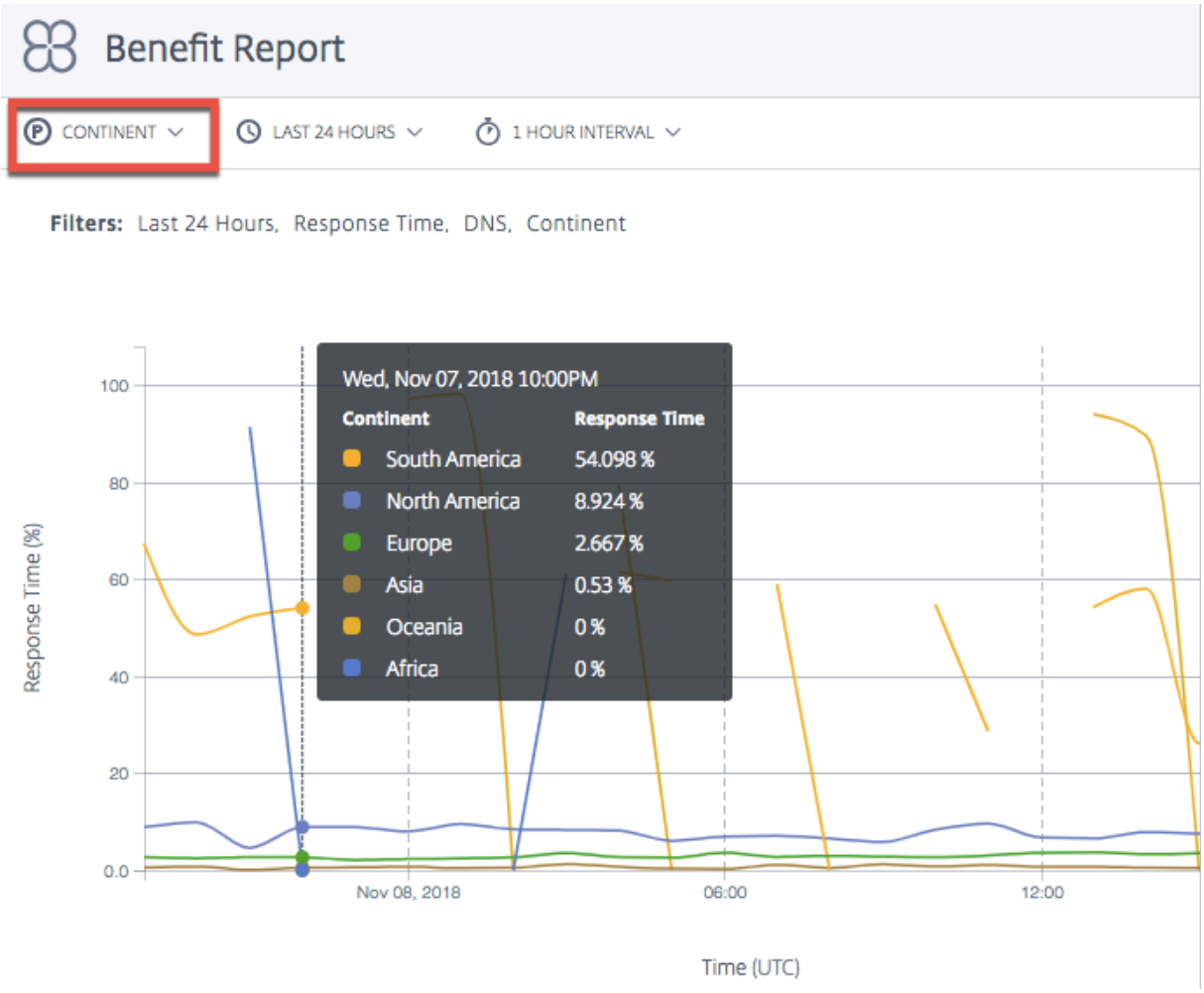
Note: 0% means that there was no extra benefit or improvement in selecting a specific platform over another.



Location (Continent, Country, Region, State)

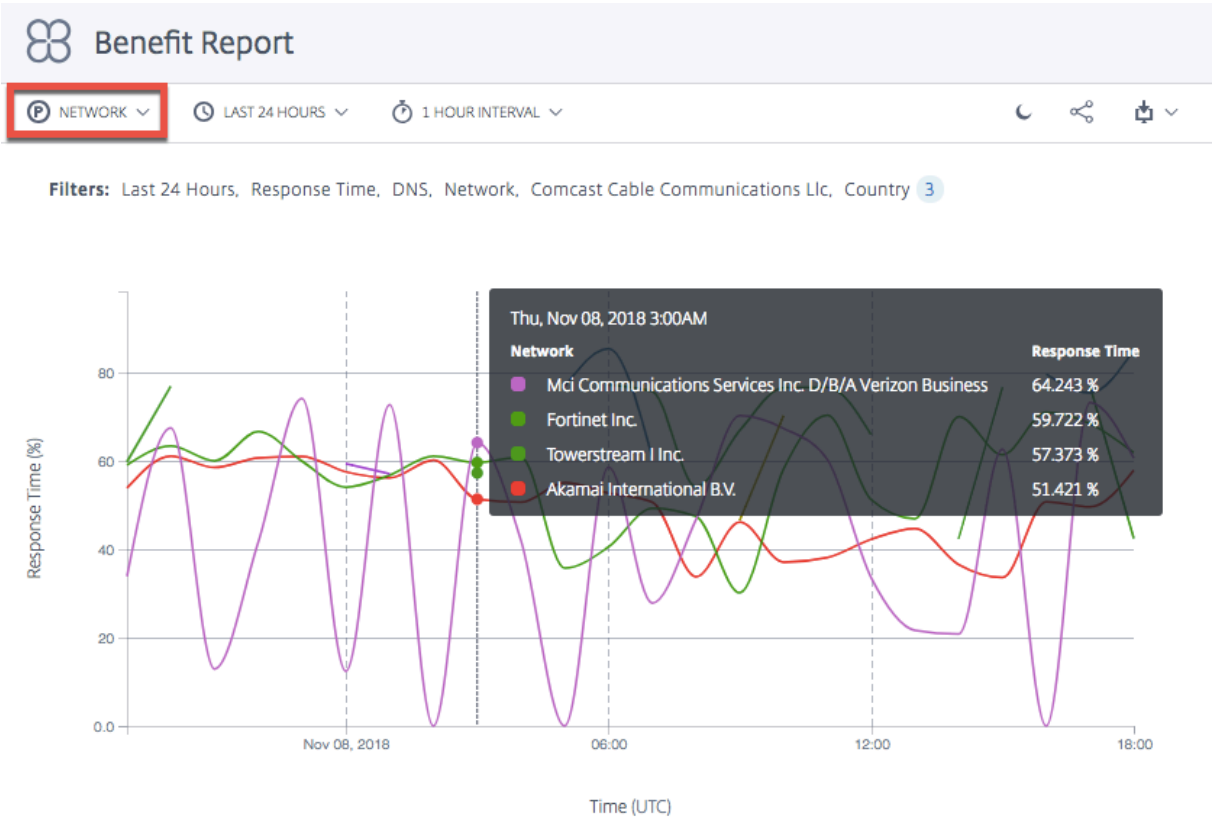
When location (**Continent**, **Country**, **Region**, or **State**) is selected as the primary dimension, the benefit report shows the average of the total percentage improvement in performance (in terms of response time or throughput) for each location. You can select location by continent, country, region, or state.

Note: Platforms that are not eligible for selection because of geo rules or any other reason is not counted in the calculation. However, platforms that are geo-fenced for the location in question are counted.



Network

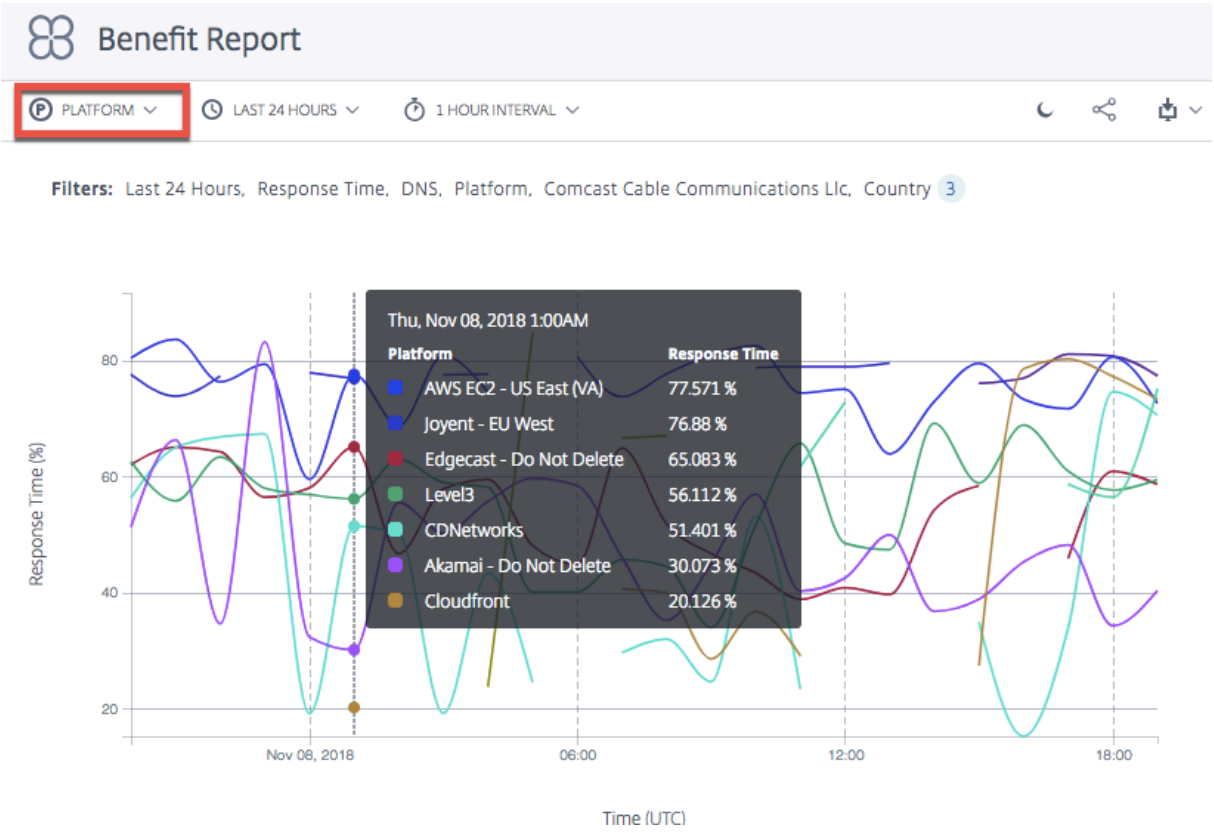
When you select **Network** as the primary dimension, you see the percentage improvement in performance for users grouped into the specific networks (or service providers) from which users access ITM. It helps you know which groups of users are seeing the performance benefit when coming from those specific networks.



Platform

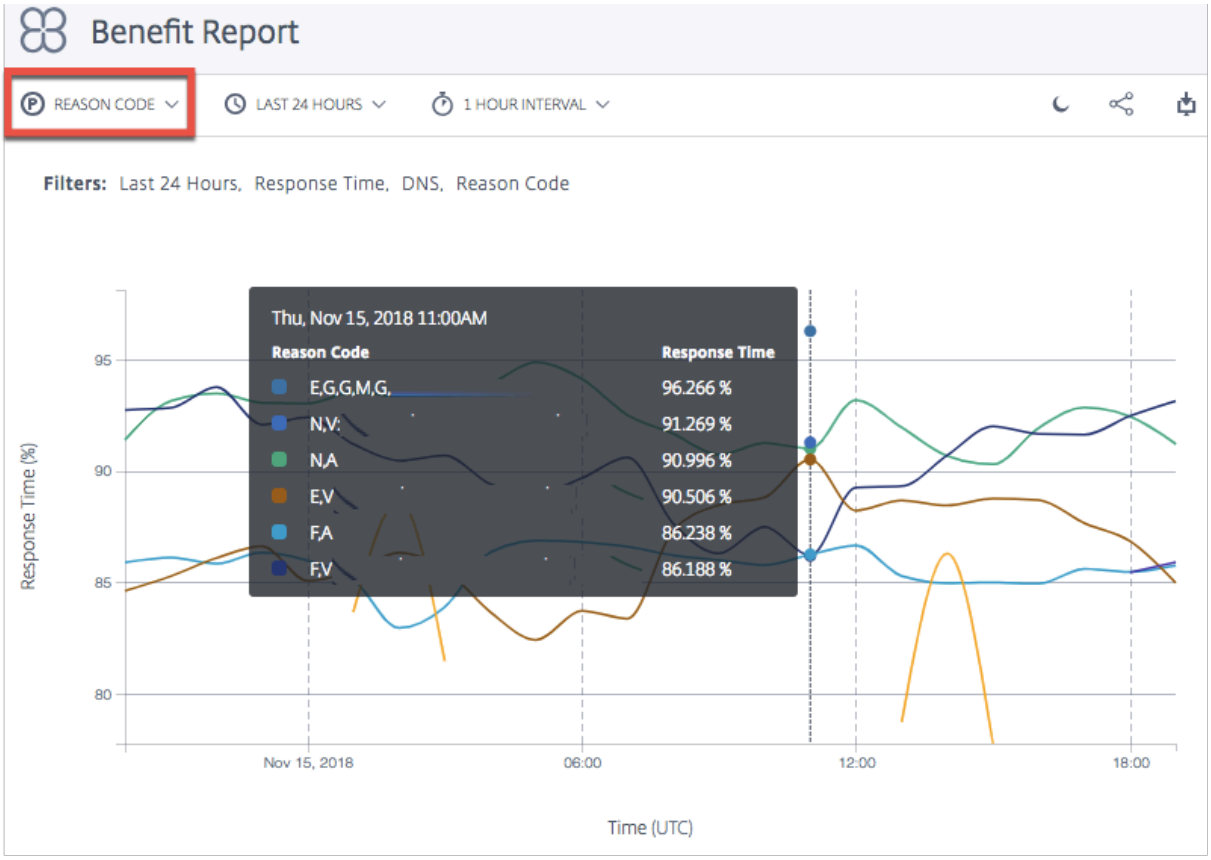
When you select **Platform** as a primary dimension, you see individual platforms chosen by different apps and the corresponding improved performance when they're chosen. The improved performance or benefit is in terms of response time or throughput (in percentage).

Note: The percentage improvement in performance shown when an app chooses that platform. The list on the chart doesn't necessarily indicate a performance ranking between these platforms.



Reason Code

When you select **Reason Code** as the primary dimension, the percentage shown in the chart is the overall average benefit when decisions are made for a specific reason code.



Ignore Platforms in Benefit Report

To improve the accuracy of **Openmix** decisions for your benefit report, you can choose to ignore certain platforms and set the app to only select from platforms that are most suitable for comparison.

For example, your application has five platforms to consider for comparison - three in Europe for European traffic, and two in the US for US traffic. Geo rules specify that the European traffic must go via the European platforms and the US traffic through the US platforms.

To ensure that computation is done using the three European platforms, you can set the app to ignore the other two non-European platforms. Use the `ignoredProvider()` method in your JavaScript.

The method takes the provider's alias (for example `provider-1`, `provider-2`) as the input argument (much like the requireProvider() method). The API must be called once per alias.

Use this sample code in your JavaScript file within the `onRequest` function:

```
function onRequest(request, response) {

  response.ignoredProvider('provider-1');
  response.ignoredProvider('provider-2');
  response.setReasonCode('Ignoring provider-1 and provider-2');
```

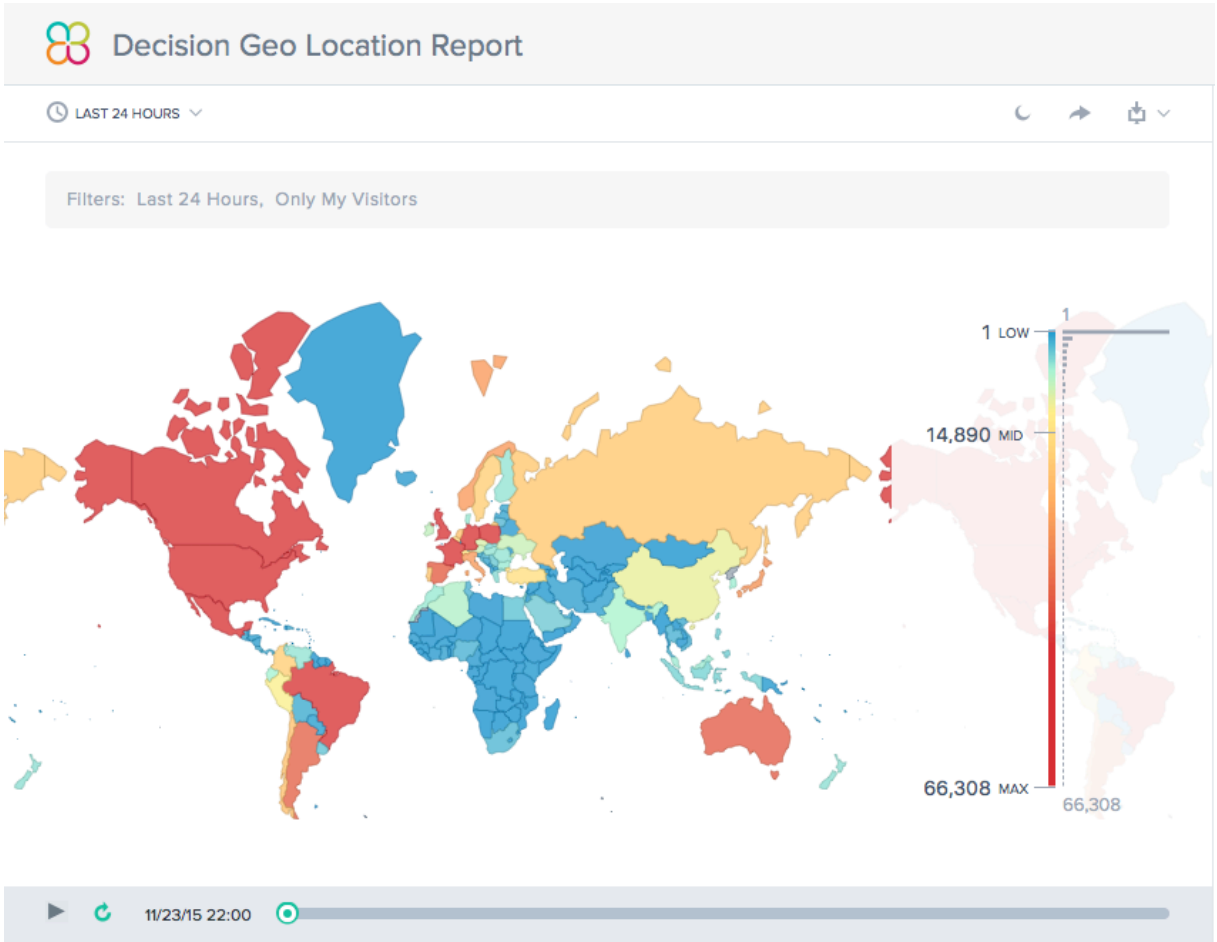


```
  response.setTTL(this.__defaultTTL);
  response.respond('provider-3', 'cmg.test.fake.cname');
 }

<!--NeedCopy-->
```

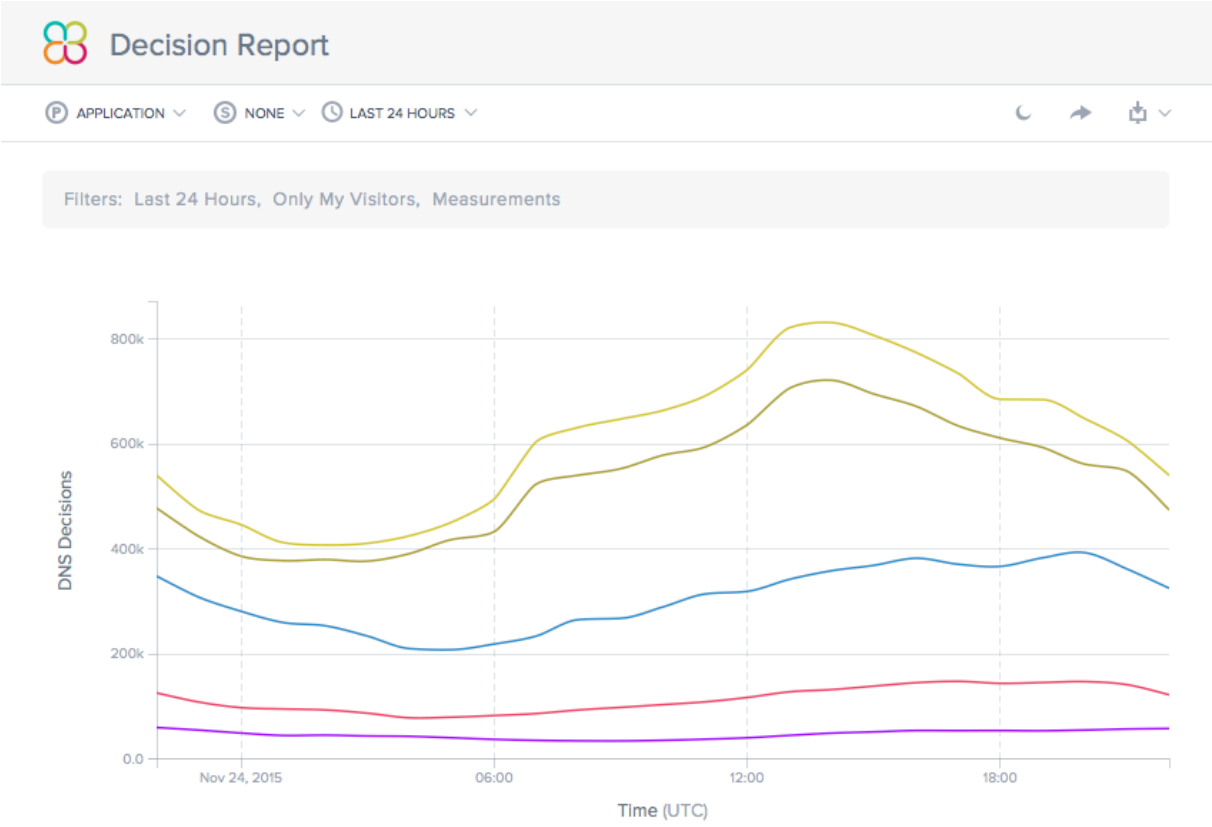
Decision Geo Location Report

This report shows the volume of Openmix decisions for each country. This map view can be viewed over time (based on the time range chosen for the report) by selecting the **Play** button at the bottom of the chart.



Decision Report

This report shows the trend of Openmix decisions for each of the applications, platforms, and geographies.



Predictive DNS

May 18, 2021

Overview

Predictive DNS is a machine-learning-based authoritative DNS platform that manages your zones and makes routing decisions based on real-time service availability. It is highly available, with multiple anycast networks, that provide flexible and reliable routing rules. It is an enterprise offering for sophisticated DNS customers who value the quality of their DNS decision making process. It is for customers who require to run a data-driven, intelligent, global traffic management policy on a robust, and high-performance infrastructure.

Predictive DNS supports primary and secondary zone creation. Zone import is also supported with the most commonly used record types such as A (IPV4 version), AAAA (IPV6 version), NS, SOA, CNAME, MX, PTR, SRV, SPF, and TXT. We also support Openmix customers with seamless integration through Openmix App records. Any number of A/AAAA/CNAME records in a zone can be made fully Openmix-intelligent at any point. Customers are also able to run Predictive DNS in a dual primary environment

using our API to drive configuration.

Predictive DNS and Openmix Integration Highlights

1. Seamless transition between static records and sophisticated, data-driven traffic management policy with zero downtime.
2. Fully configurable traffic management policies (round robin, distributed, geography-based, network-based, and so on).
3. Added real-time data awareness of global Internet traffic, endpoint health, infrastructure status, third party vendor status, and so on
4. Simple to provision or modify traffic management.
5. Deep analytics and reporting on request activity.

Steps to Set up and Delegate a Zone

Before you sign-in to the Intelligent Traffic Management Portal, here are a few high level steps to help you understand how to set up and delegate a zone.

Step 1: Define and create your zone

To begin, create a zone with the same name as your company's domain name.A zone represents a single parent domain with a collection of records within it. It provides information on how you want to route traffic for your domain and its sub domains. If you have a zone file from your current DNS provider, import it. With an imported zone file, you can quickly create all of the records for your zone.

Step 2: Add and test your records

You can either manually create records on the Predictive DNS console in the Intelligent Traffic Management Portal, or you can import a zone file with all its records. When you import a zone file, Predictive DNS replicates your original zone definition migrating all existing records within it.

You can also create zones and records programmatically by using the Predictive DNS API. The API can be found in the portal under **My Accounts > API > Configuration > authdns**.

Openmix customers can map an existing Openmix application to a CNAME or A/AAAA record through the Openmix App record type. Any number of A/AAAA/CNAME records in a zone can be made fully Openmix-intelligent at any point.

To test the records in your zone, you can use a tool called dig that queries DNS servers directly.Run dig with your zone name as the parameter.For example:

```
dig @ns1.ourdomain.net NS mydomain.com
```


`dig @ns1.ourdomain.net A host.mydomain.com`

The `@ns1.ourdomain.net` tells dig to make a request of the Intelligent Traffic Management DNS infrastructure, and the record type (NS or A) indicates which record to ask for. The NS command would ask for the NS records for the `mydomain.com` zone, and the second command `@ns1.ourdomain.net A host.mydomain.com` would be an A record for the host in the `mydomain.com` zone.

Step 4: Assign Citrix Intelligent Traffic Management as the authoritative DNS by updating your name servers

To assign us as the Authoritative DNS to manage your domain name, update the name servers that are responsible for responding to your DNS queries to our name servers. The new Citrix name servers will then respond authoritatively for your company.

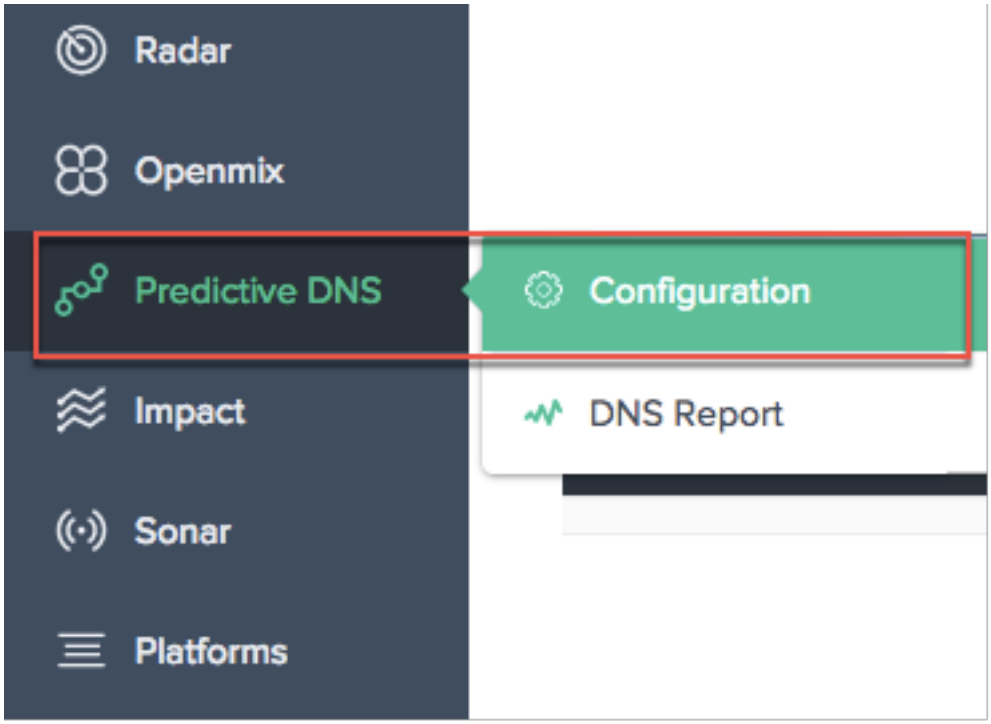
Step 5: Validate the traffic flow appropriately

Initially you see traffic running between both systems (your previous DNS service and Citrix Predictive DNS), depending on the length of the TTL in the previous system. It can take a while for the traffic to fully migrate. If you experience any errors during migration, go back to the name servers provided by your previous DNS service, and then determine what went wrong. If you see traffic flowing as expected, you have successfully migrated to Citrix Predictive DNS. The default TTL here is 3600 seconds. You may want to lower the TTL initially until you make sure the migration is successful. Once you're satisfied with the traffic flow, you can increase the TTL to a longer duration as applicable.

Navigation

To navigate to the Predictive DNS console, do the following:

1. Sign in to the Citrix Intelligent Traffic Management Portal.
2. From the left navigation menu, choose **Predictive DNS > Configuration**.



This takes you to the **Add Zone** page where you can get started by creating your zone.

Primary and Secondary Zones

A zone represents a single parent domain with a collection of records within it.
You can set up your zone in Predictive DNS as either the primary or the secondary. Primary and Secondary DNS is a way to create redundancy in the DNS. Primary is sometimes called master while the secondary is called the slave. This is because the primary has the master copy of the zone data while the secondary just clones that data through zone transfers at regular intervals or when prompted by the primary.

This process is also often called zone transfer or AXFR transfer. If your setup your primary zone with zone transfers enabled, then all changes made to the zone propagates out to all of your secondary servers automatically. Every IP that is entered as a secondary server receives this update. Similarly, you can set up a secondary zone as well.

When you create a zone, a name server (NS) record and a start of authority (SOA) record are automatically created for the zone. You can use the Predictive DNS UI to add, edit, duplicate, or delete zones.

Note: These operations (edit, duplicate, or delete) affect the entire zone, including all responses for any record within the zone. They must be done with extreme caution.

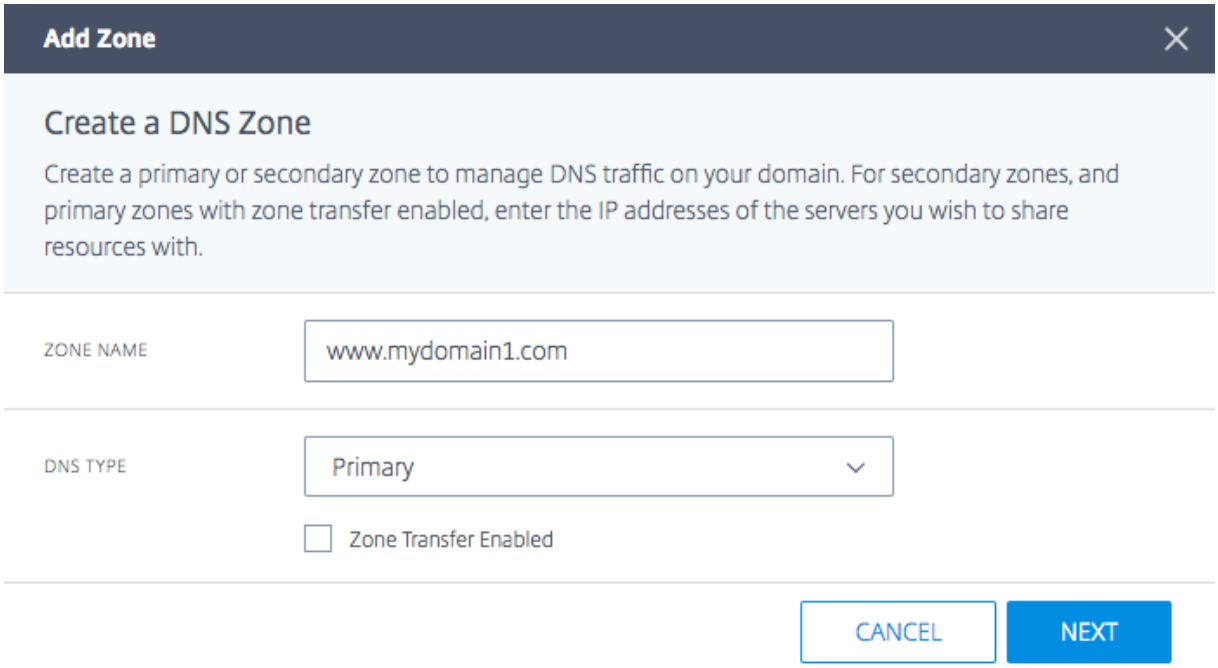
Add Zone

To add or create a zone:

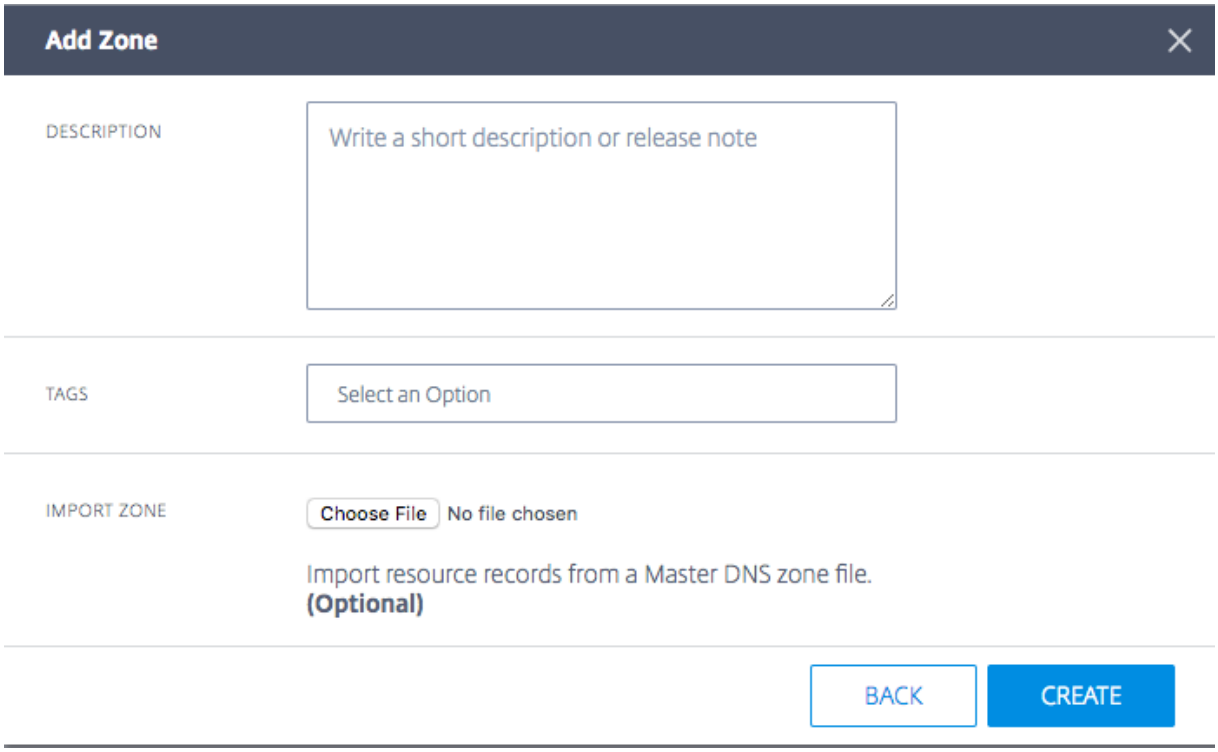
1. If this is your first time, the start-up screen shows up where you can click **Add Zone** to get started.
2. This takes you to the **Add Zone** dialog box where you can create a zone for your domain.

If this is not your first time, you see a list of existing zones (domain names) created for the domains in your company and number of records associated with each of them.

1. Click the add icon on the top right of the page to start creating a zone.
2. The **Add Zone** dialog box opens.

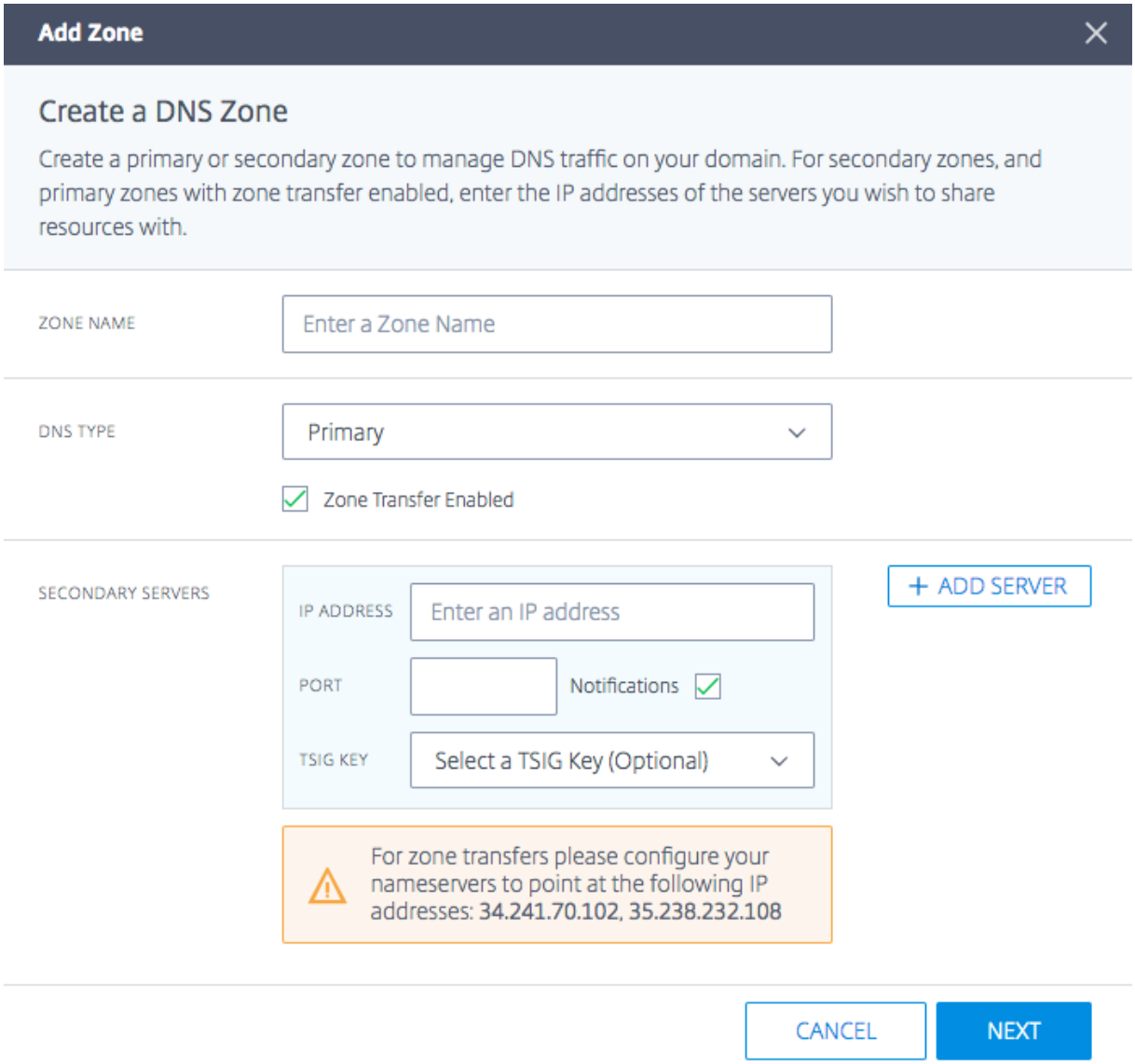


1. Enter your domain name as the **Zone Name**. For example `www.mydomain.com`. The zone name must be globally unique, which means that you cannot create a zone name that exists, or even partially overlaps with an existing zone name. However, if there is a valid scenario where you need to create a zone name that may overlap with an existing one, or if you are unable to create a zone for a domain you own, contact support.
2. Select the **DNS Type** as **Primary** or **Secondary**.
3. Click the **Zone Transfer Enabled** check box to enable zone transfer, and enter information for the **Primary** or **Secondary** server. Refer to Server Information for details.
4. Click **Next** to enter zone information such as a **Description** and **Tags**.
5. Select **Choose File** to import a zone file from your machine (if available).
6. Click **Create** to complete the addition of a new zone.



As new zones are created they appear in the list on the **Zones** page.

Server Information



IP Address

Enter the IP of the primary or secondary server.

Port

Enter the port number associated with the server. This is an optional field. It is configurable only for secondary servers. If left empty, it defaults to 53.

Notifications

Enable notifications by checking the **Notifications** check box if you want your primary DNS to notify the secondary when updates occur. If the box is unchecked then updates from the primary are sent to the secondary on regular 60 minute time intervals.

Add Server

The **Add Server** button allows you to configure multiple servers for zone transfers.

TSIG Key

You can select a **TSIG key** from the list. This list contains keys that you create and manage in the TSIG Keys section. This is an optional field for increased security. Refer to TSIG Keys for more information.

Description

Add a short description or comment regarding the zone you are about to create. This is an optional field, entirely for your own requirement. It does not affect the actual DNS responses in any manner.

Tags

Tags allow you to sort and filter your zones in a list. This is also an optional field.

Import Zone

If you have a zone import file that has the configuration for your zone, it can be imported here. To import a zone file, first create a zone with the same name as the file you are importing. The following are the requirements for import:

- The name of the zone in the zone file must match the name of the zone you are creating.
- The zone file uses a standard BIND format for records.
- The imported file must have an RFC-defined zone file format.
- You can import a maximum of 5000 records. If you need to import more than the 5000 records, contact support.

To import a zone file, do the following:

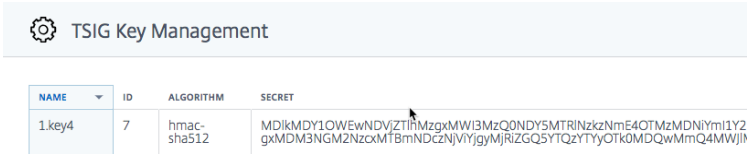
1. In the **Add Zone** dialog box, go to **Import Zone**.
2. Click **Choose File**.
3. Select the zone file you want to use to populate the zone.
4. Click **Create** to complete the process.

TSIG Keys

TSIG keys provide an extra level of security for sharing information between a primary and secondary server. The key's secret must be available on both servers (primary and secondary) in order for a successful handshake to take place.

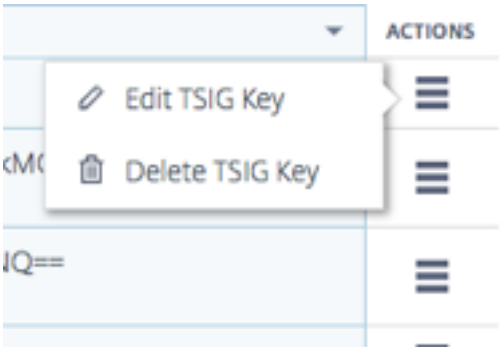
To generate and manage TSIG keys, do the following:

1. From the left navigation menu, choose **Predictive DNS**.
2. Click **TSIG Key Management**.
3. The TSIG Key Management page opens.



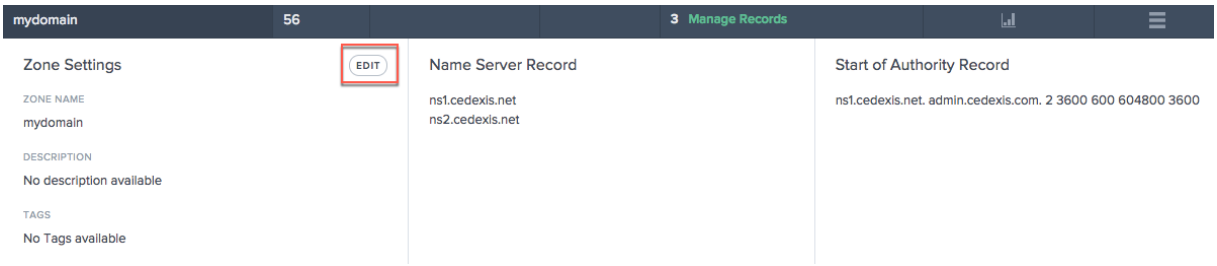
4. Click the add icon on the top right of the page.
5. The **Add TSIG Key** dialog box opens.
6. Enter a **Name** for the TSIG.
7. Select an algorithm from the list.
8. For **Secret**, you have the option to enter any word or sentence in the field. As long as what you enter is 32 characters long (without spaces) and base64 encoded, it is accepted as such. Otherwise, it is hashed according to the algorithm that you select. **Note**: The secret and algorithm values need to match between the primary and secondary systems. The value of the secret has to be base64 encoded and have a character length of 32 characters. The generate hash button is only there to help generate a hash if one does not exist already.
9. Click **Create** to complete the generation of the key. The newly created TSIG is listed on the **TSIG Key Management** page.

To edit or delete the **TSIG** key, click the **Actions** column. Choose **Edit** to modify or **Delete** to remove the key.



Edit Zone

1. Click the name of the zone you want to edit.
2. The edit drawer opens.
3. Click the **Edit** button to make changes to the zone name, description, and tags.
4. Click **Save** to save your changes.

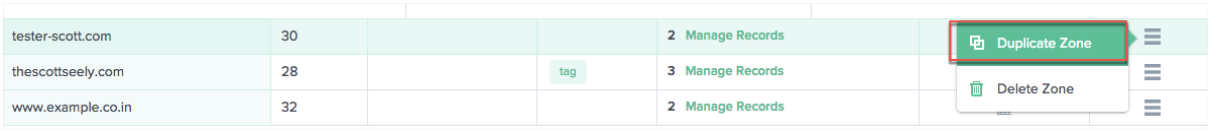


Important: Be careful when editing a zone name. Since all records in the zone are effectively suffixed with the zone name, renaming a zone changes every request.

Duplicate Zone

Duplicating a zone means to simply create another zone with information from an existing zone, but with a different zone name.

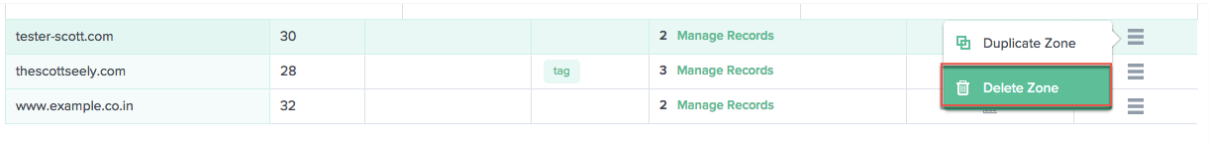
1. To duplicate a zone, click the icon in the **Actions** column.
2. Choose **Duplicate Zone**.
3. The **Add Zone** dialog box opens with information from the original zone.
4. Give the zone a new name and change whatever information you need to.
5. Click **Create** to complete the process.
6. A new zone is created with the records and information found in the original zone.



Note: You can change any information within the new zone at your own discretion. But you must change at least the **Zone Name** to create a duplicate zone. Duplicate zone names are not allowed.

Delete Zone

1. To delete a zone, click the icon in the **Actions** column.
2. Choose **Delete Zone**.
3. Click **Confirm**.



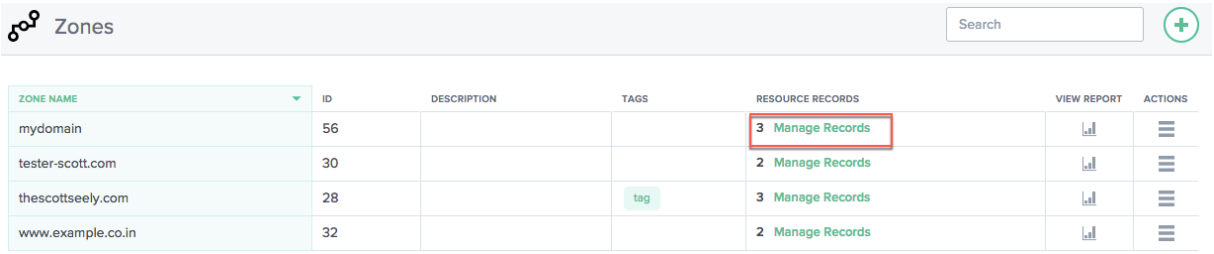
Note: This operation affects the entire zone, including all responses for any record within the zone. This must be done with extreme caution.

Records

After you create a zone for your domain (for example `mydomain.com`), you can add records to the zone. Each record you add will include a name, a record type, and other information applicable to the record type.
All records within a zone must have the zone's domain name as a suffix.For example, if `mydomain.com` is the zone, it can contain records named `www.mydomain.com`, and `www.portal.mydomain.com`, but cannot contain a record named `www.mydomain.co.in` that is, the name of each record is appended with the name of the zone.
Note: When a zone is created, the Name Server (NS) record and Start Of Authority (SOA) record types are automatically created for that zone.

Manage Records

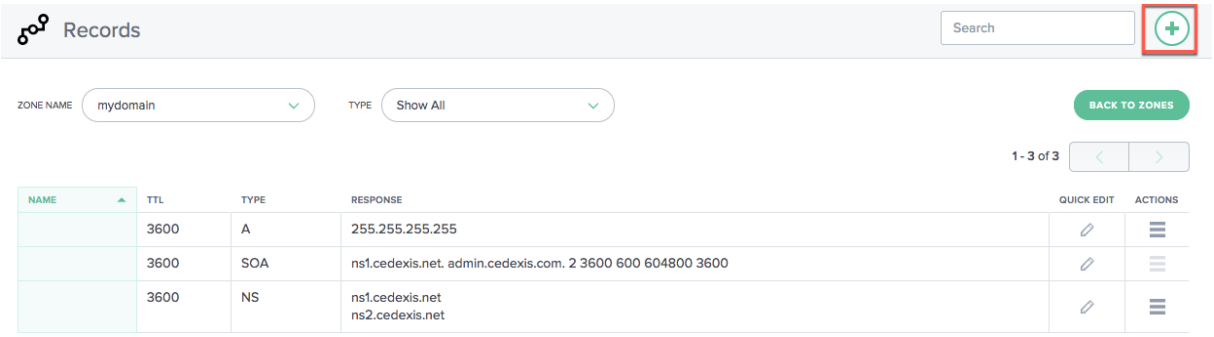
To get to the Records page and manage your records, click **Manage Records** in the **Resource Records** column of your Zone. The **Records** page opens with a list of records under the selected zone. Even if you haven't created any records yet, you see at least two record types under Resource Records for one or more zones that you created. These are the NS and SOA records that are created by default when you first create your zone.



This page enables you to add, edit, delete, or duplicate records. It also lists the TTL, Record Type, and Response for each subdomain or record.

Add Record

1. From the **Zones** page, click **Manage Records**. This takes you to the **Records page**.
2. To add a new record, click the add button on the top right corner of the **Records** page.
3. The **Add Record** dialog box opens.



Name

Enter the name of the record. If you leave this field empty, a record is created at the apex of the zone.For example, if your zone is `mydomain.com` and you want an A record at the root of this domain, you would specify this as a nameless record in the `mydomain.com` zone. Some other specifications and vendors refer to this as the @ record.

TTL

Enter a value for TTL.TTL is the amount of time, in seconds, that you want DNS recursive resolvers to cache information about this record. If you specify a longer value (for example, 172,800 seconds, or two days), resolvers will reuse a previous response and send requests to the authoritative DNS server less often. However, this means it takes longer for changes to the record to take effect because recursive resolvers use the values in their cache for longer periods instead of asking for the latest information.

Type

Select the Type of record that you would like to create. For more information on different types of records, refer to the Record Types section.

Response Type

Enter a Response that is appropriate for the value of the record Type. For all types except CNAME, you can enter more than one response value. Enter multiple response values by clicking the add icon. If multiple values are entered, all of the specified responses will be returned for each request of that type and name.

Click **Create** to add the record. The newly added record propagates out to the DNS servers and be served live when the change is made.

List Records

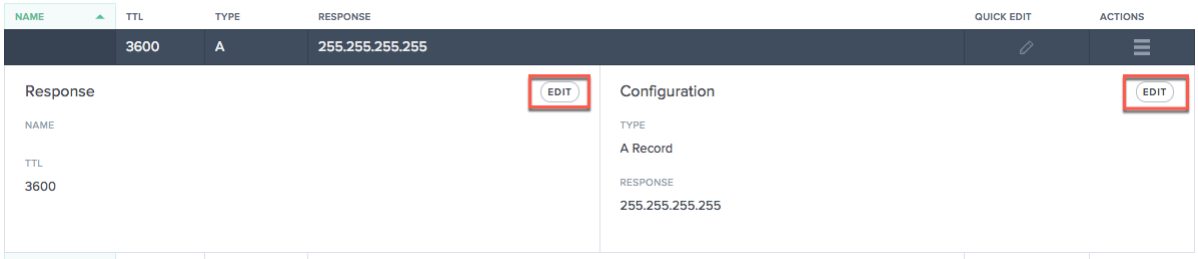
When you add a new record, it is listed on the Records page. This page lists all the records you created under a specific **Zone Name** along with the **TTL**, **Record Type**, and the **Response** for that record.

All records on this page belong to a specific zone displayed in the **Zone Name** list on the top left of the **Records** page. This list has a list of the zones already created for your company. You can switch to a different zone (and view its own records) by selecting it from the list.

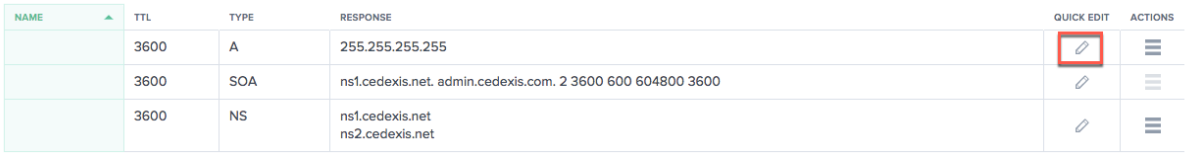
You can also use the **Record Type** list to filter this list based on record type.

Edit Record

There are two ways to edit records: detailed edit and quick edit. To perform a detailed edit, click record in the list (on the **Records** page).It opens to show the record details with buttons to edit. Click the **Edit** button to display record information. Once you're done editing, click **Save**, to save your changes.

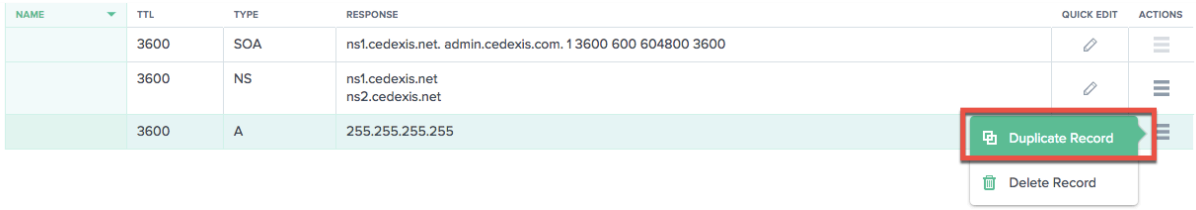


To use **Quick Edit**, simply click the edit icon (in the **Quick Edit** column) for the record you want to edit. You will be able to edit the TTL and the Response for the record. When you're done editing, click the save (check mark) icon to save your edits or cancel to undo the edits.



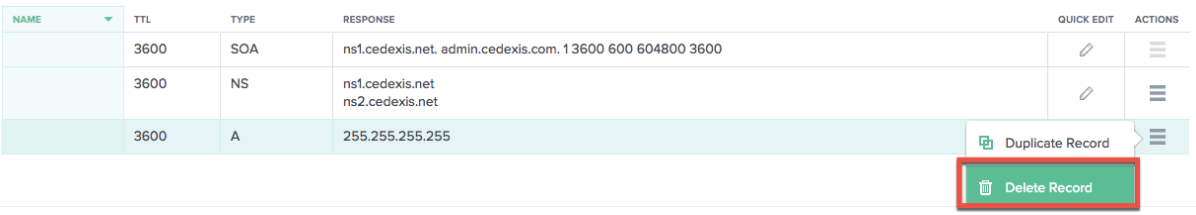
Duplicate Record

To duplicate a record click the icon in the **Actions** column. Choose Duplicate Record. The Add Record dialog box opens with information from the record you want to duplicate. Click Create to create a record with information from the original record. Please note that at least the Record Name or Type must be changed in order for the new record to be created.
Note: SOA records cannot be duplicated.



Delete Record

To delete a record click the icon in the **Actions** column. Choose Delete Record. This action deletes the record and Predictive DNS will no longer respond to queries for the record. To remove specific responses within a record, use the Quick Edit option



Note: NS and SOA records are default record types and cannot be deleted. These records will be removed only if the zone itself is deleted.

Record Types

NS Record

NS or Name Server records are responsible for delegating a DNS zone to an authoritative server. We create a name server (NS) record that is automatically assigned when you create a zone, for for example, ns1.ourdomain.net and ns2.ourdomain.net. These are the name servers you would configure in your registrar so that DNS queries can be routed to your zone.

These name servers serve to confirm the server set available to service requests for the zone, ensuring that the set of name servers returned in the delegation request, and by the delegated server, match. You can also edit the name servers to ensure they match.

We also enable you to edit name servers that you create so you can point any of your domains to another company's name servers that may hold your DNS zone and manage your records there.

Note: NS records can be edited but cannot be deleted.

SOA Record

The Start of Authority (SOA) record identifies the authoritative information about the zone. An SOA resource record is created by default when you create your zone. You can modify the record as needed.

Note: SOA records cannot be created by the user but certain parameters can be edited.

The format of an SOA record is like this: `[MNAME] [RNAME] [Serial Number] [Refresh Time] [Retry Interval] [Expire Time] [Minimum TTL]`

Here is an example: `ns1.ourdomain.net admin.mydomain.com.314 3600 600 604800 10`

The elements of the SOA record include:

- **MNAME**: The the domain name of the primary name server, such as `ns1.ourdomain.net` in the above example.
- **RNAME**: The email address of the administrator in a format with the @ symbol replaced by a period, such as `admin.mydomain.com` in the above example.
- **Serial Number**: A revision number to increment when you change the zone file and distribute changes to the DNS servers. An unsigned 32 bit integer, such as 314 in the above example.
- **Refresh Time**: Refresh time in seconds that the DNS servers wait before querying the SOA record to check for changes. An unsigned 32 bit integer time interval in seconds, such as 3600 in the above example.
- **Retry Interval**: The retry interval in seconds that a secondary server waits before retrying a failed zone transfer, such as 600 (10 minutes) in the above example. Normally, the retry time is less than the refresh time.
- **Expire Time**: The expire time in seconds that a secondary server keep trying to complete a zone transfer, such as 604800 (one week) in the above example.
- **Minimum TTL**: The minimum time to live (TTL) in seconds, such as 10 seconds in the above example.

A — IPv4 address

An IP address in IPv4 format, for example, `192.0.2.235`.The value for an A record is an IPv4 address in dotted decimal notation.

AAAA — IPv6 address

An IP address in IPv6 format, for example, `2001:0db8:85a3:0:0:8a2e:0370:7334`.The value for a AAAA record is an IPv6 address in colon-separated hexadecimal format as specified in RFC 4291/5952 representations.

CNAME — Canonical name

The is the fully qualified domain name (for example, `www.mydomain.com`) that you want Predictive DNS to return in response to DNS queries for this record. A CNAME value element is the same format as a domain name.

Important: The DNS protocol does not allow you to create a CNAME record for the root of the zone that is we do not allow nameless CNAME records. For example, if your zone is `mydomain.com`, you cannot create a CNAME record for `mydomain.com`. However, you can create CNAME records for `www.mydomain.com`, `portal.mydomain.com`, and so on.
In addition, if you create a CNAME record for a subdomain, you cannot create any other records for that subdomain.For example, if you create a CNAME record for `www.mydomain.com`, you cannot create other record types with the name `www.mydomain.com`.
Note: If a subdomain has an Openmix App record, you cannot have A, AAAA or CNAME records in the same subdomain.

MX — Mail Exchange

This is the record used in routing requests to mail servers. For example: `1 mail.mydomain.com`

Each value for an MX record contains two values:

1. The priority for the mail server which can be any 16 bit integer greater than 0.
2. The domain name of the mail server.

If you specify multiple servers, the value that you specify for the priority indicates which mail server you want email to be routed to first, second, and so on. For example, if you have two mail servers and you specify values of 1 and 2 for the priority, email always goes to the server with a priority of 1 unless it is unavailable. If you specify values of 1 and 1, email is routed to the two servers approximately equally.

Openmix (A/AAAA/CNAME)

Openmix Application customers can now have their entire record set in the zone (including static records) managed and served by the same set of services. This allows customers to make any of their hosts Openmix intelligent. So, whenever a CNAME is attached to an Openmix app, it is served with the same data-driven, dynamic, fully programmable, capability of Openmix.

For example, you can have multiple web app servers behind an Openmix App for your 'www' record and the Openmix app would decide which CNAME to respond with, using its built-in intelligent logic.

Note: An Openmix App can return a CNAME, A, or AAAA record and therefore you cannot simultaneously have an Openmix app with any of these record types using the same name.

PTR — Pointer record

PTR records are used to map an IP to a domain name, primarily for reverse DNS. Properly configured PTR records can be important for security scenarios such as validating the credibility of email senders or the reverse DNS lookup performed in SSH session establishment. A PTR record value has the same format as a domain name. For example, `hostname.mydomain.com`.

SPF — Sender Policy Framework

An SPF record identifies which mail servers are permitted to send email on behalf of your domain. It starts with v=spf, for example, v=spf1 ip4:192.168.0.1/16-all.

SRV — Service locator

An SRV record is used by voice over IP, instant messaging protocols, service discovery, and other applications. An SRV record value element consists of four space-separated values. The first three values are decimal numbers representing priority, weight, and port. The fourth value is a domain name.

The format of an SRV record is:

```
[priority] [weight] [port] [domain name]
```

For example:

```
1 10 5269 xmpp-server.example.com
```

TXT — Text

A text record can contain arbitrary text and can also be used to define machine-readable data, such as security or abuse prevention information. It is also often used for domain ownership verification (for for example you can get a certificate, register third-party tools to operate on behalf of your domain, and so on).
It just needs to contain text, for example, Sample Text Entry.

Predictive Record (A/AAAA/CNAME)

Predictive records provide various configuration options for global traffic management based on real-time service availability. Predictive records allow you to apply routing configuration across address

pools and define the behavior individually for different locations, networks, or IPs/CIDR blocks. This service combines failover and round robin routing logic to assure the highest availability, zero down-time, and seamless data-driven traffic management across platforms.

Predictive DNS customers can use the Predictive record type for CNAME, A, or AAAA response types.

As a Predictive DNS customer, when you add records to your zone, select **Predictive (A/AAAA/CNAME)** from the list of **Record Types**.

Navigation

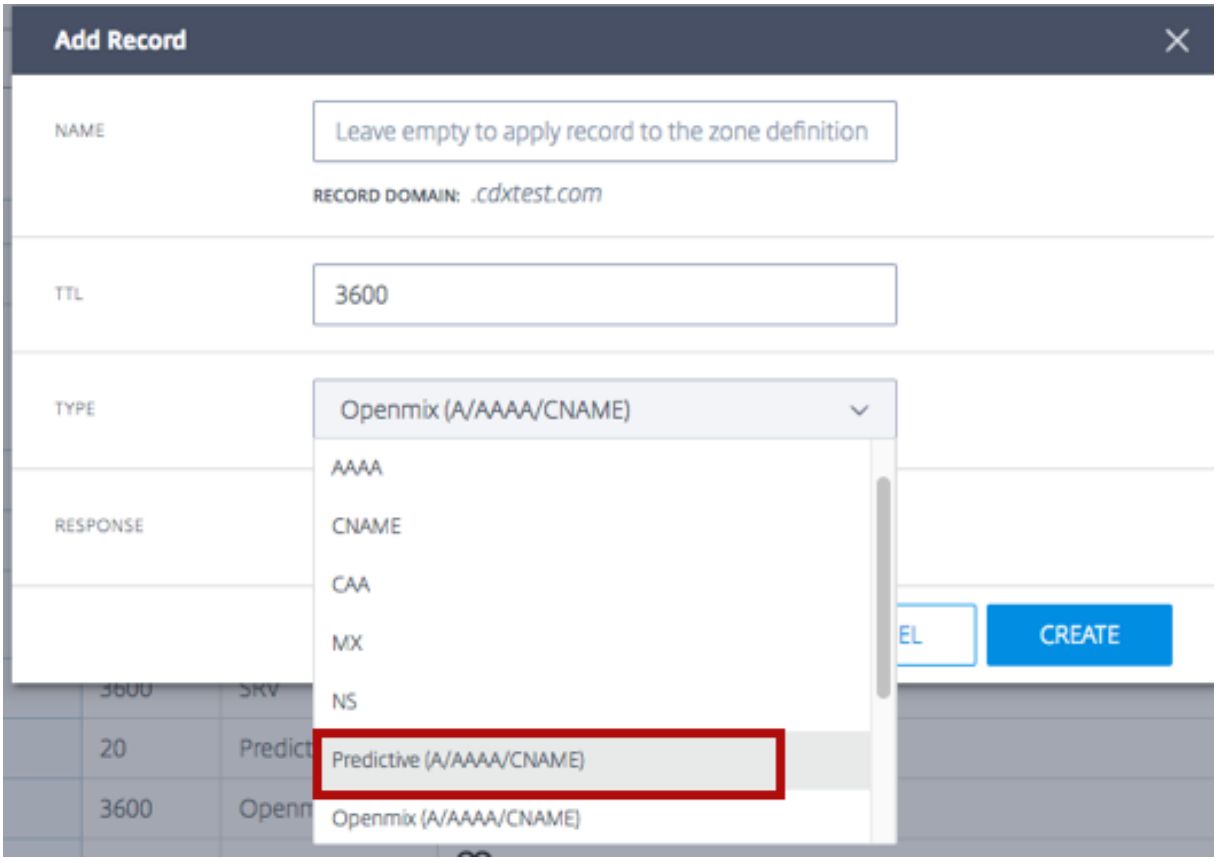
1. Go to the **Records** page of your zone.
2. Click the **Add Record button** on the Records Page. To learn more about adding records, refer to the Add Record section.
3. The **Add Record** dialog box opens.

Add Predictive Records

In the **Add Record** dialog box, enter the following:

1. **Name**: Enter a Name for the record. If left empty, the record will automatically have the zone definition. You can also use a single asterisk * as a wildcard in the leftmost part of the name to match requests for all non-existent subdomains. For example, you can use, *, *.example.com, or *.something.example.com. However, *. is invalid, that is, asterisk followed just by a dot is not allowed. We support the wildcard functionality as defined in the RFCs.
2. **TTL**: You can leave the default TTL as is, or modify it according to your need. **Note:** DNS Time to Live (TTL) tells resolvers how long they must keep the decision before asking for updates again. The TTL is used to control the volume of traffic, and also control sensitivity to changes in the data that it acts upon. The default TTL is 20 seconds. If you lower the TTL you get more volume and more real-time DNS queries. However, that may lead to added costs and lower performance (because DNS queries take time on the client). Therefore, it is recommended to not change the default value of 20 seconds.
3. **Type**: Click the **Type** list, and select Predictive (A/AAAA/CNAME).
4. **Response Type**: Click the **Response Type** list, and select your response type as A, AAAA or CNAME.
5. **Fallback**: Enter the **Fallback** response. A valid CNAME, A, AAAA must be specified for **Fallback**. The fallback is used in the event of a failure in processing of the application. **Note**: The **Fallback** response must be a valid CNAME, if the **Response Type** that you selected in the previous step is CNAME. If the **Response Type** selected is A, then the fallback response must be a CNAME or an iPV4 address. Alternatively, if the **Response Type** selected is AAAA, then the fallback response must be a CNAME or an iPV6 address.

6. Click **Create and Define Routing**.
7. The **Predictive Configuration** page opens.



Configuration Steps

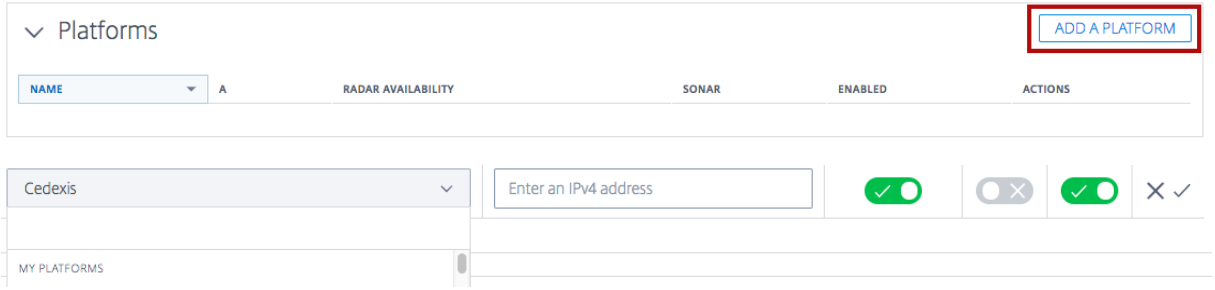
The top of this page has the **General** section that displays what your setup in the **Add Record** dialog box. It also has optional fields to add **Tags** or a **Description** to your Predictive records.

Follow the steps below to configure the record.

Step 1: Choose All Available Platforms

The first step to configuring the predictive record is to choose all the platforms that you want available for different locations, networks, or IPs/CIDR blocks. If you don't find your platform in the list, you can add it in the Platforms page.

1. Click **Add a Platform** on the top right of this section.
2. Add all the platforms that you want available for routing, including those that need to be added to address pools. You can do this by clicking the **Choose a platform** field, and selecting platforms individually from the list.
3. Depending on the **Response Type** (A, AAAA, or CNAME) that you selected in the **Add Record** list, enter an IPv4 address, IPv6 address, or CNAME for the platform. You can go back to the **General** section to edit the **Response Type**, if necessary.
4. Once the platform is selected and the **Response Type** is entered, you can enable or disable the platform by clicking the **Enabled** toggle button. You can also switch **on/off Radar Availability and Sonar** with similar toggle buttons.
5. In the **Actions** column, choose the check mark icon to save your changes or the cross mark icon to cancel.

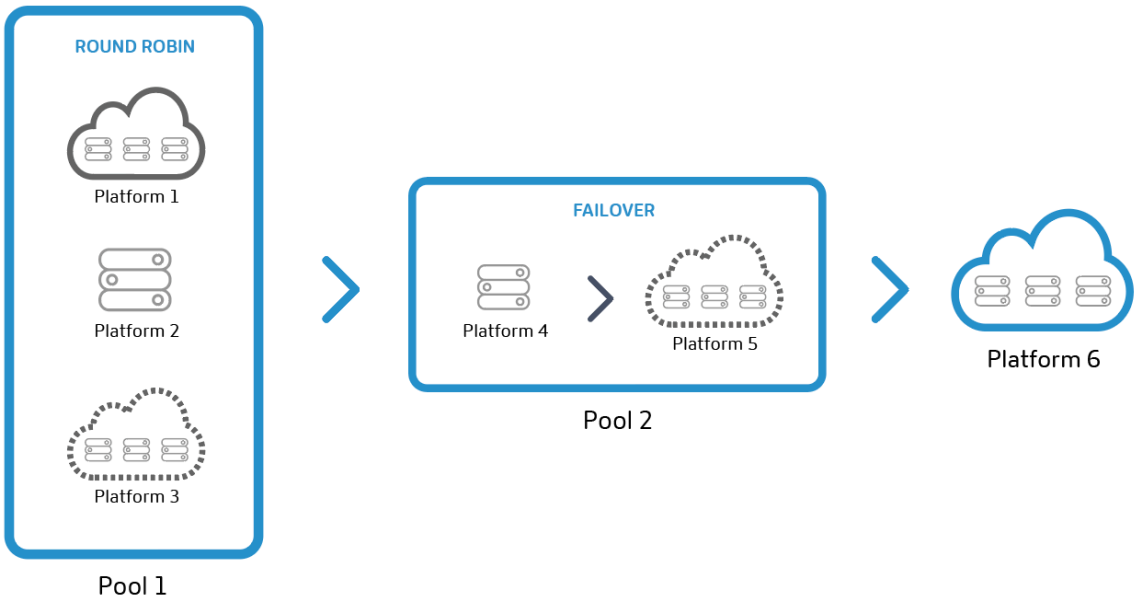


Step 2: Add and Define Address Pools

Address Pools

Address pools are a collection of platforms that follow a routing method specified by the user. The purpose of an address pool is to enable you to define logical groups of platforms that can be used with any specific routing method. You can specify **Round Robin** or **Failover** routing methods for the platforms to follow within a pool.

You can add any number of platforms in each pool, and any number of pools for each of your geographic locations. For example, you can have an EU pool (consisting of platforms that predominantly service the EU region), an Asia pool (with platforms in China, India and Singapore), and a US pool (with platforms across the United States).



Note: Address pools are optional. You can have individual platforms instead, and add them to the routing configuration.

Round Robin Routing Method

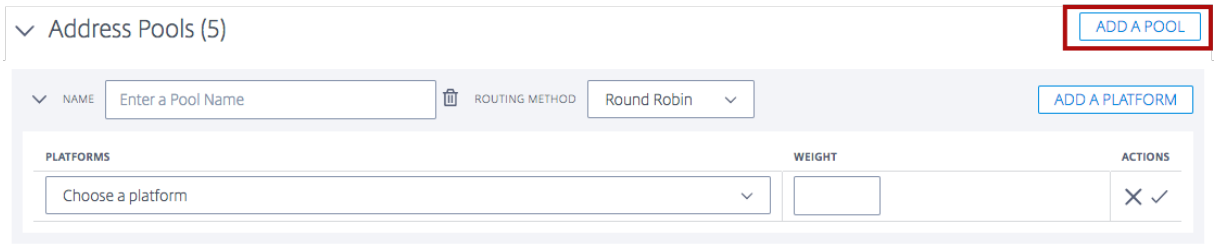
This type of routing follows a typical Global Server Load-Balancing methodology of round robin, where each CNAME/A/AAAA alternates being returned to end-users, as DNS requests are made. For example, if platforms P1, P2 and P3 meet the availability threshold —the first request is routed to P1, second to P2, third to P3, fourth to P1 again, and so on. You can also assign Weights for the prioritization and selection of each platform globally and/or by market or country.

Failover Routing Method

This routing method supports a simple routing logic where a platform is chosen based on its place in line, and its availability threshold. You can create a failover chain that decides which platform to select first, second, and so on. This failover chain can be created to work globally and/or for individual markets and countries.

Adding An Address Pool

To add an Address Pool do the following:



1. Click the **Add A Pool** button on the top right of the section.
2. Enter a **Name** for the pool. The name can be used to identify the purpose of the pool.
3. Select a **Routing Method**. You can select either **Round Robin** or **Failover**.
4. Choose a **Platform** from the list you created in the previous step.
5. You can add as many platforms to this pool as required, by clicking the **Add a Platform** button.
6. For each Platform that you choose, enter an appropriate **Weight**. The purpose of weights is to prioritize and select platforms for traffic distribution. The weights you assign to the platforms do not have to add up to 100. They can be any integer between 0 and 1,000,000. These weights when converted to percentage (in the back-end), will add up to a 100%. If all selected platforms are given the same weight, traffic will be evenly distributed across them over time. If you have only one platform, then that one will be used 100% of the time, regardless of the weight you give it.
7. When done, choose the check mark icon to save your changes or the cross mark icon to cancel.
8. You can then edit or delete your platform selection by choosing the appropriate icons in the **Actions** column.

Step 3: Configure Failover

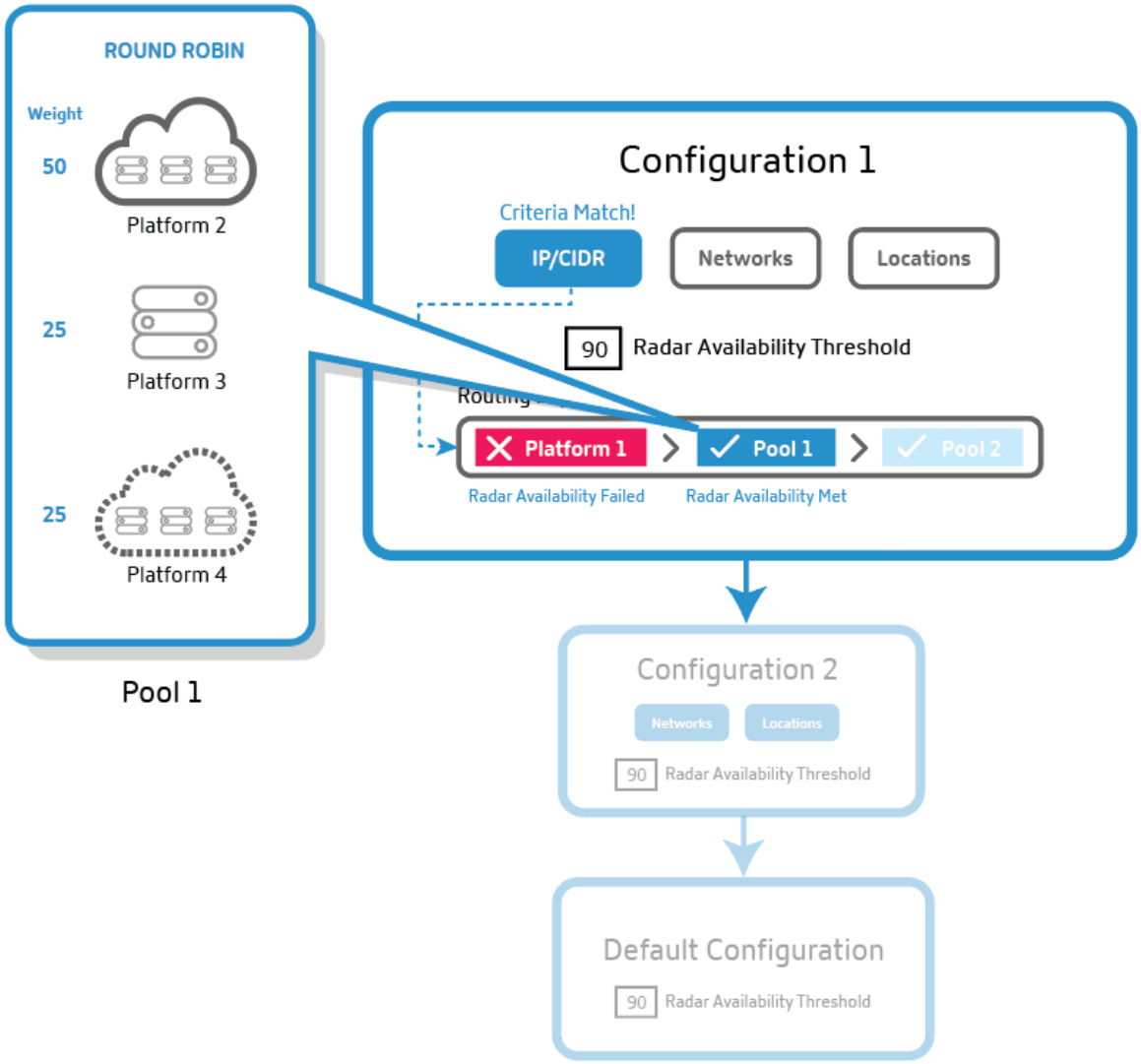
Failover applies to the entire set of address pools and/or individual platforms. It supports a simple validation method where an individual platform or pool is evaluated for routing based on the following criteria:

- Location, network, and/or IP/CIDR. At least one of these criteria needs to be specified.

  > **Note:**
  >
  > Location criteria for failover should not contain a mix of continents and countries but you can use the routing logic to create multiple failovers.

- Sonar and Radar Availability if configured, and
- Place in line

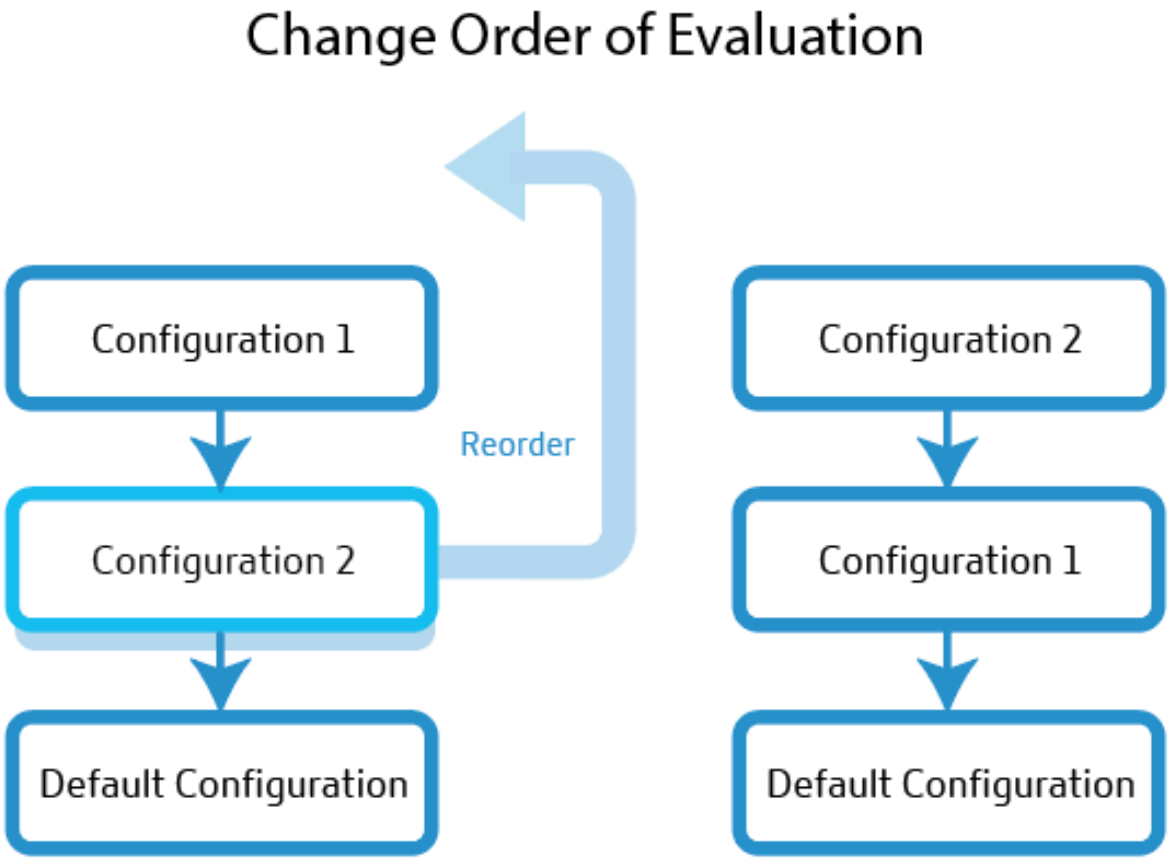
Failover For Predictive Records



1. The Predictive record evaluates the first configuration block for the required criteria (location, network, and/or IPs). If the first routing configuration block does not meet the required criteria, it moves on to the second one in line and so on.
2. The configuration block that meets all the required criteria, is chosen for traffic distribution.
3. Within the chosen configuration block, the address pools or platforms are evaluated based on their place in line and availability threshold (Radar and Sonar).
4. The first platform within the address pool (or outside it) that meets the availability threshold, is selected for traffic distribution. Round Robin or Failover routing logic then comes into play.

Note: If there is only one platform in the pool, that platform is selected 100% of the time, and round robin logic will not apply to it.

As a user, you can arrange the routing configuration blocks in such a way that the one with the highest priority comes first in line and so on. The reordering can be done manually by dragging each pool or platform to where it needs to be in the line.



Default Configuration

You are required to have at least one platform or pool in the default routing configuration block. It must contain one or more platforms or pools that the Predictive record will use if all other options fail to match the specified criteria. The default does not have any criteria to specify and it matches all requests. If the platform availability does not meet the Radar Availability threshold, then the response returns fallback.

Steps to Configure Failover

To define the configuration, do the following:

1. Enter a **Name**. This name helps identify your routing configuration block.
2. You can leave the default TTL as is, or modify it according to your need.

3. Make sure that **Radar Availability** is checked. You can set radar availability threshold to your desired level. Unchecking this disables Radar for the set of pools or platforms.
4. Select **Locations, Networks, and/or IP/CIDR**. For example, if your routing configuration applies to the Oceania region, you can specify locations, networks, and /or IP addresses of platforms or pools in this region.
5. The **Failover Configuration** field allows you to set the selection precedence for all the pools and platforms. The order in which you place these pools or platforms, will determine their selection for routing. And traffic will be routed based on the method specified (round robin or failover) in the previous step.
6. To delete a configuration block, click the trash icon beside the **Name** field.

DNS Reports

DNS reports provide powerful visibility into the volume of DNS requests based on various criteria for a specified domain or host name.They show how often specific record types are queried and provide a whole different level of drill down. This degree of granularity enables Predictive DNS users to understand trends and query volumes for specific zones, host names, request types, markets, countries, regions, states, and networks.

These reports are primarily used for better visibility and analysis.They give traffic flows for each zone or host name and help diagnose DNS related issues.They also reveal anomalies such as spikes in requests or other irregularities, by breaking down the volume of requests by record types and geographic locations.

You can also filter unnecessary noise by knowing which zones serve the most traffic, and focus only on the zones or record types that you care about.

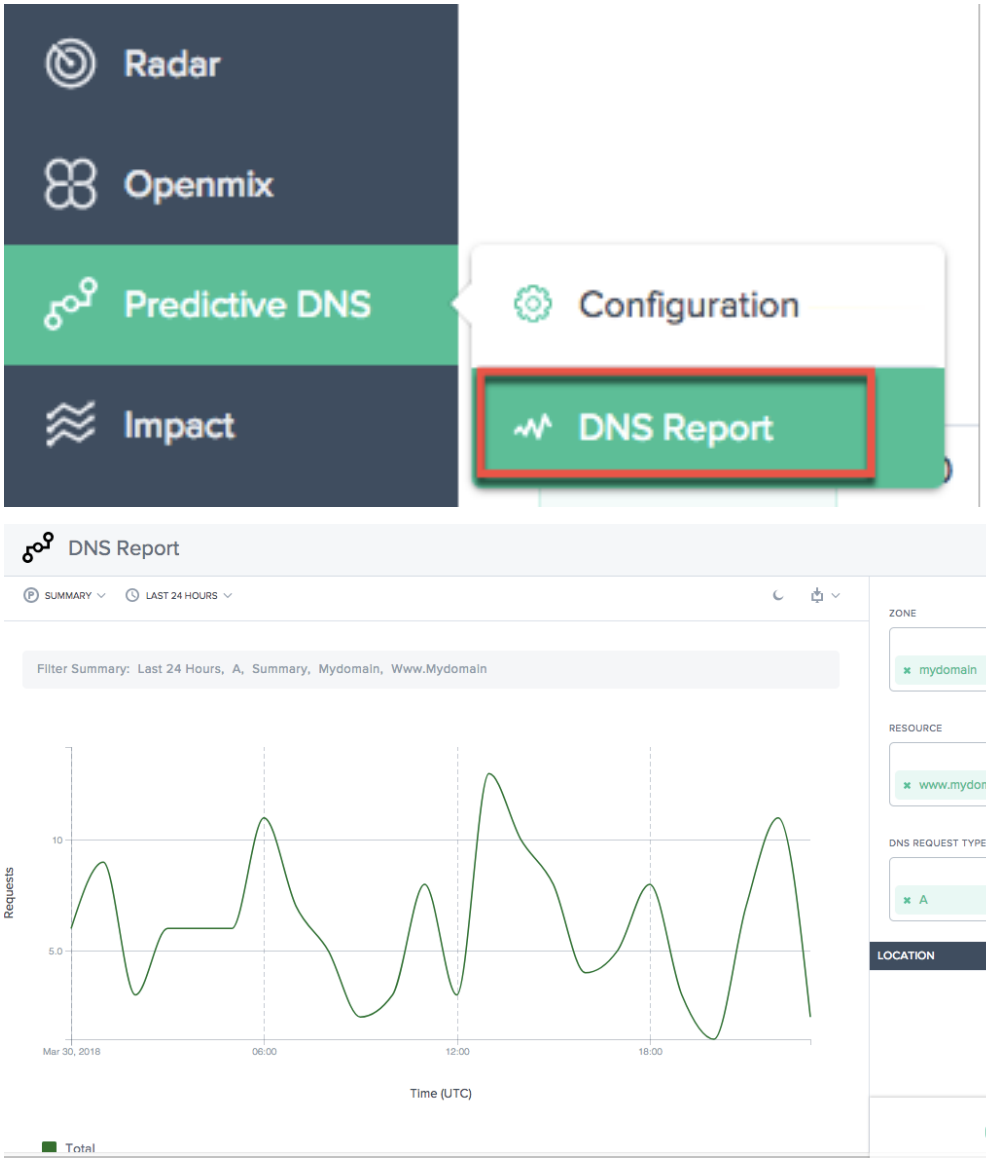
DNS vs. Openmix Reporting

For Openmix customers, reports appear within DNS reporting and within Openmix decision reports. DNS reporting provides information on requests made to our authoritative zones, while Openmix provides reports on when the Openmix intelligent platform was used to fulfill a request, either through an Openmix application record or directly to an Openmix CNAME.

Navigation

To navigate to the **DNS Report** section:

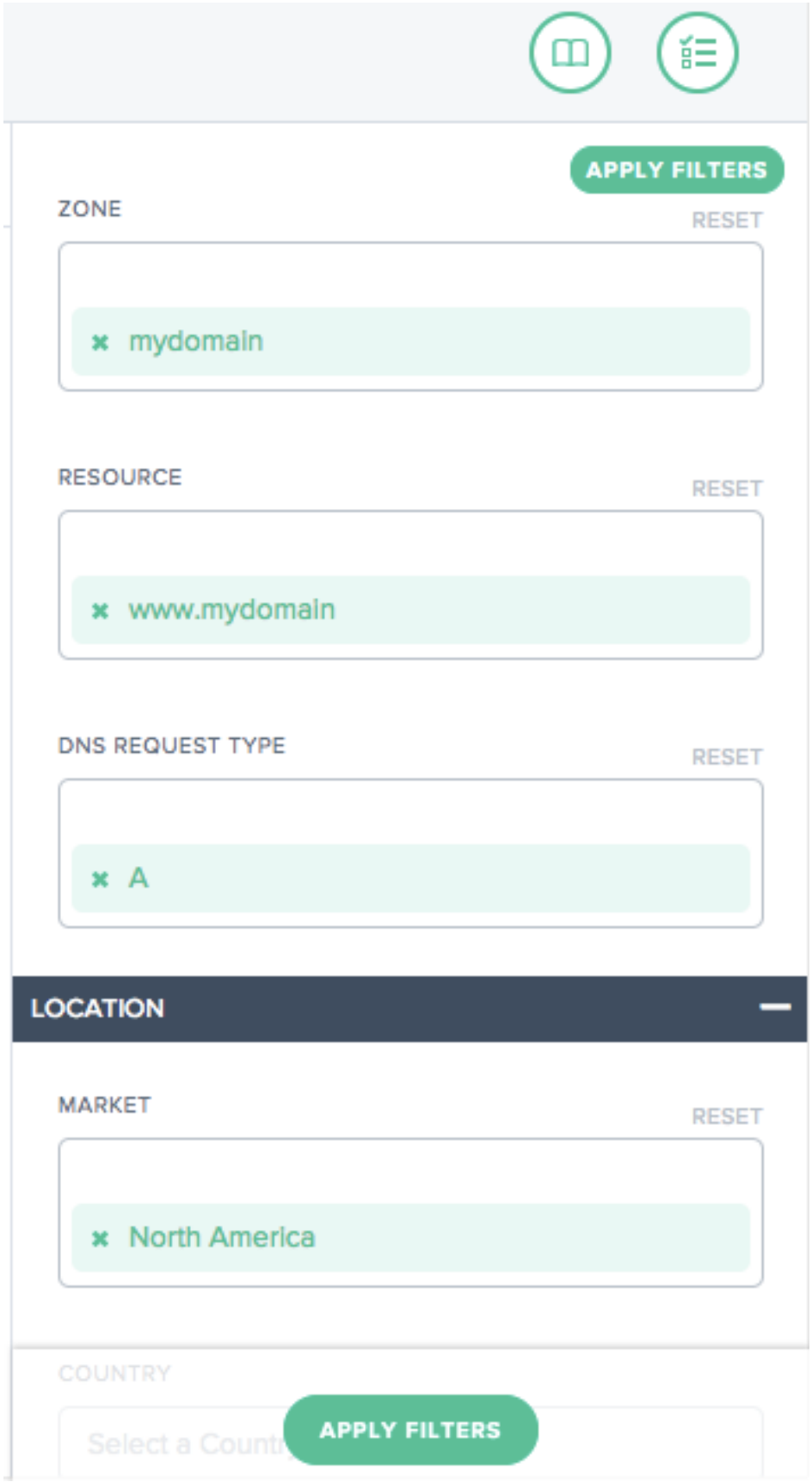
1. Click **Predictive DNS** in the left navigation menu.
2. Navigate to **DNS Report**.
3. The **DNS Report** page opens.



Apply Filters

The **Apply Filters** panel on the right helps you select and view only the data that you want displayed on the report.
You can filter based on the following:

- **Zone** – Select one or more zones to include.
- **Resource** – Select one or more host names to include.
- **DNS Request Type** – Select one or more DNS request types to include.
- **Location** – Select one or more geographic locations (Market, Region, State, or Network) to include.



Primary Dimension

Primary dimensions are selected through lists above the chart. You can use this as a powerful pivot on the report.

Summary

The Summary gives you the total number of requests with the complete set of the filters applied.

Filter by Preset Time Ranges

Relative preset time ranges can be chosen as an extra filter to further refine the reporting.

Bookmarking Reports

Once you generate a report based on the filter criteria, you can save the filters applied by bookmarking the report. Every time you visit this bookmark, an updated report is generated based on all the selected filters.
To bookmark a report do the following:

- Click the bookmark icon on the top-right of the page.
- In the Add New Bookmark dialog box, give an appropriate name to the bookmark and click Create.
- A new bookmark is now created. You can access the bookmark by clicking the bookmark icon (on the top-right corner of every report page) and selecting the bookmark.

Sonar

December 17, 2020

Sonar is a liveness check service that can be used to monitor web-based services for availability. Sonar works by making HTTP or HTTPS requests from multiple points-of-presence around the world to a URL that you specify.

Sonar Basics

Endpoints tested by Sonar are considered up or down based on the following criteria:

- Requests that result in HTTP 2xx are viewed as successes and any other result, including network issues and timeouts, are treated as failures.

- Sonar follows redirect responses that return 3xx status codes, for up to 6 redirects, until it receives non-3xx response or an error occurs.
- The endpoint status is decided based on a quorum of the reporting locations. Sonar reports whichever result (success or failure) is returned by most the points-of-presence.

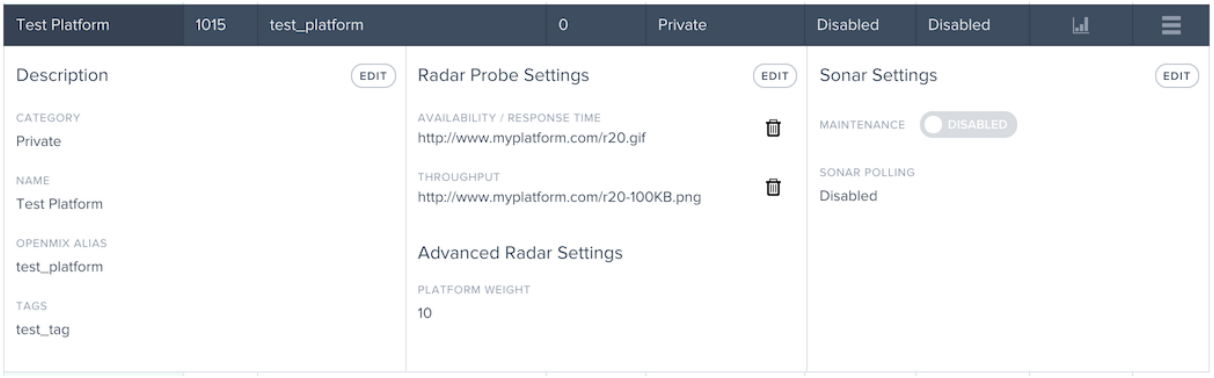
Sonar checks are performed from multiple test locations from around the world. The locations include:

- Singapore
- South Carolina, United States
- Tokyo, Japan
- St Ghislain, Belgium
- Washington, United States
- New York, United States
- London, England
- Hong Kong
- Frankfurt, Germany
- Dublin, Ireland
- Iowa, United States
- Virginia, United States
- Amsterdam, Netherlands

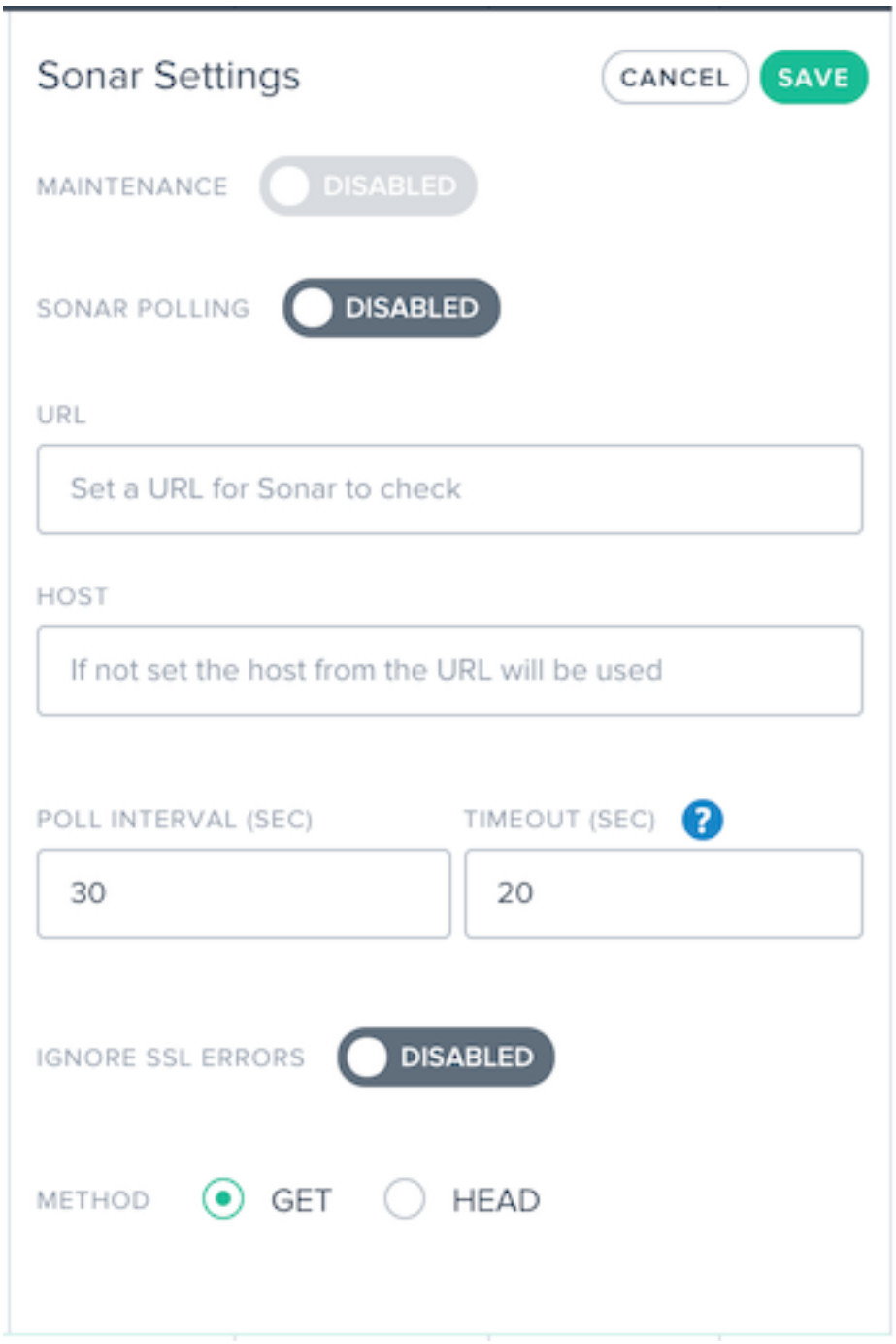
The Sonar platform is tightly integrated with the global Radar, Fusion & Openmix platform services. Sonar data is fed in real-time to all Openmix nodes around the world, to be used as an extra input for decision-making.

Platform Sonar Configuration

Sonar is configured for each platform in the Platforms page. Click a platform in the list to see the **Sonar Settings** section.



To add Sonar monitoring to the platform, click the **Edit** button in the **Sonar Settings** section.

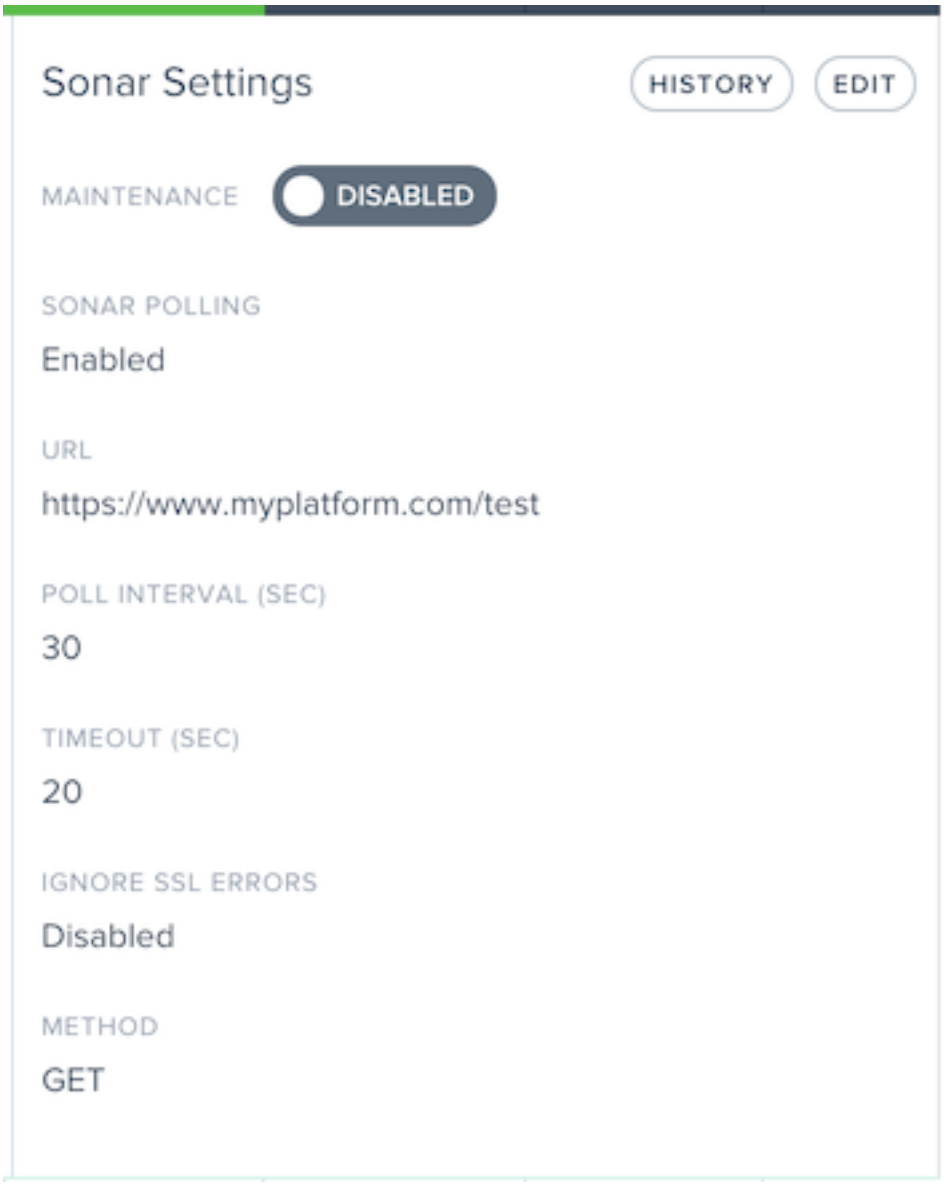


A description of the fields is below:

Input Item	Description	Default
Maintenance	When enabled, Sonar will report the service as being down regardless of actual status. This is useful when wanting to remove a platform from Openmix routing in anticipation of downtime.	Disabled
Sonar Polling	If enabled, Sonar will checks will be made on the configured URL.	Disabled
URL	The URL Sonar calls to check for availability of the service.	
Host	The value that must be used for the Host header value in the request.	v
Poll Interval	The frequency specified in seconds to test for availability of the service. Checks can have a minimum interval of every 1 second up to 300 seconds (5 minutes).	60 v
Timeout	The amount of time specified in seconds to wait for a response before assuming a failed check to the service. Checks can have a minimum timeout of 1 second up to 30 seconds. For lower poll intervals, such as below 5 seconds, the timeout is capped at 4 seconds.	20
Ignore SSL Errors	When enabled, Sonar will ignore SSL errors that occur during the request such as a mis-configured SSL certificate.	Disabled

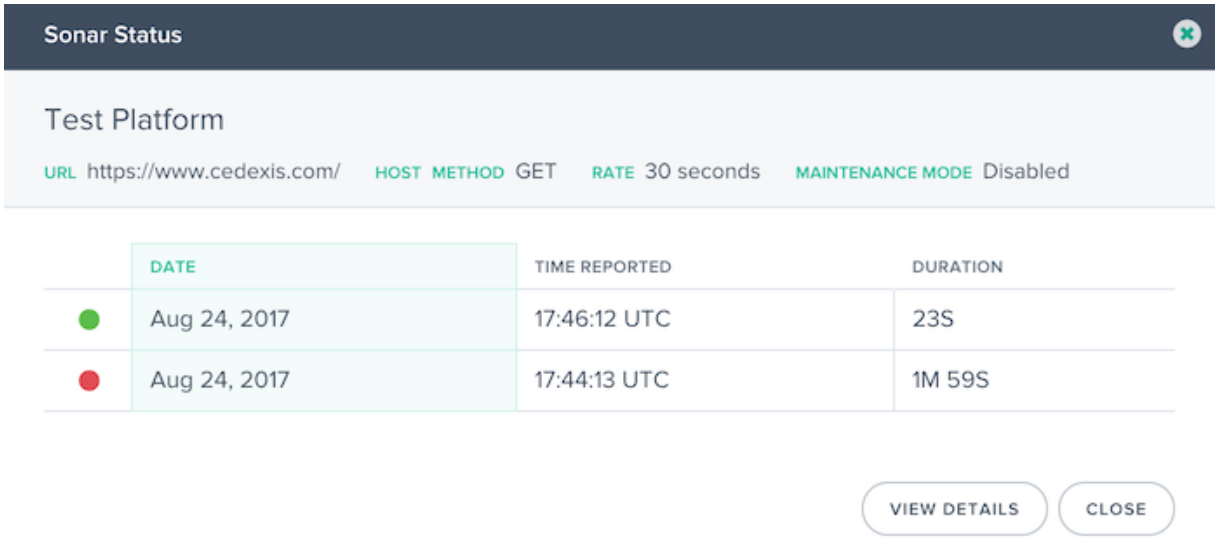
Input Item	Description	Default
Method	The HTTP method used for the check: GET or HEAD.	

To turn on Sonar, toggle **Sonar Polling** to **Enabled** and enter the service URL. Click **Save** and the checks starts.



When Sonar is enabled, the Settings display the current Sonar settings.

After Sonar has been enabled, you can click the **History** button in the **Sonar Settings** section to see the recent status changes and duration. Click the **View Details** button to go to the Sonar Platform Status page for more details and long-term status reporting.



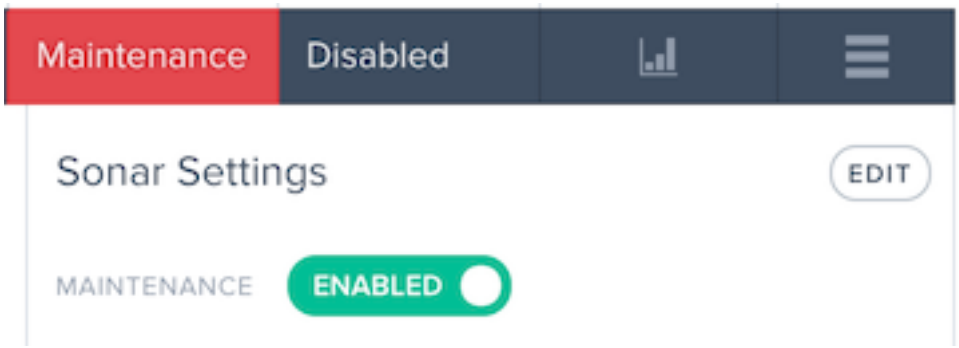
Platform Sonar Status

When Sonar is enabled for a platform, the Sonar status is shown in the platform list in the **Sonar** column. When Sonar monitoring checks against the platform, the column cell is green and displays the amount of time the platform has been reachable.

If the platform monitoring checks have failed, the **Sonar** cell is red and will display the amount of time the platform has been unreachable.

Maintenance Mode

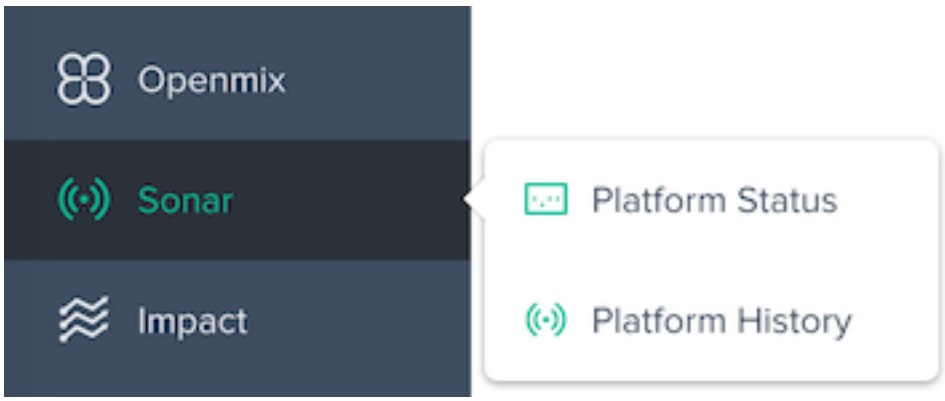
The Sonar status displays the availability of the service based on the success or failure of the synthetic checks. If you want to mark the platform as **down** even if it is reachable, in anticipation of maintenance on the platform for example, you can enable Maintenance Mode. This mode reports the platform as unavailable in the Openmix applications and will automatically stop traffic from being delivered to the platform in any Openmix application that has Sonar enabled.



The enable Maintenance Mode, toggle the **Maintenance** option to **Enabled**.

Once enabled, the platform list item displays the Sonar status as **Maintenance**.

Sonar menu



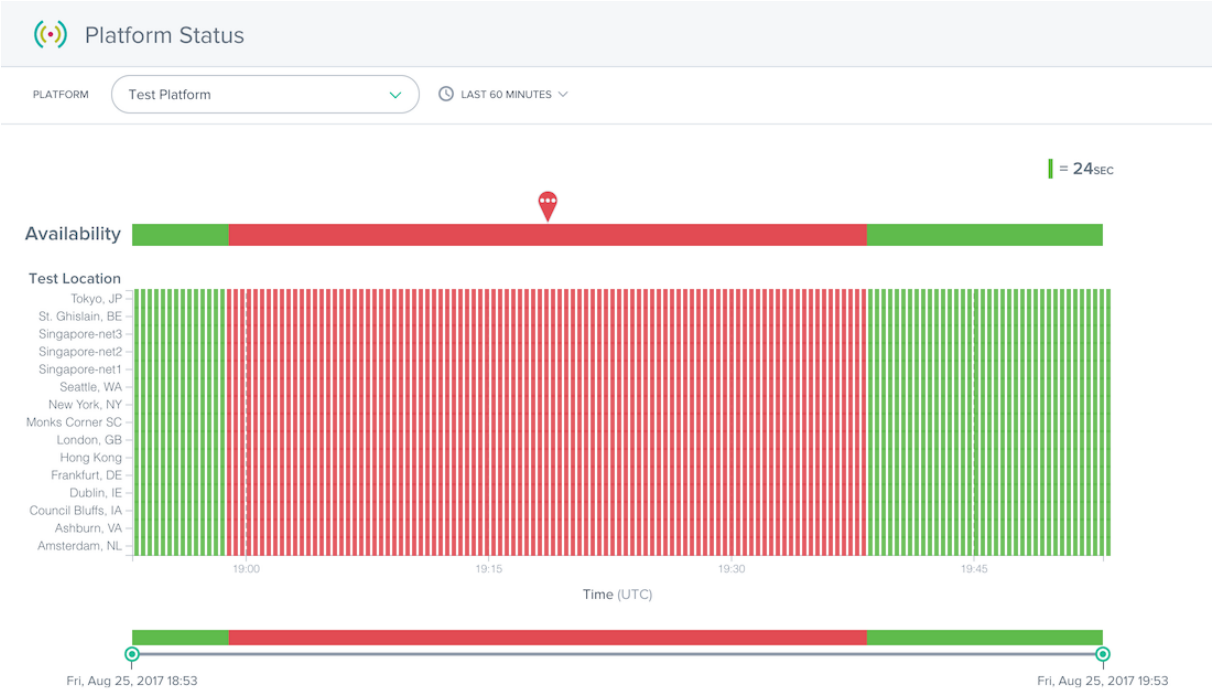
The **Sonar** menu is made up of the following options:

1. **Platform Status** – Detailed results per testing location and the overall availability status.
2. **Platform History** – Overview of the availability status over the last three months.

Platform Status

The Sonar Platform Status report shows details of the checks done by each test location and the overall status calculated from the aggregated data.

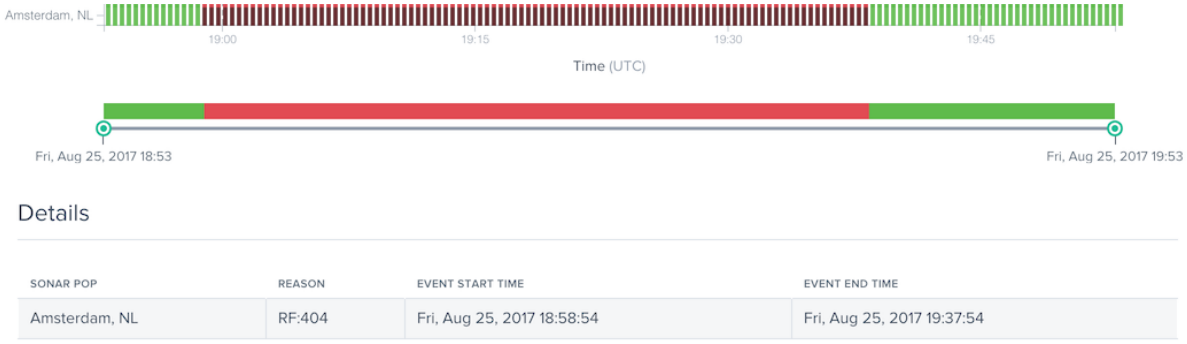
To get information about a specific platform, select a platform in the **Platforms** menu.



The Status report contains the following sections:

- Availability: At the top of the report is the Availability reported to Openmix based on the aggregated results from the individual test locations. This is the Sonar status that was used in the Openmix applications during the times specified.
- Test Locations: The results from each test location are shown.
- Time slider: The time slider allows you to easily drill into detailed time periods. Drag the time sliders to adjust the time period of the report and see more detailed time intervals.

The details of failed checks can be viewed by clicking a red marker in a test location row. The details for the test failures will be shown in the **Details** section below the report.

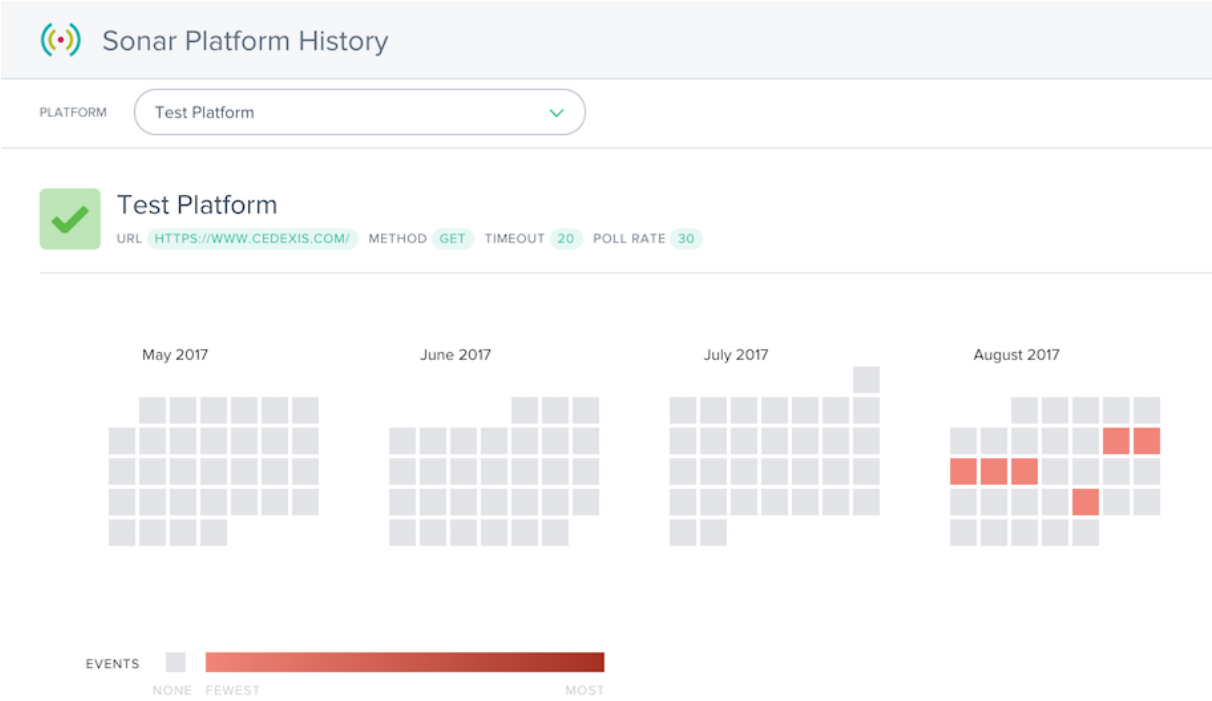


The **Reason** column provides details such as the error code that was returned from the Sonar checks that occurred in that test location.

Platform History

The Sonar Platform History report shows the availability status of the aggregated checks done by each test location over the past few months.

To get information about a specific platform, select a platform in the **Platforms** menu.



The History report displays a calendar of the last few months. The days that have service outages are shown in gradients of red. The more availability events that occurred on the day, the redder it is displayed.

Below the calendar is a list of service outages that occurred and some basic details about the events.

Details

DATE	OUTAGES	START TIME - FIRST OUTAGE	END TIME - LAST OUTAGE	DURATION
2017-08-11	1	21:29:35	23:59:59	2 hours, 30 minutes, 25 seconds
2017-08-12	1	00:00:00	23:59:59	24 hours, 0 minutes, 0 seconds
2017-08-13	1	00:00:00	23:59:59	24 hours, 0 minutes, 0 seconds
2017-08-14	1	00:00:00	21:21:18	2 days, 23 hours, 51 minutes, 43 seconds
2017-08-15	3	14:50:00	15:50:05	0 hours, 4 minutes, 3 seconds
2017-08-24	3	17:44:12	18:03:21	0 hours, 15 minutes, 25 seconds

You can click the calendar day or the date in the **Details** columns to load the Status report for more details about the service outage.

Impact

April 13, 2020

Impact offers a powerful view into the performance and business KPI data collected while visitors are on your site. Click the link for the reporting data you are interested in to see more details.

Cloud Platform Visualization Reports

The **Impact** menu is made up of the following options:

1. Navigation Timing Data – Page-level performance details, also known as our Page Load Time reports.
2. Video Playback Data – Quality of Experience and Video Delivery data.
3. Resource Timing Data – Performance details of individual resources on pages.

Navigation Timing Data

December 17, 2020

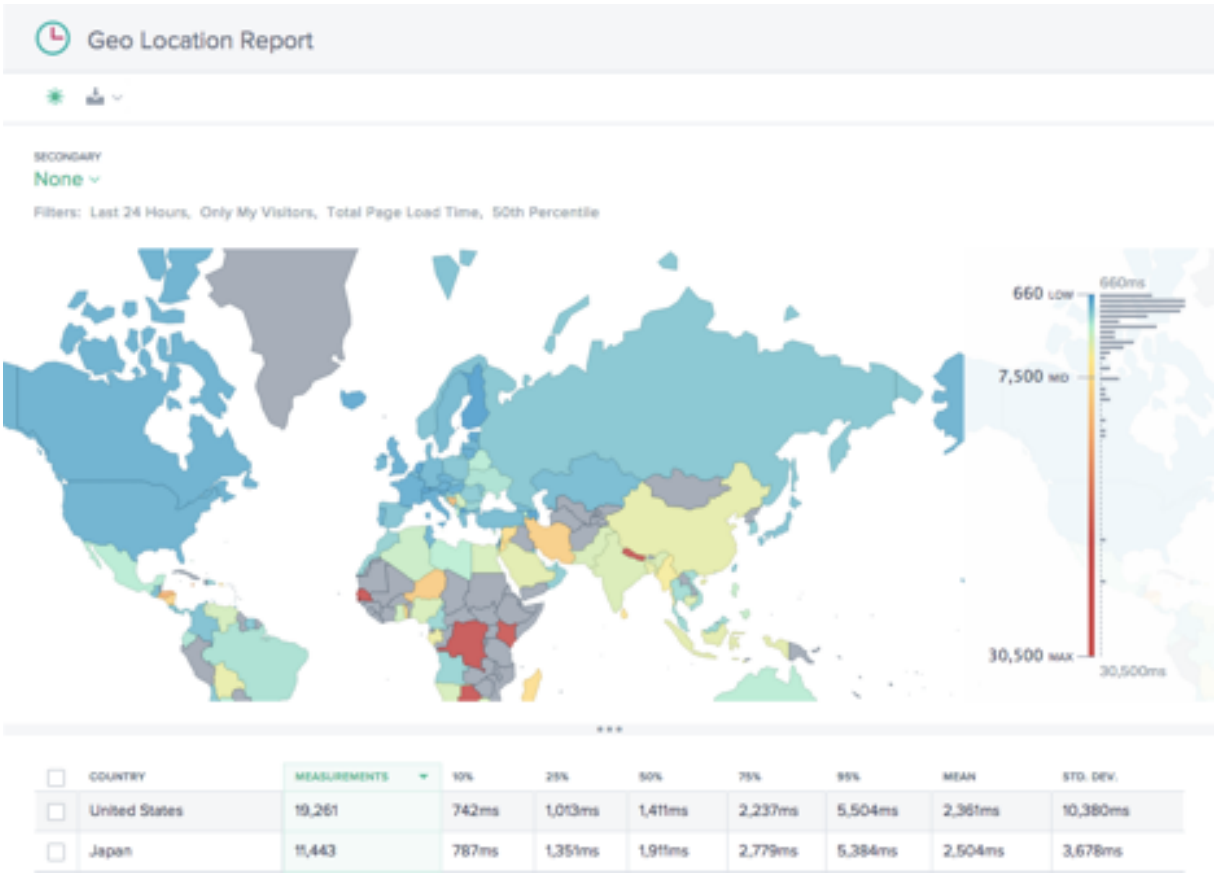
Navigation Timing reports offer a powerful view into the rich page load and event performance data collected while visitors are on your site. Following a brief description of the reports are details on how to pivot, filter, and customize the Navigation Timing reports.

Navigation Timing Reports

The **Navigation Timing** menu includes the following reports:

1. **Geo Location Report** – Report of Navigation Timing by Geographic dimension.
2. **Performance Report** – Navigation Timing measurement data over time.
3. **Statistical Distribution Report** – A view of Navigation Timing data through a statistical distribution reporting view.

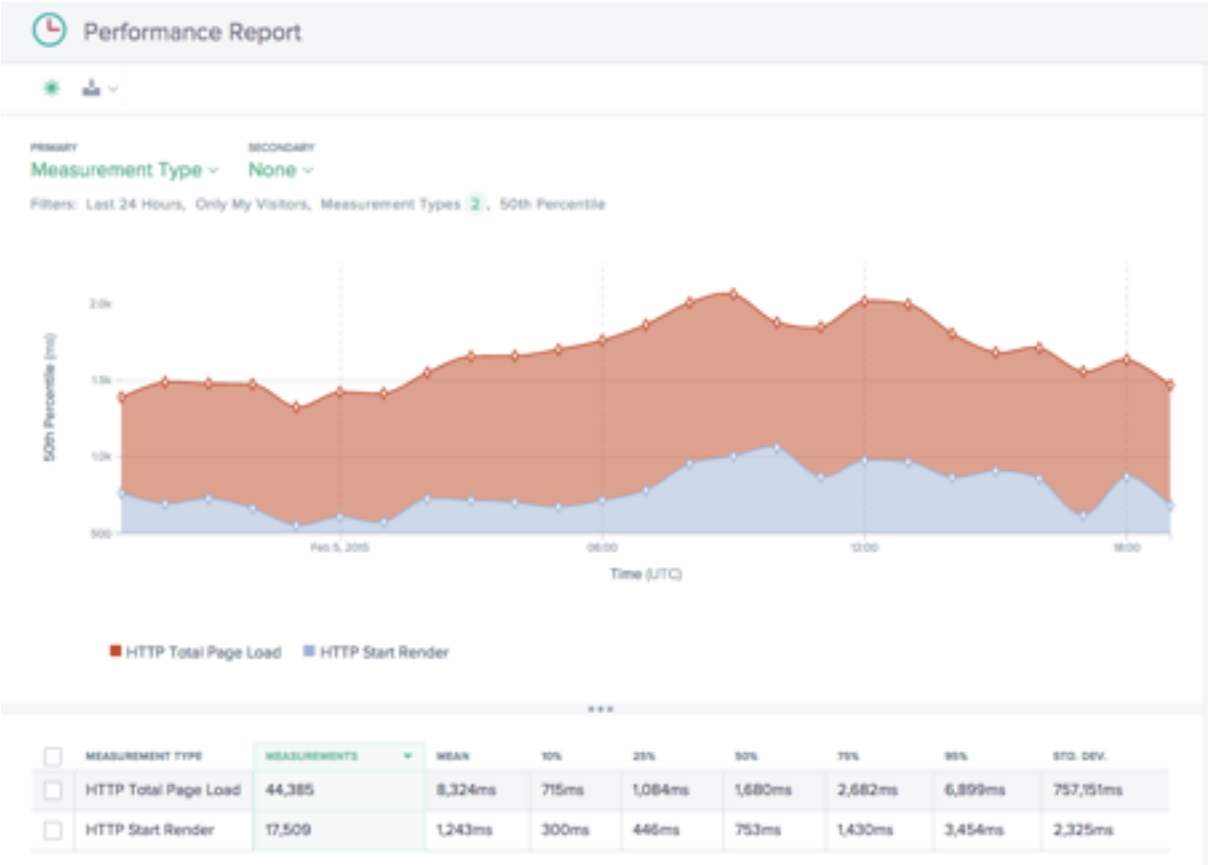
Geo Location Report



This report shows the page load time performance for each country. Zoom into the map to see greater granularity as needed.

The table lists each country with its associated page load time performance, along with the number of measurements (Page Views).

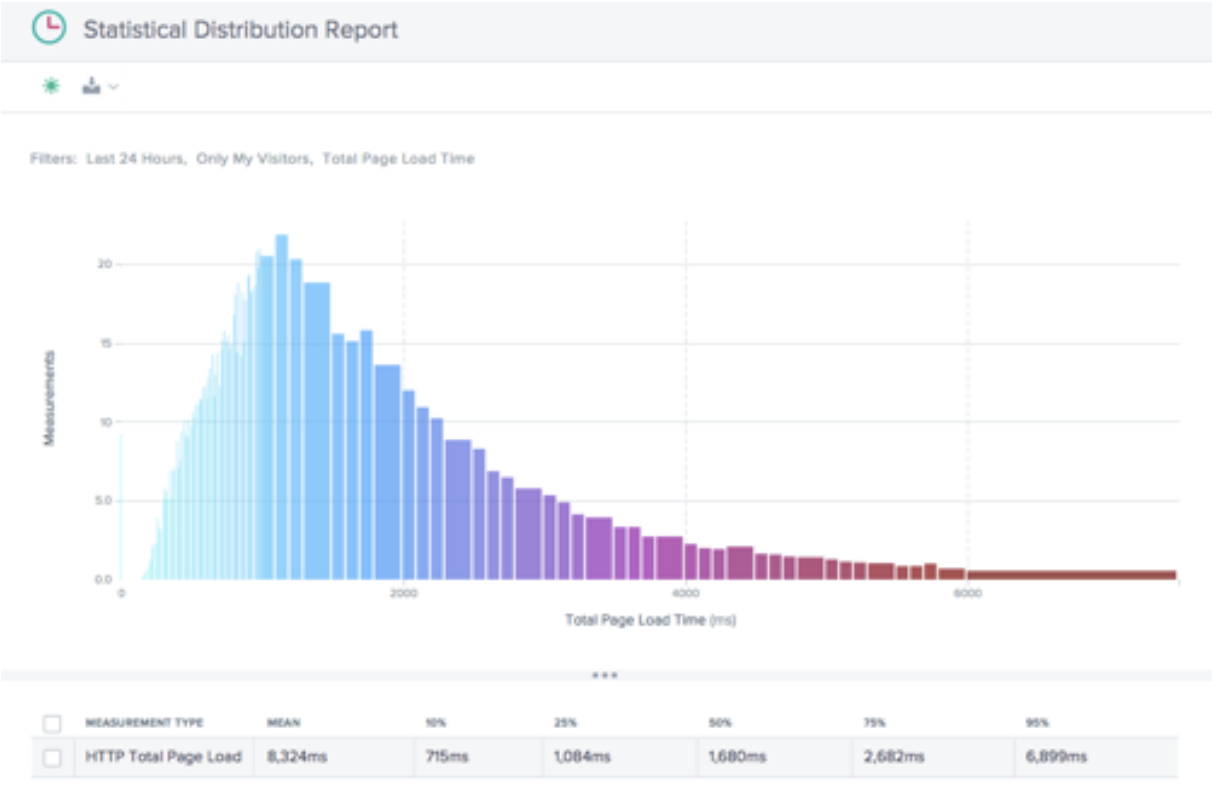
Performance Report



This report shows the Navigation Timing KPI performance over time broken down by measurement type.

By default, Start Render and Total Page Load Time are selected. Other measurement types can be added in as needed.

Statistical Distribution Report



This report shows the statistical distribution of Navigation Timing and page load time values.

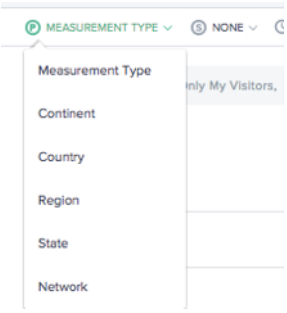
The report gives insight into how many measurements (Page Views) were collected per page load time value.

Using Navigation Timing Reports

To refine and customize the report views for specific reporting needs, use the following functionality in the Navigation Timing reports.

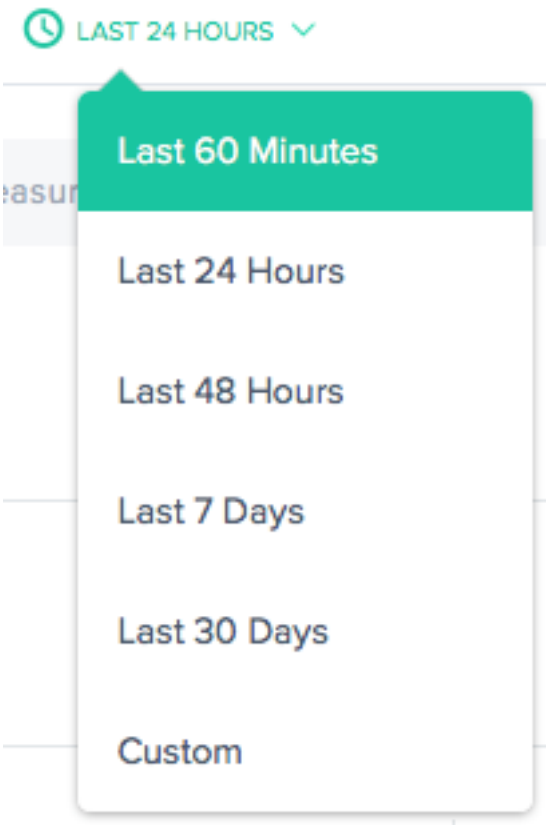
In addition to the standard features of the reports such as report sharing, background toggle, data export, and more the following features are available:

Primary and Secondary Dimensions



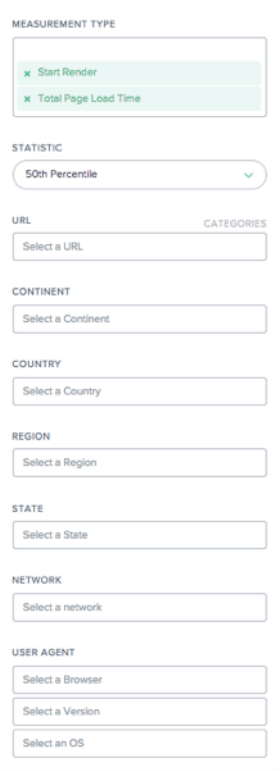
The primary dimension of the chart is selected through a selection list above the chart. Use this as a powerful pivot on the report to express the data in terms of Measurement Type (default), Continent, Country, Region, State, or Network (ASN).A secondary dimension can be chosen as well to further refine the reporting.

Filter: Report Time Range



The reports can be generated with a time range of Last 60 Minutes, Last 24 Hours, Last 48 Hours, Last 7 Days, Last 30 Days, or a custom range. The default view is the Last 24 Hours.

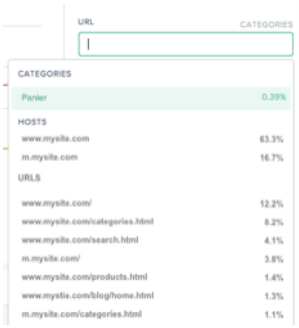
Filters: Powerful Drill down Capabilities



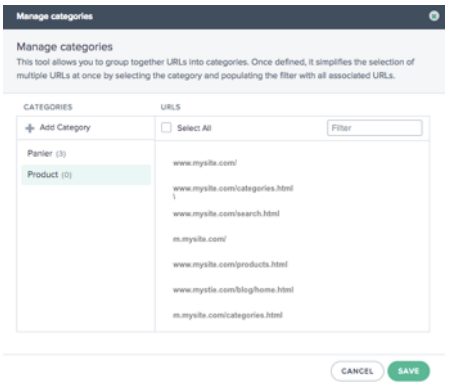
The reports vary slightly in terms of which filters are appropriate based on the data. The following are available in the Navigation Timing reports:

- **Measurement Type** – Select one or more measurement types to view. Start Render and Total Page Load Time are selected by default.
- **Statistic** – Select one statistical measure to view the data.
- **URL** – Select one or more URLs to view. Also, you can select a host name or a category of URLs (see below).
- **Continent** – Select one or more continents to include
- **Country** – Select one or more countries to include
- **Region** – Select one or more geographic regions (where applicable) to include
- **State** – Select one or more geographic states (where applicable) to include
- **Network** – Select one or more networks (ASN) to include
- **User Agent** – Select one or more browsers, browser version and/or OS to further refine the reporting data.

URL Categories



Navigation Timing reports can be filtered by URLs, Hosts, or Categories. Quickly find one or more items of interest by typing into the **URL search** box.



To create a Category, click **CATEGORIES** on the right side of the **URL** box. The **Manage Categories** dialogue box appears.

Select **Add Category** to create a category and name it as desired. Then select the URLs of interest for the new Category. To find URLs, simply start typing in the search box and the URL list will be filtered to the search text.

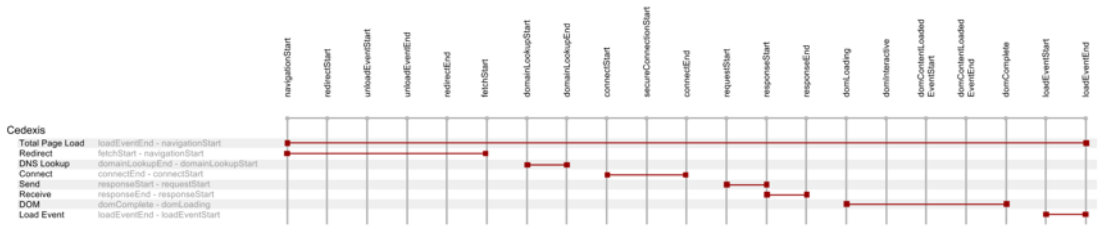
When all URLs have been selected for the Category, click the **Save** button to complete the **Category** definition.

Navigation Timing and Page Load Time Data

The Radar tag is able to collect detailed information on download performance for pages that implement the tag.Performance information from the NavTiming API is collected from those browsers that support the API (Chrome 6.5+, Firefox 8+, IE9+).

Citrix displays this information within the customer's portal where it allows them to see the performance actual end-users are experiencing when interacting with your webpages.

Below is a diagram and description of each of the page load metrics Radar provides via Navigation Timing:



Measurement	Description	Nav Timing Calculation
Total Page Load	The full download of the webpage and its corresponding components.	`loadEventEnd - navigationStart`
Redirect	The initial time used in redirecting to the page.	`fetchStart - navigationStart`
DNS Lookup	The time required for DNS resolution to complete of the base page URI.	`domainLookupEnd - domainLookupStart`
Connect	The time to make a TCP connection, inclusive of SSL if it's used.	`connectEnd - connectStart`
Send	The HTTP Request and Response time of the initial base page, excluding any message body.A good indicator of back end server latency.	`responseStart - requestStart`
Receive	The time taken to receive the Body HTML of the base document.	`responseEnd - responseStart`
DOM	The Time to download all media, objects that are called from base HTML and load them into the browser.	`domComplete - domLoading`
Load Event	The time for executing any JavaScript and rendering the page within the browser.	`loadEventEnd - loadEventStart`

Measurement	Description	Nav Timing Calculation
Start Render	The Start Render time is the first point in time that something was made available to the screen.	More timing added by Chrome/IE as an extension to the NavTiming API.

Video Playback Data

December 17, 2020

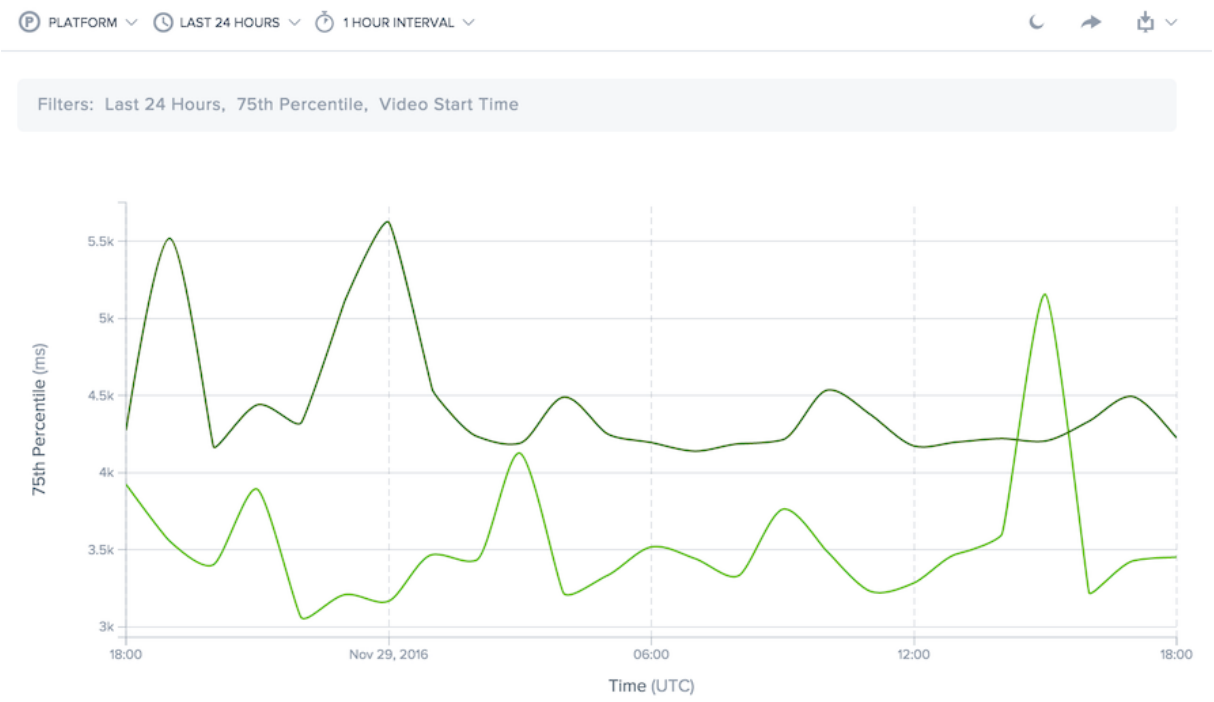
Cloud Platform Visualization collects the most pertinent video network delivery performance and quality of experience data for reporting. The video quality of experience is directly driven by the quality of the video chunk delivery. Openmix optimizes based on Radar network delivery metrics to provide the best possible viewing experience for users. Following a brief description of the reports are details on how to pivot, filter, and customize the reports.

Video Playback Reports

The **Video Playback Data** menu includes the following reports:

1. **Performance Report** – Video experience and delivery data over time.
2. **Statistical Distribution Report** – Variation in the video viewing experience over time.
3. **Histogram Comparison Report** - Compare video chunk delivery data with quality of experience KPIs.

Performance Report

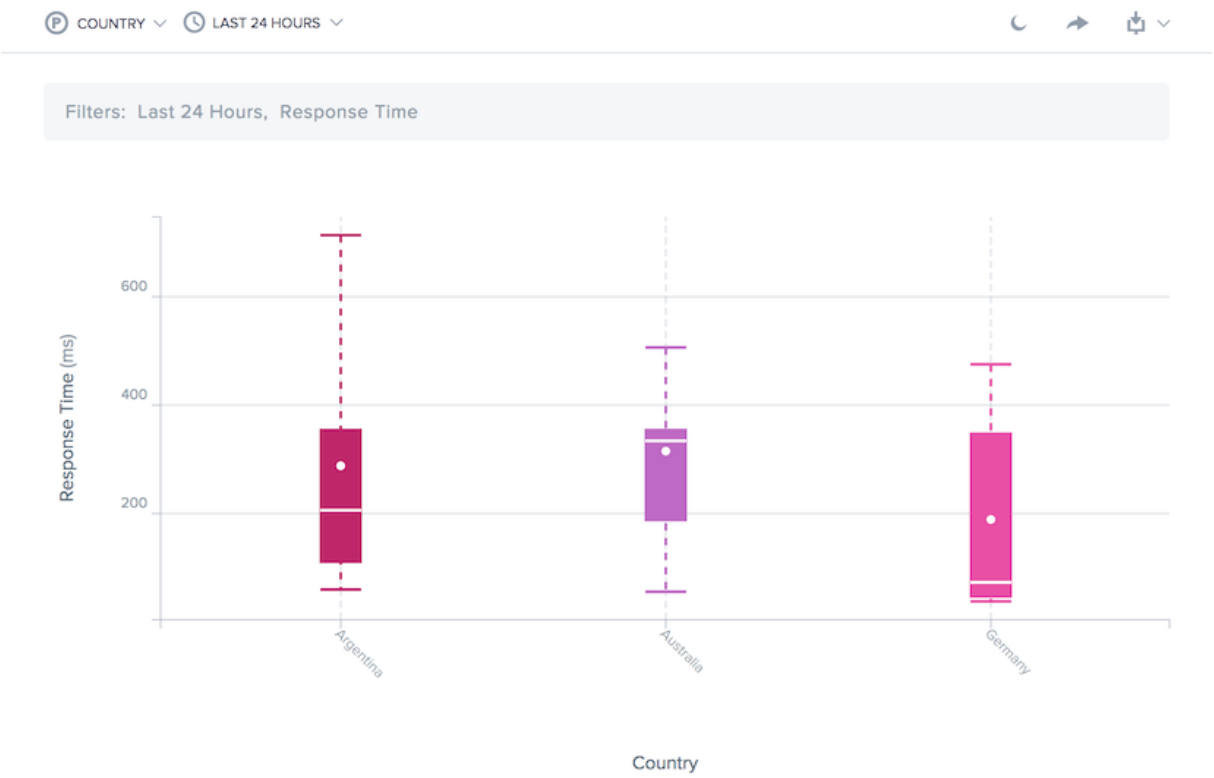


This report shows the video viewing experience over time. It allows you visualize delivery trends over time, see how much video is being watched, and the aggregate quality of the viewing experience.

The data can be viewed with dimensions that allow for comparison of multiple values. For examples, the data can be viewed by domain to compare the performance of delivery across multiple video domains.

The time period for the report can be customized from the most recent 60 minutes up to 30 days within the last 13 months.
Data can be filtered by the platform being used to serve the content, the host name, and path of the content or video chunks, geographic location, network, or viewer user agent.

Statistical Distribution Report



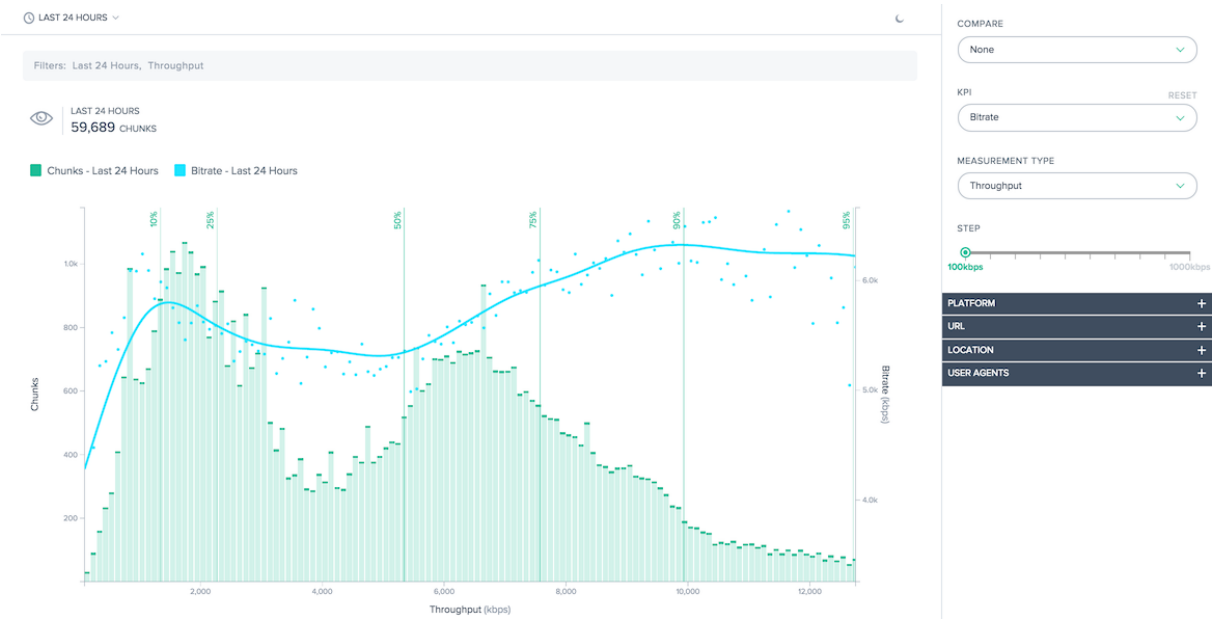
This report shows the variation in the video viewing experience over time. It allows you visualize how consistently video is delivered and better understand the viewing experiences across the whole population of users. The report calculates user performance at the 10th, 25th, 50th, 75th, and 95th percentiles and the mean.

Like the Performance Report, the data can be viewed with dimensions that allow for comparison of multiple values. For example, the data can be viewed by platform (service provider or server) to compare the consistency of delivery for multiple platforms.

The time period for the report can be customized from the most recent 60 minutes up to 30 days within the last 13 months.

Data can be filtered by the platform being used to serve the content, the host name, and path of the content or video chunks, geographic location, network, or viewer user agent.

Histogram Comparison Report



This report surfaces the relationships between video chunk delivery data and the quality of experience KPIs.

There are two main features in this report:

- The histogram shows how often video chunks were delivered with a specified level of quality, either Response Time or Throughput.
- Individual KPIs can be overlaid on the histogram. The lines chart the KPI produced when a chunk was delivered with the specified level of quality.

For example, the histogram would show the chunk throughput measured by Radar. The KPIs will likely show that bitrate is higher and rebuffering is lower when the measured throughput is higher. Together, these features help quantify the relationship between delivery quality and the quality of experience produced for the viewer.

If the default report generation is not sufficient, the histogram bucket size can customized and specific sections of the distribution can be selected for display.

In addition to relating histograms to KPIs, data can be compared directly. Multiple KPIs can be selected for viewing and previous time periods can be compared to show changes in performance over time.

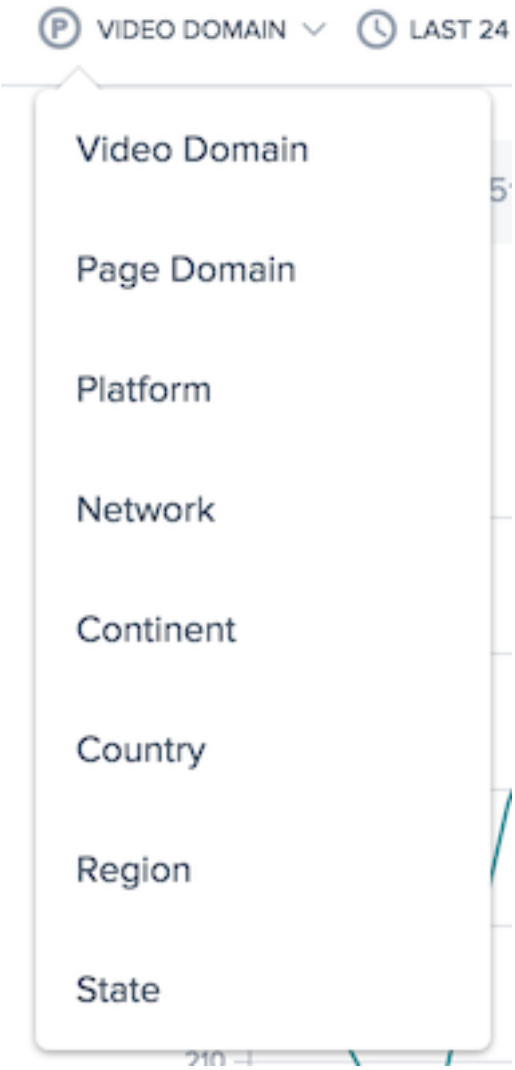
Data can be filtered by the platform being used to serve the content, the host name, and path of the content or video chunks, geographic location, network, or viewer user agent.

Using Video Playback Reports

To refine and customize the report views for specific reporting needs, use the following functionality in the Performance and Statistical Distribution Video Playback reports.

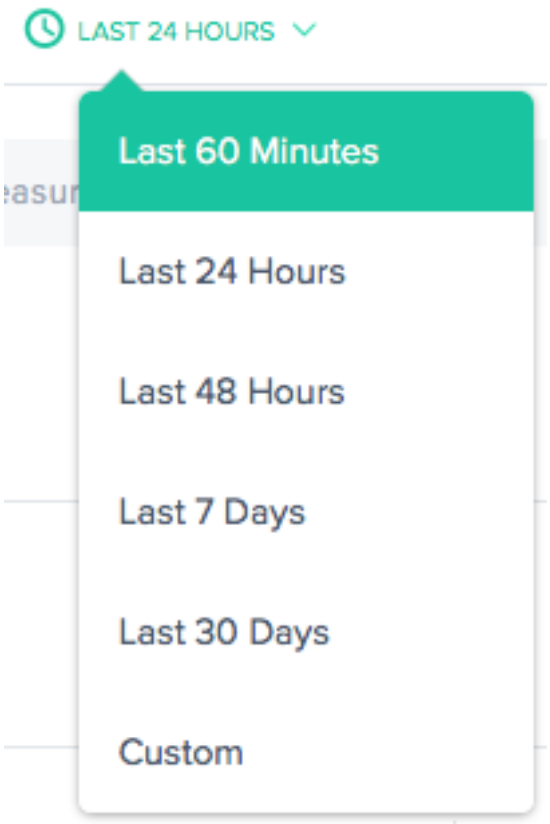
In addition to the standard features of the reports such as report sharing, background toggle, data export, and more the following features are available:

Primary Dimension



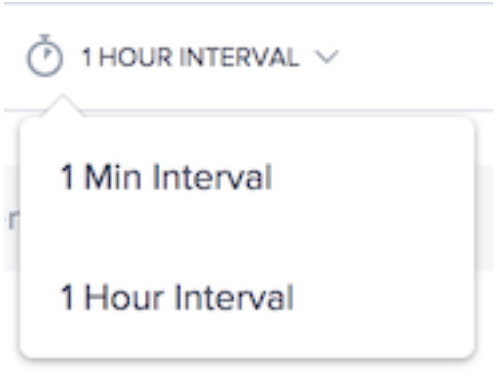
The primary dimension of the chart is selected through a selection list above the chart. Use this as a powerful pivot on the report to express the data in terms of Video Domain, Page Domain, Platform, Network (ASN), Continent, Country, Region, or State.

Filter: Report Time Range



The reports can be generated with a time range of Last 60 Minutes, Last 24 Hours, Last 48 Hours, Last 7 Days, Last 30 Days, or a custom range. The default view is the Last 24 Hours.

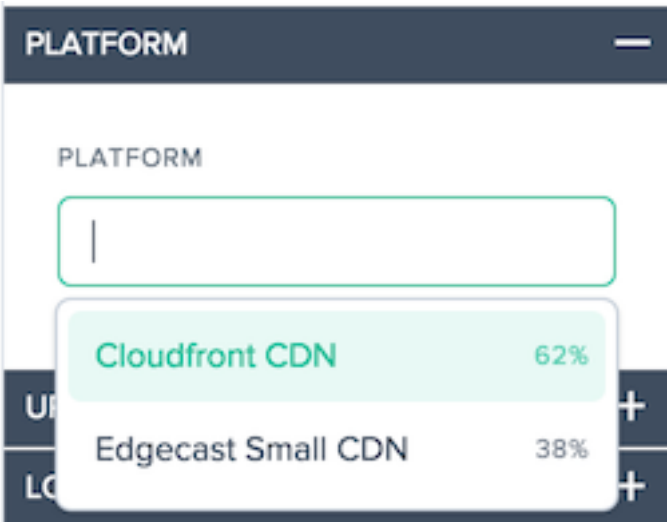
Report Interval



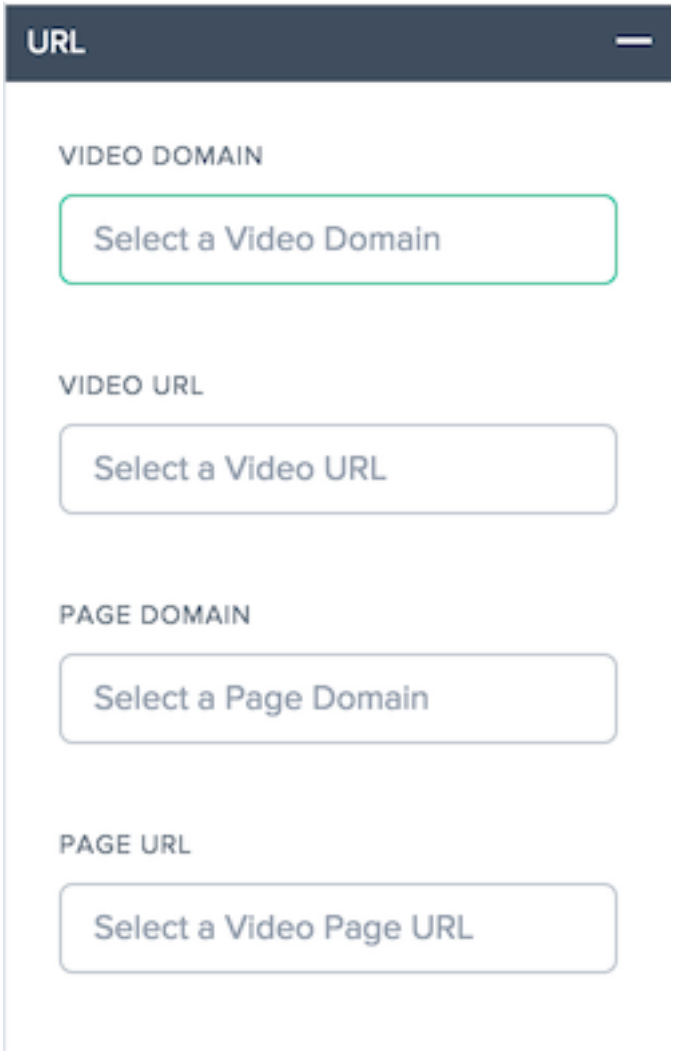
The primary dimension of the chart is selected through a selection list above the chart. This allows for granular reporting of performance data.

Filters: Powerful Drill down Capabilities

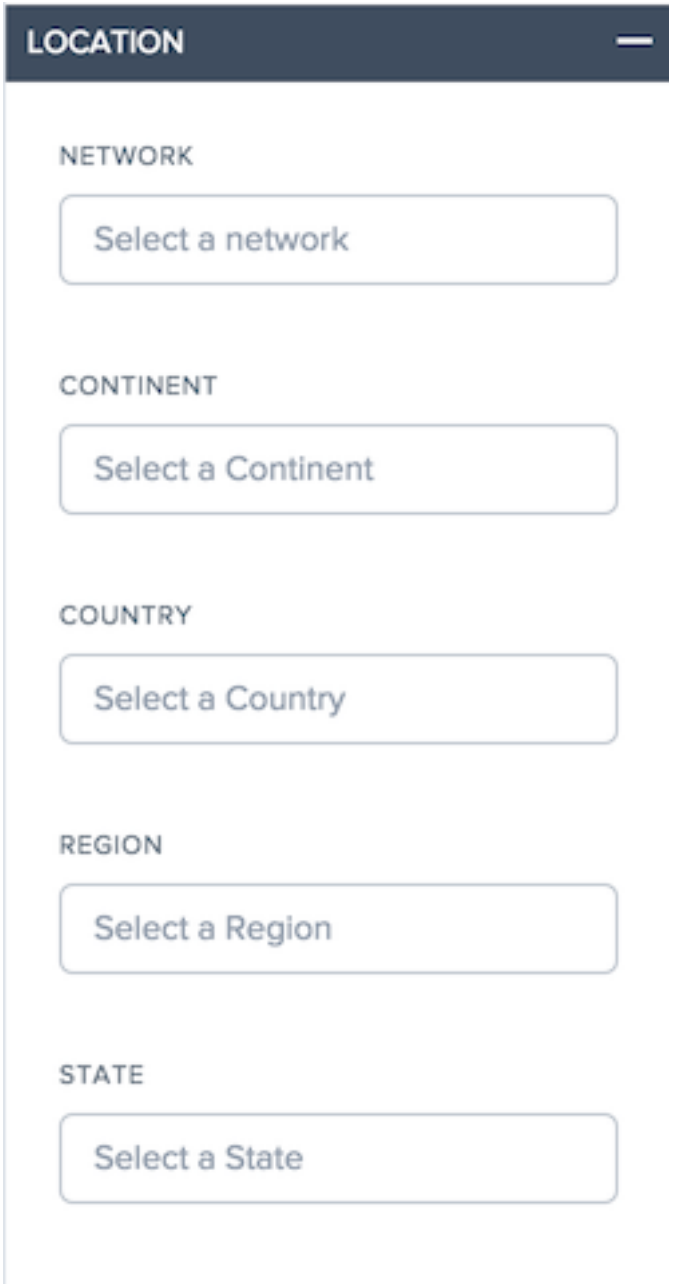
The reports vary slightly in terms of which filters are appropriate based on the data. The following are available in the Video Playback reports:



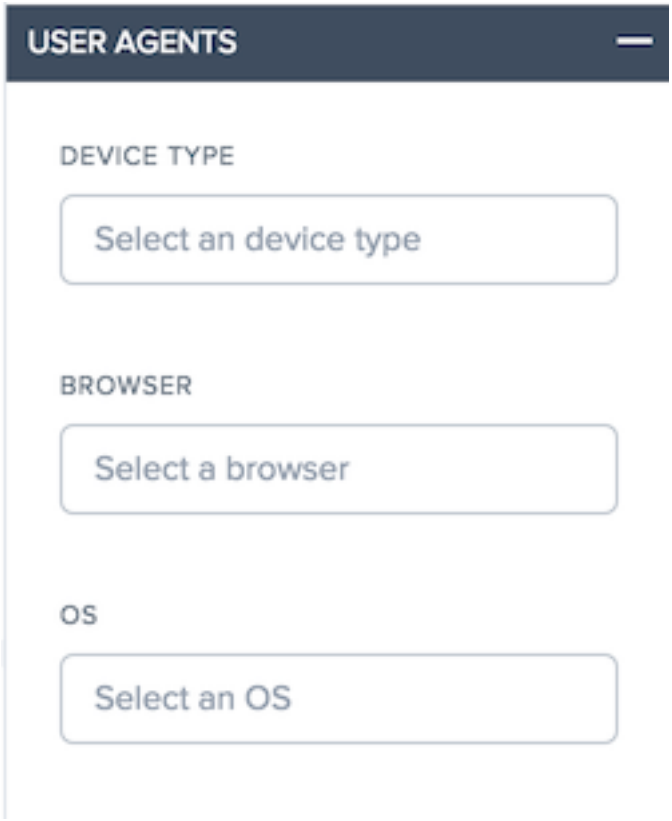
- **Platform** - Select one or Platforms to filter, by default all platforms are included in the report.



- **Video Domain** - Select one or more host names on which videos are hosted, by default all host names are included in the report.
- **Video URL** - Select one or more paths for the videos, by default all paths are included in the report.
- **Page Domain** - Select one or more host names on which pages are hosted, by default all host names are included in the report.
- **Page URL** - Select one or more paths for the pages, by default all paths are included in the report.



- **Network** – Select one or more networks (ASN) to include
- **Continent** – Select one or more continents to include
- **Country** – Select one or more countries to include
- **Region** – Select one or more geographic regions (where applicable) to include
- **State** – Select one or more geographic states (where applicable) to include

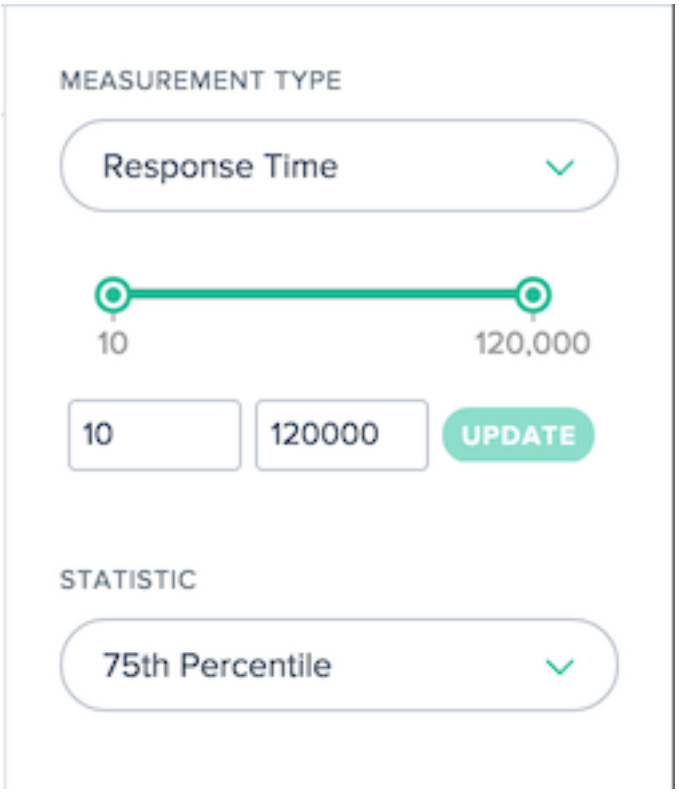


- **User Agent** – Select one or more device types, browsers, and/or OS types to further refine the reporting data.

Using Video Playback Performance Report

To refine and customize the performance report for specific reporting needs, use the following functionality in the Performance report.

Filters: Powerful Drill down Capabilities



The reports vary slightly in terms of which filters are appropriate based on the data. The following are available in the Video Playback reports:

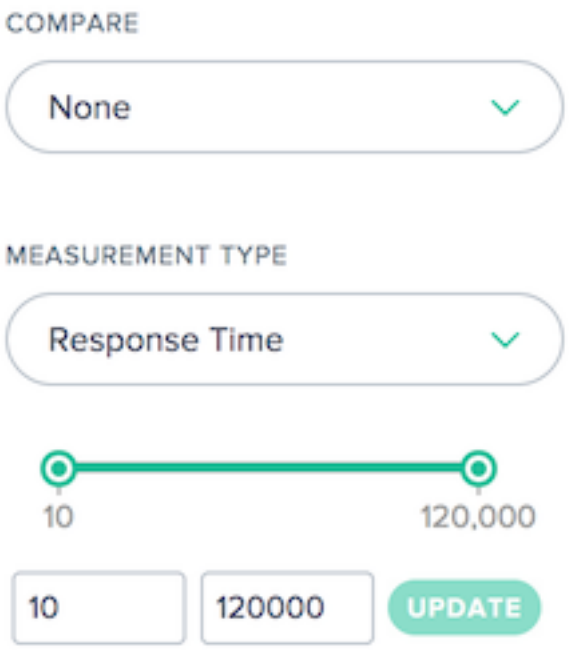
- **Measurement Type** – Select the measurement type to view, Response Time is initially selected.
- **Count Slider** - Filter the data by the minimum and maximum measurement count required to be included in the report.
- **Statistic** – Select the statistical measure to view.

In addition to these report-specific filters, the standard Video Playback filters are available to customize the results.

Using Video Playback Statistical Distribution Report

To refine and customize the report for specific reporting needs, apply the following functionality in the Statistical Distribution report.

Filters: Powerful Drill down Capabilities



The reports vary slightly in terms of which filters are appropriate based on the data. The following are available in the Video Playback reports:

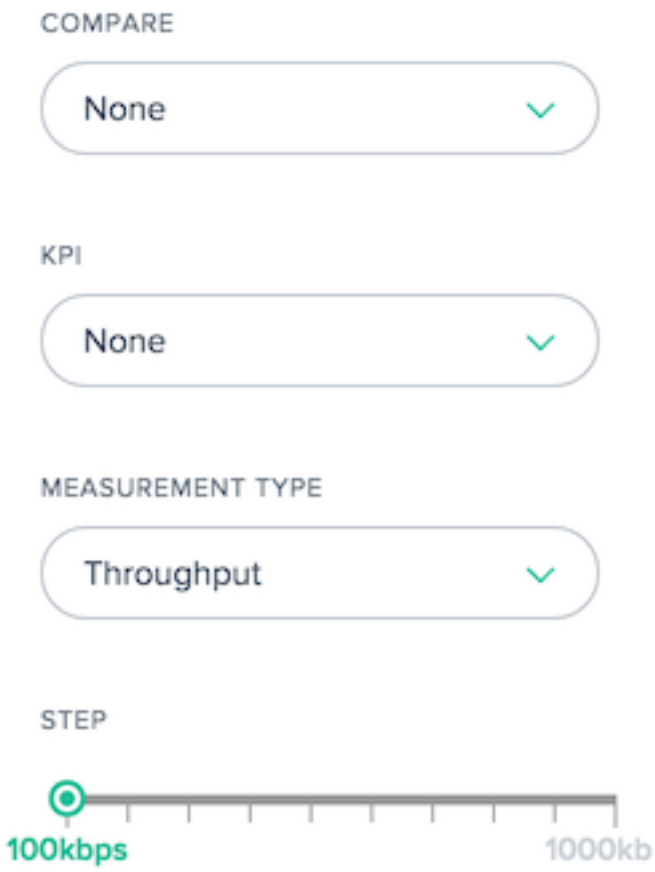
- **Compare** – Select the value used to create a comparison in the report. Based on the selection made, the specific values used to compare need be selected. The resulting distributions will be displayed side-by-side so they can be easily compared.
- **Measurement Type** – Select the measurement type to view, Response Time is initially selected.
- **Count Slider** - Filter the data by the minimum and maximum measurement count required to be included in the report.

In addition to these report-specific filters, the standard Video Playback filters are available to customize the results.

Using Video Playback Histogram Comparison Report

To refine and customize the report for specific reporting needs, apply the following functionality in the Histogram Comparison report.

Filters: Powerful Drill down Capabilities



The reports vary slightly in terms of which filters are appropriate based on the data. The following are available in the Histogram Comparison reports:

- **Compare** – Select the value used to create a comparison in the report. Based on the selection made, the specific values used to compare need be selected. The resulting histograms and KPIs will overlaid on top of each other so they can be easily compared.
- **KPI** – Select the KPI that is graphed against the histogram measurement type.
- **Measurement Type** – Select the measurement type used to populate the histogram.
- **Step Slider** - Set the size of the buckets used to generate the histogram.

In addition to these report-specific filters, the standard Video Playback filters are available to customize the results.

Video Playback Data

Data is collected using the properties and events of the HTML5 video element for quality of experience data and the Resource Timing API for the video chunk data, from the browsers that support the APIs.

Video data is displayed in the portal, where reports can be generated with information on end-user quality of experience and network delivery performance.

Below is a diagram and description of each of the video metrics that are collected:

Measurement	Description
Per-chunk Response Time	The time it takes for the chunks to start delivery based on the resource timing measurements (`responseStart - requestStart`)
Per-chunk Throughput	The speed at which video chunks were downloaded based on the resource timing measurements.(kbps)
Delivered Bitrate	The per-second bitrate of the video based on the size of the chunks delivered.(kb)
Rebuffering Ratio	The percentage of time spent rebuffering during the playback.(%)
Video Start Failures	The HTTP Request and Response time of the initial base page, excluding any message body.A good indicator of back end server latency.
Video Start Time	The amount of time it took to start video play after the play attempt is made.(ms)

Resource Timing Data

December 17, 2020

Overview

Resource Timing data offers a powerful view into the performance of individual object-level resources of your website.

Resource Timing helps customers look at the network performance of page level objects, based on the data that we provide on connection time, download time, and different response times. Examples of page level objects are, images, JavaScript files, API calls, and so on. It gives customers better visibility into the page level performances. The end result is that customers can better manage their delivery and ensure an overall better quality of user experience.

The following sections walk you though the configuration, data description, and reporting of resource timing data.

Resource Timing Configuration

The user interface in the portal allows you to directly input settings for Resource Timing configuration as an alternative to JSON coding.

Note: Even though configuration through JSON coding is still available, it is highly recommended that you use the UI for configuration.

Navigation

From the left navigation pane, choose Impact -> Resource Timing Data -> Resource Timing Configuration.

First time configuration

- Select **Start Now** on the opening page to get started.
- A **Default Configuration Setting** dialog opens for you to include or exclude resources and enter a sample rate.

Default Configuration Settings

The default configuration settings are the minimum settings required to get started. There are three main default configuration settings:

- Resources to include and exclude
- Sample Rate
- Default Provider Detection

Resources to include or exclude

This feature allows you to include or exclude specific resources to collect timing data from. If left blank, all resources are included by default (that is, nothing is excluded).

You can enter resources such as, a file name, file name extension, folder name, file path, or even a string. Anything contained within the string will be picked up as a resource.

Press **Enter** or the **Return** key each time you input a resource name to submit it. If you enter specific resources in the **Include** field, only those resources are included and all other resources are excluded. To exclude specific resources, enter them in the **Exclude** field, and everything else, will be included. You can even write a custom regex logic to customize the inclusion or exclusion process.

Sample Rate

The **Sample Rate** allows you to enter a small sample of visitors that you want to collect IRT data from. Enter a value between 0 and 100 (taken as a percentage). Ideally you must enter the lowest percentage for sample rate - a value that is enough to collect the required number of resource timing measurements.

Note: Resource timing data collection places a heavy load on the system. This feature is for customers to sample data, and is not meant to collect data for every Radar session.

Caution: For customers with a high volume of data, start with a 1% sample rate. Increase it slowly until a statistically useful rate is reached. A high sample rate can potentially cause a server overload, slow down, or even a crash.

Steps for first time sample rate setting

1. Start with a 1% sample rate. Wait 24-48 hours until you receive a few measurements.
2. Check the **IRT graph** to see if it looks smooth across multiple assets.
3. If yes, then leave the sample rate at this value, unless the customer has high web traffic.
4. Alternatively, if the graph looks choppy due to low volume of data, turn it up slowly.
5. Repeat all the checks and keep turning the rate up slowly (ideally every 24-48 hours) until you receive enough data (at about 10%).
6. For customers with low web traffic you can go up higher than 10%. But for every small increase, ensure you perform all the checks mentioned.

Select **Next** to go to the **Default Provider Detection Setting** dialog.

Default Provider Detection

Provider detection allows you to identify the provider or platform from where the resource is being served. Enter a host name that is configured to detect the provider that serves the resource. You can enter multiple host names and configure provider detection for each of them individually. See the Provider Detection section for information on how to configure Provider Detection.

Select **Complete** to complete the first-time configuration.

Sites

The **resource timing data** is set up around three main areas:

1. **Sites**
2. **Configuration**
3. **Provider Detection**

- From the left navigation pane, go to **Impact -> Resource Timing Data -> Resource Timing**.

- The **Sites** page under **Resource Timing Data** opens.

Enter the host name of the site you want to collect resource timing data from. Under **Sites**, you find the list of host names that are in the system already. If you don't find the required site (host name), you can enter it by clicking the **Add** button. The **Add Site** dialogue allows you to add a new site to configure resource timing data on.

Configuration

Navigate to **Impact > Resource Timing Data > Resource Timing Configuration** from the side navigation menu of the Portal. The **Sites** page opens under **Resource Timing Data**.

From the top navigation bar choose **Configuration**.

You can add a new configuration by clicking the add button on the top right corner of the page.

Note: You may also see a list of configurations including the default configuration on the page. Instead of adding a new configuration, you can either select a default configuration, or edit an existing one from the list.

Add Configuration

To add a new configuration, click the **Add** button on the top right corner of the page.

The **Add Resource Time Configuration** dialog opens. This allows you to enter a new configuration **Name**, add **Resources to Include or Exclude**, and add the **Sample Rate**.

Edit Configuration

To edit an existing configuration select the **Edit Configuration** button beside the configuration name.

Provider Detection

Provider detection determines which platform handles a request for a domain when that domain is load-balanced behind Openmix. It is recommended that all customers who have resource timing data enabled, configure provider detection services.

- To configure provider detection, navigate to **Impact > Resource Timing Data > Resource Timing Configuration** from the left navigation pane.
- The **Sites** page under **Resource Timing Data** opens. From the top navigation bar choose **Provider Detection**.

Click the add button in the top right corner of the page.

In the **Add Provider Detection Configuration** dialog, enter the following.

Configuration Name

Enter a name for the configuration. The name cannot contain any spaces, or special characters, and must be unique.

Host name

Enter the host name for which you want to configure provider detection. You can enter multiple host names and specify detection methods for each of them individually.

Detection Method

The detection method involves specifying the type of test object (whether standard or custom) and path (to the test object) for each host name that you've entered.

Standard Test Objects

In the case of standard test objects, the path can be specified as, **/provider-detection/platform.html** and **/provider-detection/platform.png**. For this setup, **/provider-detection/** would be your directory path.

Note: It is not mandatory to enter the path described above. However, for any path you enter, ensure that the **platform.html** and **platform.png** files are found in the directory path.

Custom Test Objects

In the case of custom test objects, you need to ensure that the test objects are found in the exact path you enter. For example, for host name `foo.com` and path `static/bar.css`, the URL `http://foo.com/static/bar.css` must be valid.

Headers

Platform Header

If you select **Platform Header**, ensure that the `X-CDN-Forward: <CDN name>` is sent on the test objects. If the `X-CDN-Forward: <CDN name>` is not found in the response headers, then the client moves on to the next test, which can be specified using **Custom**.

Custom

If you select **Custom**, ensure that the regular expression you enter, matches exactly with one of the CDN's response headers.

If you add multiple response headers, they are each tested against the regular expressions in the same order as entered in the portal.

Click **Create** to complete the process. You now see the newly created configuration in the list under **Provider Detection**. Click the edit or delete icons if you want to modify the configuration, or delete it.

Your configuration is now complete. To configure Provider Detection alternatively through JSON coding, contact your account representative.

Resource Timing Measurement Descriptions

The following table shows the Resource Timing measurements that are collected.

Measurement	Description	Resource Timing Calculation
DNS Lookup Time	The time required for the DNS resolution of the resource. Known as the DNS phase.	`domainLookupEnd - domainLookupStart`
TCP Connection Time	The time it takes a browser to establish connection with a server. Known as the TCP phase.	`connectEnd - connectStart`
Waiting Time to First Byte (TTFB)	TTFB is the amount of time a browser waits before the start of receiving the resource.	`responseStart - startTime`
Round Trip Time (RTT)	The time from the start of the request to the start of the response. Known as the Request phase.	`responseStart - requestStart`
Wait Time	The difference between the start of the response and the end of the response. Known as the Response phase. The response is usually from a server, cache, or local resource.	`responseEnd - responseStart`
Duration	The total time from the start of the process to the complete reception of the resource.	`responseEnd - startTime`

Learn more at https://www.w3.org/TR/resource-timing-1/#process

Resource Timing Reports

The **Resource Timing** menu includes the following reports:

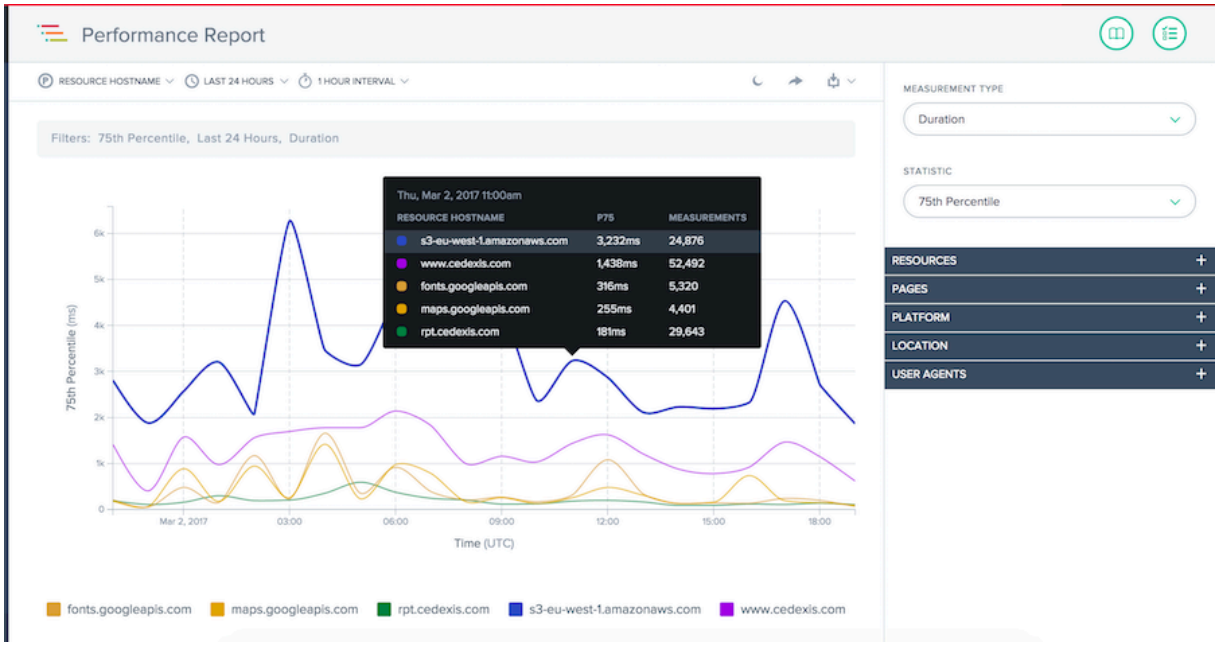
1. **Performance Report** – Resource timing measurement data over time.
2. **Statistical Distribution Report** – A view of Resource Timing data through a statistical distribution reporting view.

Performance Report

The report gives insight into performance data of resource timing over time per value selected.

Default Reporting View:

1. Dimension: Resource Host name
2. Measurement: Duration
3. Time Range: Last 24 hours

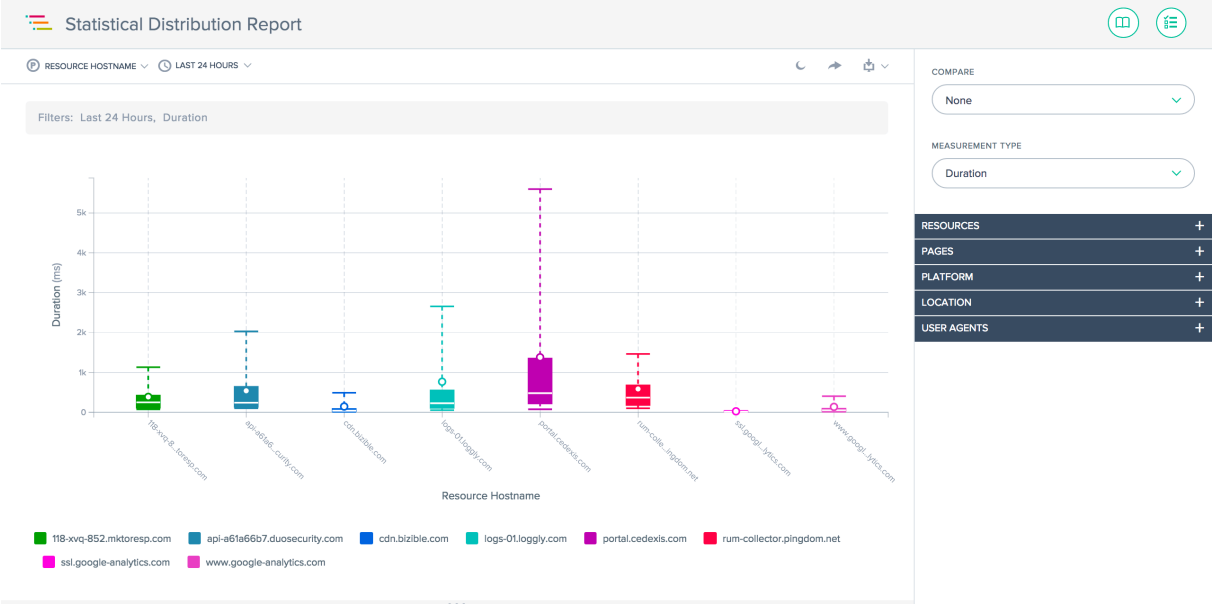


Statistical Distribution Report

This report shows the statistical distribution of Resource Timing. The report gives insight into how many measurements were collected per resource value. You can filter based on Resource, Page, Platform, Location and User Agent, switch between measurement types, and run comparisons between specific page, location, and user agent details.

Default Reporting View:

1. Dimension: Resource Host name
2. Measurement: Duration
3. Time Range: Last 24 hours

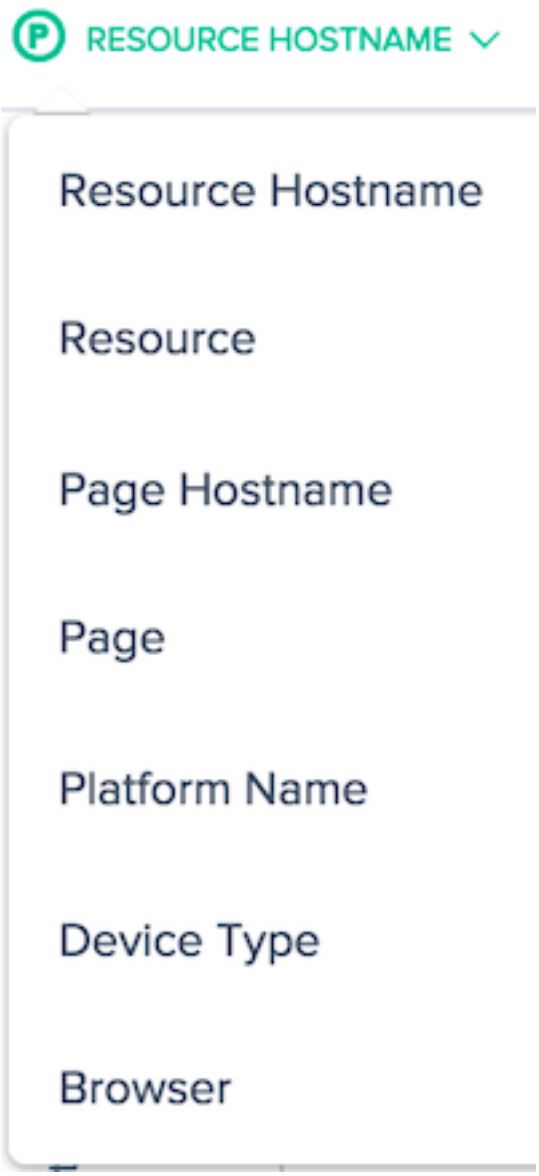


The whisker chart

Using the Reports

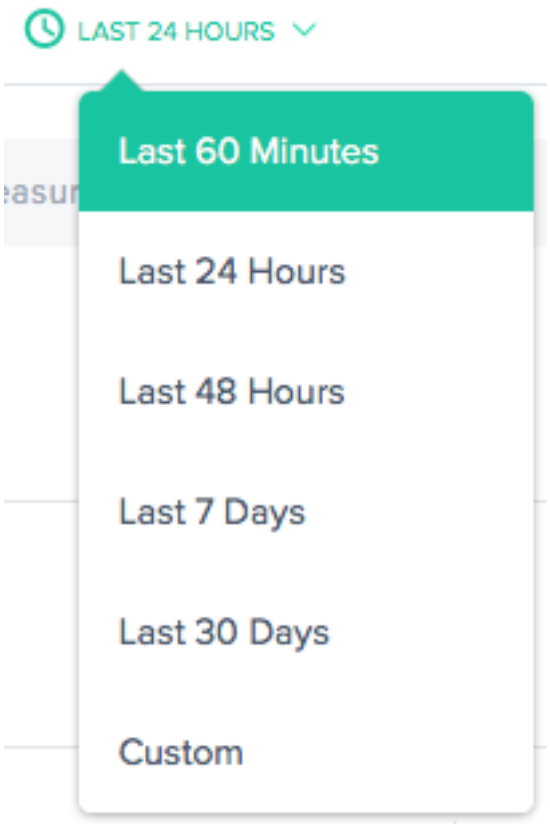
To refine and customize the report views for specific reporting needs, use the following functionality in the Performance and Statistical Distribution reports. In addition to the standard features of the reports such as report sharing, background toggle, data export, and more the following features are available:

Primary Dimension



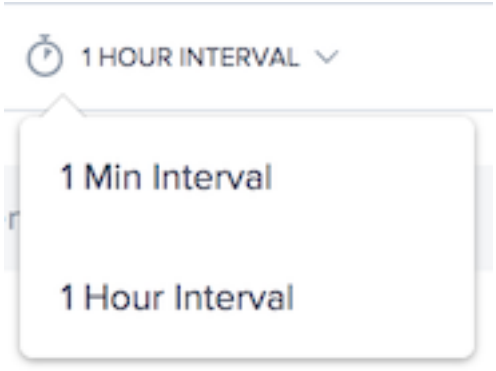
The primary dimension of the chart is selected through a menu above the chart. You can use it as powerful pivot on the report to express data in terms of Resource Host name, Page Host name, Page, and Platform Name.

Filter: Report Time Range



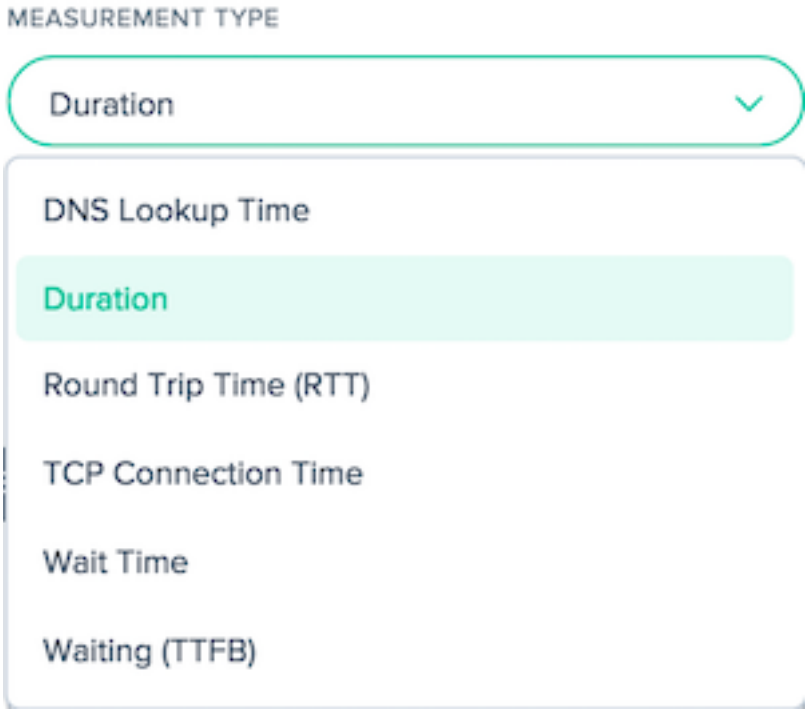
The reports can be generated with a time range of Last 60 Minutes, Last 24 Hours, Last 48 Hours, Last 7 Days, Last 30 Days, or a custom range. The default view is the Last 24 Hours.

Report Interval



Select the timing interval in which you want to view the trending graph. Depending on the date range you are viewing, you can view the graph in one minute, one hour, or one day intervals.

Measurement Types



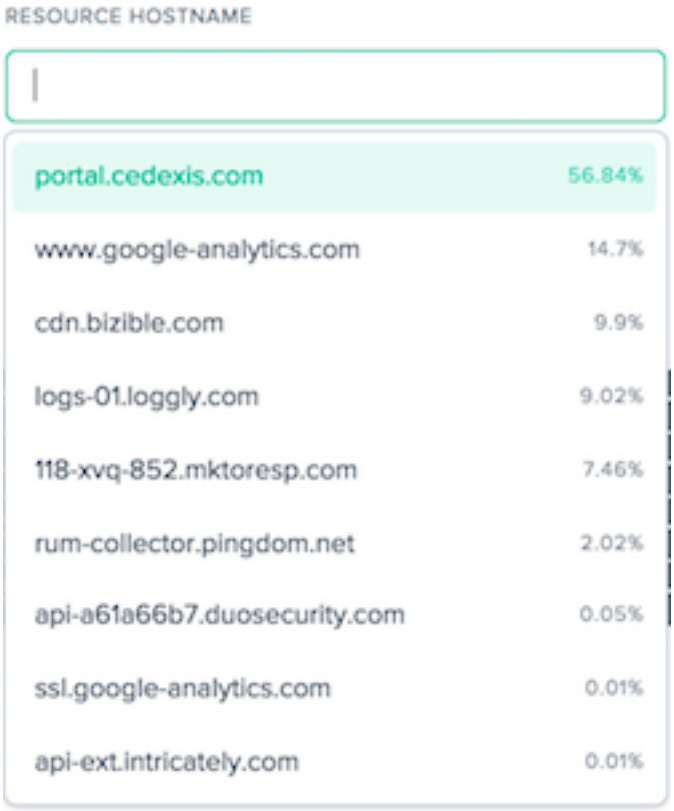
Select the measurement type which you want to view the resource timing against. Choose from Duration, DNS Lookup Time, Round Trip Time (RTT), TCP Connection Time, Wait Time and Waiting (TTFB).

Select one statistical measure to view the data.

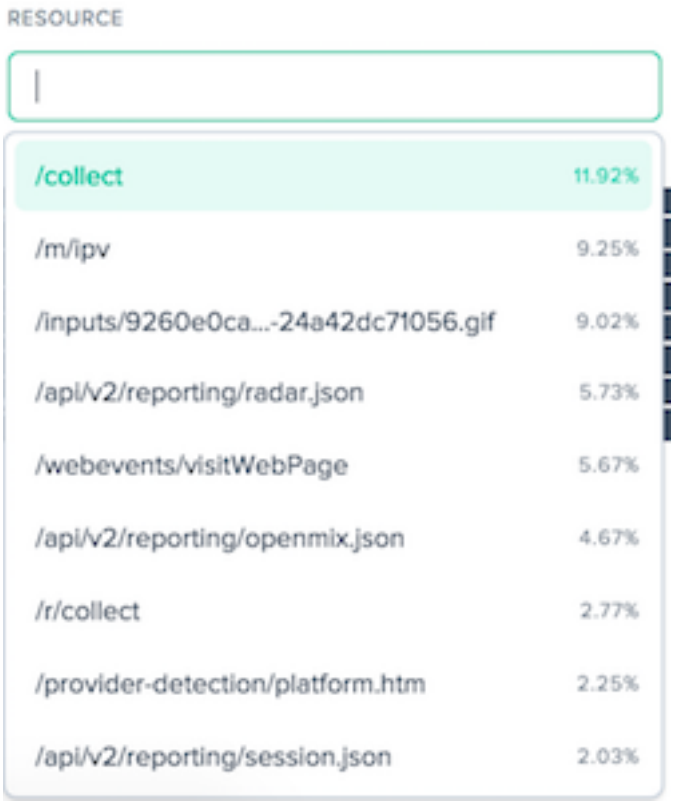
Filters: Powerful Drill-down Capabilities

The reports vary slightly in terms of which filters are appropriate based on the data. The following filter options are available in the reports:

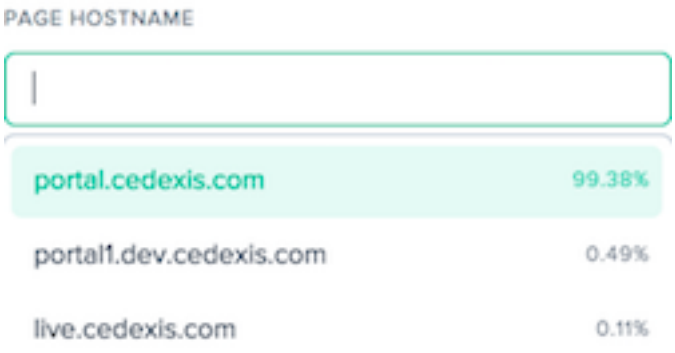
Resource Host name:



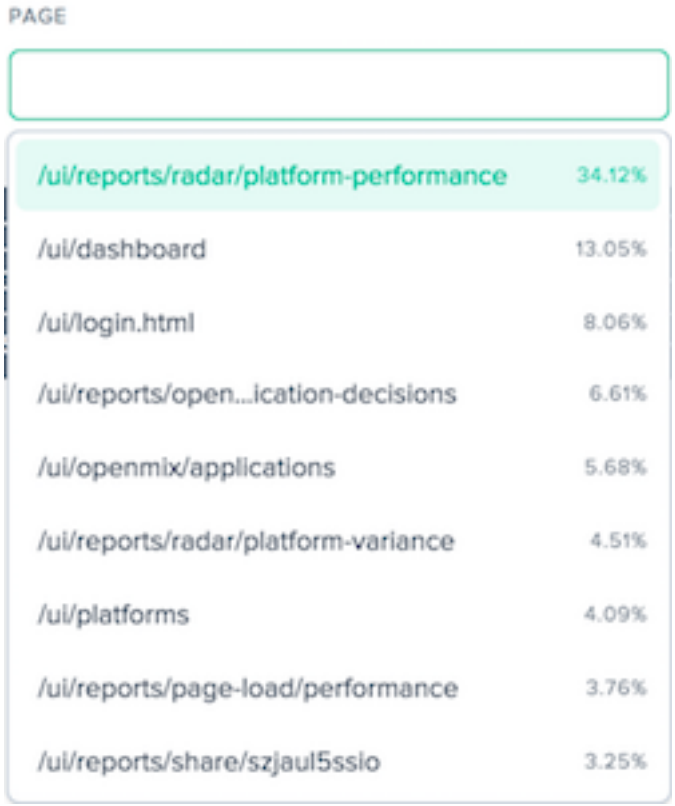
Resources:



Page Host name:

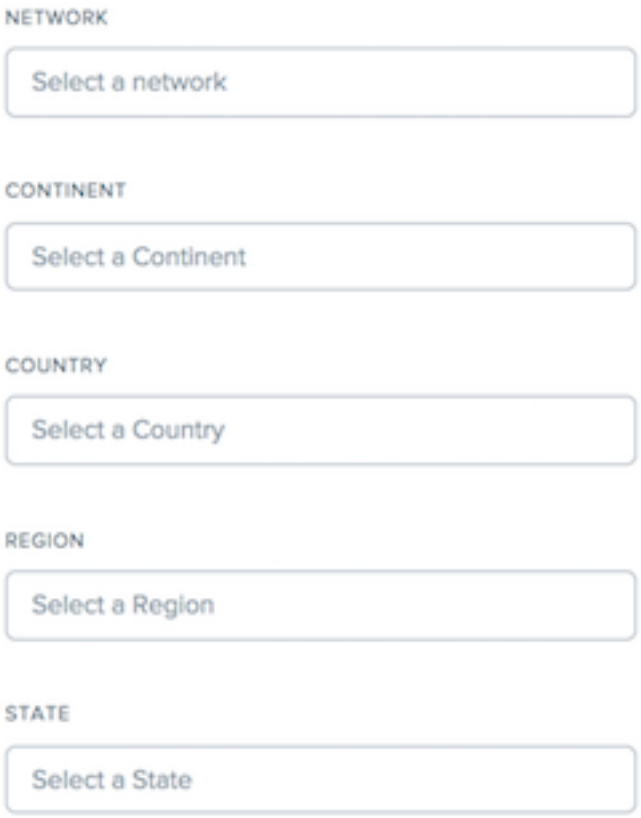


Page:

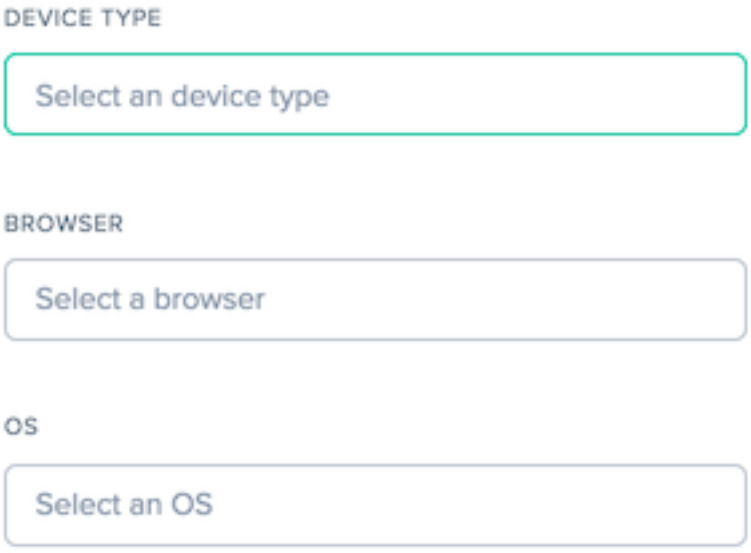


Platform Name:

Location: Network, Continent, Country, Region, and State:



User Agents: Device Type, Browser, and IOS:

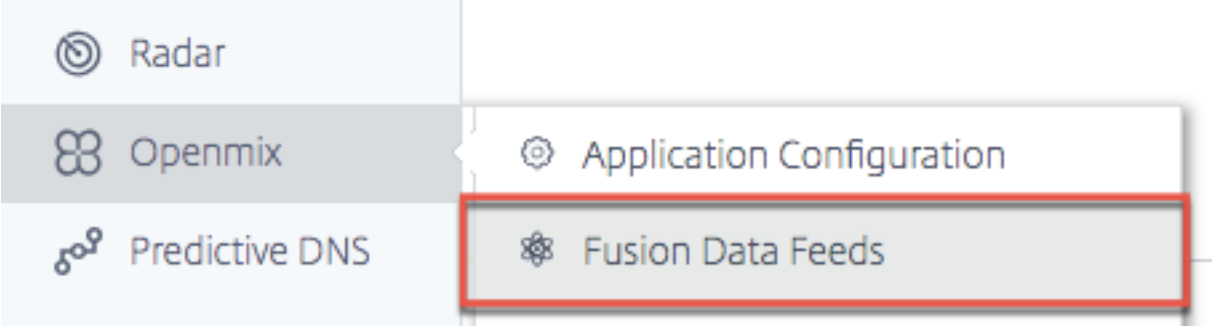


Fusion Integrations

April 13, 2020

In addition in Radar and Sonar data, Openmix can use third party data in its decision criteria. For example, you can integrate an existing synthetic monitoring service you already use. Or you can make cost-based decisions using up-in-date usage data from your CDN provider.

Fusion menu



Fusion Data Feeds can be accessed from the navigation menu, under **Openmix**.

For example, some common Fusion data feeds that work with Openmix applications:

1. **Server Availability** – Ingests data from third party providers like CatchPoint, Rigor, and Pingdom in determine the reachability of a specific host or application.
2. **Server Monitoring** – Metrics from providers like Rackspace and New Relic, allow Openmix in consider server run time metrics such as memory usage, CPU consumption, free disk space, and network latency in the routing decision. Openmix can use the metrics in make on/off routing decisions or in make graduated routing changes by shedding traffic from a loaded server.
3. **CDN Cost Control** - Ingests Bandwidth and Usage statistics from all the major CDNs and makes this data available real-time in Openmix applications in impact routing decisions.
4. **Customer defined custom data feeds** - Any data at an endpoint you provide can be ingested and made available in a custom Openmix application for use in the routing decision.

Fusion Integrations

Service	Type
Akamai	CDN Bandwidth, CDN Usage
AWS CloudFront	CDN Usage
AWS CloudWatch	Instance Metrics

Service	Type
AWS ELB	Load Balancer Metrics
AWS S3	Custom Data Feed
Azure	Instance Metrics
Catchpoint	Alerts
CDNetworks	CDN Bandwidth, CDN Usage
ChinaCache	CDN Bandwidth
ChinaNetCenter	CDN Bandwidth
Citrix ADC	Custom Data Feed
Datadog	Alerts
Edgecast	CDN Bandwidth, CDN Usage
Fastly	CDN Usage
Fusion Direct	Custom Data Feed
Highwinds	CDN Usage
HTTP GET	Custom Data Feed
HTTP GET with Availability	Custom Data Feed
JSON	Custom Data Feed
Keynote	Web Monitor
Level3	CDN Bandwidth, CDN Usage
Limelight	CDN Usage
MaxCDN	CDN Bandwidth, CDN Usage
New Relic Apdex	Application Score
New Relic Server Monitoring	Instance Metrics
NGINX	Load Balancer Metrics
NGINX+	Load Balancer Metrics
Pingdom	Web Monitor
Qbrick	CDN Usage
Rackspace	Instance Metrics
Rigor	Web Monitor
SFR	CDN Bandwidth, CDN Usage

Service	Type
TCP Ping	Web Monitor
Touchstream	Video Monitoring

Fusion Feeds

The following screen shows all of the configured Fusion data feeds. The list provides an overview of the data feeds and current status.

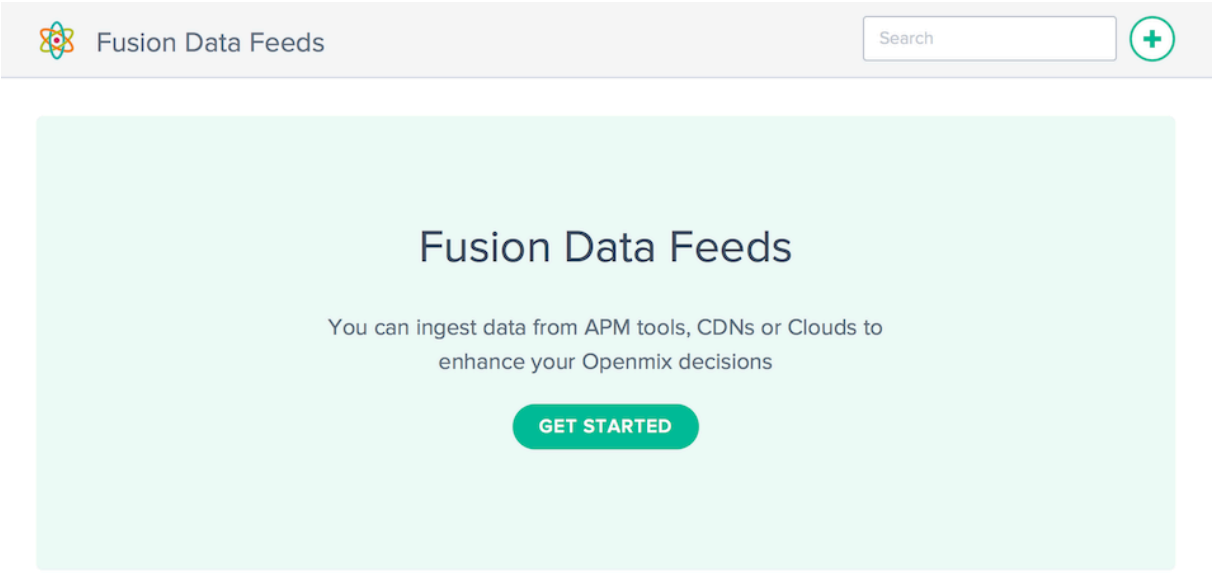
The columns provide the following information:

Heading	Description
Status	The current status of data feed. The status shows either: + green meaning the feed is successfully retrieving data from the service; + yellow meaning the feed is waiting for data in be retrieved from the service; or + red meaning the feed cannot be retrieved from the service
Data Feed Name	The name given in the data feed. Optional, will default in “Service - Platform Name” if not specified.
Service	The name of the service being used by the data feed.
ID	The ID of the data feed. This is needed for accessing Fusion via the API.
Platform Name	The name of the Platform associated with the data feed.

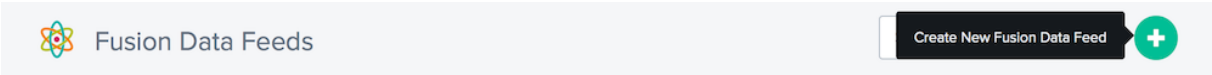
Heading	Description
Run Every	How often the data feed is updated from the service.

Creating Data Feeds

If no Fusion data feeds are configured, a welcome screen prompts you in create a data feed.

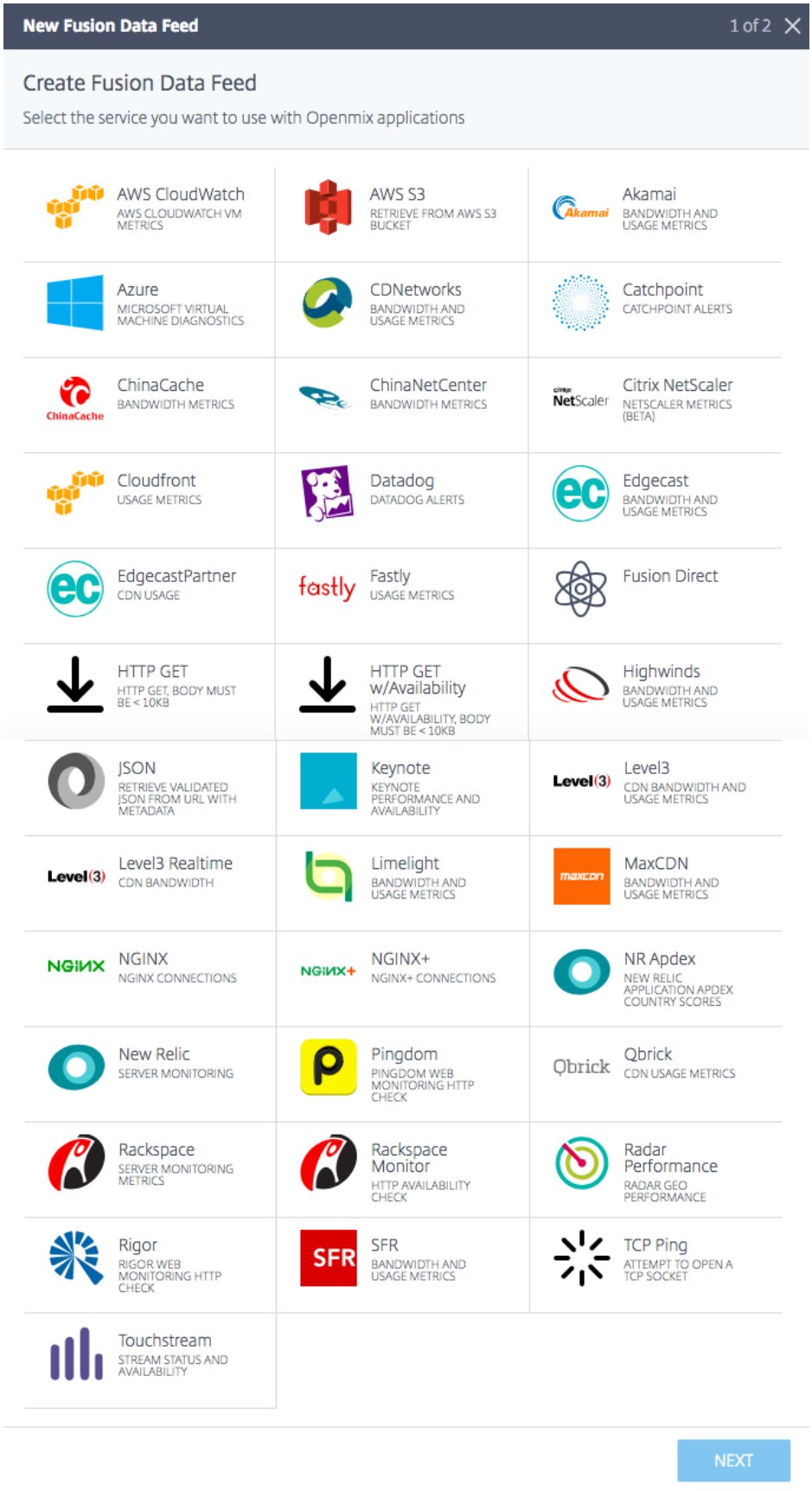


Click the **Get Started** button or **+** in set up a new data feed.

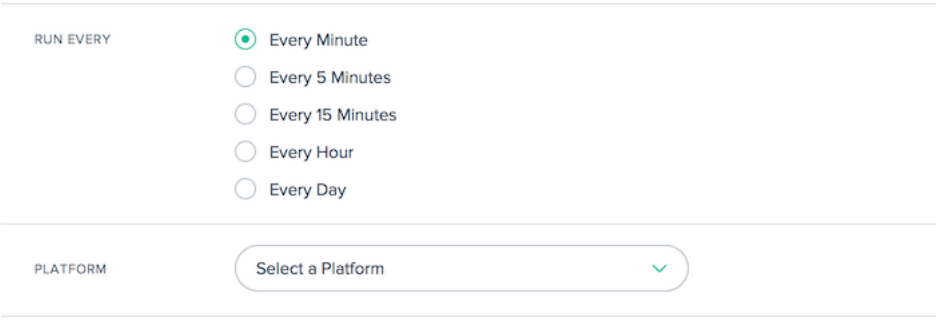


New Data Feeds

Click the icon of the service you would like in integrate and fill out the required configuration fields.



Each service requires different configuration parameters. You need a user name and password or a generated token for authentication and any additional service-specific configuration.



All Fusion data feeds are associated with a platform that was previously created in the Citrix Intelligent Traffic Management portal. This allows the Openmix application in query the external Fusion data for each platform and, based on the routing logic, determine if the platform must be considered available for a routing decision.

Most feeds need in configure the following values:

Input Item	Description
Run Every	How often the data feed is updated from the external service. Fusion calls the service at the interval specified and update the Openmix applications based on the new data.
Platform	The platform associated to the Fusion data in the Openmix application.

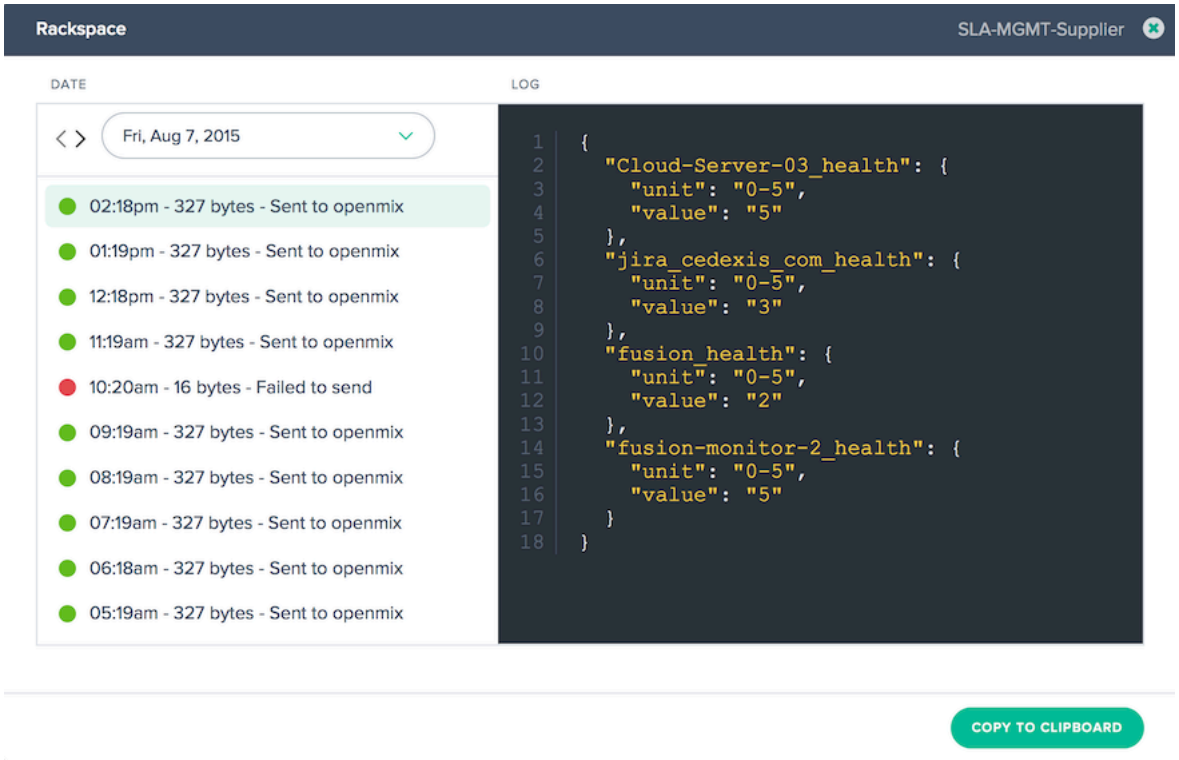
Editing Data Feeds

Editing a Fusion data feed is as easy as clicking the data feed in the table and clicking the **Edit** button.

Once you have changed the configuration, click **Save**. This brings you back in the data feed list with your changes saved and applied in the data feed.

Data Feed History

Fusion collects the last 100 responses from each time it is run in the data feed history. You can view the data feed status, information about the data and the payload returned from the service. After selecting the specific data feed in the list, click the **Log History** button in show the history for the data feed.



To change the date selected, you can click the **<** or **>** buttons to move backwards or forwards from the current selected date or choose a specific date from the list. Select the timestamp of the specific instance and the data returned from the service will be displayed.

Failing Data Feeds

Fusion Quarantine for Failing Fusion Feeds

Fusion Quarantine applies in a customer's failing Fusion data feed, if the feed is configured in run at a polling interval of less than 24 hours. Fusion applies quarantine logic in stop these failing feeds from running. This is done in save resources (CPU/Memory) and avoid any negative impact on other valid Fusion data feeds.

The quarantine logic is applied by "backing off" the failing Fusion feed at gradual intervals. This occurs until the Fusion feed is quarantined for 24 hours. At this point the Fusion feed will attempt in run every 24 hours. The failing fusion data feed is never completely shut down. It will continue in run, at a minimum of two times every 24 hours.

Important:

- The Fusion data feed will always run at least two consecutive times and fail twice before it enters in the quarantine logic. For example, if a one minute feed runs and fails twice consecutively, it will enter into the quarantine logic.
- If at any point Fusion data feed runs successfully, it is removed from the quarantine logic and

will run again at its regularly scheduled interval.

- If at any point the Fusion feed is updated (i.e. if the user entered a bad URL and has corrected it, the Fusion feed will attempt in run again within one minute regardless of the polling interval. If it is successful, it will be removed from the quarantine logic. If it continues in fail, the quarantine logic will be applied.

Global CDN Purge

April 21, 2020

Global CDN Purge is a way to purge data from several CDNs at the same time, which makes it easier to manage multiple CDNs. It allows you connect the CDNs to be purged, specify the URIs to purge across all of the attached services and click the **Purge** button. The purge is initiated across all of the connected CDNs.

Global CDN Purge functionality is built on three main components:

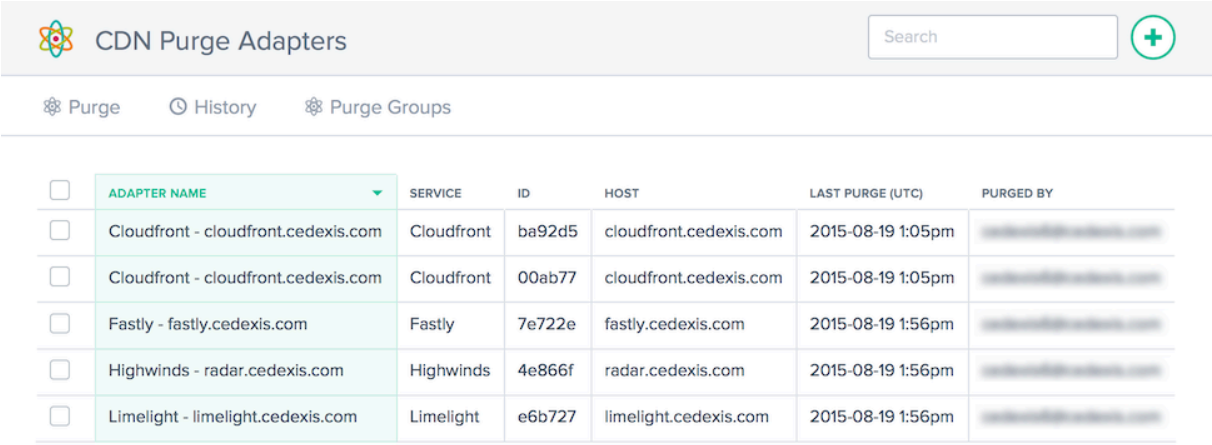
1. **CDN Purge Adapter** – A CDN purge adapter needs to be created for each CDN/host name combination that you would like to purge. The CDN Purge Adapter collects information needed to execute purges, such as: service selection, authentication information, host name, and other service specific information. You need a CDN purge adapter for each host name that is to be purged on a CDN.
2. **URI** – Purges are run against a specific location on the CDNs.
3. **Purge Group** – Purge Groups allow you to create a logical collection of CDN purge adapters and URIs that are purged with one command. For example, you can purge the '/media' directory on 2 different CDNs or a directory that exists in development, test and production environment.

CDN purge adapters must be set up to run purges.URIs and multiple CDN purges can be specified individually but it is recommended that your setup purge groups to manage common purges that are run often.

Global CDN Purge can be accessed from the top level of the navigation menu as CDN Purge.

CDN Purge Adapters

The following screen shows all of the configured CDN purge adapters. The list provides an overview of the configured CDN adapters and allows for purge execution.

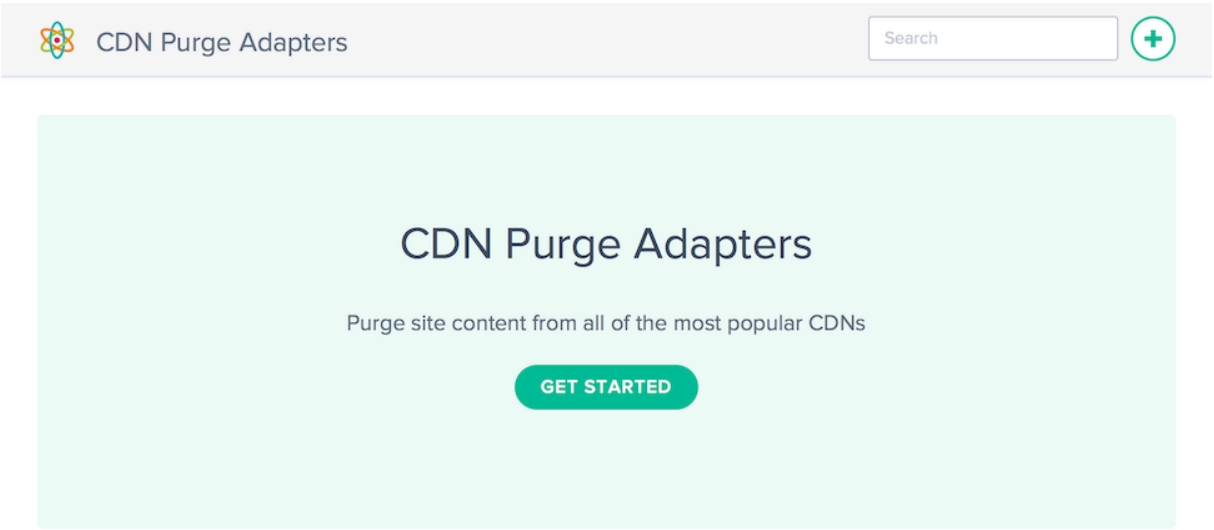


The columns provide the following information:

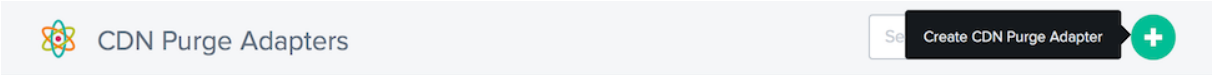
Heading	Description
Adapter Name	The name given to the adapter. Optional, will default to “Service - Host” if not specified.
Service	The name of the CDN service the purge is configured to use.
ID	The ID of the CDN adapter. This is needed for accessing Fusion via the API.
Host	The host that the purge is configured to run against. Services sometimes call this setting: host, host name, platform, and so on.
Last Purge (UTC)	The time and date, in UTC, when the purge was last run.
Purged By	The user who last ran a purge.

Creating CDN Purge Adapters

To use Global CDN Purge, you need to add your CDN and host name configurations. When you first open **CDN Purge**, you are prompted to create a CDN purge adapter.

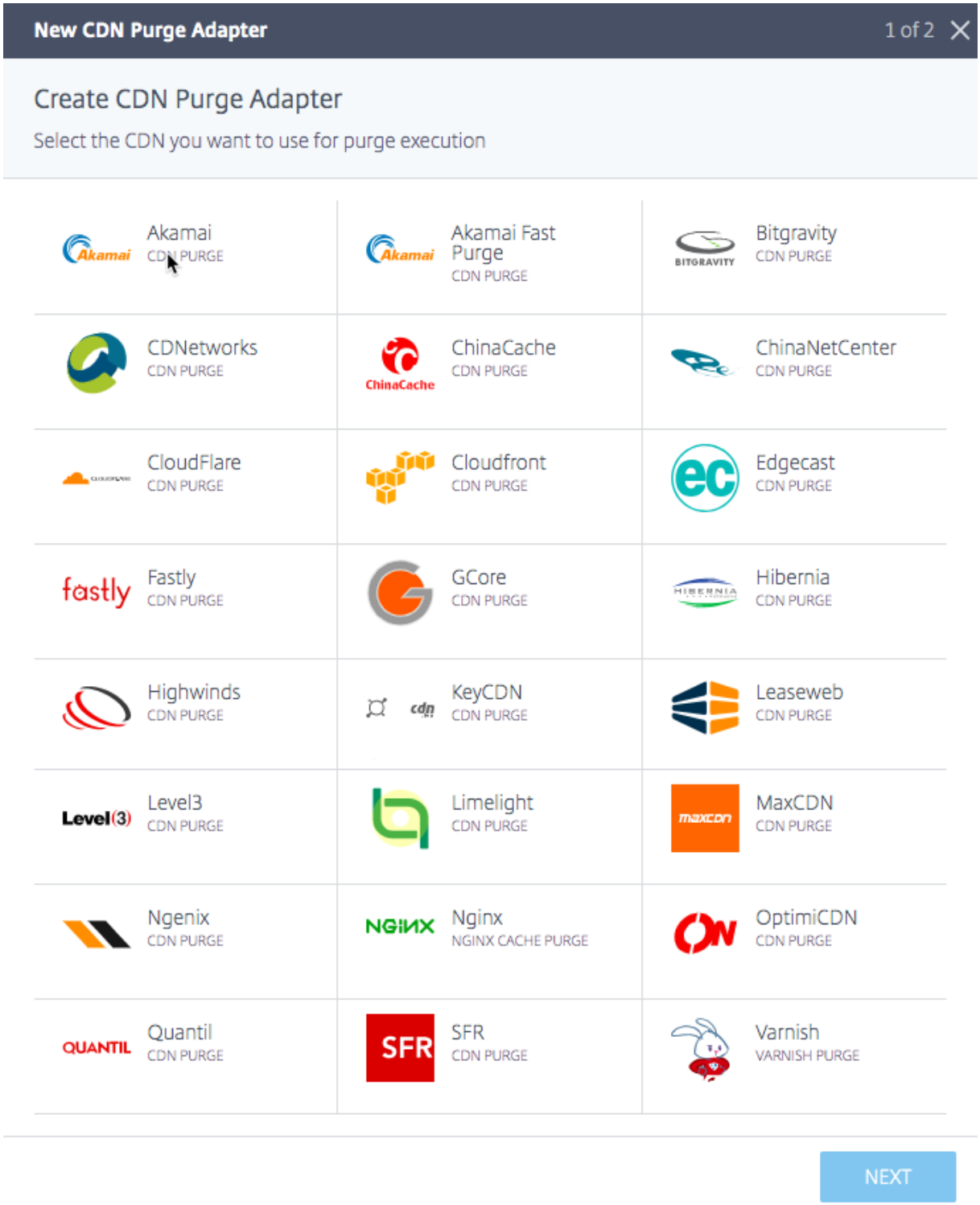


Click the **Get Started** button or **+** to set up a CDN available to purge.

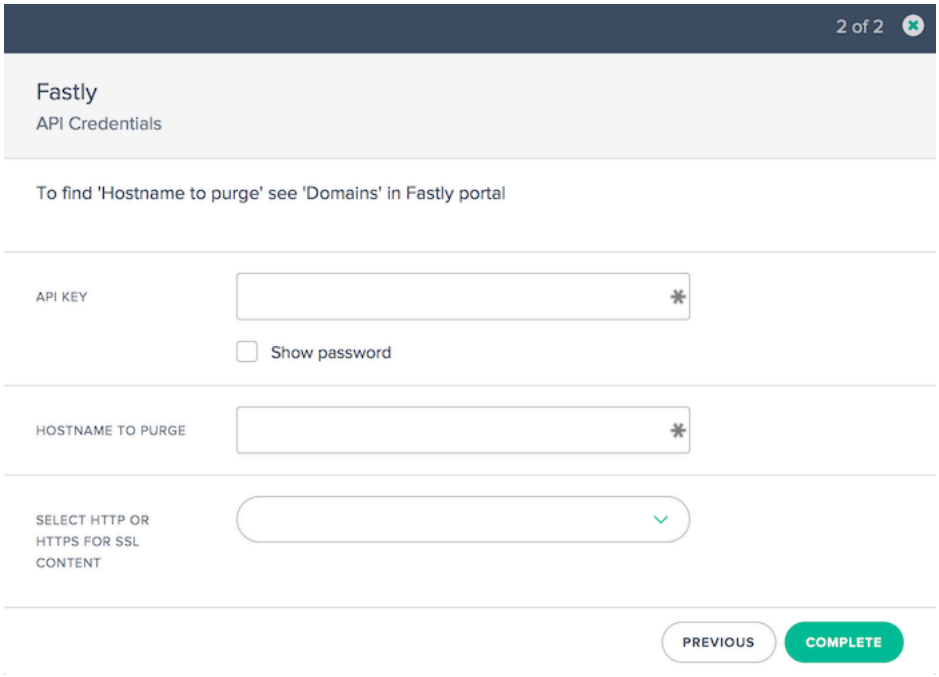


New CDN Purge Adapters

Click the icon of the service for which you would like to create a CDN purge adapter and fill out the required configuration fields.



Each purge adapter requires different configuration parameters. You would need a user name and password or a generated token for authentication and any additional service-specific configuration.



Editing CDN Purge Adapters

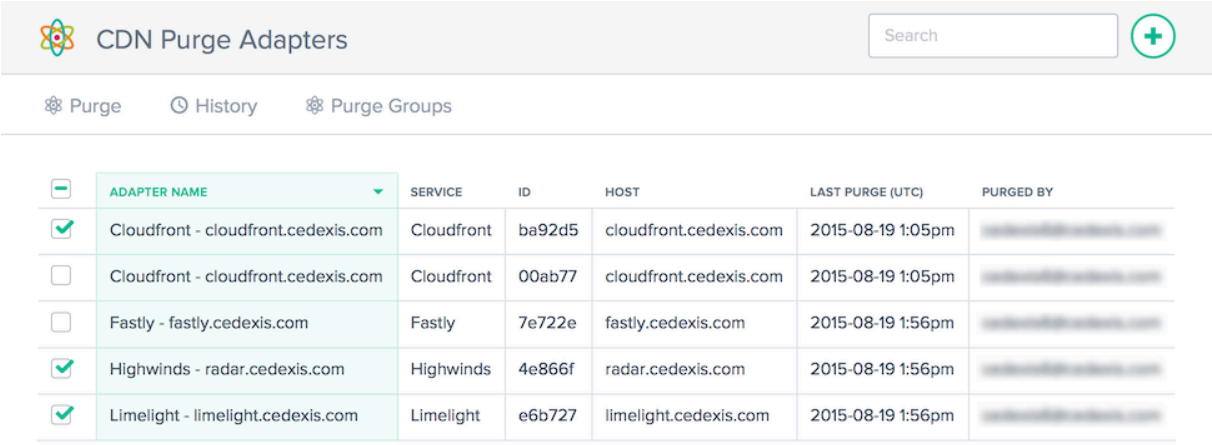
Editing a CDN purge adapter is as easy as clicking the CDN purge adapter in the table and clicking the **Edit** button.

Once you have changed the configuration, click **Save**. This brings you back to the purge adapter list with your changes saved and applied to the specific CDN purge adapter.

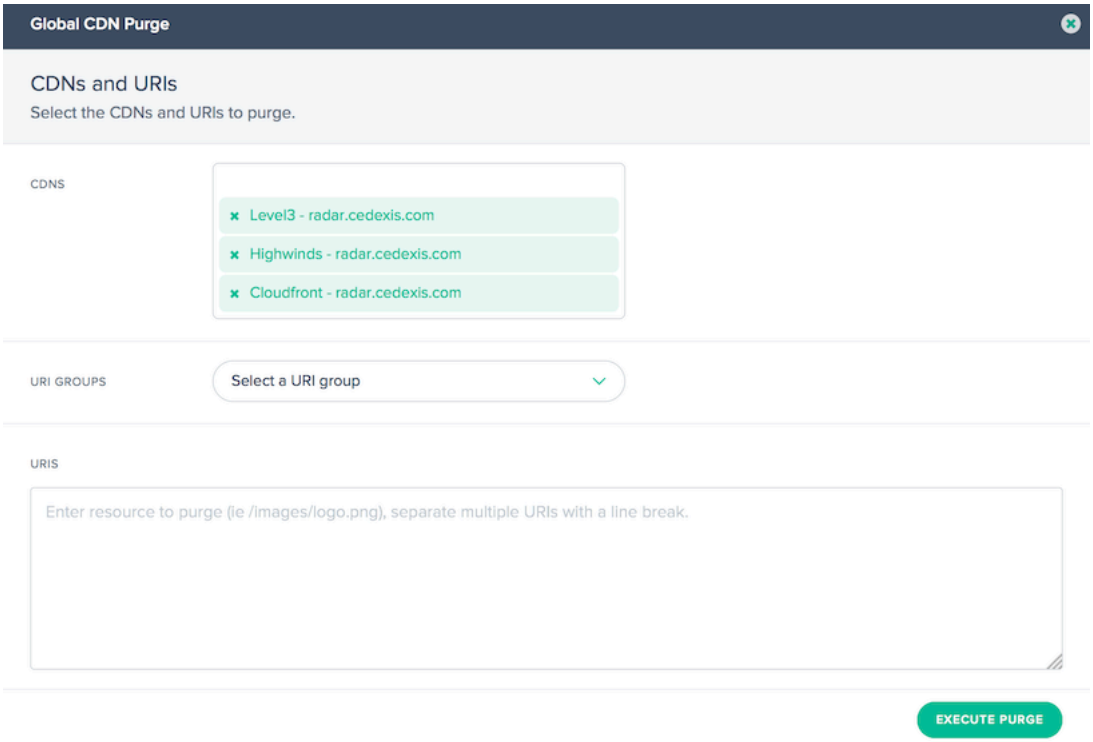
Executing a Purge

To execute a purge, select the CDN purge adapters that must be included in the purge execution.

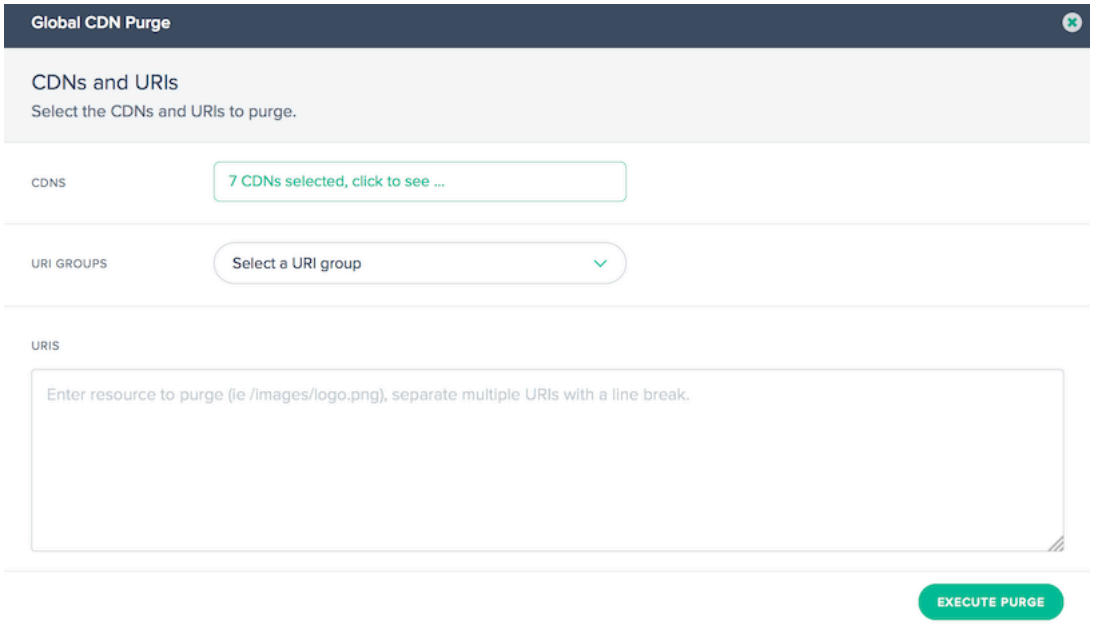
Click the **Purge** button to start the purge process.



The **Global CDN Purge** dialog opens. The dialog shows the CDN purge adapters that were selected and the URIs that are used in the purge execution.

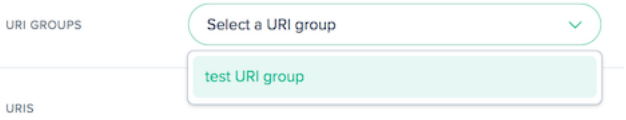


If there are 5 or less CDN purge adapters selected, the purge dialog displays the entire list of the selected CDN purge adapters. If not all CDN Purge Adapters are shown, click the **CDNs** text box that says **X CDNs selected, click to see…** to show all of the selected purge adapters.

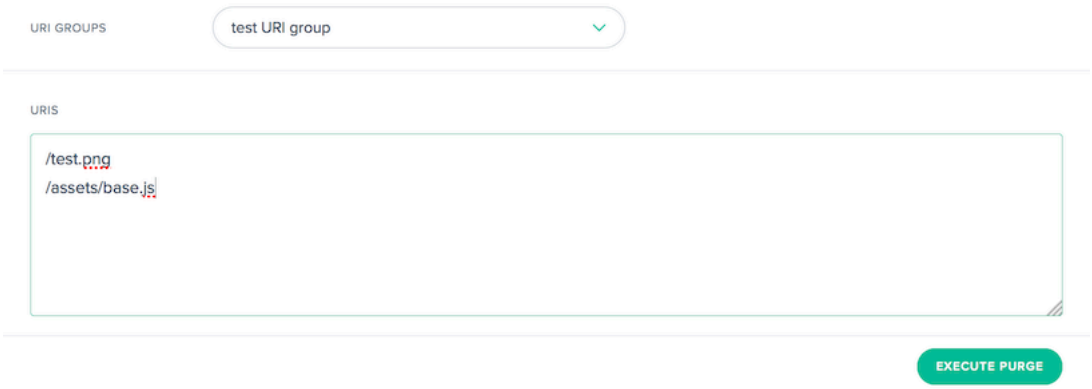


The list can be hidden by clicking the **Hide** button to the right of the list of purge adapters.

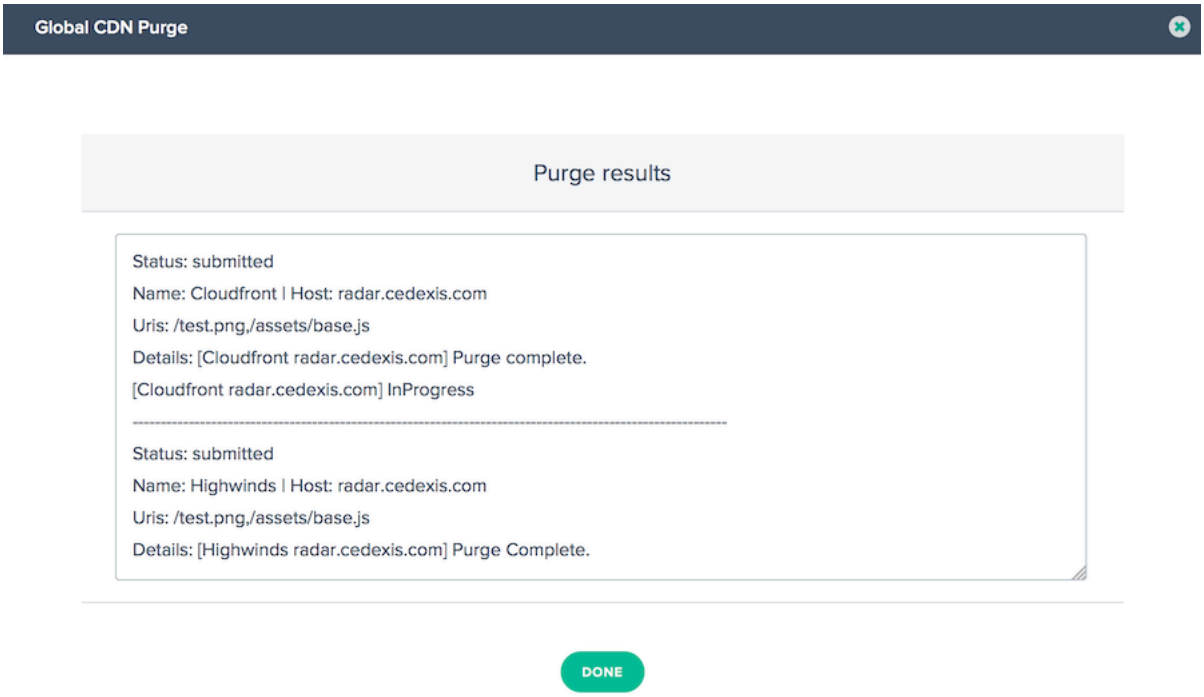
You can populate the URIs used in the purge by manually entering the URIs or by selecting from the available URI Groups. Selecting a URI group populates the URIs input with the URIs from the selected purge group.



Enter or modify the URIs for the resources that must be purged.

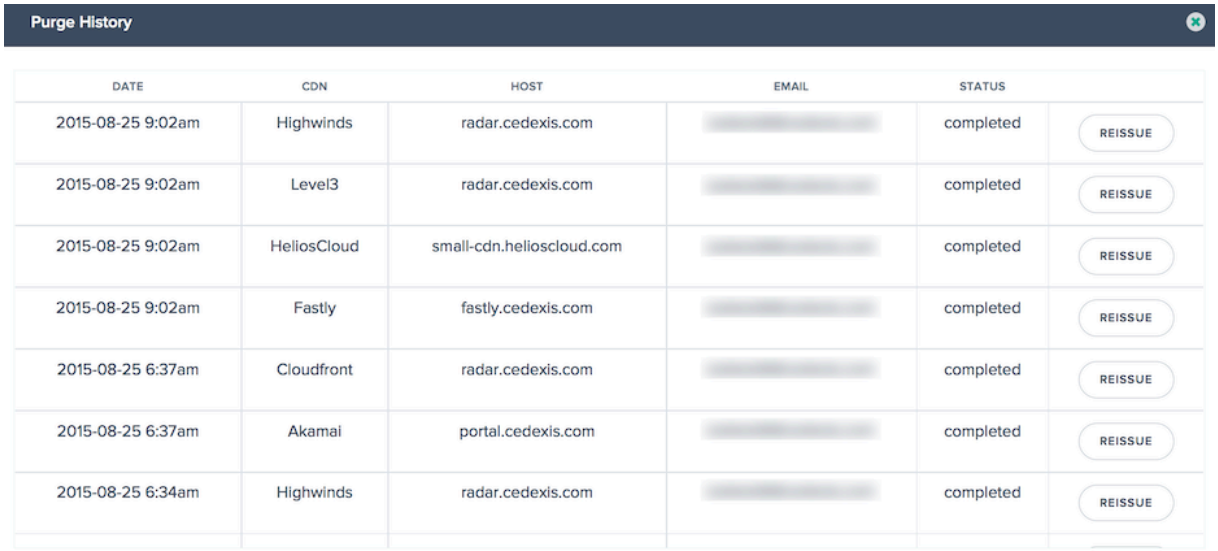


When you are ready to submit the purge request, click the **Execute Purge** button. The purge is submitted to all of the selected CDNs. The submissions and API responses are shown in the **Purge Results** dialog.

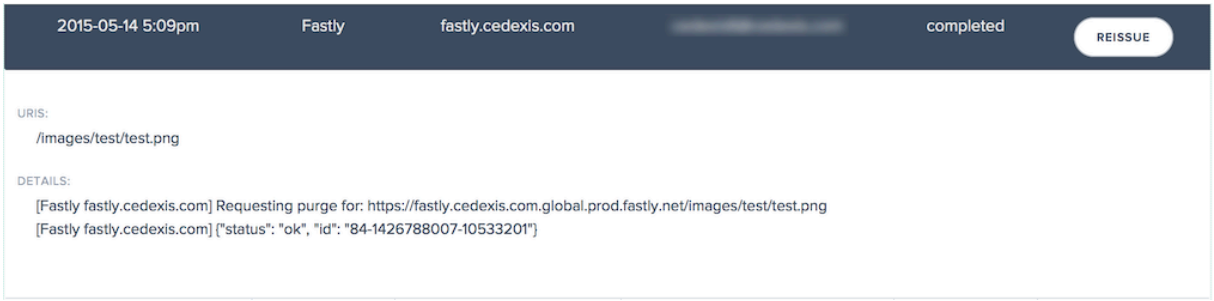


CDN Purge Adapter History

Fusion collects the purge history each time it runs. You can view the purge status, information about the purge and the messages returned from the service. To view the purge history, click the **History** button on the **CDN Purge Adapters or Purge Groups** screens.



The list includes the time and status of the last 100 purge executions. You can see the details of a purge request sent to the CDN service by clicking the desired row in the table. The detail information includes the URIs specified for the purge and the API responses returned from the service during the purge.



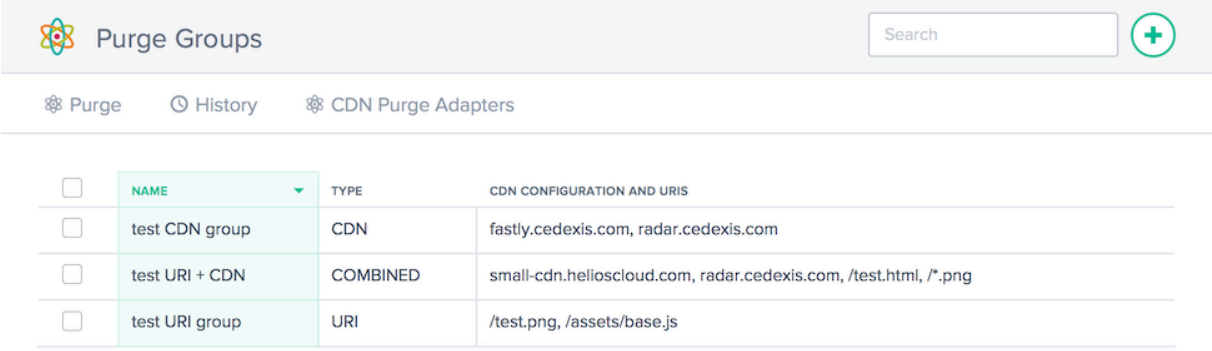
If you would like to rerun a specific purge contained the history, click the **Reissue** button to the right of the purge status information. The purge dialog appears with the data from the previous purge pre-loaded for running.

Purge Groups

Purge groups allow you to organize CDN purge adapters and URIs to make it easy to purge a logical set of resources. For example, you may want to group development, test, and production environments and purge them all at the same time. Or purge all image resources across multiple CDNs at once.

Purge groups can be made up of a collection of CDN purge adapters, purge URIs or both. Typically, a group containing only CDN purge adapters is used for purging different resources across multiple services. A combined group is often used to pre-specify a standard, reusable purge such as “all media across all of my regional websites and CDNs”.

When you have at least one purge group setup, you see this screen when opening CDN Purge.



The columns provide the following information:

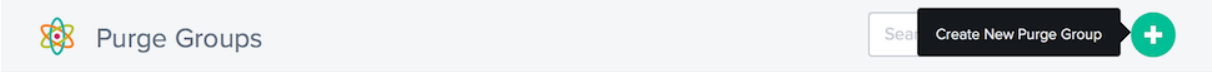
Heading	Description
Name	The name of the purge group.
Type	The content type of the group. + CDN – the purge group contains only CDN purge adapters and the user need to specify URIs when running the purge + URI – the purge group contains only URIs and the user will need to specify services when running the purge + Combined – the purge group contains both CDN purge adapters and URIs; the user will be able to run the purge without need to specify more information
CDN Configuration and URIs	The CDN purge adapters and/or URIs included in the group definition.

Creating Purge Groups

To use purge groups, you need to specify the CDN purge adapters or URIs that must be included. There are two ways to create groups:

From the CDN Purge Adapters page, you can check the desired purge adapters and then click **Create Purge Group**.

From the Purge Groups page, click **+** to create a group.

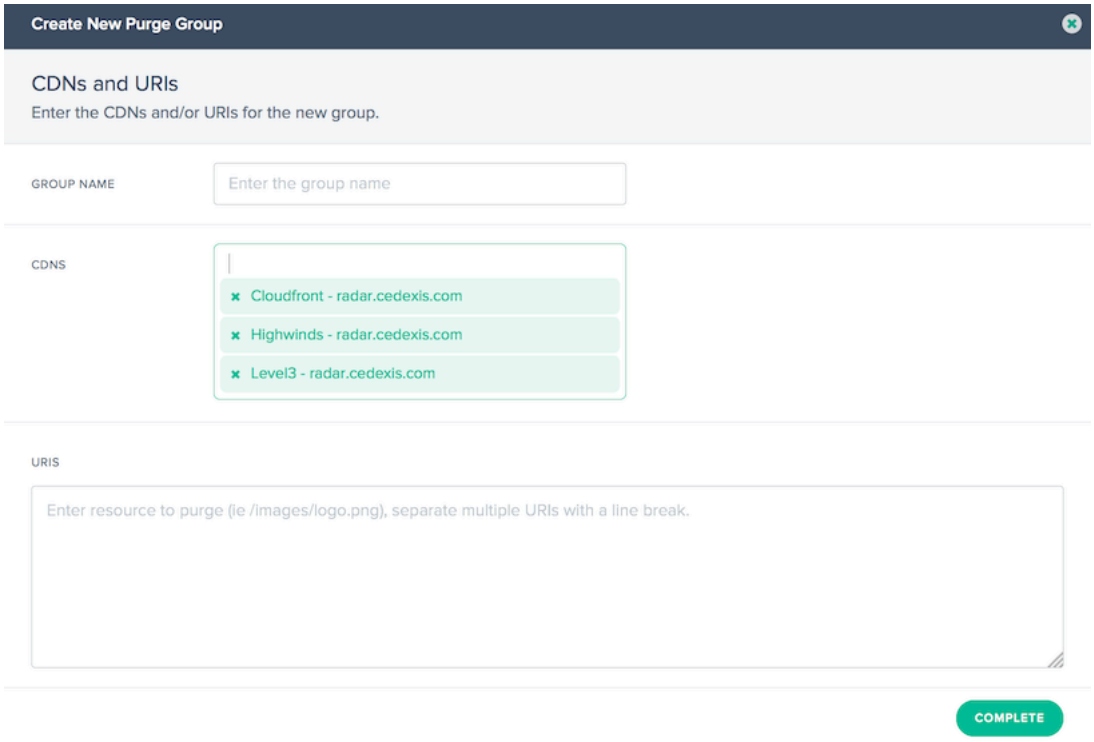


In both cases, the **Create New Group** dialog is shown.

Enter the name for the purge group.

> **NOTE:** You can add or remove CDN purge adapters from the list.

Click **Complete** to create the group.



Running a Group Purge

On the Purge Group page, select one or more groups then click the **Purge** button. The **CDN Purge** dialog opens with the parameters specified by the purge group definition.

Click the **Execute Purge** button to start the configured purge.

Alerts

February 15, 2022

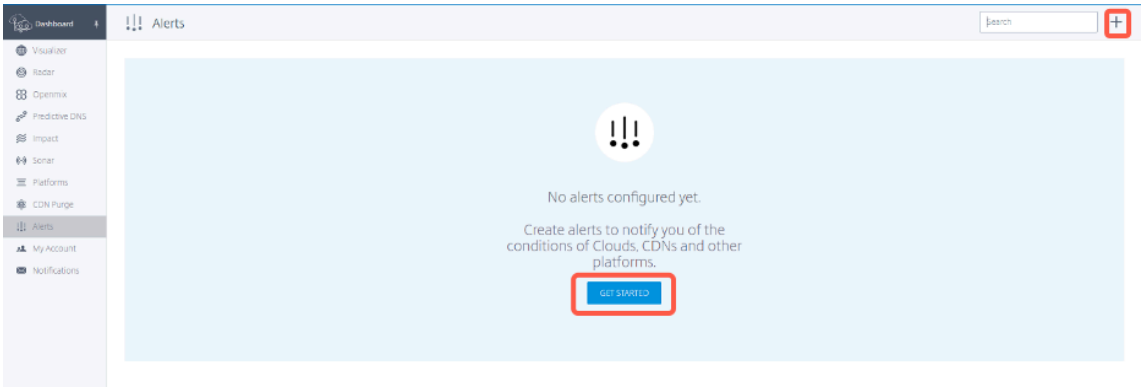
The **Alerts** feature monitors performance issues or anomalies of your configured platforms from an end-user network across the globe.

Create alerts

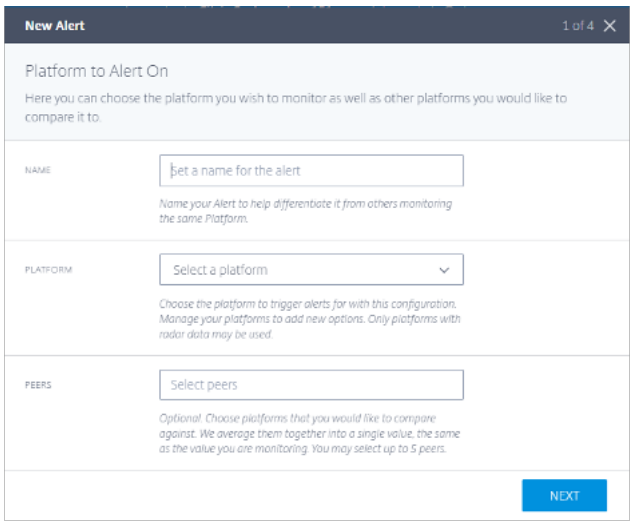
To create alerts that monitor the performance of your platforms, you first have to set up your platforms. On the left sidebar, click **Platforms** to go to the platform screen and set up your platforms.

To add new alert:

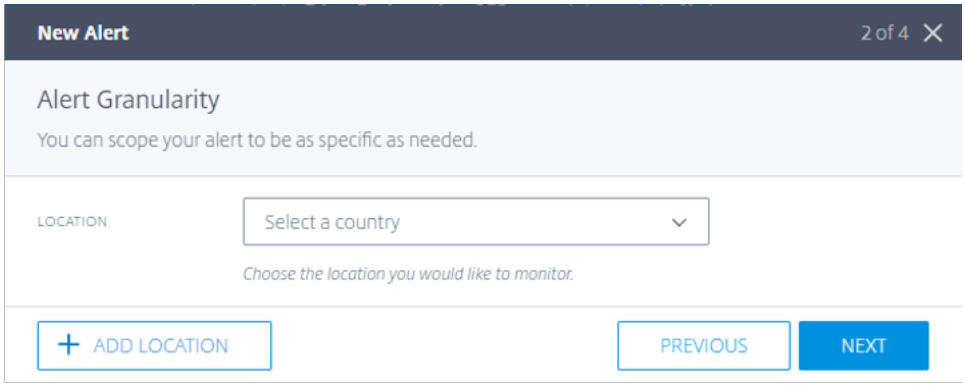
1. On the left sidebar, click **Alerts** to go to the alerts page and create alerts.
2. On the alerts page, click **GET STARTED** or the **+** symbol on the upper right corner.



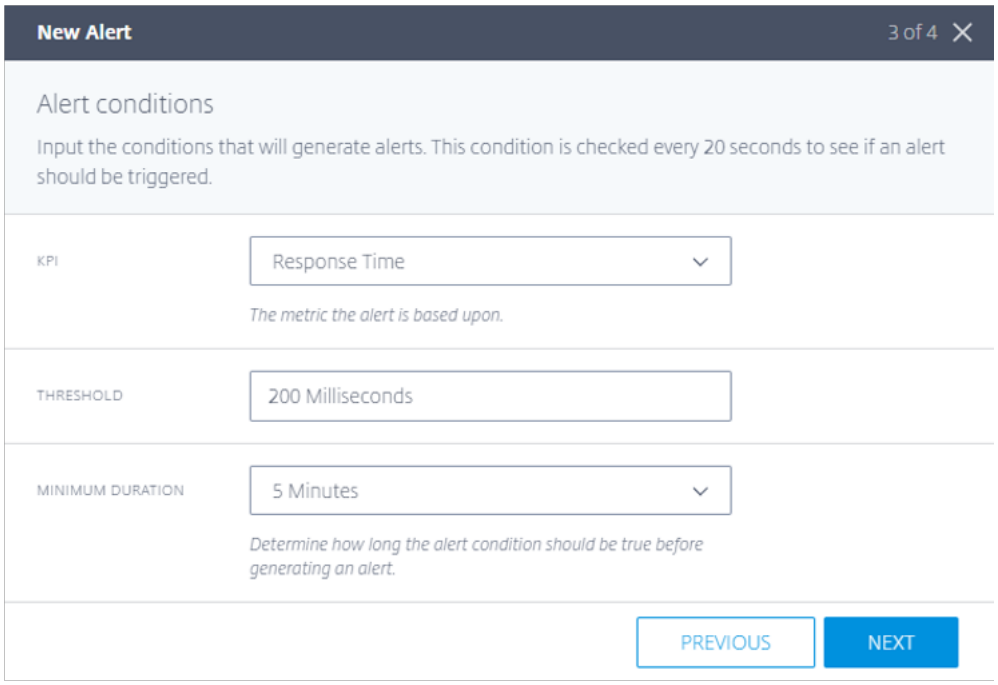
3. On the **New Alert** window:
   - enter the alert name
   - select the relative platform to be monitored
   - select peer platforms to compare with (you can select up to 5 peers). This parameter is optional.
   - click **Next**.



4. Select the **Location** and **Network** that you would like to monitor the alerts for and click **Next**.



5. Select the appropriate **KPI**, **Threshold**, and the **Minimum duration** of the event that triggers the alert.

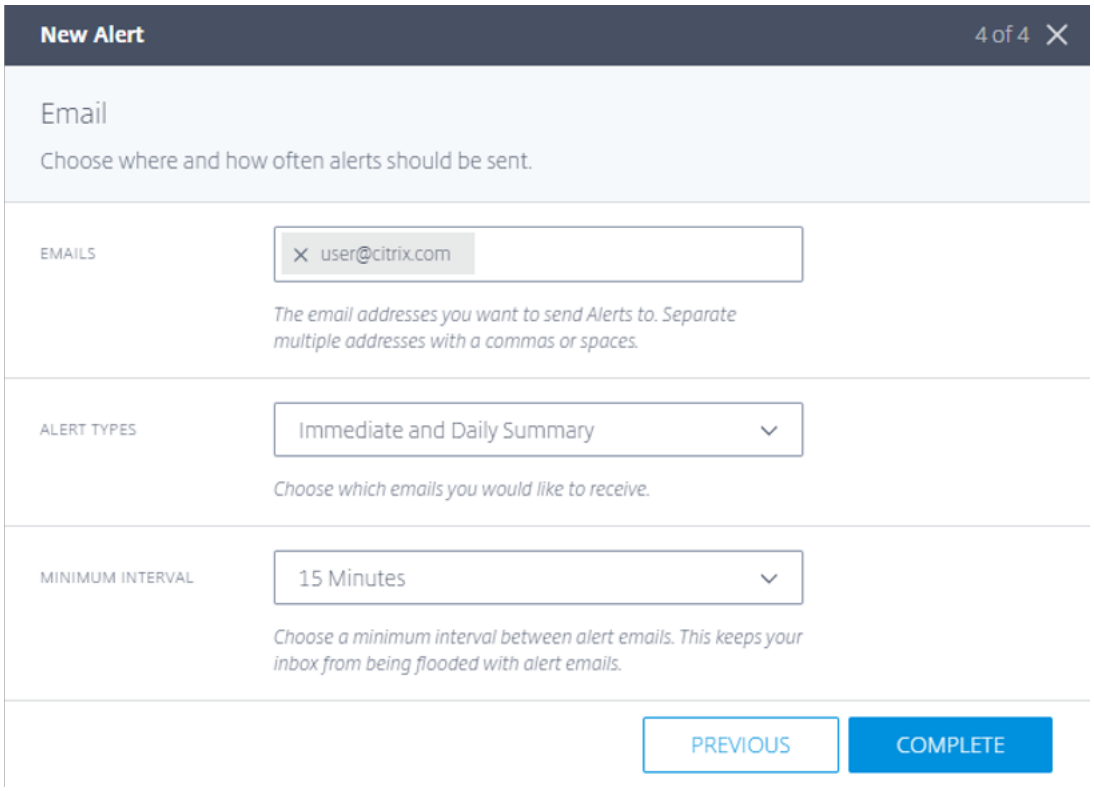


Intelligent Traffic Management provides the following KPIs:

- **Response Time:** The value of the threshold indicates the maximum value (in milliseconds) accepted before the alert is triggered. For an alert to be triggered, the measurement should be greater than the threshold for at least **time ≥ minimum_duration** that a user selected. The same alert will go off after receiving measurement below the threshold again for at least time ≥ minimum duration.
- **Availability:** The value of the threshold indicates the minimum accepted value before the alert is triggered. For an alert to be triggered, the measurement should be lower than the threshold for at least **time ≥ minimum_duration** that a user selected. The same alert will go off after receiving measurement above the threshold again for at least time greater than or equal to ≥ minimum duration.

- **Throughput:** The value of the threshold indicates the minimum value (in kbps) accepted before the alert is triggered. For an alert to be triggered, the measurement should be lower than the threshold for at least **time ≥ minimum_duration** that a user selected. The same alert will go off after receiving measurement above the threshold again for at least time greater than or equal to ≥ minimum duration.

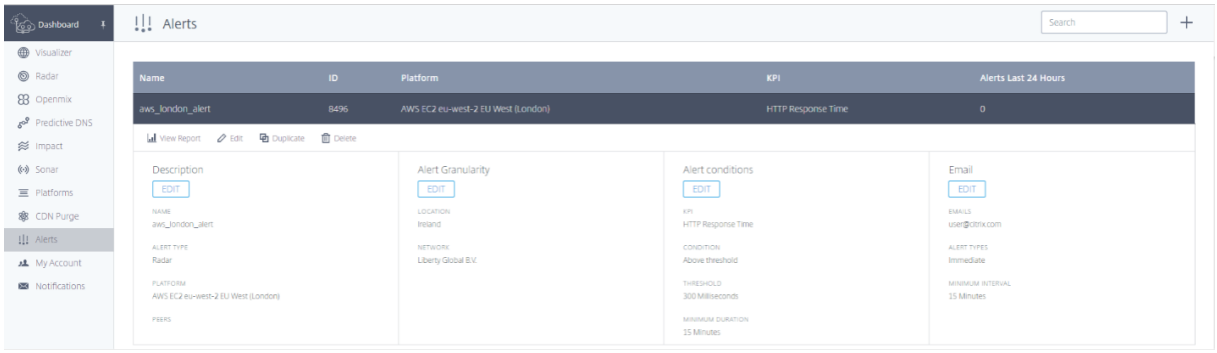
6. Enter the email addresses you want to send alerts to, select the alert type, and select the minimum interval between alert emails.



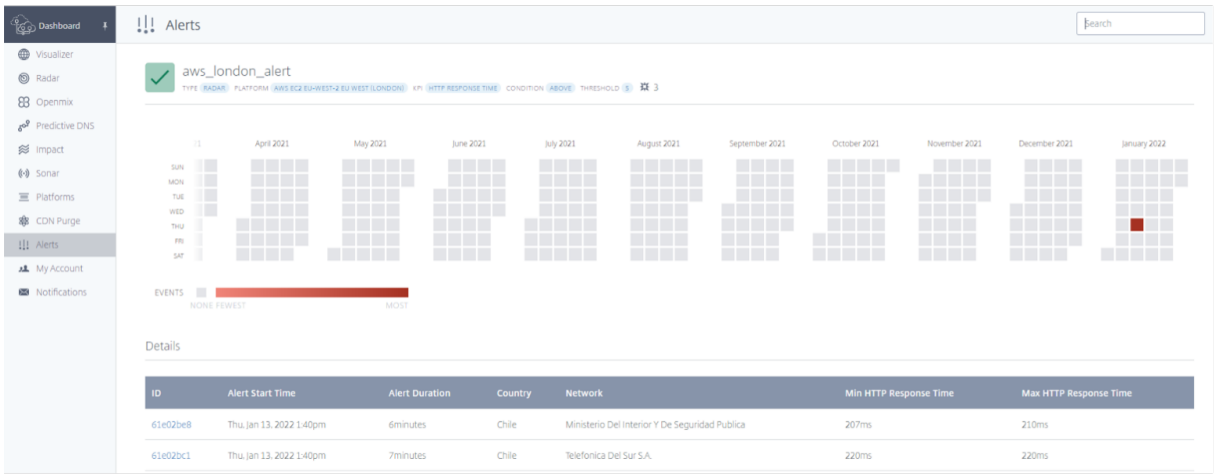
The alert types are as follows:

- **Immediate:** This option sends an email immediately when an alert is triggered.
- **Daily Summary:** This option sends only one email every midnight in Universal Time Co-ordinated (UTC), including all events that are triggered.
- **Immediate and daily summary:** This option is a combination of both immediate and daily email send.

7. After configuring an alert, you can see the alerts in the **Alerts** tab and the global map in the **Visualizer** tab. To view the report for a specific alert, click **View Report** in the **Alerts** tab.



The following report page display events that are monitored per day for every month. For example in the following screenshot, there are 3 incidents monitored in the same day of January 2022.



You can click any specific incident or event to see the details, as shown in the following image:

Network Experience Monitoring

December 21, 2022

Overview

Citrix Network Experience Monitoring (NEM) service (formerly called **Netscope**) enables service providers, enterprises, ISPs, and third party service providers to access detailed Radar measurement logs, and standard reports in the form of summarized actionable data. NEM offers several standard logs and reports that customers can use to measure the quality of their services.

This solution includes "raw" Radar Measurement delivery and access to the Citrix ITM Data API. NEM provides both the granular data (as either raw measurements or data aggregates) and data threshold alerts. These services help with discovery, isolate platform availability, and performance issues in platform peers and the underlying ISPs.

Radar "Raw" Measurements: Radar measurements provide per-event granular information that is batched daily. Radar measurements include public community and private measurement data collected by the tag. Data such as availability, response time, throughput for HTTP and HTTPS measurements are included. The following data fields are provided:

- Provider ID, resolver IP, obfuscated (/28) client IPs
- Obfuscated referrer header, user agent, end-user ASN
- Geo data for resolver and client fields

Radar Metrics that are available in the "Raw" Measurements are:

- Availability, Response Time, and Throughput (when measured)
- DNS Lookup Time (optional), TCP Connect Time (optional), and Secure Connect Time (optional)
- Latency (optional)
- Download Time (optional)

Radar Measurements are available to allow customers to do their own analysis of collected data. The data set includes information on provider performance and availability (errors) for a range of communication protocols.

Log file data is available for 7 days, from an AWS S3 or Google Cloud Storage bucket. Customers can retrieve log files of community and private data using standard bucket access methods.

Real-time Radar "Raw" Measurements (optional): Raw Radar measurements are delivered in real-time to an AWS S3 bucket. These logs are generally available within 5 minutes of collection. They provide as much granularity as the Radar Raw Measurements noted earlier.

Data API: The Citrix ITM Radar data API provides aggregates of the Radar public community and private measurement data. Data is updated continuously, and batched approximately every 60 seconds

for retrieval by the API. The data API is provided to allow customers to integrate Radar data into their own reports and dashboards.

Log Sharing and Delivery

- Radar logs can be delivered real-time and daily.
- Reports run daily.
- Results are saved to AWS S3 (S3) or Google Cloud Storage (GCS).
- Logs and reports both have a 7-day retention period and are automatically deleted one week after creation.
- Reports are usually in TSV (tab-separated value) or JSON format depending on the type of report.

Customers are given login information to access the S3 and GCS buckets. A command line tool like s3cmd or the AWS CLI for S3 or the gsutil for GCS can be used to log in. The S3cmd config file recognizes the access keys received via the Portal UI and helps the user connect to the S3 bucket.

The AWS CLI needs to be installed in the customer's computer to connect to S3 and access the logs. For GCS, the customer receives the access key file as a download via the Portal UI which can be used with the gsutil tool. For more information refer to the FAQ.

Customers receive email notifications as and when reports are available.

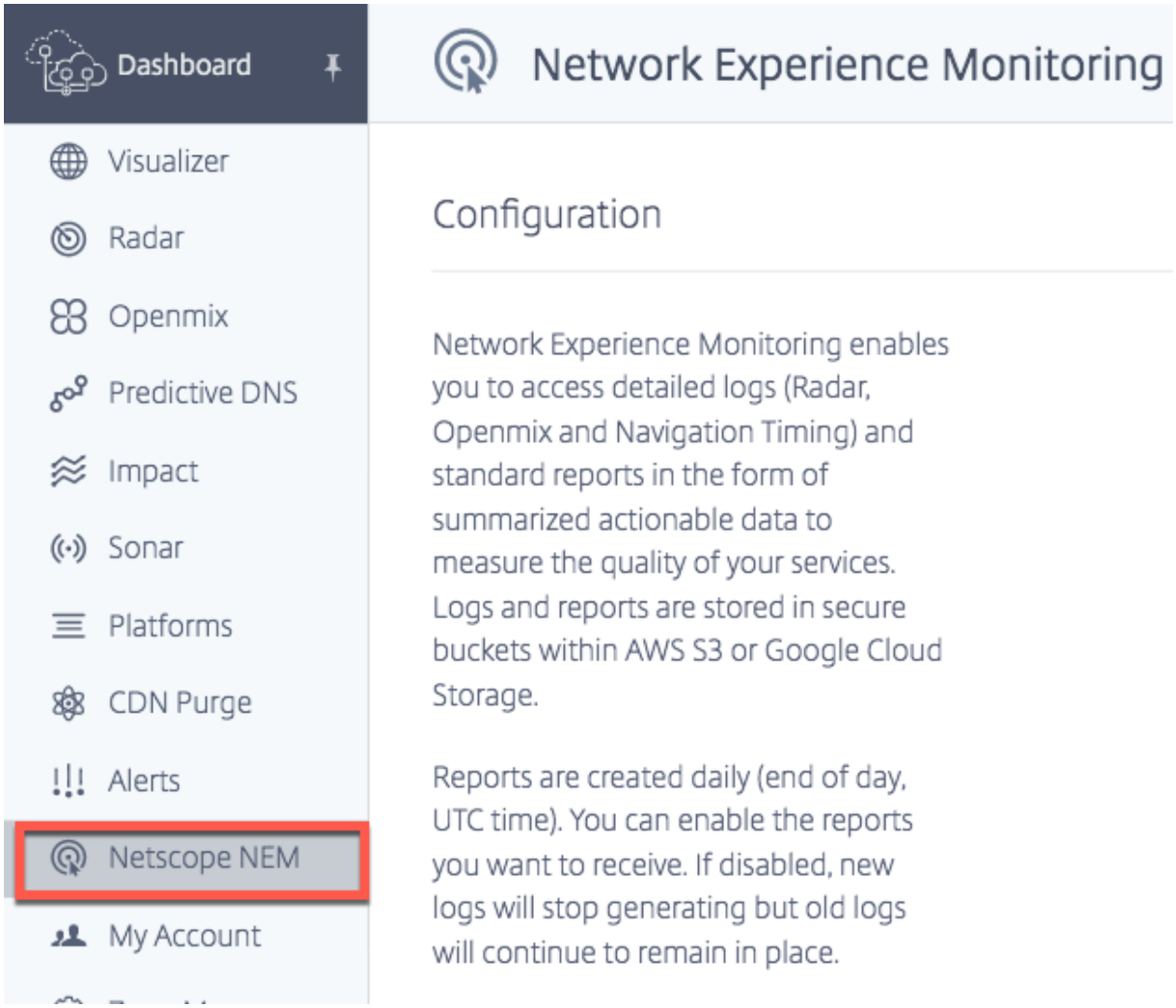
Platform Settings

You must configure your platform to support and produce the data required for Netscope NEM. Before you start, make sure that the following settings are enabled for your platform:

- For Anonymous Best reports, enable **Radar Probe Settings**.
  - For Anonymous Best RTT, enable **Response Time, and Availability**.
  - For Anonymous Best Throughput, enable **Throughput, and Availability**.
- For Cache Node ID reports, enable **Radar Probe Settings**; and in **Advanced Radar Settings**, enable **Node ID**.
- For Resource Timing Details enable **Include Timestamps** in **Advanced Radar Settings**.

Navigation

From the main menu select **Netscope NEM**. The **Network Experience Monitoring** Configuration page opens.



Platforms and Networks

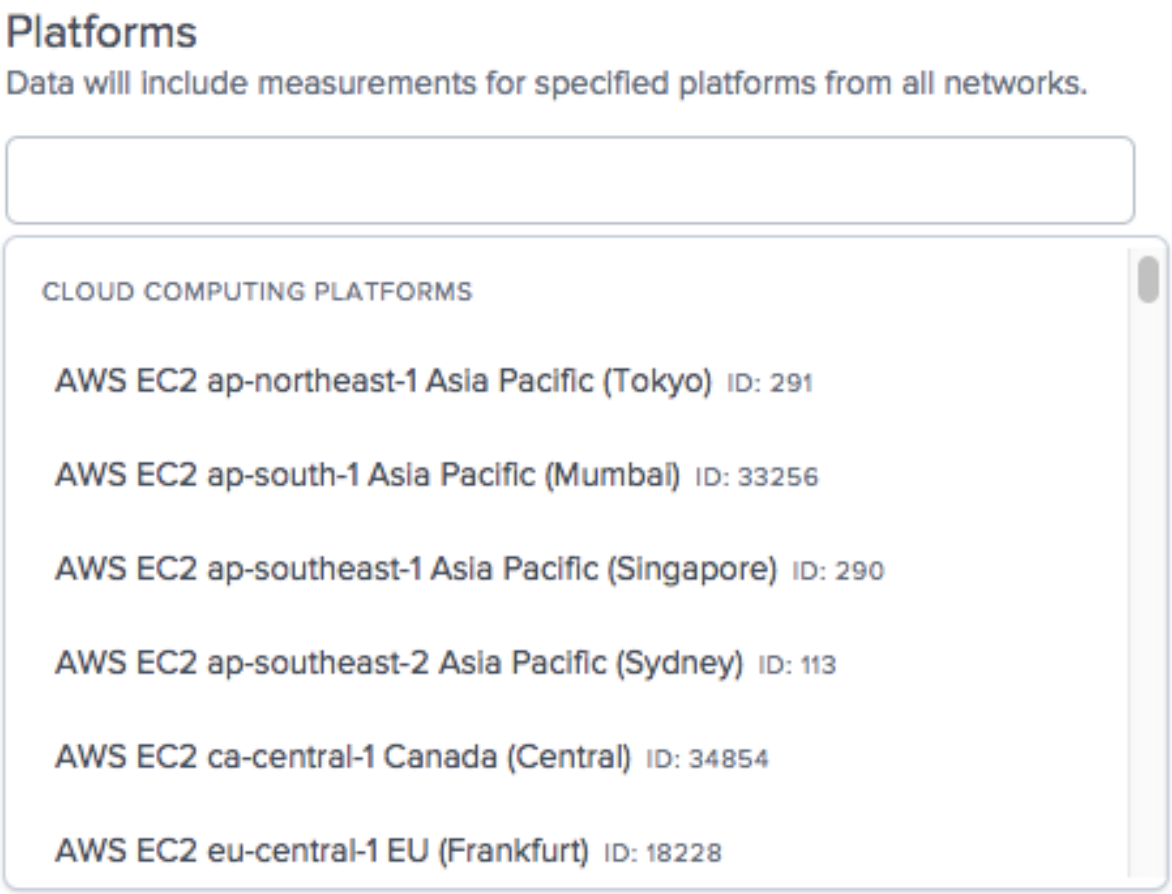
Select required **Platforms** or **Networks** (or both) to start the configuration process.

> **NOTE:**
>
> Logs and Reports can be configured and generated only if at least one **Platform** or **Network** is selected.

The summarized data that the customer receives include Radar measurements of selected Platforms (for all associated networks), or selected Networks (for all associated platform measurements).

Selecting Platforms

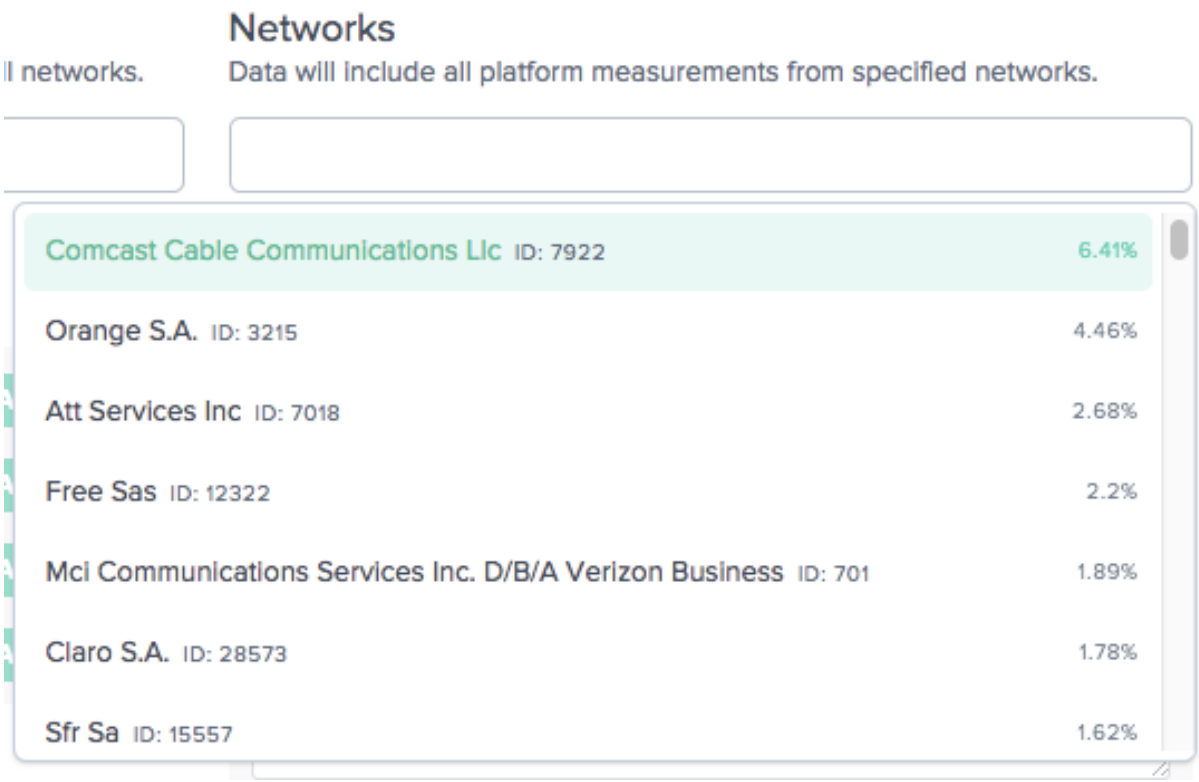
For content service providers or enterprises, select platforms such as CDNs, clouds, data centers, or other end-points. Select platforms for which measurements are required.



Selecting Networks

For ISPs, select the **Networks** from the list associated with different platforms or endpoints for which measurements are required.

> **NOTE:**
>
> If you don't find the required platform on the list, you can configure it in the **Platform** section of the portal. For unavailable Networks, contact the support team.



Platform Reports

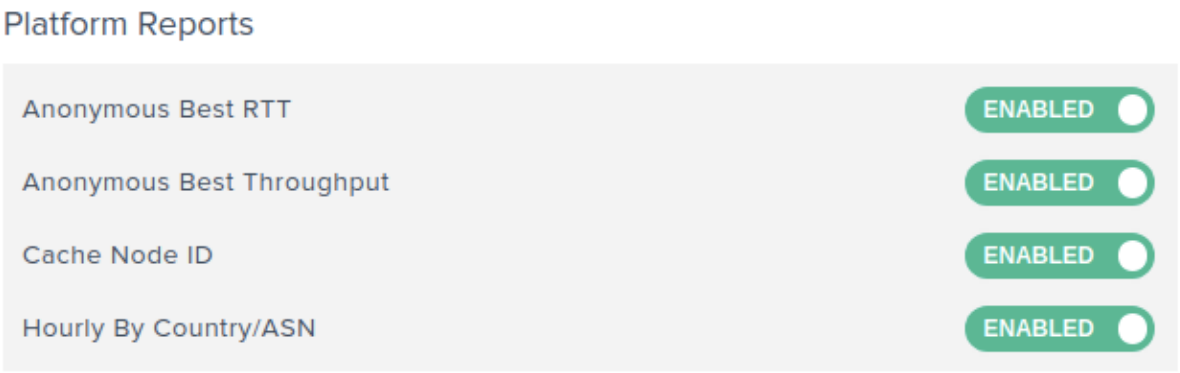
There are four types of **Platform Reports**:

1. **Anonymous Best for Round Trip Time (RTT)**
2. **Anonymous Best for Throughput**
3. **Cache Node ID**
4. **Hourly by Country/ASN**

For log descriptions go to Radar Log Descriptions and Reports for Service Providers and Enterprises.

Enable Platform Reports

Click the toggle button to enable or disable the reports you want to receive. If you disable an existing report, new logs aren't generated but old reports stay in the current location.



Anonymous Best Report for Platforms

- These reports help providers compare their performance to that of other platforms within their peer group i.e.within the same country, region, or ASN.
- The performance data of the top 15 providers in the peer group is aggregated based on the same categories. The best is listed next to the specific provider's best value.
- Anonymous Best Report for SSL Platforms is available so that their performance can be compared with other SSL platforms.
- The client IPs are truncated to /28.
- The results of the "best" provider helps clouds/CDNs focus performance efforts on high volume or business-critical ASNs that are competitively weak to their peers.
- The report provides details on performance broken down by DNS resolver IP, Client IP /28, and the caching Node that served the objects. The same is compared with the "best" platform for the same criteria.

Available for RTT and Throughput.

- For log descriptions refer to Radar Log Descriptions and Reports for Service Providers and Enterprises.

Cache Node ID Report for Platforms

- This report is used to identify the specific server or data center that responded to a request and help diagnose server issues.
- It provides the ID of the data center or machine that responded to a specific request.
- It helps to understand why the performance through a specific node (POP or machine, or node ID), was good or bad.
- The performance consists of response time, throughput, availability (probe type), the DNS resolver IP, Client IP /28, and the caching Node that served the objects.

- For log descriptions refer to [Radar Log Descriptions and Reports for Service Providers and Enterprises] (#radar-log-descriptions-and-reports-for-service-providers-and-enterprises)

Hourly by Country/ASN

- This report helps to verify if your providers' performance varies significantly during a day.
- It shows the time when the measurements were taken truncated down to the hour, for example `2018-03-11T23:00:00`.
- For log descriptions refer to Radar Log Descriptions and Reports for Service Providers and Enterprises.

Network Reports

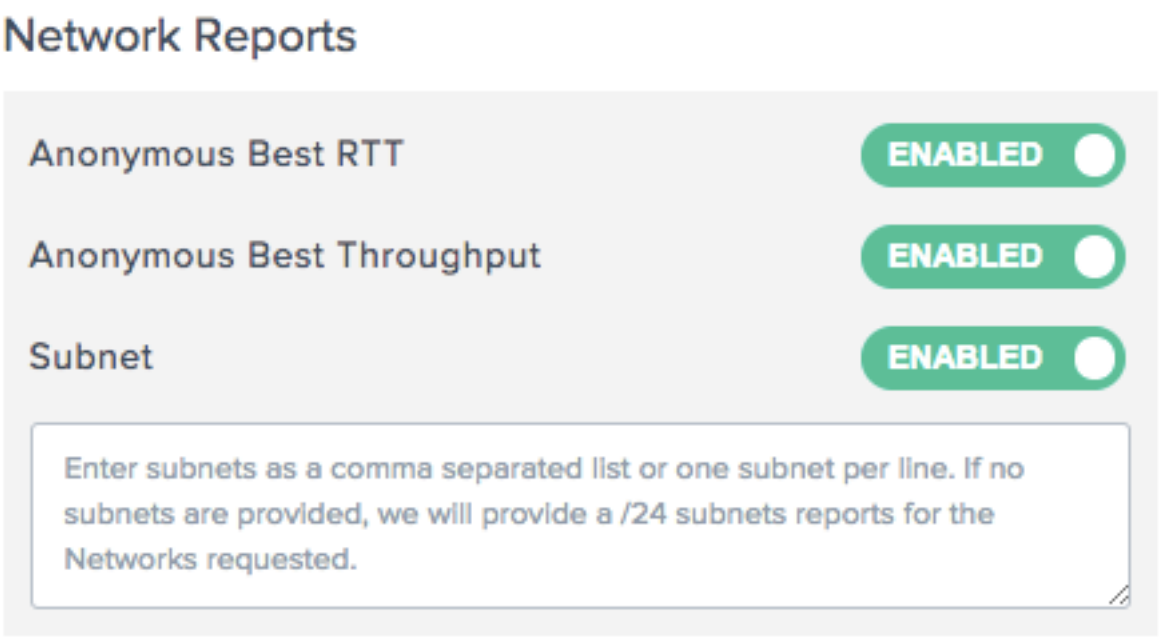
There are three types of **Network Reports**:

1. **Anonymous Best for Round Trip Time (RTT)**
2. **Anonymous Best for Throughput**
3. **Subnet**

For log descriptions refer to Radar Log Descriptions and Reports for ISPs.

Enable Network Reports

Click the toggle button to enable or disable the reports you want to receive. When disabled, new logs stop generating but old reports are in place.
To generate a subnet report, enter the specific subnets of your networks. If there aren't any subnets entered, reports are generated using the ASN CIDR block as the default subnet.



Anonymous Best Report for ISPs

- In the Anonymous Best report for ISPs, a peer group is used for the “best” comparison. The peer group is based on the location of the ISP. It’s usually the 10 most measured ISPs in a specified country, with a minimum of over 1,000 sessions.
- The results of the “best” ISP helps ISPs focus performance efforts on high volume or business-critical Platforms and areas that are competitively weak to their peers.
- The report provides details on performance broken down by geography and Platform, and compares it with the “best” ISP for the same criteria.
- Available for RTT and Throughput.
- For log descriptions refer to Radar Log Descriptions and Reports for ISPs.

Subnet Report for ISPs

- This report provides ISPs with information on how the specific subnets of their networks are performing for users through the platforms that we measure.
- It provides information about the service provider that responded to a specific request.
- It helps to understand the performance by a network subnet.
- The performance consists of response time, throughput, availability (probe types), the DNS resolver IP, Client IP /28, and the subnet of the user.
- For log descriptions refer to Radar Log Descriptions and Reports for ISPs.

Radar Logs

- Radar logs are available for Platforms and Networks.
- They include a subset of the fields available in the raw logs, with some data anonymized: client IP /28, Referer MD5 hashed.
- Every measurement taken for public platforms is provided, regardless of the page that generated the measurement.

> **NOTE:**
>
> NEM never exposes full client IPs. Instead, it exposes the /28. For example, an IP of 255.255.255.255 is shown in a report as 255.255.255.240/28.

Log Frequency

Radar Logs can be generated daily (every 24 hours) i.e.end of day, UTC time. Logs can also be generated in real-time (minute by minute).

File Format

Choose **TSV or JSON** to receive logs and reports in either of these formats.

Measurement Type

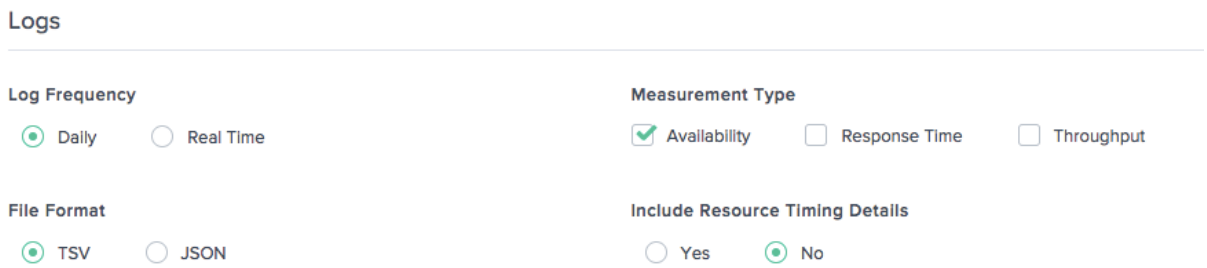
You can configure logs for the following measurement types: Availability, Response Time, and Throughput. In the report, 1: Availability, 0: HTTP Response Time, and 14: HTTP Throughput.

Resource Timing Details

You can choose to also include Resource Timing details by clicking the **Yes** or No buttons.Resource timing details include,

- DNS Lookup Time
- TCP Connection Time
- Secure Connection Time
- Download Time

For log descriptions refer to Radar Log Descriptions and Reports for Service Providers and Enterprises.



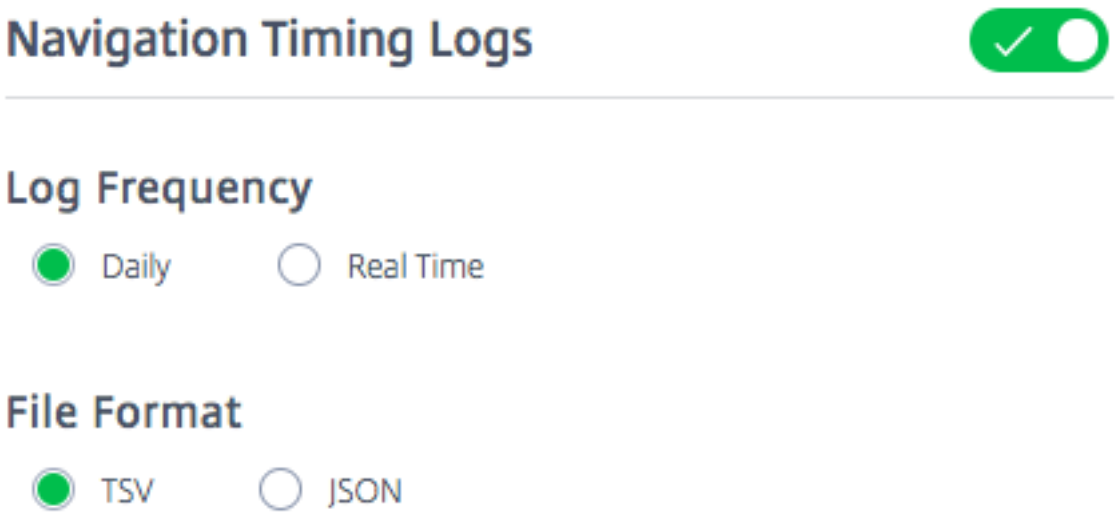
Navigation Timing Logs

Log Frequency

Navigation Timing Logs can be generated daily (every 24 hours) that is, end of day, UTC time. Logs can also be generated in real-time (minute by minute).

File Format

Choose **TSV** or **JSON** to receive Navigation Timing logs in either of these formats. For log descriptions refer to Navigation Timing Log Descriptions.



Openmix Logs

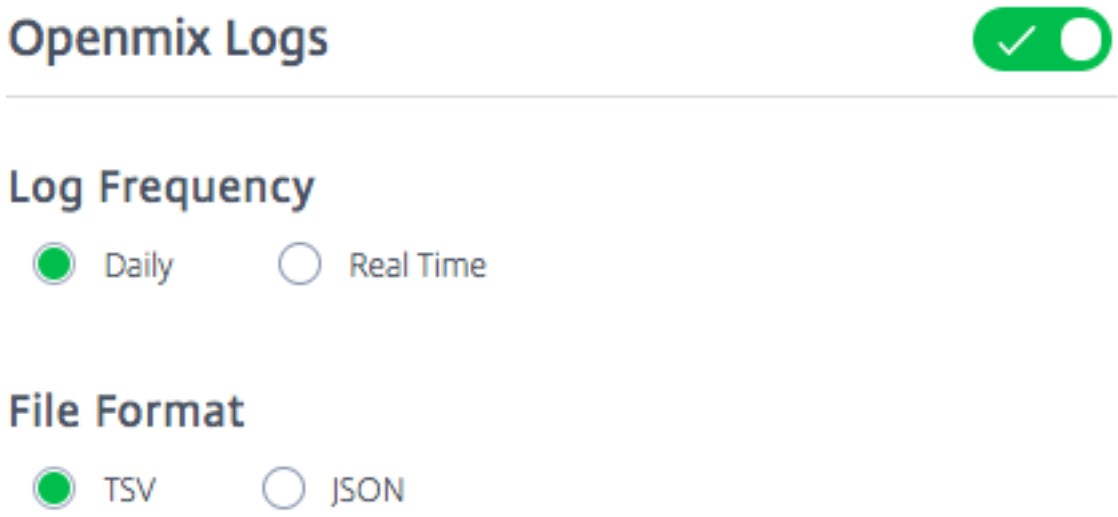
Log Frequency

Openmix logs are generated in real-time (that is, minute by minute). These logs provide real-time measurements taken for Openmix customers.

File Format

Choose **TSV** or **JSON** to receive Openmix and HTTP Openmix logs in either of these formats. JSON is however the recommended format.

For log descriptions refer to Openmix Log Descriptions.



Cloud Service Delivery

This option allows you to select the mode of delivery. You can choose to receive logs and reports in either the AWS S3 bucket or in the Google Cloud Storage (GCS) bucket.
You can access the S3 and GCS buckets with the login information provided and use s3cmd or the AWS CLI for S3. and gsutil command line for GCS.

AWS S3

For logs and reports to be delivered to the AWS S3 bucket, select **AWS S3**.

Location

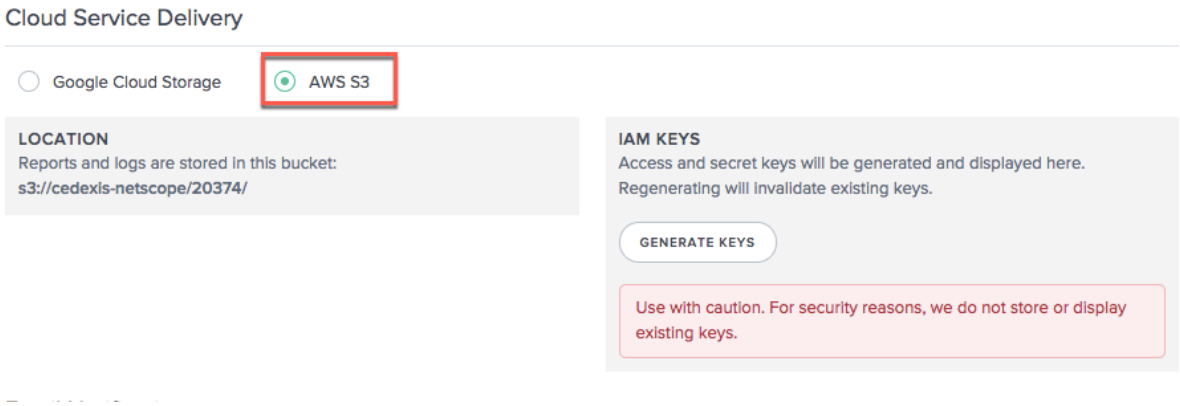
The Location represents the bucket in AWS S3 where the logs and reports are saved.

IAM Keys

If you select the **Generate Keys** button under AWS S3, the AWS IAM Keys (Access and Secret keys) are generated and displayed under IAM Keys. Be sure to record the keys because they aren't stored anywhere for viewing later.

> **NOTE:**
>
> The pair of access and secret keys are the only copy of the private keys. The customer must store them securely. Regenerating the new keys invalidate the existing ones.
> The S3cmd config file recognizes the access keys (received via the Portal UI) and helps the customer connect to the S3 bucket. The AWS CLI needs to be installed in the customer's machine to connect to the S3.

For info on how to use the access and secret keys with s3cmd to download reports from the S3 bucket, refer to the FAQ.



Google Cloud Storage

For logs and reports to be delivered to GCS, select **Google Cloud Storage**.

Location

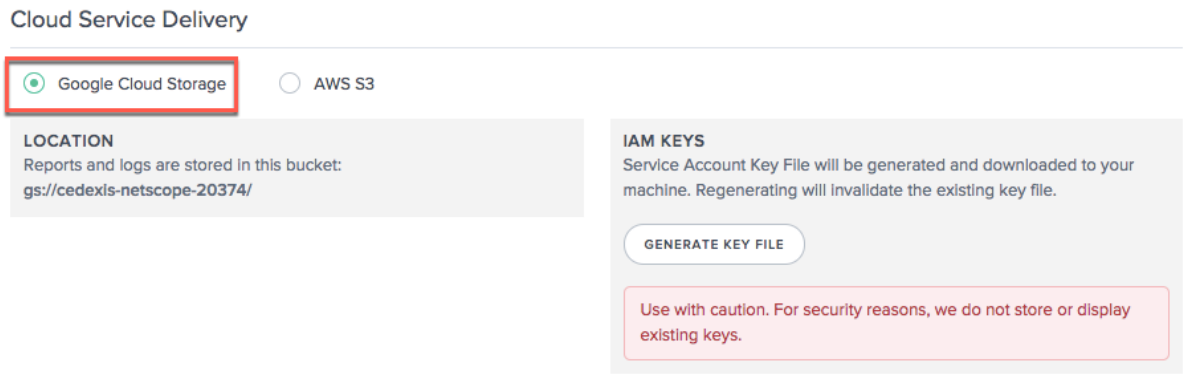
The Location represents the bucket in Google Cloud Storage where logs and reports are saved.

IAM Keys

When you select the **Generate Key File** button, the Google Service Account Key File is downloaded to your machine.

> **NOTE:**
>
> This key file serves as the only copy of the private key. Take note of your service account's email address and securely store the service account's private key file. Regenerating a new key file invalidates the existing file.

This Key File can be used with the gsutil tool to download logs and reports from the GCS bucket. For details on how to use the Key File to download log files, refer to the FAQ.



Radar Log Descriptions and Reports for Service Providers and Enterprises

Radar Logs for Providers

- These logs provide Radar measurements for benchmark partners.
- They provide every measurement taken for public platforms, regardless of the page that generated the measurement.
- Radar logs include a subset of the fields available in the raw logs, with some data anonymized: client IP /28, Referer MD5 hashed.
- Here is a sample Platform Radar Log Share in TSV file format.

> **NOTE:**
>
> - NEM never exposes full client IPs. Instead, it exposes the /28. For example, an IP of 255.255.255.255 is shown in a report as 255.255.255.240/28.
> - The client's GEO information is extracted based on the client's IPv4 which is more detailed.

Log Descriptions

The following are the columns headers and descriptions for the Radar Logs. The fields appear in the following order in the output files:

Log	Description
Timestamp	It is the UTC time of the request in YYYY-MM-DDTHH:MI:SSZ format. The actual value (down to the second) in the log tables is rounded to the nearest hour (2018-03-30T23:00:00Z) or day (2018-03-30T00:00:00Z) in the hour/day tables, respectively. Timestamp is always in UTC in all datasets.
Unique Node ID	Also known as cache node ID. It's an arbitrary value. Typically, an IP that the CDN Edge Servers return to help CDNs internally identify which server handled a particular request." (empty string): Comes from Radar clients that do not support UNI detection.0: The user agent does not support the features needed for UNI detection.1: The client encountered an error during UNI detection, such an HTTP 404 or other unsuccessful response.2: UNI detection was attempted but resulted in an error.
Provider ID	Internal ID of the platform that is being measured.
Probe Type	The probe type being measured (e.g.1: HTTP Connect Time, 0: HTTP Response Time, 14: HTTP Throughput, and so on). To indicate that the service is available, use the information returned successfully within the allowed time.
Response Code	Result of the measurement.e.g.0: success, 1: timeout, 4: error. For availability calculations the percent of measurements is taken with a 0 (success) response versus the overall number of measurements (total, regardless of response). For other probe types (RTT and throughput), the filter must only consider RTT data points with a 0 success code when calculating statistics on the RTT. Same for throughput.

Log	Description
Measurement Value	The recorded measurement value, the meaning of which varies by probe type. It represents availability (1)/ Response Time (0) measurements in milliseconds, and Throughput (14) in kbps.
Resolver Market	The market of the DNS resolver that handled the request. Generally the continent where the DNS resolver is located, where, 0: Unknown (XX), 1:North America (NA) 5: Africa (AF), 3: Europe (EU), 4: Asia (AS), 2: Oceania (OC), 6: South America (SA).
Resolver Country	The country of the DNS resolver that handled the request.IDs can be mapped to names at https://community-radar.citrix.com/ref/countries.json.gz
Resolver Region	The region of the DNS resolver that handled the request.IDs can be mapped to names at https://community-radar.citrix.com/ref/regions.json.gz **Note:** Not all the countries of the world have defined regions.
Resolver State	The state of the DNS resolver that handled the request.IDs can be mapped to names at https://community-radar.citrix.com/ref/states.json.gz Note: Not all the countries of the world have defined states.
Resolver City	The city of the DNS resolver that handled the request.Resolver city is added by looking up a resolver IP address.IDs can be mapped to names at https://community-radar.citrix.com/ref/cities.json.gz

Log	Description
Resolver ASN	The Autonomous System Number (ASN) of the DNS resolver that handled the request. Generally the ASN that has the DNS resolver IDs can be mapped to names at https://community-radar.citrix.com/ref/asns.json.gz
Resolver IP	The IP address of the DNS resolver from which our infrastructure received the DNS request.
Client Market	The market of the end user that generated this measurement. Generally the continent where the client IP is located; where, 0: Unknown (XX), 1:North America (NA) 5: Africa (AF), 3: Europe (EU), 4: Asia (AS), 2: Oceania (OC), 6: South America (SA).
Client Country	The country of the end user that generated this measurement.IDs can be mapped to names at https://community-radar.citrix.com/ref/countries.json.gz
Client Region	The region of the end user that generated this measurement. Generally the geographic region where the client IP is located.IDs can be mapped to names at https://community-radar.citrix.com/ref/regions.json.gz Note: Not all the countries of the world have defined regions.
Client State	The state of the end user that generated this measurement. Generally the state where the client IP is located.IDs can be mapped to names at https://community-radar.citrix.com/ref/states.json.gz Note, that not all the countries of the world have defined states.
Client City	The city of the end user that generated this measurement. Generally the city where the client IP is located.IDs can be mapped to names at https://community-radar.citrix.com/ref/cities.json.gz

Log	Description
Client ASN	The Autonomous System Number (ASN) of the end user that generated this measurement. Generally the ASN that contains the client IP.IDs can be mapped to names at https://community-radar.citrix.com/ref/asns.json.gz
Client IP	The IP of the end user that generated this measurement.
Referer Host MD5	The Referer information (Protocol, Host, and Path) comes from the Referer Header of the HTTP request to Radar. The Referer Host is MD5 hashed.
User Agent	It's the user agent string from the browser page that is hosting the tag. For example, if you use Chrome and browse a page with the Radar tag, the radar measurements in the background records the user agent from your Chrome browser. The measurements include Chrome browser, version of Chrome, information about the OS that Chrome is running on, and so on.
DNS Lookup Time (Optional)	With the Resource Timing API, the difference between the Domain Lookup End and the Domain Lookup Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as domainLookupEnd - domainLookupStart.
TCP Connect Time (Optional)	With the Resource Timing API, the difference between the Connect End and Connect Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as connectEnd - connectStart.

Log	Description
Secure Connect Time (Optional)	With the Resource Timing API, the difference between the Connect End and Secure Connection Start is calculates. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as connectEnd - secureConnectionStart.
Latency (Optional)	With the Resource Timing API, the difference between the Response Start and Request Start is calculated. It calculates when both the values aren't null and the response start-time is greater than the request start-time. It's calculated as responseStart - requestStart
Download Time (Optional)	With the Resource Timing API, the difference between the Response End and Response Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as responseEnd - responseStart.
Client Profile	This field helps to identify if the data is coming from mobile apps or browsers. It also allows us to differentiate between iOS, Android apps, and browsers. A number is used to identify each client profile. The values for this field are: null, 0, 1, 2, 3, 4. Where, null: Generally implies an older Radar client that does not support sending the client_profile value. 0: Browser; 1: iOS - Radar runner for iOS app written in Swift; 2: Android; 3: Browser on mobile version of website; 4: iOS - Radar Runner for iOS app written in Objective-C.
Client Profile Version	The client profile version tells us what version of the Radar Runner code (for iOS) or AndroidRadar SDK (for Android) was used in the mobile app. This field is intended for internal use only.

Log	Description
Device Category	All devices are categorized into one of the following: Smartphone, Tablet, PC, Smart TV, and Other. 'Other' is used as the default value if the parser is unable to determine the value for any of the fields.
Device	The type of device the user is on, for example an Apple iPhone. The user agent string detects it from the browser running on the page that is hosting the Radar tag.
Browser	The type of browser that the user is using, for example Mobile Safari UI/WKWebView 0.0.0. The user agent string detects it from the browser running on the page that is hosting the Radar tag.
OS	The operating system used. For example, iOS 11.0.3. the user agent string detects it from the browser running on the page that is hosting the Radar tag.
Reporting Client IP	This IP is the masked /48 public IP of the user making the measurement. It can be either IPv4 or IPv6 (when supported).

Anonymous Best Report

- Anonymous best reports help providers compare their performance to the other platform's peer group that is, within the same country, region, or ASN.
- The performance data of the top 15 providers in the peer group is aggregated based on the same categories. The best is listed next to the specific provider's best value.
- Anonymous Best Report for SSL platforms is available so that their performance can be compared with other SSL platforms.
- The client IPs are truncated to /28.
- The results of the "best" provider helps clouds/CDNs focus performance efforts on high volume or business-critical ASNs that are competitively weak to their peers.
- The report provides details on performance consists of DNS resolver IP, Client IP /28, and the caching Node that served the objects. It's compared with the "best" platform for the same criteria.

- Available for RTT or Throughput.
- The following is a sample Platform Anonymous Best Report for RTT in TSV file format.

Log Descriptions

The following are the columns headers and descriptions for the Anonymous Best Report. The fields appear in the following order in the output files.

Log	Description
Resolver Country	The country of the DNS resolver that handled the request.
Resolver Region	The region of the DNS resolver that handled the request.
Resolver State	The state of the DNS resolver that handled the request.
Resolver ASN ID	The Autonomous System Number of the DNS resolver that handled the request. Generally the ASN that has the DNS resolver.
Resolver ASN Name	The name of the ASN.
Resolver IP	The IP address of the DNS resolver from which our infrastructure received the DNS request.
Client Country	The country of the end user that generated this measurement.
Client Region	The region of the end user that generated this measurement.
Client State	The state of the end user that generated this measurement.
Client ASN ID	The Autonomous System Number (ASN) number of the end user that generated this measurement. Generally the ASN that has the client IP.
Client ASN Name	The name of the ASN of the end user that generated the measurement.
Client IP	The IP of the end user that generated the measurement.
Successes	Total number of measurements that were successful.Tip: Success / Total == Availability.

Log	Description
Timeouts	The number of measurements that timed out.
Errors	The number of measurements that were errors.
Total	The total number of measurements.
Mean	The average of all the measurements values for that row.
Best Mean	The best mean out of the top 15 providers in the peer group.
Best Mean Measurements	Total number of measurements that produced the best mean count.
Median	The 50th percentile value is the middle value of the measurements for a particular provider, when the measurements are listed in order.
Best Median	The best 50th percentile value (below which 50 percent of the measurements is found) of the top 15 providers in the peer group.
Best Median Measurements	Total number of measurements that produced the best_median
5th	The 5th percentile value for the provider.
Best 5th	The best 5th percentile value out of the top 15 providers in the peer group.
Best 5th Measurements	Total number of measurements that produced the best_5th
10th	The 10th percentile value for the provider.
Best 10th	The best 10th percentile value out of the top 15 providers in the peer group.
Best 10th Measurements	Total number of measurements that produced the best_10th
90th	The 90th percentile value for the provider.
Best 90th	The best 90th percentile value out of the top 15 providers in the peer group.
Best 90th Measurements	Total number of measurements that produced the best_90th
95th	The 95th percentile value for the provider.

Log	Description
Best 95th	The best 95th percentile value out of the top 15 providers in the peer group.
Best 95th Measurements	Total number of measurements that produced the best_95th
Stdev	The standard deviation for the provider
Best Stdev	The best standard deviation out of the top 15 providers in the peer group.
Best Stdev Measurements	Total number of measurements that produced the best std.dev.
Availability	The availability in percentage for the provider. Availability is the probe Success rate i.e.Successes / (Successes + Fails + Timeouts)
Best Availability	The best availability value out of the top 15 providers in the peer group.
Best Availability Measurements	The number of measurements that produced the best availability
Importance	Synthetic values generated to help find actionable data.
Unique Node IDs	These IDs are a comma-separated list of the Unique Node IDs for the measurements of that row.
Measurement Type	The recorded measurement value, the meaning of which varies by probe type. It's HTTP_COLD (Availability), HTTP_RTT (Round Trip Time), or HTTP_KBPS (Throughput).
Provider ID	The internal Citrix ID number for that provider.

Cache Node ID Report (previously Multi-Service Provider Report)

This report is used to identify the specific server or data center that responded to a request and help diagnose server issues.

- It provides the ID of the data center or machine that responded to a specific request.
- It helps to understand why the performance through a specific node (POP or machine, or node ID), was good or bad.

- The performance consists of response time, throughput, availability (probe type), the DNS resolver IP, Client IP /28, and the caching Node that served the objects.
- The following is a sample Platform Cache Node ID Report in TSV file format.

Log Descriptions

The following are the columns headers and descriptions for the Cache Node ID Report. The fields appear in the following order in the output files:

Log	Description
Provider Name	It's the name of the provider that is being measured.
Measurement Value	The recorded measurement value, the meaning of which varies by probe type. It's connect (1)/RTT (0) measurements in milliseconds, and throughput (14) measurements in kbps.
Unique Node ID	It's known as cache node ID. An arbitrary value, typically an IP that CDN Edge Servers return to help CDNs internally identify which server handled a particular request." (empty string): Comes from Radar clients that do not support UNI detection.0: The user agent does not support the features needed for UNI detection.1: The client finds an error during UNI detection, such an HTTP 404 or other unsuccessful response.2: UNI detection was attempted but resulted in an error.
Resolver Country	The country of the DNS resolver that handled the request.
Resolver Region	The region of the DNS resolver that handled the request.
Resolver State	The state of the DNS resolver that handled the request.
Resolver ASN	The Autonomous System Number of the DNS resolver that handled the request. Generally the ASN that has the DNS resolver.
Resolver ASN Name	The name of the ASN.

Log	Description
Resolver IP	The IP address of the DNS resolver from which our infrastructure received the DNS request.
Client Country	The country of the end user that generated this measurement.
Client Region	The region of the end user that generated this measurement.
Client State	The state of the end user that generated this measurement.
Client ASN	The Autonomous System Number (ASN) number of the end user that generated this measurement. Generally the ASN that has the client IP.
Client ASN Name	The name of the ASN of the end user that generated the measurement.
Client IP	The IP of the end user that generated the measurement.
Success	Total number of measurements that were successful.Tip: Success / Total == Availability.
Timeout	The number of measurements that timed out.
Error	The number of measurements that were errors.
Total	The total number of measurements.
Mean	The average of the measurement values for each row.
Median	The 50th percentile value is the middle value of the measurements for a particular provider, when the measurements are listed in order.
5th	The 5th percentile value for the provider.
10th	The 10th percentile value for the provider.
90th	The 90th percentile value for the provider.
95th	The 95th percentile value for the provider.
Stdev	The standard deviation for the provider.
Availability	The availability in percentage for the provider.

Log	Description
Importance	Synthetic values generated to help find actionable data.

Hourly by Country/ASN Report

- This report helps to verify if your providers' performance varies significantly during a day.
- It shows the time when the measurements were taken truncated down to the hour, for example `2018-03-11T23:00:00`.
- The following is a sample Platform Hourly by Country/ASN Report in TSV file format.

Log Descriptions

The following are the columns headers and descriptions for the Hourly by Country/ASN Report. The fields appear in the following order in the output files:

Log	Description
Timestamp 60 minutes	The UTC time when the measurements were taken truncated down to the hour, for example2018-03-11T23:00:00.
Provider Name	It's the name of the provider that is being measured.
Measurement Type	The recorded measurement value, the meaning of which varies by probe type. It's HTTP_COLD (Availability), HTTP_RTT (Round Trip Time), or HTTP_KBPS (Throughput).
Client Country	The country of the end user that generated this measurement.
Client ASN	The Autonomous System Number (ASN) number of the end user that generated this measurement. Generally the ASN that has the client IP.
Client ASN Name	The name of the ASN of the end user that generated the measurement.
Success	Total number of measurements that were successful.Tip: Success / Total == Availability.

Log	Description
Timeout	The number of measurements that timed out.
Error	The number of measurements that were errors.
Total	The total number of measurements.
Mean	The average of the measurement values for each row.
Median	The 50th percentile value is the middle value of the measurements for a particular provider, when the measurements are listed in order.
5th	The 5th percentile value for the provider.
10th	The 10th percentile value for the provider.
90th	The 90th percentile value for the provider.
95th	The 95th percentile value for the provider.
Stdev	The standard deviation for the provider.
Availability	The availability in percentage for the provider.
Importance	Synthetic value generated to help find actionable data.
Provider ID	The internal Citrix ID number for that provider.

Radar Log Descriptions and Reports for ISPs

Radar Logs for ISPs

Radar logs enable ISPs to measure their performance against global platforms in detail. ISPs can use this data to find areas where improvements must be made or to verify the expected performance.

- Provides access to Radar measurements.
- Provides measurements taken from ISPs on public platforms, regardless of the page that generated the measurement.
- Radar logs include a subset of the fields available in the raw logs, with some data anonymized: client IP /28, referer MD5 hashed.
- The log files are in TSV format.
- The following is a sample Network Radar Log Share in TSV file format.

Log Descriptions

The following are the columns headers and descriptions for the Radar logs for ISPs. The fields appear in the following order in the output files.

Log	Description
Timestamp	It's the UTC time of the request in YYYY-MM-DDTHH:MI:SSZ format. The actual value (down to the second) in the log tables is rounded to the nearest hour (2018-03-30T23:00:00Z) or day (2018-03-30T00:00:00Z) in the hour/day tables, respectively. The timestamp is always in UTC in all datasets.
Provider ID	Internal ID of the platform that is being measured.
Probe Type	The probe type being measured (e.g.1: HTTP Connect Time, 0: HTTP Response Time, 14: HTTP Throughput, and so on). The information that it returned successfully within the allowed time is used to indicate that the service is available.
Response Code	Result of the measurement.e.g.0: success, 1: timeout, 4: error. For availability calculations the percent of measurements is taken with a 0 (success) response versus The overall number of measurements (total). For other probe types (RTT and throughput), the filter must only consider RTT data points with a 0 success code when calculating statistics on the RTT. Same for throughput.
Measurement Value	The recorded measurement value, the meaning of which varies by probe type. It's availability (1)/ Response Time (0) measurements in milliseconds, and Throughput (14) in kbps.

Log	Description
Resolver Market	The market of the DNS resolver that handled the request. Generally the continent where the DNS resolver is located, where, 0: Unknown (XX), 1:North America (NA) 5: Africa (AF), 3: Europe (EU), 4: Asia (AS), 2: Oceania (OC), 6: South America (SA).
Resolver Country	The country of the DNS resolver that handled the request IDs can be mapped to names at https://community-radar.citrix.com/ref/countries.json.gz
Resolver Region	The region of the DNS resolver that handled the request IDs can be mapped to names at https://community-radar.citrix.com/ref/regions.json.gz. Not all the countries of the world have defined regions.
Resolver State	The state of the DNS resolver that handled the request IDs can be mapped to names at https://community-radar.citrix.com/ref/states.json.gz. Not all the countries of the world have defined states.
Resolver ASN	The Autonomous System Number (ASN) of the DNS resolver that handled the request. Generally the ASN that has the DNS resolver IDs can be mapped to names at https://community-radar.citrix.com/ref/asns.json.gz.
Resolver IP	The IP address of the DNS resolver from which our infrastructure received the DNS request.
Client Market	The market of the end user that generated this measurement. Generally the continent where the client IP is located; where, 0: Unknown (XX), 1:North America (NA) 5: Africa (AF), 3: Europe (EU), 4: Asia (AS), 2: Oceania (OC), 6: South America (SA).

Log	Description
Client Country	The country of the end user that generated this measurement.IDs can be mapped to names at https://community-radar.citrix.com/ref/countries.json.gz
Client Region	The region of the end user that generated this measurement. Generally the geographic region where the client IP is located. IDs can be mapped to names at https://community-radar.citrix.com/ref/regions.json.gz. Not all the countries of the world have defined regions.
Client State	The state of the end user that generated this measurement. Generally the state where the client IP is located. IDs can be mapped to names at https://community-radar.citrix.com/ref/states.json.gz. Not all the countries of the world have defined states.
Client ASN	The Autonomous System Number (ASN) of the end user that generated this measurement. Generally the ASN that has the client IP. IDs can be mapped to names at https://community-radar.citrix.com/ref/asns.json.gz
Client IP	The IP of the end user that generated this measurement.
Referer Host MD5	The Referer information (Protocol, Host, and Path) comes from the Referer Header of the HTTP request to Radar. The Referer Host is MD5 hashed.
User Agent	It's the user agent string from the browser page that is hosting the tag. For example, if you use Chrome and browse a page with the Radar tag, the radar measurements in the background records the user agent from your Chrome browser. The measurements include Chrome browser, version of Chrome, information about the OS that Chrome is running on, and so on.

Log	Description
DNS Lookup Time (Optional)	With the Resource Timing API, the difference between the Domain Lookup End and the Domain Lookup Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as domainLookupEnd - domainLookupStart.
TCP Connect Time (Optional)	With the Resource Timing API, the difference between the Connect End and Connect Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as connectEnd - connectStart.
Secure Connect Time (Optional)	With the Resource Timing API, the difference between the Connect End and Secure Connection Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as connectEnd - secureConnectionStart.
Latency (Optional)	With the Resource Timing API, the difference between the Response Start and Request Start is calculated. It calculates when both values aren't null and the response start time is greater than the request start time. It's calculated as responseStart - requestStart
Download Time (Optional)	With the Resource Timing API, the difference between the Response End and Response Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as responseEnd - responseStart.

Log	Description
Client Profile	This field helps to identify if the data is coming from mobile apps or browsers. It also allows us to differentiate between iOS, Android apps, and browsers. A number is used to identify each client profile. The values for this field are: null, 0, 1, 2, 3, 4. Where, null: Generally implies an older Radar client that does not support sending the client_profile value. 0: Browser; 1: iOS - Radar runner for iOS app written in Swift; 2: Android; 3: Browser on mobile version of website; 4: iOS - Radar Runner for iOS app written in Objective-C.
Client Profile Version	The client profile version tells us what version of the Radar Runner code (for iOS) or AndroidRadar SDK (for Android) was used in the mobile app. This field is intended for internal use only.
Device Category	All devices are categorized into one of the following: Smartphone, Tablet, PC, Smart TV, and Other. 'Other' is used as the default value if the parser is unable to determine the value for any of the fields.
Device	The type of device the user is on, for example an Apple iPhone. The user agent string detects it from the browser running on the page that is hosting the Radar tag.
Browser	The type of browser that the user is using, for example Mobile Safari UI/WKWebView 0.0.0. The user agent string detects it from the browser running on the page that is hosting the Radar tag.
OS	The operating system that is being used, for example iOS 11.0.3. The user agent string detects it from the browser running on the page that is hosting the Radar tag.

Subnet Report for ISPs

- The report provides ISPs with information on how the specific subnets of their networks perform for their users through the measured platforms.
- It provides information about the service provider that responded to a specific request.
- It helps to understand the performance by the network subnet.
- The performance consists of response time, throughput, availability (probe type), the DNS resolver IP, Client IP /28, and the caching Node that served the objects.
- The following is a sample Network Subnet Report in TSV file format.

Log Descriptions

The following are the columns headers and descriptions for the Subnet Report for ISPs. The fields appear in the following order in the output files:

Log	Description
ASN Name	The name of the Autonomous System from where the measurement was taken.
Measurement Value	The recorded measurement value, the meaning of which varies by probe type. It's connect (1)/RTT (0) measurements in milliseconds, and throughput (14) measurements in kbps.
Subnet	The subnet of the user from where the request originated.
Resolver ASN	The Autonomous System Number of the DNS resolver that handled the request. Generally the ASN that has the DNS resolver.
Resolver IP	The IP address of the DNS resolver from which our infrastructure received the DNS request.
Client ASN	The Autonomous System Number (ASN) number of the end user that generated this measurement. Generally the ASN that has the client IP.
Client IP	The IP of the end user that generated the measurement.
Platform ID	The ID of the Service Provider platform to which the query was done.

Log	Description
Platform Name	The name of the Service Provider platform to which the query was done
Success	Total number of measurements that were successful.Tip: Success / Total == Availability.
Timeout	The number of measurements that timed out.
Error	The number of measurements that were errors.
Total	The total number of measurements.
Mean	The average of the measurement values for each row.
Median	The 50th percentile value is the middle value of the measurements for a particular provider, when the measurements are listed in order.
5th	The 5th percentile value for the provider.
10th	The 10th percentile value for the provider.
90th	The 90th percentile value for the provider.
95th	The 95th percentile value for the provider.
Stdev	The standard deviation for the provider.
Availability	The availability in percentage for the provider.
Importance	Synthetic values generated to help find actionable data.
Measurement Type	The recorded measurement value, the meaning of which varies by probe type. It's HTTP_COLD (Availability), HTTP_RTT (Round Trip Time), or HTTP_KBPS (Throughput).

Anonymous Best Report for ISPs

- In the Anonymous Best report, a peer group is used for the "best" comparison. The peer group is based on the location of the ISP. It's usually the 10 most measured ISPs in a specified country, with a minimum of over 1,000 sessions.
- The results of the "best" ISP helps ISPs focus performance efforts on high volume or business-critical platforms and areas that are competitively weak to their peers.
- The report provides details on performance broken down by geography and Platform, and com-

pares it with the “best” ISP for the same criteria.
- Available for RTT and Throughput.
- The following is a sample Network Anonymous Best Report for RTT in TSV file format.

Log Descriptions

The following are the columns headers and descriptions for the Anonymous Best Report. The fields appear in the following order in the output files.

Log	Description
Measurement Type	The recorded measurement value, the meaning of which varies by probe type. It’s HTTP_COLD (Availability), HTTP_RTT (Round Trip Time), or HTTP_KBPS (Throughput).
Client Country	The country of the end user that generated this measurement.
Client Region	The region of the end user that generated this measurement.
Client State	The state of the end user that generated this measurement.
Client ASN ID	The Autonomous System Number (ASN) number of the end user that generated this measurement. Generally the ASN that has the client IP.
Client ASN Name	The name of the ASN of the end user that generated the measurement.
Resolver Country	The country of the DNS resolver that handled the request.
Resolver Region	The region of the DNS resolver that handled the request.
Resolver State	The state of the DNS resolver that handled the request.
Platform ID	The ID of the Service Provider platform to which the query was attempted.
Platform Name	The name of the Service Provider platform to which the query was attempted.

Log	Description
Successes	Total number of measurements that were successful.Tip: Success / Total == Availability.
Timeouts	The number of measurements that timed out.
Errors	The number of measurements that were errors.
Total	The total number of measurements.
Mean	The average of all the measurements values for that row.
Best Mean	The best mean out of the top 15 providers in the peer group.
Best Mean Measurements	Total number of measurements that produced the best mean count.
Median	The 50th percentile value is the middle value of the measurements for a particular provider, when the measurements are listed in order.
Best Median	The best 50th percentile value (below which 50 percent of the measurements is found) of the top 15 providers in the peer group.
Best Median Measurements	Total number of measurements that produced the best_median
5th	The 5th percentile value for the provider.
Best 5th	The best 5th percentile value out of the top 15 providers in the peer group.
Best 5th Measurements	Total number of measurements that produced the best_5th
10th	The 10th percentile value for the provider.
Best 10th	The best 10th percentile value out of the top 15 providers in the peer group.
Best 10th Measurements	Total number of measurements that produced the best_10th
90th	The 90th percentile value for the provider.
Best 90th	The best 90th percentile value out of the top 15 providers in the peer group.

Log	Description
Best 90th Measurements	Total number of measurements that produced the best_90th
95th	The 95th percentile value for the provider.
Best 95th	The best 95th percentile value out of the top 15 providers in the peer group.
Best 95th Measurements	Total number of measurements that produced the best_95th
Stdev	The standard deviation for the provider.
Best Stdev	The best standard deviation out of the top 15 providers in the peer group.
Best Stdev Measurements	Total number of measurements that produced the best std.dev.
Availability	The availability in percentage for the provider. Availability is the probe Success rate that is, Successes / (Successes + Fails + Timeouts)
Best Availability	The best availability value out of the top 15 providers in the peer group.
Best Availability Measurements	The number of measurements that produced the best availability.
Importance	Synthetic values generated to help find actionable data.

Navigation Timing Log Descriptions

Navigation Timing Data

Navigation Timing data provides insights into the various parts of the page load process for a webpage.

This data varies because of the end user's location, network issues, changes made by the provider, and so on. Customers can use Navigation Timing data to optimize the end user's experience in loading the monitored webpage.

Measurements can be taken for every Radar session (if enabled). Each session is attached to an ID number that helps track all measurements from a session. These measurements are shared with customers as Navigation Timing Logs via NEM.

The following is a sample of the Navigation Timing Data in TSV file format.

The following are the columns headers and descriptions for Navigation Timing logs. The fields appear in the following order in the output files:

Log	Description
Timestamp	It's the UTC time of the request in YYYY-MM-DDTHH:MI:SSZ format. The actual value (down to the second) in the log tables is rounded to the nearest hour (2018-03-30T23:00:00Z) or day (2018-03-30T00:00:00Z) in the hour/day tables, respectively. It's always in UTC in all datasets.
Response Code	Result of the measurement.e.g.0: success, 1: timeout, 4: error. For availability calculations the percent of measurements is taken with a 0 (success) response versus the overall number of measurements (total). For other probe types (RTT and throughput), the filter is to only consider RTT data points with a 0 success code when calculating statistics on the RTT. Same for throughput.
Resolver Market	The market of the DNS resolver that handled the request. Generally the continent where the DNS resolver is located, where, 0: Unknown (XX), 1:North America (NA) 5: Africa (AF), 3: Europe (EU), 4: Asia (AS), 2: Oceania (OC), 6: South America (SA).
Resolver Country	The country of the DNS resolver that handled the request.IDs can be mapped to names at https://community-radar.citrix.com/ref/countries.json.gz
Resolver Region	The region of the DNS resolver that handled the request.IDs can be mapped to names at https://community-radar.citrix.com/ref/regions.json.gz. Not all the countries of the world have defined regions.

Log	Description
Resolver State	The state of the DNS resolver that handled the request.IDs can be mapped to names at https://community-radar.citrix.com/ref/states.json.gz. Not all the countries of the world have defined states.
Resolver ASN	The Autonomous System Number (ASN) of the DNS resolver that handled the request. Generally the ASN that has the DNS resolver. IDs can be mapped to names at https://community-radar.citrix.com/ref/asns.json.gz
Resolver IP	The IP address of the DNS resolver from which our infrastructure received the DNS request.
Client Market	The market of the end user that generated this measurement. Generally the continent where the client IP is located; where, 0: Unknown (XX), 1:North America (NA) 5: Africa (AF), 3: Europe (EU), 4: Asia (AS), 2: Oceania (OC), 6: South America (SA).
Client Country	The country of the end user that generated this measurement.IDs can be mapped to names at https://community-radar.citrix.com/ref/countries.json.gz
Client Region	The region of the end user that generated this measurement. Generally the geographic region where the client IP is located. IDs can be mapped to names at https://community-radar.citrix.com/ref/regions.json.gz. Not all the countries of the world have defined regions.
Client State	The state of the end user that generated this measurement. Generally the state where the client IP is located. IDs can be mapped to names at https://community-radar.citrix.com/ref/states.json.gz. Not all the countries of the world have defined states.

Log	Description
Client ASN	The Autonomous System Number (ASN) of the end user that generated this measurement. Generally the ASN that has the client IP. IDs can be mapped to names at https://community-radar.citrix.com/ref/asns.json.gz
Client IP	The IP of the end user that generated the measurement.
Referer Host	The Referer information (Protocol, Host, and Path) comes from the Referer Header of the HTTP request to Radar.
Referer Protocol	The Referer information (Protocol, Host, and Path) comes from the Referer Header of the HTTP request to Radar.
Referer Path	The Referer information (Protocol, Host, and Path) comes from the Referer Header of the HTTP request to Radar.
Device Category	All devices are categorized into one of the following: Smartphone, Tablet, PC, Smart TV, and Other. 'Other' is used as the default value if the parser is unable to determine the value for any of the fields.
Device	The type of device the user is on, for example an Apple iPhone. The user agent string detects it from the browser running on the page that is hosting the Radar tag.
Browser	The type of browser that the user is using, for example Mobile Safari UI/WKWebView 0.0.0. The user agent string detects it from the browser running on the page that is hosting the Radar tag.
OS	The operating system that is being used, for example iOS 11.0.3. The user agent string detects it from the browser running on the page that is hosting the Radar tag.

Log	Description
DNS Lookup Time	With the Resource Timing API, the difference between the Domain Lookup End and the Domain Lookup Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as domainLookupEnd - domainLookupStart.
TCP Connect Time	With the Resource Timing API, the difference between the Connect End and Connect Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as connectEnd - connectStart.
Secure Connect Time	With the Resource Timing API, the difference between the Connect End and Secure Connection Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as connectEnd - secureConnectionStart.
Load Event	It's the duration or time taken to go from the start to end of the load event. It's calculated as LoadEventEnd - LoadEventStart, when both values aren't null and the end time is greater than the start time.
Redirect	It's the duration or time taken to go from Navigation Start to Fetch Start. It's calculated as FetchStart - NavigationStart, when both values are not null and the end time is greater than the start time.
Total Page Load	It's the duration or time taken to go from the start of navigation to the end of the page-load event. It's calculated as - Load Event End - Navigation Start when both values aren't null and the end time is greater than the start time.

Log	Description
DOM	The duration or time taken to go from DOM loading to DOM complete. It's calculated as DomComplete - DomLoading when both values aren't null and the end time is greater than the start time.
Latency	With the Resource Timing API, the difference between the Response Start and Request Start is calculated. It calculates when both values aren't null and the response start time is greater than the request start time. It's calculated as responseStart - requestStart
Download Time	With the Resource Timing API, the difference between the Response End and Response Start is calculated. It calculates when both values aren't null and the end time is greater than the start time. It's calculated as responseEnd - responseStart.
DOM interactive	The duration or time taken to go from Navigation Start to DOM Interactive. It's calculated as DomInteractive - NavigationStart when both values aren't null and the end time is greater than the start time.
Start Render	The duration or time taken to go from Navigation Start to Start Render. It's calculated as startRender - NavigationStart when both values aren't null and the end time is greater than the start time.

Openmix and HTTP Openmix Logs

Openmix and HTTP Openmix logs allow customers to use real-time measurements to monitor the behavior of their Openmix apps. They can use this data to find areas of improvements or to verify the expected performance of their apps.

- These logs provide real-time measurements taken for Openmix customers.
- The recommended file format for these logs is JSON, but they're available in TSV format as well.
- Here are samples of Openmix and HTTP Openmix log sharing data in TSV file format.

Openmix Log Descriptions

Log	Description
Timestamp	It's the UTC time of the request in YYYY-MM-DDTHH:MI:SSZ format. The actual value (down to the second) in the log tables is rounded to the nearest hour (2018-03-30T23:00:00Z) or day (2018-03-30T00:00:00Z) in the hour/day tables, respectively. The timestamp is always in UTC in all datasets.
App Owner Zone ID	The zone ID for the application owner servicing the request. This value is always equal to 1.
App Owner Customer ID	The customer ID for the application owner who services the request. For HTTP requests, code this ID in the request path and use it to look up which application to run.
App ID	The application ID within the customer's account that services the request. This ID is also coded in the HTTP request path. Application IDs start at 1 and are only unique to the customer. You must fully qualify queries for a specific app ID by querying on the appOwnerCustomerId.
App Version	The version of the application that serviced the account. Each time an application is updated via the portal or the API, the version is incremented. The version that was running at the time of the request is recorded. This information can be used to separate versioned logic over time as applications are updated. Hosts throughout the network generally receive updates in a similar timeframe, but almost never at precisely the same moment. It's likely that overlapping decisions in time uses different versions of an app during the process of update.

Log	Description
App Name	The name of the application that serviced the account.
Market	The market of the end user that generated this measurement.
Country	The country of the end user that generated this measurement.
Region	The region of the end user that generated this measurement.
State	The state of the end user that generated this measurement.
ASN ID	The Autonomous System Number (ASN) of the end user that generated this measurement. Generally the Autonomous System Number that has the client IP.
ASN Name	The name of the ASN of the end user that generated the measurement.
Effective IP	The effective IP is the IP used to process the request. It's the query string-specified IP overriding the requesting IP (versus The resolver/ECS/EDNS ID for the DNS flow). It's the address that the system considers the target when processing the information. This IP is either the IP of the requesting resolver, or the ECS IP address of the client if EDNS ECS is supported. So all probe performance data, geographic information, etc.passed to the application logic is based on this IP.
Resolver Market	The market of the DNS resolver that handled the request.
Resolver Country	The country of the DNS resolver that handled the request.
Resolver Region	The region of the DNS resolver that handled the request.
Resolver State	The state of the DNS resolver that handled the request.

Log	Description
Resolver ASN ID	The Autonomous System Number (ASN) of the DNS resolver that handled the request. Generally the Autonomous System Number that has the DNS resolver.
Resolver ASN Name	The name of the ASN of the resolver that handled the request.
Resolver IP	The IP address of the DNS resolver from which our infrastructure received the DNS request.
Decision Provider Name	Alias of the platform that an application selects.
Reason Code	Reason Code set within the application describing the reason behind the decision.
Reason Log	This log is a customer-defined output from the Openmix app. It's an optional string field that enables customers to log information about their Openmix app decisions.
Fallback Mode	This mode indicates whether the app was in fallback mode when it handled the request. Fallback happens when something failed during the preparation of the request for execution.
Used EDNS	True if the application uses an EDNS Client Subnet extension.
TTL	The TTL (Time To Live) that was handed back.
Response	The CNAME returned from the request.
Result	The value in this field is always 1.
Context	It's the summary of the Radar data that was available to Openmix when the request was handled. Openmix resolves Radar data relative to the effective values for every request, so two clients making requests at the same time can have different context strings.

Openmix HTTP API Log Descriptions

Log	Description
Timestamp	It's the UTC time of the request in YYYY-MM-DDTHH:MI:SSZ format. The actual value (down to the second) in the log tables is rounded to the nearest hour (2018-03-30T23:00:00Z) or day (2018-03-30T00:00:00Z) in the hour/day tables, respectively. The timestamp is always in UTC in all datasets.
App Owner Zone ID	The zone ID for the application owner servicing the request. This value is always equal to 1.
App Owner Customer ID	The customer ID for the application owner who services the request. For HTTP requests, code this ID in the request path and is used to look up which application to run.
App ID	The application ID within the customer's account who services the request. This ID is also coded in the HTTP request path. Application IDs start at 1 and are only unique to the customer. You must fully qualify queries for a specific app ID by querying on the appOwnerCustomerId.
App Version	The version of the application that serviced the account. Each time an application is updated via the portal or the API, the version is incremented. The version that was running at the time of the request is recorded. This information can be used to separate versioned logic over time as applications are updated. Hosts throughout the network generally receive updates in a similar timeframe, but almost never at precisely the same moment. It's likely that overlapping decisions in time uses different versions of an app during the process of update.
App Name	The name of the application that serviced the account.

Log	Description
Market	The market of the end user that generated this measurement.
Country	The country of the end user that generated this measurement.
Region	The region of the end user that generated this measurement.
State	The state of the end user that generated this measurement.
ASN ID	The ID of the Autonomous System Number (ASN) of the end user that generated this measurement that is, the network ID number associated with the ASN Name
ASN Name	The name of the ASN of the end user that generated the measurement.
Effective IP	The effective IP is the IP used to process the request. It's the query string-specified IP overriding the requesting IP (versus The resolver/ECS/EDNS ID for the DNS flow). It's the address that the system considers the target when processing the information. This IP is either the IP of the requesting resolver, or the ECS IP address of the client if EDNS ECS is supported. All probe-performance data, geographic information, and so on, passed to the application logic is based on this IP.
Decision Provider Name	Alias of the platform that an application selects.
Reason Code	Reason Code set within the application describing the reason behind the decision.
Reason Log	This log is a customer-defined output from the Openmix app. It's an optional string field that enables customers to log information about their Openmix app decisions.

Log	Description
Fallback Mode	This mode indicates whether the app was in fallback mode when it handled the request. Fallback happens when something failed during the preparation of the request for execution.
Response Code	Result of the measurement.e.g.0: success, 1: timeout, 4: error. For availability calculations the percent of measurements is taken with a 0 (success) response versus The overall number of measurements (total, regardless of response). For other probe types (RTT and throughput), the filter must only consider RTT data points with a 0 success code when calculating statistics on the RTT. Same for throughput.
HTTP Method	The HTTP method (GET/POST/OPTIONS/etc) relates to the request that was made to the HTTP Openmix server from a customer service. Together these methods make up portions of the URL inbound and the HTTP responses outbound.
URI	It's the request path. If customers aren't getting the behavior they want, it can be because of an improperly structured request. The logs show what our servers are receiving (Protocol, Host, and Path). The Referer information (Protocol, Host, and Path) comes from the Referer Header of the HTTP request to Radar. For HTTP OPX the whole Referer (protocol, host, and path) is included in one string labeled Referer.

Log	Description
User Agent	It's the user agent string from the browser page that is hosting the tag. For example, if you use Chrome and browse a page with the Radar tag, the radar measurements in the background records the user agent from your Chrome browser. The measurements include Chrome browser, version of Chrome, information about the OS that Chrome is running on, and so on.
Context	It's the summary of the Radar data that was available to Openmix when the request was handled. Openmix resolves Radar data relative to the effective values for every request, so two clients making requests at the same time can have different context strings.

Custom Reports for Third Party Organizations

Customers can work with Citrix to get custom reports based on Radar data that Citrix collects. Citrix can generate reports to run on a schedule. The reports are available as data files, usually in TSV format.

FAQ

Radar

How frequently are files pushed to S3 and GCS?

The frequency of file deposits is once a minute for Radar and daily for reports.

Where are the reports stored?

S3 Legacy (Location 1):

```
s3://public-radar/[customer name]/
```

S3 (Location 2):

```
s3://cedexis-netscope/[customer id]/
```

GCS (Location 3):

```
gs://cedexis-netscope-[customer id]/
```


How to get S3 access credentials if you don't have them already?

The portal provides an 'Access' and 'Secret' key. Use the keys with 's3cmd', 'awscli' or other tools to access S3. For Google Storage, the Portal downloads a file with access credentials to use with the 'gsutil' tool.

How to use the access and secret keys with s3cmd to download logs and reports from the S3 bucket?

First you would need to download and install the `s3cmd` from https://s3tools.org/download, and refer to https://s3tools.org/usage for usage, options, and commands. Then run the following command:

```
s3cmd --access_key=[access key] --secret_key=[secret key] ls s3://
    cedexis-netscope/<customer id>/radar/
<!--NeedCopy-->
```

To download the files, run the following command:

```
s3cmd --access_key=[access_key] --secret_key=[secret_key] get s3://
    cedexis-netscope/<customer id>/radar/[the_filename_to_download] [
    the_name_of_the_local_file]
<!--NeedCopy-->
```

How to use the s3cmd config to list files in the S3 bucket

The first step is to Install `s3cmd`. You can install it from http://s3tools.org/download

To configure s3cmd, run the following command

```
s3cmd ls s3://cedexis-netscope/[customer id]/
<!--NeedCopy-->
```

If you're already using `s3cmd` with another set of access and secret keys, then follow these steps:

If you already use `s3cmd`, then make a copy of the default config, at `~/.s3cfg`. For example, make a copy and name it as `~/.s3cfg_netscope`. Replace the access and secret key entries in `~/.s3cfg_netscope` with the ones that we provide.
Use the new configuration instead of the default one (your company's) to access the S3 bucket with the following command:

```
s3cmd -c ~/.s3cfg_netscope ls s3://cedexis-netscope/[customer id]/
<!--NeedCopy-->
```

The main difference is you have to put in a `-c` and where the config file is with the Citrix-provided access and secret keys.

If you want to switch between sets of keys, embed them in a file. Refer to the file with the `-c` option to specify which key pair you're using.

> **NOTE:** `-c` parameter indicates where the config file, that contains the access and secret keys, is located.

How to use the Key File with gsutil or gcloud to download log files

Once you download the google service account JSON Key File, you can use it to authenticate your google account credentials, view, or download your log files. For example, here's one way to do that using the google `gcloud` and `gsutil` command line utilities:

Step 1: Activate the Key File

The authenticating commands `gcloud auth activate-` or `gsutil config -e` are required to authenticate the key file for running gcloud or gsutil commands.

For gcloud:

Run the following command using the downloaded Key File:

```
gcloud auth activate-service-account --key-file [downloaded config file
    ]
<!--NeedCopy-->
```

Or

```
gcloud auth activate-service-account --key-file=[path and file name of
    key file]
<!--NeedCopy-->
```

For gsutil:

Run the following command using the downloaded config file:

```
gsutil config -e
<!--NeedCopy-->
```

Step 2: List the files in the GCS (Google Cloud Storage) Bucket

Once you've activated the service account key file as described in the earlier step, use the following command to list the files in the GCS bucket:

```
gsutil ls gs://cedexis-netscope-<customer id>
<!--NeedCopy-->
```

Step 3 (if necessary): Restore original credentials (or switch back and forth between accounts)

You can switch between the Citrix account and other Google Cloud credentials you've authenticated by doing the following.

First, run the following command to list all your accounts:

```
gcloud auth list
<!--NeedCopy-->
```

Then use the following command to switch to another account:

```
gcloud config set account [email of the account to switch to as shown
    in gcloud auth list]
<!--NeedCopy-->
```

You can switch back and forth between accounts using the same command, by replacing the email with the account email that you want to switch to.

What does the file name look like?

Legacy Daily:

The Radar daily log ShareFile names have this structure:

`<prefix><date: YYYY-MM-DD>.<customer_id>.part<uniq_id>.kr.txt.gz`

For example `Cedexis_Daily-2017-11-07.21222.part-cc901e1dd55eal4e.kr.txt.gz` (non-standard example)

Legacy Real-time:

The Radar real-time log ShareFile names have this structure:

`<prefix><customer_id>-YYYY-MM-DDTHH:MM<uniq_id>.txt.gz`

For example `Cedexis_3-32291-2017-11-08T20:56-cc907e8fd71eaf4e.txt.gz`

Netscope NEM Format:

The Netscope NEM format for daily and real-time log share files have this structure:

`<freq><log_type><prefix><id_type><id><iso_dt><uniq_id>.<line_format>.gz`

Where,

- `freq`: `"daily"| "rt"| "hr"`
- `log_type`: `"radar"| "opx"| "hopx"`
- `prefix`: `log_share.prefix`
- `id_type`: `"customer"| "provider"| "asn"`
- `id`: `log_share.match_id`
- `iso_dt`: `iso 8601 Date_time "YYYYMMDDTHHMMSSZ"`

- `uniq_id`: `hash(UUID)`
- `line_format`: `"tsv"| "json"`

For example `rt-radar-TestRadar1-provider-20363-20171209183034Z-cc907e8fd71eaf4e.tsv.gz`

What is the format of the output file?

For Radar, the output file format is TSV (tab-separated value), gzipped.

Openmix and Openmix HTTP API

How frequently are files pushed to S3?

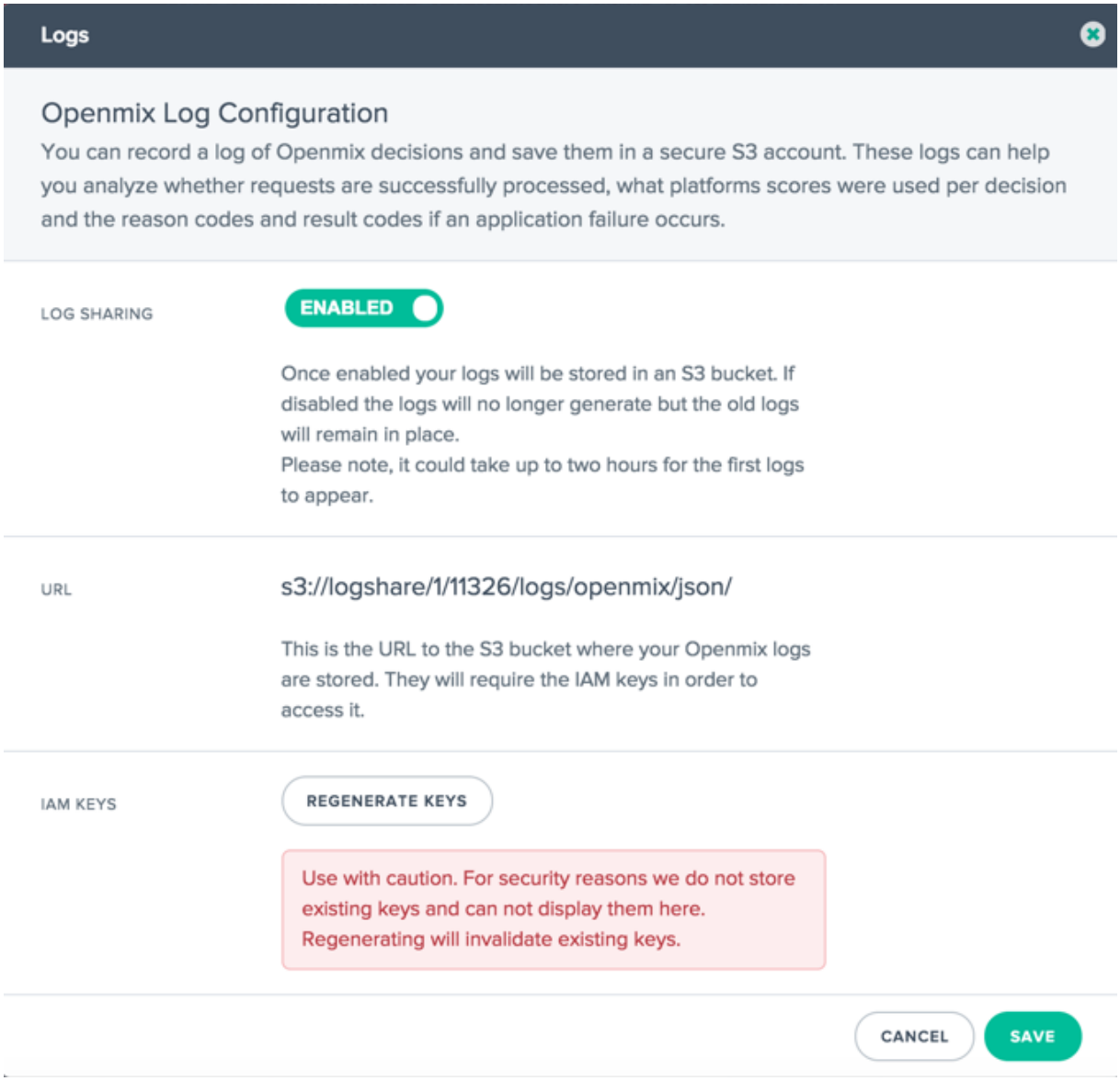
The frequency of file deposits is once a minute for Openmix and HTTP Openmix.

What if you're unable to see the option to configure Openmix and Openmix HTTP API real time log sharing?

Your Account Manager can enable the required role for you to configure and enable Openmix and Openmix HTTP API real time log sharing.

How do you turn on Openmix and an Openmix HTTP API real time log sharing and access files?

Once the role is enabled on your account, you see the **Manage Logs** icon. Click to open the **Logs** dialog where you can access Openmix Log Configuration settings. These settings are basically all you need to turn on Openmix and HTTP Openmix real time log-sharing and access files.



What is the back-end process?

Turning on Openmix log sharing enables Openmix HTTP API log sharing as well. The Openmix and Openmix HTTP API log-sharing services must start outputting logs for the customer within 10 minutes.

Where are the Openmix and HTTP Openmix reports stored?

S3 Legacy (Location 1):

```
s3://logshare/[zone ID]/[customer ID]/logs/openmix/json/[YYYY]/[MM]/[DD]/[HH]/.
```

S3 (Location 2):


```
s3://cedexis-netscope/[customer id]/
```

GCS (Location 3):

```
gs://cedexis-netscope-[customer id]/
```

What does the file name look like?

The file name structure for Openmix and HTTP Openmix typically looks like this:

Legacy Real-time:

```
[zone ID, 1][customerID]-openmix-json[YYYY][MM][DD][HH][mm][ss]Z-m1-w9-c0.
gz
```

Netscope NEM Format:

The Netscope NEM format for daily and real-time log share files have this structure:

```
<freq><log_type><prefix><id_type><id><iso_dt><uniq_id>.<line_format>.gz
```

Where,

- `freq`: `"daily"| "rt"| "hr"`
- `log_type`: `"radar"| "opx"| "hopx"`
- `prefix`: `log_share.prefix`
- `id_type`: `"customer"| "provider"| "asn"`
- `idv`: `log_share.match_id`
- `iso_dt`: `iso 8601 Date_time "YYYYMMDDTHHMMSSZ"`
- `uniq_id`: `hash(UUID)`
- `line_format`: `"tsv"| "json"`

For example `hr-opx-TestOpenmix1-provider-20363-20171209183034Z-cc907e8fd71eaf4e.tsv.gz`

What is the output file format?

The file format for Openmix and an Openmix HTTP API is JSON (gzipped).

Administration

April 13, 2020

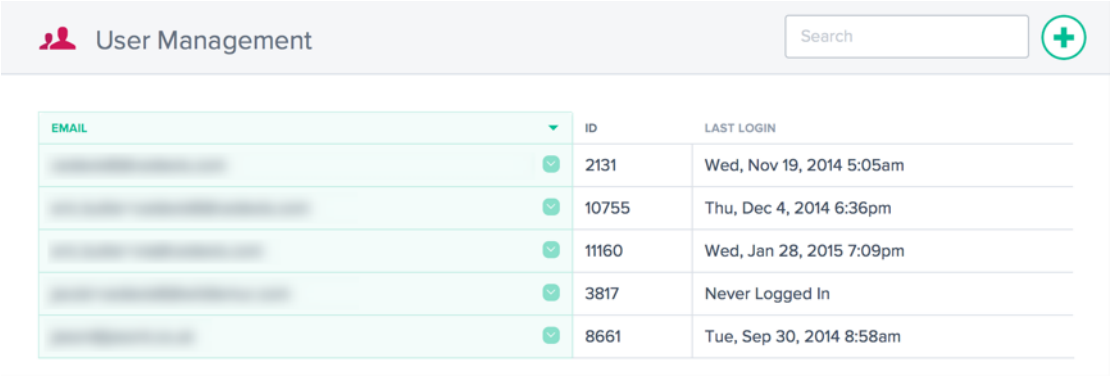
The **My Account** section is where the end-user can administer the account, the users that can access the account, and the users that can access Fusion purge features.

In addition, from the menu you can view invoices that are due, and manage OAuth API credentials.

Manage Users

Within the Users menu you can Add/Remove users and reset password access to the account.

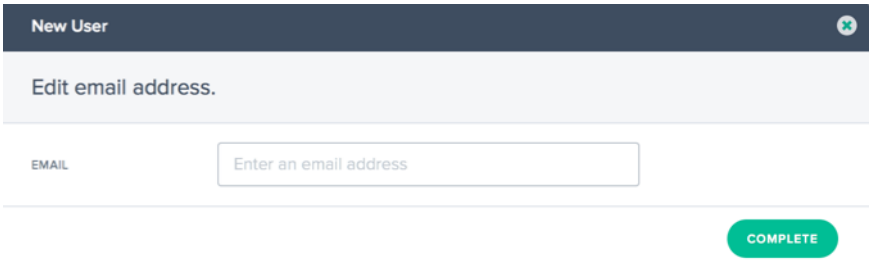
In addition to the user management you are able to enter email addresses for Service notifications and see when a user last logged in.



Adding or Removing Users and Resetting Passwords

When creating or adding users, ensure you use a valid email address. Passwords are created automatically and emailed to the email address that is entered as the user name.

To add a new user, click the **+** in the upper right corner. Enter a valid email address and click **Complete**.



To reset the password for a user, click the down arrow to the right of the user's email address, choose **Reset Password**, and confirm the action in the dialog by clicking **Yes**. A password reset email is sent to the user.

A user can be removed from the system by clicking the down arrow to the right of the user's email address and choosing **Delete**. Confirm the action and the user will be deleted from the system.

Single Sign On

We support using third party identity providers for Single Sign On login to the Portal via SAML 2.0.

Single Sign On is used for authentication of user logins. We do not currently pass authorization information in via SAML SSO. To be able to log in, a user must exist in the Intelligent Traffic Management Portal with the same email address as a user in the SSO identity provider.

Single Sign On is managed per account. Once SSO is turned on for an account, all users must use an SSO login to access the Portal.

You find the SAML configuration information in the **SSO Configuration** menu item. The information is specific your account and allows you to configure SSO in your identity provider. If you do not find the **SSO Configuration** menu, please contact the support team.

The setup is different for each identity provider but you need the following information, which is shown in the SSO Configuration page:

- Assertion Consumer Service (ACS) URL
- Entity ID
- Logout URL (optional, depending on provider)
- Start URL (optional, depending on provider)
- Name Format: Email
- Signed Response: No

Turn on Single Sign On

Generic steps for adding SSO to the Intelligent Traffic Management Portal:

1. Using the data in the SSO Configuration screen, set up the identity provider
2. Download the SSO IDP metadata file from the identity provider
3. Upload the file to the SSO Configuration page
4. When ready to enable SSO, click **Enable**
5. Users will now need to login via the SSO login page.

Turn off Single Sign On

If SSO is configured and enabled, click the **Disable** button.

Any user in the account that wants to log in will now need to use a Citrix password on the standard login screen. If a user does not know their password, an account administrator can send a password reset email or the user can request a password reset email from the login screen.

Configuration Steps for Google G Suite

The following are the steps necessary to use Single Sign On with Google G Suite logins:

In Google G Suite:

1. Open the G Suite administrative console to the Apps section
2. Click the **SAML apps** category
3. Click the **Enable SSO for a SAML Application** button
4. At the bottom of the dialog, choose **SETUP MY OWN CUSTOM APP**
5. On the Google IDP Information dialog, download the IDP metadata file under Option 2.
6. In Basic Information for your Custom App, the Application Name can be “Intelligent Traffic Management”
7. Fill in the following information from the SSO Configuration in Portal:
    - ACS URL: from the SSO Configuration info
    - Entity ID: from the SSO Configuration info
    - Start URL: from the SSO Configuration info (optional)
    - Name ID Format: EMAIL
8. Leave the Attribute Mapping dialog empty, click **FINISH** to create the SAML App
9. In the Apps list, click the vertical dots on the right of the Portal item and choose **ON for everyone**

In the Portal:

1. From the SSO Configuration page, upload the IDP metadata file; click the **Choose File** button to open the file browser and select the IDP metadata file downloaded from G Suite.
2. If the metadata file validates correctly, a green check mark appears.
3. Click **Enable** to enable SSO for all users in the account.

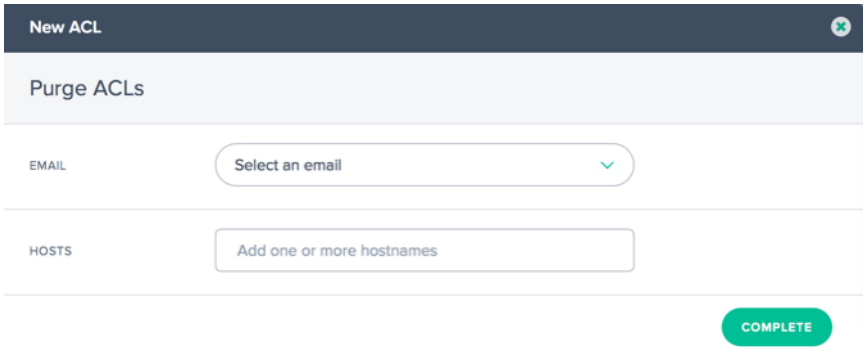
Users can now log in to the Intelligent Traffic Management Portal from the SSO login page or the **Apps** menu in **G Suite**.

For more information about Google G Suite SSO, refer to the Google help.

Setting Purge ACLs

In the **Purge ACLs** menu, users can have limitations placed on their ability to execute the Fusion Purge functionality. By default, users can run a purge on any host configured in the **Fusion Purge** settings. The Purge ACLs are used to limit users to only allow a purge on specified hosts.

Add new restrictions for a user by clicking the **’+’** button in the upper right corner. The following dialog appears:



Field	Description
Email	Select the email for the user you would like to configure limited purge access.
Hosts	Enter the host names for the user to run purges. Any host names not included in the list for the user will be unavailable for purge by the user.

Invoices

The **Invoices** menu option provides all the Invoices for the Intelligent Traffic Management services you have consumed. If there are any problems with the invoices, contact your sales representative or alternatively contact the support team.

API

Manage OAuth

The **API** menu option provides details on the Authenticated OAuth API Tokens you may want to use. If you want to use this functionality, contact your account manager.

REST API Rate Limits

REST APIs can be used to access data and settings stored in the platform. However, we limit the number of requests (to access this data) by putting a rate limit on them i.e., we limit the number of API calls a customer can make in a given time period. This is done to balance the load on the system.

Rate Limit Attributes

Rate limits have the following attributes:

- Time range (in minutes)
- Number of allowed requests
- Concurrent requests

Customers can request increases to their rate limits for their specific use case.

Default Rate Limits

The following table lists different types of API calls, and default rate limits that apply to each of them.

API Types	Default Rate Limits
Reporting Endpoints	**GET**
/v2/reporting/radar.json /v2/reporting/plt.json /v2/reporting/openmix.json /v2/reporting/sonar.json	15 requests per 15 minutes. 3 concurrent requests
Updating Applications	**PUT, POST**
/v2/config/applications/dns.json	10 requests per minute. 3 concurrent requests
Fusion Purge	**GET**
/v2/actions/fusion/purge.json	150 requests per minute
Fusion Purge	**POST**
/v2/actions/fusion/purge.json	1 request per minute. 3 concurrent requests

