Installing Thunder Observability Agent

April, 2024

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at:
a10-virtual-patent-marking.

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Introduction

The A10 Thunder Observability Agent is a custom plugin to monitor A10 Thunder® Application Delivery Agent (ADC) performance metrics and syslogs.

There are two types of A10 Thunder Observability Agent available:

Internal Thunder Observability Agent (iTOA)

This is an in-built Python plugin within ACOS which is configured using ACOS Command Line Interface (CLI) or aXAPI.

You can use iTOA:

- For ACOS v6.0.1 or later.
- For configuring vThunder using aXAPI or CLI to publish the metrics directly on the same AWS, Azure, or VMware platform where the vThunder instance is deployed with outbound internet connectivity.
- For configuring vThunder using aXAPI or CLI to publish the syslogs on:
  - AWS CloudWatch directly from vThunder with outbound internet connectivity.
  - Azure Log Analytics Workspace directly from vThunder with outbound internet connectivity to access '**.microsoftonline.com*' and '**.azure.com*'.
  - VMware vRealize Log Insight (vRLI) which is accessible from vThunder.
- For managing the data collection, processing, aggregation, and publishing internally for configured L3V partitions.
- For supporting maximum 20 partitions per vThunder instance.
- For publishing metrics or logs every 1 minute.

To configure the Internal Thunder Observability Agent, see Internal Thunder Observability Agent (iTOA).

External Thunder Observability Agent (TOA)

This external plugin can be installed on Linux, CentOS, and Ubuntu platforms as a Python Plugin installation package and Docker containerization.

You can use TOA:

- For any ACOS deployment platform.
- For any ACOS software version.
- For a Thunder with outbound internet connectivity restrictions.

  In this case, TOA can have outbound internet connectivity. It can collect data from Thunder and then publish the metrics and syslogs on the cloud monitoring tool through internet.

  TOA serves as an intermediary for managing Thunder Syslogs and 14 Thunder metrics. Syslogs can be directed to log analysis platforms like AWS, Azure, VMware, Elasticsearch (Kibana), Prometheus (Grafana), Splunk, Google Cloud Platform (GCP), and Oracle Cloud Infrastructure(OCI). Thunder metrics are exclusively sent to the platform where Thunder is deployed, which include AWS, Azure, and VMware. Additionally, TOA can send both logs and metrics to shared platforms like Elasticsearch (Kibana), Prometheus (Grafana), Splunk, GCP, and OCI.

To install the external Thunder Observability Agent, see External Thunder Observability Agent (TOA).

NOTE: It is recommended to configure any one TOA at a time.

Internal Thunder Observability Agent (iTOA)

The internal Thunder Observability Agent (iTOA) is an in-built capability in ACOS that can be configured for any vThunder device to publish the performance metrics and syslogs on the cloud monitoring tool.

The supported vThunder metrics and logs are listed below:

Supported vThunder Metrics

The following table lists the supported vThunder metrics:

Table 1 : Supported vThunder Metrics

Metric	Description
CPU Usage Percentage (Data)	Average data CPU usage, in percentage, for all data CPU configured within a vThunder instance for the last data collection cycle.
Memory Usage Percentage	Memory (RAM) usage, in percentage, of a vThunder instance for the last data collection cycle.
Disk Usage Percentage	Average disk storage usage, in percentage, for all disks associated with a vThunder instance for the last data collection cycle.
Throughput Rate (Global/BPS)	Total vThunder system global throughput bits per sec from vThunder instance to the server for the last data collection cycle.
Interface Down Count (Data)	Count of the total data network interfaces configured for a vThunder instance which is inactive for the last data collection cycle.
Total New Connection (Sec)	Count of the total new connections sent from vThunder instance to the server for the last data collection cycle per second. This includes L4-conns-per-sec, L7-conns-per-sec, L7-trans-per-sec, ssl-conns-per-sec, and ip-nat-conns-per-sec.
Transactions Rate (Sec)	Count of the total L7 transactions made per second from vThunder instance to the server for the last data collection

Table 1 : Supported vThunder Metrics

Metric	Description
	cycle.
Server Down Count	Count of the total web or app servers configured in the vThunder instance which are not reachable from vThunder for the last data collection cycle.
Server Down Percentage	Percentage of the total web or app servers configured in the vThunder instance which are not reachable from vThunder for the last data collection cycle.
SSL Errors Count	Count of the total errors that occurred during data transmission from vThunder to the server due to SSL connection, negotiate, encrypt, and decrypt for the last data collection cycle.
Server Errors Count	Count of the total errors that occurred during data transmission from vThunder to the server with status codes 4xx and 5xx for that last data collection cycle.
Total Session Count	Count of the total active sessions of the vThunder instance for the last data collection cycle.
Packet Rate (Sec)	Count of the total packets sent from or received at the vThunder instance for the last collection cycle.
Packet Drop Rate (Sec)	Count of the total packets dropped while sending data from or receiving data at the vThunder instance for the last collection cycle.

Supported vThunder Logs

The following table lists the supported vThunder logs:

Table 2 : Supported Thunder Logs

Logs	Description
SysLogs	Thunder internal logs such as: • SSL connection, negotiate, encrypt, and decrypt • Status codes 4xx and 5xx

AWS

iTOA can be configured to publish performance metrics and syslogs of a vThunder deployed on AWS.

The following topics are covered:

Publishing the vThunder Metrics on AWS

If the vThunder instance is deployed on the AWS cloud platform, the vThunder metrics can be published on the AWS CloudWatch.

To publish the vThunder metrics on AWS, perform the following steps:

1. Log in to the vThunder instance deployed on AWS using CLI with the administrative privilege:

   ```
   ACOS(config)#admin <admin_user>
   ```

   For example

   ```
   ACOS(config)#admin adminuser2
   ```

2. Import the AWS credentials and AWS configuration files:

   ```
   ACOS(config-admin:<admin_user>)#cloud-cred aws-cred import <file_
   transfer_method>
   ACOS(config-admin:<admin_user>)#cloud-cred aws-config import <file_
   transfer_method>
   ```

   The *<file_transfer_method>* can be any of the following:

```
 use-mgmt-port  Use management port as source port
 tftp:          Remote file path of tftp: file system(Format:
tftp://host/file)
 ftp:           Remote file path of ftp: file system(Format: ftp://
[user@]host[:port]/file)
 scp:           Remote file path of scp: file system(Format: scp://
[user@]host/file)
 sftp:          Remote file path of sftp: file system(Format: sftp://
[user@]host/file)
```

For example

```
ACOS(config-admin:adminuser2)#cloud-cred aws-cred import
tftp://192.168.0.0/credentials.txt
ACOS(config-admin:adminuser2)#cloud-cred aws-config import
tftp://192.168.0.0/configuration.txt
```

For a sample credentials file, see AWS Credentials File.

For a sample configuration file, see AWS Configuration File.

3. Verify if the AWS credentials and AWS configuration files are imported correctly:

```
ACOS(config-admin:<admin_user>)#cloud-cred aws-cred show
aws_access_key_id = XXXX
aws_secret_access_key = XXXX
ACOS(config-admin:<admin_user>)#cloud-cred aws-config show
region = XXXX
output = XXXX
```

4. Enable and configure the vThunder metrics.
   By default, all the metrics are disabled. You can enable one or more vThunder Metrics.

```
ACOS(config)#cloud-services cloud-provider
ACOS(config-cloud-provider)#aws
ACOS(config-cloud-provider-aws)#metrics
ACOS(config-cloud-provider-aws-metrics)#enable
ACOS(config-cloud-provider-aws-metrics)#active-partitions name
ACOS(config-cloud-provider-aws-metrics)#namespace name
ACOS(config-cloud-provider-aws-metrics)#cps enable
ACOS(config-cloud-provider-aws-metrics)#cpu enable
ACOS(config-cloud-provider-aws-metrics)#disk enable
ACOS(config-cloud-provider-aws-metrics)#interfaces enable
ACOS(config-cloud-provider-aws-metrics)#memory enable
ACOS(config-cloud-provider-aws-metrics)#packet-drop enable
ACOS(config-cloud-provider-aws-metrics)#packet-rate enable
ACOS(config-cloud-provider-aws-metrics)#server-down-count enable
ACOS(config-cloud-provider-aws-metrics)#server-down-percentage enable
ACOS(config-cloud-provider-aws-metrics)#server-error enable
ACOS(config-cloud-provider-aws-metrics)#sessions enable
ACOS(config-cloud-provider-aws-metrics)#ssl-cert enable
ACOS(config-cloud-provider-aws-metrics)#throughput enable
ACOS(config-cloud-provider-aws-metrics)#tps enable
```

NOTE: For better throughput, you must enable only those metrics that are required.

For more information on each CLI parameter, see the *Command Line Interface Reference*.

5. Verify the running configuration:

```
ACOS(config)#show running-config cloud-services cloud-provider
!Section configuration: 473 bytes
cloud-services cloud-provider
  aws
    metrics
      enable
      namespace vThunder
      active-partitions shared
      cpu enable
      memory enable
      disk enable
      throughput enable
      interfaces enable
      cps enable
      tps enable
      server-down-count enable
      server-down-percentage enable
      ssl-cert enable
      server-error enable
      sessions enable
      packet-drop enable
      packet-rate enable
!
```

6. Verify the **`thunder-observability-agent.log`** file:

```
-bash# tail -f /a10data/log/thunder-observability-agent.log
```

7. View the vThunder metrics.

   To view the Thunder metrics on AWS CloudWatch, perform the following steps:

   a. From the **AWS Management Console**, go to **CloudWatch** > **Metrics** > **All metrics**.

   b. Select **Browse** > *<your_Thunder_metric_namespace>*.

   c. Click the required metric to be monitored from the **Metrics** panel.

   d. Select the management IP of the Thunder instance to be monitored.

As the Thunder instances are selected, the metric data gets populated in the **Untitled Graph** panel for the selected the time range. For more information, see Graph a metric.

Publishing the vThunder Logs on AWS

When the vThunder instance is deployed on any AWS, Azure, or VMware cloud platform, the vThunder logs can be published to any one of the cloud platforms such as AWS CloudWatch, Azure Log Analytics Workspace, or VMware vRealize Log Insight (vRLI).

To publish the vThunder logs on AWS, perform the following steps:

1. Log in to the deployed vThunder instance using CLI with the administrative privilege:

```
ACOS(config)#admin <admin_user>
```

For example

```
ACOS(config)#admin adminuser2
```

2. Import the AWS credentials and AWS configuration files:

```
ACOS(config-admin:<admin_user>)#cloud-cred aws-cred import <file_
transfer_method>
ACOS(config-admin:<admin_user>)#cloud-cred aws-config import <file_
transfer_method>
```

The `<file_transfer_method>` can be any of the following:

```
 use-mgmt-port  Use management port as source port
 tftp:          Remote file path of tftp: file system(Format:
tftp://host/file)
 ftp:           Remote file path of ftp: file system(Format: ftp://
[user@]host[:port]/file)
 scp:           Remote file path of scp: file system(Format: scp://
[user@]host/file)
 sftp:          Remote file path of sftp: file system(Format: sftp://
[user@]host/file)
```

For example

```
ACOS(config-admin:adminuser2)#cloud-cred aws-cred import
tftp://192.168.0.0/credentials.txt
ACOS(config-admin:adminuser2)#cloud-cred aws-config import
tftp://192.168.0.0/configuration.txt
```

For a sample credentials file, see AWS Credentials File.

For a sample configuration file, see AWS Configuration File.

3. Verify if the AWS credentials and AWS configuration files are imported correctly:

```
ACOS(config-admin:<admin_user>)#cloud-cred aws-cred show
aws_access_key_id = XXXX
aws_secret_access_key = XXXX
ACOS(config-admin:<admin_user>)#cloud-cred aws-config show
region = XXXX
output = XXXX
```

4. Enable and configure the vThunder logs:

```
ACOS(config)#cloud-services cloud-provider
ACOS(config-cloud-provider)#aws
ACOS(config-cloud-provider-aws)#log
ACOS(config-cloud-provider-aws-log)#enable
ACOS(config-cloud-provider-aws-log)#log-group-name name
ACOS(config-cloud-provider-aws-log)#active-partitions name
```

For more information on each CLI parameter, see the *Command Line Interface Reference*.

5. Verify the running configuration:

```
ACOS(config)#show running-config cloud-services cloud-provider
!Section configuration: 103 bytes
cloud-services cloud-provider
  aws
    log
      enable
      log-group-name vThunder
      active-partitions shared
!
```

6. Verify the **`thunder-observability-agent.log`** file:

```
-bash# tail -f /a10data/log/thunder-observability-agent.log
```

7. View the vThunder logs on AWS CloudWatch:

   a. From the **AWS Management Console**, go to **CloudWatch** > **Logs** > **Log groups**.

   b. Click <*your_ log_group_name*>.

   c. Under the **Log streams** tab, click the required log stream to be monitored.

   The log stream format is 'DD/MM/YYYY/Management_IP/<*your_ log_group_name*>-<*Active_Partition_Name*>'.

   All logs are displayed in a tabular format with expandable details.

Sample Cloud Credentials File

The AWS cloud-cred files must be a text file and it should have the cloud-specific parameters.

AWS Credentials File

The sample AWS credentials.txt file is as follows:

```
aws_access_key_id = XXXX
aws_secret_access_key = XXXX
```

Table 3 : AWS Credentials File Parameters

Parameter	Description
`aws_access_key_id`	To get the access key ID and secret access key, perform the following steps: 1. Open the IAM console. 2. On the navigation menu, select **Users**. 3. Select your IAM username. 4. Open the **Security credentials** tab and select **Create access key**. 5. To view the new access key, select **Show**.
`aws_secret_access_key`	

AWS Configuration File

The sample AWS configuration.txt file is as follows:

```
region = XXXX
output = XXXX
```

Table 4 : AWS Config File Parameters

Parameter	Description
`region`	Specifies the AWS logged-in user's working region. **Example** `us-east-1`
`output`	Specify `json` as the AWS CLI output format.

Azure

iTOA can be configured to publish performance metrics and syslogs of a vThunder deployed on Azure.

The following topics are covered:

Publishing the vThunder Metrics on Azure

If the vThunder instance is deployed on Azure cloud platform, the vThunder metrics can be published on the Azure Application Insights.

To the publish vThunder metrics on Azure, perform the following steps:

1. Log in to the vThunder instance deployed on Azure using CLI with the administrative privilege:

```
ACOS(config)#admin <admin_user>
```

For example

```
ACOS(config)#admin adminuser2
```

2. Import the Azure credentials file:

```
ACOS(config-admin:<admin_user>)#cloud-cred azure-cred import <file_
transfer_method>
```

The *<file_transfer_method>* can be any of the following:

```
 use-mgmt-port  Use management port as source port
 tftp:          Remote file path of tftp: file system(Format:
tftp://host/file)
 ftp:           Remote file path of ftp: file system(Format: ftp://
[user@]host[:port]/file)
 scp:           Remote file path of scp: file system(Format: scp://
[user@]host/file)
 sftp:          Remote file path of sftp: file system(Format: sftp://
[user@]host/file)
```

For example

```
ACOS(config-admin:adminuser2)#cloud-cred azure-cred import
tftp://192.168.0.0/credentials.txt
```

For a sample credentials file, see Azure Credentials File.

3. Verify if the Azure credentials file is imported correctly:

```
ACOS(config-admin:<admin_user>)#cloud-cred azure-cred show
azure_workspace_primary_key = XXXX
azure_client_id = XXXX
azure_secret_id = XXXX
azure_tenant_id = XXXX
azure_location = XXXX
```

4. Enable and configure the vThunder metrics.
   By default, all the metrics are disabled. You can enable one or more vThunder Metrics.

```
ACOS(config)#cloud-services cloud-provider
ACOS(config-cloud-provider)#azure
ACOS(config-cloud-provider-azure)#metrics
ACOS(config-cloud-provider-azure-metrics)#enable
ACOS(config-cloud-provider-azure-metrics)#active-partitions name
ACOS(config-cloud-provider-azure-metrics)#resource-id ID
ACOS(config-cloud-provider-azure-metrics)#cps enable
ACOS(config-cloud-provider-azure-metrics)#cpu enable
ACOS(config-cloud-provider-azure-metrics)#disk enable
ACOS(config-cloud-provider-azure-metrics)#interfaces enable
ACOS(config-cloud-provider-azure-metrics)#memory enable
ACOS(config-cloud-provider-azure-metrics)#packet-drop enable
ACOS(config-cloud-provider-azure-metrics)#packet-rate enable
ACOS(config-cloud-provider-azure-metrics)#server-down-count enable
ACOS(config-cloud-provider-azure-metrics)#server-down-percentage enable
ACOS(config-cloud-provider-azure-metrics)#server-error enable
ACOS(config-cloud-provider-azure-metrics)#sessions enable
ACOS(config-cloud-provider-azure-metrics)#ssl-cert enable
ACOS(config-cloud-provider-azure-metrics)#throughput enable
ACOS(config-cloud-provider-azure-metrics)#tps enable
```

NOTE: For better throughput, you must enable only those metrics that are required.

To get **`resource-id`** value, go to **Azure Portal** > **Azure services** > **Virtual machine** > *<your_vThunder_instance>* > **Setting** > **Properties** and get the **Resource ID** from the right panel.

For more information on each CLI parameter, see the *Command Line Interface Reference*.

5. Verify the running configuration:

```
ACOS(config)#show running-config cloud-services cloud-provider
!Section configuration: 473 bytes
cloud-services cloud-provider
  azure
    metrics
      enable
      resource-id /subscriptions/07d34b9b-61e3-475a-abbc-
006b16812a3e/resourceGroups/vth-
rg6/providers/microsoft.insights/components/vth-vmss-app-insights
      active-partitions shared
      cpu enable
      memory enable
      disk enable
      throughput enable
      interfaces enable
      cps enable
      tps enable
      server-down-count enable
      server-down-percentage enable
      ssl-cert enable
      server-error enable
      sessions enable
      packet-drop enable
      packet-rate enable
!
```

6. Verify the **`thunder-observability-agent.log`** file:

```
-bash# tail -f /a10data/log/thunder-observability-agent.log
```

7. View the vThunder metrics on Azure Application Insights.

   a. From the **Azure Portal**, go to **Azure services** > **Resource Groups** > *<your_resource_group>* and click *<your_app_insight_name>*.

      OR

From the **Azure Portal**, go to **Azure services** > **Resource Groups** > *<your_resource_group>* and click *<your_vThunder_instance_name>* whose metric is to be monitored.

b. Click **Metrics** from the left **Monitoring** panel.

c. Select the appropriate resources whose metrics you want to view:

Table 5 : Azure Application Insight Dashboard

Field Name	Description
Scope	If you are adding the metrics from **Application Insight** window, the selected app insight name is auto-populated. If you are adding the metrics from vThunder instance window, select your app insight name.
Metric Namespace	Select **Thunder**.
Metric	Select a metric from the drop-down. For the list of available vThunder metrics, see Supported vThunder Metrics.

As a metric is selected, the corresponding data is plotted in the chart area for the selected the time range.

d. To view multiple metrics on the same chart, click **Add metric** and repeat the above step. For more information, see Metrics Explorer.

Publishing the vThunder Logs on Azure

When the vThunder instance is deployed on any AWS, Azure, or VMare cloud platform, the vThunder logs can be published to any one of the cloud platforms such as AWS CloudWatch, Azure Log Analytics Workspace, or VMware vRealize Log Insight (vRLI).

To publish the vThunder logs on Azure Log Analytics Workspace, perform the following steps:

1. Log in to the deployed vThunder instance using CLI with the administrative privilege:

```
ACOS(config)#admin <admin_user>
```

For example

```
ACOS(config)#admin adminuser2
```

2. Import the Azure credentials file:

```
ACOS(config-admin:<admin_user>)#cloud-cred azure-cred import <file_
transfer_method>
```

The *<file_transfer_method>* can be any of the following:

```
 use-mgmt-port  Use management port as source port
 tftp:          Remote file path of tftp: file system(Format:
tftp://host/file)
 ftp:           Remote file path of ftp: file system(Format: ftp://
[user@]host[:port]/file)
 scp:           Remote file path of scp: file system(Format: scp://
[user@]host/file)
 sftp:          Remote file path of sftp: file system(Format: sftp://
[user@]host/file)
```

For example

```
ACOS(config-admin:adminuser2)#cloud-cred azure-cred import
tftp://192.168.0.0/credentials.txt
```

For a sample credentials file, see Azure Credentials File.

3. Verify if the Azure credentials file is imported correctly:

```
ACOS(config-admin:<admin_user>)#cloud-cred azure-cred show
azure_workspace_primary_key = XXXX
azure_client_id = XXXX
azure_secret_id = XXXX
azure_tenant_id = XXXX
azure_location = XXXX
```

4. Enable and configure the vThunder logs:

```
ACOS(config)#cloud-services cloud-provider
ACOS(config-cloud-provider)#azure
ACOS(config-cloud-provider-azure)#log
ACOS(config-cloud-provider-azure-log)#enable
ACOS(config-cloud-provider-azure-log)#resource-id ID
ACOS(config-cloud-provider-azure-log)#workspace-id ID
ACOS(config-cloud-provider-azure-log)#active-partitions name
```

To get `resource-id` value, go to **Azure Portal** > **Azure services** > **Virtual machine** > <*your_vThunder_instance*> > **Setting** > **Properties** and get the **Resource ID** from the right panel.

To get `workspace-id` value, go to **Azure Portal** > **Azure services** > **Log Analytics workspaces** > <*your_log_analytics_workspace*> > **Settings** > **Agents**.

For more information on each CLI parameter, see the *Command Line Interface Reference*.

5. Verify the running configuration:

```
ACOS(config)#show running-config cloud-services cloud-provider
!Section configuration: 103 bytes
cloud-services cloud-provider
  azure
    log
      enable
      resource-id /subscriptions/07d34b9b-61e3-475a-abbc-
006b16812a3e/resourceGroups/vth-
rg10/providers/Microsoft.Compute/virtualMachineScaleSets/vth-
vmss/virtualMachines/1
      workspace-id dcfd78d5-3a49-425d-8410-e02e281f7991
      active-partitions shared
!
```

6. Verify the `thunder-observability-agent.log` file:

```
-bash# tail -f /a10data/log/thunder-observability-agent.log
```

7. View the vThunder logs on Azure Log Analytics Workspace:

a. From the **Azure Portal**, go to **Azure services** > **Resource Groups** > *<your_resource_group>* and click *<your_log_analytics_workspace_name>*.

b. Click **Logs** from the left **General** panel.

   You can close the **Queries** pop-up window.

c. From **New Query1** > **Tables** tab, expand **Custom Logs**.

d. Double-click **THUNDER_SYSLOG_CL**.

e. Click **Run**.

   All logs are displayed in tabular format with expandable details.

   The following table lists the Thunder Logs filter options:

Table 6 : Log Filters

Filter	Description
`log_data`	Specifies the actual log entry.
`hostname`	Displays the vThunder resource ID.
`log_type`	Displays the vThunder system logs.
`appname`	Displays the application name.
`ip`	Displays the vThunder IP address.
`agent`	Displays the agent name.
`jobid`	Displays the JOB ID provided in the **`thunder-observability-agent.log`** file.
`priority`	Displays the Notice, Info, Error, and so on as per actual log entry.
`partition`	Displays the vThunder partition name.

Sample Cloud Credential File

The Azure cloud-cred file must be a text file and it should have the cloud-specific parameters.

Azure Credentials File

The sample Azure credentials.txt file is as follows:

```
azure_workspace_primary_key = XXXX
azure_client_id = XXXX
azure_secret_id = XXXX
azure_tenant_id = XXXX
azure_location = XXXX
```

Table 7 : Azure Credentials File Parameters

Parameter	Description
azure_workspace_primary_key	To get the workspace primary key, go to **Azure Portal** > **Azure services** > **Log Analytics workspaces** > *<log_analytics_workspace>* > **Settings** > **Agents**.
azure_client_id	To get the client ID, secret ID, and tenant ID, go to **Azure Portal** > **Azure services** > **Azure Active Directory** > **App Registration** > **Owned applications** > *<application_name>*.
azure_secret_id	
azure_tenant_id	
azure_location	To get the location, go to **Azure Portal** > **Azure services** > **Resource Groups** > *<your_resource_group>* > **Overview** > **Essentials** > **Location**.

VMware

iTOA can be configured to publish performance metrics and syslogs of a vThunder instance deployed on VMware.

The following topics are covered:

Publishing the vThunder Metrics on VMware

If the vThunder instance is deployed on the VMware cloud platform, the vThunder metrics can be published on the VMware vRealize Operations Manager (vROps).

To publish the vThunder metrics on VMware, perform the following steps:

1. Log in to the vThunder instance deployed on VMware using CLI with the administrative privilege:

   ```
   ACOS(config)#admin <admin_user>
   ```

   For example

   ```
   ACOS(config)#admin adminuser2
   ```

2. Import the VMware credentials file:

   ```
   ACOS(config-admin:<admin_user>)#cloud-cred vmware-cred import <file_
   transfer_method>
   ```

   The *<file_transfer_method>* can be any of the following:

   ```
    use-mgmt-port  Use management port as source port
    tftp:          Remote file path of tftp: file system(Format:
   tftp://host/file)
    ftp:           Remote file path of ftp: file system(Format: ftp://
   [user@]host[:port]/file)
    scp:           Remote file path of scp: file system(Format: scp://
   [user@]host/file)
    sftp:          Remote file path of sftp: file system(Format: sftp://
   [user@]host/file)
   ```

   For example

   ```
   ACOS(config-admin:adminuser2)#cloud-cred vmware-cred import
   tftp://192.168.0.0/credentials.txt
   ```

   For a sample credentials file, see VMware Credentials File.

3. Verify if the VMware credentials file is imported correctly:

```
ACOS(config-admin:<admin_user>)#cloud-cred vmware-cred show
vmware_vrops_username = XXXX
vmware_vrops_password = XXXX
```

4. Enable and configure the vThunder metrics.
   By default, all the metrics are disabled. You can enable one or more vThunder Metrics.

```
ACOS(config)#cloud-services cloud-provider
ACOS(config-cloud-provider)#vmware
ACOS(config-cloud-provider-vmware)#metrics
ACOS(config-cloud-provider-vmware-metrics)#enable
ACOS(config-cloud-provider-vmware-metrics)#active-partitions name
ACOS(config-cloud-provider-vmware-metrics)#resource-id ID
ACOS(config-cloud-provider-vmware-metrics)#vrops-host num
ACOS(config-cloud-provider-vmware-metrics)#cps enable
ACOS(config-cloud-provider-vmware-metrics)#cpu enable
ACOS(config-cloud-provider-vmware-metrics)#disk enable
ACOS(config-cloud-provider-vmware-metrics)#interfaces enable
ACOS(config-cloud-provider-vmware-metrics)#memory enable
ACOS(config-cloud-provider-vmware-metrics)#packet-drop enable
ACOS(config-cloud-provider-vmware-metrics)#packet-rate enable
ACOS(config-cloud-provider-vmware-metrics)#server-down-count enable
ACOS(config-cloud-provider-vmware-metrics)#server-down-percentage
enable
ACOS(config-cloud-provider-vmware-metrics)#server-error enable
ACOS(config-cloud-provider-vmware-metrics)#sessions enable
ACOS(config-cloud-provider-vmware-metrics)#ssl-cert enable
ACOS(config-cloud-provider-vmware-metrics)#throughput enable
ACOS(config-cloud-provider-vmware-metrics)#tps enable
```

NOTE: For better throughput, you must enable only those metrics that are required.

To get `resource-id` value, go to **vRealize Operations Web UI Home** > **Environment** > **Object Browser** > **All Objects** > **vCenter Adapter** > **Virtual Machine** > **vThunder** and get the resource ID from the URL.

For more information on each CLI parameter, see the *Command Line Interface Reference*.

5. Verify the running configuration:

```
ACOS(config)#show running-config cloud-services cloud-provider
!Section configuration: 473 bytes
cloud-services cloud-provider
  azure
    metrics
      enable
      vrops-host 10.67.4.13
      active-partitions shared
      resource-id 3ae28ba2-c8b9-497f-8b98-76bedc93f31c
      cpu enable
      memory enable
      disk enable
      throughput enable
      interfaces enable
      cps enable
      tps enable
      server-down-count enable
      server-down-percentage enable
      ssl-cert enable
      server-error enable
      sessions enable
      packet-drop enable
      packet-rate enable
!
```

6. Verify the **`thunder-observability-agent.log`** file:

```
-bash# tail -f /a10data/log/thunder-observability-agent.log
```

7. View the vThunder metrics.

   To view the Thunder metrics on VMware vRealize Operations Manager, perform the following steps:

a. Ensure the vROps virtual machine is powered on and reachable.

b. Create a dashboard for vThunder. For more information, see Create a Dashboard.

c. Create an alert for vThunder. For more information, see Create an Alert.

d. Create a notification for vThunder. For more information, see Create a Notification.

e. From the **vRealize Operations Web UI**, go to **Home** > **Visualize** > **Dashboard** and select your dashboard created for the Thunder metrics.

f. From **Object List**, double-click your Thunder instance.

g. From **Metric Picker**, expand **Metrics** > **THUNDER** and double-click the following common metrics:

- Memory Usage Percentage
- Disk Usage Percentage

As a metric is selected, the corresponding data gets populated in the **Metric Chart** panel for the selected the time range.

h. From **Metric Picker**, expand **Metrics** > **THUNDER-SHARED** or **THUNDER-Px** and double-click the following metrics:

- CPU Usage Percentage (Data)
- Throughput Rate (Global/BPS)
- Interface Down Count (Data)
- Total New Connection (Sec)
- Transactions Rate (Sec)
- Server Down Count
- Server Down Percentage
- SSL Errors Count
- Server Errors Count
- Total Session Count

- Packet Rate (Sec)
- Packet Drop Rate (Sec)

As the metric is selected, the corresponding data gets populated in the **Metric Chart** panel for the selected the time range.

To view multiple metrics data, select each of those metrics. The data corresponding to each metric is displayed in the **Metric Chart** panel.

NOTE: If you encounter any resource ID issues for cross-platform log monitoring, disable the VMware metric monitoring and re-enable it.

Create a Dashboard

To create a dashboard manually, perform the following steps:

1. From the **vRealize Operations Web UI**, go to **Home** > **Visualize** > **Dashboards** and click **Create** to add a new dashboard.
2. Provide a name to the new dashboard and double-click or drag the following widgets:
   - Object List
   - Metric Picker
   - Metric Chart
3. Click **Show Interactions** to create interactions.
4. Drag the connectors and create interactions.
5. Click **Save** to save the changes.

   A dashboard for Thunder metrics is created.

Create an Alert

To create an alert definition manually, perform the following steps:

1. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** and click **Alert Definitions**.
2. Click **Add** in the **Alert Definitions** window.
3. Enter or select the appropriate values in the following fields:

Table 8 : Alert tab fields

Field Name	Description
Name	Enter the alert name. **Example** `ThunderAlert`
Base Object Type	Select **vCenter Adapter** > **Virtual Machine**.
Under the **Advanced Settings**:	
Impact	Select **Health**.
Criticality	Select **Critical**.
Alert Type & Subtype	Select **Application : Performance**.

4. Click **Next**.
5. Click **Select Specific Object** to select your Thunder instance in the **Symptoms / Conditions** tab .
6. Select your Thunder instance and click **Select** in the **Select Object** window.

   The selected Thunder instance is listed under **Conditions**.
7. Select **Metrics** > **Thunder** and drag the required metrics to the left-side panel.
8. Specify the appropriate alert condition.
9. Click **Next**.
10. Add the appropriate recommendations in the **Recommendations** tab, if needed.
11. Click **Next**.
12. Select appropriate policy in the **Policies** tab, if needed.
13. Click **Next**.

    The **Notification** tab is displayed. The notification can be created after the alert definition is created. For more information, see Create a Notification.
14. Click **Create** in the **Notification** tab.

    An alert definition is created and is listed in the **Alert Definition** window.

Create a Notification

To create a notification manually, perform the following steps:

1. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** and click **Notifications**.
2. Click **Add** in the **Notifications** window.
3. Enter or select the appropriate values in the following fields:

Table 9 : Notifications tab

Field Name	Description
Name	Enter the notification name. **Example** `ThunderAlertNotification`
Notification Status	Select **Enable**.

4. Click **Next**.
5. In the **Criteria** field, select **Object Type** from the drop-down.

   A field appears to select the object type.
6. Expand **vCenterAdapter** and select **Virtual Machine** from the drop-down.

   The selected object type is listed under **Criteria**.
7. In the **Category** field, select **Alert Definition** from the drop-down created in the Create an Alert.
8. Search your alert definition.
9. Select your alert definition and drag it to add as the criteria.
10. Click **OK**.

    The selected alert definition is listed under Category.
11. In the **Status** field under **Notify On**, select the alert status for which you want to receive the notifications.
12. Click **Next**.

13. In the **Outbound method** field, select **Standard Email Plugin** from the drop-down list.
14. Click **Create New Instance** to create a new instance for corresponding Outbound method.
15. Enter or select the appropriate values in the following fields:

Table 10 : Create New Instance

Field Name	Description
Instance Name	Enter the notification instance name. **Example** `ThunderNotificationInstance`
SMTP Host	Enter the URL or IP address of the email host server.
SMTP Port	Enter the SMTP port number used to connect with the email host server.
Secure Connection Type	Select **SSL**.
User Name	Enter the username that is used to connect to the email server.
Password	Enter the password for the connection username that appears on the notification message.
Sender Email Address	Enter the email address of the sender.
Sender Name	Enter the display name of the sender email address.
Receiver Email Address	Enter the email address of the receiver that receives the notification.

16. Click **Save** to save the changes.

    The new instance is populated in the **Select Instance** field.

17. Click **Next**.
18. Enter or select the appropriate values in the following fields for the default

template:

Table 11 : Select Payload Template tab

Field Name	Description
Recipient(s)	Enter the email addresses of the recipient to receive the notification.
Max Notifications	Enter the maximum number of notification to be sent for the active alert.
Delay to notify	Enter the delay time in minutes before sending a notification when a new alert is generated.

19. Click **Create**.

    A new notification is created for the selected alert definition and it is listed in the **Notifications** window.

Publishing the vThunder Logs on VMware

When the vThunder instance is deployed on any AWS, Azure, or VMare cloud platform, the vThunder logs can be published to any one of the cloud platforms such as AWS CloudWatch, Azure Log Analytics Workspace, or VMware vRealize Log Insight (vRLI).

To publish the vThunder logs on VMware vRealize Log Insight, perform the following steps:

1. Log in to the deployed vThunder instance using CLI with the administrative privilege:

   ```
   ACOS(config)#admin <admin_user>
   ```

   For example

   ```
   ACOS(config)#admin adminuser2
   ```

2. Import the VMware credentials file:

   ```
   ACOS(config-admin:<admin_user>)#cloud-cred vmware-cred import <file_
   transfer_method>
   ```

   The *<file_transfer_method>* can be any of the following:

```
 use-mgmt-port  Use management port as source port
 tftp:          Remote file path of tftp: file system(Format: 
tftp://host/file)
 ftp:           Remote file path of ftp: file system(Format: ftp://
[user@]host[:port]/file)
 scp:           Remote file path of scp: file system(Format: scp://
[user@]host/file)
 sftp:          Remote file path of sftp: file system(Format: sftp://
[user@]host/file)
```

For example

```
ACOS(config-admin:adminuser2)#cloud-cred vmware-cred import 
tftp://192.168.0.0/credentials.txt
```

For a sample credentials file, see VMware Credentials File.

3. Verify if the VMware credentials file is imported correctly:

```
ACOS(config-admin:<admin_user>)#cloud-cred vmware-cred show
vmware_vrops_username = XXXX
vmware_vrops_password = XXXX
```

4. Enable and configure the vThunder logs:

```
ACOS(config)#cloud-services cloud-provider
ACOS(config-cloud-provider)#vmware
ACOS(config-cloud-provider-vmware)#log
ACOS(config-cloud-provider-vmware-log)#enable
ACOS(config-cloud-provider-vmware-log)#vrli-host IP_address
ACOS(config-cloud-provider-vmware-log)#active-partitions name
```

For more information on each CLI parameter, see the *Command Line Interface Reference*.

5. Verify the running configuration:

```
ACOS(config)#show running-config cloud-services cloud-provider
!Section configuration: 103 bytes
cloud-services cloud-provider
  vmware
    log
      enable
      vrli-host 10.67.4.16
      active-partitions shared
!
```

6. Verify the **`thunder-observability-agent.log`** file:

```
-bash# tail -f /a10data/log/thunder-observability-agent.log
```

7. View the vThunder logs on the VMware vRLI:

   a. From a the **vRealize Log Insight Web UI**, go to **Home** > **Explore Logs**.

   b. Click **Add Filter** and add the following filter criteria to search all the logs received from a specific Thunder IP:

      - _index: ip
      - condition: is
      - value: *<vThunder_IP>*

   c. Add the following filter criteria to search all logs:

      - _index: source
      - condition: is
      - value: <Source_*IP>*

   d. Verify if the logs are generated.

      The following table lists the vThunder Logs filter options:

Table 12 : Log Filters

Filter	Description
`hostname`	Displays the vThunder resource ID.
`log_type`	Displays the vThunder system logs.
`appname`	Displays the application name.

Table 12 : Log Filters

Filter	Description
`ip`	Displays the vThunder IP address.
`agent`	Displays the agent name.
`jobid`	Displays the JOB ID provided in **`thunder-observability-agent.log`** file.
`priority`	Displays the Notice, Info, Error, and so on as per actual log entry.
`partition`	Displays the vThunder partition name.

Sample Cloud Credential File

The VMware cloud-cred file must be a text file and it should have the cloud-specific parameters.

VMware Credentials File

The sample VMware credentials.txt file is as follows:

```
vmware_vrops_username = XXXX
vmware_vrops_password = XXXX
```

Table 13 : VMware Credentials File Parameters

Parameter	Description
`vmware_vrops_username`	Specifies your vROps login credentials.
`vmware_vrops_password`	

External Thunder Observability Agent (TOA)

The external Thunder Observability Agent (TOA) is a lightweight autonomous data processing engine that can be externally installed and configured for any Thunder device.

The TOA offers the following capabilities for Thunder® Application Delivery Controller (ADC):

- Collects, processes, and publishes 14 Thunder metrics. The default data collection frequency is 1 minute. Thunder metrics can be sent to the platform where Thunder is deployed, which includes AWS, Azure, and VMware or can be sent to shared platforms like Elasticsearch (Kibana), Prometheus (Grafana), Splunk, Google Cloud Platform (GCP), and Oracle Cloud Infrastructure (OCI). Metrics can be sent to any one platform at a time. For more information on Thunder metrics, see Supported Thunder Metrics.
- Collects, processes, and publishes Thunder Syslogs. The default data collection frequency is 1 minute. The logs can be published on various platforms like AWS, Azure, VMware, Kibana (Elasticsearch), Grafana (Prometheus and Pushgateway), Splunk, GCP, and OCI. Logs can be sent to any one platform at a time. For more information on Thunder logs, see Supported Thunder Logs.
- Manages the data collection, processing, aggregation, and publishing internally.
- Provides multitasking capabilities to collect and process data from multiple Thunder instances and their partitions simultaneously. By default, it collects data from a shared partition.
- TOA supports Shared and L3V partitions. The maximum number of partitions supported per Thunder is 20.
- Installs on any orchestration platform such as public cloud compute instances, private cloud physical or virtual machines, hypervisor VMs, and on-premise physical hardware and is self-driven.
- Installs on Linux, CentOS, and Ubuntu platforms as a Python Plugin installation package and Docker containerization.
- Supports single or multiple Thunder instances.

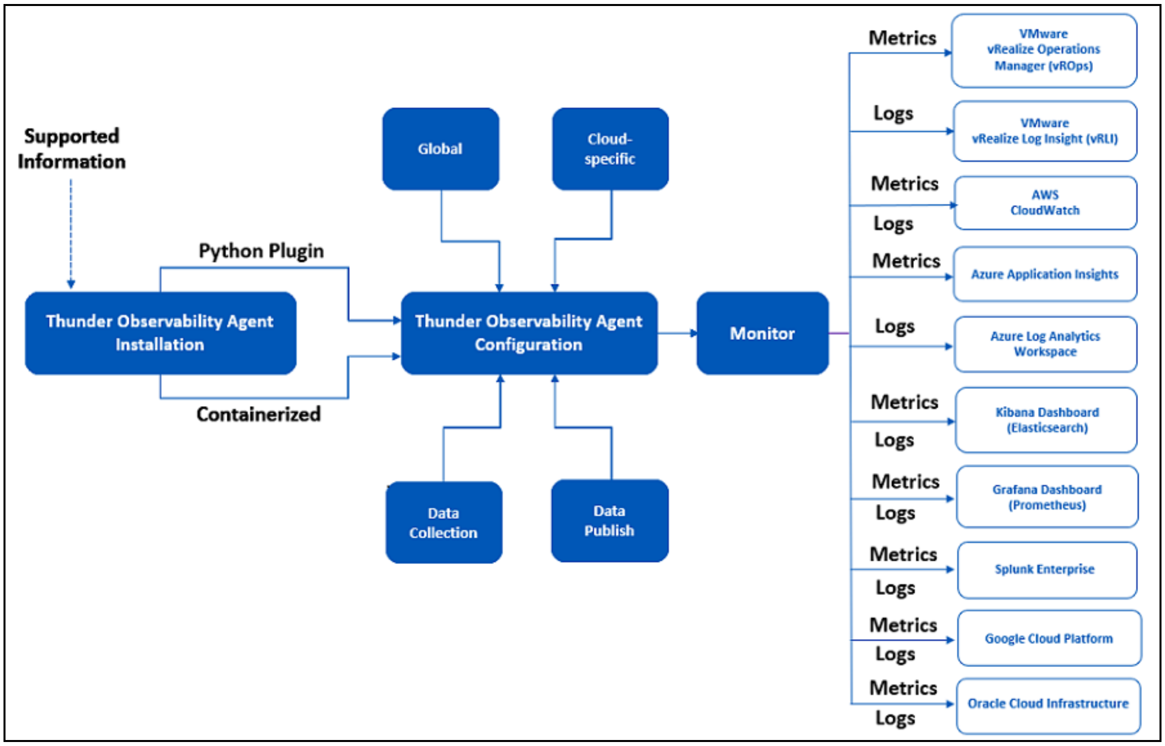
- Supports Thunder instances running under AWS Auto Scaling Group or Azure Virtual Machine Scale Set (VMSS).
- Collects data from any type of Thunder device installed on public cloud compute instances, private cloud physical or virtual machines, hypervisor VMs, and on-premise physical hardware installation.
- Publishes data to Azure Cloud, AWS Cloud, VMware ESXi, Kibana (Elasticsearch), Grafana (Prometheus and Pushgateway), Splunk,GCP, and OCI.

Download Links

- Python Central Repository
- Docker Central Repository
- A10 GitHub Repository

The following figure shows the TOA workflow.

Figure 1 : TOA Workflow


Supported Information
Python Plugin
Thunder Observability Agent Installation
Containerized
Global
Cloud-specific
Thunder Observability Agent Configuration
Data Collection
Data Publish
Monitor
Metrics
VMware vRealize Operations Manager (vROps)
Logs
VMware vRealize Log Insight (vRLI)
Metrics
Logs
AWS CloudWatch
Metrics
Azure Application Insights
Logs
Azure Log Analytics Workspace
Metrics
Logs
Kibana Dashboard (Elasticsearch)
Metrics
Logs
Grafana Dashboard (Prometheus)
Metrics
Logs
Splunk Enterprise
Metrics
Logs
Google Cloud Platform
Metrics
Logs
Oracle Cloud Infrastructure

Supported Technology

The following table provides TOA-supported technologies:

Table 14 : Supported Technologies

Name	Version	License
Python	3.10	PSF License Python 3.3 license | Python.org
Requests	2.27.1	Apache Software License 2.0 Apache 2.0 (amazon.com)
Boto3	1.24.25	
google-auth	2.22.0	Apache Software License 2.0, Apache 2.0 (google.com)
oci	2.121.1	Apache Software License and Universal Permissive License

Supported Thunder Metrics

The following table lists the TOA-supported Thunder metrics:

Table 15 : Supported Thunder Metrics

Metric	Description
CPU Usage Percentage (Data)	Average data CPU usage, in percentage, for all data CPU configured within a Thunder instance for the last data collection cycle.
Memory Usage Percentage	Memory (RAM) usage, in percentage, of a Thunder instance for the last data collection cycle.
Disk Usage Percentage	Average disk storage usage, in percentage, for all disks associated with a Thunder instance for the last data collection cycle.
Throughput Rate	Total Thunder system global throughput bits per sec from

Table 15 : Supported Thunder Metrics

Metric	Description
(Global/BPS)	Thunder instance to the server for the last data collection cycle.
Interface Down Count (Data)	Count of the total data network interfaces configured for a Thunder instance which is inactive for the last data collection cycle.
Total New Connection (Sec)	Count of the total new connections sent from Thunder instance to the server for the last data collection cycle per second. This includes L4-conns-per-sec, L7-conns-per-sec, L7-trans-per-sec, ssl-conns-per-sec, and ip-nat-conns-per-sec.
Transactions Rate (Sec)	Count of the total L7 transactions made per second from Thunder instance to the server for the last data collection cycle.
Server Down Count	Count of the total web or app servers configured in the Thunder instance that are not reachable from Thunder for the last data collection cycle.
Server Down Percentage	Percentage of the total web or app servers configured in the Thunder instance that are not reachable from Thunder for the last data collection cycle.
SSL Errors Count	Count of the total errors that occurred during data transmission from Thunder to the server due to SSL connection, negotiate, encrypt, and decrypt for the last data collection cycle.
Server Errors Count	Count of the total errors that occurred during data transmission from Thunder to the server with status codes 4xx and 5xx for that last data collection cycle.
Total Session Count	Count of the total active sessions of the Thunder instance for the last data collection cycle.
Packet Rate (Sec)	Count of the total packets sent from or received at the Thunder instance for the last collection cycle. **NOTE:** Applicable for ACOS 5.2.1-P7, ACOS 6.0.0, and higher

Table 15 : Supported Thunder Metrics

Metric	Description
Packet Drop Rate (Sec)	Count of the total packets dropped while sending data from or receiving data at the Thunder instance for the last collection cycle. **NOTE:** Applicable for ACOS 5.2.1-P7, ACOS 6.0.0, and higher

Supported Thunder Logs

The following table lists the TOA-supported Thunder logs:

Table 16 : Supported Thunder Logs

Logs	Description
SysLogs	Thunder internal logs such as: • SSL connection, negotiate, encrypt, and decrypt • Status codes 4xx and 5xx

Supported ACOS Versions

The following table provides the TOA-supported ACOS versions:

Table 17 : Supported ACOS versions

ACOS Version	TOA Version	ADC	CGN	SSLi	TPS
64-bit Advanced Core OS (ACOS) version 6.0.3-P1	>=1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 6.0.3	>=1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS)	>=1.0.0	√	X	X	X

Table 17 : Supported ACOS versions

ACOS Version	TOA Version	ADC	CGN	SSLi	TPS
version 6.0.2					
64-bit Advanced Core OS (ACOS) version 6.0.1	>=1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 6.0.0-P2-SP1	>= 1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 6.0.0-P1	>= 1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P9	>= 1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P8	>= 1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P7	>= 1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P6	>= 1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P5	>= 1.0.0	√	X	X	X
64-bit Advanced Core OS (ACOS) version 4.1.4-GR1-x	1.0.0	√	X	X	X

Supported Platforms

The following table provides the TOA supported platforms and monitoring applications:

Table 18 : Supported platforms and monitoring tools

Cloud Platform	Monitoring Applications
AWS Cloud	• CloudWatch
Azure Cloud	• Application Insights • Log Analytics Workspace
VMware ESXi (On Premise)	• vRealize Operations Manager (vROps) • vRealize Log Insight (vRLI)
Elasticsearch	• Kibana
Prometheus	• Grafana
Splunk	• Splunk Analytics • Splunk Dashboard
Google Cloud Platform (GCP)	• Metrics Explorer • Logs Explorer
Oracle Cloud Infrastructure (OCI)	• Metrics Explorer • Log Search

Install TOA

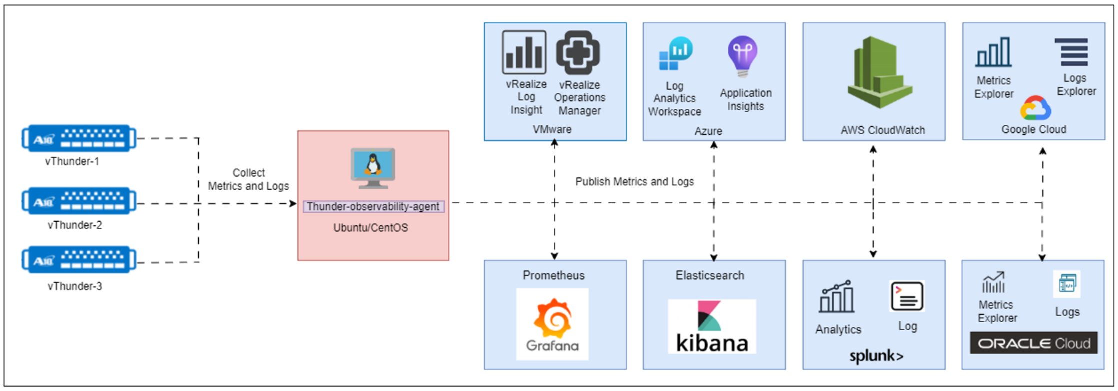
TOA is a standalone software that can be installed on any orchestration platform. The following installation options are available:

- Python Plugin Installation

  TOA is installed on Linux/CentOS/Ubuntu platform using a Python plugin.

  Figure 2 illustrates the installation of TOA in the Python plugin architecture.

Figure 2 : Python Plugin Installation Architecture

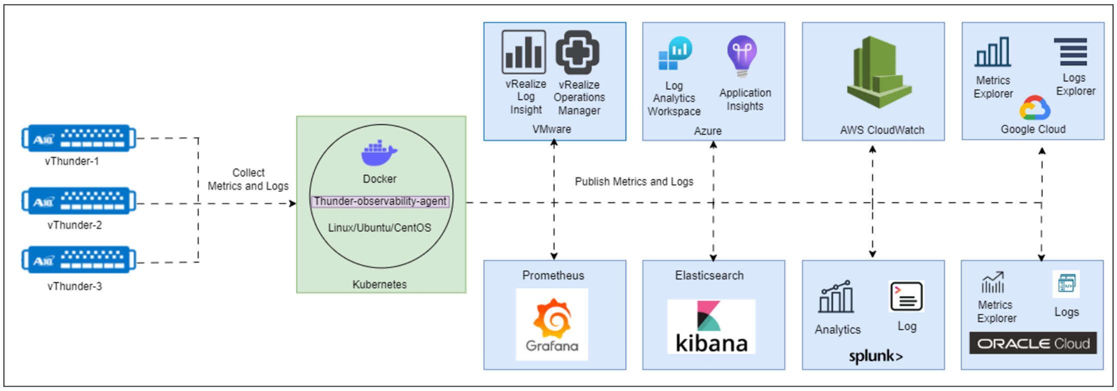


- [Containerized Installation]

TOA is installed on the Kubernetes cluster using a docker image.

Figure 3 illustrates the installation of TOA in a containerized architecture.

Figure 3 : Containerized Installation Architecture



Python Plugin Installation

This section describes how to install and configure a Thunder Observability Agent (TOA) on any public cloud, private cloud, hypervisor VM, or on-premise machine using Python plugin.

The following topics are covered:

Prerequisites

The following tables list the prerequisites for installing TOA using the Python plugin:

Hardware Dependencies

Table 19 : Hardware Dependencies

Requirement	Description
Virtual Machine	2 GB RAM, 1 CPU, 4 GB **NOTE:** The hardware configuration is applicable for one to ten Thunder instances with moderate transactions.
Platform	Any public cloud, private cloud, hypervisor VM, or on-premise machine.
Instance Type	Dedicated or Shared.

Software Dependencies

Table 20 : Software Dependencies

Requirement	Description
Operating System	• CentOS 7 or higher • Ubuntu 20 or higher
Python	3.6 or higher
Access-level	Root

Installation Steps

To install TOA using the Python plugin, perform the following steps:

1. Log in to the instance where you want to install TOA.
2. Depending on your operating system, install Python version, Crontab, and Syslog. For the installation steps, see Install Python, Crontab, and Syslog.

   If the Python version, Crontab, and Syslog are already installed, skip this step.

3. Create a virtual environment.

   ```
   pip3 install virtualenv
   cd /usr
   virtualenv toaenv
   source toaenv/bin/activate
   ```

4. Run the following command to install the TOA:

   ```
   pip3 install thunder-observability-agent
   ```

   After the execution, all the following configuration files are available at the default location `/usr/toaenv/thunder-observability-agent`:

   - `main.properties`
   - `config.json`
   - `logging.conf`
   - `init.sh`

5. Run `init.sh`, a one-time execution script, to enable crontab job for data collection and create credential files for Thunder and cloud providers:

   ```
   cd /usr/toaenv/thunder-observability-agent
   sh init.sh
   ```

   After the execution, all the following files are available at the `/root/` hidden folder:

   - `.thunder/credentials`
   - `.aws/config`
   - `.aws/credentials`
   - `.azure/credentials`
   - `.vmware/credentials`
   - `.splunk/credentials`
   - `.elasticsearch/credentials`
   - `.pushgateway/credentials`

- `.gcp/credentials`
- `.oci/credentials`

6. If you want to change the default location of the TOA config files, update the environment variable `TOA_CONFIG_PATH` and the Logging file.

   If you do not want to change the default location, skip this step.

7. If you want to change the credentials file location, update the Main Properties file.

8. Verify Crontab configuration.

9. Verify TOA installation.

   The `agent.log` file is created at the `/var/log/thunder-observability-agent` path. For the sample `agent.log` file, see TOA Logging.

10. Edit the configuration files.

    Depending on your cloud provider, configure the following files mentioned in Table 21:

    - Thunder credentials to collect data from Thunder.
    - Cloud credentials to establish a connection with the cloud provider.
    - `Config.json` to publish required metrics or logs.

Table 21 : Cloud specific Configuration Files

Cloud	File name
AWS	• Thunder Credentials • AWS Config • AWS Credentials • Config JSON
Azure	• Thunder Credentials • Azure Credentials • Config JSON
VMware	• Thunder Credentials

Table 21 : Cloud specific Configuration Files

Cloud	File name
	• VMware Credentials • Config JSON
Elasticsearch	• Thunder Credentials • Elasticsearch Credentials • Config JSON
Prometheus (PushGateway)	• Thunder Credentials • PushGateway Credentials • Config JSON
Splunk	• Thunder Credentials • Splunk Credentials • Config JSON
GCP	• Thunder Credentials • GCP Credentials • Config JSON
OCI	• Thunder Credentials • OCI Credentials • Config JSON

11. Monitor Thunder metrics and logs.

    For more information, see Monitor Dashboard.

Containerized Installation

This section describes how to install TOA in a single container pod of the Kubernetes cluster using YAML files.

The following topics are covered:

Prerequisites

The following are the prerequisites for installing TOA using Containers:

- Kubernetes environment
- Download the Kubernetes TOA manifest files installation files.

Installation Steps

To install the TOA in a container, perform the following steps:

1. Run the following command to create TOA namespace:

```
kubectl create namespace thunder-observability-agent
```

2. Run the following command to set TOA as the default Kubernetes namespace:

```
kubectl config set-context --current --namespace=thunder-observability-
agent
```

3. Edit the YAML files.

   Depending on your cloud provider, configure the following files mentioned in Table 22:

   - Thunder credentials to collect data from Thunder.
   - Cloud credentials to establish a connection with the cloud provider.
   - `configmap.yaml` to publish required metrics or logs.

Table 22 : Cloud specific Configuration Files

Cloud	File name	Reference
AWS	`aws-configmap.yaml`	• Main Properties • Config JSON • Logging
	`aws-secret.yaml`	• AWS Config • AWS Credentials • Thunder Credentials
Azure	`azure-configmap.yaml`	• Main Properties • Config JSON

Table 22 : Cloud specific Configuration Files

Cloud	File name	Reference
		• Logging
	`azure-secret.yaml`	• Azure Credentials • Thunder Credentials
VMware	`vmware-configmap.yaml`	• Main Properties • Config JSON • Logging
	`vmware-secret.yaml`	• VMware Credentials • Thunder Credentials
Elasticsearch	`elasticsearch-configmap.yaml`	• Main Properties • Config JSON • Logging
	`elasticsearch-secret.yaml`	• Elasticsearch Credentials • Thunder Credentials
Prometheus	`pushgateway-configmap.yaml`	• Main Properties • Config JSON • Logging
	`pushgateway-secret.yaml`	• PushGateway Credentials • Thunder Credentials
Splunk	`splunk-configmap.yaml`	• Main Properties • Config JSON • Logging
	`splunk-secret.yaml`	• Splunk Credentials • Thunder Credentials
Google Cloud (GCP)	`gcp-configmap.yaml`	• Main Properties

Table 22 : Cloud specific Configuration Files

Cloud	File name	Reference
		• Config JSON • Logging
	`gcp-secret.yaml`	• GCP Credentials • Thunder Credentials • Base64 Conversion
Oracle Cloud Infrastructure (OCI)	`oci-configmap.yaml`	• Main Properties • Config JSON • Logging
	`oci-secret.yaml`	• OCI Credentials • Thunder Credentials • Base64 Conversion

4. Run the following commands to apply the cloud-specific configuration:

```
kubectl apply -f <cloud-provider>-configmap.yaml
kubectl apply -f <cloud-provider>-secret.yaml
```

5. Run any of the following commands to apply and create a container:

```
kubectl apply -f <cloud-provider>-pod.yaml
```

or

```
kubectl apply -f <cloud-provider>-cronjob.yaml
```

6. Verify TOA installation.

   The `agent.log` file is created at the `/var/log/thunder-observability-agent` path. For the sample `agent.log` file, see TOA Logging.

7. Monitor Thunder metrics and logs.

   For more information, see Monitor Dashboard.

NOTE: By default, the system works using the default configuration.
TOA only supports a single pod installation.

Configure TOA

This section lists the global TOA configuration files and cloud-specific configuration files that are required to establish a connection with TOA.

The following topics are covered:

Global Configuration

The following files are used for the global TOA configurations:

- Main Properties
- Logging
- Crontab

Main Properties

This file lists the global TOA configuration parameters. If you want to change the configuration file path, this file must be updated with the correct paths.

File Path: `/usr/toaenv/thunder-observability-agent/main.properties`

Table 23 : File Parameters

Parameter	Description	Default Value
`log_collection_delay_min`	Specifies the latency of log collection in minutes. The system considers the Thunder logs that are generated from the Start Time until the End	0

Table 23 : File Parameters

Parameter	Description	Default Value
	Time as: Start Time = Last data collection time End Time = Current data collection time - *<log_collection_delay_min>* **Example** If the current data collection time is 10:00:00 AM and the last data collection time is 09:59:00 AM, then: the Start Time is 9:59:00 AM. the End Time is 10:00:00 AM (which is 10:00:00 AM - 0 minutes). So, TOA collects all the logs generated by Thunder instance from 9:59:00 AM to 10:00:00 AM.	
`cron_job_frequency_min`	Specifies the cron job frequency in minutes. This parameter should match with the `crontab -e` job definition. The system considers `crontab -e` for job scheduling. If the	`1`

Table 23 : File Parameters

Parameter	Description	Default Value
	frequency is changed in this parameter, it should also change in the `crontab` file. For more information, see Crontab.	
`http_ssl_verify`	Disables SSL certificate verification over HTTPS. If a user wants to enable SSL: • For CA signed certificate configured in Thunder, set the parameter to **True**.	`False`

Table 23 : File Parameters

Parameter	Description	Default Value
	NOTE: For a self-signed certificate configured in Thunder, create a *.pem file, import the Thunder public certificate, and provide the path in place of **True**. **Example** `/usr/toaenv/thunder-observability-agent/toa.pem`	

Table 23 : File Parameters

Parameter	Description	Default Value
	NOTE: If vROps and vRLI haves self-signed certificates, then their public certificates must be imported in *.pem file.	
`http_connection_timeout_sec`	Specifies the maximum amount of time, in seconds, that the TOA waits to set up an HTTP connection to communicate with any Thunder instance.	`15`
`max_threads`	Specifies the maximum number of threads to be created at the same time.	`2000`
`config_path`	Specifies the configuration file path for publishing logs and metrics.	`/usr/toaenv/thunder-observability-agent/config.json`
`thunder_credentials_path`	Specifies the configuration file path to collect data from any of the following: - Single Thunder	`/root/.thunder/credentials`

Table 23 : File Parameters

Parameter	Description	Default Value
	Instance - Multiple Thunder Instances - Thunder Instances in AWS Auto scaling Group - Thunder Instances in Azure VMSS.	
`aws_credentials_path`	Specifies the AWS credentials file path to establish a connection and publish the data to AWS CloudWatch. **NOTE:** Applicable only if you want to publish the Thunder data to AWS CloudWatch.	`/root/.aws/credentials`
`aws_config_path`	Specifies the AWS configuration file path to publish the data. **NOTE:** Applicable only if you want to publish the Thunder data to AWS CloudWatch.	`/root/.aws/config`
`azure_credentials_path`	Specifies the Azure credentials file path to establish the	`/root/.azure/credentials`

Table 23 : File Parameters

Parameter	Description	Default Value
	connection and publish the data. **NOTE:** Applicable only if you want to publish the Thunder data to Azure Application Insights and Azure Log Analytics Workspace.	
`vmware_credentials_path`	Specifies the VMware credentials file path to establish the connection and publish the data. **NOTE:** Applicable only if you want to publish the Thunder data to VMware vROps.	`/root/.vmware/credentials`
`elasticsearch_credentials_path`	Specifies the Elasticsearch credentials file path to establish the connection and publish the data.	`/root/.elasticsearch/credentials`

Table 23 : File Parameters

Parameter	Description	Default Value
	NOTE: Applicable only if you want to publish the Thunder data to Elasticsearch Kibana.	
`pushgateway_credentials_path`	Specifies the Pushgateway credentials file path to establish the connection and publish the data. **NOTE:** Applicable only if you want to publish the Thunder data to Prometheus Grafana.	`/root/.pushgateway/credentials`
`splunk_credentials_path`	Specifies the Splunk credentials file path to establish the connection and publish the data. **NOTE:** Applicable only if you want to publish the Thunder data to Splunk.	`/root/.splunk/credentials`
`gcp_credentials_path`	Specifies the GCP credentials file path to establish the connection and publish	`/root/.gcp/credentials`

Table 23 : File Parameters

Parameter	Description	Default Value
	the data. **NOTE:** Applicable only if you want to publish the Thunder data to Google Cloud Platform.	
`oci_credentials_path`	Specifies the OCI credentials file path to establish the connection and publish the data. **NOTE:** Applicable only if you want to publish the Thunder data to Oracle Cloud Infrastructure.	`/root/.oci/credentials`

Logging

This file lists the TOA logging configurations.

File Path: `/usr/toaenv/thunder-observability-agent/logging.conf`

```
[loggers]
keys=root

[handlers]
keys=hand01

[formatters]
keys=form01

[logger_root]
```

```
level=INFO
handlers=hand01


[handler_hand01]
class=logging.handlers.RotatingFileHandler

# ERROR, INFO
level=INFO
formatter=form01

# logFilePath, append, maxBytes, backupCount
args=('/var/log/thunder-observability-agent/agent.log', 'a', 5000000,
100)

[formatter_form01]
format=%(asctime)s - (%(filename)s:%(lineno)d) - %(levelname)s - %
(message)s
datefmt=
style=%
validate=True
class=logging.Formatter
```

Crontab

By default, TOA creates the crontab configuration file that contains the command to configure the data collection frequency. This command is executed at regular intervals.

To verify the crontab configuration, perform the following steps:

1. Run the following command to verify the Python version:

   ```
   python3 --version
   ```

   In case if the version is other than `python3.10`, then replace in the crontab.

2. Run the following command to open the crontab file:

   ```
   $ crontab -e
   ```

3. In case if required, edit the Python version as appropriate:

```
*/1 * * * * /usr/toaenv/bin/python3 /usr/toaenv/lib/python3.10/site-
packages/thunder-observability-agent/toa.py
```

NOTE: By default, TOA collects data at a frequency of 1 minute. If you are changing the frequency in the `crontab` file, you should change the `cron_job_frequency_min` parameter in the `main.properties` as well and vice-versa. For more information, see Main Properties.

Cloud-specific Configuration

The following information is required to setup the cloud-specific configuration to publish the Thunder metrics and logs.

- AWS Config
  (Applicable only if you want to publish the data to AWS CloudWatch)
- AWS Credentials
  (Applicable only if you want to publish the data to AWS CloudWatch)
- Azure Credentials
  (Applicable only if you want to publish the data to Azure Application Insights and Azure Log Analytics Workspace)
- VMware Credentials
  (Applicable only if you want to publish the data to vRealize Operations (vROps))
- Elasticsearch Credentials

  (Applicable only if you want to publish the data to Elasticsearch Kibana)
- PushGateway Credentials

  (Applicable only if you want to publish the data to Prometheus Grafana)
- Splunk Credentials

  (Applicable only if you want to publish the data to Splunk)
- GCP Credentials

  (Applicable only if you want to publish the data to GCP)

- OCI Credentials

  (Applicable only if you want to publish the data to OCI)

AWS Config

This file lists the AWS configurations to publish the Thunder metrics or logs to AWS CloudWatch.

File Path: `/root/.aws/config`

Update the following parameters according to your AWS setup:

```
[default]
region = XXXX
output = XXXX
```

Table 24 : AWS Config File Parameters

Parameter	Description
`region`	Specifies the AWS logged-in user's working region. **Example** `us-east-1`
`output`	Specify `json` as the AWS CLI output format.

For sample configuration, see Examples.

AWS Credentials

This file lists the AWS credential configurations to publish the Thunder metrics or logs to AWS CloudWatch.

File Path: `/root/.aws/credentials`

Update the following parameters according to your AWS setup:

```
[default]
aws_access_key_id = XXXX
aws_secret_access_key = XXXX
```

Table 25 : AWS Credentials File Parameters

Parameter	Description
`aws_access_key_id`	To get the access key ID and secret access key, perform the following steps: 1. Open the IAM console. 2. On the navigation menu, select **Users**. 3. Select your IAM user name. 4. Open the **Security credentials** tab, and select **Create access key**. 5. To view the new access key, select **Show**.
`aws_secret_access_key`	

For sample configuration, see Examples.

Azure Credentials

This file lists the Azure credential configurations to publish the Thunder metrics or logs to Azure Application Insights and Azure Log Analytics Workspace respectively.

File Path: `/root/.azure/credentials`

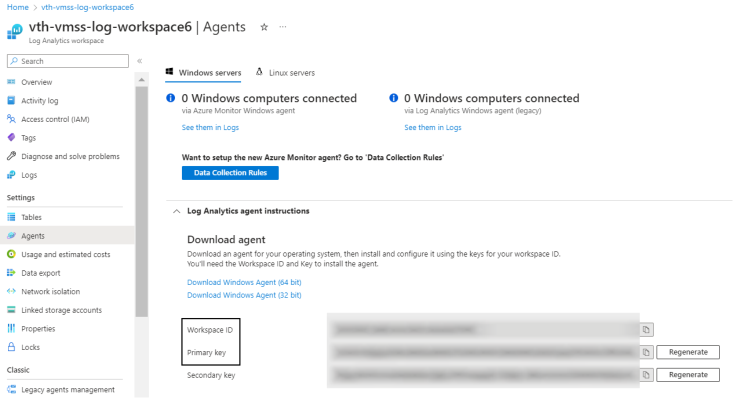
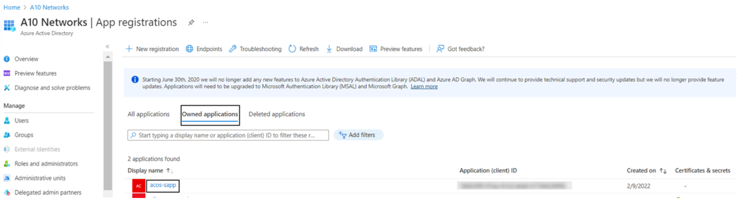
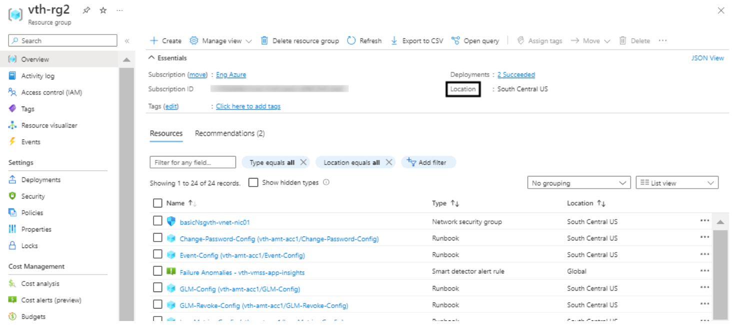
Update the following parameters according to your Azure setup:

```
azure_workspace_primary_key = XXXX
azure_client_id = XXXX
azure_secret_id = XXXX
azure_tenant_id = XXXX
azure_location = XXXX
```

Table 26 : Azure Credentials File Parameters

Parameter	Description
`azure_workspace_primary_key`	To get the workspace primary key, go to **Azure Portal** > **Azure services** > **Log Analytics workspaces** > *<log_analytics_workspace>* > **Settings** > **Agents**. Figure 4 : Agents window

Table 26 : Azure Credentials File Parameters

Parameter	Description
	 Home > vth-vmss-log-workspace6 vth-vmss-log-workspace6 | Agents Log Analytics workspace Search Overview Activity log Access control (IAM) Tags Diagnose and solve problems Logs Settings Tables Agents Usage and estimated costs Data export Network isolation Linked storage accounts Properties Locks Classic Legacy agents management Windows servers Linux servers 0 Windows computers connected via Azure Monitor Windows agent See them in Logs 0 Windows computers connected via Log Analytics Windows agent (legacy) See them in Logs Want to setup the new Azure Monitor agent? Go to 'Data Collection Rules' Log Analytics agent instructions Download agent Download an agent for your operating system, then install and configure it using the keys for your workspace ID. You'll need the Workspace ID and Key to install the agent. Download Windows Agent (64 bit) Download Windows Agent (32 bit) Workspace ID [redacted] Primary key [redacted] Secondary key [redacted]
`azure_client_id`	To get the client ID, secret ID, and tenant ID, go to **Azure Portal** > **Azure services** > **Azure Active Directory** > **App Registration** > **Owned applications** > <*application_name*>. Figure 5 : Azure active directory - App registrations window  Home > A10 Networks A10 Networks | App registrations Azure Active Directory New registration Endpoints Troubleshooting Refresh Download Preview features Got feedback? Overview Preview features Diagnose and solve problems Manage Users Groups External Identities Roles and administrators Administrative units Delegated admin partners Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more All applications Owned applications Deleted applications Start typing a display name or application (client) ID to filter these r... Add filters 2 applications found Display name | Application (client) ID | Created on | Certificates & secrets acos-sapp | [redacted] | 2/9/2022 | -
`azure_secret_id`	
`azure_tenant_id`	
`azure_location`	To get the location, go to **Azure Portal** > **Azure services** > **Resource Groups** > <*your_resource_group*> > **Overview** > **Essentials** > **Location**. Figure 6 : Resource Group window  vth-rg2 Resource group Search Overview Activity log Access control (IAM) Tags Resource visualizer Events Settings Deployments Security Policies Properties Locks Cost Management Cost analysis Cost alerts (preview) Budgets Create Manage view Delete resource group Refresh Export to CSV Open query Assign tags Move Delete JSON View Essentials Subscription (move) : Eng Azure Subscription ID [redacted] Tags (edit) : Click here to add tags Deployments : 2 Succeeded Location : South Central US Resources Recommendations (2) Filter for any field... Type equals all Location equals all Add filter No grouping List view Showing 1 to 24 of 24 records. □ Show hidden types Name | Type | Location basicNsgvth-vnet-nic01 | Network security group | South Central US Change-Password-Config (vth-amt-acc1/Change-Password-Config) | Runbook | South Central US Event-Config (vth-amt-acc1/Event-Config) | Runbook | South Central US Failure Anomalies - vth-vmss-app-insights | Smart detector alert rule | Global GLM-Config (vth-amt-acc1/GLM-Config) | Runbook | South Central US GLM-Revoke-Config (vth-amt-acc1/GLM-Revoke-Config) | Runbook | South Central US

For sample configuration, see Examples.

VMware Credentials

This file lists the VMware credential configurations to publish the metrics or logs.

File Path: `/root/.vmware/credentials`

Update the following parameters according to your VMware setup:

```
vmware_vrops_username = XXXX
vmware_vrops_password = XXXX
```

Table 27 : VMware Credentials File Parameters

Parameter	Description
`vmware_vrops_username`	Specifies your vROps login credentials.
`vmware_vrops_password`	

For sample configuration, see Examples.

Elasticsearch Credentials

This file lists the Elasticsearch credential configurations to publish the metrics or logs.

File Path: `/root/.elasticsearch/credentials`

Update the following parameters according to your Elasticsearch setup:

```
username = XXXX
password = XXXX
```

Table 28 : Elasticsearch Credentials File Parameters

Parameter	Description
`username`	Specifies your Elasticsearch login credentials.
`password`	

For sample configuration, see Examples.

PushGateway Credentials

This file lists the PushGateway credential configurations to publish the metrics or logs.

File Path: `/root/.pushgateway/credentials`

Update the following parameters according to your PushGateway setup:

```
username = XXXX
password = XXXX
```

Table 29 : PushGateway Credentials File Parameters

Parameter	Description
`username`	Specifies your PushGateway login credentials.
`password`	

For sample configuration, see Examples.

Splunk Credentials

This file lists the Splunk credential configurations to publish the metrics or logs.

File Path: `/root/.splunk/credentials`

Update the following parameters according to your Splunk setup:

```
token_log = XXXX
token_metric = XXXX
```

Table 30 : Splunk Credentials File Parameters

Parameter	Description
`token_log`	Specifies your Splunk HEC token for logs and metrics.
`token_metric`	

For sample configuration, see Examples.

GCP Credentials

This file lists the GCP credential configurations to publish the metrics or logs.

File Path: `/root/.gcp/credentials`

Update the following parameters according to your GCP setup:

```
gcp_project_id = XXXX
gcp_service_key_path = XXXX
```

Table 31 : GCP Credentials File Parameters

Parameter	Description
`gcp_project_id`	Specifies your GCP project ID and path to the service account key file. To obtain the GCP project ID and service account key file path, perform the following steps: 1. Open Google Cloud Console and select the project you want to work with. 2. Navigate to **IAM & Admin** > **Service Accounts**. 3. Click **Create Service Account** and provide the service account details. 4. Click **Create and continue**, followed by **Done**. The service account will be created. 5. On the **Service Accounts** page, select the created service account and click the three dots (**…**) on the **Action** column. 6. Select **Manage Keys**. The **Keys** page will be displayed. 7. Click the **Add key** drop-down menu and select **Create new key**. 8. Select the **Key type** as **JSON** and click **Create**. The service account key file will be download to your system. 9. Open the file in a text editor and locate the **project_id** field. The value of this keys represents `gcp_project_id` in the GCP credentials file. 10. Store the downloaded JSON securely and provide its path as the `gcp_service_key_path` in the GCP credentials file.
`gcp_service_key_path`	

For sample configuration, see Examples.

OCI Credentials

This file lists the OCI credential configurations to publish the metrics or logs.

File Path: `/root/.oci/credentials`

Update the following parameter according to your OCI setup:

```
oci_api_key_path= XXXXXXXX
```

Table 32 : OCI Credentials File Parameter

Parameter	Description
`oci_api_key_path`	Specifies the path to the private key file used for authenticating the OCI services. To obtain the `oci_api_key_path`, perform the following steps: 1. Log in to the Oracle Cloud Infrastructure console, open the **Profile** menu, and click **My Profile**. 2. In the **Resources** section, click **API Keys**. 3. Click **Add API Key**. The **Add API Key** dialog will be displayed. 4. Click **Download Private Key**. The file will be downloaded to your system. Store this file securely. 5. Click **Add**. The **Configuration File Preview** page will be displayed. This page allows you to preview the configuration file. This file includes basic authentication information required to create your configuration file. 6. Copy and paste the configuration snippet from the text box into your text editor and save the configuration file without specifying any file extension. 7. After pasting the snippet, update the **`key_file`** parameter with the location where the private key file is saved (downloaded and saved previously).

Table 32 : OCI Credentials File Parameter

Parameter	Description
	The newly created configuration file (without any file extension) is considered an OCI API key file. 8. Provide this API key file path as the `oci_api_key_path` in the OCI credentials file.

For sample configuration, see Examples.

Data Collection Configuration

In your topology, there can be a single, multiple, or auto scale Thunder instances that are either installed on AWS, Azure, or VMware compute instances. To collect the Thunder metrics or logs, configure the Thunder `credentials` file depending on the type of Thunder instance/s:

- Single Thunder Instance
- Multiple Thunder Instances
- Thunder Instances in AWS Auto scaling Group
- Thunder Instances in Azure VMSS

For more information on TOA - Thunder configuration with, see TOA Thunder Configuration Matrix.

Thunder Credentials

This file lists the Thunder credential configurations to collect the Thunder metrics, logs, or both.

File Path: `/root/.thunder/credentials`

Update the Thunder `credentials` file to provide the credentials of the Thunder instance/s whose metrics or logs are to be monitored as per the type of Thunder instance:

Single Thunder Instance

Provide the details of the Thunder instance running on any platform.

```
{
      "thunders": [{
             "ip": "XXXX",
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX",
             "active_partitions": "shared"
       }]
}
```

Multiple Thunder Instances

Provide the details of the Thunder instances running on any platform.

```
{
      "thunders": [{
             "ip": "XXXX",
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX",
             "active_partitions": "shared"
       },
               {
             "ip": "XXXX",
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX",
             "active_partitions": "shared"
       }]
}
```

Thunder Instances in AWS Auto scaling Group

Provide the details of the Thunder instances running in AWS Auto Scaling Group.

```
{
      "autoscale" : 1,
      "provider" : "aws",
      "thunders": [{
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX",
             "active_partitions": "shared"
       }]
}
```

Thunder Instances in Azure VMSS

Provide the details of Thunder instances running in Azure VMSS.

```
{
      "autoscale" : 1,
      "provider" : "azure",
      "thunders": [{
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX",
             "active_partitions": "shared"
       }]
}
```

Table 33 : Thunder Credentials File Parameters

Parameter	Description
`autoscale`	Specify 1 if the Thunder instance is in AWS auto scale group or Azure virtual machine scale set. By default, it is disabled.
`provider`	Specifies the cloud provider only if the Thunder instance is in AWS auto scale group or Azure virtual machine scale set (`autoscale=1`). The following options are available: • `aws` • `azure`

Table 33 : Thunder Credentials File Parameters

Parameter	Description
`thunders`	Specifies the Thunder instance details. The following parameters are available: • `ip` • `username` • `password` • `resource_id`
`ip`	Specifies the Thunder instance IP address.
`username`	Specifies the Thunder instance username.
`password`	Specifies the Thunder instance password.
`resource_id`	Specifies the compute instance resource IDS on which Thunder is deployed. For more information, see Get Resource ID.
`active_partitions`	Specifies one or more comma-separated partition/s for which the Thunder metrics or logs are viewed. By default, the active partition is "Shared". **For example**: "SHARED, Px" The maximum number of partitions supported per Thunder is 20. Only L3V active partitions are supported. To view Thunder metrics or logs of all active partitions, specify "*". To collect data from one active partition, one session is required through management interface. **For example**: If a user has defined 20 partitions in one Thunder device then 20 concurrent sessions are created in the device while collecting the data.

For sample configuration, see Examples.

TOA Thunder Configuration Matrix

The following table provides the TOA Thunder Configuration Matrix.

Table 34 : TOA Thunder Configuration Matrix

Logs	Metrics	Cron Cycle	Partition per Thunder	Maximum Number of Thunder devices
Enabled	Enabled	1 min	Up to 20 Partitions on each Thunder	Up to 05 Thunder Device
Enabled	Enabled	1 min	Up to 08 Partitions on each Thunder	Up to 10 Thunder Device
Enabled	Enabled	1 min	Up to 06 Partitions on each Thunder	Up to 15 Thunder Device

For example: If all logs and all metrics are enabled for every 1 minute of the data collection cycle with 20 active partitions on each Thunder device, ideally up to 5 Thunder devices can be configured per TOA instance.

Data Publish Configuration

The Thunder metrics and logs can be published on the cloud platforms such as AWS, Azure, VMware, Kibana (Elasticsearch), Grafana (Prometheus and Pushgateway), Splunk, Google Cloud Platform (GCP), or Oracle Cloud Infrastructure (OCI). To publish the Thunder metrics or logs, configure the `config.json` file with the appropriate TOA parameters for the required cloud platform:

- Metrics
  - AWS
  - Azure
  - VMware
  - Elasticsearch
  - PushGateway
  - Splunk

  - Google Cloud Platform (GCP)
  - Oracle Cloud Infrastructure (OCI)
- Logs
  - AWS
  - Azure
  - VMware
  - Elasticsearch
  - Prometheus
  - Splunk
  - Google Cloud Platform (GCP)
  - Oracle Cloud Infrastructure (OCI)

Config JSON

This file lists the TOA configurations to collect Thunder metrics or logs and enable the required cloud provider.

File Path: `/usr/toaenv/thunder-observability-agent/config.json`

Metrics

Depending on your cloud platform, configure the parameters to publish the Thunder metrics.

AWS

Configure the following parameters in the `config.json` to publish Thunder metrics to the AWS CloudWatch. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Table 35 : AWS Configuration Parameters

Parameter	Description	Default Value
`aws_provider`	Specify 1 to publish selected metric/s, logs, or both to AWS. By default, it is disabled and does not send metric to AWS. To publish metric/s it is mandatory to enable AWS as a provider.	0
`aws_metric`	Specify 1 to publish metrics to AWS CloudWatch. It sends the data only if `aws_provider` is also enabled. By default, it is disabled.	0
`aws_cpu`	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on the AWS CloudWatch. If the `aws_provider` and `aws_metrics` parameters are enabled, TOA sends this metric to the AWS CloudWatch. By default, it is enabled.	1
`aws_memory`	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on the AWS CloudWatch. By default, it is enabled.	1
`aws_disk`	Specify 1 to publish the deployed Thunder instances' storage disk usage on the AWS CloudWatch. By default, it is enabled.	1
`aws_throughput`	Specify 1 to publish the deployed Thunder instances' active throughput on the AWS CloudWatch.	1

Table 35 : AWS Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
aws_interfaces	Specify 1 to publish the deployed Thunder instances' interface down count on the AWS CloudWatch. By default, it is enabled.	1
aws_cps	Specify 1 to publish the deployed Thunder instances' new connection rate per second on the AWS CloudWatch. By default, it is enabled.	1
aws_tps	Specify 1 to publish the deployed Thunder instances' transaction rate per second on the AWS CloudWatch. By default, it is enabled.	1
aws_server_down_count	Specify 1 to publish the deployed Thunder instances' server down count on the AWS CloudWatch. By default, it is enabled.	1
aws_server_down_percentage	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on the AWS CloudWatch. By default, it is enabled.	1
aws_ssl_cert	Specify 1 to publish the deployed Thunder instances' SSL cert error count on the AWS CloudWatch. By default, it is enabled.	1
aws_server_error	Specify 1 to publish the deployed	1

Table 35 : AWS Configuration Parameters

Parameter	Description	Default Value
	Thunder instances web/app servers 4xx, 5xx errors count on the AWS CloudWatch. By default, it is enabled.	
`aws_sessions`	Specify 1 to publish the deployed Thunder instances' active session count on the AWS CloudWatch. By default, it is enabled.	1
`aws_packet_rate`	Specify 1 to publish the deployed Thunder instances' packet rate on the AWS CloudWatch. By default, it is enabled.	1
`aws_packet_drop`	Specify 1 to publish the deployed Thunder instances' packet drop count on the AWS CloudWatch. By default, it is enabled.	1

Azure

Configure the following parameters in the `config.json` to publish Thunder metrics to the Azure Application Insights. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Table 36 : Azure Configuration Parameters

Parameter	Description	Default Value
`azure_provider`	Specify 1 to publish selected metric/s, logs, or both to Azure.	0

Table 36 : Azure Configuration Parameters

Parameter	Description	Default Value
	By default, it is disabled and does not send metrics to Azure. To publish metric/s it is mandatory to enable Azure as a provider.	
`azure_metric`	Specify 1 to send metrics to Azure Application Insights. It sends the data only if `azure_provider` is also enabled. By default, it is disabled.	`0`
`azure_metric_resource_id`	Specifies the Azure Application Insights resource ID. To get this value, go to **Azure Portal** > **Azure services** > **Application Insights** > *<your_Thunder_instance>* > **Properties** > **Resource ID**. **Example** `/subscriptions/07dxxxxxxxxxx/resourceGroups/<resource_group_name>/providers/microsoft.insights/components/<app-insight-name>`	*`<azure_metric_resource_id>`*
`azure_cpu`	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on the Azure Application Insights. If the `azure_provider` and `azure_metrics` parameters are enabled, TOA sends this metric to the Azure Application Insights. By default, it is enabled.	`1`
`azure_memory`	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on the Azure Application Insights.	`1`

Table 36 : Azure Configuration Parameters

Parameter	**Description**	**Default Value**
	By default, it is enabled.	
`azure_disk`	Specify 1 to publish the deployed Thunder instances' storage disk on the Azure Application Insights. By default, it is enabled.	`1`
`azure_throughput`	Specify 1 to publish the deployed Thunder instances' active throughput on the Azure Application Insights. By default, it is enabled.	`1`
`azure_interfaces`	Specify 1 to publish the deployed Thunder instances' interfaces down count on the Azure Application Insights. By default, it is enabled.	`1`
`azure_cps`	Specify 1 to publish the deployed Thunder instances' new connection per second on the Azure Application Insights. By default, it is enabled.	`1`
`azure_tps`	Specify 1 to publish the deployed Thunder instances' transaction rate per second on the Azure Application Insights. By default, it is enabled.	`1`
`azure_server_down_count`	Specify 1 to publish the deployed Thunder instances' web/app servers down count on the Azure Application Insights. By default, it is enabled.	`1`
`azure_server_down_`	Specify 1 to publish the deployed Thunder instances' configured web/app servers down	`1`

Table 36 : Azure Configuration Parameters

Parameter	Description	Default Value
percentage	percentage on the Azure Application Insights. By default, it is enabled.	
azure_ssl_cert	Specify 1 to publish the deployed Thunder instances' SSL error count on the Azure Application Insights. By default, it is enabled.	1
azure_server_error	Specify 1 to publish the deployed Thunder instances' web/app servers 4xx, 5xx errors count on the Azure Application Insights. By default, it is enabled.	1
azure_sessions	Specify 1 to publish the deployed Thunder instances' active session count on the Azure Application Insights. By default, it is enabled.	1
azure_packet_rate	Specify 1 to publish the deployed Thunder instances' packet rate on the Azure Application Insights. By default, it is enabled.	1
azure_packet_drop	Specify 1 to publish the deployed Thunder instances' packet drop count on the Azure Application Insights. By default, it is enabled.	1

VMware

Configure the following parameters in the `config.json` to publish Thunder metrics to the VMware vROps. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Table 37 : VMware Configuration Parameters

Parameter	Description	Default Value
`vmware_provider`	Specify 1 to publish selected metric/s, logs, or both to VMware. By default, it is disabled and does not send metric to VMware. To publish metric/s it is mandatory to enable VMware as a provider.	`0`
`vmware_metric`	Specify 1 to publish the metrics to VMware vROps. It sends the data only if `vmware_provider` is also enabled. By default, it is disabled.	`0`
`vmware_vrops_host`	Specifies the VMware vROps host IP address. To get the host, go to **ESXi** host > **Virtual Machines** > *<your_vROps_VM>* > **Networking** > **IP Address**.	*`<vmware_vrops_host_or_ip>`*
`vmware_cpu`	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on the VMware vROps. If the `vmware_provider` and `vmware_metrics` parameters are enabled, TOA sends this metric to the VMware vROps. By default, it is enabled.	`1`
`vmware_memory`	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on the VMware vROps. By default, it is enabled.	`1`
`vmware_disk`	Specify 1 to publish the deployed	`1`

Table 37 : VMware Configuration Parameters

Parameter	Description	Default Value
	Thunder instances' storage disk on the VMware vROps. By default, it is enabled.	
`vmware_ throughput`	Specify 1 to publish the deployed Thunder instances' active throughput on the VMware vROps. By default, it is enabled.	1
`vmware_ interfaces`	Specify 1 to publish the deployed Thunder instances' interfaces down count on the VMware vROps. By default, it is enabled.	1
`vmware_cps`	Specify 1 to publish the deployed Thunder instances' new connections per second on the VMware vROps. By default, it is enabled.	1
`vmware_tps`	Specify 1 to publish the deployed Thunder instances' transaction rate per second on the VMware vROps. By default, it is enabled.	1
`vmware_server_ down_count`	Specify 1 to publish the deployed Thunder instances' web/app servers down count on the VMware vROps. By default, it is enabled.	1
`vmware_server_ down_percentage`	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on the VMware vROps. By default, it is enabled.	1

Table 37 : VMware Configuration Parameters

Parameter	Description	Default Value
`vmware_ssl_cert`	Specify 1 to publish the deployed Thunder instances' SSL error count on the VMware vROps. By default, it is enabled.	1
`vmware_server_error`	Specify 1 to publish the deployed Thunder instances' web/app servers 4xx, 5xx errors count on the VMware vROps. By default, it is enabled.	1
`vmware_sessions`	Specify 1 to publish the deployed Thunder instances' active session count on the VMware vROps. By default, it is enabled.	1
`vmware_packet_rate`	Specify 1 to publish the deployed Thunder instances' packet rate on the VMware vROps. By default, it is enabled.	1
`vmware_packet_drop`	Specify 1 to publish the deployed Thunder instances' packet drop count on the VMware vROps. By default, it is enabled.	1

Elasticsearch

Configure the following parameters in the `config.json` to publish Thunder metrics to Elasticsearch. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Table 38 : Elasticsearch Configuration Parameters

Parameter	Description	Default Value
`es_provider`	Specify 1 to publish selected metric/s, logs, or both to Elasticsearch. By default, it is disabled and does not send metric to Elasticsearch. To publish metric/s it is mandatory to enable Elasticsearch as a provider.	`0`
`es_metric`	Specify 1 to publish the metrics to Elasticsearch. It sends the data only if `es_provider` is also enabled. By default, it is disabled.	`0`
`es_host`	Specify the Elasticsearch host IP address.	*<host/ip:port>*
`es_cpu`	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on Elasticsearch. If the `es_provider` and `es_metrics` parameters are enabled, TOA sends this metric to Elasticsearch. By default, it is enabled.	`1`
`es_memory`	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on Elasticsearch. By default, it is enabled.	`1`
`es_disk`	Specify 1 to publish the deployed Thunder instances' storage disk on Elasticsearch. By default, it is enabled.	`1`
`es_throughput`	Specify 1 to publish the deployed Thunder instances' active throughput	`1`

Table 38 : Elasticsearch Configuration Parameters

Parameter	Description	Default Value
	on Elasticsearch. By default, it is enabled.	
es_interfaces	Specify 1 to publish the deployed Thunder instances' interfaces down count on Elasticsearch. By default, it is enabled.	1
es_cps	Specify 1 to publish the deployed Thunder instances' new connections per second on Elasticsearch. By default, it is enabled.	1
es_tps	Specify 1 to publish the deployed Thunder instances' transaction rate per second on Elasticsearch. By default, it is enabled.	1
es_server_down_count	Specify 1 to publish the deployed Thunder instances' web/app servers down count on Elasticsearch. By default, it is enabled.	1
es_server_down_percentage	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on Elasticsearch. By default, it is enabled.	1
es_ssl_cert	Specify 1 to publish the deployed Thunder instances' SSL error count on Elasticsearch. By default, it is enabled.	1
es_server_error	Specify 1 to publish the deployed	1

Table 38 : Elasticsearch Configuration Parameters

Parameter	Description	Default Value
	Thunder instances' web/app servers 4xx, 5xx errors count on Elasticsearch. By default, it is enabled.	
`es_sessions`	Specify 1 to publish the deployed Thunder instances' active session count on Elasticsearch. By default, it is enabled.	1
`es_packet_rate`	Specify 1 to publish the deployed Thunder instances' packet rate on Elasticsearch. By default, it is enabled.	1
`es_packet_drop`	Specify 1 to publish the deployed Thunder instances' packet drop count on Elasticsearch. By default, it is enabled.	1

PushGateway

Configure the following parameters in the `config.json` to publish Thunder metrics to the Pushgateway. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Table 39 : Pushgateway Configuration Parameters

Parameter	Description	Default Value
`pushgateway_ provider`	Specify 1 to publish selected metric/s, logs, or both to Pushgateway. By default, it is disabled and does not send metric to Pushgateway. To	0

Table 39 : Pushgateway Configuration Parameters

Parameter	Description	Default Value
	publish metric/s it is mandatory to enable Pushgateway as a provider.	
pushgateway_ metric	Specify 1 to publish the metrics to Pushgateway. It sends the data only if pushgateway_provider is also enabled. By default, it is disabled.	0
pushgateway_host	Specify the Pushgateway host IP address.	*<host/ip:port>*
pushgateway_cpu	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on Pushgateway. If the pushgateway_provider and pushgateway_metrics parameters are enabled, TOA sends this metric to the Pushgateway. By default, it is enabled.	1
pushgateway_ memory	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on Pushgateway. By default, it is enabled.	1
pushgateway_disk	Specify 1 to publish the deployed Thunder instances' storage disk on the Pushgateway. By default, it is enabled.	1
pushgateway_ throughput	Specify 1 to publish the deployed Thunder instances' active throughput on Pushgateway. By default, it is enabled.	1

Table 39 : Pushgateway Configuration Parameters

Parameter	Description	Default Value
pushgateway_interfaces	Specify 1 to publish the deployed Thunder instances' interfaces down count on Pushgateway. By default, it is enabled.	1
pushgateway_cps	Specify 1 to publish the deployed Thunder instances' new connections per second on Pushgateway. By default, it is enabled.	1
pushgateway_tps	Specify 1 to publish the deployed Thunder instances' transaction rate per second on Pushgateway. By default, it is enabled.	1
pushgateway_server_down_count	Specify 1 to publish the deployed Thunder instances' web/app servers down count on Pushgateway. By default, it is enabled.	1
pushgateway_server_down_percentage	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on Pushgateway. By default, it is enabled.	1
pushgateway_ssl_cert	Specify 1 to publish the deployed Thunder instances' SSL error count on Pushgateway. By default, it is enabled.	1
pushgateway_server_error	Specify 1 to publish the deployed Thunder instances' web/app servers 4xx, 5xx errors count on Pushgateway.	1

Table 39 : Pushgateway Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
`pushgateway_sessions`	Specify 1 to publish the deployed Thunder instances' active session count on Pushgateway. By default, it is enabled.	1
`pushgateway_packet_rate`	Specify 1 to publish the deployed Thunder instances' packet rate on Pushgateway. By default, it is enabled.	1
`pushgateway_packet_drop`	Specify 1 to publish the deployed Thunder instances' packet drop count on Pushgateway. By default, it is enabled.	1

Splunk

Configure the following parameters in the `config.json` to publish Thunder metrics to the Splunk. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Table 40 : Splunk Configuration Parameters

Parameter	Description	Default Value
`splunk_provider`	Specify 1 to publish selected metric/s, logs, or both to Splunk. By default, it is disabled and does not send metric to Splunk. To publish metric/s it is mandatory to enable Splunk as a provider.	0

Table 40 : Splunk Configuration Parameters

Parameter	Description	Default Value
splunk_metric	Specify 1 to publish the metrics to Splunk. It sends the data only if splunk_provider is also enabled. By default, it is disabled.	0
splunk_host	Specify the Splunk host IP address.	*<host/ip:port>*
splunk_cpu	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on Splunk. If the splunk_provider and splunk_metrics parameters are enabled, TOA sends this metric to the Splunk. By default, it is enabled.	1
splunk_memory	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on Splunk. By default, it is enabled.	1
splunk_disk	Specify 1 to publish the deployed Thunder instances' storage disk on Splunk. By default, it is enabled.	1
splunk_throughput	Specify 1 to publish the deployed Thunder instances' active throughput on Splunk. By default, it is enabled.	1
splunk_interfaces	Specify 1 to publish the deployed Thunder instances' interfaces down count on Splunk. By default, it is enabled.	1

Table 40 : Splunk Configuration Parameters

Parameter	Description	Default Value
splunk_cps	Specify 1 to publish the deployed Thunder instances' new connections per second on Splunk. By default, it is enabled.	1
splunk_tps	Specify 1 to publish the deployed Thunder instances' transaction rate per second on Splunk. By default, it is enabled.	1
splunk_server_down_count	Specify 1 to publish the deployed Thunder instances' web/app servers down count on Splunk. By default, it is enabled.	1
splunk_server_down_percentage	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on Splunk. By default, it is enabled.	1
splunk_ssl_cert	Specify 1 to publish the deployed Thunder instances' SSL error count on Splunk. By default, it is enabled.	1
splunk_server_error	Specify 1 to publish the deployed Thunder instances' web/app servers 4xx, 5xx errors count on Splunk. By default, it is enabled.	1
splunk_sessions	Specify 1 to publish the deployed Thunder instances' active session count on Splunk.	1

Table 40 : Splunk Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
`splunk_packet_rate`	Specify 1 to publish the deployed Thunder instances' packet rate on Splunk. By default, it is enabled.	1
`splunk_packet_drop`	Specify 1 to publish the deployed Thunder instances' packet drop count on Splunk. By default, it is enabled.	1

Google Cloud Platform (GCP)

Configure the following parameters in the `config.json` to publish Thunder metrics to the GCP. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Additionally, you must enable the **Strackdriver Monitoring API** in the Google Cloud console. To do so, navigate to **APIs & Services** > **Library** > **Strackdriver Monitoring API**, and click **Enable**.

Table 41 : GCP Configuration Parameters

Parameter	Description	Default Value
`gcp_provider`	Specify 1 to publish selected metric/s, logs, or both to GCP. By default, it is disabled and does not send metric to GCP. To publish metric/s it is mandatory to enable GCP as a provider.	0
`gcp_metric`	Specify 1 to publish the metrics to GCP. It	0

Table 41 : GCP Configuration Parameters

Parameter	Description	Default Value
	sends the data only if `gcp_provider` is also enabled. By default, it is disabled.	
`gcp_cpu`	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on GCP. If the `gcp_provider` and `gcp_metrics` parameters are enabled, TOA sends this metric to the GCP. By default, it is enabled.	`1`
`gcp_memory`	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on GCP. By default, it is enabled.	`1`
`gcp_disk`	Specify 1 to publish the deployed Thunder instances' storage disk on GCP. By default, it is enabled.	`1`
`gcp_throughput`	Specify 1 to publish the deployed Thunder instances' active throughput on GCP. By default, it is enabled.	`1`
`gcp_interfaces`	Specify 1 to publish the deployed Thunder instances' interfaces down count on GCP. By default, it is enabled.	`1`
`gcp_cps`	Specify 1 to publish the deployed Thunder instances' new connections per second on GCP.	`1`

Table 41 : GCP Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
`gcp_tps`	Specify 1 to publish the deployed Thunder instances' transaction rate per second on GCP. By default, it is enabled.	1
`gcp_server_down_count`	Specify 1 to publish the deployed Thunder instances' web/app servers down count on GCP. By default, it is enabled.	1
`gcp_server_down_percentage`	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on GCP. By default, it is enabled.	1
`gcp_ssl_cert`	Specify 1 to publish the deployed Thunder instances' SSL error count on GCP. By default, it is enabled.	1
`gcp_server_error`	Specify 1 to publish the deployed Thunder instances' web/app servers 4xx, 5xx errors count on GCP. By default, it is enabled.	1
`gcp_sessions`	Specify 1 to publish the deployed Thunder instances' active session count on GCP. By default, it is enabled.	1
`gcp_packet_rate`	Specify 1 to publish the deployed Thunder instances' packet rate on GCP.	1

Table 41 : GCP Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
`gcp_packet_drop`	Specify 1 to publish the deployed Thunder instances' packet drop count on GCP. By default, it is enabled.	1

Oracle Cloud Infrastructure (OCI)

Configure the following parameters in the `config.json` to publish Thunder metrics to the OCI. By default, all the metrics are enabled. You can enable one or more Thunder Metrics.

NOTE: For better throughput, you must enable only those metrics which are required.

Before publishing metrics in OCI, you must create and manage certain policies that define the necessary permissions. To do the same, see Create Policies to Publish Data in OCI.

Table 42 : OCI Configuration Parameters

Parameter	Description	Default Value
`oci_provider`	Specify 1 to publish selected metric/s, logs, or both to OCI. By default, it is disabled and does not send metric to OCI. To publish metric/s it is mandatory to enable OCI as a provider.	0
`oci_metric`	Specify 1 to publish the metrics to OCI. It sends	0

Table 42 : OCI Configuration Parameters

Parameter	Description	Default Value
	the data only if oci_provider is also enabled. By default, it is disabled.	
oci_cpu	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on OCI. If the oci_provider and oci_metrics parameters are enabled, TOA sends this metric to the OCI. By default, it is enabled.	1
oci_memory	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on OCI. By default, it is enabled.	1
oci_disk	Specify 1 to publish the deployed Thunder instances' storage disk on OCI. By default, it is enabled.	1

Table 42 : OCI Configuration Parameters

Parameter	Description	Default Value
`oci_throughput`	Specify 1 to publish the deployed Thunder instances' active throughput on OCI. By default, it is enabled.	1
`oci_interfaces`	Specify 1 to publish the deployed Thunder instances' interfaces down count on OCI. By default, it is enabled.	1
`oci_cps`	Specify 1 to publish the deployed Thunder instances' new connections per second on OCI. By default, it is enabled.	1
`oci_tps`	Specify 1 to publish the deployed Thunder instances' transaction rate per second on OCI. By default, it is enabled.	1
`oci_server_down_count`	Specify 1 to publish the deployed Thunder instances' web/app servers down count on OCI.	1

Table 42 : OCI Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
`oci_server_down_percentage`	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on OCI. By default, it is enabled.	1
`oci_ssl_cert`	Specify 1 to publish the deployed Thunder instances' SSL error count on OCI. By default, it is enabled.	1
`oci_server_error`	Specify 1 to publish the deployed Thunder instances' web/app servers 4xx, 5xx errors count on OCI. By default, it is enabled.	1
`oci_sessions`	Specify 1 to publish the deployed Thunder instances' active session count on OCI. By default, it is enabled.	1
`oci_packet_rate`	Specify 1 to publish the deployed Thunder	1

Table 42 : OCI Configuration Parameters

Parameter	Description	Default Value
	instances' packet rate on OCI. By default, it is enabled.	
`oci_packet_drop`	Specify 1 to publish the deployed Thunder instances' packet drop count on OCI. By default, it is enabled.	`1`
`oci_compartment_id`	Specify the compartment id, also known as Oracle Cloud Identifier (OCID), of your compartment in Oracle Cloud Infrastructure (OCI). To obtain the OCID, perform the following steps: 1. Open the OCI console and access the navigation menu. 2. Click **Identity & Security**, and under the **Identity** section, select **Compartments**. A list of	`ocid1.compartment.oc1..xxxxxxxx`

Table 42 : OCI Configuration Parameters

Parameter	Description	Default Value
	compartments that exist within your OCI tenancy will be displayed. 3. Click the compartment of your choice. The **Instance Information** tab will be displayed. 4. Under **General Information**, next to **OCID**, click **Show**. The full OCID value will be displayed. 5. Click **Copy** to copy the OCID to your clipboard and then paste it into the service request form field.	

Logs

Depending upon your cloud platform, configure the following parameters to publish the Thunder logs:

AWS

Configure the following parameters in the `config.json` to publish Thunder Logs to the AWS CloudWatch.

Table 43 : AWS Configuration Parameters

Parameter	Description	Default Value
`aws_provider`	Specify 1 to publish selected metric/s, logs, or both to AWS. By default, it is disabled and does not send logs to AWS. To publish logs it is mandatory to enable AWS as a provider.	0
`aws_log`	Specify 1 to publish the logs to AWS CloudWatch. It sends the data only if `aws_provider` is also enabled. By default, it is disabled.	0
`aws_log_group_name`	Specifies the log group name under which all logs are sent to AWS CloudWatch. To get this folder, it can be found under **AWS Management Console** > **CloudWatch** > **Logs** > <*log_group_name*>.	*<aws_log_group_name>*

For sample configuration, see Examples.

Azure

Configure the following parameters in the `config.json` to publish Thunder Logs to the Azure Log Analytics Workspace.

Table 44 : Azure Configuration Parameters

Parameter	Description	Default Value
`azure_provider`	Specify 1 to publish the selected metric/s, logs, or both to Azure. By default, it is disabled and does not send logs to Azure. To publish logs, it is mandatory to enable Azure as a provider.	0
`azure_log`	Specify 1 to publish the logs to Azure	0

Table 44 : Azure Configuration Parameters

Parameter	Description	Default Value
	Log Analytics Workspace. It sends the data only if `azure_provider` is also enabled. By default, it is disabled.	
`azure_log_workspace_id`	Specifies the Azure Log Analytics Workspace ID. To get this value, go to **Azure Portal** > **Azure services** > **Log Analytics workspaces** > *<your_log_analytics_workspace>* > **Settings** > **Agents**.	*<azure_log_workspace_id>*

For sample configuration, see Examples.

VMware

Configure the following parameters in the `config.json` to publish Thunder Logs to the VMware vRLI.

Table 45 : VMware Configuration Parameters

Parameter	Description	Default Value
`vmware_provider`	Specify 1 to publish the selected metric/s, logs, or both to VMware. By default, it is disabled and does not send logs to VMware. To publish logs, it is mandatory to enable VMware as a provider.	0
`vmware_log`	Specify 1 to publish the logs to VMware vRLI. It sends the data only if `vmware_provider` is also enabled. By default, it is disabled.	0
`vmware_vrli_host`	Specifies the VMware vRLI host IP address. To get the host, go to **ESXi** host	*<vmware_vrli_host_*

Table 45 : VMware Configuration Parameters

Parameter	Description	Default Value
	> **Virtual Machines** > *<your_vRLI_VM>* > **Networking** > **IP Address**.	*or_ip>*

For sample configuration, see Examples.

Elasticsearch

Configure the following parameters in the `config.json` to publish Thunder Logs to Elasticsearch.

Table 46 : Elasticsearch Configuration Parameters

Parameter	Description	Default Value
`es_provider`	Specify 1 to publish the selected metric/s, logs, or both to Elasticsearch. By default, it is disabled and does not send logs to Elasticsearch. To publish logs, it is mandatory to enable Elasticsearch as a provider.	0
`es_log`	Specify 1 to publish the logs to Elasticsearch. It sends the data only if `es_provider` is also enabled. By default, it is disabled.	0
`es_host`	Specify the Elasticsearch host IP address.	*<host/ip:port>*

For sample configuration, see Examples.

Prometheus

Configure the following parameters in the `config.json` to publish Thunder Logs to Pushgateway.

Table 47 : Prometheus Configuration Parameters

Parameter	Description	Default Value
`pushgateway_`	Specify 1 to publish the selected	0

Table 47 : Prometheus Configuration Parameters

Parameter	Description	Default Value
`provider`	metric/s, logs, or both to Pushgateway. By default, it is disabled and does not send logs to Pushgateway. To publish logs, it is mandatory to enable Pushgateway as a provider.	
`pushgateway_log`	Specify 1 to publish the logs to Pushgateway. It sends the data only if `pushgateway_provider` is also enabled. By default, it is disabled.	0
`pushgateway_host`	Specify the Pushgateway host IP address.	*<host/ip:port>*

For sample configuration, see Examples.

Splunk

Configure the following parameters in the `config.json` to publish Thunder Logs to Splunk.

Table 48 : Splunk Configuration Parameters

Parameter	Description	Default Value
`splunk_provider`	Specify 1 to publish the selected metric/s, logs, or both to Splunk. By default, it is disabled and does not send logs to Splunk. To publish logs, it is mandatory to enable Splunk as a provider.	0
`splunk_log`	Specify 1 to publish the logs to Splunk. It sends the data only if `splunk_provider` is also enabled.	0

Table 48 : Splunk Configuration Parameters

Parameter	Description	Default Value
	By default, it is disabled.	
splunk_host	Specify the Splunk host IP address.	*<host/ip:port>*

For sample configuration, see Examples.

Google Cloud Platform (GCP)

Configure the following parameters in the config.json to publish Thunder Logs to GCP.

Table 49 : GCP Configuration Parameters

Parameter	Description	Default Value
gcp_provider	Specify 1 to publish the selected metric/s, logs, or both to GCP. By default, it is disabled and does not send logs to GCP. To publish logs, it is mandatory to enable GCP as a provider.	0
gcp_log	Specify 1 to publish the logs to GCP. It sends the data only if gcp_provider is also enabled. By default, it is disabled.	0

Additionally, you must enable the **Cloud Logging API** in the Google Cloud console. To do so, navigate to **APIs & Services** > **Library** > **Cloud Logging API**, and click **Enable**.

For sample configuration, see Examples.

Oracle Cloud Infrastructure (OCI)

Configure the following parameters in the config.json to publish Thunder Logs to OCI.

Before publishing logs in OCI, you must create and manage certain policies that define the necessary permissions. To do the same, see Create Policies to Publish Data in OCI.

Table 50 : OCI Configuration Parameters

Parameter	Description	Default Value
`oci_provider`	Specify 1 to publish the selected metric/s, logs, or both to OCI. By default, it is disabled and does not send logs to OCI. To publish logs, it is mandatory to enable OCI as a provider.	`0`
`oci_log`	Specify 1 to publish the logs to OCI. It sends the data only if `oci_provider` is also enabled. By default, it is disabled.	`0`
`oci_log_id`	Specify the Oracle Cloud Identifier (OCID) of your log in OCI. To obtain the OCID, perform the following steps: 1. Open the OCI console, access the navigation menu, and click **Observability & Management**. 2. Under **Logging**, click **Log Groups**. 3. Under **List Scope**, select the compartment where you have the permissions.	`ocid1.log.oc1.xxxx.xxxxxxx`

Table 50 : OCI Configuration Parameters

Parameter	Description	Default Value
	4. On the **Log Groups** page, click **Create Log Group**. 5. Enter a log group name, description (optional), and click **Create**. A new log group will be created. 6. Click on the newly created log group, navigate to the **Logs** tab within the group and click **Create custom log**. 7. Enter a log name, select the log group created in the previous step and click **Create custom log**. 8. Click **Cancel** on the **Create agent configuration** page (if prompted). A new log will be created. 9. Click the newly created log. The OCID of the log will be displayed under the **Log Information** tab. 10. Click **Copy** to copy the OCID to your clipboard and then paste it into the service request form field.	

For sample configuration, see Examples.

Monitor Dashboard

This section describes how to track and monitor the health, throughput, and performance of your Thunder instances.

The following topics are covered:

Monitor Metrics

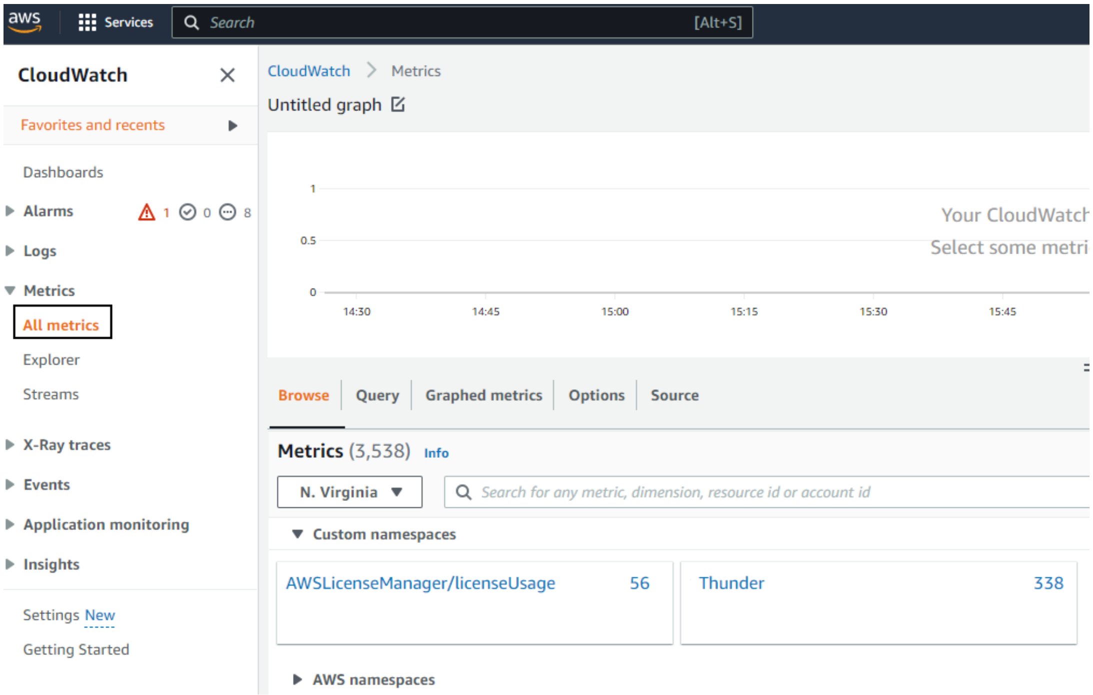
Depending on your cloud provider, the steps are provided to monitor the configured metrics.

AWS CloudWatch

To monitor the Thunder metrics on AWS CloudWatch, perform the following steps:

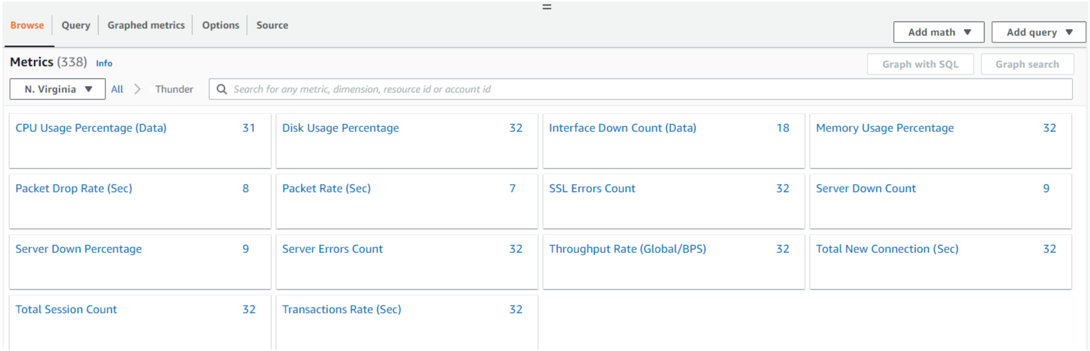
1. From the **AWS Management Console**, go to **CloudWatch** > **Metrics** > **All metrics**.

   Figure 7 : AWS All metrics



2. Select **Browse** > *<your_Thunder_metric_namespace>*.

   Figure 8 : Thunder Metrics



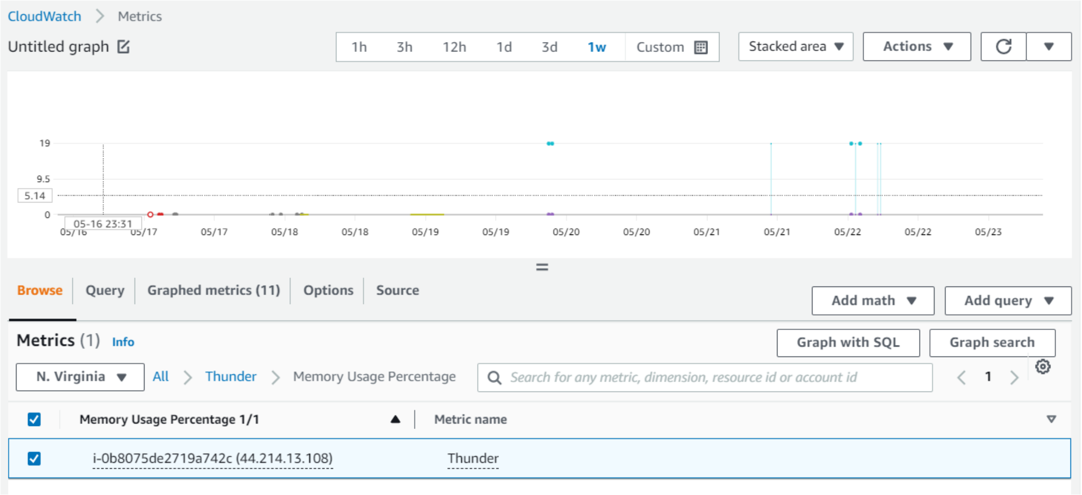
CPU Usage Percentage (Data)	31	Disk Usage Percentage	32	Interface Down Count (Data)	18	Memory Usage Percentage	32
Packet Drop Rate (Sec)	8	Packet Rate (Sec)	7	SSL Errors Count	32	Server Down Count	9
Server Down Percentage	9	Server Errors Count	32	Throughput Rate (Global/BPS)	32	Total New Connection (Sec)	32
Total Session Count	32	Transactions Rate (Sec)	32				

3. Click the required metric to be monitored from the **Metrics** panel. For the list of available Thunder metrics, see Supported Thunder Metrics.

4. Select the management IP of one or multiple Thunder instance/s to be monitored.

As the Thunder instances are selected, the metric data gets populated in the **Untitled Graph** panel for the selected the time range. For more information, see [Graph a metric](#).

Figure 9 : Selected metric graph



☑ Memory Usage Percentage 1/1	Metric name
☑ i-0b8075de2719a742c (44.214.13.108)	Thunder

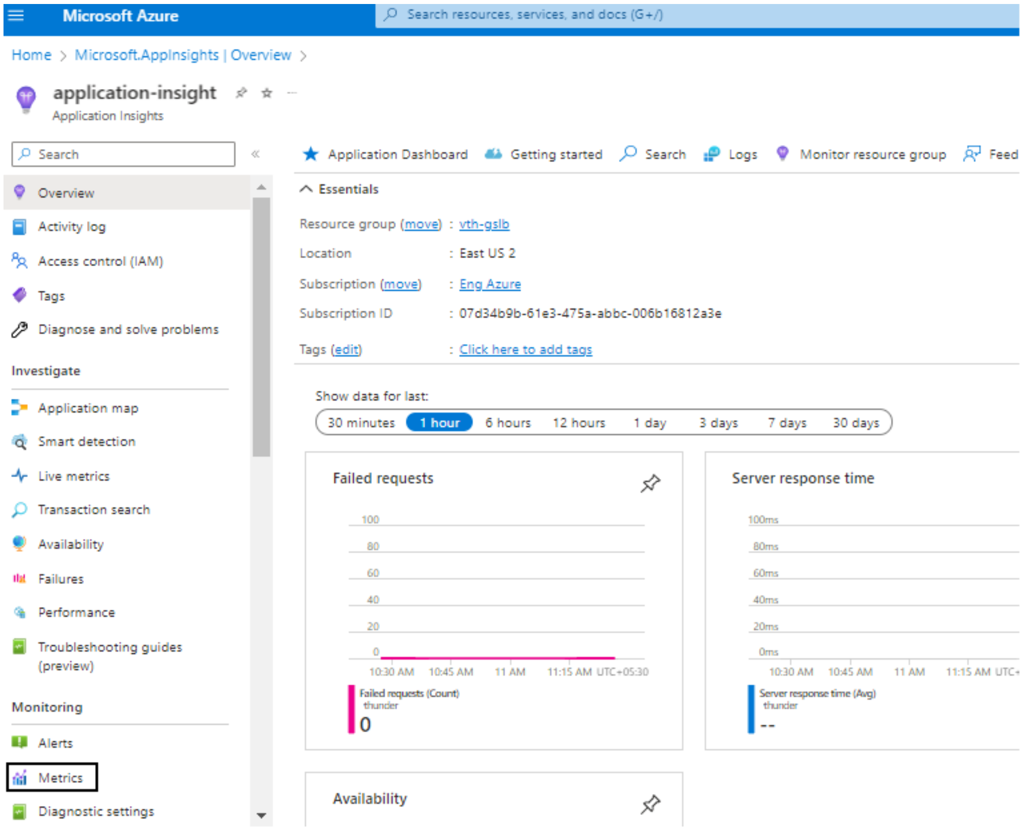
Azure Application Insight

To monitor the Thunder metrics on Azure Application Insight, perform the following steps:

1. From the **Azure Portal**, go to **Azure services** > **Resource Groups** > *<your_ resource_group>* and click your app insight name.

   The selected app insight - Overview window is displayed.

Figure 10 : Selected app insight - Overview window



Resource group (move) : vth-gslb

Location : East US 2

Subscription (move) : Eng Azure

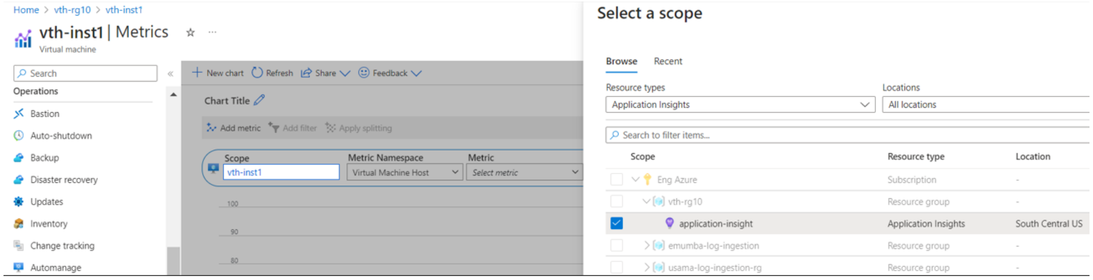
Subscription ID : 07d34b9b-61e3-475a-abbc-006b16812a3e

Tags (edit) : Click here to add tags

OR

From the **Azure Portal**, go to **Azure services** > **Resource Groups** > *<your_ resource_group>* and click your Thunder instance name whose metric is to be monitored.

Figure 11 : Thunder instance window



Scope: vth-inst1

Metric Namespace: Virtual Machine Host

Metric: Select metric

Resource types: Application Insights

Locations: All locations

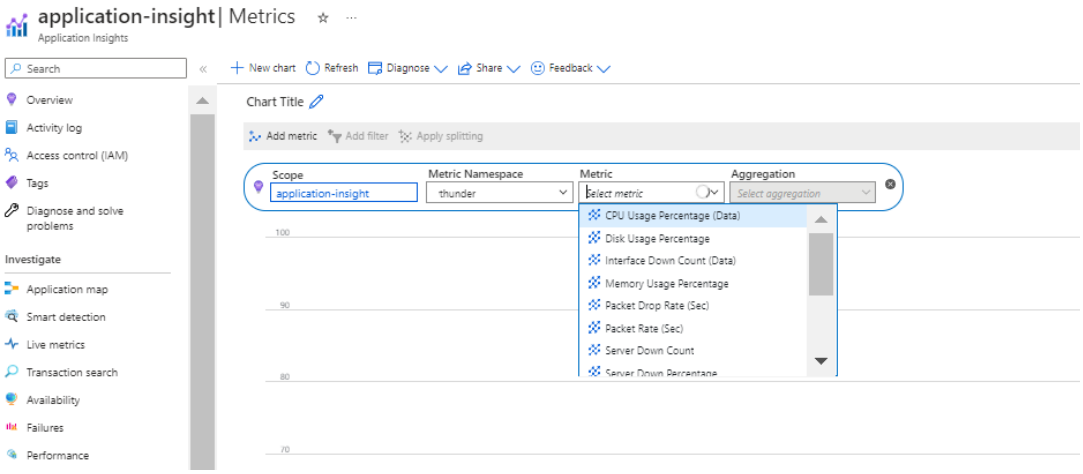
Search to filter items...

	Scope	Resource type	Location
□	Eng Azure	Subscription	-
□	vth-rg10	Resource group	-
☑	application-insight	Application Insights	South Central US
□	emumba-log-ingestion	Resource group	-
□	usama-log-ingestion-rg	Resource group	-

2. Click **Metrics** from the left **Monitoring** panel.

A scope picker is displayed in the Metric dashboard.

Figure 12 : Scope Picker



Scope: application-insight
Metric Namespace: thunder
Metric: Select metric
Aggregation: Select aggregation

- CPU Usage Percentage (Data)
- Disk Usage Percentage
- Interface Down Count (Data)
- Memory Usage Percentage
- Packet Drop Rate (Sec)
- Packet Rate (Sec)
- Server Down Count
- Server Down Percentage

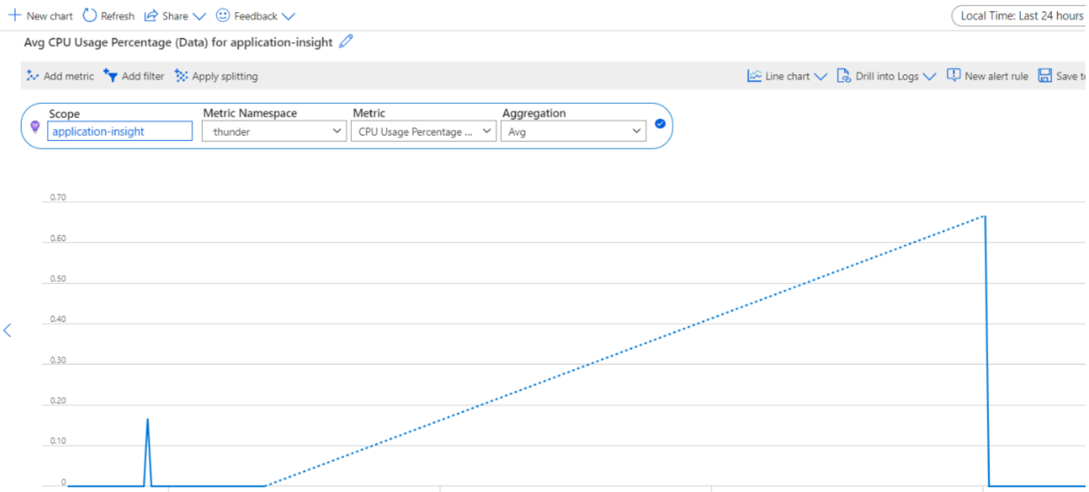
3. Select the appropriate resources whose metrics you want to view:

Table 51 : Thunder Metrics

Field Name	Description
Scope	If you are adding the metric from **Application Insight** window, the selected app insight name is auto-populated. If you are adding the metric from Thunder instance window, select your app insight name.
Metric Namespace	Select **Thunder**.
Metric	Select a metric from the drop-down. For the list of available Thunder metrics, see Supported Thunder Metrics.

As the metric is selected, the corresponding data is plotted in the chart area for the selected the time range.

Figure 13 : Plotted metric data



Avg CPU Usage Percentage (Data) for application-insight

Scope: application-insight
Metric Namespace: thunder
Metric: CPU Usage Percentage ...
Aggregation: Avg

4. To view multiple metrics on the same chart, click **Add metric** and repeat the above step. For more information, see Metrics Explorer.

VMware vROps

To monitor the Thunder metrics on vROps, perform the following steps:

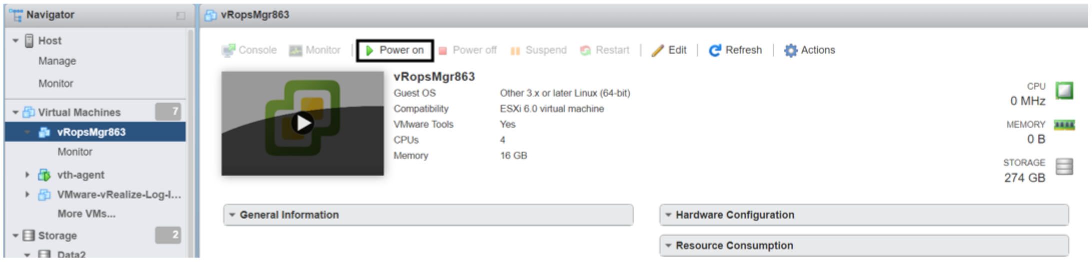
1. Start vROps VM
2. Create a Dashboard
3. Create an Alert
4. Create a Notification
5. View Thunder Metrics

Start vROps VM

To start the vROps virtual machine, perform the following steps:

1. From the **VMware ESXi** console, go to **Navigator** > **Virtual Machines** > *<your_vROps_VM>* and click **Power on**.

Figure 14 : Start vROps VM



vRopsMgr863

Guest OS	Other 3.x or later Linux (64-bit)
Compatibility	ESXi 6.0 virtual machine
VMware Tools	Yes
CPUs	4
Memory	16 GB

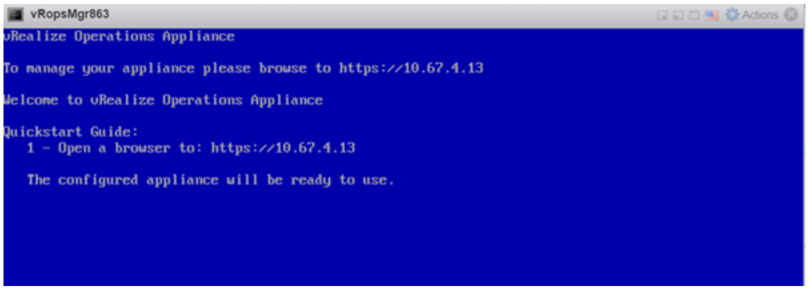
CPU 0 MHz
MEMORY 0 B
STORAGE 274 GB

NOTE: The system may take a few minutes to start the vROps virtual machine.

2. Click **Console** to launch vROps virtual machine.

   The vROps virtual machine is powered on and is reachable.

   Figure 15 : vRealize Operations Appliance

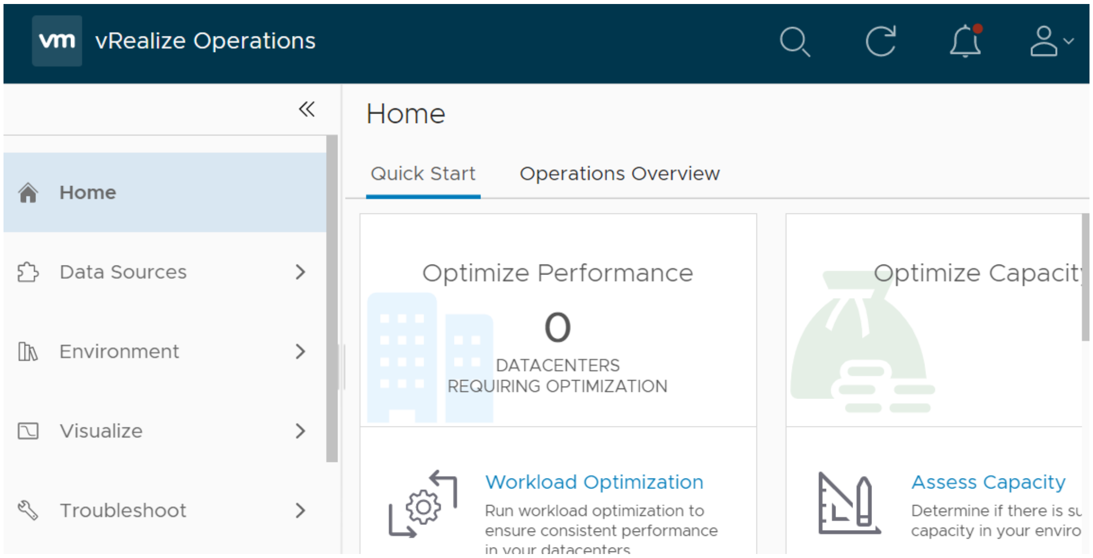


```
vRealize Operations Appliance

To manage your appliance please browse to https://10.67.4.13

Welcome to vRealize Operations Appliance

Quickstart Guide:
   1 - Open a browser to: https://10.67.4.13

   The configured appliance will be ready to use.
```

3. Log in to the **vRealize Operations Web UI** with your admin credentials.

   The vRealize Operations Home page is displayed.

Figure 16 : vRealize Operations - Home page



Create a Dashboard

The dashboard can be created using either of the following options:

- Import a dashboard template

  To import a dashboard using JSON file, see Import a Dashboard.

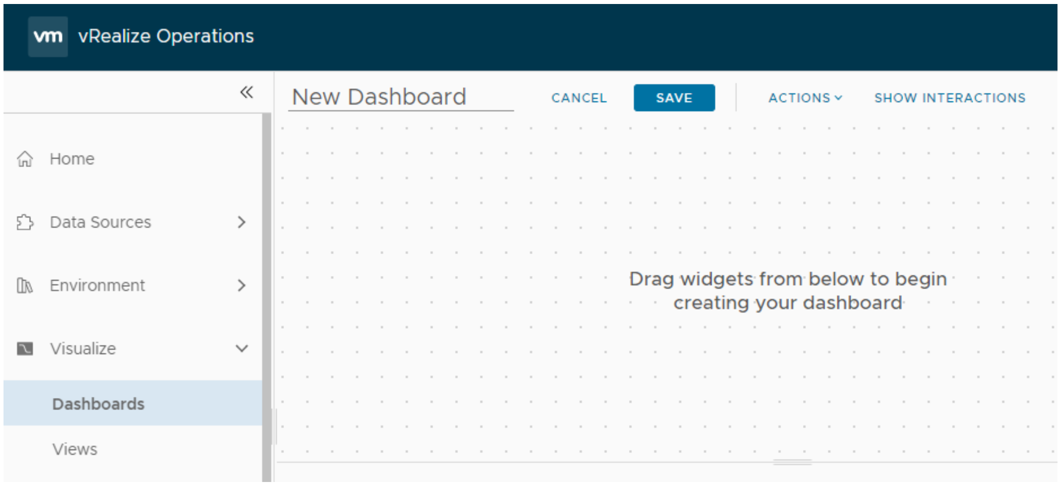
- Create a dashboard manually

  To create a dashboard manually, perform the following steps:

  1. From the **vRealize Operations Web UI**, go to **Home** > **Visualize** > **Dashboards** and click **Create** to add a new dashboard.

     The New Dashboard window is displayed.

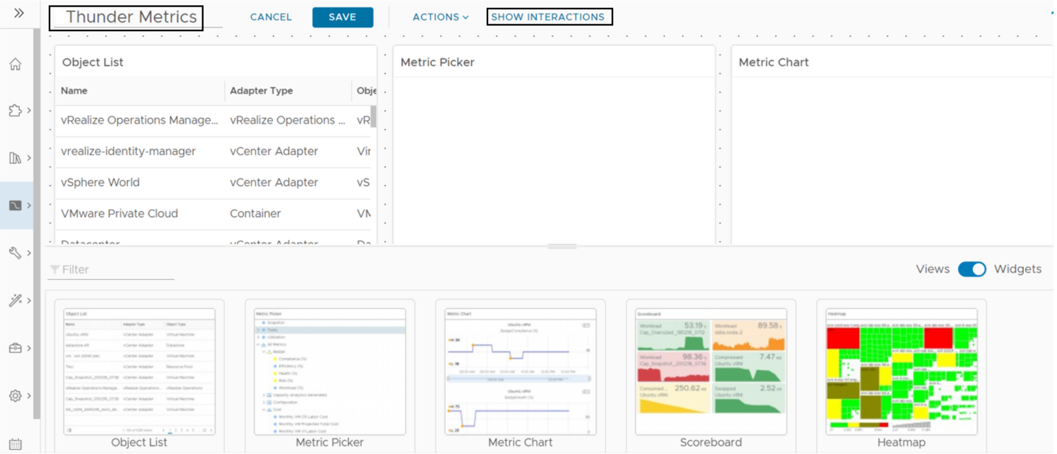
Figure 17 : New Dashboard window



Drag widgets from below to begin creating your dashboard

2. Provide a name to the new dashboard and double-click or drag the following widgets:
   - ○ Object List
   - ○ Metric Picker
   - ○ Metric Chart

Figure 18 : Dashboard widgets

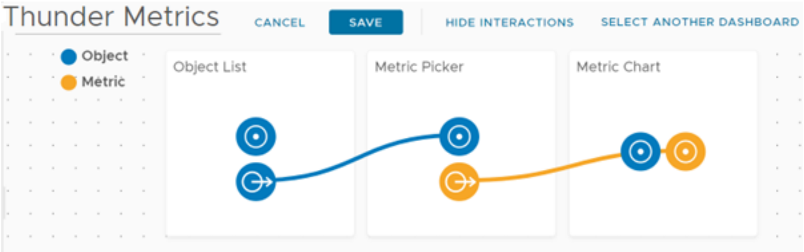


Object List

Name	Adapter Type	Obje
vRealize Operations Manage...	vRealize Operations ...	vR
vrealize-identity-manager	vCenter Adapter	Vir
vSphere World	vCenter Adapter	vS
VMware Private Cloud	Container	VN
Datacenter	vCenter Adapter	Da

3. Click **Show Interactions** to create interactions.

Figure 19 : Interactions



4. Drag the connectors and create interactions as shown in the Figure 19.
5. Click **Save** to save the changes.

   A dashboard for Thunder metrics is created.

Create an Alert

The alert definition can be created using either of the following options:

- Import an alert definition template

  To import an alert definition using XML file, see Import an Alert Definition.

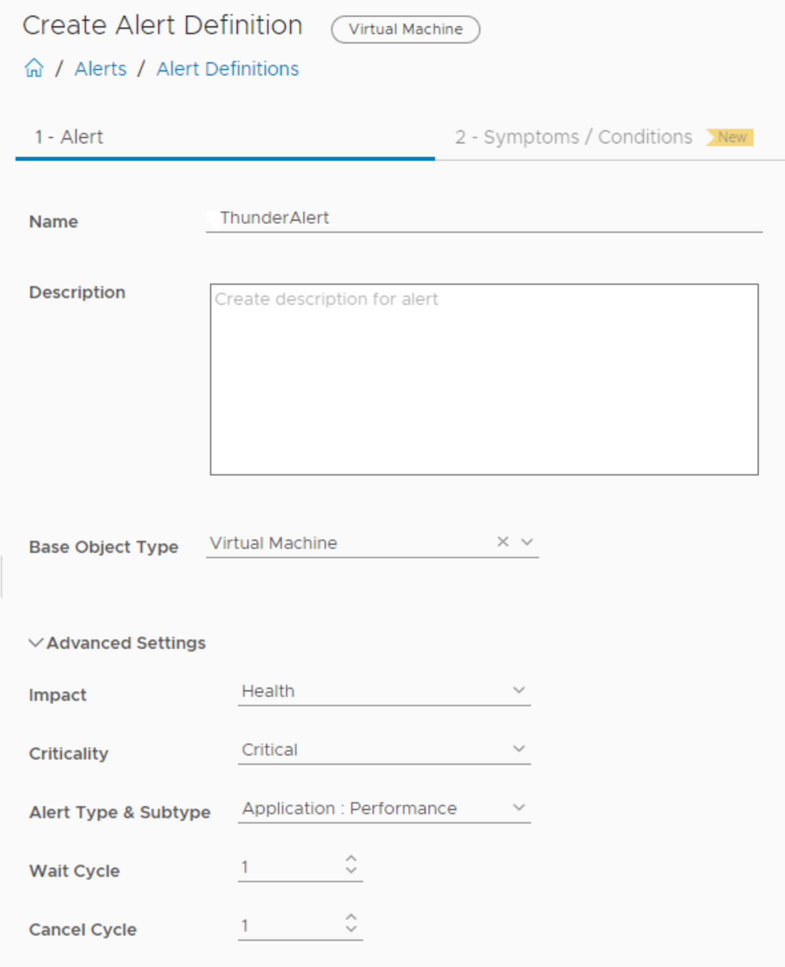
- Create an alert definition manually

  To create an alert definition manually, perform the following steps:

  1. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** and click **Alert Definitions**.
  2. Click **Add** in the **Alert Definitions** window.

     The **Create Alert Definition** panel with **Alert** tab is displayed.

Figure 20 : Create Alert Definition window



Create Alert Definition Virtual Machine

/ Alerts / Alert Definitions

1 - Alert 2 - Symptoms / Conditions New

Name: ThunderAlert

Description: Create description for alert

Base Object Type: Virtual Machine

Advanced Settings

Impact: Health

Criticality: Critical

Alert Type & Subtype: Application : Performance

Wait Cycle: 1

Cancel Cycle: 1

3. Enter or select the appropriate values in the following fields:

Table 52 : Alert tab fields

Field Name	Description
Name	Enter the alert name. **Example** In the Figure 20, the alert definition name is `ThunderAlert`.
Base Object Type	Select **vCenter Adapter** > **Virtual Machine**.

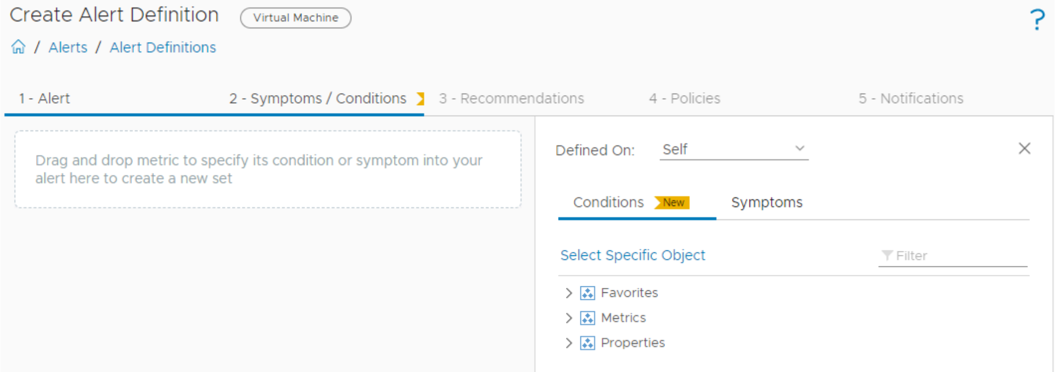
Table 52 : Alert tab fields

Field Name	Description
Under the **Advanced Settings**:	
Impact	Select **Health**.
Criticality	Select **Critical**.
Alert Type & Subtype	Select **Application : Performance**.

4. Click **Next**.

   The **Symptoms / Conditions** tab is displayed.

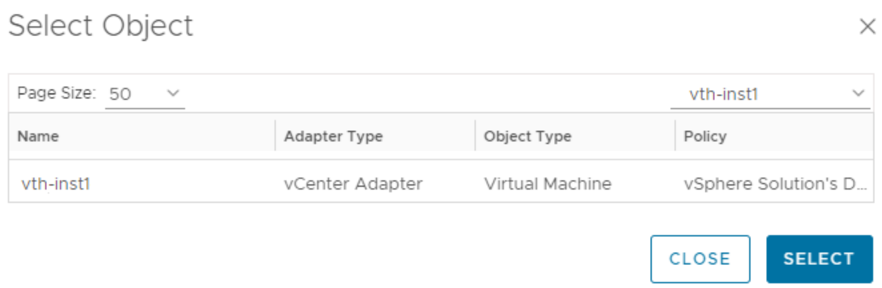
   Figure 21 : Symptoms / Conditions tab



5. Click **Select Specific Object** to select your Thunder instance.

   The **Select Object** window is displayed.

   Figure 22 : Select Object window

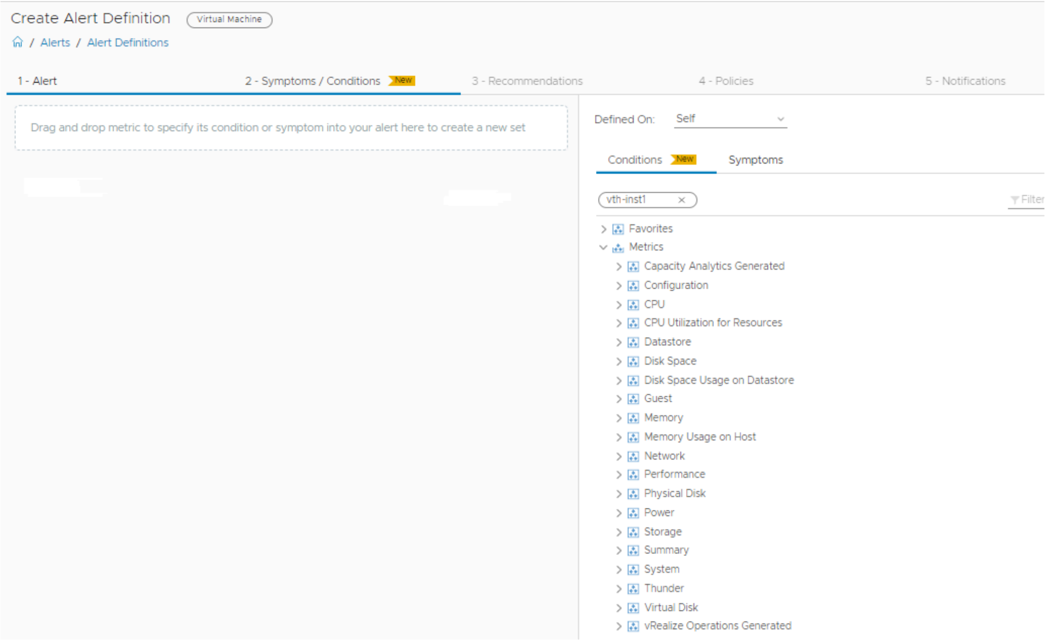


Name	Adapter Type	Object Type	Policy
vth-inst1	vCenter Adapter	Virtual Machine	vSphere Solution's D...

6. Select your Thunder instance and click **Select**.

The selected Thunder instance is listed under **Conditions**.

Figure 23 : Selected Thunder instance



Create Alert Definition (Virtual Machine)

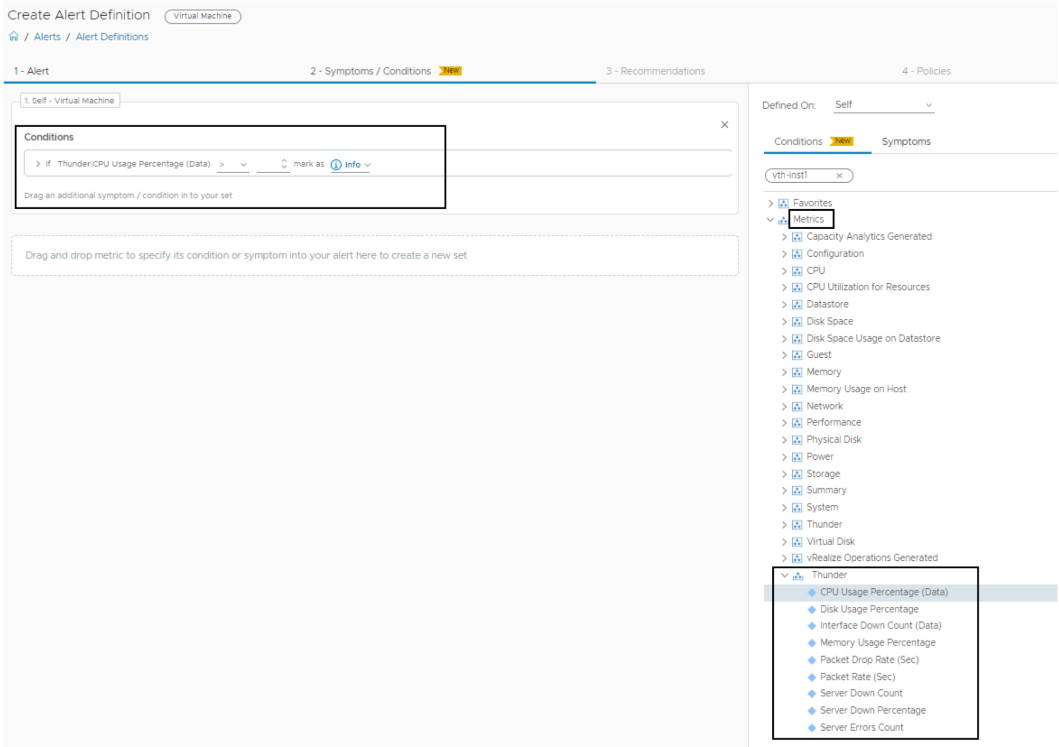
Alerts / Alert Definitions

Drag and drop metric to specify its condition or symptom into your alert here to create a new set

Defined On: Self

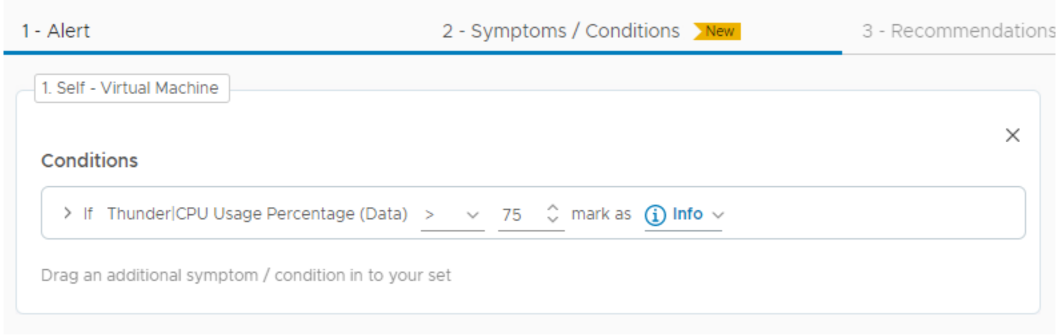
7. Select **Metrics** > **Thunder** and drag the required metrics to the left-side panel.

Figure 24 : Drag metric



8. Specify the appropriate alert condition.

Figure 25 : Alert condition

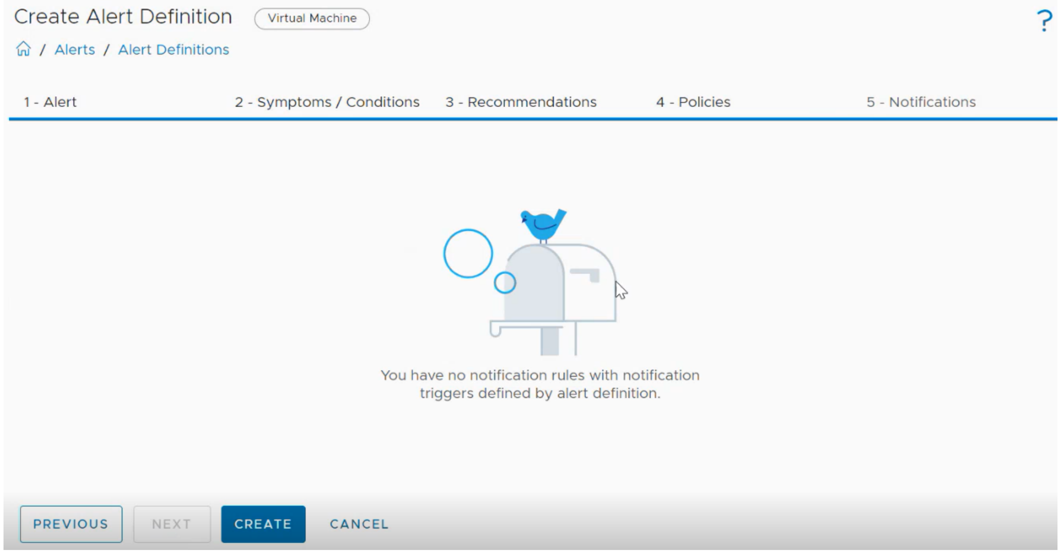

1 - Alert
2 - Symptoms / Conditions
New
3 - Recommendations
1. Self - Virtual Machine
Conditions
Drag an additional symptom / condition in to your set

If Thunder|CPU Usage Percentage (Data) > 75 mark as Info

9. Click **Next**.
10. Add the appropriate recommendations in the **Recommendations** tab, if needed.
11. Click **Next**.

12. Select appropriate policy in the **Policies** tab, if needed.
13. Click **Next**.

    The **Notification** tab is displayed. The notification can be created after the alert definition is created. For more information, see Create a Notification.

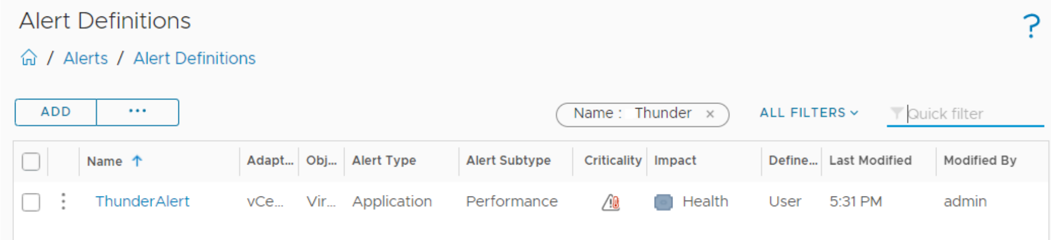


You have no notification rules with notification triggers defined by alert definition.

14. Click **Create** in the **Notification** tab.

    An alert definition is created and is listed in the **Alert Definition** window.

Figure 26 : Verify Alert Definition



□		Name ↑	Adapt...	Obj...	Alert Type	Alert Subtype	Criticality	Impact	Define...	Last Modified	Modified By
□	⋮	ThunderAlert	vCe...	Vir...	Application	Performance	⚠	Health	User	5:31 PM	admin

Create a Notification

The notification can be created using either of the following options:

- Import a notification template

  To import a notification using JSON file, see Import a Notification.

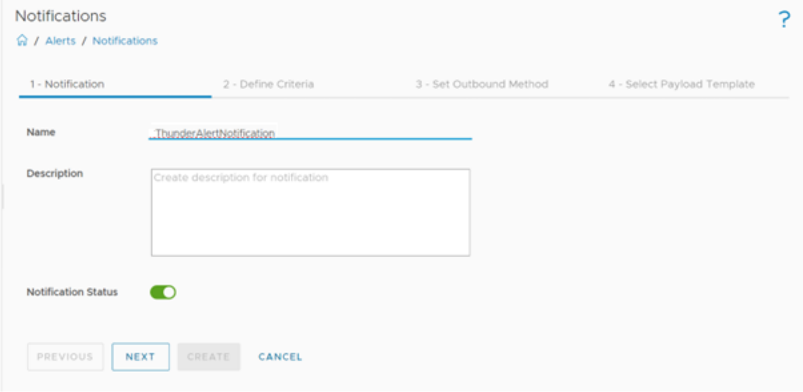
- Create a notification manually

  To create a notification manually, perform the following steps:

  1. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** and click **Notifications**.
  2. Click **Add** in the **Notifications** window.

     The **Notifications** panel with **Notification** tab is displayed.

     Figure 27 : Notifications tab

     Notifications

     Alerts / Notifications

     1 - Notification | 2 - Define Criteria | 3 - Set Outbound Method | 4 - Select Payload Template

     Name: ThunderAlertNotification

     Description: Create description for notification

     Notification Status

  3. Enter or select the appropriate values in the following fields:

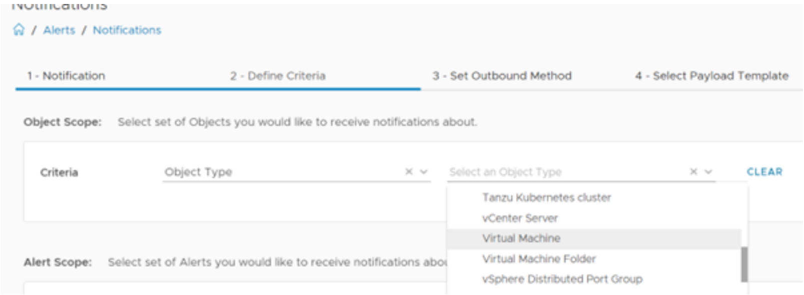
     Table 53 : Notifications tab

Field Name	Description
Name	Enter the notification name. **Example** In the Figure 27, notification name is `ThunderAlertNotification`.
Notification Status	Select **Enable**.

  4. Click **Next**.

     The **Define Criteria** tab is displayed.

Figure 28 : Define Criteria tab



Notifications

/ Alerts / Notifications

1 - Notification | 2 - Define Criteria | 3 - Set Outbound Method | 4 - Select Payload Template

Object Scope: Select set of Objects you would like to receive notifications about.

Criteria: Object Type | Select an Object Type | CLEAR

- Tanzu Kubernetes cluster
- vCenter Server
- Virtual Machine
- Virtual Machine Folder
- vSphere Distributed Port Group

Alert Scope: Select set of Alerts you would like to receive notifications abou

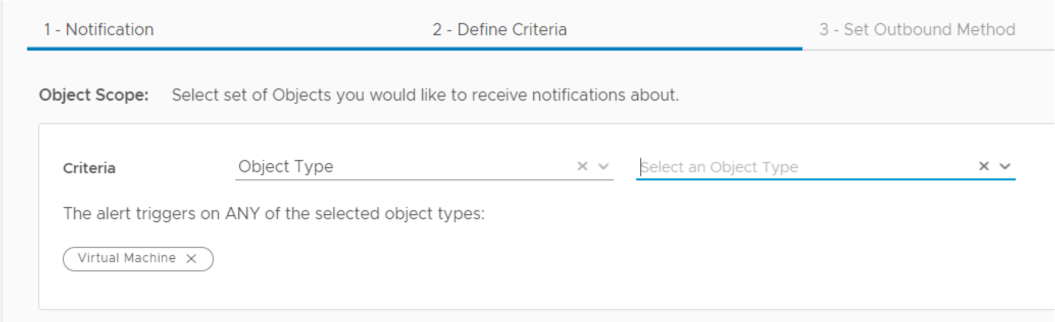
5. In the **Criteria** field, select **Object Type** from the drop-down.

   A field appears to select the object type.

6. Expand **vCenterAdapter** and select **Virtual Machine** from the drop-down.

   The selected object type is listed under **Criteria**.

   Figure 29 : Criteria defined



1 - Notification | 2 - Define Criteria | 3 - Set Outbound Method

Object Scope: Select set of Objects you would like to receive notifications about.

Criteria: Object Type | Select an Object Type

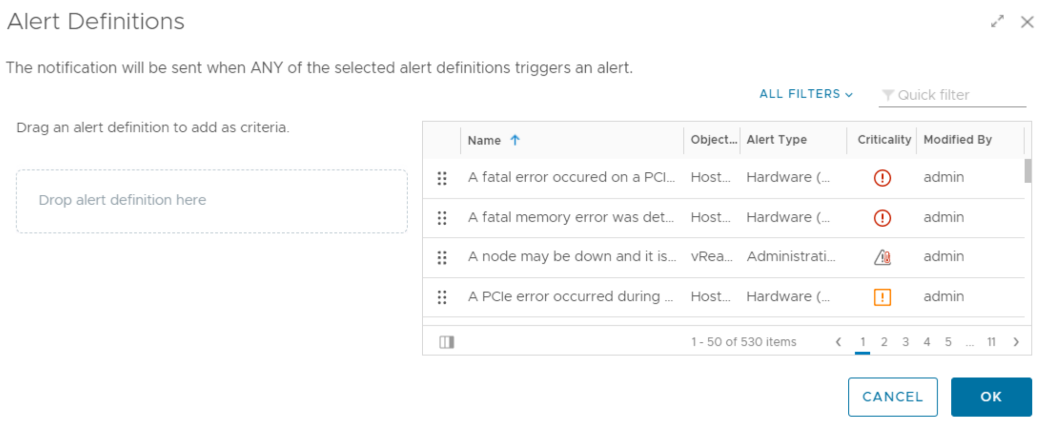
The alert triggers on ANY of the selected object types:

Virtual Machine ×

7. In the **Category** field, select **Alert Definition** from the drop-down created in the Create an Alert.

   An **Alert Definition** pop-up is displayed.

Figure 30 : Alert Definition pop-up



Alert Definitions

The notification will be sent when ANY of the selected alert definitions triggers an alert.

Drag an alert definition to add as criteria.

Drop alert definition here

ALL FILTERS ⌄ Quick filter

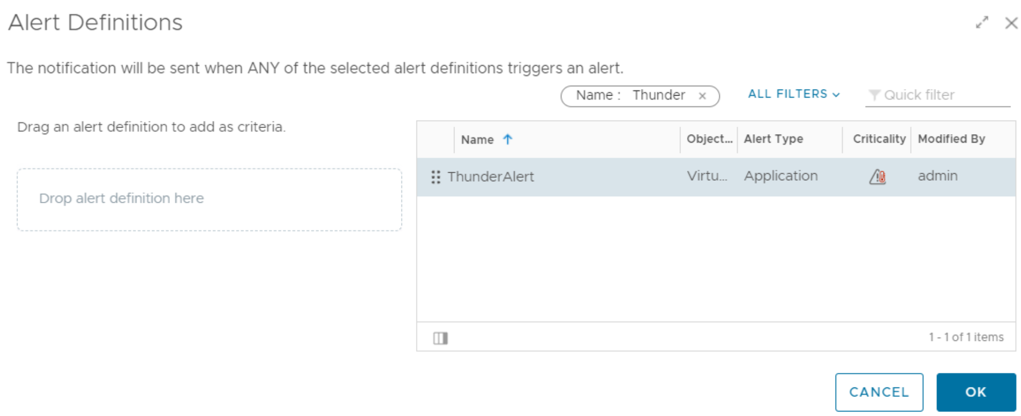
Name ↑	Object...	Alert Type	Criticality	Modified By
A fatal error occured on a PCI...	Host...	Hardware (...	!	admin
A fatal memory error was det...	Host...	Hardware (...	!	admin
A node may be down and it is...	vRea...	Administrati...	⚠	admin
A PCIe error occurred during ...	Host...	Hardware (...	!	admin

1 - 50 of 530 items ‹ 1 2 3 4 5 ... 11 ›

CANCEL OK

8. Search your alert definition.

Figure 31 : Search alert definition



Alert Definitions

The notification will be sent when ANY of the selected alert definitions triggers an alert.

Name : Thunder × ALL FILTERS ⌄ Quick filter

Drag an alert definition to add as criteria.

Drop alert definition here

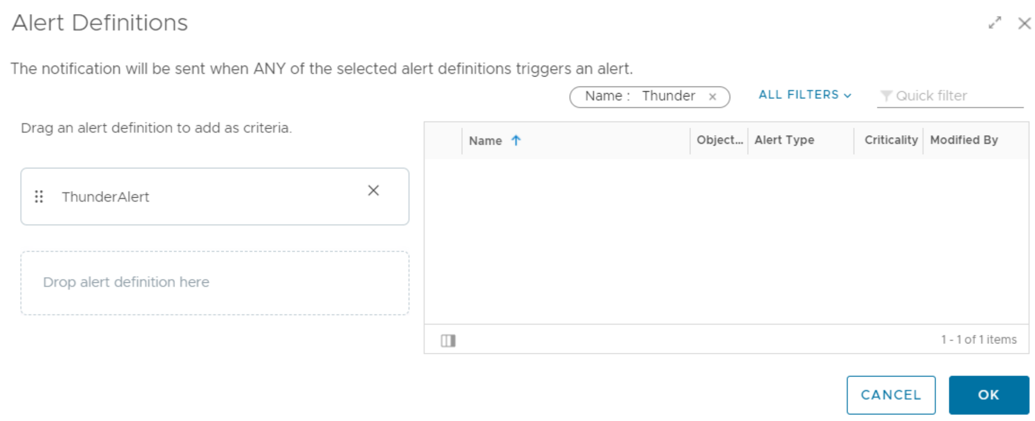
Name ↑	Object...	Alert Type	Criticality	Modified By
ThunderAlert	Virtu...	Application	⚠	admin

1 - 1 of 1 items

CANCEL OK

9. Select your alert definition and drag it to add as the criteria.

Figure 32 : Drag alert definition



Alert Definitions

The notification will be sent when ANY of the selected alert definitions triggers an alert.

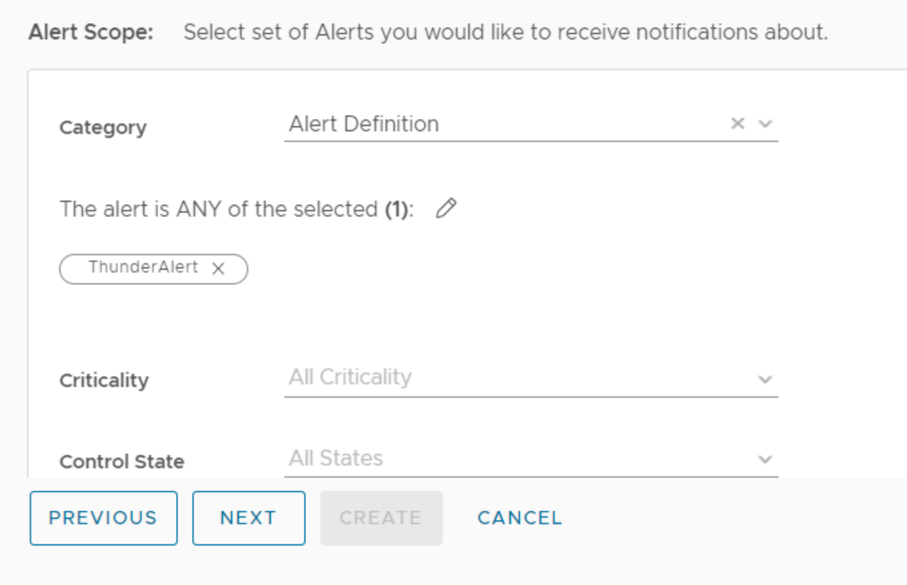
Drag an alert definition to add as criteria.

Name	Object...	Alert Type	Criticality	Modified By

10. Click **OK**.

    The selected alert definition is listed under Category.

    Figure 33 : Selected alert definition



Alert Scope: Select set of Alerts you would like to receive notifications about.

Category: Alert Definition

The alert is ANY of the selected **(1)**:

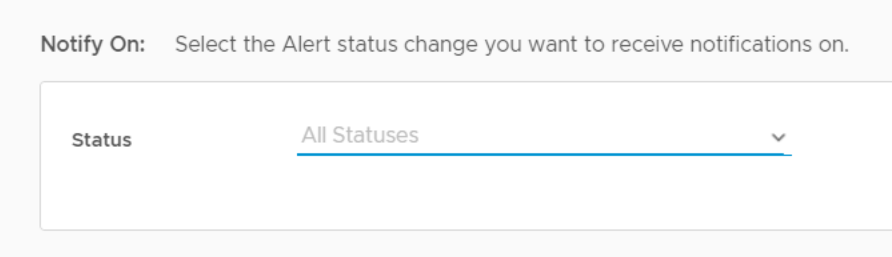
ThunderAlert

Criticality: All Criticality

Control State: All States

11. In the **Status** field under **Notify On**, select the alert status for which you want to receive the notifications.

Figure 34 : Notify On



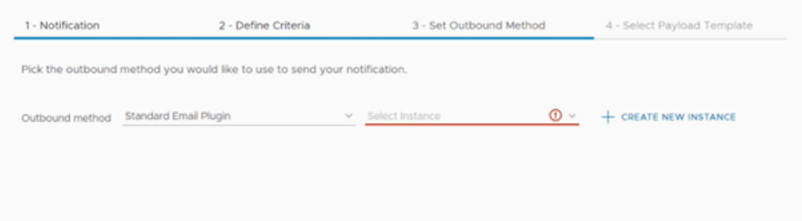
Notify On: Select the Alert status change you want to receive notifications on.

Status: All Statuses

12. Click **Next**.

    The **Set Outbound Method** tab is displayed.

    Figure 35 : Set Outbound Method tab

    Pick the outbound method you would like to use to send your notification.

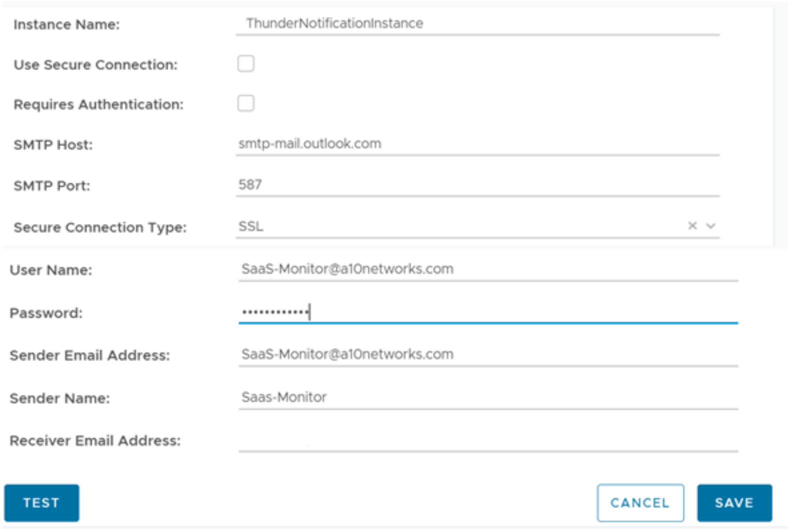
    Outbound method: Standard Email Plugin

    Select Instance

13. In the **Outbound method** field, select **Standard Email Plugin** from the drop-down list.
14. Click **Create New Instance** to create a new instance for corresponding Outbound method.

    The fields for creating a new instance are displayed.

Figure 36 : Create New Instance fields



Instance Name: ThunderNotificationInstance

Use Secure Connection: □

Requires Authentication: □

SMTP Host: smtp-mail.outlook.com

SMTP Port: 587

Secure Connection Type: SSL

User Name: SaaS-Monitor@a10networks.com

Password: ••••••••••••

Sender Email Address: SaaS-Monitor@a10networks.com

Sender Name: Saas-Monitor

Receiver Email Address:

15. Enter or select the appropriate values in the following fields:

Table 54 : Create New Instance

Field Name	Description
Instance Name	Enter the notification instance name. **Example** In the Figure 36, the notification instance name is `ThunderNotificationInstance`.
SMTP Host	Enter the URL or IP address of the email host server.
SMTP Port	Enter the SMTP port number used to connect with the email host server.
Secure Connection Type	Select **SSL**.
User Name	Enter the username that is used to connect to the email server.
Password	Enter the password for the connection username that appears on the notification message.

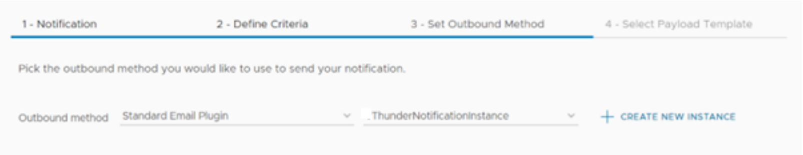
Table 54 : Create New Instance

Field Name	Description
Sender Email Address	Enter the email address of the sender.
Sender Name	Enter the display name of the sender email address.
Receiver Email Address	Enter the email address of the receiver that receives the notification.

16. Click **Save** to save the changes.

    The new instance is populated in the **Select Instance** field.

    Figure 37 : Selected New Instance



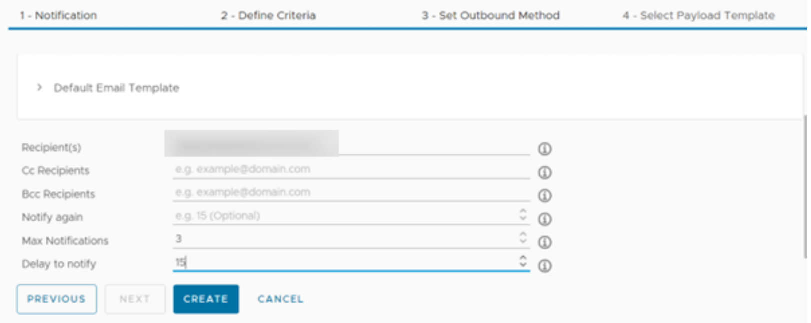
Pick the outbound method you would like to use to send your notification.

Outbound method: Standard Email Plugin | ThunderNotificationInstance

17. Click **Next**.

    The **Select Payload Template** tab is displayed.

    Figure 38 : Select Payload Template tab



Recipient(s): [redacted]
Cc Recipients: e.g. example@domain.com
Bcc Recipients: e.g. example@domain.com
Notify again: e.g. 15 (Optional)
Max Notifications: 3
Delay to notify: 15

18. Enter or select the appropriate values in the following fields for the default template:

Table 55 : Select Payload Template tab

Field Name	Description
Recipient(s)	Enter the email addresses of the recipient to

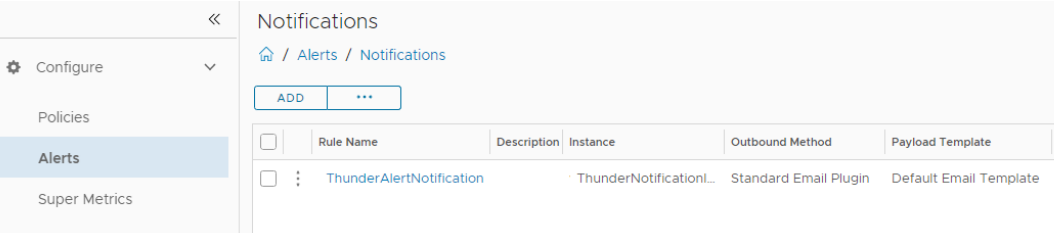
Table 55 : Select Payload Template tab

Field Name	**Description**
	receive the notification.
Max Notifications	Enter the maximum number of notification to be sent for the active alert.
Delay to notify	Enter the delay time in minutes before sending a notification when a new alert is generated.

19. Click **Create**.

    A new notification is created for the selected alert definition and it is listed in the **Notifications** window.

    Figure 39 : Verify Notification



□		Rule Name	Description	Instance	Outbound Method	Payload Template
□	⋮	ThunderAlertNotification		ThunderNotificationI...	Standard Email Plugin	Default Email Template

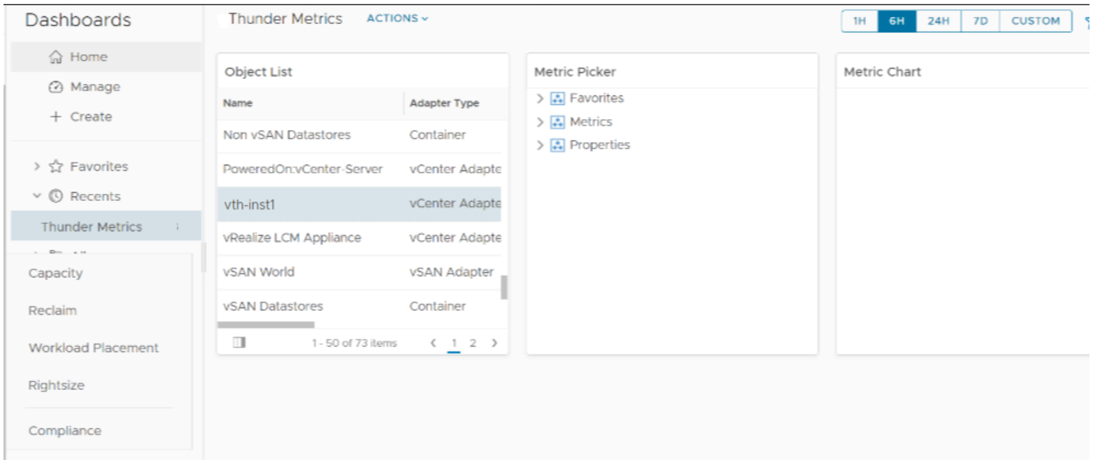
View Thunder Metrics

To view the Thunder metrics, perform the following steps:

1. From the **vRealize Operations Web UI**, go to **Home** > **Visualize** > **Dashboard** and select your dashboard created for Thunder metrics.

   The selected dashboard is displayed.

Figure 40 : Selected dashboard



Object List

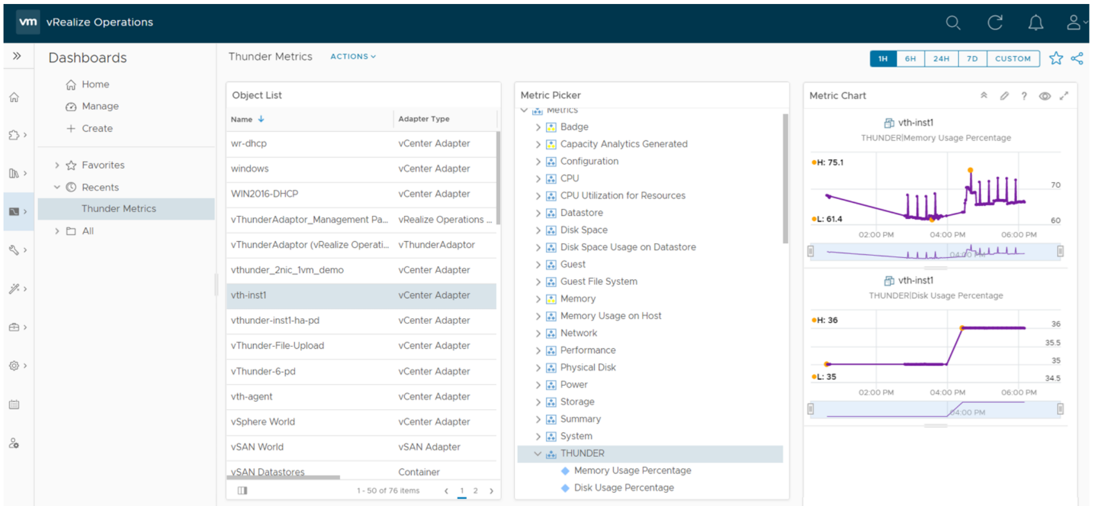
Name	Adapter Type
Non vSAN Datastores	Container
PoweredOn:vCenter-Server	vCenter Adapte
vth-inst1	vCenter Adapte
vRealize LCM Appliance	vCenter Adapte
vSAN World	vSAN Adapter
vSAN Datastores	Container

1 - 50 of 73 items 1 2

2. From **Object List**, double-click your Thunder instance.
3. From **Metric Picker**, expand **Metrics** > **THUNDER** and double-click the following common metrics:
   - Memory Usage Percentage
   - Disk Usage Percentage

   As the metric is selected, the corresponding data gets populated in the **Metric Chart** panel for the selected the time range.

Figure 41 : THUNDER Dashboard



Object List

Name	Adapter Type
wr-dhcp	vCenter Adapter
windows	vCenter Adapter
WIN2016-DHCP	vCenter Adapter
vThunderAdaptor_Management Pa...	vRealize Operations ...
vThunderAdaptor (vRealize Operati...	vThunderAdaptor
vthunder_2nic_1vm_demo	vCenter Adapter
vth-inst1	vCenter Adapter
vthunder-inst1-ha-pd	vCenter Adapter
vThunder-File-Upload	vCenter Adapter
vThunder-6-pd	vCenter Adapter
vth-agent	vCenter Adapter
vSphere World	vCenter Adapter
vSAN World	vSAN Adapter
vSAN Datastores	Container

1 - 50 of 76 items 1 2

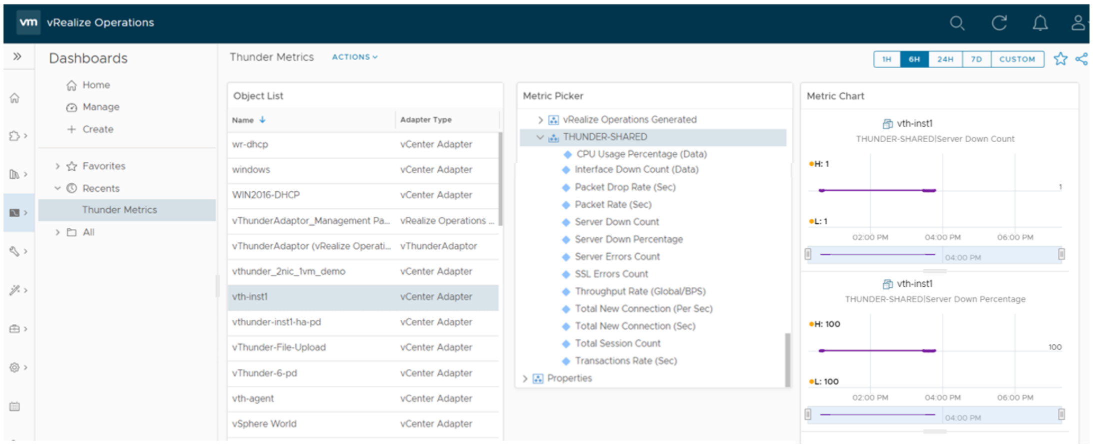
4. From **Metric Picker**, expand **Metrics** > **THUNDER-SHARED** or **THUNDER-Px** and

double-click the following metrics:

- CPU Usage Percentage (Data)
- Throughput Rate (Global/BPS)
- Interface Down Count (Data)
- Total New Connection (Sec)
- Transactions Rate (Sec)
- Server Down Count
- Server Down Percentage
- SSL Errors Count
- Server Errors Count
- Total Session Count
- Packet Rate (Sec)
- Packet Drop Rate (Sec)

As the metric is selected, the corresponding data gets populated in the **Metric Chart** panel for the selected the time range.

Figure 42 : THUNDER-SHARED Dashboard



Object List

Name	Adapter Type
wr-dhcp	vCenter Adapter
windows	vCenter Adapter
WIN2016-DHCP	vCenter Adapter
vThunderAdaptor_Management Pa...	vRealize Operations ...
vThunderAdaptor (vRealize Operati...	vThunderAdaptor
vthunder_2nic_1vm_demo	vCenter Adapter
vth-inst1	vCenter Adapter
vthunder-inst1-ha-pd	vCenter Adapter
vThunder-File-Upload	vCenter Adapter
vThunder-6-pd	vCenter Adapter
vth-agent	vCenter Adapter
vSphere World	vCenter Adapter

To view multiple metrics data, select each of those metrics. The data corresponding to each metric is displayed in the **Metric Chart** panel. For the list of available Thunder metrics, see Supported Thunder Metrics.

Kibana (Elasticsearch)

To monitor the Thunder metrics on Kibana UI, perform the following steps:

1. Import the Kibana dashboard.

   To import the Kibana dashboard, perform the following steps:

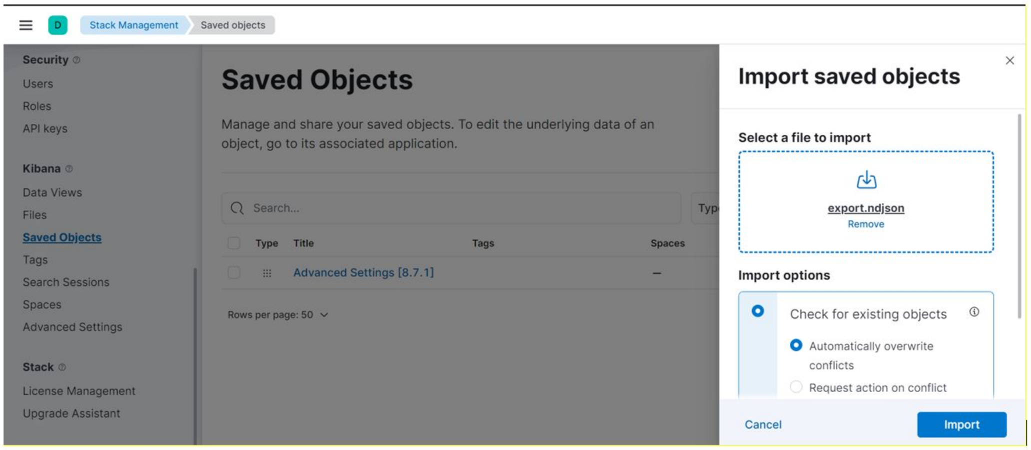
   a. Download the dashboard-template JSON file.

   b. Log in to Kibana.

   c. Navigate to **Menu** > **Management** > **Saved Objects** > **Import**.

   d. Select the downloaded Kibana dashboard file and click **Import**.

   Figure 43 : Importing Dashboard File



Saved Objects

Manage and share your saved objects. To edit the underlying data of an object, go to its associated application.

Type	Title	Tags	Spaces
	Advanced Settings [8.7.1]		–

Import saved objects

Select a file to import

export.ndjson

Import options

● Check for existing objects

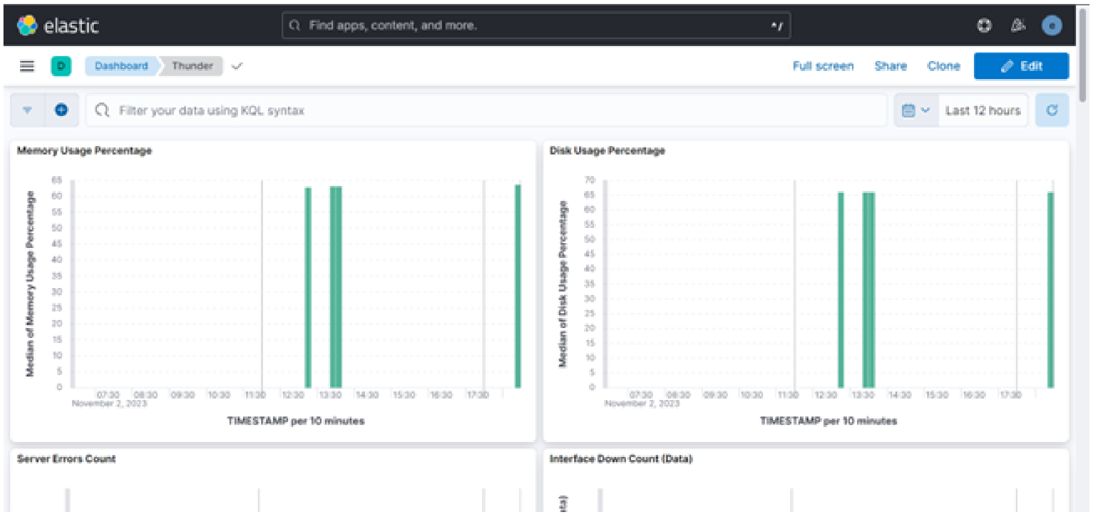
● Automatically overwrite conflicts

○ Request action on conflict

2. View the Metrics.

   To view the metrics, navigate to **Menu** > **Dashboard**. All the metrics are displayed as shown below:

Figure 44 : Thunder Dashboard



3. View the Metric Hits.

   To view all the metric hits along with meta field details, navigate to **Menu** > **Discover** > **Thunder-Metrics**.

Grafana (Prometheus)

To monitor the Thunder metrics on Grafana UI, perform the following steps:

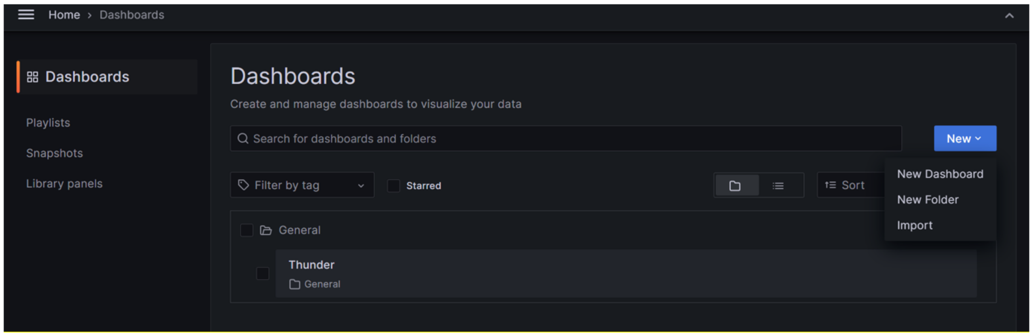
1. Import the Grafana dashboard.

   a. Download the dashboard-template JSON file.

   b. Log in to Grafana.

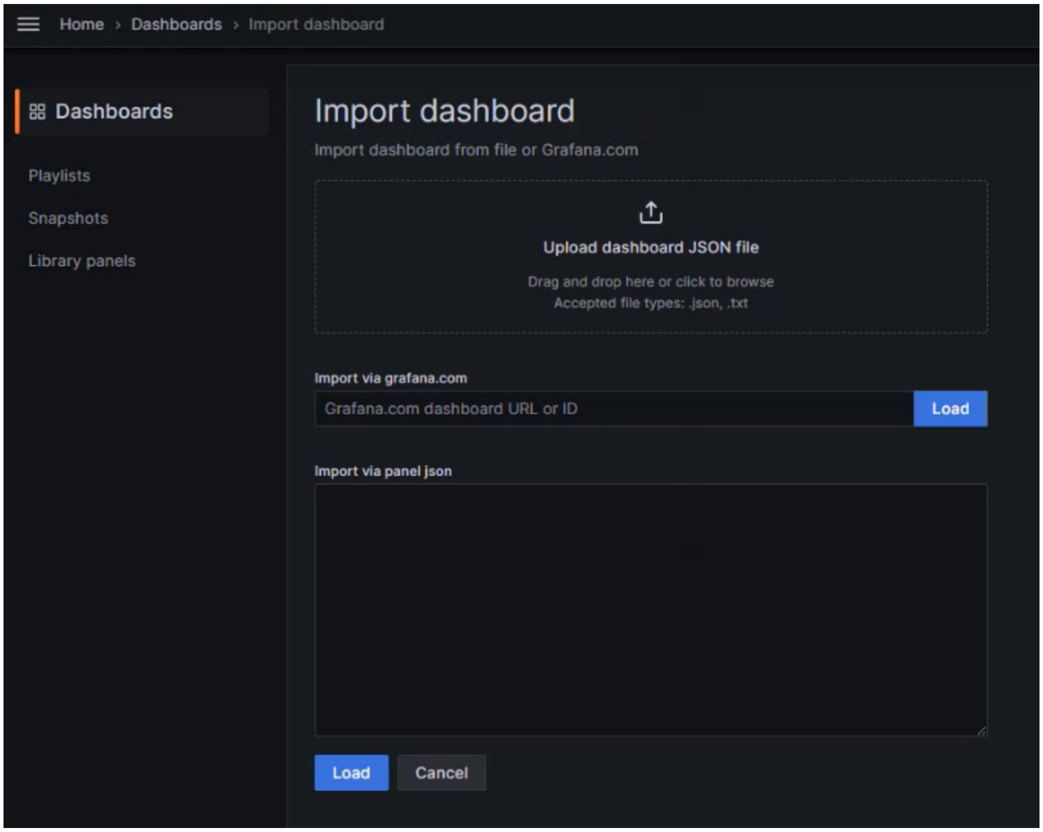
   c. Navigate to **Menu** > **Dashboard** and click **New** > **Import**.

   Figure 45 : Dashboards



d. On the **Import Dashboard** page, click **Upload dashboard JSON file**.

Figure 46 : Import dashboard

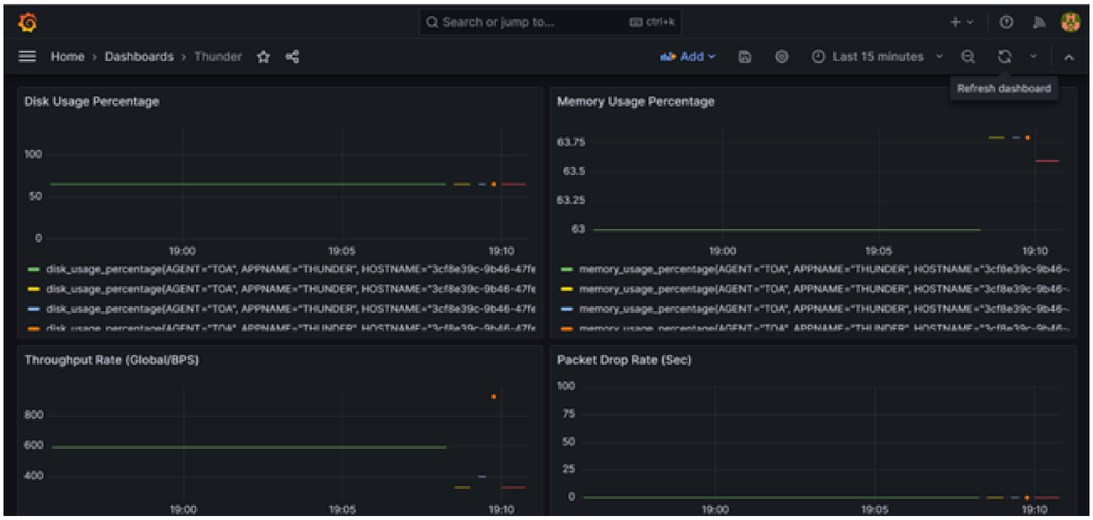

Home › Dashboards › Import dashboard
Dashboards
Playlists
Snapshots
Library panels
Import dashboard
Import dashboard from file or Grafana.com
Upload dashboard JSON file
Drag and drop here or click to browse
Accepted file types: .json, .txt
Import via grafana.com
Grafana.com dashboard URL or ID
Load
Import via panel json
Load
Cancel

e. Browse the downloaded Grafana dashboard file and click **Load**.

2. View the dashboard.

   To view the dashboard, navigate to **Menu** > **Dashboard**. All the metrics are displayed as shown below:

Figure 47 : Grafana Metrics Dashboard


Search or jump to...
ctrl+k
Home › Dashboards › Thunder
Add
Last 15 minutes
Refresh dashboard
Disk Usage Percentage
100
50
0
19:00
19:05
19:10
disk_usage_percentage{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe
disk_usage_percentage{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe
disk_usage_percentage{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe
Memory Usage Percentage
63.75
63.5
63.25
63
19:00
19:05
19:10
memory_usage_percentage{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-
memory_usage_percentage{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-
memory_usage_percentage{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-
Throughput Rate (Global/BPS)
800
600
400
19:00
19:05
19:10
Packet Drop Rate (Sec)
100
75
50
25
0
19:00
19:05
19:10

Splunk

To monitor the Thunder metrics in Splunk Enterprise, perform the following steps:

1. Log in to Splunk Enterprise.
2. Create an HTTP Event Collector (HEC) for the metrics.

   To use HEC, you need to configure at least one token. The token is used to authenticate and send data to Splunk.

   a. Navigate to **Settings** > **Data Inputs** > **HTTP Event Collector**.

   b. Click **New Token**.

   c. Enter the token name as 'collectorMetric' and click **Next**.

   d. Select a source type as `log2metrics_json` from the **Source Type** drop-down list box.

   e. Click **Create a new index**.

   f. Enter the name as 'thunder_metrics' and select the **Index Data Type** as **Metrics**. Click **Save**.

      The index will be add to the **Available Items** list box.

   g. Choose the newly created index ('thunder_metrics') from the **Available Items** list box.

h. Click **Review** to review the settings and then click **Submit**.

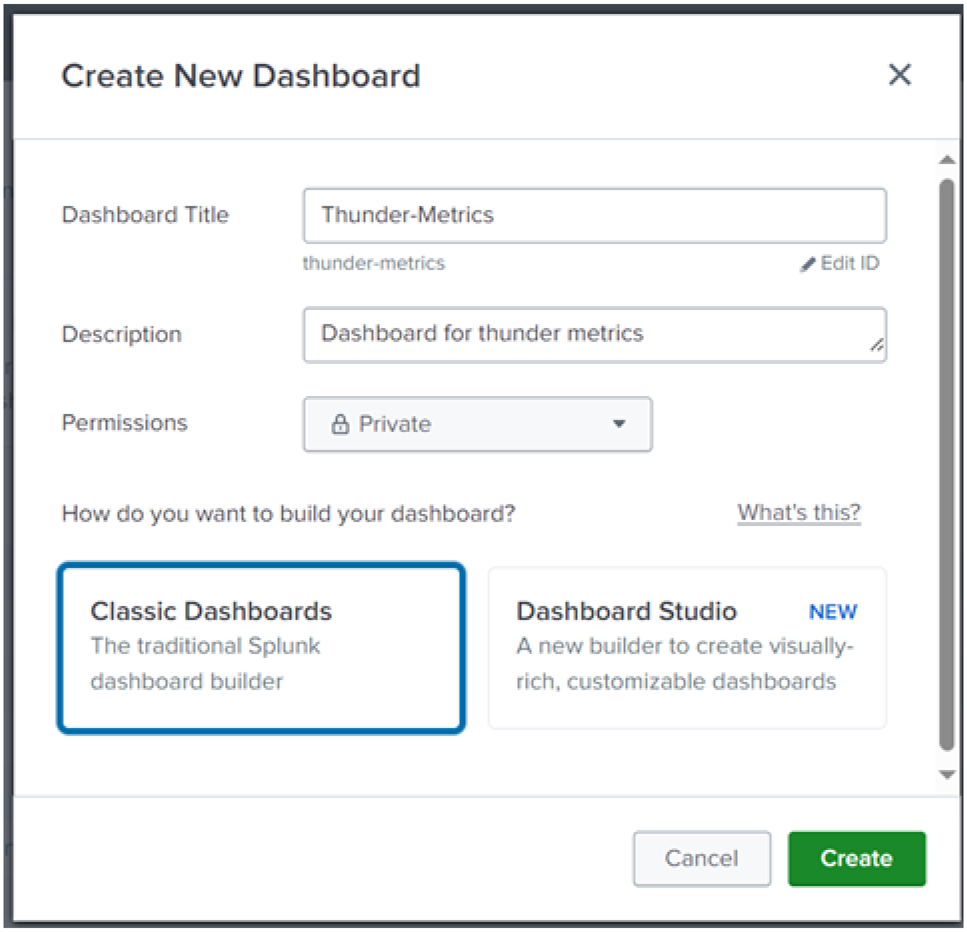
The token is created.

i. Store the token generated for later reference.

NOTE: If you have already created this token then ensure that the dashboard xml file contains the same index name.

3. Navigate to **Apps** > **Search & Reporting** > **Dashboard** and then click **Create New Dashboard**.

Figure 48 : Create New Dashboard



Create New Dashboard

Dashboard Title: Thunder-Metrics
thunder-metrics

Description: Dashboard for thunder metrics

Permissions: Private

How do you want to build your dashboard?

Classic Dashboards
The traditional Splunk dashboard builder

Dashboard Studio NEW
A new builder to create visually-rich, customizable dashboards

4. On the **Create New Dashboard** form, perform the following steps:

a. Enter a name for the dashboard in the **Dashboard Title** field.

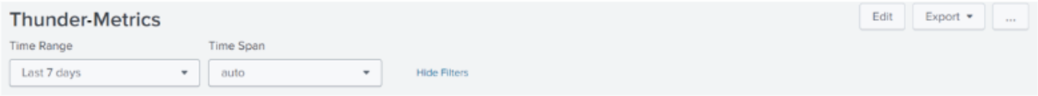
b. Enter description in the **Description** field.

c. Select the appropriate permissions from **Permissions** drop-down menu.

d. Under **How do you want to build your dashboard?**, select **Classic Dashboard** framework, and then click **Create**.

5. On the newly created dashboard, first click **Edit** and then click **Source**.

Figure 49 : Metrics Dashboard



6. Copy the XML code from the dashboard template file and paste it in the editor.

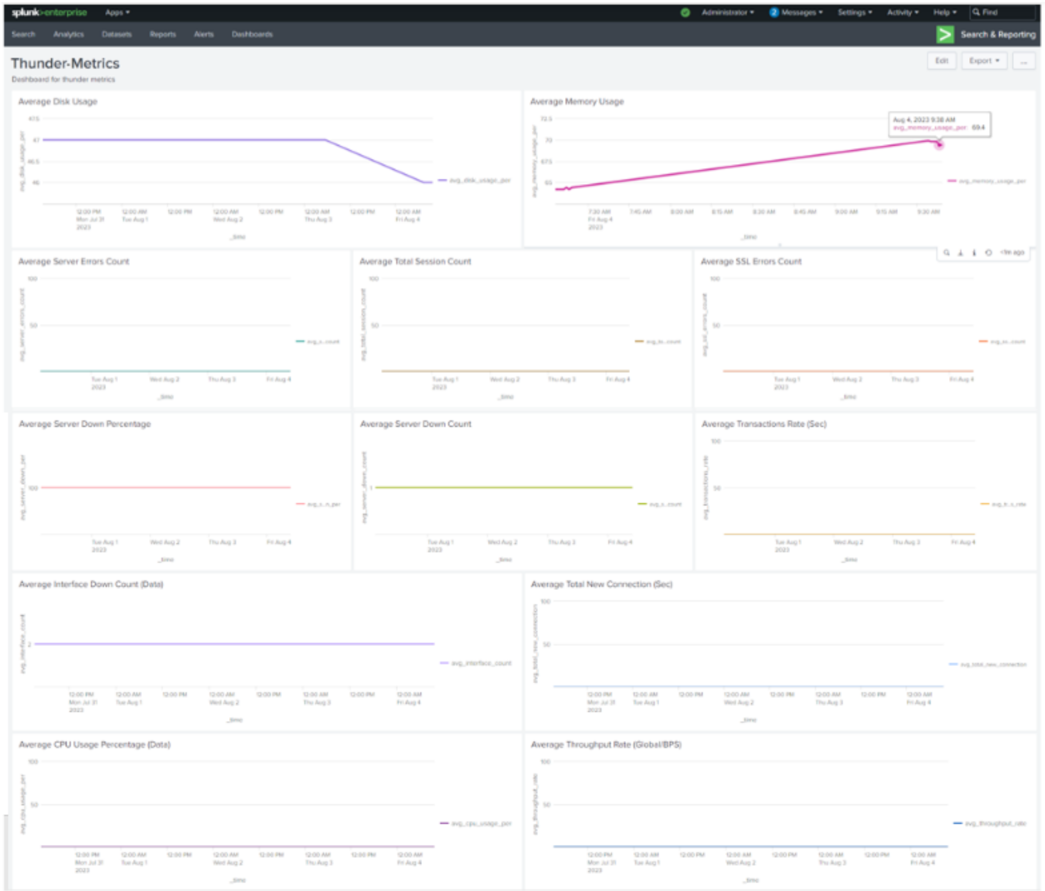
7. Edit the following tags:

- **label** - It must be same as the **Dashboard Title** entered in Step-4a.
- **description** - It must be same as the dashboard **Description** entered in Step-4b.
- **query** - The **index** mentioned in this tag must be same as the one in use.

8. Click **Save**.

9. Verify if the metrics are displayed.

Figure 50 : Thunder-Metrics Dashboard


splunk>enterprise
Apps
Administrator
Messages
Settings
Activity
Help
Find
Search
Analytics
Datasets
Reports
Alerts
Dashboards
Search & Reporting
Thunder-Metrics
Dashboard for thunder metrics
Edit
Export
Average Disk Usage
Average Memory Usage
Average Server Errors Count
Average Total Session Count
Average SSL Errors Count
Average Server Down Percentage
Average Server Down Count
Average Transactions Rate (Sec)
Average Interface Down Count (Data)
Average Total New Connection (Sec)
Average CPU Usage Percentage (Data)
Average Throughput Rate (Global/BPS)
_time

GCP Metrics Explorer

To monitor the Thunder metrics on the GCP Metric Explorer, perform the following steps:

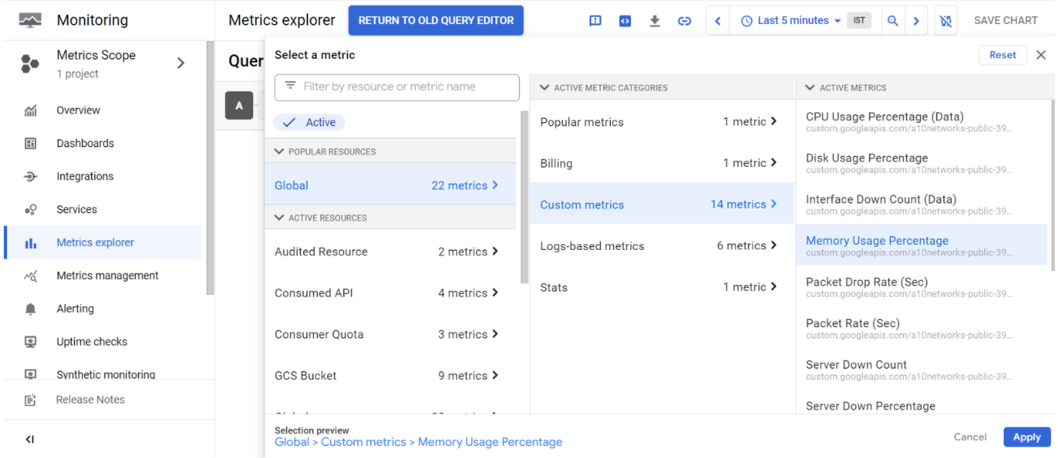
1. View the Metric
2. Customize the Metric
3. Create a Dashboard

View the Metric

1. Open Google Cloud Console and select the project you want to work with.
2. In the navigation menu, select **Monitoring** and then navigate to **Metrics Explorer**.

3. In the **Metric** section, click **Select a metric** to open a drop-down menu.
4. Select **Global** to access metrics applicable to your entire project.
5. Navigate to **Custom metrics**, scroll through the list of custom metrics and select the metric you want to monitor. For example, to chart the memory utilization, you can select **Memory Usage Percentage** metric as show in Figure 51.

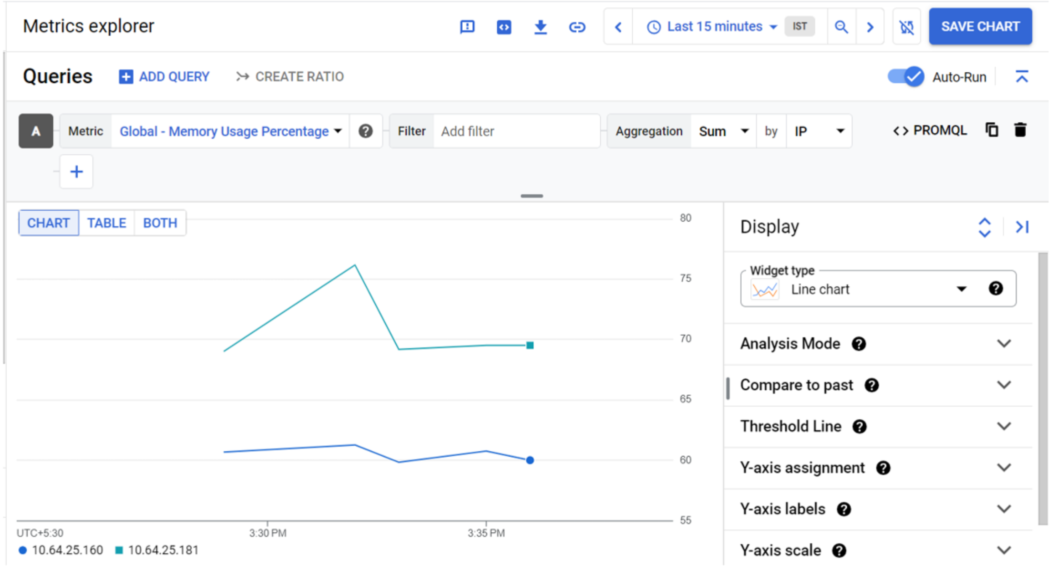
Figure 51 : Select a metric in GCP



6. Click **Apply**.

The metric will be displayed as show in .

Memory usage percentage metric



Metric: Global - Memory Usage Percentage

Filter: Add filter

Aggregation: Sum by IP

Widget type: Line chart

7. Click **Save Chart** in the Metrics Explorer toolbar to save the chart to an existing dashboard or you can create a new dashboard. To create a new dashboard, see Create Dashboard.

Customize the Metric

To customize and analyze your metrics data effectively, you can employ the following options:

- The **Widget type** drop-down menu within the **Display** pane allows you to choose from a variety of chart types including line charts, stacked area charts, and stacked bar charts.
- The **Threshold line** within the **Display** pane allows you to add a threshold line to the metric. You can also set an alert to receive notifications when the threshold is breached.
- The **Compare to Past** option under the **Display** pane allows you to select a time range from the past for comparing the metrics.
- The **Filter** element allows you to narrow down the metrics data based on specific filtering criteria such as resource labels, metric labels, resource types, and other metadata.

- The **Aggregation** element allows you to apply aggregation functions, using such as sum, average, count, min, max, and percentile to aggregate metric data.

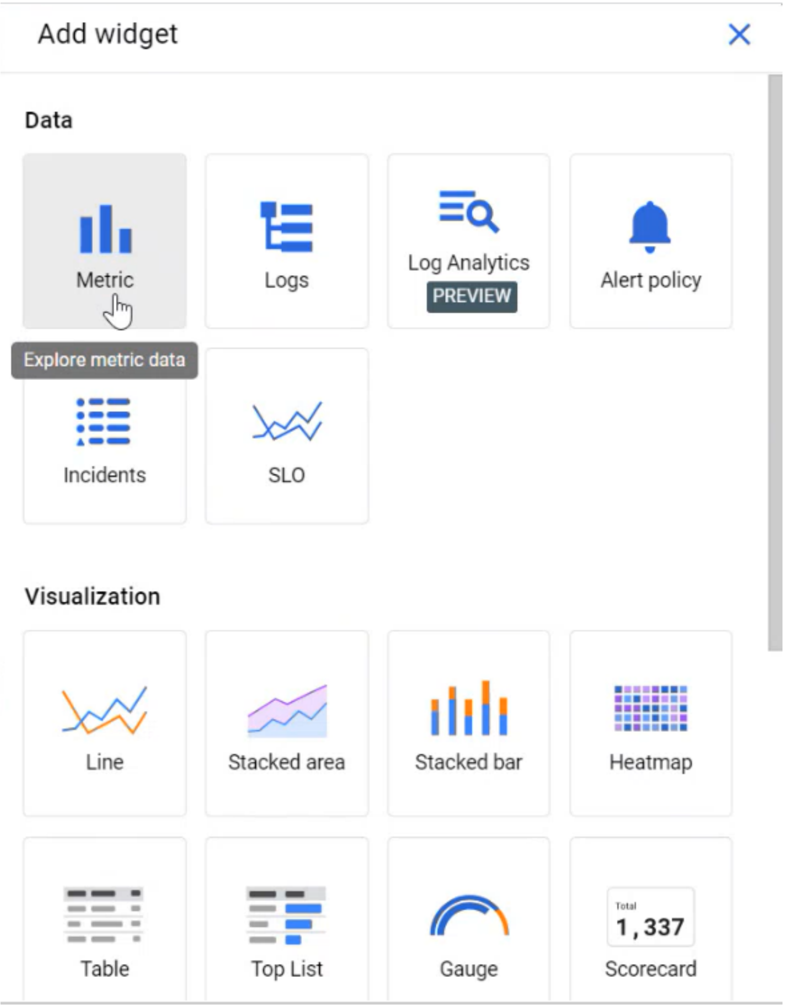
Additionally, to add another metrics to the current chart, you can click **Add Query** and specify the metrics to be monitored. This allows comparing multiple metrics or data series within the same chart.

Create a Dashboard

To create a custom dashboard and monitor a metric, perform the following steps:

1. In the navigation panel, select **Monitoring**, and then click **Dashboards**.
2. On the **Dashboards Overview** page, click **Create Dashboard**.
3. Click the dashboard's title, enter a name for the dashboard and click **Save**.
4. Click **+ Add Widget** and select the **Metrics** widget as shown in Figure 52.

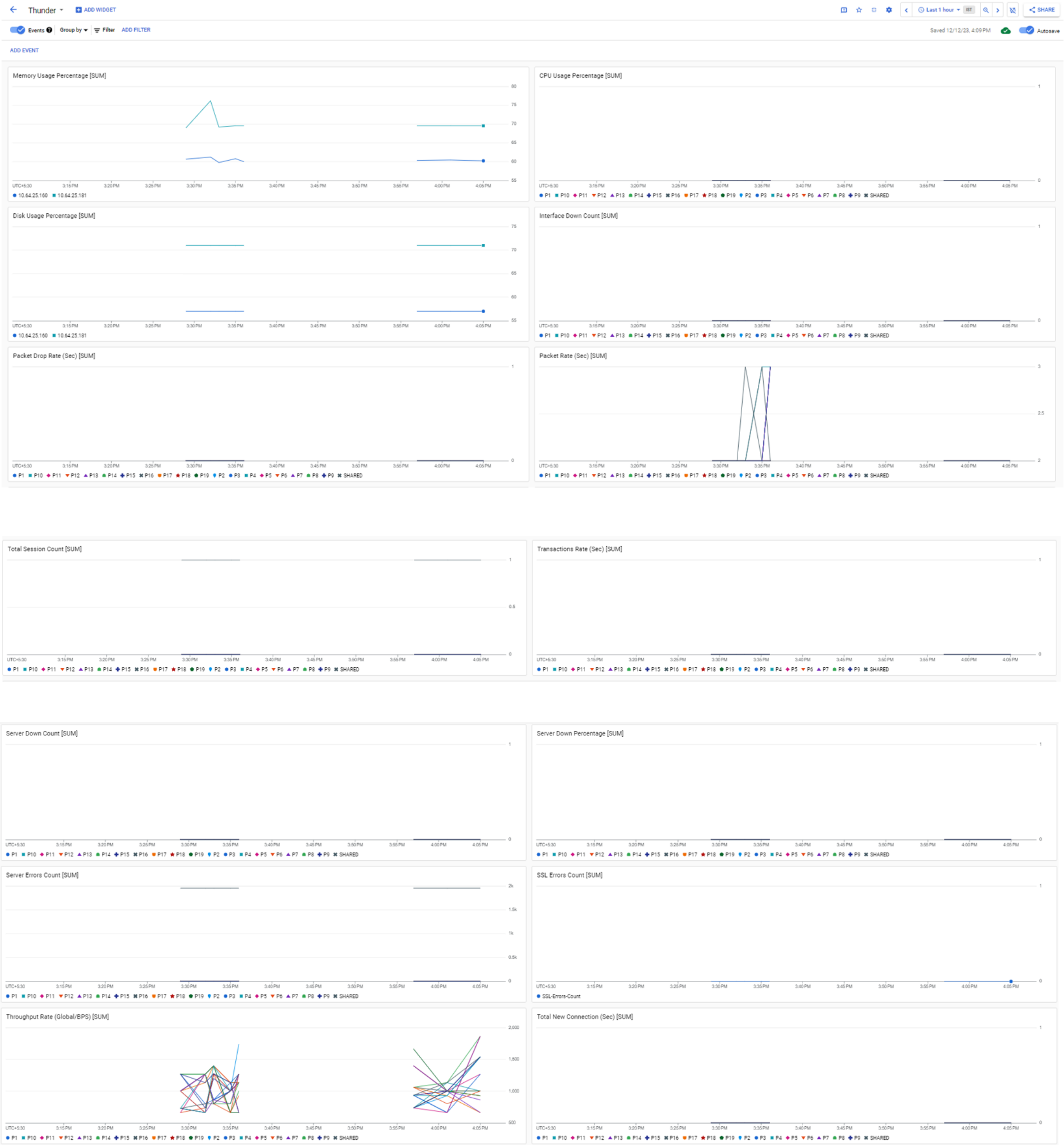
Figure 52 : GCP Dashboard - Add Widget



5. Click **Select a metric** and navigate to **Global** > **Custom metrics**.
6. Scroll through the list of metrics, select a metric you want to monitor, and click **Apply**.

   The metric will be added to the dashboard as shown in the following images.

Figure 53 : Metrics Dashboard in GCP


Thunder
ADD WIDGET
Events
Group by
Filter
ADD FILTER
ADD EVENT
Last 1 hour
SHARE
Autosave
Memory Usage Percentage [SUM]
CPU Usage Percentage [SUM]
Disk Usage Percentage [SUM]
Interface Down Count [SUM]
Packet Drop Rate (Sec) [SUM]
Packet Rate (Sec) [SUM]
Total Session Count [SUM]
Transactions Rate (Sec) [SUM]
Server Down Count [SUM]
Server Down Percentage [SUM]
Server Errors Count [SUM]
SSL Errors Count [SUM]
Throughput Rate (Global/BPS) [SUM]
Total New Connection (Sec) [SUM]

OCI Metrics Explorer

To monitor the Thunder metrics on the OCI Metric Explorer, perform the following steps:

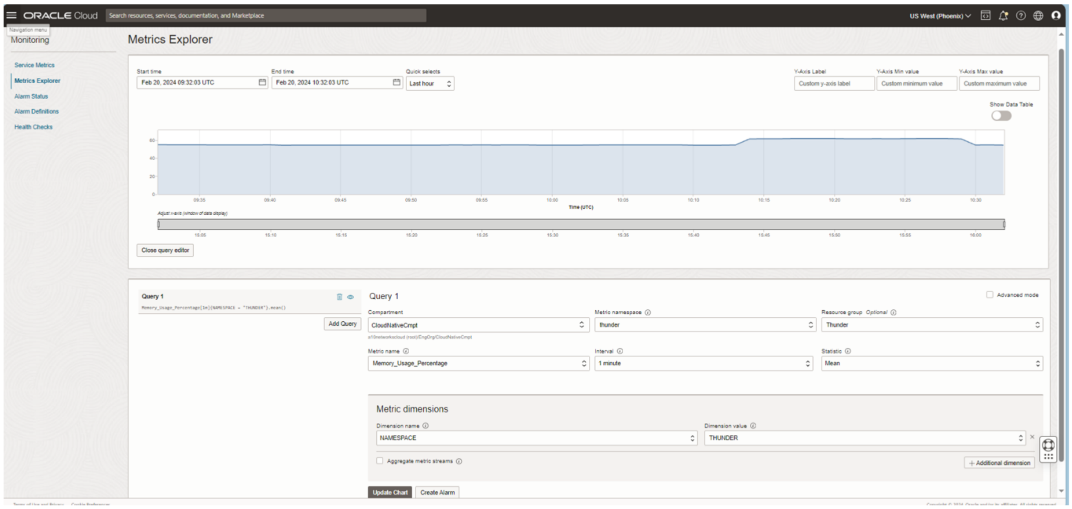
1. View the Chart
2. Create an Alarm
3. Create a Dashboard

View the Chart

1. Log in to the OCI console, open the navigation menu and click **Observability & Management**.
2. Under **Monitoring** , click **Metrics Explorer**.

   The **Metric Explorer** page will be displayed as shown in Figure 54. This page is divided into two sections, the graph section, where the graphs are displayed and the query section, where you can define a query.

   Figure 54 : OCI - Metric Explorer



Start time: Feb 20, 2024 09:32:03 UTC
End time: Feb 20, 2024 10:32:03 UTC
Quick selects: Last hour
Y-Axis Label: Custom y-axis label
Y-Axis Min value: Custom minimum value
Y-Axis Max value: Custom maximum value

Query 1
```
Memory_Usage_Percentage[1m]{NAMESPACE = "THUNDER"}.mean()
```

Compartment: CloudNativeCmpt
a10networkscloud (root)/EngOrg/CloudNativeCmpt
Metric namespace: thunder
Resource group Optional: Thunder
Metric name: Memory_Usage_Percentage
Interval: 1 minute
Statistic: Mean

Metric dimensions
Dimension name: NAMESPACE
Dimension value: THUNDER
□ Aggregate metric streams
□ Advanced mode

3. In the query section, enter the following:

   - **Compartment** - Select a compartment where you want to do the analysis and you have access.
   - **Metric namespace** - Select the metric namespace for querying metric data; `thunder` in this case. This drop-down lists metric namespaces for the selected compartment.
   - **Resource group** - Select a resource group; `Thunder` in this case. Specifying a resource group ensures that only metric data for the resources within that group are returned.
   - **Metric Name** - Select a metric name from the drop-down menu.

- **Metric Dimensions** (optional) - Set dimensions by selecting a dimension name and specifying a dimension value. By selecting appropriate dimensions, you can limit the metric data.

4. Click **Update Chart**.

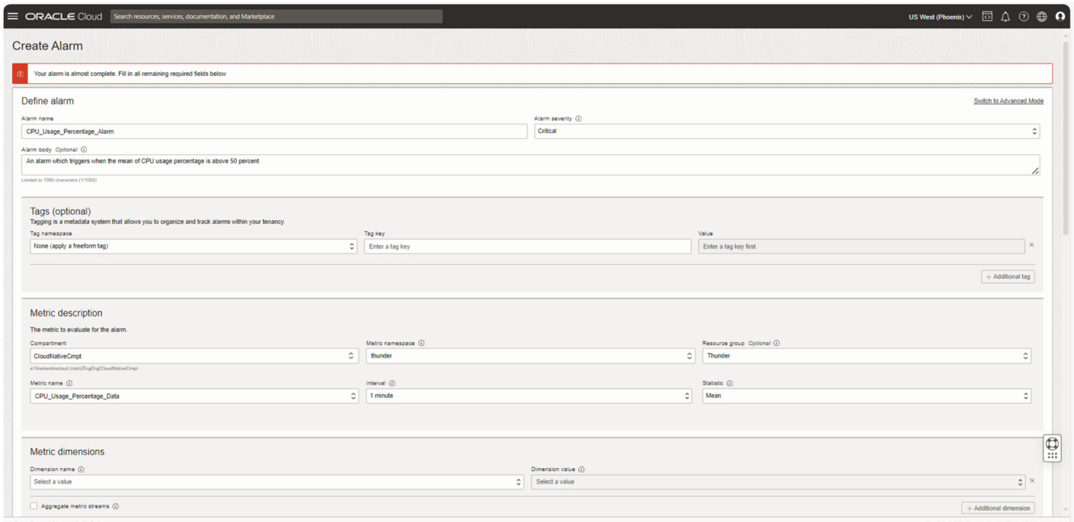
   The updated chart will be displayed in the graph section.

Create an Alarm

1. In the query portion of the **Metric Explorer** page, click **Create Alarm**.

   The **Create Alarm** page will be displayed as shown in Figure 55.

   Figure 55 : OCI - Create Alarm Page



Create Alarm

Your alarm is almost complete. Fill in all remaining required fields below

Define alarm

Alarm name: CPU_Usage_Percentage_Alarm

Alarm severity: Critical

Alarm body Optional: An alarm which triggers when the mean of CPU usage percentage is above 50 percent.

Limited to 1000 characters (1/1000)

Tags (optional)

Tagging is a metadata system that allows you to organize and track alarms within your tenancy.

Tag namespace: None (apply a freeform tag)

Tag key: Enter a tag key

Value: Enter a tag key first

Metric description

The metric to evaluate for the alarm.

Compartment: CloudNativeCmpt

a10networkscloud (root)/EngOrg/CloudNativeCmpt

Metric namespace: thunder

Resource group Optional: Thunder

Metric name: CPU_Usage_Percentage_Data

Interval: 1 minute

Statistic: Mean

Metric dimensions

Dimension name: Select a value

Dimension value: Select a value

2. Enter a name for the alarm and select the severity level from the **Alarm Severity** drop-down menu.

3. In the **Alarm body** text box, enter a notification message.

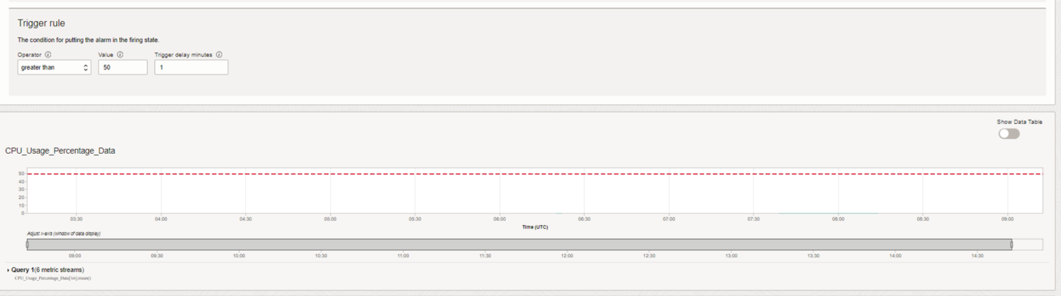
4. In the **Metric description** area, enter the following metric details:

   - **Compartment** - Select the compartment that contains the resources to generate metrics evaluated by the alarm. This compartment also serves as a storage location for the alarm.

   - **Metric namespace** - Select a service that generates the metrics for the resources that you want to monitor; `thunder` in this case. The drop-down lists all metric namespaces for the selected compartment.
   - **Resource group** - Select the resource group that the metric belongs to; `Thunder` in this case.
   - **Metric name** - Select a metric name you wish to evaluate for the alarm.
   - **Interval** - Select a time-frame or a frequency at which data points are aggregated.
   - **Statistics** - Select a statistical function to aggregate data points. The options available are **Mean**, **Rate**, **Sum**, **Max**, **Min**, and more.

5. In the **Metric Dimensions** area, select a dimension name and specify a dimension value. By selecting appropriate dimensions, you can narrow the metric data to be evaluated.
6. In the **Trigger rule** area, specify the condition to be satisfied for the alarm to be triggered. Set the following parameters:
   - **Operator** - Select an operator to be used for the condition threshold. For example, **greater than** or **less than** operators.
   - **Value** - Enter the value to be used for the condition threshold.
   - **Trigger delay minutes** - Enter the number of minutes before the alarm is triggered.

   Figure 56 : Create Alarm - Trigger rule area



Trigger rule

The condition for putting the alarm in the firing state.

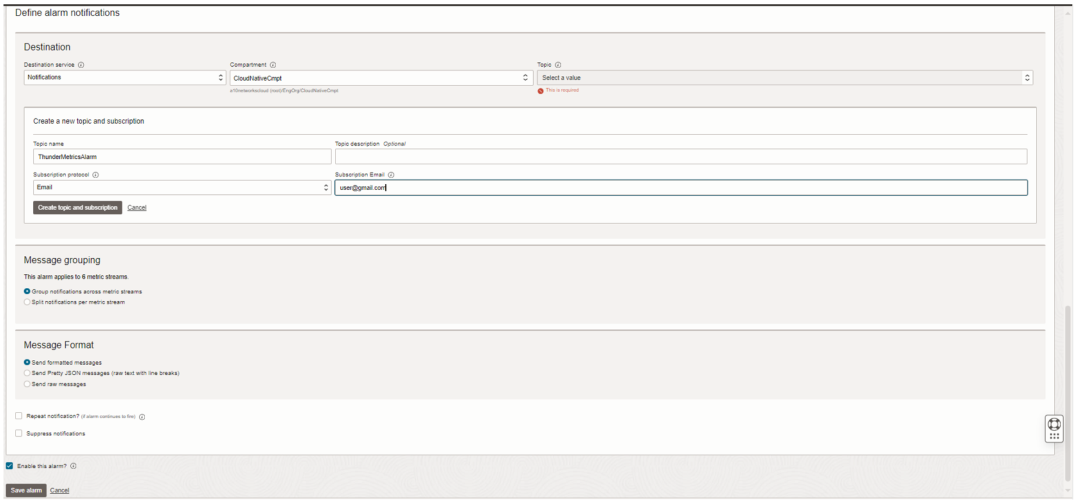
Operator: greater than

Value: 50

Trigger delay minutes: 1

7. In the **Destination** area under **Define alarm notifications**, select the destination for alarm notifications.

Figure 57 : OCI - Define Alarm Notifications



Define alarm notifications

Destination

Destination service: Notifications

Compartment: CloudNativeCmpt

a10networkscloud (root)/EngOrg/CloudNativeCmpt

Topic: Select a value

This is required

Create a new topic and subscription

Topic name: ThunderMetricsAlarm

Topic description *Optional*

Subscription protocol: Email

Subscription Email: user@gmail.com

Message grouping

This alarm applies to 6 metric streams.

● Group notifications across metric streams

○ Split notifications per metric stream

Message Format

● Send formatted messages

○ Send Pretty JSON messages (raw text with line breaks)

○ Send raw messages

□ Repeat notification? (if alarm continues to fire)

□ Suppress notifications

☑ Enable this alarm?

Set the following parameters:

- **Destination service** - Select one of the following:
  - **Notifications** - Send alarm notifications to a topic. A topic is a communication channel for sending messages.
  - **Streaming** - Send alarm messages to a stream. A stream is an append-only log.
- **Compartment** - Select the compartment that contains the resources that generate metrics evaluated by the alarm.
- **Stream** (If **Destination service** selected is **Streaming**) - Select a stream for alarm notification.
- **Topic** (If **Destination service** selected is **Notifications**) - Select a topic to be used for notifications.

  You can select an existing topic or create a new one. To create a new topic, click **Create a topic**. Enter a topic name, description, **Subscription Protocol** (Email, SMS, Custom URL, and more) and click **Create topic and subscription**.

8. Select **Enable the alarm?** checkbox.

When the alarm is enabled, the configured metric is evaluated and alarm messages are sent to the selected destination service when the metric data satisfies a condition and triggers the rule.

9. Click **Save Alarm**.

   You can view the newly created alarm by navigating to **Monitoring** > **Alarm Definitions**. Here, you can enable, disable, and edit the alarm as well.

For more information of Alarms, see Managing Alarms.

Create a Dashboard

To create a dashboard, perform the following steps:

1. Log in to the OCI console, open the navigation menu and click **Observability & Management**.
2. Under **Logging Analytics**, click **Dashboard**.

   The **Dashboards** page with a list of existing dashboards will be displayed.

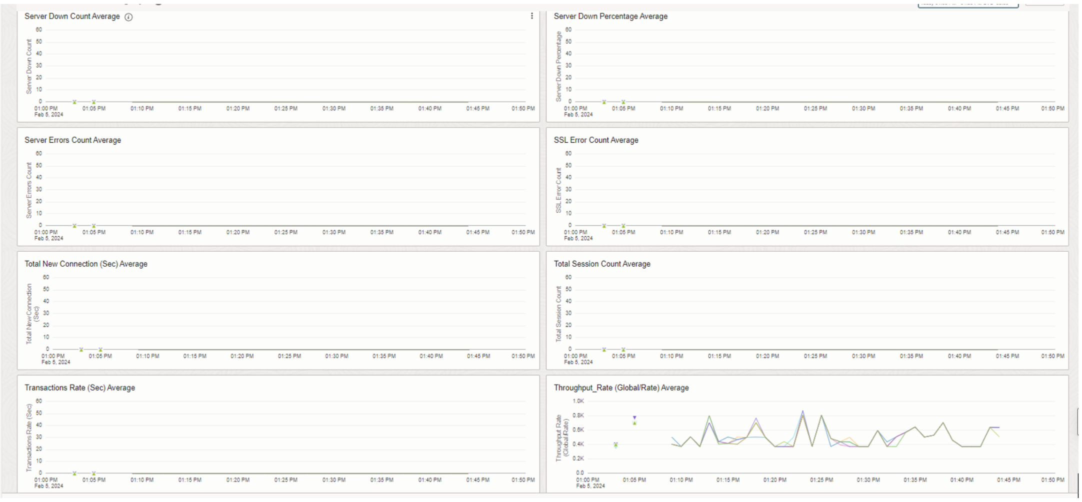
3. Click **Create dashboard**.
4. In the **About** tab, enter a **Dashboard name**, **Dashboard compartment**, and **Dashboard description**.
5. In the **Widgets** tab, click **+**.

   Here you can select one of the following:

   - **Create Widget** - This option allows you to add a variety of pre-configured widgets to your dashboard. To create a widget, see Create Widget .
   - **Create Query-Based Widget** - The option allows you to add widgets based on queries executed against your data. To create a query-based widget, see Create Query-Based Widget.

   After creating and saving the widgets, they are automatically added to the dashboard as shown in Figure 58.

Figure 58 : Widgets added to Dashboard



Monitor Logs

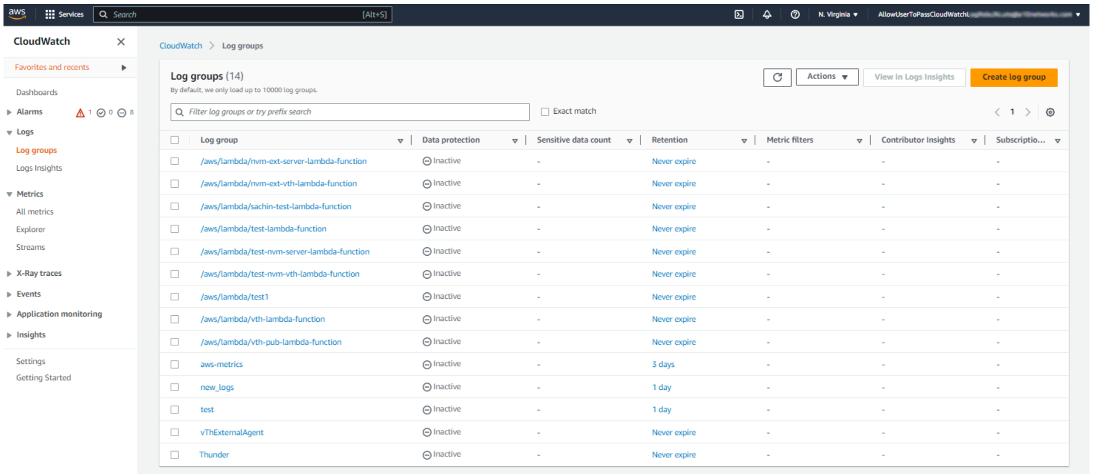
Depending on your cloud provider, the steps are provided to monitor the configured logs.

AWS CloudWatch

To monitor the Thunder logs on the AWS CloudWatch, perform the following steps:

1. From the **AWS Management Console**, go to **CloudWatch** > **Logs** > **Log groups**.

Figure 59 : AWS Log Groups

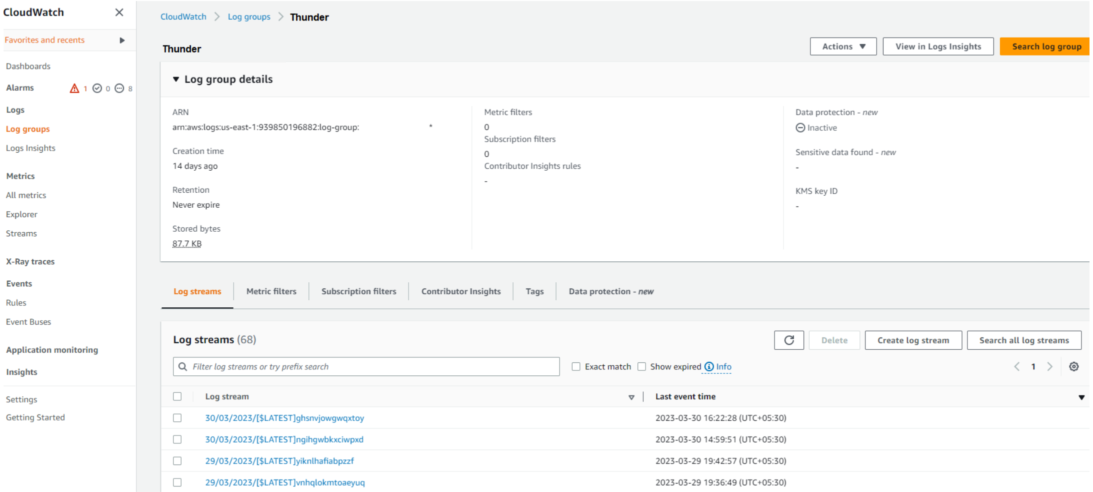


Log groups (14)
By default, we only load up to 10000 log groups.

Log group	Data protection	Sensitive data count	Retention	Metric filters	Contributor Insights	Subscriptio...
/aws/lambda/nvm-ext-server-lambda-function	Inactive	-	Never expire	-	-	-
/aws/lambda/nvm-ext-vth-lambda-function	Inactive	-	Never expire	-	-	-
/aws/lambda/sachin-test-lambda-function	Inactive	-	Never expire	-	-	-
/aws/lambda/test-lambda-function	Inactive	-	Never expire	-	-	-
/aws/lambda/test-nvm-server-lambda-function	Inactive	-	Never expire	-	-	-
/aws/lambda/test-nvm-vth-lambda-function	Inactive	-	Never expire	-	-	-
/aws/lambda/test1	Inactive	-	Never expire	-	-	-
/aws/lambda/vth-lambda-function	Inactive	-	Never expire	-	-	-
/aws/lambda/vth-pub-lambda-function	Inactive	-	Never expire	-	-	-
aws-metrics	Inactive	-	3 days	-	-	-
new_logs	Inactive	-	1 day	-	-	-
test	Inactive	-	1 day	-	-	-
vThExternalAgent	Inactive	-	Never expire	-	-	-
Thunder	Inactive	-	Never expire	-	-	-

2. Click **Thunder** log group.

Figure 60 : Thunder Log Group



Thunder

Log group details

ARN
arn:aws:logs:us-east-1:939850196882:log-group: *

Creation time
14 days ago

Retention
Never expire

Stored bytes
87.7 KB

Metric filters
0

Subscription filters
0

Contributor Insights rules
-

Data protection - *new*
Inactive

Sensitive data found - *new*
-

KMS key ID
-

Log streams (68)

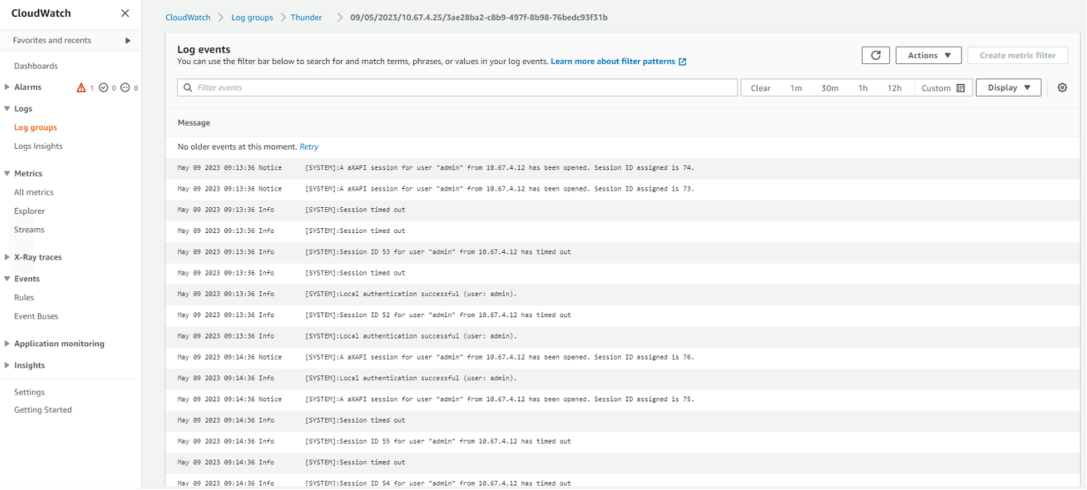
□ Exact match □ Show expired Info

□	Log stream	Last event time
□	30/03/2023/[$LATEST]ghsnvjowgwqxtoy	2023-03-30 16:22:28 (UTC+05:30)
□	30/03/2023/[$LATEST]ngihgwbkxciwpxd	2023-03-30 14:59:51 (UTC+05:30)
□	29/03/2023/[$LATEST]yiknlhafiabpzzf	2023-03-29 19:42:57 (UTC+05:30)
□	29/03/2023/[$LATEST]vnhqlokmtoaeyuq	2023-03-29 19:36:49 (UTC+05:30)

3. Under the **Log streams** tab, click the required log stream to be monitored.

   All logs are displayed in tabular format with expandable details.

Figure 61 : Logs events on AWS CloudWatch



Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

Message

No older events at this moment. *Retry*

```
May 09 2023 09:13:36 Notice    [SYSTEM]:A aXAPI session for user "admin" from 10.67.4.12 has been opened. Session ID assigned is 74.
May 09 2023 09:13:36 Notice    [SYSTEM]:A aXAPI session for user "admin" from 10.67.4.12 has been opened. Session ID assigned is 73.
May 09 2023 09:13:36 Info      [SYSTEM]:Session timed out
May 09 2023 09:13:36 Info      [SYSTEM]:Session timed out
May 09 2023 09:13:36 Info      [SYSTEM]:Session ID 53 for user "admin" from 10.67.4.12 has timed out
May 09 2023 09:13:36 Info      [SYSTEM]:Session timed out
May 09 2023 09:13:36 Info      [SYSTEM]:Local authentication successful (user: admin).
May 09 2023 09:13:36 Info      [SYSTEM]:Session ID 52 for user "admin" from 10.67.4.12 has timed out
May 09 2023 09:13:36 Info      [SYSTEM]:Local authentication successful (user: admin).
May 09 2023 09:14:36 Notice    [SYSTEM]:A aXAPI session for user "admin" from 10.67.4.12 has been opened. Session ID assigned is 76.
May 09 2023 09:14:36 Info      [SYSTEM]:Local authentication successful (user: admin).
May 09 2023 09:14:36 Notice    [SYSTEM]:A aXAPI session for user "admin" from 10.67.4.12 has been opened. Session ID assigned is 75.
May 09 2023 09:14:36 Info      [SYSTEM]:Session timed out
May 09 2023 09:14:36 Info      [SYSTEM]:Session ID 55 for user "admin" from 10.67.4.12 has timed out
May 09 2023 09:14:36 Info      [SYSTEM]:Session timed out
May 09 2023 09:14:36 Info      [SYSTEM]:Session ID 54 for user "admin" from 10.67.4.12 has timed out
```

Azure Log Analytics Workspace

To monitor the Thunder logs on the Azure Log Analytics Workspace, perform the following steps:

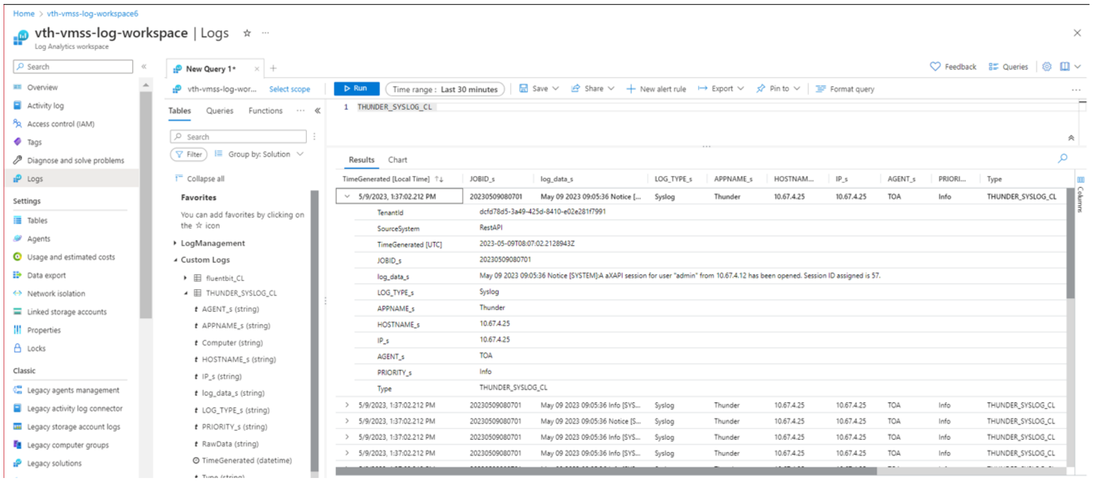
1. From the **Azure Portal**, go to **Azure services** > **Resource Groups** > *<your_ resource_group>* and click your log analytics workspace name.
2. Click **Logs** from the left **General** panel.

   You can close the **Queries** pop-up window.

3. From **New Query1** > **Tables** tab, expand **Custom Logs**.
4. Double click **THUNDER_SYSLOG_CL**.

   The THUNDER_SYSLOG_CL query window is displayed.

   Figure 62 : Custom Logs window



```
THUNDER_SYSLOG_CL
```

TimeGenerated [Local Time]	JOBID_s	log_data_s	LOG_TYPE_s	APPNAME_s	HOSTNAM...	IP_s	AGENT_s	PRIORI...	Type
5/9/2023, 1:37:02.212 PM	20230509080701	May 09 2023 09:05:36 Notice [...	Syslog	Thunder	10.67.4.25	10.67.4.25	TOA	Info	THUNDER_SYSLOG_CL

TenantId	dcfd78d5-3a49-425d-8410-e02e281f7991
SourceSystem	RestAPI
TimeGenerated [UTC]	2023-05-09T08:07:02.2128943Z
JOBID_s	20230509080701
log_data_s	May 09 2023 09:05:36 Notice [SYSTEM]:A aXAPI session for user "admin" from 10.67.4.12 has been opened. Session ID assigned is 57.
LOG_TYPE_s	Syslog
APPNAME_s	Thunder
HOSTNAME_s	10.67.4.25
IP_s	10.67.4.25
AGENT_s	TOA
PRIORITY_s	Info
Type	THUNDER_SYSLOG_CL

TimeGenerated [Local Time]	JOBID_s	log_data_s	LOG_TYPE_s	APPNAME_s	HOSTNAM...	IP_s	AGENT_s	PRIORI...	Type
5/9/2023, 1:37:02.212 PM	20230509080701	May 09 2023 09:05:36 Info [SYS...	Syslog	Thunder	10.67.4.25	10.67.4.25	TOA	Info	THUNDER_SYSLOG_CL
5/9/2023, 1:37:02.212 PM	20230509080701	May 09 2023 09:05:36 Notice [S...	Syslog	Thunder	10.67.4.25	10.67.4.25	TOA	Info	THUNDER_SYSLOG_CL
5/9/2023, 1:37:02.212 PM	20230509080701	May 09 2023 09:05:36 Info [SYS...	Syslog	Thunder	10.67.4.25	10.67.4.25	TOA	Info	THUNDER_SYSLOG_CL
5/9/2023, 1:37:02.212 PM	20230509080701	May 09 2023 09:05:36 Info [SYS...	Syslog	Thunder	10.67.4.25	10.67.4.25	TOA	Info	THUNDER_SYSLOG_CL

5. Click **Run**.

   All logs are displayed in tabular format with expandable details.

   The following table lists the Thunder Logs filter options:

   Table 56 : Log Filters

Filter	Description
`log_data`	Specifies the actual log entry.
`hostname`	Displays the vThunder resource ID.
`log_type`	Displays the vThunder system logs.
`appname`	Displays the application name.

Table 56 : Log Filters

Filter	Description
`ip`	Displays the vThunder IP address.
`agent`	Displays the agent name.
`jobid`	Displays the JOB ID provided in the **`thunder-observability-agent.log`** file.
`priority`	Displays the Notice, Info, Error, and so on as per actual log entry.
`partition`	Displays the vThunder partition name.

VMware vRLI

To monitor the Thunder logs on the VMware vRLI, perform the following steps:

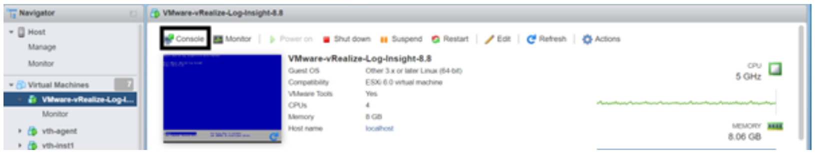
1. [Start vRLI VM](#)
2. [View Logs](#)

Start vRLI VM

To start the vRLI virtual machine, perform the following steps:

1. From the **VMware ESXi** console, go to **Navigator** > **Virtual Machines** > <*your_vRLI_VM*> and click **Power on**.

   Figure 63 : Start vRLI VM

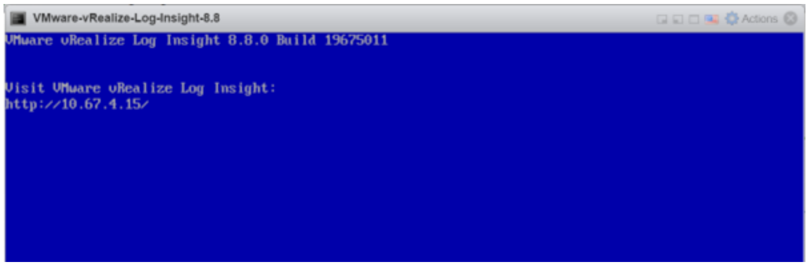
   VMware-vRealize-Log-Insight-8.8
   Guest OS Other 3.x or later Linux (64-bit)
   Compatibility ESXi 6.0 virtual machine
   VMware Tools Yes
   CPUs 4
   Memory 8 GB
   Host name localhost

NOTE: The system may take a few minutes to start the vRLI virtual machine.

2. Click **Console** to launch vRLI virtual machine.

   The vRLI virtual machine is powered on and reachable.

Figure 64 : VMware vRealize Log Insight



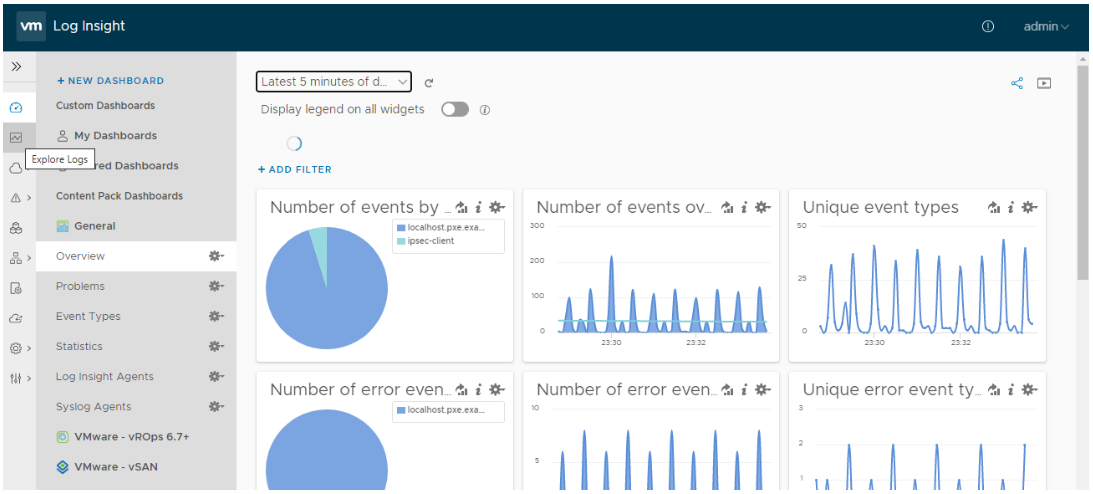
```
VMware vRealize Log Insight 8.8.0 Build 19675011

Visit VMware vRealize Log Insight:
http://10.67.4.15/
```

View Logs

1. From the **vRealize Log Insight Web UI**, go to **Home** > **Explore Logs** to view the logs.

   The **Logs** window is displayed.

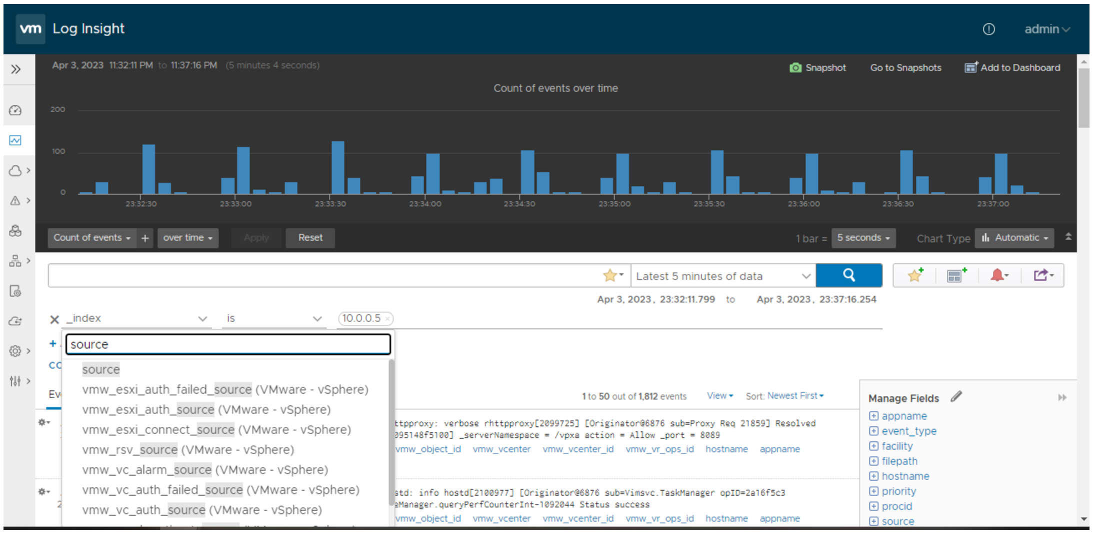
   Figure 65 : vRealize Log Insight - Overview window

2. Click **Add Filter** and add the following filter criteria to search all the logs received from a specific Thunder IP:

   - _index: ip
   - condition: is
   - value: *<Thunder_IP>*

Figure 66 : vRealize Log Insight - Add Filter



Apr 3, 2023, 23:32:11.799 to Apr 3, 2023, 23:37:16.254

_index | is | 10.0.0.5

source

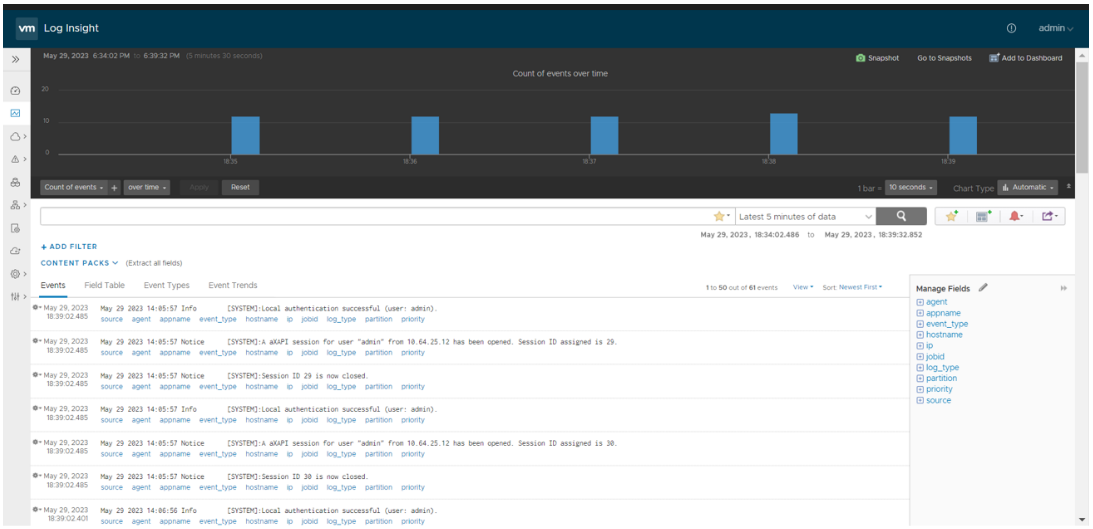
- source
- vmw_esxi_auth_failed_source (VMware - vSphere)
- vmw_esxi_auth_source (VMware - vSphere)
- vmw_esxi_connect_source (VMware - vSphere)
- vmw_rsv_source (VMware - vSphere)
- vmw_vc_alarm_source (VMware - vSphere)
- vmw_vc_auth_failed_source (VMware - vSphere)
- vmw_vc_auth_source (VMware - vSphere)

1 to 50 out of 1,812 events

```
ttpproxy: verbose rhttpproxy[2099725] [Originator@6876 sub=Proxy Req 21859] Resolved
395148f5100] _serverNamespace = /vpxa action = Allow _port = 8089
std: info hostd[2100977] [Originator@6876 sub=Vimsvc.TaskManager opID=2a16f5c3
eManager.queryPerfCounterInt-1092044 Status success
```

3. Add the following filter criteria to search all logs received from TOA:
   - _index: source
   - condition: is
   - value: *<TOA_IP>*
4. Verify if the logs are generated.

Figure 67 : Logs on vRealize Log Insight



May 29, 2023, 18:34:02.486 to May 29, 2023, 18:39:32.852

1 to 50 out of 61 events

```
May 29, 2023 18:39:02.485  May 29 2023 14:05:57 Info      [SYSTEM]:Local authentication successful (user: admin).
May 29, 2023 18:39:02.485  May 29 2023 14:05:57 Notice    [SYSTEM]:A aXAPI session for user "admin" from 10.64.25.12 has been opened. Session ID assigned is 29.
May 29, 2023 18:39:02.485  May 29 2023 14:05:57 Notice    [SYSTEM]:Session ID 29 is now closed.
May 29, 2023 18:39:02.485  May 29 2023 14:05:57 Info      [SYSTEM]:Local authentication successful (user: admin).
May 29, 2023 18:39:02.485  May 29 2023 14:05:57 Notice    [SYSTEM]:A aXAPI session for user "admin" from 10.64.25.12 has been opened. Session ID assigned is 30.
May 29, 2023 18:39:02.485  May 29 2023 14:05:57 Notice    [SYSTEM]:Session ID 30 is now closed.
May 29, 2023 18:39:02.401  May 29 2023 14:06:56 Info      [SYSTEM]:Local authentication successful (user: admin).
```

The following table lists the Thunder Logs filter options:

Table 57 : Log Filters

Filter	Description
`hostname`	Displays the Thunder resource ID.
`log_type`	Displays the Thunder system logs.
`appname`	Displays the application name.
`ip`	Displays the Thunder IP address.
`agent`	Displays the agent name.
`jobid`	Displays the JOB ID provided in TOA in the **`thunder-observability-agent.log`** file.
`priority`	Displays the Notice, Info, or Error, and so on as per actual log entry.
`partition`	Displays the Thunder partition name.

Kibana (Elasticsearch)

To monitor the Thunder logs on Kibana UI, perform the following steps:

1. Import the Kibana dashboard.

   To import the Kibana dashboard, perform the following steps:

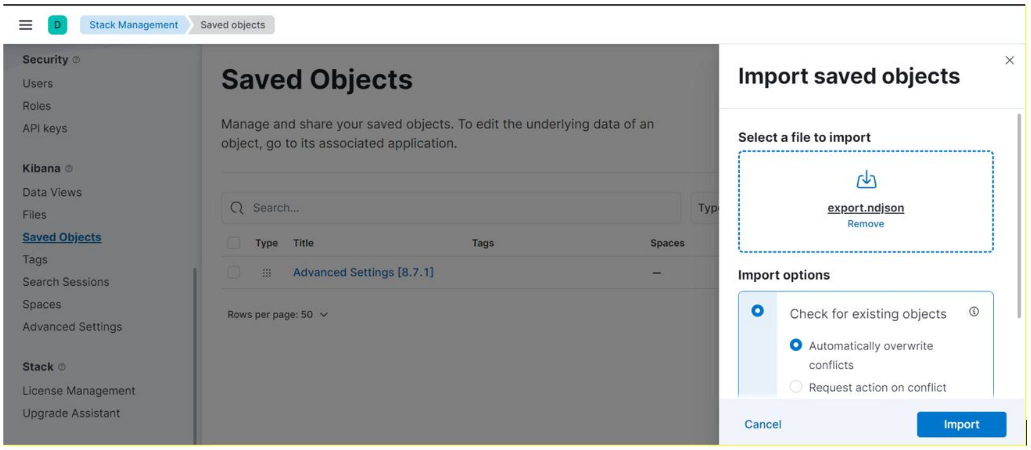
   a. Download the dashboard-template JSON file.

   b. Log in to Kibana.

   c. Navigate to **Menu** > **Management** > **Saved Objects** > **Import**.

   d. Select the downloaded Kibana dashboard file and click **Import**.

Figure 68 : Importing Dashboard File



Saved Objects

Manage and share your saved objects. To edit the underlying data of an object, go to its associated application.

Type	Title	Tags	Spaces
	Advanced Settings [8.7.1]	—	

Import saved objects

Select a file to import: export.ndjson

Import options

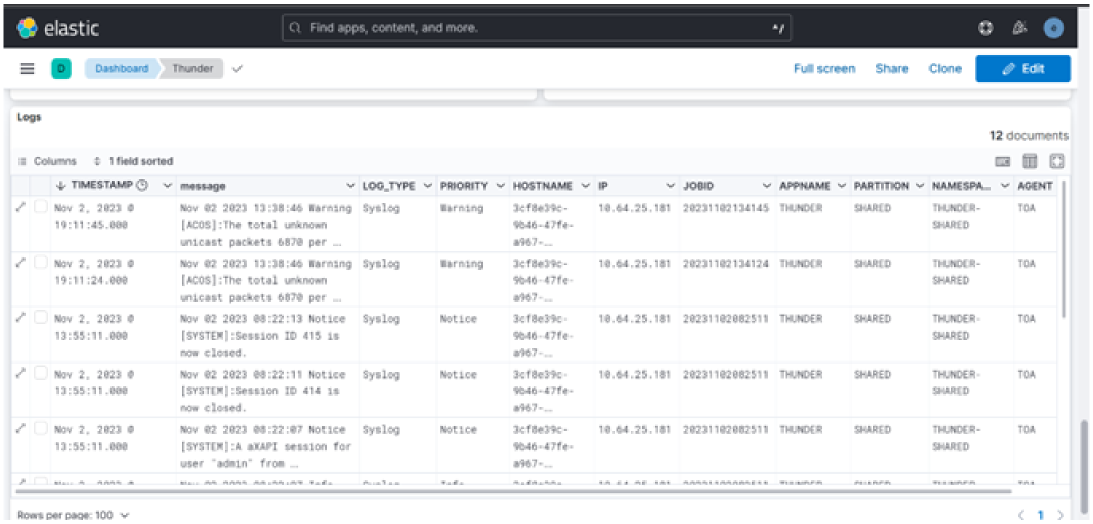
● Check for existing objects

- ● Automatically overwrite conflicts
- ○ Request action on conflict

2. View the Logs.

   To view the logs, navigate to **Menu** > **Dashboard**. All the logs are visible below the metrics as shown:

   Figure 69 : Logs Dashboard



Logs

12 documents

TIMESTAMP	message	LOG_TYPE	PRIORITY	HOSTNAME	IP	JOBID	APPNAME	PARTITION	NAMESPA...	AGENT
Nov 2, 2023 @ 19:11:45.000	Nov 02 2023 13:38:46 Warning [ACOS]:The total unknown unicast packets 6870 per ...	Syslog	Warning	3cf8e39c-9b46-47fe-a967-...	10.64.25.181	20231102134145	THUNDER	SHARED	THUNDER-SHARED	TOA
Nov 2, 2023 @ 19:11:24.000	Nov 02 2023 13:38:46 Warning [ACOS]:The total unknown unicast packets 6870 per ...	Syslog	Warning	3cf8e39c-9b46-47fe-a967-...	10.64.25.181	20231102134124	THUNDER	SHARED	THUNDER-SHARED	TOA
Nov 2, 2023 @ 13:55:11.000	Nov 02 2023 08:22:13 Notice [SYSTEM]:Session ID 415 is now closed.	Syslog	Notice	3cf8e39c-9b46-47fe-a967-...	10.64.25.181	20231102082511	THUNDER	SHARED	THUNDER-SHARED	TOA
Nov 2, 2023 @ 13:55:11.000	Nov 02 2023 08:22:11 Notice [SYSTEM]:Session ID 414 is now closed.	Syslog	Notice	3cf8e39c-9b46-47fe-a967-...	10.64.25.181	20231102082511	THUNDER	SHARED	THUNDER-SHARED	TOA
Nov 2, 2023 @ 13:55:11.000	Nov 02 2023 08:22:07 Notice [SYSTEM]:A aXAPI session for user "admin" from ...	Syslog	Notice	3cf8e39c-9b46-47fe-a967-...	10.64.25.181	20231102082511	THUNDER	SHARED	THUNDER-SHARED	TOA

3. View the Log Hits.

   To view all the log hits along with meta field details, navigate to **Menu** > **Discover** > **Thunder-Logs**.

Grafana (Prometheus)

To monitor the Thunder logs on Grafana UI, perform the following steps:

1. Import the Grafana dashboard.

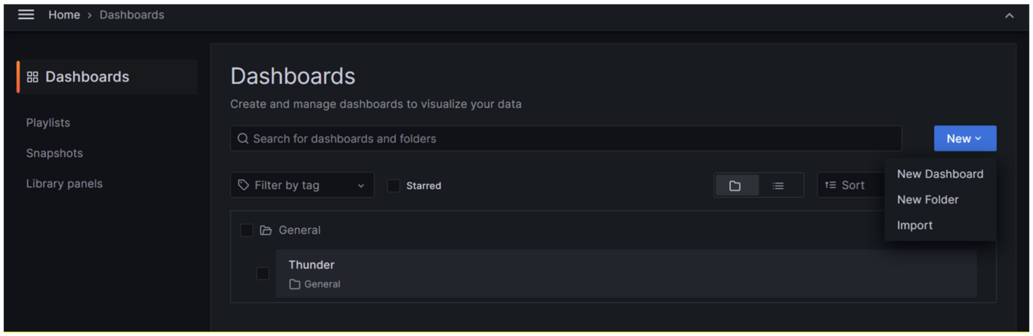
   To import the Grafana dashboard, perform the following steps:

   a. Download the dashboard-template JSON file.

   b. Log in to Grafana.

   c. Navigate to **Menu** > **Dashboard** and click **New** > **Import**.

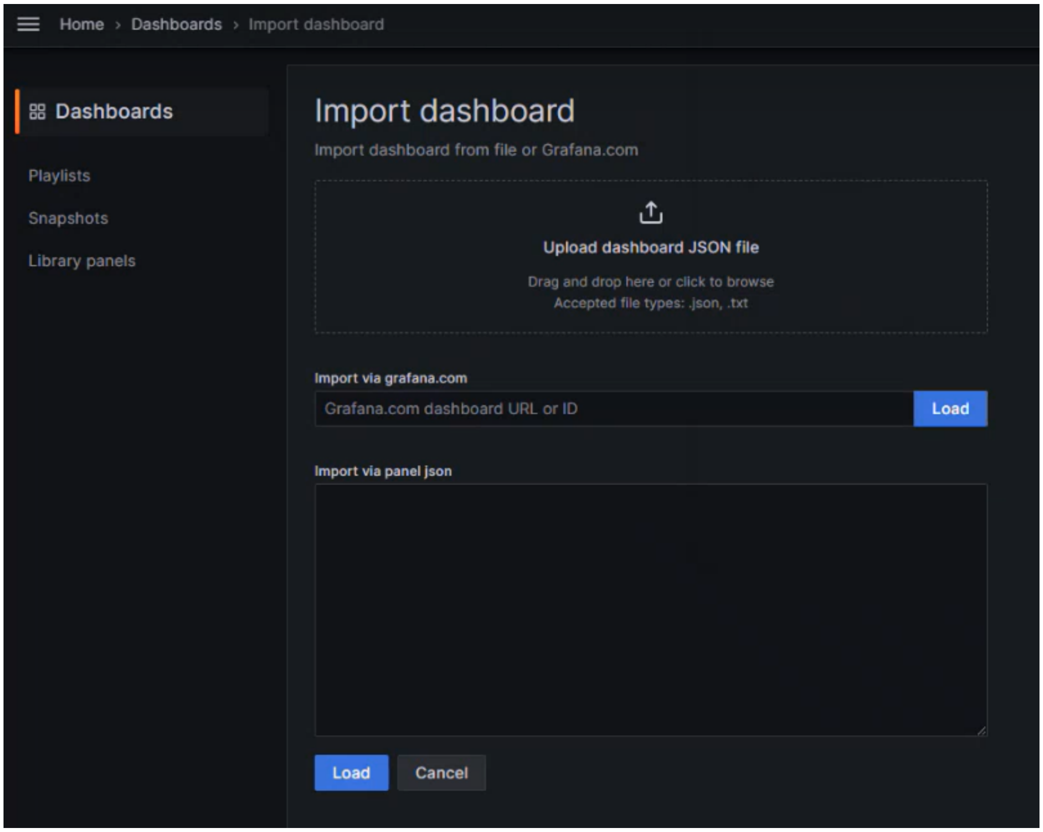
   Figure 70 : Dashboards



Create and manage dashboards to visualize your data

   d. On the **Import dashboard** page, click **Upload dashboard JSON file**.

Figure 71 : Import dashboard

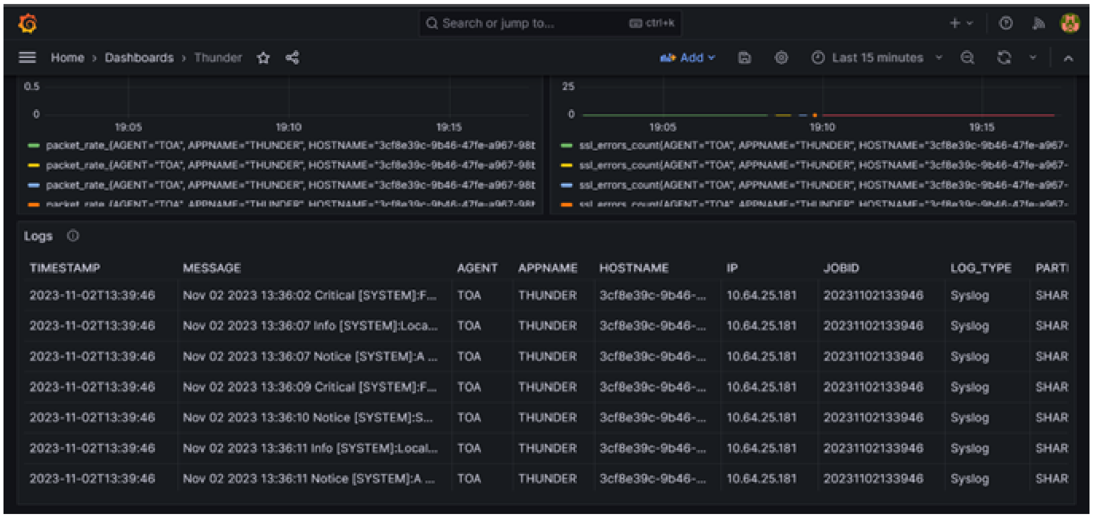

Home › Dashboards › Import dashboard
Dashboards
Playlists
Snapshots
Library panels
Import dashboard
Import dashboard from file or Grafana.com
Upload dashboard JSON file
Drag and drop here or click to browse
Accepted file types: .json, .txt
Import via grafana.com
Grafana.com dashboard URL or ID
Load
Import via panel json
Load
Cancel

e. Browse the downloaded Grafana dashboard file and click **Load**.

2. View the dashboard.

   To view the dashboard, navigate to **Menu** > **Dashboard**. All the logs are visible below the metrics as shown:

Figure 72 : Grafana Logs Dashboard


Search or jump to...
ctrl+k
Home › Dashboards › Thunder
Add
Last 15 minutes
0.5
0
19:05
19:10
19:15
packet_rate_{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe-a967-98t
packet_rate_{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe-a967-98t
packet_rate_{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe-a967-98t
25
0
19:05
19:10
19:15
ssl_errors_count{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe-a967-
ssl_errors_count{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe-a967-
ssl_errors_count{AGENT="TOA", APPNAME="THUNDER", HOSTNAME="3cf8e39c-9b46-47fe-a967-
Logs

TIMESTAMP	MESSAGE	AGENT	APPNAME	HOSTNAME	IP	JOBID	LOG_TYPE	PARTI
2023-11-02T13:39:46	Nov 02 2023 13:36:02 Critical [SYSTEM]:F...	TOA	THUNDER	3cf8e39c-9b46-...	10.64.25.181	20231102133946	Syslog	SHAR
2023-11-02T13:39:46	Nov 02 2023 13:36:07 Info [SYSTEM]:Loca...	TOA	THUNDER	3cf8e39c-9b46-...	10.64.25.181	20231102133946	Syslog	SHAR
2023-11-02T13:39:46	Nov 02 2023 13:36:07 Notice [SYSTEM]:A ...	TOA	THUNDER	3cf8e39c-9b46-...	10.64.25.181	20231102133946	Syslog	SHAR
2023-11-02T13:39:46	Nov 02 2023 13:36:09 Critical [SYSTEM]:F...	TOA	THUNDER	3cf8e39c-9b46-...	10.64.25.181	20231102133946	Syslog	SHAR
2023-11-02T13:39:46	Nov 02 2023 13:36:10 Notice [SYSTEM]:S...	TOA	THUNDER	3cf8e39c-9b46-...	10.64.25.181	20231102133946	Syslog	SHAR
2023-11-02T13:39:46	Nov 02 2023 13:36:11 Info [SYSTEM]:Local...	TOA	THUNDER	3cf8e39c-9b46-...	10.64.25.181	20231102133946	Syslog	SHAR
2023-11-02T13:39:46	Nov 02 2023 13:36:11 Notice [SYSTEM]:A ...	TOA	THUNDER	3cf8e39c-9b46-...	10.64.25.181	20231102133946	Syslog	SHAR

Splunk

To monitor the configured logs in Splunk Enterprise, perform the following steps:

1. Log in to Splunk Enterprise.
2. Create an HTTP Event Collector (HEC) for the logs.

   To use HEC, you need to configure at least one token. The token is used to authenticate and send data to Splunk.

   a. Navigate to **Settings** > **Data Inputs** > **HTTP Event Collector**.

   b. Click **New Token**.

   c. Enter the token name as 'collectorLog' and click **Next**.

   d. Select a source type as `_json` from the **Source Type** drop-down list box.

   e. Click **Create a new index**.

   f. Enter the name as 'thunder_log' and select the **Index Data Type** as **Events**. Click **Save**.

      The index will be add to the **Available Items** list box.

   g. Choose the newly created index ('thunder_logs') from the **Available Items** list box.

   h. Click **Review** to review the settings and then click **Submit**.

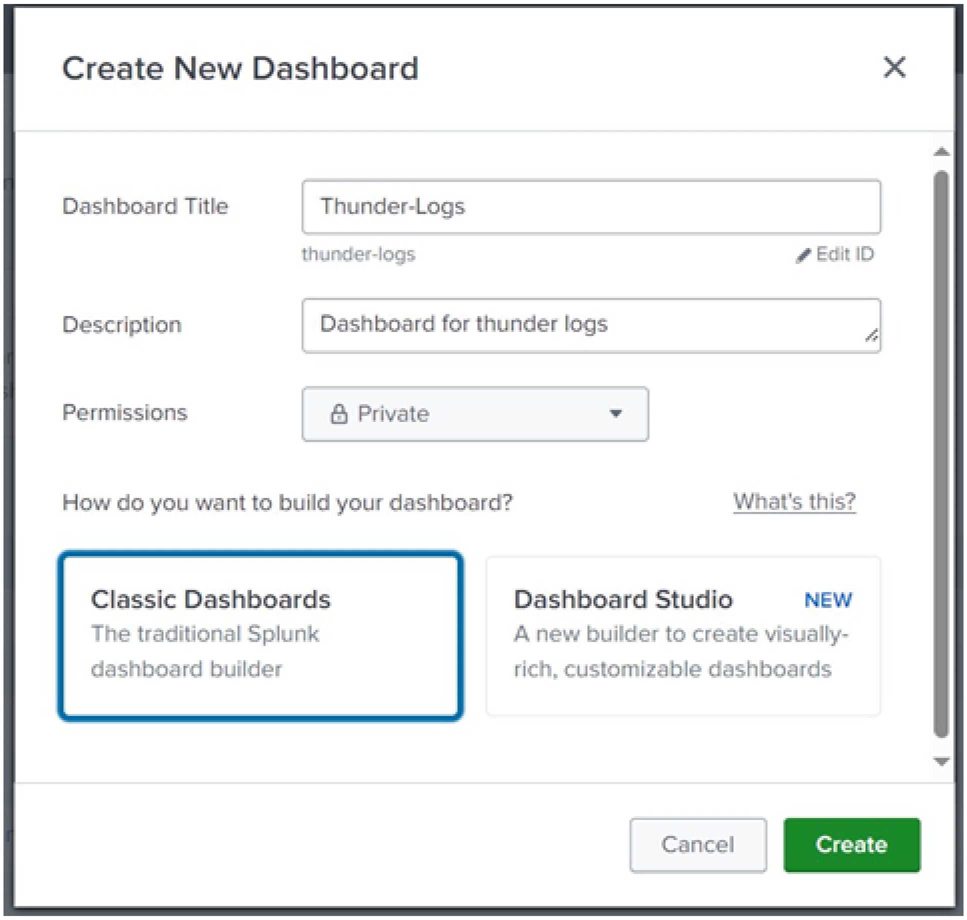
The token is created.

i. Store the token generated for later reference.

NOTE: If you have already created this token then ensure that the dashboard xml file contains the same index name.

3. Navigate to **Apps** > **Search & Reporting** > **Dashboard** and then click **Create New Dashboard**.

Figure 73 : Create New Dashboard



Create New Dashboard

Dashboard Title: Thunder-Logs
thunder-logs

Description: Dashboard for thunder logs

Permissions: Private

How do you want to build your dashboard?

Classic Dashboards
The traditional Splunk dashboard builder

Dashboard Studio NEW
A new builder to create visually-rich, customizable dashboards

4. On the **Create New Dashboard** form, perform the following steps:

   a. Enter a name for the dashboard in the **Dashboard Title** field.

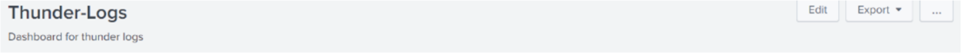
   b. Enter description in the **Description** field.

c. Select the appropriate permissions from **Permissions** drop-down menu.

d. Under **How do you want to build your dashboard?**, select **Classic Dashboard** framework, and click **Create**.

5. On the newly created dashboard, first click **Edit** and then click **Source**.

Figure 74 : Logs Dashboard



Thunder-Logs

Dashboard for thunder logs

6. Copy the XML code from the dashboard template file and paste it in the editor.

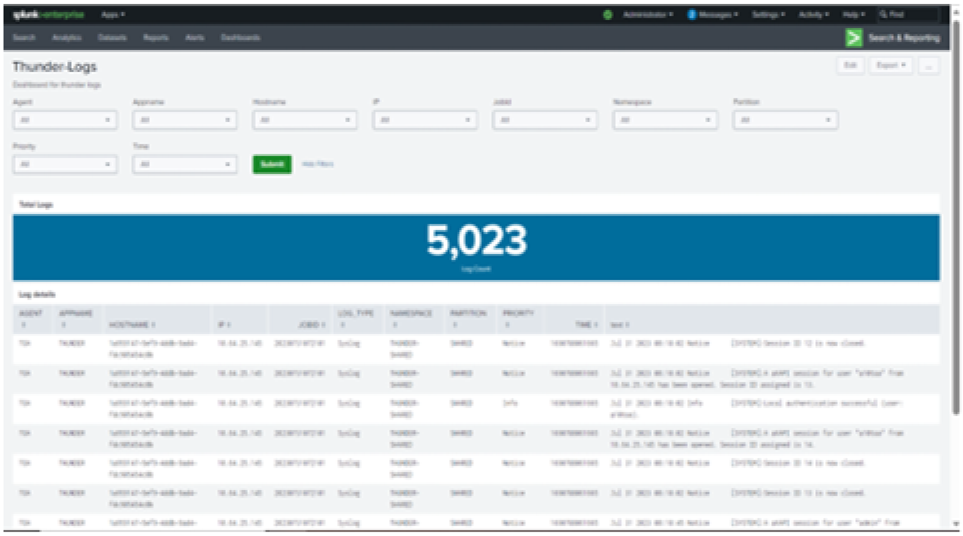
7. Edit the following tags:

- **label** - It must be same as the **Dashboard Title** entered in Step-4a.
- **description** - It must be same as the dashboard **Description** entered in Step-4b.
- **query** - The **index** mentioned in this tag must be same as the one in use.

8. Click **Save**.

9. Verify if the logs are displayed.

Figure 75 : Thunder-Logs Dashboard



GCP Logs Explorer

To monitor the configured logs in GCP Logs Explorer, perform the following steps:

1. View Thunder Logs
2. Configure Query Parameters
3. Create an Alert
4. Add Logs to Dashboard

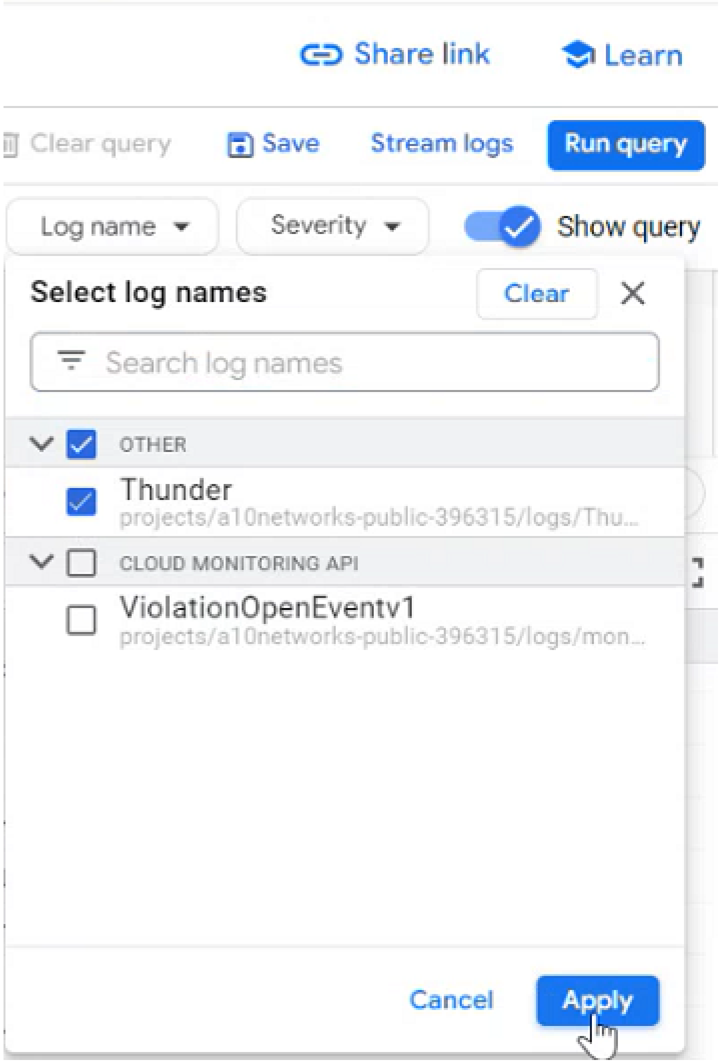
View Thunder Logs

1. Open Google Cloud Console and select the project you want to work with.
2. In the navigation menu, select **Logging**, and then navigate to **Logs Explorer**.

   The **Log Explorer** interface will be displayed. This interface allows you to retrieve logs, parse and analyze log data, and refine query parameters.

3. Click **Log name** drop down menu on the toolbar, select the log name `thunder` (default name of the log generated), and click **Apply** as shown in .

   Logs Explorer - Log name



Select log names

Clear

Search log names

☑ OTHER

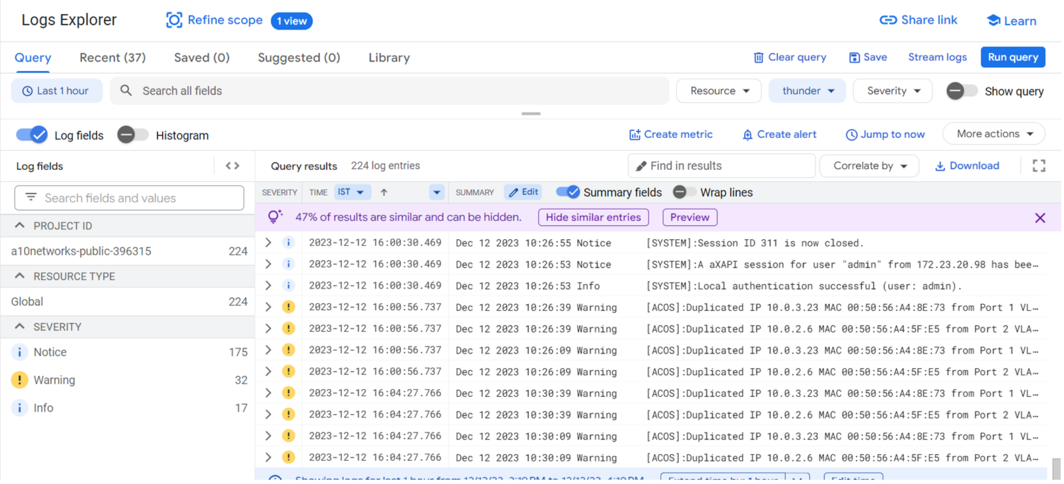
☑ Thunder
projects/a10networks-public-396315/logs/Thu...

□ CLOUD MONITORING API

□ ViolationOpenEventv1
projects/a10networks-public-396315/logs/mon...

The log is displayed in the **Query results** pane as shown in Figure 76.

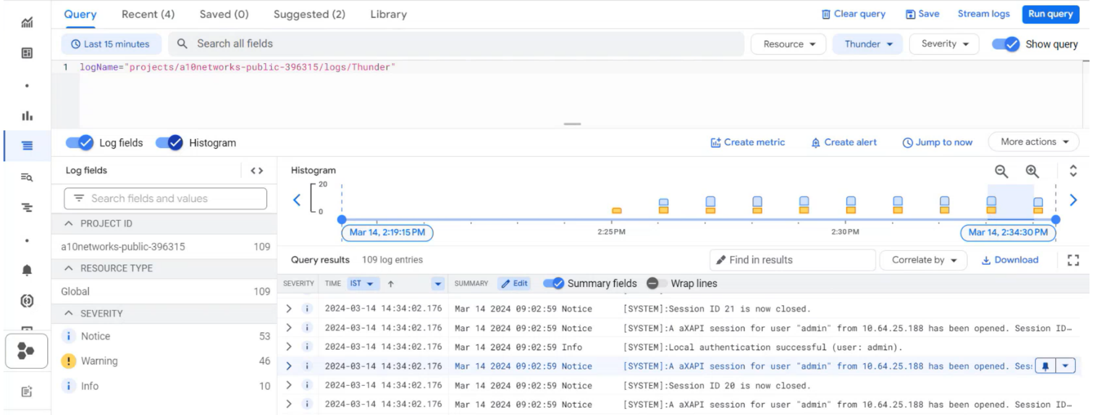
Figure 76 : Logs Explorer Interface



```
2023-12-12 16:00:30.469  Dec 12 2023 10:26:55 Notice    [SYSTEM]:Session ID 311 is now closed.
2023-12-12 16:00:30.469  Dec 12 2023 10:26:53 Notice    [SYSTEM]:A aXAPI session for user "admin" from 172.23.20.98 has bee…
2023-12-12 16:00:30.469  Dec 12 2023 10:26:53 Info      [SYSTEM]:Local authentication successful (user: admin).
2023-12-12 16:00:56.737  Dec 12 2023 10:26:39 Warning   [ACOS]:Duplicated IP 10.0.3.23 MAC 00:50:56:A4:8E:73 from Port 1 VL…
2023-12-12 16:00:56.737  Dec 12 2023 10:26:39 Warning   [ACOS]:Duplicated IP 10.0.2.6 MAC 00:50:56:A4:5F:E5 from Port 2 VLA…
2023-12-12 16:00:56.737  Dec 12 2023 10:26:09 Warning   [ACOS]:Duplicated IP 10.0.3.23 MAC 00:50:56:A4:8E:73 from Port 1 VL…
2023-12-12 16:00:56.737  Dec 12 2023 10:26:09 Warning   [ACOS]:Duplicated IP 10.0.2.6 MAC 00:50:56:A4:5F:E5 from Port 2 VLA…
2023-12-12 16:04:27.766  Dec 12 2023 10:30:39 Warning   [ACOS]:Duplicated IP 10.0.3.23 MAC 00:50:56:A4:8E:73 from Port 1 VL…
2023-12-12 16:04:27.766  Dec 12 2023 10:30:39 Warning   [ACOS]:Duplicated IP 10.0.2.6 MAC 00:50:56:A4:5F:E5 from Port 2 VLA…
2023-12-12 16:04:27.766  Dec 12 2023 10:30:09 Warning   [ACOS]:Duplicated IP 10.0.3.23 MAC 00:50:56:A4:8E:73 from Port 1 VL…
2023-12-12 16:04:27.766  Dec 12 2023 10:30:09 Warning   [ACOS]:Duplicated IP 10.0.2.6 MAC 00:50:56:A4:5F:E5 from Port 2 VLA…
```

Additionally, you can select **Histogram** in the **Results** toolbar to provide a visual representation of log data distribution. This also helps in the identification of patterns, anomalies, and trends within the log data.

Figure 77 : Logs Explorer - Histogram



```
logName="projects/a10networks-public-396315/logs/Thunder"
```

```
2024-03-14 14:34:02.176  Mar 14 2024 09:02:59 Notice   [SYSTEM]:Session ID 21 is now closed.
2024-03-14 14:34:02.176  Mar 14 2024 09:02:59 Notice   [SYSTEM]:A aXAPI session for user "admin" from 10.64.25.188 has been opened. Session ID…
2024-03-14 14:34:02.176  Mar 14 2024 09:02:59 Info     [SYSTEM]:Local authentication successful (user: admin).
2024-03-14 14:34:02.176  Mar 14 2024 09:02:59 Notice   [SYSTEM]:A aXAPI session for user "admin" from 10.64.25.188 has been opened. Ses:
2024-03-14 14:34:02.176  Mar 14 2024 09:02:59 Notice   [SYSTEM]:Session ID 20 is now closed.
2024-03-14 14:34:02.176  Mar 14 2024 09:02:59 Notice   [SYSTEM]:A aXAPI session for user "admin" from 10.64.25.188 has been opened. Session ID…
```

Configure Query Parameters

A query in **Logs Explorer** specifies parameters and conditions to retrieve specific log data, thereby aiding log analysis and troubleshooting. Following are the commonly configured query parameters:

- The **Severity** option in the **Log fields** pane allows you to filter log entries based on their severity level, enabling you to quickly identify and prioritize issues. The severity levels include DEBUG, INFO, WARNING, ERROR, and CRITICAL, representing varying degrees of importance and urgency.

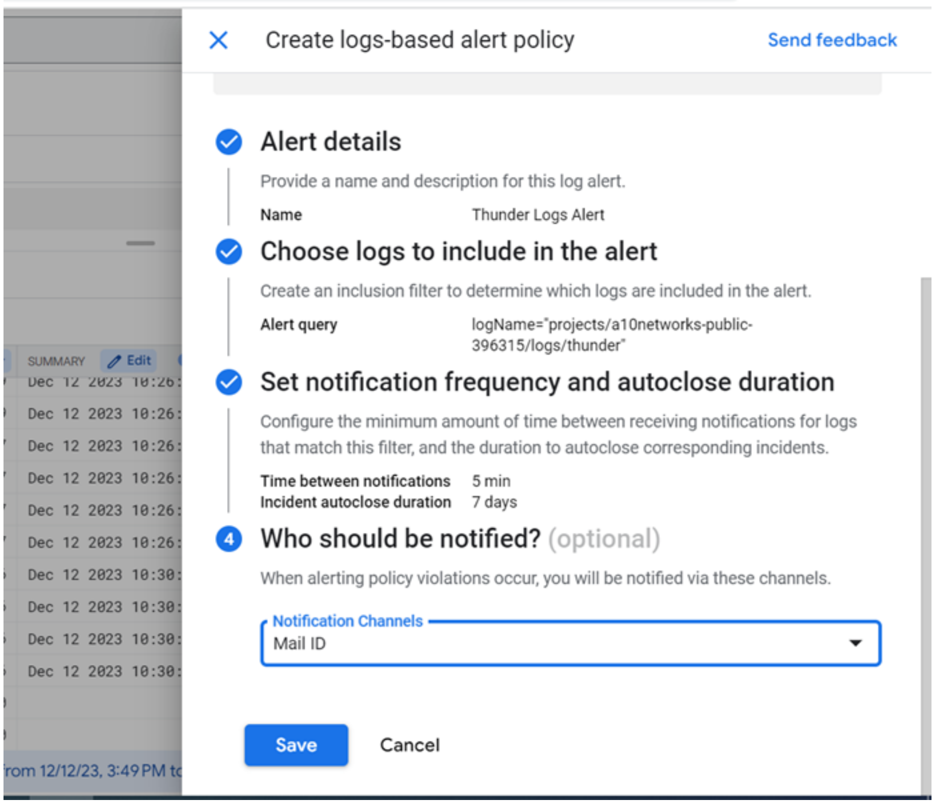
- The **Time-range selector** in the **Query** pane allows you to specify the time range for which you want to view the log data. You can select predefined time ranges (e.g., last hour, last 24 hours) or define a custom time range by specifying the start time and end time.
- The **Search-text box** in the **Query** pane allows you to perform text-based searches within logs, making it easier to find log entries containing specific information or events of interest. For example, entering **error** in the text box helps pinpoint logs related to errors in the application.

After configuring the query parameters, click **Save** in the **Query** pane to save the query i.e., store the specific set of parameters and conditions for future purpose.

Create an Alert

1. On the **Results** toolbar, click **Create alert**.
2. In the **Alert details** pane, enter the **policy name**, select an option from the **severity level** drop-down menu, and click **Next**.
3. In the **Choose logs to include in the alert** pane, check the configured query and log results by clicking **Preview logs**.

   The query for the thunder logs (created in the previous steps by specifying various filtering parameters) will be displayed in this pane. You can also edit the query in this pane. After editing the query, you can check the results by clicking **Preview logs**.
4. Click **Next**.
5. In the **Set notification frequency and autoclose duration** pane, select values for **Time Between Notification** and the **Incident autoclose duration**, and click **Next**.
6. In the **Who should be notified** pane, you can select one or more notification channels for the alert. If you already have an SMS or email notification channel configured, then you can select it from the list. Else, click **Manage notification channels** and add a notification channel. For more information, see Create and Manage Notification channels.
7. Click **Save**.



Create logs-based alert policy

Alert details

Provide a name and description for this log alert.

Name Thunder Logs Alert

Choose logs to include in the alert

Create an inclusion filter to determine which logs are included in the alert.

Alert query logName="projects/a10networks-public-396315/logs/thunder"

Set notification frequency and autoclose duration

Configure the minimum amount of time between receiving notifications for logs that match this filter, and the duration to autoclose corresponding incidents.

Time between notifications 5 min
Incident autoclose duration 7 days

4 Who should be notified? (optional)

When alerting policy violations occur, you will be notified via these channels.

Notification Channels
Mail ID

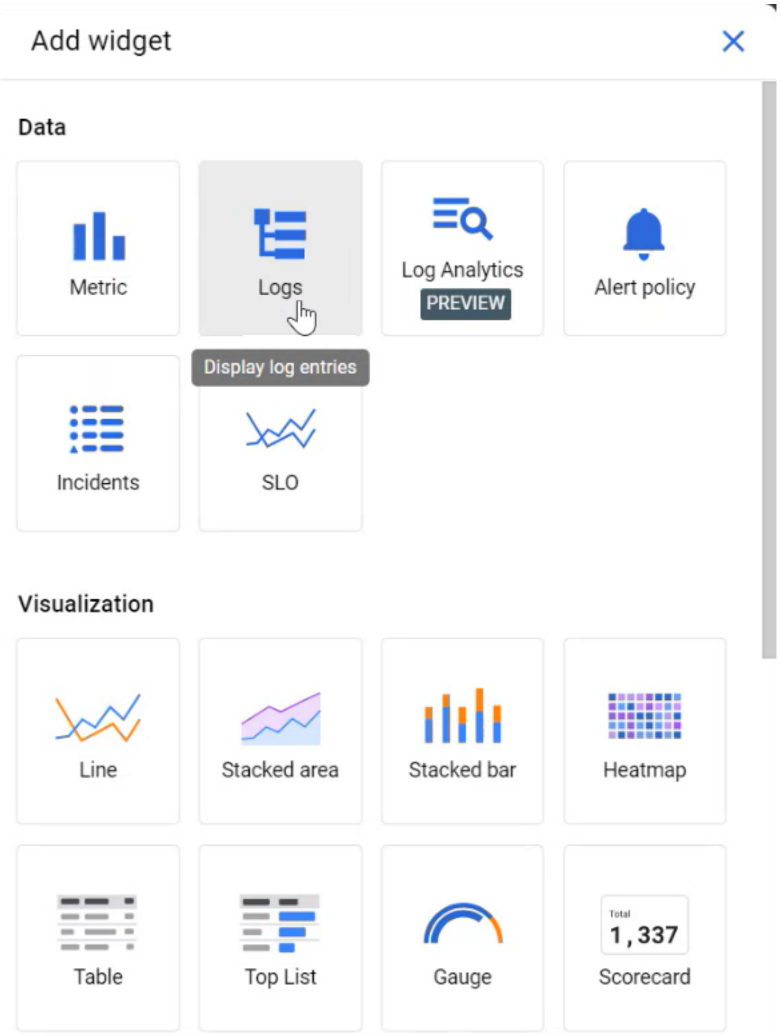
Your log-based alert policy is configured.

For more information, see Configure log-based alerts.

Add Logs to Dashboard

1. In the navigation panel, select **Monitoring**, and then click **Dashboards**.
2. On the **Dashboards Overview** page, click **Create Dashboard**.
3. Click the dashboard's title, enter a name for the dashboard, and click **Save**.
4. Click **+ Add Widget** and select **Logs** as shown in Figure 78.

Figure 78 : Dashboard - Add Widget - Log

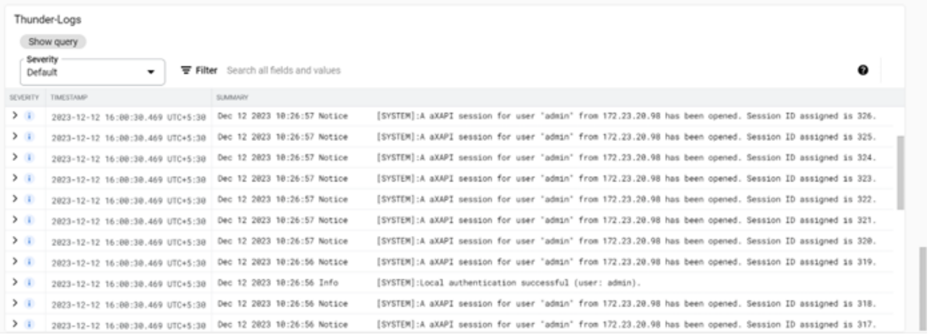


The **Configure Widget** page will be displayed.

5. On the **Configure Widget** page, click **Log name** drop-down menu, select the log name as **Thunder**, and click **Apply**.
6. Click **Severity** drop-down menu , select the severity from the list, and click **Apply**.

   The queried log will be added to the dashboard as shown in the following image.

Figure 79 : GCP Dashboard with Logs



```
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:57 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 326.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:57 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 325.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:57 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 324.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:57 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 323.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:57 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 322.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:57 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 321.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:57 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 320.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:56 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 319.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:56 Info     [SYSTEM]:Local authentication successful (user: admin).
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:56 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 318.
2023-12-12 16:00:30.469 UTC+5:30   Dec 12 2023 10:26:56 Notice   [SYSTEM]:A aXAPI session for user 'admin' from 172.23.20.98 has been opened. Session ID assigned is 317.
```

OCI Logs Search

To monitor the configured logs in Oracle Cloud Infrastructure (OCI) Logs Search, perform the following steps:

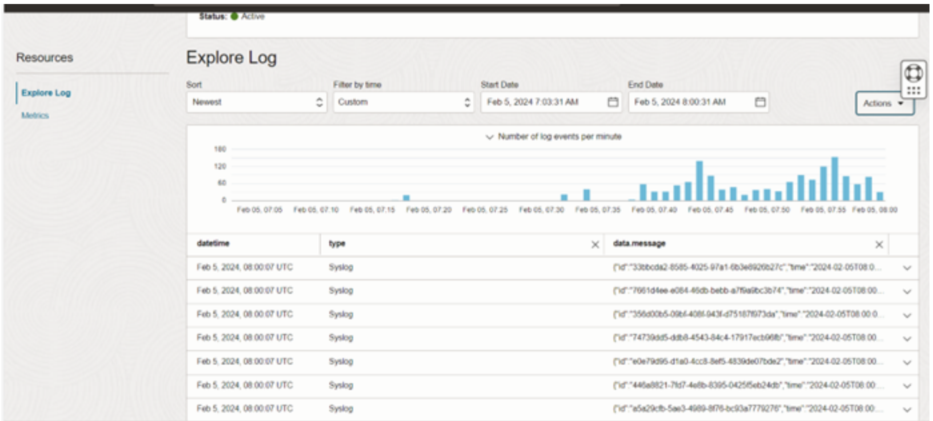
1. View Thunder Logs
2. Filter and Search Logs

View Thunder Logs

1. Log in to the OCI console, open the navigation menu and click **Observability & Management**.
2. Under **Logging**, click **Log Group**, and select your log group.
3. On the **Log Group** page, under **Resources**, click **Logs**.
4. From the list, select the log name for which the logs are being collected i.e., the logs for which OCID is mentioned in the config.json file for publishing purpose.

   The log data is displayed in the **Explore Log** area as show in Figure 80.

Figure 80 : OCI - Viewing log details



Explore Log

Sort: Newest

Filter by time: Custom

Start Date: Feb 5, 2024 7:03:31 AM

End Date: Feb 5, 2024 8:00:31 AM

datetime	type	data.message
Feb 5, 2024, 08:00:07 UTC	Syslog	{"id":"33bbcda2-8585-4025-97a1-6b3e8926b27c","time":"2024-02-05T08:0...
Feb 5, 2024, 08:00:07 UTC	Syslog	{"id":"7661d4ee-e084-46db-bebb-a7f9a9bc3b74","time":"2024-02-05T08:00...
Feb 5, 2024, 08:00:07 UTC	Syslog	{"id":"356d00b5-09bf-408f-943f-d75187f973da","time":"2024-02-05T08:00:0...
Feb 5, 2024, 08:00:07 UTC	Syslog	{"id":"74739dd5-ddb8-4543-84c4-17917ecb96fb","time":"2024-02-05T08:00...
Feb 5, 2024, 08:00:07 UTC	Syslog	{"id":"e0e79d95-d1a0-4cc8-8ef5-4839de07bde2","time":"2024-02-05T08:00...
Feb 5, 2024, 08:00:07 UTC	Syslog	{"id":"446a8821-7fd7-4e8b-8395-0425f5eb24db","time":"2024-02-05T08:00...
Feb 5, 2024, 08:00:07 UTC	Syslog	{"id":"a5a29cfb-5ae3-4989-8f76-bc93a7779276","time":"2024-02-05T08:00...

The **Explore Log** area provides various fields to help analyze log data effectively. Some of the common fields are:

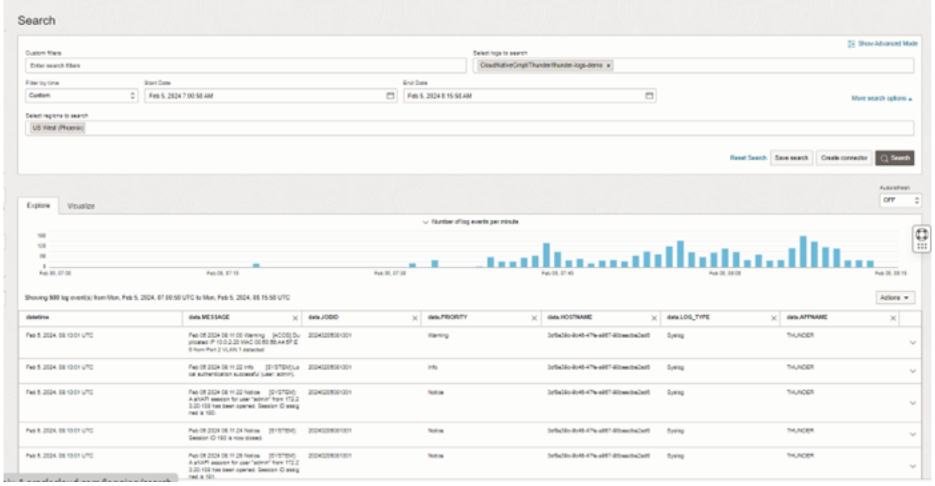
- **Sort** - This field allows you to arrange the log entries based on their timestamp (newest or oldest entries).
- **Filter by Time** - This field allows you to narrow down log data for a specified time period. You can select a predetermined time range from the list or select **Custom** to specify a date range using **Start Date** and **End Date** fields.
- **Actions** - This drop-down menu has the following options:
    - **Wrap-lines** - This option ensures that all content remains visible without the need for horizontal scrolling.
    - **Explore with Log Search** - This option allows you to view the log data on the **Search** page directly that provides various search and filtering options.

For more information on options, see Search and Filter logs.

Filter and Search Logs

The **Explore with Log Search** option under **Actions** drop-down menu provides powerful features such as advanced search syntax, aggregation functions, and visualization tools that help you perform complex analysis and investigation related to log data. By clicking this option, the log data can be viewed on the **Search** page as shown in Figure 81.

Figure 81 : OCI - Log Search



Showing 500 log event(s) from Mon, Feb 5, 2024, 07:00:58 UTC to Mon, Feb 5, 2024, 08:15:58 UTC

datetime	data.MESSAGE	data.JOBID	data.PRIORITY	data.HOSTNAME	data.LOG_TYPE	data.APPNAME
Feb 5, 2024, 08:10:01 UTC	Feb 05 2024 08:11:00 Warning [ACOS]:Duplicated IP 10.0.2.20 MAC 00:50:56:A4:8F:E5 from Port 2 VLAN 1 detected	[illegible]	Warning	[illegible]	Syslog	THUNDER
Feb 5, 2024, 08:10:01 UTC	Feb 05 2024 08:11:22 Info [SYSTEM]:Local authentication successful (user: admin).	[illegible]	Info	[illegible]	Syslog	THUNDER
Feb 5, 2024, 08:10:01 UTC	Feb 05 2024 08:11:22 Notice [SYSTEM]: A aXAPI session for user "admin" from 172.23.20.108 has been opened. Session ID assigned is 180.	[illegible]	Notice	[illegible]	Syslog	THUNDER
Feb 5, 2024, 08:10:01 UTC	Feb 05 2024 08:11:24 Notice [SYSTEM]: Session ID 180 is now closed.	[illegible]	Notice	[illegible]	Syslog	THUNDER
Feb 5, 2024, 08:10:01 UTC	Feb 05 2024 08:11:26 Notice [SYSTEM]: A aXAPI session for user "admin" from 172.23.20.108 has been opened. Session ID assigned is 181.	[illegible]	Notice	[illegible]	Syslog	THUNDER

The Search page is typically divided into two main parts, the filter section and the display section.

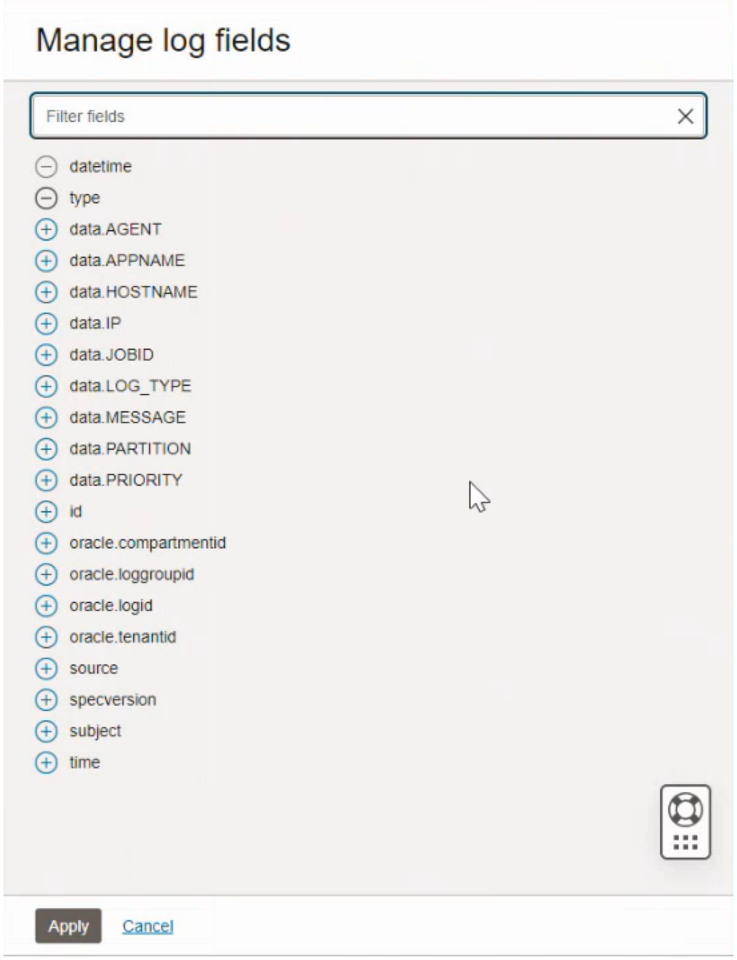
- **Filter Section** - It allows you to specify criteria to narrow down the search results. Some of the common features in this section are:
  - **Custom Filters** - This feature allows you to create custom filters based on attributes such as log source, severity level, specific keywords or phrases within log messages, or custom metadata fields associated with the log entries.

    Start typing in the text box to automatically display filtering options, along with the operators. For example, entering `d` displays filters starting with that letter. Select a filter from the list and use an operator to create a filter. For example, `data.JOBID= '<jobid_value>'`.
  - **Filter by Time** - This field allows you to narrow down log data for a specified time period. You can select a predetermined time range from the list or select **Custom** to specify a date range using **Start Date** and **End Date** fields.
  - **Select Regions to Search** - This filter allows you to filter logs based on specific regions. You can choose to include or exclude certain regions from the search scope based on your monitoring requirements.
  - **Save search** - This option allows you to save the above-mentioned filter settings for future purpose.
- **Display Section** - This section has the **Explore** and **Visualize** tabs that present log

data in accordance to the above-mentioned filters and search criteria.

- On the **Explore** tab, a graph displays the number of log events per minute. This tab displays a maximum of 100 search results. Some other commonly used options under this tab are:
  - **Manage log fields** under **Actions** menu allows you to add fields to the **Explore** tab. The new fields are appended to the right of the first three default fields (`datetime`, `type`, `data.message`). You can add a maximum of six log fields using this option.

Figure 82 : Manage log fields



Manage log fields

Filter fields ✕

⊖ datetime
⊖ type
⊕ data.AGENT
⊕ data.APPNAME
⊕ data.HOSTNAME
⊕ data.IP
⊕ data.JOBID
⊕ data.LOG_TYPE
⊕ data.MESSAGE
⊕ data.PARTITION
⊕ data.PRIORITY
⊕ id
⊕ oracle.compartmentid
⊕ oracle.loggroupid
⊕ oracle.logid
⊕ oracle.tenantid
⊕ source
⊕ specversion
⊕ subject
⊕ time

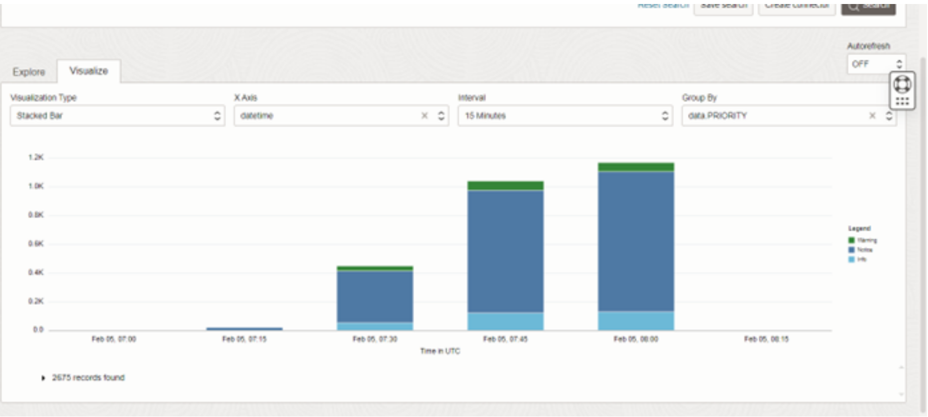
Apply Cancel

- **Export log data (JSON)** under **Actions** menu allows you to export log data to a JSON file that can be downloaded to your local storage.
- **JSON** tab (available after expanding any entry in the **data.message** field) allows you to view the log data fields and values, as well as collapse and expand nodes.
- **Before & After** tab (available after expanding any entry in the **data.message** field) provides context around a selected log entry by displaying logs that occur immediately before and after it.

For more information on options in the **Explore** tab, see Viewing Search Results.

- The **Visualize** tab allows you to visualize the log data as a chart to identify patterns, trends, and anomalies more effectively. For more information on options in this tab, see Visualizing Search Results.

Figure 83 : OCI - Visualize Logs



Visualization Type: Stacked Bar
X Axis: datetime
Interval: 15 Minutes
Group By: data.PRIORITY
Autorefresh: OFF

Troubleshoot

TOA Logging

TOA creates the `agent.log` file at the default directory `/var/log/thunder-observability-agent` path when the TOA cron is executed. This file contains the readable system logs from Thunder devices as per the configured frequency. It is used to troubleshoot any encountered issue.

The log file format contains logging level information. The logging level can be changed to DEBUG level for troubleshooting purpose.

A sample log file is shown below:

```
2023-05-29 06:47:01,831 - INFO - ##### TOA  ###### All Rights Reserved
@A10 Networks Inc ##### TOA #####
2023-05-29 06:48:02,063 - INFO - Job No              : 20230529104802.
2023-05-29 06:48:02,063 - INFO - Job Start Time      : 2023-05-29
10:48:02.006315+00:00.
2023-05-29 06:48:02,064 - WARNING - WARNING          : No log or metric is
enabled. To enable [metric, log set to [1]] in config.json.
2023-05-29 06:48:02,064 - INFO - Job Execution       : 0.058001 seconds.
2023-05-29 06:48:02,064 - INFO - Job End Time        : 2023-05-29
10:48:02.064316+00:00
2023-05-29 06:48:02,064 - INFO - Documentation       : www.a10networks.com or
https://github.com/a10networks/thunder-observability-agent.
2023-05-29 06:48:02,064 - INFO - ##### TOA  ###### All Rights Reserved
@A10 Networks Inc ##### TOA #####
2023-05-29 06:49:01,301 - INFO - Job No              : 20230529104901.
2023-05-29 06:49:01,301 - INFO - Job Start Time      : 2023-05-29
10:49:01.244429+00:00.
2023-05-29 06:49:01,301 - WARNING - WARNING          : No log or metric is
enabled. To enable [metric, log set to [1]] in config.json.
2023-05-29 06:49:01,301 - INFO - Job Execution       : 0.057536 seconds.
2023-05-29 06:49:01,302 - INFO - Job End Time        : 2023-05-29
10:49:01.301965+00:00
2023-05-29 06:49:01,302 - INFO - Documentation       : www.a10networks.com or
https://github.com/a10networks/thunder-observability-agent.

2023-05-29 06:49:01,302 - INFO - ##### TOA  ###### All Rights Reserved
@A10 Networks Inc ##### TOA #####
2023-05-29 06:50:01,533 - INFO - Job No              : 20230529105001.
2023-05-29 06:50:01,533 - INFO - Job Start Time      : 2023-05-29
10:50:01.477199+00:00.
2023-05-29 06:50:01,533 - ERROR - Error              : File not found or
corrupt. Please check file and path: [/usr/toaenv/thunder-observability-
agent/config.json]. Application config not found. Please check [config_
path] in main.properties.
```

```
2023-05-29 06:50:01,533 - INFO - Job Execution    : 0.056567 seconds.
2023-05-29 06:50:01,533 - INFO - Job End Time     : 2023-05-29
10:50:01.533766+00:00
2023-05-29 06:50:01,533 - INFO - Documentation    : www.a10networks.com or
https://github.com/a10networks/thunder-observability-agent.

2023-05-29 06:54:01,462 - INFO - ##### TOA  ###### All Rights Reserved
@A10 Networks Inc ##### TOA #####
2023-05-29 06:55:01,738 - INFO - Job No           : 20230529105501.
2023-05-29 06:55:01,738 - INFO - Job Start Time   : 2023-05-29
10:55:01.680906+00:00.
2023-05-29 06:55:01,738 - INFO - Log Provider     : VMWARE.
2023-05-29 06:55:01,738 - INFO - Log              : VMWARE_LOG.
2023-05-29 06:55:01,738 - INFO - Metric Provider  : VMWARE.
2023-05-29 06:55:01,739 - INFO - Metric           : VMWARE_METRIC.
2023-05-29 06:55:01,739 - INFO - No of Thunders   : 1 ['10.64.25.13'].
2023-05-29 06:55:01,739 - WARNING - WARNING       : No partitions found
for thunder [], setting to default 'SHARED'. Multiple L3V partition can be
configured as comma separated for example if we have partition 'P1' and
'P2' then we can define as ['partition' : ' Shared,P1,P2'] upto 20
partitions.
2023-05-29 06:55:01,739 - INFO - No of Partitions : 10.64.25.13 [Count: 1]
[shared].
2023-05-29 06:55:02,068 - INFO - Published Log    : 10.64.25.13 THUNDER-
SHARED [Count: 3].
2023-05-29 06:55:02,112 - INFO - Published Metric : 10.64.25.13 THUNDER
[Count: 2] [{'Memory Usage Percentage': 63.4, 'Disk Usage Percentage':
36}].
2023-05-29 06:55:02,151 - INFO - Published Metric : 10.64.25.13 THUNDER-
SHARED [Count: 10] [{'Server Errors Count': 0, 'Total Session Count': 0,
'SSL Errors Count': 0, 'Server Down Percentage': 0, 'CPU Usage Percentage
(Data)': 0.0, 'Total New Connection (Sec)': 0, 'Interface Down Count
(Data)': 0, 'Server Down Count': 0, 'Transactions Rate (Sec)': 0,
'Throughput Rate (Global/BPS)': 0}].
2023-05-29 06:55:02,161 - INFO - Job Execution    : 0.480912 seconds.
2023-05-29 06:55:02,161 - INFO - Job End Time     : 2023-05-29
10:55:02.161818+00:00
2023-05-29 06:55:02,162 - INFO - Documentation    : www.a10networks.com or
```

```
https://github.com/a10networks/thunder-observability-agent.

2023-05-29 07:00:02,016 - INFO - ##### TOA  ###### All Rights Reserved
@A10 Networks Inc ##### TOA #####
2023-05-29 07:01:01,258 - INFO - Job No           : 20230529110101.
2023-05-29 07:01:01,258 - INFO - Job Start Time   : 2023-05-29
11:01:01.201609+00:00.
2023-05-29 07:01:01,259 - INFO - Log Provider     : VMWARE.
2023-05-29 07:01:01,259 - INFO - Log              : VMWARE_LOG.
2023-05-29 07:01:01,259 - INFO - Metric Provider  : VMWARE.
2023-05-29 07:01:01,259 - INFO - Metric           : VMWARE_METRIC.
2023-05-29 07:01:01,259 - INFO - No of Thunders   : 1 ['10.64.25.13'].
2023-05-29 07:01:01,259 - INFO - No of Partitions : 10.64.25.13 [Count: 2]
[{'shared', 'p1'}].
2023-05-29 07:01:01,592 - INFO - Published Log    : 10.64.25.13 THUNDER-P1
[No Data Found].
2023-05-29 07:01:01,664 - INFO - Published Metric : 10.64.25.13 THUNDER-P1
[Count: 10] [{'Total Session Count': 0, 'Server Errors Count': 0, 'Server
Down Percentage': 0, 'SSL Errors Count': 0, 'Server Down Count': 0,
'Transactions Rate (Sec)': 0, 'Interface Down Count (Data)': 1,
'Throughput Rate (Global/BPS)': 0, 'Total New Connection (Sec)': 0, 'CPU
Usage Percentage (Data)': 0.0}].
2023-05-29 07:01:01,673 - INFO - Published Metric : 10.64.25.13 THUNDER-
SHARED [Count: 10] [{'SSL Errors Count': 0, 'Server Down Percentage': 0,
'Server Errors Count': 0, 'Total Session Count': 0, 'Server Down Count':
0, 'Interface Down Count (Data)': 0, 'CPU Usage Percentage (Data)': 0.0,
'Transactions Rate (Sec)': 0, 'Throughput Rate (Global/BPS)': 0, 'Total
New Connection (Sec)': 0}].
2023-05-29 07:01:01,682 - INFO - Published Metric : 10.64.25.13 THUNDER
[Count: 2] [{'Disk Usage Percentage': 36, 'Memory Usage Percentage':
66.8}].
2023-05-29 07:01:01,701 - INFO - Published Log    : 10.64.25.13 THUNDER-
SHARED [Count: 10].
2023-05-29 07:01:01,712 - INFO - Job Execution    : 0.51061 seconds.
2023-05-29 07:01:01,712 - INFO - Job End Time     : 2023-05-29
11:01:01.712219+00:00
2023-05-29 07:01:01,712 - INFO - Documentation    : www.a10networks.com or
https://github.com/a10networks/thunder-observability-agent.
```

```
2023-05-29 05:57:02,553 - INFO - ##### TOA  ###### All Rights Reserved
@A10 Networks Inc ##### TOA #####
2023-05-29 05:58:01,786 - INFO - Job No            : 20230529095801.
2023-05-29 05:58:01,787 - INFO - Job Start Time    : 2023-05-29
09:58:01.730452+00:00.
2023-05-29 05:58:01,787 - INFO - Log Provider      : VMWARE.
2023-05-29 05:58:01,787 - INFO - Log               : VMWARE_LOG.
2023-05-29 05:58:01,787 - INFO - Metric Provider   : VMWARE.
2023-05-29 05:58:01,787 - INFO - Metric            : VMWARE_METRIC.
2023-05-29 05:58:01,787 - INFO - No of Thunders    : 1 ['10.64.25.13'].
2023-05-29 05:58:01,787 - INFO - No of Partitions : 10.64.25.13 [Count: 1]
[*].
2023-05-29 05:58:02,848 - INFO - Published Metric : 10.64.25.13 THUNDER
[Count: 2] [{'Disk Usage Percentage': 35, 'Memory Usage Percentage':
61.6}].
2023-05-29 05:58:02,923 - INFO - Published Metric : 10.64.25.13 THUNDER-P1
[Count: 10] [{'Total Session Count': 0, 'Server Errors Count': 0, 'SSL
Errors Count': 0, 'Server Down Count': 0, 'Transactions Rate (Sec)': 0,
'Total New Connection (Sec)': 0, 'CPU Usage Percentage (Data)': 0.0,
'Server Down Percentage': 0, 'Interface Down Count (Data)': 1, 'Throughput
Rate (Global/BPS)': 0}].
2023-05-29 05:58:03,210 - INFO - Published Log     : 10.64.25.13 THUNDER-P5
[No Data Found].
2023-05-29 05:58:03,216 - INFO - Published Log     : 10.64.25.13 THUNDER-P8
[No Data Found].
2023-05-29 05:58:03,252 - INFO - Published Metric : 10.64.25.13 THUNDER-P7
[Count: 10] [{'Server Errors Count': 0, 'Total Session Count': 0, 'SSL
Errors Count': 0, 'Server Down Percentage': 0, 'Server Down Count': 0,
'Transactions Rate (Sec)': 0, 'Throughput Rate (Global/BPS)': 0, 'Total
New Connection (Sec)': 0, 'Interface Down Count (Data)': 0, 'CPU Usage
Percentage (Data)': 0.0}].
2023-05-29 05:58:03,288 - INFO - Published Log     : 10.64.25.13 THUNDER-
SHARED [Count: 6].
2023-05-29 05:58:03,379 - INFO - Published Log     : 10.64.25.13 THUNDER-
P19 [No Data Found].
2023-05-29 05:58:03,381 - INFO - Published Metric : 10.64.25.13 THUNDER-
P15 [Count: 10] [{'Total Session Count': 0, 'Server Errors Count': 0,
'Server Down Percentage': 0, 'SSL Errors Count': 0, 'Server Down Count':
0, 'Transactions Rate (Sec)': 0, 'CPU Usage Percentage (Data)': 0.0,
'Interface Down Count (Data)': 0, 'Throughput Rate (Global/BPS)': 0,
'Total New Connection (Sec)': 0}].
2023-05-29 05:58:03,422 - INFO - Published Metric : 10.64.25.13 THUNDER-P2
[Count: 10] [{'Server Down Count': 1, 'Server Down Percentage': 100.0,
'Server Errors Count': 0, 'Total Session Count': 0, 'SSL Errors Count': 0,
'Interface Down Count (Data)': 0, 'Transactions Rate (Sec)': 0, 'Total New
Connection (Sec)': 0, 'CPU Usage Percentage (Data)': 0.0, 'Throughput Rate
(Global/BPS)': 0}].
2023-05-29 05:58:03,502 - INFO - Published Metric : 10.64.25.13 THUNDER-
P11 [Count: 10] [{'SSL Errors Count': 0, 'Total Session Count': 0, 'Server
Errors Count': 0, 'Transactions Rate (Sec)': 0, 'Server Down Percentage':
0, 'Server Down Count': 0, 'Total New Connection (Sec)': 0, 'Interface
Down Count (Data)': 0, 'CPU Usage Percentage (Data)': 0.0, 'Throughput
Rate (Global/BPS)': 0}].
2023-05-29 05:58:03,547 - INFO - Published Metric : 10.64.25.13 THUNDER-P4
[Count: 10] [{'Total Session Count': 0, 'SSL Errors Count': 0, 'Server
Errors Count': 0, 'Total New Connection (Sec)': 0, 'Server Down
Percentage': 0, 'Server Down Count': 0, 'Transactions Rate (Sec)': 0,
'Throughput Rate (Global/BPS)': 0, 'CPU Usage Percentage (Data)': 0.0,
'Interface Down Count (Data)': 0}].
2023-05-29 05:58:03,608 - INFO - Published Log    : 10.64.25.13 THUNDER-
P11 [No Data Found].
2023-05-29 05:58:03,626 - INFO - Published Log    : 10.64.25.13 THUNDER-P1
[No Data Found].
2023-05-29 05:58:03,620 - INFO - Published Log    : 10.64.25.13 THUNDER-
P16 [No Data Found].
2023-05-29 05:58:03,674 - INFO - Published Metric : 10.64.25.13 THUNDER-P5
[Count: 10] [{'SSL Errors Count': 0, 'Server Errors Count': 0, 'Total
Session Count': 0, 'Server Down Count': 0, 'Transactions Rate (Sec)': 0,
'Server Down Percentage': 0, 'Total New Connection (Sec)': 0, 'CPU Usage
Percentage (Data)': 0.0, 'Interface Down Count (Data)': 0, 'Throughput
Rate (Global/BPS)': 0}].
2023-05-29 05:58:03,740 - INFO - Published Metric : 10.64.25.13 THUNDER-
P17 [Count: 10] [{'Server Errors Count': 0, 'SSL Errors Count': 0, 'Total
Session Count': 0, 'Server Down Count': 0, 'Transactions Rate (Sec)': 0,
'Server Down Percentage': 0, 'Throughput Rate (Global/BPS)': 0, 'CPU Usage
```

```
Percentage (Data)': 0.0, 'Total New Connection (Sec)': 0, 'Interface Down
Count (Data)': 0}].
2023-05-29 05:58:03,809 - INFO - Published Metric : 10.64.25.13 THUNDER-P8
[Count: 10] [{'Server Down Percentage': 0, 'Server Errors Count': 0, 'SSL
Errors Count': 0, 'Total Session Count': 0, 'Transactions Rate (Sec)': 0,
'Server Down Count': 0, 'Total New Connection (Sec)': 0, 'CPU Usage
Percentage (Data)': 0.0, 'Throughput Rate (Global/BPS)': 0, 'Interface
Down Count (Data)': 0}].
2023-05-29 05:58:04,082 - INFO - Published Log    : 10.64.25.13 THUNDER-P9
[No Data Found].
2023-05-29 05:58:04,248 - INFO - Published Metric : 10.64.25.13 THUNDER-
P19 [Count: 10] [{'Total Session Count': 0, 'Server Down Count': 0,
'Server Errors Count': 0, 'SSL Errors Count': 0, 'Transactions Rate
(Sec)': 0, 'Server Down Percentage': 0, 'CPU Usage Percentage (Data)':
0.0, 'Throughput Rate (Global/BPS)': 0, 'Interface Down Count (Data)': 0,
'Total New Connection (Sec)': 0}].
2023-05-29 05:58:04,250 - INFO - Published Metric : 10.64.25.13 THUNDER-P9
[Count: 10] [{'Server Down Count': 3, 'Server Down Percentage': 100.0,
'Server Errors Count': 0, 'Total Session Count': 0, 'SSL Errors Count': 0,
'Total New Connection (Sec)': 0, 'Transactions Rate (Sec)': 0, 'Interface
Down Count (Data)': 0, 'CPU Usage Percentage (Data)': 0.0, 'Throughput
Rate (Global/BPS)': 0}].
2023-05-29 05:58:04,258 - INFO - Published Metric : 10.64.25.13 THUNDER-P3
[Count: 10] [{'Server Down Percentage': 100.0, 'Server Down Count': 2,
'Transactions Rate (Sec)': 0, 'SSL Errors Count': 0, 'Total Session
Count': 0, 'Server Errors Count': 0, 'Total New Connection (Sec)': 0,
'Throughput Rate (Global/BPS)': 0, 'CPU Usage Percentage (Data)': 0.0,
'Interface Down Count (Data)': 0}].
2023-05-29 05:58:04,260 - INFO - Published Log    : 10.64.25.13 THUNDER-
P12 [No Data Found].
2023-05-29 05:58:04,267 - INFO - Published Metric : 10.64.25.13 THUNDER-
P13 [Count: 10] [{'Total Session Count': 0, 'Server Down Percentage': 0,
'Server Errors Count': 0, 'SSL Errors Count': 0, 'Transactions Rate
(Sec)': 0, 'Server Down Count': 0, 'Total New Connection (Sec)': 0, 'CPU
Usage Percentage (Data)': 0.0, 'Interface Down Count (Data)': 0,
'Throughput Rate (Global/BPS)': 0}].
2023-05-29 05:58:04,308 - INFO - Published Log    : 10.64.25.13 THUNDER-P4
[No Data Found].
```

```
2023-05-29 05:58:04,377 - INFO - Published Log    : 10.64.25.13 THUNDER-
P18 [No Data Found].
2023-05-29 05:58:04,396 - INFO - Published Metric : 10.64.25.13 THUNDER-
SHARED [Count: 10] [{'Server Errors Count': 0, 'Total Session Count': 0,
'SSL Errors Count': 0, 'Transactions Rate (Sec)': 0, 'Server Down Count':
0, 'CPU Usage Percentage (Data)': 0.0, 'Throughput Rate (Global/BPS)': 0,
'Server Down Percentage': 0, 'Total New Connection (Sec)': 0, 'Interface
Down Count (Data)': 0}].
2023-05-29 05:58:04,468 - INFO - Published Log    : 10.64.25.13 THUNDER-P7
[No Data Found].
2023-05-29 05:58:04,469 - INFO - Published Metric : 10.64.25.13 THUNDER-
P16 [Count: 10] [{'Server Errors Count': 0, 'SSL Errors Count': 0, 'Total
Session Count': 0, 'Server Down Percentage': 0, 'Server Down Count': 0,
'CPU Usage Percentage (Data)': 0.0, 'Throughput Rate (Global/BPS)': 0,
'Interface Down Count (Data)': 0, 'Total New Connection (Sec)': 0,
'Transactions Rate (Sec)': 0}].
2023-05-29 05:58:04,472 - INFO - Published Log    : 10.64.25.13 THUNDER-P2
[No Data Found].
2023-05-29 05:58:04,474 - INFO - Published Log    : 10.64.25.13 THUNDER-
P15 [No Data Found].
2023-05-29 05:58:04,599 - INFO - Published Log    : 10.64.25.13 THUNDER-
P17 [No Data Found].
2023-05-29 05:58:04,607 - INFO - Published Log    : 10.64.25.13 THUNDER-
P10 [No Data Found].
2023-05-29 05:58:04,624 - INFO - Published Metric : 10.64.25.13 THUNDER-
P10 [Count: 10] [{'Transactions Rate (Sec)': 0, 'SSL Errors Count': 0,
'Total Session Count': 0, 'Server Down Percentage': 0, 'Server Down
Count': 0, 'Server Errors Count': 0, 'Total New Connection (Sec)': 0,
'Interface Down Count (Data)': 0, 'Throughput Rate (Global/BPS)': 0, 'CPU
Usage Percentage (Data)': 0.0}].
2023-05-29 05:58:04,742 - INFO - Published Log    : 10.64.25.13 THUNDER-
P14 [No Data Found].
2023-05-29 05:58:04,844 - INFO - Published Metric : 10.64.25.13 THUNDER-
P18 [Count: 10] [{'SSL Errors Count': 0, 'Total Session Count': 0, 'Server
Errors Count': 0, 'Server Down Count': 0, 'Server Down Percentage': 0,
'Throughput Rate (Global/BPS)': 0, 'Interface Down Count (Data)': 0,
'Transactions Rate (Sec)': 0, 'Total New Connection (Sec)': 0, 'CPU Usage
Percentage (Data)': 0.0}].
```

```
2023-05-29 05:58:04,910 - INFO - Published Log    : 10.64.25.13 THUNDER-P3
[No Data Found].
2023-05-29 05:58:04,919 - INFO - Published Log    : 10.64.25.13 THUNDER-
P13 [No Data Found].
2023-05-29 05:58:04,922 - INFO - Published Metric : 10.64.25.13 THUNDER-
P14 [Count: 10] [{'Server Down Percentage': 0, 'Total New Connection
(Sec)': 0, 'Transactions Rate (Sec)': 0, 'Total Session Count': 0, 'Server
Errors Count': 0, 'SSL Errors Count': 0, 'Server Down Count': 0, 'CPU
Usage Percentage (Data)': 0.0, 'Interface Down Count (Data)': 0,
'Throughput Rate (Global/BPS)': 0}].
2023-05-29 05:58:04,942 - INFO - Published Log    : 10.64.25.13 THUNDER-P6
[No Data Found].
2023-05-29 05:58:04,978 - INFO - Published Metric : 10.64.25.13 THUNDER-
P12 [Count: 10] [{'Server Errors Count': 0, 'Server Down Percentage': 0,
'Total Session Count': 0, 'SSL Errors Count': 0, 'CPU Usage Percentage
(Data)': 0.0, 'Server Down Count': 0, 'Transactions Rate (Sec)': 0,
'Interface Down Count (Data)': 0, 'Throughput Rate (Global/BPS)': 0,
'Total New Connection (Sec)': 0}].
2023-05-29 05:58:05,002 - INFO - Published Metric : 10.64.25.13 THUNDER-P6
[Count: 10] [{'Throughput Rate (Global/BPS)': 0, 'Server Down Percentage':
0, 'Server Errors Count': 0, 'Total New Connection (Sec)': 0, 'SSL Errors
Count': 0, 'Transactions Rate (Sec)': 0, 'Total Session Count': 0, 'Server
Down Count': 0, 'CPU Usage Percentage (Data)': 0.0, 'Interface Down Count
(Data)': 0}].
2023-05-29 05:58:05,013 - INFO - Job Execution    : 3.282716 seconds.
2023-05-29 05:58:05,013 - INFO - Job End Time     : 2023-05-29
09:58:05.013168+00:00
2023-05-29 05:58:05,013 - INFO - Documentation    : www.a10networks.com or
https://github.com/a10networks/thunder-observability-agent.
```

Examples

The following topics are covered:

AWS

Borse Inc. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their AWS platform. The instances are configured as an ADC load balancer for their gaming applications named [Pokers]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using AWS CloudWatch and to get an email alert when the aggregated CPU usage crosses 75% so that proper action can be taken on time.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Pokers_TH
Password	Thunder@Borse@3201
Resource_Name	North_Virginia_Online_Pokers_TH
resource_id	i-1234567890abcdef0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Pokers_TH2
Password	Thunder@Borse@3202
Resource_Name	North_Virginia_Online_Pokers_TH2

Parameter	Description
resource_id	i-1234567890uvwxyz0
Thunder instance	3
User Name	Online_Pokers_TH3
Password	Thunder@Borse@3203
Resource_Name	vth-auto-scale-group
AWS Monitoring details	
aws_log_group_name	Thunder
aws_access_key_id	AKIA5VU3P46JEI7OQU54
aws_secret_access_key	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
region	us-east-1

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on AWS CloudWatch:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

```
apt update
apt-get install python3.10
apt install python3-pip
apt install cron
apt install rsyslog
```

2. Install TOA.

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install thunder_observability_agent
```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending on the type of Thunder instance:

Single instance

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Pokers_TH",
             "password": "Thunder@Borse@3201",
             "resource_id": "i-1234567890abcdef0",
             "active_partitions": "shared"
        }]
}
```

Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Pokers_TH",
          "password": "Thunder@Borse@3201",
          "resource_id": i-1234567890abcdef0,
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Pokers_TH2",
          "password": "Thunder@Borse@3202",
          "resource_id": "i-1234567890uvwxyz0",
          "active_partitions": "P1"
        }]
}
```

Auto Scale instance

```
{
      "autoscale" : 1,
      "provider" : "AWS",
      "thunders": [{
             "username": "Online_Pokers_TH",
             "password": "Thunder@Borse@3201",
             "resource_id": "vth-auto-scale-group-name",
             "active_partitions": "shared"
       }]
}
```

b. Update the following configurations in the `/root/.aws/config` file.

```
[default]
     region = us-east-1
     output = json
```

c. Update the AWS credentials in the `/root/.aws/credentials` file.

```
[default]
     aws_access_key_id = AKIA5VU3P46JEI7OQU54
     aws_secret_access_key = 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
```

d. Update AWS configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "aws_provider": 1,
  "aws_metric": 1,
  "aws_cpu": 1,
  "aws_memory": 1,
  "aws_disk": 1,
  "aws_throughput": 1,
  "aws_interfaces": 1,
  "aws_cps": 1,
  "aws_tps": 1,
  "aws_server_down_count": 1,
  "aws_server_down_percentage": 1,
  "aws_ssl_cert": 1,
  "aws_server_error": 1,
  "aws_sessions": 1,
  "aws_packet_rate": 1,
  "aws_packet_drop": 1,
  "aws_log": 1,
  "aws_log_group_name": "Thunder",
 }
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

Azure

ABC Corp. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their Azure platform. The instances are configured as an ADC load balancer for their gaming applications named [Football]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using Azure Application Insight and Log Analytics Workspace and to get an email alert when the aggregated CPU usage crosses 75% so that proper action can be taken on time.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Football_TH
Password	Thunder@ABC@3201
Resource_Name	North_Virginia_Online_Football_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Football_TH2
Password	Thunder@ABC@3202
Resource_Name	North_Virginia_Online_Football_TH2
resource_id	i-1234567890rstuvw0
Thunder instance	3
User Name	Online_Football_TH3
Password	Thunder@ABC@3203
Resource_Name	vth-auto-scale-group
Azure Monitoring details	
azure_location	southcentralus
azure_metric_resource_id	/subscriptions/07d34b9b-61e3-475a-abbc-006b16812a3e/resourceGroups/vth-rg6/providers/microsoft.insights/components/vth-vmss-app-insights
azure_workspace_primary_key	tewPsyMYkdGOThRjEyl**F8CzJ49ZRgw==

Parameter	Description
azure_client_id	10724xxx-xxx-xxxx-xxxx-xxxx2c14726d
azure_secret_id	9-xxx~jIxxxEVyxxxxHNxxxOwv_xxxxZLxxxTM
azure_tenant_id	91d27xxx-xxxx-xxxx-xxxx-xxxxf81fcb2f
azure_log_workspace_id	dcfd7xxx-xxxx-xxxx-xxxx-xxxxf81fc991

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on the Azure platform:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

```
apt update
apt-get install python3.10
apt install python3-pip
apt install cron
apt install rsyslog
```

2. Install TOA.

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install thunder_observability_agent
```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending upon the type of Thunder instance:

   Single instance

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Football_TH",
             "password": "Thunder@ABC@3201",
             "resource_id": "i-1234567890lmnopq0"
             "active_partitions": "shared"
        }]
}
```

Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Football_TH",
          "password": "Thunder@ABC@3201",
          "resource_id": "i-1234567890lmnopq0"
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Football_TH2",
          "password": "Thunder@ABC@3202",
          "resource_id": "i-1234567890rstuvw0"
          "active_partitions": "shared"
        }]
}
```

Auto Scale (VMSS) instance

```
{
      "autoscale" : 1,
      "provider" : "Azure",
      "thunders": [{
             "username": "Online_Football_TH3",
             "password": "Thunder@ABC@3203",
             "resource_id": "vth-auto-scale-group"
             "active_partitions": "shared"
       }]
}
```

b. Update the Azure credentials in the `/root/.azure/credentials` file.

```
       azure_workspace_primary_key =
"tewPsyMYkdGOThRjEyl***********************************************
*********F8CzJ49ZRgw=="
       azure_client_id = "10724xxx-xxx-xxxx-xxxx-xxxx2c14726d"
       azure_secret_id = "9-xxx~jIxxxEVyxxxxHNxxxOwv_xxxxZLxxxTM"
       azure_tenant_id = "91d27xxx-xxxx-xxxx-xxxx-xxxxf81fcb2f"
       azure_location = "southcentralus"
```

c. Update Azure configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "azure_provider": 1,
  "azure_metric": 1,
  "azure_metric_resource_id": "/subscriptions/07d34b9b-61e3-475a-
abbc-006b16812a3e/resourceGroups/vth-
rg6/providers/microsoft.insights/components/vth-vmss-app-insights",
  "azure_cpu": 1,
  "azure_memory": 1,
  "azure_disk": 1,
  "azure_throughput": 1,
  "azure_interfaces": 1,
  "azure_cps": 1,
  "azure_tps": 1,
  "azure_server_down_count": 1,
  "azure_server_down_percentage": 1,
  "azure_ssl_cert": 1,
  "azure_server_error": 1,
  "azure_sessions": 1,
  "azure_packet_rate": 1,
  "azure_packet_drop": 1,
  "azure_log": 1,
  "azure_log_workspace_id": "dcfd7xxx-xxxx-xxxx-xxxx-xxxxf81fc991"
}
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

VMware

LMQ Corp. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their VMware platform. The instances are configured as an ADC load balancer for their gaming applications named [Baseball]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using VMware vRealize Operations Manager (vROps) and vRealize Log Insight (vRLI) and to get an email alert when the aggregated CPU usage crosses 75% so that proper action can be taken on time.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Baseball_TH
Password	Thunder@LMQ@3201
Resource_Name	North_Virginia_Online_Baseball_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Baseball_TH2
Password	Thunder@LMQ@3202
Resource_Name	North_Virginia_Online_Baseball_TH2
resource_id	i-1234567890rstuvw0
VMware Monitoring details	
vRLI IP	10.22.32.11
vROPs IP	10.22.32.12

Parameter	Description
vROPs User Name	vROPsAdmin
vROPs Password	vROPs@Borse@3212

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on the VMware platform:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

```
apt update
apt-get install python3.10
apt install python3-pip
apt install cron
apt install rsyslog
```

2. Install TOA.

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install thunder_observability_agent
```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending upon the type of Thunder instance:

   Single instance

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Baseball_TH",
             "password": "Thunder@LMQ@3201",
             "resource_id": "i-1234567890lmnopq0",
             "active_partitions": "shared"
        }]
}
```

Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Baseball_TH",
          "password": "Thunder@LMQ@3201",
          "resource_id": "i-1234567890lmnopq0",
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Baseball_TH2",
          "password": "Thunder@LMQ@3202",
          "resource_id": "i-1234567890rstuvw0",
          "active_partitions": "shared"
        }]
}
```

b. Update the VMware credentials in the `/root/.vmware/credentials` file.

```
vmware_vrops_username = vROPsAdmin
vmware_vrops_password = vROPs@Borse@3212
```

c. Update VMware configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "vmware_provider": 1,
  "vmware_metric": 1,
  "vmware_vrops_host": "10.22.32.12",
  "vmware_cpu": 1,
  "vmware_memory": 1,
  "vmware_disk": 1,
  "vmware_throughput": 1,
  "vmware_interfaces": 1,
  "vmware_cps": 1,
  "vmware_tps": 1,
  "vmware_server_down_count": 1,
  "vmware_server_down_percentage": 1,
  "vmware_ssl_cert": 1,
  "vmware_server_error": 1,
  "vmware_sessions": 1,
  "vmware_packet_rate": 1,
  "vmware_packet_drop": 1,
  "vmware_log": 1,
  "vmware_vrli_host": "10.22.32.11"
 }
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

Elasticsearch

LMQ Corp. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their Elasticsearch platform. The instances are configured as an ADC load balancer for their gaming applications named [Baseball]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using Elasticsearch and Kibana.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Baseball_TH
Password	Thunder@LMQ@3201
Resource_Name	North_Virginia_Online_Baseball_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Baseball_TH2
Password	Thunder@LMQ@3202
Resource_Name	North_Virginia_Online_Baseball_TH2
resource_id	i-1234567890rstuvw0
Elasticsearch Monitoring details	
Elasticsearch User Name	Elastic
Elasticsearch Password	BWFAN28DOPy8jpxh8tJQ
Elasticsearch Host	127.0.0.0:9200

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on Elasticsearch:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

```
apt update
apt-get install python3.10
apt install python3-pip
apt install cron
apt install rsyslog
```

2. Install TOA.

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install thunder_observability_agent
```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending on the type of Thunder instance:

Single instance

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Baseball_TH",
             "password": "Thunder@LMQ@3201",
             "resource_id": "i-1234567890lmnopq0",
             "active_partitions": "shared"
        }]
}
```

Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Baseball_TH",
          "password": "Thunder@LMQ@3201",
          "resource_id": i-1234567890lmnopq0,
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Baseball_TH2",
          "password": "Thunder@LMQ@3202",
          "resource_id": "i-1234567890rstuvw0",
          "active_partitions": "shared"
        }]
}
```

b. Update the Elasticsearch credentials in the `/root/.elasticsearch/credentials` file.

```
username = elastic
password = BWFAN28DOPy8jpxh8tJQ
```

c. Update Elasticsearch configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "es_provider": 1,
  "es_metric": 1,
  "es_host": "127.0.0.0:9200",
  "es_cpu": 1,
  "es_memory": 1,
  "es_disk": 1,
  "es_throughput": 1,
  "es_interfaces": 1,
  "es_cps": 1,
  "es_tps": 1,
  "es_server_down_count": 1,
  "es_server_down_percentage": 1,
  "es_ssl_cert": 1,
  "es_server_error": 1,
  "es_sessions": 1,
  "es_packet_rate": 1,
  "es_packet_drop": 1,
  "es_log": 1
   }
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

Prometheus

LMQ Corp. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their Prometheus platform. The instances are configured as an ADC load balancer for their gaming applications named [Baseball]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using Prometheus, Pushgateway & Grafana.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51

Parameter	Description
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Baseball_TH
Password	Thunder@LMQ@3201
Resource_Name	North_Virginia_Online_Baseball_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Baseball_TH2
Password	Thunder@LMQ@3202
Resource_Name	North_Virginia_Online_Baseball_TH2
resource_id	i-1234567890rstuvw0
Prometheus Monitoring details	
Pushgateway User Name	admin
Pushgateway Password	pushgateway123
Pushgateway Host	127.0.0.0:9091

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on Prometheus:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

```
apt update
apt-get install python3.10
apt install python3-pip
apt install cron
apt install rsyslog
```

2. Install TOA.

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install thunder_observability_agent
```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending on the type of Thunder instance:

   Single instance

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Baseball_TH",
             "password": "Thunder@LMQ@3201",
             "resource_id": "i-1234567890lmnopq0",
             "active_partitions": "shared"
        }]
}
```

   Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Baseball_TH",
          "password": "Thunder@LMQ@3201",
          "resource_id": i-1234567890lmnopq0,
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Baseball_TH2",
          "password": "Thunder@LMQ@3202",
          "resource_id": "i-1234567890rstuvw0",
          "active_partitions": "shared"
        }]
}
```

b. Update the Pushgateway credentials in the `/root/.pushgateway/credentials` file.

```
username = admin
password = pushgateway123
```

c. Update Pushgateway configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "pushgateway_provider": 1,
  "pushgateway_metric": 1,
  "pushgateway_host": "127.0.0.0:9091",
  "pushgateway_cpu": 1,
  "pushgateway_memory": 1,
  "pushgateway_disk": 1,
  "pushgateway_throughput": 1,
  "pushgateway_interfaces": 1,
  "pushgateway_cps": 1,
  "pushgateway_tps": 1,
  "pushgateway_server_down_count": 1,
  "pushgateway_server_down_percentage": 1,
  "pushgateway_ssl_cert": 1,
  "pushgateway_server_error": 1,
  "pushgateway_sessions": 1,
  "pushgateway_packet_rate": 1,
  "pushgateway_packet_drop": 1,
  "pushgateway_log": 1
   }
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

Splunk

XYZ Corp. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their Splunk platform. The instances are configured as an ADC load balancer for their gaming applications named [Volleyball]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using the Splunk dashboard and Splunk Analytics. Additionally, the client also wants to get an email alert when the aggregated CPU usage exceeds 75% to take an appropriate action.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Volleyball_TH
Password	Thunder@XYZ@3201
Resource_Name	North_Virginia_Online_Volleyball_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Volleyball_TH2
Password	Thunder@XYZ@3202
Resource_Name	North_Virginia_Online_Volleyball_TH2
resource_id	i-1234567890rstuvw0
Splunk Monitoring details	
token_log	2acdaae2a-0497-4a6c-97b7-b155e79aa88
token_metric	f944d49-37f4-4bba-a2f6-df0cd-be86fcbd
splunk_host	127.0.0.0:8088

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on the Splunk platform:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

```
apt update
apt-get install python3.10
apt install python3-pip
apt install cron
apt install rsyslog
```

2. Install TOA.

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install thunder_observability_agent
```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending on the type of Thunder instance:

Single instance

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Volleyball_TH",
             "password": "Thunder@XYZ@3201",
             "resource_id": "i-1234567890lmnopq0",
             "active_partitions": "shared"
        }]
}
```

Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Volleyball_TH",
          "password": "Thunder@XYZ@3201",
          "resource_id": i-1234567890lmnopq0,
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Volleyball_TH2",
          "password": "Thunder@XYZ@3202",
          "resource_id": "i-1234567890rstuvw0",
          "active_partitions": "shared"
        }]
}
```

b. Update the Splunk credentials in the `/root/.splunk/credentials` file.

```
token_log=2acdaae2a-0497-4a6c-97b7-b155e79aa88
token_metric=f944d49-37f4-4bba-a2f6-df0cdbe86fcbd
```

c. Update Splunk configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "splunk_provider": 1,
  "splunk_metric": 1,
  "splunk_cpu": 1,
  "splunk_memory": 1,
  "splunk_disk": 1,
  "splunk_throughput": 1,
  "splunk_interfaces": 1,
  "splunk_cps": 1,
  "splunk_tps": 1,
  "splunk_server_down_count": 1,
  "splunk_server_down_percentage": 1,
  "splunk_ssl_cert": 1,
  "splunk_server_error": 1,
  "splunk_sessions": 1,
  "splunk_packet_rate": 1,
  "splunk_packet_drop": 1,
  "splunk_log": 1,
  "splunk_host": "127.0.0.0:8088"
```

```
    }
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

Google Console Platform

JKQ Corp. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their Google Cloud Platform (GCP). The instances are configured as an ADC load balancer for their gaming applications named [Baseball]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using the GCP Logs Explorer and GCP Metrics Explorer. Additionally, the client also wants to get an email alert in case of error logs to take the appropriate action.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Baseball_TH
Password	Thunder@JKQ@2828
Resource_Name	North_Virginia_Online_Baseball_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Baseball_TH2
Password	Thunder@JKQ@2829
Resource_Name	North_Virginia_Online_Baseball_

Parameter	Description
	TH2
resource_id	i-1234567890rstuvw0
GCP Monitoring details	
gcp_project_id	jkq-public-396315
gcp_service_key_path	C:/Users/Desktop/keyFolder/jkq-public-396315-db3b0f.json
gcp_log_name	thunder

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on the GCP platform:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

   ```
   apt update
   apt-get install python3.10
   apt install python3-pip
   apt install cron
   apt install rsyslog
   ```

2. Install TOA.

   ```
   pip install virtualenv
   virtualenv venv
   source venv/bin/activate
   pip install thunder_observability_agent
   ```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending on the type of Thunder instance:

      Single instance

      ```
      {
            "autoscale" : 0,
            "provider" : "XXXX",
      ```

```
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Baseball_TH",
             "password": "Thunder@JKQ@2828",
             "resource_id": "i-1234567890lmnopq0",
             "active_partitions": "shared"
        }]
}
```

Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Baseball_TH",
          "password": "Thunder@JKQ@2828",
          "resource_id": i-1234567890lmnopq0,
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Baseball_TH2",
          "password": "Thunder@JKQ@2829",
          "resource_id": "i-1234567890rstuvw0",
          "active_partitions": "shared"
        }]
}
```

b. Update the GCP credentials in the `/root/.gcp/credentials` file.

```
gcp_project_id = jkq-public-396315
gcp_service_key_path = C:/Users/Desktop/keyFolder/jkq-public-
396315-db3b0f.json
```

c. Update GCP configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "gcp_provider": 1,
  "gcp_metric": 1,
  "gcp_cpu": 1,
  "gcp_memory": 1,
  "gcp_disk": 1,
  "gcp_throughput": 1,
  "gcp_interfaces": 1,
  "gcp_cps": 1,
  "gcp_tps": 1,
  "gcp_server_down_count": 1,
  "gcp_server_down_percentage": 1,
  "gcp_ssl_cert": 1,
  "gcp_server_error": 1,
  "gcp_sessions": 1,
  "gcp_packet_rate": 1,
  "gcp_packet_drop": 1,
  "gcp_log": 1
}
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

Oracle Cloud Infrastructure

TUV Corp. is a regular A10 client. The company has purchased multiple instances of Thunder and deployed it on their OCI platform. The instances are configured as an ADC load balancer for their gaming applications named [Volleyball]. The company is receiving timeout/failover complaints from their online customers especially when there is a high traffic load caused by an event, festival, or holiday. The client wants a standard way to monitor using the OCI Logs and OCI Metrics Explorer. Additionally, the client also wants to get an email alert in case of error logs to take the appropriate action.

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51

Parameter	Description
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Volleyball_TH
Password	Thunder@TUV@2828
Resource_Name	North_Virginia_Online_Volleyball_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Volleyball_TH2
Password	Thunder@TUV@2829
Resource_Name	North_Virginia_Online_Volleyball_TH2
resource_id	i-1234567890rstuvw0
OCI Monitoring details	
oci_api_key_path	C:/Users/Desktop/keyFolder/tuvconfig
oci_compartment_id	ocid1.compartment.oc1..amlkjytrnpczhiafkgum6yjjhltv6frnn3wb6y3442fr5tc3j4kljhgfsq
oci_log_id	ocid1.log.oc1.phx.amlkjytrnpczhiafkgfrfvuboum6yjjhltv6frnn3wb6y3442fr5tc3j4sq

Solution

A10 Support team will propose to install **Thunder Observability Agent (TOA)** for collecting and publishing logs on the OCI platform:

1. Install Python if the recommended version is not already installed on the shared Linux instance IP 10.22.32.51.

```
apt update
apt-get install python3.10
apt install python3-pip
apt install cron
apt install rsyslog
```

2. Install TOA.

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install thunder_observability_agent
```

3. Configure TOA.

   a. Configure Thunder details in the `/root/.thunder/credentials` file depending on the type of Thunder instance:

Single instance

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "10.22.32.01",
             "username": "Online_Volleyball_TH",
             "password": "Thunder@TUV@2828",
             "resource_id": "i-1234567890lmnopq0",
             "active_partitions": "shared"
        }]
}
```

Multiple instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
          "ip": "10.22.32.01",
          "username": "Online_Volleyball_TH",
          "password": "Thunder@TUV@2828",
          "resource_id": i-1234567890lmnopq0,
          "active_partitions": "shared"
        },
        {
          "ip": "10.22.32.02",
          "username": "Online_Volleyball_TH2",
          "password": "Thunder@TUV@2829",
          "resource_id": "i-1234567890rstuvw0",
          "active_partitions": "shared"
        }]
}
```

b. Update the OCI credentials in the `/root/.oci/credentials` file.

```
oci_api_key_path = C:/Users/Desktop/keyFolder/tuvconfig
```

c. Update OCI configuration properties in the `/usr/toaenv/thunder-observability-agent/config.json` file.

```
{
  "oci_provider": 1,
  "oci_metric": 1,
  "oci_compartment_id":
"ocid1.compartment.oc1..amlkjytrnpczhiafkgum6yjjhltv6frnn3wb6y3442f
r5tc3j4kljhgfsq" ,
  "oci_cpu": 1,
  "oci_memory": 1,
  "oci_disk": 1,
  "oci_throughput": 1,
  "oci_interfaces": 1,
  "oci_cps": 1,
  "oci_tps": 1,
  "oci_server_down_count": 1,
  "oci_server_down_percentage": 1,
  "oci_ssl_cert": 1,
  "oci_server_error": 1,
  "oci_sessions": 1,
  "oci_packet_rate": 1,
  "oci_packet_drop": 1,
  "oci_log": 1,
  "oci_log_id":
"ocid1.log.oc1.phx.amlkjytrnpczhiafkgfrfvuboum6yjjhltv6frnn3wb6y344
2fr5tc3j4sq"
}
```

4. Check logs at `/var/log/thunder-observability-agent/agent.log`.

For more examples, see GitHub.

What's New

3.0.0

In this release, the TOA is enhanced to support the following enterprise solutions for data collection and analytics:

- Google Cloud Platform (GCP)
- Oracle Cloud Infrastructure (OCI)

2.0.0

In this release, the TOA is enhanced to support the following enterprise solutions for data collection and analytics:

- Elasticsearch - Kibana
- Prometheus (Pushgateway) - Grafana
- Splunk - Splunk Analytics and Splunk Dashboard

1.0.0

This release has the following enhancements for Thunder® Application Delivery Controller (ADC):

- TOA supports Linux, CentOS, and Ubuntu platforms as a Python Plugin installation package and Docker containerization.
- TOA supports AWS, Azure, and VMware cloud providers.
- Single, multiple, and auto scale Thunder instances can be configured for TOA.
- TOA provides multitasking capabilities to collect and process data from multiple Thunder instances and its partitions simultaneously. By default, it collects data from shared partition.
- TOA supports Shared and L3V partitions. The maximum number of partitions supported per Thunder is 20.
- TOA collects, processes and publishes 14 Thunder metrics. The default data collection frequency is 1 minute. The metrics can be published on the same platform where the Thunder instance is deployed. For more information on Thunder metrics, see Supported Thunder Metrics.
- TOA collects, processes, and publishes Thunder Syslogs. The default data collection frequency is 1 minute. The logs can be published on the same platform where the Thunder instance is deployed or it can also be published to any AWS, Azure, or VMware platforms. For more information on Thunder logs, see Supported Thunder Logs.

Appendix

Get Resource ID

To get the resource ID for single or multiple Thunder instance/s, perform the following steps depending on your cloud provider:

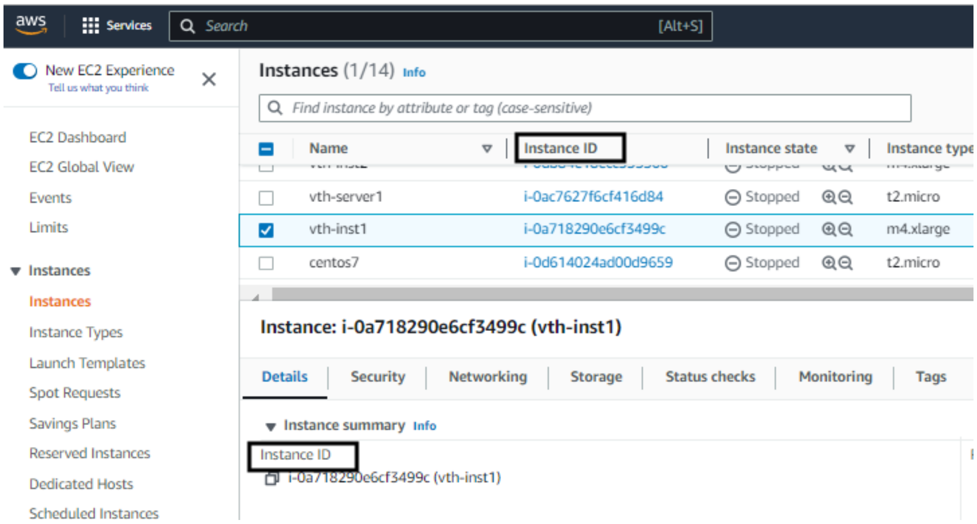
AWS

1. Go to **AWS Management Console** > **EC2** > **Instances** and select your Thunder

instance.

2. From the **Details** tab, get the **Instance ID**.

Figure 84 : Thunder instance Resource ID



	Name	Instance ID	Instance state	Instance type
□	vth-server1	i-0ac7627f6cf416d84	⊖ Stopped	t2.micro
☑	vth-inst1	i-0a718290e6cf3499c	⊖ Stopped	m4.xlarge
□	centos7	i-0d614024ad00d9659	⊖ Stopped	t2.micro

Instance: i-0a718290e6cf3499c (vth-inst1)

Instance ID
i-0a718290e6cf3499c (vth-inst1)

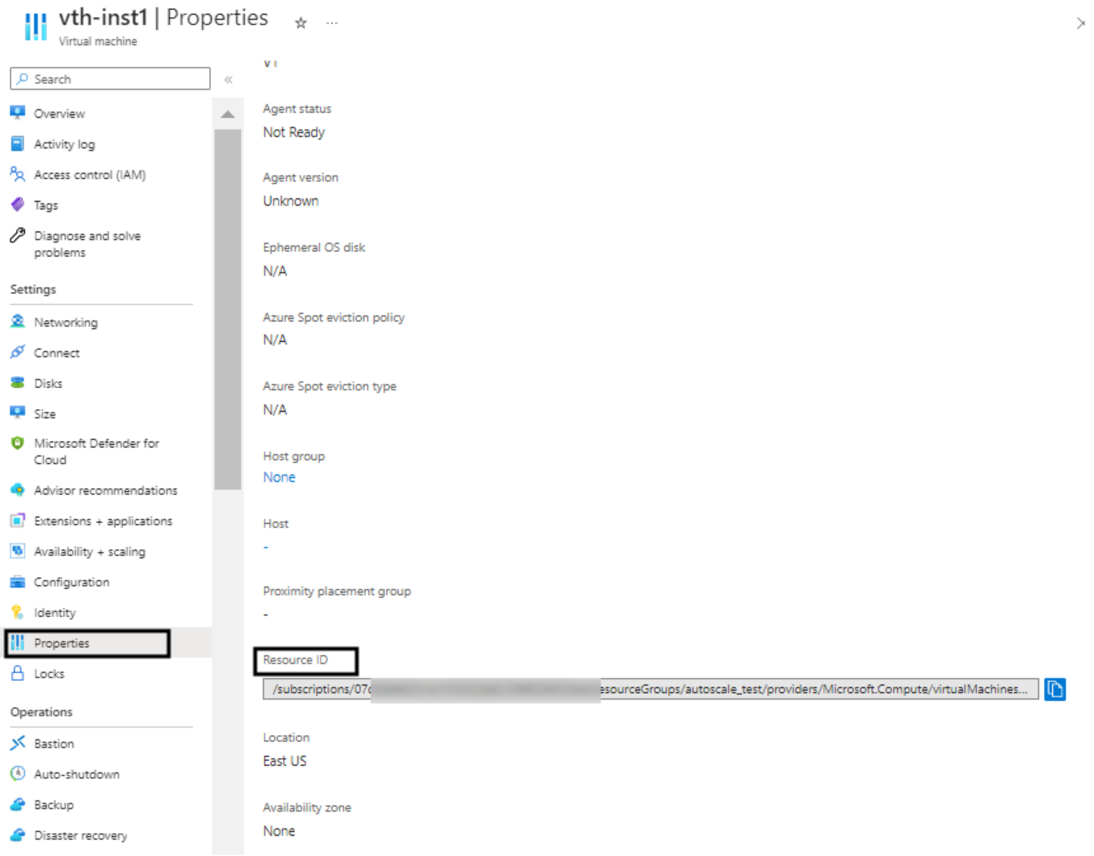
Azure

1. Go to **Azure Portal** > **Azure services** > **Virtual machine** and select your Thunder instance.

2. From the left panel, click **Setting** > **Properties**.

3. Get the **Resource ID** from the right panel.

Figure 85 : Thunder instance Resource ID



Agent status
Not Ready

Agent version
Unknown

Ephemeral OS disk
N/A

Azure Spot eviction policy
N/A

Azure Spot eviction type
N/A

Host group
None

Host
-

Proximity placement group
-

Resource ID
/subscriptions/07[redacted]esourceGroups/autoscale_test/providers/Microsoft.Compute/virtualMachines...

Location
East US

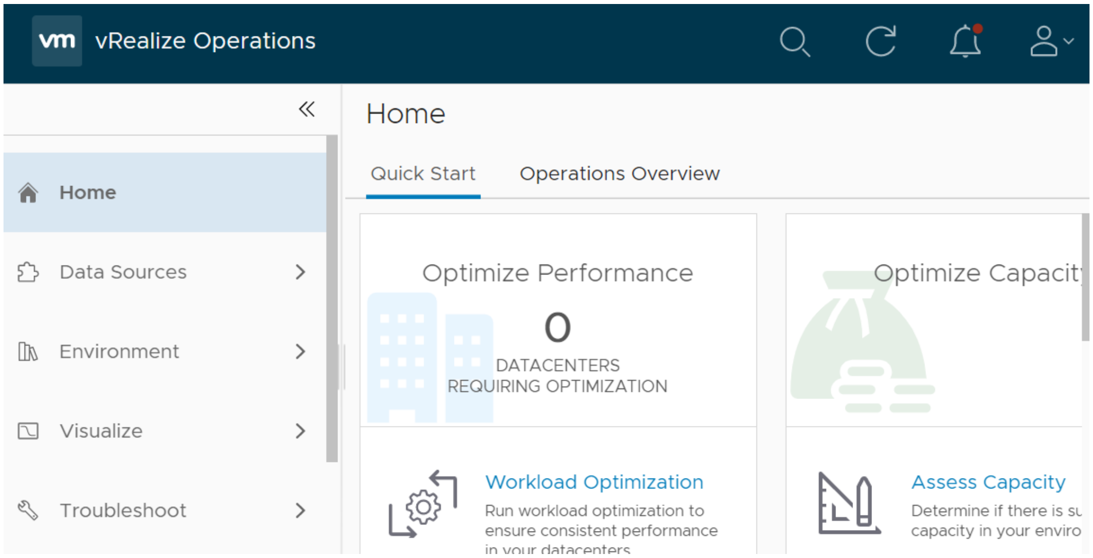
Availability zone
None

VMware

1. Log in to the **vRealize Operations Web UI** with your admin credentials to get the Thunder Resource ID once your vROps virtual machine is powered on.

   The vRealize Operations Home page is displayed.

Figure 86 : vRealize Operations - Home page

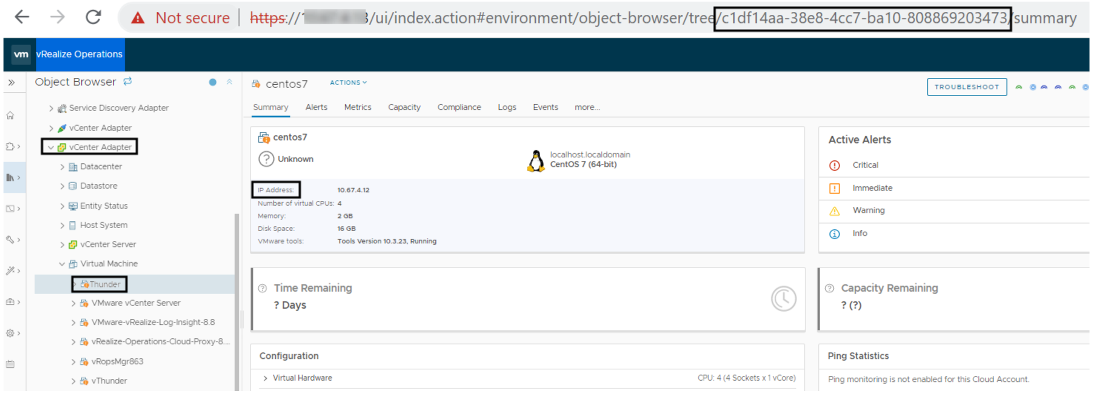


Run workload optimization to ensure consistent performance in your datacenters

Determine if there is su capacity in your enviro

2. Go to **Home** > **Environment** > **Object Browser** > **All Objects** > **vCenter Adapter** > **Virtual Machine** and click **Thunder**.

Figure 87 : vRealize Operations - Virtual machine window



https://[redacted]/ui/index.action#environment/object-browser/tree/c1df14aa-38e8-4cc7-ba10-808869203473/summary

centos7
Unknown
localhost.localdomain
CentOS 7 (64-bit)

IP Address: 10.67.4.12
Number of virtual CPUs: 4
Memory: 2 GB
Disk Space: 16 GB
VMware tools: Tools Version 10.3.23, Running

CPU: 4 (4 Sockets x 1 vCore)

Ping monitoring is not enabled for this Cloud Account.

3. Get the resource ID from the URL.

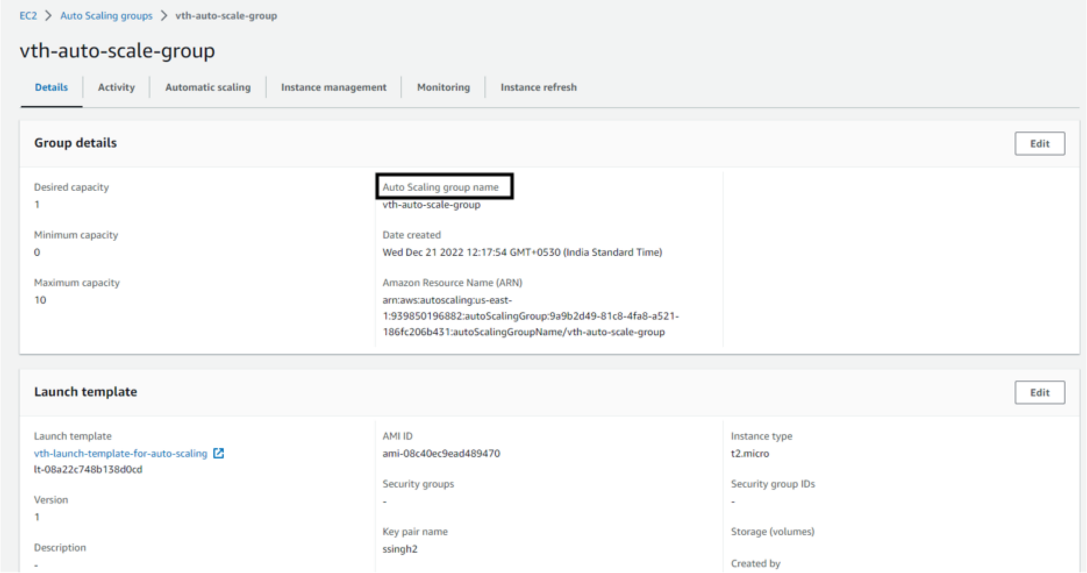
NOTE: This resource ID is necessary only when directing VM metrics data exclusively to vRealize Operations. However, for sending Thunder metrics data to platforms other than VMware, any custom name can be assigned as the resource ID, for example, 'vm-123'. Additionally, for sending Thunder Syslogs to VMware and other platforms, this resource ID is optional, and any custom name, such as 'vm-123', can be assigned as the resource ID.

To get the resource ID for Thunder instance in auto scaling group or in VMSS, perform the following steps depending on your cloud provider:

AWS

1. Go to **AWS Management Console** > **EC2** > **Auto Scaling Groups** and select your Thunder auto scale group instance.
2. From the **Details** tab, get the **Auto Scaling group name**.

Figure 88 : Thunder Auto Scaling instance Resource ID



Group details

Desired capacity
1

Minimum capacity
0

Maximum capacity
10

Auto Scaling group name
vth-auto-scale-group

Date created
Wed Dec 21 2022 12:17:54 GMT+0530 (India Standard Time)

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:939850196882:autoScalingGroup:9a9b2d49-81c8-4fa8-a521-186fc206b431:autoScalingGroupName/vth-auto-scale-group

Launch template

Launch template
vth-launch-template-for-auto-scaling
lt-08a22c748b138d0cd

Version
1

Description
-

AMI ID
ami-08c40ec9ead489470

Security groups
-

Key pair name
ssingh2

Instance type
t2.micro

Security group IDs
-

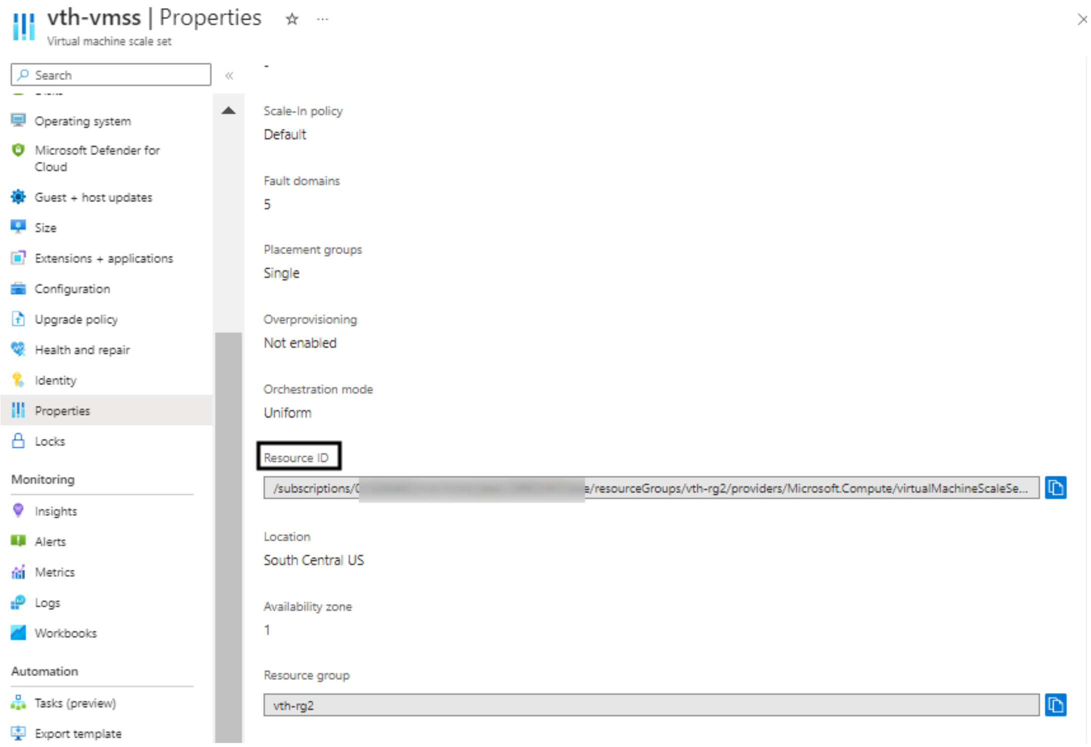
Storage (volumes)

Created by

Azure

1. Go to **Azure Portal** > **Azure services** > **Virtual machine scale set** and select your Thunder VMSS instance.
2. From the left panel, click **Setting** > **Properties**.
3. Get the **Resource Group name** from the right panel.

Figure 89 : Thunder VMSS instance Resource ID



Scale-In policy
Default

Fault domains
5

Placement groups
Single

Overprovisioning
Not enabled

Orchestration mode
Uniform

Resource ID
/subscriptions/[redacted]/resourceGroups/vth-rg2/providers/Microsoft.Compute/virtualMachineScaleSe...

Location
South Central US

Availability zone
1

Resource group
vth-rg2

Install Python, Crontab, and Syslog

Depending on your operation system, install Python (3.6 or higher), Crontab, and Syslog:

CentOS

To install latest Python from OS repository, perform the following steps:

```
yum install -y python3
```

To install Crontab and Syslog, perform the following steps:

```
yum install cronie
yum install rsyslog
```

Linux/Ubuntu

To install Python, perform the following steps:

```
apt update
apt-get install python3.10
apt install python3-pip
```

To install Crontab and Syslog, perform the following steps:

```
apt install cron
apt install rsyslog
```

Uninstall TOA

To uninstall TOA, perform the following steps:

1. Run the following commands to uninstall TOA:

```
cd /usr
source toaenv/bin/activate
pip uninstall thunder-observability-agent
```

2. Run the following commands to remove the cloud-specific configuration files:

```
cd /root
rm -rf .aws .azure .vmware .thunder
```

3. Run the following commands to remove the TOA configuration files:

```
cd /usr
rm -rf toaenv
```

4. Run the following command to remove the crontab configuration:

```
crontab -e
```

5. Remove the following entry from the crontab file:

```
*/1 * * * * /usr/toaenv/bin/python3 /usr/toaenv/lib/python3.10/site-
packages/thunder-observability-agent/toa.py
```

6. Run the following commands to remove TOA:

```
cd /var/log/
rm -rf thunder-observability-agent
```

Import vROps Template

The vRealize Operations Manager (vROps) creates a dashboard and a notification by importing a JSON files. It also creates alert definition by importing an XML file.

The following topics are covered:

- Import a Dashboard
- Import an Alert Definition
- Import a Notification

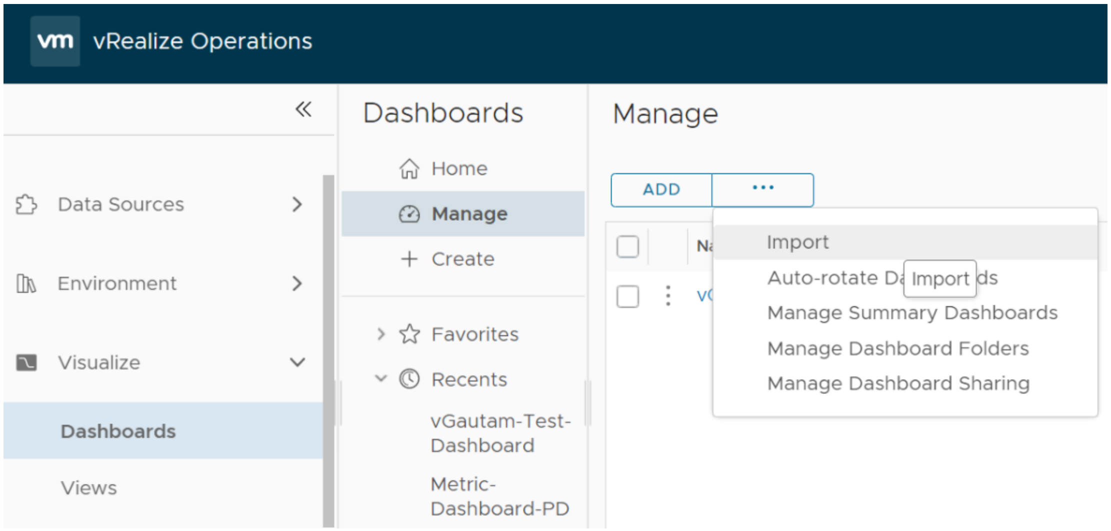
Import a Dashboard

To import a dashboard using the JSON file, perform the following steps:

1. Download and open the dashboard-template JSON file.
2. Edit the following parameter values in the JSON file:
   - `id`
   - `name`
3. Save the changes in the JSON file.
4. From the **vRealize Operations Web UI**, go to **Home** > **Visualize** > **Dashboards** and click **Manage**.

   The **Manage** window is displayed.

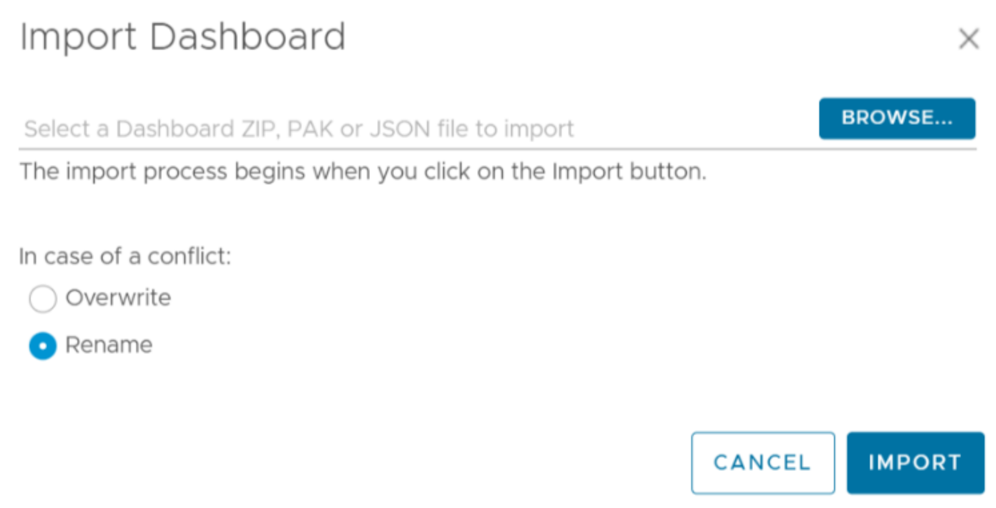
Figure 90 : Manage window



5. Click **...** > **Import** in the **Manage** panel.

   The **Import Dashboard** window is displayed.

   Figure 91 : Import Dashboard window



Import Dashboard ×

Select a Dashboard ZIP, PAK or JSON file to import BROWSE...

The import process begins when you click on the Import button.

In case of a conflict:

○ Overwrite

● Rename

CANCEL IMPORT

6. Browse and select the **dashboard-template.json** file.
7. Click **Import**.

   The new dashboard is imported and listed in the **Dashboards** window.

Import an Alert Definition

To import an alert definition using the XML file, perform the following steps:

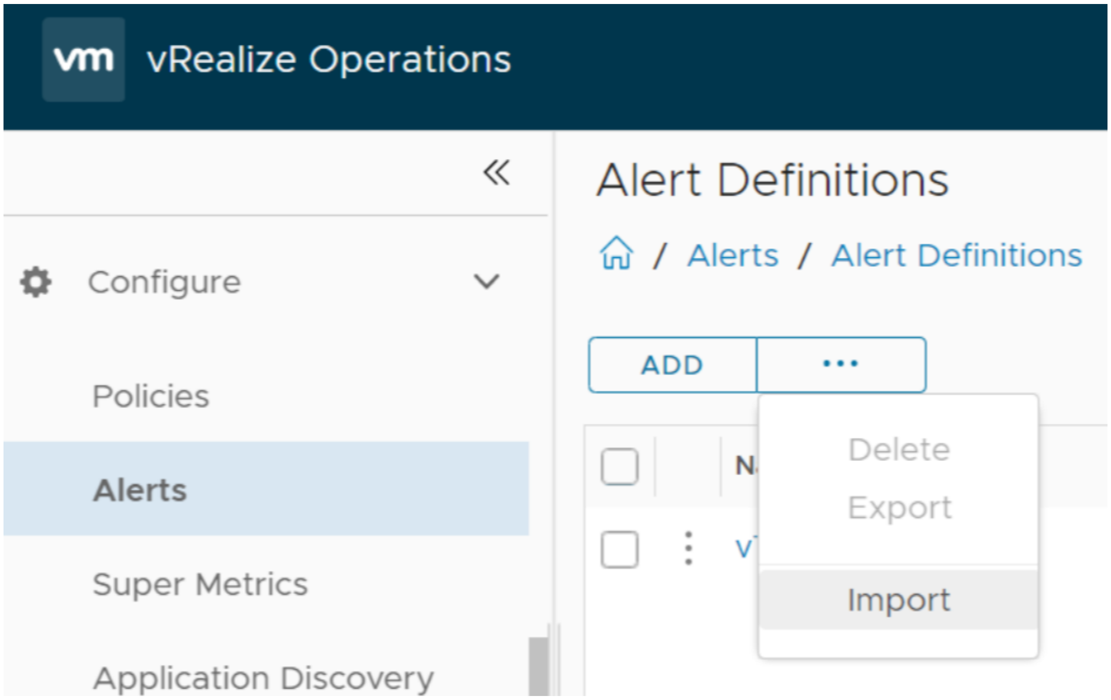
1. Download and open the alert-template XML file.
2. Enter the following parameter values in the XML file as appropriate:
   - `id`
   - `name`

NOTE: The `id` and `name` must have unique values.

3. Save the changes in the XML file.
4. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** and click **Alert Definitions**.

   The **Alert Definitions** window is displayed.

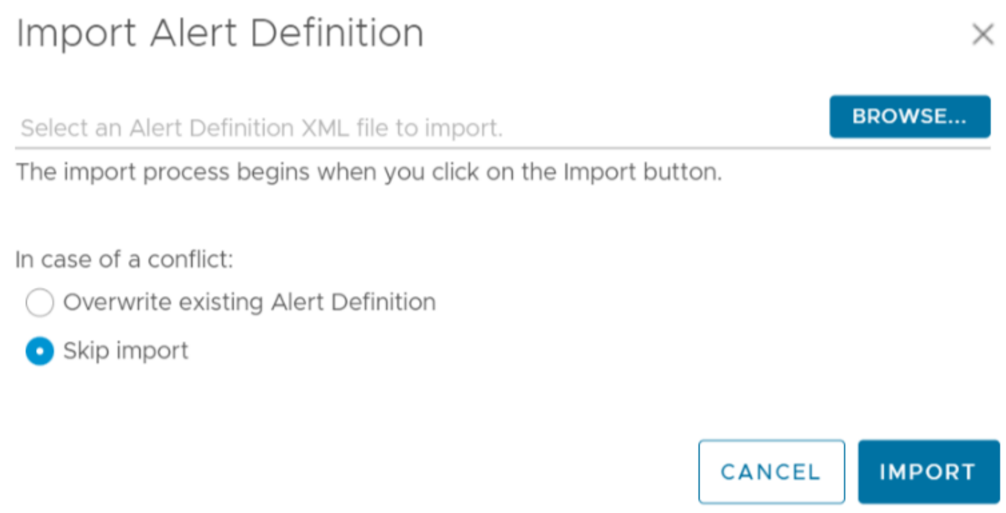
Figure 92 : Alert Definitions window



5. Click **...** > **Import** in the **Alert Definition** window.

   The **Import Alert Definition** window is displayed.

Figure 93 : Import Alert Definition window



Import Alert Definition

Select an Alert Definition XML file to import.

The import process begins when you click on the Import button.

In case of a conflict:

○ Overwrite existing Alert Definition

● Skip import

6. Browse and select the **alert-template.json**.
7. Click **Import**.

   The new alert definition is imported and listed in the **Alert Definitions** window.

Import a Notification

To import a notification using the JSON file, perform the following steps:

1. Download and open the notification-template JSON file.
2. Update the alert definition id in the following parameter:

```
 {
          "ConditionType":"ALERT_DEFINITION_ID",
          "NotificationRuleAlertDefinitionCondition":{
               "AlertDefinitionIds":[
                  {
                   "AlertDefinitionID":"AlertDefinition-<alert-
definition-id>"
                  }
                 ]
               }
        }
```

NOTE: The `AlertDefinitionID` must have the same value as provided in the **alert-template.json**.

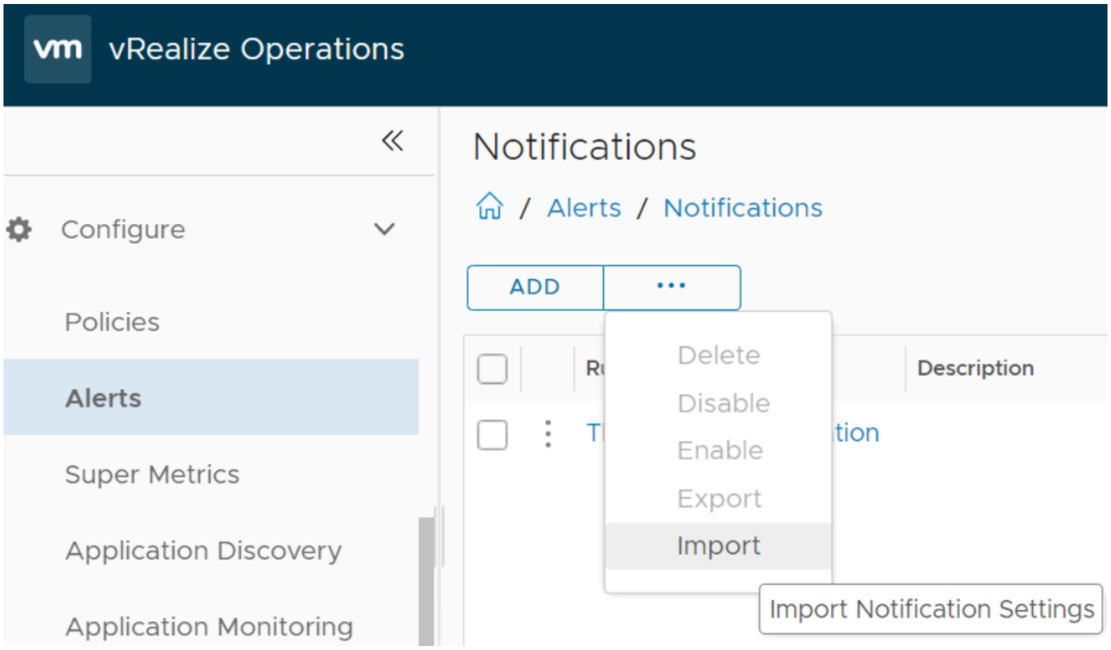
3. Update the sender and recipient email address values in the following parameter:

```
"PluginNotificationProperty":[
                  {
                     "PropertyName":"emailaddr",
                     "PropertyValue":"user1@example.com"
                  },
                  {
                     "PropertyName":"ccRecipients",
                     "PropertyValue":"usergroup@example.com"
                  }
               ],
```

4. Save the changes in the JSON file.
5. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** and click **Notifications**.

   The **Notifications** window is displayed.

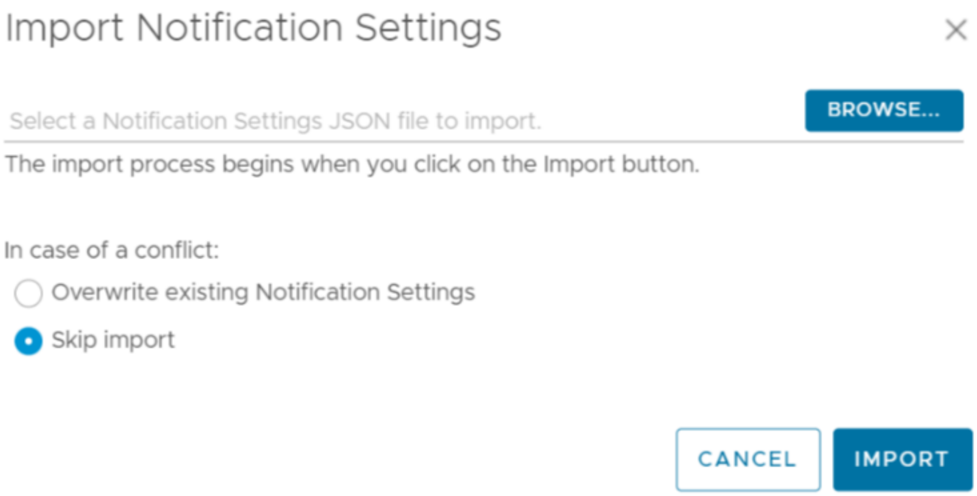
Figure 94 : Notifications window



6. Click **...** > **Import** in the **Notifications** panel.

   The **Import Notification Settings** window is displayed.

   Figure 95 : Import Notification Settings window

   Import Notification Settings

   Select a Notification Settings JSON file to import.

   The import process begins when you click on the Import button.

   In case of a conflict:

   ○ Overwrite existing Notification Settings

   ● Skip import

7. Browse and select the **notification-template.json** file.
8. Click **Import**.

   The new notification is imported and listed in the **Notifications** window.

Installing vROps and vRLI

vROps

To install vROps on an ESXi host, see vROps Installation.

vRLI

To install vRLI on an ESXi host, see vRLI Installation.

Base64 Conversion Examples

Base64 is an encoding technique used to convert binary data into an ASCII text format. The process of converting a JSON file to Base64 is particularly relevant for cloud platforms like GCP and OCI, primarily due to the presence of private keys structured in the JSON format.

Google Cloud Platform

In this example, the private key contained in **gcpServiceKeyFile.json** is converted to Base64 and then placed in the YAML file.

- Contents of **gcpServiceKeyFile.json**:

```
{
"type": "service_account",
"project_id": "xxxx",
"private_key_id": "xxxx",
"private_key": "-----BEGIN PRIVATE KEY-----\xxxxn-----END PRIVATE KEY-----\n",
"client_email": "xxxx",
"client_id": "xxxx",
"auth_uri": "xxxx",
"token_uri": "xxxx",
"auth_provider_x509_cert_url": "xxxx",
"client_x509_cert_url": "xxxx",
"universe_domain": "xxxx"
}
```

- Encoded output after converting JSON to Base64 :

```
ewogICJ0eXBlIjogInNlcnZpY2VfYWNjb3VudCIsCiAgInByb2plY3RfaWQiOiAiYTEwbmV0d29ya3Mt
```

NOTE: The curly brackets must also be included during the conversion.

- The encoded Base64 string is placed in the YAML file in the following manner:

```
---
apiVersion: v1
kind: Secret
metadata:
name: gcp-service-key-file-secret
namespace: thunder-observability-agent
type: Opaque
```

```
data:
gcpServiceKeyFile.json: |
ewogICJ0eXBlIjogInNlcnZpY2VfYWNjb3VudCIsCiAgInByb2plY3RfaWQiOiAiYTEwbmV
0d29ya3Mt
```

The Base64 string must follow the pipe character. Ensure that the entire encoded key is indented to align under the **gcpServiceKeyFile.json** field, maintaining the YAML structure.

Oracle Cloud Infrastructure

In this example, the private key contained in **ociPrivateKey.pem** is converted to Base64 and then placed in the YAML file.

- Contents of **ociPrivateKey.pem**:

```
-----BEGIN PRIVATE KEY-----
xxxxxxxx
-----END PRIVATE KEY-----
```

- Encoded output after converting JSON to Base64:

```
LS0tLS1CRUdJTiBQUklWQVRFIEtFWS0tLS0tCk1JSUV2Z0lCQURBTkJna3Foa2lHOXcwQkF
RRUZBQVNDQktnd2dnU
```

- The encoded Base64 string is placed in the YAML file in the following manner:

```
---
apiVersion: v1
kind: Secret
metadata:
name: oci-private-key-file-secret
namespace: thunder-observability-agent
type: Opaque
data:
ociPrivateKey.pem: |
LS0tLS1CRUdJTiBQUklWQVRFIEtFWS0tLS0tCk1JSUV2Z0lCQURBTkJna3Foa2lHOXcwQkF
RRUZBQVNDQktnd2dnU
---
```

The Base64 string must follow the pipe character. Ensure that the entire encoded key is indented to align under the **ociPrivateKey.pem** field, maintaining the YAML structure.

Creating Widgets in OCI

You can add widgets to your dashboard within the Oracle Cloud Infrastructure (OCI) Logging Analytics service to visualize and analyze your data effectively.

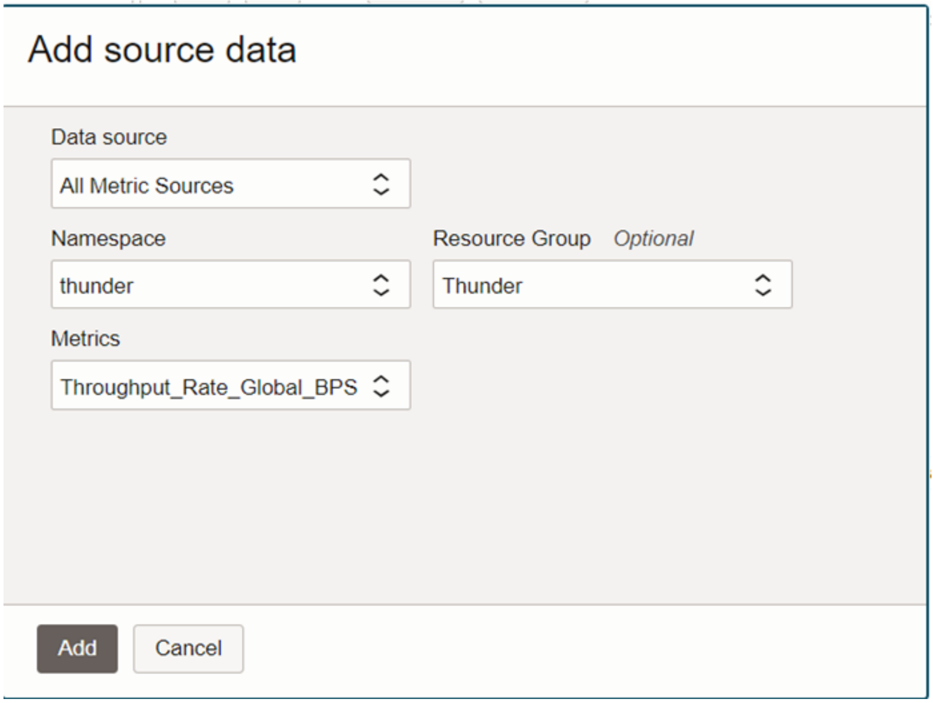
While creating a dashboard, the **Widget** tab on the dashboard creation page provides the following options to create a widget:

Create Widget

This option allows you to add a variety of pre-configured widgets to your dashboard. To create a widget using this method, perform the following steps:

1. On the **metric widget creation** page, under **Data** panel, click **+**.
2. The **Add Source Data** dialog box is displayed as shown in Figure 96.

Figure 96 : Create Widget - Add Source Data



Add source data

Data source: All Metric Sources

Namespace: thunder

Resource Group *Optional*: Thunder

Metrics: Throughput_Rate_Global_BPS

3. Enter data source information to generate the metrics:

- **Data Source** - Select **All Metric Sources**.
- **Namespace** - Select the namespace you have access to; in this case `thunder`.

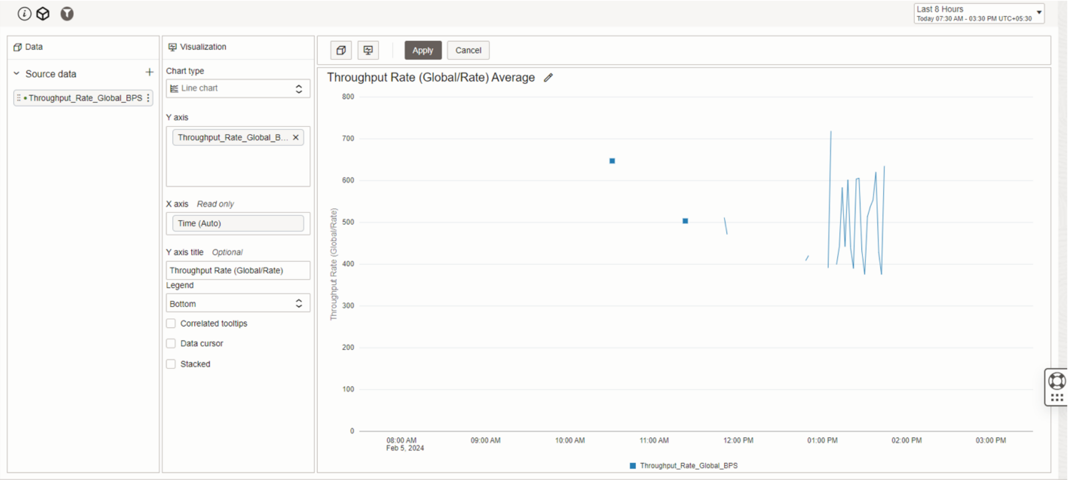
- **Resource Group** - Select the resource group; in this case `Thunder`.
- **Metrics** - Based on the previous three selections, this menu gets refreshed with the names of all the metrics available. Select a metric of your choice.

4. Click **Add**.

   All the selected metrics will be listed under **Source Data**.

5. Drag and drop the metrics that you want to visualize from **Source Data** section to **Y Axis** section under **Visualization** panel as shown in Figure 97.

   Figure 97 : Create Widget



Data
Source data
Throughput_Rate_Global_BPS

Visualization
Chart type: Line chart
Y axis: Throughput_Rate_Global_B...
X axis *Read only*: Time (Auto)
Y axis title *Optional*: Throughput Rate (Global/Rate)
Legend: Bottom
□ Correlated tooltips
□ Data cursor
□ Stacked

Last 8 Hours
Today 07:30 AM - 03:30 PM UTC+05:30

   You can see the chart where the selected data is plotted along Y axis. Additionally, you can add more metrics to the Y Axis section and visualize multiple metrics together. You can customize the visualization by specifying or modifying the visualization options. Some of the common options are:

   - **Time Range** - Select the time range from the time selector.
   - **Chart Type** - You can select the chart type as **Area Chart**, **Line Chart** or **Bar Chart**.
   - **Y Axis Title** - Specify a title for the data projected on the Y Axis.
   - **Stacked** - In case of multiple metrics, you can use this option to stack charts for better viewing.

6. Specify the name for the widget in the field provided above the chart.

7. Click **Apply** to save the widget.

   The widget will be added to the dashboard as well.

For more information, see Creating Widgets.

Create Query-Based Widget

The option allows you to add widgets based on queries executed on your data. To create a query-based widget, perform the following steps:

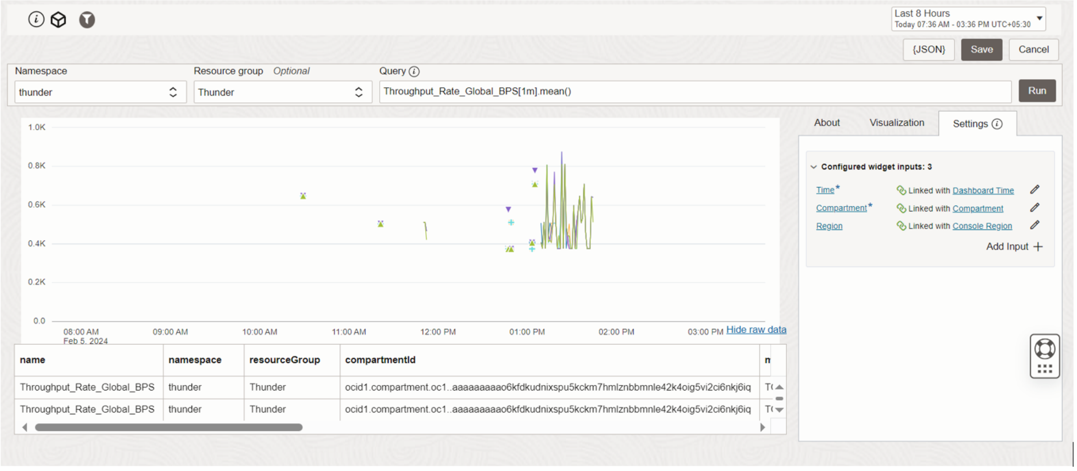
1. After clicking **Create query-based widget**, the query-based widget builder is displayed.
2. Specify the following metric details:
   - **Namespace** - Select the applicable namespace; **`thunder`** in this case.
   - **Resource group** - Select a resource group; **`Thunder`** in this case.
   - **Query** - Enter a query, in MQL syntax, to retrieve the metric information you want to display in the widget. For example, **`CPU_Usage_Percentage_Data [auto].grouping().mean()`**
3. Click **Run**.

   The query is executed and the metric data is displayed in a tabular format.
4. In the **About** tab, enter a name for the widget, select a compartment where you want the widget to reside, and add a description.
5. In the **Visualization** tab, select a chart type and customize the visualization. You can customize the visualization by specifying or modifying the visualization options. Some of the common options are:
   - **X axis** - Select the data attribute to be projected on the X axis.
   - **Y axis** - Select the data attribute to be projected on the Y axis.
   - **Series** - Select the data attribute to be plotted in a separate series in the chart.
   - **Color by** - Select the data attribute for which you want to assign different colors.
   - **X Axis Title** - Specify a title for the data projected on the X Axis.

- **Y Axis Title** - Specify a title for the data projected on the Y Axis.
- **Stacked** - In case of multiple metrics, you can use this option to stack charts for better viewing.

6. In **Settings** tab, you can review and edit the widget inputs, if needed.
7. Click **Save** to save the widget.

   The widget will be added automatically to the dashboard as well.



Namespace: thunder

Resource group *Optional*: Thunder

Query: Throughput_Rate_Global_BPS[1m].mean()

Configured widget inputs: 3

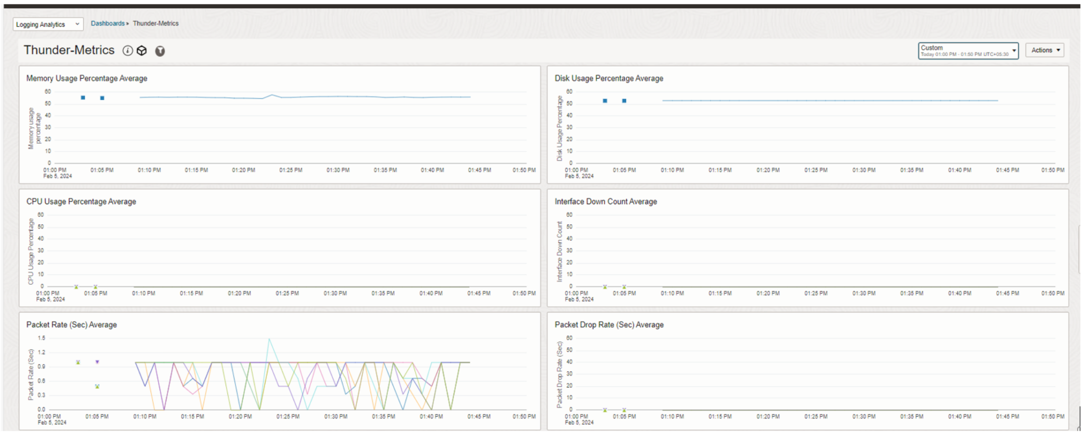
- Time* — Linked with Dashboard Time
- Compartment* — Linked with Compartment
- Region — Linked with Console Region

Add Input +

name	namespace	resourceGroup	compartmentId	n
Throughput_Rate_Global_BPS	thunder	Thunder	ocid1.compartment.oc1..aaaaaaaaao6kfdkudnixspu5kckm7hmlznbbmnle42k4oig5vi2ci6nkj6iq	T(
Throughput_Rate_Global_BPS	thunder	Thunder	ocid1.compartment.oc1..aaaaaaaaao6kfdkudnixspu5kckm7hmlznbbmnle42k4oig5vi2ci6nkj6iq	T(

8. Similarly, you can add other metrics to the dashboard as shown in Figure 98.

   Figure 98 : OCI Dashboard


Logging Analytics
Dashboards » Thunder-Metrics
Thunder-Metrics
Custom
Today 01:00 PM - 01:50 PM UTC+05:30
Actions
Memory Usage Percentage Average
Disk Usage Percentage Average
CPU Usage Percentage Average
Interface Down Count Average
Packet Rate (Sec) Average
Packet Drop Rate (Sec) Average
01:00 PM
Feb 5, 2024
01:05 PM
01:10 PM
01:15 PM
01:20 PM
01:25 PM
01:30 PM
01:35 PM
01:40 PM
01:45 PM
01:50 PM

For more information of creating query-based widgets, see Creating Query-based Widgets.

Create Policies to Publish Data in OCI

To publish metrics and logs in OCI, you need to create and manage certain policies that define the necessary permissions. These policies specify which groups or users have access to perform certain actions on resources within specific compartments.

To create a policy, perform the following steps:

1. Log in to the OCI console and navigate to **Identity & Security** > **Policies**.
2. On the Policies page, click **Create Policy**.
3. In the **Create Policy** section, enter a policy name, description, and specify the compartment where you want to create the policy.
4. Under **Policy Builder**, click the **Show manual editor**.
5. Enter the policy rules based on the data that needs to be published:
   - To publish metrics, enter Policies for Metrics
   - To publish logs, enter Policies for Logs
6. Click **Create**.

Policies for Metrics

To publish metrics you need to grant permission to the following policies in OCI:

- **`Allow group`** *`<group_name>`* **`to read metrics in compartment`** *`<compartment_name>`*

  This policy allows the specified group to read metrics within the specified compartment.

- **`Allow group`** *`<group_name>`* **`to manage alarms in compartment`** *`<compartment_name>`*

  This policy grants the specified group permission to manage alarms within the specified compartment.

- **`Allow group`** *`<group_name>`* **`to manage ons-topics in compartment`** *`<compartment_name>`*

This policy provides the specified group with permissions to manage Oracle Notification Service (ONS) topics within the specified compartment.

- **Allow group** *<group_name>* **to use streams in compartment** *<compartment_name>*

  This policy enables the specified group to use streams within the specified compartment. Streams are used for real-time data ingestion, processing, and analysis.

Policies for Logs

To publish logs you need to grant permission to the following policies in OCI:

- **Allow group** *<group_name>* **to use log-groups in compartment** *<compartment_name>*

  This policy allows the specified group to access and view log-groups within the specified compartment.

- **Allow group** *<group_name>* **to manage log-groups in compartment** *<compartment_name>*

  This policy allows the specified group to create, update, and delete log groups within the specified compartment.

- **Allow group** *<group_name>* **to write logs in compartment** *<compartment_name>*

  This policy permits the specified group to write logs to log groups within the specified compartment.

For more information on policies, see Managing Policies.

Disclaimer

IMPORTANT READ CAREFULLY

To use TOA, the user must license and install the following software. All such software is licensed separately by the owner of such software. A10 Networks has no responsibility for such software, nor does it provide any representation, warranty, or other attestation of it. A description of the licenses for such software is provided below for your convenience only, however, it is up to you to confirm the license terms for such software at the time of installation and comply with them.

If you have any questions about the open-source software needed to use the TOA product, please email support@a10networks.com. In the subject line of your email, please reference: 'Open-Source Software'.

Open-Source Licenses and Copyright Notices for Required Software

The following table lists the open-source software which must be licensed and installed in order to use TOA and the open-source license type for each tool.

Tool	License
Python 3.10+	PSF LICENSE AGREEMENT FOR PYTHON 3.10.11
Requests 2.27.1+	Apache Software License 2.0
Boto3 1.23.10+	Apache 2.0 (amazon.com) https://aws.amazon.com/apache-2-0/
Botocore 1.29.121	Apache Software License 2.0
google-auth 2.22.0	Apache Software License 2.0, Apache 2.0 (google.com)
oci 2.121.1	Apache Software License, Universal Permissive License
certifi 2022.12.7	Mozilla Public License 2.0
charset-normalizer 3.1.0	MIT License
idna 3.4	MIT License
jmespath 1.0.1	MIT License
python-dateutil 2.8.2	Apache Software License 2.0, BSD License
s3transfer 0.6.0	Apache Software License 2.0
Six 1.16.0	MIT License
urllib3 1.26.15	MIT License

License

For TOA License information, see THUNDER OBSERVABILITY AGENT END USER SOFTWARE LICENSE AGREEMENT.

Contact Us