



Installing Thunder Observability Agent 1.0.0

May, 2023

© 2023 A10 Networks, Inc. All rights reserved.

Information in this document is subject to change without notice.

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](#)

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

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	5
Supported Information	6
Supported Technology	6
Supported Thunder Metric	7
Supported Log Thunder	8
Supported ACOS versions	9
Supported Cloud Apps	9
Install TOA	11
Python Plugin Installation	12
Prerequisites	12
Hardware Dependencies	12
Software Dependencies	13
Installation Steps	13
Containerized Installation	16
Prerequisites	16
Installation Steps	16
Configure TOA	19
Global Configuration	19
Main Properties	19
Logging	22
Crontab	23
Cloud-specific Configuration	24
AWS Config	24
AWS Credentials	25
Azure Credentials	26
VMware Credentials	28
Data Collection Configuration	28
Thunder Credentials	28

Data Publish Configuration	31
Config JSON	32
Monitor Dashboard	44
Monitor Metrics	44
Monitor Logs	68
Troubleshoot	74
TOA Logging	74
Examples	77
AWS	77
Azure	81
VMware	86
What's New	90
1.0.0	90
Appendix	91
Get Resource ID	91
Install Python	95
Uninstall TOA	96
Import vROps Template	97
Import a Dashboard	97
Import an Alert Definition	99
Import a Notification	101

Introduction

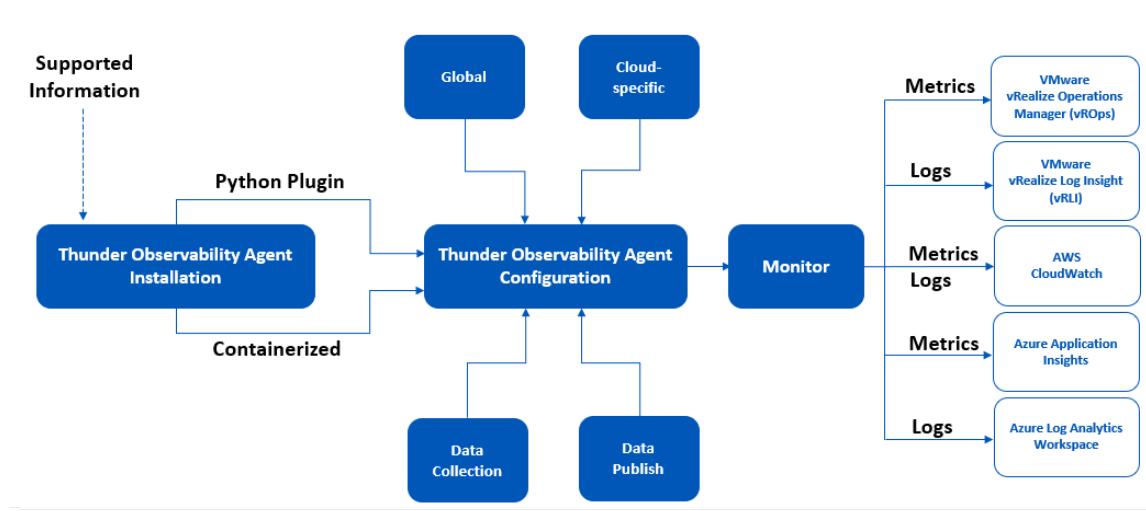
Thunder Observability Agent (TOA) is a lightweight autonomous data processing engine that can be externally installed and configured for any Thunder device.

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

- 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 Metric](#).
- 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. Additionally, logs can also be sent to any AWS, Azure, or VMware platforms.
- Manages the data collection, processing, aggregation, and publishing internally.
- Provides multitasking capabilities to collect and process multiple Thunder instances simultaneously.
- 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.
- Supports 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](#), and [VMware ESXi](#).

The following figure shows the TOA workflow.

Figure 1 : TOA Workflow



Supported Information

The following topics are covered:

- [Supported Technology](#)
- [Supported Thunder Metric](#)
- [Supported Log Thunder](#)
- [Supported ACOS versions](#)
- [Supported Cloud Apps](#)

Supported Technology

TOA supports the following technologies:

Table 1 : Supported Technologies

Name	Version	License
Python	3.10	PSF License Python 3.3 license Python.org

Table 1 : Supported Technologies

Name	Version	License
Requests	2.27.1	Apache Software License 2.0
Boto3	1.24.25	Apache 2.0 (amazon.com)

Supported Thunder Metric

TOA supports the following Thunder metrics:

Table 2 : 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 (Global/BPS)	Total Thunder system global throughput bits per sec from 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, ip-nat-conns-per-sec).
Transactions Rate (Sec)	Count of the total L7 transactions made per second from Thunder to the server for the last data collection cycle.

Table 2 : Supported Thunder Metrics

Metric	Description
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 code 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
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 Log Thunder

TOA supports the following Thunder logs:

Table 3 : Supported Thunder Logs

Logs	Description
SysLogs	<p>Thunder internal logs such as:</p> <ul style="list-style-type: none"> • SSL connection, negotiate, encrypt, and decrypt • Status codes 4xx and 5xx

Supported ACOS versions

TOA supports the following ACOS versions:

Table 4 : Supported ACOS versions

ACOS Version	TOA Version
64-bit Advanced Core OS (ACOS) version 6.0.0-p2, build 22	1.0.0
64-bit Advanced Core OS (ACOS) version 6.0.0-p1, build 47	1.0.0
64-bit Advanced Core OS (ACOS) version 5.2.1-p7, build 160	1.0.0
64-bit Advanced Core OS (ACOS) version 5.2.1-p6, build 74	1.0.0
64-bit Advanced Core OS (ACOS) version 4.1.4-GR1, build 34	1.0.0

Supported Cloud Apps

TOA supports the following Cloud platforms and monitoring applications:

Table 5 : Supported Cloud platforms and their monitoring applications

Cloud Platform	Monitoring Applications
AWS	<ul style="list-style-type: none">• CloudWatch
Azure	<ul style="list-style-type: none">• Application Insights• Log Analytics Workspace
VMware ESXi	<ul style="list-style-type: none">• vRealize Operations Manager (vROps)• vRealize Log Insight (vRLI)

Install TOA

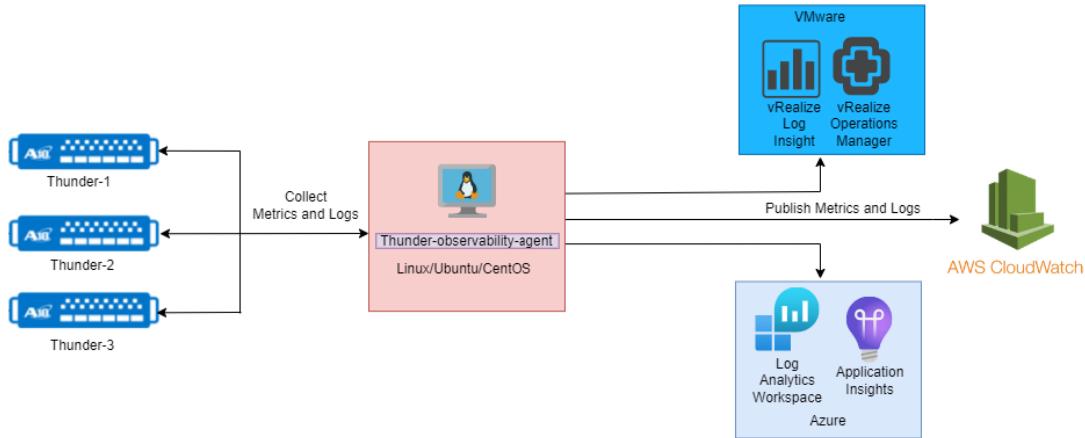
TOA is a standalone software that can be installed on any orchestration platform. The following installation options are available:

- [Python Plugin Installation](#)

TOA is installed on Linux/CentOS/Ubuntu platform using a Python plugin and is available in both online and offline installation modes.

[Figure 2](#) illustrates the installation of TOA in the Python plugin architecture.

Figure 2 : Python Plugin Installation Architecture

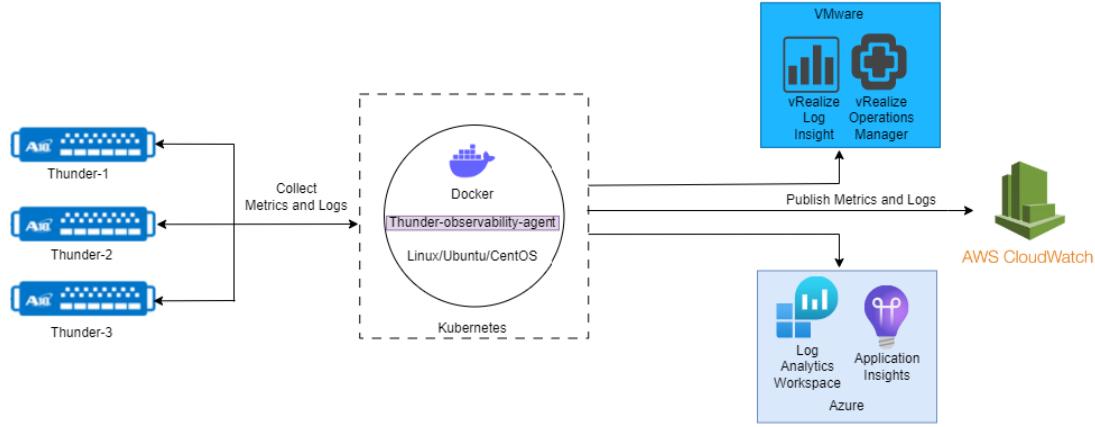


- [Containerized Installation](#)

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.

Python plugin can install TOA either in the online or offline mode.

The following topics are covered:

- [Prerequisites](#)
- [Installation Steps](#)

Prerequisites

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

Hardware Dependencies

Table 6 : Hardware Dependencies

Requirement	Description
Virtual Machine	2 GB RAM, 1 CPU, 4 GB

Table 6 : Hardware Dependencies

Requirement	Description
	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 7 : Software Dependencies

Requirement	Description
Operating System	<ul style="list-style-type: none"> • CentOS 7 or higher • Ubuntu 20 or higher
Python	3.6 or higher (Recommended Python 3.10)
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 the Python (recommended version), Crontab, and Syslog. For installation steps, see [Install Python](#). If the version is already installed, skip this step.
3. Create a virtual environment.

```
pip3 install virtualenv
cd /usr
virtualenv toaenv -p python3.10
source toaenv/bin/activate
```

4. Install TOA using any of the following options:

Online

Download the Python Plugin [Online](#) installation files.

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

Offline

Download the executable from [Offline](#) package and then run the following command:

```
pip3 install thunder_observability_agent-1.0.0-py3-none-any.whl
```

After the execution of the above command, the TOA default environment variable is set to `TOA_CONFIG_PATH = /usr/toaenv/thunder-observability-agent` and the following files are installed at `/usr/toaenv/thunder-observability-agent` path:

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

5. If you want to change the default location of the TOA config files, update the [Main Properties](#) and [Logging](#) files.

If not, skip this step.

6. Run `init.sh`, a one-time execution script, to enable the data collection frequency, data collection, and publish settings:

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

If the execution of `init.sh` is successful, the following configuration files are installed in the `/root/` folder and a cronjob is scheduled:

- `.thunder/credentials`
- `.aws/config`
- `.aws/credentials`

- .azure/credentials
- .vmware/credentials

7. To change the following default settings, update the [Crontab](#) file:

- Default data collection frequency
- Default recommended Python version (if your installed Python version is different)
- Default TOA package installation directory.

If not, skip this step.

8. Verify if the `agent.log` file is created at the `/var/log/thunder-observability-agent` path. For sample `agent.log` file, see [TOA Logging](#).

9. Update the appropriate parameters in the following files to collect the Thunder data and publish the selected metrics or logs on your required cloud platform. This step is mandatory.

Table 8 : Cloud specific Configuration Files

Cloud	File name
AWS	<ul style="list-style-type: none"> • Thunder Credentials • AWS Config • AWS Credentials • Config JSON
Azure	<ul style="list-style-type: none"> • Thunder Credentials • Azure Credentials • Config JSON
VMware	<ul style="list-style-type: none"> • Thunder Credentials • VMware Credentials • Config JSON

10. 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](#)
- [Installation Steps](#)

Prerequisites

The following tables lists the prerequisites for installing TOA using Containers:

- Kubernetes environment
- Download the [Kubernetes TOA manifest files](#).
- Download the [Docker Container](#) 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. Configure the corresponding cloud provider YAML file to publish the data collected from Thunder instances:

Table 9 : Cloud specific Configuration Files

Cloud	File name	Reference
AWS	aws-configmap.yaml	• Main Properties

Table 9 : Cloud specific Configuration Files

Cloud	File name	Reference
		<ul style="list-style-type: none"> • Config JSON • Logging
Azure	aws-secret.yaml	<ul style="list-style-type: none"> • AWS Config • AWS Credentials • Thunder Credentials
	azure-configmap.yaml	<ul style="list-style-type: none"> • Main Properties • Config JSON • Logging
VMware	azure-secret.yaml	<ul style="list-style-type: none"> • Azure Credentials • Thunder Credentials
	vmware-configmap.yaml	<ul style="list-style-type: none"> • Main Properties • Config JSON • Logging
	vmware-secret.yaml	<ul style="list-style-type: none"> • VMware Credentials • Thunder Credentials

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. To change the following default Crontab settings, update the <cloud-provider>-pod.yaml or <cloud-provider>-cronjob.yaml file:

- Default data collection frequency
- Default recommended Python version (if your installed Python version is different)
- Default TOA package installation directory.

If not, skip this step.

7. Verify if the `agent.log` file is created at the `/var/log/thunder-observability-agent` path. For example, see [TOA Logging](#).
8. Monitor Thunder metrics and logs. For more information, see [Monitor Dashboard](#).

NOTE: By default, the system works using all 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 connection with TOA.

The following topics are covered:

Global Configuration	19
Cloud-specific Configuration	24
Data Collection Configuration	28
Data Publish Configuration	31

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 10 : 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 Time as:	0

Table 10 : File Parameters

Parameter	Description	Default Value
	<p>Start Time = Last data collection time</p> <p>End Time = Current data collection time - <log_collection_delay_min></p> <p>Example</p> <p>If the current data collection time is 10:00:00 AM and the last data collection time is 09:59:00 AM, then:</p> <p>the Start Time is 9:59:00 AM.</p> <p>the End Time is 10:00:00 AM (which is 10:00:00 AM - 0 minutes).</p> <p>So, TOA collects all the logs generated by Thunder instance from 9:59:00 AM to 10:00:00 AM.</p>	
cron_job_frequency_min	<p>Specifies the cron job frequency in minutes.</p> <p>This parameter should match with the crontab -e job definition. The system considers crontab -e for job scheduling. If the frequency is changed in this parameter, it should also change in the crontab file.</p> <p>For more information, see Crontab.</p>	1
http_ssl_verify	<p>Disables SSL certificate verification over HTTPS.</p> <p>If a user wants to enable SSL:</p> <ul style="list-style-type: none"> For CA signed certificate configured in Thunder, set the parameter to True. 	False

Table 10 : File Parameters

Parameter	Description	Default Value
	<p>NOTE: For a self-signed certificate configured in Thunder, create a *.pem file, import Thunder public certificate, and provide the path in place of True.</p> <p>Example <code>/usr/toaenv/thunder-observability-agent/toa.pem</code></p> <hr/> <p>NOTE: If vROps and vRLI has self-signed certificates, then their public certificates must be imported in *.pem file.</p>	
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.	6
max_threads	Specifies the maximum number of threads to be created at the same time.	60
config_path	Specifies the configuration file path for publishing logs and metrics.	/usr/toaenv/thunder-observability-agent/config.json
thunder_credentials_path	<p>Specifies the configuration file path to collect data from any of the following:</p> <ul style="list-style-type: none"> • Single Thunder Instance • Multiple Thunder Instances • Thunder Instances in AWS Autoscaling Group • Thunder Instances in Azure VMSS. 	/root/.thunder/credentials

Table 10 : File Parameters

Parameter	Description	Default Value
aws_credentials_path	<p>Specifies the AWS credentials file path to establish connection and publish the data to AWS CloudWatch.</p> <p>NOTE: Applicable only if you want to publish the Thunder data to AWS CloudWatch.</p>	/root/.aws/credentials
aws_config_path	<p>Specifies the AWS configuration file path to publish the data.</p> <p>NOTE: Applicable only if you want to publish the Thunder data to AWS CloudWatch.</p>	/root/.aws/config
azure_credentials_path	<p>Specifies the Azure credentials file path to establish the connection and publish the data.</p> <p>NOTE: Applicable only if you want to publish the Thunder data to Azure Application Insights and Azure Log Analytics Workspace.</p>	/root/.azure/credentials
vmware_credentials_path	<p>Specifies the VMware credentials file path to establish the connection and publish the data.</p> <p>NOTE: Applicable only if you want to publish the Thunder data to VMware vROps.</p>	/root/.vmware/credentials

Logging

This file lists the TOA logging configurations.

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

Configure TOA

```
[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 change the default data collection frequency, perform the following steps:

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

```
$ crontab -e
```

2. Edit the following in the crontab file, as appropriate:

```
*/* * * * /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 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 VMware vROps)

AWS Config

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

File Path: /root/.aws/config

Update the following parameters according to your AWS setup:

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

Table 11 : 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 metrics or logs.

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 12 : 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 metrics or logs.

File Path: /root/.azure/credentials

Update the following parameters according to your Azure setup:

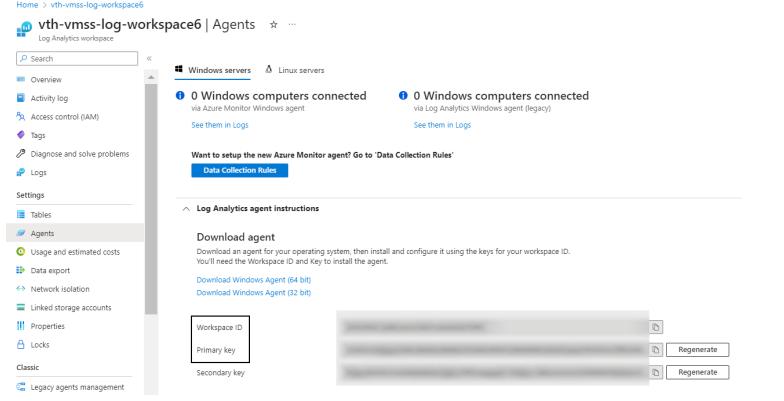
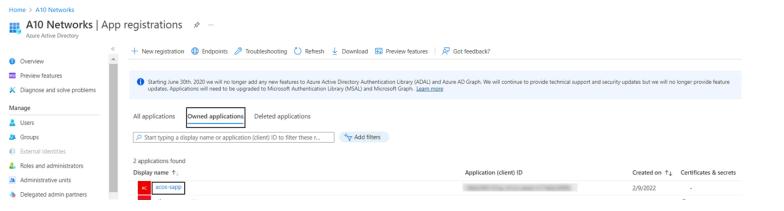
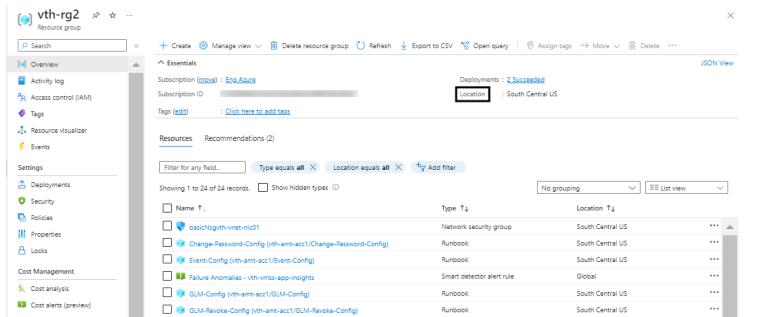
```
azure_workspace_primary_key = XXXX
azure_client_id = XXXX
azure_secret_id = XXXX
azure_tenant_id = XXXX
azure_location = XXXX
```

Table 13 : 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

Configure TOA

Table 13 : Azure Credentials File Parameters

Parameter	Description
	
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 .
	

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 14 : VMware Credentials File Parameters

Parameter	Description
vmware_vrops_username	Specifies your vROps login credentials.
vmware_vrops_password	

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 Autoscaling Group](#)
- [Thunder Instances in Azure VMSS](#)

Thunder Credentials

This file lists the Thunder credential configurations to collect the 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 in the below mentioned format:

Single Thunder Instance

Provide the details of Thunder instance running on any platform.

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

Multiple Thunder Instances

Provide the details of Thunder instances running on any platform.

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

Thunder Instances in AWS Autoscaling Group

Provide the details of Thunder instances running in AWS Autoscaling Group.

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

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"
        }
    ]
}
```

Table 15 : Thunder Credentials File Parameters

Parameter	Description
autoscale	Specify 1 if the Thunder instance is in AWS autoscale 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 autoscale group or Azure virtual machine scale set (<code>autoscale=1</code>). The following options are available: <ul style="list-style-type: none"> • aws • azure
thunders	Specifies the Thunder instance details. The following parameters are available:

Table 15 : Thunder Credentials File Parameters

Parameter	Description
	<ul style="list-style-type: none"> • 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	<p>Specifies the compute instance resource IDS on which Thunder is deployed.</p> <p>For more information, see Get Resource ID.</p>

For sample configuration, see [Examples](#).

Data Publish Configuration

The Thunder metrics and logs can be published on the cloud platforms such as AWS, Azure, or VMware. 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](#)
- Logs
 - [AWS](#)
 - [Azure](#)
 - [VMware](#)

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 below parameters to publish the Thunder metrics:

AWS

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

Table 16 : 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_metric_namespace	Specifies the AWS namespace where all the metrics must be published.	<aws_namespace>
aws_cpu	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on the AWS CloudWatch as the CPU Usage Percentage (Data) metrics. If the aws_provider and aws_metrics parameters	1

Table 16 : AWS Configuration Parameters

Parameter	Description	Default Value
	are enabled, TOA sends this metric to the AWS CloudWatch. By default, it is enabled.	
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. By default, it is enabled.	1
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_	Specify 1 to publish the deployed	1

Table 16 : AWS Configuration Parameters

Parameter	Description	Default Value
count	Thunder instances' server down count on the AWS CloudWatch. By default, it is enabled.	
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 Thunder instances web/app servers 4xx, 5xx errors count on the AWS CloudWatch. By default, it is enabled.	1
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.	1

Table 16 : AWS Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	

Azure

Configure the following parameters in the `config.json` to publish Thunder metrics to the Azure Application Insights.

Table 17 : Azure Configuration Parameters

Parameter	Description	Default Value
azure_provider	<p>Specify 1 to publish selected metric/s, logs, or both to Azure.</p> <p>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.</p>	0
azure_metric	<p>Specify 1 to send metrics to Azure Application Insights. It sends the data only if <code>azure_provider</code> is also enabled.</p> <p>By default, it is disabled.</p>	0
azure_metric_resource_id	<p>Specifies the Azure Application Insights resource ID.</p> <p>To get this value, go to Azure Portal > Azure services > Application Insights > <your_Thunder_instance> > Properties > Resource ID.</p> <p>Example</p> <pre>/subscriptions/07dxxxxxxxxxxxxx/ resourceGroups/ <resource_group_name>/ providers/microsoft.insights/ components/<app-insight-name></pre>	<azure_metric_resource_id>

Table 17 : Azure Configuration Parameters

Parameter	Description	Default Value
azure_cpu	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on the Azure Application Insights as the CPU Usage Percentage (Data) metrics. If the <code>azure_provider</code> and <code>azure_metrics</code> 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. By default, it is enabled.	1
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

Table 17 : Azure Configuration Parameters

Parameter	Description	Default Value
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_percentage	Specify 1 to publish the deployed Thunder instances' configured web/app servers down percentage on the Azure Application Insights. By default, it is enabled.	1
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.	1

Table 17 : Azure Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
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.

Table 18 : 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 <code>vmware_provider</code> 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 as the CPU Usage Percentage	1

Table 18 : VMware Configuration Parameters

Parameter	Description	Default Value
	(Data) metrics. If the <code>vmware_provider</code> and <code>vmware_metrics</code> parameters are enabled, TOA sends this metric to the VMware vROps. By default, it is enabled.	
<code>vmware_memory</code>	Specify 1 to publish the deployed Thunder instances' memory usage (percentage) on the VMware vROps. By default, it is enabled.	1
<code>vmware_disk</code>	Specify 1 to publish the deployed Thunder instances' storage disk on the VMware vROps. By default, it is enabled.	1
<code>vmware_throughput</code>	Specify 1 to publish the deployed Thunder instances' active throughput on the VMware vROps. By default, it is enabled.	1
<code>vmware_interfaces</code>	Specify 1 to publish the deployed Thunder instances' interfaces down count on the VMware vROps. By default, it is enabled.	1
<code>vmware_cps</code>	Specify 1 to publish the deployed Thunder instances' new connections per second on the VMware vROps. By default, it is enabled.	1
<code>vmware_tps</code>	Specify 1 to publish the deployed Thunder instances' transaction rate per second on the VMware vROps.	1

Table 18 : VMware Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
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
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	1

Table 18 : VMware Configuration Parameters

Parameter	Description	Default Value
	the VMware vROps. By default, it is enabled.	

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 19 : 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 <code>aws_provider</code> 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> .	< <code>aws_log_group_name</code> >

Azure

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

the Azure Log Analytics Workspace.

Table 20 : Azure Configuration Parameters

Parameter	Description	Default Value
azure_provider	<p>Specify 1 to publish the selected metric/s, logs, or both to Azure.</p> <p>By default, it is disabled and does not send logs to Azure. To publish logs, it is mandatory to enable Azure as a provider.</p>	0
azure_log	<p>Specify 1 to publish the logs to Azure Log Analytics Workspace. It sends the data only if <code>azure_provider</code> is also enabled.</p> <p>By default, it is disabled.</p>	0
azure_log_workspace_id	<p>Specifies the Azure Log Analytics Workspace ID.</p> <p>To get this value, go to Azure Portal > Azure services > Log Analytics workspaces > <your_log_analytics_workspace> > Settings > Agents.</p>	<azure_log_workspace_id>

VMware

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

Table 21 : VMware Configuration Parameters

Parameter	Description	Default Value
vmware_provider	<p>Specify 1 to publish the selected metric/s, logs, or both to VMware.</p> <p>By default, it is disabled and does not send logs to VMware. To publish logs, it is mandatory to enable VMware as a</p>	0

Table 21 : VMware Configuration Parameters

Parameter	Description	Default Value
	provider.	
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 > Virtual Machines > <your_vRLI_VM> > Networking > IP Address .	<vmware_vrli_host_or_ip>

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:

<u>Monitor Metrics</u>	44
<u>Monitor Logs</u>	68

Monitor Metrics

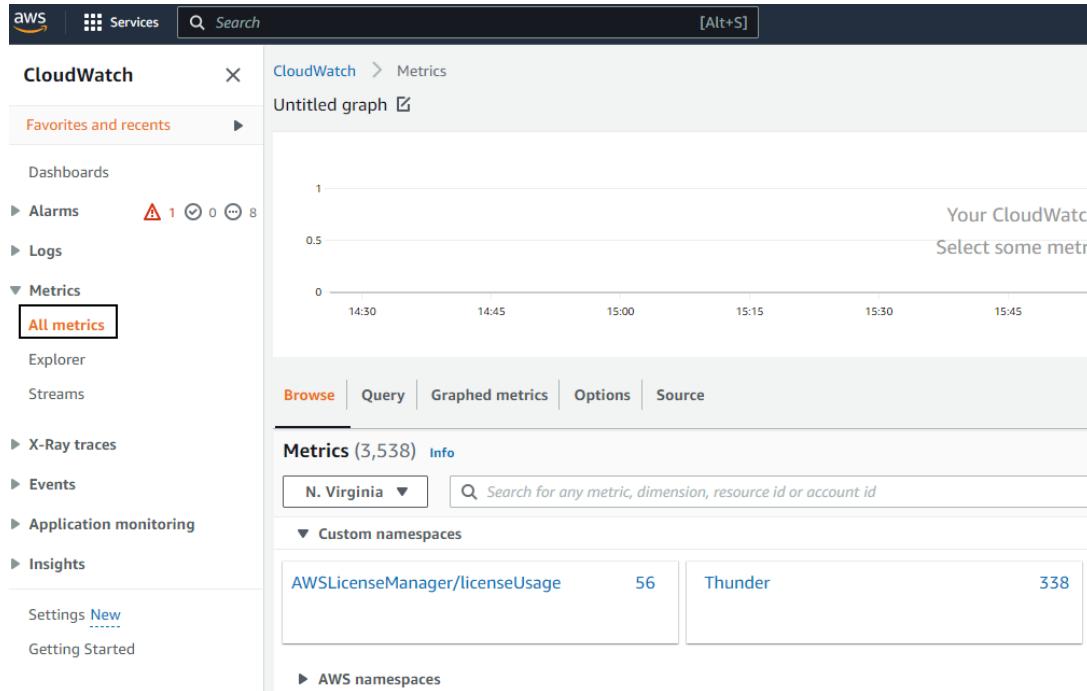
Depending on your cloud provider, the steps are provided to monitor the configured metrics.

AWS CloudWatch

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

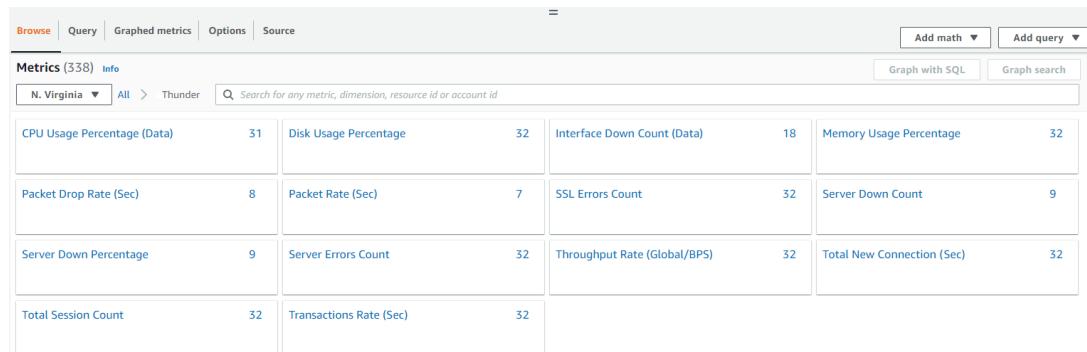
1. From the **AWS Management Console**, go to **CloudWatch > Metrics > All metrics**.

Figure 7 : AWS All metrics



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

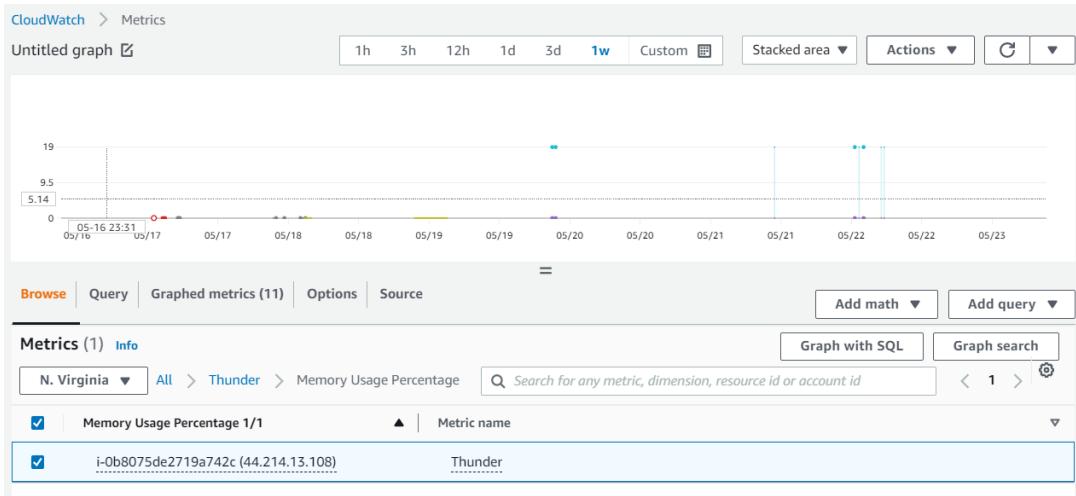
Figure 8 : Thunder metrics



3. Click the required metric to be monitored from the **Metrics** panel. For the list of available Thunder metrics, see [Supported Thunder Metric](#).
4. Select one or multiple Thunder instance/s resource ID 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



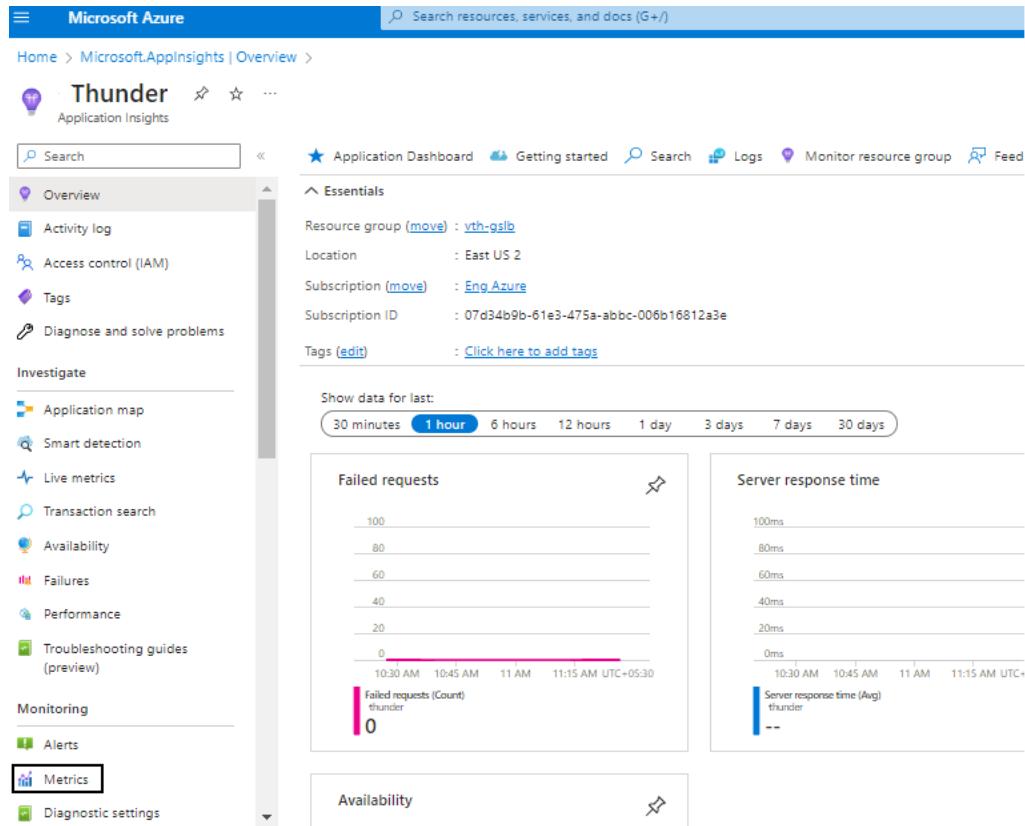
Azure Application Insight

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

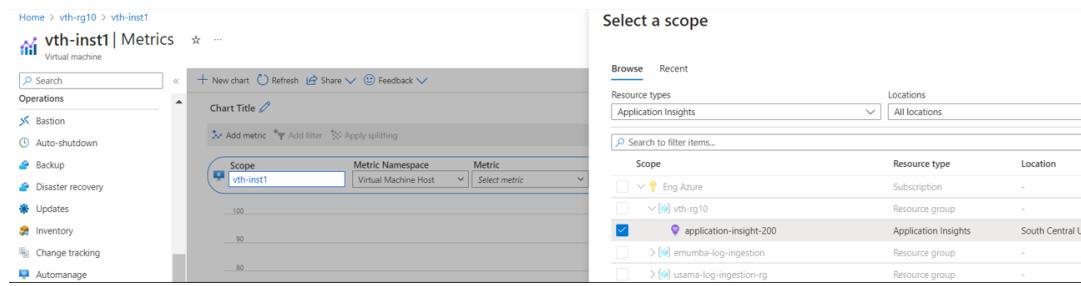


The screenshot shows the Azure Application Insights Overview page for the 'Thunder' instance. The left sidebar has 'Metrics' selected. The main area displays three charts: Failed requests, Server response time, and Availability.

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

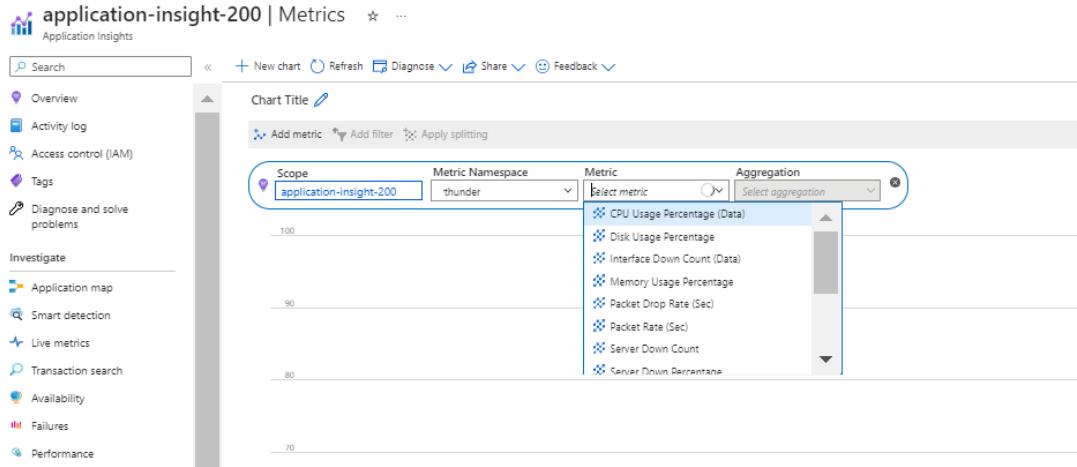


The screenshot shows the Azure portal's Metrics section for the 'vth-inst1' instance. The 'Metrics' tab is selected in the left sidebar. The right side shows the 'Select a scope' pane, which lists resources by type (Subscription, Resource group, Application Insights) and location. The 'application-insight-200' entry under 'vth-rg10' is selected.

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

A scope picker is displayed in the Metric dashboard.

Figure 12 : Scope Picker



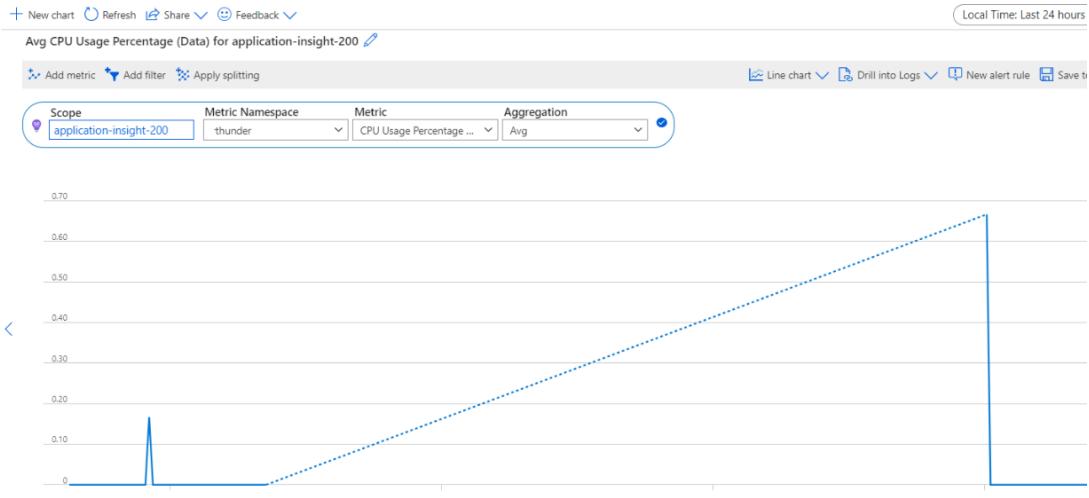
3. Select the appropriate resources whose metrics you want to view:

Table 22 : 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 Metric .

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



- 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:

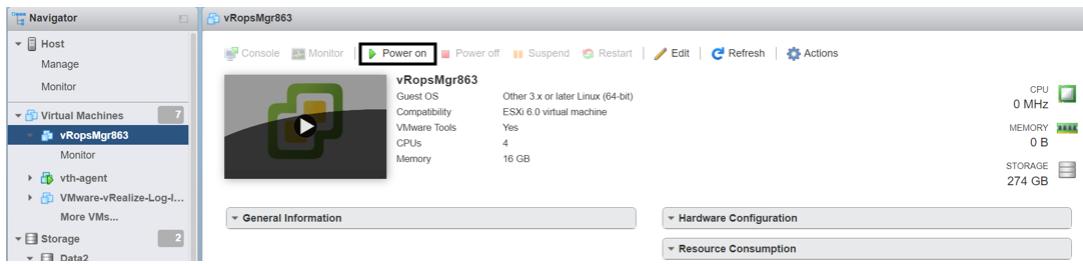
- [Start vROps VM](#)
- [Create a Dashboard](#)
- [Create an Alert](#)
- [Create a Notification](#)
- [View VMware Metrics](#)

Start vROps VM

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

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

Figure 14 : Start vROps VM

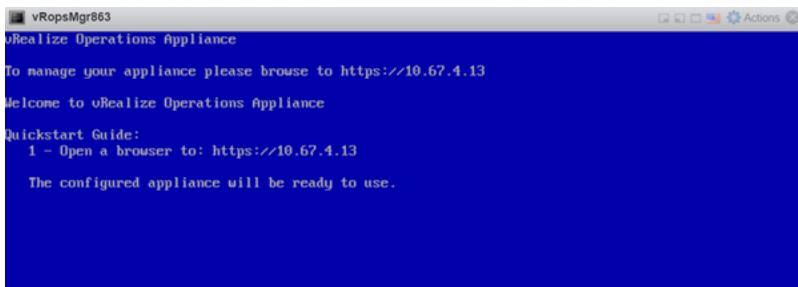


NOTE: The system may take a few minutes to start the vROps virtual machine.

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

The vROps virtual machine is powered on and is reachable.

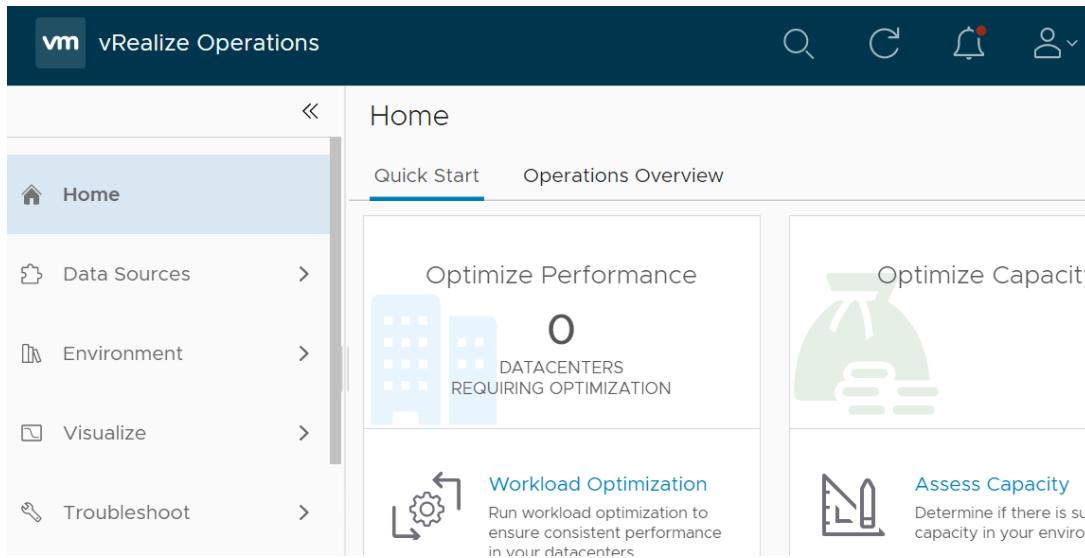
Figure 15 : vRealize Operations Appliance



- 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:

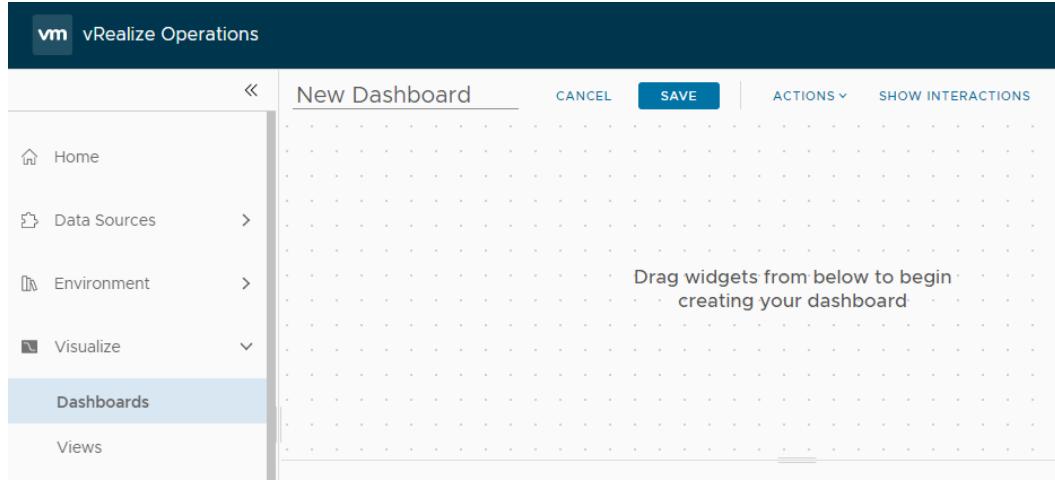
- Import a dashboard template
To import a dashboard using JSON file, see [Import a Dashboard](#).
- Create a dashboard manually

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.

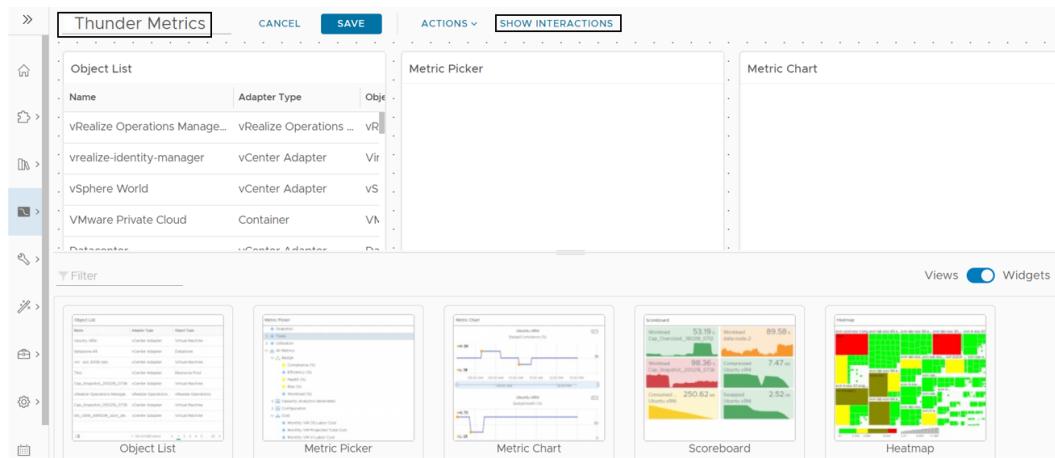
The New Dashboard window is displayed.

Figure 17 : New Dashboard window



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

Figure 18 : Dashboard widgets



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](#).

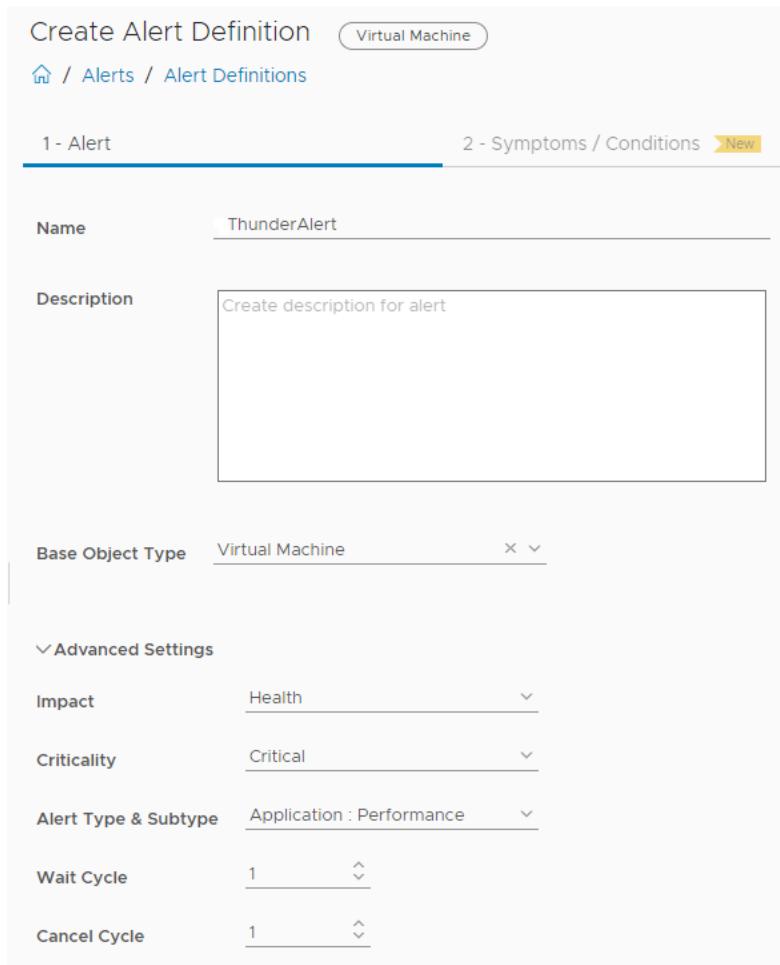
- Create an alert definition manually

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	<input type="text" value="ThunderAlert"/>
Description	<input type="text" value="Create description for alert"/>
Base Object Type	<input type="text" value="Virtual Machine"/> X ▼

▼ Advanced Settings

Impact	<input type="text" value="Health"/> ▼
Criticality	<input type="text" value="Critical"/> ▼
Alert Type & Subtype	<input type="text" value="Application : Performance"/> ▼
Wait Cycle	<input type="text" value="1"/> ^ ▼
Cancel Cycle	<input type="text" value="1"/> ^ ▼

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

Table 23 : 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.

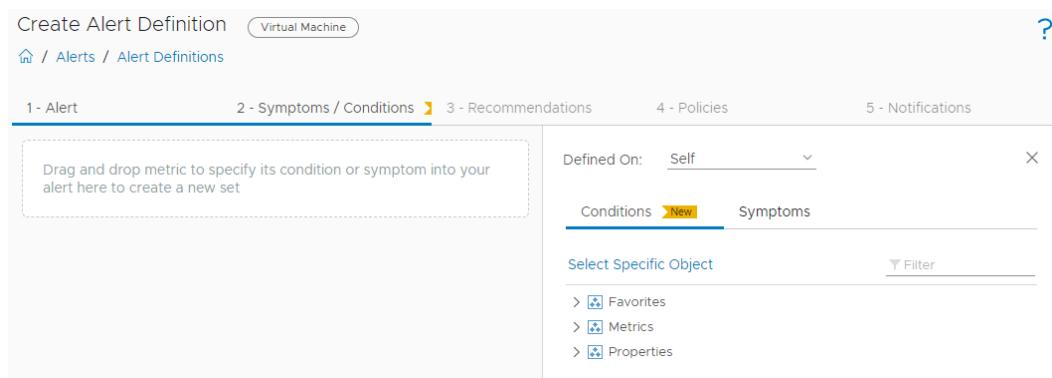
Table 23 : Alert tab fields

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.

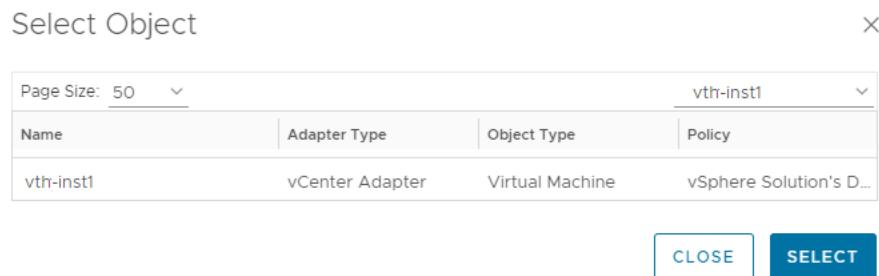
Figure 21 : Symptoms / Conditions tab



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

The **Select Object** window is displayed.

Figure 22 : Select Object window

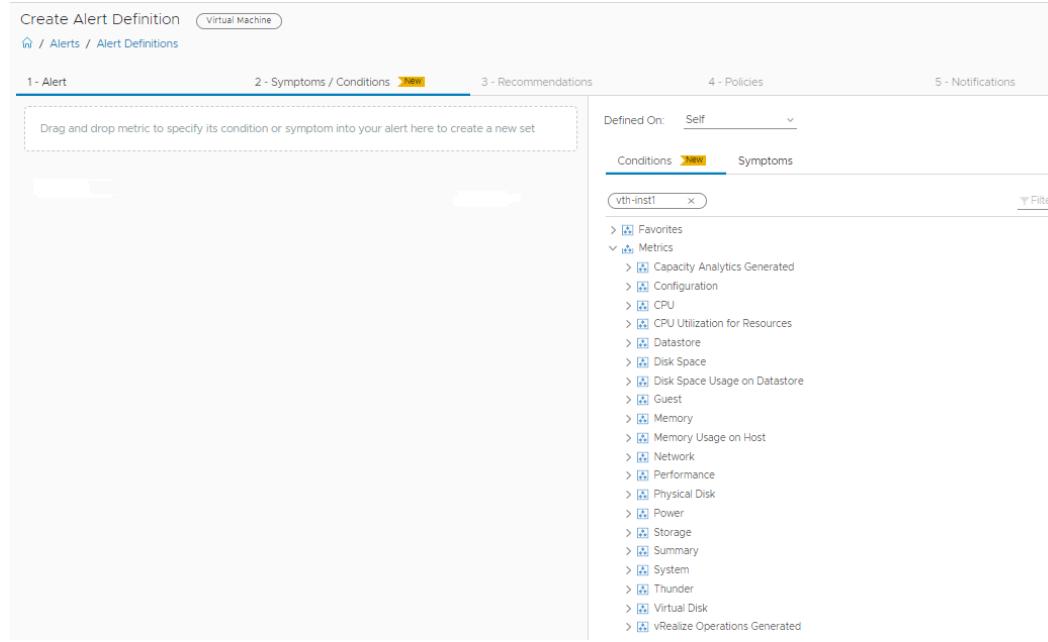


Name	Adapter Type	Object Type	Policy
vth-inst1	vCenter Adapter	Virtual Machine	vSphere Solution's D...

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

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

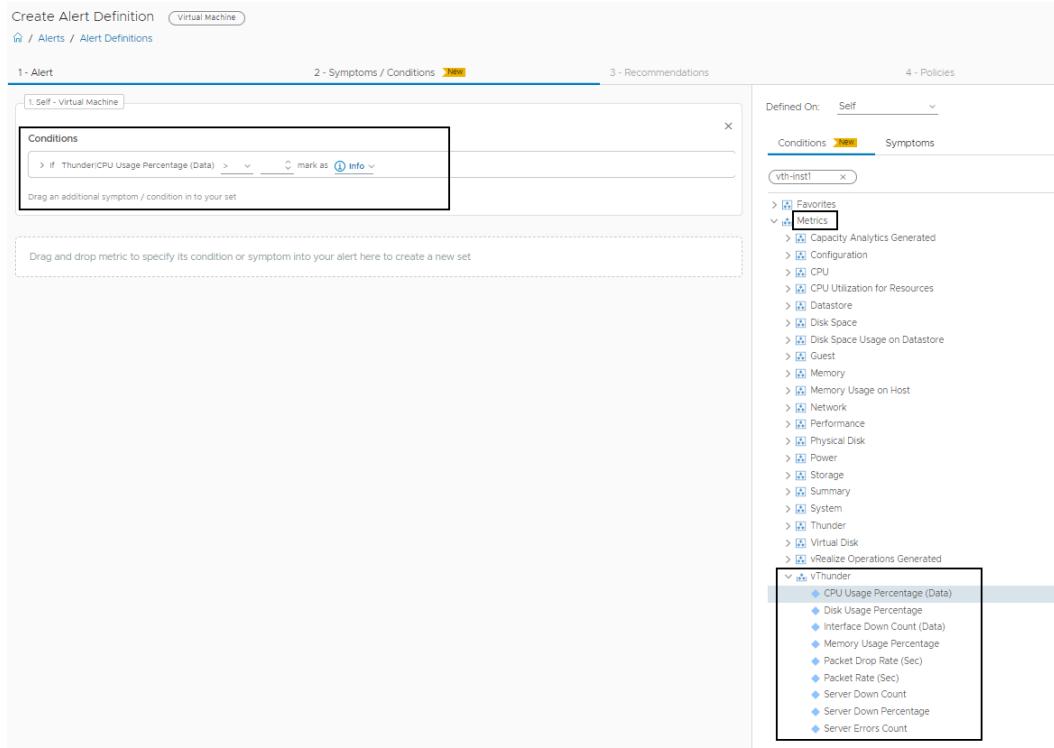
Figure 23 : Selected Thunder instance



The screenshot shows the 'Create Alert Definition' interface with the 'Virtual Machine' filter applied. The top navigation bar includes 'Create Alert Definition' (with a 'Virtual Machine' filter), 'Alerts / Alert Definitions', and tabs for '1 - Alert', '2 - Symptoms / Conditions', '3 - Recommendations', '4 - Policies', and '5 - Notifications'. The '2 - Symptoms / Conditions' tab is active, indicated by a yellow background. A sub-header 'Drag and drop metric to specify its condition or symptom into your alert here to create a new set.' is present. On the right, a sidebar titled 'Defined On: Self' shows a dropdown menu. Below it, the 'Conditions' tab is selected, showing a search bar with 'vth-inst1' and a 'Filter' button. A list of metrics is displayed, starting with 'Metrics' expanded, showing items like Capacity Analytics Generated, Configuration, CPU, CPU Utilization for Resources, Datastore, Disk Space, Disk Space Usage on Datastore, Guest, Memory, Memory Usage on Host, Network, Performance, Physical Disk, Power, Storage, Summary, System, Thunder, Virtual Disk, and vRealize Operations Generated.

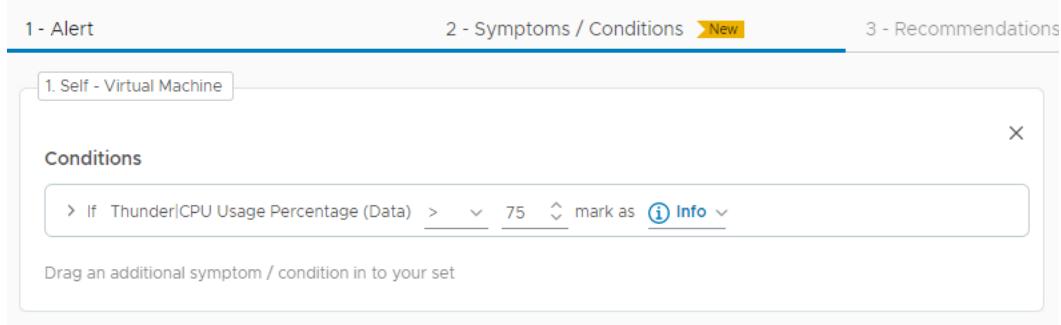
7. Select **Metrics > Thunder** and drag the required metrics to the left-side panel.

Figure 24 : Drag metric



8. Specify the appropriate alert condition.

Figure 25 : Alert condition

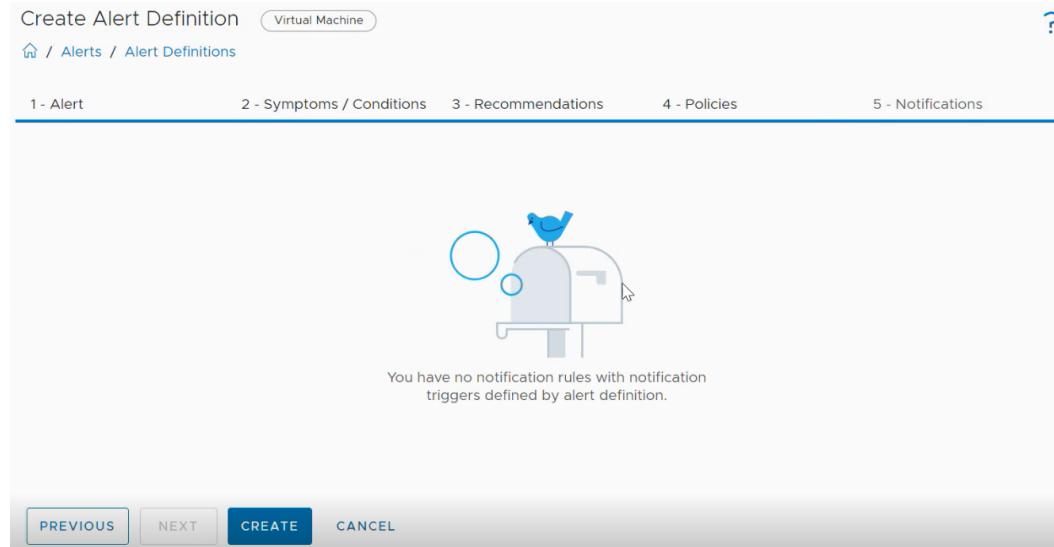


9. Click **Next**.

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**.

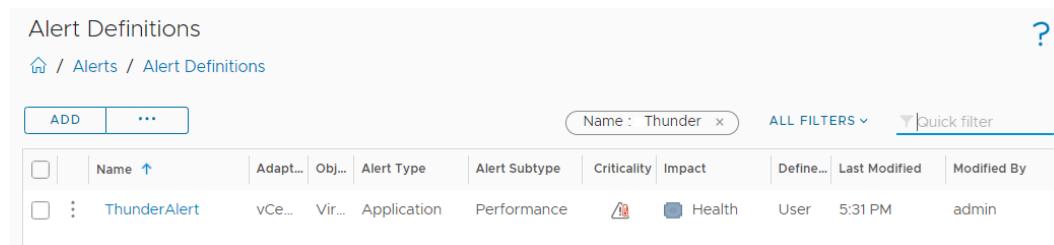
The **Notification** tab is displayed.



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



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](#).

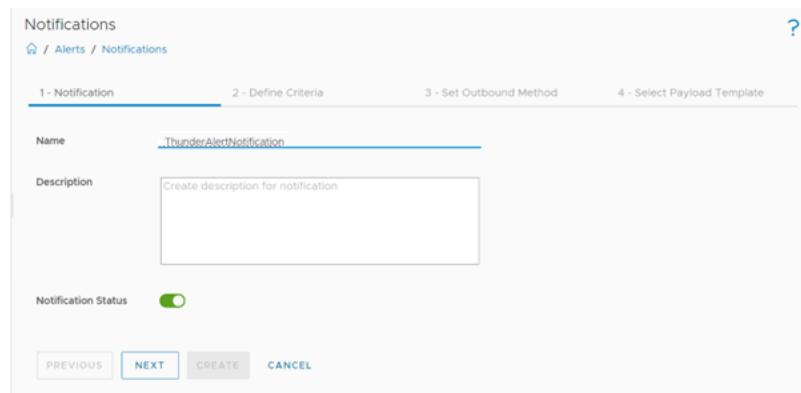
- Create a notification manually

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



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

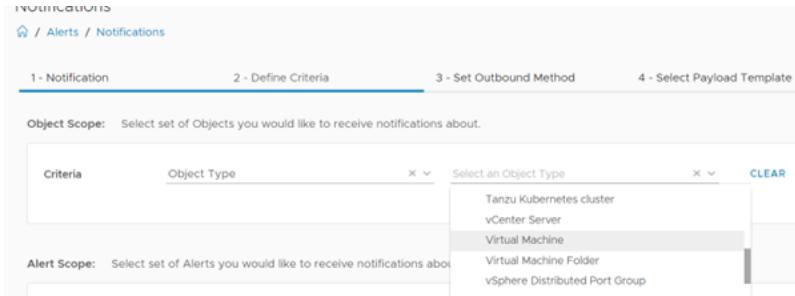
Table 24 : Notifications tab

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



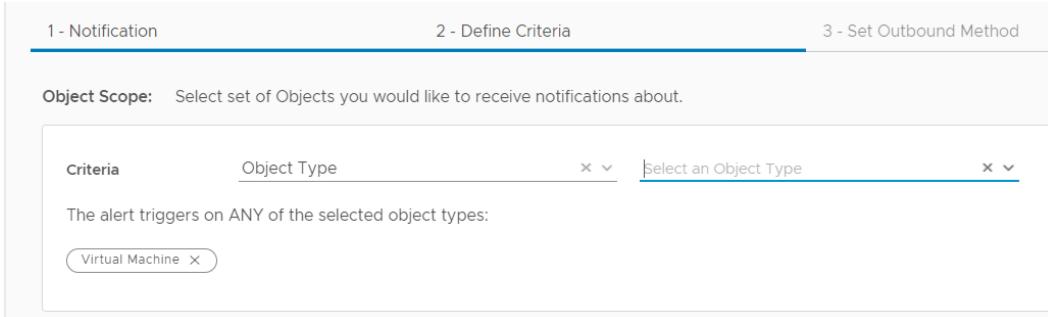
5. In the **Criteria** field, select **Object Type** from the drop-down.

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



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.

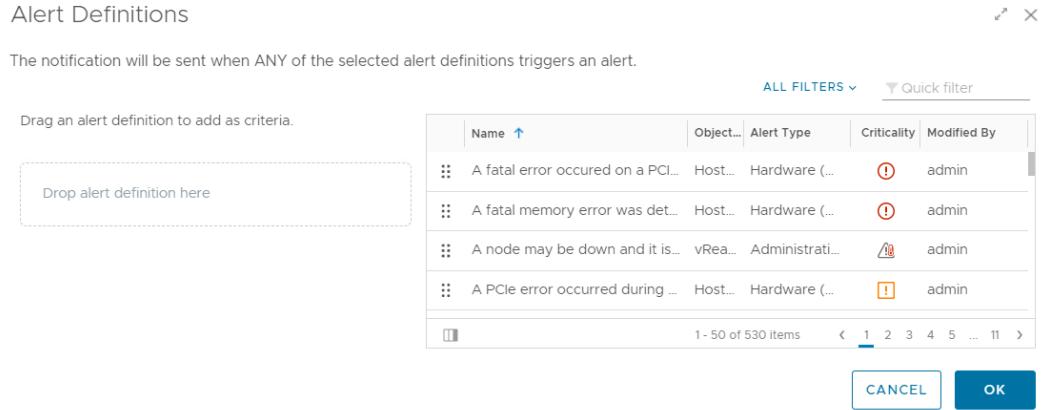
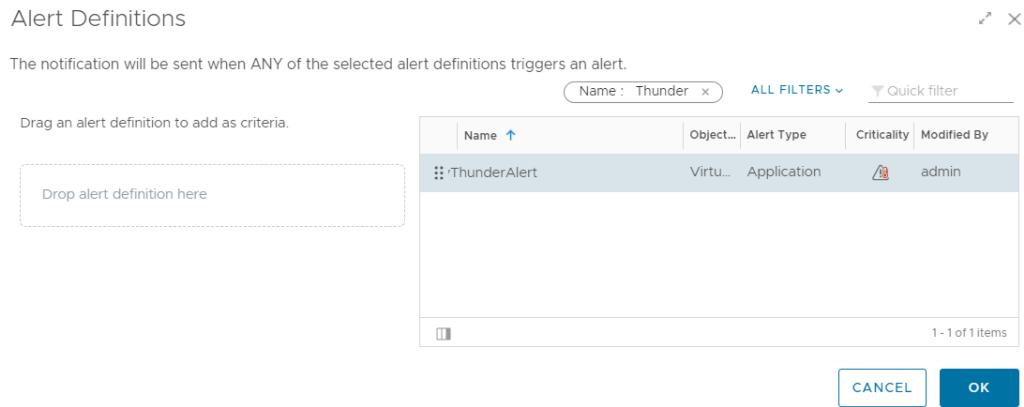
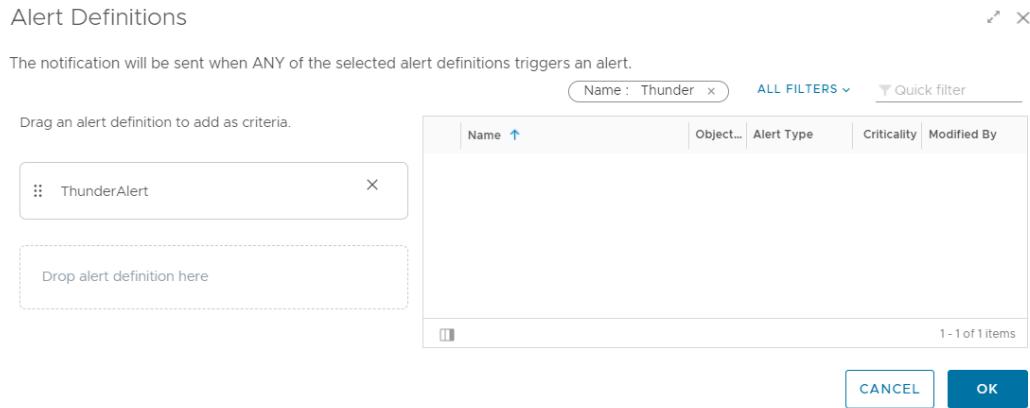
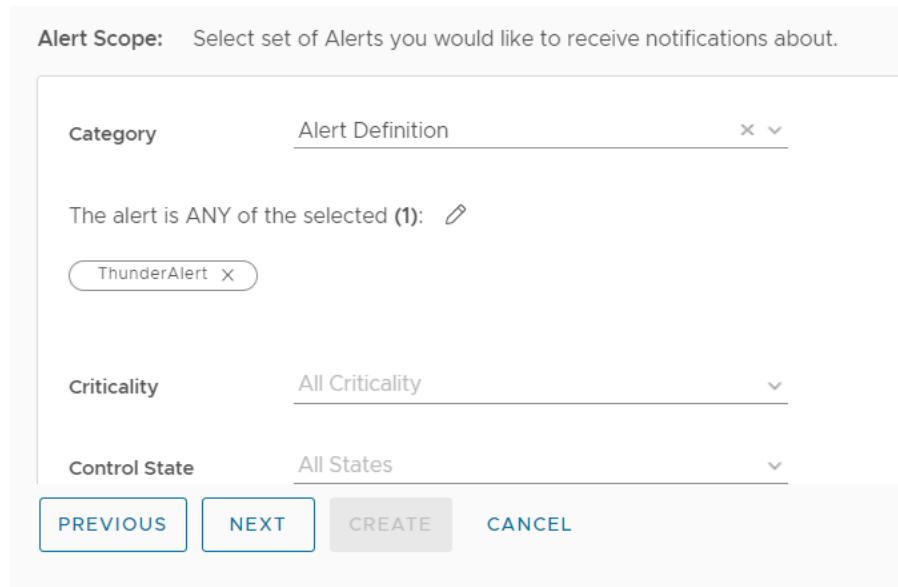
Figure 30 : Alert Definition pop-up**8. Search your alert definition.****Figure 31 : Search alert definition****9. Select your alert definition and drag it to add as the criteria.**

Figure 32 : Drag alert definition

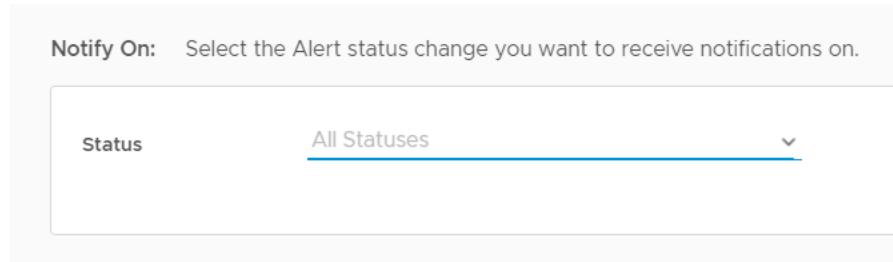
10. Click OK.

The selected alert definition is listed under Category.

Figure 33 : Selected alert definition

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

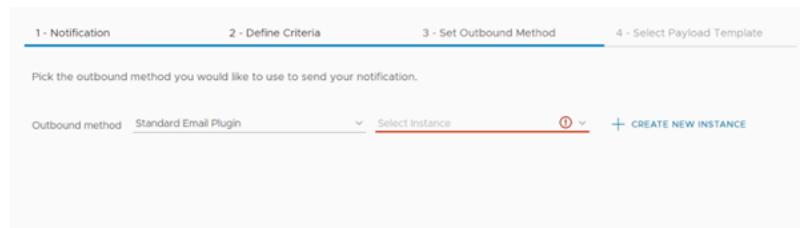
Figure 34 : Notify On



12. Click **Next.**

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

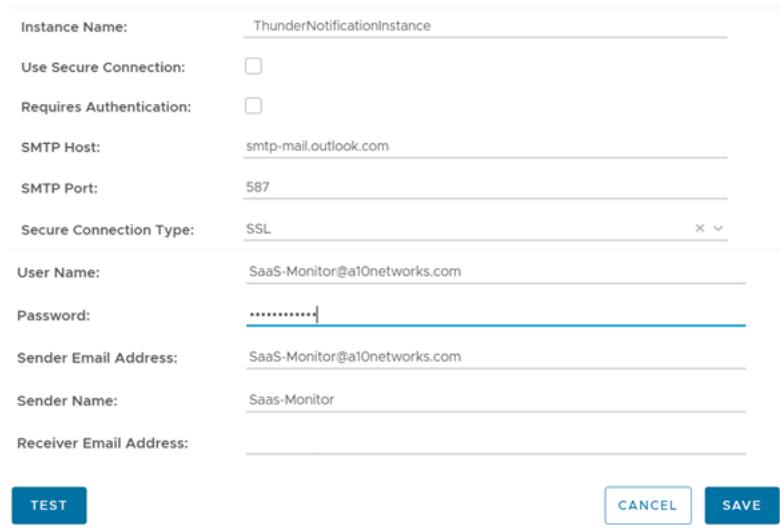
Figure 35 : Set Outbound Method tab



- 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:

TEST **CANCEL** **SAVE**

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

Table 25 : Create New Instance

Field Name	Description
Instance Name	<p>Enter the notification instance name.</p> <p>Example</p> <p>In the Figure 36, the notification instance name is ThunderNotificationInstance.</p>
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.

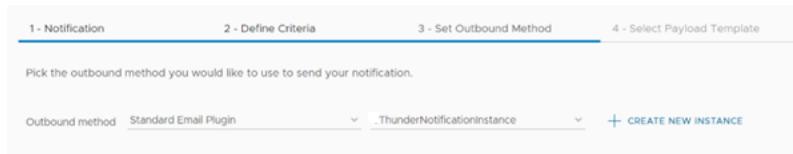
Table 25 : Create New Instance

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

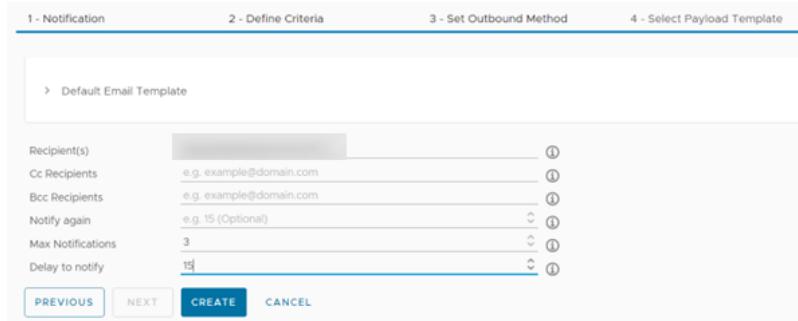


A screenshot of a dropdown menu titled "Select Instance". The menu lists "ThunderNotificationInstance" as the selected item. Other options visible include "Standard Email Plugin" and a "+ CREATE NEW INSTANCE" button.

17. Click **Next**.

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

Figure 38 : Select Payload Template tab



A screenshot of the "Select Payload Template" tab. The "Default Email Template" section is active. It shows fields for "Recipient(s)" (e.g. example@domain.com), "Cc Recipients" (e.g. example@domain.com), "Bcc Recipients" (e.g. example@domain.com), "Notify again" (e.g. 15 (Optional)), "Max Notifications" (3), and "Delay to notify" (15). Below the form are buttons for "PREVIOUS", "NEXT", "CREATE", and "CANCEL".

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

Table 26 : Select Payload Template tab

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

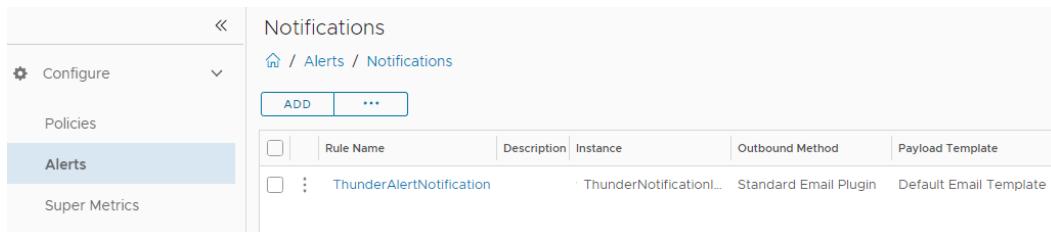
Table 26 : Select Payload Template tab

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
<input type="checkbox"/>	ThunderAlertNotification	ThunderNotification...		Standard Email Plugin	Default Email Template

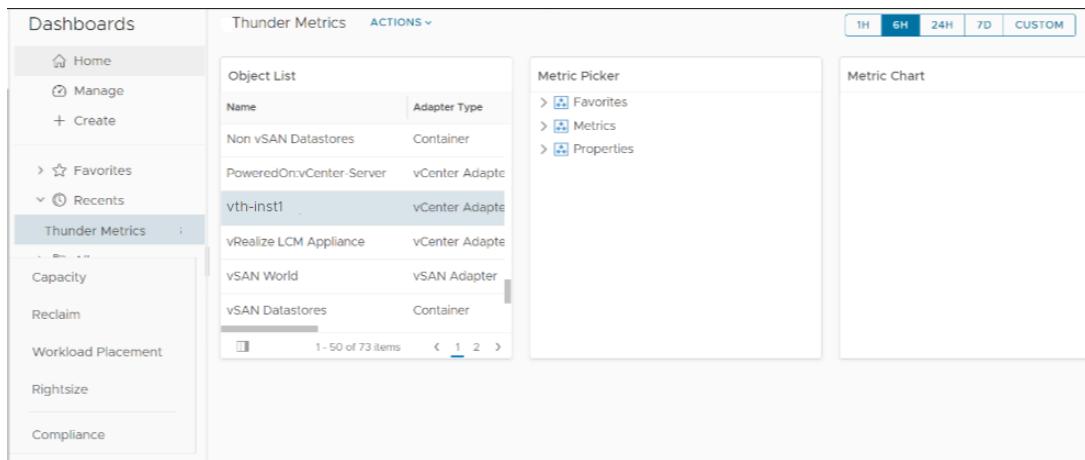
View VMware Metrics

To view the VMware 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

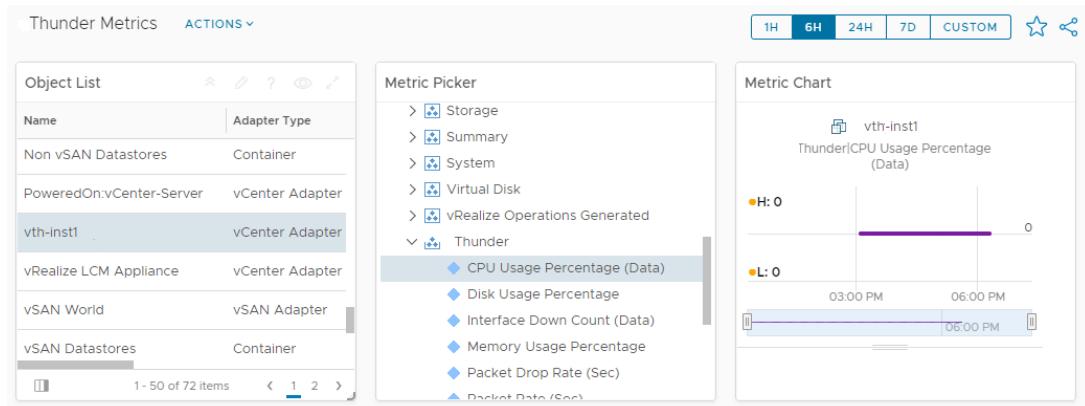


The screenshot shows the 'Monitor Dashboard' interface. On the left, a sidebar lists various monitoring categories like Home, Manage, Create, Favorites, Recents, Capacity, Reclaim, Workload Placement, Rightsize, and Compliance. The 'Thunder Metrics' option is selected. The main content area has three panels: 'Object List' (listing items like Non vSAN Datastores, PoweredOn:vCenter-Server, vRealize LCM Appliance, vSAN World, and vSAN Datastores), 'Metric Picker' (showing options for Favorites, Metrics, and Properties, with 'Metrics' expanded to show categories like Storage, Summary, System, Virtual Disk, vRealize Operations Generated, and Thunder, where 'CPU Usage Percentage (Data)' is selected), and a 'Metric Chart' panel which is currently empty.

2. From **Object List**, double-click your Thunder instance.
3. From **Metric Picker**, expand **Metrics > Thunder** and double-click the required metric.

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

Figure 41 : Thunder Metrics



This screenshot shows the 'Thunder Metrics' interface. The 'Object List' panel shows items like Non vSAN Datastores, PoweredOn:vCenter-Server, vRealize LCM Appliance, vSAN World, and vSAN Datastores. The 'Metric Picker' panel shows the 'Thunder' category expanded, with 'CPU Usage Percentage (Data)' selected. The 'Metric Chart' panel displays a chart for 'vth-inst1' showing 'Thunder|CPU Usage Percentage (Data)' over a 6-hour period from 03:00 PM to 06:00 PM. The chart shows a single data series starting at 0% usage at 03:00 PM and ending at 0% usage at 06:00 PM.

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 Metric](#).

Monitor Logs

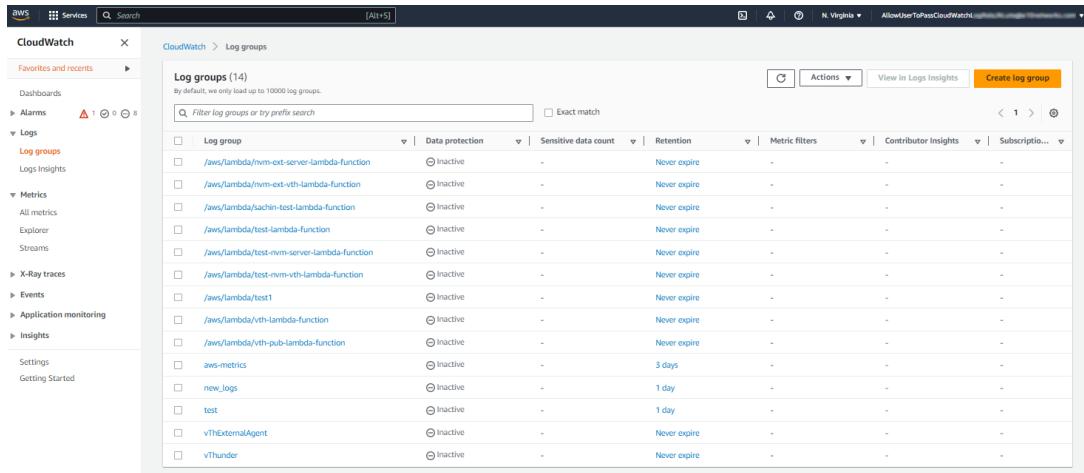
Depending on your cloud provider, the steps are provided to monitor the configured logs.

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**.

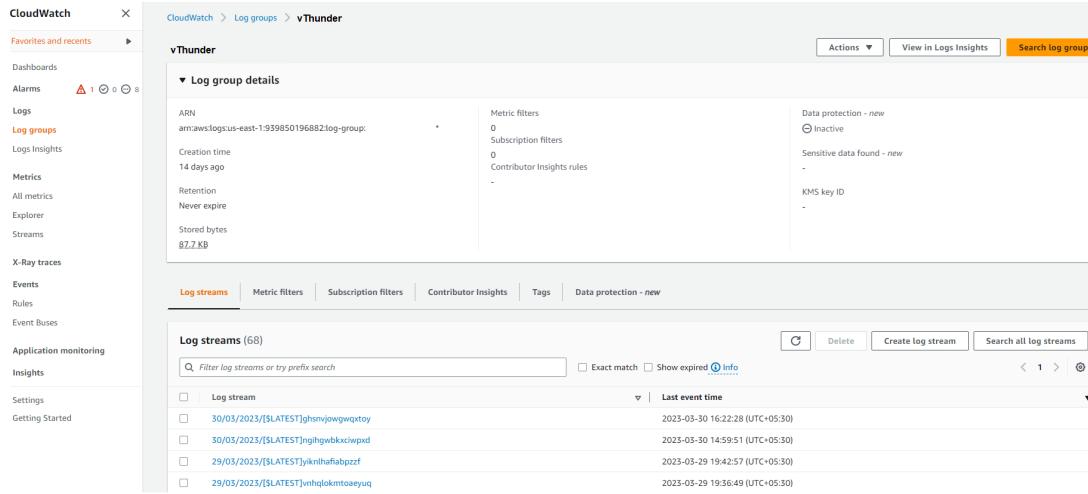
Figure 42 : AWS Log Groups



The screenshot shows the AWS CloudWatch interface with the 'Logs' section selected. On the left, there's a sidebar with various monitoring options like Alarms, Metrics, X-Ray traces, Events, Application monitoring, and Insights. The main area is titled 'Log groups (14)' and displays a table of log groups. Each row in the table represents a log group with columns for name, data protection (set to 'Inactive' for all), sensitive data count (empty), retention (set to 'Never expire' for all), metric filters (empty), contributor insights (empty), and subscriptions (empty). The log groups listed include '/aws/lambda/nvm-ext-server-lambda-function', '/aws/lambda/nvm-ext-veth-lambda-function', '/aws/lambda/test-veth-lambda-function', '/aws/lambda/test-nvm-server-lambda-function', '/aws/lambda/test-nvm-veth-lambda-function', '/aws/lambda/test1', '/aws/lambda/veth-lambda-function', '/aws/lambda/vth-lambda-function', 'aws-metrics', 'new_logs', 'test', 'vTHExternalAgent', and 'vThunder'.

2. Click **Thunder** log group.

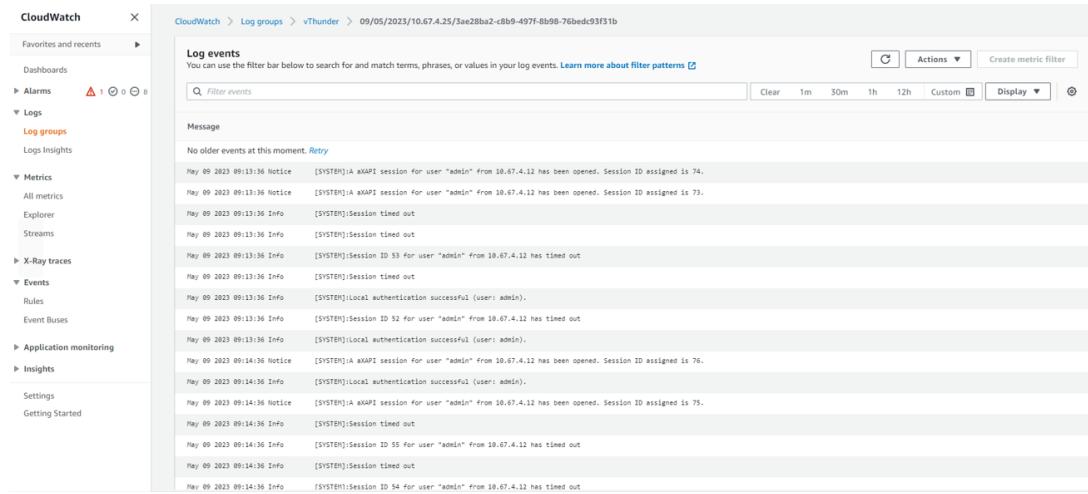
Figure 43 : Thunder Log Group



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 44 : Logs events on AWS CloudWatch



Azure Log Analytics Workspace

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

- From the **Azure Portal**, go to **Azure services > Resource Groups > <your_resource_group>** and click your log analytics workspace name.

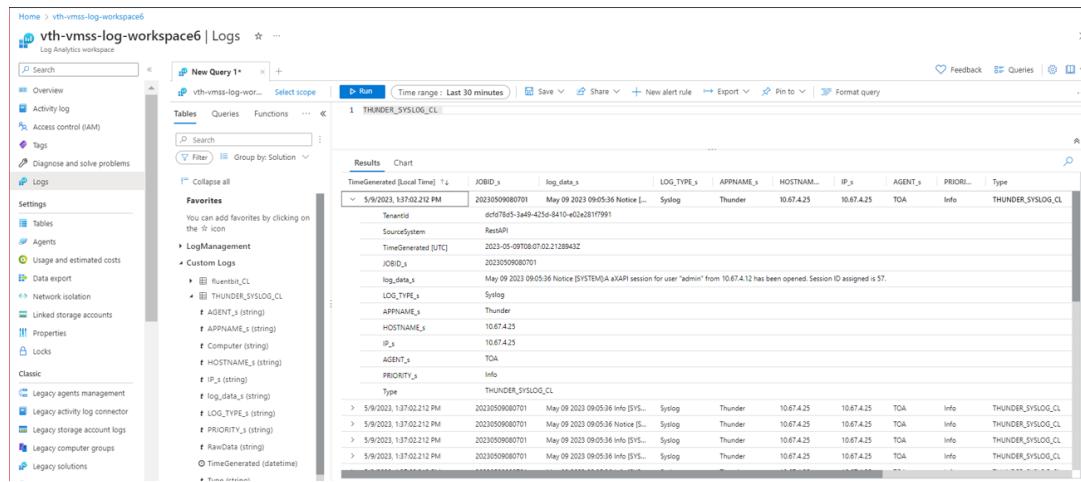
2. Click **Logs** from the left **General** panel.

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

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

The **THUNDER_SYSLOGS_CL** query window is displayed.

Figure 45 : Custom Logs window



TimeGenerated [Local Time]	JOBID_s	log_data_s	LOG_TYPE_s	APPNAME_s	HOSTNAME_s	IP_s	AGENT_s	PRIORITY_s	Type
5/9/2023, 1:37:02.212 PM	20230509080701	May 09 2023 09:05:36 Notice [SYSTEM]A xAPI session for user "admin" from 10.67.4.12 has been opened. Session ID assigned is 57.	Syslog	Thunder	10.67.4.25	10.67.4.25	TOA	Info	THUNDER_SYSLOG_CL
		dctf785-3a49-425d-8410-e02e281f7991							
		Resource							
		TimeGenerated [UTC]							
		2023-05-09T08:01:32Z							
		JOBID_s							
		20230509080701							
		AGENT_s							
		AGENT_s (string)							
		APPNAME_s							
		APPNAME_s (string)							
		Computer							
		Computer (string)							
		HOSTNAME_s							
		HOSTNAME_s (string)							
		IP_s							
		IP_s (string)							
		log_data_s							
		log_data_s (string)							
		LOG_TYPE_s							
		LOG_TYPE_s (string)							
		PRIORITY_s							
		PRIORITY_s (string)							
		RawData							
		RawData (string)							
		TimeGenerated							
		TimeGenerated (datetime)							

5. Click **Run**.

All logs are displayed in tabular format with expandable details.

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 46 : Start vRLI VM

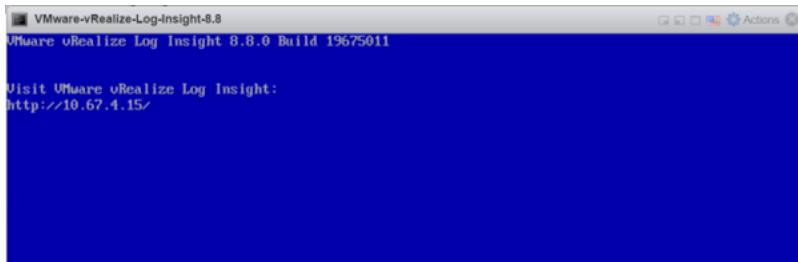
**NOTE:**

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.

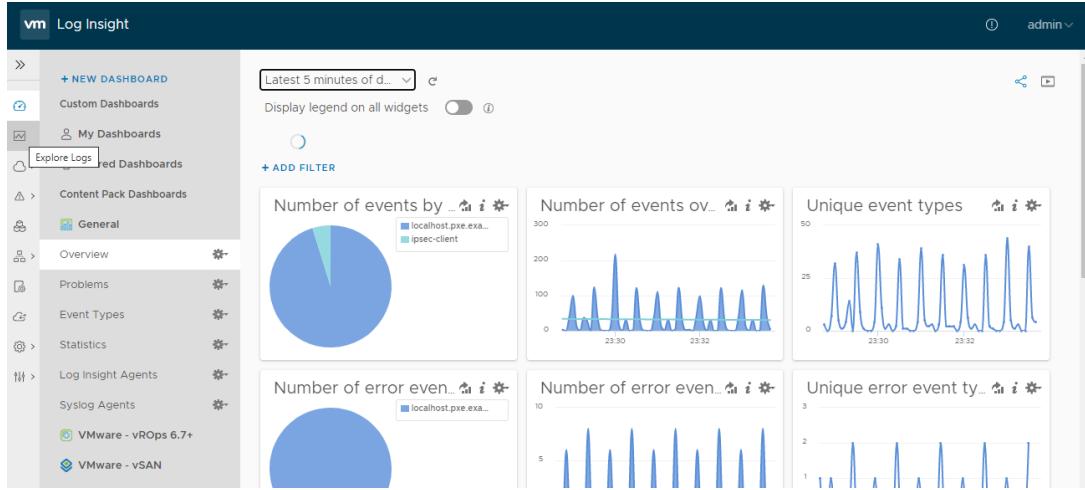
Figure 47 : VMware vRealize Log Insight

**View Logs**

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

The **Logs** window is displayed.

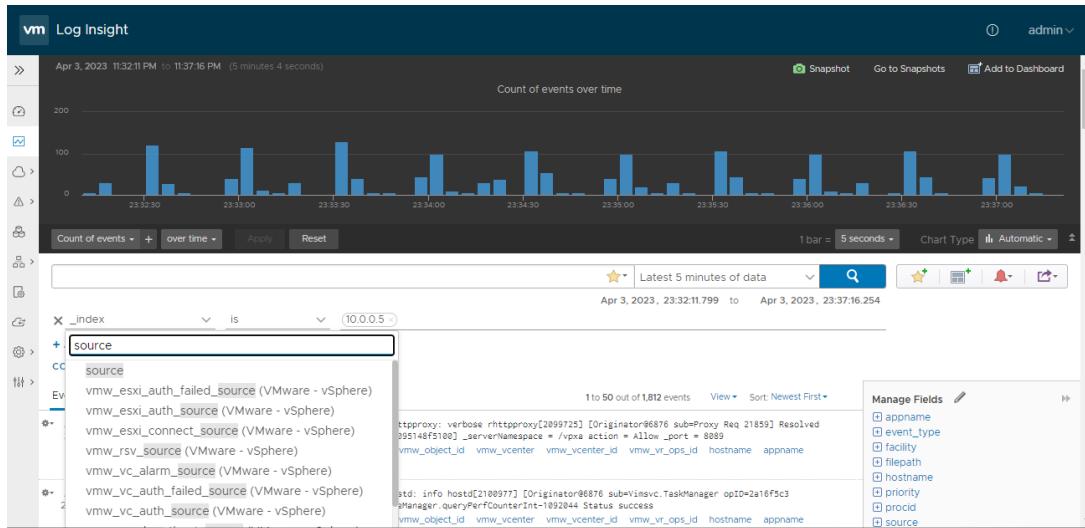
Figure 48 : vRealize Log Insight - Overview window



- 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 49 : vRealize Log Insight - Add Filter

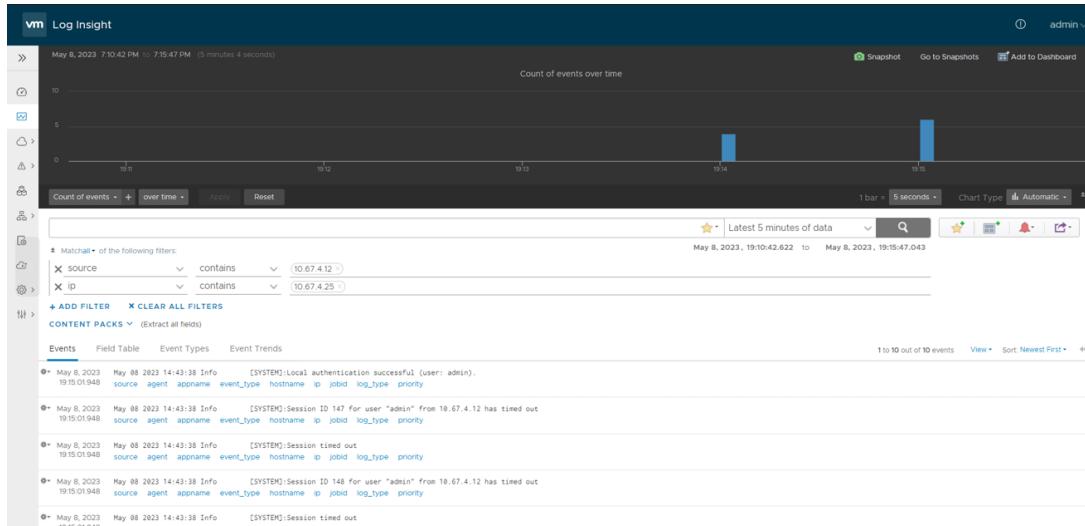


3. Add the following filter criteria to search all logs received from TOA:

- `_index: source`
- `condition: is`
- `value: <TOA_IP>`

4. Verify if the logs are generated.

Figure 50 : Logs on vRealize Log Insight



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:

```
2023-05-08 10:46:11,639 - INFO - Job No : 20230508051610.
2023-05-08 10:46:11,641 - INFO - Job Start Time : 2023-05-08
05:16:10.509064+00:00.
2023-05-08 10:46:11,645 - INFO - Job Execution : 0.521645 seconds.
2023-05-08 10:46:11,645 - INFO - Job End Time : 2023-05-08
05:16:11.644678+00:00
2023-05-08 10:46:11,646 - INFO - Documentation : www.a10networks.com
or https://github.com/a10networks/thunder-observability-agent.
2023-05-08 10:46:11,646 - INFO - ##### TOA ##### All Rights Reserved
@A10 Networks Inc #####
2023-05-08 10:46:11,639 - INFO - Job No : 20230508051610.
2023-05-08 10:46:11,641 - INFO - Job Start Time : 2023-05-08
05:16:10.509064+00:00.
2023-05-08 10:46:11,643 - ERROR - Error : File not found or
corrupt. Please check path: [/usr/toaenv/thunder-observability-
agent/config.json]. Application config not found. Please check [config_
path] in main.properties.
2023-05-08 10:46:11,644 - ERROR - Error : File not found or
corrupt. Please check path: [/root/.thunder/credentials]. Thunder
credentials not found. Please check [thunder_credentials_path] in
main.properties.
2023-05-08 10:46:11,645 - INFO - Job Execution : 1.135614 seconds.
2023-05-08 10:46:11,645 - INFO - Job End Time : 2023-05-08
```

[Troubleshoot](#)

```

05:16:11.644678+00:00
2023-05-08 10:46:11,646 - INFO - Documentation      : www.a10networks.com or
https://github.com/a10networks/thunder-observability-agent.
2023-05-08 10:47:20,253 - INFO - ##### TOA ##### All Rights Reserved
@a10 Networks Inc ##### TOA #####
2023-05-08 10:50:38,584 - INFO - Job No          : 20230508052038.
2023-05-08 10:50:38,586 - INFO - Job Start Time   : 2023-05-08
05:20:38.074260+00:00.
2023-05-08 10:50:38,593 - INFO - No of Thunders   : 1 ['10.67.4.25'].
2023-05-08 10:50:38,595 - WARNING - WARNING        : No log or metric is
enabled. To enable [metric, log set to [1]] in config.json.
2023-05-08 10:50:38,595 - INFO - Job Execution     : 0.521645 seconds.
2023-05-08 10:50:38,595 - INFO - Job End Time     : 2023-05-08
05:20:38.595905+00:00
2023-05-08 10:50:38,596 - INFO - Documentation      : www.a10networks.com or
https://github.com/a10networks/thunder-observability-agent.
2023-05-08 11:02:45,685 - INFO - ##### TOA ##### All Rights Reserved
@a10 Networks Inc ##### TOA #####
2023-05-08 11:03:53,685 - INFO - Job No          : 20230508053353.
2023-05-08 11:03:53,687 - INFO - Job Start Time   : 2023-05-08
05:33:53.158894+00:00.
2023-05-08 11:03:53,693 - INFO - No of Thunders   : 1 ['10.67.4.25'].
2023-05-08 11:03:53,694 - INFO - Log Provider     : VMWARE.
2023-05-08 11:03:53,694 - INFO - Log             : VMWARE_LOG.
2023-05-08 11:03:53,695 - INFO - Metric Provider    : VMWARE.
2023-05-08 11:03:53,695 - INFO - Metric           : VMWARE_METRIC.
2023-05-08 11:03:59,028 - INFO - Published Log     : 10.67.4.25 [Count: 8].
2023-05-08 11:04:00,364 - INFO - Published Metric : 10.67.4.25 [Count: 14]
[{'Server Down Percentage': 100.0, 'Server Down Count': 1, 'Packet Drop
Rate (Sec)': 0, 'Packet Rate (Sec)': 0, 'Server Errors Count': 0, 'Total
Session Count': 0, 'SSL Errors Count': 0, 'Interface Down Count (Data)': 0,
'Throughput Rate (Global/BPS)': 200, 'Transactions Rate (Sec)': 0,
'Total New Connection (Per Sec)': 0, 'Disk Usage Percentage': 39, 'CPU
Usage Percentage (Data)': 0.0, 'Memory Usage Percentage': 64.7}].  

2023-05-08 11:04:00,365 - INFO - Job Execution     : 7.206818 seconds.
2023-05-08 11:04:00,366 - INFO - Job End Time     : 2023-05-08
05:34:00.365712+00:00
2023-05-08 11:04:00,366 - INFO - Documentation      : www.a10networks.com or

```

```
https://github.com/a10networks/thunder-observability-agent.  
2023-05-08 11:04:00,366 - INFO - ##### TOA ##### All Rights Reserved  
@A10 Networks Inc ##### TOA #####
```

Examples

The following topics are covered:

AWS	77
Azure	81
VMware	86

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
<i>Thunder details</i>	
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

Parameter	Description
User Name	Online_Pokers_TH2
Password	Thunder@Borse@3202
Resource_Name	North_Virginia_Online_Pokers_TH2
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_metric_namespace	Thunder
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.

- 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"
        }
    ]
}
```

Multiple instances

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

Autoscale instance

```
{
    "autoscale" : 1,
```

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

- 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_metric_namespace": "Thunder",  
    "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
<i>Thunder details</i>	
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-1234567890rstuvwxyz0
Thunder instance	3
User Name	Online_Football_TH3
Password	Thunder@ABC@3203
Resource_Name	vth-auto-scale-group
<i>Azure Monitoring details</i>	
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	tewPsyMYkdGOThRjEy ***** ***** 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-xxxxf81fcbb2f
azure_log_workspace_id	dcf7xxx-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"
    } ]
}

```

Multiple instances

```

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

```

Autoscale instance

```

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

```

- 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~jIxxyEVyxxxxHNxxxOwv_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
<i>Thunder details</i>	
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-1234567890rstuvwxyz0
Thunder instance	3
User Name	Online_Baseball_TH3
Password	Thunder@LMQ@3203

Parameter	Description
Resource_Name	vth-auto-scale-group
VMware Monitoring details	
vRLI IP	10.22.32.11
vROPs IP	10.22.32.12
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"
    } ]
}

```

Multiple instances

```

{
    "autoscale" : 0,
    "provider" : "XXXX",
    "thunders": [
        {
            "ip": "10.22.32.01",
            "username": "Online_Baseball_TH",
            "password": "Thunder@LMQ@3201",
            "resource_id": "i-1234567890lmnopq0"
        },
        {
            "ip": "10.22.32.02",
            "username": "Online_Baseball_TH2",
            "password": "Thunder@LMQ@3202",
            "resource_id": "i-1234567890rstuvwxyz0"
        }
    ]
}

```

- 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](#).

What's New

1.0.0

This release has the following enhancements:

- 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 autoscale Thunder instances can be configured for TOA.
- 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 Metric](#).
- TOA also 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.

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**.

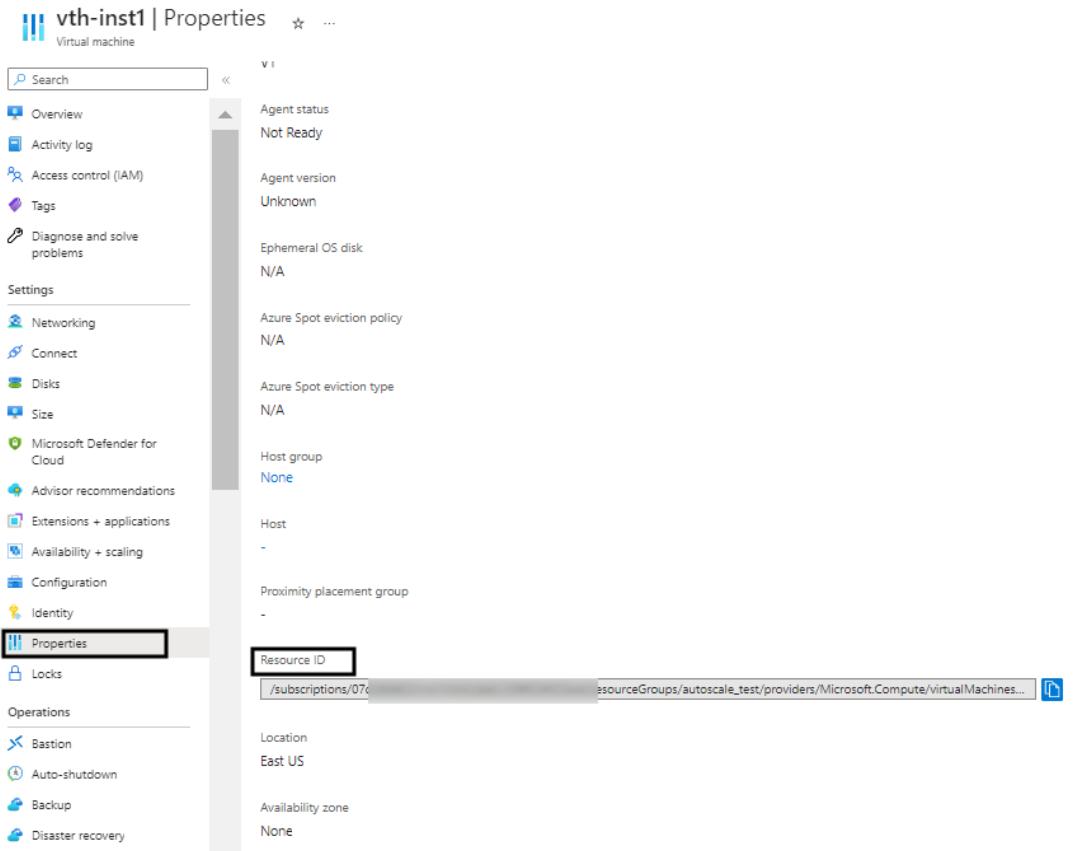
Figure 51 : Thunder instance Resource ID

The screenshot shows the AWS Management Console interface for the EC2 service. On the left, there's a navigation sidebar with links like EC2 Dashboard, EC2 Global View, Events, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Scheduled Instances. The main content area is titled 'Instances (1/14)'. It features a table with columns for Name, Instance ID, Instance state, and Instance type. A row for 'vth-inst1' is selected, highlighted with a blue border. Below the table, a section titled 'Instance: i-0a718290e6cf3499c (vth-inst1)' shows the 'Details' tab selected. Under the 'Details' tab, there's a 'Instance summary' section with a table showing the 'Instance ID' field containing '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 52 : Thunder instance Resource ID

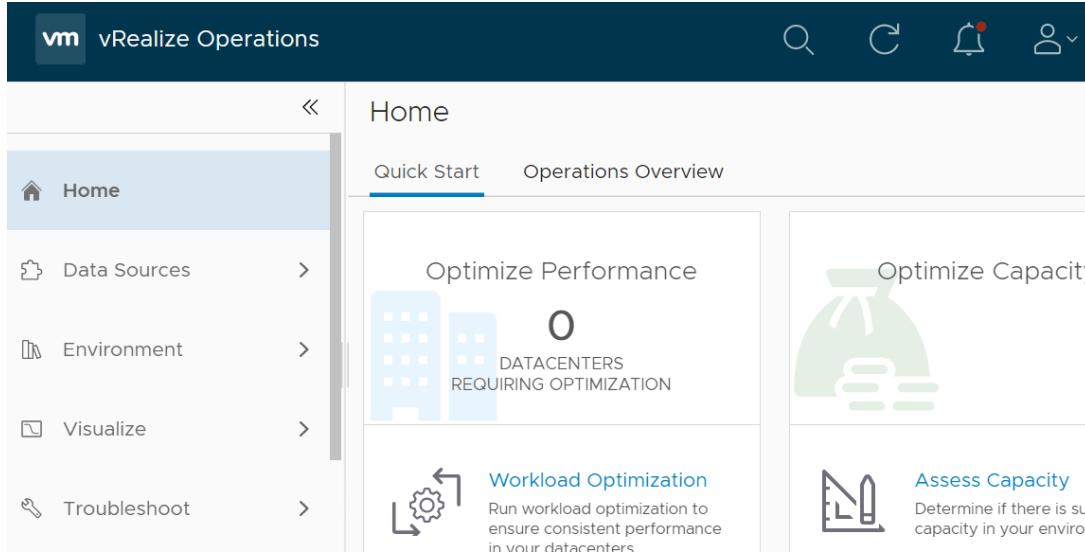


VMware

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.

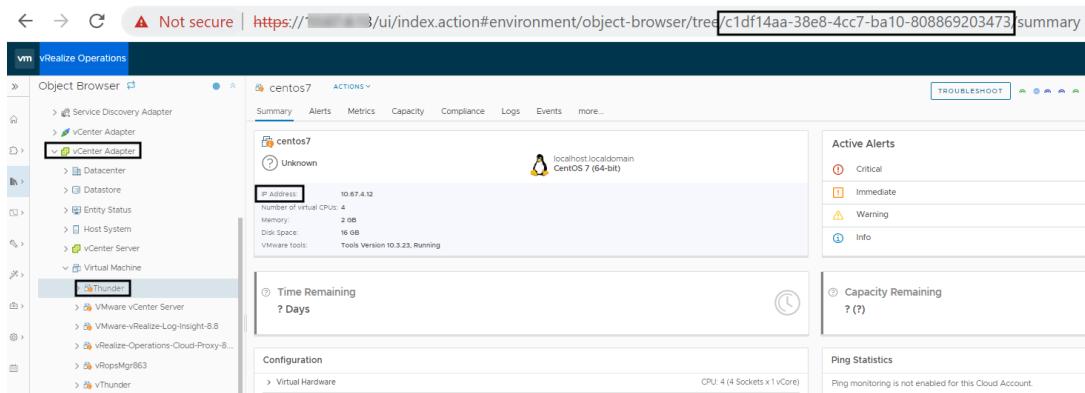
The vRealize Operations Home page is displayed.

Figure 53 : vRealize Operations - Home page



2. Go to **Home > Environment > Object Browser > All Objects > vCenter Adapter > Virtual Machine** and click **Thunder**.

Figure 54 : vRealize Operations - Virtual machine window



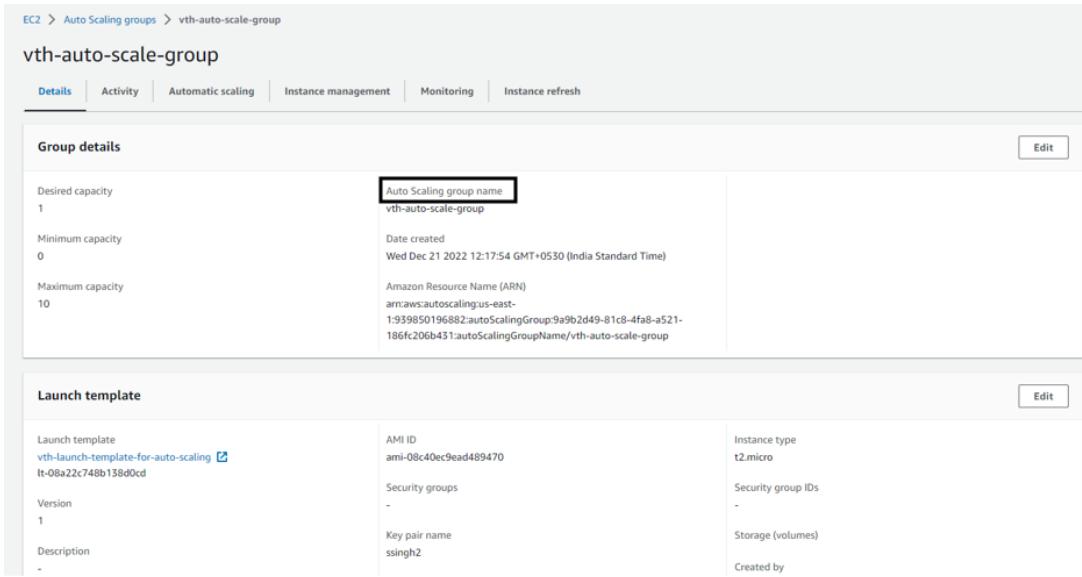
3. Get the resource ID from the URL.

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 autoscale group instance.
2. From the **Details** tab, get the **Auto Scaling group name**.

Figure 55 : Thunder Auto Scaling instance Resource ID



The screenshot shows the AWS EC2 Auto Scaling Groups page. The navigation bar at the top says "EC2 > Auto Scaling groups > vth-auto-scale-group". Below the navigation, there's a tab bar with "Details" selected, followed by "Activity", "Automatic scaling", "Instance management", "Monitoring", and "Instance refresh".

Group details

Desired capacity	1	Auto Scaling group name	vth-auto-scale-group
Minimum capacity	0	Date created	Wed Dec 21 2022 12:17:54 GMT+0530 (India Standard Time)
Maximum capacity	10	Amazon Resource Name (ARN)	arn:aws:autoscaling:us-east-1:939850196882:autoScalingGroup:9a9b2d49-81c8-4fa8-a521-186fc206b431:autoScalingGroupName/vth-auto-scale-group

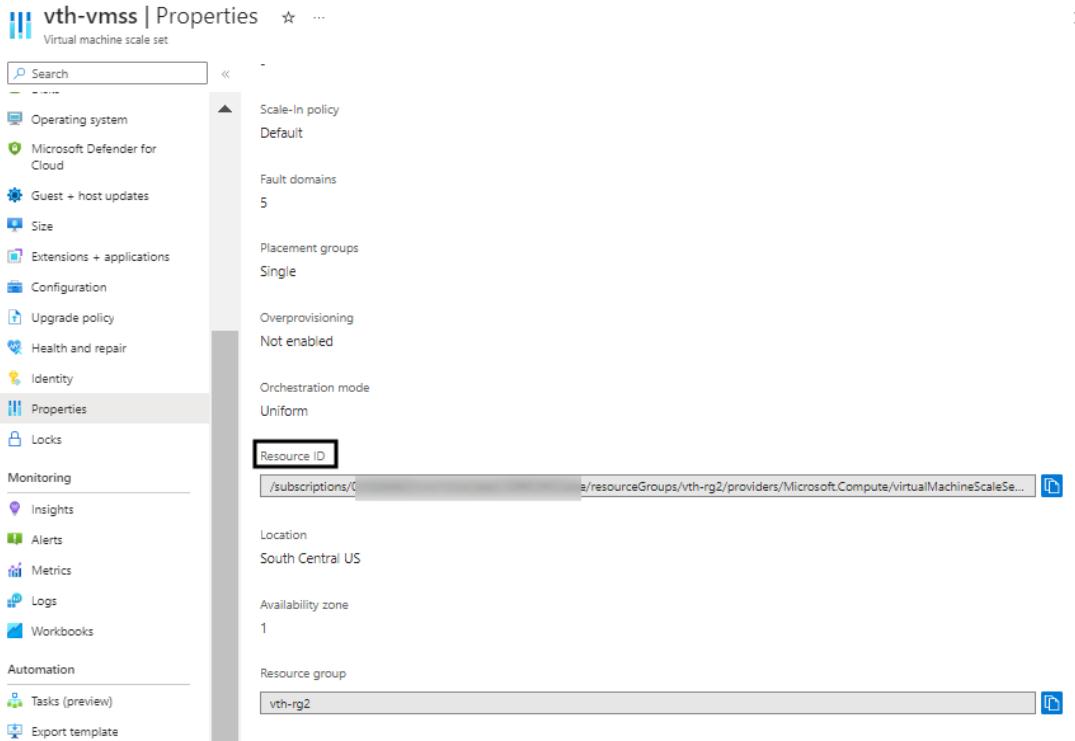
Launch template

Launch template	vth-launch-template-for-auto-scaling	AMI ID	ami-08c40ec9ead489470	Instance type	t2.micro
Version	1	Security groups	-	Security group IDs	-
Description	*	Key pair name	ssingh2	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 56 : Thunder VMSS instance Resource ID



Install Python

Depending on your operation system, install Python, Crontab, and Syslog.

CentOS

To install Python on CentOS, perform the following steps:

```
yum update -y  
reboot  
yum groupinstall "Development Tools" -y  
yum install openssl-devel libffi-devel bzip2-devel -y  
yum install wget -y  
wget https://www.python.org/ftp/python/3.10.0/Python-3.10.0.tgz  
tar xvf Python-3.10.0.tgz  
cd Python-3.10.0  
../configure --enable-optimizations  
make altinstall  
  
yum install cronie  
yum install rsyslog
```

Linux/Ubuntu

To install Python on Linux or Ubuntu, perform the following steps:

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

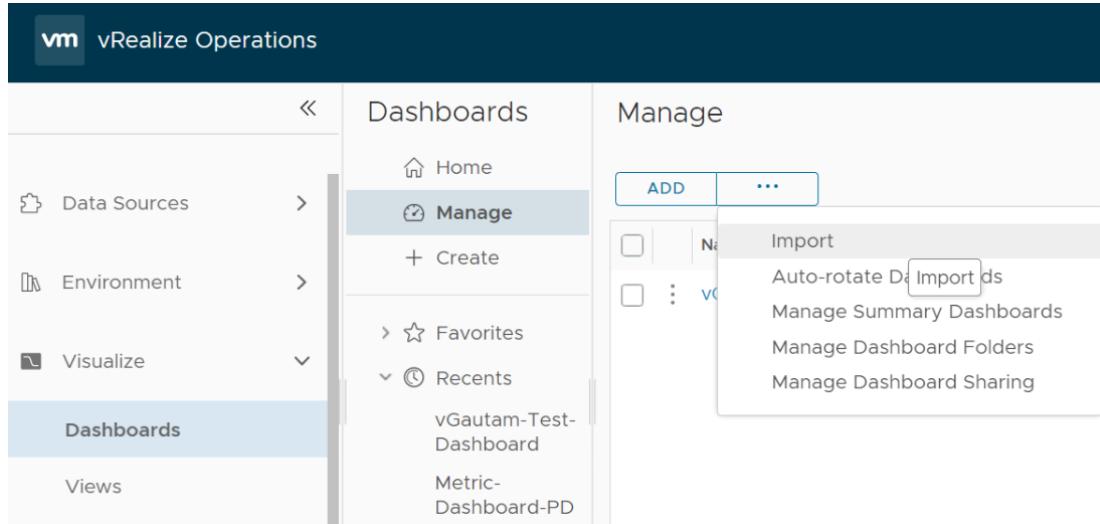
To import a dashboard using the JSON file, perform the following steps:

1. Download and open the [ThunderMetricDashboard.json](#) file.
2. Edit the following parameter values in the JSON file:
 - id
 - name
3. Save the changes in the JSON file.

- From the vRealize Operations Web UI, go to Home > Visualize > Dashboards and click Manage.

The **Manage** window is displayed.

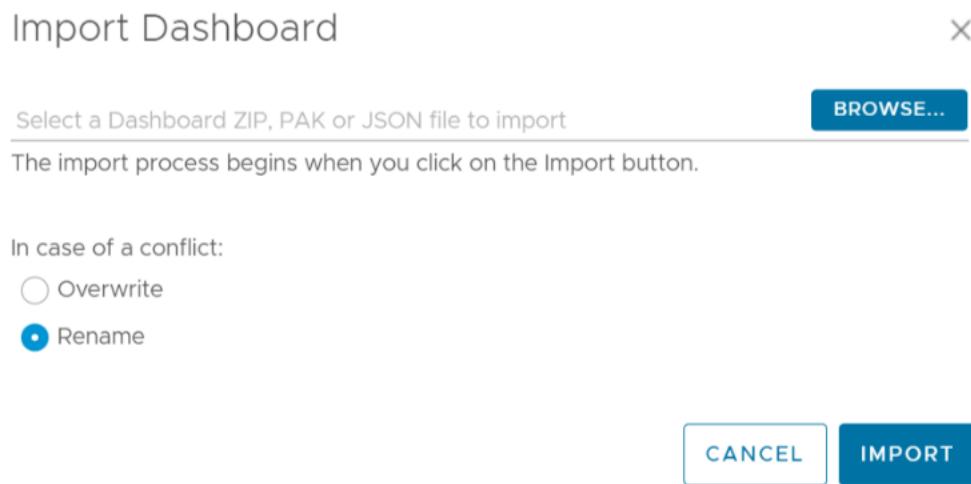
Figure 57 : Manage window



- Click ... > Import in the **Manage** panel.

The **Import Dashboard** window is displayed.

Figure 58 : Import Dashboard window



- Browse and select the **ThunderMetricDashboard.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:

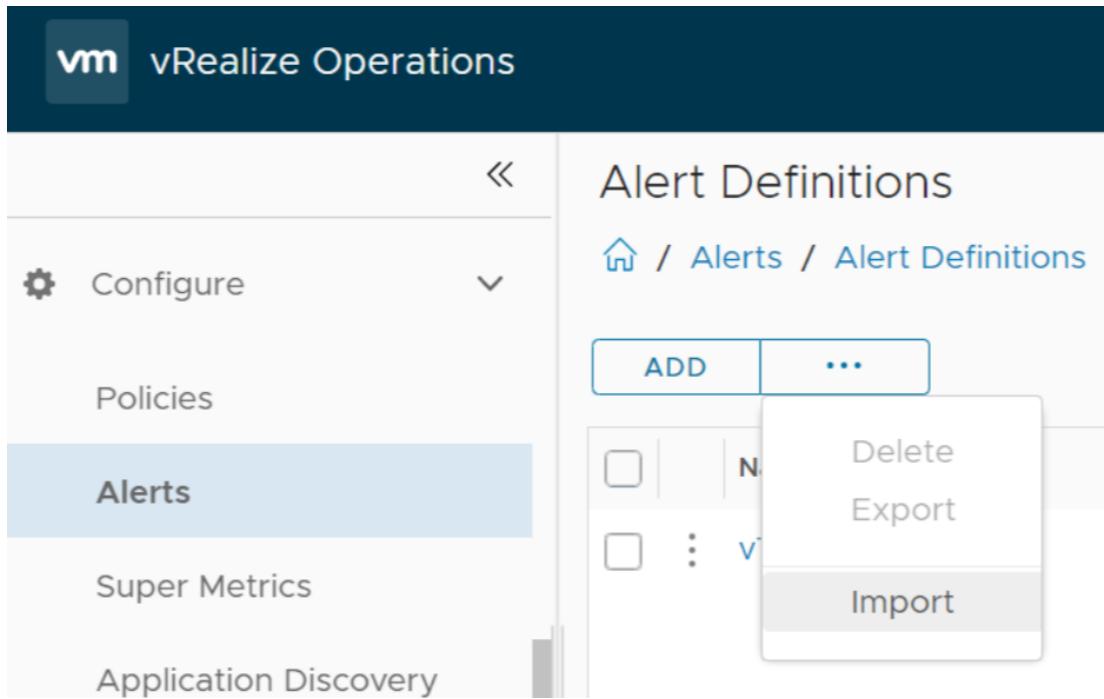
1. Download and open the [ThunderAlertDefinition.xml](#) file.
2. Enter the following parameter values in the XML file as appropriate:
 - id
 - name

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.

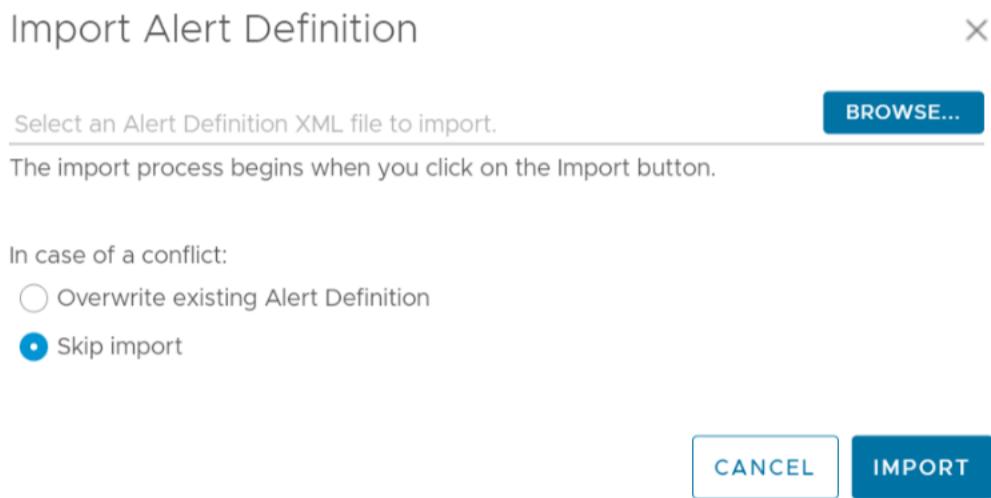
Figure 59 : Alert Definitions window



5. Click ... > Import in the Alert Definition window.

The Import Alert Definition window is displayed.

Figure 60 : Import Alert Definition window



6. Browse and select the **ThunderAlertDefinition.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 [ThunderNotification.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 [ThunderAlertDefinition.json](#).

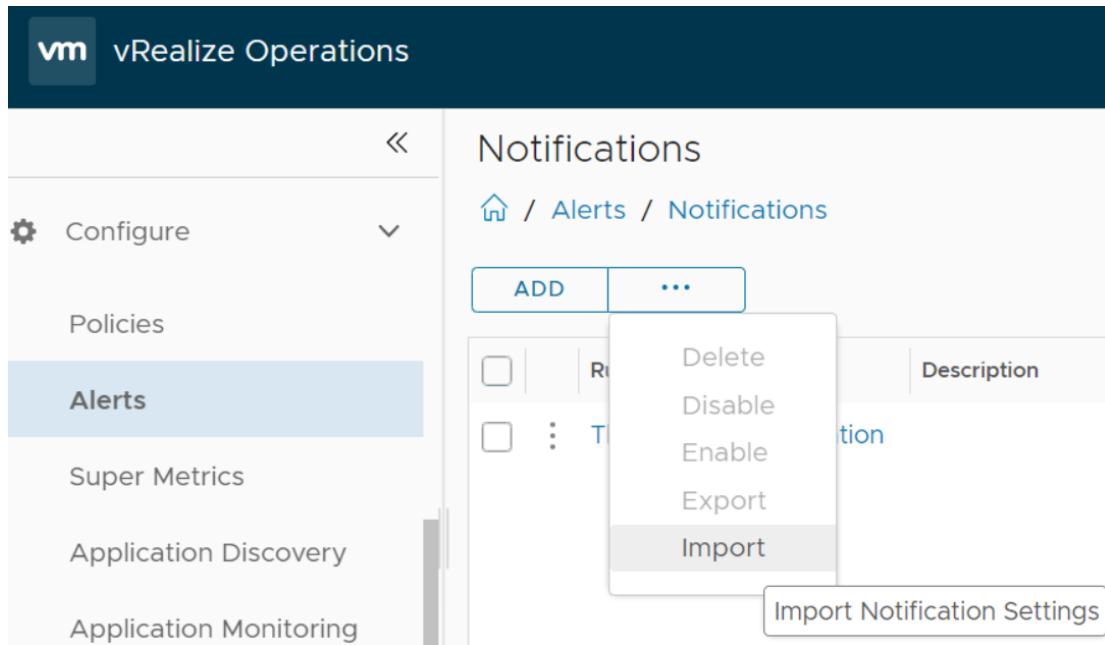
3. Update the sender and recipient email address values in the following parameter:

```
"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.

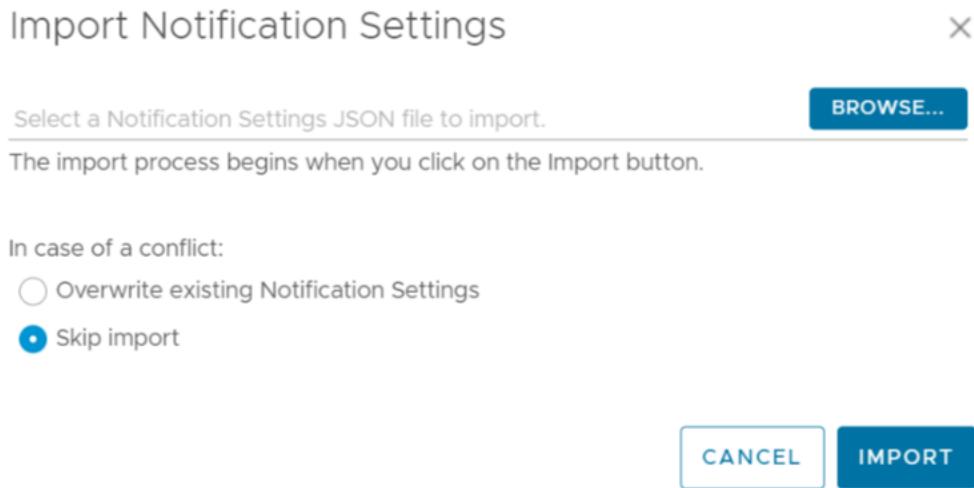
Figure 61 : Notifications window



6. Click ... > Import in the **Notifications** panel.

The **Import Notification Settings** window is displayed.

Figure 62 : Import Notification Settings window



7. Browse and select the **ThunderNotification.json**.

8. Click **Import**.

The new notification is imported and listed in the **Notifications** window.

