Installing Thunder Observability Agent 1.0.0

May, 2023

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

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 Application Delivery Controller (ADC):

- Collects, processes, and publishes fourteen 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 Thunder Metric Support.
- 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 autoscale 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.

Support Information

The following topics are covered:

- Technology Support
- Thunder Metric Support
- Thunder Log Support
- ACOS Support
- Cloud Support
- Performance Support

Technology Support

TOA supports the following key technologies:

Table 1 : Key Technologies in TOA

Name	Version	License
Python	3.10	PSF License Python 3.3 license | Python.org
Requests	2.27.1	Apache Software License 2.0 Apache 2.0 (amazon.com)
Boto3	1.24.25	

Thunder Metric Support

TOA provides the following metrics:

Table 2 : Available Thunder Metrics

Metric (Since 1.0.0)	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	Memory (RAM) usage, in percentage, of a Thunder

Table 2 : Available Thunder Metrics

Metric (Since 1.0.0)	Description
Percentage	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 (Per 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 (Per Sec)	Count of the total L7 transactions made per second from Thunder to the server for the last data collection cycle.
Server Down Count	Count of the total web or app servers configured in the Thunder instance that are not reachable from Thunder for the last data collection cycle.
Server Down Percentage	Percentage of the total web or app servers configured in the Thunder instance that are not reachable from Thunder for the last data collection cycle.
SSL Errors Count	Count of the total errors that occurred during data transmission from Thunder to the Server due to SSL connection, negotiate, encrypt, and decrypt for the last data collection cycle.
Server Errors Count	Count of the total errors that occurred during data transmission from Thunder to the server with status code 40x 50x for that last data collection cycle.

Table 2 : Available Thunder Metrics

Metric (Since 1.0.0)	Description
Total Session Count	Count of the total active sessions of the Thunder instance for the last data collection cycle.
Packet Rate (per seconds)	Count of the total packets sent from or received at the Thunder instance for the last collection cycle. **NOTE:** Applicable for ACOS 6.0.0 and higher
Packet Drop Rate (per seconds)	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 6.0.0 and higher

Thunder Log Support

TOA supports the following logs:

Table 3 : Available Thunder Logs

Logs	Description
SysLogs	Thunder internal logs such as: • SSL connection, negotiate, encrypt, and decrypt • Status codes 40x 50x

ACOS Support

TOA supports the following ACOS versions:

Table 4 : Support Matrix

ACOS Version	TOA Version
64-bit Advanced Core OS (ACOS) version 6.0.0-p2, build 22	1.0.0

Table 4 : Support Matrix

ACOS Version	TOA Version
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-p6, build 74	1.0.0
64-bit Advanced Core OS (ACOS) version 5.2.1-p5, build 114	1.0.0

Cloud Support

TOA supports the following Cloud platforms and monitoring applications:

Table 5 : Supported Cloud platforms and their monitoring applications

Cloud Platform	Monitoring Applications
AWS	• CloudWatch
Azure	• Application Insights • Log Analytics Workspace
VMware ESXi	• vRealize Operations Manager (vROps) • vRealize Log Insight (vRLI)

Performance Support

TOA supports the following performance metrics:

Table 6 : Performance Support

Platform	Number of Thunder Instances	Time (in seconds)
AWS	Up to 10	Approximately 10-15 seconds

Table 6 : Performance Support

Platform	Number of Thunder Instances	Time (in seconds)
Azure	Up to 10	Approximately 10-15 seconds
VMware	Up to 5	Approximately 1-5 seconds

NOTE: Performance depends on the hardware, operating system, Thunder deployment topology, target monitoring service, network settings, and network accelerated settings.

The current benchmarking is in consideration with same region, same availability zone, and same domain deployment and network settings.

Install TOA

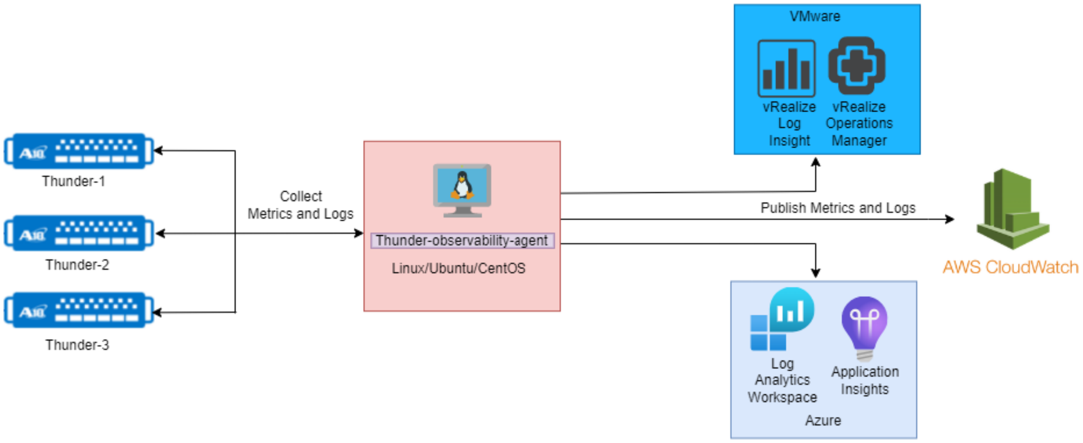
TOA is a standalone software that can be installed on any orchestration platform. The following installation options are available:

- 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 1 illustrates the installation of TOA in the Python plugin architecture.

  Figure 1 : Python Plugin Installation Architecture

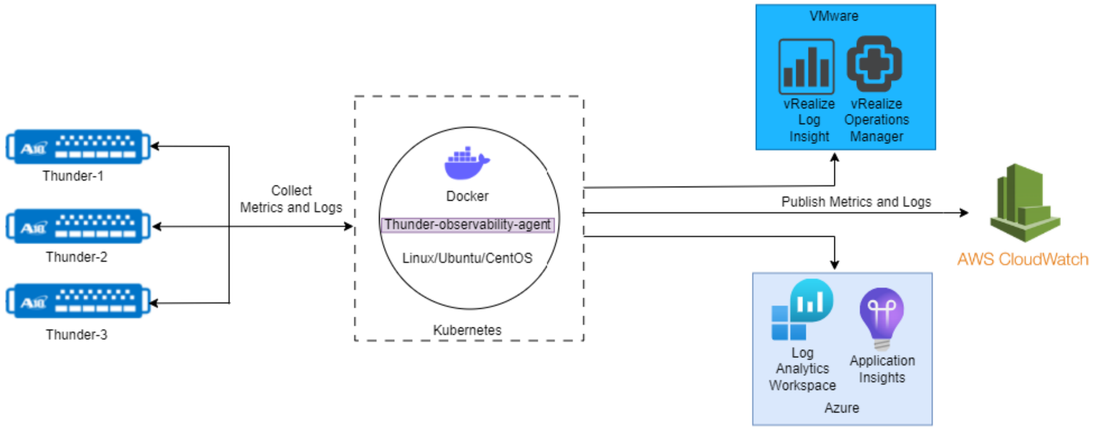


- Containerized Installation

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

  Figure 2 illustrates the installation of TOA in a containerized architecture.

Figure 2 : 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 7 : Hardware Dependencies

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

Table 7 : 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 8 : Software Dependencies

Requirement	Description
Operating System	• 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

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

Offline

Download the executable from TOA 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 9 : Cloud specific Configuration Files

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

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.

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 10 : Cloud specific Configuration Files

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

Table 10 : Cloud specific Configuration Files

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

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.

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

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 11 : 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 11 : File Parameters

Parameter	Description	Default Value
	Start Time = Last data collection time End Time = Current data collection time - *<log_collection_delay_min>* **Example** If the current data collection time is 10:00:00 AM and the last data collection time is 09:59:00 AM, then: the Start Time is 9:59:00 AM. the End Time is 10:00:00 AM (which is 10:00:00 AM - 0 minutes). So, TOA collects all the logs generated by Thunder from 9:59:00 AM to 10:00:00 AM.	
`cron_job_frequency_min`	Specifies the cron job frequency in minutes. This parameter should match with the `crontab -e` job definition. The system considers `crontab -e` for job scheduling. If the frequency is changed in this parameter, it should also change in the `crontab` file. For more information, see Crontab.	`1`
`http_ssl_verify`	Disables SSL certificate verification over HTTPS. If a user wants to enable SSL: • For CA signed certificate configured in Thunder, set the parameter to **True**.	`False`

Table 11 : File Parameters

Parameter	Description	Default Value
	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**. **Example** `/usr/toaenv/thunder-observability-agent/toa.pem` **NOTE:** If vROps and vRLI has self-signed certificates, then their public certificates must be imported in *.pem file.	
`http_connection_timeout_sec`	Specifies the maximum amount of time, in seconds, that the TOA waits to set up an HTTP connection to communicate with any Thunder instance.	`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`	Specifies the configuration file path to collect data from Thunder.	`/root/.thunder/credentials`
`aws_credentials_path`	Specifies the AWS credentials file path to establish connection.	`/root/.aws/credentials`
`aws_config_path`	Specifies the AWS configuration file path to publish the data.	`/root/.aws/config`

Table 11 : File Parameters

Parameter	Description	Default Value
azure_credentials_path	Specifies the Azure credentials file path to establish the connection and publish the data.	/root/.azure/credentials
vmware_credentials_path	Specifies the VMware credentials file path to establish the connection and publish the data.	/root/.vmware/credentials

Logging

This file lists the TOA logging configurations.

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

```
[loggers]
keys=root

[handlers]
keys=hand01

[formatters]
keys=form01

[logger_root]
level=INFO
handlers=hand01


[handler_hand01]
class=logging.handlers.RotatingFileHandler

# ERROR, INFO
level=INFO
formatter=form01

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

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

Crontab

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

To 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:

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

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

Cloud-specific Configuration

The following information is required to setup the cloud-specific configuration:

- AWS Config
- AWS Credentials

- Azure Credentials
- VMware Credentials

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 12 : 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 13 : 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`

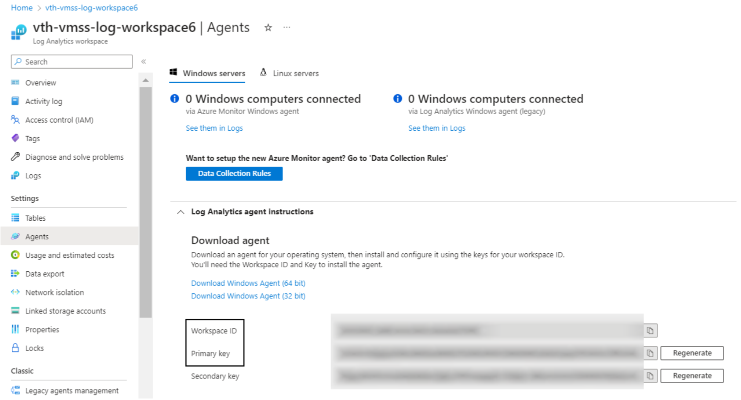
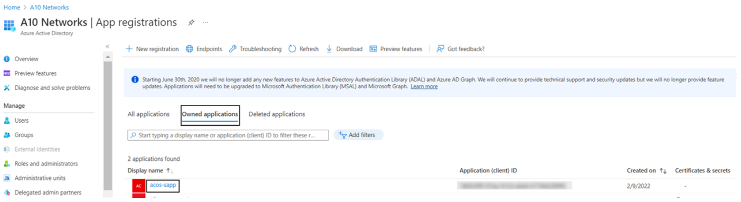
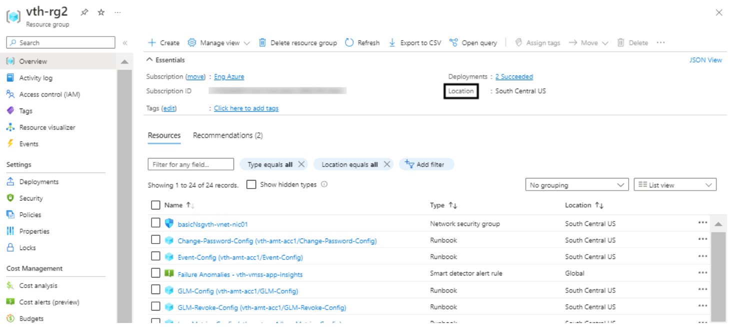
Update the following parameters according to your Azure setup:

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

Table 14 : 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 3 : Agents window

Table 14 : Azure Credentials File Parameters

Parameter	Description
	 Home > vth-vmss-log-workspace6 vth-vmss-log-workspace6 | Agents Log Analytics workspace Search Overview Activity log Access control (IAM) Tags Diagnose and solve problems Logs Settings Tables Agents Usage and estimated costs Data export Network isolation Linked storage accounts Properties Locks Classic Legacy agents management Windows servers Linux servers 0 Windows computers connected via Azure Monitor Windows agent See them in Logs 0 Windows computers connected via Log Analytics Windows agent (legacy) See them in Logs Want to setup the new Azure Monitor agent? Go to 'Data Collection Rules' Log Analytics agent instructions Download agent Download an agent for your operating system, then install and configure it using the keys for your workspace ID. You'll need the Workspace ID and Key to install the agent. Download Windows Agent (64 bit) Download Windows Agent (32 bit) Workspace ID [redacted] Primary key [redacted] Secondary key [redacted]
`azure_client_id`	To get the client ID, secret ID, and tenant ID, go to **Azure Portal** > **Azure Services** > **Azure Active Directory** > **App Registration** > **Owned applications** > <*application_name*>. Figure 4 : Azure active directory - App registrations window  Home > A10 Networks A10 Networks | App registrations Azure Active Directory New registration Endpoints Troubleshooting Refresh Download Preview features Got feedback? Overview Preview features Diagnose and solve problems Manage Users Groups External Identities Roles and administrators Administrative units Delegated admin partners Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more All applications Owned applications Deleted applications Start typing a display name or application (client) ID to filter these r... Add filters 2 applications found Display name | Application (client) ID | Created on | Certificates & secrets AC acos-sapp | [redacted] | 2/9/2022 | -
`azure_secret_id`	
`azure_tenant_id`	
`azure_location`	To get the location, go to **Azure Portal** > **Azure Services** > **Resource Groups** > <*your_resource_group*> > **Overview** > **Essentials** > **Location**. Figure 5 : Resource Group window  vth-rg2 Resource group Search Overview Activity log Access control (IAM) Tags Resource visualizer Events Settings Deployments Security Policies Properties Locks Cost Management Cost analysis Cost alerts (preview) Budgets Create Manage view Delete resource group Refresh Export to CSV Open query Assign tags Move Delete JSON View Essentials Subscription (move) : Eng Azure Subscription ID [redacted] Tags (edit) : Click here to add tags Deployments : 2 Succeeded Location : South Central US Resources Recommendations (2) Filter for any field... Type equals all Location equals all Add filter No grouping List view Showing 1 to 24 of 24 records. □ Show hidden types Name | Type | Location basicNsgvth-vnet-nic01 | Network security group | South Central US Change-Password-Config (vth-amt-acc1/Change-Password-Config) | Runbook | South Central US Event-Config (vth-amt-acc1/Event-Config) | Runbook | South Central US Failure Anomalies - vth-vmss-app-insights | Smart detector alert rule | Global GLM-Config (vth-amt-acc1/GLM-Config) | Runbook | South Central US GLM-Revoke-Config (vth-amt-acc1/GLM-Revoke-Config) | Runbook | South Central US

For sample configuration, see Examples.

VMware Credentials

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

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

Update the following parameters according to your VMware setup:

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

Table 15 : 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 autoscaled 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 Auto Scaling Group or VMSS

Thunder Credentials

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

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

Depending on your environment, 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

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "XXXX",
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX"
       }]
}
```

Multiple Thunder Instances

```
{
      "autoscale" : 0,
      "provider" : "XXXX",
      "thunders": [{
             "ip": "XXXX",
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX"
       },
               {
             "ip": "XXXX",
             "username": "XXXX",
             "password": "XXXX",
             "resource_id": "XXXX"
       }]
}
```

Thunder Instances in Auto Scaling Group or VMSS

AWS

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

Azure

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

Table 16 : Thunder Credentials File Parameters

Parameter	Description
`autoscale`	Specify 1 if the Thunder instance is inside 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 inside AWS autoscale group or Azure virtual machine scale set (`autoscale=1`). The following options are available: • `aws` • `azure`
`thunders`	Specifies the Thunder instance details. The following parameters are available: • `ip` • `username` • `password`

Table 16 : Thunder Credentials File Parameters

Parameter	Description
	• `resource_id`
`ip`	Specifies the Thunder instance IP address.
`username`	Specifies the Thunder instance username.
`password`	Specifies the Thunder instance password.
`resource_id`	Specifies the compute instance resource IDS on which Thunder is deployed. For more information, see Get Resource ID.

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 17 : 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 are enabled, TOA sends this metric to the AWS CloudWatch. By default, it is enabled.	`1`
`aws_memory`	Specify 1 to publish the deployed Thunder instances' memory usage	`1`

Table 17 : AWS Configuration Parameters

Parameter	Description	Default Value
	(percentage) on the AWS CloudWatch. By default, it is enabled.	
`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_count`	Specify 1 to publish the deployed Thunder instances' server down count on the AWS CloudWatch. By default, it is enabled.	`1`
`aws_server_down_percentage`	Specify 1 to publish the deployed Thunder instances' configured web/app	`1`

Table 17 : AWS Configuration Parameters

Parameter	Description	Default Value
	servers down percentage on the AWS CloudWatch. By default, it is enabled.	
`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. By default, it is enabled.	1

Azure

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

Table 18 : Azure Configuration Parameters

Parameter	Description	Default Value
`azure_provider`	Specify 1 to publish selected metric/s, logs, or both to Azure. 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.	0
`azure_metric`	Specify 1 to send metrics to Azure Application Insights. It sends the data only if `azure_provider` is also enabled. By default, it is disabled.	0
`azure_metric_resource_id`	Specifies the Azure Application Insights resource ID. To get this value, go to **Azure Portal** > **Azure Services** > **Application Insights** > **Thunder** > **Properties** > **Resource ID**. **Example** `/subscriptions/07dxxxxxxxxxxx/resourceGroups/<resource_group_name>/providers/microsoft.insights/components/<app-insight-name>`	*<azure_metric_resource_id>*
`azure_cpu`	Specify 1 to publish the deployed Thunder instances' average data CPU usage (percentage) on the Azure Application Insights as the CPU Usage Percentage (Data) metrics. If the `azure_provider` and `azure_metrics` parameters are enabled, TOA sends this metric to the Azure Application Insights. By default, it is enabled.	1

Table 18 : Azure Configuration Parameters

Parameter	Description	Default Value
`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
`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.	1

Table 18 : Azure Configuration Parameters

Parameter	Description	Default Value
	By default, it is enabled.	
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. By default, it is enabled.	1
azure_packet_drop	Specify 1 to publish the deployed Thunder instances' packet drop count on the Azure Application Insights. By default, it is enabled.	1

VMware

Configure the following parameters in the `config.json` to publish Thunder metrics to the VMware vROps.

Table 19 : VMware Configuration Parameters

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

Table 19 : VMware Configuration Parameters

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

Table 19 : VMware Configuration Parameters

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

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 20 : AWS Configuration Parameters

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

Azure

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

Table 21 : Azure Configuration Parameters

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

Table 21 : Azure Configuration Parameters

Parameter	Description	Default Value
	enabled. By default, it is disabled.	
`azure_log_workspace_id`	Specifies the Azure Log Analytics Workspace ID. To get this value, go to **Azure Portal** > **Azure Services** > **Log Analytics workspaces** > *<your_log_analytics_workspace>* > **Settings** > **Agents**.	*`<azure_log_workspace_id>`*

VMware

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

Table 22 : VMware Configuration Parameters

Parameter	Description	Default Value
`vmware_provider`	Specify 1 to publish the selected metric/s, logs, or both to VMware. By default, it is disabled and does not send logs to VMware. To publish logs, it is mandatory to enable VMware as a provider.	`0`
`vmware_log`	Specify 1 to publish the logs to VMware vRLI. It sends the data only if `vmware_provider` is also enabled. By default, it is disabled.	`0`
`vmware_vrli_host`	Specifies the VMware vRLI host IP address. To get the host, go to **ESXi** host > **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 your system performance and monitor the health, throughput, and performance of your virtual machines (VM), hypervisor VMs, on-premise, or cloud applications.

The following topics are covered:

Monitor Metrics

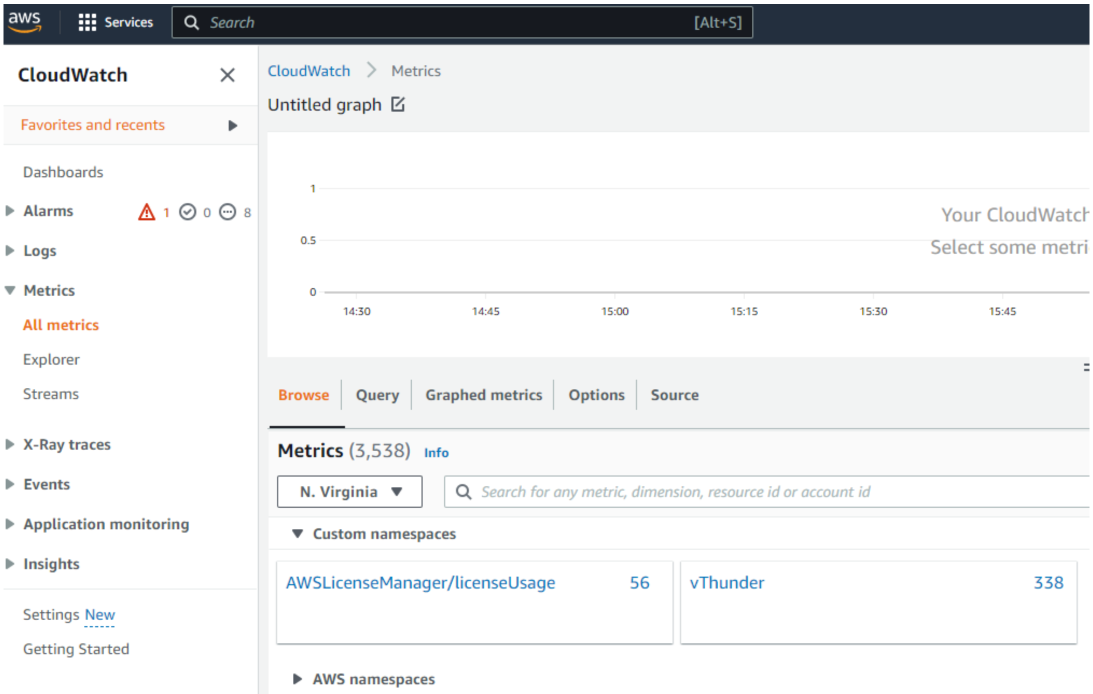
Depending on your cloud provider, the steps are provided to monitor the configured metrics.

AWS CloudWatch

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

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

Figure 6 : AWS All metrics



Metrics (3,538)

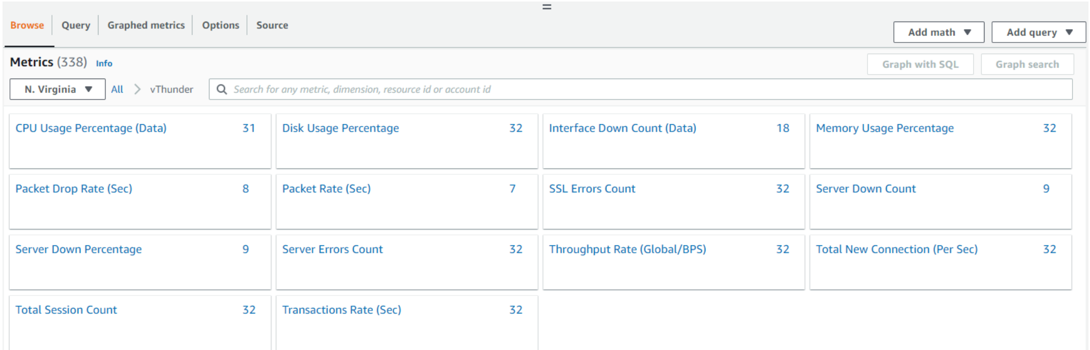
Custom namespaces

AWSLicenseManager/licenseUsage	56
vThunder	338

AWS namespaces

2. Select **Browse** > **vThunder** metrics namespace.

Figure 7 : vThunder metrics



Metrics (338)

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

3. Select the required metrics to be monitored.

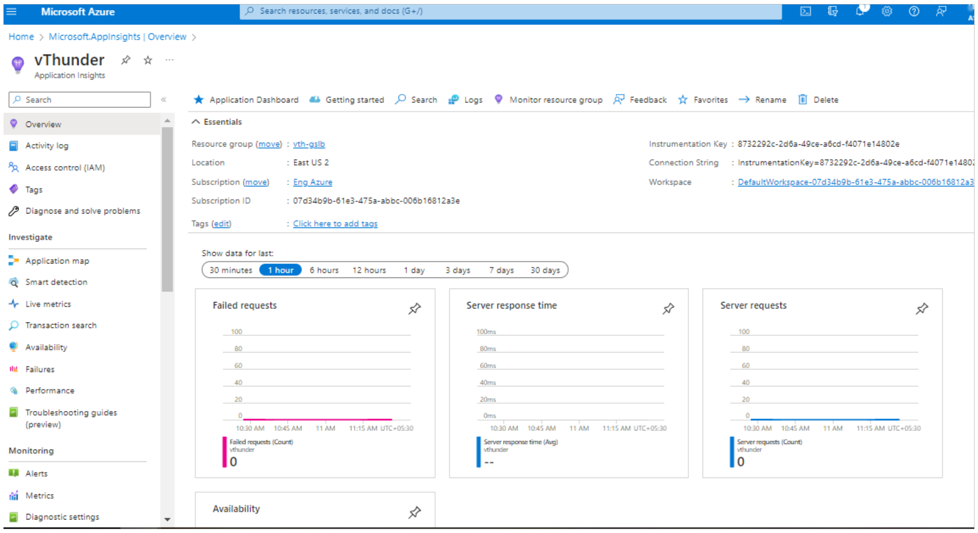
Azure Application Insight

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

1. From the **Azure Portal**, go to **Azure Services** > **Application Insights** > **vThunder**.

   The vThunder - Overview window is displayed.

   Figure 8 : vThunder - Overview window

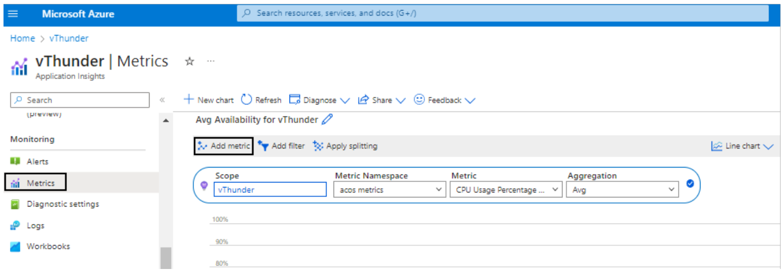


Resource group (move) : vth-gslb
Location : East US 2
Subscription (move) : Eng Azure
Subscription ID : 07d34b9b-61e3-475a-abbc-006b16812a3e
Tags (edit) : Click here to add tags
Instrumentation Key : 8732292c-2d6a-49ce-a6cd-f4071e14802e
Connection String : InstrumentationKey=8732292c-2d6a-49ce-a6cd-f4071e1480
Workspace : DefaultWorkspace-07d34b9b-61e3-475a-abbc-006b16812a3

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

3. Under **Chart Title**, click **Add metric**.

   Figure 9 : vThunder - Metrics window



Avg Availability for vThunder

Scope: vThunder
Metric Namespace: acos metrics
Metric: CPU Usage Percentage ...
Aggregation: Avg

4. Select the appropriate values in the following fields:

Table 23 : vThunder - Metrics

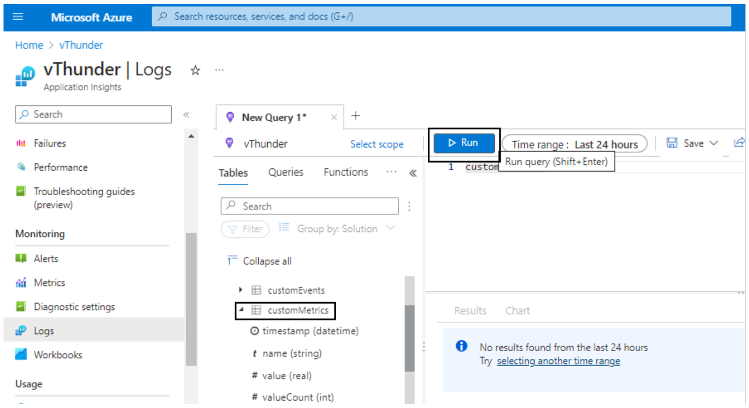
Field Name	Description
Scope	Enter **vThunder**.
Metrics Namespace	Select **acos metrics**.

Table 23 : vThunder - Metrics

Field Name	**Description**
Metrics	Select a metric from the drop-down.

5. View different metrics.
6. Click **Logs** from the left **Monitoring** panel.
   Close the **Queries** pop-up window.
7. From **New Query1** > **Tables** tab, double-click **customMetrics**.

   Figure 10 : vThunder - Logs window

8. Click **Run**.

   All logs are displayed in tabular format with expandable details.

VMware vROPs

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

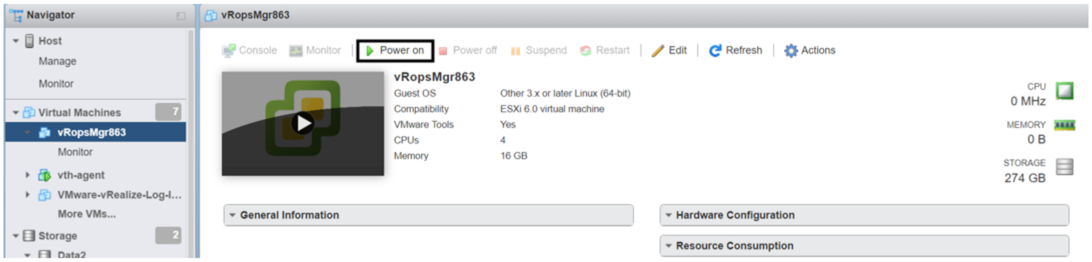
1. Start vROps VM
2. Create a Dashboard
3. Create a Notification
4. Create an Alert
5. View VMware Metrics

Start vROps VM

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

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

   Figure 11 : Start vROPs VM



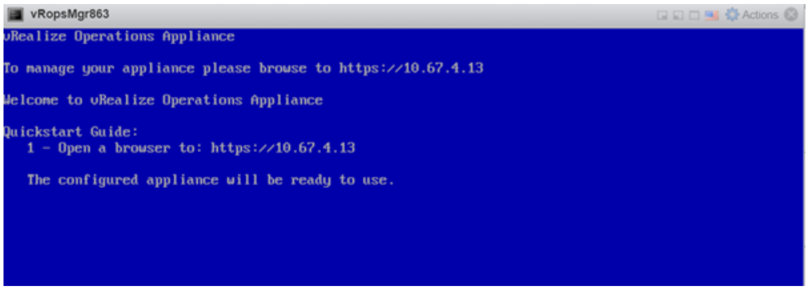
vRopsMgr863
Guest OS: Other 3.x or later Linux (64-bit)
Compatibility: ESXi 6.0 virtual machine
VMware Tools: Yes
CPUs: 4
Memory: 16 GB
CPU: 0 MHz
MEMORY: 0 B
STORAGE: 274 GB

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

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

   The vROPs virtual machine is powered on and reachable.

   Figure 12 : vRealize Operations Appliance



```
vRealize Operations Appliance

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

Welcome to vRealize Operations Appliance

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

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

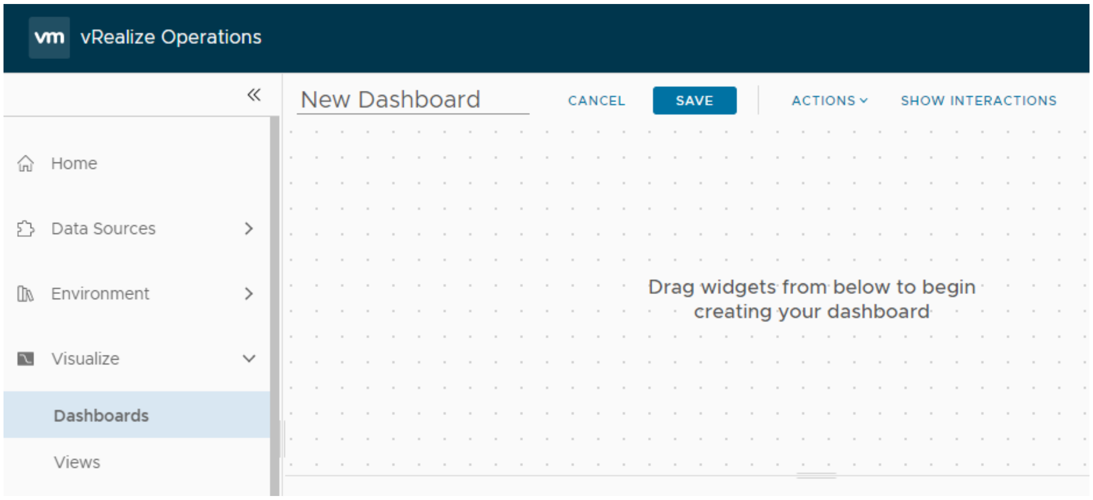
Create a Dashboard

To create a dashboard, 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 13 : New Dashboard window

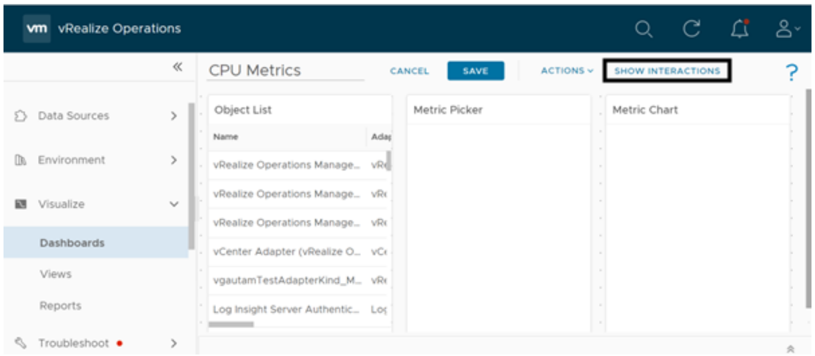


Drag widgets from below to begin creating your dashboard

2. Provide a name to the new dashboard and double-click or drag the following widgets:

- Object List
- Metric Picker
- Metric Chart

Figure 14 : Dashboard widgets

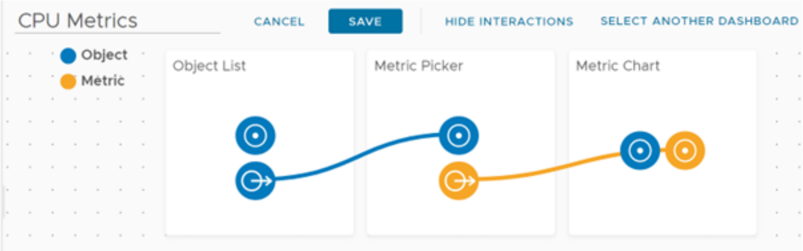


CPU Metrics

Name	Adap
vRealize Operations Manage...	vRe
vRealize Operations Manage...	vRe
vRealize Operations Manage...	vRe
vCenter Adapter (vRealize O...	vCe
vgautamTestAdapterKind_M...	vRe
Log Insight Server Authentic...	Log

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

Figure 15 : Interactions



4. Drag the connectors and create interactions as shown in the Figure 15.
5. Click **Save** to save the changes.

   A dashboard is created.

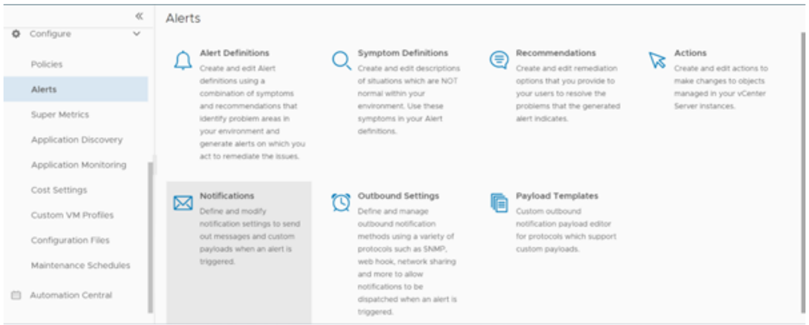
Create a Notification

To create a notification, perform the following steps:

1. From the **vRealize Operations Web UI**, go to **Home** > **Configure** and click **Alerts**.

   The **Alerts** window is displayed.

   Figure 16 : Alerts window



Alert Definitions
Create and edit Alert definitions using a combination of symptoms and recommendations that identify problem areas in your environment and generate alerts on which you act to remediate the issues.

Symptom Definitions
Create and edit descriptions of situations which are NOT normal within your environment. Use these symptoms in your Alert definitions.

Recommendations
Create and edit remediation options that you provide to your users to resolve the problems that the generated alert indicates.

Actions
Create and edit actions to make changes to objects managed in your vCenter Server instances.

Notifications
Define and modify notification settings to send out messages and custom payloads when an alert is triggered.

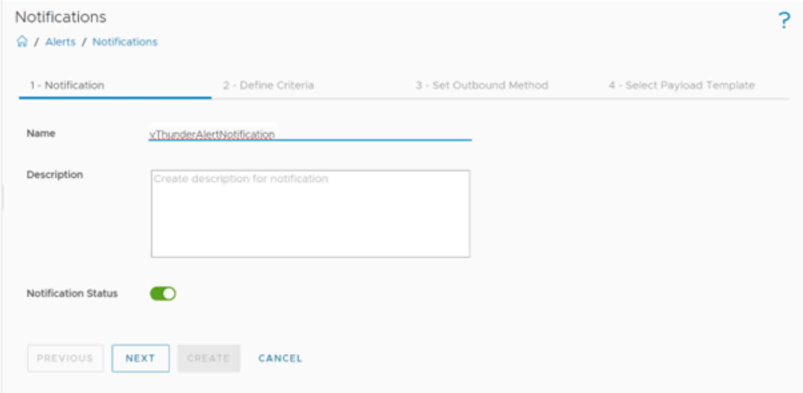
Outbound Settings
Define and manage outbound notification methods using a variety of protocols such as SNMP, web hook, network sharing and more to allow notifications to be dispatched when an alert is triggered.

Payload Templates
Custom outbound notification payload editor for protocols which support custom payloads.

2. Click **Notifications** to add a new notification.
3. Click **Add**.

   A panel with **Notification** tab is displayed.

Figure 17 : Notifications tab



Name: vThunderAlertNotification

Description: Create description for notification

Notification Status: enabled

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

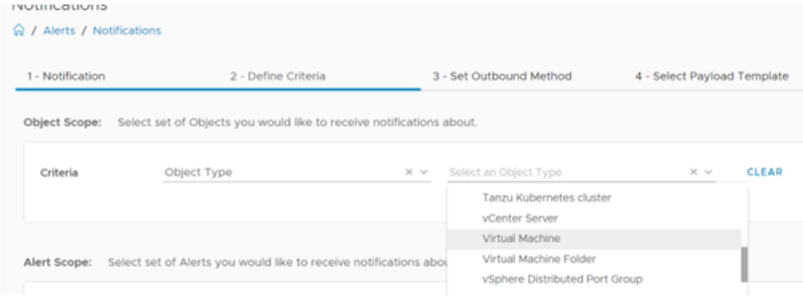
Table 24 : Notifications tab

Field Name	Description
Name	Enter the notification name. **Example** In the Figure 17, `vThunderAlertNotification` is the notification name.
Notification Status	Select **Enable**.

5. Click **Next**.

   The **Define Criteria** tab is displayed.

   Figure 18 : Define Criteria tab



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

Criteria: Object Type | Select an Object Type

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

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

A field appears to select the object type.

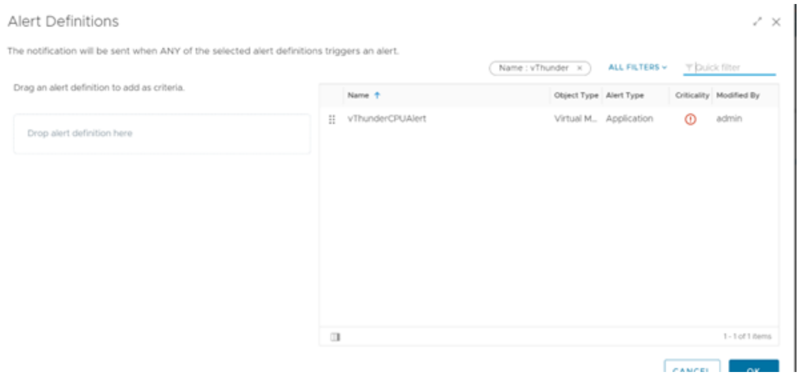
7. Select **vCenterAdapter** > **Virtual Machine** from the drop-down.

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

8. In the **Category** field, select **Alert Definition** from the drop-down.

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

   Figure 19 : Alert Definition pop-up



Alert Definitions

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

Drag an alert definition to add as criteria.

Drop alert definition here

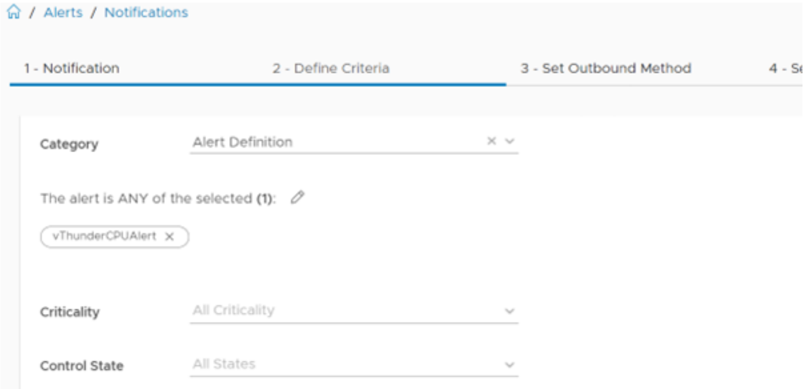
Name	Object Type	Alert Type	Criticality	Modified By
vThunderCPUAlert	Virtual M...	Application	!	admin

1 - 1 of 1 items

9. Search and select **vThunderCPUAlert**, and then click **OK**.

   The selected alert definition is listed under Category.

   Figure 20 : Selected alert definition



Category: Alert Definition

The alert is ANY of the selected (1):

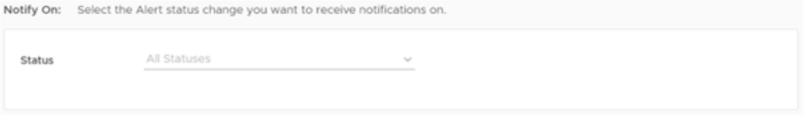
vThunderCPUAlert ×

Criticality: All Criticality

Control State: All States

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

Figure 21 : Notify On



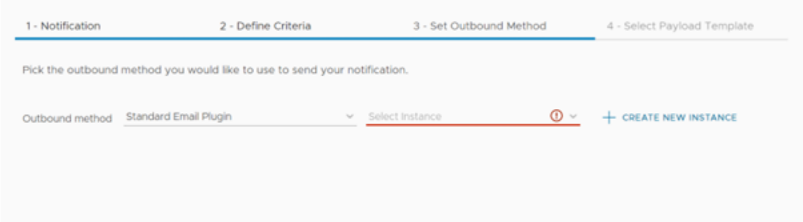
Notify On: Select the Alert status change you want to receive notifications on.

Status: All Statuses

11. Click **Next**.

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

    Figure 22 : Set Outbound Method tab

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

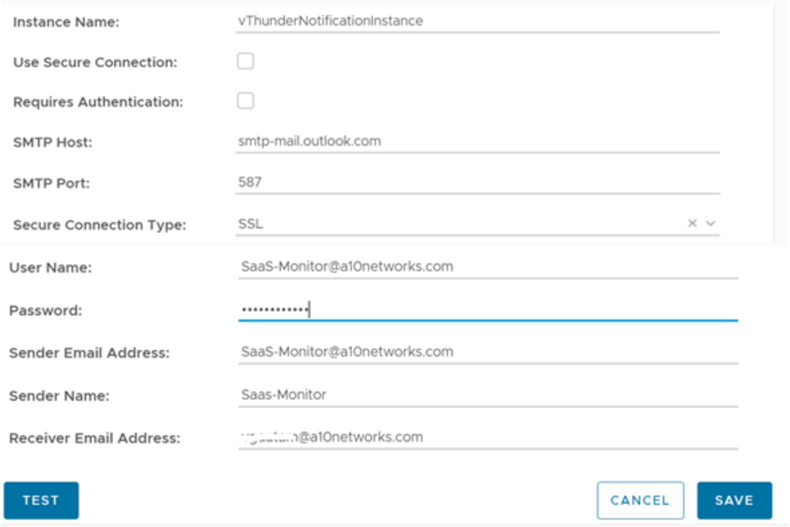
    Outbound method: Standard Email Plugin | Select Instance

12. In the **Outbound method** field, select **Standard Email Plugin** from the drop-down list.

13. Click **Create New Instance**.

    The fields required for creating a new instance are displayed.

    Figure 23 : Create New Instance fields

Instance Name:	vThunderNotificationInstance
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:	[redacted]n@a10networks.com

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

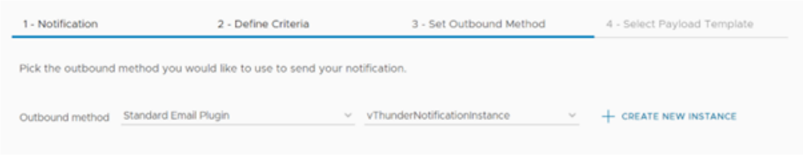
Table 25 : Create New Instance

Field Name	**Description**
Instance Name	Enter the notification instance name. **Example** In the Figure 23, `vThunderNotificationInstance` is the notification instance name.
SMTP Host	Enter the URL or IP address of the email host server.
SMTP Port	Enter the SMTP port number used to connect with the email host server.
Secure Connection Type	Select **SSL**.
User Name	Enter the username that is used to connect to the email server.
Password	Enter the password for the connection username that appears on the notification message.
Sender Email Address	Enter the email address of the sender.
Sender Name	Enter the display name of the sender email address.
Receiver Email Address	Enter the email address of the receiver that receives the notification.

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

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

    Figure 24 : Selected New Instance

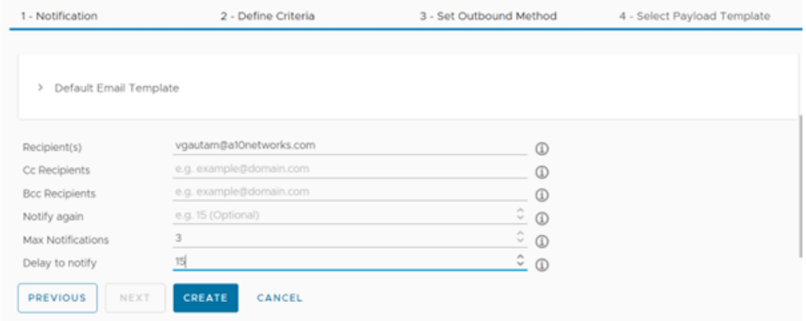
    Pick the outbound method you would like to use to send your notification.

    Outbound method: Standard Email Plugin | vThunderNotificationInstance

16. Click **Next**.

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

Figure 25 : Select Payload Template tab



Recipient(s) vgautam@a10networks.com
Cc Recipients e.g. example@domain.com
Bcc Recipients e.g. example@domain.com
Notify again e.g. 15 (Optional)
Max Notifications 3
Delay to notify 15

17. 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 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.

18. Click **Create**.

    A notification is created.

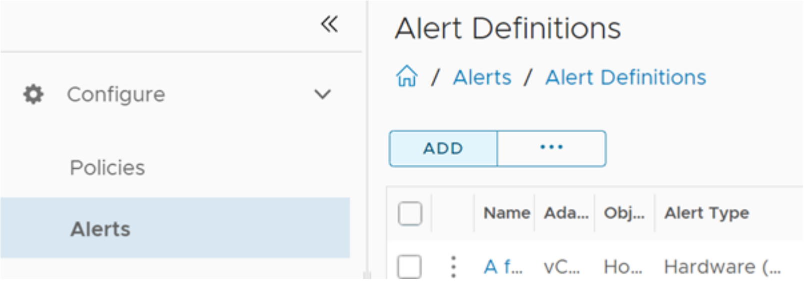
Create an Alert

To create an alert for metrics, perform the following steps:

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

   The **Alert Definitions** window is displayed.

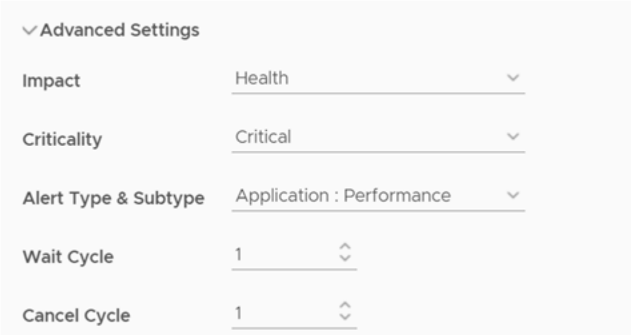
Figure 26 : Alert Definitions window



2. Click **Add**.

   A panel with **Alert** tab is displayed.

   Figure 27 : Alert Definitions window



Impact	Health
Criticality	Critical
Alert Type & Subtype	Application : Performance
Wait Cycle	1
Cancel Cycle	1

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

Table 27 : Alert Definitions

Field Name	Description
Name	Enter the alert name. **Example** `vThunderAlerts` is the notification instance name.
Base Object Type	Select **vCenter Adapter** > **Virtual Machine**.
Under the **Advanced Settings**:	
Impact	Select **Health**.
Criticality	Select **Critical**.

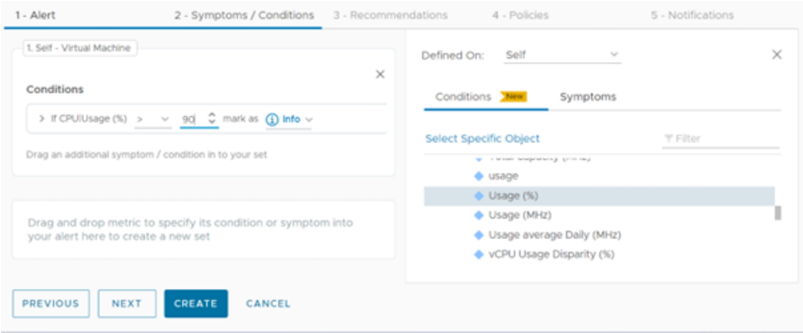
Table 27 : Alert Definitions

Field Name	Description
Alert Type & Subtype	Select **Application : Performance**.

4. Click **Next**.

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

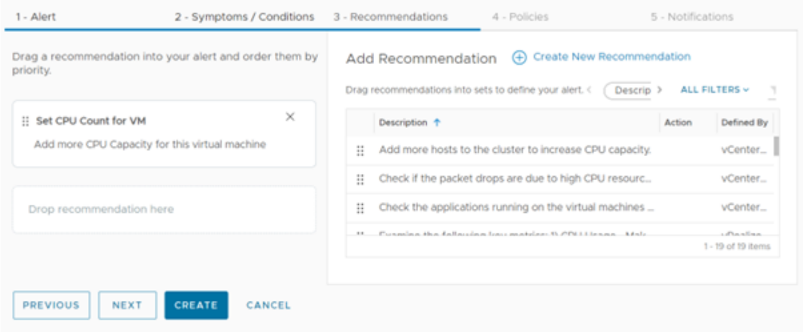
   Figure 28 : Symptoms / Conditions tab


1 - Alert
2 - Symptoms / Conditions
3 - Recommendations
4 - Policies
5 - Notifications
1. Self - Virtual Machine
Conditions
> If CPU|Usage (%) > 90 mark as Info
Drag an additional symptom / condition in to your set
Drag and drop metric to specify its condition or symptom into your alert here to create a new set
Defined On: Self
Conditions New
Symptoms
Select Specific Object
Filter
usage
Usage (%)
Usage (MHz)
Usage average Daily (MHz)
vCPU Usage Disparity (%)
PREVIOUS
NEXT
CREATE
CANCEL

5. Select **Metric** > **CPU** > **Usage (%)** and drag to the left-side panel.
6. Add the appropriate condition.
7. Click **Next**.

   The **Recommendations** tab is displayed.

   Figure 29 : Recommendations tab


1 - Alert
2 - Symptoms / Conditions
3 - Recommendations
4 - Policies
5 - Notifications
Drag a recommendation into your alert and order them by priority.
Set CPU Count for VM
Add more CPU Capacity for this virtual machine
Drop recommendation here
Add Recommendation
Create New Recommendation
Drag recommendations into sets to define your alert.
Descrip
ALL FILTERS
Description
Action
Defined By
Add more hosts to the cluster to increase CPU capacity.
vCenter...
Check if the packet drops are due to high CPU resourc...
vCenter...
Check the applications running on the virtual machines ...
vCenter...
1 - 19 of 19 items
PREVIOUS
NEXT
CREATE
CANCEL

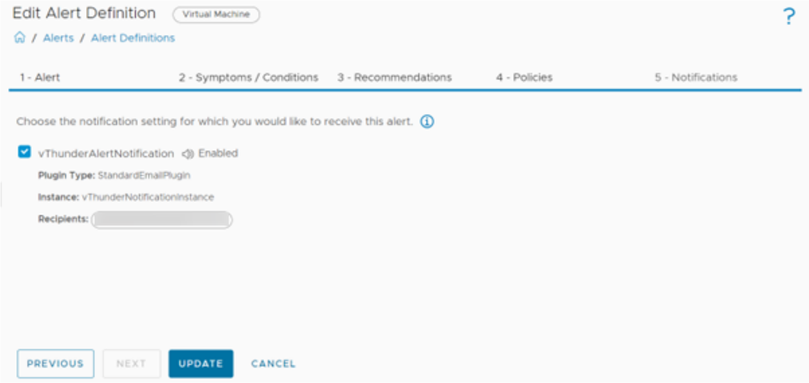
8. Add the appropriate recommendations.
9. Click **Next**.

The **Policies** tab is displayed.

10. Select appropriate policy.
11. Click **Next**.

    The **Notifications** tab is displayed.

    Figure 30 : Notifications tab



Choose the notification setting for which you would like to receive this alert.

☑ vThunderAlertNotification Enabled

Plugin Type: StandardEmailPlugin

Instance: vThunderNotificationInstance

Recipients: [redacted]

12. Select the notification created in the above step.
13. Click **Create**.

    An alert definition is created.

View VMware Metrics

To view the VMware metrics, perform the following steps:

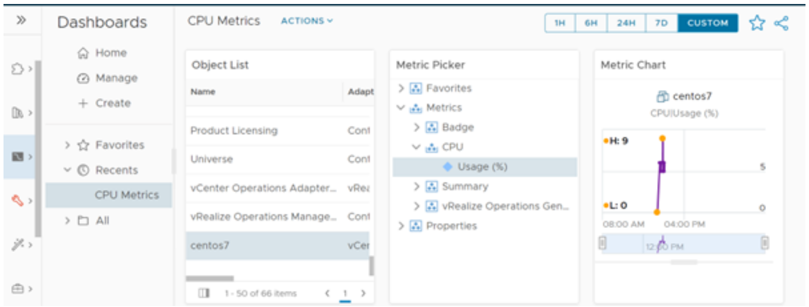
1. From the **vRealize Operations Web UI**, go to **Home** > **Visualize** > **Dashboard** and select the required metrics.
   For example, select **CPU Metrics** to verify CPU Metrics on the dashboard.

   a. From **Object List**, double-click *<your_Thunder_instance>*.

   b. From **Metric Picker**, double-click **Usage (%)**.

   c. From the dashboard, select the time range for which you want to display the CPU Usage.

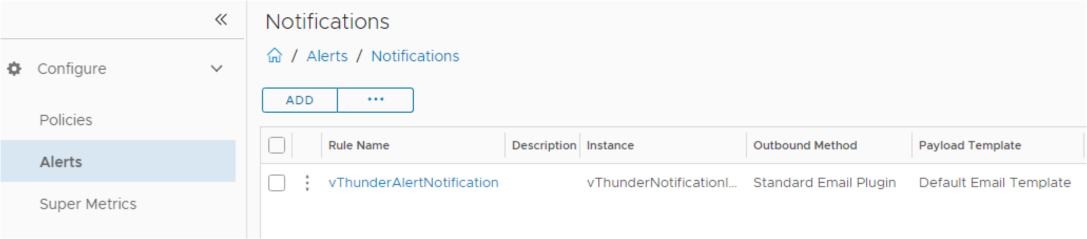
Figure 31 : CPU Metrics



Name	Adapt
Product Licensing	Cont
Universe	Cont
vCenter Operations Adapter...	vRea
vRealize Operations Manage...	Cont
centos7	vCer

2. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** > **Notifications** to verify if the notification is created.

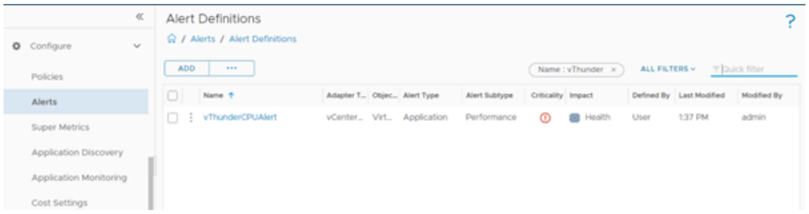
Figure 32 : Verify Notification



☐	Rule Name	Description	Instance	Outbound Method	Payload Template
☐	vThunderAlertNotification		vThunderNotificationI...	Standard Email Plugin	Default Email Template

3. From the **vRealize Operations Web UI**, go to **Home** > **Configure** > **Alerts** > **Alert Definitions** to verify if the alert definition is created.

Figure 33 : Verify Alert Definition



☐	Name ↑	Adapter T...	Objec...	Alert Type	Alert Subtype	Criticality	Impact	Defined By	Last Modified	Modified By
☐	vThunderCPUAlert	vCenter...	Virt...	Application	Performance	ⓘ	Health	User	1:37 PM	admin

Monitor Logs

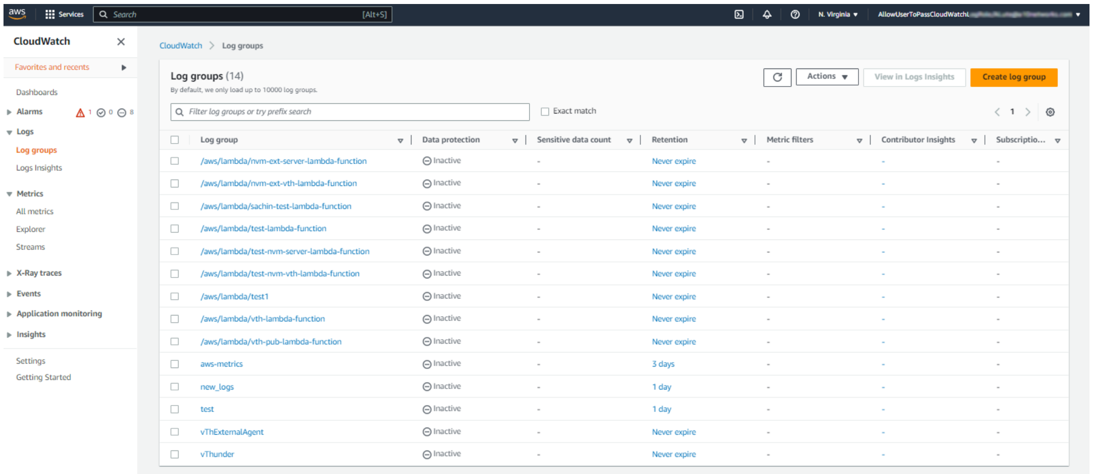
Depending on your cloud provider, the steps are provided to monitor the configured logs.

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 34 : AWS Log Groups



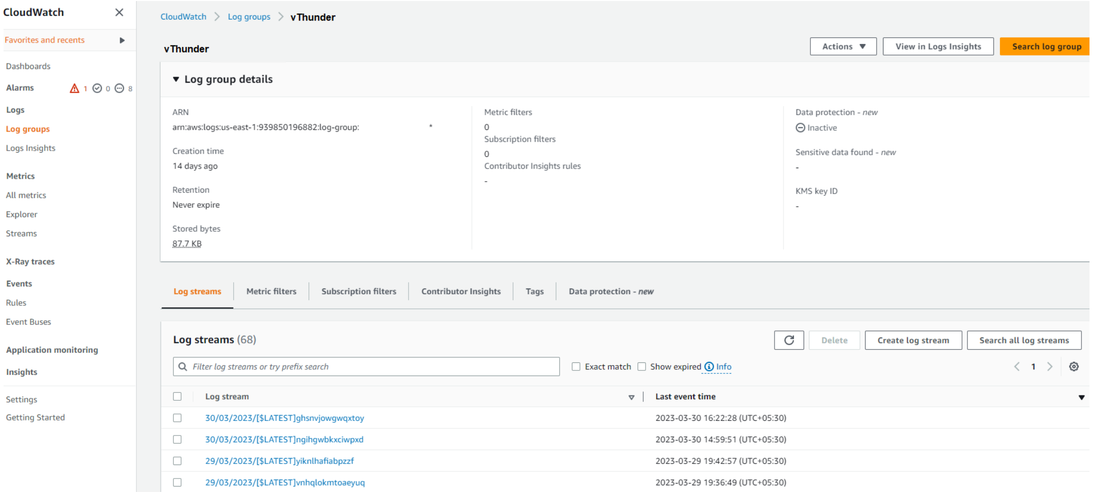
Log groups (14)

By default, we only load up to 10000 log groups.

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

2. Select Thunder log group.

   Figure 35 : vThunder Log Group



vThunder

Log group details

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

Creation time
14 days ago

Retention
Never expire

Stored bytes
87.7 KB

Metric filters
0

Subscription filters
0

Contributor Insights rules
-

Data protection - *new*
Inactive

Sensitive data found - *new*
-

KMS key ID
-

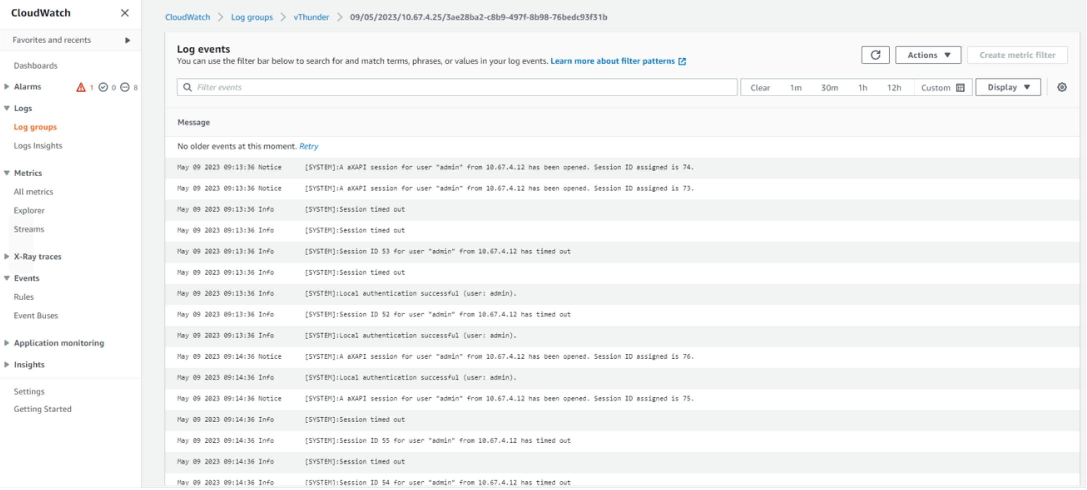
Log streams (68)

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

3. Under the **Log streams** tab, apply the appropriate filters to search for a specific log.

   All logs are displayed in tabular format with expandable details.

Figure 36 : Logs events on AWS CloudWatch



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

No older events at this moment. Retry

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

Azure Log Analytics Workspace

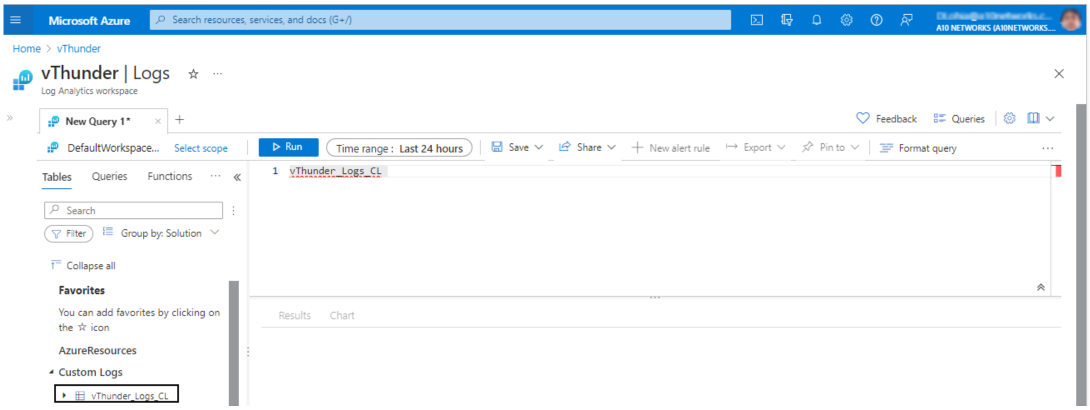
To monitor the Thunder logs on the Azure Log Analytics Workspace, perform the following steps:

1. From the **Azure Portal**, go to **Azure Services** > **Log Analytics workspaces** > **vThunder**.
2. Click **Logs** from the left **General** panel.

   Close the **Queries** pop-up window.
3. From **New Query1** > **Tables** tab, expand **Custom Logs**.
4. Double click **vThunder_Logs_CL**.

   The **vThunder_Logs_CL** query window is displayed.

Figure 37 : vThunder - Logs - Custom Logs window

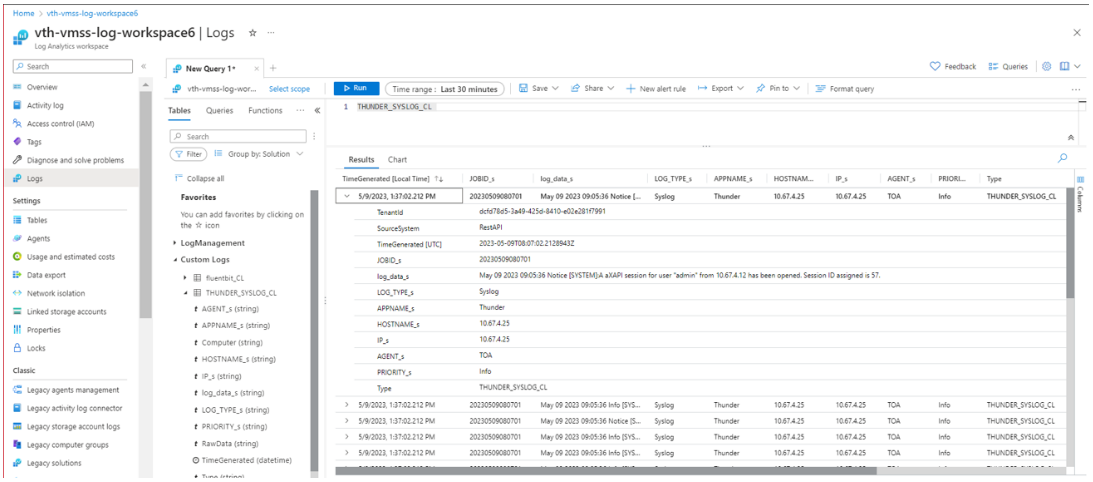


```
vThunder_Logs_CL
```

5. Click **Run**.

   All logs are displayed in tabular format with expandable details.

   Figure 38 : Logs on Azure Log Analytics Workspace



```
THUNDER_SYSLOG_CL
```

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

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

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

VMware vRLI

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

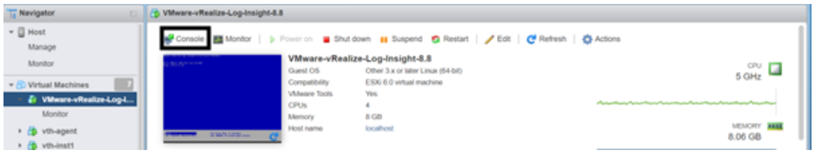
1. Start vRLI VM
2. View Logs

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 39 : Start vRLI VM

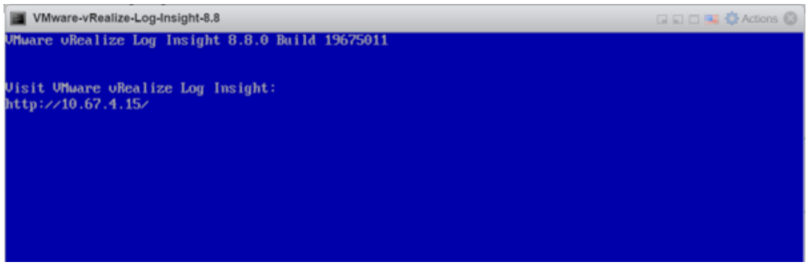
   VMware-vRealize-Log-Insight-8.8
   Guest OS Other 3.x or later Linux (64-bit)
   Compatibility ESXi 6.0 virtual machine
   VMware Tools Yes
   CPUs 4
   Memory 8 GB
   Host name localhost

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

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

   The vRLI virtual machine is powered on and reachable.

   Figure 40 : VMware vRealize Log Insight

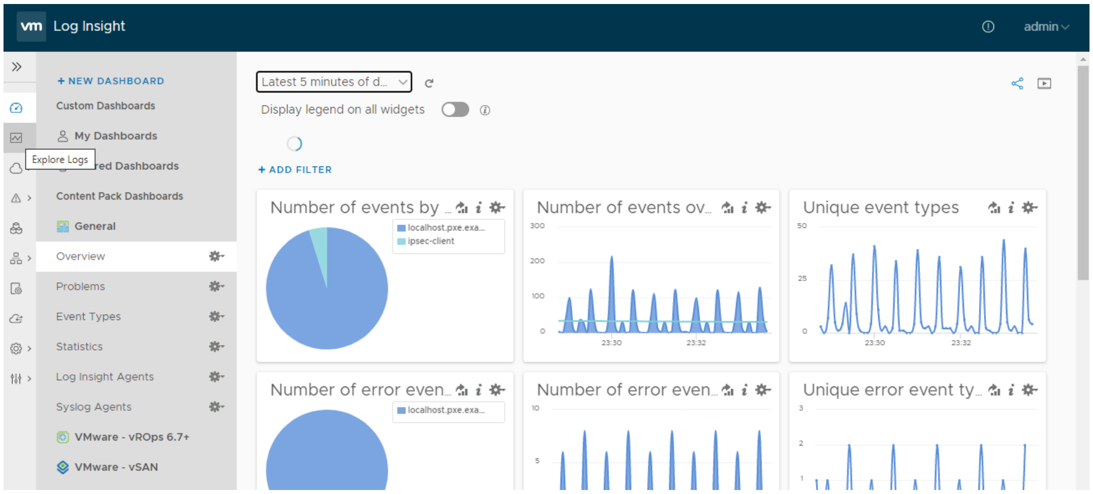
```
VMware vRealize Log Insight 8.8.0 Build 19675011


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

View Logs

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

   The **Logs** window is displayed.

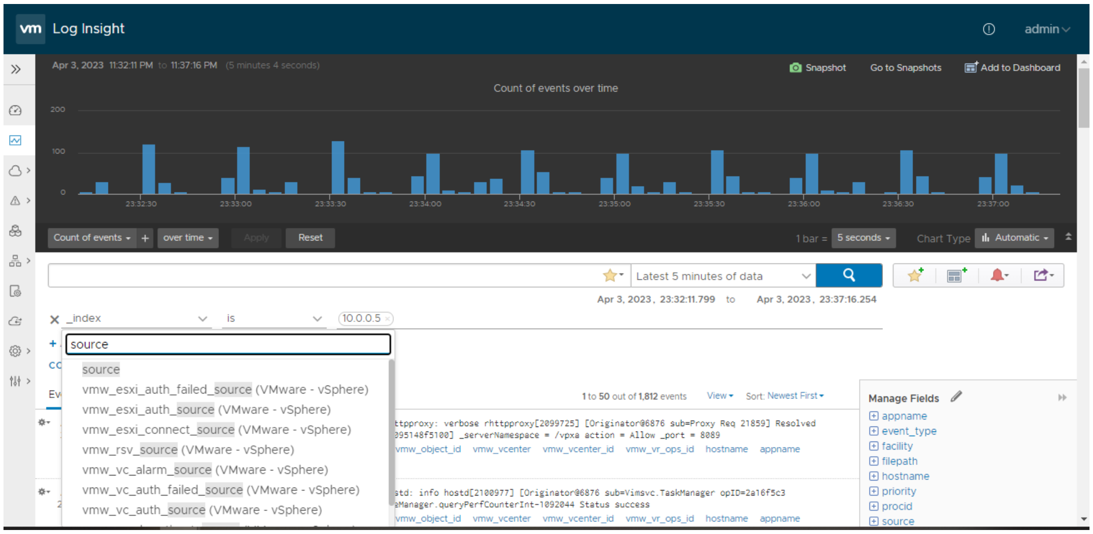
Figure 41 : vRealize Log Insight - Overview window



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

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

Figure 42 : vRealize Log Insight - Add Filter



_index is 10.0.0.5

source

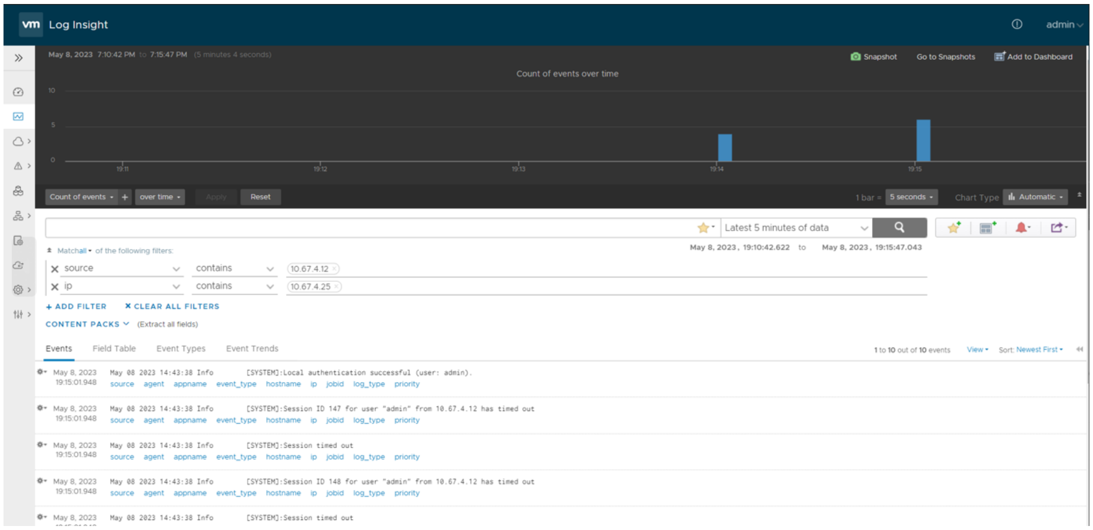
- source
- vmw_esxi_auth_failed_source (VMware - vSphere)
- vmw_esxi_auth_source (VMware - vSphere)
- vmw_esxi_connect_source (VMware - vSphere)
- vmw_rsv_source (VMware - vSphere)
- vmw_vc_alarm_source (VMware - vSphere)
- vmw_vc_auth_failed_source (VMware - vSphere)
- vmw_vc_auth_source (VMware - vSphere)

3. Add the following filter criteria to search all logs received from thunder-observability-agent IP:

- _index: source
- condition: is
- value: <*thunder-observability-agent_IP*>

4. Verify if the logs are generated.

Figure 43 : Logs on vRealize Log Insight



May 8, 2023 7:10:42 PM to 7:15:47 PM (5 minutes 4 seconds)

May 8, 2023 , 19:10:42.622 to May 8, 2023 , 19:15:47.043

Matchall of the following filters:

source	contains	10.67.4.12
ip	contains	10.67.4.25

1 to 10 out of 10 events

```
May 8, 2023     May 08 2023 14:43:38 Info       [SYSTEM]:Local authentication successful (user: admin).
19:15:01.948    source  agent  appname  event_type  hostname  ip  jobid  log_type  priority

May 8, 2023     May 08 2023 14:43:38 Info       [SYSTEM]:Session ID 147 for user "admin" from 10.67.4.12 has timed out
19:15:01.948    source  agent  appname  event_type  hostname  ip  jobid  log_type  priority

May 8, 2023     May 08 2023 14:43:38 Info       [SYSTEM]:Session timed out
19:15:01.948    source  agent  appname  event_type  hostname  ip  jobid  log_type  priority

May 8, 2023     May 08 2023 14:43:38 Info       [SYSTEM]:Session ID 148 for user "admin" from 10.67.4.12 has timed out
19:15:01.948    source  agent  appname  event_type  hostname  ip  jobid  log_type  priority

May 8, 2023     May 08 2023 14:43:38 Info       [SYSTEM]:Session timed out
```

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 ##### TOA #####
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
```

```
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

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

The client has shared the following environment details:

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

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	vThunder
aws_log_group_name	vThunder
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 Thunder.

```
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_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": "vThunder",
  "aws_log_group_name": "vThunder",
 }
```

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

For more examples, see GitHub.

Azure

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

The client has shared the following environment details:

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

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

Solution

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

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

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

2. Install Thunder.

```
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-1234567890rstuvw0"
       }]
}
```

Autoscale instance

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

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

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

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

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

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

For more examples, see GitHub.

VMware

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

The client has shared the following environment details:

Parameter	Description
Linux Environment IP	10.22.32.51
Hardware	2 GB RAM, 1 CPU, 4 GB memory
Thunder details	
Thunder instance	1
Thunder IP	10.22.32.01
User Name	Online_Baseball_TH
Password	Thunder@LMQ@3201
Resource_Name	North_Virginia_Online_Baseball_TH
resource_id	i-1234567890lmnopq0
Thunder instance	2
Thunder IP	10.22.32.02
User Name	Online_Baseball_TH2
Password	Thunder@LMQ@3202
Resource_Name	North_Virginia_Online_Baseball_TH2
resource_id	i-1234567890rstuvw0
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 Thunder.

```
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-1234567890rstuvw0"
       }]
}
```

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

Download Installation Files:

- Python Plugin Online | Offline
- Docker Container Online

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 fourteen 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 Thunder Metric Support.
- 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

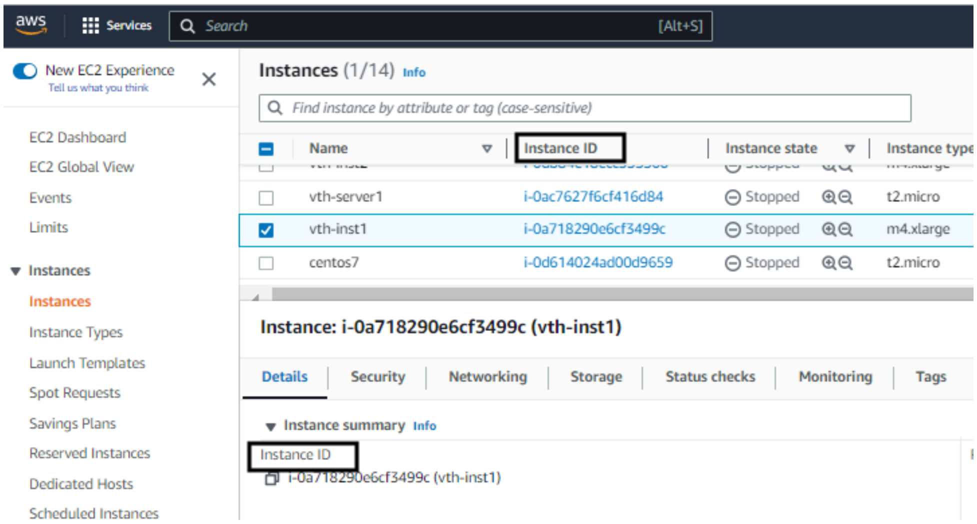
Get Resource ID

To get the resource ID for Thunder non-autoscale instance, 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 44 : Thunder instance Resource ID



Instances (1/14)

	Name	Instance ID	Instance state		Instance type
□	vth-server1	i-0ac7627f6cf416d84	⊖ Stopped	🔍🔍	t2.micro
☑	vth-inst1	i-0a718290e6cf3499c	⊖ Stopped	🔍🔍	m4.xlarge
□	centos7	i-0d614024ad00d9659	⊖ Stopped	🔍🔍	t2.micro

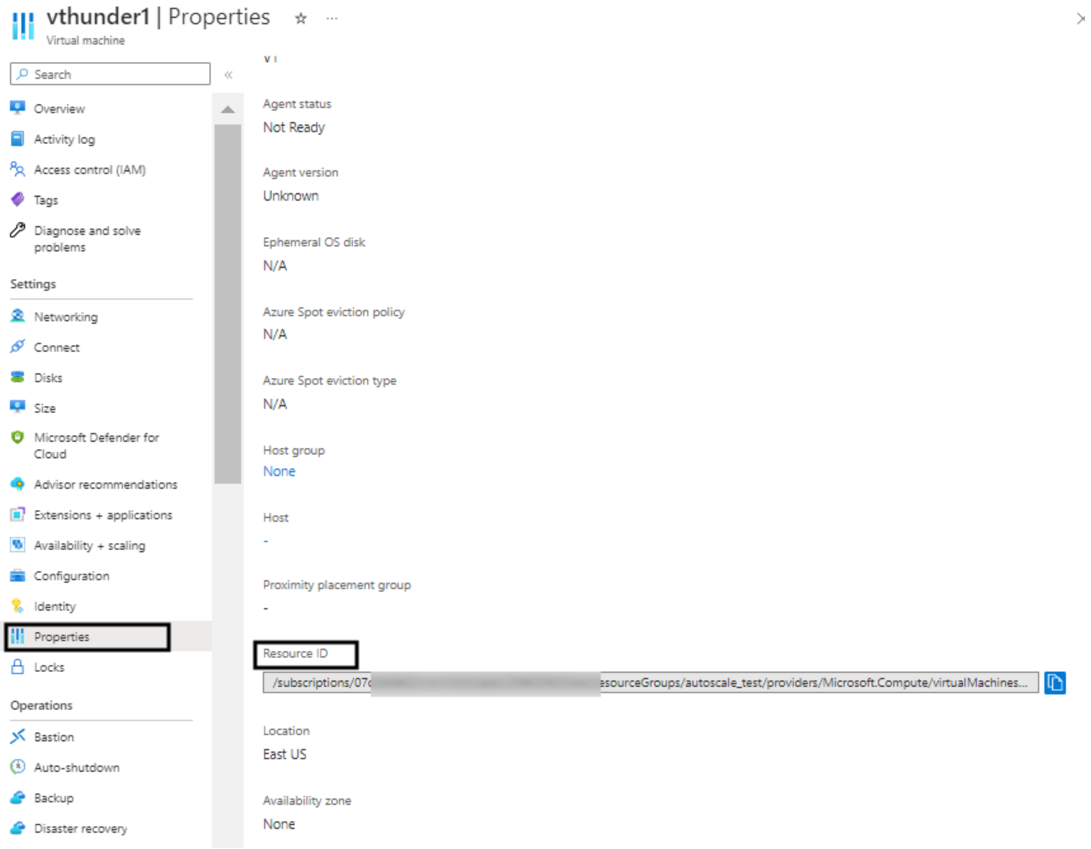
Instance: i-0a718290e6cf3499c (vth-inst1)

Instance ID
i-0a718290e6cf3499c (vth-inst1)

Azure

1. Go to **Azure Portal** > **Azure Services** > **Virtual machine** and select your Thunder instance.
2. From the left panel, click **Setting** > **Properties**.
3. Get the **Resource ID** from the right panel.

Figure 45 : Thunder instance Resource ID



Agent status
Not Ready

Agent version
Unknown

Ephemeral OS disk
N/A

Azure Spot eviction policy
N/A

Azure Spot eviction type
N/A

Host group
None

Host
-

Proximity placement group
-

Resource ID
/subscriptions/07[redacted]esourceGroups/autoscale_test/providers/Microsoft.Compute/virtualMachines...

Location
East US

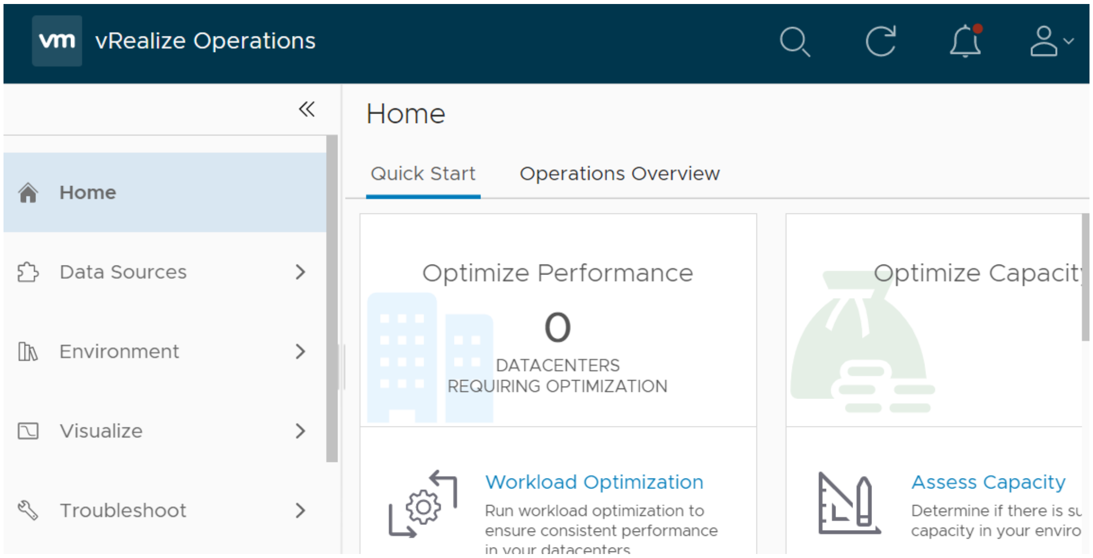
Availability zone
None

VMware

1. Log in to the **vRealize Operations Web UI** with your admin credentials to get the vThunder Resource ID once your vROPs virtual machine is powered on.

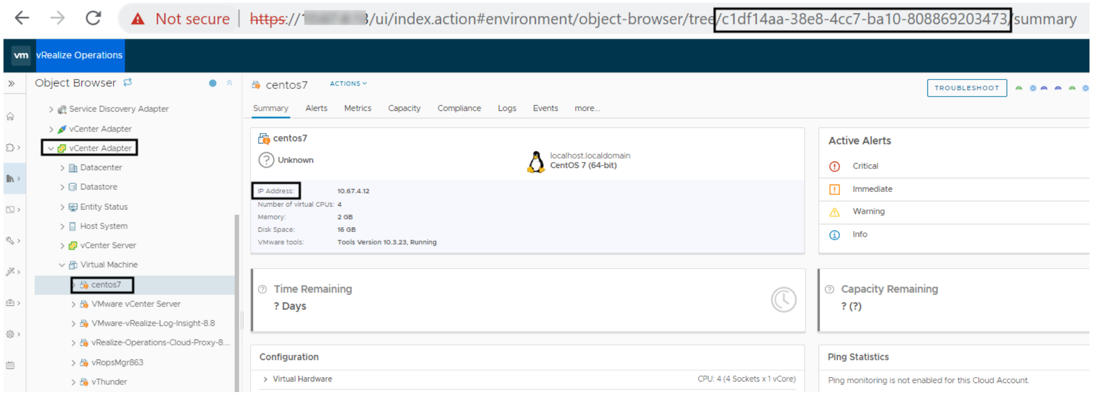
   The vRealize Operations Home page is displayed.

Figure 46 : vRealize Operations - Home page



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

Figure 47 : vRealize Operations - Virtual machine window


Not secure | https://1[redacted]3/ui/index.action#environment/object-browser/tree/c1df14aa-38e8-4cc7-ba10-808869203473/summary

centos7
Unknown
localhost.localdomain
CentOS 7 (64-bit)

IP Address: 10.67.4.12
Number of virtual CPUs: 4
Memory: 2 GB
Disk Space: 16 GB
VMware tools: Tools Version 10.3.23, Running

Active Alerts
Critical
Immediate
Warning
Info

Time Remaining
? Days

Capacity Remaining
? (?)

Configuration
Virtual Hardware CPU: 4 (4 Sockets x 1 vCore)

Ping Statistics
Ping monitoring is not enabled for this Cloud Account.

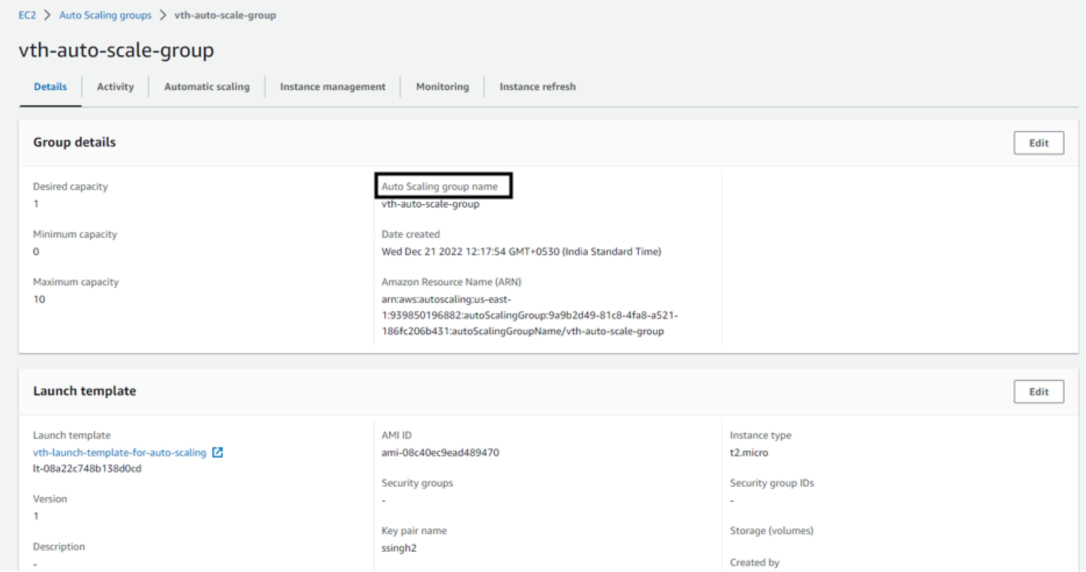
3. Get the resource ID from the URL.

To get the resource ID for Thunder autoscale instance, 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 48 : Thunder Auto Scale instance Resource ID



vth-auto-scale-group

Group details

Desired capacity
1

Minimum capacity
0

Maximum capacity
10

Auto Scaling group name
vth-auto-scale-group

Date created
Wed Dec 21 2022 12:17:54 GMT+0530 (India Standard Time)

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:939850196882:autoScalingGroup:9a9b2d49-81c8-4fa8-a521-186fc206b431:autoScalingGroupName/vth-auto-scale-group

Launch template

Launch template
vth-launch-template-for-auto-scaling
lt-08a22c748b138d0cd

Version
1

Description
-

AMI ID
ami-08c40ec9ead489470

Security groups
-

Key pair name
ssingh2

Instance type
t2.micro

Security group IDs
-

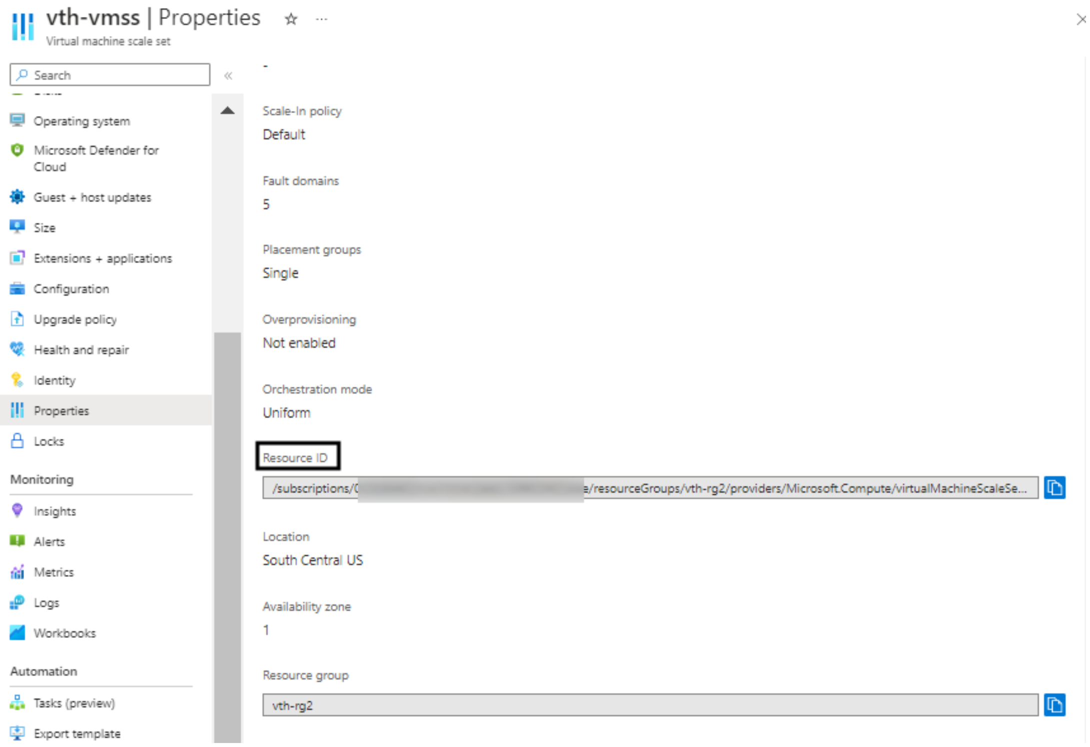
Storage (volumes)

Created by

Azure

1. Go to **Azure Portal** > **Azure Services** > **Virtual machine scale set** and select your Thunder autoscale instance.
2. From the left panel, click **Setting** > **Properties**.
3. Get the **Resource Group name** from the right panel.

Figure 49 : Thunder Auto Scale instance Resource ID



Scale-In policy
Default

Fault domains
5

Placement groups
Single

Overprovisioning
Not enabled

Orchestration mode
Uniform

Resource ID
/subscriptions/[redacted]e/resourceGroups/vth-rg2/providers/Microsoft.Compute/virtualMachineScaleSe...

Location
South Central US

Availability zone
1

Resource group
vth-rg2

Install Python

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

CentOS

```
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

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

Contact Us