



Connectors

- [What are Connectors, on page 1](#)
- [Virtual Appliances for Connectors, on page 85](#)
- [Life Cycle Management of Connectors, on page 96](#)
- [Configuration Management on Connectors and Virtual Appliances, on page 101](#)
- [Troubleshooting, on page 116](#)
- [Connector Alerts, on page 145](#)

What are Connectors

Connectors allow Secure Workload to integrate with other resources, for various purposes including:

- [Connectors for Flow Ingestion](#)
- [Connectors for Inventory Enrichment](#)
- [Cloud Connectors](#)
- [Connectors for Endpoints](#)
- [Connectors for Alert Notifications](#)

Connectors require a virtual appliance. For more information, see [Virtual Appliances for Connectors](#).

Navigating to the Connectors Page

To configure and work with connectors, click **Manage** > **Connectors** in the navigation bar at the left side of the window.

Connectors for Flow Ingestion

Connectors stream flow observations from different Network switches, routers, and other middle-boxes (such as load balancers and firewalls) to Secure Workload for flow ingestion.

Secure Workload supports flow ingestion through NetFlow v9, IPFIX, and custom protocols. In addition to flow observations, middle-box connectors actively stitch client-side and server-side flows to understand which client flows are related to which server flows.

Connector	Description	Deployed on Virtual Appliance
NetFlow	Collect NetFlow V9 and/or IP-FIX telemetry from network devices such as routers and switches.	Secure Workload Ingest
F5 BIG-IP	Collect telemetry from F5 BIG-IP, stitch client, and server side flows, enrich client inventory with user attributes.	Secure Workload Ingest
Citrix NetScaler	Collect telemetry from Citrix ADC, stitch client, and server side flows.	Secure Workload Ingest
Cisco Secure Connector Firewall	Collect telemetry data from Secure Firewall ASA, Secure Firewall Threat Defense, stitch client, and server side flows.	Secure Workload Ingest
Meraki	Collect telemetry data from Meraki firewalls.	Secure Workload Ingest
ERSPAN	Collect ERSPAN telemetry data from network devices which support ERSPAN	Secure Workload Ingest
See also	Cloud Connectors	–

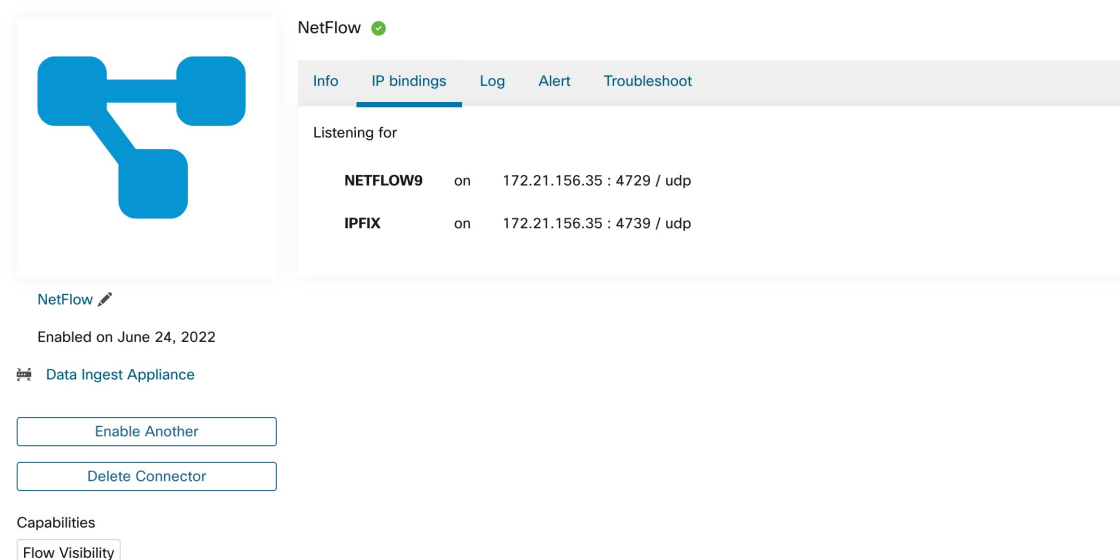
For more information about required virtual appliances, see [Virtual Appliances for Connectors](#).

NetFlow Connector

NetFlow connector allows Secure Workload to ingest flow observations from routers and switches in the network.

This solution enables the hosts to avoid running software agents since the Cisco switches relay NetFlow records to a NetFlow connector hosted in a Secure Workload Ingest appliance for processing.

Figure 1: NetFlow connector




NetFlow ✔


Info IP bindings Log Alert Troubleshoot

Listening for

NETFLOW9	on	172.21.156.35 : 4729 / udp
IPFIX	on	172.21.156.35 : 4739 / udp

NetFlow 

Enabled on June 24, 2022

 Data Ingest Appliance

Enable Another

Delete Connector

Capabilities

Flow Visibility

What is NetFlow

NetFlow protocol allows routers and switches to aggregate traffic passing through them into flows and export these flows to a flow collector.

The flow collector receives these flow records and stores them in their flow storage for offline querying and analysis. Cisco routers and switches support NetFlow.

Typically, the setup involves the following steps:

1. Enable the NetFlow feature on one or more network devices and configure the flow templates that devices should export.
2. Configure the NetFlow collector endpoint information on the remote network devices. This NetFlow collector is listening on the configured endpoint to receive and process NetFlow flow records.

Flow Ingestion to Secure Workload

NetFlow connector is essentially a NetFlow collector. The connector receives the flow records from the network devices and forwards them to Secure Workload for flow analysis. You can enable a NetFlow connector on a Secure Workload Ingest appliance and run it as a Docker container.

NetFlow connector also registers with Secure Workload as a Secure Workload NetFlow agent. NetFlow connector decapsulates the NetFlow protocol packets (that is, flow records); then processes and reports the flows like a regular Secure Workload agent. Unlike a Deep Visibility Agent, it does not report any process or interface information.



Note NetFlow connector supports NetFlow v9 and IPFIX protocols.



Note Each NetFlow connector should report only flows for one VRF. The connector exports the flows and places them in the VRF based on the Agent VRF configuration in the Secure Workload cluster.

To configure the VRF for the connector, choose **Manage > Agents** and click the **Configuration** tab. In this page, under the *Agent Remote VRF Configurations* section, click *Create Config* and provide the details about the connector.

The form requests you to provide: the name of the VRF, the IP subnet of the connector, and the range of port numbers that can potentially send flow records to the cluster.

Rate Limiting

NetFlow connector accepts up to 15000 flows per second. Note that a given NetFlow v9 or IPFIX packet could contain one or more flow and template records. NetFlow connector parses the packets and identifies the flows. If the connector parses more than 15000 flows per second, it drops the additional flow records.

Also note the Secure Workload customer supports the NetFlow connector only if the flow rate is within this acceptable limit.

If the flow rate exceeds 15000 flows per second, we recommend first adjusting the flow rate to fall within the limits, and maintaining this level for at least three days (to rule out issues related to higher incoming flow rate).

If the original issue persists, customer support starts to investigate the issue and identify proper workaround and/or solution.

Supported Information Elements

NetFlow connector *only* supports the following information elements in NetFlow v9 and IPFIX protocols. For more information, see [IP Flow Information Export \(IPFIX\) Entities](#).

Element ID	Name	Description	Mandatory
1	octetDeltaCount	Number of octets in incoming packets for this flow.	Yes
2	packetDeltaCount	Number of incoming packets for this flow.	Yes
4	protocolIdentifier	The value of the protocol number in the IP packet header.	Yes
6	tcpControlBits	TCP control bits observed for packets of this flow. The agent handles FIN, SYN, RST, PSH, ACK, and URG flags.	No
7	sourceTransportPort	The source port identifier in the transport header.	Yes

Element ID	Name	Description	Mandatory
8	sourceIPv4Address	The IPv4 source address in the IP packet header.	Either 8 or 27
11	destinationTransportPort	The destination port identifier in the transport header.	Yes
12	destinationIPv4Address	The IPv4 destination address in the IP packet header.	Either 12 or 28
27	sourceIPv6Address	The IPv6 source address in the IP packet header.	Either 8 or 27
28	destinationIPv6Address	The IPv6 destination address in the IP packet header.	Either 12 or 28
150	flowStartSeconds	The absolute timestamp of the first packet of the flow (in seconds).	No
151	flowEndSeconds	The absolute timestamp of the last packet of the flow (in seconds).	No
152	flowStartMilliseconds	The absolute timestamp of the first packet of the flow (in milliseconds).	No
153	flowEndMilliseconds	The absolute timestamp of the last packet of the flow (in milliseconds).	No
154	flowStartMicroseconds	The absolute timestamp of the first packet of the flow (in microseconds).	No
155	flowEndMicroseconds	The absolute timestamp of the last packet of the flow (in microseconds).	No
156	flowStartNanoseconds	The absolute timestamp of the first packet of the flow (in nanoseconds).	No
157	flowEndNanoseconds	The absolute timestamp of the last packet of the flow (in nanoseconds).	No

How to configure NetFlow on the Switch

The following steps are for a Nexus 9000 switch. The configurations may slightly differ for other Cisco platforms. In any case, refer to the official Cisco configuration guide for the Cisco platform you're configuring.

Step 1 Enter global configuration mode.

```
switch# configure terminal
```

Step 2 Enable NetFlow feature.

```
switch(config)# feature netflow
```

Step 3 Configure a flow record.

The following example configuration shows how to generate five tuple information of a flow in a NetFlow record.

```
switch(config)# flow record ipv4-records
switch(config-flow-record)# description IPv4Flow
switch(config-flow-record)# match ipv4 source address
switch(config-flow-record)# match ipv4 destination address
switch(config-flow-record)# match ip protocol
switch(config-flow-record)# match transport source-port
switch(config-flow-record)# match transport destination-port
switch(config-flow-record)# collect transport tcp flags
switch(config-flow-record)# collect counter bytes
switch(config-flow-record)# collect counter packets
```

Step 4 Configure a flow exporter.

The following example configuration specifies the NetFlow protocol version, NetFlow template exchange interval, and NetFlow collector endpoint details. Specify the IP and port on which you enable the NetFlow connector on a Secure Workload Ingest appliance.

```
switch(config)# flow exporter flow-exporter-one
switch(config-flow-exporter)# description NetFlowv9ToNetFlowConnector
switch(config-flow-exporter)# destination 172.26.230.173 use-vrf management
switch(config-flow-exporter)# transport udp 4729
switch(config-flow-exporter)# source mgmt0
switch(config-flow-exporter)# version 9
switch(config-flow-exporter-version-9)# template data timeout 20
```

Step 5 Configure a flow monitor.

Create a flow monitor and associate it with a flow record and flow exporter.

```
switch(config)# flow monitor ipv4-monitor
switch(config-flow-monitor)# description IPv4FlowMonitor
switch(config-flow-monitor)# record ipv4-records
switch(config-flow-monitor)# exporter flow-exporter-one
```

Step 6 Apply the flow monitor to an interface.

```
switch(config)# interface Ethernet 1/1
switch(config-if)# ip flow monitor ipv4-monitor input
```

The above steps configure NetFlow on the Nexus 9000 to export NetFlow v9 protocol packets for ingress traffic going through interface 1/1. It sends the flow records to 172.26.230.173:4729 over a UDP protocol. Each flow record includes five tuple information of the traffic and the byte/packet count of the flow.

Figure 2: Running configuration of NetFlow on Cisco Nexus 9000 Switch

```
[switch# show running-config netflow

!Command: show running-config netflow
!Time: Wed Mar 21 04:25:21 2018

version 7.0(3)I7(1)
feature netflow

flow timeout 60
flow exporter flow-exporter-173
  destination 172.26.230.173 use-vrf management
  transport udp 4729
  source mgmt0
  version 9
    template data timeout 20
flow record ipv4-records
  match ipv4 source address
  match ipv4 destination address
  match ip protocol
  match transport source-port
  match transport destination-port
  collect transport tcp flags
  collect counter bytes
  collect counter packets
  collect timestamp sys-uptime first
  collect timestamp sys-uptime last
flow monitor ipv4-monitor
  record ipv4-records
  exporter flow-exporter-173

interface Ethernet1/1
  ip flow monitor ipv4-monitor input

interface Ethernet1/2
  ip flow monitor ipv4-monitor input

switch#
```

How to Configure the Connector

For information about required virtual appliances, see [Virtual Appliances for Connectors](#). For NetFlow connectors, IPv4 and IPv6 (dual stack mode) addresses are supported. However, do note that dual stack support is a BETA feature.

The following configurations are allowed on the connector.

- *Log*: For more information, see [Log Configuration](#).

In addition, the listening ports of IPFIX protocol on the connector can be updated on the Docker container in Secure Workload Ingest appliance using an allowed command. This command can be issued on the appliance by providing the connector ID of the connector, type of the port to be update, and the new port information. The connector ID can be found on the connector page in Secure Workload UI. For more information, see [update-listening-ports](#).

Limits

Metric	Limit
Maximum number of NetFlow connectors on single Secure Workload Ingest appliance	3
Maximum number of NetFlow connectors on one Tenant (root scope)	10
Maximum number of NetFlow connectors on Secure Workload	100

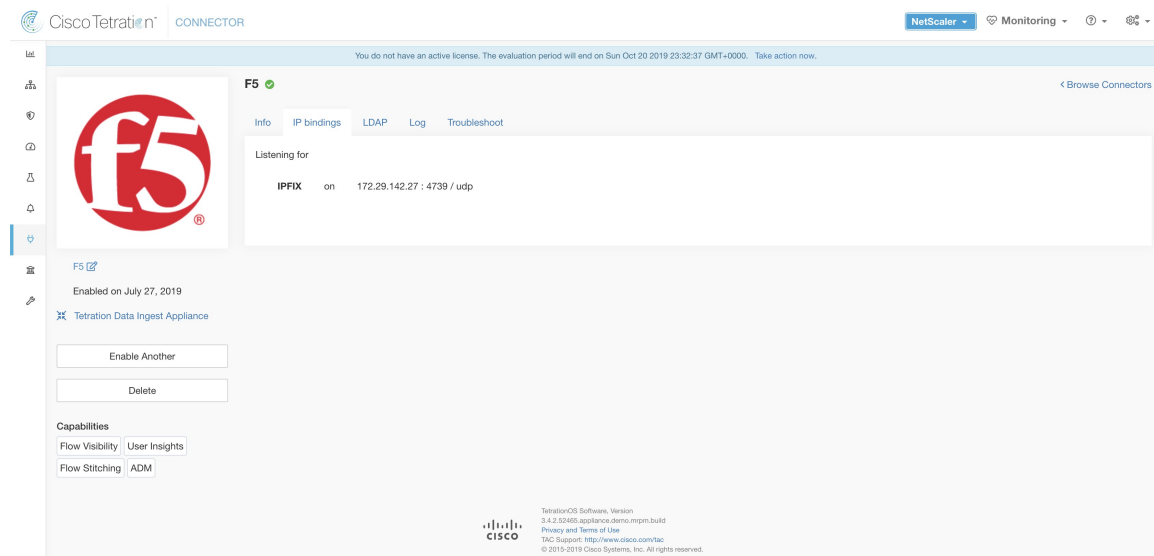
F5 Connector

The F5 connector allows Secure Workload to ingest flow observations from F5 BIG-IP ADCs.

It allows Secure Workload to remotely monitor of flow observations on F5 BIG-IP ADCs, stitching client-side and server-side flows, and annotating users on the client IPs (if user information is available).

Using this solution, the hosts don't need to run software agents because F5 BIG-IP ADCs configure the export of IPFIX records to the F5 connector for processing.

Figure 3: F5 connector



What is F5 BIG-IP IPFIX

[F5 BIG-IP IPFIX logging](#) collects flow data for traffic going through the F5 BIG-IP and exports IPFIX records to flow collectors.

Typically, the setup involves the following steps:

1. Create the IPFIX Log-Publisher on the F5 BIG-IP appliance.
2. Configure the IPFIX Log-Destination on the F5 BIG-IP appliance. This log-destination listens on the configured endpoint to receive and process flow records.
3. Create an F5 iRule that publishes IPFIX flow records to the log-publisher.
4. Add the F5 iRule to the virtual server of interest.



Note F5 connector supports F5 BIG-IP software version 12.1.2 and above.

Flow Ingestion to Secure Workload

F5 BIG-IP connector is essentially an IPFIX collector. The connector receives the flow records from F5 BIG-IP ADCs, stitch the NATed flows, and forwards them to Secure Workload for flow analysis. In addition, if LDAP configuration is provided to the F5 connector, it determines values for configured LDAP attributes of a user associated with the transaction (if F5 authenticates the user before processing the transaction). The attributes are associated to the client IP address where the flow happened.



Note F5 connector supports only the IPFIX protocol.



Note Each F5 connector reports only flows for one VRF. The connector puts the flows it exports into the VRF based on the Agent VRF configuration in the Cisco Secure Workload cluster.

To configure the VRF for the connector, choose **Manage > Agents** and click the **Configuration** tab. In this page, under the *Agent Remote VRF Configurations* section, click the *Create Config* and provide the details about the connector. The form requests you to provide: the name of the VRF, the IP subnet of the connector, and the range of port numbers that can potentially send flow records to the cluster.

How to configure IPFIX on F5 BIG-IP

The following steps are for F5 BIG-IP load balancer. (Ref: [Configuring F5 BIG-IP for IPFIX](#))

Purpose	Description
1. Create a pool of IPFIX collectors.	On a F5 BIG-IP appliance, create the pool of IPFIX collectors. These are the IP addresses associated with F5 connectors on a Secure Workload Ingest appliance. F5 connectors run in Docker containers on the VM listen on port 4739 for IPFIX packets.
2. Create a log-destination.	The log destination configuration on a F5 BIG-IP appliance specifies the actual pool of IPFIX collectors that are used.
3. Create a log-publisher.	A log publisher specifies where F5 BIG-IP sends the IPFIX messages. The publisher is bound with a log-destination.
4. Add a F5 and Secure Workload approved iRule.	Secure Workload and F5 developed iRules that will export flow records to F5 connectors. These iRules will export complete information about a given transaction: including all the endpoints, byte and packet counts, flow start and end time (in milliseconds). F5 connectors will create 4 independent flows and match each flow with its related flow.
5. Add the iRule to the virtual server.	In the iRule settings of a virtual server, add the Secure Workload, approved iRule to the virtual server.

The above steps configures IPFIX on F5 BIG-IP load balancer to export IPFIX protocol packets for traffic going through the appliance. Here is a sample config of F5.

Figure 4: Running configuration of IPFIX on F5 BIG-IP load balancer

```

root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmsh)# show running-config ltm virtual vip-1 rules
ltm virtual vip-1 {
  rules {
    ipfix-rule-1
  }
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmsh)# show running-config ltm pool ipfix-pool-1
ltm pool ipfix-pool-1 {
  members {
    10.28.118.6:ipfix {
      address 10.28.118.6
      session monitor-enabled
      state up
    }
  }
  monitor gateway_icmp
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmsh)# show running-config ltm virtual vip-1 rules
ltm virtual vip-1 {
  rules {
    ipfix-rule-1
  }
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmsh)# show running-config sys log-config
sys log-config destination ipfix ipfix-collector-1 {
  pool-name ipfix-pool-1
  transport-profile udp
}
sys log-config publisher ipfix-pub-1 {
  destinations {
    ipfix-collector-1 { }
  }
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmsh)#

```

In the example above, flow records will be published to *ipfix-pub-1*. *ipfix-pub-1* is configured with log-destination *ipfix-collector-1* which sends the IPFIX messages to IPFIX pool *ipfix-pool-1*. *ipfix-pool-1* has 10.28.118.6 as one of the IPFIX collectors. The virtual server *vip-1* is configured with IPFIX iRule *ipfix-rule-1* which specifies the IPFIX template and how the template gets filled and sent.

- F5 and Secure Workload approved iRule for TCP virtual server. For more information, see [L4 iRule for TCP virtual server](#).
- F5 and Secure Workload approved iRule for UDP virtual server. For more information, see [L4 iRule for UDP virtual server](#).
- F5 and Secure Workload approved iRule for HTTPS virtual server. For more information, see [iRule for HTTPS virtual server](#).



Note Before using the iRule downloaded from this guide, update the **log-publisher** to point to the log-publisher configured in the F5 connector where you add the iRule.



Note F5 has published a GitHub repository, [f5-tetration](#) to help you to start with flow-stitching. The iRules for publishing IPFIX records to the F5 connector for various protocol types are available at: [f5-tetration/irules](#).

Visit the site for the latest iRule definitions. In addition, F5 also develops a script to:

1. Install the correct iRule for the virtual servers.
2. Add a pool of IPFIX collector endpoints (where F5 connectors listen for IPFIX records).
3. Configure the log-collector and log-publisher.
4. Bind the correct iRule to the virtual servers.

This tool minimizes manual configuration and user error while enabling flow-stitching use-case. The script is available at [f5-tetration/scripts](#).

How to Configure the Connector

For information about required virtual appliances, see [Virtual Appliances for Connectors](#).

The following configurations are allowed on the connector.

- **LDAP:** LDAP configuration supports discovery of LDAP attributes and provide a workflow to pick the attribute that corresponds to username and a list of up to 6 attributes to fetch for each user. For more information, see [Discovery](#).
- **Log:** For more information, see [Log Configuration](#).

In addition, the listening ports of IPFIX protocol on the connector can be updated on the Docker container in Secure Workload Ingest appliance using a command that is allowed to be run on the container. This command can be issued on the appliance by providing the connector ID of the connector, type of the port to be update, and the new port information. The connector ID can be found on the connector page in Secure Workload UI. For more information, see [update-listening-ports](#).

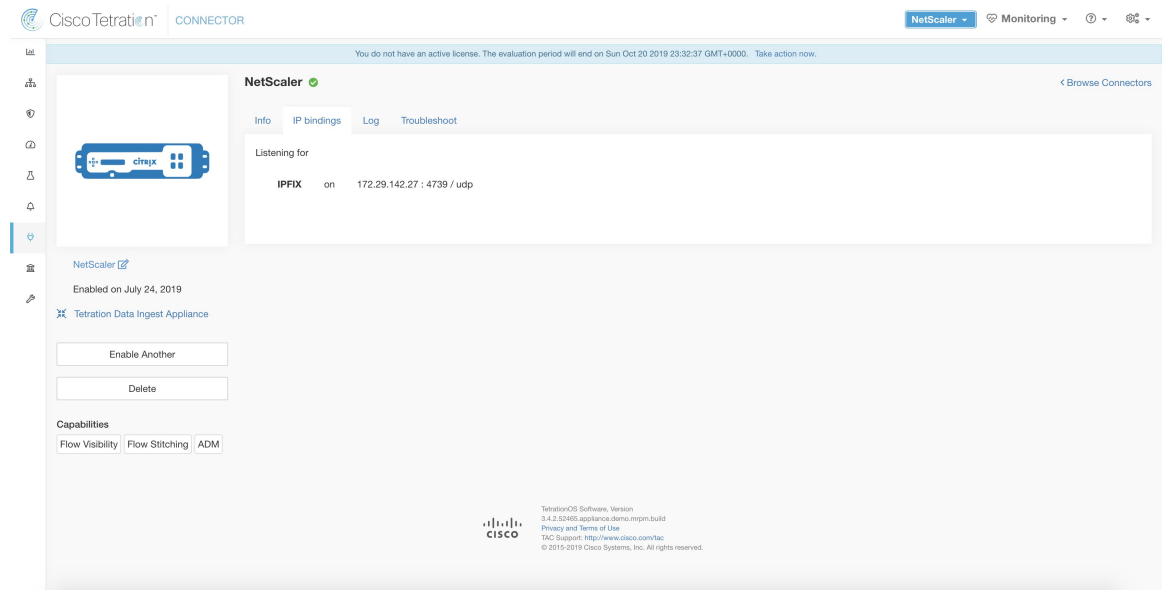
Limits

Metric	Limit
Maximum number of F5 connectors on one Secure Workload Ingest appliance	3
Maximum number of F5 connectors on one Tenant (rootscope)	10
Maximum number of F5 connectors on Secure Workload	100

NetScaler Connector

NetScaler connector allows Secure Workload to ingest flow observations from Citrix ADCs (Citrix NetScalers). It allows Secure Workload to remotely monitor flow observations on Citrix ADCs and stitch client-side and server-side flows. Using this solution, the hosts do not need to run software agents, because Citrix ADCs will be configured to export IPFIX records to NetScaler connector for processing.

Figure 5: NetScaler connector



What is Citrix NetScaler AppFlow

Citrix NetScaler AppFlow collects flow data for traffic going through the NetScaler and exports IPFIX records to flow collectors. Citrix AppFlow protocol uses IPFIX to export the flows to flow collectors. Citrix AppFlow is supported in Citrix NetScaler load balancers.

Typically, the setup involves the following steps:

1. Enable AppFlow feature on one or more Citrix NetScaler instances.
2. Configure the AppFlow collector endpoint information on the remote network devices. This AppFlow collector will be listening on configured endpoint to receive and process flow records.
3. Configure AppFlow actions and policies to export flow records to AppFlow collectors.



Note NetScaler connector supports Citrix ADC software version 11.1.51.26 and above.

Flow Ingestion to Secure Workload

NetScaler connector is essentially a Citrix AppFlow (IPFIX) collector. The connector receives the flow records from Citrix ADCs, stitch the NATed flows and forwards them to Secure Workload for flow analysis. A NetScaler connector can be enabled on a Cisco Secure Workload Ingest appliance and runs as a Docker container. NetScaler connector also registers with Secure Workload as a Secure Workload NetScaler agent.



Note NetScaler connector supports only IPFIX protocol.



Note Each NetScaler connector should report only flows for one VRF. The flows exported by the connector is put in the VRF based on the Agent VRF configuration in the Secure Workload cluster. To configure the VRF for the connector, go to: **Manage > Agents** and click the Configuration tab. In this page, under *Agent Remote VRF Configurations* section, click *Create Config* and provide the details about the connector. The form requests the user to provide: the name of the VRF, IP subnet of the connector, and range of port numbers that can potentially send flow records to the cluster.

How to configure AppFlow on NetScaler

The following steps are for NetScaler load balancer. (Ref: [Configuring AppFlow](#))

Step 1 Enable AppFlow on NetScaler.

```
enable ns feature appflow
```

Step 2 Add AppFlow collector endpoints.

The collector receives the AppFlow records from NetScaler. Specify the IP and port of NetScaler connector enabled on a Secure Workload Ingest appliance as an AppFlow collector.

```
add appflow collector c1 -IPAddress 172.26.230.173 -port 4739
```

Step 3 Configure an AppFlow action.

This lists the collectors that will get AppFlow records if the associated AppFlow policy matches.

```
add appflow action a1 -collectors c1
```

Step 4 Configure an AppFlow policy.

This is a rule that has to match for an AppFlow record to be generated.

```
add appflow policy p1 CLIENT.TCP.DSTPORT(22) a1
add appflow policy p2 HTTP.REQ.URL.SUFFIX.EQ("jpeg") a1
```

Step 5 Bind AppFlow policy to Virtual Server.

Traffic hitting the IP of the virtual server (VIP) will be evaluated for AppFlow policy matches. On a match, a flow record is generated and sent to all collectors listed in the associated AppFlow action.

```
bind lb vserver lb1 -policyname p1 -priority 10
```

Step 6 Optionally, bind AppFlow policy globally (for all virtual servers).

An AppFlow policy could also be bound globally to all virtual servers. This policy applies to all traffic that flows through Citrix ADC.

```
bind appflow global p2 1 NEXT -type REQ_DEFAULT
```

Step 7 Optionally, template refresh interval.

Default value for template refresh is 60 seconds.

```
set appflow param -templatereferesh 60
```

The above steps configures AppFlow on Citrix NetScaler load balancer to export IPFIX protocol packets for traffic going through NetScaler. The flow records will be sent to either 172.26.230.173:4739 (for traffic going through vserver lb1) and to 172.26.230.184:4739 (for all traffic going through the NetScaler). Each flow record includes 5 tuple information of the traffic and the byte/packet count of the flow.

The following screenshot shows a running configuration of AppFlow on a Citrix NetScaler load balancer.

Figure 6: Running configuration of AppFlow on Citrix NetScaler load balancer

```
MAARUMUG-M-M1PB:~ maarumug$ ssh nsroot@172.26.231.131
#####
#                                                                    #
#      WARNING: Access to this system is for authorized users only    #
#      Disconnect IMMEDIATELY if you are not an authorized user!      #
#                                                                    #
#####
Password:
Last login: Fri Dec 15 12:32:45 2017 from 10.128.140.136
Done
> sh run | grep appflow
add appflow collector c1 -IPAddress 172.26.230.174
add appflow collector c2 -IPAddress 172.26.230.173
set appflow param -templateRefresh 60 -connectionChaining ENABLED
add appflow action act1 -collectors c1 c2
add appflow policy pol1 true act1
bind appflow global pol1 1 NEXT -type REQ_DEFAULT
> █
```

How to Configure the Connector

For information about required virtual appliances, see [Virtual Appliances for Connectors](#). The following configurations are allowed on the connector.

- *Log:* . For more information, see [Log Configuration](#).

In addition, the listening ports of IPFIX protocol on the connector can be updated on the Docker container in Secure Workload Ingest appliance using a an allowed command. This command can be issued on the appliance by providing the connector ID of the connector, type of the port to be update, and the new port information. The connector ID can be found on the connector page in Secure Workload UI. . For more information, see [update-listening-ports](#).

Limits

Table 1: Limits

Metric	Limit
Maximum number of NetScaler connectors on one Secure Workload Ingest appliance	3
Maximum number of NetScaler connectors on one Tenant (rootscope)	10
Maximum number of NetScaler connectors on Secure Workload	100

Cisco Secure Firewall Connector

Secure Firewall Connector (formerly known as ASA Connector) allows Secure Workload to ingest flow observations from Secure Firewall ASA (formerly known as Cisco ASA) and Secure Firewall Threat Defense (formerly known as Firepower Threat Defense or FTD). Using this solution, the hosts do not need to run software agents, because the Cisco switches will relay NetFlow Secure Event Logging (NSEL) records to Secure Firewall Connector hosted in a Secure Workload Ingest appliance for processing.

Figure 7: Secure Firewall Connector

Cisco Secure Firewall ✔

Info IP bindings Log Alert Troubleshoot

Listening for

NETFLOW9	on	172.21.156.34 : 4729 / udp
----------	----	----------------------------

Cisco Secure Firewall ✎

Enabled on June 23, 2022

☰ Data Ingest Appliance

Enable Another

Delete Connector

Capabilities

Flow Visibility Flow Stitching

ADM

[Cisco Secure Firewall ASA NetFlow Secure Event Logging \(NSEL\)](#) provides a stateful, IP flow monitoring that exports significant events in a flow to a NetFlow collector. When an event causes a state change on a flow, an NSEL event is triggered that sends the flow observation along with the event that caused the state change to the NetFlow collector. The flow collector receives these flow records and stores them in their flow storage for offline querying and analysis.

Typically, the setup involves the following steps:

1. Enable NSEL feature on Secure Firewall ASA and/or Secure Firewall Threat Defense.
2. Configure the Secure Firewall connector endpoint information on Secure Firewall ASA and/or Secure Firewall Threat Defense. Secure Firewall connector will be listening on configured endpoint to receive and process NSEL records.

Flow Ingestion to Secure Workload

Secure Firewall connector is essentially a NetFlow collector. The connector receives the NSEL records from Secure Firewall ASA and Secure Firewall Threat Defense, and forwards them to Secure Workload for flow analysis. Secure Firewall connector can be enabled on a Secure Workload Ingest appliance and runs as a Docker container.

Secure Firewall connector also registers with Secure Workload as a Secure Workload agent. Secure Firewall connector decapsulates the NSEL protocol packets (i.e., flow records); then processes and reports the flows like a regular Secure Workload agent. Unlike a Deep Visibility Agent, it does not report any process or interface information.



Note Secure Firewall connector supports NetFlow v9 protocol.



Note Each Secure Firewall connector should report only flows for one VRF. The flows exported by the connector is put in the VRF based on the Agent VRF configuration in Secure Workload cluster. To configure the VRF for the connector, go to: **Manage > Agents** and click the **Configuration** tab. In this page, under *Agent Remote VRF Configurations* section, click *Create Config* and provide the details about the connector. The form requests the user to provide: the name of the VRF, IP subnet of the connector, and range of port numbers that can potentially send flow records to the cluster.

Handling NSEL Events

The following table shows how various NSEL events are handled by Secure Firewall connector. For more information about these elements, see [IP Flow Information Export \(IPFIX\) Entities](#) document.

Flow Event Element ID: 233 Element Name: <i>NF_F_FW_EVENT</i>	Extended Flow Event Element ID: 33002 Element Name: <i>NF_F_FW_EXT_EVENT</i>	Action on Secure Firewall connector
0 (default, ignore this value)	Don't care	No op
1 (Flow created)	Don't care	Send flow to Secure Workload
2 (Flow deleted)	> 2000 (indicates the termination reason)	Send flow to Secure Workload

Flow Event Element ID: 233 Element Name: <i>NF_F_FW_EVENT</i>	Extended Flow Event Element ID: 33002 Element Name: <i>NF_F_FW_EXT_EVENT</i>	Action on Secure Firewall connector
3 (Flow denied)	1001 (denied by ingress ACL)	Send flow with disposition marked as rejected to Secure Workload
	1002 (denied by egress ACL)	
	1003 (denied connection by ASA interface or denied ICMP(v6) to device)	
	1004 (first packet on TCP is not SYN)	
4 (Flow alert)	Don't care	No op
5 (Flow updated)	Don't care	Send flow to Secure Workload

Based on the NSEL record, Secure Firewall connector sends flow observation to Secure Workload. NSEL flow records are bidirectional. So, Secure Firewall connector sends 2 flows: forward flow and reverse flow to Secure Workload.

Here are the details about flow observation sent by Secure Firewall connector to Secure Workload.

Forward Flow observation

Field	NSEL Element ID	NSEL Element Name
Protocol	4	<i>NF_F_PROTOCOL</i>
Source Address	8	<i>NF_F_SRC_ADDR_IPV4</i>
	27	<i>NF_F_SRC_ADDR_IPV6</i>
Source Port	7	<i>NF_F_SRC_PORT</i>
Destination Address	12	<i>NF_F_DST_ADDR_IPV4</i>
	28	<i>NF_F_DST_ADDR_IPV6</i>
Destination Port	11	<i>NF_F_DST_PORT</i>
Flow Start Time	152	<i>NF_F_FLOW_CREATE_TIME_MSEC</i>
Byte Count	231	<i>NF_F_FWD_FLOW_DELTA_BYTES</i>
Packet Count	298	<i>NF_F_FWD_FLOW_DELTA_PACKETS</i>

Reverse Flow Information

Field	NSEL Element ID	NSEL Element Name
Protocol	4	<i>NF_F_PROTOCOL</i>

Field	NSEL Element ID	NSEL Element Name
Source Address	12	<i>NF_F_DST_ADDR_IPV4</i>
	28	<i>NF_F_DST_ADDR_IPV6</i>
Source Port	11	<i>NF_F_DST_PORT</i>
Destination Address	8	<i>NF_F_SRC_ADDR_IPV4</i>
	27	<i>NF_F_SRC_ADDR_IPV6</i>
Destination Port	7	<i>NF_F_SRC_PORT</i>
Flow Start Time	152	<i>NF_F_FLOW_CREATE_TIME_MSEC</i>
Byte Count	232	<i>NF_F_REV_FLOW_DELTA_BYTES</i>
Packet Count	299	<i>NF_F_REV_FLOW_DELTA_PACKETS</i>

NAT

If the client to ASA flow is NATed, NSEL flow records indicate the NATed IP/port on the server side. Secure Firewall connector uses this information to stitch server to ASA and ASA to client flows.

Here is the NATed flow record in the forward direction.

Field	NSEL Element ID	NSEL Element Name
Protocol	4	<i>NF_F_PROTOCOL</i>
Source Address	225	<i>NF_F_XLATE_SRC_ADDR_IPV4</i>
	281	<i>NF_F_XLATE_SRC_ADDR_IPV6</i>
Source Port	227	<i>NF_F_XLATE_SRC_PORT</i>
Destination Address	226	<i>NF_F_XLATE_DST_ADDR_IPV4</i>
	282	<i>NF_F_XLATE_DST_ADDR_IPV6</i>
Destination Port	228	<i>NF_F_XLATE_DST_PORT</i>
Flow Start Time	152	<i>NF_F_FLOW_CREATE_TIME_MSEC</i>
Byte Count	231	<i>NF_F_FWD_FLOW_DELTA_BYTES</i>
Packet Count	298	<i>NF_F_FWD_FLOW_DELTA_PACKETS</i>

The forward flow will be marked as related to the NATed flow record in the forward direction (and vice versa)

Here is the NATed flow record in the reverse direction

Field	NSEL Element ID	NSEL Element Name
Protocol	4	<i>NF_F_PROTOCOL</i>

Field	NSEL Element ID	NSEL Element Name
Source Address	226	<i>NF_F_XLATE_DST_ADDR_IPV4</i>
	282	<i>NF_F_XLATE_DST_ADDR_IPV6</i>
Source Port	228	<i>NF_F_XLATE_DST_PORT</i>
Destination Address	225	<i>NF_F_XLATE_SRC_ADDR_IPV4</i>
	281	<i>NF_F_XLATE_SRC_ADDR_IPV6</i>
Destination Port	227	<i>NF_F_XLATE_SRC_PORT</i>
Flow Start Time	152	<i>NF_F_FLOW_CREATE_TIME_MSEC</i>
Byte Count	232	<i>NF_F_REV_FLOW_DELTA_BYTES</i>
Packet Count	299	<i>NF_F_REV_FLOW_DELTA_PACKETS</i>

The reverse flow will be marked as related to the NATed flow record in the reverse direction (and vice versa).



Note Only NSEL element IDs listed in this section are supported by Secure Firewall connector.

TCP Flags Heuristics

The NSEL records do not have TCP flags information. The Secure Firewall connector uses the following heuristics to set the TCP flags so that the flows can be further analyzed by automatic policy discovery:

- If there are at least one forward packets, adds `SYN` to the forward flow TCP flags.
- If there are at least two forward packets and one reverse packet, adds `ACK` to the forward flow TCP flags and `SYN-ACK` to the reverse flow TCP flags.
- If the previous condition holds true and the flow event is Flow deleted, adds `FIN` to both forward and reverse TCP flags.

How to Configure NSEL on Secure Firewall ASA

The following steps are guidelines on how to configure NSEL and export NetFlow packets to a collector (i.e., Secure Firewall connector). For more information, see the official Cisco configuration guide at [Cisco Secure Firewall ASA NetFlow Implementation Guide](#) for more details.

Here is an example NSEL configuration.

```

flow-export destination outside 172.29.142.27 4729
flow-export template timeout-rate 1
!
policy-map flow_export_policy
class class-default
flow-export event-type flow-create destination 172.29.142.27
flow-export event-type flow-teardown destination 172.29.142.27
flow-export event-type flow-denied destination 172.29.142.27
flow-export event-type flow-update destination 172.29.142.27

```



```

user-statistics accounting
service-policy flow_export_policy global

```

In this example, Secure Firewall ASA appliance is configured to send NetFlow packets to `172.29.142.27` on port `4729`. In addition, *flow-export* actions are enabled on *flow-create*, *flow-teardown*, *flow-denied*, and *flow-update* events. When these flow events occur on ASA, a NetFlow record is generated and sent to the destination specified in the configuration.

Assuming a Secure Firewall connector is enabled on Secure Workload and listening on `172.29.142.27:4729` in a Secure Workload Ingest appliance, the connector will receive NetFlow packets from Secure Firewall ASA appliance. The connector processes the NetFlow records as discussed in [Handling NSEL Events](#) and exports flow observations to Secure Workload. In addition, for NATed flows, the connector stitches the related flows (client-side and server-side) flows.

How to Configure the Connector

For information about required virtual appliances, see [Virtual Appliances for Connectors](#). The following configurations are allowed on the connector.

- *Log*: For more information, see [Log Configuration](#).

In addition, the listening ports of IPFIX protocol on the connector can be updated on the Docker container in Secure Workload Ingest appliance using a an allowed command. This command can be issued on the appliance by providing the connector ID of the connector, type of the port to be update, and the new port information. The connector ID can be found on the connector page in Secure Workload UI. For more information, see [update-listening-ports](#).

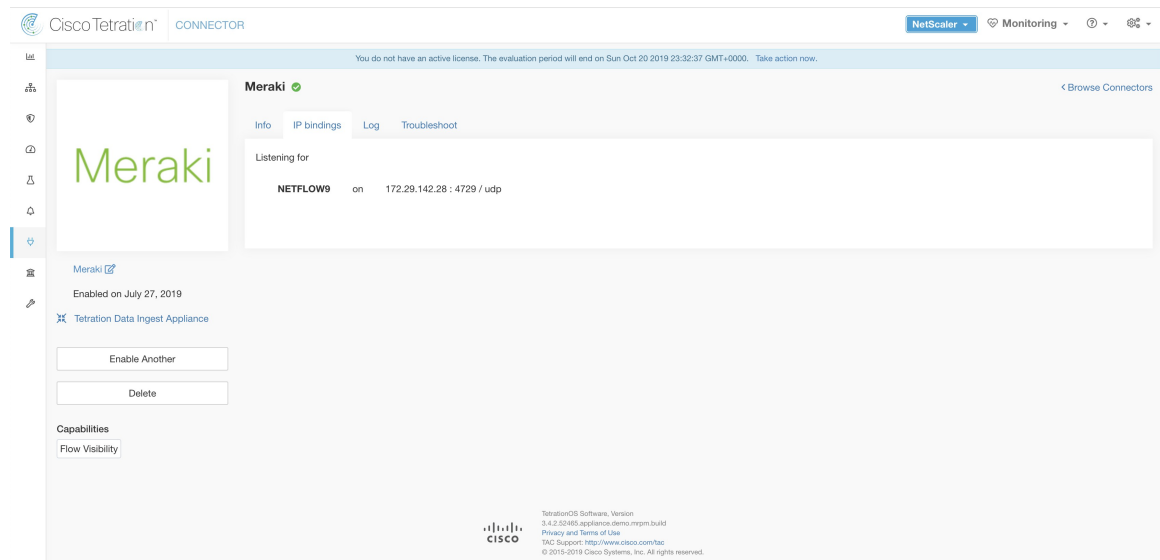
Limits

Metric	Limit
Maximum number of Secure Firewall connectors on one Secure Workload Ingest appliance	1
Maximum number of Secure Firewall connectors on one Tenant (rootscope)	10
Maximum number of Secure Firewall connectors on Secure Workload	100

Meraki Connector

Meraki connector allows Secure Workload to ingest flow observations from Meraki firewalls (included in Meraki MX security appliances and wireless access points). Using this solution, the hosts do not need to run software agents, because the Cisco switches will relay NetFlow records to Meraki connector hosted in a Secure Workload Ingest appliance for processing.

Figure 8: Meraki connector



What is NetFlow

NetFlow protocol allows network devices such as [Meraki Firewall](#) to aggregate traffic that passes through them into flows and export these flows to a flow collector. The flow collector receives these flow records and stores them in their flow storage for offline querying and analysis.

Typically, the setup involves the following steps:

1. Enable NetFlow statistics reporting on Meraki Firewall.
2. Configure the NetFlow collector endpoint information on Meraki Firewall.

Flow Ingestion to Secure Workload

Meraki connector is essentially a NetFlow collector. The connector receives the flow records from the Meraki firewalls that are configured to export NetFlow traffic statistics. It processes the NetFlow records and sends the flow observations reported by Meraki firewalls to Secure Workload for flow analysis. A Meraki connector can be enabled on a Secure Workload Ingest appliance and runs as a Docker container.

Meraki connector also registers with Secure Workload as a Secure Workload Meraki agent. Meraki connector decapsulates the NetFlow protocol packets (i.e., flow records); then processes and reports the flows like a regular Secure Workload agent. Unlike a Deep Visibility Agent, it does not report any process or interface information.



Note Meraki connector supports NetFlow v9 protocol.



Note Each Meraki connector should report only flows for one VRF. The flows exported by the connector is put in the VRF based on the Agent VRF configuration in Secure Workload cluster. To configure the VRF for the connector, go to: **Manage > Agents** and click the **Configuration** tab. In this page, under *Agent Remote VRF Configurations* section, click *Create Config* and provide the details about the connector. The form requests the user to provide: the name of the VRF, IP subnet of the connector, and range of port numbers that can potentially send flow records to the cluster.

Handling NetFlow Records

Based on the NetFlow record, Meraki connector sends flow observation to Secure Workload. Meraki NetFlow flow records are bidirectional. So, Meraki connector sends 2 flows: forward flow and reverse flow to Secure Workload.

Here are the details about flow observation sent by Meraki connector to Secure Workload.

Forward Flow observation

Field	Element ID	Element Name
Protocol	4	<i>protocolIdentifier</i>
Source Address	8	<i>sourceIPv4Address</i>
Source Port	7	<i>sourceTransportPort</i>
Destination Address	12	<i>destinationIPv4Address</i>
Destination Port	11	<i>destinationTransportPort</i>
Byte Count	1	<i>octetDeltaCount</i>
Packet Count	2	<i>packetDeltaCount</i>
Flow Start Time		Set based on when the NetFlow record for this flow is received on the connector

Reverse Flow Information

Field	Element ID	
Protocol	4	<i>protocolIdentifier</i>
Source Address	8	<i>sourceIPv4Address</i>
Source Port	7	<i>sourceTransportPort</i>
Destination Address	12	<i>destinationIPv4Address</i>
Destination Port	11	<i>destinationTransportPort</i>
Byte Count	23	<i>postOctetDeltaCount</i>

Field	Element ID	
Packet Count	24	<i>postPacketDeltaCount</i>
Flow Start Time		Set based on when the NetFlow record for this flow is received on the connector

How to configure NetFlow on Meraki Firewall

The following steps show how to configure NetFlow reporting on Meraki Firewall.

- Step 1** Login to Meraki UI console.
- Step 2** Navigate to **Network-wide > General**. In *Reporting* settings, enable **NetFlow traffic reporting** and make sure the value is set to *Enabled: send NetFlow traffic statistics*.
- Step 3** Set **NetFlow collector IP** and **NetFlow collector port** to the IP and port on which Meraki connector is listening in Secure Workload Ingest appliance. Default port on which Meraki connector listens for NetFlow records is 4729.
- Step 4** Save the changes.

Figure 9: Enabling NetFlow on a Meraki Firewall

The screenshot shows the Meraki UI console for a network named 'Woodstock'. The left sidebar shows the navigation menu with 'Network-wide' selected. The main content area displays the 'Reporting' settings. Under 'Reporting', the 'NetFlow traffic reporting' dropdown is set to 'Enabled: send netflow traffic statistics'. Below it, the 'NetFlow collector IP' field is empty and highlighted with a blue border, and the 'NetFlow collector port' field is also empty. A yellow warning box at the bottom right indicates 'You have unsaved changes.' with 'Save' and 'cancel' buttons.

How to Configure the Connector

For information about required virtual appliances, see [Virtual Appliances for Connectors](#). The following configurations are allowed on the connector.

- *Log*: For more information, see [Log Configuration](#).

In addition, the listening ports of NetFlow v9 protocol on the connector can be updated on the Docker container in Secure Workload Ingest appliance using an allowed command. This command can be issued on the appliance by providing the connector ID of the connector, type of the port to be update, and the new port information. The connector ID can be found on the connector page in Secure Workload UI. For more information, see [update-listening-ports](#).

Limits

Metric	Limit
Maximum number of Meraki connectors on one Secure Workload Ingest appliance	1
Maximum number of Meraki connectors on one Tenant (rootscope)	10
Maximum number of Meraki connectors on Secure Workload	100

ERSPAN Connector

ERSPAN connector allows Secure Workload to ingest flow observations from routers and switches in the network. Using this solution, the hosts do not need to run software agents, because the Cisco switches will relay the hosts' traffic to the ERSPAN connector for processing.

What is ERSPAN

Encapsulated Remote Switch Port Analyzer (ERSPAN) is a feature present in most of Cisco switches. It mirrors frames seen by a network device, encapsulates them in a IP packet and sends them to a remote analyzer. Users can select a list of interfaces and/or VLANs on the switch to be monitored.

Commonly, the setup involves configuring source ERSPAN monitoring session(s) on one or more network devices and configuring the destination ERSPAN monitoring session(s) on the remote network device(s) directly connected to a traffic analyzer.

The Secure Workload ERSPAN connector provides both the destination ERSPAN session and traffic analyzer functionalities; therefore there is no need to configure any destination sessions on the switches with the Secure Workload solution.

What are the SPAN Agents

Each ERSPAN connector registers a SPAN agent with the cluster. The Secure Workload SPAN agents are regular Secure Workload agents configured to only process ERSPAN packets: Like Cisco destination ERSPAN sessions, they decapsulate the mirrored frames; then they process and report the flows like a regular Secure Workload agent. Unlike Deep Visibility Agents, they do not report any process or interface information.

What is the Ingest Appliance for ERSPAN

The Secure Workload Ingest appliance for ERSPAN is a VM that internally runs three ERSPAN Secure Workload connectors. It uses the same OVA or QCOW2 as the normal Ingest appliance.

Each connector runs inside a dedicated Docker container to which one vNIC and two vCPU cores with no limiting quota are exclusively assigned.

The ERSPAN connector registers a SPAN agent with the cluster with the container hostname: <VM hostname>-<interface IP address>.

The connectors and agents are preserved/restored upon VM, Docker daemon or Docker container crash/reboot.



Note The ERSPAN connector's status will be reported back to the Connector page. See the Agent List page and check the corresponding SPAN agents state.

For more information about required virtual appliances, see [Virtual Appliances for Connectors](#). For ERSPAN connectors, IPv4 and IPv6 (dual stack mode) addresses are supported. However, do note that dual stack support is a BETA feature.

How to configure the source ERSPAN session

The following steps are for a Nexus 9000 switch. The configurations may slightly differ for other Cisco platforms. For configuring a Cisco platform, see the Cisco Secure Workload User Guide.

Figure 10: Configuring ERSPAN source on Cisco Nexus 9000

```

Enter the configuration mode
# config terminal

Configure the erspan source IP address
(config)# monitor erspan origin ip-address 172.28.126.1 global

Create and configure the source erspan session
(config)# monitor session 10 type erspan-source
(config-erspan-src)# source interface ethernet 1/23 both
(config-erspan-src)# source vlan 315, 512
(config-erspan-src)# destination ip 172.28.126.194

Turn on the monitor session
(config-erspan-src)# no shut

Persist the configuration
# copy runnin-config startup-confi

```

The above steps created a source ERSPAN session with id 10. The switch will mirror the frames ingressing and egressing (both) the interface eth1/23 and the ones on VLANS 315 and 512. The outer GRE packet carrying the mirrored frame will have source IP 172.28.126.1 (must be the address of a L3 interface on this switch) and destination IP 172.28.126.194. This is one of the IP addresses configured on the ERSPAN VM.

Supported ERSPAN formats

The Secure Workload SPAN Agents can process ERSPAN type I, II and III packets described in the proposed [ERSPAN RFC](#). Therefore they can process ERSPAN packets generated by Cisco devices. Among the non

RFC compliant formats, they can process the ERSPAN packets generated by VMware vSphere Distributed Switch (VDS).

Performance considerations when configuring ERSPAN source

Carefully choose the ERSPAN source's port/VLAN list. Although the SPAN agent has two dedicated vCPUs, the session may generate considerable amount of packets which could saturate the processing power of the agent. If an agent is receiving more packets than it can process, it will be shown in the Agent Packet Misses graph on the cluster's Deep Visibility Agent page.

More fine grained tuning on which frames the ERSPAN source will mirror can be achieved with ACL policies, usually via the filter configuration keyword.

If the switch supports it, the ERSPAN source session can be configured to modify the maximum transport unit (MTU) of the ERSPAN packet (commonly the default value 1500 bytes), usually via a mtu keyword. Decreasing it will limit the ERSPAN bandwidth usage in your network infrastructure, but it will have no effect on the SPAN Agent load, given the agent's workload is on a per-packet basis. When reducing this value, allow room for 160 bytes for the mirrored frame. For the ERSPAN header overhead details, see the proposed [ERSPAN RFC](#).

There are three versions of ERSPAN. The smaller the version, the lower the ERSPAN header overhead. Version II and III allow for applying QOS policies to the ERSPAN packets, and provide some VLAN info. Version III carries even more settings. Version II is usually the default one on Cisco switches. While Secure Workload SPAN Agents support all three versions, at the moment they do not make use of any extra information the ERSPAN version II and III packets carry.

Security considerations

The Ingest Virtual Machine for ERSPAN guest Operating System is CentOS 7.9, from which OpenSSL server/clients packages were removed.



Note CentOS 7.9 is the guest operating system for Ingest and Edge virtual appliances in Secure Workload 3.8.1.19 and earlier releases. Starting Secure Workload 3.8.1.36, the operating system is AlmaLinux 9.2.

Once the VM is booted and the SPAN agent containers are deployed (this takes a couple of minutes on first time boot only), no network interfaces, besides the loopback, will be present in the Virtual Machine. Therefore the only way to access the appliance is via its console.

The VM network interface are now moved inside the Docker containers. The containers run a centos:7.9.2009 based Docker image with no TCP/UDP port open.



Note Starting Secure Workload 3.8.1.36, the containers run almalinux/9-base:9.2.

Also, the containers are run with the base privileges (no `--privileged` option) plus the `NET_ADMIN` capability. In the unlikely case a container is compromised, the VM guest OS should not be compromisable from inside the container.

All the other security consideration valid for Secure Workload Agents running inside a host do also apply to the Secure Workload SPAN Agents running inside the Docker containers.

Troubleshooting

Once SPAN Agents show in active state in the cluster Monitoring/Agent Overview page, no action is needed on the ERSPAN Virtual Machine, user does not need to log into it. If that is not happening or if the flows are not reported to the cluster, following information will help pinpoint deployment problems.

In normal conditions, on the VM:

- `systemctl status tet_vm_setup` reports an *inactive* service with *SUCCESS* exit status;
- `systemctl status tet-nic-driver` reports an *active* service;
- `docker network ls` reports five networks: *host*, *none* and three *erspan-`<iface name>`*;
- `ip link` only reports the loopback interface;
- `docker ps` reports three running containers;
- `docker logs <cid>` for each container contains the message: `INFO success: tet-sensor entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)`
- `docker exec <cid> ifconfig` reports only one interface, besides the loopback;
- `docker exec <cid> route -n` reports the default gateway;
- `docker exec <cid> iptables -t raw -S PREROUTING` reports the rule `-A PREROUTING -p gre -j DROP`;

If any of the above does not hold true, check the deployment script logs in `/local/tetration/logs/tet_vm_setup.log` for the reason why the SPAN agent containers deployment failed.

Any other agent registration/connectivity issue can be troubleshooted the same way it is done for agents running on a host via the `docker exec` command:

- `docker exec <cid> ps -ef` reports the two `tet-engine`, `tet-engine check_conf` instances and two `/usr/local/tet/tet-sensor -f /usr/local/tet/conf/.sensor_config` instances, one with root user and one with `tet-sensor` user, along with the process manager `/usr/bin/python /usr/bin/supervisord -c /etc/supervisord.conf -n` instance.
- `docker exec <cid> cat /usr/local/tet/log/tet-sensor.log` shows the agent's logs;
- `docker exec <cid> cat /usr/local/tet/log/fetch_sensor_id.log` shows the agent's registration logs;
- `docker exec <cid> cat /usr/local/tet/log/check_conf_update.log` shows the configuration update polling logs;

If necessary, traffic to/from the container can be monitored with `tcpdump` after setting into the container's network namespace:

1. Retrieve the container's network namespace (`SandboxKey`) via `docker inspect <cid> | grep SandboxKey`;
2. Set into the container's network namespace `nsenter --net=/var/run/docker/netns/...`;
3. Monitor `eth0` traffic `tcpdump -i eth0 -n`.

Limits

Metric	Limit
Maximum number of ERSPAN connectors on one Secure Workload Ingest appliance	3
Maximum number of ERSPAN connectors on one Tenant (rootscope)	24 (12 for TaaS)
Maximum number of ERSPAN connectors on Secure Workload	450

Connectors for Endpoints

Connectors for endpoints provide endpoint context for Secure Workload.

Connector	Description	Deployed on Virtual Appliance
AnyConnect	Collect telemetry data from Cisco AnyConnect Network Visibility Module (NVM) and enrich endpoint inventories with user attributes	Secure Workload Ingest
ISE	Collect information about endpoints and inventories managed by Cisco ISE appliances and enrich endpoint inventories with user attributes and secure group labels (SGL).	Secure Workload Edge

For more information about required virtual appliances, see [Virtual Appliances for Connectors](#).

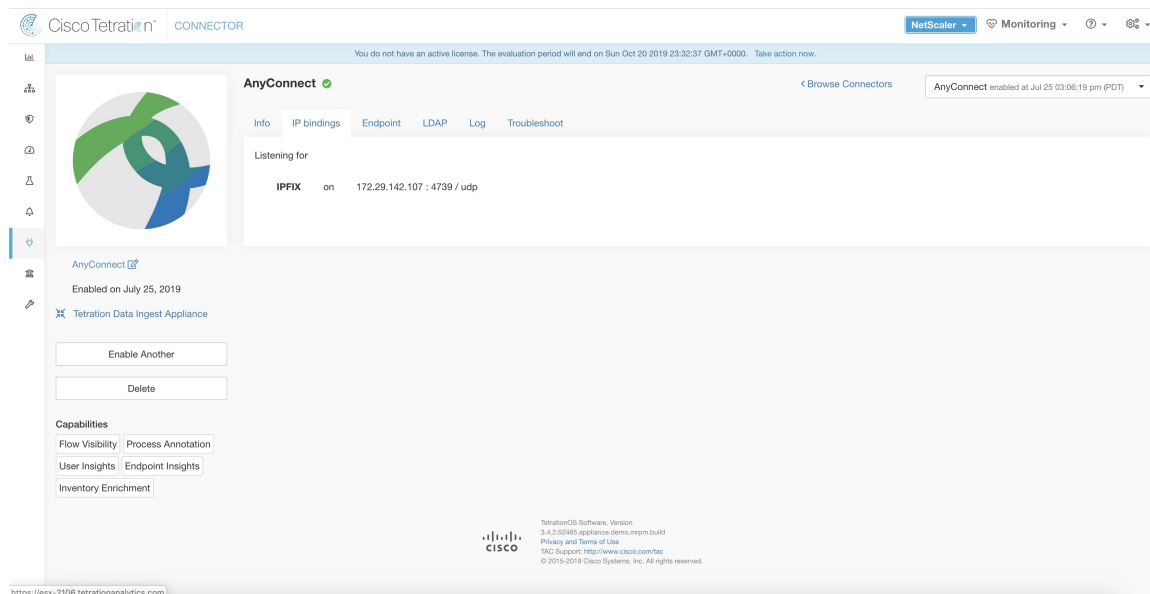
AnyConnect Connector

AnyConnect connector monitors endpoints that run [Cisco AnyConnect Secure Mobility Client](#) with [Network Visibility Module \(NVM\)](#). Using this solution, the hosts do not need to run any software agents on endpoints, because NVM sends host, interface, and flow records in IPFIX format to a collector (e.g., AnyConnect connector).

AnyConnect connector does the following high-level functions.

1. Register each endpoint (supported user devices such as a desktop, a laptop, or a smartphone) on Cisco Secure Workload as an AnyConnect agent.
2. Update interface snapshots from these endpoints with Secure Workload.
3. Send flow information exported by these endpoints to Secure Workload collectors.
4. Periodically send process snapshots for processes that generate flows on the endpoints tracked by the AnyConnect connector.
5. Label endpoint interface IP addresses with Lightweight Directory Access Protocol (LDAP) attributes corresponding to the logged-in-user at each endpoint.

Figure 11: AnyConnect connector



What is AnyConnect NVM

AnyConnect NVM provides visibility and monitoring of endpoint and user behavior both on and off premises. It collects information from endpoints that includes the following context.

1. **Device/Endpoint Context:** Device/endpoint specific information.
2. **User Context:** Users associated with the flow.
3. **Application Context:** Processes associated with the flow.
4. **Location Context:** Location specific attributes -if available.
5. **Destination Context:** FQDN of the destination. AnyConnect NVM generates 3 types of records.

NVM Record	Description
Endpoint Record	Device/endpoint information including unique device identifier (UDID), hostname, OS name, OS version and manufacturer.
Interface Record	Information about each interface in the endpoint including the endpoint UDID, interface unique identifier (UID), interface index, interface type, interface name, and MAC address.
Flow Record	Information about flows seen on the endpoint including endpoint UDID, interface UID, 5-tuple (source/destination ip/port and protocol), in/out byte counts, process information, user information, and fqdn of the destination.

Each record is generated and exported in IPFIX protocol format. When the device is in a trusted network (on-premise/VPN), AnyConnect NVM exports records to a configured collector. AnyConnect connector is an example IPFIX collector that can receive and process IPFIX stream from AnyConnect NVM.



Note AnyConnect connector supports AnyConnect NVM from 4.2+ versions of Cisco AnyConnect Secure Mobility Client.

How to configure AnyConnect NVM

See [How to Implement AnyConnect NVM](#) document for step by step instructions on how to implement AnyConnect NVM using either [Cisco Secure Firewall ASA](#) or [Cisco Identity Services engine \(ISE\)](#). Once NVM module is deployed, an NVM profile should be specified and pushed to and installed on the endpoints running Cisco AnyConnect Secure Mobility Client. When specifying NVM profile, the IPFIX collector should be configured to point to AnyConnect connector on port 4739.

AnyConnect connector also registers with Secure Workload as a Secure Workload AnyConnect Proxy agent.

Processing NVM records

AnyConnect connector processes AnyConnect NVM records as shown below.

Endpoint Record

Upon receiving an endpoint record, AnyConnect connector registers that endpoint as AnyConnect agent on Secure Workload. AnyConnect connector uses the endpoint specific information present in the NVM record along with AnyConnect connector's certificate to register the endpoint. Once an endpoint is registered, data-plane for the endpoint is enabled by creating a new connection to one of the collectors in Secure Workload. Based on the activity (flow records) from this endpoint, AnyConnect connector checks-in the AnyConnect agent corresponding to this endpoint with the cluster periodically (20-30 minutes).

AnyConnect NVM starts to send agent version from 4.9. By default, the AnyConnect endpoint would be registered as version 4.2.x on Secure Workload. This version indicates the minimum supported AnyConnect NVM version. For the AnyConnect endpoints with version 4.9 or newer, the corresponding AnyConnect agent on Secure Workload would show the actual version installed.



Note The AnyConnect agent installed version is not controlled by Secure Workload. Attempting to upgrade the AnyConnect endpoint agent on Secure Workload UI would not take effect.

Interface Record

Interface Record IP address for an interface is not part of the AnyConnect NVM interface record. IP address for an interface is determined when flow records start coming from the endpoint for that interface. Once IP address is determined for an interface, AnyConnect connector sends a complete snapshot of all interfaces of that endpoint whose IP address is determined to config server of Secure Workload. This associates the VRF with the interface data and flows coming in on these interfaces will now be marked with this VRF.

Flow Record

Upon receiving a flow record, AnyConnect connector translates the record to the format that Secure Workload understands and sends FlowInfo over the dataplane corresponding to that endpoint. Furthermore, it stores process information included in the flow record locally. In addition, if LDAP configuration is provided to AnyConnect connector, it determines values for configured LDAP attributes of the logged-in-user of the endpoint. The attributes are associated to the endpoint IP address where the flow happened. Periodically, process information and user labels are pushed to Secure Workload.



Note Each AnyConnect connector will report only endpoints/interfaces/ flows for one VRF. The endpoints and interfaces reported by AnyConnect connector are associated with the VRF based on the Agent VRF configuration in Secure Workload. The flows exported by the AnyConnect connector agent on behalf of the AnyConnect endpoint belong to the same VRF. To configure the VRF for the agent, go to: **Manage > Agents** and click the **Configuration** tab. In this page, under “Agent Remote VRF Configurations” section, click “Create Config” and provide the details about the AnyConnect connector. The form requests the user to provide: the name of the VRF, IP subnet of the host on which the agent is installed, and range of port numbers that can potentially send flow records to the cluster.

Duplicate UDIDs in Windows Endpoints

If endpoint machines are cloned from the same golden image, it is possible that the UDID of all cloned endpoints are identical. In such cases, AnyConnect connector receives endpoint records from these endpoints with identical UDID and registers them on Secure Workload with same UDID. When interface/flow records are received by the connector from these endpoints, it is impossible for the connector to determine the correct AnyConnect agent on Secure Workload to associate the data. The connector associates all the data to one endpoint (and it is not deterministic).

To deal with this problem, AnyConnect NVM 4.8 release ships a tool called *dartcli.exe* to find and regenerate UDID on the endpoint.

- *dartcli.exe -u* retrieves the UDID of the endpoint.
- *dartcli.exe -nu* regenerates the UDID of the endpoint. To run this tool, use the following steps.

```
C:\Program Files (x86)\Cisco\Cisco AnyConnect Secure Mobility Client\DART>dartcli.exe
-u
UDID : 8D0D1E8FA0AB09BE82599F10068593E41EF1BFFF

C:\Program Files (x86)\Cisco\Cisco AnyConnect Secure Mobility Client\DART>dartcli.exe
-nu
Are you sure you want to re-generate UDID [y/n]: y
Adding nonce success
UDID : 29F596758941E606BD0AFF49049216ED5BB9F7A5

C:\Program Files (x86)\Cisco\Cisco AnyConnect Secure Mobility Client\DART>dartcli.exe
-u
UDID : 29F596758941E606BD0AFF49049216ED5BB9F7A5
```

Periodic Tasks

Periodically, AnyConnect connector sends process snapshots and user labels on AnyConnect endpoint inventories.

1. **Process Snapshots:** every 5 minutes, AnyConnect connector walks through the processes it maintains locally for that interval and sends process snapshot for all the endpoints that had flows during that interval.
2. **User Labels:** every 2 minutes, AnyConnect connector walks through the LDAP user labels it maintains locally and updates User Labels on those IP addresses.

For user labels, AnyConnect connector creates a local snapshot of LDAP attributes of all users in the organization. When AnyConnect connector is enabled, configuration for LDAP (server/port information, attributes to fetch for a user, attribute that contains the username) may be provided. In addition, the LDAP user credentials to access LDAP server may be provided. LDAP user credentials are encrypted and never revealed in the AnyConnect connector. Optionally, an LDAP certificate may be provided for securely accessing LDAP server.



Note AnyConnect connector creates a new local LDAP snapshot every 24 hours. This interval is configurable in LDAP configuration of the connector.

How to Configure the Connector

For information about required virtual appliances, see [Virtual Appliances for Connectors](#). The following configurations are allowed on the connector.

- *LDAP:* LDAP configuration supports discovery of LDAP attributes and provide a workflow to pick the attribute that corresponds to username and a list of up to 6 attributes to fetch for each user. For more information, see [Discovery](#).
- *Endpoint:* For more information, see [Endpoint Configuration](#).
- *Log:* For more information, see [Log Configuration](#).

In addition, the listening ports of IPFIX protocol on the connector can be updated on the Docker container in Secure Workload Ingest appliance using an allowed command. This command can be issued on the appliance by providing the connector ID of the connector, type of the port to be update, and the new port information. The connector ID can be found on the connector page in Secure Workload UI. For more information, see [update-listening-ports](#).

Limits

Metric	Limit
Maximum number of AnyConnect connectors on one Secure Workload Ingest appliance	1
Maximum number of AnyConnect connectors on one Tenant (rootscope)	50
Maximum number of AnyConnect connectors on Secure Workload	500

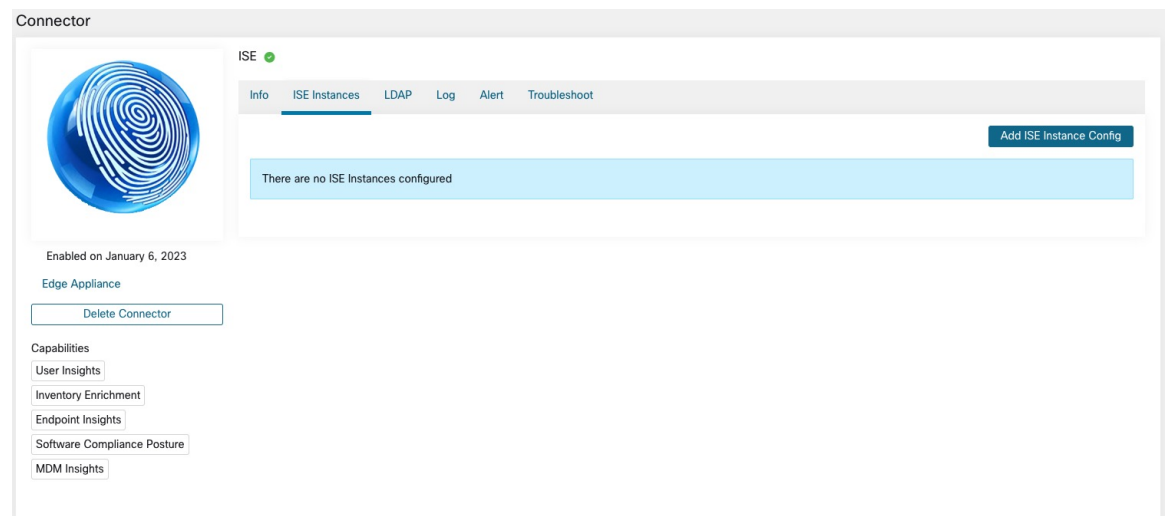
ISE Connector

ISE connector in Secure Workload connects with [Cisco Identity Services Engine \(ISE\)](#), using [Cisco Platform Exchange Grid \(pxGrid\)](#), to retrieve contextual information about endpoints reported by ISE. Using these solutions, we can obtain enriched metadata for endpoints.

An ISE connector performs these functions:

1. Register each endpoint viewed by ISE on Secure Workload as an ISE endpoint agent.
2. Update metadata information regarding these endpoints to Secure Workload including MDM details, authentication, Security Group labels, and others.
3. Periodically take a snapshot and update cluster with active endpoints visible on ISE.

Figure 12: ISE connector



Note Each ISE connector will register only endpoints and interfaces for one VRF. The endpoints and interfaces reported by ISE connector are associated with the VRF based on the Agent VRF configuration in Secure Workload. To configure the VRF for the agent, go to: **Manage > Workloads > Agents** and click the **Configuration** tab. In this page, under the **Agent Remote VRF Configurations** section, click **Create Config** and provide the details about the ISE connector. The form requests the user to provide: the name of the VRF, IP subnet of the host on which the agent is installed, and range of port numbers that can potentially register ISE endpoints and interfaces on Secure Workload.



Note The ISE endpoint agents are not listed on the Agents List page; instead ISE endpoints with the attributes can be viewed on the Inventory page.

How to Configure the Connector



Note ISE version 2.4+ is required for this integration.

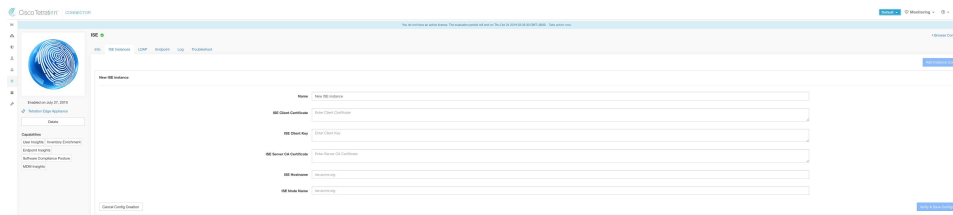
For information about required virtual appliances, see [Virtual Appliances for Connectors](#). For ISE connectors, IPv4 and IPv6 (dual stack mode) addresses are supported. However, do note that dual stack support is a BETA feature.

The following configurations are allowed on the connector.

- **ISE Instance:** ISE connector can connect to multiple instances of ISE using provided configurations. Each instance requires ISE certificate credentials along with hostname and nodename to connect to ISE. For more information, see [ISE Instance Configuration](#).
- **LDAP:** LDAP configuration supports discovery of LDAP attributes and provides a workflow to select the attribute that corresponds to username and a list of up to six attributes to fetch for each user. For more information, see [Discovery](#).
- **Endpoint:** For more information, see [Endpoint Configuration](#).
- **Log:** For more information, see [Endpoint Configuration](#).

ISE Instance Configuration

Figure 13: ISE instance config



Note Starting Cisco Secure Workload version 3.7, the SSL certificate for Cisco ISE pxGrid node requires Subject Alternative Names (SAN) for this integration. Ensure the certification configuration of the ISE nodes is done by your ISE administrator prior to performing the integration with Secure Workload.

To verify your pxGrid node's certificate and confirm if SAN is configured, you need to do the following to verify the certificate from ISE.


- Step 1** Go to **Certificates** under **Administration > System**.
- Step 2** Under **Certificate Management**, select **System Certificates**, select your "Used by" pxGrid certificate and choose **View** to review the pxGrid node cert.
- Step 3** Scroll the certificate and ensure the Subject Alternative Names are configured for this certificate.
- Step 4** This certificate should be signed by a valid Certificate Authority (CA), which should also be used to sign the pxGrid client certificate used for the Secure Workload ISE connector.

Figure 14: Example of a valid ISE pxGrid node

Certificate Hierarchy □

ca. [REDACTED].com

ce-ise27. [REDACTED]



ce-ise27. [REDACTED]

Issued By : ca. [REDACTED].com

Expires : Fri, 2 Aug 2024 19:19:37 UTC

Certificate status is good

Organization Unit (OU) **Tetration Engineering**

Organization (O) **SBG**

City (L) **San Jose**

State (ST) **California**

Country (C) **US**

Serial Number [REDACTED] **C0:C2:03:1B:D5:80:57:00:00:00:00:00:0C**

Subject Alternative Names **IP:172.[REDACTED], IP:1[REDACTED], DNS:ce-ise27.[REDACTED], DNS:ce-ise27.[REDACTED]**

Close

certificate

Step 5 You can now generate the pxGrid client certificate signing request using the following template on any host installed with OpenSSL.

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = YOUR_COUNTRY
ST = YOUR_STATE
L = YOUR_CITY
O = YOUR_ORGANIZATION
OU = YOUR_ORGANIZATION_UNIT
CN = ise-connector.example.com
[v3_req]
subjectKeyIdentifier = hash
```



```
basicConstraints = critical,CA:false
subjectAltName = @alt_names
keyUsage = critical,digitalSignature,keyEncipherment
extendedKeyUsage = serverAuth,clientAuth
[alt_names]
IP.1 = 10.x.x.x
DNS.1 = ise-connector.example.com
```

Save the file as 'example-connector.cfg' and use the OpenSSL command from your host to generate a Certificate Signing Request (CSR) and the certificate private key with the following command.

```
openssl req -newkey rsa:2048 -keyout example-connector.key -nodes -out example-connector.csr -config
example-connector.cfg
```

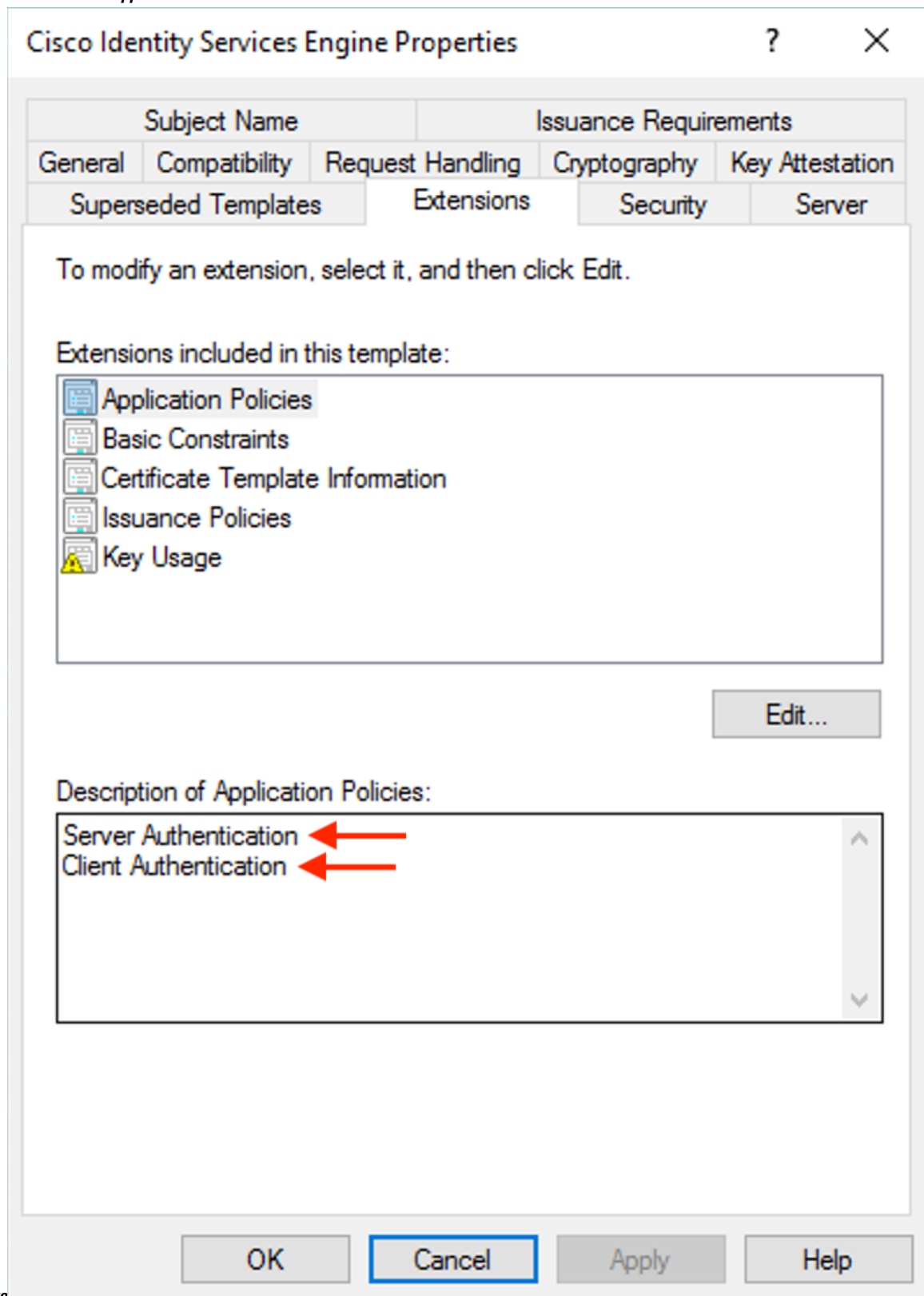
Step 6

Sign the Certificate Signing Request (CSR) by your CA using a Windows CA server. If you are also using a Windows CA server, run the following command to sign the pxGrid client's CSR.

```
certreq -submit -binary -attrib "CertificateTemplate:CiscoIdentityServicesEngine" example-connector.csr
example-connector.cer
```

Note Windows CA requires a Certificate Template. This template should contain the following extensions.

Figure 15: Extensions of Application Policies for a Certificate



Template

Step 7 Copy the signed client certificate and the root CA in PEM format onto your host. This is the same host that generates the client CSR and the private key. Use OpenSSL to ensure the client certificate is in X.509 PEM format. Run the following command using OpenSSL to convert the signed client certificate to the X.509 PEM format.

```
openssl x509 -inform der -in example-connector.cer -out example-connector.pem
```

Step 8 You can also confirm the PEM that is signed by the CA, use the following command.

```
openssl verify -CAfile root-ca.example.com.pem example-connector.pem
example-connector.pem: OK
```

Note For multi-node ISE deployment with pxGrid, all the pxGrid nodes must trust the Certs used for the Secure Workload ISE Connector.

Step 9 Using the above example's file names, copy the ISE client cert - example-connector.pem, client key - example-connector.key and CA – root-ca.example.com.pem into the respective fields on the ISE configuration page on Secure Workload as shown below.

Figure 16: ISE Connector

The screenshot shows the 'New ISE Instance' configuration page in the ISE management console. The page includes a sidebar with a fingerprint icon and a 'Delete' button. The main content area has a 'Name' field with a red prompt 'Enter ISE instance name'. Below it are three certificate fields: 'ISE Client Certificate' (prompt: 'Copy and paste ISE client certificate'), 'ISE Client Key' (prompt: 'Copy and paste client key. This should be clear key, not password protected.'), and 'ISE Server CA Certificate' (prompt: 'Copy and paste RootCA certificate'). There are also 'ISE Hostname' and 'ISE Node Name' fields with prompts 'Enter ISE hostname(FQDN)' and 'Enter ISE nodename' respectively. A 'Cancel Config Creation' button is at the bottom left, and a 'Verify & Save Configs' button is at the bottom right.

configuration



Note

- If an IP Address is used instead of FQDN for the ISE Hostname, then use the IP address in the ISE CA certificate SAN, else, there may be connection failures.
- Number of active endpoints on ISE is not a snapshot, it depends on the configurations on ISE and the aggregation duration for computing the metric. The agent count on Secure Workload is always a snapshot based on last pull from ISE and pxgrid updates, typically the active device count over last one day (default refresh frequency for full snapshots is a day). Due to the difference in the way these numbers are depicted, it is possible that these two numbers will not always match.

Processing ISE records

ISE connector processes records as described below.

Endpoint Record

ISE connector connects to ISE instance and subscribes for any updates for endpoints over pxGrid. Upon receiving an endpoint record, ISE connector registers that endpoint as ISE agent on Secure Workload. ISE connector uses the endpoint specific information present in endpoint record along with ISE connector's certificate to register the endpoint. Once an endpoint is registered, ISE connector uses the endpoint object for inventory enrichment by sending this as user labels on Secure Workload. When ISE connector gets a disconnected endpoint from ISE, it deletes the inventory enrichment from Secure Workload.

Security Group Record

ISE connect also subscribes for updates about Security Group Labels change via pxGrid. On receiving this record, ISE connectors maintains a local database. It uses this database to map SGT name with value on receiving an endpoint record.

Periodic Tasks

ISE connector periodically shares user labels on ISE endpoint inventories.

1. **Endpoint Snapshots:** Every 20 hours, ISE connector fetches a snapshot of endpoints and security group labels from ISE instance and updates the cluster if any change is detected. This call does not compute for endpoints that are disconnected in case we do not see endpoints on Secure Workload coming from ISE.
2. **User Labels:** Every 2 minutes, ISE connector scans through the LDAP user and ISE endpoint labels maintained locally and updates user labels on those IP addresses.

For user labels, ISE connector creates a local snapshot of LDAP attributes of all users in the organization. When ISE connector is enabled, configuration for LDAP (server/port information, attributes to fetch for a user, attribute that contains the username) may be provided. In addition, the LDAP user credentials to access LDAP server may be provided. LDAP user credentials are encrypted and never revealed in the ISE connector. Optionally, an LDAP certificate may be provided for securely accessing LDAP server.



Note ISE connector creates a new local LDAP snapshot every 24 hours. This interval is configurable in LDAP configuration of the connector.



Note On upgrading Cisco ISE device, ISE connector will need to be re-configured with new certificates generated by ISE after upgrade.

Limits

Metric	Limit
Maximum number of ISE instances that can be configured on one ISE connector	20

Metric	Limit
Maximum number of ISE connectors on one Secure Workload Edge appliance	1
Maximum number of ISE connectors on one Tenant (rootscope)	1
Maximum number of ISE connectors on Secure Workload	150



Note Maximum number of ISE agents supported per connector is 20000. If there is a use case that requires support for more ISE agents, please contact Secure Workload support.

Connectors for Inventory Enrichment

Connectors for inventory enrichment provides additional meta-data and context about the inventories (IP addresses) monitored by Secure Workload.

Connector	Description	Deployed on Virtual Appliance
ServiceNow	Collect endpoint information from ServiceNow instance and enrich the inventory with ServiceNow attributes.	Secure Workload Edge
See also:	Cloud Connectors	–

For more information about required virtual appliances, see [Virtual Appliances for Connectors](#).

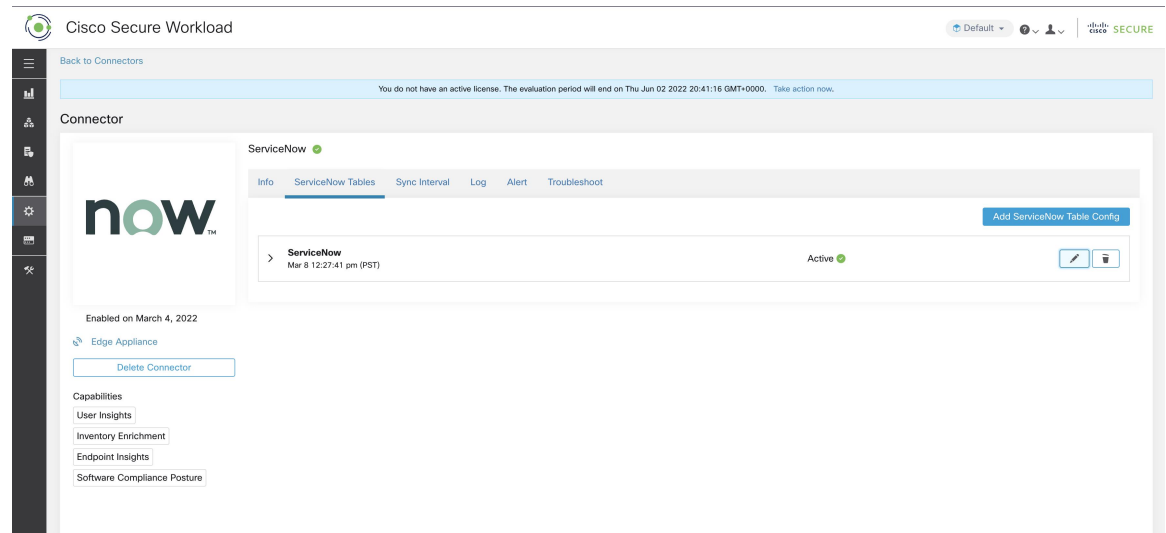
ServiceNow Connector

ServiceNow connector connects with [ServiceNow Instance](#) to get all the ServiceNow CMDB related labels for the endpoints in ServiceNow inventory. Using this solution, we can get enriched metadata for the endpoints in Cisco Secure Workload.

ServiceNow connector does the following high-level functions.

1. Update ServiceNow metadata in Secure Workload's inventory for these endpoints.
2. Periodically take snapshot and update the labels on these endpoints.

Figure 17: ServiceNow connector



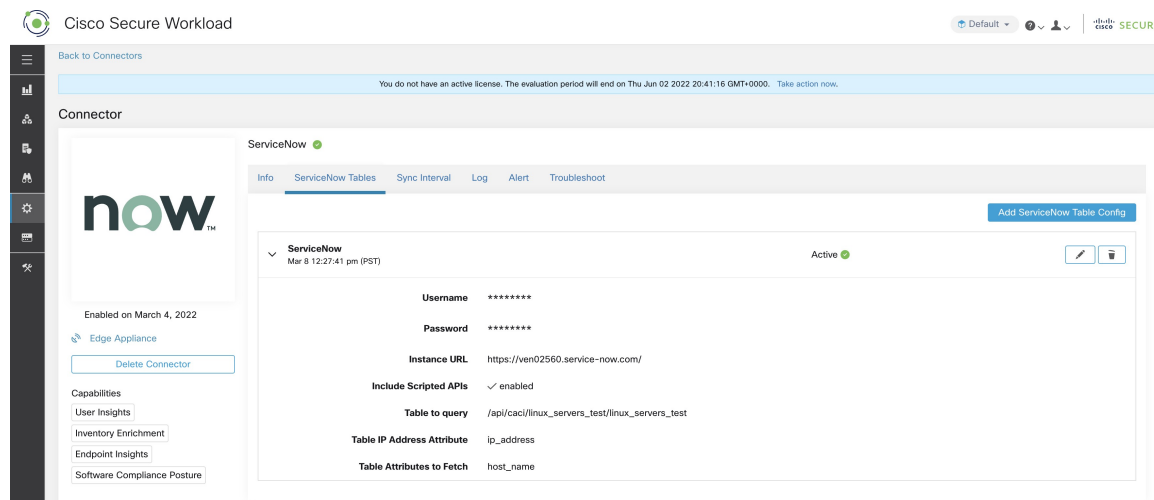
How to Configure the ServiceNow Connector

For information about required virtual appliances, see [Virtual Appliances for Connectors](#). The following configurations are allowed on the connector.

- *ServiceNow Tables*: ServiceNow Tables configures the ServiceNow instance with its credentials, and the information about ServiceNow tables to fetch the data from.
- *Scripted REST api*: [ServiceNow scripted REST API](#) tables can be configured similar to ServiceNow tables.
- *Sync Interval*: Sync Interval configuration allows to make change the periodicity at which Secure Workload should query ServiceNow instance for updated data.
- *Log*: For more information, see [Log Configuration](#).

ServiceNow Instance Configuration

Figure 18: ServiceNow instance config



You will need the following items to successfully configure a ServiceNow instance.

1. ServiceNow username
2. ServiceNow password
3. ServiceNow Instance URL
4. Include Scripted APIs

Subsequently, Secure Workload performs a discovery of all the tables from the ServiceNow Instance and Scripted REST API's (only if Include Scripted APIs checkbox is enabled). It presents user with the list of tables to choose from, once a user selects table, Secure Workload fetches all the list of attributes from that table for the user to select. User has to choose the ip_address attribute from the table as the key. Subsequently, user can choose up to 10 unique attributes from the table. See the following figures for each step.



Note ServiceNow Connector can only support integrating with tables having **IP Address** field.



Note To integrate with ServiceNow Scripted REST API's you need to enable the Scripted APIs checkbox, which would give you a similar workflow to any other table.



Note For Scripted REST API's to integrate with ServiceNow Connector, they cannot have path parameters. Also, they need to support **sysparm_limit, sysparm_fields and sysparm_offset** as query parameters.



Note The ServiceNow user roles need to include **cmdb_read** for tables and **web_service_admin** for Scripted REST API's to integrate with Cisco Secure Workload.

Figure 19: ServiceNow instance config first step

Figure 20: Secure Workload Fetches the Table Info from ServiceNow Instance

Figure 21: Secure Workload presents the list of tables

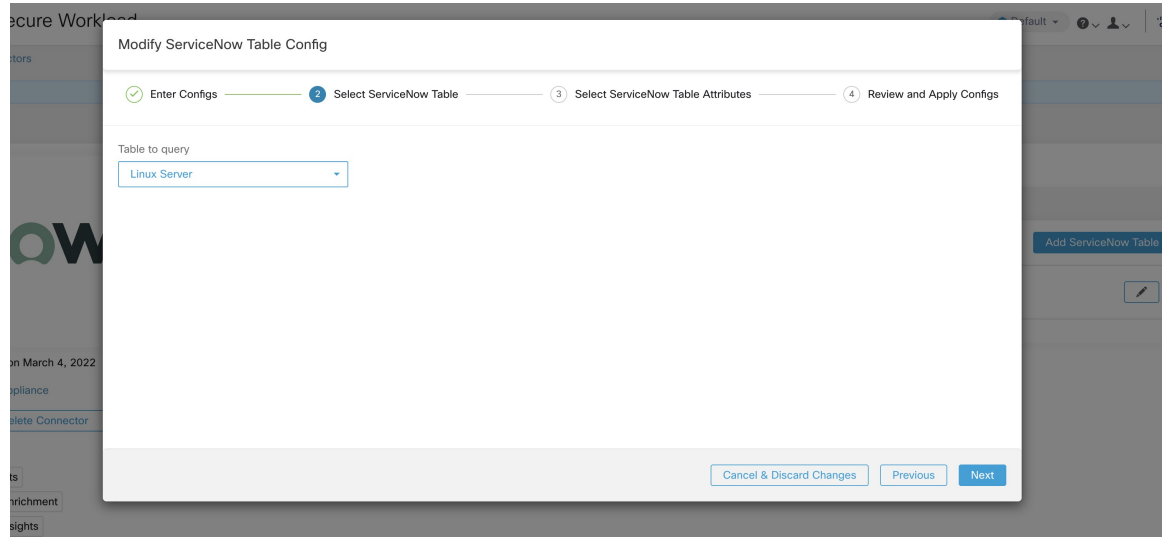


Figure 22: User selects the ip_address attribute, and other attribute in the table

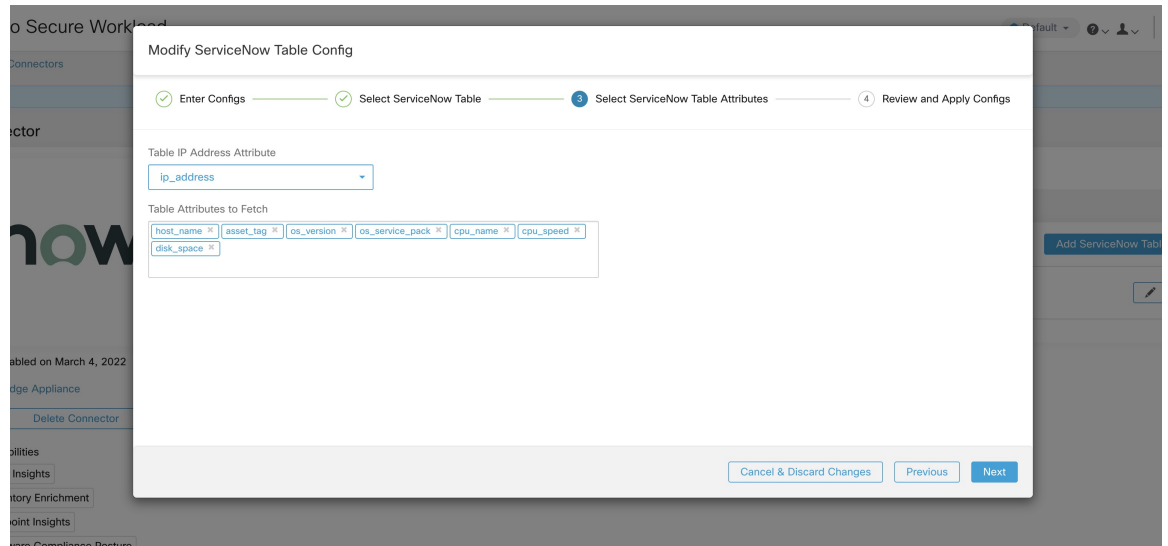


Figure 23: User finalizes the ServiceNow config

Processing ServiceNow records

Based on the instance url you gives in configuration, ServiceNow connector connects to ServiceNow Instance. ServiceNow Instance uses HTTP calls using `https://{Instance URL}/api/now/doc/table/schema`, to obtain the initial table schema from the ServiceNow Table API. Based on the configured Tables, it queries those tables to fetch the ServiceNow labels/metadata. Secure Workload annotates the ServiceNow labels to IP addresses in its inventory. ServiceNow connector periodically fetches new labels and updates Secure Workload inventory.



Note Secure Workload fetches records from ServiceNow tables periodically. This is configurable under SyncInterval tab in the ServiceNow connector. The default sync interval is 60 minutes. For cases where integrating with ServiceNow table with large number of entries, this sync interval should be set to a higher value.



Note Secure Workload will delete any entry not seen for 10 continuous sync intervals. In case the connection to ServiceNow instance is down for that long that could result in cleaning up of all labels for that instance.

Sync Interval Configuration

1. Secure Workload ServiceNow connector provides a way to configure the frequency of sync between Secure Workload and ServiceNow instance. By default the sync interval is set to 60 minutes, but it can be changed under the sync interval configuration as **Data fetch frequency**.
2. For detecting deletion of a record, Secure Workload ServiceNow connector relies on syncs from ServiceNow instances. If an entry is not seen in 48 consecutive sync intervals, we go ahead and delete the entry. This can be configured under sync interval config as **Delete entry interval**.

3. If any additional parameters are to be passed when calling REST api's for ServiceNow tables, you can configure them as part of *Additional Rest API url params*. This configuration is optional. For example, to get a reference lookup from ServiceNow the following url parameters can be used **sysparm_exclude_reference_link=true&sysparm_display_value=true**

Figure 24: Sync Interval Configuration

The screenshot shows the Cisco Secure Workload interface for the ServiceNow connector. The 'Sync Interval' tab is selected, showing the following configuration:

Parameter	Value
Data fetch frequency (in minutes)	60
Delete entry interval (in multiple of fetch frequency)	48
Additional Rest API url params	sysparm_exclude_reference_link=true&sysparm_display_value=true

Other visible details include the connector name 'ServiceNow', a 'Delete Connector' button, and a list of capabilities: User Insights, Inventory Enrichment, Endpoint Insights, and Software Compliance Posture.

Explore Command to Delete the Labels

In case user wants to cleanup the labels for a particular IP for a given instance immediately, without waiting for delete interval, they can do so using an explore command. Here are the steps to run the command.

1. Finding vrf ID for a Tenant
2. Getting to Explore command UI
3. Running the commands

For TaaS cluster, contact TaaS Operation team to cleanup labels for ServiceNow labels.

Finding VRF ID for a Tenant

Site Admins and **Customer Support** users can access the **Tenant** page under the **Platform** menu in the navigation bar at the left side of the window. This page displays all of the currently configured Tenants and VRFs. For more information, see the Tenants section for more details.

On Tenants page, ID field of `Tenants` table is vrf ID for the Tenant.

Getting to Explore Command UI

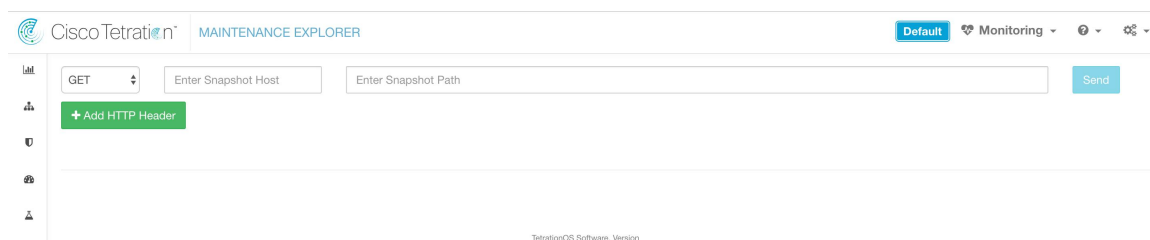
To reach the Maintenance Explorer command interface, choose **Troubleshoot > Maintenance Explorer** from the left navigation bar in the Secure Workload web interface.



Note Customer Support privileges are required to access explore menu. If explore tab does not show up, the account may not have needed permissions.

Click on explore tab in the drop down menu to get to the Maintenance Explorer page.

Figure 25: Maintenance Explorer tab



Running the Commands

- Choose the action as `POST`
- Enter snapshot host as `orchestrator.service.consul`
- Enter snapshot path

To delete the labels for a particular IP for a `servicenow` instance:

```
servicenow_cleanup_annotations?args=<vrf-id> <ip_address> <instance_url> <table_name>
```

- Click Send



Note If after deleting using explore command, we see the record show up in ServiceNow instance, it will be repopulated

Frequently Asked Questions

1. What if ServiceNow CMDB table does not have IP address.

In such case, the recommendation is to create a [View on ServiceNow](#) which will have desired fields from current table along with IP address (potentially coming from a JOIN operation with another table). Once such a view is created, it can be used in place of table name.

2. What if ServiceNow instance requires MFA.

Currently we do not support integrating with ServiceNow instance with MFA.

Limits

Metric	Limit
Maximum number of ServiceNow instances that can be configured on one ServiceNow connector	20
Maximum number of attributes that can be fetched from one ServiceNow instance	10
Maximum number of ServiceNow connectors on one Secure Workload Edge appliance	1
Maximum number of ServiceNow connectors on one Tenant (rootscope)	1
Maximum number of ServiceNow connectors on Secure Workload	150

Connectors for Alert Notifications

Connectors for alert notifications enable Secure Workload to publish Secure Workload alerts on various messaging and logging platforms. These connectors run on TAN service on Secure Workload Edge Appliance.

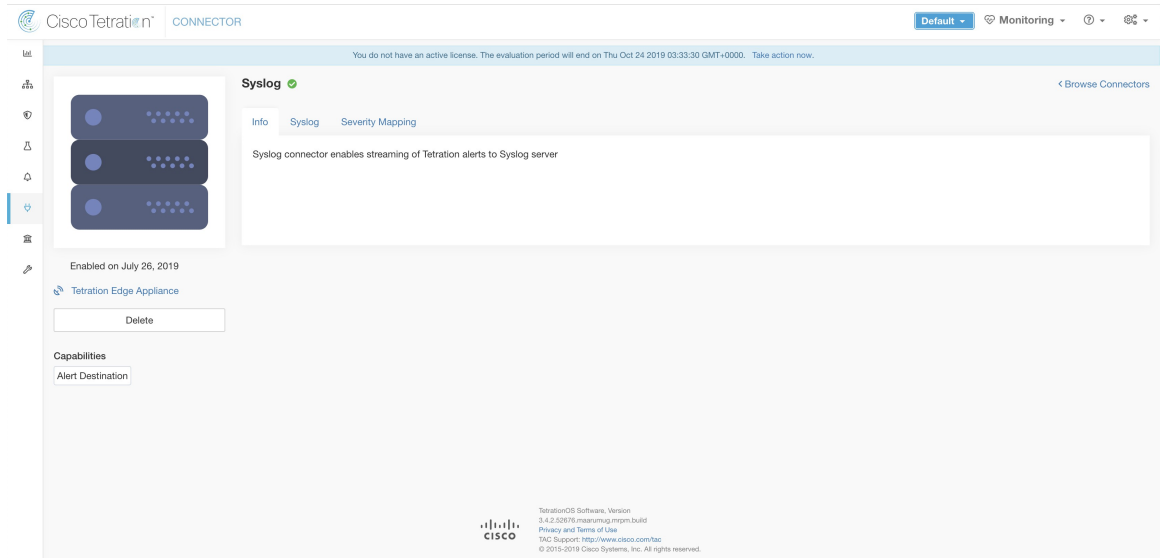
Connector	Description	Deployed on Virtual Appliance
Syslog	Send Secure Workload alerts to Syslog server.	Secure Workload Edge
Email	Send Secure Workload alerts on Email.	Secure Workload Edge
Slack	Send Secure Workload alerts on Slack.	Secure Workload Edge
Pager Duty	Send Secure Workload alerts on Pager Duty.	Secure Workload Edge
Kinesis	Send Secure Workload alerts on Amazon Kinesis.	Secure Workload Edge

For more information about required virtual appliances, see [Virtual Appliances for Connectors](#).

Syslog Connector

When enabled, TAN service on Cisco Secure Workload Edge appliance can send alerts to Syslog server using configuration.

Figure 26: Syslog connector



The following table explains the configuration details for publishing Secure Workload alerts on Syslog server. For more information, see [Syslog Notifier Configuration](#).

Parameter Name	Type	Description
Protocol	dropdown	Protocol to use to connect to server
	• UDP	
	• TCP	
Server Address	string	IP address or hostname of the Syslog server
Port	number	Listening port of Syslog server. Default port value is 514.

Figure 27: Sample configuration for Syslog Connector

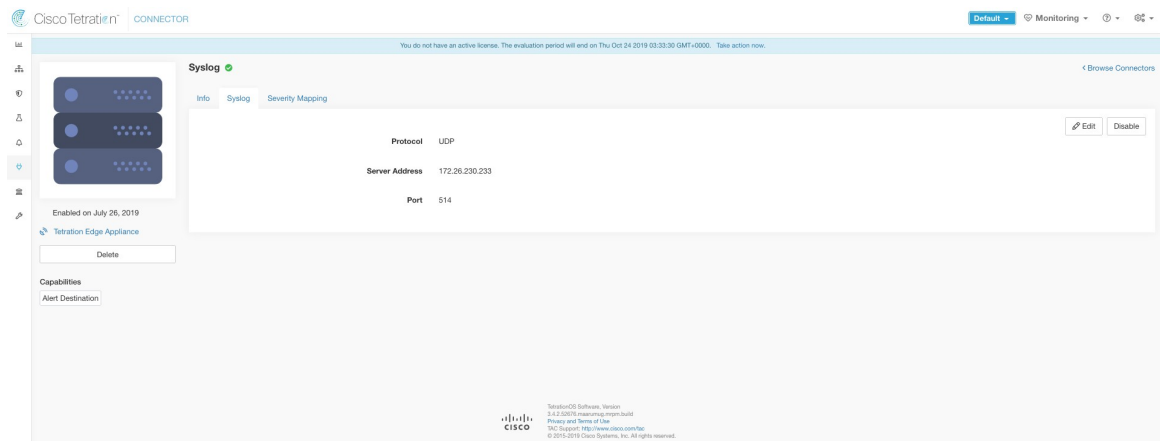


Figure 28: Sample alert

```

Jul 28 21:53:06 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [CRIT] {"keyId": "cfec2a77-5a8e-3b1d-8b07-f2a2971a1211", "eventTime": "1564350600000", "alertTime": "1564350820334", "alertText": "Enforcement Annotated Flows contains escaped for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "CRITICAL", "tenantId": "0", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "UDP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "policy_violations", "value": "contains"}, "value": "escaped"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350600000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "0", "application_id": "5d3b41c14974f01fac220a", "escaped_count": "2"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b41fbd91577e0895eb08"}
Jul 28 21:53:06 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [CRIT] {"keyId": "4e497e64-cab7-3e0b-9e68-62a6cc549e9f", "eventTime": "1564350600000", "alertTime": "1564350820334", "alertText": "Enforcement Annotated Flows contains escaped for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "CRITICAL", "tenantId": "0", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "TCP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "policy_violations", "value": "contains"}, "value": "escaped"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350600000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "25", "application_id": "5d3b41c14974f01fac220a", "escaped_count": "2"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b41fbd91577e0895eb08"}
Jul 28 21:54:22 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [CRIT] {"keyId": "3ba07063-8065-3a6d-9792-25a6f81819c0", "eventTime": "1564350720000", "alertTime": "1564350900881", "alertText": "Enforcement Annotated Flows contains escaped for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "CRITICAL", "tenantId": "0", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "TCP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "policy_violations", "value": "contains"}, "value": "escaped"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350720000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "443", "application_id": "5d3b41c14974f01fac220a", "escaped_count": "8"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b41fbd91577e0895eb08"}
Jul 28 21:54:22 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [DEBUG] {"keyId": "4c800807-2a3f-3253-afba-b9660c0d059b", "eventTime": "1564350720000", "alertTime": "1564350900881", "alertText": "Enforcement Rejected Flows u003e-1 for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "LOW", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "TCP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "rejected_count", "value": "gt"}, "value": "-1"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350720000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "443", "application_id": "5d3b41c14974f01fac220a", "rejected_count": "0"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b5234e9d1574832a7125e"}
Jul 28 21:54:22 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [DEBUG] {"keyId": "c961d0f8-c182-3e75-a997-493e954b638", "eventTime": "1564350720000", "alertTime": "1564350900881", "alertText": "Enforcement Rejected Flows u003e-1 for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "LOW", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "UDP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "rejected_count", "value": "gt"}, "value": "-1"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350720000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "53", "application_id": "5d3b41c14974f01fac220a", "rejected_count": "0"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b5234e9d1574832a7125e"}
Jul 28 21:54:22 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [DEBUG] {"keyId": "da0a22d-9bab-380b-83a3-4106e6b75f31", "eventTime": "1564350720000", "alertTime": "1564350900881", "alertText": "Enforcement Rejected Flows u003e-1 for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "LOW", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "UDP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "rejected_count", "value": "gt"}, "value": "-1"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350720000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "53", "application_id": "5d3b41c14974f01fac220a", "rejected_count": "0"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b5234e9d1574832a7125e"}
Jul 28 21:54:22 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [CRIT] {"keyId": "cfec2a77-5a8e-3b1d-8b07-f2a2971a1211", "eventTime": "1564350720000", "alertTime": "1564350900881", "alertText": "Enforcement Annotated Flows contains escaped for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "CRITICAL", "tenantId": "0", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "UDP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "policy_violations", "value": "contains"}, "value": "escaped"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350720000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "0", "application_id": "5d3b41c14974f01fac220a", "escaped_count": "2"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b41fbd91577e0895eb08"}
Jul 28 21:54:22 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [CRIT] {"keyId": "ad6e167-c2d9-336f-b465-b6d6eb46f6d", "eventTime": "1564350720000", "alertTime": "1564350900881", "alertText": "Enforcement Annotated Flows contains escaped for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "CRITICAL", "tenantId": "0", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "UDP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "policy_violations", "value": "contains"}, "value": "escaped"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350720000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "123", "application_id": "5d3b41c14974f01fac220a", "escaped_count": "2"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b41fbd91577e0895eb08"}
Jul 28 21:54:22 tan-5d3b5191f786e9752c77031 Tetratlon Alert[4235]: [CRIT] {"keyId": "ca083366-8c2b-3192-4a07-60364d3dbd", "eventTime": "1564350720000", "alertTime": "1564350900881", "alertText": "Enforcement Annotated Flows contains escaped for u003capplication_id:5d3b41c14974f01fac220a0u003e", "severity": "CRITICAL", "tenantId": "0", "type": "COMPLIANCE", "alertDetails": {"provider_scope_ids": ["5d3a744e4974f446636ff13"], "consumer_scope_ids": ["5d3a744e4974f446636ff13"], "protocol": "UDP", "policy_type": "ENFORCED_POLICY", "internal_trigger": {"datasource": "compliance", "rules": {"field": "policy_violations", "value": "contains"}, "value": "escaped"}, "label": "Alert Trigger"}, "consumer_scope_names": ["Default"], "provider_scope_names": ["Default"], "time_range": ["1564350720000", "1564350799999"], "policy_category": ["ESCAPED"], "provider_port": "53", "application_id": "5d3b41c14974f01fac220a", "escaped_count": "2"}, "rootScopeId": "5d3a744e4974f446636ff13", "alertConfId": "5d3b41fbd91577e0895eb08"}

```

Syslog Severity Mapping

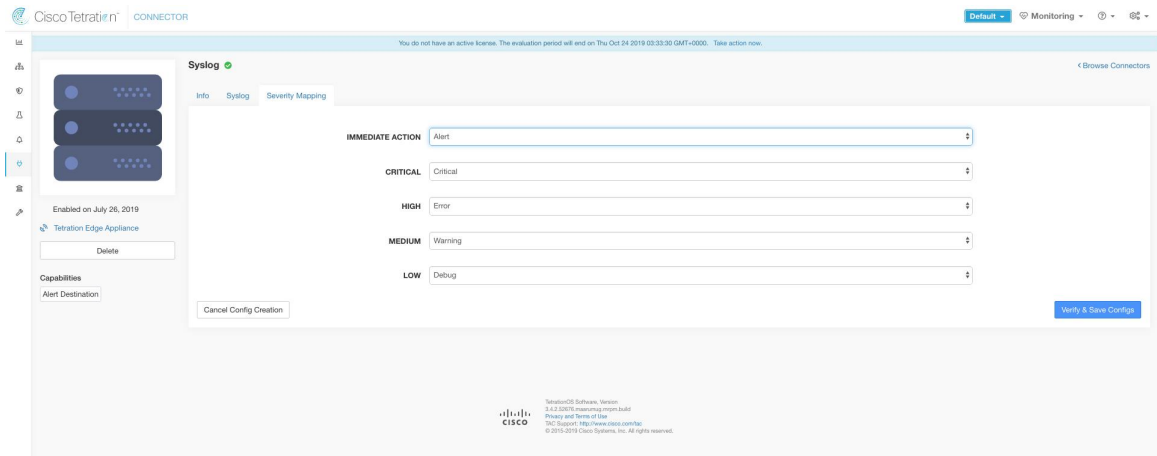
The following table shows the default severity mapping for Secure Workload alerts on Syslog.

Secure Workload Alerts Severity	Syslog Severity
LOW	LOG_DEBUG
MEDIUM	LOG_WARNING
HIGH	LOG_ERR
CRITICAL	LOG_CRIT
IMMEDIATE ACTION	LOG_EMERG

This setting can be modified using **Severity Mapping** configuration under Syslog Connector. You can choose any corresponding Syslog priority for each Secure Workload Alert Severity and change the Severity Mapping. For more information, see [Syslog Severity Mapping Configuration](#).

Parameter Name	Dropdown of mappings
IMMEDIATE_ACTION	• <i>Emergency</i>
CRITICAL	• <i>Alert</i>
HIGH	• <i>Critical</i>
MEDIUM	• <i>Error</i>
LOW	• <i>Warning</i>
	• <i>Notice</i>
	• <i>Informational</i>
	• <i>Debug</i>

Figure 29: Sample config for Syslog Severity Mapping.



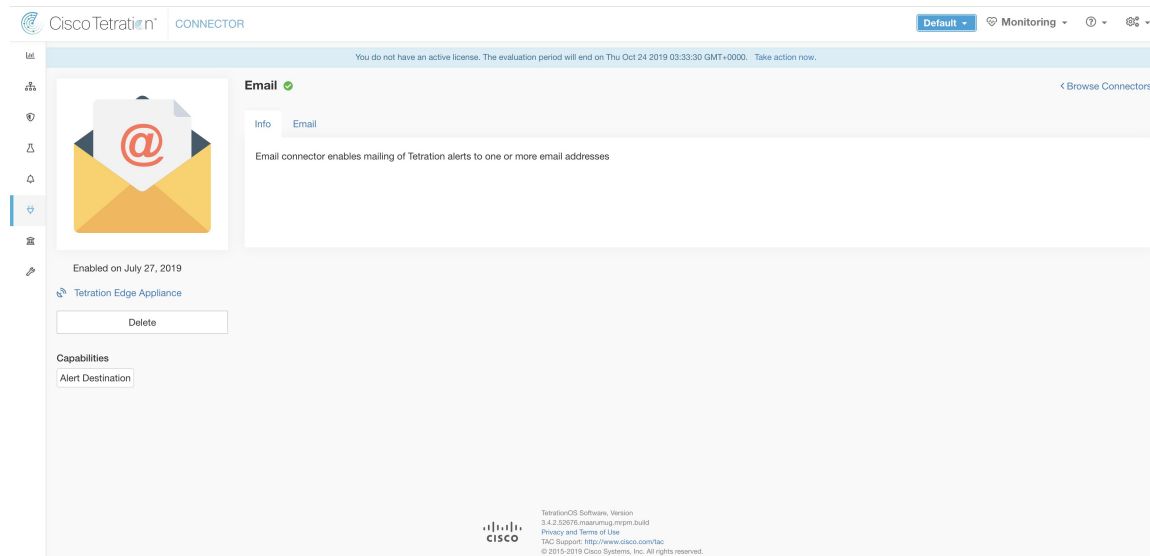
Limits

Metric	Limit
Maximum number of Syslog connectors on one Secure Workload Edge appliance	1
Maximum number of Syslog connectors on one Tenant (rootscope)	1
Maximum number of Syslog connectors on Secure Workload	150

Email Connector

When enabled, TAN service on Secure Workload Edge Appliance can send alerts to given configuration.

Figure 30: Email connector



The following table explains the configuration details for publishing Secure Workload alerts on Email. For more information, see [Email Notifier Configuration](#).

Table 2: Email Notifier Configuration for more details

Parameter Name	Type	Description
SMTP Username	string	SMTP server username. This parameter is optional.
SMTP Password	string	SMTP server password for the user (if given). This parameter is optional.
SMTP Server	string	IP address or hostname of the SMTP server
SMTP Port	number	Listening port of SMTP server. Default value is 587.
Secure Connection	checkbox	Should SSL be used for SMTP server connection?
From Email Address	string	Email address to use for sending alerts
Default Recipients	string	Comma separated list of recipient email addresses

Figure 31: Sample configuration for Email Connector

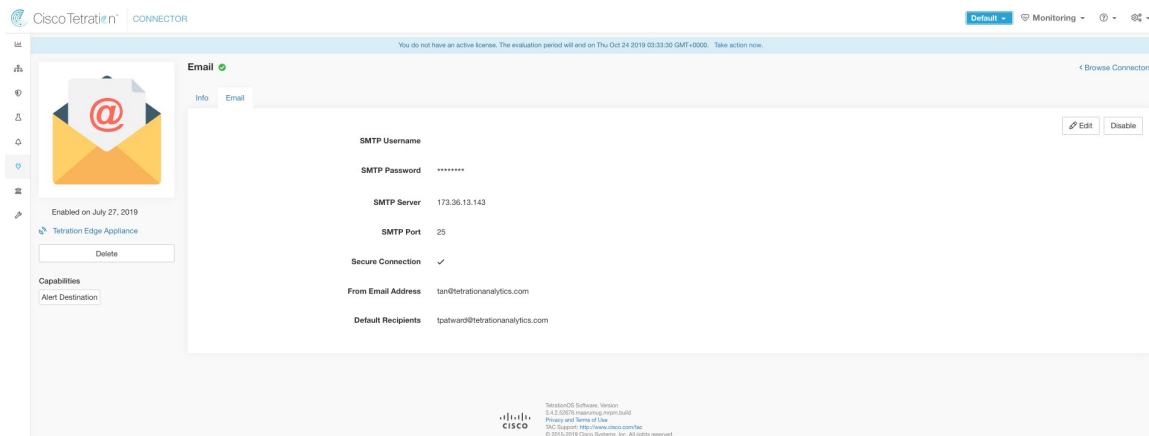


Figure 32: Sample alert

**Note**

- SMTP username/password is optional. If no username is provided, we try to connect to SMTP server without any auth.
- If secure connection box is not checked, we will send alerts notification over non-secure connection.
- Default Recipients list is used to send alert notifications. This can be overridden per alert if required in Alert configuration.

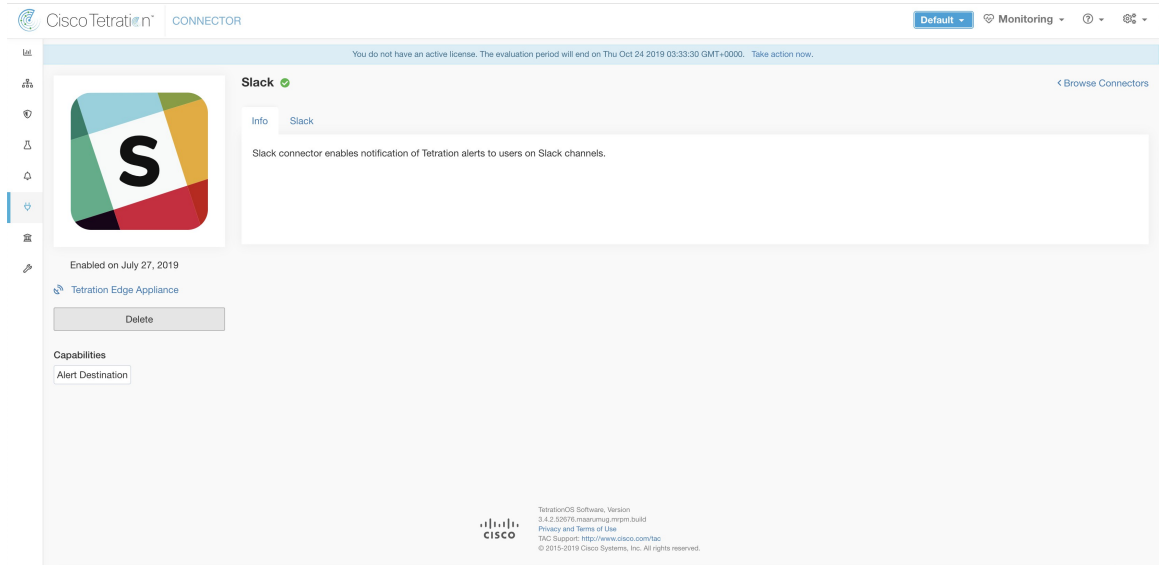
Limits

Metric	Limit
Maximum number of Email connectors on one Secure Workload Edge appliance	1
Maximum number of Email connectors on one Tenant (rootscope)	1
Maximum number of Email connectors on Secure Workload	150

Slack Connector

When enabled, TAN service on Secure Workload Edge appliance can send alerts to Slack using configuration.

Figure 33: Slack connector



The following table explains the configuration details for publishing Secure Workload alerts on Slack. For more information, see [Slack Notifier Configuration](#).

Parameter Name	Type	Description
Slack Webhook URL	string	Slack webhook on which Secure Workload alerts should be published



Note • To generate slack webhook go [here](#).

Figure 34: Sample configuration for Slack Connector

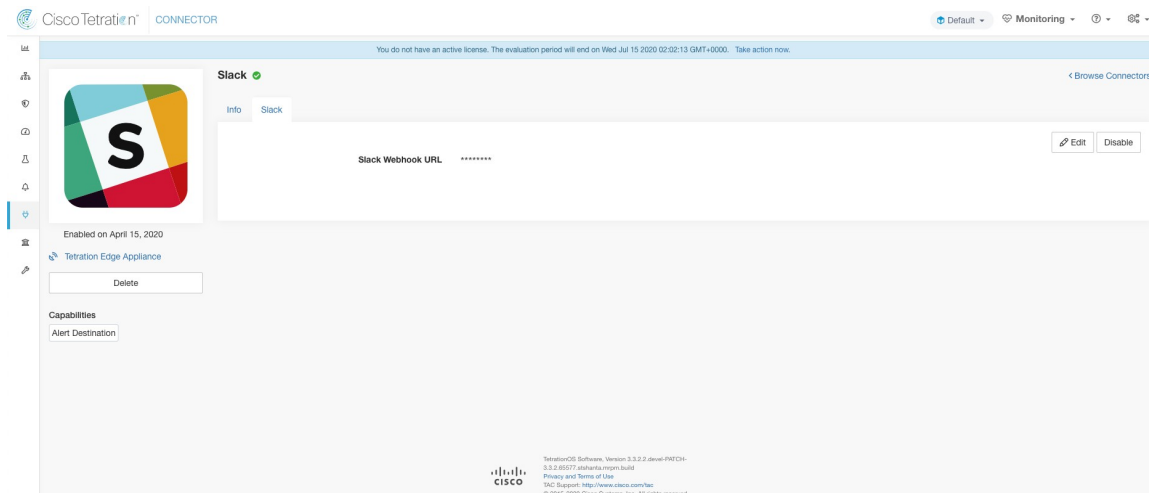
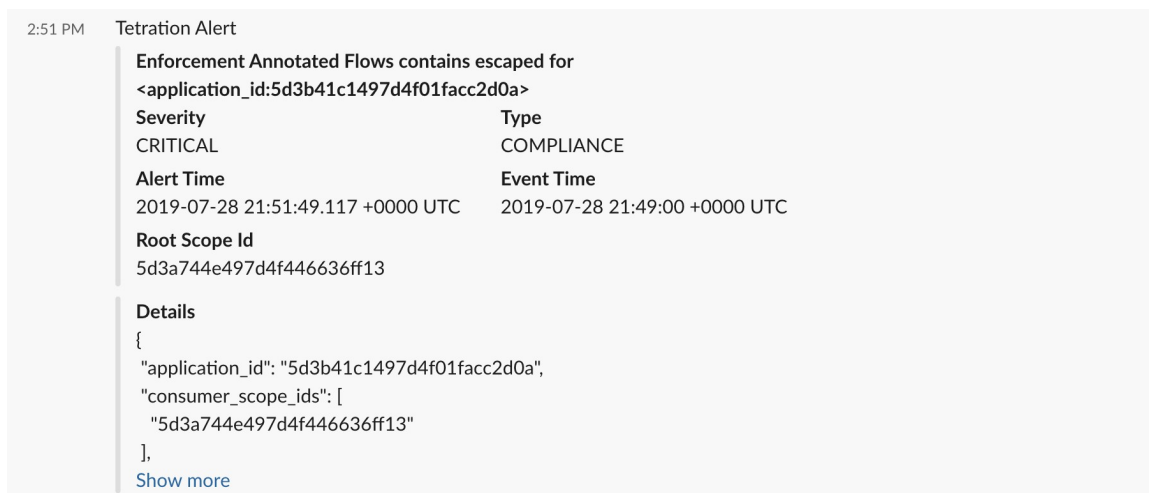


Figure 35: Sample alert



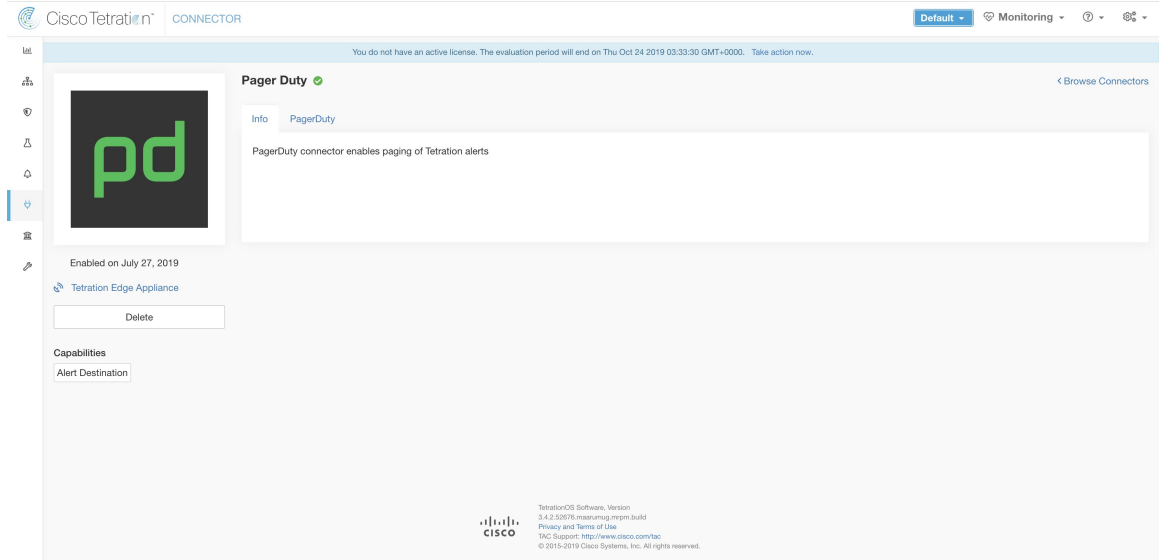
Limits

Metric	Limit
Maximum number of Slack connectors on one Secure Workload Edge appliance	1
Maximum number of Slack connectors on one Tenant (rootscope)	1
Maximum number of Slack connectors on Secure Workload	150

PagerDuty Connector

When enabled, TAN service on Secure Workload Edge appliance can send alerts to PagerDuty using configuration.

Figure 36: PagerDuty connector



The following table explains the configuration details for publishing Secure Workload alerts on PagerDuty. For more information, see [PagerDuty Notifier Configuration](#).

Parameter Name	Type	Description
PagerDuty Service Key	string	PagerDuty service key for pushing Secure Workload alerts on PagerDuty.

Figure 37: Sample configuration for PagerDuty Connector

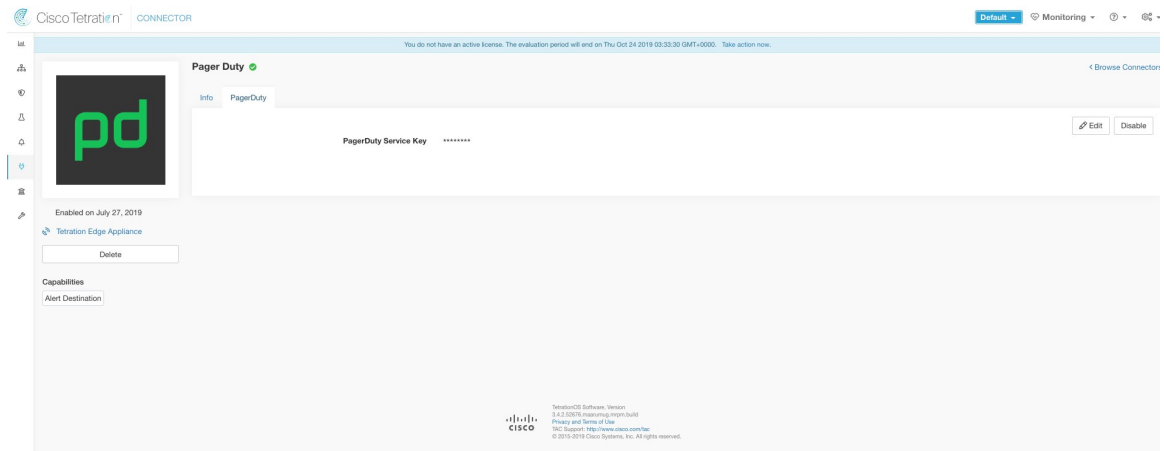


Figure 38: Sample alert

INCIDENTS > INCIDENT #408756

Enforcement Annotated Flows contains escaped for <application_id:5d3b41c1497d4f01facc2d0a>

1 alert

! Acknowledge Reassign More Actions

Alerts Timeline Similar Incidents

ALERTS
1 triggered

FILTERS: No active table filters Per Page: 25 1-1 of 1

Resolve Customize Columns

Status	Severity	Summary	Created	Service
Triggered	Critical	Enforcement Annotated Flows contains escaped for <application_id:5d3b41c1497d4f01facc2d0a>	at 2:58 PM	TanDemo

HIDE DETAILS

CUSTOM DETAILS

Alert Details

```

HIDE DETAILS
{"provider_scope_ids":["5d3a744e497d4f446636ff13"],"consumer_scope_ids":["5d3a744e497d4f446636ff13"],"protocol":"ICMP","policy_type":"ENFORCED_POLICY","internal_trigger":{"datasource":"compliance","rules":{"field":"policy_violations","type":"contains","value":"escaped"},"label":"Alert Trigger"},"consumer_scope_names":["Default"],"provider_scope_names":["Default"],"time_range":["1564350900000,1564350959999"],"policy_category":["ESCAPED"],"provider_port":0,"application_id":"5d3b41c1497d4f01facc2d0a","escaped_count":2}

```

View Message

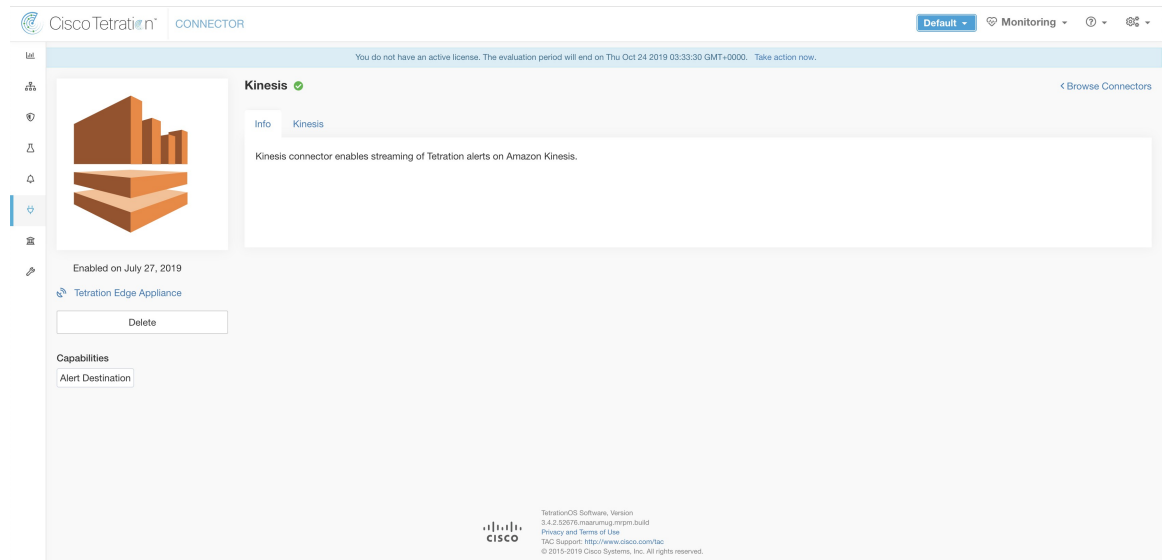
Limits

Metric	Limit
Maximum number of PagerDuty connectors on one Secure Workload Edge appliance	1
Maximum number of PagerDuty connectors on one Tenant (rootscope)	1
Maximum number of PagerDuty connectors on Secure Workload	150

Kinesis Connector

When enabled, TAN service on Secure Workload Edge appliance can send alerts using configuration.

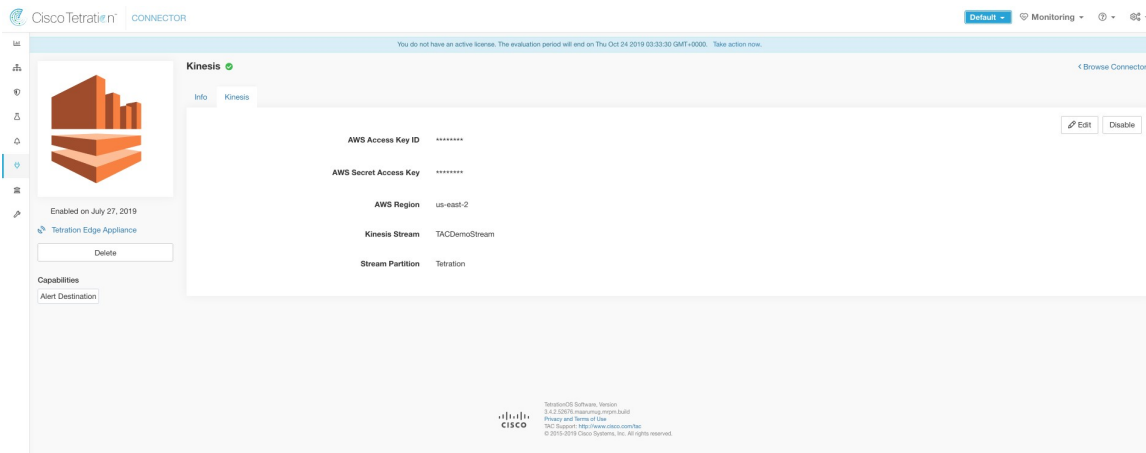
Figure 39: Kinesis connector



The following table explains the configuration details for publishing Secure Workload alerts on Amazon Kinesis. For more information, see [Kinesis Notifier Configuration](#).

Parameter Name	Type	Description
AWS Access Key ID	string	AWS access key ID to communicate with AWS
AWS Secret Access Key	string	AWS secret access key to communicate with AWS
AWS Region	dropdown of AWS regions	Name of the AWS region where Kinesis stream is configured
Kinesis Stream	string	Name of the Kinesis stream
Stream Partition	string	Partition Name of the stream

Figure 40: Sample configuration for Kinesis Connector.



Limits

Metric	Limit
Maximum number of Kinesis connectors on one Secure Workload Edge appliance	1
Maximum number of Kinesis connectors on one Tenant (rootscope)	1
Maximum number of Kinesis connectors on Secure Workload	150

Cloud Connectors

You can use a cloud connector for Secure Workload features on cloud-based workloads.

Cloud connectors do not require a virtual appliance.

Connector	Supported Features	Deployed on Virtual Appliance
AWS	For Amazon Web Services VPCs: <ul style="list-style-type: none"> Collect metadata (labels) Collect flow logs Enforce segmentation policies From Elastic Kubernetes Service (EKS) clusters: <ul style="list-style-type: none"> Collect metadata 	N/A

Connector	Supported Features	Deployed on Virtual Appliance
Azure	For Azure VNets: <ul style="list-style-type: none"> • Collect metadata (labels) • Collect flow logs • Enforce segmentation policies From Azure Kubernetes Service (AKS) clusters: <ul style="list-style-type: none"> • Collect metadata 	N/A
GCP	For Google Cloud Platform VPCs: <ul style="list-style-type: none"> • Collect metadata (labels) • Collect flow logs • Enforce segmentation policies From Google Kubernetes Engine (GKE) clusters: <ul style="list-style-type: none"> • Collect metadata (labels) 	N/A

AWS Connector

Amazon Web Services (AWS) connector connects with [AWS](#) to perform the following high-level functions:

- Automated ingestion of inventory (and its labels) live from an AWS Virtual Private Cloud (VPC)**
 AWS allows you to assign metadata to your resources in the form of tags. Secure Workload query the tags for these resources which can then be used for inventory and traffic flow data visualization, and policy definition. This capability keeps the resource tag mapping updated by constantly synchronizing this data.

 The tags from workloads and network interfaces of an AWS VPC are ingested. If you configure both workloads and network interfaces, Secure Workload merges and displays the tags. For more information, see [Labels Generated by Cloud Connectors](#).
- Ingestion of VPC-level flow logs** If you have set up VPC flow logs in AWS for monitoring purposes, Secure Workload can ingest flow log information by reading the corresponding S3 bucket. You can use this telemetry for visualization and segmentation policy generation.
- Segmentation** When the segmentation option is enabled, Secure Workload programs security policies using AWS native Security Groups. When enforcement is enabled for a VPC, relevant policies are automatically programmed as security groups.
- Automated ingestion of metadata from EKS clusters** When Elastic Kubernetes Services (EKS) is running on AWS, you can choose to gather all node, service, and pod metadata related to all selected Kubernetes clusters.

You can choose which capabilities to enable for each VPC.



Note We don't currently support China Regions.

Requirements and Prerequisites for AWS

For all capabilities: Create a dedicated user in AWS, or identify an existing AWS user for this connector. The connector configuration wizard generates a CloudFormation Template (CFT) that you can use to assign required privileges to this user. Make sure you have permissions in AWS to upload this CFT.

For granting cross AWS account access to the dedicated user, see [\(Optional\) Configure cross AWS account access in AWS, on page 62](#), including required access privileges.

Each VPC can belong to only one AWS connector. A Secure Workload cluster can have multiple AWS connectors. Gather the information described in the tables in [Configure an AWS Connector](#).

This connector doesn't require a virtual appliance.

For gathering labels and inventory: No additional prerequisites are required.

For ingesting flow logs: VPC level flow log definitions are required in order to trigger the collection of flow logs.

Only VPC-level flow logs can be ingested.

Flow logs must be published to Amazon Simple Storage Service (S3); Secure Workload cannot collect flow data from Amazon CloudWatch logs.

Secure Workload can ingest flow logs from an S3 bucket associated with any account, if the AWS user account credentials provided during connector creation have access to both the VPC flow logs and the S3 bucket.

The following flow log attributes (in any order) are required in the flow log: Source Address, Destination Address, Source Port, Destination Port, Protocol, Packets, Bytes, Start Time, End Time, Action, TCP Flags, Interface-ID, Log status and Flow Direction. Any other attributes are ignored.

Flow logs must capture both Allowed and Denied traffic.



Note The Secure Workload AWS connector does not support VPC flow logs partition on an hourly basis.

For segmentation: Enabling segmentation requires Gather Labels to be enabled.

Back up your existing security groups before enabling segmentation in the connector, as all existing rules are overwritten when you enable segmentation for a VPC.

For more information, see [Best Practices When Enforcing Segmentation Policy for AWS Inventory](#).

For managed Kubernetes services (EKS): If you enable the Kubernetes option, see [Requirements and Prerequisites for EKS](#) in the Managed Kubernetes Services Running on AWS (EKS) section, including required access privileges.

(Optional) Configure cross AWS account access in AWS

If the given user credentials has access to VPCs belonging to other AWS accounts, they will be available for processing as part of the AWS connector.

1. The designated Secure Workload user should have the following AWS access permissions:

1. iam:GetPolicyVersion
2. iam:ListPolicyVersions
3. iam:ListAttachedUserPolicies
4. iam:GetUser

Example AWS policy JSON:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "iam:GetPolicyVersion",
        "iam:ListPolicyVersions",
        "iam:ListAttachedUserPolicies",
        "iam:GetUser"
      ],
      "Resource": "*"
    }
  ]
}
```

2. Create an AWS IAM role in the desired AWS account of which the designated Secure Workload user is NOT part of.
3. Allow the AWS IAM role to be assumed by the Secure Workload user. This can be done by adding the Secure Workload user ARN to the AWS IAM role trust policy.

Example AWS IAM role trust policy JSON:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "<Secure Workload_user_arn>"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

4. Perform the steps 2 and 3 for all the desired AWS accounts which the Secure Workload user does not belong to.
5. Create a customer managed policy (NOT Inline policy) with permission to assume all the created AWS roles from different accounts.



Note In AWS connector, Customer Inline Policy is not supported.

Example Managed policy JSON:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
```

```

    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": [<AWS_role_cross_account_1_arn>, <AWS_role_cross_account_2_arn>...]
  }
]
}

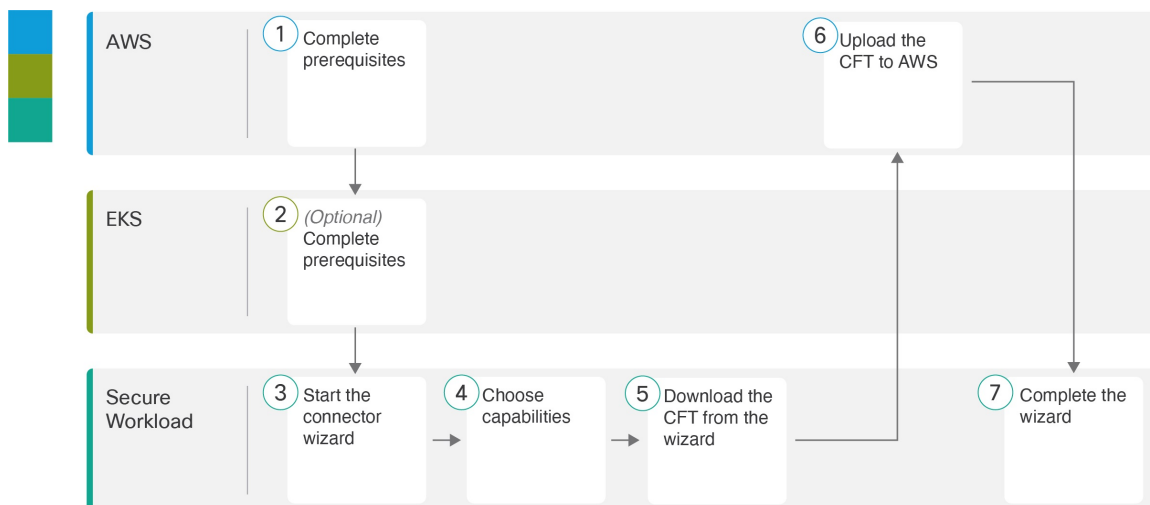
```

6. [Attach](#) the created customer managed policy to the Secure Workload user.
7. The connector configuration wizard will provide a CloudFormation Template. After uploading the CFT as-is to the designated Secure Workload user, you will edit the template and upload the edited template to the CloudFormation portal to grant the required permissions to the AWS IAM roles. For details, see [Configure an AWS Connector](#).

AWS Connector Configuration Overview

The following graphic gives a high-level overview of the connector configuration process. For essential details, see the next topic ([Configure an AWS Connector](#).)

Figure 41: AWS connector configuration overview



(Note that the numbers in the graphic do not correspond to step numbers in the detailed procedure.)

Configure an AWS Connector

- Step 1** From the navigation bar at the left side of the window, choose **Manage > Connectors**.
- Step 2** Click the AWS connector.
- Step 3** Click **Enable** for the first connector (in a root scope) or **Enable Another** for additional connectors in the same root scope.
- Step 4** Understand and meet [Requirements and Prerequisites for AWS](#), then click **Get Started**.
- Step 5** Name the connector and choose the desired capabilities, then click **Next**.

Selections that you make on this page are used only to determine the privileges included in the CloudFormation Template (CFT) that are in the next step, and to display the settings that you must configure.

To enable segmentation, you must also enable **Gather Labels**.

Enabling Segmentation on this page does not enable policy enforcement or affect existing security groups. Policy enforcement and deletion of existing security groups occurs only if you enable Segmentation for individual VPCs later in the wizard. You can return to this wizard later to enable segmentation policy enforcement for individual VPCs.

Step 6 Download the generated CloudFormation Template (CFT).

This template has the IAM privileges that are required for the capabilities that you selected in the previous step.

If you enabled the Kubernetes option, you must separately configure permissions for EKS. See the Managed Kubernetes Services Running on AWS (EKS) section below.

Step 7 Upload the CFT to the AWS CloudFormation portal to assign privileges to the user for this connector. Ensure that the AWS user has required privileges before continuing to the next step in the wizard.

Note We recommend this task whether you are using AWS cross-account access or not.

You can apply the CFT using either the portal or the CLI. For instructions, see:

- **Portal:** [AWS Management Console](#)
- **CLI:** [Creating a stack](#)

When you upload the CFT, AWS asks for the following:

- a. Name of the policy (Name of the policy, for example, Secure WorkloadConnector).
- b. List of bucket ARNs And Object ARNs (Default: *)
- c. Username: Name of the AWS user to which you are applying the CFT
- d. List of VPC ARNs (Default: *)

Step 8 If you are using AWS cross-account access, follow these additional steps:

- a. Edit the CloudFormation Template that you downloaded from the wizard:

The relevant part of the template before the edit:

```

    },
    "Users": [
      {
        "Ref": "Username"
      }
    ]
  }
},
"Parameters": {
  "PolicyName": {
    "Type": "String",
    "Default": "",
    "Description": "Name of the policy. Example: CiscoSecureWorkloadPolicy"
  },
  "Username": {
    "Type": "String",
    "Default": "",
    "Description": "User name. Example: \"SecureWorkloadUser\""
  }
}
}

```

The same part of the template after the edit:

- b. Upload the edited CFT to the AWS CloudFormation portal of each AWS account where the desired IAM role exists.

You can apply the CFT using either the portal or the CLI, as described in the previous step.

When you upload the CFT, AWS will ask for the following:

1. Name of the policy (This can be anything. For example, Secure WorkloadConnector)
2. List of bucket ARNs And Object ARNs (Default: *)
3. Rolename: Name of the AWS IAM role to which you are applying the CFT
4. List of VPC ARNs (Default: *)

Step 9 Configure settings:

Attribute	Description
Access Key	ACCESS KEY ID associated with the AWS user that has the privileges described in the CFT above.
Secret Key	SECRET KEY associated with the ACCESS KEY ID above.
HTTP Proxy	Proxy required for Secure Workload to reach AWS. Supported proxy ports: 80,8080, 443, and 3128.
Full Scan Interval	Frequency with which Secure Workload refreshes complete inventory data from AWS. Default and minimum is 3600 seconds.
Delta Scan Interval	Frequency with which Secure Workload fetches incremental changes in inventory data from AWS. Default and minimum is 600 seconds.

Step 10 Click Next. It may take a few minutes for the system to obtain the list of VPCs and EKS clusters from AWS.

Step 11 From the list of VPCs (Virtual Networks) and EKS clusters for each VPC, choose the VPCs and EKS clusters for which you want to enable your selected capabilities.

Generally, you should enable flow ingestion as soon as possible, so that Secure Workload can begin to collect enough data required to suggest accurate policies.

Note that since EKS only supports Gather Labels capability, no explicit capability selection has been provided. Selecting an EKS cluster will implicitly enable the supported capability. For each cluster for which you enable this capability, enter the **Assume Role ARN** (The Amazon resource number of the role to assume while connecting to Secure Workload.)

Generally, you should not choose **Enable Segmentation** during initial configuration. Later, when you are ready to enforce segmentation policy for specific VPCs, you can edit the connector and enable segmentation for those VPCs. See the Best Practices When Enforcing Segmentation Policy for AWS Inventory.

Step 12 Once your selections are complete, click **Create** and wait a few minutes for the validation check to complete.

The View Groups page shows all VPCs that you enabled for any functionality on the previous page, grouped by region. Each region, and each VPC in each region, is a new scope.

- Step 13** Choose the parent scope under which to add the new set of scopes. If you have not yet defined any scopes, your only option is the default scope.
- Step 14** To accept all settings configured in the wizard *including* the hierarchical scope tree, click **Save**.
To accept all settings *except* the hierarchical scope tree, click **Skip** this step.
You can manually create or edit the scope tree later, under **Organize > Scopes and Inventory**.
-

What to do next

If you have enabled gathering labels, ingesting flow data, and/or segmentation:

- If you enabled flow ingestion, it may take up to 25 minutes for flows to begin appearing on the **Investigate > Traffic** page.
- (Optional) For richer flow data and other benefits including visibility into host vulnerabilities (CVEs), install the appropriate agent for your operating system on your VPC-based workloads. For requirements and details, see the agent installation chapter.
- After you have successfully configured the AWS connector to gather labels and ingest flows, follow the standard process for building segmentation policies. For example: Allow Secure Workload to gather sufficient flow data to generate reliable policies; define or modify scopes (typically one for each VPC); create a workspace for each scope; automatically discover policies based on your flow data, and/or manually create policies; analyze and refine your policies; ensure that your policies meet the guidelines and best practices below; and then, when you are ready, approve and enforce those policies in the workspace. When you are ready to enforce segmentation policy for a particular VPC, return to the connector configuration to enable segmentation for the VPC. For details, see [Best Practices When Enforcing Segmentation Policy for AWS Inventory, on page 68](#).

If you have enabled the Kubernetes managed services (EKS) option:

- Install Kubernetes agents on your container-based workloads. For details, see the *Kubernetes/Openshift Agents - Deep Visibility and Enforcement* section in the agent deployment chapter.

Edit an AWS Connector

You can edit an AWS connector, for example to enable segmentation enforcement for specific VPCs or to make other changes.

Changes are not saved until you finish the wizard.

- Step 1** From the navigation bar at the left side of the window, choose **Manage > Workloads > Connectors**.
- Step 2** Click **AWS**.
- Step 3** If you have more than one AWS connector, choose the connector to edit from the top of the window.
- Step 4** Click **Edit Connector**.
- Step 5** Click through the wizard again and make changes. For detailed descriptions of the settings, see [Configure an AWS Connector](#).
- Step 6** If you enable different capabilities (gathering labels, ingesting flows, enforcing segmentation, or gathering EKS data), you must download the revised CloudFormation Template (CFT) and upload it to AWS before continuing the wizard.

- Step 7** To enable enforcement of segmentation policy, first make sure you have completed recommended prerequisites described in [Best Practices When Enforcing Segmentation Policy for AWS Inventory](#). On the page that lists the VPCs, choose **Enable Segmentation** for the VPCs on which you want to enable enforcement.
- Step 8** If you have already created scopes for any of the selected VPCs, either using the wizard or manually, click **Skip this step** to complete the wizard.
- You can edit the scope tree manually using the **Organize > Scopes and Inventory** page.
- Step 9** If you have not already created any scopes for the selected VPCs and you want to keep the proposed hierarchy, choose the parent scope from above the scope tree, then click **Save**.

Deleting Connectors and Data

If you delete a connector, data already ingested by that connector is not deleted.

Labels and inventory are automatically deleted from active inventory after 24 hours.

Best Practices When Enforcing Segmentation Policy for AWS Inventory



Warning Before you enable segmentation enforcement on any VPC, create a backup of the security groups on that VPC. Enabling segmentation for a VPC removes existing Security Groups from that VPC. Disabling segmentation does not restore the old security groups.

When creating policies:

- As with all discovered policies, ensure that you have enough flow data to produce accurate policies.
- Because AWS allows only ALLOW rules in security groups, your segmentation policies should include only Allow policies, except the Catch-All policy, which should have the Deny action.

We recommend that you enable enforcement in the workspace before you enable segmentation for the associated VPC. If you enable segmentation for a VPC that is not included in a workspace that has enforcement enabled, all traffic will be allowed on that VPC.

When you are ready to enforce policy for a VPC, edit the AWS connector (see [Edit an AWS Connector](#)) and enable segmentation for that VPC.

View AWS Inventory Labels, Details, and Enforcement Status

To view summary information for an AWS connector, navigate to the connector page (Manage > Connectors), then choose the connector from the top of the page. For more details, click a VPC row.

To view information about AWS VPC inventory, click an IP address on the AWS Connectors page to see the Inventory Profile page for that workload. For more information about inventory profiles, see [Inventory Profile](#).

For information about labels, see:

- [Labels Generated by Cloud Connector](#)
- [Labels Related to Kubernetes Clusters](#)

Concrete policies for VPC inventory are generated based on their `orchestrator_system/interface_id` label value. You can see this on the Inventory Profile page.

To view enforcement status, choose **Defend > Enforcement Status** from the navigation bar on the left side of the Secure Workload window. For more information, see [Enforcement Status for Cloud Connectors](#).

Troubleshoot AWS Connector Issues

Problem: The Enforcement Status page shows that a Concrete Policy was SKIPPED.

Solution: This occurs when the number of security groups exceeds the AWS limits, as configured in the AWS connector.

When a concrete policy shows as SKIPPED, the new security groups are not implemented and the previously existing security groups on AWS remain in effect.

To resolve this issue, see if you can consolidate policies, for example by using a larger subnet in one policy rather than multiple policies with smaller subnets.

If you choose to increase limits on the number of rules, you must contact Amazon before changing the limits in the AWS connector configuration.

Background:

Concrete policies are generated for each VPC when segmentation is enabled. These concrete policies are used to create security groups in AWS. However, AWS and Secure Workload count policies differently. When converting Secure Workload policies to AWS security groups, AWS counts each unique subnet as one rule.

Accounting example:

Consider the following example Secure Workload policy:

OUTBOUND: Consumer Address Set -> Provider Address Set Allow TCP port 80, 8080

AWS counts this policy as (the number of unique subnets in the Provider Address set) multiplied by (the number of unique ports).

So, if the provider address set consists of 20 Unique subnets, then this single Secure Workload policy counts in AWS as $20(\text{unique subnets}) * 2(\text{Unique ports}) = 40$ rules in security groups.

Keep in mind that because the VPCs are dynamic, the rule count is also dynamic, so the counts are approximate.

Problem: AWS unexpectedly allows all traffic

Solution: Make sure your Catch-All policy in Secure Workload is set to Deny.

Managed Kubernetes Services Running on AWS (EKS)

If you have deployed Amazon Elastic Kubernetes Service (EKS) on your AWS cloud, then you can use an AWS connector to pull in inventory and labels (EKS tags) from your Kubernetes cluster.

When an AWS connector is configured to pull metadata from managed Kubernetes services, Secure Workload connects to the cluster's API server and tracks the status of nodes, pods and services in that cluster. For the Kubernetes labels gathered and generated using this connector, see [Labels Related to Kubernetes Clusters](#).

Requirements and Prerequisites for EKS

- Verify that your Kubernetes version is supported. See <https://www.cisco.com/go/secure-workload/requirements/integrations>.
- Configure the required access in EKS, as described below.

EKS Roles and Access Privileges

User credentials and AssumeRole (if applicable) must be configured with a minimum set of privileges. The user/role must be specified in the aws-auth.yaml config map. The aws-auth.yaml config map can be edited using the following command.

```
$ kubectl edit configmap -n kube-system aws-auth
```

If AssumeRole is not used, the user must be added to the “mapUsers” section of the aws-auth.yaml config map with appropriate group. If AssumeRole ARN is specified, the role must be added to the “mapRoles” section of the aws-auth.yaml config map. A sample aws-auth.yaml config map with AssumeRole is provided below.

```
apiVersion: v1
data:
  mapAccounts: |
    []
  mapRoles: |
    - "groups":
      - "system:bootstrappers"
      - "system:nodes"
      "rolearn": "arn:aws:iam::938996165657:role/eks-cluster-202101141814452347000000a"

      "username": "system:node:{EC2PrivateDNSName}"
    - "rolearn": arn:aws:iam::938996165657:role/BasicPrivilegesRole
      "username": secure.workload.read.only-user
      "groups":
        - secure.workload.read.only

  mapUsers: |
    []
kind: ConfigMap
metadata:
  creationTimestamp: "2021-01-14T18:14:47Z"
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:data:
        .: {}
        f:mapAccounts: {}
        f:mapRoles: {}
        f:mapUsers: {}
    manager: HashiCorp
    operation: Update
    time: "2021-01-14T18:14:47Z"
  name: aws-auth
  namespace: kube-system
  resourceVersion: "829"
  selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth
  uid: 6c5a3ac7-58c7-4c57-a9c9-cad701110569
```

EKS specific RBAC considerations

Create a cluster role binding of the cluster role and the user/service account.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: csw-clusterrolebinding
subjects:
- kind: User
  name: csw.read.only
```

```

apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: csw.read.only
apiGroup: rbac.authorization.k8s.io
kubectl create -f clusterrolebinding.yaml
clusterrolebinding.rbac.authorization.k8s.io/csw-clusterrolebinding created

```

For information on EKS roles and access, see the [EKS Roles and Access Privileges](#) section.

Configure EKS Settings in the AWS Connector Wizard

You enable the Managed Kubernetes Services capability when you configure the AWS connector. See [Configure an AWS Connector](#).

You will need the Assume Role ARN for each EKS cluster. For more information, see: https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html

If you are using the AWS user to access the EKS cluster, allow the user to access the Assume Role.

If you are using a cross-account IAM role, allow the IAM role to access the Assume Role.

Azure Connector

The Azure connector connects with your Microsoft Azure account to perform the following high-level functions:

- **Automated ingestion of inventory (and its tags) live from your Azure virtual networks (VNETs)**
Azure allows you to assign metadata to your resources in the form of tags. Secure Workload can ingest the tags associated with virtual machines and network interfaces, which can then be used as labels in Secure Workload for inventory and traffic flow data visualization and policy definitions. This metadata is synchronized constantly.

The tags from workloads and network interfaces of the subscription associated with the connector are ingested. If both workloads and network interfaces are configured then the tags are merged and displayed in Secure Workload. For more information, see [Labels Generated by Cloud Connectors](#).
- **Ingestion of flow logs** The connector can ingest flow logs that you set up in Azure for your Network Security Groups (NSGs). You can then use this telemetry data in Secure Workload for visualization and segmentation policy generation.
- **Segmentation** When enforcement of segmentation policy is enabled for a virtual network, Secure Workload policies will be enforced using Azure's native Network Security Groups.
- **Automated ingestion of metadata from AKS clusters** When Azure Kubernetes Services (AKS) are running on Azure, you can choose to gather all node, service, and pod metadata related to all selected Kubernetes clusters.

You can choose which of the above capabilities to enable for each VNet.

Azure connector supports multiple subscriptions.



Note China Regions are currently not supported.

Requirements and Prerequisites for Azure

For all capabilities: A single connector can handle multiple subscriptions. You will need a subscription ID to configure the connector. This subscription ID can be one of the many subscription IDs that are being onboarded to a connector.

In Azure, create/register an application using Azure Active Directory (AD). You will need the following information from this application:

- Application (client) ID
- Directory (tenant) ID
- Client credentials (you can use either a certificate or a client secret)
- Subscription ID

The connector configuration wizard will generate an Azure Resource Manager (ARM) template that you can use to create a custom role with the permissions needed for the connector capabilities you choose to enable. These permissions will apply to all resources in the subscription you specify for the connector. Make sure you have permissions in Azure to upload this template.

If required for connectivity, ensure that you have an HTTP proxy available for this integration.

Each virtual network (VNet) can belong to only one Azure connector. An Azure account can have multiple Azure connectors.

This connector does not require a virtual appliance.

For gathering labels and inventory: No additional prerequisites are required.

For ingesting flow logs: Each virtual network (VNet) must have at least one subnet configured.

Every subnet under each VNet must have a Network Security Group (NSG) associated with it. You can associate a single NSG with multiple subnets. You can specify any Resource Group when configuring the NSG.

Only traffic that hits an NSG rule will be included in flow logs. Therefore, every NSG should have at least one rule each for inbound traffic and for outbound traffic that applies to any source, any destination – the equivalent of a catch-all rule in Secure Workload. (By default, NSGs include these rules.)

Each NSG must have flow logs enabled.

- A storage account in Azure is required. Access permissions must be included for the subscription you are using for this connector.
- The flow logs must use Version 2.
- Retention time can be 2 days (the connector pulls new flow data every minute, and two days should allow enough time for any connection failures to be remedied.)

For segmentation: Enabling segmentation requires Gather Labels to be enabled.

When you enable segmentation for a virtual network (VNet), all existing rules are removed from the NSGs associated with subnets and the network interfaces that are part of those subnets. Back up your existing NSG rules on subnet and network interface before you enable segmentation in the connector.

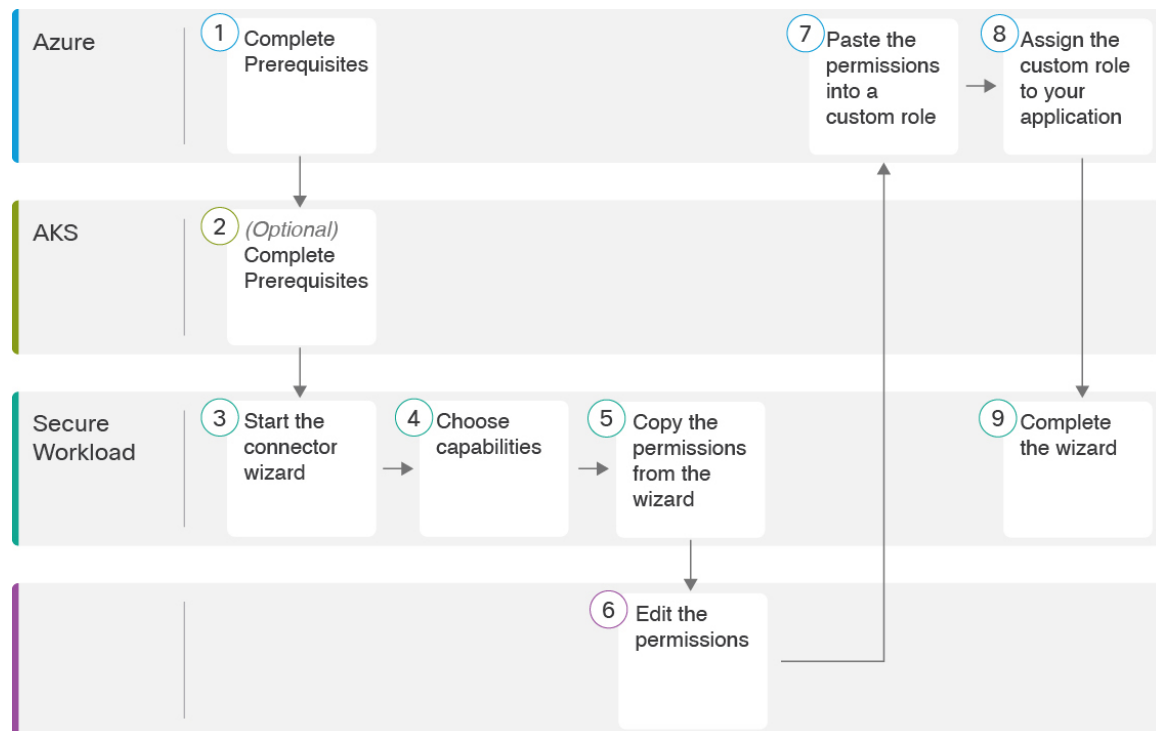
See also [Best Practices When Enforcing Segmentation Policy for Azure Inventory, on page 76](#), below.

For managed Kubernetes services (AKS): If you will enable the Kubernetes AKS option, see requirements and prerequisites in the Managed Kubernetes Services running on Azure (AKS) section below, .

Azure Connector Configuration Overview

The following graphic gives a high-level overview of the connector configuration process. For essential details, see the next topic ([Configure an Azure Connector](#).)

Figure 42: Azure connector configuration overview



(Note that the numbers in the graphic do not correspond to step numbers in the detailed procedure.)

Configure an Azure Connector

- Step 1** From the navigation bar at the left side of the window, choose **Manage > Connectors**.
- Step 2** Click the Azure connector.
- Step 3** Click **Enable** for the first connector (in a root scope) or **Enable Another** for additional connectors in the same root scope.
- Step 4** Understand and meet requirements and prerequisites in [Requirements and Prerequisites for Azure](#), then click Get Started.
- Step 5** Name the connector and choose desired capabilities:
- Selections you make on this page are used only to determine the privileges included in the Azure Resource Manager (ARM) template that will be generated in the next step, and to display the settings that you will need to configure.
- In order to enable segmentation, you must also enable **Gather Labels**.
- Enabling Segmentation on this page does not in itself enable policy enforcement or affect existing network security groups. Policy enforcement and deletion of existing security groups occurs only if you enable Segmentation for individual VNets later in the wizard. You can return to this wizard later to enable segmentation policy enforcement for individual VNets.
- Step 6** Click **Next** and read the information on the configuration page.

Step 7 Your subscription must have the required privileges before you can continue to the next page in the wizard.

To use the provided Azure Resource Manager (ARM) template to assign required permissions for the connector:

- a. Download the ARM template from the wizard.
- b. Edit the template text to replace `<subscription_ID>` with your subscription ID.

Note For a connector, you can create multiple subscription IDs in the Azure account.
You can enter multiple subscription IDs where the credentials belong to the same subscription ID.
- c. In Azure, create a custom role in the applicable subscription.
- d. In the custom role form, for the Baseline permissions, choose **Start from scratch**.
- e. In the JSON tab of the custom role creation form, paste the text from the edited file you downloaded from the connector wizard.
- f. Save the custom role.
- g. Attach the custom role to the application you configured in the prerequisites for this procedure.

This template has the IAM permissions required for the capabilities that you selected in the previous step.

If you enabled the Kubernetes managed services option, you must separately configure permissions for AKS. See [Managed Kubernetes Services Running on Azure \(AKS\), on page 77](#).

Step 8 Configure settings:

Attribute	Description
SubscriptionID	The ID of the Azure subscription that you are associating with this connector.
ClientID	The Application (client) ID from the application that you created in Azure for this connector.
TenantID	The Directory (tenant) ID from the application that you created in Azure for this connector.
Client Secret or Client Certificate	For authentication, you can use either a client secret or a client certificate and key. Obtain either from the Client credentials link in the application that you created in Azure for this connector. If you use a certificate: The certificate should be unencrypted. Only RSA certificates are supported. Private keys can be either PKCS1 or PKCS8.
HTTP Proxy	Proxy required for Secure Workload to reach Azure. Supported proxy ports: 80, 8080, 443, and 3128.
Full Scan Interval	Frequency with which Secure Workload refreshes complete inventory data from Azure. Default and minimum is 3600 seconds.

Attribute	Description
Delta Scan Interval	Frequency with which Secure Workload fetches incremental changes in inventory data from Azure. Default and minimum is 600 seconds.

- Step 9** Click **Next**. It may take a few minutes for the system to obtain the list of VNets and AKS clusters from Azure.
- Step 10** From the list of VNets and AKS clusters for each VNet, choose the VNets and AKS clusters for which you want to enable your selected capabilities.
- Generally, you should enable flow ingestion as soon as possible, so that Secure Workload can begin to collect enough data to suggest accurate policies.
- Note that since AKS only supports Gather Labels capability, no explicit capability selection has been provided. Selecting an AKS cluster will implicitly enable the supported capability. Upload the client certificate and key for each cluster for which you enable this functionality.
- Generally, you should not choose **Enable Segmentation** during initial configuration. Later, when you are ready to enforce segmentation policy for specific VNets, you can edit the connector and enable segmentation for those VNets. See [Best Practices When Enforcing Segmentation Policy for Azure Inventory, on page 76](#).
- Step 11** Once your selections are complete, click **Create** and wait a few minutes for the validation check to complete.
- The View Groups page shows all VNets that you enabled for any functionality on the previous page, grouped by region. Each region, and each VNet in each region, is a new scope.
- Step 12** (Optional) Choose the parent scope under which to add the new set of scopes. If you have not yet defined any scopes, your only option is the default scope.
- Step 13** (Optional) To accept all settings configured in the wizard *including* the hierarchical scope tree, click **Save**.
- To accept all settings *except* the hierarchical scope tree, click **Skip** this step.
- You can manually create or edit the scope tree later, under **Organize > Scopes and Inventory**.

What to do next

If you have enabled gathering labels, ingesting flows data, and/or segmentation:

- If you enabled flow ingestion, it may take up to 25 minutes for flows to begin appearing on the **Investigate > Traffic** page.
- (Optional) For richer flow data and other benefits including visibility into host vulnerabilities (CVEs), install the appropriate agent for your operating system on your VNet-based workloads. For requirements and details, see the agent installation chapter.
- After you have successfully configured the Azure connector to gather labels and ingest flows, follow the standard process for building segmentation policies. For example: Allow Secure Workload to gather sufficient flow data to generate reliable policies; define or modify scopes (typically one for each VNet); create a workspace for each scope; automatically discover policies based on your flow data, and/or manually create policies; analyze and refine your policies; ensure that your policies meet the guidelines and best practices below; and then, when you are ready, approve and enforce those policies in the workspace. When you are ready to enforce segmentation policy for a particular VNet, return to the

connector configuration to enable segmentation for the VNet. For details, see [Best Practices When Enforcing Segmentation Policy for Azure Inventory, on page 76](#).

If you have enabled the Kubernetes managed services (AKS) option:

- Install Kubernetes agents on your container-based workloads. For details, see [Installing Kubernetes or OpenShift Agents for Deep Visibility and Enforcement](#) in the agent deployment chapter.

Edit an Azure Connector

You can edit an Azure connector, for example to enable segmentation enforcement for specific VNets or to make other changes.

Changes are not saved until you finish the wizard.

-
- Step 1** From the navigation bar at the left side of the window, choose **Manage > Connectors**.
- Step 2** Click **Azure**.
- Step 3** If you have more than one Azure connector, choose the connector to edit from the top of the window.
- Step 4** Click **Edit Connector**.
- Step 5** Click through the wizard again and make changes. For detailed descriptions of the settings, see [Configure an Azure Connector, on page 73](#).
- Step 6** If you enable different capabilities (gathering labels, ingesting flows, enforcing segmentation, or gathering AKS data), you must download the revised ARM template, edit the new template text to specify the subscription ID, and upload the new template to the custom role you created in Azure before continuing the wizard.
- Step 7** To enable enforcement of segmentation policy, first make sure you have completed recommended prerequisites described in [Best Practices When Enforcing Segmentation Policy for Azure Inventory, on page 76](#). Then, on the wizard page that lists the VNets, choose **Enable Segmentation** for the VNets on which you want to enable enforcement.
- Step 8** If you have already created scopes for any of the selected VNets, either using the wizard or manually, click **Skip this step** to complete the wizard.
- You can edit the scope tree manually using the **Organize > Scopes and Inventory** page.
- Step 9** If you have not already created any scopes for the selected VNets and you want to keep the proposed hierarchy, choose the parent scope from above the scope tree, then click **Save**.
-

Deleting Connectors and Data

If you delete a connector, data already ingested by that connector is not deleted.

Labels and inventory are automatically deleted from active inventory after 24 hours.

Best Practices When Enforcing Segmentation Policy for Azure Inventory



Warning

Before you enable segmentation enforcement on any VNet, create a backup of the network security groups on that VNet. Enabling segmentation for a VNet removes existing rules from the network security group associated with that virtual network. Disabling segmentation does not restore the old network security groups.

When creating policies: As with all discovered policies, ensure that you have enough flow data to produce accurate policies.

We recommend that you enable enforcement in the workspace before you enable segmentation for the associated VNet. If you enable segmentation for a VNet that is not included in a workspace that has enforcement enabled, all traffic will be allowed on that VNet.

When you are ready to enforce policy for a VNet, edit the Azure connector (see [Edit an Azure Connector, on page 76](#)) and enable segmentation for that VNet.

Note that if a subnet does not have a Network Security Group associated with it, Secure Workload does not enforce segmentation policy on that subnet. When you enforce segmentation policy on a VNet, the NSG at the subnet level is changed to allow all traffic, and Secure Workload policies overwrite the interface-level NSGs. An NSG for the interface is automatically created if not already present.

View Azure Inventory Labels, Details, and Enforcement Status

To view summary information for an Azure connector, navigate to the connector page (Manage > Connectors), then choose the connector from the top of the page. For more details, click a VNet row.

To view information about Azure VNet inventory, click an IP address on the Azure Connectors page to view the Inventory Profile page for that workload. For more information about inventory profiles, see [Inventory Profile](#).

For information about labels, see:

- [Labels Generated by Cloud Connectors](#)
- [Labels Related to Kubernetes Clusters](#)

Concrete policies for VNet inventory are generated based on their `orchestrator_system/interface_id` label value. You can see this on the Inventory Profile page.

To view enforcement status, choose **Defend > Enforcement Status** from the navigation bar on the left side of the Secure Workload window. For more information, see [Enforcement Status for Cloud Connectors](#).

Troubleshoot Azure Connector Issues

Problem: Azure unexpectedly allows all traffic

Solution: Make sure your Catch-All policy in Secure Workload is set to Deny.

Managed Kubernetes Services Running on Azure (AKS)

If you have deployed Azure Kubernetes Services (AKS) on your Azure cloud, then you can use an Azure connector to dynamically pull in inventory and labels (AKS tags) from your Kubernetes cluster.

When an Azure connector is configured to pull metadata from managed Kubernetes services, Secure Workload tracks the status of nodes, pods and services in that cluster.

For the Kubernetes labels gathered and generated using this connector, see [Labels Related to Kubernetes Clusters](#).

Requirements and Prerequisites for AKS

- Verify that your Kubernetes version is supported. See the [Compatibility Matrix](#) for the operating systems, external systems, and connectors for Secure Workload agents.

- Enable and configure the Managed Kubernetes Services (AKS) capability when you configure the Azure connector. See [Configure an Azure Connector](#) for details.

GCP Connector

The Google Cloud Platform connector connects with GCP to perform the following high-level functions:

- **Automated ingestion of inventory (and its tags) live from GCP Virtual Private Cloud (VPC)**

GCP allows you to assign metadata to your resources in the form of tags. Secure Workload will query the tags for these resources which can then be used for inventory and traffic flow data visualization, and policy definition. This capability keeps the resource tag mapping updated by constantly synchronizing this data.

The tags from workloads and network interfaces of a GCP VPC are ingested. If both workloads and network interfaces are configured then the tags are merged and displayed in Secure Workload. For more information, see [Labels Generated by Cloud Connectors](#).

- **Ingestion of flow logs from VPC** If you have set up VPC flow logs in GCP for monitoring purposes, Secure Workload can ingest flow log information by reading the corresponding Google Storage bucket. This telemetry can be used for visualization and segmentation policy generation.
- **Segmentation** Enabling this option will allow Secure Workload to program security policies using GCP native VPC firewall. When enforcement is enabled for a VPC, relevant policies will be automatically programmed to the VPC firewall.
- **Automated ingestion of metadata from GKE clusters** (K8s capabilities) when Google Kubernetes Engine (GKE) is running on GCP, you can choose to gather all node, service, and pod metadata related to all selected Kubernetes clusters.

You can choose which of the above capabilities to enable for each VPC.

Requirements and Prerequisites for GCP Connector

For all capabilities: Create a dedicated service account in GCP, or identify an existing GCP service account for this connector. The connector configuration wizard generates a IAM policy list that you can use to assign required privileges to this service account. Make sure you have permissions in GCP to upload this IAM policy list.



Note The recommended method for applying the permission in the IAM policy list to the service account is through the CLI.

Each VPC can belong to only one GCP connector. An Secure Workload cluster can have multiple GCP connectors. Gather the information described in the tables in [Configure a GCP Connector, on page 81](#), below.

This connector does not require a virtual appliance.

- **For gathering labels and inventory:** No additional prerequisites are required.
- **For ingesting flow logs:** VPC level flow log definitions are required in order to trigger the collection of flow logs.

To use the flow log ingestion, user is required to enable flow logs on the desired VPCs and setup a log router sink.

Inclusion filter for the log router sink:

1. `resource.type="gce-subnetwork"`
2. `log_name="projects/<project_id>/logs/compute.googleapis.com%2Fvpc_flows"`

Choose the sink destination as a cloud storage bucket and then choose the desired storage bucket.

While configuring the GCP connector with ingress flow logs, it is mandatory to enter the storage bucket name.

Only flow logs from VPC can be ingested.

Flow logs must be published to Google storage bucket; Secure Workload cannot collect flow data from Google Cloud Operations Suite.

Secure Workload can ingest flow logs from an Google Storage bucket associated with any account, if the GCP user account provided during connector creation have access to both the VPC flow logs and the Google storage bucket.

The following flow log attributes (in any order) are required in the flow log: Source Address, Destination Address, Source Port, Destination Port, Protocol, Packets, Bytes, Start Time, End Time, Action, TCP Flags, Interface-ID, Log status and Flow Direction. Any other attributes are ignored.

Flow logs must capture both Allowed and Denied traffic.

- **For segmentation:** Enabling segmentation requires Gather Labels to be enabled.
Back up your existing security groups before enabling segmentation in the connector, as all existing rules will be overwritten when you enable segmentation policy enforcement for a VPC.
See also [Best Practices When Enforcing Segmentation Policy for GCP Inventory](#), on page 84, below.
- **For managed Kubernetes services (GKE):** If you enable the Kubernetes option, see requirements and prerequisites in the [Managed Kubernetes Services Running on GCP \(GKE\)](#), on page 85 section below, including required access privileges.

Configure Multiple Projects Access in GCP

To configure cross multiple projects access in GCP, you can follow these steps:

-
- Step 1** Sign in to your [GCP](#) console.
- Step 2** Click on the project drop-down menu in the top navigation bar and select **New Project** or you can either create a new Project or use an existing project with service Account.
- Step 3** Enter a name for your new project. Choose the organization that own the new project or select **No organization** if you do not have one.
- Step 4** Click on the **Create** button to create the new project.
- Note** You can repeat the step 2 to 4 to create as many projects as you need.
- Step 5** To link multiple projects in a single service account, navigate to **IAM & Admin** page and choose **Service Account**.
- Step 6** Click on the **Create Service Account** button. Follow the prompts to create the service account and grant it the necessary permissions.
- Note** You can either use an existing service account or create a new service account.

Step 7 From the **Keys** tab, click **Add Key** to generate a private key in JSON file.

Step 8 Go to the **IAM & Admin** page in the GCP console and select **IAM**.

Note You have to first change the project before you click on IAM & Admin and then try to grant privilege.

Step 9 Click on the **Grant access** button to add a new project.

Step 10 In the **New principals** field, enter the email address of the service account you want to link to the project.

Step 11 Click on the **Save** button to associate the service account to your project.

Note Repeat these steps for each project that you want to link to your original project.

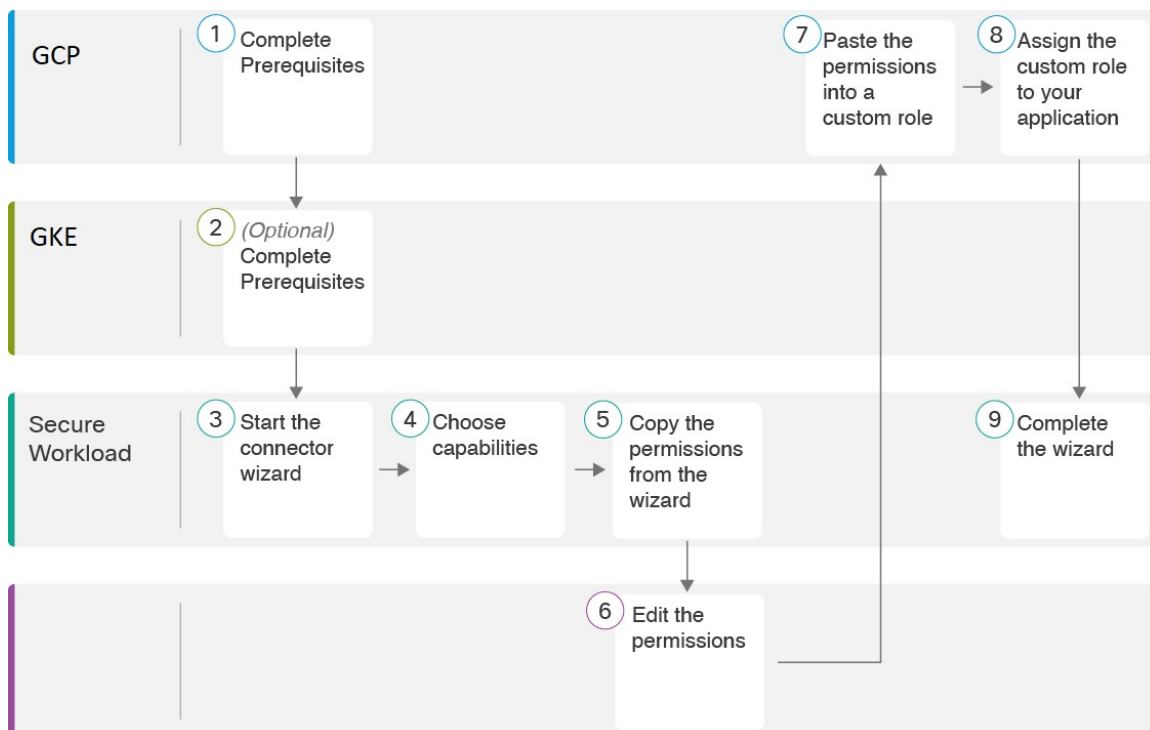
You can manage the service account permissions by going to the **IAM & Admin** page in the GCP console and selecting **IAM** for each project.

Step 12 Make sure that the Service Account has permissions to least common ancestor (common ancestor to all the projects selected) resource level, such as a folder or organization.

GCP Connector Configuration Overview

The following graphic gives a high-level overview of the connector configuration process. For essential details, see the next topic ([Configure a GCP Connector, on page 81.](#))

Figure 43: GCP connector configuration overview



(Note that the numbers in the graphic do not correspond to step numbers in the detailed procedure.)

Configure a GCP Connector

Step 1 From the navigation bar at the left side of the window, choose **Manage > Connectors**.

Step 2 Click the **GCP connector**.

Step 3 Click **Enable** for the first connector (in a root scope) or **Enable Another** for additional connectors in the same root scope.

Step 4 Understand and meet requirements and prerequisites in [Requirements and Prerequisites for GCP Connector, on page 78](#) and [Managed Kubernetes Services Running on GCP \(GKE\), on page 85](#), then click **Get Started**.

Step 5 Enter a name for the connector and choose desired capabilities, then click **Next**.

Selections you make on this page are used only to determine the privileges included in the IAM policy list that will be generated in the next step, and to display the settings that you will need to configure.

In order to enable **segmentation**, you must enable **Gather Labels**.

Step 6 Create **service account** in the [Google Cloud console](#).

Step 7 Download the generated IAM custom role policy list from the Secure Workload GCP Connector configuration windows. This IAM custom role policy list has the IAM privileges required for the capabilities that you selected in the previous step.

If you enabled the Kubernetes option, you must separately configure permissions for GKE. See the [Managed Kubernetes Services Running on GCP \(GKE\), on page 85](#).

Step 8 (Optional) To generate IAM custom role using [Google Cloud CLI](#).

Sample command for custom role creation using Google Cloud CLI is as shown below.

```
gcloud iam roles create <Role Name> --project=<Project id> --file=<File path>
```

Step 9 Attach the created role to the [service account](#) in GCP console.

Step 10 Upload service account json file with required capabilities you have prepared in the prerequisites above.

Note In GCP, the single connector supports multiple projects and ensure that the service account is directly linked to all projects.

Step 11 Enter the **Flow Log Storage Bucket Name** if the the Ingress Flow logs capability is checked.

Step 12 Enter the **Root Resource Id**, which is the GCP folder ID or organization ID.

Note To obtain the Root Resource ID, you can view it directly in the Cloud Console by navigating to the [IAM & Admin, Settings](#) section. Alternatively, you can also utilize the [Cloud SDK command](#) to retrieve the Root Resource ID

Step 13 Configure the following settings:

Attribute	Description
HTTP Proxy	Proxy required for Secure Workload to reach GCP.
Full Scan Interval	Frequency with which Secure Workload refreshes complete inventory data from GCP. Default and minimum is 3600 seconds.
Delta Scan Interval	Frequency with which Secure Workload fetches incremental changes in inventory data from GCP. Default and minimum is 600 seconds.

Step 14 Click **Next**. It may take a few minutes for the system to obtain the list of virtual network and GKE clusters from your GCP project(s).

Step 15 From the list of VPCs (Virtual Networks) and GKE clusters, choose the resources and their respective capabilities.

Generally, you should enable flow ingestion as soon as possible, so that Secure Workload can begin to collect enough data required to suggest accurate policies.

Generally, you should not choose **Enable Segmentation** during initial configuration. Later, when you are ready to enforce segmentation policy for specific VPCs, you can edit the connector and enable segmentation for those VPCs. See the Best Practices When Enforcing Segmentation Policy for GCP Inventory.

Step 16 Click **Create** and wait a few minutes for the validation check to complete.

The View Groups page shows all VPCs that you enabled for any functionality on the previous page, grouped by logical_group_id (CSW), which is also a project_id (GCP). Each logical_group_id, and each VPC in each logical_group_id, is a new scope.

Step 17 Choose the parent scope under which to add the new set of scopes. If you have not yet defined any scopes, your only option is the default scope.

Step 18 To accept all settings configured in the wizard *including* the hierarchical scope tree, click **Save**.

To accept all settings *except* the hierarchical scope tree, click **Skip** this step.

You can manually create or edit the scope tree later, under **Organize > Scopes and Inventory**.

What to do next

If you have enabled gathering labels, ingesting flow data, and/or segmentation:

- If you enabled flow ingestion, it may take up to 25 minutes for flows to begin appearing on the **Investigate > Traffic** page.

- (Optional) For richer flow data and other benefits including visibility into host vulnerabilities (CVEs), install the appropriate agent for your operating system on your VPC-based workloads. For requirements and details, see the agent installation chapter.
- After you have successfully configured the GCP connector to gather labels and ingest flows, follow the standard process for building segmentation policies. For example: Allow Secure Workload to gather sufficient flow data to generate reliable policies; define or modify scopes (typically one for each VPC); create a workspace for each scope; automatically discover policies based on your flow data, and/or manually create policies; analyze and refine your policies; ensure that your policies meet the guidelines and best practices below; and then, when you are ready, approve and enforce those policies in the workspace. When you are ready to enforce segmentation policy for a particular VPC, return to the connector configuration to enable segmentation for the VPC. For details, see [Best Practices When Enforcing Segmentation Policy for GCP Inventory](#), on page 84.

If you have enabled the Kubernetes managed services (GKE) option:

- Install Kubernetes agents on your container-based workloads. For details, see [Kubernetes/Openshift Agents - Deep Visibility and Enforcement](#) in the agent deployment chapter.

Edit a GCP Connector

If you want to enable gathering data from different or additional VPCs or GKE clusters, you may need to upload a service account json file with required capabilities with different permissions before you can select different VPCs or GKEs.

Changes are not saved until you finish the wizard.

-
- Step 1** From the navigation bar at the left side of the window, choose **Manage > Workloads > Connectors**.
 - Step 2** Click **GCP Connector**.
 - Step 3** If you have more than one GCP connector, choose the connector to edit from the top of the window.
 - Step 4** Click **Edit Connector**.
 - Step 5** Click through the wizard again and make changes. For detailed descriptions of the settings, see [Configure a GCP Connector](#), on page 81.
 - Step 6** If you enable different capabilities (gathering labels, ingesting flows, enforcing segmentation, or gathering GKE data), you must download the revised IAM template and upload it to GKE before continuing the wizard.
 - Step 7** To enable enforcement of segmentation policy, first ensure that you have completed recommended prerequisites described in [Best Practices When Enforcing Segmentation Policy for GCP Inventory](#), on page 84. On the page that lists the VPCs, select **Enable Segmentation** for the VPCs on which you want to enable enforcement.
 - Step 8** If you have already created scopes for any of the selected VPCs, either using the wizard or manually, click **Skip this step** to complete the wizard.

You can edit the scope tree manually using the **Organize > Scopes and Inventory** page.
 - Step 9** If you have not already created any scopes for the selected VPCs and you want to keep the proposed hierarchy, choose the parent scope from above the scope tree, then click **Save**.
-

Deleting Connectors and Data GCP

If you delete a connector, data already ingested by that connector is not deleted.

Labels and inventory are automatically deleted from active inventory after 24 hours.

Best Practices When Enforcing Segmentation Policy for GCP Inventory



Warning Before you enable segmentation enforcement on any VPC, create a backup of the security groups on that VPC. Enabling segmentation for a VPC removes existing Security Groups from that VPC. Disabling segmentation does not restore the old security groups.

When creating policies:

- As with all discovered policies, ensure that you have enough flow data to produce accurate policies.
- Because GCP allows both ALLOW/DENY rules in firewall policy. Since GCP has very strict limitation on number of rules. So, it is better to have only ALLOW-list.

We recommend that you enable enforcement in the workspace before you enable segmentation for the associated VPC. If you enable segmentation for a VPC that is not included in a workspace that has enforcement enabled, all traffic will be allowed on that VPC.

When you are ready to enforce policy for a VPC, edit the GCP connector (see [Edit a GCP Connector, on page 83](#)) and enable segmentation for that VPC.

GKE Inventory Labels, Details, and Enforcement Status

To view summary information for a GCP connector, navigate to **Connector** > and choose GCP Connector on the Connectors page.

To view information about inventory, click the IP address of a particular workload from the Scopes and Inventory page. You can also access the Inventory Profile from the interface tab on the VPC Profile. For more information about the Inventory profile, see [Inventory Profile](#).

Similarly, to view all Concrete Policies under the VPC profile, from the Inventory Profile Concrete Policies tab, navigate to the parent VPC Profile to see all the Concrete Policies under the VPC.

The VPC Profile is accessible from the GCP Configuration or Enforcement Status page (either global or within a workspace). You can view the Enforcement Status and Concrete Policies at the VPC level on the VPC Profile. You can also view the combined VPC Firewall Policies of all the interfaces on the VPC Firewall Policies tab.

For more information on labels, see:

- [Labels Generated by Cloud Connectors](#)
- [Labels Related to Kubernetes Clusters](#)

Troubleshoot GCP Connector Issues

Problem: The Enforcement Status page shows that a Concrete Policy was SKIPPED.

Solution: This occurs when the number of rules in firewall policy exceeds the GCP limits, as configured in the GCP connector.

When a concrete policy shows as SKIPPED, the new security groups are not implemented and the previously existing security groups on GCP remain in effect.

To resolve this issue, see if you can consolidate policies, for example by using a larger subnet in one policy rather than multiple policies with smaller subnets.

Background:

Concrete policies are generated for each VPC when segmentation is enabled. These concrete policies are used to create firewall policy in GCP. However, GCP and Secure Workload count policies differently. When converting Secure Workload policies to GCP firewall rules in firewall policy, GCP counting mechanism is complex. For more details, see [GCP](#).

Problem: GCP unexpectedly allows all traffic

Solution: Make sure your Catch-All policy in Secure Workload is set to Deny.

Managed Kubernetes Services Running on GCP (GKE)

You can use a cloud connector to gather metadata from Google Kubernetes Engine (GKE) clusters running on Google Cloud Platform (GCP).

The connector gathers all node, service, and pod metadata related to all selected Kubernetes clusters.

Requirements and Prerequisites

Secure Workload requirements: This connector does not require a virtual appliance.

Platform requirements:

- Make sure you have permissions in GCP to configure the required access for this connector.
- Each GKE cluster can only belong to one GCP connector.
- Gather the information described in the tables in *Configure a GCP connector*, below.

GKE requirements:

- You must configure the required access privileges in GKE.
- To support Managed K8s capabilities, the roles required by the service account are:
 - Compute Network Viewer is an IAM role that gives read-only access to all network resources in GCP. <https://cloud.google.com/compute/docs/access/iam#compute.networkViewer>
 - Kubernetes Engine Viewer is a GKE cluster role that provides read-only access to resources within GKE clusters, such as nodes, pods, and GKE API objects. <https://cloud.google.com/iam/docs/understanding-roles#kubernetes-engine-roles>

Virtual Appliances for Connectors

Most connectors are deployed on Secure Workload virtual appliances. You will deploy required virtual appliances on an ESXi host in VMware vCenter using the OVA templates or on other KVM-based hypervisors using the QCOW2 image. The procedure to deploy virtual appliances is described in [Deploying a Virtual Appliance](#).

Types of Virtual Appliances

Each connector that requires a virtual appliance can be deployed on one of two types of virtual appliances.

Secure Workload Ingest

Secure Workload Ingest appliance is a software appliance that can export flow observations to Secure Workload from various connectors.

Specification

- Number of CPU cores: 8
- Memory: 8 GB
- Storage: 250 GB
- Number of network interfaces: 3
- Number of connectors on one appliance: 3
- Operating System: CentOS 7.9 (Secure Workload 3.8.1.19 and earlier), AlmaLinux 9.2 (Secure Workload 3.8.1.36 and later)

See important limits at [Secure Workload Virtual Appliances for Connectors](#).



Note Each root scope on Secure Workload can have at most 100 Secure Workload Ingest appliances deployed.

Figure 44: Secure Workload Ingest appliance

Tetraton Data Ingest Appliance ACTIVE Decommission

Checked In Sep 4 2020 04:45:59 pm (PDT) **Registered** Aug 25 2020 06:47:59 pm (PDT) **Created** Aug 25 2020 01:55:33 pm (PDT)

Connectors

- AWS
- AnyConnect
- F5

Info | VM | NTP | Log | Alert | Troubleshoot

Tetraton Data Ingest appliance is a software appliance that can export flow data to Tetraton from various connectors. At most 3 connectors may be enabled on an appliance. When Alerts are enabled, the following alerts may be generated:

1. Tetraton Data Ingest appliance is down (due to missing heartbeats).
2. Informational alert on high CPU/Memory/Disk usage.

Tetraton Cluster → **Tetraton Ingest Appliance** → **Campus**

User, Process, Flows, and more

Secure Workload Ingest appliance allows at most 3 connectors to be enabled on an appliance. There can be more than one instance of the same connector enabled on the same appliance. For the ERSPAN Ingest appliance

three ERSPAN connectors are always automatically provisioned. Many of the connectors deployed on Ingest appliance collect telemetry from various points in the network, these connectors need to listen on specific ports on the appliance. Each connector is therefore bound to one of the IP address and the default ports on which the connector should be listening to collect telemetry data. As a result, each IP address is essentially a slot that a connector occupies on the appliance. When a connector is enabled, a slot is taken (thereby, the IP corresponding to the slot). And, when a connector is disabled, the slot occupied by the connector is released (thereby, the IP corresponding to the slot). See the *Secure Workload Ingest appliance slots* for how to ingest appliance maintains the state of the slots.

Figure 45: Secure Workload Ingest appliance slots

```
[root@beretta-ingest-1 tetter]# cat /local/tetration/appliance/appliance.conf
{
  "type": "TETRATION_DATA_INGEST",
  "slots": [
    {
      "available": false,
      "index": 0,
      "mapped_ip": "172.29.142.26",
      "share_volume": true,
      "count": 1,
      "service_containers": {
        "5d379fac6e37d85f2bdeff45": {
          "connector_id": "5d379fac6e37d85f2bdeff44",
          "service_id": "5d379fac6e37d85f2bdeff45",
          "container_id": "2c7a7ed4f853e85f3d620c663f1c7f5395b53b9dd6696276ac439d34fe142bf1",
          "image_name": "netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow:5d379fac6e37d85f2bdeff45",
          "container_name": "nf-5d379fac6e37d85f2bdeff45",
          "service_type": "NETFLOW_SENSOR",
          "ip_bindings": [
            {
              "ip": "172.29.142.26",
              "port": "4729",
              "protocol": "udp"
            },
            {
              "ip": "172.29.142.26",
              "port": "4739",
              "label": 1,
              "protocol": "udp"
            }
          ]
        }
      },
      "volume_id": "373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439"
    }
  ],
  {
    "available": true,
    "index": 1,
    "mapped_ip": "172.29.142.27",
    "share_volume": true,
    "count": 0,
    "service_containers": null
  },
  {
    "available": true,
    "index": 2,
    "mapped_ip": "172.29.142.28",
    "share_volume": true,
    "count": 0,
    "service_containers": null
  }
]
}[root@beretta-ingest-1 tetter]#
```

Allowed Configurations

- *NTP*: Configure NTP on the appliance. For more information, see [NTP Configuration](#).
- *Log*: Configure Logging on the appliance. For more information, see [Log Configuration](#).

Secure Workload Edge

Secure Workload Edge is a control appliance that streams alerts to various notifiers and collects inventory metadata from network access controllers such as Cisco ISE. In a Secure Workload Edge appliance, all alert notifier connectors (such as Syslog, Email, Slack, PagerDuty, and Kinesis), ServiceNow connector, Workload AD connector, and ISE connector can be deployed.

Specification

- Number of CPU cores: 8
- Memory: 8 GB
- Storage: 250 GB
- Number of network interfaces: 1
- Number of connectors on one appliance: 8
- Operating System: CentOS 7.9 (Secure Workload 3.8.1.19 and earlier), AlmaLinux 9.2 (Secure Workload 3.8.1.36 and later)

See important limits at [Secure Workload Virtual Appliances for Connectors](#).



Note Each root scope on Secure Workload can have at most one Secure Workload Edge appliance deployed.

Figure 46: Secure Workload Edge appliance

The screenshot displays the Cisco Tetration Virtual Appliance management interface. At the top, it shows the 'Tetration Edge Appliance' is in an 'ACTIVE' state. Below this, there are three columns for 'Checked In' (Jul 28 2019 11:28:00 am (PDT)), 'Registered' (Jul 26 2019 11:30:25 am (PDT)), and 'Created' (Jul 26 2019 11:06:27 am (PDT)). A 'Decommission' button is visible in the top right. The main area is divided into 'Connectors' and 'Info' sections. The 'Connectors' section lists: Syslog, Email, Slack, Pager Duty, Kinesis, and ISE, each with a green checkmark. Below the list is a '+ Enable Another Connector' button. The 'Info' section contains a description: 'Tetration Edge is a control appliance that streams alerts to various notifiers and collects inventory metadata from network access controllers such as Cisco ISE.' To the right of the text is a diagram showing a 'Tetration Cluster' connected to a 'Tetration Edge Appliance' via 'Secure Channel Encryption'. The 'Tetration Edge Appliance' is then connected to various notification services: Slack, PagerDuty, and others. At the bottom, there are instructions for deploying the appliance, starting with: '1. Use the OVA image downloaded from Cisco Software Download page to deploy a new OVF template on the designated ESXi host.'

The connectors deployed on Secure Workload Edge appliance do not listen on ports. Therefore, the Docker containers instantiated for the connectors on Secure Workload Edge appliance do not expose any ports to the host.

Allowed Configurations

- *NTP*: Configure NTP on the appliance. For more information, see [NTP Configuration](#).
- *Log*: Configure Logging on the appliance. For more information, see [Log Configuration](#).

Deploying a Virtual Appliance

You will deploy virtual appliances on an ESXi host in VMware vCenter or other KVM-based hypervisors such as Red Hat Virtualization. This procedure will prompt you to download virtual appliance OVA template or QCOW2 image from the [Cisco Software Download page](#).



Attention To deploy a Secure Workload external appliance, the ESXi host where the appliance is created should have the following specifications:

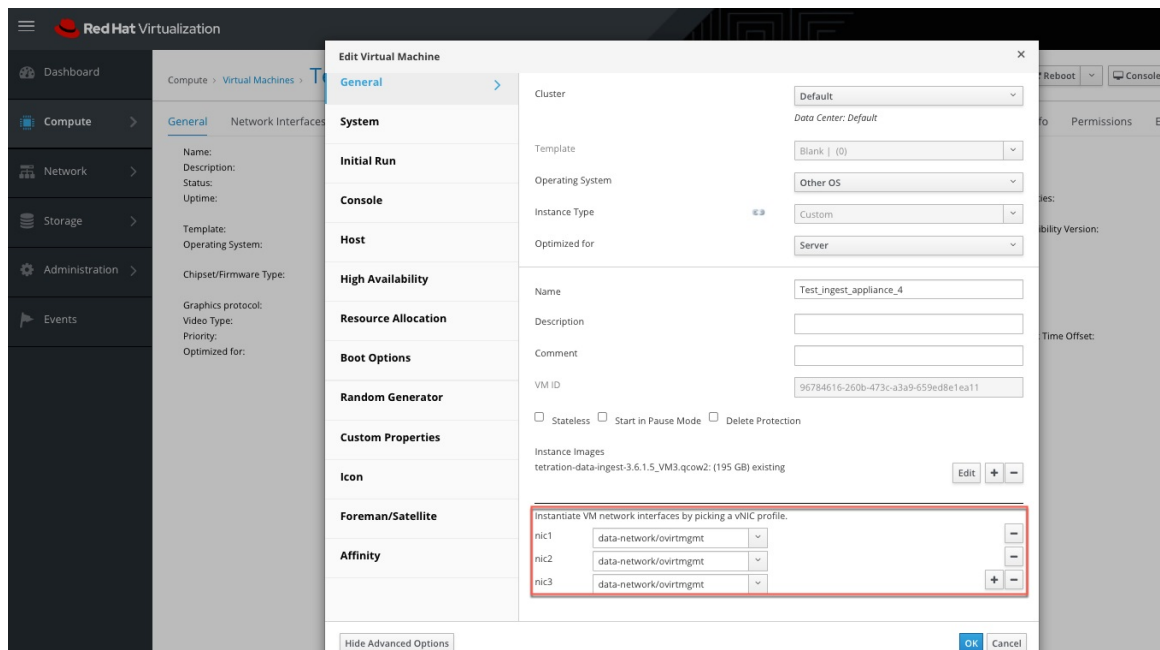
- **vSphere**: version 5.5 or better.
- **CPU**: at least 2.2 GHz per core, and has enough reservable capacity for the appliance.
- **Memory**: at least enough space to fit the appliance.

To deploy a virtual appliance to collect data from connectors:

-
- Step 1** In the Secure Workload web portal, choose **Manage > Virtual Appliances** from the navigation bar on the left.
- Step 2** Click **Enable a Connector**. The type of virtual appliance you need to deploy depends on the type of connector you are enabling.
- Step 3** Click the type of connector for which you need to create the virtual appliance. For example, click the NetFlow connector.
- Step 4** On the connector page, click **Enable**.
- Step 5** If you see a notice telling you that you need to deploy a virtual appliance, click **Yes**. If you do not see this notice, you may already have a virtual appliance that this connector can use, in which case you do not need to perform this procedure.
- Step 6** Click the link to download the OVA template or QCOW2 image for the virtual appliance. Leave the wizard open on your screen without clicking anything else.
- Step 7** Use the downloaded:
- OVA to deploy a new OVF template on a designated ESXi host.
 - To deploy an OVA on a vSphere Web Client, follow the instructions on how to [Deploy an OVF Template](#).
 - Ensure that the deployed VM settings match the recommended configuration for the virtual appliance type.
 - **Do not power on the deployed VM**
 - QCOW2 image to create a new VM on KVM hypervisors such as Red Hat Virtualization.
- Step 8** After the VM is deployed, but before you power it on, return to the virtual appliance deployment wizard in the Secure Workload web portal.
- Step 9** Click **Next** in the virtual appliance deployment wizard.

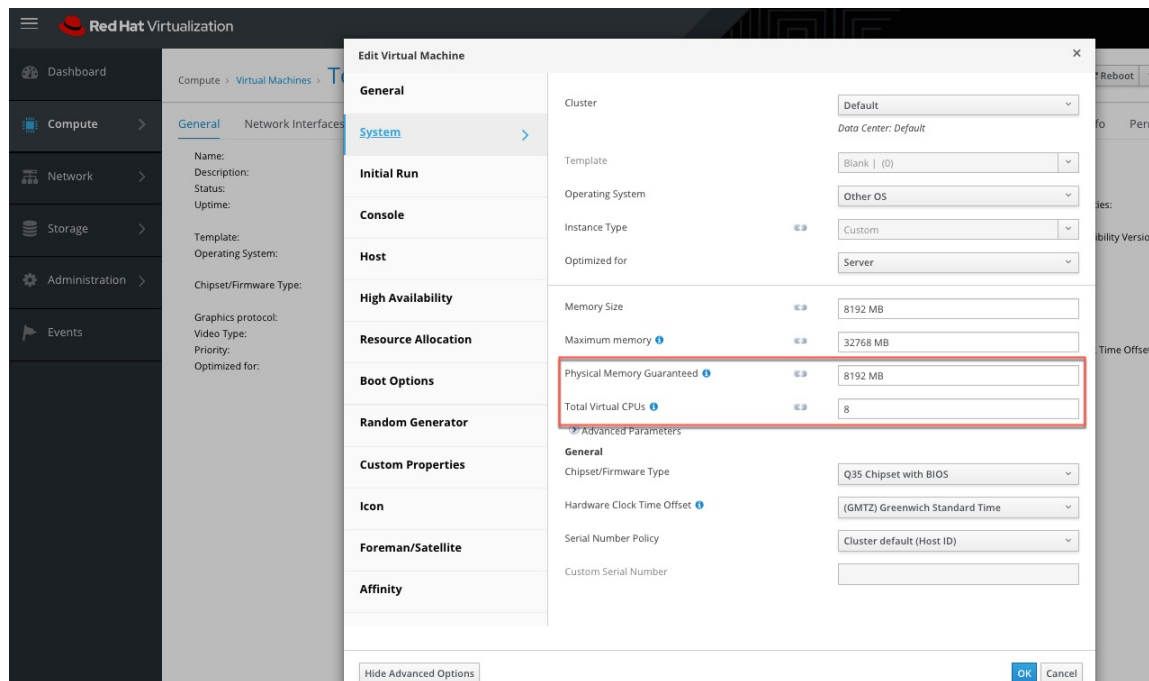
- Step 10** Configure the virtual appliance by providing IP address(es), gateway(s), hostname, DNS, proxy server settings and docker bridge subnet configuration. See the screenshot for *Configuring the VM with network parameters*.
- If the appliance needs to use proxy server to reach Secure Workload, check the box *Use proxy server to connect to Secure Workload*. If this is not set correctly, connectors may not be able to communicate with Secure Workload for control messages, register connectors, and send flow data to Secure Workload collector.
 - If the IP address(es) and gateways(s) of the appliance conflict with the default docker bridge subnet (*172.17.0.1/16*), the appliance can be configured with a customized docker bridge subnet specified in *Docker Bridge (CIDR format)* field. This requires appliance OVA 3.3.2.16 or later.
- Step 11** Click **Next**.
- Step 12** In the next step, a VM configuration bundle will be generated and available for download. Download the VM configuration bundle. See the screenshot for *Download the VM configuration bundle*.
- Step 13** Upload the VM configuration bundle to the datastore corresponding to the target ESXi host or other virtualization host.
- Step 14** [Applicable only when using QCOW2 image] Complete the following configurations on the other virtualization host where you have uploaded the VM configuration bundle:
- For ingest appliances, configure three network interfaces.

Figure 47: Example of configuring network interfaces in KVM-based environments



- In the memory allocation, specify the minimum requirement of 8192 MB of RAM.
- Specify the total number of virtual CPUs to be 8.

Figure 48: Example of configuring system resources in KVM-based environments



Step 15 Edit the VM settings and mount the VM configuration bundle from the datastore to the CD/DVD drive. Make sure to select **Connect at Power On** checkbox.

Step 16 Power on the deployed VM.

Step 17 Once the VM boots up and configures itself, it will connect back to Secure Workload. This may take a few minutes. The appliance status on Secure Workload should transition from *Pending Registration* to *Active*. See the screenshot for *Secure Workload Ingest appliance in Pending Registration state*.

Note We do not recommend vMotion to be enabled for Secure Workload external appliances.

Note We recommend to use Secure Workload external appliance OVAs as-is and to reserve 8 vCPU cores and 8192 MB of memory for QCOW2 images to deploy VMs. If sufficient resources are not available, the VM setup script would fail after the boot.

Once the appliance is *Active*, connectors can be enabled and deployed on it.

Figure 49: playing a Secure Workload Ingest appliance

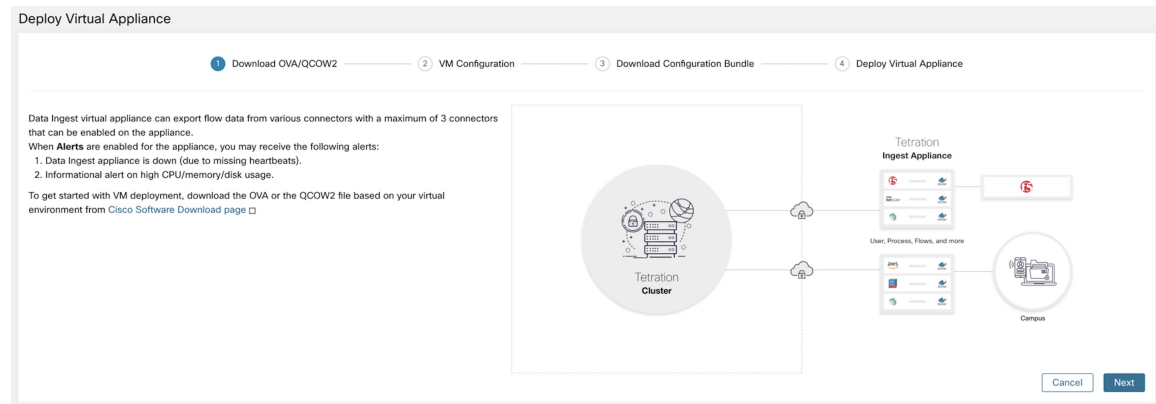


Figure 50: Configuring the VM with network parameters

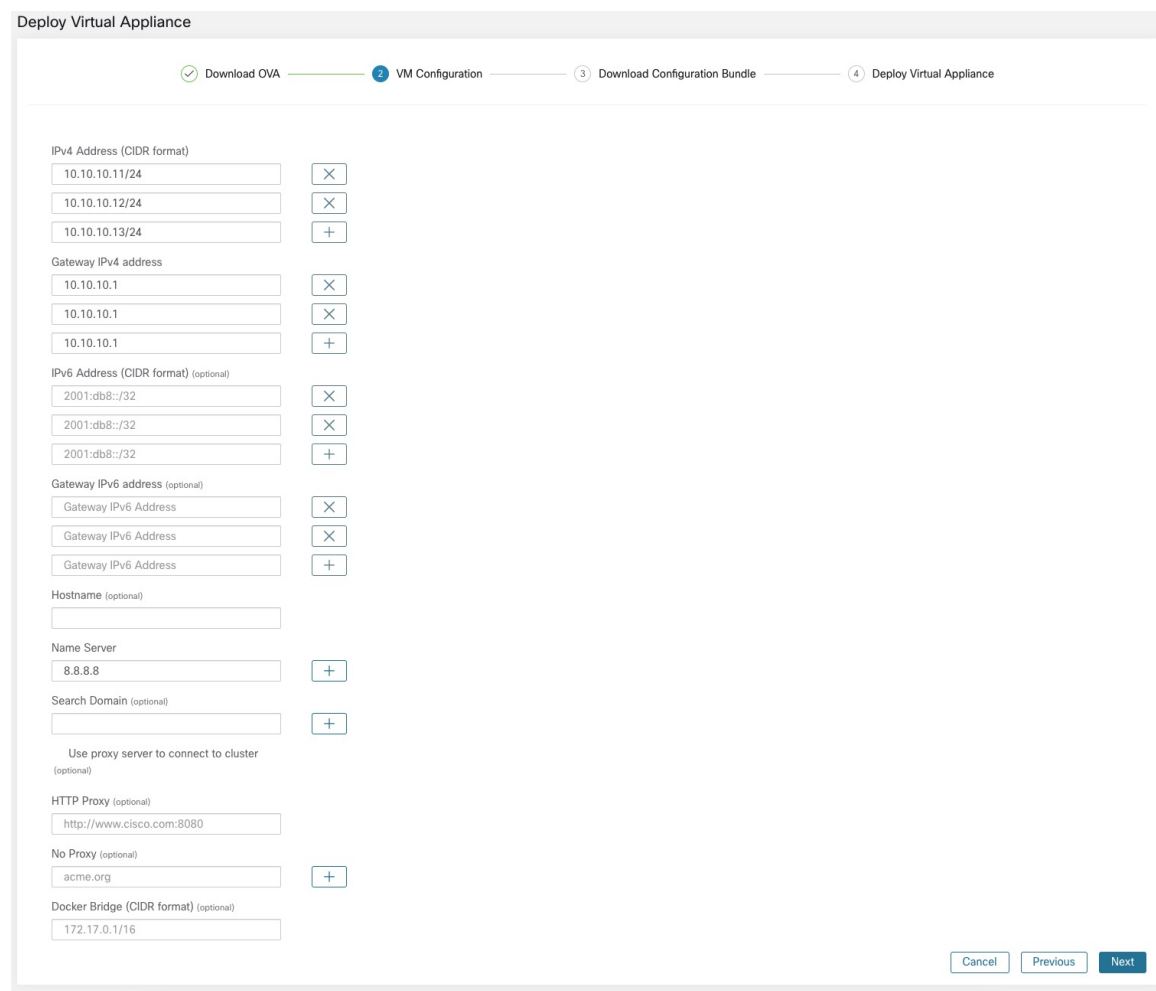


Figure 51: Download the VM configuration bundle

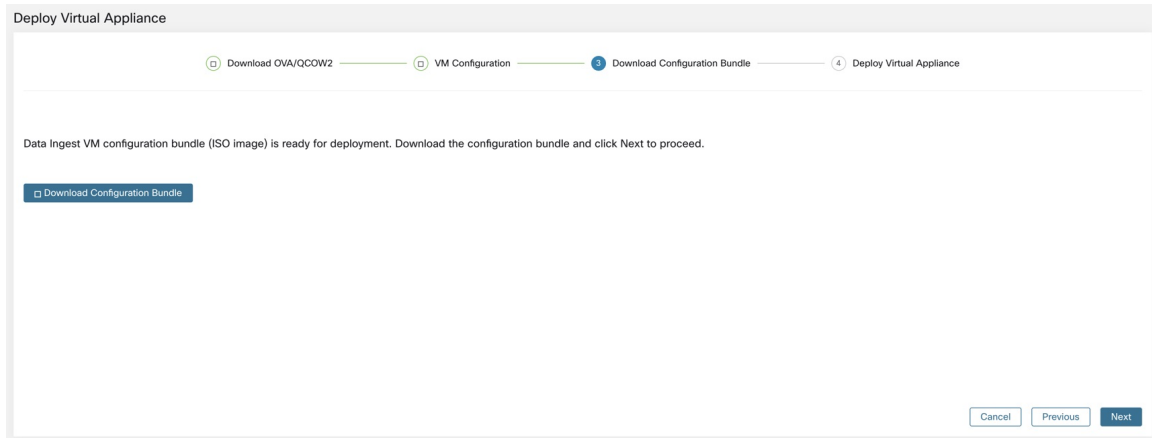


Figure 52: Deploy the VM

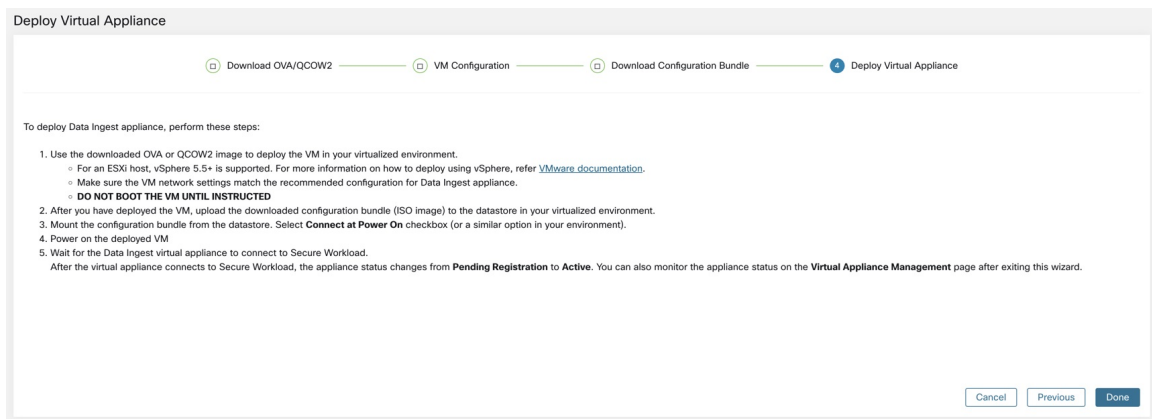
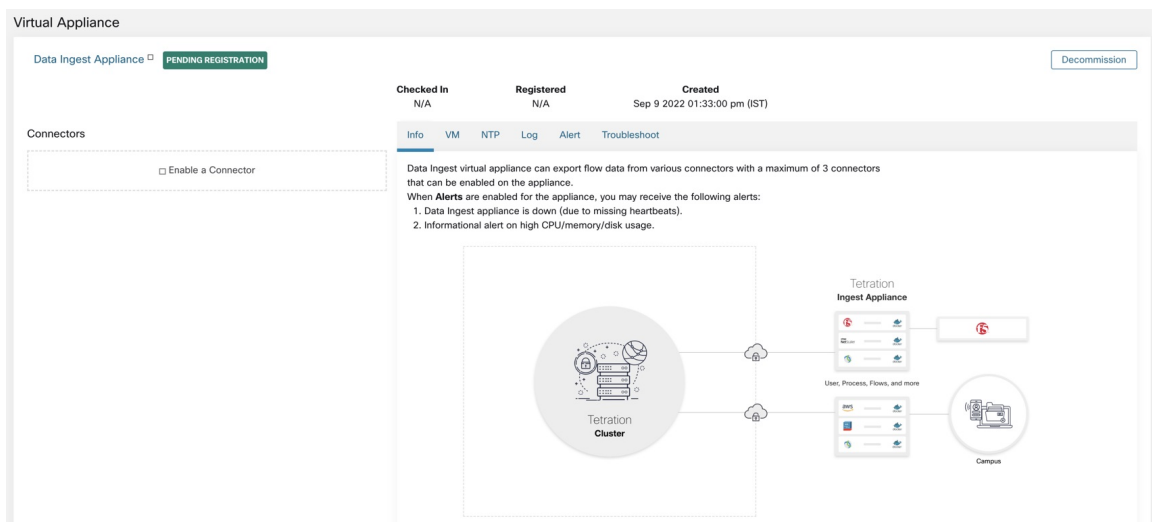


Figure 53: Secure Workload Ingest appliance in Pending Registration state



When a virtual appliance is deployed and booted up for the first time, *tet-vm-setup* service executes and sets up the appliance. This service is responsible for the following tasks

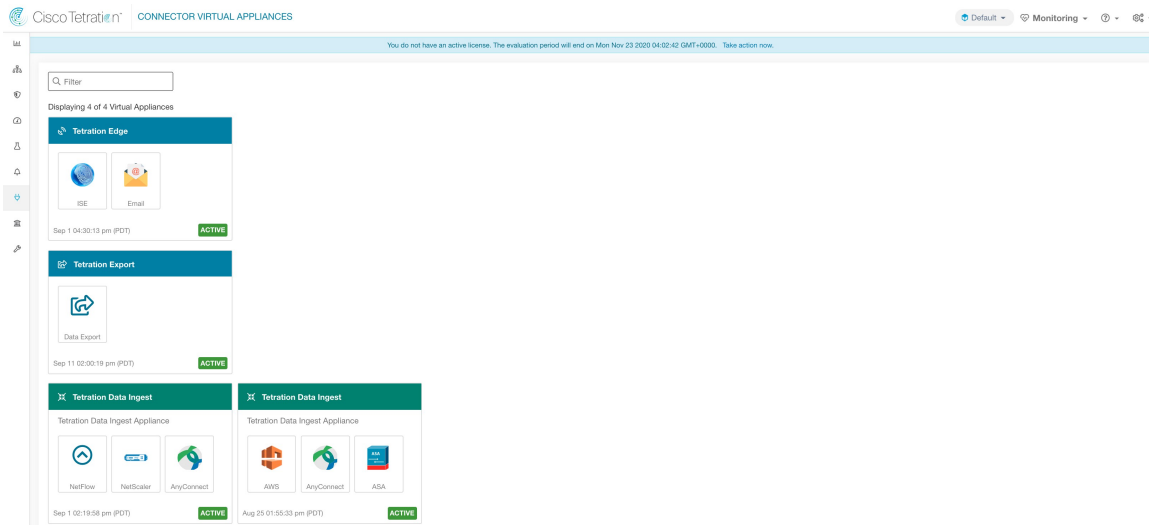
- a. **Validate the appliance:** validate the appliance for mandatory resource requirements for the type of the virtual appliance deployed.
- b. **IP address assignment:** assign IP addresses to all the network interfaces provisioned on the appliance.
- c. **Hostname assignment:** assign hostname for the appliance (if hostname is configured).
- d. **DNS configuration:** update the DNS *resolv.conf* file (if *nameserver* and/or *search-domain* parameters are configured).
- e. **Proxy server configuration:** update *HTTPS_PROXY* and *NO_PROXY* settings on the appliance (if provided).
- f. **Prepare appliance:** copies cert bundle for the Kafka topic over which appliance management messages are sent and received.
- g. **Install appliance controller:** install and bring up *Appliance Controller* which is managed by *supervisord* as *tet-controller* service.

Once *tet-controller* is instantiated, it takes over the management of the appliance. This service is responsible for the following functions:

- a. **Registration:** registers the appliance with Secure Workload. Until the appliance is registered, no connectors can be enabled on the appliance. When Secure Workload receives a registration request for an appliance, it updates the state of the appliance to *Active*.
- b. **Deploying a connector:** deploys a connector as a Docker service on the appliance. For more information, see [Enabling a Connector](#).
- c. **Deleting a connector:** stops and removes the Docker service and the corresponding Docker image from the appliance. For more information, see [Deleting a Connector](#).
- d. **Configuration updates on appliances:** tests and applies configuration updates on the appliance. For more information, see [Configuration Management on Connectors and Virtual Appliances](#).
- e. **Troubleshooting commands on appliances:** executes allowed set of commands on the appliances for troubleshooting and debugging issues on the appliance. For more information, see the [Troubleshooting](#).
- f. **Heartbeats:** periodically sends heartbeats and statistics to Secure Workload to report the health of the appliance. For more information, see [Monitoring a Virtual Appliance](#).
- g. **Pruning:** periodically prune all Docker resources that are unused or dangling in order to recover storage space. This task is executed once every 24 hours.
- h. **Decommissioning the appliance:** decommissions and deletes all Docker instances from the appliance. For more information, see [Decommissioning a Virtual Appliance](#).

The list of deployed virtual appliances can be found at: **Manage > Virtual Appliances**

Figure 54: List of deployed virtual appliances



Decommissioning a Virtual Appliance

A virtual appliance can be decommissioned from Secure Workload. When an appliance is decommissioned, the following actions are triggered.

1. All configurations on the appliance and the connectors enabled on the appliance are removed.
2. All the connectors enabled on the appliance are deleted.
3. The appliance is marked *Pending Delete*.
4. When the appliance replies back with a successful delete response, appliance Kafka topic and certs are deleted.



Note Decommissioning an appliance cannot be undone. To restore the appliance and the connectors, a new appliance should be deployed and the connectors should be enabled on the new appliance.

Monitoring a Virtual Appliance

Secure Workload virtual appliances periodically send heartbeats and statistics to Secure Workload. The heartbeat interval is 5 minutes. The heartbeat messages include statistics about the health of the appliance include system statistics, process statistics, and statistics about how many messages sent/received/error-ed over the Kafka topic that is used for the appliance management.

All metrics are available in *Digger* (OpenTSDB) and are labelled with appliance ID and root scope name. Additionally, Grafana dashboards for *Appliance Controller* are also available for important metrics from the appliance.

Security Considerations

The Ingest/Edge Virtual Machine's guest Operating System is CentOS 7.9, from which OpenSSL server/clients packages were removed. Therefore, the only way to access the appliance is via its console.



Note CentOS 7.9 is the guest operating system for Ingest and Edge virtual appliances in Secure Workload 3.8.1.19 and earlier releases. Starting Secure Workload 3.8.1.36, the operating system is AlmaLinux 9.2.

The containers run a centos:7.9.2009 based Docker image. Most the containers are run with the base privileges (no-privileged option), except for ERSPAN container, which has the NET_ADMIN capability.



Note Starting Secure Workload 3.8.1.36, the containers run almalinux/9-base:9.2.

In the unlikely case a container is compromised, the VM guest OS should not be compromisable from inside the container.

Life Cycle Management of Connectors

Connectors can be enabled, deployed, configured, troubleshooted, and deleted from Secure Workload directly.

Enabling a Connector

From the Connectors page (**Manage > Connectors**), a connector can be selected and enabled. The connector can be deployed on a new virtual appliance (which has to be provisioned first and become *Active* before a connector can be enabled on it) or an existing virtual appliance. Once the virtual appliance is chosen, Secure Workload sends the rpm package for the connector to the appliance.

When Appliance Controller on the chosen appliance receives the rpm, it does the following:

1. Construct a Docker image using the rpm package received from Secure Workload. This Docker image includes the configuration required to communicate with Kafka topic on which appliance management messages are sent. This enables the service instantiated from this image to be able to send and receive messages for managing the corresponding connector.
2. Create a Docker container from the Docker image.
3. On Secure Workload Ingest appliance, the following additional tasks are performed.
 - A free slot is identified and the corresponding IP address is determined.
 - Connector listening ports (for example, 4729 and 4739 ports on NetFlow connector to receive flow records from NetFlow V9 or IPFIX enabled switches and routers), are exposed to the host on IP corresponding to the chosen slot.
 - A Docker volume is created and added to the container.

- The Docker container is started and it executes the connector as a *supervisord* managed service. The service starts *Service Controller* as *tet-controller* which registers with Secure Workload and spawns the actual connector service.

Figure 55: Docker Images

```
[root@beretta-ingest-1 tetter]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow	5d379fac6e37d85f2bdeff45	2635145b44c8	About a minute ago	650MB
tet-service-base	latest	6be171bbe648	4 days ago	519MB
artifacts.tet.wtf:6555/centos	7.3.1611	c5d48e81b986	4 months ago	192MB

```
[root@beretta-ingest-1 tetter]#
```

Figure 56: Docker Volumes

```
[root@beretta-ingest-1 tetter]# docker volume ls
```

DRIVER	VOLUME NAME
local	373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439

```
[root@beretta-ingest-1 tetter]#
```

Figure 57: Docker containers

```
[root@beretta-ingest-1 tetter]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATE
D	STATUS	PORTS	NAMES
2c7a7ed4f853	netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow:5d379fac6e37d85f2bdeff45	"/usr/bin/supervisor..."	About
a minute ago	Up About a minute	172.29.142.26:4729->4729/udp, 172.29.142.26:4739->4739/udp	nf-5d379fac6e37d85f2bdeff45

```
[root@beretta-ingest-1 tetter]#
```

Figure 58: Slot used by the Docker container and list of exposed ports

```
[root@beretta-ingest-1 tetter]# cat /local/tetration/appliance/appliance.conf
{
  "type": "TETRATION_DATA_INGEST",
  "slots": [
    {
      "available": false,
      "index": 0,
      "mapped_ip": "172.29.142.26",
      "share_volume": true,
      "count": 1,
      "service_containers": {
        "5d379fac6e37d85f2bdeff45": {
          "connector_id": "5d379fac6e37d85f2bdeff44",
          "service_id": "5d379fac6e37d85f2bdeff45",
          "container_id": "2c7a7ed4f853e85f3d620c663f1c7f5395b53b9dd6696276ac439d34fe142bf1",
          "image_name": "netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow:5d379fac6e37d85f2bdeff45",
          "container_name": "nf-5d379fac6e37d85f2bdeff45",
          "service_type": "NETFLOW_SENSOR",
          "ip_bindings": [
            {
              "ip": "172.29.142.26",
              "port": "4729",
              "protocol": "udp"
            },
            {
              "ip": "172.29.142.26",
              "port": "4739",
              "label": 1,
              "protocol": "udp"
            }
          ]
        },
        "volume_id": "373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439"
      }
    },
    {
      "available": true,
      "index": 1,
      "mapped_ip": "172.29.142.27",
      "share_volume": true,
      "count": 0,
      "service_containers": null
    },
    {
      "available": true,
      "index": 2,
      "mapped_ip": "172.29.142.28",
      "share_volume": true,
      "count": 0,
      "service_containers": null
    }
  ]
}
[root@beretta-ingest-1 tetter]#
```

Figure 59: List of ports exposed by Docker container

```
[root@beretta-ingest-1 tetter]# docker port 2c7a7ed4f853
4729/udp -> 172.29.142.26:4729
4739/udp -> 172.29.142.26:4739
[root@beretta-ingest-1 tetter]#
```

Figure 60: Docker Volume mounted to a container

```
[root@beretta-ingest-1 tetter]# docker inspect --format='{{json .Mounts}}' 2c7a7ed4f853
[{"Type": "volume", "Name": "373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439", "Source": "/var/lib/docker/volumes/373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439/_data", "Destination": "/local/tetration", "Driver": "local", "Mode": "z", "RW": true, "Propagation": ""}]
[root@beretta-ingest-1 tetter]#
```

Service Controller is responsible for the following functions:

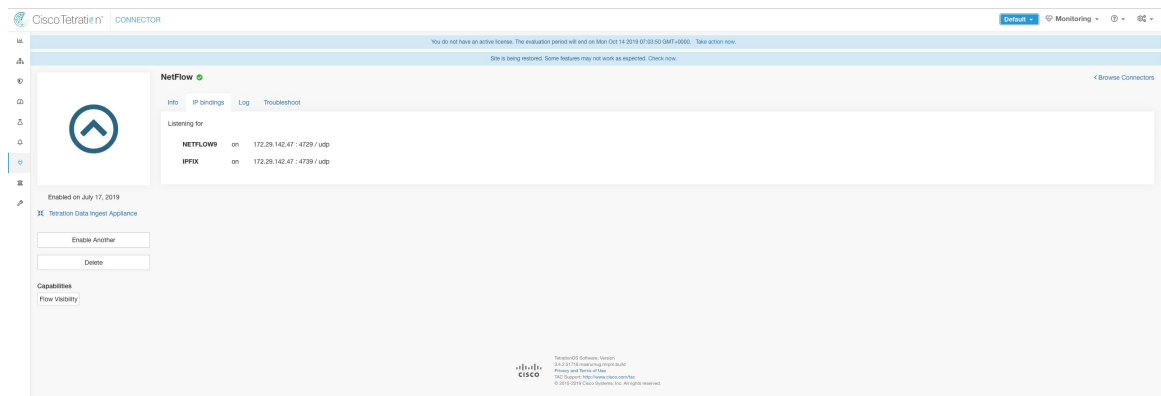
1. **Registration:** registers the connector with Secure Workload. Until the connector is registered and marked *Enabled*, no configuration updates can be pushed to the connector. When Secure Workload receives a registration request for a connector, it updates the state of the connector to *Enabled*.
2. **Configuration updates on connector:** tests and applies configuration updates on the connector. For more information, see [Configuration Management on Connectors and Virtual Appliances](#).
3. **Troubleshooting commands on connector:** executes allowed commands on the connector service for troubleshooting and debugging issues on the connector service. For more information, see [Troubleshooting](#).
4. **Heartbeats:** periodically sends heartbeats and statistics to Secure Workload to report the health of the connector. For more information, see [Monitoring a Virtual Appliance](#).

Viewing Connector-Related Information

Enabled Connectors: A list of all enabled connectors can be found by clicking **Manage > Connectors** in the navigation bar at the left side of the window.

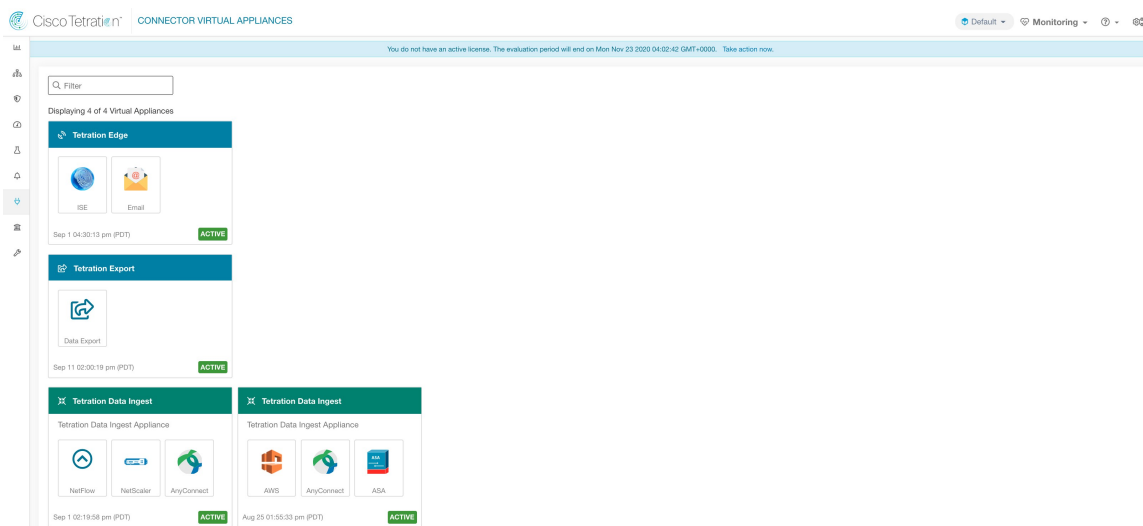
Connector Details: Details about the connector can be fetched by clicking on the connector. This page shows the port bindings -if any- that can be used to configure upstream network elements to send telemetry data to the correct IP and port.

Figure 61: Connector details



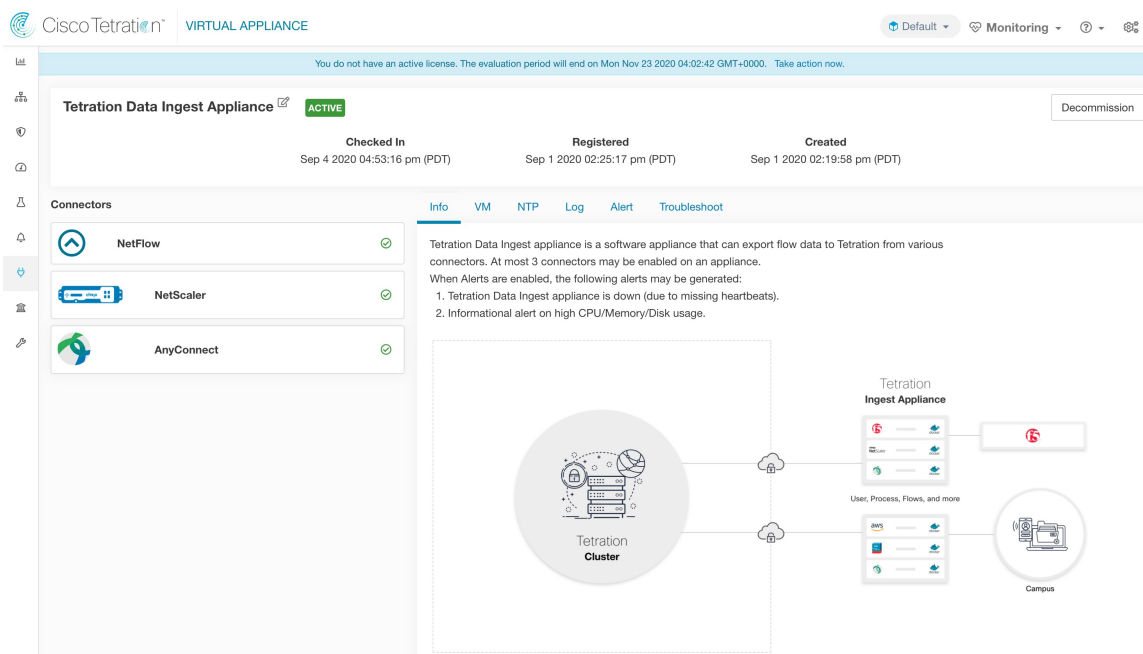
Deployed Virtual Appliances: A list of deployed virtual appliances can be found at: **Manage > Virtual Appliances**.

Figure 62: List of deployed virtual appliances



Virtual Appliance Details A detailed view of an appliance can be fetched by clicking on the appliance directly from *List of deployed virtual appliances*.

Figure 63: Appliance details and the connectors



Deleting a Connector

When a connector is deleted, Appliance Controller on the appliance where the connector is enabled will receive a message to remove the services created for the connector. Appliance Controller does the following:

1. Stop the Docker container corresponding to the connector.

2. Remove the Docker container.
3. If the connector is deployed on a Secure Workload Ingest appliance and it exposes ports, then remove the Docker volume that was mounted to the container.
4. Remove the Docker image that was created for the connector.
5. Finally, send a message back to Secure Workload indicating the status of the delete request.

Monitoring a Connector

Connector services periodically send heartbeats and statistics to Secure Workload. The heartbeat interval is 5 minutes. The heartbeat messages include statistics about the health of the service include system statistics, process statistics, and statistics about how many messages sent/received/error-ed over the Kafka topic that is used for the appliance management. In addition, it includes statistics exported by the connector service itself.

All metrics are available in *Digger* (OpenTSDB) and are annotated with appliance ID, connector ID, and root scope name. Additionally, Grafana dashboards for connector services are also available for important metrics from the service.

Configuration Management on Connectors and Virtual Appliances

Configuration updates can be pushed to appliances and connectors from Secure Workload. The appliance should have registered successfully with Secure Workload and be *Active* before configuration updates can be initiated. Similarly, the connectors should have registered with Secure Workload before configuration updates can be initiated on the connector services.

There are three modes of configuration updates possible in appliances and connectors.

1. **Test and Apply:** Test the configuration and on successful test, commit the configuration.
2. **Discovery:** Test the configuration, and on successful test, discovery additional properties that can be enabled for the configuration.
3. **Remove:** Remove the configuration.



Note ERSPAN appliance and connector do not support configuration updates.

Test and Apply

Configurations that support *Test and Apply* mode verify the configuration before applying (committing) the configuration on the desired appliance and/or connector.

NTP Configuration

NTP configuration allows the appliance to synchronize the clock with the specified NTP server(s).

Parameter Name	Type	Description
Enable NTP	checkbox	Should NTP sync be enabled?
NTP Servers	listof strings	List of NTP servers. At least one server should be given and at most 5 servers may be provided.

Test: Test if a UDP connection can be made to the given NTP servers on port 123. If an error occurs for any of the NTP servers, do not accept the configuration.

Apply: Update `/etc/ntp.conf` and restart `ntpd` service using `systemctl restart ntpd.service`. Here is the template for generating the `ntp.conf`

```
# --- GENERAL CONFIGURATION ---
server <ntp-server>
...
server 127.127.1.0
fudge 127.127.1.0 stratum 10
# Drift file
driftfile /etc/ntp/drift
```



Note Applicable to Secure Workload 3.8.1.19 and earlier.

For Secure Workload 3.8.1.36 and later, update `/etc/chrony.conf` and restart `chronyd` service using `systemctl restart chronyd.service`. Here is the template for generating the `chrony.conf`

```
# Secure Workload appliance chrony.conf.
server <ntp-server> iburst
...
driftfile /var/lib/chrony/drift
makestep 1.0 3
rtcsync
```

Allowed Cisco Secure Workload virtual appliances: All

Allowed connectors: None

Figure 64: Error while testing NTP configuration

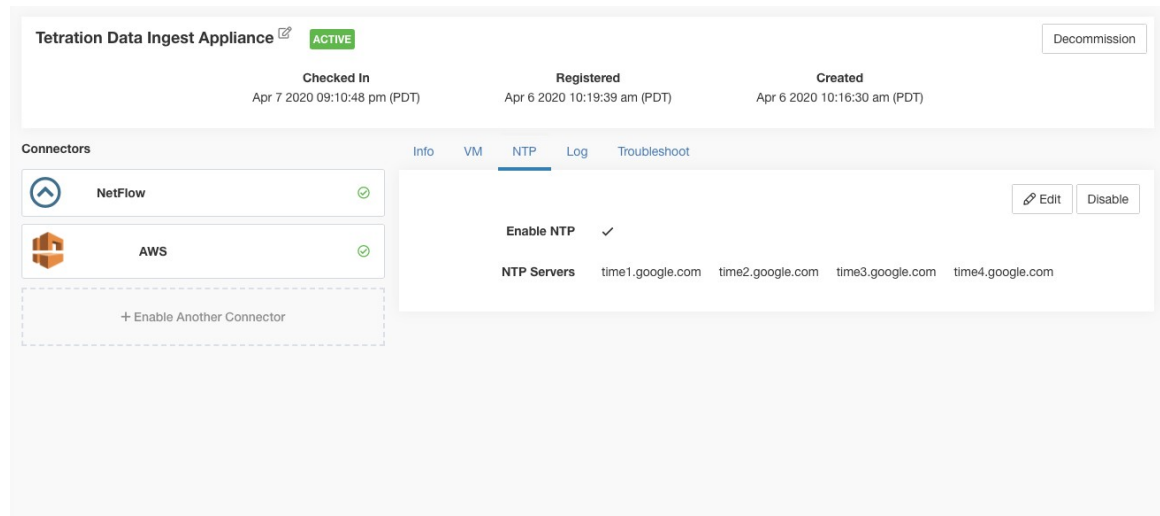
The screenshot shows the configuration page for a 'Tetration Data Ingest Appliance'. At the top, it indicates the appliance is 'ACTIVE' and provides a 'Decommission' button. Below this, the status is shown as 'Checked In' (Apr 7 2020 09:05:45 pm (PDT)), 'Registered' (Apr 6 2020 10:19:39 am (PDT)), and 'Created' (Apr 6 2020 10:16:30 am (PDT)).

The 'Connectors' section on the left lists 'NetFlow' and 'AWS', both with green checkmarks, and a '+ Enable Another Connector' button. The main 'NTP' configuration area has tabs for 'Info', 'VM', 'NTP', 'Log', and 'Troubleshoot'. The 'NTP' tab is active, showing 'Enable NTP' checked and a single 'NTP Servers (optional)' field containing 'a.b.com'. A red error message is displayed: 'Error: could not connect to server a.b.com: dial udp: lookup a.b.com on 171.70.168.183:53: no such host'. At the bottom, there are 'Cancel Config Creation' and 'Verify & Save Configs' buttons.

Figure 65: NTP configuration with valid NTP servers

This screenshot shows the same configuration page as Figure 64, but with four valid NTP servers entered in the 'NTP Servers (optional)' field: 'time1.google.com', 'time2.google.com', 'time3.google.com', and 'time4.google.com'. Each server entry has a small 'x' button to its right. The 'Error' message is no longer present. The 'Verify & Save Configs' button is now blue, indicating a successful configuration.

Figure 66: NTP configuration verified and applied



Log Configuration

Log configuration updates the log levels, maximum size of the log files, and log rotation parameters on the appliance and/or connector. If the configuration update is triggered on the appliance, appliance controller log settings are updated. On the other hand, if the configuration update is triggered on a connector, service controller and service log settings are updated.

Parameter Name	Type	Description
Logging level	dropdown	Logging level to be set
	• <i>debug</i>	Debug log level
	• <i>info</i>	Informational log level
	• <i>warn</i>	Warning log level
	• <i>error</i>	Error log level
Max log file size (in MB)	number	Maximum size of a log file before log rotation kicks in
Log rotation (in days)	number	Maximum age of a log file before log rotation kicks in
Log rotation (in instances)	number	Maximum instances of log files kept

Test: No op.

Apply: If the configuration is triggered on an appliance, update the configuration file of *tet-controller* on the appliance. If the configuration is triggered on a connector, update the configuration files of *tet-controller* and the service managed by the controller on the Docker container responsible for the connector.

Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, ISE, ASA, and Meraki.

Figure 67: Log configuration on the appliance

The screenshot shows the configuration page for a 'Tetration Data Ingest Appliance'. At the top, it indicates the appliance is 'ACTIVE' and provides a 'Decommission' button. Below this, there are three status indicators: 'Checked In' (Apr 7 2020 09:05:45 pm (PDT)), 'Registered' (Apr 6 2020 10:19:39 am (PDT)), and 'Created' (Apr 6 2020 10:16:30 am (PDT)).

The main configuration area is titled 'Connectors' and includes tabs for 'Info', 'VM', 'NTP', 'Log', and 'Troubleshoot'. The 'Log' tab is selected. On the left, there are two connector cards: 'NetFlow' and 'AWS', both with green checkmarks. Below them is a dashed box with the text '+ Enable Another Connector'. The right side of the page shows the log configuration settings for the selected connector:

- Logging Level:** A dropdown menu with 'debug' selected. Other options are 'info', 'warn', and 'error'.
- Max Log File Size (in MB):** A text input field.
- Log Rotation (in days):** A text input field.
- Log Rotation (in instances):** A text input field with the value '20'.

At the bottom of the configuration area, there are two buttons: 'Cancel Config Creation' and 'Verify & Save Configs'.



Note Since all alert notifier Connectors (Syslog, Email, Slack, PagerDuty, and Kinesis) run on a single Docker service (Secure Workload Alert Notifier) on Secure Workload Edge, it is not possible to update the log config of a connector without impacting the config of another alert notifier connector. The log configurations of Secure Workload Alert Notifier (TAN) Docker service on Secure Workload Edge appliance can be updated using an allowed command.

See [Update Alert Notifier Connector Log Configuration](#) for more details.

Endpoint Configuration

Endpoint configuration specifies the inactivity timeout for endpoints on AnyConnect and ISE connectors. When an endpoint times out, the connector stops checking in with Secure Workload and purges the local state for the endpoint on the connector.

Parameter Name	Type	Description
InactivityTimeout for Endpoints(in minutes)	number	Inactivity timeout for endpoints published by AnyConnect / ISE connectors. On timeout, the endpoint will not longer checkin Secure Workload. Default is 30 minutes.

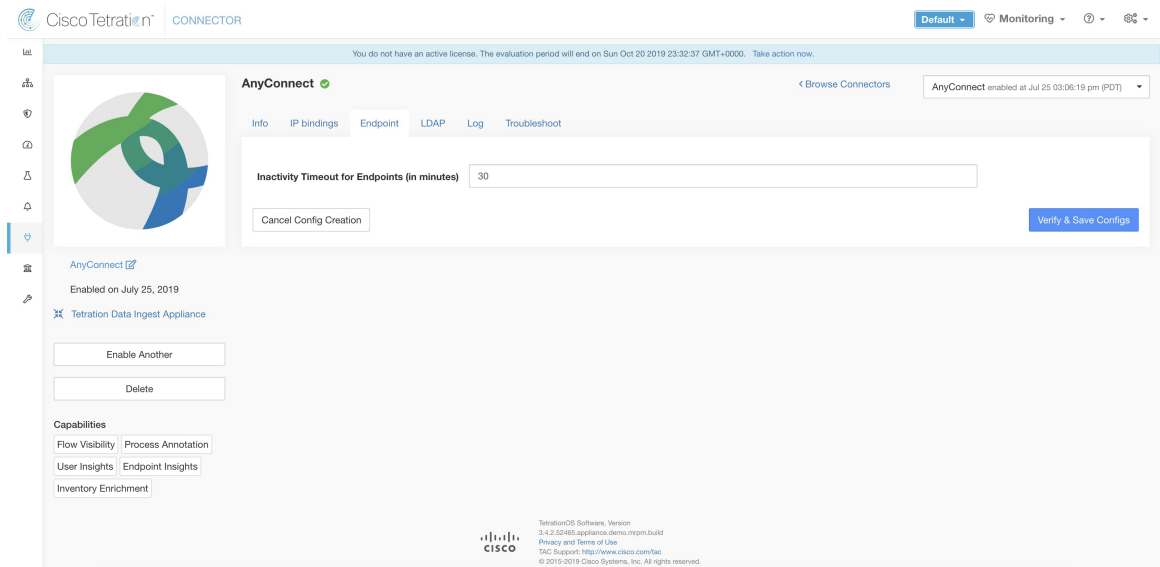
Test : No op.

Apply : Update the configuration file of the connector with the new value

Allowed Secure Workload virtual appliances: None

Allowed connectors: AnyConnect and ISE

Figure 68: Endpoint inactivity timeout configuration on AnyConnect connector



Slack Notifier Configuration

Default configuration for publishing Secure Workload alerts on Slack.

Parameter Name	Type	Description
Slack Webhook URL	string	Slack webhook on which Secure Workload alerts should be published

Test: Send a test alert to Slack using the webhook. If the alert is posted successfully, the test passes.

Apply: Update configuration file of the connector with the specified parameters.

Allowed Secure Workload virtual appliances: None

Allowed connectors: Slack

PagerDuty Notifier Configuration

Default configuration for publishing Secure Workload alerts on PagerDuty.

Parameter Name	Type	Description
PagerDuty Service Key	string	PagerDuty service key for pushing Secure Workload alerts on PagerDuty

Test: Send a test alert to PagerDuty using the service key. If the alert is published successfully, the test passes.

Apply: Update configuration file of the connector with the specified parameters.

Allowed Secure Workload virtual appliances: None

Allowed connectors: PagerDuty

Kinesis Notifier Configuration

Default configuration for publishing Secure Workload alerts on Amazon Kinesis.

Parameter Name	Type	Description
AWS Access Key ID	string	AWS access key ID to communicate with AWS
AWS Secret Access Key	string	AWS secret access key to communicate with AWS
AWS Region	dropdown of AWS regions	Name of the AWS region where Kinesis stream is configured
Kinesis Stream	string	Name of the Kinesis stream
Stream Partition	string	Partition Name of the stream

Test: Send a test alert to the Kinesis stream using the given configuration. If the alert is published successfully, the test passes.

Apply: Update configuration file of the connector with the specified parameters.

Allowed Secure Workload virtual appliances: None

Allowed connectors: Kinesis

Email Notifier Configuration

Default configuration for publishing Secure Workload alerts on Email.

Parameter Name	Type	Description
SMTP Username	string	SMTP server username. This parameter is optional.
SMTP Password	string	SMTP server password for the user (if given). This parameter is optional.
SMTP Server	string	IP address or hostname of the SMTP server
SMTP Port	number	Listening port of SMTP server. Default value is 587.
Secure Connection	checkbox	Should SSL be used for SMTP server connection?
From Email Address	string	Email address to use for sending alerts
Default Recipients	string	Comma separated list of recipient email addresses

Test: Send a test email using the given configuration. If the alert is published successfully, the test passes.

Apply: Update configuration file of the connector with the specified parameters.

Allowed Secure Workload virtual appliances: None

Allowed connectors: Email

Syslog Notifier Configuration

Default configuration for publishing Secure Workload alerts on Syslog.

Parameter Name	Type	Description
Protocol	dropdown	Protocol to use to connect to server
	•UDP	
	•TCP	
Server Address	string	IP address or hostname of the Syslog server
Port	number	Listening port of Syslog server. Default port value is 514.

Test: Send a test alert to Syslog server using the given configuration. If the alert is published successfully, the test passes.

Apply: Update configuration file of the connector with the specified parameters.

Allowed Secure Workload virtual appliances: None

Allowed connectors: Syslog

Syslog Severity Mapping Configuration

The following table shows the default severity mapping for Secure Workload alerts on Syslog

Secure Workload Alerts Severity	Syslog Severity
LOW	LOG_DEBUG
MEDIUM	LOG_WARNING
HIGH	LOG_ERR
CRITICAL	LOG_CRIT
IMMEDIATE ACTION	LOG_EMERG

You can modify this setting using this configuration.

Parameter Name	Dropdown of mappings
IMMEDIATE_ACTION	<ul style="list-style-type: none"> • <i>Emergency</i> • <i>Alert</i> • <i>Critical</i> • <i>Error</i> • <i>Warning</i> • <i>Notice</i> • <i>Informational</i> • <i>Debug</i>
CRITICAL	
HIGH	
MEDIUM	
LOW	

Test: No op.

Apply: Update configuration file of the connector with the specified parameters.

Allowed Secure Workload virtual appliances: None

Allowed connectors: Syslog

ISE Instance Configuration

This configuration provides the parameters required to connect to the Cisco Identity Services Engine (ISE). By providing multiple instances of this configuration, the ISE connector can connect and pull metadata about endpoints from multiple ISE appliances. Up to 20 instances of ISE configuration may be provided.

Parameter Name	Type	Description
ISE Client Certificate	string	ISE client certificate to connect to ISE using pxGrid
ISE Client Key	string	ISE client key to connect to ISE
ISE Server CA Certificate	string	CA certificate of ISE
ISE Hostname	string	FQDN of ISE pxGrid
ISE Nodename	string	Node name of ISE pxGrid

Test: Connect to ISE using the given parameters. On successful connection, accept the configuration.

Apply: Update configuration file of the connector with the specified parameters.

Allowed Secure Workload virtual appliances: None

Allowed connectors: ISE

Discovery

Configurations that support *Discovery* mode do the following.

1. Collect a basic configuration from the user.

2. Verify the basic configuration.
3. Discovery additional properties about the configuration and present them to the user.
4. Let the user enhance the configuration using the discovered properties.
5. Verify and apply the enhanced configuration.

In the 3.3.1.x release, LDAP configuration supports discovery mode.

LDAP Configuration

LDAP configuration specifies how to connect to LDAP, what is the base Distinguished Name (DN) to use, what is the attribute that corresponds to username, and what attributes to fetch for each username. LDAP attributes are properties of LDAP that are specific to that environment.

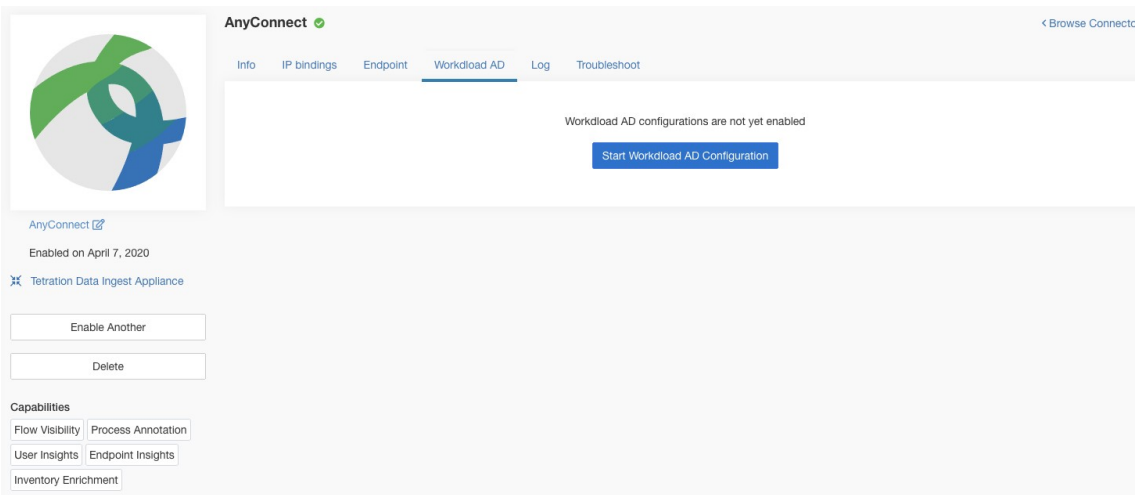
Given the configuration of how to connect to LDAP and the base DN, it is possible to discover the attributes of users in LDAP. These discovered attributes can then be presented to the user in the UI. From these discovered attributes, the user selects the attribute that corresponds to the username and a list of up to six attributes to collect for each username from LDAP. As a result, this eliminates the manual configuration of the LDAP attributes and reduces errors.

Here are the detailed steps for creating LDAP configuration through discovery.

Step 1 Start the LDAP Configuration

Initiate an LDAP configuration for the connector.

Figure 69: Start the LDAP configuration discovery



Step 2 Provide Basic LDAP Configuration

Specify the basic configuration for connecting to LDAP. In this configuration, the users provide the LDAP Bind DN or username to connect to LDAP server, LDAP password to use to connect to LDAP server, LDAP server address, LDAP server port, Base DN to connect to, and a filter string to fetch users that match this filter.

Parameter Name	Type	Description
LDAP Username	string	LDAP username or bind DN to access LDAP server*
LDAP Password	string	LDAP password for the username to access LDAP server*
LDAP Server	string	LDAP server address
LDAP Port	number	LDAP server port
Use SSL	checkbox	Should the connector connect to LDAP securely? Optional. Default is false.
Verify SSL	checkbox	Should the connector verify LDAP cert? Optional. Default is false.
LDAP Server CA Cert	string	Server CA certificate. Optional.
LDAP Server Name	string	Servename for which the LDAP cert is issued (mandatory if <i>Verify SSL</i> is checked).
LDAP Base DN	string	LDAP base DN, the starting point for directory searches in LDAP
LDAP Filter String	string	LDAP filter prefix string. Filter the search result that match only this condition.
Snapshot Sync Interval (in hours)	number	Specify the time interval in hours to (re)create LDAP snapshot. Optional. Default is 24 hours.
Use Proxy to reach LDAP	checkbox	Should the connector use proxy server to access LDAP server?
Proxy Server to reach LDAP	string	Proxy server to access LDAP

Minimum user permissions needed to configure LDAP on Connectors is a **standard domain User**.

Figure 70: Initial LDAP configuration

The screenshot displays the 'AnyConnect' configuration page for 'Workload AD'. The interface is divided into a left sidebar and a main configuration area. The sidebar includes the AnyConnect logo, a status indicator 'Enabled on April 7, 2020', and a list of capabilities: Flow Visibility, Process Annotation, User Insights, Endpoint Insights, and Inventory Enrichment. The main configuration area is titled 'Enter Configs' and contains the following fields:

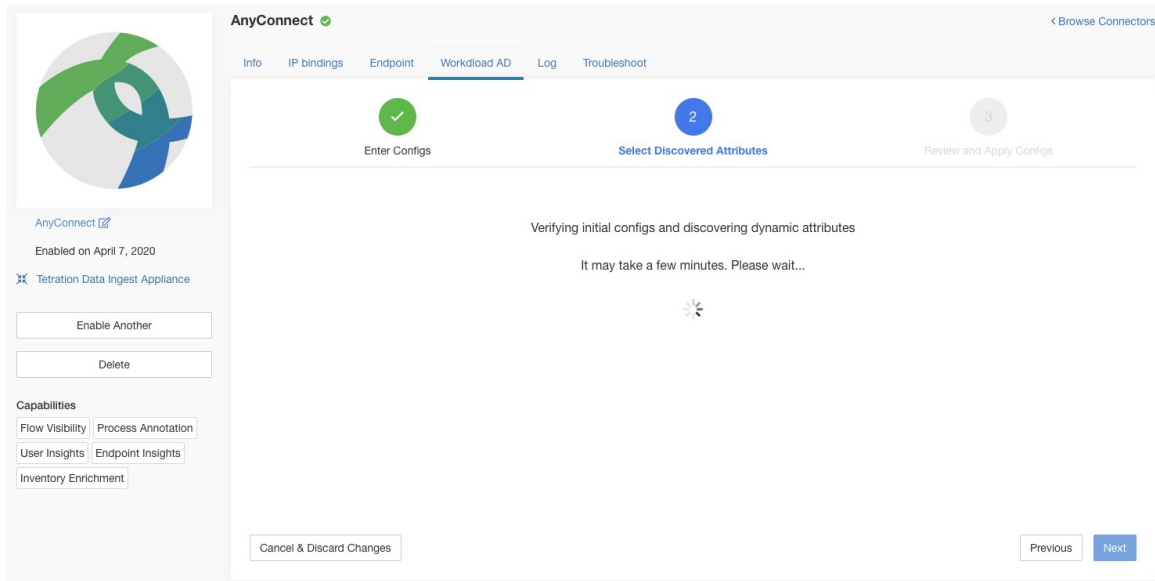
- LDAP Username:** cn=ldapadmin,dc=tetrationanalytics,dc=com
- LDAP Password:** [Redacted]
- LDAP Server:** 172.26.230.174
- LDAP Port:** 389
- Use SSL:**
- Verify SSL:**
- LDAP Server CA Cert (optional):** [Empty text area]
- LDAP Server Name (optional):** Enter LDAP Server Name
- LDAP Base DN:** ou=People,dc=tetrationanalytic,dc=com
- LDAP Filter String:** (&(objectClass=organizationalPerson))
- Snapshot Sync Interval (in hours) (optional):** 24
- Use Proxy to reach LDAP:**
- Proxy Server to reach LDAP (optional):** http://1.1.1.1:8080

Navigation buttons include 'Cancel' and 'Next'.

Step 3 Discovery in Progress

Once the user clicks *Next*, this configuration is sent to the connector. The connector establishes a connection with LDAP server using the given configuration. It fetches up to 1000 users from LDAP server and identifies all the attributes. Furthermore, it computes a list of all the single-valued attributes are common across all 1000 users. The connector returns this result back to Secure Workload.

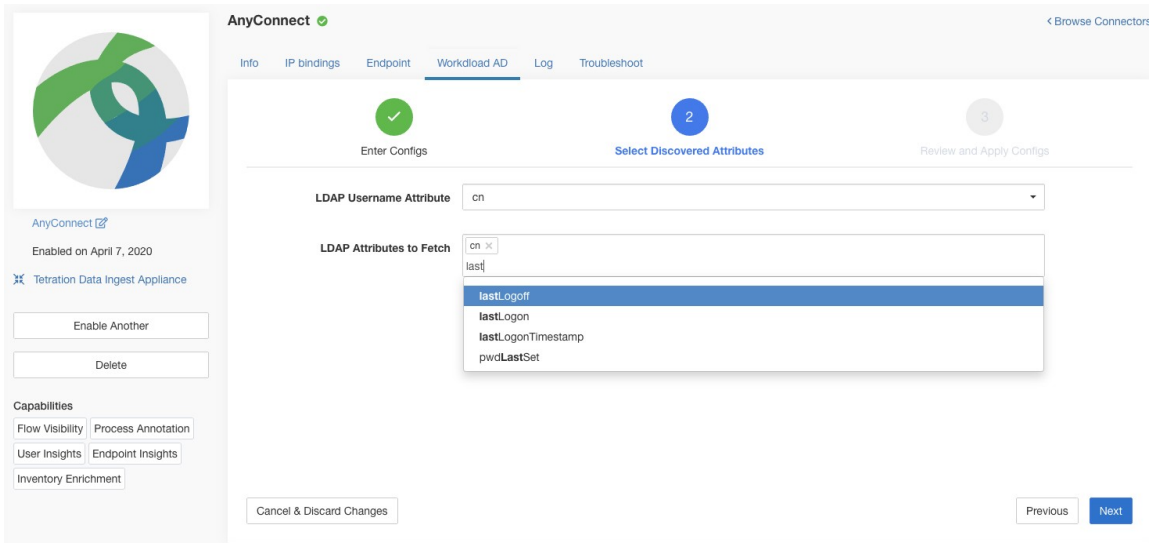
Figure 71: Discovery in progress



Step 4 Enhance the Configuration with Discovered Attributes

The user has to pick which attribute corresponds to username and select up to six attributes that the connector has to fetch and snapshot for each user in the organization (i.e., users matching the filter string). This action is performed using a dropdown of list of discovered attributes. Thus, eliminating manual errors and misconfiguration.

Parameter Name	Type	Description
LDAP Username Attribute	string	LDAP attribute that contains the username
LDAP Attributes to Fetch	list of strings	List of LDAP attributes that should be fetched for a user

Figure 72: Discovered LDAP attributes

The screenshot shows the AnyConnect configuration interface for a Workload AD connector. The interface is divided into three steps: 1. Enter Configs (marked with a green checkmark), 2. Select Discovered Attributes (marked with a blue '2'), and 3. Review and Apply Configs (marked with a grey '3').

On the left sidebar, the connector is identified as 'AnyConnect' and is noted as 'Enabled on April 7, 2020'. It is associated with the 'Tetration Data Ingest Appliance'. Below this, there are buttons for 'Enable Another' and 'Delete'. A 'Capabilities' section includes 'Flow Visibility', 'Process Annotation', 'User Insights', 'Endpoint Insights', and 'Inventory Enrichment'.

The main configuration area shows the 'LDAP Username Attribute' set to 'cn'. Below it, the 'LDAP Attributes to Fetch' list contains 'cn', 'lastLogoff', 'lastLogon', 'lastLogonTimestamp', and 'pwdLastSet'. A 'Cancel & Discard Changes' button is at the bottom left, and 'Previous' and 'Next' buttons are at the bottom right.

Figure 73: Identify username attribute and attributes to collect for each username

Step 5 Finalize, Save, and Apply the Configuration

Finally, the configuration is completed by clicking *Save and Apply Changes*.

Figure 74: Complete LDAP configuration discovery and commit

The figure consists of two screenshots of the AnyConnect configuration interface for LDAP Workload AD.

Top Screenshot: Select Discovered Attributes (Step 2)

- Step 1: Enter Configs (Completed)
- Step 2: Select Discovered Attributes (Active)
- Step 3: Review and Apply Configs (Pending)
- LDAP Username Attribute:
- LDAP Attributes to Fetch:
- Buttons: Cancel & Discard Changes, Previous, Next

Bottom Screenshot: Review and Apply Configs (Step 3)

- Step 1: Enter Configs (Completed)
- Step 2: Select Discovered Attributes (Completed)
- Step 3: Review and Apply Configs (Active)
- LDAP Username: *****
- LDAP Password: *****
- LDAP Server: 172.26.230.174
- LDAP Port: 389
- Use SSL:
- Verify SSL:
- LDAP Server CA Cert: [Empty]
- LDAP Server Name: [Empty]
- LDAP Base DN: ou=People,dc=tetrationanalytics,dc=com
- LDAP Filter String: (&(objectClass=organizationalPerson))
- LDAP Username Attribute: cn
- LDAP Attributes to Fetch: cn description title street displayName
- Snapshot Sync Interval (in hours): 24
- Use Proxy to reach LDAP:
- Proxy Server to reach LDAP: [Empty]
- Buttons: Cancel, Previous, Save & Apply Configs

The connector receives the completed configuration. It creates a local snapshot of all users matching the filter string and fetches only the selected attributes. Once the snapshot is completed, the connector services can start using the snapshot for annotating users and their LDAP attributes in inventories.

Allowed Secure Workload virtual appliances: None

Allowed connectors: AnyConnect, ISE, and F5.

Remove

You can remove all the configurations that you have added from the connectors and/or appliances using the *Delete* button available for each configuration.

Troubleshooting

Connectors and virtual appliances support various troubleshooting mechanisms to debug possible issues.



Note This section does not apply to the following:

ERSPAN virtual appliance: Refer to the ERSPAN appliance page for the troubleshooting details.

Cloud connectors: To troubleshoot cloud connectors, see the section for your cloud connector, for example [Troubleshoot AWS Connector Issues](#).

Allowed set of commands

The allowed set of commands enables you to run some debug commands on the appliances and Docker containers (for connectors). Allowed commands include the ability to retrieve logs and current running configuration, test network connectivity, and capture packets matching a specified port.

Figure 75: Troubleshoot page on Secure Workload virtual appliance

The screenshot displays the Cisco Tetration Virtual Appliance interface. At the top, it shows 'Cisco Tetration VIRTUAL APPLIANCE' and a 'Monitoring' status. A notification banner indicates that the user does not have an active license and that the evaluation period will end on Sun Oct 20 2019 23:32:37 GMT+0000. Below this, the 'Tetration Data Ingest Appliance' is shown as 'ACTIVE'. A table provides details for this appliance:

Checked In	Registered	Created
Jul 25 2019 06:28:53 am (PDT)	Jul 24 2019 07:17:36 pm (PDT)	Jul 24 2019 07:13:00 pm (PDT)

The 'Connectors' section lists three connectors: NetFlow, NetScaler, and AnyConnect, all marked as active with green checkmarks. The 'Troubleshoot' section provides instructions on running commands and displays a list of issued commands:

Command	Timestamp	Status	Actions
Execute docker instance command	Jul 24 07:39:31 pm (PDT)	Ready	View, Copy
Execute docker command	Jul 24 07:39:10 pm (PDT)	Ready	View, Copy
Update the listening port on a connector	Jul 24 07:38:40 pm (PDT)	Ready	View, Copy
Test network connectivity	Jul 24 07:37:47 pm (PDT)	Ready	View, Copy
List a directory	Jul 24 07:37:22 pm (PDT)	Ready	View, Copy
Execute docker instance command	Jul 24 07:36:57 pm (PDT)	Ready	View, Copy
Execute docker instance command	Jul 24 07:36:45 pm (PDT)	Ready	View, Copy



Note Troubleshooting using the allowed set of commands is available on the appliances and connectors only for users with the *Customer Support* role.

Show Logs

Show the contents of a controller log file and optionally grep the file for a specified pattern. Secure Workload sends the command to appliance/connector where the command was issued. The controller on the appliance/connector service returns the result (tailed for the last 5000 lines). When the result is available at Secure Workload, a download button is presented to download the file.

Argument Name	Type	Description
Grep Pattern	string	Pattern string to grep from the logfile

Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 76: Download Show Logs output from Secure Workload Ingest appliance

The screenshot displays the Cisco Tetration n° Virtual Appliance interface. A modal dialog box titled 'Command Info & Results' is open, showing the command 'Show logs' issued on Jul 25 06:55:50 am (PDT). The dialog includes fields for 'Log File', 'Controller log', and 'Grep Pattern', and a 'Download Output File' button. The background interface shows a list of connectors (NetFlow, NetScaler, AnyConnect) and a table of issued commands, including 'Show logs', 'Execute docker instance command', 'Execute docker command', 'Update the listening port on a connector', 'Test network connectivity', 'List a directory', and another 'Execute docker instance command'.

Show Service Logs

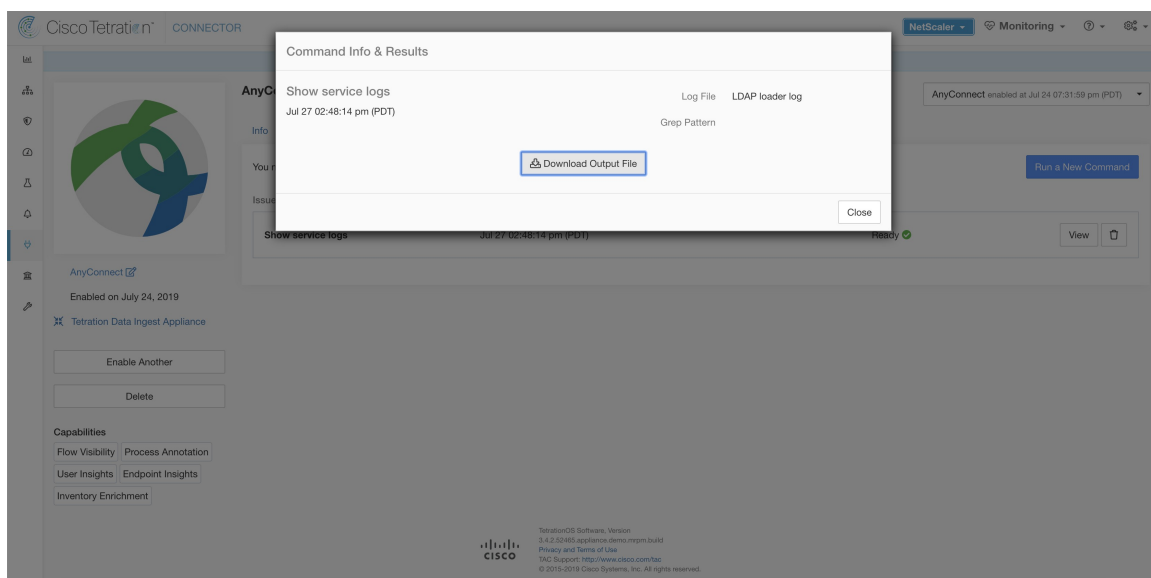
Show the contents of service log files and optionally grep the file for a specified pattern. Secure Workload sends the command to appliance/connector where the command was issued. The controller on the appliance/connector service returns the result (tailed for the last 5000 lines). When the result is available at Secure Workload, a download button is presented to download the file.

Argument Name	Type	Description
Log File	dropdown	The name of the logfile to collect
	• <i>Service log</i>	Logs of the connector service
	• <i>Upgrade log</i>	Upgrade logs of the service
	• <i>LDAP loader log</i>	Logs of the LDAP snapshot for connectors that have LDAP enabled
Grep Pattern	string	Pattern string to grep from the logfile

Allowed Secure Workload virtual appliances: None (only available on valid connector services)

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 77: Download Show Service Logs output from AnyConnect connector for LDAP loader log log file



Show Running Configuration

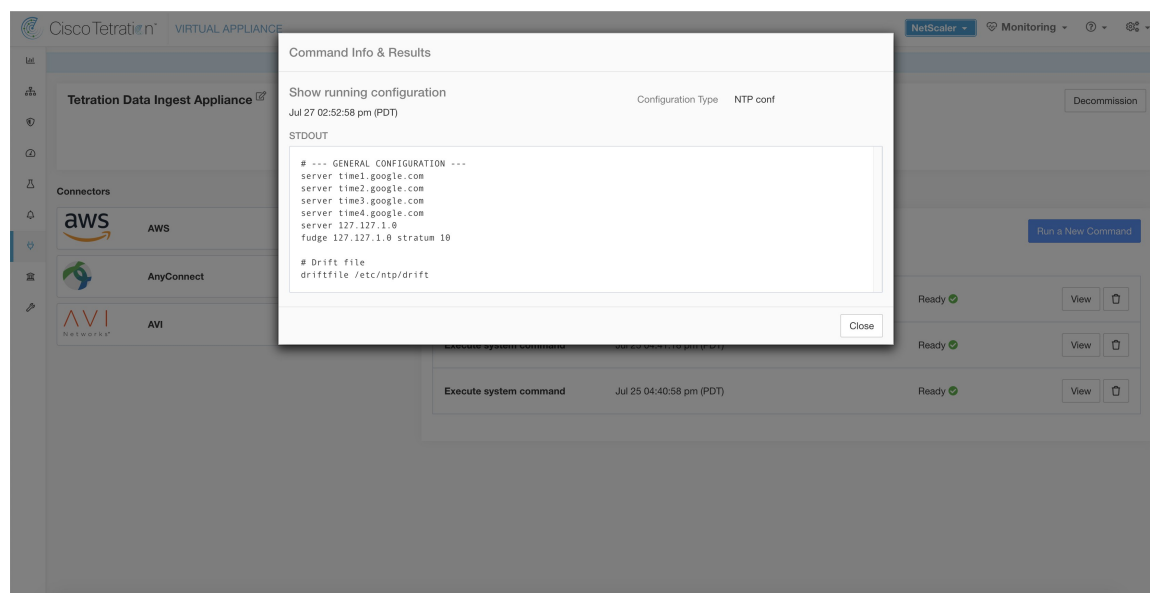
Show running configuration of an appliance/connector controllers. The controller on appliance/connector retrieves the configuration corresponding to the requested argument and returns the result. When the result is available at Secure Workload, the contents of the configuration are shown in a text box.

Argument Name	Type	Description
Configuration Type	dropdown	Configuration file to collect
	• <i>Controller conf</i>	Configuration file of the appliance controller
	• <i>Supervisor conf</i>	Configuration file of the supervisor that runs the controller
	• <i>NTP conf</i>	NTP configuration file
	• <i>Chrony conf</i>	/etc/chrony.conf

Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 78: Show running configuration for NTP conf on a Secure Workload Ingest Appliance



Show Service Running Configuration

Show running configuration of an services instantiated for connectors on the appliances. The controller on the service retrieves the configuration corresponding to the requested argument and returns the result. When the result is available at Secure Workload, the contents of the configuration are shown in a text box.

Argument Name	Type	Description
Configuration Type	dropdown	Configuration file to collect.
	• <i>Controller conf</i>	Configuration file of the service controller.
	• <i>Supervisor conf</i>	Configuration file of the supervisor that runs the controller.
	• <i>Service conf</i>	Service configuration file.
	• <i>LDAP conf</i>	LDAP configuration for connectors that have LDAP enabled.

Allowed Secure Workload virtual appliances: None (only available on valid connector services)

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Show System Commands

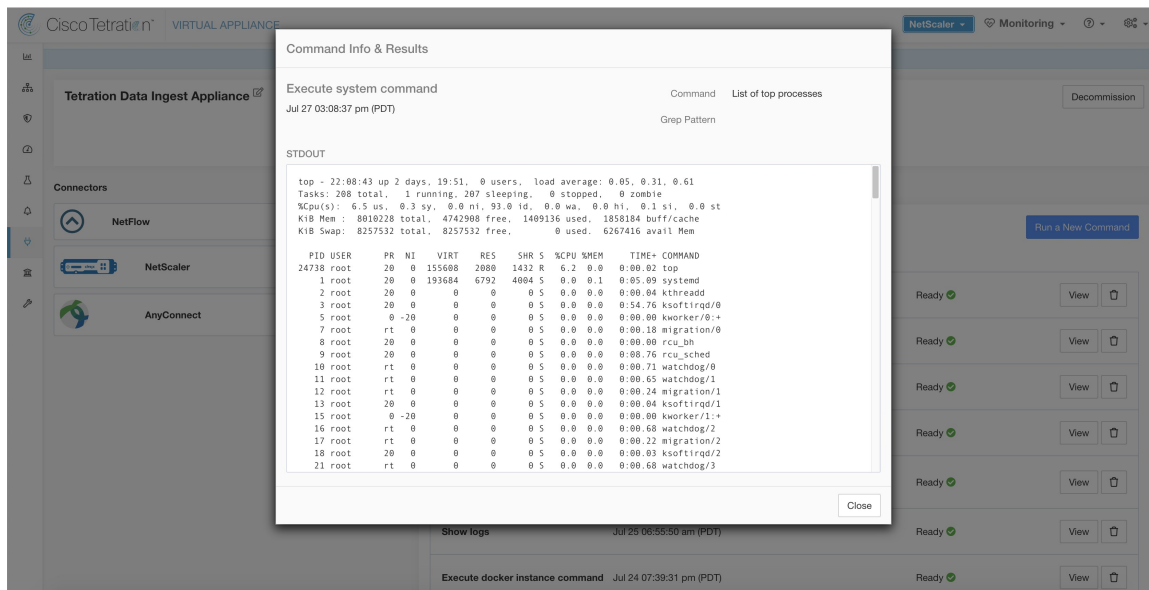
Execute a system command and optionally grep for a specified pattern. The controller on the appliance/connector service returns the result (tailed for the last 5000 lines). Optionally, a grep pattern can be provided as argument and the output is filtered accordingly. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
System Command	dropdown	System command to execute
	• <i>IP configuration</i>	ifconfig
	• <i>IP route configuration</i>	ip route
	• <i>IP packet filtering rules</i>	iptables -L
	• <i>Network status</i>	netstat
	• <i>Network status (EL9)</i>	ss
	• <i>Process status</i>	ps -aux
	• <i>List of top processes</i>	top -b -n 1
	• <i>NTP status</i>	ntpstat
	• <i>NTP query</i>	ntpq -pn
	• <i>Chrony status (EL9)</i>	chronyc tracking
	• <i>Chrony query (EL9)</i>	chronyc sources
	• <i>CPU info</i>	lscpu
	• <i>Memory info</i>	lsmem
• <i>Disk free</i>	df -H	
Grep Pattern	string	Pattern string to grep from the output

Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 79: Show system command on Secure Workload Ingest appliance to retrieve list of top processes



Show Docker Commands

Execute a Docker command and optionally grep for a specified pattern. The command is executed on the appliance by the appliance controller. The result tailed for the last 5000 lines. Optionally, a grep pattern can be provided as argument and the output is filtered accordingly. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
Docker Command	dropdown	Docker command to execute
	• <i>Docker info</i>	docker info
	• <i>List images</i>	docker images --no-trunc
	• <i>List containers</i>	docker ps --no-trunc
	• <i>List networks</i>	docker network ls --no-trunc
	• <i>List volumes</i>	docker volume ls
	• <i>Container stats</i>	docker stats --no-trunc--no-stream
	• <i>Docker disk usage</i>	docker system df -v
	• <i>Docker system events</i>	docker system events --since '10m'
	• <i>Version</i>	docker version

Argument Name	Type	Description
Grep Pattern	string	Pattern string to grep from the output

Allowed Secure Workload virtual appliances: All

Allowed connectors: None

Figure 80: Execute a docker command on Secure Workload Ingest appliance to show container stats

The screenshot shows the Cisco Tetration Virtual Appliance interface. A modal window titled 'Command Info & Results' is open, displaying the following information:

Execute docker command
 Jul 27 03:12:03 pm (PDT)

Command: Container stats
 Grep Pattern

STDOUT

CONTAINER ID	NAME	CPU %
68c5b9b95aefdc24be386a8eea3681b4cc87c19d624befd88ffba387ba019fb	nf-5d391177ff16335b4aa6bd1b	10.27%
adc259a446976aef5ee17ed0e5670446d76c0c562f34dcd59a98d2a1a3007381	ac-5d39149fff16335b4aa6bd1f	0.18%
613969a4066020791ac96220a1bd0f86d36e2f35f51653e1e8d5dc565c0340ea	ns-5d391341fff16335b4aa6bd1d	0.21%

The background interface shows a list of connectors (NetFlow, NetScaler, AnyConnect) and a command execution log with entries like 'Execute system command' and 'Execute docker instance command'.

Show Docker Instance Commands

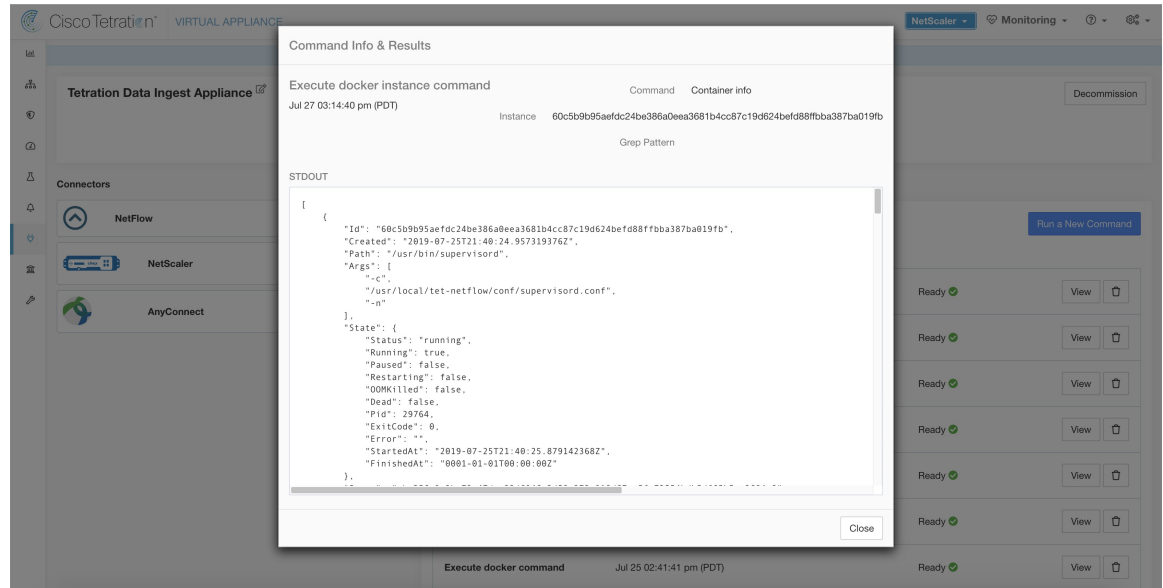
Execute a docker command on a specific instance of a Docker resource. The instance ID can be fetched using [Show Docker Commands](#). The command is executed on the appliance by the appliance controller. The result tailed for the last 5000 lines. Optionally, a grep pattern can be provided as argument and the output is filtered accordingly. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
Docker Command	dropdown	Docker command to execute
	• <i>Image info</i>	docker images --no-trunc <instance>
	• <i>Network info</i>	docker network inspect <instance>
	• <i>Volume info</i>	docker volume inspect <instance>
	• <i>Container info</i>	docker container inspect--size <instance>
	• <i>Container logs</i>	docker logs --tail 5000 <instance>
	• <i>Container port mappings</i>	docker port <instance>
	• <i>Container resource usage stats</i>	docker stats --no-trunc--no-stream <instance>
• <i>Container running processes</i>	docker top <instance>	
Instance	string	Docker resource (image, network, volume, container) ID (See Show Docker Commands)
Grep Pattern	string	Pattern string to grep from the output

Allowed Secure Workload virtual appliances: All

Allowed connectors: None

Figure 81: Execute a docker instance command on Secure Workload Ingest appliance to retrieve container info



Show Supervisor Commands

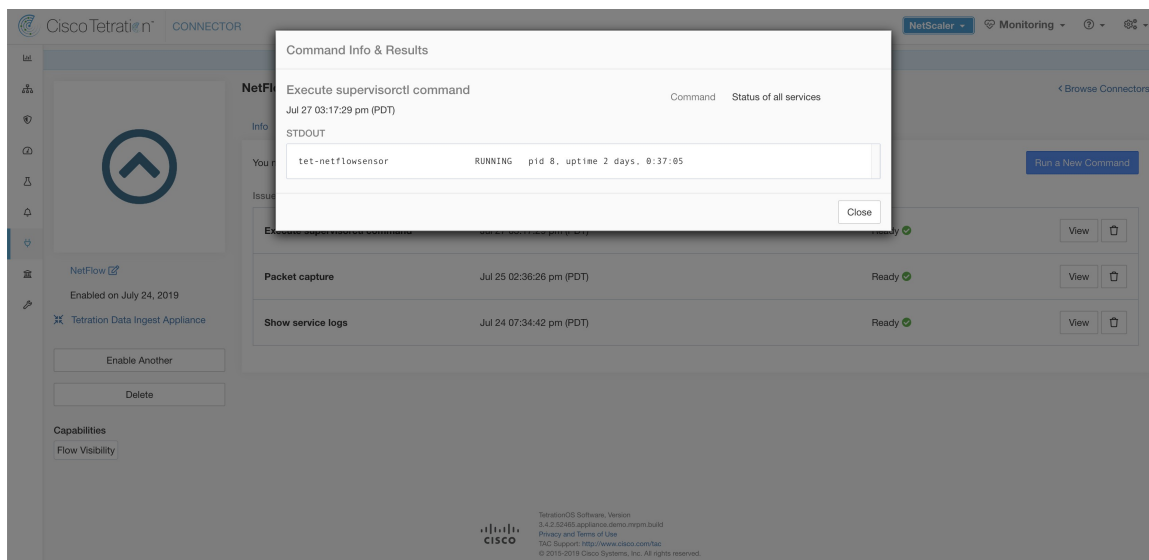
Execute a supervisorctl command and return the result. Secure Workload sends the command to appliance/connector where the command was issued. The controller on the appliance/connector service returns the result. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
SupervisorCtl Command	dropdown	<i>supervisorctl</i> command to execute
	• <i>Status of all services</i>	supervisorctl status
	• <i>PID of supervisor</i>	supervisorctl pid
	• <i>PID of all services</i>	supervisorctl pid all

Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 82: Execute supervisorctl command on NetFlow connector to get the status of all services

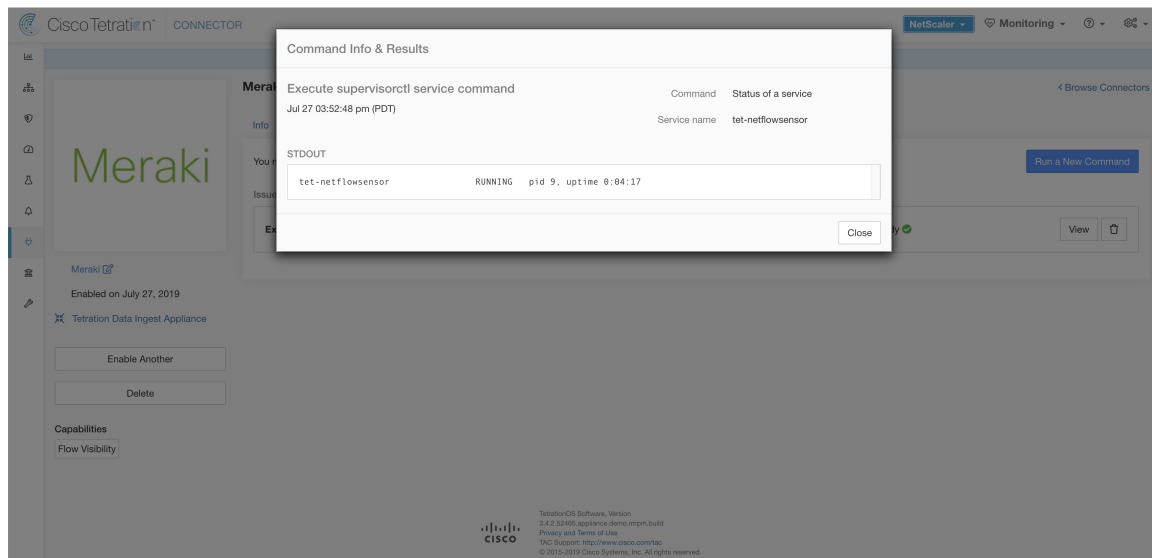


Show Supervisor Service Commands

Execute a supervisorctl command on a specific service. The service name can be fetched using [Show Supervisor Commands](#). Secure Workload sends the command to appliance/connector where the command was issued. The controller on the appliance/connector service returns the result. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
SupervisorCtl Command	dropdown	<i>supervisorctl</i> command to execute
	• <i>Status of a service</i>	supervisorctl status <service name>
	• <i>PID of a service</i>	supervisorctl pid <service name>
Service name	string	Name of the supervisor controlled service (see Show Supervisor Commands)

Figure 83: Execute supervisorctl command on NetFlow connector to get the status of specified service name



Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Network Connectivity Commands

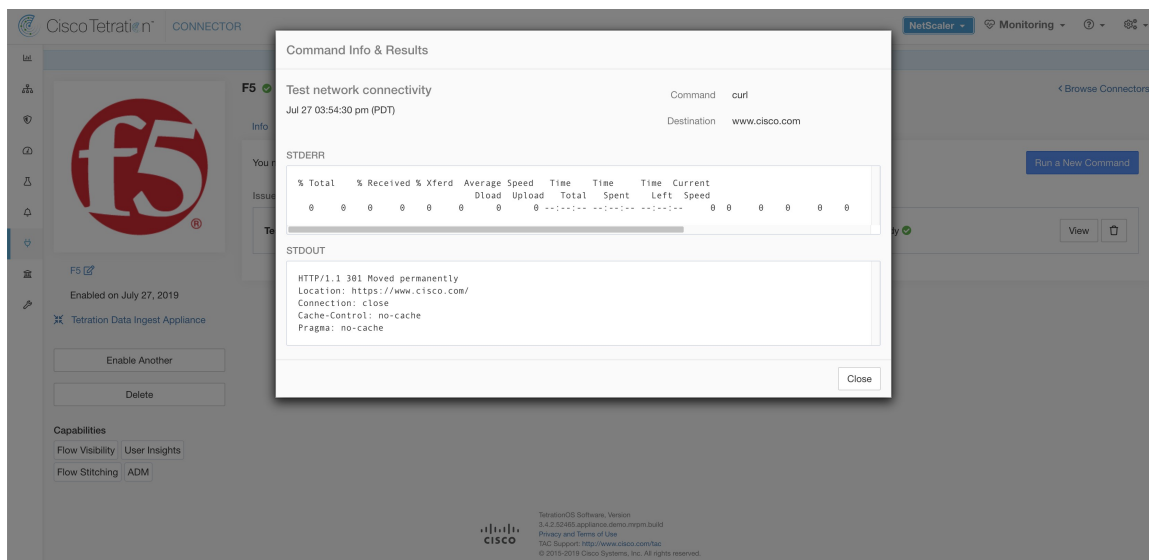
Test network connectivity from the appliance/connector. The command is executed on the appliance by the appliance controller. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
Network Command	dropdown	Network connectivity command to execute
	• <i>ping</i>	ping -c 5 <destination>
	• <i>curl</i>	curl -I <destination>
Destination	string	Destination to use for the test

Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 84: Test network connectivity on F5 connector by running a curl



List Files

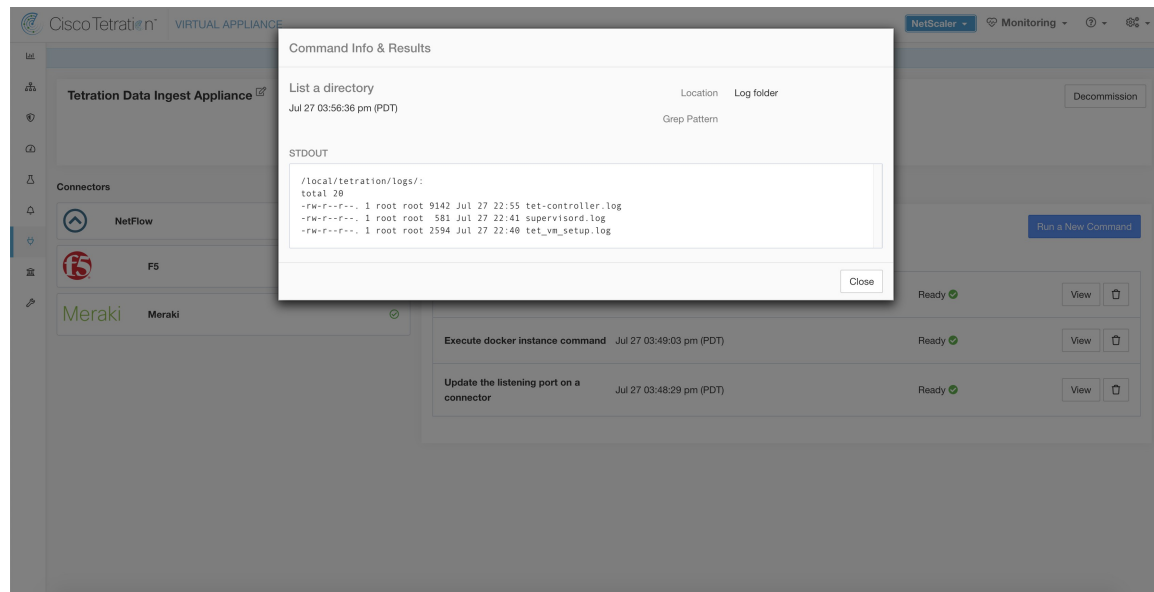
List the files in well known locations of the appliance. Optionally, grep for a specified pattern. Secure Workload sends the command to appliance where the command was issued. The controller on the appliance returns the result. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
Location	dropdown	List files in a target location
	<ul style="list-style-type: none"> • <i>Controller configuration folder</i> 	List the contents in the folder where controller configuration files are kept.
	<ul style="list-style-type: none"> • <i>Controller cert folder</i> 	List the contents in the folder where controller certs are kept.
	<ul style="list-style-type: none"> • <i>Log folder</i> 	List the contents in the folder where log files are present.
Grep Pattern	string	Pattern string to grep from the output

Allowed Secure Workload virtual appliances: All

Allowed connectors: None

Figure 85: List the files in log folder in Secure Workload Ingest appliance



List Service Files

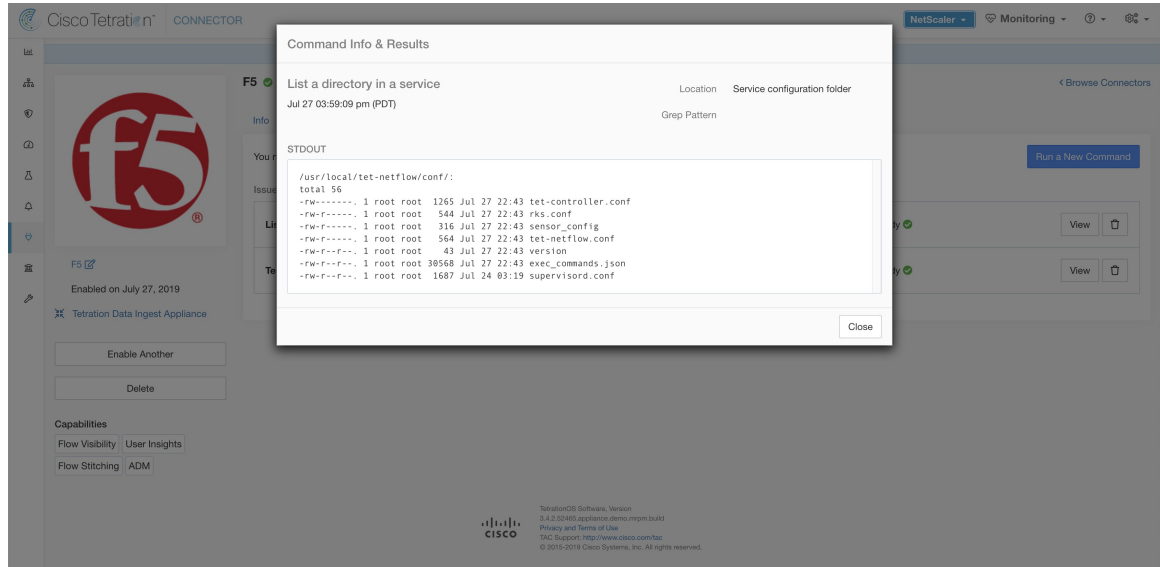
List the files in well known locations of the connector service. Optionally, grep for a specified pattern. Secure Workload sends the command to connector where the command was issued. The controller on the connector service returns the result. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
Location	dropdown	List files in a target location.
	• <i>Service configuration folder</i>	List the contents in the folder where service configuration files are kept.
	• <i>Service cert folder</i>	List the contents in the folder where service certs are kept.
	• <i>Log folder</i>	List the contents in the folder where log files are present.
• <i>DB folder</i>	List the contents in the folder where state of endpoints (esp. for AnyConnect and ISE connectors) are kept.	
Grep Pattern	string	Pattern string to grep from the output

Allowed Secure Workload virtual appliances: None

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 86: List the files in configuration folder of F5 connector in Secure Workload Ingest appliance



Packet Capture

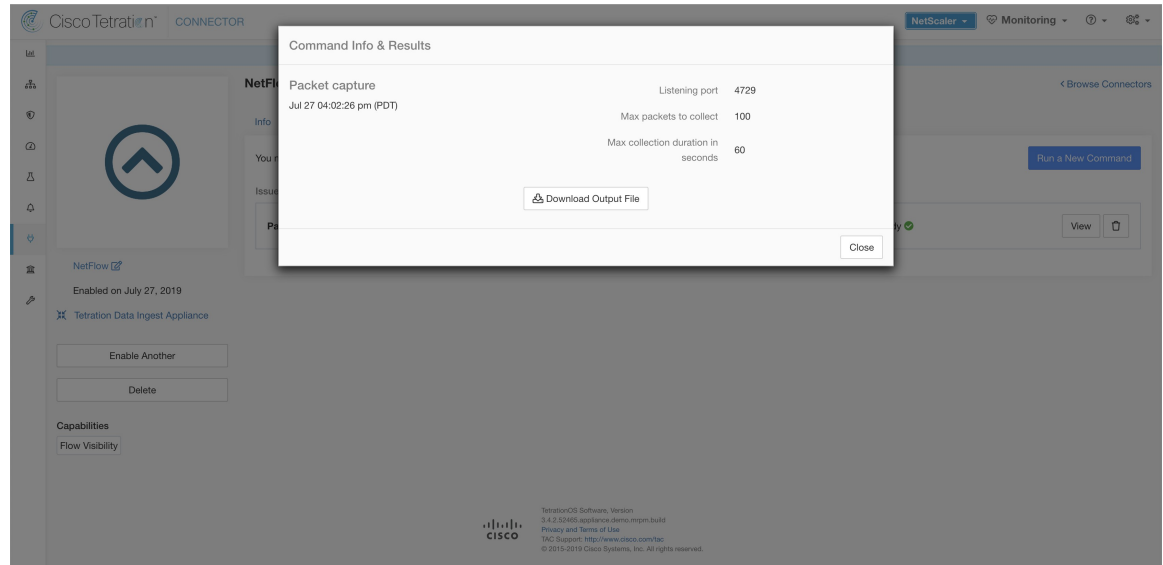
Capture incoming packets on an appliance/connector. Secure Workload sends the command to the appliance/connector where the command was issued. The controller on the appliance/connector service captures packets, encodes them and returns the result to Secure Workload. When the result is available at Secure Workload, a download button is presented to download the file in `.pcap` format.

Argument Name	Type	Description
Listening port	number	Capture packets that are sent/received on this port
Max packets to collect	number	Maximum packets to collect before returning the result. Should be <1000
Max collection duration in seconds	number	Maximum duration to collect before return the result. Should be <600 seconds.

Allowed Secure Workload virtual appliances: All

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, and Meraki.

Figure 87: Capture packets on a given port on NetFlow connector



Update Listening Ports of Connectors

Update the listening port on a connector in Secure Workload Ingest appliance. Secure Workload sends the command to the appliance controller on the appliance where the command is issued. The controller does the following actions:

- Stops the Docker service corresponding to the connector.
- Collect the current running configuration of the service.
- Remove the Docker service.
- Update the running configuration of the service to use the new ports.
- Start a new container from the same Docker image that was used in the removed container with new exposed ports. Also, if a Docker volume was mounted to the removed container earlier, the same volume is mounted to the new container.
- Return the new IP bindings of the connector to Secure Workload.
- Secure Workload shows the result in a text box.

Argument Name	Type	Description
Connector ID	string	Connector ID of the connector for which listening ports need to be updated
Listening port label	dropdown	The type of port that is updated.
	<i>NET-FLOW9</i>	NetFlow v9 listening port
	<i>IPFIX</i>	IPFIX listening port

Argument Name	Type	Description
Listening port	string	New port for the connector

Allowed Secure Workload virtual appliances: Secure Workload Ingest

Allowed connectors: None

Figure 88: Update listening port on Meraki connector to 2055 in Secure Workload Ingest appliance

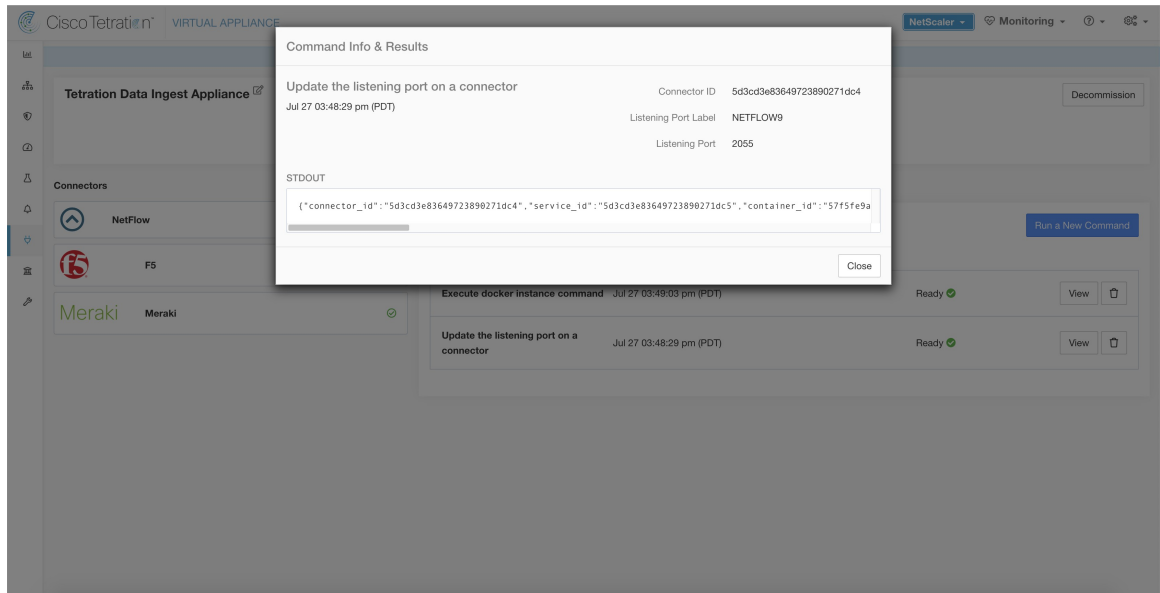
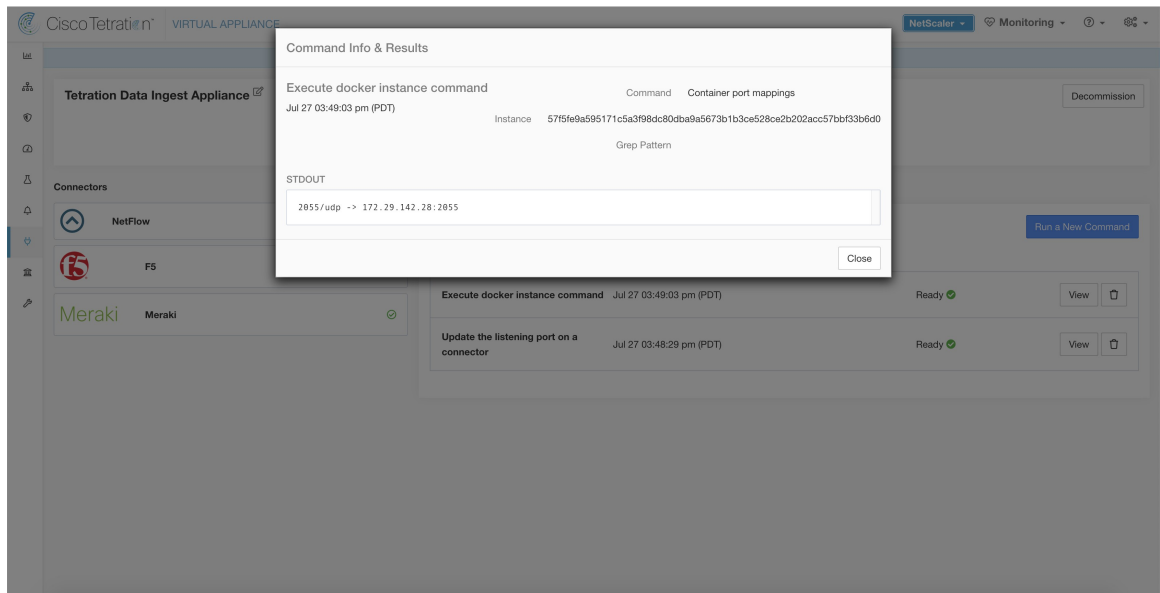


Figure 89: Retrieve the port mappings on Meraki connector in Secure Workload Ingest appliance



Update Alert Notifier Connector Log Configuration

Update log configuration for Secure Workload Alert Notifier (TAN) service that hosts Syslog, Email, Slack, PagerDuty, and Kinesis alert notifier connectors. Since TAN hosts multiple connectors, log configuration cannot be updated from connector page directly. This allowed command allows the user to update the log configuration.

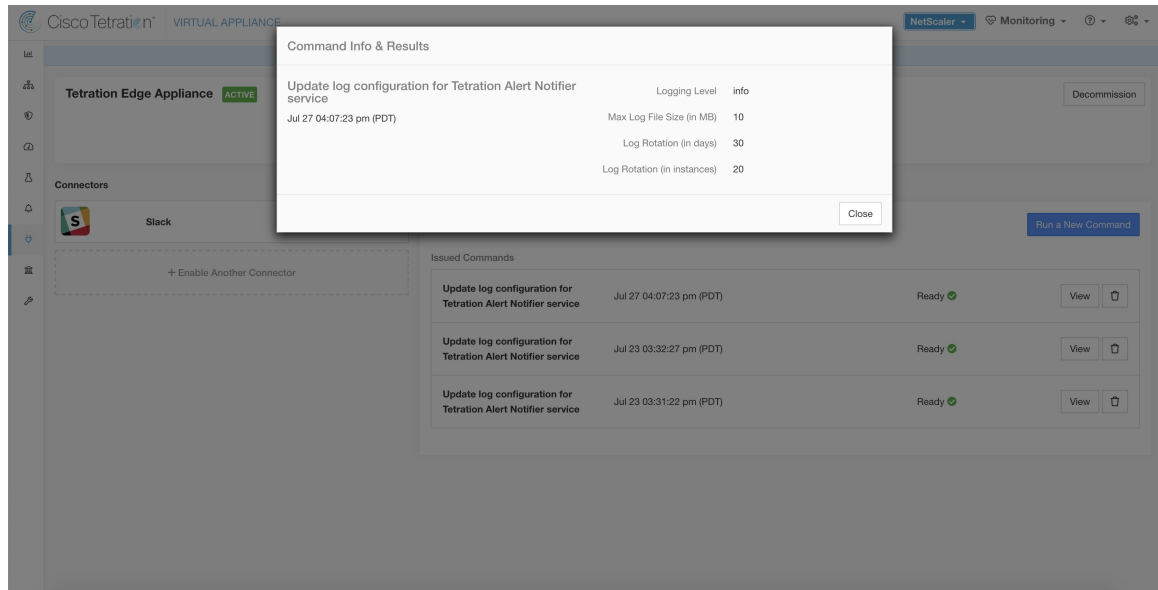
Secure Workload sends the command to the service controller on TAN Docker service of Secure Workload Edge appliance. The controller applies the configuration on the service and returns the status of the configuration update.

Argument Name	Type	Description
Logging level	dropdown	Logging level to be used by the service
	• <i>debug</i>	Debug log level
	• <i>info</i>	Informational log level
	• <i>warn</i>	Warning log level
	• <i>error</i>	Error log level
Max log file size (in MB)	number	Maximum size of a log file before log rotation kicks in
Log rotation (in days)	number	Maximum age of a log file before log rotation kicks in
Log rotation (in instances)	number	Maximum instances of log files kept

Allowed Secure Workload virtual appliances:Secure Workload Edge

Allowed connectors:None

Figure 90: Update the log configuration on Secure Workload Alert Notifier Docker service in Secure Workload Edge appliance



Collect Snapshot From Appliance

Secure Workload sends the command to the appliance where the command was issued. When the controller on the appliance receives this command from Secure Workload, it collects appliance snapshot, encodes them and returns the result to Secure Workload. When the result is available at Secure Workload, a download button is presented to download the file in `.tar.gz` format.

Files included in the snapshot:

- `/local/tetration/appliance/appliance.conf`
- `/local/tetration/{logs, sqlite, user.cfg}`
- `/opt/tetration/tet_vm_setup/conf/tet-vm-setup.conf`
- `/opt/tetration/tet_vm_setup/docker/Dockerfile`
- `/opt/tetration/ova/version`
- `/usr/local/tet-controller/conf`
- `/usr/local/tet-controller/cert/{topic.txt, kafkaBrokerIps.txt}`
- `/var/run/supervisord.pid`
- `/etc/resolv.conf`

Command outputs included in the snapshot:

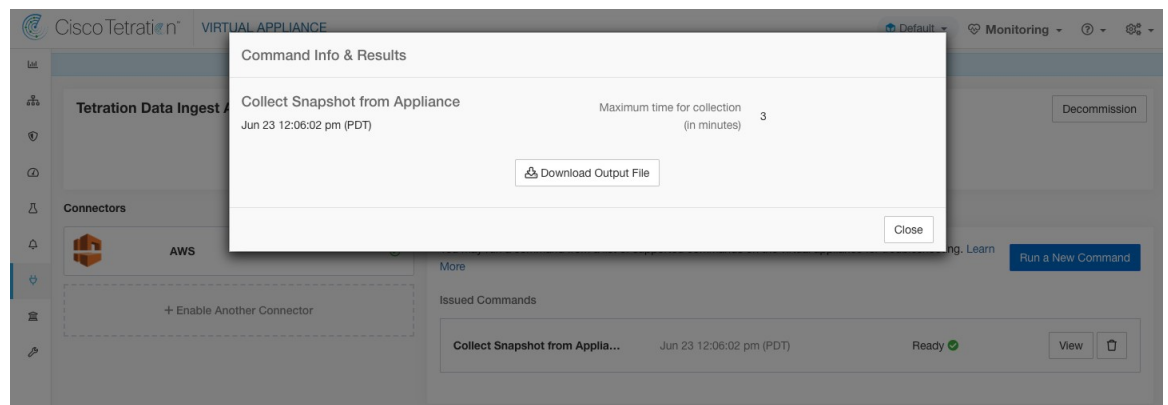
- `ps aux`
- `iptables -L`
- `netstat {-nat, -rn, -suna, -stna, -tunlp}`
- `ss {-nat, -rn, -suna, -stna, -tunlp}`

- /usr/local/tet-controller/tet-controller -version
- supervisorctl status
- rpm -qi tet-nic-driver tet-controller
- du -shc /local/tetration/logs
- ls {/usr/local/tet-controller/cert/, -l /local/tetration/sqlite/, -l /opt/tetration/tet_vm_setup/.tet_vm.done, -l /opt/tetration/tet_vm_setup/templates/}
- docker {images, ps -a}
- blkid/lsblk/lscpu/uptime
- free -m
- df -h

Argument Name	Type	Description
Max time for collection in minutes	number	Maximum duration to collect before returning the results. Should be <20 minutes.

Allowed Secure Workload virtual appliances : Secure Workload Ingest and Secure Workload Edge

Figure 91: Collect snapshot from Secure Workload appliance



Collect Snapshot From Connector

Secure Workload sends the command to the appliance where the connector is deployed. According to connector ID, the controller collects connector snapshot, encodes them and returns the result to Secure Workload. When the result is available at Secure Workload, a download button is presented to download the file in .tar.gz format.

Files included in the snapshot:

- /usr/local/tet-netflow/conf
- /local/tetration/{logs, sqlite}
- /var/run/{supervisord.pid, tet-netflow.pid}

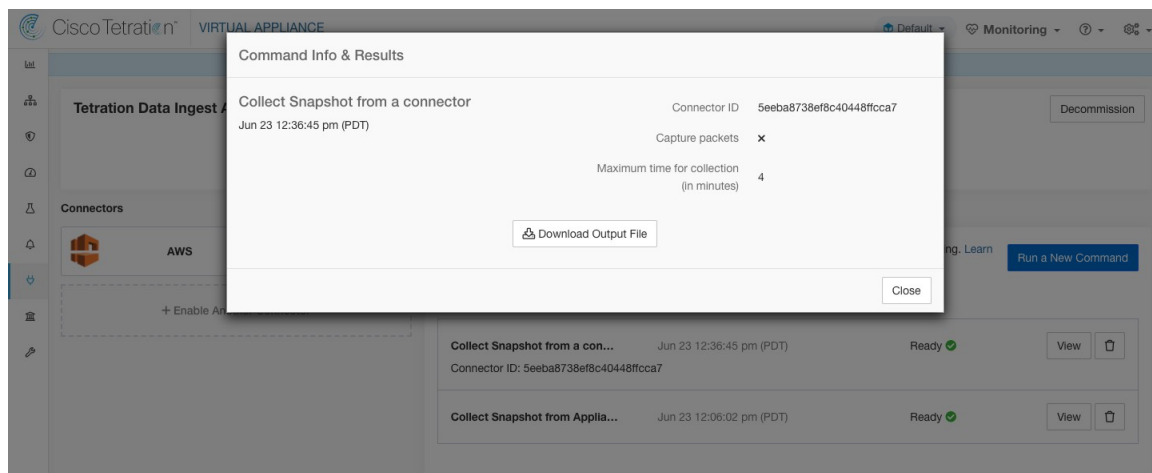
Command outputs included in the snapshot:

- ps aux
- netstat {-nat, -rn, -suna, -stna, -tunlp}
- ss {-nat, -rn, -suna, -stna, -tunlp}

Argument Name	Type	Description
Connector ID	string	Connector ID of the connector for which the snapshot command is run.
Capture packets	check-box	Should packets be captured?
Max time for collection in minutes	number	Maximum duration to collect before returning the results. Should be < 20 minutes.

Allowed Secure Workload virtual appliances: Secure Workload Ingest and Secure Workload Edge

Figure 92: Collect snapshot from Secure Workload connector on designated connector ID



Collect Controller Profile

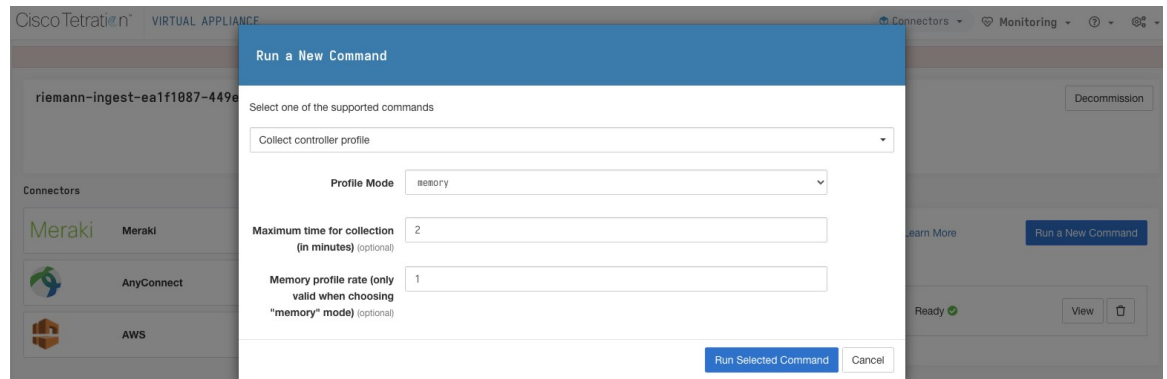
Collect controller process profiling result on appliance or connectors. Secure Workload sends the command to the connector where the command was issued. The service controller restarts the connector service in the specified profiling mode. After collecting the profiling result, service controller restarts the service in normal mode and send the result to Secure Workload. When the result is available at Secure Workload, a download button is presented to download the file in `.tar.gz` format.

Argument Name	Type	Description
Profile Mode	dropdown	Profiling mode.
	• <i>memory</i>	Memory profiling mode.
	• <i>cpu</i>	CPU profiling mode.
	• <i>block</i>	Block profiling mode.
	• <i>mutex</i>	Mutex profiling mode.
	• <i>goroutine</i>	Goroutine profiling mode.
Maximum time for collection (in minutes)	number	Maximum duration to collect before returning the result.
Memory profile rate (only valid when choosing “memory” mode)	number	Memory profiling rate. This field is optional. If not provided, default value in Golang will be used.

Allowed Secure Workload virtual appliances: Secure Workload Ingest and Secure Workload Edge

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, and Meraki.

Figure 93: Collect controller profile from Secure Workload appliance



Collect Connector Profile

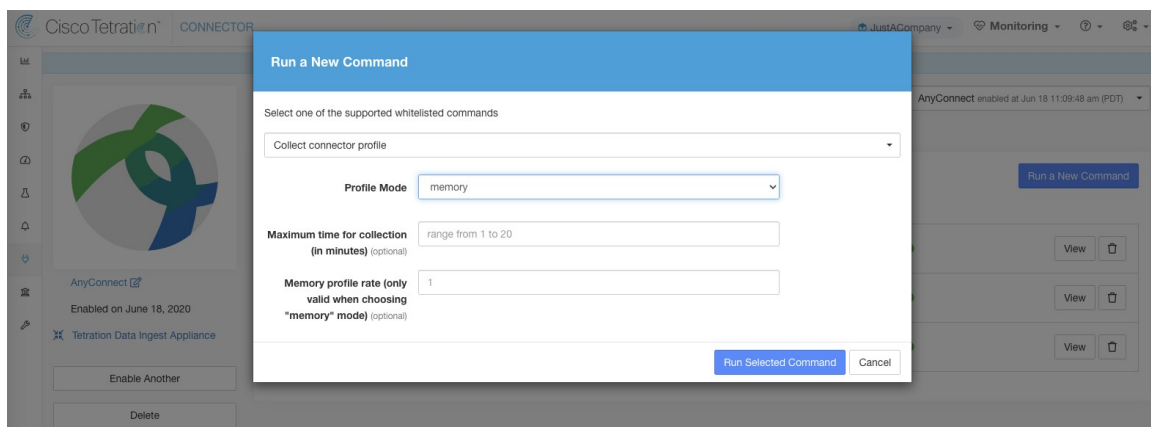
Collect connector process profiling result on connectors. Secure Workload sends the command to the connector where the command was issued. The service controller restart the connector service in the specified profiling mode. After collecting the profiling result, service controller restart the service in normal mode and send the result to Secure Workload. When the result is available at Secure Workload, a download button is presented to download the file in `.tar.gz` format.

Argument Name	Type	Description
Profile Mode	dropdown	Profiling mode.
	• <i>memory</i>	Memory profiling mode.
	• <i>cpu</i>	CPU profiling mode.
	• <i>block</i>	Block profiling mode.
	• <i>mutex</i>	Mutex profiling mode.
	• <i>goroutine</i>	Goroutine profiling mode.
Maximum time for collection (in minutes)	number	Maximum duration to collect before returning the result.
Memory profile rate (only valid when choosing “memory” mode)	number	Memory profiling rate. This field is optional. If not provided, default value in Golang will be used.

Allowed Secure Workload virtual appliances: Secure Workload Ingest and Secure Workload Edge

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, and Meraki.

Figure 94: Collect connector profile from Secure Workload connector



Override connector alert interval for Appliance

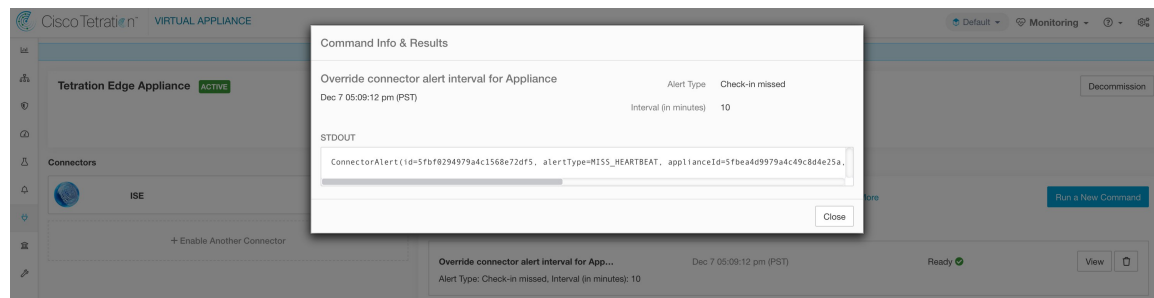
Override default connector alert interval for appliance. Secure Workload restricts same connector alert to send only once a day in default. This command is for administrator to override interval when they think once a day is too long. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
Alert Type	dropdown	The connector alert type to override.
	• <i>Check-in missed</i>	Miss appliance's check-in.
	• <i>CPU usage</i>	High CPU usage.
	• <i>Memory usage</i>	High memory usage.
	• <i>Disk usage</i>	High disk usage.
Interval (in minutes)	number	Duration to override interval in minutes.

Allowed Secure Workload virtual appliances: Secure Workload Ingest and Secure Workload Edge

Allowed connectors: None

Figure 95: Override connector alert interval for Secure Workload appliance



Override connector alert interval for Connector

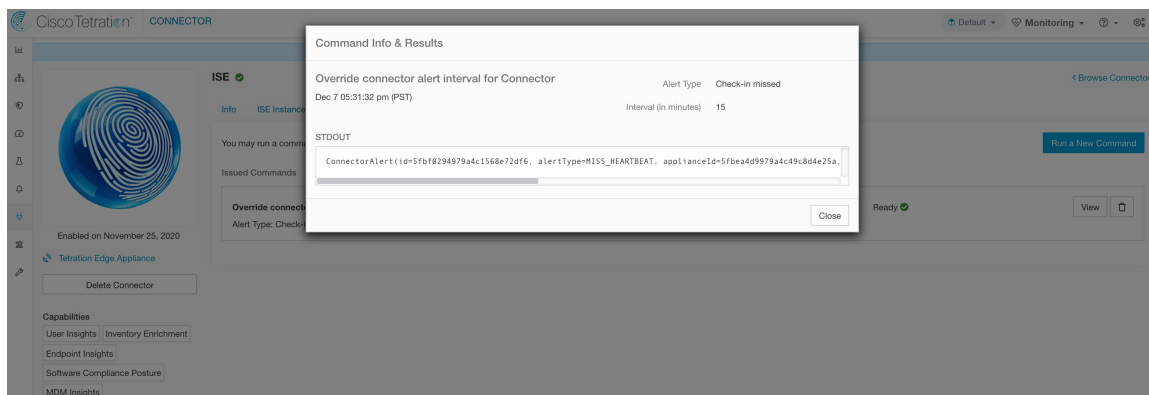
Override default connector alert interval for connector. Secure Workload restricts same connector alert to send only once a day in default. This command is for administrator to override interval when they think once a day is too long. When the result is available at Secure Workload, the result is shown in a text box.

Argument Name	Type	Description
Alert Type	dropdown	The connector alert type to override.
	• <i>Check-in missed</i>	Miss connector's check-in.
Interval (in minutes)	number	Duration to override interval in minutes.

Allowed Secure Workload virtual appliances: None

Allowed connectors: NetFlow, NetScaler, F5, AnyConnect, Syslog, Email, Slack, PagerDuty, Kinesis, ISE, ASA, Meraki, ServiceNow, WAD.

Figure 96: Override connector alert interval for Secure Workload connector



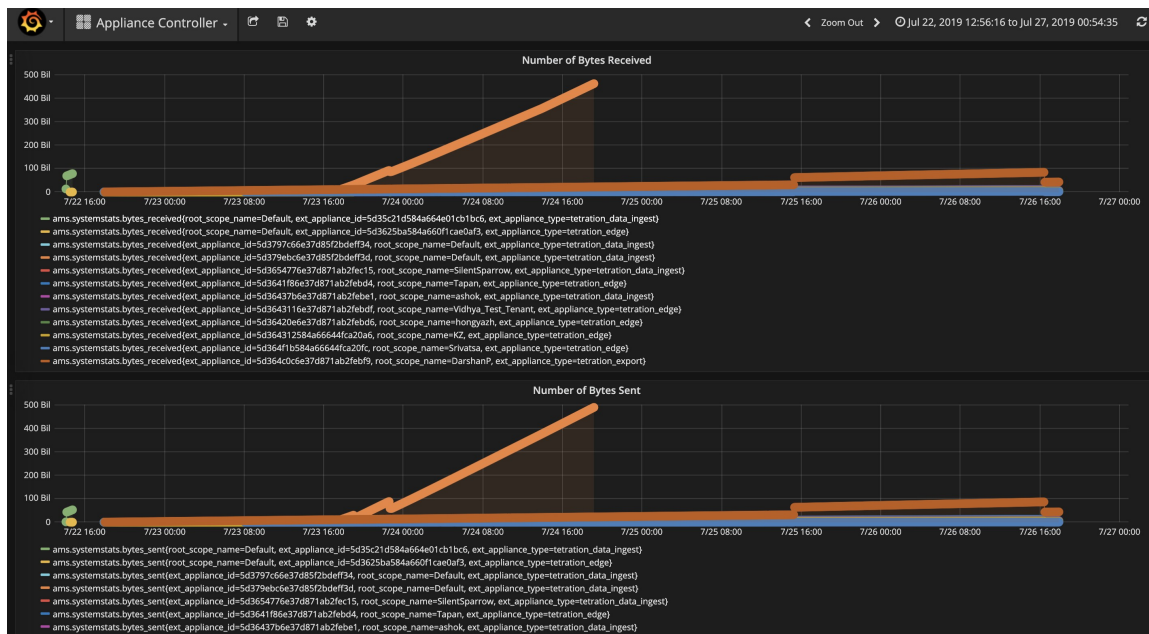
Hawkeye Dashboards

Hawkeye dashboards provide insights about health of the connectors and virtual appliances where the connectors are enabled.

Appliance Controller Dashboard

Appliance controller dashboard provides information about network statistics, system metrics such as CPU usage percentage, memory usage percentage, disk usage percentage, and number of open file descriptors.

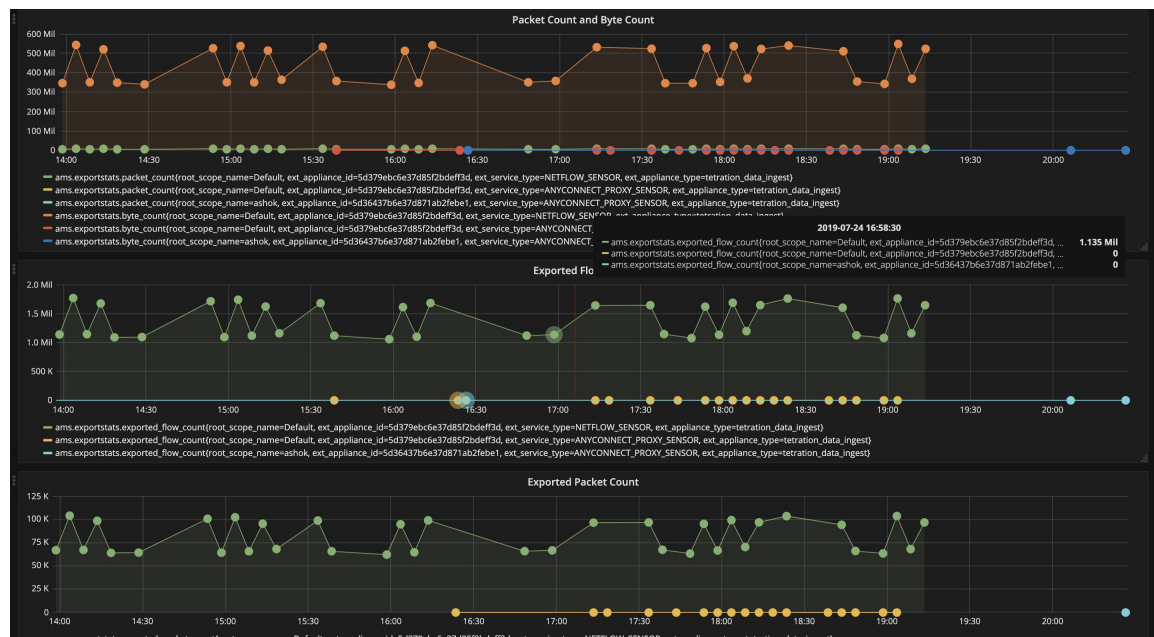
Figure 97: Appliance controller dashboard



Service Dashboard

Service dashboard provides information about export metrics -if applicable- including number of flow observations exported to Secure Workload, number of packets exported to Secure Workload, and number of bytes exported to Secure Workload. In addition, this dashboard also provides information about protocol processing and decoding (for example, services that process NetFlow v9, and IPFIX). Metrics such as decoded count, decoded error count, flow count, packet count, and byte count are available in this dashboard. Furthermore, system metrics for the Docker container where the service is running are also included in this dashboard. Metrics such as CPU usage percentage, memory usage percentage, disk usage percentage, and number of open file descriptors are part of this dashboard.

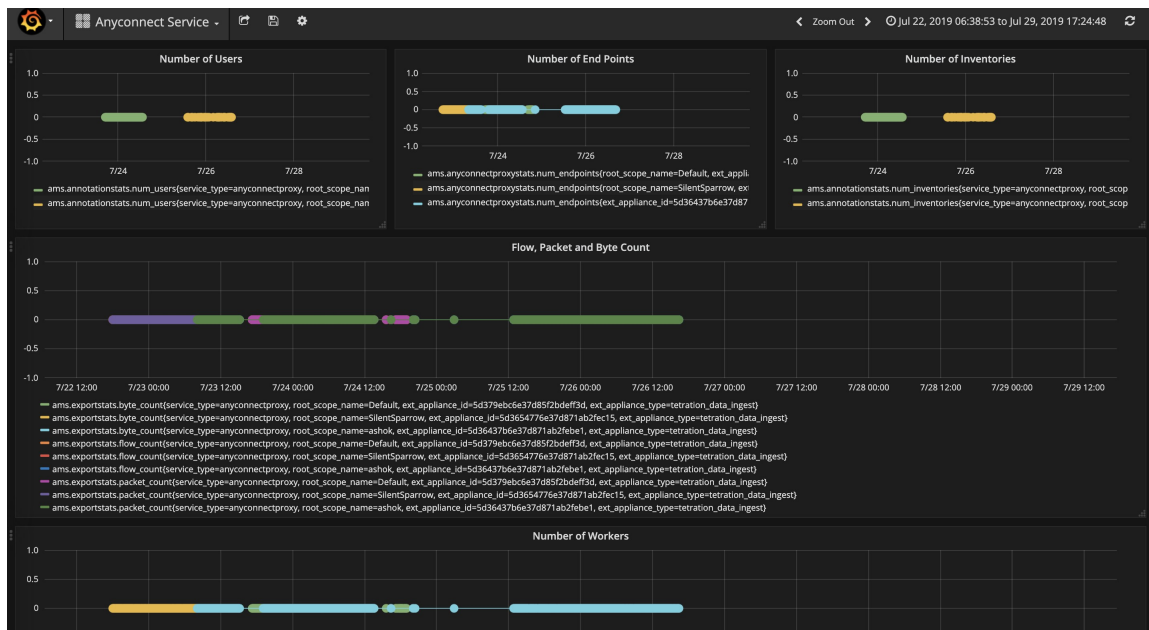
Figure 98: Service dashboard



AnyConnect Service Dashboard

AnyConnect service dashboard provides information about AnyConnect specific service information. Metrics such as number of endpoints, number of inventories, number of users reported by AnyConnect connector to Secure Workload are available in this dashboard. In addition, this dashboard also provides information about IPFIX protocol processing and decoding. Metrics such as decoded count, decoded error count, flow count, packet count, and byte count are available in this dashboard.

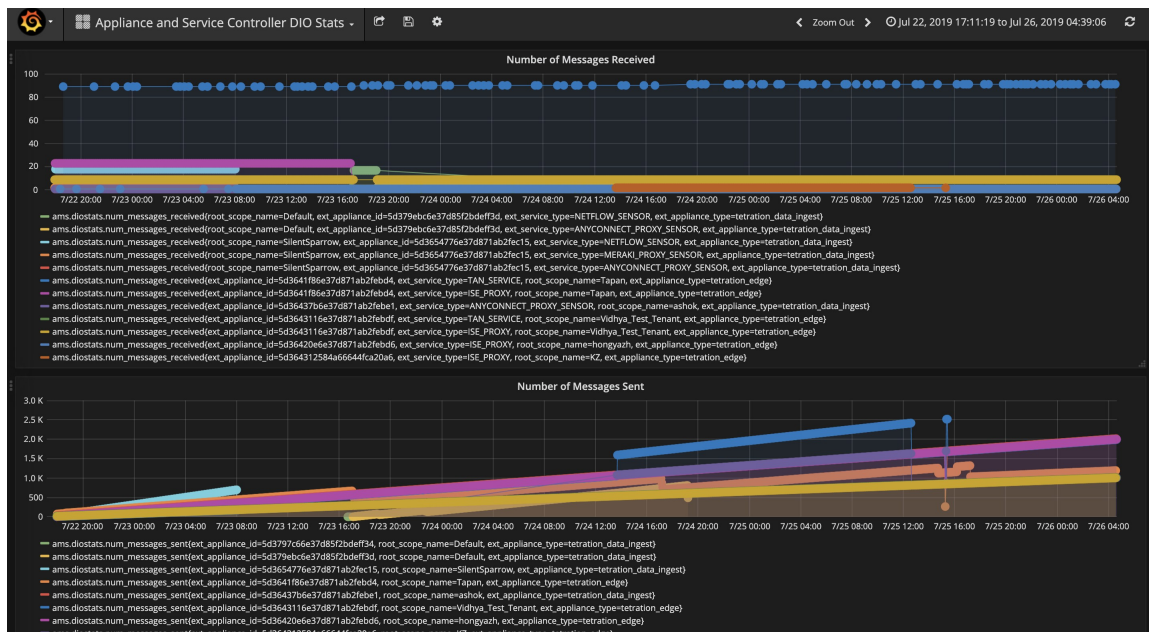
Figure 99: AnyConnect dashboard



Appliance and Service DIO Dashboard

Appliance and service DIO dashboard provides information about number of messages exchanged in the Kafka topic on which the appliance manager and appliance/service controllers communicate. Metrics such as number of messages received, number of messages sent, number of messages failed are included in this dashboard. In addition, the last offset read by the controllers are also provided to understand whether the controller is lagging behind in processing the control messages from the manager.

Figure 100: Appliance and service DIO dashboard



General Troubleshooting Guidelines

Once a connector shows in active state in connectors page in Secure Workload, no action is needed on the appliance where the connector is enabled; user does not need to log into it. If that is not happening, following information helps to troubleshoot such problems.

In normal conditions, on the appliance:

- `systemctl status tet_vm_setup.service` reports an *inactive* service with *SUCCESS* exit status.
- `systemctl status tet-nic-driver` reports an *active* service.
- `supervisorctl status tet-controller` reports *RUNNING* service. This indicates that the appliance controller is up and running.
- `docker network ls` reports three networks: `bridge`, `host`, and `none`.
- `docker ps` reports the containers that are running on the appliance. Typically, when a connector is enabled successfully on an appliance, a Docker container is instantiated on the appliance. For Syslog, Email, Slack, PagerDuty and Kinesis connectors, a Secure Workload alert notifier service is instantiated as a Docker container on Secure Workload edge appliance.
- `docker logs <cid>` for each container should report that `tet-netflowsensor` entered *RUNNING* state.
- `docker exec <cid> ifconfig` reports only one interface, besides the loopback.
- `docker exec <cid> netstat -rn` reports the default gateway.
- `cat /local/tetration/appliance/appliance.conf` on the appliance to see the list of Docker services running on the appliance. It includes details about service ID, connector ID, container, image ID and port mappings (if applicable). On a Secure Workload Ingest appliance, at most three services are running on the appliance. The port mappings and Docker volumes that are mounted on the containers are available in this file.

Figure 101: Secure Workload appliance deployment service and status

```
[root@esx-2106-ingest tetter]# systemctl status tet_vm_setup.service
• tet_vm_setup.service - Tetration Appliance Setup
  Loaded: loaded (/etc/systemd/system/tet_vm_setup.service; enabled; vendor preset: disabled)
  Active: inactive (dead) since Sat 2019-07-27 23:51:29 UTC; 21h ago
  Main PID: 1249 (code=exited, status=0/SUCCESS)

Jul 27 23:51:12 localhost.localdomain python[1249]: mount: /dev/sr0 is write-protected, mounting read-only
Jul 27 23:51:29 esx-2106-ingest python[1249]: Docker version 18.09.8, build 0dd43dd87f
Jul 27 23:51:29 esx-2106-ingest python[1249]: REPOSITORY          TAG          IMAGE ID          CREATE...  SIZE
Jul 27 23:51:29 esx-2106-ingest python[1249]: userPrivateKey.key
Jul 27 23:51:29 esx-2106-ingest python[1249]: intermediateCA.cert
Jul 27 23:51:29 esx-2106-ingest python[1249]: kafkaBrokerIps.txt
Jul 27 23:51:29 esx-2106-ingest python[1249]: userCA.cert
Jul 27 23:51:29 esx-2106-ingest python[1249]: kafkaCA.cert
Jul 27 23:51:29 esx-2106-ingest python[1249]: topic.txt
Jul 27 23:51:29 esx-2106-ingest python[1249]: Created symlink from /etc/systemd/system/multi-user.target.wants/s...vice.
Hint: Some lines were ellipsized, use -l to show in full.
[root@esx-2106-ingest tetter]#
```

Figure 102: Secure Workload network driver service status

```
[root@esx-2106-ingest tetter]# systemctl status tet-nic-driver.service
● tet-nic-driver.service - NIC network driver plugin for Docker
   Loaded: loaded (/etc/systemd/system/tet-nic-driver.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2019-07-27 23:51:12 UTC; 21h ago
   Main PID: 733 (nic)
   Memory: 4.4M
   CGroup: /system.slice/tet-nic-driver.service
           └─733 /usr/local/tet/nic-driver/nic -log-level debug

Jul 27 23:51:12 localhost.localdomain systemd[1]: Started NIC network driver plugin for Docker.
Jul 27 23:51:12 localhost.localdomain systemd[1]: Starting NIC network driver plugin for Docker...
Jul 27 23:51:12 localhost.localdomain nic[733]: time="2019-07-27T23:51:12Z" level=info msg="NIC network driver started"
Hint: Some lines were ellipsized, use -l to show in full.
[root@esx-2106-ingest tetter]#
```

Figure 103: Appliance controller status

```
[root@esx-2106-ingest tetter]# supervisorctl status tet-controller
tet-controller          RUNNING pid 1971, uptime 21:43:29
[root@esx-2106-ingest tetter]#
```

If any of the preceding doesn't hold true, check the deployment script logs in `/local/tetration/logs` for the reason why the appliance and/or the connector deployment failed.

You can troubleshoot any other connector registration/connectivity issues as follows.

```
docker exec <cid> ps -ef reports tet-netflowsensor-engine, /usr/local/tet/ tet-netflowsensor
-config /usr/local/tet-netflow/conf/tet-netflow.conf instances, along with the process manager
/usr/bin/supervisord -c /usr/local/tet-netflow/conf/supervisord.conf -n instance.
```

Figure 104: Running processes on Secure Firewall ASA connector in Secure Workload Ingest appliance

```
[root@esx-2106-ingest tetter]# docker ps
CONTAINER ID        IMAGE                                     PORTS                NAMES
c82decfaa877       asa_sensor-3.4.2.52465.appliance.demo.mrpm.build-asa:5d3ce5e43649723890271dd3  "/usr/bin/supervis
... " 22 hours ago    Up 22 hours         172.29.142.27:4729->4729/udp  asa-5d3ce5e43649723890271dd3
eddd5cd59839       aws_sensor-3.4.2.52465.appliance.demo.mrpm.build-aws:5d3ce3b73649723890271dce  "/usr/bin/supervis
... " 22 hours ago    Up 22 hours         aws-5d3ce3b73649723890271dce
[root@esx-2106-ingest tetter]# docker exec c8 ps -ef
UID                PID PPID C STIME TTY          TIME CMD
root                1  0  0 00:01 ?            00:00:15 /usr/bin/python /usr/bin/supervisord -c /usr/local/tet-netflow/conf/supe
rvisord.conf -n
root                8  1  0 00:01 ?            00:02:24 /usr/local/tet-netflow/tet-netflowsensor-engine -ctrl-config /usr/local/
tet-netflow/conf/tet-controller.conf -upgrade-script /usr/local/tet-netflow/scripts/check_config_update.sh -service /usr
/local/tet-netflow/tet-netflowsensor -config /usr/local/tet-netflow/conf/tet-netflow.conf
root               27002  8  0 21:31 ?            00:00:00 /usr/local/tet-netflow/tet-netflowsensor -config /usr/local/tet-netflow/
conf/tet-netflow.conf
root               27024  0  0 21:32 ?            00:00:00 ps -ef
[root@esx-2106-ingest tetter]#
```

Log Files

The following commands can be used to view the logs from various services on the appliance.

- `/local/tetration/logs/tet-controller.log` shows the logs of the appliance controller.
- `docker exec <cid> cat /local/tetration/logs/tet-controller.log` shows the logs of the service controller on the connector.
- `docker exec <cid> cat /local/tetration/logs/tet-netflow.log` shows the logs of the connector service.
- `docker exec <cid> cat /local/tetration/logs/tet-ldap-loader.log` shows the logs of LDAP snapshot creation (if LDAP config is applicable for the connector).

- `docker exec <cid> cat /local/tetration/logs/check_conf_update.log` shows the configuration update polling logs (for connectors on the Ingest appliance).



Note There are allowed set of commands on Secure Workload that can pull these logs from the appliance and/or connectors directly. For more information, see [Allowed set of commands](#).

Debug Mode

The default logging level for the appliance/service controller and connector service is set to *info* level. For troubleshooting issues, we may need to set the agent in *debug* mode. To do this, update the log configuration on the appliance/connector on Secure Workload directly for the desired appliance/connector. The log levels for both the controller and services are updated if the configuration is updated on the connector. For more information, see [Log Configuration](#).

Connector Alerts

An appliance/service creates a connector alert when it experiences abnormal behavior.

Alert Configuration

The alert configuration for appliances and connectors enables you to generate alerts for various events. In the 3.4 release, this configuration enables all types of alerts that are potentially possible for the configured appliance/connector.

Parameter Name	Type	Description
Enable Alert	checkbox	Should alert be enabled?



Note The default value for *Enable Alert* is *true*.

Figure 105: Show alert configuration on a Secure Workload Data Ingest Appliance

The screenshot shows the Cisco Tetration Virtual Appliance interface. At the top, it displays 'Cisco Tetration VIRTUAL APPLIANCE' and 'Monitoring'. A notification banner states: 'You do not have an active license. The evaluation period will end on Thu Sep 10 2020 01:09:39 GMT+0000. Please notify admin.' Below this, the 'Tetration Data Ingest Appliance' is shown as 'ACTIVE'. A table lists the appliance's status: 'Checked In' (Jun 11 2020 10:55:42 pm (PDT)), 'Registered' (Jun 11 2020 10:45:40 pm (PDT)), and 'Created' (Jun 11 2020 10:38:07 pm (PDT)). The 'Connectors' section is expanded to show 'AnyConnect' with a green checkmark. The 'Alert' tab is selected, showing the 'Enable Alert' checkbox checked. There are buttons for 'Cancel Config Creation' and 'Verify & Save Configs'.

Alert Type

The Info Tab on the appliance and connector pages contains various alert types specific to each appliance and connector.

Figure 106: Alert list info

The screenshot shows the Cisco Tetration interface for the AnyConnect connector. The main content area is titled 'AnyConnect' and has several tabs: 'Info', 'IP bindings', 'Endpoint', 'Workload AD', 'Log', 'Alert', and 'Troubleshoot'. The 'Info' tab is active, displaying the following text:

Collect telemetry data from Cisco AnyConnect Network Visibility Module (NVM). AnyConnect NVM provides visibility and monitoring of endpoint and user behavior both on and off premises. It sends host, interface, and flow records in IPFIX format to a collector (e.g., AnyConnect connector). AnyConnect connector registers each AnyConnect endpoint as an agent within Tetration and provide insight of the endpoint network behavior.

Alerts:

1. AnyConnect is down
2. CPU/Memory usage is too high
3. Can not connect to LDAP server

The page also shows a license notice at the top: 'You do not have an active license. The evaluation period will end on Thu Sep 10 2020 01:09:39 GMT+0000. Please notify admin.'

Appliance/Connector down

An alert generates when an appliance (or a connector) is potentially down due to missing heartbeats from the appliance/connector.

Alert text: Missing <Appliance/Connector> heartbeats, it might be down.

Severity: High

Figure 107: Alert for connector down

The screenshot shows the Cisco Tetration interface for 'CURRENT ALERTS'. The main content area has a 'Filters' section with 'Status = ACTIVE' selected. Below the filters is a table of alerts:

Event Time	Status	Alert Text	Severity	Type	Actions
11:25 PM	ACTIVE	Missing AnyConnect heartbeats, it might be down	HIGH	CONNECTOR	z/z

Below the table is a 'Details' section with the following information:

- Appliance ID: 5ee314bf1bf0541577c6349e
- Appliance Ip: 172.29.142.63
- Deep Link: marge.tetrationanalytics.com/#/connectors/details/ANYCONNECT?id=5ee316f05411a65ca2d8f2fd
- Last Checkin At: Jun 12 2020 06.10.51 AM UTC
- Name: ANYCONNECT
- Type: ANYCONNECT

Allowed Secure Workload virtual appliances: Secure Workload Ingest and Secure Workload Edge

Allowed connectors: All

Appliance/Connector system usage

When system usage (CPU, memory, and disk) is more than 90% on an appliance (and a connector). The appliance (and/or connector) generates an informational alert to indicate that it's currently handling an increased system load.

It's normal for appliances and connectors to consume more than 90% of system resources during heavy processing activity.

Alert text: <Number> of CPU/Memory/Disk usage on <Appliance/Connector> is too high.

Severity: High

Figure 108: Alert for connector system usage too high

The screenshot shows the Cisco Tetration Alerts interface. At the top, there's a navigation bar with 'Cisco Tetration' and 'CURRENT ALERTS'. Below that, a message states: 'You do not have an active license. The evaluation period will end on Thu Sep 10 2020 01:09:39 GMT+0000. Please notify admin.' The main area is titled 'Alerts Configuration' and includes a filter for 'Status = ACTIVE'. A table lists alerts with columns for Event Time, Status, Alert Text, Severity, Type, and Actions. One alert is shown: '12:51 AM', 'ACTIVE', '5.55% of MEMORY usage on AnyConnect is too high', 'HIGH', 'CONNECTOR'. Below the table is a 'Details' section with the following information:

- Appliance ID: 5ee314bf1bf0541577c6349e
- Appliance Ip: 172.29.142.63
- Deep Link: marge.tetrationanalytics.com/#/connectors/details/ANYCONNECT?id=5ee316f05411a65ca2d8f2fd
- Last Checkin At: Jun 12 2020 07:51:27 AM UTC
- Name: ANYCONNECT
- Type: ANYCONNECT

Allowed Secure Workload virtual appliances: Secure Workload Ingest and Secure Workload Edge

Allowed connectors: All

Connector config error

When a configuration for a connector can't connect to the configured server, the system generates the alert to indicate a potential issue with the configuration after accepting and deploying it.

For example, The AnyConnect connector can take an LDAP configuration, validate and accept the configuration. However, during the normal operation, it's possible that the configuration is no longer valid.

Alert captures the scenario and indicates that you have to take corrective action to update the configuration.

Alert text: Cannot connect to <Appliance/Connector> server, check <Appliance/Connector> config.

Severity: High, Low

Server	Connector
LDAP server	AnyConnect, F5, ISE, WDC
ISE server	ISE
ServiceNow server	ServiceNow

Figure 109: Alert for config status error

The screenshot shows the Cisco Tetration Alerts interface. At the top, there's a navigation bar with 'Cisco Tetration' and 'CURRENT ALERTS'. A notification banner states: 'You do not have an active license. The evaluation period will end on Thu Sep 10 2020 01:09:39 GMT+0000. Please notify admin.' Below this, the 'Alerts' section is active, with a filter for 'Status = ACTIVE'. A table lists alerts, with one entry at 11:00 PM, ACTIVE status, and the text 'Can't connect to LDAP server, please check LDAP config'. The severity is HIGH and the type is CONNECTOR. A 'Details' panel is open for this alert, showing the following information:

- Appliance ID: 5ee314bf1bf0541577c6349e
- Appliance Ip: 172.29.142.63
- Deep Link: marge.tetrationanalytics.com/#/connectors/details/ANYCONNECT?id=5ee316f05411a65ca2d8f2fd
- Last Checkin At: Jun 12 2020 06:00:51 AM UTC
- Name: ANYCONNECT
- Reason: Invalid Credentials Original Error Text LDAP Result Code 49 Invalid Credentials 80090308 Ldap Err DSID 0 C 090446 Comment Accept Security Context Error Data 52 E V 2580
- Type: ANYCONNECT

Allowed Secure Workload virtual appliances: Secure Workload Ingest and Secure Workload Edge.

Allowed connectors: AnyConnect, F5, ISE, WDC, and ServiceNow.

Connector UI Alert Details

Figure 110: Connector UI Alert details

The detailed view of the alert shows the following information:

- Appliance ID: 5ee314bf1bf0541577c6349e
- Appliance Ip: 172.29.142.63
- Deep Link: marge.tetrationanalytics.com/#/connectors/details/ANYCONNECT?id=5ee316f05411a65ca2d8f2fd
- Last Checkin At: Jun 12 2020 06:56:28 AM UTC
- Name: ANYCONNECT
- Reason: Invalid Credentials Original Error Text LDAP Result Code 49 Invalid Credentials 80090308 Ldap Err DSID 0 C 090446 Comment Accept Security Context Error Data 52 E V 2580
- Type: ANYCONNECT

Alert Details

See [Common Alert Structure](#) for general alert structure and information about fields. The `alert_details` fields structure contains the following subfields for connector alerts.

Field	Type	Description
Appliance ID	String	Appliance ID
Appliance IP	String	Appliance IP
Connector ID	String	Connector ID

Field	Type	Description
Connector IP	String	Connector IP
Deep Link	Hyperlink	Redirect to appliance/connector page
Last CheckIn At	String	Last checkin time
Name	String	Appliance/Connector name
Reason	String	The reason that Appliance/Connector can't connect to Secure Workload
Type	String	Appliance/Connector type

Example of Alert Details

After parsing alert_details as JSON (unstringified), it will display as follows.

```
{
  "Appliance ID": "5f1f3d26d674b01832c6792a",
  "Connector ID": "5f1f3e47baba512a70abee43",
  "Connector IP": "172.29.142.22",
  "Deep Link":
"bingo.tetrationanalytics.com/#/connectors/details/F5?id=5f1f3e47baba512a70abee43",
  "Last checkin at": "Aug 04 2020 20.37.33 PM UTC",
  "Name": "F5",
  "Reason": "Invalid Credentials (Original error text: LDAP Result Code 49 \"Invalid
Credentials\": )",
  "Type": "F5"
}
```

