



Cloud Analytics Engine Feature Guide for the QFX Series

Release

15.1X53



Modified: 2016-10-04

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Cloud Analytics Engine Feature Guide for the QFX Series

15.1X53

Copyright © 2016, Juniper Networks, Inc.

All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	ix
	Documentation and Release Notes	ix
	Supported Platforms	ix
	Using the Examples in This Manual	ix
	Merging a Full Example	x
	Merging a Snippet	x
	Documentation Conventions	xi
	Documentation Feedback	xiii
	Requesting Technical Support	xiii
	Self-Help Online Tools and Resources	xiii
	Opening a Case with JTAC	xiv
Part 1	Cloud Analytics Engine	
Chapter 1	Using the Cloud Analytics Engine	3
	Cloud Analytics Engine Overview	3
	Installing Junos for Cloud Analytics Engine	4
	Understanding Cloud Analytics Engine Architecture and Deployment	4
	Supported Topologies	4
	Software Components	5
	Support for Devices that Do Not Support Cloud Analytics Engine	6
	User Interfaces	6
	Basic Workflow	6
	Installing and Configuring Cloud Analytics Engine Compute Agent	7
	Configuring Compute Agent by Running the Interactive Setup Program	8
	Configuring Compute Agent Initial Configuration by Using a Configuration File	8
	Creating a Compute Agent Configuration File	9
	Installing and Configuring Cloud Analytics Engine Data Learning Engine	11
	Preparing for Installation	11
	Installing Data Learning Engine	12
	Discovering Compute Agents	12
	Log File	12
	Configuring Data Learning Engine	12
	Configuring Cloud Analytics Engine on Devices	13
	Using Cloud Analytics Engine Compute Agent	13
	Using Cloud Analytics Engine Data Learning Engine	14
	Integrating Cloud Analytics Engine with Network Director	14

	Cloud Analytics Engine Known Behaviors	15
	General Known Behaviors	15
	Known Behaviors for Networking Devices That Do Not Support Cloud Analytics Engine	15
	Cloud Analytics Engine Bandwidth Measurement and Mirroring Known Behaviors	16
	Known Behaviors for Cloud Analytics Engine Monitoring Virtual Chassis	16
Part 2	Configuration Statements and Operational Commands	
Chapter 2	Configuration Statements	19
	probe	19
Chapter 3	Operational Commands	21
	show application-monitor probe flows	22
	show application-monitor probe measurements	24
	show application-monitor probe mirrors	26
	show overlay vxlan vni	28
	show overlay vxlan vtep	32

List of Figures

Part 1	Cloud Analytics Engine	
Chapter 1	Using the Cloud Analytics Engine	3
	Figure 1: Bare Metal Application Servers Connected by a Layer 3 IP Fabric Network	4
	Figure 2: KVM Virtual Machine Application Servers Connected by VxLAN Tunnels Overlaid on a layer 3 Fabric Network.	5
	Figure 3: KVM Virtual Machine Application Servers Connected by a Layer 3 IP Fabric Network	5

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	xi
	Table 2: Text and Syntax Conventions	xi
Part 1	Cloud Analytics Engine	
Chapter 1	Using the Cloud Analytics Engine	3
	Table 3: Compute Agent Configuration File JSON Format	9
Part 2	Configuration Statements and Operational Commands	
Chapter 3	Operational Commands	21
	Table 4: show application-monitor probe flows Output Fields	22
	Table 5: show application-monitor probe measurements Output Fields	24
	Table 6: show application-monitor probe mirrors Output Fields	26
	Table 7: show overlay vxlan vni Command Output	28
	Table 8: show overlay vxlan vtep Command Output	32

About the Documentation

- Documentation and Release Notes on page ix
- Supported Platforms on page ix
- Using the Examples in This Manual on page ix
- Documentation Conventions on page xi
- Documentation Feedback on page xiii
- Requesting Technical Support on page xiii

Documentation and Release Notes

To obtain the most current version of all Juniper Networks[®] technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Supported Platforms

For the features described in this document, the following platforms are supported:

- [QFX Series](#)

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see [CLI Explorer](#).

Documentation Conventions

[Table 1 on page xi](#) defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

[Table 2 on page xi](#) defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none">Introduces or emphasizes important new terms.Identifies guide names.Identifies RFC and Internet draft titles.	<ul style="list-style-type: none">A policy <i>term</i> is a named structure that defines match conditions and actions.<i>Junos OS CLI User Guide</i>RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none">To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level.The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i>>;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none">In the Logical Interfaces box, select All Interfaces.To cancel the configuration, click Cancel.

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <http://www.juniper.net/techpubs/feedback/>.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>

- Download the latest versions of software and review release notes:
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:
<http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum:
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Cloud Analytics Engine

- [Using the Cloud Analytics Engine on page 3](#)

CHAPTER 1

Using the Cloud Analytics Engine

- [Cloud Analytics Engine Overview on page 3](#)
- [Installing Junos for Cloud Analytics Engine on page 4](#)
- [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)
- [Installing and Configuring Cloud Analytics Engine Compute Agent on page 7](#)
- [Installing and Configuring Cloud Analytics Engine Data Learning Engine on page 11](#)
- [Configuring Cloud Analytics Engine on Devices on page 13](#)
- [Using Cloud Analytics Engine Compute Agent on page 13](#)
- [Using Cloud Analytics Engine Data Learning Engine on page 14](#)
- [Integrating Cloud Analytics Engine with Network Director on page 14](#)
- [Cloud Analytics Engine Known Behaviors on page 15](#)

Cloud Analytics Engine Overview

Cloud Analytics Engine uses network data analysis to improve application performance and availability. It includes data collection, analysis, correlation, and visualization, helping you better understand the behavior of workloads and applications across the physical and virtual infrastructure. Cloud Analytics Engine provides an aggregated and detailed level of visibility, tying applications and the network together, and an application-centric view of network status, improving your ability to quickly roll out new applications and troubleshoot problems.

Cloud Analytics Engine provides these major capabilities:

- Application visibility and performance management, by controlling application flows and workload placement.
- Capacity planning and optimization, by detecting hotspots and monitoring latency and microbursts.
- Troubleshooting and root cause analysis, by correlating overlay and underlay networks.

Related Documentation

- [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)
- [Cloud Analytics Engine Known Behaviors on page 15](#)

Installing Junos for Cloud Analytics Engine

Cloud Analytics Engine functionality is built into supported Junos releases, and is enabled by default.

Related Documentation

- [Cloud Analytics Engine Overview on page 3](#)
- [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

Understanding Cloud Analytics Engine Architecture and Deployment

This topic describes the architecture and deployment of Cloud Analytics Engine.

This topic includes these sections:

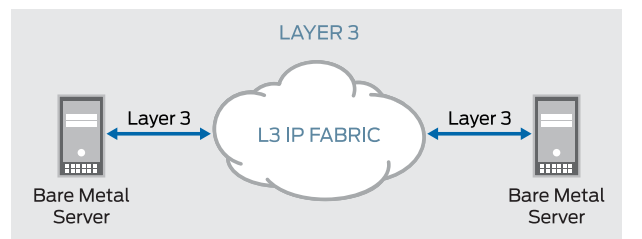
- [Supported Topologies on page 4](#)
- [Software Components on page 5](#)
- [Support for Devices that Do Not Support Cloud Analytics Engine on page 6](#)
- [User Interfaces on page 6](#)
- [Basic Workflow on page 6](#)

Supported Topologies

Cloud Analytics Engine supports data center networks that use these topologies:

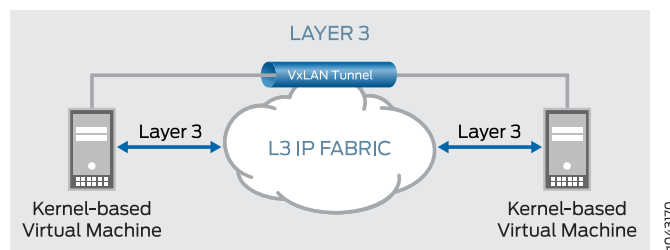
- Bare metal application servers connected by a layer 3 IP fabric network.
[Figure 1 on page 4](#) illustrates this type of topology.

Figure 1: Bare Metal Application Servers Connected by a Layer 3 IP Fabric Network



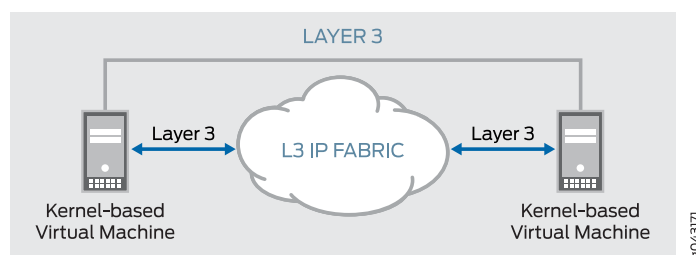
- KVM virtual machine application servers connected by VxLAN tunnels overlaid on a layer 3 fabric network. [Figure 2 on page 5](#) illustrates this type of topology.

Figure 2: KVM Virtual Machine Application Servers Connected by VxLAN Tunnels Overlaid on a layer 3 Fabric Network.



- KVM virtual machine application servers connected by a layer 3 IP fabric network. There is no overlay network. [Figure 3 on page 5](#) illustrates this type of topology.

Figure 3: KVM Virtual Machine Application Servers Connected by a Layer 3 IP Fabric Network



Software Components

Cloud Analytics Engine consists of the following software components:

- Cloud Analytics Engine Junos component — A required component that is built into supported Junos releases. This component processes requests from Compute Agent to collect data. The collected data is sent to Compute Agent.
- Compute Agent — A required software component that is installed on a compute node. The Compute Agent works with the Cloud Analytics Engine Junos component to configure cloud analytics data collection on network devices and collect the requested data. The Compute Agent is controlled by either an API or by the Data Learning Engine component.
- Data Learning Engine — An optional software component that is installed on a compute node. It may not be installed on the same compute node as Compute Agent. It provides longer storage and additional processing of network analytics data. It provides a REST API for integrating with Network Director to allow Network Director to configure analytics data collection and visualize network analytics data. Other applications can also use the REST API to integrate with Cloud Analytics Engine.

Cloud Analytics Engine integrates with Junos Space Network Director to allow Network Director to configure analytics data collection and visualize network analytics data.

Support for Devices that Do Not Support Cloud Analytics Engine

Cloud Analytics Engine will work with devices that do not support Cloud Analytics Engine, including non-Juniper devices, in the application flow path, but the only data it will collect from them is their IP address and hop number.

It is recommended that network devices that connect to application servers (called penultimate hop devices) should be Juniper devices that support Cloud Analytics Engine. If you have penultimate hop devices that do not support Cloud Analytics Engine, ensure that your data collection requests specify an accurate hop count.

User Interfaces

Cloud Analytics Engine has several user interfaces:

- On networking devices that support Cloud Analytics Engine, CLI statements and commands allow you enable and disable cloud analytics data collection, and view some of the collected cloud analytics data.
- The Compute Agent API allows you to configure and collect cloud analytics data.
- The Data Learning Engine REST API allows you to configure and collect cloud analytics data. It uses the Compute Agent API to collect data. Data Learning Engine performs some additional data processing, so it can provide more types of data than Compute Agent.
- Network Director integrates with Data Learning Engine to allow you to configure, collect, and visualize cloud analytics data using the Network Director user interface. Network Director uses the Data Learning Engine REST API to configure and collect cloud analytics data, and it can perform additional data processing, so it can provide more types of data than Data Learning Engine.

Basic Workflow

This is the basic Cloud Analytics Engine workflow:

1. You start generating cloud analytics data.

You can use several methods to start generating cloud analytics data: Network Director, the Data Learning Engine REST API, or the Compute Agent API.

You select what data to generate by specifying an application flow or a VxLAN tunnel to monitor. Each device in the path of a monitored application flow or tunnel that supports Cloud Analytics Engine generates cloud analytics data for that flow or tunnel.

2. You view or collect the generated data.

You can use show commands on networking devices to view some cloud analytics data. You can use several methods to collect cloud analytics data: Network Director, the Data Learning Engine REST API, or the Compute Agent API. You select what data to collect by specifying an application flow or a VxLAN tunnel.

3. You visualize the collected data.

You can use the Compute Agent API and the Data Learning Engine REST API to collect analytics data. Both APIs return analytics data in JSON format. If you use Network Director, it reads the data and provides features to visualize it.

- Related Documentation**
- [Cloud Analytics Engine Overview on page 3](#)
 - [Cloud Analytics Engine Known Behaviors on page 15](#)
 - [Cloud Analytics Engine Compute Agent API Reference](#)
 - [Cloud Analytics Engine Data Learning Engine REST API Reference](#)

Installing and Configuring Cloud Analytics Engine Compute Agent

Cloud Analytics Engine Compute Agent (Compute Agent) runs on a server running CentOS 6.5. Compute Agent can run on a CentOS native server or on a KVM hypervisor. Install Compute Agent on each application server that you want to monitor with Cloud Analytics Engine.

The Compute Agent must have network connectivity to all networking devices that it will monitor. Network port 8080 on the Compute Agent server must be accessible by all devices to be monitored. If you change that port number from the default 8080, ensure that the new port number you specify is accessible.

Compute Agent is delivered as an RPM file. You install it using the operating system's standard package installation procedure.



NOTE: Compute Agent does not support prelinking. Compute Agent related binaries are ignored during prelinking. This configuration is stored as a new file `/etc/prelink.conf.d/cagent.conf`, which is created during Compute Agent RPM installation. Compute Agent requires prelink version 0.4.0-1 or higher for this configuration to work correctly.

Compute Agent requires some configuration. To configure it, you have these options:

- After installing Compute Agent, run the interactive setup program and enter configuration parameters.
- Create a configuration file in a specified location on the server, then install Compute Agent. The Compute Agent installer will detect the configuration file and use the configuration defined in it instead of running the interactive setup program.

The methods of configuring Compute Agent are described in these sections:

- [Configuring Compute Agent by Running the Interactive Setup Program on page 8](#)
- [Configuring Compute Agent Initial Configuration by Using a Configuration File on page 8](#)
- [Creating a Compute Agent Configuration File on page 9](#)

Configuring Compute Agent by Running the Interactive Setup Program

To configure Compute Agent by running the interactive setup program:

1. Enter the command **cagent configure** to start the setup program.
2. Enter the type of environment in which the Compute Agent is installed:
CA Environment (options - Native, Hypervisor_KVM): Hypervisor_KVM
3. If you entered a hypervisor CA environment type, Enter the type of tunnelling used in the network that the Compute Agent will monitor. If you enter nothing, the default value is None.
Tunnel used (options - None, VXLAN): None
4. Enter a Compute Agent server name. If the server is registered in DNS, the DNS server name is provided as the default value:
CA server name (press enter to use system FQDN value 'host1.example.net'):
5. Enter a Compute Agent server IP address. If the server is registered in DNS, the DNS IP address is provided as the default value:
CA server IP Address (press enter to use system FQDN derived value '10.94.198.110')::
6. Enter the physical interfaces on the server that are connected to the network, separated by commas. Do not include spaces between the port names:
Network connected ports (eg: eth0, eth1) : eth0,lo
7. (Optional) Enter the web server IP address:
Web server address (optional, press enter to skip):
8. (Optional) Enter the web server IP interface name:
Web server interface name (optional, press enter to skip): eth0
9. (Optional) Enter the web server port:
Web server port (optional, default:8080): 80
10. (Optional) Enter the Compute Agent interface name:
CA Interface name (optional, press enter to skip): eth0
11. (Optional) Enter the Compute Agent log directory. If you enter nothing, the default log directory is **/var/log/cagent/**.
CA log directory (optional, default:/var/log/cagent/): /var/log/cagent/

Configuring Compute Agent Initial Configuration by Using a Configuration File

Rather than running the interactive setup program after installing Compute Agent, you can create a configuration file that the Compute Agent installer will use to configure Compute Agent during installation.

To configure Compute Agent by using a configuration file:

1. Create a Compute Agent configuration file named **cagent.conf**.

See [“Creating a Compute Agent Configuration File” on page 9](#) for more information about creating this file.

2. Put the Compute Agent configuration file in this location on the Compute Agent server: **/etc/cagent/cagent.conf**.
3. Install the Compute Agent using the operating system's standard package installation procedure.

The Compute Agent installer will detect the configuration file and use the configuration defined in it instead of running the interactive setup program.

Creating a Compute Agent Configuration File

Compute Agent reads a configuration file at startup, which contains configuration settings for Compute Agent to use. The Compute Agent configuration file is a text file in JSON format. The syntax of the file is described in [Table 3 on page 9](#).

Table 3: Compute Agent Configuration File JSON Format

Variable	Values	Description	Mandatory?
Section: Environment			
CA Mode	Native, Hypervisor_KVM	The type of environment in which the Compute Agent is installed. Enter Native if you are installing Compute Agent on a native server (not a virtual machine). Enter Hypervisor_KVM if you are installing Compute Agent on a KVM hypervisor.	Yes
CA Server IP Address	IP address	IP address of the Compute Agent server.	No
CA Server Name	Server name	Compute Agent server name.	No
Tunnel Mode	None, VXLAN	The type of tunnelling used in the network that the Compute Agent will monitor.	No Default value: None
Section: Network Ports			
Interface Name	Names of server physical interfaces connected to the network. Examples: eth0, br0.	The physical interfaces on the server that are connected to the network. Compute Agent monitors these to create a database of active flows.	Yes
IPv4 address	IPv4 address	IPv4 address of a network port.	No
IPv6 address	IPv6 address	IPv6 address of a network port.	No
Mac address	Mac address	Mac address of a network port.	No

Table 3: Compute Agent Configuration File JSON Format (*continued*)

Variable	Values	Description	Mandatory?
Section: CA interface This configuration section is optional and will be used only if the Tunnel Mode variable is set to None . The CA interface is the network interface the Compute Agent uses for receiving probe responses from devices.			
Interface Name	Names of server physical interfaces. Examples: eth0, br0.	Interface name	No
IP Address Mode	Static, DHCP	How the CA interface gets a an IP address, either by static configuration or DHCP. If you enter Static , you must assign an IP address using the IP Address variable in this section.	If an interface name is specified, this field is mandatory.
IP Address	IPv4 address	Static IPv4 Address for the CA interface	If the IP Address Mode variable is set to Static then this field is mandatory.
Netmask	IPv4 network mask	IPv4 network mask for the CA interface IP address	If the IP Address Mode variable is set to Static then this field is mandatory.
Section: CA Web Interface This section is optional. The CA web interface is the interface for the Compute Agent API.			
Interface Name	Names of server physical interfaces. Examples: eth0, br0.	Interface name	No
IP Address	IPv4 address	Static IPv4 Address for the CA web interface	No
Port	Valid TCP port	TCP port on which Compute Agent listens for web services	No Default value: 8080
Section: CA Logging .This section is optional.			
Log Directory	Complete path to the directory where logs will be saved in the file cagent.log .	Optional	Optional Default value: /var/log/cagent/cagent.log

This is an example of a Compute Agent configuration file:


```
{
  "CA Web Interface": {
    "IP Address": "10.94.201.8",
    "TCP Port": 8080
  },
  "Environment": {
    "CA Mode": "Native",
    "CA Server IP Address": "10.94.201.8",
    "CA Server Name": "host1.example.net"
  },
  "Network Ports": [
    {
      "IPv4 address": [
        "10.0.0.2"
      ],
      "IPv6 address": [
        "2001:db8:215:17ff:feab:aef9%eth1"
      ],
      "Interface Name": "eth1",
      "Mac address": "00:15:17:ab:ae:f9"
    },
  ]
}
```

- Related Documentation**
- [Cloud Analytics Engine Overview on page 3](#)
 - [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

Installing and Configuring Cloud Analytics Engine Data Learning Engine

This topic describes how to install and configure Data Learning Engine.

This topic includes these sections:

- [Preparing for Installation on page 11](#)
- [Installing Data Learning Engine on page 12](#)
- [Discovering Compute Agents on page 12](#)
- [Log File on page 12](#)
- [Configuring Data Learning Engine on page 12](#)

Preparing for Installation

Cloud Analytics Engine Data Learning Engine (Data Learning Engine) runs on a compute node running CentOS 6.5. The server can be native or a KVM virtual machine.

The following server and network configurations are required for Data Learning Engine:

- The Data Learning Engine must have network connectivity to all Compute Agents that it will communicate with.
- The Data Learning Engine server, all devices to be monitored, all Compute Agent servers, and the Network Director server (if you are using Network Director) must have network connectivity to each other, and have the following system time configurations:

- Configured with the same time zone.
- System clocks synchronized by the same Network Time Protocol (NTP) server.
- The following network ports on the Data Learning Engine server must be accessible to all devices to be monitored, and to the Network Director server (if you are using Network Director): 8080, 4242, 50005, 50006, 9160, 7000, 8282, 8081, and 9042. If you change any of these default port numbers, ensure that the new port numbers you specify are accessible.

Installing Data Learning Engine

Data Learning Engine is delivered as an RPM file. Install it using the operating system's package installation process.

Discovering Compute Agents

To inform Data Learning Engine about the Compute Agents it can work with, do the following:

1. Append the contents of the file `/etc/cagent/cagent.conf` on each Compute Agent server to the file `/opt/cae/dle/compute-agents/ca-discovery.json` on the Data Learning Engine server.
2. Restart the Data Learning Engine by entering the command **service dle restart**.

Log File

The Data Learning Engine log file is `/opt/cae/dle/log/dle.log`.

Configuring Data Learning Engine

Data Learning Engine is installed with a default configuration. You can configure it by editing properties in several properties files, then restarting the Data Learning Engine service by entering the command **service dle restart**. These properties files contain comments describing the properties and their possible values.

The properties files that you can edit are in the directory `/opt/cae/dle/conf`:

- **dle.yaml**—Contains the Data Learning Engine service configuration. Data Learning Engine reads this configuration file and starts all the Data Learning Engine services that are marked as 'start: true'.
- **datastore.properties**—The property `net.juniper.analytics.dle.cassandra.cluster` sets the comma separated hostnames to configure the Cassandra host names.
- **tcp-collector.properties**—Used by the High Frequency Statistics collector.
- **webserver.properties**—Sets web server properties:
 - `net.juniper.analytics.dle.webserver.hostname`—Sets the Data Learning Engine web server host name.
 - `net.juniper.analytics.dle.webserver.port`—Sets the Data Learning Engine web server port.

- Related Documentation**
- [Cloud Analytics Engine Overview on page 3](#)
 - [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

Configuring Cloud Analytics Engine on Devices

By default, supported networking devices accept and process the network probes that the Compute Agent sends to communicate with networking devices. When you disable probe handling on a device, the device responds to probes like a device that does not support Cloud Analytics Engine. Only IP address and hop count are reported for the device.

1. To disable Cloud Analytics Engine probes on the device:

set services analytics probe disable

2. To enable Cloud Analytics Engine probes on the device:

set services analytics probe enable

- Related Documentation**
- [Cloud Analytics Engine Overview on page 3](#)
 - [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

Using Cloud Analytics Engine Compute Agent

Cloud Analytics Engine Compute Agent (Compute Agent) is a software component that is installed on a compute node. The Compute Agent works with the Cloud Analytics Junos component to configure cloud analytics data collection on networking devices and collect the requested data. Compute Agent is controlled by either an API or by the Data Learning Engine.

Compute Agent generates network analytics data by sending probe packets onto the network that emulate application flow packets. Analytics data is generated on each networking device that handles a probe packet. To view or collect the analytics data, you have these options:

- Enter cloud analytics show commands on a networking device.
- Use the Compute Agent API.
- Use the Data Learning Engine REST API.
- Use Network Director.

When you use the Compute Agent API to collect analytics data, the Compute Agent returns the data in JSON format.

Compute Agent has an administration program, which is located at `/usr/local/bin/cagent`. To administer Compute Agent using this program:

1. Log into the Compute Agent server.
2. To stop Compute Agent, enter the command `/usr/local/bin/cagent stop`.

3. To start Compute Agent, enter the command `/usr/local/bin/cagent start`.
4. To configure Compute Agent, enter the command `/usr/local/bin/cagent configure`.
This command runs the interactive setup program, which is described in [“Installing and Configuring Cloud Analytics Engine Compute Agent” on page 7](#).
5. To view help for the configure command, enter the command `/usr/local/bin/cagent -h`.

**Related
Documentation**

- [Cloud Analytics Engine Overview on page 3](#)
- [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

Using Cloud Analytics Engine Data Learning Engine

Cloud Analytics Engine Data Learning Engine (Data Learning Engine) is an optional software component that is installed on a compute node. Data Learning Engine works with Compute Agents installed on application servers to configure analytics data collection and retrieval. It does some processing of the data, so it can provide richer data than Compute Agent. You can use Data Learning Engine in these ways:

- Integrate it with Junos Space Network Director, which enables Network Director to configure analytics data collection and visualize network analytics data. See the Network Director documentation for information about this integration.
- Use the Data Learning Engine REST API, which is documented in the *Cloud Analytics Engine REST API Reference Guide*.

**Related
Documentation**

- [Cloud Analytics Engine Overview on page 3](#)
- [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

Integrating Cloud Analytics Engine with Network Director

Cloud Analytics Engine can integrate with Junos Space Network Director (Network Director) to enable Network Director to configure analytics data collection and visualize network analytics data. Integrating with Network Director enables the maximum Cloud Analytics Engine functionality. See [Understanding Cloud Analytics Engine and Network Director](#) in the Network Director documentation for details on how these two products work together.

**Related
Documentation**

- [Cloud Analytics Engine Overview on page 3](#)
- [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

Cloud Analytics Engine Known Behaviors

This topic describes Cloud Analytics Engine known behaviors.

Known behaviors in this topic are categorized into these sections:

- [General Known Behaviors on page 15](#)
- [Known Behaviors for Networking Devices That Do Not Support Cloud Analytics Engine on page 15](#)
- [Cloud Analytics Engine Bandwidth Measurement and Mirroring Known Behaviors on page 16](#)
- [Known Behaviors for Cloud Analytics Engine Monitoring Virtual Chassis on page 16](#)

General Known Behaviors

- All application servers and networking devices to be monitored must be able to send and receive ICMP, UDP, and TCP traffic.
- To downgrade to a previous version of Compute Agent, uninstall the existing version by entering the command **rpm -e cagent**, then install the previous version.
- Egress IFL statistics for an IRB interface will be encoded zero if the ARP for the destination IP address of the flow is not yet resolved on the switch.
- AE/LAG link members that are down are reported by device for egress IFL statistics in the probe response.
- If NTP is enabled on a device, probe responses will not indicate timestamp as NTP type. Instead it will mark it as LCPU.
- If the destination IP Address's ARP is not resolved at the penultimate HOP, then it will not be recognized as a penultimate HOP.
- The detailed output of the commands **show overlay vxlan vni** and **show overlay vxlan vtep** always shows a value of 0 in the fields **Source VM Port**, **Destination VM Port**, and **Proto**.

Known Behaviors for Networking Devices That Do Not Support Cloud Analytics Engine

- Cloud Analytics Engine will retrieve only the IP address and hop count from networking devices that do not support Cloud Analytics Engine.
- It is recommended that network devices that connect to application servers (called penultimate hop devices) should be Juniper devices that support Cloud Analytics Engine. If you have penultimate hop devices that do not support Cloud Analytics Engine:
 - Ensure that your data collection requests specify an accurate hop count.
 - Cloud Analytics Engine probes might be consumed.

Cloud Analytics Engine Bandwidth Measurement and Mirroring Known Behaviors

- For a Cloud Analytics Engine probe to install an ERSPAN mirror filter, the analyzer IP address should be a unicast route type in the route forwarding table. Additionally, mirror will fail to install if the analyzer IP address resolves to a unicast route over an AE (LAG) interface.
- Bandwidth measurement and mirroring are not supported in the egress direction.
- These limitations of Junos on QFX Series devices apply to Cloud Analytics Engine operation:
 - Four analyzers (mirror destinations) are supported.
 - The analyzer IP address should not be reachable by the management interface.
- Mirror does not respond to route changes for the analyzer IP address.
- Bandwidth measurement and mirroring are not supported for ingress IRB interfaces.
- When the ERSPAN encapsulation causes the packet size to exceed the interface MTU, these packets will be dropped.
- ERSPAN mirror install via Cloud Analytics Engine and CLI may interact unfavorably. If Cloud Analytics Engine tries to install a mirror to an analyzer IP Address that is already configured via CLI, then the Cloud Analytics Engine mirror install request will fail. Similarly if an analyzer is already configured via Cloud Analytics Engine, then the CLI request to install an ERSPAN related configuration for that analyzer IP address will fail.
- When the bandwidth measurement counters on the device are zero, the device doesn't send the bandwidth counter TLV to compute Agent.

Known Behaviors for Cloud Analytics Engine Monitoring Virtual Chassis

- On a Virtual Chassis, when the flow path uses different FPCs for ingress and egress, no interface statistics are generated for the egress interface. This also affects fetching of ECMP Next Hop. The probe is forwarded along the flow path, and ingress interface statistics are generated.
- If a mirror next hop is on a different line card, mirror install for that flow will fail.

Related Documentation

- [Cloud Analytics Engine Overview on page 3](#)
- [Understanding Cloud Analytics Engine Architecture and Deployment on page 4](#)

PART 2

Configuration Statements and Operational Commands

- Configuration Statements on page 19
- Operational Commands on page 21

CHAPTER 2

Configuration Statements

- [probe on page 19](#)

probe

Syntax	probe [enable disable]
Hierarchy Level	[edit services analytics]
Release Information	Statement introduced in Junos OS Release 14.1X53-D15 for the QFX Series.
Description	By default, supported networking devices accept and process the network probes that the Compute Agent sends to communicate with networking devices. When you disable probe handling on a device, the device responds to probes like a device that does not support Cloud Analytics Engine. Only IP address and hop count are reported for the device.
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Cloud Analytics Engine Overview on page 3• Understanding Cloud Analytics Engine Architecture and Deployment on page 4

CHAPTER 3

Operational Commands

- `show application-monitor probe flows`
- `show application-monitor probe measurements`
- `show application-monitor probe mirrors`
- `show overlay vxlan vni`
- `show overlay vxlan vtep`

show application-monitor probe flows

Syntax	show application-monitor probe flows
Release Information	Command introduced in Junos OS Release 15.1X53-D30 on the QFX Series.
Description	Display information about all the flows that Cloud Analytics Engine is monitoring.
Options	This command has no options.
Additional Information	
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Cloud Analytics Engine Overview on page 3 • Understanding Cloud Analytics Engine Architecture and Deployment on page 4
List of Sample Output	show application-monitor probe flows on page 23
Output Fields	Table 4 on page 22 describes the output fields for the show application-monitor probe flows command. Output fields are listed in the approximate order in which they appear.

Table 4: show application-monitor probe flows Output Fields

Field Name	Field Description
*	Indicates overlay flow. Appears in the column to the left of the Flow ID .
+	Indicates pending requests for this flow.
Flow ID	Flow ID.
Interface	Interface name.
Src-Addr	Source IP address of the flow.
Src Port	Source port of the flow.
Dest-Addr	Destination IP address of the flow.
Dest Port	Destination port of the flow.
Proto	Protocol of the flow.
BW	Bandwidth measurement configuration. Supported values: "IN"=ingress.
Mirror	Mirroring configuration. Supported mirroring configurations: "IN"=ingress.

Sample Output

show application-monitor probe flows

```
user@host> show application-monitor probe flows
* - Indicates Overlay Flow
+ - Indicates Pending requests for this Flow
```

Flow Mirror ID	Interface	Src-Addr	Src Port	Dest-Addr	Dest Port	Proto	BW
1	et-0/0/2.0	10.0.2.101	37710	10.0.1.101	5677	UDP	IN
NONE 2	et-0/0/63.0	10.0.1.101	52936	10.0.2.101	5678	UDP	IN
NONE 3	et-0/0/2.0	10.0.2.101	42407	10.0.1.101	5677	UDP	NONE IN
* 4	et-0/0/63.0	10.0.1.101	48203	10.0.2.101	5678	UDP	NONE IN

show application-monitor probe measurements

Syntax	show application-monitor probe measurements <flow-id <i>flow-id</i> >
Release Information	Command introduced in Junos OS Release 15.1X53-D30 on the QFX Series.
Description	Display bandwidth measurement information about a flow or flows that Cloud Analytics Engine is monitoring.
Options	flow-id <i>flow-id</i> —(Optional) Flow ID (numeric value > 0) of the flow for which to display bandwidth measurement information. If not specified, display information about all active flows.
Additional Information	
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Cloud Analytics Engine Overview on page 3 • Understanding Cloud Analytics Engine Architecture and Deployment on page 4
List of Sample Output	show application-monitor probe measurements—for all active flows (no flow-id specified) on page 25 show application-monitor probe measurements—for a specified flow on page 25
Output Fields	Table 5 on page 24 describes the output fields for the show application-monitor probe measurements command. Output fields are listed in the approximate order in which they appear.

Table 5: show application-monitor probe measurements Output Fields

Field Name	Field Description
Flow ID	Flow ID.
Interface	Interface name.
Src-Addr	Source IP address of the flow.
Src Port	Source port of the flow.
Dest-Addr	Destination IP address of the flow.
Dest Port	Destination port of the flow.
Proto	Protocol of the flow.
BW	Bandwidth measurement configuration. Supported values: "IN"=ingress.

Table 5: show application-monitor probe measurements Output Fields (*continued*)

Field Name	Field Description
Counter	Bandwidth counter. Displayed only when a <i>flow-id</i> is specified.

Sample Output

show application-monitor probe measurements—for all active flows (no flow-id specified)

```
user@host> show application-monitor probe measurements
```

```
* - Indicates Overlay Flow
```

```
+ - Indicates Pending requests for this Flow
```

Flow ID	Interface	Src-Addr	Src Port	Dest-Addr	Dest Port	Proto	BW
1	et-0/0/2.0	10.0.2.101	37710	10.0.1.101	5677	UDP	IN
2	et-0/0/63.0	10.0.1.101	52936	10.0.2.101	5678	UDP	IN

show application-monitor probe measurements—for a specified flow

```
user@host> show application-monitor probe measurements flow-id 1
```

```
* - Indicates Overlay Flow
```

```
+ - Indicates Pending requests for this Flow
```

Flow ID	Interface	Src-Addr	Src Port	Dest-Addr	Dest Port	Proto	BW
1	et-0/0/2.0	10.0.2.101	37710	10.0.1.101	5677	UDP	IN
Counter							30642

show application-monitor probe mirrors

Syntax	show application-monitor probe mirrors <flow-id <i>flow-id</i> >
Release Information	Command introduced in Junos OS Release 15.1X53-D30 on the QFX Series.
Description	Display information about the mirrors for an active flow or flows configured by Cloud Analytics Engine.
Options	flow-id <i>flow-id</i> —(Optional) Flow ID (numeric value > 0) of the flow for which to display mirror information. If not specified, display information about all active flows with mirrors.
Additional Information	
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Cloud Analytics Engine Overview on page 3 • Understanding Cloud Analytics Engine Architecture and Deployment on page 4
List of Sample Output	show application-monitor probe mirrors—for all active flows (no flow-id specified) on page 27 show application-monitor probe mirrors—for a specified flow on page 27
Output Fields	Table 6 on page 26 lists the output fields for the show application-monitor probe mirrors command. Output fields are listed in the approximate order in which they appear.

Table 6: show application-monitor probe mirrors Output Fields

Field Name	Field Description
Flow ID	Flow ID.
Interface	Interface name.
Src-Addr	Source IP address of the flow.
Src Port	Source port of the flow.
Dest-Addr	Destination IP address of the flow.
Dest Port	Destination port of the flow.
Proto	Protocol of the flow.
Mirror	Mirroring configuration. Supported mirror configurations: "IN"=ingress.

Table 6: show application-monitor probe mirrors Output Fields (*continued*)

Field Name	Field Description
Mirror-Type	Type of mirroring installed. Supported mirror types: "ERSPAN".
Analyzer-IP	Mirror analyzer IP address.

Sample Output

show application-monitor probe mirrors—for all active flows (no flow-id specified)

```

user@host> show application-monitor probe mirrors
* - Indicates Overlay Flow
+ - Indicates Pending requests for this Flow

Flow  Interface      Src-Addr      Src  Dest-Addr      Dest  Proto Mirror
ID                                     Port
3     et-0/0/2.0      10.0.2.101    42407 10.0.1.101      5677  UDP   IN
      Mirror-Type  ERSPAN        Analyzer-IP 10.0.0.101
4     et-0/0/63.0     10.0.1.101    48203 10.0.2.101      5678  UDP   IN
      Mirror-Type  ERSPAN        Analyzer-IP 10.0.0.101

```

show application-monitor probe mirrors—for a specified flow

```

user@host> show application-monitor probe mirrors flow-id 3
* - Indicates Overlay Flow
+ - Indicates Pending requests for this Flow

Flow  Interface      Src-Addr      Src  Dest-Addr      Dest  Proto Mirror
ID                                     Port
3     et-0/0/2.0      10.0.2.101    42407 10.0.1.101      5677  UDP   IN
      Mirror-Type  ERSPAN        Analyzer-IP 10.0.0.101

```

show overlay vxlan vni

Syntax	show overlay vxlan vni <vni-id> <summary detail>
Release Information	Command introduced in Junos OS Release 15.1X53-D30 on the QFX Series.
Description	Display information about Virtual Extensible LANs (VXLANs) for overlay application flows that are being monitored by Cloud Analytics Engine.
Options	<p>vni-id—(Optional) Virtual Extensible LAN Network Identifier (VNI) that identifies a VXLAN. Display information about the specified VNI for an overlay application flow. If this option is specified with no other options, the default display is the same as the summary option.</p> <p>summary—(Optional) Display summary information about the specified VNI, or all VNIs being monitored if no vni-id is specified.</p> <p>detail—(Optional) Display more detailed information about the specified VNI, or all VNIs being monitored if no vni-id is specified.</p> <p>None—By default, if no options are specified, display summary information about all VNIs being monitored.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Cloud Analytics Engine Overview on page 3 • Understanding Cloud Analytics Engine Architecture and Deployment on page 4
List of Sample Output	<p>show overlay vxlan vni—for all monitored VNIs (no options specified, default display is summary format) on page 30</p> <p>show overlay vxlan vni summary—for all monitored VNIs (no vni-id specified) on page 30</p> <p>show overlay vxlan vni detail—for all monitored VNIs (no vni-id specified) on page 30</p> <p>show overlay vxlan vni <vni-id> detail—detailed view for specified VNI on page 31</p>
Output Fields	Table 7 on page 28 describes the output fields for the show overlay vxlan vni command. Output fields are listed in the approximate order in which they appear for summary or detailed displays.

Table 7: show overlay vxlan vni Command Output

Field Name	Field Description
Summary Option Output Fields	
Virtual Extensible LAN Network Identifier (VNI)	VXLAN identifier (VNI) for the VXLAN being monitored. When showing results for multiple VNIs, all output following this field until the next field of this type pertains to the specified VNI.

Table 7: show overlay vxlan vni Command Output (*continued*)

Field Name	Field Description
Source Virtual Tunnel End Point (VTEP) - IP-address	IP address of the source VTEP.
Destination Virtual Tunnel End Point (VTEP) - IP-address	IP address of the destination VTEP.
Source VM IP	IP address of the source VM.
Source VM MAC	MAC address of the source VM.
Dest VM IP	IP address of the destination VM.
Dest VM MAC	MAC address of the destination VM.
Detail Option Output Fields	
Virtual Extensible LAN Network Identifier (VNI)	VNI for the VXLAN being monitored. When showing results for multiple VNIs, all output following this field until the next field of this type pertains to the specified VNI.
Src VTEP IP-Addr	IP address of the source VTEP.
Src VTEP Port	Port of the source VTEP.
Interface	Interface of the source VTEP.
Dest VTEP IP-Addr	IP address of the destination VTEP.
Dest VTEP Port	Port of the destination VTEP.
Filters Installed	Filters installed on the VNI (if configured).
Bandwidth Counter	Bandwidth counter (if bandwidth monitoring is configured).
Detail Option Output fields—Application Flow Details	
Source VM IP	IP address of the source VM.
Destination VM IP	IP address of the destination VM.
Source VM MAC	MAC address of the source VM.
Destination VM MAC	MAC address of the destination VM.
Source VM Port	Port of the source VM.
Destination VM Port	Port of the destination VM.
Proto	Protocol of the flow.

Sample Output

show overlay vxlan vni—for all monitored VNIs (no options specified, default display is summary format)

```
user@host> show overlay vxlan vni
Virtual Extensible LAN Network Identifier: 100

Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.1.101
Destination Virtual Tunnel End Point (VTEP) - IP-address: 10.0.2.101

Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
10.1.1.1          ce:50:71:5b:8a:43  10.2.2.2        76:ca:b7:a2:58:4e

Virtual Extensible LAN Network Identifier: 200

Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.2.101
Destination Virtual Tunnel End Point (VTEP) - IP-address: 10.0.1.101

Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
10.2.2.2          76:ca:b7:a2:58:4e  10.1.1.1        ce:50:71:5b:8a:43
```

show overlay vxlan vni summary—for all monitored VNIs (no vni-id specified)

```
user@host> show overlay vxlan vni summary
Virtual Extensible LAN Network Identifier: 100

Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.1.101
Destination Virtual Tunnel End Point (VTEP) - IP-address: 10.0.2.101

Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
10.1.1.1          ce:50:71:5b:8a:43  10.2.2.2        76:ca:b7:a2:58:4e

Virtual Extensible LAN Network Identifier: 200

Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.2.101
Destination Virtual Tunnel End Point (VTEP) - IP-address: 10.0.1.101

Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
10.2.2.2          76:ca:b7:a2:58:4e  10.1.1.1        ce:50:71:5b:8a:43
```

show overlay vxlan vni detail—for all monitored VNIs (no vni-id specified)

```
user@host> show overlay vxlan vni detail
Virtual Extensible LAN Network Identifier: 100

Src VTEP IP-Addr: 10.0.1.101      Src VTEP Port: 1234  Interface: et-0/0/63.0

Dest VTEP IP-Addr: 10.0.2.101      Dest VTEP Port: 4789

Application Flow Details:-
Source VM IP: 10.1.1.1              Destination VM IP: 10.2.2.2
Source VM MAC: ce:50:71:5b:8a:43    Destination VM MAC: 76:ca:b7:a2:58:4e
Source VM Port: 0                    Destination VM Port: 0
Proto: 0

Virtual Extensible LAN Network Identifier: 200

Src VTEP IP-Addr: 10.0.2.101      Src VTEP Port: 1234  Interface: et-0/0/2.0
```

```
Dest VTEP IP-Addr: 10.0.1.101      Dest VTEP Port: 4789

Application Flow Details:-
Source VM IP: 10.2.2.2             Destination VM IP: 10.1.1.1
Source VM MAC: 76:ca:b7:a2:58:4e   Destination VM MAC: ce:50:71:5b:8a:43
Source VM Port: 0                  Destination VM Port: 0
Proto: 0
```

show overlay vxlan vni <vni-id> detail—detailed view for specified VNI

```
user@host> show overlay vxlan vni 100 detail
Virtual Extensible LAN Network Identifier: 100

Src VTEP IP-Addr: 10.0.1.101      Src VTEP Port: 1234  Interface: et-0/0/63.0

Dest VTEP IP-Addr: 10.0.2.101      Dest VTEP Port: 4789

Application Flow Details:-
Source VM IP: 10.1.1.1             Destination VM IP: 10.2.2.2
Source VM MAC: ce:50:71:5b:8a:43   Destination VM MAC: 76:ca:b7:a2:58:4e
Source VM Port: 0                  Destination VM Port: 0
Proto: 0
```

show overlay vxlan vtep

Syntax	<code>show overlay vxlan vtep</code> <code><vtep-source-address></code> <code><summary detail></code>
Release Information	Command introduced in Junos OS Release 15.1X53-D30 on the QFX Series.
Description	Display information about Virtual Tunnel End Points (VTEPs) for overlay application flows that are being monitored by Cloud Analytics Engine.
Options	<p>vtep-source-address—(Optional) Source IP address of a VTEP. Display information about the VTEP with the specified source IP address for an overlay application flow. If this option is specified with no other options, the default display is the same as the summary option.</p> <p>summary—(Optional) Display summary information about the specified VTEP, or all VTEPs being monitored if no vtep-source-address is specified.</p> <p>detail—(Optional) Display more detailed information about the specified VTEP, or all VTEPs being monitored if no vtep-source-address is specified.</p> <p>None—By default, if no options are specified, display summary information about all VTEPs being monitored.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Cloud Analytics Engine Overview on page 3 • Understanding Cloud Analytics Engine Architecture and Deployment on page 4
List of Sample Output	<p>show overlay vxlan vtep—for all monitored VTEPs (no options specified, default display is summary format) on page 34</p> <p>show overlay vxlan vtep summary—for all monitored VTEPs (no vtep-source-address specified) on page 34</p> <p>show overlay vxlan vtep detail—for all monitored VTEPs (no vtep-source-address specified) on page 34</p> <p>show overlay vxlan vtep <vtep-source-address> detail—detailed view for specified VTEP on page 35</p>
Output Fields	Table 8 on page 32 lists the output fields for the show overlay vxlan vtep command. Output fields are listed in the approximate order in which they appear.

Table 8: show overlay vxlan vtep Command Output

Field Name	Field Description
Summary Option Output Fields	

Table 8: show overlay vxlan vtep Command Output (*continued*)

Field Name	Field Description
Source Virtual Tunnel End Point (VTEP) - IP-address	IP address of the source VTEP. When showing results for multiple VTEPs, all output following this field until the next field of this type pertains to this VTEP.
Destination VTEP - IP-address	IP address of the destination VTEP.
VxLAN ID (VNI)	Virtual Extensible LAN Network Identifier (VNI) for the VTEP.
Source VM IP	IP address of the source VM.
Source VM MAC	MAC address of the source VM.
Dest VM IP	IP address of the destination VM.
Dest VM MAC	MAC address of the destination VM.
Detail Option Output Fields	
Src VTEP IP-Addr	IP address of the source VTEP. When showing results for multiple VTEPs, all output following this field until the next field of this type pertains to this VTEP.
Src VTEP Port	Port of the source VTEP.
Interface	Interface of the source VTEP.
Dest VTEP IP-Addr	IP address of the destination VTEP.
Dest VTEP Port	Port of the destination VTEP.
VxLAN ID (VNI)	VNI for the VTEP.
Detail Option Output fields—Application Flow Details	
Source VM IP	IP address of the source VM.
Destination VM IP	IP address of the destination VM.
Source VM MAC	MAC address of the source VM.
Destination VM MAC	MAC address of the destination VM.
Source VM Port	Port of the source VM.
Destination VM Port	Port of the destination VM.
Proto	Protocol of the flow.

Sample Output

show overlay vxlan vtep—for all monitored VTEPs (no options specified, default display is summary format)

```
user@host> show overlay vxlan vtep
Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.1.101

      Destination VTEP - IP-address: 10.0.2.101      VxLAN ID (VNI): 100

      Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
      10.1.1.1          ce:50:71:5b:8a:43  10.2.2.2        76:ca:b7:a2:58:4e

Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.2.101

      Destination VTEP - IP-address: 10.0.1.101      VxLAN ID (VNI): 200

      Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
      10.2.2.2          76:ca:b7:a2:58:4e  10.1.1.1        ce:50:71:5b:8a:43
```

show overlay vxlan vtep summary—for all monitored VTEPs (no vtep-source-address specified)

```
user@host> show overlay vxlan vtep summary
Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.1.101

      Destination VTEP - IP-address: 10.0.2.101      VxLAN ID (VNI): 100

      Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
      10.1.1.1          ce:50:71:5b:8a:43  10.2.2.2        76:ca:b7:a2:58:4e

Source Virtual Tunnel End Point (VTEP) - IP-address: 10.0.2.101

      Destination VTEP - IP-address: 10.0.1.101      VxLAN ID (VNI): 200

      Source VM IP      Source VM MAC      Dest VM IP      Dest VM MAC
      10.2.2.2          76:ca:b7:a2:58:4e  10.1.1.1        ce:50:71:5b:8a:43
```

show overlay vxlan vtep detail—for all monitored VTEPs (no vtep-source-address specified)

```
user@host> show overlay vxlan vtep detail
Src VTEP IP-Addr: 10.0.1.101      Src VTEP Port: 1234  Interface: et-0/0/63.0

      Dest VTEP IP-Addr: 10.0.2.101      Dest VTEP Port: 4789  VxLAN ID (VNI): 100

      Application Flow Details:-
      Source VM IP: 10.1.1.1      Destination VM IP: 10.2.2.2
      Source VM MAC: ce:50:71:5b:8a:43  Destination VM MAC: 76:ca:b7:a2:58:4e
      Source VM Port: 0      Destination VM Port: 0
      Proto: 0

Src VTEP IP-Addr: 10.0.2.101      Src VTEP Port: 1234  Interface: et-0/0/2.0

      Dest VTEP IP-Addr: 10.0.1.101      Dest VTEP Port: 4789  VxLAN ID (VNI): 200

      Application Flow Details:-
      Source VM IP: 10.2.2.2      Destination VM IP: 10.1.1.1
      Source VM MAC: 76:ca:b7:a2:58:4e  Destination VM MAC: ce:50:71:5b:8a:43
      Source VM Port: 0      Destination VM Port: 0
```


Proto: 0

show overlay vxlan vtep <vtep-source-address> detail—detailed view for specified VTEP

```
user@host> show overlay vxlan vtep 10.0.1.101 detail
Src VTEP IP-Addr: 10.0.1.101      Src VTEP Port: 1234  Interface: et-0/0/63.0

Dest VTEP IP-Addr: 10.0.2.101      Dest VTEP Port: 4789  VxLAN ID (VNI): 100

Application Flow Details:-
Source VM IP: 10.1.1.1              Destination VM IP: 10.2.2.2
Source VM MAC: ce:50:71:5b:8a:43    Destination VM MAC: 76:ca:b7:a2:58:4e
Source VM Port: 0                   Destination VM Port: 0
Proto: 0
```

