Contrail Service Orchestration

Contrail Service Orchestration Installation and Upgrade Guide

Release
4.0.0

Modified: 2018-07-02

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

Contrail Service Orchestration Contrail Service Orchestration Installation and Upgrade Guide
4.0.0

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 https://www.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

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 https://www.juniper.net/documentation/.

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 https://www.juniper.net/books.

Documentation Conventions

Table 1 on page x 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 x 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**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	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** *domain-name*

Table 2: Text and Syntax Conventions (continued)

Convention	Description	Examples
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• 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 *metric*>;**
| (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** **(*string1* | *string2* | *string3*)**
# (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 [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  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.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (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 https://www.juniper.net/documentation/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 https://www.juniper.net/documentation/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 https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://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: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: https://www.juniper.net/cm/

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

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 https://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 https://www.juniper.net/support/requesting-support.html.

CHAPTER 1

Introduction

Contrail Service Orchestration Overview

Contrail Service Orchestration (CSO) is a software platform that designs, secures, automates and runs the entire service life cycle across NFX Series Network Services Platforms, MX Series Routers, and SRX Series Services Gateways, along with the vSRX Virtual Firewall and vMX Virtual Router, available in public cloud marketplaces.

The solution supports both Juniper Networks and third-party virtualized network functions (VNFs). CSO provides a RESTful APIs to connect with service providers' operational support systems (OSS) and business support systems (BSS) applications. It is responsible for many management and orchestration (MANO) activities in the deployment.

The following CSO components connect to Network Service Orchestrator through its RESTful API:

- Administration Portal: GUI to manage resources, customers, and availability of network services. It uses the RESTful APIs of other Contrail Service Orchestration components.
- Customer Portal: GUI to manage sites, customer premises equipment (CPE) devices, and network services for organizations.

The portals offer role-based access control (RBAC) for administrators and operators. Administrators can make changes; however, operators can only view the portal.

This guide provides information about installing and upgrading the Contrail Service Orchestration (CSO) Release 4.0.0 components. A full-version installer is available at CSO Download page.

From CSO Release 4.0.0 onward, you can install CSO by using the new installation GUI as well as through the CLI.

NOTE: The GUI Installation option is available by default for a fresh installation of CSO Release 4.0.0. You can, however, use the CLI, if you prefer.

CAUTION: **Currently, the upgrade process is not supported through the GUI.**

CSO 4.0.0 supports following types of deployments:

- Small Deployment: To manage approximately 450 devices. You cannot configure high availability with small deployments.
- Medium Deployment: To manage approximately 4000 devices. High availability is supported on medium deployments.
- Large Deployment: To manage approximately 5500 devices. High availability is support on large deployments.

For detailed information on configuring Contrail Service Orchestration, refer to the Contrail Service Orchestration Deployment Guide.

Related Documentation

- *Contrail Service Orchestration (CSO) Deployment Guide*

CHAPTER 2

Hardware and Software Requirements

Hardware and Software Required for Contrail Service Orchestration

Contrail Service Orchestration requires commercial off-the-shelf (COTS) node servers or servers, specific network devices, and specific software versions. The following sections list the hardware and software that are required and have been tested for the Cloud CPE and SD-WAN solutions.

Node Servers and Servers Tested in Contrail Service Orchestration

Use COTS node servers or servers for the following functions:

- Contrail Service Orchestration (CSO) central and regional servers
- Contrail Analytics servers
- Contrail controller and compute nodes in the centralized deployment

Table 3 on page 17 lists the node servers and servers that have been tested for these functions.

Table 3: COTS Node Servers and Servers Tested in the Cloud CPE and SD-WAN Solutions

Option	Vendor	Model	Type
1	QuantaPlex	T41S-2U 4-Node server	Multinode server accepting 4 nodes
2	Supermicro	SuperServer Model SYS-2028TPHC1TR-OTO-4	Multinode server accepting 4 nodes
3	Dell	PowerEdge R420 rack server	1U rack-mounted server

Table 4 on page 18 shows the software that has been tested for COTS servers used in the Cloud CPE solution. You must use these specific versions of the software when you implement the Cloud CPE and SD-WAN solutions.

Table 4: Software Tested for the COTS Nodes and Servers

Description	Version
Operating system for all COTS nodes and servers	Ubuntu 14.04.5 LTS NOTE: Ensure that you perform a fresh install of Ubuntu 14.04.5 LTS on the CSO servers in your deployment because upgrading from a previous version to Ubuntu 14.04.5 LTS might cause issues with the installation.
Operating system for VMs on CSO servers	• Ubuntu 14.04.5 LTS for VMs that you configure manually and not with the provisioning tool. • The provisioning tool installs Ubuntu 14.04.5 LTS in all VMs.
Hypervisor on CSO servers	KVM provided by the Ubuntu operating system on the server or VMware ESXi Version 5.5.0 onwards
Additional software for CSO servers	Secure File Transfer Protocol (SFTP)
Software defined networking (SDN) for a centralized deployment	Contrail Cloud Platform Release 3.2.5 with Heat v2 APIs
Contrail Analytics	Contrail Release 4.1.1.0-130

Network Devices and Software Tested in the Centralized Deployment

Table 5 on page 18 shows the network devices that have been tested for the centralized deployment.

Table 5: Network Devices Tested for the Centralized Deployment

Function	Device	Model	Quantity
SDN gateway router	Juniper Networks MX Series 3D Universal Edge Router	MX80-48T router with two 10-Gigabit Ethernet XFP optics	1
Management switch	Juniper Networks EX Series Ethernet Switch	EX3300-48T switch with: • 48 10/100/1000-Gigabit Ethernet interfaces • 4 built-in 10-Gigabit Ethernet SFP transceiver interfaces	1
Data switch	Juniper Networks QFX Series Switch	QFX 5100-48S-AFI switch with: • 48 SFP+ transceiver interfaces • 6 QSFP+ transceiver interfaces	1

Table 6 on page 19 shows the software tested for the centralized deployment. You must use these specific versions of the software when you implement a centralized deployment.

Table 6: Software Tested in the Centralized Deployment

Function	Software and Version
Operating system for MX Series router	Junos OS Release 14.2R3
Operating system for QFX Series switch	Junos OS Release 13.2X51-D38
Hypervisor on CSO servers	KVM provided by the Ubuntu operating system on the server or VMware ESXi Version 5.5.0
Element management system software	EMS microservice Junos Space Network Management Platform Release 15.1R1 (See *VNFs Supported by the Contrail Service Orchestration Solutions* for VNFs that require this product)
Software defined networking (SDN), including Contrail Analytics, for a centralized deployment	Contrail Release 3.2.5 with OpenStack Mitaka
Virtualized infrastructure manager (VIM) and virtual machine (VM) orchestration	OpenStack Mitaka
Authentication and Authorization	OpenStack Mitaka
Network Functions Virtualization (NFV)	CSO Release 4.0.0

Network Devices and Software Tested in the Hybrid WAN (Distributed CPE) and SD-WAN Deployments

Table 7 on page 19 shows the network devices that have been tested for the distributed deployment and the SD-WAN implementation.

Table 7: Network Devices Tested for the Distributed Deployment and SD-WAN Implementation

Function	Device	Model
PE router and IPsec concentrator (Hybrid WAN distributed deployment only)	Juniper Networks MX Series 3D Universal Edge Router	• MX960, MX480, or MX240 router with a Multiservices MPC line card • MX80 or MX104 router with Multiservices MIC line card • Other MX Series routers with a Multiservices MPC or Multiservices MIC line card See MPCs Supported by MX Series Routers and MICs Supported by MX Series Routers for information about MX Series routers that support Multiservices MPC and MIC line cards.

Table 7: Network Devices Tested for the Distributed Deployment and SD-WAN Implementation (continued)

Function	Device	Model
Cloud hub device (SD-WAN implementation only)	Juniper Networks MX Series 3D Universal Edge Router Juniper Networks SRX Series Services Gateway	• MX104, MX240, MX480, or MX960 router with an Multiservices MIC line card. See MPCs Supported by MX Series Routers and MICs Supported by MX Series Routers for information about MX Series routers that support Multiservices MPC and MIC line cards. • SRX1500 Services Gateway • SRX4100 Services Gateway • SRX4200 Services Gateway
On-premise hub device (SD-WAN implementation only)	Juniper Networks SRX Series Services Gateway vSRX on an x86 server	• SRX1500 Services Gateway • SRX4100 Services Gateway • SRX4200 Services Gateway • vSRX
CPE device (Hybrid WAN deployment) or spoke device (SD-WAN implementation)	• NFX Series Network Services Platforms • SRX Series Services Gateways • vSRX on an x86 server	• NFX250-LS1 device • NFX250-S1 device • NFX250-S2 device • NFX150-S1 • NFX150-S1E • NFX150-C-S1 • NFX150-C-S1-AE/AA • NFX150-C-S1E-AE/AA • SRX300 Services Gateway • SRX320 Services Gateway • SRX340 Services Gateway • SRX345 Services Gateway • vSRX

Table 8 on page 20 shows the software tested for the distributed deployment. You must use these specific versions of the software when you implement a distributed deployment and SD-WAN solution.

Table 8: Software Tested in the Distributed Deployment and SD-WAN Solution

Function	Software and Version
Hypervisor on CSO servers	KVM provided by the Ubuntu operating system on the server or VMware ESXi Version 5.5.0
Authentication and Authorization	OpenStack Mitaka
Network Functions Virtualization (NFV)	CSO Release 4.0.0.0
Contrail Analytics	Contrail Release 4.1.1.0-9

Table 8: Software Tested in the Distributed Deployment and SD-WAN Solution (continued)

Function	Software and Version
NFX Software	Junos OS Release 15.1X53-D490
Routing and Security for NFX250 device	vSRX KVM Appliance 15.1X49-D143
Operating system for vSRX used as a CPE device on an x86 server	vSRX KVM Appliance 15.1X49-D143
Operating system for SRX Series Services Gateway used as a CPE device or spoke device	Junos OS Release 15.1X49-D143
Operating system for MX Series router used as PE router	Junos OS Release 16.1R3.00
Operating system for MX Series router used as a hub device for an SD-WAN implementation	Junos OS Release 16.1R5.7
Operating system for SRX Series Services Gateway used as a hub device for an SD-WAN implementation	Junos OS Release 18.2R1

Related Documentation

- Minimum Requirements for Servers and VMs on page 21

Minimum Requirements for Servers and VMs

- Minimum Hardware Requirements for Node Servers and Servers on page 21
- Minimum Requirements for VMs on CSO Node Servers and Servers on page 23

Minimum Hardware Requirements for Node Servers and Servers

For information about the makes and models of node servers and servers that you can use in the Cloud CPE solution, see Table 4 on page 18. When you obtain node servers and servers for the Cloud CPE and SD-WAN solution, we recommend that you:

- Select hardware that was manufactured within the last year.
- Ensure that you have active support contracts for servers so that you can upgrade to the latest firmware and BIOS versions.

Table 9 on page 21 shows the specification for the node servers and servers for the Cloud CPE or SD-WAN solution.

Table 9: Specification for Node Servers and Servers

Item	Requirement
Storage	Greater than 750 GB of one of the following types: • Serial Advanced Technology Attachment (SATA) • Serial Attached SCSI (SAS) • Solid-state drive (SSD)

Table 9: Specification for Node Servers and Servers (continued)

Item	Requirement
CPU	One 64-bit dual processor, type Intel Sandybridge, such as Intel Xeon E5-2670v3 @ 2.4 Ghz or higher specification
Network interface	One 1-Gigabit Ethernet or 10-Gigabit Ethernet interface

The number of node servers and servers that you require depends on whether you are installing a small, medium, or large deployment.

Table 10 on page 22 shows the required hardware specifications for node servers and servers in the supported deployments. The server specifications are slightly higher than the sum of the virtual machine (VM) specifications listed in "Minimum Requirements for VMs on CSO Node Servers and Servers" on page 23, because some additional resources are required for the system software.

Table 10: Server Requirements

Function	Small Deployment	Medium Deployment	Large Deployment
Contrail Service Orchestration (CSO) Servers			
NOTE: If you install a small deployment with virtualized network functions (VNFs) that require Junos Space as the Element Management System (EMS), you must install Junos Space on a VM on another server. This server specification for a small deployment does not include Junos Space. For information about Junos Space VM requirements, see Table 11 on page 23.			
Number of nodes or servers	1	4	9 • 3 central servers • 3 regional servers
vCPUs per node or server	40	48	48
RAM per node or server	224 GB	256 GB	256 GB
Contrail Analytics Servers			
Number of servers	None—Contrail Analytics is in a VM	None—Contrail Analytics is in a VM	3
vCPUs per node or server	—	—	48
RAM per node or server	—	—	256 GB
Contrail Cloud Platform for a Centralized Deployment NOTE: These servers are not needed for Hybrid WAN or SD-WAN solutions			

Table 10: Server Requirements (continued)

Function	Small Deployment	Medium Deployment	Large Deployment
Number of nodes or servers	1	4–8 • 3 nodes for Contrail controller and analytics • 1–4 Contrail compute nodes	4–28 • 3 nodes for Contrail controller and analytics • 1–25 Contrail compute nodes
vCPUs per node or server	4	48	48
RAM per node or server	16 GB	256 GB	256 GB
Total Numbers of Servers			
Centralized deployment	2	7–11	10–34
Hybrid WAN or SD-WAN	1	3	9

Minimum Requirements for VMs on CSO Node Servers and Servers

The number of VMs needed and minimum requirements for CSO VMs depend on the deployment environment and whether or not you use high availability (HA):

- For a small deployment, see Table 11 on page 23.
- For a medium deployment, see Table 12 on page 24.
- For a large deployment, see Table 13 on page 26.

For information about the ports that must be open on VMs for all deployments, see Table 14 on page 30.

Use small deployments for managing approximately 450 sites. You cannot configure high availability with small deployments.

Table 11 on page 23 shows details about the VMs for a small deployment.

Table 11: Details of VMs for a Small Deployment

Name of VM	Components That Installer Places in VM	Resources Required
csp-installer-vm	—	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-infravm	Third-party applications used as infrastructure services	• 8 CPU • 48 GB RAM • 200 GB hard disk storage

Table 11: Details of VMs for a Small Deployment (continued)

Name of VM	Components That Installer Places in VM	Resources Required
csp-central-msvm	All microservices, including GUI applications	• 8 CPU • 48 GB RAM • 200 GB hard disk storage
csp-contrailanalytics-1	Contrail Analytics for a distributed deployment For a centralized or combined deployment, you use Contrail Analytics in the Contrail Cloud Platform.	• 8 vCPUs • 48 GB RAM • 500 GB hard disk storage
csp-regional-sblb	Load balancer for device to Fault Management Performance Management (FMPM) microservice connectivity	• 4 vCPUs • 8 GB RAM • 300 GB hard disk storage
csp-space-vm (optional)	Junos Space Virtual Appliance and database—required only if you deploy virtualized network functions (VNFs) that use this EMS	• 4 vCPUs • 16 GB RAM • 200 GB hard disk storage
csp-vrr-vm	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 200 GB hard disk storage
csp-central-k8mastervm	Central K8 Master VM	• 4 vCPUs • 8 GB RAM • 200 GB hard disk storage

Use medium deployments for approximately 3500 sites. High availability is supported on medium deployments.

Table 12 on page 24 shows details about the VMs for a medium deployment.

Table 12: Details of VMs for a Medium Deployment

Name of VM or Microservice Collection	Components That Installer Places in VM	Resources Required
csp-installer-vm	—	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-infravm1	Third-party applications used as infrastructure services	• 8 vCPUs • 48 GB RAM • 500 GB hard disk storage

Table 12: Details of VMs for a Medium Deployment (continued)

Name of VM or Microservice Collection	Components That Installer Places in VM	Resources Required
csp-central-infravm2	Third-party applications used as infrastructure services	• 8 vCPUs • 48 GB RAM • 500 GB hard disk storage
csp-central-infravm3	Third-party applications used as infrastructure services	• 8 vCPUs • 48 GB RAM • 500 GB hard disk storage
csp-central-lbvm1	Load-balancing applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-lbvm2	Load-balancing applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-lbvm3	Load-balancing applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-msvm1	All microservices, including GUI applications	• 16 CPUs • 64 GB RAM • 500 GB hard disk storage
csp-central-msvm2	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-central-msvm3	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-space-vm	Junos Space Virtual Appliance and database—required only if you deploy VNFs that use this EMS	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-elkvm1	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-elkvm2	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage

Table 12: Details of VMs for a Medium Deployment (continued)

Name of VM or Microservice Collection	Components That Installer Places in VM	Resources Required
csp-central-elkvm3	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-contrailanalytics-1	Contrail Analytics for a distributed deployment. For a centralized or combined deployment, you use Contrail Analytics in the Contrail Cloud Platform.	• 16 vCPUs • 64 GB RAM • 300 GB hard disk storage
csp-contrailanalytics-2	Contrail Analytics for a distributed deployment. For a centralized or combined deployment, you use Contrail Analytics in the Contrail Cloud Platform.	• 16 vCPUs • 64 GB RAM • 300 GB hard disk storage
csp-contrailanalytics-3	Contrail Analytics for a distributed deployment. For a centralized or combined deployment, you use Contrail Analytics in the Contrail Cloud Platform.	• 16 vCPUs • 64 GB RAM • 300 GB hard disk storage
csp-regional-sblb1	Load balancer for device to FMPM microservice connectivity	• 4 vCPUs • 16 GB RAM • 300 GB hard disk storage
csp-regional-sblb2	Load balancer for device to FMPM microservice connectivity	• 4 vCPUs • 16GB RAM • 300 GB hard disk storage
csp-vrr-vm	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 200 GB hard disk storage

Use large deployments for managing approximately 5000 sites. High availability is supported on medium deployments.

Table 13 on page 26 shows details about the VMs for a large deployment.

Table 13: Details of VMs for a Large Deployment

Name of VM or Microservice Collection	Components That Installer Places in VM	Resources Required
csp-installer-vm	—	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage

Table 13: Details of VMs for a Large Deployment (continued)

Name of VM or Microservice Collection	Components That Installer Places in VM	Resources Required
csp-central-infravm1	Third-party applications used as infrastructure services	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-central-infravm2	Third-party applications used as infrastructure services	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-central-infravm3	Third-party applications used as infrastructure services	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-central-lbvm1	Load-balancing applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-lbvm2	Load-balancing applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-lbvm3	Load-balancing applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-msvm1	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-central-msvm2	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-central-msvm3	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-regional-infravm1	Third-party applications used as infrastructure services	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-regional-infravm2	Third-party applications used as infrastructure services	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage

Table 13: Details of VMs for a Large Deployment (continued)

Name of VM or Microservice Collection	Components That Installer Places in VM	Resources Required
csp-regional-infravm3	Third-party applications used as infrastructure services	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-regional-msvm1	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-regional-msvm2	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-regional-msvm3	All microservices, including GUI applications	• 16 vCPUs • 64 GB RAM • 500 GB hard disk storage
csp-space-vm (optional)	Junos Space Virtual Appliance and database—required only if you deploy VNFs that use this EMS	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-elkvm1	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-elkvm2	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-central-elkvm3	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-regional-elkvm1	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-regional-elkvm2	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-regional-elkvm3	Logging applications	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage

Table 13: Details of VMs for a Large Deployment (continued)

Name of VM or Microservice Collection	Components That Installer Places in VM	Resources Required
csp-regional-sblb1	Load balancer for device to FMPM microservice connectivity	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-regional-sblb2	Load balancer for device to FMPM microservice connectivity	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-vrr-vm1	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-vrr-vm2	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-vrr-vm3	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-vrr-vm4	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-vrr-vm5	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-vrr-vm6	Virtual route reflector (VRR)	• 4 vCPUs • 32 GB RAM • 300 GB hard disk storage
csp-contrailanalytics-1	Contrail Analytics for a distributed deployment. For a centralized or combined deployment, you use Contrail Analytics in the Contrail Cloud Platform.	• 48 vCPUs • 256 GB RAM • 300 GB hard disk storage
csp-contrailanalytics-2	Contrail Analytics for a distributed deployment. For a centralized or combined deployment, you use Contrail Analytics in the Contrail Cloud Platform.	• 48 vCPUs • 256 GB RAM • 300 GB hard disk storage
csp-contrailanalytics-3	Contrail Analytics for a distributed deployment. For a centralized or combined deployment, you use Contrail Analytics in the Contrail Cloud Platform.	• 48 vCPUs • 256 GB RAM • 300 GB hard disk storage

Table 14 on page 30 shows the ports that must be open on all CSO VMs to enable the following types of CSO communications:

- External—CSO UI and CPE connectivity
- Internal—Between CSO components

The provisioning tool opens these ports on each VM; however, if you provision the VMs manually, you must manually open the ports on each VM.

Table 14: Ports to Open on CSO VMs

Port Number	CSO Communication Type	Port Function
22	External and internal	SSH
80	Internal	HAProxy
83	External	Network Service Designer
179	External	BGP for VRR
443	External and internal	HTTPS, including Administration Portal and Customer Portal
514	Internal	Syslog receiving port
1414	Internal	Cassandra Java Virtual Machine (JVM)
1936	External	HAProxy status page
1947	External	Icinga service
2181	Internal	ZooKeeper client
2379	Internal	etcd client communication
2380	Internal	etcd peer
2888	Internal	ZooKeeper follower
3000	External	Grafana
3306	Internal	MySQL
3514	External	Contrail Analytics Syslog receiving port
3888	Internal	ZooKeeper leader
4001	Internal	SkyDNS etcd discover
4505, 4506	Internal	Salt communications

Table 14: Ports to Open on CSO VMs (continued)

Port Number	CSO Communication Type	Port Function
5000	External	Keystone public
5044	Internal	Beats
5543	Internal	Logstash UDP
5601	External	Kibana UI
5665	Internal	Icinga API
5666	Internal	icinga nrpe
5671	Internal	RabbitMQ SSL listener
5672	Internal	RabbitMQ client
6000	Internal	Swift Object Server
6001	Internal	Swift Container Server
6002	Internal	Swift Account Server
6379	Internal	Redis
6543	Internal	Virtualized Network Function manager (VNFM)
7804	External	Device connectivity
8006	Internal	Network Service Orchestrator
8016	Internal	Notification engine
8080	Internal	cAdvisor
8082	Internal	Device Management Service (DMS) central
8083	Internal	Activation Service (AS) central
8085	Internal	DMS Schema
8086	Internal	Contrail Analytics
8090, 8091	Internal	Generic container
8529	Internal	ArangoDB
9042	Internal	Cassandra native transport

Table 14: Ports to Open on CSO VMs (continued)

Port Number	CSO Communication Type	Port Function
9090	Internal	Swift Proxy Server
9091	Internal	xmltec-xmlmail tcp
9101	External and internal	HA proxy exporter
9102	Internal	jetdirect
9160	Internal	Cassandra
9200	Internal	Elasticsearch
10248	Internal	kubelet healthz
15100	Internal	Logstash TCP
15672	Internal	RabbitMQ management
30000-32767	Internal	Kubernetes service node range
30900	External	Prometheus
35357	Internal	Keystone private

Related Documentation

- Hardware and Software Required for Contrail Service Orchestration on page 17
- Provisioning VMs on Contrail Service Orchestration Nodes or Servers on page 44

CHAPTER 3

Installing Contrail Service Orchestration with GUI

CSO GUI Installer Overview

The CSO installer walks you through the steps needed to install and configure CSO onto your virtual machine (VM.) It provides features such as faster downloading and installation, auto-provisioning of the CSO virtual machines, and a UI for interaction and installation status. The installer can be run on the following operating systems:

- Apple OSX and later.
- Microsoft Windows 10 and later.
- Ubuntu 14.04 and later.

NOTE: **Juniper Networks recommends using the CSO installer to install and configure CSO because of its ease-of-use and is less error-prone. You can, however, use the CLI if you prefer.**

The overall flow for using the CSO installer is as follows:

1. Download the installer from the Juniper Networks website.
2. Launch the installer and log in using your Juniper Networks credentials.
3. Enter setup information, such as server credentials and the solution to install.
4. The CSO installer downloads the required packages and creates an installer VM.
5. Enter configuration information, such as your topology and deployment size.

6. The CSO installer creates and deploys the necessary VMs.

7. Launch the CSO administration portal.

NOTE: Use the CSO installer only for new installations. You cannot currently upgrade from a previously installed version using the CSO installer.

The CSO installer has two main components that perform the steps listed above:

- Downloader—Downloads the CSO packages and creates an Installer Virtual Machine (IVM).
- Installer—Creates the necessary VMs and installs the CSO packages.

About the Downloader Component

The downloader component of the CSO installer performs the following tasks:

- Downloads the necessary CSO packages.
- Creates an IVM.
- Transfers the CSO packages to the IVM.
- Launches the Installer component web-based user interface from the IVM.

The downloader component gives you the option to install now or to install later. The install now option performs all the tasks listed above. The install later option downloads the necessary packages but does not create the IVM or transfer the packages to the IVM.

For example, if you do not know the host machine IP address and root password, you might choose to download the necessary packages now and wait to create the IVM until you have that information.

Or, you might choose install later to download the necessary packages and use the CLI to install CSO.

About the Installer Component

The installer component of the CSO installer starts automatically after the downloader component finishes, and performs the following tasks:

- Creates the required VMs.
- Installs the CSO packages in the VMs.

When running the installer component, you select the following options in addition to configuring the VMs. Each option is described in detail in the installer user interface.

- Size of the network to manage—small, medium, or large. The option you select determines the number of servers and the resource per server required. See Table 15 on page 35.

Table 15: Resources per Size

	Small	Medium	Large
Approximate number of managed sites	450	3500	5000
High availability	No	Yes	Yes
Servers	1	4	9
vCPUs/server	48	48	48
RAM/server	256 GB	256 GB	256 GB
Disk space/server	750 GB	750 GB	750 GB

- Express install or custom install. The express install uses pre-defined defaults and requires less user input. Select custom install if you want full control over the installation and configuration parameters.
- Network type—CSO reachable directly or CSO behind a NAT gateway.

  CSO reachable directly means you can access the managed devices and CSO without going through a NAT gateway. Here, the CSO and devices IP's are routable to each other within the enterprises private network. This topology is common in a campus environment where multiple locations are connected through VPNs as a single logical private network.

  For CSO behind a NAT gateway, the managed devices are typically remote devices not residing in the data center where CSO is installed. These devices reach CSO through a NAT gateway using a public IP address exposed for the data center. This topology is common when CSO manages customer's remote or on-premises devices. For example, the branch locations of a bank or restaurant chain.

Installing Contrail Service Orchestration with the GUI Installer

If you prefer to install CSO using the CLI, following the instructions below and select the **Install Later** option to download the CSO files to your local drive. Manually transfer the CSO files to your installer VM using scp, ftp, or other similar programs. Log in to your installer VM and follow the instructions described in "Installing and Configuring Contrail Service Orchestration" on page 77.

NOTE: Upgrading from a previously installed version only downloads the CSO packages. It does not perform the actual upgrade. You must use the CLI to upgrade CSO. For more information, see "Upgrading Contrail Service Orchestration Overview" on page 97.

To download and run the CSO installer:

1. From your browser, go to CSO download page.

2. On the page that appears, click the **Software** tab and select **4.0** from the Version drop-down menu.

3. Click the CSO downloader link corresponding to your operating system to download the file to your local drive.

4. Locate the file on your local drive and launch it.
   - For Windows, double-click the executable file.
   - For Linux, enter the following command:

     dpkg –i cso-downloader.deb
   - For Apple OS, drag the **.dmg** file to the installation window.
5. Enter your Juniper Networks support credentials and click **Next**.

6. Follow the instructions to specify the software setup, including:
   - Default download location on your local system.
   - CSO release version to install.
   - Hypervisor server type—KVM or ESXi.
   - New installation or upgrade from a previously installed version.
   - CSO solution to install—Contrail Service Orchestrator or CSO Network Service Controller.
7. Read and accept the license agreement, then click **Next**.

8. Select when to create the IVM and install the software.
   - Select **Install Now** if you already know the host hypervisor information and want to create the IVM now.
     a. Enter the hypervisor server IP address and root password, then click **Next**.

     b. Enter the IP address, the new root password, and the virtual bridge (or port group if using ESXi) for the IVM that is to be created.

        All CSO VMs will be attached to this virtual bridge.

NOTE: **Remember the password as you will need it later.**

c. (ESXi only) Select the datastore where the IVM is to be created.

d. Click **Next**.

- Select **Install Later** if you do not know the host hypervisor server IP address and root password, or want to use the CLI to install and configure CSO.

  If you select this option, files are downloaded but the IVM is not created. To continue at a later time, re-launch the application as shown in Step 4 above.

Additional files are downloaded as needed. Depending on your Internet bandwidth, it might take 30 minutes or more to download the files. After the download is complete, the CSO installer verifies the MD5 checksum of each file.

If you selected **Install Later**, the CSO installer stops here.

If you selected **Install Now**, the CSO installer creates the IVM and opens the installer component user interface in your default browser.

9. On the installer welcome page, enter the IVM password you created in Step 8 and click **Login**.

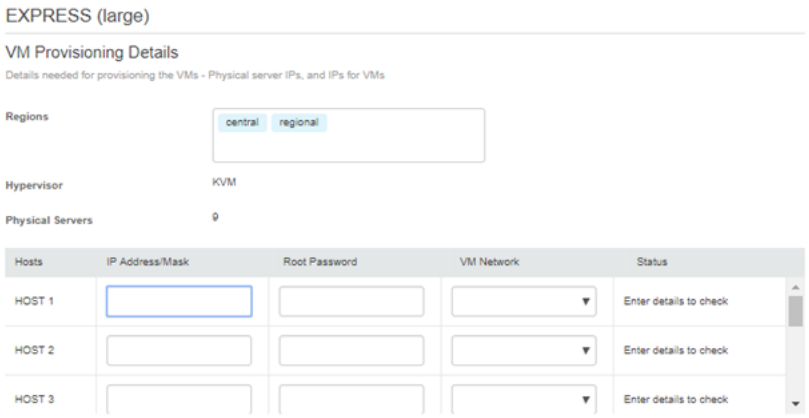
10. Identify the size of the network to be managed and click either **Express** or **Custom**.

    The express install uses predefined defaults and requires less user input. Whichever option you select, you can click **Back** on the next page to return to this page to select the other option.

11. If you selected the express install, the Express window appears.

    Figure 1 on page 37 shows an example of the Express window for a large install.

Figure 1: Example Express Install Window



For small and medium-sized managed networks, all hosts belong to a common CSO cluster or region.

For large networks, you must have a minimum of two regions. By default, a central region and regional region is configured for you. You can add an additional two more regions. Note that adding more regions requires more physical hosts.

a. For each host, enter the IP address with subnet mask, the root password, and select the VM network (or datastores for ESXi servers) from the drop-down menu.

b. Select the network type—**CSO Directly Reachable** or **CSO Behind NAT**. For more information, see "CSO GUI Installer Overview" on page 33.

c. Enter the IP addresses for the VMs.

- Click **Input IP Range** to add the IP addresses as a range.
- Click **Input IP** to add a list of individual IP addresses, separated with a comma.

d. (CSO Behind NAT only) Enter the central NAT gateway IP address.

This is the NAT gateway public-facing IP address.

e. Enter the regional NAT gateway IP address.

Each CSO region or cluster can have a different NAT gateway or the same NAT gateway.

f. Enter the NTP server IP address or FQDN name.

g. Verify the default Kubernetes overlay network IP address and subnet mask and update as needed.

h. Click **Install**.

The CSO installer now creates the required CSO VMs and installs services within these VMs. A status window displays the progress.

12. If you selected the custom install, the Custom install window appears.

a. For each host, enter the IP address with subnet mask, the root password, and select the VM network (or datastores for ESXi servers) from the drop-down menu.

b. Select the network type—**CSO Directly Reachable** or **CSO Behind NAT**. For more information, see "CSO GUI Installer Overview" on page 33.

c. Enter the IP addresses for the VMs.

- Click **Input IP Range** to add the IP addresses as a range.
- Click **Input IP** to add a list of individual IP addresses, separated with a comma.

d. (CSO Behind NAT only) Enter the central NAT gateway IP address.

This is the NAT gateway public-facing IP address.

e. Enter the regional NAT gateway IP address.

   Each CSO region or cluster can have a different NAT gateway or the same NAT gateway.

f. Enter the NTP server IP address or FQDN name.

g. Verify the default Kubernetes overlay network IP address and subnet mask and update as needed.

h. For each region, enter the virtual IP address and hostname. See Figure 2 on page 39.

Figure 2: Custom Install Virtual IP Address and Hostname

For small and medium-sized managed networks, all hosts belong to a common CSO cluster or region.

For large networks, you must have a minimum of two regions. By default, a central and regional region is configured for you. You can add an additional two more regions. Note that adding more regions requires more physical hosts.

For secure communication between devices to CSO services, digital certificates must be uploaded to CSO. These certificates are mapped to a hostname or an associated IP / Virtual IP (VIP) address. For example, two certificates can be uploaded for a single VM. One maps to its VIP and the other to its hostname. Typically, one hostname or VIP is required per region.In case of small and medium installs, there is no concept of different regions. All VMs are part of a common central region. Therefore, a single set of certificates is uploaded for this central region. In case of a large install, where the CSO VMs are replicated across multiple regions, one or more certificates for each region should be uploaded.

CSO generates certificates for VIPs and hostnames. However, these certificates are not signed by a trusted Certificate Authorities (CA). This results in “untrusted site” and “add security exception” warning messages to users. If you prefer to upload trusted CA signed certificates tied to a specific VIP or hostname, you can upload them to CSO (one or more for each region).

The regional VIP and hostname must be provided wherever high availability of services is available. For example, in medium and large installs. The small install does not provide high availability. The VIP address is used as a front-end address for a set of load-balancers of a region in the back-end. In addition, an FQDN hostname should be provided for this VIP address. For example,

cso-central.domain.net. Note that the uploaded signed certificate should be generated for this hostname.

i. Click **Choose Files** to locate the certificate for that specific region.

j. Review the default component configuration settings and update as needed.

k. If you select **External** from the Keystone Service menu, enter the following additional information:

- Keystone IP address.
- Keystone administrator password.
- External Keystone service token.

  If you do not know the service token, log in to your external keystone server. View the **/etc/keystone/keystonerc** file and search for the **ES_SERVICE_TOKEN** variable. For example:

  ```
  export OS_SERVICE_TOKEN=abcdefg1234567
  ```

- Keystone administrator e-mail address.

l. Click **Install**.

The CSO installer now creates the required CSO VMs and installs services within these VMs. A status window displays the progress.

Related Documentation

- Hardware and Software Required for Contrail Service Orchestration on page 17
- Removing a Previous Deployment on page 43

Troubleshooting the CSO GUI Installer-Related Errors

Use the following to troubleshoot software issues with the CSO GUI Installer.

Downloader Component

If you encounter any errors with the downloader component, forward the installer log file to Juniper Networks Technical Support. The installer log file is located at:

<*home*>/juniper/application.log

For example, **C:\Users\bob\juniper\application.log**.

Installer Component

The installer component screens validate user entries. For example, IP addresses are pinged to verify they are valid and reachable. However, errors can occur after you click **Install**. When this happens, you are presented the following two options:

- Retry—This option retries the operation that failed.
- View Logs—This option lets you download the installer component logs, which you can then send to Juniper Networks Technical Support for assistance.

CHAPTER 4

Installing Contrail Service Orchestration with CLI

Removing a Previous Deployment

You should remove a previous deployment and perform a new installation if the architecture of the VMs on the CSO server node or server has changed significantly between releases.

If you do not have previous deployment, proceed with "Provisioning VMs on Contrail Service Orchestration Nodes or Servers" on page 44

To remove a previous installation:

1. Remove VMs on the physical server.
   a. Log in to the CSO node or server as root.
   b. View the list of VMs.

      For example:

      ```
      root@host:~/# virsh list --all
      ```

      Output:

      ```
      Id  Name   State
      2  csp-ui-vm  running
      ```

c. Remove each VM and its contents.

For example:

```
root@host:~/# virsh destroy csp-ui-vm
root@host:~/# virsh undefine csp-ui-vm
```

Where, *csp-ui-vm* is the name of VM you want to delete.

d. Delete the Ubuntu source directories and VM.

For example:

```
root@host:~/# rm -rf /root/disks
root@host:~/# rm -rf /root/disks_can
root@host:~/# cd /root/ubuntu_vm
root@host:~/# rm -rf <vm-name>
```

2. Delete the Salt minion keys.

For example:

```
root@host:~/# salt-key -D
```

Related Documentation

- Provisioning VMs on Contrail Service Orchestration Nodes or Servers on page 44

Provisioning VMs on Contrail Service Orchestration Nodes or Servers

Virtual Machines (VMs) on the central and regional Contrail Service Orchestration (CSO) nodes or servers host the infrastructure services and some other components. All servers and VMs for the solution should be in the same subnet. To set up the VMs, you can:

- Use the provisioning tool to create and configure the VMs if you use the KVM hypervisor or VMware ESXi on a CSO node or server.

  The VMs created by provisioning tool have Ubuntu preinstalled.

  or

- Manually configure Virtual Route Reflector (VRR) VMs on a CSO node or server, if you use the VMware ESXi VM.

  The provisioning feature is not supported for upgrades.

NOTE: If you use the KVM hypervisor while installing a Distributed CPE (Hybrid WAN) or an SD-WAN solution, you must create a bridge interface on the physical server. The bridge interface should map the primary network interface (Ethernet management interface) on each CSO server node or server to a virtual interface before you create VMs. This action enables the VMs to communicate with the network.

This approach is applicable only if you are installing a Distributed CPE (Hybrid WAN) or an SD-WAN solution. It is not required for a centralized solution.

The VMs required on a CSO node or server depend on whether you configure:

- Small deployment. (See Table 11 on page 23)
- Medium deployment. See (Table 12 on page 24)
- Large deployment. See (Table 13 on page 26)

The small and medium deployments are always region-less deployment whereas the large deployment is always region-based deployment.

See "Minimum Requirements for Servers and VMs" on page 21 for details of the VMs and associated resources required for each deployment.

The following sections describe the procedures for provisioning the VMs:

- Before You Begin on page 45
- Downloading the Installer on page 46
- Creating a Bridge Interface for KVM on page 46
- Creating a Data Interface for a Distributed Deployment on page 48
- Customizing the Configuration File for the Provisioning Tool on page 50
- Provisioning VMs with the Provisioning Tool for the KVM Hypervisor on page 72
- Provisioning VMware ESXi VMs Using the Provisioning Tool on page 72
- Manually Provisioning VRR VMs on the Contrail Service Orchestration Node or Server on page 75
- Verifying Connectivity of the VMs on page 75

Before You Begin

Before you begin you must:

- Configure the physical servers or node servers and nodes.
- Install Ubuntu 14.04.5 LTS as the operating system for the physical servers.
- Configure the Contrail Cloud Platform and install Contrail OpenStack if you are performing a centralized CPE deployment.

Downloading the Installer

To download the installer package:

1. Log in as root to the central CSO node or server.

   The current directory is the home directory.

2. Download the appropriate installer package from the CSO Download page.

   - Use the Contrail Service Orchestration installer if you have purchased licenses for a centralized deployment or both Network Service Orchestrator and Network Service Controller licenses for a distributed deployment.

     This installer includes all the Contrail Service Orchestration graphical user interfaces (GUIs).

   - Use the Network Service Controller installer if you have purchased only Network Service Controller licenses for a distributed deployment or SD-WAN implementation.

     This installer includes Administration Portal and Service and Infrastructure Monitor, but not the Designer Tools.

3. Expand the installer package, which has a name specific to its contents and the release. For example, if the name of the installer package is **csoVersion.tar.gz**:

   ```
   root@host:~/# tar –xvzf csoVersion.tar.gz
   ```

   The expanded package is a directory that has the same name as the installer package and contains the installation files.

Creating a Bridge Interface for KVM

If you use the KVM hypervisor, you must create a bridge interface on the physical server that maps the primary network interface (Ethernet management interface) on each CSO server node or server to a virtual interface before you create VMs. This action enables the VMs to communicate with the network.

A physical server or node needs Internet access to install the **libvirt-bin** package.

To create the bridge interface:

1. Log in as root on the central CSO node or server.

2. Update the index files of the software packages installed on the server to reference the latest versions.

   ```
   root@host:~/# apt-get update
   ```

3. View the network interfaces configured on the server to obtain the name of the primary interface on the server.

   ```
   root@host:~/# ifconfig
   ```

4. Install the libvirt software.

   ```
   root@host:~/# apt-get install libvirt-bin
   ```

5. View the list of network interfaces, which now includes the virtual interface virbr0.

   ```
   root@host:~/# ifconfig
   ```

6. Open the **/etc/network/interfaces** file and modify it to map the primary network interface to the virtual interface virbr0.

   For example, use the following configuration to map the primary interface eth0 to the virtual interface virbr0:

   ```
   # This file describes the network interfaces available on your system
   # and how to activate them. For more information, see interfaces (5).
   # The loopback network interface
   auto lo
   iface lo inet loopback

   # The primary network interface
   auto eth0
   iface eth0 inet manual
       up ifconfig eth0 0.0.0.0 up

   auto virbr0
   iface virbr0 inet static
            bridge_ports eth0
            address 192.168.1.2
            netmask 255.255.255.0
            network 192.168.1.0
            broadcast 192.168.1.255
            gateway 192.168.1.1
            dns-nameservers 8.8.8.8
            dns-search example.net
   ```

7. Modify the default virtual network by customizing the file **default.xml**:

   a. Customize the IP address and subnet mask to match the values for the virbr0 interface in the **/etc/network/interfaces** file.

   b. Turn off the Spanning Tree Protocol (STP) option.

   c. Remove the NAT and DHCP configurations.

   For example:

   ```
   root@host:~/# virsh net-edit default
   ```

   Before modification:

   ```
   <network>
        <name>default</name>
        <uuid>0f04ffd0-a27c-4120-8873-854bbfb02074</uuid>
   ```

```
      <forward mode='nat'/>
      <bridge name='virbr0' stp='on' delay='0'/>
      <ip address='192.168.1.2' netmask='255.255.255.0'>
        <dhcp>
          <range start='192.168.1.1' end='192.168.1.254'/>
        </dhcp>
      </ip>
</network>
```

After modification:

```
<network>
     <name>default</name>
     <uuid>0f04ffd0-a27c-4120-8873-854bbfb02074</uuid>
     <bridge name='virbr0' stp='off' delay='0'/>
     <ip address='192.168.1.2' netmask='255.255.255.0'>
     </ip>
</network>
```

8. Reboot the physical machine and log in as root again.

9. Verify that the primary network interface is mapped to the virbr0 interface.

```
root@host:~/# brctl show

bridge name   bridge id       STP enabled interfaces
  virbr0      8000.0cc47a010808   no      em1
                                          vnet1
                                          vnet2
```

Creating a Data Interface for a Distributed Deployment

For a distributed deployment on KVM hypervisor, you create a second bridge interface that the VMs use to send data communications to the CPE device.

A physical server or node needs Internet access to install **libvirt-bin package**.

To create a data interface:

1. Log in to the central CSO server as root.

2. Configure the new virtual interface and map it to a physical interface.

   For example:

```
root@host:~/# virsh brctl addbr virbr1
root@host:~/# virsh brctl addif virbr1 eth1
```

3. Create a file with the name **virbr1.xml** in the **/var/lib/libvirt/network**directory.

4. Paste the following content into the **virbr1.xml** file, and edit the file to match the actual settings for your interface.

   For example:

```
 <network>
                <name>default</name>
                <uuid>0f04ffd0-a27c-4120-8873-854bbfb02074</uuid>
                <bridge name='virbr1' stp='off' delay='0'/>
                <ip address='192.0.2.1' netmask='255.255.255.0'>
                </ip>
</network>
```

5. Open the **/etc/network/interfaces** file and add the details for the second interface.

   For example:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces (5).
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet manual
  up ifconfig eth0 0.0.0.0 up

auto eth1
iface eth1 inet manual
  up ifconfig eth1 0.0.0.0 up

auto virbr0
iface virbr0 inet static
         bridge_ports eth0
         address 192.168.1.2
         netmask 255.255.255.0
         network 192.168.1.0
         broadcast 192.168.1.255
         gateway 192.168.1.1
         dns-nameservers 8.8.8.8
         dns-search example.net
auto virbr1
iface virbr1 inet static
    bridge_ports eth1
    address 192.0.2.1
    netmask 255.255.255.0
```

6. Reboot the server.

7. Verify that the secondary network interface, eth1, is mapped to the second interface.

```
root@host:~/# brctl show

bridge name   bridge id       STP enabled interfaces
  virbr0      8000.0cc47a010808   no      em1
                                          vnet1
                                          vnet2
  virbr1      8000.0cc47a010809   no      em2
                                          vnet0
```

Customizing the Configuration File for the Provisioning Tool

The provisioning tool uses a configuration file, which you must customize for your network. The configuration file is in YAML format.

To customize the configuration file:

1. Log in as root to the central CSO node or server.

2. Access the **confs** directory that contains the sample configuration files. For example, if the name of the installer directory is **csoVersion**.

   ```
   root@host:~/# cd csoVersion/confs
   ```

3. Access the directory for the environment that you want to configure.

   Table 16 on page 50 shows the directories that contain the sample configuration file.

Table 16: Location of Configuration Files for Provisioning VMs

Deployment	Directory for Sample Configuration File
Small deployment	**confs/cso4.0.0/trial/nonha/provisionvm/provision_vm_collocated_example.conf**
Medium deployment	**confs/cso4.0.0/production/ha/provisionvm/provision_vm_collocated_example.conf**
Large deployment	**confs/cso4.0.0/production/ha/provisionvm/provision_vm_example.conf**

4. Make a copy of the sample configuration file in the **/confs** directory and name it **provision_vm.conf**.

   For example:

   ```
   root@host:~/cspVersion/confs# cp
   /confs/cso4.0.0/trial/nonha/provisionvm/provision_vm_collocated_example.conf
   provision_vm.conf
   ```

5. Open the **provision_vm.conf** file with a text editor.

6. In the [TARGETS] section, specify the following values for the network on which CSO resides.

   - **installer_ip**—IP address of the management interface of the host on which you deployed the installer.
   - **ntp_servers**—Comma-separated list of fully qualified domain names (FQDNs) of Network Time Protocol (NTP) servers. For networks within firewalls, specify NTP servers specific to your network.

- **physical**—Comma-separated list of hostnames of the CSO nodes or servers.
- **virtual**—Comma-separated list of names of the virtual machines (VMs) on the CSO servers.

7. Specify the following configuration values for each CSO node or server that you specified in Step 6.
   - **[hostname]**—Hostname of the CSO node or server
   - **management_address**—IP address of the Ethernet management (primary) interface in classless Interdomain routing (CIDR) notation
   - **management_interface**—Name of the Ethernet management interface, virbr0
   - **gateway**—IP address of the gateway for the host
   - **dns_search**—Domain for DNS operations
   - **dns_servers**—Comma-separated list of DNS name servers, including DNS servers specific to your network
   - **hostname**—Hostname of the node
   - **username**—Username for logging in to the node
   - **password**—Password for logging in to the node
   - **data_interface**—Name of the data interface. Leave blank for a centralized deployment. Specify the name of the data interface, such as virbr1, that you configured for a distributed deployment.
8. Specify configuration values for each VM that you specified in Step 6.
   - **[VM name]**—Name of the VM
   - **management_address**—IP address of the Ethernet management interface in CIDR notation
   - **hostname**—Fully qualified domain name (FQDN) of the VM
   - **username**—Login name of user who can manage all VMs
   - **password**—Password for user who can manage all VMs
   - **local_user**—Login name of user who can manage this VM
   - **local_password**—Password for user who can manage this VM
   - **guest_os**—Name of the operating system
   - **host_server**—Hostname of the CSO node or server
   - **memory**—Required amount of RAM in GB
   - **vCPU**—Required number of virtual central processing units (vCPUs)
   - **enable_data_interface**—True enables the VM to transmit data and false prevents the VM from transmitting data. The default is false.
9. For the Junos Space VM, specify configuration values for each VM that you specified in Step 6.

- **[VM name]**—Name of the VM.
- **management_address**—IP address of the Ethernet management interface in CIDR notation.
- **web_address**—Virtual IP (VIP) address of the primary Junos Space Virtual Appliance. (Setting only required for the VM on which the primary Junos Space Virtual Space appliance resides.)
- **gateway**—IP address of the gateway for the host. If you do not specify a value, the value defaults to the gateway defined for the CSO node or server that hosts the VM.
- **nameserver_address**—IP address of the DNS nameserver.
- **hostname**—FQDN of the VM.
- **username**—Username for logging in to Junos Space.
- **password**—Default password for logging in to Junos Space.
- **newpassword**—Password that you provide when you configure the Junos Space appliance.
- **guest_os**—Name of the operating system.
- **host_server**—Hostname of the CSO node or server.
- **memory**—Required amount of RAM in GB.
- **vCPU**—Required number of virtual central processing units (vCPUs).
- **vm_type**—Preinstalled OSS with VM types of baseInfra/baseMS.
- **volumes**—Data partitions of CSO (Eg: /mnt/data:400G.
- **base_disk_size**—OS partitions of CSO (Eg: 100G).

10. Save the file.

11. Run the following command to start virtual machines.

```
root@host:~/# ./provision_vm.sh
```

The following sections show examples of customized configuration files for for a small, a medium, and a large deployment.

Sample Configuration File for Provisioning VMs in a Small Deployment

```
# This config file is used to provision KVM-based virtual machines using lib virt
 manager.

[TARGETS]
# Mention primary host (installer host) management_ip

installer_ip =

ntp_servers = ntp.juniper.net

# The physical server where the Virtual Machines should be provisioned
```

```
# There can be one or more physical servers on
# which virtual machines can be provisioned
physical = cso-central-host, cso-regional-host

# Note: Central and Regional physical servers are used as "csp-central-ms" and
"csp-regional-ms" servers.

# The list of servers to be provisioned and mention the contrail analytics servers
 also in "server" list.
server = csp-central-infravm,  csp-installer-vm, csp-space-vm,
csp-contrailanalytics-1, csp-central-elkvm, csp-central-msvm,  csp-vrr-vm,
csp-regional-sblb, csp-central-k8mastervm


# Physical Server Details
[cso-central-host]
management_address = 192.168.1.2/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-central-host
username = root
password = passw0rd
data_interface =

[cso-regional-host]
management_address = 192.168.1.3/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-regional-host
username = root
password = passw0rd
data_interface =


[csp-contrailanalytics-1]
management_address = 192.168.1.9/24
management_interface =
hostname = canvm.example.net
username = root
password = passw0rd
vm = false


# VM Details

[csp-central-infravm]
management_address = 192.168.1.4/24
hostname = centralinfravm.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host
memory = 65536
vcpu = 16
enable_data_interface = false
```

```
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-space-vm]
management_address = 192.168.1.6/24
web_address = 192.168.1.7/24
gateway = 192.168.1.1
nameserver_address = 192.168.1.254
hostname = spacevm.example.net
username = admin
password = abc123
newpassword = jnpr123!
guest_os = space
host_server = cso-regional-host
memory = 32768
vcpu = 4

[csp-installer-vm]
management_address = 192.168.1.8/24
hostname = installer.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host
memory = 16384
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-elkvm]
management_address = 192.168.1.10/24
hostname = centralelkvm.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-msvm]
management_address = 192.168.1.12/24
hostname = centralmsvm.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host
memory = 65536
vcpu = 16
```

```
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-sblb]
management_address = 192.168.1.14/24
hostname = regional-sblb.example.net
username = root
password = passw0rd
local_user = sblb
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host
memory = 4096
vcpu = 4
enable_data_interface = true
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-vrr-vm]
management_address = 192.168.1.15/24
hostname = vrr.example.net
gateway = 192.168.1.1
newpassword = passw0rd
guest_os = vrr
host_server = cso-regional-host
memory = 8192
vcpu = 4


[csp-central-k8mastervm]
management_address = 192.168.1.14/24
hostname = centralk8mastervm.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host
memory = 8192
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G
```

Sample Configuration File for Provisioning VMs in a Medium Deployment

```
# This config file is used to provision KVM-based virtual machines using lib virt
 manager.

[TARGETS]
# Mention primary host (installer host) management_ip

installer_ip =

ntp_servers = ntp.juniper.net
```

```
# The physical server where the Virtual Machines should be provisioned
# There can be one or more physical servers on
# which virtual machines can be provisioned
physical = cso-central-host1, cso-central-host2, cso-central-host3,
cso-regional-host1, cso-regional-host2, cso-regional-host3


# The list of servers to be provisioned and mention the contrail analytics servers
 also in "server" list.
server = csp-central-infravm1, csp-central-infravm2, csp-central-infravm3,
csp-central-lbvm1, csp-central-lbvm2, csp-central-lbvm3, csp-space-vm,
csp-installer-vm, csp-contrailanalytics-1, csp-contrailanalytics-2,
csp-contrailanalytics-3, csp-central-elkvm1, csp-central-elkvm2,
csp-central-elkvm3, csp-central-msvm1, csp-central-msvm2, csp-central-msvm3,
csp-vrr-vm, csp-regional-sblb1, csp-regional-sblb2

# Physical Server Details
[cso-central-host1]
management_address = 192.168.1.2/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-central-host1
username = root
password = passw0rd
data_interface =

[cso-central-host2]
management_address = 192.168.1.3/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-central-host2
username = root
password = passw0rd
data_interface =

[cso-central-host3]
management_address = 192.168.1.4/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-central-host3
username = root
password = passw0rd
data_interface =


[cso-regional-host1]
management_address = 192.168.1.5/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-regional-host1
username = root
password = passw0rd
data_interface =
```

```
[cso-regional-host2]
management_address = 192.168.1.6/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-regional-host2
username = root
password = passw0rd
data_interface =

[cso-regional-host3]
management_address = 192.168.1.7/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-regional-host3
username = root
password = passw0rd
data_interface =


[csp-contrailanalytics-1]
management_address = 192.168.1.17/24
management_interface =
hostname = can1.example.net
username = root
password = passw0rd
vm = false

[csp-contrailanalytics-2]
management_address = 192.168.1.18/24
management_interface =
hostname = can2.example.net
username = root
password = passw0rd
vm = false

[csp-contrailanalytics-3]
management_address = 192.168.1.19/24
management_interface =
hostname = can3.example.net
username = root
password = passw0rd
vm = false

# VM Details

[csp-central-infravm1]
management_address = 192.168.1.8/24
hostname = centralinfravm1.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 65536
vcpu = 16
```



```
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-infravm2]
management_address = 192.168.1.9/24
hostname = centralinfravm2.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-infravm3]
management_address = 192.168.1.10/24
hostname = centralinfravm3.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-space-vm]
management_address = 192.168.1.14/24
web_address = 192.168.1.15/24
gateway = 192.168.1.1
nameserver_address = 192.168.1.254
hostname = spacevm.example.net
username = admin
password = abc123
newpassword = jnpr123!
guest_os = space
host_server = cso-central-host2
memory = 32768
vcpu = 4

[csp-installer-vm]
management_address = 192.168.1.16/24
hostname = installervm.example.net
username = root
password = passw0rd
local_user = installervm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 16384
```

```
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-lbvm1]
management_address = 192.168.1.20/24
hostname = centrallbvm1.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-lbvm2]
management_address = 192.168.1.21/24
hostname = centrallbvm2.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-lbvm3]
management_address = 192.168.1.22/24
hostname = centrallbvm3.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-elkvm1]
management_address = 192.168.1.26/24
hostname = centralelkvm1.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
```

```
guest_os = ubuntu
host_server = cso-central-host1
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-elkvm2]
management_address = 192.168.1.27/24
hostname = centralelkvm2.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-elkvm3]
management_address = 192.168.1.28/24
hostname = centralelkvm3.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-msvm1]
management_address = 192.168.1.32/24
hostname = centralmsvm1.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-msvm2]
management_address = 192.168.1.33/24
hostname = centralmsvm2.example.net
username = root
```

```
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-msvm3]
management_address = 192.168.1.34/24
hostname = centralmsvm3.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-sblb1]
management_address = 192.168.1.38/24
hostname = regional-sblb1.example.net
username = root
password = passw0rd
local_user = sblb
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host1
memory = 4096
vcpu = 4
enable_data_interface = true
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-sblb2]
management_address = 192.168.1.39/24
hostname = regional-sblb2.example.net
username = root
password = passw0rd
local_user = sblb
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host2
memory = 4096
vcpu = 4
enable_data_interface = true
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-vrr-vm]
```



```
management_address = 192.168.1.41/24
hostname = vrr.example.net
gateway = 192.168.1.1
newpassword = passw0rd
guest_os = vrr
host_server = cso-regional-host3
memory = 32768
vcpu = 4
```

Sample Configuration File for Provisioning VMs in a Large Deployment

```
# This config file is used to provision KVM-based virtual machines using lib virt
 manager.

[TARGETS]
# Mention primary host (installer host) management_ip

installer_ip =

ntp_servers = ntp.juniper.net

# The physical server where the Virtual Machines should be provisioned
# There can be one or more physical servers on
# which virtual machines can be provisioned
physical = cso-central-host1, cso-central-host2, cso-central-host3,
cso-regional-host1, cso-regional-host2, cso-regional-host3


# The list of servers to be provisioned and mention the contrail analytics servers
 also in "server" list.
server = csp-central-infravm1, csp-central-infravm2, csp-central-infravm3,
csp-regional-infravm1, csp-regional-infravm2, csp-regional-infravm3,
csp-central-lbvm1, csp-central-lbvm2, csp-central-lbvm3, csp-regional-lbvm1,
csp-regional-lbvm2, csp-regional-lbvm3, csp-space-vm, csp-installer-vm,
csp-contrailanalytics-1, csp-contrailanalytics-2, csp-contrailanalytics-3,
csp-central-elkvm1, csp-central-elkvm2, csp-central-elkvm3, csp-regional-elkvm1,
 csp-regional-elkvm2, csp-regional-elkvm3, csp-central-msvm1, csp-central-msvm2,
 csp-central-msvm3, csp-regional-msvm1, csp-regional-msvm2, csp-regional-msvm3,
csp-vrr-vm, csp-regional-sblb1, csp-regional-sblb2

# Physical Server Details
[cso-central-host1]
management_address = 192.168.1.2/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-central-host1
username = root
password = passw0rd
data_interface =

[cso-central-host2]
management_address = 192.168.1.3/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-central-host2
username = root
password = passw0rd
```

```
data_interface =

[cso-central-host3]
management_address = 192.168.1.4/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-central-host3
username = root
password = passw0rd
data_interface =


[cso-regional-host1]
management_address = 192.168.1.5/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-regional-host1
username = root
password = passw0rd
data_interface =


[cso-regional-host2]
management_address = 192.168.1.6/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-regional-host2
username = root
password = passw0rd
data_interface =

[cso-regional-host3]
management_address = 192.168.1.7/24
management_interface = virbr0
gateway = 192.168.1.1
dns_search = example.net
dns_servers = 192.168.10.1
hostname = cso-regional-host3
username = root
password = passw0rd
data_interface =


[csp-contrailanalytics-1]
management_address = 192.168.1.17/24
management_interface =
hostname = can1.example.net
username = root
password = passw0rd
vm = false

[csp-contrailanalytics-2]
management_address = 192.168.1.18/24
management_interface =
hostname = can2.example.net
```



```
username = root
password = passw0rd
vm = false

[csp-contrailanalytics-3]
management_address = 192.168.1.19/24
management_interface =
hostname = can3.example.net
username = root
password = passw0rd
vm = false

# VM Details

[csp-central-infravm1]
management_address = 192.168.1.8/24
hostname = centralinfravm1.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-infravm2]
management_address = 192.168.1.9/24
hostname = centralinfravm2.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G


[csp-central-infravm3]
management_address = 192.168.1.10/24
hostname = centralinfravm3.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
```

```
base_disk_size = 500G

[csp-regional-infravm1]
management_address = 192.168.1.11/24
hostname = regionalinfravm1.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host1
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-infravm2]
management_address = 192.168.1.12/24
hostname = regionalinfravm2.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host2
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-infravm3]
management_address = 192.168.1.13/24
hostname = regionalinfravm3.example.net
username = root
password = passw0rd
local_user = infravm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host3
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseInfra
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-space-vm]
management_address = 192.168.1.14/24
web_address = 192.168.1.15/24
gateway = 192.168.1.1
nameserver_address = 192.168.1.254
hostname = spacevm.example.net
username = admin
password = abc123
newpassword = jnpr123!
guest_os = space
host_server = cso-central-host2
```



```
memory = 32768
vcpu = 4

[csp-installer-vm]
management_address = 192.168.1.16/24
hostname = installervm.example.net
username = root
password = passw0rd
local_user = installervm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 16384
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G


[csp-central-lbvm1]
management_address = 192.168.1.20/24
hostname = centrallbvm1.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-lbvm2]
management_address = 192.168.1.21/24
hostname = centrallbvm2.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-lbvm3]
management_address = 192.168.1.22/24
hostname = centrallbvm3.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
```

```
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-lbvm1]
management_address = 192.168.1.23/24
hostname = regionallbvm1.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host1
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-lbvm2]
management_address = 192.168.1.24/24
hostname = regionallbvm2.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host2
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-lbvm3]
management_address = 192.168.1.25/24
hostname = regionallbvm3.example.net
username = root
password = passw0rd
local_user = lbvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host3
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-elkvm1]
management_address = 192.168.1.26/24
hostname = centralelkvm1.example.net
username = root
password = passw0rd
local_user = elkvm
```



```
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-elkvm2]
management_address = 192.168.1.27/24
hostname = centralelkvm2.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-elkvm3]
management_address = 192.168.1.28/24
hostname = centralelkvm3.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-elkvm1]
management_address = 192.168.1.29/24
hostname = regionalelkvm1.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host1
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-elkvm2]
management_address = 192.168.1.30/24
hostname = regionalelkvm2.example.net
```

```
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host2
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-elkvm3]
management_address = 192.168.1.31/24
hostname = regionalelkvm3.example.net
username = root
password = passw0rd
local_user = elkvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host3
memory = 32768
vcpu = 4
enable_data_interface = false
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-msvm1]
management_address = 192.168.1.32/24
hostname = centralmsvm1.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host1
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-central-msvm2]
management_address = 192.168.1.33/24
hostname = centralmsvm2.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host2
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G
```

```
[csp-central-msvm3]
management_address = 192.168.1.34/24
hostname = centralmsvm3.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-central-host3
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-msvm1]
management_address = 192.168.1.35/24
hostname = regionalmsvm1.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host1
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-msvm2]
management_address = 192.168.1.36/24
hostname = regionalmsvm2.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host2
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-msvm3]
management_address = 192.168.1.37/24
hostname = regionalmsvm3.example.net
username = root
password = passw0rd
local_user = msvm
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host3
memory = 65536
vcpu = 16
enable_data_interface = false
vm_type = baseMS
```

```
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-regional-sblb1]
management_address = 192.168.1.38/24
hostname = regional-sblb1.example.net
username = root
password = passw0rd
local_user = sblb
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host1
memory = 4096
vcpu = 4
enable_data_interface = true
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G



[csp-regional-sblb2]
management_address = 192.168.1.39/24
hostname = regional-sblb2.example.net
username = root
password = passw0rd
local_user = sblb
local_password = passw0rd
guest_os = ubuntu
host_server = cso-regional-host2
memory = 4096
vcpu = 4
enable_data_interface = true
vm_type =
volumes = swap:64G,/data1:1G,/data2:1G
base_disk_size = 500G

[csp-vrr-vm]
management_address = 192.168.1.41/24
hostname = vrr.example.net
gateway = 192.168.1.1
newpassword = passw0rd
guest_os = vrr
host_server = cso-regional-host3
memory = 32768
vcpu = 4
```

Provisioning VMs with the Provisioning Tool for the KVM Hypervisor

If you use the KVM hypervisor for the CSO node or server, you can use the provisioning tool to:

- Create and configure the VMs for the CSO and Junos Space components.

The VMs provisioned by the tool will have pre-installed Ubuntu and Junos Space Network Management Platform software in the Junos Space VM

To provision VMs with the provisioning tool:

1. Log in as root to the central CSO node or server.

2. Access the directory for the installer. For example, if the name of the installer directory is **csoVersion**:

   ```
   root@host:~/# cd /~/csoVersion/
   ```

3. Run the provisioning tool.

   ```
   root@host:~/cspVersion/# ./provision_vm.sh
   ```

   The provisioning begins.

4. During installation, observe detailed messages in the log files about the provisioning of the VMs.

   - **provision_vm.log**—Contains details about the provisioning process
   - **provision_vm_console.log**—Contains details about the VMs
   - **provision_vm_error.log**—Contains details about errors that occur during provisioning

   For example:

   ```
   root@host:~/cspVersion/# cd logs
   root@host:/cspVersion/logs/# tail -f LOGNAME
   ```

Provisioning VMware ESXi VMs Using the Provisioning Tool

If you use the VMware ESXi (Version 6.0) VMs on the CSO node or server, you can use the provisioning tool—that is, provision_vm_ESXI.sh—to create and configure VMs for CSO.

NOTE: You cannot provision a Virtual Route Reflector (VRR) VM by using the provisioning tool. You must provision the VRR VM manually.

Before you begin, ensure that the maximum supported file size for a datastore in a VMware ESXi is greater than 512 MB. To view the maximum supported file size in datastore, you

can establish an SSH session with the ESXi host and run the **vmfkstools -P *datastorePath*** command.

To provision VMware ESXi VMs using the provisioning tool:

1. Download the CSO Release 4.0.0 installer package from the Software Downloads page to the local drive.

2. Log in as root to the Ubuntu VM with Internet access and a kernel version 4.4.0-31-generic. The VM must have the following specifications:

   - 8 GB RAM
   - 2 vCPUs

3. Copy the installer package from your local drive to the VM.

   ```
   root@host:~/# scp Contrail_Service_Orchestration_4.0.0.tar.gz root@VM :/root
   ```

4. On the VM, extract the installer package.

   For example, if the name of the installer package is Contrail_Service_Orchestration_4.0.0.tar.gz,

   ```
   root@host:~/# tar –xvzf Contrail_Service_Orchestration_4.0.0.tar.gz:
   ```

   The contents of the installer package are extracted in a directory with the same name as the installer package.

5. Navigate to the **confs** directory in the VM.

   For example:

   ```
   root@host:~/# cd Contrail_Service_Orchestration_4.0.0/confs
   root@host:~/Contrail_Service_Orchestration_4.0.0/confs#
   ```

6. Make a copy of the sample configuration file, provision_vm_example_ESXI.conf, that is available in the **confs** directory and rename it provision_vmI.conf.

   For example:

   ```
   root@host:~/Contrail_Service_Orchestration_4.0/confs# cp
   /confs/cso4.0.0/production/nonha/provisionvm/provision_vm_collocated_ESXI.conf
   provision_vm.conf
   ```

7. Open the **provision_vm.conf** file with a text editor.

8. In the [TARGETS] section, specify the following values for the network on which CSO resides.

   - **installer_ip**—IP address of the management interface of the VM on which you are running the provisioning script.

- **ntp_servers**—Comma-separated list of fully qualified domain names (FQDNs) of Network Time Protocol (NTP) servers. For networks within firewalls, specify NTP servers specific to your network.

You need not edit the following values:

- **physical**—Comma-separated list of hostnames of the CSO nodes or servers are displayed.
- **virtual**—Comma-separated list of names of the virtual machines (VMs) on the CSO servers are displayed.

9. Specify the following configuration values for each ESXi host on the CSO node or server.
   - **management_address**—IP address of the Ethernet management (primary) interface in Classless Interdomain Routing (CIDR) notation of the VM network. For example, 192.0.2.0/24.
   - **gateway**—Gateway IP address of the VM network
   - **dns_search**—Domain for DNS operations
   - **dns_servers**—Comma-separated list of DNS name servers, including DNS servers specific to your network
   - **hostname**—Hostname of the VMware ESXi host
   - **username**—Username for logging in to the VMware ESXi host
   - **password**—Password for logging in to the VMware ESXi host
   - **vmnetwork**—Label for each virtual network adapter. This label is used to identify the physical network that is associated to a virtual network adapter.

     The **vmnetwork** data for each VM is available in the Summary tab of a VM in the vSphere Client. You must not specify **vmnetwork** data within double quotation marks..
   - **datastore**—Datastore value to save all VM files.

     The **datastore** data for each VM is available in the Summary tab of a VM in the vSphere Client. You must not specify **datastore** data within double quotes.
10. Save the **provision_vm.conf** file.
11. Run the **provision_vm_ESXI.sh** script to create the VMs.

    ```
    root@host:~/Contrail_Service_Orchestration_4.0.0/# ./provision_vm_ESXI.sh
    ```
12. Copy the **provision_vm.conf** file to the installer VM.

    For example:

    ```
    root@host:~/Contrail_Service_Orchestration_4.0.0/#scp confs/provision_vm.conf
    root@installer_VM_IP:/root/Contrail_Service_Orchestration_4.0.0/confs
    ```

This action of provisioning VMs using the Provisioning Tool brings up VMware ESXi VMs with the configuration provided in the files.

Manually Provisioning VRR VMs on the Contrail Service Orchestration Node or Server

You cannot use the provision tool—**provision_vm_ESXI.sh**—to provision the Virtual Route Reflector (VRR) VM. You must manually provision the VRR VM.

To manually provision the VRR VM:

1. Download the VRR Release 15.1R6.7 software package (**.ova** format) for VMware from the Virtual Route Reflector page, to a location accessible to the server.
2. Launch the VRR by using vSphere or vCenter Client for your ESXi server and log in to the server with your credentials.
3. Set up an SSH session to the VRR VM.
4. Execute the following commands:

```
root@host:~/# configure
root@host:~/# delete groups global system services ssh root-login deny-password
root@host:~/# set system root-authentication plain-text-password
root@host:~/# New Password:<password>
root@host:~/# Retype New Password:<password>
root@host:~/# set system services ssh
root@host:~/# set system services netconf ssh
root@host:~/# set routing-options rib inet.3 static route 0.0.0.0/0 discard
root@host:~/# commit
root@host:~/# exit
```

Verifying Connectivity of the VMs

From each VM, verify that you can ping the IP addresses and hostnames of all the other servers, nodes, and VMs in the CSO.

CAUTION: If the VMs cannot communicate with all the other hosts in the deployment, the installation will fail.

Related Documentation

- Installing and Configuring Contrail Service Orchestration on page 77

Setting Up the Installation Package and Library Access

- Copying the Installer Package to the Installer VM on page 76
- Creating a Private Repository on an External Server on page 76

Copying the Installer Package to the Installer VM

After you have provisioned the VMs, copy the installer package from the central server to the installer VM.

1. Copy the installer package file from the central CSO server to the installer VM.
2. Log in to the installer VM as root.
3. Expand the installer package.

   For example, if the name of the installer package is **csoVersion.tar.gz**:

   ```
   root@host:~/# tar –xvzf csoVersion.tar.gz
   ```

   The contents of the installer package are placed in a directory with the same name as the installer package. In this example, the name of the directory is **csoVersion**.
4. If you have created an installer VM by using the provisioning tool, you must copy the **/csoVersion/confs/provision_vm.conf** file from the Ubuntu VM to the **/csoversion/confs/provision_vm.conf** directory of the installer VM.
5. Open the **provision_vm.conf** file with a text editor.
6. For **installer_ip** in the [TARGETS] section, specify the IP address of the installer VM.
7. Save the file.

Creating a Private Repository on an External Server

You use a private repository to download the libraries required for Contrail Service Orchestration. Use of a private repository for the libraries means that you do not require Internet access during the installation.

You can use a private repository either on the installer VM (the default choice) or on an external server.

- If you use the installer VM for the private repository, it is created when you install the solution, and you can skip this procedure.
- If you use an external server for the private repository, use the following procedure to create it.

To create the private repository on an external server:

1. Install the required Ubuntu release on the server that you use for the private repository.
2. Copy the installer package to the server.

3. Uncompress the installer package.

   For example, if the name of the installer package is **csoVersion.tar.gz**:

   ```
   root@host:~/# tar –xvzf csoVersion.tar.gz
   ```

   The contents of the installer package are placed in a directory with the same name as the installer package. In this example, the name of the directory is **csoVersion**.

4. Access the installer directory:

   For example:

   ```
   root@host:~/# cd csoVersion
   ```

5. Execute the **create_private_repo.sh** script to create the private repository.

   ```
   root@host:~/csoVersion#./create_private_repo.sh
   ```

   The script creates the private repository.

6. When you run the **setup_assist** script to create configuration files, specify that you use an external private repository. See "Installing and Configuring Contrail Service Orchestration" on page 77.

Related Documentation

- Provisioning VMs on Contrail Service Orchestration Nodes or Servers on page 44
- Installing and Configuring Contrail Service Orchestration on page 77

Installing and Configuring Contrail Service Orchestration

You use the same installation process for both Contrail Service Orchestration (CSO) and Network Service Controller and for both KVM and ESXi environments.

Before You Begin

Before you begin:

- Provision the virtual machines (VMs) for the CSO node or server. (See "Provisioning VMs on Contrail Service Orchestration Nodes or Servers" on page 44).
- Copy the installer package to the installer VM and expand it. (See "Setting Up the Installation Package and Library Access" on page 75)
- If you have created an installer VM using the provisioning tool, you must copy the **/Contrail_Service_Orchestration_4.0.0/confs/provision_vm.conf** file from the Ubuntu VM to the **/csoversion/confs/provision_vm.conf** directory of the installer VM.
- If you use an external server rather than the installer VM for the private repository that contains the libraries for the installation, create the repository on the server. (See "Setting Up the Installation Package and Library Access" on page 75).

  The installation process uses a private repository so that you do not need Internet access during the installation.
- Determine the following information:
  - The size of deployment: small, medium, large
  - For large deployment, 2 regions are configured by default - *central* and *regional*. You can add up to 2 additional regions with desired names. (e.g.- "Tokyo" or "East-Coast").
  - The IP address of the VM that hosts the installer.
  - The time zone for the servers in the deployment, based on the Ubuntu time zone guidelines.

    The default value for this setting is the current time zone of the installer host.
  - The fully qualified domain name (FQDN) of each Network Time Protocol (NTP) server that the solution uses. For networks within firewalls, use NTP servers specific to your network.

    For example: ntp.example.net
  - If you want to access the Administration Portal with the single sign-on method, enter the name of the public domain in which the CSO servers reside. Alternatively if you want to access the Administration Portal with local authentication, you need to enter a dummy domain name.
  - For a distributed deployment, whether you use transport layer security (TLS) to encrypt data that passes between the CPE device and CSO.

    You must use TLS unless you have an explicit reason for not encrypting data between the CPE device and CSO.
  - Whether you use the CSO Keystone or an external Keystone for authentication of CSO operations.
    - A CSO Keystone is installed with CSO and resides on the central CSO server.

      This default option is recommended for all deployments, and is required for a distributed deployment. Use of a CSO Keystone offers enhanced security because the Keystone is dedicated to CSO and is not shared with any other applications.

    - An external Keystone resides on a different server to the CSO server and is not installed with CSO.

      You specify the IP address and access details for the Keystone during the installation.

      - The Contrail OpenStack Keystone in the Contrail Cloud Platform for a centralized deployment is an example of an external Keystone.

        In this case, customers and Cloud CPE infrastructure components use the same Keystone token.

      - You can also use your own external Keystone that is not part of the CSO or Contrail OpenStack installation.

- If you use an external Keystone, the username and service token.
- The IP address of the Contrail controller node for a centralized deployment. For a centralized deployment, you specify this external server for Contrail Analytics.
- Whether you use a common password for all VMs or a different password for each VM, and the value of each password.
- The CIDR address of the subnet on which the CSO VMs reside.
- If you use NAT with your CSO installation, the public IP addresses used for NAT for the central and regional regions.
- The primary interface for all VMs.

  The default is eth0.
- The following information for each server and VM in the deployment:
  - Management IP address in CIDR notation

    For example: 192.0.2.1/24
  - FQDN of each host

    For example: central-infravm.example.net
  - Password for the root user

    If you use the same password for all the VMs, you can enter the password once. Otherwise, you must provide the password for each VM.
- For the microservices in the central and each regional region:
  - The IP address of the Kubernetes overlay network address in Classless Interdomain Routing (CIDR) notation.

    The default value is 172.16.0.0/16. If this value is close to your network range, use a similar address with a /16 subnet.
  - The range of the Kubernetes service overlay network addresses, in CIDR notation.

    The default value is 192.168.3.0/24.
  - The IP address of the Kubernetes service API server, which is on the service overlay network.

This IP address must be in the range you specify for the Kubernetes Service overlay network. The default value is 192.168.3.1.

- The IP address of the Kubernetes Cluster Domain Name System (DNS)

  This IP address must be in the range you specify for the Kubernetes Service overlay network. The default value is 192.168.3.1.

- The tunnel interface unit range that CSO uses for an SD-WAN implementation with an MX Series hub device.

  You must choose values that are different to those that you configured for the MX Series router. The possible range of values is 0–16,385, and the default range is 4000–6000.

- The FQDN that the load balancer uses to access the installation.
  - For the small deployment, the IP address and the FQDN of the VM that hosts the HAProxy.
  - For medium or large deployments, the virtual IP address and the associated hostname that you configure for the HAProxy.
- The replication factor for each microservice is pre-determined based on the size of the deployment.

Creating the Configuration Files

You use an interactive script to create configuration files for the environment topology. The installer uses these configuration files to customize the topology when you deploy the solution.

To run the installation tool:

1. Log in as root to the host on which you deployed the installer.

2. Access the directory for the installer. For example, if the name of the installer directory is **csoVersion**:

   ```
   root@host:~/# cd /~/csoVersion/
   ```

3. Run the setup tool:

   ```
   root@host:~/cspVersion/# ./setup_assist.sh
   ```

   The script starts, sets up the installer, and requests that you enter information about the installation.

4. Specify the management IP address of the VM that hosts the installer file.

5. Specify the deployment environment:
   - trial—Trial environment

- production—Production environment

6. Specify whether CSO is behind Network Address Translation (NAT).
   - y—CSO is behind NAT. After you deploy CSO, you must apply NAT rules. For information about NAT rules, see "Applying NAT Rules if CSO Is Deployed Behind NAT" on page 91.
   - n—CSO is not behind NAT (default)
7. Accept the default time zone or specify the Ubuntu time zone for the servers in the topology.
8. Specify a comma-separated list of FQDNs of NTP servers.

   For example: ntp.example.net, ntp.example.com
9. Specify whether the deployment uses high availability (HA).
   - y—Deployment uses HA
   - n—Deployment does not use HA
10. Specify whether the deployment has multiple regions
    - y—Deployment uses single region
    - n—Deployment uses multiple regions
11. Press enter if you use only one region or specify a comma-separated list or regions if you use multiple regions. You can configure a maximum of three regions. The default region is **regional**.
12. Specify the CSO certificate validity in days.

    The default value is 365 days.
13. For a distributed deployment, specify whether you use TLS to enable secure communication between the CPE device and CSO.

    Accept the default unless you have an explicit reason for not using encryption for communications between the CPE device and CSO.
    - n—Specifies that TLS is not used.
    - y—Specifies use of TLS. This is the default setting.
14. Specify the e-mail address of Admin User.
15. Specify a domain name to determine how you access the Administration Portal, the main CSO GUI:
    - If you want to access the Administration Portal with the single sign-on method, specify the name of the public domain in which the CSO servers reside.

      For example: *organization*.com, where *organization* is the name of your organization.

- If you want to use local authentication for the Administration portal, you specify a dummy name.

  For example: example.net

16. Specify whether you use an external Keystone to authenticate CSO operations, and if so, specify the OpenStack Keystone service token.
    - n—Specifies use of the CSO Keystone which is installed with and dedicated to CSO. We recommend that you use this default option unless you have a specific requirement for an external Keystone.
    - y—Specifies use of an external OpenStack Keystone, such as a Keystone specific to your network. Select the IP address and access details for the Keystone.
17. Specify whether you use an external Contrail Analytics server:
    - y—Specifies use of Contrail Analytics in Contrail OpenStack for a centralized or combined deployment.

      You must provide the IP address of the Contrail controller node.
    - n—Specifies use of the Contrail Analytics VM for a distributed deployment.
18. Specify whether you use a common password for all CSO VMs, and if so, specify the password.
19. Specify the following information for the virtual route reflector (VRR) that you create:
    a. Specify whether VRR is behind NAT.
       - Specify the number of VRR instances.
         - For non-HA deployments, you must create at least one VRR.
         - For HA deployments, we recommend that you create VRRs in even numbers, and you must create at least two VRRs. Each VRR must be in a different redundancy group. If the primary VRR fails or connectivity is lost, the session remains active as the secondary VRR continues to receive and advertise LAN routes to a site, thereby providing redundancy.
       - y—VRR is behind NAT. If you are deploying a VRR in a private network, the NAT instance translates all requests (BGP traffic) to a VRR from a public IP address to a private IP address.
       - n—VRR is not behind NAT (default).

    b. Specify whether you use a common password for all VRRs.
       - y—Specify the common password for all VRRs.
       - n—Specify the password for each VRR.

    c. Specify the public IP address for each VRR that you create. For example, 192.0.20.118/24.
    d. Specify the redundancy group for each VRR that you have created.
       - For non-HA deployments, specify the redundant group of the VRR as zero.

- For HA deployments, the VRRs must be distributed among the redundancy groups. There can be two groups—group 0 and group 1. For example, if you have two VRRs, specify the redundancy group for VRR1 as 0 and the VRR2 as 1.

20. Starting with the central region, specify the following information for each server in the deployment of each region.

    The script prompts you for each set of information that you must enter.

    - Management IP address with CIDR

      For example: 192.0.2.1/24

    - Password for the root user (only required if you use different passwords for each VM)
    - The IP address of the Kubernetes overlay network address, in CIDR notation, that the microservices use.

      The default value is 172.16.0.0/16. If this value is close to your network range, use a similar address with a /16 subnet.

    - The range of the Kubernetes service overlay network addresses, in CIDR notation.

      The default value is 192.168.3.0/24. It is unlikely that there will be a conflict between this default and your network, so you can usually accept the default. If, however, there is a conflict with your network, use a similar address with a /24 subnet.

    - The IP address of the Kubernetes service API server, which is on the service overlay network.

      This IP address must be in the range you specify for the Kubernetes Service overlay network. The default value is 192.168.3.1.

    - The IP address of the Kubernetes Cluster DNS server.

      This IP address must be in the range you specify for the Kubernetes Service overlay network. The default value is 192.168.3.1.

    - Specify the range of tunnel interface units that CSO uses for an SD-WAN implementation with an MX Series hub device

      The default setting is 4000–6000. You specify values in the range 0–16,385 that are different to those that you configured on the MX Series router.

    - The IP address and FQDN of the host for the load balancer:
      - For non-HA deployments, the IP address and FQDN of the VM that hosts the HAProxy.
      - For HA deployments, the virtual IP address and associated FQDN that you configure for the HAproxy.
    - The replication factor for each microservice is pre-determined based on the size of the deployment.

    The tool uses the input data to configure each region and indicates when the configuration stage is complete.

21. Configure settings for each region in the deployment:

    - Specify the IP address and prefix of the Kubernetes overlay network that the microservices use.
    - Specify the fully-qualified domain names of the host for the load balancer.
        - For a non-HA deployment, the IP address or FQDN of the VM that hosts the HAProxy
        - For an HA deployment, the virtual IP address that you configure for the HAProxy.
    - Specify a unique virtual router identifier in the range 0–255 for the HAProxy VM in each region.

      NOTE: **Use a different number for this setting in each region.**

    - Specify the number of instances of microservices:
        - For non-HA installations, specify 1.
        - For HA installations, specify 2.

    The tool uses the input data to configure each region and indicates when the configuration stage is complete.

22. Specify the subnet in CIDR notation on which the CSO VMs reside.

    The script requires this input, but uses the value only for distributed deployments and not for centralized deployments.

23. Specify the range for tunnel interface unit.

24. Accept or specify the primary interface for all VMs.

    The default is eth0. Accept this value unless you have explicitly changed the primary interface on your host of VMs.

25. When all regions are configured, the tool starts displaying the deployment commands.

    -

```
root@host:~/# DEPLOYMENT_ENV=central ./deploy_infra_services.sh
root@host:~/# DEPLOYMENT_ENV=regional ./deploy_infra_services.sh
root@host:~/# DEPLOYMENT_ENV=central ./deploy_micro_services.sh
root@host:~/# DEPLOYMENT_ENV=regional ./deploy_micro_services.sh
```

NOTE: The password for each infrastructure component and the Administration Portal password are displayed on the console after you complete answering the Setup Assistance questions. You must note the password that is displayed on the console as they are not saved in the system. To enhance the password security, the length and pattern for each password is different and the password is encrypted, and passwords in the log file are masked.

Deploying Infrastructure Services

To deploy infrastructure services:

1. Log in as root to the Installer VM.

2. Deploy the central infrastructure services.

```
root@host:~/# run "DEPLOYMENT_ENV=central ./deploy_infra_services.sh"
```

CAUTION: Wait at least ten minutes before executing the next command. Otherwise, the microservices might not be deployed correctly.

3. Deploy the regional infrastructure services and wait for the process to complete.

```
root@host:~/# run "DEPLOYMENT_ENV=regional ./deploy_infra_services.sh"
```

If you have configured multiple regions, then you can deploy the infrastructure services on the regions in any order after deploying the central infrastructure.

NOTE: The **deploy_infra_services.sh** script performs a health check of infrastructure services. If you encounter an error, you must rerun the **deploy_infra_services.sh** script.

Deploying Microservices

To deploy the microservices:

1. Log in as root to the Installer VM.

2. Deploy the central microservices.

```
root@host:~/# -run "DEPLOYMENT_ENV=central ./deploy_micro_services.sh"
```

CAUTION: Wait at least ten minutes before executing the next command. Otherwise, the microservices might not be deployed correctly.

3. Deploy the regional microservices and wait for the process to complete:

```
root@host:~/# -run "DEPLOYMENT_ENV=regional ./deploy_micro_services.sh"
```

Checking the Status of the Microservices

To check the status of the microservices:

1. Log in as root into the VM or server that hosts the central microservices.

2. Run the following command with required region – central or regional.

```
root@host:~/# kubectl get pods | grep -v Running –n <region>
```

If the result is an empty display, as shown below, the microservices are running and you can proceed to the next section.

```
root@host:~/# kubectl get pods | grep -v Running –n <region>
NAME                                READY   STATUS             RESTARTS  AGE
```

If the display contains an item with the status **CrashLoopBackOff** or **Terminating**, a microservice is not running.

3. Delete and restart the pod.

```
root@host:~/# kubectl get pods

 NAME                                READY   STATUS             RESTARTS   AGE
 csp.ams-3909406435-4yb0l            1/1     CrashLoopBackOff   0          8m
 csp.nso-core-3445362165-s55x8       0/1     Running            0          8m
```

The first item in the display shows the microservice and the second item shows its pod.

```
root@host:~/# kubectl delete pods -l microservice=csp.nso-core –n <region>
```

4. Wait a couple of minutes and then check the status of the microservice and its pod.

```
root@host:~/#kubectl get pods –n <region>

 NAME                                READY     STATUS      RESTARTS    AGE

 csp.ams-4890899323-3dfd02           1/1       Running     0           1m
 csp.nso-core-09009278633-fr234f     0/1       Running     0           1m
```

Loading Data

After you check that the microservices are running, you must load data to import plug-ins and data design tools.

To load data:

1. Ensure that all the microservices are up and running on the central and each regional microservices host.

2. (Optional) Specify the value of the regional subnet in the **/micro_services/data/inputs.yaml** file on the installer VM. By default, the subnet address is the management address of the regional microservices host that you specify in the **topology.conf** file.

3. Access the home directory of the installer VM.

4. Execute the **./load_services_data.sh** command.

   ```
   root@host:~/#./load_services_data.sh
   ```

NOTE: You must not execute load_services_data.sh more than once after a new deployment.

Performing a Health Check of Infrastructure Components

After you install or upgrade CSO, you can run the **components_health.sh** script to perform a health check of all infrastructure components. This script detects whether any infrastructure component has failed and displays the health status of the following infrastructure components:

- Cassandra
- Elasticsearch
- Etcd
- MariaDB
- RabbitMQ
- ZooKeeper
- Redis
- ArangoDb
- SimCluster
- ELK Logstash
- ELK Kibana
- Contrail Analytics
- Keystone
- Swift
- Kubernetes

To check the status of infrastructure components:

1. Log in to the installer VM as root.

2. Navigate to the CSO directory in the installer VM.

   For example:

   ```
   root@host:~/# cd Contrail_Service_Orchestration_4.0.0
   root@host:~/Contrail_Service_Orchestration_3.3#
   ```

3. Run the **components_health.sh** script.

   To check the status of infrastructure components of the central environment, run the following command:

   ```
   root@host:~/Contrail_Service_Orchestration_3.3#./components_health.sh central
   ```

   To check the health component of the regional environment, run the following command:

```
root@host:~/Contrail_Service_Orchestration_3.3#./components_health.sh regional
```

To check the health component of central and regional environments, run the following command:

```
root@host:~/Contrail_Service_Orchestration_3.3# ./components_health.sh
```

After a couple of minutes, the status of each infrastructure component for central and regional environments is displayed.

For example:

```
**************************************************************************

HEALTH CHECK FOR INFRASTRUCTURE COMPONENTS STARTED IN CENTRAL ENVIRONMENT

**************************************************************************

INFO     Health Check for Infrastructure Component Cassandra Started
INFO     The Infrastructure Component Cassandra is Healthy

INFO     Health Check for Infrastructure Component ElasticSearch Started
INFO     The Infrastructure Component ElasticSearch is Healthy

INFO     Health Check for Infrastructure Component Etcd Started
INFO     The Infrastructure Component Etcd is Healthy

INFO     Health Check for Infrastructure Component MariaDb Started
INFO     The Infrastructure Component MariaDb is Healthy

INFO     Health Check for Infrastructure Component RabbitMQ Started
INFO     The Infrastructure Component RabbitMQ is Healthy

INFO     Health Check for Infrastructure Component ZooKeeper Started
INFO     The Infrastructure Component ZooKeeper is Healthy

INFO     Health Check for Infrastructure Component Redis Started
INFO     The Infrastructure Component Redis is Healthy

INFO     Health Check for Infrastructure Component ArangoDb Started
INFO     The Infrastructure Component ArangoDb is Healthy

INFO     Health Check for Infrastructure Component Sim_Cluster Started
INFO     The Infrastructure Component Sim_Cluster is Healthy

INFO     Health Check for Infrastructure Component Elk_Logstash Started
INFO     The Infrastructure Component Elk_Logstash is Healthy

INFO     Health Check for Infrastructure Component Elk_Kibana Started
INFO     The Infrastructure Component Elk_Kibana is Healthy

INFO     Health Check for Infrastructure Component Keystone Started
INFO     The Infrastructure Component Keystone is Healthy

INFO     Health Check for Infrastructure Component Swift Started
INFO     The Infrastructure Component Swift is Healthy

INFO     Health Check for Infrastructure Component Kubernetes Started
INFO     The Infrastructure Component Kubernetes is Healthy
```

```
INFO     Health Check for Infrastructure Component Contrail_Analytics Started
INFO     The Infrastructure Component Contrail_Analytics is Healthy


Overall result:

        The following Infrastructure Components are Healthy:

                ['Cassandra', 'ElasticSearch', 'Etcd', 'MariaDb', 'RabbitMQ',
 'ZooKeeper', 'Redis', 'ArangoDb', 'Sim_Cluster', 'Elk_Logstash', 'Elk_Kibana',
 'Keystone', 'Swift', 'Kubernetes', 'Contrail_Analytics']
```

Related Documentation

- Provisioning VMs on Contrail Service Orchestration Nodes or Servers on page 44
- Generating and Encrypting Passwords for Infrastructure Components on page 90

Generating and Encrypting Passwords for Infrastructure Components

From Contrail Service Orchestration (CSO) Release 3.3 onwards, CSO uses an algorithm to automatically generate a dynamic password for the following infrastructure components:

- Cassandra
- Keystone
- MariaDB
- RabbitMQ
- Icinga
- Prometheus
- ArangoDB

The automatically generated passwords for each infrastructure component and the cspadmin password for Administration Portal are displayed on the console after you complete answering the Setup Assistance questions.

NOTE: **You must note the automatically generated password that is displayed on the console as they are not saved in the system.**

To enhance the password security, the length and pattern for each password is different and the password is encrypted. The passwords in the log file are masked.

Related Documentation

- Installing and Configuring Contrail Service Orchestration on page 77

Applying NAT Rules if CSO Is Deployed Behind NAT

If you have deployed Contrail Service Orchestration (CSO) behind NAT, you must apply NAT rules after you run the **setup_assit.sh** script on central and regional hosts. The NAT rule set determines the direction of the traffic to be processed.

NOTE: If you do not apply NAT rules after you install or upgrade CSO, you cannot access the Administration Portal, the Kibana UI, and the Rabbit MQ console.

To apply NAT rules:

1. Log in to the installer VM as root.

2. Apply NAT rules for central and regional NAT servers.
   - To quickly apply the NAT rules for central NAT servers:
     a. Copy the following commands and paste them into a text file.

```
iptables -t nat -A PREROUTING -p tcp -d
central-nat-server-public-ip-address--dport 443 -j DNAT --to-destination
northbound-virtual-private-ip-address:443
iptables -t nat -A PREROUTING -p tcp -d
central-nat-server-public-ip-address--dport 35357 -j DNAT --to-destination
northbound-virtual-private-ip-address:35357
iptables -t nat -A PREROUTING -p tcp -d
central-nat-server-public-ip-address--dport 5601 -j DNAT --to-destination
northbound-virtual-private-ip-address:5601
iptables -t nat -A PREROUTING -p tcp -d
central-nat-server-public-ip-address --dport 9200 -j DNAT --to-destination
northbound-virtual-private-ip-address:9200
iptables -t nat -A PREROUTING -p tcp -d
central-nat-server-public-ip-address --dport 1947 -j DNAT --to-destination
northbound-virtual-private-ip-address:1947
```

     b. You must specify IP addresses in the command to match your network configuration.

     c. Copy and paste the updated commands into the CLI.

   - To quickly apply the NAT rules for regional NAT servers:
     a. Copy the following commands and paste them into a text file.

```
iptables -t nat -A POSTROUTING -o virbr0 -p tcp --dport 5601 -d
northbound-virtual-private-ip-address -j SNAT --to-source
regional-management-interface-ip-address
iptables -t nat -A PREROUTING -p tcp -d
regional-nat-server-public-ip-address --dport 5601 -j DNAT --to-destination
northbound-virtual-private-ip-address:5601
iptables -t nat -A POSTROUTING -o virbr0 -p tcp --dport 7804 -d
northbound-virtual-private-ip-address -j SNAT --to-source
regional-management-interface-ip-address
```

```
iptables -t nat -A PREROUTING -p tcp -d
regional-nat-server-public-ip-address --dport 7804 -j DNAT --to-destination
northbound-virtual-private-ip-address:7804
iptables -t nat -A POSTROUTING -o virbr0 -p tcp --dport 3514 -d
southbound-virtual-private-ip-address -j SNAT --to-source
regional-management-interface-ip-address
iptables -t nat -A PREROUTING -p tcp -d
regional-nat-server-public-ip-address--dport 3514 -j DNAT --to-destination
southbound-virtual-private-ip-address:3514
iptables -t nat -A POSTROUTING -o virbr0 -p tcp --dport 514 -d
southbound-virtual-private-ip-address-j SNAT --to-source
regional-management-interface-ip-address
iptables -t nat -A PREROUTING -p tcp -d
regional-nat-server-public-ip-address --dport 514 -j DNAT --to-destination
southbound-virtual-private-ip-address:514
iptables -t nat -A POSTROUTING -o virbr0 -p tcp --dport 443 -d
southbound-virtual-private-ip-address -j SNAT --to-source
regional-management-interface-ip-address
iptables -t nat -A PREROUTING -p tcp -d
regional-nat-server-public-ip-address --dport 443 -j DNAT --to-destination
northbound-virtual-private-ip-address:443
iptables -t nat -A PREROUTING -d regional-nat-server-public-ip-address/32
-p tcp -m tcp --dport 2216 -j DNAT --to-destination
southbound-virtual-private-ip-address:2216
iptables -t nat -A POSTROUTING -d southbound-virtual-private-ip-address/32
-o virbr0 -p tcp -m tcp --dport 2216 -j SNAT --to-source
regional-management-interface-ip-address
```

b. You must specify IP addresses in the commands to match your network configuration.

c. Copy and paste the updated commands into the CLI.

The NAT rules are applied for central and regional NAT servers, and you can access Administration Portal, Kibana UI, and Rabbit MQ console.

Related Documentation

- Installing and Configuring Contrail Service Orchestration on page 77

Applying Security Patches

You can apply in-service patches to CSO microservices without having to reboot.

This feature is applicable only to microservices and is not supported for infrastructure components like Cassandra, RabbitMQ,.../kernel etc. The process does not impact sites or CSO workflows.

You can always revert the applied patch in case of unsuccessful execution.

You can follow the following steps to apply security patches:

- Download the tar file that contains the hotfix.
- Run patch script - **patch.sh** to apply security patches.

  The script is bundled in the tar file which needs to be executed on the installer VM

The script performs in-service patching of CSO microservices.

Related Documentation

- *Contrail Service Orchestration Monitoring and Troubleshooting Guide*
- Viewing Information About Microservices on page 93

Viewing Information About Microservices

When you log into Kibana, you see the Discover page, which displays a chart of the number of logs for a specific time period and a list of events for the deployment. You can filter this data to view subsets of logs and add fields to the table to find the specific information that you need. You can also change the time period for which you view events.

Table 17 on page 93 provides basic functions of each microservice. The list is limited to some of the external facing microservices.

Table 17: Functions of Microservices

Microservice	Description
Activation Service (Central)	Provides network activation functions to enable zero touch provisioning of devices.
ams	Monitors and autonomously collects data without system or human intervention.
cslm	Maintains EMS device data model for device management functions. The data model contains information like device objects, abstract configuration, device inventory object, configuration template object, device profile object, device image object etc.
Configuration Template Service	Provides configuration template management features for the CSO solution. The features include maintaining a database of config templates, template syntanx validation (e.g jinja2, python, yang rpc), template execution with input parameters using Yang RPC, and input/output validation (provided corresponding schema is given).
Device Management Service (Central)	• Manages the lifecycle of device objects. Each device object provides an abstraction for one or more physical or virtual network devices. • Provides APIs for device management.
Dataview Service (Central)	Serves the northbound applications such as portals or OSS systems, read-only data with paging, sorting and rich queries.

Table 17: Functions of Microservices (continued)

Microservice	Description
design-tools-central	Provides interface to Network Function Virtualization Design Tools to create config templates, VNF definitions and network service definitions.
Element Management Service (Central)	Maintains EMS device data model for device management functions. This data model contains device objects, abstract config, device inventory object, config template object, device profile object, device image object etc.
Fault and Performance Monitoring (FMPM) Collector Services	Describes APIs used by fault monitoring and performance monitoring system for collecting service check results from telemetry agents.
IAM Service	Provides identity and access management features.
IAM Service (No Authentication)	Provides identity and access management features during password recovery procedures.
Image Management Service (Central)	Provides image management functions.
Inventory Management Service (Central)	Provides generic inventory management functions.
Job Service	• Provides job management functionality • Supports creation of synchronous and asynchronous jobs, track status, rack start and completion time.
Intent based Policy Management	Provides Policy and SLA profile object management service to enable software-defined WAN (SD-WAN) functions.
Policy and SLA management Service	Enables software-defined WAN (SDWAN) function
Routing Manager Service	Provides APIs to manage routing operations such as to create VPN, interface to route-reflector, enable routing on CPE locations.
Schema Service	• Provides highly available, persistent data store for various schemas used by CSP applications. • Provides APIs to create, read, update, and delete schemas.
Shared Object Service	Varies based on type of schema
Signature Manager Service	Manages application signatures

Table 17: Functions of Microservices (continued)

Microservice	Description
Template Service	Provides config template of CSO management feature. This feature maintains database of config templates, template syntax validation (e.g - jinja2, python, yang rpc), template execution with input parameters using Yang RPC, and input/output validation ((provided corresponding schema is given).
Topology Service	Provides API for modeling topologies and working with network elements like devices, hubs, spokes, policy enforcement points and other objects.
Tenant, Site and Service Manager Service	Provides APIs for tenant, site and service management
VIM	Provides common APIs to create virtual networks, virtual links, instantiate VNFs, instantiate service chains for various virtual network infrastructures.

Related Documentation

- *Contrail Service Orchestration Monitoring and Troubleshooting Guide*

CHAPTER 5

Upgrading Contrail Service Orchestration

Upgrading Contrail Service Orchestration Overview

If your installed version is Contrail Service Orchestration (CSO) Release 3.2.1 or higher, you can use a script to upgrade to CSO Release 4.0.0.

CAUTION: **The upgrade process supports *n*–1 release upgrade approach. For example, if you have CSO Release 3.2.1 and you intend to upgrade it to CSO Release 4.0.0, you must upgrade to CSO Release 3.3 or CSO Release 3.3.1 first and then follow the same steps to upgrade to CSO Release 4.0.0.**

We recommend that you take snapshots of your current VMs before you proceed with the upgrade process.

You can roll back to the previous CSO release if the upgrade is unsuccessful, provided you have taken snapshots of the VMs.

To upgrade to CSO Release 4.0.0, you must run the scripts that are available in the **Contrail_Service_Orchestration_4.0.0.tar.gz** file in the following order:

1. **upgrade.sh**—This script upgrades CSO software from Release 3.2.1 to Release 4.0.0. The **upgrade.sh** script, puts CSO in maintenance mode, takes a snapshot of all VMs so that you can roll back to the previous release if the upgrade fails (optional), upgrades all microservices and infrastructure components if required, performs health checks at various levels, validates if all VMs, infracomponents, and microservices are up and running, and puts the CSO in live mode.

NOTE: Before you upgrade ensure that all ongoing jobs are stopped; otherwise, the upgrade process will fail. During the upgrade, you experience a downtime as CSO goes into maintenance mode.

2. **revert.sh**—Run this script only if the upgrade fails and if you have taken a snapshot of all VMs. This script reverts to the previously installed version.

Upgrade to CSO Release 4.0.0 is independent of the deployment type (HA and non-HA), environment type (small, medium or large), infrastructure components and microservices used, and the hypervisor type (KVM or VMware ESXi).

To ensure a smooth upgrade, the scripts perform a number of health checks before and after the upgrade. Health checks are performed to determine the operational condition of all components, the host, and VMs. If there is an error during the health check, the upgrade process is paused. You can rerun the script to rectify the error that you encounter.

Following are the types of health checks that are performed:

- Component health checks—Checks the operational condition of the infrastructure components.
- System health checks—Checks the following parameters of VMs and the host machine.
  - Available space on the host machine and VMs
  - Operating System (OS) version of the host machine and VMs
  - Kernel version of the host machine and VMs
  - Disk space on the host machine and VMs

Limitations

NOTE: The upgrade process is applicable to CSO Release 3.2.1 and later.

Upgrade to CSO Release 4.0.0 has the following limitations:

- There are no changes to the device image or device configurations.
- Upgrade cannot be performed through the GUI.
- CSO Release 4.0.0 allows to separate OS and Data partitions for CSO deployments. This feature is not applicable for upgrades.

Impact of the CSO Upgrade

Table 18 on page 99 describes the impact of the CSO upgrade to Release 4.0.0.

Table 18: Impact of the CSO Upgrade

Feature	After the Upgrade
Security Management	• Release 4.0.0 security management-related features are supported on devices that are onboarded in Release 3.3 or 3.3.1.
SD-WAN	• For the Application Visibility feature, the trend data is reset after the upgrade. You can access the Release 3.2.1 trend data through the REST APIs. • The Application Quality of Experience (AppQoE) feature works only for the tenants that you create in Release 4.0.0. For more information on AppQoE, see *Application Quality of Experience (AppQoE) Overview* in the *Contrail Service Orchestration User Guide*. • Device Management functions work for Release 3.3.1 or higher sites.
Cloud CPE	• All functionalities of centralized and distributed deployments continues to work on Release 4.0.0 sites or devices that are onboarded in Release 3.3.1 or higher. • Multi-region support for centralized deployment is not supported on Release 3.3.1 sites or devices that are onboarded in Release 3.3.1. • Device Management functions work for Release 4.0.0 sites. • High availability (HA) for VRRs is not supported for sites that are created in Release 3.2.1.

Related Documentation

• Upgrading to Contrail Service Orchestration Release 4.0.0 on page 99

Upgrading to Contrail Service Orchestration Release 4.0.0

The upgrade process supports *n*–1 release upgrade approach. For example, if you have CSO Release 3.2.1 and you intend to upgrade it to CSO Release 4.0.0, you must upgrade to CSO Release 3.3 or CSO Release 3.3.1 first and then follow the same steps to upgrade to CSO Release 4.0.0.

If your CSO release version is lower than 3.2.1, you have to opt for fresh installation.

We recommend that you take a snapshot of your current configuration and VMs before you proceed with upgrade process.

NOTE: **The upgrade process is applicable to CSO Release 3.2.1 and later.**

This upgrade procedure is independent of the deployment type (trial and production), environment type (non-HA and HA), infrastructure components and microservices used, and the hypervisor type (KVM or VMware ESXi).

Before you begin the upgrade:

- Ensure that you are in Contrail Service Orchestration (CSO) Release 3.2.1 or later.
- Ensure the installer Virtual Machine (VM) is up and running.
- If you are using VMware ESXi VMs, you must create the **provision_vm.conf** file in the **Contrail_Service_Orchestration_4.0.0/confs/** directory.

To upgrade to CSO Release 4.0.0:

1. Download the CSO Release 4.0.0 installer package from the Software Downloads page to the local drive.

2. Log in to the installer VM as root.

3. Copy the installer package from your local folder to the installer VM.

   ```
   root@host:~/# scp Contrail_Service_Orchestration_4.0.0.tar.gz root@installer
   VM  :/root
   ```

4. On the installer VM, extract the installer package.

   For example, if the name of the installer package is **Contrail_Service_Orchestration_4.0.0.tar.gz**,

   ```
   root@host:~/# tar -xvzf Contrail_Service_Orchestration_4.0.0.tar.gz
   ```

   The contents of the installer package are extracted in a directory with the same name as the installer package.

5. Navigate to the CSO Release 4.0.0 directory in the installer VM.

   ```
   root@host:~/# cd Contrail_Service_Orchestration_4.0.0
   root@host:~/Contrail_Service_Orchestration_4.0.0#
   ```

   (Optional) You can view the list of files in the Contrail_Service_Orchestration_4.0.0.

   ```
   root@host:~/Contrail_Service_Orchestration_4.0.0# ls
   ```

   The Contrail_Service_Orchestration_4.0.0.tar.tz file includes the following scripts:

   - **upgrade.sh**
   - **revert.sh**

6. Run the **upgrade.sh** script.

   WARNING: **Before you upgrade ensure that all ongoing jobs in Administration Portal and Customer Portal are stopped; otherwise, the upgrade process will fail. During the upgrade, you experience a downtime as CSO goes into maintenance mode.**

   This script upgrades CSO software from Release 3.3.1 to Release 4.0.0. The **upgrade.sh** script puts CSO in maintenance mode; takes a snapshot of running status of all VMs (optional); upgrades all microservices and infrastructure components, if required; performs health checks at various levels; validates whether all VMs, infrastructure components, and microservices are up and running; and puts the CSO in live mode. The **upgrade.sh** script takes a snapshot of all VMs by default.

NOTE: The script does not take a snapshot of the Installer VM and Virtual Route Reflector (VRR) VM.

```
root@host:~/Contrail_Service_Orchestration_4.0.0# ./upgrade.sh

INFO      Configuration Upgrade : success
INFO      System Health Check : success
INFO      CSO Health-Check Before Upgrade : success
INFO      CSO Maintenance Mode Enabled : success
INFO      Kernel Upgrade : NA
INFO      VM Snapshot : success
INFO      Selective Infra Components Upgrade : NA
INFO      Central Infra Upgrade : success
INFO      Regional Infra Upgrade : success
INFO      Microservices pre-deploy scripts execution : success
INFO      Central Microservices Upgrade : success
INFO      Regional Microservices upgrade : success
INFO      Microservices post-deploy scripts execution : success
INFO      CSO Health-Check after Upgrade : success
INFO      Enable CSO Services : success
INFO      Load Microservices Data : success
INFO     ==========================================
INFO     CSO is successfully upgraded to Release
Contrail_Service_Orchestration_4.0.0
INFO     ==========================================
```

NOTE: The password for each infrastructure component is displayed on the console after the upgrade is successful. You must note the password that is displayed on the console as they are not saved in the system. To enhance the password security, the length and pattern for each password is different, the password is encrypted, and passwords in the log file are masked.

The time taken to complete the upgrade process depends on the hypervisor type and the environment type. All VMs on KVM are shut down while a snapshot is being taken. All VMs on VMware ESXi are up and running while a snapshot is being taken

If an error occurs, you must fix the error and rerun the **upgrade.sh** script. When you rerun the **upgrade.sh** script, the script continues to execute from the previously failed step.

You can view the following log files that are available at **root/Contrail_Service_Orchestration_4.0.0/logs**:

- **upgrade_console.log**
- **upgrade_error.log**
- **upgrade.log**

7. (Optional) If you are unable to troubleshoot the error you can roll back to your previous release. Run the **revert.sh** script.

NOTE: You can roll back to the previous release only if you have taken snapshot of VMs before starting the upgrade process.

```
root@host:~/Contrail_Service_Orchestration_4.0.0# ./revert.sh

INFO     revert    revert.py            Overall Revert Summary
INFO     revert    revert.py ============================================
INFO     revert    revert.py  Revert VM Snapshots : success
INFO     revert    revert.py  Revert Kernel Upgrade : NA
INFO     revert    revert.py  Revert Salt Master Configurations : success
INFO     revert    revert.py  Post-revert processes : success
INFO     revert    revert.py  CSO Health-Check after Revert : success
INFO     revert    revert.py  Start Kubernet pods : success
INFO     revert    revert.py  Enable CSO Services : success
INFO     revert    revert.py
============================================================
INFO     revert    revert.py CSO successfully reverted to the previously
installed version.
INFO     revert    revert.py
============================================================
```

After a successful upgrade, CSO is functional and you can log in to Administrator Portal and Customer Portal.

NOTE: After you have successfully upgraded from CSO Release 3.3.1 to Contrail Service Orchestration (CSO) Release 4.0.0, ensure that you download the application signatures before installing signatures on the device. This is a one-time operation after the upgrade.

Related Documentation

- Upgrading Contrail Service Orchestration Overview on page 97

Adding Virtual Route Reflectors (VRRs) After Upgrading to CSO Release 4.0.0

To support high availability (HA) for Virtual Route Reflectors (VRRs), you must add VRRs and create redundancy groups after you upgrade to Contrail Service Orchestrator (CSO) Release 4.0.0.

To add VRRs:

1. Log in to the installer VM as root.

2. Navigate to the CSO Release 4.0.0 directory in the installer VM.

```
root@host:~/# cd Contrail_Service_Orchestration_4.0.0
root@host:~/Contrail_Service_Orchestration_4.0.0#
```

3. Run the **add_vrr.sh** script.

```
root@host:~/Contrail_Service_Orchestration_4.0.0# ./add_vrr.sh
```

The existing VRR details are displayed.

```
========== Existing VRR Details ========

     host-name               |    redundancy-group

     vrr-192.204.243.28                0

=========================================
```

NOTE: By default, VRRs that are created with Release 3.2.1 or later belong to the redundancy group *group 0*.

4. To add VRRs, you are prompted to answer the following questions:

   a. Specify whether VRR is behind NAT.

      - y—VRR is behind NAT. If you are deploying a VRR in a private network, the NAT instance translates all requests (BGP traffic) to a VRR from a public IP address to a private IP address.
      - n—VRR is not behind NAT (default).

   b. Specify whether you want to use a common password for all VRRs.

      - If you want to use a common password for all VRRs, enter **y** and specify the common password.
      - If you want to use a different password for each VRR, enter **n** and specify the password for each VRR.

   c. Specify the number of VRR instances.

      - For non-HA deployments, you must create at least one VRR.
      - For HA deployments, we recommend that you create VRRs in even numbers, and you must create at least two VRRs. Each VRR must be in a different redundancy group. If the primary VRR fails or connectivity is lost, the session remains active as the secondary VRR continues to receive and advertise LAN routes to a site, thereby providing redundancy.

   d. For each VRR instance, specify the following:

      - Specify the public IP address for each VRR that you create. For example, 192.110.20.118/24.
      - Specify the redundancy group for each VRR that you have created.
        - For non-HA deployments, specify the redundancy group of the VRR as zero.
        - For HA deployments, the VRRs must be distributed among the redundancy groups. There can be two groups—group 0 and group 1. For example, if you

have two VRRs, specify the redundancy group for VRR1 as 0 and the VRR2 as 1.

- Specify the username for each VRR.
- Specify the password for each VRR.

  If you have chosen a common password for all VRRs, you are prompted to specify the common password only for the first VRR instance.

You can view the newly added VRRs through the APIs: routing-manager (GET: https://*IP Address of Administration Portal*/routing-manager/vrr-instance) or ems-central (GET: https://*IP Address of Administration Portal*/ems-central/device).

Each hub or spoke device establishes a BGP peering session with VRRs that you have created and assigned to different redundancy groups, thereby providing redundancy.

Related Documentation

- Upgrading to Contrail Service Orchestration Release 4.0.0 on page 99

Troubleshooting Upgrade-Related Errors

This topic describes the possible errors that you might encounter while you are upgrading Contrail Service Orchestrator (CSO).

It also suggests how to resolve those errors.

Salt Synchronization Error

Problem

Description: The upgrade or revert status is displayed as **Going to sync salt...** for a considerable time while upgrading CSO to CSO Release 4.0.0 or reverting to the previously installed release.

The Salt Master on the installer VM might be unable to reach all Salt Minions on the other VMs and a salt timeout exception might occur.

Solution

Based on the output of the **salt '*' test.ping** command, you must restart either the Salt Master or the Salt Minion.

To resolve the error:

1. Open another instance of installer VM.
2. Run the **salt '*' test.ping** command, to check whether the Salt Master on the installer VM is able to reach other VMs.

```
root@host:~/# salt '*' test.ping
```

- Restart the Salt Master if the following error occurs:

  Salt request timed out. The master is not responding. If this error persists after verifying the master is up,worker_threads may need to be increased

  ```
  root@host:~/# service salt-master restart
  ```

- If there are no errors, review the output.

  ```
  root@host:~/# salt '*' test.ping

  csp-regional-sblb.DB7RFF.regional:
  True
  csp-contrailanalytics-1.8V1O2D.central:
      True
  csp-central-msvm.8V1O2D.central:
      True
  csp-regional-k8mastervm.DB7RFF.regional:
      True
  csp-central-infravm.8V1O2D.central:
      False
  csp-regional-msvm.DB7RFF.regional:
      False
  csp-regional-infravm.DB7RFF.regional:
      True
  csp-central-k8mastervm.8V1O2D.central:
      True
  ```

  You must log in to the VM and restart the Salt Minion whether the status of the VM is False.

  ```
  root@host:~/csp-central-infravm.8V1O2D.central# service salt-minion restart
  ```

3. Rerun the **salt '*' test.ping** command to verify whether the status for all VMs is True.

Cache Clearance Error

Problem **Description:** The following error might occur while upgrading CSO to CSO Release 4.0.0:
Could not free cache on host server *ServerName*

Solution You must clear the cache on the host server.

To resolve the error:

1. Log in to the host server through SSH.

2. To clear the cache, run the following command:

   ```
   root@host:~/Contrail_Service_Orchestration_4.0.0# free && sync && echo 3 >
   /proc/sys/vm/drop_caches && free
   ```

   The following output is displayed:

```
             total       used       free     shared    buffers     cached
Mem:     264036628  214945716   49090912      15092     198092   71878992
-/+ buffers/cache:  142868632  121167996
Swap:    390233084     473808  389759276
             total       used       free     shared    buffers     cached
Mem:     264036628  142165996  121870632      15092       3256      75792
-/+ buffers/cache:  142086948  121949680
Swap:    390233084     473808  389759276
```

The cache is cleared on the host server.

Kube-system Pod Error

Problem **Description:** The following error might occur while upgrading CSO to CSO Release 4.0.0:

One or more kube-system pods are not running

Solution Check the status of the *kube-system* pod, and restart *kube-proxy* if required.

To resolve the error:

1. Log in to the central or regional microservices VM through SSH.
2. Run the following command to view the status of the *kube-system* pod:

   ```
   root@host:~/# kubectl get pods –namespace=kube-system
   ```

   The following output is displayed:

   ```
   NAME                                    READY     STATUS    RESTARTS   AGE
   etcd-empty-dir-cleanup-10.213.20.182    1/1       Running   2          3d
   kube-addon-manager-10.213.20.182        1/1       Running   2          3d
   kube-apiserver-10.213.20.182            1/1       Running   2          3d
   kube-controller-manager-10.213.20.182   1/1       Running   6          3d
   kube-dns-v11-4cmhl                      4/4       Running    0          3h
   kube-proxy-10.213.20.181                0/1       Error    2          3d
   kube-scheduler-10.213.20.182            1/1       Running   6          3d
   ```

   Check the status of *kube-proxy*. You must restart *kube-proxy* if the status is *Error*, *Crashloopback*, or *MatchNodeSelector*.

3. Run the following command to restart *kube-proxy*

   ```
   root@host:~/# Kubectl apply –f /etc/kubernetes/manifests/kube-proxy.yaml
   ```

The *kube-system* pod-related error is resolved.

Kubernetes Node Error

Problem **Description:** The following error might occur while upgrading CSO to CSO Release 4.0.0:

One or more nodes down

Solution Check the status of *kube-master* or *kube-minion* and restart the nodes, if required.

To resolve the issue:

1. Log in to the central or regional microservices VM through SSH.

2. Run the following command to check the status of each node:

```
root@host:~/# kubectl get nodes

NAME            STATUS     AGE       VERSION
10.213.20.181   Not Ready    3d        v1.6.0
10.213.20.182   Ready        3d        v1.6.0
```

   Identify the node that is in the *Not Ready* status. You must restart the node if the status is *Not Ready*.

3. Restart the node if the status is *Not Ready* by logging in to the node through SSH and running the following command:

```
root@host:~/# service kubelet restart
```

4. Rerun the following command to check the status of the node that you restarted.

```
root@host:~/# kubectl get nodes
```

The Kubernetes node-related error is resolved.

Related Documentation

- Upgrading to Contrail Service Orchestration Release 4.0.0 on page 99

