Juniper Networks | Engineering Simplicity

# cPCE Deployment Guide

Published
2025-08-05

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



*cPCE Deployment Guide*


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

# Table of Contents


About This Guide | v

1 Overview

Containerized PCE | 2

Resource Requirements | 7

2 Install and Configure

Install using Docker | 11

Install cPCE and cPCEPAdaptor Using Docker | 11

Install and Verify Docker | 11

Download Software | 12

Create cPCE | 12

Create cPCEPAdaptor | 13

Configure External Path Computation Components | 14

Overview | 14

Configure cPCE | 15

Create and Configure cPCEPAdaptor | 17

Configure R1 as PCC | 18

Configure Router R2 | 20

Verify cPCEPAdaptor Status | 21

Verify BGP on cPCE | 23

Verify PCC Status | 24

3 Troubleshooting

Debug cPCE and cPCEP Adaptor | 27

Troubleshooting Container | 27

Verify Docker | 28

Verify Reachability of cPCE Bash Shell | 29

Verify Sample outputs from cPCE | 29

Verify Reachability from R1 | 30

Verify Sample Outputs from cPCEPAdaptor | **32**

Verify Sample Outputs from R1 | **33**

View Trace Files | **34**

Restart cPCEPAdaptor | **35**

# About This Guide

Use this guide to install the containerized path computation engine (cPCE) in the Linux environment. This guide also includes cPCE configuration procedures.

1
CHAPTER

# Overview

**IN THIS CHAPTER**


- Containerized PCE | 2
- Resource Requirements | 7

# Containerized PCE

IN THIS SECTION


- Overview | 5
- Benefit of cPCE | 7
- Licensing | 7

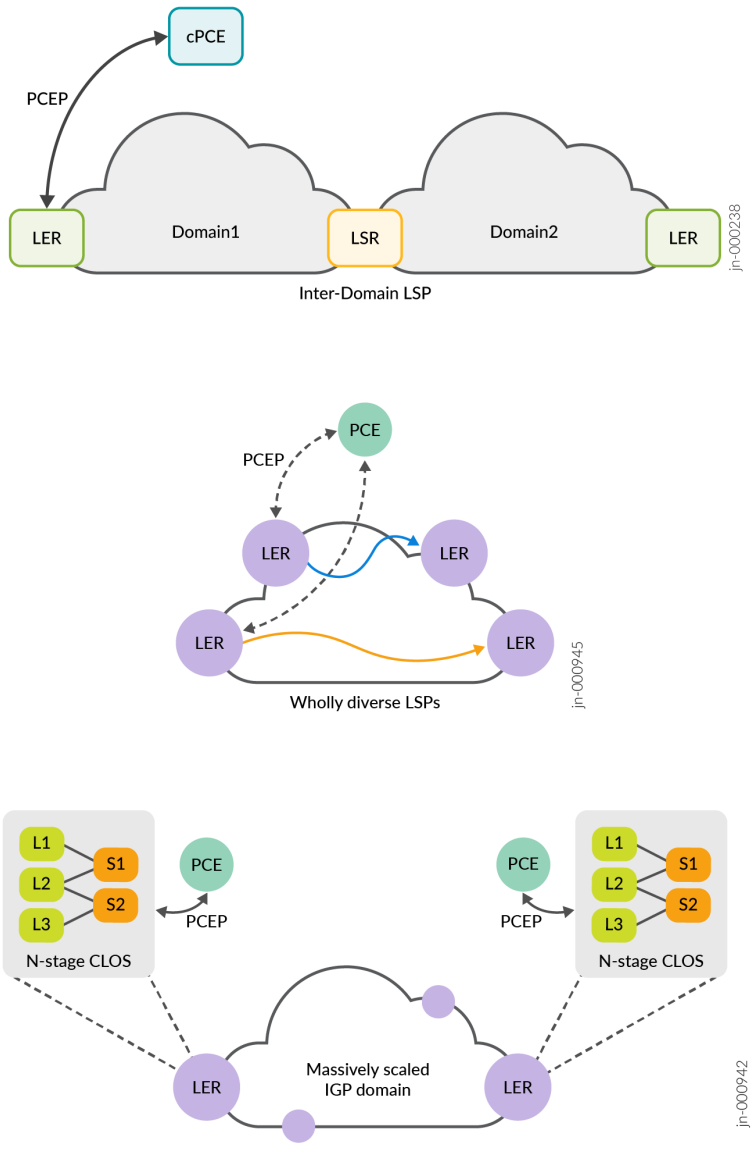

Containerized Path Computation Element (cPCE) is a light-weight micro-services based containerized Path Computation Engine application that can be deployed on or off box. Compute Off-load is a Traffic Engineering (TE) use-case where the computation of TE-Paths is off-loaded by the head-end router onto an off-box computation engine.

To perform Constraint-based Path Computation (CSPF). you need to configure external components outside of the router. The goal of the PCE architecture is to provide special computation components for computing CSPF, especially for more complex use-cases such as Inter-domain path establishment or wholly diverse paths. The PCE architecture also provides a foundation for more general use-cases such as the coupling of powerful computation services with less sophisticated or resource constrained head-end or ingress Label Edge Routers (LERs) that are becoming more and more prevalent in modern networks.

### Use Case Inter-domain Path

The cPCE architecture provides components for computing CSPF in the use cases such as inter-domain path establishment or wholly diverse paths.

**Figure 1: Inter-domain Path**

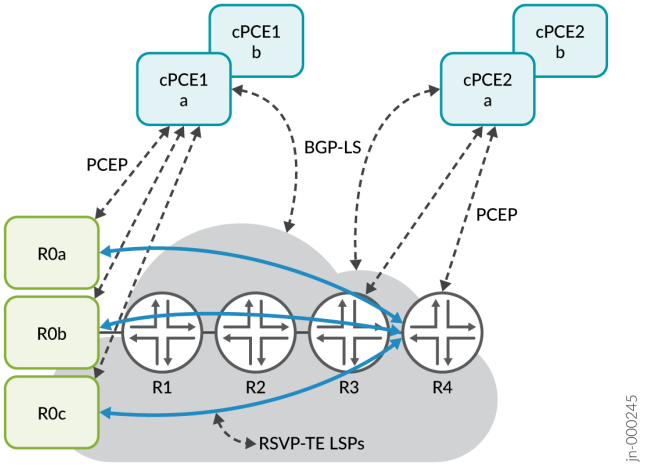



## Use Case Compute Offload

cPCE is useful for computing and delegating traffic-engineered label-switched paths (LSPs) in an RSVP-Traffic Engineering (RSVP-TE) network development.

R0a, R0b, and R0c are ingress routers that are offloading path computation to cPCE. cPCE1a is acting as primary PCE and cPCE1b is secondary PCE. cPCE2a primary is catering to R4 using PCEP messages and cPCE2b is secondary.

**Figure 2: Compute Off-load**

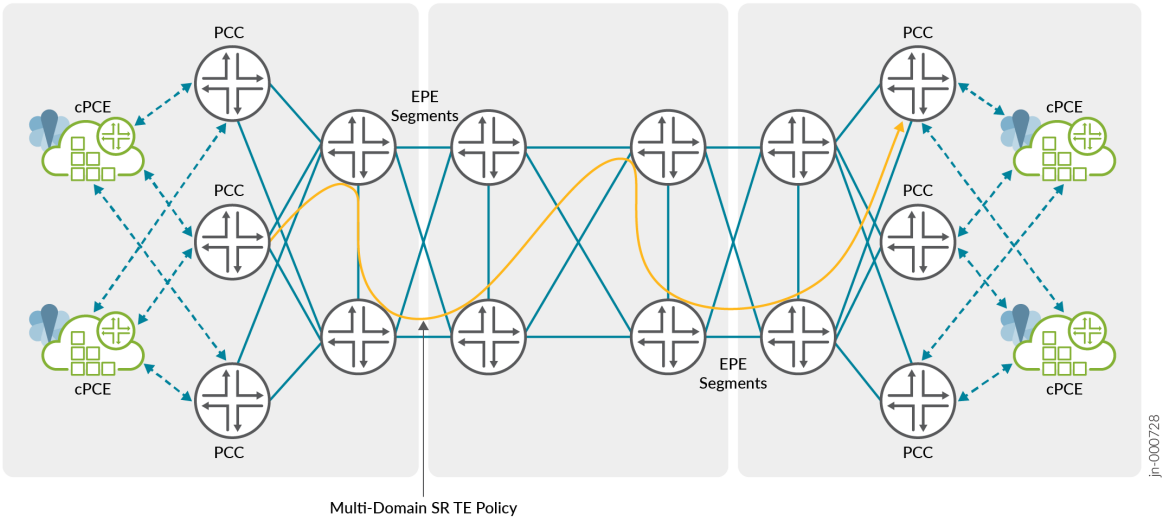



## Use case eODN

This use case describes cPCE distributed enhanced On-Demand Nexthop (eODN) use case in a network deployment where Segment Routing traffic engineering (TE) is used as the path-control technology. eODN is a TE use case where interdomain TE tunnels or policies are dynamically computed and placed when a BGP VPN route arrives on an ingress node with a protocol nexthop for a destination from remote domain.

Domain specific cPCE enables inter-domain on-demand SR policy creation. Each domain has one or more redundant cPCE responsible for inter-domain computation services.

**Figure 3: eODN**




## Overview

cPCE combines multiple Junos components to form a PCE. These components are shown in Figure 4 on page 6 Each container instance is individually manageable through Junos CLI.

The cPCE comprises of the following services as shown in Figure 5 on page 6:

- Path Computation Element (PCE) Communication Protocol (PCEP) session management and maintenance
- Topology acquisition
- Path computation

**Figure 4: PCE**

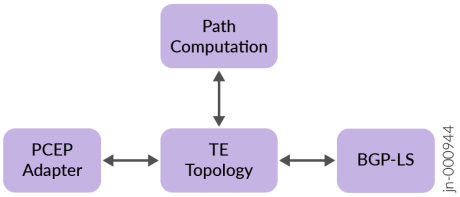
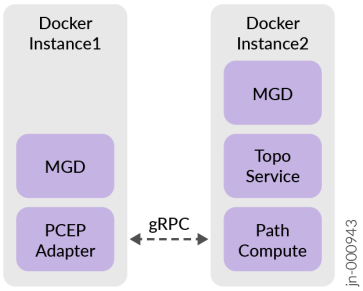



**Figure 5: Components of RPD PCE**




cPCE offers the flexibility to deploy a PCE in different ways; on-box, or off-box, or as a third party application on Junos OS Evolved. Furthermore, you can deploy cPCE in a scale-out fashion to cater to the specific scalability, redundancy, or performance needs of a given deployment

The cPCE solution consists of two containerized components.

- cPCEPAdaptor
- cPCE (cRPD)

PCEP describes the protocol for PCE and Path Computation Client (PCC) to communicate with each other. PCEP provides mechanisms for PCEs to perform path computations in response to PCC requests.

PCEP extends the PCE communication protocol for stateful PCEs by specifying a set of extensions to PCEP to enable stateful control of LSPs within and across PCEP sessions. It includes mechanisms to effect LSP state synchronization between PCCs and PCEs, delegation of control over LSPs to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions.

cPCEPAdaptor container instance handles the PCEP messages and maintains the LSP state. cPCE (cRPD) instance provides the actual path computation service. The cPCE instance learns topology information from BGP-LS connections.

The cPCEPAdaptor processes the PCEP reports from the ingress PCC device and triggers path-computation requests and sends them to the cPCE. The cPCE receives the set of constraints for the LSP and delivers computation results to the cPCEPAdaptor.

**Figure 6: Deployment Model**


## Benefit of cPCE

- Scale out compute services

## Licensing

You need a license to activate cPCE software features. To understand more about cPCE licenses, see Flex Licenses for cRPD, Flex Software License Model, and Managing cRPD Licenses.

RELATED DOCUMENTATION

Configure External Path Computation Components | 14

Express Segment

# Resource Requirements

IN THIS SECTION

- cPCE Scaling | 8

This section presents an overview of minimum requirements for deploying a cPCE and cPCEP Adaptor on a Linux server. For information on resource requirements, see cPCE Requirements.

**Table 1: cPCEP Adaptor Resource Requirements**

| Description | Minimum Value |
|---|---|
| CPU | 1 core |
| Memory | 512 MB |
| Disk space | 512 MB |

**Table 2: Minimmum Host Requirements**

| Component | Specification |
|---|---|
| Linux OS support | Ubuntu 18.04 |
| Linux Kernel | 4.15 |
| Docker Engine | 18.09.1 |
| Host processor type | ARM64 multicore CPU |
| Network Interface | Ethernet |

## cPCE Scaling

A single cPCE (container set) will support up to 8 PCCs & 8,000 LSPs

**Table 3: cPCE Scaling**

| Component | Specification |
|---|---|
| PCC | 8 |
| LSP | 8000 |

2

CHAPTER

# Install and Configure

**IN THIS CHAPTER**


- Install using Docker | **11**

# Install using Docker

IN THIS SECTION


- Install cPCE and cPCEPAdaptor Using Docker | 11
- Configure External Path Computation Components | 14


## Install cPCE and cPCEPAdaptor Using Docker

SUMMARY

IN THIS SECTION


- Install and Verify Docker | 11
- Download Software | 12
- Create cPCE | 12
- Create cPCEPAdaptor | 13


### Install and Verify Docker

Install and configure Docker on Linux host platform to implement the Linux container environment, see Install Docker for installation instructions on the supported Linux host operating systems.

Verify the Docker installation.

To install the latest Docker:

Log in and download the software.

```
root@ubuntu-vm18:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

root@ubuntu-vm18:~# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

root@ubuntu-vm18:~# apt update

root@ubuntu-vm18:~# apt install docker-ce
```

## Download Software

The cPCE and cPCEPAdaptor software is available as a Docker file from the Juniper software download page.

Before you import the software, ensure that Docker is installed on the Linux host and that the Docker Engine is running.

After you install the Docker Engine on the host, perform the following to download and start using the cPCE image:

To download the software:

1. Download the cRPD software image from the Juniper Networks website.
2. Load the downloaded docker image on the local Linux host where docker is installed.

   `root@ubuntu-vm18#` **docker load -i junos-routing-crpd-docker-amd64-23.2R1.13.tgz**
3. Verify the cPCE image in docker image repository.

   `root@ubuntu-vm18#` **docker images**

```
REPOSITORY   TAG         IMAGE ID       CREATED        SIZE
crpd         23.2R1.13   0cf5adbda509   5 months ago   498MB
```

4. Download the PCEP adaptor software image from the Juniper Networks website.
5. Load the downloaded docker image on the local Linux host where docker is installed.

   `root@ubuntu-vm18#` **docker load -i junos-cpcep-adaptor-docker-amd64-23.2R1.9.tgz**
6. Verify the cpcepadaptor image in docker image repository.

   `root@ubuntu-vm18#` **docker images**

```
REPOSITORY            TAG                    IMAGE ID           CREATED            SIZE
crpd           23.2R1.13   0cf5adbda509   5 months ago   498MB
cpcepadaptor   23.2R1.9    9979e81995c5   5 months ago   404MB
```

## Create cPCE

To create cPCE:

1. Run the command to launch cRPD-cpce instance:

   `root@ubuntu-vm18:~#` **docker run --name cRPD-cpce -d --privileged -p 179:179 crpd:23.2R1.13**
2. Log in to docker console.

root@ubuntu-vm18:~# **docker exec -it cRPD-cpce /bin/bash**

```
===>
           Containerized Routing Protocols Daemon (CRPD)
 Copyright (C) 2020-2023, Juniper Networks, Inc. All rights reserved.
                                                                  <===
```

3. Enter into configuration mode.

   root@cRPD-cpce:/# **cli**

   root@cRPD-cpce> **configure**

```
Entering configuration mode

[edit]
root@cRPD-cpce#
```

## Create cPCEPAdaptor

To create cPCEPAdaptor:

1. Run the command to launch cPCEPAdaptor:

   root@ubuntu-vm18:~# **docker run --name CPCE1-pcepAdaptor -d --privileged -p 4189:4189 cpcepadaptor:23.2R1.9**

2. Log in to the cPCEPAdaptor container using CLI.

   root@ubuntu-vm18:~# **docker exec -it CPCE1-pcepAdaptor /bin/bash**

```
===>
           Containerized PCEP Adaptor (cPCEP Adaptor)
 Copyright (C) 2021-2023, Juniper Networks, Inc. All rights reserved.
                                                                  <===
```

3. Enter into configuration mode.

   root@CPCE1-pcepAdaptor:/# **cli**

   root@CPCE1-pcepAdaptor> **configure**

```
Entering configuration mode
```

```
[edit]
root@CPCE1-pcepAdaptor#
```

RELATED DOCUMENTATION

Configure External Path Computation Components | 14

# Configure External Path Computation Components

IN THIS SECTION


- Overview | 14
- Configure cPCE | 15
- Create and Configure cPCEPAdaptor | 17
- Configure R1 as PCC | 18
- Configure Router R2 | 20
- Verify cPCEPAdaptor Status | 21
- Verify BGP on cPCE | 23
- Verify PCC Status | 24


This section outlines the steps to install the cPCE container and cPCEPAdaptor in a Linux server environment that is running Ubuntu or Red Hat Enterprise Linux (RHEL). The cPCE and cPCEPAdaptor containers are packaged in a Docker image and run in the Docker Engine on the Linux host.

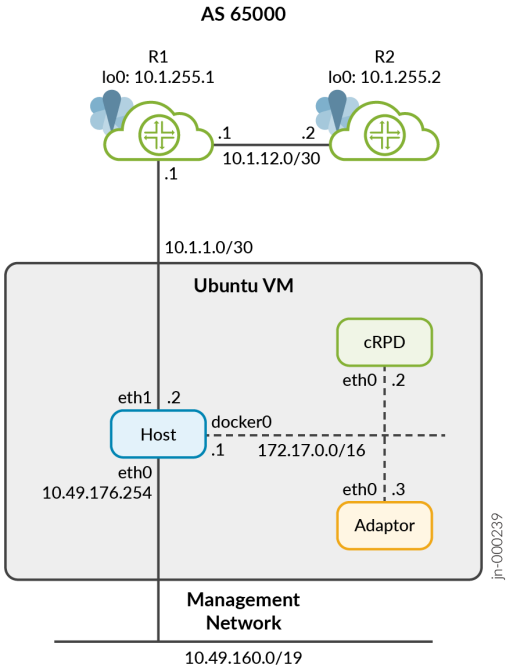
## Overview

In the Figure 7 on page 15, R1 is the Path Computation Client (PCC). Traffic engineered tunnels are configured and delegated to cPCE. This section provides information on how you can configure the connections between the cPCE, cPCEPAdaptor and the BGP link-state sessions necessary for topology acquisition.

You can install both the cPCE and cPCEPAdaptor on same server or on different hosts or routers that support the docker environment. The cPCEPAdaptor manages the PCEP sessions with the PCCs. Theis section describes how you can configure the gRPC connection between one or more PCCs and a

cPCEPAdaptor. The cPCEPAdapter requires a set of gRPC connection parameters to be configured for establishing and maintaining a gRPC channel with the cPCE path computation service.

**Figure 7: Topology**




The PCC (R1) and cPCEPAdaptor communicate through standard PCEP messages. The cPCEPAdaptor and the cPCE communicate through gRPC messages. In this topology, IP address 10.1.255.1 is the loopback address or the router Id of the PCC (R1) router. The loopback address or the router Id of the PCC (R2) router is 10.1.255.2.

The cPCE requires a set of connection parameters to establish a gRPC channel with the cRPD for path computation service.

## Configure cPCE

To configure cPCE:

1. Set the root authentication password by entering a cleartext password, an encrypted password, or an SSH public key string (DSA or RSA).

```
root@cRPD-cpce# set system root-authentication plain-text-password

New password: xxx
Retype new password: xxx
```

2. Specify the IP address and the port number to build the gRPC connection between cPCEPAdaptor and cPCE.

   [edit system services]

   ```
   root@cRPD-cpce# set extension-service request-response grpc clear-text port 50051
   root@cRPD-cpce# set extension-service request-response grpc clear-text address 0.0.0.0
   root@cRPD-cpce# set extension-service request-response grpc max-connections 30
   ```

3. Configure a policy statement to import the BGP-LS acquired topology into the local traffic-engineering database.

   [edit policy-options]

   ```
   root@cRPD-cpce# set policy-statement bgpl2_rt_2_ted term 1 from family traffic-engineering
   root@cRPD-cpce# set policy-statement bgpl2_rt_2_ted term 1 then accept
   ```

   [edit protocols mpls]

   ```
   root@cRPD-cpce# set traffic-engineering database export policy bgpl2_rt_2_ted
   ```

   Configure if cPCE is catering to segment routing traffic-engineering paths.

   [edit protocols mpls]

   ```
   root@cRPD-cpce# set traffic-engineering database export l3-unicast-topology
   ```

4. Configure the autonomous system number and establish reachability to the network. For example, a static route to Router R1.

   [edit routing-options]

   ```
   root@cRPD-cpce# set autonomous-system 65000
   root@cRPD-cpce# set static route 0.0.0.0/0 next-hop 172.17.0.1
   ```

5. Configure topology acquisition using BGP-LS on cPCE. The PCC router should have IP reachability with the cPCEPAdaptor and cPCE for PCEP session and BGP-LS session establishment.

   [edit protocols bgp]

   ```
   root@cRPD-cpce# set group 65000 type internal
   root@cRPD-cpce# set group 65000 passive
   root@cRPD-cpce# set group 65000 family traffic-engineering unicast
   ```

```
root@cRPD-cpce# set group 65000 allow 0.0.0.0/0
```

6. Apply templates (optional) that can be applied to the cPCE controlled LSPs for computing paths and to enable specific timers. For example, optimize-timer.

   [edit protocols mpls]

   ```
   root@cRPD-cpce# set lsp-external-controller remote-pce pce-controlled-lsp * label-switched-path-template t1

   root@cRPD-cpce# set label-switched-path t1 template

   root@cRPD-cpce# set label-switched-path t1 optimize-timer 45
   ```

7. Commit the configuration.

   ```
   root@cRPD-cpce# commit
   ```

```
commit complete
```

## Create and Configure cPCEPAdaptor

To configure cPCEPAdaptor:

1. Configure cPCEPAdaptor.

   ```
   root@CPCE1-pcepAdaptor# set system root-authentication plain-text-password
   ```

```
New password: xxx
Retype new password: xxx
```

2. Configure the gRPC request response service on the adaptor by specifying IP address and the port number to establish the connection between the cPCEPAdaptor and cPCE.

   [edit system services]

   ```
   root@CPCE1-pcepAdaptor# set extension-service request-response grpc clear-text address 0.0.0.0

   root@CPCE1-pcepAdaptor# set extension-service request-response grpc clear-text port 50051

   root@CPCE1-pcepAdaptor# set extension-service request-response grpc max-connections 30
   ```

   [edit services]

   ```
   root@CPCE1-pcepAdaptor# set path-computation-adaptor pce cRPD-cpce ipv4-address 172.17.0.2

   root@CPCE1-pcepAdaptor# set path-computation-adaptor pce cRPD-cpce port 50051

   root@CPCE1-pcepAdaptor# set path-computation-adaptor pce cRPD-cpce login-id root

   root@CPCE1-pcepAdaptor# set path-computation-adaptor pce cRPD-cpce password xxxx
   ```

3. Commit the configuration.

   ```
   root@CPCE1-pcepAdaptor# commit
   ```

   ```
   commit complete
   ```

## Configure R1 as PCC

Establish a Path Computation Element Protocol (PCEP) connection with the cPCEPAdaptor or cPCE. R1 will also be configured to export the traffic-engineering topology to cPCE through BGP-LS.

1. Configure the interfaces of the Router R1.

   [edit interfaces]

   ```
   root@R1# set ge-0/0/0 unit 0 family inet address 10.1.1.1/30
   root@R1# set ge-0/0/1 unit 0 family inet address 10.1.12.1/30
   root@R1# set ge-0/0/1 unit 0 family mpls
   root@R1# set lo0 unit 0 family inet address 10.1.255.1/32
   ```

2. Configure the BGP-LS traffic engineering database import policy.

   [edit policy-options]

   ```
   root@R1 set policy-statement TED_to_BGP-LS term 1 from protocol isis
   root@R1 set policy-statement TED_to_BGP-LS term 1 from protocol ospf
   root@R1 set policy-statement TED_to_BGP-LS term 1 then accept
   ```

   [edit protocols]

   ```
   root@R1 set mpls traffic-engineering database import policy TED_to_BGP-LS
   ```

3. Configure the router ID and assign an autonomous system number. .

   [edit routing-options]

   ```
   root@R1 set router-id 10.1.255.1
   root@R1 set autonomous-system 65000
   ```

4. Configure BGP internal group, the link-state address-family and assign the export policy

   edit [policy-options]

   ```
   root@R1 set policy-statement export_TED then accept
   ```

   [edit protocols bgp]

   ```
   root@R1 set group 65000 type internal
   root@R1 set group 65000 local-address 10.1.255.1
   ```

```
root@R1 set group 65000 family traffic-engineering unicast

root@R1 set group 65000 export export_TED

root@R1 set group 65000 allow 0.0.0.0/0

root@R1 set group 65000 neighbor 172.17.0.2
```

5. Specify the loopback address of the PCC router as the local address and the destination IP address of the host where cPCEPAdaptor instance is spawned. Configure the destination port for the PCC (R1) router that connects to the cPCEPAdaptor using PCEP and the PCE type.

   [edit protocols pcep]

```
root@R1# set pce cPCE1 local-address 10.1.255.1

root@R1# set pce cPCE1 destination-ipv4-address 172.17.0.3

root@R1# set pce cPCE1 destination-port 4189

root@R1# set pce cPCE1 pce-type active

root@R1# set pce cPCE1 pce-type stateful

root@R1# set pce cPCE1 lsp-provisioning

root@R1# set pce cPCE1 spring-capability
```

6. Configure IS-IS, Resource Reservation Protocol (RSVP), and segment routing to illustrate external control through cPCE of both traffic engineering (SR-TE) and RSVP-TE traffic engineered tunnels.

   [edit protocols rsvp]

```
root@R1# set interface ge-0/0/1.0

root@R1# set interface lo0.0
```

   [edit protocols isis]

```
root@R1# set interface ge-0/0/1.0

root@R1# set interface lo0.0 passive

root@R1# set source-packet-routing node-segment ipv4-index 401

root@R1# set traffic-engineering l3-unicast-topology
```

7. Configure RSVP-TE label-switched path (LSP) and enable external control.

   [edit protocols mpls]

```
root@R1# set lsp-external-controller pccd

root@R1# set interface ge-0/0/1.0

root@R1# set label-switched-path to-R2 to 10.1.255.2
```

```
root@R1# set label-switched-path to-R2 lsp-external-controller pccd
```

8. Configure SR-TE LSP and enable external control.

   [edit protocols source-packet-routing]

   ```
   root@R1# set lsp-external-controller pccd
   ```

   [edit protocols source-packet-routing]

   ```
   root@R1# set source-routing-path computels1 to 10.1.255.2
   root@R1# set source-routing-path computels1 lsp-external-controller pccd
   root@R1# set source-routing-path computels1 primary p1 compute compute1
   root@R1# set compute-profile compute1 maximum-computed-segment-lists 1
   ```

9. Commit the configuration.

   ```
   root@R1# commit
   ```

## Configure Router R2

1. Configure interfaces of Router R2.

   [edit interfaces]

   ```
   root@R2# set ge-0/0/0 unit 0 family inet address 10.1.12.2/30
   root@R2# set ge-0/0/0 unit 0 family mpls
   root@R2# set lo0 unit 0 family inet address 10.1.255.2/32
   ```

2. Configure the router ID.

   [edit routing-options]

   ```
   root@R2# set router-id 10.1.255.2
   ```

3. Configure MPLS interface.

   [edit protocols mpls]

   ```
   root@R2# set interface ge-0/0/0.0
   ```

4. Configure IS-IS, RSVP, and segment routing .

   [edit protocols rsvp]

   ```
   root@R2# set interface ge-0/0/0.0
   root@R2# set interface lo0.0
   ```

   [edit protocols isis]

   ```
   root@R2# set interface ge-0/0/0.0
   root@R2# set interface lo0.0 passive
   ```

```
root@R2# set source-packet-routing node-segment ipv4-index 402
```

```
root@R2# set traffic-engineering l3-unicast-topology
```

5. Commit the configuration.

```
root@R1# commit
```

**SEE ALSO**

PCEP Configuration

## Verify cPCEPAdaptor Status

1. Verify the gRPC connection and registration status.

```
root@CPCE1-pcepAdaptor# run show path-computation-adaptor pce status
```

```
PCE IP          Port    Type   Status       Elapsed       PCE Name
                                            Time
------          ----    ----   ------       -------       --------
172.17.0.2      50051   gRPC   Connected    00:00:04      cRPD-cpce
```

```
root@CPCE1-pcepAdaptor# run show path-computation-adaptor pce status
```

```
PCE IP          Port    Type   Status       Elapsed       PCE Name
                                            Time
------          ----    ----   ------       -------       --------
172.17.0.2      50051   gRPC   Registered   00:15:52      cRPD-cpce
```

2. Verify path computation client status. This command displays the list of PCC routers that established PCEP connection with the cPCEPAdaptor.

```
root@CPCE1-pcepAdaptor# run show path-computation-adaptor pcc
```

```
PCC IP          Port    Session    Status      Elapsed
                        ID                     Time
------          ----    -------    ------      -------
10.1.255.1      53554   2          SessionUP   01:55:48
```

3. Verify the path computation client lsp information. This command displays the list of LSPs reported from a given PCC connected to the cPCEPAdaptor.

   root@CPCE1-pcepAdaptor# **run show path-computation-adaptor pcc ip 10.1.255.1 lsp**

```
To              From            State     PLSP   LSP    Setup  Control    LSP
                                          ID     ID     Type   Type       Name
--              ----            -----     ----   ---    -----  -------    ----
10.1.255.2      10.1.255.1      ACTIVE    1      4      RSVP   Delegated  to-R2
10.1.255.2      10.1.255.1      ACTIVE    2      0      SR     Delegated  computels1/p1
Total LSPs: 2
```

4. Verify the path computation client lsp information for segment routing.

   root@CPCE1-pcepAdaptor# **run show path-computation-adaptor pcc ip 10.1.255.1 lsp-name computels1/p1**

```
LSP Name: computels1/p1, PLSP ID: 2
 PCC ID: 10.1.255.1
 Source: 10.1.255.1, Destination: 10.1.255.2
 Setup Type: SR
 Control Type: Delegated (SRP ID: 400018)
 LSP ID: 0, Tunnel ID: 2, Ext Tunnel ID: 10.1.255.2
 State: ACTIVE, Since: 0
Lspflags:
  Delegate: 1, Sync: 41, Remove: 0, Administrative: 1
  Operational: 2, Create: 0, P2MP: 0, Fragmentation: 0
Attributes:
  Metric Type: TE
  Priorities (Setup/Hold): 0/0
  Include All: 0, Include Any: 0,
  Exclude All: 0
  Eq Cost Tie Breaker: RANDOM
  Bandwidth: 1036
Computed Path:
  SID: 1048575
  ERO: 10.1.12.1-10.1.12.2
  Metrics: 0
Recorded Route:
  RRO: 10.1.12.1-10.1.12.2
Statistics:
```

```
RROSID: 1048575
Recent Path Request ID: 10025
Recent Path Response ID: 10023
Total Path Computation Requests sent: 2
Total Path Computation Responses received: 1
Total Unsolicited responses received: 0
Average time for computing the path by cPCE: 9 msec
Recent LSP-ID notified to cPCE: 0
Recent LSP-ID received from cPCE: 0
Recent Notified State: Up
```

**SEE ALSO**

Distributed CSPF for Segment Routing LSPs

## Verify BGP on cPCE

1. Verify BGP configuration.

   [edit protocols]

   root@cRPD-cpce# **run show bgp summary**

```
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Unconfigured peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
lsdist.0
                       7          7          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.1.255.1            65000      52365      52348       0       0 2w2d 8:29:55 Establ
  lsdist.0: 7/7/7/0
```

2. Verify LSP configuration.

   root@cRPD-cpce# **run show mpls lsp**

```
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
```

```
10.1.255.2      10.1.255.1      Up     0 *                    10.1.255.1-to-R2
Total 1 displayed, Up 1, Down 0
```

## Verify PCC Status

1. Verify the PCEP session status and LSP summary between the PCC (R1) and the connected PCEs.

   root@R1> **show path-computation-client status**

```
Session              Type                        Provisioning   Status         Uptime
  cPCE1              Stateful Active             On             Up              2161

LSP Summary
  Total number of LSPs        : 1
  Static LSPs                 : 0
  Externally controlled LSPs  : 1
  Externally provisioned LSPs : 0/16000 (current/limit)
  Orphaned LSPs               : 0

cPCE1 (main)
  Delegated              : 1
  Externally provisioned : 0
```

   root@R1> **show mpls lsp ingress**

```
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
10.1.255.2      10.1.255.1      Up     0 *                      to-R2
Total 1 displayed, Up 1, Down 0
```

   root@R1> **show ospf neighbor**

```
Address          Interface              State           ID               Pri  Dead
10.1.12.2        ge-0/0/1.0             Full            10.1.255.2       128    31
```

2. Verify LSP configuration.

   [edit]

root@R1# **run show spring-traffic-engineering lsp**

```
To              State     LSPname
10.1.255.2      Up        computels1


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

root@R1# **run show spring-traffic-engineering lsp detail**

```
Name: computels1
  Tunnel-source: Static configuration
  Tunnel Forward Type: SRMPLS
  To: 10.1.255.2
  Te-group-id: 0
  State: Up
    Path: p1
    Path Status: NA
    Outgoing interface: ge-0/0/1.0
    Delegation info:
      Control-status: Externally controlled
      Routing-status: Externally routed
    Auto-translate status: Disabled Auto-translate result: N/A
    BFD status: N/A BFD name: N/A
    Segment ID : 128
    ERO Valid: true
      SR-ERO hop count: 1
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 10.1.12.1 -> 10.1.12.2
          SID type: 20-bit label, Value: 1048575


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

3

CHAPTER

# Troubleshooting

**IN THIS CHAPTER**


- Debug cPCE and cPCEP Adaptor | **27**

# Debug cPCE and cPCEP Adaptor

IN THIS SECTION


- Troubleshooting Container | 27
- Verify Docker | 28
- Verify Reachability of cPCE Bash Shell | 29
- Verify Sample outputs from cPCE | 29
- Verify Reachability from R1 | 30
- Verify Sample Outputs from cPCEPAdaptor | 32
- Verify Sample Outputs from R1 | 33
- View Trace Files | 34
- Restart cPCEPAdaptor | 35


Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem.

## Troubleshooting Container

You can implement various docker commands to monitor and troubleshoot issues at container level when cPCE is deployed as a docker container.

- `docker ps`: List out active containers and their state.
- `docker stats`: Continuous monitor the resource utilization.
- `docker logs`: Extract container logs in case the container terminates unexpectedly.
- `docker stop`: Stop the Docker from the current state.
- `docker start`: Restart the Docker container.

## Verify Docker

To verify docker details:

1. Verify the installed Docker Engine version by using the `docker version` command.

   root@ubuntu-vm18:~# **docker version**

```
Client: Docker Engine - Community
 Version:           24.0.2
 API version:       1.43
 Go version:        go1.20.4
 Git commit:        cb74dfc
 Built:             Thu May 25 21:52:13 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          24.0.2
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.4
  Git commit:       659604f
  Built:            Thu May 25 21:52:13 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.21
  GitCommit:        3dce8eb055cbb6872793272b4f20ed16117344f8
 runc:
  Version:          1.1.7
  GitCommit:        v1.1.7-0-g860f061
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

2. View the software and hardware information in the system.

   root@ubuntu-vm18:~# **uname -a**

```
Linux ubuntu18-04-3 4.15.0-55-generic #60-Ubuntu SMP Tue Jul 2 18:22:20 UTC 2019 x86_64
x86_64 x86_64 GNU/Linux
```

3. View the version of ubuntu.

   root@ubuntu-vm18:~# **lsb_release -a**

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.3 LTS
Release:        18.04
Codename:       bionic
```

## Verify Reachability of cPCE Bash Shell

To access the cRPD using bash shell:

1. Run the command to launch the Junos shell.

   root@ubuntu-vm18:~# **docker exec -it cRPD-cpce bash**

2. Run the command to ping the host ubuntu VM from cPCE.

   root@cRPD-cpce:/# **ping 172.17.0.1 -c 2**

```
PING 172.17.0.1 (172.17.0.1) 56(84) bytes of data.
64 bytes from 172.17.0.1: icmp_seq=1 ttl=64 time=0.166 ms
64 bytes from 172.17.0.1: icmp_seq=2 ttl=64 time=0.093 ms
--- 172.17.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.093/0.129/0.166/0.036 ms
```

## Verify Sample outputs from cPCE

1. Verify route details.

   root@cRPD-cpce> **show route**

```
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[Static/5] 00:59:38
                    >  to 172.17.0.1 via eth0
172.17.0.0/16      *[Direct/0] 00:59:39
                    >  via eth0
172.17.0.2/32      *[Local/0] 00:59:39
                       Local via eth0
```

2. Verify interfaces.

   root@cRPD-cpce> **show interfaces terse**

```
Interface@link   Oper State     Addresses
erspan0@NONE     DOWN
eth0@if6         UP             172.17.0.2/16 fe80::42:acff:fe11:2/64
gre0@NONE        UNKNOWN
gretap0@NONE     DOWN
ip6tnl0@NONE     UNKNOWN        fe80::a8bf:23ff:fee3:a267/64
irb              UNKNOWN        fe80::9c50:5bff:fe2a:fdb3/64
lo               UNKNOWN        127.0.0.1/8 ::1/128
lo0.0            UNKNOWN        fe80::50c3:f7ff:fe7e:8ab6/64
lsi              UNKNOWN        fe80::9090:eff:fe81:7b7b/64
sit0@NONE        UNKNOWN        ::172.17.0.2/96 ::127.0.0.1/96
tunl0@NONE       UNKNOWN
```

3. Run the command to start shell mode.

   root@cRPD-cpce> **start shell**

4. Run the command to ping R1 from cPCE.

   root@cRPD-cpce# **ping 10.1.255.1 -c 2**

```
PING 10.1.255.1 (10.1.255.1) 56(84) bytes of data.
64 bytes from 10.1.255.1: icmp_seq=1 ttl=63 time=1.33 ms
64 bytes from 10.1.255.1: icmp_seq=2 ttl=63 time=0.975 ms
```

## Verify Reachability from R1

1. Configure host device to connect to R1.

root@ubuntu-vm18:~# **ip link**

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT
group default qlen 1000
    link/ether 54:04:0a:31:b0:fe brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen
1000
    link/ether 56:04:08:00:fb:87 brd ff:ff:ff:ff:ff:ff
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
group default
    link/ether 02:42:09:4f:c5:c2 brd ff:ff:ff:ff:ff:ff
```

root@ubuntu-vm18:~# **ip addr add 10.1.1.2/30 dev eth1**

root@ubuntu-vm18:~# **ip link set eth1 up**

root@ubuntu-vm18:~# **ip route add 10.1.0.0/16 via 10.1.1.1**

root@ubuntu-vm18:~# **ip route**

```
default via 10.49.191.254 dev eth0 proto dhcp src 10.49.176.254 metric 100
10.1.0.0/16 via 10.1.1.1 dev eth1
10.1.1.0/30 dev eth1 proto kernel scope link src 10.1.1.2
10.49.160.0/19 dev eth0 proto kernel scope link src 10.49.176.254
10.49.191.254 dev eth0 proto dhcp scope link src 10.49.176.254 metric 100
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1
```

2. Ping R1 from host device.

root@ubuntu-vm18:~# **ping 10.1.1.1 -c 2**

```
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=2.02 ms

--- 10.1.1.1 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.047/1.535/2.023/0.488 ms
```

## Verify Sample Outputs from cPCEPAdaptor

1. Start shell mode on Adaptor.

   root@CPCE1-pcepAdaptor> **start shell**

   root@CPCE1-pcepAdaptor# **ifconfig**

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.3  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:03  txqueuelen 0  (Ethernet)
        RX packets 11767  bytes 816903 (816.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10957  bytes 867427 (867.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 155010  bytes 15506032 (15.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 155010  bytes 15506032 (15.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. Ping cPCE from Adaptor.

   root@CPCE1-pcepAdaptor# **ping 172.17.0.2 -c 2**

```
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=6.04 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.086 ms

--- 172.17.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.086/3.062/6.038/2.976 ms
```

3. Ping Ubuntu host VM from Adaptor.

root@CPCE1-pcepAdaptor# **ping 172.17.0.1 -c 2**

```
PING 172.17.0.1 (172.17.0.1) 56(84) bytes of data.
64 bytes from 172.17.0.1: icmp_seq=1 ttl=64 time=0.076 ms
64 bytes from 172.17.0.1: icmp_seq=2 ttl=64 time=0.183 ms

--- 172.17.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1013ms
rtt min/avg/max/mdev = 0.076/0.129/0.183/0.053 ms
```

4. Ping Router R1 from Adaptor.

root@CPCE1-pcepAdaptor# **ping 10.1.255.1 -c 2**

```
PING 10.1.255.1 (10.1.255.1) 56(84) bytes of data.
64 bytes from 10.1.255.1: icmp_seq=1 ttl=63 time=0.993 ms
64 bytes from 10.1.255.1: icmp_seq=2 ttl=63 time=1.03 ms

--- 10.1.255.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.993/1.013/1.033/0.020 ms
```

## Verify Sample Outputs from R1

1. Verify that the LSP is up.

root@R1> **show mpls lsp ingress**

```
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
10.1.255.2      10.1.255.1      Up     0 *                      to-R2
Total 1 displayed, Up 1, Down 0
```

2. Verify OSPF neighbor.

```
root@R1> show ospf neighbor
```

```
Address          Interface              State           ID               Pri  Dead
10.1.12.2        ge-0/0/1.0             Full            10.1.255.2       128    31
```

3. Verify BGP summary.

```
root@R1> show bgp summary
```

```
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
lsdist.0
                       0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
172.17.0.2            65000         91         96       0       0       39:17 Establ
  lsdist.0: 0/0/0/0
```

## View Trace Files

1. Configure the gRPC trace options.

```
root@CPCE1-pce# set system services extension-service traceoptions file grpc.log
```

```
root@CPCE1-pce# set system services extension-service traceoptions flag all
```

2. Configure the programmable RPD trace options.

```
root@CPCE1-pce# set routing-options programmable-rpd traceoptions file grpc-server.log
```

```
root@CPCE1-pce# set routing-options programmable-rpd traceoptions flag te-path-compute
```

3. Configure bgp trace options.

```
root@CPCE1-pce# set protocols bgp traceoptions file bgp.log
```

```
root@CPCE1-pce# set protocols bgp traceoptions file size 1m
```

```
root@CPCE1-pce# set protocols bgp traceoptions file world-readable
```

```
root@CPCE1-pce# set protocols bgp traceoptions flag all
```

4. Configure mpls trace options.

```
root@CPCE1-pce# set protocols mpls traceoptions file extctrl.log
root@CPCE1-pce# set protocols mpls traceoptions file size 1m
root@CPCE1-pce# set protocols mpls traceoptions file world-readable
root@CPCE1-pce# set protocols mpls traceoptions flag externally-controlled-lsp
root@CPCE1-pce# set protocols mpls traceoptions flag all
```

## Restart cPCEPAdaptor

Specify the command to restart if the pcepAdaptor gets disconnected.

```
root@CPCE1-pcepAdaptor# run restart pceserver

Pce Server Daemon started, pid 24564
```