



Illumio Core[®]

Version 22.5

PCE Installation and Upgrade Guide

April 2024

20000-200-22.5

Legal Notices

Copyright © 2023 Illumio 920 De Guigne Drive, Sunnyvale, CA 94085. All rights reserved.

The content in this documentation is provided for informational purposes only and is provided "as is," without warranty of any kind, expressed or implied of Illumio. The content in this documentation is subject to change without notice.

Product Version

PCE Version: 22.5

For the complete list of Illumio Core components compatible with Core PCE, see the Illumio Support portal (login required).

For information on Illumio software support for Standard and LTS releases, see [Versions and Releases](#) on the Illumio Support portal.

Resources

Legal information, see <https://www.illumio.com/legal-information>

Trademarks statements, see <https://www.illumio.com/trademarks>

Patent statements, see <https://www.illumio.com/patents>

License statements, see <https://www.illumio.com/eula>

Open source software utilized by the Illumio Core and their licenses, see [Open Source Licensing Disclosures](#)

Contact Information

To contact Illumio, go to <https://www.illumio.com/contact-us>

To contact the Illumio legal team, email us at legal@illumio.com

To contact the Illumio documentation team, email us at doc-feedback@illumio.com

Contents

Chapter 1 Overview of PCE Installation	6
<hr/>	
About This Installation Guide	6
How to Use This Guide	6
Before Reading This Guide	6
Notational Conventions in This Guide	7
PCE Installation Overview	7
Nodes and Clusters	7
Single-node Clusters	8
Software Distribution: PCE and UI Packages	9
Chapter 2 Prepare for PCE Installation	10
<hr/>	
PCE Installation Planning	10
Planning Checklist	10
PCE Capacity Planning	11
CPU, Memory, and Storage	12
Maximum Flow Capacity	17
PCE Storage Device Partitions	17
PCE Storage Device Layout	17
Runtime Parameters for Traffic Datastore on Data Nodes	21
Scale Traffic Database to Multiple Nodes	22
Port Ranges for Cluster Communication	25
Requirements for PCE Installation	26
Load Balancer Requirements	26
PCE IP Address	27
DNS Requirements	27
SMTP Requirements	27
TLS Requirements	27
(Optional) Verify PCE Package Signature	31
(Optional) Configure SAML IdP for User Login	32
OS Setup and Package Dependencies	32
About Your Organization Name and ID	39
Chapter 3 PCE Installation	40
<hr/>	
Install the PCE and UI	40
Download the Software	41
Install the PCE and UI Packages	41

Configure the PCE	42
Set Configuration File Location	42
Run the PCE Setup Script	43
General Configuration	44
Command-line Batch or List Mode	45
Advanced Runtime Environment Parameters	45
Additional Options	46
(Optional) Validate and Configure TLS Certificate	46
Install Certificate	48
Verify the PCE Runtime Environment	49
Start and Initialize the PCE	49
Start the PCE	49
Initialize the PCE	50
VEN Deployment	52
Additional PCE Installation Tasks	52
Configure PCE Backups	52
Internal Syslog and Events Configuration Required	52
(Optional) Configure PCE Internal syslog	53
After PCE Installation	57
RPM Installation Directories	57
RPM Runtime User and Group	58
PCE Control Interface and Other Commands	58
PCE Service Script illumio-pce for Boot	59
PCE Runlevels	60
Alternative: Install the PCE Tarball	60
Process for Installing PCE Tarball	60
Upgrade PCE Tarball Installation	61
Change Tarball to RPM Installation	62
Chapter 4 PCE Upgrade, Downgrade, and Uninstall	64
PCE Upgrade Prerequisites	64
Upgrade Paths and Planning Tool	64
Upgrade Prerequisites	64
Upgrade the PCE	65
Back Up the PCE	66
Download the Software	67
Stop the PCE	67

Install the New PCE and UI	67
Update the Runtime Environment File	68
Migrate the PCE Database	68
Set Runlevel 5	69
Verify Success	70
PCE UI-Only Upgrade	71
Downgrade PCE to Previous Version	71
Downgrade the PCE	71
Verify Success of Downgrade	74
Uninstall the PCE	75
Chapter 5 PCE Installation Reference	76
Reference: PCE Runtime Parameters	76
FIPS Compliance for PCE and VEN	86
FIPS Prerequisites	86
FIPS-related Government and Vendor Documentation	87
Non-Government Customers without FIPS Requirement	87
Compliance Affirmation Letters	87
Prerequisites for Linux VEN FIPS Compliance	87
Prerequisites for Windows VEN FIPS Compliance	88
Enable PCE FIPS Compliance	88
FIPS Compliance for Red Hat/Linux VENS	88
FIPS Compliance for Windows VENS	89
OpenSSL 3.0 Module and RHEL 8 FIPS 140-2 Certification	89
Chapter 6 PCE Installation Troubleshooting	90
PCE Troubleshooting Scenarios	90
Session Limits Too Low	90
Database Migrations Mismatch	91
Database Already Exists	92
PCE UI Missing	92

Overview of PCE Installation

This chapter contains the following topics:

About This Installation Guide	6
PCE Installation Overview	7

Understanding the concepts in this overview will help you achieve a successful PCE installation.

About This Installation Guide

The following sections give useful information to help you get the most out of this guide.

How to Use This Guide

This guide describes how to install the PCE software for the Illumio Core, including details to complete the following tasks:

- Understand important concepts for a successful installation.
- Download and install the software.
- Configure the PCE, including required and optional settings.
- Start and initialize the PCE.
- Understand the next steps that are required for a full Illumio Core installation after the PCE is installed and running.

Before Reading This Guide

Illumio recommends that you be familiar with the following technology:

- Your organization's security goals
- General knowledge of Illumio Core
- General computer system administration of Linux and Windows operating systems, including startup/shutdown, and common processes or services
- Linux shell (bash), Windows PowerShell, or both
- TCP/IP networks, including protocols, well-known ports, and the Domain Name System (DNS)
- Familiarity with TLS/SSL certificates

Notational Conventions in This Guide

- Newly introduced terminology is italicized. Example: *activation code* (also known as pairing key)
- Command-line examples are monospace. Example: `illumio-ven-ctl --activate`
- Arguments on command lines are monospace italics. Example: `illumio-ven-ctl -
-activate activation_code`
- In some examples, the output might be shown across several lines but is actually on one single line.
- Command input or output lines not essential to an example are sometimes omitted, as indicated by three periods in a row. Example:

```
...  
some command or command output  
...
```

PCE Installation Overview

This overview introduces some essential concepts that you'll need to understand before installing the PCE.

Nodes and Clusters

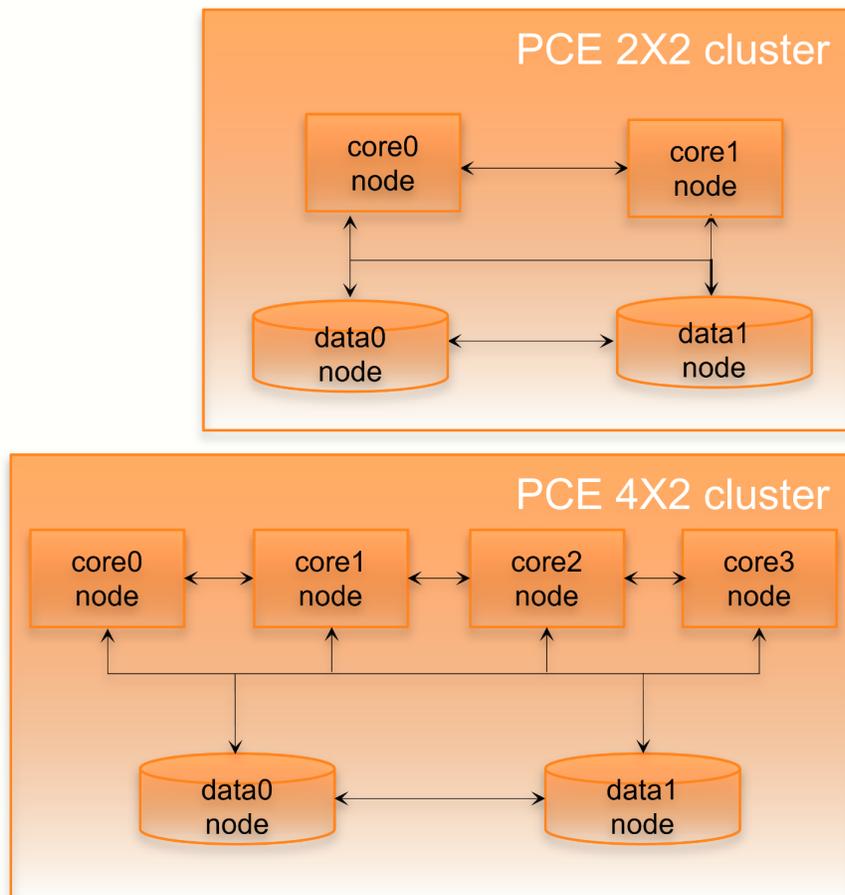
A *PCE node* is a single host (server or VM) that runs the PCE. Each node in the cluster is configured by its node type, which defines its services:

- Core node, known as `core0`, `core1`, `core2`, and `core3`
- Data node, known as `data0` and `data1`
- Single node in an *single-node cluster* (SNC), which combines core and data nodes in one

The total collection of nodes is a *PCE cluster*. In production, the PCE is typically deployed as a *multiple-node cluster* (MNC).

- For smaller deployments where high availability is not necessary, you can deploy a PCE SNC.
- In a typical PCE deployment, for redundancy, you deploy two instances of each node type in a *PCE 2x2 cluster*.
- For larger deployments, you can expand the PCE cluster to four core nodes and two data nodes in a *PCE 4x2 cluster*.
- To construct a single administrative domain that spans two or more replicating PCE clusters, deploy a *PCE supercluster*. See [PCE Supercluster Deployment](#).

PCE Multi-node Cluster Types



Single-node Clusters

In an SNC, some special considerations apply.

Because it contains only a single node, an SNC does not provide high availability (HA) features. The SNC is a single point of failure. Therefore, Illumio recommends taking some additional precautionary steps:

- Set up periodic, automated backups.
- Practice restoring from backup to a separate machine (physical or virtual) before putting the SNC into production use.
- Store a copy of the PCE software installation packages, the PCE database backup, and the `runtime_env.yml` file, which stores the PCE's configuration. Store them on a separate physical machine, preferably in a different datacenter, using fault tolerant storage.
- If you are running the SNC as a virtual machine, you can make use of the hypervisor's high availability and disaster recovery (HA/DR) features.

To prepare for PCE installation on an SNC:

- Have a reserved virtual machine or physical machine ready for the backups of the PCE software, database, and `runtime_env.yml`.
- This machine must be able to use the existing IP address of the PCE. Alternatively, you can reserve a new IP address for the backup machine, and configure this IP address in the PCE.

Software Distribution: PCE and UI Packages

Illumio distributes PCE software as two packages: PCE and UI. The PCE package contains the software for the Policy Compute Engine (PCE), and the UI package contains the PCE web console. You can choose to install these packages separately or together:

- **PCE package plus UI package:** This choice is the most common installation scenario. See [Installing the PCE and UI](#).
- **PCE package alone:** The PCE still serves responses to API calls, but there is no graphical user interface for display in a browser.
- **UI package alone:** With this separate package, you can upgrade the UI whenever you want more recent UI fixes and features, without having to upgrade the entire PCE. The UI-only installation procedure is much simpler than the full installation. For the UI to work, a compatible version of the PCE must already be installed. See [UI-Only Upgrade](#).

Prepare for PCE Installation

This chapter contains the following topics:

PCE Installation Planning	10
PCE Capacity Planning	11
PCE Storage Device Partitions	17
Port Ranges for Cluster Communication	25
Requirements for PCE Installation	26

Before installing the PCE software, be sure to fulfill the prerequisites and do the setup steps in this section.

PCE Installation Planning

This section describes the decisions you must make and the preparatory tasks you must do before installing the PCE.

Planning Checklist

The following checklist helps you to plan your PCE installation. Details for each task are described in later sections.

Prerequisite	See section...
Capacity sizing for CPUs, RAM, and storage device size and IOPS	PCE Capacity Planning
PCE storage device partitions	PCE Storage Device Partitions
Verify PCE reserved port ranges (for MNCs; does not apply in an SNC)	Port Ranges for Cluster Communication

Prerequisite	See section...
Load balancer setup	Load Balancer Requirements
IP address for the PCE	PCE IP Address
DNS domain name setup	DNS Requirements
Mail software	SMTP Requirements
TLS setup, including SSL certificate types and settings	<ul style="list-style-type: none"> • TLS Requirements • TLS Versions for Communications • (Optional) Validate and Configure TLS Certificate either before or after configuring the PCE
(Optional) SAML IdP	SAML IdP
OS package dependencies, libraries, NTP, iptables, UTF-8, Trusted CA, syslog, process and file limits, and kernel parameters	OS Setup and Package Dependencies
Your full organization name	About Your Organization Name and ID
VEN installation, including planning and prerequisites	<i>VEN Installation and Upgrade Guide</i>

PCE Capacity Planning

Use these guidelines and requirements to estimate host system capacity based on typical usage patterns.

The exact requirements vary based on a large number of factors, including, but not limited to:

- Whether you are using physical or virtual hardware
- Number of managed workloads
- Number of unmanaged workloads and other labeled objects, such as virtual services
- Policy complexity, which includes the following factors:
 - Number of rules in your rulesets
 - Number of labels, IP lists, and other objects in your rules
 - Number of IP ranges in your IP lists
 - Number of workloads affected by your rules

- Frequency at which your policies change
 - Frequency at which workloads are added or deleted, or workload context changes, such as, change of IP address
 - Volume of traffic flows per second reported to the PCE from all VENS
- See the “Maximum Flow Capacity” table for information about maximum flow capacity of the PCE.
- Total number of unique flows reported to the PCE from all VENS

CPU, Memory, and Storage

The capacity planning tables in this section list the minimum recommended sizes for CPU, memory, and storage. This section provides two tables, one for physical hardware and one for virtual machines. Use these tables to plan your deployment.

NOTE:

Based on your actual usage and other factors, your capacity needs might be greater than the recommended sizes. For example, if you have installed additional software along with the PCE, such as application performance management (APM) software or an endpoint protection agent, this consumes additional system resources.

Data nodes are configured with a dedicated storage device for each database on the data nodes. This configuration accommodates growth in traffic data, which is used by Explorer. See [Runtime Parameters for Traffic Datastore on Data Nodes](#).

For more than 150 IOPS, locally attached, spinning hard disk drives (HDD) are not sufficient. You will require either mixed-use Solid-State Disk (SSD) or Storage Area Network (SAN).

The PCE does not require that you set up swap memory, but it is permissible to enable swap memory. As long as the PCE nodes are provisioned with the recommended memory (RAM) as shown in the tables below, the use of swap memory should not cause any issues.

Physical Hardware

Use this table if you are installing the PCE on physical hardware. If you are using virtual machines, see the table [Virtual Hardware](#).

MNC Type + Work-loads/VENs	Cores/Clock Speed	RAM per Node	Storage Device Size and IOPS	
			Core Nodes	Data Nodes
SNC <ul style="list-style-type: none"> • 250 VENs¹ • 2500 work-loads 	<ul style="list-style-type: none"> • 3 cores² • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or equivalent 	16GB	A single node including both core and data: <ul style="list-style-type: none"> • 1 x 50GB⁴ • 100 IOPS per device⁵ 	N/A
2x2 Small <ul style="list-style-type: none"> • 2,500 VENs¹ • 12,500 work-loads Cluster type: 4node_v0_small	<ul style="list-style-type: none"> • 4 cores per node² • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or equivalent 	32GB	Minimum: <ul style="list-style-type: none"> • Disk: 50GB^{3, 4} • 150 IOPS per device⁵ 	Minimum: <ul style="list-style-type: none"> • Disk 1: 250GB⁴ • Disk 2: 250GB⁴ • 600 IOPS per device⁵
2x2 <ul style="list-style-type: none"> • 10,000 VENs¹ • 50,000 work-loads Cluster type: 4node_v0 OR 4node_dx	<ul style="list-style-type: none"> • 16 cores per node^{2, 6} • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or equivalent 	<ul style="list-style-type: none"> • Recommended: 128GB⁶ • Minimum: 64GB 	Minimum: <ul style="list-style-type: none"> • Disk: 50GB^{3, 4} • 150 IOPS per device⁵ 	Minimum: <ul style="list-style-type: none"> • Disk 1: 1TB⁴ • Disk 2: 1TB⁴ • 1,800 IOPS per device⁵
4x2 <ul style="list-style-type: none"> • 25,000 VENs¹ • 125,000 work- 	<ul style="list-style-type: none"> • 16 cores per node^{2, 6} • Intel® Xeon(R) CPU E5-2695 	128GB ⁶	Minimum: <ul style="list-style-type: none"> • Disk: 50GB^{3, 4} 	<ul style="list-style-type: none"> • Disk 1: 1TB⁴ • Disk 2: 1TB⁴

MNC Type + Workloads/VENs	Cores/Clock Speed	RAM per Node	Storage Device Size and IOPS	
			Core Nodes	Data Nodes
loads Cluster type: 6node_v0 OR 6node_dx	v4 at 2.10GHz or equivalent.		<ul style="list-style-type: none"> 150 IOPS per device⁵ 	<ul style="list-style-type: none"> 5,000 IOPS per device⁵

Footnotes:

¹ Number of VENs/workloads is the sum of both the number of managed VENs and the number of unmanaged workloads.

² CPUs:

- The recommended number of cores is based only on physical cores from allocated CPUs, irrespective of hyper-threading.

³ This is the absolute minimum needed. In the future, other applications, support reports, or new features may require additional disk.

⁴ Additional disk notes:

- Storage requirements for network traffic data can increase rapidly as the amount of network traffic increases.
- Network File Systems (NFS) is not supported for Illumio directories specified in runtime; for example, `data_dir`, `persistent_data_dir`, `ephemeral_data_dir`.

⁵ Input/output operations per second (IOPS) are based on 8K random write operations. IOPS specified for an average of 300 flow summaries (80% unique `src_ip`, `dest_ip`, `dest_port`, `proto`) per workload every 10 minutes. Different traffic profiles might require higher IOPS.

⁶ In the case of fresh installs or upgrades of a 2x2 for 10,000 VENs or a 4x2 for 25,000 VENs, if you deploy a system without sufficient cores, memory, or both, then the PCE will automatically reduce the object limits to 2,500 workloads. Object limit is the number of VENs (agents) per PCE. Adding more than 2,500 workloads will fail and an event is logged indicating that object limits have been exceeded. The workaround is to increase the number of cores, memory, or both to the recommended specifications and then increase the object limits manually. See [PCE Default Object Limits](#) in the *PCE Administration Guide*.

Virtual Hardware

Use this table if you are installing the PCE on virtual machines. If you are using physical hardware, see the table [Physical Hardware](#).

MNC Type + Workloads/VENs	Virtual Cores/Clock Speed	RAM per Node	Storage Device Size and IOPS	
			Core Nodes	Data Nodes
SNC <ul style="list-style-type: none"> • 250 VENs¹ • 2500 workloads 	<ul style="list-style-type: none"> • 6 virtual cores (vCPU)² • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or higher 	16GB ⁷	Minimum: <ul style="list-style-type: none"> • Disk: 50GB^{3, 4} • 150 IOPS per device⁵ 	N/A
2x2 Small <ul style="list-style-type: none"> • 2,500 VENs¹ • 12,500 workloads Cluster type: 4node_v0_small	<ul style="list-style-type: none"> • 8 virtual cores (vCPU) per node² • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or higher 	32GB ⁷	Minimum: <ul style="list-style-type: none"> • Disk: 50GB^{3, 4} • 150 IOPS per device⁵ 	Minimum: <ul style="list-style-type: none"> • Disk 1: 250GB • Disk 2: 250GB • 600 IOPS per device
2x2 <ul style="list-style-type: none"> • 10,000 VENs¹ • 50,000 workloads Cluster type: 4node_v0 or 4node_dx	<ul style="list-style-type: none"> • 32 virtual cores (vCPU) per node^{2, 6} • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or higher 	<ul style="list-style-type: none"> • Recommended: 128GB^{6, 7} • Minimum: 64GB 	Minimum: <ul style="list-style-type: none"> • Disk: 50GB^{3, 4} • 150 IOPS per device⁵ 	Minimum: <ul style="list-style-type: none"> • Disk 1: 1TB⁴ • Disk 2: 1TB⁴ • 1,800 IOPS per device⁵

MNC Type + Workloads/VENs	Virtual Cores/Clock Speed	RAM per Node	Storage Device Size and IOPS	
			Core Nodes	Data Nodes
4x2 <ul style="list-style-type: none"> • 25,000 VENs¹ • 125,000 workloads Cluster type: 6node_v0 or 6node_dx	<ul style="list-style-type: none"> • 32 virtual cores (vCPU) per node², 6 • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or higher 	128GB ^{6, 7}	Minimum: <ul style="list-style-type: none"> • Disk: 50GB^{3, 4} • 150 IOPS per device⁵ 	<ul style="list-style-type: none"> • Disk 1: 1TB⁴ • Disk 2: 1TB⁴ • 5,000 IOPS per device⁵

Footnotes:

¹ Number of VENs/workloads is the sum of both the number of managed VENs and the number of unmanaged workloads.

² Full reservations for vCPU. No overcommit.

³ This is the absolute minimum needed. In the future, other applications, support reports, or new features may require additional disk.

⁴ Additional disk notes:

- Storage requirements for network traffic data can increase rapidly as the amount of network traffic increases.
- Network File Systems (NFS) is not supported for Illumio directories specified in runtime; for example, data_dir, persistent_data_dir, ephemeral_data_dir.

⁵ Input/output operations per second (IOPS) are based on 8K random write operations. IOPS specified for an average of 300 flow summaries (80% unique src_ip, dest_ip, dest_port, proto) per workload every 10 minutes. Different traffic profiles might require higher IOPS.

⁶ In the case of fresh installs or upgrades of a 2x2 for 10,000 VENs or a 4x2 for 25,000 VENs, if you deploy a system without sufficient cores, memory, or both, then the PCE will automatically reduce the object limits to 2,500 workloads. Object limit is the number of VENs (agents) per PCE. Adding more than 2,500 workloads will fail and an event is logged indicating that object limits have been exceeded. The workaround is

to increase the number of cores, memory, or both to the recommended specifications and then increase the object limits manually. See [PCE Default Object Limits](#) in the *PCE Administration Guide*.

⁷ Full reservations for vRAM. No overcommit.

Maximum Flow Capacity

The following table shows the maximum capacity of the PCE to accept flow data from all VENS.

MNC Type + Workloads/VENs	Flow Rate (flow-summaries/second)	Equivalent Flow Rate (flows/second) ²
SNC <ul style="list-style-type: none"> • 250 VENS • 2500 workloads 	100	1,030
2x2 <ul style="list-style-type: none"> • 2,500 VENS • 12,500 workloads 	1,000	10,300
2x2 <ul style="list-style-type: none"> • 10,000 VENS • 50,000 workloads 	4,100	422,000
4x2 <ul style="list-style-type: none"> • 25,000 VENS • 125,000 workloads 	10,400 ¹	1,070,000

Footnotes:

¹ The PCE might need to be tuned to achieve this rate. If you need to tune the PCE, please contact Illumio Support for assistance.

² Real-world observation shows that 102 flows result in one flow summary on average.

PCE Storage Device Partitions

PCE Storage Device Layout

You should create separate storage device partitions to reserve the amount of space specified below. These recommendations are based on [PCE Capacity Planning](#).

The values given in these recommendation tables are guidelines based on testing in Illumio’s labs. If you wish to deviate from these recommendations based on your own platform standards, please first contact your Illumio support representative for advice and approval.

PCE Single-Node Cluster for 250 VENS

Storage Device	Partition mount point	Size to Allocate	Node Types	Notes
Device 1, Partition A	/	8GB	Core, Data	The size of this partition assumes the system temporary files are stored in /tmp and core dump file size is set to zero. The PCE installation occupies approximately 500MB of this space.
Device 1, Partition B	/var/log	16GB	Core, Data	The size of this partition assumes that PCE application logs and system logs are both stored in /var/log. PCE application logs are stored in the /var/log/illumio-pce directory. The recommended size assumes average use by the OS with common packages installed and logging levels set to system defaults. Log size limits are configurable, so your system may require more or less log space. To find the potential maximum disk space required for your logs, use this command: <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <pre>\$ sudo -u ilo-pce illumio-pce-env logs --diag</pre> </div>
Device 1, Partition C	/var/lib/illumio-pce	Balance of Device 1	Core, Data	The size of this partition assumes that Core nodes use local storage for application code in /var/lib/illumio-pce, and also assumes that PCE support report files, and other temporary (ephemeral) files, etc., are stored in /var/lib/illumio-pce/tmp.

PCE 2x2 Multi-Node Cluster for 2,500 VENs

Storage Device	Partition mount point	Size to Allocate	Node Types	Notes
Device 1, Partition A	/	16GB	Core, Data	The size of this partition assumes the system temporary files are stored in /tmp and core dump file size is set to zero.
Device 1, Partition B	/var/log	32GB	Core, Data	The size of this partition assumes that PCE application logs and system logs are both stored in /var/log. PCE application logs are stored in the /var/log/illumio-pce directory.
Device 1, Partition C	/var/lib/illumio-pce	Balance of Device 1	Core, Data	The size of this partition assumes that Core nodes use local storage for application code in /var/lib/illumio-pce, and also assumes that PCE support report files, and other temporary (ephemeral) files, etc. are stored in /var/lib/illumio-pce/tmp.
Device 2, Single partition. Applicable in a two-storage-device configuration	/var/lib/illumio-pce/data/Explorer	All of Device 2 (250GB)	Data	For network traffic data in a two-storage-device configuration for the data nodes, it should be a separate device that is mounted on this directory. Set the runtime_env.yml to data_dir: /var/lib/illumio-pce/data/Explorer, which will automatically create a sub-directory called /var/lib/illumio-pce/data/Explorer/traffic_datastore

Storage Device	Partition mount point	Size to Allocate	Node Types	Notes
				The partition mount point and the runtime setting must match. If you customize the mount point, make sure that you also change the runtime setting accordingly.

PCE 2x2 Multi-Node Cluster for 10,000 VENs and

PCE 4x2 Multi-Node Cluster for 25,000 VENs

Storage Device	Partition mount point	Size to Allocate	Node Types	Notes
Device 1, Partition A	/	16GB	Core, Data	The size of this partition assumes the system temporary files are stored in /tmp and core dump file size is set to zero.
Device 1, Partition B	/var/log	32GB	Core, Data	The size of this partition assumes that PCE application logs and system logs are both stored in /var/log. PCE application logs are stored in the /var/log/illumio-pce directory.
Device 1, Partition C	/var/lib/illumio-pce	Balance of Device 1	Core, Data	The size of this partition assumes that Core nodes use local storage for application code in /var/lib/illumio-pce, and also assumes that PCE support report files, and other temporary (ephemeral) files, etc. are stored in /var/lib/illumio-pce/tmp.
Device 2, Single Partition Applicable in	/var/lib/illumio-pce/data/traffic	All of Device 2 (1TB)	Data	For network traffic data in a two-storage-device configuration for the data nodes, it should be a separate device

Storage Device	Partition mount point	Size to Allocate	Node Types	Notes
a two-storage-device configuration				<p>that is mounted on this directory.</p> <p>In <code>runtime_env.yml</code>, set the <code>traffic_datastore : data_dir</code> parameter to match the value of the partition mount point (see previous column) as follows: <code>traffic_datastore: data_dir: /var/lib/illumio-pce/data/traffic</code>.</p> <p>The partition mount point and the runtime setting must match. If you customize the mount point, make sure that you also change the runtime setting accordingly.</p>

Runtime Parameters for Traffic Datastore on Data Nodes

For the traffic datastore, set the following parameters in `runtime_env.yml`:

`traffic_datastore:`

`data_dir: path_to_second_disk` (e.g. `/var/lib/illumio-pce/data/traffic`)

`max_disk_usage_gb`: Set this parameter according to the table below.

`partition_fraction`: Set this parameter according to the table below.

The recommended values for the above parameters, based on PCE node cluster type and estimated number of workloads (VENs), are as follows:

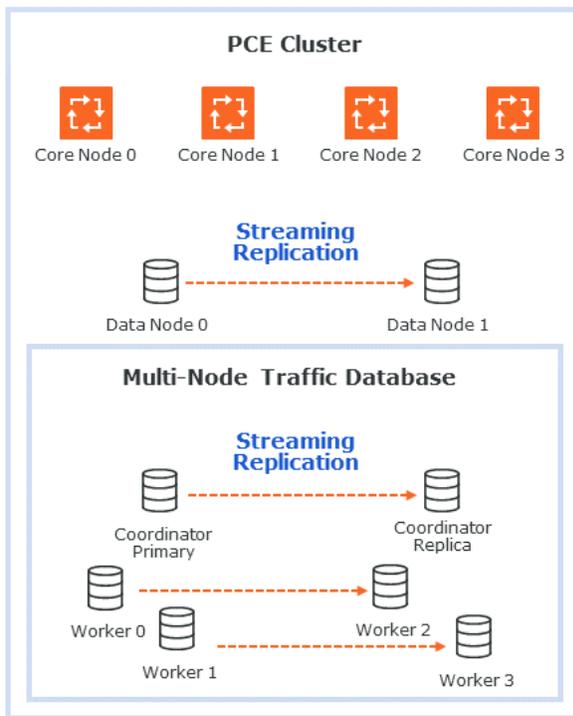
Setting	2x2 2,500 VENs	2x2 10,000 VENs	4x2 25,000 VENs	Note
<code>traffic_datastore:max_disk_usage_gb</code>	100 GB	400 GB	400 GB	<p>This size reflects only part of the required total size, as detailed in PCE Capacity Planning.</p> <p>The remaining disk capacity is needed for database internals and</p>

Setting	2x2 2,500 VENs	2x2 10,000 VENs	4x2 25,000 VENs	Note
traffic_data-store:partition_fraction	0.5	0.5	0.5	data migration during upgrades.

For additional ways to avoid disk capacity issues, see [Manage Data and Disk Capacity](#) in the *PCE Administration Guide*.

Scale Traffic Database to Multiple Nodes

When deploying the PCE, you can scale traffic data by sharding it across multiple PCE data nodes. In this way, you can store more data and improve the performance of read and write operations on traffic data. The traffic database is sharded by setting up two coordinator nodes, each of which has at least one pair of worker nodes.



Hardware Requirements for Multi-Node Traffic Database

The following table shows the minimum required resources for a multi-node traffic database.

CPU	RAM	Storage	IOPS
16 vCPU	128GB	1TB	5,000

Cluster Types for Multi-Node Traffic Database

The following PCE cluster types support scaling the traffic database to multiple nodes:

- `4node_dx` - 2x2 PCE with multi-node traffic database. The 2x2 numbers do not include the coordinator and worker nodes.
- `6node_dx` - 4x2 PCE with multi-node traffic database. The 4x2 numbers do not include the coordinator and worker nodes.

Node Types for Multi-Node Traffic Database

The following PCE node types support scaling the traffic database to multiple nodes:

- `citus_coordinator` - The sharding module communicates with the PCE through the coordinator node. There must be two (2) coordinator nodes in the PCE cluster. The two nodes provide high availability. If one node goes down, the other takes over.
- `citus_worker` - The PCE cluster can have any even number of worker nodes, as long as there are at least two (2) pairs. As with the coordinator nodes, the worker node pairs provide high availability.

Runtime Parameters for Multi-Node Traffic Database

The following runtime parameters in `runtime_env.yml` support scaling the traffic database to multiple nodes:

- `traffic_datastore:num_worker_nodes` - Number of traffic database worker node pairs. The worker nodes must be added to the PCE cluster in sets of two. This supports high availability (HA). For example, if there are 4 worker nodes, `num_worker_nodes` is 2.
- `node_type` - This runtime parameter can be assigned one of the values `citus_coordinator` and `citus_worker`. They are used to configure coordinator and worker nodes.
- `datacenter` - In a multi-datacenter deployment, the value of this parameter tells which datacenter the node is in. The value is any desired descriptive name, such as "west" and "east."

Set Up a Multi-Node Database

When setting up a new PCE cluster with a multi-node traffic database, use the same installation steps as usual, with the following additions.

- Install the PCE software on core, data, coordinator, and worker nodes, using the same version of the PCE on all nodes.
- There must be exactly two (2) coordinator nodes. There must be two (2) or more pairs of worker nodes.
- Set up the `runtime_env.yml` configuration on every node as follows. For examples, see [Example Configurations for Multi-Node Traffic Database](#).
 - Set the cluster type to `4node_dx` for a 2x2 PCE or `6node_dx` for a 4x2 PCE.
 - In the `traffic_datastore` section, set `num_worker_nodes` to the number of worker node pairs. For example, if the PCE cluster has 4 worker nodes, set this parameter to 2.
 - On each coordinator node, in addition to the settings already described, set `node_type` to `citus_coordinator`.
 - On each worker node, in addition to the settings already described, set `node_type` to `citus_worker`.
 - If you are using a split-datacenter deployment, set the `datacenter` parameter on each node to an arbitrary value that indicates what part of the datacenter the node is in.

For installation steps, see [Install the PCE and UI](#) for a new PCE, or [Upgrade the PCE](#) for an existing PCE.

Example Configurations for Multi-Node Traffic Database

Following is a sample configuration for a coordinator node. This node is in a 4x2 PCE cluster (not counting the coordinator and worker nodes) with two pairs of worker nodes:

```
cluster_type: 6node_dx
node_type: citus_coordinator
traffic_datastore:
  num_worker_nodes: 2
```

Following is a sample configuration for a worker node. This node is in a 4x2 PCE cluster (not counting the coordinator and worker nodes) with two pairs of worker nodes:

```
cluster_type: 6node_dx
node_type: citus_worker
traffic_datastore:
  num_worker_nodes: 2
```

Following is a sample configuration for a split-datacenter configuration.

The following settings are for nodes on the left side of the datacenter:

```
cluster_type: 6node_dx
traffic_datastore:
  num_worker_nodes: 2
datacenter: left
```

The following settings are for nodes on the right side of the datacenter:

```
cluster_type: 6node_dx
traffic_datastore:
  num_worker_nodes: 2
datacenter: right
```

Port Ranges for Cluster Communication

(For MNC; does not apply in SNC.)

The following ports or port ranges are needed for communications between the PCE cluster nodes.

Protocols	Ports or Port Range
TCP	3100 to 3600
TCP	5100 to 6300
TCP	8443

Protocols	Ports or Port Range
	<p>IMPORTANT: When using an SLB for load balancing or your PCE core nodes use DNS load balancing on port 8443 TCP, your PCE cluster must be able to access the PCE VIP. This requirement also applies when you have defined any custom port for <code>front_end_https_management_port</code> in the PCE runtime settings.</p>
TCP and UDP	8000 to 8400
TCP	11200 to 11300
TCP and UDP	24200 to 25300

Requirements for PCE Installation

Before installing the PCE, be sure your underlying systems are sufficient to successfully install and run the PCE. Check all the following system requirements.

Load Balancer Requirements

For MNC; does not apply in a SNC.

A server load balancer or DNS-level load balancer is required to distribute traffic to the PCE core nodes.

Configure the load balancer to use the Illumio REST API to monitor which cluster core nodes are available to receive requests. See the *REST API Developer Guide* for exact usage.

```
GET [api_version]/node_available
```

No authentication is required to call this API. An HTTP status code of 200 means the node is able to receive requests. Any other status code or no response means the node is unable accept requests. Unhealthy or unresponsive nodes should be removed from the load balancing pool.

- The PCE Health Check API can experience up to a 30-second delay to return the actual status of the node.
- In the 4x2 configuration, a maximum of two core nodes are available (return a status code of 200) at any time.

- When using a DNS load balancer, it should only serve IP addresses for the cluster FQDN of those nodes that respond with a 200 to the `/node_available` API. For rapid failover in the event of a core node failure, Illumio recommends a DNS TTL of between 30 and 60 seconds.

PCE IP Address

Illumio recommends a statically-assigned IP address. By default, the PCE automatically uses the first available private IP address on the node. The PCE does not automatically bind to a public IP address.

When you use a public IP address or the node has multiple interfaces, you need to configure the PCE with the interface you want to use. To do so, set `internal_service_ip` in the configuration file `runtime_env.yml`. For example:

```
internal_service_ip: 10.2.8.89
```

To configure networking, see your OS vendor's documentation on the `ifcfg-ethN` script.

DNS Requirements

Your Domain Name System (DNS) must resolve the PCE's fully qualified domain name (FQDN). The FQDN must be resolvable on all managed workloads, on all nodes in the PCE cluster, and for all users of the PCE web console and REST API.

If you are using DNS-level load balancing, the PCE FQDN should resolve to the IP addresses of the core nodes. If you are using a server load balancer, the PCE FQDN should resolve to the VIPs of the server load balancer.

SMTP Requirements

An SMTP relay is required to send user invitations and "forgot password" email replies from the PCE.

The SMTP configuration parameter during PCE installation is `smtp_relay_address`. Allowable values are either an IP address with its SMTP port (default 587) or a resolvable FQDN with the SMTP port.

TLS Requirements

PCE communication is secured using the Transport Layer Security (TLS) protocol, the successor to the deprecated Secure Sockets Layer (SSL) protocol. TLS is used for securing the following communication sessions:

- User access to the PCE web console and REST API over the HTTPS protocol.
- Communication between the PCE and VENs.

VEN-to-PCE communications for the EventService (default is port 8444) are secured by the ECDHE suite of cryptographic ciphers, which use an elliptic curve Diffie-Hellman key exchange. This exchange is signed with RSA signature algorithms.

- Communication between PCE nodes in a multi-node cluster.

If you want to generate a temporary, self-signed certificate, see [Understanding Illumio Trial Certificates](#) in the Knowledge Base (log in required).

For an in-depth discussion of deploying the PCE with TLS, see [Preparing Certificates for a PCE deployment](#) in the Knowledge Base (log in required).

X.509 Certificate

An X.509 server certificate must be installed on each PCE node during installation. When any client (the VEN) opens a TLS session to the PCE (for example, pairing a workload, accessing the PCE web console, retrieving updated policy), the PCE presents the server certificate to secure the communication. The server certificate is uploaded as part of a certificate bundle that contains the server certificate and the chain of CA certificates (Intermediate or Root) to establish the chain of trust back to a Root CA.

CAUTION:

The client must be able to validate the chain of trust back to the Root CA for this certificate; otherwise, the TLS handshake fails. You might need to add all the certificates in the chain of trust to the keychain of the client.

The certificate package for the Illumio PCE must meet the following basic criteria:

- The file must contain PEM-encoded certificates.
- The subject value and issuer of the certificate must start with a leading slash character (/).
- As a best practice, duplicate the subject in the Subject Alternative Name (subjectAltName).
- The certificate's signature algorithm must be SHA256WithRSAEncryption.
- The certificate's signature algorithm must *not* be RSASSA-PSS.
- The file must contain the server certificate and the entire certificate chain necessary to establish the chain of trust back to a Root CA.

- a. The package must include all of the CA certificates (Intermediate and/or Root) needed to establish the chain of trust back to a Root CA.
 - If the certificate is generated by a Private CA, all certificates in the chain of trust back to the Root CA must be included. This includes the Root CA certificate and any applicable Intermediate CA certificates.
 - If the certificate is generated by a major Public CA (such as, VeriSign, GeoTrust, Entrust, or Thawte), any Intermediate CA certificates needed to establish the chain of trust back to the Public Root CA must be included.
- b. Pay careful attention to the order of the certificates in the bundle. The server certificate must be first. If you have an Apache-style bundle generated by a standard certificate request process, you need to open the file in a text editor and reverse the order of the certificates. Apache always expects the root certificate to come first, then any intermediates in order (from the root down), and the server certificate is last. The PCE uses nginx, which expects the opposite order. For additional details, see the [Nginx documentation](#).

The certificate bundle should look something like this:

```
-----BEGIN CERTIFICATE-----  
<server cert goes here>  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
<intermediate CA cert goes here>  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
<root CA cert goes here>  
-----END CERTIFICATE-----
```

- All certificates in the bundle must be valid for the current date, which depends on the system time being set correctly.
- A trusted root store must be available for OpenSSL to validate certificates.
- The certificate must match the PCE FQDN, which can be an exact match (for example, pce.mycompany.com) or a wildcard match (for example, *.mycompany.com)

The certificate must support both Server and Client authentication. Client authentication is used between nodes in an MNC. Run the following command and verify TLS

Web Server Authentication, TLS Web Client Authentication appears within the X509v3 Extended Key Usage section.

```
$ openssl x509 -text -noout -in pce.mycompany.com.bundle.crt
...
X509v3 Extended Key Usage:
    TLS Web Server Authentication, TLS Web Client Authentication
...
```

RSASSA-PSS Signature Algorithm Not Supported

The certificate signature algorithm RSASSA-PSS, which is based on PKCS 1 version 2.1, is not supported, because it cannot be validated. This limitation is a widely known problem with this signature algorithm.

The PCE certificate requires the SHA256WithRSEncryption signature.

CAUTION:

If you use Microsoft Certificate Authority (CA) to sign PCE certificates, make sure to use the SHA256WithRSEncryption. PKCS#1 version 2.1 is enabled by default on Microsoft CAs and produces the unsupported RSASSA-PSS signature algorithm.

Private Keys

The private key that matches the X.509 certificate must be installed on each PCE node during installation, and the following guidelines must be met:

- The private key must be PEM-encoded.
- The file must not be encoded.
- The file must not be password protected.

TLS Versions for Communications

The PCE uses Transport Layer Security (TLS) version 1.2 by default for VEN-to-PCE communications, the PCE's web server for the PCE web console, and the REST API.

- The PCE default minimum version is TLS 1.2.
- For VEN versions 18.1 and later, all VENs use TLS 1.2.
- Older operating systems might not support TLS 1.2. For example, SUSE VEN version 17.1.x, a legacy VEN version that is no longer supported by Illumio, requires minimum version TLS 1.0.

- Windows Server 2008 R2 SP1: The HTTP Client library, WinHttp, does not have the necessary API to limit SSL negotiation only to TLS 1.2. This must be configured through the Registry. See the Microsoft Support article “[Update to enable TLS 1.1 and TLS 1.2.](#)”

Changing Minimum TLS version

The default minimum TLS version is TLS 1.2. The minimum TLS version is configurable, but it is recommended that you leave the setting at its default of 1.2. Earlier TLS versions, such as 1.0 and 1.1, are considered less secure than 1.2, so it is recommended you do not use them. In some rare circumstances, you might need to change the minimum TLS version, such as when using older VEN operating systems that do not support TLS 1.2. In addition, you should verify that any browser you use is capable of negotiating the minimum version you set.

If you want to change the minimum TLS version, edit the following parameter in `runtime_env.yml`:

```
min_tls_version
```

The value of `min_tls_version` configures the PCE front end ports in `runtime_env.yml`:

- `front_end_https_port` (default 8443)
- `front_end_https_management_port` (defaults to `front_end_https_port`)
- `front_end_event_service_port` (default 8444)

Allowable values:

- `tls1_0` allows TLS 1.0, 1.1, and 1.2.
- `tls1_1` allows TLS 1.1 and 1.2.
- `tls1_2` allows TLS 1.2.

(Optional) Verify PCE Package Signature

For additional security, verify the identity of the downloaded PCE packages against the Illumio public key. The public key is available in the file `illumio_pce_pub.key`, which ships with the packages.

For information about using a public key to verify package signatures, see [Checking a Package's Signature](#) on the Red Hat Customer Portal.

(Optional) Configure SAML IdP for User Login

After installation, you can configure the PCE to rely on an external, third-party SAML identity provider system. See [Single Sign-on Configuration](#) in the *PCE Administration Guide*. The guide has step-by-step details for a wide variety of IdPs.

OS Setup and Package Dependencies

For information, see [PCE OS Support and Package Dependencies](#) on the Illumio Support portal.

NTP

Set up a Network Time Protocol (NTP) client for time synchronization. It is recommended that you use `chrony`, although `ntpd` can also be used. On RHEL8, `chrony` is the default.

To install and configure the NTP client, use the procedure in the documentation for the client on your operating system.

After you finish installing the PCE, you can use the following command to verify that the NTP client is installed, running, and synchronized to a time source:

```
# sudo -u ilo-pce illumio-pce-env check
```

IPTables

For the initial installation, you should disable iptables.

If iptables is enabled, you must configure it to allow inbound HTTPS connections to the PCE core nodes and service ports.

```
# service iptables stop
# On CentOS 7.x, use the systemctl stop firewalld command.
# chkconfig iptables off
```

Language: UTF-8

Set the system language to a UTF-8 variant of English: either `en_US.UTF-8` or `en_GB.UTF-8`.

Set the variable `LANG="en_US.UTF-8"` or `LANG="en_GB.UTF-8"` in the file `/etc/locale.conf`.

Trusted Public CA Store

A trusted root public Certificate Authority (CA) store must be available for OpenSSL to validate certificates.

If you rely on a certificate signed by a public CA, be sure to install the latest public root CA certificates `ca-certificates` package.

```
# yum install ca-certificates
```

When your certificate is signed by a private CA or the signing CAs are already included in each node's trusted root CA store, the `ca-certificates` package is not required.

PCE Internal Syslog

The PCE comes with an internal syslog configuration. The purpose of the PCE internal syslog is to help organizations use syslog without installing it themselves. See [\(Optional\) Configure PCE Internal Syslog](#).

Process and File Limits and Kernel Parameters

This section describes how to set the process and file limits and OS kernel parameters that are required for PCE operation. The approach is different depending on whether you are configuring an SNC or MNC, and which operating system you are using, so look for the appropriate sections in the discussion that follows.

Three categories of settings must be configured:

- Process and file limits
- OS kernel parameters
- Kernel module tuning

WARNING:The parameter modifications described in this section are strict requirements and must be followed to ensure proper functionality of the Illumio Core. If an Illumio support case is opened, and analysis finds that these parameters are not met, you will be directed to meet these requirements before any additional troubleshooting can be performed.

Keep the following in mind when managing these parameters:

- Root access is needed for many of these procedures. Before you start, be sure you have login credentials for a user account with root permissions.

- When your settings are already greater than these, you do not need to reduce them to these values.
- Make sure you do not have any automated processes that change these values.

SNC Process and File Limits and OS Kernel Parameters

The following table shows the required process and file limits for single-node clusters. To set these values, see [Set and Verify Process and File Limits](#).

Parameter	Value
core (hard)	0
core (soft)	0
nofile (hard) ¹	65535
nofile (soft) ¹	65535
nproc (hard)	65535
nproc (soft)	65535

¹ When you run additional processes on the PCE, such as monitoring or other operations processes, you might need to increase the value of `nofile`.

The following table shows the required OS kernel parameter values for single-node clusters. To set these values, see [Set and Verify OS Kernel Parameters](#).

Parameter	Value
fs.file-max	2000000
net.core.somaxconn	16384
vm.overcommit_memory	1

The following table shows the required SNC kernel module tuning. To set this value, see [Tune the Kernel Module](#).

Parameter	Value
nf_conntrack hashsize	262144

MNC Process and File Limits and OS Kernel Parameters

The following table shows the required process and file limits for multi-node clusters. To set these values, see [Set and Verify Process and File Limits](#).

Parameter	Core Nodes	Data Nodes
core (hard)	0	0
core (soft)	0	0
nofile (hard) ¹	65535	65535

Parameter	Core Nodes	Data Nodes
nofile (soft) ¹	65535	65535
nproc (hard)	65535	65535
nproc (soft)	65535	65535

The following table shows the required OS kernel parameter values for multi-node clusters. To set these values, see [Set and Verify OS Kernel Parameters](#).

Parameter	Core Nodes	Data Nodes
fs.file-max	2000000	2000000
net.core.somaxconn	16384	Use system default setting
vm.overcommit_memory	Use system default setting	1

The following table shows the required kernel module tuning. To set this value, see [Tune the Kernel Module](#).

Parameter	Core Nodes
nf_conntrack hashsize	262144

Set and Verify Process and File Limits

Process and file limits are set by editing configuration files and issuing commands. The techniques vary depending on the operating system version and which system management daemon you are using, `systemd` or `init.d`. If you are not sure which system management daemon is being used, run the following command:

```
$ ps -p1 | grep "init|upstart|systemd"
```

CentOS 7.x or RHEL 7.x/8.x with systemd

On every core and data node, do the following steps:

1. As root, edit the following Illumio-specific configuration file:

```
/etc/systemd/system/illumio-pce.service.d/override.conf
```

2. Place the following lines in the file.:

```
[Service]
LimitCORE=0
LimitNOFILE=65535
LimitNPROC=65535
```

3. Reload the daemon configuration and restart the service to apply the change:

```
# systemctl daemon-reload
# systemctl restart illumio-pce.service
```

4. Verify that the correct settings are now in effect. As the PCE runtime user, run the following command. Verify that the output is as shown:

```
$ sudo -u ilo-pce systemctl show illumio-pce.service | egrep
"LimitCORE|LimitNPROC|LimitNOFILE"
LimitCORE=0
LimitNOFILE=65535
LimitNPROC=65535
```

See also the [Linux systemd man page](#), especially "Table 1. Resource limit directives."

CentOS 7.x or RHEL 7.x/8.x with init.d

On **every core and data node**, do the following steps:

1. As root, edit the following Illumio-specific configuration file:

```
/etc/security/limits.d/99-illumio.conf
```

2. Place the following lines in the file. The `ilo-pce` on each line indicates that the limits apply to only the PCE runtime user, which is `ilo-pce` unless this default user name was overridden during PCE installation. If you want the limits to apply to all users, use asterisks (*) instead of `ilo-pce`.

```
ilo-pce    soft    core    0
ilo-pce    hard    core    0
ilo-pce    soft    nofile  65535
ilo-pce    hard    nofile  65535
ilo-pce    soft    nproc   65535
ilo-pce    hard    nproc   65535
```

3. Apply the change:

```
# sysctl -p /etc/security/limits.d/99-illumio.conf
```

4. Verify that the correct settings are now in effect. As the PCE runtime user, run the following commands. Verify that the output is as shown:

```
$ sudo -u ilo-pce ulimit -n
65535

$ sudo -u ilo-pce ulimit -u
65535

$ sudo -u ilo-pce ulimit -c
0
```

Set and Verify OS Kernel Parameters

Kernel parameters are set by editing configuration files and issuing commands. The commands are the same on all PCE-supported versions of CentOS and RHEL, but the techniques vary depending on whether you are configuring an SNC or MNC.

SNC: Set and verify OS kernel parameters

1. As root, edit the following Illumio-specific configuration file:

```
/etc/sysctl.d/99-illumio.conf
```

2. Place the following lines in the file:

```
fs.file-max          = 2000000
vm.overcommit_memory = 1
net.core.somaxconn   = 16384
```

3. Apply the settings:

```
# sysctl -p /etc/sysctl.d/99-illumio.conf
```

4. Verify that the correct settings are now in effect. As the PCE runtime user, run the following command. Verify that the output is as shown:

```
$ sudo -u ilo-pce sysctl -a 2>/dev/null | egrep "fs.file-max|vm.overcommit_
memory|net.core.somaxconn"
fs.file-max = 2000000
net.core.somaxconn = 16384
vm.overcommit_memory = 1
```

For more information, see [Configuring Kernel Parameters at Runtime](#) in the Red Hat documentation.

MNC: Set and verify OS kernel parameters

1. As root, on **each core node**, edit `/etc/sysctl.d/99-illumio.conf` and add the following lines:

```
fs.file-max          = 2000000
net.core.somaxconn   = 16384
```

2. As you go, on **each core node**, apply the settings:

```
# sysctl -p /etc/sysctl.d/99-illumio.conf
```

3. As root, on **each data node**, edit `/etc/sysctl.d/99-illumio.conf` and add the following lines:

```
fs.file-max          = 2000000
vm.overcommit_memory = 1
```

4. As you go, on **each data node**, apply the settings:

```
# sysctl -p /etc/sysctl.d/99-illumio.conf
```

5. Verify that the correct settings are now in effect. As the PCE runtime user, run the following command. Verify that the output is as shown:

```
$ sudo -u ilo-pce sysctl -a 2>/dev/null | egrep "fs.file-max|vm.overcommit_
memory|net.core.somaxconn"
fs.file-max = 2000000
net.core.somaxconn = 16384
vm.overcommit_memory = 1
```

For more information, see [Configuring Kernel Parameters at Runtime](#) in the Red Hat documentation.

Tune the Kernel Module

Adjust the hash size setting for the kernel `contrack` module as follows. For this setting, the commands are the same on all PCE-supported versions of CentOS and RHEL.

On all core nodes:

1. As root, run the following commands to tune the kernel conntrack module. The commands take effect immediately.

```
# modprobe nf_conntrack
# echo 262144 > /sys/module/nf_conntrack/parameters/hashsize
```

2. Run the following command to apply the same setting automatically on reboot:

```
# echo "options nf_conntrack hashsize=262144" > /etc/modprobe.d/illumio.conf
```

3. Verify that the correct setting is now in effect. Run the following command to inspect the hash size. Verify that the output is as shown:

```
# cat /sys/module/nf_conntrack/parameters/hashsize
262144
```

About Your Organization Name and ID

An organization is a group of policies and users in the Illumio Core. An organization can contain any number of users, workloads, and policy objects (rulesets, IP lists, services, and security settings). When you sign up with Illumio, you will receive an email invitation to create your company's organization in Illumio Core.

Have ready your full organization name, which you specify at installation.

For on-premise PCE deployments, installation creates an organization identifier (org ID) and assigns the value of 1 to org ID. The value 1 distinguishes your on-premises PCE from the Illumio Core Cloud (SaaS) service, where each customer has a unique org ID.

The org ID is needed with the REST API, where you set org-ID to 1 for the on-premises PCE, and for other purposes.

PCE Installation

This chapter contains the following topics:

Install the PCE and UI	40
Configure the PCE	42
Start and Initialize the PCE	49
Additional PCE Installation Tasks	52
After PCE Installation	57
Alternative: Install the PCE Tarball	60

This section provides step-by-step instructions for installing PCE software. Before performing these steps, be sure to understand the concepts in the [Overview](#), and make sure your system is ready for installation as described in [Prepare for PCE Installation](#).

Install the PCE and UI

When installing the PCE and UI packages together, you perform the following high-level steps:

1. Prepare for installation by planning your deployment and reviewing the prerequisites, such as capacity planning and OS setup. See [PCE Installation Planning](#) for information.
2. [Download the software](#).
3. [Install the PCE and UI software](#).
4. [Configure the PCE](#).
5. (Optional) [Validate TLS certificate and private key](#).

6. [Install the TLS certificate and private key.](#)
7. [Verify the runtime environment](#) was configured correctly.
8. [Start the PCE.](#)
9. [Initialize the PCE.](#)
10. [Install Virtual Enforcement Nodes \(VENs\)](#) to enable the PCE to manage your workloads as described in the *VEN Installation and Upgrade Guide*

At this point, the PCE is up and running, receiving communication about workloads from the VENs.

After installing the PCE software, perform these additional procedures to complete your PCE deployment.

11. [Configure backups.](#)
12. (Optional) Configure the internal syslog. See [\(Optional\) Configure PCE Internal syslog](#) for information.

NOTE:

The following tasks describe installing the PCE as an MNC. When you install the PCE as an SNC, you do not repeat the steps on the additional nodes. You can disregard those instructions in the following tasks.

Download the Software

1. Download the software from the [Illumio Support portal](#) (login required).
2. On the **core nodes only**, copy the Illumio PCE UI RPM file to the /tmp folder. The following steps refer to this file as `illumio_ui_rpm`.
3. On **each node** in the cluster, copy the Illumio PCE software RPM file to the /tmp folder. The following steps refer to this file as `illumio_pce_rpm`.

Install the PCE and UI Packages

The packages to install depend on the type of PCE node:

- **Core nodes:** Two packages, the PCE RPM and UI RPM.
- **Data nodes:** One package, the PCE RPM.

1. On **each core node** in the cluster, log in as root and install the PCE RPM:

```
$ rpm -Uvh illumio_pce_rpm
```

For `illumio_pce_rpm`, substitute the path and filename of the software you downloaded from the Illumio Support portal.

2. On **each core node** in the cluster, log in as root and install the UI RPM:

```
$ rpm -Uvh illumio_ui_rpm
```

For `illumio_ui_rpm`, substitute the path and filename of the software you downloaded from the Illumio Support portal.

3. On **each data node** in the cluster, log in as root and install the PCE RPM:

```
$ rpm -Uvh illumio_pce_rpm
```

For `illumio_pce_rpm`, substitute the path and filename of the software you downloaded from the Illumio Support portal.

4. After installing the RPMs, configure the software using the PCE setup wizard. See [Configure the PCE](#) for information.

Configure the PCE

Before running the PCE, set up its runtime configuration.

Use the PCE Runtime Environment File (`runtime_env.yml`) to configure the PCE software. By default, the file is located in `/etc/illumio-pce/runtime_env.yml`. You can create the `runtime_env.yml` file manually or use the PCE software setup script to create and modify the file using interactive prompts at the command line.

For detailed descriptions of the runtime parameters, see [Reference: PCE Runtime Parameters](#).

IMPORTANT:

- The `runtime_env.yml` file contains sensitive information that should be kept secret, such as encryption keys. Take steps to ensure the confidentiality of this file.
- The `runtime_env.yml` file is not included in automatic PCE backups. You must manually back up this file to a secure location.

Set Configuration File Location

By default, `runtime_env.yml` is located in `/etc/illumio-pce/runtime_env.yml`. You can override the default location by setting the `ILLUMIO_RUNTIME_ENV` environment variable. If

you do, you must also set `ILLUMIO_RUNTIME_ENV` in the file `/etc/sysconfig/illumio-pce` to enable the PCE software start-up script to find the file. Log in as root and run the following command (replace *location* with the actual full path).

```
root> echo "ILLUMIO_RUNTIME_ENV=location/runtime_env.yml" >
/etc/sysconfig/illumio-pce
```

For example, if the location is `/var/lib/illumio/data`, run the following command:

```
root> echo "ILLUMIO_RUNTIME_ENV=/var/lib/illumio/data/runtime_env.yml" >
/etc/sysconfig/illumio-pce
```

Run the PCE Setup Script

From the host command line, *as root*, run the following command to launch the setup script:

```
[root]# illumio-pce-env setup
```

The setup script interactively prompts you to provide values for configuration parameters. For descriptions of all the parameters, see [Reference: PCE Runtime Parameters](#). A few of these values, such as `node_type`, will not be the same on all nodes of the PCE cluster; however, many of the values will be the same on all the nodes.

```
WARNING:
The service_discovery_encryption_key value must be identical on all the
nodes in the cluster or the PCE won't start. Be sure to use the same value
for this parameter on all nodes.
```

```
WARNING:
The cluster_type runtime parameter must be set on all PCE nodes, except in
a single node cluster (SNC).
```

When you start the setup script, it checks whether the `$ILLUMIO_RUNTIME_ENV` environment variable is set.

```
$ Illumio PCE Runtime Setup (new configuration -> ENV=my_pce.yml):
```

The `ILLUMIO_RUNTIME_ENV` variable controls where the runtime file will be stored. When the `ILLUMIO_RUNTIME_ENV` variable is not set, the setup script alerts you that the configuration is new and displays the default directory for the runtime file: `/etc/illumio-pce/runtime_env.yml`.

```
$ Illumio PCE Runtime Setup (new configuration)
```

General Configuration

The setup script displays descriptive help text followed by a prompt where you can accept the previous value, the default value, or enter a new value. When the field is optional, press Enter to leave the field empty or accept the default value (if one exists). Fields that have default values display `# default` next to the values.

The prompt shows the previous value in brackets:

```
node_type [core]:
```

Press Enter to use the value in brackets.

TIP:

To determine whether a value is the previously set, default, or recommended value, enter a question mark (?) to display the default value when one exists:

```
opts => core [ data0 data1 ]
node_type [core]: ?
```

When a field has multiple options, type the first few characters of the option and press Tab to auto-complete the field or suggest choices. When prompted for a directory or filename, using auto-complete can help you quickly populate the field.

Press CTRL+C to escape to a control menu, which provides the following options:

- Quit without saving
- Restart the script (with an optional field value)
- Skip to a future field (with a field value)
- Save (with an optional target file)
- Exit

For example, entering this command saves the configuration to a different file and quits the setup.

```
$ Type (q)uit, (r)estart, (f)ield, (s)ave to file or default resume: save  
/tmp/sample.cfg
```

Command-line Batch or List Mode

To operate the setup script from the command line, use the `--batch` option. Instead of prompting for each value, it accepts any previous or default values automatically. When the configuration is missing required fields, the script displays an error and returns a non-zero exit code.

To set a value on the command line:

```
[root]# illumio-pce-env setup front_end_https_port=7443 pce_  
fqdn="sample.illumio.com" -b
```

This command sets the values instead of prompting for them. You can also pre-set the values in non-batch mode by using `key=value` arguments.

NOTE:

Batch mode automatically saves the new configuration in a new configuration file unless there is an error.

To display the currently configured values and replace them with command-line values, use the `--list` option. The `--list` option does not prompt for values or save the configuration to the `runtime_env.yml` file. The `--list` option is useful to [validate your TLS certificate](#).

Advanced Runtime Environment Parameters

Your Illumio Support Representative might provide advanced parameters to add to your `runtime_env.yml` file. When you include the name of these parameters on the command line, the setup script prompts for them.

```
[root]# illumio-pce-env setup advanced_parameter_name_1 advanced_parameter_name_2  
...
```

Additional Options

When using the setup script, several additional options are available. You can use `-h` to display these options.

Usage

```
[root]# illumio-pce-env setup [options...] [field[:field...]=[value[,value...]]...]
```

Display Options

Option	Descriptions
<code>-b, --batch</code>	Don't prompt for field values.
<code>-d, --default</code>	Show default values.
<code>-e, --empty</code>	Display empty fields (implies <code>-d</code>).
<code>-f, --field *[:*] [,...]</code>	Specify a field pattern list; only process these items.
<code>-g, --[no-]guide</code>	Show descriptive information for each field where available (default).
<code>-h, --help</code>	Provide usage statement.
<code>-m, --macros</code>	Show list of available shortcut keys.
<code>-o, --[no-]optional</code>	Process optional fields (default).
<code>-q, --quiet</code>	Don't display help text for each field (same as <code>--no-guide</code>)
<code>-r, --reveal</code>	Don't mask secret keys in field output.
<code>-t, --text</code>	Use regular text instead of colors.

File Options

Option	Description
<code>-c, --config <file></code>	Process a different environment file (<code>new=-</code>).
<code>-s, --save <file></code>	Save results to a different file (<code>stdout=-</code> , system default= <code>!</code>).
<code>-z, --zap</code>	Remove pre-existing default fields.

(Optional) Validate and Configure TLS Certificate

The PCE validates your TLS certificates at start up and displays an error message when the certificate or its chain of trust is invalid.

For information on the contents and formats of your certificates, see [TLS Requirements](#).

You can validate the certificates yourself before or after configuring the PCE as described in [Run the PCE Setup Script](#).

To validate your TLS certificate yourself, including the chain of trust and other aspects, run the following command:

```
illumio-pce-env setup --list
```

The specifying `--list` option checks your configuration and certificates, and indicates possible problems; it does not create a new `runtime_env.yml` configuration file.

Validate After Configuring PCE

To validate the certificates you have already configured and saved in the locations defined in the `runtime_env.yml` file, run the following command:

```
illumio-pce-env setup --list --test 5
```

Specify a verbosity level argument—1 (least) to 5 (most)—with the `--test` option. At verbosity level 5, the command displays the results of its certificate validation.

Alternative Syntax for Certificate Validation

After configuring the PCE, you can validate your certificates in the following additional ways:

- ```
illumio-pce-env setup --list --test 5:some.alternative.hostAndDomainName
```

This syntax checks the certificate and chain against the specified `some.alternative.hostAndDomainName`, such as the FQDN you plan to use for the PCE in production.

- ```
illumio-pce-env setup --list --test 5+
```

The `+` syntax creates a loopback OpenSSL server running on port 4433 and attempts to curl to it.

Validate Before Configuring PCE Certificates

If you have not configured your `runtime_env.yml` file yet, and want to validate your certificates before copying them to your planned production location, run the following command.

```
# illumio-pce-env setup --batch --list \
email=required@emailaddress node=value \
cert=/path/to/cert \
pkey=/path/to/private_key \
trust=/path/to/certificate_chain \
--test 5
```

Option	Description
<code>email=<i>required@emailaddress</i></code>	(Required) Your email address.
<code>node=<i>value</i></code>	Topology to check. For allowable values, see the parameter <code>node_type</code> in Optional Runtime Parameters and see the discussion in Nodes and Clusters .
<code>cert=<i>/path/to/cert</i></code>	The absolute path to your certificate.
<code>pkey=<i>/path/to/private_key</i></code>	The absolute path to your certificate's private key.
<code>trust=<i>/path/to/certificate_chain</i></code>	The absolute path to your certificate's CA chain of trust.

Messages, Errors, and Warnings

These messages indicate correctly configured certificates:

- Valid: Certificate chain is verified
- Valid: web_service_certificate tests passed.

These error message indicate possible problems with your certificates:

- Warning: group xxx can write to web_service_certificate
- Error: unable to find trusted_ca_bundle yyy
- Warning: trusted_ca_bundle missing or inaccessible.
- Missing CA
- Error: unable to verify certificate chain
- Error: unable to validate web_service_certificate

Install Certificate

Copy the TLS certificate and private key to each nodes in your deployment.

You can store the files in any readable location on the node. The PCE RPM installation creates the `/var/lib/illumio-pce/cert` directory where you can store these files.

The certificate and private key must be readable by the PCE runtime user.

Verify the PCE Runtime Environment

1. After configuring the `runtime_env.yml` file, run the environment check command as the PCE runtime user to ensure the node is set up correctly:

```
# sudo -u ilo-pce illumio-pce-env check
Checking PCE runtime environment.
OK
```

This command checks various aspects of the PCE setup. For example, it verifies that the NTP client is installed, running, and synchronized to a time source.

2. Correct any errors.
3. Proceed to the next task: [Start and Initialize the PCE](#).

Start and Initialize the PCE

Starting and initializing the PCE are the final steps in installing it. After completing these steps, you are ready to install VENs on hosts in your environment as described in the *VEN Installation and Upgrade Guide*.

Start the PCE

As the PCE runtime user, perform the following steps:

1. On *all nodes*, start the PCE at runlevel 1:

```
# sudo -u ilo-pce illumio-pce-ctl start --runlevel 1
```

Troubleshooting: If this command fails, verify that you have set `service_discovery_encryption_key` to the same value in `runtime_env.yml` on all PCE nodes.

Wait while all the nodes process the start command, which can take up to 10 minutes. When a node has finished, its status is `RUNNING`.

2. On *all nodes*, verify that they started:

```
# sudo -u ilo-pce illumio-pce-ctl status
```

Expected output:

```
Checking Illumio Runtime          RUNNING 0.38s
```

If any nodes do not start after 10 minutes, check the following issues:

- Network connectivity between nodes and iptables is configured correctly. See [IPTables](#) for information.
- The certificates must be configured correctly. See [TLS Requirements](#) for information.
- The system locale must be UTF-8. See [Language: UTF-8](#) for information.
- The runtime environment is configured correctly. See [Verify the PCE Runtime Environment](#) for information.

Initialize the PCE

As the *PCE runtime user*, perform the following steps:

1. On *any node*, initialize the PCE database:

```
# sudo -u ilo-pce illumio-pce-db-management setup
```

2. On the *data0 node*, bring the system up to runlevel 5:

```
# sudo -u ilo-pce illumio-pce-ctl set-runlevel 5
```

3. On *any core node*, check the status of the cluster:

```
# sudo -u ilo-pce illumio-pce-ctl cluster-status
```

Make sure the cluster status is `RUNNING` before proceeding to the next step.

4. On *any core node*, create the initial PCE user and organization name:

```
# sudo -u ilo-pce illumio-pce-db-management create-domain --user-name user-email-address --full-name user-full-name --org-name organization-name
```

You are prompted for a password. The password must conform to these restrictions: at least 8 characters, no more than 128 characters, at least 1 upper case character, 1 lower case character and 1 number.

For example:

```
# sudo -u ilo-pce illumio-pce-db-management create-domain --user-name
myuser@mycompany.com --full-name
'Joe User' --org-name 'ACME Inc.'

Reading /var/illumio-pce-data/runtime_env.yml.
INSTALL_ROOT=/var/illumio-pce
RENV=production (defaulted because not set in runtime_env.yml)
Please enter a password with at least 8 characters with one uppercase, one
lowercase and
one number.

Enter Password:
Re-enter Password:
-----
Running cd /var/illumio-pce/illumio/webservices/people && RAILS_
ENV=production bundle exec rails
runner script/create_org_owner
--output-file /tmp/illumio/org.yml --user-name myuser@mycompany.com --create-
org
--org-name 'ACME Inc.'
Completed in 5.471846432 sec. Exit Code = 0
-----
Running cd /var/illumio-pce/illumio/webservices/agent && RAILS_ENV=production
bundle
exec rails runner script/create_org_defaults
--input-file /tmp/Illumio/org.yml
Completed in 5.609754678 sec. Exit Code = 0
-----
Running cd /var/illumio-pce/illumio/webservices/login && RAILS_ENV=production
ILO_*****bundle exec rails runner
script/setup_initial_config --org-data /tmp/Illumio/org.yml
--user-name myuser@mycompany.com
--full-name 'Joe User'
domain_name=mycompany.com
Completed in 5.303522871 sec. Exit Code = 0
Done.
```

5. (RHEL 7+ only) Check to be sure the expected session limits for `nofile` and `nproc` meet the minimum requirements for the PCE (see [Process and File Limits](#)). Use the following command:

```
cat /proc/$(pgrep -f config_listener.rb)/limits | grep -e open -e processes
```

If the limits are too low, correct the issue. See [Session Limits Too Low](#).

6. Point a web browser to the PCE FQDN and log in using the account you just created. You should see the PCE web console.

VEN Deployment

In addition to deploying PCE nodes, you must also deploy the Virtual Enforcement Node (VEN) on your distributed, on-premise systems. For more information, see the *VEN Installation and Upgrade Guide*.

Additional PCE Installation Tasks

After installing the PCE, perform these additional tasks.

Configure PCE Backups

You should maintain and perform regular backups of the PCE database based on your company's backup policy. Additionally, always back up your PCE database before upgrading to a new version of the PCE. See [PCE Database Backup](#) in the PCE Administration Guide.

Internal Syslog and Events Configuration Required

This section applies to you if you are:

- Performing a fresh installation of Illumio 20.2.0 or later rather than upgrading from a previous version, and
- You want to send events and traffic flow summaries to an external SIEM.

For new installations, you must configure the syslog and set up events forwarding.

In previous PCE versions, a `local` syslog configuration was created by default. This local setting is no longer created. If you want to gather events data, the internal syslog must be configured. This was previously an optional installation step.

You must configure the following:

- Set up the internal syslog. See [\(Optional\) Configure PCE Internal syslog](#).
- Set up events forwarding. See [Events Settings](#) in the *Events Administration Guide*.

If you are upgrading from a previous PCE version, you can also do this configuration, if needed. However, it is more likely that you already have an appropriate configuration in place.

(Optional) Configure PCE Internal syslog

Configuring the PCE internal syslog is optional only if you are performing either of these tasks:

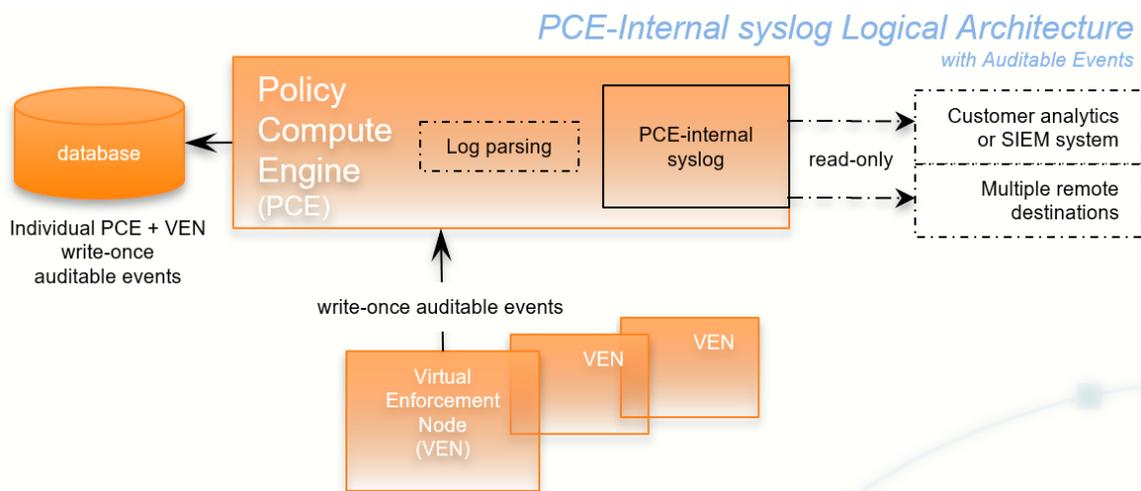
- You are upgrading to Illumio 21.2.0 or later from an earlier version where you already have an appropriate configuration in place.
- You are performing a fresh installation of Illumio 21.2.0 or later, but you don't care about gathering events data or sending events and traffic flow summaries to an external SIEM.

In every other case, it is required.

With the PCE internal syslog, you use the PCE web console to control and configure the relaying of syslog messages from the PCE to multiple remote destinations.

This feature eliminates the need to manage syslog on the PCE by yourself.

You can achieve a smooth transition from existing syslog installations by using a default configuration called “Local.” Using this default, the PCE internal syslog relays messages to the existing syslog.



Utilizing the internal syslog works well with the PCE's auditable events data. See the *Events Administration Guide*.

The PCE internal syslog has the following features:

- Syslog message routing to an unlimited number of remote destinations
- Auditable events for syslog service, as required by [Common Criteria](#)
- Integration with PCE Support Reports
- Common timestamps defined by [RFC 3339](#), including fractional timestamps, such as milliseconds
- PCE log rotation and disk usage management
- SIEM support by enabling sending events to remote destinations
- Optional data-in-motion encryption

Do Not Write Additional Information to `log_dir`

Though *not recommended*, you can put the PCE internal syslog into operation while still running any syslog implementation you already have. However, keep the following information in mind.

CAUTION:

Do not store auditable events in `log_dir`

If you continue to use a previously configured syslog (prior to Illumio Core version 18.2), Illumio recommends that your own local syslog configuration be changed to *not* store any additional information in `log_dir`. The `log_dir` parameter in `runtime_env.yml` defines where logs are written and by default is `/var/log/illumio-pce`. This recommendation includes avoiding storing your auditable events logs in this directory.

The PCE Support Report includes all data in this directory. Illumio considers the auditable event information as private, confidential data. Storing it in `log_dir` could inadvertently release this information by way of the PCE Support Report to persons other than your organization's auditors.

Configure Events and Syslog

After installing the PCE, configure events and the syslog server using the PCE web console.

For information, including configuring remote syslog destinations, see [Events Settings](#) in the *Events Administration Guide*.

(Optional) Customize PCE Log File Rotation

Internal PCE log file rotation is governed by two values: maximum file size (default: 100MB) and maximum retention (default: 10 files). In larger-scale deployments, these values could be an insufficient amount of log data to successfully troubleshoot runtime issues.

To customize the rotation of PCE log files, run the following command:

```
sudo -u ilo-pce illumio-pce-env logs --modify logfile[:size][/rotation]
```

In `logfile`, enter the name of the file. If you do not already know the name of the log file, run this command to list all logs:

```
sudo -u ilo-pce illumio-pce-env logs
```

In `size`, specify a number and append `m` to specify a size in MB or `g` to specify a size in GB. In `rotation`, enter a number to control how many past rotated log files to keep. When this number is exceeded, the oldest file is deleted. To return to the default log rotation values of 100MB and 10 files, run this command with `logfile` alone, without the `size` or `rotation` parameters.

For example:

Argument	Result
<code>haproxy.log:1g/20</code>	Rotate the haproxy log when it reaches 1GB, and keep the last 20 rotated files.
<code>haproxy.log:3m</code>	Set the haproxy.log to 3MB, indicated by the <code>m</code> .
<code>haproxy.log/5</code>	Keep the 5 most recent haproxy.log files after rotation. Discard older ones.
<code>nginx.log</code>	Return the nginx.log file to the default settings.

To confirm that the hosts have sufficient disk space to accommodate the log files with these rotation settings, run this command:

```
sudo -u ilo-pce illumio-pce-ctl check-env
```

It issues a warning if the log usage is too great for the partition size.

(Optional) Set Path to Custom TLS Certificate Bundle

When you enable Transport Layer Security (TLS) mutual authentication, the channel to the remote syslog destination can be secured by your own TLS CA certificate bundle. A CA bundle is a file that contains root and intermediate certificates. The end-entity certificate along with a CA bundle constitutes the certificate chain.

The value of the `runtime_env.yml` file optional parameter `trusted_ca_bundle` is the path to your own CA certificate bundle.

- When a custom TLS bundle is provided by the user during configuration, this bundle is used for certificate verification.
- When a custom TLS bundle is not configured for a particular destination, the PCE trust store is used (`runtime_env.yml` parameter `trusted_ca_bundle`).

Remote Destination Setup for Syslog Server

Enabling TLS with the syslog protocol allows you to secure the communication to your syslog service with public CA certificates or with TLS certificates from your own CA.

On the remote syslog server, ensure restricted access to the data by relying on the OS-level user access mechanisms. In addition, limit the number of users allowed access to the syslog storage itself. If possible, rely on an enterprise-class log management system to post-process the event data.

RFC 5424 Message Format Required

Ensure that your remote syslog destination is configured to use the message format defined by [RFC 5424, The Syslog Protocol](#), with the exception.

Traffic flow summary messages include a prefix of an octal number, like the string **611** highlighted in bold at the beginning of the snippet of a LEEF record below. Ensure that your parsing programs on the remote syslog destination account for this prefix:

```
611 <14>1 2018-08-06T11:47:26.000000+00:00 core1-2x2devtest59 illumio_
pce/collector 22724 - [meta sequenceId="3202"] sec=556046.963 sev=INFO
pid=22724 tid=30548820 rid=e163020f-32c5-4c59-ab06-dfb93b60ff4e
LEEF:2.0|Illumio|PCE|18.2.0|flow_allowed|cat=flow_summary
...
```

NOTE:

Notes on RFC 5424

- You must ensure that your remote syslog uses the `network(flags(syslog-protocol))` form for receiving messages.
- RFC 5424-formatted messages might not be fully functional with rsyslog versions earlier than 5.3.4.

Message Size: 8K

The size of the PCE internal syslog messages is up to 8K bytes. However, many implementations of syslog have a default message size of 4K bytes. Ensure that your remote syslog configuration is set for 8K message size. Configuring the remote destination's syslog message size depends on your implementation of syslog. Consult your vendor documentation for information.

After PCE Installation

This section describes some of the basic things you can see immediately after installing the PCE.

RPM Installation Directories

The PCE software RPM installs to the following directories:

Location	Contents at Installation	Permissions / Ownership
/opt/illumio-pce/	PCE software	dr-xr-x---. root ilo-pce
/etc/illumio-pce	Empty	drwxr-x---. root ilo-pce
/etc/init.d/illumio-pce	Service script	-rwxr-xr-x. root root
/var/lib/illumio-pce/	Empty	drwxr-x---. root ilo-pce
tmp/		drwx-----. ilo-pce ilo-pce
runtime/		drwx-----. ilo-pce ilo-pce
data/		drwx-----. ilo-pce ilo-pce
keys/		drwx-----. ilo-pce ilo-pce
cert/		drwxr-x---. root ilo-pce
/var/log/illumio-pce	Log files	drwx-----. ilo-pce ilo-pce

RPM Runtime User and Group

The PCE installation creates a runtime user and group named `i1o-pce` to run the PCE software. For security, the `i1o-pce` user is configured without a login shell or home directory.

CAUTION:

For better security, do not give the `i1o-pce` user a login shell or home directory.

You should run PCE commands as root or as a user belonging to the `i1o-pce` group. You run the PCE software with `sudo`, as shown throughout this guide:

```
# sudo -u i1o-pce somePCEcommand
```

You might put several users into the `i1o-pce` group for shared maintenance or other needs. However, only the `i1o-pce` user is actually used to run the software.

PCE Control Interface and Other Commands

The Illumio PCE control interface `illumio-pce-ctl` is a command-line tool for performing key tasks for operating your PCE cluster, such as starting and stopping nodes, setting cluster runlevels, and checking the cluster status.

IMPORTANT:

In this guide, all command-line examples are based on an RPM installation. When you install the PCE using the tarball, you must modify the commands based on your PCE user account and the directory where you installed the software.

The PCE includes other command-line utilities used to set up and operate your PCE:

- `illumio-pce-env`: Verify and collect information about the PCE runtime environment.
- `illumio-pce-db-management`: Manage the PCE database.
- `supercluster-sub-command`: Manage specific Supercluster operations.

The PCE control interface can only be executed by the PCE runtime user (`i1o-pce`), which is created during the PCE RPM installation.

Control Command Access with `/usr/bin`

For easier command execution, PCE installation creates softlinks in `/usr/bin` by default for the Illumio PCE control commands. The `/usr/bin` directory is usually included by default in the `PATH` environment variable in most Linux systems. When your `PATH` does not include `/usr/bin`, add it to your `PATH` with the following command. You might want to add this command to your login files (`$HOME/.bashrc` or `$HOME/.cshrc`).

```
export PATH=$PATH:/usr/bin
```

Syntax of `illumio-pce-ctl`

To make it simpler to run the PCE command-line tools, you can run the following Linux softlink commands or add them to your `PATH` environment variable.

```
$ cd /usr/bin
$ sudo ln -s /opt/illumio-pce/illumio-pce-ctl ./illumio-pce-ctl
$ sudo ln -s /opt/illumio-pce/illumio-pce-db-management ./illumio-pce-db-
management
$ sudo ln -s /opt/illumio-pce/illumio-pce-env ./illumio-pce-env
```

After these commands are executed, you can run the PCE command-line tools using the following syntax:

```
$ sudo -u ilo-pce illumio-pce-ctl sub-command --option
```

Where:

`sub-command` is an argument displayed by `illumio-pce-ctl --help`.

PCE Service Script `illumio-pce` for Boot

The `illumio-pce` service script in `/etc/init.d/illumio-pce` switches to the runtime user (`ilo-pce`) prior to running other PCE programs. The primary purpose of the `init.d` service script is to start the product on boot. The service script can also be run with the `/sbin/service` command:

```
$ service illumio-pce
Usage: illumio-pce {start|stop|restart|[cluster-]status|{set|get}-
runlevel|control|database|environment|setup}
```

PCE Runlevels

PCE runlevels define the system services started for common operations, such as upgrade, downgrade, and restore.

The runlevel is set with the following command:

```
illumio-pce-ctl set-runlevel numeric_runlevel
```

The `numeric_runlevel` varies by type of operation.

Setting the runlevel might take some time to complete, depending on the cluster configuration. Check the progress with the following command:

```
illumio-pce-ctl cluster-status -w
```

Alternative: Install the PCE Tarball

You can use these alternative steps instead of the normal installation procedure described in [Install the PCE and UI](#).

NOTE:

The preferred installation mechanism is the RPM distribution, which is easier than the tarball installation.

Process for Installing PCE Tarball

If you are installing the PCE tarball distribution, perform the following tasks on each node in your deployment:

1. Create the PCE user account.
2. Resolve OS dependencies.
3. Create the directory structure for the PCE. The PCE tarball supports a configurable directory structure. This feature allows you to choose the directory structure that best meets your needs.

The following table lists the directories used by the PCE. You need to create these directories and update the listed PCE Runtime Environment File with the proper values.

Directory	Use	Permissions	Example
<code>install_root</code>	PCE binaries and scripts	Read/Execute	<code>/opt/illumio-pce</code>
<code>persistent_data_root</code>	A writable location where the PCE writes its persistent data Must be owned by the user that runs the PCE.	Read/Write	<code>/var/lib/illumio-pce/data</code>
<code>runtime_data_root</code>	A writable location where the PCE writes runtime data Must be owned by the user that runs the PCE.	Read/Write	<code>/var/lib/illumio-pce/runtime</code>
<code>ephemeral_data_root</code>	A writable location where the PCE writes temporary files	Read/Write	<code>/var/lib/illumio-pce/tmp</code>
<code>log_dir</code>	Directory where the PCE writes text file logs You must configure <code>logrotate</code> (or similar) to ensure log files do not grow too large.	Read/Write	<code>/var/log/illumio-pce</code>

The default location of the PCE Runtime Environment File is `/etc/illumio-pce/runtime_env.yml`, but for the exact location on your systems, check the value of the `log_dir` parameter.

- Copy the PCE tarball to the `install_root` directory and untar it.
- Create an init script to run `install_root/illumio-pce-ctlstart` at boot.

Upgrade PCE Tarball Installation

The `$ILLUMIO_RUNTIME_ENV` shell environment variable defines the location of the `runtime_env.yml` file.

The following variables used in this section refer to entries in the `runtime_env.yml` file for each node in the cluster:

- `install_root`
- `persistent_data_root`
- `<log_dir>`

On *all nodes* in the cluster, perform the following steps:

1. Move the old PCE version to a backup directory:

```
$ mv install_root install_root_previous_release
```

For example:

```
$ mv /opt/illumio-pce /opt/illumio-pce-previous-release
```

2. Install the new PCE TGZ version:

```
$ mkdir install_root  
$ cd install_root  
$ tar -xzf illumio_pce_tar_gz
```

Change Tarball to RPM Installation

Perform these steps to install a first-time RPM to replace the previous tarball installation.

1. On *all nodes*, as the *previous PCE runtime user*, stop the PCE:

```
# illumio-pce-ctl stop set-runlevel 1
```

2. Move all files under the `pce_installation_root` directory to a backup directory:

```
# mv pce_installation_root previousinstall-root
```

3. Change the previous PCE runtime user and group to `ilo-pce:ilo-pce`:

```
# usermod --login ilo-pce previous-user  
# groupmod --new-name ilo-pce previous-group
```

4. Install the PCE via the RPM:

```
# rpm -ivh --nopre illumio-pce-16.6-0.x86_64
```

NOTE:

The `--nopre` option prevents the RPM from creating these two empty directories: `/var/lib/illumio-pce` and `/var/log/illumio-pce`.

5. Move the existing `runtime_env.yml` file to `/etc/illumio-pce`.
6. Update the `ILLUMIO_RUNTIME_ENV` environment variable to `/etc/illumio-pce/runtime_env.yml` or delete this environment variable. The PCE looks for the runtime environment file in this location.
7. If necessary, change the `install_root` parameter in the `runtime_env.yml` file to `/opt/illumio-pce`.
8. On *all nodes*, as the *new PCE runtime user*, start the PCE:

```
# sudo -u ilo-pce illumio-pce-ctl start
```

9. On the *data0 node*, as the *new PCE runtime user*, migrate the database:

```
# sudo -u ilo-pce illumio-pce-db-management migrate
```

10. As the *new PCE runtime user*, bring the PCE to runlevel 5:

```
# sudo -u ilo-pce illumio-pce-ctl set-runlevel 5
```

Chapter 4

PCE Upgrade, Downgrade, and Uninstall

This chapter contains the following topics:

PCE Upgrade Prerequisites	64
Upgrade the PCE	65
PCE UI-Only Upgrade	71
Downgrade PCE to Previous Version	71
Uninstall the PCE	75

This section describes how to change to a newer or previous PCE software version (upgrade or downgrade) or remove PCE software. This section assumes that you have previously installed the PCE as described in [PCE Installation](#).

PCE Upgrade Prerequisites

This section provides information on how to prepare to upgrade the PCE.

Upgrade Paths and Planning Tool

For information about upgrade paths for versions of the PCE and VEN, see [Versions and Releases](#) on the Illumio Support portal (login required).

For information to help you plan your upgrade, see the [Illumio Core Upgrade Path](#) page on the Illumio Support portal (login required).

Upgrade Prerequisites

Consider the following important requirements before you begin upgrading or downgrading the PCE:

- Do not upgrade your VENs until the PCE version upgrade is successful. After Illumio VENs are upgraded, rolling back the PCE upgrade is not supported.
- Ensure that no asynchronous jobs have been submitted right before you begin the upgrade. As a best practice, wait until all asynchronous jobs have finished before upgrading the PCE.
- For a multi-version upgrade, the “Back Up PCE Database and Current Software” steps should only be done a single time at the beginning of the first upgrade sequence. This method allows you to roll back to the starting version if there is an issue with the upgrade.
- If you are upgrading from a PCE version earlier than 21.2, and you used the scripts `ilo-pipgen` and `ilo-vpngen` to install the earlier version, remove those scripts. They are not needed in version 21.2 and later. If installed in the default location, the scripts can be found at `/var/tmp/illumio-pipgen/ilo-pipgen` and `/var/tmp/illumio-vpngen/ilo-vpngen`.

Upgrade the PCE

This section describes how to upgrade the PCE and its UI together. To upgrade the UI alone, see [UI-Only Upgrade](#) for information.

When upgrading the PCE and UI packages together, perform the following high-level tasks:

1. Verify that all [Upgrade Prerequisites](#) are met.
2. Perform PCE installation planning and prerequisite steps if they have not already been done on this PCE, as described in [Preparing for PCE Installation](#).
3. [Back up the PCE](#).
4. [Download the software](#).
5. [Stop the PCE](#).
6. [Install the new PCE and UI](#).
7. [Update the runtime environment file](#).
8. [Migrate the database](#).
9. [Set runlevel to 5](#).
10. [Verify successful upgrade](#).

Back Up the PCE

When you are upgrading from a previous PCE version, the first step is to back up your existing data.

Back Up PCE Data

1. (On an SNC, skip this step.) Before you back up the PCE, determine which data node is running the `agent_traffic_redis_server` service:

```
$ sudo -u ilo-pce illumio-pce-ctl cluster-status
```

You see the following output:

```
SERVICES (runlevel: 5) NODES (Reachable: 1 of 1)
=====
agent_background_worker_service 192.168.33.90
agent_service NOT RUNNING
agent_slony_service 192.168.33.90
agent_traffic_redis_cache 192.168.33.90
agent_traffic_redis_server 192.168.33.90      <=== run the dump command
from this node
agent_traffic_service NOT RUNNING
...
```

2. On the *data node* that is running the `agent_traffic_redis_server` service, run the following command:

```
$ sudo -u ilo-pce illumio-pce-db-management dump --file <location-of-db-dump-
file>
```

In `location-of-db-dump-file`, enter a file name.

NOTE:
On an SNC, run this command on the single node.

3. After the dump command finishes, copy the backup files to a fault-tolerant storage location.

Back up the PCE Runtime Environment File

Store a copy of each node's `runtime_env.yml` file on a system that is not part of the cluster or Supercluster. The default location of the PCE Runtime Environment File is `/etc/illumio-pce/runtime_env.yml`.

Download the Software

1. Download the software from the [Illumio Support portal](#) (login required).
2. On the *core nodes only*, copy the Illumio PCE UI RPM file to the `/tmp` folder. In the following steps, this file is referred to as `illumio_ui_rpm`.
3. On *all nodes* in the cluster, copy the Illumio PCE software RPM file to the `/tmp` folder. In the following steps, this file is referred to as `illumio_pce_rpm`.

Stop the PCE

1. On *all nodes* in the cluster, stop the PCE:

```
$ sudo -u ilo-pce illumio-pce-ctl stop --wait
```

2. On *all nodes* in the cluster, verify the PCE status is STOPPED:

```
$ sudo -u ilo-pce illumio-pce-ctl status -sv --wait
```

Install the New PCE and UI

The packages to install depend on the type of PCE node:

- **Core nodes:** Two packages, the PCE RPM and UI RPM.
- **Data nodes:** One package, the PCE RPM.

1. On **each core node** in the cluster, log in as root and install the PCE RPM and the UI RPM. Be sure to specify both of the RPM file names on the command line:

```
$ rpm -Uvh illumio_pce_rpm illumio_ui_rpm
```

For `illumio_pce_rpm` and `illumio_ui_rpm`, substitute the paths and filenames of the two RPM files you downloaded from the Illumio Support portal.

2. On **each data node** in the cluster, log in as root and install the PCE RPM:

```
$ rpm -Uvh illumio_pce_rpm
```

For `illumio_pce_rpm`, substitute the path and filename of the software you downloaded from the Illumio Support portal.

Update the Runtime Environment File

See [What's New and Changed in This Release](#) to determine if any changes to the PCE Runtime Environment File (`runtime_env.yml`) are required to upgrade.

WARNING:

The `cluster_type` runtime parameter *must* be set on all PCE nodes starting in PCE 21.5.0, except on a single node cluster (SNC). If you are upgrading from a version earlier than 21.5.0, be sure to set this parameter. For details about what value to use on each type of node, see [Reference: PCE Runtime Parameters](#).

To make changes to the runtime environment configuration:

1. On *all nodes* in the cluster, update the `runtime_env.yml` file.
2. On *all nodes* in the cluster, check the validity of the `runtime_env.yml` file:

```
$ sudo -u ilo-pce illumio-pce-ctl check-env
```

If any issues are reported by this command, correct them before moving on to the next step.

Migrate the PCE Database

Ensure that you have upgraded all nodes in the cluster to the same version before you perform these steps. Otherwise, none of the nodes in your cluster will start.

1. On *all nodes* in the cluster, start the PCE:

```
$ sudo -u ilo-pce illumio-pce-ctl start --runlevel 1
```

2. For some upgrades, you might be prompted to upgrade the database PostgreSQL software on one of the data nodes.

At the prompt, enter `yes` to continue the upgrade.

If you do not see this prompt, go to the next step.

```
Upgrade needed. The agent database is currently running at postgres version
9.6 and needs to be upgraded to version 11.4
Upgrade needed. The traffic database is currently running at postgres version
9.6 and needs to be upgraded to version 11.4
The PCE software will now upgrade to a newer version of the postgres software
from an older version.
You need to migrate the PCE database after this process has finished.
Prior to this upgrade, Illumio recommends you make a backup of the relevant
data directories:
/var/illumio_pce_data/persistent
The upgrade must not be interrupted, otherwise the data might get corrupted
and prevent the PCE from starting.
Do you wish to continue with the database upgrade. [yes/no]: yes
Proceeding with database upgrade.
Please wait until the system has reached runlevel 1.
```

3. On *all nodes* in the cluster, verify the PCE status:

```
$ sudo -u ilo-pce illumio-pce-ctl status -sv --wait
```

4. Verify that the runlevel is 1:

```
$ sudo -u ilo-pce illumio-pce-ctl get-runlevel
```

5. On *any node* in the PCE cluster, migrate the database to the latest schema version:

```
$ sudo -u ilo-pce illumio-pce-db-management migrate
```

NOTE:

This command can take some time to complete, depending on the amount of data. To check progress, view the Health page in the PCE web console. If the migration is still underway, you will see a message like "Traffic database migration in progress."

Set Runlevel 5

Bring the PCE to runlevel 5, full operation:

```
$ sudo -u ilo-pce illumio-pce-ctl set-runlevel 5
```

IMPORTANT:

If you did not run the `illumio-pce-db-management migrate` command on the primary database, you *cannot* bring the node up to runlevel 5 and you *cannot* start the other nodes in the cluster. If some of the nodes in the cluster are already running, they will shut down until you successfully migrate the database. If you attempt to start the upgraded PCE cluster without migrating the database, this error is displayed:

```
$ sudo -u ilo-pce illumio-pce-ctl start
Starting Illumio Runtime STARTING 20.96s
$
$ Stopping PCE software: DB migrations mismatch for DB: avenger_
executor_dev: Missing migrations.
```

Verify Success

1. On *all nodes* in the cluster, verify the PCE status:

```
$ sudo -u ilo-pce illumio-pce-ctl status -s -v -w
```

2. On *any node*, verify the cluster status:

```
$ sudo -u ilo-pce illumio-pce-ctl cluster-status -w
```

3. When you have a front end load balancer (F5 or DNS), make sure that the load balancer is sending requests to the two core nodes in the cluster.
4. Log into the PCE web console, pair a VEN, and verify VEN sync status is showing as “Verified” for a few randomly selected workloads. Select a Workload’s details page and verify that the Policy Sync section shows “Verified.”
5. Test that the firewall is correctly configured by connecting to restricted services using `telnet`. When the PCE is running, connections on TCP port 8300 should be accepted from other PCE hosts but rejected otherwise. Connections on TCP port 8443 should be accepted from all sources (unless modified using the `-p` option).

PCE UI-Only Upgrade

You can upgrade the PCE web console UI alone, as long as the UI version is compatible with the installed PCE version. This upgrade method is useful for taking advantage of bug fixes and other updates that affect the PCE web console only.

NOTE:

You can't install the UI by itself. For the UI to work, a compatible version of the PCE must already be installed.

You do not need to change the PCE runlevel or stop the PCE to upgrade the UI.

To upgrade the UI only:

1. Download the UI software from the [Illumio Support portal](#) (login required).
2. On the *core nodes only*, copy the Illumio PCE UI RPM file to the /tmp folder. In the next step, this file is referred to as `illumio_ui_rpm`.
3. On *all core nodes* in the cluster, upgrade to the new PCE UI:

```
$ rpm -Uvh illumio_ui_rpm
```

Downgrade PCE to Previous Version

This section describes how to roll back the PCE to a previous version in the event of a PCE upgrade failure or defect. You can downgrade to any currently supported PCE version; see [Versions, Compatibility & Support Status](#) on the Illumio Support site.

Downgrade the PCE

To downgrade to a previous PCE version, you will need the following files:

- PCE software installation files for the older version. See [Download the Software](#).
- Database backups taken on the previous version. See [Database Backup](#) in the PCE Administration Guide.
- Backup of the `runtime_env.yml` file from the previous version. See [Back Up the PCE Runtime Environment File](#) in the PCE Administration Guide.

To downgrade the PCE, perform the following steps:

1. On *all nodes* in the cluster, stop the PCE:

```
$ sudo -u ilo-pce illumio-pce-ctl stop
```

2. On *all nodes* in the cluster, downgrade the installation by installing the older version of the PCE and UI.

RPM Installation:

```
$ sudo rpm -Uh --force illumio_pce_rpm illumio_ui_rpm
```

For example:

```
$ sudo rpm -Uh --force illumio-pce-21.2.8-2.c6.x86_64.rpm illumio-pce-ui-21.2.8.UI1-1.x86_64.rpm
```

Tarball Installation:

```
$ mv install_root_previous_release install_root
```

For example:

```
$ mv /opt/illumio-pce-previous-release /opt/illumio-pce
```

3. If you changed the `runtime_env.yml` file, restore the previous version of the file:

```
$ cp /etc/illumio-pce/runtime_env.yml-backup /etc/illumio-pce/runtime_env.yml
```

4. On *all nodes* in the cluster, reset the node:

```
$ sudo -u ilo-pce illumio-pce-ctl reset
```

5. On *all nodes* in the cluster, start the PCE at runlevel 1:

```
$ sudo -u ilo-pce illumio-pce-ctl start --runlevel 1
```

6. On *all nodes* in the cluster, verify the PCE status and runlevel:

```
$ sudo -u ilo-pce illumio-pce-ctl status -s -v -w
$ sudo -u ilo-pce illumio-pce-ctl cluster-status -w
```

7. Set up the database. First, determine the primary database (on an SNC, you can skip this step, as there is only one possible node):

```
$ sudo -u ilo-pce illumio-pce-db-management show-primary
```

8. On the *primary data node*, run this command to set up the database:

```
$ sudo -u ilo-pce illumio-pce-db-management setup
```

9. Restore the PCE policy database. On *one of the data nodes* of the cluster (or in an SNC, on the single node), restore a known good backup:

```
$ sudo -u ilo-pce illumio-pce-db-management restore --file Location_of_db_dump_file
```

For example, if you are downgrading because of an unsuccessful upgrade attempt, restore the backup you took before doing the upgrade.

10. (On an SNC, you can skip this step.) Copy the restored Illumination data file to the *other data node*. The file is located in the following directory:

```
persistent_data_root/redis/redis_traffic_0_master.rdb
```

11. Migrate the PCE database. On *one of the data nodes* in the cluster, migrate the database to the latest schema version:

```
$ sudo -u ilo-pce illumio-pce-db-management migrate
```

12. Restore the traffic database. Run this command on the same node where you took the traffic database backup:

```
$ sudo -u ilo-pce illumio-pce-db-management traffic restore --file /path/to/traffic_db_dump_file
```

This command prompts you to return the PCE to runlevel 5. You can choose to go to runlevel 5 or not.

13. If you did not accept the change to runlevel 5 after restoring the traffic database, you can bring the PCE to runlevel 5, full operation, at any time using the following command:

```
$ sudo -u ilo-pce illumio-pce-ctl set-runlevel 5
```

IMPORTANT:

If you did not run the `illumio-pce-db-management migrate` command, you *cannot* bring the node up to runlevel 5 and you *cannot* start the other nodes in the cluster. If some of the nodes in the cluster are already running, they will shut down until you successfully migrate the database. If you attempt to start the upgraded PCE cluster without migrating the database, this error is displayed:

```
$ sudo -u ilo-pce illumio-pce-ctl start
Starting Illumio Runtime STARTING 20.96s
$
$ Stopping PCE software: DB migrations mismatch for DB: avenger_
executor_dev: Missing migrations.
```

Verify Success of Downgrade

1. On *all nodes* in the cluster, verify the PCE status:

```
$ sudo -u ilo-pce illumio-pce-ctl status -s -v -w
```

2. On *any node*, verify the cluster status:

```
$ sudo -u ilo-pce illumio-pce-ctl cluster-status -w
```

3. When you have a front end load balancer (F5 or DNS), make sure that the load balancer is sending requests to the two core nodes in the cluster.
4. Log into the PCE web console, pair a VEN, and verify VEN sync status is showing as “Verified” for a few randomly selected workloads. Select a Workload’s details page and verify that the Policy Sync section shows “Verified.”

5. Test that the firewall is correctly configured by connecting to restricted services using `telnet`. When the PCE is running, connections on TCP port 8300 should be accepted from other PCE hosts but rejected otherwise. Connections on TCP port 8443 should be accepted from all sources (unless modified using the `-p` option).

Uninstall the PCE

In order to completely uninstall and remove the PCE for your system, perform the following steps:

1. Run this command to remove the PCE:

```
$ rpm -e illumio-pce
```

2. Manually delete these directories:

```
/var/lib/illumio-pce  
/var/log/illumio-pce  
/etc/illumio-pce
```

PCE Installation Reference

This chapter contains the following topics:

Reference: PCE Runtime Parameters	76
FIPS Compliance for PCE and VEN	86

This section contains reference material related to PCE installation. You might need to refer to this information while you are performing the installation steps in [Install the PCE and UI](#), or refer to it for ongoing needs after the PCE is installed.

Reference: PCE Runtime Parameters

This section lists important PCE runtime configuration parameters, their meaning, their purpose, and their exposure levels.

When configuring the PCE with the `illumio-pce-env` setup script, you are prompted for many of these parameters. See [Configure the PCE](#) for information.

IMPORTANT:

- The `runtime_env.yml` file contains sensitive information that should be kept secret, such as encryption keys. Take steps to ensure the confidentiality of this file.
- The `runtime_env.yml` file is not included in automatic PCE backups. You must manually back up this file to a secure location.

Runtime File Exposure Levels

The Illumio PCE `runtime_env.yml` file provides the following exposure levels for PCE configuration:

- **Public Stable** (`public_stable`): These `runtime_env.yml` parameters can be used by all customers. All changes are backward compatible.
- **Public Experimental** (`public_experimental`): These `runtime_env.yml` parameters can be used by all customers but might change from release to release with no guarantee of backwards compatibility.

Required Runtime Parameters

The following table lists the required `runtime_env.yml` file parameters for each PCE software node you deploy. All required parameters have no default values. All paths configured in this file must be absolute.

Required Parameter	Description	Exposure
<code>enabled_preview_features</code>	Includes sub-parameters to enable identified preview features	
<code>install_root</code>	<p>The full path to the location of the PCE binaries and scripts</p> <p>The software does not write to any files in this directory, so it can be read-only.</p> <p>For example:</p> <pre>install_root: /opt/illumio-pce</pre>	Public Stable
<code>runtime_data_root</code>	<p>The full path to the location where the PCE writes runtime data</p> <p>This data can be deleted on reboot if necessary. This directory should have 700 permissions, but all of its files will have 600 permissions. This directory must be owned by the user that runs the PCE software.</p> <p>For example:</p> <pre>runtime_data_root: /var/lib/illumio-pce/runtime</pre>	Public Stable
<code>persistent_data_root</code>	<p>The full path to the location where the PCE writes persistent data</p> <p>This data must persist across reboots for the software to work properly. This directory should have 700 permissions,</p>	Public Stable

Required Parameter	Description	Exposure
	<p>but all of its files will have 600 permissions. This directory must be owned by the user that runs the PCE software.</p> <p>For example:</p> <pre data-bbox="354 499 1240 583">persistent_data_root: /var/lib/illumio-pce/data</pre>	
ephemeral_data_root	<p>The full path to the location where the PCE writes temporary files</p> <p>These files must not be deleted while the software is running, but they should be deleted on reboot. This directory should have 700 permissions, but all of its files will have 600 permissions.</p> <p>For example:</p> <pre data-bbox="354 951 1240 1035">ephemeral_data_root: /var/lib/illumio-pce/tmp</pre>	Public Stable
log_dir	<p>The directory where the PCE software writes some text file logs (although most PCE services log to syslog)</p> <p>logrotate (or similar) should be used to manage these files.</p> <p>For example:</p> <pre data-bbox="354 1268 1240 1352">log_dir: /var/log/illumio-pce</pre>	Public Stable
pce_fqdn	<p>The fully qualified domain name (FQDN) of the PCE cluster</p> <p>For example:</p> <pre data-bbox="354 1484 1240 1568">pce_fqdn: pce.mycompany.com</pre>	Public Stable
cluster_public_ips: cluster_fqdn	<p>The FQDN of your entire cluster</p> <pre data-bbox="354 1640 1240 1843"> NOTE: If you change the value of cluster_public_ips, wait for the paired VENs to receive the new IP addresses and begin heartbeating to them.</pre>	Public Stable
web_ser-	Full path to the X.509 public certificate used by this node for	Public

Required Parameter	Description	Exposure
vice_certificate	<p>TLS</p> <p>See TLS Requirements for more information on the contents of the certificate files.</p> <p>For example:</p> <pre>web_service_certificate: /etc/pki/tls/certs/my_cert.crt</pre>	Stable
web_service_private_key	<p>The RSA private key for TLS that matches the public certificate</p> <p>The private key must be PEM encoded in PKCS#12 format without a password.</p> <p>For example:</p> <pre>web_service_private_key: /var/lib/illumio-pce/cert/rsa_private_key.key</pre> <p>Alternatively, you can specify a script (using \$ notation) that outputs the private key. This approach is useful when you need to store the key in a hardware security module (HSM) or other key store.</p> <p>For example:</p> <pre>web_service_private_key: \$ /var/lib/illumio-pce/cert/get_rsa_private_key.sh</pre> <p>This script can be located anywhere on the file system as long as it is executable by the ilo-pce user.</p> <p>Example script output:</p> <pre>\$ /local/scripts/get_rsa_private_key.sh -----BEGIN RSA PRIVATE KEY----- MIIE... many lines trimmed here -----END RSA PRIVATE KEY-----</pre>	Public Stable

Required Parameter	Description	Exposure
email_address	Email sender address used by the PCE when sending emails from the system; for example, to send invitations and notifications For example: <pre data-bbox="354 541 1240 625">email_address: noreply@exampleblocked_traffic.com</pre>	Public Stable
service_discovery_fqdn	The FQDN or IP address of the first core node	Public Experimental
service_discovery_encryption_key	The key used to encrypt Service Discovery node traffic. This value must be the same for all PCE nodes. This key must be 16 bytes that are base64 encoded. For example: <pre data-bbox="354 1031 1240 1108">service_discovery_encryption_key: 05T1qH1W0cKcK797DV73yg==</pre>	Public Stable
node_type	The type of the PCE software node Allowable values: <ul data-bbox="358 1255 1219 1577" style="list-style-type: none"> • core: core node • data0: data node • data1: data node • snc0: single-node cluster • citus_coordinator: coordinator node for multi-node traffic database • citus_worker: worker node for multi-node traffic database For example: <pre data-bbox="354 1675 1240 1753">node_type: core</pre>	Public Stable
login_banner	A custom message on the PCE login screen typically used to display legal notice or company policy when a user logs in	Public Stable

Required Parameter	Description	Exposure
cluster_type	<p>PCE cluster type. Required on every node in a multi-node cluster (MNC). Not required on a single-node cluster (SNC).</p> <p>One of the following:</p> <ul style="list-style-type: none"> 4node_v0: 2x2 PCE cluster 4node_v0_small: 2x2 PCE cluster with fewer compute and memory resources 6node_v0: 4x2 PCE cluster 4node_dx: 2x2 PCE cluster with multi-node traffic database 6node_dx: 4x2 PCE cluster with multi-node traffic database <p>Default: 4node_v0</p>	Public Stable

Optional Runtime Parameters

The following table lists common optional `runtime_env.yml` file parameters for each PCE software node you deploy. Your Illumio Professional Services representative might provide additional parameters to configure certain advanced functions.

Optional Parameter	Description	Exposure
ven_repo_url	<p>The base URL used to fetch the VENs and to enable workload pairing with the PCE</p> <p>Required format: <code>https://host[:port]/repo_dir</code></p> <p>You can use alternate ports by specifying the port at the end of hostname. <code>repo_dir</code> cannot be empty.</p> <p>For example:</p> <pre>https://repo.example.com:8443/onpremgCBURz8Y4zkGk1u7N9ia1jPG1Z</pre> <p>Default: None</p>	Public Stable
ven_repo_ips	<p>IP addresses of the VEN repository</p> <p>These IP addresses are injected into iptables to allow outbound access to the <code>yum/apt</code> get repositories without having to write an explicit PCE policy.</p>	Public Stable

Optional Parameter	Description	Exposure
	<p>Setting this parameter allows outbound access on ports 80 and 443 to these IP addresses. You can specify both single IP addresses or IP addresses with CIDR notation.</p> <p>When you do not specify this parameter, the VEN won't be allowed to access the repository containing VEN software packages.</p> <p>For example:</p> <pre data-bbox="375 646 1219 814"> ven_repo_ips: - 1.2.3.4 - 5.6.7.8/8 </pre> <p>Default: None</p>	
<p>internal_service_ip</p>	<p>The IP address of the PCE</p> <p>Set this value manually only when you want to use a public IP address or the PCE node has multiple interfaces.</p> <p>For example:</p> <pre data-bbox="375 1108 1219 1192"> internal_service_ip: 10.2.8.89 </pre> <p>Default: The first available private IP address on the node</p>	<p>Public Stable</p>
<p>front_end_https_port</p>	<p>The front end HTTPS port</p> <p>When the cluster is front-ended by a server load balancer, such as F5, it must be configured to forward this port.</p> <p>For example:</p> <pre data-bbox="375 1497 1219 1581"> front_end_https_port: 8443 </pre> <p>Default: TCP 8443 if not set by front_end_management_https_port or front_end_https_port</p>	<p>Public Stable</p>
<p>front_end_event_service_port</p>	<p>The front end Event Service port</p> <p>When the cluster is front-ended by a server load balancer, such as F5, it must be configured to forward this port. The idle connection timeout on the server load balancer might</p>	<p>Public Stable</p>

Optional Parameter	Description	Exposure
	<p>need to be configured to maintain the connections on this port. Please contact your Illumio Professional Services representative for information on configuring your server load balancer.</p> <p>For example:</p> <pre data-bbox="375 541 1218 627">front_end_event_service_port: 8444</pre> <p>Default: 8444</p>	
front_end_management_https_port	<p>The port for PCE web console and REST API</p> <p>This key separates different kinds of communication. See also front_end_https_port.</p> <p>Default: TCP 8443 if not set by front_end_management_https_port or front_end_https_port</p>	Public Stable
syslog_event_export_format	<p>The export format (CEF, LEEF, or JSON) for VEN flow summaries and Organization events.</p> <p>When you specify CEF or LEEF format, you will continue getting traffic flows and Organization events in JSON format.</p> <p>For example:</p> <pre data-bbox="375 1264 1218 1350">syslog_event_export_format: cef</pre> <p>Default: json</p>	Public Stable
min_tls_version	<p>The minimum Transport Layer Security (TLS) version used to secure VEN-to-PCE communications, the PCE's web server for the PCE web console, and the REST API. It is recommended that you use the default setting, 1.2. Earlier TLS versions, such as 1.0 and 1.1, are considered less secure, so it is recommended you do not use them. In rare circumstances, such as when using older operating systems, you might need to change the minimum TLS version; see TLS Versions for Communications.</p> <p>For more information, see Illumio PCE (Core & Edge) Ver-</p>	Public Stable

Optional Parameter	Description	Exposure
	<p>sion 21.2.0 Default Configuration Only Supports TLS 1.2 (requires login).</p> <p>Allowable values: <code>tls1_0</code>, <code>tls_1_1</code>, <code>tls1_2</code>.</p> <p>For example:</p> <pre data-bbox="375 512 1214 596">min_tls_version: tls1_2</pre> <p>Default: <code>tls1_2</code></p>	
<p><code>insecure_tls_weak_ciphers_enabled</code></p>	<p>Specifies whether to allow the use of weaker TLS ciphers, such as cipher block chaining (CBC) ciphers. Stronger ciphers are recommended.</p> <p>Illumio recommends you keep the default value (<code>true</code>) for this setting when using clients or operating systems that can only negotiate TLS using CBC ciphers. If your environment is not impacted by this limitation, Illumio recommends that you change the value to <code>false</code> so that you use strong ciphers.</p> <p>For example:</p> <pre data-bbox="375 1157 1214 1241">insecure_tls_weak_ciphers_enabled: true</pre> <p>Default: <code>true</code></p>	<p>Public Stable</p>
<p><code>trusted_ca_bundle</code></p>	<p>The path to the trusted root certificate bundle.</p> <p>The PCE uses this parameter to validate that the certificates are trusted and indicates the path to the trusted root certificate bundle file.</p> <p>For example:</p> <pre data-bbox="375 1583 1214 1709">trusted_ca_bundle: /etc/ssl/certs/ca-bundle.crt</pre> <p>Default: <code>/etc/ssl/certs/ca-bundle.crt</code></p>	<p>Public Stable</p>
<p><code>email_display_name</code></p>	<p>Email display name to be used when sending email from the system. For example, to send invitations and noti-</p>	<p>Public Stable</p>

Optional Parameter	Description	Exposure
	<p>fications from the PCE.</p> <p>For example:</p> <pre data-bbox="375 411 1219 495">email_display_name: 'noreply'</pre> <p>Default: noreply</p>	
smtp_relay_address	<p>SMTP relay information used by the PCE to send email; for example, to send invitations and notifications.</p> <p>The PCE assumes that an SMTP Relay runs on localhost and listens on 127.0.0.1/587. When this isn't the case, you must specify the configuration on the <i>core nodes</i>.</p> <p>Use <i>one</i> of the following formats:</p> <ul data-bbox="375 884 932 968" style="list-style-type: none"> • ip_address (e.g. 127.0.0.1) • ip_address:port (e.g. 127.0.0.1:587) <p>For example:</p> <pre data-bbox="375 1062 1219 1146">smtp_relay_address: 127.0.0.1:587</pre> <p>Default: 127.0.0.1:587</p>	Public Stable
export_flow_summaries_to_fluentd	<p>The types of traffic flow summaries to export to Fluentd.</p> <p>Values: accepted (allowed), potentially_blocked, blocked</p> <p>For example:</p> <pre data-bbox="375 1402 1219 1612">export_flow_summaries_to_fluentd: - accepted - potentially_blocked - blocked</pre>	Public Experimental
export_flow_summaries_to_syslog	<p>Enables traffic flow summaries to syslog.</p> <p>Values: accepted (allowed), potentially_blocked, blocked</p> <p>For example:</p> <pre data-bbox="375 1801 1219 1869">export_flow_summaries_to_syslog:</pre>	Public Experimental

Optional Parameter	Description	Exposure
	<ul style="list-style-type: none"> - accepted - potentially_blocked - blocked <p>To export blocked traffic summaries, include only the flow summary type when specifying the parameter; for example:</p> <pre>export_flow_summaries_to_syslog: - blocked</pre>	
internal_syslog_fqdn_enabled	<p>Specifies whether to use the PCE's fully-qualified domain name (FQDN) or the hostname in syslog messages. The FQDN can be more helpful if the short hostnames are difficult to distinguish.</p> <p>Values: true (the host= field uses the FQDN), false (default)</p> <p>For example:</p> <pre>internal_syslog_fqdn_enabled: true</pre>	Public Experimental

FIPS Compliance for PCE and VEN

NOTE: This release supports FIPS compliance for the PCE and Linux and Windows VENs. It does not support FIPS compliance for the AIX and Solaris VENs.

This section describes the operational requirements for compliance with Federal Information Processing Standard (FIPS) 140-2 for the PCE and VEN.

FIPS Prerequisites

- PCE server hardware requires the [Intel Ivy Bridge CPU](#) (2012) or later.
- RedHat v7.4 or later required.

- Customer-provided SSL certificates from a public CA or a customer CA. The certificates must have a minimum key size of 2048 to secure PCE communications.

FIPS-related Government and Vendor Documentation

- [Federal Information Processing Standard \(FIPS\) 140-2](#), Security Requirements for Cryptographic Modules
- [Red Hat Enterprise Linux OpenSSL Cryptographic Module NIST Security Policy](#)
- [RHEL v7.1 Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v4.0](#)
- [RHEL v7.4 Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v5.0](#)
- [RHEL v8.x Red Hat Enterprise Linux 8 OpenSSL Cryptographic Module v8.0](#)
- [Windows Server 2012 NIST Security Policy](#)
- [Windows Server 2016 NIST Security Policy](#)

Non-Government Customers without FIPS Requirement

Compliance to FIPS 140-2 requires additional operational restrictions, such as specific OS versions and server hardware.

Illumio recommends that non-government customers who do not have requirement for FIPS 140-2 do not configure and deploy Illumio Core to support FIPS compliance.

Compliance Affirmation Letters

Third-party FIPS-compliance affirmation letters for the Illumio Core are available at [FIPS 140-2 Affirmation Letters](#) (PDF download).

Prerequisites for Linux VEN FIPS Compliance

For SecureConnect (IPsec encryption among workloads), to claim FIPS compliance, the VEN must be installed on RHEL v7.1, RHEL v7.4, or RHEL v8.0 and configured to operate in FIPS mode as described in the following vendor documents:

- RedHat v7.1, Section 9.1 (“Cryptographic Officer Guidance”) of the RHEL v7.1 [Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v4.0](#).
- RedHat v7.4, Section 9.1 (“Cryptographic Officer Guidance”) of the RHEL v7.4 [Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v5.0](#).
- RHEL v8.0, Section 9.1 (“Crypto Officer Guidance”) of the RHEL v8.0 [Red Hat Enterprise Linux 8 OpenSSL Cryptographic Module v8.0](#)

The Linux VEN versions do not have other special OS requirements or additional configurations to enable FIPS-compliant OpenSSL communications. The Linux VEN's FIPS OpenSSL module is built directly into the VEN and is not supplied by the underlying OS; the Linux VEN operates by default in FIPS mode.

Prerequisites for Windows VEN FIPS Compliance

For FIPS compliance on Windows, either Windows Server 2012 or Windows Server 2016 must be configured according to the following vendor documents:

- Windows 2012 conforming with Section 2 of the [Windows Server 2012 NIST Security Policy](#)
- Windows 2016 conforming Section 2 of the [Windows Server 2016 NIST Security Policy](#)

Enable PCE FIPS Compliance

1. After installing RHEL7.4, follow the required steps in Section 9.1, Crypto Officer Guidance, [Red Hat Enterprise Linux OpenSSL Cryptographic Module NIST Security Policy](#).
2. Reboot the system.
3. After reboot, verify that the setting `/proc/sys/crypto/fips_enabled` is equal to 1.
4. Install the Illumio PCE RPM. See [After PCE Installation](#) for information.
5. During PCE installation, provide the PCE with SSL certificates that have a minimum RSA key size of 2048.

After completing the PCE setup, the PCE is FIPS compliant.

FIPS Compliance for Red Hat/Linux VENs

For all Illumio supported Linux workloads, the standard 18.1 GA VEN release and later support VEN Linux FIPS compliance.

Starting with the Linux VEN 18.1 release, all VEN OpenSSL communications by default operate in a FIPS compliant mode.

- FIPS is supported on the VEN 18.1 release through the 20.2 release.
- FIPS is *not* supported on the VEN 21.1 release through the 21.5 release due to the OpenSSL 1.1 upgrade.
- FIPS is supported on the VEN 22.2 release and later.

FIPS for SecureConnect

To claim FIPS compliance for the VEN SecureConnect feature (IPsec encryption between workloads), the VEN must be installed on RHEL v7.1 or RHEL v7.4 and configured to operate in FIPS mode as documented in either of the following documents:

- Section 9.1 (“Cryptographic Officer Guidance”) of the RHEL v7.1 [Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v4.0](#)
- Section 9.1 (“Cryptographic Officer Guidance”) of the RHEL v7.4 [Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v5.0](#)

FIPS Compliance for Windows VENS

For Windows workloads, the standard 18.1 GA VEN release and later support VEN Windows FIPS compliance.

Windows VEN is FIPS compliant when installed on Windows Server 2012 or Windows Server 2016.

To operate the FIPS-compliant Windows VEN, the Windows system must be configured to operate in FIPS mode as documented in Section 2 of the [Windows Server 2012 NIST Security Policy](#) or Section 2 of the [Windows Server 2016 NIST Security Policy](#).

OpenSSL 3.0 Module and RHEL 8 FIPS 140-2 Certification

OpenSSL 3.0 module and RHEL 8.0 OS are both currently undergoing certification for FIPS 140-2.

For more on the latest certification status for RHEL 7.x and RHEL 8.x, see the following NIST Cryptographic Module Validation Program (CMVP) document: [Cryptographic Module Validation Program CMVP Modules In Process List](#)

PCE Installation Troubleshooting

This chapter contains the following topics:

PCE Troubleshooting Scenarios	90
-------------------------------------	----

This section describes issues that can arise during PCE installation or upgrade and how to resolve them.

PCE Troubleshooting Scenarios

This section describes issues that can arise during PCE installation or upgrade and how to resolve them.

Session Limits Too Low

(RHEL7+ only)

Symptom:

The expected session limits, configured in `/etc/security/limits.conf`, might not be in effect for the PCE. This can cause a severe performance impact that may go unnoticed for some time.

Cause:

It typically affects systems which have a PAM authentication configuration utilizing the `loginuid` module. This type of configuration relies on `systemd` rather than the traditional security limits for the application's session limits. The issue can also arise if the documented installation preparation steps are not followed (see [Process and File Limits](#)) or if the values are later altered.

You can verify the session limits by inspecting a running PCE with the following command:

```
cat /proc/$(pgrep -f config_listener.rb)/limits | grep -e open -e processes
```

Solution:

If the output of this command shows values for `nofile` and `nproc` that are lower than required (see [Process and File Limits](#)), provide an override file to properly configure these limits.

1. Create the following file on the PCE:
`/etc/systemd/system/illumio-pce.service.d/override.conf`
2. Add the following lines to the file and save:

```
[Service]
LimitNOFILE=65535
LimitNPROC=65535
```

3. Reboot or restart the PCE.
4. Run the `grep` command above again and inspect the `config_listener.rb` session limits to ensure that they are now correct.

Database Migrations Mismatch

Symptom:

Error message “Stopping PCE software: DB migrations mismatch for DB: avenger_executor_dev: Missing migrations” when you try to bring the PCE to runlevel 5.

Cause:

Attempted to start the upgraded PCE without migrating the database. If you did not run the `illumio-pce-db-management migrate` command on the primary database node, you will not be able to bring any PCE node up to runlevel 5, and you will not be able to start the other nodes in the cluster. If some of the nodes in the cluster are already running, they will be shut down until you successfully migrate the database.

Solution:

Follow the steps in [Migrate the PCE Database](#).

Database Already Exists

Symptom:

The `illumio-pce-db-management setup` command finishes abnormally with Exit Code = 1 and displays the following type of messages:

```
Database 'traffic_prod' already exists
...
psql:/opt/illumio-pce/illumio/webservices/traffic_query/db/structure.sql:52:
ERROR:  relation "ar_internal_metadata" already exists
rake aborted!
failed to execute:
psql -v ON_ERROR_STOP=1 -q -f /opt/illumio-pce/illumio/webservices/traffic_
query/db/structure.sql traffic_prod
```

Cause:

The database has already been set up.

Solution:

If you are trying to set up a new database, you must first remove the existing database.

WARNING:
All existing data will be lost.

To remove the database and all its data, run the command `illumio-pce-db-management drop`. Then, retry the `illumio-pce-db-management setup` command.

PCE UI Missing

Symptom:

Error message displayed in the browser, such as “PCE UI Missing.”

Cause:

The PCE package has been installed without the UI package.

Solution:

Install the UI package as described in [UI-Only Upgrade](#).