Illumio Adaptive Security Platform 18.2.1 PCE Deployment Guide

01/24/2019

20000-100-18.2.1

Table of Contents

Product Version

Illumio® Adaptive Security Platform®

Current PCE Version: 18.2.1

Current VEN Version: 18.2.1

Note: 18.2.1 has not been designated as a Long Term Support (LTS) release. In the future an 18.2.x LTS release will be designated.

About Illumio

Illumio products and services are built on Illumio's patented technologies. For more information, see Illumio Patents.

Illumio Professional Services for Deployment

To ensure optimal deployment of the Illumio Adaptive Security Platform, contact your Illumio Professional Services representative.

Preview Features Only for Evaluation Before General Availability

Any preview features in this release of Illumio Adaptive Security Platform are for your evaluation only.

> **Do not deploy preview features in a production environment**
> Be sure to install these preview features only on non-production systems. To avoid inadvertently impacting your current operations, do *not* install the preview features on production systems.
> The purpose of preview features is to make them more useful for your needs before general availability.

Illumio welcomes your comments and suggestions for improving preview features and documentation. For more information and to send feedback, contact Illumio Customer Support.

Illumio Adaptive Security Platform Training

Illumio offers a wide yet focused training curriculum for Illumio Adaptive Security Platform, from beginning to advanced topics.

To see available courses, log into your Illumio support account and select the **Training** tab.

Search Knowledge Base and Documentation

For useful short articles about Illumio Adaptive Security Platform, log into your Illumio support account and select the **Knowledge Base** or **Documentation** tab.

Illumio Adaptive Security Platform Support

If you cannot find what you are looking for in this document or in support Knowledge Base and Documentation, contact us at:

- support@illumio.com
- +1-888-631-6354
- +1-408-831-6354

Recommended Skills

Illumio recommends that you be familiar with the following:

- Your organization's security goals.
- General knowledge of Illumio Adaptive Security Platform.
- General computer system administration of Linux and Windows operating systems, including startup/shutdown, 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.

Related Documentation

Illumio® Adaptive Security Platform® documentation is available from the Support portal.

- *Policy Compute Engine (PCE) Web Console Guide*: working with Illumination®, designing security policy, and provisioning and administering VENs.
- *Policy Compute Engine (PCE) Deployment Guide*: planning and installing the PCE.
- *Policy Compute Engine (PCE) Operations Guide:* common management tasks of the PCE.
- *Advanced Command-line Tool Interface Guide*: common PCE-related tasks to use on your local computer.
- *Policy Compute Engine (PCE) Supercluster Deployment and Usage Guide*: designing, deploying, and managing the PCE Supercluster of multiple, distributed standard PCE clusters.
- *Policy Compute Engine (PCE) REST API Guide*: web-programming Illumio Adaptive Security Platform.

- *Virtual Enforcement Node (VEN) Deployment Guide:* installing and activating the VEN, including PCE-based distribution of the VEN and on-worload installation and management
- *Virtual Enforcement Node (VEN) Operations Guide*: common management tasks of the VEN.
- *Auditable Events and SIEM Integration Guide*: analyzing significant events on the PCE and VEN and securely transferring event records to analytics or Security Information and Event (SIEM) systems.

Notational Conventions

- 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:
  ```
  ...
  some command or command output
  ...
  ```
- References to section titles in this guide are in double quotation marks. Example: See "Basic Theory of Operation".
- Reference to other guides in the Illumio library are *italicized*. Example: See the *PCE Web Console User Guide*.

How to Use This Guide

This guide has several high-level divisions:

- Conceptual overview.
- Sections on deployment planning and prerequisites.
- Downloading and installing the PCE.
- Additional deployment tasks.

Overview to PCE Deployment

This document describes the general process and tasks for deploying the on-premises Policy Compute Engine (PCE).

PCE Multiple Node Clusters

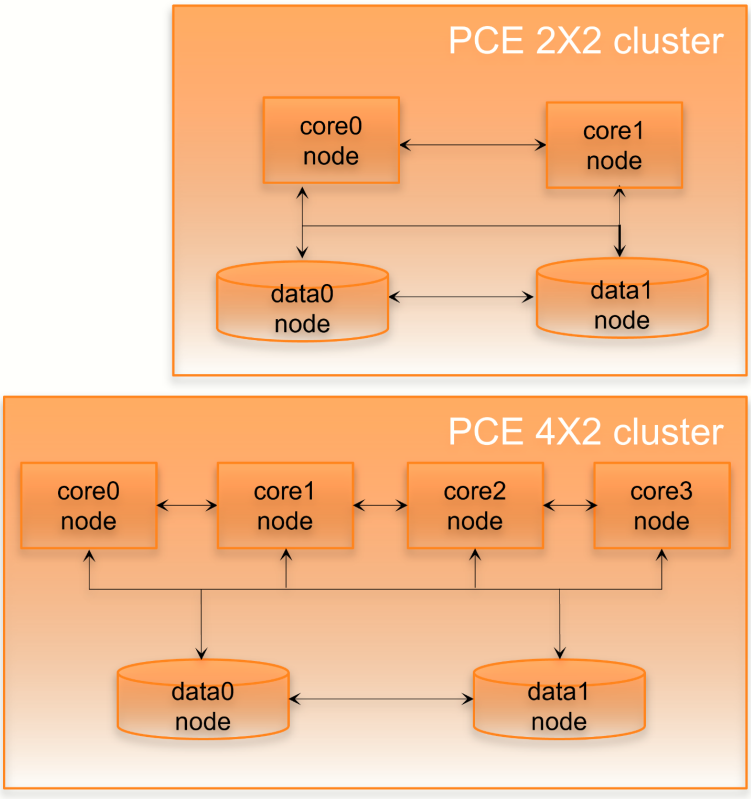
A *PCE node* is a single host (server or VM) that runs the PCE. Each node in the cluster is configured by its node type, which defines its services:

- Core node, known as Core0, Core1, Core 2, and Core 3.
- Data node, known as Data 1 and Data 2.

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

- 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 4X4 cluster*.

PCE Multi-node Cluster Types



PCE Deployment Planning and Prerequisites

Planning Checklist

Below is a checklist planning your deployment. These details 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	Reserved Port Ranges for PCE Cluster Communications
Load balancer setup	Load Balancer Requirements
DNS domain name setup	DNS Requirements
Mail software	SMTP Requirements
TLS setup, including SSL certificate types and settings	• TLS (SSL) Requirements • Negotiation of TLS versions • Optional -- validate your TLS/SSL certificate
OS package dependencies, libraries, NTP, IPTables, UTF-8, Trusted CA, syslog, process and file limits, and kernel parameters	Operating System Setup and Package Dependencies
Download, install, and configure the PCE software	• Download the PCE • Install the PCE • Configure the PCE
VEN deployment planning	See "VEN Deployment Models" in the VEN Deployment Guide.

Upgrade paths and planning tool

For details on upgrade paths for versions of the PCE and VEN, see Versions and Releases on the Illumio support site.

An upgrade planning tool is also available to help you plan your deployments.

PCE Capacity Planning

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

Exact requirements vary on a large number of factors, including, but not limited to:

- Number of managed workloads.
- Number of unmanaged workloads and other labeled objects, such as Bound Services.
- Policy complexity, which includes the following:
  - 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 workload 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.
- Total number of unique flows reported to the PCE from all VENs.

Recommended vs minimum sizes

The capacity planning table below shows minimal and recommended sizes. Illumio encourages you to plan for the recommended sizes. In addition, based on your actual usage and the various factors listed above, your capacity needs might be even greater than the recommended sizes.

There are two configurations for data nodes:

1. A single storage device shared between the data nodes.
2. A dedicated storage device for each data node. This configuration is to accommodate growth in traffic data, which is used by the Explorer. See also "PCE Storage Device Partitions".

MNC Type + Workloads/ VENs[1]	Cores/Clock Speed[2]	RAM per Node[3]	Storage Device Size[4] and IOPS[5]	
			Core Nodes	Data Nodes

2X2
• 2,500 VENs
• 12,500 workloads
	• Four cores per node.
	• Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or equivalent.
		32 GB	• 1 x 100 GB
			• 100 IOPS
				• Recommended:
				  • 2 x 250 GB
				  • 600 IOPS per device
				• Minimum:
				  • 1 x 250 GB
				  • 600 IOPS

2X2
• 10,000 VENs
• 50,000 workloads
	• 16 cores per node
	• Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or equivalent.
		• Recommended: 128 GB
		• Minimum: 64 GB
			• 1 x 200 GB
			• 100 IOPS
				• Recommended:
				  • 2 x 1 TB
				  • 1,800 IOPS per device
				• Minimum:
				  • 1 x 1 TB
				  • 1,800 IOPS

4X2 • 25,000 VENs • 125,000 workloads	• 16 cores per node • Intel® Xeon(R) CPU E5-2695 v4 at 2.10GHz or equivalent.	128 GB	• 1 x 200 GB • 100 IOPS	• 2 x 1 TB • 5,000 IOPS per device

Footnotes

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

[2] CPUs:

- The recommended number of cores is based only on physical cores from allocated CPUs, irrespective of hyper-threading or virtual cores. For example, in AWS one vCPU is only a single hyperthread running on a physical core. that is. half a core. So 16 physical cores equates to 32 vCPUs in AWS.
- Full reservations for vCPU. No overcommit.

[3] Full reservations for vRAM. No overcommit.

[4] Storage requirements for network traffic data can increase rapidly as the amount of network traffic increases. Allocating a separate, large storage device for traffic data can accommodate these rapid changes without potentially interrupting service.

[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.

For more than 100 IOPS, either mixed-use Solid-State Disk (SSD) or Storage Area Network (SAN) is required. Locally attached, spinning hard disk drives (HDD) are not sufficient.

PCE Storage Device Partitions

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

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

Device	Partition mount point	Size to Allocate	Node Types	Notes
Device 1	`/`	16 GB	Core, Data	
Device 1	`/var/log`	32 GB	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	`/var/lib/illumio_pce`	Balance of Device 1	Core, Data	
Device 2 in two-storage-device configuration	`/var/lib/illumio_pce/data/traffic_datastore`	All of Device 2 (250 GB)	Data	For network traffic data, in a two-device configuration for the data nodes, this should be a separate device that is mounted on this directory. `/var/lib/illumio_pce/data/traffic_datastore` is the default value of network traffic datastore's `traffic_datastore:data_dir` runtime environment setting. When you change the defined path in this parameter, make sure that the new value matches the path you actually mount.

PCE 2x2 Multi-Node Cluster for 10,000 VENs or PCE 4x2 Multi-Node Cluster for 25,000 VENs

Storage Device	Partition mount points	Size to Allocate	Node Types	Notes
Device 1	`/`	16 GB	Core, Data	
Device 1	`/var/log`	32 GB	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	`/var/lib/illumio_pce`	Balance of Storage Device 1	Core, Data	
Device 2 in two-storage-device configuration	`/var/lib/illumio_pce/data/traffic_datastore`	All of Storage Device 2 (1 TB)	Data	For network traffic data, in a two-device configuration for the data nodes, this should be a separate device that is mounted on this directory. `/var/lib/illumio_pce/data/traffic_datastore` is the default value of network traffic datastore's `traffic_datastore:data_dir` runtime environment setting. When you change the defined path in this parameter, make sure that the new value matches the path you actually mount.

Runtime parameters for Two-Storage-Device Configuration

In the two-storage-device configuration, to accommodate growth in the traffic datastore, set the following parameters in `runtime_env.yml`.

If you are deploying the two-device configuration, you must set these parameters.

```
traffic_datastore:
    data_dir: path_to_second_disk
```

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

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

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

The following are recommended values for these parameters based on cluster type and estimated number of workloads.

Setting	PCE 2x2 Multi-Node Cluster for 2,500 VENs	PCE 2x2 Multi-Node Cluster for 10,000 VENs	PCE 4x2 Multi-Node Cluster for 25,000 VENs	Note
`traffic_datastore:max_disk_usage_gb`	100 GB	400 GB	400 GB	This size reflects only part of the required total size, as detailed in "PCE Capacity Planning".
`traffic_datastore:partition_fraction`	0.5	0.5	0.5	
`traffic_datastore:time_bucket_type`	day	day	day	

Reserved Port Ranges for PCE Cluster Communications

The following port ranges are needed for communications among the nodes of the PCE cluster.

Protocols	Port Range
TCP	3100 to 3600
TCP	5100 to 6300
TCP and UDP	8000 to 8400
TCP	11200 to 11300

Load Balancer Requirements

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

Program the load balancer with the Illumio REST API to monitor the PCE's health check and determine if the cluster core nodes are available. See the *REST API 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 healthy and connected to the rest of the cluster. Any other status code or no response means the node is unhealthy and cannot accept requests. Unhealthy or unresponsive nodes should be removed from the load balancing pool.

- There can be up to a 30 second delay for the health check API 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.
- If you are using a DNS load balancer to handle traffic to the PCE, the DNS must be able to run health checks against the `/node_available` API, and the DNS load balancer should only serve IP addresses for the cluster FQDN of those nodes that respond to the `/node_available` API.

IP Address

A statically-assigned IP address is highly recommended. By default, the PCE uses the first available private IP address you define.

> If you are using a public IP address or if the node has multiple interfaces, you need to configure the PCE to use a different private IP address. For assistance, contact Illumio Customer Support.

To configure networking, see your OS vendor's documentation on the `ifcfg-eth`*`N`* `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 VIP(s) 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 (SSL) 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 this Illumio Support KB article for instructions.

For an in-depth discussion of deploying the PCE with TLS, see this this KB titled Preparing Certificates for a PCE deployment.

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. T

> ⚠ 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:

1. The file must contain PEM-encoded certificates.
2. The certificate's signature algorithm must be SHA256WithRSAEncryption.
3. The certificate's signature algorithm must ***not*** be RSASSA-PSS.
4. 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.
        i. 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.
        ii. If the certificate is generated by a major Public CA (e.g., VeriSign, GeoTrust, Entrust, 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 cert request process, you'll need to open the file up in a text editor and reverse the order of the certs. Apache always expects the root cert 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-----
```

5. All certificates in the bundle must be valid for the current date. Note that this depends on the system time being set correctly.
6. A trusted root store must be available for OpenSSL to validate certificates.
7. The certificate must match the PCE FQDN. This can be an exact match (e.g., pce.mycompany.com) or a wildcard match (e.g., *. mycompany.com).

The certificate must support both Server and Client authentication. Client authentication is used between nodes in a multi-node cluster. 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, Use SHA256WithRSEencryption

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

The PCE certificate requires the SHA256WithRSEencryption signature.

> ⚠ If you use Microsoft Certificate Authority (CA) to sign PCE certificates, make sure to use the SHA256WithRSEencryption. PKCS#1 version 2.1 is enabled by default on Microsoft CAs and thus 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:

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

Negotiation of TLS Versions for Communications

The PCE negotiates the use of Transport Layer Security (TLS) versions 1.0, 1.1 or 1.2 for VEN-to-PCE communications, the PCE's web server for the PCE web console, and the REST API. The PCE selects the highest version that the your workloads support.

- The PCE default minimum version is TLS 1.0.
- For VEN versions 18.1 and later, all VENs are use TLS 1.2.
- SUSE VEN version 17.1.x 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 via the Registry.

Changing Default TLS version

Except VEN 17.1 for SUSE, which requires TLS 1.0, you can use TLS 1.0, 1.1 or 1.2 with any version of the VEN. 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 only TLS 1.2.

Optionally 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 the section "Single Sign-On" in the *PCE Web Console Guide*. The guide has step-by-step details for a wide variety of IdPs.

Operating System Setup and Package Dependencies

The PCE supported operating systems, with package dependencies, are on the support site.

NTP

Set up a Network Time Protocol (NTP) client for time synchronization.

To install and configure NTP, run the following commands:

```
# yum install ntp # Install ntp module
# date # Verify that the timezone is set correctly. If wrong, fix the timezone with timedatectl set-timezone
```

```
# systemctl enable ntpd # Set NTP to start at boot
# service start ntpd # Start the ntpd daemon
# chkconfig ntpd on # Verify the ntpd daemon configuration
```

IPTables

For the initial installation, you might want to 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 following files:`

- RHEL 6: `/etc/sysconfig/i18n`
- RHEL 7: `/etc/locale.conf`

Trusted Public Certificate Authority (CA) Store

A trusted root public 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
```

If your certificate is signed by a private CA or if 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 "Optionally configure PCE internal syslog".

Process and File Limits

For best performance, modify the parameters detailed here in the `/etc/security/limits.conf` file for each node.

Core Nodes values in limits.conf

Failure to set these values correctly can severely impact system performance.

- If your settings are already greater than these, you do not need to reduce them to these values.
- If you have automated processes that change these values, adjust those processes to not change them.
- To restrict this change to only the PCE runtime user, then replace the asterisk shown below with the Unix user-id of the defined PCE runtime user.
- If you run additional processes on the PCE, such as monitoring or other operations processes, you might need to increase the value of `nofile`.

```
* soft    core            unlimited
* hard    core            unlimited

* hard    nproc           65535
* soft    nproc           65535

* hard    nofile          65535
* soft    nofile          65535
```

Data Nodes values in limits.conf

```
* soft    core            unlimited
* hard    core            unlimited
```

Core Nodes values in 90-nproc.conf or 20-nproc.conf

If the `/etc/security/limits.d/90-nproc.conf` for RHEL6 or `20-nproc.conf` for RHEL7 file is configured on your system, you must also change its `nproc` values.

Be sure there are no additional configuration files in `/etc/security/limits.d` that might override the recommended limits.

```
* hard    nproc           65535
* soft    nproc           65535
```

Kernel Parameters in sysctl.conf

For optimal performance of the PCE, set the following kernel parameters for each node.

If your settings are greater than these, you do not need to lower them.

Parameters are configured in the `/etc/sysctl.conf` file. After the settings are configured, apply them to the kernel with the following command. Otherwise, the changes take effect at the next boot.

```
# sysctl -p
```

Core Nodes in sysctl.conf

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

Data Nodes in sysctl.conf

```
 fs.file-max          = 2000000
kernel.shmmax        = 60000000
vm.overcommit_memory = 1
```

About Your Organization Name and Organization ID

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 Adaptive Security Platform 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.

Download the PCE

Download the software from the Illumio Support site.

Install the PCE

The PCE RPM is the easiest way to install the software if you can use the default directory locations and runtime user account (`ilo-pce`).

As root, run this command to install the PCE on each of the nodes in your deployment:

```
# rpm -ivh /path_to/pce_rpm_file
```

After the installation and and configuration of the PCE, you do not need to run the PCE as root.

After installing the RPM, configure with the PCE setup wizard. See "Configure 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-xr-x. root root`
`/etc/init.d/illumio-pce`	Service script	`-rwxr-xr-x. root root`

`/var/lib/illumio-pce/`
`tmp/`
`runtime/`
`data/`
`keys/`
`cert/`
	Empty	`drwxr-x---. root ilo-pce`
		`drwx------. ilo-pce ilo-pce`
		`drwx------. ilo-pce ilo-pce`
		`drwx------. ilo-pce ilo-pce`
		`drwx------. ilo-pce ilo-pce`
		`drwx------. ilo-pce 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 `ilo-pce` to run the PCE software. For security, the `ilo-pce` user is configured without a login shell or home directory.

> **No login shell or home directory**
> For better security, do not give the `ilo-pce` user a login shell or home directory.

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

```
# sudo -u ilo-pce somePCEcommand
```

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

PCE Control Interface illumio-pce-ctl 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.

The PCE also includes two other command-line utilities used to setup and operate your PCE:

- `illumio-pce-env`. Used for verifying and collecting information about the PCE runtime environment.
- `illumio-pce-db-management`. Used for PCE database management.
- `supercluster-sub-command`. Used for Supercluster specific operations.

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

In this document, all command-line examples assume an RPM installation. If you installed the PCE tarball, you will need to modify the commands based on your PCE user account and the directory where you installed the software.

Control command access via /usr/bin. By default, for easier command execution, the installation of the PCE creates softlinks in `/usr/bin` for the Illumio PCE control commands. The `/usr/bin` directory is usually included by default in the PATH environment variable in most Linux systems. If for some reason 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

In this document, all command-line examples assume a RPM installation. If you installed the PCE tarball, you will need to modify the commands based on your PCE user account and the directory where you installed the software.

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

```
$ 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` is switches to the runtime user (`ilo-pce`) prior to running other PCE program. 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}
```

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 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
```

Configure the PCE

After the basic installation, configure the PCE.

Before running the PCE, be sure it is properly configured with a runtime configuration.

The PCE Runtime Environment File (`runtime_env.yml`) is used to configure the PCE software. The default location of this file is `/etc/illumio-pce/runtime_env.yml`. You can override this location by setting the `ILLUMIO_RUNTIME_ENV` environment variable. You can create the `runtime_env.yml` file manually or use the PCE software setup script to create and modify the file.

You are prompted to provide these parameters during the setup.

Essential Reading: Complete Details on Runtime Environment File Parameters

For detailed descriptions of the runtime parameters, see "Reference: Runtime Environment File Parameters".

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
```

When you first launch the setup script from the command prompt, the script will indicate if the `$ILLUMIO_RUNTIME_ENV` environment variable is set:

These first two screens will only appear if you launch the setup script from the command line (i.e., you installed directly from RPM and did not use the ISO).

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

The `ENV` environment variable indicates that the new configuration will be written to the file defines for `ILLUMIO_RUNTIME_ENV`. If the `ILLUMIO_RUNTIME_ENV` environment variable is not set, the setup will display

that this is a new configuration and write the `runtime_env.yml` file to the default location `/etc/illumio-pce/runtime_env.yml`.

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

General Configuration

The setup script displays any descriptive help text followed by a prompt where you can either accept the previous or default value, or enter a new value. If the field is optional, pressing Enter on your keyboard will clear the field from view if the resulting value is empty. If instead there is a corresponding default value it displays `# default` next to that value.

The prompt itself encapsulates the previous value in brackets:

```
node_type [core]:
```

Pressing Enter will keep the value in brackets. Any previously-set value always takes precedence at the prompt; e.g., if there's a previous value, it will be displayed instead of any default one.

If you are unsure whether the value displayed by the prompt is a previously set or default or recommended value, you can enter a question mark (`?`). This will display the default or recommended value, if available:

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

If there are multiple options, you may use the auto-complete functionality by typing the first few characters and pressing Tab on your keyboard to auto-complete or suggest any remaining choices. When the prompt is for a directory or filename, you may use the autocomplete function to more 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 will save the configuration to a different file and quit 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

The --batch option is used to operate the setup script from the command-line. Instead of prompting for each value, it automatically accepts any previous/default value automatically. If there are 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 sets these values instead of prompting for them. You can also pre-set these values in non-batch mode by using key=value arguments.

> **Batch mode creates new configuration file**
> Batch mode automatically saves the new configuration unless there is an error.

The `--list` option also does not prompt for values. It displays the currently configured values, replacing them with any specified command-line values. The `--list` option does not save the configuration to the `runtime_env.yml` file. The `--list` option is useful to validate your TLS/SSL certificate.

Advanced Runtime Environment Parameters

Your Illumio support representative may provide you with certain advanced parameters to add to your `runtime_env.yml` file. If you include the name of these parameters on the command line, the setup script will prompt 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
`-b, --batch`	Don't prompt for field values.
`-d, --default`	Show default values.
`-e, --empty`	Display empty fields (implies `-d`).
`-f, --field *[:*][,...]`	Specify a field pattern list; only process these items.
`-g, --[no-]guide`	Show descriptive information for each field where available (default).
`-h, --help`	Provide usage statement.
`-m, --macros`	Show list of available shortcut keys.
`-o, --[no-]optional`	Process optional fields (default).
`-q, --quiet`	Don't display help text for each field (same as `--no-guide`)
`-r, --reveal`	Don't mask secret key(s) in field output.
`-t, --text`	Use regular text instead of colors.

File Options

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

Optionally Validate and Configure the TLS Certificate and Private Key

Your TLS certificates are validated at start-up of the PCE. An error message is displayed if the certificate or its chain of trust is not valid.

For information on the contents and formats of your certificates see "TLS Requirements".

You can validate the certificates yourself, either before or after configuring the PCE as described in "Configure the PCE with the Setup Script".

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

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

The `--list` option does not create a new `runtime_env.yml` configuration file, which is created when you configure the PCE as discussed in "Configure the PCE with the Setup Script". Insteadthe command runs a series of checks on your configuration, including certificates, and gives a more complete indication of possible problems.

Validate after configuring PCE

If you have already configured your certificates in the locations defined in the `runtime_env.yml` file, as described in "Configure the PCE with the Setup Script", you can validate with the following command. The `--test` option takes a verbosity level argument, which is from 1 (least verbose) to 5 (most verbose). With verbosity level 5, the command displays the results of its validation of your certificates.

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

Alternative syntax for certificate validation after configuring the PCE

Additional mechanisms for certificate validation include:

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

  This syntax checks the certificate and chain against the specified *`some.alternative.hostname`*, 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 without runtime_env.yml file before configuring PCE certificates

If you have not yet configured your `runtime_env.yml` file, discussed in "Configure the PCE with the Setup Script", and want to validate your certificates before copying them to your planned production location, use 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
`email=`*`required`*`@`*`emailaddress`*	Required. Your email address.
`node=`*`value`*	Topology to check. For allowable values, see the parameter `node_type` in "Optional Runtime Parameters" and see the discussion in "PCE Multiple Node Clusters".
`cert=`*`/path/to/cert`*	The absolute path to your certificate.
`pkey=`*`/path/to/private_key`*	The absolute path to your certificate's private key.
`trust=`*`/path/to/certificate_chain`*	The absolute path to your certificate's CA chain of trust.

Messages for valid certificates, errors, and warnings

Correctly configured certificates are indicated by these messages:

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

Possible problems with the certificates are indicated by error messages such as the following:

- `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 of the 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

After configuring the `runtime_env.yml` file, run the environment check command to ensure the node is properly set up.

As the PCE runtime user, run the following command:

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

Correct any errors before proceeding.

PCE Start

As the PCE runtime user, perform the following steps:

1. **On each node**, start the PCE at runlevel 1.

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

2. **On each node**, check system status. Make sure the node status is RUNNING before proceeding to the next step. It can take up to 10 minutes for the various services to start.

   ```
   # sudo -u ilo-pce illumio-pce-ctl status
   Checking Illumio Runtime            RUNNING 0.38s
   ```

   If the node does not come up properly after 10 minutes, check the following:
   a. Runtime environment file
   b. Network connectivity between nodes/iptables
   c. Certificates
   d. System locale (must be UTF-8)

Initialize the PCE

As the PCE runtime user, perform the following steps:

1. **On any node**, run the following command to 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
   ```

   Important: 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. 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.

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

   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. Point a web browser to the PCE FQDN and log in using the account you just created.
6. The PCE is now up and running.

Additional Deployment Tasks

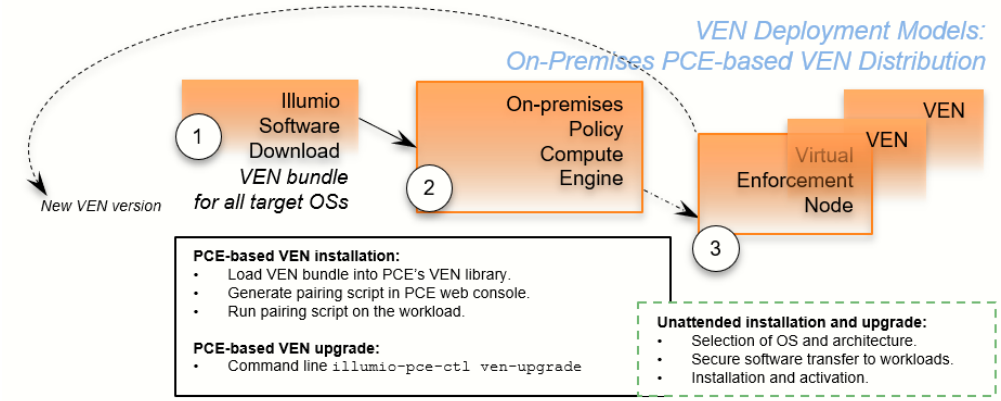
VEN Deployment Models

The VEN has two deployment models. The two models for VEN deployment are nearly identical and achieve the same goal: VEN installation and upgrade.

- Integrated VEN deployment from an on-premises PCE. This is called *PCE-based VEN deployment*.
- Manual VEN installation on individual workloads with your own software deployment tools. This is called *standalone VEN installation*.

On-Premises PCE-Based VEN Deployment

The PCE-based VEN deployment model is more automated than the standalone VEN deployment model but gives you less control over optional aspects of VEN installation and upgrade.



The PCE-based deployment model starts with a *VEN software bundle*. A VEN software bundle is a collection of a particular VEN software version for all supported workload OSs.

- On the on-premises PCE, you load a VEN software bundle into the *VEN library* . The VEN library is a collection of all VEN software versions you have loaded.
- For VEN installation:
    - In the PCE web console, you generate a pairing script to install and activate the VEN on target workloads.
    - You copy the pairing script to the target workload and run it.
    - The pairing script:
        - Determines the OS and CPU architecture of the target workload.
        - Securely transfers the VEN software to the target workloads.
        - Installs the VEN software.
        - Activates/pairs the VEN with its PCE.
- For VEN upgrade, on the on-premises PCE command-line, you run `illumio-pce-ctl ven-upgrade` for either all workloads or selective workloads.

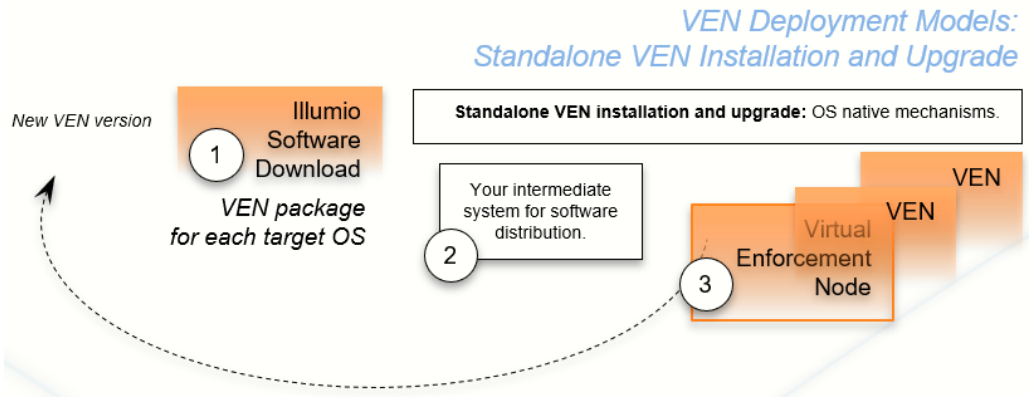
- Some features are not available with PCE-based deployment, such as Kerberos support and custom settings with environment variables.

The PCE-based deployment feature is:

- Optional.
- Available for the RPM, Debian, and Windows distributions of the VEN software. Other workload operating systems are not supported.
- Available only for PCE and VEN version 18.2 and later.

Standalone VEN Installation and Upgrade

It gives you great control over optional aspects of VEN installation, activation, and upgrade.



The standalone VEN installation model starts with downloading a *VEN package*. A VEN package is the VEN software for a single supported workload OS and CPU architectures. Installation and upgrade rely on standard native OS tools.

- For VEN installation with the standalone model:
  - You determine the OS and CPU architecture of the target workloads and download the appropriate single VEN packages.
  - You are responsible for securely transferring the VEN software to the target workload with your own software deployment mechanisms.
  - You can set environment variables or command-line options for custom installation directories and custom user and group names. You can also set up Kerberos-based authentication for VEN to PCE communications.
  - You run native OS mechanisms.
  - You activate/pair the VEN with its PCE either during or after installation.
  - You can use a "prepare script" to install the VEN software on machine images and activate it at the next boot.
- For VEN upgrade, with the workload command line, you run native OS mechanism.

For more information, see the *VEN Deployment Guide*.

Configure PCE backups

You should maintain and perform regular backups of the PCE database based on your company's backup policy. Additionally, always backup your PCE database before upgrading to a new version of the PCE.

As the PCE runtime user, run this command to back up the PCE database to a file:

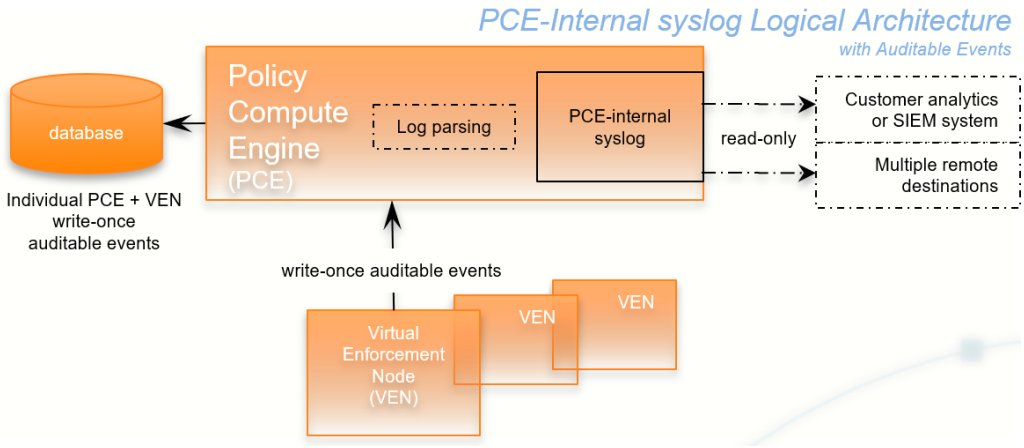
```
# sudo -u ilo-pce illumio-pce-db-management dump –-file <location-of-db-dump-file>
```

Optionally configure PCE internal syslog

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.

Smooth transition from existing syslog installations is achieved by a default configuration called "Local" . Via this default, the PCE internal syslog relay messages to existing syslog.



It is particularly useful in conjunction with the PCE's auditable events data. See the *Auditable Events and SIEM Integration 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.

Recommendation – Do Not Write Any Additional Information to log_dir

You can put the PCE internal syslog into operation while still running any syslog implementation you already have.

> **Do not store auditable events in log_dir**
> If you continue to use a previously configured syslog (prior to Illumio Adaptive Security Platform 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/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 new auditable event information (first released with Illumio ASP version 18.2) 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.

Configuring Events and syslog in the PCE Web Console

For details, including configuring remote syslog destinations, see the P*CE Web Console User Guide*, section "Settings > Events".

Optional – Setting Path to Custom TLS Certificate Bundle in runtime_env.yml

If you enable TLS mutual authentication, the channel to the remote syslog destination can be secured by your own Transport Layer Security (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.

- If a custom TLS bundle is provided by the user during configuration, this bundle is used for certificate verification.
- If 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: Secure Syslog Data Transport and Storage

Enabling Transport Layer Security (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, you should ensure restricted access to the data by relying on the OS-level user access mechanisms. In addition, you should limit the number of users allowed access to the syslog storage itself. If possible, rely on an enterprise-class log management system for post-processing the event data.

Remote Destination: 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
...
```

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.

Remote Destination: 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 details.

PCE Upgrade/Downgrade

This section provides information on how to upgrade or downgrade the PCE . Important considerations before you begin:

- For upgrade, you should directly invoke the `illumio-pce-ctl` control script. For example:

  `$ **sudo -u ilo-pce illumio-pce-ctl *command***`
  Do not use the `service illumio-pce start` or any service commands when upgrading. The service command is designed to be run without prompting, which is required for certain upgrades, so do not use any the PCE service commands during this upgrade process.
- After upgrading the PCE version, the `illumio-pce-db-management migrate`' command must be run on any node before bringing the cluster to runlevel 5.
- Make sure you upgrade all nodes in the cluster to the same version before restarting the nodes; otherwise, none of the nodes in your cluster will start.
- 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.
- Check to ensure that any asynchronous jobs have not been submitted right before you plan to do the upgrade. As a general best practice, you should wait until all asynchronous jobs have finished before upgrading the PCE.
- For a multi-version upgrade, in the following "Prepare for Upgrade" section, the 'Backup PCE Database and Current Software' tasks below should only be done a single time at the beginning of the first upgrade sequence. This allows you to rollback to the starting version if there is an issue with the upgrade.

Backup the PCE

1. Before you begin the backup, you need to determine the Data node that requires a backup. To find out which node runs this service, use the `illumio-pce-ctl cluster-status` command:

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

   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          <=== dump command should run
   from this node
   agent_traffic_service NOT RUNNING
   ...
   ```

2. Run one of the following commands on the Data node that is running the `agent_traffic_redis_server` service.

   For upgrading, you need to only run the first command.

Both Policy and Traffic Databases

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

Only Traffic Database

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

3. After the commands complete, 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 Supercluster. The default location of the PCE Runtime Environment File is `/etc/illumio-pce/ runtime_env.yml`.

Upgrade the PCE

The upgrade process includes these general steps:

- Upgrade the PCE with RPM or Tarball
- Remove older events version 1 records from the database
- Migrate the PCE database

Stop the PCE

On **each** node in the cluster, stop the PCE .

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

Upgrade RPM Installation

On **each** node in the cluster, upgrade to the new PCE RPM version:

```
$ rpm -Uvh <illumio_pce_rpm>
```

Update PCE Runtime Environment File

Consult the Release Notes to determine if any changes to the PCE Runtime Environment File (`runtime_env.yml`) are required to upgrade. If changes are required:

1. On **each** node in the cluster, update the `runtime_env.yml` file.
2. On **each** node in the cluster, check the validity of the `runtime_env.yml` file by running the following command:

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

Migrate the PCE Database

Start the PCE at Runlevel 1 (Database Operations Only)

1. On **each** node in the cluster, start the PCE at runlevel 1:

   ```
   $ 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, when asked if you want to continue the upgrade, type `yes` and then Enter on your keyboard.

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

   ```
   The PCE software is running a newer version(9.6.1) of the postgres software than
   the database version(9.3.)
   The PCE software will upgrade the database to the newer release.
   Prior to this upgrade, Illumio recommends that you make a backup/copy of your /
   var/lib/illumio-pce/data directory

   Do you wish to continue with the database upgrade. [yes/no]: yes
   Proceeding with database upgrade
   ```

3. On **each** node in the cluster, verify the PCE status by running these commands:

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

4. On any node, run this command to migrate the database to the latest schema version:

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

Bring the PCE to Runlevel 5 – Full Operation

1. Set runlevel 5 to bring the cluster to a running state:

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

   Note: If you did not run the '`illumio-pce-db-management migrate`' command on the database master node in the first step of this task, you will not be able to bring the node up to level 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, then they will be shutdown 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.
   ```
2. On each node in the cluster, verify the PCE status by running these commands:

   ```
   $ sudo -u ilo-pce illumio-pce-ctl status -s -v -w
   $ sudo -u ilo-pce illumio-pce-ctl cluster-status -w
   ```
3. If you are using 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. From PCE web console, log in and verify VEN synch status is showing as "Verified" for a few randomly selected Workloads.
5. You can view a Workload's VEN policy status by selecting a Workload's details page.
6. Under the section VEN, make sure that Policy Sync shows "Verified." Illumio recommends checking a few randomly selected Workloads to verify policy sync for the VEN.

Downgrade/Rollback to a Previous Version

This section describes the tasks necessary to roll back the PCE to a previous version in the event of a PCE upgrade failure or defect.

Stop the PCE

On **each** node in the cluster, stop the PCE .

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

Downgrade RPM Installation

On **each** node in the cluster, run this command:

```
$ rpm -Uh <illumio_pce_rpm> --oldpackage
```

Downgrade Tarball Installation

On **each** node in the cluster, run this command:

```
$ mv <install_root_previous_release> <install_root>
```

For example:

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

Revert PCE Runtime Environment File

If you made changes to the `runtime_env.yml` file, restore the previous version of the file:

For example:

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

Remove PCE Data

On **each** node in the cluster, run this command:

```
$ rm -rf /var/lib/illumio-pce/data/*
```

Start the PCE at Runlevel 1 (Database Operations Only)

1. On **each** node in the cluster, start at runlevel 1:

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

2. On **each** node in the cluster, verify the PCE status by running these commands:

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

Revert the PCE Data

On **one** of the Data nodes of the cluster, run this command to restore the backup you took at the beginning of the upgrade:

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

Copy the restored Illumination® data file to the **other** Data node. The file is located in the following directory:

```
/var/lib/illumio-pce/data/redis/redis_traffic_0_master.rdb
```

Migrate the PCE Database

On **one** of the Data nodes of the cluster, run this command to migrate the database to the latest schema version:

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

Bring the PCE to Runlevel 5 – Full Operation

1. Set runlevel 5 to bring the cluster to a running state:

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

Note: If you did not run the '`illumio-pce-db-management migrate`' command on the database master node in the first step of this task, you will not be able to bring the node up to level 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, then they will be shutdown until you successfully migrate the database. If you attempt to start the upgraded PCE cluster without migrating the database, you will see this error:

```
$ 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.
```

2. On each node in the cluster, verify the PCE status by running these commands:

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

3. If you are using 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. From PCE web console, log in and verify VEN synch status is showing as "Verified" for a few randomly selected Workloads.
5. You can view a Workload's VEN policy status by selecting a Workload's details page.
6. Under the section VEN, make sure that Policy Sync shows "Verified." Illumio recommends checking a few randomly selected Workloads to verify policy sync for the VEN.

Reference: Runtime Environment File Parameters

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

Relation to setup script: illumio-pce-env setup

At configuration of the PCE with the `illumio-pce-env setup` script, you are prompted for many of these parameters. See "Configure the PCE".

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 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 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.

Runtime Environment File Parameter	Description	Exposure Level
`enabled_preview_features`	Includes sub-parameters to enable identified preview features	
`install_root`	The full path to the location of the PCE binaries and scripts. The software does not write to any files in this directory, so it can be read-only. For example: `install_root: /opt/illumio-pce`	Public Stable
`runtime_data_root`	The full path to the location where the PCE writes runtime data. 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. For example: `runtime_data_root: /var/lib/illumio-pce/runtime`	Public Stable

Runtime Environment File Parameter	Description	Exposure Level
`persistent_data_root`	The full path to the location where the PCE writes persistent data. This data must persist across reboots for the software to work properly. 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. For example: `persistent_data_root: /var/lib/illumio-pce/data`	Public Stable
`ephemeral_data_root`	The full path to the location where the PCE writes temporary files. 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. **Note**: Illumio does not recommend using '`/tmp`' due to the '`tmpwatch`' utility on RHEL/CentOS 6. For example: `ephemeral_data_root: /var/lib/illumio-pce/tmp`	Public Stable
`log_dir`	The PCE software writes some text file logs to this directory (although most PCE services log to syslog). `logrotate` (or similar) should be used to manage these files. For example: `log_dir: /var/log/illumio-pce`	Public Stable

Runtime Environment File Parameter	Description	Exposure Level
`pce_fqdn`	The Fully Qualified Domain Name (FQDN) of the PCE cluster. For example: `pce_fqdn: pce.mycompany.com`	Public Stable
`cluster_public_ips: cluster_fqdn`	The FQDN of your entire cluster. **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.	Public Stable
`web_service_certificate`	Full path to the X.509 public certificate used by this node for Transport Layer Security (TLS). See the 'Certificate Requirements' section above for more information on the contents of the certificate files. For example: `web_service_certificate: /etc/pki/tls/certs/my_cert.crt`	Public Stable

Runtime Environment File Parameter	Description	Exposure Level

`web_service_private_key`	Specify the RSA Private Key for TLS that matches the public certificate.
	
	The Private Key must be PEM encoded in PKCS#12 format, without a password.
	
	For example:
	
	`web_service_private_key: /var/lib/illumio-pce/cert/rsa_private_key.key`
	
	Altneratively, you may specify a script (using $ notation) that outputs the private key. This is useful if you need to store the key in a Hardware Security Module (HSM) or other key store.
	
	For example:
	
	`web_service_private_key: $ /var/lib/illumio-pce/cert/get_rsa_private_key.sh`
	
	Note that this script can be located anywhere on the file system, as long as it is executable by the `ilo-pce user`.
	
	Example script output:
	
	`$ /local/scripts/get_rsa_private_key.sh`
	`-----BEGIN RSA PRIVATE KEY-----`
	`MIIE...`
	`many lines trimmed here`
	`-----END RSA PRIVATE KEY-----`
		Public Stable

`email_address`	Email sender address to be used by the PCE when sending emails from the system. For example, to send invitations and notifications. For example: `email_address: noreply @exampleblocked_traffic.com`	Public Stable
`service_discovery_fqdn`	The FQDN or IP address of the first core node.	Public Experimental

Runtime Environment File Parameter	Description	Exposure Level
`service_discovery_encryption_key`	Key used to encrypt Service Discovery node traffic. This value must be the same for all PCE nodes. This key also must be 16 bytes that are base64 encoded. For example: `service_discovery_encryption_key: 6h09ACGeLksZXkG5OtkcDw==`	Public Stable
`node_type`	The type of the PCE software node. Allowable values: • `core` • `data0` • `data1` For example: `node_type: core`	Public Stable
`login_banner`	Custom message on the PCE login screen, typically used to display legal notice or company policy when a user logs in.	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 may provide additional parameters to configure certain advanced functions.

Runtime Environment File Parameter	Description	Exposure Level
`ven_repo_url`	The base URL used to fetch the VENs and to enable Workload pairing with the PCE. This value must be in the form `https://host[:port]/repo_dir` Alternate ports can be used by specifying the port at the end of hostname and `repo_dir` cannot be empty. For example: `https://repo.example.com:8443/onpremgCBURz8Y4zkGk1u7N9ialjPGlZ` **Default Value**: None.	Public Stable

`ven_repo_ips`	IP addresses of the VEN repository.
	
	These IP addresses are injected into iptables to allow outbound access to the `yum/apt get` repos without having to write an explicit PCE policy.
	
	Setting this parameter also allows outbound access on 80 and 443 to these IP addresses. You can specify both single IP addresses or IP addresses with CIDR notation.
	
	If this parameter is not specified, the VEN will not be allowed to access the repository containing VEN software packages.
	
	For example:
	
	`ven_repo_ips:`
	`- 1.2.3.4`
	`- 5.6.7.8/8`
	
	Default Value: None.
		Public Stable

Runtime Environment File Parameter	Description	Exposure Level
`cluster_type`	PCE cluster type. One of these two types: • `4node_v0`: 2x2 PCE Cluster • `6node_v0`: 4x2 PCE Cluster **Default Value**: `4node_v0`.	Public Stable
`front_end_https_port`	The front end HTTPS port. If the cluster is front-ended by a SLB such as F5, then the SLB must be configured to forward this port. For example: `front_end_https_port: 8443` **Default Value**: See also `front_end_management_https_port`. If neither `front_end_management_https_port` nor `front_end_https_port` have been set, the default is TCP 8443.	Public Stable

Runtime Environment File Parameter	Description	Exposure Level
`front_end_event_service_port`	The front end Event Service port. If not specified, then port 8444 is used. If the cluster is front-ended by a SLB such as F5, then the SLB must be configured to forward this port. The idle connection timeout on the SLB may also need to be configured to maintain the connections on this port. Please consult with your Illumio Professional Services representative for additional information on configuring your load balancer. For example: `front_end_event_service_port: 8444` **Default Value**: `8444`	Public Stable
`front_end_management_https_port`	The port for PCE Web Console and REST API. The purpose of this key is to separate different kinds of communication. See also `front_end_https_port`. If neither `front_end_management_https_port` nor `front_end_https_port` have been set, the default is TCP 8443.	Public Stable
`syslog_event_export_format`	Allows you to specify VEN flow summaries and Organization events to the following event formats for export: CEF, LEEF, or JSON. **Note**: If you specify CEF or LEEF, you will continue getting traffic flows and Organization events in JSON.	Public Stable

Runtime Environment File Parameter	Description	Exposure Level
`trusted_ca_bundle`	Path to the Trusted Root Certificate bundle. This parameter is used by the PCE to validate that the certificates are trusted and indicates the path to the trusted root certificate bundle file. For example: `trusted_ca_bundle: /etc/ssl/certs/ca-bundle.crt` **Default Value**: `/etc/ssl/certs/ca-bundle.crt`	Public Stable
`email_display_name`	Email display name to be used when sending email from the system. For example, to send invitations and notifications from the PCE. For example: `email_display_name:'noreply'` **Default Value**: `noreply`	Public Stable

Runtime Environment File Parameter	Description	Exposure Level

`smtp_relay_address`	SMTP relay information used by the PCE to send email; for example, to send invitations and notifications.
	
	It is assumed that an SMTP Relay runs on localhost and listens on `127.0.0.1/587`.
	
	If that is not true then the configuration needs to be specified. This value only needs to be specified on the Core nodes.
	
	The form used is either:
	
	`ip_address (e.g. 127.0.0.1)`
	
	Or
	
	`ip_address:port (e.g. 127.0.0.1:587)`
	
	Note: If no port is specified, then port 587 is used.
	
	For example:
	
	`smtp_relay_address: 127.0.0.1:587`
	
	Default Value: `127.0.0.1:587`
		Public Stable

`export_flow_summaries_to_fluentd`	Used to specify which types of traffic flow summaries you want to export to Fluentd: allowed ('`accepted`'), potentially blocked, and blocked.
	
	For example:
	
	`export_flow_summaries_to_fluentd: - accepted`
	`- potentially_blocked`
	`- blocked`
		Public Experimental

Runtime Environment File Parameter	Description	Exposure Level

`export_flow_summaries_to_syslog`	Used to enable traffic flow summaries to syslog. You can export blocked,
	potentially blocked, and/or allowed (`accepted`)
	
	For example:
	
	`export_flow_summaries_to_syslog:`
	`- accepted`
	`- potentially_blocked`
	`- blocked`
	If you only wanted to export blocked traffic summaries, then you
	would only include the flow summary type when defining the parameter.
	
	For example:
	`export_flow_summaries_to_syslog:`
	`- blocked`
		Public
		Experimental

`syslog_event_export_format`	Used to indicate the output format for both audit events and traffic summaries to syslog, either JSON, CEF, or LEEF. For example, if you only wanted to export events to the CEF format, you would configure this parameter as follows: `syslog_event_export_format: cef` If you leave this parameter undefined, the PCE will only export events to JSON.	Public Stable

FIPS Compliance for PCE and VEN

This section details the operational requirements for compliance with Federal Information Processing Standard (FIPS) 140-2 for both the Illumio Adaptive Security Platform Policy Computer Engine (PCE) and the Linux and Windows Virtual Enforcement Node (VEN).

This release of the Illumio Adaptive Security Platform supports FIPS compliance for the Policy Compute Engine (PCE) and Virtual Enforcement Node (VEN) on Linux and Windows.

FIPS compliance is not supported for the PCE Virtual Appliance, the VEN for AIX, and the VEN for Solaris.

FIPS-related U.S. Goverment and Third-Party 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
- Windows Server 2012 NIST Security Policy
- Windows Server 2016 NIST Security Policy

Non-Government Customers with No FIPS Requirement

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

Illumio recommends that non-government customers who do not have requirement for FIPS 140-2 *not* configure and deploy the Illumio Adaptive Security Platform to support FIPS compliance.

Compliance Affirmation Letters

Third-party FIPS-compliance affirmation letters for Illumio Adaptive Security Platform are available on Illumio's Federal Solutions page.

Prerequisites for PCE FIPS Compliance

1. PCE server hardware requires the Intel Ivy Bridge CPU (2012) or later.
2. RedHat v7.4 required.
3. 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.

Prerequisites for Linux VEN FIPS Compliance

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

- For RedHat 7.1, Section 9.1 ("Cryptographic Officer Guidance") of the RHEL v7.1 Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v4.0.
- For RedHat 7.4, Section 9.1 ("Cryptographic Officer Guidance") of the RHEL v7.4 Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v5.0.

For all Linux versions of the VEN, there are no other special OS requirements or additional configurations required 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

Steps to Enable FIPS Compliance for the PCE

To enable FIPS compliance on the PCE:

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 ASP RPM as detailed in this guide.
5. During PCE installation, provide the PCE with SSL certificates that have a minimum RSA key size of 2048.

After completing the remainder of the PCE set up, the PCE is FIPS compliant.

FIPS Compliance for Linux Workloads

For all Illumio supported Linux Workloads, the standard 18.1 GA VEN release (and all later releases) support VEN Linux FIPS compliance. Starting with the VEN Linux 18.1 release, all VEN OpenSSL communications by default operate in a FIPS compliant mode.

To claim FIPS compliance for the VEN SecureConnect feature (IPSec encryption between workloads), the VEN must be installed on either RHEL v7.1 or RHEL v7.4 and configured to operate in FIPS mode as documented in Section 9.1 ("Cryptographic Officer Guidance") of the RHEL v7.1 Red Hat Enterprise Linux Kernel Crypto API Cryptographic Module v4.0 or in 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 Workloads

For Windows Workloads, the standard 18.1 GA VEN release (and all later releases) support VEN Windows FIPS compliance. The Windows VEN is FIPS-compliant when installed on either 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.

Alternative to PCE RPM Installation – Install the PCE Tarball

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

If you are installing the PCE tarball distribution, do the following tasks on each of the nodes 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 enables you to choose the directory structure that best meets your needs.

Directory	Use	Permissions	Example
`install_root`	PCE binaries and scripts.	Read / Execute	`/opt/illumio-pce`
`persistent_data_root`	A writable location where the PCE writes its persistent data. Must be owned by the user that runs the PCE.	Read / Write	`/var/lib/illumio-pce/data`
`runtime_data_root`	A writable location where the PCE writes runtime data. Must be owned by the user that runs the PCE.	Read / Write	`/var/lib/illumio-pce/runtime`
`ephemeral_data_root`	A writable location where the PCE writes temporary files.	Read / Write	`/var/lib/illumio-pce/tmp`

Directory	Use	Permissions	Example
`log_dir`	The PCE writes text file logs to this directory. You must configure logrotate (or similar) to ensure log files do not grow too large.	Read / Write	`/var/log/illumio-pce`

The table below 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. 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

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

Upgrade 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 **each** node in the cluster, do 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 Installation to RPM Installation

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

1. **As the previous PCE runtime user**, stop the PCE on each node

```
# 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 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. Either 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. As the new PCE runtime user, start the PCE on each node

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

9. **As the new PCE runtime user**, migrate the database on the data0 node.

```
# 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 5Revision History
```

Uninstall PCE

If you want to upgrade the PCE, rather than remove it, see "PCE Upgrade/Downgrade".

To uninstall the PCE:

1. Run the RPM command to remove the PCE package:

   `rpm -e pcePackageName.rpm`

2. Delete the data directory `/var/lib/illumio-pce` (default) or the custom data directory path you might have specified at installation.

The RPM uninstall does not remove the PCE `runtime_env.yml` file or SSL/TLS certificates, which can be reused on a fresh installation.

Revision History

Illumio Adaptive Security Platform PCE Deployment Guide

Document ID: 20000-100-18.2.1

Date	Description
2019-01-23	Updated for Illumio Adaptive Security Platform version 18.2.1: • New recommended storage capacity for deployment: two-storage-device configuration, with a separate device for large amounts of network traffic data. • New storage device layout/partitioning recommendations. • PCE internal syslog. • Modified API and Objects included in PCE system inventory report. • Optional – Setting Path to Custom TLS Certificate Bundle in runtime_env.yml.
2018-10-23	Clarified in "Changing Default TLS version" that restriction of TLS version to TLS 1.0 applies only to VEN for SUSE.
2018-10-17	For system limits, added `20-nproc.conf` file for RHEL7, along with previously documented `90-nproc.conf` for RHEL6.
2018-09-13	Added details about optionally configuring a SAML IdP.
2018-09-06	• Updated for Illumio Adaptive Security Platform version 18.2. • Added "Optionally validate your certificate". • "Upgrade the PCE" moved from *PCE Operations Guide* to *PCE Deployment Guide* (this guide).
2018-06-11	PKI certificates are no longer required to download the PCE.

Date	Description
2018-06-08	Include details on how to optionally validate your TLS/SSL certificate.
2018-06-01	• Include details on upgrade paths and planning tool. • Miscellaneous minor corrections/clarifications.
2018-05-10	Updated for Illumio Adaptive Security Platform version 18.1: • Removal of section numbering. • Start of revision history.