Deploying Cisco Smart PHY

This chapter provides information about deploying the Cisco Smart PHY software product package in an offline environment (without Internet connectivity).

Offline Deployment Overview

Feature History

Feature Name	Release information	Description
Support for SMI path based URL routing in the Deployer VM.	Cisco Smart PHY 22.3	Deployer VM supports path based URL. Smart PHY only needs two DNS entries as a prerequisite for installation. • cluster.example.com (DNS entry for Cluster) • deployer.example.com (DNS entry for Deployer)

Cisco Smart PHY supports deployment in an offline operator-managed vSphere virtualization environment.

You can download the Smart PHY software package as a compressed file from the Cisco.com website. The software package contains instructions, sample cluster configuration files, a cluster deployment tool, and the Smart PHY software.

Deployment Components

- Deploy tool—A deployment automation tool that controls the deployment of an Smart PHY cluster.
- Staging Environment, on page 3—A desktop operating system or virtual machine that meets the requirements to run the deploy tool.

- Cluster Configuration, on page 4 file—An YAML-formatted text file that contains the Smart PHY cluster configuration, including the vSphere configuration, and “Deployer” VM configuration. An 'admin' user creates the configuration file.
- Deployer VM—The deploy tool instantiates the Deployer VM. This virtual machine hosts the software, container image, and VM image repositories needed to complete an offline Smart PHY cluster deployment.

Deployment Overview

At a high level, deploying the Cisco Smart PHY cluster consists of the following steps:

1. (Optional) Configuring UCS Servers for Hosting Smart PHY. This step is only required when deploying on to Cisco UCS servers dedicated to the cluster.
2. Preparing the Staging Environment
3. Creating the cluster Configuration File
4. Executing the deploy tool. For more information, see Deploying the Deployer VM and Cisco Smart PHY Cluster, on page 15. To deploy another cluster, repeat creating the configuration file and deploying the cluster procedures.

The deploy tool, which is run from your staging environment, reads the Smart PHY cluster configuration from the specified cluster configuration file. After validating the values from the cluster configuration file, the deploy tool instantiates a “Deployer” VM in your designated vSphere environment.

Once the “Deployer” VM boots completely, the deploy tool syncs the Smart PHY cluster configuration to a software agent running on the “Deployer” VM. The agent executes the following Smart PHY cluster operations:

- Copying cluster VM images to the vSphere datastore
- Instantiating cluster VM
- Configuring Guest OS
- Installing and configuring container orchestrations software
- Finally, launching Smart PHY’s containerized micro-services

Deployment Types

The deploy tool can create two types of Cisco Smart PHY clusters:

- All-in-one (AIO) cluster—Runs as a single VM on an ESXi host.
  - AIO clusters are best suited for labs and small production environments where high availability is not required.
- Multinode cluster—Consists of 12 VMs deployed across three ESXi hosts.
  - Each ESXi host runs one instance of these four VMs: control-plane, etcd, Infra, and Operations.
  - Multinode clusters provide high availability and continues to operate even after a failure of one ESXi host.
  - Two multimode cluster sizes are available:

- Small—Best suited for labs and small production environment. This is the default deployment size when no value is specified in the cluster configuration file.
- Normal—Best suited for large production environment.

Prerequisites for Deployment

This section provides details on prerequisites that must be met before deploying a Cisco Smart PHY cluster.

The following resources are required to deploy, operate, and manage the Cisco Smart PHY cluster.

Related Topics

Staging Environment, on page 3
Domain Name System, on page 3
Cluster Configuration, on page 4
VMware vSphere, on page 6
VMware ESXi Host Running the Deployer VM, on page 6
VMware ESXi Hosts Running Smart PHY Cluster VMs, on page 6
Connectivity, on page 8
NTP Server, on page 8

Staging Environment

The staging environment is any Operating System or virtual machine with:

- High-speed, low latency connectivity to the vSphere environment
- At least 50GB of free disk

Prerequisites for Staging Environment

The following software must be installed:

- UNIX compatible shell
- Docker 18.09.7 or later
- Python 3.6 or later

Domain Name System

You can assign a Fully Qualified Domain Names (FQDN) to both the Smart PHY cluster and Deployer VM so that these resources function properly.

Two types of FQDN are available:

1. User-Specified FQDNs (Recommended)
2. Autogenerate FQDNs

User-Specified FQDN

User-Specified FQDN enables you to specify the hostname for both the Smart PHY cluster and deployer VM. You can specify your preferred cluster and deployer VM FQDNs using the “ingress-hostname” key-value pair (optional parameter) in the cluster configuration file.

Prerequisites for User-Specified FQDN

- Supports only alphanumeric characters in the FDQN.
- Unique FQDNs must be assigned to the cluster and deployer VM.

Ensure that you configure the corresponding DNS records (listed below) correctly in your authoritative DNS server before conducting a deployment with your specified FQDNs. If the records are not resolved correctly, then your cluster deployment fails.

Required DNS Records by entity:

```
- Operations Hub cluster:
    <cluster-fqdn>
- Deployer VM:
    <deployer-vm-fqdn>
```

For example, if the `ingress-hostname` value in the **clusters** section of your configuration file is `opshub.example.com`, then the required DNS records is `opshub.example.com`.

Autogenerate FQDN

You can trigger autogeneration of an FQDN by omitting the optional parameter “ingress-hostname” key-value pair from the relevant sections of the cluster configuration file.

The deploy tool autogenerates FQDNs by combining the entities’ management IP address, specified in the cluster configuration file, with the “nip.io” domain name.

For example, if 10.0.22.22 is assigned to the cluster management VIP and the `ingress-hostname` key-value pair is omitted, the autogenerated cluster FQDN will be `10.0.22.22.nip.io`.

Prerequisites for Autogenerate FQDN

- Ensure that your DNS servers resolve the nip.io domain properly. If nip.io resolution is blocked, then your cluster deployment fails.

Cluster Configuration

You need the following information to prepare the cluster configuration file.

vSphere Environment

Collect or prepare the following information:

- vCenter server hostname or IPv4 address
- vCenter credentials (username and password)
- vCenter Datacenter
- vCenter cluster name
- vCenter networks

- Datastore name
- Datastore folder path (optional)
- ESXi hostnames
- DNS hostname or IPv4 addresses
- Search domains
- NTP hostnames or IPv4 addresses
- HTTPS Proxy Server IPv4 address (if required)
- Environment name (This name is referenced in the cluster configuration file)

Deployer VM

Collect or prepare the following information:

- Guest OS management network:
  - One IPv4 Address
  - Subnet mask in CIDR notation
  - Gateway address
- Ingress Hostname (Optional, but recommended. For more information, see Domain Name System, on page 3.
- Username (you'll need to choose a username)
- Deployer name (This name is referenced in the cluster configuration file)

Smart PHY Cluster

Collect or prepare the following information:

- Cluster & Guest OS Management network:
  - Cluster
    - Virtual Router Redundancy Protocol (VRRP) ID
    - One IPv4 Virtual IP Address (VIP)
    - Subnet mask in CIDR notation
    - Gateway address
  - Guest OS (Must be in the same IPv4 subnet as the cluster management network):
    - AIO: One IPv4 Address
    - Multinode: 12 IPv4 Addresses. (1 address for the Guest OS on each of the 12 cluster VMs)
- Ingress Hostname (Optional, but recommended. For more information, see Domain Name System, on page 3.

- Deployment size (small or normal)
- Username (you must choose an username)

To prepare the cluster configuration file, see Cluster Configuration File, on page 10.

VMware vSphere

VMware vSphere is the only virtualization environment where the Smart PHY clusters are supported.

Supported Hypervisors

- VMware ESXi 7.0

Supported ESXi Host Management

- VMware vCenter Server 7.0

If VMware ESXi 7.0 is installed on the host, then ensure that the VMware vCenter Server version is also 7.0.

Note We recommend that you use VMware vCenter Server 7.0 with VMFS 6 Datastore type.

VMware ESXi Host Running the Deployer VM

One ESXi host is required to run the "Deployer" VM. The "Deployer" VM must not be co-located on the ESXi hosts running cluster VMs.

Prerequisites for VMware ESXi Host Running the Deployer VM

Ensure that the VMware ESXi host has the recommended capacity for compute, storage, and networking which are listed in the following table:

Table 1: Minimum System Requirements - ESXi Hosts

Parameter	Value
Processor	8 vCPUs
Memory	16 GB
Storage	320 GB Minimum 50,000 IOPS (Input/output operations per second) Latency of < 5 ms
NIC	10G vNIC

VMware ESXi Hosts Running Smart PHY Cluster VMs

Three ESXi hosts are required to run a Cisco Smart PHY multinode cluster. Cluster VMs must not be co-located on the ESXi host running the "Deployer" VM.

Prerequisites for VMware ESXi Hosts Running Smart PHY Cluster VMs

Ensure that the VMware ESXi host has the recommended capacity for compute, storage, and networking which are listed in the following table:

Table 2: Minimum System Requirements - ESXi Hosts

Cluster Size	Small	Normal
Processor	20 vCPUs	34 vCPUs
Memory	160 GB	304 GB
Storage	1640 GB Minimum 50,000 IOPS (Input/output operations per second) Latency of < 5 ms	2440 GB Minimum 50,000 IOPS (Input/output operations per second) Latency of < 5 ms
NIC	2x 10G vNIC	2x 10G vNIC

The following tables show the minimum requirements for AIO cluster:

AIO

VM Type	CPU Cores	RAM Size (GB)	SSD Storage Size (GB)
AIO	18	96	1541

The following tables show the minimum requirements for each of the VM types deployed in a multimode cluster:

Small Multimode Cluster

VM Type	CPU Cores	RAM Size (GB)	SSD Storage Size (GB)
Control Plane	2	16	125
etcd	2	16	125
infra	8	64	1000
ops	8	64	320

Normal Multimode Cluster

VM Type	CPU Cores	RAM Size (GB)	SSD Storage Size (GB)
Control Plane	2	16	125
etcd	2	16	125
infra	14	96	1500
ops	16	176	620

Connectivity

From the Staging environment, the deploy tool must have connectivity to the following resources:

- Local DNS server
- vCenter server
- IPv4 subnet assigned to the Guest OS management network on the “Deployer” VM
- IPv4 subnet assigned to the Cluster VIP and Guest OS management network on the cluster VMs

The “Deployer” VM, once created, must have connectivity to the following resources:

- Local DNS server
- vCenter server
- IPv4 subnet assigned to the Cluster VIP and Guest OS management network on the cluster VMs

NTP Server

Ensure that the clocks for the staging server, Deployer VM, and Cluster VM are in sync, preferably pointing to the same NTP server.

Preparing the Staging Environment

This section provides details on how to prepare your staging environment for a Smart PHY cluster deployment.

Preparing the staging environment involves the execution of the following procedures

- Transferring the Smart PHY release package to your Staging Environment, on page 8
- Extracting Installation files and Performinng Signature Verification, on page 8

Transferring the Smart PHY release package to your Staging Environment

Step 1 Download the Smart PHY release package (`smartphy-installer-<version>.SPA.tgz`) from Cisco.com. The package is approximately 15 GB.

Step 2 Securely copy the release package to your Staging Environment.

Extracting Installation files and Performinng Signature Verification

The commands listed in this procedure should be executed from the shell in your Staging Environment.

Step 1 Run the `tar` command against the `smartphy-installer-<version>.SPA.tgz` release package to extract the installation files.

Example:

```
tar -zxovf smartphy-installer-<version>.SSA.tgz
```

The following files are extracted:

- `smartphy-installer-<version>.tgz`
- `smartphy-installer-<version>.tgz.signature`
- `cs-verify.sh`
- `SMART_PHY_REL_KEY-CCO_RELEASE.cer`
- `signed_files`

Step 2 Run the `cs-verify.sh` script to verify the signature of the `smartphy-installer-<version>.tgz` installer image.

Example:

```
./cs-verify.sh SMART_PHY_REL_KEY-CCO_RELEASE.cer smartphy-installer-<version>.tgz
```

Example output:

```
Verifying signature

Signature verification succeeded
```

If the signature verification fail, then delete the previously extracted installation files and the installer image. Re-download the Smart PHY release package from Cisco.com and start this procedure again.

Step 3 Run the `tar` command against the extracted `smartphy-installer-<version>.tgz` installer image to create the staging directory.

Example:

```
tar -zxovf smartphy-installer-<version>.tgz
```

A new directory named `smartphy-installer-<version>` has been created. This directory is known as the staging directory.

Step 4 Navigate to the `smartphy-installer-<version>` staging directory.

Example:

```
cd smartphy-installer-<version>
```

The staging directory `smartphy-installer-<version>` contains the following files and folders:

:

```
smartphy-installer-<version>
├── README.md
├── cluster-deployer-<version>.tar
├── cluster-deployer-<version>.tar.signature
├── deploy
├── deploy.signature
├── docker-images
│   ├── ccmts-customization_<version>.tar
│   └── ccmts-customization_<version>.tar.signature
├── examples
│   ├── aio-smartphy-config.yaml
│   ├── aio-smartphy-standby-config.yaml
│   ├── deployer-sample-config.yaml
│   ├── multinode-smartphy-config.yaml
│   └── multinode-smartphy-standby-config.yaml
├── offline-products
│   ├── cee-<versioin>.tar
│   ├── cee-<versioin>.tar.signature
│   ├── opshub.tar
│   ├── opshub.tar.signature
│   ├── smartphy-<version>.tar.signature
│   └── smartphy-<version>.tar
├── smi-install-disk.iso
├── smi-install-disk.iso.signature
├── upgrade-prep
├── upgrade-prep.signature
└── utility-images
    ├── autodeploy_<version>.tar
    ├── autodeploy_<version>.tar.signature
    ├── cluster-manager-docker-deployer_<version>.tar
    └── cluster-manager-docker-deployer_<version>.tar.signature
```

Preparing Cluster Configuration File

This section provides info on how to create the Cluster Configuration file required by Smart PHY's deploy tool. This configuration file stores all of the information the deploy tool needs in order to deploy your Smart PHY cluster.

Sample Configuration Files

During the extraction of the Smart PHY release package an `example` directory is created under the staging directory. The `example` directory contains the following sample configuration files:

- `deploy-sample-config.yaml`—A configuration file with a deployer, but no cluster.
- `aio-smartphy-config.yaml`—A configuration file with a deployer and a single-node Smart PHY cluster.
- `multinode-smartphy-config.yaml`—A configuration file a deployer and multinode Smart PHY cluster.
- `aio-smartphy-standby-config.yaml`—A configuration file with a deployer and a single-node Smart PHY cluster that is configured for standby (without CIN configuration).
- `multinode-smartphy-standby-config.yaml`—A configuration file with a deployer and a multinode Smart PHY cluster configured standby (without CIN configuration).

Cluster Configuration File

Place the configuration file in the staging directory. This configuration file is in the standard YAML language format, with the following three sections:

- Environments
- Deployers
- Clusters (Smart PHY multi-node/single-node)

Each section can contain multiple items. Replace <...> with actual values.

Environment Configuration

The `environments` section defines a vSphere deployment domain. This environment is referenced in the `deployers` and `clusters` sections, which you define shortly.

```
environments:
    <env-name>:                           # Environment name
        server: <value>                   # vCenter Server IPv4 Address or name
        username: <value>                # vCenter username, user will be prompted for the
 password
        datacenter: <valuer>              # vCenter Datacenter Name
        cluster: <value>                  # vCenter Cluster Name
        nics: [ <list> ]                  # List of vCenter nics (port groups)
        nameservers: [ <value> ]          # List of DNS Server IPv4 Addresses
        search-domains: [ <value> ]       # List of Search domains
        ntp: [ <list> ]                   # List od NTP Server IPv4 Addresses or name
        https-proxy: <value>              # Optional HTTPS Proxy (Ex:
http://proxyhost.domain.tld:port)
        no-proxy: <value>                 # Optional HTTPS Proxy bypass
```

Guidelines for Defining an Environment

- The environment name can have only lowercase letters, digits, and hyphens (-).
- The `nics` list must contain only one network, although the `nics` configuration allows multiple networks. This network is used as the management network by the deployer or cluster that refers to this environment.
- Create multiple `environments` if your vCenter has more than one network that serves as a management network. Create one `environments` for each network. In addition, refer to the corresponding `environments` in the deployer or cluster based on the management network it uses.
- Make sure the `nics`, `nameservers`, and `search-domains` values are structured as YAML lists.

Deployer Configuration

The `deployers` section defines the configuration of a Deployer VM.

```
deployers:
   <deployer-vm-name>:                 # Deployer VM name
       environment: <value>            # Environment name defined in the 'Environments'
section.
       address: <value>                # Deployer VM IPv4 Address in CIDR format
       gateway: <value>                # Deployer VM's Gateway IP Address
       ingress-hostname: <value>       # Optional FQDN (Ex. "deployer1.example.com")
       username: <value>              # Deployer VM username, you will be prompted for the
 password
       private-key-file: <value>       # Optional SSH private-key-file (.pem) with path
relative to the staging
                                       #   directory. Key will be auto-generated, if one
if not provided.
       host: <value>                   # IPv4 Address of the ESXi Host
       datastore: <value>              # Datastore for the Deployer VM
       datastore-folder: <value>       # Optional Datastore folder path. When omitted
deployer and
                                       #   cluster VMs are created under root of the
Datastore.
```

```
                                #
# 'docker-subnet-override' and its values are optional. It should only be
#  used when you need to customize the deployer VM's Docker IP Addressing.
#  When omitted, the Docker bridge defaults to 172.17.0.0/16.
                                #
docker-subnet-override:         #
- pool-name: <value>            # Docker bridge address pool name
  base: <value>                 # Docker bridge subnet in CIDR format
  size: <value>                 # Docker bridge subnet size: 8-24
```

Guidelines for Defining a Deployer

- The `deployer-vm-name` can have only lowercase letters, digits, and hyphens (-).
- The `private-key-file`, when present, must refer to the SSH private key file (.pem). This file must be in the staging directory and must not be accessible (read/write/execute) to other users.
- If the `private-key-file` line is omitted, the deploy tool generates an SSH private key for the deployer and places it in the .sec subdirectory under the staging directory. The filename is `<deployer-vm-name>_auto.pem`.
- The values associated with `docker-subnet-override` are optional. Those values should only be included in the configuration when you need to customize the deployer VM's Docker IP addressing. When omitted, the Docker bridge on the Deployer VM defaults to 172.17.0.0/16.
- To avoid resource contention, do not host the deployer VM on the same ESXi hosts running any of the cluster VMs.
- When you configure a custom `ingress-hostname`, ensure that the following entries are in the DNS:

```
<host.domain.tld>
charts.<host.domain.tld>
files-offline.smi-cluster-deployer.<host.domain.tld>
deployer-ui.smi-cluster-deployer.<host.domain.tld>
cli.smi-cluster-deployer.<host.domain.tld>
restconf.smi-cluster-deployer.<host.domain.tld>
docker.<host.domain.tld>
```

Cluster Configuration

The `clusters` section defines the type and configuration of the `cluster`. At least one `environment` and one `deployer` must be defined in the cluster configuration file in order for a `cluster` to be deployed.

`Clusters` can be deployed on a single ESXi host or across three ESXi hosts. The single host deployment is known as a single-node deployment, or an All-in-one (AIO), while the three-node deployment is known as a multi-node deployment.

The `clusters` configuration below shows the mandatory and optional key-value pairs required for a multi-node deployment.

```
clusters:
   <cluster-name>:                                       # Cluster name
       type: <value>                                     # Cluster type must be 'opshub'
       size: <value>                                    # Optional cluster size: 'small'
 or 'normal'. Defaults to small when not specified.
       environment: <value>                              # Environment name defined in
the 'Environments' section.
       gateway: <value>                                  # Cluster Gateway IPv4 Address
       ingress-hostname: <value>                           # Optional FQDN (Ex.
“smartphy.example.com”)
       username: <value>                                   # Cluster username, User will
be prompted for the cluster password
       private-key-file: <value>                           # SSH private-key-file (.pem)
including path relative to the staging directory
                                                           #  Key will be auto-generated,
if not provided.
                                                           #
       primary-vip: <value>                               # Management Virtual IPv4 Address
 in CIDR format
       vrouter-id:  <value>                                # Management Keepalived Virtual
 Router ID, value must be between 0-255
                                                           #
       enable-http-redirect: value                         # Optional. Defaults to false
when not specified. Set to “true” to redirect HTTP requests to HTTPS.
       # The next three key-value pairs are optional. They should only be used
       # when you need to customize the cluster's internal IP Addressing:
                                                            #
       pod-subnet: <value>                                 # K8s Pod subnet in CIDR format.
 If omitted, defaults to: 192.168.0.0/16
       service-subnet: <value>                              # K8s Service subnet in CIDR
format. If omitted, defaults to 10.96.0.0/12
       docker-bridge-subnet: [ <addr-list> ]               # List of Docker bridge subnets
 in CIDR format. If omitted, defaults to 172.17.0.0/16
                                                            #
       nodes:                                               #
       -  host: <value>                                     # ESXi Host 1: IPv4 Address
          addresses: [ <addr-list> ]                        # ESXi Host 1: List of Mgmt
IPv4 addr assigned to control-plane, etcd, infra, and Ops VMs respectively
          datastore: <value>                               # ESXi Host 1: IPv4 Address for
 vCenter Datastore
       -  host: <value>                                     # ESXi Host 2: IPv4 Address
          addresses: [ <addr-list> ]                        # ESXi Host 2: List of Mgmt
IPv4 addr assigned to control-plane, etcd, infra, and Ops VMs respectively
          datastore: <value>                               # ESXi Host 2: IPv4 Address for
 vCenter Datastore
       -  host: <value>                                     # ESXi Host 3: IPv4 Address
          addresses: [ <addr-list> ]                        # ESXi Host 3: List of Mgmt
IPv4 addr assigned to control-plane, etcd, infra, and Ops VMs respectively
          datastore: <value>                               # ESXi Host 3: IPv4 Address for
 vCenter Datastore
                                                            #
       apps:                                                #
       - smartphy:                                          # All of the parameters below
are Smart PHY specific
         nodes:                                             #
           - host: <value>                                  # ESXi Host 1: IPv4 Address
(Same address as used earlier in the nodes section.)
             nics: [ <list> ]                               # ESXi Host 1: vCenter list of
 Network for CIN
             ops:                                           # -- The following parameters
apply to Ops VM 1. --
               interfaces:                                  # Ops VM 1: CIN Interface
configuration:
                 - addresses: [ <addr-list> ]               #   Ops VM 1: List of CIN IPv4
 or v6 Addresses in CIDR format
                  vip: [ <vip-list> ]                     #   Ops VM 1: List of CIN Virtual
 IPv4 or v6 Addresses in CIDR format
                   vrouter-id: <value>                      #   Ops VM 1: CIN Keepalived
Virtual Router ID, value must be between 0-255
                   routes:                                  #   Ops VM 1: Optional Route
configuration:
                     - { dest: [ <list> ], nhop: <value> }  #      Ops VM 1: Optional list
 of Destination Subnets; Next-Hop IP Address
                     - { dest: [ <list> ], nhop: <value> }  #      Ops VM 1: Optional list
 of Destination Subnets; Next-Hop IP Address
          - host: <value>                                   # ESXi Host 2: IPv4 Address
(Same address as used earlier in the nodes section.)
             nics: [ <list> ]                               # ESXi Host 2: vCenter list of
 Network for CIN
             ops:                                           # -- The following parameters
apply to Ops VM 2. --
               interfaces:                                  # Ops VM 2: CIN Interface
configuration:
                 - addresses: [ <addr-list> ]               #   Ops VM 2: List of CIN IPv4
 or v6 Addresses in CIDR format
                  vip: [ <vip-list> ]                     #   Ops VM 2: List of CIN Virtual
 IPv4 or v6 Addresses in CIDR format
                   vrouter-id: <value>                      #   Ops VM 2: CIN Keepalived
Virtual Router ID, value must be between 0-255
                   routes:                                  #   Ops VM 2: Optional Route
configuration:
                     - { dest: [ <list> ], nhop: <value> }  #     Ops VM 2: Optional list
of Destination Subnets; Next-Hop IP Address
                     - { dest: [ <list> ], nhop: <value> }  #     Ops VM 2: Optional list
of Destination Subnets; Next-Hop IP Address
       - host: <value>                                      # ESXi Host 3: IPv4 Address
(Same address as used earlier in the nodes section.)
             nics: [ <list> ]                               # ESXi Host 3: vCenter list of
 Network for CIN
             ops:                                           # -- The following parameters
apply to Ops VM 3. --
               interfaces:                                  # Ops VM 3: CIN Interface
configuration:
                 - addresses: [ <addr-list> ]               #   Ops VM 3: List of CIN IPv4
 or v6 Addresses in CIDR format
                  vip: [ <vip-list> ]                     #   Ops VM 3: List of CIN Virtual
 IPv4 or v6 Addresses in CIDR format
                   vrouter-id: <value>                      #   Ops VM 3: CIN Keepalived
Virtual Router ID, value must be between 0-255
                   routes:                                  #   Ops VM 3: Optional Route
configuration:
                     - { dest: [ <list> ], nhop: <value> }  #     Ops VM 3: Optional list
of Destination Subnets; Next-Hop IP Address
                     - { dest: [ <list> ], nhop: <value> }  #     Ops VM 3: Optional list
of Destination Subnets; Next-Hop IP Address
```

Guidelines for Defining a Cluster

- The `cluster-name` can have only lowercase letters, digits, and hyphens (-).
- When you specify a FQDN in the `ingress-hostname`, ensure that the corresponding entries are configured in your DNS servers:

  ```
  <host.domain.tld>
  opscenter.<host.domain.tld>
  ```

- If you do not specify an FQDN in the `ingress-hostname`, the cluster's `primary-vip` (also known as the Management Virtual IP address) is used to generate an FQDN leveraging nip.io as the domain and top-level domain (TLD). For example, if the `primary-vip` is 10.0.0.2, the generated FQDN is 10.0.0.2.nip.io. Your DNS servers must allow the resolution of the nip.io domain. If resolution of nip.io is blocked, you cannot access the cluster.
- The `private-key-file`, when present, must refer to the SSH private key file (.pem). This file must be in the staging directory and must not be accessible (read/write/execute) to other users.

- If the `private-key-file` line is omitted, the deploy tool generates an SSH private key for the `cluster` and places it in the .sec subdirectory under the staging directory. The filename is `<cluster-name>_auto.pem`.
- If multiple clusters share the same management subnet, the management `vrouter-id` (VRRP ID) of each cluster must be unique.

Cisco Smart PHY CIN Configuration

Configure Converged Interconnect Network (CIN) for the Cisco Smart PHY cluster. One or more CIN networks can be present. Configure CIN under each node.

Guidelines for Defining Smart PHY's CIN Interfaces

- The CIN Virtual IP addresses (`vip`) filed is mandatory. You can configure up to one IPv4 and one IPv6 addresses per CIN network.
- The CIN Virtual IP addresses (`vip`) and the VRRP ID (`vrouter-id`) fields are used only in multi-node cluster deployments. They are configured on the first node.
- If multiple Smart PHY clusters share a CIN subnet, the VRRP ID (`vrouter-id`) should be unique for each cluster.
- For multi-node clusters, all Ops VMs must have the same number of CIN interfaces. If the `nics` or `route` fields are missing for the second or third Ops VM nodes, the corresponding value from the first Ops VM node is used.
- You can setup a Smart PHY cluster as a backup cluster. To do so, do not include any CIN configuration. The configuration should not have `ops` and `interfaces` under `nodes`.

Note Cisco Smart PHY clusters can connect to multiple CIN networks using multiple network interfaces configured during deployment.

Adding CIN Configuration Without Cluster Reboot

After deploying Smart PHY, you can add new CIN Configuration without restarting the cluster by using the following steps:

1. In the Day-0 config, add additional CIN Configurations details and provide the necessary details of CIN, interface address, vip, and vrouter-id config.
2. Run cluster deployment with the `-np` argument. Example: `./deploy -c day0.yaml -np`

Once deployment is done successfully, the cluster is updated with CIN Configuration information. The cluster does not reboot.

Deploying the Deployer VM and Cisco Smart PHY Cluster

This sections explains how to use the deploy tool to deploy the Deployer VM and Cisco Smart PHY cluster.

From the staging environment, run the deploy tool to deploy the clusters using the following command:

```
$ ./deploy
Usage ./deploy -c <config_file> [-v]
  -c <config_file> : Configuration File, <Mandatory Argument>
  -v               : Config Validation Flag, [Optional]
  -f               : Day0: Force VM Redeploy Flag [Optional]
  -u               : Cluster chart Upgrade Flag [Optional]
  -s               : Skip Compare Flag [Optional]
  -sc              : Skip Compatibility check during upgrade Flag [Optional]
  -D               : Enable Debug Logs [Optional]
  -np              : Provision new NIC without cluster reboot [Optional]
```

The following options are available in the deploy tool:

- `-c <config_file>`: Configuration file (Mandatory Argument). This option is the first option in the command.
- `-u`: Cluster chart Update Flag [Optional]
- `-v`: Config Validation Flag, [Optional]
- `-f`: Redeploy the cluster. If you redeploy the cluster, cluster VM's will be rebooted and the data persisted on disk will be retained. You can use this option to modify some of the cluster parameters.

The `-u` flag is for updating CNF/charts in cluster.

The deploy tool triggers the docker command that requires root permission to run. Depending on your setting, you can use the **sudo** to the deploy command.

The deploy tool does the following operations:

- If you are running the deploy tool for the first time, it prompts you to enter all passwords required for installation.
  - For vCenter environment: vCenter password for the user specified in the environment configuration.
  - For deployer: SSH password of the user admin for the deployer's Operation Center.
  - For Cisco Smart PHY cluster: SSH password for all VMs in the cluster (or user-specified in the cluster's configuration file). Also, the SSH passwords for the three Operation Centers (Cisco Smart PHY, Operations Hub, and CEE); for user admin.

  You are prompted twice to enter each password. The password is saved inside the staging directory in encrypted form for future use.
- Passwords for the deployer, the cluster, and the Operation Centers must be eight characters long, and must have a lowercase letter, uppercase letter, a digit, and a special character.
- The deploy tool generates an SSH key pair when the `private-key-file` line is missing for the deployer or the cluster in the configuration file. The generated private key files are in the `.sec` sub directory under the staging directory, with `<cluster-name>_auto.pem` filename.
- The root user owns the generated private keys. When logging in using SSH and these private key files, make sure that you run it with `sudo`.
- If the deployer VM is not running, the deploy tool installs the deployer VM.
- The deploy tool checks if the deployer VM is missing any of the product packages that are found in the `offline-images` directory, and if it finds any missing, it uploads them to the deployer VM.

- The tool also generates the configuration for each cluster and pushes them to the deployer VM.
- The deploy tool triggers the deployer VM to perform the sync operation for the cluster. The sync operation applies the configuration to the cluster. If you have not set up the cluster, it installs the cluster. Or the sync operation updates the cluster with the configuration.
- If the sync operation times out, the deploy tool triggers the sync operation again. The tool waits for the sync operation to complete, and then continues to monitor the cluster to make sure that all helm charts are deployed and all pods are created.

You can repeat the deploy tool to deploy more than one cluster by providing the corresponding configuration files. Alternatively, you can run this command appending a `-v` flag. The `-v` flag forces the deploy tool to skip the synchronizing operation. Use this option to push the configuration of a cluster to the deployer without deploying or updating the cluster.

Wait for the installation process to complete. Following is a sample output after the process is complete:

```
Friday 22 October 2021  07:53:52 +0000 (0:00:00.123)       0:12:22.518 ********
install-cm-offline : Extract cluster manager file into /data ---------- 545.16s
vm-vsphere-iso : Wait for ssh ------------------------------------------ 88.51s
install-cm-offline : Deploy cluster manager ---------------------------- 85.14s
install-ntp-iso : force_time_sync --------------------------------------- 7.34s
vm-vsphere-iso : Create VM ---------------------------------------------- 3.85s
vm-vsphere-iso : Get VM Update needed ----------------------------------- 1.65s
install-ntp-iso : Cleaning cache ---------------------------------------- 1.53s
Gathering Facts --------------------------------------------------------- 1.34s
vm-vsphere-iso : Check if ISO file exists ------------------------------- 0.79s
vm-vsphere-iso : Test vCenter credentials are valid --------------------- 0.60s
install-ntp-iso : apt_update -------------------------------------------- 0.55s
vm-vsphere-iso : Create user data ISO ----------------------------------- 0.52s
install-ntp-iso : Remove "ntp" package ---------------------------------- 0.47s
install-cm-offline : Ensure /data/cm-install folder NOT exists ---------- 0.36s
install-ntp-iso : Install offline APT repo GPG key ---------------------- 0.34s
install-cm-offline : Ensure /data folder exists ------------------------- 0.33s
install-ntp-iso : restart_chrony ---------------------------------------- 0.28s
install-ntp-iso : enable chrony ntp ------------------------------------- 0.28s
download-iso : download base image ISO file ----------------------------- 0.28s
vm-vsphere-iso : Create netplan Template -------------------------------- 0.18s

Create deployers completed
```

Deploying the Cluster with CA signed certificate using the deploy command

When you deploy the Cisco SmartPHY cluster, the cluster is configured with a self-signed certificate by default. You can deploy the cluster with a CA signed certificate by performing the following steps before running deploy tool.

1. Generate a CA signed certificate with a common name as `ingress-hostname` used in the day 0 configuration YAML file.
2. On the stanging environment, create a directory with the cluster name as the directory name under `<staging directory>/certs/client_certificates`. For example, if you use cluster name `testcluster`, the created directory will be `<staging directory>/certs/client_certificates/testcluster`. This directory is called **cluster ingress certificates directory**.
3. Create `cert-api-ingress` and `default-ssl-certificate` directories under **cluster ingress certificates directory**.
4. Place the CA Signed certificate and keys under `cert-api-ingress` directory. The CA signed certificate file has `.crt` extension and key file has `.key` extension.

Verifying Installation

After successfully deploying the Cisco Smart PHY application using the deploy tool, the console shows a success message.

Log in to one of the control-plan nodes and make sure that all the pods are in the `Running` state.

```
kubectl get pod --all-namespaces
```

A few internal services and pods may need more time to complete the startup tasks and successfully establish communication with other services within the cluster. After a few minutes, you can initiate all operations from the Cisco Smart PHY web UI page.

Configuring Dual Stack on External Cluster Interfaces

Table 3: Feature History

Feature Name	Release Information	Description
IPv6 Dual Stack on support on external cluster interfaces	Cisco Smart PHY, Release 22.2	Support that is extended to configure IPv6 dual stack configuration parameters such as mode, gateway, subnet address, and so on, on an external cluster interface.

Prerequisites

1. Dual stack must be configured at the time of cluster creation. It cannot be added to a previously created Smart PHY cluster.
2. The ESXi hosts must be connected to dual stack enabled networks.

To configure dual stack on an external cluster interface, perform the following steps:

1. Navigate to the cluster configuration file in the staging environment.
2. Add the following parameters in the configuration file.

Table 4: IPV6 Dual Stack Parameters

Parameter	Description	Value to set
ipv6-mode (Optional)	Specifies to set whether the IPv6 mode is true or false.	• true—To enable the dual stack on the cluster interface. • false—To disable the dual stack on the cluster interface.
primary-vip-ipv6	Specifies the IPv6 address where the Nginx ingress controller binds to.	This parameter is mandatory if the ipv6-mode is set as true.

Parameter	Description	Value to set
ipv6-gateway	Specifies the IPv6 gateway address.	This parameter is mandatory if the ipv6-mode is set as true.
pod-subnet-ipv6	Specifies IPv6 subnet address.	This parameter is optional. If no value is specified, then by default the value "fd20::0/112" is assigned.
service-subnet-ipv6	Specifies the IPv6 service subnet address.	This parameter is optional. If no value is specified, then by default the value "fd20::0/112" is assigned. This parameter is valid only if the ipv6-mode is set as true.
addresses-v6	Specifies the virtual machine SSH IPv6 address.	This parameter is mandatory if the ipv6-mode is set as true.

Deployment Limitations

- Modification of cluster parameters such as server names, NTP server configuration details, data store file path, subnets, IP addresses of VMs, and so on, requires a VM restart. You can restart VM using the **deploy-f** command.
- Autodeployer only supports Application Product chart and docker image upgrades. The modification of cluster configuration is not supported as part of the upgrade process.
- When you enable dual-stack, you must redeploy the cluster.
- The Deployer VM must be saved so that you can use the VM while upgrading the cluster.
- Manual provisioning of NIC interfaces on VMs must be performed through vCenter during the SmartPHY installation on top of the running Operations Hub platform.
- Removal of exiting NICs requires a VM restart. You can restart VM using the **deploy-f** command.
- Data store folders must be created in vCenter manually before starting the Smart PHY installation.