CPS vDRA Installation Guide for VMware, Release 26.1.0

First Published: 2026-03-16

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

PREFACE

Preface v

About This Guide v

Audience v

Additional Support vi

Conventions (all documentation) vi

Communications, Services, and Additional Information vii

Important Notes viii

CHAPTER 1

Pre-Installation Requirements 1

Installation Overview 1

Sample vDRA System 2

Installation Order 2

Requirements 2

VMware Interface Name and Order 3

Environment Artifacts 4

CHAPTER 2

Installing CPS vDRA 5

Create Installer VM in vSphere 5

Upload the VDMK File 5

Convert CPS Deployer VMDK to ESXi Format 5

Create CPS Installer VM 5

Configure network 7

Binding-VNF 7

VMware disk encryption 8

Limitations and restrictions for disk encryption 9

Configure the native key provider 10

Back up the native key provider 10
Create a VM storage policy for encryption 11
Create an encryption-enabled cluster 11
Add ESXi hosts to the cluster 12
Configure the environment file for encryption 12
CPS Installer Commands 13
Upgrading VMs using Diagnostics and Redeployment Health Check 16
Ranking Details 16
Resume Redeployment 17
Validate Deployment 17
show system status 17
show system diagnostics 18
show docker engine 18
show docker service 18
Redeploy VMs during the ISSM Operation 19
Redeploy VMs during the ISSM Operation with Overlay Network 20
ISSM Initial Configuration for Overlay Network 20
ISSM Upgrade in Overlay Setup 22

APPENDIX A **Installation Examples** 27
DRA-VNF Example 27
Artifacts Structure Example 27
Top Level Directory 29
example-dra-vnf/vms/role 31
Data Disk 31

APPENDIX B **Listening Ports in DRA Deployment** 35
Listening Ports in DRA Deployment 35

Preface

- About This Guide, on page v
- Audience, on page v
- Additional Support, on page vi
- Conventions (all documentation), on page vi
- Communications, Services, and Additional Information, on page vii
- Important Notes, on page viii

About This Guide

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This document is a part of the Cisco Policy Suite documentation set.

For information about available documentation, see the *CPS Documentation Map* for this release at Cisco.com.

Note The PATS/ATS, ANDSF, and MOG products have reached end of life and are not supported in this release. Any references to these products (specific or implied), their components or functions in this document are coincidental and are not supported. Full details on the end of life for these products are available at: https://www.cisco.com/c/en/us/products/wireless/policy-suite-mobile/eos-eol-notice-listing.html.

Audience

This guide is best used by these readers:

- Network administrators

- Network engineers
- Network operators
- System administrators

This document assumes a general understanding of network architecture, configuration, and operations.

Additional Support

For further documentation and support:

- Contact your Cisco Systems, Inc. technical representative.
- Call the Cisco Systems, Inc. technical support number.
- Write to Cisco Systems, Inc. at support@cisco.com.
- Refer to support matrix at https://www.cisco.com/c/en/us/support/index.html and to other documents related to *Cisco Policy Suite*.

Conventions (all documentation)

This document uses the following conventions.

Conventions	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
courier font	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.

Conventions	Indication
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means reader take note. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means reader be careful. In this situation, you might perform an action that could result in equipment damage or loss of data.

Warning IMPORTANT SAFETY INSTRUCTIONS.

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Note Regulatory: Provided for additional information and to comply with regulatory and customer requirements.

Communications, Services, and Additional Information

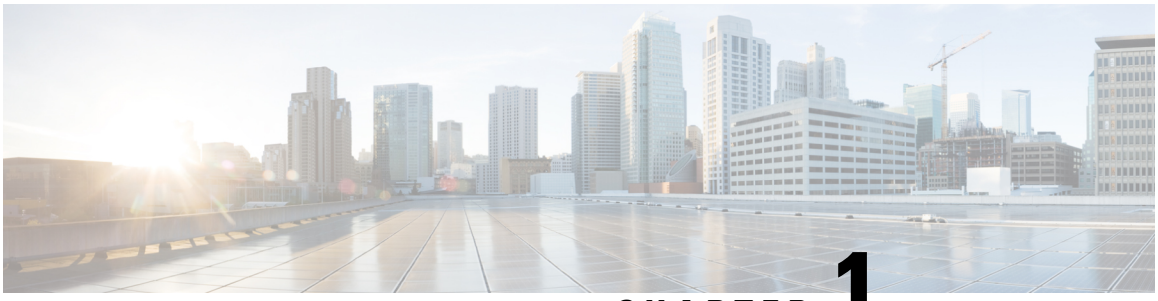
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Important Notes

Important Any feature or GUI functionality that is not documented may not be supported in this release or may be customer specific, and must not be used without consulting your Cisco Account representative.



CHAPTER 1

Pre-Installation Requirements

- Installation Overview, on page 1
- Sample vDRA System, on page 2
- Installation Order, on page 2
- Requirements, on page 2
- Environment Artifacts, on page 4

Installation Overview

The vDRA vSphere installer launches vDRA VMs as specified in the User Input structure. Once the VMs are launched, all VMs must be registered with the master as displayed using the command `show running-config docker | tab`. Also, the system percent-complete must reach 100% as displayed using the command `show system status`.

Once the VMs are registered, the installer is done and you can proceed with configuring the vDRA system.

VMware ESXi 6.7/7.0ESXi 8.0 must be installed on all the blades that are used to host the vDRA system. For more details see link.

Installing vDRA on vSphere includes the following:

- Create a vDRA installer VM in vSphere using the latest vDRA Deployer Host VMDK.
- Create the artifacts that describe the VM roles, CPS ISO (dra-vnf or binding-vnf), IP addresses, hostnames, target ESXi servers, and so on.
- Run the `cps install <vnf directory>` command.

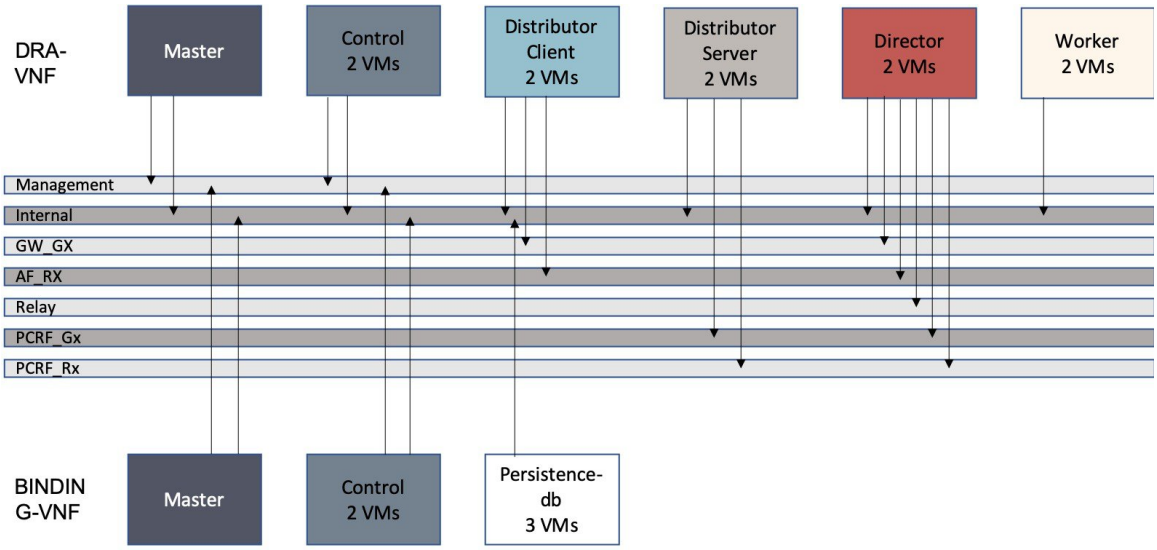
Note The ESXi servers must be configured to use the Network Time Protocol (NTP) to synchronize their clocks.

In vSphere 6.7/7.0 and later, the vSphere Web Client is installed as part of the vCenter Server on Windows or the vCenter Server Appliance deployment.

Sample vDRA System

The following network diagram, configuration and VM layout are for illustration purposes only. Contact Cisco Account representative for your specific vDRA requirements.

Figure 1: Sample vDRA System



Installation Order

The following installation order should be used:

1. Binding VNF
2. DRA VNF

Note VMs per VNF must be installed in parallel. There are no VM ordering requirements while installing a vDRA VNF.

Requirements

Note For blade requirements, contact your Cisco Account representative.

Virtual Machine (VM)

The table list the VM requirements for vDRA:

Table 1: VM Requirements

Role	vCPU	RAM (GB)	Primary Disk (GB)	Data Disk (GB)
master	16	64	100	200
control	16	64	100	200
dra-director	40	128	100	-
dra-distributor	16	32	100	-
dra-worker	16	128	100	-
persistence-db	8	64	100	-
Installer	8	32	100	-

vSphere

vSphere 6.7/7.0

ESXi Servers

- UCSB-B200-M5
- 512 GB RAM
- 2 SSD Drivers
- 2 CPUs with 28 cores each
- NTP Enabled

VMware Interface Name and Order

In VMware, the NETWORK definition from the `env` files map to the following Linux interface names:

Table 2: Network Definition Mapping to Linux Interface Name

NETWORK_	Linux Interface Name
0	ens160
1	ens192
2	ens224
3	ens256
4	ens161
5	ens193

NETWORK_	Linux Interface Name
6	ens225
7	ens257
8	ens162
9	ens194

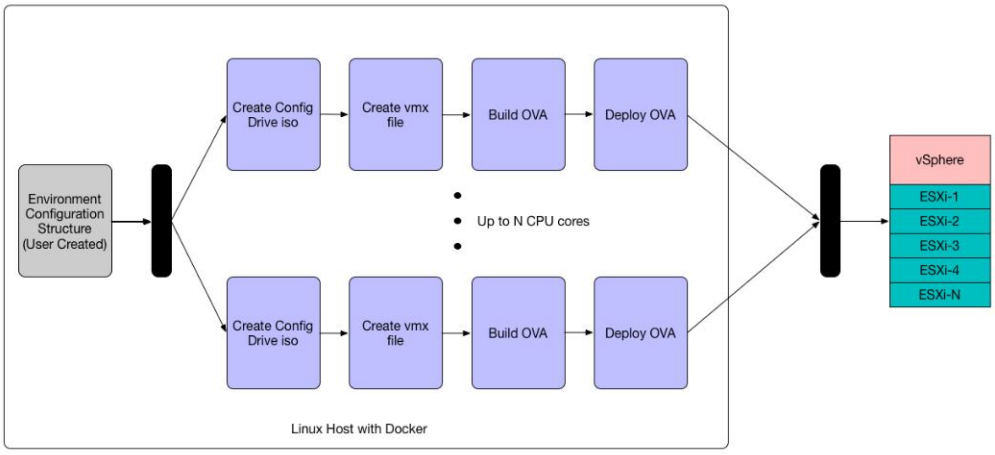
Environment Artifacts

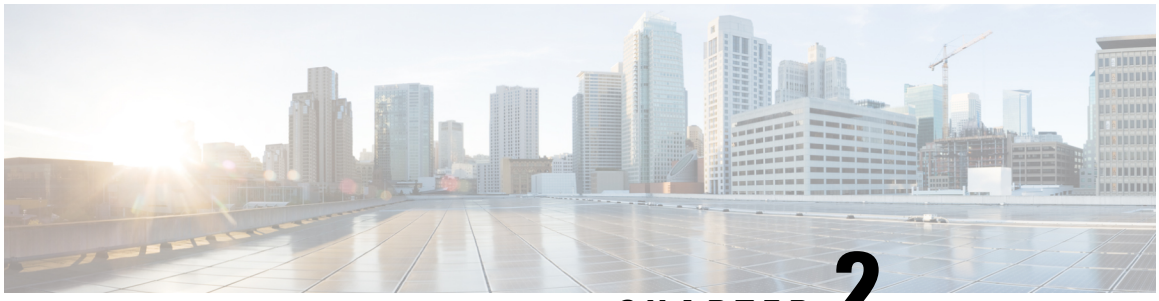
You can specify the test bed configuration settings for global, role, and VM in increasing precedence using a directory structure and files containing key-value environment variables.

The Jinja2 templates are used to create user_data files for cloud-init, ovftool options, and VMware Virtual Machine VMX configuration files. The environment variables are applied to the various Jinja2 template files using `envtpl`.

The installer loops over the directory structure sourcing global environment, role environment, and finally VM environment settings. Once at the VM level, the installer applies the environment variables to the Jinja2 templates to create the cloud-init configuration drive files (meta_data.json, user_data, and interfaces file (content/0000)), the VMX files for creating OVAs, and `ovftool` command line options. The VM artifacts are stored in `data/vmware/<vm name>`.

Figure 2: Installer Flow





CHAPTER 2

Installing CPS vDRA

- Create Installer VM in vSphere, on page 5
- Binding-VNF, on page 7
- VMware disk encryption , on page 8
- CPS Installer Commands, on page 13
- Validate Deployment, on page 17
- Redeploy VMs during the ISSM Operation, on page 19
- Redeploy VMs during the ISSM Operation with Overlay Network, on page 20

Create Installer VM in vSphere

Create the installer VM in VMware vSphere.

Download the vDRA deployer VMDKs and base image VMDKs.

Upload the VDMK File

Upload the VDMK file as shown in the following example:

```
ssh root@my-esxi-1.cisco.com
cd /vmfs/volumes/<datastore>
mkdir cps-images
cd /vmfs/volumes/<datastore>/cps-images
wget http:/<your_host>/cps-deployer-host_<version>.vmdk
```

Convert CPS Deployer VMDK to ESXi Format

Convert the CPS deployer host VMDK to ESXi format as shown in the following example:

```
ssh root@my-esxi-1.cisco.com
cd /vmfs/volumes/<datastore>/cps-images
vmkfstools --diskformat thin -i cps-deployer-host_<version>.vmdk
cps-deployer-host_<version>-esxi.vmdk
```

Create CPS Installer VM

Using the vSphere client, create the CPS Installer VM.

Procedure

Step 1 Login to the vSphere Web Client and select the blade where you want to create a new VM to install the cluster manager VM.

Step 2 Right-click on the blade and select **New Virtual Machine**. **New Virtual Machine** window opens up.

Step 3 Select **Create a new virtual machine** and click **Next** to open **Select a name and folder**.

Step 4 Enter a name for the virtual machine (for example, CPS Cluster Manager) and select the location for the virtual machine. Click **Next**.

Step 5 Select blade IP address from **Select a compute resource** window and click **Next** to open **Select storage** window.

Step 6 From **Select storage** window, select *datastorename* and click **Next** to open **Select compatibility** window.

Step 7 From **Compatible with:** drop-down list, select **`ESXi 6.7 and later`** **`ESXi 7.0 and later`** and click **Next** to open **Select a guest OS** window.

Note
Support for VMX11 is added only for fresh install. For upgrade flow (option 2/option 3), upgrade of VMX is not supported.

Step 8 From **Guest OS Family:** drop-down list, select **`Linux`** and from **Guest OS Version:** drop-down list, select **`Ubuntu Linux (64-bit)`**.

Step 9 Click **Next** to open **Customize settings** window.

Step 10 In **Virtual Hardware** tab:

a) Select 4 CPUs.

b) Select **Memory** size as **`32 GB`**.

c) Delete **New Hard Disk** (VM will use the existing disk created earlier with vmkfstools command).

d) Expand **New SCSI controller** and from **Change Type** drop-down list, select **VMware Paravirtual**.

e) 2 NICs are required (one for eth1 as internal and second for eth2 as management). One NIC already exists as default under **New Network**.

Under **New Network**, check **Connect At Power On** is selected.

f) To add another NIC, click **ADD NEW DEVICE** and from the list select **Network Adapter**.

Under **New Network**, check **Connect At Power On** is selected.

g) Click the **VM options** tab to configure additional options for the virtual machine:

- Expand the **Boot options**. In the **Firmware** field, choose the **BIOS** mode from the drop-down list that will be used to boot the virtual machine.

h) Click **Next** to open **Ready to complete** window.

Step 11 Review the settings displayed on **Ready to complete** window and click **Finish**.

Step 12 Press **Ctrl** + **Alt** +2 to go back to **Hosts and Clusters** and select the VM created above (*CPS Cluster Manager*).

a) Right-click and select **Edit Settings...**. **Virtual Hardware** tab is displayed as default.

b) Click **ADD NEW DEVICE** and from the list select **Existing Hard Disk** to open **Select File** window.

c) Navigate to **`cps-deployer-host_<version>-esxi.vmdk`** file created earlier with the vmkfstools command and click **OK**.

Step 13 Adjust hard disk size.

a) Press **Ctrl + Alt +2** to go back to **Hosts and Clusters** and select the VM created above (*CPS Cluster Manager*).
b) Right-click and select **Edit Settings...**. **Virtual Hardware** tab is displayed as default.
c) In the **Hard disk 1** text box enter `100` and click **OK**.

Step 14 Power ON the VM and open the console.

Configure network

Procedure

Step 1 Log into the VM Console as user:

Example:

```
cps, password: <password>
```

Step 2 Create the `/etc/network/interfaces` file using `vi` or using the `here document` syntax as shown in the example:

```
cps@ubuntu:~$ sudo -i
root@ubuntu:~# cat > /etc/network/interfaces <<EOF
auto lo
iface lo inet loopback

auto ens160
iface ens160 inet static
address 10.10.10.5
netmask 255.255.255.0
gateway 10.10.10.1
dns-nameservers 192.168.1.2
dns-search cisco.com
EOF
root@ubuntu:~#
```

Step 3 Restart networking as shown in the following example:

```
root@ubuntu:~# systemctl restart networking
root@ubuntu:~# ifdown ens160
root@ubuntu:~# ifup ens160
root@ubuntu:~# exit
cps@ubuntu:~$
```

What to do next

You can log in remotely using the SSH login cps/cisco123.

Binding-VNF

The process for installing the binding-vnf is the same as the dra-vnf. Create the configuration artifacts for the binding-vnf using the same VMDK. But use the binding ISO instead of DRA ISO. Similar to the dra-vnf, add a 200 GB data disk to the master and control VMs.

Artifacts Structure

```
cps@installer:/data/deployer/envs/binding-vnf$ tree
.
|-- base.env
|-- base.esxi.env
|-- user_data.yml
|-- user_data.yml.pam
`-- vms
    |-- control-0
    |   |-- control-binding-0
    |   |   |-- interfaces.esxi
    |   |   |-- user_data.yml
    |   |   |-- user_data.yml.pam
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
    |   `-- role.esxi.env
    |-- control-1
    |   |-- control-binding-1
    |   |   |-- interfaces.esxi
    |   |   |-- user_data.yml
    |   |   |-- user_data.yml.pam
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
    |   |-- role.esxi.env
    |   `-- user_data.yml.disk
    |-- master
    |   |-- master-binding-0
    |   |   |-- interfaces.esxi
    |   |   |-- user_data.yml
    |   |   |-- user_data.yml.functions
    |   |   |-- user_data.yml.pam
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
    |   `-- role.esxi.env
    `-- persistence-db
        |-- persistence-db-1
        |   |-- interfaces.esxi
        |   |-- vm.env
        |   `-- vm.esxi.env
        |-- persistence-db-2
        |   |-- interfaces.esxi
        |   |-- vm.env
        |   `-- vm.esxi.env
        |-- persistence-db-3
        |   |-- interfaces.esxi
        |   |-- vm.env
        |   `-- vm.esxi.env
        |-- role.env
        `-- role.esxi.env

11 directories, 38 files
cps@installer:/data/deployer/envs/binding-vnf$
```

VMware disk encryption

VMwareDdsk encryption is a security feature that:

- provides data-at-rest protection for virtual machine disks (VMDKs),

- Encrypts only the primary disk, not the secondary disks. The secondary disk stores only Prometheus data and saved ISO files, so it must remain intact during ISSM because it contains persistent data. Encrypting the secondary disk could lead to data corruption.
- leverages the AES-256-XTS algorithm for robust encryption, and
- integrates with VMware vSphere 7.0U2 and later versions to secure virtual environments.

This solution specifically targets the primary disks of Diameter Routing Agent (DRA) virtual machines to ensure compliance with security standards such as ISO/IEC 27001 and NIST SP 800-53.

A Native Key Provider (NKP) is a vSphere component that simplifies VM encryption management by removing the requirement for an external Key Management Server (KMS),:generates and manages cryptographic keys directly within the vCenter Server, and enables features like Encrypted vMotion and Encrypted Fault Tolerance (FT).

System requirements for VMware disk encryption

To enable disk encryption for DRA, the environment must meet these specifications:

Hardware and firmware:

- VMware-certified hardware equipped with AES-NI.
- BIOS with AES-NI enabled.

Software and licensing:

- vCenter Server version 7.0U2 or later.
- ESXi version 7.0 or later.
- VMware Enterprise Plus license.
- DRA Release 26.1 or later

.

Limitations and restrictions for disk encryption

The following constraints apply when using Native Key Provider (NKP) and disk encryption:

- NKP is not FIPS 140-2 certified. If FIPS certification is required, use an external KMS.
- vCenter must operate in FIPS-disabled mode to utilize NKP.
- Key material cannot be shared between different vCenter Servers. Disaster recovery or high availability scenarios require importing a backup to the target vCenter.
- If the NKP backup file or password is lost, all encrypted virtual machines become unrecoverable.
- Only VMFS 6 datatypes are supported for encryption. vSAN is currently not supported.
- Encryption applies only to the primary disk. Secondary disks, such as /data and /stats partitions, are not encrypted

  .

Configure the native key provider

Establish a key provider within vCenter to manage encryption keys without an external server.

Perform this task before deploying DRA virtual machines to ensure the encryption infrastructure is ready.

Note Always store the Native Key Provider backup file and its associated password in a secure, external vault. If the backup is lost or the password is forgotten, you cannot recover data from encrypted virtual machines.

Before you begin

Ensure that the vCenter Server is in FIPS-disabled mode before attempting to use the Native Key Provider. NKP is not compatible with FIPS-enabled vCenter environments.

These steps explain how to configure the Native Key Provider

Procedure

Step 1 Log in to the vSphere Client.

Step 2 Navigate to **Configure** > **Key Providers**.

Step 3 Click **Add** > **Add Native Key Provider**.

Step 4 Enter a unique name for the provider.

Example:

VM Disk Encryption.

Step 5 Select **Use Key Provider only with TPM-protected ESXi hosts** if your hardware supports it.

Step 6 Click **Add Key Provider**.

The Native Key Provider is created and appears in the list of available providers.

Back up the native key provider

Create a secure backup of the key material to prevent permanent data loss.

You must back up the NKP before it can be used to encrypt virtual machines.

Note To ensure geo redundancy, all failover sites must use the same vCenter and the same NKP file or Storage Policy Profile.

Before you begin

Ensure you have a secure location to store the backup file.

These steps describe how to back up native key provider:

Procedure

Step 1 Select the newly created key provider from the **Key Providers** list.
Step 2 Click **Backup**.
Step 3 Set a secure password in the **password** field.
Step 4 Select the **I have saved the password in a secure place** checkbox to confirm.
Step 5 Complete the backup process.

The Native Key Provider status changes to "Active" or "Backed Up," allowing it to be used for encryption tasks.

Create a VM storage policy for encryption

Define the storage rules that enforce encryption on virtual machine disks.

This policy is referenced during the DRA deployment process to ensure the primary disk is encrypted.

These steps explain how to create an encryption storage policy.

Procedure

Step 1 Navigate to **Policies and Profiles** > **VM Storage Policies**, and then click **Create**
Step 2 In the **Name** field, enter DRA Encryption.
Step 3 In the **Description** field, enter DRA Encryption Policies.
Step 4 Under **Policy Structure**, enable **Host-Based Rules**.
Step 5 In **Host Based Services** > **Encryption**, choose Use storage policy component as **Default encryption properties**.
Step 6 Click **Next** through the **Storage Compatibility** section.
Step 7 Review the configuration and click **Finish.**

The DRA Encryption policy is available for assignment to virtual machines.

Create an encryption-enabled cluster

Organize ESXi hosts into a managed group that supports DRA encryption requirements.

DRA ESXi hosts must reside in a dedicated cluster to support VMware Disk Encryption.

These steps explain how to create a new cluster.

Procedure

Step 1 Navigate to your Datacenter and then click **New Cluster**.
Step 2 In the **Name** field, enter DRA Encryption.
Step 3 (Optional) Enable **vSphere DRS** and **vSphere HA** based on your availability requirements.
Step 4 Enable **Manage all hosts in the cluster with a single image** and select the appropriate ESXi version.
Step 5 Click **Create**.

A new cluster is created in the inventory.

Add ESXi hosts to the cluster

Move existing ESXi hosts into the encryption-enabled cluster.

All hosts intended to run encrypted DRA VMs must be members of the cluster created in the previous task.

Follow these steps to add hosts to the cluster:

Procedure

Step 1 Select the DRA Encryption cluster.
Step 2 Click **Actions** > **Add Hosts** > **Existing Hosts**.
Step 3 Select the desired ESXi hosts from the list and click **Next**.
Step 4 Review the **Host Summary** and click **Next**.
Step 5 Click **Finish**.

The ESXi hosts are moved into the cluster and inherit the cluster-level configurations.

Configure the environment file for encryption

Update the DRA configuration file to enable encryption during deployment.

The *vm.esxi.env* file must be manually edited by the user to trigger the encryption logic during the VM deployment or redeployment.

These steps explain how to configure the environment file:

Before you begin

Ensure the POLICY_PROFILE matches the name of the storage policy created in vCenter.

Procedure

Step 1 Open the *vm.esxi.env* file in a text editor.

Step 2 Locate the encryption parameters.

Step 3 Set the *VM_ENCRYPTION* variable to **Enabled**. The options supported are only Enabled or Disabled.

Step 4 Set the *POLICY_PROFILE* variable to the name of your policy. .

Example:

```
VM_ENCRYPTION=Enabled
POLICY_PROFILE=DRA-Encryption-Policy
```

Step 5 Save and close the file.

The DRA deployment scripts will now apply the specified encryption policy to the primary disk of the virtual machines.

CPS Installer Commands

Command Usage

Use the `cps` command to deploy VMs. The command is a wrapper around the docker command that is required to run the deployer container.

Example:

```
function cps () {
     docker run \
         -v /data/deployer:/data/deployer \
         -v /data/vmware/:/export/ \
         -it --rm dockerhub.cisco.com/cps-docker-v2/cps deployer/deployer:latest  \
         /root/cps "$@"
}
```

To view the help for the command, run the following command: `cps -h`

```
cps@installer:~$ cps -h
usage: cps [-h] [--artifacts_abs_root_path ARTIFACTS_ABS_ROOT_PATH]
           [--export_dir EXPORT_DIR] [--deploy_type DEPLOY_TYPE]
           [--template_dir TEMPLATE_DIR]
           [--status_table_width STATUS_TABLE_WIDTH] [--skip_create_ova]
           [--skip_delete_ova]
           {install,delete,redeploy,list,poweroff,poweron,datadisk}
           vnf_artifacts_relative_path [vm_name [vm_name ...]]

positional arguments:
  {install,delete,redeploy,list,poweroff,poweron,datadisk}
                        Action to perform
  vnf_artifacts_relative_path
                        VNF artifacts directory relative to vnf artifacts root
                        path. Example: dra-vnf
  vm_name               name of virtual machine

optional arguments:
```

```
-h, --help            show this help message and exit
--artifacts_abs_root_path ARTIFACTS_ABS_ROOT_PATH
                      Absolute path to artifacts root path. Example:
                      /data/deployer/envs
--export_dir EXPORT_DIR
                      Abosolute path to store ova files and rendered
                      templates
--deploy_type DEPLOY_TYPE
                      esxi
--template_dir TEMPLATE_DIR
                      Absolute path to default templates
--status_table_width STATUS_TABLE_WIDTH
                      Number of VMs displayed per row in vm status table
--skip_create_ova     Skip the creation of ova files. If this option is
                      used, the ova files must be pre-created. This if for
                      testing and debugging
--skip_delete_ova     Skip the deletion of ova files. If this option is
                      used, the ova files are not deleted. This if for
                      testing and debugging
```

List VMs in Artifacts

Use the following command to list VMs in artifacts:

```
cps list example-dra-vnf
```

where, *example-dra-vnf* is the VNF artifacts directory.

Deploy all VMs in Parallel

Use the following command to deploy all VMs in parallel:

```
cps install example-dra-vnf
```

Deploy one or more VMs

The following example command shows how to deploy dra-director-2 and dra-worker-1:

```
cps install example-dra-vnf dra-director-2 dra-worker-1
```

Deploy all VMs with or without a Hypervisor Flag

Use the following command to install all VMs that are tagged with a ESXIHOST value matching hypervisor name as esxi-host-1 in their *vm.esxi.env* file:

cps install dra-vnf --hypervisor esxi-host-1

The following cps install command allows you to perform activities on more than one artifiactory files, which are tagged with or without *--hypervisor* flag.

cps install –addartifact *artifact-env-2*
--hypervisor *hypervisor-name*

Health Checks

Using the **--hypervisor** option that you can perform health check of docker engine and consul status of other VMs before making changes on the requested VM.

For example, if you run cps install **--hypervisor** *esxi-host-1*, then any VMs that are tagged with *esxi-host-1* are excluded and the remaining set of VMs from the artifact file is considered for health check.

VM Name	ESXiHOST
vm01	esxi-host-1
vm02	esxi-host-2
vm03	esxi-host-2

This is done to ensure that VM's on other blades are stable before performing the requested changes on their partner blade VMs. The health check fetches details of the master VM automatically from the artifiactory file and performs SSH to master, to check if the docker engine and consul status of vm02 and vm03 are in a proper state. If the state is proper, then *cps* command starts the requested operation such as install, power on, or redeploy and so on.

Delete one or more VMs

The following command is an example for deleting dra-director-1 and dra-worker-1 VMs:

Note VM deletion can disrupt services.

```
cps delete example-dra-vnf dra-director-1 dra-worker-1
```

Redeploy all VMs

Redeploying VMs involves deleting a VM and then redeploying them. If more the one VM is specified, VMs are processed serially. The following command is an example for redeploing all VMs:

Note VM deletion can disrupt services.

```
cps redeploy example-dra-vnf
```

Redeploy one or more VMs

Redeploying VMs involves deleting a VM and then redeploying them. If more the one VM is specified, VMs are processed serially. The following command is an example for redeploing two VMs:

Note VM deletion can disrupt services.

```
cps redeploy example-dra-vnf dra-director-1 control-1
```

Power down one or more VMs

The following command is an example for powering down two VMs:

Note Powering down the VM can disrupt services.

```
cps poweroff example-dra-vnf dra-director-1 dra-worker-1
```

Power up one or more VMs

The following command is an example for powering up two VMs:

Note Powering Up the VM can disrupt services.

```
cps poweron example-dra-vnf dra-director-1 dra-worker-1
```

Upgrading VMs using Diagnostics and Redeployment Health Check

Diagnostics of VMs

Use the following command to perform system diagnostics on VMs from vDRA to DB VNFs.

```
cps diagnostics dra-vnf
```

Redeployment Health Check for VMs

Use the following command to perform the redeployment health check on VMs.

```
cps redeploy dra-vnf --healthcheck yes --sysenv dra
```

Ranking Details

To upgrade the VMs, create a group of specific VMs from artifact files and place it under `/data/deployer/envs/upgradelist.txt`. It is a one-time creation process and the file has a ranking mechanism.

Based on ranking, separate the contents with a comma(,) as given.

```
Example:

cat /data/deployer/envs/upgradelist.txt
1,sk-master0
2,sk-control0,sk-dra-worker2
3,sk-control1,sk-dra-worker1
4,sk-dra-directo1,sk-dra-director2
```

The pre and postchecks for Master and Control VMs vary from other VMs.

Ranking Details		
Rank 1	Master VM Example: 1,sk-master0	If there is no master VM, then remove Rank1(1,sk-master0) from the upgradelist.txt file not to disturb the other ranks.

Ranking Details		
Rank 2	Control VM Example: 2,sk-control0, sk-dra-worker2 3,sk-control1, sk-dra-worker1	• Declare the control VMs for Ranks 2 and 3 and add one or more VMs. • If you do not redeploy control VMs, do not declare any values in the upgradelist.txt file starting with Rank 2 and 3.
Rank 3		
Rank 4	Other VMs Example: 4,sk-dra-directo1,sk-dra-director2	Do not contain either master or control VMs.

The differentiation between Rank 1(Master) and Rank2(Control) VMs is because the pre and postchecks for Master and Control VMs varies withing themselves.

Resume Redeployment

The resume option starts the VM redeployment from the last successful completion.

Consider the following scenario where the deployment occurs until site2-binding-control-0. For some reason, the VMs after site2-binding-control0 faces a problem and the automation feature terminates the execution.

```
root@ubuntu:~# cat /data/deployer/envs/upgradelist.txt
1,site2-binding-master-1
2,site2-binding-control-0,site2-persistence-db-1
3,site2-binding-control-1,site2-persistence-db-2
```

Use the **cps redeploy /data/deployer/envs/dba-vnf/ --healthcheck yes --sysenv dba** command to resume the redeployment.

Configuration and Restriction:

- The diagnostics and redeployment of VMs with the health check works only if the Master VM is active.
- For a proper health check, copy the cps.pem key used for connecting to the Master VM to the `/data/deployer/envs` folder.

Validate Deployment

Use the CLI on the master VM to validate the installation.

Connect to the CLI using the default user and password (admin/admin).

```
ssh -p 2024 admin@<master management ip address>
```

show system status

Use `show system status` command to display the system status.

Note System status percent-complete should be 100%.

```
admin@orchestrator[master-0]# show system status
system status running     true
system status upgrade     false
system status downgrade   false
system status external-services-enabled true
system status debug       false
system status percent-complete 100.0
admin@orchestrator[master-0]#
```

show system diagnostics

No diagnostic messages should appear using the following command:

```
admin@orchestrator[master-0]# show system diagnostics | tab | exclude pass
NODE       CHECK ID                       IDX  STATUS   MESSAGE
-------------------------------------------------------------

admin@orchestrator[master-0]#
```

show docker engine

All DRA-VNF VMs should be listed and in the CONNECTED state.

```
admin@orchestrator[master-0]# show docker engine
                             MISSED
ID                 STATUS    PINGS
--------------------------------------
control-0          CONNECTED  0
control-1          CONNECTED  0
dra-director-1     CONNECTED  0
dra-director-2     CONNECTED  0
dra-distributor-1  CONNECTED  0
dra-distributor-2  CONNECTED  0
dra-worker-1       CONNECTED  0
dra-worker-2       CONNECTED  0
master-0           CONNECTED  0

admin@orchestrator[master-0]#
```

show docker service

No containers should be displayed when using the exclude HEAL filter.

```
admin@orchestrator[master-0]# show docker service | tab | exclude HEAL
                                                              PENALTY
MODULE  INSTANCE NAME  VERSION  ENGINE  CONTAINER ID  STATE  BOX     MESSAGE
-------------------------------------------------------------------------------

admin@orchestrator[master-0]#
```

Redeploy VMs during the ISSM Operation

To redeploy VMs during In-Service Software Migration (ISSM) , use the following procedure:

Procedure

Step 1 Find the consul container that is having a consul leader role:

a) To find the consul leader use the following command:

 # docker exec consul-1 consul operator raft list-peers

For example, in the following output consul-3 is the leader.

```
admin@orchestrator[an-master]# docker exec consul-1 "consul operator raft list-peers"
==========output from container consul-1===========
Node                  ID                                    Address           State     Voter
RaftProtocol
consul-2.weave.local  52d5b25c-77fc-1163-0304-493b117096cd  10.46.128.2:8300  follower  true   3
consul-4.weave.local  fe68543b-ef72-66a7-7830-1c0405fd06a0  10.32.128.1:8300  follower  true   3
consul-5.weave.local  21539d8a-7d55-9cdb-c3e0-7680b448b5d5  10.32.160.1:8300  follower  true   3
consul-3.weave.local  f7a87957-a129-a12e-eb44-03bc3b385ec1  10.46.160.2:8300  leader    true   3
consul-1.weave.local  2d14416d-cc22-bcbd-e686-04bdc860332d  10.32.0.3:8300    follower  true   3
consul-7.weave.local  a3b0ba51-a8d4-68b4-b899-c20ede286e09  10.47.160.1:8300  follower  true   3
consul-6.weave.local  36d06c94-2ec5-094d-7acf-7ea190b36825  10.46.224.1:8300  follower  true   3
admin@orchestrator[an-master]#
```

Step 2 Use the following command to find the VM in which the consul leader is running:

show docker service | tab | include consul

For example, in the following output the consul leader is running in the director-0 vm.

```
admin@orchestrator[an-master]# show docker service | tab | include consul
consul               1        consul-1                 23.2.0-release  an-master         consul-1
                         HEALTHY  false    -
consul               1        consul-2                 23.2.0-release  an-control-0      consul-2
                         HEALTHY  false    -
consul               1        consul-3                 23.2.0-release  an-control-1      consul-3
                         HEALTHY  false    -
consul-dra           1        consul-4                23.2.0-release  an-dra-director-0  consul-4
                         HEALTHY  false    -
consul-dra           1        consul-5                23.2.0-release  an-dra-director-1  consul-5
                         HEALTHY  false    -
consul-dra           1        consul-6                 23.2.0-release  an-dra-worker-0    consul-6
                         HEALTHY  false    -
consul-dra           1        consul-7                 23.2.0-release  an-dra-worker-1    consul-7
                         HEALTHY  false    -
admin@orchestrator[an-master]#
```

Step 3 Perform consul leader failover in the consul leader container using **docker exec <consul-leader-container> "supervisorctl stop consul-server"** command .

Example: If the consul leader VM is same as the VM to be redeployed, then stop the consul-server in the consul leader container to perform consul leader failover.

```
admin@orchestrator[an-master]# docker exec consul-3 "supervisorctl stop consul-server"
==========output from container consul-3===========
consul-server: stopped
admin@orchestrator[an-master]#
```

Step 4 Verify the consul leader failover with another VM that will not be redeployed. Use the **docker exec consul-1 "consul operator raft list-peers"** command to verify the details as shown in the sample configuration.

```
admin@orchestrator[an-master]# docker exec consul-1 "consul operator raft list-peers"
==========output from container consul-1===========
Node                  ID                                    Address           State     Voter
RaftProtocol
consul-2.weave.local  52d5b25c-77fc-1163-0304-493b117096cd  10.46.128.2:8300  follower  true   3
consul-4.weave.local  fe68543b-ef72-66a7-7830-1c0405fd06a0  10.32.128.1:8300  leader    true   3
consul-5.weave.local  21539d8a-7d55-9cdb-c3e0-7680b448b5d5  10.32.160.1:8300  follower  true   3
consul-3.weave.local  f7a87957-a129-a12e-eb44-03bc3b385ec1  10.46.160.2:8300  follower  true   3
consul-1.weave.local  2d14416d-cc22-bcbd-e686-04bdc860332d  10.32.0.3:8300    follower  true   3
consul-7.weave.local  a3b0ba51-a8d4-68b4-b899-c20ede286e09  10.47.160.1:8300  follower  true   3
consul-6.weave.local  36d06c94-2ec5-094d-7acf-7ea190b36825  10.46.224.1:8300  follower  true   3
admin@orchestrator[an-master]#
```

Step 5 Start the consul server in the consul container stopped in step 3.

Step 6 Verify the health of the consul using the **show docker service | tab | include consul** command to ensure that the consul containers are healthy after consul leader failover.

```
admin@orchestrator[an-master]# show docker service | tab | include consul
consul               1        consul-1                  23.2.0-release  an-master         consul-1
                         HEALTHY  false    -
consul               1        consul-2                  23.2.0-release  an-control-0      consul-2
                         HEALTHY  false    -
consul               1        consul-3                  23.2.0-release  an-control-1      consul-3
                         HEALTHY  false    -
consul-dra           1        consul-4                  23.2.0-release  an-dra-director-0  consul-4
                         HEALTHY  false    -
consul-dra           1        consul-5                  23.2.0-release  an-dra-director-1  consul-5
                         HEALTHY  false    -
consul-dra           1        consul-6                  23.2.0-release  an-dra-worker-0    consul-6
                         HEALTHY  false    -
consul-dra           1        consul-7                  23.2.0-release  an-dra-worker-1    consul-7
                         HEALTHY  false    -
admin@orchestrator[an-master]#
```

Step 7 Redeploy the VM.

Redeploy VMs during the ISSM Operation with Overlay Network

ISSM Initial Configuration for Overlay Network

Weave network is used to enable the communication between containers across VMs in the vDRA solution. When the Weave network is replaced with Overlay Docker network, you can redeploy theVMs during ISSM operation.

Note This procedure is applicable only from CPS 25.1.0 or above versions with Overlay network.

Procedure

To redeploy the VM in Overlay network environment, complete the initial configuration steps:

Step 1 In the `base.env` file located in the deployer VM, add the new properties mentioned here:

```
WEAVE_ENABLED=false
OVERLAY_NETWORK=overlay
JOIN_TOKEN=SWMTKN-1-4ptz26lhvvw5y4y0hqwnbpo3n1ovxknomohutf7muepr9y8guf-btjzf1trrnu5lpnzj8mletxg4
SWARM_PORT=2377
LEADER_IP=192.168.00.00
BRIDGE_SUBNET_IPV4=172.20.0.0/16
BRIDGE_GATEWAY_IPV4=172.20.0.1
BRIDGE_SUBNET_IPV6=fd00:3984:3989::/64
BRIDGE_GATEWAY_IPV6=fd00:3984:3989::1
```

- **WEAVE_ENABLED=false**–This option allows you to launch the VM with overlay network. In case, if you want to redeploy the VM in Weave network change the **WEAVE_ENABLED=true**.
- OVERLAY_NETWORK=overlay–This is the default name of the overlay network. During VM deployment, ensure that the property value of the OVERLAY_NETWORK and the OVERLAY_NW_NAME property value remains the same.

Note
The OVERLAY_NW_NAME is located at the `/data/orchestrator/overlay-scripts` file path in the Master VM.

a) Run the **docker swarm join-token { worker | manager }** command in the master VM to get the JOIN_TOKEN, Swarm_PORT , and LEADER_IP values:

Sample Configuration:

```
cps@WPS-DRA-master:~$ docker swarm join-token manager
docker swarm join –token
SWMTKN-1-4ptz26lhvvw5y4y0hqwnbpo3n1ovxknomohutf7muepr9y8guf-8l3w5f09wi26vs0spgp6pmfcg
192.168.00.00:2377
```

Example:

```
JOIN_TOKEN= SWMTKN-1-4ptz26lhvvw5y4y0hqwnbpo3n1ovxknomohutf7muepr9y8guf-8l3w5f09wi26vs0spgp6pmfcg
LEADER_IP = 192.168.00.00
SWARM_PORT=2377
```

b) Get the properties value from the `/data/orchestrator/overlay-scripts/docker-overlay.conf` file:

```
BRIDGE_SUBNET_IPV4=172.00.0.0/16
BRIDGE_GATEWAY_IPV4=172.00.0.1
BRIDGE_SUBNET_IPV6=fd00:3984:3989::/64
BRIDGE_GATEWAY_IPV6=fd00:3984:3989::1
```

Step 2 Find the WEAVE_PASSWORD value and add the mentioned property values subsequently in the `user_data.yml` file :

```
{% if WEAVE_ENABLED is defined %}"weave_enabled": "{{ WEAVE_ENABLED }}", {% endif %}
{% if OVERLAY_NETWORK is defined %}"overlay_network": "{{ OVERLAY_NETWORK }}", {% endif %}
{% if JOIN_TOKEN is defined %}"join_token": "{{ JOIN_TOKEN }}", {% endif %}
{% if SWARM_PORT is defined %}"swarm_port": "{{ SWARM_PORT }}", {% endif %}
```

```
{% if LEADER_IP is defined %}"leader_ip": "{{ LEADER_IP }}", {% endif %}
{% if BRIDGE_SUBNET_IPV4 is defined %}"bridge_subnet_ipv4": "{{ BRIDGE_SUBNET_IPV4 }}", {% endif %}
{% if BRIDGE_GATEWAY_IPV4 is defined %}"bridge_gateway_ipv4": "{{ BRIDGE_GATEWAY_IPV4 }}", {% endif
 %}
{% if BRIDGE_SUBNET_IPV6 is defined %}"bridge_subnet_ipv6": "{{ BRIDGE_SUBNET_IPV6` }}", {% endif %}
{% if BRIDGE_GATEWAY_IPV6 is defined %}"bridge_gateway_ipv6": "{{ BRIDGE_GATEWAY_IPV6 }}", {% endif
 %}
```

ISSM Upgrade in Overlay Setup

To perform the ISSM Upgrade in Overlay Network setup maintain seven SWARM manager VMs and remaining VMs as SWARM worker.

Before you begin

- Run the **docker node demote <list of swarm manager vm name>** depromote command before redeploying the swarm manager VM.
- In ISSM upgrade/redeployment, each set of the ISSM should not contain VMs which are marked as Leader/Reachable and empty in MANAGER STATUS column together. Separate the empty value VMs as a different set and Reachable/Leader VMs as a different set.

Procedure

Step 1 Use the **docker node ls** command to view the VM SWARM setup details:

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                    STATUS    AVAILABILITY   MANAGER STATUS

o83ancg9iy9a1ezqgcyhsu2z2     WPS-DRA-control-0           Ready     Active         Leader

jj0tmks06n6nemxhrbvo6x9ir     WPS-DRA-control-1           Ready     Active         Reachable

qh3t8vk12kqcwhnd5xvn0y62p     WPS-DRA-dra-director-1      Ready     Active         Reachable

pq9tf4bslnnhzqejjwurr9q5b     WPS-DRA-dra-director-2      Ready     Active         Reachable

yyxvcwc1dgchz6gzbqkyavuxg     WPS-DRA-dra-distributor-1   Ready     Active

vnt2pdvn7jfzw3cnjxbgpv3bt     WPS-DRA-dra-worker-1        Ready     Active         Reachable

ifg80p0jwmnwjf7wnk9d6vz61     WPS-DRA-dra-worker-2        Ready     Active         Reachable

gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master              Ready     Active         Reachable
cps@WPS-DRA-master:~$
```

Step 2 Login to the the master VM and enter `docker node demote WPS-DRA-master` command to demote to SWARM worker.

```
cps@WPS-DRA-master:~$ docker node demote WPS-DRA-master
```

Step 3 In deployer VM, delete the master VM using this command:

```
root@ubuntu:/data/deployer/envs# cps delete dra-vnf WPS-DRA-master
```

Step 4 Run the `docker swarm join-token manager` command with any one of the control VMs and get the manager token and leader IP. Update the `base.env` file in deployer VM.

```
cps@WPS-DRA-control-0:~$ docker swarm join-token manager
result:
docker swarm join --token
SWMTKN-1-4ptz26lhvvw5y4y0hqwnbpo3n1ovxknomohutf7muepr9y8guf-8l3w5f09wi26vs0spgp6pmfcg 192.168.00.00:0000
```

Step 5 In deployer VM, install a new master VM using **cps install dra-vnf WPS-DRA-master** command. and remove the old master VM entry from the docker node list using the given command:

a) Ensure the system is healthy

b) Remove the old master VM entry from the docker node list using this command:

```
cps@WPS-DRA-master:~$ docker node rm <old node ID which is down status now>
```

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                  STATUS    AVAILABILITY   MANAGER STATUS
xslmmoig3oori0lsubgbazlsf     WPS-DRA-control-0         Ready     Active         Reachable
pb5ppz6o8iop5ifabxs8bnzd0     WPS-DRA-control-1         Ready     Active         Reachable
ow8e1li67kneu91766vsndari     WPS-DRA-dra-director-1    Ready     Active         Leader
m2vbfprnw0ivkpdt9cpxpp96t     WPS-DRA-dra-director-2    Ready     Active         Reachable
7uo7c7ru0hl55g7r23jlkra9y     WPS-DRA-dra-distributor-1  Ready     Active
k7pdag4jv50r0hws0x262nbxa     WPS-DRA-dra-worker-1      Ready     Active         Reachable
v9hsifa9ck7tmvb6aoputieal     WPS-DRA-dra-worker-2      Ready     Active         Reachable
gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master            Ready     Active         Reachable
qley4ntzs00bpwmh9whfdjku4     WPS-DRA-master            Down      Active
```

For example, enter the given command for the given ID:

```
cps@WPS-DRA-master:~$ docker node rm qley4ntzs00bpwmh9whfdjku4
```

Here is the output after removing the old master entry:

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                  STATUS    AVAILABILITY   MANAGER STATUS

xslmmoig3oori0lsubgbazlsf     WPS-DRA-control-0         Ready     Active         Reachable

pb5ppz6o8iop5ifabxs8bnzd0     WPS-DRA-control-1         Ready     Active         Reachable

ow8e1li67kneu91766vsndari     WPS-DRA-dra-director-1    Ready     Active         Leader

m2vbfprnw0ivkpdt9cpxpp96t     WPS-DRA-dra-director-2    Ready     Active         Reachable

7uo7c7ru0hl55g7r23jlkra9y     WPS-DRA-dra-distributor-1  Ready     Active

k7pdag4jv50r0hws0x262nbxa     WPS-DRA-dra-worker-1      Ready     Active         Reachable

v9hsifa9ck7tmvb6aoputieal     WPS-DRA-dra-worker-2      Ready     Active         Reachable

gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master            Ready     Active         Reachable
```

Step 6 Run the given command in master VM and get the manager token and leader IP. Update the `base.env` file in the deployer VM.

```
cps@WPS-DRA-master:~$ docker swarm join-token manager
 docker swarm join --token
SWMTKN-1-4ptz26lhvvw5y4y0hqwnbpo3n1ovxknomohutf7muepr9y8guf-8l3w5f09wi26vs0spgp6pmfcg 192.168.00.00:0001
```

Step 7 The ISSM should not contain VMs marked as Leader/Reachable and empty in MANAGER STATUS column together. Run the given command to depromote and redeploy the swarm manager VMs:

```
cps@WPS-DRA-master:~$ docker node demote WPS-DRA-control-0 WPS-DRA-dra-director-1 WPS-DRA-dra-worker-1

root@ubuntu:/data/deployer/envs# cps delete dra-vnf WPS-DRA-control-0 WPS-DRA-dra-director-1
WPS-DRA-dra-worker-1
root@ubuntu:/data/deployer/envs# cps install dra-vnf WPS-DRA-control-0 WPS-DRA-dra-director-1
WPS-DRA-dra-worker-1


cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                    STATUS    AVAILABILITY   MANAGER STATUS

o83ancg9iy9a1ezqgcyhsu2z2     WPS-DRA-control-0           Ready     Active         Reachable

xslmmoig3oori0lsubgbazlsf     WPS-DRA-control-0           Down      Active
pb5ppz6o8iop5ifabxs8bnzd0     WPS-DRA-control-1           Ready     Active         Reachable

ow8e1li67kneu91766vsndari     WPS-DRA-dra-director-1      Down      Active
qh3t8vk12kqcwhnd5xvn0y62p     WPS-DRA-dra-director-1      Ready     Active         Reachable

m2vbfprnw0ivkpdt9cpxpp96t     WPS-DRA-dra-director-2      Ready     Active         Leader

7uo7c7ru0hl55g7r23jlkra9y     WPS-DRA-dra-distributor-1   Ready     Active

k7pdag4jv50r0hws0x262nbxa     WPS-DRA-dra-worker-1        Down      Active
vnt2pdvn7jfzw3cnjxbgpv3bt     WPS-DRA-dra-worker-1        Ready     Active         Reachable

v9hsifa9ck7tmvb6aoputieal     WPS-DRA-dra-worker-2        Ready     Active         Reachable

gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master              Ready     Active         Reachable
```

For example, enter this command for the given set of IDs:

```
cps@WPS-DRA-master:~$ docker node rm xslmmoig3oori0lsubgbazlsf ow8e1li67kneu91766vsndari
k7pdag4jv50r0hws0x262nbxa
```

Here is the sample output:

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                    STATUS    AVAILABILITY   MANAGER STATUS

o83ancg9iy9a1ezqgcyhsu2z2     WPS-DRA-control-0           Ready     Active         Reachable

pb5ppz6o8iop5ifabxs8bnzd0     WPS-DRA-control-1           Ready     Active         Reachable

qh3t8vk12kqcwhnd5xvn0y62p     WPS-DRA-dra-director-1      Ready     Active         Reachable

m2vbfprnw0ivkpdt9cpxpp96t     WPS-DRA-dra-director-2      Ready     Active         Leader

7uo7c7ru0hl55g7r23jlkra9y     WPS-DRA-dra-distributor-1   Ready     Active

vnt2pdvn7jfzw3cnjxbgpv3bt     WPS-DRA-dra-worker-1        Ready     Active         Reachable

v9hsifa9ck7tmvb6aoputieal     WPS-DRA-dra-worker-2        Ready     Active         Reachable

gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master              Ready     Active         Reachable
```

Step 8 It is importanat for the ISSM not to have VMs marked as Leader/Reachable and empty in MANAGER STATUS column concurrently. Ensure the system is healthy and start another set of ISSM manager VMs upgrade.

```
cps@WPS-DRA-master:~$ docker node demote WPS-DRA-control-1 WPS-DRA-dra-director-2 WPS-DRA-dra-worker-2

root@ubuntu:/data/deployer/envs# cps delete dra-vnf WPS-DRA-control-1 WPS-DRA-dra-director-2
WPS-DRA-dra-worker-2
```

```
root@ubuntu:/data/deployer/envs# cps install dra-vnf WPS-DRA-control-1 WPS-DRA-dra-director-2
WPS-DRA-dra-worker-2
```

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                    STATUS    AVAILABILITY   MANAGER STATUS
o83ancg9iy9a1ezqgcyhsu2z2     WPS-DRA-control-0           Ready     Active         Leader
pb5ppz6o8iop5ifabxs8bnzd0     WPS-DRA-control-1           Down      Active
jj0tmks06n6nemxhrbvo6x9ir     WPS-DRA-control-1           Ready     Active         Reachable
qh3t8vk12kqcwhnd5xvn0y62p     WPS-DRA-dra-director-1      Ready     Active         Reachable
m2vbfprnw0ivkpdt9cpxpp96t     WPS-DRA-dra-director-2      Down      Active
pq9tf4bslnnhzqejjwurr9q5b     WPS-DRA-dra-director-2      Ready     Active         Reachable
7uo7c7ru0hl55g7r23jlkra9y     WPS-DRA-dra-distributor-1   Ready     Active
vnt2pdvn7jfzw3cnjxbgpv3bt     WPS-DRA-dra-worker-1        Ready     Active         Reachable
v9hsifa9ck7tmvb6aoputieal     WPS-DRA-dra-worker-2        Down      Active
ifg80p0jwmnwjf7wnk9d6vz61     WPS-DRA-dra-worker-2        Ready     Active         Reachable
gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master              Ready     Active         Reachable
```

Remove the old VM entries that are in down status:

```
cps@WPS-DRA-master:~$ docker node rm pb5ppz6o8iop5ifabxs8bnzd0 m2vbfprnw0ivkpdt9cpxpp96t
v9hsifa9ck7tmvb6aoputieal
```

Here is the output:

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                    STATUS    AVAILABILITY   MANAGER STATUS

o83ancg9iy9a1ezqgcyhsu2z2     WPS-DRA-control-0           Ready     Active         Leader

jj0tmks06n6nemxhrbvo6x9ir     WPS-DRA-control-1           Ready     Active         Reachable

qh3t8vk12kqcwhnd5xvn0y62p     WPS-DRA-dra-director-1      Ready     Active         Reachable

pq9tf4bslnnhzqejjwurr9q5b     WPS-DRA-dra-director-2      Ready     Active         Reachable

7uo7c7ru0hl55g7r23jlkra9y     WPS-DRA-dra-distributor-1   Ready     Active

vnt2pdvn7jfzw3cnjxbgpv3bt     WPS-DRA-dra-worker-1        Ready     Active         Reachable

ifg80p0jwmnwjf7wnk9d6vz61     WPS-DRA-dra-worker-2        Ready     Active         Reachable
gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master              Ready     Active         Reachable
```

Step 9 Ensure the system is healthy and start another set of ISSM swam worker VMs upgrade. For the given ISSM upgrade, use the join-token as worker for the non-swarm manager VMs. For worker token, run the mentioned command in master and update the `base.env` file:

```
cps@WPS-DRA-master:~$ docker swarm join --token
SWMTKN-1-4ptz26lhvvw5y4y0hqwnbpo3n1ovxknomohutf7muepr9y8guf-btjzf1trrnu5lpnzj8mletxg4 192.168.00.00:0002

root@ubuntu:/data/deployer/envs# cps delete WPS-DRA-dra-distributor-1
root@ubuntu:/data/deployer/envs# cps install WPS-DRA-dra-distributor-1
```

Note
The sample setup has one swarm worker VM, but you can add additional swarm worker VMs as required. It is possible to have multiple swarm worker VMs in other setups.

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                    STATUS    AVAILABILITY   MANAGER STATUS
o83ancg9iy9a1ezqgcyhsu2z2     WPS-DRA-control-0           Ready     Active         Leader
jj0tmks06n6nemxhrbvo6x9ir     WPS-DRA-control-1           Ready     Active         Reachable
qh3t8vk12kqcwhnd5xvn0y62p     WPS-DRA-dra-director-1      Ready     Active         Reachable
pq9tf4bslnnhzqejjwurr9q5b     WPS-DRA-dra-director-2      Ready     Active         Reachable
7uo7c7ru0hl55g7r23jlkra9y     WPS-DRA-dra-distributor-1   Down      Active
yyxvcwc1dgchz6gzbqkyavuxg     WPS-DRA-dra-distributor-1   Ready     Active
```

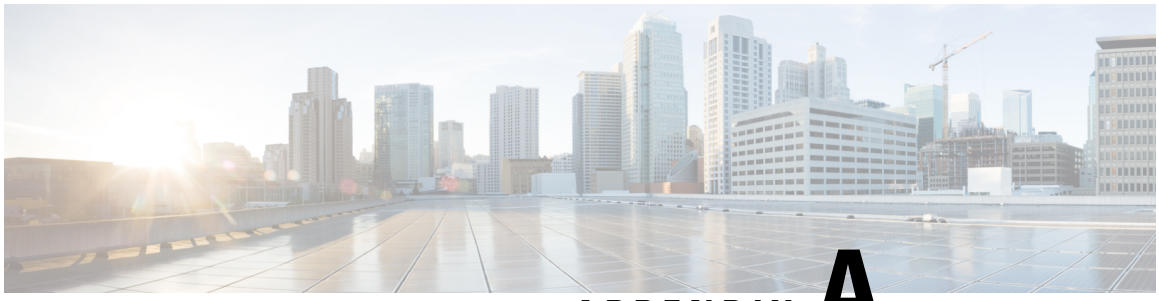
```
vnt2pdvn7jfzw3cnjxbgpv3bt     WPS-DRA-dra-worker-1        Ready     Active         Reachable
ifg80p0jwmnwjf7wnk9d6vz61     WPS-DRA-dra-worker-2        Ready     Active         Reachable
gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master              Ready     Active         Reachable
```

Remove the old VM entries that are in down status:

```
cps@WPS-DRA-master:~$ docker node rm 7uo7c7ru0hl55g7r23jlkra9y
```

Here is the output:

```
cps@WPS-DRA-master:~$ docker node ls
ID                            HOSTNAME                    STATUS    AVAILABILITY   MANAGER STATUS
o83ancg9iy9a1ezqgcyhsu2z2     WPS-DRA-control-0           Ready     Active         Leader
jj0tmks06n6nemxhrbvo6x9ir     WPS-DRA-control-1           Ready     Active         Reachable
qh3t8vk12kqcwhnd5xvn0y62p     WPS-DRA-dra-director-1      Ready     Active         Reachable
pq9tf4bslnnhzqejjwurr9q5b     WPS-DRA-dra-director-2      Ready     Active         Reachable
yyxvcwc1dgchz6gzbqkyavuxg     WPS-DRA-dra-distributor-1   Ready     Active
vnt2pdvn7jfzw3cnjxbgpv3bt     WPS-DRA-dra-worker-1        Ready     Active         Reachable
ifg80p0jwmnwjf7wnk9d6vz61     WPS-DRA-dra-worker-2        Ready     Active         Reachable
gjgnp3s34agpkoue83wqobhju *   WPS-DRA-master              Ready     Active         Reachable
```



APPENDIX A

Installation Examples

- DRA-VNF Example, on page 27

DRA-VNF Example

This section provides an example for configuring the installer with a dra-vnf test bed. The dra-vnf example includes the following roles and VMs:

- master:

  master-0

- control:

  control-0

  control-1

- DRA Director:

  dra-director-1

  dra-director-2

- DRA Worker:

  dra-worker-1

  dra-worker-2

- DRA Distributor:

  dra-distributor-1

  dra-distributor-2

  dra-distributor-3

  dra-distributor-4

Artifacts Structure Example

```
cps@installer:/data/deployer/envs/dra-vnf$ tree
.
```

```
|-- base.env
|-- base.esxi.env
|-- user_data.yml
|-- user_data.yml.pam
`-- vms
    |-- control-0
    |   |-- control-0
    |   |   |-- interfaces.esxi
    |   |   |-- user_data.yml
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
    |   `-- role.esxi.env
    |-- control-1
    |   |-- control-1
    |   |   |-- interfaces.esxi
    |   |   |-- user_data.yml
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
    |   `-- role.esxi.env
    |-- dra-director
    |   |-- dra-director-1
    |   |   |-- interfaces.esxi
    |   |   |-- user_data.yml
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- dra-director-2
    |   |   |-- interfaces.esxi
    |   |   |-- user_data.yml
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
    |   `-- role.esxi.env
    |-- dra-distributor
    |   |-- dra-distributor-1
    |   |   |-- interfaces.esxi
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- dra-distributor-2
    |   |   |-- interfaces.esxi
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- dra-distributor-3
    |   |   |-- interfaces.esxi
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- dra-distributor-4
    |   |   |-- interfaces.esxi
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
    |   |-- role.esxi.env
    |   `-- user_data.yml
    |-- dra-worker
    |   |-- dra-worker-1
    |   |   |-- interfaces.esxi
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- dra-worker-2
    |   |   |-- interfaces.esxi
    |   |   |-- vm.env
    |   |   `-- vm.esxi.env
    |   |-- role.env
```

```
    |   `-- role.esxi.env
    `-- master
        |-- master-0
        |   |-- interfaces.esxi
        |   |-- user_data.yml
        |   |-- vm.env
        |   `-- vm.esxi.env
        |-- role.env
        `-- role.esxi.env

18 directories, 55 files
cps@installer:/data/deployer/envs/dra-vnf$
```

Top Level Directory

```
/data/deployer/envs/example-dra-vnf/base.env
/data/deployer/envs/example-dra-vnf/base.esxi.env
/data/deployer/envs/example-dra-vnf/user_data.yml
/data/deployer/envs/example-dra-vnf/base.esxi.env
/data/deployer/envs/example-dra-vnf/esxi
/data/deployer/envs/example-dra-vnf/vms
```

base.env

All the settings in the `base.env` file can be overridden in `vms/role/role.env` and `vms/role/vm_name/vm.env` files.

```
MASTER_IP=192.169.21.10
INTERNAL_NETWORK=192.169.21.0/24
WEAVE_PASSWORD=cisco123
CLUSTER_ID=test-cluster
SYSTEM_ID=test-system
```

MASTER_IP: Internal address of master VM.

base.esxi.env

All the settings in the `base.esxi.env` file can be overridden in the `vms/role/role.esxi.env` and `vms/role/vm_name/vm.esxi.env` files.

```
VMDK="cps-docker-host_18.0.1.dra.vmdk"
VMDK_DISK_TYPE="thick"
VSPHERE_HOST="example-vsphere.cisco.com"
VSPHERE_USER="administrator@vsphere.local"
VSPHERE_PASSWORD="foo123"
VSPHERE_DISABLE_SSL_VERIFICATION="True"
VSPHERE_RESERVE_MEMORY="True"
DATACENTER="Microservices"
```

- VMDK: Place the VMDK file at the top level directory of your VNF environment structure `example-dra-vnf/microservices.vmdk_file_name`.

  Another option is to specify the full path such as `/data/deployer/envs/images/microservices.vmdk_file_name`

  Replace *microservices.vmdk_file_name* with the actual VMDK file name.

- VMDK_DISK_TYPE: VMDK disk type. See the link for a list of supported disk types.
- VSPHERE_HOST: DNS name or IP address of the vSphere host.

- VSPHERE_USER: (Optional) Login user for vSphere. If the user name is not specified, installer prompts user for vSphere login user name.
- VSPHERE_PASSWORD: (Optional) vSphere password. If the password is not specified, installer prompts user for password
- VSPHERE_DISABLE_SSL_VERIFICATION: (Optional) Disable verification of vSphere SSL Certificate. This is necessary if your vSphere server is using a Self Signed Certificate
- VSPHERE_RESERVE_MEMORY: (Optional) Reserve VM's memory before starting the VM
- DATACENTER: Datacenter for VM placement.

user_data.yml

Use the Jinja2 template to create the user data file for cloud-init.

Cloud-init user data template: This file is for reference only. You need to create cloud-init file based on your requirements.

```
#cloud-config
debug: True
output: {all: '| tee -a /var/log/cloud-init-output.log'}

users:
  - name: cps
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    groups: docker
    ssh-authorized-keys:
      - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDzjJjndIvUiBta4VSIbd2gJmlMWcQ8wtejg
        AbiXtoFZdtMdo9G0ZDEOtxHNNDPwWujMiYAkZhZWX/zON9raavU8lg cps@root-public-key

resize_rootfs: true

write_files:
  - path: /root/swarm.json
    content: |
     {
        "role": "{{ ROLE }}",
        "identifier": "{{ IDENTIFIER }}",
        "master": "{{ MASTER_IP }}",
        "network": "{{ INTERNAL_NETWORK }}",
        {% if WEAVE_PASSWORD is defined %}"weavePw": "{{ WEAVE_PASSWORD }}", {% endif %}
        "zing": "{{ RUN_ZING | default(1) }}",
        "cluster_id": "{{ CLUSTER_ID }}",
        "system_id": "{{ SYSTEM_ID }}"
     }
    owner: root:root
    permissions: '0644'
  - path: /home/cps/.bash_aliases
    encoding: text/plain
    content: |
      # A convenient shortcut to get to the Orchestrator CLI
      alias cli="ssh -p 2024 admin@localhost"
    owner: cps:cps
    permissions: '0644'
runcmd:
 - [vmware-toolbox-cmd, timesync, enable ]
```

example-dra-vnf/vms/*role*

```
example-dra-vnf/master/role.env
example-dra-vnf/master/role.esxi.env
example-dra-vnf/master/master-0
```

role.env

All settings in the `role.env` file can be overridden in the `vms/role/vm_name/vm.env` file. In non-master roles the `role.env` file is empty.

```
CPS_ISO="cisco-policy-dra.iso"
```

where, *CPS_ISO* is the CPS ISO file. This is required for master virtual machines.

Not used in non-master virtual machines. It is possible to specify this with a full path `/data/deployer/envs/images/cisco-policy-dra.iso`.

role.esxi.env

All settings in the `role.esxi.env` file can be overridden in the `vms/vm_name/vm.esxi.env` file.

```
CPU=16
RAM=65536
NETWORK_0=Management
NETWORK_1=Internal
# Data disk size in GB
VM_DATA_DISK_SIZE="200"
VM_DATA_DISK_TYPE="thick"
```

- CPU: Number of CPUs.
- RAM: Memory in megabytes (65536/1024 = 64 GB)
- NETWORK_0: The name of the first network assigned to the VM. Name is case sensitive and must match the network name configured in vSphere. Network interface names are defined using the scheme in "Interface Numbering" section.

  Add a NETWORK_*N* setting for each network required.
- VM_DATA_DISK_SIZE: Data disk size in GB for master and control VMs.
- VM_DATA_DISK_TYPE: VM data disk type. See the link for a list of supported disk types.

Data Disk

A data disk is a separate disk for the control and master virtual machines and is configured in the artifacts environment files before installing a CPS system. The data has a `/data` partition and a `/stats` partition. Perform the following steps to add a data disk to master and control VMs.

- Specify VM_DATA_DISK_SIZE and VM_DATA_DISK_TYPE in `example-env/vms/<role>/role.esxi.env` file.
- Specify VM_DATA_VMDK_ROOT_PATH and VM_DATA_DISK_NAME in `example-env/vms/<role>/role.esxi.env` file.
- Specify disk file system and mount point in `example-env/vms/<role>/<vm_name>/user_data.yml` file.

The installer checks for an existing data disk in `VM_DATA_VMDK_ROOT_PATH/<disk_name>`. If a data disk exists, the disk is attached to the target VM. If a data disk does not exist, the installer creates a new VMDK disk and attaches it to the VM. Cloud init is responsible for formatting the disk and mounting it. If the data disk has an ext-4 file system, cloud-int does not reformat the disk, preserving existing data.

If a VM is deleted with the deployer container's `cps delete example-dra control-0` command, the data disk is detached before the VM is deleted. Detached disks are not deleted when the VM is deleted.

master-0

The master-0 directory is the name of a VM. This directory name must match the hostname of the VM.

```
example-dra-vnf/vms/master/vm_name
```

Directory containing configuration information for a VM

```
example-dra-vnf/vms/master/master-0/interfaces.esxi
example-dra-vnf/vms/master/master-0/vm.env
example-dra-vnf/vms/master/master-0/vm.esxi.env
```

interfaces.esxi

The contents of the `interfaces.esxi` file are placed in `/etc/network/interfaces` file on the VM. Any valid content for the ubuntu `/etc/network/interfaces` file can be placed in `interfaces.esxi`.

```
auto lo
iface lo inet loopback

auto ens160
iface ens160 inet static
address 10.10.10.155
netmask 255.255.255.0
gateway 10.10.10.1
dns-nameservers 172.10.5.25 172.11.5.25 172.12.5.25

auto ens192
iface ens192 inet static
address 192.169.21.10
netmask 255.255.255.0
```

vm.env

```
HOSTNAME=master-0
FQDN=master-0.local
```

vm.esxi.env

```
ESXI_DNS_NAME="example-esxi-1.cisco.com"
DATASTORE="datastore1"
VM_DATA_VMDK_ROOT_PATH="[datastore1] data-disks"
VM_DATA_DISK_NAME="master-0-data.vmdk"
```

- ESXI_DNS_NAME: DNS name of the VM's target ESXi server.
- ESXI_IP: IP address of ESXi server. This can be used instead of ESXI_DNS_NAME. If both, ESXI_DNS_NAME and ESXI_IP are specified, ESXI_DNS_NAME is used.

vCenter always directs the API client to the DNS name of the target ESXi server regardless if the EXSi host's IP address or DNS name is specified. The installation fails if the deployer VM cannot resolve the ESXi's DNS

name. To avoid this, update the "cps" bash function in the file `/etc/bash.aliases` and add `--add-host <esxi dns name>:<ip address>` for each ESXi server. Use `sudo` to modify the file.

```
/etc/bash.aliases
function cps () {
     docker run \
         --add-host esxi-1.example.com:10.0.0.1 \
         --add-host esxi-2.example.com:10.0.0.2 \
         -v /data/deployer:/data/deployer \
         -v /data/vmware/:/export/ \
         -it --rm dockerhub.cisco.com/cps-docker-v2/cps-deployer/deployer:latest \
         /root/cps "$@"
}
```

- DATASTORE: Case sensitive name of the vSphere datastore used to store the VM.
- VM_DATA_VMDK_ROOT_PATH: Root path to store the master or control VM's data disk.
- VM_DATA_DISK_NAME: Name of the VMDK disk.

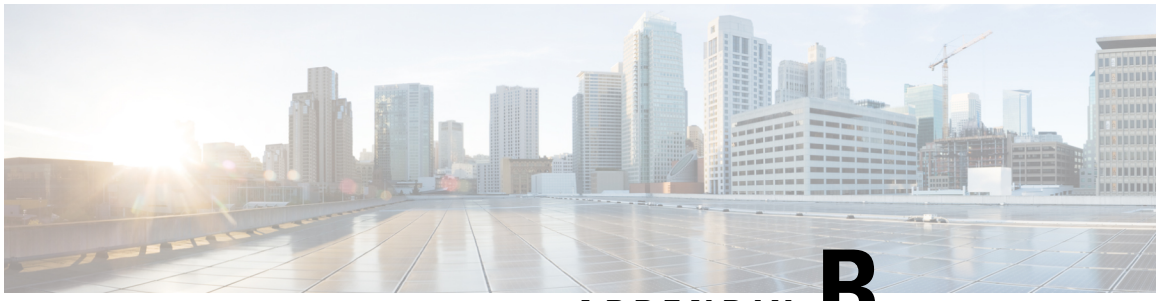
VM Level user_data.yml for Data Disks

Place this file at the VM level for master and control VMs when using a separate data disks.

Note This file is for reference only. You need to create `user_data.yml` file based on your requirements.

```
#cloud-config
# ESC velocity escape variable during deployment
#set ( $DS = "$" )
debug: True
output: {all: '| tee -a /var/log/cloud-init-output.log'}

users:
  - name: cps
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    groups: docker
    ssh-authorized-keys:
      - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDzjJjndIvUiBta4VSIbd2g
        JmlMWcQ8wtejgAbiXtoFZdtMdo9G0ZDEOtxHNNDPwWujMiYAkZhZWX/zON9raav
        U8lgD9+YcRopWUtujIC71YjtoxIj   EWEaj/50jegN cps@root-public-key

resize_rootfs: true

write_files:
  - path: /root/swarm.json
    content: |
     {
        "role": "{{ ROLE }}",
        "identifier": "{{ IDENTIFIER }}",
        "master": "{{ MASTER_IP }}",
        "network": "{{ INTERNAL_NETWORK }}",
        {% if WEAVE_PASSWORD is defined %}"weavePw": "{{ WEAVE_PASSWORD }}", {% endif %}
        "zing": "{{ RUN_ZING | default(1) }}",
        "cluster_id": "{{ CLUSTER_ID }}",
        "system_id": "{{ SYSTEM_ID }}"
     }
    owner: root:root
    permissions: '0644'
  - path: /home/cps/.bash_aliases
    encoding: text/plain
```

```
    content: |
      # A convenient shortcut to get to the Orchestrator CLI
      alias cli="ssh -p 2024 admin@localhost"
      alias pem="wget --quiet http://171.70.34.121/microservices/latest/cps.pem ; chmod 400
 cps.pem ; echo 'Retrieved \"cps.pem\" key file'"
    owner: cps:cps
    permissions: '0644'

disk_setup:
  /dev/sdb:
      table_type: 'gpt'
      layout:
          - 35
          - 65
      overwrite: False
fs_setup:
  - label: DATA
    device: /dev/sdb
    filesystem: 'ext4'
    partition: auto
    overwrite: False
  - label: STATS
    device: /dev/sdb
    filesystem: 'ext4'
    partition: auto
    overwrite: False

mounts:
 - [ "LABEL=DATA", /data, "ext4", "defaults,nofail", "0", "2" ]
 - [ "LABEL=STATS", /stats, "ext4", "defaults,nofail", "0", "2" ]
runcmd:
 - [vmware-toolbox-cmd, timesync, enable ]
```



APPENDIX B

Listening Ports in DRA Deployment

• Listening Ports in DRA Deployment, on page 35

Listening Ports in DRA Deployment

The following tables provides information about listening ports in DRA deployment.

Table 3: DRA-VNF Listening Ports

VM	Protocol	Local Address	Program Name	Module Name
Deployer	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp6	:::22	sshd	SSH Daemon
DRA-Master	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp6	:::22	sshd	SSH Daemon
	tcp6	:::443	docker-proxy	haproxy-common
	tcp6	:::2022	docker-proxy	configuration-engine
	tcp6	:::5000	docker-proxy	registry
	tcp6	:::2024	docker-proxy	configuration-engine
	tcp6	:::5001	docker-proxy	registry
	tcp6	:::9997	docker-proxy	haproxy-common
	tcp6	:::9998	docker-proxy	haproxy-common
	tcp6	:::9999	docker-proxy	haproxy-common

VM	Protocol	Local Address	Program Name	Module Name
Distributor VMs	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp6	:::9100	node_exporter	Node Exporter
	tcp6	:::22	sshd	SSH Daemon
Worker VMs	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp6	:::9100	node_exporter	Node Exporter
	tcp6	:::22	sshd	SSH Daemon
DRA-Control VMs	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp6	:::27027	docker-proxy	Mongo Daemon
	tcp6	:::27028	docker-proxy	Mongo Daemon
	tcp6	:::27029	docker-proxy	Mongo Daemon
	tcp6	:::27030	docker-proxy	Mongo Daemon
	tcp6	:::22	sshd	SSH Daemon
	tcp6	:::27031	docker-proxy	Mongo Daemon
	tcp6	:::27032	docker-proxy	Mongo Daemon
	tcp6	:::27033	docker-proxy	Mongo Daemon
	tcp6	:::27034	docker-proxy	Mongo Daemon
	tcp6	:::8443	docker-proxy	haproxy-dra
	tcp6	:::27035	docker-proxy	Mongo Daemon
	tcp6	:::443	docker-proxy	haproxy-common
	tcp6	:::27036	docker-proxy	Mongo Daemon

VM	Protocol	Local Address	Program Name	Module Name
DRA-Control VMs	tcp6	:::27037	docker-proxy	Mongo Daemon
	tcp6	:::27038	docker-proxy	Mongo Daemon
	tcp6	:::27039	docker-proxy	Mongo Daemon
	tcp6	:::27040	docker-proxy	Mongo Daemon
	tcp6	:::27041	docker-proxy	Mongo Daemon
	tcp6	:::27042	docker-proxy	Mongo Daemon
	tcp6	:::27043	docker-proxy	Mongo Daemon
	tcp6	:::27044	docker-proxy	Mongo Daemon
	tcp6	:::27045	docker-proxy	Mongo Daemon
	tcp6	:::27046	docker-proxy	Mongo Daemon
	tcp6	:::27047	docker-proxy	Mongo Daemon
	tcp6	:::2023	docker-proxy	configuration-engine
	tcp6	:::2024	docker-proxy	configuration-engine
	tcp6	:::27017	docker-proxy	Mongo Daemon
	tcp6	:::2025	docker-proxy	configuration-engine

VM	Protocol	Local Address	Program Name	Module Name
DRA-Control VMs	tcp6	:::27018	docker-proxy	Mongo Daemon
	tcp6	:::2026	docker-proxy	Stats
	tcp6	:::27019	docker-proxy	Mongo Daemon
	tcp6	:::6379	docker-proxy	control-plane
	tcp6	:::10443	docker-proxy	zvision
	tcp6	:::27020	docker-proxy	Mongo Daemon
	tcp6	:::9100	node_exporter	Node Exporter
	tcp6	:::27021	docker-proxy	Mongo Daemon
	tcp6	:::9997	docker-proxy	haproxy-common
	tcp6	:::27022	docker-proxy	Mongo Daemon
	tcp6	:::9998	docker-proxy	haproxy-common
	tcp6	:::27023	docker-proxy	Mongo Daemon
	tcp6	:::9999	docker-proxy	haproxy-common
	tcp6	:::27024	docker-proxy	Mongo Daemon
	tcp6	:::27025	docker-proxy	Mongo Daemon
	tcp6	:::27026	docker-proxy	Mongo Daemon

VM	Protocol	Local Address	Program Name	Module Name
Director VMs	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp6	:::4868	docker-proxy	diameter-endpoint
	tcp6	:::4869	docker-proxy	diameter-endpoint
	tcp6	:::4870	docker-proxy	diameter-endpoint
	tcp6	:::4871	docker-proxy	diameter-endpoint
	tcp6	:::4872	docker-proxy	diameter-endpoint
	tcp6	:::4873	docker-proxy	diameter-endpoint
	tcp6	:::4874	docker-proxy	diameter-endpoint
	tcp6	:::4875	docker-proxy	diameter-endpoint
	tcp6	:::6379	docker-proxy	control-plane
	tcp6	:::4876	docker-proxy	diameter-endpoint
	tcp6	:::9100	node_exporter	Node Exporter
	tcp6	:::4877	docker-proxy	diameter-endpoint
	tcp6	:::9997	docker-proxy	haproxy-common
	tcp6	:::4878	docker-proxy	diameter-endpoint
	tcp6	:::9998	docker-proxy	haproxy-common
	tcp6	:::9999	docker-proxy	haproxy-common
	tcp6	:::22	sshd	SSH Daemon
	tcp6	:::443	docker-proxy	haproxy-common

Table 4: Binding-VNF Listening Ports

VM	Protocol	Local Address	Program Name	Module Name
Binding-Master	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp6	:::27019	docker-proxy	Mongo Daemon
	tcp6	:::27020	docker-proxy	Mongo Daemon
	tcp6	:::9100	node_exporter	Node Exporter
	tcp6	:::27021	docker-proxy	Mongo Daemon
	tcp6	:::9997	docker-proxy	haproxy-common
	tcp6	:::27022	docker-proxy	Mongo Daemon
	tcp6	:::9998	docker-proxy	haproxy-common
	tcp6	:::27023	docker-proxy	Mongo Daemon
	tcp6	:::9999	docker-proxy	haproxy-common
	tcp6	:::27024	docker-proxy	Mongo Daemon
	tcp6	:::27025	docker-proxy	Mongo Daemon
	tcp6	:::27026	docker-proxy	Mongo Daemon
	tcp6	:::27027	docker-proxy	Mongo Daemon

VM	Protocol	Local Address	Program Name	Module Name
Binding-Master	tcp6	:::27028	docker-proxy	Mongo Daemon
	tcp6	:::27029	docker-proxy	Mongo Daemon
	tcp6	:::27030	docker-proxy	Mongo Daemon
	tcp6	:::22	sshd	SSH Daemon
	tcp6	:::27031	docker-proxy	Mongo Daemon
	tcp6	:::27032	docker-proxy	Mongo Daemon
	tcp6	:::27033	docker-proxy	Mongo Daemon
	tcp6	:::27034	docker-proxy	Mongo Daemon
	tcp6	:::27035	docker-proxy	Mongo Daemon
	tcp6	:::443	docker-proxy	haproxy-common
	tcp6	:::27036	docker-proxy	Mongo Daemon
	tcp6	:::27037	docker-proxy	Mongo Daemon
	tcp6	:::27038	docker-proxy	Mongo Daemon

VM	Protocol	Local Address	Program Name	Module Name
Binding-Master	tcp6	:::27039	docker-proxy	Mongo Daemon
	tcp6	:::27040	docker-proxy	Mongo Daemon
	tcp6	:::27041	docker-proxy	Mongo Daemon
	tcp6	:::27042	docker-proxy	Mongo Daemon
	tcp6	:::27043	docker-proxy	Mongo Daemon
	tcp6	:::27044	docker-proxy	Mongo Daemon
	tcp6	:::27045	docker-proxy	Mongo Daemon
	tcp6	:::27046	docker-proxy	Mongo Daemon
	tcp6	:::2022	docker-proxy	configuration-engine
	tcp6	:::27047	docker-proxy	Mongo Daemon
	tcp6	:::5000	docker-proxy	registry
	tcp6	:::2024	docker-proxy	configuration-engine
	tcp6	:::27017	docker-proxy	Mongo Daemon
	tcp6	:::5001	docker-proxy	registry
	tcp6	:::27018	docker-proxy	Mongo Daemon

VM	Protocol	Local Address	Program Name	Module Name
Binding-Control VMs	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp6	:::27025	docker-proxy	Mongo Daemon
	tcp6	:::27026	docker-proxy	Mongo Daemon
	tcp6	:::27027	docker-proxy	Mongo Daemon
	tcp6	:::27028	docker-proxy	Mongo Daemon
	tcp6	:::27029	docker-proxy	Mongo Daemon
	tcp6	:::27030	docker-proxy	Mongo Daemon
	tcp6	:::22	sshd	SSH Daemon
	tcp6	:::27031	docker-proxy	Mongo Daemon
	tcp6	:::27032	docker-proxy	Mongo Daemon
	tcp6	:::27033	docker-proxy	Mongo Daemon
	tcp6	:::27034	docker-proxy	Mongo Daemon
Binding-Control VMs	tcp6	:::27035	docker-proxy	Mongo Daemon
	tcp6	:::443	docker-proxy	Mongo Daemon
	tcp6	:::27036	docker-proxy	Mongo Daemon
	tcp6	:::27037	docker-proxy	Mongo Daemon
	tcp6	:::27038	docker-proxy	Mongo Daemon
	tcp6	:::27039	docker-proxy	Mongo Daemon
	tcp6	:::27040	docker-proxy	Mongo Daemon
	tcp6	:::27041	docker-proxy	Mongo Daemon
	tcp6	:::27042	docker-proxy	Mongo Daemon
	tcp6	:::27043	docker-proxy	Mongo Daemon
	tcp6	:::27044	docker-proxy	Mongo Daemon
	tcp6	:::27045	docker-proxy	Mongo Daemon
	tcp6	:::27046	docker-proxy	Mongo Daemon

VM	Protocol	Local Address	Program Name	Module Name
Binding-Control VMs	tcp6	:::27047	docker-proxy	Mongo Daemon
	tcp6	:::2024	docker-proxy	configuration-engine
	tcp6	:::27017	docker-proxy	Mongo Daemon
	tcp6	:::27018	docker-proxy	Mongo Daemon
	tcp6	:::2026	docker-proxy	Stats
	tcp6	:::27019	docker-proxy	Mongo Daemon
	tcp6	:::27020	docker-proxy	Mongo Daemon
	tcp6	:::9100	node_exporter	Node Exporter
	tcp6	:::27021	docker-proxy	Mongo Daemon
	tcp6	:::9997	docker-proxy	haproxy-common
	tcp6	:::27022	docker-proxy	Mongo Daemon
	tcp6	:::9998	docker-proxy	haproxy-common
	tcp6	:::27023	docker-proxy	Mongo Daemon
	tcp6	:::9999	docker-proxy	haproxy-common
	tcp6	:::27024	docker-proxy	Mongo Daemon
Persistence-DB VMs	tcp	0.0.0.0:6783	weaver	Weave Daemon
	tcp	0.0.0.0:22	sshd	SSH Daemon
	tcp6	:::27037	docker-proxy	Mongo Daemon
	tcp6	:::27038	docker-proxy	Mongo Daemon
	tcp6	:::27039	docker-proxy	Mongo Daemon
	tcp6	:::27040	docker-proxy	Mongo Daemon
	tcp6	:::27041	docker-proxy	Mongo Daemon
	tcp6	:::27042	docker-proxy	Mongo Daemon
	tcp6	:::27043	docker-proxy	Mongo Daemon
	tcp6	:::27044	docker-proxy	Mongo Daemon
	tcp6	:::27045	docker-proxy	Mongo Daemon
	tcp6	:::27046	docker-proxy	Mongo Daemon

VM	Protocol	Local Address	Program Name	Module Name
Persistence-DB VMs	tcp6	:::27047	docker-proxy	Mongo Daemon
	tcp6	:::27017	docker-proxy	Mongo Daemon
	tcp6	:::27018	docker-proxy	Mongo Daemon
	tcp6	:::27019	docker-proxy	Mongo Daemon
	tcp6	:::27020	docker-proxy	Mongo Daemon
	tcp6	:::9100	node_exporter	Node Exporter
	tcp6	:::27021	docker-proxy	Mongo Daemon
	tcp6	:::27022	docker-proxy	Mongo Daemon
	tcp6	:::27023	docker-proxy	Mongo Daemon
	tcp6	:::27024	docker-proxy	Mongo Daemon
	tcp6	:::27025	docker-proxy	Mongo Daemon
	tcp6	:::27026	docker-proxy	Mongo Daemon
Persistence-DB VMs	tcp6	:::27027	docker-proxy	Mongo Daemon
	tcp6	:::27028	docker-proxy	Mongo Daemon
	tcp6	:::27029	docker-proxy	Mongo Daemon
	tcp6	:::27030	docker-proxy	Mongo Daemon
	tcp6	:::22	sshd	SSH Daemon
	tcp6	:::27031	docker-proxy	Mongo Daemon
	tcp6	:::27032	docker-proxy	Mongo Daemon
	tcp6	:::27033	docker-proxy	Mongo Daemon
	tcp6	:::27034	docker-proxy	Mongo Daemon
	tcp6	:::27035	docker-proxy	Mongo Daemon
	tcp6	:::27036	docker-proxy	Mongo Daemon

Listening Ports in DRA Deployment