



## **CPS vDRA Installation Guide for VMware, Release 24.2.0**

**First Published:** 2024-09-18

### **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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2024 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### PREFACE

<b>Preface</b>	<b>v</b>
About This Guide	v
Audience	v
Additional Support	vi
Conventions (all documentation)	vi
Communications, Services, and Additional Information	vii
Important Notes	viii

---

### CHAPTER 1

<b>Pre-Installation Requirements</b>	<b>1</b>
Installation Overview	1
Sample vDRA System	2
Installation Order	2
Requirements	2
VMware Interface Name and Order	3
Environment Artifacts	4

---

### CHAPTER 2

<b>Installing CPS vDRA</b>	<b>5</b>
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
CPS Installer Commands	8
Upgrading VMs using Diagnostics and Redeployment Health Check	11
Ranking Details	11

- Resume Redeployment 12
- Validate Deployment 12
  - show system status 13
  - show system diagnostics 13
  - show docker engine 13
  - show docker service 13
- Redeploy VMs during the ISSM Operation 14

---

**APPENDIX A**

- Installation Examples 17**
  - DRA-VNF Example 17
    - Artifacts Structure Example 17
    - Top Level Directory 19
    - example-dra-vnf/vms/role 21
    - Data Disk 21

---

**APPENDIX B**

- Listening Ports in DRA Deployment 25**
  - Listening Ports in DRA Deployment 25



## 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](https://www.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](mailto: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
<b>bold font</b>	Commands and keywords and user-entered text appear in <b>bold font</b> .
<i>italic font</i>	Document titles, new or emphasized terms, and arguments for which you supply values are in <i>italic font</i> .
[ ]	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 impact 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 6.7/7.0 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.

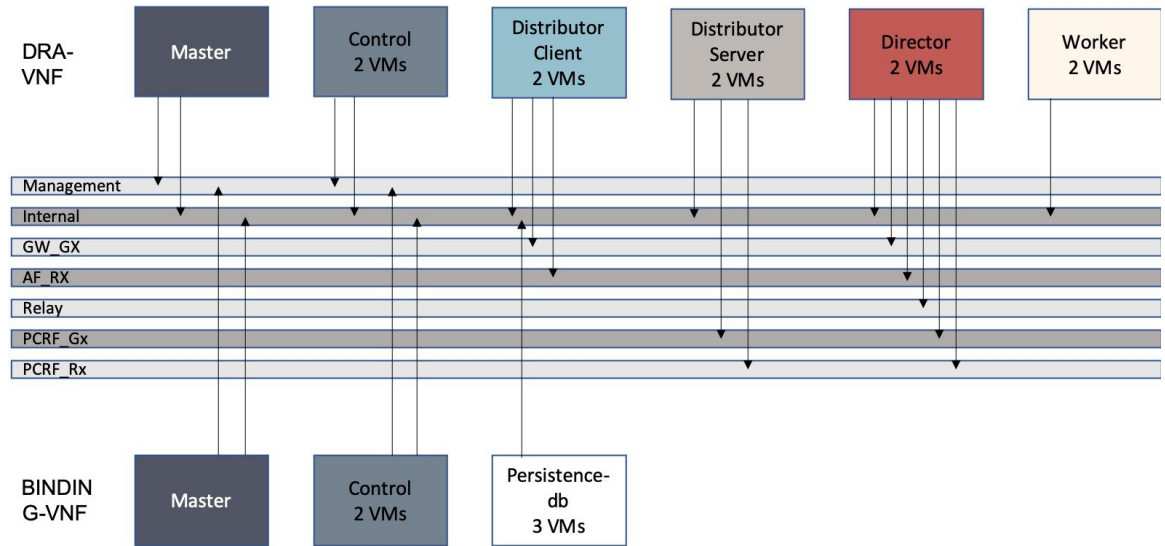
---

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

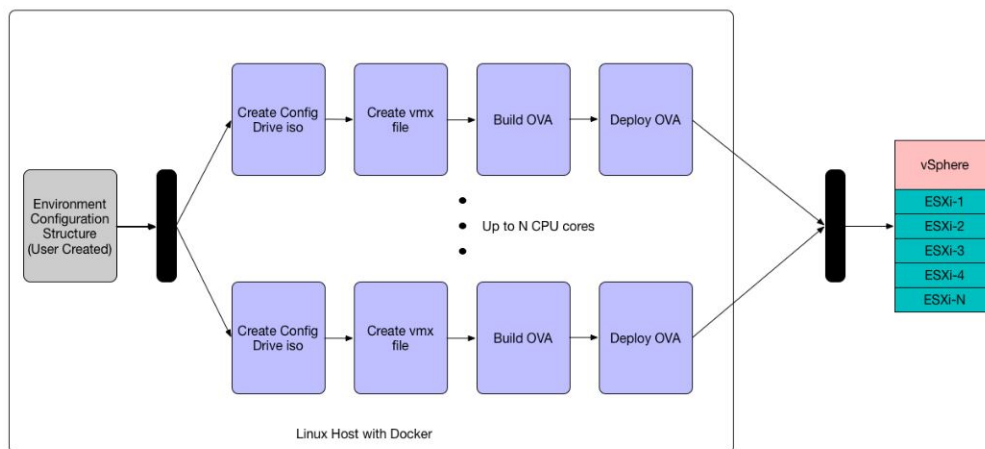
## Environment Artifacts

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](#)
- [CPS Installer Commands, on page 8](#)
- [Validate Deployment, on page 12](#)
- [Redeploy VMs during the ISSM Operation, on page 14](#)

## 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.

- 
- 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** 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 hardware** window.
- Step 10** In **Virtual Hardware** tab:
- Select 4 CPUs.
  - Select **Memory** size as **32 GB**.
  - Delete **New Hard Disk** (VM will use the existing disk created earlier with vmkfstools command).
  - Expand **New SCSI controller** and from **Change Type** drop-down list, select **VMware Paravirtual**.
  - 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.
  - 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.
  - 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*).
- Right-click and select **Edit Settings...** **Virtual Hardware** tab is displayed as default.
  - Click **ADD NEW DEVICE** and from the list select **Existing Hard Disk** to open **Select File** window.
  - Navigate to `cps-deployer-host_<version>-esxi.vmdk` file created earlier with the vmkfstools command and click **OK**.
- Step 13** Adjust hard disk size.
- Press **Ctrl + Alt +2** to go back to **Hosts and Clusters** and select the VM created above (*CPS Cluster Manager*).
  - Right-click and select **Edit Settings...** **Virtual Hardware** tab is displayed as default.
  - In the **Hard disk 1** text box enter **100** and click **OK**.
- Step 14** Power ON the VM and open the console.
-

## Configure Network

**Step 1** Log into the VM Console as user: cps, password: cisco123.

**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\$

## 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
                        Absolute 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 is 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 is 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 artifact file, 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 artifact 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 than one VM is specified, VMs are processed serially. The following command is an example for redeploying 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 than one VM is specified, VMs are processed serially. The following command is an example for redeploying 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-director1,sk-dra-director2
```

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

Ranking Details		
<b>Rank 1</b>	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.
<b>Rank 2</b>	Control VM	<ul style="list-style-type: none"> <li>• Declare the control VMs for Ranks 2 and 3 and add one or more VMs.</li> <li>• If you do not redeploy control VMs, do not declare any values in the upgradelist.txt file starting with Rank 2 and 3.</li> </ul>
<b>Rank 3</b>	Example: 2,sk-control0, sk-dra-worker2 3,sk-control1, sk-dra-worker1	
<b>Rank 4</b>	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[an-master-0]#
```

## Redeploy VMs during the ISSM Operation

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







## APPENDIX **A**

# Installation Examples

---

- [DRA-VNF Example, on page 17](#)

## 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-vmware.cisco.com"
VSPHERE_USER="administrator@vmware.local"
VSPHERE_PASSWORD="fool23"
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 AAAAB3NzaC1yc2EAAAADAQABAAQDzjJjndIvUiBta4VSId2gJm1MwcQ8wtejg
    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-init 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 ESXi 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 AAAAB3NzaC1yc2EAAAADAQABAAQDZjJjndIvUiBta4VS1bd2g
    JmlMwCQ8wtejgAbiXtoFZdtMdo9G0ZDEotxHNNDPwWujMiYakZhZWX/zON9raav
    U8lgD9+YcRopWUtujIC7lYjtoxIj 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 25](#)

### 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

<b>VM</b>	<b>Protocol</b>	<b>Local Address</b>	<b>Program Name</b>	<b>Module Name</b>
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

<b>VM</b>	<b>Protocol</b>	<b>Local Address</b>	<b>Program Name</b>	<b>Module Name</b>
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
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	

