



# Installing CPS vDRA

---

- [Create Installer VM in vSphere, on page 1](#)
- [Binding-VNF, on page 3](#)
- [cps Installer Commands, on page 4](#)
- [Validate Deployment, on page 6](#)

## 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** Log into vSphere and select **Hosts and Clusters**.

**Step 2** Select the target ESXi host.

- Step 3** Select **Actions > New Virtual Machine**.
- Step 4** Select **Create a new virtual machine** and click **Next**.
- Step 5** Enter a name for the virtual machine (for example, `cps-installer`) and select the location for the virtual machine. Click **Next**.
- Step 6** Select blade IP address from **Select a compute resource** window and click **Next** to open **Select storage** window.
- Step 7** From **Select storage** window, select `datastorename` and click **Next** to open **Select compatibility** window.
- Step 8** From **Compatible with:** drop-down list, select **ESXi 6.0 and later** and click **Next** to open **Select a guest OS** window.
- Step 9** From **Guest OS Family:** drop-down list, select **Linux** and from **Guest OS Version:** drop-down list, select **Ubuntu Linux (64-bit)**.
- Step 10** Click **Next** to open **Customize hardware** window.
- Step 11** 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).
  - Select **Management** network from the **New Network** drop-down list and check **Connect At Power On**.
- Step 12** Create hard disk.
- Select **Existing Hard Disk** from **New Device:** drop-down list and click **ADD**.
  - Navigate to `cps-deployer-host_<version>-esxi.vmdk` file created earlier with the `vmkfstools` command and click **OK**.
  - Click **Next**.
  - Click **Finish**.
- Step 13** Adjust hard disk size.
- Press **Ctrl + Alt +2** to go back to **Hosts and Clusters** and select the VM created above (`cps-installer`).
  - 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 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,powerson,datadisk}
          vnf_artifacts_relative_path [vm_name [vm_name ...]]

```

#### positional arguments:

```

{install,delete,redeploy,list,poweroff,powerson,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 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
```

### 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 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
```

## 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]#
```

show docker service