CPS vDRA Installation Guide for VMware, Release 19.4.0 (1)

First Published: 2019-07-25

Last Modified: 2019-08-30

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 NONINFRINGEMENT 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.

CONTENTS

PREFACE

CHAPTER 1

CHAPTER 2

Preface

About This Guide

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.

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.
[ ]	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 Marketplace.
- 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

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 6.5 must be installed on all the blades that are used to host the vDRA system. For more details, see http://www.vmware.com/products/esxi-and-esx/overview.html.

Installing vDRA on vSphere includes the following:

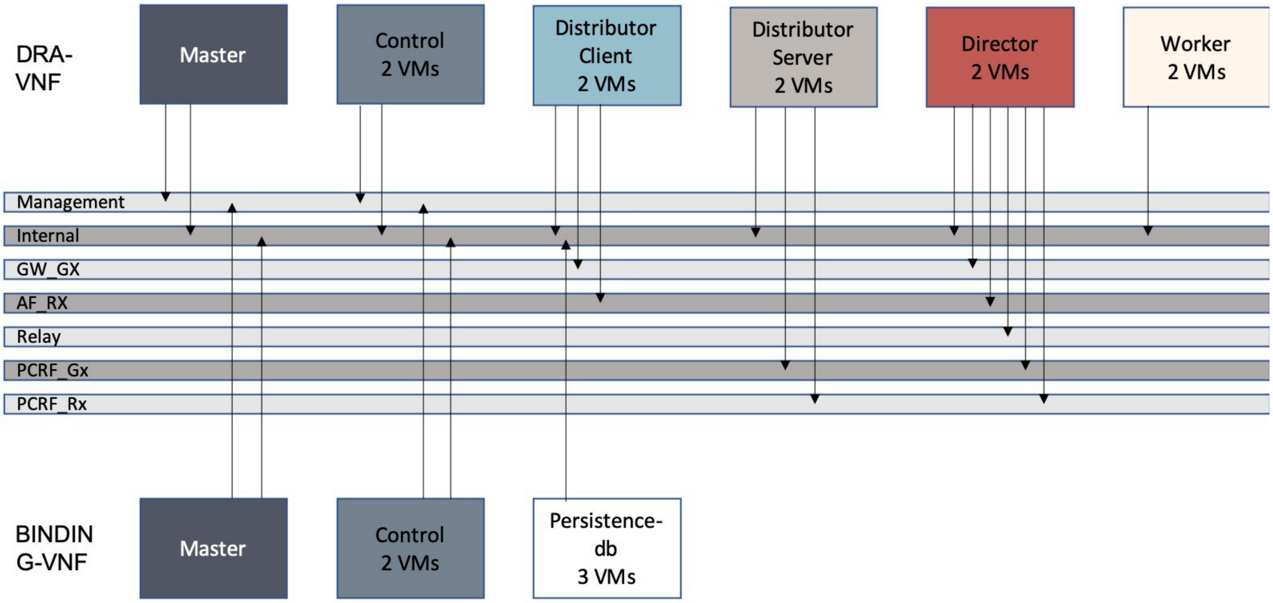
- Create a vDRA installer VM in vSphere using the vDRA Deployer Host VMDK.
- Create the artifacts that describe the VM roles, CPS ISO (dra-vnf or binding-vnf), IP addresses, hostnames, target ESXi servers, and so on.
- Run the `cps install` *`<vnf directory>`* command.

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

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 6.5

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
6	ens225

NETWORK_	Linux Interface Name
7	ens257
8	ens162
9	ens194

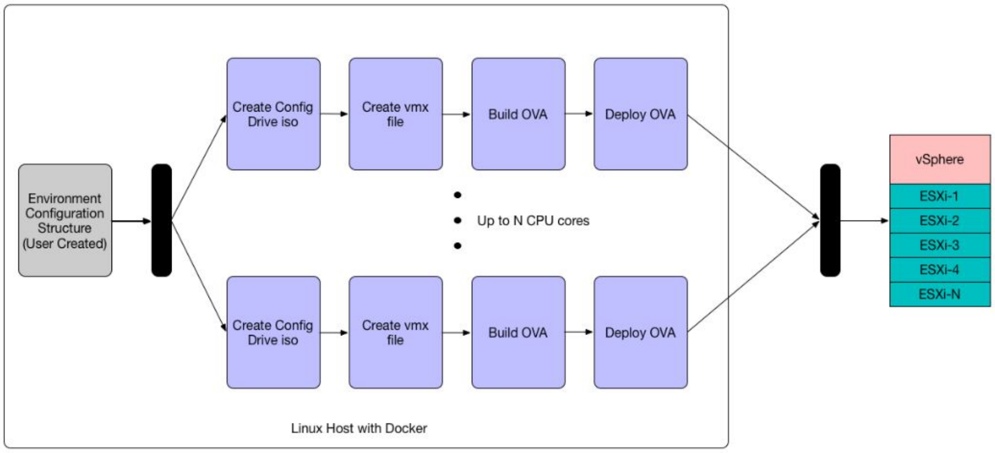
Environment Artifacts

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

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 EXSi 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:

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) Select **Management** network from the **`New Network`** drop-down list and check **Connect At Power On**.

Step 12 Create hard disk.

a) Select **Existing Hard Disk** from **New Device:** drop-down list and click **`ADD`**.

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

c) Click **Next**.

d) Click **Finish**.

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-installer*).

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

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,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
```

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
```

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]#
```

CHAPTER 3

In-Service Migration from MongoDB Sharding to Application Sharding

Prerequisites

- Existing mated pair setup on CPS 18.2.0.
  - IPv6 relay based on IPv6 address ranges is already configured to relay Rx AARs to sites D1/D2 or D3/D4 or D5/D6 based on IPv6 pools (D1, D3, D5 should be the primaries during migration).
  - Set BindingExpiryTimer to very large value (3 months) at all sites. It is recommended to be done 1 week before migration window so that the existing bindings are updated.
- On CPS 18.2.0 master/control VMs, data disks are present and /data is mounted on it.
- Application call flows are executed, and calls are running fine.
- Make sure there is no NTP and connectivity related issues across the sites.
- All the new database VMs that will be later deployed on all the sites should be accessible to all the worker VMs.
- Decide the statistics partition size as per the customer requirements.

Limitations

- During migration of a mated-pair, the GR resiliency is not fully available until both the sites are migrated. For example, if Site1 is migrated and all the new sessions for Site1 are getting stored in new application-sharded databases, then when Site1 goes down, the existing sessions from Site1 cannot be updated or deleted from Site2 (as Site2 is still using the MongoDB sharded cluster as the Primary database). This results from a change to have all the sites be able to read application-sharded databases for secondary lookups.
- DRA does not support mixed databases with different shard-type. A DRA VNF site should not be configured where session database has MongoDB based sharding and binding database has other type of sharding. The connections on all the databases (session and bindings) should be configured at the same time.
- When DRA VNF is configured with two database connections on MongoDB sharded as well as application-sharded databases, then the session/binding expiration is performed on the records in the Primary DB (default primary application-sharded DB). MongoDB sharded database can be made as primary using `dra migration enable-mongo-sharded-db-as-primary-db` command.

Create Installer VM

Step 1 Take a backup of envs from CPS 18.2.0 installer VM.

Step 2 Create CPS 19.4.0 installer VM. For more information, see the *CPS vDRA Installation Guide for VMware*.

Step 3 Copy env backup to CPS 19.4.0 VM.

Step 4 Make changes in `user_data.yml` for /stats partition for master/control VMs. For more information, see the *CPS vDRA Installation Guide for VMware*.

Step 5 Create configuration for new database VMs as follows:

a) Update `user_data.yml` for mode configuration.

```
write_files:
  - path: /root/swarm.json
    content: |
     {
        "role": "{{ ROLE }}",
        "identifier": "{{ IDENTIFIER }}",
        "master": "{{ MASTER_IP }}",
        "network": "{{ INTERNAL_NETWORK }}",
        "mode": "{{ MODE }}",
        {% if WEAVE_PASSWORD is defined %}"weavePw": "{{ WEAVE_PASSWORD }}", {% endif %}
        "zing": "{{ RUN_ZING | default(1) }}",
        "cluster_id": "{{ CLUSTER_ID }}",
        "system_id": "{{ SYSTEM_ID }}"
     }
```

b) Update `vm.env` file on the new VMs with the following configuration:

Note This configuration mode (app-based-sharding) is needed only for new database VMs and not for old VMs.

```
MODE="app-based-sharding"
```

For example,

```
cps@ubuntu:/data/deployer/envs/example-binding-vnf/vms/persistence-db/sk-persistence-db-3$ cat
vm.env
HOSTNAME=sk-persistence-db-3
FQDN=sk-persistence-db-3.local
MODE="app-based-sharding"
cps@ubuntu:/data/deployer/envs/example-binding-vnf/vms/persistence-db/sk-persistence-db-3$
```

Note

- Make sure “VMDK” field points to 19.4.0 base vmdk.
- Make sure all the artifacts are updated according to the 19.4.0 configuration requirements.

Upgrade Binding VNF

Step 1 1. Increase the size of external disk by amount required for stats partition for master/control VMs through VMware Web UI Console or vSphere client. For more information, see VMware Documentation.

Step 2 Upgrade to 19.4.0 ISO.

Step 3 Post upgrade, to login to CLI execute the following command:

```
ssh -oKexAlgorithms=+diffie-hellman-group14-sha1 -p 2024 admin@localhost
```

Step 4 If upgrade is stuck for binding VNF due to consul-binding change for router VMs, execute the following command:

```
docker logs orchestrator --since 1m | grep -A5 -B4 consul-binding | tail
Verify the exception similar to following is displayed:
WARN  [2019-07-19 09:50:29,062] AUDIT: Upgrading module consul-binding, instance 1
INFO  [2019-07-19 09:50:29,062]
com.broadhop.orchestration.engine.services.schedulers.UpgradeDowngradeScheduler: Upgrading module
consul-binding-1
WARN  [2019-07-19 09:50:29,062]
com.broadhop.orchestration.engine.services.schedulers.UpgradeDowngradeScheduler: Exception processing
 run loop
! javax.ws.rs.WebApplicationException: Service specified does not exist - consul-6
! at
com.broadhop.orchestration.engine.services.api.BaseModuleServices.buildModule(BaseModuleServices.java:115)
! at
com.broadhop.orchestration.engine.services.api.BaseModuleServices.buildUpdatedModule(BaseModuleServices.java:265)
! at
com.broadhop.orchestration.engine.services.api.BaseModuleServices.updateModule(BaseModuleServices.java:242)
cps@sk-master-binding-0:~$
```

a) Login to orchestrator container.

```
docker exec -it orchestrator bash
mongo
use orchestration
db.services.remove({"module.name": "consul-binding"})
db.modules.remove({"module.name": "consul-binding"})
```

Step 5 Verify the upgrade has continued using show system status and show scheduling status and wait till the upgrade is complete.

Step 6 Verify that there are no errors in diagnostics.

Step 7 Redeploy database and router VMs from 19.4.0 installer (Make sure 19.4.0 artifacts are updated properly for latest base vmdk and other changes). For more information, see the *CPS vDRA Installation Guide for VMware*.

Step 8 Verify that the system status is 100% and no errors are reported in diagnostics.

Step 9 Stop consul on control -1 VM.

```
docker connect consul-3
 supervisorctl stop all
```

Step 10 Redeploy control-1 VM from 19.4.0 installer. For more information, see the *CPS vDRA Installation Guide for VMware*.

Once control-1 VM is up, refer to Create New Partition for /stats, on page 17 to confirm the stats partition is not created.

Step 11 Stop consul on control-0 VM.

```
docker connect consul-2
supervisorctl stop all
```

Step 12 Redeploy control-0 VM. For more information, see the *CPS vDRA Installation Guide for VMware*.

Once control-0 VM is up, refer to Create New Partition for /stats, on page 17 to confirm the stats partition is not created.

Step 13 Stop consul on master VM.

```
docker connect consul-1
supervisorctl stop all
```

Step 14 Redeploy master VM. For more information, see the *CPS vDRA Installation Guide for VMware*.

Once master VM is up, refer to Create New Partition for /stats, on page 17 to confirm the stats partition is not created.

Step 15 Create new database VMs. For more information, refer to Create New Database VMs, on page 23.

Step 16 Apply database configuration for application based MongoDB cluster. For more information, see the *CPS vDRA Configuration Guide*.

Step 17 Post upgrade, to see old statistics, you need to update Grafana datasource to view old statistics. For more information, refer to Loading Old Statistics in the System, on page 21.

Upgrade DRA VNF

Step 1 1. Increase the size of external disk by amount required for stats partition for master/control VMs through VMware Web UI Console or vSphere client. For more information, see VMware Documentation.

Step 2 Load 19.4.0 ISO using `system software iso load` command.

For example, `scheduler# system software iso load category product file cisco-policy-dra.iso activate true`

Step 3 Activate 19.4.0 ISO using `system software iso activate` command.

For example,

```
scheduler# system software iso activate category product name cisco-policy-dra version 12.9.9
qualifier dra.2017-05-17.441.6968d89
```

Step 4 Stop the system by executing `system stop` command.

Step 5 Execute upgrade ISO using `system upgrade` command.

Only orchestrator is updated and system status is 0% as system is in stopped state.

Step 6 Post upgrade, to login to CLI execute the following command.

```
ssh -oKexAlgorithms=+diffie-hellman-group14-sha1 -p 2024 admin@localhost
```

Step 7 Stop consul servers on master/control VMs using CLI:

```
docker connect consul-(1/2/3)
supervisorctl stop all
```

Step 8 Delete all DRA VNF VMs using 19.4.0 installer.

Step 9 Make sure 19.4.0 artifacts are updated.

Step 10 Execute `system start` command from CLI.

Step 11 Create DRA VNF using `cps install` *`<vnf-config>`* command.

Step 12 Verify if `/stats` partition is created properly. Refer to Create New Partition for /stats, on page 17 to confirm the stats partition is created.

Step 13 Post upgrade, to see old statistics, you need to update Grafana datasource to view old statistics. For more infor-mation, refer to Upgrade Binding VNF, on page 15.

Step 14 Enable the connection to new application-sharded databases but still have the MongoDB sharded database as primary database. For more information, refer to Enable Connection to Application-sharded Database Cluster, on page 24.

Create New Partition for /stats

Note In 18.2.0, statistics were getting stored in /data directory. In 19.4.0, statistics directory is changed to /stats and it is a new partition on external disk.

Create a new partition before upgrade to CPS 19.4.0.

In 19.4.0, `/data` size is reduced to 70 GB. 18.2.0 has 100 GB for `/data`. With the following steps, only additional `/stats` partition is created. `/data` directory partition is not modified.

The following must be executed on control-1, control-0 and master VMs one by one.

Note Placement of `config_servers` on master/control VMs before migration requires VM reboot. If a VM is hosting primary, then move primary to different VM.

For certain statistics such as, CPU statistics, plugin in node exporter is changed hence after backup and restore in 19.4.0, statistics are present but not visible in Grafana as query is changed.

For example, for CPU idle statistics query in 18.2.0 is:

```
100 - (avg(irate(node_cpu{mode="idle"}[5m])) by (instance) * 100)
```

In 19.4.0, query is changed to:

```
100 - (avg(irate(node_cpu_seconds_total{mode="idle"}[5m])) by (instance) * 100)
```

Hence, to display the old statics, you should use 18.2.0 query.

Note This is applicable to fewer stats and not for all statistics.

Verify Partition for /stats is Created

Execute `lsblk` command and check if the two partitions are visible for external disk.

For example,

```
root@site1-dra-control0:~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0      2:0    1    4K  0 disk
sda      8:0    0 39.1G  0 disk
|-sda1   8:1    0  190M  0 part /boot
|-sda2   8:2    0 1022M  0 part [SWAP]
`-sda3   8:3    0 37.9G  0 part /
sdb      8:16   0   20G  0 disk
|-sdb1   8:17   0   15G  0 part /data
`-sdb2   8:18   0    5G  0 part
sr0     11:0    1  364K  0 rom
```

If the second partition is not visible, execute the following steps:

Step 1 Verify if increase in disk size is seen using `lsblk` command.

```
root@sk-master-binding-0:~# lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
fd0      2:0    1     4K  0 disk
sda      8:0    0  99.1G  0 disk
|-sda1   8:1    0   190M  0 part /boot
|-sda2   8:2    0  1022M  0 part [SWAP]
`-sda3   8:3    0  97.9G  0 part /
sdb      8:16   0    20G  0 disk
`-sdb1   8:17   0    15G  0 part /data
sr0     11:0    1   364K  0 rom
sr1     11:1    1 650.2M  0 rom  /mnt/iso
root@sk-master-binding-0:~#
```

Step 2 Execute the following command:

`sgdisk -p <device_name>`

where, <*device_name*> is disk on which `/data` is mounted and can be seen in `lsblk` command.

Note the end sector of data partition. There should be only one partition with name `/data`.

For example,

```
root@sk-master-binding-0:~# sgdisk -p /dev/sdb
Disk /dev/sdb: 41943040 sectors, 20.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): F4FBFF33-D16C-4B8F-8347-A4481D5C627C
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 31457246
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
```

```
   1            2048        31457246   15.0 GiB    8300
root@sk-master-binding-0:~#
```

Step 3 Take the backup of existing partition.

```
sgdisk -b=/root/bkup_partition.bin <device_name>
```

For example,

```
root@sk-master-binding-0:~# sgdisk -b=/root/bkup_partition.bin  /dev/sdb
The operation has completed successfully.
root@sk-master-binding-0:~# file bkup_partition.bin
bkup_partition.bin: DOS/MBR boot sector; partition 1 : ID=0xee, start-CHS (0x0,0,2), end-CHS
(0x3ff,255,63), startsector 1, 31457279 sectors, extended partition table (last)
root@sk-master-binding-0:~#
```

Step 4 Move the second header to the end of the disk.

```
sgdisk -e <device_name>
```

For example,

```
root@sk-master-binding-0:~# sgdisk -e /dev/sdb
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot or after you
run (8) or kpartx(8)
The operation has completed successfully.
root@sk-master-binding-0:~#
```

Step 5 Find the end sector name.

```
sgdisk -E <device_name>
```

For example,

```
root@sk-master-binding-0:~# sgdisk -E /dev/sdb
41943006
root@sk-master-binding-0:~#
```

Step 6 Assuming only one partition is present hence, you need to create the partition with number 2.

`sgdisk -n 2:`*`<start_sector>`*`:`*`<end_sector> <device_name>`*

<*start_sector*> is the end_sector of first partition +1.

<*end_sector*> is the output of the command `sgdisk -e <device_name>` in Step 5, on page 19.

For example,

```
root@sk-master-binding-0:~# sgdisk -n 2:31457247:41943006 /dev/sdb
Information: Moved requested sector from 31457247 to 31457280 in
order to align on 2048-sector boundaries.
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot or after you
run partprobe(8) or kpartx(8)
The operation has completed successfully.
root@sk-master-binding-0:~#
root@sk-master-binding-0:~# sgdisk -p /dev/sdb
Disk /dev/sdb: 41943040 sectors, 20.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): F4FBFF33-D16C-4B8F-8347-A4481D5C627C
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 41943006
Partitions will be aligned on 2048-sector boundaries
Total free space is 2047 sectors (1023.5 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        31457246   15.0 GiB    8300
   2        31457280        41943006   5.0 GiB     8300
root@sk-master-binding-0:~#
```

Step 7 Stop consul service running on that VM and then reboot the VM.

For example, on master VM, consul-1 is service.

```
docker exec -it consul-1 bash
supervisorctl stop all
```

Step 8 Wait till the system status shows 100% in CLI.

Verify /stats is Mounted Correctly

Execute `lsblk` command to check whether the /stats is mounted correctly.

Here is a sample configuration:

```
cps@site1-dra-master0:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0      2:0    1    4K  0 disk
sda      8:0    0 39.1G  0 disk
|-sda1   8:1    0  190M  0 part /boot
|-sda2   8:2    0 1022M  0 part [SWAP]
`-sda3   8:3    0 37.9G  0 part /
sdb      8:16   0   20G  0 disk
|-sdb1   8:17   0   15G  0 part /data
`-sdb2   8:18   0    5G  0 part /stats
sr0     11:0    1  364K  0 rom
sr1     11:1    1  1.7G  0 rom
cps@site1-dra-master0:~$
```

If the `/stats` is not mounted correctly, execute the following commands as a root user:

Step 1 Create director for storing statistics.

```
mkdir -p /stats
chmod -R 0755 /stats
```

Step 2 Create the label for file system.

```
/sbin/mkfs.ext4 /dev/sdb2 -L 'STATS'
```

Step 3 Update `/etc/fstab` to save the changes.

```
'LABEL=STATS /stats ext4 defaults,nofail,comment=cloudconfig 0 2'
```

Step 4 Mount the file system.

```
mount -a
```

Step 5 Verify the changes using `lsblk` command.

```
root@sk-master-binding-0:~# lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
fd0      2:0    1      4K  0 disk
sda      8:0    0  99.1G  0 disk
|-sda1   8:1    0   190M  0 part /boot
|-sda2   8:2    0  1022M  0 part [SWAP]
`-sda3   8:3    0  97.9G  0 part /
sdb      8:16   0    20G  0 disk
|-sdb1   8:17   0    15G  0 part /data
```

```
`-sdb2   8:18   0     5G  0 part /stats
sr0     11:0    1   364K  0 rom
sr1     11:1    1 650.2M  0 rom
root@sk-master-binding-0:~#
```

Note Wait till the system is up and running to 100% status. Verify that there are no errors in diagnostics.

What to do next

Verify Prometheus Statistics is Present in /stats Directory

To verify if Prometheus directories are present in `/stats` directory, execute the following command:

```
cps@site2-binding-master-0:~$ ls /stats/
bulkstats/         prometheus/        prometheus-hi-res/   prometheus-planning/ prometheus-trending/
```

a) If the directories are not created, restart the containers:

```
prometheus-hi-res-<id>
prometheus-planning-<id>
prometheus-trending-<id>
```

where, <*id*> is the container ID.

Loading Old Statistics in the System

Post upgrade, to see old statistics, you need to update Grafana datasource to view old statistics. Use the following steps:

a) Login to Grafana container from CLI.

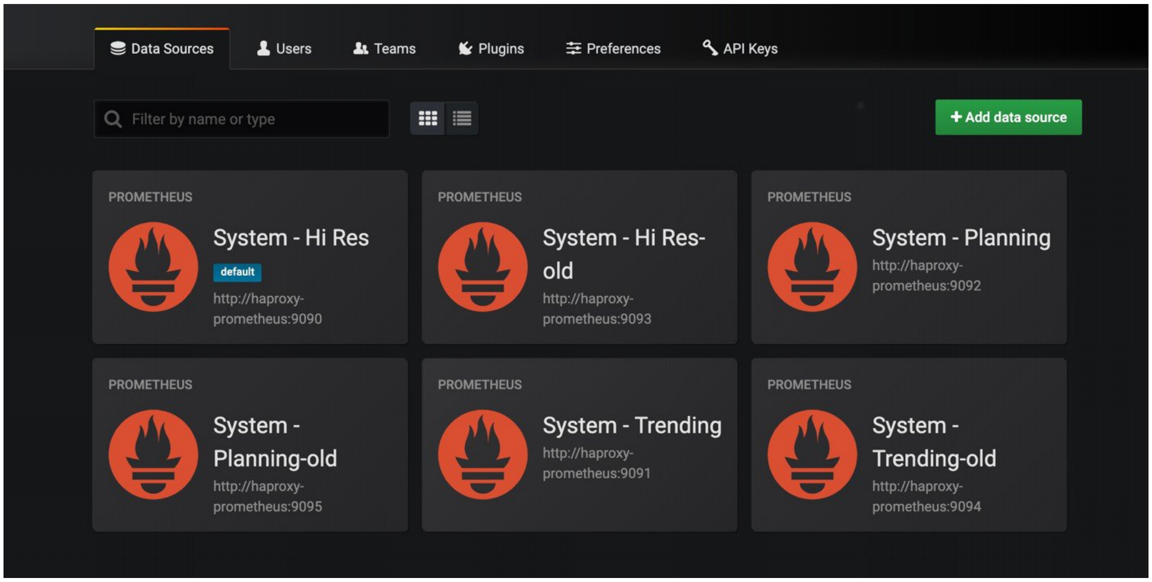
```
admin@orchestrator[sk-master-binding-0]# docker connect grafana
```

b) Execute the following commands in Grafana container shell.

```
rm -f /var/broadhop/grafana/data/grafana.db
touch /var/broadhop/grafana/data/grafana.db
sqlite3 /var/broadhop/grafana/data/grafana.db < /var/tmp/grafana.sql
supervisorctl restart grafana
```

c) In Grafana, check whether you can see the old datasources along with new datasources.

Figure 3: Datasources



PROMETHEUS
System - Hi Res
http://haproxy-prometheus:9090

PROMETHEUS
System - Hi Res-old
http://haproxy-prometheus:9093

PROMETHEUS
System - Planning
http://haproxy-prometheus:9092

PROMETHEUS
System - Planning-old
http://haproxy-prometheus:9095

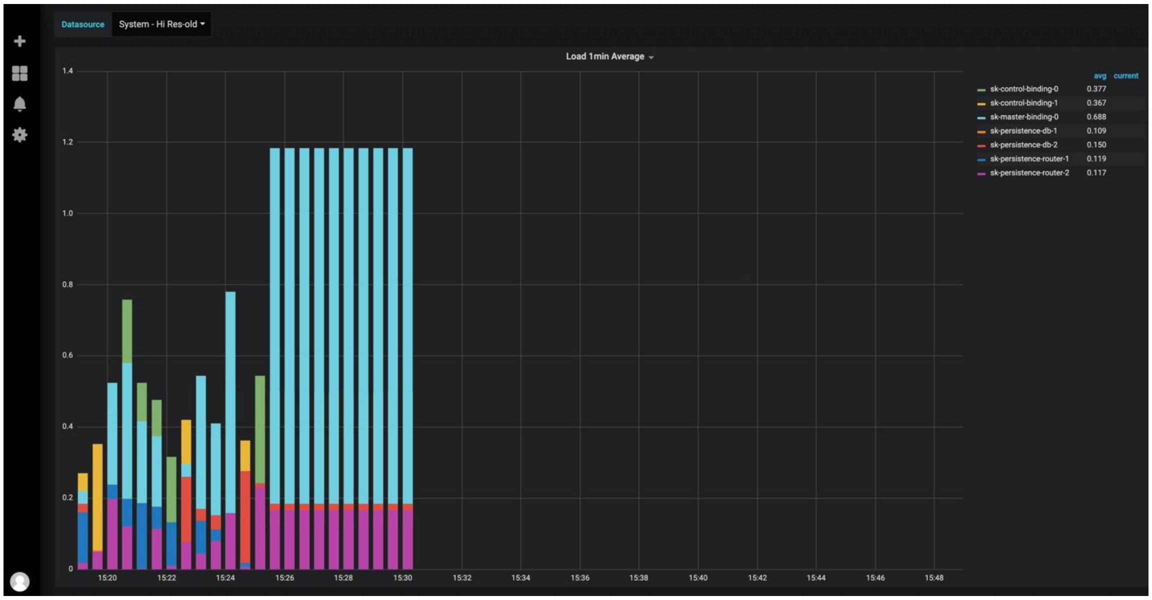
PROMETHEUS
System - Trending
http://haproxy-prometheus:9091

PROMETHEUS
System - Trending-old
http://haproxy-prometheus:9094

d) Click on datasource and update datasource without any modification.

e) Old statistics can be viewed by selecting appropriate datasource. The following is an example datasource.

Figure 4: Example Datasource


Datasource
System - Hi Res-old
Load 1min Average
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
15:20
15:22
15:24
15:26
15:28
15:30
15:32
15:34
15:36
15:38
15:40
15:42
15:44
15:46
15:48
avg
current
sk-control-binding-0 0.377
sk-control-binding-1 0.367
sk-master-binding-0 0.688
sk-persistence-db-1 0.109
sk-persistence-db-2 0.150
sk-persistence-router-1 0.119
sk-persistence-router-2 0.117

For system summary dashboard, there is a query change for some statistics. Hence, when viewing old statistics, you need to create new dashboard in Grafana for the statistics. The following table list some example queries:

Table 3: Query Statistics - Examples

Statistics	Old Query	New Query
Cpu stats/idle	"100 - (avg(irate(node_cpu{mode=\"idle\"} [5m])) by (instance) * 100)"	"100 - (avg(irate(node_cpu_seconds_ total{mode=\"idle\"} [5m])) by (instance) * 100)"
Cpu stats/interrupt	"avg(irate(node_cpu{mode= ~\"irq|softirq\"} [5m])) by (instance) * 100"	"avg(irate(node_cpu_seconds_total {mode=~\"irq|softirq\"}[5m])) by (instance) * 100"
Cpu stats/iowait	"avg(irate(node_cpu{mode= ~\"iowait\"} [5m])) by (instance) * 100"	"avg(irate(node_cpu_seconds_total {mode=~\"iowait\"} [5m])) by (instance) * 100"
Disk read	"sum(irate(node_disk_read_time_ms [5m])) by (instance)"	"sum(irate(node_disk_read_ time_seconds_total [5m])) by (instance)"
Disk write	"sum(irate(node_disk_write_time_ms [5m])) by (instance)"	"sum(irate(node_disk_write_ time_seconds_total [5m])) by (instance)"
Disk bytes read	"sum(irate(node_disk_bytes_read [5m])) by (instance)"	"sum(irate(node_disk_ read_bytes_total [5m])) by (instance)"
Disk bytes write	"sum(irate(node_disk_bytes_written [5m])) by (instance)"	"sum(irate(node_disk_ written_bytes_total [5m])) by (instance)"

Create New Database VMs

For migration, new VMs need to be created with role as persistence-db and additional configuration in swarm.json file as - "mode": "app-based-sharding".

Step 1 Make sure all the configurations are completed as per Create Installer VM, on page 14.

Step 2 Create new VMs.

```
cps install <binding-vnf_config> <db1> <db2>
```

For example, `cps install example-binding-vnf sk-persistence-db-3 sk-persistence-db-4`

Step 3 Add the required configuration in `swarm.json` file.

Sample configuration:

```
root@site2-persistence-db-3:~# cat swarm.json
{
   "role": "persistence-db",
   "identifier": "",
   "master": "192.168.31.81",
   "network": "192.168.31.0/24",
   "mode": "app-based-sharding",
   "weavePw": "cisco123",
   "zing": "1",
   "cluster_id": "site2-cluster",
   "system_id": "site2-system"
}
```

Create Application based Sharding MongoDB Cluster

Step 1 Create database configurations for application based cluster. For more information, see the *CPS vDRA Configuration Guide*.

Step 2 In case of Site1, when Site2 VMs are not yet up, make sure seed is created for shard from Site1.

Step 3 Sharding database replica-set members should be deployed only on the new database VMs (Master and Control VMs cannot be used as they are still using Mongo 3.4.5).

Step 4 Sharding database replica-set can have members from both sites.

What to do next

For more information, see *Configuring Application based Sharding* and *Configuring MongoDB Authentication* sections in the *CPS vDRA Configuration Guide*.

Enable Connection to Application-sharded Database Cluster

Use the following steps to enable the application-sharded databases in DRA VNF and use them as secondary/backup DB for binding lookup:

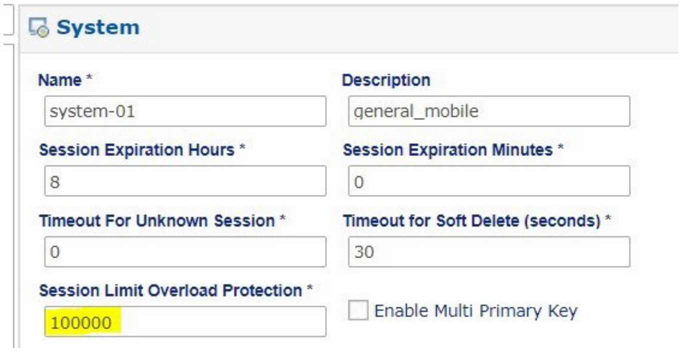
Note The commands must be entered in the order mentioned below to make sure that the MongoDB sharded database is considered primary database. By default, 19.4.0 considers application-sharded database as primary database when database connection with sharded databases is enabled.

Before you begin

Before enabling traffic to upgraded 19.4.0, make sure the following

- Configure Policy Builder for CPS session limit overload protection. (This is not a DRA specific parameter and is part of CPS core session handling. If this is not configured then it will report error log to configure this field).

Figure 5: Session Limit Overload Protection



System

Name * system-01

Description general_mobile

Session Expiration Hours * 8

Session Expiration Minutes * 0

Timeout For Unknown Session * 0

Timeout for Soft Delete (seconds) * 30

Session Limit Overload Protection * 100000

□ Enable Multi Primary Key

- For the newly configured database clusters with application-sharding make sure that all the shards are up. The ‘show database status’ should display the correct status (PRIMARY / SECONDARY / ARBITER) for all the shard members for this site.
- Any members which belong to the other mated site should also show correct status if it is already upgraded. If the mate site is not migrated, it should show ‘NO_CONNECTION’

Step 1 After Site-1 DRA VNF is upgraded, set the configuration.

```
dra migration enable-migration true
dra migration enable-mongo-sharded-db-as-primary-db true
commit
```

Step 2 Apply the binding shard-metadata-db-connection for all the required database to initiate connections with application-sharded database.

```
binding shard-metadata-db-connection session <ip> <port>
binding shard-metadata-db-connection ipv6 <ip> <port>
..
commit
```

Step 3 Make sure that the traffic related errors are not seen in logs and Grafana graphs are also clean.

Divert Traffic to Application-sharded Database Cluster

Use the following steps to start using application-sharded database as primary database.

Before you begin

Before diverting traffic, make sure the following:

- Enable application-sharded binding databases and use them as backup DBs for ALL SITES before enabling the application-sharded cluster as primary database. If this is not done, then VoLTE calls may fail.
  - All the sites in the network should be upgraded to 19.4.0 and have migration enabled with mongo-sharded-db as the primary database.
- If BindingExpiryTime was set to 3 months for migration then revert this expiry-time values so that all the bindings getting stored in application-sharded cluster expire as per the desired timeout values.

Apply the new migrated configurations to start using application-sharded databases.

```
no dra migration enable-mongo-sharded-db-as-primary-db
dra migration enable-skipping-probe-message-binding-lookup
commit
```

The traffic switches to start using the new application-sharded cluster while MongoDB-sharded cluster is used as secondary database for binding lookups only.

Note If session overload protection is required, use the following commands:

```
binding db-max-record-limit drasession <record-limit>
binding db-max-record-limit ipv6 <record-limit>
```

Verification Steps

Step 1 Verify there are no error timeouts in Grafana.

Step 2 Verify if all the statistics are seen properly.

Step 3 Using CLI, verify that all the services are healthy.

Step 4 Verify that the database status is good for all the clusters.

Step 5 Verify there are no exceptions/errors in orchestrator.

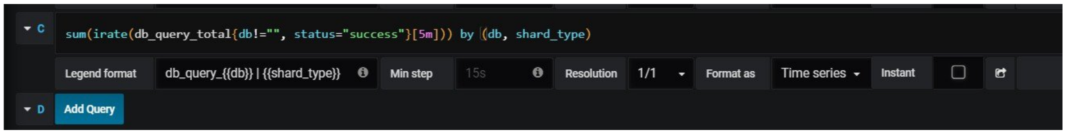
Step 6 If traffic is enabled from a PCEF peer towards the site, use the following data points to confirm which database cluster is getting used.

a) Open the ‘Database Monitoring’ Grafana dashboard for Binding VNF. This should now show the new database-cluster in the ‘Cluster’ drop-down box at the top. Select the correct ‘cluster’ specific to application-sharded database. Confirm that the database update/delete/query operations are getting processed by this database.

b) Select the cluster specific to MongoDB sharded databases and if all the sites specific to this database are migrated then it should show only ‘query’ operations. If the mated site is yet to be upgraded, then it displays update/delete operations as well.

c) In DRA VNF ‘Application Summary’ dashboard, under the ‘Success Query’ graph add the following new metric which shows the binding lookup queries going to specific databases.

Figure 6: Binding Lookup Queries

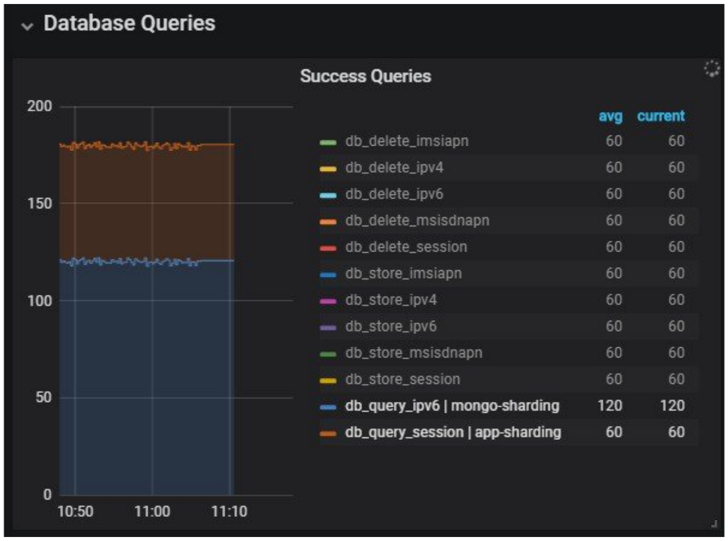


```
C  sum(irate(db_query_total{db!="", status="success"}[5m])) by (db, shard_type)
```

Legend format db_query_{{db}} | {{shard_type}} Min step 15s Resolution 1/1 Format as Time series Instant □

For example, the following sample screen shot shows IPv6 binding looks are happening on MongoDB sharded database while session queries are happening on the application-sharded database.

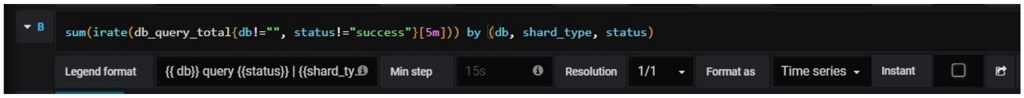
Figure 7: Sample



	avg	current
db_delete_imsiapn	60	60
db_delete_ipv4	60	60
db_delete_ipv6	60	60
db_delete_msisdnapn	60	60
db_delete_session	60	60
db_store_imsiapn	60	60
db_store_ipv4	60	60
db_store_ipv6	60	60
db_store_msisdnapn	60	60
db_store_session	60	60
db_query_ipv6 | mongo-sharding	120	120
db_query_session | app-sharding	60	60

d) Similarly, you can add new metrics in ‘Failed Queries’ graph.

Figure 8: Failed Queries - 1

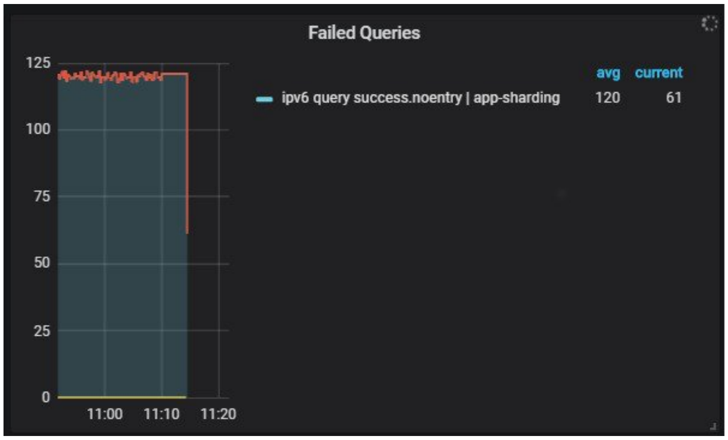


```
B  sum(irate(db_query_total{db!="", status!="success"}[5m])) by (db, shard_type, status)
```

Legend format {{ db}} query {{status}} | {{shard_ty Min step 15s Resolution 1/1 Format as Time series Instant □

For example, the following sample screen-shot for failed queries graph is showing those queries that were made to the app-sharded database but binding record was not found (noentry).

Figure 9: Failed Queries - 2



Rollback Steps

Once DRA and Binding VNFs are upgraded to 19.4.0, if any issue is encountered during the migration then you should use 19.4.0 build and continue with the existing MongoDB sharded databases (on Mongo 3.4.5).

If rollback is needed, you need to divert traffic for MongoDB based cluster. To divert the traffic, perform the following steps:

Step 1 Enable the migration with MongoDB based sharding as the primary database.

```
dra migration enable-mongo-sharded-db-as-primary-db true
```

Step 2 Disable the database connections to application-sharded databases.

```
no binding shard-metadata-db-connection
```

If required, rolling restart of binding containers can be performed.

MongoDB based Cluster Cleanup Post Migration

DRA VNF

Step 1 Login to CLI.

Step 2 Go to config mode and execute the following commands:

```
no dra migration
no binding db-connection
commit
```

For example,

```
admin@orchestrator[site1-dra-master0](config)# no dra migration
admin@orchestrator[site1-dra-master0](config)# commit
Commit complete.
admin@orchestrator[site1-dra-master0](config)#

admin@orchestrator[site1-dra-master0](config)# no binding db-connection
admin@orchestrator[site1-dra-master0](config)# commit
Commit complete.
admin@orchestrator[site1-dra-master0](config)#
```

Binding VNF

Step 1 Delete MongoDB based database cluster.

a) Login to CLI, go to config mode and execute the following command:

```
nacm rule-list any-group rule data-base access-operations delete action permit
commit
```

b) Exit from CLI and relogin.

Step 2 Delete MongoDB based sharding cluster.

a) Login to CLI, go to config mode and execute the following command:

```
admin@orchestrator[site1-binding-master-0](config)# no database cluster binding-db
admin@orchestrator[site1-binding-master-0](config)# no database cli
                                                                ^
% Invalid input detected at '^' marker.
admin@orchestrator[site1-binding-master-0](config)# no database cluster session-ipv6-db
admin@orchestrator[site1-binding-master-0](config)# commit
Commit complete.
admin@orchestrator[site1-binding-master-0](config)#
```

Step 3 Delete the old database VMs, router VMs using 19.4.0 installer.

For example,

```
cps delete example-binding-vnf-site1  site1-persistence-db-1 site1-persistence-db-2
--artifacts_abs_root_path /data/deployer/envs/site1
cps delete example-binding-vnf-site1 site1-persistence-router-1 site1-persistence-router-2
--artifacts_abs_root_path /data/deployer/envs/site1
```

Step 4 Remove the docker engine configuration using CLI for old db/router VMs.

```
admin@orchestrator[site1-binding-master-0](config)# no docker engine site1-persistence-db-1
admin@orchestrator[site1-binding-master-0](config)# no docker engine site1-persistence-db-2
admin@orchestrator[site1-binding-master-0](config)# no docker engine site1-persistence-router-1
admin@orchestrator[site1-binding-master-0](config)# no docker engine site1-persistence-router-2
admin@orchestrator[site1-binding-master-0](config)# commit
Commit complete.
```

Step 5 Repeat the following steps for master/control VMs.

a) Login to CLI and execute show running-config command and note the VM’s name.

```
show running-config docker | tab
```

b) Get mongo-s service ID for a VM using show docker service command.

```
show docker service | tab | include mongo-s | include <vm_name>
```

For example,

```
show docker service | tab | include mongo-s | include site2-binding-master-0
```

```
admin@orchestrator[site2-binding-master-0]# show docker service | tab | include mongo-s | include
 site2-binding-master-0
mongo-node          101       mongo                  3.4.5.5
site2-binding-master-0      mongo-s101                  HEALTHY  false    -
mongo-status        101       mongo-status           19.4.1-2019-08-02.7357.0e6dabc
site2-binding-master-0      mongo-status-s101           HEALTHY  false    -
admin@orchestrator[site2-binding-master-0]#
```

Note down the service ID. For example, 101 is ID for master VM for mongo module.

Step 6 Stop the orchestrator on master VM.

```
supervisorctl stop orchestration-engine
```

Step 7 Login to orchestrator container on master VM and remove modules of MongoDB node for master/control VMs.

```
docker exec -it orchestrator bash
mongo
use orchestration
db.modules.remove({"_id" : "mongo-node<id>"})
For example,
orchestrator:PRIMARY> db.modules.remove({"_id" : "mongo-node101"})
WriteResult({ "nRemoved" : 1 })
orchestrator:PRIMARY>
```

Step 8 Start the orchestrator on master VM.

```
supervisorctl start orchestration-engine
```

Step 9 Login to CLI and restart mongo-s containers on master/control VMs.

```
docker restart container-id mongo-s<ID>
```

For example,

```
docker restart container-id mongo-s101
```

Step 10 Verify using `show docker service | tab | include mongo-s` if all the mongo-s versions are latest.

APPENDIX A

Installation Examples

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 ]
```