



Install

Data Center Automation Premium 2017.05

Document Release Date: May 2017

Software Release Date: May 2017



This document is an export from the HPE Software Documentation Portal. For the latest documentation, refer <https://docs.software.hpe.com>.

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2017 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

AMD is a trademark of Advanced Micro Devices, Inc.

Intel® and Intel® Itanium® are trademarks of Intel Corporation in the U.S. and other countries.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft®, Windows®, and Windows Server® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <https://softwaresupport.hpe.com/>. You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support web site at: <https://softwaresupport.hpe.com/>. Most of the support areas require that you register as an HPE Passport user to sign in. Many also require a support contract. To register for an HPE Passport ID, click Register on HPE Support site or click Create an Account on the HPE Passport login page.

Table of Contents

1	Legal Notices	2
2	Install	6
3	Decide on your deployment model.....	6
3.1	Related topics	7
3.2	Support matrix	7
3.2.1	Hardware	8
3.2.2	Operating systems	8
3.2.3	Supported Docker images	8
3.2.4	High-Availability products	9
3.2.5	Web browsers	9
3.2.6	Additional requirements.....	9
3.2.6.1	Languages	9
3.2.6.2	Integrations	9
3.2.7	Managed devices, systems, and applications	9
3.2.8	Performance and sizing	10
3.2.9	Tuning.....	11
3.2.9.1	Related topics	11
4	Installation checklist.....	11
5	Enable your Docker Hub account	12
5.1	Where to go from here	13
6	Prerequisites	13
6.1	System requirements.....	13
6.1.1	DCA server requirements	13
6.1.1.1	Configure Chrony	15
6.1.2	Disk setup (optional).....	16
6.1.3	ITOM CDF license.....	17
6.1.3.1	Where to go from here	17
7	Install and configure prerequisite components.....	17

7.1	Where to go from here	19
7.2	Related topics	19
8	Install ITOM CDF	19
8.1	Where to go from here	19
8.2	Related topics	19
8.3	Download, verify, and unzip the contents of the ITOM CDF installation package	20
8.3.1	Where to go from here	21
8.3.2	Related topic	21
8.4	Create an NFS exported path	22
8.4.1	Use one of the nodes as the NFS server	22
8.4.1.1	Where to go from here	23
8.4.1.2	Related topics	23
8.5	Configure the install.properties file	23
8.5.1	Where to go from here	29
8.5.2	Related topics	29
8.6	Modes of deployment	29
8.6.1	Where to go from here	30
8.6.2	Related topics	30
8.6.3	Install ITOM CDF on the master node or on a single server	30
8.6.3.1	Installed directories and files	31
8.6.4	Install ITOM CDF on a worker node	34
8.6.4.1	Where to go from here	34
9	Install DCA	34
9.1	Ensure that the master node has access to Docker Hub	35
9.2	Prepare DCA images	35
9.3	Configure the NFS server share	36
9.4	Configure ChatOps	37
9.4.1	Create a Slack team	37
9.4.2	Create a private Slack app	38
9.4.3	Enable incoming web hook integration.....	39
9.4.4	Authorize the DCA bot with your Slack team	40
9.5	Install DCA	41

9.6	Verify the installation.....	44
9.6.1	Verify the deployment using the ITOM CDF interface	44
9.6.2	Verify the deployment using the command prompt.....	44
9.6.3	Using the DCA interface	44
9.6.3.1	Where to go from here	45
9.6.3.2	Related topics	45
10	Post installation tasks	45
10.1	Configure SA	45
10.1.1	Prerequisites	45
10.1.2	Post SA configuration	55
10.2	Update the default UCMDB password	56
10.3	Copy database software binaries.....	56
10.3.1	Software and patch binaries	56
10.3.1.1	Oracle 11g.....	56
10.3.1.2	Oracle 12c.....	56
10.3.1.3	Microsoft SQL Server 2008.....	57
10.3.1.4	Microsoft SQL Server 2012.....	57
10.3.1.5	Microsoft SQL Server 2014.....	57
10.3.1.6	Where to go from here	57
11	Uninstall	57
11.1	Back up image files	57
11.2	Uninstall ITOM CDF	58
11.3	Uninstall DCA.....	58
11.3.1	Related topic	58
12	Send documentation feedback.....	59

Install

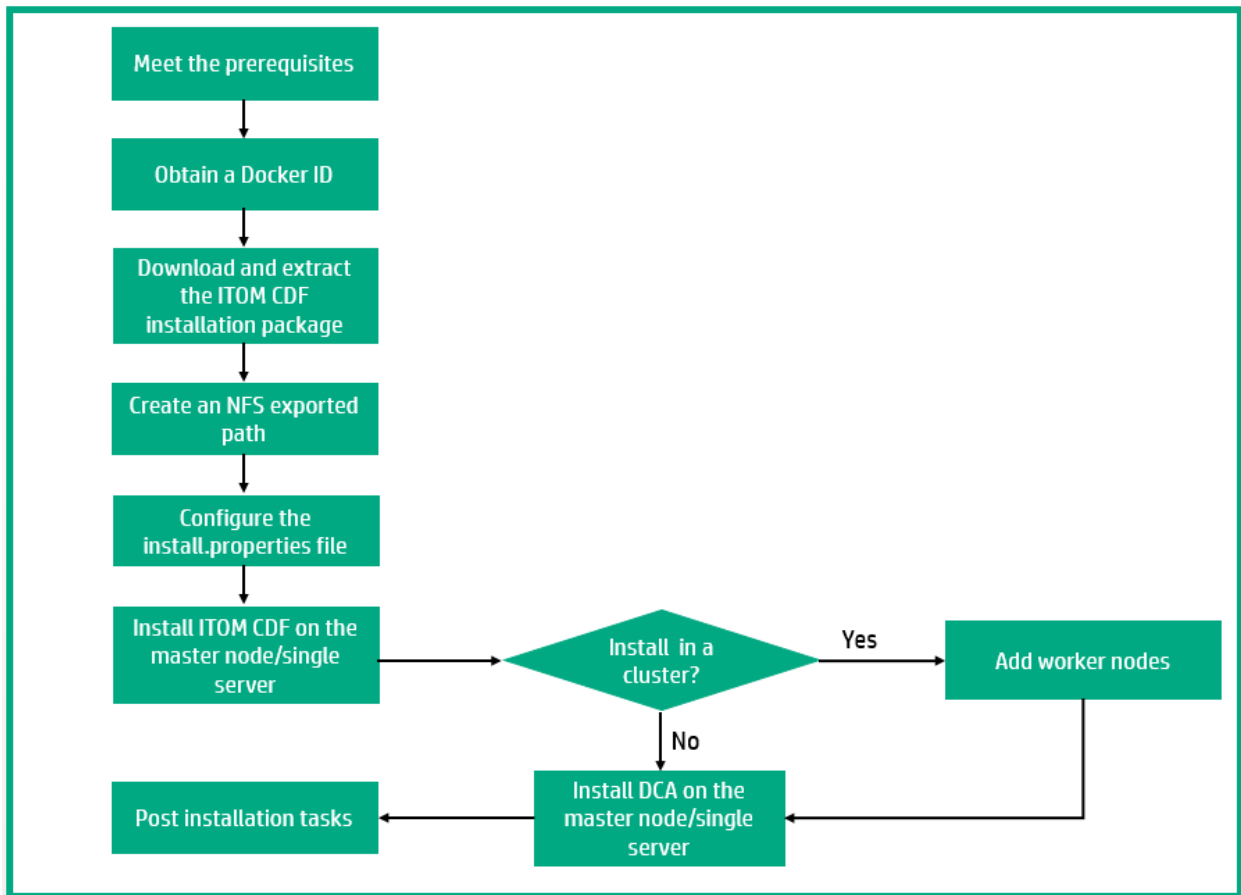
DCA is installed in the containerized mode that leverages technology based on [Docker](#) and [Kubernetes](#). In this mode, each suite component is deployed as a containerized application that is integrated with other components in the suite. You first install a container management framework (referred to as **ITOM Container Deployment Foundation (CDF)**) and then install DCA from a user interface based on this framework. DCA components are deployed quickly and integrated seamlessly, requiring little user intervention.

Installing DCA involves the following steps:

1. [Decide on your deployment model](#)
2. [Meet the system requirements as documented in the Support Matrix](#)
3. [Enable your Docker Hub account](#)
4. [Meet the prerequisites](#)
5. [Install and configure prerequisite components](#)
6. [Install ITOM CDF](#)
7. [Install DCA](#)
8. [Post installation tasks](#)
9. [Uninstall](#)

Decide on your deployment model

Your installation steps will vary depending on your deployment mode. Use the following flowchart to help you identify the steps to follow depending on your deployment mode:



In addition, see the [Support matrix](#) section for hardware and software requirements for your deployment.

Related topics

[Modes of deployment](#)

Support matrix

This section provides information about the supported hardware and software for DCA that you must have in order to successfully install and run the product.

Note

This release of DCA 2017.05 does not support multiple master nodes.

- [Hardware](#)
- [Operating systems](#)
- [Supported Docker images](#)
- [High-Availability products](#)

- [Web browsers](#)
- [Additional requirements](#)
 - [Languages](#)
 - [Integrations](#)
- [Managed devices, systems, and applications](#)
- [Performance and sizing](#)
- [Tuning](#)

Hardware

The following table shows the hardware requirements for DCA:

CPU	RAM	Disk Space
8 cores <ul style="list-style-type: none"> • Intel x64 processors (x86_64) • AMD x64 processors (AMD64) 	32 GB	200 GB excluding NFS Server

Operating systems

DCA can be used with the following operating systems:

Operating System	Architecture Type	Version
Red Hat Enterprise Linux	x86_64	7.2

Supported Docker images

DCA supports the following Docker images for deployment and compliance:

Docker image ID:tag	Bundled OS	Bundled application
httpd:2.4	Ubuntu 12.04 LTS	Apache HTTP Server 2.4
tomcat:6	Ubuntu 12.04 LTS	Apache Tomcat 6
mysql:5.6	Ubuntu 12.04 LTS	MySQL Community Server 5.6
ubuntu:12.04	Ubuntu 12.04 LTS	None

High-Availability products

To provide high availability to its components, DCA uses Kubernetes in its infrastructure layers.

Web browsers

One of the following web browsers is required to run DCA:

Browser	Version
Internet Explorer	11
Mozilla Firefox	52 ESR
Google Chrome	58.0
Safari (on MacOS)	10.1

Additional requirements

Languages

Localization is not supported for DCA.

Integrations

Information about the additional components that DCA requires to perform the analytics, reporting, and provisioning functions is available at [Integrate](#).

Managed devices, systems, and applications

DCA enables you to automatically provision, upgrade, and patch databases on discovered hardware (physical or virtual). The following table provides a list of supported operating systems that can be managed by this version of the DCA:

Operating system	Version
Red Hat Enterprise Linux	5.x, 6.x, 7.x
Microsoft Windows	Microsoft Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016
	Microsoft Windows 7, 8.1, 10
Ubuntu	12.04, 14.04, 16.04

CentOS	7.x
Oracle Enterprise Linux	5.x, 6.x, 7.x
Open Suse	42.2
Oracle Solaris	11 x86, 11 SPARC

The following table shows the supported hypervisor that can be managed by this version of DCA:

Hypervisor	Version
VMware ESXi	5.0, 5.1, 5.5, 6.0

The following table provides a list of supported databases that can be managed by this version of DCA along with the resource discovery and compliance support:

Database	Version	Platform	Resource Discovery	Compliance Scan (Ad Hoc)
Oracle	11g	Red Hat Enterprise Linux 6.x	✓	✓
		Red Hat Enterprise Linux 7.x	✓	✓
		CentOS 7.x	✓	✓
		Open Suse 42.2	✓	✗
		Oracle Solaris 11 x86, 11 SPARC	✓	✓
	12c	Red Hat Enterprise Linux 6.x	✓	✓
		Red Hat Enterprise Linux 7.x	✓	✓
		CentOS 7.x	✓	✓
		Open Suse 42.2	✓	✗
		Oracle Solaris 11 x86, 11 SPARC	✓	✓
IBM DB2	10.5	Red Hat Enterprise Linux 6.7	✓	✓

Performance and sizing

The following is the sizing recommendations for DCA:

The deployment size is defined as follows:

- Maximum number of resources: 5000
- Maximum number of concurrent operations: 179

Hardware requirement:

- **1 master and 2 worker nodes**
8 CPU cores / 32 GB RAM / 200 GB disk for each node
- **1 separate NFS Server**
200 GB, RAID 10, IO: 280 MB/s

Tuning

The following is the tuning recommendation for DCA:

- On the NFS server, increase the NFS server process count to 16.
- On all DCA hosts (master and worker nodes) change the following parameters:
 - `vm.dirty_background_ratio=5`
 - `vm.dirty_ratio=10`
 - `vm.swappiness= 10`

Related topics

[Prerequisites](#)

[Install DCA](#)

Installation checklist

The following section is a checklist of installation tasks that you need to perform to complete the DCA installation.

State	Steps
Plan	<input type="checkbox"/> Read about the hardware requirements, operating systems, and architecture to plan your suite deployment
	<input type="checkbox"/> Plan your deployment mode
Prepare	<input type="checkbox"/> Obtain a Docker Hub account from HPE
	<input type="checkbox"/> Meet the hardware and software prerequisites
	<input type="checkbox"/> Install the prerequisite software

	<input type="checkbox"/> Download and unzip the ITOM CDF installation package from the HPE Software Entitlement Portal
Install ITOM CDF	<input type="checkbox"/> Configure the install.properties file
	<input type="checkbox"/> Create an NFS exported path
	<input type="checkbox"/> Install ITOM CDF on the master node
	<input type="checkbox"/> Add worker nodes
Install DCA	<input type="checkbox"/> Prepare DCA images
	<input type="checkbox"/> Configure ChatOps
	<input type="checkbox"/> Run the suite installer
Post installation tasks	<input type="checkbox"/> Configure SA
	<input type="checkbox"/> Copy the database binaries

Enable your Docker Hub account

You must create a Docker Hub account, and then ask HPE to enable your Docker Hub account so that you can download (pull) the DCA suite images from Docker Hub.

To enable your Docker Hub account:

1. Create a Docker Hub account:
 - a. Go to <https://hub.docker.com>.
 - b. Type a Docker ID, your company email address, and then a password.
 - c. Click **Sign Up**.
You receive an email from Docker Hub, asking you to confirm your email address. Confirm your email address swiftly.
2. Log on to <https://hub.docker.com> with your Docker ID.
3. On the top right corner of the page, click **Settings** under your profile and take a screenshot to include your Docker ID and the linked email address.
4. Send the following information together with the screenshot to the HPE software fulfillment and licensing team specific for your region to enable your Docker Hub account:
 - Your company name
 - Your HPE Customer SAID (must be valid and active)
 - HPE ITOM CDF edition (that is, DCA)

i Info

Send your information to the email address of the HPE software fulfillment and licensing team for your region:

- Americas region: dockersupport.ams@hpe.com
- APJ region: dockersupport.apj@hpe.com
- EMEA region: dockersupport.emea@hpe.com

Once your Docker ID is enabled, you will receive a confirmation from HPE.

Where to go from here

[Prerequisites](#)

Prerequisites

This section provides information on preparing the environment to install and deploy ITOM CDF and DCA.

The following subsections are included:

- [System requirements](#)
 - [DCA server requirements](#)
 - [Disk setup \(optional\)](#)
 - [ITOM CDF license](#)

Other than the requirements mentioned in this topic, DCA also requires you to install certain prerequisite components. For more information, see [Install and configure prerequisite components](#).

System requirements

See the [Support matrix](#) section for hardware and software requirements.

DCA server requirements

The following requirements must be met on the DCA server before you install ITOM CDF and DCA:

- Ensure that the user installing ITOM CDF and DCA has root access or a sudo access to the host system.

- Ensure that the DCA server does not have Docker or Kubernetes installed. Uninstall Docker and Kubernetes if they are already installed.
- Disable the firewall by running the following commands:
 - `systemctl stop firewalld`
 - `systemctl disable firewalld`
- Ensure that the following ports that are required for installation are not used by the DCA server:

Port number	Description
111	ITOM CDF
2380	ITOM CDF
2048	ITOM CDF
2049	ITOM CDF
4001	ITOM CDF
4194	ITOM CDF
5000	ITOM CDF
5443	ITOM CDF interface
8080	ITOM CDF
8200	ITOM CDF
8443	ITOM CDF
33071	Universal configuration management database (UCMDB)
33080	Application Programming Interface (API)
33081	DCA user interface
33085	Model adapter
33092	Kafka service
33111	UCMDB
33332	PostgreSQL for HPE OO
33432	PostgreSQL for UCMDB

33443	Identity management (IDM)
33444	API
33445	HPE OO central
33480	HPE OO central
33532	PostgreSQL
33822	IDM
10250-10255	ITOM CDF

- Install the following RPM packages on all the DCA servers using the `yum install [package name]` command.
 - device-mapper-libs
 - java-1.8.0-openjdk
 - libgcrypt
 - libseccomp
 - libtool-ltdl
 - lsof
 - net-tools
 - nfs-utils
 - systemd-libs (version \geq 219)
 - unzip
- Ensure that the /tmp directory of the [targeted system](#) have enough free space (at least 2.5 GB) when adding worker node from ITOM CDF UI.
- If you have previously installed ITOM CDF, remove the shared NFS folder by running the `rm -rf /var/vols/itom/core/*` command.
- Ensure that every node use static IP address.
- Add an IP address to the `no_proxy` list for all master nodes and worker nodes:
 - Add the IP address of the master node to the `no_proxy` list in single-master node deployment mode.

Configure Chrony

Configure NTP using Chrony to synchronize time on all of the DCA host systems. Chrony is installed by default on a few versions of Red Hat/CentOS. However, if Chrony is not installed or running on your system, ensure that the NFS server, master nodes, and the worker nodes are installed under the same subnet.

1. Install Chrony using the `# yum install chrony` command.
2. Start Chrony using the following commands:


```
# systemctl start chronyd
# systemctl enable chronyd
```
3. Verify that Chrony is operating correctly using the `# chronyc tracking` command.

Note

You can use other tools to synchronize system time, for example, `ntp`.

Disk setup (optional)

Perform the following steps below to ensure that you have enough logical volumes for the ITOM CDF installation:

Note

You can choose any volume group name, logical volume, name, and disk location address for your installation according to your system.

1. Prepare a physical disk for the ITOM CDF cluster nodes.
2. Create a volume group using the `# vgcreate [volume group name] [logical volume name]` command.

Example

```
# vgcreate core-platform /dev/sdb
```

3. Create a logical volume for the ITOM CDF installation using the `# lvcreate -l 100%FREE -n [logical volume name] [volume group name]` command. For example, utilize 100% of the volume group.

Example

```
# lvcreate -l 100%FREE -n mylv core-platform
```

4. Activate the volume group using the `# vgchange -ay [volume group name]` command.

Example

```
# vgchange -ay core-platform
```

5. Format the file system using the `# mkfs.ext3 [logical volume path]` command.

Example

```
# mkfs.ext3 /dev/core-platform/mylv
```

6. Mount the volume group under the folder where you install the ITOM CDF using the **# mount [logical volume path] [platform installation folder]** command.

Example

```
# mount /dev/core-platform/mylv /opt/coreplatform
```

7. Configure the installation path in the [Configure the install.properties](#) file to use your specific path.

ITOM CDF license

The ITOM CDF license is included in the DCA license.

Where to go from here

[Install and configure prerequisite components](#)

Install and configure prerequisite components

DCA requires additional components to perform analytics, reporting, and provisioning functions. The following table lists the necessary components that must be installed and configured before you start DCA installation.

Component	Version	Install	Configure
Server Automation (SA)	10.50*, 10.51, 10.60	Yes	Yes
Operations Bridge Reporter (OBR)	10.x	Yes	Yes
(Optional) Cloud Optimizer (CO)	3.01	Yes	Yes

* The ad hoc compliance scan and remediation job on the resources discovered through SA 10.50 will work only if the ROLLUP_10.50.002_71107 is applied to SA 10.50.

For information about integrating the prerequisite components, see [Integrate](#).

To install the components:



Note

Ensure that you follow the order of the installation that is provided in this section.

1. Install SA. For more information about SA, click [here](#).
2. Install and [configure OBR](#). For more information about OBR, see the [HPE Support website](#).
3. Note down the Vertica database details of OBR. The Vertica database installed during the OBR installation serves as the external analytical data store of DCA.



Tip

Later, in the DCA installation procedure, you will be required to specify the access details of a Vertica database server. Note down the following details of the Vertica database server while creating the [Vertica database schema](#).

- Vertica database server host name
- Vertica port
- Vertica database name
- Vertica admin user name
- Vertica admin password

4. Install the OBR-DCA Suite Content Pack version 10.10.001:



Note

It is not mandatory that the OBR-DCA Suite Content Pack be installed at this stage. You can complete this task even after installing DCA.

- a. Download the OBR-DCA Suite Content Pack version 10.10.001 from **<installation_directory>/dcashare/content/DCAReportingContent**.
- b. Copy the **OBR-DCA-Reports-Content-Pack-38-20170111.zip** file to the OBR core under the **/opt/HP/BSM/PMDB/packages** directory.
- c. Extract the contents of the zip file, where you will find the **HPE_Data_Center_Automation** folder.
- d. Log on to the OBR Administration Console to install the DCA Suite Reports Content Pack.
- e. Go to **Administration > Deployment Manager**.
- f. Select the **HPE DCA Suite** check box and the required components from the **Data Source Application** column. Then, scroll down to the bottom of the page and click **Install/Upgrade**. The installation process starts and after completion, the **Installation Successful** message is displayed.

- g. Monitor the log files related to the content pack installation on the OBR server, in the **cd /opt/HP/BSM/PMDB/log** directory:
tail -f packagemanager.log
5. (Optional) Install HPE Cloud Optimizer. For more information about the Cloud Optimizer, click [here](#).

Where to go from here

[Install ITOM CDF](#)

Related topics

[Integrate with Cloud Optimizer](#)

[Integrate with OBR](#)

[Integrate with Server Automation](#)

Install ITOM CDF

DCA must be deployed on ITOM CDF that provides a graphic user interface for suite administrators to deploy and administer suites.

This section contains information about installing ITOM CDF. Installing ITOM CDF comprises the following steps:

- [Download, verify, and unzip the contents of the ITOM CDF installation package](#)
- [Create an NFS exported path](#)
- [Configure the install.properties file](#)
- [Modes of deployment](#)

For information on uninstalling ITOM CDF, see [Uninstall](#).

Where to go from here

[Install DCA](#)

Related topics

[Prerequisites](#)

[Install and configure prerequisite components](#)

[Administer](#)

Download, verify, and unzip the contents of the ITOM CDF installation package

Once your environment meets the [Prerequisites](#), you can download the ITOM CDF installation package to the master node and verify the package.

1. Download the ITOM CDF installation package (HPESW_ITOM_DCA_Platform_2017.03.00200.zip) from the [HPE Software Entitlement Portal](#) to a temporary folder of a server with the [supported operating system](#).
2. Verify if the md5 checksum of the installation package matches the corresponding md5 checksum value listed in the following table:

File name	Size	MD5 checksum
HPESW_ITOM_DCA_Platform_2017.03.00200.zip	3635434673	a98506a9d1304c6213b4c4d533bb4387

3. Run the following commands to unzip the ITOM CDF installation package:

```
unzip HPESW_ITOM_DCA_Platform_2017.03.00200.zip
```

The HPESW_ITOM_DCA_Platform_2017.03.00200.zip file includes the following files and directories.

File name	Description	Type
bin	<p>The bin directory includes:</p> <ul style="list-style-type: none"> • All the runtime files that are core of the container platform: docker runtime binaries (docker, docker-containerd, docker-containerd-ctr, docker-container-shim, dockerd, docker-proxy, docker-runc), the binary to access the distributed configuration database (etcdctl), the runtime to interact with Kubernetes(kubectl). • The scripts used to check the ITOM CDF (kube-restart.sh, kube-start.sh, kube-stop.sh). • The script to check that everything is running (kube-status.sh). • The script used during installation to create the configuration for Docker (mk-docker-opts.sh) and vault that is used for security purposes to store sensitive information and to generate and manage certificates for the ITOM CDF and the suite deployment. 	Directory
cfg	The initial user and role information that will be seeded into IDM to create user accounts (single sign on).	Directory
images	All the core platform images and share services images	Directory

File name	Description	Type
install	The binary that needs to be run to install ITOM CDF	File
install.properties	The properties file used to configure the installation.	File
jar	-	Directory
manifests	The manifests contain YAML files that describe how to deploy a container The image for Kubernetes.	Directory
objectdefs	-	Directory
rpm	-	Directory
scripts	-	Directory
uninstall.sh	Use to uninstall ITOM CDF	File
version.txt	-	File
zip	-	Directory

Where to go from here

[Create an NFS exported path](#)

Related topic

[Administer](#)

Create an NFS exported path

The ITOM CDF requires an NFS exported directory to be used as the persistent volume. To do this, you can either configure a separate NFS server or use one of the nodes in which you will install DCA as the NFS server.

If you configure a separate NFS server, then you can use the NFS server irrespective of whether all the nodes are running or not. If you use one of the nodes as the NFS server, and if the node stops running, then you will lose your NFS instance causing DCA to stop working.

Note

It is recommended that you configure a separate NFS server.

Use one of the nodes as the NFS server

1. Go to the **HPESW_ITOM_Suite_Platform_2017.03.00200/scripts** directory.
2. Run the `./setupNFS.sh` script on the master node.

Configure a separate NFS server

Note

The instructions in this procedure pertain to RHEL 7.0. If you are using a different OS, see the OS documentation for relevant commands.

1. Install the NFS server packages with the command: `yum install -y nfs-utils`
2. Enable the NFS service with the commands:
`systemctl restart rpcbind`
`systemctl enable rpcbind`
`systemctl restart nfs-server`
`systemctl enable nfs-server`
3. Create a directory for the persistent volume to be used during the ITOM CDF installation:
`mkdir -p /var/vols/itom/core`
`chown -R 1999:1999 /var/vols/itom/core`

**Notes**

The NFS export path that you provide here must be different from the one used when installing DCA.

You can create a directory using any name of your choice. Use the same directory name in the new line that you add in step 4.

Ensure that the GID and UID (represented by 1999 in the command above) are not used by any other application.

4. Export the directory that you created in step 3:
 - a. Edit **/etc/exports** and add a new line as below:

```
/var/vols/itom/core  
*(rw, sync, anonuid=1999, anongid=1999, all_squash)
```
 - b. Run `exportfs -ra`

**Tip**

Use `exportfs` to check whether the directory has been exported.

**Best practice**

If you are unable to connect to the NFS server, check if the firewall is disabled on the NFS server or not.

Where to go from here

[Configure the `install.properties` file](#)

Related topics


[Administer](#)

Configure the `install.properties` file

To correctly configure the Kubernetes cluster, you must configure the following parameters in the `install.properties` file.



 **Tip**



List all the IP addresses of all the cluster nodes that you are going to install. You can update this file with the correct IP address when installing other nodes.


 **Note**

When you set FQDNs for the cluster nodes in the `install.properties` file, make sure the FQDNs are resolved to correct IP addresses, not the loop back IP 127.0.0.1.

Parameter	Description	Notes
<i>MASTER_NODES</i>	Lists the cluster master nodes (IPV4 address or FQDN), separated by a blank and enclosed in double-quotes. Example: <code>MASTER_NODES="10.10.10.10"</code>	Mandatory Only a single master node deployment is supported in this DCA release.
<i>WORKER_NODES</i>	Lists the cluster worker nodes, separated by a blank and enclosed in double-quotes. Suites are run on worker nodes. Example: <code>WORKER_NODES="10.10.10.20 10.10.10.21 10.10.10.22"</code>	Mandatory If you are not using a cluster setup, set the the same value in both the <code>MASTER_NODES</code> and <code>WORKER_NODES</code> parameters.

Parameter	Description	Notes
<i>INGRESS_HOST</i>	<p>Defines the IP address (IPV4 address or FQDN) of the node on which you want to start the Ingress Controller. You must use one of the master or worker nodes.</p> <div style="border: 1px solid #f0e68c; padding: 10px; margin: 10px 0;"> <p> Everything that runs on a cluster is actually on a private network, which is not externally accessible. If you want any suite functionality to be available from outside the network (for example, a Help Desk operative on client machine on another network that needs to access Service manager), you must provide an ingress into the cluster to be able to access the functionality. This is done by configuring the <code>INGRESS_HOST</code> and <code>EXTERNAL_ACCESS_HOST</code> parameters.</p> </div> <p>Example:</p> <p><code>INGRESS_HOST=10.10.10.10</code> (IPV4 address or FQDN of one of the master nodes)</p>	Mandatory
<i>EXTERNAL_ACCESS_HOST</i>	<p>Defines a fully-qualified hostname for external clients to access cluster services. The specified name must resolve the IP address where the ingress is running.</p> <div style="border: 1px solid #f0e68c; padding: 10px; margin: 10px 0;"> <p> Everything that runs on a cluster is actually on a private network, which is not externally accessible. If you want any suite functionality to be available from outside the network (for example, a Help Desk operative on client machine on another network that needs to access Service manager), you must provide an ingress into the cluster to be able to access the functionality. This is done by configuring the <code>INGRESS_HOST</code> and <code>EXTERNAL_ACCESS_HOST</code> parameters.</p> </div> <p>Example:</p> <p><code>EXTERNAL_ACCESS_HOST=myd.XXXX.YYY.net</code></p>	Mandatory

Parameter	Description	Notes
<i>NFS_SERVER</i>	<p>Specifies the IPV4 address or FQDN of the NFS server that serves the persistent volumes of the cluster services.</p> <div style="border: 1px solid #f96; padding: 10px; margin: 10px 0;"> <p> If a container stops and is restarted, all changes made inside the container are lost. If you want to save information such as configuration files, any other files, or databases, they must be located outside the container in a persistent volume provided by a Network File System (NFS). When you install the ITOM CDF, you must install an NFS server that shares out the network volumes. The server can be a master node or an external server.</p> </div> <p>Example: NFS_SERVER=16.255.25.255</p>	Mandatory
<i>NFS_FOLDER</i>	<p>Specifies the root folder (fully-qualified directory) for the persistent volume that the NFS server exports.</p> <div style="border: 1px solid #f96; padding: 10px; margin: 10px 0;"> <p> Note If a container stops and is restarted, all changes made inside the container are lost. If you want to save information such as configuration files, any other files, or databases, they must be located outside the container in a persistent volume provided by a Network File System (NFS). When you install the ITOM CDF, you must install an NFS server that shares out the network volumes. The server can be a master node or an external server.</p> </div> <p>Example: NFS_FOLDER=/var/vols/itom</p>	Mandatory
<i>ROOTCA</i>	<p>Specifies the root or intermediate CA certificate for generating server and client certificates. The value of the parameter is the file name of the CA certificate, including the absolute path.</p> <p>When you install the ITOM CDF, all communication between the components is secured by using https. Therefore, communications use certificates to maintain security. These certificates can be self-signed or signed with a Certificate Authority provided by the customer. The default value is a self-signed certificate.</p> <p>Example: ROOTCA=/tmp/ca.crt</p>	Optional

Parameter	Description	Notes
<code>ROOTCAKEY</code>	<p>Specifies the CA key for generating server and client certificates. The value of the parameter is the file name of the CA key, including the absolute path.</p> <p>When you install the ITOM CDF, all communication between the components is secured by using https. Therefore, communications use certificates to maintain security. These certificates can be self-signed or signed with a Certificate Authority provided by the customer. The default value is a self-signed certificate.</p> <p>Example:</p> <pre>ROOTCA=/tmp/ca.key</pre>	Optional
<code>NFS_STORAGE_SIZE</code>	<p>Specifies the size of the NFS volume exported by the NFS server.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>If a container stops and is restarted, all changes made inside the container are lost. If you want to save information such as configuration files, any other files, or databases, they must be located outside the container in a persistent volume provided by a Network File System (NFS). When you install the ITOM CDF, you must install an NFS server that shares out the network volumes. The server can be a master node or an external server.</p> </div> <p>Example:</p> <pre>NFS_STORAGE_SIZE=50Gi</pre>	Optional
<code>K8S_HOME</code>	<p>Specifies the installation directory (fully-qualified directory) for the core platform binaries.</p> <p>Example:</p> <pre>K8S_HOME=/opt/kubernetes</pre>	Optional
<code>MASTER_API_PORT</code>	<p>Specifies the http port for the Kubernetes (K8S) API server.</p> <p>If you want to use K8S, you must dock to the K8S API server. The kubectl command line tool communicates with the K8S server.</p> <p>Example:</p> <pre>MASTER_API_PORT=8080</pre>	Optional

Parameter	Description	Notes
<code>MASTER_API_SSL_PORT</code>	<p>Specifies the https port for the K8S API server.</p> <p>If you want to use K8S, you must dock to the K8S API server. The kubectl command line tool communicates with the K8S server.</p> <p>Example:</p> <pre>MASTER_API_SSL_PORT=6443</pre>	Optional
<code>THINPOOL_DEVICE</code>	<p>Specifies the Docker devicemapper storage driver.</p> <p>Format : Path to a device</p> <p>To configure the thinpool device, see https://docs.docker.com/engine/userguide/storagedriver/device-mapper-driver/#configure-direct-lvm-mode-for-production</p> <p>Note: If this parameter is specified, then the installation uses the devicemapper (direct-lvm) Docker storage driver. If it is not specified, then the installation uses devicemapper (loop).</p> <p>For production use, HPE recommends devicemapper (direct-lvm).</p> <p>Example:</p> <pre>THINPOOL_DEVICE= /dev/mapper/docker-thinpool</pre>	Optional
<code>DOCKER_HTTP_PROXY</code>	<p>Enter the HTTP proxy settings for Docker. Configure this parameter if access to the Docker Hub or registry requires a proxy (the default value is no proxy). The value of the parameter is any valid HTTP proxy URL.</p> <p>When you launch containers on Docker inside the Kubernetes cluster, you may need to download images from the internet, for which you need to use proxies.</p> <p>Example:</p> <pre>DOCKER_HTTP_PROXY="https://web.proxy.host.domain:8080"</pre>	Optional if the host system or all the nodes (in case of a cluster setup) is connected to the Internet.
<code>DOCKER_HTTPS_PROXY</code>	<p>Enter the HTTPS proxy settings for Docker. Configure this parameter if access to the Docker Hub or registry requires a proxy (the default value is no proxy). The value of the parameter is any valid HTTPS proxy URL.</p> <p>When you launch containers on Docker inside the Kubernetes cluster, you may need to download the images from the internet, for which you need to use proxies.</p> <p>Example:</p> <pre>DOCKER_HTTPS_PROXY="https://web.proxy.host.domain:8080"</pre>	Optional if the host system or all the nodes (in case of a cluster setup) is connected to the Internet.

Parameter	Description	Notes
<code>DOCKER_NO_PROXY</code>	<p>Enter a list of fully qualified domain names or IP addresses that can be accessed without a proxy.</p> <p>Example:</p> <p><code>DOCKER_NO_PROXY=127.0.0.1,localhost,<host name>,<IP address></code></p> <p><i><IP address></i> refers to an IP address that can be accessed from ITOM CDF without using a proxy.</p>	Optional
<code>REGISTRY_ORGNAME</code>	<p>Specifies the organization name where suite images are placed. The default name is <code>hpeswitomsandbox</code>.</p> <p>Format: A string</p> <p>Example:</p> <p><code>REGISTRY_ORGNAME=hpeswitom</code></p>	Optional
<code>FLANNEL_IFACE</code>	<p>Specify the interface for docker inter-host communication to use.</p> <p>Format: A single IPV4 address or interface name</p> <p>Example:</p> <p><code>FLANNEL_IFACE=10.10.10.10</code></p>	Optional

Where to go from here

[Modes of deployment](#)

Related topics

[Manual verification commands](#)

Modes of deployment

ITOM CDF supports the following modes of deployment:

- [One master node](#) or a single machine (non-cluster setup) (best practice in a Proof of Concept (POC) environment)
- [One master node and one/multiple worker nodes](#) (best practice in a development/quality assessment environment and production environment)

Based on your deployment scenario, choose the appropriate topic for information about installing ITOM CDF.

Where to go from here

[Install DCA](#)

Related topics

[Support matrix](#)

Install ITOM CDF on the master node or on a single server

Note

Add the IP address of the master node into the NO_PROXY list for both master node and worker nodes. If you are installing ITOM CDF in a single machine setup (non-cluster setup), ensure that you have set the same value in both the MASTER_NODES and WORKER_NODES parameters in the [install.properties](#) file.

1. Go to the installation directory:
`cd HPESW_ITOM_Suite_Platform_2017.03.00200/`
2. Run the following script:
 - `./install` (with the root user)
 - `sudo ./install` (with the non-root user)

The following components are installed:

- Base installation files
- Docker
- Certificates
- etcd
- Flannel
- Internal network
- Vault
- Images
- Configuration for K8S
- Persistent volumes
- All the base ITOM CDF services such as the postgresql for IDM, the management portal
- More SSL certificates for Nginx used for proxy requests into ITOM CDF

The Successfully completed configuring the HPE ITOM Core Platform on this server message indicates that the installation completed successfully.

Log on to the ITOM CDF UI using the `https://<dca-hostname>:5443` URL. Enter the default user name and password.

i Notes

To see what was installed, run the following commands:

```
cd <value of K8S_HOME in the install.properties file>
```

```
ls -l
```

To see the installation log, run the `vi /tmp/install-<timestamp>.log` command.

Installed directories and files

The following table lists the files and directories that are installed as part of the ITOM CDF installation:

Name	Description	Type	Remarks
bin	<p>The bin directory includes:</p> <ul style="list-style-type: none"> All the runtime files that are core to the container platform: docker runtime binaries (docker, docker-containerd, docker-containerd-ctr, docker-container-shim, dockerd, docker-proxy, docker-runc), the binary to access the distributed configuration database (etcdctl), the runtime to interact with Kubernetes (kubectl). Scripts used to check ITOM CDF (kube-restart.sh, kube-start.sh, kube-stop.sh) Script to check that everything is running (kube-status.sh) Script used during installation to create the configuration for Docker (mk-docker-opts.sh) and vault that is used for security purposes to store sensitive information and to generate and manage certificates for ITOM CDF and the suite deployment. 	Directory	

cf g	The cfg directory includes the Docker configuration. It includes docker and docker-bootstrap and idm. There are two Docker daemons running on each node. Only Docker is physically running on the host and everything else is containerized. So services or programs that you would typically run directly on the host, are now also run inside a container: docker-bootstrap instance. It runs etcd and flannel.	Di re ct or y	<ul style="list-style-type: none"> • To see what is running inside docker, run: <code>docker ps</code>. Kubernetes is actually running inside Docker. • To see what is running inside the bootstrap docker, run command: <code>docker -H unix:///var/run/docker-bootstrap.sock ps</code>. It runs flannel, vault, and etcd, which are containerized. Docker provides an abstraction layer from the host. • To see what is running inside the bootstrap-docker, which is a separate instance, you need to pass the socket of bootstrap-docker. run: <code>ps -cf grep dockerd</code>. K8S is actually running inside Docker. There are two K8S instances running: docker and bootstrap-docker that run on two different sockets. • To see what is running in docker, run: <code>docker ps</code>.
da ta	Data that is generated by K8S and is the runtime data for K8S.	Di re ct or y	To see what is in the data directory, run: <code>ls data/*</code> .
im ag es	All the core platform images that have been imported locally.	Di re ct or y	To see what is in the data directory, run: <code>ls images</code> .
lo gs	Logs of some of the components that are currently running.	Di re ct or y	To see what is in the log directory, run: <code>ls log</code> . To do a recursive log, run: <code>ls -R log</code> . All the components put their running information in the logs.

jar		Directory	
manifests	<p>manifests contain YAML files that describe how to deploy a container The image for Kubernetes.</p> <ul style="list-style-type: none"> manifests. contains YAML files that have to run on every node. They are K8S components: <ul style="list-style-type: none"> kube-apiserver.yaml for the K8S API server. kube-controller-manager.yaml controls access to the K8S server. kube-proxy.yaml contains proxy connections. kube-scheduler.yaml schedules on what node to execute a container. kube-registry-proxy.yaml starts the kube registry proxy container. 	Directory	
objectdefs	objectdefs contains more YAML files for autopass, idm, persistent volumes, registry proxies, vault, management portal, Nginx controller, and the suite installer.	Directory	
rpm	rpm is an installable package used to enable the installation of an NFS server. The NFS utility helps sharing data via a networked volume.	Directory	
runconf	runconf is a transient directory used during the installation.	Directory	
ssl	ssl contains all the certificates and the keys that have been generated by the running ITOM CDF.	Directory	

un in st all .s h	Uninstall script	Di re ct or y	To uninstall ITOM CDF run <code>./uninstall.sh</code> . The uninstall process stops containers and removes them, removes daemons, and more. You need to reboot the server afterwards.
to ol s	The support toolset for troubleshooting. For more information, see Support toolset .	Di re ct or y	
zi p	The zip directory includes a subset of files used to install a new cluster node from the Management Portal Add Node functionality.	Di re ct or y	

Where to go from here

[Install ITOM CDF on a worker node](#)

Install ITOM CDF on a worker node

To install ITOM CDF on worker nodes, in the ITOM CDF management portal:

1. Click **ADMINISTRATION > Nodes**.
2. In the Nodes area, click **+ ADD**.

Repeat this procedure for the number of worker nodes you want to add.

To see the installation log, run the `vi /tmp/install-<timestamp>.log` command.

For information about the components, files, and directories that are installed, see [Install ITOM CDF on the master node or on a single server](#).

Where to go from here

[Install DCA](#)

Install DCA

After installing ITOM CDF, you can use the ITOM CDF UI to install DCA. Installing DCA comprises the following main steps:

1. [Ensure that the master node has access to Docker Hub](#)
2. [Prepare DCA images](#)
3. [Configure the NFS server share](#)
4. [Configure ChatOps](#)

5. [Install DCA](#)
6. [Verify the installation](#)

Ensure that the master node has access to Docker Hub

Ensure that the master node has access to Docker Hub by running the `docker pull hello-world` command.

Note

If you are unable to successfully run the command, it means that the master node does not have access to Docker Hub. In that case, you can [download the images to another system and copy them to the master node](#).

Prepare DCA images

After ensuring that the master node has access to Docker Hub, you must download and then upload DCA images to a private registry by running the `downloadimages.sh` and `uploadimages.sh` scripts:

Important

Perform steps 1 to 4 in this procedure only if the `DOCKER_HTTP_PROXY` and `DOCKER_HTTPS_PROXY` parameters were not [set in the install.properties file](#) while installing ITOM CDF.

1. Create a directory using the `mkdir -p /usr/lib/systemd/system/docker.service.d` command.
2. Configure a proxy:


```
cat << EOF > /usr/lib/systemd/system/docker.service.d/http_proxy.conf
[Service]
Environment="HTTP_PROXY=<Your Proxy>" "HTTPS_PROXY=<Your Proxy>"
EOF
```
3. Restart ITOM CDF services:


```
$K8S_HOME/bin/kube-restart.sh
```
4. Wait for a few minutes and check if the ITOM CDF services have restarted without any failures:


```
$K8S_HOME/bin/kube-status.sh
```
5. (*Optional*) If the Docker content trust is to be enabled during the DCA image download, export the following proxy variables in a shell environment:


```
export http_proxy=<your proxy>
export https_proxy=<your proxy>
```
6. Run the following commands to execute the `downloadimages.sh` script:


```
cd $K8S_HOME/scripts
./downloadimages.sh -r docker -o hpeswitom -s dca -c on -v 2017.05 -u
<username> -p <password>
where
```

<username> and <password> are the Docker hub credentials that you obtained from HPE

`-c on` enables the Docker content trust for image download

This is an optional parameter and must be performed only if you perform step 5.

The script starts the downloading process. When the script has finished execution, the following message is displayed:

Successfully downloaded the DCA suite version: 2017.05...

You can see the .tar files of the images in the .tar directory of the suite images (default: `/var/opt/kubernetes/offline/suite_images`).

7. Run the `./uploadimages.sh -s dca` command. This command uploads DCA images to a private registry.



Best practice

After running the `uploadimages.sh` script, ensure that the suite data was successfully imported. The "Upload suite feature data completed" message in the log files in the `/tmp` directory indicates a successful import

The **Upload-process successfully completed** message will be displayed to indicate that the images are loaded. Ensure that you check the

`/tmp/uploadsuiteimages-<timestamp>.log` file for errors.

Configure the NFS server share



Note

The NFS export path that you provide here must be different from the [one you provide when installing ITOM CDF](#).

1. Log on to the NFS server.
If you do not have a separate NFS server, use the master node of the ITOM CDF cluster (also referred to as Kubernetes cluster). The master node was configured as an NFS server when you [configured the install.properties file](#) before installing ITOM CDF.
2. Create a directory to store the suite data. For example, in this procedure, create the `/vols/dca` directory. To create this directory, run the following command:

```
mkdir -p /vols/dca
```


The directory will be shared using NFS, to make it available to DCA.
3. Export the directory that you created in step 3:
 - a. Edit `/etc/exports` and add a new line as below:
`<export_path> *(rw, sync, no_subtree_check, no_root_squash)`
 - b. Run `exportfs -ra`

**Note**

Using * will allow any machine to access the share. You can restrict the machines that have access to the export, by using host names with wildcards, specifying IP networks, or by explicitly specifying the nodes. If you are specifying the nodes, you must update the configuration each time you add a new node to the cluster.

For example, to export the /vols/dca directory, add the following line in /etc/exports:
/vols/dca *(rw, sync, no_subtree_check, no_root_squash)

Your NFS server is configured. You can now log on to the ITOM CDF UI and navigate to suite installation page to start the DCA installation.

Configure ChatOps

ChatOps is available out-of-the-box when DCA is installed. You need to configure ChatOps before you can start using it. This section contains the configuration tasks pertaining to ChatOps.

**Note**

Configuring ChatOps is optional. You can skip these steps if you will not use ChatOps.

To set up ChatOps to work with DCA:

1. [Create a Slack team](#)
2. [Create a private Slack app](#)
3. [Enable incoming web hook integration](#)
4. [Authorize the HPE DCA bot with your Slack team](#)

Create a Slack team

To start using ChatOps, you have to first create a Slack team.

1. Go to <https://slack.com/>.
2. Type your email ID and click **Create New Team**.
You will receive an email that contains the confirmation code.
3. Type the confirmation code in the Slack website.
4. Type your full name and a user name.
5. Type a password. This is the password that you will use to log into Slack.
6. Type a name for your team.
7. Type your team domain. You can invite only those people who have email IDs that belong to the domain to be a part of your Slack team.

8. Type the email IDs of the people whom you want to be a part of the team and click **Send Invitations**. The email IDs must belong to the domain you specified in the previous step. Alternatively, you can also choose to send invitations later.

Every invitee receives a link from Slack.com to join the Slack team.

Create a private Slack app

1. Go to <https://api.slack.com/slack-apps>.
2. Click **Create an App**.

Create an App

App Name

e.g. Super Service

Don't worry; you'll be able to change this later.

Development Slack Team

Slack team

This team owns your Slack app (and if you lose access to the team, you won't be able to administer the app). You can't change this later.

I plan to submit this app to the Slack App Directory.
We'll help you get your app ready for submission. If you're not sure, you can decide to submit later.

By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel Create App

3. In the **Create an App** dialog box:
 - a. In the **App Name** field, type a name for the app.
 - b. In the **Development Slack Team** field, select the team.
 - c. Select the **I plan to submit this app to the Slack App Directory** check box.
4. Click **Create App**. The **App Credentials** screen that contains the Client ID and Client Secret is displayed. Copy the Client ID and Client Secret.

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

Client ID

Client Secret

You'll need to send this secret along with your client ID when making request to our API.

5. Go to **Bot Users** and click **Add a Bot User**.
6. Assign a name to the bot and click **Add Bot User**. For example, you can assign the name "Otto".

- Basic Information
- Collaborators
- OAuth & Permissions
- Bot Users
- Interactive Messages
- Slash Commands
- Event Subscriptions
- Submit to App Directory

Slack ♥

- Help
- Contact
- Policies
- Our Blog

Bot User

You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about [how bot users work](#).

Default username

If this username isn't available on any team that tries to install it, we will slightly change it to make it work. Usernames must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores.

7. Click **OAuth & Permissions**.
8. In the **Redirect URL(s)** field, type <http://localhost:4000/oauth>.
9. Click **Save Changes**.

Enable incoming web hook integration

1. Go to <https://api.slack.com/incoming-webhooks>

This document is an export from the HPE Software Documentation Portal. For the latest documentation, refer <https://docs.software.hpe.com>.

39

2. Click the **incoming web hook** integration link.

Send data into Slack in real-time.

Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a **JSON** payload that includes the message text and some options. **Message Attachments** can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

Start by setting up an incoming webhook integration in your Slack team to try these features out:

1. [Sending messages](#)
2. [Adding links](#)
3. [Customizations for custom integrations](#)
4. [Make it fancy with advanced formatting](#)
5. [Putting it all together](#)
6. [Distributing as a Slack app](#)

3. In the **Post to Channel** form, select the #general channel from the list. You can click on any existing channel.
4. Click the **Add Incoming WebHooks integration** button.
5. Note down the Webhook URL that is generated. This will be the value of the **Formatting Webhook URL** field that will be specified when you install ChatOps for DCA.

Authorize the DCA bot with your Slack team

1. Start the Slack app authorization process by running the following commands:


```
docker run -ti -p 4000:4000 -e "HTTP_PROXY=http://<proxy-server-name>:8080/" -e "HTTPS_PROXY=http://<proxy-server-name>:8080/" -e "NO_PROXY=<dca-hostname>,localhost,127.0.0.1,localaddress,.localdomain.com" -e "http_proxy=http://<proxy-server-name>:8080/" -e "https_proxy=http://<proxy-server-name>:8080/" -e "no_proxy=<dca-hostname>,localhost,127.0.0.1,localaddress,.localdomain.com" <URL to the ChatOps container image> /bin/bash
```

<URL to the ChatOps container image> is the location in which the ChatOps Docker image has been uploaded.
2. Run the following command from the **/bot** directory after you are connected to the container shell:


```
coffee install-slackapp.coffee <client_id> <client_secret> <bot_name>
```

where *client_id* and *client_secret* are the values of the Client ID and Client Secret you obtained while creating a private Slack app and *bot_name* is the name you assigned to the bot.

3. Go to `http://<dca-hostname>:4000/`.
4. Click **Add to Slack**.

5. Confirm that you are adding the DCA bot to the right team, and click **Authorize**. Once the authorization process is successfully complete, your browser then returns a **Success!** message, and the command line console returns the following messages:

Slack App is successfully installed

Run hubot with the following environment variables:

```
export HUBOT_SLACK_TOKEN=<TOKEN>
```

```
export SLACK_APP_TOKEN=<TOKEN>
```

You will need the values of HUBOT_SLACK_TOKEN and SLACK_APP_TOKEN when installing ChatOps for DCA.

 **Note**

If you are redirected to a page that displays the "This site cannot be reached localhost refused to connect" message, in the URL, change <localhost> to the DCA host name in all the steps of this procedure and click Enter.

Install DCA

After setting up the NFS server, you are ready to install DCA by using the ITOM CDF UI. This section will guide you through the installation steps.

 **Note**

During the DCA installation, do not use any of your browser buttons (such as **Back** or **Refresh**); unexpected errors may occur.

1. Launch ITOM CDF on a [supported web browser](#):
`https://<dca-hostname>:5443`
 <dca-hostname> is the fully qualified domain name (FQDN) of the host where you installed ITOM CDF.
2. Log on to ITOM CDF as the admin user. Use the [password that you specified after your initial login](#).
3. On the left navigation bar, expand the **SUITE** node and click **Installation**. A Welcome page is displayed.
4. Click **Next**.
5. The Review Licensing Agreement screen is displayed. Select both the check boxes to agree to the license agreement and privacy policy.
6. Click **Next**.
7. Select **Data Center Automation**. The current version of the suite is automatically displayed and selected.
8. Click **Next**.
9. Customize your suite deployment. Currently, only the **Premium** option is available.
10. Select **Premium** if not already selected and click **Next**.

11. Configure the suite storage. Currently only NFS is supported.
If you are using the [same NFS server as the one you provided when installing the ITOM CDF](#), select **Use system default NFS server** and specify the path to which you exported in the [Configure the NFS server share](#) section.
If you are using a different server, select **Use another NFS server** and specify the host and the path you configured in the [Set up the NFS server](#) section.
12. Click **Next** to configure the DCA installation.
13. In **General Settings**, type a web proxy URL in the **HTTP Proxy** field. This is an optional step.
14. Click **Next** to display the OS Provisioning page.
15. Specify the Server Automation (SA) host and user credentials.
For more information about SA user permissions for OS provisioning, see [OS provisioning permissions](#) in the SA document.
16. Click **Next** to display Analytics page.
17. Choose either of the analytical data store type:
 - Internal Analytical Data Store

**Note**

Choose Internal Analytical Data Store only for non-production environments.

- External Analytical Data Store
 - Specify values for the following:

**Tip**

Type the values that you noted down in [step 3](#) in [Install and configure prerequisite components](#).

- Veritica Hostname/IP Address
- Veritica Port
- Veritica Admin UserName
- Veritica Admin Password
- Veritica Database Name

Click **Next** to display Cloud Optimizer configuration page.

**Note**

Configuring Cloud Optimizer is optional.

18. Select the Configure Cloud Optimizer option, and then type the following details:

- Database Hostname/IP Address: Type the FQDN or IP address of the HPE Cloud Optimizer database.
- Database Port: Type the port of the HPE Cloud Optimizer database.
- Database Password: Type the password of the HPE Cloud Optimizer database. Do not type anything if you did not configure HPE Cloud Optimizer to use a non-default database password.

19. Click **Next** to display ChatOps configuration page.



Note

Configuring ChatOps is optional.

20. Specify the following Slack information if you select the **Install ChatOps** check box:
- In the **Hubot Slack Token** field, type the value of the HUBOT_SLACK_TOKEN token you generated in the [Authorize the HPE DCA bot with your Slack team](#) section.
 - In the **Slack App Token** field, type the value of the SLACK_APP_TOKEN token you generated in the [Authorize the HPE DCA bot with your Slack team](#) section.
 - In the **Formatting Webhook URL** field, type the value of the Webhook URL that you generated in the [Enable incoming web hook integration](#) section.
21. Click **Next** to display the Install page.
22. Click **Install DCA Suite** to install DCA.



Best practice

Ensure that you use the host name to connect to the ITOM CDF UI (for example, `https://hostname:5443/`).

The Status page displays the installation progress and overall status of all the services you chose to deploy.

- A check mark and a green progress bar are displayed against the services that are installed successfully.
- A cross mark and a red progress bar indicating the installation failure are displayed against the services that failed to install.

The status of the installation is displayed at the bottom of the page:

- If the installation is success, the following message is displayed along with the link to the DCA login page:
"Data Center Automation Suite services have successfully started. [Click here to login.](#)"

- If the installation fails, a corresponding error message is displayed.



Note

To install DCA license, see [License Management](#).

Verify the installation

To verify if DCA is installed successfully, perform one of the following:

Verify the deployment using the ITOM CDF interface

1. Log into ITOM CDF. Open the following in a supported browser:
`https://<dca-hostname>:5443`
 where *dca-hostname* is the FQDN of the system on which DCA is installed.
2. Type the user name and password of the admin user.
3. Click the **RESOURCES** menu on the left pane.
4. Select the namespace you created from the **Namespace** submenu.
5. Click **Workloads**.
6. Click **Pods**.

If the pods are running or have succeeded, then DCA is successfully installed.

Verify the deployment using the command prompt

Run the following command on the DCA server:

```
kubectl get pods --namespace <namespace-name>
```

If the status of all the pods is "Running", then DCA is successfully installed.

Using the DCA interface

1. Log on to DCA: `https://<dca-hostname>:33081`
 where *<dca-hostname>* is the FQDN of the system on which DCA is installed.
2. Type the user name and password for the default user. The default user name is admin and the default password is propel.

If the DCA dashboard is displayed, then DCA is successfully installed.

The **Review Licensing Agreement** screen is displayed. Select both check boxes to agree to the license agreement and privacy policy.

Where to go from here

[Post installation tasks](#)

Related topics

[Support matrix](#)

[Prerequisites](#)

[Install ITOM CDF](#)

Post installation tasks

Before using DCA, perform these tasks:

- [Configure SA](#)
 - [Prerequisites](#)
 - [Post SA configuration](#)
- [Update the default UCMDB password](#)
- [Copy database software binaries](#)
 - [Software and patch binaries](#)

Configure SA

The managed and unprovisioned servers are synced from SA into the UCMDB of DCA using the existing SA-UCMDB integration. The [SA Integration Guide](#) contains detailed information.

 In case of an existing setup, stop the SA-UCMDB integration by running the following commands:

1. `/etc/init.d/opsware-sas stop telldaemon`
2. `/opt/opsware/tell/bin/disable`

Skip this step in case of a new integration.

Prerequisites

 **Note**

These steps must be performed by an SA Administrator.

1. Login to SA as a user.
2. On the SA console, go to **Library**.

3. Go to the **/Opware/Tools/Database & Middleware** folder.
4. Delete the following files (if present):

- dma_oo_client_code_linux.zip.MD5.zip
- dma_oo_client_code_linux.zip
- dma_oo_client_bin_linux.zip.MD5.zip
- dma_oo_client_bin_linux.zip
- dma_oo_client_code_windows.zip.MD5.zip
- dma_oo_client_code_windows.zip
- dma_oo_client_bin_windows.zip.MD5.zip
- dma_oo_client_bin_windows.zip
- dma_oo_client_bin_solaris.zip
- dma_oo_client_code_solaris.zip.MD5.zip
- dma_oo_client_bin_solaris.zip.MD5.zip
- dma_oo_client_code_solaris.zip

To integrate SA with DCA:

1. Open the **mapping.xml** file from the following location in an XML editor: **/etc/opt/opsware/tell/metadata**
2. Replace the file content with the following content:

```
<?xml version='1.0' ?>
<DB-UCMDB-HIGHLEVEL-MAPPING>
<!-- generates installed_software.xml -->
<Model-Definition model-name='sa' enable='true'>
<CI ucmdb-ci-type-name='server_automation_system' enable='true'
base-class='server_automation_system'>
<Attribute source='SA/Description' target-attr='description'
enable='true'/>
<Attribute source='SA/Name' target-attr='name' enable='true'/>
<Attribute-Default target-attr='version' target-attr-
value='10.0' enable='true'/>
</CI>
</Model-Definition>

<!-- generates node.xml -->
<Model-Definition model-name='hosts' enable='true'>
<CI ucmdb-ci-type-name='server_automation_system' reference-
ci='true' enable='true'/>

<CI ucmdb-ci-type-name='ip_address' enable='true' base-
class='node'>
<Attribute source='IpAddress/PrimaryIpName' target-attr='name'
enable='true'/>
<Attribute source='IpAddress/RoutingDomain' target-
```

```

attr='routing_domain' enable='true'/>
</CI>

<CI ucmdb-ci-type-name='node' enable='true' base-class='node'>
<Attribute source='Node/Name' target-attr='name' enable='true'/
>
<Attribute source='Node/Description' target-attr='description'
enable='true'/>
<Attribute source='Node/BiosAssetTag' target-
attr='bios_asset_tag' enable='true'/>
<Attribute source='Node/BiosSerialNumber' target-
attr='serial_number' enable='true'/>
<Attribute source='Node/BiosUuid' target-attr='bios_uuid'
enable='true'/>
<Attribute source='Node/NetBiosName' target-
attr='net_bios_name' enable='true'/>
<Attribute source='Node/NodeModel' target-attr='node_model'
enable='true'/>
<Attribute source='Node/MemorySize' target-attr='memory_size'
enable='true'/>
<Attribute source='Node/OsDescription' target-
attr='os_description' enable='true'/>
<Attribute source='Node/OsFamily' target-attr='os_family'
enable='true'/>
<Attribute source='Node/ExtendedOsFamily' target-
attr='extended_os_family' enable='true'/>
<Attribute source='Node/Vendor' target-attr='vendor'
enable='false'/>
<Attribute source='Node/Node Server Type' target-
attr='host_servertype' enable='true'/>
<Attribute source='IpAddress/ManagementIpName' target-
attr='ip_address' enable='false'/>
<CI-Filter enable='true'><![CDATA[(DEVICES.OPSW_LIFECYCLE =
'MANAGED') or (DEVICES.OPSW_LIFECYCLE = 'UNPROVISIONED')]]></
CI-Filter>
</CI>

<Relation ucmdb-relation-type-name='containment' ucmdb-
relation-from-ci-type-name='node' ucmdb-relation-to-ci-type-
name='ip_address' enable='true' ucmdb-relation-id-link='true'/>
<Relation ucmdb-relation-type-name='aggregation' ucmdb-
relation-from-ci-type-name='server_automation_system' ucmdb-
relation-to-ci-type-name='node' enable='true' ucmdb-relation-
id-link='false'/>
</Model-Definition>

```

```

<Model-Definition model-name='device-groups' enable='true'>
<CI ucmdb-ci-type-name='dca_resource_group' enable='true' base-
class='dca_resource_group'>
<Attribute source='DCA Resource Group/Name' target-attr='name'
enable='true'/>
<Attribute source='DCA Resource Group/Display Label' target-
attr='name' enable='true'/>
<Attribute source='DCA Resource Group/Description' target-
attr='description' enable='true'/>
<!-- <Attribute-Default target-attr='is_read_only' target-attr-
value='true' enable='true'/> -->
</CI>
</Model-Definition>

<Model-Definition model-name='devicegroup-relations'
enable='true'>
<CI ucmdb-ci-type-name='dca_resource_group' base-
class='dca_resource_group' reference-ci='true' enable='true'/>

<CI ucmdb-ci-type-name='node' base-class='node' reference-
ci='true' enable='true'/>

<Relation ucmdb-relation-type-name='aggregation' ucmdb-
relation-from-ci-type-name='dca_resource_group' ucmdb-relation-
to-ci-type-name='node' ucmdb-relation-id-link='true'
enable='true'/>
</Model-Definition>
</DB-UCMBD-HIGHLEVEL-MAPPING>

```

3. Open the **1_node_template.xml** file in an XML editor from the following location: **/opt/opsware/tell/metadata/template**
4. Replace the file content with the following content:

```

<?xml version = "1.0" ?>

<DB-UCMBD-MAPPING>
<Model-Definition model-name='hosts'>

<DB-Query>
<DB-Select-
Clause>DEVICES.DVC_ID,DEVICES.SYSTEM_NAME,DEVICES.DVC_DESC,DEVI
CES.ASSET_TAG,DEVICES.SERIAL_NUM,DEVICES.UUID,DEVICES.DEFAULT_G
W,DEVICES.WINDOWS_NETBIOS_NAME,DEVICES.DVC_MODEL,DEVICES.PRIMAR
Y_IP,MEMORY_COMPONENTS.QUANTITY,PLATFORMS.DISPLAY_NAME,PLATFORM
S.PLATFORM_SHORT_NAME,ACCOUNTS.ACCT_NAME,REALMS.REALM_NAME,DEVI
CES.MANAGEMENT_IP,DATA_CENTERS.DATA_CENTER_NAME,DEVICES.VIRTUAL
IZATION_TYPE_ID,DEVICES.OPSW_LIFECYCLE,DEVICES.DVC_MFG</DB-

```



```

Select-Clause>
<DB-From-Clause>TRUTH.DEVICES left join TRUTH.MEMORY_COMPONENTS
on (DEVICES.DVC_ID = MEMORY_COMPONENTS.DVC_ID) AND
(MEMORY_COMPONENTS.MEMORY_TYPE = 'RAM') left join
TRUTH.PLATFORMS on DEVICES.PLATFORM_ID = PLATFORMS.PLATFORM_ID
left join TRUTH.DEVICE_ROLES on DEVICES.DVC_ID =
TRUTH.DEVICE_ROLES.DVC_ID left join TRUTH.CUSTOMER_CLOUDS on
DEVICE_ROLES.CUST_CLD_ID = CUSTOMER_CLOUDS.CUST_CLD_ID left
join TRUTH.DATA_CENTERS ON CUSTOMER_CLOUDS.DATA_CENTER_ID =
DATA_CENTERS.DATA_CENTER_ID left join TRUTH.ACCOUNTS on
CUSTOMER_CLOUDS.ACCT_ID = ACCOUNTS.ACCT_ID left join
TRUTH.REALMS ON DEVICES.REALM_ID = REALMS.REALM_ID</DB-From-
Clause>
<DB-Query-Primary-key>DEVICES.DVC_ID</DB-Query-Primary-key>
</DB-Query>

<!-- put this maybe in the templates only so when it translate,
the PK can be translated -->
<BASE-CLASS base-class-name='node' db-primary-
key='DEVICES.DVC_ID'>
</BASE-CLASS>

<!-- define the SA and the db to attribute mappings -->

<!-- define CIs and the db to attribute mappings -->
<CI-MAPPING-DEFINITION>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/Name" db-table-
name='TRUTH.DEVICES' db-column-name='DEVICES.SYSTEM_NAME'
ucmdb-attribute-name='name' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/Description" db-
table-name='TRUTH.DEVICES' db-column-name='DEVICES.DVC_DESC'
ucmdb-attribute-name='description' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/BiosAssetTag" db-
table-name='TRUTH.DEVICES' db-column-name='DEVICES.ASSET_TAG'
ucmdb-attribute-name='bios_asset_tag' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/BiosSerialNumber"
db-table-name='TRUTH.DEVICES' db-column-
name='DEVICES.SERIAL_NUM' ucmdb-attribute-name='serial_number'
enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/BiosUuid" db-
table-name='TRUTH.DEVICES' db-column-name='DEVICES.UUID' ucmdb-
attribute-name='bios_uuid' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/
DefaultGatewayIpAddress" db-table-name='TRUTH.DEVICES' db-
column-name='DEVICES.DEFAULT_GW' ucmdb-attribute-
name='default_gateway_ip_address' enable="true"/>

```

```

<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/NetBiosName" db-
table-name='TRUTH.DEVICES' db-column-
name='DEVICES.WINDOWS_NETBIOS_NAME' ucldb-attribute-
name='net_bios_name' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/NodeModel" db-
table-name='TRUTH.DEVICES' db-column-name='DEVICES.DVC_MODEL'
ucldb-attribute-name='node_model' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/Node Server Type"
db-table-name='TRUTH.DEVICES' db-column-
name='DEVICES.OPSW_LIFECYCLE' ucldb-attribute-
name='host_servertype' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/MemorySize" db-
table-name='TRUTH.MEMORY_COMPONENTS' db-column-
name='MEMORY_COMPONENTS.QUANTITY' ucldb-attribute-
name='memory_size' conversion-
name='com.hp.tell.ConversionMethod
$com.hp.tell.ConvertKiloToMega' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/OsDescription"
db-table-name='TRUTH.PLATFORMS' db-column-
name='PLATFORMS.DISPLAY_NAME' ucldb-attribute-
name='os_description' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/OsFamily" db-
table-name='TRUTH.PLATFORMS' db-column-
name='PLATFORMS.PLATFORM_SHORT_NAME' ucldb-attribute-
name='os_family' conversion-name='com.hp.tell.ConversionMethod
$com.hp.tell.ConvertOSFamily' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/ExtendedOsFamily"
db-table-name='TRUTH.PLATFORMS' db-column-
name='PLATFORMS.PLATFORM_SHORT_NAME' ucldb-attribute-
name='extended_os_family' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/TenantOwner" db-
table-name='TRUTH.ACCOUNTS' db-column-name='ACCOUNTS.ACCT_NAME'
ucldb-attribute-name='TenantOwner' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/Facility" db-
table-name='TRUTH.DATA_CENTERS' db-column-
name='DATA_CENTERS.DATA_CENTER_NAME' ucldb-attribute-
name='facility' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/
VirtualizationTypeId" db-table-name='TRUTH.DEVICES' db-column-
name='DEVICES.VIRTUALIZATION_TYPE_ID' ucldb-attribute-
name='virtualization_type_id' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="Node/Vendor" db-table-
name='TRUTH.DEVICES' db-column-name='DEVICES.DVC_MFG' ucldb-
attribute-name='vendor' enable="true"/>

```

```

<ATTRIBUTE-MAPPING-DEFINITION virt-name="IpAddress/
ManagementIpName" db-table-name='TRUTH.DEVICES' db-column-
name='DEVICES.MANAGEMENT_IP' ucldb-attribute-name='ip_address'
enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="IpAddress/
PrimaryIpName" db-table-name='TRUTH.DEVICES' db-column-
name='DEVICES.PRIMARY_IP' ucldb-attribute-name='name'
enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="IpAddress/
RoutingDomain" db-table-name='TRUTH.REALMS' db-column-
name='REALMS.REALM_NAME' ucldb-attribute-name='routing_domain'
enable="true"/>
</CI-MAPPING-DEFINITION>

<!-- define all the model relationships, be sure to only used
CIs defined above -->
<MODEL-RELATION-DEFINITION>
</MODEL-RELATION-DEFINITION>

<Vault-Insert-Trigger>
<Vault-Table-Name db-table-name='TRUTH.DEVICES'>
<Vault-Query-Key-Column-Name db-column-name='DEVICES.DVC_ID' />
</Vault-Table-Name>
</Vault-Insert-Trigger>

<Vault-Update-Trigger>
<Vault-Table-Name db-table-name='TRUTH.DEVICES'>
<Vault-Query-Key-Column-Name db-column-name='DEVICES.DVC_ID' />
</Vault-Table-Name>
</Vault-Update-Trigger>
<Vault-Update-Trigger>
<Vault-Table-Name db-table-name='TRUTH.MEMORY_COMPONENTS'>
<Vault-Query-Key-Column-Name db-column-
name='MEMORY_COMPONENTS.DVC_ID' />
</Vault-Table-Name>
</Vault-Update-Trigger>

<Vault-Update-Trigger>
<Vault-Table-Name db-table-name='TRUTH.DEVICE_ROLES'>
<Vault-Query-Key-Column-Name db-column-
name='DEVICE_ROLES.DVC_ID' />
<Vault-Change-Column-Name db-table-name='TRUTH.DEVICE_ROLES'
db-column-name='DEVICE_ROLES.CUST_CLD_ID' />
</Vault-Table-Name>
</Vault-Update-Trigger>

```

```
</Model-Definition>
</DB-UCMBD-MAPPING>
```

5. Create a new file called **7_devicegroup_template.xml** at **/opt/opsware/tell/metadata/template** as follows:

```
<?xml version = "1.0" ?>
<DB-UCMBD-MAPPING>
<Model-Definition model-name='device-groups'>
<DB-Query>
<DB-Select-Clause>

ROLE_CLASSES.ROLE_CLASS_ID,ROLE_CLASSES.PARENT_ROLE_CLASS_ID,RO
LE_CLASSES.STACK_ID,ROLE_CLASSES.ROLE_CLASS_SHORT_NAME,ROLE_CLA
SSES.ROLE_CLASS_FULL_NAME,ROLE_CLASSES.STATUS

</DB-Select-Clause>

<DB-From-Clause>TRUTH.ROLE_CLASSES</DB-From-Clause>
<DB-Where-Clause>ROLE_CLASSES.STACK_ID = 17 AND
ROLE_CLASSES.STATUS = 'ACTIVE'</DB-Where-Clause>
<DB-Query-Primary-key>ROLE_CLASSES.ROLE_CLASS_ID</DB-Query-
Primary-key>
</DB-Query>

<!--
<BASE-CLASS base-class-name='dca_resource_group_child' db-
primary-key='ROLE_CLASSES.ROLE_CLASS_ID'>
</BASE-CLASS>

<BASE-CLASS base-class-name='dca_resource_group_parent' db-
primary-key='ROLE_CLASSES.PARENT_ROLE_CLASS_ID'>
</BASE-CLASS>
-->
<BASE-CLASS base-class-name='dca_resource_group' db-primary-
key='ROLE_CLASSES.ROLE_CLASS_ID'>
</BASE-CLASS>

<!-- define CIs and the db to attribute mappings -->
<CI-MAPPING-DEFINITION>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="DCA Resource Group/
Name" db-table-name='TRUTH.ROLE_CLASSES' db-column-
name='ROLE_CLASSES.ROLE_CLASS_SHORT_NAME' ucmdb-attribute-
name='name' enable="true"/>
<ATTRIBUTE-MAPPING-DEFINITION virt-name="DCA Resource Group/
Description" db-table-name='TRUTH.ROLE_CLASSES' db-column-
name='ROLE_CLASSES.ROLE_CLASS_FULL_NAME' ucmdb-attribute-
name='description' enable="true"/>
```

```

<ATTRIBUTE-MAPPING-DEFINITION virt-name="DCA Resource Group/
FullyQualifiedName" db-table-name='TRUTH.ROLE_CLASSES' db-
column-name='ROLE_CLASSES.ROLE_CLASS_FULL_NAME' ucmbd-
attribute-name='fully_qualified_name' enable="true"/>
</CI-MAPPING-DEFINITION>

<!-- define all the model relationships, be sure to only used
CIs defined above -->
<MODEL-RELATION-DEFINITION>
</MODEL-RELATION-DEFINITION>

<Vault-Insert-Trigger>
<Vault-Table-Name db-table-name='TRUTH.ROLE_CLASSES'>
<Vault-Query-Key-Column-Name db-column-
name='ROLE_CLASSES.ROLE_CLASS_ID' />
</Vault-Table-Name>
</Vault-Insert-Trigger>

<Vault-Update-Trigger>
<Vault-Table-Name db-table-name='TRUTH.ROLE_CLASSES'>
<Vault-Query-Key-Column-Name db-column-
name='ROLE_CLASSES.PARENT_ROLE_CLASS_ID' />
</Vault-Table-Name>
</Vault-Update-Trigger>

</Model-Definition>
</DB-UCMBD-MAPPING>

```

6. Create a new file called **7_devicegroup_relation_template.xml** at **/opt/opsware/tell/metadata/template** as follows:

```

<?xml version = "1.0" ?>

<DB-UCMBD-MAPPING>
<Model-Definition model-name='devicegroup-relations' model-
depends-on-name='device-groups'>

<DB-Query>
<DB-Select-
Clause>DEVICE_ROLE_CLASSES.DVC_ID,DEVICE_ROLE_CLASSES.ROLE_CLAS
S_ID</DB-Select-Clause>
<DB-From-Clause>TRUTH.DEVICE_ROLE_CLASSES</DB-From-Clause>
<DB-Where-Clause>TRUTH.DEVICE_ROLE_CLASSES.CONFLICTING = 'N'</
DB-Where-Clause>
<DB-Query-Primary-key>TRUTH.DEVICE_ROLE_CLASSES.DVC_ID</DB-
Query-Primary-key>
</DB-Query>

```

```

<BASE-CLASS base-class-name='dca_resource_group' db-primary-
key='DEVICE_ROLE_CLASSES.ROLE_CLASS_ID'>
</BASE-CLASS>

<BASE-CLASS base-class-name='node' db-primary-
key='DEVICE_ROLE_CLASSES.DVC_ID'>
</BASE-CLASS>

<!-- define CIs and the db to attribute mappings -->
<CI-MAPPING-DEFINITION>
</CI-MAPPING-DEFINITION>

<!-- define all the model relationships, be sure to only used
CIs defined above -->
<MODEL-RELATION-DEFINITION>
</MODEL-RELATION-DEFINITION>

<Vault-Insert-Trigger>
<Vault-Table-Name db-table-name='TRUTH.DEVICE_ROLE_CLASSES'>
<Vault-Query-Key-Column-Name db-column-
name='DEVICE_ROLE_CLASSES.ROLE_CLASS_ID' />
</Vault-Table-Name>
</Vault-Insert-Trigger>

<Vault-Update-Trigger>
<Vault-Table-Name db-table-name='TRUTH.DEVICE_ROLE_CLASSES'>
<Vault-Query-Key-Column-Name db-column-
name='DEVICE_ROLE_CLASSES.DVC_ID' />
</Vault-Table-Name>
</Vault-Update-Trigger>

</Model-Definition>
</DB-UCMDB-MAPPING>

```


7. Start the sync process by running the following command:

```

/opt/opsware/tell/bin/enable --host <ip> --port 33071 --protocol
http --user <username> --password <password>
/etc/init.d/opsware-sas start telldaemon

```

Where host *<ip>* is the IP address of the DCA server and *<username>* is the UCMDB user name and *<password>* is the UCMDB password.

 In a multimaster SA installation, the SA-UCMDB integration is configured on one server. See the [SA Integration Guide](#) for more information on the server from the multimaster server installation that can be selected for the integration.

Post SA configuration

The SA Administrator must perform these steps after the DCA client binaries are successfully uploaded to SA.

1. Launch `https://<dca-hostname>:33445`. Login with your OO user credentials.
 - a. Go to **Run Management > Flow Launcher**.
 - b. In the Flow Launcher area, go to **Integrations > Hewlett-Packard-Enterprise > Data Center Automation > Utilities > SA Import Package**.
 - c. Click **Run** to run the workflow.

2. Select the following files in the **/Opware/Tools/Database & Middleware** folder in the **SA Library**:
 - `dma_oo_client_code_linux.zip.MD5.zip`
 - `dma_oo_client_code_linux.zip`
 - `dma_oo_client_bin_linux.zip.MD5.zip`
 - `dma_oo_client_bin_linux.zip`
 - a. Double click on file name from the SA repository and set the default install path to: **/opt/hp/dma/ooclient**
 - b. Set the OS to **Red Hat, SUSE, UBUNTU, HP-UX, Oracle Linux CentOS**.

2. Select the following files in the **/Opware/Tools/Database & Middleware** folder in the **SA Library**:
 - `dma_oo_client_code_windows.zip.MD5.zip`
 - `dma_oo_client_code_windows.zip`
 - `dma_oo_client_bin_windows.zip.MD5.zip`
 - `dma_oo_client_bin_windows.zip`
 - a. Double click on file name from the SA repository and set the default install path to: **C:\Program Files\HP\DMA\OOClient**
 - b. Set the OS to **Windows**.

3. Select the following files in the **/Opware/Tools/Database & Middleware** folder in the **SA Library**:
 - `dma_oo_client_bin_solaris.zip`
 - `dma_oo_client_code_solaris.zip.MD5.zip`
 - `dma_oo_client_bin_solaris.zip.MD5.zip`
 - `dma_oo_client_code_solaris.zip`
 - a. Double click on file name from the SA repository and set the default install path to: **/opt/hp/dma/ooclient**
 - b. Set the OS to **Solaris**.

Update the default UCMDB password

It is recommended that you change the default UCMDB password. For information about changing default passwords, see [Change the default UCMDB password](#).

Copy database software binaries

You must copy the third-party vendor-provided binaries to ensure that database software applications such as Oracle, Microsoft SQL Server are successfully installed.

1. Go to the export path that you created when [setting up the NFS server](#). For example, /vols/dca
2. Go to the vendor-binaries directory that is present in the path. For example, /vols/dca/vendor-binaries/
3. Copy all the database software and patch binaries to the vendor-binaries directory. See [Software and patch binaries](#) for a list of all the database software and patch binaries to be copied.

Software and patch binaries

DCA requires the software and patches for patching and provisioning of databases.

Download the following Oracle software and patches from the Oracle Downloads page: <https://www.oracle.com/downloads/index.html>

- Oracle 11g
 - p13390677_112040_Linux-x86-64_1of7.zip
 - p13390677_112040_Linux-x86-64_2of7.zip
 - p13390677_112040_Linux-x86-64_3of7.zip
 - p13390677_112040_Linux-x86-64_4of7.zip
 - p23054359_112040_Linux-x86-64.zip
 - p6880880_112000_Linux-x86-64.zip
 - p24006111_112040_Linux-x86-64.zip
- Oracle 12c
 - linuxamd64_12102_database_1of2.zip
 - linuxamd64_12102_database_2of2.zip
 - p6880880_121010_Linux-x86-64.zip
 - p19769480_121020_Linux-x86-64.zip
 - p21948354_121020_Linux-x86-64.zip

Download the following MSSQL software and patches from the Download Center: <https://www.microsoft.com/en-us/download>

- Microsoft SQL Server 2008
 - SQL08R2-Enterprise.zip
 - SQLServer2008R2SP1-KB2528583-x64-ENU.exe
 - SQLServer2008R2SP2-KB2630458-x64-ENU.exe
 - SQLServer2008R2SP3-KB2979597-x64-ENU.exe

- Microsoft SQL Server 2012
 - SQL12.zip

- Microsoft SQL Server 2014
 - SQL14.zip
 - SQLServer2014SP2-KB3171021-x64-ENU.exe
 - SQLServer2014-KB3188778-x64.exe

Where to go from here

[Use](#)

Uninstall

To remove DCA:

1. [Back up image files](#)
2. [Uninstall ITOM CDF](#)
3. [Uninstall DCA](#)

Back up image files



Note

This step is optional.

You can back up the image files from the local private registry to a remote registry before you uninstall ITOM CDF.

1. Go to the directory where the **local_backup.sh** file is located:
cd <installation-folder>/script
2. Move the **jq** file to the **/usr/local/bin/** directory using the following commands:
chmod 777 jq
mv jq /usr/local/bin

3. Run the following command: **chmod 777 local_backup.sh**
Ensure that the script file format is UNIX.
4. Run the following command: **./local_backup.sh <registryHost>**
For example:
./local_backup.sh 10.10.10.10:5000
The TAR files are saved in **image_tars/xxx.tar**.

Uninstall ITOM CDF

To uninstall ITOM CDF:

1. Run the `uninstall.sh` script. The uninstall process stops containers and removes containers and daemons.
bash ./uninstall.sh
2. Restart the DCA server.

Uninstall DCA

1. [Uninstall ITOM CDF](#).
2. Remove directories from NFS or remove the NFS configuration from the DCA server.

Related topic

[Install ITOM CDF](#)

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email.

Add the following information in the subject line: Feedback on Data Center Automation 2017.05 - Premium

Just add your feedback to the email and send your feedback to docs.feedback@hpe.com.

We appreciate your feedback.