



Get started

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	Get started	7
3	Related topics	7
4	Overview of DCA.....	7
4.1	General architecture	7
4.2	Detailed architecture	8
4.3	The DCA console.....	10
4.3.1	Header	11
4.3.2	Navigation bar.....	12
4.3.3	Dashboard	12
4.3.3.1	Related topics	14
4.4	Author	14
4.5	Resource Management.....	16
4.6	Settings.....	18
5	Personas	19
5.1	IT Administrator	19
5.2	Suite Administrator.....	19
5.2.1	Related topics	20
6	Use cases	20
6.1	Import and discovery of resources.....	22
6.2	Provision and configure infrastructure.....	22
6.2.1	Sample provisioning use cases	23
6.2.1.1	Provisioning the Linux OS.....	23
6.2.1.2	Provisioning a database	24
6.2.1.3	Provision a Docker host.....	25
6.3	Ensure compliance of managed resources	25

6.3.1	Policy-based compliance scan	26
6.3.2	Ad hoc compliance scan	27
6.3.3	Sample use cases	27
6.3.3.1	Perform policy-based compliance scan	27
6.3.3.2	Perform ad hoc compliance scan.....	27
6.4	Perform tasks in DCA using ChatOps	28
6.5	Generate reports	28
7	Overview of the ITOM Container Deployment Foundation	29
7.1	New and agile software delivery platform.....	29
7.2	Architecture.....	30
7.3	The ITOM CDF console	31
7.3.1	SUITE	31
7.3.2	ADMINISTRATION	33
7.3.3	RESOURCES	33
7.3.3.1	Related topics	34
8	Glossary	34
8.1	-A-	34
8.1.1	Authentication	34
8.1.2	Authorization.....	34
8.2	-B-.....	35
8.2.1	Benchmark	35
8.3	-C-	35
8.3.1	ChatOps	35
8.3.2	Cloud Optimizer (CO)	35
8.3.3	Cluster.....	35
8.3.4	Config Map.....	36
8.3.5	Containers	36
8.3.6	Controls	36
8.4	-D-.....	37
8.4.1	Daemon sets.....	37
8.4.2	Database Middleware Automation	37
8.4.3	Docker.....	37
8.5	-I-	38

8.5.1	Identity Management.....	38
8.5.2	Ingress	38
8.6	-K-.....	38
8.6.1	Kubernetes.....	38
8.7	-L-	38
8.7.1	Label	38
8.7.2	Lightweight single sign-on (LWSSO)	39
8.8	-M-	39
8.8.1	Master node.....	39
8.9	-N-	39
8.9.1	Namespaces	39
8.10	-O-	40
8.10.1	Operations Bridge Reporter	40
8.11	-P-.....	40
8.11.1	Persistent volume	40
8.11.2	Pet set.....	40
8.11.3	Pod.....	40
8.11.4	Policy	41
8.11.5	Provisioning	41
8.11.6	-R-.....	41
8.11.6.1	Replica sets.....	41
8.11.6.2	Replication Controller	41
8.11.6.3	Resource types	42
8.11.6.4	Resources	42
8.11.7	-S-	42
8.11.7.1	Secret.....	42
8.11.7.2	Server Automation.....	42
8.11.7.3	Service	42
8.11.7.4	Slack	43
8.11.8	-T-.....	43
8.11.8.1	Tag.....	43
8.11.8.2	Template	43
8.11.9	-W-.....	43
8.11.9.1	Worker node.....	43
9	Send documentation feedback.....	44

Get started

Read this section to get started with DCA. Topics in this section provide you with an overview of DCA, key concepts, and use cases. You can also learn more about the ITOM Container Deployment Framework (CDF)—the installation and administration utility that helps you install and maintain DCA with ease.

Explore the following topics in this section to develop a well-rounded understanding of DCA before moving on to perform advanced tasks:

- To know about the basic architecture of DCA and its components, see [Overview of DCA](#).
- To read about key features available with this version of DCA, see [Author, Resource Management, and Settings](#).
- To understand how to use DCA to automate daily tasks at a data center, see [Use cases](#).
- To familiarize yourself with terminologies used in this library, see [Glossary](#).
- To know about the technology used to facilitate easy installation of DCA, see [Overview of the ITOM Container Deployment Foundation](#).

Related topics

[Overview of DCA](#)

[Use cases](#)

Overview of DCA

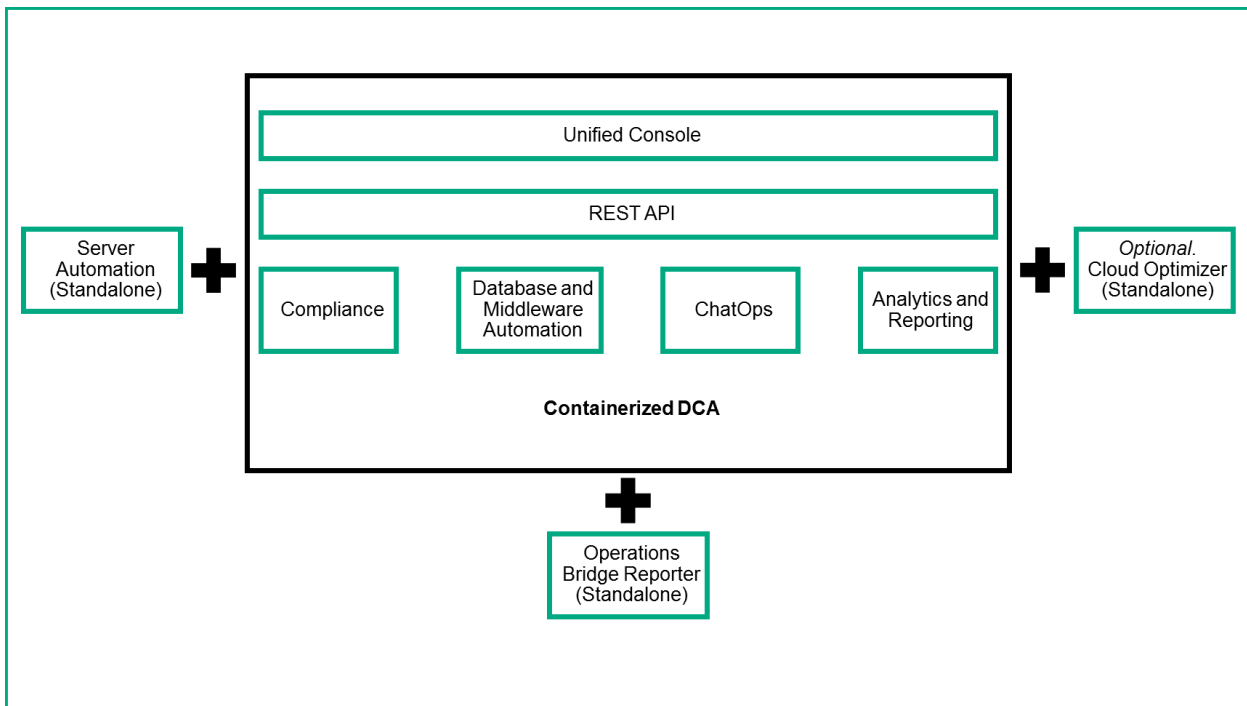
DCA helps you automate provisioning, compliance audit, and remediation of operating systems, Docker containers, databases, and middleware in a heterogeneous data center. Powered by an intuitive user interface and open APIs, DCA can simplify your data center management tasks by:

- Automating mundane, time-consuming provisioning jobs and deployments
- Running periodic, policy-driven compliance scans
- Showing reports on compliance scan results

The container-based installation of DCA helps you realize quick time-to-value; the ChatOps capability enables you to check in on active deployments and provisioning tasks at any time from the comfort of your home.

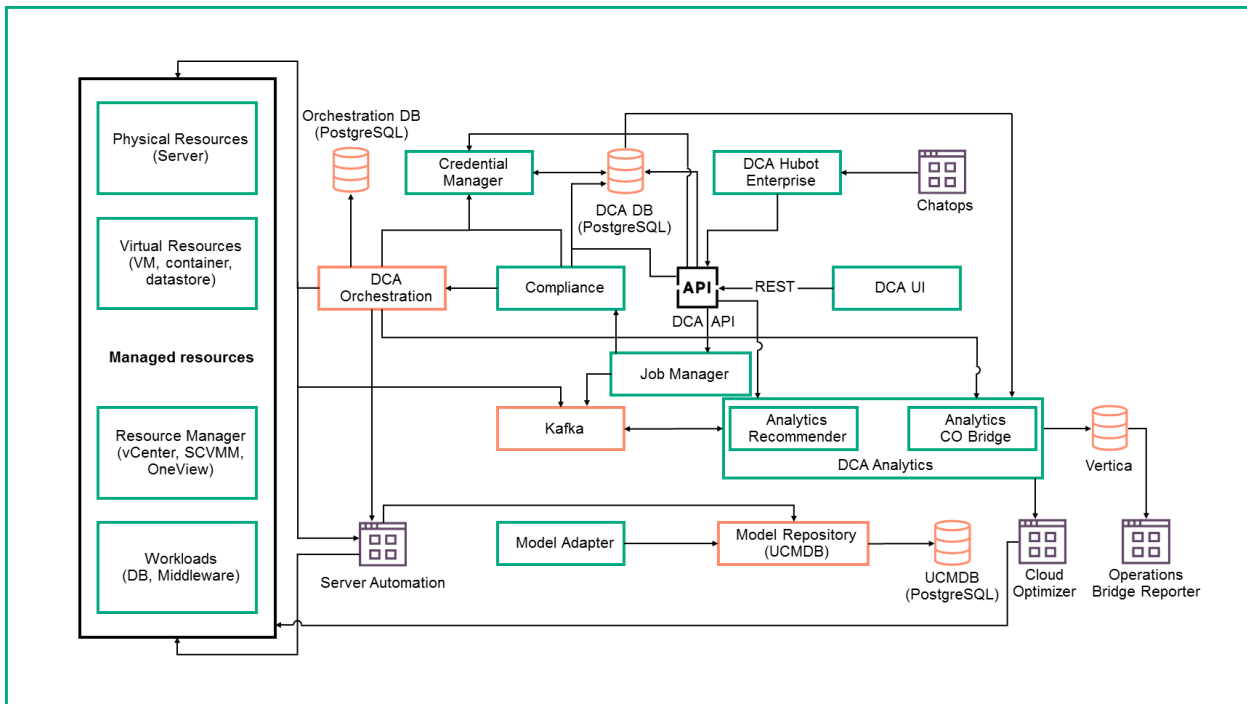
General architecture

The general architecture diagrams provide an overview of the components that comprise DCA.



Detailed architecture

The detailed architectural diagram of DCA is as follows:



DCA comprises of the following components:

Component	Description
Analytics	This component will provide log analytics and visual analytics for DCA.
Analytics CO Bridge	Connector to Cloud Optimizer.
Vertica (Experimental)	Vertica server that stores the time series metric and log data of DCA.
ChatOps	Hubot Enterprise-based container with scripts for dca-bot.
Cloud Optimizer	Server monitoring component of DCA. This will also provide placement services for virtual machines.
Compliance	Allows you to view the overall compliance levels for all servers and groups of servers in your facility. From the Compliance dashboard, you can remediate resources that are out of compliance.
Content container	<p>A content container is a Docker image that is a part of DCA and runs as a job within the DCA installation. The content container contains the content pack artifacts that are used by different services in DCA. For example, compliance service, provisioning service, and OO service.</p> <p>During the DCA installation, the content container runs as a job and creates the following directories:</p> <ul style="list-style-type: none"> • /dcafileshare/importScripts: Contains the scripts to import content artifacts of various DCA services • /dcafileshare/dca: Contain the content artifacts related to various DCA services • /dcafileshare/content: Contain the content artifacts related to various DCA services • /dcafileshare/logs: Contains a log file for each of the scripts in the ImportScripts folder
Credential Manager	Manages credentials for managed resources.
DCA API	External-facing REST APIs that conform to the REST API guidelines. These APIs are built on the resource model that DCA exposes to its users. DCA console interacts with the backend through these APIs.
DCA UI	Grommet-based user interface that is the facade of DCA.
Analytics Recommender	Provides various recommendations for DCA. For example, target recommendation for deployment and benchmark recommendation for templates.
Kafka	Message bus for DCA.

Model Adapter	Listens to topology updates on Kafka and updates them on the Model Repository (uCMDB).
Model Repository	A no-frills container carved out of uCMDB that will work as the single source of truth for all our inventory and configuration data. This comes without any of the Discovery components of uCMDB. The components that are part of DCA form the source of inventory data to this repository.
Operation Bridge Reporter (OBR)	Data warehousing component in DCA that is used for reporting.
DCA Orchestration	Component that creates structured sequences called workflows.
PostgreSQL server	DCA contains the following PostgreSQL servers for use with the following: <ul style="list-style-type: none"> • All DCA components • DCA Orchestration • Model Repository (uCMDB)
Server Automation	Server Automation (SA) automates tasks like provisioning, patching, configuring, and release management of servers and application servers.
Zookeeper	Configuration subsystem for Kafka.

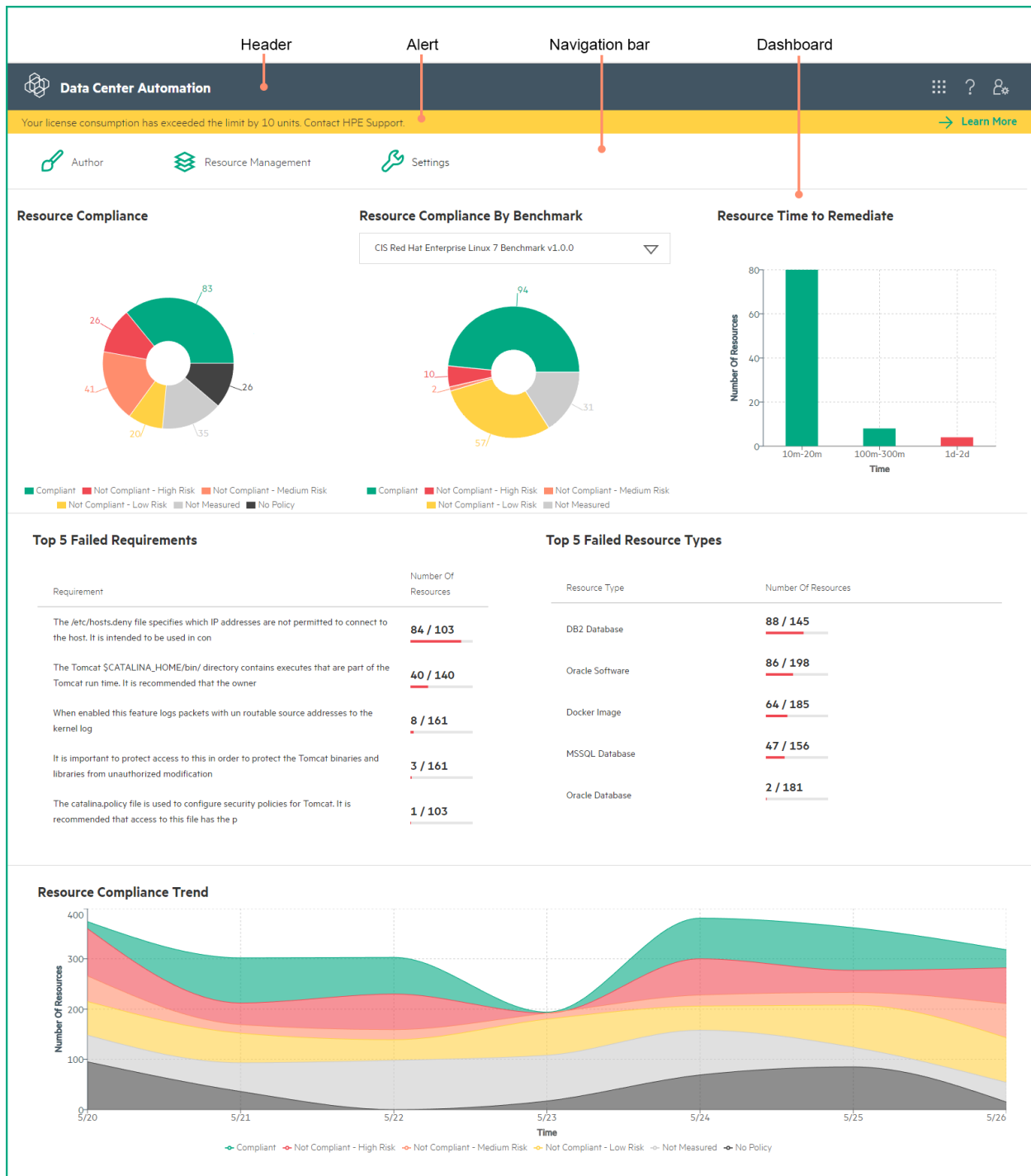
The DCA console

The console is the graphical user interface (GUI) of DCA. It presents a clean and unified overview of the key operational metrics obtained from multiple sources within your data center. The DCA console provides you the following capabilities:

- Advanced management capabilities
- Ease of navigation across the DCA portfolio
- Drill-down capabilities to view and analyze detailed compliance data on demand
- Responsive layout for a seamless user experience across desktops, laptops, and mobile devices





See [Use](#) to implement all the functions that the DCA console offers.

The primary components of the DCA console are as follows:




Header

The header incorporates the following elements:

UI element	Description
	Takes you to the default dashboard.
	<p>Displays the navigation icons:</p> <ul style="list-style-type: none"> • Author: Takes you to the Author page. • Resource Management: Takes you to the Resource Management page. • Settings: Takes you to the Settings page. <p>The navigation icons are also available on the Navigation Bar.</p>
	<p>Displays the following options:</p> <ul style="list-style-type: none"> • Logout: Logs you out of an active session.
	Launches the DCA documentation home page.

Navigation bar

The navigation bar, available on the DCA console, provides a quick means of navigation. You can also use the navigation icon  available on every page to navigate across the DCA portfolio. The navigation bar incorporates the following elements:

UI element	Description
Author	Takes you to the Author page.
Resource Management	Takes you to the Resource Management page.
Settings	Takes you to the Settings page.


Dashboard

Dashboard helps you assess the following:

- Level of compliance in your data center
- Efficiency of your remediation process
- Compliance failures across the data center

Label	Description
Resource Compliance chart	<p>The Resource Compliance chart shows the compliance states of all the resources in the data center in color-coded segments in a pie chart. Possible states are:</p> <ul style="list-style-type: none"> • Compliant • Not Compliant - High Risk • Not Compliant - Medium Risk • Not Compliant - Low Risk • Not Measured <p>Click on each segment to view more details of the resources represented by the segment.</p>
Resource Compliance by Benchmark chart	<p>The Resource Compliance by Benchmark chart shows the compliance states of all the resources for a specific benchmark in color-coded segments in a pie chart. Possible states are:</p> <ul style="list-style-type: none"> • Compliant • Not Compliant - High Risk • Not Compliant - Medium Risk • Not Compliant - Low Risk • Not Measured <p>Click on each segment to view more details of the resources represented by the segment.</p>
Resource Time to Remediate chart	<p>This chart helps you assess the approximate time taken to remediate the resources that failed the compliance scan.</p> <p>On this chart, DCA represents data gathered from the latest compliance scan and remediation in the form of vertical bars—each indicating the number of resources that required the same time duration for remediation.</p> <p>You can click on each bar to see more information about the resources.</p>
Top 5 Failed Requirements report	<p>This report shows top 5 failed requirements in the managed data center in a top-n-style table.</p> <p>Each row of the table represents a requirement and shows the following details:</p> <ul style="list-style-type: none"> • Requirement name • The total number of resources associated with the requirement that failed the compliance scan • The total number of resources associated with requirement <p>Click on a requirement on this report to view more details of the resources that are associated with the requirement.</p>

<p>Top 5 Failed Resource Types</p>	<p>This report shows top 5 failed resource types in a top-n-style table. Each row of the table represents a resource type and shows the total number of resources that failed the compliance test.</p> <p>Click on a resource type on this report to view more details of the resources of this type that failed the compliance test.</p>
<p>Resource Compliance Trend Analysis chart</p>	<p>This chart represents the compliance trend of resources over the past seven days. The chart shows the compliance states of resources in your data center in color-coded segments in area graphs. Possible states are:</p> <ul style="list-style-type: none"> • Compliant • Not Compliant - High Risk • Not Compliant - Medium Risk • Not Compliant - Low Risk • Not Measured

 **Note**

Results of ad hoc compliance scans are not shown on the dashboard.

Related topics

[Personas](#)

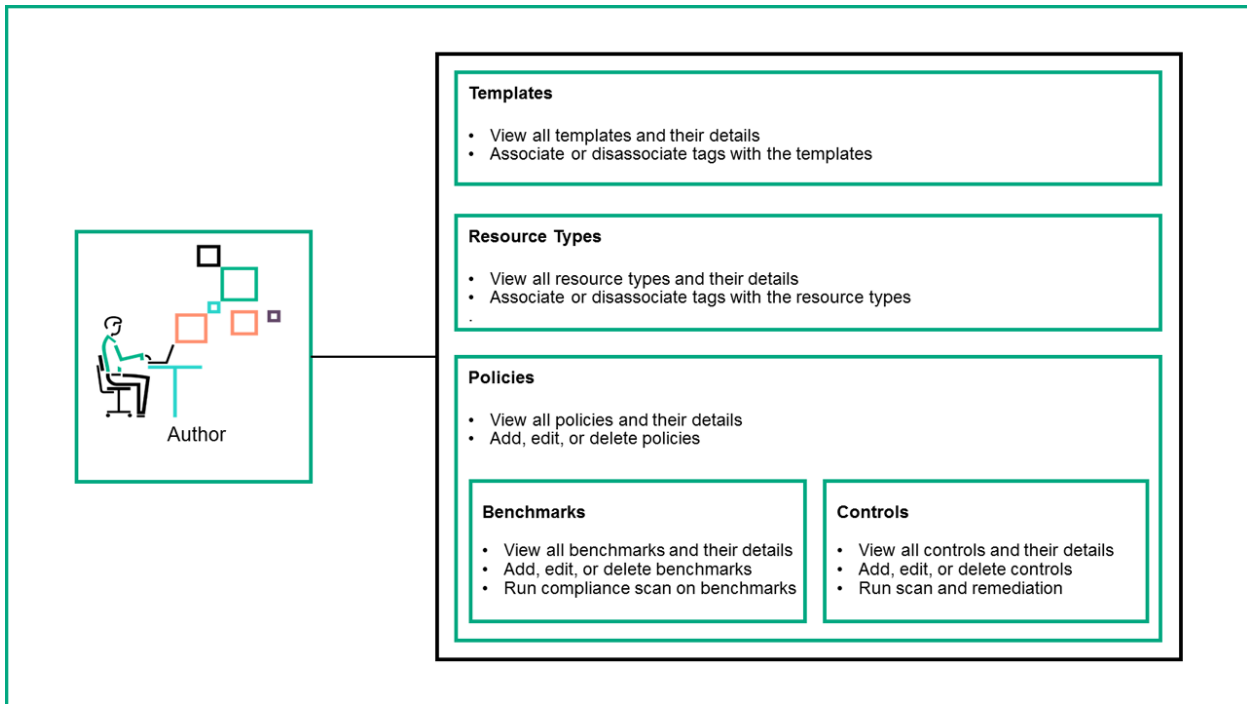
[Use cases](#)

[Glossary](#)

Author

The **Author** page enables you to view the list of available resource types, templates, and benchmarks, under the related tabs. You can sort and filter the content in the lists based on the options available for each tab. All objects within the lists have drill-down capabilities, thereby enabling you to view their details.

The following diagram represents the Author workflow:



The **Author** page displays the following tabs:

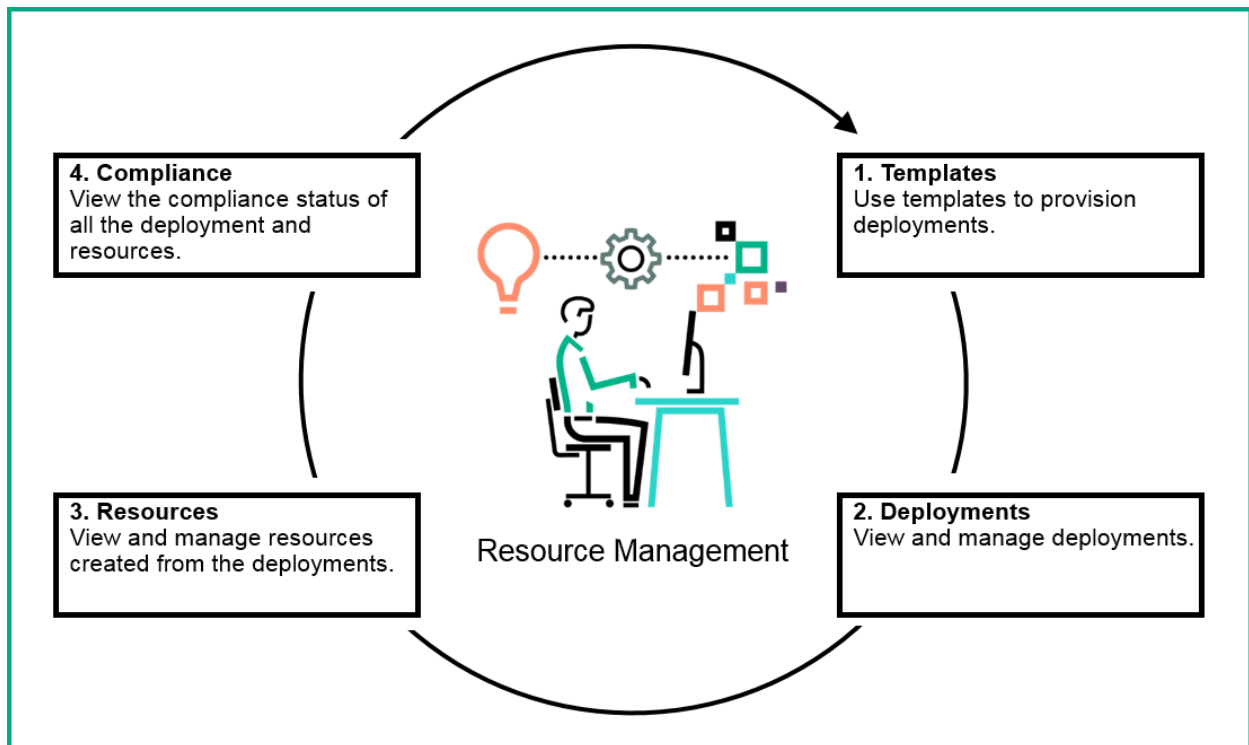
Tab	Allowed actions
Resource Types	View all resource types and their details
	Filter and sort resource types
	Search for a resource type
	Associate or disassociate tags with resource types
Templates	View all templates and their details
	Sort templates
	Search for a template
	Associate or disassociate tags with resource types
Controls	View all controls and their details
	Filter and sort controls
	Search for controls
	Edit controls
	Add new controls
	Delete controls

Benchmarks	View all benchmarks and their details
	Sort benchmarks
	Search for benchmarks
	Add a new benchmark
	Edit a benchmark
	Clone an existing benchmark
	Delete a benchmark
	Run a compliance scan for benchmark
Policies	View all policies and their details
	Search for a policy
	Sort policies
	Add a new policy
	Edit a policy
	Associate a benchmark to an existing policy
	Disassociate a benchmark from an existing policy
	Delete a policy
	Scan a resource attached to a policy
	Remediate a resource attached to a policy

Resource Management

The **Resource Management** page enables you to deploy resources, manage unallocated resources, and monitor resources for availability.

The following diagram represents the Resource Management workflow:



The **Resource Management** page displays the following tabs:

Tab	Allowed actions
Templates	Deploy templates
	View templates used for deployments
	Sort templates
	View template details used for deployments
Deployments	View all deployments
	Filter and sort deployments
	Search for a deployment
	View deployment details
	Run a scan compliance job
	Run a remediation job
	Change template associated with a deployment
Resources	Import resources

	Discover contained resources
	View resources
	Power off virtual machines
	Associate a template with a resource
	Associate a credential with a resource
	Run compliance scan for a resource
	View and filter compliance scan results
	Remediate benchmark rules in a resource
	View compliance scan or remediation job status
	View compliance scan results
Resource groups	View all resource groups
	Create a resource group
	Set a maintenance schedule to the resource group
	Attach a policy to the resource group
Compliance	View compliance status of all resources and benchmarks
	Filter and sort objects by compliance status
	Search for an object by compliance status
	View compliance details
Activity	View and filter jobs
	View and filter compliance scan status
	Remediate a resource

Settings

The **Settings** page enables you to manage credentials for the resources in your infrastructure. Every credential is assigned with a credential ID, which can be used for credential mapping of the resources during the discovery process.

The **Settings** page displays the following tabs:

Tab	Allowed actions
Credentials	View all credentials
	Add a credential
	Edit a credential
	Delete a credential

Personas

IT Administrator

The IT Administrator is a super administrator. This user has ability to request or add resources and has wide access permissions.

The IT Administrator can perform the following functions:

- Manage the ITOM CDF and all suite products
- Manage the grow/shrink functions
- Add and remove working nodes (machines).

See [Overview of the ITOM Container Deployment Foundation](#) to implement all the functions that the ITOM CDF console offers.

Suite Administrator

The Suite Administrator can manage the specific suite product, in this case, DCA. DCA Suite Administrators are typically tasked with addressing the challenges in provisioning software resources on a multitude of hardware units in a data center. They are often required to audit and maintain compliance across data centers by enforcing Service-Level Objectives (SLOs).

The Suite Administrator does not have access to the [SUITE](#) and [ADMINISTRATION](#) menus on the [ITOM CDF console](#). This user can only access the [RESOURCES](#) menu and all the child menus under it, including Namespaces, Workloads, Service and discovery, Persistent Volume Claims, Configuration and all the relevant subsidiaries.

A DCA Suite Administrator is interested in:

- Deploying a wide range of operating systems on a variety of hardware platforms
- Performing compliance audits and publishing audit results
- Viewing the results of compliance scans in the forms of graphs and reports
- Automatically remediating non-compliant resources in data centers

All of these goals can be achieved by different tools and menus in the DCA console. See [Use](#) to implement all the functions that the DCA console offers.

Related topics

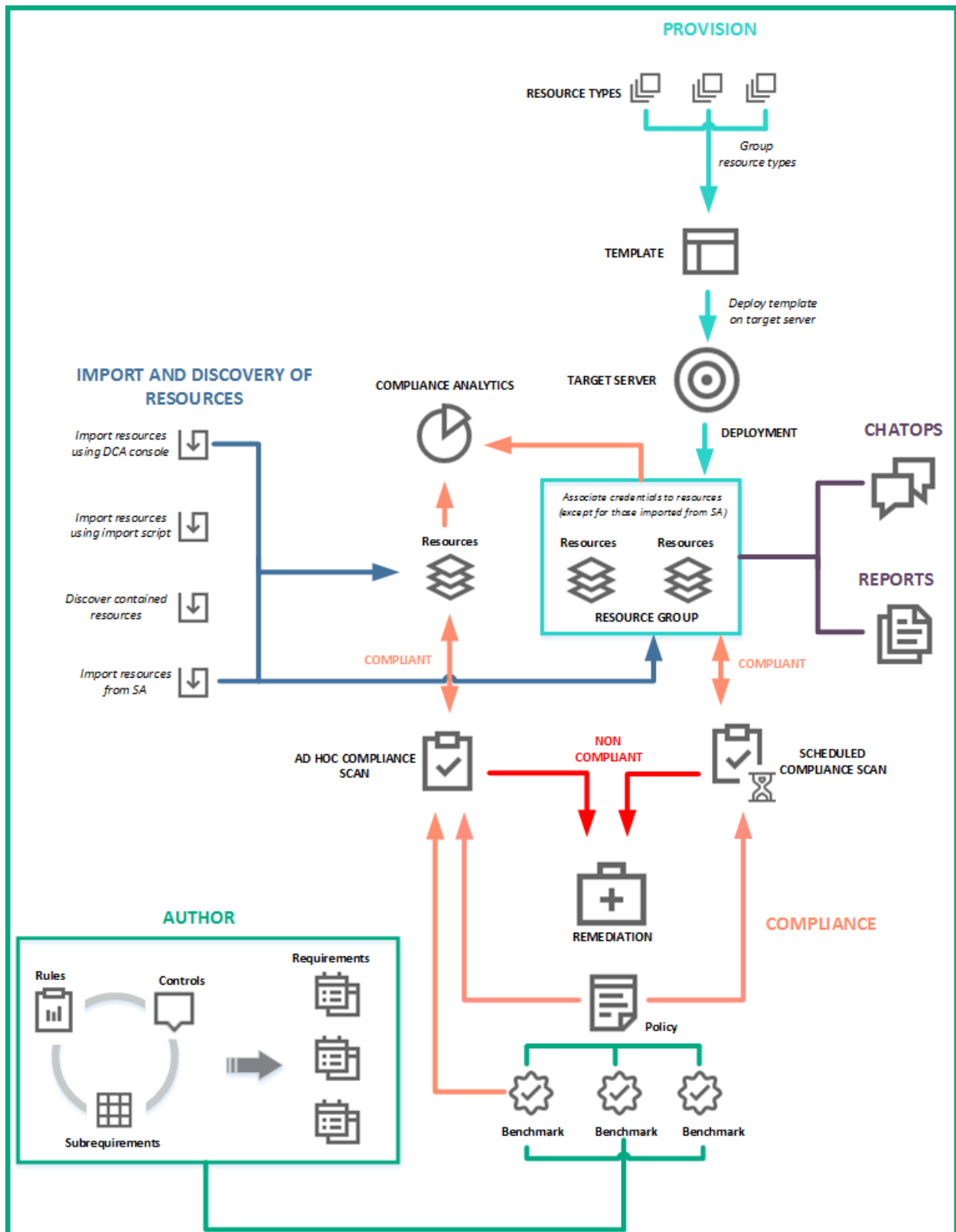
[Use cases](#)

[Provision and configure infrastructure](#)

[Ensure compliance of managed resources](#)

Use cases

DCA enables the automation of several critical processes in the data center environment. The following illustration provides a summary of all the point use cases that are available in DCA:



The following sections provide detailed information about each of the use cases:

- [Import and discovery of resources](#)
- [Provision and configure infrastructure](#)
- [Ensure compliance for managed resources](#)
- [Perform tasks in DCA using ChatOps](#)
- [Generate reports](#)

Import and discovery of resources

A [resource](#) is a compute or a software resource that should fall under the management of DCA.

A **discovered resource** is one that is imported by DCA but not attached to any template. A discovered resource can be associated with a template to convert it into a managed resource. A **managed resource** is one that is deployed using an infrastructure template, or a discovered resource that is associated with a template. A **contained resource** is an application that is managed by DCA, which may or may not be provisioned. Resources are grouped together to create [resource groups](#). To be able to associate resource groups with compliance policies, they will first need to be discovered within DCA or imported into DCA from SA.

The following steps will enable you to import and discover resources in DCA:

1. [Import resources](#)
2. [Discover contained resources](#)

Provision and configure infrastructure

Provisioning is the process of deploying resources on the target server using factory-integrated templates. Provisioning begins with a user-initiated action of selecting a template for resource deployment. DCA then runs a pre-check to ensure that the selected resource is not provisioned. A successful pre-check is followed by the installation and configuration of the software (OS, DB, middleware, and applications) based on the deployment parameters. As the final step in provisioning, DCA runs post-checks on individual resources to verify the successful installation, configuration, and compliance states of the resources.

Note

Before you perform provisioning in DCA, see [SA Provisioning](#) to understand how resources must be added to SA. Following this, [Configure SA](#) to integrate SA with DCA.

As a DCA user, you can provision operating systems and databases. The process of provisioning involves the following steps:

1. [Import resources](#)
2. [Associate a credential with the target server](#)
3. [Verify target system requirements](#)
4. [Copy database of software binaries](#)
5. [Discover contained resources](#)
6. [Select a template to deploy on a target server](#)
7. [Deploy the template on the target server](#)

To perform patching, [change the template](#) associated with a deployment.

Sample provisioning use cases

Provisioning the Linux OS

Following is a sample procedure to provision the Linux OS on the target server:

1. Login to the DCA console.
2. Add credentials of the target system. Copy the credential ID.
3. Rename the sample_resources.csv under the `/var/vols/itom/dca/dca-dca01/content/dca/api/resourceDiscovery` folder in the host VM to resources.csv.
4. Add the target system information and add the credential ID to the column “credential_id” in the resources.csv file.
5. Edit the server.properties file under the `/var/vols/itom/dca/content/dca/api/resourceDiscovery` folder in the host VM to add the IP address of the DCA host VM.
For example:
UCMDB_HOST_IPADDRESS=**10.0.0.0**
UCMDB_EXTERNAL_PORT=33071
UCMDB_USERNAME=<administrator user name>
UCMDB_PASSWORD=<administrator password>
IDM_HOST_IPADDRESS=**10.0.0.0**
IDM_EXTERNAL_PORT=33443
DCA_API_HOST_IPADDRESS=**10.0.0.0**
DCA_API_EXTERNAL_PORT=33444
6. Discover resources.
7. On the DCA console, select **Resource Management > Templates**.
8. Select the **Linux OS** template. The following information is displayed:
 - **Resource Types:** The type definition of a resource that has a set of predefined attributes. In this case, the Linux template will have a Linux resource type.
 - **Recent Deployments:** Lists the most recent deployments made using this template.
9. Click **Deploy**.
10. In the **Deploy** dialog box:
 - Provide a name for the deployment in the **Name** field.
 - Provide an appropriate description for the deployment in the **Description** field.
 - In **Resource Types**, click the edit icon. In this case, Linux is the OS which will be installed on the unprovisioned server. This resource requires the following deployment parameters as inputs from the user:
 - **Target:** A list of options displays the unprovisioned servers from SA. Select a server to be used as a target during OS deployment.
 - **Build Plan ID:** Obtain the ID corresponding to the desired Linux OS from SA and insert it in this location.

- **Credential:** A list of options displays the names of the existing credentials from Credential Manager. The username and password associated to the selected credential will be used to login to the server after the OS is installed.

11. Click **Deploy**.

The result of the deployment process is a managed server, with the Linux OS installed. You can locate this server by navigating to **Resource Management > Resources**. Alternatively, the server can also be found in SA under Managed Servers.

Note: Employ the same procedure to deploy the Windows OS on a target server by selecting the **Windows OS** template in **Resource Management > Templates**.

Provisioning a database


Following is a sample procedure to provision a database on the target server:

1. Login to the DCA console.
2. Add credentials of the target system. Copy the credential ID.
3. Rename the sample_resources.csv under the **/var/vols/itom/dca/dca-dca01/content/dca/api/resourceDiscovery** folder in the host VM to resources.csv.
4. Add the target system information and add the credential ID to the column “credential_id” in the resources.csv file.
5. Edit the server.properties file under the **/var/vols/itom/dca/content/dca/api/resourceDiscovery** folder in the host VM to add the IP address of the DCA host VM.
For example:
UCMDB_HOST_IPADDRESS=**10.0.0.0**
UCMDB_EXTERNAL_PORT=33071
UCMDB_USERNAME=<administrator user name>
UCMDB_PASSWORD=<administrator password>
IDM_HOST_IPADDRESS=**10.0.0.0**
IDM_EXTERNAL_PORT=33443
DCA_API_HOST_IPADDRESS=**10.0.0.0**
DCA_API_EXTERNAL_PORT=33444
6. Discover resources by running the shell script **/var/vols/itom/dca/dca-dca01/content/dca/api/resourceDiscovery/discover_resources.sh**.
7. Check if the credential is mapped correctly for the newly added resource. If it is not mapped, edit the credentials and map the appropriate credential.
8. Add DCA host VM credentials in OO.
9. Navigate to the **Resource Management > Templates** tab, select a template and click **Deploy**.
10. In the **Deploy** dialog box:
 - Provide a name for the deployment in the **Name** field.
 - Provide an appropriate description for the deployment in the **Description** field.
 - In **Resource Types**, click the edit icon for each resource type.
 - Depending on the resource type, specify the deployment parameters.
11. Click **Deploy**.
12. View the deployment status under the **Deployment** tab.

Provision a Docker host

Following is a sample procedure to provision a Docker host on a target server running Linux OS:

1. [Log in](#) to the DCA console.
2. [Add a credential](#) for the target server. Copy the credential ID.
3. On the DCA server, go to `/var/vols/itom/dca/dca-dca01/content/dca/api/resourceDiscovery`.
4. Rename the `sample_resources.csv` file to `resources.csv`.
5. Edit the `resources.csv` file to add the UCMDB CI type attributes of the target server. Add the credential ID under the header “`credential_id`”.
6. Edit the `server.properties` file to add the required properties.
For example:

```
UCMDB_HOST_IPADDRESS=10.0.0.0
UCMDB_EXTERNAL_PORT=33071
UCMDB_USERNAME=<administrator user name>
UCMDB_PASSWORD=<administrator password>
IDM_HOST_IPADDRESS=10.0.0.0
IDM_EXTERNAL_PORT=33443
DCA_API_HOST_IPADDRESS=10.0.0.0
DCA_API_EXTERNAL_PORT=33444
```
7. To discover resources, run the command **`bash ./discover_resources.sh`**.
8. Check if the credential is mapped correctly for the newly added resource. If it is not mapped, [edit the credential](#) and [map the appropriate credential](#).
9. Add the DCA server credentials to OO.
10. On the DCA console, go to **Resource Management > Templates**.
11. Click **CIS Compliant Docker Host** and then click **Deploy**.
12. On the **Deploy** page:
 - a. Enter a name and description for the Docker host.
 - b. Under **Resource Types**, click  and then select the target server.
 - c. Enter the correct Docker version. DCA supports Docker versions 1.6, 1.7, 1.8, and 1.11.
 - d. Click **OK**.
 - e. Click **Deploy**. After successful deployment, DCA runs a compliance scan on the created resources.
13. To view the compliance states and other details related to the deployment, go to **Resource management > Deployments**, and then click the name of the deployment.

Ensure compliance of managed resources

Compliance is the process of auditing a resource against a benchmark or a policy. DCA supports the following types of compliance:

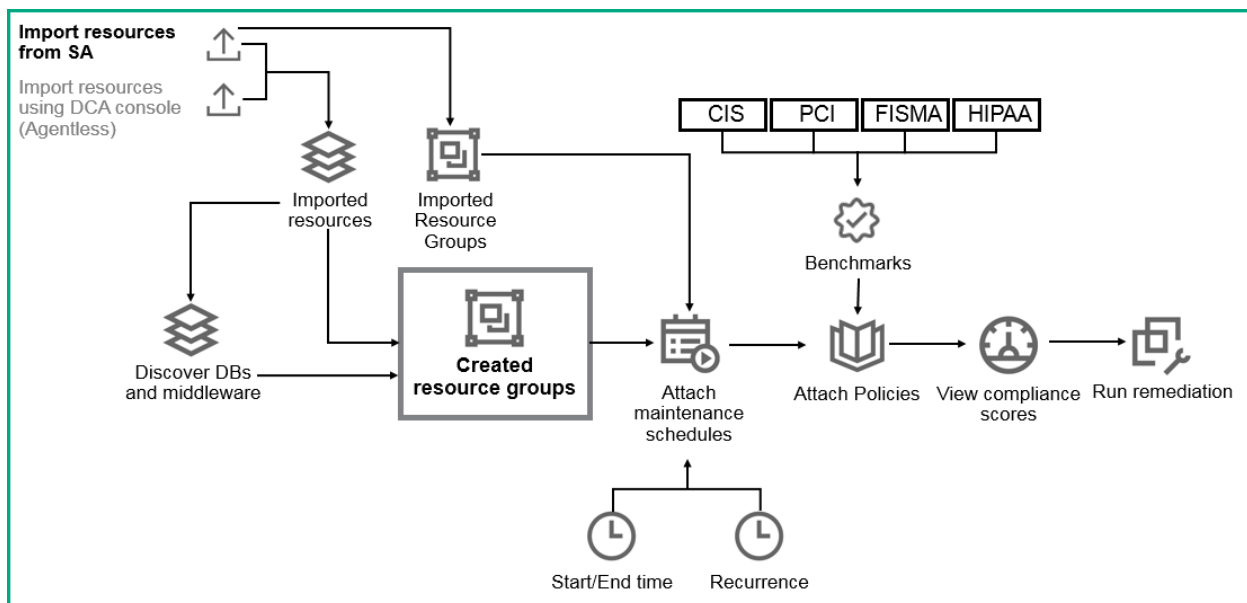
- **Policy-based compliance:** The policy-based compliance achieves continuous monitoring and remediation of compliance status through policy subscription on the resource groups. The

policy subscription attaches a policy to a given resource group. The policy-based compliance ensures ongoing compliance and remediation of managed resources (OS and application resources).

DCA provides the following out-of-the box policies:

- Center for Internet Security (CIS)
- Payment Card Industry Data Security Standard (PCI DSS)
- Federal Information Security Management Act (FISMA) Special Publication 800-53 (SP800-53)
- Health Insurance Portability and Accountability Act of 1996 (HIPAA)

- **Ad hoc compliance:** The ad hoc compliance assesses the compliance status of a given resource against a benchmark or a policy at any given time.



As a DCA user, you can run compliance scan and remediate non-compliant resources. The following sections detail the process of compliance scan and remediation job:

Policy-based compliance scan

The policy-based compliance scan is performed on a resource group. The following steps are involved:

1. [Import resources](#)
2. [Create resource groups](#)
3. [Set a maintenance schedule to the resource group](#)
4. [Attach policy to resource group](#)
5. [View compliance score](#)
6. [Remediate a non-compliant resource](#)

Ad hoc compliance scan

The ad hoc compliance is performed on a resource. The following steps are involved:

1. [Import resources](#)
2. [Run compliance scan](#)
3. [View compliance score](#)
4. [Remediate a non-compliant resource](#)

Sample use cases

This section provides sample use case procedures for the following:

- [Perform policy-based compliance scan](#)
- [Perform ad hoc compliance scan](#)

Perform policy-based compliance scan

Following is a sample procedure to perform policy-based compliance scan to ensure CIS compliance on all the resources imported from SA:

1. Import resources from SA. Create resource groups.
2. When you imported resources from SA into DCA, existing resource groups from SA will also be imported. In the DCA console, go to **Resource Management > Resource Groups** and select the resource group with all the resources.
3. Set a maintenance schedule for the selected resource group.
Policy-based compliance scan will be performed by the system at a scheduled maintenance window.
4. Attach the policy to the selected resource group. Select the policy as CIS when attaching the policy.
5. View the compliance score to determine the variance.
The compliance score can be viewed only after the scheduled job is run at least once.
6. Based on the variance score, you can perform remediation on the non-compliant resource.

Perform ad hoc compliance scan

Following is a sample procedure to perform ad hoc compliance scan to ensure FISMA compliance on a particular OS resource:

1. Import resources from SA.
2. Select the OS resource to check for compliance.
3. Run the ad hoc compliance scan for the selected resource. Chose the Policy option and select FISMA from the available options.
Ad hoc compliance scan will be run immediately.
4. View the compliance score to determine the variance against the benchmarks.

5. Based on the variance score, you can perform ad hoc remediation on the non-compliant resource.

Perform tasks in DCA using ChatOps

As a DCA user, you can use ChatOps to perform tasks in real time using a chat room. You are enabled to chat with a bot (chat robot) in a interactive environment and use specific functions of DCA without accessing the user interface.

To perform tasks in DCA using ChatOps:

1. [Setup ChatOps](#) in the DCA environment.
2. [Run ChatOps commands](#).

Following are the ChatOps commands that are currently available:

Command	Description
get deployment	Obtains the summary of a deployment.
get resource	Lists the details of a resource
list watches	Obtains a list of all the deployments that are being watched
unwatch deployment	Unwatches a deployment
watch deployment	Watches the compliance score of a resource or the status of a deployment

Generate reports

As a DCA user, you can generate reports to view data regarding compliance and deployment.

To be able to generate reports in DCA:

1. [Install OBR version 10.10](#)
2. [Install OBR-DCA Suite Content Pack version 10.10.001](#)
3. [Install DCA](#)
4. [Configure OBR to enable DCA reporting](#)
5. [Use reports](#)

Currently, DCA supports the following reports:

Report	Description
Compliance Job History	Provides information about the details of jobs for all benchmarks, with resources and job execution time.
Policy Compliance Details	Provides information about the compliance status and details run for each policy.
Policy Compliance Summary	Provides information about the detailed compliance state of a benchmark (or multiple benchmarks) for each resource.
Resource Compliance Details	Provides information about the compliance details and scan status of a resource.
SLO Conformance Summary	Provides information about the Measurement SLO and Remediation SLO for every benchmark for a selected policy.

Overview of the ITOM Container Deployment Foundation

DCA suite must be deployed on HPE ITOM Container Deployment Foundation (ITOM CDF), which powers the deployment of the following container-based software suites and drives significantly the overall time to value of customers:

- Data Center Automation (DCA)
- IT Service Management Automation (ITSMA)
- IT Event Correlation and Management (OpsBridge)
- Helion Cloud Management (HCM)



Tip

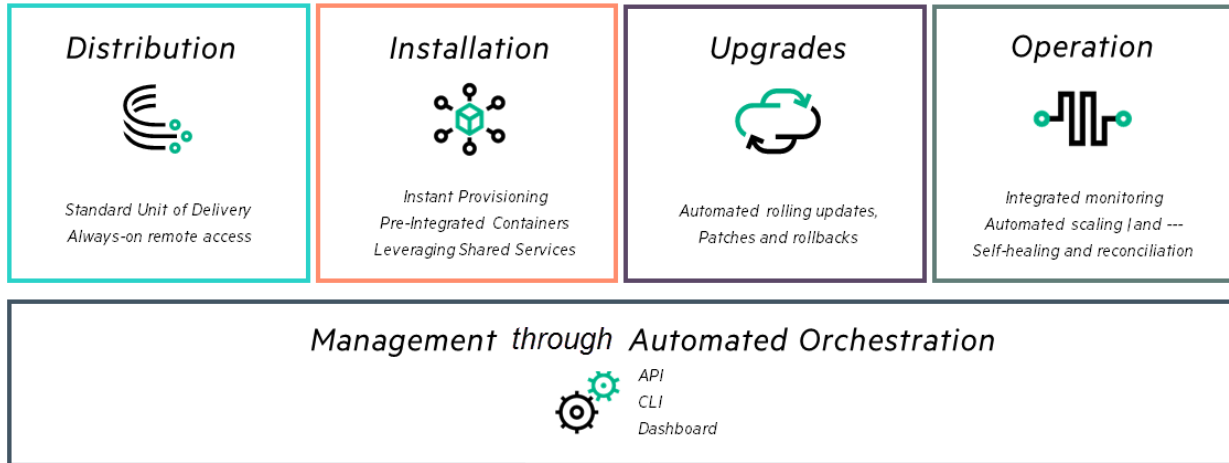
For a better understanding of the terms used in this documentation, see [Glossary](#).

New and agile software delivery platform

The installation and deployment of complex software packages in an IT environment has been traditionally very complex and expensive, involving many teams and additional resources, such as professional services. Time is also consumed for any post-deployment activities such as integration work. Upgrades have also been very slow to be adopted with very large regression cycles gated by many environments until updates finally are pushed to your production environment.

By building a new and agile software delivery platform, alongside modernized software, ITOM CDF allows customers to install pre-integrated suite capabilities. Not only Day-1 type of operations have been resolved immediately due to the nature of a container based infrastructure, the same platform allows for easier access and deployment and operation of subsequent upgrades.

The distribution unit of the software delivery is container-based, leveraging the speed and format of the new containerized environment. By bundling an orchestration layer to bootstrap and manage the lifecycle of many suite-related containers, you are able to standardize on deployment, upgrade and patching, scaling and rollbacks.



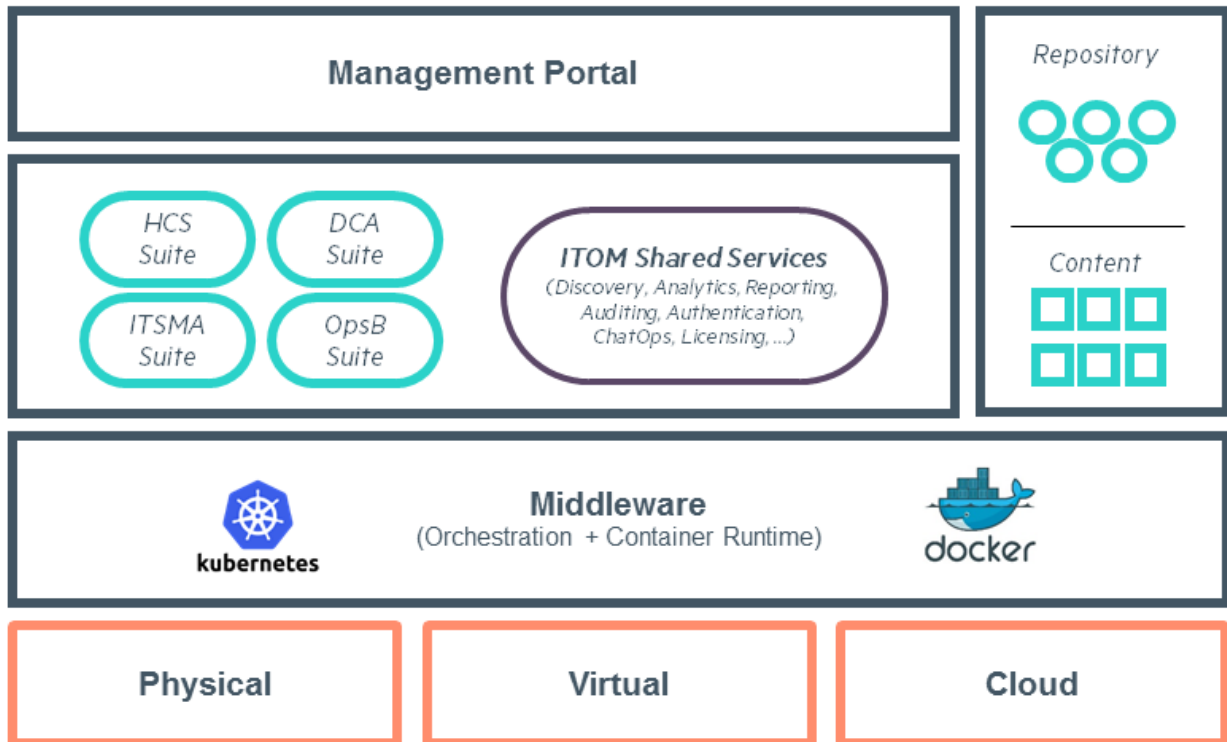
Architecture

ITOM CDF is a container deployment foundation that is built to run on many environments, offering the right level of flexibility. Deploy your IT Operations Management suite either on a bare metal or a virtual environment. ITOM CDF is using cloud native toolset, such as Docker technologies and Kubernetes from the Cloud Native Computing Foundation (CNCF) that allows the management of container-based applications at scale.

The following figure depicts the ITOM CDF architecture.

This foundation comes with additional services, including:

- A secured configuration store powered by **Vault**
- A networking layer managed by **Flannel**
- A distributed configuration database - **etcd** (runs on all of the Docker hosts) provides a unified view - file system type database to store the configuration information for the container platform.
- **A set of shared services** such as Identity Management – to plug against an existing enterprise authentication store such as LDAP or Microsoft Director, or the Licensing service to track license consumption throughout your suite
- **A Repository and Content** server. A local Container registry service handles the management of our container image delivery.
- **A Management Portal.** This is a web-based application and dashboard that you will use to install and manage suites, manage the container images required to bootstrap the suite, and manage the underlying infrastructure.



The ITOM CDF console

See [Administer the ITOM Container Deployment Foundation](#) to implement all the functions that the ITOM CDF console offers.

The primary components of the ITOM CDF console are the following tabs:

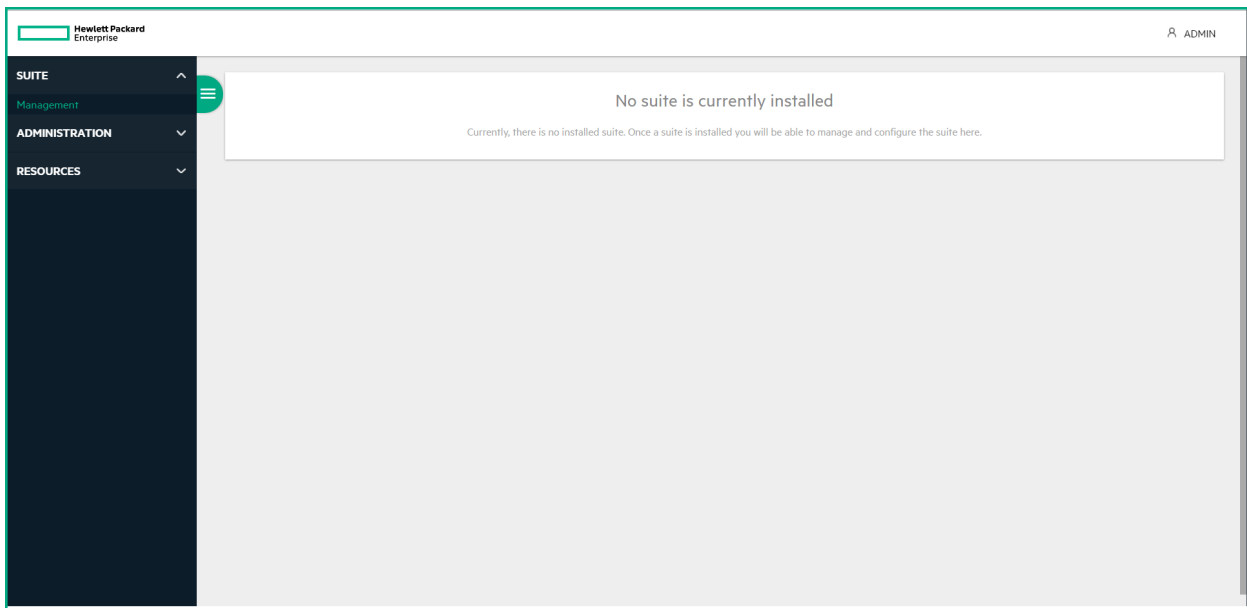
SUITE

The **SUITE** menu provides information on how to install the ITOM suites and to manage suites by exporting a deployment log or uninstalling a deployment.

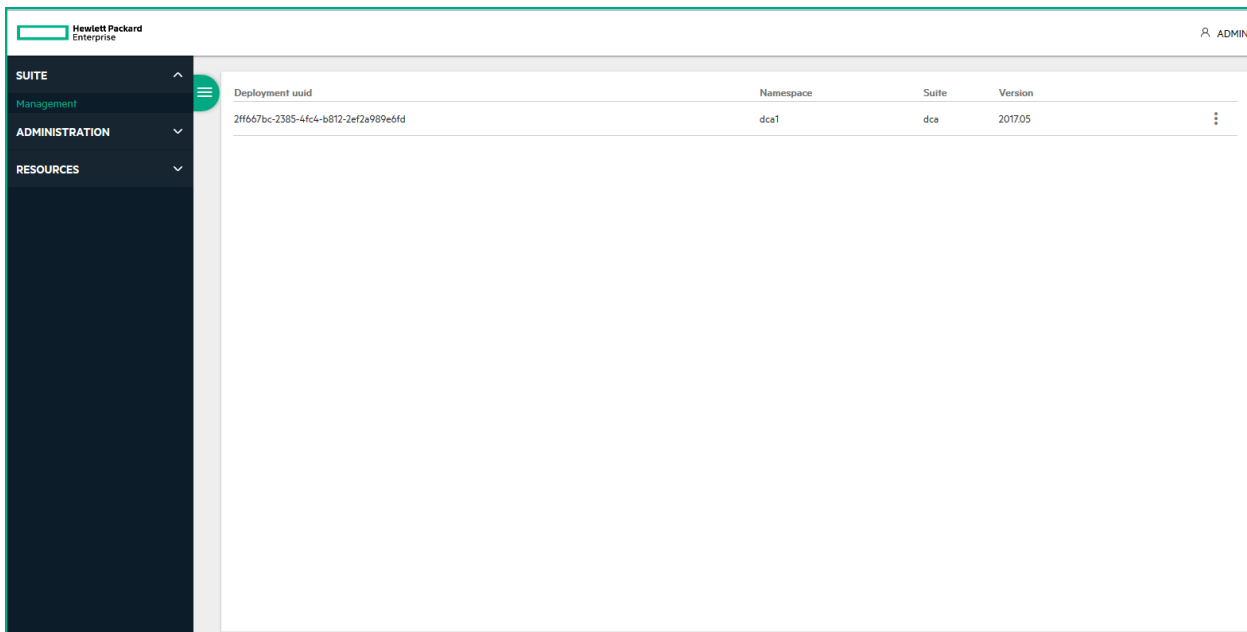
Click **SUITE** to access the following options and perform related operations.

UI element	Description
Installation	Enables you to install suites. See Install for more information.
Management	Enables you to manage the installed suites.

Before the suite is installed, click **SUITE > Management** to open the following page.



After installed suite, click **SUITE > Management**. The **Management** page opens.



Click **? Unknown Attachment** and the required option to perform the following operations on the selected suite:

UI element	Description
Export logs	Downloads the suite installation log to the local registry.
Uninstall	Uninstalls the installed suite.

Reconfigure	Reconfigures the installed suite.
--------------------	-----------------------------------

ADMINISTRATION

The **ADMINISTRATION** menu facilitates the management of infrastructure elements.

Click **ADMINISTRATION** to access the following options and perform related operations.

UI element	Operation
Admin	Provides detailed information about namespaces, nodes, and persistent volumes.
Nodes	Provides the CPU and memory usage history of the selected namespace, a list of the predefined labels, and the list of nodes of the selected namespace.
LDAP	Enables you to configure the integration of an external LDAP directory.
LWSSO	Enables you to set up a single sign-on with other products.
User Management	Enables you to create and delete users. You can also edit or view user information.
License	Enables you to manage your suite licenses.
Local Registry	Lists the images that are in the local registry.

RESOURCES

The **RESOURCES** menu enables you to deploy containerized applications to a Kubernetes cluster, troubleshoot them, and manage the cluster and its resources itself. You can use it for getting an overview of applications running on the cluster, as well as for creating or modifying individual Kubernetes resources and workloads, such as daemon sets, pet sets, replica sets, jobs, replication controllers and corresponding services, or pods.

It also provides information on the state of pods and replication controllers, and errors that might have occurred. You can inspect and manage the Kubernetes resources, as well as your deployed containerized applications. You can also change the number of replicated pods, delete pods, and deploy new applications using a deploy wizard.

Click **RESOURCES** to access the following options and perform related operations.

UI element	Description
Namespace	Provides details about the selected namespace.

Workloads	Displays information about namespaces, deployments, replica sets, replication controllers, daemon sets, jobs, pods, filtered by the selected namespace.
Service and Discovery	Displays information about services and ingresses.
Persistent Volume Chains	Displays information about the currently running persistent volumes.
Configuration	Displays information about secrets and config maps.

Related topics

[Install](#)

[Overview of DCA](#)

Glossary

This topic contains list of terms that are commonly used in DCA.

-A-

Authentication

Authentication is the process of verifying a user's identity to ascertain who the user is. The process usually requires users to provide their credentials (username and password) or a valid Auth token. This input is then used to verify if the users are really the people who they claims to be. Authentication always precedes Authorization.

Authorization

Authorization is the process of verifying the access rights of users against the secure resources of the system, and granting appropriate access based on the user roles. This process ensures that the users can access only those resources and perform only those operations for which they have the necessary permissions.

A

[Authentication](#)

[Authorization](#)

B

[Benchmark](#)

C

[ChatOps](#)

[Cloud Optimizer \(CO\)](#)

[Cluster](#)

[Config Map](#)

[Containers](#)

[Controls](#)

D

[Daemon sets](#)

-B-

Benchmark

A benchmark is a system representation of a corporate regulatory or government policy, such as PCI, SOX, or FISMA.

In DCA, each benchmark has an hierarchy of requirements, with each requirement having one or more rules. You can create your own benchmarks or use the default ones, that come out-of-the box when installing DCA.

-C-

ChatOps

ChatOps is an HPE software that allows DCA users to perform tasks in real time using a chat room.

Cloud Optimizer (CO)

Cloud Optimizer is a web-based analysis and visualization tool that analyzes the performance trends of elements in virtualized environments. It enables virtualization monitoring by providing an overview of the environment, near-real-time and historical data analysis and triaging using an interactive dashboard. It also enables monitoring for cloud and hypervisor environments. Cloud Optimizer provides essential recommendations in capacity analysis and optimization such as right-sizing, placement of VMs, forecast for resources based on their usage, and impact of adding or deleting resources in your environment.

Cluster

Kubernetes is an open-source platform, which uses clusters to run your applications. A cluster is a set of physical or virtual machines, and other infrastructure resources. A single cluster should be able to satisfy the needs of multiple users or groups of users or user communities. Each user community must be able to work in isolation from other communities and will comprise of the following:

- resources (pods, services, and replication controllers)
- policies (who can or cannot perform actions in their community)

Database
Middleware
Automation

Docker

I

Identity Management

Ingress

K

Kubernetes

L

Label

Lightweight single
sign-on (LWSSO)

M

Master node

N

Namespaces

O

Operations Bridge
Reporter

P

Persistent volume

Pet set

Pod

Policy

Provisioning

R

Replica sets

Replication
Controller

Resource types

Resources

S

Server Automation

- constraints (for example, this community is allowed this much quota)

A cluster operator may create a [Namespace](#) for each unique user community. Kubernetes coordinates a highly available cluster of computers that are connected to work as a single unit. The abstractions in Kubernetes allow you to deploy containerized applications to a cluster without associating them specifically to individual machines. To make use of this new model of deployment, applications need to be packaged in a way that decouples them from individual hosts; that is, they need to be containerized. Containerized applications are more flexible and available than in past deployment models, where applications were installed directly onto specific machines as packages deeply integrated into the host. Kubernetes automates the distribution and scheduling of application containers across a cluster in a more efficient way.

A Kubernetes cluster consists of two types of resources:

- Master: Coordinates the cluster
- Nodes: Workers that run applications

See <http://kubernetes.io/docs/admin/cluster-management/> for detailed information.

Config Map

The [ConfigMap](#) API resource holds key-value pairs of configuration data that can be consumed in pods or used to store configuration data for system components such as controllers. ConfigMap is similar to [Secrets](#), but designed to more conveniently support working with strings that do not contain sensitive information.

See <http://kubernetes.io/docs/user-guide/configmap/> for detailed information.

Containers

In containers, everything required to make a piece of software run is packaged into isolated containers. Unlike VMs, containers do not bundle a full operating system - only libraries and settings required to make the software work are needed. This makes for efficient, lightweight, self-contained systems, which guarantee that the software will always run the same, regardless of where it is deployed.

Controls

[Controls](#) define what to measure, how to evaluate compliance, and how to remediate non-compliance. A control is a reusable, shared function or test used in a benchmark to create a rule, and it is associated with benchmark

Service
Slack
T
Tag
Template
W
Worker node

requirements through rules. The same control can be used multiple times in the same benchmark. A control can be broken down into a script that is associated with a rule. When you scan a deployment, scripts are run in the background and based on the results of the scripts, scan results are determined. Controls are associated with rules and this association happens out-of-the-box in this version of DCA.

-D-

Daemon sets

A [Daemon Set](#) ensures that all (or some) nodes run a copy of a pod. Pods are added to the cluster along with nodes. As nodes are removed from the [cluster](#), those pods are garbage collected. Deleting a Daemon set will clean up the pods it created.

If you are running clustered Kubernetes and are using static pods to run a pod on every node, you should use a Daemon set. Static pods are managed directly by the kubelet daemon on a specific node, without the API server observing it. Kubelet automatically creates mirror pods on the Kubernetes API server for each static pod, so that the pods are visible. Without an associated replication controller, the kubelet daemon watches and restarts static pods that crash. However, there is no health check performed. Static pods are always bound to one kubelet daemon and always run on the same node with it.

See <http://kubernetes.io/docs/admin/daemons/> for detailed information.

Database Middleware Automation

HPE Database and Middleware Automation (HPE DMA) software automates tasks like provisioning and configuring, compliance, patching, and release management for databases and application servers.

Docker

Docker is an open-source platform that provides a way to run any application securely in an isolated container.

-I-

Identity Management

A service that provides authentication and single sign-on for all components in DCA.

Ingress

An [Ingress](#) is a collection of rules that allow inbound connections to reach the cluster services. It can be configured to provide service URLs, load balance traffic, terminate SSL, and offer name-based virtual hosting. Users can request Ingress by POSTing the Ingress resource to the API server. An Ingress controller is responsible for fulfilling the Ingress, usually with a loadbalancer, though it may also configure your edge router or additional frontends to help handle the traffic in a HA environment.

See <http://kubernetes.io/docs/user-guide/ingress/> for detailed information.

-K-

Kubernetes

Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts. It groups containers that make up an application into logical units for easy management and discovery.

See <http://kubernetes.io/docs/> for detailed information.

-L-

Label

A label is a key/value pair that is attached to objects, such as pods, to convey a user-defined identifying attribute. Labels can be used to organize and to select subsets of resources. Labels can be attached to objects at the time of creation, and subsequently added and modified at any time. Each object can have a set of key/value labels defined. Each key must be unique to a given object.

For example, you can use the labels when you install a suite so that you can categorize nodes to serve certain functions. This enables you to install a suite only on certain nodes and facilitates the partitioning of your cluster.

See <http://kubernetes.io/docs/user-guide/labels/> for detailed information.

Lightweight single sign-on (LWSSO)

A method of access control that enables a user to log on once and gain access to the resources of multiple software systems without being prompted to log on again.

LWSSO is an HPE solution that enables single sign-on using one authentication across various HPE applications.

LWSSO shares a cookie between HPE products that are accessed from a web browser. As a result, a user has to log on once and can gain access to the resources of multiple HPE software systems without being prompted to log on again.

-M-

Master node

The controlling services in a Kubernetes cluster are called the master components. These operate as the main management contact points for administrators, and also provide many cluster-wide systems for the [worker nodes](#). These services can be installed on a single machine, or distributed across multiple machines.

-N-

Namespaces

A [namespace](#) is a mechanism to partition resources created by users into a logically named group. A namespace is similar to a prefix to the name of a resource. Namespaces help different projects, teams, or customers to share a cluster, by preventing name collisions between unrelated teams.

There are two default namespaces:

- **default:** The default namespace for objects with no other namespace
- **kube-system:** The namespace for objects created by the Kubernetes system

See <http://kubernetes.io/docs/admin/namespaces/> for detailed information.

-O-

Operations Bridge Reporter

Operations Bridge Reporter (OBR) is a solution based on Big Data technology HPE Vertica, and has been built to specifically address the challenges of reporting in dynamic IT environments. In addition to consolidating performance data and metrics from multiple domain-focused collectors, it also collects and collates specific information on the relationships between the IT elements and the business services. OBR provides sophisticated data collection and aggregation coupled with industry-leading report definition and generation capabilities.

-P-

Persistent volume

A PersistentVolume (PV) is a piece of networked storage in the cluster that has been provisioned by an administrator. PVs are volume plugins that have a lifecycle independent of any individual pod that uses the PV. This API object captures the details of the implementation of the storage of NFS, iSCSI, or a cloud-provider-specific storage system.

A **Persistent Volume Claim** (PVC) is a request for storage by a user. Just as pods consume node resources, PVCs consume PV resources. Claims can request specific size and access modes; that is, they can be mounted, read or written once or read-only many times.

See <http://kubernetes.io/docs/user-guide/persistent-volumes/> for detailed information.

Pet set

A **pet set** is a controller that provides a unique identity to its pods. It provides a guarantee about the order of deployment and scaling.

Pod

A **pod** is a co-located group of containers and volumes. Pods serve as units of scheduling, deployment, and horizontal scaling/replication. They are scheduled onto the same host. Pods share fate and some resources, such as storage volumes and IP addresses.

See <http://kubernetes.io/docs/user-guide/pods/> for detailed information.

Policy

A [policy](#) contains software resources such as packages, patches, RPM packages, scripts, application configurations, and server objects that need to be installed on managed servers.

Provisioning

Provisioning is the process of deploying, configuring, and managing the resources within your data center. Provisioning begins with a user-initiated action of selecting a template for resource deployment. DCA then runs a pre-check to ensure that the selected resource is not provisioned. A successful pre-check is followed by the installation and configuration of the software (OS, DB, middleware, and applications) based on the deployment parameters. As the final step in provisioning, DCA runs post-checks on individual resources to verify the successful installation, configuration, and compliance states of the resources.

-R-

Replica sets

A [replica set](#) ensures that a specified number of pod “replicas” are running at any given time. However, a deployment is a higher-level concept that manages replica sets and provides declarative updates to pods. Therefore, we recommend using deployments instead of directly using replica sets, unless you require custom update orchestration or do not require updates at all.

See <http://kubernetes.io/docs/user-guide/replicasets/> for detailed information.

Replication Controller

A [replication controller](#) ensures that a specified number of pod replicas are running at any one time. It allows both for easy scaling of replicated systems and handles re-creation of a pod when the machine it is being rebooted or has failed.

See <http://kubernetes.io/docs/user-guide/replication-controller/> for detailed information.

Resource types

A resource type is a pre-state to deploying a resource. It is an entity that will be deployed on the chosen target. A resource type is identified by a unique name and each resource type can have many attributes.

See [Resource types](#) for detailed information.

Resources

A resource is a compute, storage, or a network component that falls under the management of the DCA Suite, and can be hardware or software.

See [Resources](#) for detailed information.

-S-

Secret

A secret stores sensitive data, such as authentication tokens, which can be made available to containers upon request.

See <http://kubernetes.io/docs/user-guide/secrets/> for detailed information.

Server Automation

[Server Automation \(SA\)](#) automates tasks such as provisioning, patching and compliance across physical and virtual server environments. For example, by using SA Provisioning, you can set standards for different types of servers and automatically provision the servers, saving time and ensuring that operating system builds are consistent. You can establish patch policies to install and maintain patches for supported operating systems running on managed servers in your IT environment.

By using compliance, you have visibility across your managed servers to see which servers are out of compliance. You can then remediate non-compliant servers to bring them back into compliance, based on the policies you created.

Service

A [service](#) defines a set of pods and a means to access them, such as single stable IP address and corresponding DNS name (for example, a web service or API server) that directs and loads-balances traffic to the set of pods that it covers.

See <http://kubernetes.io/docs/user-guide/services/> for detailed information.

Slack

Slack acts as a platform for ChatOps. It prioritizes API integration and offers a conversational interface for implementing ChatOps in DCA.

-T-

Tag

A [tag](#) classifies and categorize entities (resource types, templates, and deployments) within your infrastructure. You can [associate](#) a tag with any number of entities. This helps you to quickly [search for](#) or [sort](#) the entities by the associated tags.

Template

A [template](#) is a prototype for a deployment that defines infrastructure resources, attributes, parameters, and policies.

-W-

Worker node

Worker nodes run the applications. A node may be a VM or physical machine, depending on the [cluster](#). Each node has the services necessary to run [pods](#) and is managed by the [master components](#). The services on a worker node include Docker, kubelet, and kube-proxy.

See <http://kubernetes.io/docs/admin/node/> for detailed information.

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.