



Hewlett Packard
Enterprise

Codar/Cloud Service Automation

Software version: 1.81/4.81

For Microsoft Windows® and Linux operating systems

Docker Cloud Integration

Document release date: May 2017

Software release date: May 2017

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.

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

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

UNIX® is a registered trademark of The Open Group.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission.

Documentation updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to the following URL and sign-in or register: <https://softwaresupport.hpe.com>.

Select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, whitepapers, and other information sources.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your Hewlett Packard Enterprise sales representative for details.

Support

Visit the Hewlett Packard Enterprise Software Support Online web site at <https://softwaresupport.hpe.com>.

Contents

Legal notices	2
Introduction	4
Configuration Requirements	4
Supported versions	4
Configure Docker Cloud Resource Provider	4
Pre-requisites	8
Embrace Docker Image as Component	10
Component Properties	11
Application Blueprints	13
Send documentation feedback	15

Introduction

The containerization technology is rapidly emerging and inspiring many software companies to adopt and use it in their DevOps tool chain to get maximum resource utilization and reduce time-to-market. Docker containerization has become habitual than a buzzword. Docker Cloud offers container datacenter for the application and operations team to install or deploy software; and application with pre-baked dependencies inside a container ensures the application deployment certainty. This whitepaper describes how HPE Cloud Service Automation (CSA) and HPE Codar are integrated with Docker Cloud and the benefits the end-user will get by using this integration. HPE CSA is primarily used by the IT Ops or Central IT for providing IaaS & PaaS for their LOB's or application team. HPE Codar offers release pipeline automation along with deployment automation intelligence. Both the products can utilize Docker Cloud platform to deploy containers which have software or application in them.

Configuration Requirements

The following configurations must be completed, tested and should be operational before you proceed with the integration:

- Configure HPE Codar.
- Configure Docker Cloud
- Configure HPE Operations Orchestration.

Supported versions

The following table shows the major components required to use this implementation.

Component	Supported version	Recommended version
HPE Codar	1.81	1.81
HPE Operations Orchestration	10.22 and later	10.60 with the following HPE Operations Orchestration: <ul style="list-style-type: none">• oo10-base-cp-1.8.0.jar• oo10-cloud-cp-1.8.2.jar• oo10-hpe-solutions-cp-1.8.2.jar• oo10-sa-cp-1.2.2.jar• oo10-sm-cp-1.0.3.jar• oo10-virtualization-cp-1.8.0.jar
Docker Cloud	1.2	1.2

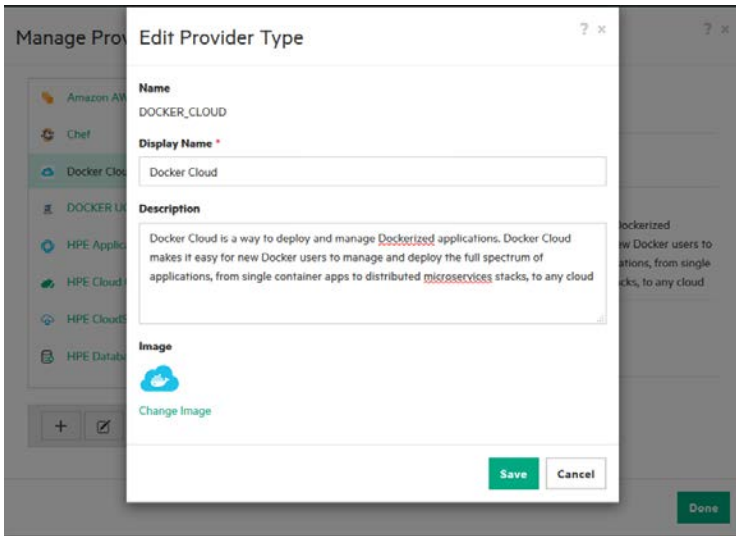
Configure Docker Cloud Resource Provider

Docker Cloud resource provider can be used to integrate with Docker Cloud 1.2 to import image as component and spawn containers across multiple nodes configured as part of Docker Cloud. Docker Cloud endpoints with credentials and can be configured within HPE CSA and HPE Codar through OOTB resource provider.

Creating Docker Cloud Provider Type

1. Login to HPE Codar/ CSA.
2. Click Provider tile.
3. Click Manage Provider Types and Click "+" icon to add new provider.
4. In Create Provider Type window enter the Display Name and description (optional).
5. Click Create to complete.

Note: Please make sure that the Display Name is DOCKER CLOUD, so that there are no CODAR/CSA update issues.




Edit Resource Provider ? x

Provider Type
Docker Cloud

Display Name *
Docker Cloud Data Center

Description

Image

[Change Image](#)

Access Point Configuration

Service Access Point *
https://cloud.docker.com ?

User ID *
docker_user

Password *
●●●●●●●●

Confirm Password *
●●●●●●●●



Enabled ?

Save **Cancel**

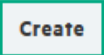
Instructions:
Provide the Docker Cloud endpoint and credentials to connect to the Docker Cloud server.

Service Access Point: Docker Cloud HTTPS endpoint.
User ID: Docker ID
Password: Docker Cloud API Key

Fig: 1 → Docker Cloud Resource Provider

 **Docker Cloud Data Center** 

Overview **Properties** **Environments** **Resource Offerings** **Resource Pools** **Components**

Properties 





Proxy Host web-proxy.com Hostname of the Proxy Server	String 
Proxy Password Not Set Password of the Proxy Server	String 
Proxy Port 8080 Port of the Proxy Server	String 
Proxy Username Not Set Username of the Proxy Server	String 

Fig: 2 → Additional properties to configure “Docker Cloud” endpoints and credentials.

Pre-requisites

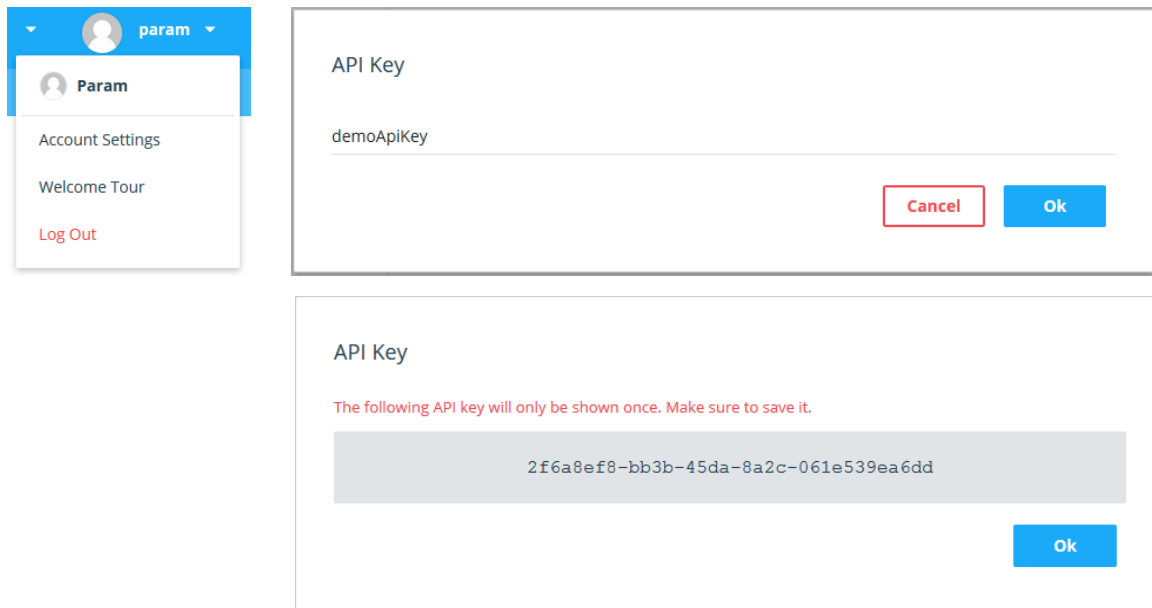
Docker Cloud API Key

Docker Cloud currently offers an **HTTP REST API** and a **Websocket Stream API** which are used by both the [Web UI](#) and the [CLI](#).

To make requests to the Docker Cloud API, you should first obtain an API Key for your account. For this, log into Docker Cloud, click on the menu on the upper right corner of the screen, select **Account Settings** and then select **API keys**.

Click **Add API key** and give the label for the API Key and click Ok. Now, you'll see the API Key generated.

Note: API key will only be shown once. Make sure to save it properly.



Now, you'll see newly added API key with the tag name **demoApiKey** in that section

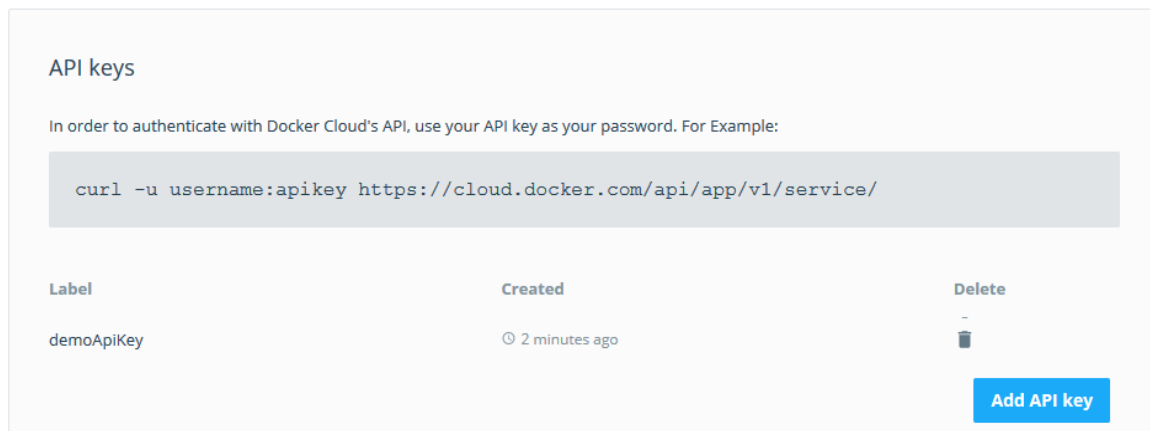


Fig: 3 → Add API Key for the user in Docker Cloud

Docker Cloud Nodes / Nodes Cluster

Before starting deployments of service, we need to have one or more infrastructure node set up under your docker cloud setup. To add, just go to Node cluster under **Infrastructure** and you can either create a node cluster or import existing one on any of the multiple resource provider options available.

Node Clusters / Wizard

NAME	node1
DEPLOY LABELS	Select...
PROVIDER	Amazon Web Services
REGION	Amazon Web Services
VPC	Digital Ocean
SUBNET	Microsoft Azure
SECURITY GROUP	SoftLayer
TYPE/SIZE	Packet
	x default
	t2.nano [1 CPUs, 512 MB RAM]

Node disk size

60 GB

Number of nodes

2 nodes

Launch node cluster

Fig: 4 → Adding Node Cluster in Docker Cloud

Embrace Docker Image as Component

HPE CSA and HPE Codar support topology-based designs. The components are the building blocks for the service blueprints and can be imported from various resources like HPE Operations Orchestration, HPE Server Automation, Chef, Puppet, etc.

In the same fashion, we can either import the component using HPE Operations Orchestration or create the component and then later import the specific operations.

The image names can be mentioned from Docker Cloud's Repositories or Docker Hub.

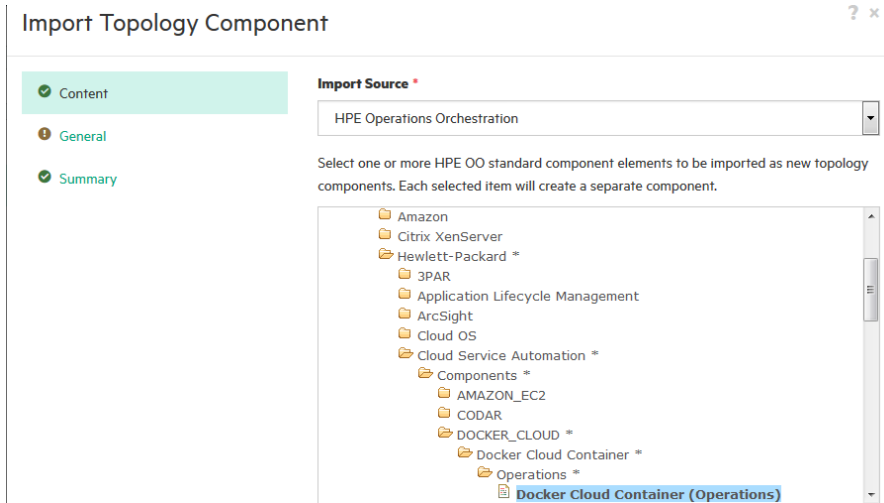


Fig: 4 → Importing Docker Cloud components

Imported Components

The imported component needs to be configured for the relationship, if required, and for setting the capabilities and characteristics. The default operations available are Deploy and Undeploy. Other custom operations can be imported from the content.

The custom operation includes the following:

1. Start Service
2. Get Service
3. Stop Service
4. List Service

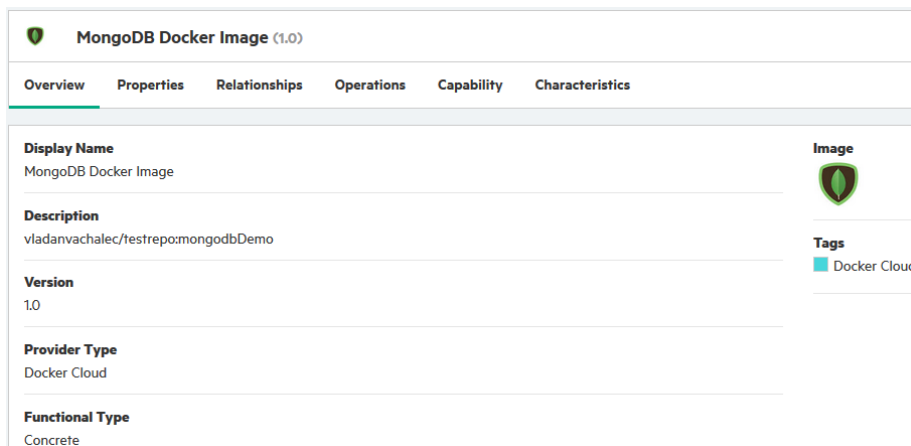


Fig: 5 → Sample Docker Cloud component

Component Properties

The component properties include input, output, and provider properties. The input properties are given below.

Refer the information below to pass inputs to create containers in a non-controlled environment. If it is made visible to consumer, the end-users can pass the docker run options on their own and pass any inputs based on their requirement. They can be hardcoded too and these arguments need not be made visible.

Property	Description
containerSize	(Optional argument) The number of containers to run for this service initially (default: 1)
portsTobeExposed	(Optional argument) An array of objects with port information to be published in the containers for this service, which will be added to the image port information, i.e. {"protocol": "tcp", "inner_port": 80, "outer_port": 80}
serviceName	(Optional argument) A human-readable name for the service, i.e. my-hello-world-app (default: image without namespace)
imageName	(Required argument) The image used to deploy this service in docker format, i.e. tutum/hello-world

You can mask component properties like 'containerSize' from end-user by unchecking the options like 'designer visible and consumer visible'.

Note: Image name should not be modified once set, and it is strongly recommended to create different component for different images.

All the other property values are automatically fetched from the provider property.

MongoDB Docker Image (1.0)					
Overview	Properties	Relationships	Operations	Capability	Characteristics
Properties					
containerSize	dockerProxyHost	imageName			
Not Set	Not Set	Not Set			
dockerApiKey	dockerProxyPassword	nodePubliIdp			
[*****]	Not Set	Not Set			
dockerCloudUrl	dockerProxyPort	portsTobeExposed			
Not Set	Not Set	Not Set			
dockerId	dockerProxyUsername	serviceId			
Not Set	Not Set	Not Set			
dockerProtocol	exposedPorts	serviceName			
Not Set	Not Set	Not Set			

Fig: 6 → Default Component Properties

MongoDB Docker Image (1.0)

Overview Properties Relationships **Operations** Capability Characteristics

Operations

Deploy Service on Docker Cloud
 containerSize, dockerApiKey, dockerCloudUrl, dockerId, dockerProtocol, dockerProxyHost, dockerProxyPassword, dockerProxyPort, dockerProxyUsername, imageName, portsTobeExposed, serviceName

Undeploy Service on Docker Cloud
 dockerApiKey, dockerCloudUrl, dockerId, dockerPassword, dockerProxyHost, dockerProxyPassword, dockerProxyPort, dockerProxyUsername, serviceld

Fig: 7 → Default operations as part of Docker Cloud component

The Deploy operation will have some properties exposed as output properties which can be used further in different components. There is no need to change these property mappings.

The properties include the following:

Output Parameter Mappings

<input type="radio"/>	exposedPorts	exposedPorts Component Property
<input type="radio"/>	nodePublidlp	nodePublidlp Component Property
<input type="radio"/>	serviceld	serviceld Component Property

Fig: 8 → Deploy flow output properties

Application Blueprints

You can create application blueprints by adding the imported components. A sample application blueprint is given below with a component which represents a 'tomcat:8.0' Docker image.

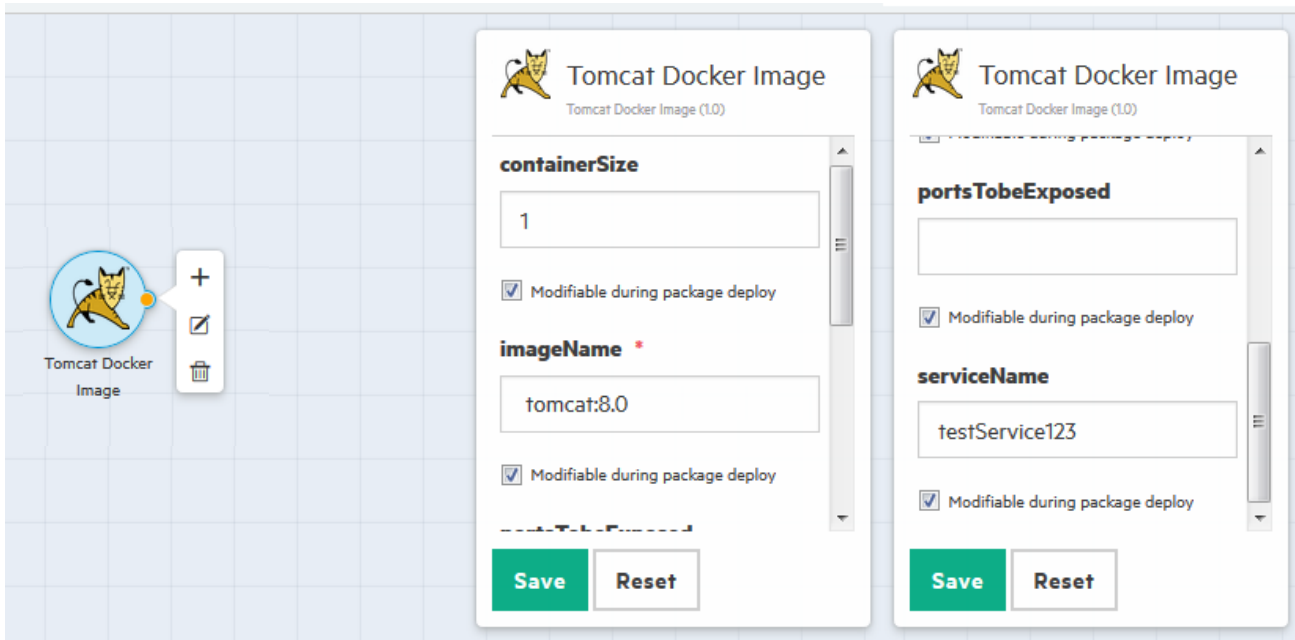


Fig: 9 → Sample Docker image component which represents tomcat application server

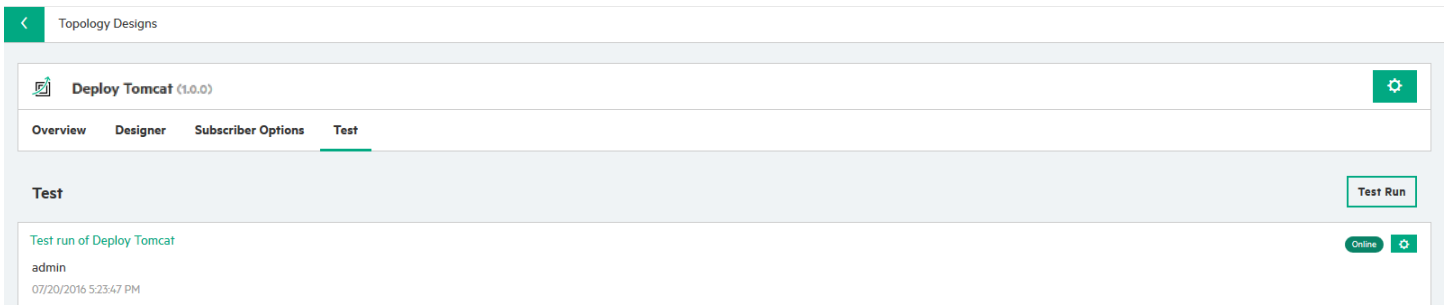


Fig 10: Test Run Result

Test run of Deploy Tomcat (Deployment Status) ⚙️

🟢 Online

Overview Events **Topology** Providers

Topology Search ☰ 👤


🔍

🔍


✖

⋮

🔍



Tomcat Docker Image

 Tomcat Docker Image

[← Properties](#)

containerSize	1	STRING
exposedPorts	8444	STRING
imageName	tomcat:8.0	STRING
nodePubIdIp	54.173.164.178	STRING
portsToBeExposed		STRING
serviceId	f9ad870a-ba38-43f3-9a78-e425da261520	STRING

Fig 11 → Topology view with output properties like docker service ID, published ports, and IP address it is hosted on.

Send documentation feedback

If you have comments about this document, you can send them to clouddocs@hpe.com.