

Cisco CloudCenter Suite 5.0 with ACI 4.0 v1



Last Updated: 10-April-2019

About This Demonstration

This guide for the preconfigured demonstration includes:

About This Demonstration

Requirements

About This Solution

Topology

Get Started

Scenario 1. User Role - Application Deployment

Scenario 2. Architect Role - Editing Application Profiles

Scenario 3. Admin Role - Create Constructs Within CloudCenter

Scenario 4. CI/CD Solution using GitLab, Jenkins, Artifactory and CloudCenter

Limitations

- Certain features of Cisco APIC 4.0 are outside the scope of this demonstration, because the demonstration uses a simulated fabric rather than a physical fabric:
 - All configuration will be lost after a reboot of the APIC simulator
 - No traffic will pass between devices connected to the simulated fabric for this reason we have 2 network interfaces on the CloudCenter deployed application.
 - Screen refresh may take slightly longer than expected
- Explanation of the features of Jenkins, Artifactory, and Git Bash are outside the scope of this demonstration.

Requirements

The table below outlines the requirements for this preconfigured demonstration.

Required	Optional
Laptop	Cisco AnyConnect®

About This Solution

CloudCenter provides customers with a single, intuitive platform that helps them manage the entire application lifecycle across simple or complex hybrid IT environments. The CloudCenter platform provides a compelling solution for modern IT organizations whether they are moving their first applications to the cloud, implementing self-service IT, or wanting to gain visibility and control across a vast portfolio of clouds, applications and users.

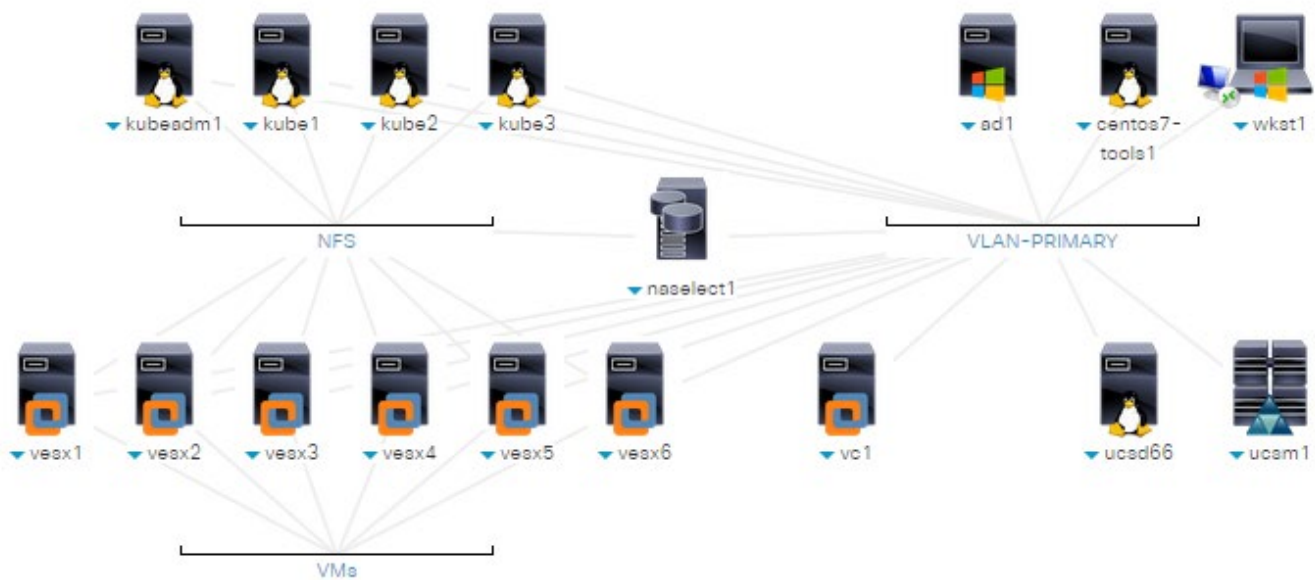
The solution provides customers with several key benefits:

- Global profile deployment: CloudCenters solution allows customers to create a single, cloud agnostic application profile once and then deploy it across any data center, public or private cloud.
- Consistent policy application: CloudCenter automatically applies a customers access control and security policies to an application, and then ensures that those policies move with the application.
- Application optimization across hybrid cloud environments: CloudCenter will measure both price and performance of applications on any cloud environment, helping users to make informed decisions about the best place for their application on any data center or cloud.
- One-click management: CloudCenter provides a single management interface to give customers complete visibility and control across applications, cloud environments and users.

Topology

This content includes preconfigured users and components to illustrate the scripted scenarios and features of the solution. Most components are fully configurable with predefined administrative user accounts. You can see the IP address and user account credentials to use to access a component by clicking the component icon in the Topology menu of your active session and in the scenario steps that require their use.

Figure 1. dCloud Topology



Get Started

BEFORE PRESENTING

Cisco dCloud strongly recommends that you perform the tasks in this document with an active session before presenting in front of a live audience. This will allow you to become familiar with the structure of the document and content.

It may be necessary to schedule a new session after following this guide in order to reset the environment to its original configuration.

PREPARATION IS KEY TO A SUCCESSFUL PRESENTATION.

Follow the steps to schedule a session of the content and configure your presentation environment.

1. Initiate your dCloud session. [\[Show Me How\]](#)

NOTE: It may take up to 10 minutes for your session to become active.

2. For best performance, connect to the workstation with **Cisco AnyConnect VPN** [\[Show Me How\]](#) and the **local RDP client** on your laptop [\[Show Me How\]](#)
 - **Workstation 1:** 198.18.133.36, **Username:** DCLOUD\demouser, **Password:** C1sco12345

NOTE: You can also connect to the workstation using the Cisco dCloud Remote Desktop client [\[Show Me How\]](#). The dCloud Remote Desktop client works best for accessing an active session with minimal interaction.








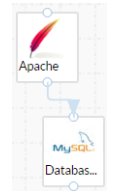
Scenario 1. User Role - Application Deployment


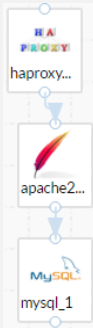



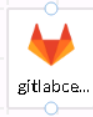

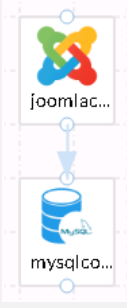




The purpose of this scenario is to deploy an application that can be monitored and managed by Cisco CloudCenter.







NOTE: If demonstrating to a customer, it is recommended that a deployment of an application be completed before demonstrating to the customer. This will decrease waiting time.

Any application in the Application Profiles list can be used for this scenario, depending on customer interest. The table below shows the applications that exist by default in the Application Profiles list.

Pre-installed Application Profiles

APPLICATION NAME	TOPOLOGY	NUMBER/TYPE OF VMs	APPROXIMATE DEPLOYMENT TIME IN MINUTES	NOTES
Applications (with ACI)				
 Apache Mesos		1 – CentOS_6 Load Balancer VM 3 – CentOS_6 Master Nodes 3 – CentOS_6 Slave Nodes	10 minutes	Complex application, longer deployment time, confirmation page with application connectivity
 Magento		1 – Nginx Load Balancer VM 1 – Apache2 Web VM 1 – MySQL Database VM	8 minutes	
 MySQL		1 – CentOS_6 MySQL Node	4 minutes	
 OpenCart		1 – Apache Web VM 1 – MySQL Database VM	7 minutes	

APPLICATION NAME	TOPOLOGY	NUMBER/TYPE OF VMs	APPROXIMATE DEPLOYMENT TIME IN MINUTES	NOTES
 WordPress		1 – HAproxy Load Balancing VM 1 – Apache2 Web VM 1 – MySQL Database VM	8 minutes	Complex application, medium deployment time
Applications (Containerized)				
 Artifactory OSS		1 – Artifactory OSS container	2 minutes	
 GitLab CE		1 – GitLab CE container	2 minutes	
 Joomla		1 – Joomla container 1 – MySQL container	3 minutes	
 Odoo		1 – Odoo container 1 – PostgreSQL container	3 minutes	
 OpenCart		1 – OpenCart container 1 – MySQL container	3 minutes	

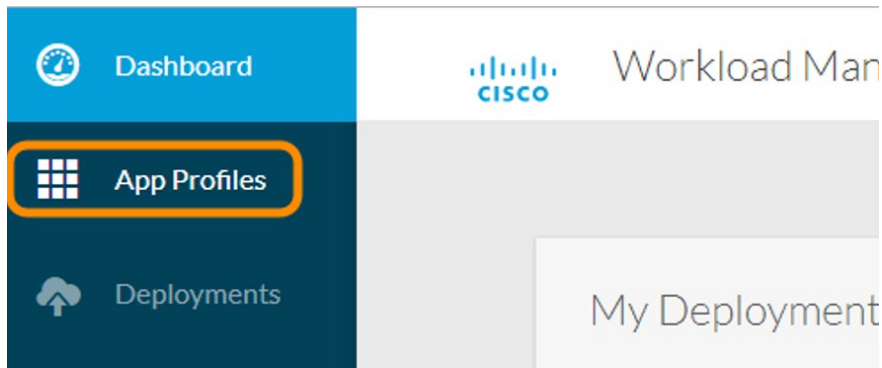
APPLICATION NAME	TOPOLOGY	NUMBER/TYPE OF VMs	APPROXIMATE DEPLOYMENT TIME IN MINUTES	NOTES
 WordPress		1 – WordPress container 1 – MySQL container	3 minutes	
SINGLE VM WITH OS INSTALLED (visible by Admin only)				
 CentOS 6		1 – CentOS 6 VM	3 minutes	
 CentOS 7		1 – CentOS 7 VM	3 minutes	

Steps

Deploy a VM-Based Application in Production (with ACI)

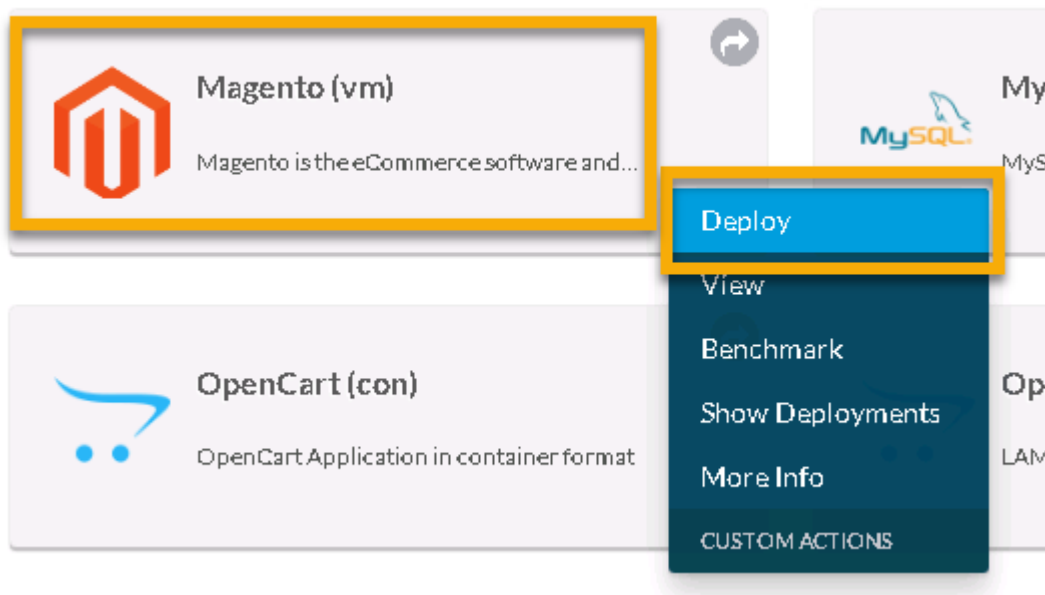
- Open a Chrome browser and perform the following actions:
 - Open a new browser tab and click the CloudCenter Suite bookmark. Log in using the credentials **user@dcloud.cisco.com / C1sco12345**, tenant id: **demo**.
 - Open a second browser tab and click the **vCenter (HTML)** bookmark. Select the **Use Windows session authentication** and click on **Login** to login as demouser.
 - Open a third browser tab and click the **Cisco APIC** bookmark. Log in using the credentials **admin / C1sco12345**.
- Switch to the **CloudCenter** browser tab.
- If not at the Workload Manager screen, click on the **Dashboard icon**.

- From the side menu, click **App Profiles** to display the available Application Profiles.

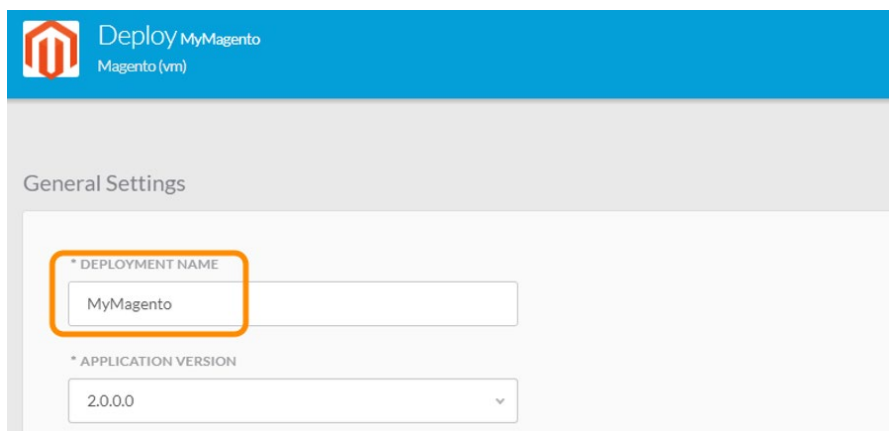


NOTE: Cisco CloudCenter provides role-based and object-based access control, so that when a user logs in, the content provided to them is based on their profile.

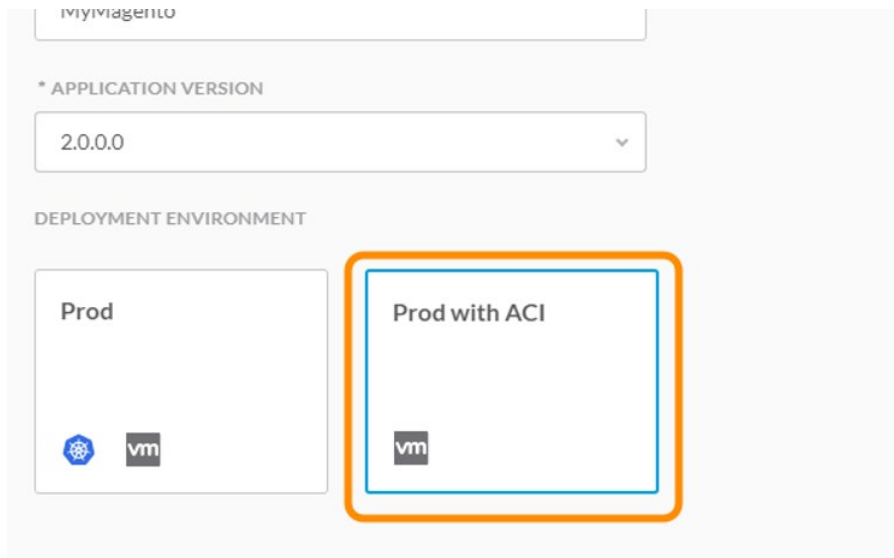
- Mouse over the **Magento (vm)** Application Profile and click **Deploy** to deploy a Magento application.



- Enter **MyMagento** at the Deployment Name.



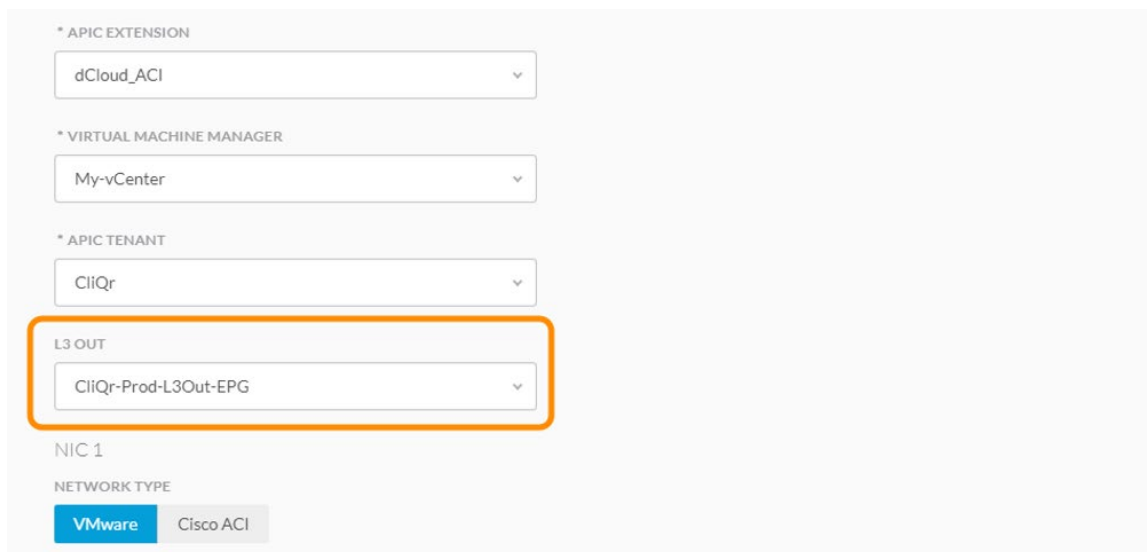
7. Choose **Prod with ACI** at the Deployment Environment.



The screenshot shows a form with the following elements:

- A dropdown menu at the top with the value "myvmagent10".
- A section titled "* APPLICATION VERSION" with a dropdown menu showing "2.0.0.0".
- A section titled "DEPLOYMENT ENVIRONMENT" containing two cards:
 - A card labeled "Prod" with icons for a gear and "vm".
 - A card labeled "Prod with ACI" with a "vm" icon. This card is highlighted with an orange border.

8. Leave the other fields to their default values and click **Next** at the bottom of the form.
9. At the next form, explore the **default selected options** (Cloud, Tiers, Instance Types, Network settings).
10. At the nginx_2 tier, scroll to the network section and at the **L3 OUT** field, choose **CliQr-Prod-L3Out-EPG**.

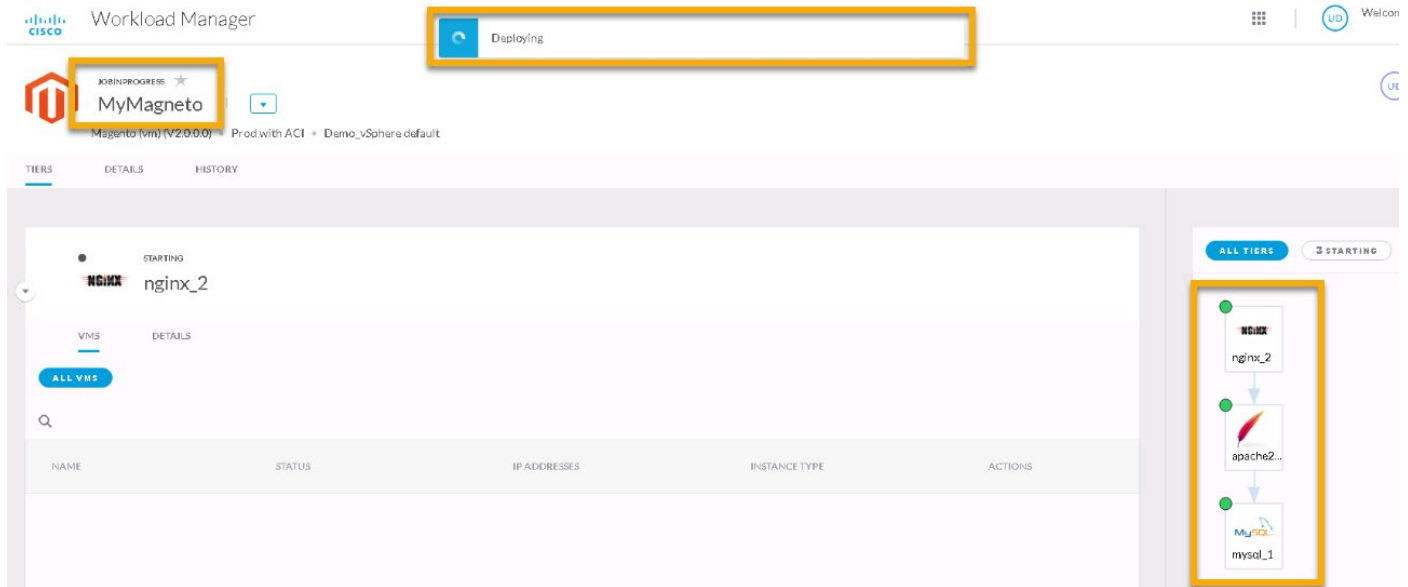


The screenshot shows a form with the following elements:

- A section titled "* APIC EXTENSION" with a dropdown menu showing "dCloud_ACI".
- A section titled "* VIRTUAL MACHINE MANAGER" with a dropdown menu showing "My-vCenter".
- A section titled "* APIC TENANT" with a dropdown menu showing "CliQr".
- A section titled "L3 OUT" with a dropdown menu showing "CliQr-Prod-L3Out-EPG". This section is highlighted with an orange border.
- A section titled "NIC 1" with a "NETWORK TYPE" section containing two buttons: "VMware" (highlighted in blue) and "Cisco ACI".

11. At the bottom of the form, click **Deploy**.

- The screen should change to the deployment form, where each tier will be deployed in sequence. Once the application is fully deployed, an **Access Application Button** will appear at the top of the form.



- Wait for the application to fully deploy and proceed with the verification steps below. To verify the application using the Access Application button, skip to Step 3 in the Verify Application Deployment section

NOTE: If you pre-configured an application prior to the demonstration, you can proceed immediately with the following steps. If not, it may take 5–10 minutes for the application to fully deploy and show the **Access Application** button.

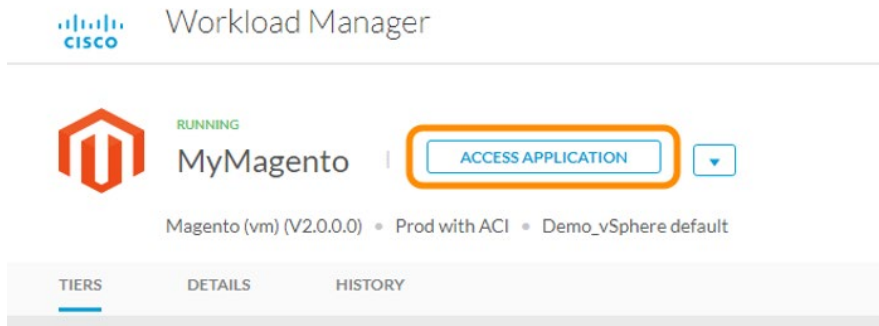
Verify Application Deployment

- On the left menu, click **Deployments**.
- Click **MyMagento** on the previously deployed application (the rest of this scenario is based on MyMagento).

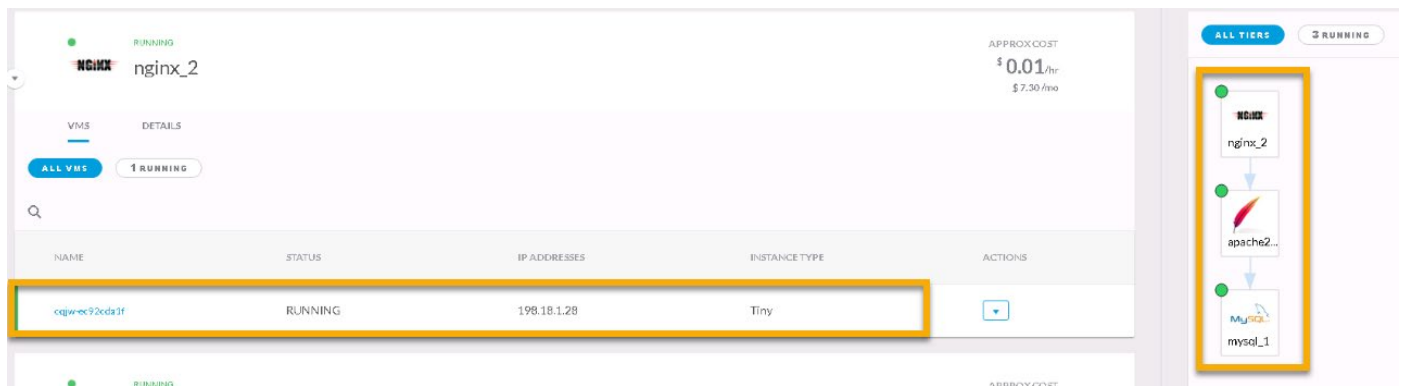
NOTE: Wait until the applications Status is Deployed before continuing.


Deployments						
All Deployments		All Applications		<input type="checkbox"/> Show Hidden		
NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	
MyMagento Magento (vm) (V2.0.0.0) Demo_vSphere default	Deployed	Prod with ACI	Mar 27, 2019 at 09:40 AM	Not Available	\$0.03/hr \$21.90/mo	
PetClinic-Pipeline PetClinic (pipeline) (V1) Demo_vSphere default	Deployed	Prod with ACI	Mar 22, 2019 at 04:27 PM	9 days 14 hrs	\$0.02/hr \$14.60/mo	

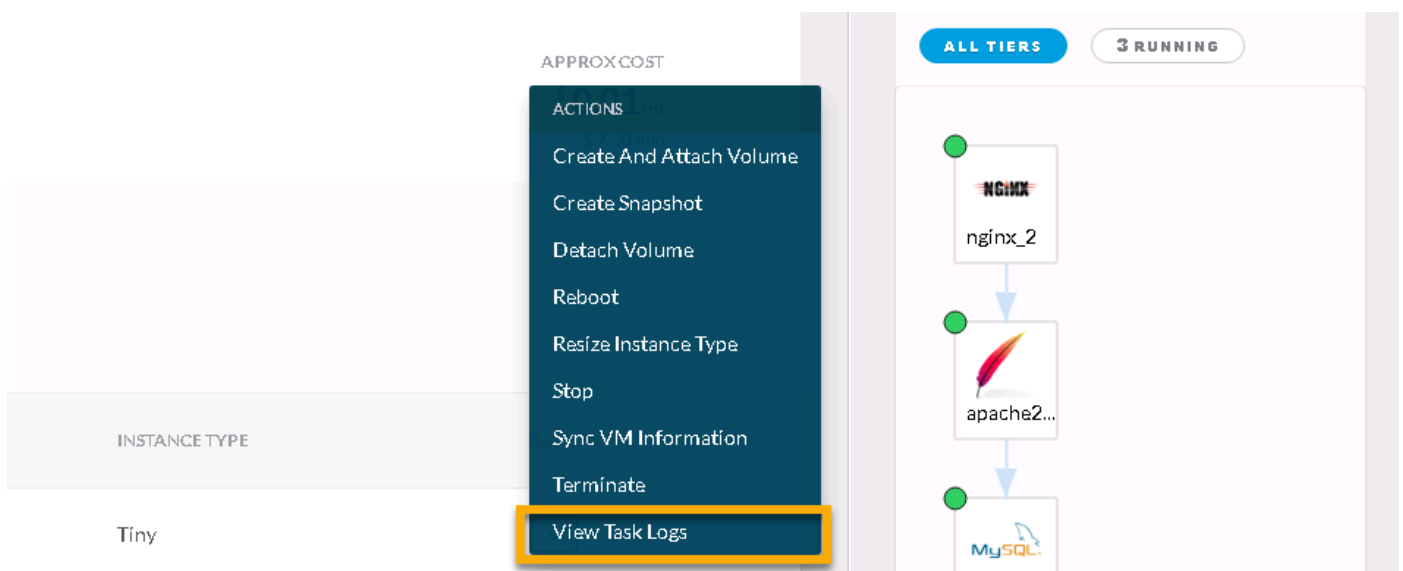
14. On the resulting screen, click the **Access Application** button to proceed to the application.



15. When the Magento page is displayed in the Chrome tab, **close the tab**. (Magento configuration is outside the scope of this demonstration.)
16. Return to the **Cisco CloudCenter** tab and observe the green circles indicating the health of the deployed services. Then observe the allocated VM names and IP Addresses. Finally, observe the current status of the VMs.



17. Next, click at the  icon underneath the **Actions** to view the available actions for this server. Click on the **View Task Logs** option to view the actions performed against this server during deployment.

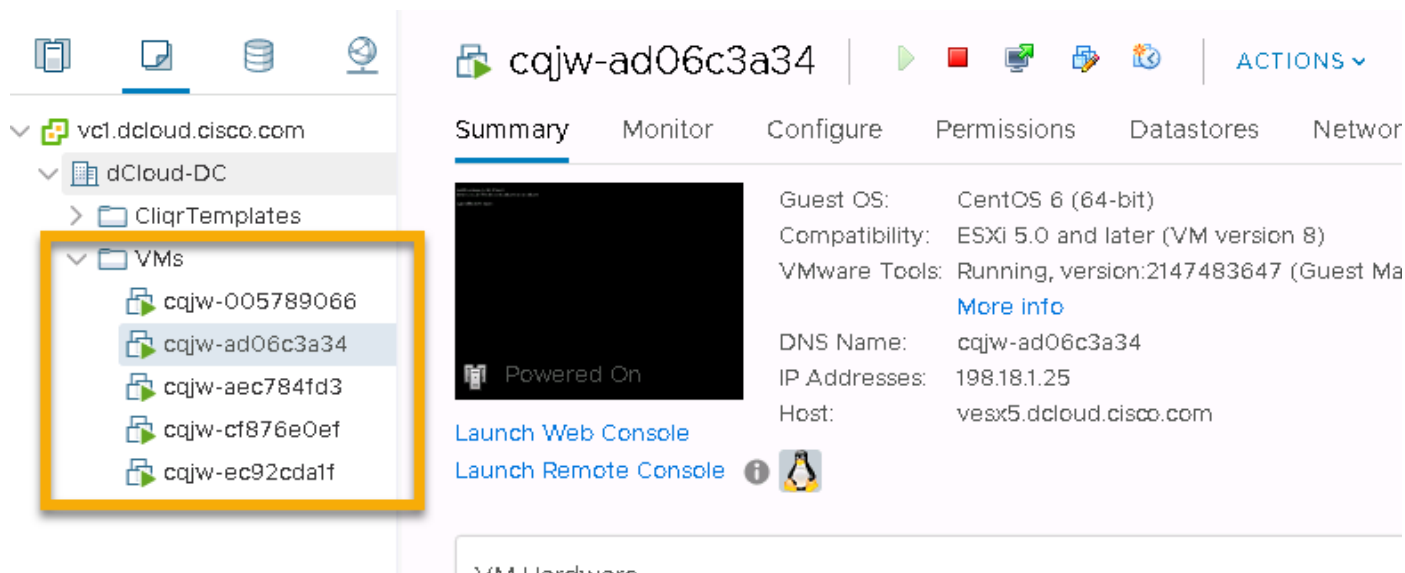


18. Switch to the **vSphere browser** tab. Log on again if the session has been timed out.

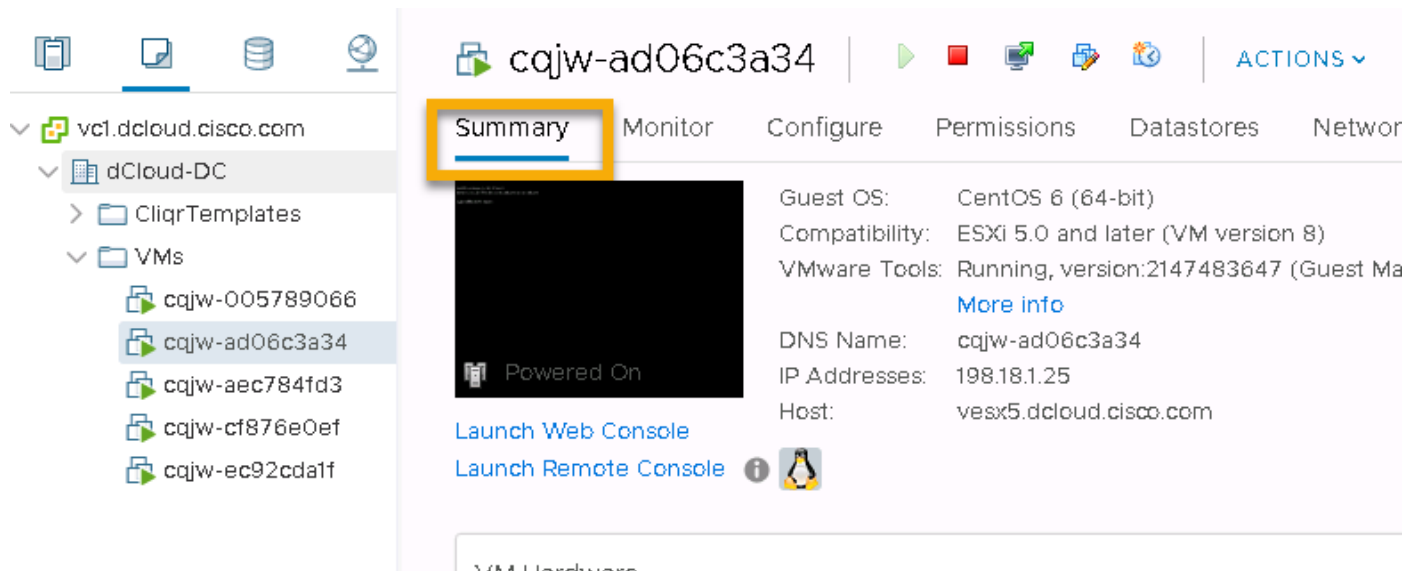
19. Click the **VMs and Templates** icon.



20. Expand the **dCloud-DC data center tree** and open the **VMs folder**. The VMs from the deployment should be in the folder vm list.



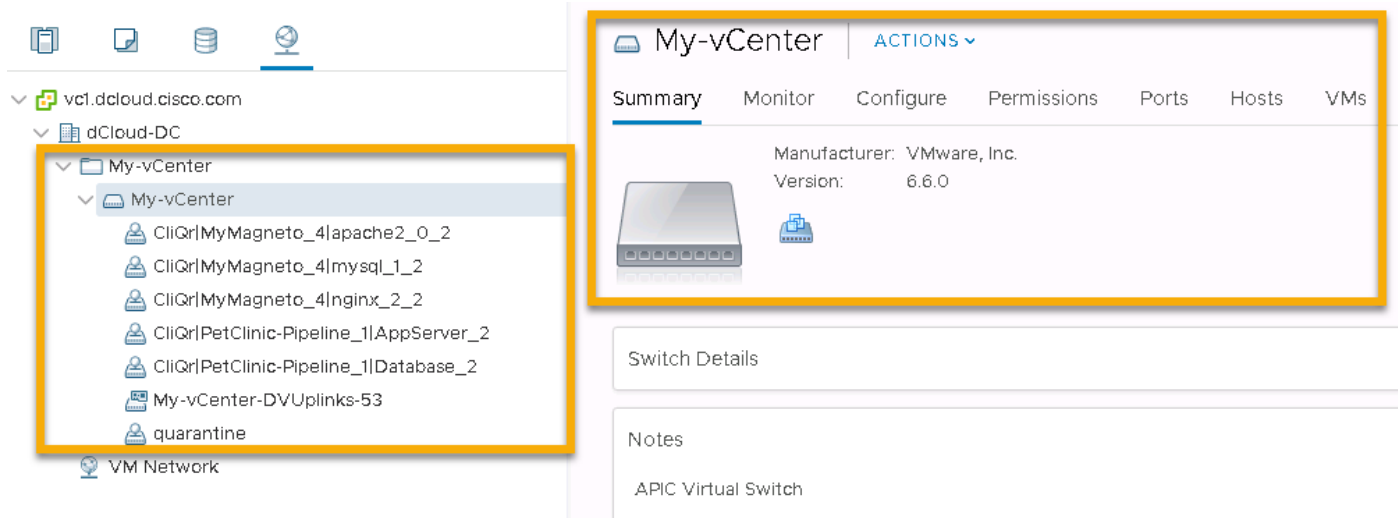
21. Click on one of the **deployment VMs** and click the **Summary** tab.



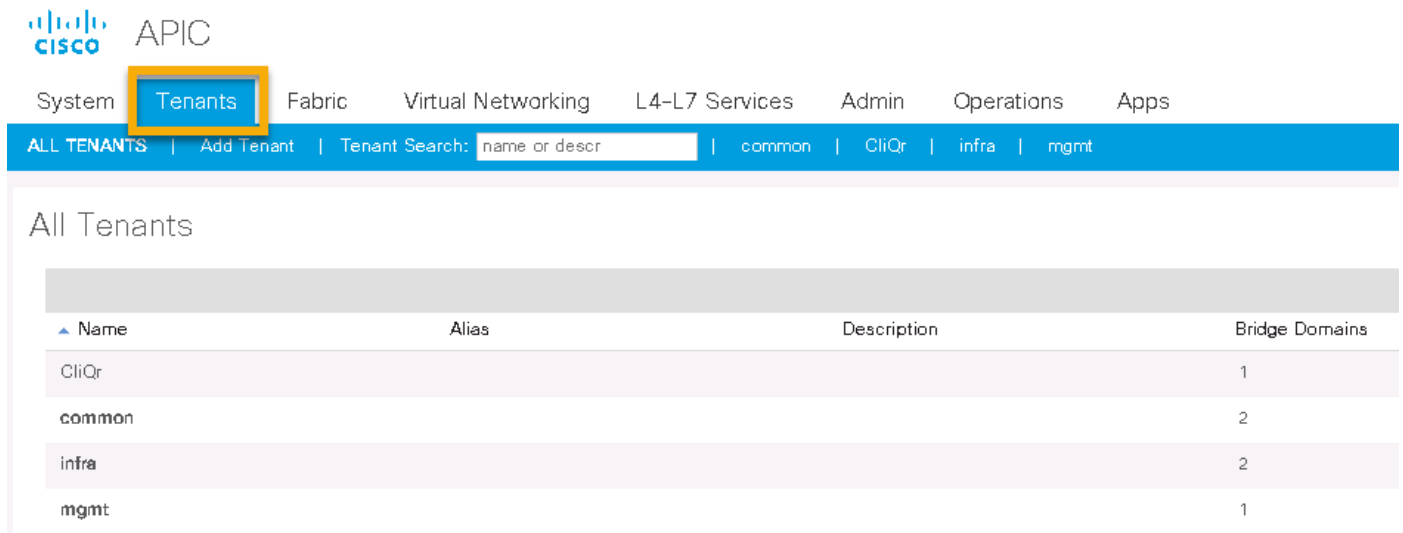
22. Click on one **Networking** icon to inspect the current network configuration



23. Expand the dCloud-DC data center tree and open the **My-vCenter** folder, then open the **My-vCenter switch**. Notice the ACI configuration that has been deployed for this application (CliQr|MyMagento_XXX|{tier name}).



24. Next, switch to the Cisco APIC browser tab (login again if the session has been timed out) and click on **Tenants**.



25. Click on the **CliQr Tenant**.
26. Under Tenant CliQr, open the **Application Profiles folder**, then the **MyMagento_XXX app profile** and then the **Application EPGs folder**.
27. Notice the newly created EPGs that correspond to the Application Tiers deployed from CloudCenter.

Name	Alias	Description	Class ID	Preferred Group Member
apache2_0_2			49154	Exclude
mysql_1_2			16390	Exclude
nginx_2_2			32770	Exclude

28. To view the currently configured ACI policy for communication between the EPGs, click on the **MyMagento_XXX app profile** and then click **Topology** at the right-hand side of the window.

Application Profile - MyMagento_4

Summary **Topology** Policy State Health Faults History

CloudCenter de... (CDO) CloudCenter de... (CDO)

mysql_2_2 (MyMagento_4) nginx_2_2 (MyMagento_4) mysql_1_2 (MyMagento_4)

Relation Indicators

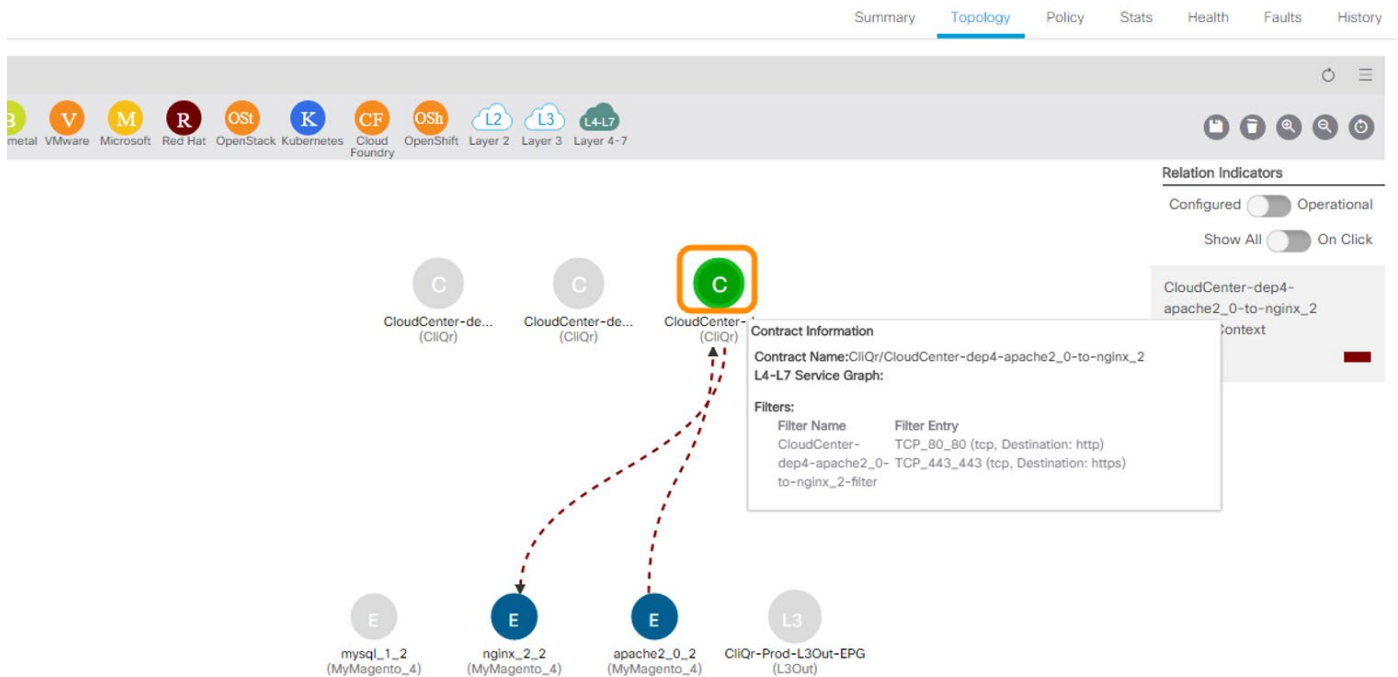
Configured ☐ Operational ☐

Show All ☐ On Click ☐

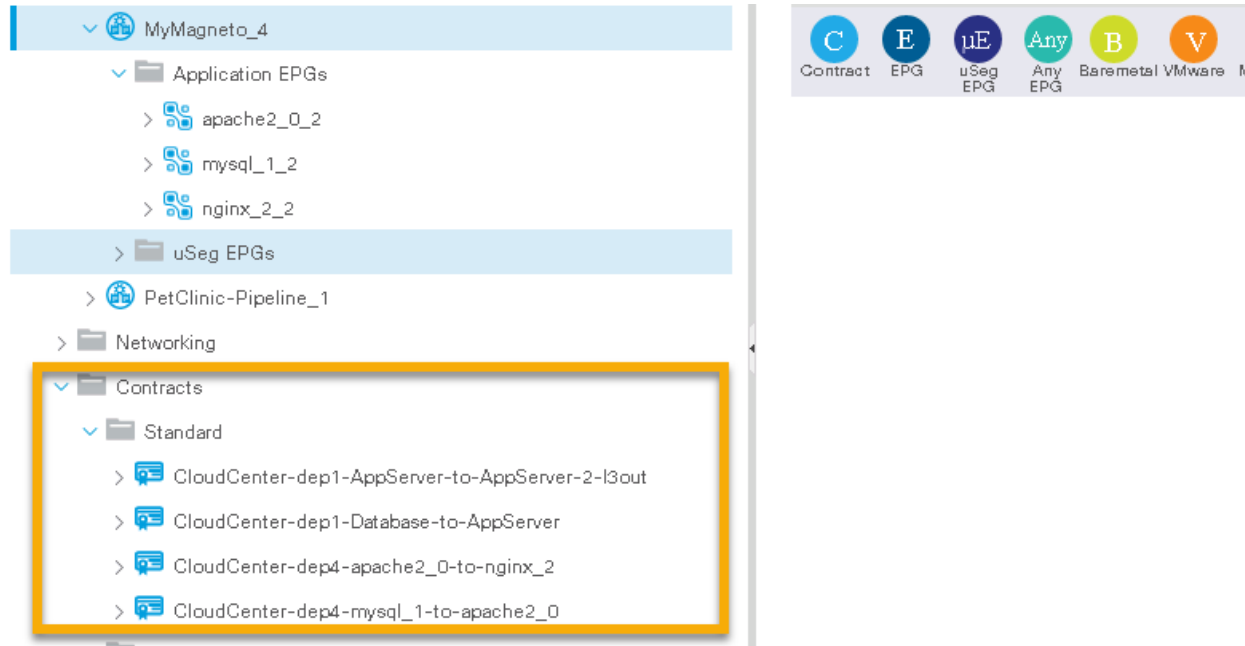
Provider ☐ Consumer ☐ Intra EPG ☐ Provider (from Master) ☐ Consumer (from Master) ☐ Intra EPG (from Master) ☐ Master EPG ☐

Cancel Submit

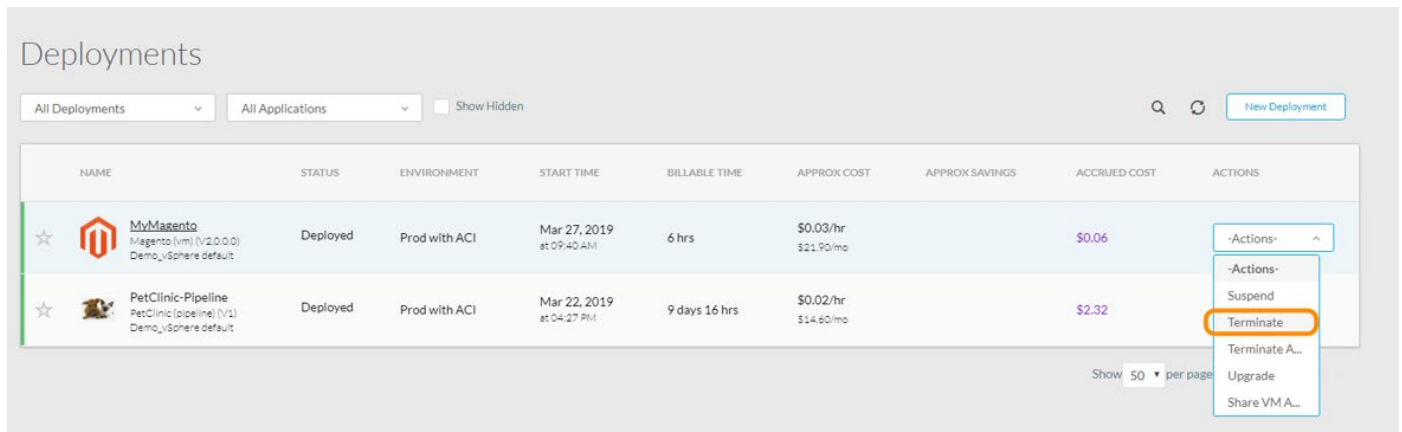
29. Click on one of the contracts the view more details on the traffic policy being applied by the specific contract.





30. Next, expand the **Contracts** folder to view more details about the contracts configured on the ACI, including the contracts created by the CloudCenter deployment.



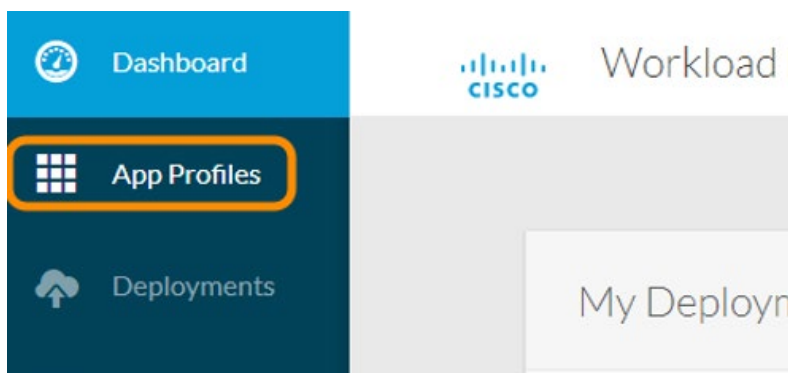
31. Optionally, return to the CloudCenter deployments and terminate the MyMagento deployment (it may take a few minutes for the deployment to be terminated). Then switch back to the Cisco APIC web page and observe the ACI objects related to the CloudCenter deployment having been removed.



NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	APPROX SAVINGS	ACCRUED COST	ACTIONS
 MyMagento Magento (vm) (V2.0.0.0) Demo_vSphere default	Deployed	Prod with ACI	Mar 27, 2019 at 09:40 AM	6 hrs	\$0.03/hr \$21.90/mo		\$0.06	-Actions- Suspend Terminate Terminate A... Upgrade Share VM A...
 PetClinic-Pipeline PetClinic (pipeline) (V1) Demo_vSphere default	Deployed	Prod with ACI	Mar 22, 2019 at 04:27 PM	9 days 16 hrs	\$0.02/hr \$14.60/mo		\$2.32	

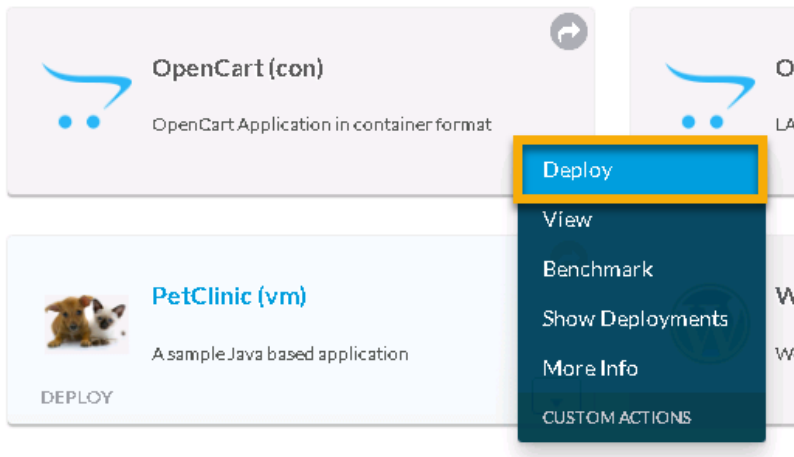
Deploy a Container-Based Application in Production

1. Open a Chrome browser and perform the following actions:
 - a. Open a new browser tab and click the **CloudCenter Suite** bookmark. Log in using the credentials user@dcloud.cisco.com / C1sco12345, tenant id: demo.
 - b. Open a second browser tab and click the **Kubernetes Dashboard** bookmark. Select the Token authentication method. Locate the file **Demo Kubernetes Cluster - clusteradmin Token.txt** saved on the desktop and copy/paste its contents to the **Token** field, then click **Sign In**.
2. Switch to the **CloudCenter** browser tab.
3. If not already open to **Workload Manager** screen, click on the **Dashboard** module.
4. From the side menu, click **App Profiles** to display the available Application Profiles.

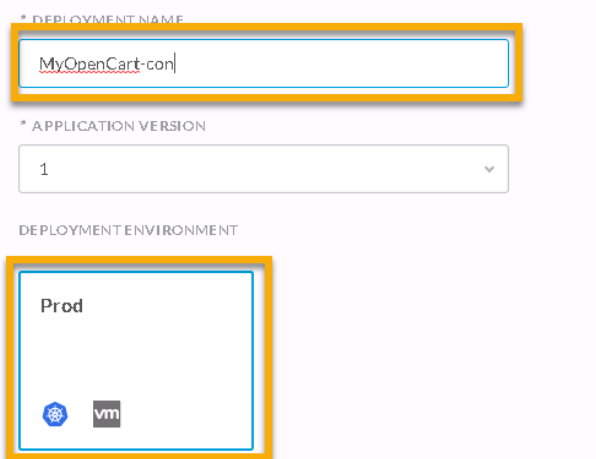


NOTE: Cisco CloudCenter provides role-based and object-based access control, so that when a user logs in the content provided to them is based on their profile.

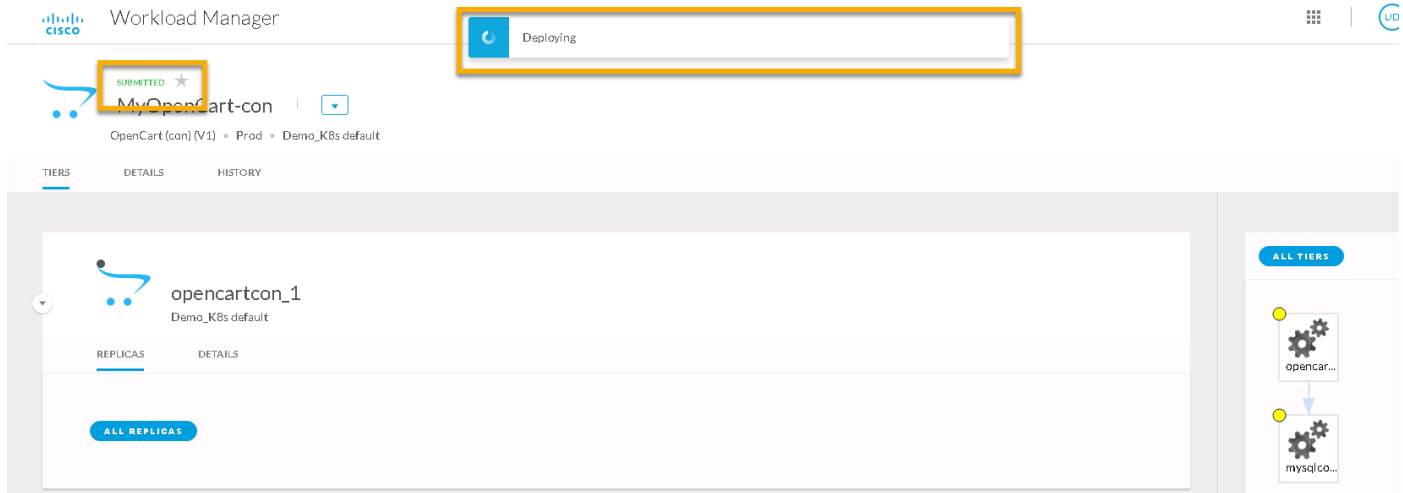
5. Mouse over the **OpenCart (con)** Application Profile and click **Deploy** to deploy an OpenCart application.



6. Enter **MyOpencart-con** at the Deployment Name field.
7. Choose **Prod** at the Deployment Environment.

A screenshot of the deployment form. The 'DEPLOYMENT NAME' field is highlighted with a blue border and contains the text 'MyOpenCart-con'. Below it, the 'APPLICATION VERSION' dropdown menu is set to '1'. The 'DEPLOYMENT ENVIRONMENT' section shows a list of environments, with 'Prod' highlighted by a blue border. The 'Prod' environment is represented by a blue box with the text 'Prod' and two small icons at the bottom: a gear icon and a 'vm' icon.

8. Leave the other fields to their default values and click **Next** at the bottom of the form.
9. At the next form, explore the default selected options (Cloud, 2 Tiers, Namespace, Instance Types) and then click **Deploy**.
10. The screen displays the deployment form, where each tier will be deployed in sequence. Once the application is fully deployed, an **Access Application Button** should appear at the top of the form.



- Wait for the application to fully deploy and proceed with the verification steps below. To verify the application using the Access Application button, skip to Step 3 in the Verify Application Deployment section

NOTE: If you pre-configured an application prior to the demonstration, you can proceed immediately with the following steps. If not, it may take up to 5 minutes for the application to fully deploy and show the **Access Application** button.

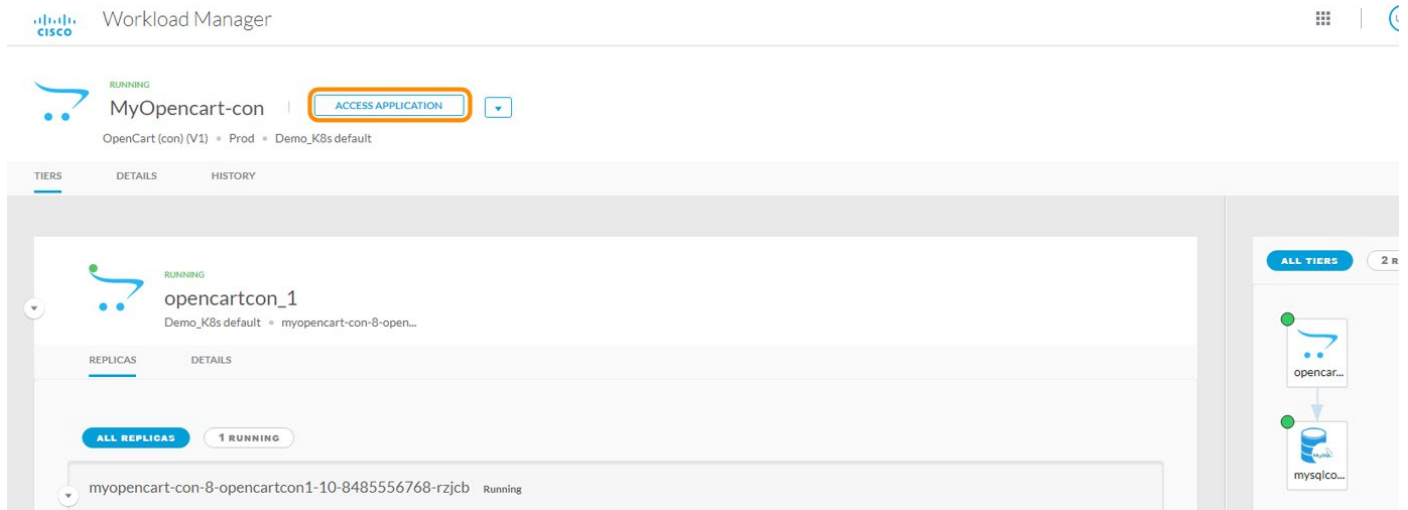
Verify Application Deployment

- On the left menu, click **Deployments**.
- Click **MyOpencart-con** on the previously deployed application (the rest of this scenario is based on MyOpencart-con).

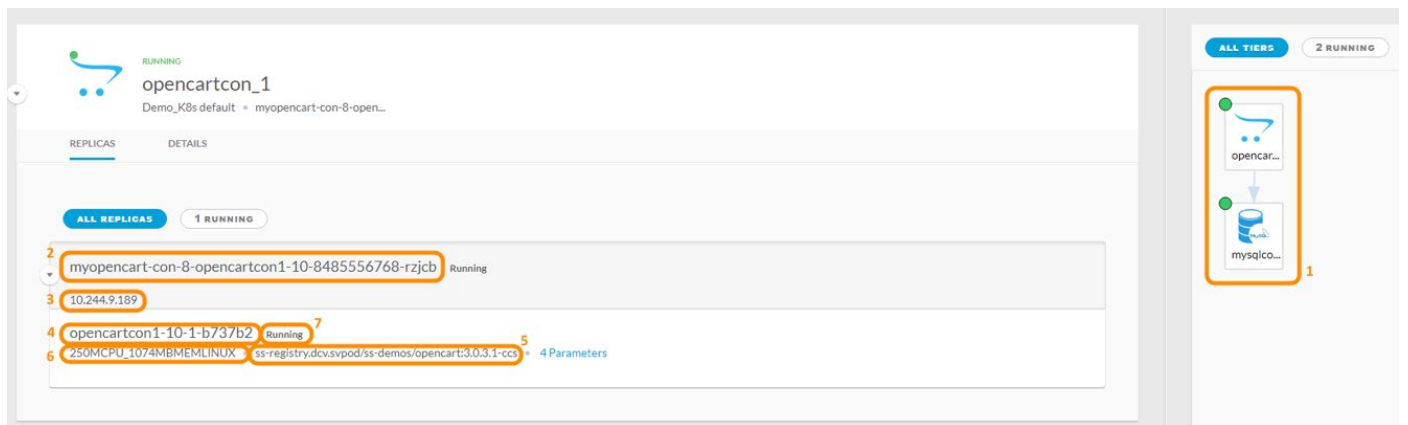
NOTE: Wait until the applications Status is Deployed before continuing.

All Deployments		All Applications		Show Hidden					New Deployment	
NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	APPROX SAVINGS	ACCRUED COST	ACTIONS		
MyOpencart-con OpenCart (con) (V1) Demo_K8s default	Deployed	Prod	Mar 27, 2019 at 11:16 AM	Not Available	\$0.00/hr \$0.00/mo		\$0			
MyMagento Magento (vm) (V2.0.0.0) Demo_VSphere default	Deployed	Prod with ACI	Mar 27, 2019 at 09:40 AM	12 hrs	\$0.03/hr \$21.90/mo		\$0.12			
PetClinic-Pipeline PetClinic (pipeline) (V1) Demo_VSphere default	Deployed	Prod with ACI	Mar 22, 2019 at 04:27 PM	9 days 18 hrs	\$0.02/hr \$14.60/mo		\$2.34			

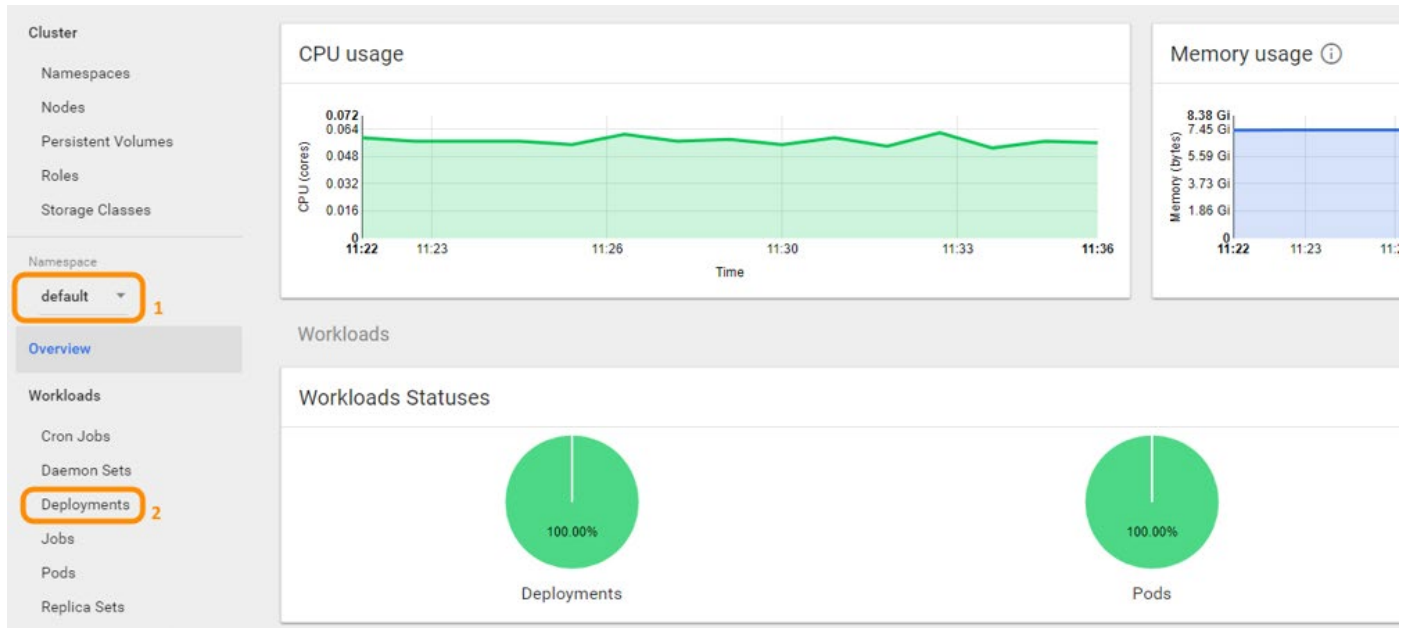
- On the resulting screen, click the **Access Application** button to proceed to the application



4. When the **Opencart** page is displayed in the Chrome tab, close the tab. (Opencart configuration is outside the scope of this demonstration.)
5. Return to the **Cisco CloudCenter** tab and observe the green circles indicating the health of the deployed services . Then observe the allocated container names, IP Addresses, deployment names, and source container registry . Finally, observe the container resource requests and current status .



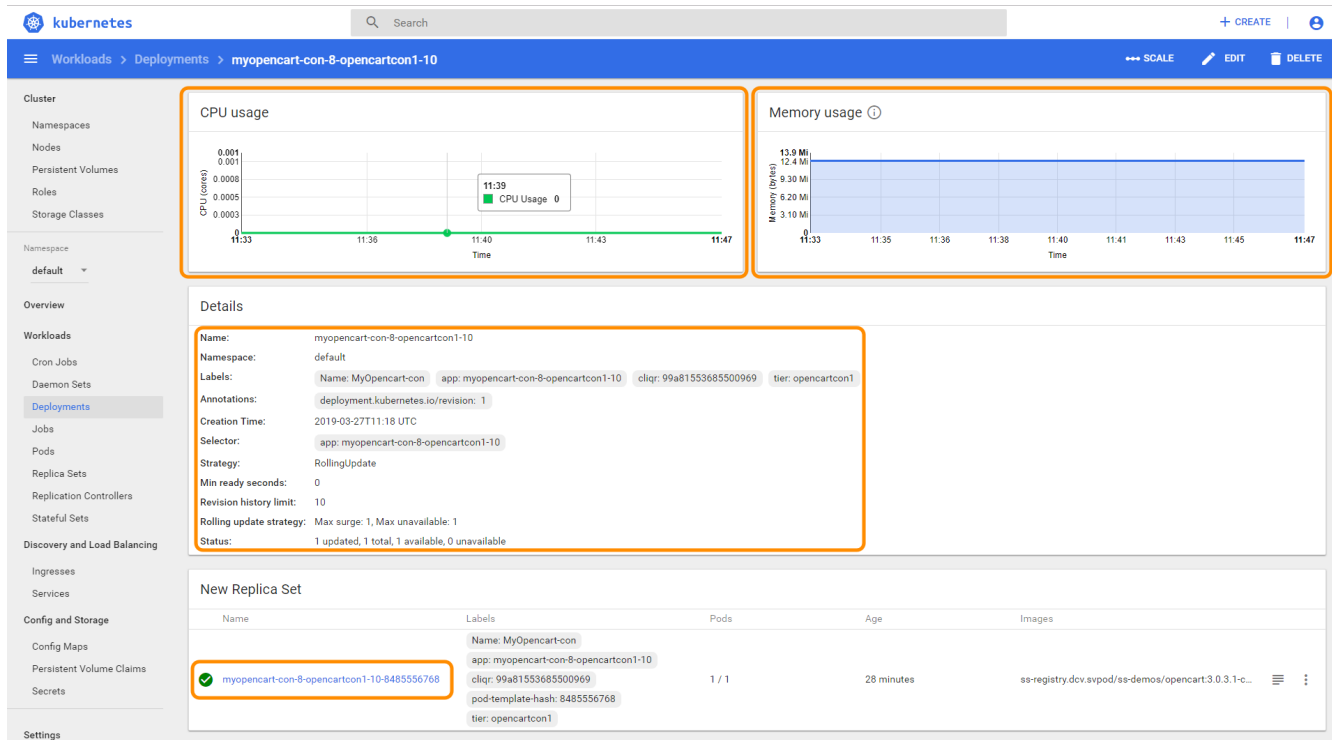
6. Note down the container names used for this deployment and switch to the **Kubernetes Dashboard** browser tab.
7. Ensure the select **Namespace** is **Default** and click on **Deployments**.



8. Observe the two new deployments present on the list, the **myopencart-con-XX-opencartcon1-YY** and **myopencart-con-XX-mysqlcon1-ZZ** (the XX, YY and ZZ numbers can vary).

Name	Labels	Pods	Age	Images
✓ myopencart-con-8-opencartcon1-10	Name: MyOpencart-con app: myopencart-con-8-opencartcon1-10 cliqr: 99a81553685500969 tier: opencartcon1	1 / 1	19 minutes	ss-registry.dcv.svpod/ss-demos/opencart:3.0.3.1-c...
✓ myopencart-con-8-mysqlcon1-9	Name: MyOpencart-con app: myopencart-con-8-mysqlcon1-9 cliqr: 54671553685410645 tier: mysqlcon1	1 / 1	21 minutes	ss-registry.dcv.svpod/ss-demos/mysql:5.7
✓ jenkins-slave-mvn3-docker	app: jenkins-slave-mvn3-docker	1 / 1	26 days	ss-registry.dcv.svpod/ss-demos/jnlp-slave-with-mv...
✓ gitlab	app: gitlab	1 / 1	26 days	ss-registry.dcv.svpod/ss-demos/gitlab-ce:latest
✓ artifactory	app: artifactory	1 / 1	26 days	ss-registry.dcv.svpod/ss-demos/artifactory-oss:lat...
✓ jenkins	app: jenkins	1 / 1	26 days	ss-registry.dcv.svpod/ss-demos/jenkins:its

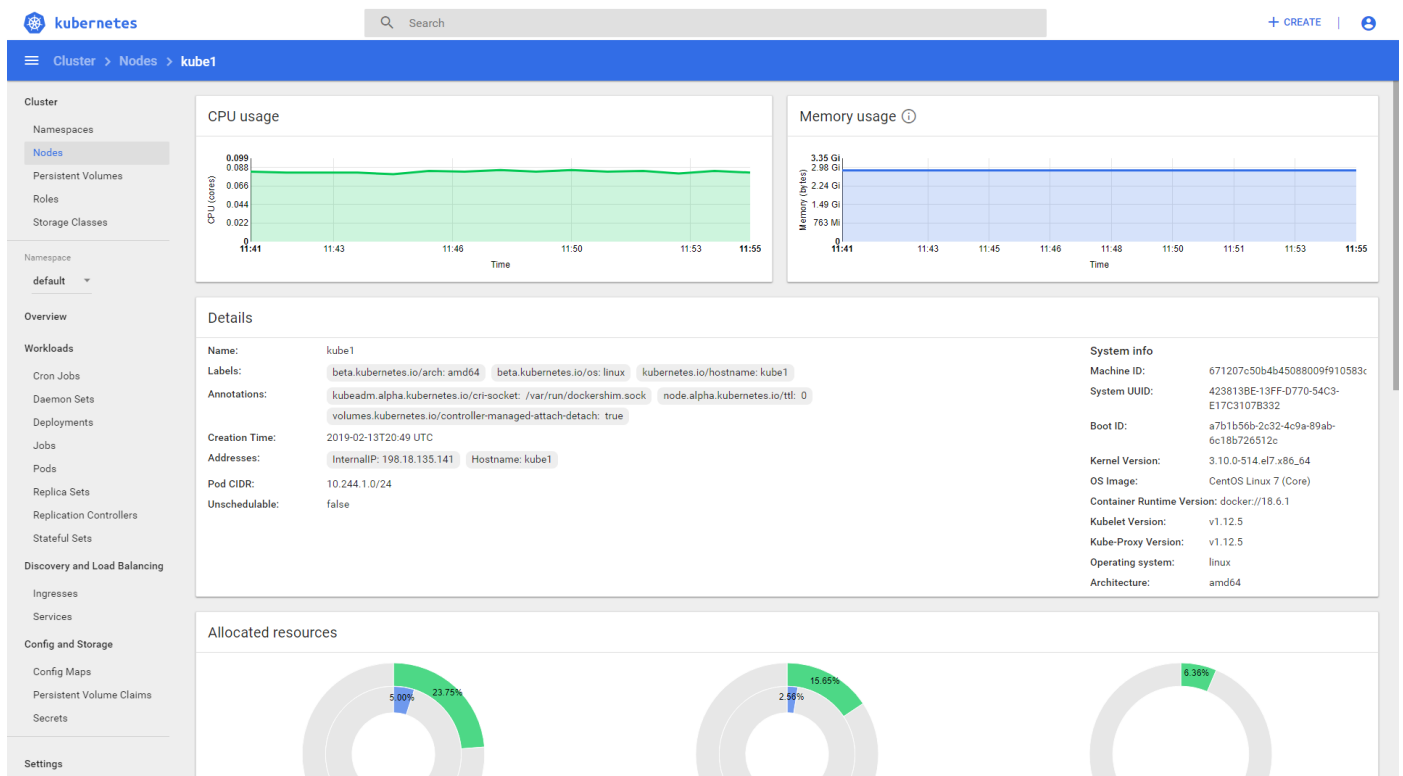
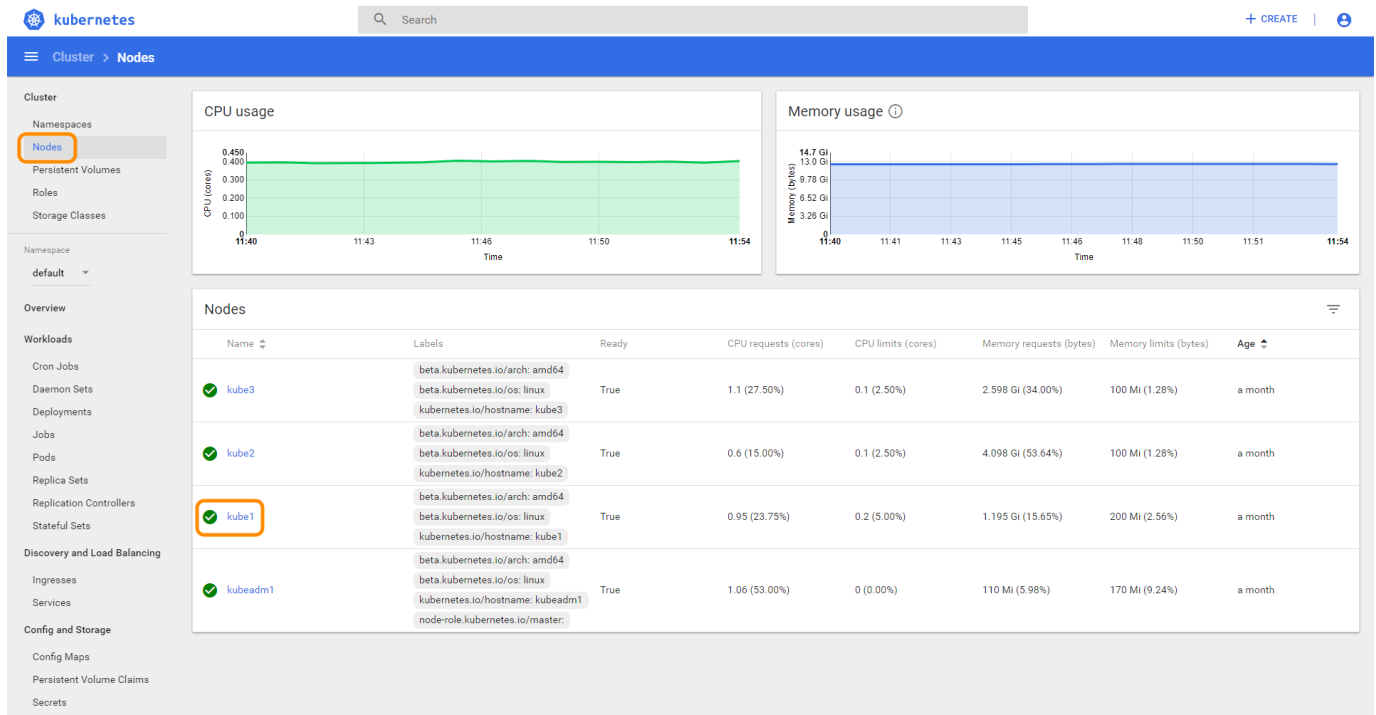
9. To obtain container-level information, **click on one of the deployments**, e.g. **myopencart-con-XX-opencartcon1-YY**. Observe the current CPU and Memory usage for the container, as well as the configured settings (Labels, Annotations, Strategy, etc.)



10. To review the allocated service IP to the **MyOpencart-con container deployment**, click on **Services**. To locate the service that links to the deployment, find the service that contains the labels `app:myopencart-con-XX-opencartcon1-YY` and `tier:opencartcon1`. The External endpoints column should list the same IP address as the **Access Application** in the **CloudCenter Deployment**.

Cluster	Services					
Namespaces	Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
Nodes	web-10-b737b2	app: myopencart-con-8-opencartcon1-10 cliqr: 99a81553685500969 tier: opencartcon1	10.103.77.112	web-10-b737b2:80 TCP web-10-b737b2:32112 TCP	198.18.135.153:80	33 minutes
Persistent Volumes	mysql-9-34ed5c	app: myopencart-con-8-mysqcon1-9 cliqr: 54671553685410645 tier: mysqlcon1	10.110.122.255	mysql-9-34ed5c:3306 TCP	-	35 minutes
Roles	gitlab	app: gitlab	10.102.130.92	gitlab:80 TCP gitlab:32218 TCP gitlab:443 TCP gitlab:30079 TCP	198.18.135.151:80 198.18.135.151:443	26 days
Storage Classes	artifactory	app: artifactory	10.108.116.41	artifactory:8081 TCP artifactory:30962 TCP	198.18.135.152:8081	26 days
	jenkins	app: jenkins	10.110.61.109	jenkins:80 TCP jenkins:30083 TCP jenkins:50000 TCP jenkins:31380 TCP	198.18.135.150:80 198.18.135.150:50000	26 days
Namespace	kubernetes	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP	-	a month
default						
Overview						
Workloads						
Cron Jobs						
Daemon Sets						
Deployments						
Jobs						
Pods						
Replica Sets						
Replication Controllers						
Stateful Sets						
Discovery and Load Balancing						
Ingresses						
Services						

11. To observe the overall cluster CPU and Memory usage, click on **Nodes**. To drill down on the resource usage per node, click on each node name (e.g. kube1)



- Optionally, return to the CloudCenter deployments and terminate the MyOpencart-con deployment (it may take a few minutes for the deployment to be terminated). Then switch back to the Kubernetes Dashboard web page and observe the deployment, pod and service objects related to the CloudCenter deployment having been removed.

Deployments

All Deployments ▾ All Applications ▾ ☐ Show Hidden 🔍 ↻ [New Deployment](#)

	NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	APPROX SAVINGS	ACCURED COST	ACTIONS
★	MyOpenCart-con OpenCart (con) (V1) Demo_K8s default	Deployed	Prod	Mar 27, 2019 at 11:16 AM	Not Available	\$0.00/hr \$0.00/mo		\$0	-Actions- -Actions- Terminate Terminate A... Share VM A...
★	MyMagento Magento (vm) (V2.0.0.0) Demo_vSphere default	Deployed	Prod with ACI	Mar 27, 2019 at 09:40 AM	12 hrs	\$0.03/hr \$21.90/mo		\$0.12	
★	PetClinic-Pipeline PetClinic (pipeline) (V1) Demo_vSphere default	Deployed	Prod with ACI	Mar 22, 2019 at 04:27 PM	9 days 18 hrs	\$0.02/hr \$14.60/mo		\$2.34	

Dashboard Overview

The purpose of this section is to review the dashboard now that more than one application is deployed.

- Return to CloudCenter and click **Dashboard** on the side menu. If the Magento application is not reflected in the graphs yet, Refresh until it is. It may take up to ten minutes until the Magento application is fully deployed and displayed on the Dashboard.
- Review and explain the graphs.

My Deployments (last 30 days)

Cloud Cost: 1.25 HRS
VMware: 100%
VM LIST

Toggle View by Cloud vs View by Application

My Plan Usage
Unlimited Subscription

Recent Deployments

NAME	ENVIRONMENT
WP1	
MySplunk	vCenter vCenter-DC-West

[VIEW MORE DEPLOYMENTS](#)

VM Usage by Cloud Over Time

Number of VMs

Jun 1, 4PM Jun 1, 5PM

Cloud Status
UCSD UCSD Datacenter

Favorite deployments
You don't have any favorite deployments yet.

Notifications

- 06-01-2017, 05:08:17 The node 42207784-aab1-fa50-eb41-68f9936eddaa was started for the job 355
- 06-01-2017, 05:07:12 The node 4220fcae-eb15-99f2-0810-c1dc09fb93c5 was started for the job 356
- 06-01-2017, 05:05:27 The node 42205bdc9-be0e-4b6b-9dc6-2c1ed5b1693f was started for the job 354
- 06-01-2017, 04:58:02 Job 'WP1 apache2_1' is starting. [View](#)
- 06-01-2017, 04:56:20 Job 'WP1 haproxy_2' is starting. [View](#)

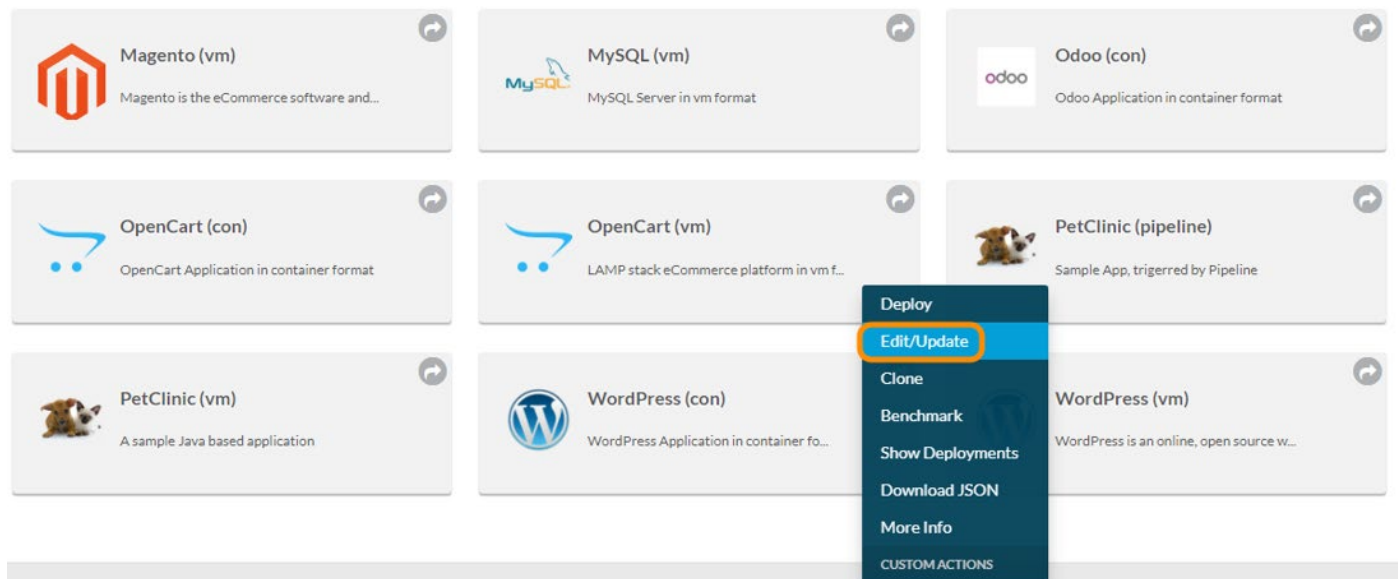
[VIEW MORE NOTIFICATIONS](#)

Scenario 2. Architect Role – Editing Application Profiles

The purpose of this section is to use the Edit/Update option to walk through an existing Application Profile, showing the different sections of an Application Profile and their role, as well as to demonstrate a typical scenario of reconfiguring an application for scalability. The Edit/Update option is only available to Architects/Power Users.

Steps

1. Navigate to the **CloudCenter** UI, login as **poweruser@dcloud.cisco.com / C1sco12345**, tenant: **demo**
2. Select **Workload Manager**.
3. Click on **App Profiles**.
4. Mouse over **Opencart (vm)** and choose **Edit/Update**.



The Edit window is divided into three tabs:

- **Basic Information** – Allows the architect to set parameters such as name, version, application category, tags, URL, and micro segmentation, as well as choose the application icon that will appear with the application in the Application Profile catalog.
- **Global Parameters** – Allows the Architect to configure a list of information that the user will supply during application deployment.
- **Topology Modeler** – Allows the Architect to select a list of services that will be dynamically deployed and configured in when the application profile is deployed.

5. Navigate to the **Topology Modeler** tab.

Edit OpenCart (vm) Application Template

Version: 1.0 > 2.0 > 3.0 > 4.0 (Revision: 1)

Basic Information Global Parameters **Topology Modeler**

Services

OS Service

Orchestration

Message Bus

Web Server

Apache2
Open-source HTTP server for OS

Apache2 Conta...
Apache2 Running in Container

Geronimo3
Open source application server

IIS
Web server for Windows-based apps

Properties

Select a component to display its properties.

6. At the left-hand side of the form, under **Services**, expand **Load Balancer**.

Edit OpenCart (vm) Application Template

Version: 1.0 > 2.0 > 3.0 > 4.0 (Revision: 1)

Basic Information Global Parameters Topology Modeler

Services

OS Service

Orchestration

Message Bus

Web Server

Custom Service

Workflow

Frontend Cache

Load Balancer

Elastic Load Ba...
AWS Elastic Load Balancer

HAProxy
TCP/HTTP load balancer

Nginx

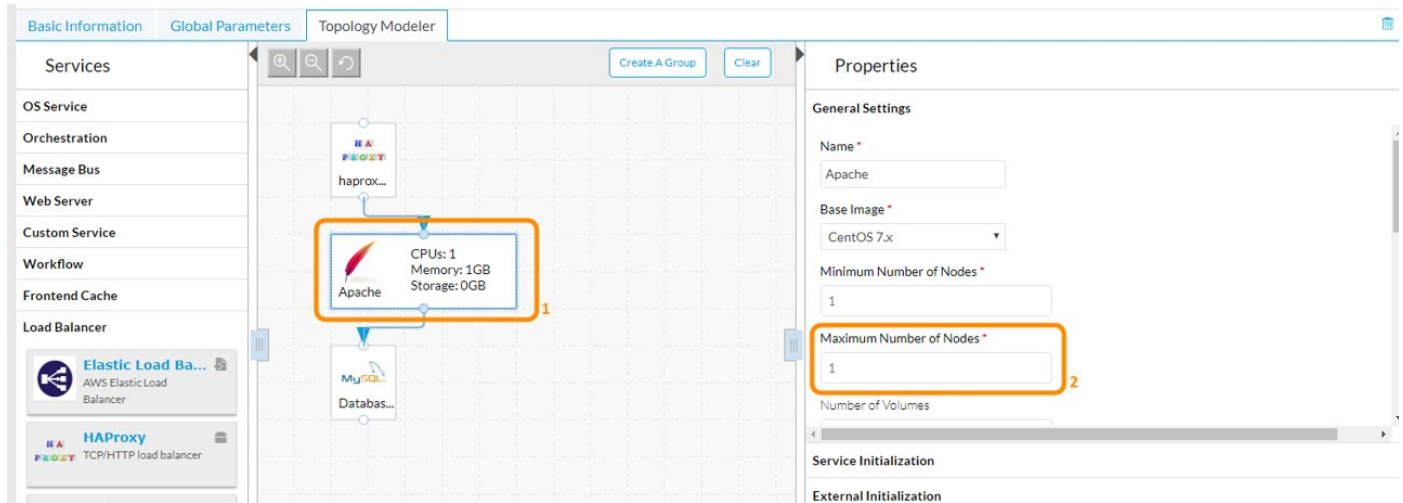
Properties

Select a component to display its properties.

7. Drag and drop **HAProxy** to the **Modeler** section.

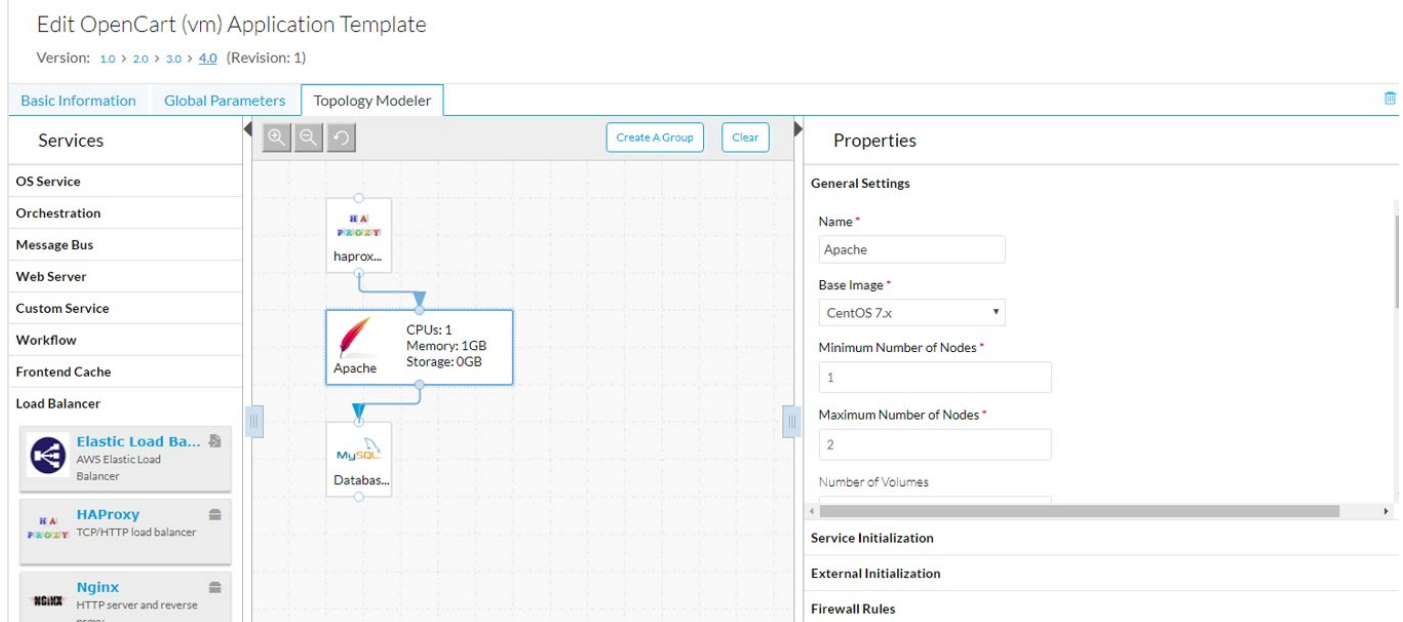
8. Link the **HAProxy** service to the **Apache** service (drag the bottom circle of the haproxy service to the top circle of the Apache service, an arrow should form).

- Click on the **Apache** service. At the right-hand side, under **Properties**, click in the **Maximum Number of Nodes** field. Change the value from 1 to 2.

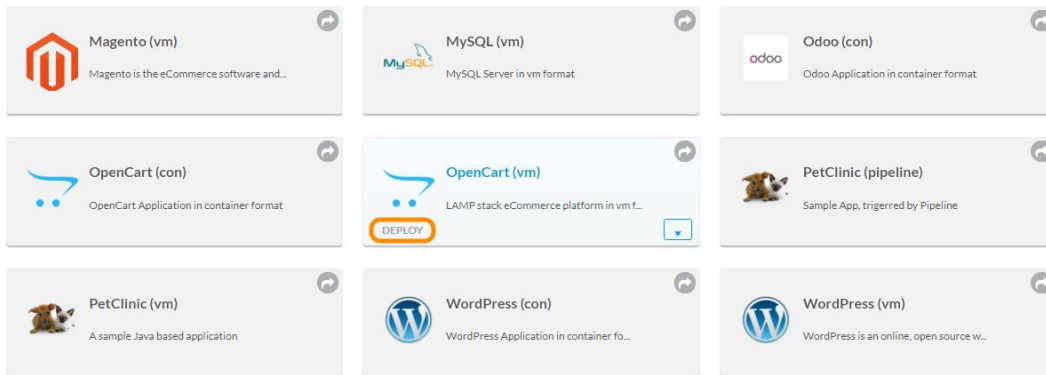


- Click **Save App**.

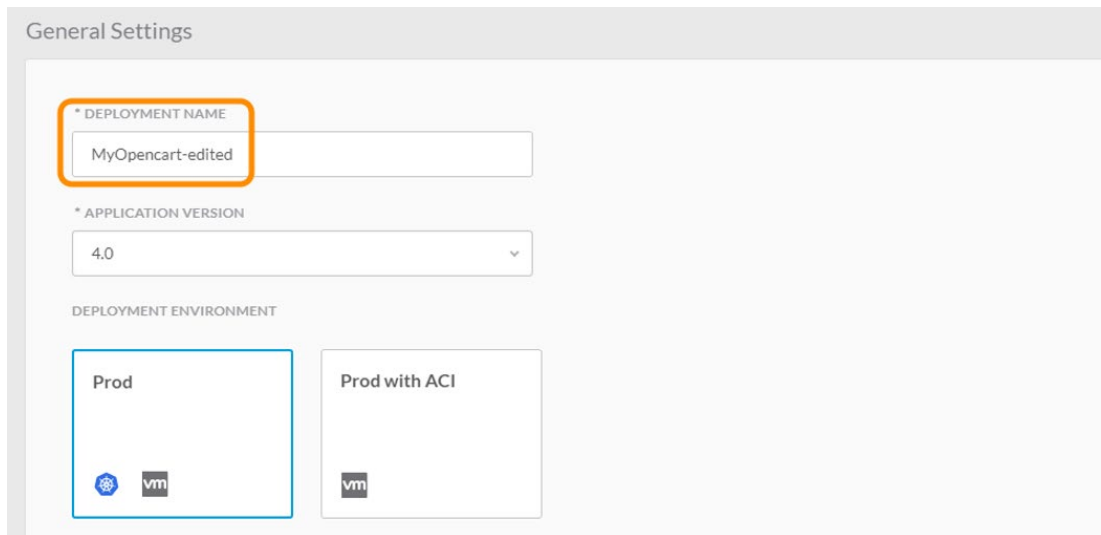
Edit OpenCart (vm) Application Template
Version: 1.0 > 2.0 > 3.0 > 4.0 (Revision: 1)



11. Click on **Opencart (VM)** to deploy a new application.



12. Enter **MyOpencart-edited** at the Deployment Name field.



The 'General Settings' form for deploying an application. It includes the following fields and options:

- * DEPLOYMENT NAME:** A text input field containing 'MyOpencart-edited', which is highlighted with an orange border.
- * APPLICATION VERSION:** A dropdown menu showing '4.0'.
- DEPLOYMENT ENVIRONMENT:** Two selectable options: 'Prod' (highlighted with a blue border) and 'Prod with ACI'. Both options include a 'vm' icon.

13. Leave the other fields to their default values and click **Next**.
14. Leave all the fields at their default values and click **Deploy**.

haproxy_1 \$ 0.01 /hour

Filter Instance Types / [SHOW](#)

AVAILABLE INSTANCE TYPES (4)

TINY	SMALL	MEDIUM	LARGE
1 VIRTUAL CPU	1 VIRTUAL CPU	2 VIRTUAL CPU	4 VIRTUAL CPU
1 GB MEMORY	2 GB MEMORY	2 GB MEMORY	4 GB MEMORY
\$ 0.01 /hour approx 7.30/month	\$ 0.02 /hour approx 14.60/month	\$ 0.04 /hour approx 29.20/month	\$ 0.08 /hour approx 58.40/month

Volumes

VOLUME	PRICE	SIZE
Root Disk ⓘ	Full clone has not been enabled. Root disk size cannot be manually configured.	

Cloud Settings

ENABLE FULL CLONE

☐ NO

15. Once the application is deployed, to verify its proper operation, click on the **Access Application** at the top of the form to verify the application is working properly. Close the application browser tab and return to the CloudCenter tab.

MyOpencart-edited | [ACCESS APPLICATION](#) APPROX COST \$ 0.03 /hr \$ 21.90 /mo

OpenCart (vm) (V4.0) • Prod • Demo_vSphere default

TIER DETAILS HISTORY

ALL VMs 1 RUNNING

NAME	STATUS	IP ADDRESSES	INSTANCE TYPE	ACTIONS
cgw-228d7043e	RUNNING	198.18.1.31	Tiny	SSH

ALL TIERS 3 RUNNING

```

graph TD
    Haproxy[haproxy] --> Apache[Apache]
    Apache --> MySQL[MySQL]
    MySQL --> Databases[Databases]
  
```

16. Locate the **Apache tier** and click at the down arrow next to it. Select **Add a VM**.

NAME	STATUS	IP ADDRESSES	INSTANCE TYPE	ACTIONS
cqjw-228d7043e	RUNNING	198.18.1.31	Tiny	SSH

17. The status of the deployment should change to Scaling Up. Wait for the deployment to automatically deploy the new VM and configure it, followed by reconfiguring the higher tier (haproxy) to accommodate for the change.

Scaling up

APPROX COST: \$0.03/hr (\$7.30/mo)

NAME	STATUS	IP ADDRESSES	INSTANCE TYPE	ACTIONS
cqjw-228d7043e	RUNNING	198.18.1.31	Tiny	SSH
cqjw-c391b2a0	STARTING	-	Tiny	VIEW TASK LOGS



18. Once the deployment has finished adding and reconfiguring the new resources, click **Access Application** at the top of the form to verify the Application is working properly.

MyOpencart-edited

ACCESS APPLICATION

APPROX COST
\$ 0.04/hr
\$ 29.20 /mo

OpenCart (vm) (V4.0) • Prod • Demo_vSphere default

TIERSDetailsHistory

Apache

APPROX COST
\$ 0.02/hr
\$ 14.60 /mo

VMSDETAILS

ALL VMS2 RUNNING

Q

NAME	STATUS	IP ADDRESSES	INSTANCE TYPE	ACTIONS
cqjw-228d7043e	RUNNING	198.18.1.31	Tiny	SSH
cqjw-c3391b2a0	RUNNING	198.18.1.32	Tiny	SSH

ALL TIERS3 RUNNING

haproxy

Apache

MySQL Database

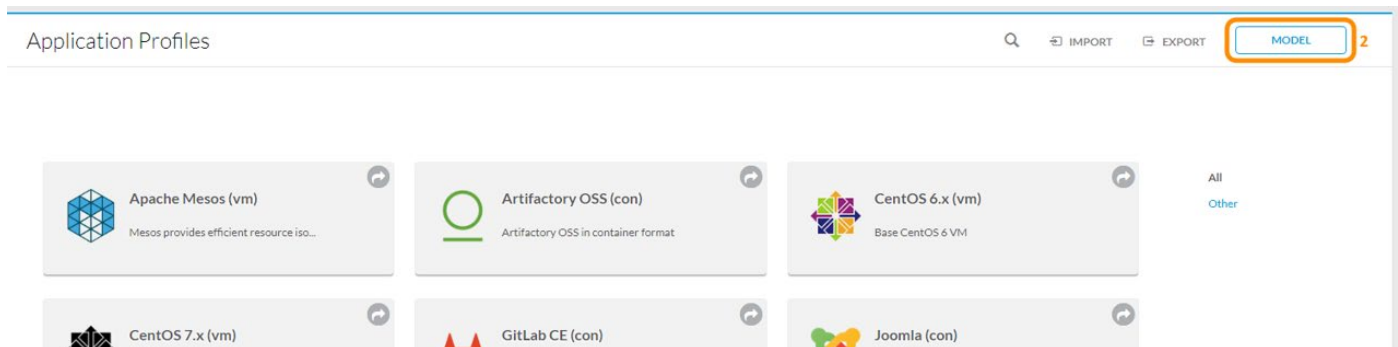
Scenario 3. Admin Role – Create Constructs Within CloudCenter

The purpose of this section is to demonstrate some of the functions of the Admin role:

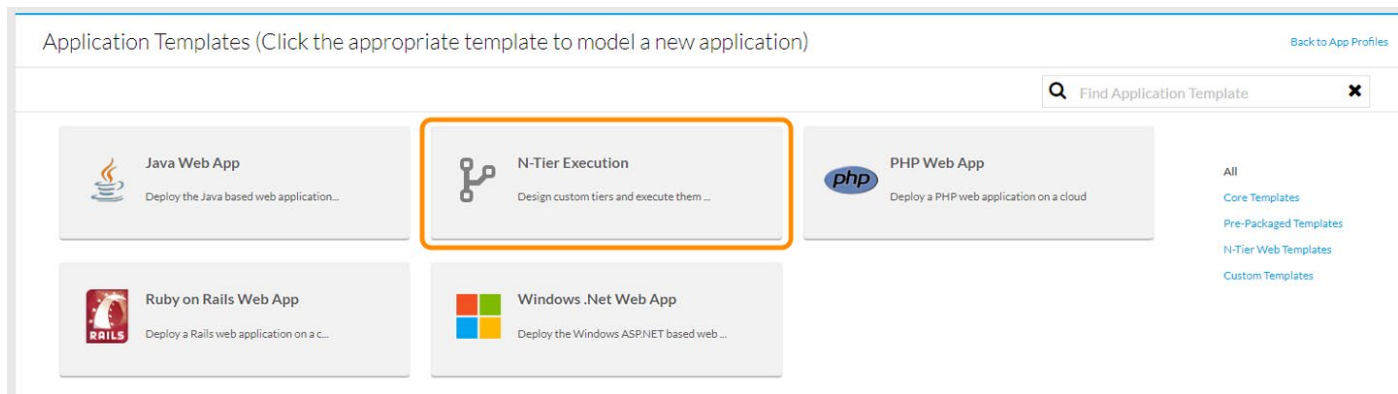
- Creating a VM-Based Application Profile
- Creating a Container-Based Application Profile
- Creating a Usage Plan
- Creating a Bundle

Creating a VM-Based Application Profile

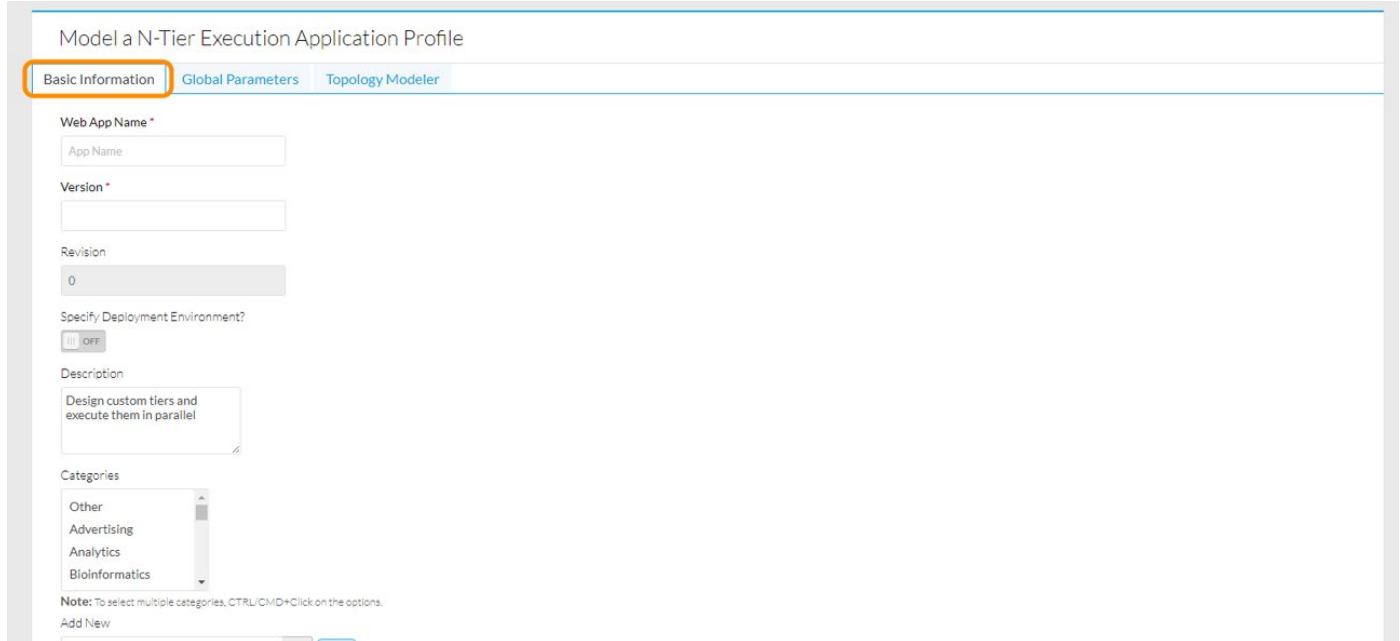
1. Navigate to the CloudCenter UI, login as **admin@dcloud.cisco.com / C1sco12345**, tenant: **demo**.
2. Select **Workload Manager**.
3. Click on **App Profiles**, then click on **MODEL**.



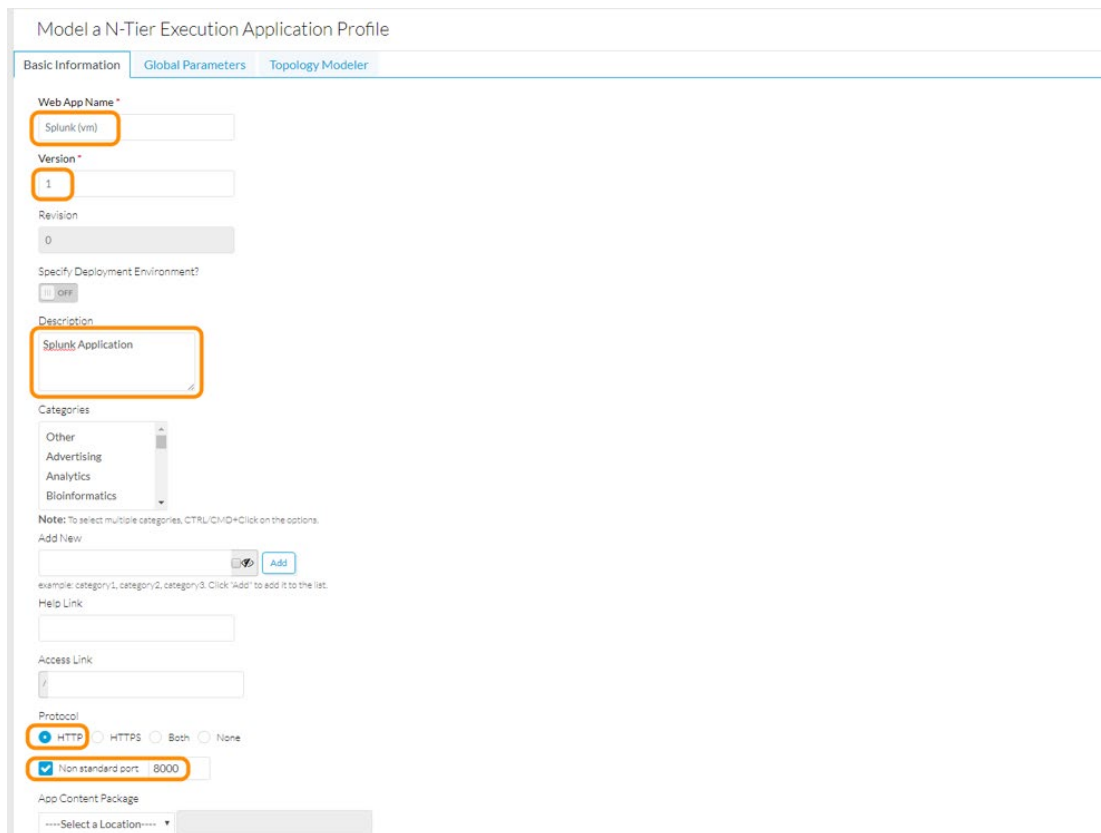
4. At the **Application Templates**, click on **N-Tier Execution**.



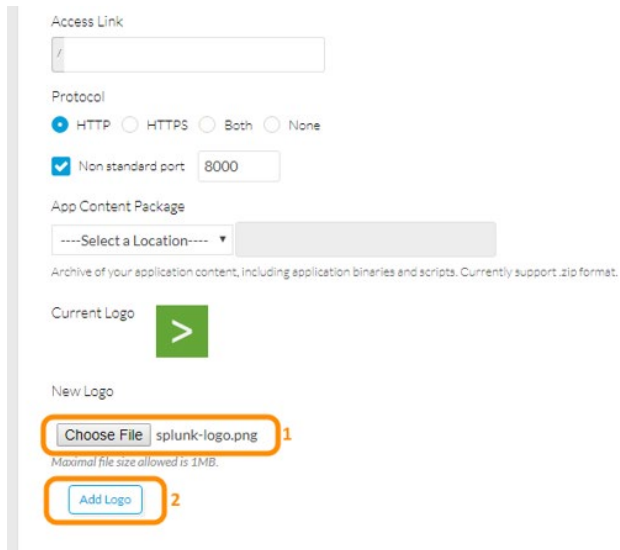
- Click on the **Basic Information** tab.



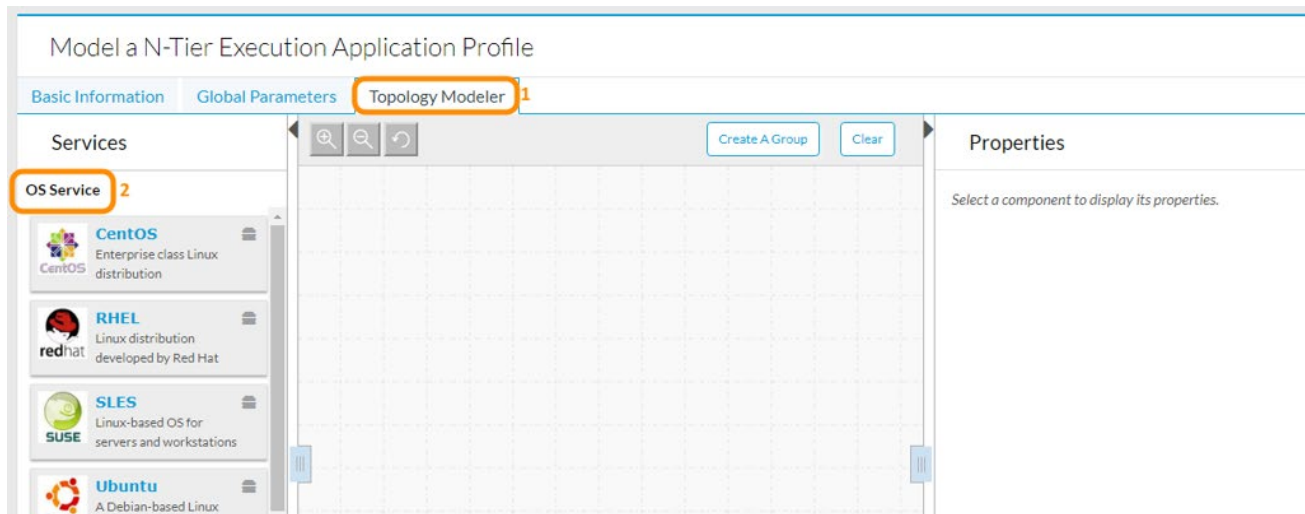
- Enter **Splunk (vm)** at the **Web App Name** field, **1** at the **Version** field and **Splunk Application** at the **Description**. Then, at the Protocol Section **ensure HTTP is selected** and **enable Non standard port**. Enter the value **8000** as the non-standard port.



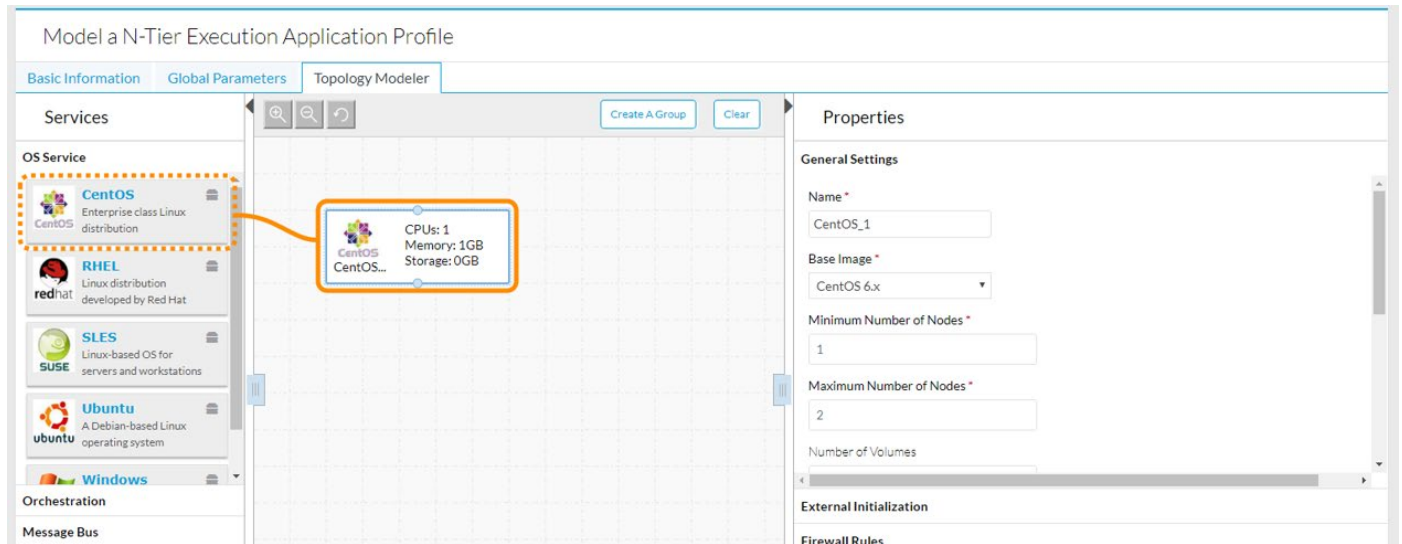
7. Scroll down and locate the **New Logo** section. Click on **Choose File**. Navigate to **Pictures** and select the **splunk-logo.png** file, then click **Open**. Next, click **Add Logo**. The logo should show up.



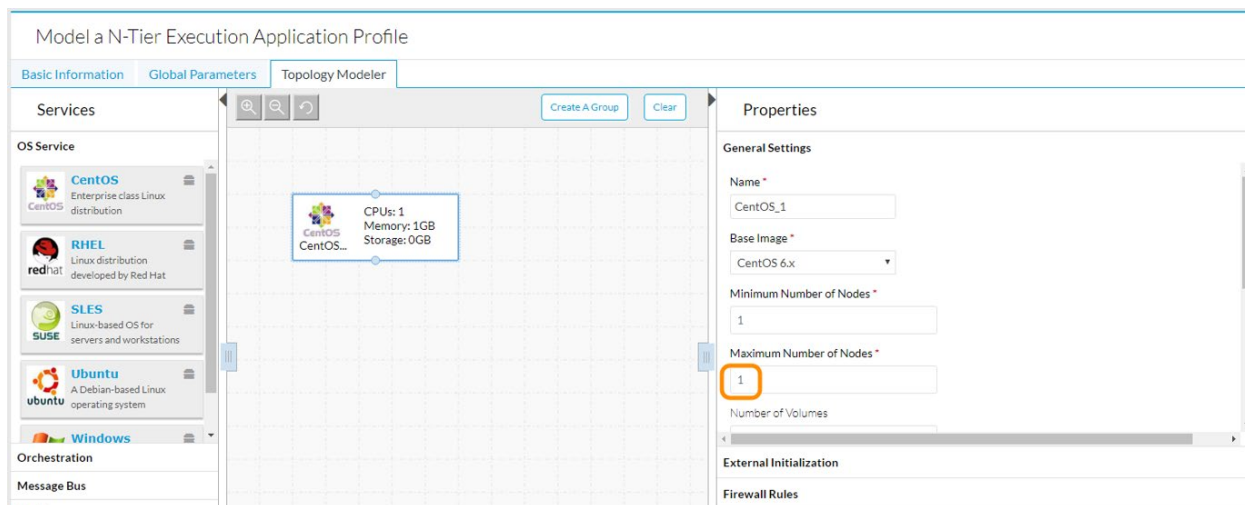
8. Next, navigate to the **Topology Modeler** at the top of the form. At the **Services** section on the left-hand side, expand the **OS Service** section.



- Locate the **CentOS service** and drag and drop it to the modeler.



- Click on the new object and at the Properties section on the right-hand side, under **General Settings**, change the value of Maximum Number of Replicas from 2 to 1.



- At the **Firewall Rules Section**, create the following records:

Protocol	For Port	To Port	IP/CIDR/TIER
TCP	80	80	0.0.0.0/0
TCP	8000	8000	0.0.0.0/0
TCP	8065	8065	0.0.0.0/0
TCP	8089	8089	0.0.0.0/0
TCP	8191	8191	0.0.0.0/0

Model a N-Tier Execution Application Profile

Basic Information Global Parameters Topology Modeler

Services

OS Service

- CentOS Enterprise class Linux distribution
- RHEL Linux distribution developed by Red Hat
- SLES Linux-based OS for servers and workstations
- Ubuntu A Debian-based Linux operating system
- Windows

Orchestration

Message Bus

Web Server

Properties

General Settings

External Initialization

Firewall Rules

You can add firewall rules that your application may need here

Protocol	From Port	To Port	IP/CIDR/TIER	Actions
TCP	1234	5678	0.0.0.0/0	Add
TCP	80	80	0.0.0.0/0	^
TCP	8000	8000	0.0.0.0/0	^
TCP	8065	8065	0.0.0.0/0	^
TCP	8089	8089	0.0.0.0/0	^
TCP	8191	8191	0.0.0.0/0	^

Rules are not saved to the app until you click Save at the bottom of the page.

Deployment Parameters (0)

12. At the **Node Initialization & Clean Up** section, at the Initialization Script, choose dcloud-internal and then type in apps/splunk/splunk_cent.sh to use a pre-configured script that configures Splunk.

Model a N-Tier Execution Application Profile

Basic Information Global Parameters Topology Modeler

Services

OS Service

- CentOS Enterprise class Linux distribution
- RHEL Linux distribution developed by Red Hat
- SLES Linux-based OS for servers and workstations
- Ubuntu A Debian-based Linux operating system
- Windows

Orchestration

Message Bus

Web Server

Custom Service

Workflow

Frontend Cache

Load Balancer

Properties

General Settings

External Initialization

Firewall Rules

Deployment Parameters (0)

Minimum Resource Specifications

Environment Variables

Node Initialization & Clean Up

Are there any node initialization or clean-up actions that need to be taken?

Initialization script

1 dcloud-internal

2 apps/splunk/splunk_cent.sh

---Select a Location---

HTTP Repositories

dcloud-internal

Artifactory Repositories

demo-artifactory

Other Input

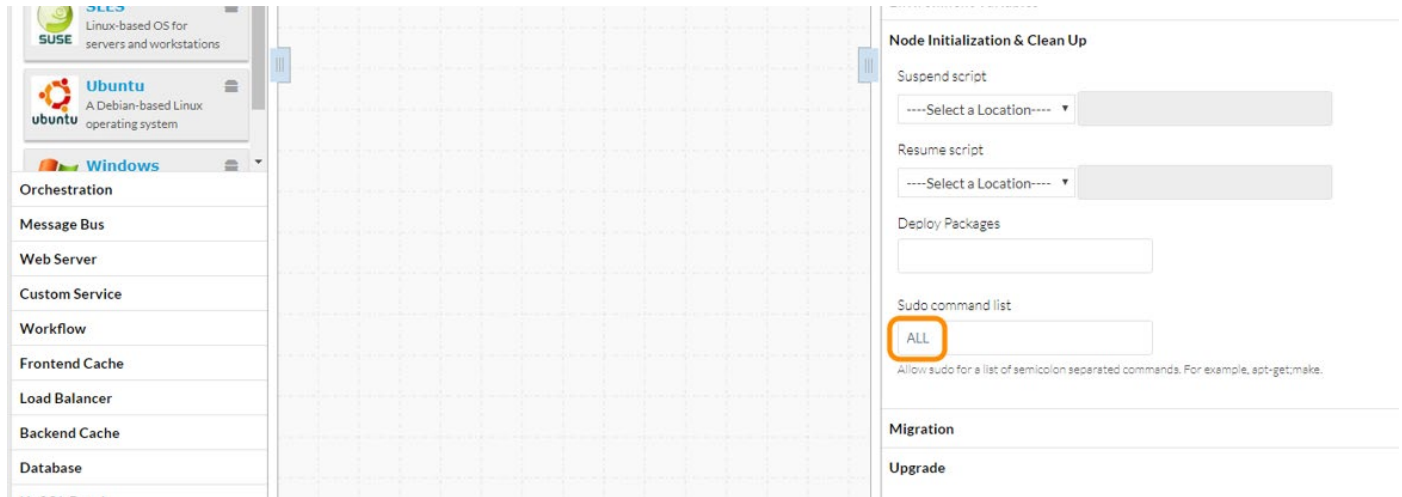
File in Package


URL or Command

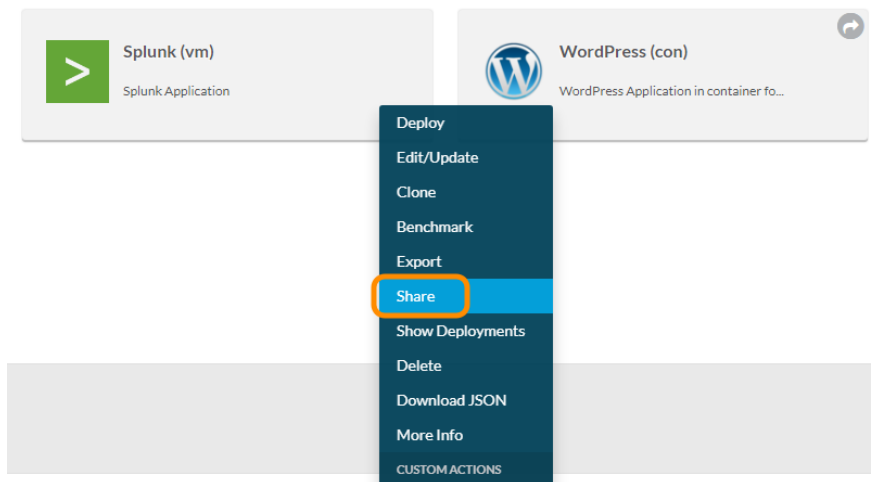
Workflows

Resume script

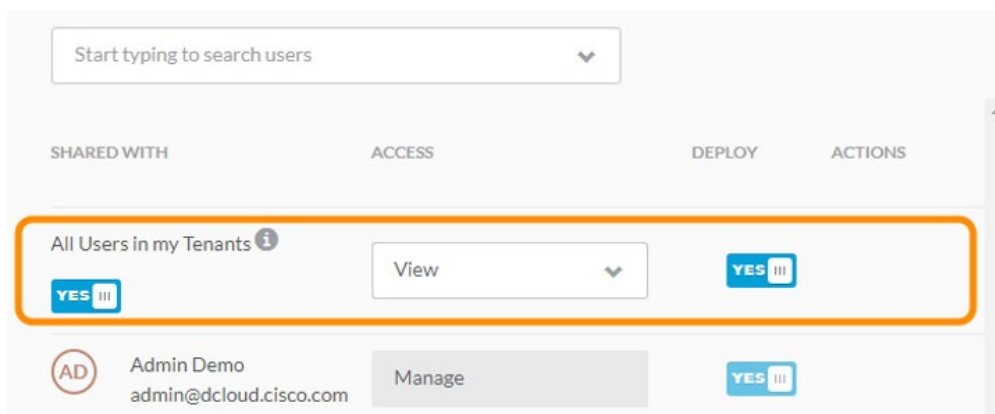
13. In the same section, at the Sudo command list text box, enter the value **ALL** (all capitals). This is to ensure the script has adequate permissions to perform admin-level tasks during the deployment.



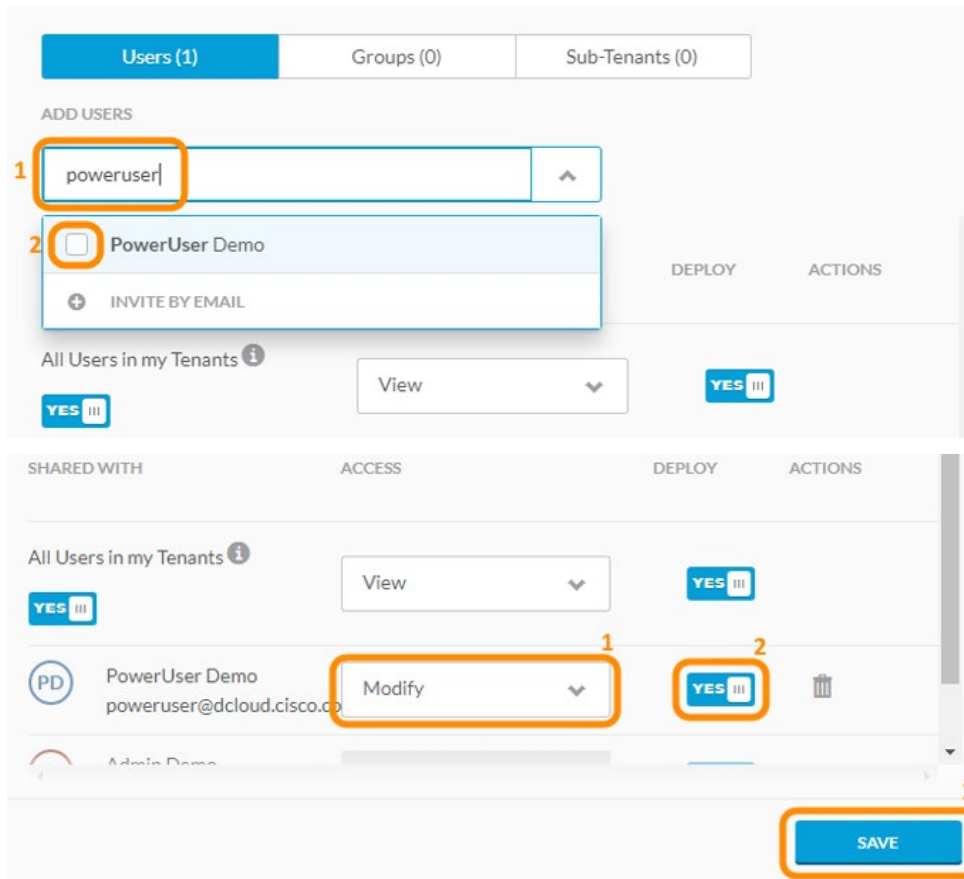
14. Click **Save As App** at the bottom of the form to save the new App Profile.
15. To Share this App Profile so other users can use it, click on the  icon and select **Share**.



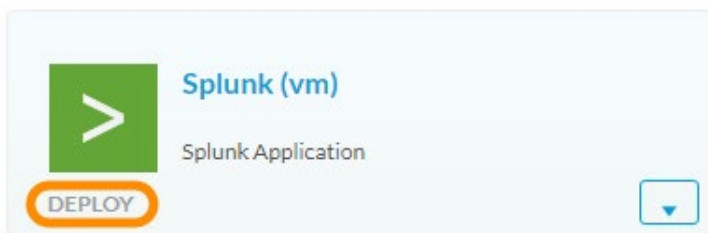
16. At the next form, under **All Users in my Tenants**, set the switch from **No** to **Yes**, leave the **Access to View** and under **Deploy** set the switch from **No** to **Yes**.



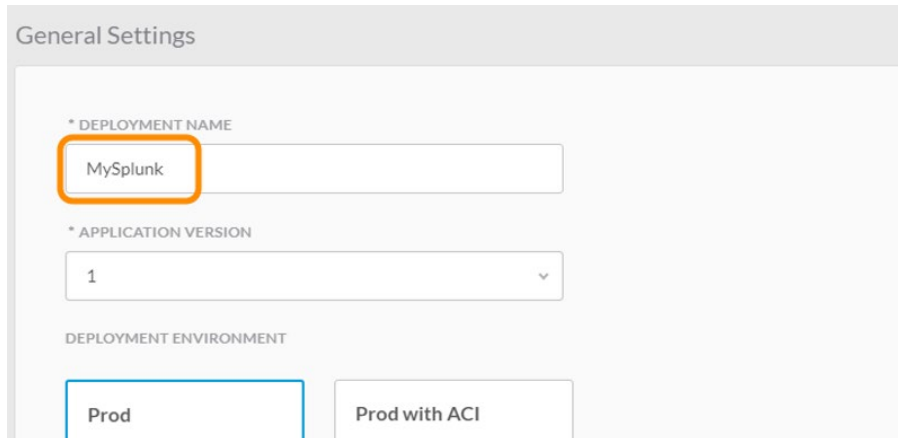
17. Next, at the Add Users text box, type **PowerUser Demo** and select the user from the list. Set the Access to **Modify** and the Deploy to **Yes** for this user, then click **Save**.



18. Once back at the App Profiles form, click on the **Splunk (vm) App Profile** to initiate a deployment.



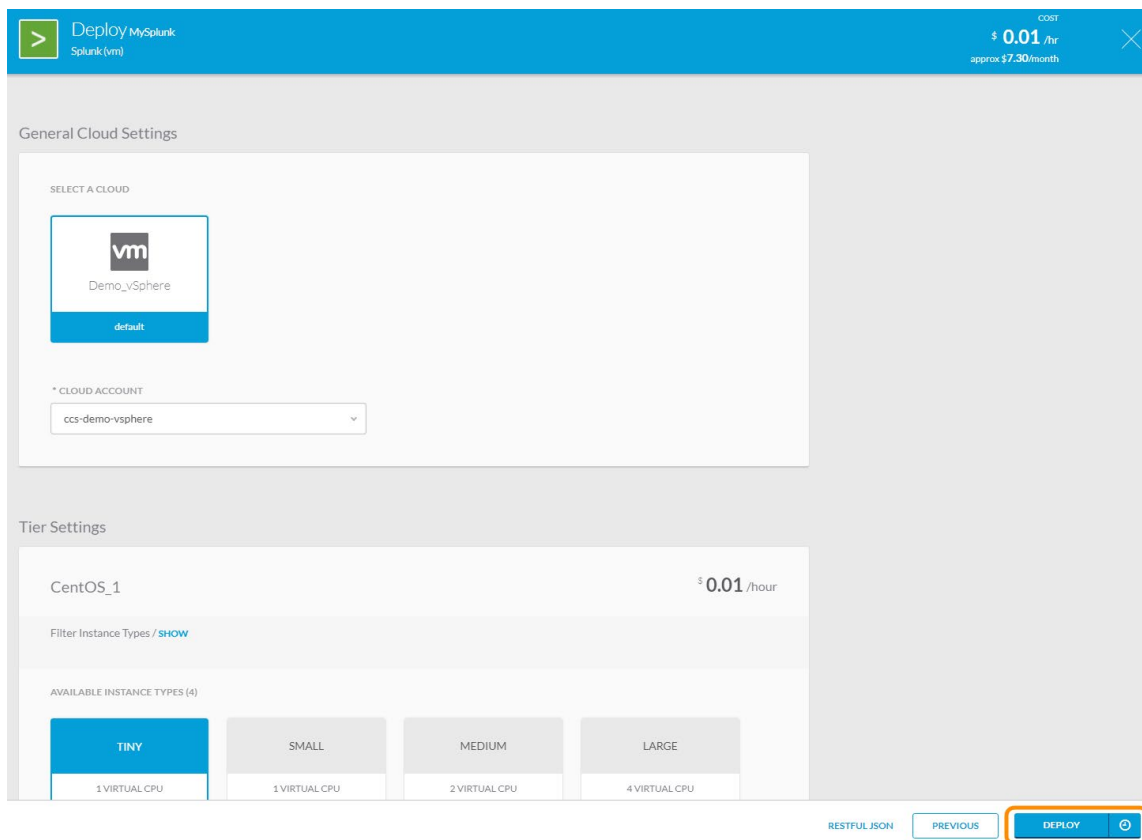
19. Enter **MySplunk** at the **Deployment Name**.



The screenshot shows the 'General Settings' form for deploying Splunk. The 'DEPLOYMENT NAME' field is highlighted with an orange border and contains the text 'MySplunk'. Below it, the 'APPLICATION VERSION' dropdown is set to '1'. At the bottom, the 'DEPLOYMENT ENVIRONMENT' section shows two buttons: 'Prod' (which is highlighted with a blue border) and 'Prod with ACI'.

20. Leave the rest of the fields to their default values and click **Next** at the bottom of the form.

21. At the next form, leave the defaults and click **Deploy**.



The screenshot shows the 'Deploy MySplunk' form. At the top, a blue header bar displays 'Deploy MySplunk', 'Splunk (vm)', and a cost of '\$ 0.01 /hr' (approx \$7.30/month). The 'General Cloud Settings' section includes a 'SELECT A CLOUD' dropdown set to 'vm Demo_vSphere' (labeled 'default') and a 'CLOUD ACCOUNT' dropdown set to 'ccs-demo-vsphere'. The 'Tier Settings' section shows 'CentOS_1' with a cost of '\$ 0.01 /hour'. Below this, there's a 'Filter Instance Types / SHOW' link and a section for 'AVAILABLE INSTANCE TYPES (4)'. Four buttons are shown: 'TINY' (1 VIRTUAL CPU, highlighted with a blue border), 'SMALL' (1 VIRTUAL CPU), 'MEDIUM' (2 VIRTUAL CPU), and 'LARGE' (4 VIRTUAL CPU). At the bottom right, there are three buttons: 'RESTFUL JSON', 'PREVIOUS', and 'DEPLOY' (highlighted with an orange border and a power icon).

22. Once the status of the deployment changes to Running, click on the **Access Application** to launch a browser connection to the new application.

NOTE: Should the user wish to further experiment with the specific application, the admin credentials for this application are **admin / changeme**.

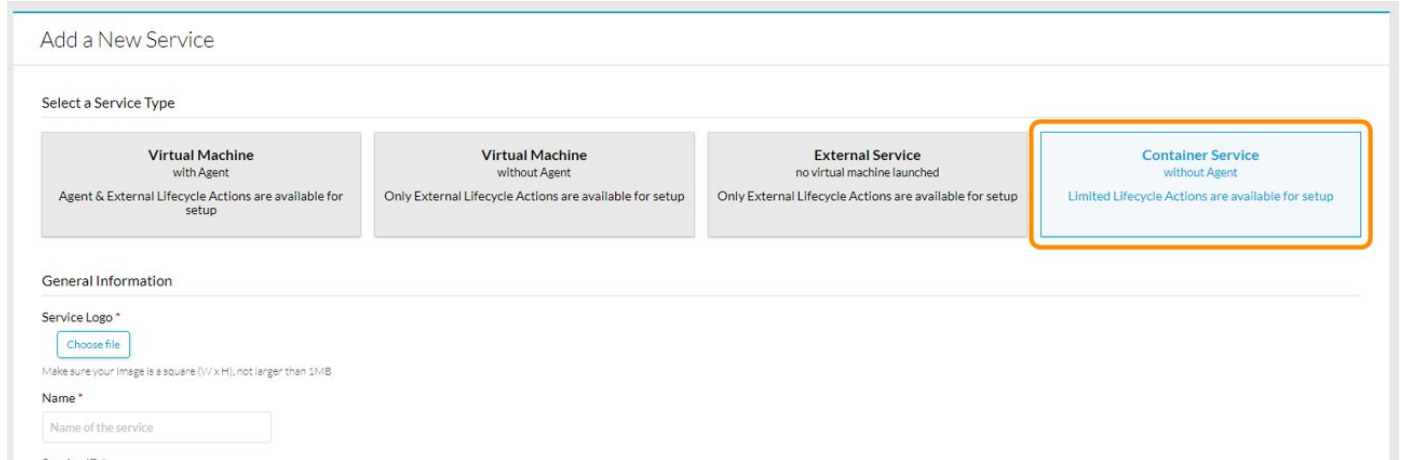
Creating a Container-Based Application Profile

1. Navigate to the CloudCenter UI, login as **admin@dcloud.cisco.com / C1sco12345**, tenant: **demo**.
2. Select **Workload Manager**.

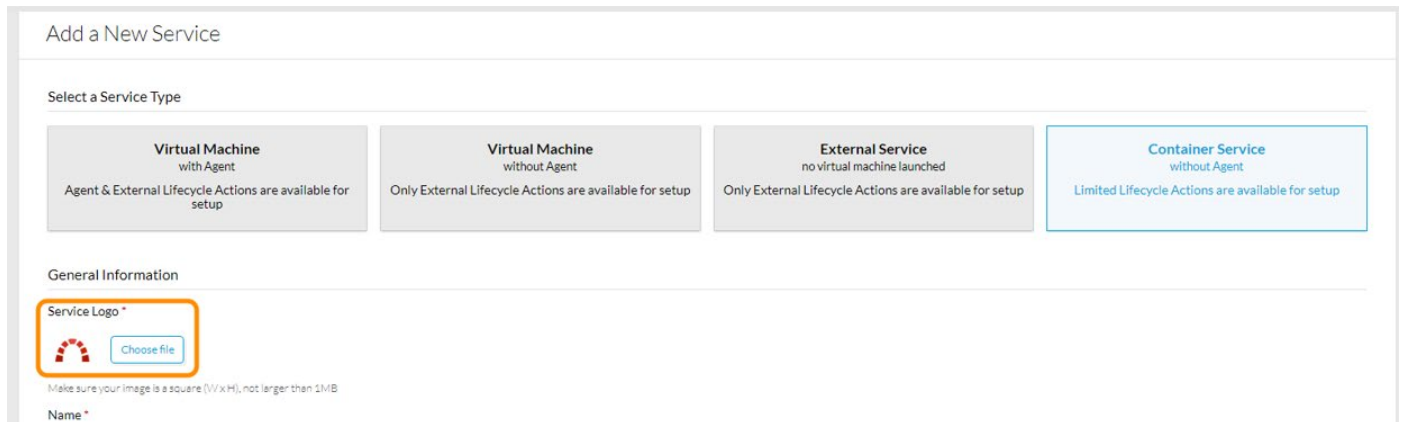


3. Click on **Admin**.
4. Click on **Services**, then click on **Add Service**.

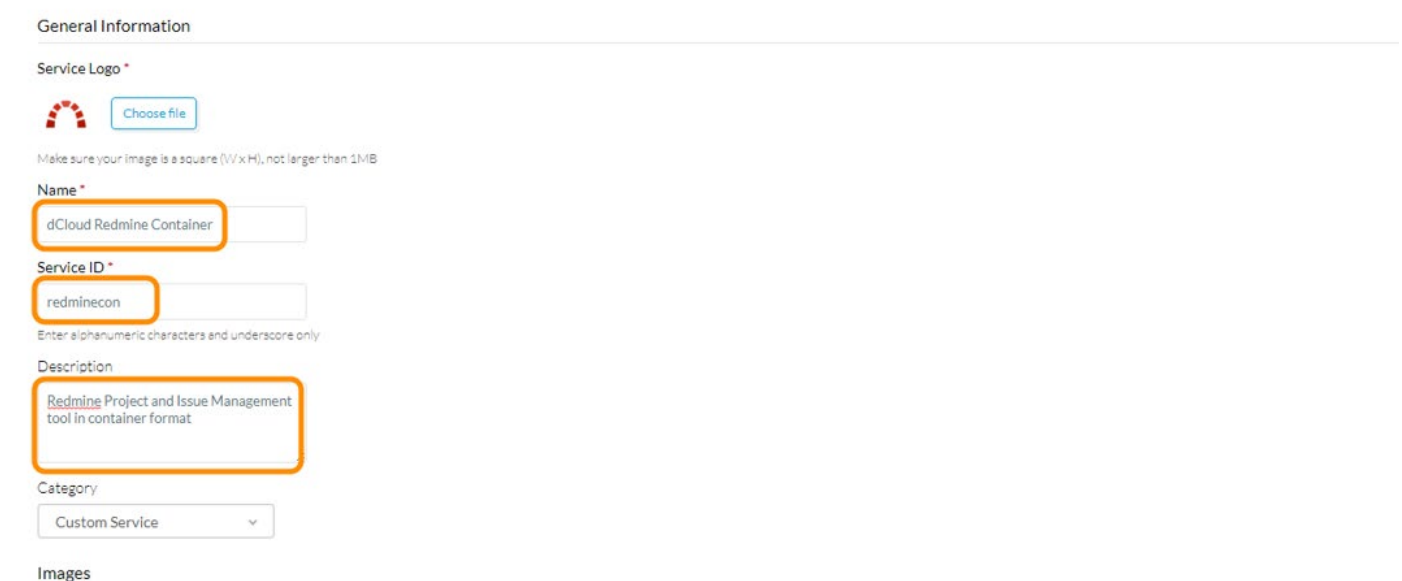
5. Select **Container Service** at the Service Type



6. Click **Choose File** at the Service Logo. Then, navigate to **Pictures** and locate the file **redmine-logo.png**. Select the file and click **Open**. Ensure the application logo shows up.



7. Enter **dCloud Redmine Container** at the Name, then enter **redminecon** at the Service ID. Next, enter **Redmine Project and Issue Management tool in container format** at the Description.



8. Enter **docker.io/redmine:latest** at the Image and click **Add**. The image should show up on the list below.

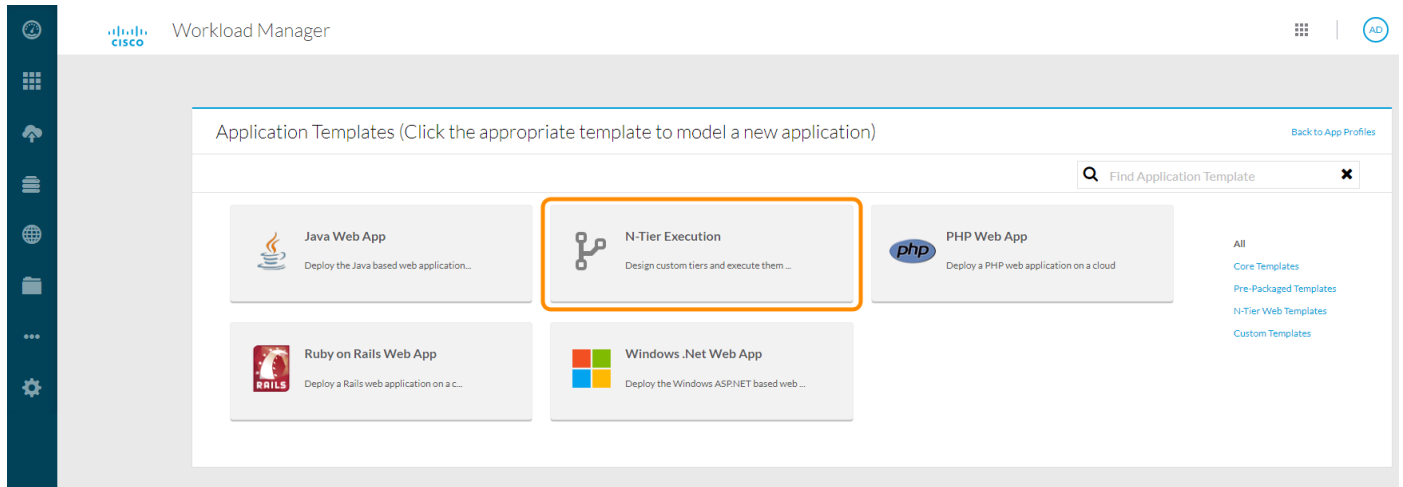
9. At the Container Ports, enter **TCP** for Protocol, **redmine** for Name and **3000** for Port, then click **Add**. The mapping should show up at a list below the fields.

10. Leave the rest of the fields at their default values and click **Save**.

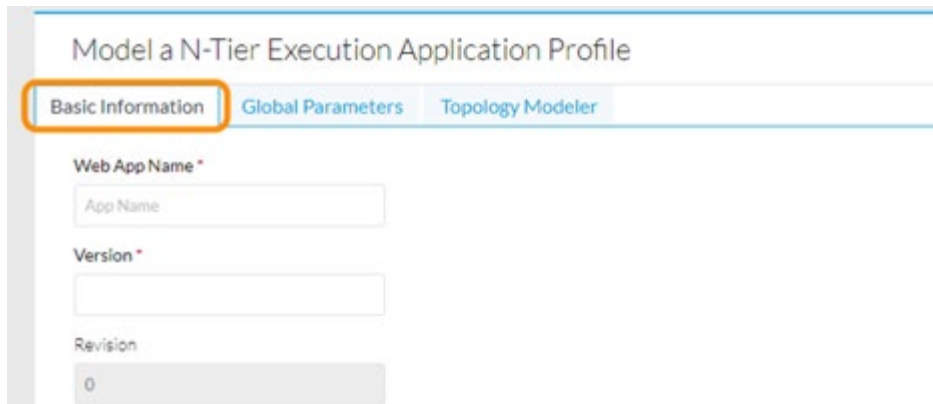
11. At the left-hand side, click on **MAIN MENU** to return to the main choices.

12. Click on **App Profiles**, then click on **MODEL**.

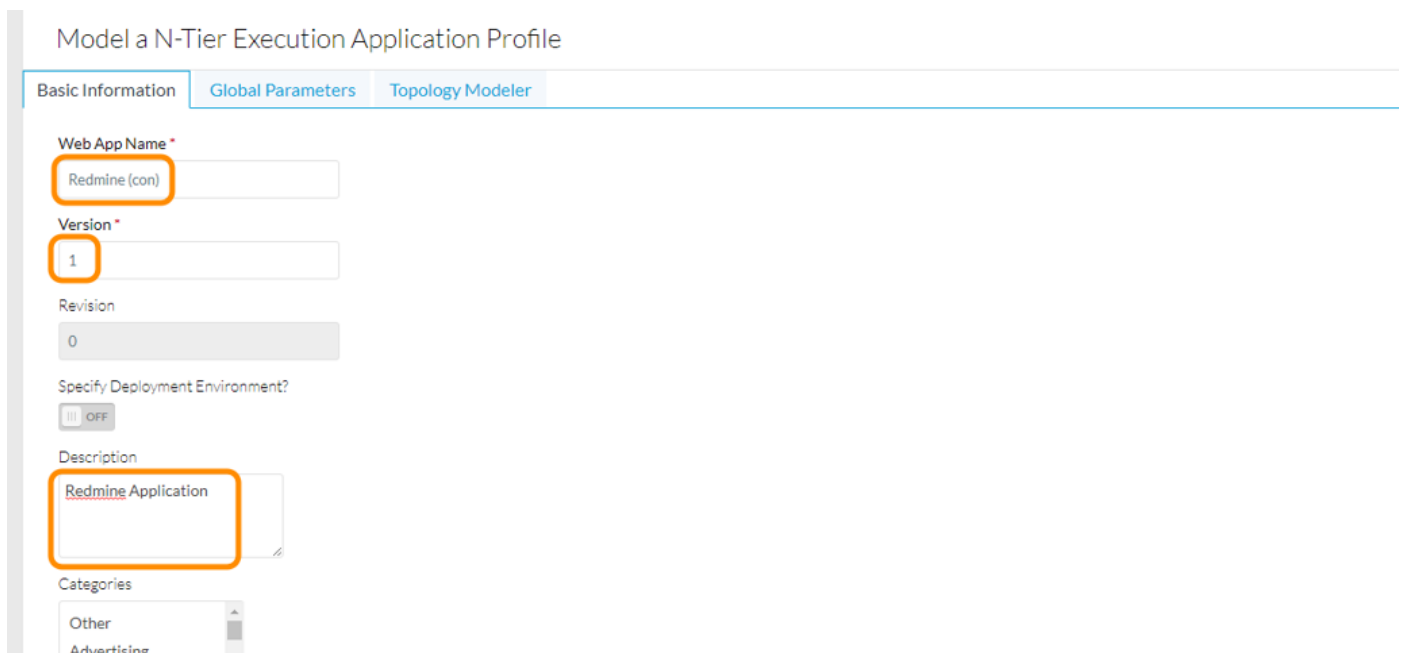
13. At the Application Templates, click on **N-Tier Execution**.



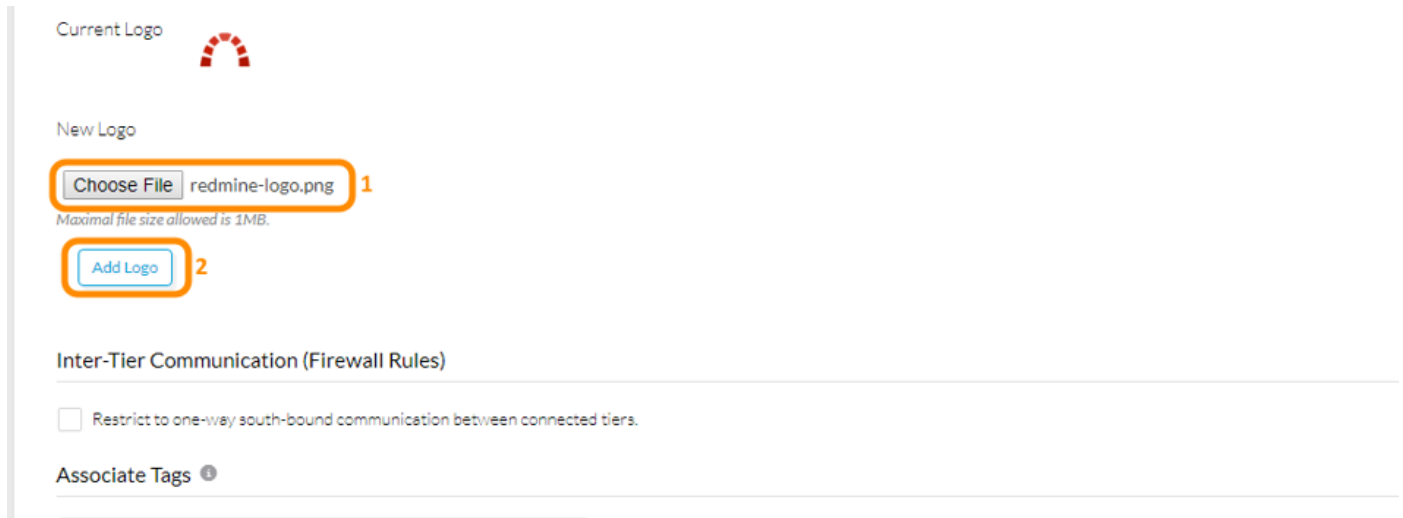
14. Click on the **Basic Information** tab.



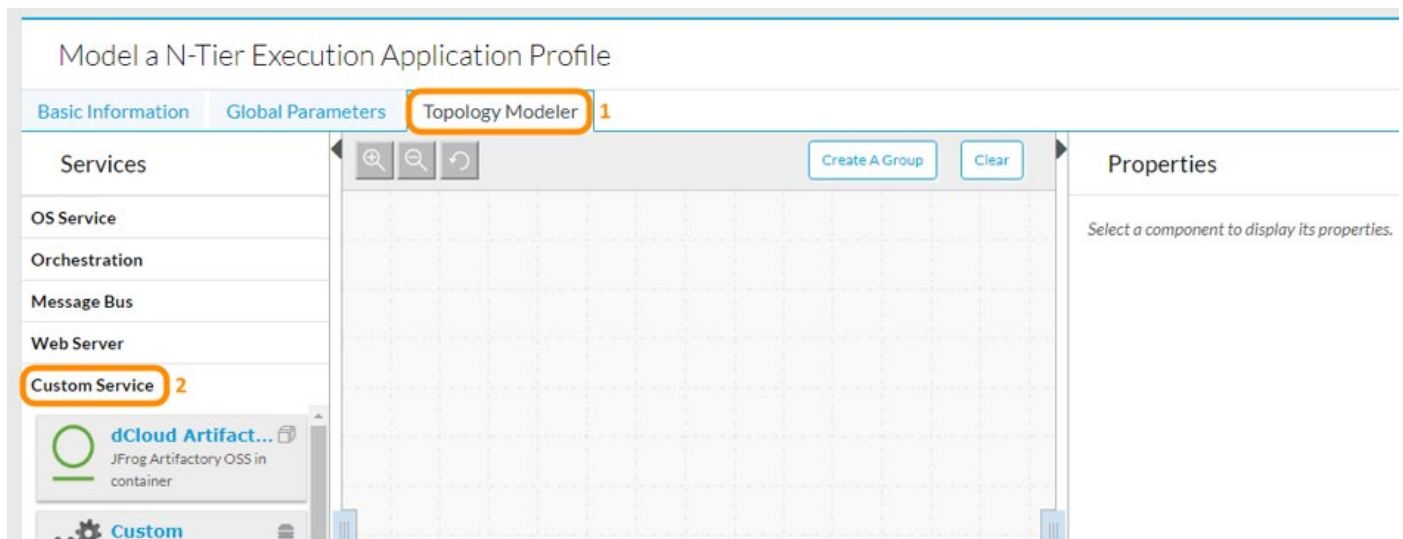
15. Enter **Redmine (con)** at the **Web App Name** field, **1** at the Version field and **Redmine Application** at the Description.



16. Scroll down and locate the New Logo section. Click on **Choose File**. Navigate to Pictures and select the **redmine-logo.png** file, then click **Open**. Next, click **Add Logo**. The logo should show up.



17. Next, navigate to the **Topology Modeler** at the top of the form. At the Services section on the left-hand side, expand the **Custom Service** section.



18. Locate the **dCloud Redmine Container** service and drag and drop it to the modeler.

19. Click on the new object and at the Properties section on the right-hand side, under **General Settings**, change the value of Maximum Number of Replicas from **2** to **1**.

20. Select the **Network Services** section and remove all the records starting with Cluster IP by clicking on the icon at the right-hand side. Then, create a new record of type Load Balancer, enter **80** as Service Port, **3000/TCP** as Container Port/Protocol and **redmine** as Port Name, then click **Add**. Verify that a new record has been added to the list.

Service Port	Container Port / Protocol	Port Name	Actions
Cluster IP		3000 / TCP	
Cluster IP	3000	3000 / TCP	redmine

Orchestration

Message Bus

Web Server

Custom Service

container

placementgrou...
Placement Group
Placeholder Service

dCloud Postgre...
PostgreSQL Server
running in container

dCloud Redmin...
Redmine Project and
Issue Management tool in

dCloud WordPr...
WordPress Application
running in a container

redmine...
Milli CPUs: 100
Memory: 250MB

Volumes

Deployment Parameters (0)

Network Services

1	2	3	4	5
Load Balancer	Service Port	Container Port / Protocol	Port Name	Actions
Load Balancer	80	3000 / TCP	redmine	Add

Network Services are not saved to the app until you click save at the bottom of the page.

Firewall Rules

Minimum Resource Specifications

21. At the Firewall Rules section, ensure a record with the values **3000, TCP, 0.0.0.0/0** exists.

Web Server

Custom Service

container

placementgrou...
Placement Group
Placeholder Service

dCloud Postgre...
PostgreSQL Server
running in container

dCloud Redmin...
Redmine Project and
Issue Management tool in

dCloud WordPr...
WordPress Application
running in a container

redmine...
Memory: 250MB

Network Services

Firewall Rules

You can add firewall rules that your application may need here

Container Port / Protocol	IP/CIDR/TIER	Actions
TCP / 3000	0.0.0.0/0	Add
3000	TCP	0.0.0.0/0

Rules are not saved to the app until you click Save at the bottom of the page.

Minimum Resource Specifications

22. Then, at the Minimum Resource Specifications section, change the value at **MilliCPUs** from **100** to **250** and the Memory from **250** to **1024 MB**.

Message Bus

Web Server

Custom Service

container

placementgrou...
Placement Group
Placeholder Service

dCloud Postgre...
PostgreSQL Server
running in container

dCloud Redmin...
Redmine Project and
Issue Management tool in

dCloud WordPr...
WordPress Application
running in a container

redmine...
Milli CPUs: 250
Memory: 1024MB

Volumes

Deployment Parameters (0)

Network Services

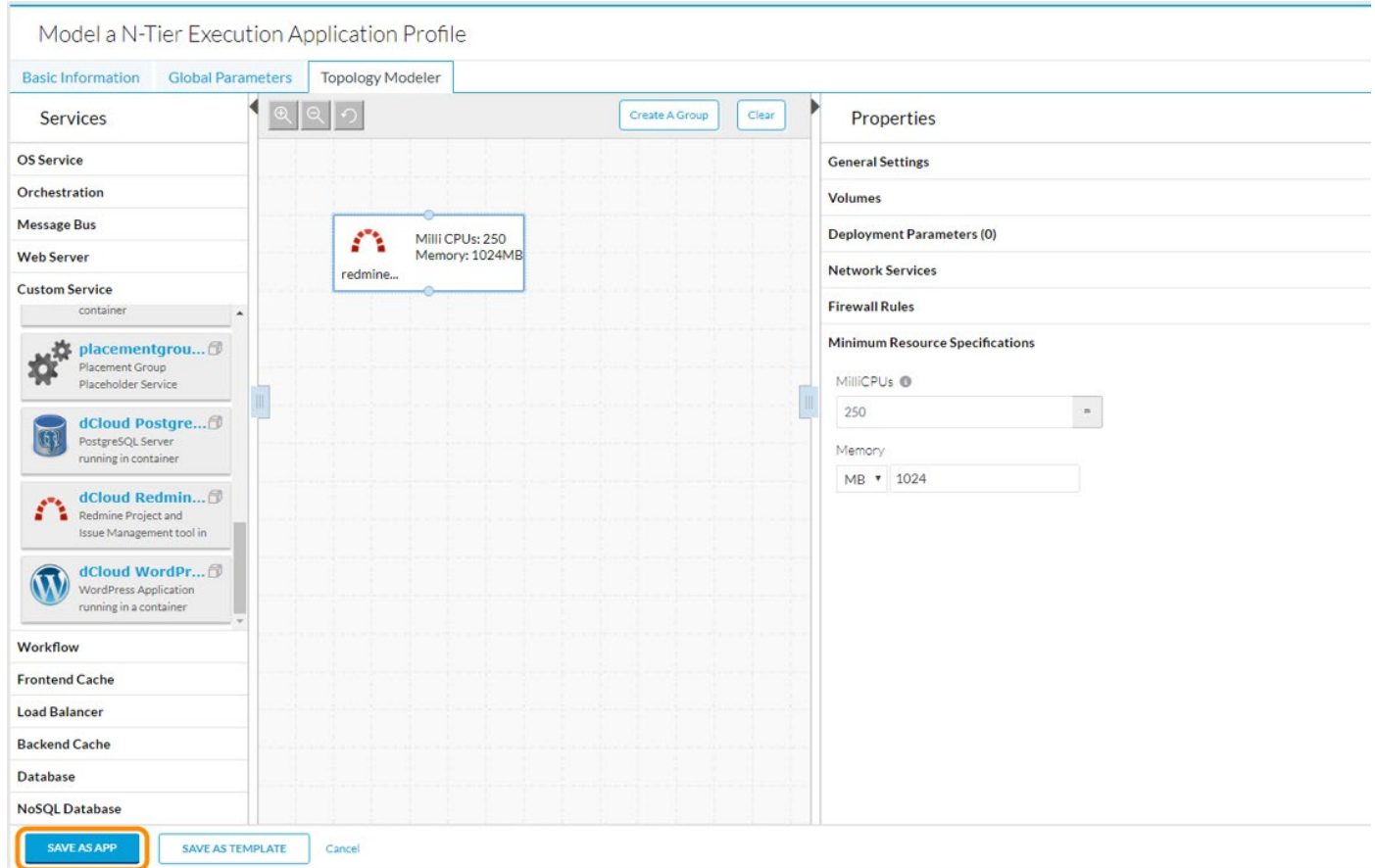
Firewall Rules

Minimum Resource Specifications

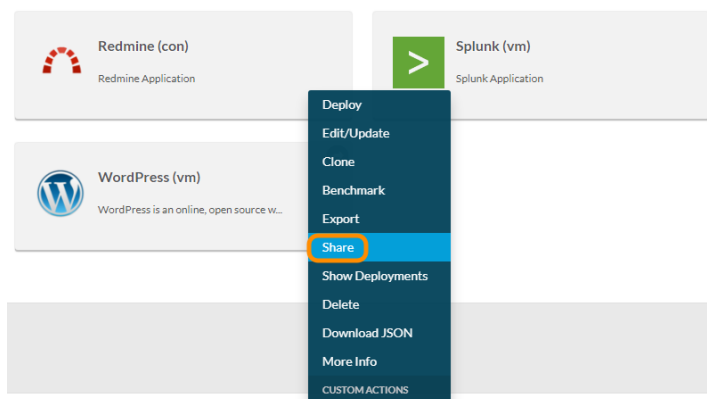
MilliCPUs
250

Memory
MB 1024

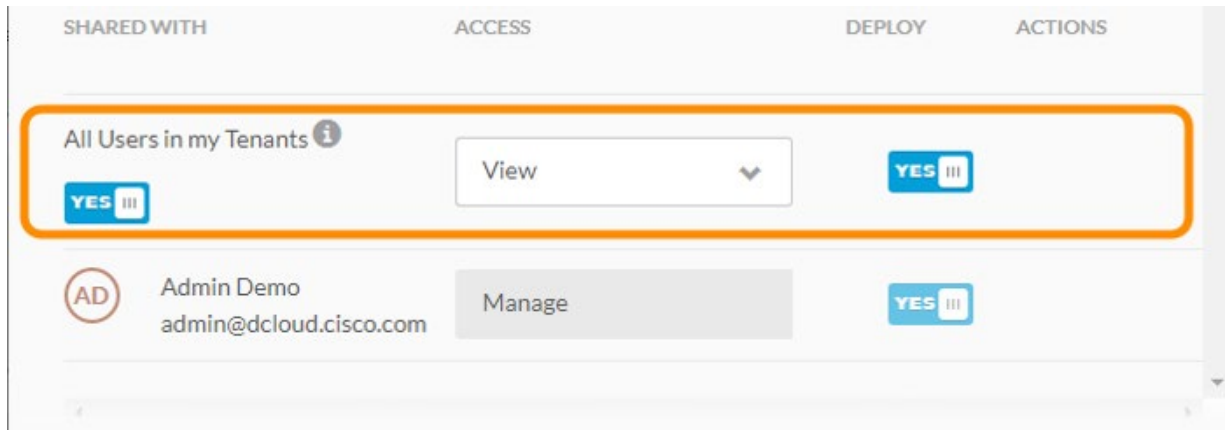
23. Click **Save As App** at the bottom of the form to save the new App Profile.



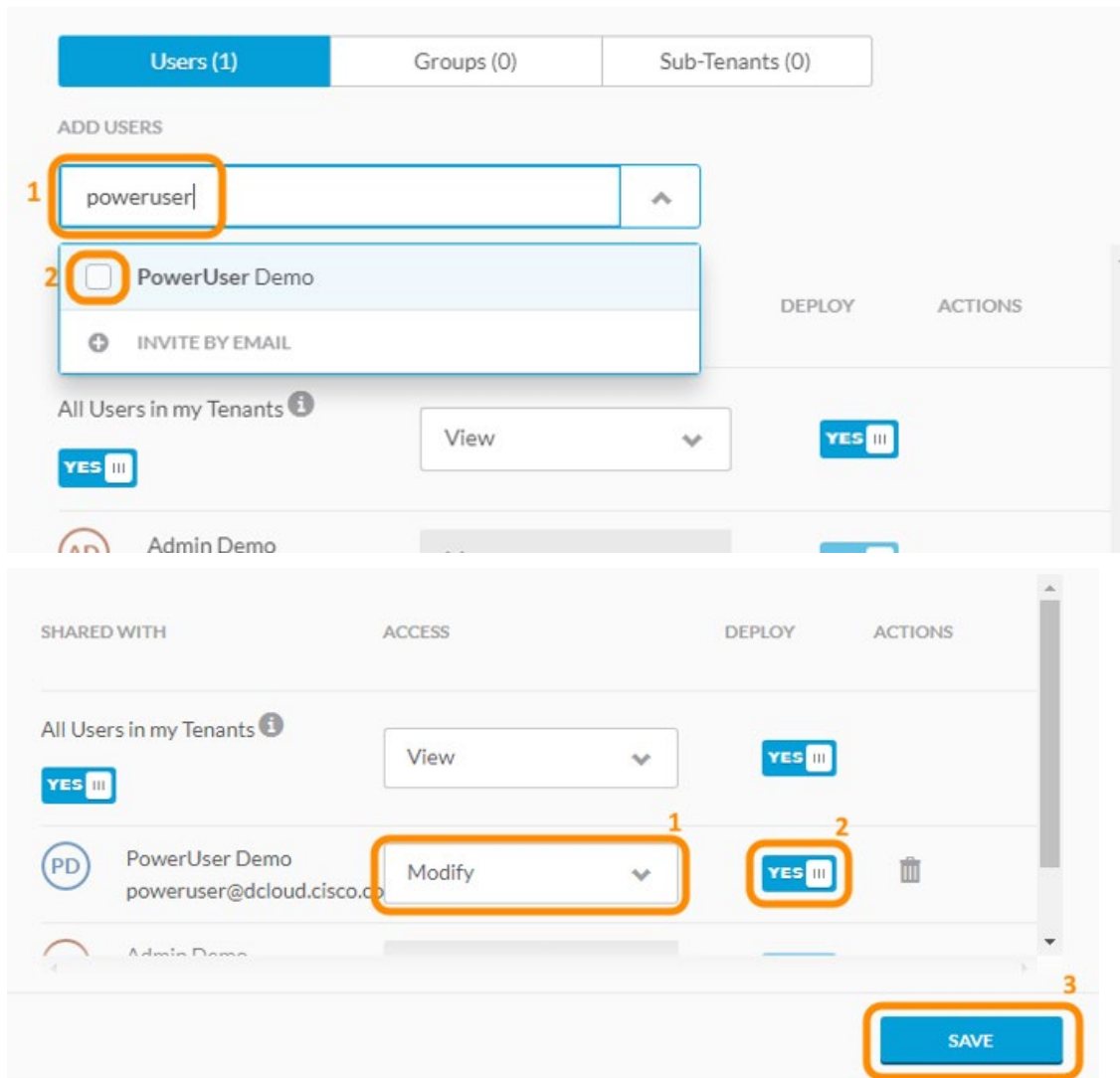
24. To Share this App Profile so other users can use it, click on **Share**.



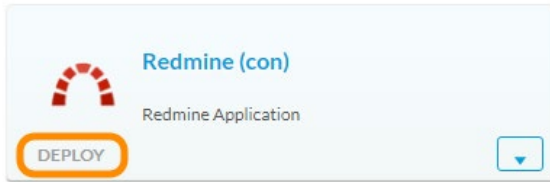
25. At the next form, under All Users in my Tenants, set the switch from **No** to **Yes**, leave the Access to View and under Deploy set the switch from **No** to **Yes**.



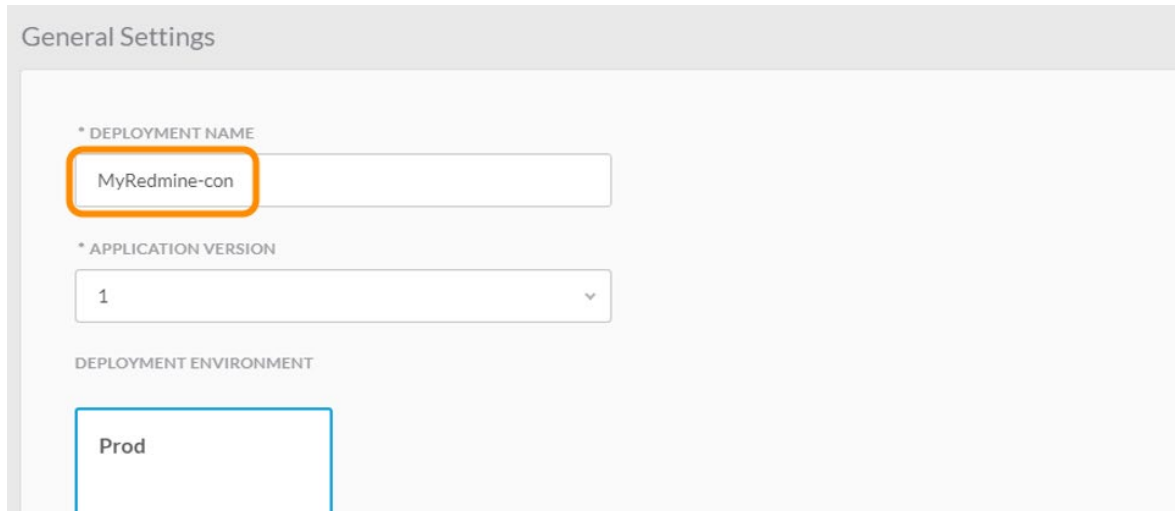
26. Next, at the Add Users text box, type **PowerUser Demo** and select the user from the list. Set the Access to **Modify** and the Deploy to **Yes** for this user, then click **Save**.



27. Once back at the App Profiles form, click on the **Redmine (con) App Profile** to initiate a deployment.



28. Enter **MyRedmine-con** at the Deployment Name.



General Settings

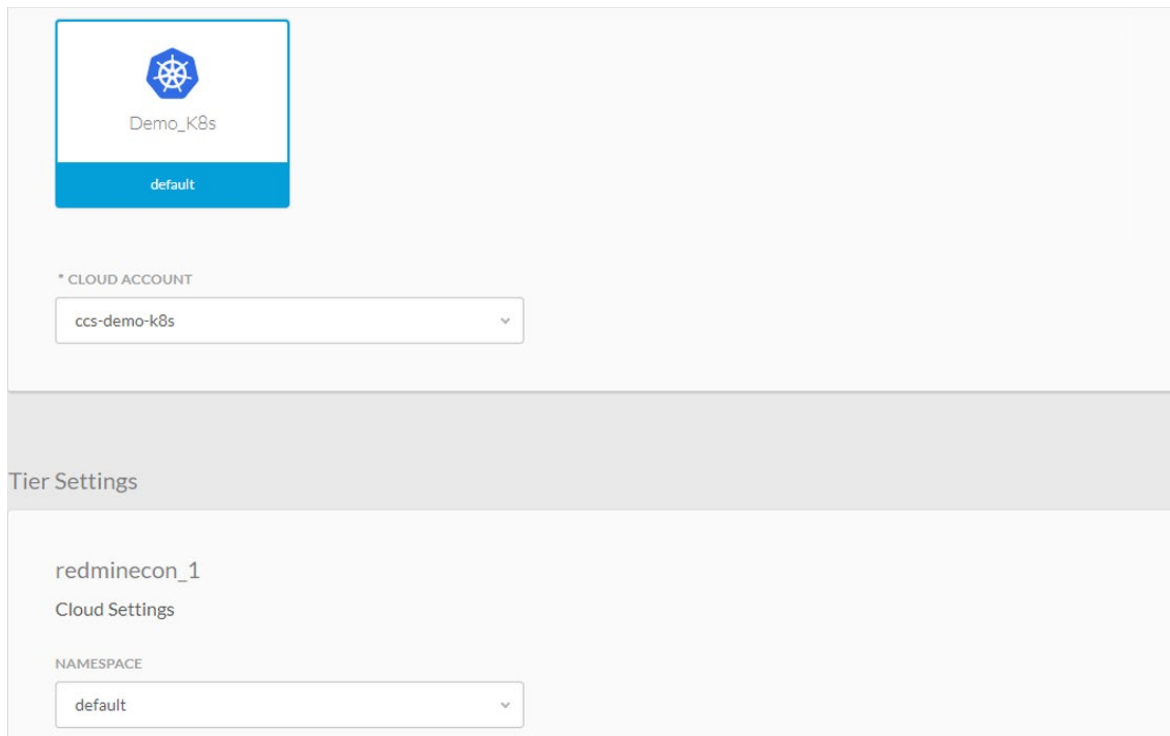
* DEPLOYMENT NAME
MyRedmine-con

* APPLICATION VERSION
1

DEPLOYMENT ENVIRONMENT
Prod

29. Leave the rest of the fields to their default values and click **Next** at the bottom of the form.

30. At the next form, leave the defaults and click **Deploy**.



Demo_K8s
default

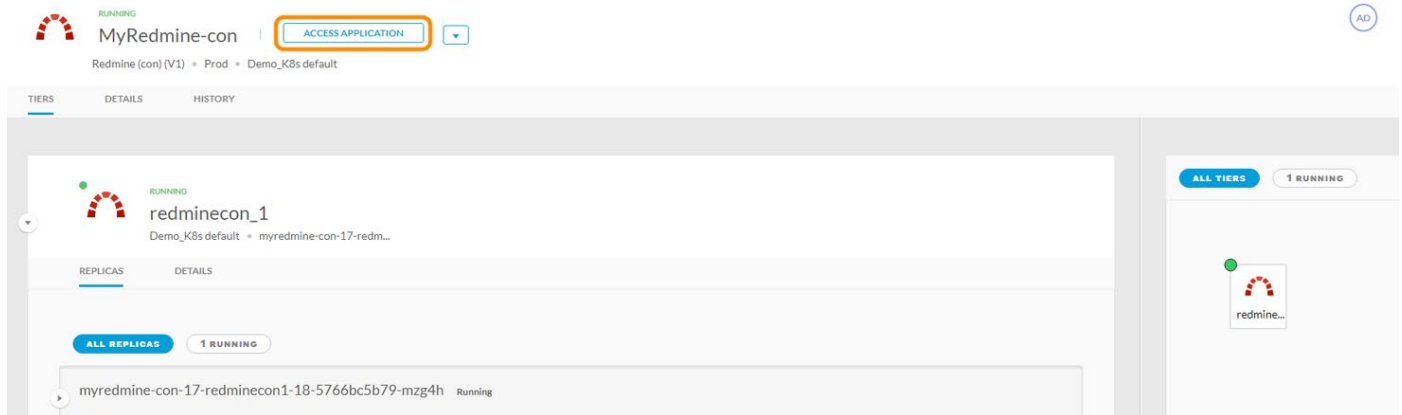
* CLOUD ACCOUNT
ccs-demo-k8s

Tier Settings

redminecon_1
Cloud Settings

NAMESPACE
default

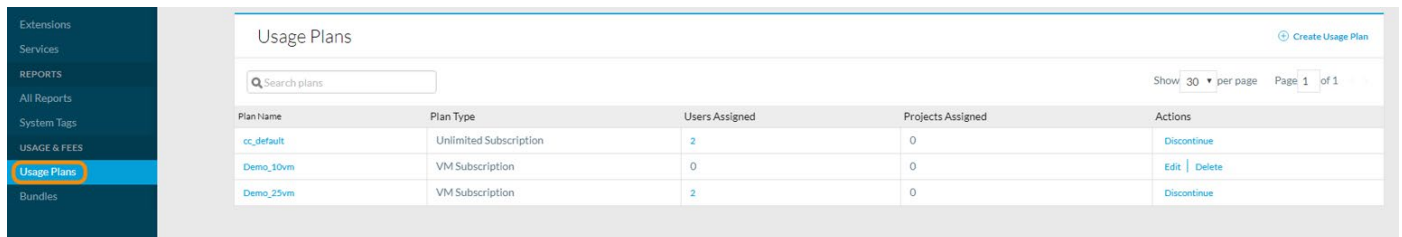
- Once the status of the deployment changes to **Running**, click on the **Access Application** to launch a browser connection to the new application.



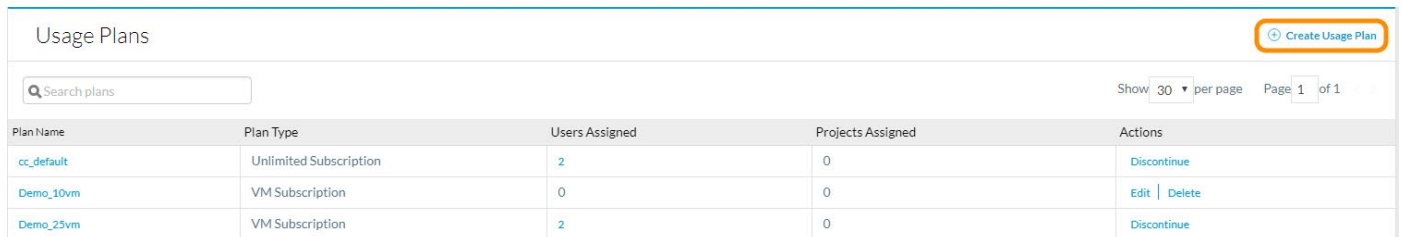
NOTE: Should the user wish to further experiment with the specific application, the admin credentials for this application are **admin / admin**.

Creating a Usage Plan

- Log in to Cisco CloudCenter as admin (**admin@dcloud.cisco.com / C1sco12345**, tenant id: **demo**).
- Click **Admin** on the side menu.
- At the admin menu, click on **Usage Plans**.



- Click the **Create Usage Plan** link.



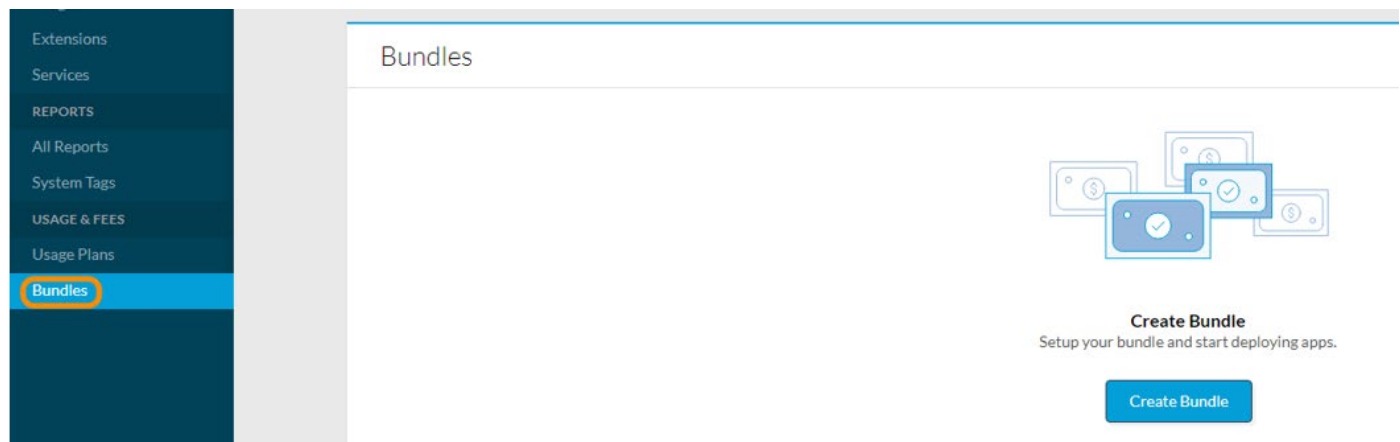
5. Explain the parameters of a usage plan and the possible values, as follows:

Parameter	Description	Values
Plan Name and Description		
Plan Type	Determines the type of bundle on which this usage plan will be based – users of the system will have a plan associated to their profile, and will be able to create applications according to the terms of their plan.	VM-hour Subscription VM Subscription Prepaid VM-hour Business Prepaid Budget Bundle Unlimited Subscription
Monthly VM Hours	The number of VM-hours that will be allowed under this plan – only required for plans that are hour-based	Determined by company policy
Usage Increment Units	Controls the increment steps. e.g.: if a 10 min increment unit is configured and the usage is 11 mins, the metering will be evaluated as 10 + 10 = 20 mins	Determined by company policy
Only Visible to Tenant Admin	Determines whether this usage plan will only be visible by the Tenant Admin or all admins for this module.	True / False

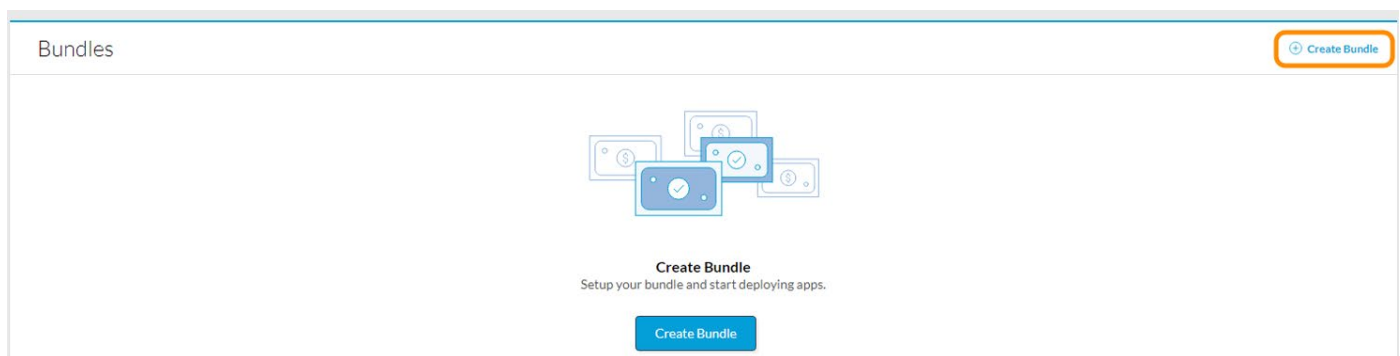
6. Click **Cancel** without creating a usage profile.

Creating a Bundle

1. Click **Bundles** on the side menu.



2. Click the **Create Bundle** link or **Create Bundle** button to show the process for creating a bundle.



3. Explain the parameters of a bundle and the possible values, as follows:

Parameter	Description	Values
Plan Name and Description		
Type	Determines whether a usage plan that uses this bundle will be based on hourly usage or a set number of dollars	Hour Based Budget Based
VM-hour Limit	Number of VM-hours allowed in an Hour Based plan – only required for Hour Based plans	
Budget Limit	The number of dollars per month that users (each user, or the user community as a whole?) can spend on VMs - only required for Budget Based plans	
Expiration Type		
Expiration Date	If the Expiration Type is Fixed Date, the date on which the bundle will expire	
Only Visible to Admin	Determines whether users can see the details of the usage plan	

4. Click **Cancel** without creating a Bundle.

Scenario 4. CI/CD Solution using GitLab, Jenkins, Artifactory and CloudCenter

The purpose of scenario is to demonstrate how developers can move code along the Continuous Integration / Continuous Delivery pipeline using CloudCenter, Jenkins, Artifactory and GitLab. This workflow is often referred to by those pursuing DevOps methodologies. The integration simplifies the movement of code along the tools in the tool chain and exhibits how a critical system variable – such as the BUILD_NUMBER – can be exchanged between the different platforms.

The first part of the scenario is a walk-through of the environment, showing how the application is configured in CloudCenter, Artifactory, and Jenkins.

The second part of the scenario is to clone a Git branch, check out the code, modify one of the files and observe Jenkins perform two builds:

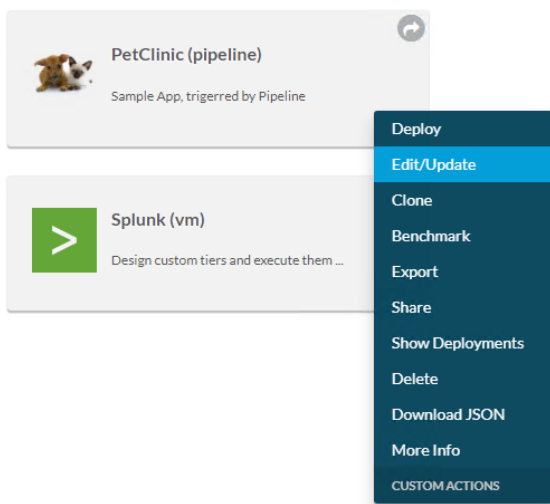
- The PetClinic.Build build takes the newest code to produce artifacts and store them into Artifactory
- The PetClinic.Deploy build deploys the application profile with the newly stored artifacts, leveraging the plugin for CloudCenter

Steps

Environment Walkthrough

The purpose of this section is to become familiar with the CloudCenter, Artifactory, and Jenkins environments of the PetClinic application.

1. Log in to Cisco CloudCenter (admin@dcloud.cisco.com / C1sco12345 / demo).
2. Select **Workload Manager**.
3. Click **App Profiles** in the side menu.
4. Mouseover the **PetClinic (pipeline)** application. Click the down arrow and select **Edit/Update** from the resulting menu to review the application parameters.



5. Click the **Global Parameters** tab.
6. Note that the Parameter Name is **JOB_BUILD_NUMBER**, a variable. The build number is passed from CloudCenter to Artifactory to Jenkins.

Edit "PetClinic" Application Profile

Version: 1.0.1 > **1.0.2** (Revision: 6)

Basic Information **Global Parameters** Topology Modeler

If you would like to add additional parameters, this is the place.

Global Parameters | add a parameter »

Parameter

Parameter Name *

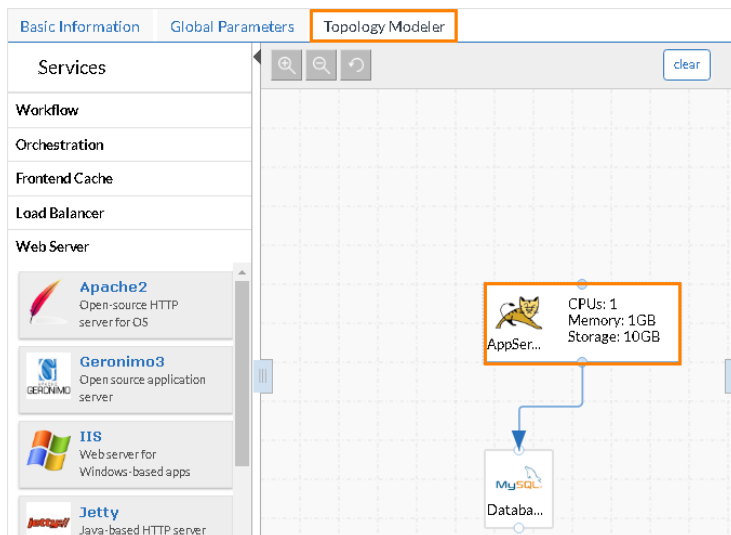
JOB_BUILD_NUMBER

Display Name *

Job Build Number

7. Click the **Topology Modeler** tab.

8. Click the **Apache Server**.



9. Scroll down the **Properties** list to the **App Package**.
10. Note that the application package is an **Artifactory** package.
11. Double-click in the Package File field and scroll over to show that the **JOB_BUILD_NUMBER** is part of the package name.

Properties

General Settings

0 GB

Persistent data storage.

App Run-time *

JDK 7

App Package *

demo-artifactory | **jc-1.0.1-%JOB_BUILD_NUMBER%.war**

Application package file. The file is in relative path from http://artifactory.dcloud.cisco.com:8081/artifactory/.


App Config files

Application config files that contain CI/CD system tokens and will be modified at deployment time. The config file is a Groovy script that will be executed on the target system. It can be used to configure the system at deployment time.

Service Initialization


External Initialization

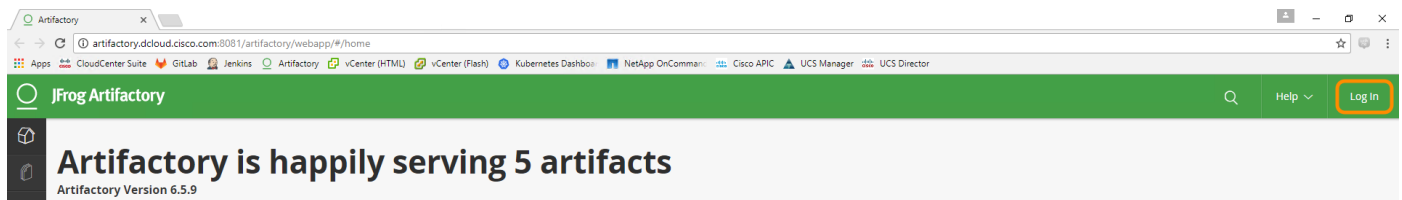
Firewall Rules

12. Click **More > Repositories**  in the side menu and note that **Artifactory (demo-artifactory)** is one of the repositories.
13. If desired, click **demo-artifactory** to examine the settings.

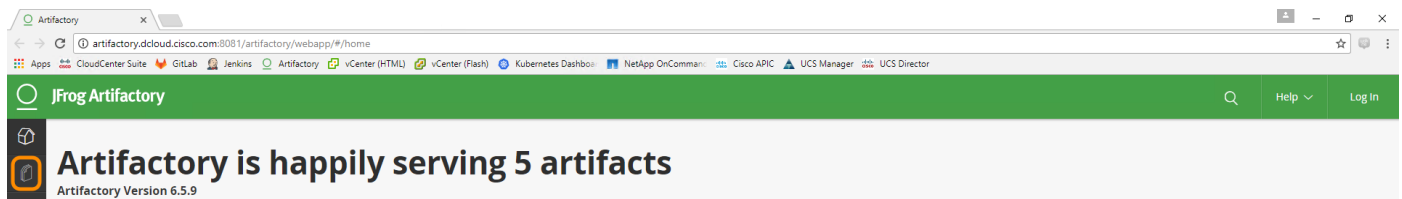


Name	Description	Type	Actions
dcloud-internal		HTTP	 
demo-artifactory		ARTIFACTORY	 

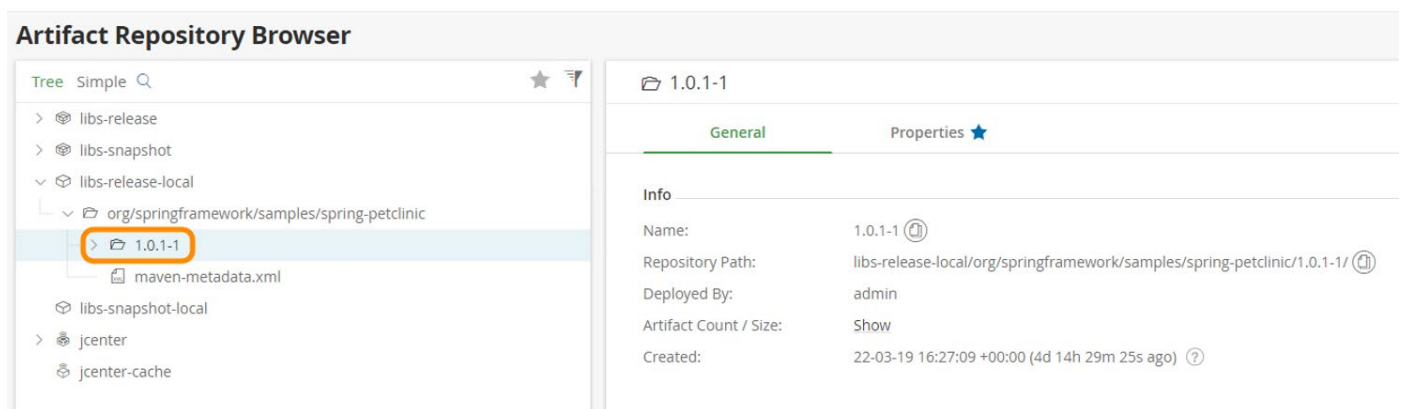
14. Open a Chrome browser and click the **Artifactory**  bookmark.
15. Click **Log In** and log in with the credentials **admin / C1sco12345**.



16. Click the **Artifacts** icon in the side menu.



17. Expand the **libs-release-local repository**, then the **org/springframework/samples/spring-petclinic** directory and click the latest release number, **1.0.1-1**. Show the parameters.



Artifact Repository Browser

Tree Simple

- libs-release
- libs-snapshot
- libs-release-local
 - org/springframework/samples/spring-petclinic
 - 1.0.1-1**
 - maven-metadata.xml
 - libs-snapshot-local
 - jcenter
 - jcenter-cache

1.0.1-1

General Properties

Info

Name: 1.0.1-1

Repository Path: libs-release-local/org/springframework/samples/spring-petclinic/1.0.1-1/

Deployed By: admin

Artifact Count / Size: Show

Created: 22-03-19 16:27:09 +00:00 (4d 14h 29m 25s ago)

18. Open a new browser tab and click the GitLab bookmark  to open GitLab.

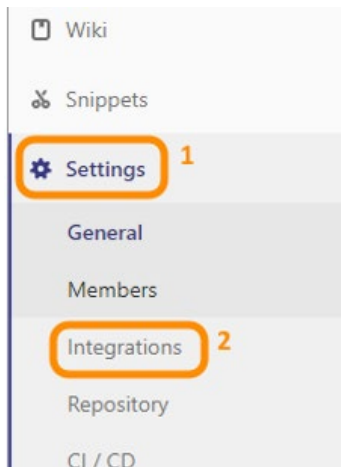
19. Sign in with the default login credentials (**demouser@dcloud.cisco.com** / **C1sco12345**).

20. Click **dCloud Demo User / PetClinic**.



21. Click **Repository** in the top menu to view the repository files.

22. Click the **Settings** icon and then select **Integrations** from the resulting menu.



23. Scroll down to the middle of the form to locate the existing integration and click **Edit**.

- ☐ **Job events**
This URL will be triggered when the job status changes
- ☐ **Pipeline events**
This URL will be triggered when the pipeline status changes
- ☐ **Wiki Page events**
This URL will be triggered when a wiki page is created/updated

SSL verification

- ☒ **Enable SSL verification**

[Add webhook](#)

Webhooks (1)

<http://jenkins.dcloud.cisco.com/project/PetClinic/PetClinic.Build>
Push Events Tag Push Events

SSL Verification: disabled

[Edit](#)

[Test](#) 

24. Observe the configured settings for this webhook, the **Change Notification URL**, the Notification Authentication Token and the **Notification Event Filters**, then click at the **PetClinic** link to exit without saving changes.

Integrations

Webhooks can be used for binding events when something is happening within the project.

URL
http://jenkins.dcloud.cisco.com/project/PetClinic/PetClinic.Build

Secret Token
2d58da9bd44a1357a40e1f2b343f2d06

Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-Token HTTP header.

Trigger

- ☒ **Push events**
This URL will be triggered by a push to the repository
Branch name or wildcard pattern to trigger on (leave blank for all)
- ☒ **Tag push events**
This URL will be triggered when a new tag is pushed to the repository
- ☐ **Comments**
This URL will be triggered when someone adds a comment
- ☐ **Confidential Comments**
This URL will be triggered when someone adds a comment on a confidential issue
- ☒ **Issues events**
This URL will be triggered when an issue is created/updated/merged
- ☐ **Confidential Issues events**
This URL will be triggered when a confidential issue is created/updated/merged
- ☐ **Merge request events**
This URL will be triggered when a merge request is created/updated/merged
- ☐ **Job events**
This URL will be triggered when the job status changes
- ☐ **Pipeline events**
This URL will be triggered when the pipeline status changes
- ☐ **Wiki Page events**
This URL will be triggered when a wiki page is created/updated

SSL verification
☐ Enable SSL verification

[Save changes](#) [Test](#) [Remove](#)

25. Open a new browser tab and click the **Jenkins** bookmark. Log in with the default credentials (**admin/C1sco12345**). The dashboard shows the **PetClinic** folder containing the two builds for the PetClinic application - the artifact build (PetClinic.Build) and the application deployment build (PetClinic.Deploy).

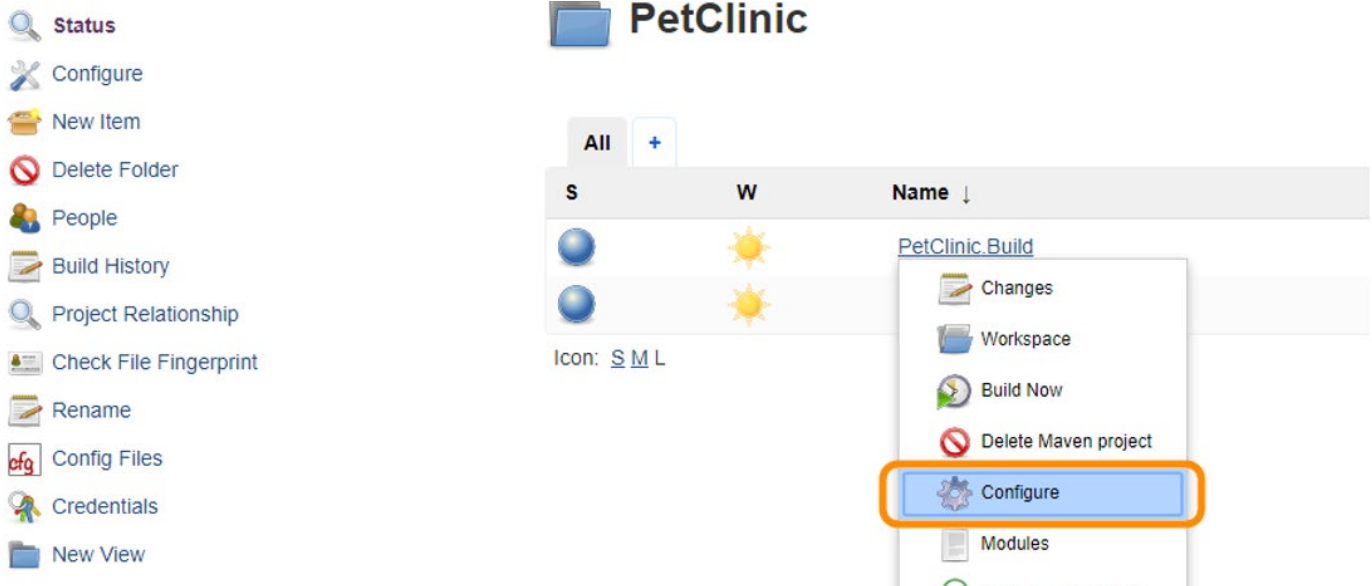
26. Click on the **PetClinic** folder to view the two jobs related to PetClinic.

Jenkins [Dashboard]

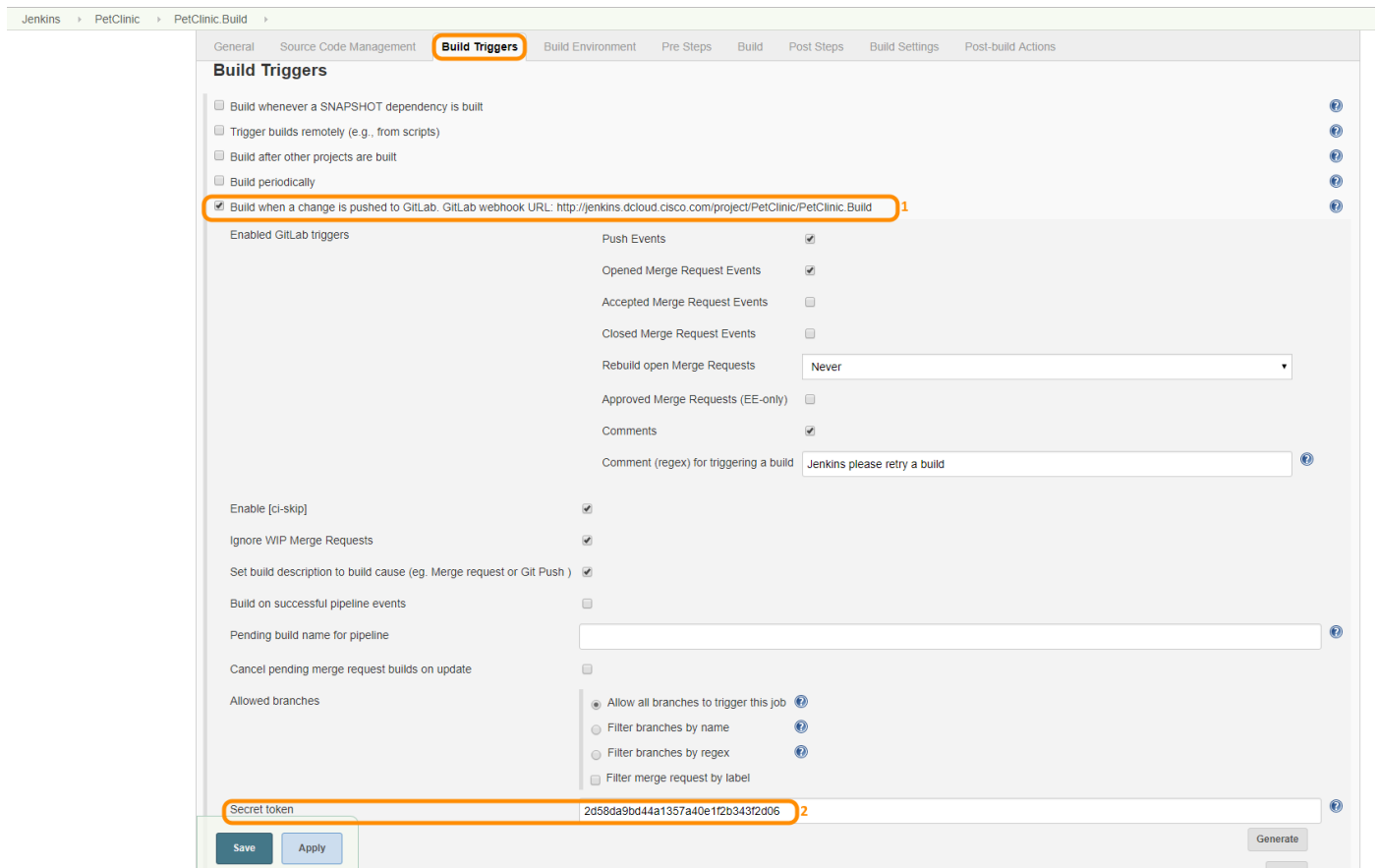
S	W	Name ↓	Last Success	Last Failure
		PetClinic	N/A	N/A

Icon: [S](#) [M](#) [L](#)

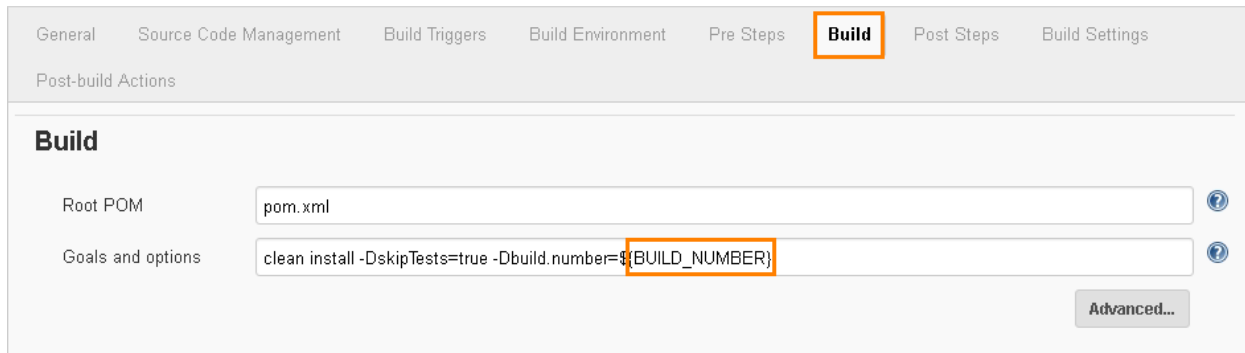
27. Mouseover **PetClinic.Build** to display the down arrow. Click the down arrow and select **Configure** from the menu.



28. Click **Build Triggers** to observe the configuration used for receiving build notifications. Notice the configured setting for enabling GitLab integration and the Secret token used to authenticate incoming requests .



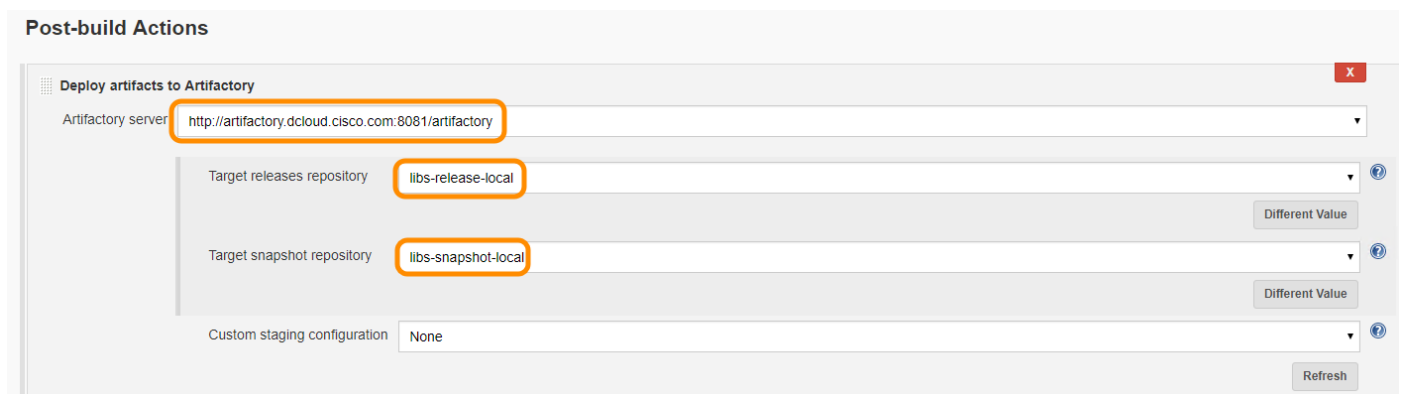
29. Click **Build** and show that the **BUILD_NUMBER** parameter is passed in Jenkins.



The image shows the Jenkins configuration page for a build job. The top navigation bar includes tabs for General, Source Code Management, Build Triggers, Build Environment, Pre Steps, **Build** (highlighted with an orange box), Post Steps, and Build Settings. Below the tabs, the 'Build' section is active. It contains two input fields: 'Root POM' with the value 'pom.xml' and 'Goals and options' with the value 'clean install -DskipTests=true -Dbuild.number=\${BUILD_NUMBER}'. The 'Goals and options' field is highlighted with an orange box. An 'Advanced...' button is located at the bottom right of the Build section.

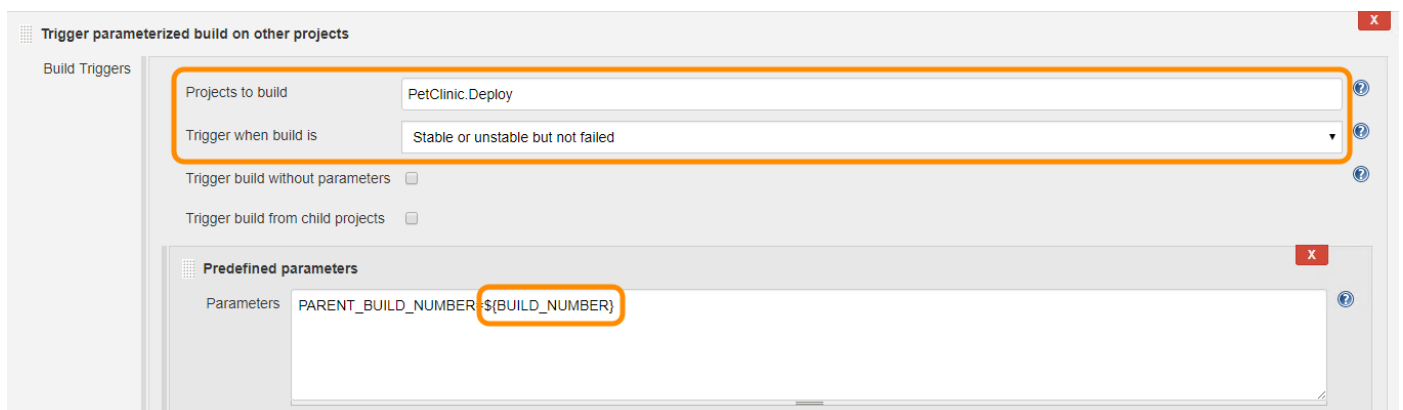
30. If desired, click **Post Steps** in the top menu to show the settings.

31. Click **Build Settings** in the top menu. Scroll down and show the **Deploy artifacts to Artifactory** settings. Note the Artifactory server, the Target releases repository, and the Target snapshot repository. These parameters set the repositories on the Artifactory server that will house the new artifacts and code from the Jenkins builds.



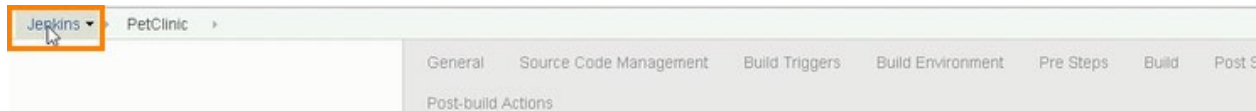
The image shows the 'Post-build Actions' section of the Jenkins configuration page. The 'Deploy artifacts to Artifactory' section is expanded. It contains the following settings: 'Artifactory server' set to 'http://artifactory.dcloud.cisco.com:8081/artifactory' (highlighted with an orange box), 'Target releases repository' set to 'libs-release-local' (highlighted with an orange box), 'Target snapshot repository' set to 'libs-snapshot-local' (highlighted with an orange box), and 'Custom staging configuration' set to 'None'. There are 'Different Value' buttons next to the repository fields and a 'Refresh' button at the bottom right.

32. Scroll down to the **Trigger** parameterized build on other projects section and show that the Deploy_PetClinic project triggers when a build (from the PetClinic job) is Stable or Unstable but not failed. Also, note that the **BUILD_NUMBER** parameter is passed here.

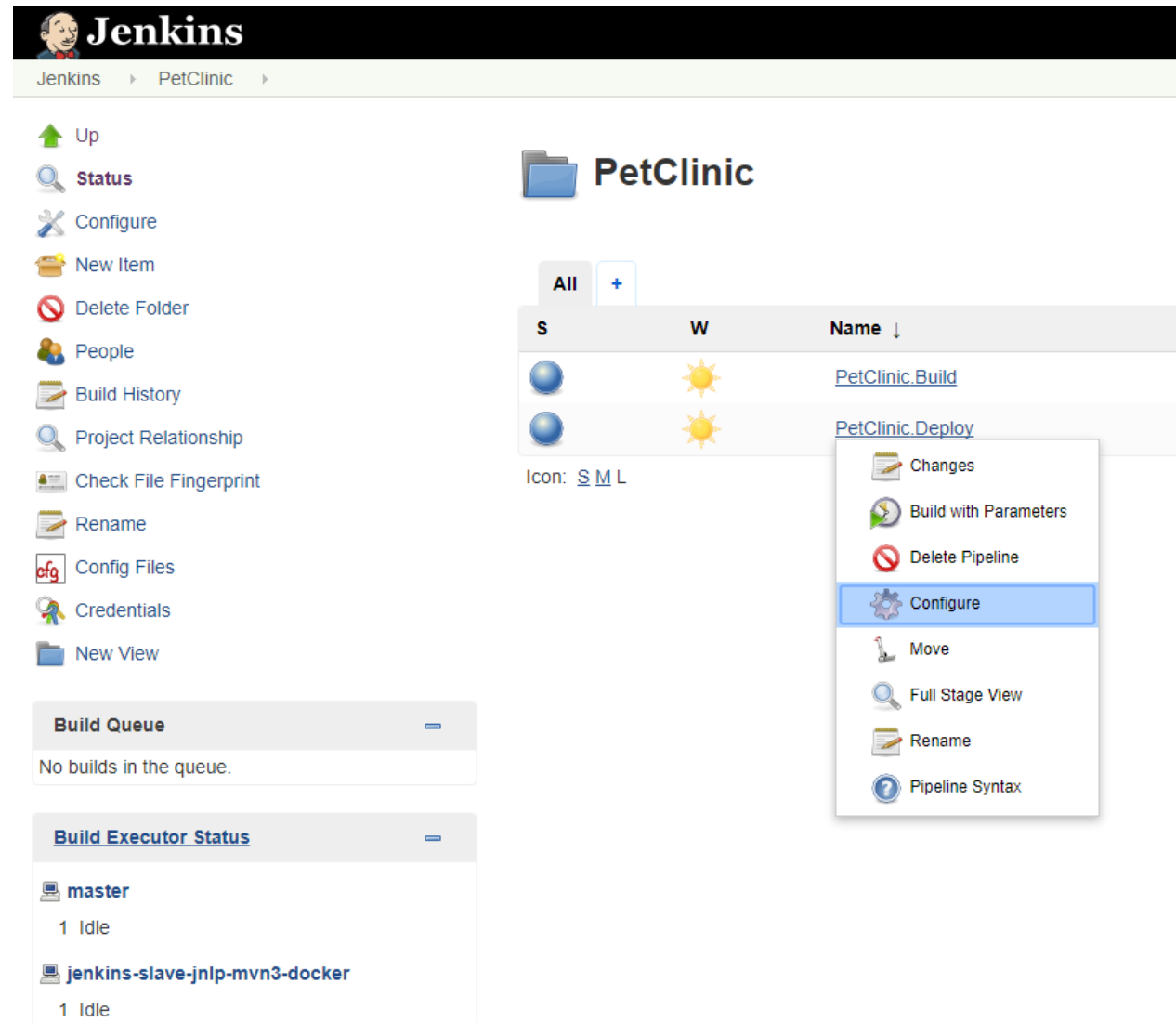


The image shows the 'Trigger parameterized build on other projects' section of the Jenkins configuration page. The 'Build Triggers' section is expanded. It contains the following settings: 'Projects to build' set to 'PetClinic.Deploy', 'Trigger when build is' set to 'Stable or unstable but not failed' (both highlighted with an orange box), 'Trigger build without parameters' unchecked, and 'Trigger build from child projects' unchecked. Below this is the 'Predefined parameters' section, which contains a 'Parameters' field with the value 'PARENT_BUILD_NUMBER \${BUILD_NUMBER}' (highlighted with an orange box).

33. Click the **Jenkins** breadcrumb at the top of the screen to return to the dashboard.



34. Mouseover **PetClinic.Deploy** and click the down arrow. Select **Configure** from the resulting menu.



Jenkins

Jenkins > PetClinic

Up

Status

Configure

New Item

Delete Folder

People

Build History

Project Relationship

Check File Fingerprint

Rename

Config Files

Credentials

New View

PetClinic

All +

S	W	Name ↓
		PetClinic.Build
		PetClinic.Deploy

Icon: [S](#) [M](#) [L](#)

- Changes
- Build with Parameters
- Delete Pipeline
- Configure**
- Move
- Full Stage View
- Rename
- Pipeline Syntax

Build Queue

No builds in the queue.

Build Executor Status

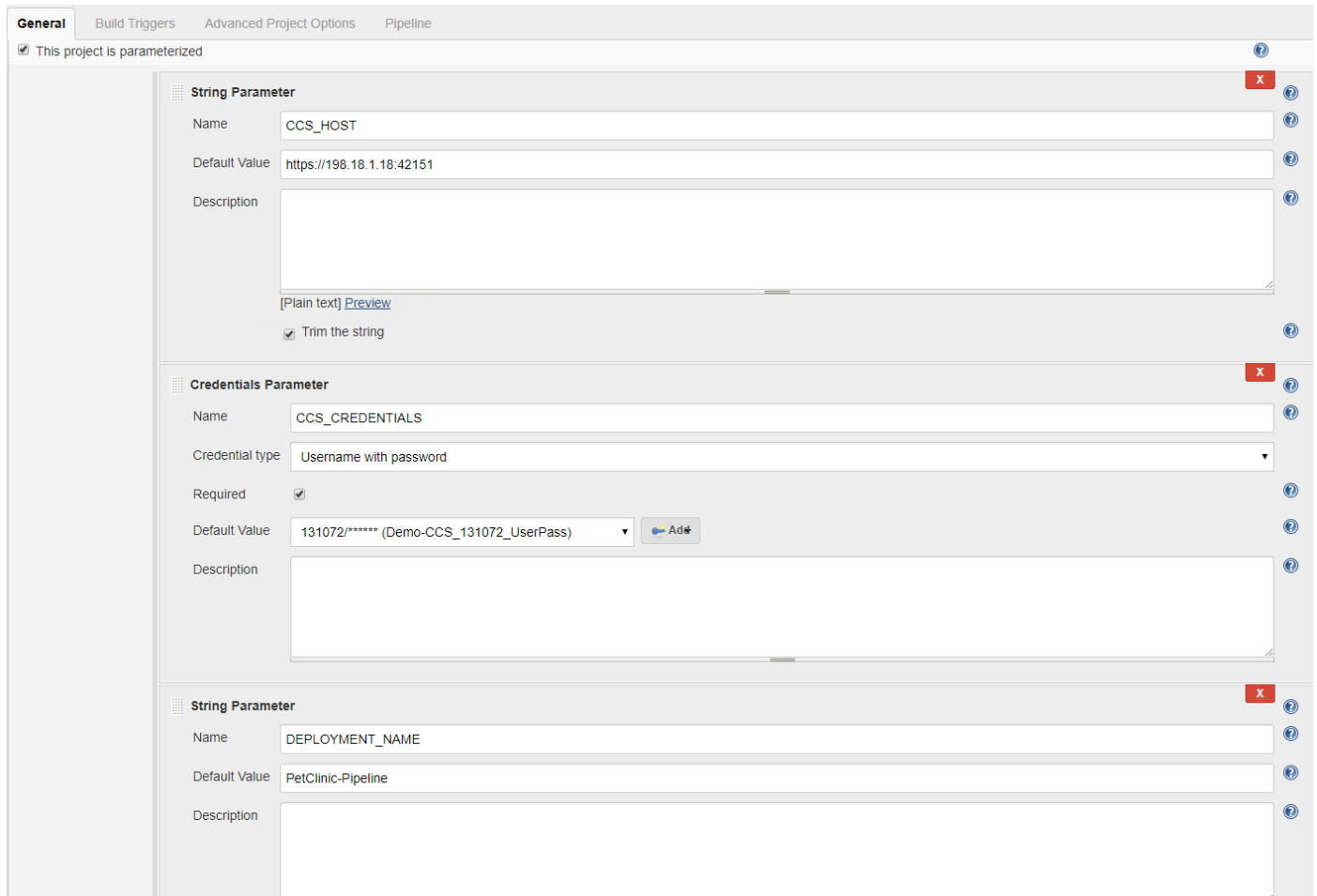
master

1 Idle

jenkins-slave-jnlp-mvn3-docker

1 Idle

35. Scroll to the This project is parameterized section to observe the parameters used to connect to CloudCenter, namely **CCS_HOST** and **CCS_CREDENTIALS**, as well as the **DEPLOYMENT_NAME** to define the name of the deployment in CloudCenter.



The screenshot shows the 'General' tab of a project configuration interface. At the top, there are tabs for 'General', 'Build Triggers', 'Advanced Project Options', and 'Pipeline'. Below the tabs, a checkbox labeled 'This project is parameterized' is checked. The main content area displays three parameter sections:

- String Parameter**:
 - Name: CCS_HOST
 - Default Value: https://198.18.1.18:42151
 - Description: [Plain text] [Preview](#)
 - ☒ Trim the string
- Credentials Parameter**:
 - Name: CCS_CREDENTIALS
 - Credential type: Username with password
 - Required: ☒
 - Default Value: 131072/***** (Demo-CCS_131072_UserPass) [Add](#)
 - Description: [Plain text] [Preview](#)
- String Parameter**:
 - Name: DEPLOYMENT_NAME
 - Default Value: PetClinic-Pipeline
 - Description: [Plain text] [Preview](#)
 - ☒ Trim the string

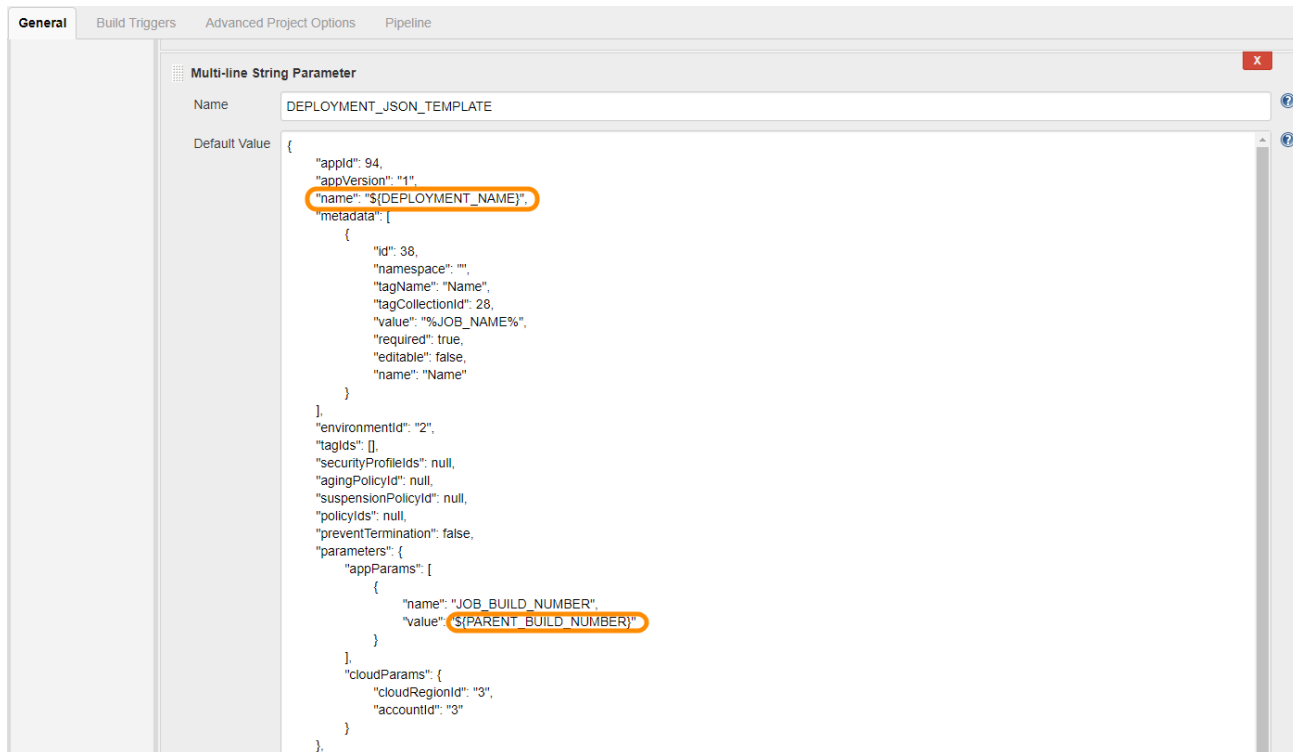
36. Further down the parameters list, observe the **PARENT_BUILD_NUMBER** used to receive the index to the latest compiled java binary.



The screenshot shows the 'General' tab of the project configuration interface, focusing on the 'String Parameter' section for PARENT_BUILD_NUMBER:

- String Parameter**:
 - Name: PARENT_BUILD_NUMBER
 - Default Value: [Empty field]
 - Description: [Plain text] [Preview](#)
 - ☒ Trim the string

37. To observe the API template used to initiate the deployment against CloudCenter, scroll to the **DEPLOYMENT_JSON_TEMPLATE** parameter.



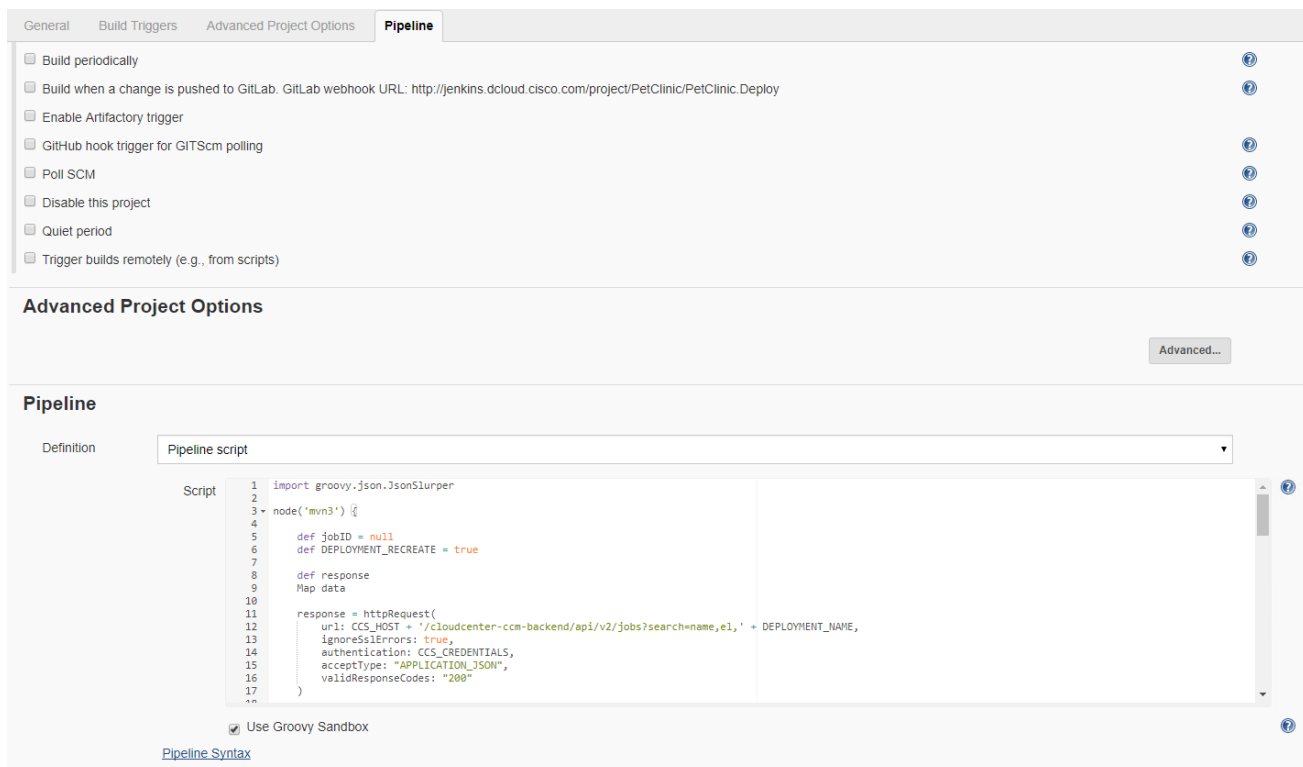
Multi-line String Parameter

Name:

Default Value:

```
{
  "appId": 94,
  "appVersion": "1",
  "name": "${DEPLOYMENT_NAME}",
  "metadata": {
    {
      "id": 38,
      "namespace": "",
      "tagName": "Name",
      "tagCollectionId": 28,
      "value": "%JOB_NAME%",
      "required": true,
      "editable": false,
      "name": "Name"
    }
  },
  "environmentId": "2",
  "tags": [],
  "securityProfiles": null,
  "agingPolicyId": null,
  "suspensionPolicyId": null,
  "policyIds": null,
  "preventTermination": false,
  "parameters": {
    "appParams": [
      {
        "name": "JOB_BUILD_NUMBER",
        "value": "${PARENT_BUILD_NUMBER}"
      }
    ],
    "cloudParams": {
      "cloudRegionId": "3",
      "accountId": "3"
    }
  }
}
```

38. To observe the pipeline script used to interface with CloudCenter and manage the deployments, scroll down to the Pipeline section.



General **Build Triggers** **Advanced Project Options** **Pipeline**

☐ Build periodically

☐ Build when a change is pushed to GitLab. GitLab webhook URL: <http://jenkins.dcloud.cisco.com/project/PetClinic/PetClinic.Deploy>

☐ Enable Artifactory trigger

☐ GitHub hook trigger for GITScm polling

☐ Poll SCM

☐ Disable this project

☐ Quiet period

☐ Trigger builds remotely (e.g., from scripts)

Advanced Project Options

[Advanced...](#)

Pipeline

Definition:

Script:

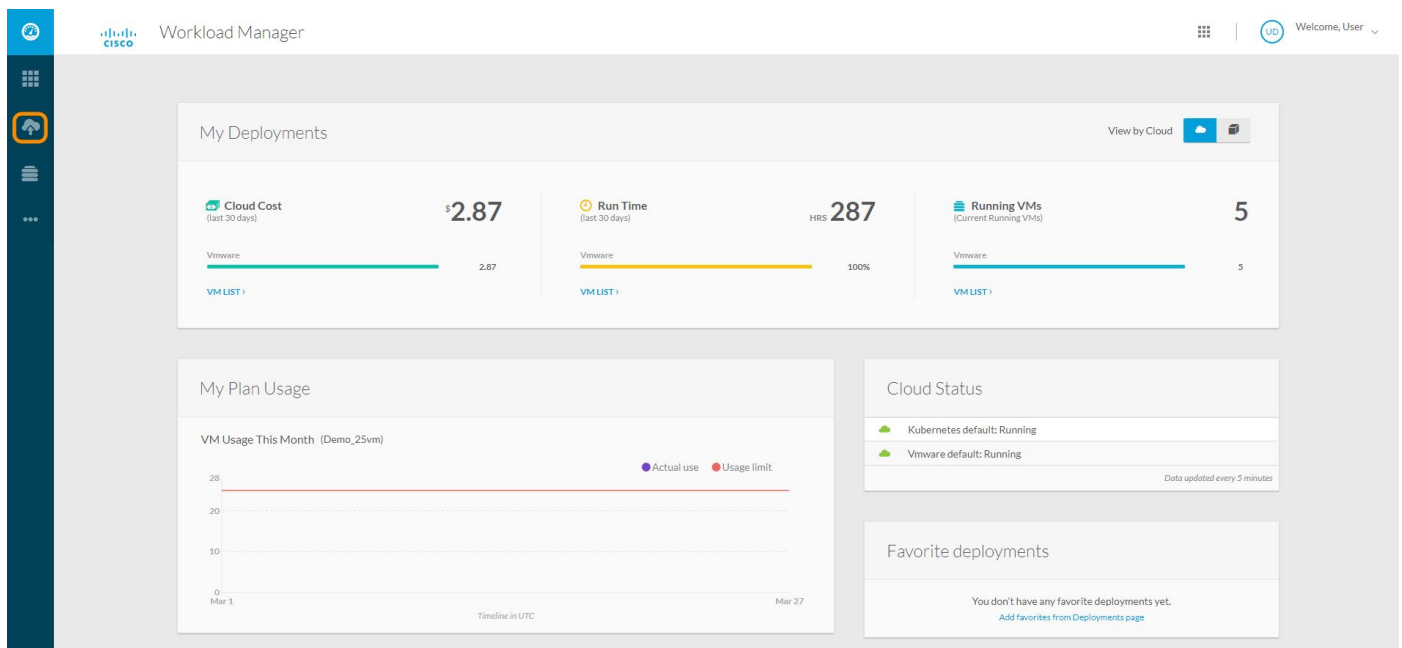
```
1 import groovy.json.JsonSlurper
2
3 node('mvn3') {
4
5     def jobId = null
6     def DEPLOYMENT_RECREATE = true
7
8     def response
9     Map data
10
11     response = httpRequest(
12         uri: CCS_HOST + '/cloudcenter-ccm-backend/api/v2/jobs?search=name,e1,' + DEPLOYMENT_NAME,
13         ignoreSslErrors: true,
14         authentication: CCS_CREDENTIALS,
15         acceptType: "APPLICATION_JSON",
16         validResponseCodes: "200"
17     )
18 }
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Interact with Application Code

1. Open a **Chrome** browser and perform the following:
 - a. Open a new browser tab and click the **CloudCenter Suite** bookmark. Log in using the credentials **user@dcloud.cisco.com / C1sco12345**, tenant id: **demo**.
 - b. Open a second browser tab and click the **GitLab** bookmark. Log in using the credentials **demouser / C1sco12345**.
 - c. Open a third browser tab and click the **Jenkins** bookmark. Log in using the credentials **admin / C1sco12345**.
 - d. Open a forth browser tab and click the **Artifactory** bookmark. Log in using the credentials **admin / C1sco12345**.
2. Switch to the **CloudCenter** browser tab
3. Select the **Workload Manager** module.
4. Click on **Deployments**.

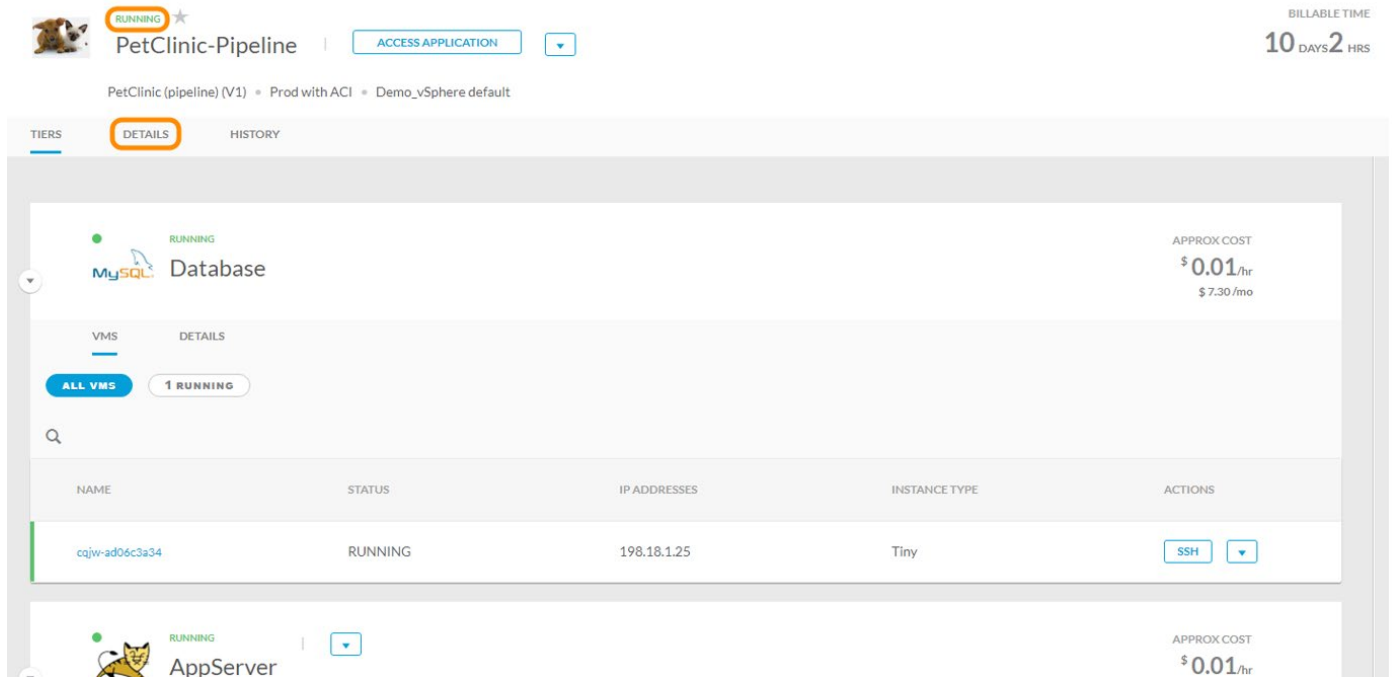


5. From the list of running deployments, locate and click on the **PetClinic-Pipeline** deployment.

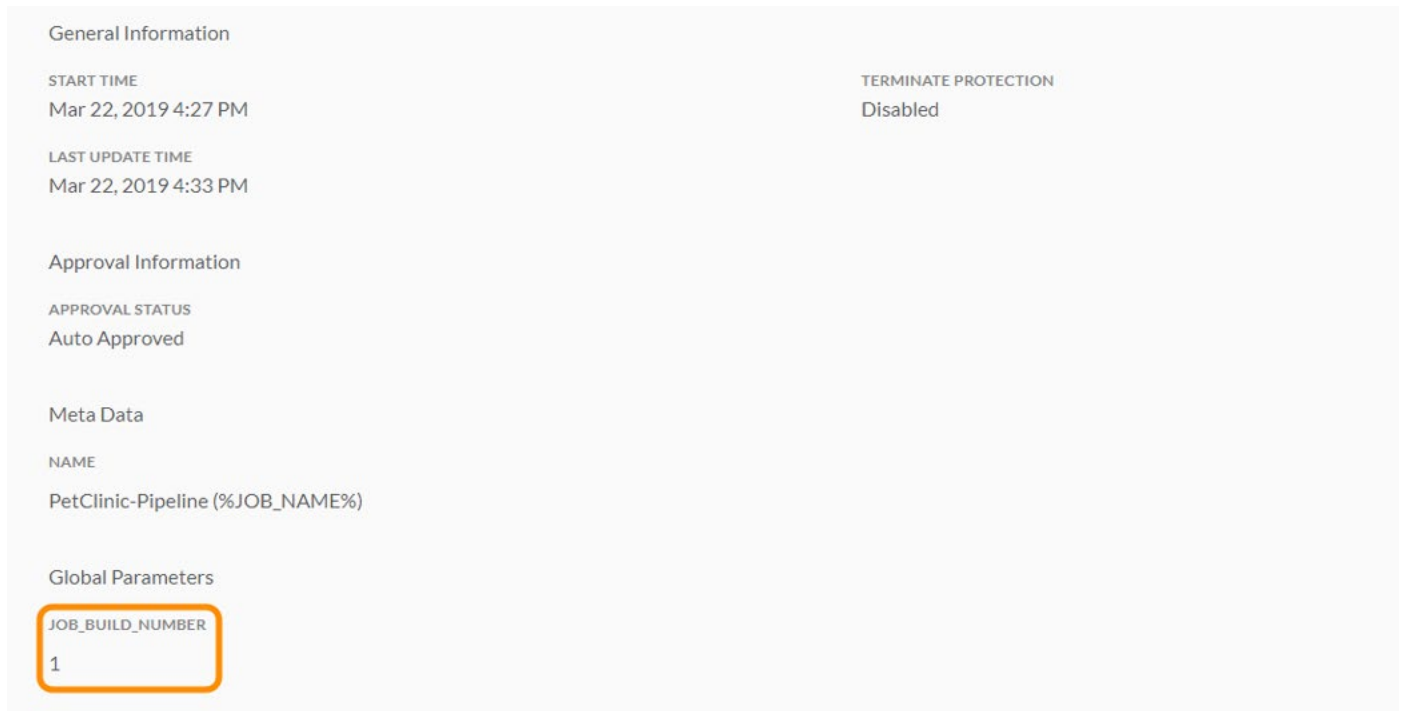
The screenshot shows the 'Deployments' page in the Cisco dCloud interface. It features a table listing various deployments. The 'PetClinic-Pipeline' deployment is highlighted with an orange box. The table includes columns for deployment details such as Name, Status, Environment, Start Time, Billable Time, Approx Cost, Approx Savings, Accrued Cost, and Actions.

NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	APPROX SAVINGS	ACCRUED COST	ACTIONS
MyOpencart-con OpenCart (con) (v1) Demo_K8s default	Deployed	Prod	Mar 27, 2019 at 11:15 AM	Not Available	\$0.00/hr \$0.00/mo		\$0	
MyMagento Magento (vm) (v2.0.0.0) Demo_vSphere default	Deployed	Prod with ACI	Mar 27, 2019 at 09:40 AM	12 hrs	\$0.03/hr \$21.90/mo		\$0.12	
PetClinic-Pipeline PetClinic (pipeline) (v1) Demo_vSphere default	Deployed	Prod with ACI	Mar 22, 2019 at 04:27 PM	9 days 18 hrs	\$0.02/hr \$14.40/mo		\$2.34	

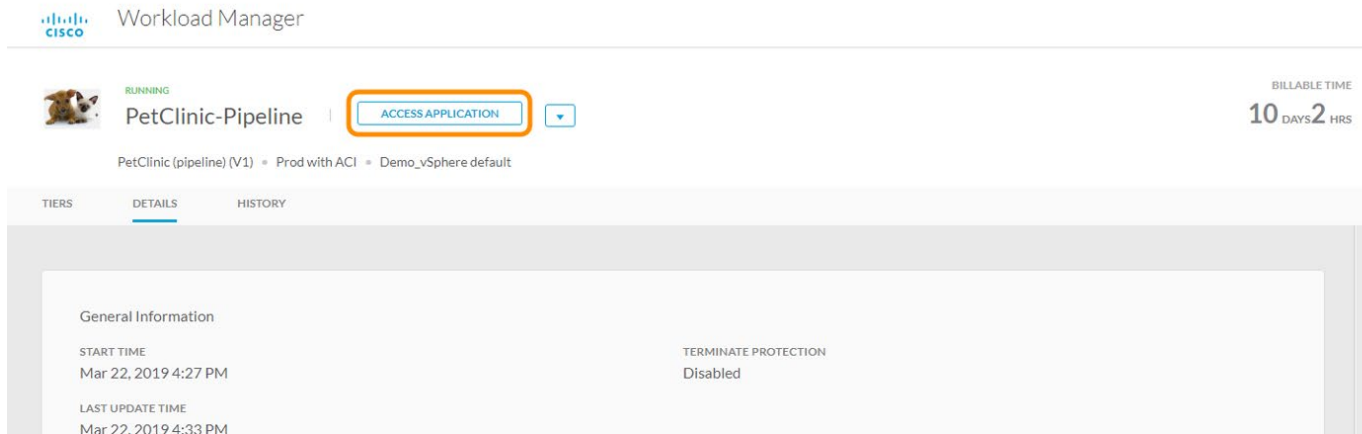
6. Notice the Application already deployed and currently in **Running** state. Click on **Details** at the top of the form



7. Note the **JOB_BUILD_NUMBER** which has the value of 1, pointing to the first compiled artifact having been produced by the CI/CD pipeline and already deployed



8. To access the already deployed application, click on the **Access Application** button at the top of the form



The screenshot shows the Cisco Workload Manager interface. At the top, there's a header with the Cisco logo and 'Workload Manager'. Below this, the application 'PetClinic-Pipeline' is shown with a 'RUNNING' status and an 'ACCESS APPLICATION' button highlighted with an orange box. To the right, the 'BILLABLE TIME' is displayed as '10 DAYS 2 HRS'. Below the application name, there's a breadcrumb trail: 'PetClinic (pipeline) (V1) > Prod with ACI > Demo_vSphere default'. The main content area has tabs for 'TIERS', 'DETAILS', and 'HISTORY', with 'DETAILS' selected. Under 'DETAILS', there's a 'General Information' section with fields for 'START TIME' (Mar 22, 2019 4:27 PM), 'LAST UPDATE TIME' (Mar 22, 2019 4:33 PM), and 'TERMINATE PROTECTION' (Disabled).

9. Note the Welcome message, which is going to be changed in the following steps by editing the application source code.

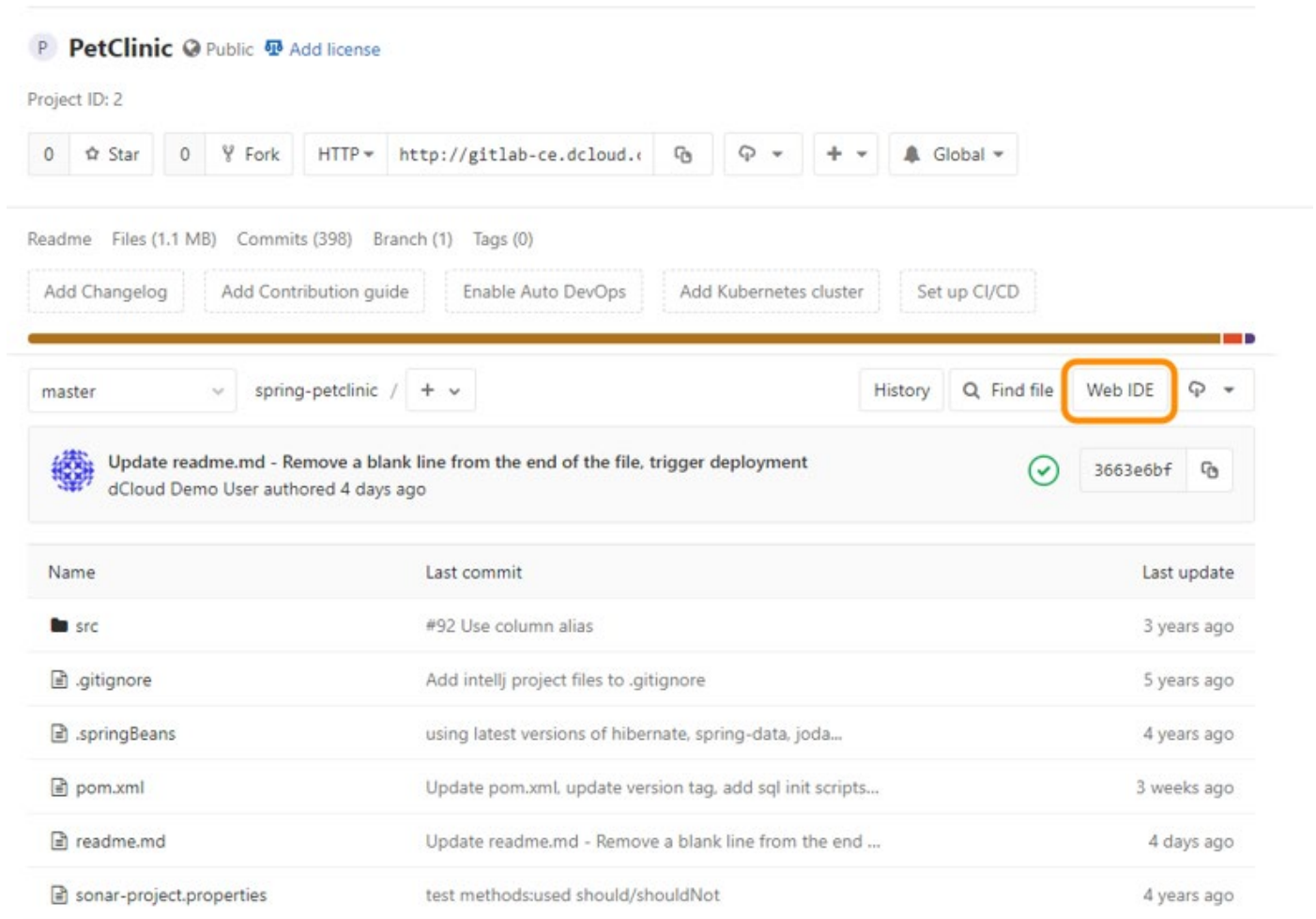


10. Switch to the **GitLab** browser tab
11. Click on the **dCloud Demo User / PetClinic** project.



The screenshot shows the GitLab Projects page. At the top, there's a 'Projects' header with a 'New project' button. Below this, there's a section for 'Your projects' with tabs for 'Your projects', 'Starred projects', and 'Explore projects'. A search bar 'Filter by name...' and a dropdown 'Last updated' are also present. Under 'Your projects', there's a list of projects. The first project is 'dCloud Demo User / PetClinic', which is highlighted with an orange box. It has a 'Maintainer' role and a status of 'updated 4 days ago'.

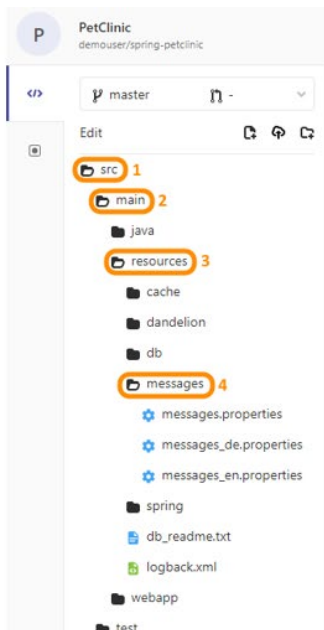
12. To start editing the source code in this git repository using an IDE, click on the **Web IDE** button at the top-right corner.



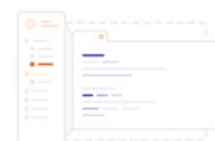
The screenshot shows the GitLab repository page for 'PetClinic'. At the top, there's a header with 'PetClinic', 'Public', and 'Add license'. Below that, 'Project ID: 2' is shown. A row of buttons includes '0 Star', '0 Fork', 'HTTP', a URL 'http://gitlab-ce.dcloud.c...', a 'Clone' button, a '+', and a 'Global' dropdown. A secondary row of buttons includes 'Add Changelog', 'Add Contribution guide', 'Enable Auto DevOps', 'Add Kubernetes cluster', and 'Set up CI/CD'. Below these is a breadcrumb 'master > spring-petclinic /' with a '+' dropdown. To the right are 'History', 'Find file', and the 'Web IDE' button, which is highlighted with an orange box. Below the breadcrumb is a commit message 'Update readme.md - Remove a blank line from the end of the file, trigger deployment' by 'dCloud Demo User' from 4 days ago, with a green checkmark and commit hash '3663e6bf'. A table of files follows:

Name	Last commit	Last update
src	#92 Use column alias	3 years ago
.gitignore	Add intellj project files to .gitignore	5 years ago
.springBeans	using latest versions of hibernate, spring-data, joda...	4 years ago
pom.xml	Update pom.xml, update version tag, add sql init scripts...	3 weeks ago
readme.md	Update readme.md - Remove a blank line from the end ...	4 days ago
sonar-project.properties	test methods:used should/shouldNot	4 years ago

13. Once the Web IDE is up, navigate to **src / main / resources / messages**.



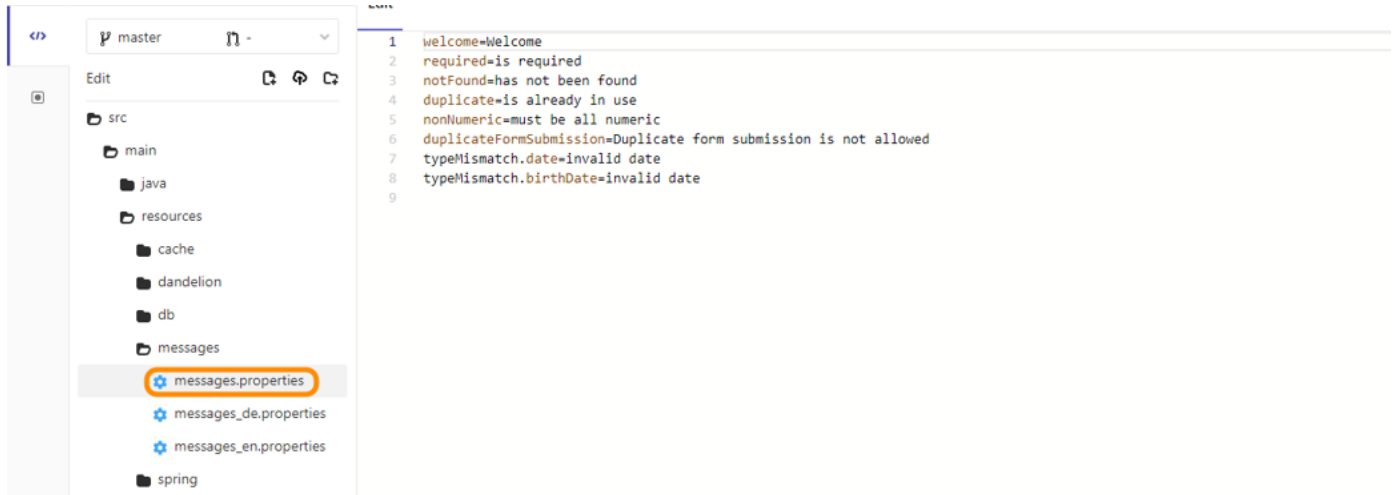
The screenshot shows the GitLab Web IDE interface. On the left, a file explorer sidebar is visible. The path 'src / main / resources / messages' is highlighted with orange boxes and numbered 1 through 4. The 'src' folder is circled with a red box and labeled '1'. The 'main' folder is circled with a red box and labeled '2'. The 'resources' folder is circled with a red box and labeled '3'. The 'messages' folder is circled with a red box and labeled '4'. Below the folders, several files are listed, including 'messages.properties', 'messages_de.properties', 'messages_en.properties', 'spring', 'db_readme.txt', 'logback.xml', 'webapp', and 'test'.



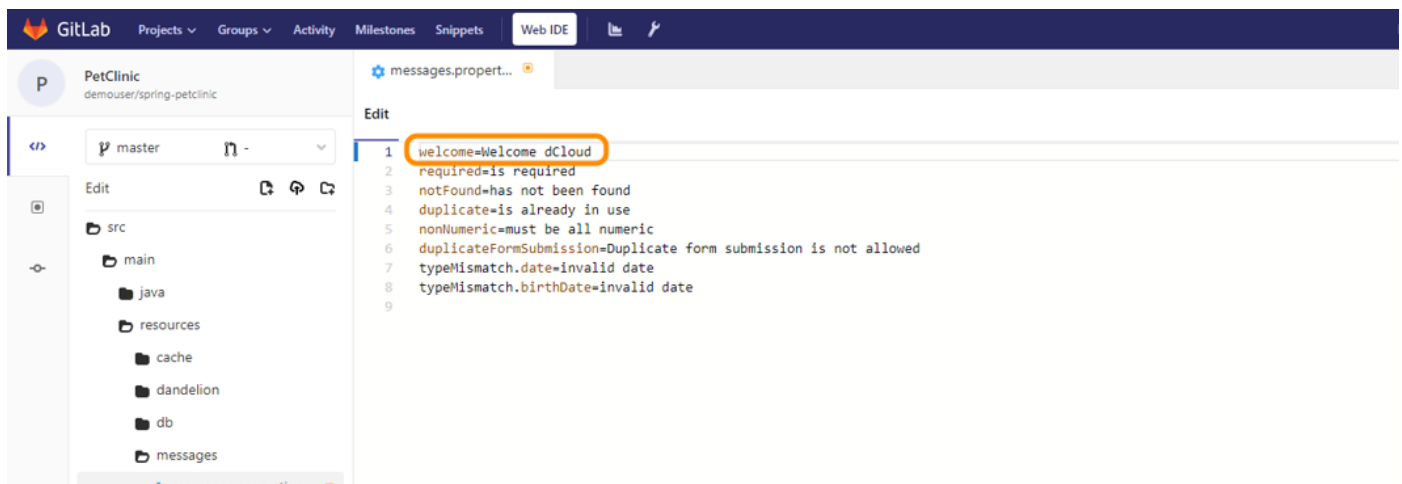
Welcome to the GitLab IDE

Select a file from the left sidebar to begin editing. Afterwards, you'll be able to commit your changes.

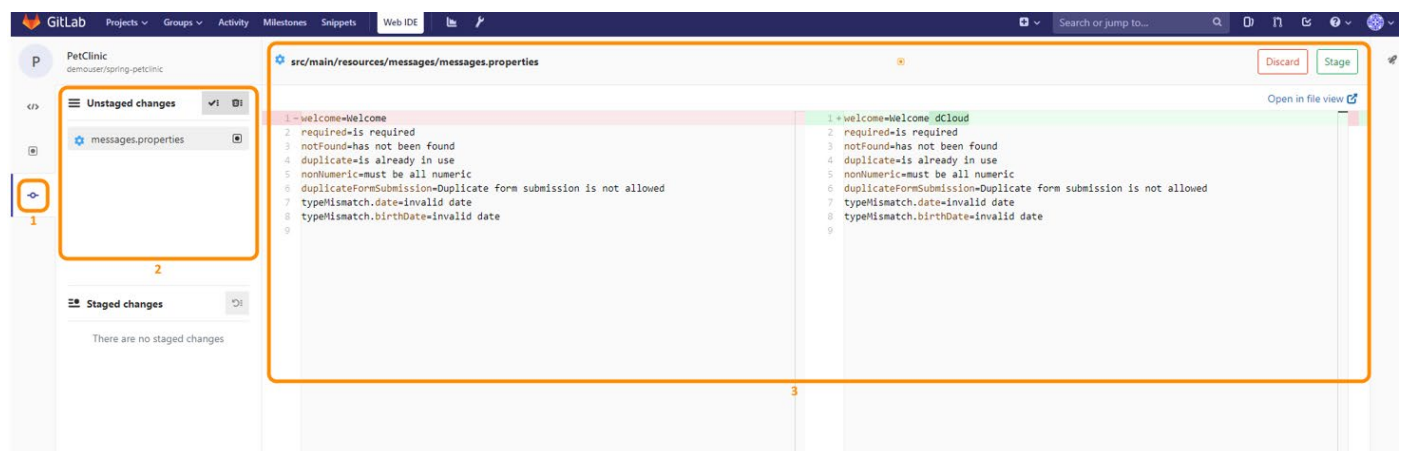
14. Click on the **messages.properties** file. Notice that a file editor has been loaded with the contents of this file.



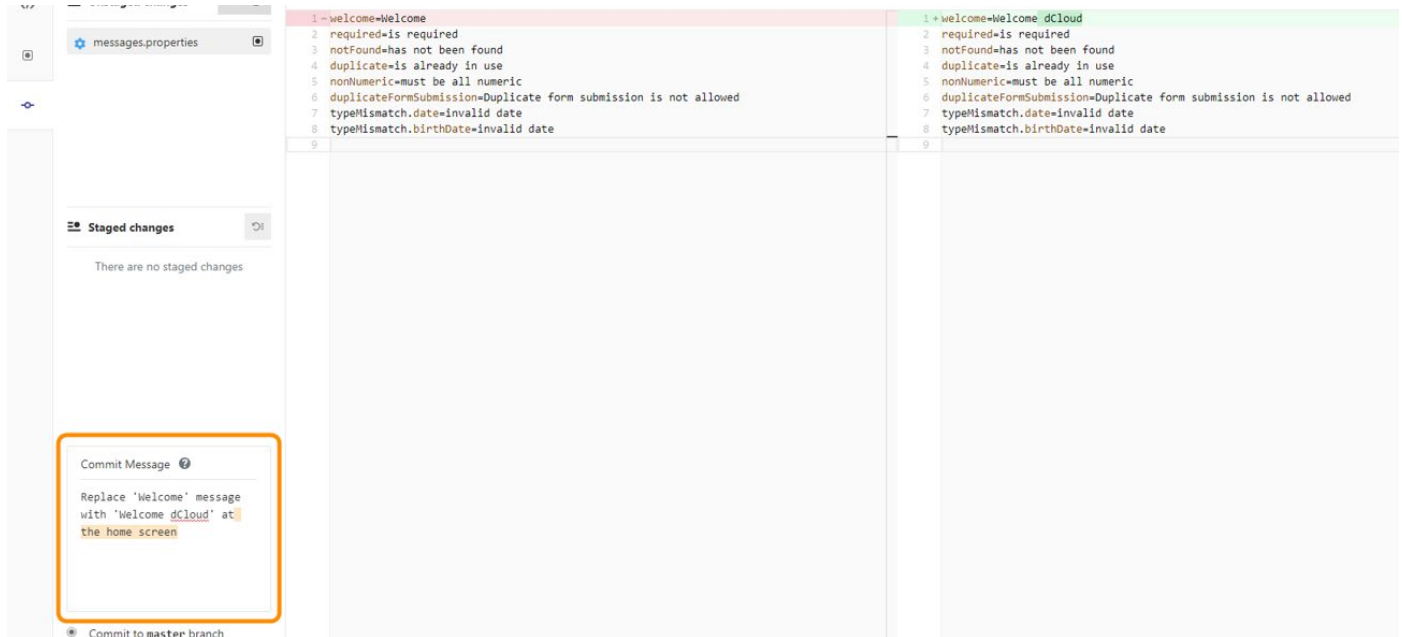
15. Change the **Welcome=Welcome** string to **Welcome=Welcome dCloud**. Notice that the Commit... button at the bottom-left corner has been enabled, indicating there are changes pending in this commit. Do not press the Commit... button at this stage.



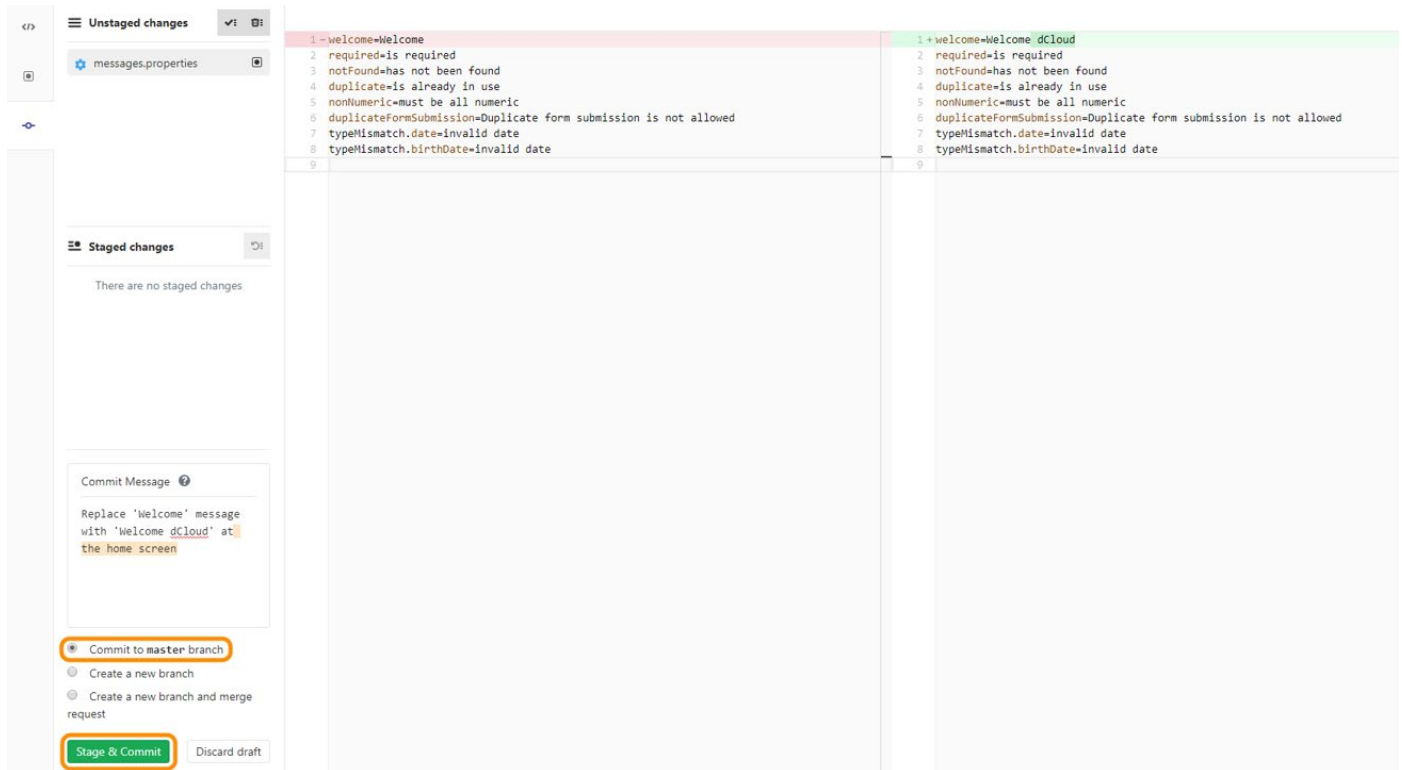
16. At the left-hand side, click on the **Commit** icon to view the changed files in this commit and review the changes for each file in this commit. In this case, only one file has changed, the **messages.properties** file.



17. At the **Commit Message** text box, enter the text **Replace Welcome message** with **Welcome dCloud** at the home screen.



18. Next, leave the default setting to **Commit** to master branch and click on the **Stage & Commit** button at the bottom of the page. At the stage, GitLab will notify Jenkins there is a new build pending.



19. Switch to the **Jenkins browser** tab to observe the relevant jobs being executed. First, the **PetClinic/PetClinic.Build** job will be queued

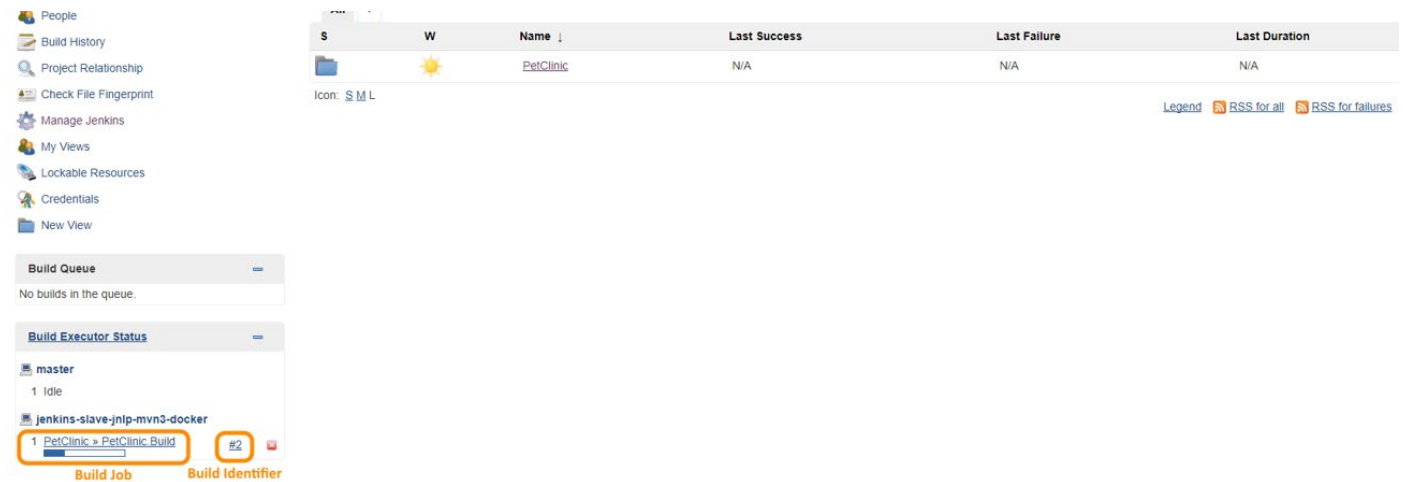


The screenshot shows the Jenkins web interface. On the left is a sidebar with navigation links: People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, Credentials, and New View. The main area displays a table of jobs. The table has columns: S, W, Name, Last Success, Last Failure, and Last Duration. A single job is listed: 'PetClinic' with a status of 'S' (Success) and a weather icon. Below the table, there is a 'Build Queue (1)' section showing 'PetClinic > PetClinic.Build'.

S	W	Name	Last Success	Last Failure	Last Duration
S	☀	PetClinic	N/A	N/A	N/A

Build Queue (1)
PetClinic > PetClinic.Build

20. *[Observe-only step]* After a few seconds, the **PetClinic/PetClinic.Build** job will be executed at the available Jenkins slave node. Please note that the Build number next to the build job represents the identifier that is passed to the subsequent tasks (only if the build is successful), which points to the new java artifact(s) to be deployed.



The screenshot shows the Jenkins web interface. The 'Build Queue' section now says 'No builds in the queue.' The 'Build Executor Status' section shows the 'jenkins-slave-jnp-mvn3-docker' node with one build in progress: '1 PetClinic > PetClinic.Build'. The build number '1' is circled in orange, and the build identifier '#2' is also circled in orange. Below the build entry, there are labels 'Build Job' and 'Build Identifier'.

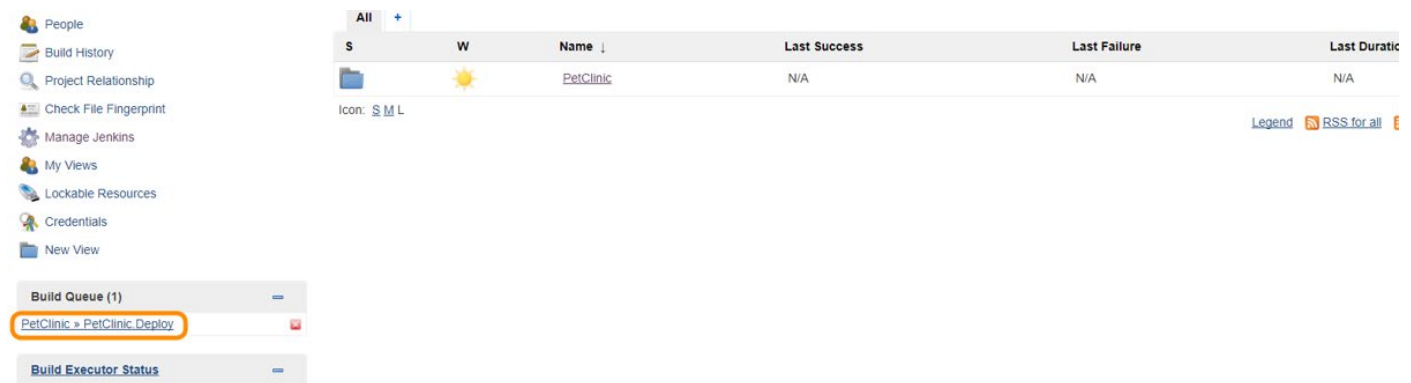
S	W	Name	Last Success	Last Failure	Last Duration
S	☀	PetClinic	N/A	N/A	N/A

Build Queue
No builds in the queue.

Build Executor Status
master
1 Idle
jenkins-slave-jnp-mvn3-docker
1 PetClinic > PetClinic.Build #2

Build Job Build Identifier

21. *[Observe-only step]* Once the PetClinic/PetClinic.Build job is complete (if successful), the PetClinic/PetClinic.Deploy job is queued for execution.



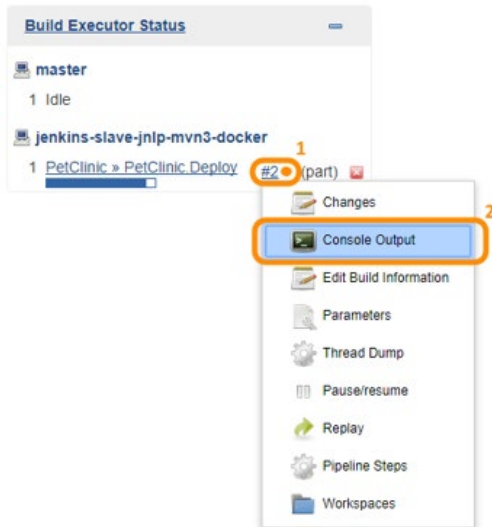
The screenshot shows the Jenkins web interface. The 'Build Queue' section now shows 'PetClinic > PetClinic.Deploy'. The 'Build Executor Status' section is empty.

S	W	Name	Last Success	Last Failure	Last Duration
S	☀	PetClinic	N/A	N/A	N/A

Build Queue (1)
PetClinic > PetClinic.Deploy

Build Executor Status

22. After a few seconds, the **PetClinic/PetClinic.Deploy** job will be executed at the available Jenkins slave node. Once the job enters the execution phase, **click on the small down arrow** next to the job number and click on **Console Output** to observe the progress of the job.



23. Observe the output logs of this job while executed (comments have been added to the output logs in order to explain the tasks being executed).

Console Output

Started by upstream project "PetClinic/PetClinic.Build" build number 2
originally caused by:
Started by GitLab push by dCloud Demo User
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on jenkins-slave-jnlp-mvn3-docker in /home/jenkins/workspace/PetClinic/PetClinic.Deploy
[Pipeline] {
[Pipeline] httpRequest
HttpMethod: GET
URL: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs?search-name=el.PetClinic-Pipeline>
Accept: application/json
Using authentication: Demo-CCS_131872_UserPass
Sending request to url: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs?search-name=el.PetClinic-Pipeline>
Response Code: HTTP/1.1 200
Success code from [200..200]
[Pipeline] echo
Deployment PetClinic-Pipeline found. Deleting ...
[Pipeline] httpRequest
HttpMethod: DELETE
URL: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1>
Accept: application/json
Using authentication: Demo-CCS_131872_UserPass
Sending request to url: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1>
Response Code: HTTP/1.1 202
Success code from [202..202]
[Pipeline] echo
Waiting for deployment to finish deleting ...
[Pipeline] sleep
Sleeping for 10 sec
[Pipeline] httpRequest
HttpMethod: GET
URL: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1>
Accept: application/json
Using authentication: Demo-CCS_131872_UserPass
Sending request to url: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1>
Response Code: HTTP/1.1 200
Success code from [200..200]
[Pipeline] echo
status:JobStopping
[Pipeline] sleep
Sleeping for 10 sec
[Pipeline] httpRequest
HttpMethod: GET
URL: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1>
Accept: application/json
Using authentication: Demo-CCS_131872_UserPass
Sending request to url: <https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1>

```
Jenkins > PetClinic > PetClinic.Deploy > #2
[Pipeline] httpRequest
  HttpMethod: GET
  URL: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1
  Accept: application/json
  Using authentication: Demo-CCS_131872_UserPass
  Sending request to url: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1
  Response Code: HTTP/1.1 200
  Success code from [200..200]
  [Pipeline] echo
    status:JobStopping
  [Pipeline] sleep
    Sleeping for 10 sec
  [Pipeline] httpRequest
  HttpMethod: GET
  URL: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1
  Accept: application/json
  Using authentication: Demo-CCS_131872_UserPass
  Sending request to url: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1
  Response Code: HTTP/1.1 200
  Success code from [200..200]
  [Pipeline] echo
    status:JobStopping
  [Pipeline] sleep
    Sleeping for 10 sec
  [Pipeline] httpRequest
  HttpMethod: GET
  URL: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1
  Accept: application/json
  Using authentication: Demo-CCS_131872_UserPass
  Sending request to url: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs/1
  Response Code: HTTP/1.1 200
  Success code from [200..200]
  [Pipeline] echo
    status:JobCanceled
  [Pipeline] sleep
    Sleeping for 5 sec
  [Pipeline] echo
    Deploying PetClinic-Pipeline
  [Pipeline] httpRequest
  HttpMethod: POST
  URL: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs
  Accept: application/json
  Content-type: application/json
  Using authentication: Demo-CCS_131872_UserPass
  Sending request to url: https://198.18.1.18:42151/cloudcenter-ccm-backend/api/v2/jobs
  Response Code: HTTP/1.1 200
  Success code from [200..200]
  [Pipeline] }
  [Pipeline] // node
  [Pipeline] End of Pipeline
Finished: SUCCESS
```

Old deployment still exists. Sleeping for 10 seconds

Old deployment still exists. Sleeping for 10 seconds

Old deployment has been terminated. Sleeping for 5 seconds

Issuing command to initiate new deployment with new build

New deployment has been initiated successfully

24. Once the new deployment request has been submitted successfully to CloudCenter, switch to the CloudCenter browser tab.
25. At the left-hand side, click on **Deployments** to observe the new deployment being rolled out. Notice the new PetClinic-Pipeline deployment being rolled out. Click on the newly created deployment to view the deployment details.

Deployments

All Deployments
All Applications
☐ Show Hidden

NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	APPROX SAVINGS	ACCRUED COST	ACTIONS
PetClinic-Pipeline PetClinic (pipeline) (v1) Demo_vSphere default	Starting	Prod with ACI	Mar 27, 2019 at 04:41 PM	Not Available	\$0.00/hr \$0.00/mo		\$0	
MyOpenCart-con OpenCart (con) (v1) Demo_vSphere default	Deployed	Prod	Mar 27, 2019 at 11:16 AM	Not Available	\$0.00/hr \$0.00/mo		\$0	
MyMagento Magento (vm) (v2.0.0.0) Demo_vSphere default	Deployed	Prod with ACI	Mar 27, 2019 at 09:40 AM	1 day 9 hrs	\$0.03/hr \$21.90/mo		\$0.33	

Show 50 per page
Page 1 of 1

Deployments

All Deployments ▾ All Applications ▾ ☐ Show Hidden 🔍 ↺ New Deployment

	NAME	STATUS	ENVIRONMENT	START TIME	BILLABLE TIME	APPROX COST	APPROX SAVINGS	ACCURED COST	ACTIONS
★	PetClinic-Pipeline PetClinic (pipeline) (V1) Demo_vSphere default	Starting	Prod with ACI	Mar 27, 2019 at 04:41 PM	Not Available	\$0.00/hr \$0.00/mo		\$0	
★	MyRedmine-con Redmine (con) (V1) Demo_vSphere default	Deployed	Prod	Mar 27, 2019 at 02:58 PM	Not Available	\$0.00/hr \$0.00/mo		\$0	
★	MySplunk Splunk (vm) (V1) Demo_vSphere default	Deployed	Prod	Mar 27, 2019 at 02:09 PM	4 hrs	\$0.01/hr \$7.30/mo		\$0.04	

26. Click on the **Details** tab and observe the **JOB_BUILD_NUMBER** having now changed to **2**, indicating the new build being used for roll-out. Click on the **Tiers** tab to return to the previous screen. Wait for the new deployment to fully deploy and then proceed to the next step (5-10min).

JOBINPROGRESS ★
PetClinic-Pipeline
▾

PetClinic (pipeline) (V1) • Prod with ACI • Demo_vSphere default

TIERS 2
DETAILS
HISTORY

General Information

START TIME

Mar 27, 2019 4:41 PM

LAST UPDATE TIME

Mar 27, 2019 4:41 PM

TERMINATE PROTECTION

Disabled

Approval Information

APPROVAL STATUS

27. Once the deployment completes, click on the **Access Application** button.

RUNNING
PetClinic-Pipeline
ACCESS APPLICATION
▾

PetClinic (pipeline) (V1) • Prod with ACI • Demo_vSphere default

TIERS
DETAILS
HISTORY

RUNNING
Database

VMS

DETAILS

ALL VMS
1 RUNNING

NAME

STATUS

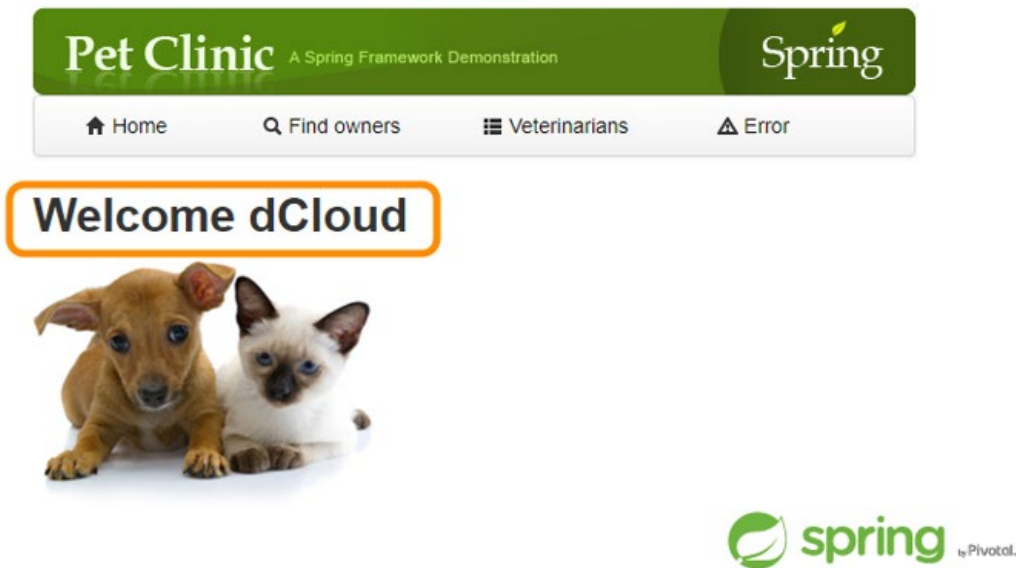
IP ADDRESSES

INSTANCE TYPE

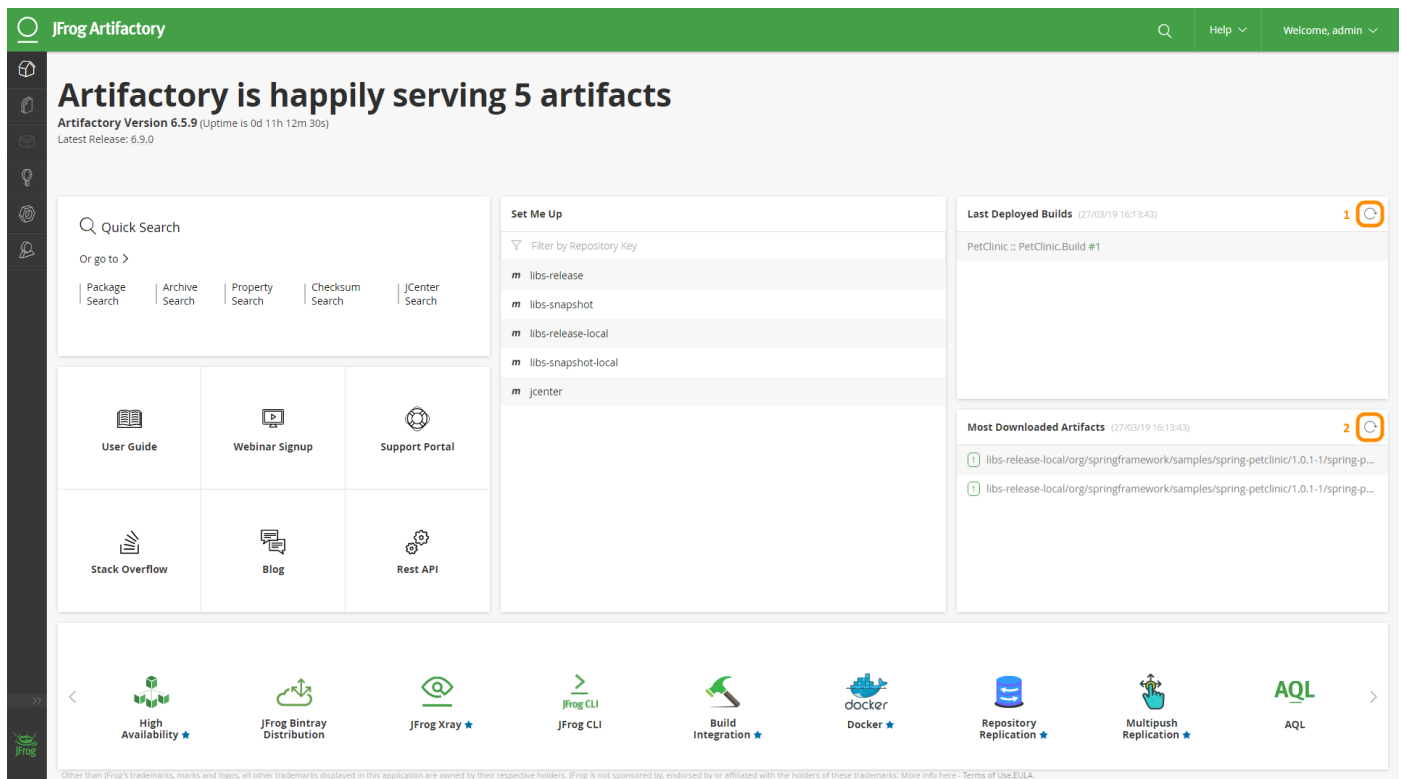
ACTIONS

cqjw-e77bafd82	RUNNING	198.18.1.35	Tiny	▾
----------------	---------	-------------	------	---

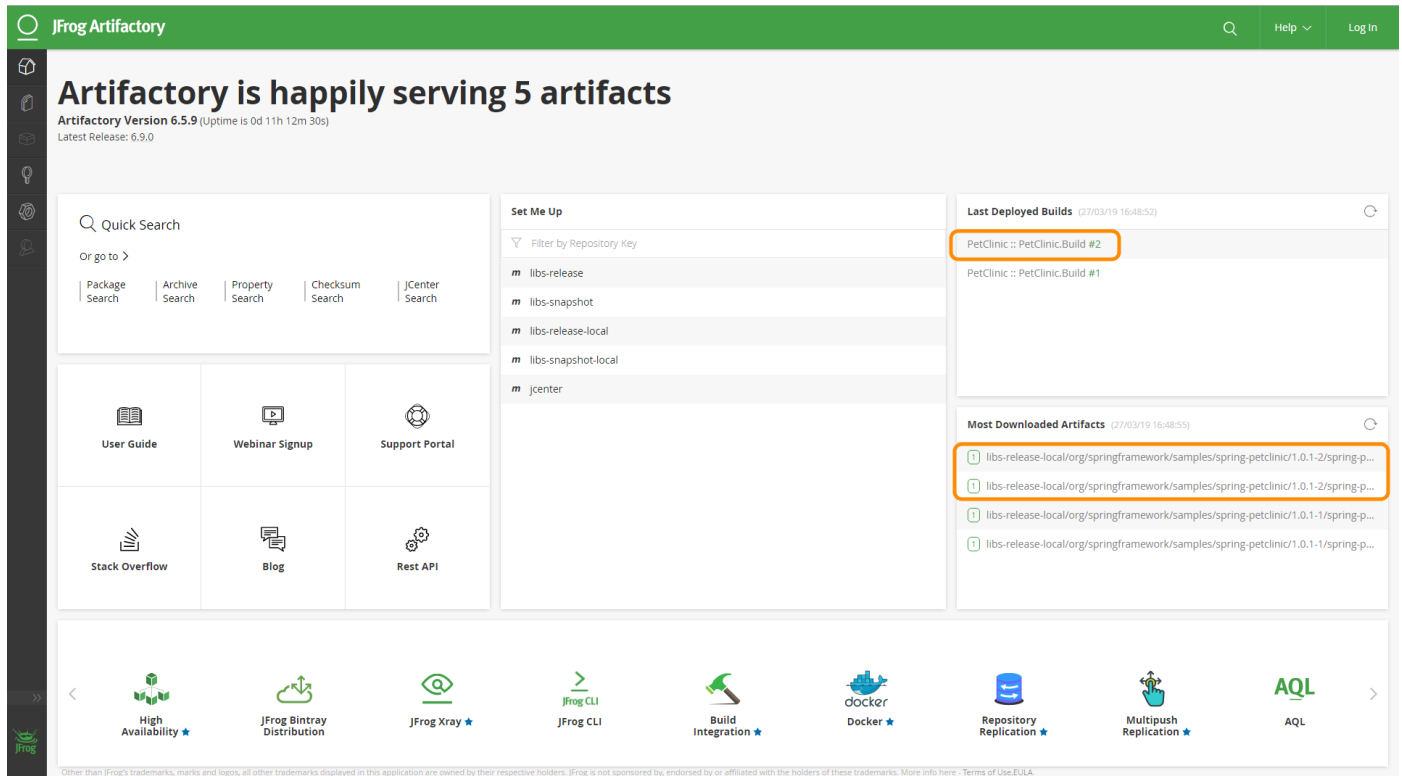
28. You should witness the Welcome message at the home screen having changed to Welcome dCloud.



29. To observe the newly compiled artifacts and their repository location, switch to the Artifactory browser tab. The Web GUI should still be showing the old artifacts. Click on the **refresh** buttons next to Last Deployed Builds and Most Downloaded Artifacts.



30. After refreshing the two frames, the new artifacts should become visible at the web page.



JFrog Artifactory

Artifactory is happily serving 5 artifacts
Artifactory Version 6.5.9 (Uptime is 0d 11h 12m 30s)
Latest Release: 6.9.0

Quick Search
Or go to >
Package Search | Archive Search | Property Search | Checksum Search | JCenter Search

Set Me Up
Filter by Repository Key
m libs-release
m libs-snapshot
m libs-release-local
m libs-snapshot-local
m jcenter

Last Deployed Builds (27/03/19 16:48:52)
PetClinic :: PetClinic.Build #2
PetClinic :: PetClinic.Build #1

Most Downloaded Artifacts (27/03/19 16:48:55)
libs-release-local/org/springframework/samples/spring-petclinic/1.0.1-2/spring-p...
libs-release-local/org/springframework/samples/spring-petclinic/1.0.1-2/spring-p...
libs-release-local/org/springframework/samples/spring-petclinic/1.0.1-1/spring-p...
libs-release-local/org/springframework/samples/spring-petclinic/1.0.1-1/spring-p...

Navigation Bar:
High Availability ★ | JFrog Bintray Distribution | JFrog Xray ★ | JFrog CLI | Build Integration ★ | Docker ★ | Repository Replication ★ | Multipush Replication ★ | AQL

Other than JFrog's trademarks, marks and logos, all other trademarks displayed in this application are owned by their respective holders. JFrog is not sponsored by, endorsed by or affiliated with the holders of these trademarks. More info here - Terms of Use EULA



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)