

Infrastructure Preview

In this lab, you build a sophisticated deployment in minutes using Marketplace. This lab shows several of the GCP infrastructure services in action and illustrates the power of the platform. It introduces technologies that are covered in detail later in the class.

Objectives

In this lab, you learn how to perform the following tasks:

- Use Marketplace to build a Jenkins Continuous Integration environment.
- Verify that you can manage the service from the Jenkins UI.
- Administer the service from the Virtual Machine host through SSH.

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click Start Lab, shows how long Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access the Google Cloud Platform for the duration of the lab.


What you need

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).
 - Time to complete the lab.
- Note:** If you already have your own personal GCP account or project, do not use it for this lab.

Task 1: Use Marketplace to build a deployment

Navigate to Marketplace

1. In the GCP Console, on the **Navigation menu** () , click **Marketplace**.
2. Locate the Jenkins deployment by searching for **Jenkins Certified by Bitnami**.
3. Click on the deployment and read about the service provided by the software.

Jenkins is an open-source continuous integration environment. You can define jobs in Jenkins that can perform tasks such as running a scheduled build of software and backing up data. Notice the software that is installed as part of Jenkins shown in the left side of the description.

The service you are using, Marketplace, is part of Google Cloud Platform. The Jenkins template is developed and maintained by an ecosystem partner named Bitnami. Notice on the left side a field that says "Last updated." How recently was this template updated?

GCP Marketplace lets you quickly deploy functional software packages by providing pre-defined templates with which GCP service?



Firestore



Terraform



Template Manager



Deployment Manager

Submit

Launch Jenkins

1. Click **Launch on Compute Engine**.
2. Verify the deployment, accept the terms and services and click **Deploy**.
3. Click **Close** on the Welcome to Deployment Manager window.

It will take a minute or two for Deployment Manager to set up the deployment. You can watch the status as tasks are being performed. Deployment Manager is acquiring a virtual machine instance and installing and configuring software for you. You will see **jenkins-1 has been deployed** when the process is complete.

Deployment Manager is a GCP service that uses templates written in a combination of YAML, python, and Jinja2 to automate the allocation of GCP resources and perform setup tasks. Behind the scenes a virtual machine has been created. A startup script was used to install and configure software, and network Firewall Rules were created to allow traffic to the service.

Click *Check my progress* to verify the objective.

Launch Jenkins

Check my progress

Task 2: Examine the deployment

In this section, you examine what was built in GCP.

View installed software and login to Jenkins

1. In the right pane, click **More about the software** to view additional software details. Look at all the software that was installed.
2. Copy the **Admin user** and **Admin password** values to a text editor.
3. Click **Visit the site** to view the site in another browser tab. If you get an error, you might have to reload the page a couple of times.
4. Log in with the **Admin user** and **Admin password** values.
5. After logging in, you will be asked to Customize Jenkins. Click **Install suggested plugins**, and then click **Restart** after the installation is complete. The restart will take a couple of minutes.

Note: If you are getting an installation error, **retry** the installation and if it fails again, **continue** past the error and **save and finish** before restarting. The code of this solution is managed and supported by Bitnami.


Explore Jenkins

1. In the Jenkins interface, in the left pane, click **Manage Jenkins**. Look at all of the actions available. You are now prepared to manage Jenkins. The focus of this lab is GCP infrastructure, not Jenkins management, so seeing that this menu is available is the purpose of this step.
2. Leave the browser window open to the Jenkins service. You will use it in the next task.

Now you have seen that the software is installed and working properly. In the next task you will open an SSH terminal session to the VM where the service is hosted, and verify that you have administrative control over the service.

Task 3: Administer the service

View the deployment and SSH to the VM

1. In the GCP Console, on the **Navigation menu**() , click **Deployment Manager**.
2. Click **jenkins-1**.
3. Click **SSH** to connect to the Jenkins server.

The Console interface is performing several tasks for you transparently. For example, it has transferred keys to the virtual machine that is hosting the Jenkins software so that you can connect securely to the machine using SSH.

Shut down and restart the services

1. In the SSH window, enter the following command to shut down all the running services:

```
sudo /opt/bitnami/ctlscript.sh stop
```

2. Refresh the browser window for the Jenkins UI. You will no longer see the Jenkins interface because the service was shut down.
3. In the SSH window, enter the following command to restart the services:

```
sudo /opt/bitnami/ctlscript.sh restart
```

4. Return to the browser window for the Jenkins UI and refresh it. You may have to do it a couple of times before the service is reachable.
5. In the SSH window, type `exit` to close the SSH terminal session.

Task 4: Review

In a few minutes you were able to launch a complete Continuous Integration solution. You demonstrated that you had user access through the Jenkins UI, and you demonstrated that you had administrative control over Jenkins by using SSH to connect to the VM where the service is hosted and by stopping and then restarting the services.