

This tutorial shows you how to set up Jenkins on GKE to help orchestrate your software delivery pipeline.

- Creating a Kubernetes cluster with GKE.
- Installing Jenkins using Helm.
- Connecting to Jenkins.

This tutorial uses billable components of Google Cloud, including:

- Compute Engine
- Google Kubernetes Engine
- Cloud Build

Use the [Pricing Calculator](/products/calculator/) (/products/calculator/) to generate a cost estimate based on your projected usage. New Google Cloud users might be eligible for a [free trial](/free-trial/) (/free-trial).

1. [Sign in](https://accounts.google.com/Login) (https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[Go to the project selector page](https://console.cloud.google.com/projectselector2/home/dashboard) (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](/billing/docs/how-to/modify-project) (/billing/docs/how-to/modify-project).

4. Enable the Compute Engine, GKE, and Cloud Build APIs.

[Enable the APIs](https://console.cloud.google.com/flows/enableapi?apiid=compute_component,container,cloudbuild.googleapis.com) (https://console.cloud.google.com/flows/enableapi?apiid=compute_component,container,cloudbuild.googleapis.com)

First, prepare your deployment environment.

1. Activate [Cloud Shell](/shell/docs/overview) (/shell/docs/overview). Cloud Shell gives you access to the command line in Cloud Console, and includes Cloud SDK and other tools you need for Google Cloud development. Cloud Shell can take several minutes to provision.

[Activate Cloud Shell](https://console.cloud.google.com/?cloudshell=true) (https://console.cloud.google.com/?cloudshell=true)

After the process completes, you'll see the following output:

2. Set the default Compute Engine zone to `us-east1-d`:

3. Clone the sample code, or [download the zip file](#)

(https://github.com/GoogleCloudPlatform/continuous-deployment-on-kubernetes/archive/master.zip).

The Git repository contains Kubernetes manifests that you'll use to deploy Jenkins. The manifests and their settings are described in [Configuring Jenkins for GKE](/solutions/configuring-jenkins-kubernetes-engine) (/solutions/configuring-jenkins-kubernetes-engine).

4. Navigate to the sample code directory:

You can use GKE to create and manage your Kubernetes cluster.

1. Provision a Kubernetes cluster using GKE. This step can take up to several minutes to complete.

The extra scopes enable Jenkins to access Cloud Source Repositories and Container Registry.

2. Confirm that your cluster is running.

Look for **RUNNING** in the **STATUS** column.

3. Confirm that you can connect to your cluster.

If you are able to successfully connect to your cluster, Kubernetes component URLs display.

Use Helm to deploy Jenkins from the repository.

1. Download and install the Helm binary:

2. Unzip the file to your local system:

3. Add yourself as a cluster administrator in the cluster's RBAC so that you can give Jenkins permissions in the cluster:

4. Grant Tiller, the server side of Helm, the `cluster-admin` role in your cluster:

5. Initialize Helm. This ensures that the Tiller is properly installed in your cluster.

6. Ensure Helm is properly installed by running the following command:

You will see versions appear for both the client and the server as `v2.14.1`:

You will use a custom [values file](https://github.com/helm/charts/tree/master/stable/jenkins) (<https://github.com/helm/charts/tree/master/stable/jenkins>) to configure the Jenkins installation. For details on the configuration, look at the `jenkins/values.yaml` file.

1. Use the Helm CLI to deploy the chart with your configuration set:

For more information on configuring the Jenkins installation, visit the Jenkins chart's [documentation page](https://github.com/helm/charts/tree/master/stable/jenkins) (<https://github.com/helm/charts/tree/master/stable/jenkins>).

2. After that command completes, ensure the Jenkins pod goes to the `Running` state and the container is in the `READY` state:

You will see `1/1` in the `READY` column and `Running` in the `STATUS` column. It can take a few minutes for Jenkins to complete its initialization.

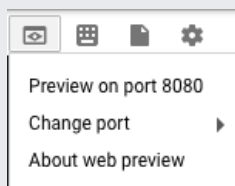
3. Set up port forwarding to the Jenkins UI from Cloud Shell:

4. Check that the Jenkins Service was created properly.

Example Output:

The Jenkins installation is using the [Kubernetes Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Kubernetes+Plugin) (<https://wiki.jenkins-ci.org/display/JENKINS/Kubernetes+Plugin>) to create builder agents. They will be automatically launched as necessary when the Jenkins master needs to run a build. When their work is done, they are automatically terminated and their resources are added back to the cluster's resource pool.

1. Retrieve the admin password that was automatically created by the Jenkins Helm chart:
2. To open the Jenkins user interface, click **Web Preview** in Cloud Shell and click **Preview on port 8080**.



3. Click **log in** on the top right of the window. Enter `admin` for the **User** field and the password value from the previous step for the **Password** field.
4. Click the **log in** button.

You now have access to Jenkins and a Kubernetes cluster managed by GKE. To take this solution further, you could use these components in your continuous delivery pipeline.

After you've finished the tutorial, clean up the resources you created on GCP so you won't be billed for them in the future.

The easiest way to eliminate billing is to delete the project that you created for the tutorial.

To delete the project:


! **Caution:** Deleting a project has the following effects:

- **Everything in the project is deleted.** If you used an existing project for this tutorial, when you delete it, you also delete any other work you've done in the project.
- **Custom project IDs are lost.** When you created this project, you might have created a custom project ID that you want to use in the future. To preserve the URLs that use the project ID, such as an `appspot.com` URL, delete selected resources inside the project instead of deleting the whole project.

If you plan to explore multiple tutorials and quickstarts, reusing projects can help you avoid exceeding project quota limits.

1. In the Cloud Console, go to the **Manage resources** page.

[Go to the Manage resources page](https://console.cloud.google.com/iam-admin/projects) (<https://console.cloud.google.com/iam-admin/projects>)

2. In the project list, select the project you want to delete and click **Delete** .

3. In the dialog, type the project ID, and then click **Shut down** to delete the project.

1. In the Cloud Console, go to the **Disks** page.

[Go to the Disks page](https://console.cloud.google.com/compute/disks) (<https://console.cloud.google.com/compute/disks>)

2. Click the checkbox for your `jenkins-home` disk.


3. Click **Delete**  to delete the disk.

To delete a Compute Engine instance:

1. In the Cloud Console, go to the **VM Instances** page.

[Go to the VM Instances page](https://console.cloud.google.com/compute/instances) (<https://console.cloud.google.com/compute/instances>)

2. Click the checkbox for the instance you want to delete.

3. Click **Delete**  to delete the instance.

- Learn more about [Jenkins on GKE best practices](/solutions/jenkins-on-kubernetes-engine/) (/solutions/jenkins-on-kubernetes-engine).
- Learn about how to [configure Jenkins for GKE](/solutions/configuring-jenkins-kubernetes-engine/) (/solutions/configuring-jenkins-kubernetes-engine).

- Learn about how to set up [continuous deployment to GKE using Jenkins](/solutions/continuous-delivery-jenkins-kubernetes-engine/) (/solutions/continuous-delivery-jenkins-kubernetes-engine).
- Try out other Google Cloud features for yourself. Have a look at our [tutorials](/docs/tutorials/) (/docs/tutorials).