# Abstract

The document aims to illustrate how to create an IAM service account on google cloud console, how to create a reusable Compute Engine image, how to install and configure Google Cloud Engine(GCE) plugin, and how to run a job(e.g., building a Redback frontend project) on an a Jenkins agent created automatically on google cloud.

# terminology:

an IAM service account is “a special kind of account used by an application or compute workload, such as a Compute Engine virtual machine (VM) instance, rather than a person” [1]. This account will be used to create a private key for Jenkins to access google cloud.

The Google Compute Engine (GCE) Plugin allows developers to use GCE virtual machines (VMs) with Jenkins to execute build tasks. GCE VMs provision quickly, are destroyed by Jenkins when idle, and therefore the VMs run at a much lower price than regular VMs [2].

Jenkins is an automation tool that helps in automating the various parts of the Software Development Life Cycle(SDLC) process [3]. The Google Compute Engine (GCE) Plugin is one of the numerous plugins Jenkins provide.

Packer is an open source tool for creating machine images. Here Packer is used to build a reusable Compute Engine image that contains the software and tools needed to run as a Jenkins executor [4].

# Steps:

## Create and configure an IAM service account

to create an IAM service account on the dashboard of google cloud, click “Activate Cloud Shell” on the top-right corner

Graphical user interface, text, application, chat or text message

Description automatically generated

Input “set project [projectID]” and enter:

set project sit-22t2-redback-infra-612f89e

input the following commands and enter:

gcloud iam service-accounts create jenkins --display-name jenkins

export SA\_EMAIL=$(gcloud iam service-accounts list \

--filter="displayName:jenkins" --format='value(email)')

export PROJECT=$(gcloud info --format='value(config.project)')

Bind the following roles to your service account. Remember that a ordinary account does not have such permissions to grant those roles to the account “jenkins”. Therefore, please ask the admin account owner(i.e., “scott.blackburn@deakin.edu.au”) to help execute those permissions:

gcloud projects add-iam-policy-binding $PROJECT \

--role roles/storage.admin --member serviceAccount:$SA\_EMAIL

gcloud projects add-iam-policy-binding $PROJECT --role roles/compute.instanceAdmin.v1 \

--member serviceAccount:$SA\_EMAIL

gcloud projects add-iam-policy-binding $PROJECT --role roles/compute.networkAdmin \

--member serviceAccount:$SA\_EMAIL

gcloud projects add-iam-policy-binding $PROJECT --role roles/compute.securityAdmin \

--member serviceAccount:$SA\_EMAIL

gcloud projects add-iam-policy-binding $PROJECT --role roles/iam.serviceAccountActor \

--member serviceAccount:$SA\_EMAIL

After successfully execute the above commands, execute the following command to create a private key used for Jenkins to get access to google cloud compute engine in the later steps:

gcloud iam service-accounts keys create jenkins-sa.json --iam-account $SA\_EMAIL

In Cloud Shell, click “More”, and then click Download file. Type jenkins-sa.json. Click Download to save the file locally.

## create a Jenkins agent image on google cloud console

create a SSH key pair on google console. If one already exists, this command uses that key pair; otherwise, it creates a new one:

ls ~/.ssh/id\_rsa.pub || ssh-keygen -N ""

Add the Cloud Shell public SSH key to your project's metadata:

gcloud compute project-info describe \

--format=json | jq -r '.commonInstanceMetadata.items[] | select(.key == "ssh-keys") | .value' > sshKeys.pub

echo "$USER:$(cat ~/.ssh/id\_rsa.pub)" >> sshKeys.pub

gcloud compute project-info add-metadata --metadata-from-file ssh-keys=sshKeys.pub

download Packer software

wget https://releases.hashicorp.com/packer/1.7.10/packer\_1.7.10\_linux\_amd64.zip

unzip packer\_1.7.10\_linux\_amd64.zip

Create the configuration file for your Packer image builds:

export PROJECT=$(gcloud info --format='value(config.project)')

cat > jenkins-agent.json <<EOF

{

"builders": [

{

"type": "googlecompute",

"project\_id": "$PROJECT",

"source\_image\_family": "ubuntu-2004-lts",

"source\_image\_project\_id": "ubuntu-os-cloud",

"zone": "us-central1-a",

"disk\_size": "50",

"image\_name": "jenkins-agent-{{timestamp}}",

"image\_family": "jenkins-agent",

"ssh\_username": "ubuntu"

}

],

"provisioners": [

{

"type": "shell",

"inline": ["sudo apt-get update && sudo apt-get install -y default-jdk"]

}

]

}

EOF

Build the image by running Packer:

./packer build jenkins-agent.json

If the following prompt appears, it means the image is built successfully:

==> Builds finished. The artifacts of successful builds are:

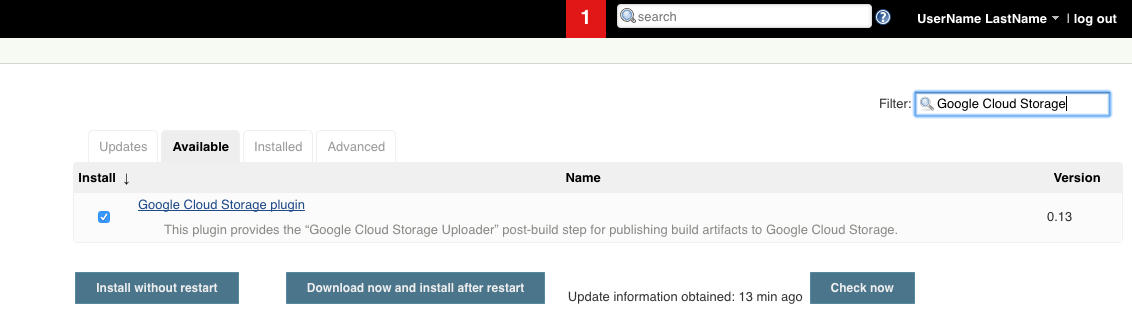
--> googlecompute: A disk image was created:

## install Google Cloud Engine(GCE) plugin

In this section, we can use Cloud Marketplace to provision a Jenkins instance [4], or we can use our locally installed Jenkins. Here we use the locally installed Jenkins. However, for productional use, I recommend using a custom Jenkins instance VM provisioned by Cloud Marketplace, which costs some money.

To install Google Cloud Engine(GCE) plugin. First, we log on the Jenkins admin website. Then, in the Jenkins dashboard UI, select “Manage Jenkins” -> ”Manage Plugins”, Click the Available tab and use the Filter bar to find the following plugins and select the boxes next to them:

Compute Engine plugin



Click “Download now and install after restart” and then click the Restart Jenkins when installation is complete and no jobs are running checkbox [4]. Jenkins will restarts and completes the plugin installations [4].

## Create plugin credentials

Log in to Jenkins again, and click **Manage Jenkins** -> **Manage Credentials** -> **(global)** -> **“add credentials”**:

* 1. Set **Kind** to **Google Service Account from private key**.
  2. In the **Project Name** field, enter Google Cloud project ID(i.e., sit-22t2-redback-infra-612f89e).
  3. Click **Choose file**.
  4. Select the jenkins-sa.json file which was previously downloaded from Cloud console Shell.
  5. Click **OK**.

## configure Google Cloud Engine(GCE) plugin

1. Click Manage Jenkins -> Manage Nodes and Clouds -> Configure Clouds
2. Click ***Add a new Cloud***, then click ***Google Compute Engine***.
3. Set the following settings and replace [YOUR\_PROJECT\_ID] with your Google Cloud project ID:

Name: [YOUR\_PROJECT\_ID]  
Project ID: [YOUR\_PROJECT\_ID]  
Instance Cap: 8

1. Choose the service account from the Service Account Credentials drop-down list. It is listed as your Google Cloud project ID [4].
2. click **Add** under **Instance Configurations**.
3. Enter the following General settings:

* **Name-prefix**: gce
* **Description**: Ubuntu agent
* **Labels**: ubuntu-2004

1. Enter the following for Location settings:

* Region: australia-southeast1
* Zone: australia-southeast1-b

1. Click **Advanced**.
2. For **Machine Configuration**, choose the **Machine Type** of **n1-standard-1**.
3. Under *Networking*, choose the following settings:

* **Network**: Leave at *default* setting.
* **Subnetwork**: Leave at *default* setting.
* Select **Attach External IP?**.

1. Select the following for Boot Disk settings:

* For **Image project**, choose your Google Cloud project.
* For **Image name**, select the image you built earlier using Packer.

1. Click Save to persist your configuration changes.

## configure and run a project

* 1. Click **+ New Item** in the Jenkins interface to create a new job.

1. Enter GCPtest as the item name.
2. Click **Freestyle project**, and then click **OK**.
3. Check the **Execute concurrent builds if necessary** and **Restrict where this project can run** boxes.
4. In the **Label Expression** field, enter ubuntu-2004.
5. In the **Build Steps** section, click **Add build step**.
6. Click **Execute Shell**.
7. In the command box, enter a test string:

echo "hostname:`hostname`"

1. Click **Save.**
2. Click **Build Now** to start a build.

If successful, the Console output should look like this below:

Graphical user interface, text, application, email

Description automatically generated

## troubleshoot

If building a project fails, and the prompt is like that:

A picture containing text

Description automatically generated

To troubleshoot, click “Manage Jenkins”->”System Log” to find relevant error logs. Here list one error log and its corresponding solution:

error log:  
solution:

go to “Manage nodes and clouds”->”Configure Clouds”->”Advanced…”, select default on the droplist of Network, and default on the droplist of Subnetwork.

# References:

[1] “Service accounts  |  IAM documentation  |  google cloud,” *Google*. [Online]. Available: https://cloud.google.com/iam/docs/service-accounts. [Accessed: 11-Dec-2022].

[2] *Google Compute engine*. [Online]. Available: https://plugins.jenkins.io/google-compute-engine/. [Accessed: 11-Dec-2022].

[3] Jagrat, “Jenkins build jobs - how to create and trigger build jobs in Jenkins?,” *TOOLSQA*, 11-Dec-2022. [Online]. Available: https://www.toolsqa.com/jenkins/jenkins-build-jobs/. [Accessed: 11-Dec-2022].

[4] “Using Jenkins for distributed builds on Compute engine  |  cloud architecture center  |  google cloud,” *Google*. [Online]. Available: https://cloud.google.com/architecture/using-jenkins-for-distributed-builds-on-compute-engine. [Accessed: 11-Dec-2022].