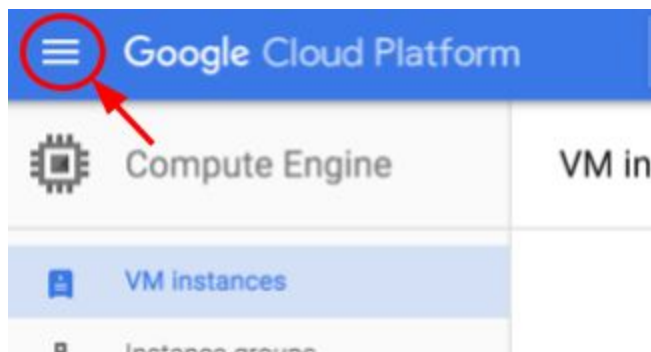


Task1:Create a new instance from the Cloud Console

.

In the Cloud Console, on the top left of the screen, select **Navigation menu** > **Compute Engine** > **VM Instances**:



This may take a minute to initialize for the first time.

To create a new instance, click **Create**.

Compute Engine VM instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. You can choose from micro-VMs to large instances running Debian, Windows, or other standard images. Create your first VM instance, import it by CloudEndure migration service or try the quickstart to build a sample app.

Create

or

Import

or

Take the quickstart

There are many parameters you can configure when creating a new instance. Use the following for this lab:

Field	Value	Additional Information
Name	nucleus-jumphost	Name for the VM instance
Region	us-central1 (Iowa)	Learn more about regions in Regions & Zones documentation .
Zone	us-central1-c Note: remember the zone that you selected, you'll need it later.	Learn more about zones in Regions & Zones documentation .
Series	N1/f1-micro	Name of the series

Machine Type	<p>2 vCPUs</p> <p>This is a (n1-standard-2), 2-CPU, 7.5GB RAM instance.</p> <p>There are a number of machine types, ranging from micro instance types to 32-core/208GB RAM instance types. Learn more in the Machine Types documentation.</p>	<p>Note: A new project has a default resource quota, which may limit the number of CPU cores. You can request more when you work on projects outside of this lab.</p>
Boot Disk	<p>New 10 GB standard persistent disk</p> <p>OS Image: Debian GNU/Linux 9 (stretch)</p>	<p>There are a number of images to choose from, including: Debian, Ubuntu, CoreOS as well as premium images such as Red Hat Enterprise Linux and Windows Server. See Operating System documentation for more detail.</p>
Firewall	<p>Check Allow HTTP traffic</p> <p>Check this option so to access a webserver that you'll install later.</p>	<p>Note: This will automatically create firewall rule to allow HTTP traffic on port 8</p>

Task 2: Kubernetes

Step1: `gcloud config set compute/zone us-east1-b`

Step 2: `gcloud container clusters create my-cluster1`

Step3: `gcloud container clusters get-credentials my-cluster1`

Step4: `kubectl create deployment hello-server`

`--image=gcr.io/google-samples/hello-app:2.0`

Step5: `kubectl expose deployment hello-server --type=LoadBalancer --port 8080`

Task3: Load balancer

step1:

```
cat << EOF > startup.sh
```

```
#!/bin/bash
```

```
apt-get update
```

```
apt-get install -y nginx
```

```
service nginx start
```

```
sed -i -- 's/nginx/Google Cloud Platform - '"$HOSTNAME"'/'
```

```
/var/www/html/index.nginx-debian.html
```

```
EOF
```

Step2 :

```
gcloud compute instance-templates create nginx-template \
```

```
--metadata-from-file startup-script=startup.sh
```

Step3:

```
gcloud compute target-pools create nginx-pool
```

Step4:

```
gcloud compute instance-groups managed create nginx-group \
```

```
--base-instance-name nginx \
```

```
--size 2 \
```

```
--template nginx-template \
```

```
--target-pool nginx-pool
```

Step5:

```
gcloud compute firewall-rules create www-firewall --allow tcp:80
```

Step6:

```
gcloud compute http-health-checks create http-basic-check
```

Step7:

```
gcloud compute backend-services create nginx-backend \
```

```
--protocol HTTP --http-health-checks http-basic-check --global
```

Step8:

```
gcloud compute url-maps create web-map \  
--default-service nginx-backend
```

Step9:

```
gcloud compute forwarding-rules create http-content-rule \  
--global \  
--target-http-proxy http-lb-proxy \  
--ports 80
```