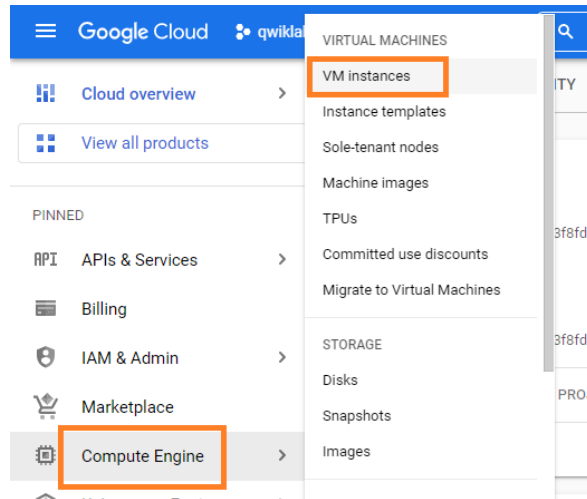


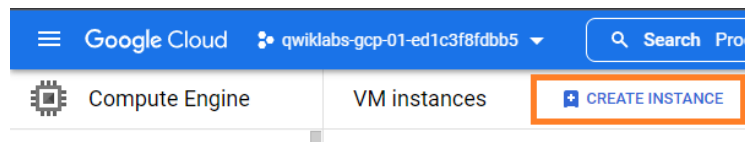
Tarefa 1: crie uma instância para o projeto jumphost

Acessar o console e ir ao painel de serviços do lado esquerdo da tela:

Vá em “Compute engine” depois clique em “VM instances”.

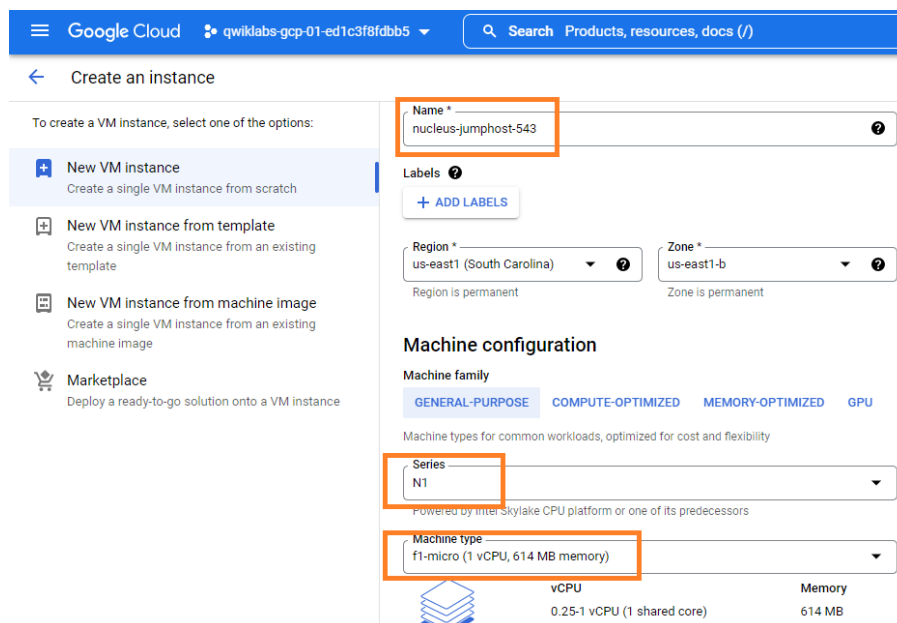


Clicar no botão “Create instance”




Informe o nome da máquina conforme indicado no seu laboratório.

Escolha o tipo da máquina como foi especificada na sua tarefa.



Confira se a imagem de Boot está como Debian.

Boot disk ⓘ

Name	nucleus-jumphost-543
Type	New balanced persistent disk
Size	10 GB
License type ⓘ	Free
Image	 Debian GNU/Linux 11 (bullseye)

[CHANGE](#)

Clique em “Create” ao final da página.

Aguarde a máquina virtual ser criada e depois volte na página do laboratório para validar a tarefa.

Clique em *Verificar meu progresso* para ver o objetivo.



Crie uma instância para o projeto jumphost

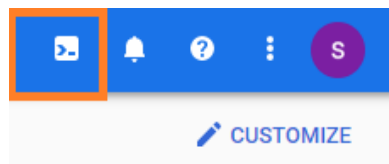
[Verificar meu progresso](#)

Assessment Completed!

Tarefa 1 pronta!

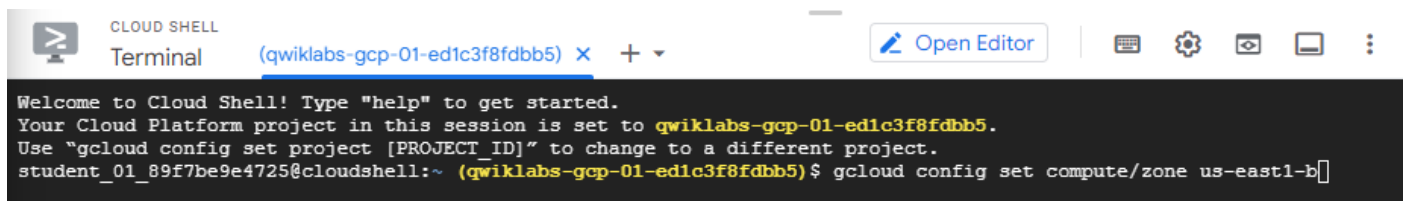
Tarefa 2: crie um cluster de serviço do Kubernetes

Na barra superior, clique no ícone do “Cloud Shell”



Aguarde abrir o Cloud Shell e digite o seguinte comando **alterando a zona para a que foi indicada no seu laboratório** e depois aperte ENTER:

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

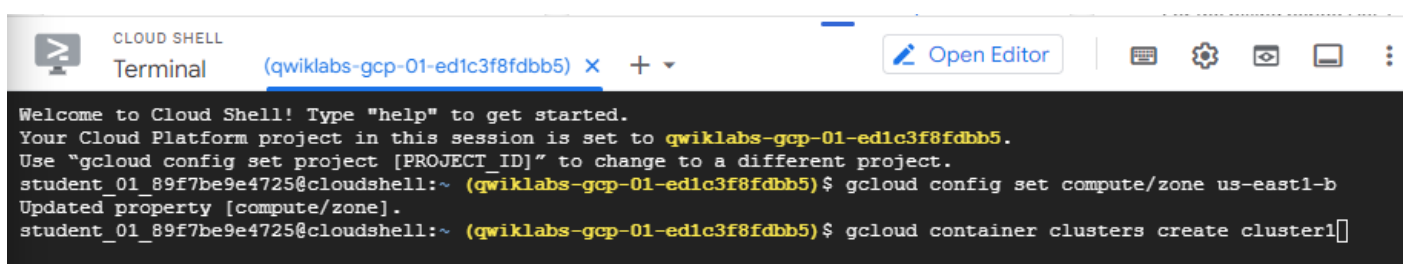


```
CLOUD SHELL
Terminal (qwiklabs-gcp-01-ed1c3f8fdbb5) x + v [Open Editor]

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to quwiklabs-gcp-01-ed1c3f8fdbb5.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ gcloud config set compute/zone us-east1-b
```

Na sequência, digite o seguinte comando para criar o cluster, depois aperte ENTER e aguarde.

```
gcloud container clusters create cluster1
```



```
CLOUD SHELL
Terminal (qwiklabs-gcp-01-ed1c3f8fdbb5) x + v [Open Editor]

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to quwiklabs-gcp-01-ed1c3f8fdbb5.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ gcloud config set compute/zone us-east1-b
Updated property [compute/zone].
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ gcloud container clusters create cluster1
```

Depois, digite o seguinte comando para continuar

```
gcloud container clusters get-credentials cluster1
```

```
CLOUD SHELL
Terminal (qwiklabs-gcp-01-ed1c3f8fdbb5) x + v [Open Editor]

Creating cluster cluster1 in us-east1-b... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/qwiklabs-gcp-01-ed1c3f8fdbb5/zones/us-east1-b/clusters/cluster1].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload/_gcloud/us-east1-b/clu
ect=qwiklabs-gcp-01-ed1c3f8fdbb5
kubeconfig entry generated for cluster1.
NAME: cluster1
LOCATION: us-east1-b
MASTER VERSION: 1.22.12-gke.2300
MASTER IP: 35.231.110.145
MACHINE TYPE: e2-medium
NODE VERSION: 1.22.12-gke.2300
NUM NODES: 3
STATUS: RUNNING
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ gcloud container clusters get-credentials cluster1
```

O próximo comando irá criar um aplicativo no cluster: (confira com o nome e imagem do seu laboratório)

```
kubectl create deployment hello-server --image=gcr.io/google-samples/hello-app:2.0
```

```
CLOUD SHELL
Terminal (qwiklabs-gcp-01-ed1c3f8fdbb5) x + v [Open Editor]

kubeconfig entry generated for cluster1.
NAME: cluster1
LOCATION: us-east1-b
MASTER VERSION: 1.22.12-gke.2300
MASTER IP: 35.231.110.145
MACHINE TYPE: e2-medium
NODE VERSION: 1.22.12-gke.2300
NUM NODES: 3
STATUS: RUNNING
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ gcloud container clusters get-credentials cluster1
Fetching cluster endpoint and auth data.
kubeconfig entry generated for cluster1.
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ kubectl create deployment hello-server --image=gcr.io/google-sam
ples/helloapp:2.0
```

Verifique qual o número da porta indicada no seu laboratório e substitua no final do comando abaixo


```
kubectl expose deployment hello-server --type=LoadBalancer --port 8083
```

```
CLOUD SHELL
Terminal (qwiklabs-gcp-01-ed1c3f8fdbb5) x + v [Open Editor]

NODE VERSION: 1.22.12-gke.2300
NUM NODES: 3
STATUS: RUNNING
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ gcloud container clusters get-credentials cluster1
Fetching cluster endpoint and auth data.
kubeconfig entry generated for cluster1.
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ kubectl create deployment hello-server --image=gcr.io/google-sam
ples/helloapp:2.0
deployment.apps/hello-server created
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5) $ kubectl expose deployment hello-server --type=LoadBalancer --por
t 8080
```

Valide a tarefa 2.

Clique em *Verificar meu progresso* para ver o objetivo.



Crie um cluster do Kubernetes

Verificar meu progresso

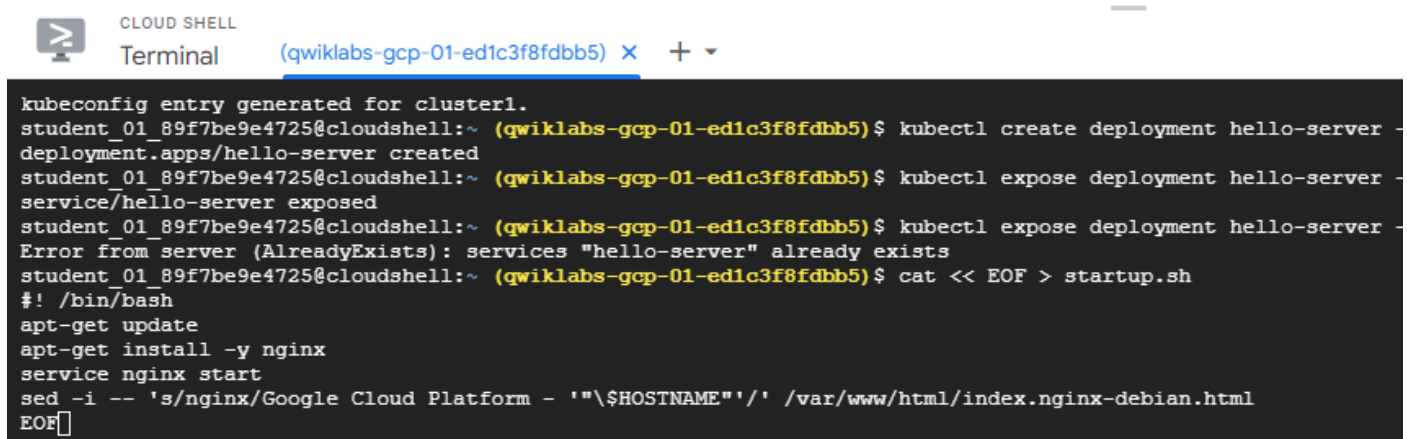
Assessment Completed!

Tarefa 2 pronta!

Tarefa 3: configure um balanceador de carga HTTP

Execute o seguinte comando apresentado em seu laboratório:

```
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
```



The screenshot shows a Cloud Shell terminal window with the title "Terminal (qwiklabs-gcp-01-ed1c3f8fdbb5)". The terminal output shows the following commands and their results:

```
kubeconfig entry generated for cluster1.
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5)$ kubectl create deployment hello-server
deployment.apps/hello-server created
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5)$ kubectl expose deployment hello-server
service/hello-server exposed
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5)$ kubectl expose deployment hello-server
Error from server (AlreadyExists): services "hello-server" already exists
student_01_89f7be9e4725@cloudshell:~ (qwiklabs-gcp-01-ed1c3f8fdbb5)$ 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
```

Próximo item será criar um modelo de instância com o seguinte comando

```
gcloud compute instance-templates create web-server-template \
  --metadata-from-file startup-script=startup.sh \
  --network default \
  --machine-type g1-small \
  --region us-east1
```

Depois crie um pool de destino.

```
gcloud compute instance-groups managed create web-server-group \
  --base-instance-name web-server \
  --size 2 \
  --template web-server-template \
  --region us-east1
```

Feito isso, defina uma regra de firewall [\[com o nome do seu laboratório\]](#) para permitir a porta 80.

```
gcloud compute firewall-rules create \[mudar com o nome do seu laboratório\] \
  --allow tcp:80 \
  --network default
```

Crie uma verificação de integridade

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

Depois crie um grupo de instâncias gerenciadas.

```
gcloud compute instance-groups managed \
  set-named-ports web-server-group \
  --named-ports http:80 \
  --region us-east1
```

Para criar um serviço de back-end e anexar o grupo gerenciado de instâncias à porta chamada (http:80) são dois comandos

```
gcloud compute backend-services create web-server-backend \
  --protocol HTTP \
  --http-health-checks http-basic-check \
  --global
```

Depois, o segundo comando:

```
gcloud compute backend-services add-backend web-server-backend \
  --instance-group web-server-group \
  --instance-group-region us-east1 \
  --global
```

Na sequência crie um mapa de URL e direcione para ele o encaminhamento de solicitações do proxy HTTP

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

Agora crie uma regra de encaminhamento, são três comandos. Primeiro:

```
gcloud compute target-http-proxies create http-lb-proxy \
  --url-map web-server-map
```

Segundo:

```
gcloud compute forwarding-rules create [mudar com o nome do seu laboratório] \
  --global \
  --target-http-proxy http-lb-proxy \
  --ports 80
```

Terceiro

```
gcloud compute forwarding-rules list
```

Aguarde a criação do balanceamento por volta de 7min antes de verificar se a atividade está concluída

Depois de aguardar, volte para validar a tarefa 3.

Clique em *Verificar meu progresso* para ver o objetivo.



Crie o site por trás do balanceador de carga HTTP

[Verificar meu progresso](#)

Assessment Completed!