**Task 1: Set the default region and zone for all resources**

1. In Cloud Shell, set the default zone:

gcloud config set compute/zone us-central1-a

Copied!

content\_copy

1. Set the default region:

gcloud config set compute/region us-central1

Copied!

content\_copy

Learn more about choosing zones and regions here: [Regions and Zones documentation](https://cloud.google.com/compute/docs/zones)

**Task 2: Create multiple web server instances**

For this load balancing scenario, create three Compute Engine VM instances and install Apache on them, then add a firewall rule that allows HTTP traffic to reach the instances.

1. Create three new virtual machines in your default zone and give them all the same tag. The code provided sets the zone to us-central1-a. Setting the tags field lets you reference these instances all at once, such as with a firewall rule. These commands also install Apache on each instance and give each instance a unique home page.

gcloud compute instances create www1 \

--image-family debian-9 \

--image-project debian-cloud \

--zone us-central1-a \

--tags network-lb-tag \

--metadata startup-script="#! /bin/bash

sudo apt-get update

sudo apt-get install apache2 -y

sudo service apache2 restart

echo '<!doctype html><html><body><h1>www1</h1></body></html>' | tee /var/www/html/index.html"

Copied!

content\_copy

gcloud compute instances create www2 \

--image-family debian-9 \

--image-project debian-cloud \

--zone us-central1-a \

--tags network-lb-tag \

--metadata startup-script="#! /bin/bash

sudo apt-get update

sudo apt-get install apache2 -y

sudo service apache2 restart

echo '<!doctype html><html><body><h1>www2</h1></body></html>' | tee /var/www/html/index.html"

Copied!

content\_copy

gcloud compute instances create www3 \

--image-family debian-9 \

--image-project debian-cloud \

--zone us-central1-a \

--tags network-lb-tag \

--metadata startup-script="#! /bin/bash

sudo apt-get update

sudo apt-get install apache2 -y

sudo service apache2 restart

echo '<!doctype html><html><body><h1>www3</h1></body></html>' | tee /var/www/html/index.html"

Copied!

content\_copy

1. Create a firewall rule to allow external traffic to the VM instances:

gcloud compute firewall-rules create grant-tcp-rule-522 \

--target-tags network-lb-tag --allow tcp:80

Copied!

content\_copy

Now you need to get the external IP addresses of your instances and verify that they are running.

1. Run the following to list your instances. You'll see their IP addresses in the EXTERNAL\_IP column:

gcloud compute instances list

Copied!

content\_copy

1. Verify that each instance is running with curl, replacing **[IP\_ADDRESS]** with the IP address for each of your VMs:

curl http://[IP\_ADDRESS]

Copied!

content\_copy

Check your lab progress

Click **Check my progress** below to verify that you've created a group of web servers.

Create multiple web server instances

Check my progress

**Task 3: Configure the load balancing service**

When you configure the load balancing service, your virtual machine instances will receive packets that are destined for the static external IP address you configure. Instances made with a Compute Engine image are automatically configured to handle this IP address.

For more information, see [Setting Up Network Load Balancing](https://cloud.google.com/compute/docs/load-balancing/network/).

1. Create a static external IP address for your load balancer:
2. gcloud compute addresses create network-lb-ip-1 \

--region us-east1

Copied!

content\_copy

(Output)

Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-03-xxxxxxxxxxx/regions/us-central1/addresses/network-lb-ip-1].

1. Add a legacy HTTP health check resource:

gcloud compute http-health-checks create basic-check

Copied!

content\_copy

1. Add a target pool in the same region as your instances. Run the following to create the target pool and use the health check, which is required for the service to function:

gcloud compute target-pools create www-pool \

--region us-east1 --http-health-check basic-check

Copied!

content\_copy

1. Add the instances to the pool:

gcloud compute target-pools add-instances www-pool \

--instances www1,www2,www3

Copied!

content\_copy

1. Add a forwarding rule:

gcloud compute forwarding-rules create www-rule \

--region us-east1 \

--ports 80 \

--address network-lb-ip-1 \

--target-pool www-pool

Copied!

content\_copy

Check your lab progress

Click **Check my progress** below to verify that you've created an L4 network load balancer that points to the web servers.

Configure the load balancing service

Check my progress

**Task 4: Sending traffic to your instances**

Now that the load balancing service is configured, you can start sending traffic to the forwarding rule and watch the traffic be dispersed to different instances.

Enter the following command to view the external IP address of the www-rule forwarding rule used by the load balancer:

gcloud compute forwarding-rules describe www-rule --region us-east1

Copied!

content\_copy

Use curl command to access the external IP address, replacing IP\_ADDRESS with an external IP address from the previous command:

while true; do curl -m1 IP\_ADDRESS; done

Copied!

content\_copy

The response from the curl command alternates randomly among the three instances. if your response is initially unsuccessful, wait approximately 30 seconds for the configuration to be fully loaded and for your instances to be marked healthy before trying again.

Use **Ctrl** + **c** to stop running the command.

**Task 5: Create an HTTP load balancer**

HTTP(S) Load Balancing is implemented on Google Front End (GFE). GFEs are distributed globally and operate together using Google's global network and control plane. You can configure URL rules to route some URLs to one set of instances and route other URLs to other instances. Requests are always routed to the instance group that is closest to the user, if that group has enough capacity and is appropriate for the request. If the closest group does not have enough capacity, the request is sent to the closest group that *does* have capacity.

To set up a load balancer with a Compute Engine backend, your VMs need to be in an instance group. The managed instance group provides VMs running the backend servers of an external HTTP load balancer. For this lab, backends serve their own hostnames.

1. First, create the load balancer template:

gcloud compute instance-templates create lb-backend-template \

--region=us-east1 \

--network=default \

--subnet=default \

--tags=allow-health-check \

--image-family=debian-9 \

--image-project=debian-cloud \

--metadata=startup-script='#! /bin/bash

apt-get update

apt-get install apache2 -y

a2ensite default-ssl

a2enmod ssl

vm\_hostname="$(curl -H "Metadata-Flavor:Google" \

http://169.254.169.254/computeMetadata/v1/instance/name)"

echo "Page served from: $vm\_hostname" | \

tee /var/www/html/index.html

systemctl restart apache2'

Copied!

content\_copy

1. Create a managed instance group based on the template:

gcloud compute instance-groups managed create lb-backend-group \

--template=lb-backend-template --size=2 --zone=us-central1-a

Copied!

content\_copy

1. Create the fw-allow-health-check firewall rule. This is an ingress rule that allows traffic from the Google Cloud health checking systems (130.211.0.0/22 and 35.191.0.0/16). This lab uses the target tag allow-health-check to identify the VMs.

gcloud compute firewall-rules create fw-allow-health-check \

--network=default \

--action=allow \

--direction=ingress \

--source-ranges=130.211.0.0/22,35.191.0.0/16 \

--target-tags=allow-health-check \

--rules=tcp:80

Copied!

content\_copy

1. Now that the instances are up and running, set up a global static external IP address that your customers use to reach your load balancer.

gcloud compute addresses create lb-ipv4-1 \

--ip-version=IPV4 \

--global

Copied!

content\_copy

Note the IPv4 address that was reserved:

gcloud compute addresses describe lb-ipv4-1 \

--format="get(address)" \

--global

Copied!

content\_copy

1. Create a healthcheck for the load balancer:

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

--port 80

Copied!

content\_copy

1. Create a backend service:

gcloud compute backend-services create web-backend-service \

--protocol=HTTP \

--port-name=http \

--health-checks=http-basic-check \

--global

Copied!

content\_copy

1. Add your instance group as the backend to the backend service:

gcloud compute backend-services add-backend web-backend-service \

--instance-group=lb-backend-group \

--instance-group-zone=us-central1-a \

--global

Copied!

content\_copy

1. Create a URL map to route the incoming requests to the default backend service:

gcloud compute url-maps create web-map-http \

--default-service web-backend-service

Copied!

content\_copy

1. Create a target HTTP proxy to route requests to your URL map:

gcloud compute target-http-proxies create http-lb-proxy \

--url-map web-map-http

Copied!

content\_copy

1. Create a global forwarding rule to route incoming requests to the proxy:

gcloud compute forwarding-rules create http-content-rule \

--address=lb-ipv4-1\

--global \

--target-http-proxy=http-lb-proxy \

--ports=80

Copied!

content\_copy

Check your lab progress

Click **Check my progress** below to verify that you've created an L7 HTTP(S) load balancer.

Create an HTTP load balancer

Check my progress

**Task 6: Testing traffic sent to your instances**

1. In the Cloud Console, from the **Navigation menu**, go to **Network services** > **Load balancing**.
2. Click on the load balancer that you just created (web-map-http).
3. In the **Backend** section, click on the name of the backend and confirm that the VMs are **Healthy**. If they are not healthy, wait a few moments and try reloading the page.
4. When the VMs are healthy, test the load balancer using a web browser, going to http://IP\_ADDRESS/, replacing IP\_ADDRESS with the load balancer's IP address.

This may take three to five minutes. If you do not connect, wait a minute, and then reload the browser.

Your browser should render a page with content showing the name of the instance that served the page, along with its zone (for example, Page served from: lb-backend-group-xxxx).

**Congratulations!**