Create an Internal Load Balancer

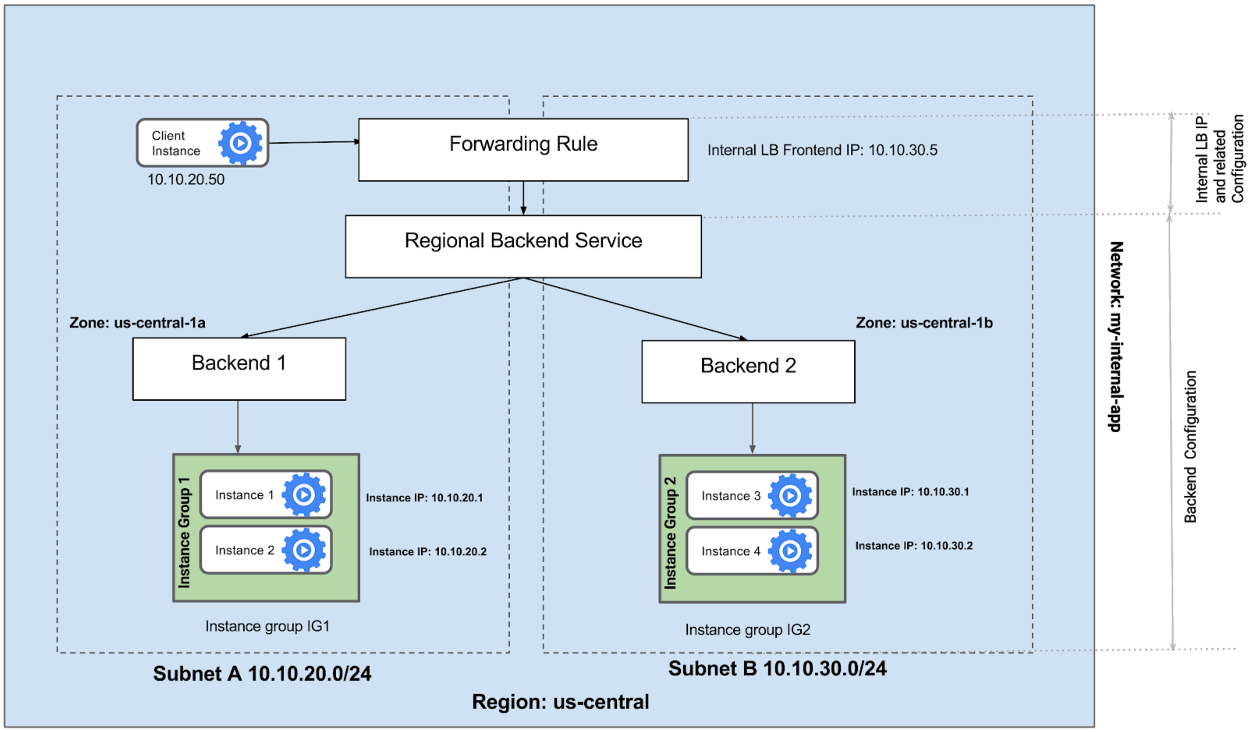
50 minutes7 Credits
Rate Lab

GSP216

Overview

Google Cloud offers Internal Load Balancing for your TCP/UDP-based traffic. Internal Load Balancing enables you to run and scale your services behind a private load balancing IP address that is accessible only to your internal virtual machine instances.

In this lab, you create two managed instance groups in the same region. Then, you configure and test an Internal Load Balancer with the instances groups as the backends, as shown in this network diagram:



Objectives

In this lab you learn how to perform the following tasks:

- Create HTTP and health check firewall rules
- Configure two instance templates
- Create two managed instance groups
- Configure and test an internal load balancer

Setup and requirements

Before you click the Start Lab button
Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need
To complete this lab, you need:

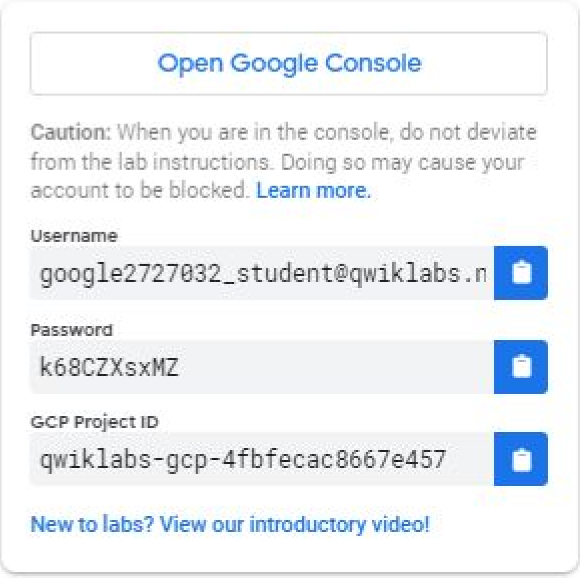
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

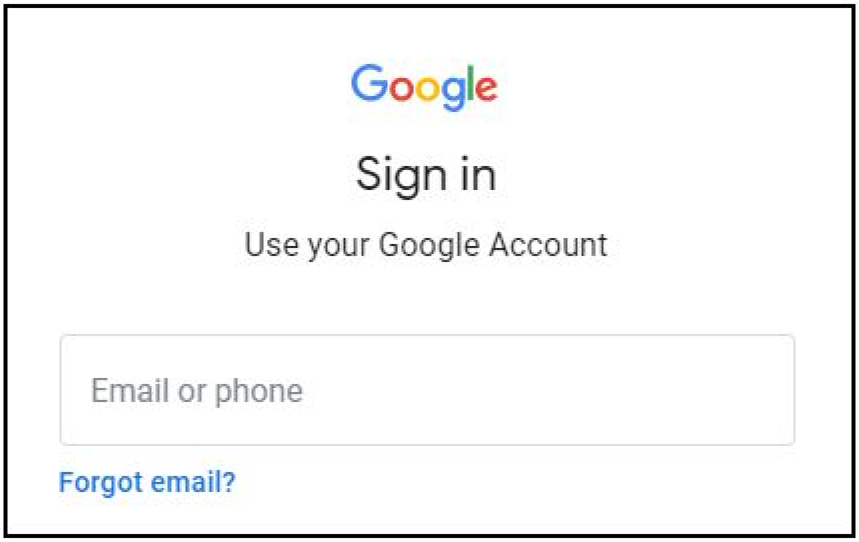
Username
google2727032_student@qwiklabs.n

Password
k68CZXsxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

New to labs? View our introductory video!

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



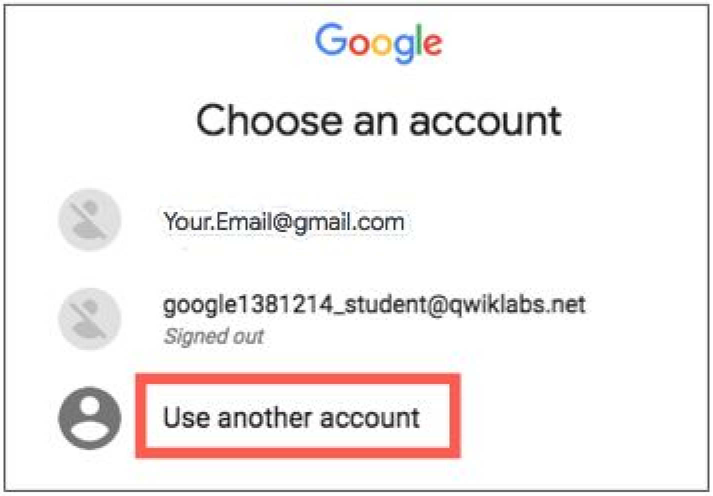
Google

Sign in

Use your Google Account

Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another Account**.



Google

Choose an account

Your.Email@gmail.com

google1381214_student@qwiklabs.net
Signed out

3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

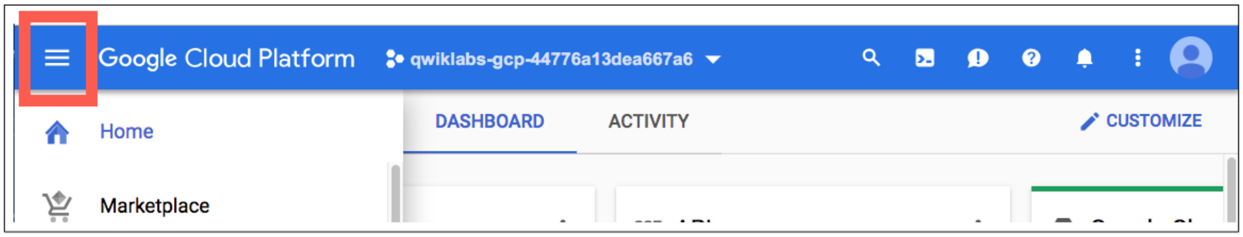
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. Click through the subsequent pages:
   - Accept the terms and conditions.

  - Do not add recovery options or two-factor authentication (because this is a temporary account).
  - Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



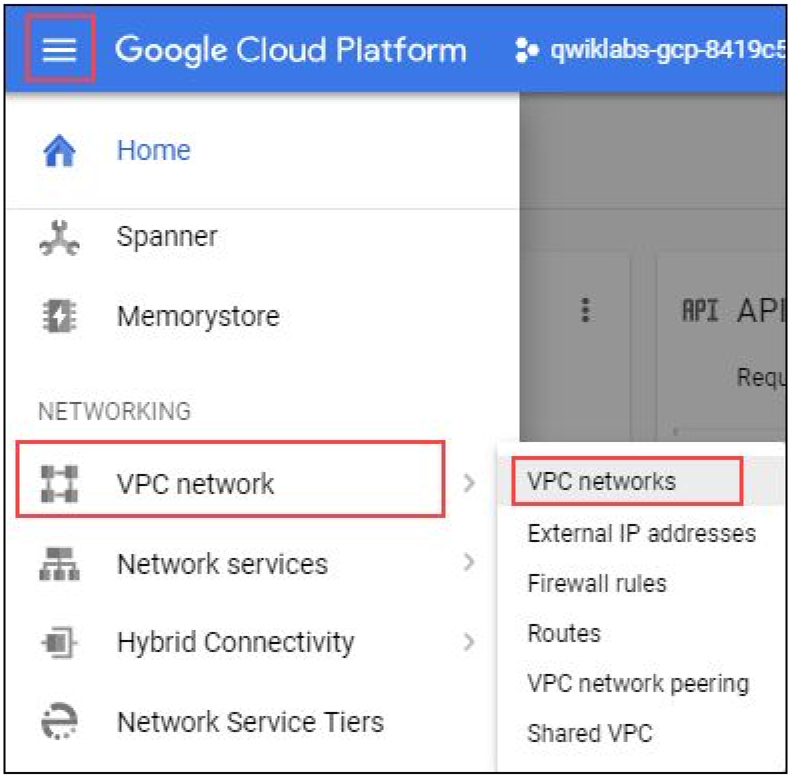
Configure HTTP and health check firewall rules

Configure firewall rules to allow HTTP traffic to the backends and TCP traffic from the Google Cloud health checker.

Explore the my-internal-app network

The network `my-internal-app` with subnet-a and subnet-b along with firewall rules for RDP, SSH, and ICMP traffic have been configured for you.

1. In the Console, navigate to **Navigation menu** > **VPC network** > **VPC networks**.



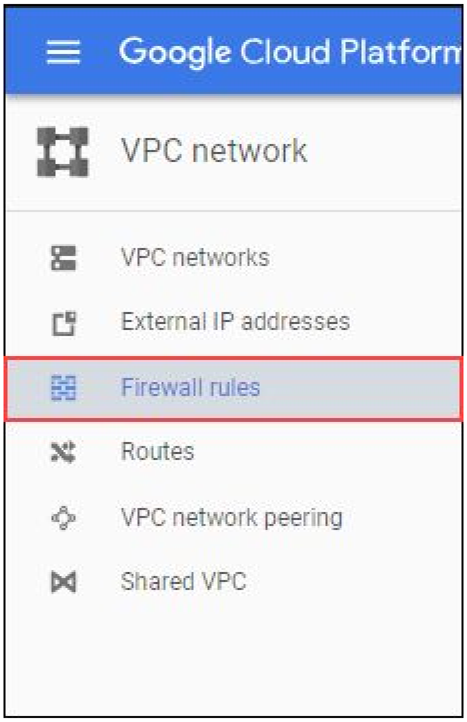
2. Notice the **my-internal-app** network with its subnets: **subnet-a** and **subnet-b**
   Each Google Cloud project starts with the **default** network. In addition, the **my-internal-app** network has been created for you, as part of your network diagram.
   You will create the managed instance groups in **subnet-a** and **subnet-b**.
   Both subnets are in the **us-central1** region because an Internal Load

Balancer is a regional service. The managed instance groups will be in different zones, making your service immune to zonal failures.

Create the HTTP firewall rule

Create a firewall rule to allow HTTP traffic to the backends from the Load Balancer and the internet (to install Apache on the backends).

1. Still in **VPC network**, in the left pane click **Firewall rules**.



2. Notice the **app-allow-icmp** and **app-allow-ssh-rdp** firewall rules. These firewall rules have been created for you.
3. Click **Create Firewall Rule**.

4. Set the following values, leave all other values at their defaults:

Property	Value (type value or select option as specified)
Name	app-allow-http
Network	my-internal-app
Targets	Specified target tags
Target tags	lb-backend
Source filter	IP Ranges
Source IP ranges	0.0.0.0/0
Protocols and ports	Specified protocols and ports, and then *check* tcp, *type:* 80

Make sure to include the **/0** in the **Source IP ranges** to specify all networks.

5. Click **Create**.

Create the health check firewall rules

Health checks determine which instances of a Load Balancer can receive new connections. For Internal load balancing, the health check probes to your load balanced instances come from addresses in the ranges `130.211.0.0/22` and `35.191.0.0/16`. Your firewall rules must allow these connections.

1. Still in the **Firewall rules** page, click **Create Firewall Rule**.
2. Set the following values, leave all other values at their defaults:

Property	Value (type value or select option as specified)
Name	app-allow-health-check
Targets	Specified target tags
Target tags	lb-backend

Source filter	IP Ranges
Source IP ranges	130.211.0.0/22 35.191.0.0/16
Protocols and ports	Specified protocols and ports, and then *check* tcp

Make sure to enter the two **Source IP ranges** one-by-one and pressing SPACE in between them.

3. Click **Create**.

Click Check my progress to verify the objective.

Configure HTTP and health check firewall rules

Check my progress

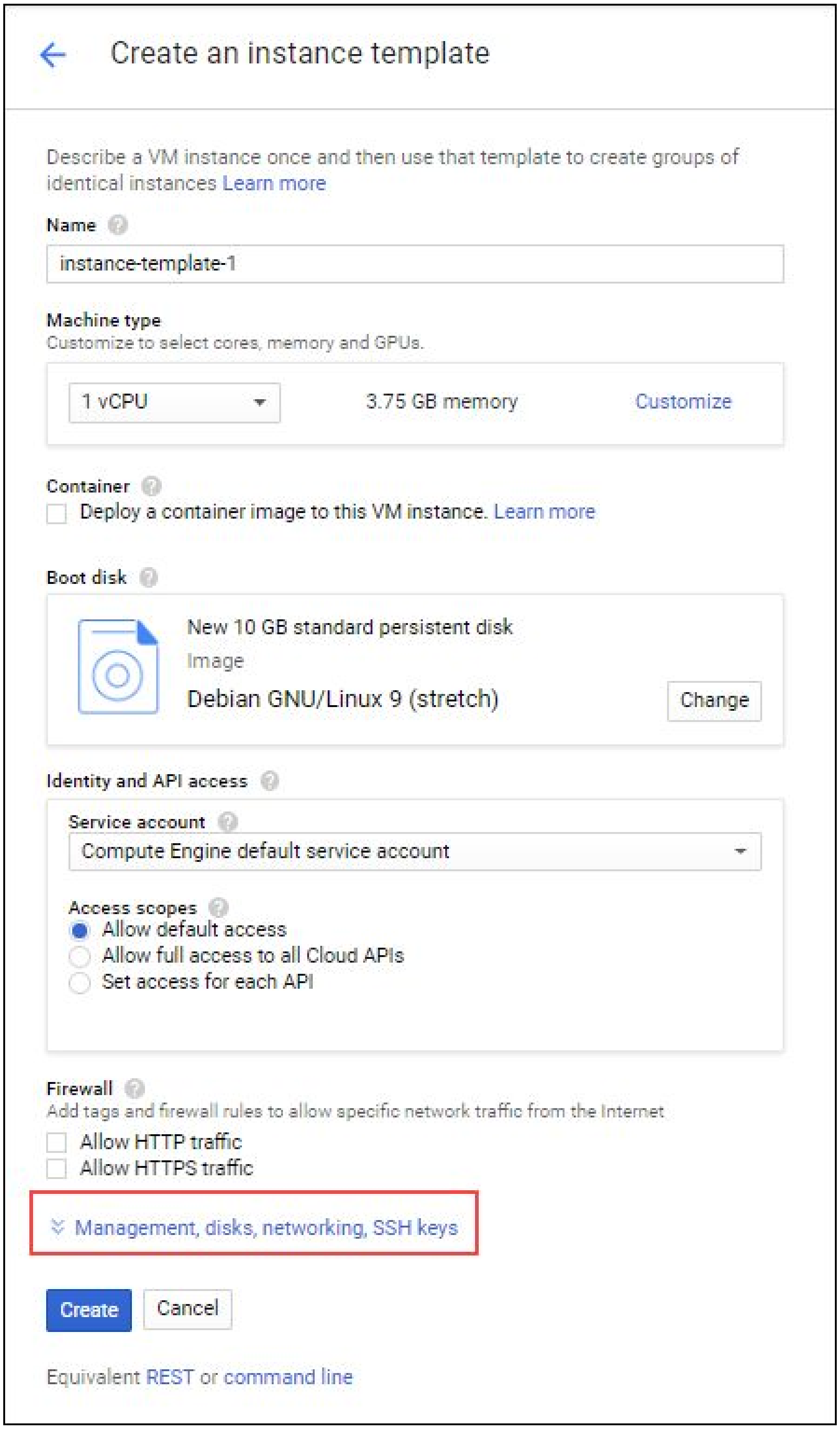
Configure instance templates and create instance groups

A managed instance group uses an instance template to create a group of identical instances. Use these to create the backends of the Internal Load Balancer.

Configure the instance templates

An instance template is an API resource that you can use to create VM instances and managed instance groups. Instance templates define the machine type, boot disk image, subnet, labels, and other instance properties. Create an instance template for both subnets of the **my-internal-app** network.

1. In the Console, navigate to **Navigation menu** > **Compute Engine** > **Instance templates**.
2. Click **Create instance template**.
3. For **Name**, type **instance-template-1**.
4. Click **Management, disks, networking, SSH keys**.



← Create an instance template

Describe a VM instance once and then use that template to create groups of identical instances Learn more

Name
instance-template-1

Machine type
Customize to select cores, memory and GPUs.
1 vCPU ▾ 3.75 GB memory Customize

Container
□ Deploy a container image to this VM instance. Learn more

Boot disk
New 10 GB standard persistent disk
Image
Debian GNU/Linux 9 (stretch) Change

Identity and API access
Service account
Compute Engine default service account ▾

Access scopes
● Allow default access
○ Allow full access to all Cloud APIs
○ Set access for each API

Firewall
Add tags and firewall rules to allow specific network traffic from the Internet
□ Allow HTTP traffic
□ Allow HTTPS traffic

Management, disks, networking, SSH keys

Create Cancel

Equivalent REST or command line

5. Click **Management**.

6. Under **Metadata**, specify the following:

Key	Value
startup-script-url	gs://cloud-training/gcpnet/ilb/startup.sh

The **startup-script-url** specifies a script that will be executed when instances are started. This script installs Apache and changes the welcome page to include the client IP and the name, region and zone of the VM instance. Feel free to explore this script here.

7. Click **Networking**.

8. For **Network interfaces**, set the following values, leave all other values at their defaults:

Property	Value (type value or select option as specified)
Network	my-internal-app
Subnetwork	subnet-a
Network tags	lb-backend

The network tag **lb-backend** ensures that the **HTTP** and **Health Check** firewall rules apply to these instances.

9. Click **Create**.
10. Wait for the instance template to be created.

Configure the next instance template

Create another instance template for **subnet-b** by copying **instance-template-1**:

1. Still in **Instance templates**, select **instance-template-1** and click **Copy**.
2. Click **Management, disks, networking, SSH keys**.
3. Click **Networking**.

4. For **Network interfaces**, select **subnet-b** as the **Subnetwork**.
5. Click **Create**.

Create the managed instance groups

Create a managed instance group in **subnet-a** (us-central1-a) and one **subnet-b** (us-central1-b).

1. Still in **Compute Engine**, in the left pane click **Instance groups**, and then click **Create Instance group**.
2. Set the following values, leave all other values at their defaults:

Property	**Value (type value or select option as specified)**
Name	instance-group-1
Location	Single-zone
Region	us-central1
Zone	us-central1-a
Group type	Managed instance group
Instance template	instance-template-1
Autoscale based on	CPU usage
Target CPU usage	80
Minimum number of instances	1
Maximum number of instances	5
Cool-down period	45

Managed instance groups offer **autoscaling** capabilities that allow you to automatically add or remove instances from a managed instance group based on increases or decreases in load. Autoscaling helps your applications gracefully handle increases in traffic and reduces cost when the need for resources is lower. You just define the autoscaling policy and the autoscaler performs automatic scaling based on the measured load.

3. Click **Create**.

   Repeat the same procedure for **instance-group-2** in **us-central1-b**:

4. Click **Create Instance group**.

5. Set the following values, leave all other values at their defaults:

Property	Value (type value or select option as specified)
Name	instance-group-2
Location	Single-zone
Region	us-central1
Zone	us-central1-b
Group type	Managed instance group
Instance template	instance-template-2
Autoscale based on	CPU usage
Target CPU usage	80
Minimum number of instances	1
Maximum number of instances	5
Cool-down period	45

6.

   Click **Create**.

Verify the backends

Verify that VM instances are being created in both subnets and create a utility VM to access the backends' HTTP sites.

1. Still in **Compute Engine**, click **VM instances**.
2. Notice two instances that start with `instance-group-1` and `instance-group-2`.

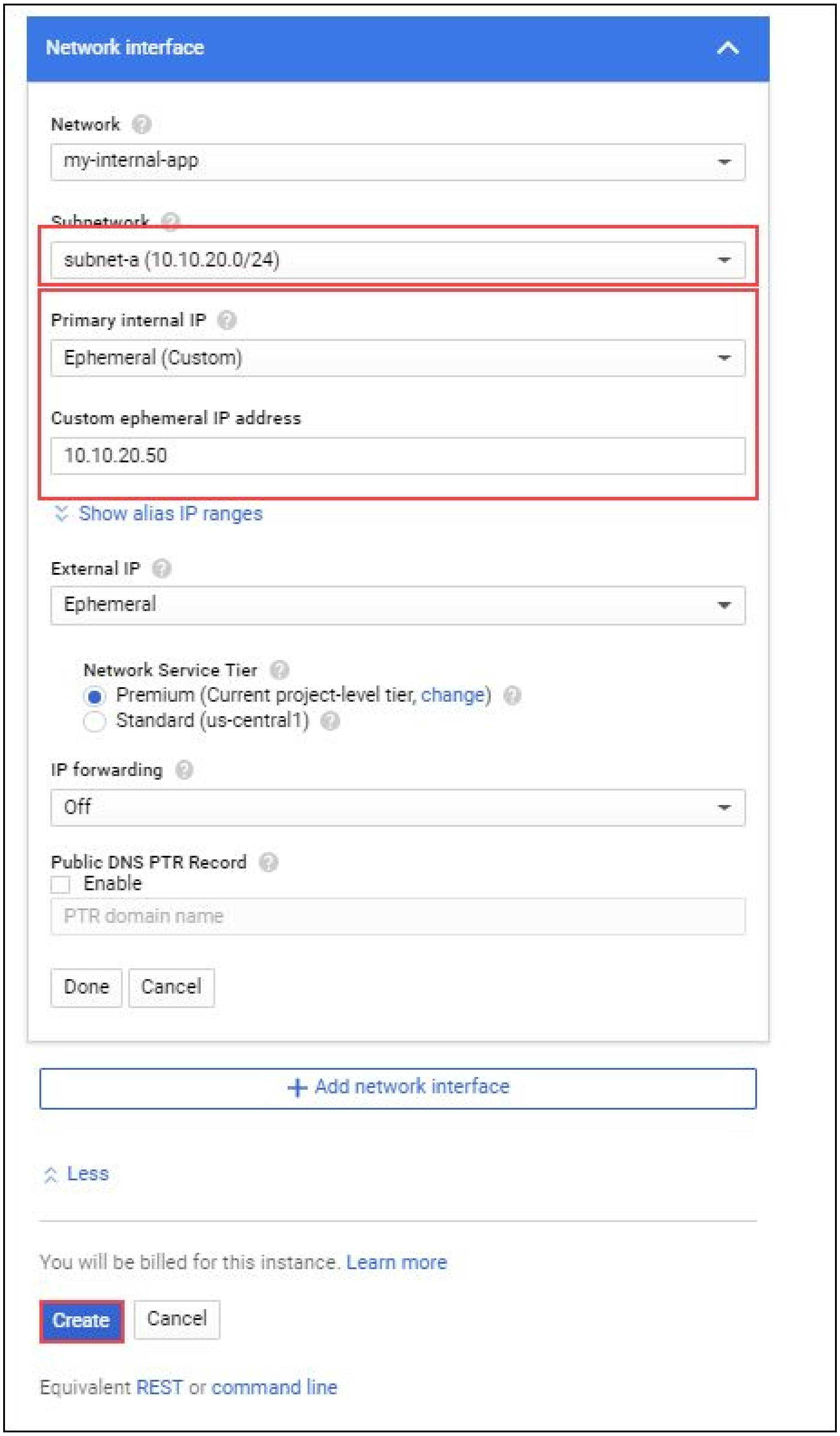
   These instances are in separate zones and their internal IP addresses are part of the **subnet-a** and **subnet-b** CIDR blocks.
3. Click **Create an instance**.
4. Set the following values, leave all other values at their defaults:

Property	Value (type value or select option as specified)
Name	utility-vm
Region	us-central1
Zone	us-central1-f
Machine type	micro (1 shared vCPU)

5. Click **Management, disks, networking, SSH keys**.
6. Click **Networking**.
7. For **Network interfaces**, click the pencil icon to edit.
8. Set the following values, leave all other values at their defaults:

Property	Value (type value or select option as specified)
Network	my-internal-app
Subnetwork	subnet-a
Primary internal IP	Ephemeral (Custom)
Custom ephemeral IP address	10.10.20.50

9. Click **Create**.



Network interface

Network: my-internal-app

Subnetwork: subnet-a (10.10.20.0/24)

Primary internal IP: Ephemeral (Custom)

Custom ephemeral IP address: 10.10.20.50

External IP: Ephemeral

Network Service Tier
● Premium (Current project-level tier, change)
○ Standard (us-central1)

IP forwarding: Off

Public DNS PTR Record
□ Enable
PTR domain name

You will be billed for this instance. Learn more

Equivalent REST or command line

Click Check my progress to verify the objective.

Configure instance templates and create instance groups

Check my progress

10. Note that the internal IP addresses for the backends are `10.10.20.2` and `10.10.30.2`.

If these IP addresses are different, replace them in the two **curl** commands below.

11. For **utility-vm**, click **SSH** to launch a terminal and connect.
12. To verify the welcome page for `instance-group-1-xxxx`, run the following command:

```
curl 10.10.20.2
```

content_copy

The output should look like this (**do not copy; this is example output**):

```
<h1>Internal Load Balancing Lab</h1><h2>Client IP</h2>Your IP address :
10.10.20.50<h2>Hostname</h2>Server Hostname:
 instance-group-1-1zn8<h2>Server Location</h2>Region and Zone: us-central1-a
```

content_copy

13. To verify the welcome page for `instance-group-2-xxxx`, run the following command:

```
curl 10.10.30.2
```

content_copy

The output should look like this (**example output**):

```
<h1>Internal Load Balancing Lab</h1><h2>Client IP</h2>Your IP address :
10.10.20.50<h2>Hostname</h2>Server Hostname:
 instance-group-2-q5wp<h2>Server Location</h2>Region and Zone: us-central1-b
```

content_copy

Which of these fields identify the location of the backend?

Server Hostname

Server Location

Client IP

Submit

The **curl** commands demonstrate that each VM instance lists the Client IP and its own name and location. This will be useful when verifying that the Internal Load Balancer sends traffic to both backends.

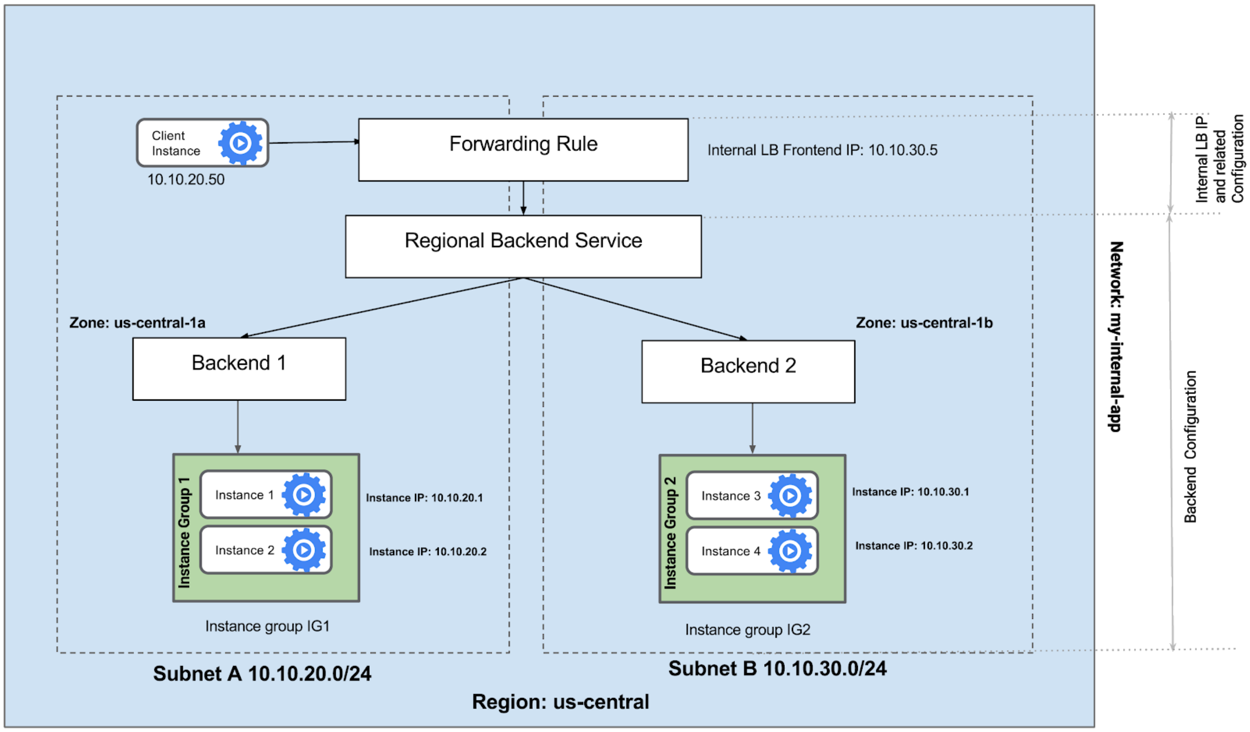
14. Close the SSH terminal to **utility-vm**:

```
exit
```

content_copy

Configure the Internal Load Balancer

Configure the Internal Load Balancer to balance traffic between the two backends (**instance-group-1** in us-central1-a and **instance-group-2** in us-central1-b), as illustrated in this diagram:



Start the configuration

1. In the Cloud Console, navigate to **Navigation menu** (☰) > **Network Services** > **Load balancing**, and then click **Create load balancer**.
2. Under **TCP Load Balancing**, click on **Start configuration**.
3. For **Internet facing or internal only**, select **Only between my VMs**.

Choosing **Only between my VMs** makes this Load Balancer internal. This choice requires the backends to be in a single region (us-central1) and does not allow offloading TCP processing to the Load Balancer.

4. Click **Continue**.
5. For **Name**, type `my-ilb`.

Configure the regional backend service

The backend service monitors instance groups and prevents them from exceeding configured usage.

1. Click on **Backend configuration**.
2. Set the following values, leave all other values at their defaults:

Property	Value (select option as specified)
Region	us-central1
Network	my-internal-app
Instance group	instance-group-1 (us-central1-a)

3. Click **Add backend**.
4. For **Instance group**, select **instance-group-2 (us-central1-b)**.
5. For **Health Check**, select **Create a health check**.
6. Set the following values, leave all other values at their defaults:

Property	Value (select option as specified)
Name	my-ilb-health-check
Protocol	TCP
Port	80

Health checks determine which instances can receive new connections. This HTTP health check polls instances every 5 seconds, waits up to 5 seconds for a response and treats 2 successful or 2 failed attempts as healthy or unhealthy, respectively.

7. Click **Save and Continue**.

8. Verify that there is a blue check mark next to **Backend configuration** in the Cloud Console. If not, double-check that you have completed all the steps above.

Configure the frontend

The frontend forwards traffic to the backend.

1. Click on **Frontend configuration**.
2. Specify the following, leaving all other values with their defaults:

Property	Value (type value or select option as specified)
Subnetwork	subnet-b
Internal IP	Reserve a static internal IP address

3\.

Specify the following, leaving all other values with their defaults:

Property	Value (type value or select option as specified)
Name	my-ilb-ip
Static IP address	Let me choose
Custom IP address	10.10.30.5

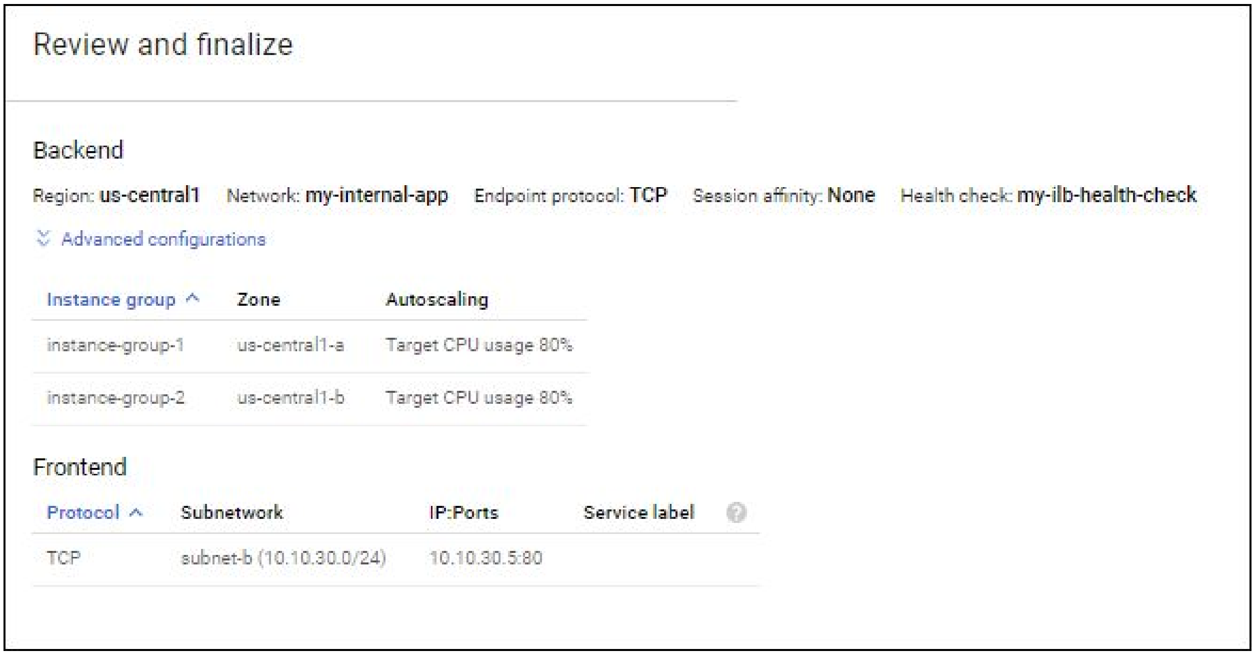
4\.

Click **Reserve**.

5. For **Ports**, type `80`.
6. Click **Done**.

Review and create the Internal Load Balancer

1. Click on **Review and finalize**.
2. Review the **Backend** and **Frontend**.



Review and finalize

Backend

Region: us-central1 Network: my-internal-app Endpoint protocol: TCP Session affinity: None Health check: my-ilb-health-check

Instance group	Zone	Autoscaling
instance-group-1	us-central1-a	Target CPU usage 80%
instance-group-2	us-central1-b	Target CPU usage 80%

Frontend

Protocol	Subnetwork	IP:Ports	Service label
TCP	subnet-b (10.10.30.0/24)	10.10.30.5:80	

3. Click on **Create**. Wait for the Load Balancer to be created, before moving to the next task.

Click Check my progress to verify the objective.

Configure the Internal Load Balancer

Check my progress

Test the Internal Load Balancer

Verify that the `my-ilb` IP address forwards traffic to **instance-group-1** in us-central1-a and **instance-group-2** in us-central1-b.

Access the Internal Load Balancer

1. In the Console, navigate to **Navigation menu** > **Compute Engine** > **VM instances**.
2. For **utility-vm**, click **SSH** to launch a terminal and connect.
3. To verify that the Internal Load Balancer forwards traffic, run the following command:

```
curl 10.10.30.5
```

content_copy

The output should look like this (**example output**):

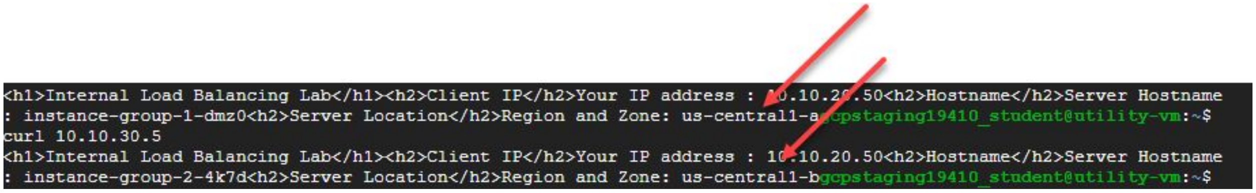
```
<h1>Internal Load Balancing Lab</h1><h2>Client IP</h2>Your IP address :
10.10.20.50<h2>Hostname</h2>Server Hostname:
 instance-group-1-1zn8<h2>Server Location</h2>Region and Zone: us-central1-a
```

content_copy

As expected, traffic is forwarded from the Internal Load Balancer (10.10.30.5) to the backend.

4. Run the same command a couple more times.

You should be able to see responses from **instance-group-1** in us-central1-a and **instance-group-2** in us-central1-b.



```
<h1>Internal Load Balancing Lab</h1><h2>Client IP</h2>Your IP address : 10.10.20.50<h2>Hostname</h2>Server Hostname
: instance-group-1-dmz0<h2>Server Location</h2>Region and Zone: us-central1-agcpstaging19410_student@utility-vm:~$
curl 10.10.30.5
<h1>Internal Load Balancing Lab</h1><h2>Client IP</h2>Your IP address : 10.10.20.50<h2>Hostname</h2>Server Hostname
: instance-group-2-4k7d<h2>Server Location</h2>Region and Zone: us-central1-bgcpstaging19410_student@utility-vm:~$
```