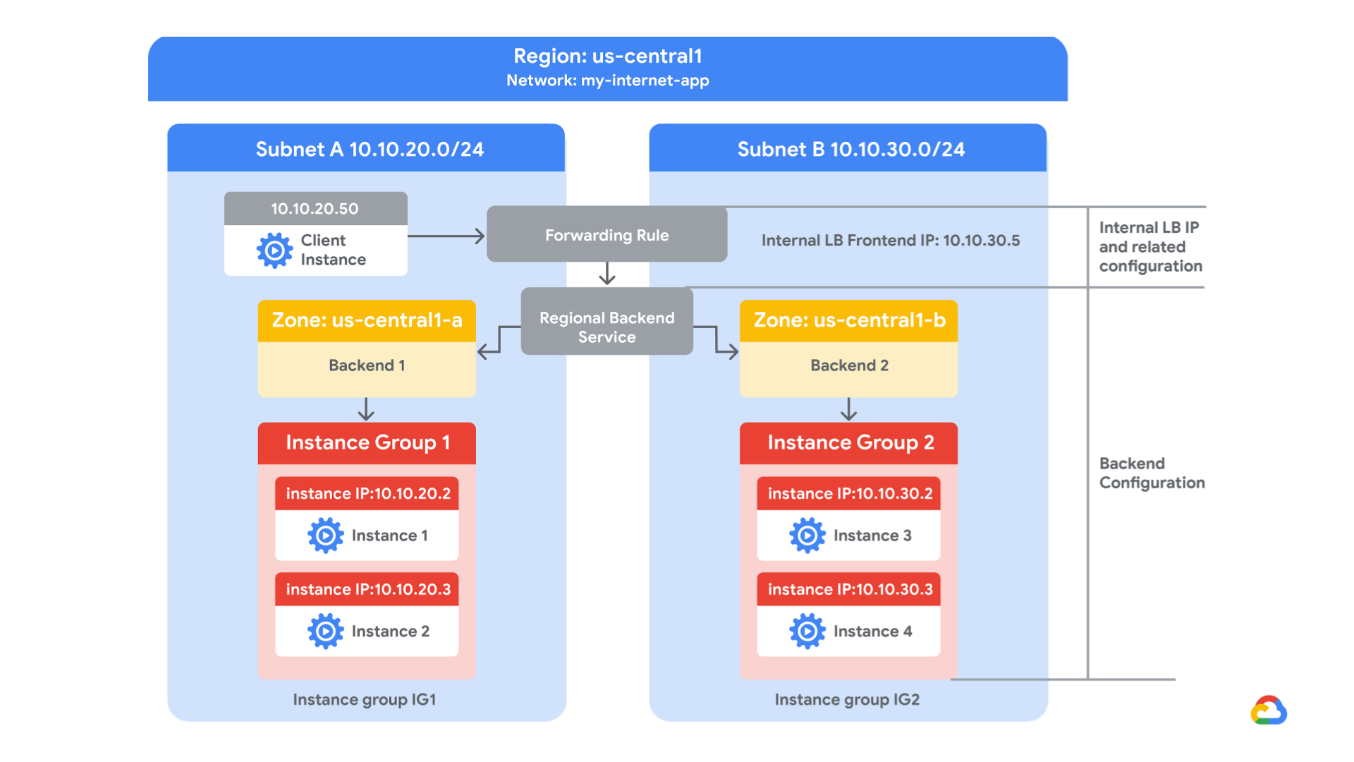
**Configuring an Internal Load Balancer**

experimentLabschedule1 hour 30 minutesuniversal\_currency\_alt5 Creditsshow\_chartIntroductory

**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 will learn how to perform the following tasks:

* Create internal traffic and health check firewall rules.
* Create a NAT configuration using Cloud Router.
* Configure two instance templates.
* Create two managed instance groups.
* Configure and test an internal load balancer.

**Setup and requirements**

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, 1:15:00), and make sure you can finish within that time.  
   There is no pause feature. You can restart if needed, but you have to start at the beginning.
3. When ready, click **Start lab**.
4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.
5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.  
   If you use other credentials, you'll receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

**Note:** Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

**Task 1. Configure internal traffic and health check firewall rules**

Configure firewall rules to allow internal traffic connectivity from sources in the 10.10.0.0/16 range. This rule allows incoming traffic from any client located in the subnet.

Health checks determine which instances of a load balancer can receive new connections. For HTTP 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.

Explore the my-internal-app network

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

* In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **VPC network > VPC networks**.  
  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 **europe-west1** 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 firewall rule to allow traffic from any sources in the 10.10.0.0/16 range

Create a firewall rule to allow traffic in the 10.10.0.0/16 subnet.

1. On the **Navigation menu** (Navigation menu icon), click **VPC network > Firewall**.  
   Notice the **app-allow-icmp** and **app-allow-ssh-rdp** firewall rules.

These firewall rules have been created for you.

1. Click **Create Firewall Rule**.
2. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | fw-allow-lb-access |
| Network | my-internal-app |
| Targets | Specified target tags |
| Target tags | backend-service |
| Source filter | IPv4 ranges |
| Source IPv4 ranges | 10.10.0.0/16 |
| Protocols and ports | Allow all |

**Note:**Make sure to include the **/16**in the **Source IPv4 ranges**.

1. Click **Create**.

Create the health check rule

Create a firewall rule to allow health checks.

1. On the **Navigation menu** (Navigation menu icon), click **VPC network > Firewall**.
2. Click **Create Firewall Rule**.
3. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | fw-allow-health-checks |
| Network | my-internal-app |
| Targets | Specified target tags |
| Target tags | backend-service |
| Source filter | IPv4 Ranges |
| Source IPv4 ranges | 130.211.0.0/22 **and** 35.191.0.0/16 |
| Protocols and ports | Specified protocols and ports |

**Note:**Make sure to include the **/22**and **/16**in the **Source IPv4 ranges**.

1. For **tcp**, check the checkbox and specify port **80**.
2. Click **Create**.

Click *Check my progress* to verify the objective.

Assessment Completed!

Configure internal traffic and health check firewall rules

Check my progress

*Assessment Completed!*

**Task 2. Create a NAT configuration using Cloud Router**

The Google Cloud VM backend instances that you setup in Task 3 will not be configured with external IP addresses.

Instead, you will setup the Cloud NAT service to allow these VM instances to send outbound traffic only through the Cloud NAT, and receive inbound traffic through the load balancer.

Create the Cloud Router instance

1. On the Google Cloud console title bar, type **Network services** in the Search field, then click **Network services** in the **Products & Page** section.
2. On the **Network service** page, click **Pin** next to Network services.
3. Click **Cloud NAT**.
4. Click **Get started** to configure a NAT gateway.
5. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Gateway name | nat-config |
| Network | my-internal-app |
| Region | europe-west1 |

1. Click **Cloud Router**, and select **Create new router**.
2. For **Name**, type **nat-router-europe-west1**.
3. Click **Create**.
4. In Create Cloud NAT gateway, click **Create**.

**Note:**Wait until the NAT Gateway Status changes to Running before moving onto the next task.

Click *Check my progress* to verify the objective.

Assessment Completed!

Create a NAT configuration using Cloud Router

Check my progress

*Assessment Completed!*

**Task 3. 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.

This task has been performed for you at the start of this lab. You will need to SSH into each instance group VM and run the following command to setup the environment.

1. On the **Navigation menu**, click **Compute Engine > VM instances**.  
   Notice the instances that start with **instance-group-1** and **instance-group-2**.
2. Select the SSH button next to **instance-group-1** to SSH into the VM.
3. If prompted allow SSH-in-browser to connect to VMs, click **Authorize**.
4. Run the following command to re-run the instance's startup script:

sudo google\_metadata\_script\_runner startup

Copied!

content\_copy

1. Repeat the previous steps for **instance-group-2**.
2. Wait for both startup scripts to finish executing, then close the SSH terminal to each VM. The output of the startup script should state the following:

Finished running startup scripts.

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. On the **Navigation menu**, click **Compute Engine > VM instances**.  
   Notice the 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.

1. Click **Create Instance**.
2. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | utility-vm |
| Region | europe-west1 |
| Zone | europe-west1-b |
| Series | E2 |
| Machine type | e2-medium (2 vCPU, 4 GB memory) |
| Boot disk | Debian GNU/Linux 12 (bookworm) |

1. Click **Advanced options**.
2. Click **Networking**.
3. For **Network interfaces**, click the dropdown to edit.
4. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | my-internal-app |
| Subnetwork | subnet-a |
| Primary internal IPv4 address | Ephemeral (Custom) |
| Custom ephemeral IP address | 10.10.20.50 |
| External IPv4 address | None |

1. Click **Done**.
2. Click **Create**.
3. Note that the internal IP addresses for the backends are **10.10.20.2** and **10.10.30.2**.

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

Click *Check my progress* to verify the objective.

Assessment Completed!

Configure instance templates and create instance groups

Check my progress

*Assessment Completed!*

1. For **utility-vm**, click **SSH** to launch a terminal and connect.
2. If prompted allow SSH-in-browser to connect to VMs, click **Authorize**.
3. To verify the welcome page for **instance-group-1-xxxx**, run the following command:

curl 10.10.20.2

Copied!

content\_copy

The output should look like this.

**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:

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

curl 10.10.30.2

Copied!

content\_copy

The output should look like this.

**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:

Which of these fields identify the location of the backend?



Client IP

checkServer Hostname

checkServer Location

Submit

**Note:**This will be useful when verifying that the internal load balancer sends traffic to both backends.

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

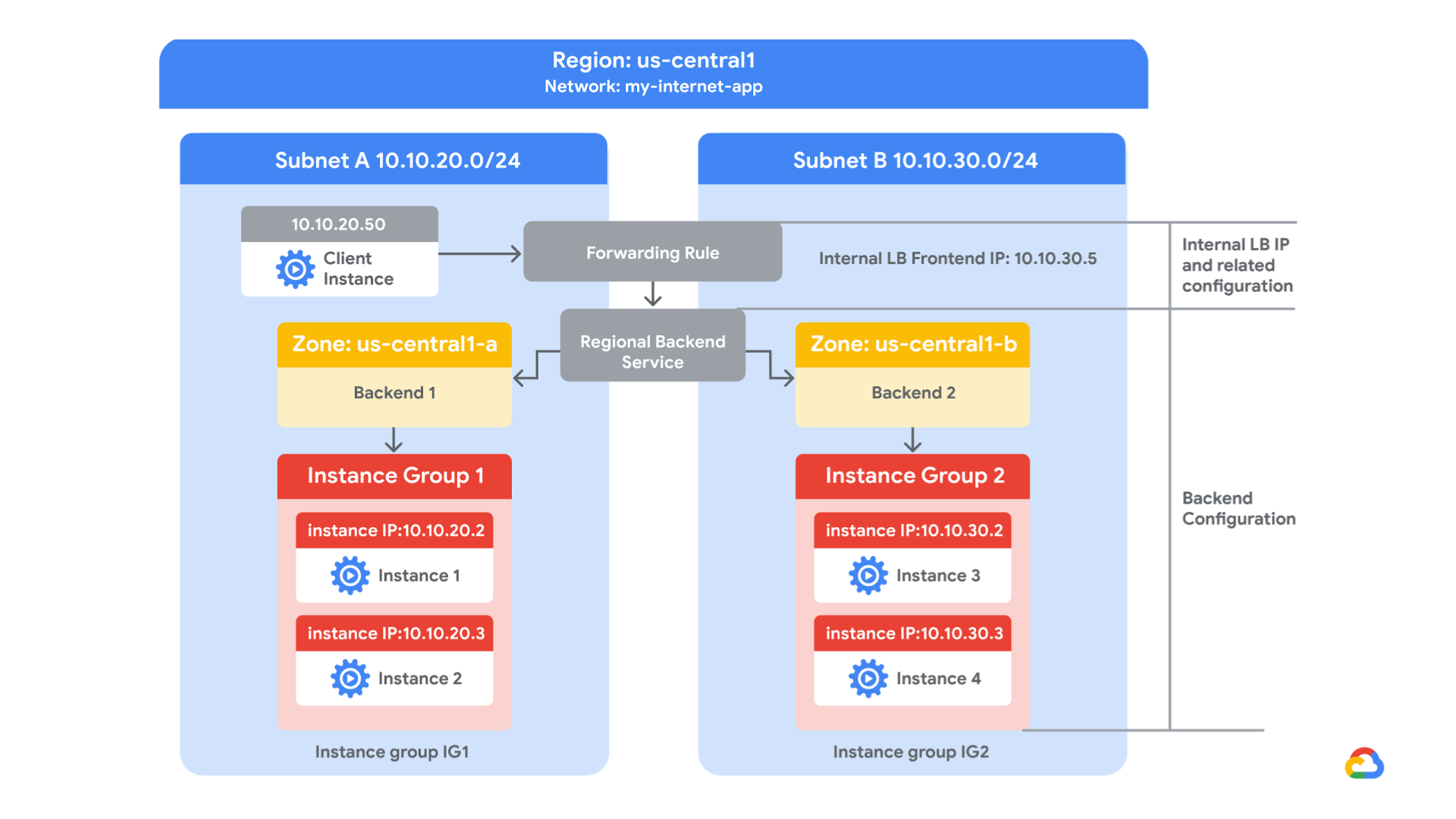
exit

Copied!

content\_copy

**Task 4. Configure the internal load balancer**

Configure the internal load balancer to balance traffic between the two backends (**instance-group-1** in europe-west1-d and **instance-group-2** in europe-west1-c), as illustrated in the network diagram.



Start the configuration

1. In the Cloud Console, on the **Navigation menu (Navigation menu icon)**, click **Network services > Load balancing**.
2. Click **Create load balancer**.
3. For **Type of load balancer**, select **Network Load Balancer (TCP/UDP/SSL)**, click **Next**.
4. For **Proxy or passthrough**, select **Passthrough load balancer** and click **Next**.
5. For **Public facing or internal**, select **Internal** and click **Next**.
6. For **Create load balance**, click **Configure**.
7. For **Load balancer name**, type my-ilb.
8. For **Region**, type **europe-west1**.
9. For **Network**, select my-internal-app from the dropdown.

Configure the regional backend service

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

1. Click **Backend configuration**.
2. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (select option as specified)** |
| Instance group | instance-group-1 (europe-west1-d) |

1. Click **Done**.
2. Click **Add a backend**.
3. For **Instance group**, select **instance-group-2 (europe-west1-c)**.
4. Click **Done**.
5. For **Health Check**, select **Create a health check**.
6. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (select option as specified)** |
| Name | my-ilb-health-check |
| Protocol | TCP |
| Port | 80 |
| Check interval | 10 sec |
| Timeout | 5 sec |
| Healthy threshold | 2 |
| Unhealthy threshold | 3 |

**Note:**Health checks determine which instances can receive new connections. This HTTP health check polls instances every 10 seconds, waits up to 5 seconds for a response, and treats 2 successful or 3 failed attempts as healthy threshold or unhealthy threshold, respectively.

1. Click **Save**.
2. Verify that there is a blue checkmark next to **Backend configuration** in the Cloud Console. If there isn't, double-check that you have completed all the steps above.

Configure the frontend

The frontend forwards traffic to the backend.

1. Click **Frontend configuration**.
2. Specify the following, and leave the remaining settings as their defaults.

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Subnetwork | subnet-b |
| Internal IP purpose > IP address | Create IP address |

1. Specify the following, and leave the remaining settings as 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 |

1. Click **Reserve**.
2. Under **Ports**, for **Port number**, type 80.
3. Click **Done**.

Review and create the internal load balancer

1. Click **Review and finalize**.
2. Review the **Backend** and **Frontend**.
3. Click **Create**.  
   Wait for the load balancer to be created before moving to the next task.

Click *Check my progress* to verify the objective.

Assessment Completed!

Configure the Internal Load Balancer

Check my progress

*Assessment Completed!*

**Task 5. Test the internal load balancer**

Verify that the **my-ilb** IP address forwards traffic to **instance-group-1** in europe-west1-d and **instance-group-2** in europe-west1-c.

Access the internal load balancer

1. On the **Navigation menu**, click **Compute Engine > VM instances**.
2. For **utility-vm**, click **SSH** to launch a terminal and connect.
3. If prompted allow SSH-in-browser to connect to VMs, click **Authorize**.
4. To verify that the internal load balancer forwards traffic, run the following command:

curl 10.10.30.5

Copied!

content\_copy

The output should look like this.

**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-2-1zn8<h2>Server Location</h2>Region and Zone: europe-west1-c

**Note:**As expected, traffic is forwarded from the internal load balancer (10.10.30.5) to the backend.

1. Run the same command a couple of times:

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

curl 10.10.30.5

Copied!

content\_copy

You should be able to see responses from **instance-group-1** in europe-west1-d and **instance-group-2** in europe-west1-c. If not, run the command again.

**Congratulations!**

In this lab, you created two managed instance groups in the europe-west1 region and a firewall rule to allow HTTP traffic to those instances and TCP traffic from the Google Cloud health checker. Then you configured and tested an internal load balancer for those instance groups.

**End your lab**