Configuring an Internal Load Balancer

1 hour 30 minutesFree

**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:

Timeline

Description automatically generated with medium confidence

Objectives

In this lab, you 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

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

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example,  and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click A green rectangle with white text

   Description automatically generated with medium confidence.
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. Graphical user interface, text, application

   Description automatically generated
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

1. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with 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), 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 **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 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), 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 | IP Ranges |
| Source IP ranges | 10.10.0.0/16 |
| Protocols and ports | Allow all |

Make sure to include the **/16**in the **Source IP ranges**.

1. Click **Create**.

**Create the health check rule**

Create a firewall rule to allow health checks.

1. On the **Navigation menu** (Navigation menu), 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 | IP Ranges |
| Source IP ranges | 130.211.0.0/22 **and** 35.191.0.0/16 |
| Protocols and ports | Specified protocols and ports |

Make sure to include the **/22**and **/16**in the **Source IP ranges**.

1. For **tcp**, specify port **80**.
2. Click **Create**.

Click *Check my progress* to verify the objective.

Configure internal traffic and health check firewall rules

Check my progress

**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. In the Cloud Console, on the **Navigation menu** (Navigation menu), click **Network services** > **Cloud NAT**.
2. Click **Get started**.
3. Specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Gateway name | nat-config |
| VPC network | my-internal-app |
| Region | us-central1 |

1. Click **Cloud Router**, and select **Create new router**.
2. For **Name**, type **nat-router-us-central1**.
3. Click **Create**.
4. In Create a NAT gateway, click **Create**.

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

Click *Check my progress* to verify the objective.

Create a NAT configuration using Cloud Router

Check my progress

**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.

**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. On the **Navigation menu** (Navigation menu), click **Compute Engine** > **Instance templates**.
2. Click **Create instance template**.
3. For **Name**, type **instance-template-1**
4. Under **Machine configuration**, For **Series**, Select **N1**.
5. Machine type **f1-micro(1 vCPU)**.
6. Click **Management, security, disks, networking, sole tenancy**.
7. Click **Management**.
8. 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 is 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. You can explore this script [here](https://storage.googleapis.com/cloud-training/gcpnet/ilb/startup.sh).

1. Click **Networking**.
2. For **Network interfaces**, specify the following, and leave the remaining settings as their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | my-internal-app |
| Subnet | subnet-a |
| Network tags | backend-service |
| External IP | None |

The network tag **backend-service** ensures that the firewall rule to allow traffic from any sources in the 10.10.0.0/16 subnet and the Health Check firewall rule applies to these instances.

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

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

1. Select the **instance-template-1** and click **Copy**.
2. Click **Management, security, disks, networking, sole tenancy**.
3. Click **Networking**.
4. For **Network interfaces**, select **subnet-b** as the **Subnet**.
5. Click **Create**.

**Create the managed instance groups**

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

1. On the **Navigation menu** (Navigation menu), click **Compute Engine** > **Instance groups**.
2. Click **Create Instance group**.
3. Specify the following, and leave the remaining settings as 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 |
| Instance template | instance-template-1 |
| **Autoscaling metrics** > **metrics type** (Click the pencil edit icon) | CPU utilization |
| Target CPU utilization | 80 |
| Cool-down period | 45 |
| Minimum number of instances | 1 |
| Maximum number of instances | 5 |

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. Just define the autoscaling policy, and the autoscaler performs automatic scaling based on the measured load.

1. Click **Create**.

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

1. Click **Create Instance group**.
2. Specify the following, and leave the remaining settings as 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 |
| Instance template | instance-template-2 |
| **Autoscaling metrics** > **metric type** (Click the pencil edit icon) | CPU utilization |
| Target CPU utilization | 80 |
| Cool-down period | 45 |
| Minimum number of instances | 1 |
| Maximum number of instances | 5 |

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

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 | us-central1 |
| Zone | us-central1-f |
| Series | N1 |
| Machine type | f1-micro (1 vCPU) |
| Boot disk | Debian GNU/Linux 10 (buster) |

1. Click **Management, security, disks, networking, sole tenancy**.
2. Click **Networking**.
3. For **Network interfaces**, click the pencil icon 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 IP | Ephemeral (Custom) |
| Custom ephemeral IP address | 10.10.20.50 |
| External IP | 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*.

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

Click *Check my progress* to verify the objective.

Configure instance templates and create instance groups

Check my progress

1. For **utility-vm**, click **SSH** to launch a terminal and connect. If you see the **Connection via Cloud Identity-Aware Proxy Failed** popup, click **Retry**.
2. 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

1. 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 (**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-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 Location



Server Hostname



Client IP

Submit

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

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 us-central1-a and **instance-group-2** in us-central1-b), as illustrated in the network diagram:

Timeline

Description automatically generated with medium confidence

**Start the configuration**

1. In the Cloud Console, on the **Navigation menu** (Navigation menu), click **Network Services** > **Load balancing**.
2. Click **Create load balancer**.
3. Under **TCP Load Balancing**, click **Start configuration**.
4. 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.

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

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

1. Click **Done**.
2. Click **Add backend**.
3. For **Instance group**, select **instance-group-2 (us-central1-b)**.
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 |

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 and Continue**.
2. Verify that there is a blue check mark 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 > IP address | Reserve static internal 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. For **Ports**, 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.

Configure the Internal Load Balancer

Check my progress

**Task 5. 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. On the **Navigation menu**, click **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 (**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-2-1zn8<h2>Server Location</h2>Region and Zone: us-central1-b

content\_copy

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

content\_copy

You should be able to see responses from **instance-group-1** in us-central1-a and **instance-group-2** in us-central1-b. If not, run the command again.

**Task 6. Review**

In this lab, you created two managed instance groups in the us-central1 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**

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2020 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.