**Task 1. 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.

**Create the HTTP firewall rule**

Create a firewall rule to allow HTTP traffic to the backends.

1. In the Cloud Console, navigate to **Navigation menu** (Navigation menu icon) > **VPC network** > **Firewall**.
2. Notice the existing **ICMP**, **internal**, **RDP**, and **SSH** firewall rules.

Each Google Cloud project starts with the **default** network and these firewall rules.

1. 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 | default-allow-http |
| Network | default |
| Targets | Specified target tags |
| Target tags | http-server |
| Source filter | IPv4 Ranges |
| Source IP ranges | 0.0.0.0/0 |
| Protocols and ports | Specified protocols and ports, and then *check* tcp, *type:* 80 |

1. **Note:**Make sure to include the **/0** in the **Source IP ranges** to specify all networks.
2. Click **Create**.

**Create the health check firewall rules**

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.

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 | default-allow-health-check |
| Network | default |
| Targets | Specified target tags |
| Target tags | http-server |
| Source filter | IPv4 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 |

1. **Note:**Make sure to enter the two **Source IP ranges** one-by-one and press SPACE in between them.
2. Click **Create**.

Click **Check my progress** to verify the objective.

Configure HTTP and health check firewall rules

Check my progress

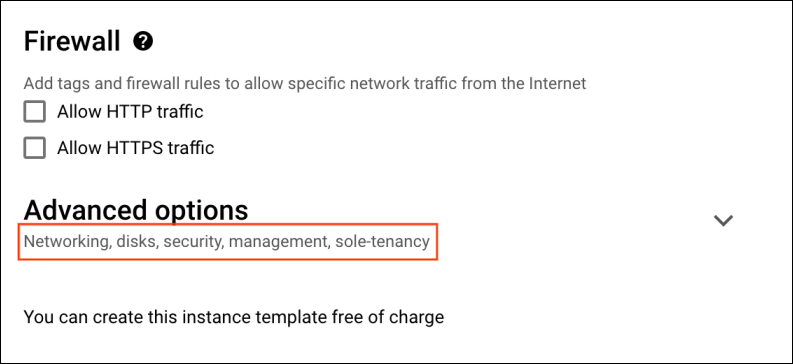
**Task 2. Configure instance templates and create instance groups**

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

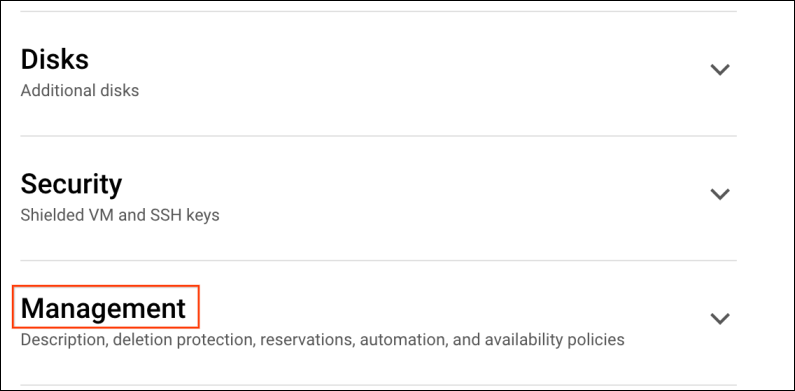
Configure the instance templates

An instance template is an API resource that you 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 one instance template for **us-east1** and one for **europe-west1**.

1. In the Cloud Console, navigate to **Navigation menu** (Navigation menu icon) > **Compute Engine** > **Instance templates**, and then click **Create instance template**.
2. For **Name**, type **us-east1-template**.
3. For **Series**, select **N1**.
4. Click **Management, security, disks, networking, sole tenancy**.



1. Click the **Management** tab.



1. Under **Metadata**, , click **+ADD ITEM** specify the following:

|  |  |
| --- | --- |
| **Key** | **Value** |
| startup-script-url | gs://cloud-training/gcpnet/httplb/startup.sh |

**Note:**The startup-script-url specifies a script that executes 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](https://storage.googleapis.com/cloud-training/gcpnet/httplb/startup.sh" \t "_blank).

1. Click **Networking**, for **Network tags**, type http-server.
2. For **Network interfaces** expand default network and set the following values.

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Network | default |
| Subnet | default (us-east1) |

**Note:**The network tag **http-server** ensures that the **HTTP** and **Health Check** firewall rules apply to these instances.

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

Now create another instance template for **subnet-b** by copying **us-east1-template**:

1. Click on **us-east1-template** and then click on the **CREATE SIMILAR** option from the top.
2. For **Name**, type **europe-west1-template**.
3. Click **Management, security, disks, networking, sole tenancy**.
4. Click **Networking**, expand default network.
5. For **Subnet**, select **default (europe-west1)**.
6. Click **Create**.

Create the managed instance groups

Create a managed instance group in **us-east1** and one in **europe-west1**.

1. In the **Navigation menu** (Navigation menu icon) click **Compute Engine > Instance groups** in the left menu.
2. Click **Create instance group**.
3. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | us-east1-mig |
| Location | Multiple zones |
| Region | us-east1 |
| Instance template | us-east1-template |
| Autoscaling > Autoscaling metrics (click the dropdown icon to edit) > Metric type | CPU utilization |
| Target CPU utilization | 80, click **Done**. |
| Cool-down period | 45 |
| Minimum number of instances | 1 |
| Maximum number of instances | 5 |

1. **Note:**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.
2. Click **Create**.

Now repeat the same procedure for create a second instance group for **europe-west1-mig** in **europe-west1**:

1. 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 | europe-west1-mig |
| Location | Multiple zones |
| Region | europe-west1 |
| Instance template | europe-west1-template |
| Autoscaling > Autoscaling metrics (click the dropdown icon to edit) > Metric type | CPU utilization |
| Target CPU utilization | 80, click **Done**. |
| Cool-down period | 45 |
| Minimum number of instances | 1 |
| Maximum number of instances | 5 |

1. Click **Create**.

Click **Check my progress** to verify the objective.

Configure instance templates and instance group

Check my progress

Verify the backends

Verify that VM instances are being created in both regions and access their HTTP sites.

1. Still in **Compute Engine**, click **VM instances** in the left menu.
2. Notice the instances that start with us-east1-mig and europe-west1-mig.

These instances are part of the managed instance groups.

1. Click on the **External IP** of an instance of us-east1-mig.

You should see the **Client IP** (your IP address), the **Hostname** (starts with us-east1-mig) and the **Server Location** (a zone in us-east1).

1. Click on the **External IP** of an instance of europe-west1-mig.

You should see the **Client IP** (your IP address), the **Hostname** (starts with europe-west1-mig) and the **Server Location** (a zone in europe-west1).

**Note:**The **Hostname** and **Server Location** identifies where the HTTP Load Balancer sends traffic.

Which of these fields identify the region of the backend?



Client IP



Hostname

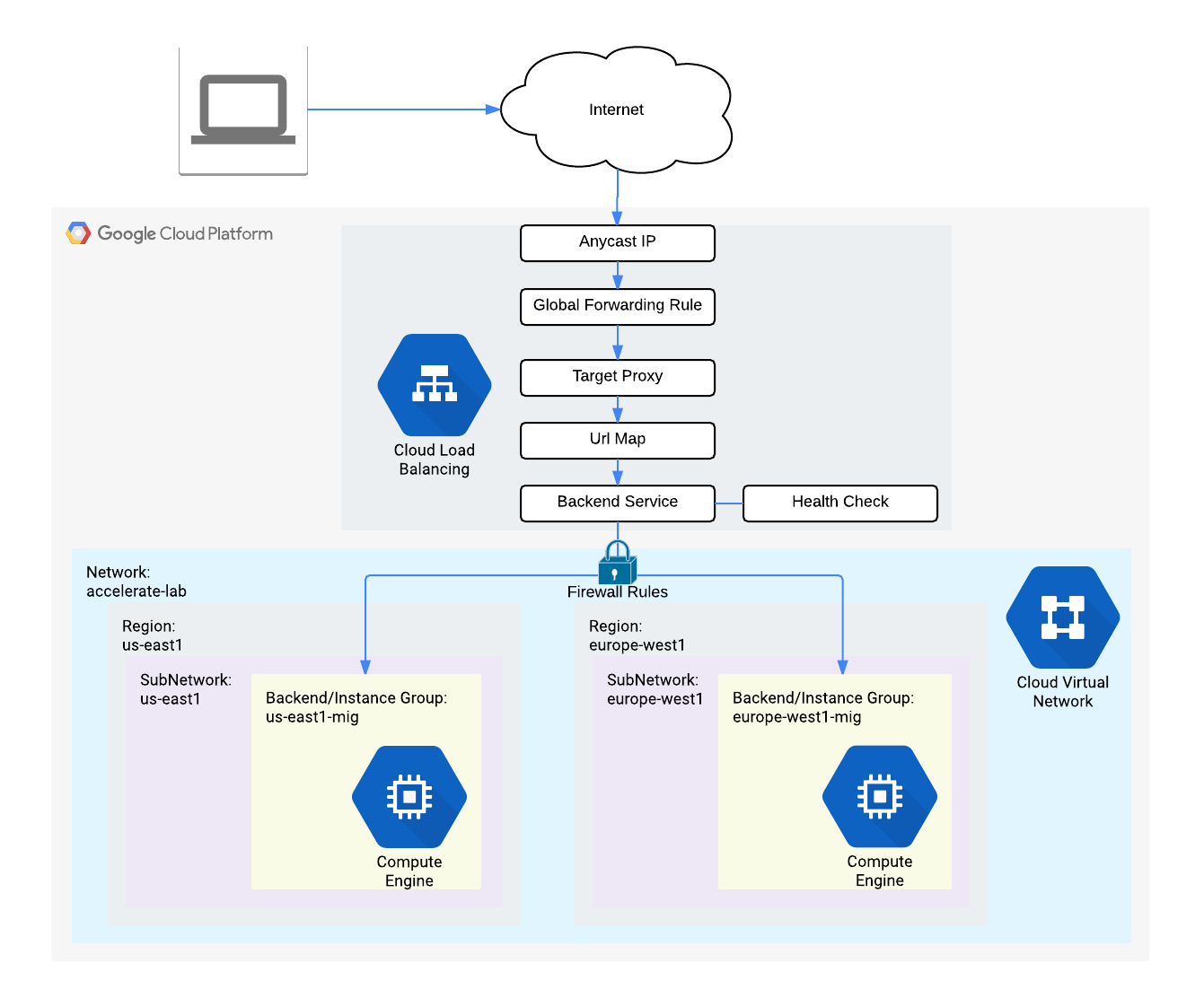


Server Location

Submit

**Task 3. Configure the HTTP Load Balancer**

Configure the HTTP Load Balancer to balance traffic between the two backends (**us-east1-mig** in us-east1 and **europe-west1-mig** in europe-west1), as illustrated in the network diagram:



Start the configuration

1. In the Cloud Console, click **Navigation menu** (Navigation menu icon) > click **Network Services** > **Load balancing**, and then click **Create load balancer**.
2. Under **HTTP(S) Load Balancing**, click on **Start configuration**.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. Select **From Internet to my VMs or serverless services**, and click **Continue**.
2. Set the **Name** to http-lb.

Configure the backend

Backend services direct incoming traffic to one or more attached backends. Each backend is composed of an instance group and additional serving capacity metadata.

1. Click on **Backend configuration**.
2. Click **Backend services & backend buckets** dropdown, click **Create a backend service**.
3. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (select option as specified)** |
| Name | http-backend |
| Instance group | us-east1-mig |
| Port numbers | 80 |
| Balancing mode | Rate |
| Maximum RPS | 50 |
| Capacity | 100 |

1. **Note:**This configuration means that the load balancer attempts to keep each instance of **us-east1-mig** at or below 50 requests per second (RPS).
2. Click **Done**.
3. Click **Add backend**.
4. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (select option as specified)** |
| Instance group | europe-west1-mig |
| Port numbers | 80 |
| Balancing mode | Utilization |
| Maximum backend utilization | 80 |
| Capacity | 100 |

1. **Note:**This configuration means that the load balancer attempts to keep each instance of **europe-west1-mig** at or below 80% CPU utilization.
2. Click **Done**.
3. For **Health Check**, select **Create a health check**.
4. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (select option as specified)** |
| Name | http-health-check |
| Protocol | TCP |
| Port | 80 |

1. **Note:**Health checks determine which instances 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.
2. Click **Save**.
3. Check the **Enable Logging** box.
4. Set the **Sample Rate** to 1:
5. Click **Create** to create the backend service, click **ok**.

Configure the frontend

The host and path rules determine how your traffic will be directed. For example, you could direct video traffic to one backend and static traffic do another backend. However, you are not configuring the Host and path rules in this lab.

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

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Protocol | HTTP |
| IP version | IPv4 |
| IP address | Ephemeral |
| Port | 80 |

1. Click **Done**.
2. Click **Add Frontend IP and port**.
3. Specify the following, leaving all other values at their defaults:

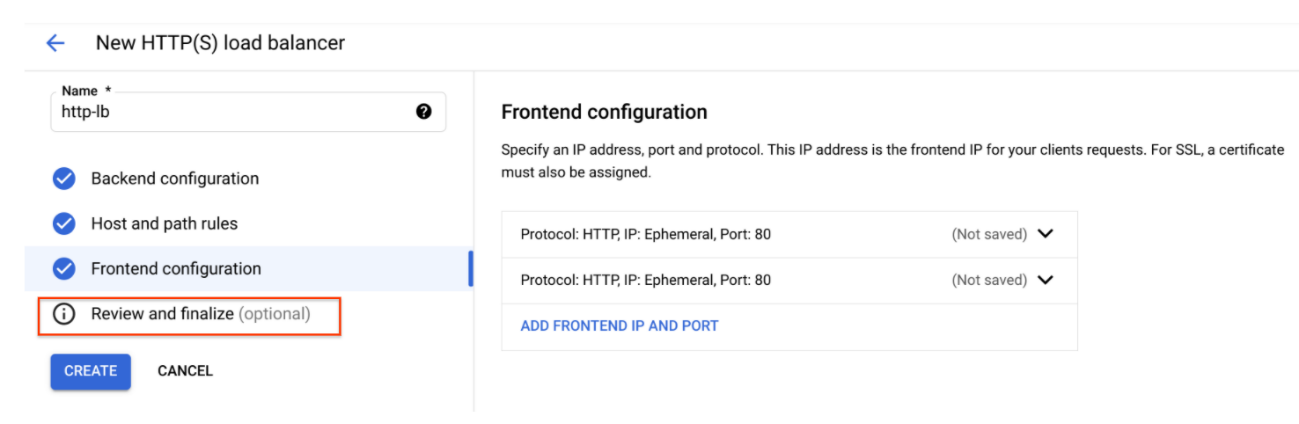
|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Protocol | HTTP |
| IP version | IPv6 |
| IP address | Ephemeral |
| Port | 80 |

1. Click **Done**.

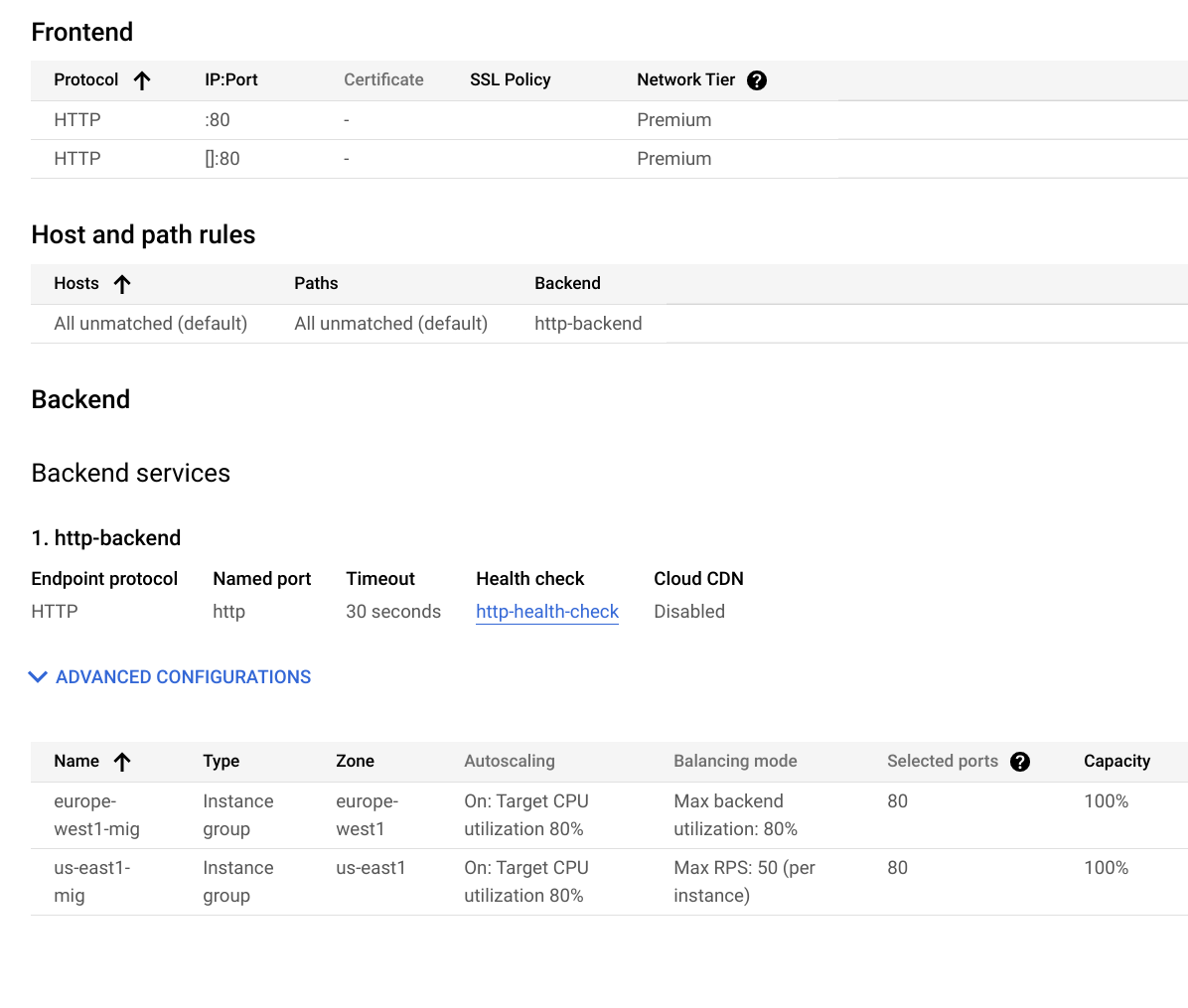
**Note:**HTTP(S) load balancing supports both IPv4 and IPv6 addresses for client traffic. Client IPv6 requests are terminated at the global load balancing layer, then proxied over IPv4 to your backends.

Review and create the HTTP Load Balancer

1. Click **Review and finalize**.



1. Review the **Backend services** and **Frontend**.



1. Click **Create**.
2. Wait for the load balancer to be created.
3. Click on the name of the load balancer (**http-lb**).
4. Note the IPv4 and IPv6 addresses of the load balancer for the next task. They will be referred to as [LB\_IP\_v4] and [LB\_IP\_v6], respectively.

**Note:**The IPv6 address is the one in hexadecimal format.

Click **Check my progress** to verify the objective.

Configure the HTTP Load Balancer

Check my progress

**Task 4. Test the HTTP Load Balancer**

Now that you created the HTTP Load Balancer for your backends, verify that traffic is forwarded to the backend service.

The HTTP load balancer should forward traffic to the region that is closest to you.



True



False

Access the HTTP Load Balancer

* To test IPv4 access to the HTTP Load Balancer, open a new tab in your browser and navigate to http://[LB\_IP\_v4]. Make sure to replace [LB\_IP\_v4] with the IPv4 address of the load balancer.

**Note:**It might take up to 5 minutes to access the HTTP Load Balancer. In the meantime, you might get a 404 or 502 error. Keep trying until you see the page of one of the backends.**Note:**Depending on your proximity to **us-east1** and **europe-west1**, your traffic is either forwarded to a **us-east1-mig** or **europe-west1-mig** instance.

If you have a local IPv6 address, try the IPv6 address of the HTTP Load Balancer by navigating to http://[LB\_IP\_v6]. Make sure to replace [LB\_IP\_v6] with the IPv6 address of the load balancer.

Stress test the HTTP Load Balancer

Create a new VM to simulate a load on the HTTP Load Balancer using siege. Then, determine if traffic is balanced across both backends when the load is high.

1. In the Console, navigate to **Navigation menu** (Navigation menu icon) > **Compute Engine** > **VM instances**.
2. Click **Create instance**.
3. Set the following values, leave all other values at their defaults:

|  |  |
| --- | --- |
| **Property** | **Value (type value or select option as specified)** |
| Name | siege-vm |
| Region | us-west1 |
| Zone | us-west1-c |
| Series | N1 |

**Note:**Given that **us-west1** is closer to **us-east1** than to **europe-west1**, traffic should be forwarded only to **us-east1-mig** (unless the load is too high).

1. Click **Create**.
2. Wait for the **siege-vm** instance to be created.
3. For **siege-vm**, click **SSH** to launch a terminal and connect.
4. Run the following command, to install siege:

sudo apt-get -y install siege

Copied!

content\_copy

Click **Check my progress** to verify the objective.

Test the HTTP Load Balancer

Check my progress

1. To store the IPv4 address of the HTTP Load Balancer in an environment variable, run the following command, replacing [LB\_IP\_v4] with the IPv4 address:

export LB\_IP=[LB\_IP\_v4]

Copied!

content\_copy

1. To simulate a load, run the following command:

siege -c 250 http://$LB\_IP

Copied!

content\_copy

The output should look like this:

New configuration template added to /home/cloudcurriculumdeveloper/.siege

Run siege -C to view the current settings in that file

1. In the Cloud Console, on the **Navigation menu** (Navigation menu icon), click **Network Services > Load balancing**.
2. Click **Backends**.
3. Click **http-backend**.
4. Navigate to http-lb.
5. Click on the **Monitoring tab**.
6. Monitor the **Frontend Location (Total inbound traffic)** between North America and the two backends for 2 to 3 minutes.

At first, traffic should just be directed to **us-east1-mig** but as the RPS increases, traffic is also directed to **europe-west1-mig**.



This demonstrates that by default traffic is forwarded to the closest backend but if the load is very high, traffic can be distributed across the backends.

1. Return to the **SSH** terminal of **siege-vm**.
2. Press CTRL+C to stop siege.

**Task 5. Create Cloud Armor rate limiting policy**

In this section you will use Cloud Armor to denylist the **siege-vm** from accessing the HTTP Load Balancer by setting a rate limiting policy.

1. In Cloud Shell, create security policy via gcloud:

gcloud compute security-policies create rate-limit-siege \

--description "policy for rate limiting"

Copied!

content\_copy

1. Next, add a rate limiting rule:

gcloud beta compute security-policies rules create 100 \

--security-policy=rate-limit-siege \

--expression="true" \

--action=rate-based-ban \

--rate-limit-threshold-count=50 \

--rate-limit-threshold-interval-sec=120 \

--ban-duration-sec=300 \

--conform-action=allow \

--exceed-action=deny-404 \

--enforce-on-key=IP

Copied!

content\_copy

1. Attach the security policy to the backend service http-backend:

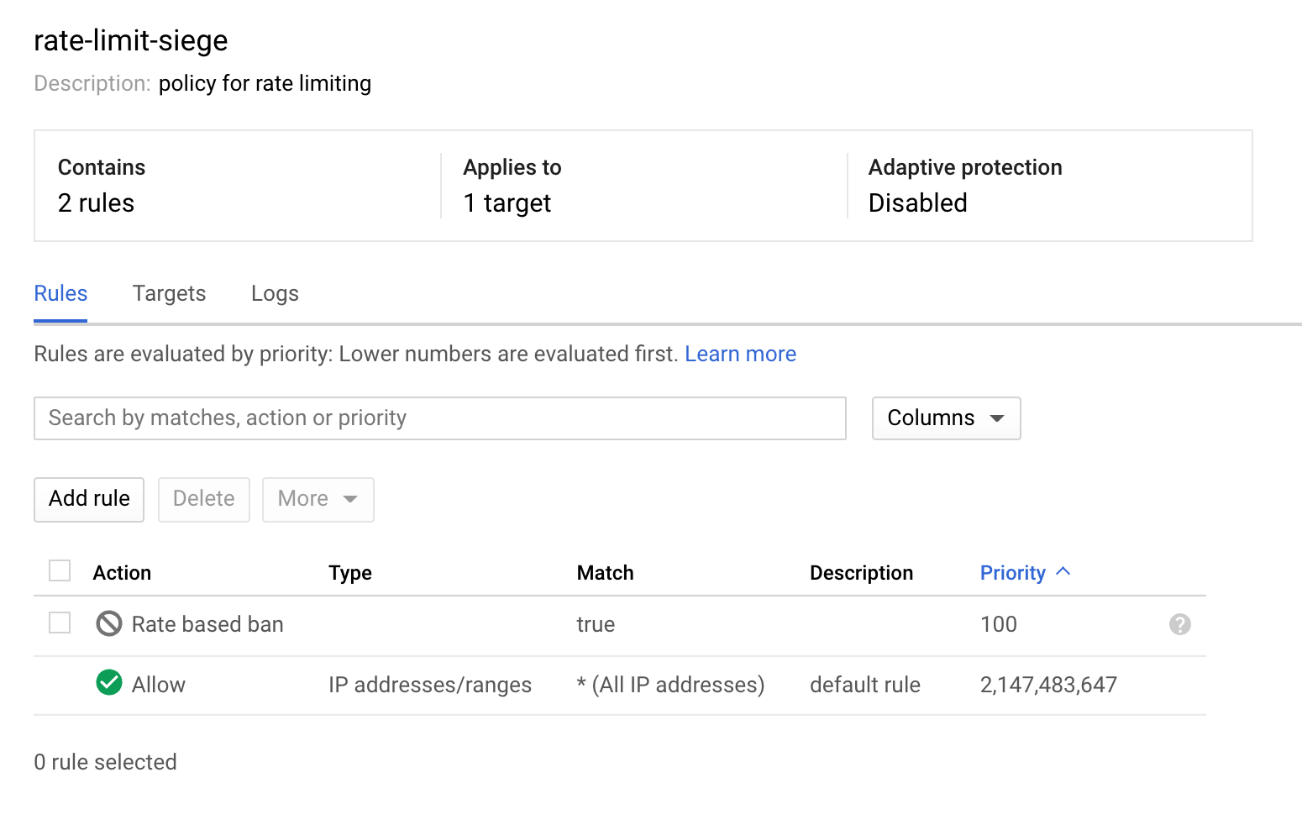
gcloud compute backend-services update http-backend \

--security-policy rate-limit-siege

Copied!

content\_copy

1. At the prompt to choose a region or global, type 1 to select global and hit **Enter**.
2. In the Console, navigate to **Navigation menu** > **Network Security** > **Cloud Armor**.
3. Click rate-limit-siege. Your policy should resemble the following:



Click **Check my progress** to verify the objective.

Create Cloud Armor Rate Limiting Policy

Check my progress

**Task 6. Verify the security policy**

1. Return to the SSH terminal of siege-vm.
2. Run a curl against the LB IP to verify you can still connect to it, should receive a 200 response:

curl http://$LB\_IP

Copied!

content\_copy

1. In the SSH terminal of siege-vm, to simulate a load, run the following command:

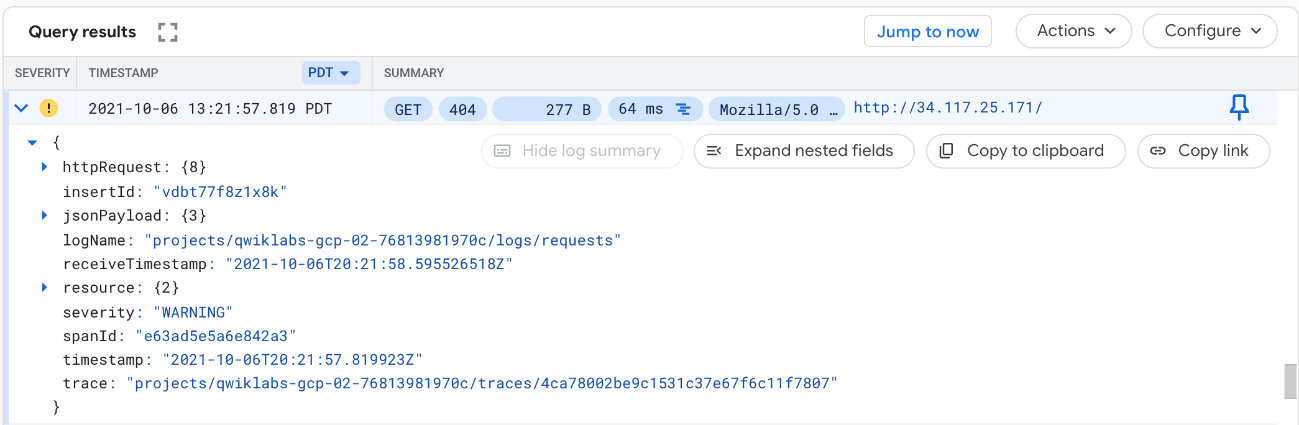
siege -c 250 http://$LB\_IP

Copied!

content\_copy

The command will not generate any output.

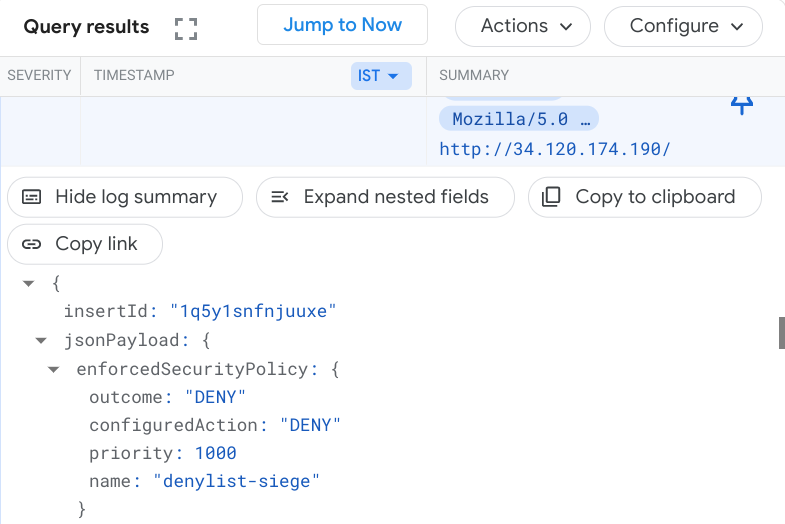
1. Explore the security policy logs to determine if this traffic is also blocked.
2. In the Console, navigate to **Navigation menu > Network Security > Cloud Armor**.
3. Click **rate-limit-siege**.
4. Click **Logs**.
5. Click **View policy logs**.
6. On the Logging page, make sure to clear all the text in the Query preview.
7. Select resource to **Cloud HTTP Load Balancer > http-lb-forwarding-rule > http-lb** then click **Apply**.
8. Now click **Run Query**.
9. Expand a log entry in Query results.



1. Expand **httpRequest**.

The request should be from the **siege-vm** IP address. If not, expand another log entry.

1. Expand jsonPayload.
2. Expand enforcedSecurityPolicy.



Notice that the configuredAction is to **DENY** with the name **rate-limit-siege**.

**Note:**Cloud Armor security policies create logs that can be explored to determine when traffic is denied and when it is allowed, along with the source of the traffic.

**Congratulations!**

You configured an HTTP Load Balancer with backends in us-east1 and europe-west1. Then, you stress tested the Load Balancer with a VM and denylisted the IP address via rate limiting with Cloud Armor. You were able to explore the security policy logs to identify why the traffic was blocked.