Creating Firewall Rules

1. In the left pane, clic **VPC Network -> Firewall Rules.**
2. Specify the following:

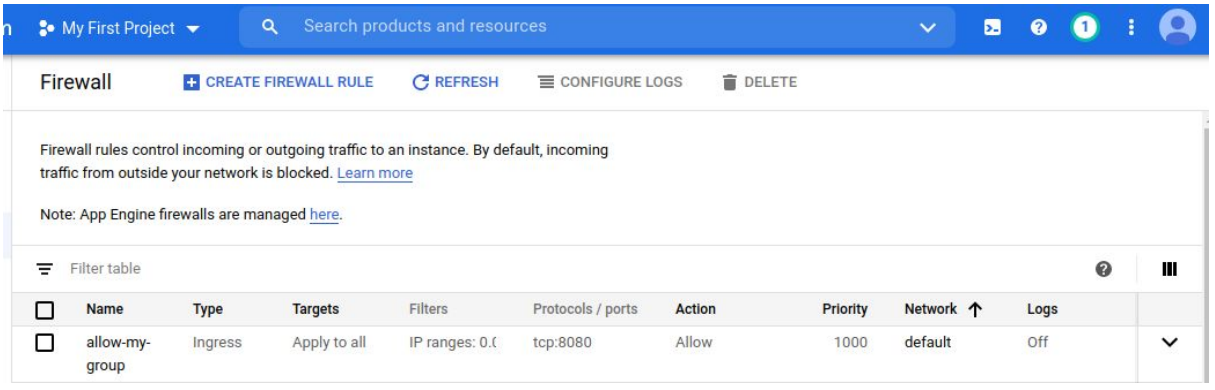
Property	Value
Name	Allow-my-group
Network	Default
Priority	1000
Direction	Ingress
Action On Match	Allow
Target	All Instances
Source filter	IP Ranges
Source IP ranges	0.0.0.0/0
Second source filter	None
Protocols and ports	tcp:8080

Resolution:

Using the **gcloud cli**

```
gcloud compute firewall-rules create allow-my-group --network=default
--priority=1000 --direction=Ingress --source-ranges=0.0.0.0/0 --allow tcp:8080
```

```
Creating firewall...done.
NAME            NETWORK  DIRECTION  PRIORITY  ALLOW     DENY  DISABLED
allow-my-group  default  INGRESS    1000      tcp:8080        False
```



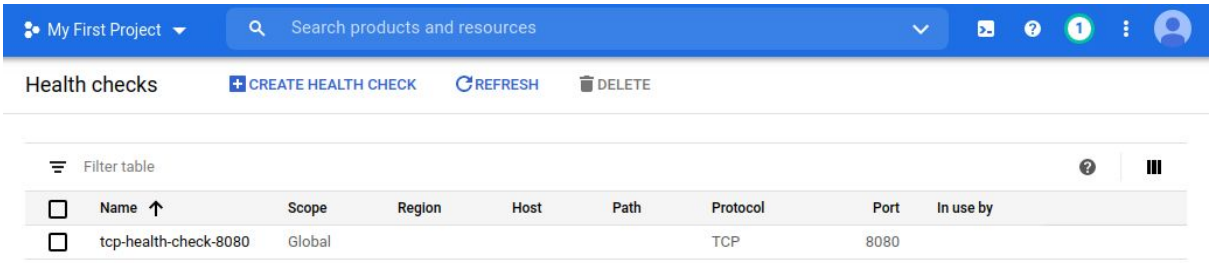
Creating Health Check

3. In the left pane, clic **Compute Engine -> Health checks.**
4. Clic **Create Health checks for HTTP or HTTPS.**
5. Specify the following step if you need create TCP health check:

```
gcloud compute health-checks create tcp tcp-health-check-8080 --port=8080
```

Resolution:

```
gonzalo@mix:~/Globant/mentoring-gcp$ gcloud compute health-checks create
tcp tcp-health-check-8080 --port=8080
Created
[https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/global/hea
lthChecks/tcp-health-check-8080].
NAME                  PROTOCOL
tcp-health-check-8080  TCP
gonzalo@mix:~/Globant/mentoring-gcp$
```



Creating Template Instance

6. In the left pane, clic **Compute Engine -> Instance Template.**
7. Clic **Create Instance Template.**
8. Specify the following:

Property	Value (type value or select option as specified)
Name	**instance-template-test**
Machine Type	**Select “small” or “micro”**
Image	**Ubuntu 18**
Identity and API access	**Allow default access**
Firewall	**Checks HTTP and HTTPS**

Script startup	#! /bin/bash echo "[1/5] Updating system..." sudo apt-get update echo "[2/5] Installing Java 8..." sudo apt-get -y install openjdk-8-jre-headless cd /opt echo "[3/4] Creating services folder" sudo mkdir services cd services echo "[4/5] Downloading app..." sudo wget http://github.com/RogelioDavid/apps/raw/master/app.jar echo "[5/5] Running app..." java -DAPP_ENV=gcp -jar app.jar > ${HOME}/logs.log &

4. click **Create.**

Resolution:

First I will create a Firewall to allow HTTP (TCP 80) and HTTPS (TCP 443)

```
gonzalo@mix:~/Globant/mentoring-gcp$ gcloud compute firewall-rules create
mentoring-rule --allow tcp:443,tcp:80 --source-tags=mentoring,roger
Creating firewall...⠧Created
[https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/global/firewalls/mentoring-rule].
Creating firewall...done.
NAME          NETWORK  DIRECTION  PRIORITY  ALLOW         DENY
DISABLED
mentoring-rule  default  INGRESS    1000      tcp:443,tcp:80       False
gonzalo@mix:~/Globant/mentoring-gcp$
```

Now I will create the file script as

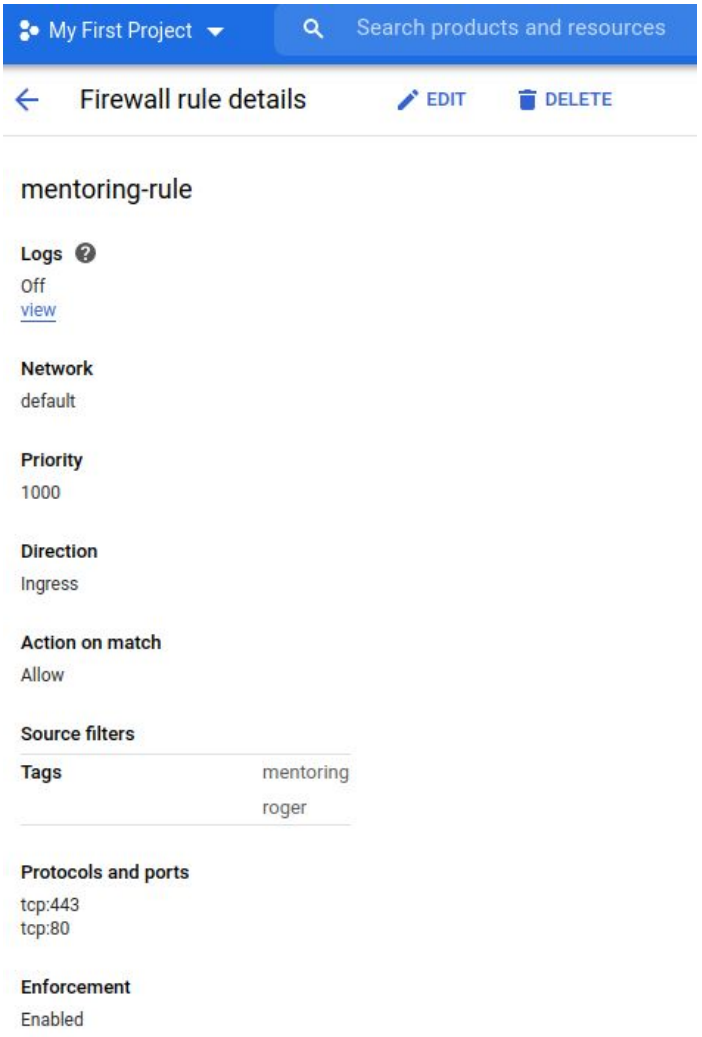
```
#! /bin/bash
sudo add-apt-repository -y ppa:webupd8team/java
sudo apt-get install -y software-properties-common debconf-utils
sudo apt-get update
sudo echo "oracle-java8-installer shared/accepted-oracle-license-v1-1 select true" |
sudo debconf-set-selections
```

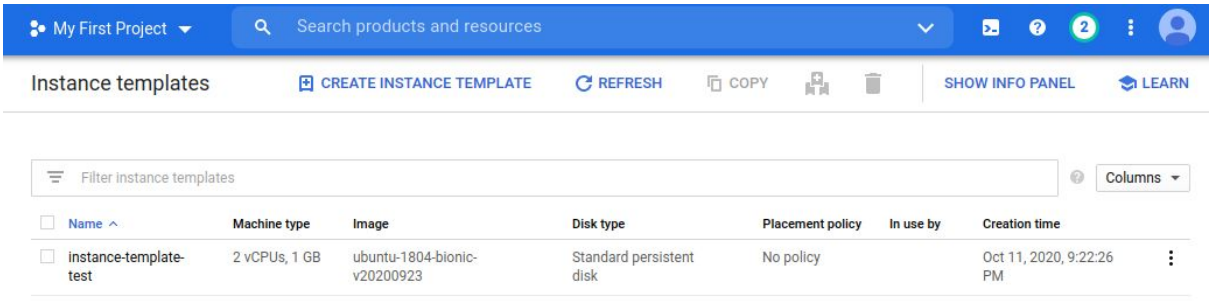
```
sudo apt-get install -y oracle-java8-installer
cd /opt
sudo mkdir services
cd services
sudo wget http://github.com/RogelioDavid/apps/raw/master/app.jar
java -DAPP_ENV=gcp -jar app.jar > /opt/logs.log &
```

Finally, I will create the template running

```
gonzalo@mix:~/Globant/mentoring-gcp$ gcloud compute instance-templates
create instance-template-test --machine-type=e2-micro
--image-project=ubuntu-os-cloud --image=ubuntu-1804-bionic-v20200923
--tags=mentoring,roger --metadata-from-file startup-script=./startup.sh
Created
[https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/global/inst
anceTemplates/instance-template-test].
NAME                  MACHINE_TYPE  PREEMPTIBLE  CREATION_TIMESTAMP
instance-template-test  e2-micro                2020-10-11T17:22:26.846-07:00
gonzalo@mix:~/Globant/mentoring-gcp$
```

Results





Creating Instance Groups

1. In the left pane, clic **Compute Engine -> Instance Groups.**
2. Click **Create Instance Groups.**
3. Specify the following:

Property	Value (type value or select option as specified)

Name	**instance-group-test**
Location	**Single Zone**
Region	**<SELECTED ZONE WITH LESS LATENCY ACORDING YOUR ZONE>**
Zone	**<SELECTED ZONE WITH NEWER LETTER>**
Group Type	Managed Instance group
Instance template	Instance template Test
AutoScaling	On
Autoscale based on	% CPU
Target CPU usage	60
Minimum number of instances	1
Maximum number of instances	6
Cool-down period	60
Health check	<HEALTH CHECK CREATED>

Resolution:

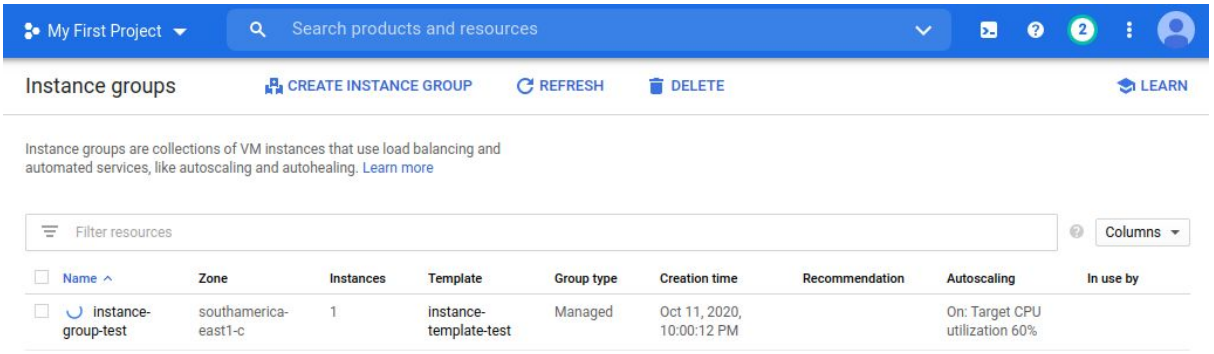
First, I will create the Instance Group (Managed type)

```
gonzalo@mix:~/Globant/mentoring-gcp$ gcloud compute instance-groups
managed create instance-group-test --zone=southamerica-east1-c
--template=instance-template-test --health-check=tcp-health-check-8080 --size=1
Created
[https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/zones/sout
hamerica-east1-c/instanceGroupManagers/instance-group-test].
NAME                LOCATION            SCOPE  BASE_INSTANCE_NAME   SIZE
TARGET_SIZE  INSTANCE_TEMPLATE       AUTOSCALED
instance-group-test  southamerica-east1-c  zone   instance-group-test  0     1
instance-template-test  no
gonzalo@mix:~/Globant/mentoring-gcp$
```

Then, I will create an Auto scaling for this instance group

```
gonzalo@mix:~/Globant/mentoring-gcp$ gcloud compute instance-groups
managed set-autoscaling instance-group-test --zone=southamerica-east1-c
--scale-based-on-cpu --target-cpu-utilization=0.6 --min-num-replicas=1
--max-num-replicas=6 --cool-down-period=60 --mode=on
Created
```

```
[https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/zones/southamerica-east1-c/autoscalers/instance-group-test-xfxr].
---
autoscalingPolicy:
  coolDownPeriodSec: 60
  cpuUtilization:
    utilizationTarget: 0.6
  maxNumReplicas: 6
  minNumReplicas: 1
  mode: ON
creationTimestamp: '2020-10-11T18:10:32.059-07:00'
id: '3537780098779190247'
kind: compute#autoscaler
name: instance-group-test-xfxr
selfLink:
https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/zones/southamerica-east1-c/autoscalers/instance-group-test-xfxr
status: ACTIVE
target:
https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/zones/southamerica-east1-c/instanceGroupManagers/instance-group-test
zone:
https://www.googleapis.com/compute/v1/projects/tonal-plasma-292323/zones/southamerica-east1-c
gonzalo@mix:~/Globant/mentoring-gcp$
```



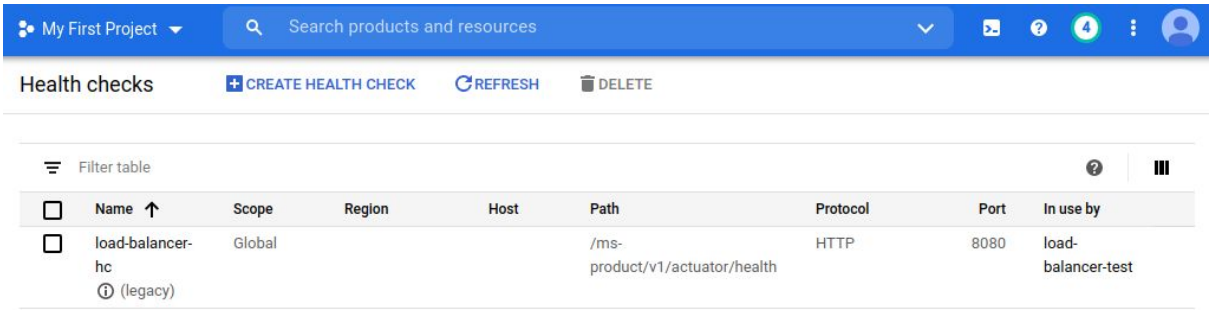
Creating Load Balancer

1. In the left pane, click **Network Services -> Load Balancing.**
2. Click **Create Load Balancer.**
3. Selected TCP Load Balancing -> Start
4. In Internet facing or internal only Option Internet facing or internal only
5. in Múltiple regions or single regions - Single region only
6. in Connection termination - No (TCP)
7. Specify the following:

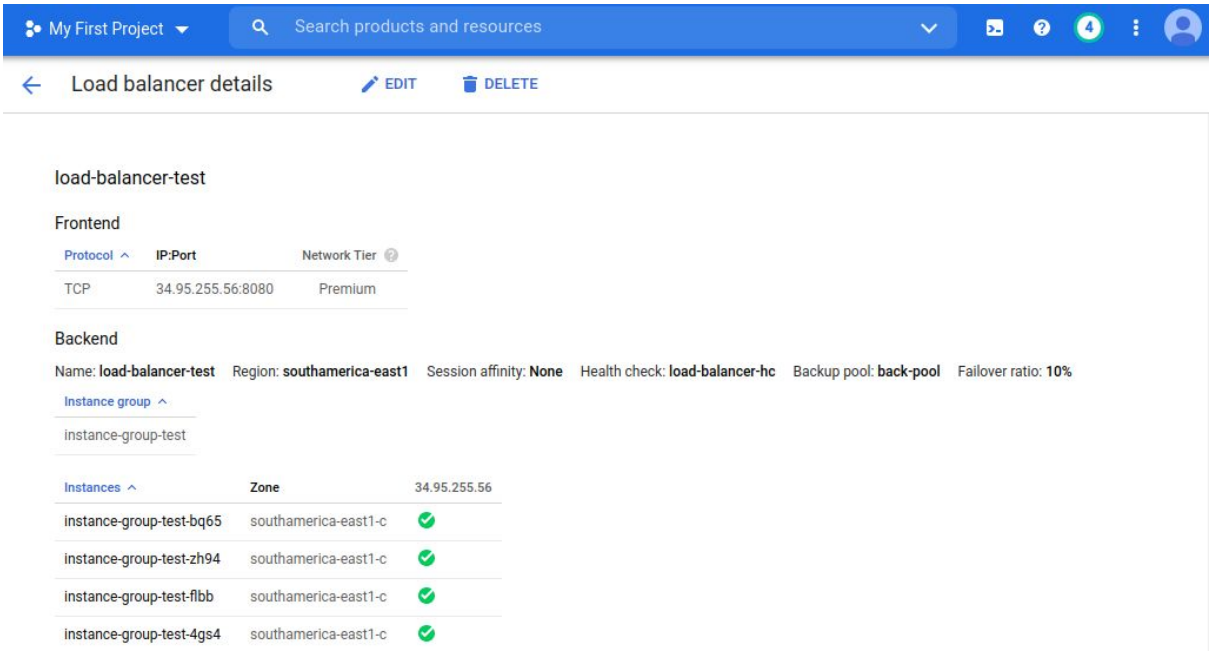
Property	Value (type value or select option as specified)
Name	Load-balancer-test
Location	Single Zone
Backend configuration	- Check same name - Region -> same region with creating Instance group - Selected Instance Group already create - Health check -> select Health check already create

Frontend configuration	- For Each port
	  - Add name
	  - Network Service Tier: Premium
	  - IP: Ephemeral
	  - Port: your port

Resolution:
First I will create another Health Check, for any reason (unkown for me) the console doesn't let me take a created health check

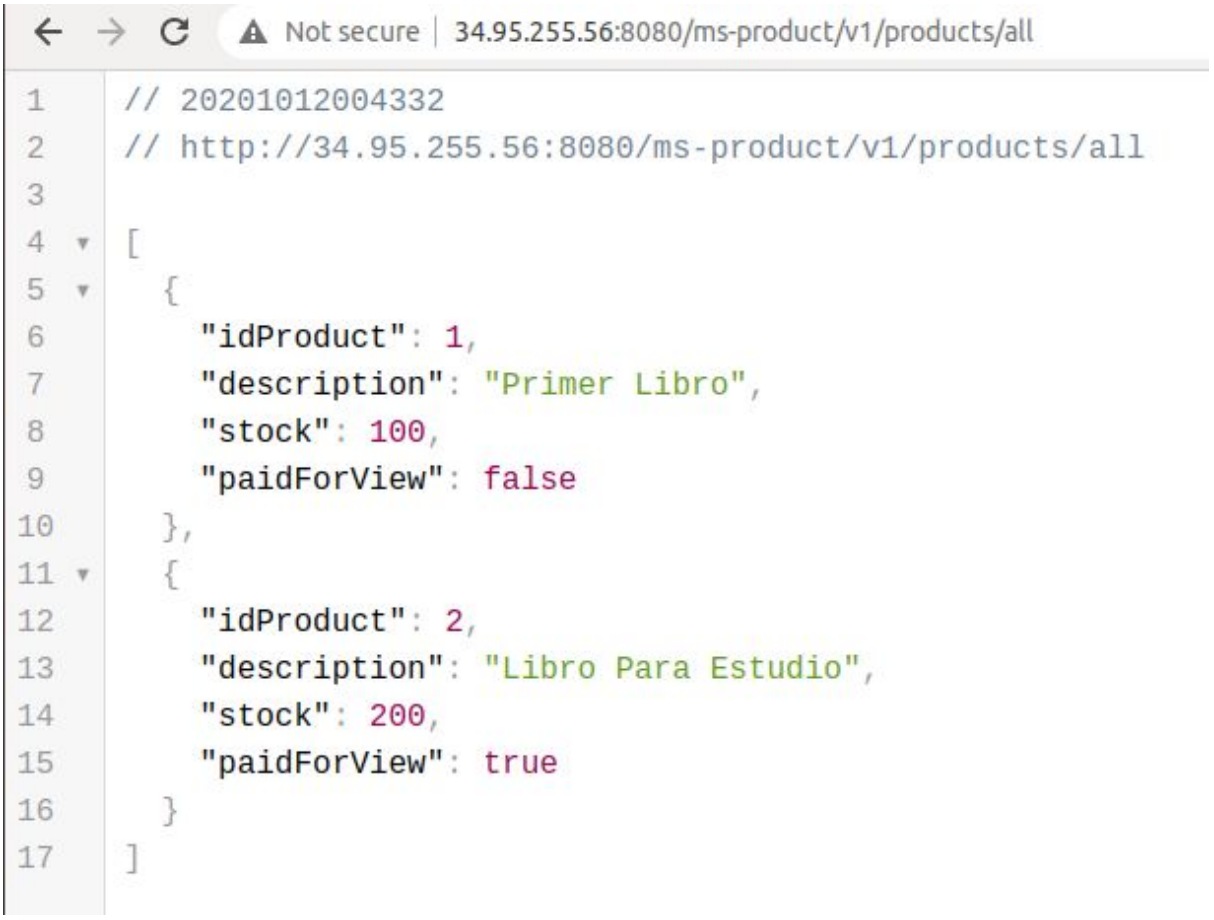


Now I will create the Load Balancer for this Health Check



Finally I will checkout the Application through Load Balancer

```
http://34.95.255.56:8080/ms-product/v1/actuator/health
```


Not secure | 34.95.255.56:8080/ms-product/v1/products/all
// 20201012004332
// http://34.95.255.56:8080/ms-product/v1/products/all
[
{
"idProduct": 1,
"description": "Primer Libro",
"stock": 100,
"paidForView": false
},
{
"idProduct": 2,
"description": "Libro Para Estudio",
"stock": 200,
"paidForView": true
}
]