**PRACTICAL: 3**

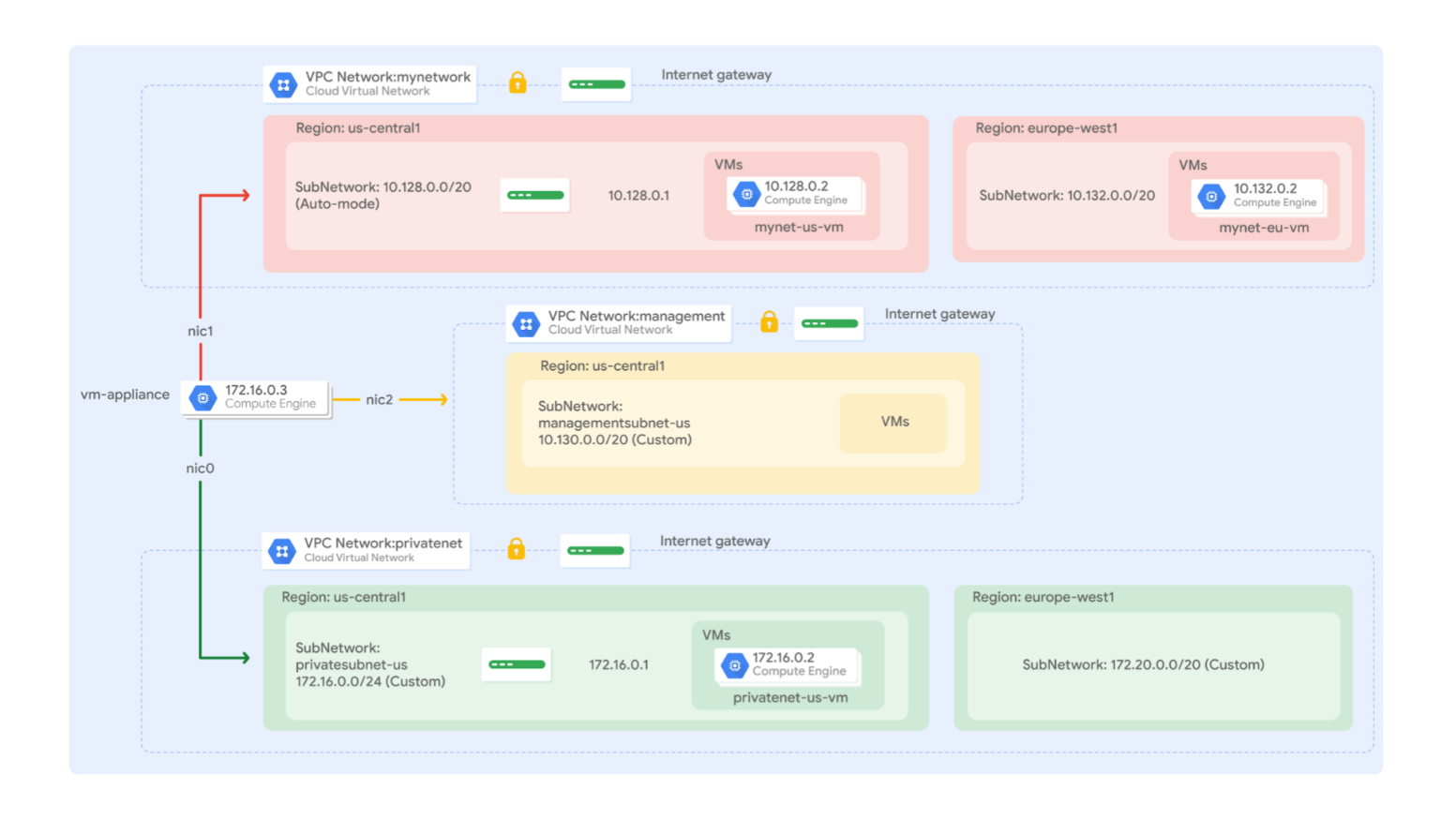
**AIM:**

Networking in the Google Cloud: Networking is a principle theme of cloud computing—it is the underlying structure of GCP and it is what connects all your resources and services to one another. This fundamental-level quest lab will cover essential GCP networking services and will give you hands-on practice with specialized tools for developing mature networks. From learning the ins-and-outs of VPCs, to creating enterprise-grade load balancers, Networking in the Google Cloud will give you the practical experience needed so you can start building robust networks right away.

**THEORY:**

Google Cloud Virtual Private Cloud (VPC) networks are by default isolated private networking domains. Networks have a global scope and contain regional subnets. VM instances within a VPC network can communicate among themselves via internal IP addresses as long as firewall rules permit. However, no internal IP address communication is allowed between networks, unless you set up mechanisms such as VPC Network Peering or Cloud VPN.

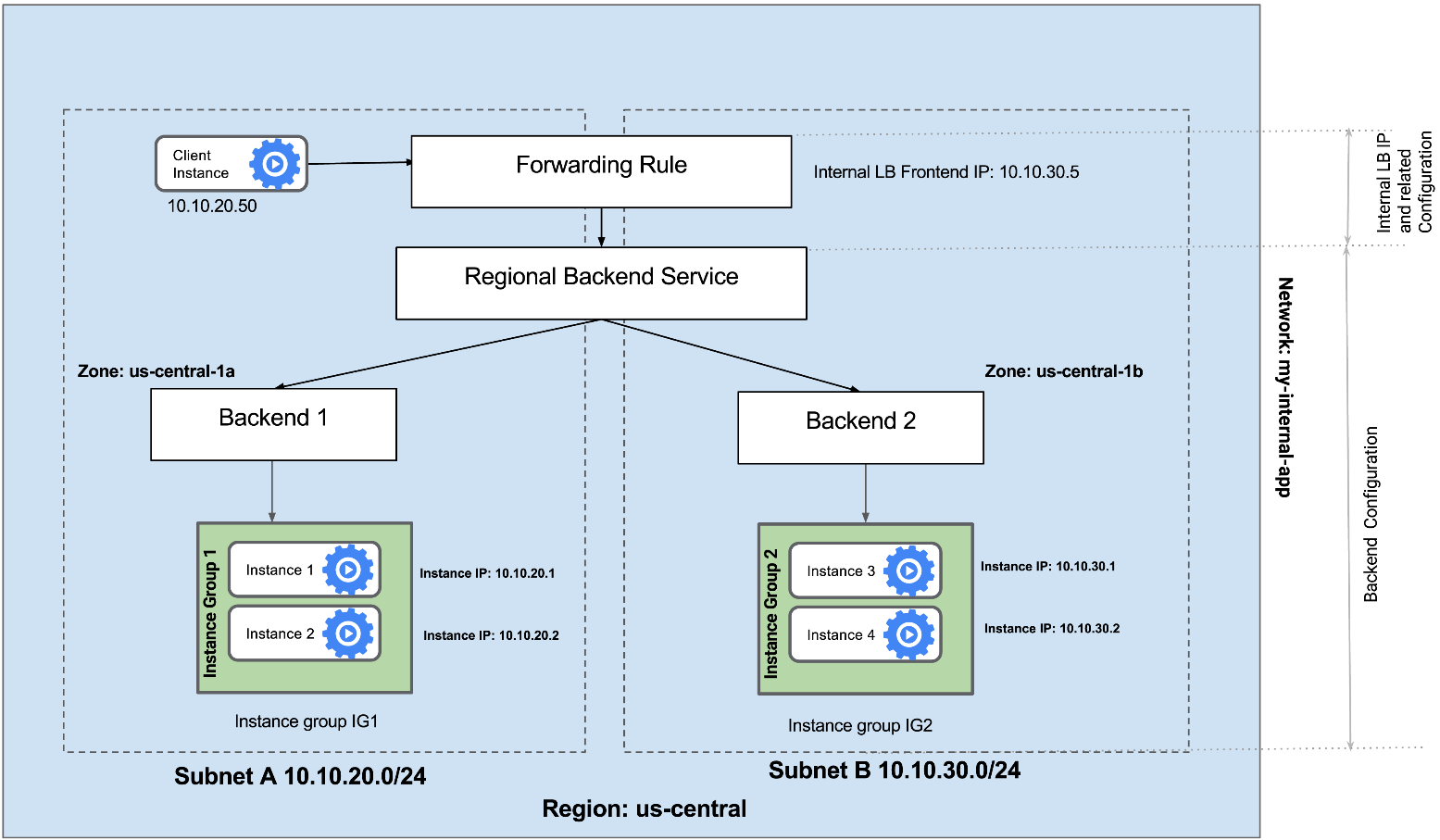
Every instance in a VPC network has a default network interface. You can create additional network interfaces attached to your VMs, but each interface must attach to a different VPC network. Multiple network interfaces enable you to create configurations in which an instance connects directly to several VPC networks. Each of the interfaces must have an internal IP address, and each interface can also have an external IP address. Each instance can have up to 8 interfaces, depending on the instance's type.



Google Cloud HTTP(S) load balancing is implemented at the edge of Google's network in Google's points of presence (POP) around the world. User traffic directed to an HTTP(S) load balancer enters the POP closest to the user and is then load balanced over Google's global network to the closest backend that has sufficient capacity available.

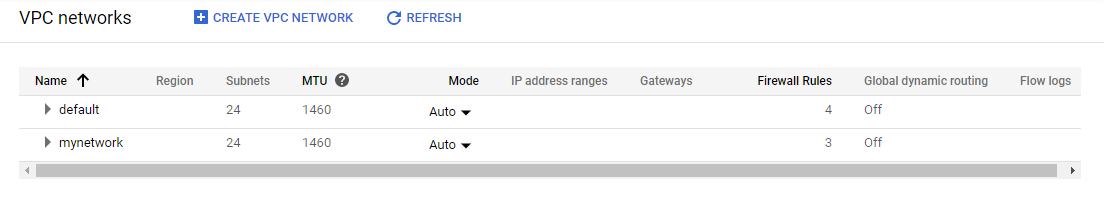
Cloud Armor IP allow list/deny list enable you to restrict or allow access to your HTTP(S) load balancer at the edge of the Google Cloud, as close as possible to the user and to malicious traffic. This prevents malicious users or traffic from consuming resources or entering your virtual private cloud (VPC) networks.

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.

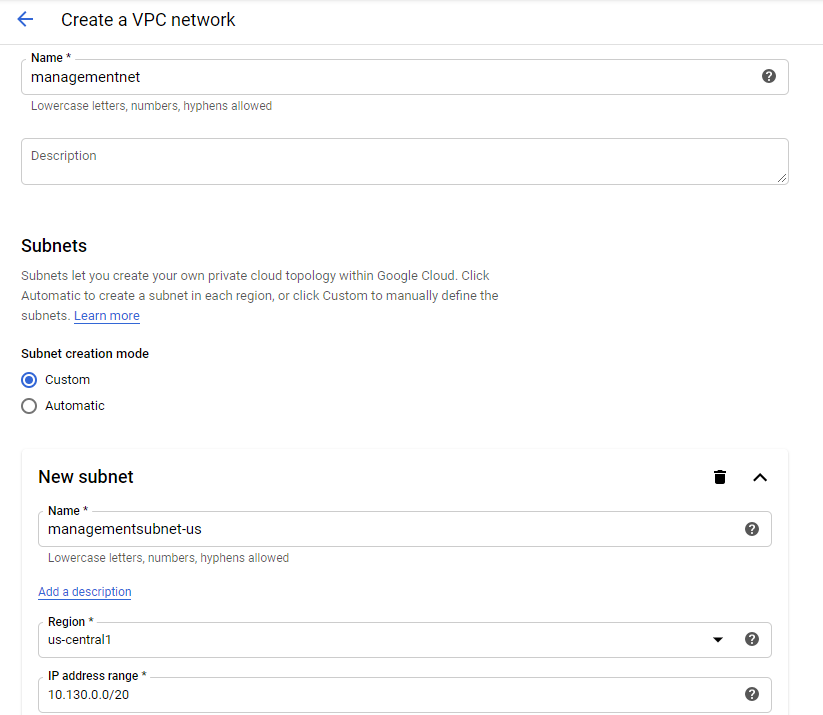


**OUTPUT:**

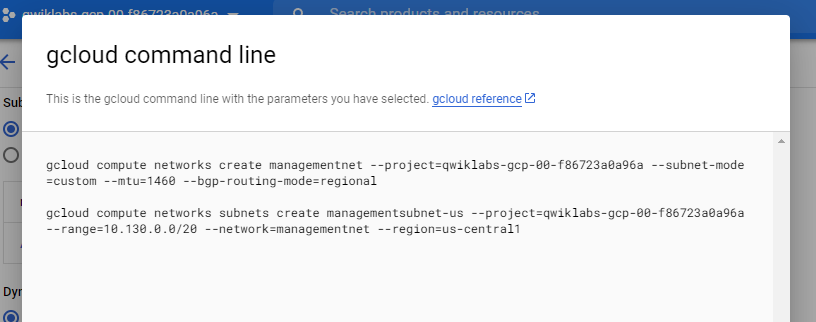
**3.2 - Multiple VPC Networks**



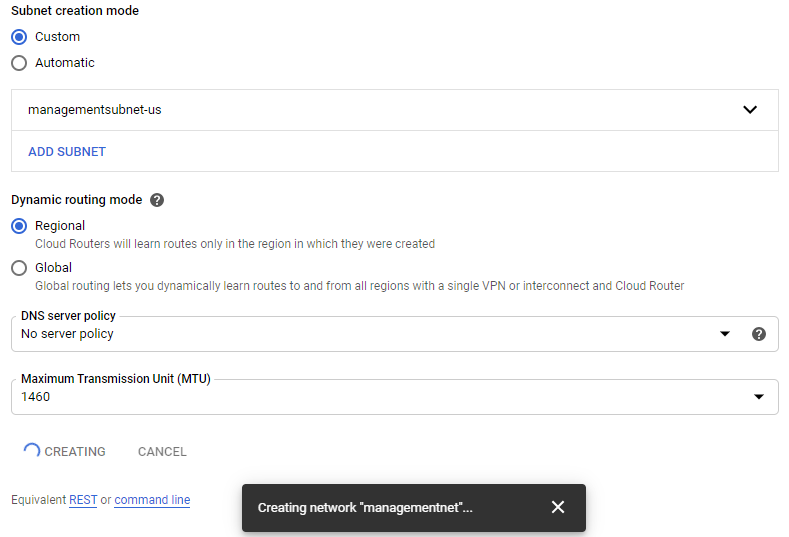
**Default is default vpc network provided by gcp while mynetwork is configure for this lab.**



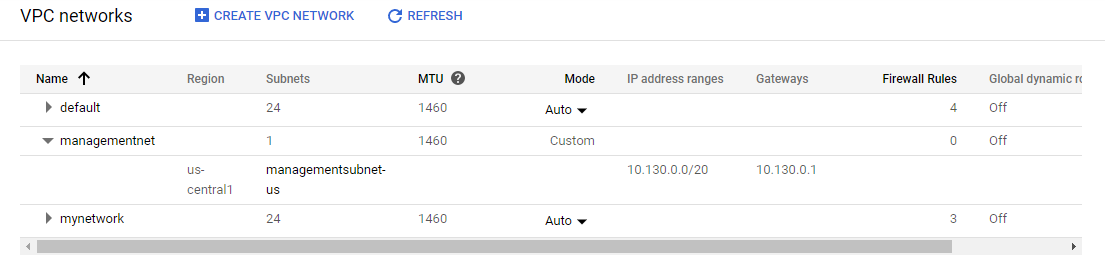
**Creating new VPC network with name “managementnet”**



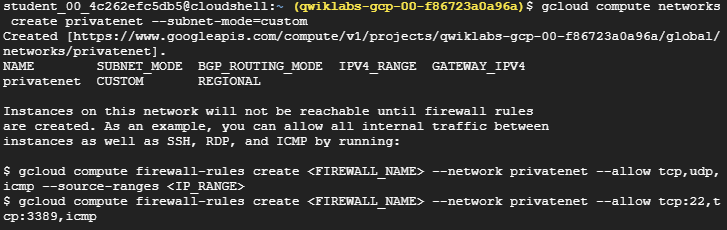
**Besided create button UI provide a gcloud command line sample syntax if we want to replicate exact configure VPC network through cloud shell**



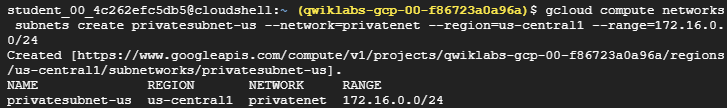
**After setting up require configuration network start creating**



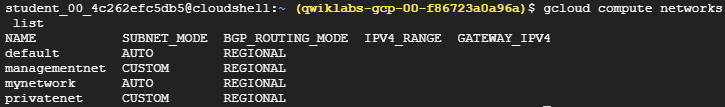
**Managmenentent is now created and visualize in VPC dashboard UI**



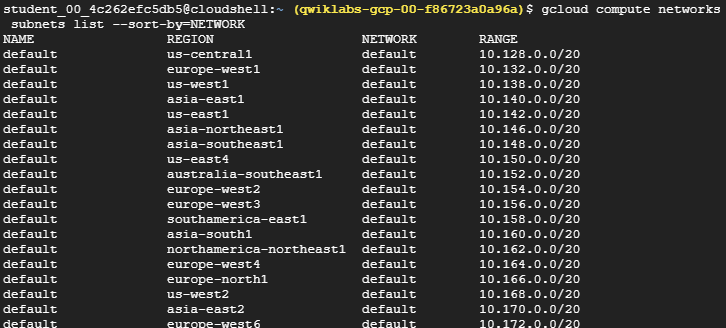
**Now creating another VPC network through cloud shell with name “privatenet”**



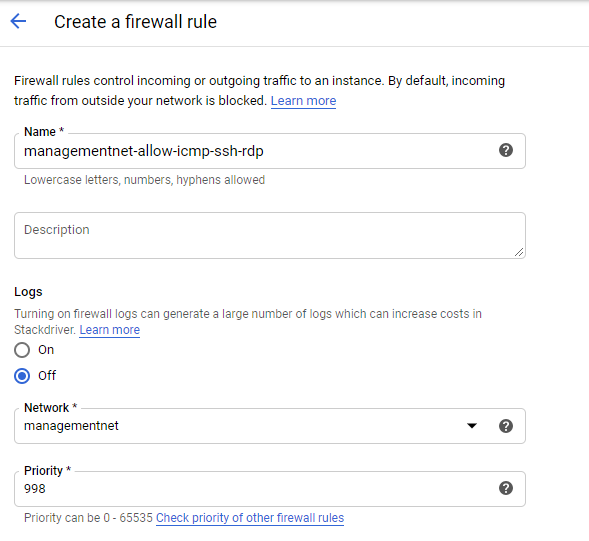
**Configure privatenet by creating subnets and assigning zone and region thorough command line**



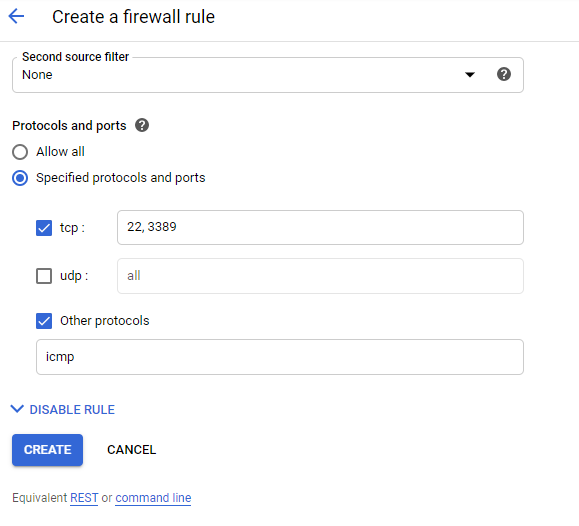
**Listing out all VPC networks**



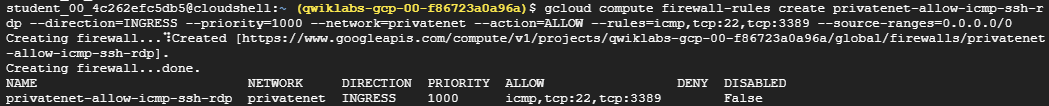
**List out all available network subnets**



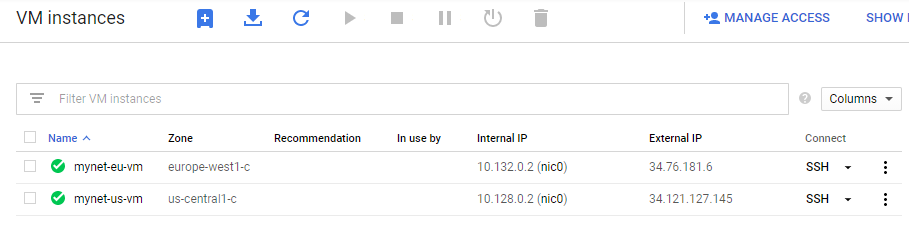
**Creating new firwall rules assigning with managementnet**



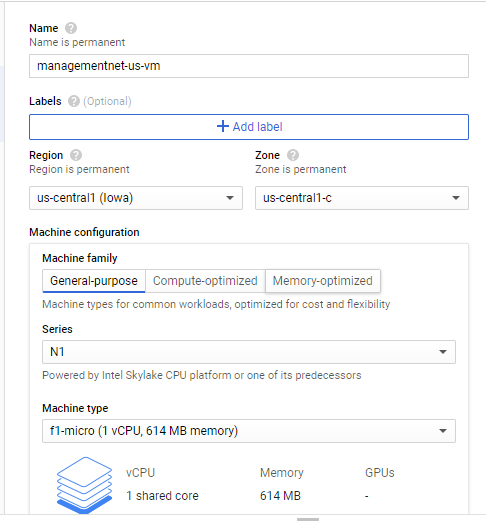
**Configure TCP ports and in other protocols like ICMP**



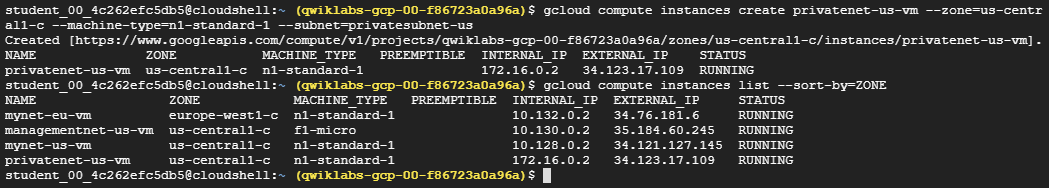
**Creating firwall rules for privatenet from command line**



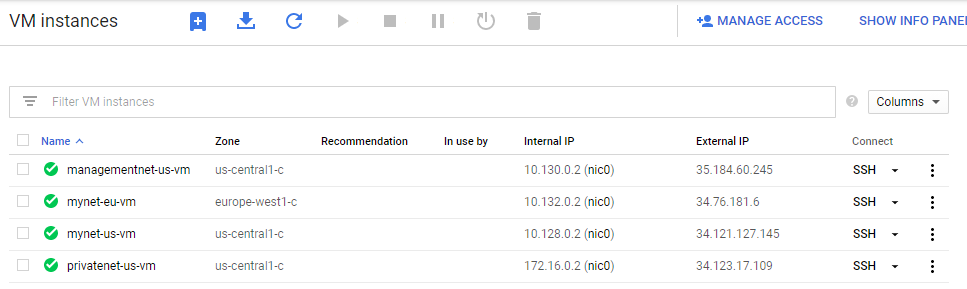
**Two vm instances are running**



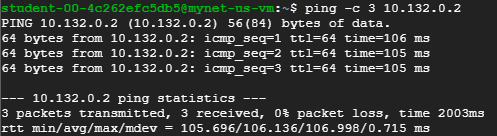
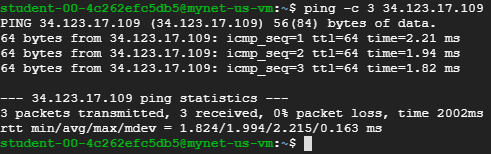
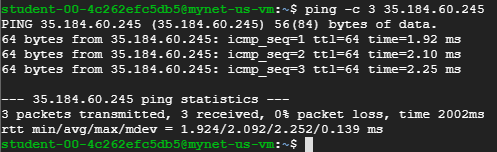
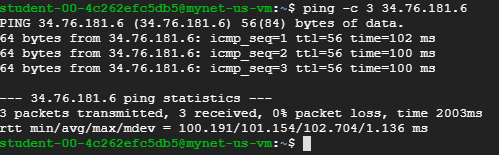
**Creating new VM instance with name managementnet-us-vm**

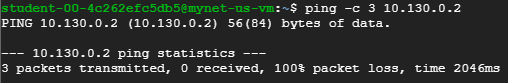


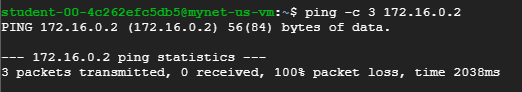
**Similarly create another vm instance from command line with name privatenet-us-vm**



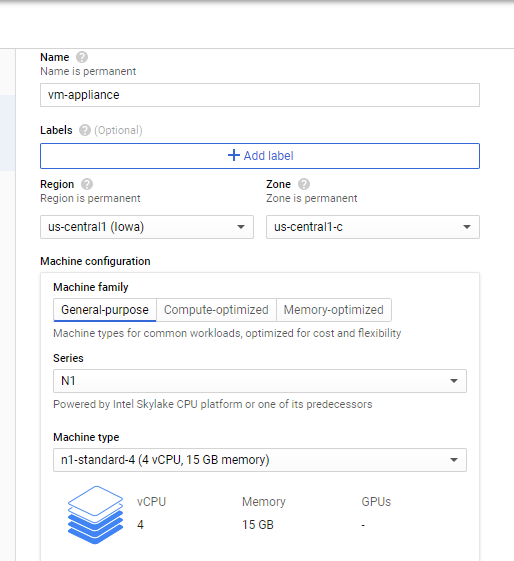
**All 4 instances are up and running.**



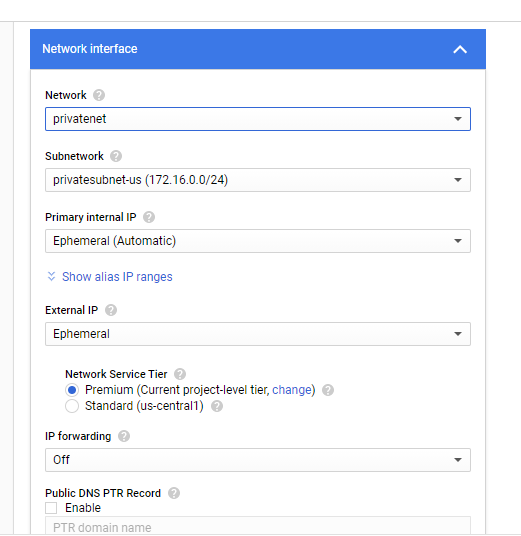




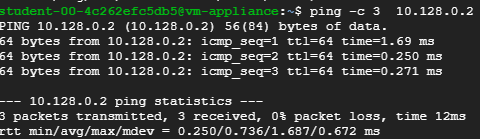
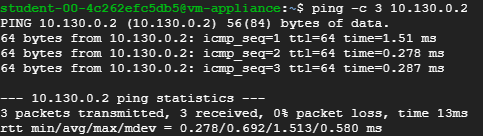
**Now our main task is to ping different IP including both internal as well as external from privatenet instance. Now observe the further snapshots and note the different IP for each instance to understand pinging with instance for a particular ss.**



**Creating one more instance with name VM appliances**

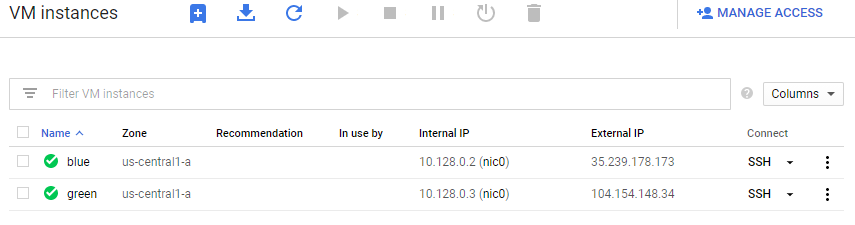


**Configure network as privatenet**

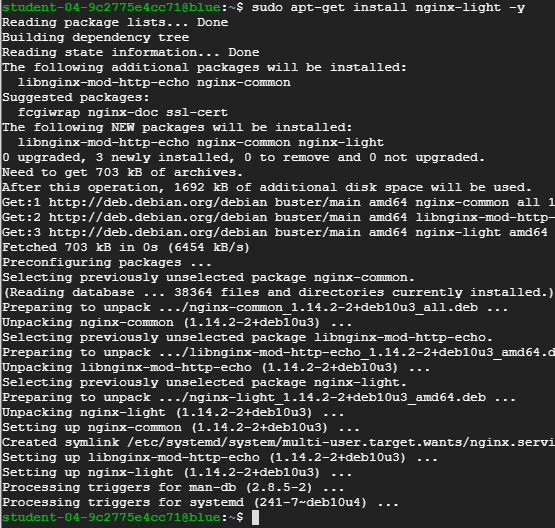


**At this time no packet loss**

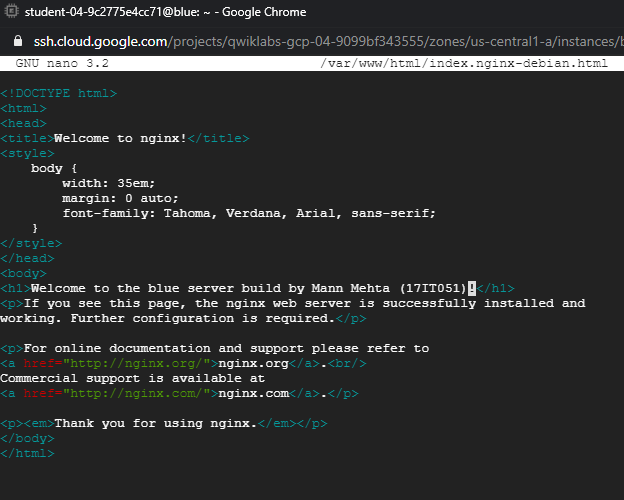
**3.3 - VPC Networks - Controlling Access**



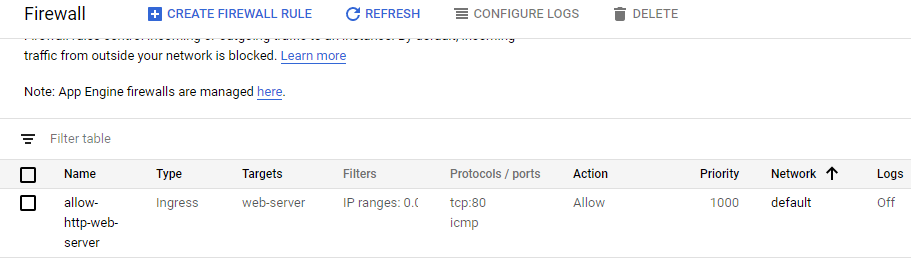
**Created two vm instances blue and green**



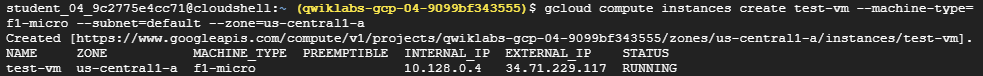
**Installed nginx-light in both the instances**

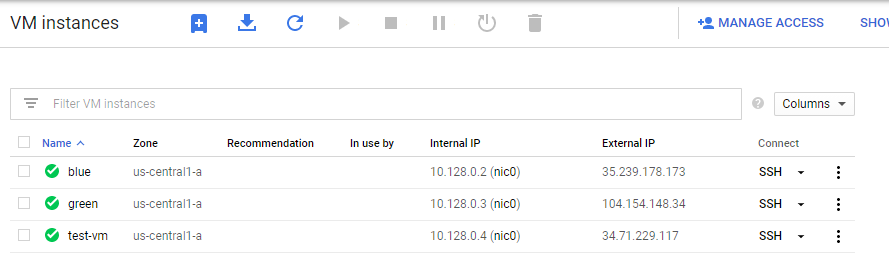


**Editing default HTML pages in both the instances in one typed blue and in other typed green**

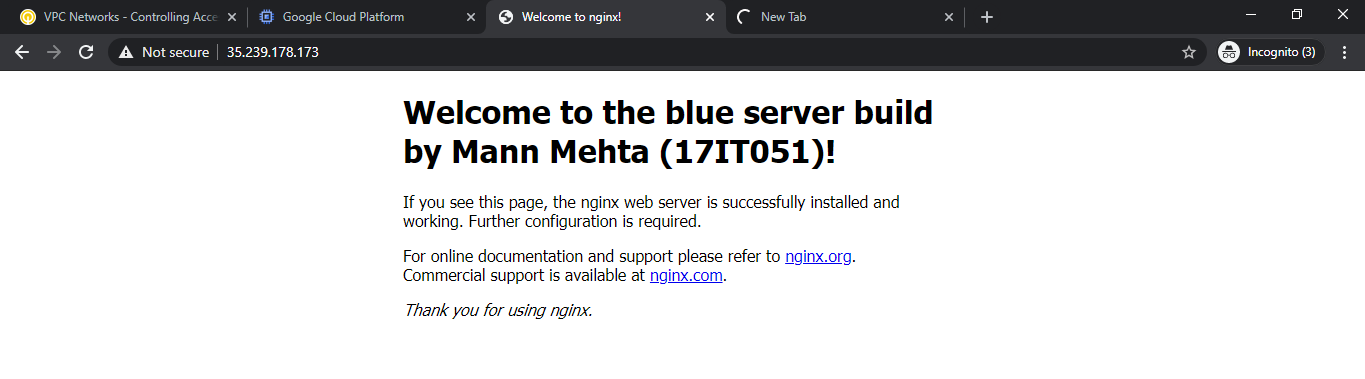


**Create new firewall rule**

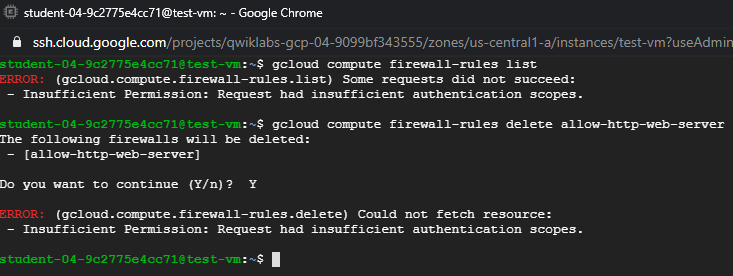




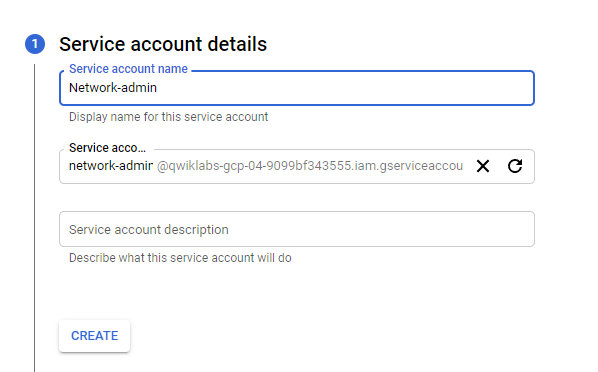
**Create new instance test-vm from cloud shell**



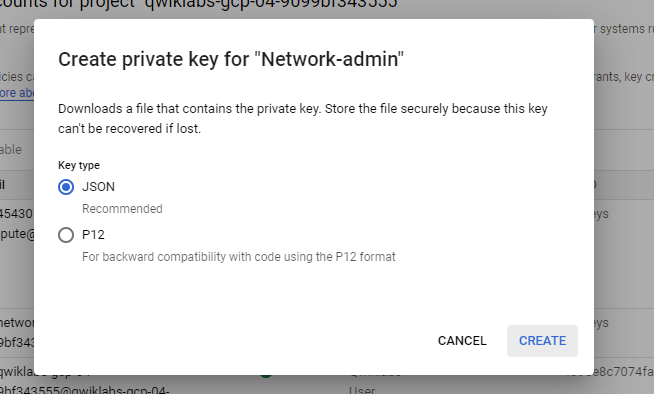
**Able to access blue server from external IP**



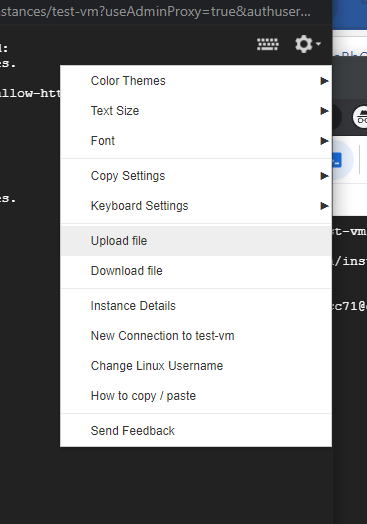
**Compute engine default service account does not have right permissions to allow to list or delete firewall rules.**



**Creating new service account**



**One of the json copy of private key saved to local computer need to upload on cloud shell during authentication**

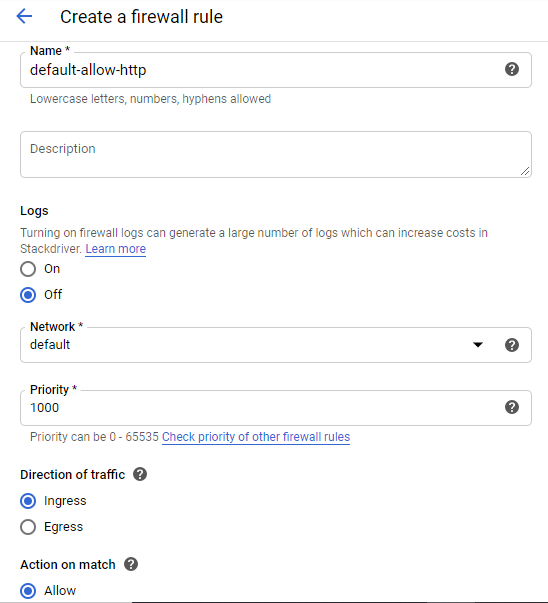


**This is how we upload the private key on cloud shell**

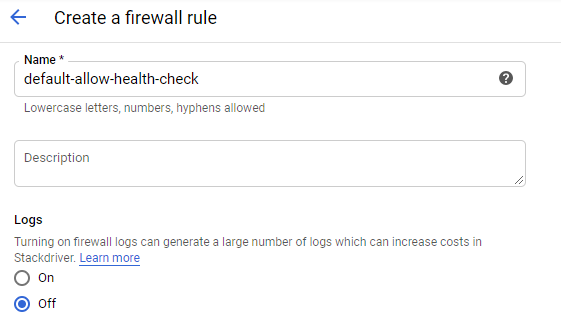


**Hence, we able to authenticate with service account successfully**

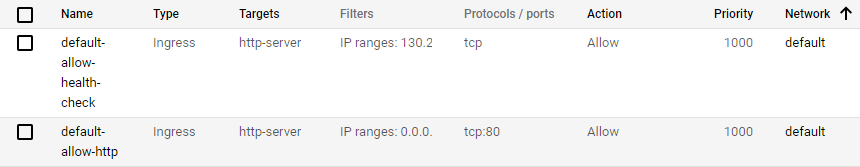
**3.4 - HTTP Load Balancer with Cloud Armor**



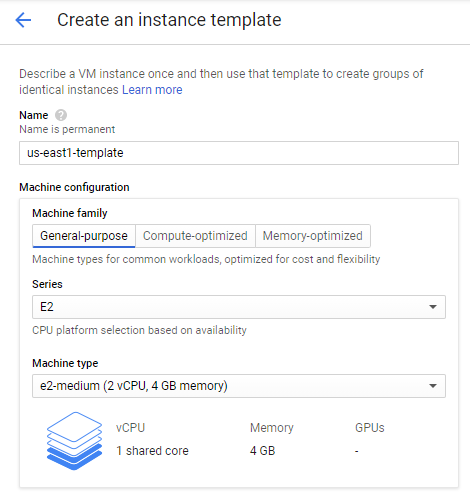
**Creating firewall rule default-allow-http**



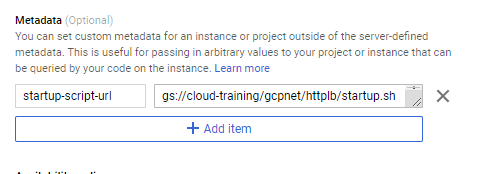
**Creating another firewall rule with default-allow-health-check**



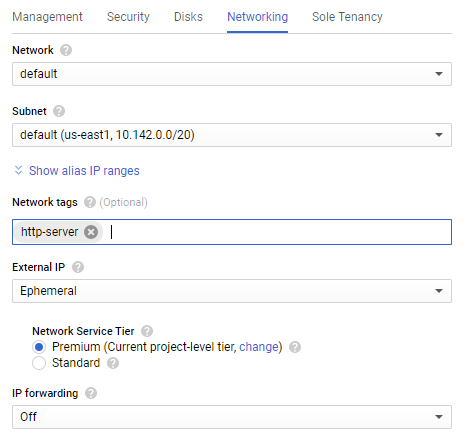
**Firewall rules created successfully**



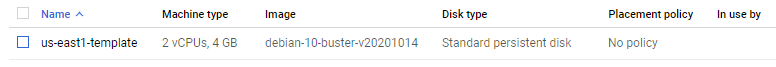
**Creating an instance template**



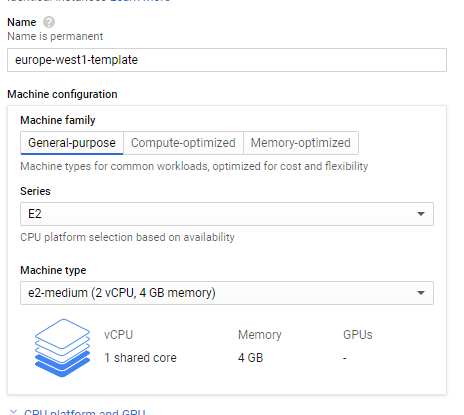
**Providing startup script in meta data**

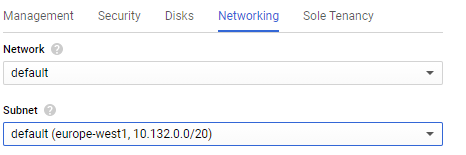


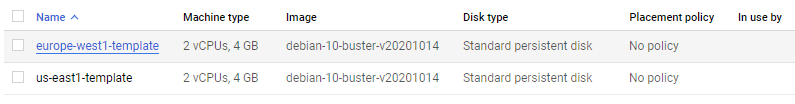
**Providing network tags**



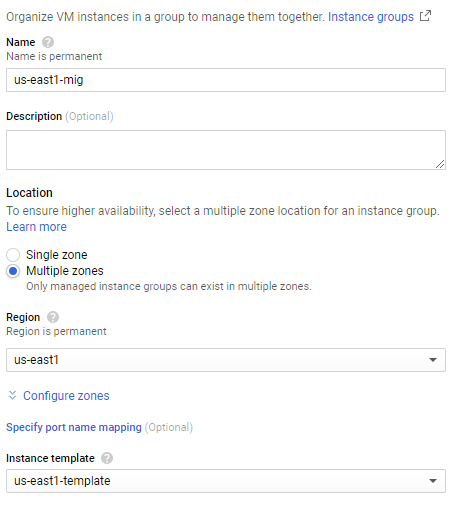
**Template create successfully**



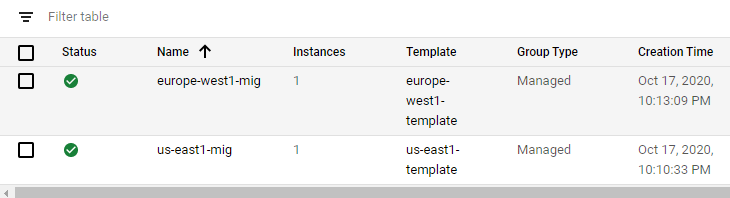




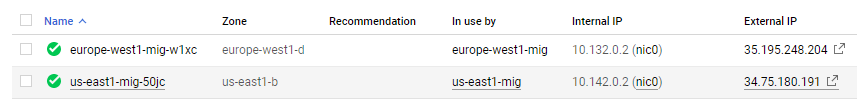
**Similarly created another template**



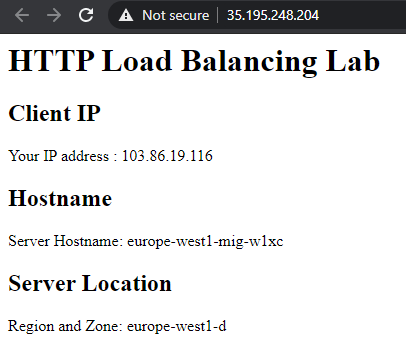
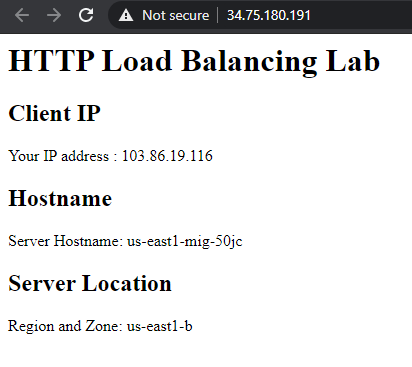
**Creating new instance group**



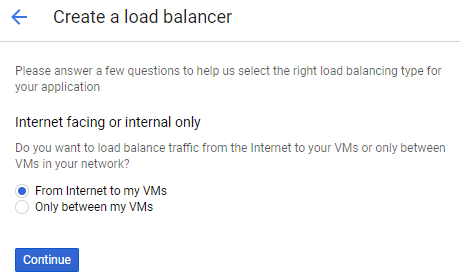
**Similarly created one more instance group**



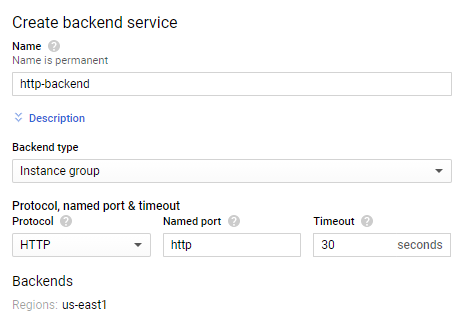
**Now we can see the “In use by” field is now using the templates based on our assignment**

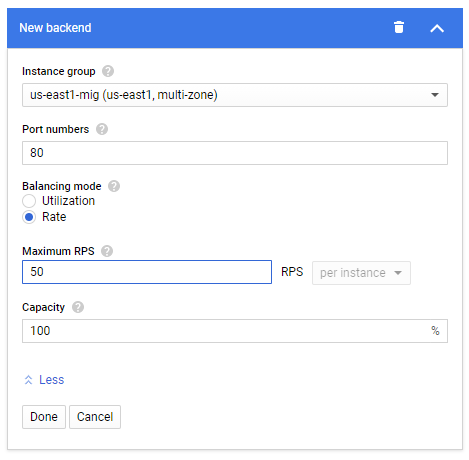


**Accessing both templates external IP address working fine**

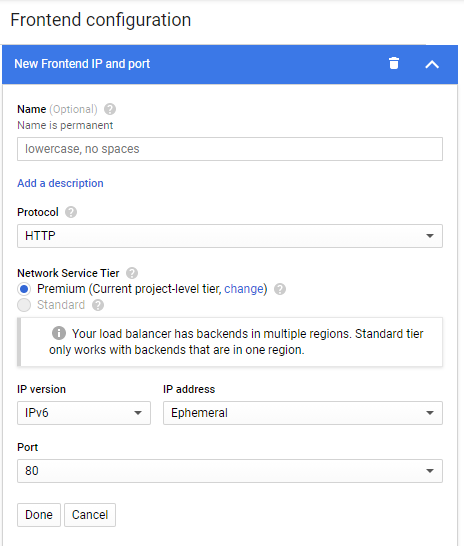


**Creating Load Balancer**

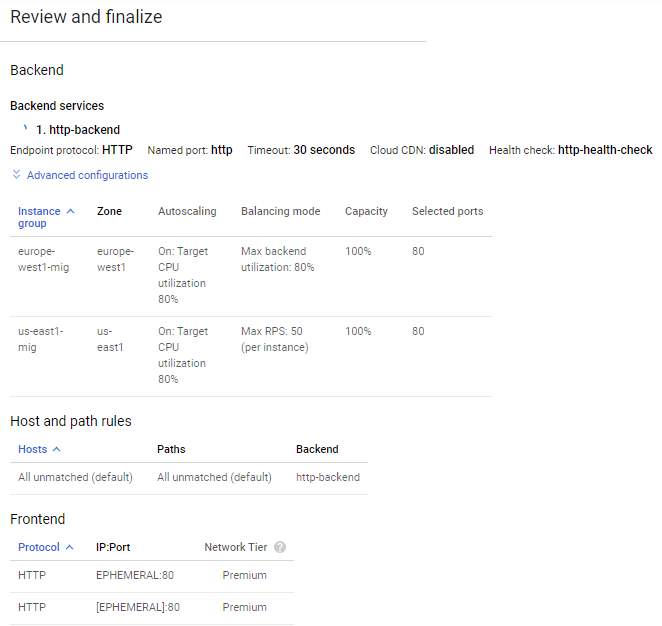




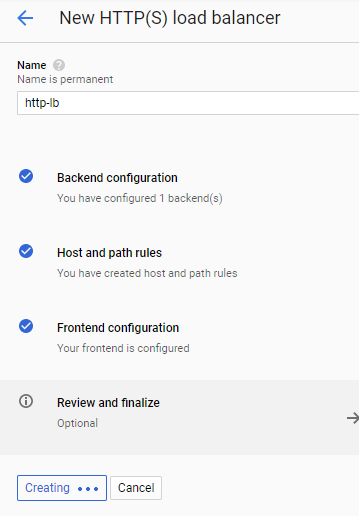
**Creating backend services**

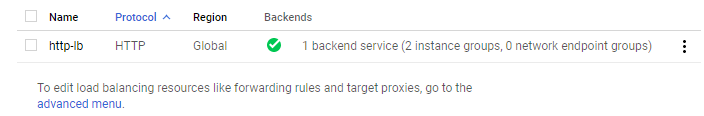


**Setting up frontend IP and Port**

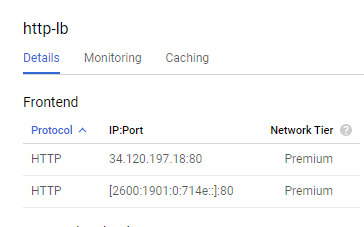


**Review and finalize use to verify all the configuration set up correctly or not**

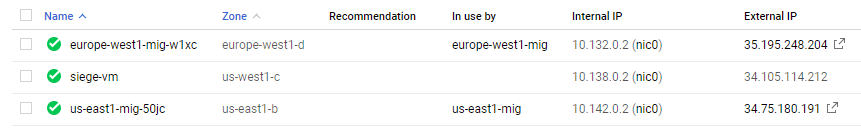




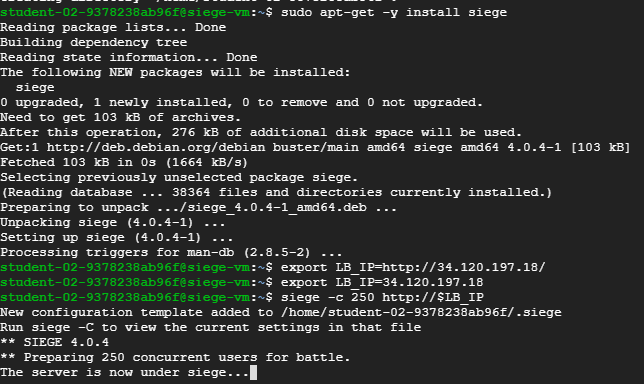
**Creating new http load balancer with name http-lb**



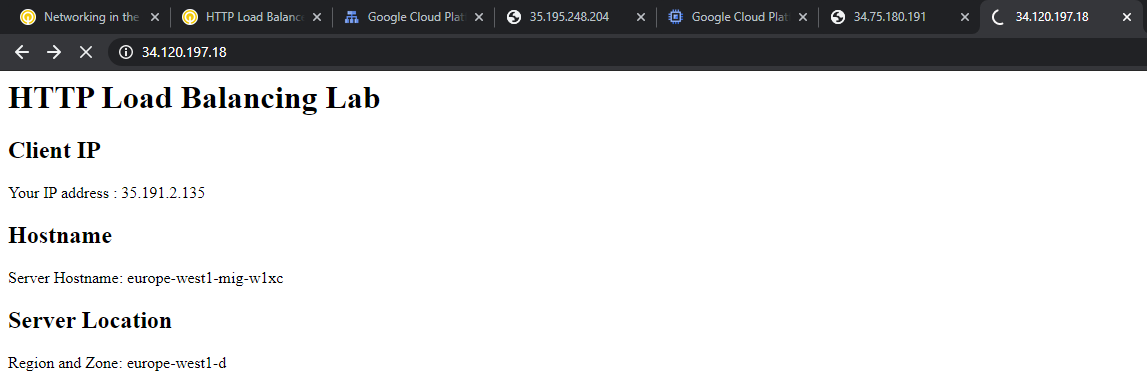
**We can monitor IPV4 and IPV6 of http-lb hence http load balancer forward traffic to the region which is closer to us**



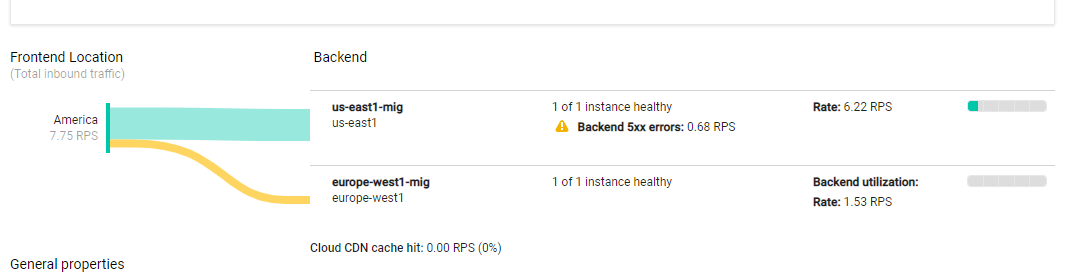
**Creating another Instance**



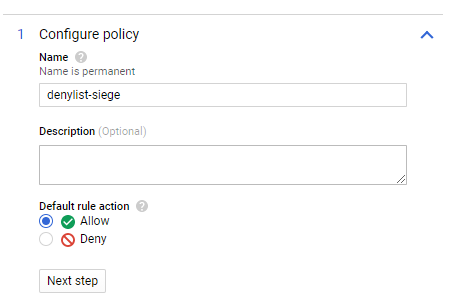
**Using ssh to install siege**

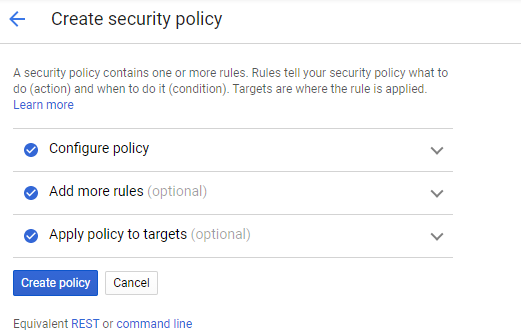


**Able to access external IP address**

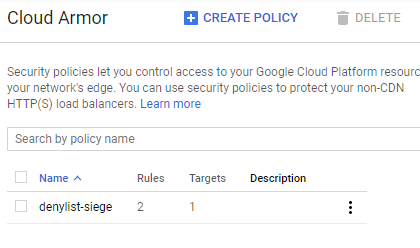


**We can analyze the forwarding done by load balancer**

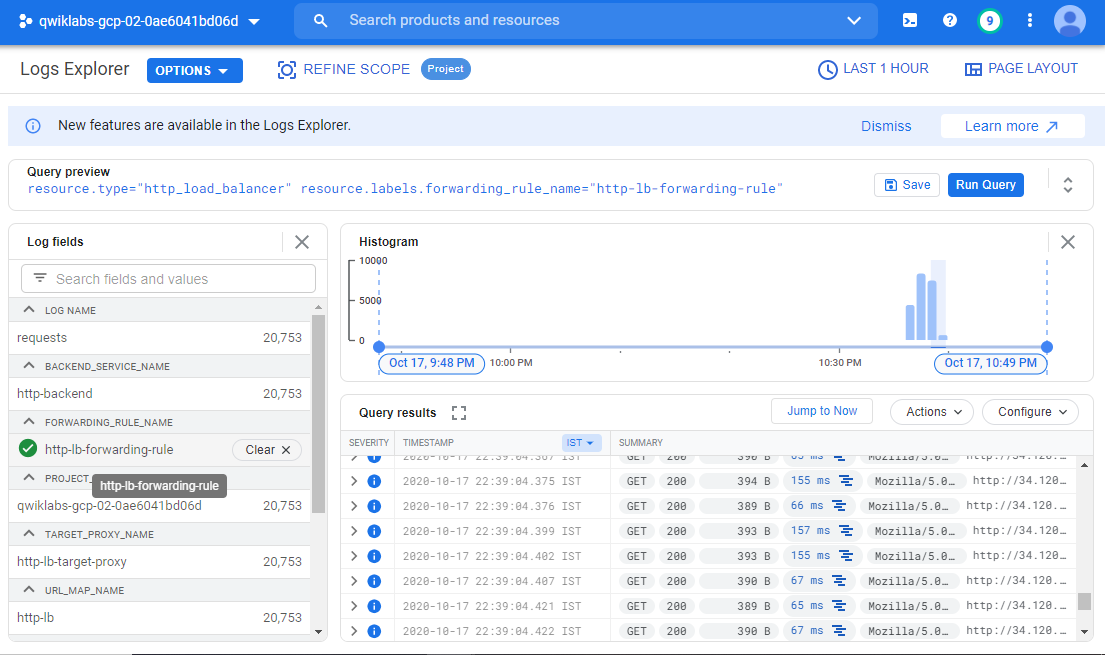


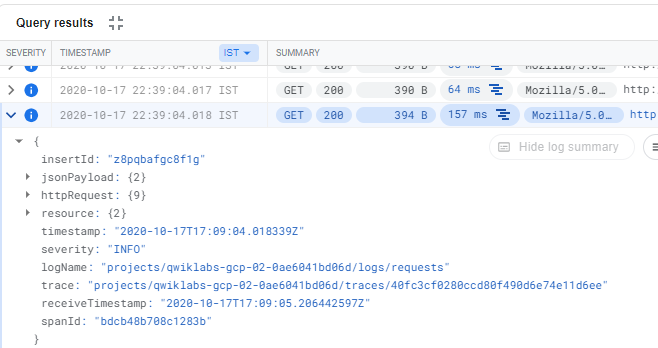


**Using cloud armor**

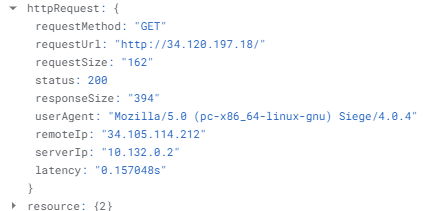


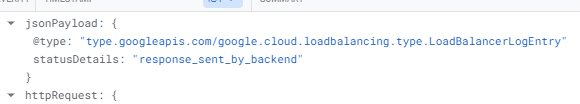
**Policy created successfully**





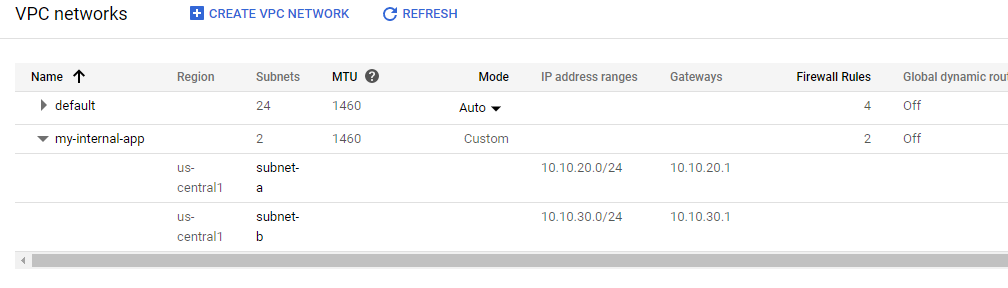
**Monitoring logs**



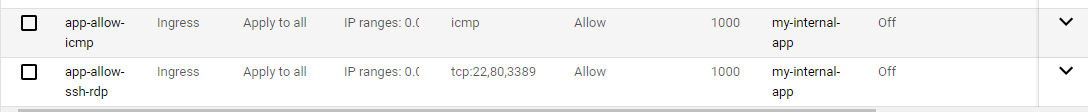


**Monitoring JSON payloads**

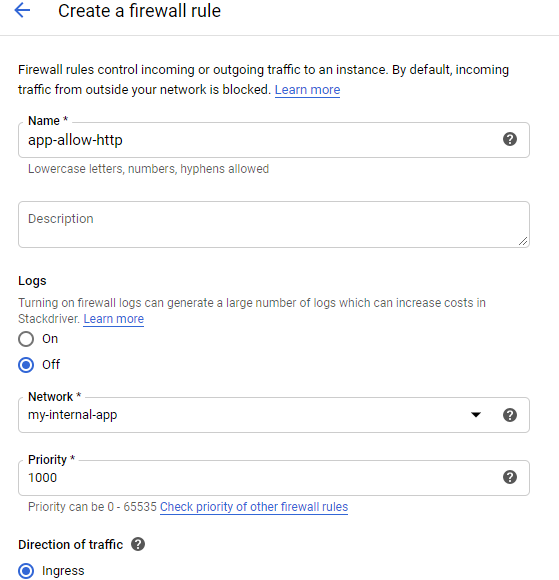
**3.5 - Create an Internal Load Balancer**



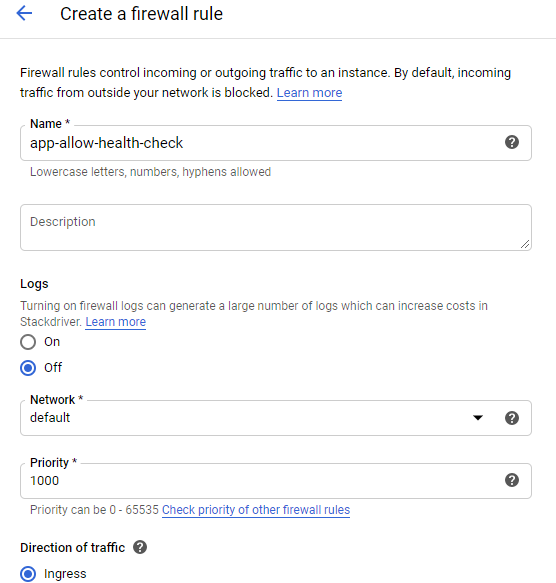
**By default my-internal-app instance created with two subnets a and b**



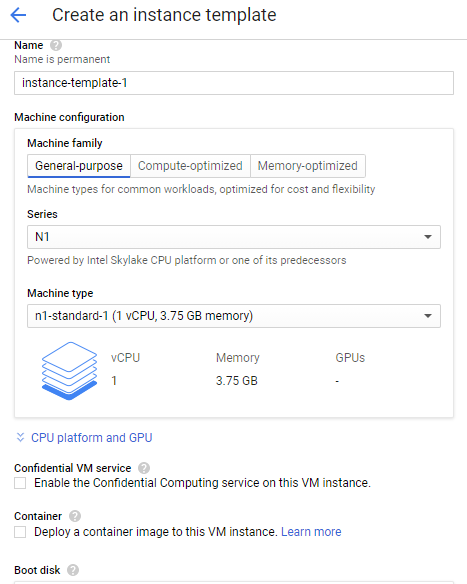
**Similarly, two firewall rules**

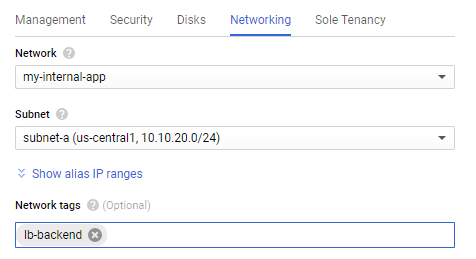


**Creating new firewall rule**

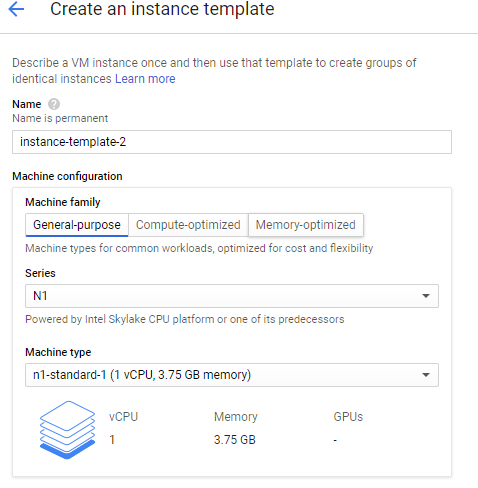


**Creating one more firewall rule**

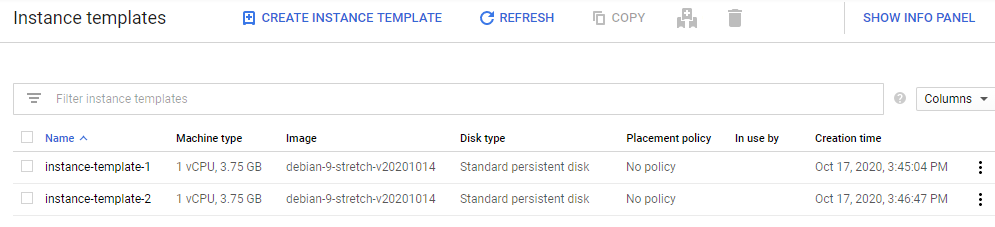




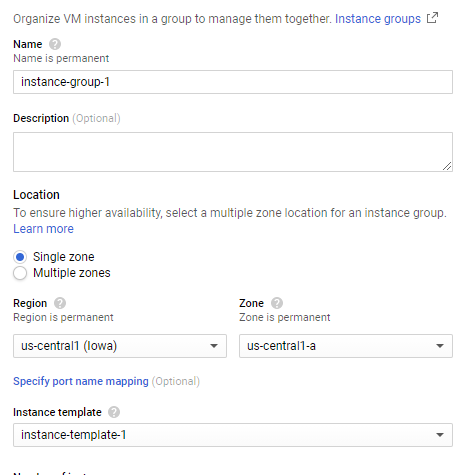
**Creating instance template with network tags**

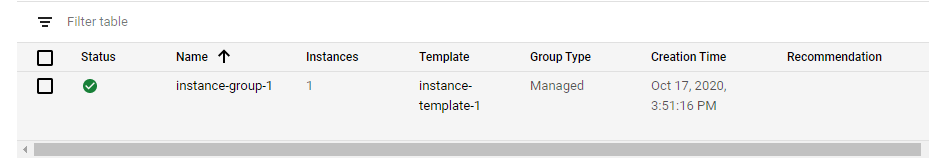


**Creating another template**

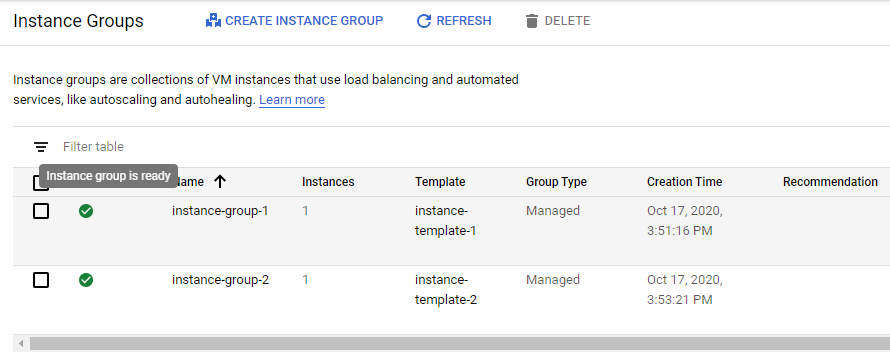


**Finally two firewall rules and two instance templates created**

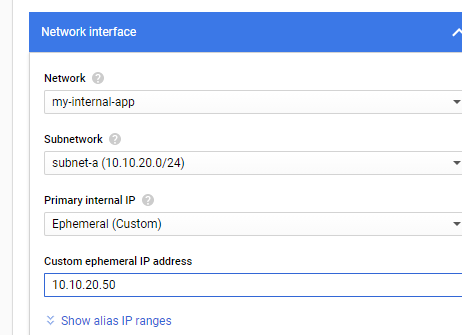


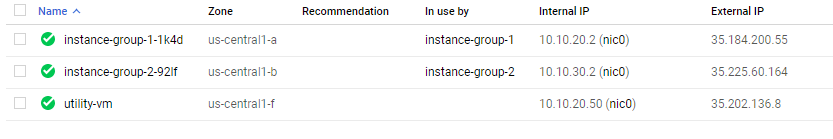


**Creating instance group**

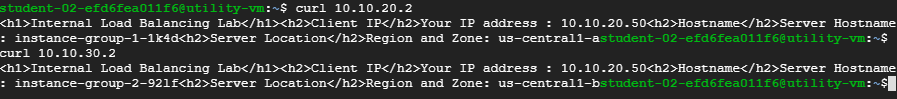


**Creating another instance group**

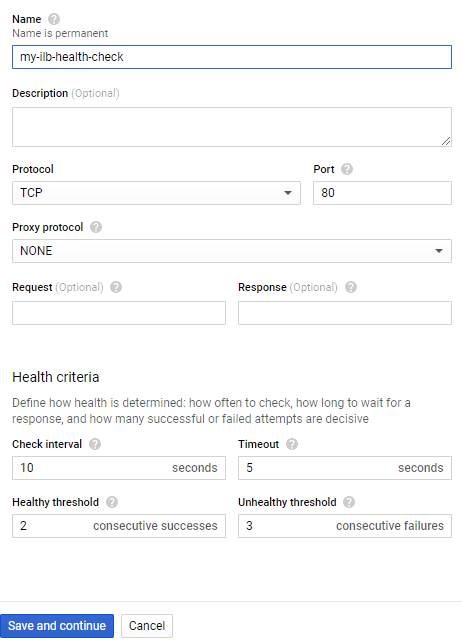




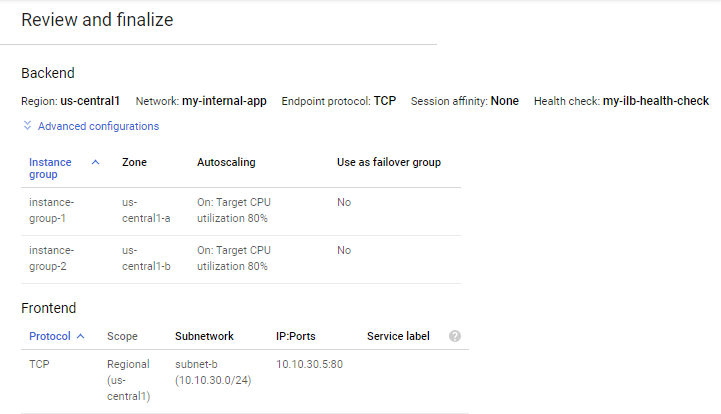
**Configured the network interface**



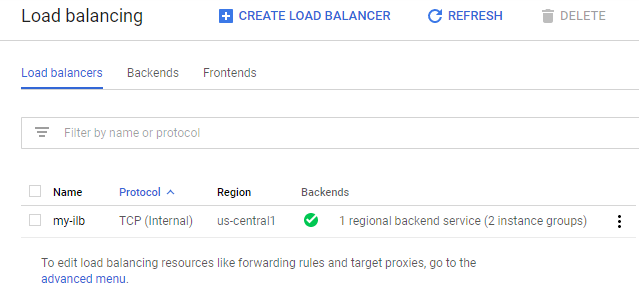
**Culr to insternal IP of instance group 1**



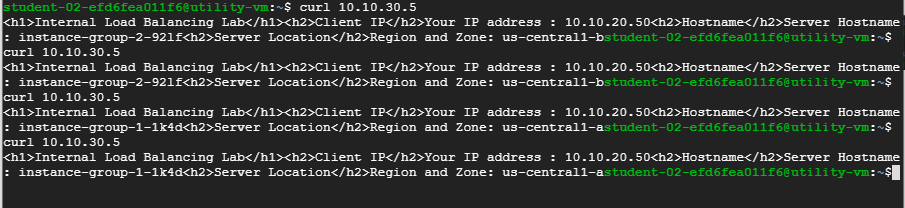
**Creating load balancer**



**Review and finalizing stage**



**Load balancer created successfully**



**We can see the it will select the instance group 2 and 1 both based on the load balancer health checks**

**LATEST APPLICATIONS:**

A VPC network provides the following: Provides connectivity for your Compute Engine virtual machine (VM) instances, including Google Kubernetes Engine (GKE) clusters, App Engine flexible environment instances, and other Google Cloud products built on Compute Engine VMs.

Google **Cloud Armor** helps protect your infrastructure and applications from distributed denial-of-service (DDoS) attacks by using Google's global infrastructure and security systems.

Internal Load Balancer offers you the possibility to load balance TCP/UDP traffic without exposing your VMs via a public IP to the Internet. In this lab we will create a public facing web server to serve the result of a simple web application

**LEARNING OUTCOME:**

3.2 - Created custom mode VPC networks with firewall rules, VM instances using compute engine, explore connectivity of VM instances across VM instances across VPC networks hence finally created VPC network with multiple network interfaces.

3.3 – Created nginx server assigned firewall rules the created service account with IAM roles and access permissions of network admin and security admin roles.

3.4 – Created HTTP and health check firewall rules, configure two instance template, two instance group, configure http load balancer with IPV4 and IPV6, stress test on load balancer and finally denylist an IP address to restrict access to a load balancer.

3.5 – In this we configure and test internal load balancer

**REFERENCE:**

1. <https://google.qwiklabs.com/quests/31>
2. <https://cloud.google.com/vpc/docs/multiple-interfaces-concepts#:~:text=Multiple%20network%20interfaces%20enable%20you,have%20an%20external%20IP%20address.&text=Each%20interface%20on%20a%20VM,MTU%20of%20the%20attached%20network>.