1. Setting up http load balancer: Objectives • Create HTTP and health check firewall rules • Configure two instance templates • Create two managed instance groups • Configure an HTTP Load Balancer with IPv4 and IPv6 • Stress test an HTTP Load Balancer

Diagram

Description automatically generated

Let’s discuss some theoretical concepts:

* Clients: clients are defined as the end user of the services provided by the service provider.
* Internet: internet is referred as the inter connections of the devices through a network. They can communicate through the one part of world to another by the networks.
* Anycast IP: refereed as Static or External or Public IP that is linked with the domain name to reach the services that are provided by the service provided.
* Global Forwarding Rule: refereed as the external firewall or rules define in external firewall to route the requests come from Internet and clients for the server global forward rule is responsible for the routing the traffic come from the internet to appropriate proxy server.
* Target Proxy: It provides extra layer of security to the servers that provides different service to client. It is used to achieve the anonymity of the server to the outer world it is impersonate as server to outer world or end user.
* URL Map: it defined as the mapping of internal service IP according to the request made by end user. For example when users from different regions are trying to connect to [www.google.com](http://www.google.com) , when request come to the URL Map then might be internal URL point the google server according to the regions and point to the Indian server in my case [www.google.in](http://www.google.in) and [www.google.uk](http://www.google.uk) for Europe reason.
* Backend Service: referrer to a cloud computing service model that serves as the middleware that provides developers with ways to connect their Web and mobile applications to cloud services via application programming interfaces (API) and software developers' kits (SDK).
* Health Check: is a mechanisms to check the status and errors of the instances who provides the services so that the high arability and throughput can be achieve.

Let’s started with creating firewall rules: Navigate through Menu->VPC networking->Firewall. Then go for CREATE FIREWALL RULE and 1st rule name it as “default-allow-http” set network default and in protocol and ports add TCP port for http traffic. Then create 2nd rule for health check and name is “default-allow-health-check”, network is default, network tag should be “http-server” filter by “IP ranges” and ranges should be “130.211.0.0/22, 35.191.0.0/16” followed by space between IP and check the TCP box in Protocol and Port field. Create these both rules and our first objective is completed. Graphical user interface, application, Word

Description automatically generatedGraphical user interface, text, application, Word

Description automatically generated

Move to second objective that is configuring Templates. So Navigate through Menu->Compute engine->Instance templates then click to create templete and add following configure and remaiing leave it to default so name should be “us-east1-templete” for the series we can chose accordengly our use but here lets chose “N1” leave all fields defult go to “Management” and under Metadata add “key=startup-script-url” and “value=gs://cloud-training/gcpnet/httplb/startup.sh” it loads the script fro the URL and that script is for installing the apache2 and php and removing defult html pages and add some custom PHP page. Hten move to networking and network should be defult and subnet is “us-east1” and network tag should be “http-server” so that the firewall rule that we created “allow-http-trafic” and “http-health-check” should be applied to the templete and create the templete. For the second instance just click to the 1st instance and copy the same instance then just edit the copy instanced network and change it to “europe-west1” nad remmaning shoud be same and name of the instance shold be “europe-west1”. Lets quickly varify is instance templetes created or notGraphical user interface, text, application, email

Description automatically generatedquickly see the configuration of each machinesGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, Word, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, Word, email

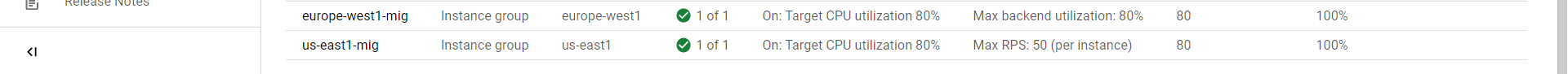
Description automatically generated

Now we must move to 3rd objective of the task and that is creating managed group for both the templates so that auto scale is performed when it need with same configuration. Let’s see how we can do Navigate to Menu->compute engine->instance template and select a template to create a Instance group and do the following configuration and leave all other with default values. So config is “name=us-east1-mig, Location=Multiple zones, Region=us-east1, Instance template=us-east1-template, Autoscaling > Autoscaling metrics > Click Pencil icon> Metric type=CPU utilization, Target CPU utilization=80, Minimum instance=1 maximum instance=5, Cool-down period=5” and create the instance group. Repeat same steps for another template and finally check the group instance that will up and run shown in Instance running tab. Graphical user interface, application

Description automatically generated

We have finished 3 tasks now move to Task 4. In Task 4 we have to create load balancer so quickly Navigate through Menu->Network services->Load balancing. Click to create Load Balancer and select HTTP’S Load Balancing and go to START CONFRIGUTION. Do following config and remainnig leave with defaults. Config is “Name=http-lb” and click to Backend Confrigation and select Backend services and Backend Bucket and select backend serivces and click to create a backend servics. Now set the configuration of backend services “Name=http-backend, Instance group=us-east1-mig, Port numbers=80, Balancing mode=Rate, Maximum RPS=50, Capacity=100” then click to add and done then add other backend service for europe region “Instance group=europe-west1-mig, Port numbers=80, Balancing mode=Utilization, Maximum backend utilization=80, Capacity=100” and select done.

Now chose to health check and create health check and set following values and remainning leave it to defaults “name=http-health-check, Protocol=TCP,Port=80” save and continue now configure Frontend confrigation “Protocol=HTTP,IP Version=IPv4,IP address=Ephemeral,Port=80” save it and Done now configure Frontend IP and Port set the following values “Protocol=HTTP,IP version=IPv4,IP address=Ephemeral,PORT=80” and DONE finally review the full configuration and CREATE. Now Test the HTTP Load Balancer by putting the <http://LB_IP> in browser. We can find the Load Balancer IP by clicking the load balancer. Graphical user interface, text, application, email

Description automatically generated

Now we have final task to do that is Stress test of HTTP load balancer. Steps are following :

* Create a Virtual mechine with minimal requirement in any zome with defult network configuration.
* Go to ssh console of newely created VM and update the VM by sudo apt-get update
* Install the siege by command sudo apt-get install -y siege.
* Set envoirment variable for load balance IP by export LB\_IP=34.111.215.148
* And go for siege by siege -c 250 $LB\_IP.

Now go to cloud console and navigate through Menu->Network Service->load balance ->http-lb->monitoring->http-backend now we can see the graphical representation of the traffic coming to different VM in different regions. A picture containing chart

Description automatically generatedGraphical user interface, diagram, Word

Description automatically generated

PTO