**Challenge #1**

**A 3-tier environment is a common setup. Use a tool of your choosing/familiarity create these**

**resources on a cloud environment (GCP Cloud). Please remember we will not be judged on the**

**outcome but more focusing on the approach, style and reproducibility.**

**Step1:**  
 We need to create GCP account. Under that we need to create GCP project.

**Note:** To access any resources in GCP we need to enable to API.

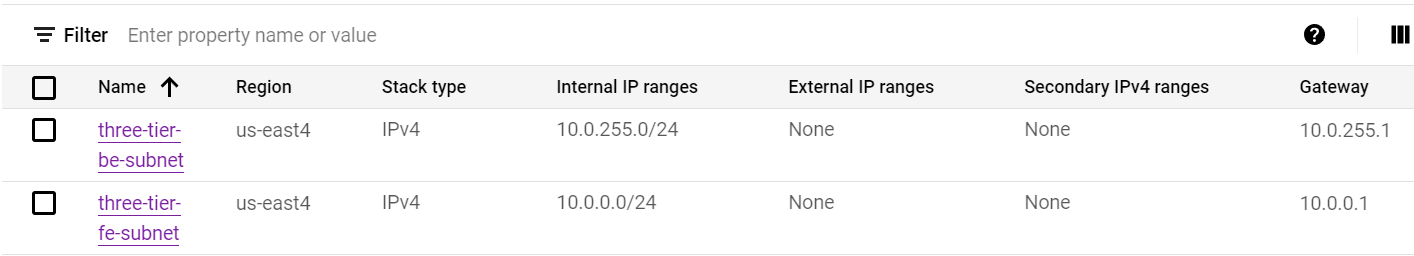
**Step2:**

Create the service account or group under IAM.

Grant the access to group/service account allow to create resources in GCP.

**Step3:**

1. Create a VPC and three Subnet using below steps.
2. In the Google Cloud console, go to the VPC networks page.
3. Go to VPC networks 🡪 Click Create VPC network🡪Enter a Name for the network.
4. Choose Custom for the Subnet creation mode 🡪 Provide a Name for the subnet, Region. Enter an IP address range.
5. Private Google Access: Choose whether to enable Private Google Access for the subnet when you create it or later by editing it.
6. Click Done.
7. To add more subnets, click Add subnet and repeat the previous steps for two more times.

Below is the example:  
 

The Frontend subnet will host web framework(Django), backend will host backend instance group for Load balancer and db.

**Step4:**

Did the instance setup, For the demo, I am going to host our application in a VM instance,

To create VM, I am going to follow below setps:

google cloud console 🡪 In search bar type "GCE" and click on VM instances 🡪 click on Create Instance

1. Give a proper name to your instance and allow full access to all cloud APIs and then left everything as default.
2. Under the firewall setting section enable the Allow http checkbox.
3. Under Advanced options select Network Interfaces now disable the External IPv4 address.

Click on Create

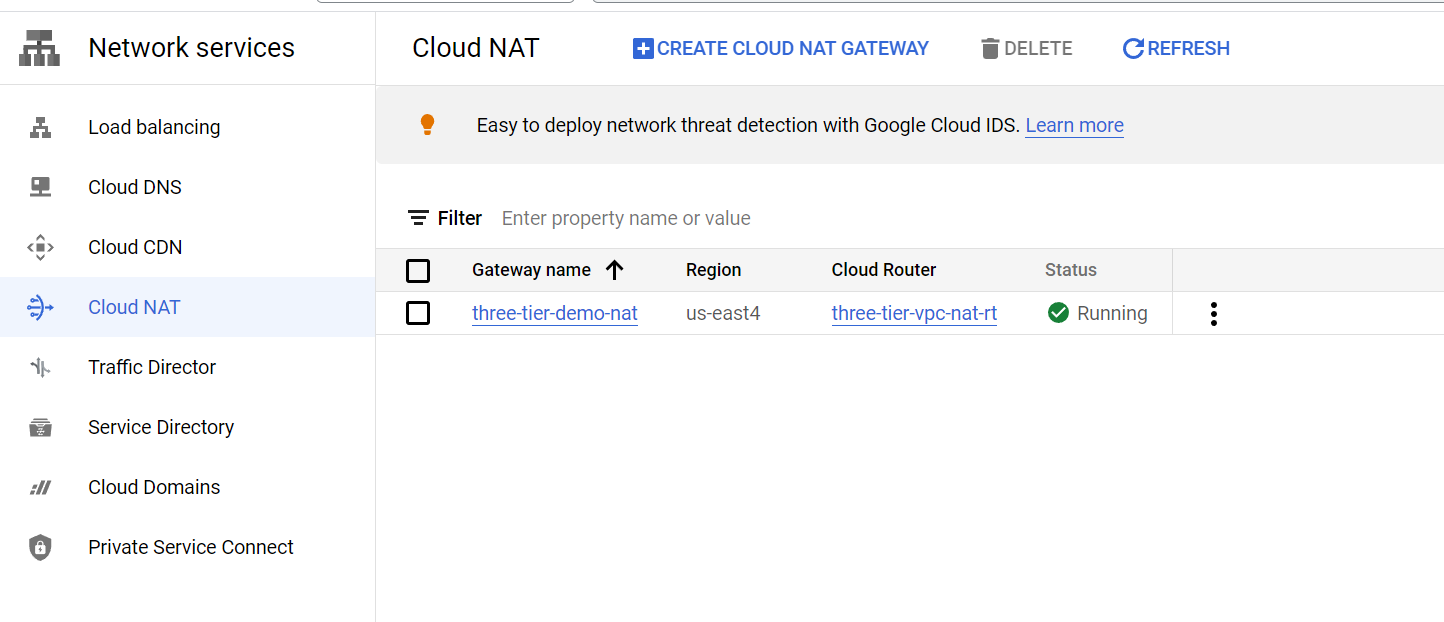
**Note:** Since I have disabled the external ip in our VM, it can’t able to connect to the internet for different purposes like package installation etc. so here we need to configure NAT so that our VM can access internet.

**Step5:**

Go to search bar at the top and search Network services and click on it.

Click on Cloud NAT option on the left of the screen.

1. Enter a Gateway name and then select the network in which our VM is present and select a region us-east4 as in our case.



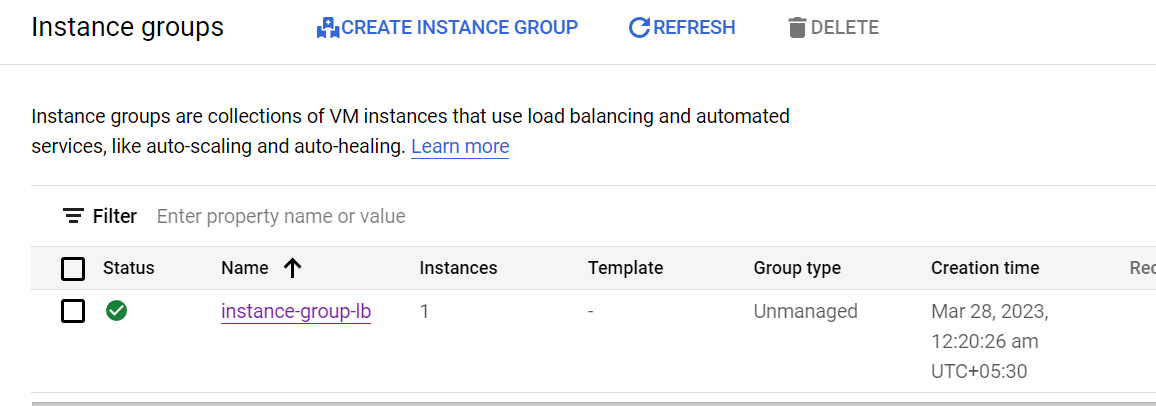
**Step6**:

Integrate the application code inside our VM, You need to install and configure your environment first, such as programming language dependencies and compatibility, before you can work on it. I have used ssh onto the VM use git clone command.

**Step7:**

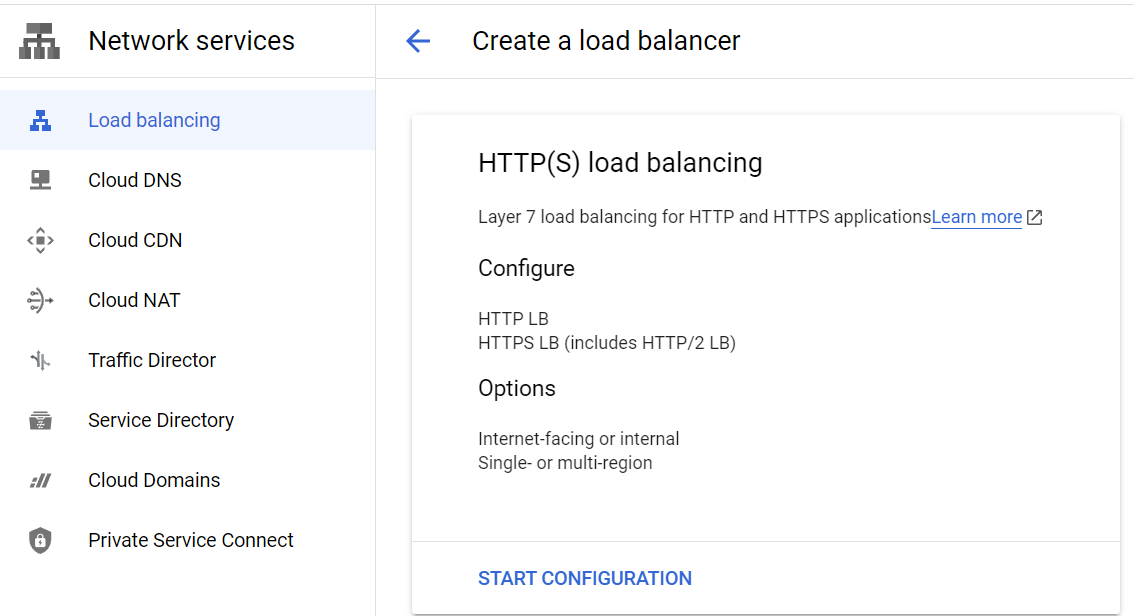
Load Balancer Setup.

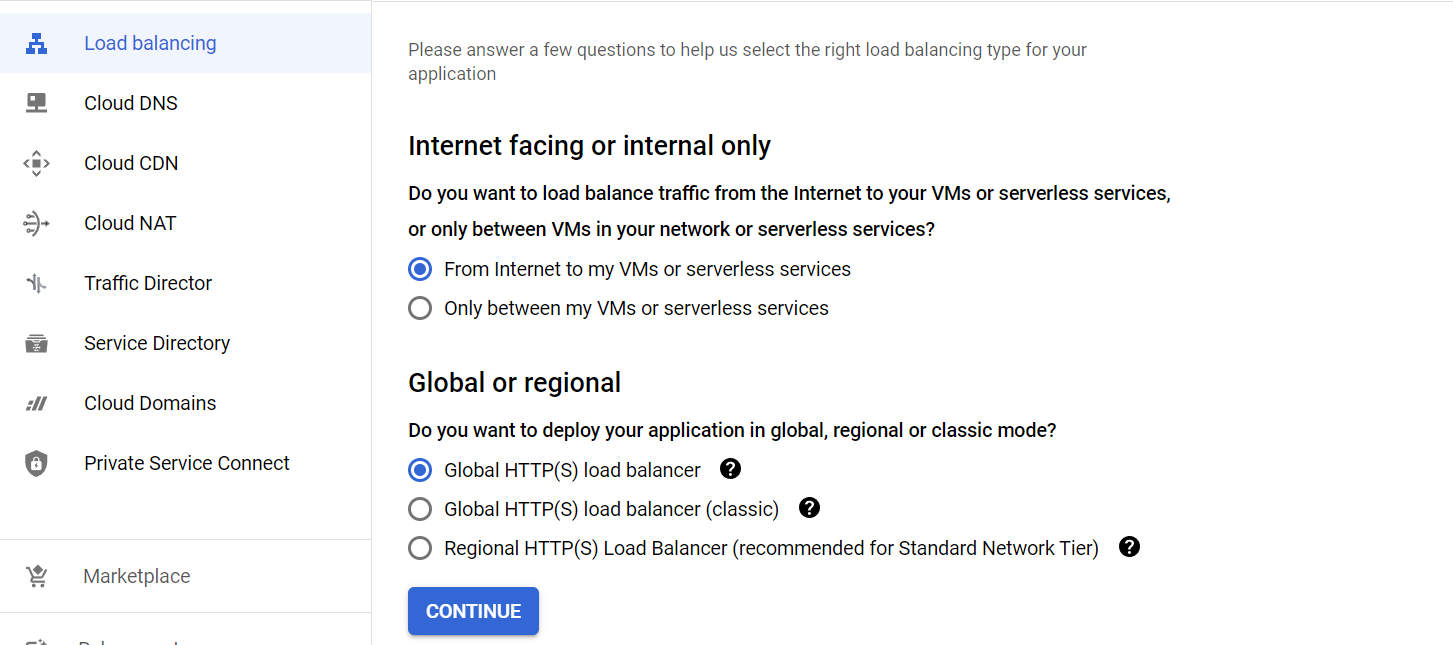
Before going for load balancer setup, I have created unmanaged instance group.

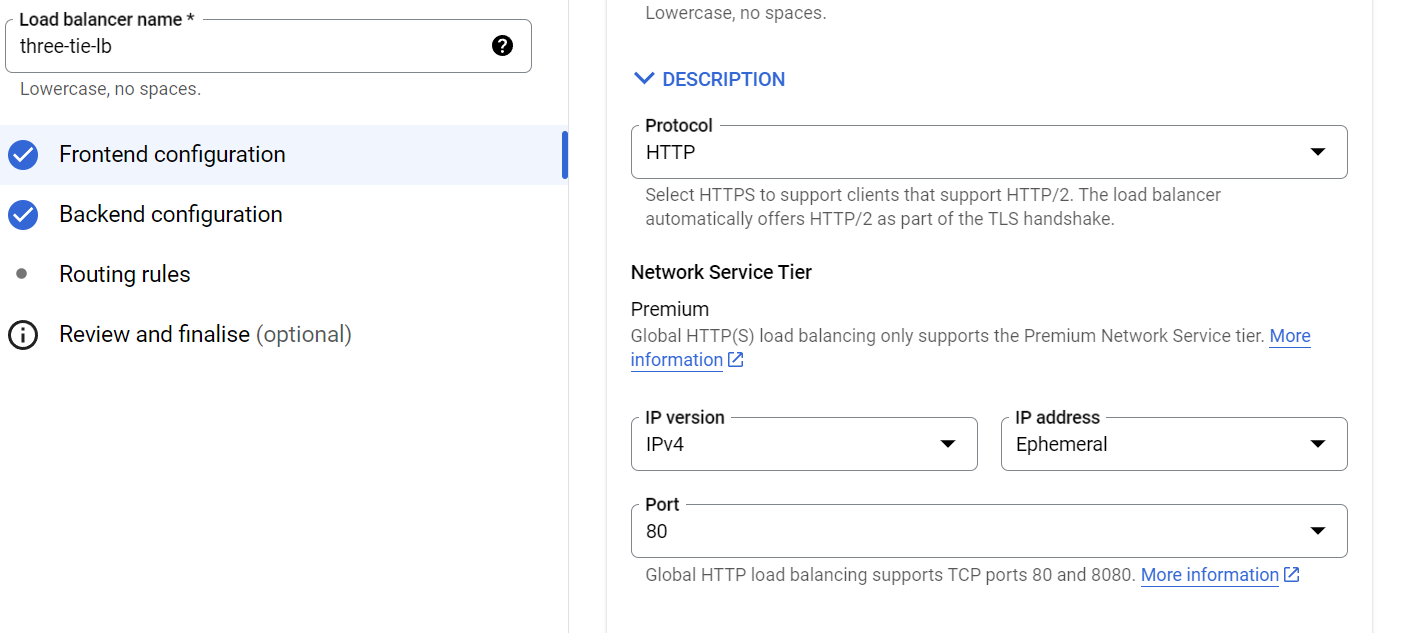


Now I am going to create HTTP Load balancer:

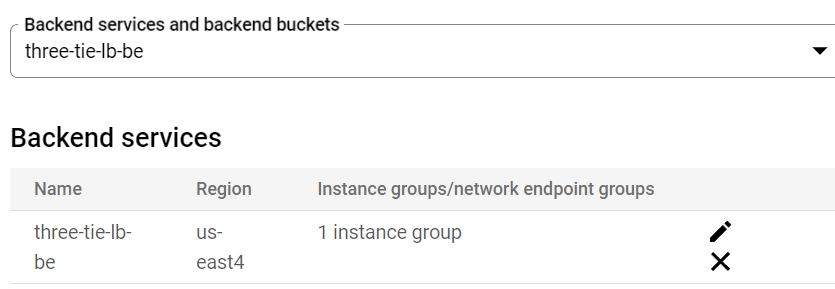
1. Go to top at search bar and type Network Services and click on the it.
2. Click on Load balancing on the left and select Start Configuration.

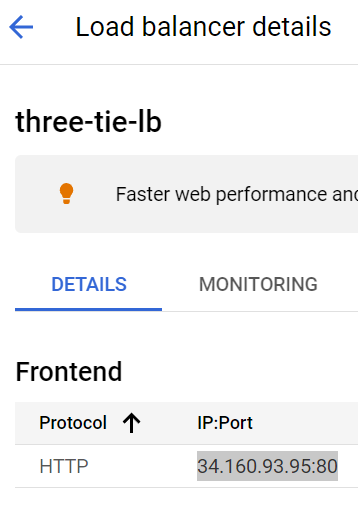






1. create one backend service using unmanaged instance group:



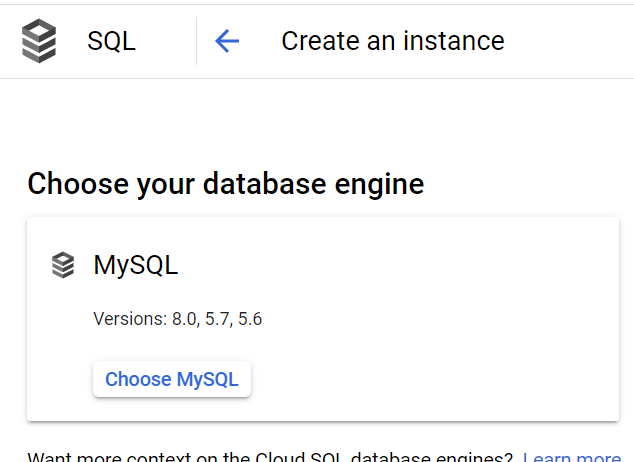
1. Created a load balancer:  
   

**Step8**:

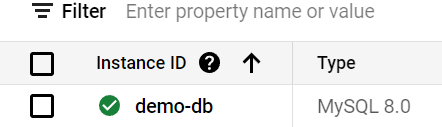
Cloud SQL for storing the data into the database.

Go to the search bar at the top and enter Cloud SQL and select the Cloud SQL service.

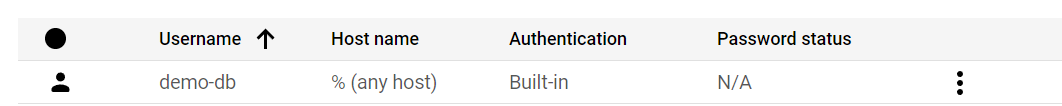
1. Click on Create Instance.
2. Select the MySQL as your database engine



1. Enter the Instance ID of your cloud SQL instance and generate a password.
2. Select the latest database version and also choose Development mode for low cost as it is the need of this project.
3. Go to connection and choose private ip which is a more secure way of accessing the database as it will not have a public ip address and connection from outside are not possible.
4. Click on ADD NETWORK as we want to specify which IP address can access the database.
5. Enter the frontend-IP (application IP) of load balancer as we will post the data using our application.
6. Click on Create Instance.



1. Now I have created a new user to setup the access on db.



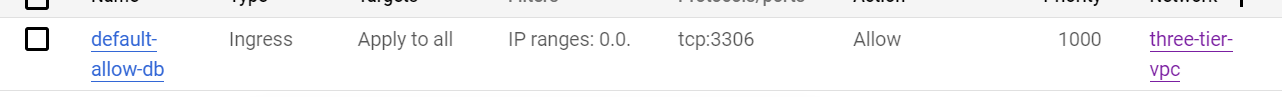
After creating the instance, we have to make sure our ports for connection of SQL instance are open.

**Step9:**

VPC Network Setting

On search bar type VPC Network and go to the VPC network page.

1. Click on firewall.
2. Click on Create Firewall Rule.
3. Enter the name default-allow-db as we will be using Cloud SQL later to connect at this port 3306.
4. Targets: All instances in the network only for the simplicity in the real world it is better to use specified target tags for better security.



**Step10:** I will perform testing with some testcases.

This is how three tier application architecture works in GCP