**Capstone Project 1 documentation**

**Project Title →**

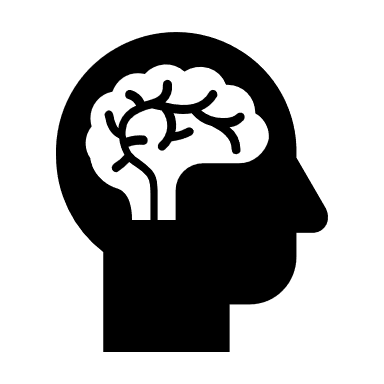
**We are creating and hosting a three-tier web app on GCP.**

**Created By:** → **Sarthak Deshmukh (Cloud Engineer Intern at** [**Nervescape Analytics LLP**](https://www.nervescape.com/)**)**

**Check out my profile →** [**(LinkedIn)**](https://www.linkedin.com/in/sarthak-deshmukh-5756a329a?lipi=urn%3Ali%3Apage%3Ad_flagship3_profile_view_base_contact_details%3BykldYyhYQw2a2gkOrE0a1A%3D%3D)

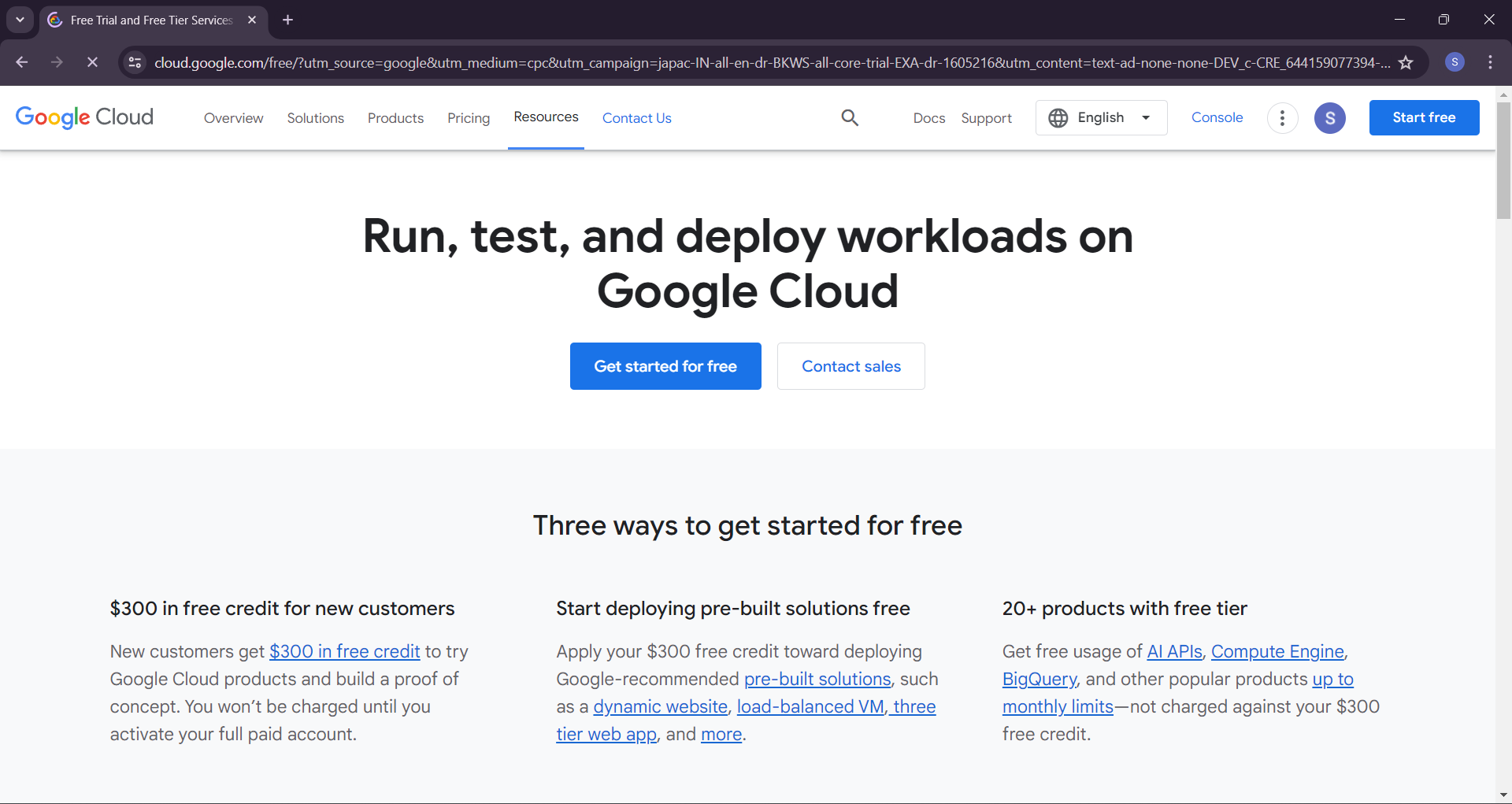
**The following are the given tasks:** →

* **Create VM instances**
* **Configure Bastion server/jump server**
* **Configure NAT/IAP/ROUTES**
* **Connect your VM instance with SSH tools like PUTTY**
* **Configure Web server**
* **Configure VPC/SUBNETS**
* **Configure Load balancer**
* **Configure Autoscaling and auto-healing**
* **Configure DNS and Cloudflare**
* **Configure CDN**
* **Configure SQL**

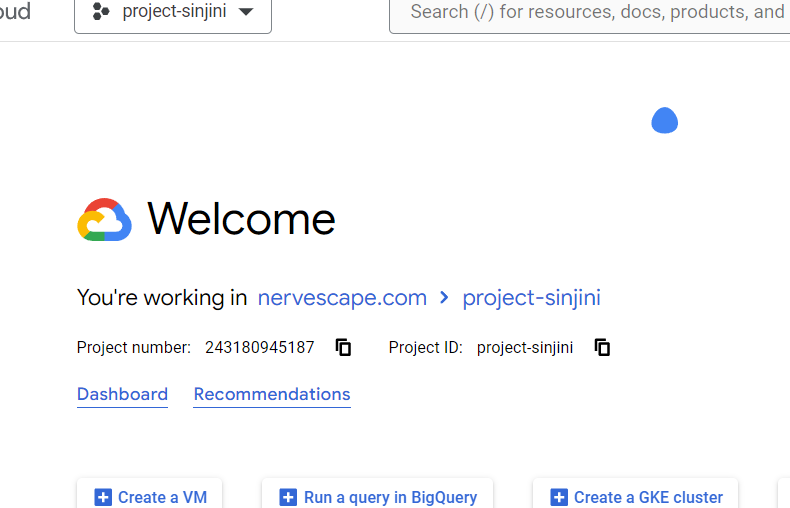
let's have a quick introduction to [**Google Cloud Platform**](https://cloud.google.com/free/?utm_source=google&utm_medium=cpc&utm_campaign=japac-IN-all-en-dr-BKWS-all-core-trial-EXA-dr-1605216&utm_content=text-ad-none-none-DEV_c-CRE_644159077391-ADGP_Hybrid+%7C+BKWS+-+EXA+%7C+Txt+-GCP-General-core+brand-main-KWID_43700074766895889-kwd-87853815&userloc_9154229-network_g&utm_term=KW_gcp&gad_source=1&gclid=Cj0KCQjwvb-zBhCmARIsAAfUI2sGSpP_uTVDMMWn92dHujDKjc0XZ_PUpKE3pgeVIQL0pvtga3EqBTAaAnGlEALw_wcB&gclsrc=aw.ds)and all the related terms one by one and how to configure it in Google Cloud Platform.

What is the Google Cloud platform?

* Google Cloud Platform (GCP) is a suite of cloud computing services provided by Google. It offers a range of infrastructure, platforms, and software services that run on the same infrastructure that Google uses for its end-user products, such as Google Search, Gmail, and YouTube.
* GCP provides a wide variety of services designed to meet the needs of businesses, developers, and IT professionals, enabling them to build, deploy, and scale applications, websites, and services efficiently.
* To perform all this task first you need to log in [Google Cloud Platform](https://cloud.google.com/free/?utm_source=google&utm_medium=cpc&utm_campaign=japac-IN-all-en-dr-BKWS-all-core-trial-EXA-dr-1605216&utm_content=text-ad-none-none-DEV_c-CRE_644159077394-ADGP_Hybrid+%7C+BKWS+-+EXA+%7C+Txt+-GCP-General-core+brand-main-KWID_43700074766895889-kwd-87853815&userloc_9154229-network_g&utm_term=KW_gcp&gad_source=1&gclid=CjwKCAjwgpCzBhBhEiwAOSQWQahhT2GNqTotewdPti-L1WQZawNxLekPcAqON-LDJu615uKDfQdyZxoCvlgQAvD_BwE&gclsrc=aw.ds) and Navigate to the Console.

****

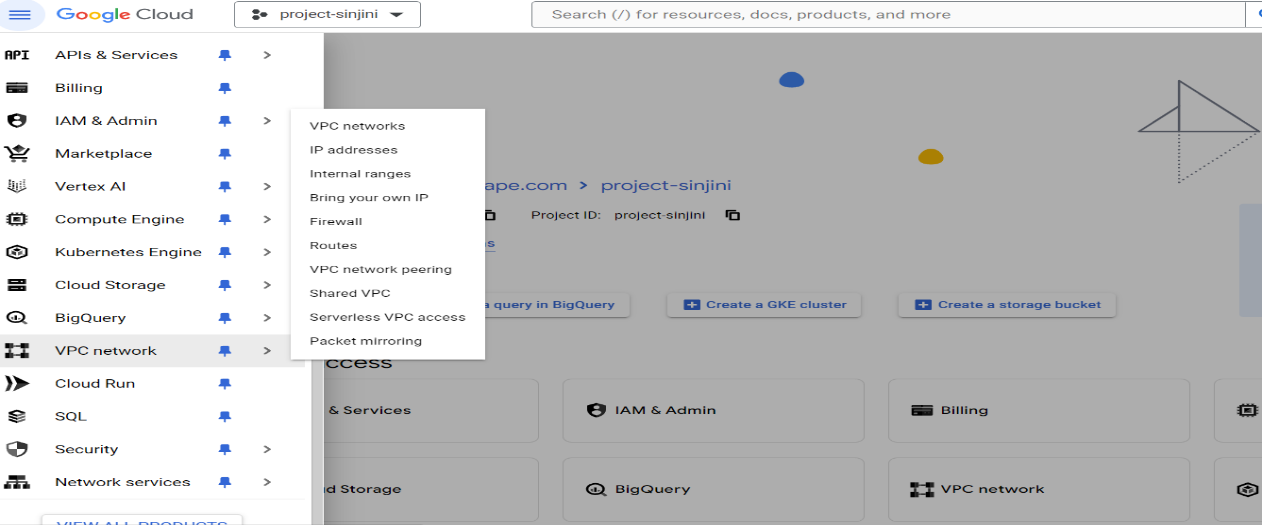
After login in-console you will arrive at the following section here you can check your project details

****

• **CREATING A VPC**

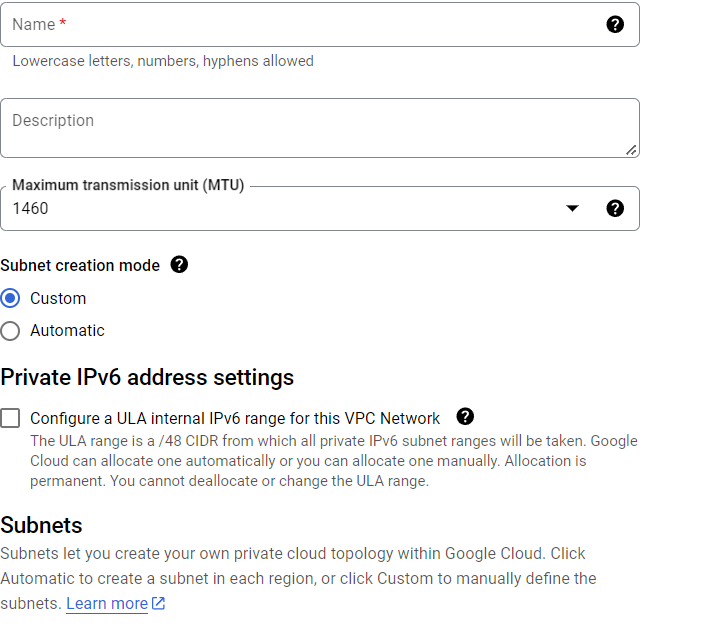
A Virtual Private Cloud (VPC) is a secure, isolated private cloud hosted within a public cloud. It allows organizations to create their private network within a public cloud infrastructure in the Google Cloud Platform.

* In the cloud console navigate to the navigation menu there you will see the VPC network.



•Click on CREATE VPC NETWORK → Enter a name for your **VPC.**

•Subnet creation mode **→** choose **CUSTOM.**

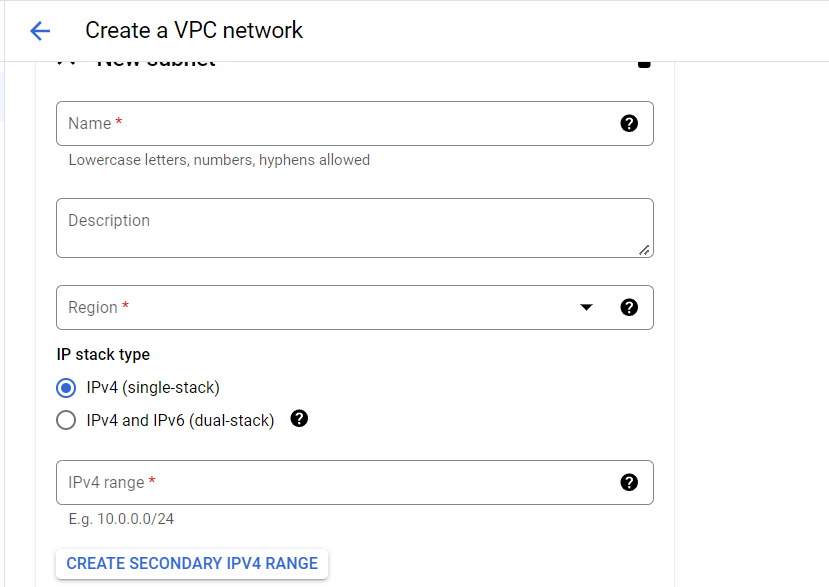


•Under New subnet **→** enter the name for your subnet.

•Enter the **IPv4 range** for your subnet.

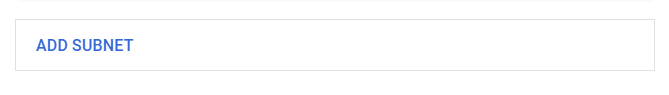
•We are going to create two Subnets one is going to be public and the second private.

•This subnet is going to be a public **subnet** thus we will keep **private Google access off**.



• Create Another subnet which will be a Private subnet.

* Click **→** **ADD SUBNET**.



• Keep private **Google access on** for your private subnet (2nd).

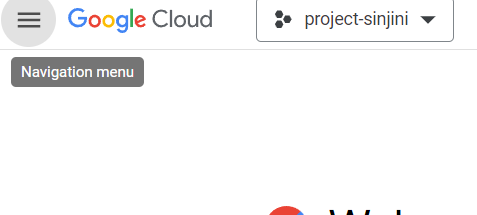
• Click Create.

* **Here we go our VPC is ready**!

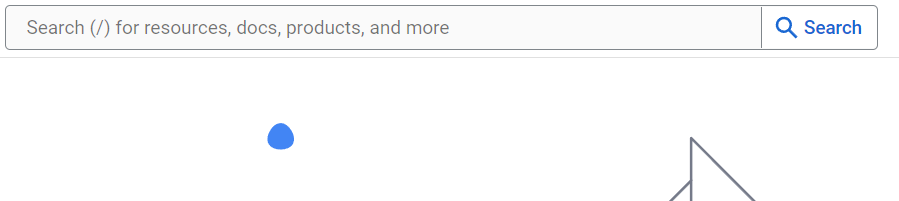
•**Creating VM instances**

To use all products in GCP you can navigate to the navigation menu or you can search in the search bar.

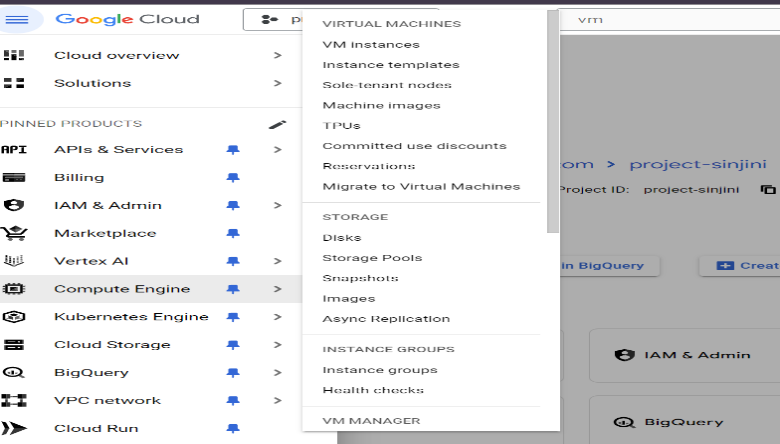
•Navigation Menu



• Search bar



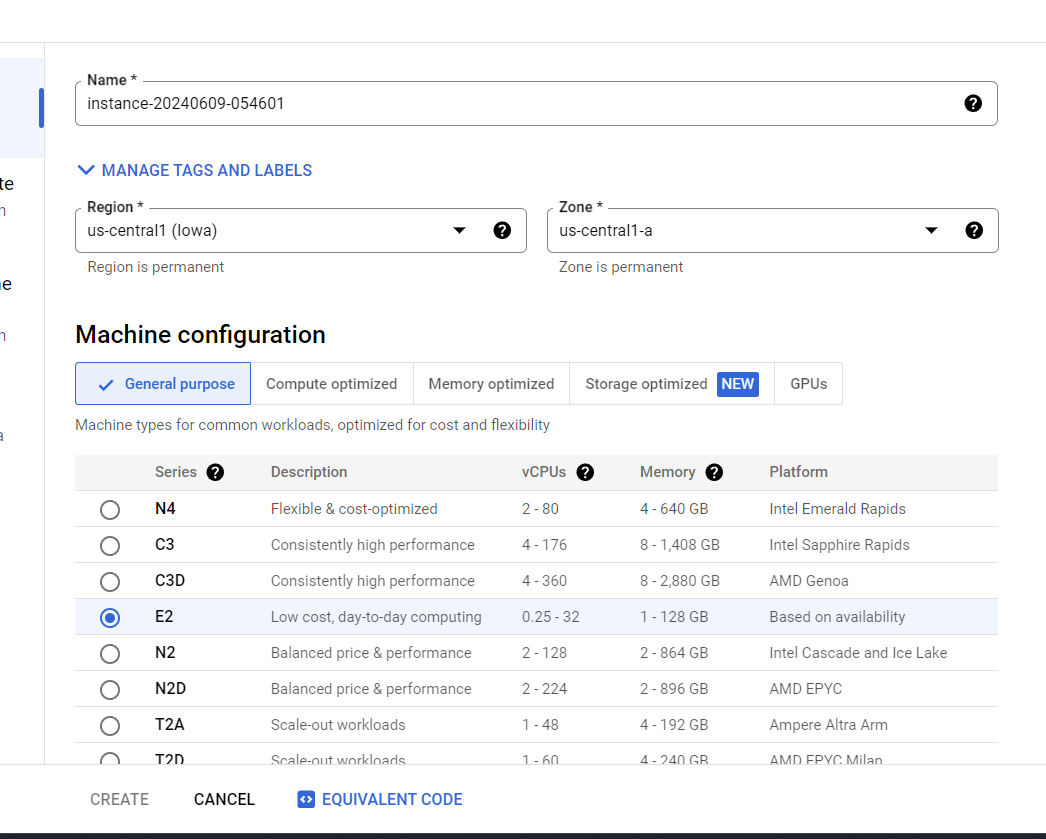
* Navigate to Compute engine **→ VM instances.**
* click on Create VM instance**.**



•**CREATING BASTION HOST**

A bastion server, also known as a jump server, is a special-purpose server designed to provide secure access to a private network from an external network, typically the Internet. It ensures that only authorized users can access the internal network resources.

* Click on **Create Instance**.
* You will navigate to the following page.
* In the name section enter the **Name** for your VM.
* Add **Region & zone**.
* In the **Machine configuration** section, add your machine type.



* Under **Machine type** section, add your preferred machine type

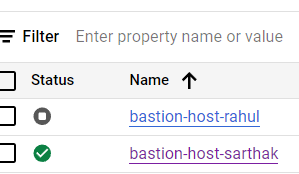
**(NOTE -CHECK COST)**

****

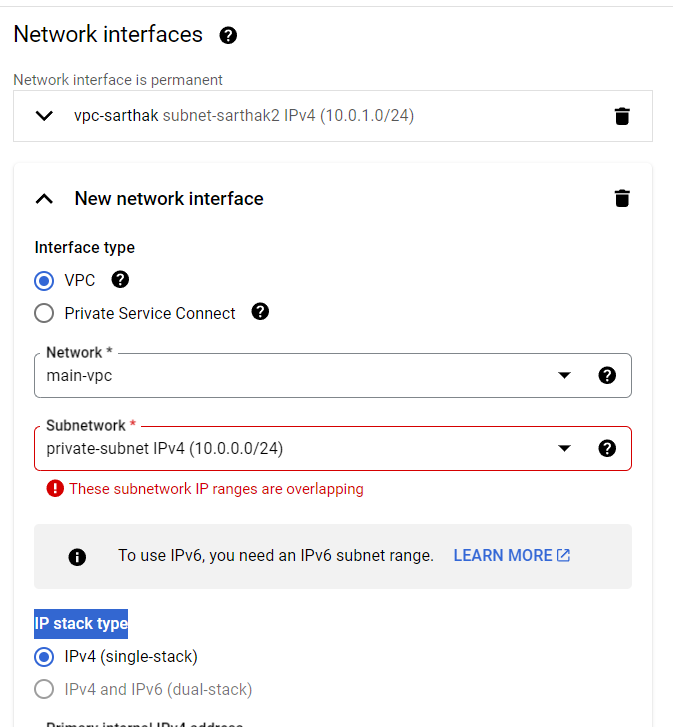
* Scroll down a bit you will see the **Boot disk** option click on that, and select the OS, storage capacity, etc. for your VM.

\

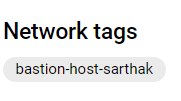
* For **Access scope** accept **Default.**
* For **Firewall don’t allow any HTTP or HTTPS traffic.**
* Click on Create **VM.**
* This green check mark indicates that your VM is **up and running**

****

* Click on Advanced option **→ Networking**
* Choose your created VPC in the Network field under **Interfaces Network.**



* In the **subnetwork field** Choose private network since your **webserver** will be inside this **private subnet**.
* This private subnet not going to have a public IP thus, click on **External IPv4 address** and choose **None**. This is very important.
* Click on External IPv4 address → click on **Ephemeral**
* In the Network tags field add tag.
* Click **save** and **create**.



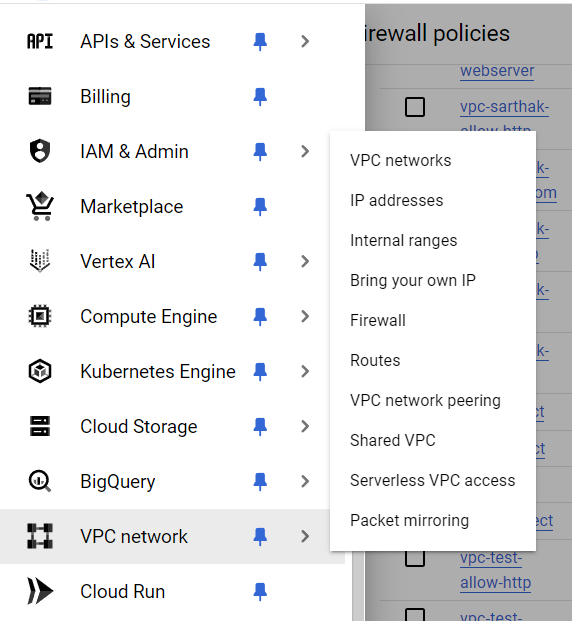
•**CREATING VM FOR WEB SERVER**

A web server hosts web applications, websites, and APIs, serving content to end-users over the internet.

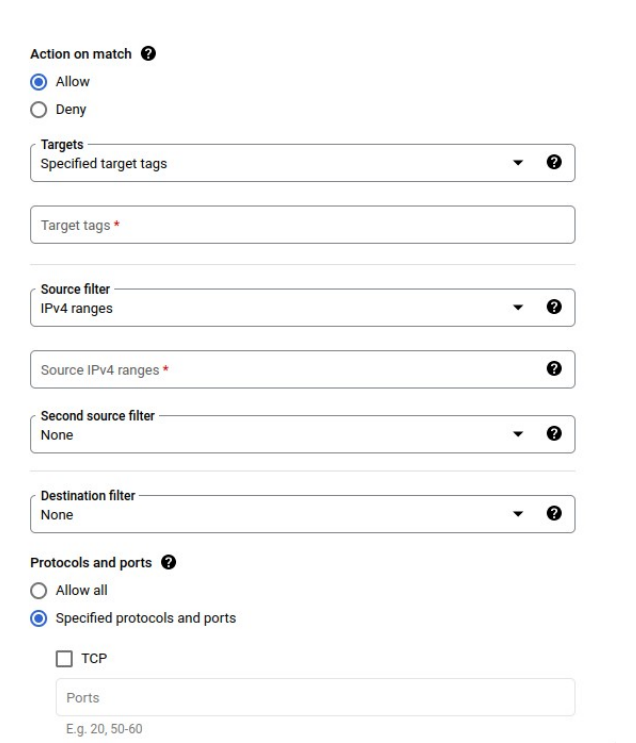
* Creating another VM for **Webserver** as we have previously proceeded to create our VM for **Bastion host**.
* Allow **HTTP** and **HTTPS** traffic.
* External IPv4 address → click on **None**
* After making these few changes click Create.
* Wait a few seconds your web server will be up and running.

•**Adding firewall rules**

* Navigate to VPC network → **Firewall**

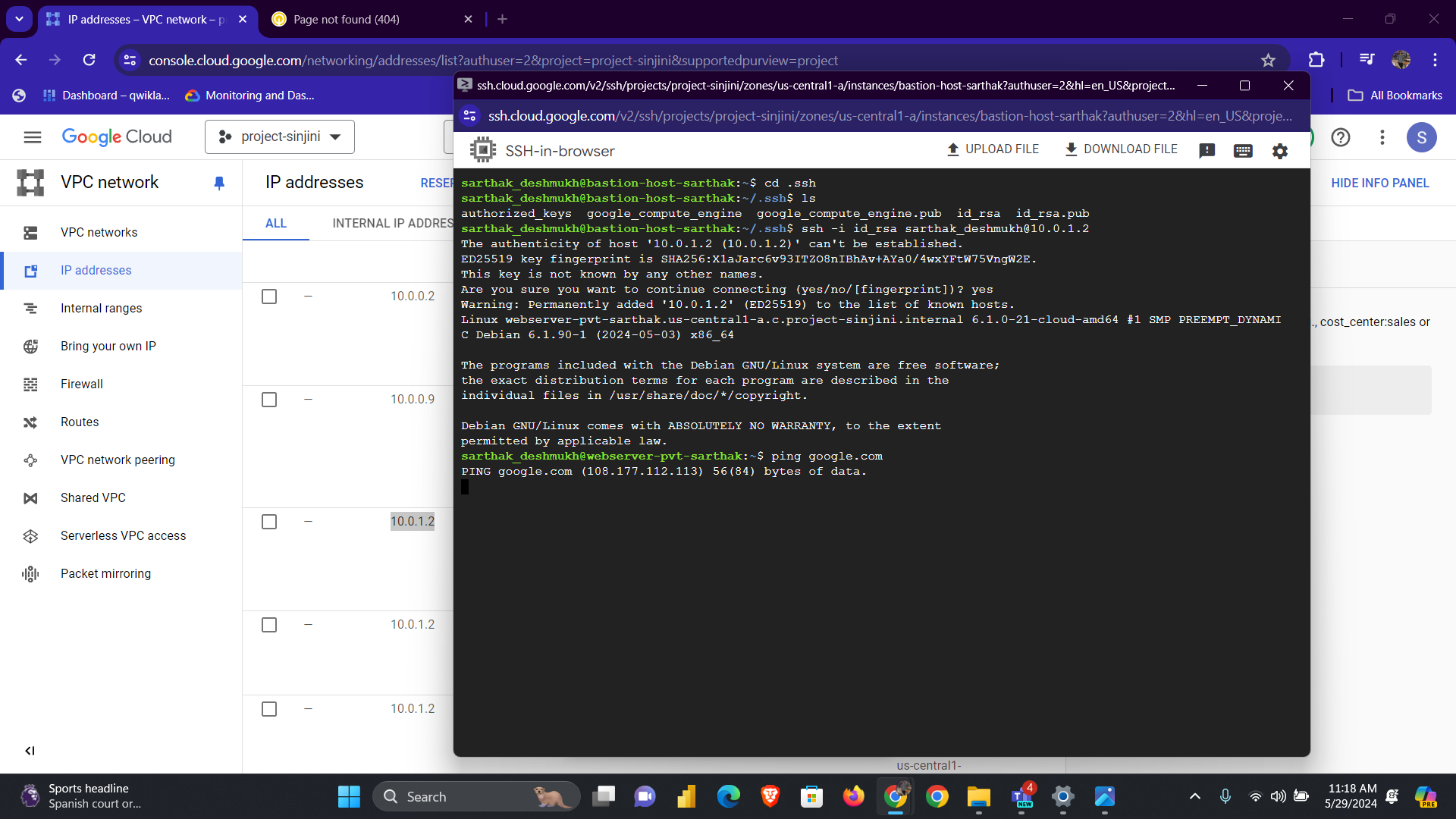
****

* Click on **Create Firewall Rules**.
* Name firewall rule under Name field →internet to bastion.
* Choose the name of your created VPC under the Network\* field (VPC-sarthak).
* In Targets→ **Specified target tags**.
* For source filter →IPv4 Ranges.
* **0.0.0.0/0** Enter this rangefor source IPv4 ranges, this range will attract traffic from all around the world to your host.
* For **protocols and ports** choose **specified protocols and ports** and for **TCP select 22.**
* **Add a Starting script “sudo apt update && sudo apt -y install apache2**”
* Click on **Create**.



* Add one more firewall rule **→ CREATE FREWALL RULES.**
* Name the firewall rule → **bastion-to-webserver.**
* Enter the name of your targeted tags → webserver this will transfer the targets from Bastion to the webserver.
* Choose Source tags.
* Enter the tag name from where you want the traffic **from Bastion**
* Protocols and ports → Protocols and ports Specified **TCP 22**
* Click on **CREATE.**

**Accessing to bastion SSH and through bastion server connecting WEBSERVER, here bastion is acting as a bridge for administrator access.**



•**Configure NAT/ROUTES**

•**CLOUD NAT**

Cloud NAT is a managed service provided by Google Cloud Platform that allows instances in a private VPC network (those without external IP addresses) to access the internet for outbound connections, such as downloading updates or connecting to APIs, without exposing these instances directly to the internet.

•**NAT GATEWAY**

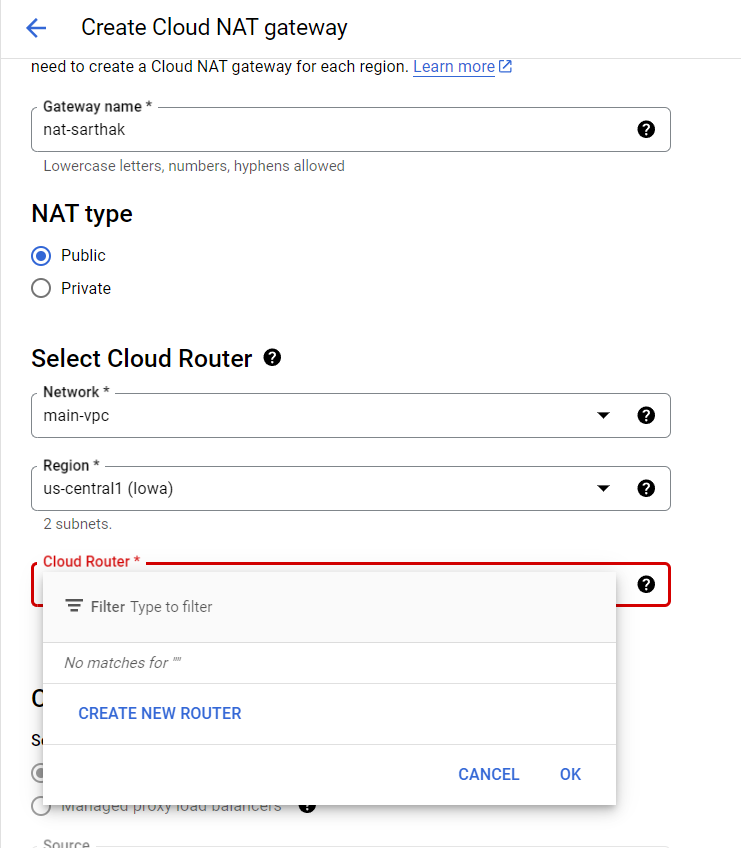
A NAT gateway is a specialized, managed network resource that performs Network Address Translation for instances in a private subnet to allow them to access the internet or other external services while preventing inbound traffic from directly accessing those instances. This is particularly important in cloud environments such as AWS, Azure, and Google Cloud Platform (GCP).

•**ROUTES**

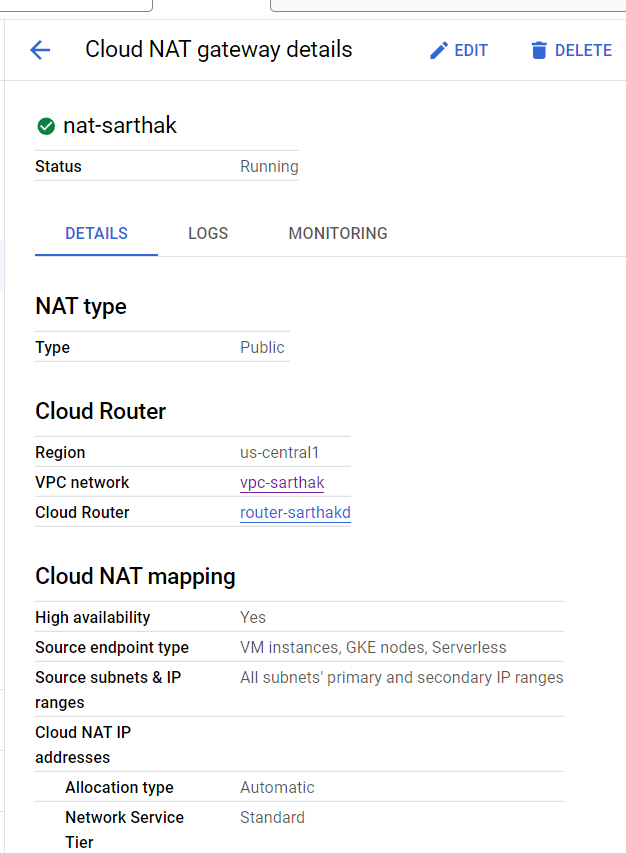
Routes are rules within a VPC network that determine how packets are directed to their destinations. Routes control the flow of traffic within your VPC network and between your VPC network and other networks, including the Internet.

* Navigate to → **Cloud NAT**
* Click → create cloud NAT gateway



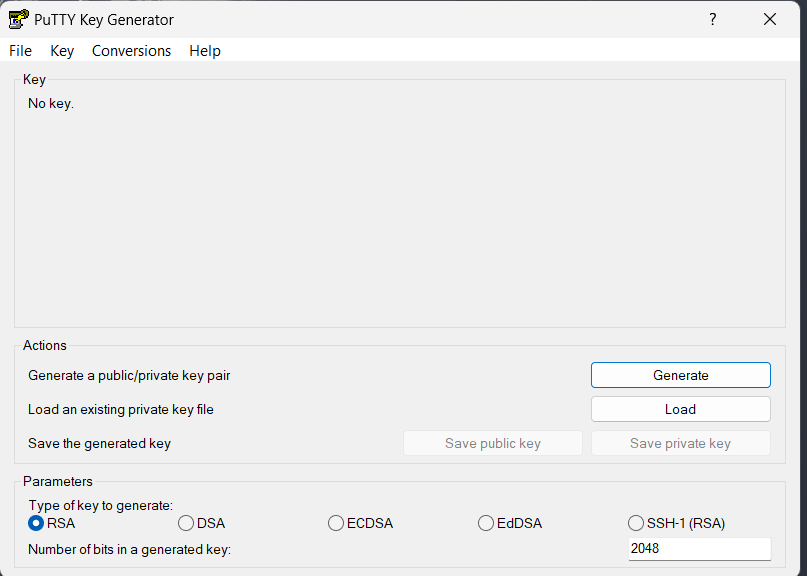
* In the **Gateway name** section give your gateway your desired name
* NAT type → **Public**.
* For cloud Router→Network→ **Select your VPC**.
* For region → Default (present in section).
* In Cloud Router →Create new router.
* Add **name** in cloud router section.
* Back to the Cloud Router section and select the **router name** you just created.

•**Green check mark indicates NAT& ROUTES are ready, thus the section should be like this.**



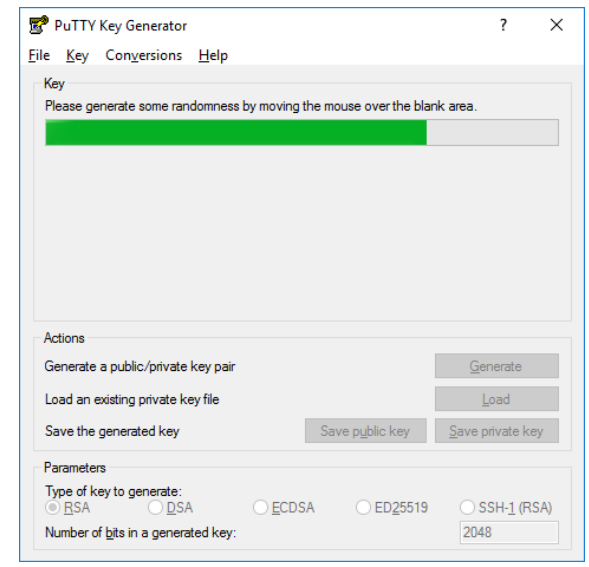
• Connecting **bastion** with **web server** in SSH client **– PUTTY**

* Go to VM instances and start the VM instances.
* You will only see an SSH connection for the **Bastion server** and not for the webserver and also there will be no **external IP**.
* Now open **Putty Gen** on your local computer.

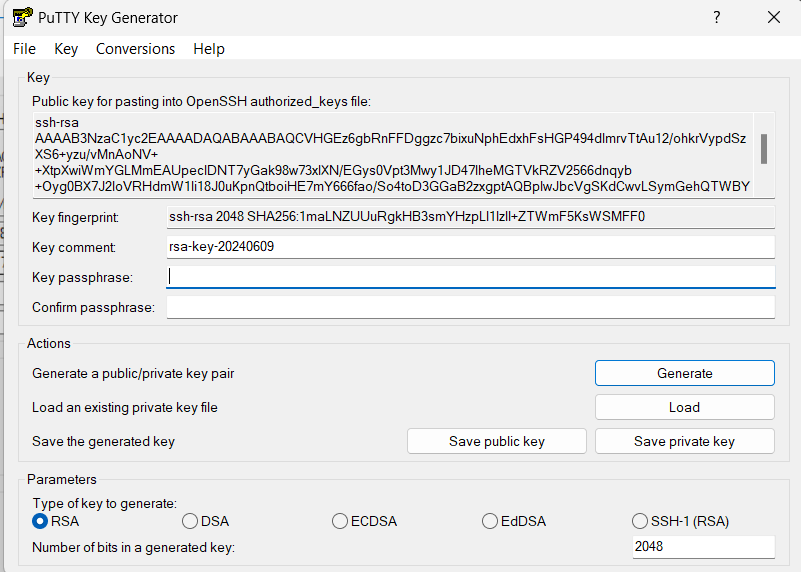


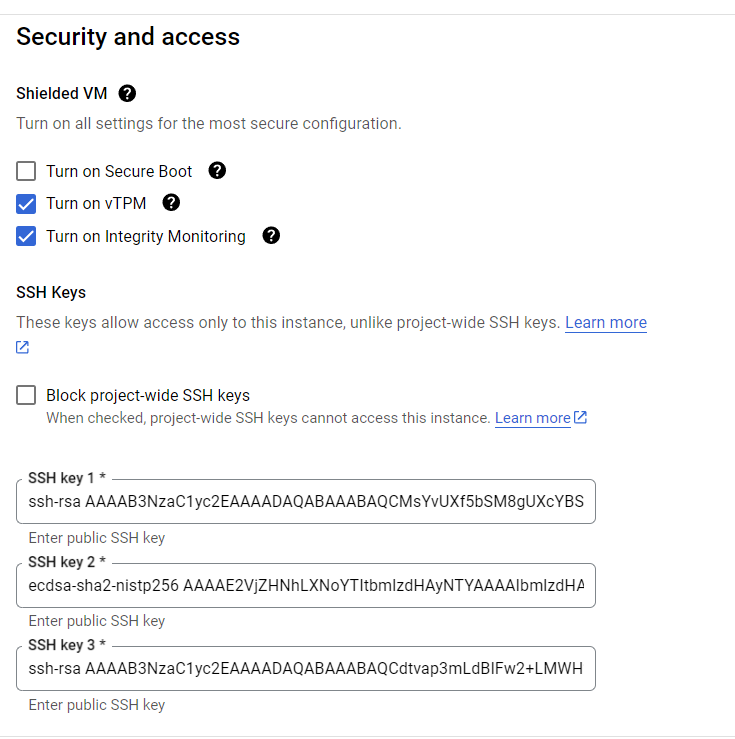
**Parameters section**

* Type of key →**RSA**
* Number of bits in key →**2048**
* Click **Generate.**

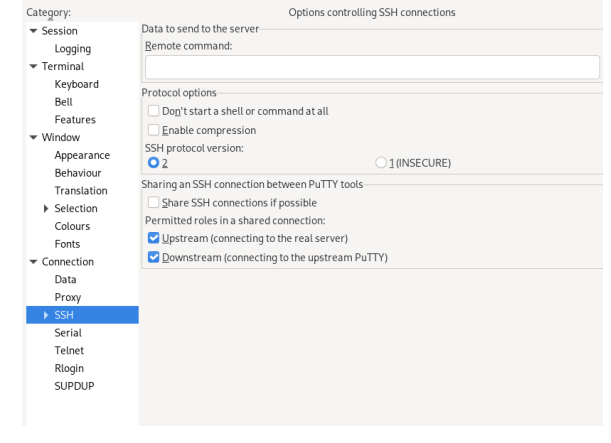


* Copy this generated key, this key is the **public key.**
* Now go to Bastion VM instance and click →Edit

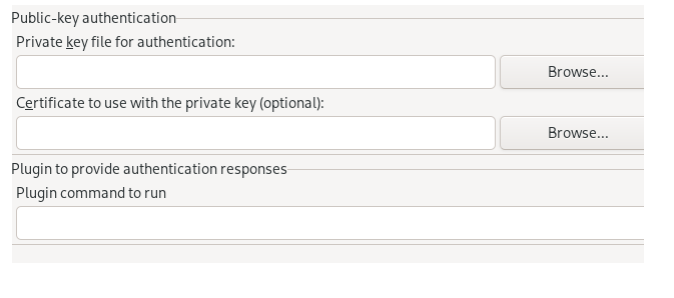




* Paste the previously copied public key here.
* Click **save.**
* Open the putty gen window and click on Save private key → Save the private key and remember the destination of the file.
* Save the file as (.PPK).
* Now open **Putty** navigate →**SSH**



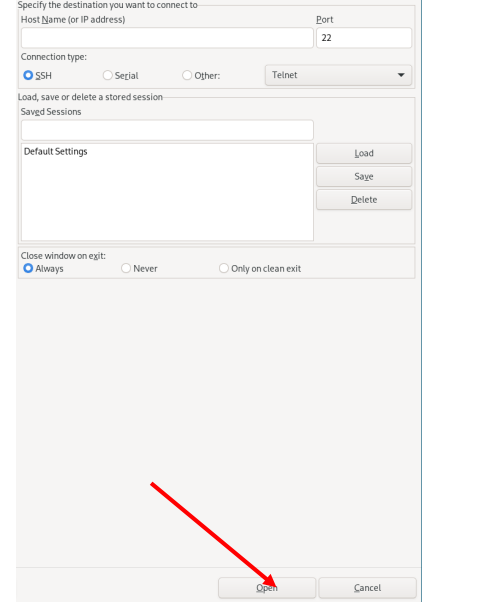
* Click on dropdown button of SSH → **Dropdown button of Auth** →**Credentials**



• Click on Browse → browse the private key (.PPK) you have saved

• Click on the session → under Hostname field → Enter the **external IP of** **Bastion VM → Port 22 → click Open**.

* Make sure the connection type is **SSH.**



**FOR LINUX**

* Open **Terminal.**
* **Generate private key**→
* Putty gen -t rsa -b 2048-c[username]-o [name \_the \_key]. ppk
* Enter a passphrase if needed and press enter or just press enter to skip.
* Generate public key from the created private key →
* Putty gen -L [name\_ of\_ the \_key].ppk.
* Copy the public key in the bastion server like we did previously (in the WINDOWS case) Save.
* Open PUTTY → Navigate to SSH.
* Click on dropdown button of SSH → Dropdown button of Auth → Credentials.
* Click on Browse → browse the private key (.ppk) you have saved.
* Click on the session → under Hostname field → Enter the external ip of Bastion VM→
* Port 22 →click Open.

**LOG IN:**

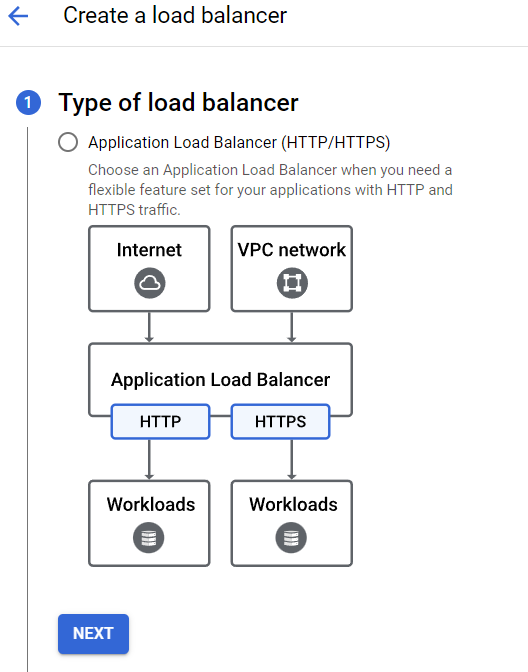
• Enter the username → press enter and you should be inside try-bastion server.

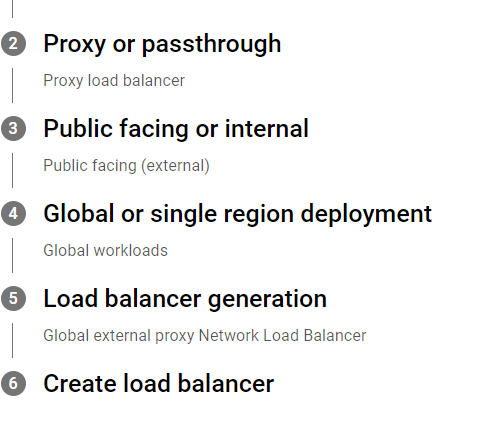
• Once you are inside bastion VM, generate a private key and public key for the webserver from bastion VM and try to connect it.

* **Configure Load balancer**

A load balancer acts as an intermediary between clients and servers, distributing incoming network or application traffic across multiple servers. By balancing the load, it ensures no single server becomes overwhelmed, which enhances the performance, reliability, and scalability of applications and websites.

* Navigate to load balancer → click on create load balancer
* Select the type of load balancer you want (in GCP there are different type of load balancers make to study all of them before selecting).
* I have selected HTTP &HTTPS.
* Click Next





**Make sure tostudy all of above terms throughly before creating load balancer.**

* Click on **Configure (refer pages 23 to 27)**

First, we need to create an **Instance Group & Instance Template** for the backend service.

**Instance Group**

Instance groups are collections of VM instances that you can manage as a single entity. There are two types of instance groups: unmanaged and managed.

**Instance Template**

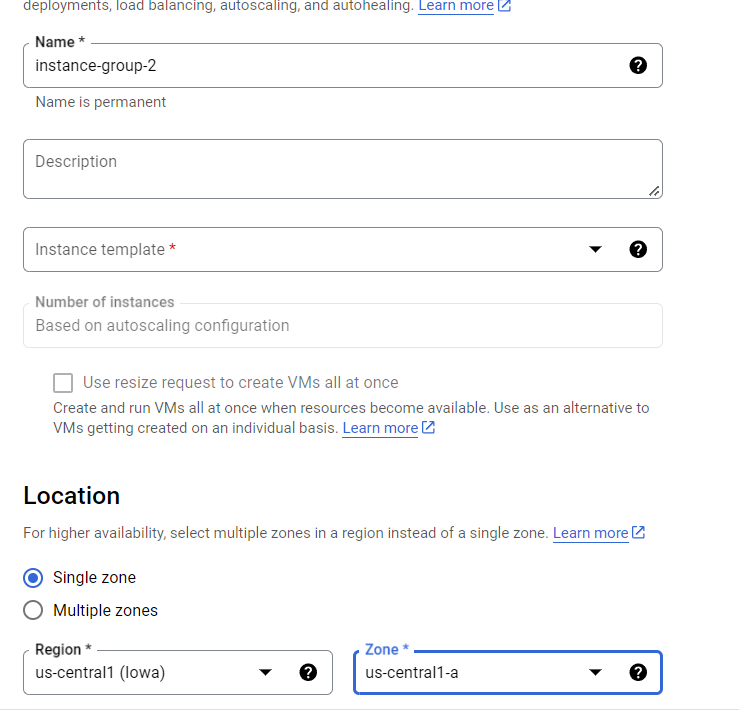
Instance templates are resource configurations that define the settings for VM instances, such as machine type, disk images, startup scripts, and network settings. They act as blueprints for creating new VM instances.

Instance templates, instance groups, and load balancers in Google Cloud Platform (GCP) are interconnected components used to manage and scale applications efficiently:

1. **Instance Templates**: Blueprints that define the configuration for VM instances, including machine type, disk image, network settings, and metadata.
2. **Instance Groups**: Collections of VM instances are managed as a single entity.
   * **Managed Instance Groups (MIGs)** use instance templates to automatically create and manage identical VM instances, providing features like autoscaling, load balancing, rolling updates, and self-healing.
   * **Unmanaged Instance Groups** require manual management of instances.
3. **Load Balancers**: Distribute incoming network traffic across multiple VM instances in an instance group to ensure high availability and reliability of applications. They work with instance groups to evenly distribute the load and handle traffic spikes.

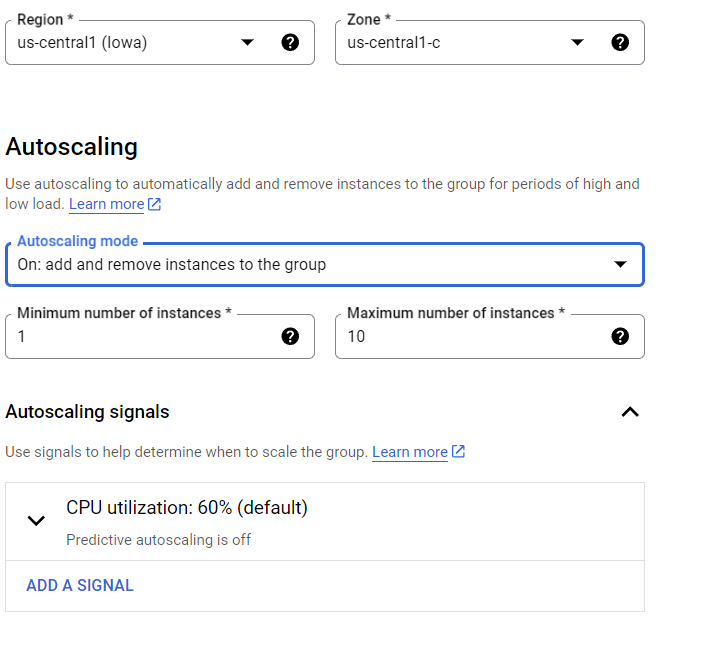
**Correlation**:

* **Instance Templates** define the settings for VM instances.
* **Managed Instance Groups (MIGs)** use these templates to create and manage VM instances automatically.
* **Load Balancers** distribute incoming traffic across the instances in the instance group, ensuring efficient load distribution and high availability.
* Navigate to Compute Engine → **Instance Templates**.
* Name→ Your choice.
* Accept the default setting**, click Create** **Templates**.
* Now Navigate to→ **Instance Groups.**
* Add previously created **Instance Template** in this section.
* Select the preferred **Region** & **Zone.**



In this section, I (we) will also configure **AUTOSCALING and AUTOHEALING**

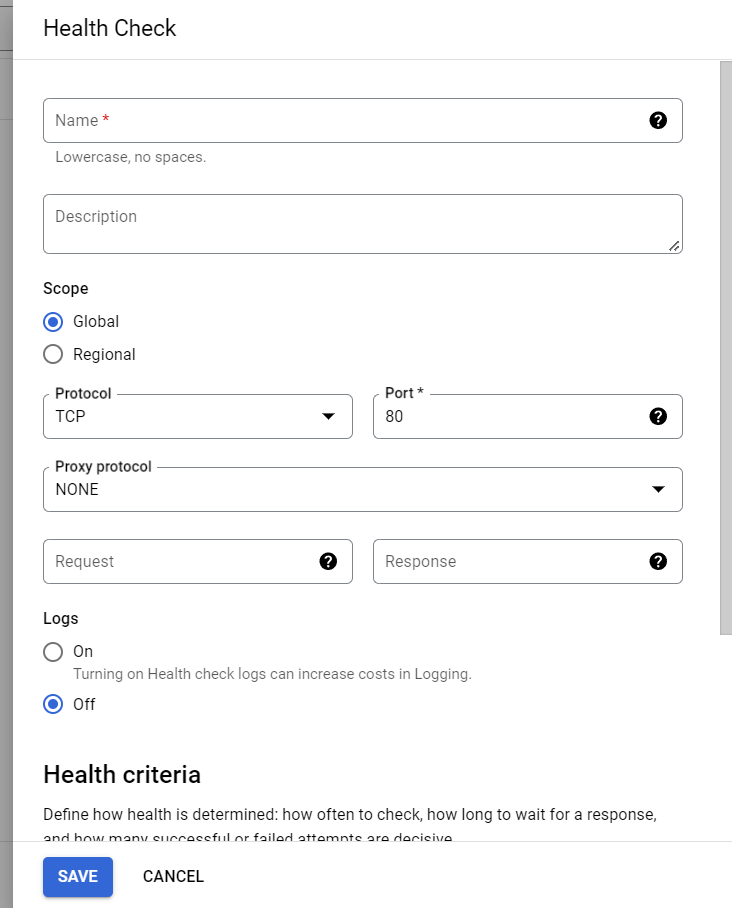
* **Configure Autoscaling and auto-healing**
* For **Autoscaling mode** select → **on: add and remove instances to the group.**
* For minimum number. of instances → 1
* Maximum num. of instances→ 3(your choice or need).
* Autoscaling signals →signal type → CPU utilization.
* For target CPU utilization →**60%,70% or 75**
* **Leave the rest to default.**

****

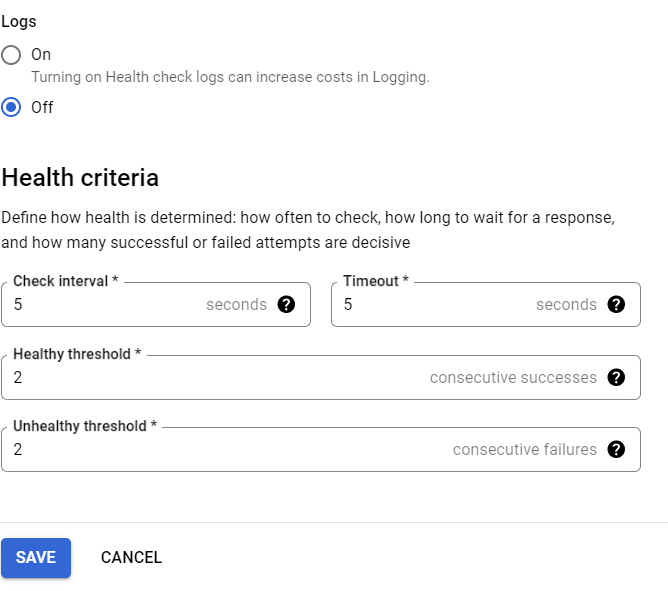
* Scroll down and look for the **auto-healing** section.
* Click on **Create a health check.**



* For health check name→ its your choice.
* scope→ Global.
* Protocol → **TCP­→ port → 80.**



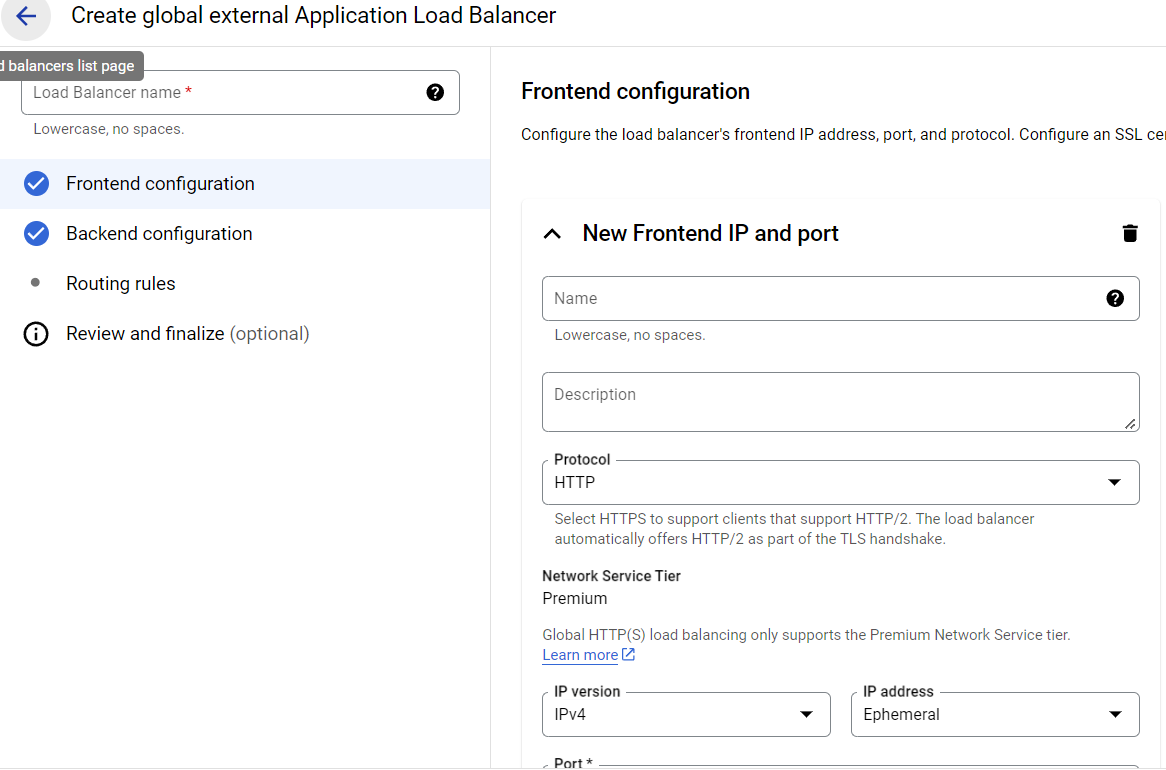
* Health criteria→ Accept default.
* Click save.



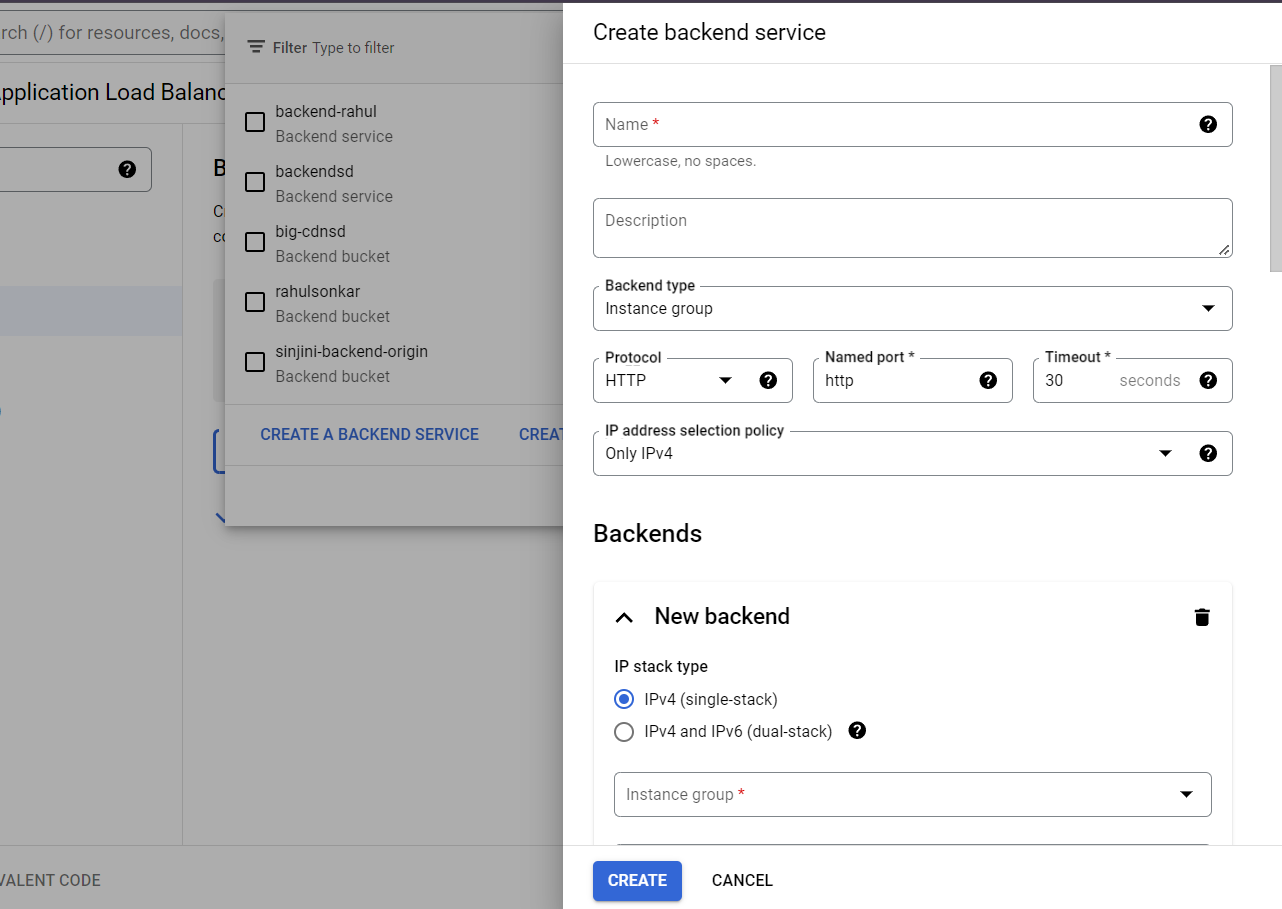
* Now add that health check we created in this section, your created health check will be here.
* Click **Create**



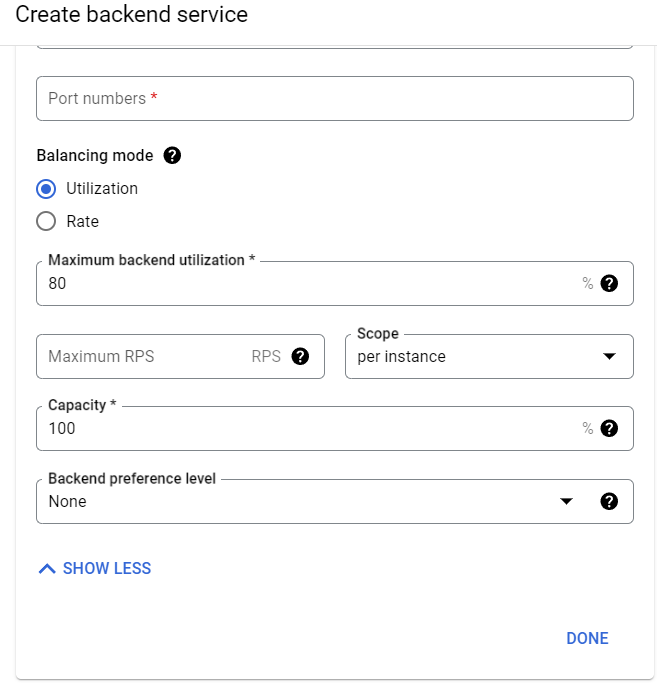
* Now Go back to **Load Balancer** (refer page 17).
* Give you Load Balancer → name.
* Click on →**Frontend configuration**→ name protocol →HTTP
* IP version → IPv4 IP address → for now **Ephemeral**(note)
* Port →80
* **DONE.**



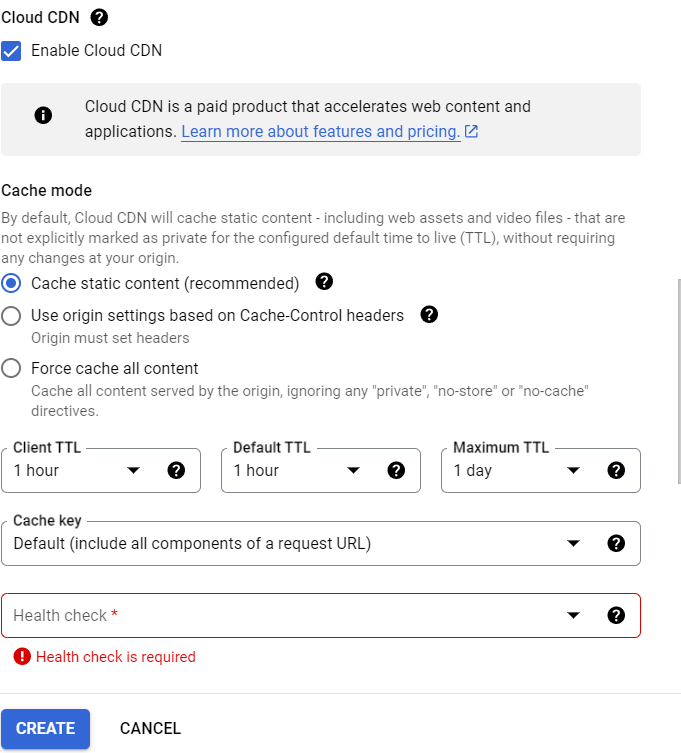
* For **Backend configuration.**
* Click **on Create a service Backend.**
* You will arrive at following page.
* Give you backend service specific name.
* Leave the reset **to default.**
* Click on → instance group → select your **instance group** which we create previously.



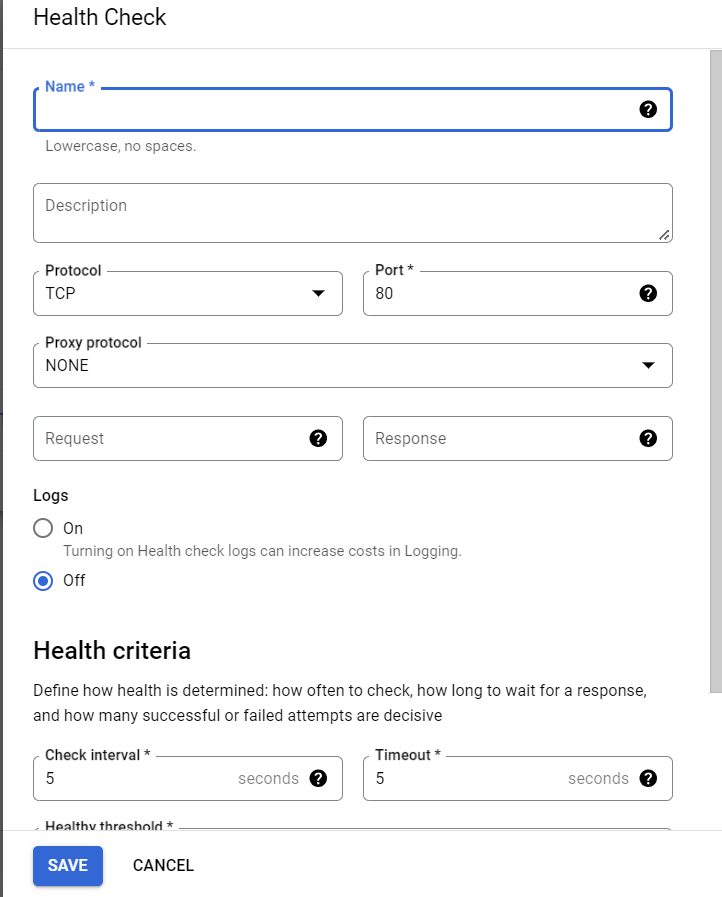
* For port Numbers → 80 (since it is global it will serve for your Apache).
* For maximum backend utilization → 80



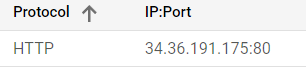
* Leave rest to default click **done.**
* Enable Cloud **CDN (**will give brief introduction letter**)**
* **Leave** rest as it is
* For Health check → click on create health check.

****

* For health check name → type name,
* For now accept default settings.
* **Click save**
* Now select your created health check.
* Click on create.

**\**

**Your load balancer is ready and running click on your load balancer name you will find an IP there copy and paste in browser if its working you will find Apache page there.(VM must be on)**

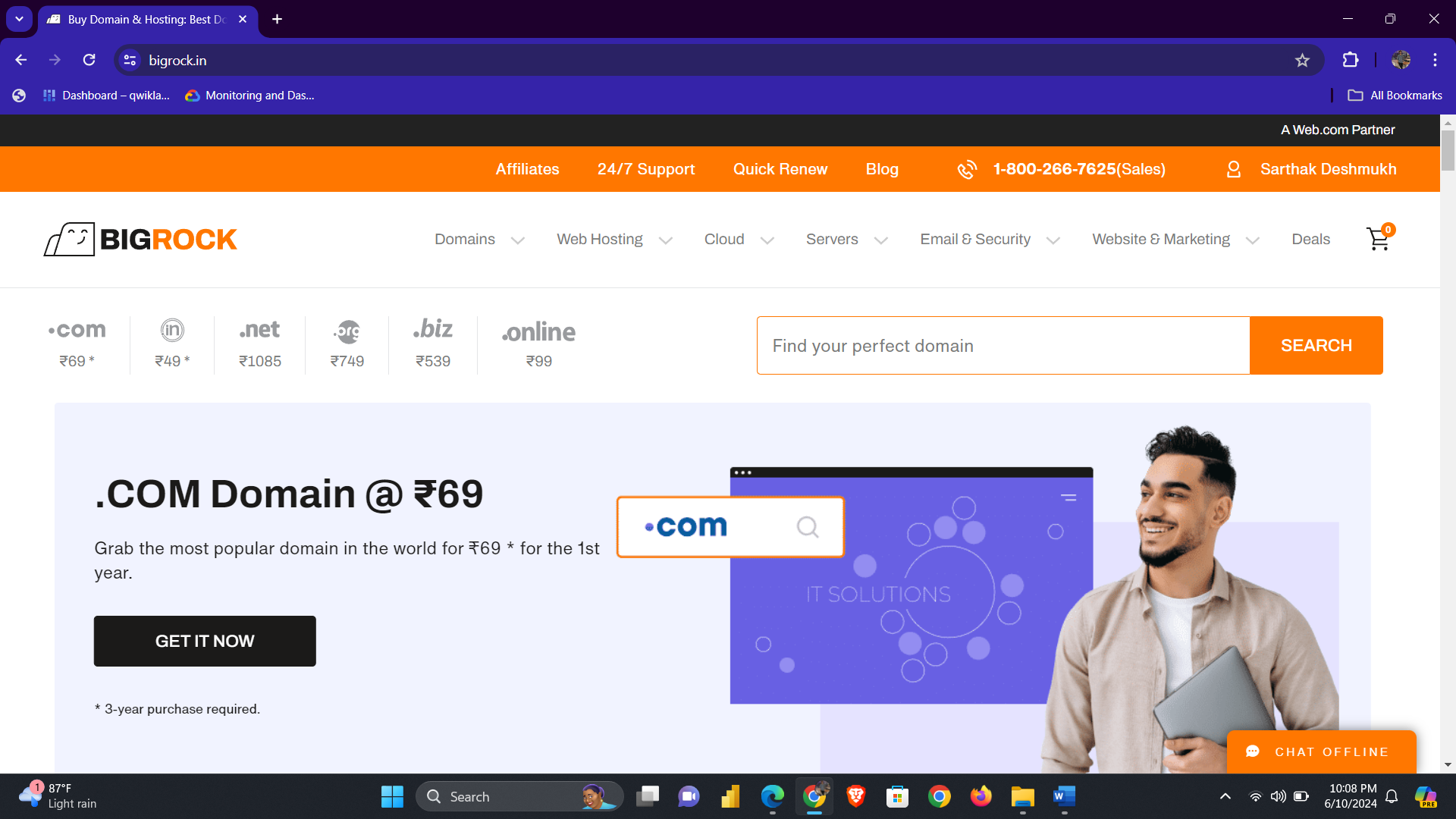
****

* **Configure DNS and Cloud flare**

Cloud DNS is a scalable, reliable, and managed domain name system (DNS) service provided by cloud platforms such as Google Cloud, AWS, or Azure. **It translates domain names into IP addresses, allowing users to access websites using human-readable names instead of numeric IP addresses**. Cloud DNS offers high availability, low latency, and automatic scaling, ensuring that DNS queries are resolved quickly and efficiently, even during high traffic periods. It is often used to improve the performance and reliability of web services and applications.

**Buying Domain from BIGROCK**

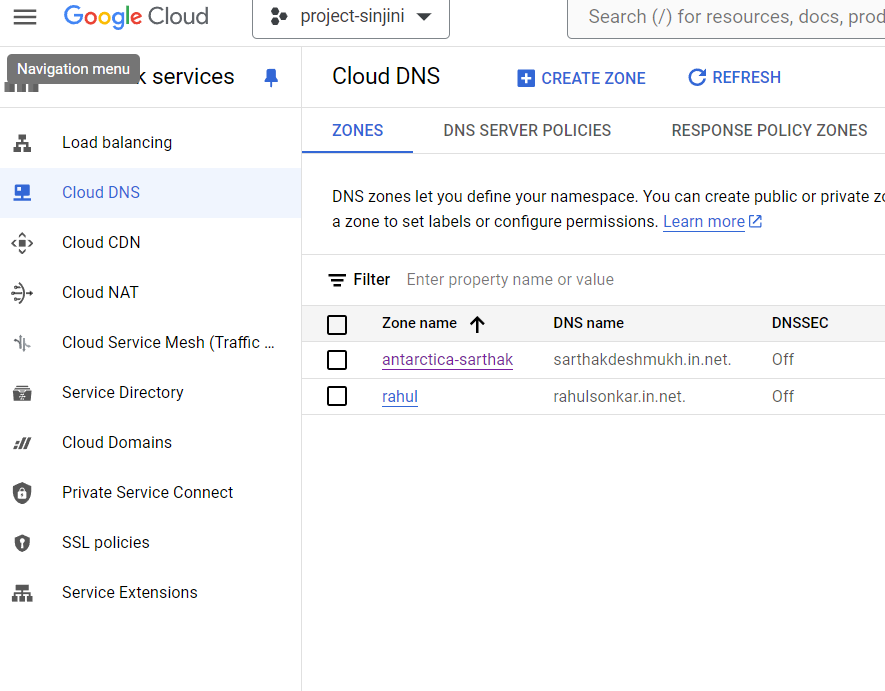
* Purchase your desired domain from [**BIGROCK**](https://www.bigrock.in/tld/com-domain?gad_source=1&gclid=CjwKCAjwyJqzBhBaEiwAWDRJVK5TswnU58uAUf-Q8rIviHSuHpm3MN_uWKMyYB0LWFyElQinLisLOhoC2Y4QAvD_BwE)**.**
* Create account → in search bar type your desired name.
* Select make a payment. (81rs)



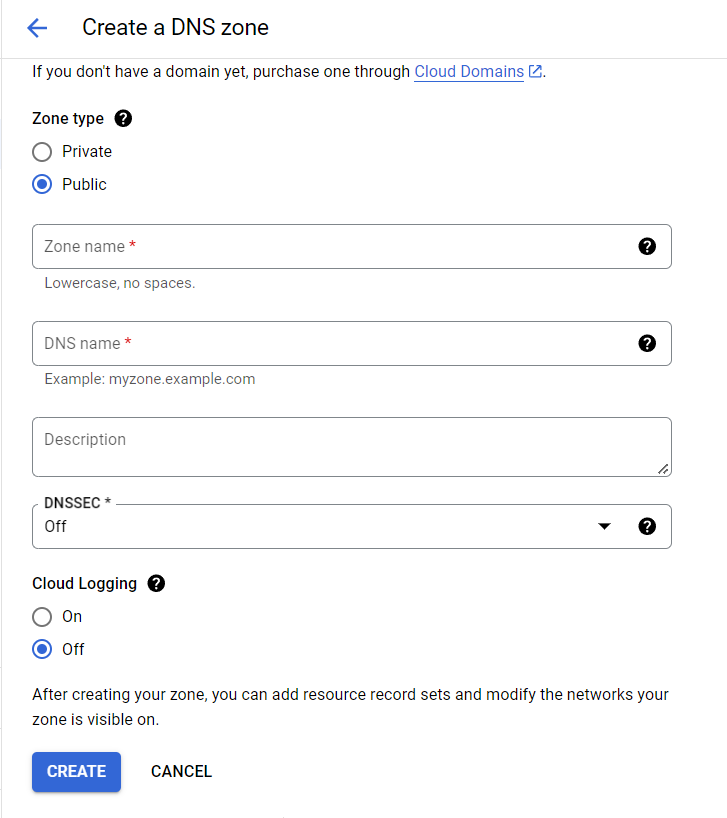
* In home section you can you can find your purchased domain.
* Domain [**Sarthak**](http://sarthakdeshmukh.in.net/)**.**

****

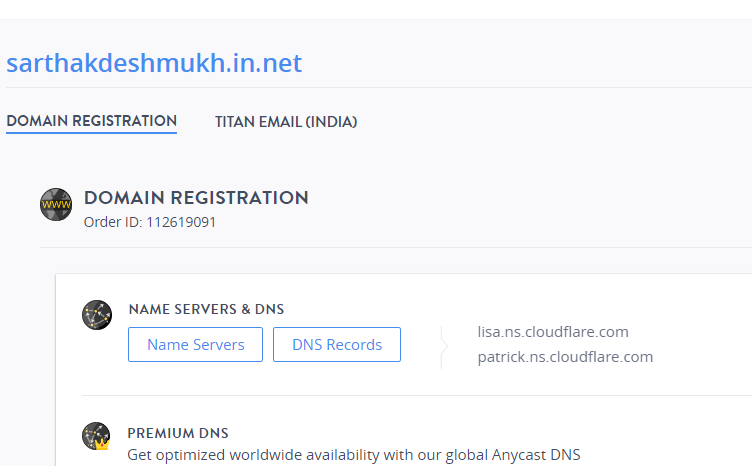
* **Configuring Cloud DNS**
* Navigation menu → **Network services** → **Cloud DNS**.
* Click create → **zone**.



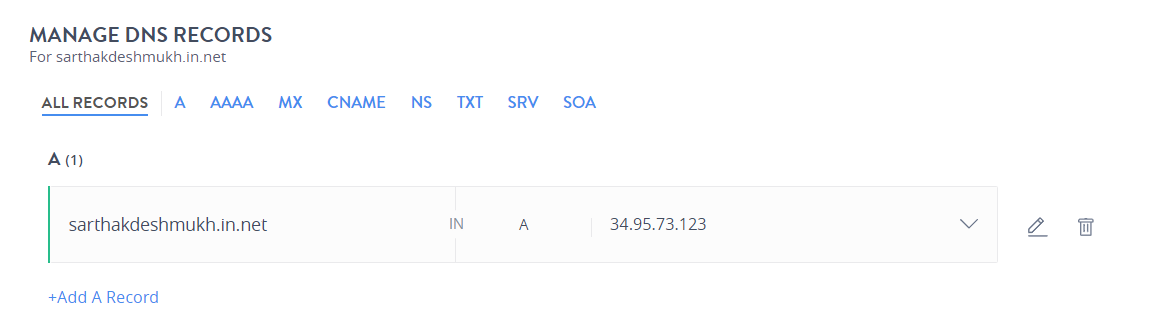
* For the zone type **→ public**
* For zone name → your choice
* For the DNS name, enter a DNS name suffix for the zone by using a domain name that you registered (for example, example.com) (sarthakdeshmukh.in.net)
* For **DNSSEC** section → OFF.
* Click create to create zone populated with NS and SOA records.
* To point your registered domain name to the IP address of the hosting server, you must add an **A record to your zone**.

****

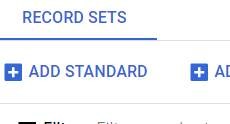
* **How to add A record.**
* Go back to [**BIGROCK**](https://myorders.bigrock.in/orders/manage/sarthakdeshmukh.in.net/domain)→ your account →DNS records.



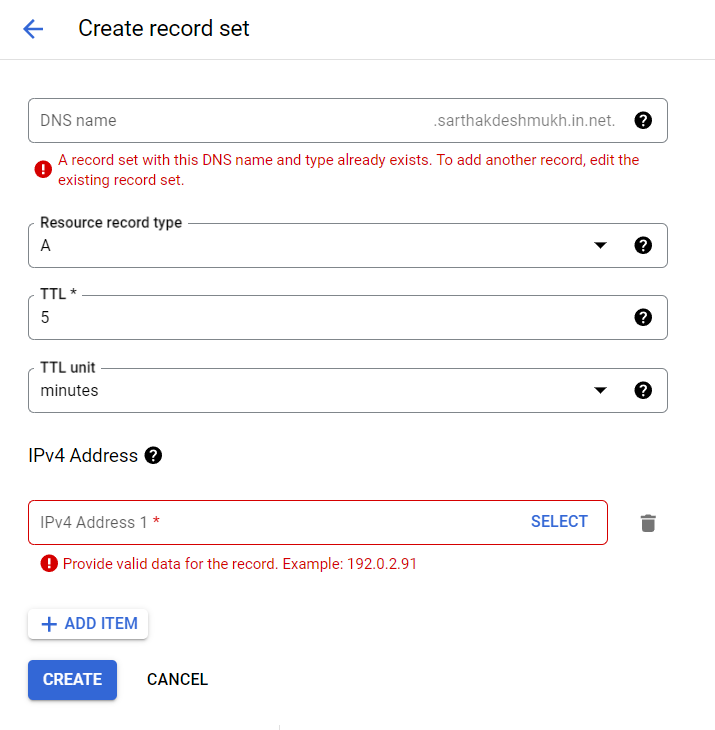
* Click → Add A Record (like this)

****

* Next click **→** Add standard.

****

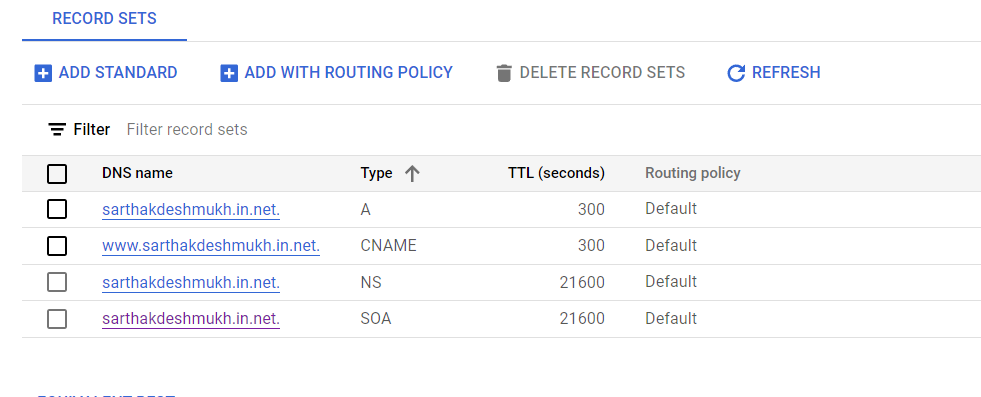
* For DNS name **→** your prefix DNS name.
* Resource record type **→ A.**
* for IPv4 Address **→ External IP address** of your Load Balancer.
* Click to create A record for your zone.

****

* For Resource type**→ CNAME.**
* For Canonical Name **→** Enter your Domain name.
* Click create.

****

* **This is how your record set will look like.**

****

* **Configure Cloud flare**

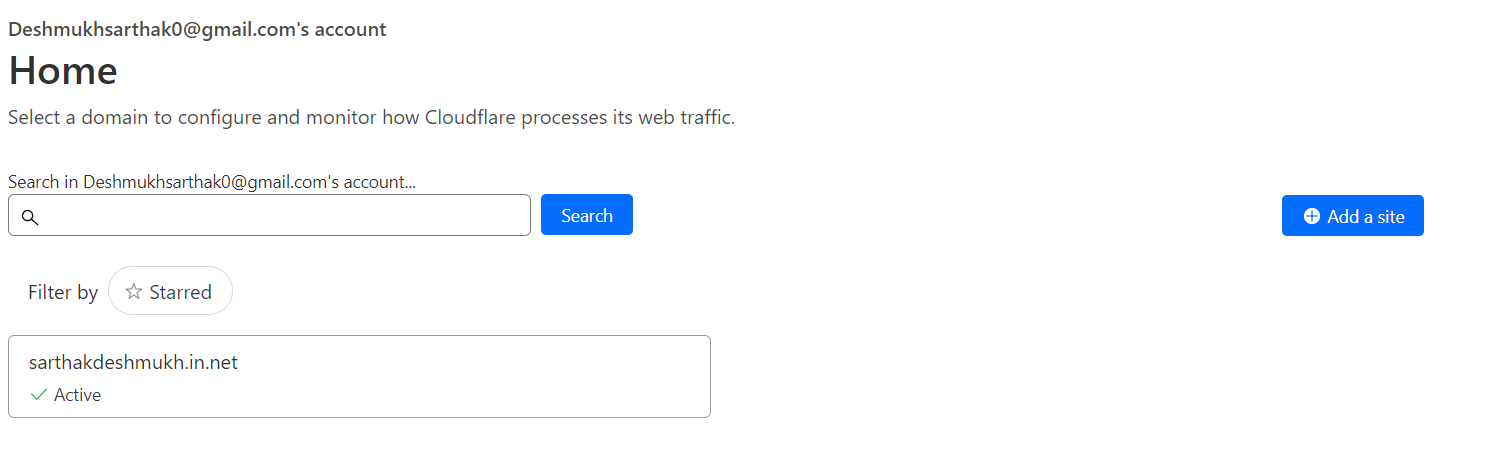
**Cloudflare** provides a suite of services including CDN &DNS management, security features, and performance enhancements for your website.

* Go to [**Cloud flare**](https://www.cloudflare.com/) website sign up or Log in.

A close up of a screen

Description automatically generated

* After logging in Click on **→ Add site**.
* Enter your domain here **→ Search→ continue**.



* Choose A free plan for your Domain.
* Click **→** continue.

A screenshot of a computer

Description automatically generated **Cloudflare will automatically scan your current DNS records and display them for review.**

**Verify that all your DNS records are correct. You can add, edit, or delete records as needed.**

* Click **→** Add records.
* Select type **→ A**
* For **Name→** enter @.
* For **IPv4 Address→ The External IP address** of your load balancer.
* Click on **Save.**

A screenshot of a computer

Description automatically generated

* Again, click on **→** **Add Record** and now choose **CNAME** in the **Type** field.
* For **Name** enter**→ www** and for **Target** field enter your domain**→** (example → sarthakdeshmukh.in.net)
* Click on **Save**.

A screenshot of a computer

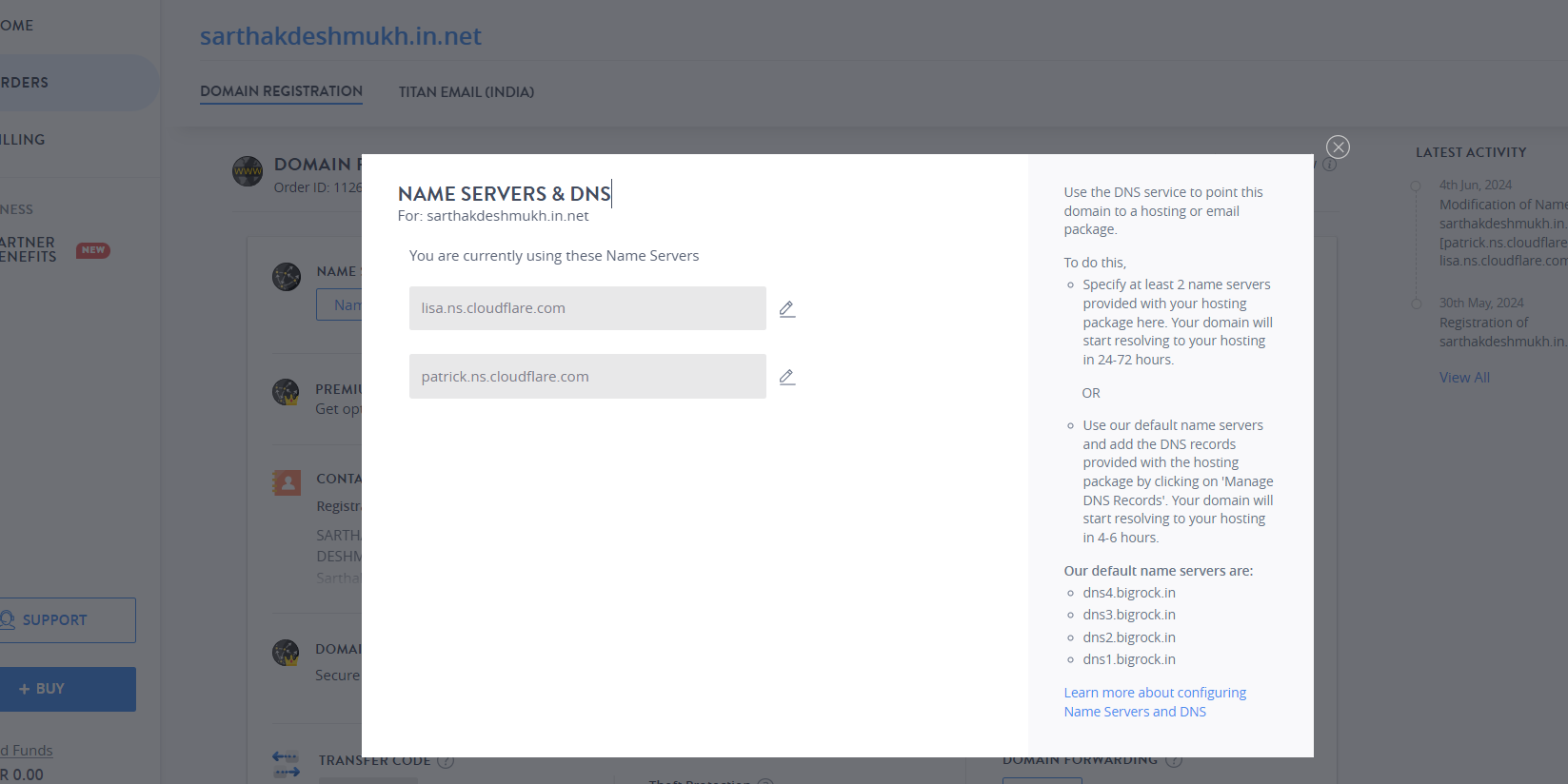
Description automatically generated

**Update your name servers.**

* Cloudflare will provide you with a set of nameservers.
* Copy both the nameservers.



* Navigate to Big Rock **→ Name servers.**
* Replace these default name servers with the name servers provided by Cloud Flare.
* Click save changes.

****

* Now go back to Cloudflare and click on **Check nameservers**.
* Nameserver changes can take some time to propagate, typically up to 24-48 hours.
* Cloudflare will automatically detect when your nameservers have been updated. You will receive an email notification once the process is complete.

A close-up of a computer screen

Description automatically generated

* Now click on Improve Security uncheck the automatic HTTPS rewrites and click on Save.

A screenshot of a computer

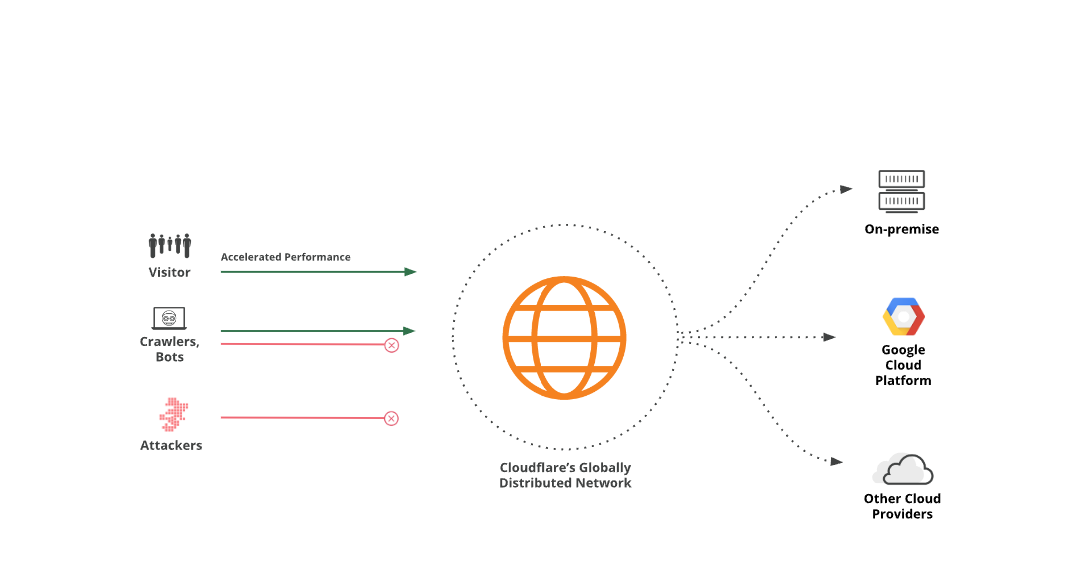
Description automatically generated

* And then uncheck the Always use HTTPS and click on save.
* Click Save
* Click Finish.

A screenshot of a computer

Description automatically generated

This is how Cloud Flare works.

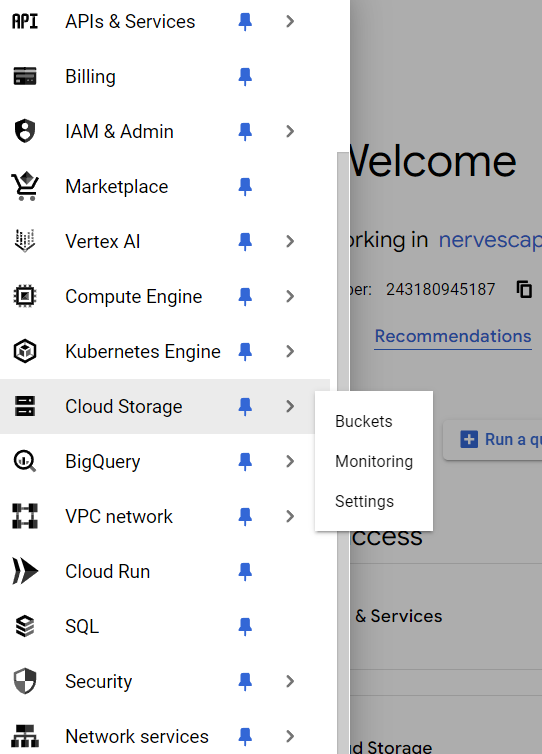


* **Configure CDN**

A Content Delivery Network (CDN) is a system of distributed servers that deliver web content and other digital assets to users based on their geographic location. **CDNs improve website** **performance by caching content closer to users, reducing latency, and decreasing server load**. This results in faster load times, improved reliability, and a better user experience, especially for high-traffic websites and applications. CDNs also enhance security by protecting against DDoS attacks and other malicious activities.

**Set Up a cloud storage bucket.**

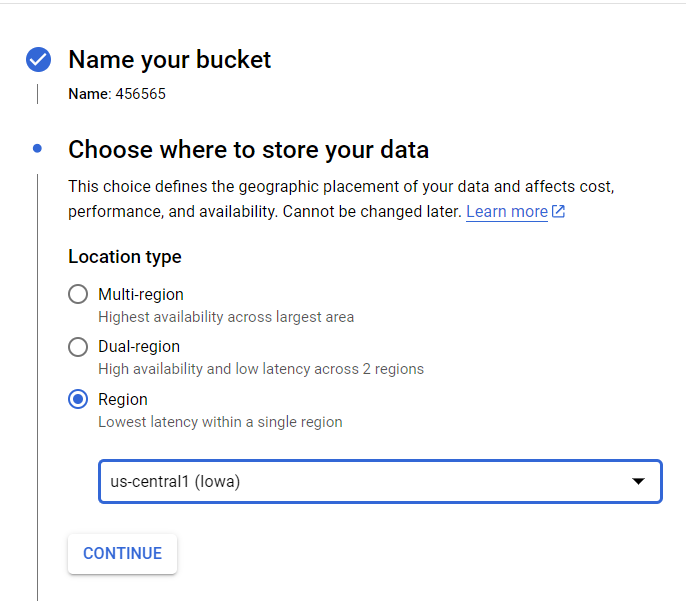
* Navigation Menu**→** Cloud storage **→** Buckets.

****

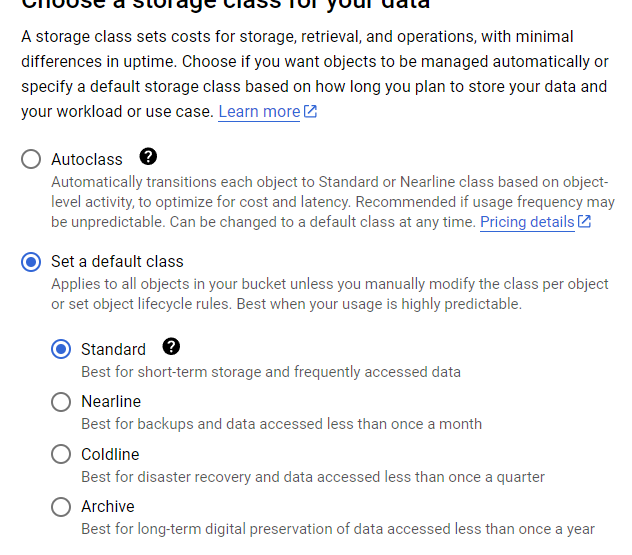
* Click Create.

****

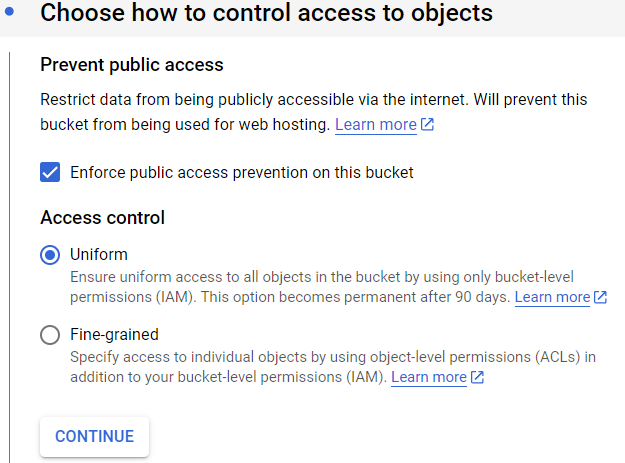
* Give your bucket a Unique name.
* Location Type **→ Region→ us-central1.**
* Click **→ Continue.**

****

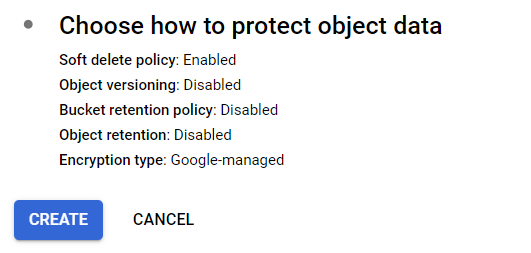
* For storage class → **Standard.**

****

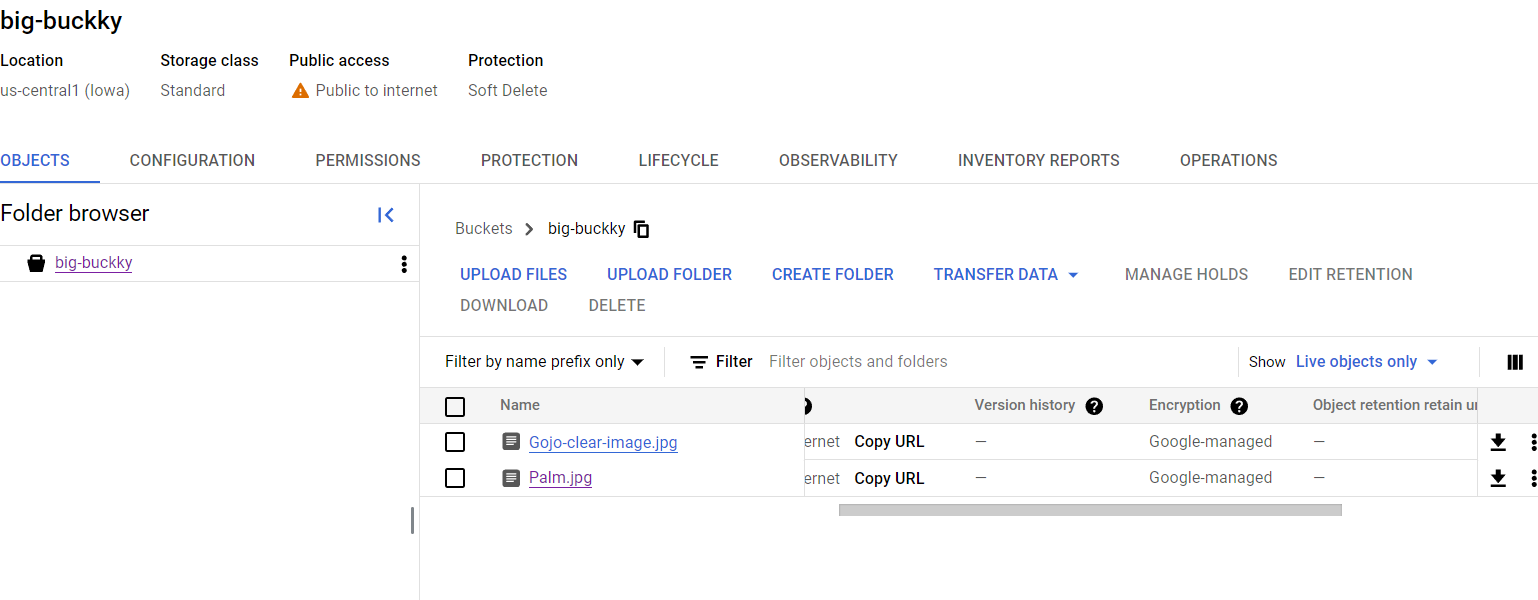
* Access Control → **Uniform**
* Uncheck **→ prevent public access**.

****

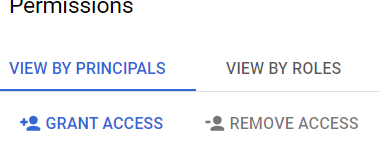
* Accept the default for choosing how to protect object data.
* Click → create.

****

* Your bucket will resemble as following.
* Click on → **Upload files.**
* Upload the file from your local computer.

****

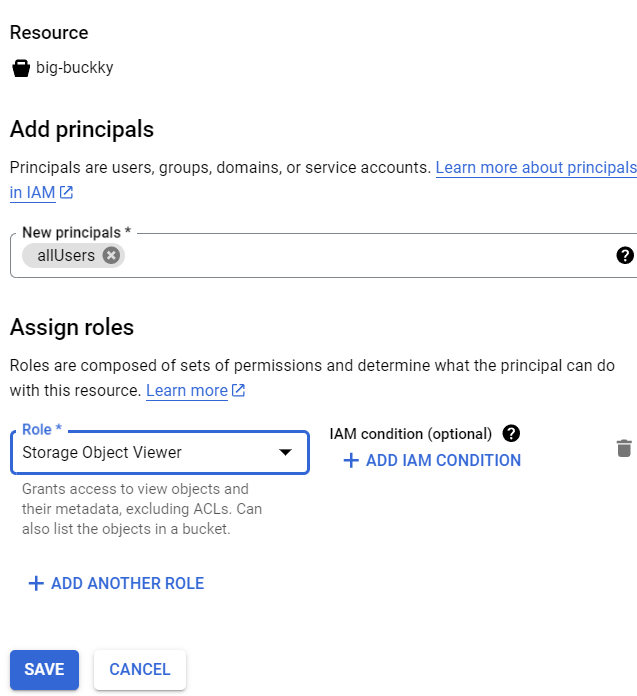
* On The same page Navigate to → **Permissions.**
* Click → **Grant access.**



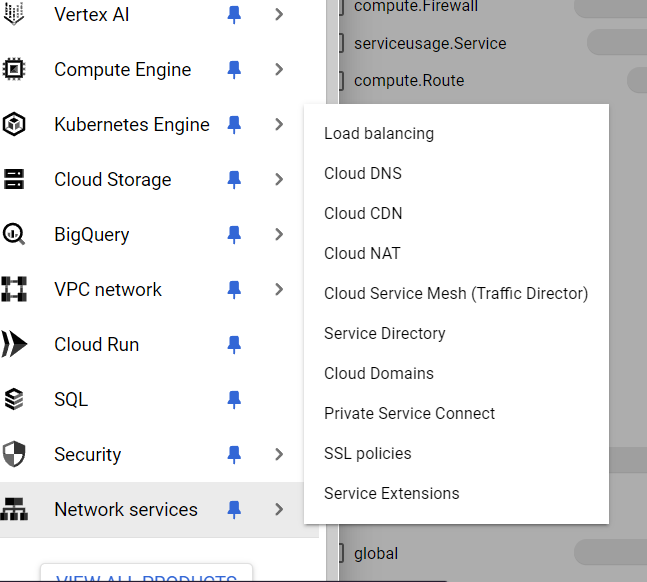
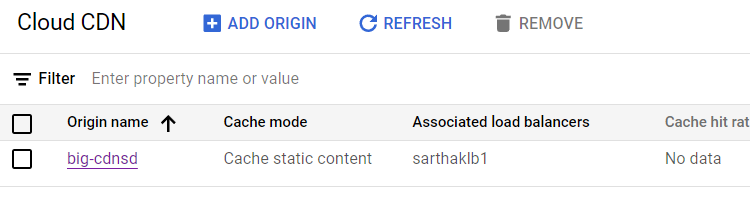
* Add principals → all users
* Roles → Cloud storage → storage object viewer.
* Click → Save.

**Now your image is accessible to the public. copy the URL and paste it into the browser to check.**

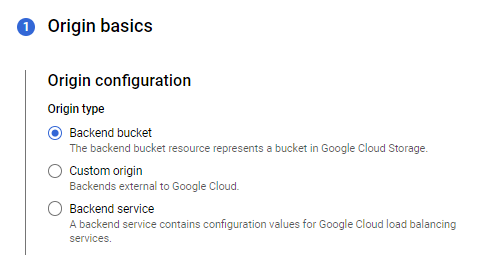
[**(click here )**](https://storage.googleapis.com/big-buckky/nervescape.jpg)



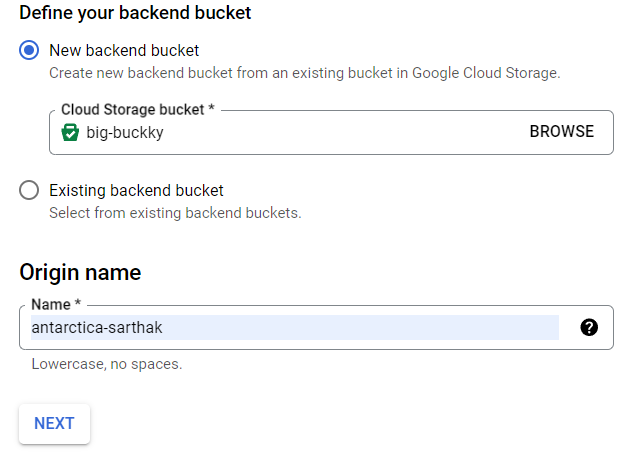
* **Configuring CDN**
* Navigation Menu → Network services →CDN**.**



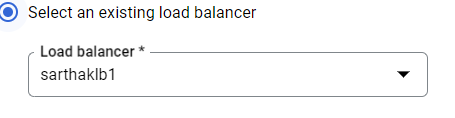
* Click On **Add Origin.**
* Origin type **→ Backend Bucket.**



* Click on browse and select the bucket you just created.
* For Origin Name give a unique and appropriate name.
* Click **next**.



* Select your load balancer.



* Click on Add **host and pat**h rules.

IT will determine how your traffic will be directed.

* Click on **+add host and path rule** and in **hosts** enter your domain name (eg. sarthakdeshmukh.in.net) and in **paths** enter the path for your image file in bucket (e.g. /map.jpg).
* Select the default settings for Cache performance and click on **DONE**.

A screenshot of a computer program

Description automatically generated

**We have completed our DNS setup.**

**Open a new tab and paste your domain name and path of your image like** [**https://sarthakdeshmukh.in.net/nervescape.jpg**](https://sarthakdeshmukh.in.net/nervescape.jpg)

****

**NOTE -It might take some time to load the image.**

**If you have any doubts feel free to connect with me (**[**LinkedIn**](https://www.linkedin.com/in/sarthak-deshmukh-5756a329a?lipi=urn%3Ali%3Apage%3Ad_flagship3_profile_view_base_contact_details%3BOhSoc%2BeYTKKA%2FkSCgqhVIQ%3D%3D)**)**

