INSTALLATION

OF

EBER

INDEX

Installation of EBER

(I) Things need to be Pre-installed.

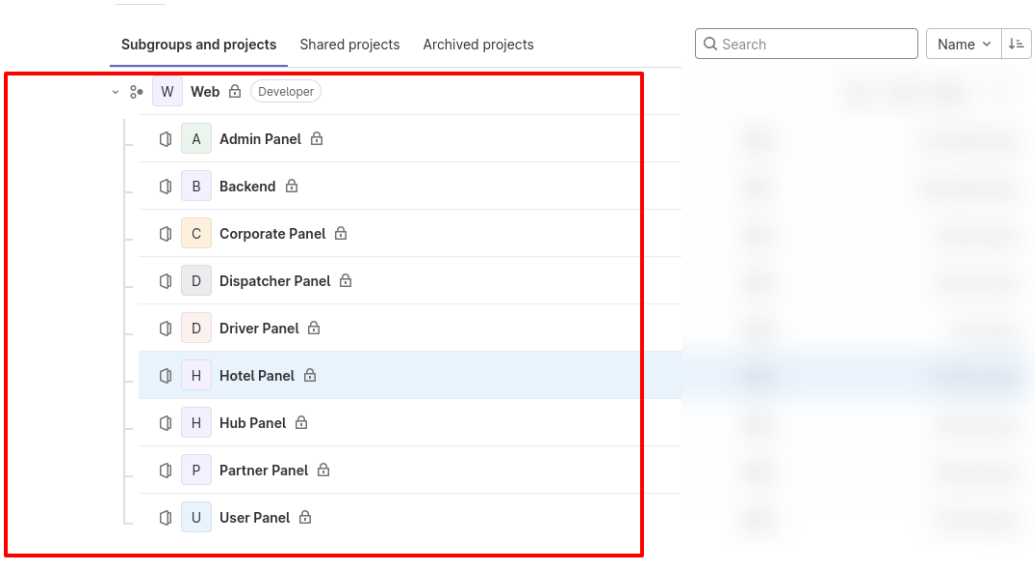
- Ubuntu OS (minimum version 20.04)
- Code Editor (any)
- FileZilla (for transferring files)
- Putty (for connecting Server)
- MongoDB Compass or Robo-3t (for connecting DB)

(II) Before we move further, things to consider.

(A) Project on Server

(i)Things to keep handy.

- Purchased Domain
- Confirm repo access, as listed below.



- Follow every step mentioned from Section 1 of this document.

(B) Project in Local

(i)Steps to Skip

- Skip all the following steps from section 1 of this document.

(ii)Perform the following steps

- Download the **EBER_INSTALLATION.sh** file.
  - Ask for this bash file or you can get it directly from the following location, navigate to **backend/server/documentation/installation documentation/*EBER_INSTALLATION.sh***
- Navigate to the path of downloaded **sh** file and open the terminal using right-click and run the command ***bash EBER_INSTALLATION.sh***
- Press 1 for full installation
- Skip the following mentioned steps in the process of running bash file.
  - Step-10
  - Step-12
  - Step-13
- After the completion of the process navigate to the folder using the file system where you want to clone the project and then open the terminal using the right click.
- Follow the steps to clone the code locally .
  - Create directory using ***mkdir <ProjectName>***
  - For Frontend(Admin, Corporate, Driver, Dispatcher, Hub, Hotel, Partner and User).
    - 1. Clone using ***git clone <https git url>***
    - 2. Enter Username And Password
    - 3. Enter the cloned folder ***cd <FolderName>***
    - 4. Install dependencies using ***npm i***
    - 5. Exit that folder ***cd ..***
    - Repeat the above step(1,2,3,4,5 steps) for all the frontend git project urls.
  - For Backend(Backend).
    - 1. Clone using ***git clone <https git url>***
    - 2. Enter Username And Password
    - 3. Enter the cloned folder ***cd <FolderName>***
    - 4. Enter the folder ***cd <FolderName>***(server, history, payments and mass notification)
    - 5. Install dependencies using ***npm i***
    - 6. Exit that folder ***cd ..***
    - Repeat the above step(4,5,6 steps) only for the backend git project url

- Navigate to the project and run the following commands:
  - Navigate to the **initial_data.js** file to fill all the basic data in DB.
    - backend/server/settingsdata/initial_data.js
    - ***node initial_data.js***
  - Navigate to the **angular.json** file of all the panels other than the backend and replace the "**dist_new**" with "**dist**" by searching the field **"outputPath"**

(C) Important notes.

(i) Default admin creds

- Username **"eber"** password **"developertest123abcxyz@"**

(ii) Important commands

- For Running the panels in local
  - ***npm start***
- For Running the backend in local
  - ***npm start***
- **For serving all the panels and backend code on the server, the bash file will do everything but in case you may need to know the command to serve the code, below code can be used for reference.**
- For Running the panels on Server
  - Need to upload the build and then run the server.js file after navigating to each panel folder..
  - ***pm2 start server.js --name <PanelName>***
  - The above command needs to be followed for every panel.
- For Running the backend on Server
  - Navigate to the backend folder one by one, ie. server, history-earning, payments, mas_notification and use the following command.
  - ***pm2 start server.js --name <BackendName>***

Note: If you run the panels using the **npm start** command it will use the environment.js file and if we run the panels using **npm run build** command it will use the environment.prod.js file.

1. Purchasing Server

Follow the steps to purchase instance:- Below there are total 3 ways to purchase and

create instance on cloud,

i.e: AWS(1.1) or Digital Ocean(1.2) or Vultr(1.3) you can choose any of the

mentioned cloud services according to your convenience.

1.1 Purchasing Server On AWS

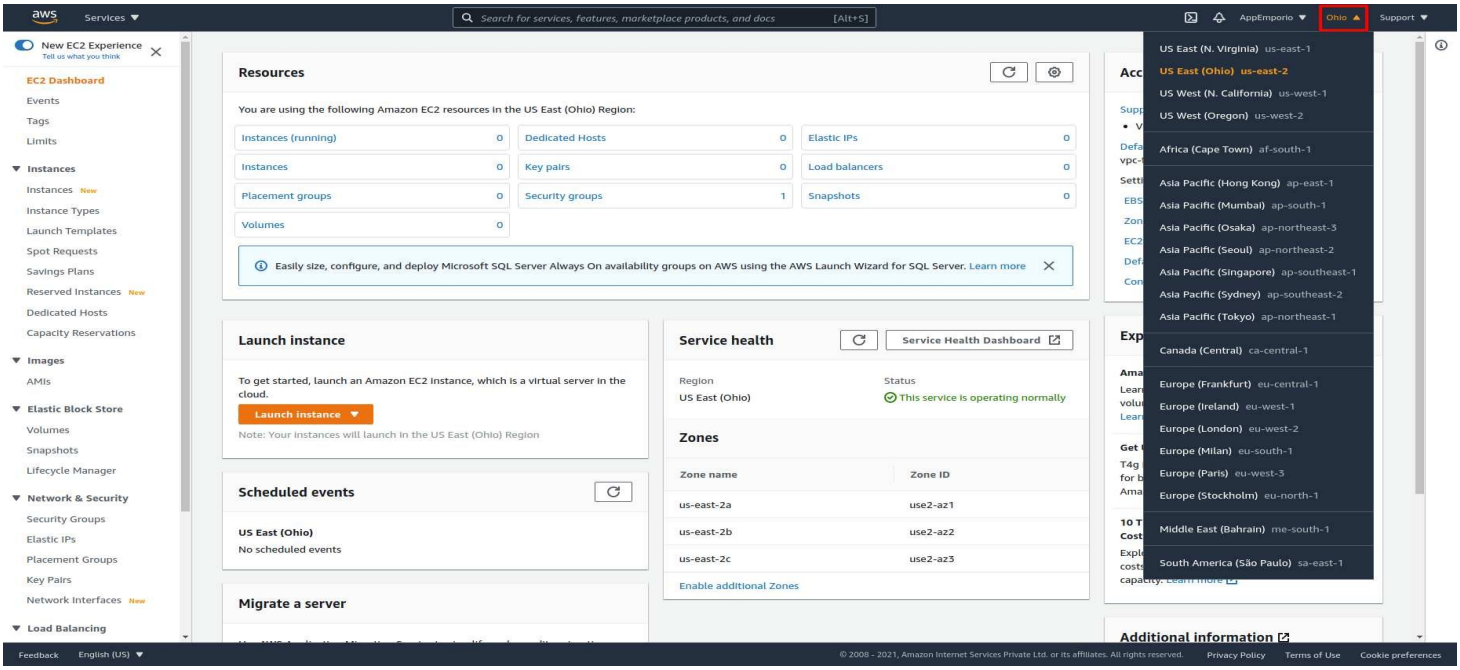
Create an account on AWS and complete basic registration steps that are required to get done before we can access AWS benefits.

After registration now we need to purchase an instance(server) where we can install our code. So for that, we need to purchase instances.

Register here :- AWS account Register Login here :- as root AWS account Login

1.1.1 Selecting region for instance

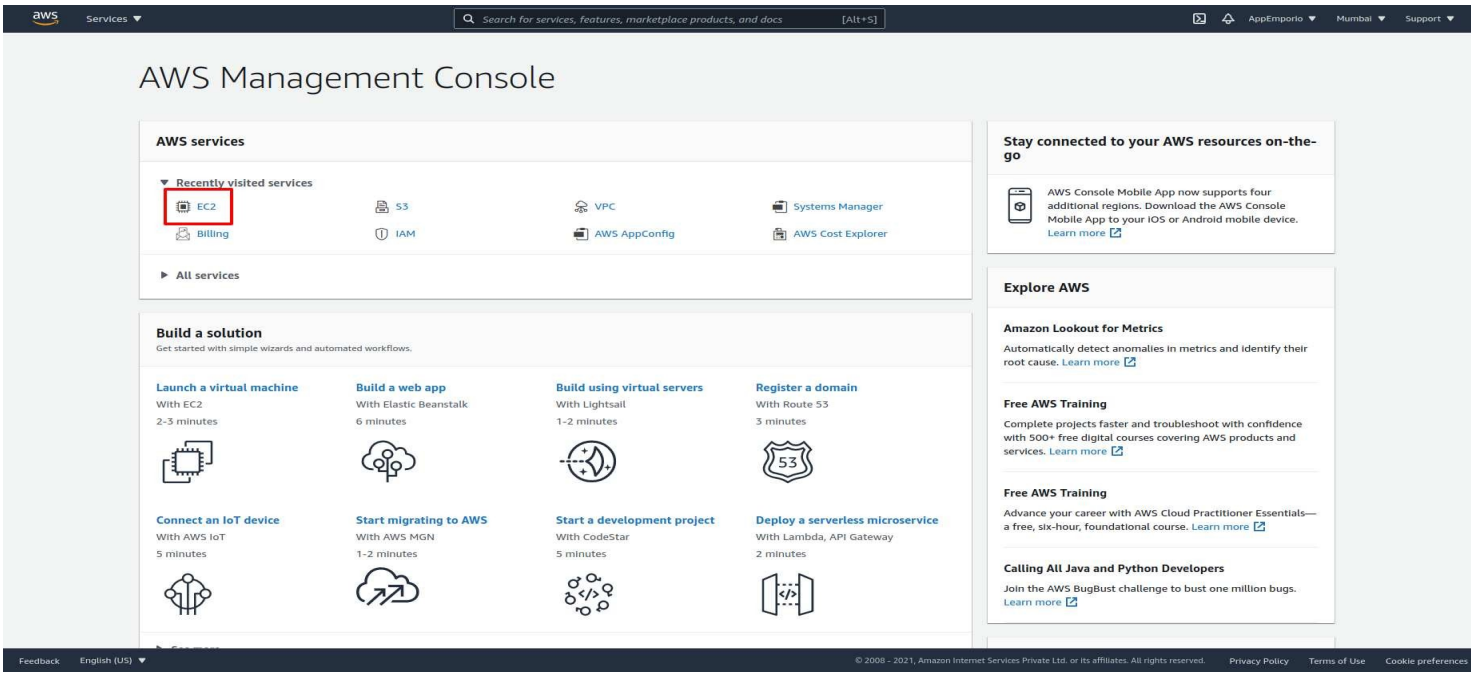
- We need to select the region (from the top right corner) wisely so that we get the best latency and API responses in less time. Select the region where this application is going to be used most of the time.
- If a region is not available in a particular country or state, then use the region which is nearest.



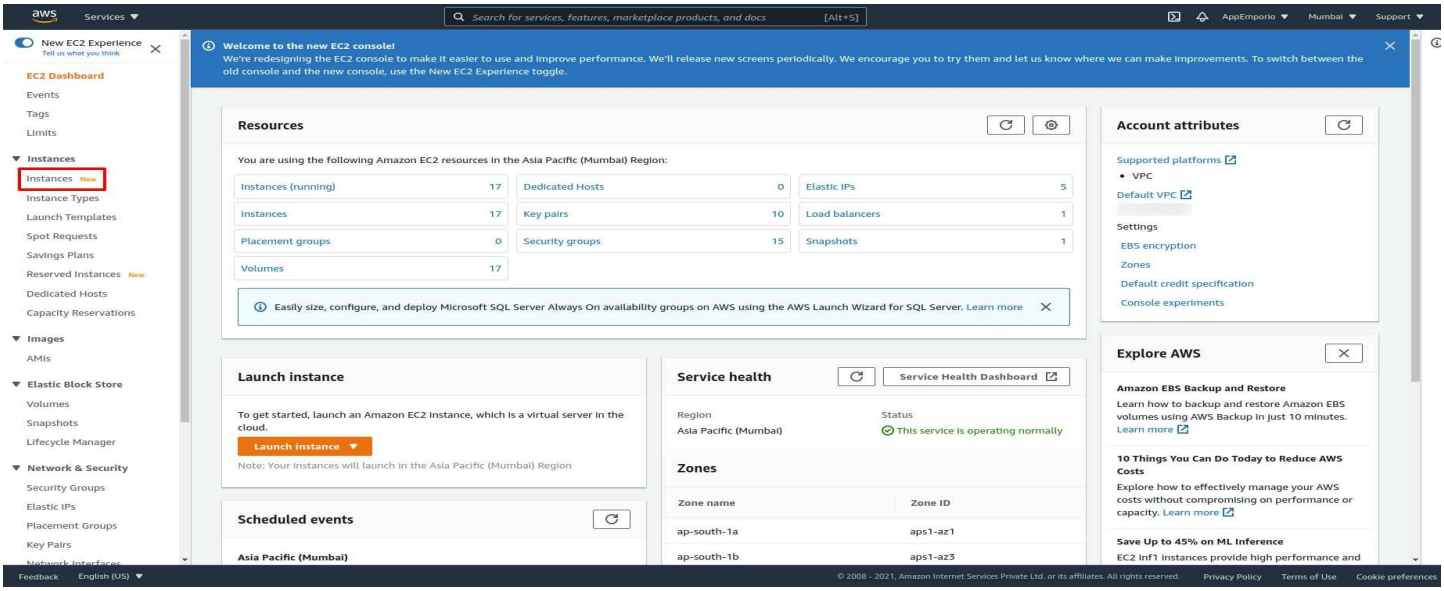
1.1.2 Purchasing Instance

1.1.2.1: Select EC2.

- Navigate to the EC2.
    - Search for **EC2**
    - Select EC2

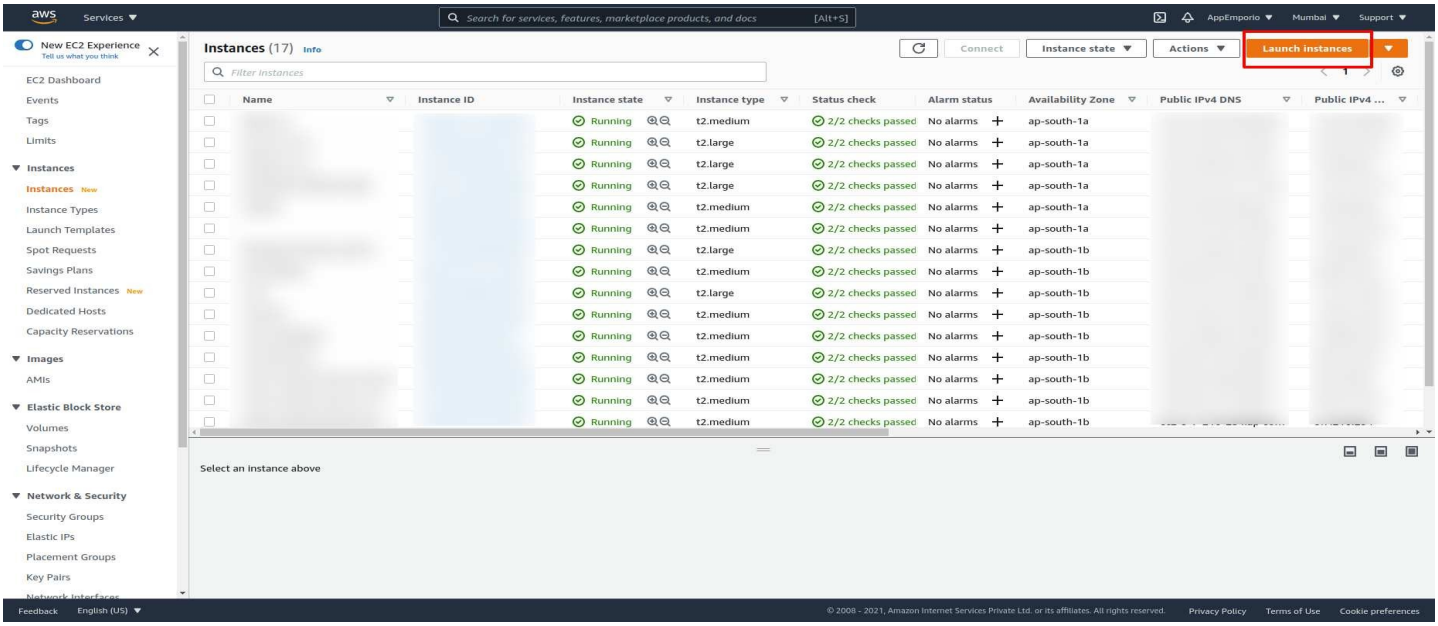


1.1.2.2: Select instances.



1.1.2.3: Select Launch

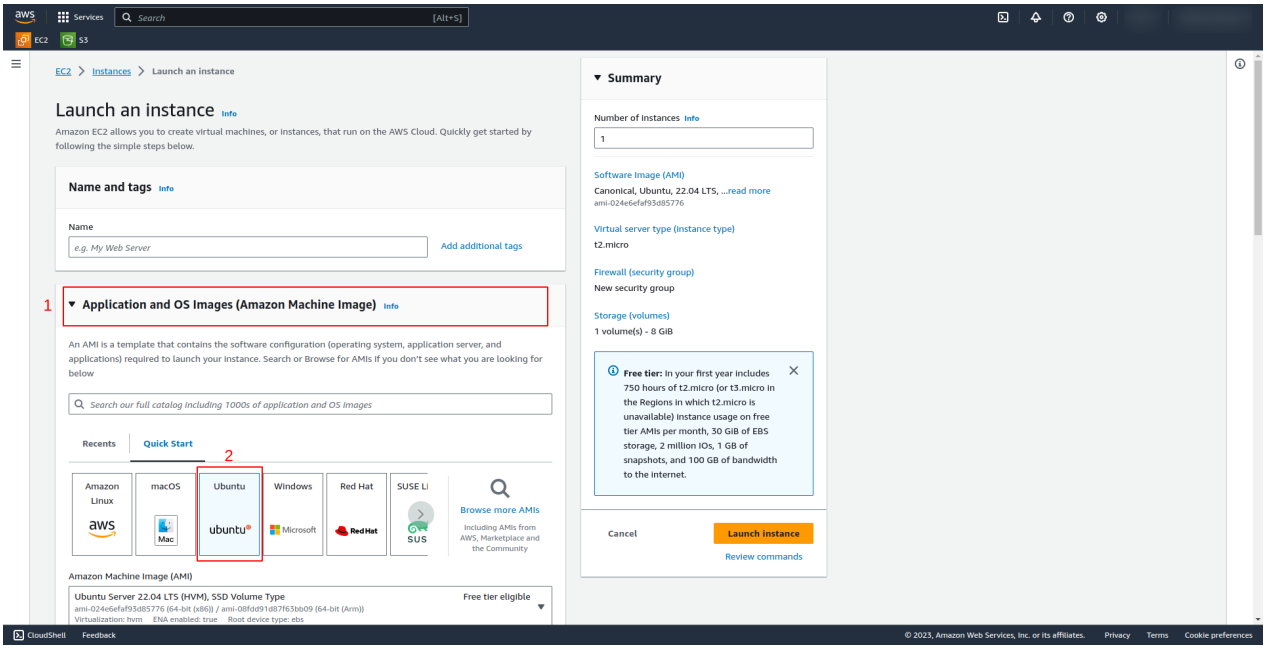
Select Launch from the top right corner.



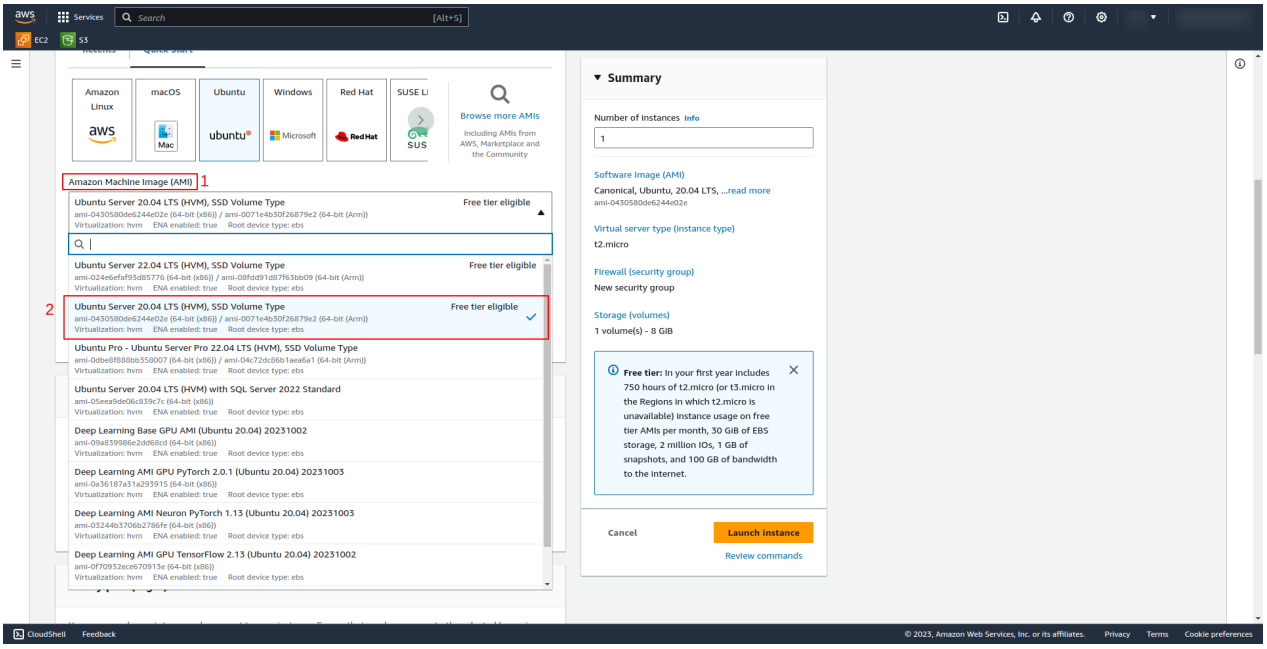
1.1.2.4: Launching our instance

1.2.4(i) Choose an AMI

- AMI is a bunch of basic preloaded software configurations like Operating Systems.

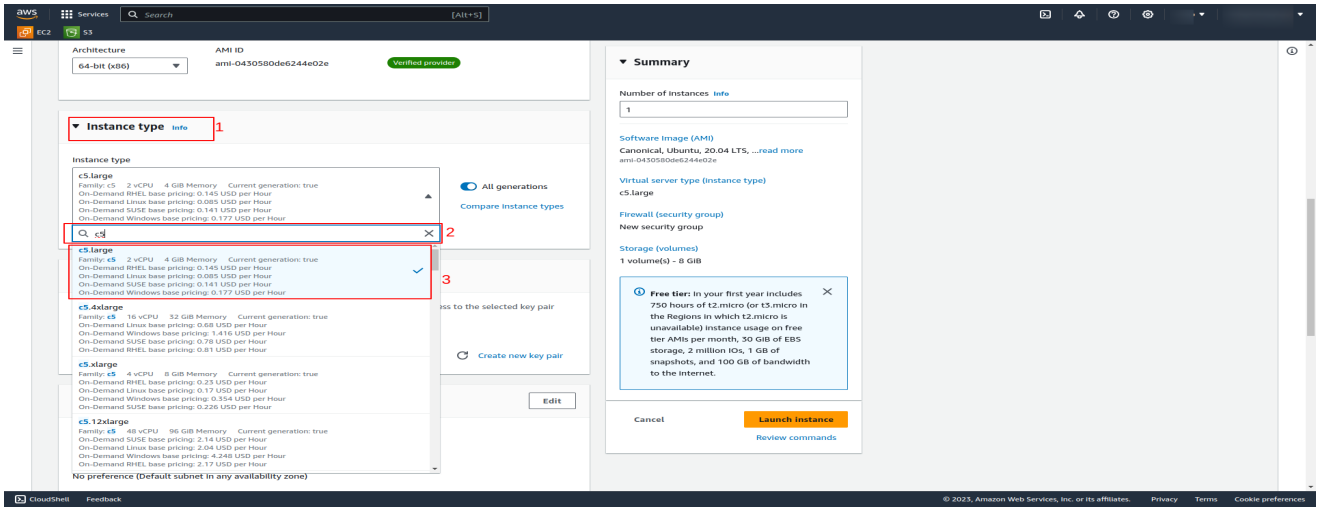


- Select Ubuntu which is most preferable according to our use case(minimum version 20.04).



1.1.2.4(ii) Choose instance type

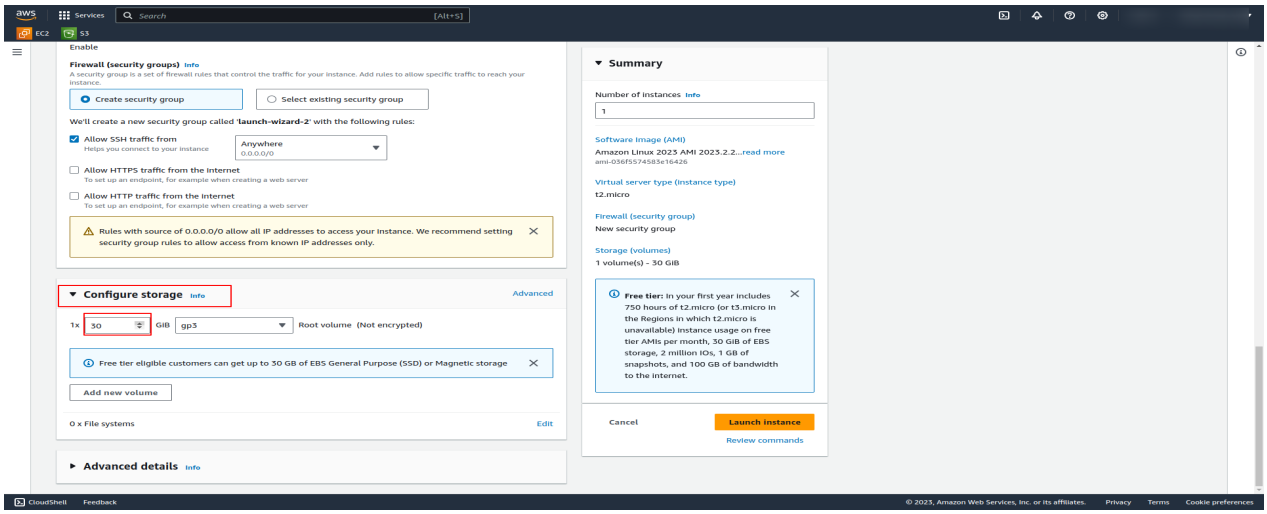
- Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications.
- Preferable selection c5.large for backend



1.1.2.4(iv) Configure storage

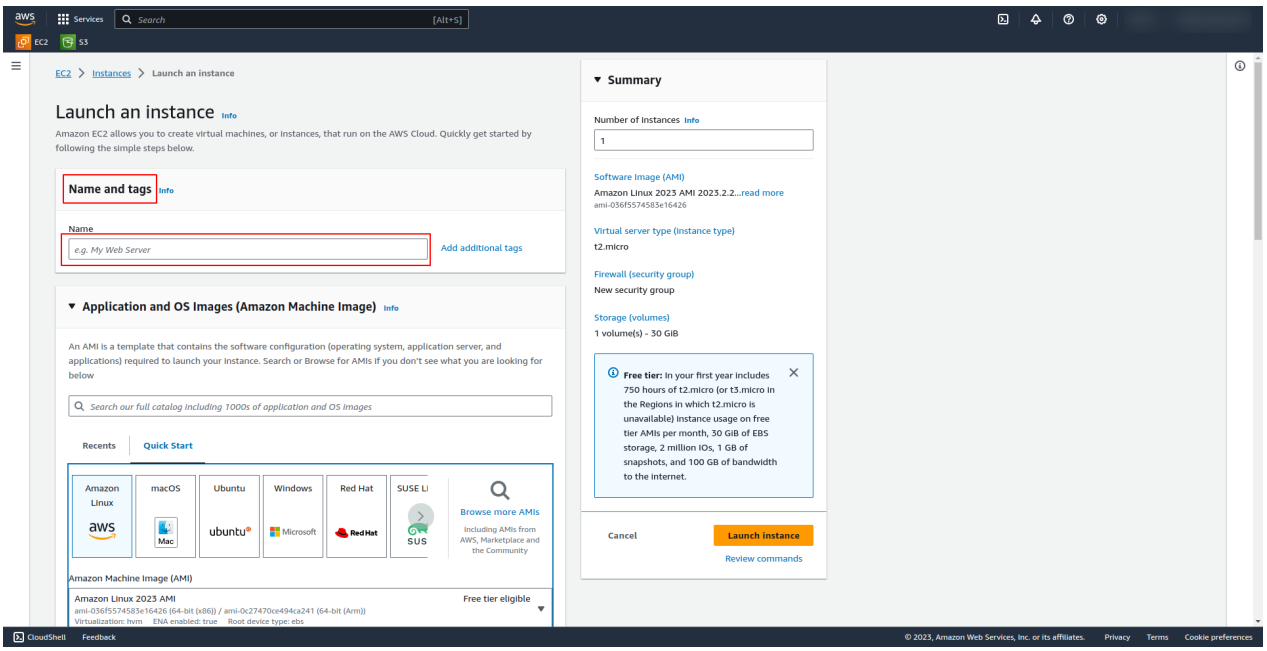
- You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans.

- Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Select 30 GiB.



1.1.2.4(v)Name and Tags(Optional)

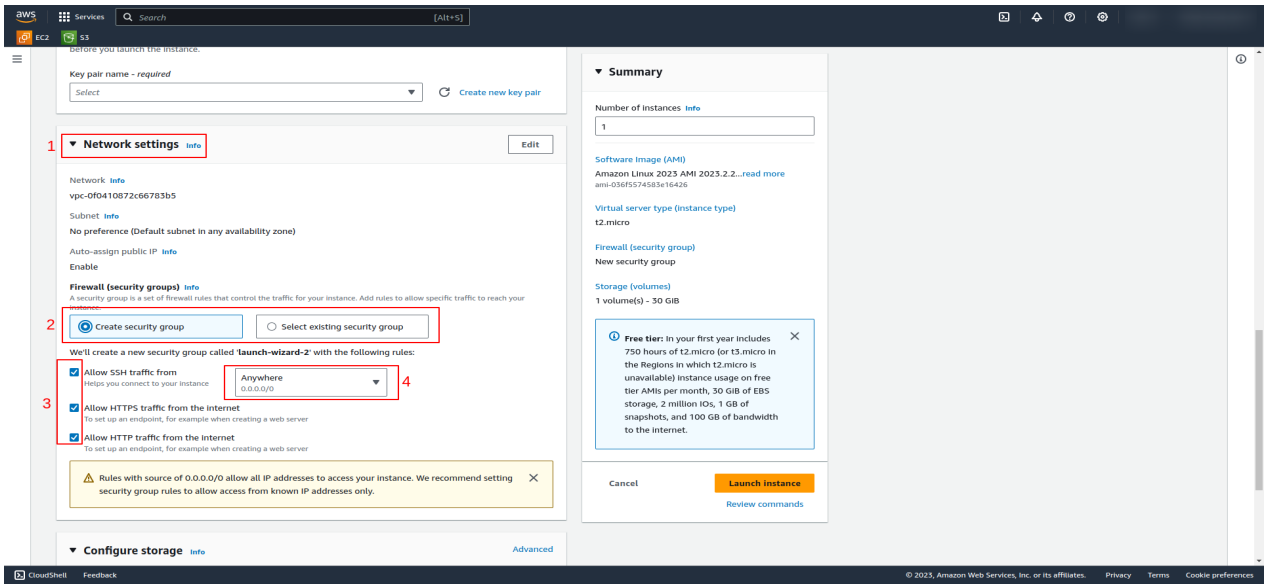
- Give the name to the instance.
- A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Web Server. A copy of a tag can be applied to volumes, instances or both.
- This step is optional for us.



1.1.2.4(vi) Configure security group

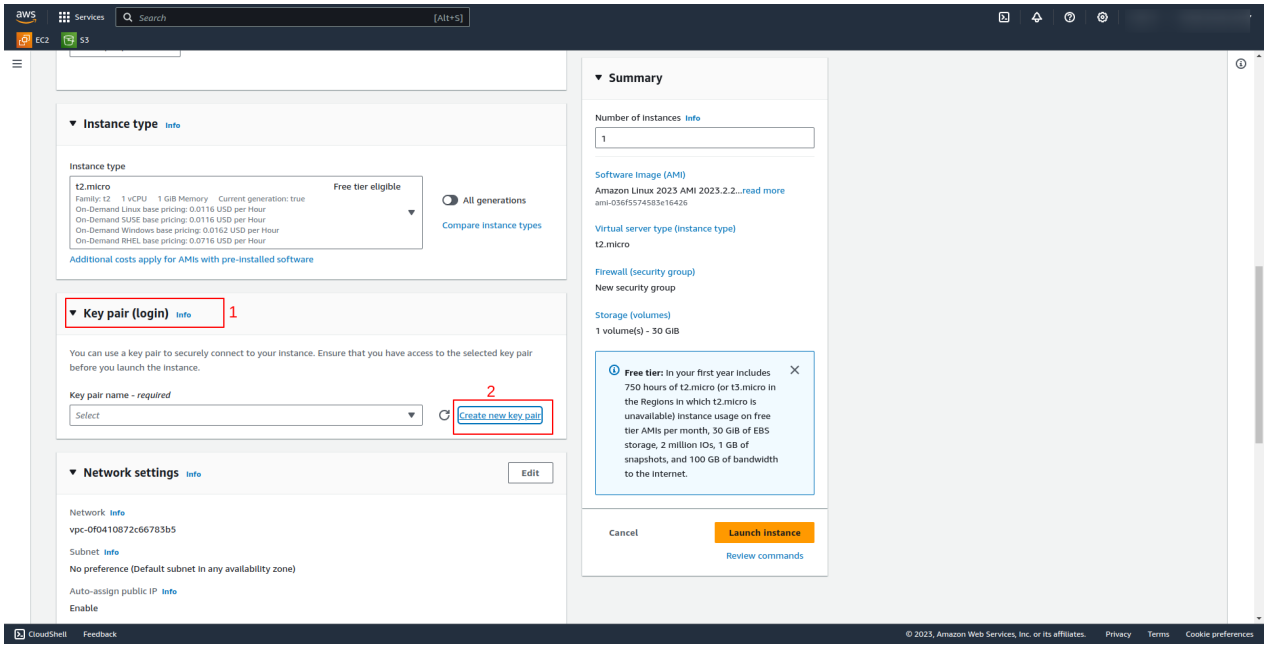
- A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance.
- We will require SSH, HTTP, and HTTPS security groups to have ssh connection and http and https connections from web browsers.

Create a new security group or select the existing one.
If creating a new security group, follow the steps mentioned below and select the option anywhere at step 4.

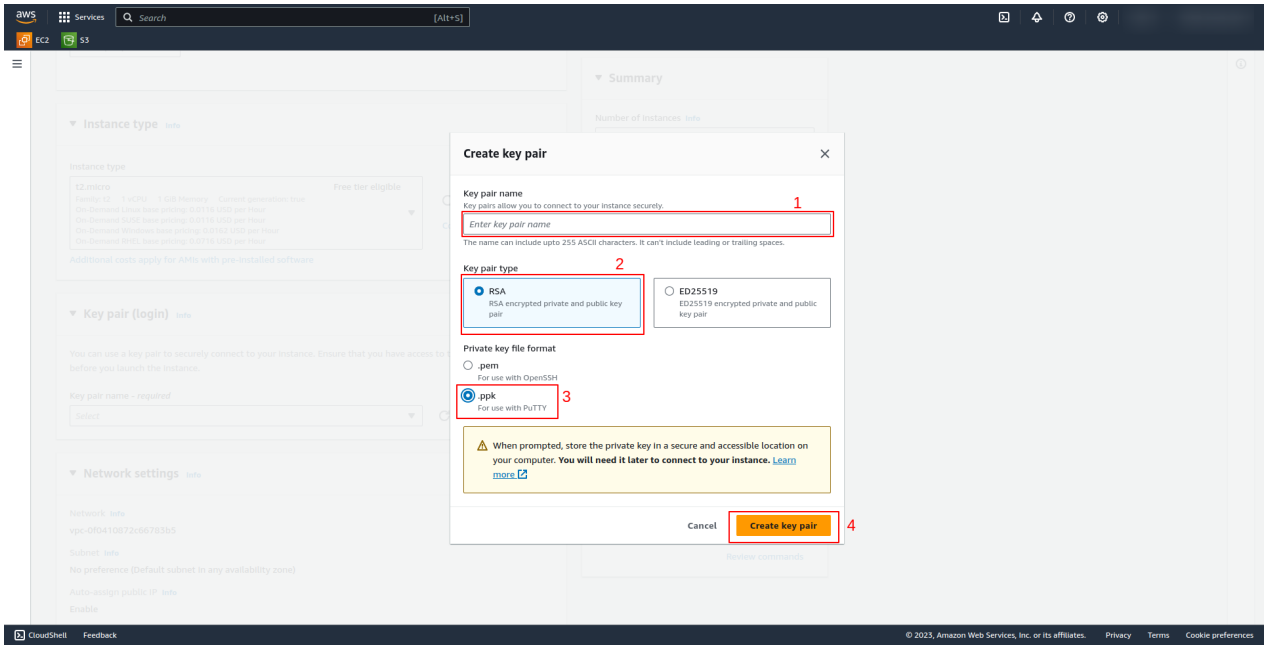


1.1.2.4(vii) Security Keys

- As shown in the image below while reviewing we can see a popup for security. If you have any security keys then select that or generate a new one for our server.
- For creating new security keys we need to select Create a new key pair and enter the appropriate name for your key and press download key pair(store it at a safe place as1 we will need it to connect the server).

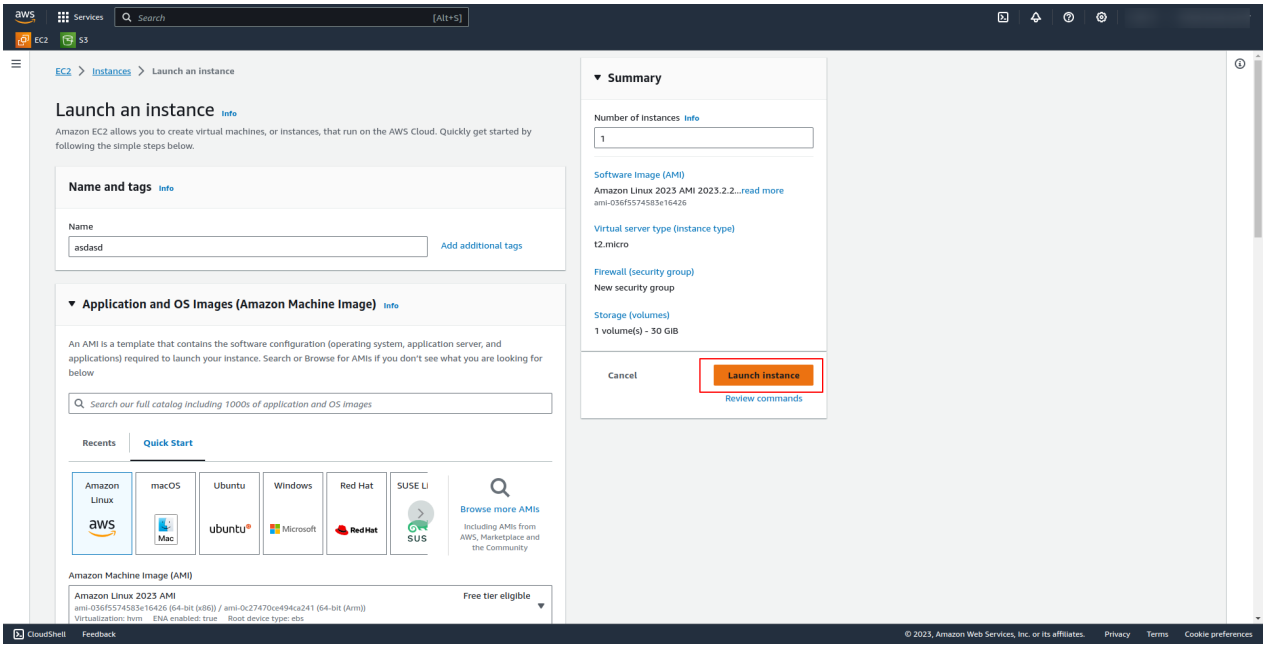


Follow the steps highlighted below.



1.1.2.4(viii) Launch Instance

After reviewing all the things properly click on the launch instance. After a few minutes the server will be up and running.

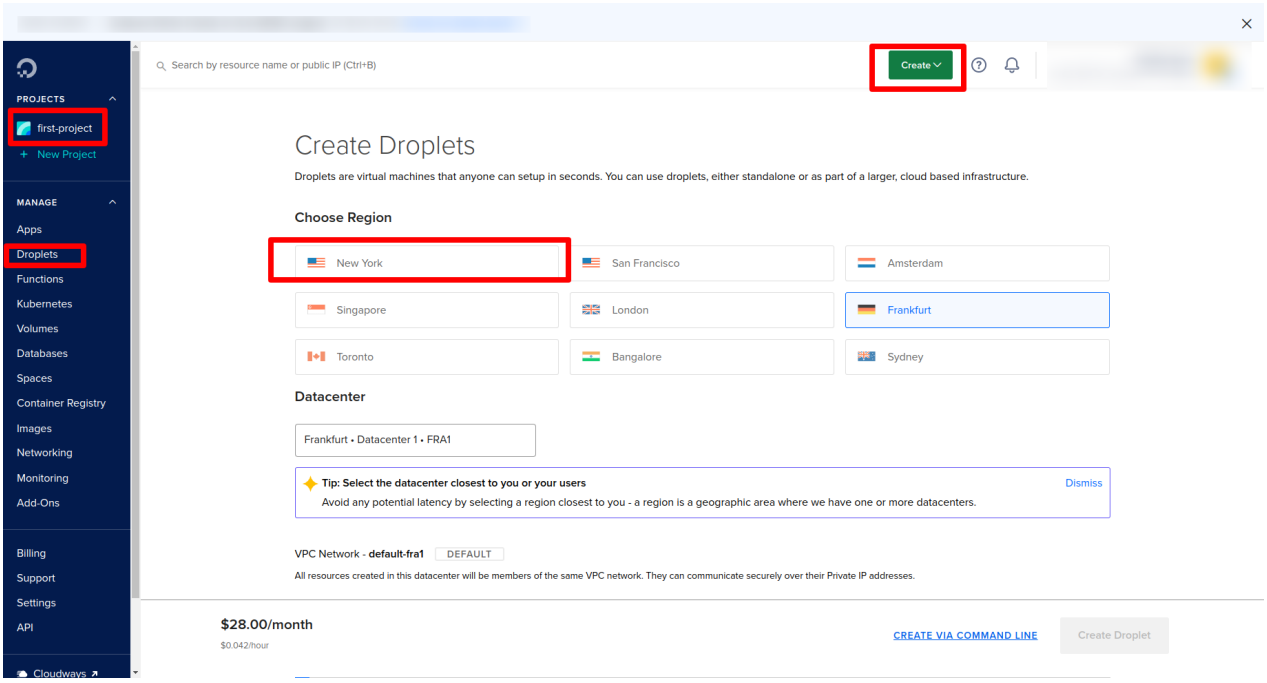

Services
Search
[Alt+S]
EC2
S3
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
asdasd
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 AMI
Free tier eligible
ami-036f5574583e16426 (64-bit (x86)) / ami-0c27470ce494ca241 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Summary
Number of Instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.2.2...read more
ami-036f5574583e16426
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 30 GiB
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

1.2 Purchasing Server On Digitalocean

Account Register or Login

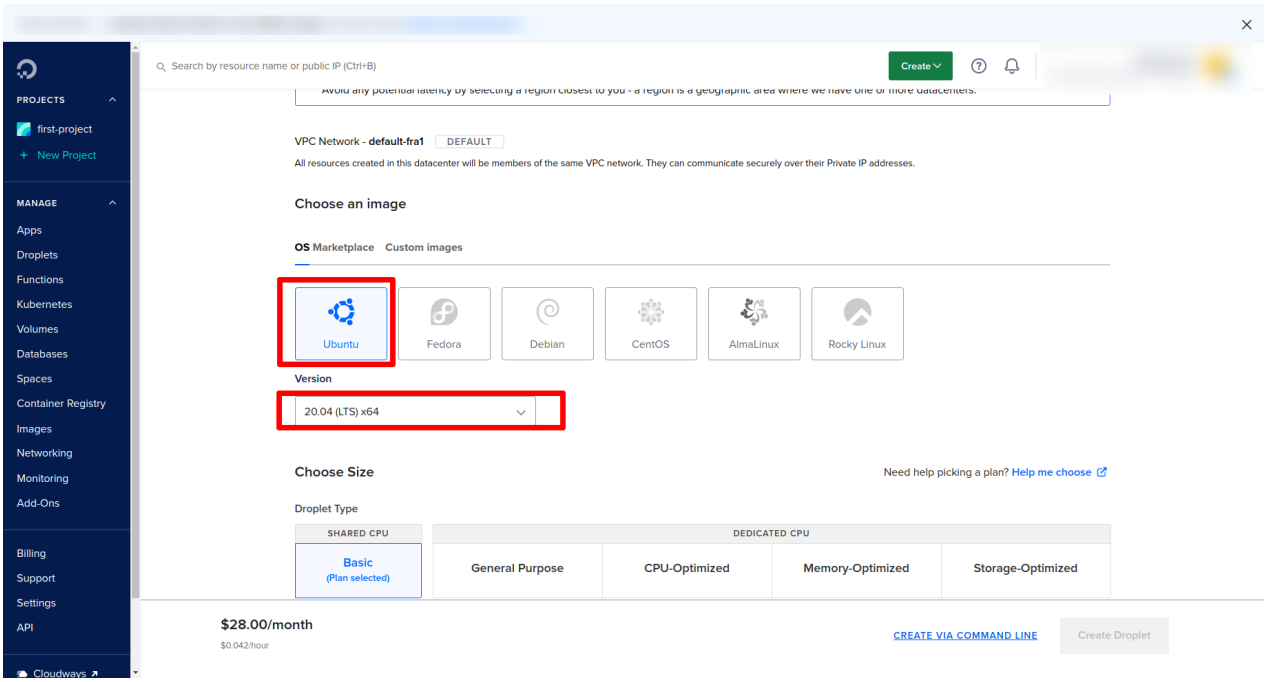
1.2.1) Creating a Droplet

choose organization, go to droplet, if there is no droplet. Create a new droplet with a create button. Choose your nearest region as per your requirements. Most of the time we go through new york region.



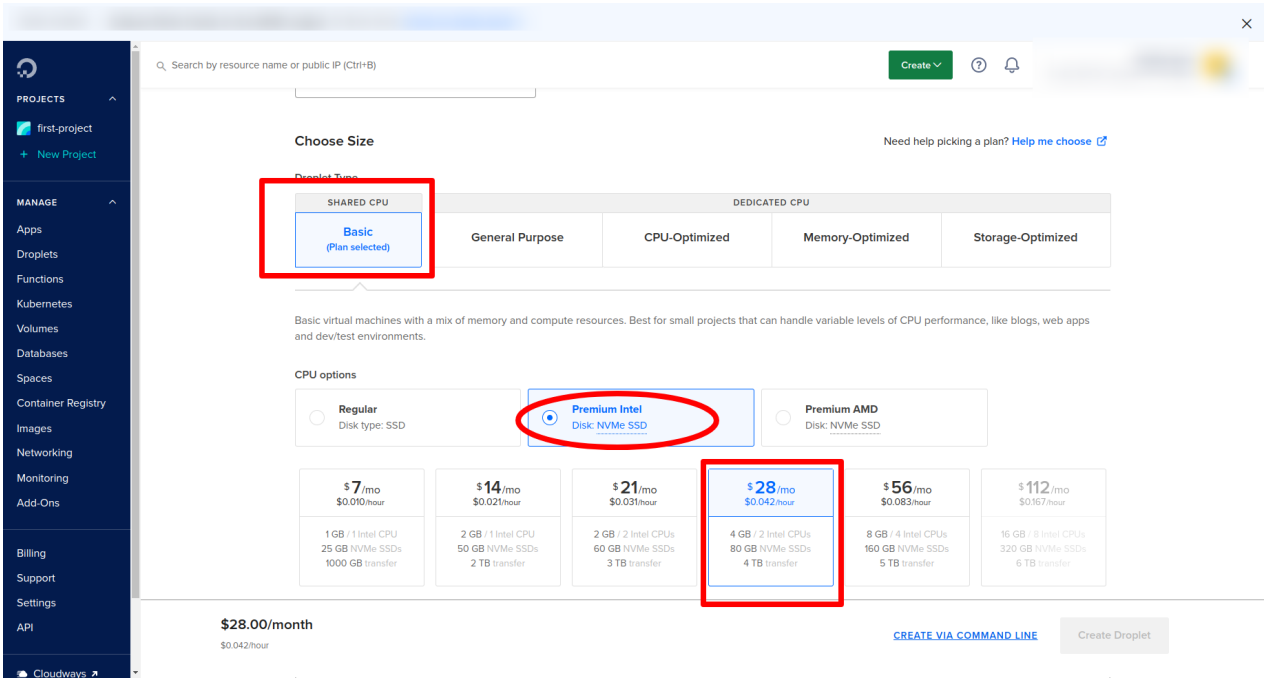
1.2.2) Choose OS

ubuntu 20.04 LTS



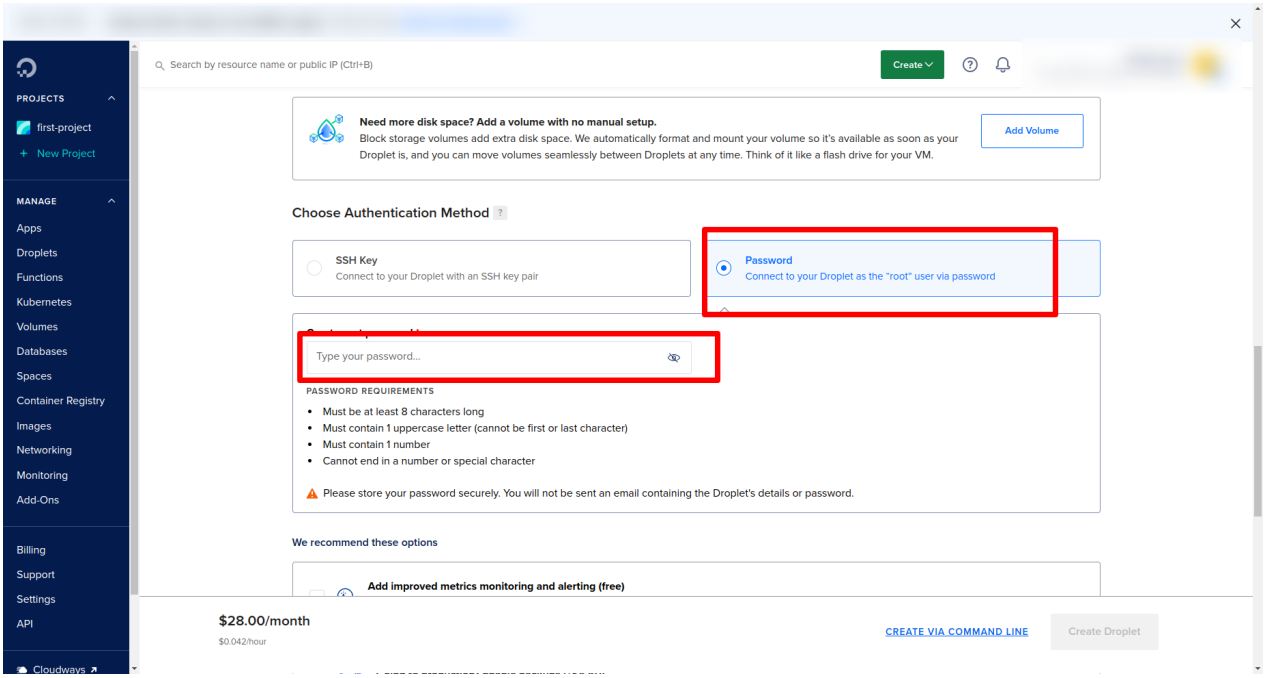
1.2.3) Choose size for droplet.

Go for a basic plan. Select premium intel CPU. choose 28$/month plan



1.2.4) Choose an authentication method.

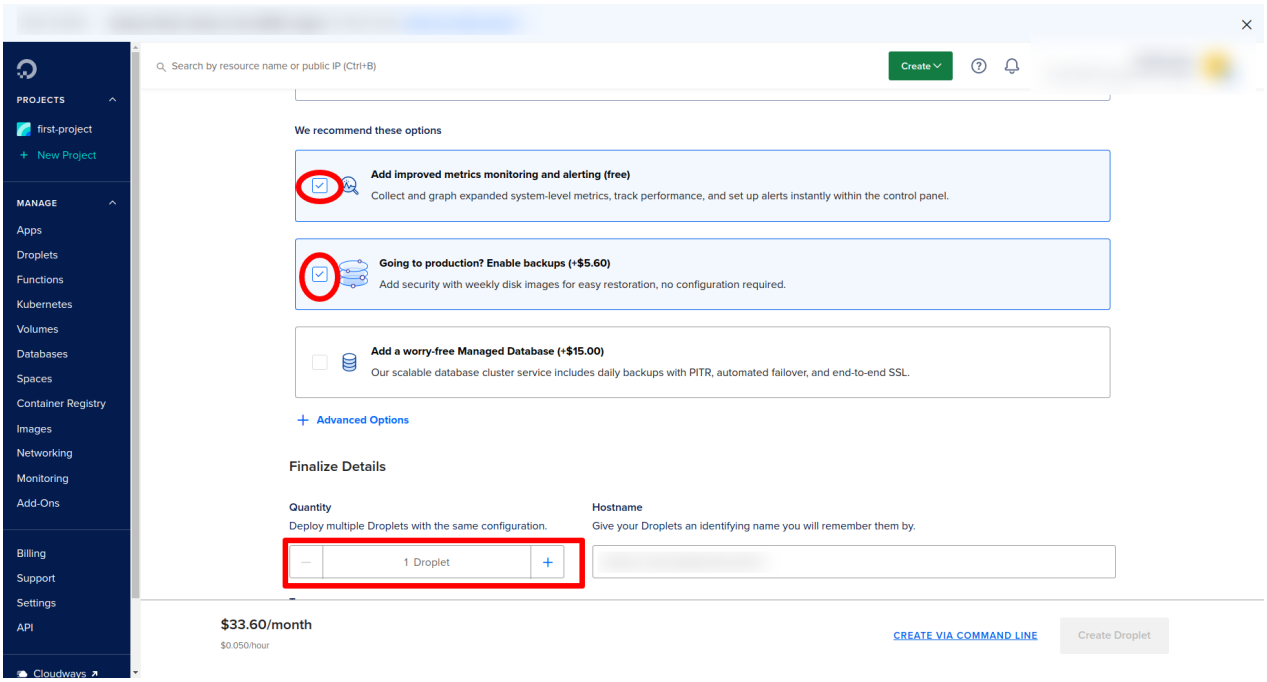
Go to the password method. Use root password. You can create your own or you can create a password with the help of password generator. It will give you a strong password (copy it and save it at safe place).



1.2.5) Check Recommendation

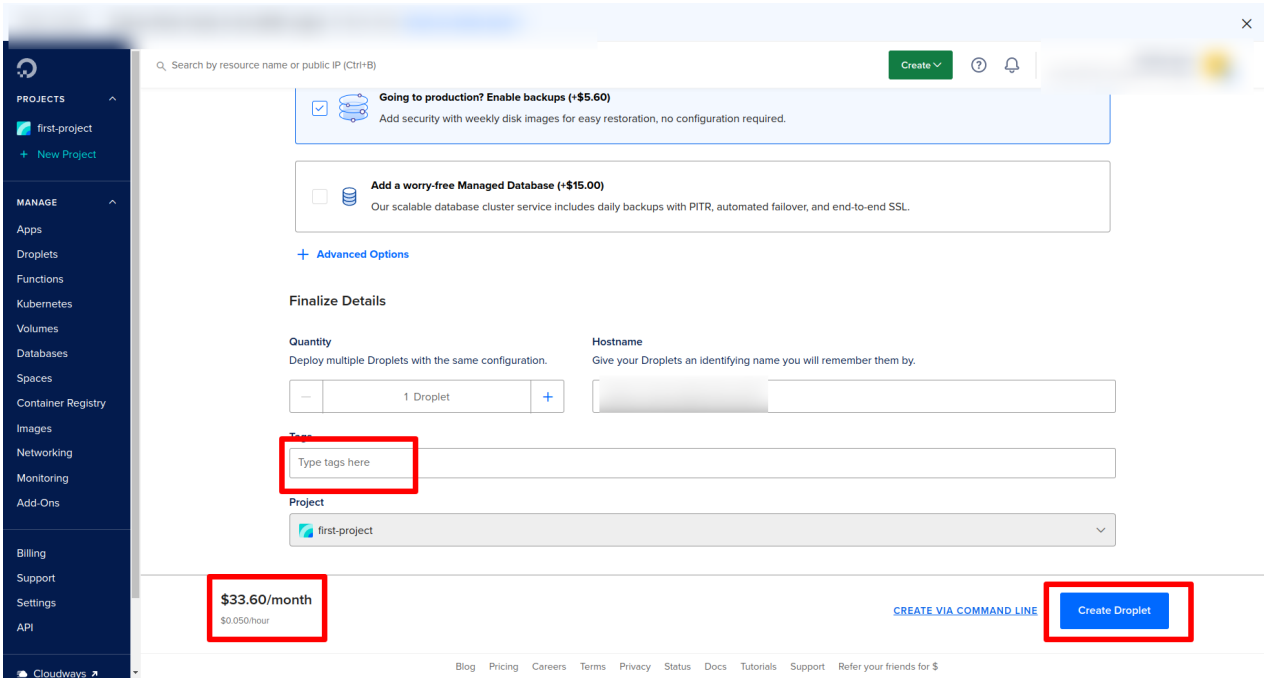
After creating a password check the first free recommendation option. The 2nd option is backup of the server. If want this will charge 5$

Quantity 1 droplet and auto create hostname



1.2.6) Optional add tags

If you go through this it approximately charges 33$. Hit create droplet button



On success creation of droplet you will redirect the droplet section where you get ip and password(which you created before while creating droplet). This credential will help you to connect with server.

1.3 Purchasing Server On Vultr:

Create an account on Vultr and complete basic registration steps that are required to get done before we can access Vultr benefits.

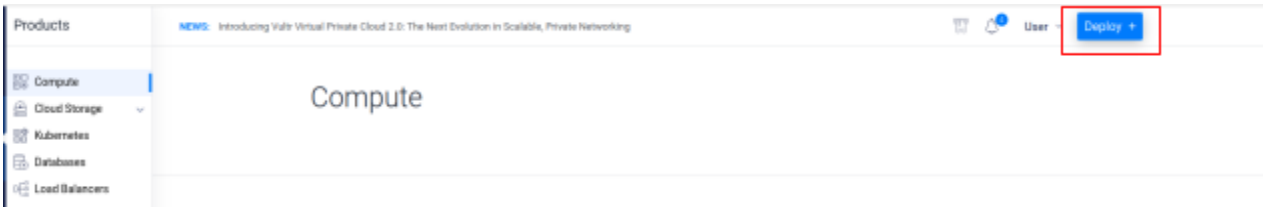
After registration now we need to purchase an instance(server) where we can install our code. So for that, we need to purchase instances.

Register here :- Vultr account Register Login here :- as root Vultr account Login

1.3.1 Create instance on Vultr dashboard

- We need to select the region wisely so that we get the best latency and API responses in less time. Select the region where this application is going to be used most of the time.

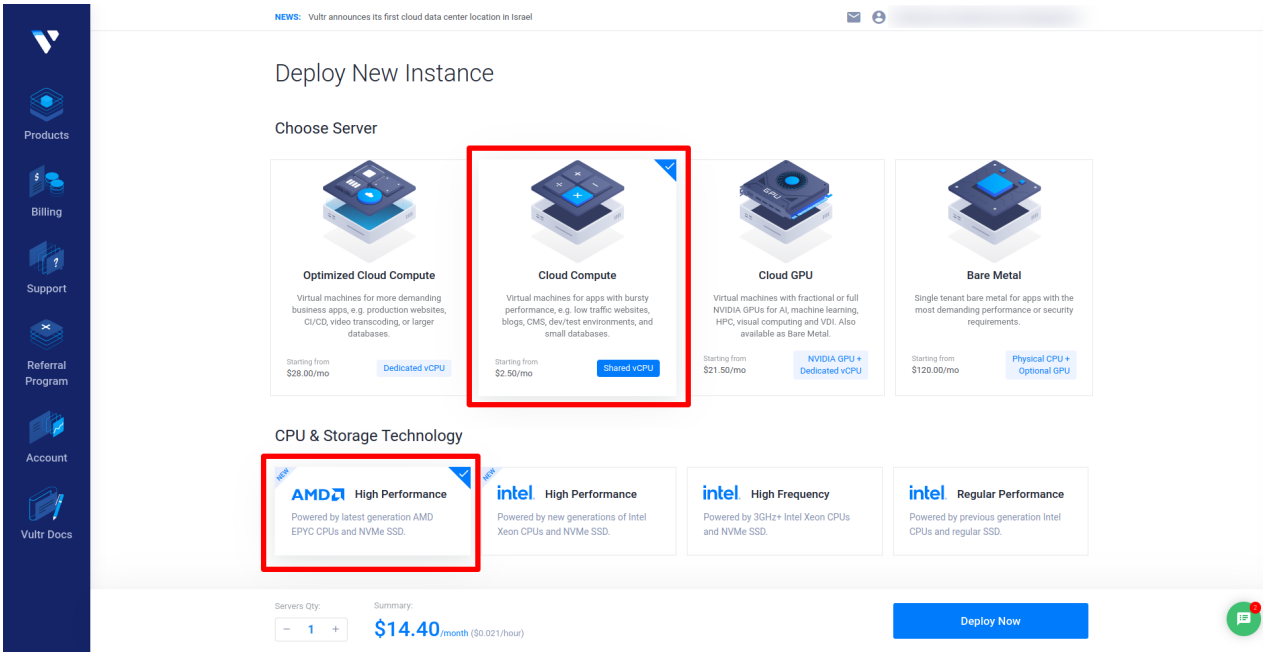
- If a region is not available in a particular country or state, then use the region which is nearest.



1.3.2 Select the Server and CPU

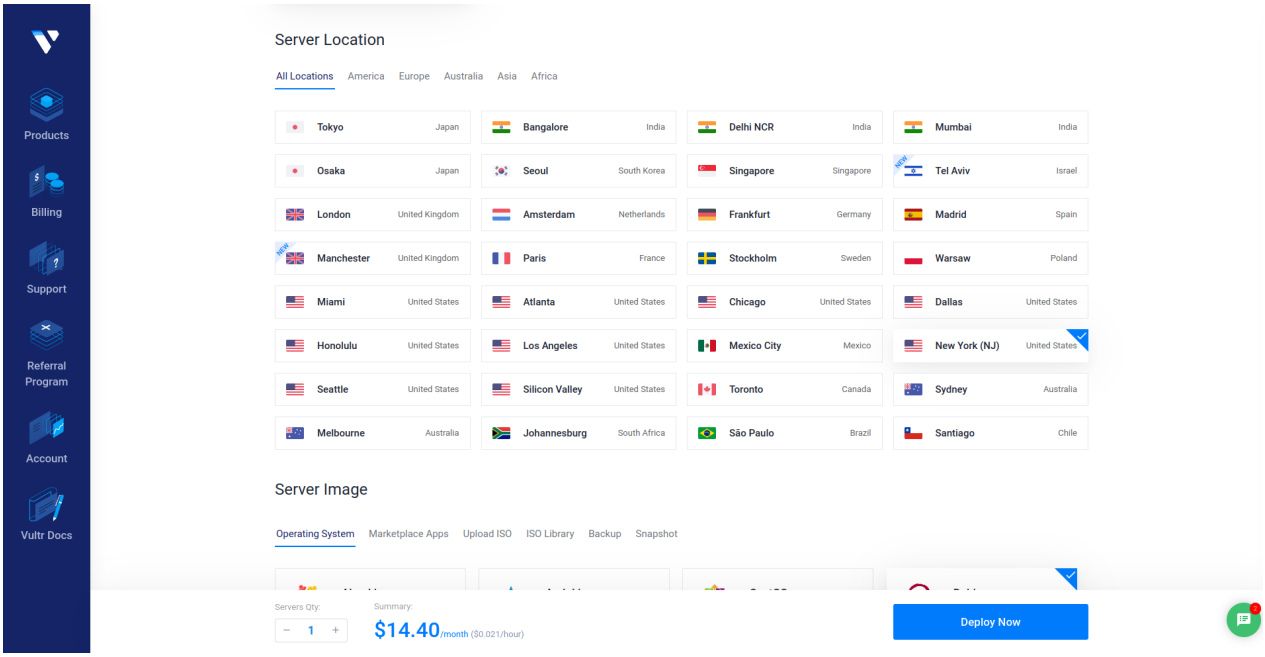
1.3.2.1 Choose Server

Choose the server accordingly as mentioned in the screenshots below.



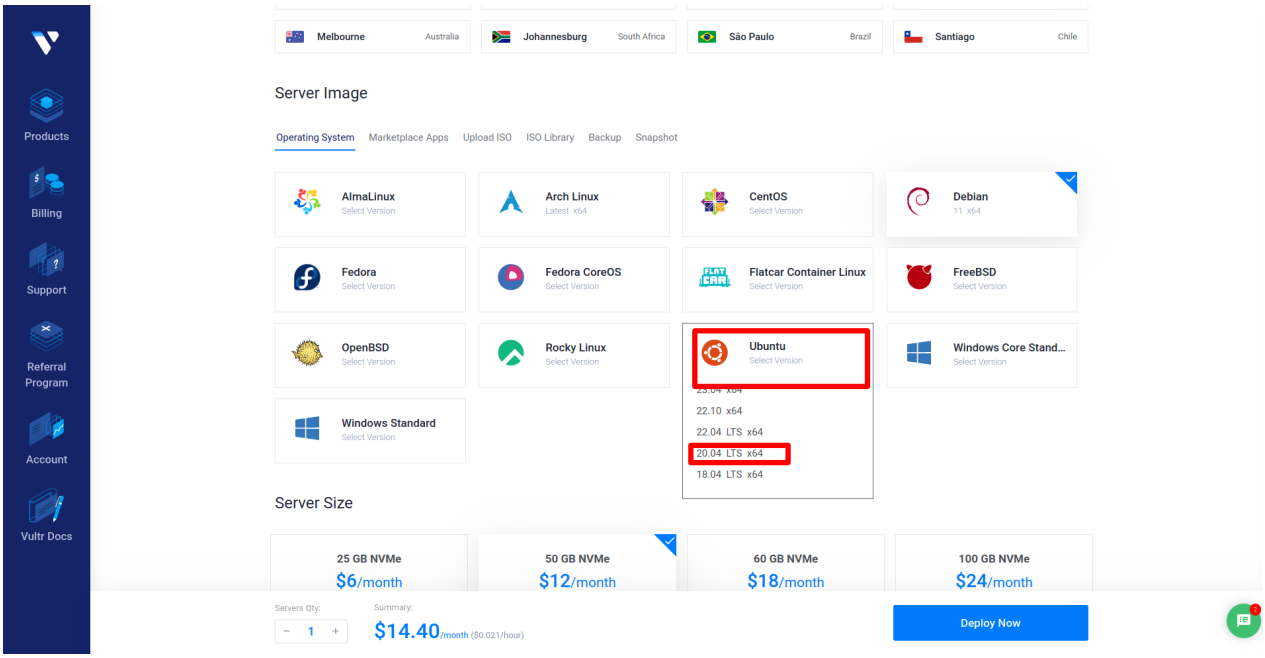
1.3.2.2 Select nearest server

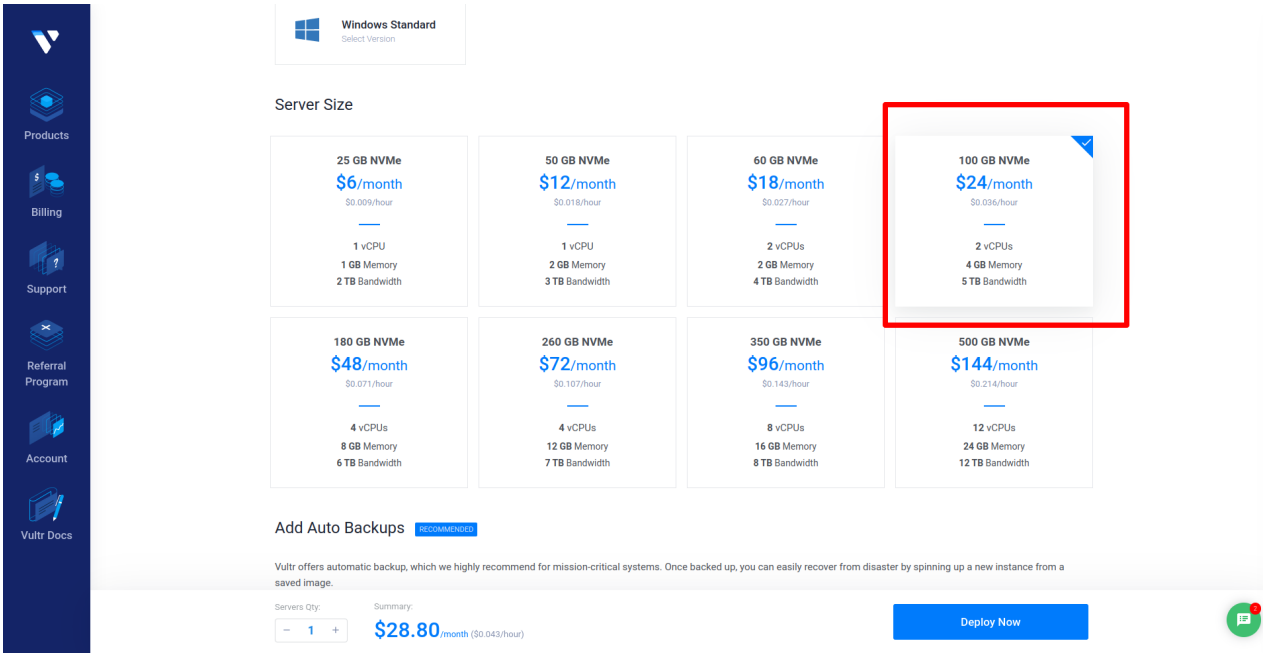
As mentioned earlier choose the server location nearest to the user's location.



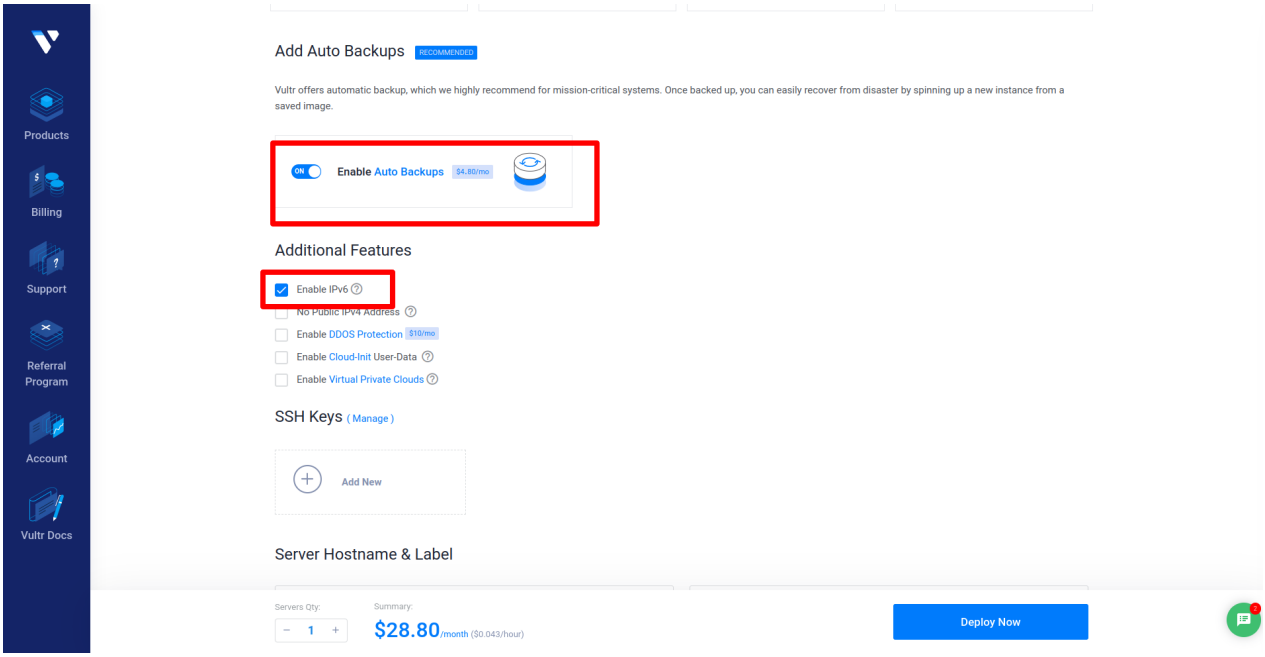
1.3.2.3 Select the server Image

Select Ubuntu (minimum version 20.04)
And select the server size highlighted in the image given below.



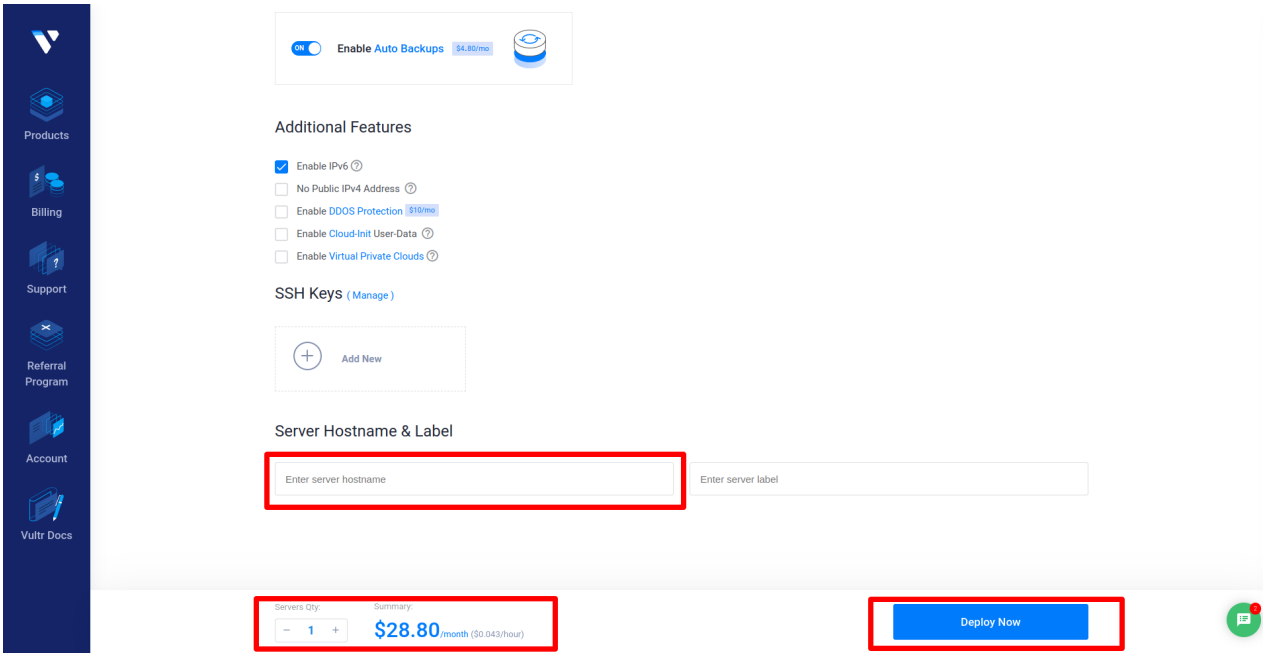


1.3.2.4 Keep these settings as default.



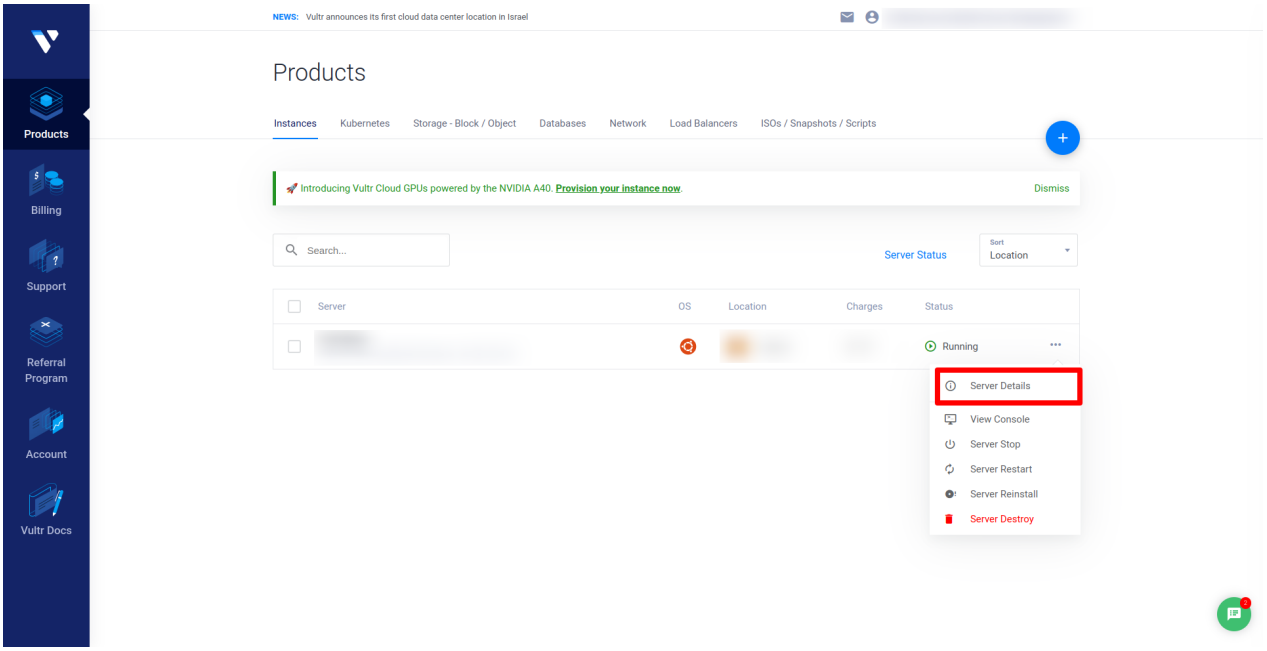
1.3.2.5 Server hostname

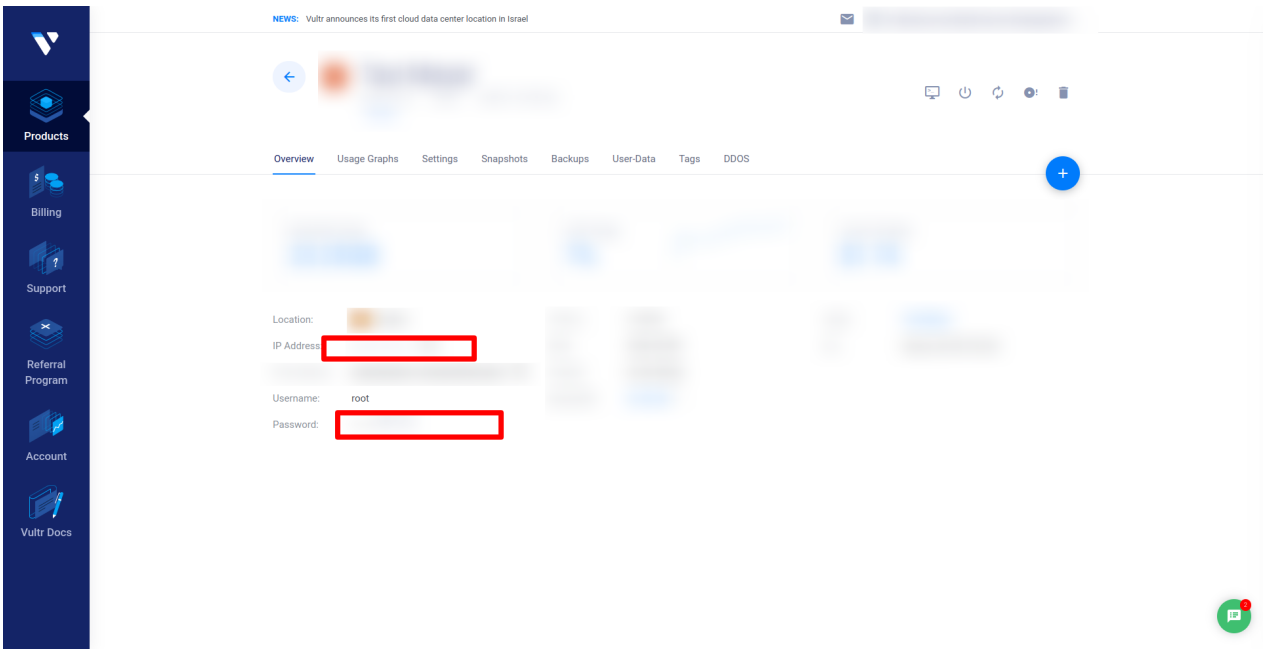
Server hostname you want as tag. In the footer you can show estimation. Everything is ok then click deploy now. Check review at last before you deploy



1.3.2.6 Server details

Click on server detail to see IP, passwords and other server details





2. Security Groups

Security groups work as firewalls for instance.

Follow accordingly:

i.e: For AWS(2.1) or For Digital Ocean(2.2) or For Vultr(2.3)

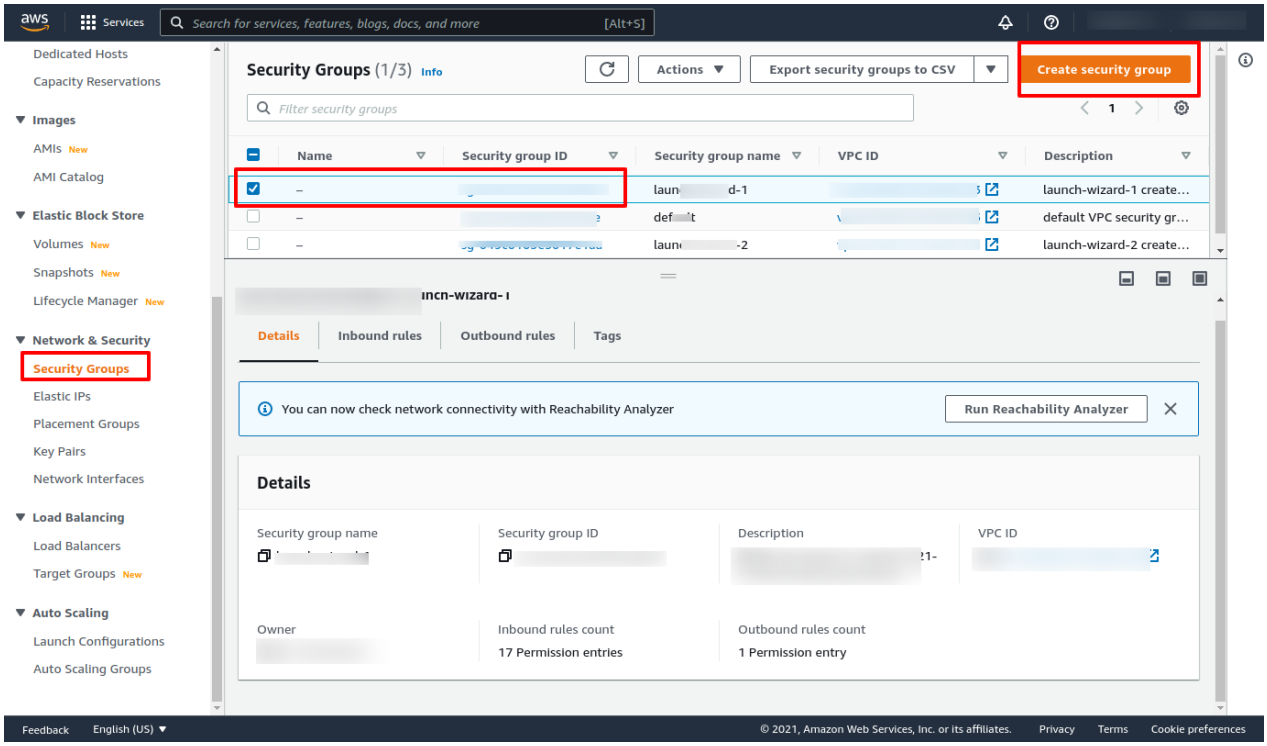
2.1 Security groups used in AWS?

Why are security groups used in AWS?
A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your instance. If you don't specify a security group, Amazon EC2 uses the default security group.

2.1.1 In the navigation pane, choose Security Groups.

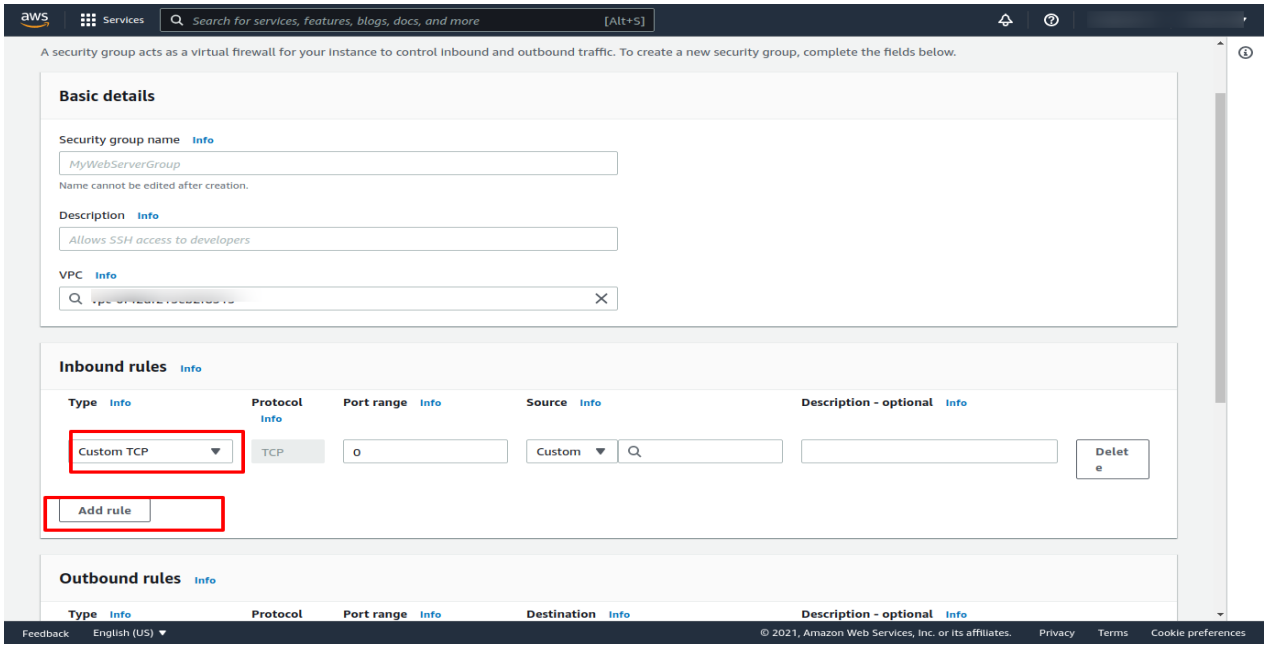
Skip this security group section if you have already done at the time of Creating Instance ec2.
Jump to section 3

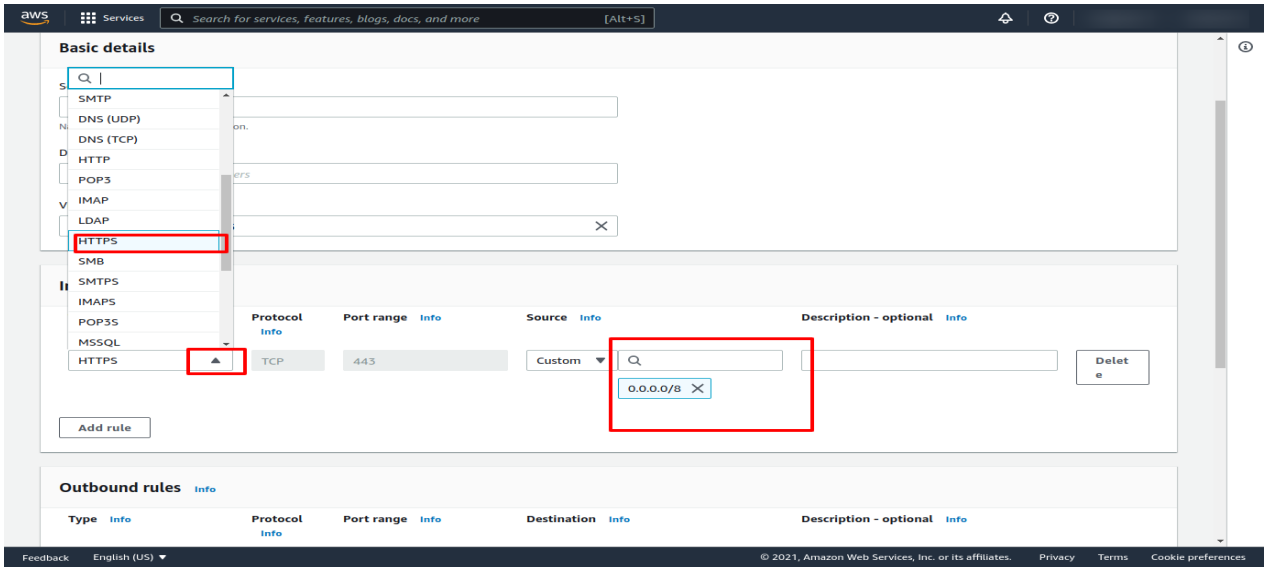
2.1.2 Choose your instance and create a security group.



2.1.3 Configure Security Inbounds

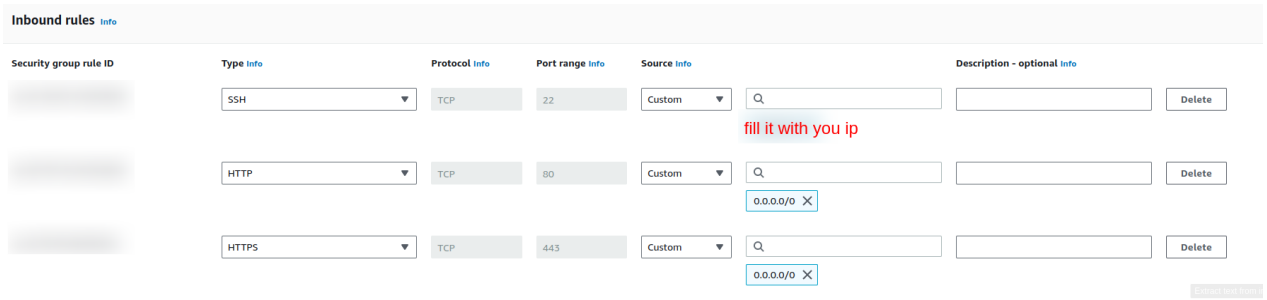
First click on the add rule and then click on the custom button and select the type HTTPS.





2.1.4 Add IPv4 addresses with HTTPS

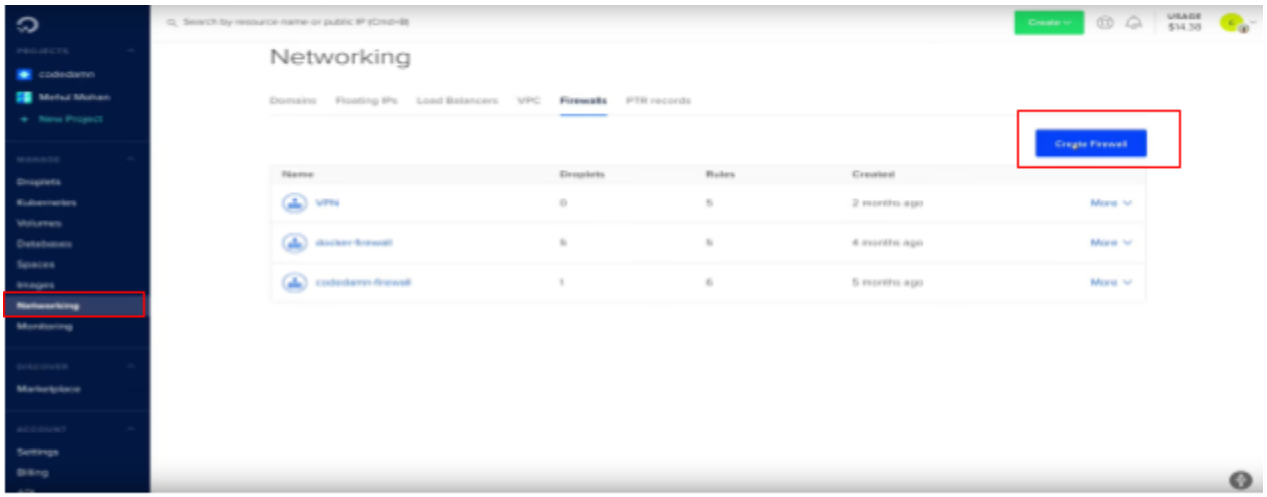
which are given below, and Save Rules.


Inbound rules Info
Security group rule ID
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
SSH
TCP
22
Custom
Delete
fill it with you ip
HTTP
TCP
80
Custom
Delete
0.0.0.0/0
HTTPS
TCP
443
Custom
Delete
0.0.0.0/0

2.2 Security groups used in Digital Ocean.

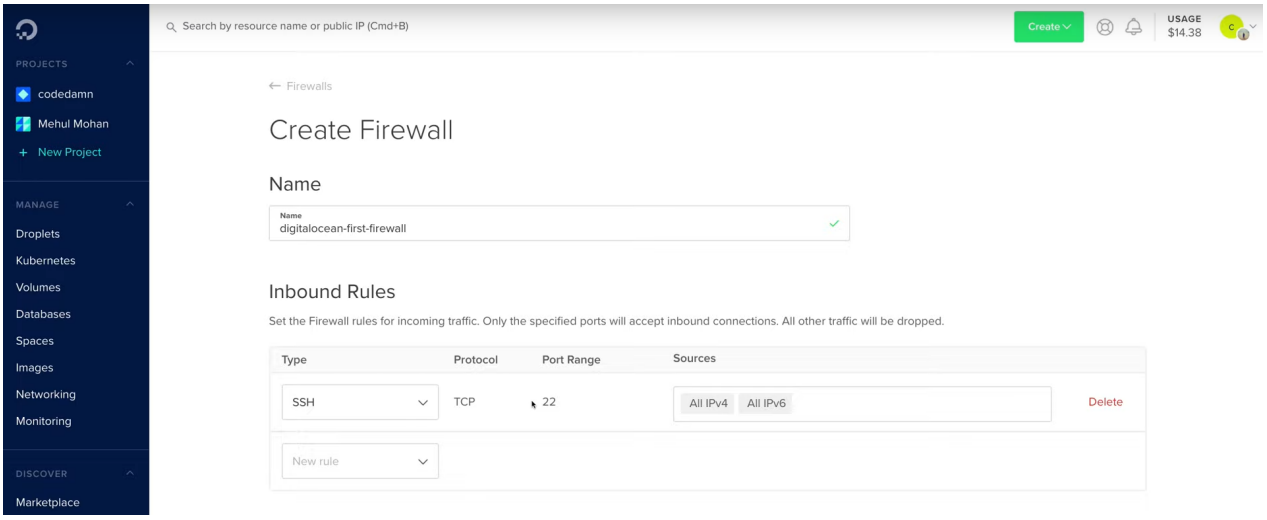
2.2.1 Create Firewall

In the navigation pane select “**Networking**” then from tabs select “**Firewalls**” and then “**Create Firewall**”.



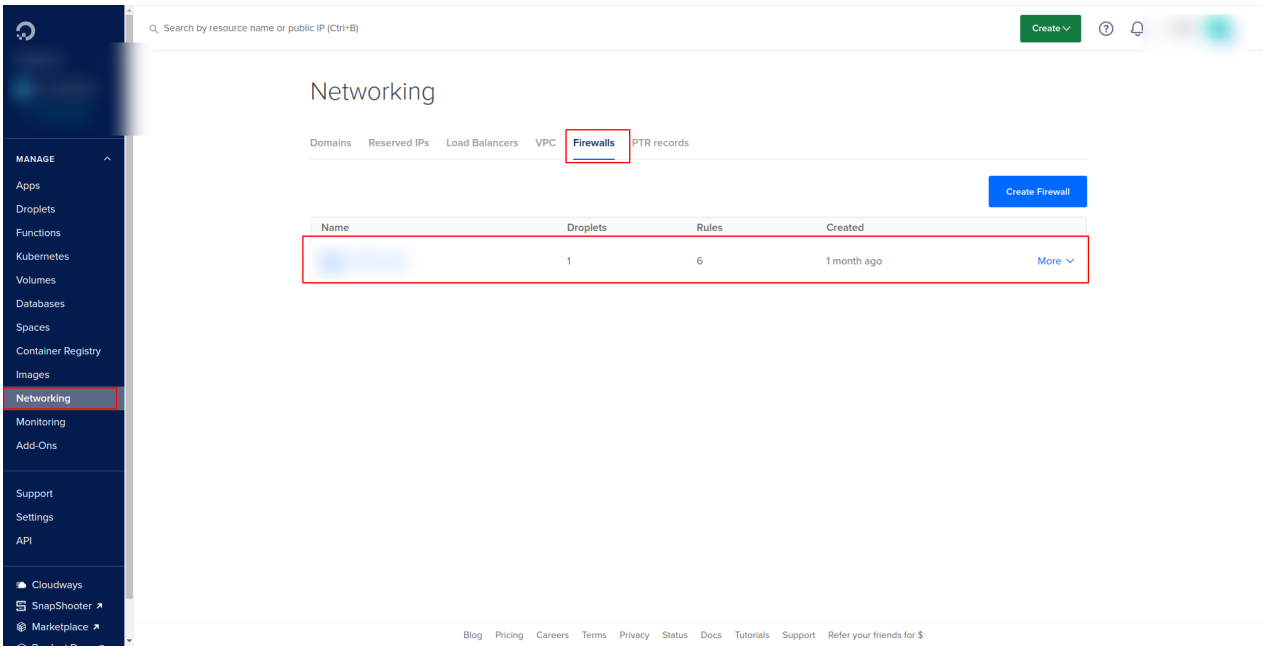
2.2.2 Adding Rules to the Firewall group

Similar to the previously mentioned groups in the AWS, use the same rules and add it here, refer (2.1.4)

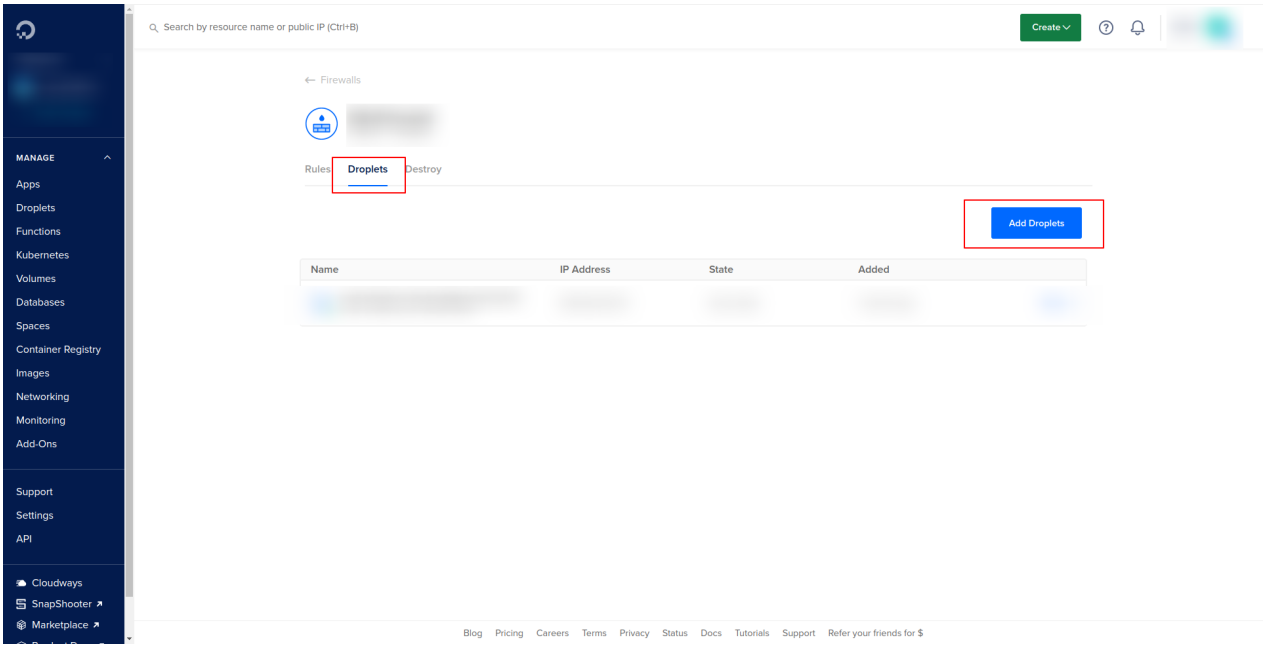


2.2.3 Connecting the Firewall Group With the Droplet

Select the firewall group you want to add to the droplet



After that, click on **Droplets** Tab, then after click on **Add Droplets** button and then select the droplet add it with the firewall group.

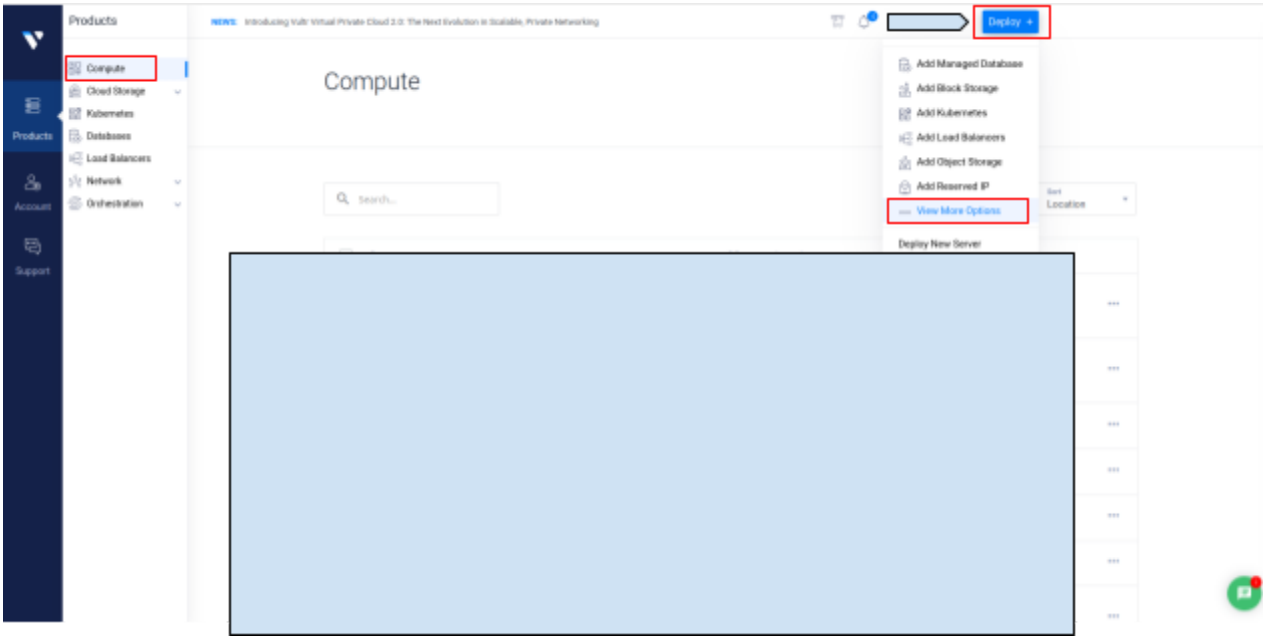


2.3 Security groups used in Vultr.

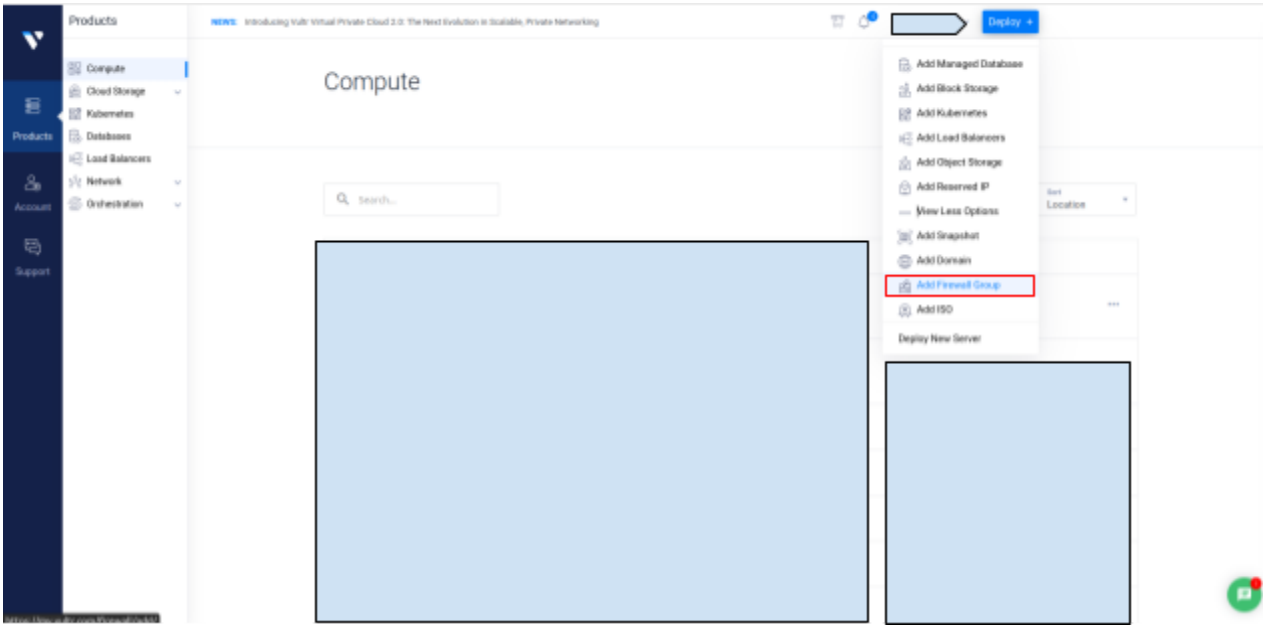
Here in Vultr there is firewall, which works same as security groups in AWS.

2.3.1 Navigate to Firewall

In the navigation pane, click on “**compute**” and hover on “**Deploy**”.
Then click on **“View More Options”**

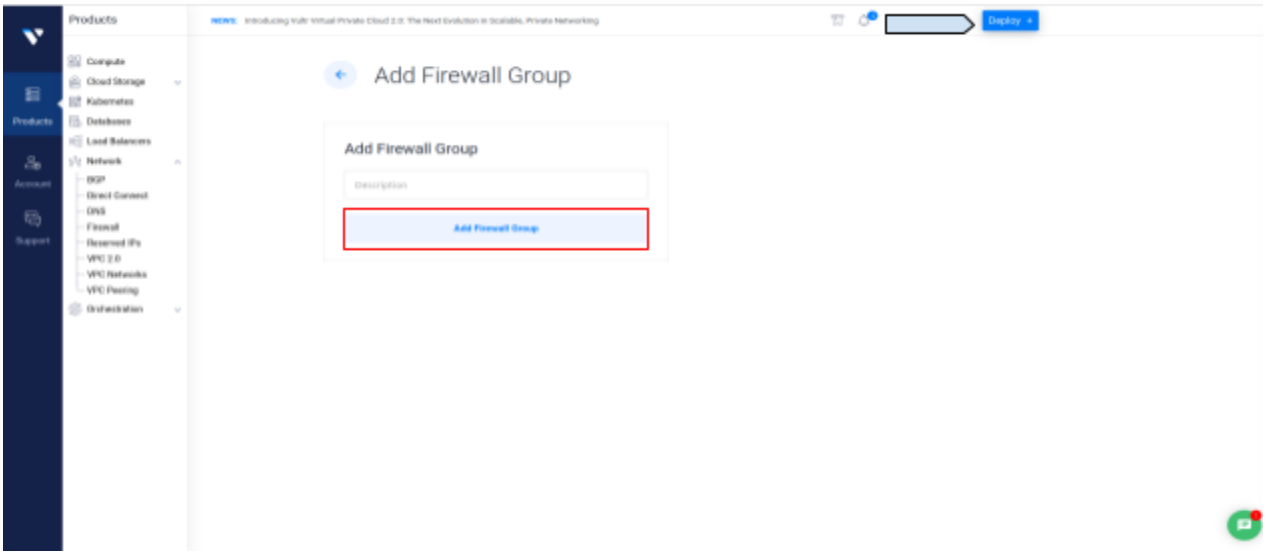


Then click on “**Add Firewall Group**”



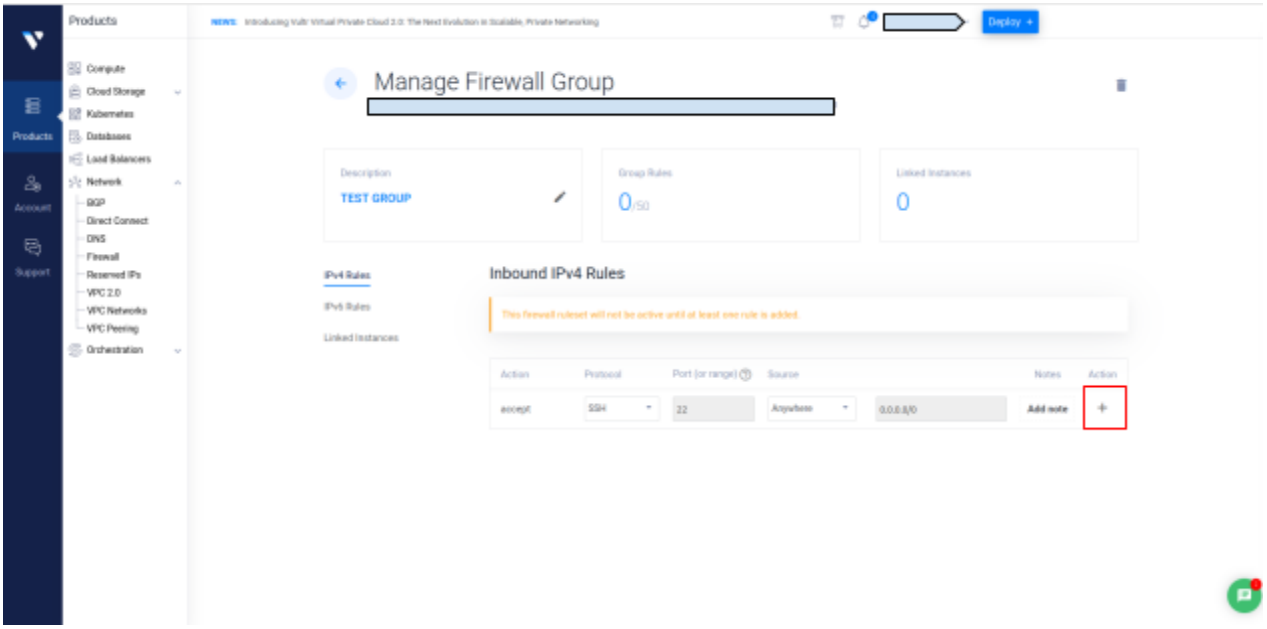
2.3.2 Add Firewall group

Add description in the field and click “**Add Firewall Group**” button.



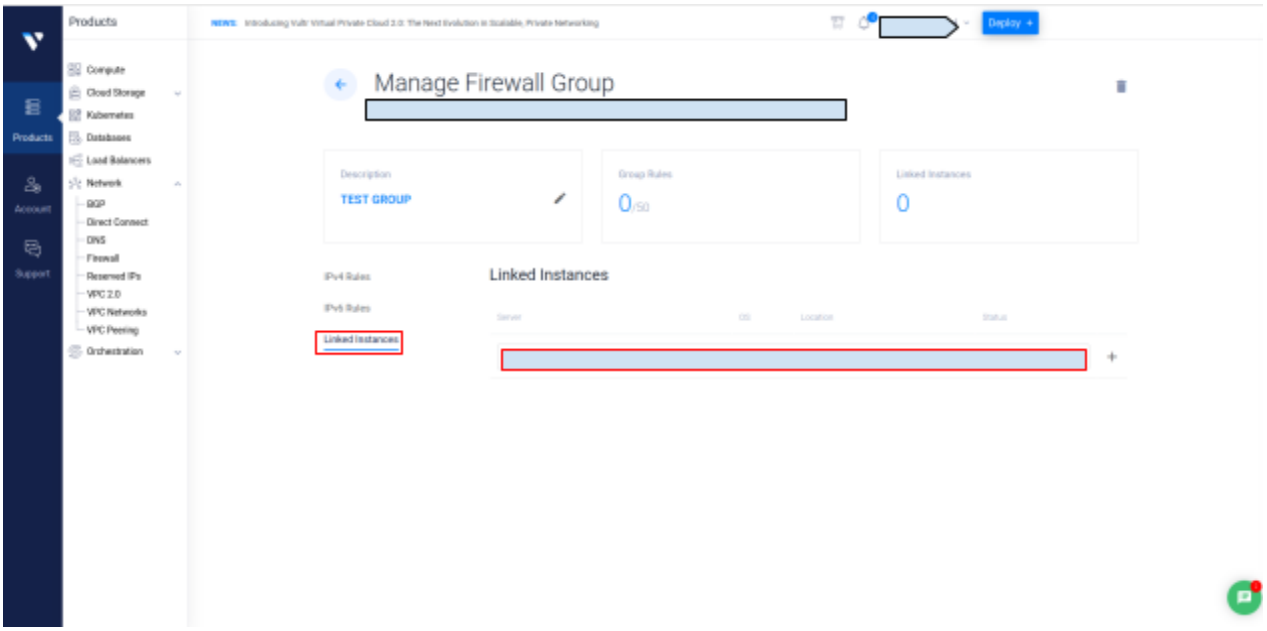
2.3.3 Add Rules in the group

Similar to the previously mentioned groups in the AWS, use the same rules and add it here, refer (2.1.4)



2.3.4 Connect Firewall group with the instance

After the rules are properly added to the group, now we will connect the instance with the group. Click on the "**Linked Instances**" and then select the instance.



3. Elastic IP / Reserved IP

What is Elastic Ip and why is it useful?

The Elastic IP address is a public **static IPv4 address** that is reachable from the Internet. Basically, Elastic IP addresses are used by AWS to manage its dynamic cloud computing services. Within the AWS infrastructure, customers have virtual private clouds (VPC, within the VPCs, users have instances. So when you launch an EC2 instance, you receive a Public IP address by which that instance is reachable from the internet. Once you stop that instance and restart the instance you get a new Public IP for the same instance. So it's basically a problem to connect your instance from the internet for not having a static IP. To overcome this problem, we attach an Elastic IP to an Instance that doesn't change after you stop/start the instance.

Follow accordingly:

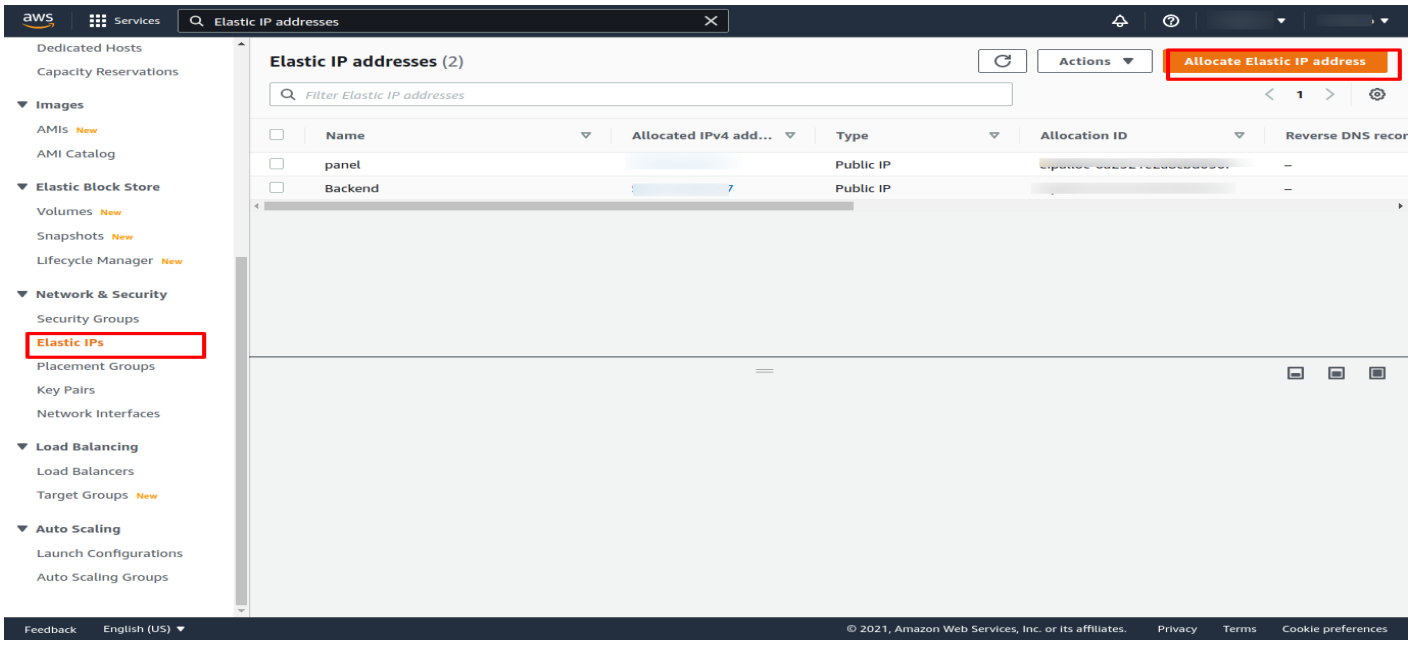
i.e: For AWS(3.1) or For Digital Ocean(3.2) or For Vultr(3.3)

3.1 Elastic IP using AWS.

Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/

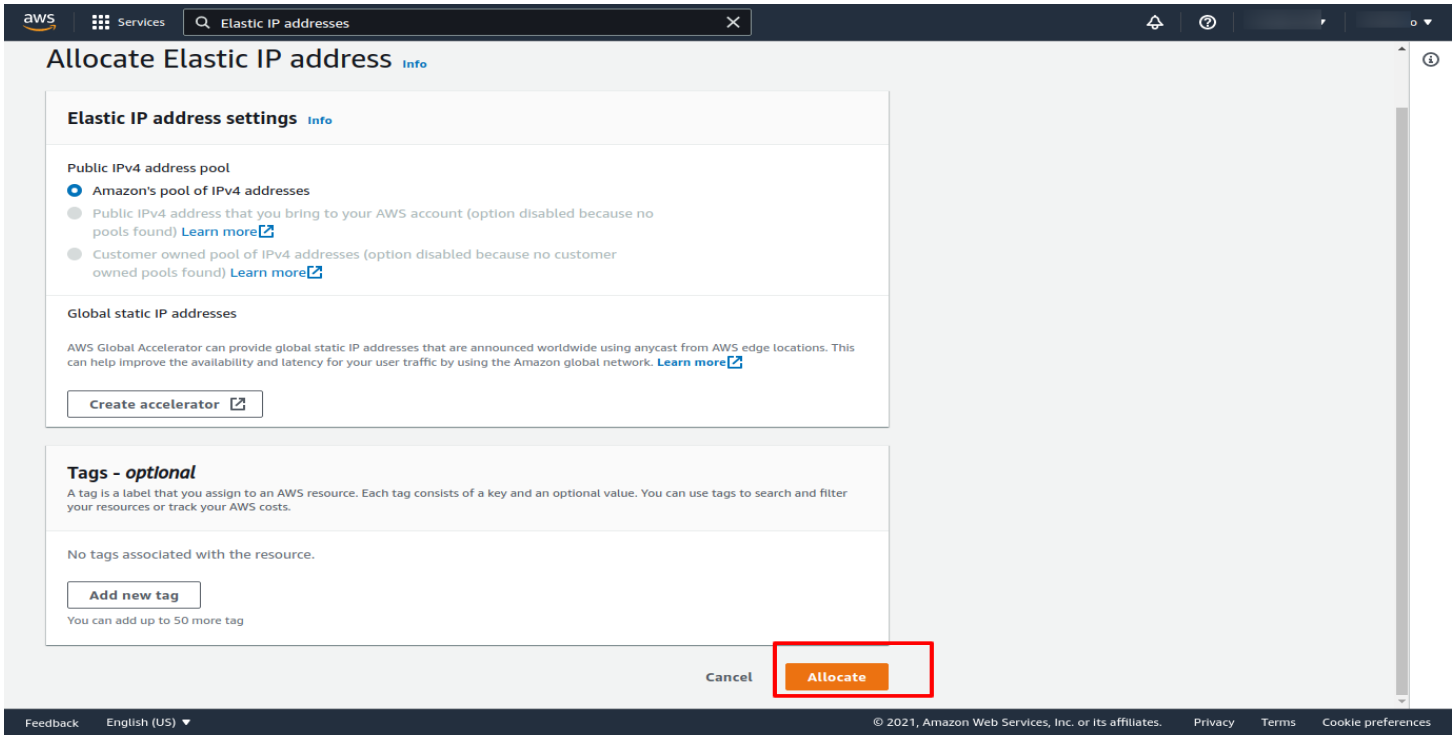
In the navigation pane, choose Network & Security, Elastic IPs.

3.1.1 Choose Allocate Elastic IP address.



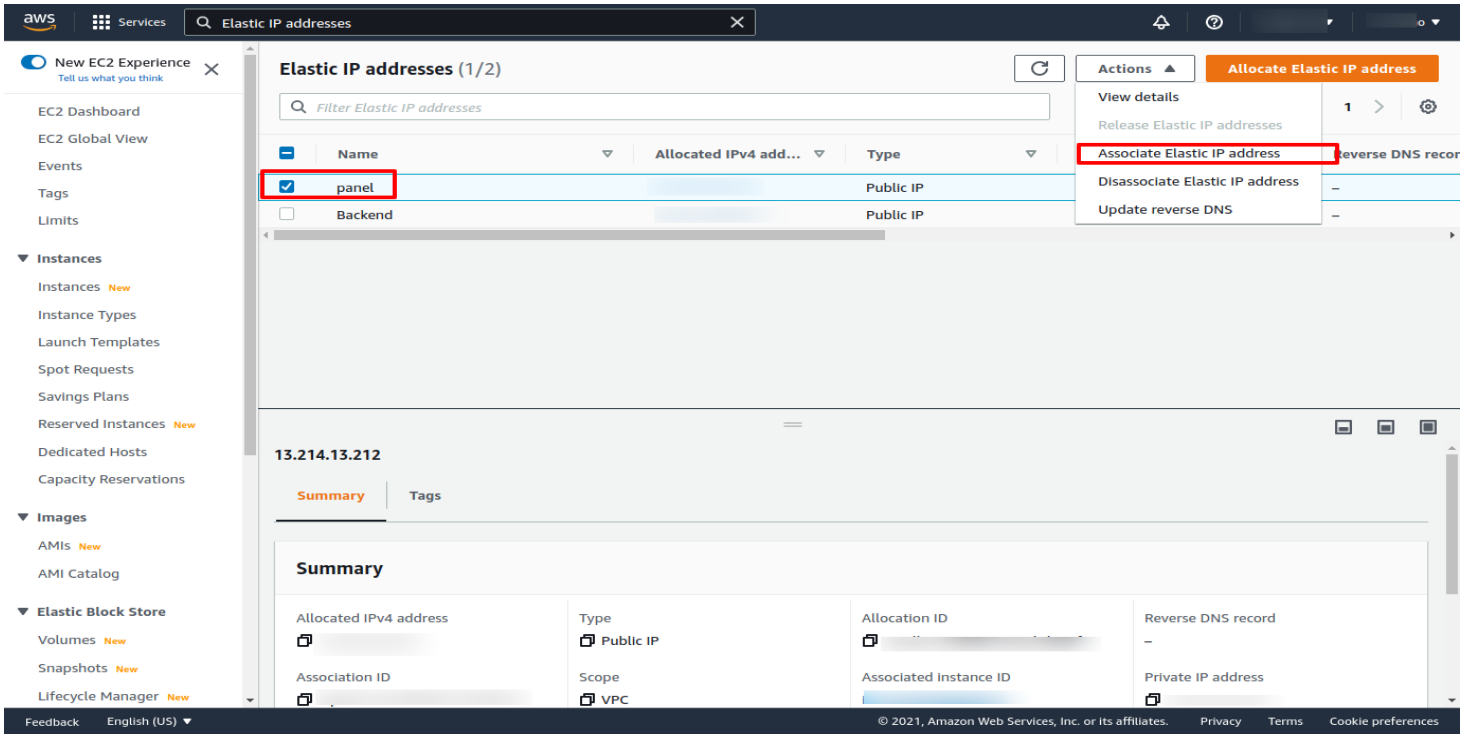
3.1.2 Navigate to the page

For the Public IPv4 address pool, choose one of the following and then allocate



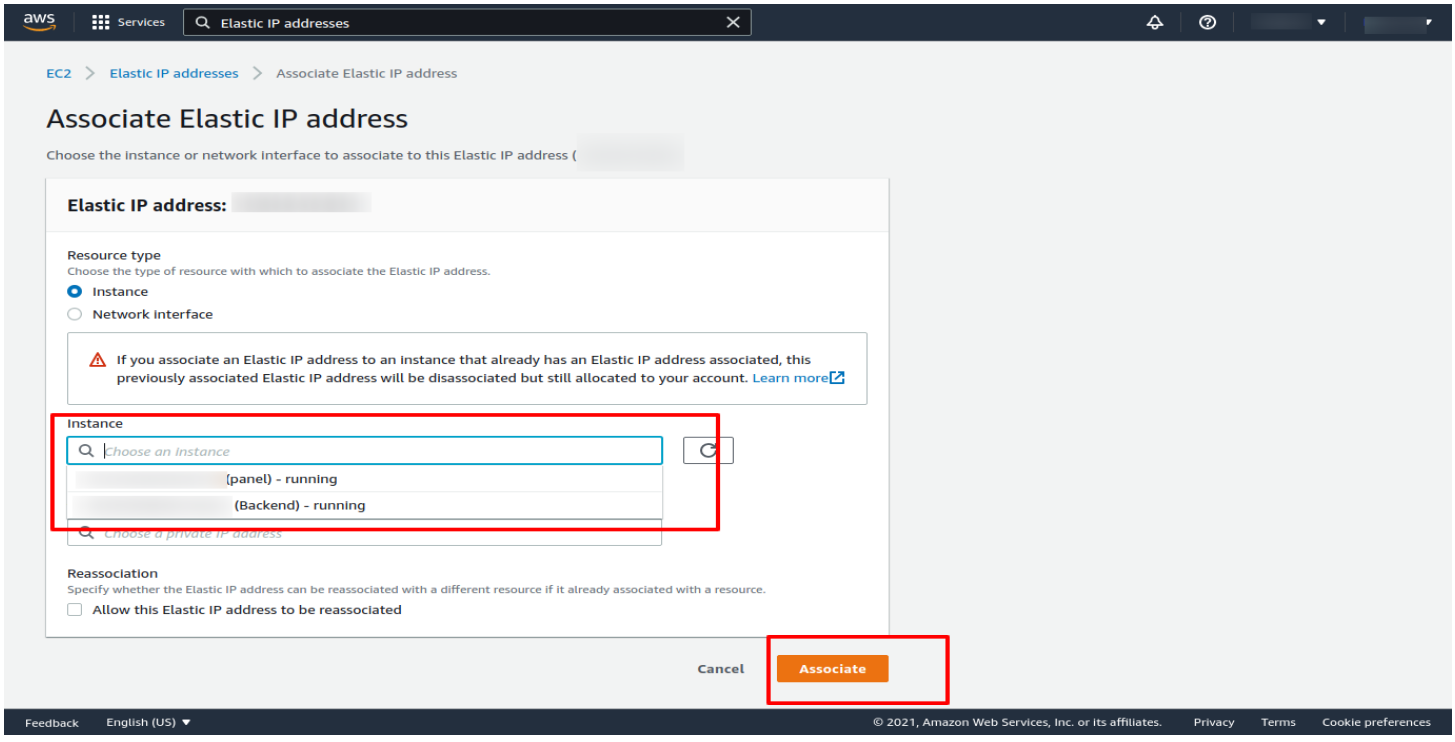
3.1.3 Select IP

Now Select IP and click on allocate an elastic IP address.



3.1.4 Select your project

Select your project Instance and allocate.



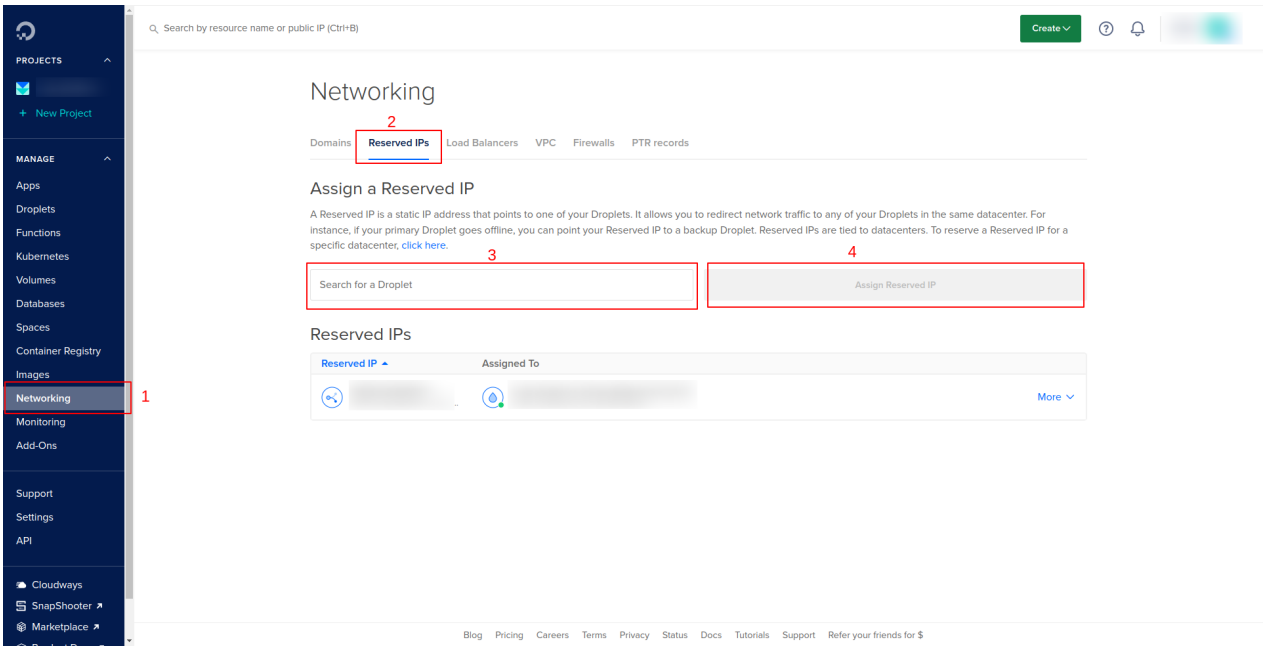
3.2 Elastic IP using DigitalOcean

3.2.1 Navigate to Reserved IPs

Follow Steps,

1. Networking
2. Reserved IPs
3. Search and Select the droplet
4. Assign Reserved IP

The reserved IP is now assigned with Droplet.

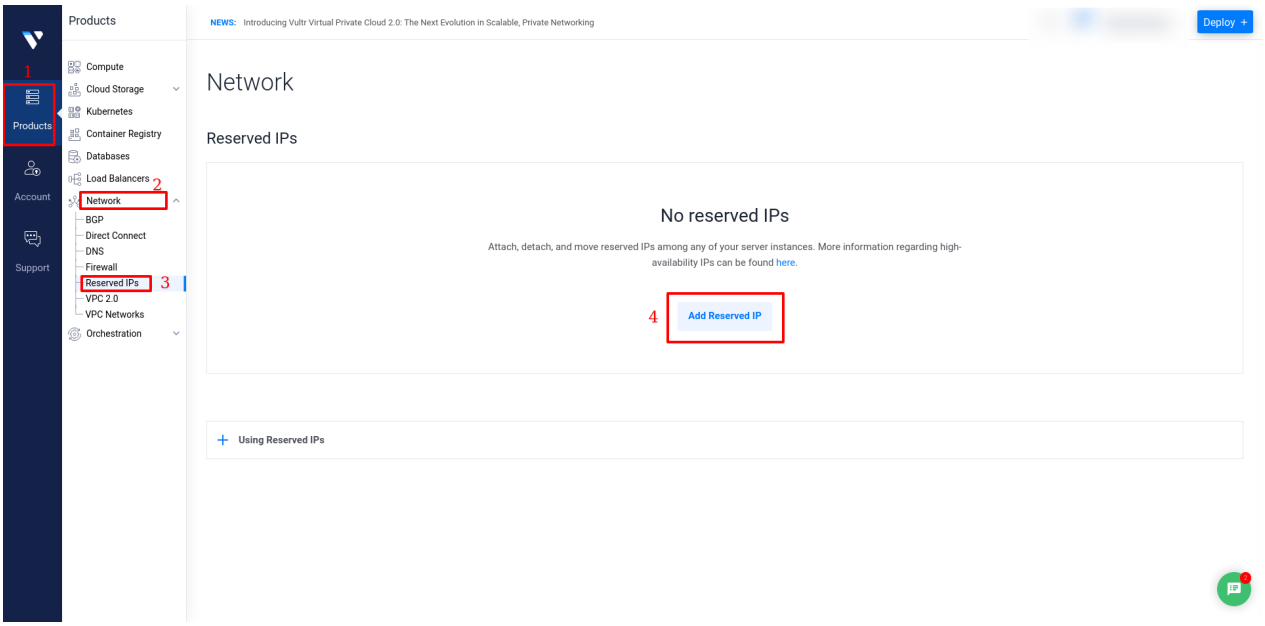


3.3 Elastic IP using Vultr

3.3.1 Navigate to Reserved IPs

Follow Steps,

1. Products
2. Network
3. Reserved IPs
4. Add Reserved IP



5. Please refer here for reference.

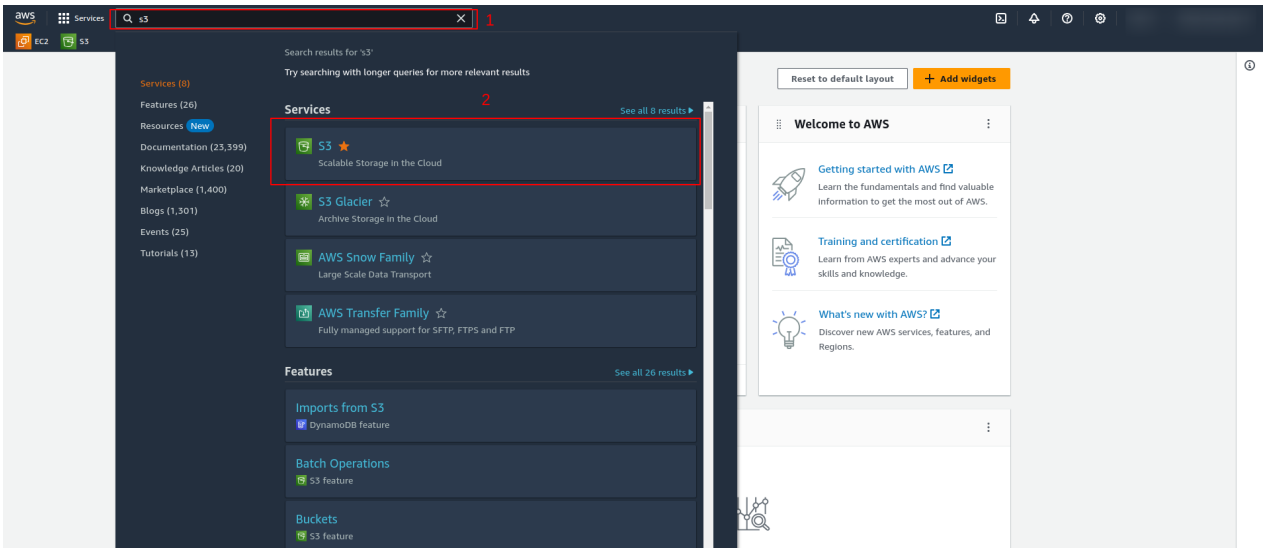
4. Create Storage(Optional).

If we are using the bucket facility, we need to upload all the static images from the project to the bucket manually.

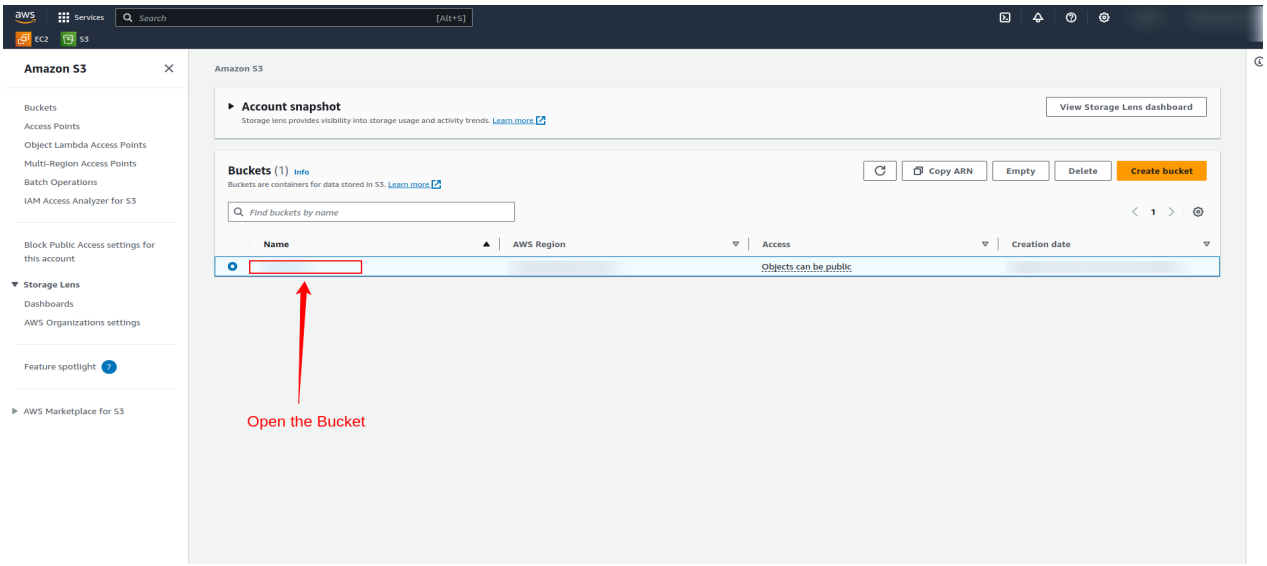
Note: The process of drag and drop all the files needs to be completed after the creation of the bucket and after the completion of the installation process on the server.

(i) For AWS S3

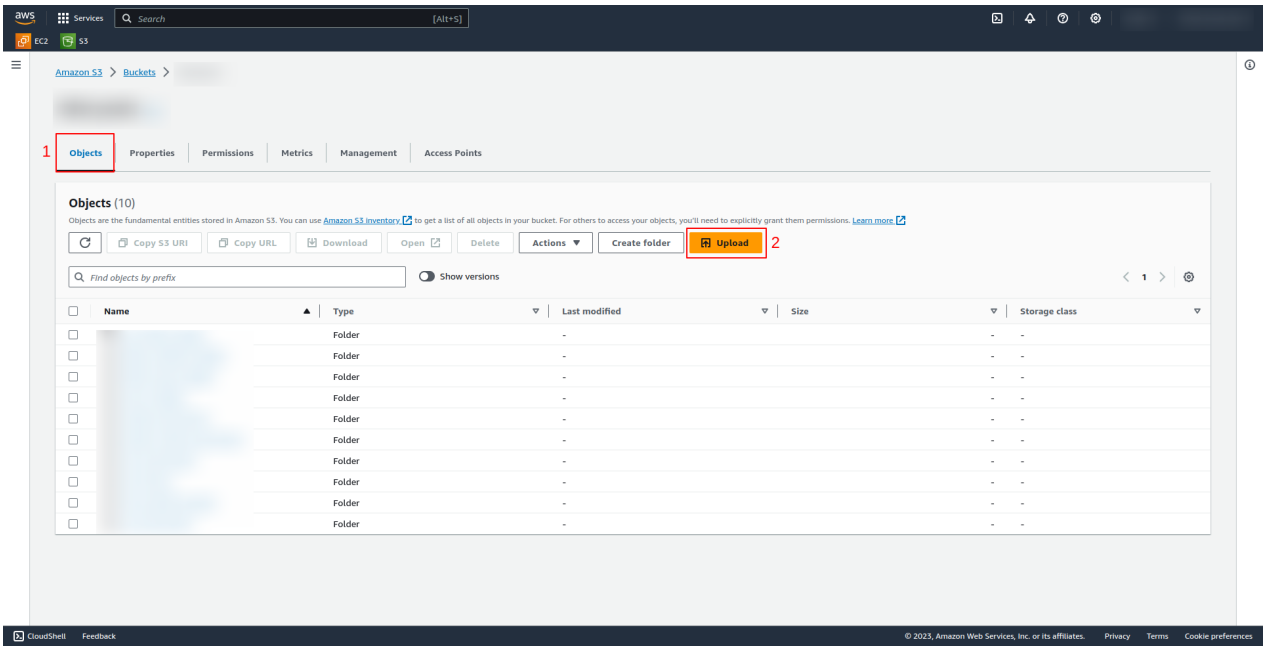
- Navigate to the bucket.
    1. Search for **S3**
    2. Select S3



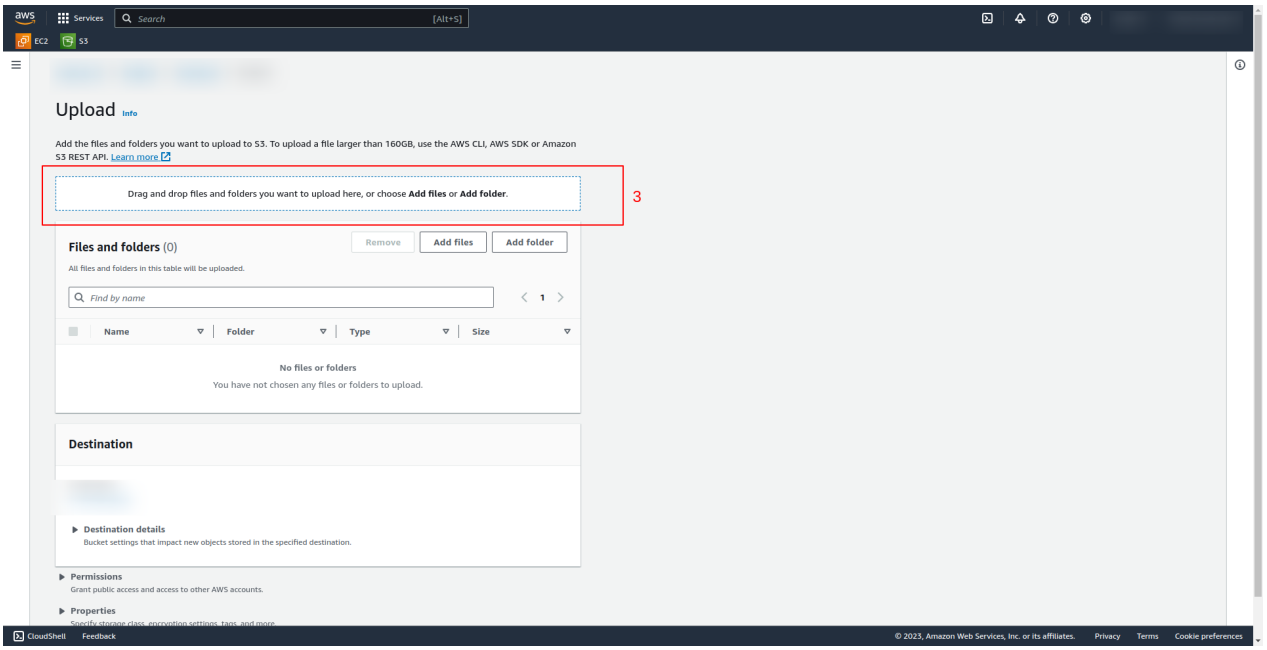
- Open the Bucket



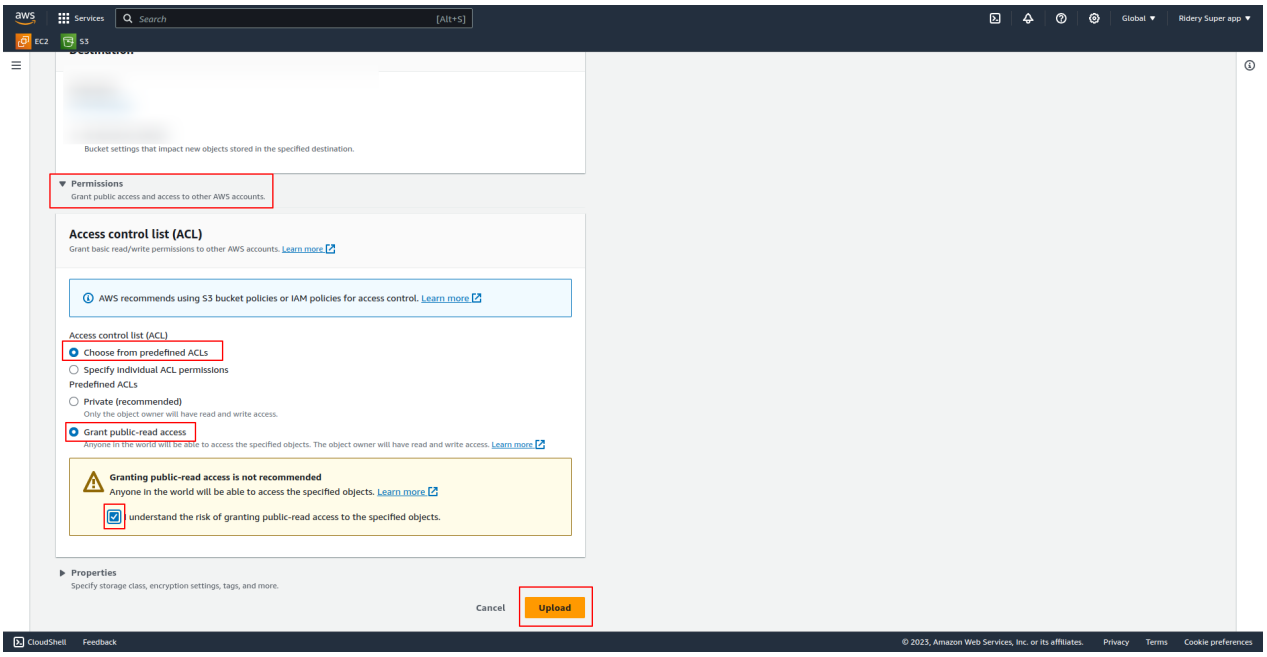
- Upload all the files in the **data** folder manually.
    - Navigate to the data folder
        - **backend/server/data**
    - Select all the files and now **drag** and **drop** it in the bucket.
        1. Objects
        2. Upload



3. Drop all the files here



4. Change the permission and click upload and that's it.



(ii) For Digital-Ocean Spaces

Navigate to the Spaces

And refer here.

And also change the permission of the files after uploading all the files on the spaces.

(4.1)For AWS S3 or (4.2)For Digital-Ocean Spaces

4.1 Using AWS S3 Bucket

4.1.1 What is S3 Bucket and why do we use it?

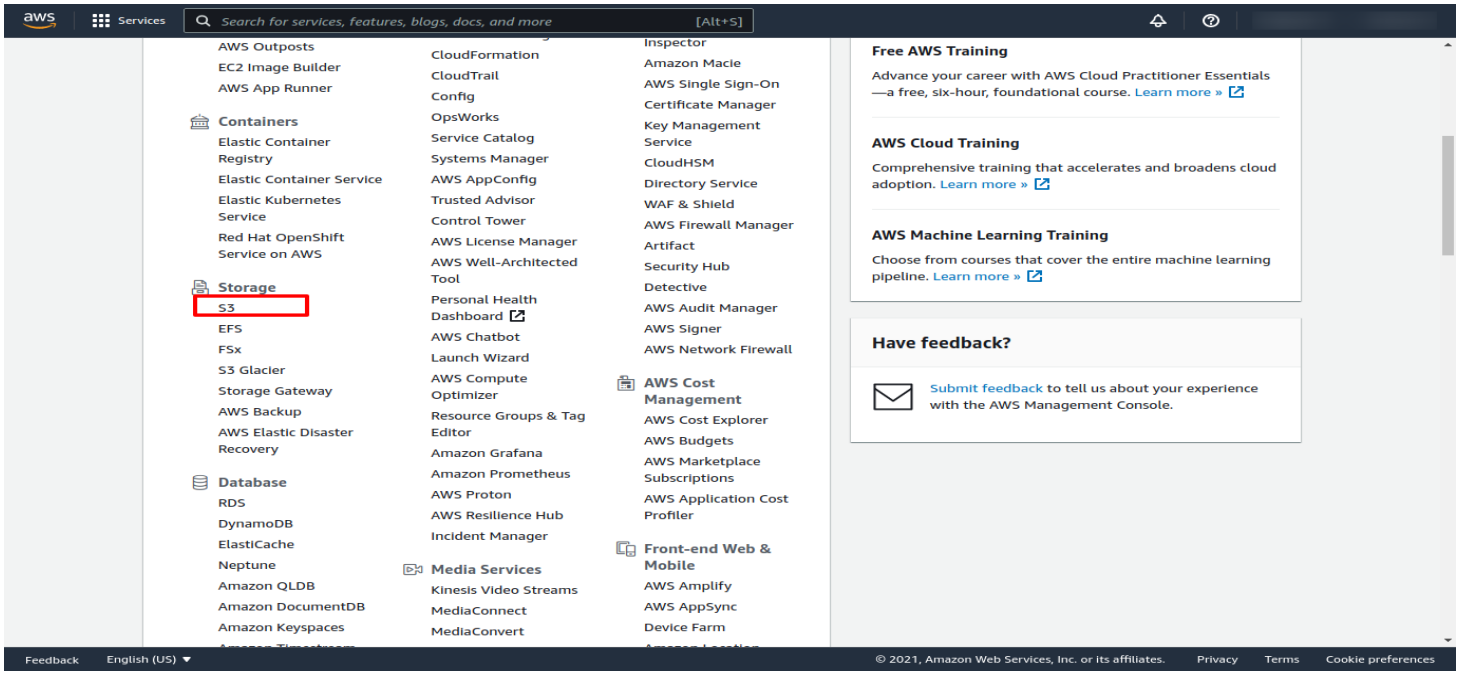
An Amazon S3 bucket is a public cloud storage resource available in Amazon Web Services' (AWS) Simple Storage Service (S3), an object storage offering. Amazon S3 buckets, which are similar to file folders, store objects, which consist of data and its descriptive metadata.

Amazon S3 is a program that's built to store, protect, and retrieve data from "buckets" at anytime from anywhere on any device. ... Use cases include websites, mobile apps, archiving, data backups and restorations, IoT devices, enterprise application storage, and

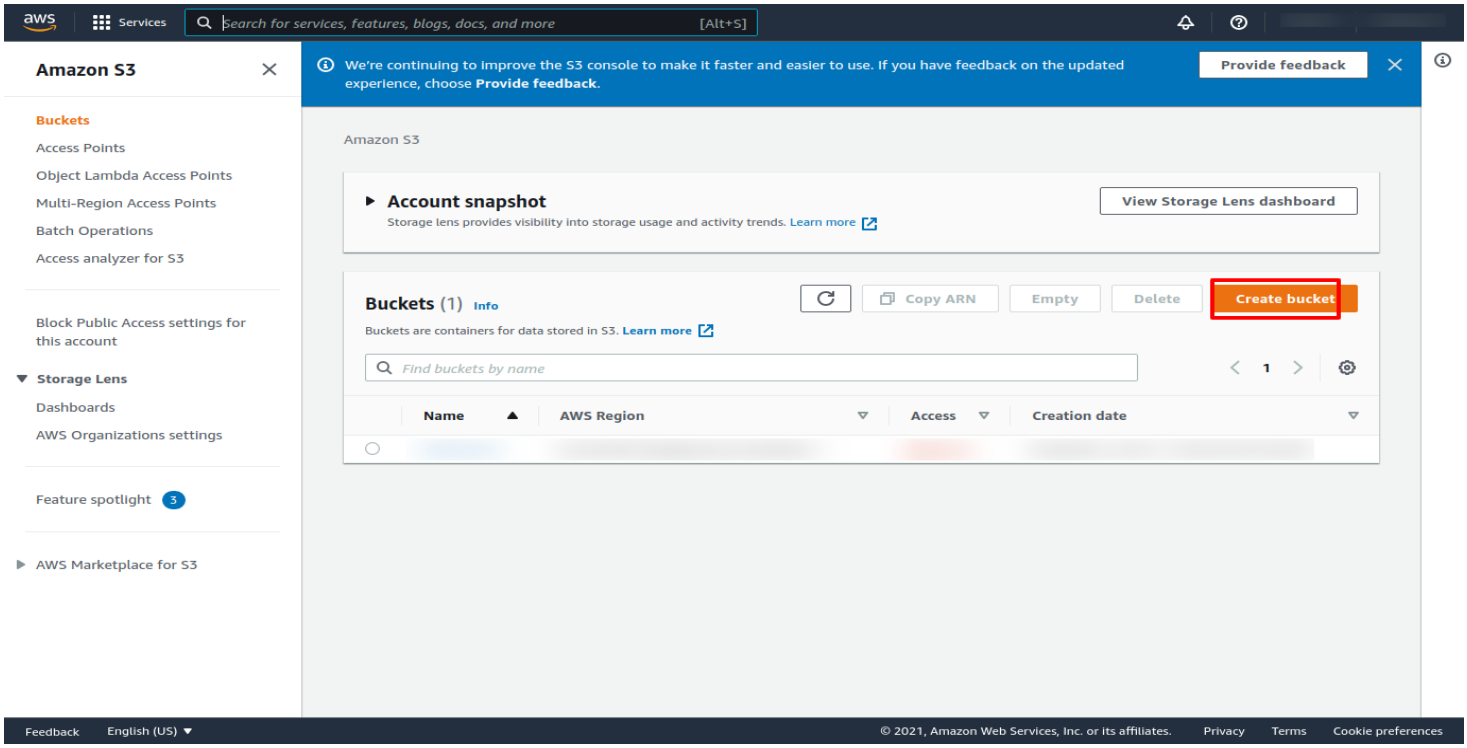
providing the underlying storage layer for your data lake.

In short, using the AWS S3 bucket it can help us to keep our server clean or we can say that the space will be occupied less, as all the files will be stored on S3 bucket.

4.1.2 Click on s3 from all services

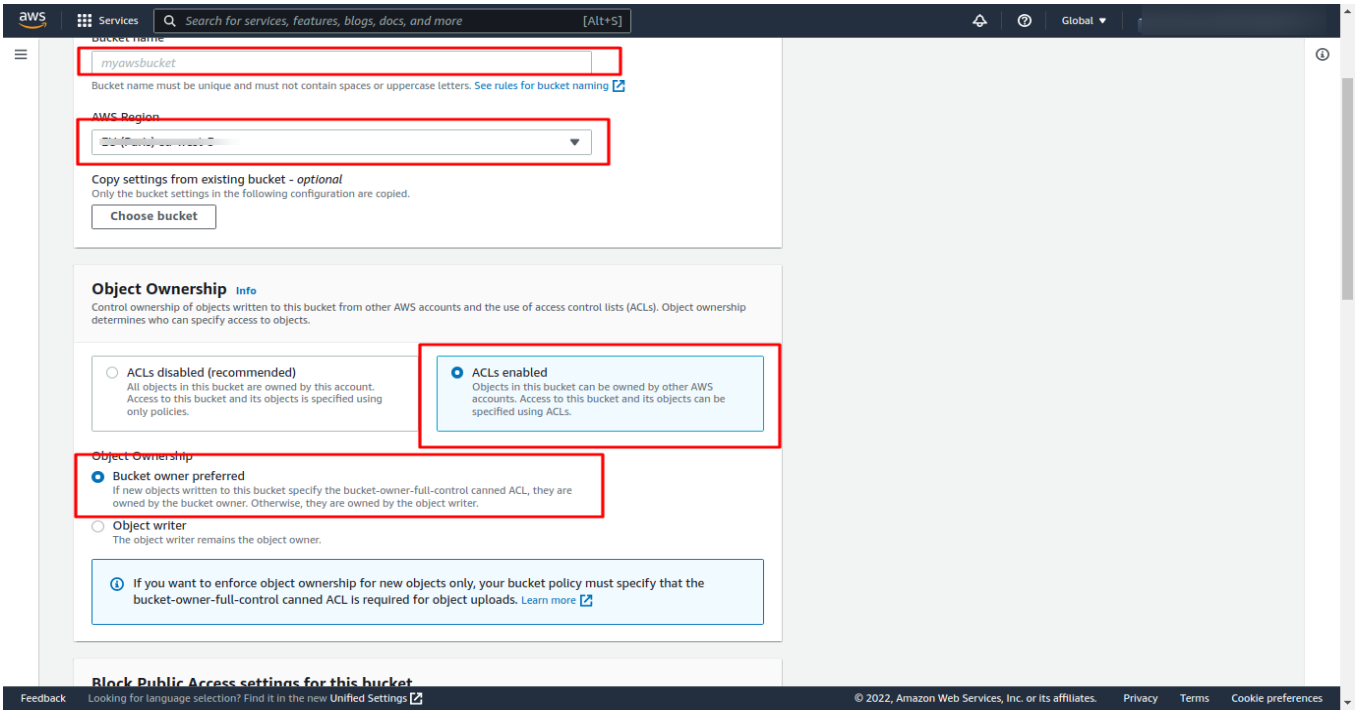


4.1.3 Click on Create bucket



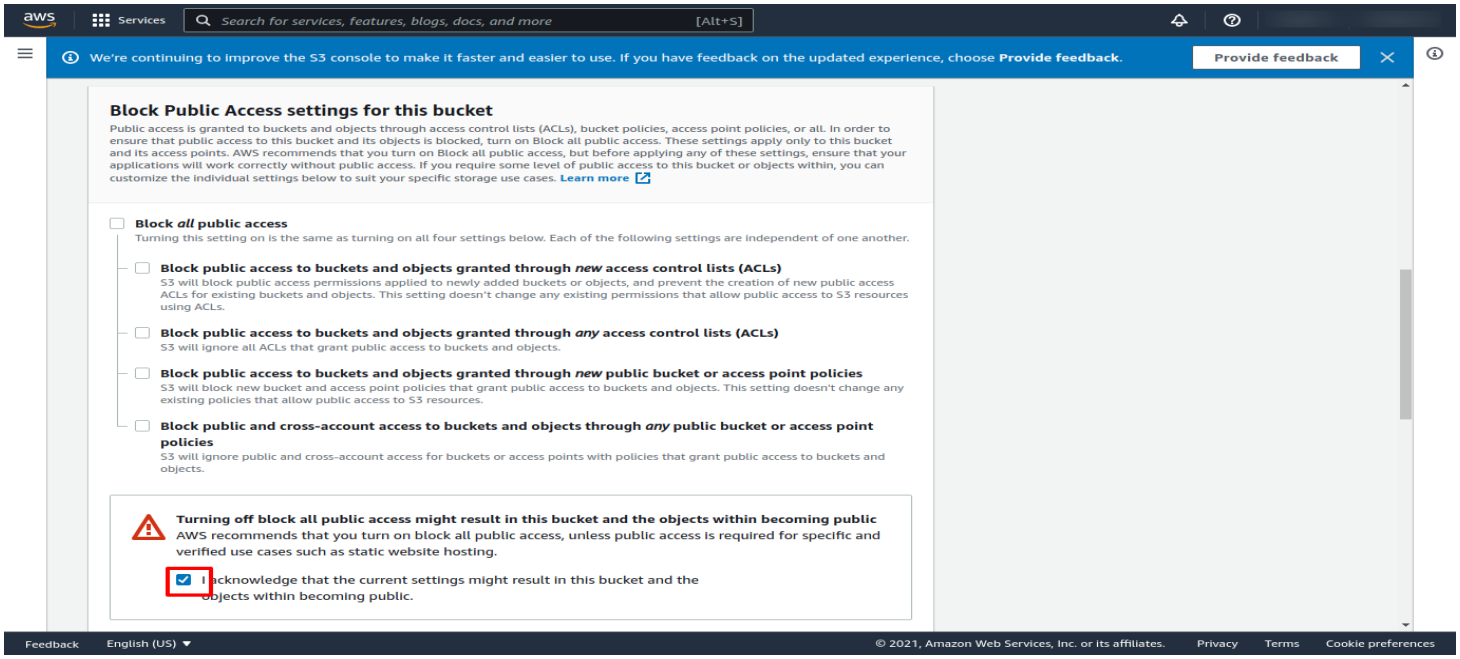
4.1.4 Add bucket name and select region according to country.

- In Bucket name, enter a unique name for your bucket.
- Choose a Region close to you to minimize latency and costs and address regulatory requirements. Objects stored in a Region never leave that Region unless you explicitly transfer them to another Region.

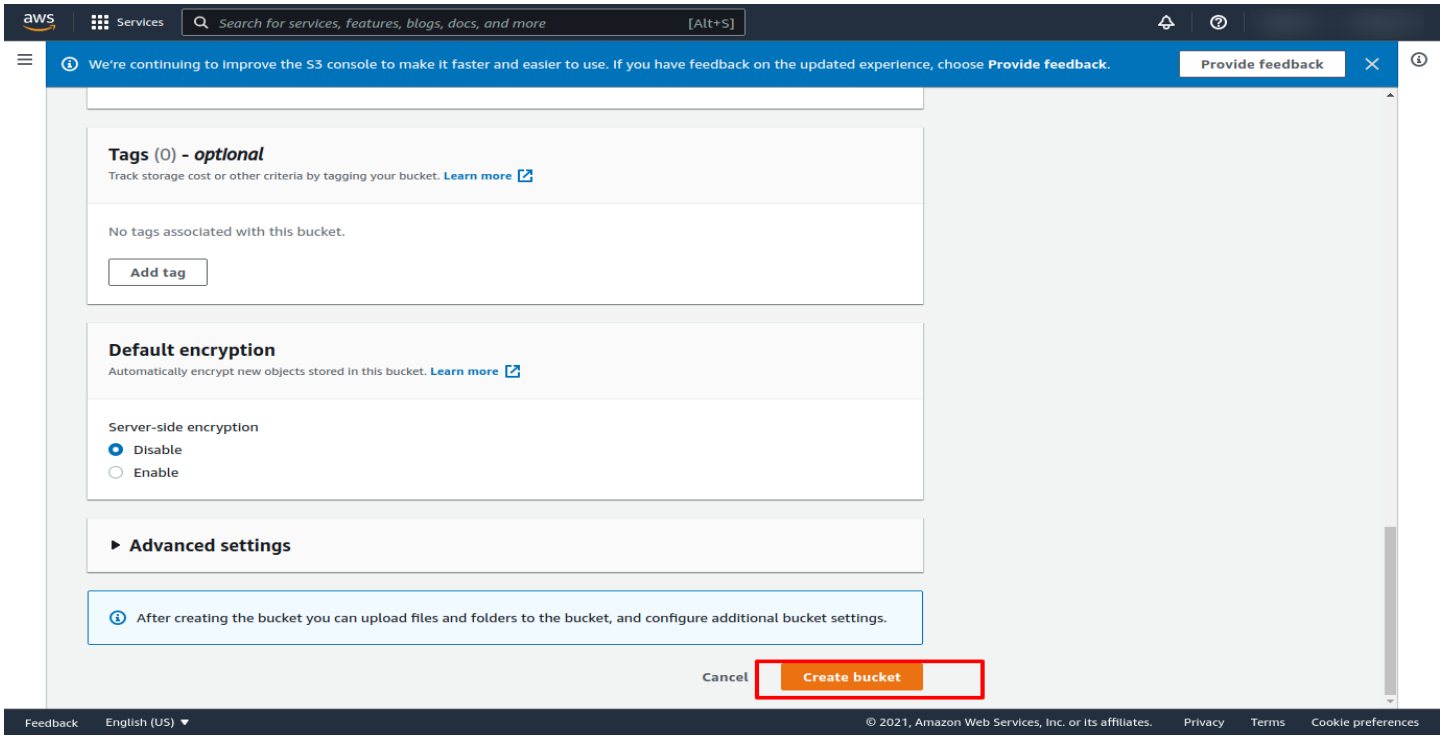


4.1.5 Unblock public access.

- We need to disable the block off public access
- Accept the Acknowledgement because our bucket needs the public access.



4.1.6 Now Click on Create Bucket.



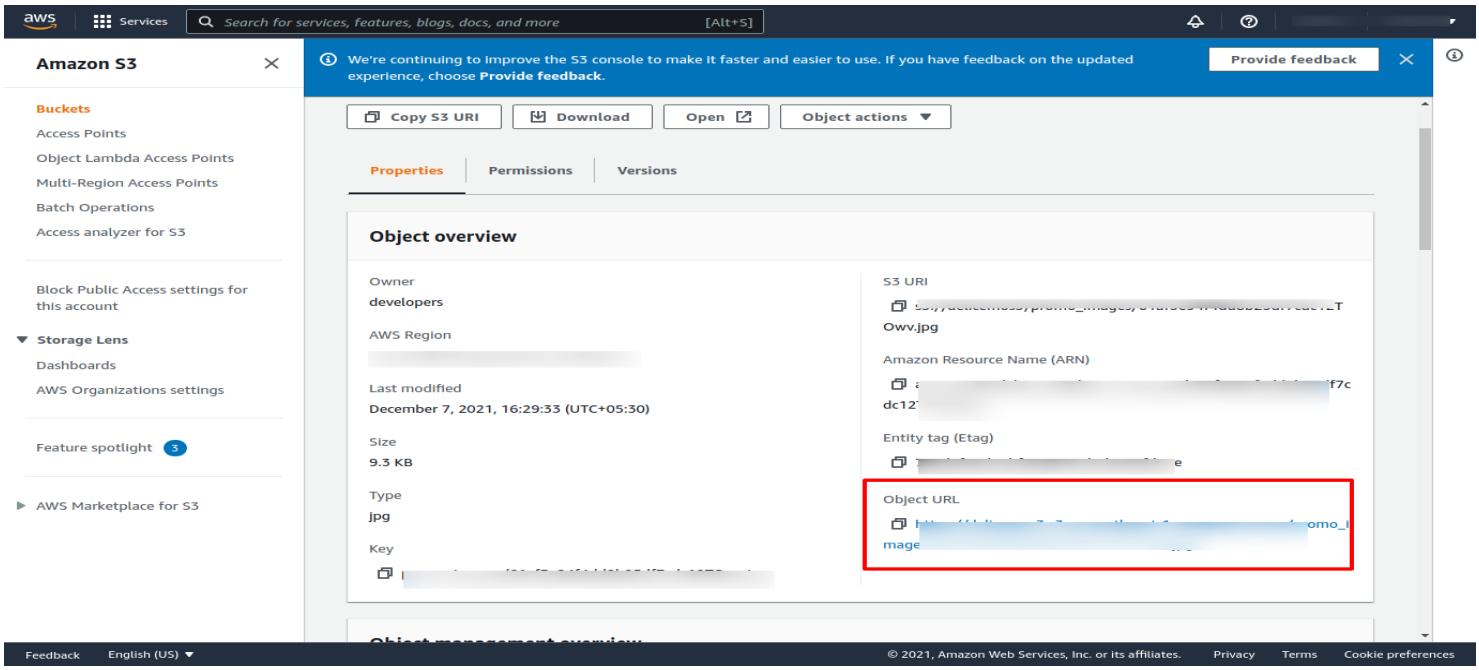
4.1.7 How to get an S3 bucket image URL

1) Upload image in s3 bucket and select that image and open properties.
2) In object overview copy that Object URI.
3) The **Image Url** will look like - “http://your-bucket.s3-website-us-east-1.amazonaws.com”

(a) Update the **image url** value in the following listed files after the installation process has completed using bash file.

a. admin-panel/src/environments/environment.prod.ts
b. admin-panel/src/environments/environment.ts
c. corporate-panel/src/environments/environment.prod.ts
d. corporate-panel/src/environments/environment.ts
e. dispatcher_panel/src/environments/environment.prod.ts
f. dispatcher_panel/src/environments/environment.ts
g. driver-panel/src/environments/environment.prod.ts
h. driver-panel/src/environments/environment.ts

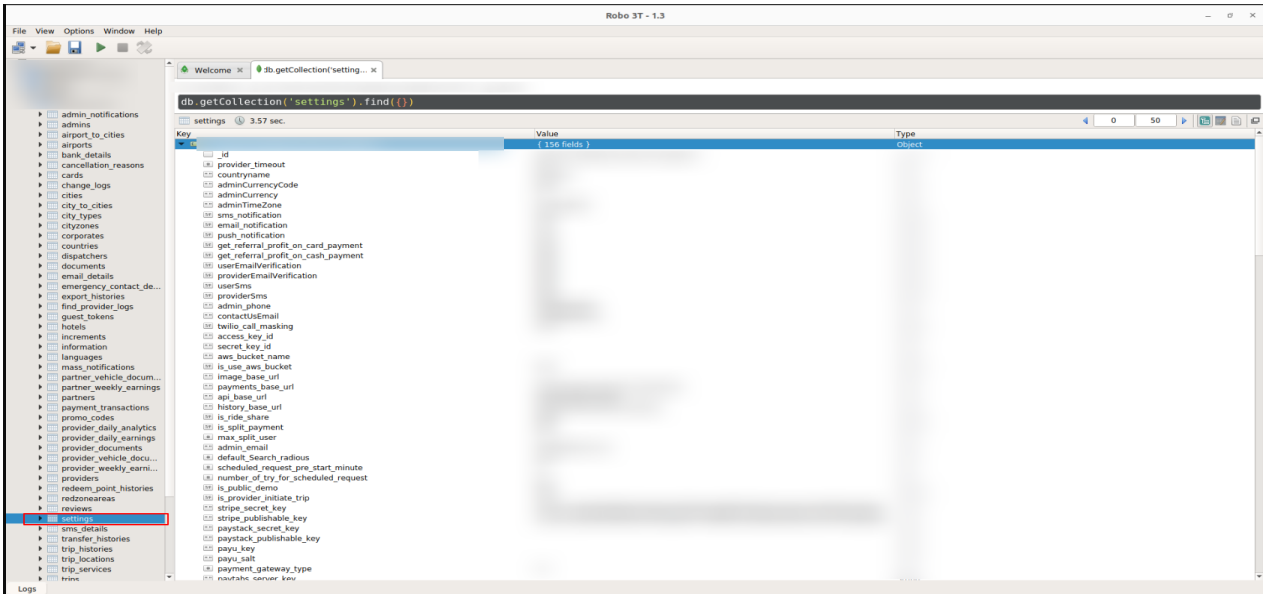
i. hotel_panel/src/environments/environment.prod.ts
j. hotel_panel/src/environments/environment.ts
k. partner-panel/src/environments/environment.prod.ts
l. partner-panel/src/environments/environment.ts
m. user-panel/src/environments/environment.prod.ts
n. user-panel/src/environments/environment.ts
o. hub-panel/src/environments/environment.prod.ts
p. hub-panel/src/environments/environment.ts

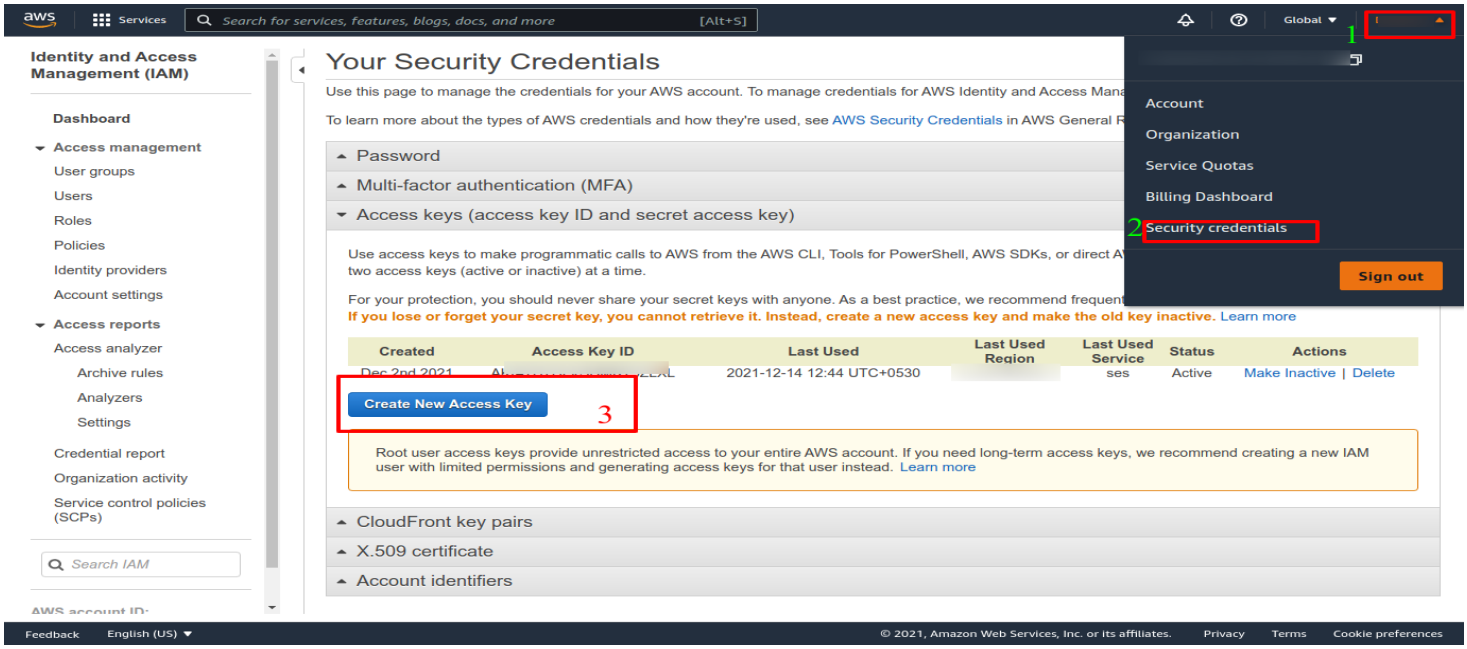


4.1.8 S3 bucket Access key and secret key download and save it in the database.

Open the Robo-3t or MongoDb compass, then navigate to DB->setting collection->update the below highlighted keys.

1) Save this key & add this to '**access_key_id**' & '**secret_key_id**'.
2) Add bucket name '**aws_bucket_name**'.
3) Set the boolean of **'is_use_aws_bucket**' to **true**.
4) Add the '**image_base_url**'.

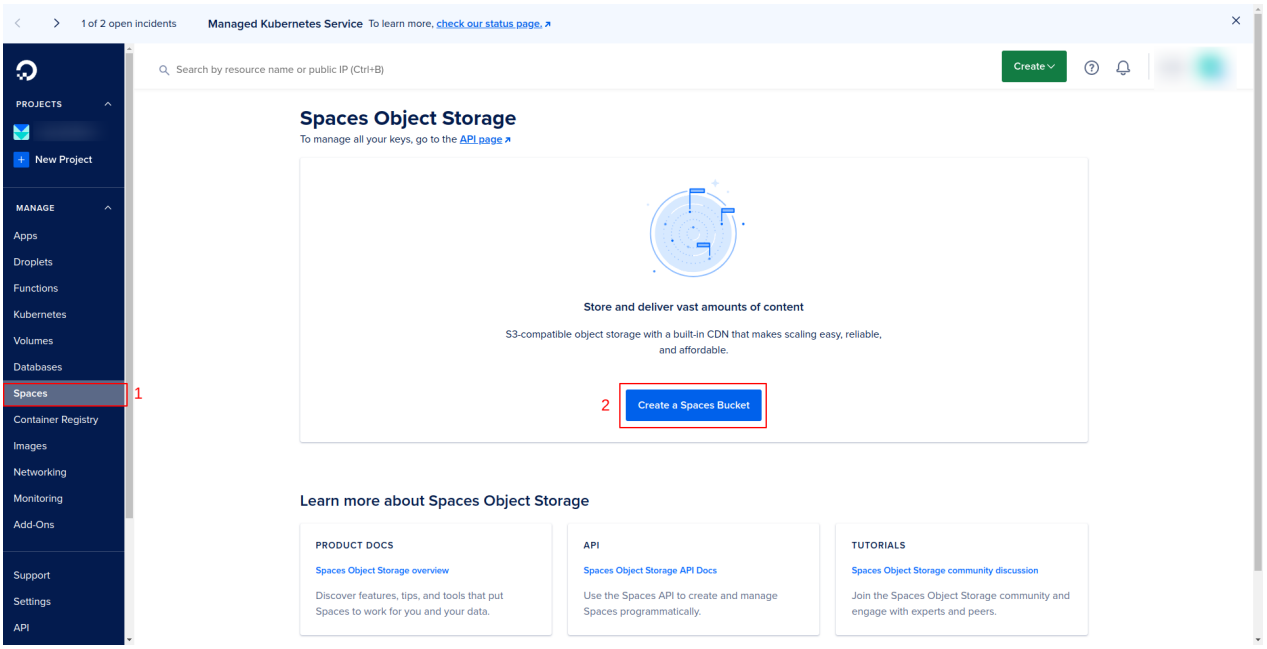

Robo 3T - 1.3
File View Options Window Help
Welcome
db.getCollection('settings').find({})
settings 3.57 sec.
Key
Value
Type
{ 156 fields }
Object
admin_notifications
admins
airport_to_cities
airports
bank_details
cancellation_reasons
cards
change_logs
cities
city_to_cities
city_types
cityzones
corporates
countries
dispatchers
documents
email_details
emergency_contact_de...
export_histories
find_provider_logs
guest_tokens
hotels
increments
information
languages
mass_notifications
partner_vehicle_docum...
partner_weekly_earnings
partners
payment_transactions
promo_codes
provider_daily_analytics
provider_daily_earnings
provider_documents
provider_vehicle_docu...
provider_weekly_earni...
providers
redeem_point_histories
redzoneareas
reviews
settings
sms_details
transfer_histories
trip_histories
trip_locations
trip_services
Logs
_id
provider_timeout
countryname
adminCurrencyCode
adminCurrency
adminTimeZone
sms_notification
email_notification
push_notification
get_referral_profit_on_card_payment
get_referral_profit_on_cash_payment
userEmailVerification
providerEmailVerification
userSms
providerSms
admin_phone
contactUsEmail
twilio_call_masking
access_key_id
secret_key_id
aws_bucket_name
is_use_aws_bucket
image_base_url
payments_base_url
api_base_url
history_base_url
is_ride_share
is_split_payment
max_split_user
admin_email
default_Search_radious
scheduled_request_pre_start_minute
number_of_try_for_scheduled_request
is_public_demo
is_provider_initiate_trip
stripe_secret_key
stripe_publishable_key
paystack_secret_key
paystack_publishable_key
payu_key
payu_salt
payment_gateway_type


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
1
Account
Organization
Service Quotas
Billing Dashboard
2 Security credentials
Sign out
Identity and Access Management (IAM)
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
Search IAM
Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Mana
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General R
Password
Multi-factor authentication (MFA)
Access keys (access key ID and secret access key)
Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, or direct A
two access keys (active or inactive) at a time.
For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequen
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. Learn more
Created
Access Key ID
Last Used
Last Used Region
Last Used Service
Status
Actions
2021-12-14 12:44 UTC+0530
ses
Active
Make Inactive | Delete
Create New Access Key
3
Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead. Learn more
CloudFront key pairs
X.509 certificate
Account identifiers
Feedback
English (US)
© 2021, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

4.2 Using Digital-Ocean Spaces

The Digital-Ocean Spaces bucket is similar to the AWS S3 bucket.

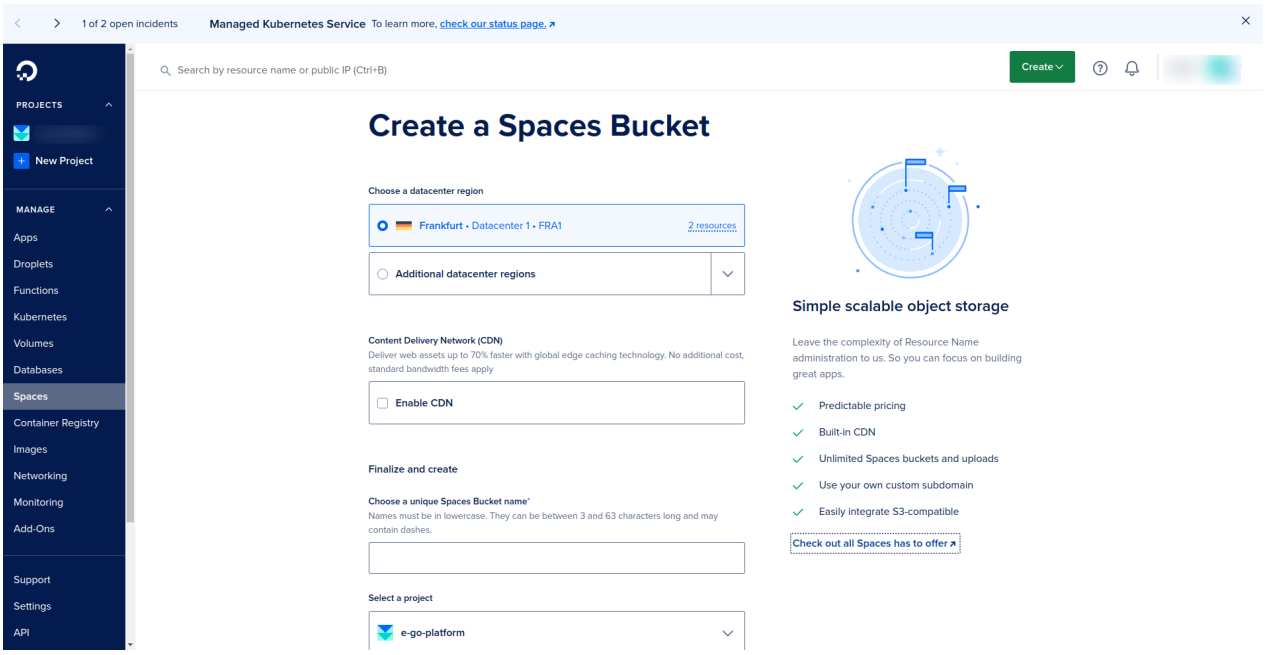
4.2.1 Create Spaces Bucket



4.2.2 Configure the Spaces Bucket

1. **Choose a datacenter region**. The datacenter region you choose will also become part of a bucket's endpoint URL. See regional availability for Spaces for more information on the available options.
2. Optionally, enable the Spaces **CDN (Content Delivery Network)**. If you click **Enable CDN**, you can customize the **Edge Cache TTL**, which is the amount of time the edge servers will cache your content.
3. Choose to **restrict file listing** or **enable file listing**. The visibility of a bucket's file listing has no effect on the visibility of the files themselves. You can change the file listing visibility at any time after creation.
4. **Choose a unique name for your bucket**. The name of the bucket makes up part of its endpoint URL and cannot be changed once it is created.

For reference click me



Once you get all the keys and details follow the similar steps mentioned here.

5. Pointing Domain name, and SSL certificate

Now as our code is running on the server, we can't open our application from search engines or web browsers using domain names.

For that, we need to setup a few things and that are:
1. Domain Name
2. SSL certificate

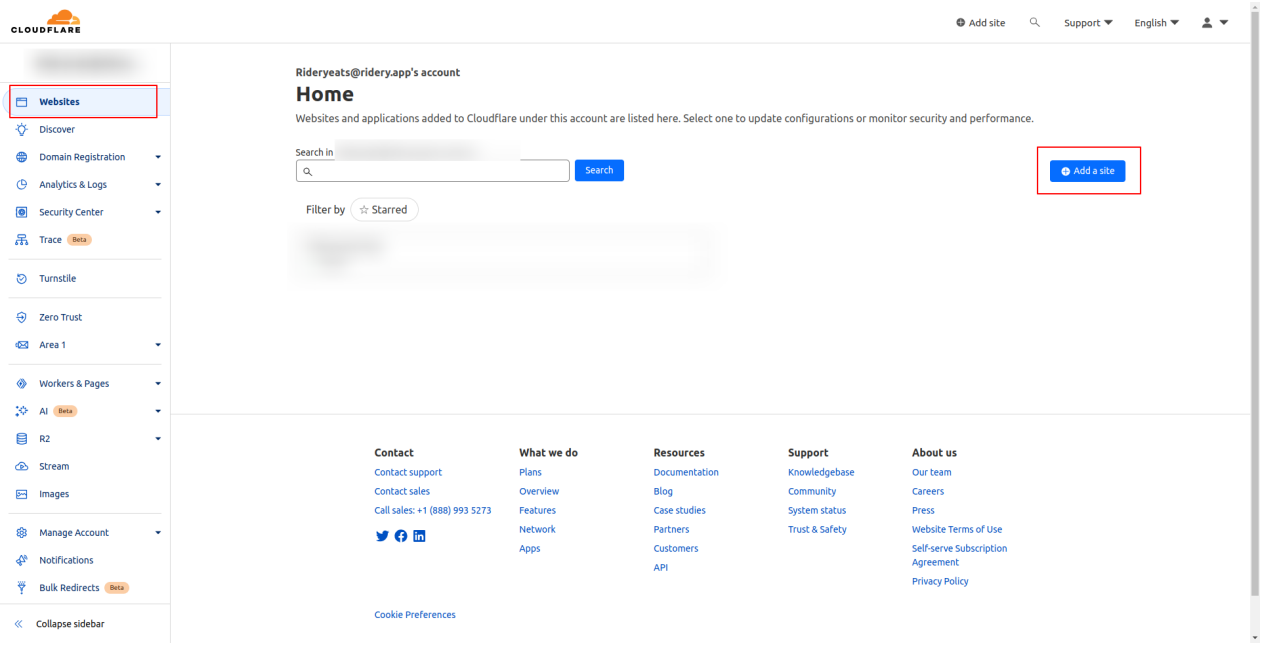
5.1 Domain Name(DNS)

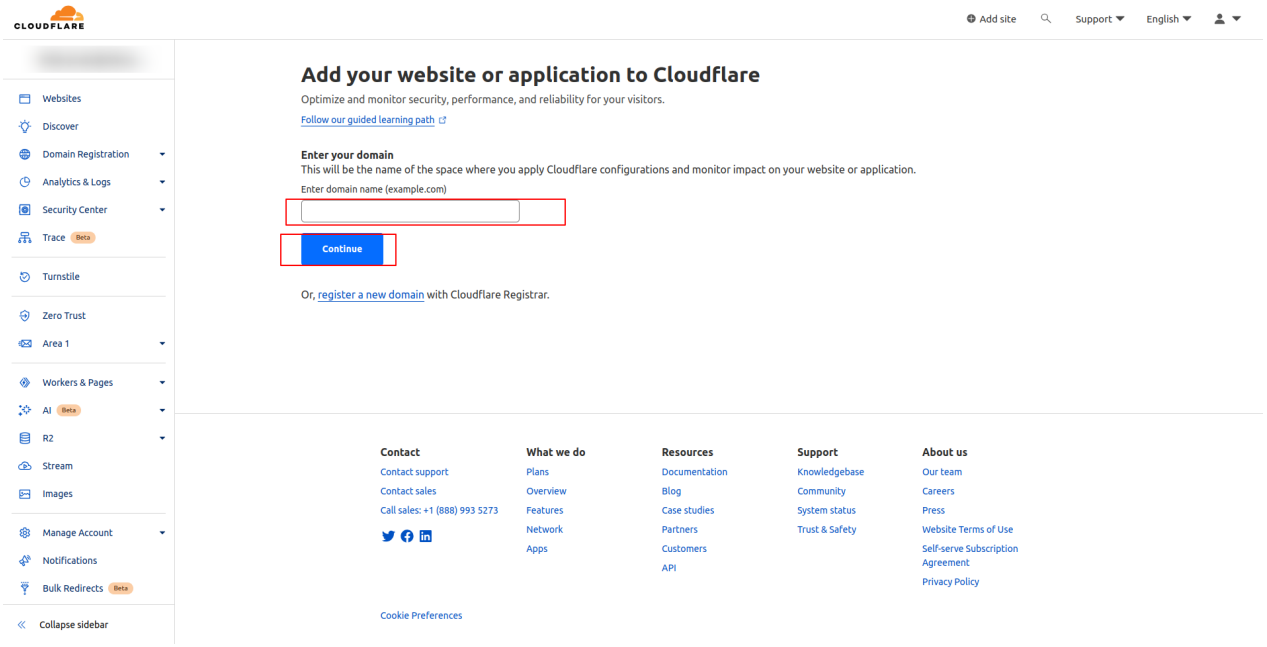
Follow the steps to add the domain to the cloudflare and configure the domain and subdomains.

For Domain configuration we are using Cloudflare (https://www.cloudflare.com). Register your account and go to the DNS section and click Add Record. On the given input field in the type section select A (A records) and in name enter @ (refers to main domain and also it will be the user panel) and in IPv4 enter the IP address of our server (the elastic Ip which we generated previously or the IP of the server), then click Save. For subdomains click add new record again and in the type section select A (A records) and in the name section enter admin and in IPv4 enter IP address. Similarly, add 10 more subdomains for api, history, payment, notification, driver, corporate, dispatcher, partner, hotel and hub.

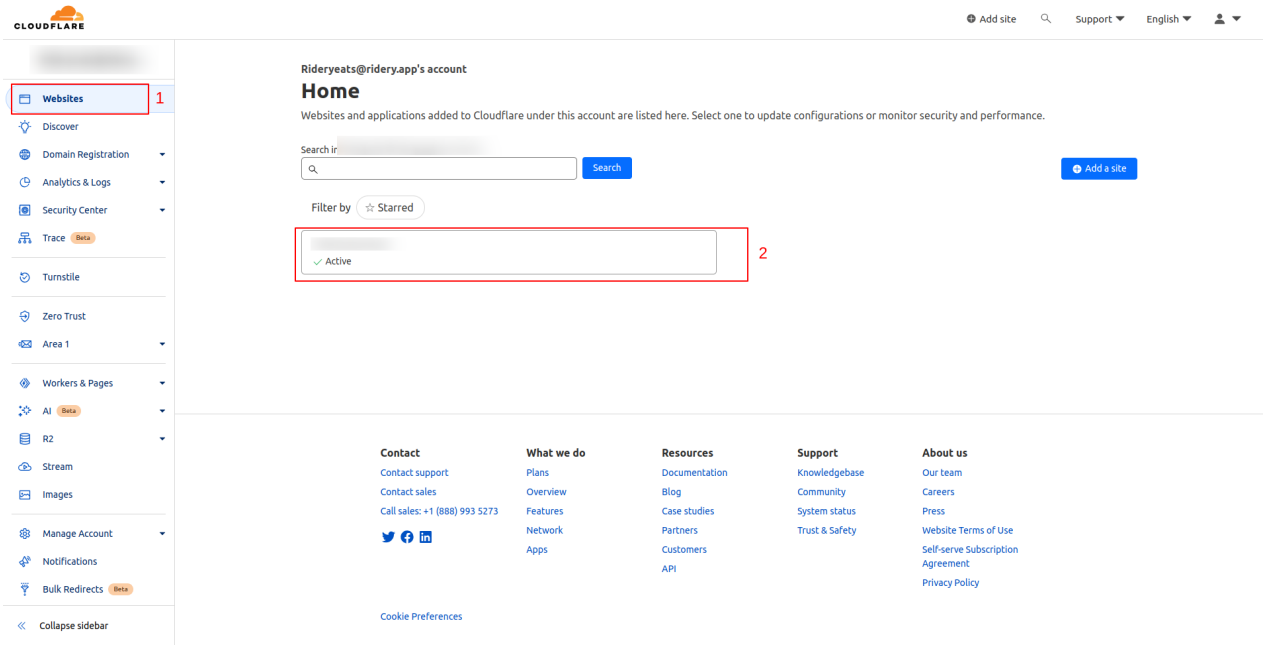
And now we are done with our Domain pointing.

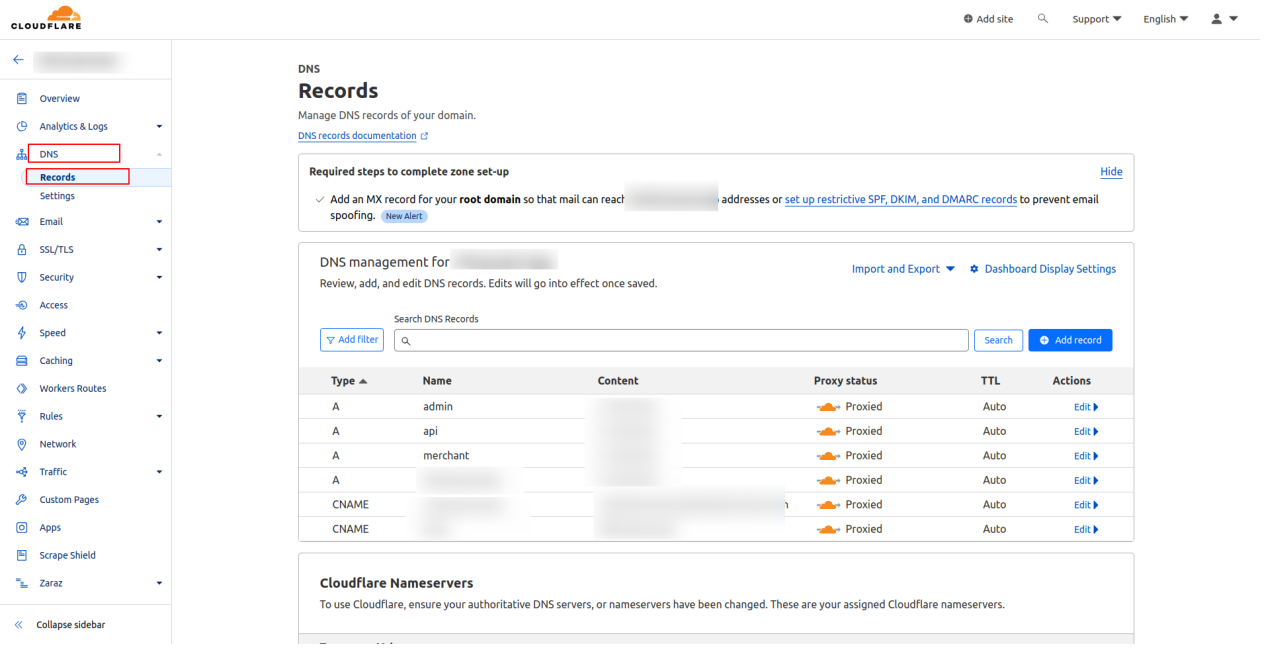
Follow the steps as below highlighted.



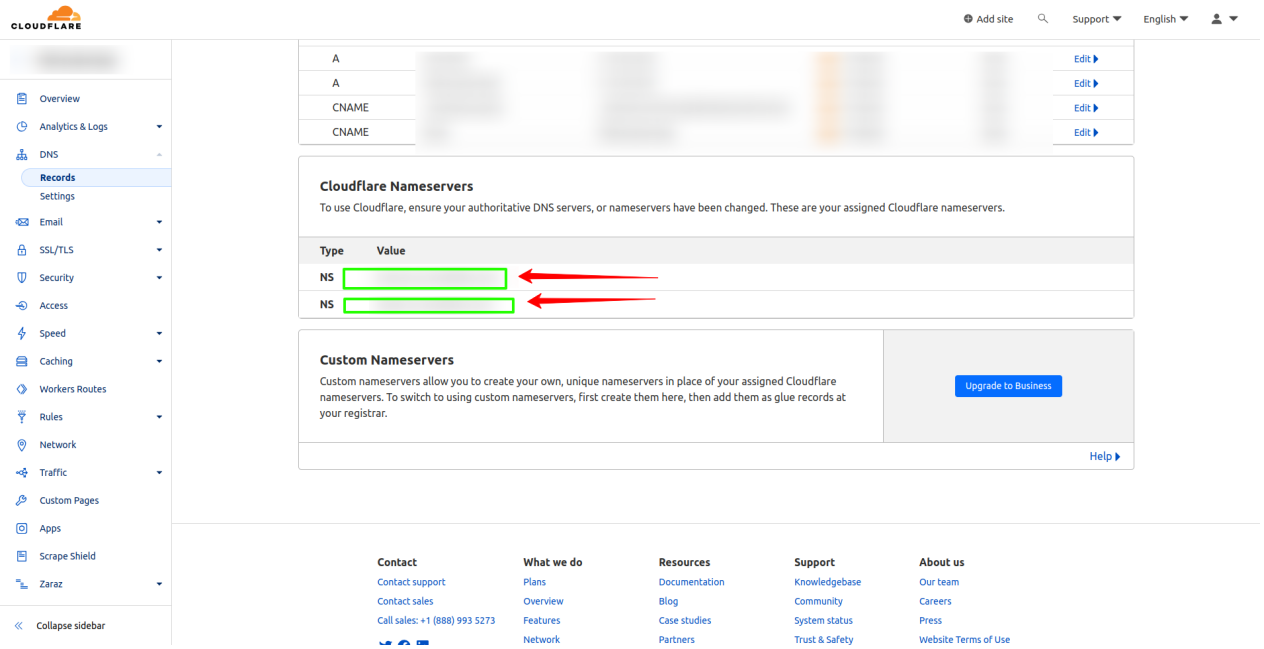


After adding the **domain** and once active follow the steps.





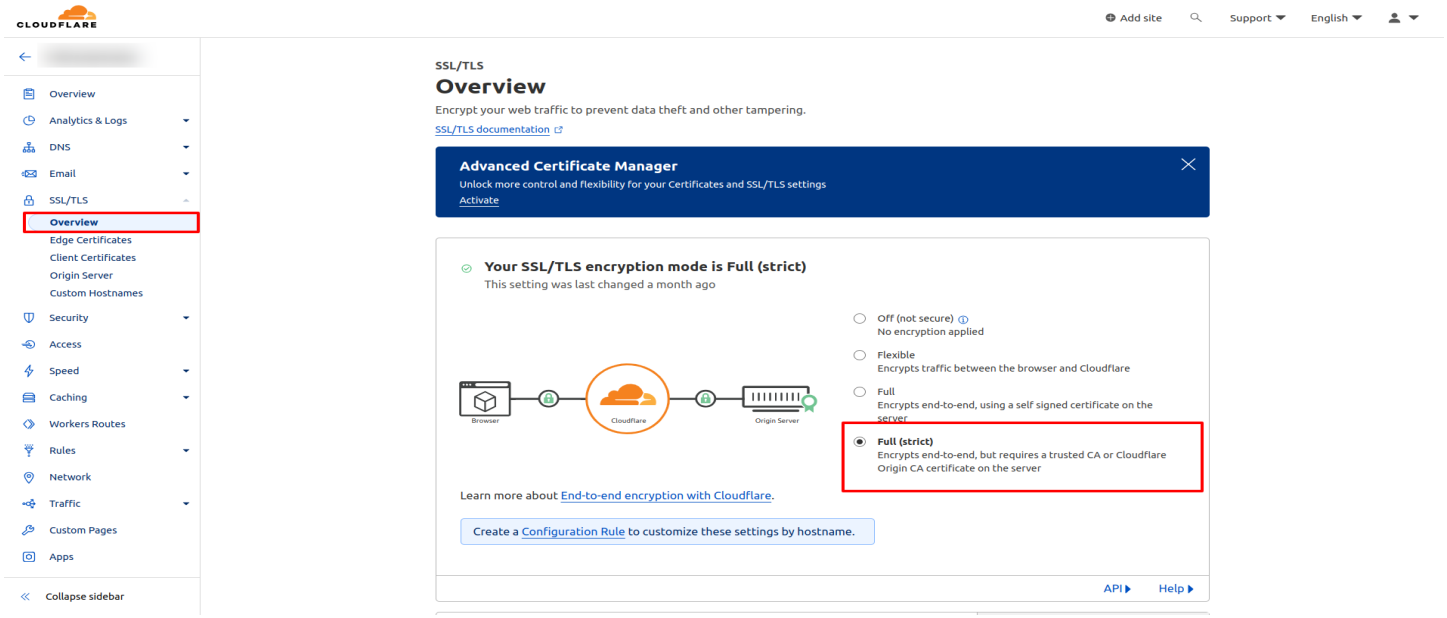
Moreover we need to replace these nameservers on the domain provider platform with the nameservers provided by cloudflare.



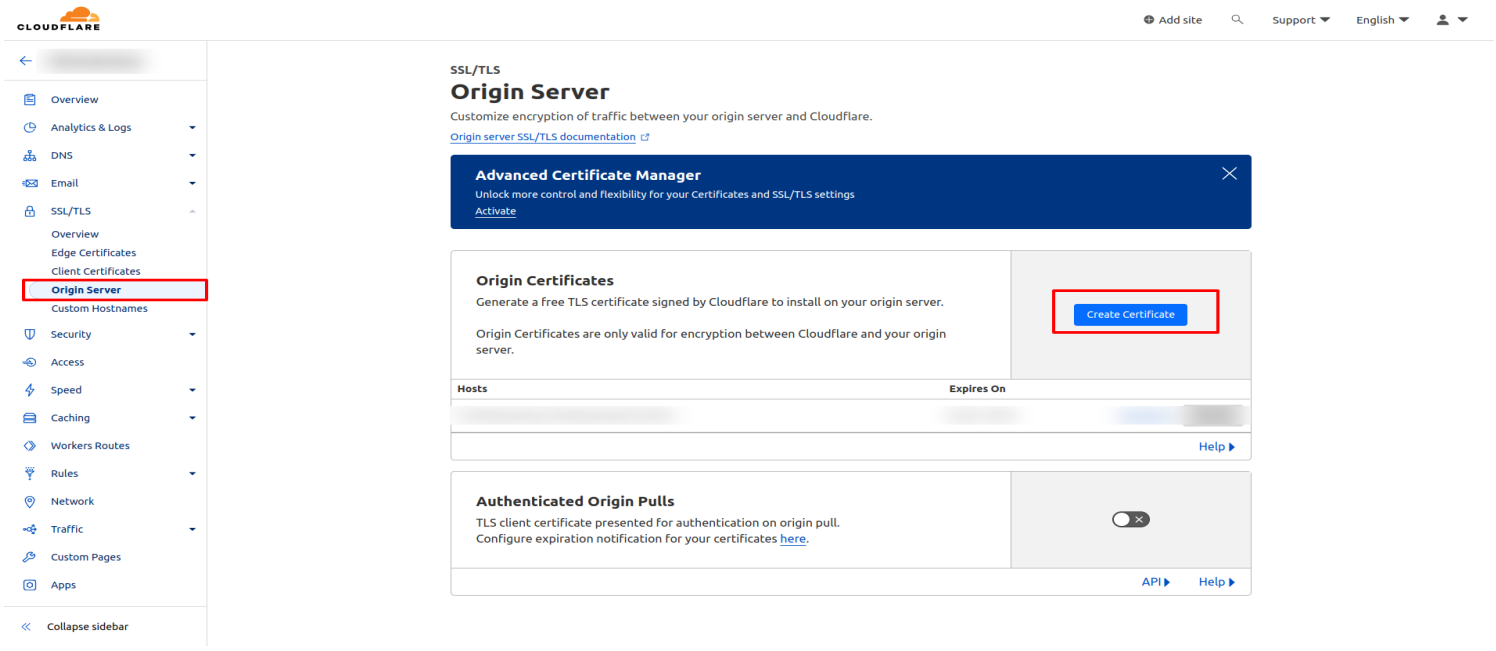
The changes will take few minutes or even more, keep checking the overview tab.

5.2 SSL certificate

Now we have a domain pointed to our IP address, let's create an SSL certificate to secure our domain. Let's head towards Cloudflare for SSL certificates (https://www.cloudflare.com). Head towards the SSL/TLS section(SSL\TLS > Overview) and select your plan(Preferred full strict). For certificates go to **SSL/TLS** -> **Origin Server** to generate certificates. Please check screenshots for References.



Store the **private** and **origin** pem files at a safe place(it will be required in further steps) as these are SSL certificates.



6 Establishing SSH connection

Establishing SSH connection to connect server and installing environment for our code to run.

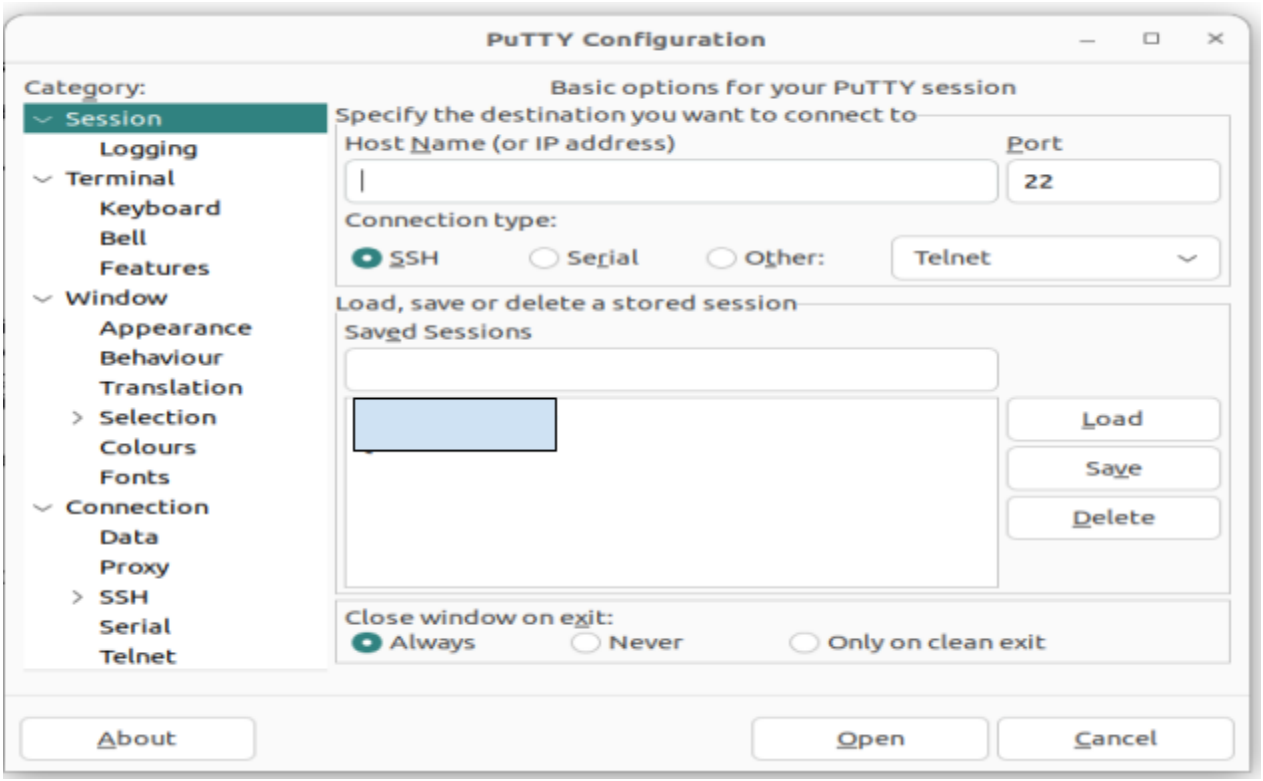
We will be Using putty for establishing SSH connection between our system and server.

What is **PuTTY** ?

PuTTY is a free and open-source terminal emulator, serial console, and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. The name "PuTTY" has no official meaning.in short, it is used for connecting the ubuntu server in any operating system for application installation on the domain server.

6.1 Open PuTTY **or** Terminal

Follow the steps to connect the server using Putty or with a local Terminal.

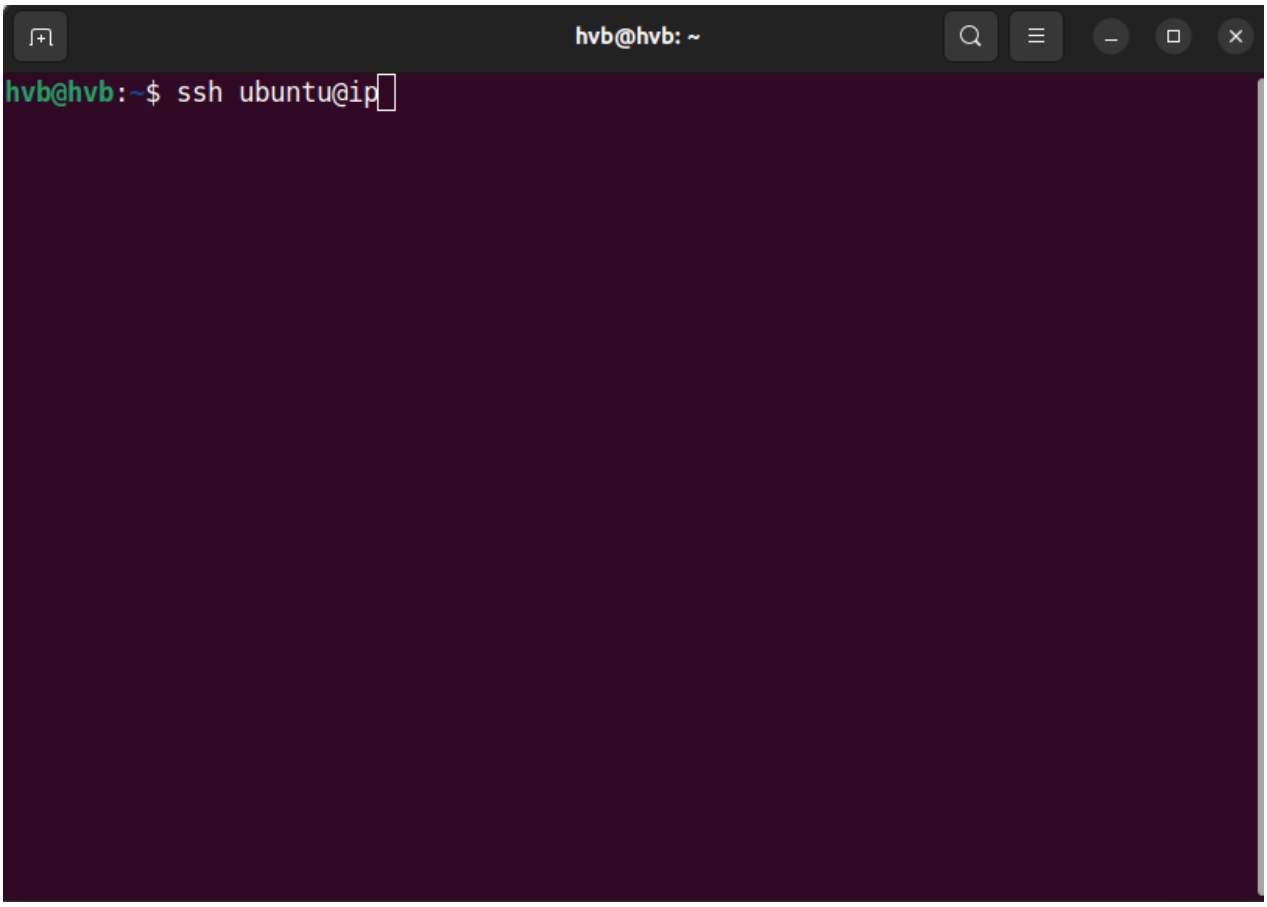


6.2 Connect with Password.

- Connect using local Terminal

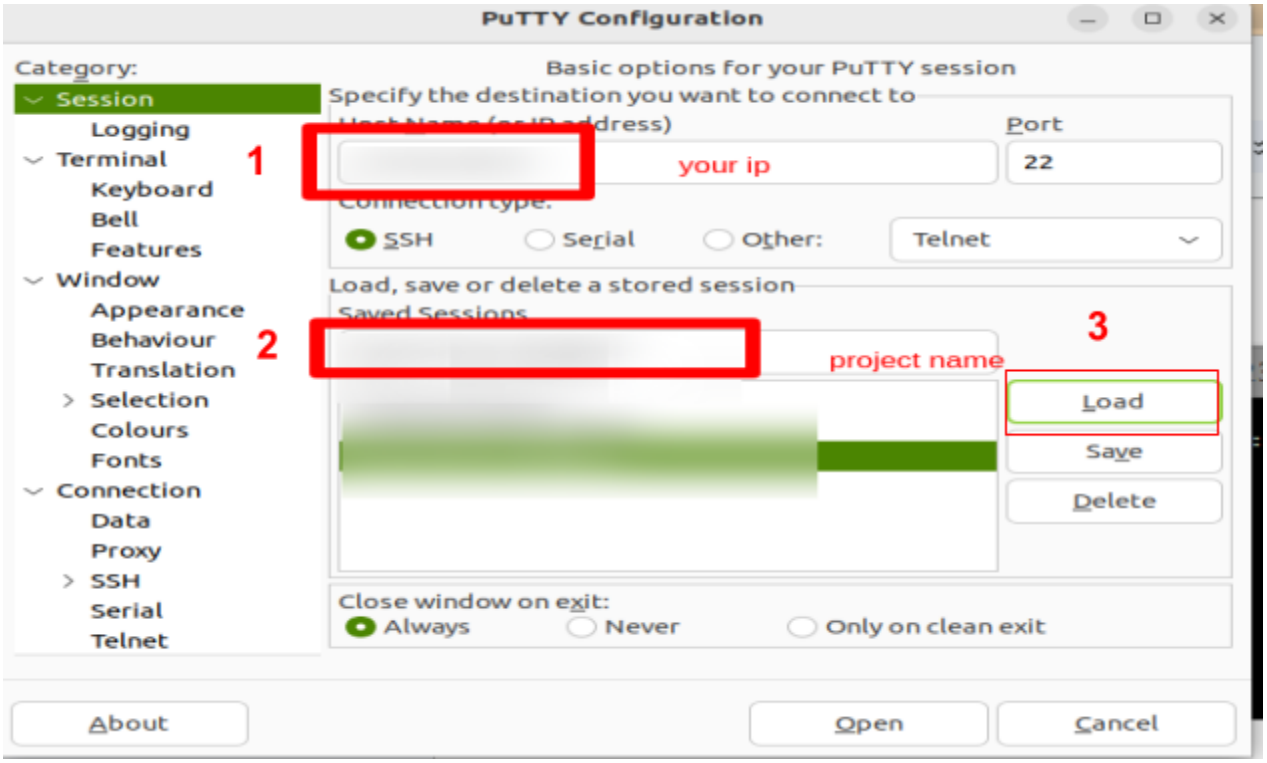
  Root : ssh username@ip

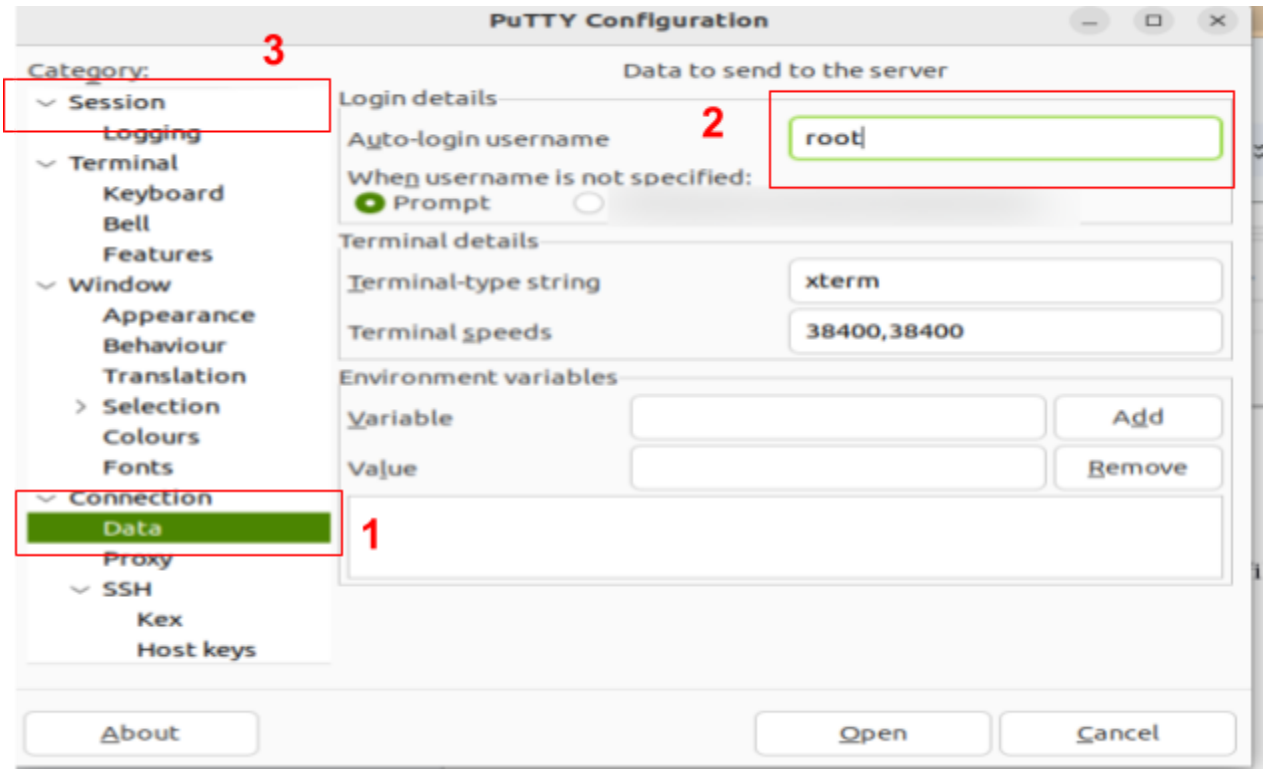
  Replace username with **ubuntu** or **root** accordingly.
  Replace ip with actual Ip address of the server.



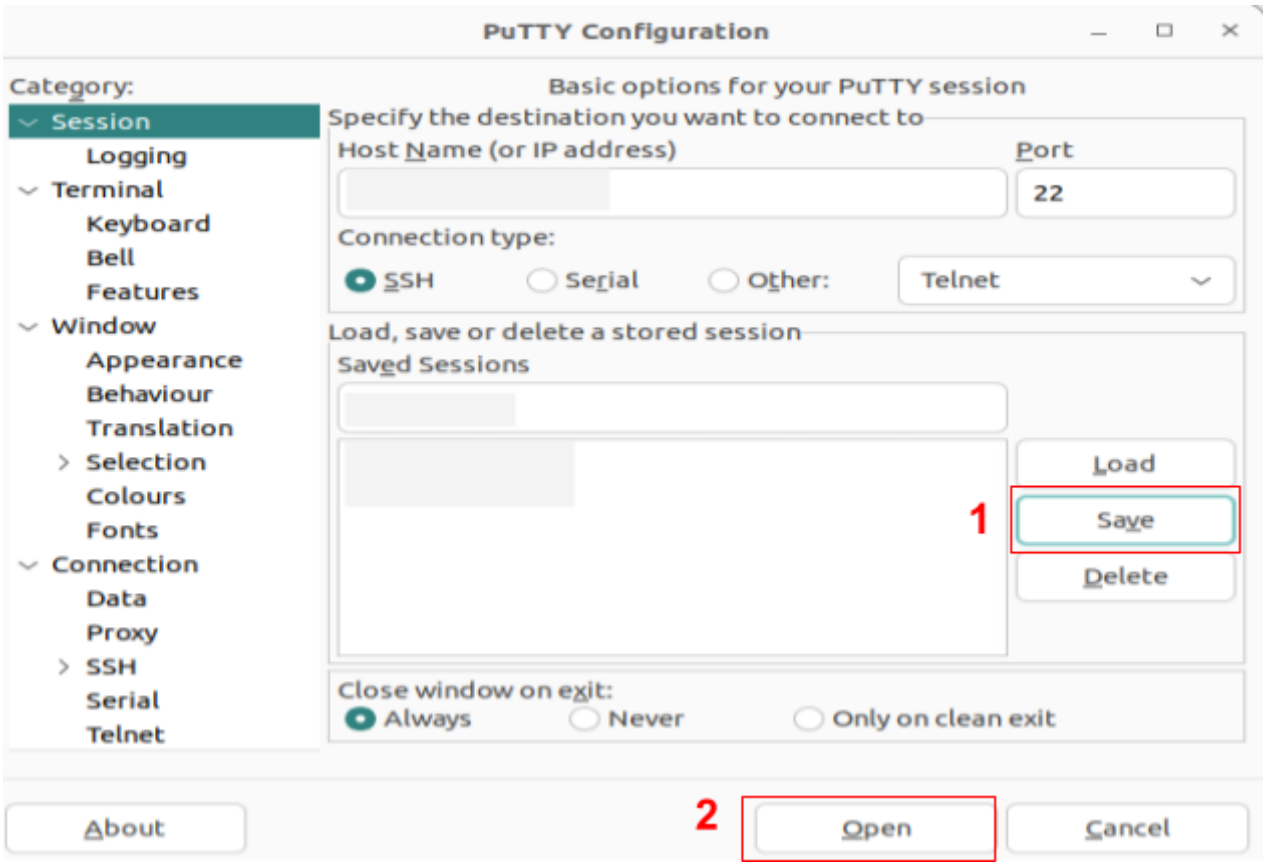
- Connect using Putty

  Follow Steps as highlighted

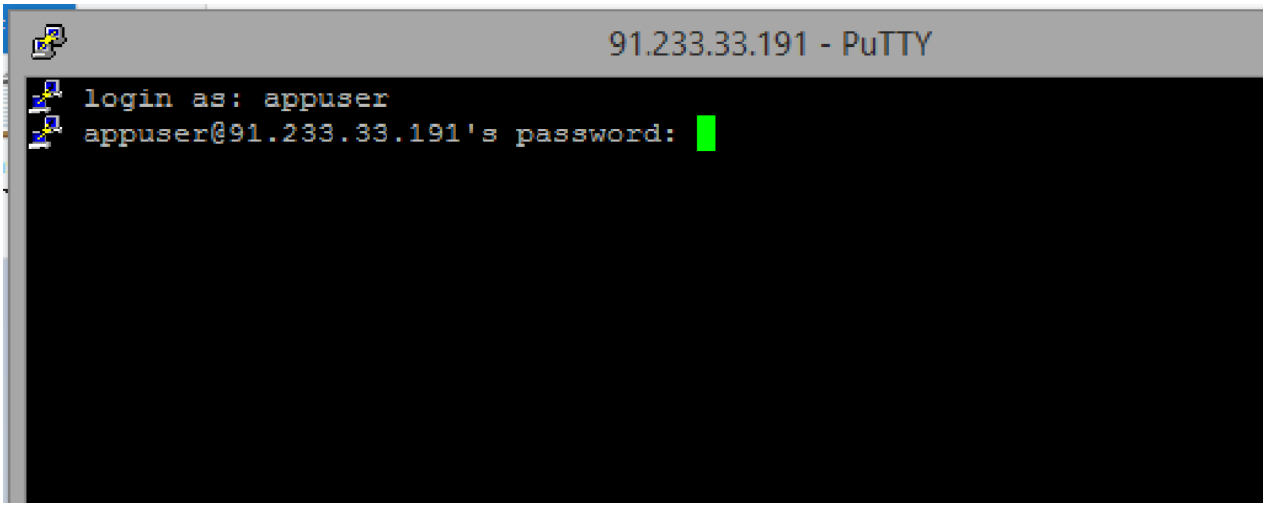




After this go to the session page and save the detail try to connect with open button.



Use your password of root and you will get connected.

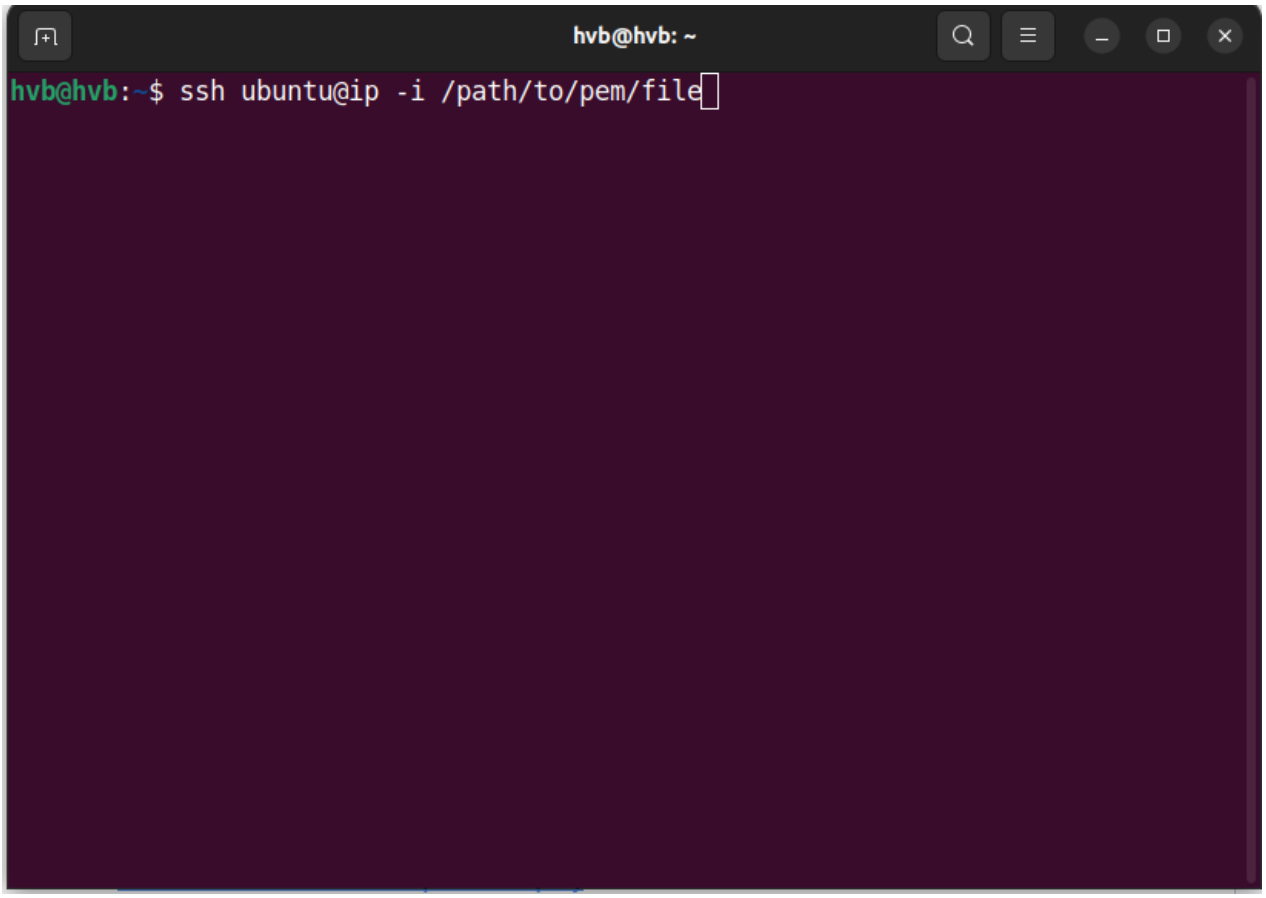


6.3 Connect with PEM / PPK File

- Connect using local terminal
  ubuntu : ssh username@ip -i /path/to/pem/file

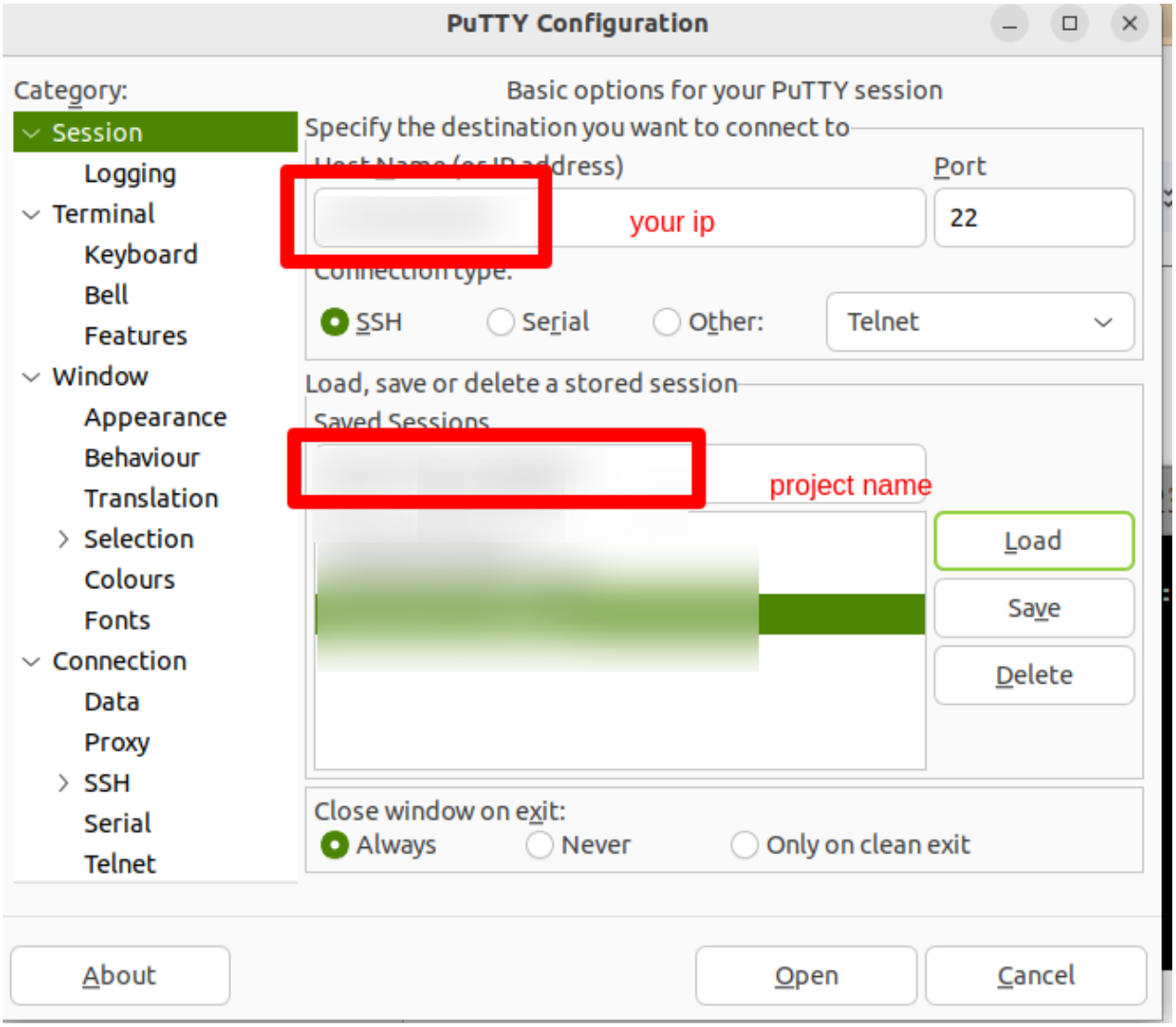
  Replace username with ubuntu or root accordingly.

  Replace ip with actual Ip address of the server.

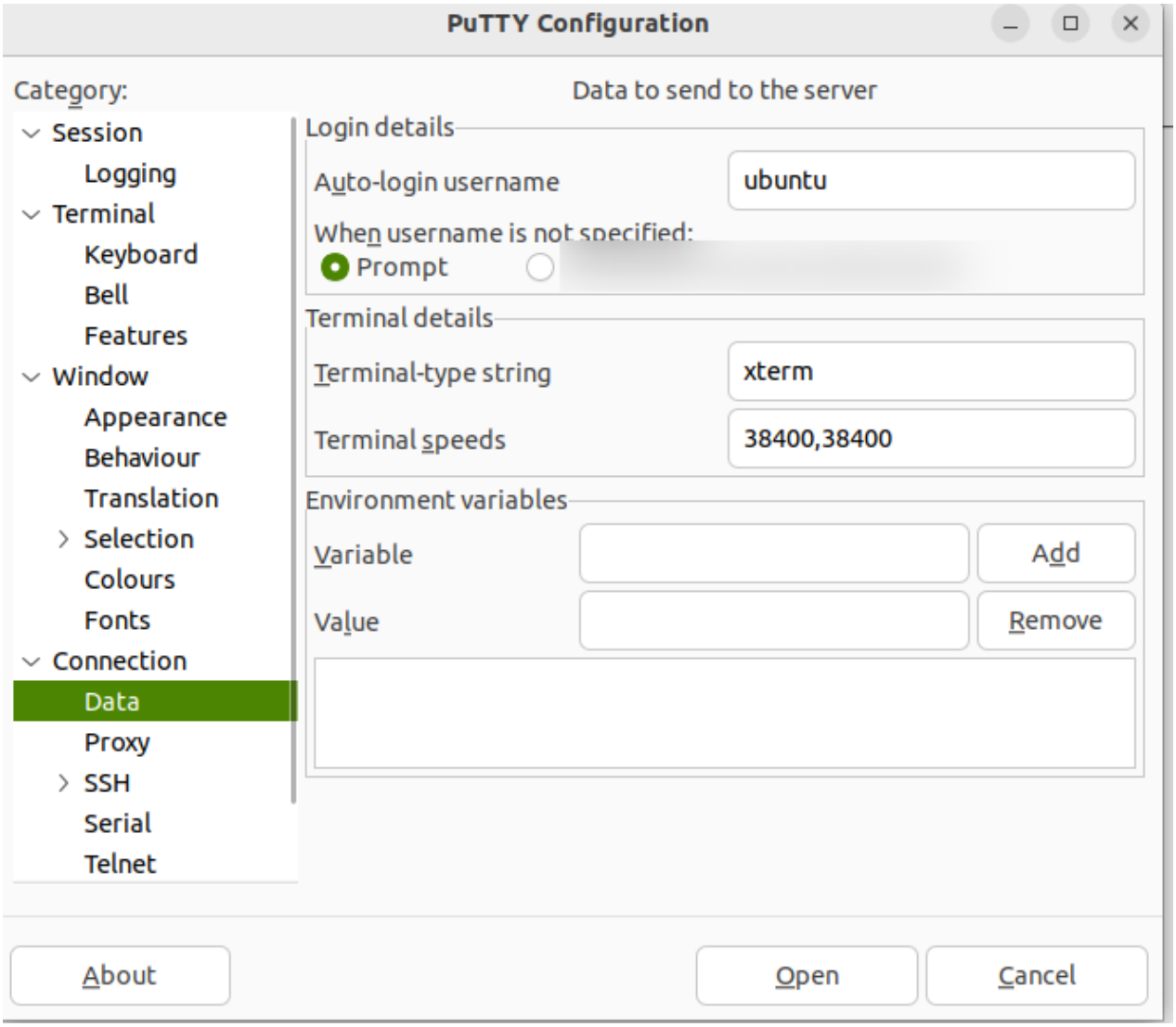


- Connect using Putty

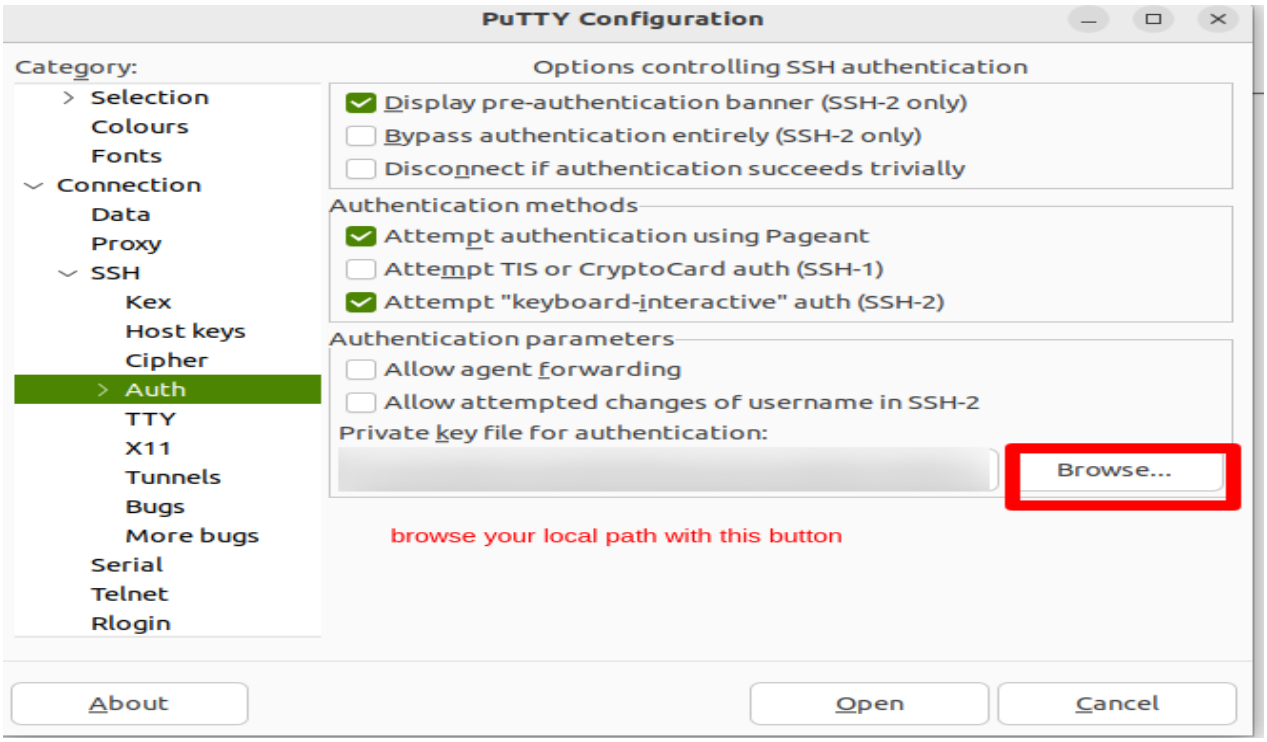
  Follow steps as highlighted.



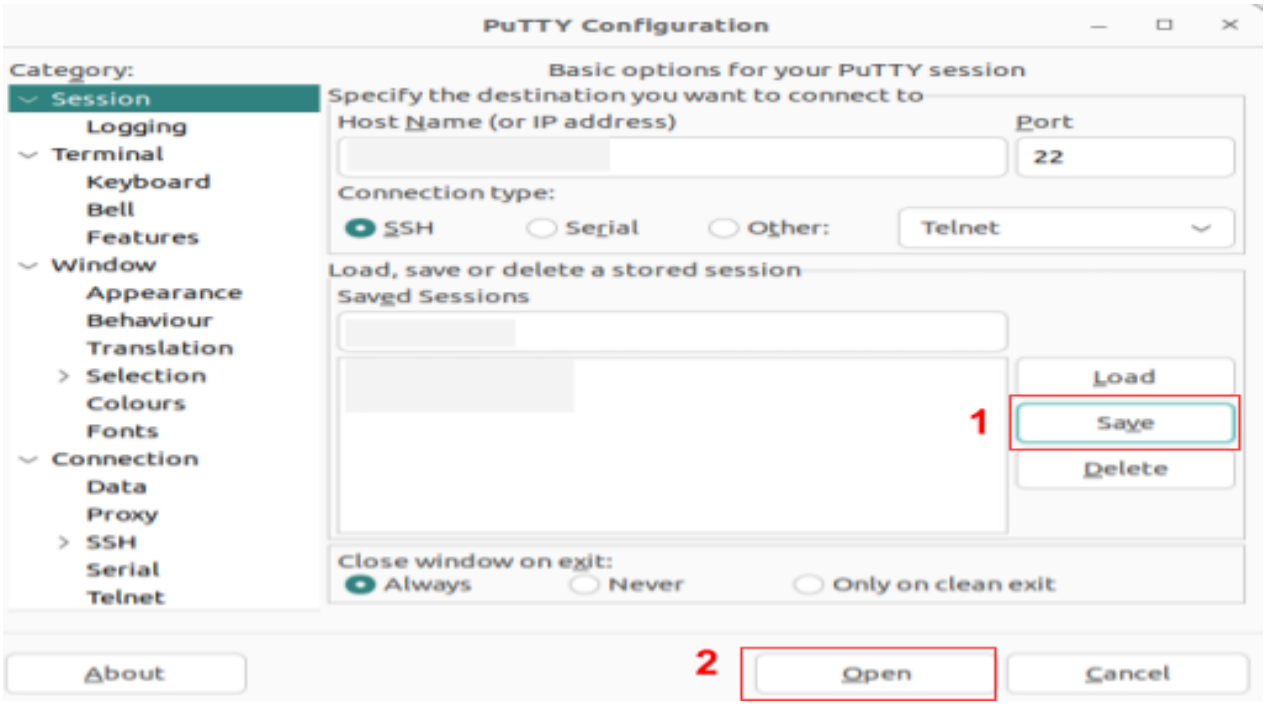
In the **Auto-login username** field replace the username with root or ubuntu accordingly.



Then navigate to the **Auth** tab and select the PEM or PPk file.



After this, go to the session page and save the details, try to connect and it will get connected to the server.



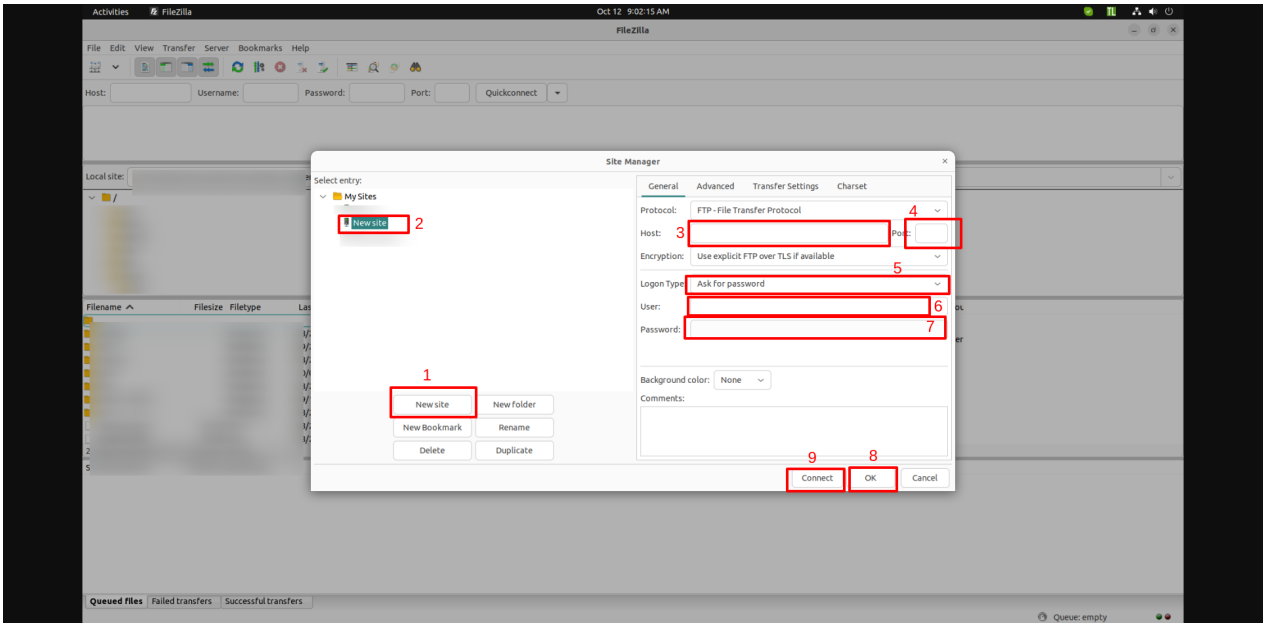
6.4 Perform the below steps to install all dependencies.

6.4.1 Place the bash file and SSL using FileZilla

- Place the **EBER_INSTALLATION.sh** file and **SSL** certificate by following the mentioned steps.

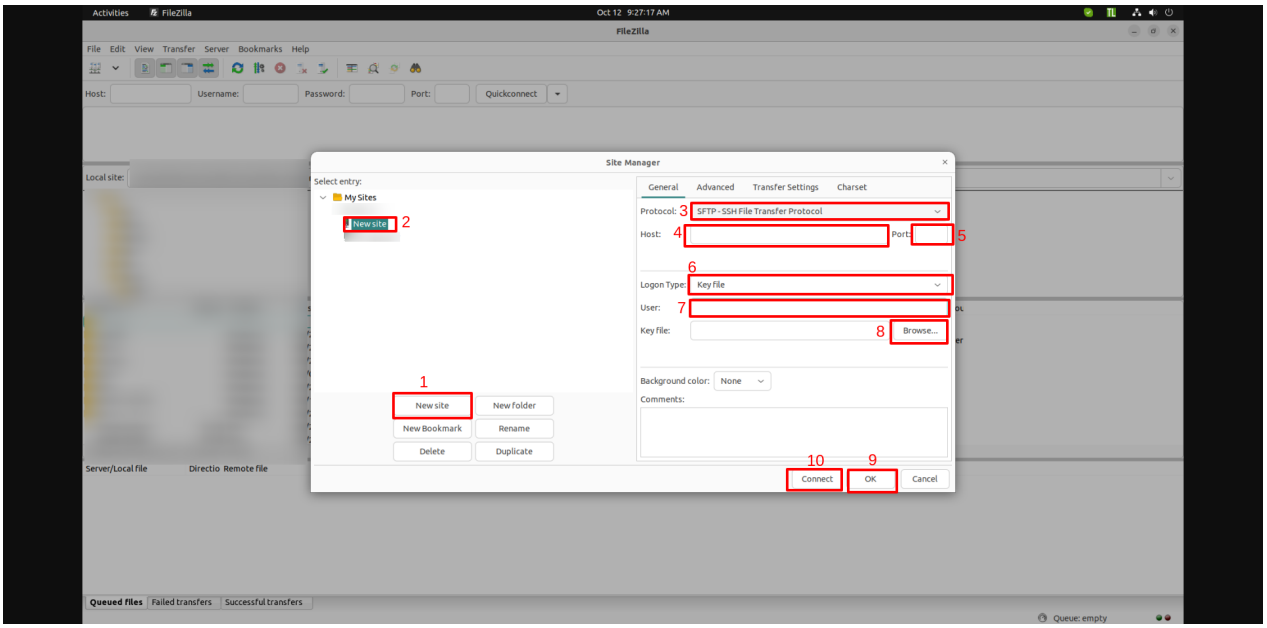
    - Connect to Server using fileZilla.

        - With Password Credentials
            - **New Site**
            - **Rename**
            - **Server IP**
            - Port **22**
            - Select Logon Type **Normal** from the list
            - Username **root** or **ubuntu**
            - **Password**
            - **OK**
            - **Connect**



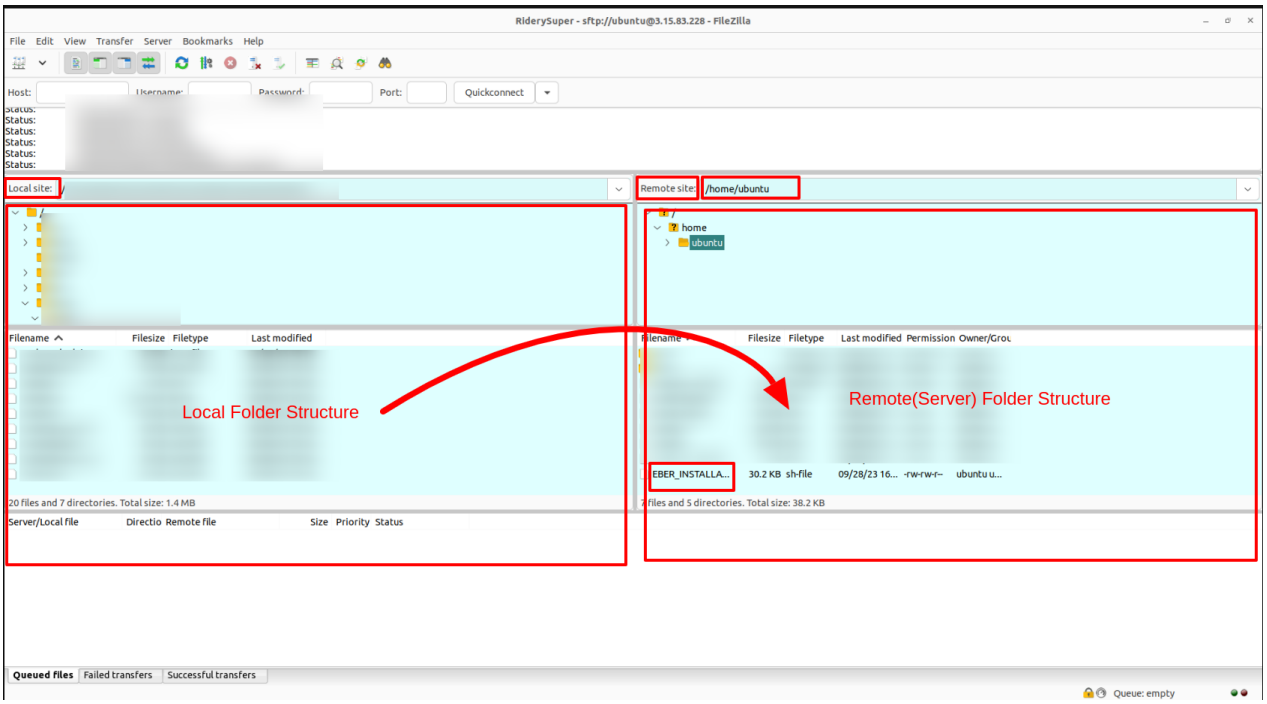
- With Pem/Ppk file key

    - **New Site**
    - **Rename**
    - Protocol **SFTP**
    - **Server IP**
    - Port **22**
    - Select Logon Type **Key File** from the list
    - Username **root** or **ubuntu**
    - **Browse** file
    - **OK**
    - **Connect**

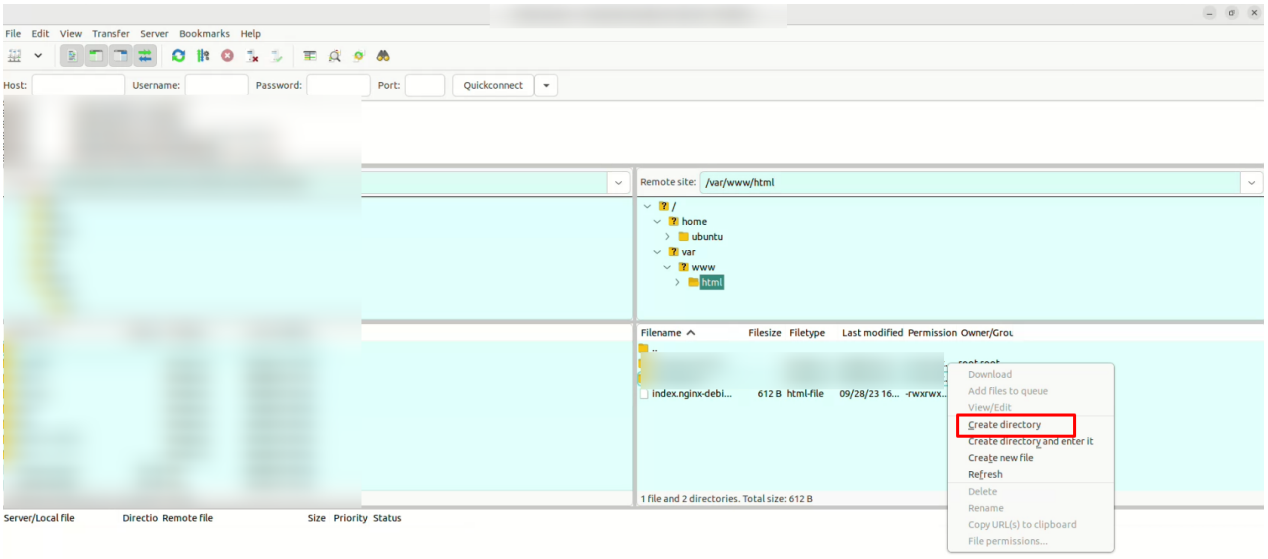


    - Place the files on the server manually.

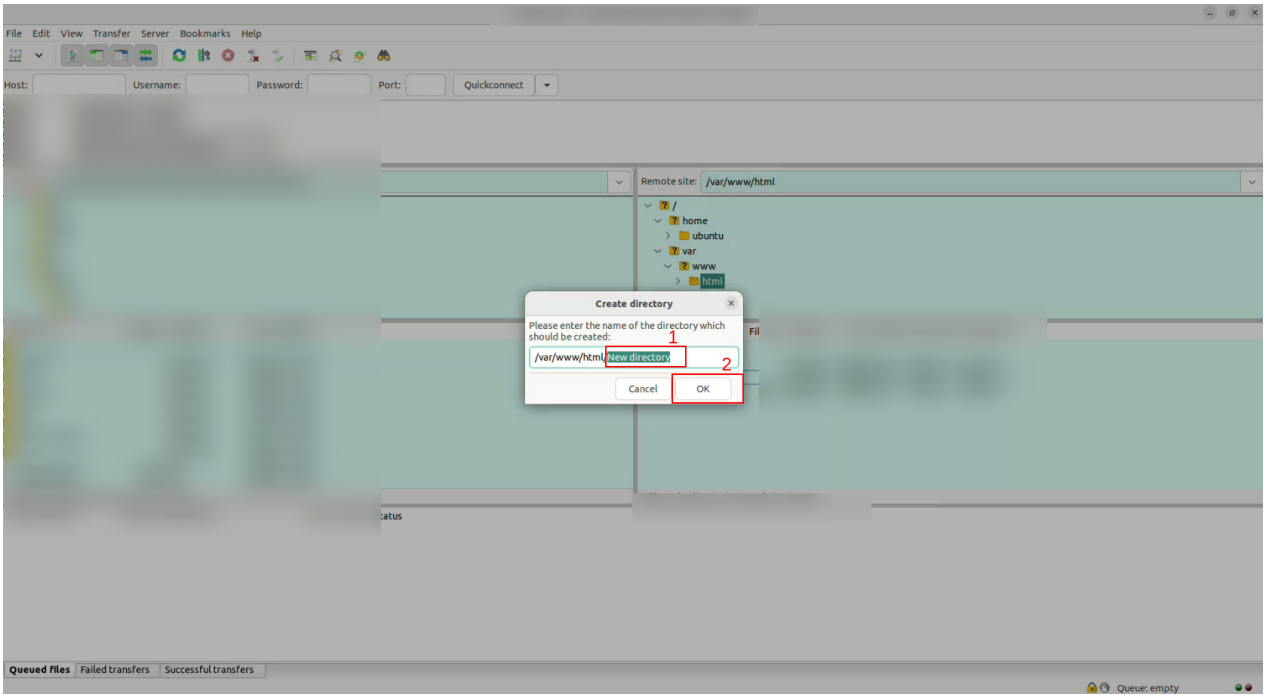
        - EBER_INSTALLTATION.sh bash file
            - After connecting successfully navigate to the folder path(/home/**ubuntu**(or)**root**) highlighted in the remote folder structure.
            - Then after drag and drop the bash file from the local folder structure to the remote folder structure.



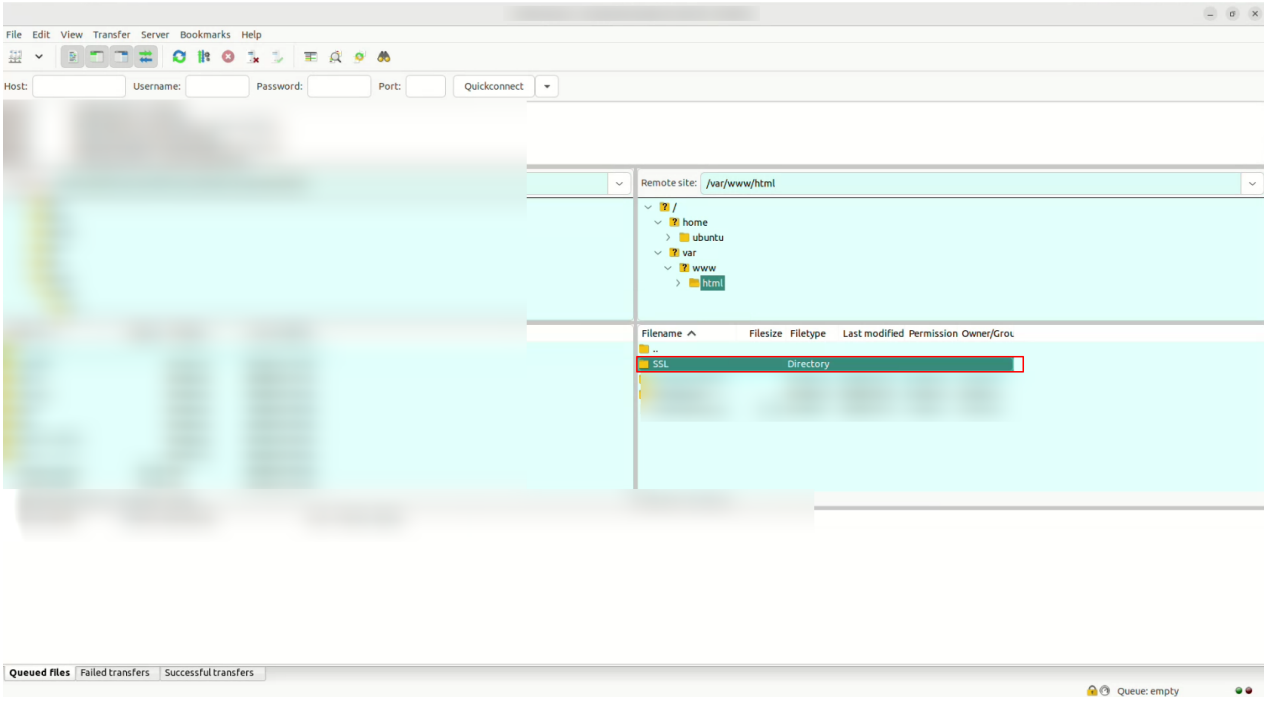
- SSL certificates
  - Right click on the remote folder structure create new directory



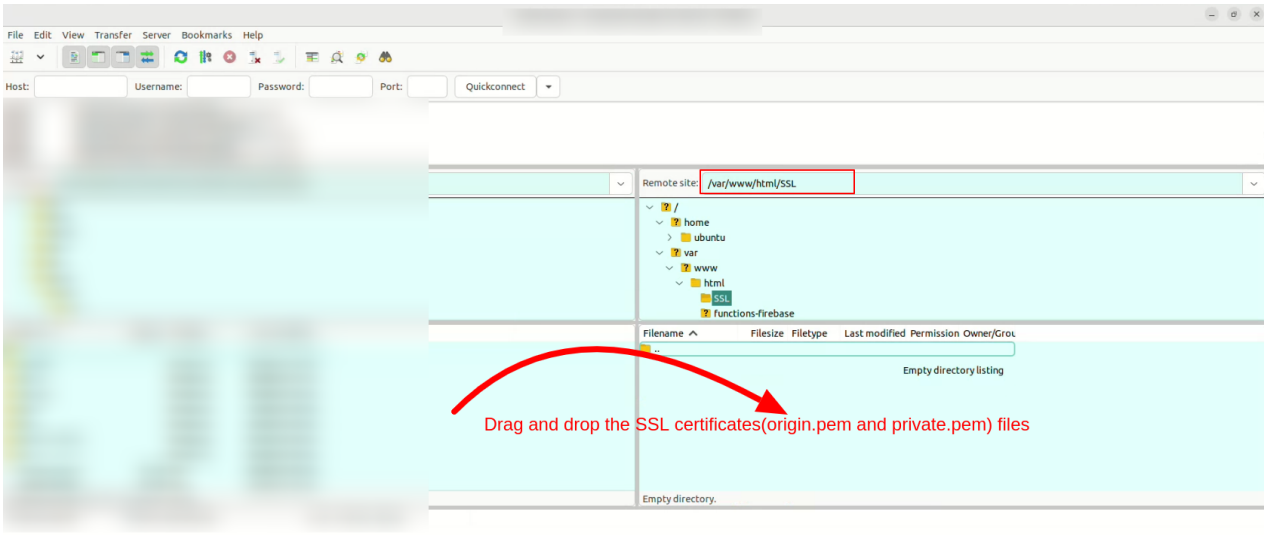
  - Rename the **new directory** with **SSL(**keep the folder name in UPPERCASE**)** then press OK.



- Open the **SSL** directory



- Confirm the path as highlighted and then drag and drop the SSL certificates(origin.pem and private.pem) files.



- Now we have placed the **SSL** certificates and **Bash** file on the Server

6.4.2 Loading the Server with all the requirements.

Perform the below steps to install all dependencies or important server domain:

Command to change permission from root (Used in AWS) :

sudo chown <hostname>

hostname = root/ (for digitalocean/vultr domain)

ubuntu/ (for AWS domain)

Installing environment for our code to run

Prerequisite:

- Code Repo Urls
- Basic Knowledge of command line.

○ Place installation bash script file in the instance where you want to perform installation(we done already at step 6.4.1).

○ Run this file with the use of following command:

bash BASH_FILE_PATH

○ It will ask you to enter choice,
Press **1** if you want full Installation.
Press **2** if you want custom Installation.

1) For Full Installation , you have to do following steps

○ This process will install all required dependencies, packages and applications automatically.
○ This installer file has appropriate description at each stage.
○ It will ask for version selection for Nodejs(Step-2) and MongoDB(Step-5).
○ Keep the git urls ready with you(Step-10). At every stage you will be asked to select option and appropriately select the option and set up the project.
○ In the "default file" configuration stage it will ask to input total number of domains along with the port number on which associate server is running (Step-12).
○ If you already have the SSL certificates, skip (Step-13).

2) For Custom Installation, you will get following list

0) Exit
1) Update Packages
2) NodeJS
3) Angular
4) Nginx
5) MongoDB
6) Redis
7) Nodemon
8) PM2
9) Install Git
10) Setup Your Project
11) PM2 startup
12) Setup Nginx Default File
13) Install Letsencrypt

○ You can choose from the above list.

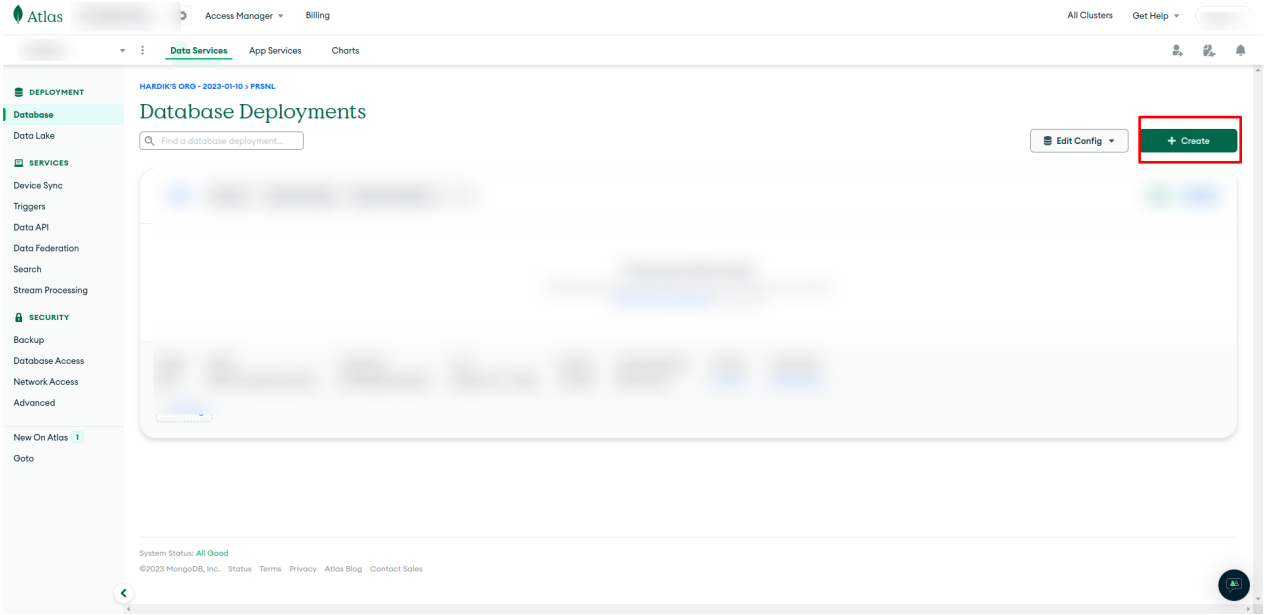
Note:
If you select the default nginx file config then you have to select 12 numbers of domains here.

-user panel - port - 4000 - domain - <domain name>
-backend server(main) port - 5000 - domain - api.<domain name>
-backend history port- 5001 - domain - history.<domain name>
-backend payment port - 5002 - domain - payment.<domain name>
-backend notification port - 5003 - domain - notification.<domain name>
-driver panel - port - 6000 - domain - driver.<domain name>

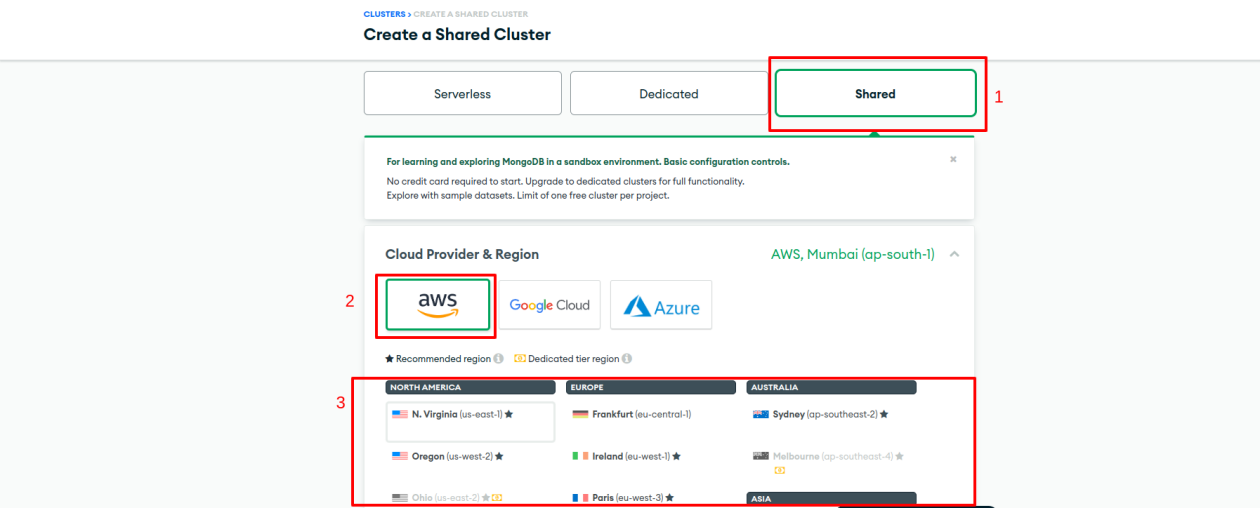
-hub panel - port - 6001 - domain - hub.<domain name>
-dispatcher panel - port - 7000 - domain - dispatcher.<domain name>
-corporate panel - port - 7500 - domain - corporate.<domain name>
-hotel panel - port - 8000 - domain - hotel.<domain name>
-partner panel - port - 8500 - domain - partner.<domain name>
-admin panel - port - 9000 - domain - admin.<domain name>

6.5 Create MongoDB Atlas

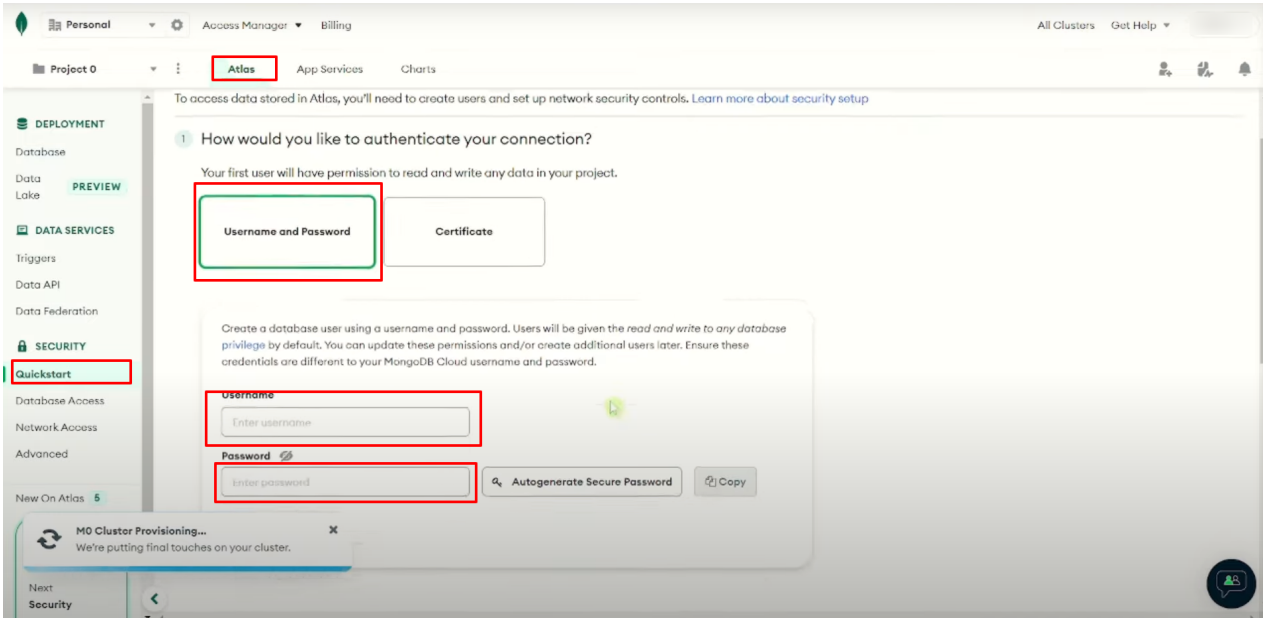
- Mongodb Atlas
  - Account
    - Login or Register
      - If you registered a new account, check the mail and verify your email.
  - Create DB



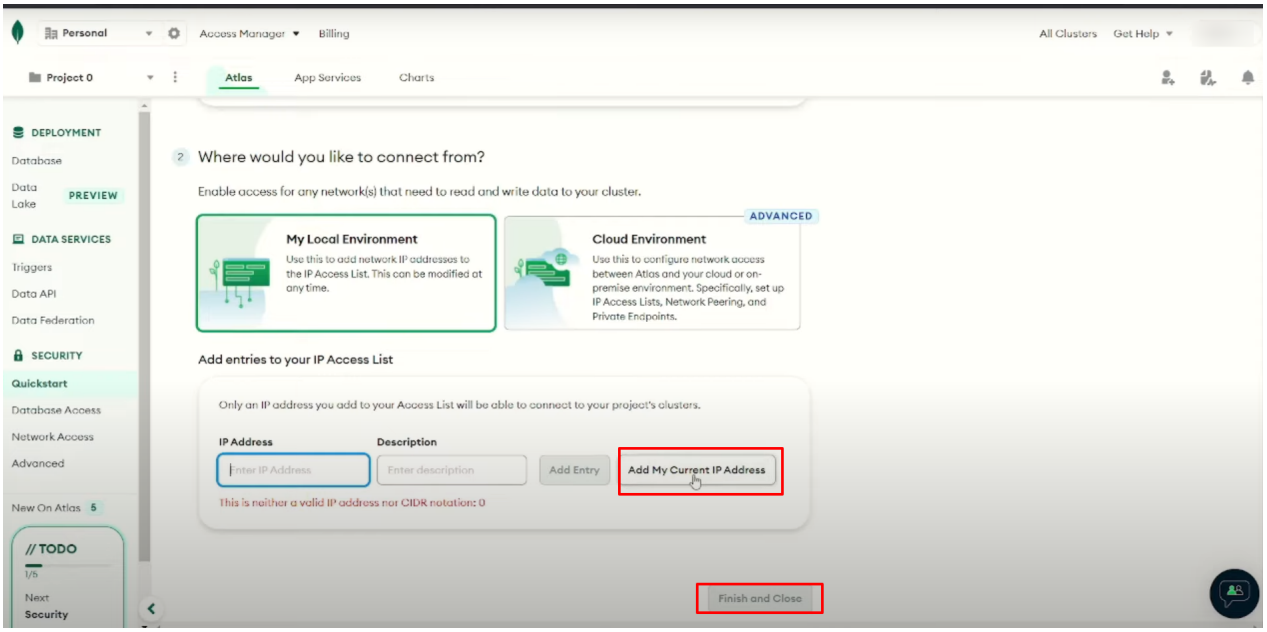
Select **Shared** as it is free and it provides us up to 512 MB.



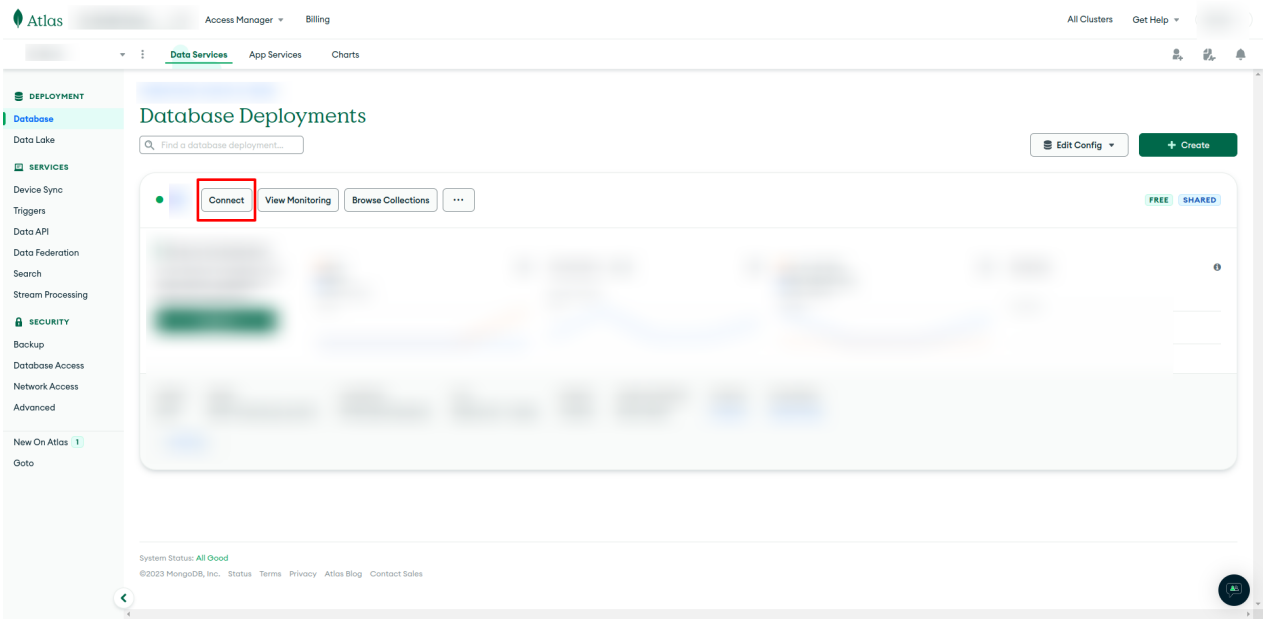
Fill the username and password(store at a safe place) to access the db.

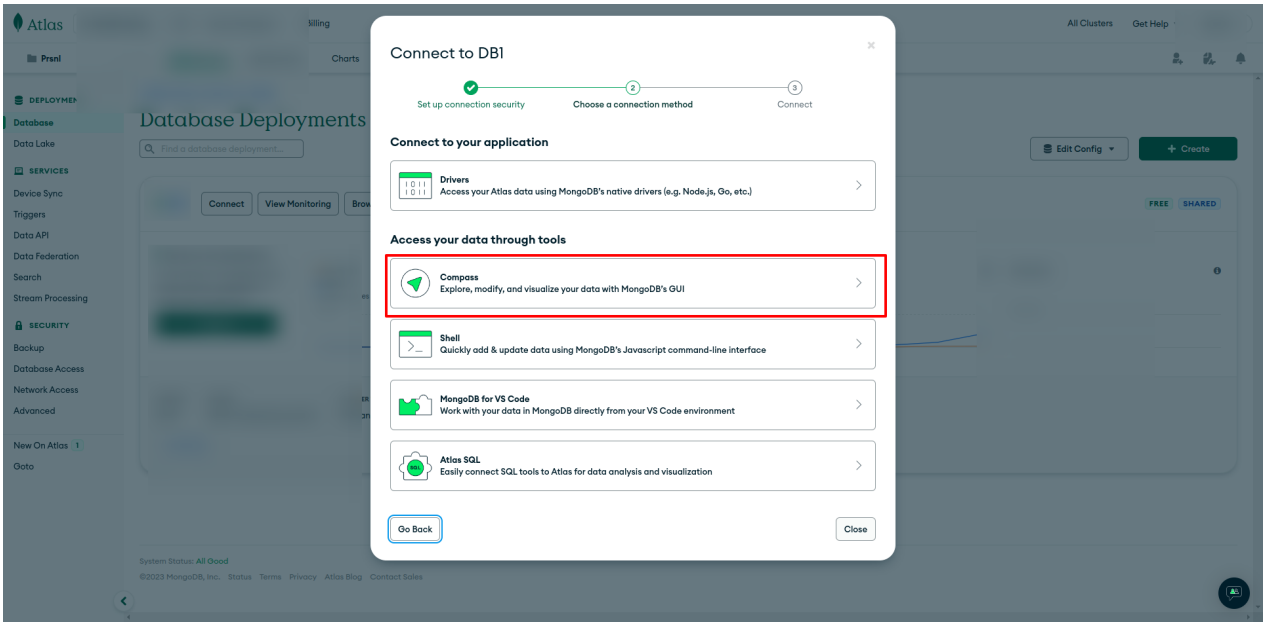


Add your current Ip address so that you can access the db and also add the IP address of the server obtained while creating the server on cloud.
Then click finish and close.



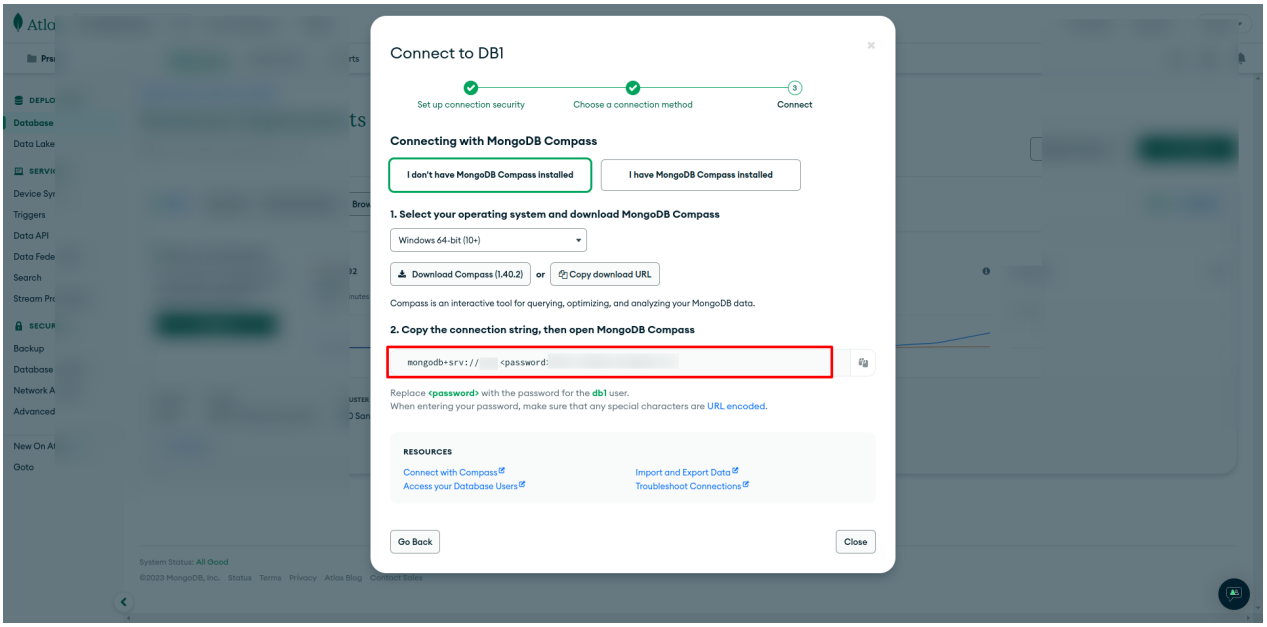
- Get the DB url to connect it with the project.





Save this string.

ie. ***mongodb+srv://username:password@xyz.mongodb.net/***



- And Update it with the string you copied as highlighted in the above image.
- Navigate to the path of files listed below and comment all other **db:**

ie. db: '**mongodb+srv://username:@xyzmongodb.net/ProjectName**'
Add the project name at the end of the string as mentioned above.

Edit the following listed files using filezilla.

- backend/server/config/env/development.js
- backend/payments/config/env/development.js
- backend/mass_notification/config/env/development.js
- backend/history-earning/config/env/development.js

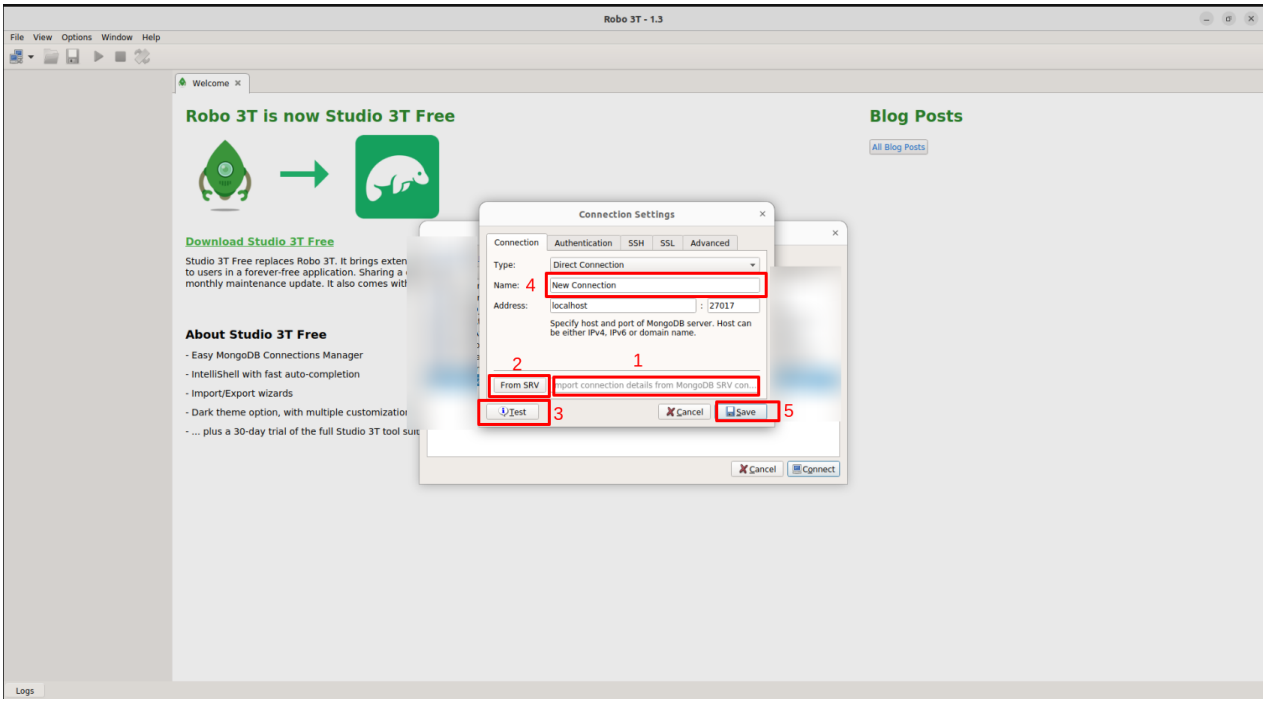
Once the above mentioned files are updated on the server, navigate the **initial_data.js** file as mentioned below.

- cd var/www/html/<ProjectName>/backend/server/settingsdata/
- Run this code ***node initial_data.js***

This will create all the initial files in the DB.

6.6 Connection to the DB

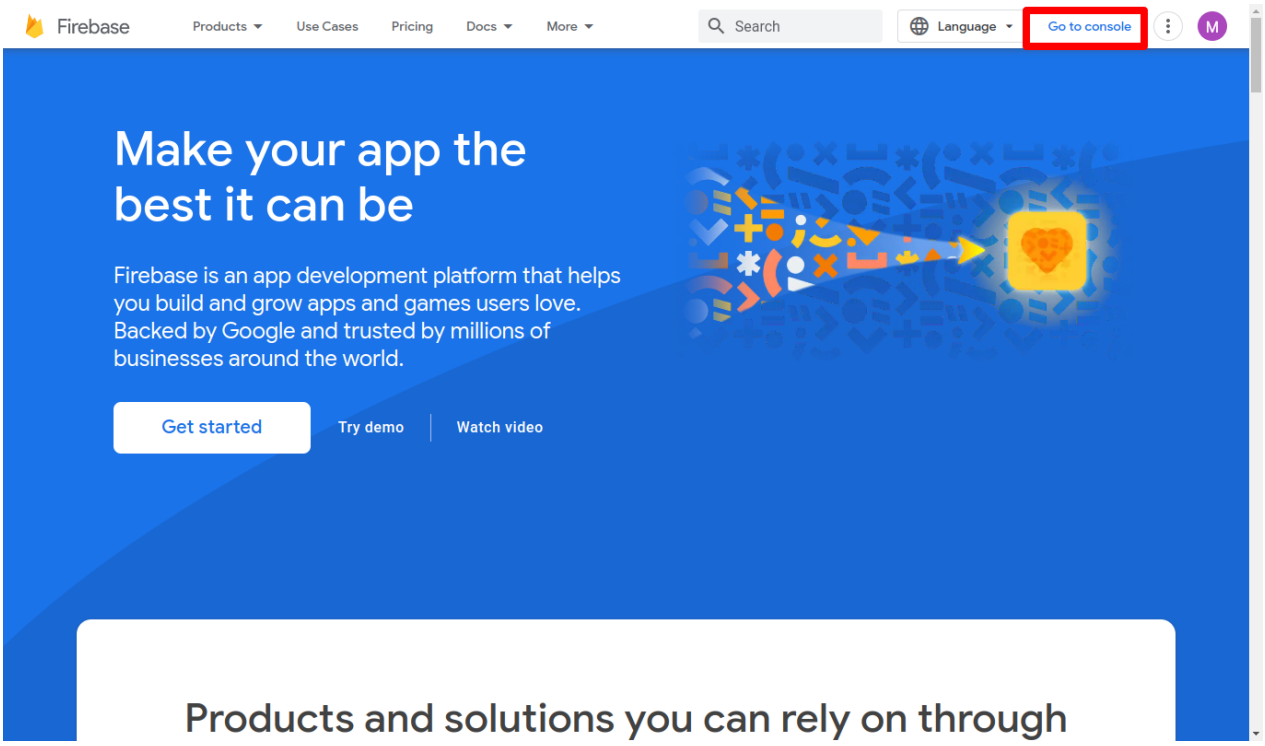
- Use the string which we copied from the mongodb atlas platform and follow the steps to view the DB.
- Below I have mentioned the steps to connect using Robo-3t, to connect using other platforms it is very similar to the below steps.
  Steps:
  - Paste the String
  - Click **From SRV**
  - Click **Test**(It should get success status)
  - Give the Name
  - Click **Save**



7. Firebase Configuration

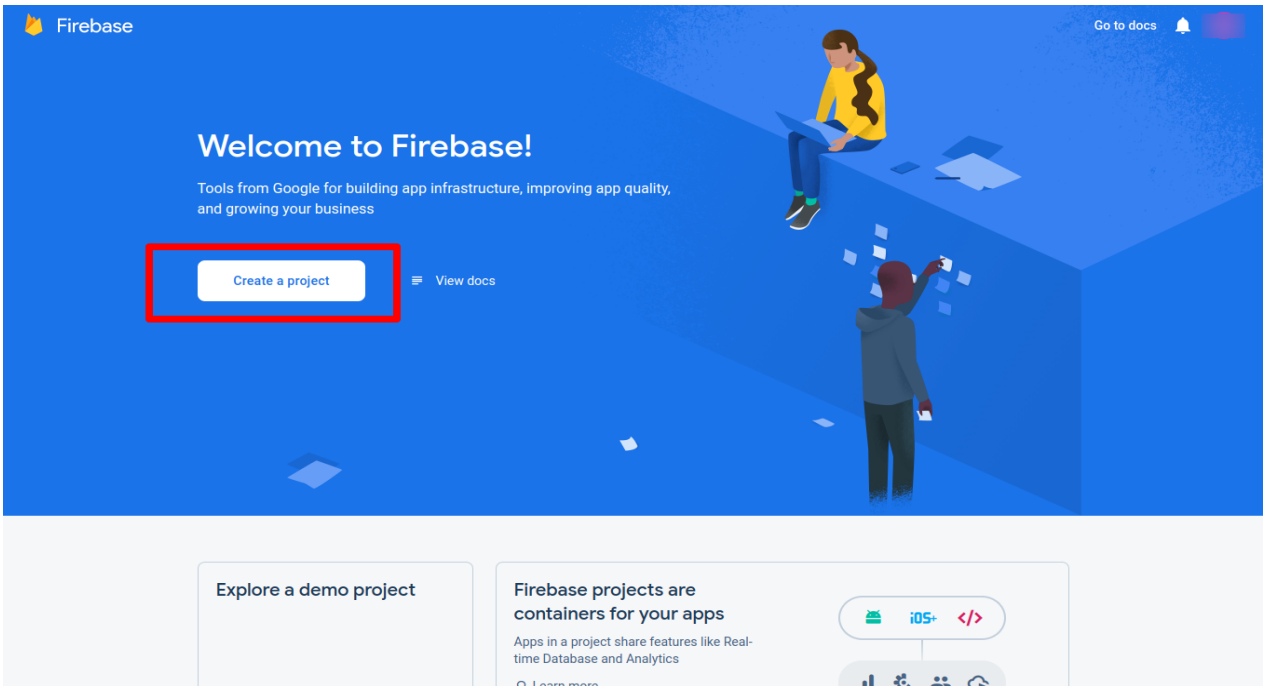
→ Open firebase console click me

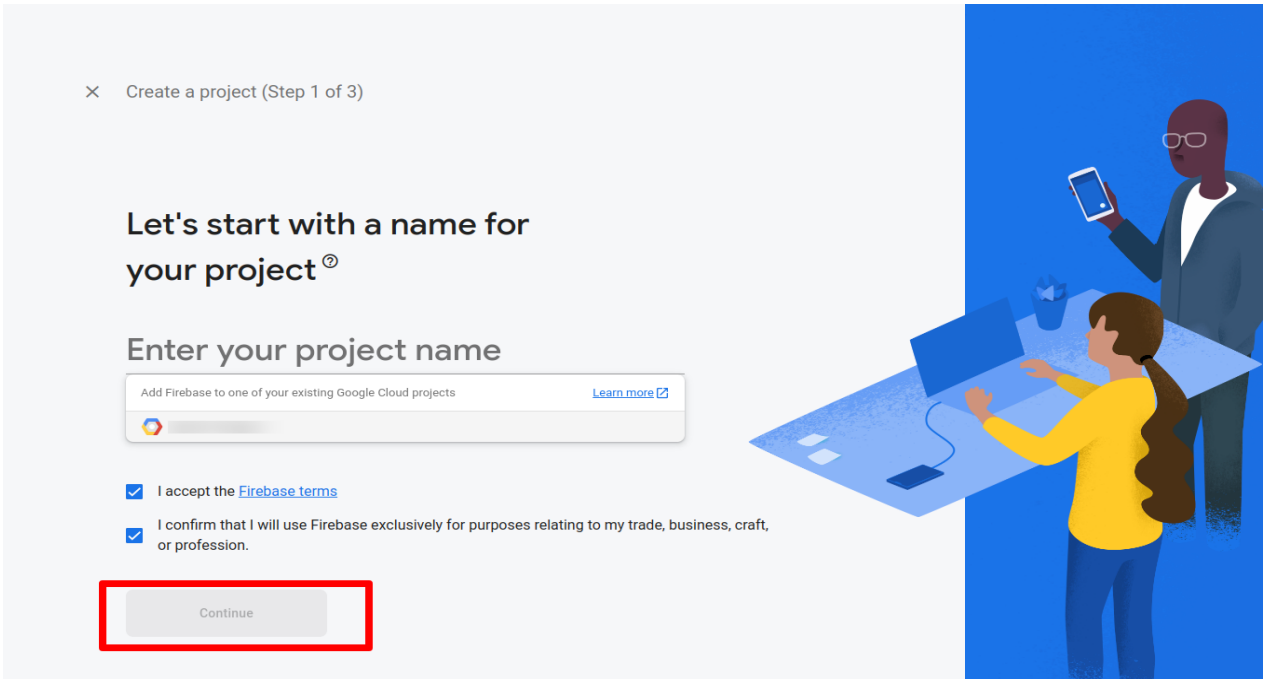
7.1 Go to the console.

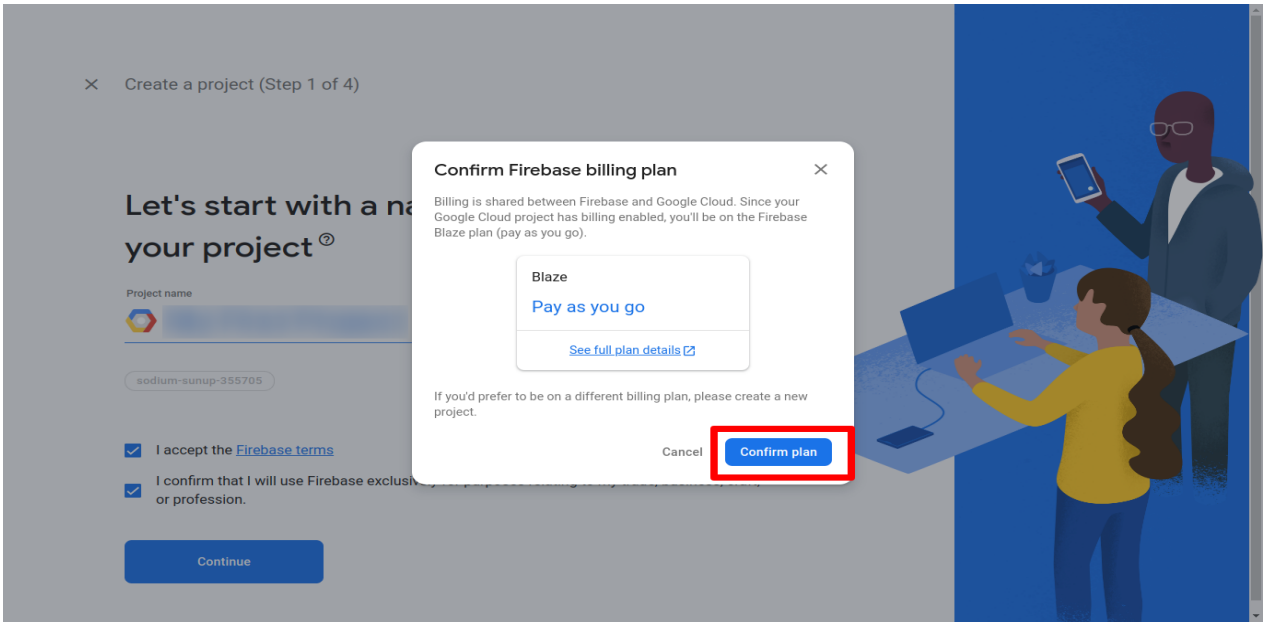


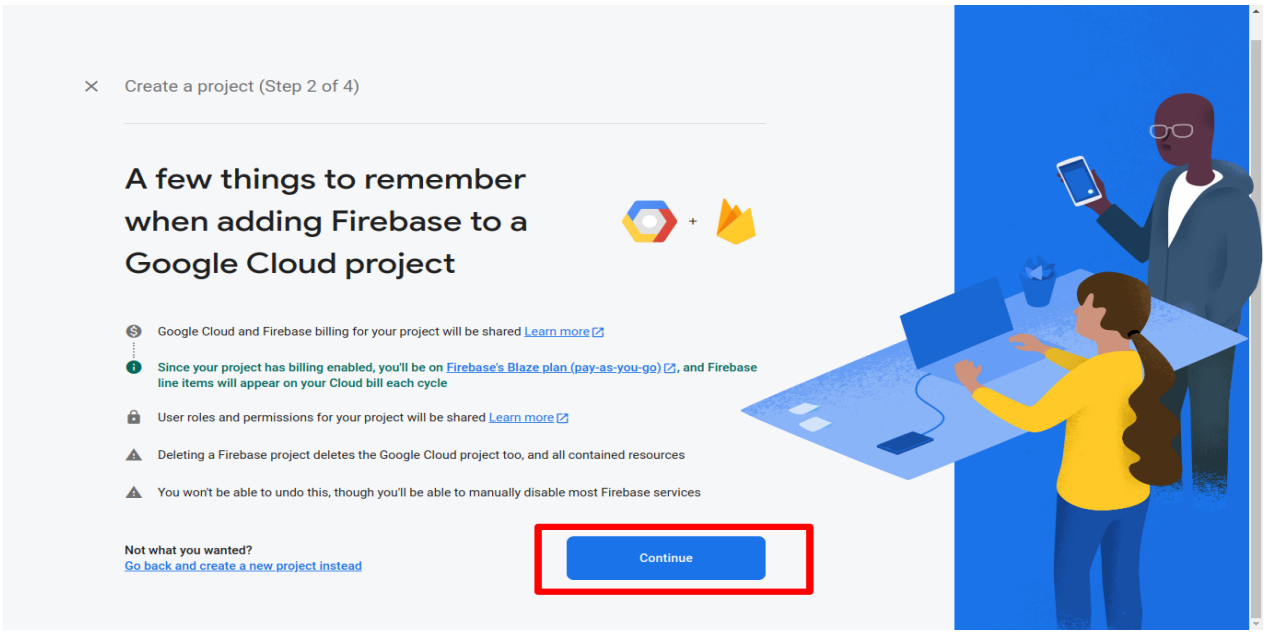
7.2 Create Project

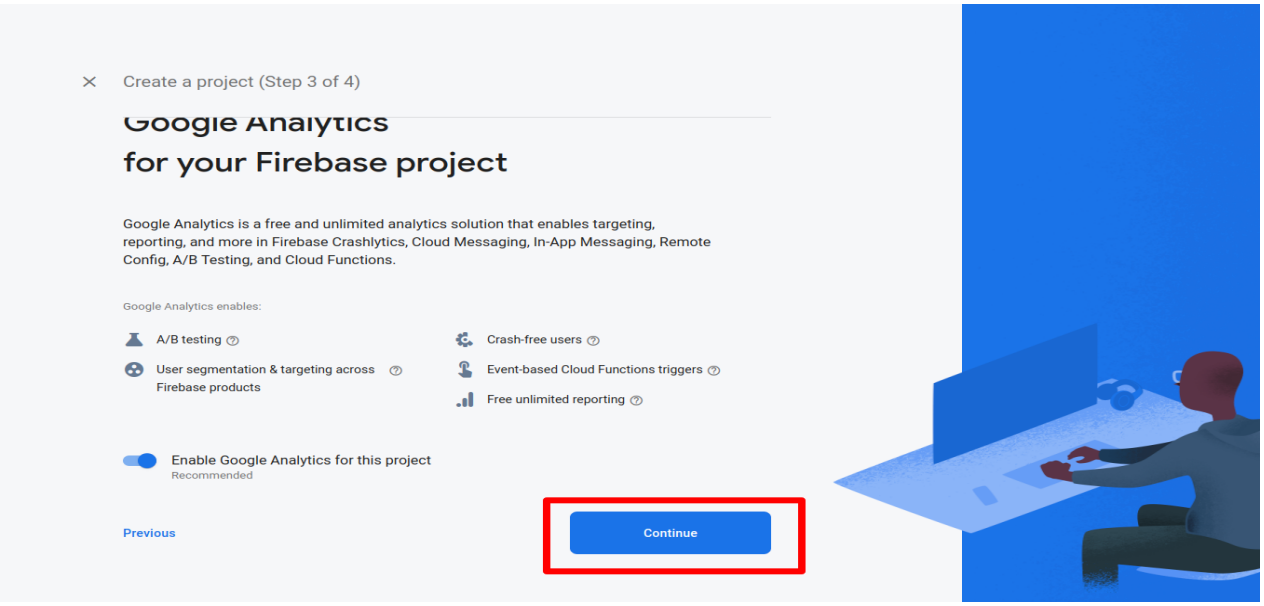
Create a project,and complete that 4 steps with the given image below OR you have to use the existing one if the project is already created.

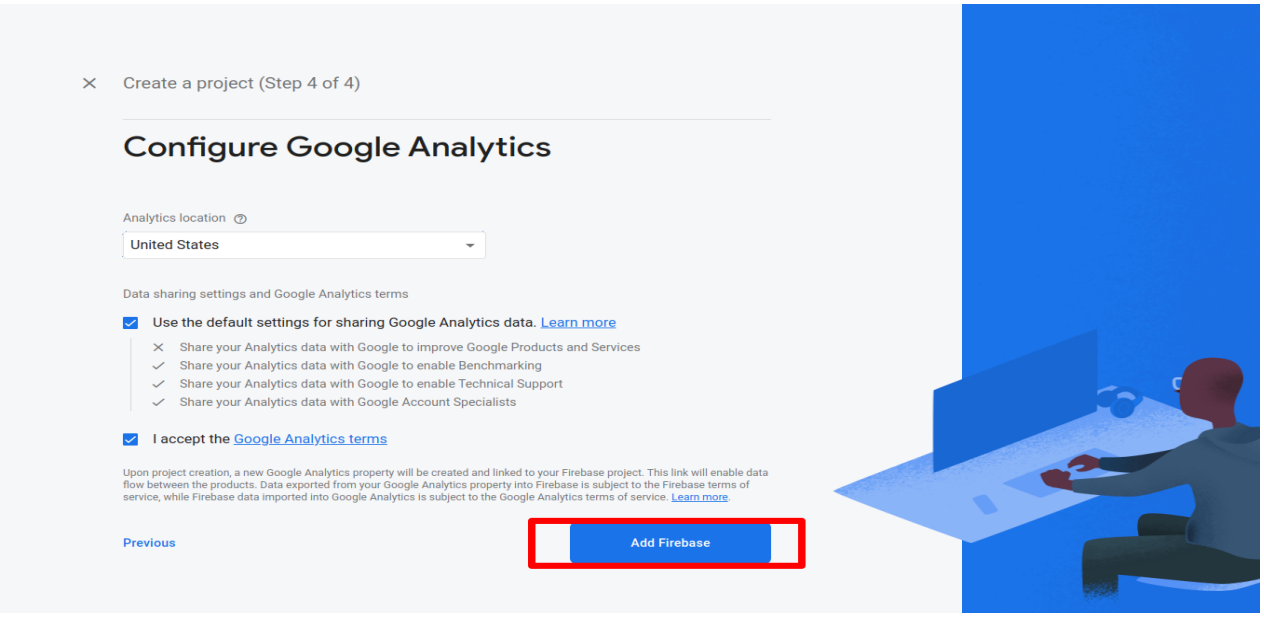

Firebase
Go to docs
Welcome to Firebase!
Tools from Google for building app infrastructure, improving app quality, and growing your business
Create a project
View docs
Explore a demo project
Firebase projects are containers for your apps
Apps in a project share features like Real-time Database and Analytics
iOS+


Create a project (Step 1 of 3)
Let's start with a name for your project
Enter your project name
Add Firebase to one of your existing Google Cloud projects
Learn more
I accept the Firebase terms
I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.
Continue

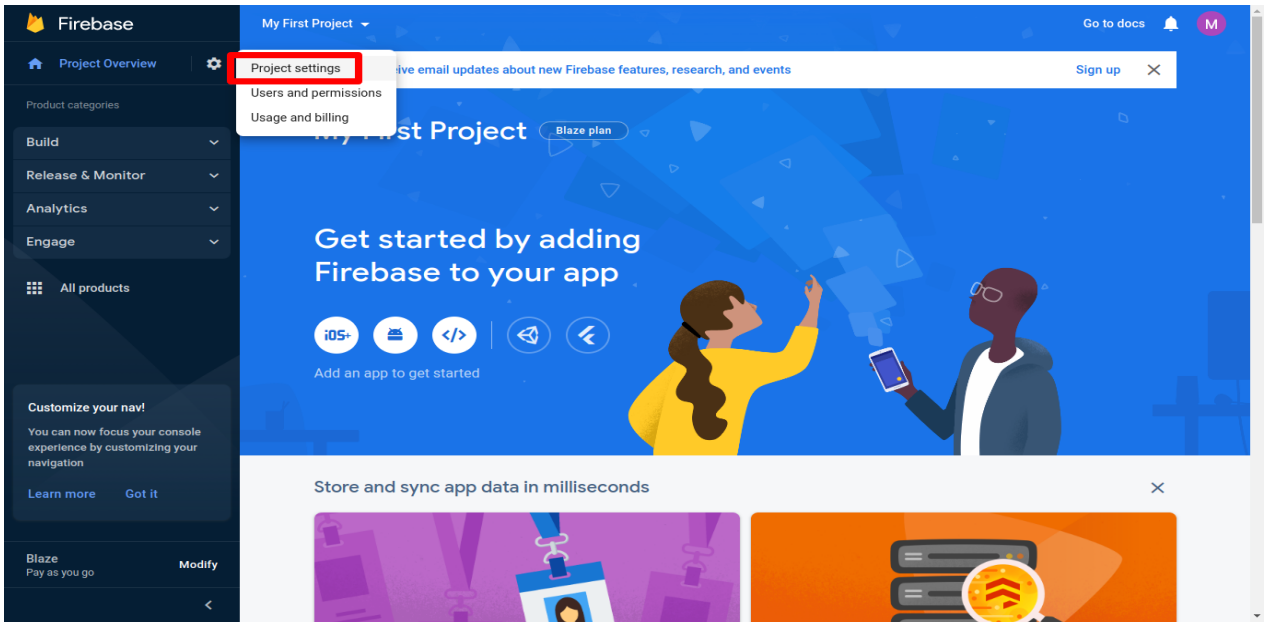

Create a project (Step 1 of 4)
Let's start with a na
your project
Project name
sodium-sunup-355705
I accept the Firebase terms
I confirm that I will use Firebase exclusi
or profession.
Continue
Confirm Firebase billing plan
Billing is shared between Firebase and Google Cloud. Since your Google Cloud project has billing enabled, you'll be on the Firebase Blaze plan (pay as you go).
Blaze
Pay as you go
See full plan details
If you'd prefer to be on a different billing plan, please create a new project.
Cancel
Confirm plan

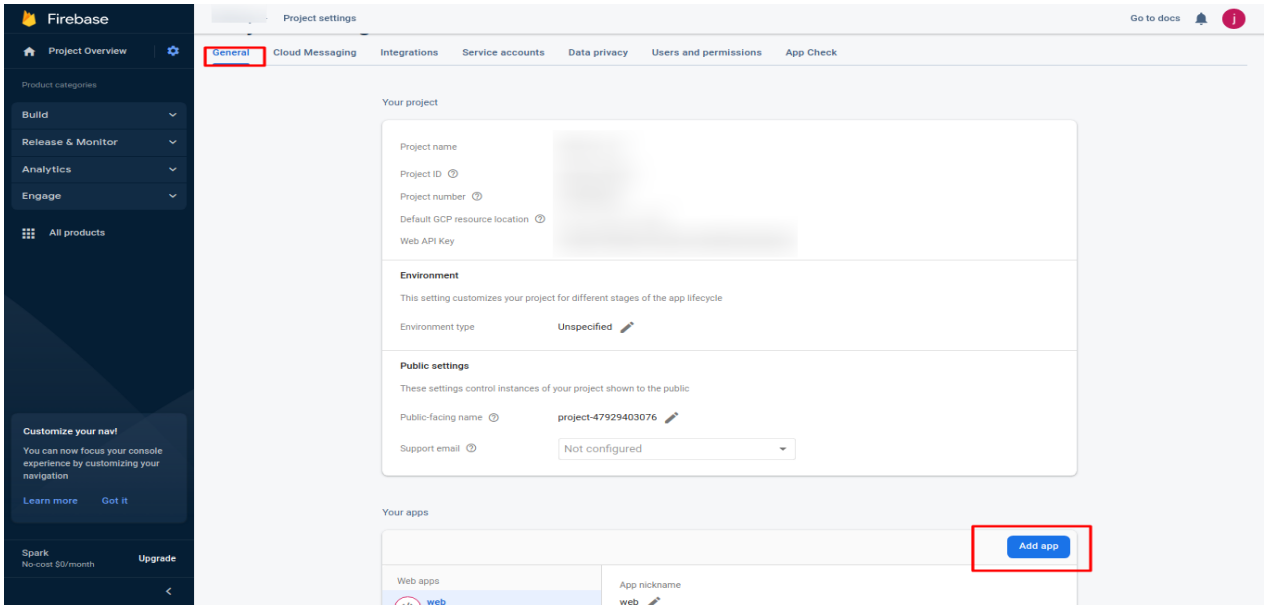

Create a project (Step 2 of 4)
A few things to remember when adding Firebase to a Google Cloud project
Google Cloud and Firebase billing for your project will be shared Learn more
Since your project has billing enabled, you'll be on Firebase's Blaze plan (pay-as-you-go), and Firebase line items will appear on your Cloud bill each cycle
User roles and permissions for your project will be shared Learn more
Deleting a Firebase project deletes the Google Cloud project too, and all contained resources
You won't be able to undo this, though you'll be able to manually disable most Firebase services
Not what you wanted?
Go back and create a new project instead
Continue

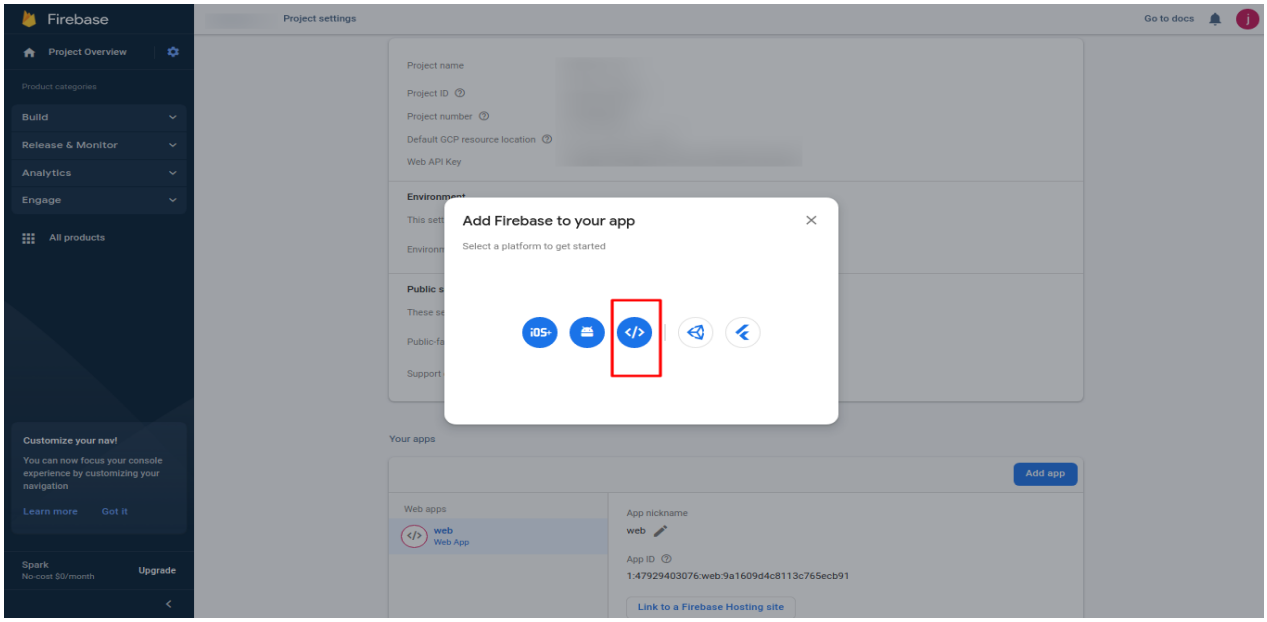

Create a project (Step 3 of 4)
for your Firebase project
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.
Google Analytics enables:
A/B testing
User segmentation & targeting across Firebase products
Crash-free users
Event-based Cloud Functions triggers
Free unlimited reporting
Enable Google Analytics for this project
Recommended
Previous
Continue


Create a project (Step 4 of 4)
Configure Google Analytics
Analytics location
United States
Data sharing settings and Google Analytics terms
Use the default settings for sharing Google Analytics data. Learn more
Share your Analytics data with Google to improve Google Products and Services
Share your Analytics data with Google to enable Benchmarking
Share your Analytics data with Google to enable Technical Support
Share your Analytics data with Google Account Specialists
I accept the Google Analytics terms
Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. Learn more
Previous
Add Firebase

7.3 Create a web app in firebase General project settings.

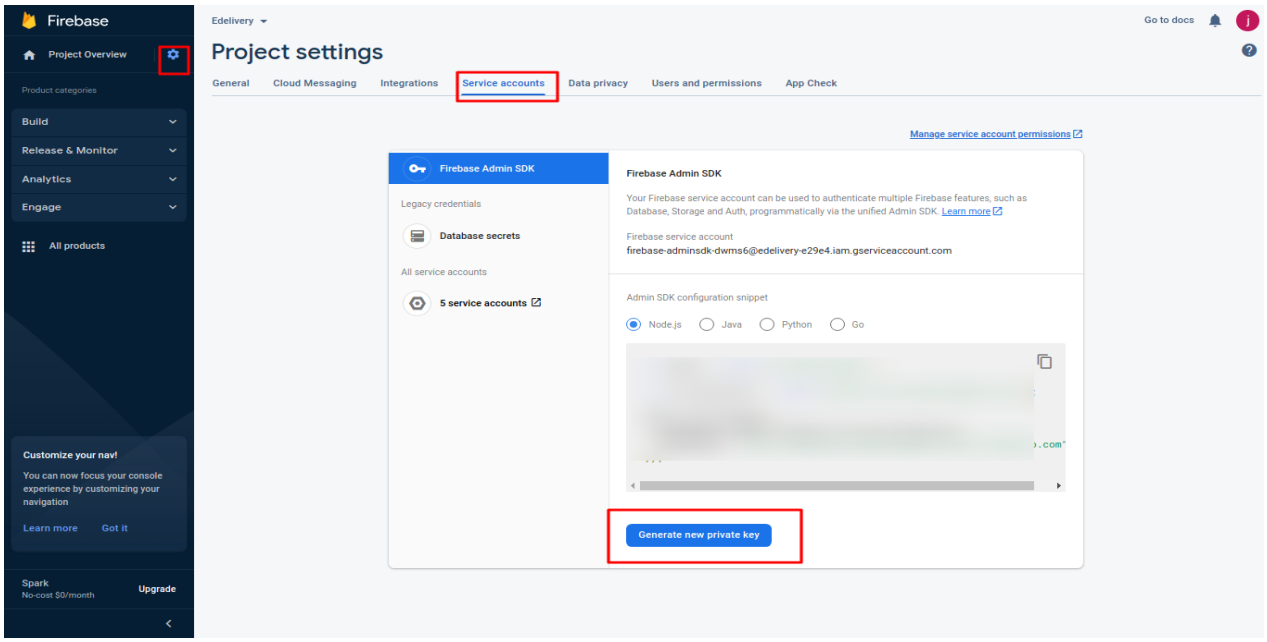




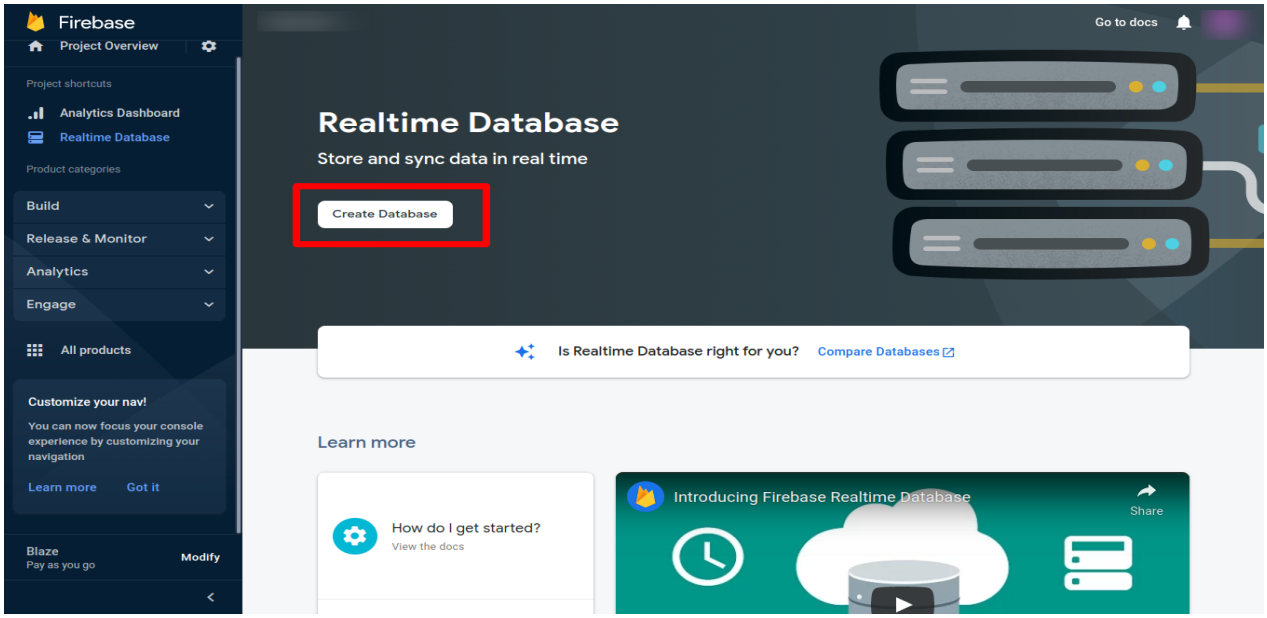


7.4 Create a Service account

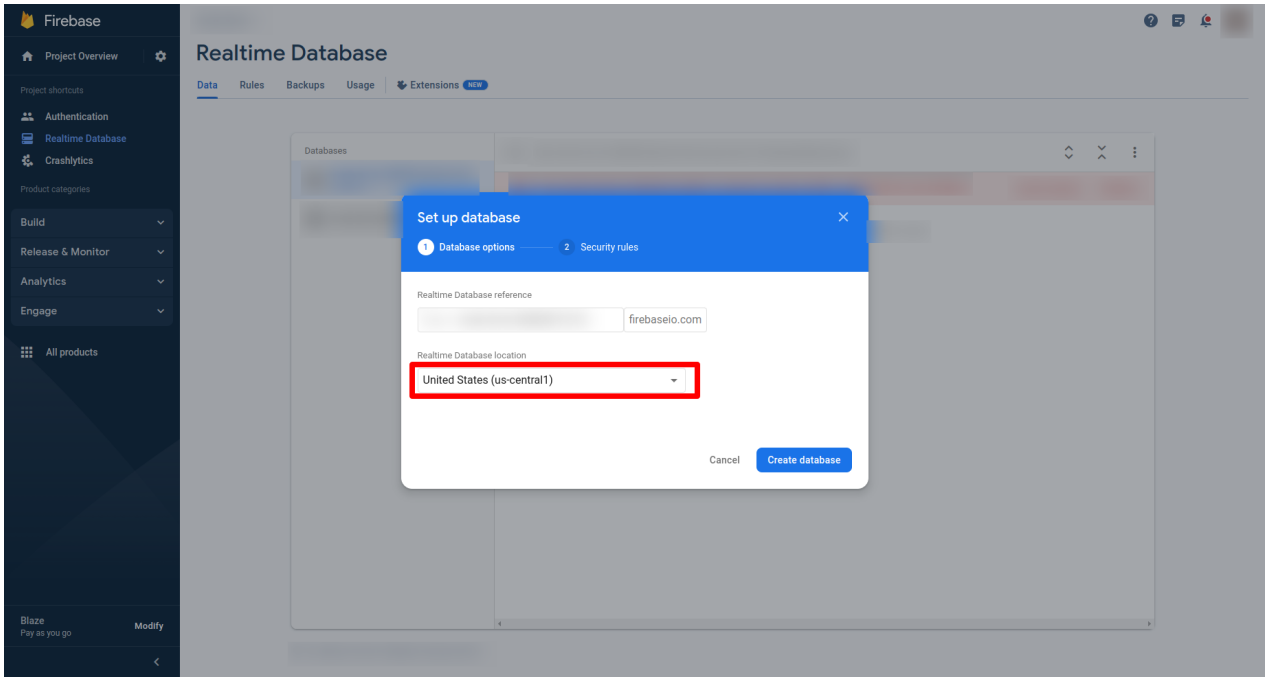
Create a Service account and download, generate a new private key and add in the setting database.

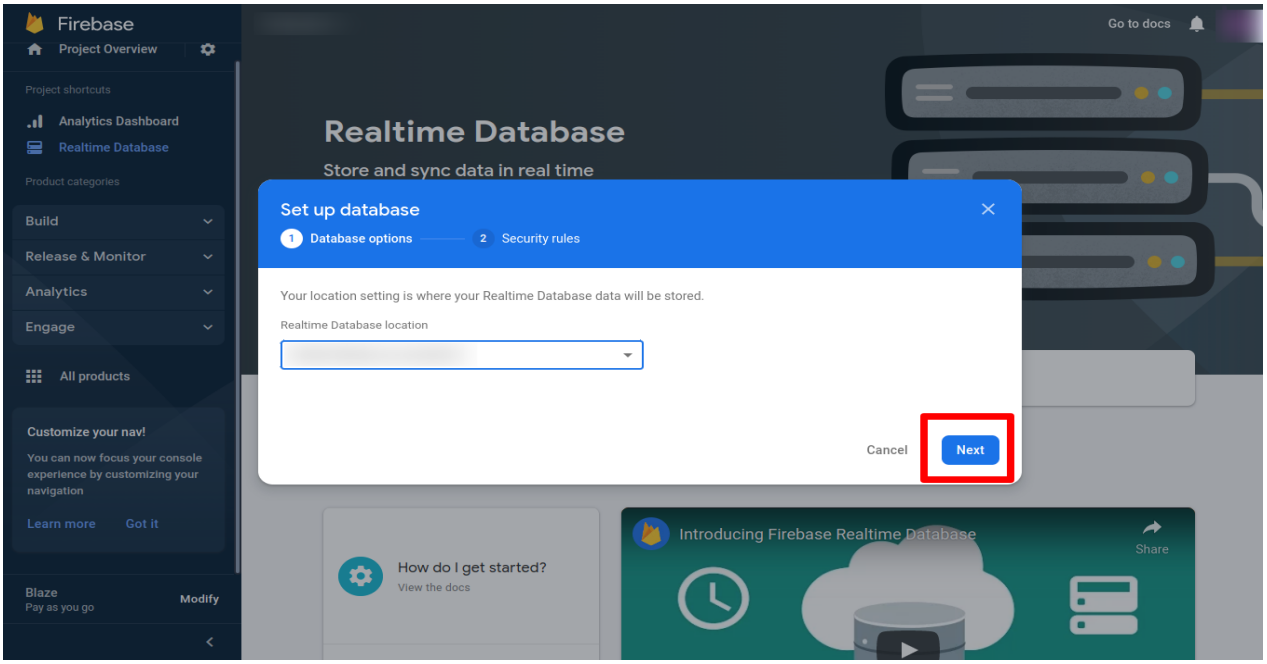


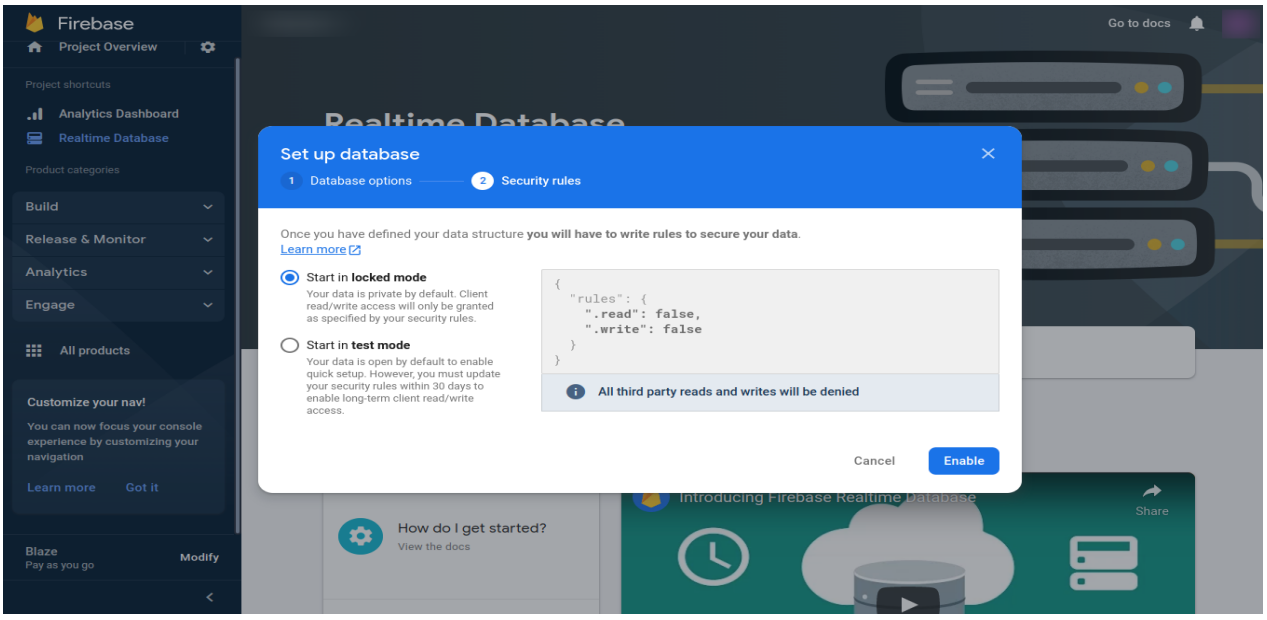
7.5 Create a Realtime Database.



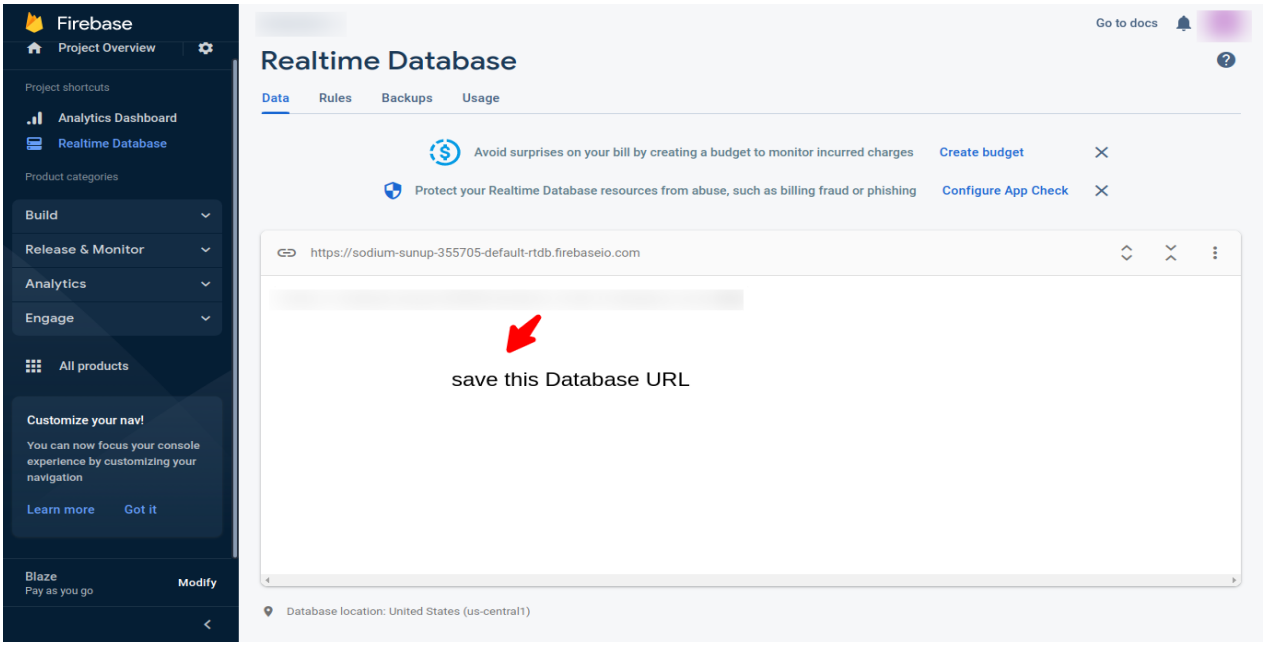
NOTE: Always select the US region like this.







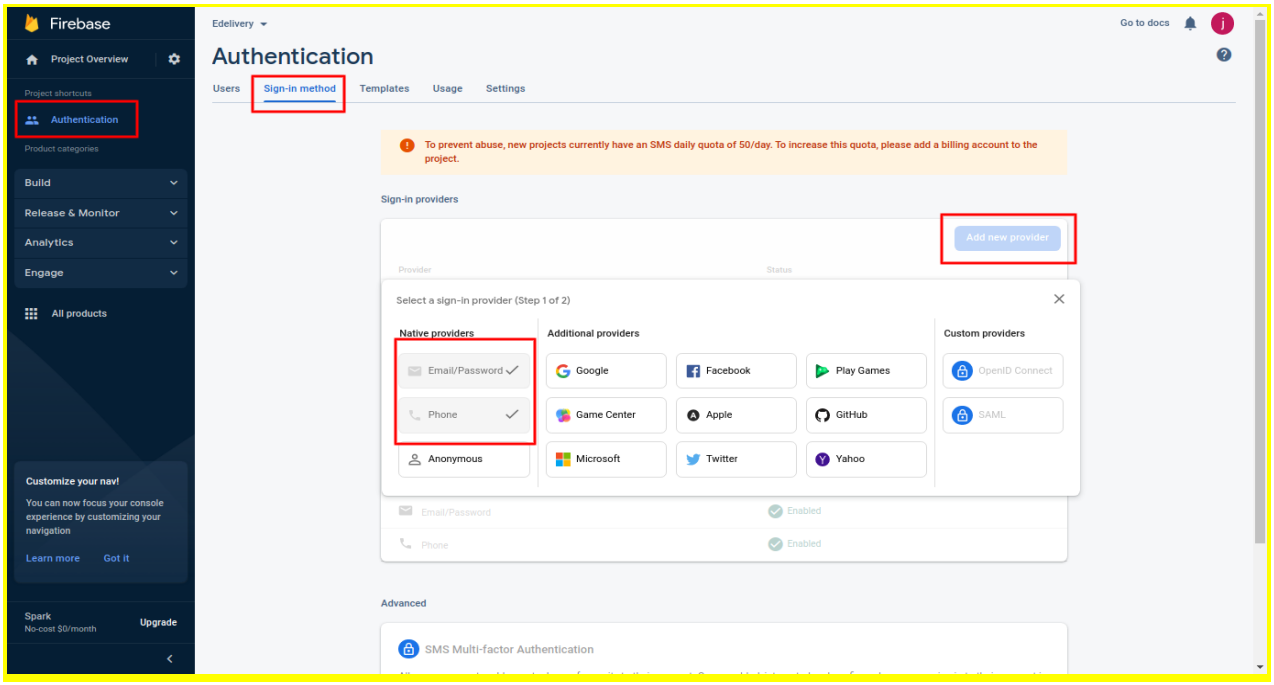
IMPORTANT NOTE : After selecting locked mode(false) we have to do manually rules to true and publish.



*Note -> Save this url into the database in settings collection.

7.6 Add Auth Sign-In Methods.

Enable two Sign-In methods, **Phone and Email / Password** as highlighted below.

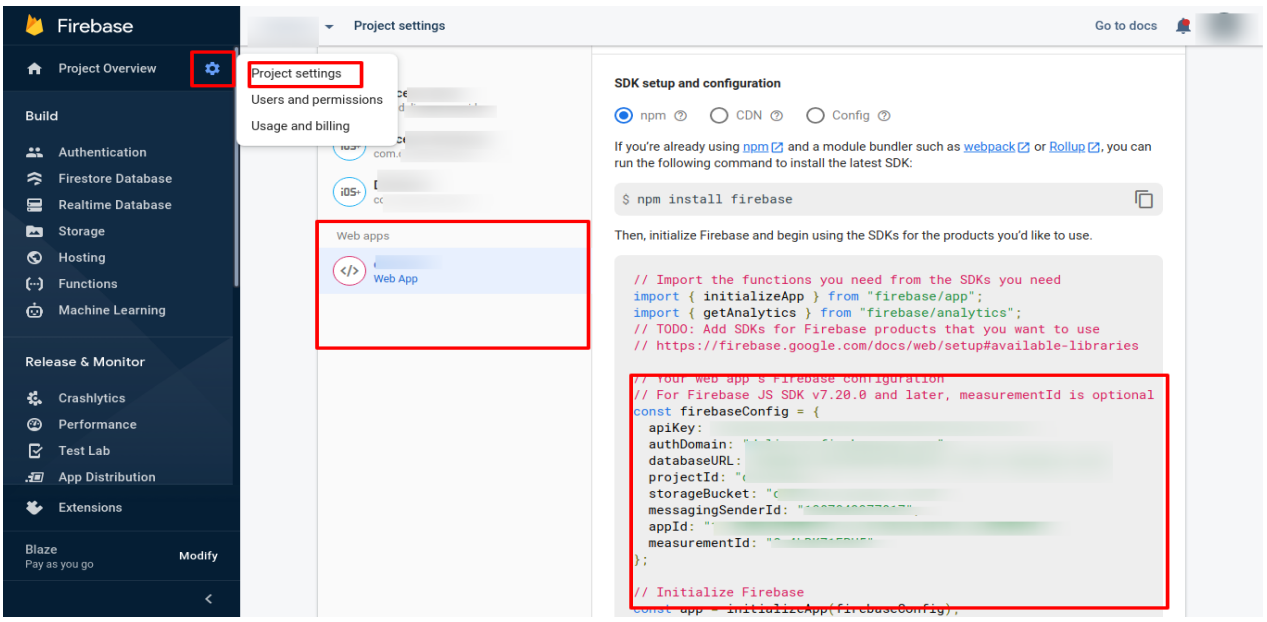


7.7 Add this firebase config

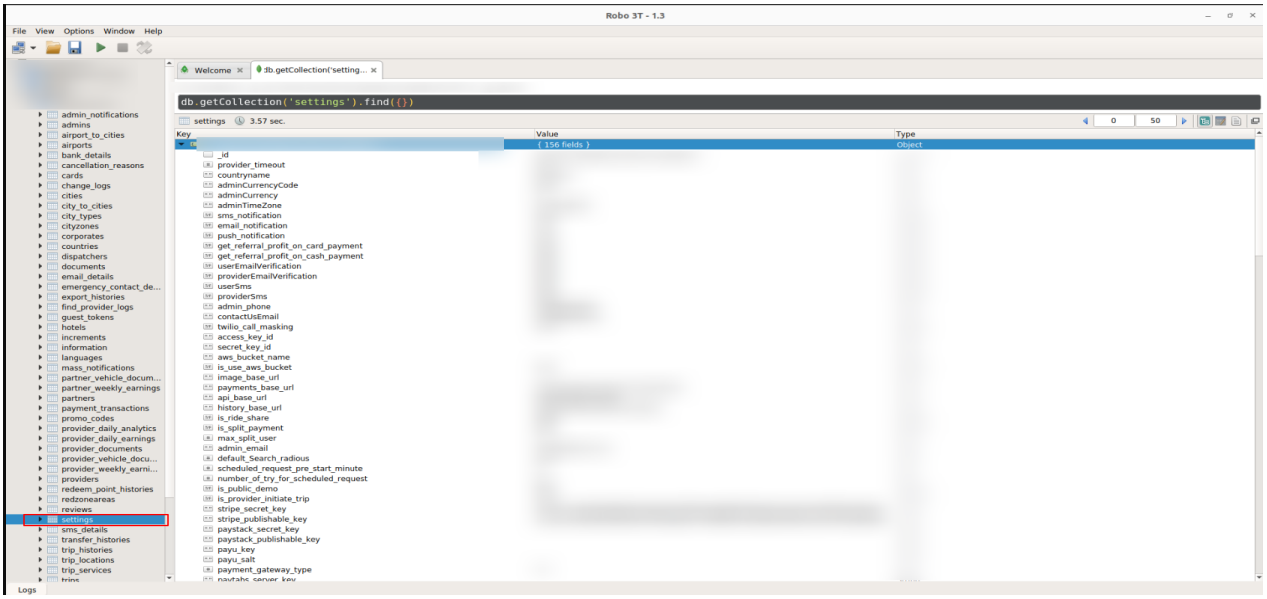
Replace this firebase config(object) in your project in all panels listed here and setting collection in the database.

src/environments/environment.ts

src/environments/environment.prod.ts



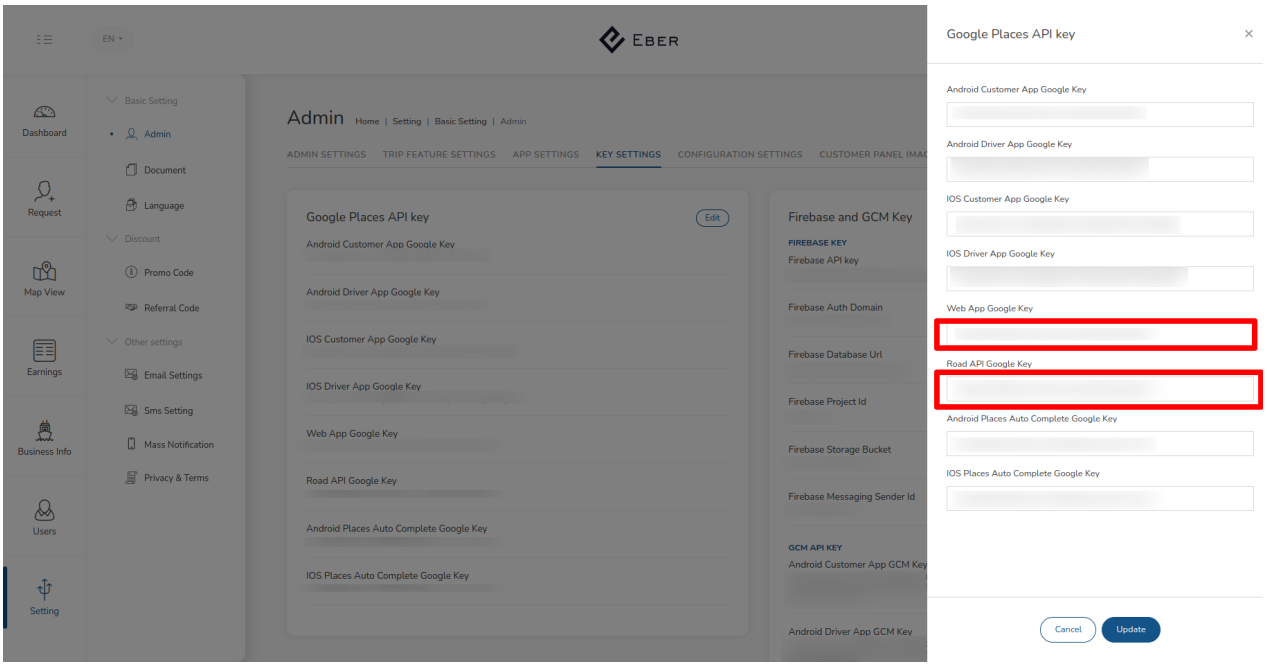
Update setting collection



8. Basic setup on Admin panel

Basic setup on admin panel for making application fully functional.
Now our applications are live so now we need to do some configurations to make our application fully functional. We need to set Google API keys to make maps work.
As shown in the image head to app settings and press the edit button to set your to google API keys.

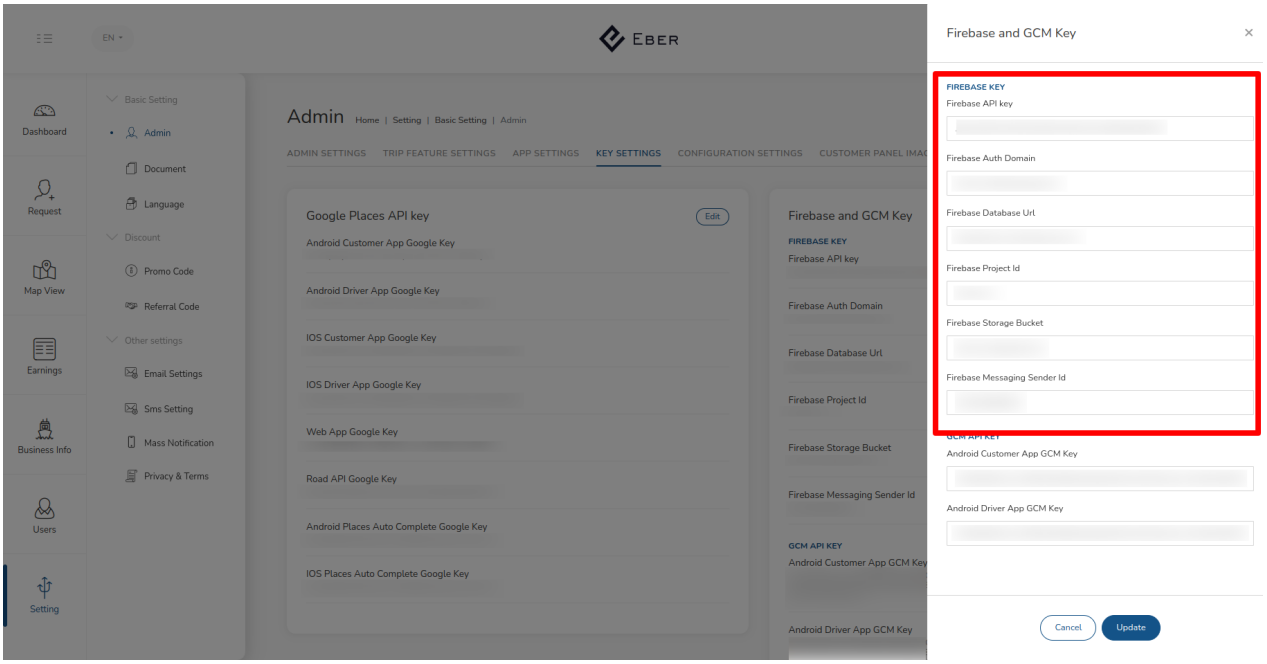
8.1 Set up Google api keys



- In the admin panel, navigate to settings, admin, and then key settings.
- Press the edit button located at the top-right corner in User App settings
- Place your google API keys in appropriate fields.

8.2 Set up Notification Firebase Keys

- In the admin panel, navigate to **settings**->**admin** -> **Key settings**.



- Now enter your firebase keys in appropriate fields.

9. Chat Push Firebase

9.1 Open putty with Backend instance of your project

Start using the following command:

sudo npm install -g firebase-tools --unsafe-perm
cd /var/www/html OR cd /root
sudo mkdir functions-firebase
sudo firebase login --no-localhost

=>you will get a link
open that link to your browser and log in to your Gmail then you will get
Success in terminal

cd functions-firebase
sudo firebase init

select cloud function
use an existing project
select project
select javascript
Deny ESlint
install dependencies with npm Y

cd functions
then edit your index.js file (this index file you have to make as your DB structure i am giving here my demo file)

sudo nano index.js

Add(copy and paste) the following code in index.js file.

```
// Code Start
    const functions = require('firebase-functions');
    const admin = require('firebase-admin');
    admin.initializeApp(functions.config().firebase);
    exports.sendAdminNotification =
      functions.database.ref('/{Topic}/{UniqueId}/{OrderId}/').onWrite(event => {
        var topic = event.after._path.split('/');
        event.after._data.chat_type = String(event.after._data.chat_type);

        const payload = {
```

```
            notification: {
              'chat_type': String(event.after._data.chat_type),
              'id': event.after._data.id,
              'is_read': String(event.after._data.is_read),
              'body': event.after._data.message,
              'sender_type': String(event.after._data.sender_type),
              'time': event.after._data.time,
              'order_id': topic[1]
            }
          };
            if(!event.after._data.is_read)
            {
                return admin.messaging().sendToTopic(event.after._data.receiver_id,
payload)
                    .then(function(response) {
                        return console.log('Notification sent successfully:',
response);
                    })
                    .catch(function(error) {
                        return console.log('Notification sent failed:', error);
                    });
            }

      });
// Code End
```

cd ..
sudo firebase deploy

10. Google API key

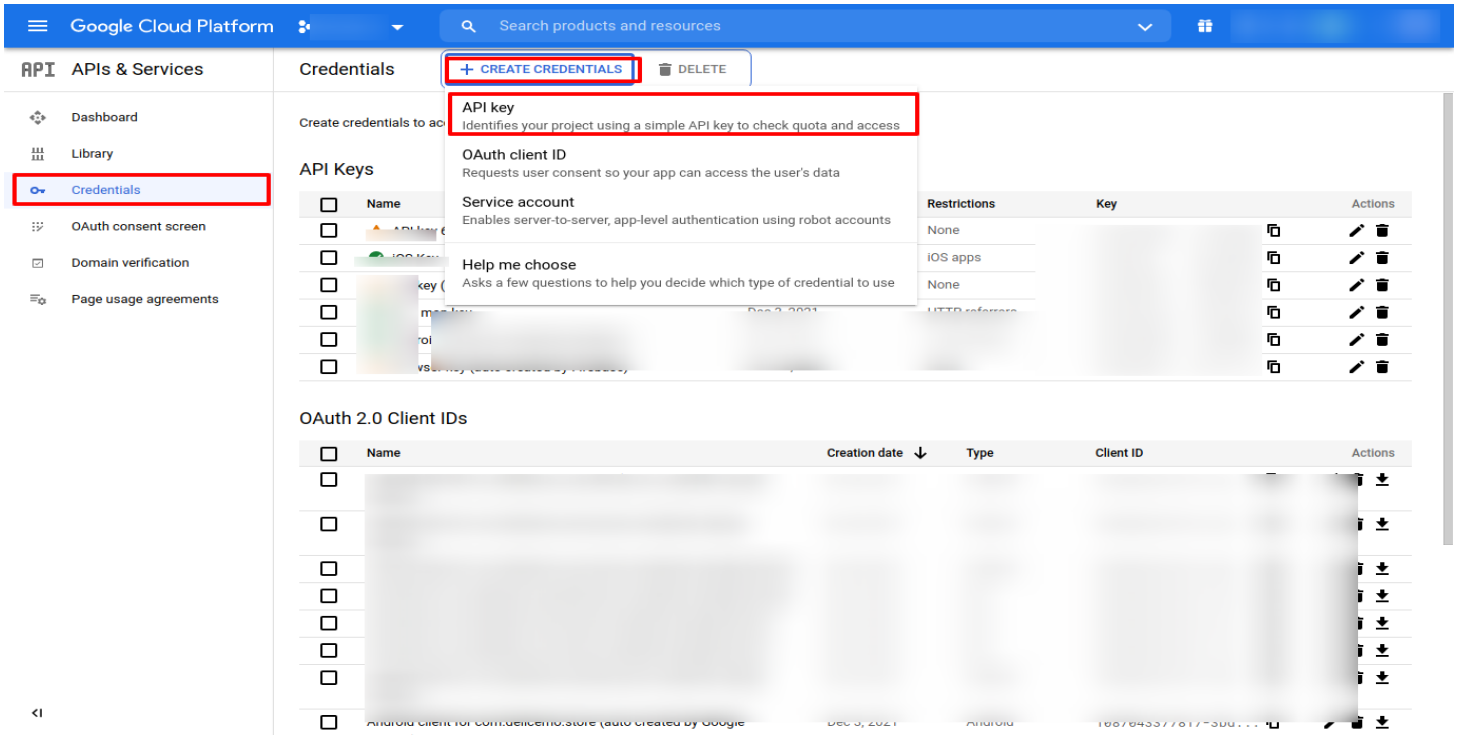
10.1 Open Google cloud console.

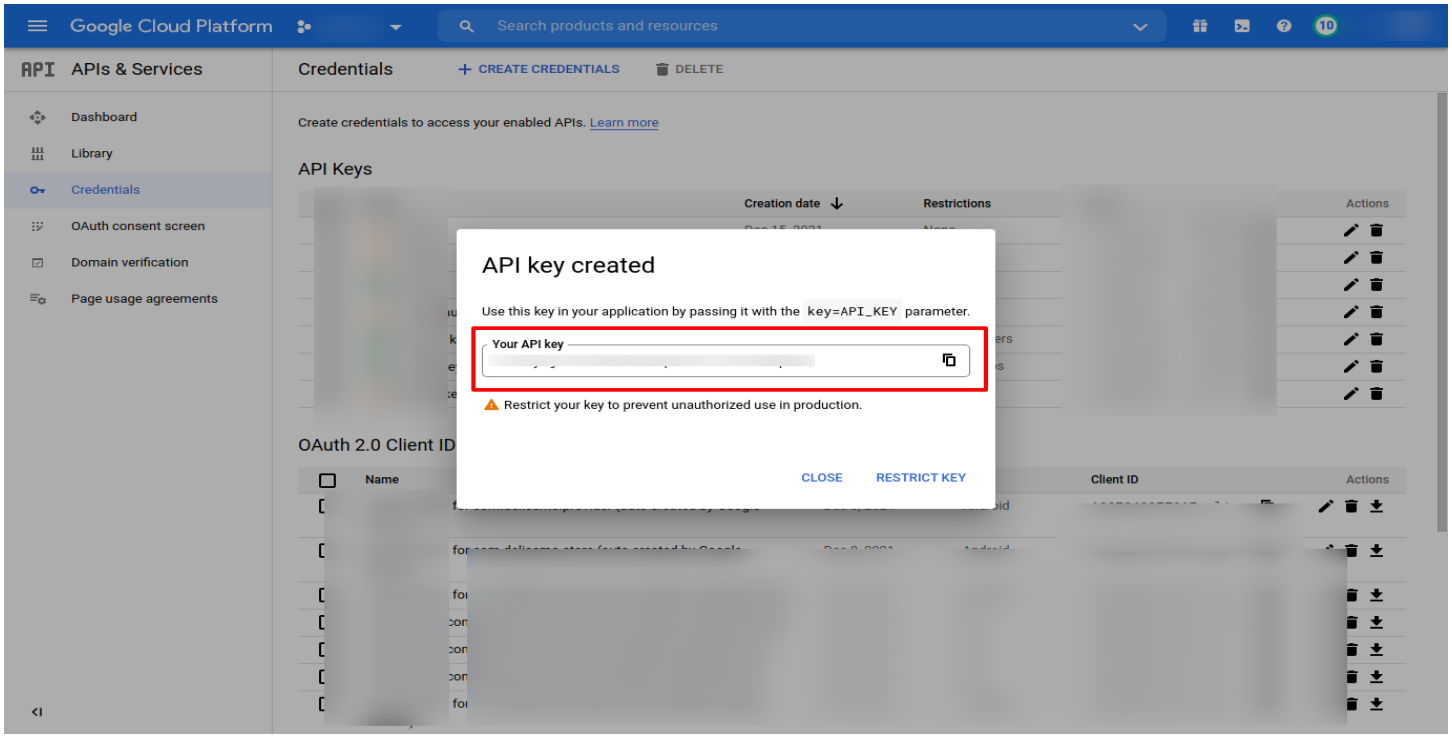
Go to console [Click me]

10.2 Copy that API key

Copy that API key and save it in a text file and use it in Your project

10.3 Set this API Key in the given files.

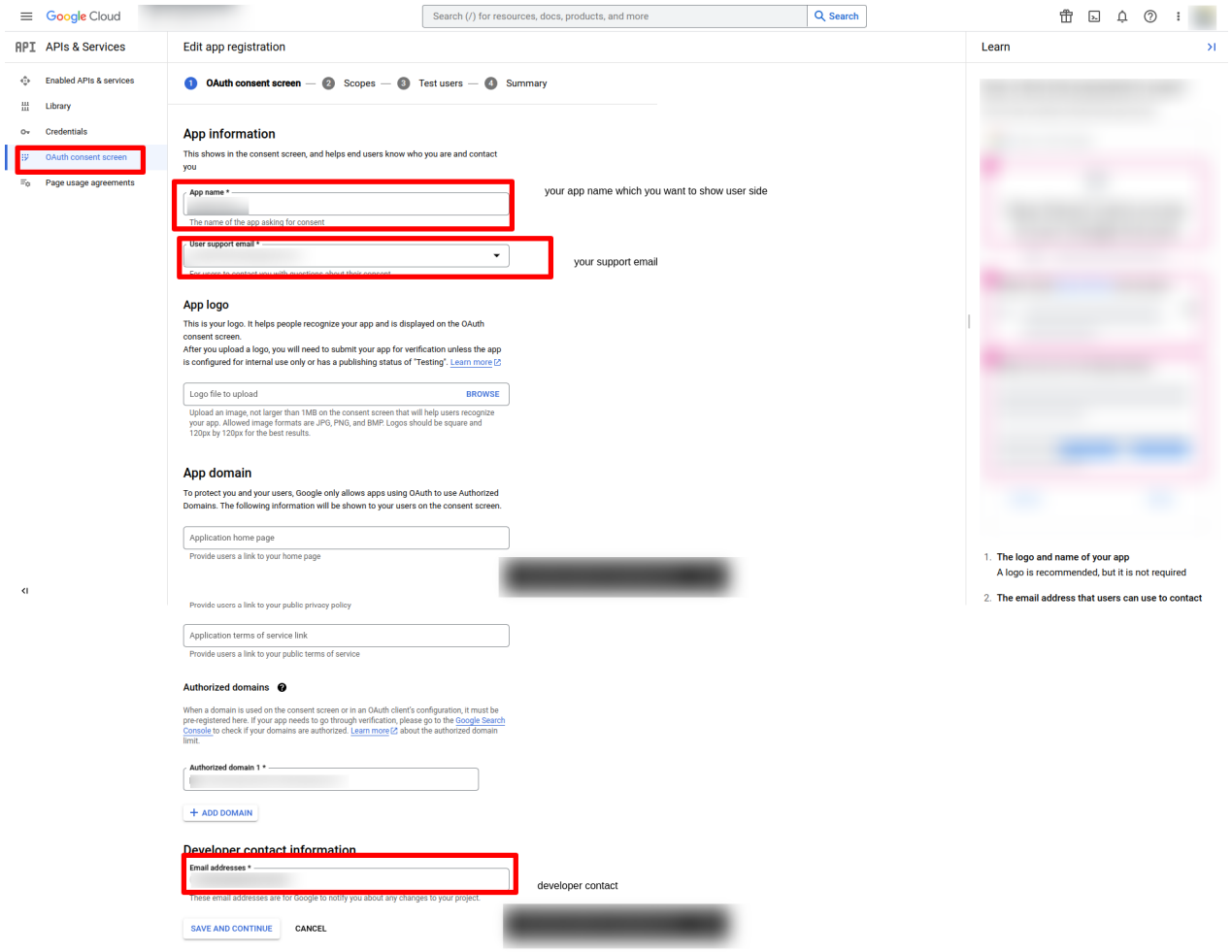




10.4 Activate following Google API Keys

Directions API
Distance Matrix API
Geocoding API
Geolocation API
Maps JavaScript API
Maps SDK for Android
Maps SDK for iOS
Maps Static API
Places API
Roads API

10.5 Configure Auth consent screen if not



2nd step -> press button save and continue

3rd step -> test user save and continue

4th step -> review summary and back to dashboard

10.6 Set up APIs in code

Admin

-> src/environments/environment.ts

-> src/environments/environment.prod.ts

-> src/index.html

Corporate, Dispatcher, Driver, Partner, User, Hotel, Hub.

-> src/environments/environment.ts
-> src/environments/environment.prod.ts

For reference click me.

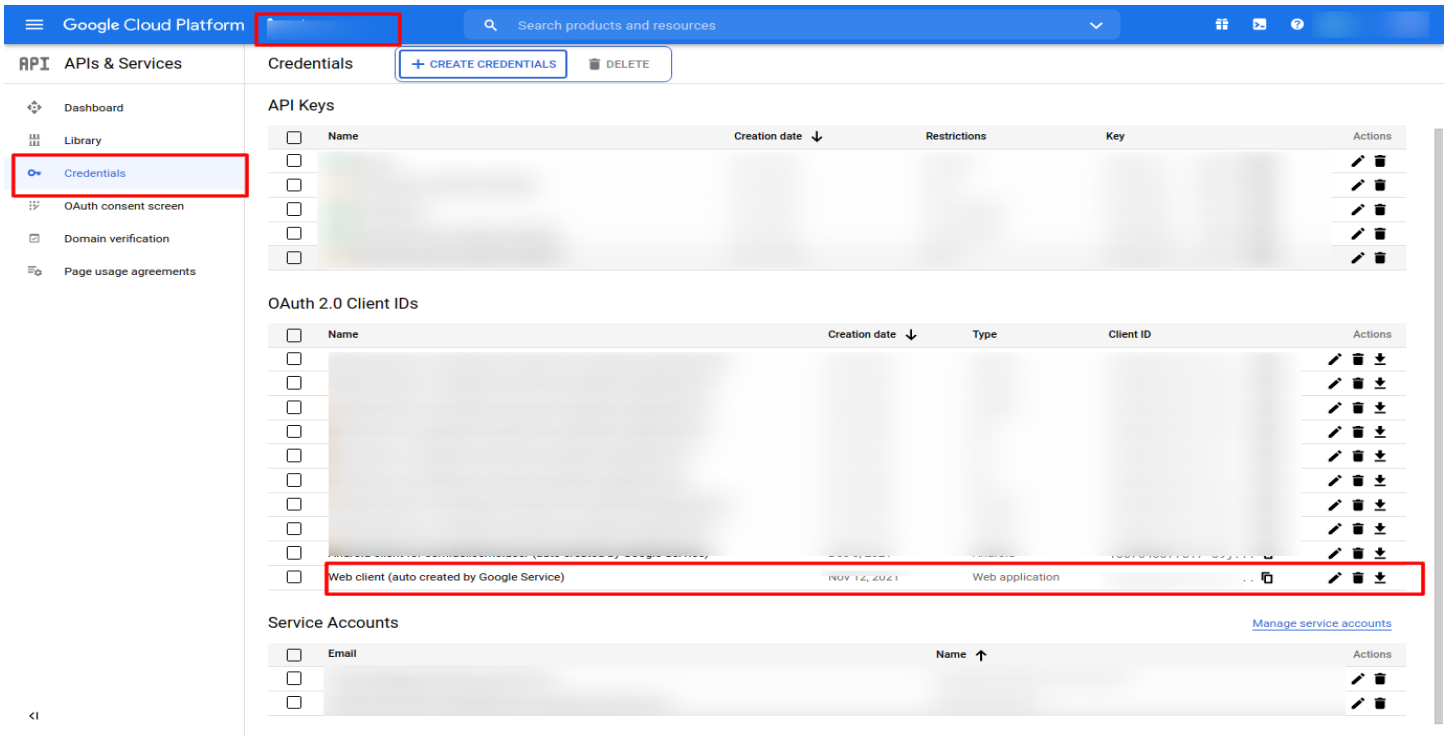
11. Social login using Gmail

11.1 Open google console

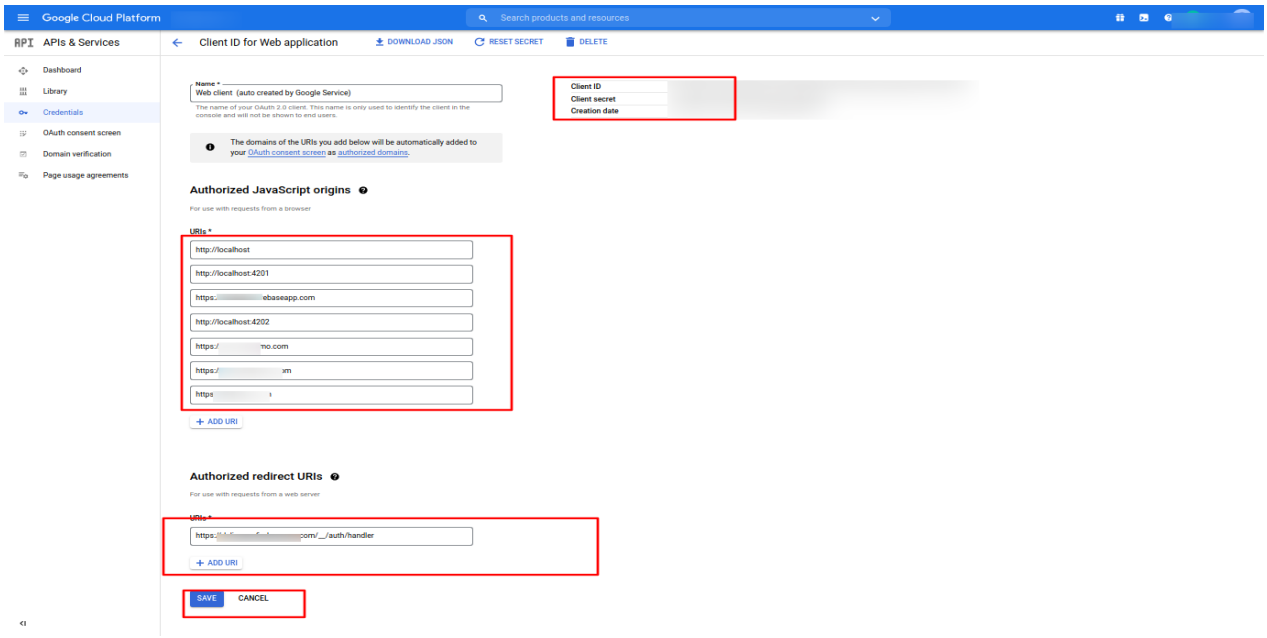
Open google console and select your project. [Click me]

11.2 Select Web client auto service.

* Note -> Set google app id in the project after generating.



11.3 Add your domain name to URI



11.4 And replace **client_id** in your project Database.

For hint search this string '**`apps.googleusercontent.com`**'

- Add client_id in driver panel (driver-panel/src/app/views/app/auth/auth.component.ts)
- User panel (user-panel/src/app/containers/pages/auth-modal/auth-modal.comp OBJ OBJ OBJ OBJ OBJ OBJ OBJ onent.ts)

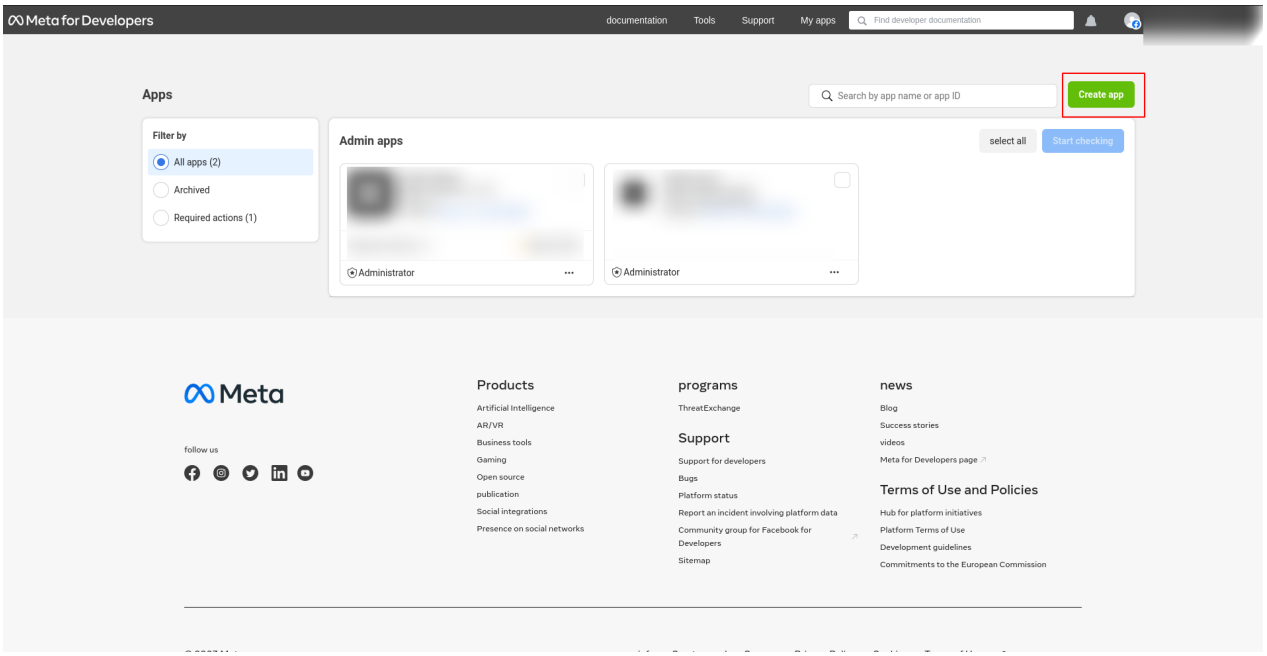
12. Facebook Login.

12.1 Open Facebook developer console.

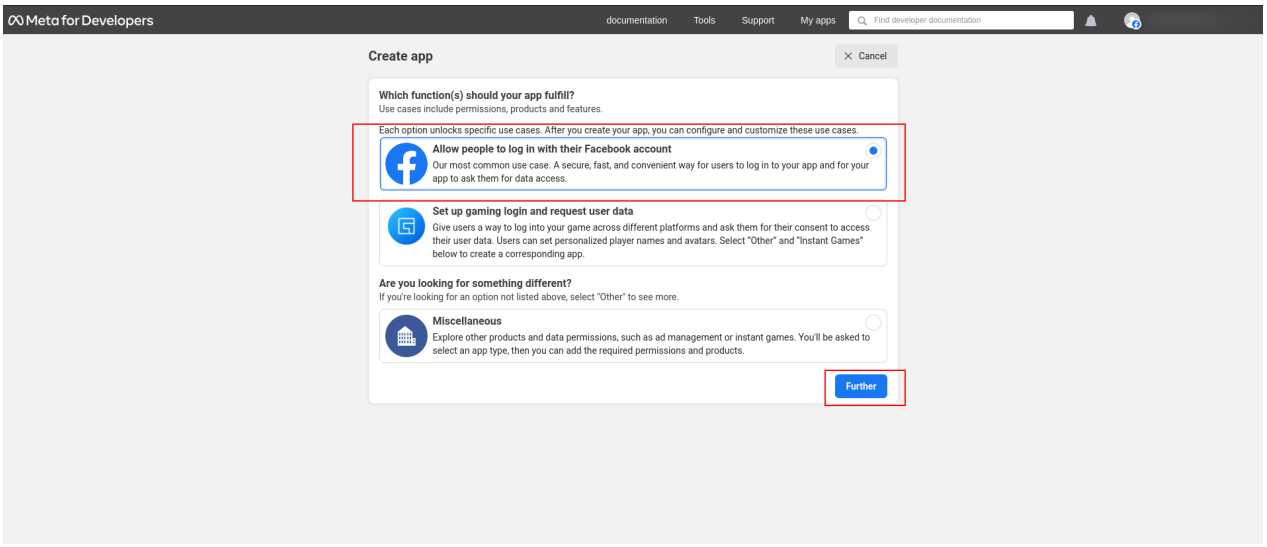
Account Register or Login

12.2 Create app

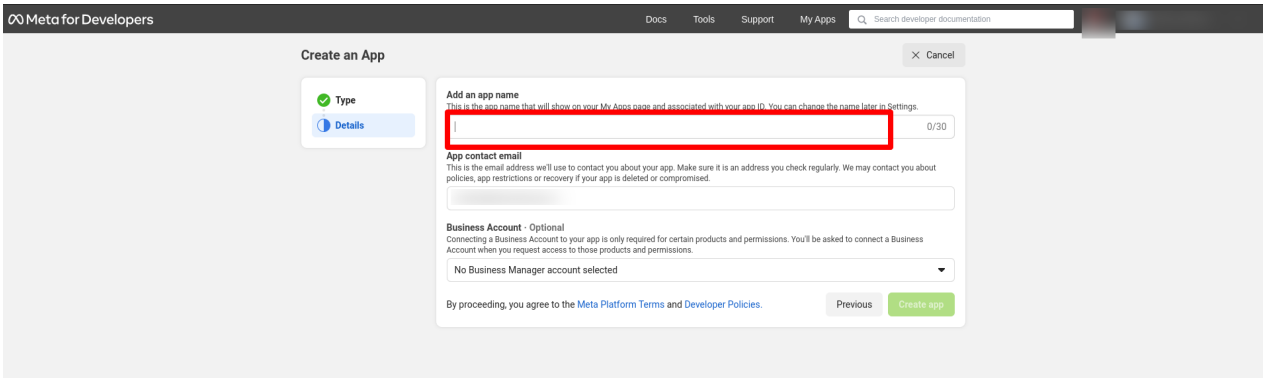
You will have to create a total 2 apps for your project.



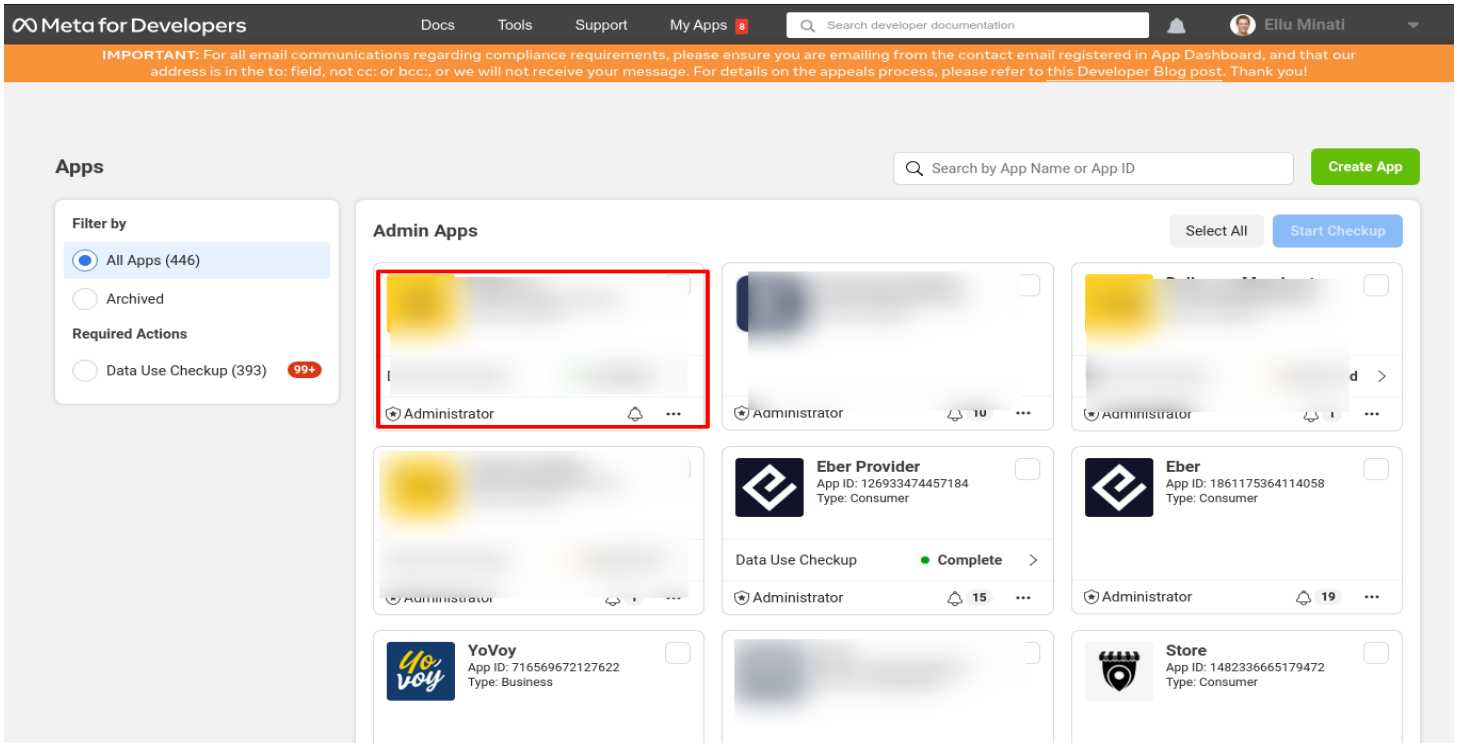
1 -> select accordingly



2 -> your app name

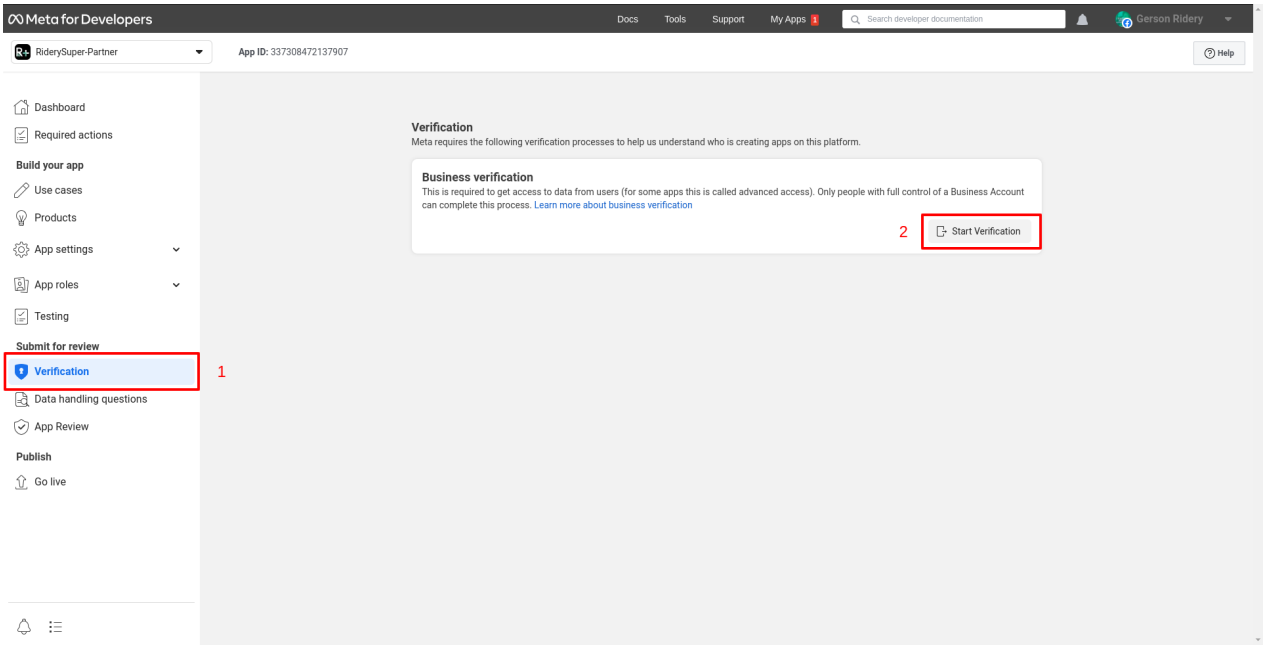


12.3 Select your project app.



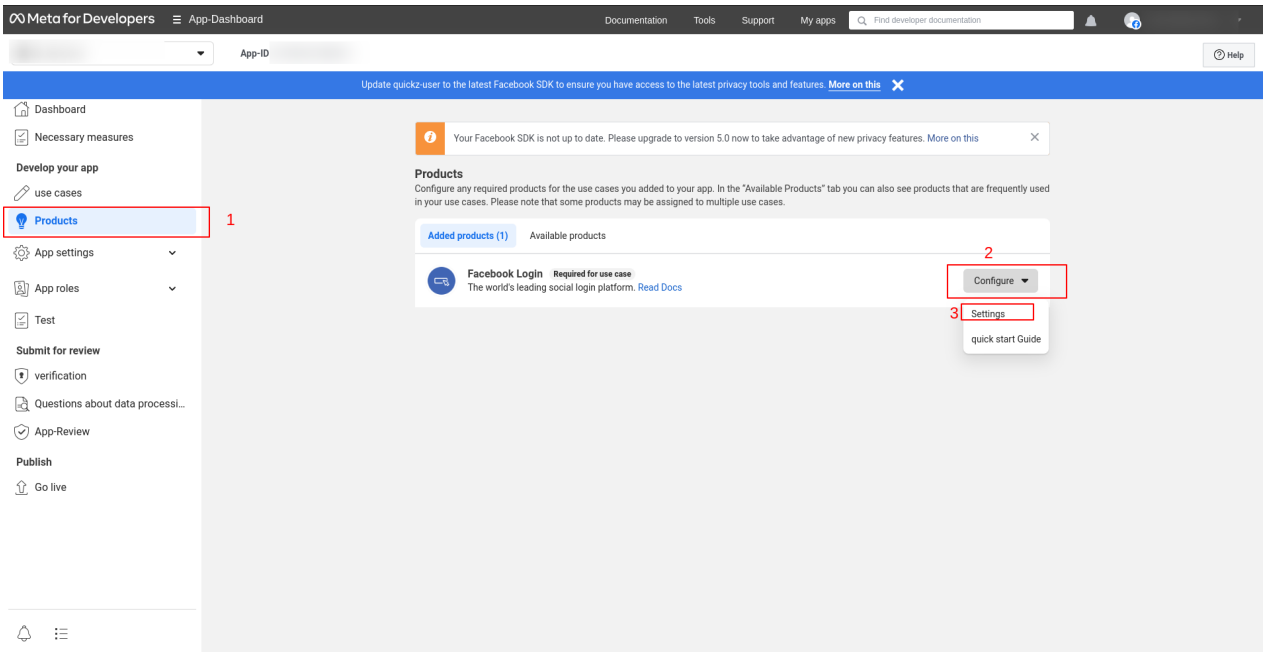
12.4 Click on Verification

Click on verification -> Start verification and fill all the basic required details and complete the process before moving ahead.



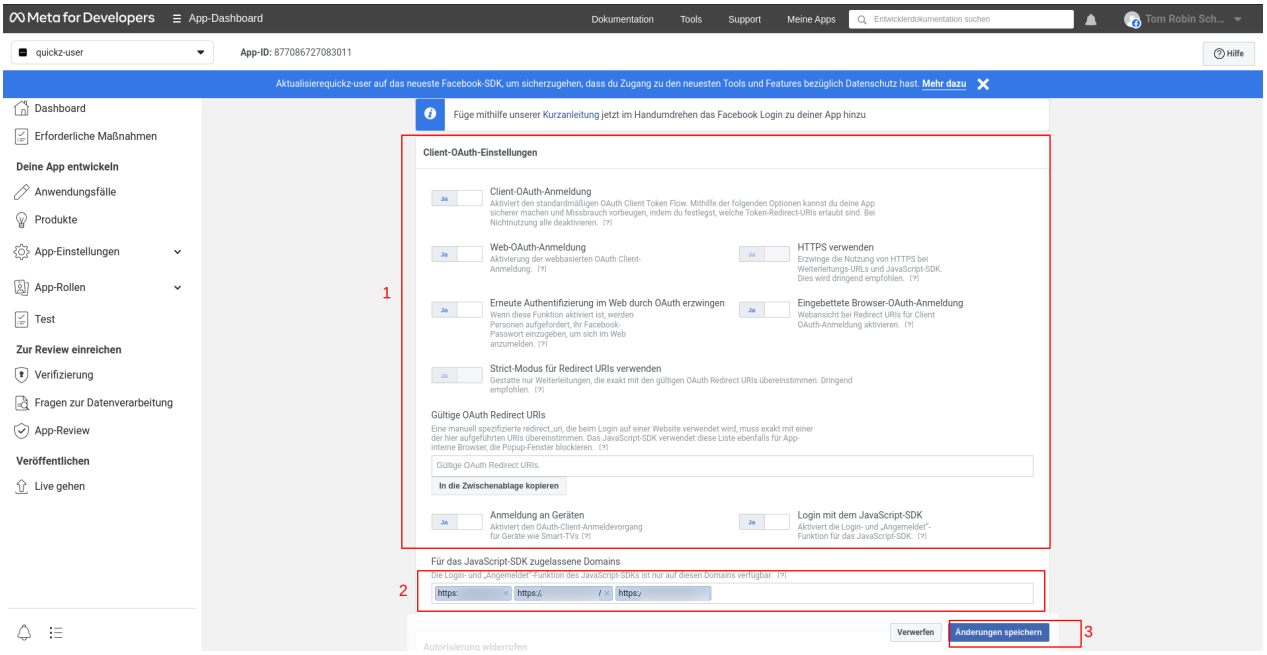
12.5 Click on Products

Click on **Products** -> **Configure** -> **Settings**



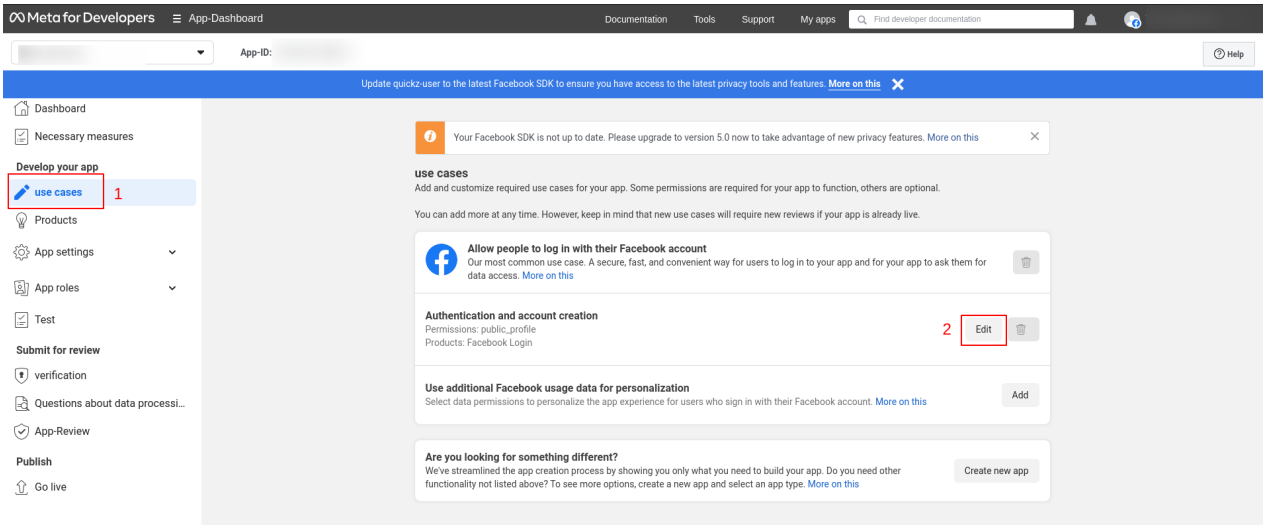
12.6 Add Domains of your project

1. Toggle all to **yes**, 2. Fill the domains, from which you will use, 3.Click and continue.

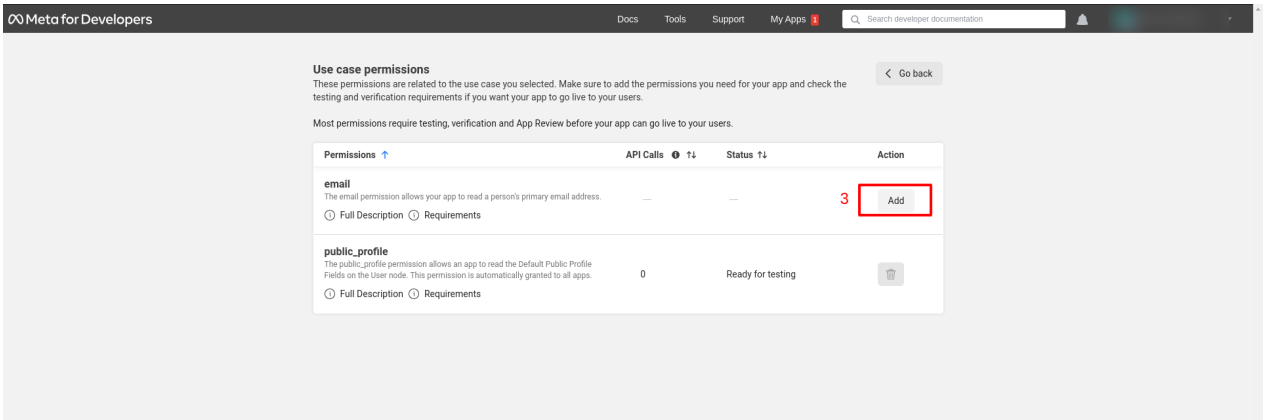


12.7 Give advance access

1. **Click use cases**
2. **Click Edit**



3. **Add**



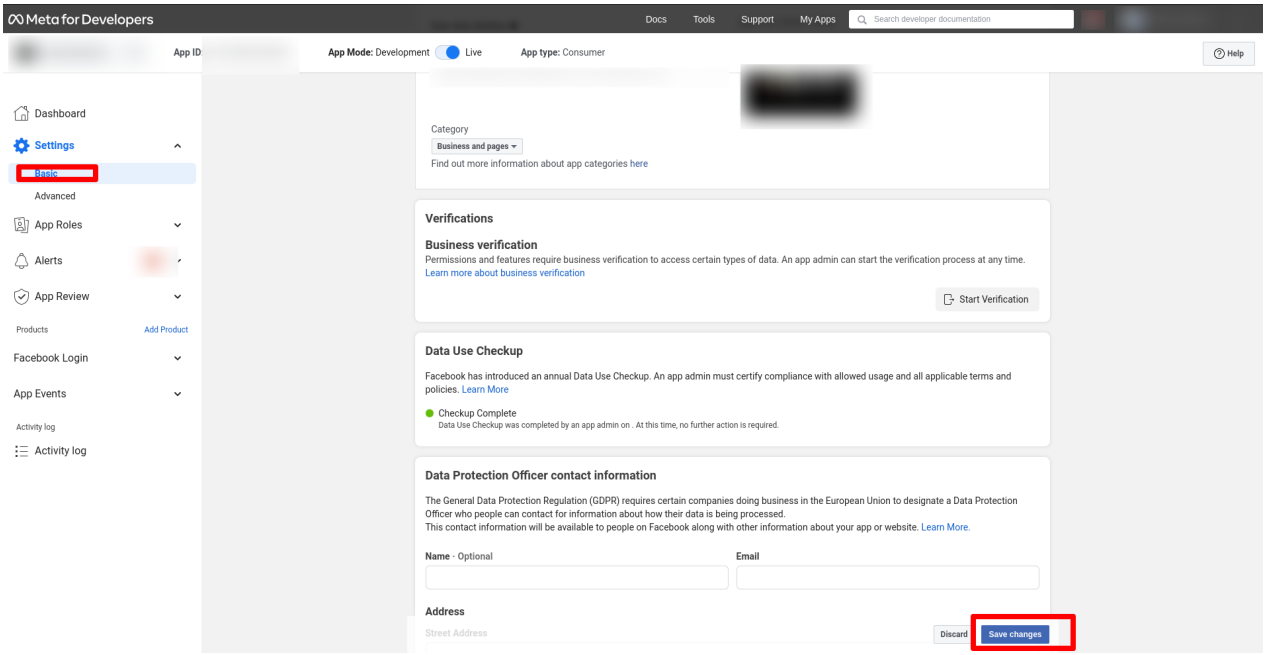
12.8 Complete basic settings

Complete basic settings, privacy-terms condition, data deletion url, logo change if want, category business and pages, add website, app, ios.

12.9 Data use check up

Data use check up if it already completes its signal as green. If not you have to complete 2 easy setup. Only tap yes this both steps

12.10 Save the changes



IMPORTANT:

12.11 Replace FB App ID

Add fb **appId** in user panel and driver panel
Search for (**FB.init**) in the below mentioned file.

-> Project-Folder/driver-panel/src/app/views/app/auth/auth.component.ts
-> Project-Folder/user-panel/src/app/containers/pages/auth-modal/auth-modal.component.ts

13. Note For Folder Structure.

In the delivered code you might not get following files/folders because it will be generated or added after installation.
Some of the folders contain dynamic files/folders.

13.1 List of empty folders in clone:

- data/partner_document : Uploaded Images For Partner Document
- data/partner_profile : Uploaded Images For Partner Profile
- data/provider_document : Uploaded Images For Driver Document
- data/provider_profile : Uploaded Images For Driver Profile
- data/service_type_images : Uploaded Images For Vehicle Image
- data/service_type_map_pin_images : Uploaded Images For Vehicle Map Pin
- data/user_document : Uploaded Images For User Document
- data/user_profile : Uploaded Images For User Profile

13.2 File/Folder structure

The following file/folder will get auto generated after the installation

- node_modules : Auto Generate After Installation Of Node Dependencies
- package-lock.json : Auto Generate After Installation Of Node Dependencies
- log_files : Auto Generate Files For Logs
- data/xlsheet : Auto Generate xl sheets
- config/data : Auto Generate For Temporary Image Upload Cache

13.3 The following keys/file

The following keys/files are removed that need to be configured from a client account that needs to be pushed in the repository provided by the IOS developer.

- backend/server/app/ios_push/push_file.p8 - need to be added by admin from client account.

13.4 Overview of code.

Here the following instruction is for understanding the code quickly.

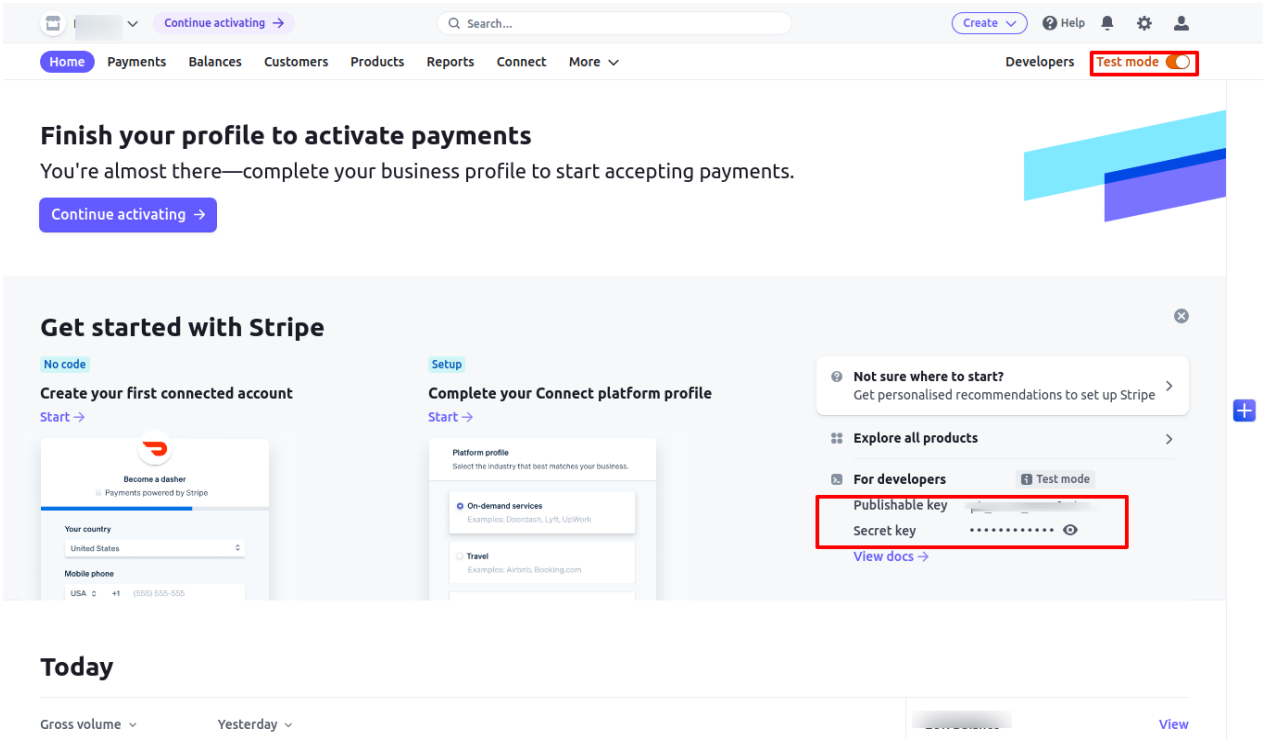
- Backend
    - There are a total of 4 backends in this project and all are necessary.
        - **server**
        - **history-earning**
        - **payments**
        - **mass_notification**
    - From the above list, the **server** is main. We can say, others are all very easy to understand.
        - **app** folder contains all the files like routes, controllers, models, utils etc.
        - **config** folder contains the env and basic configuration files.
        - **data** folder contains all the images, icons etc.
        - **documentation** folder contains files that explains the project.
        - The **app** and the **config** folder in the folder other than the server are similar to the server's app and config folder as explained above.
- Panels(Frontend)
    - All the panels follow Angular folder structure, therefore knowledge of Angular is must for understanding the panels code.

14. Stripe Configuration.

14.1 Login Stripe Account

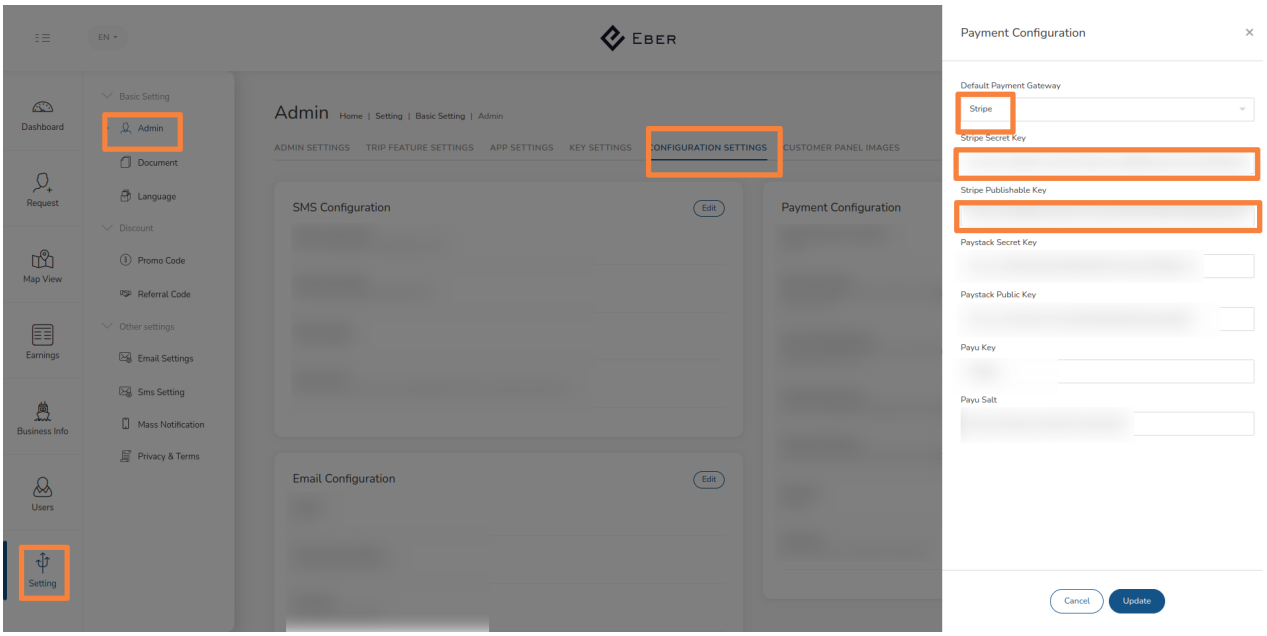
Account Login or Register

Keep Test mode toggle on as shown in the image below.
Use the following highlighted key for future reference. i.e, “**Publishable Key**” and “**Secret Key**”



14.2 Use this test mode private and public key,and save this key in the Admin panel.

Navigate to, **Admin panel** - > ***Setting*** -> ***Admin*** -> ***Configuration Settings*** -> ***Payment Configuration***

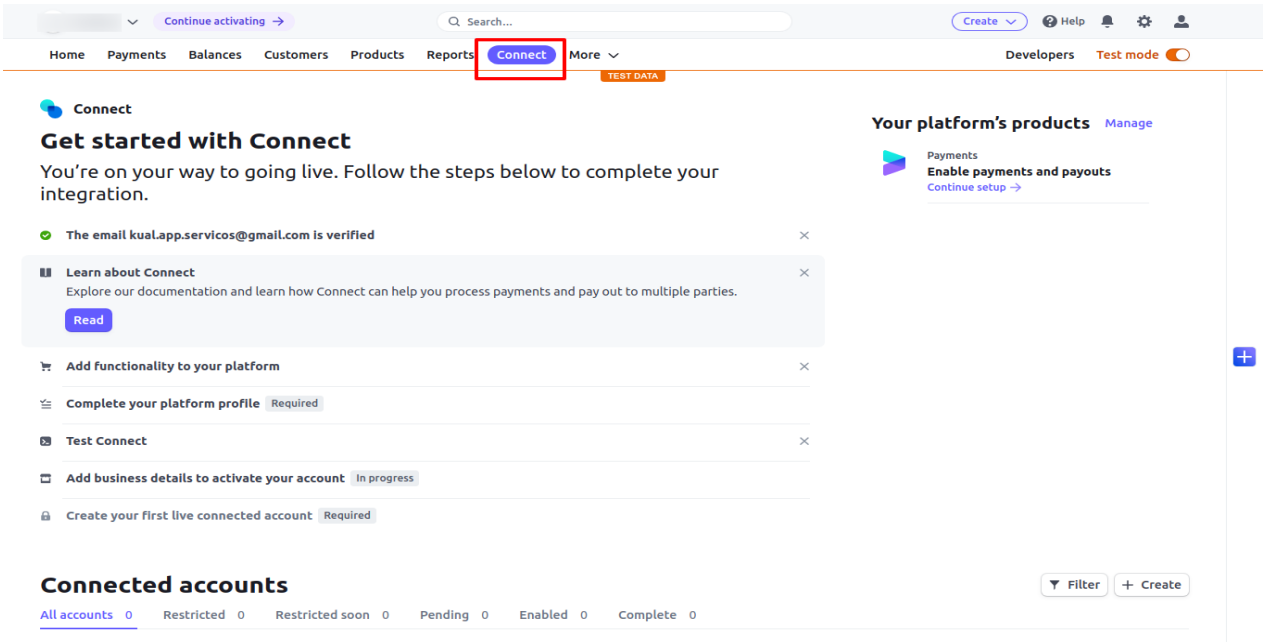


14.3 Add stripe publishable key in to user panel, partner panel & driver panel in code

-> src/environments/environment.ts
-> src/environments/environment.prod.ts

14.4 And start the Connect Account

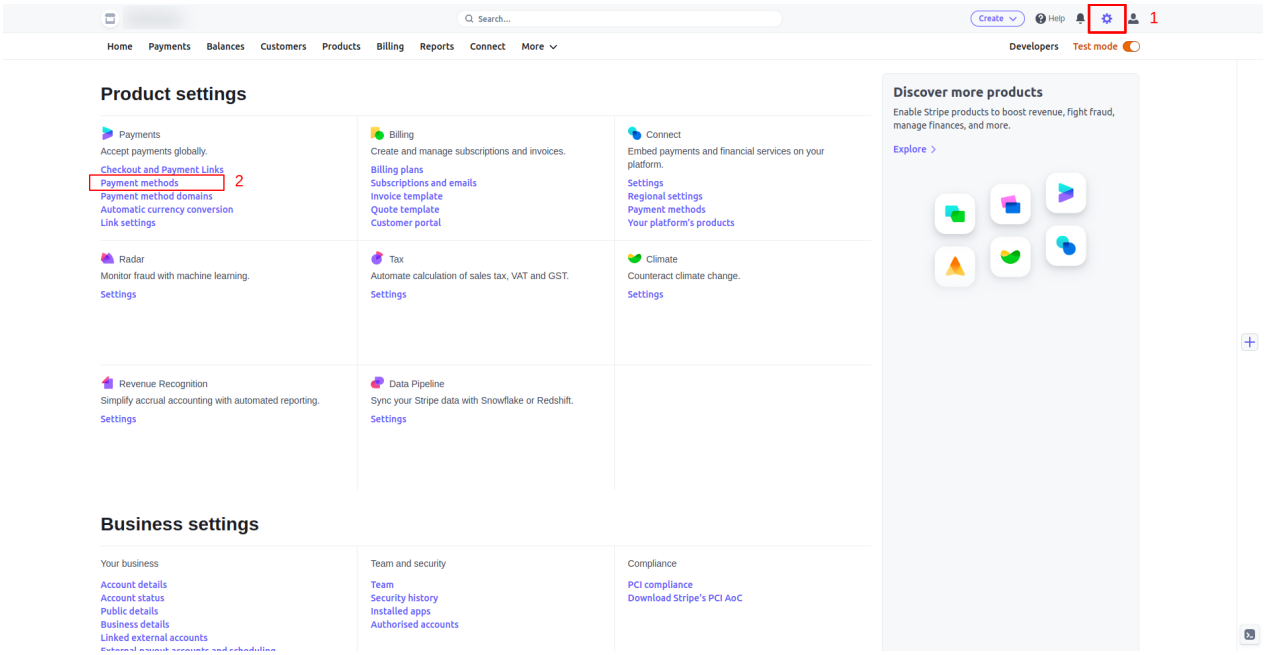
Go to the home page and navigate to the connect tab and if it is not visible search for **connect** in the search bar.



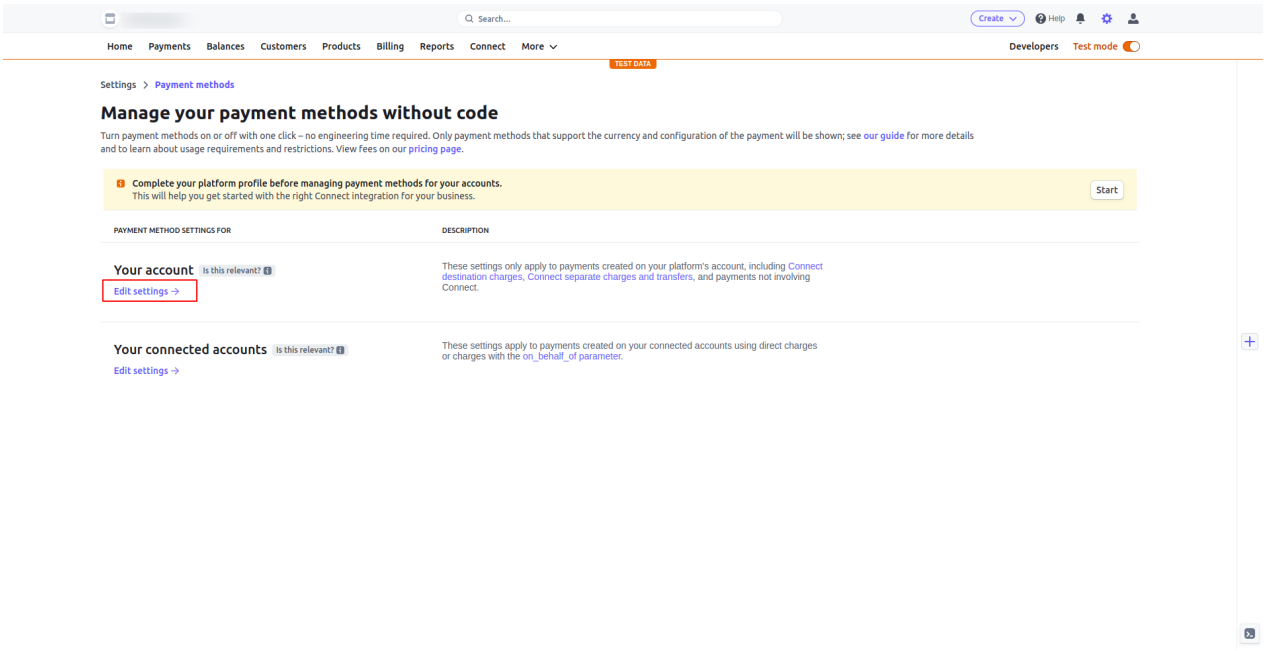
14.5 Activate Apple Pay

Need to activate the Apple pay method before using its feature.

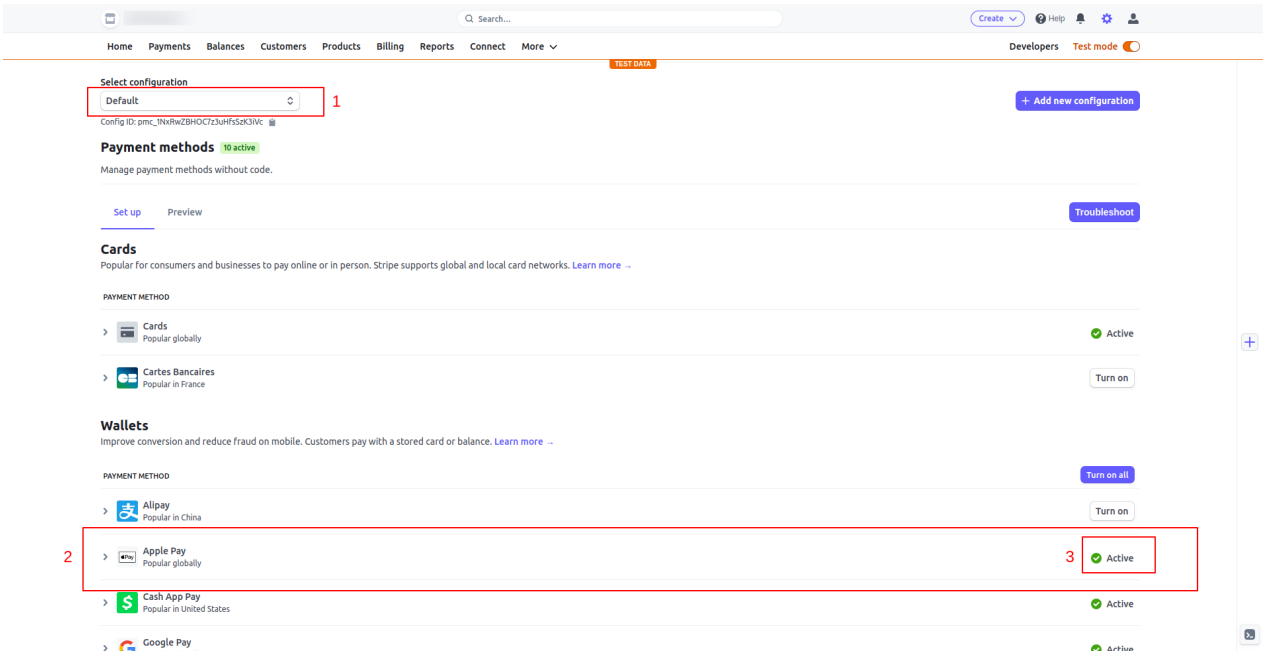
1. **Go to settings**
2. **Payment Methods**



Under the you account click on **Edit Settings**



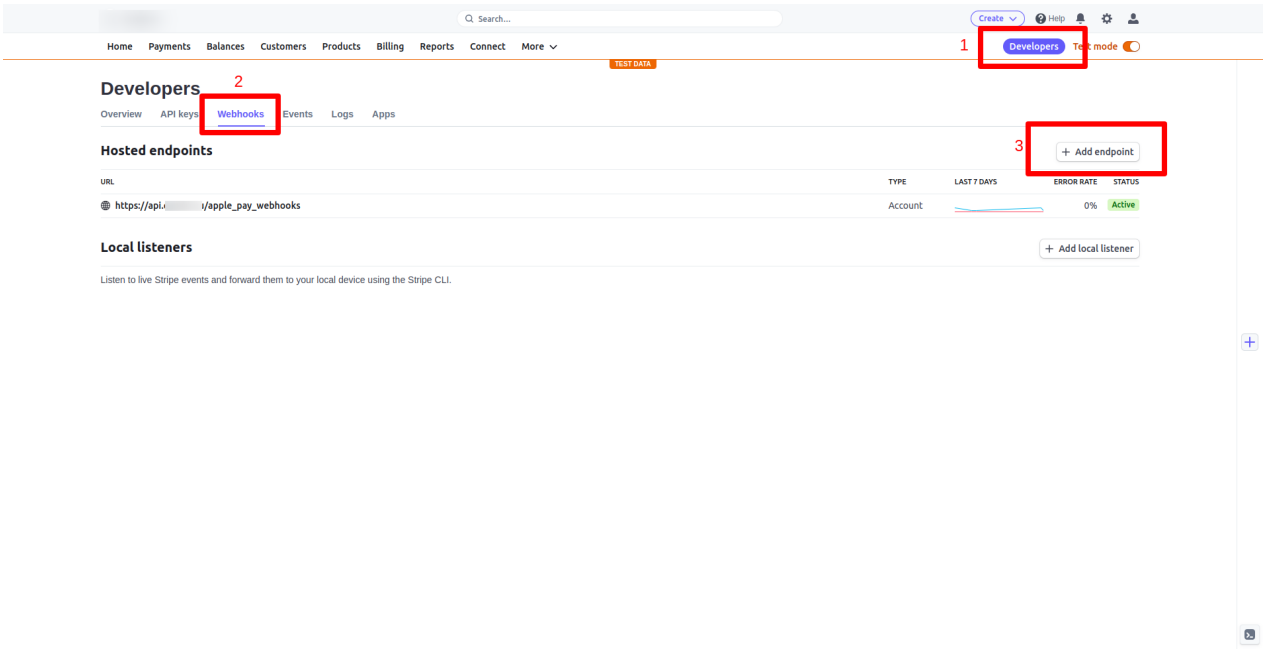
1. **Default**
2. **Apple Pay**
3. **Turn on** (here in the screenshot it is **Active**, but for you it will be **Turn on** button)



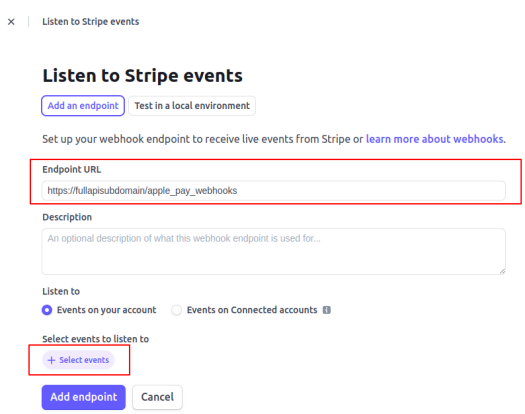
And then complete the process.

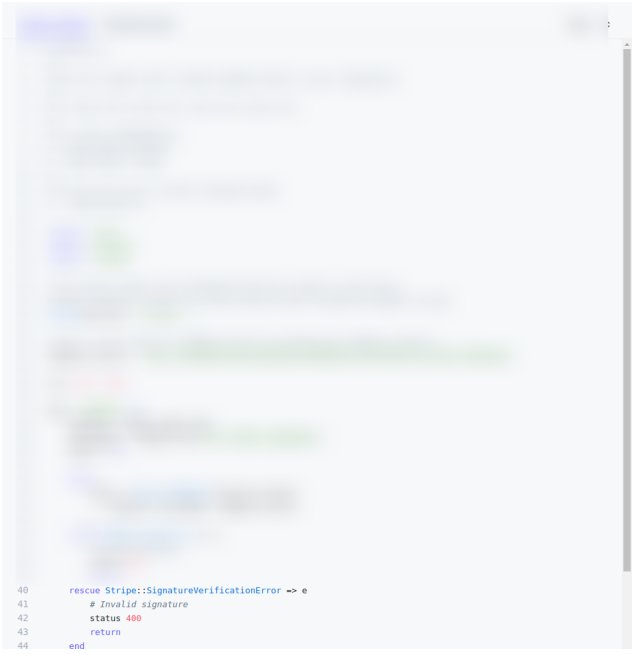
14.6 Add Webhook Url for Using Apple pay.

Navigate to, 1.**Developer** -> 2.**Webhooks** -> 3. **Add endpoint**

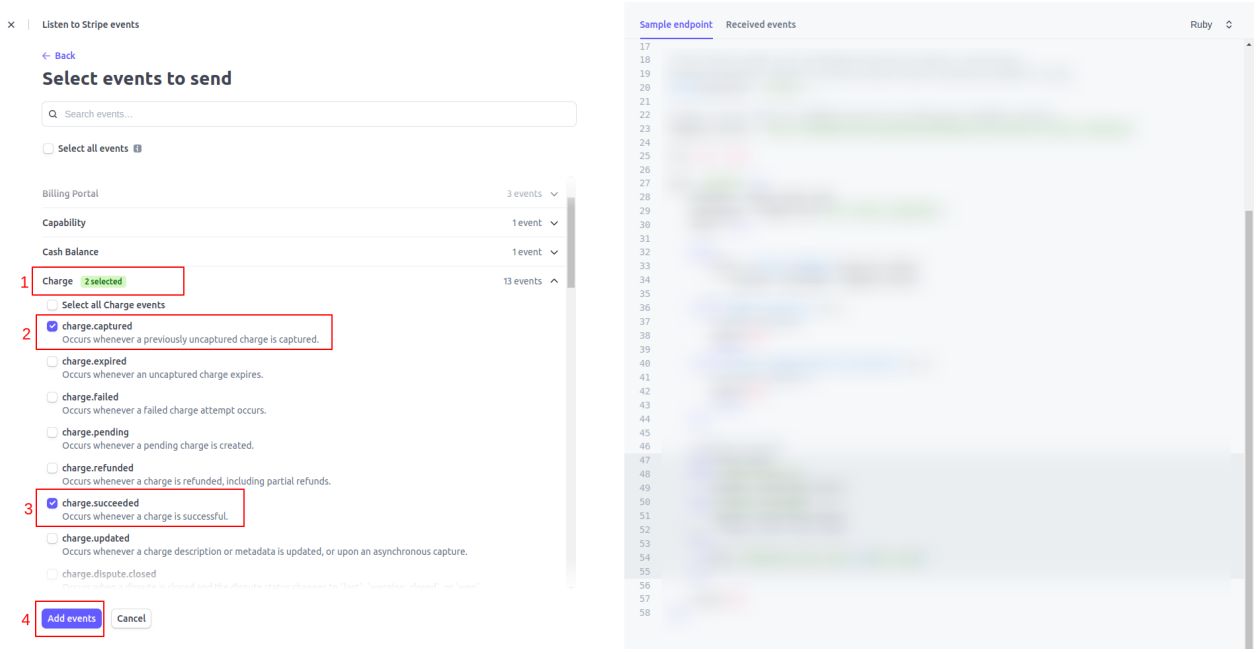


Fill the full url same as given below and replace the “fullapisubdomain” with your api url sub-domain. i.e, *https://fullapisubdomain/apple_pay_webhooks*
Then click on “Select events”.

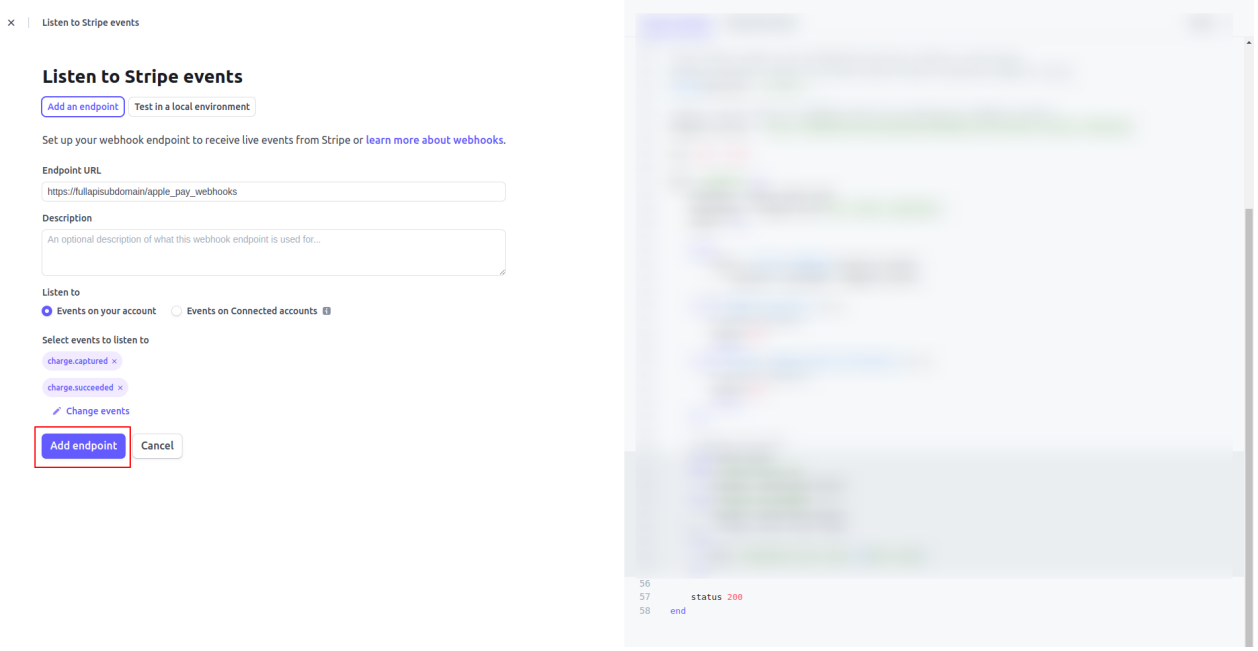




Select *1.**Charge*** -> *2.**charge.captured*** -> *3.**charge.succeeded***, and then click
4. **Add events**



Click the **Add endpoint** button at the end and it will create an endpoint.

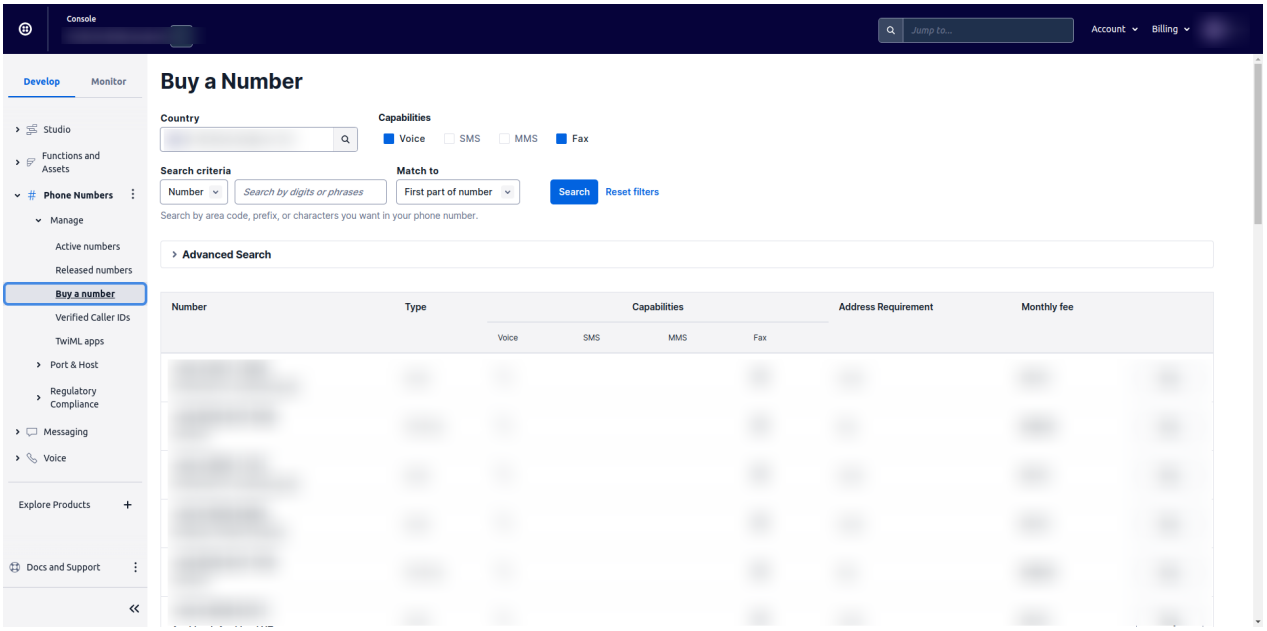


15. Twilio Configuration.

15.1 Login twilio

Account Login or Register

15.2 Purchase number which have sms and voice capability

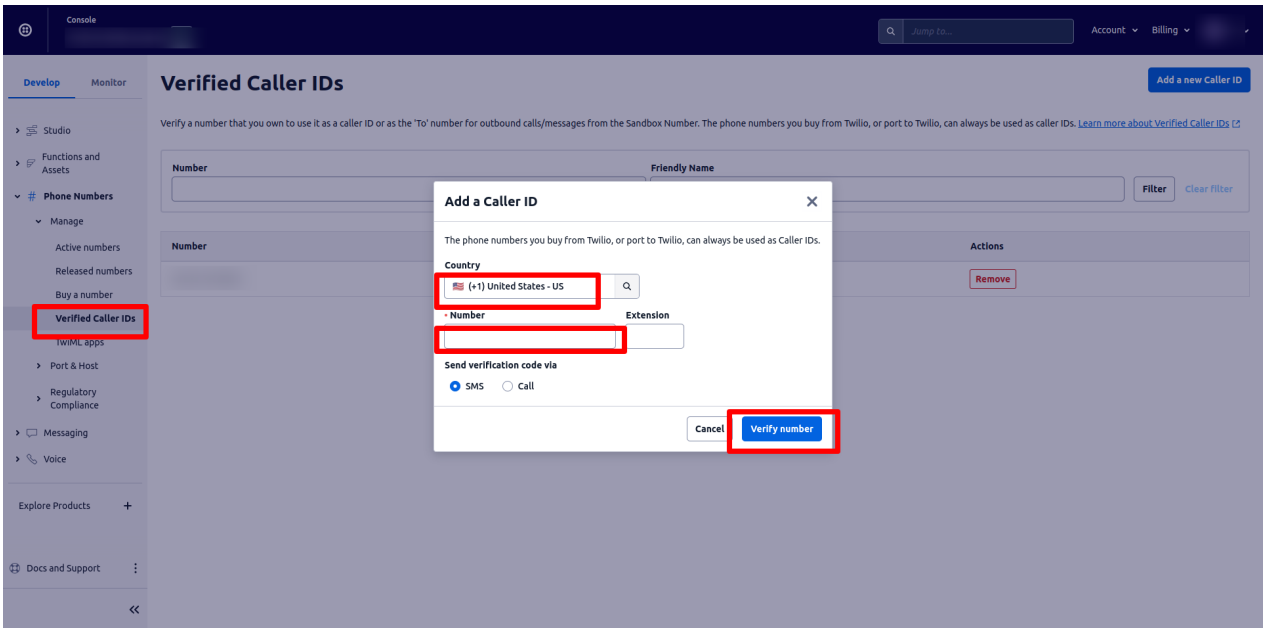


Suggested numbers are in the list, purchase number as per your country code.

For example, a US number starts with code (+1).

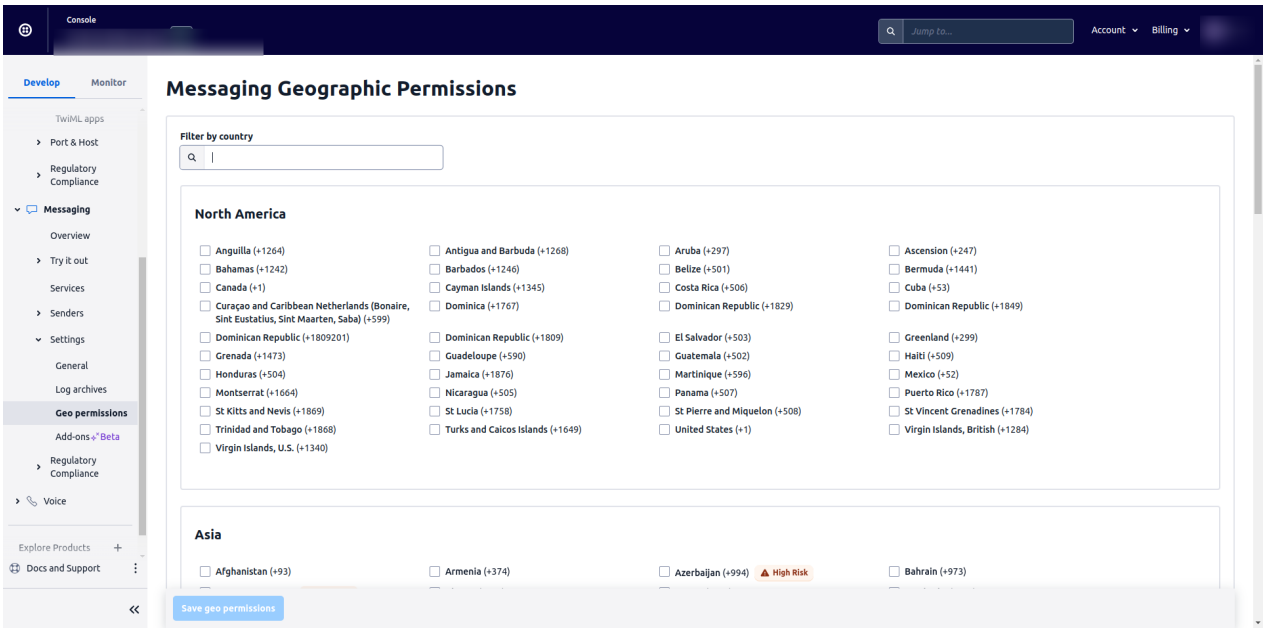
If you are choosing a US number, you will have to register for A2P 10DLC according to the new policy of Twilio.

15.3 In the trial account you have to verify number for test purpose.

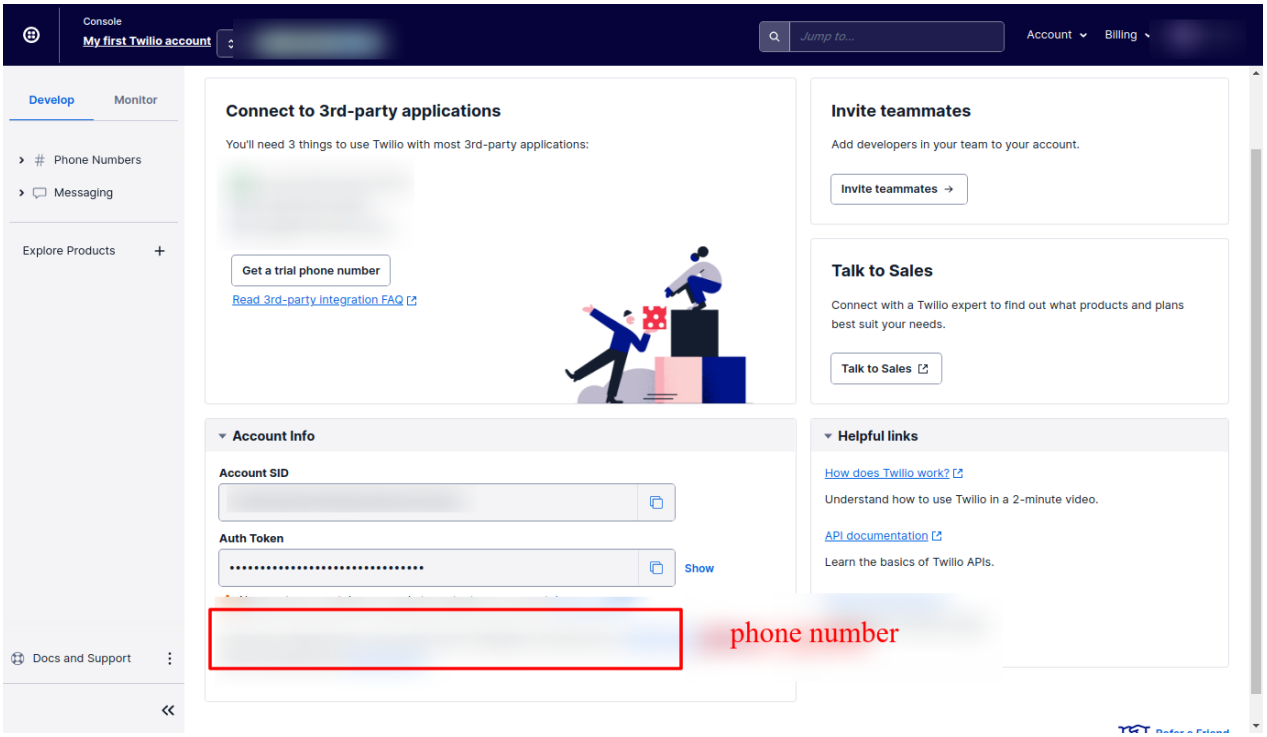


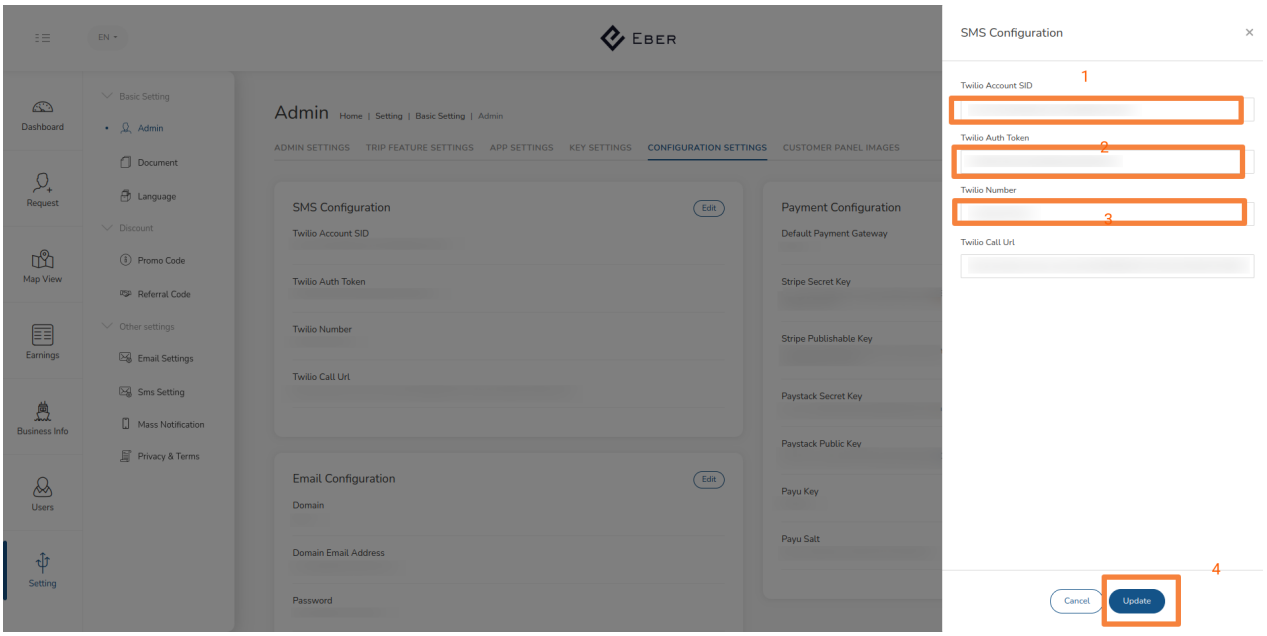
15.4 Check your country

Check your country to have permission to enable service for the number you verified above. Go to messaging -> settings -> geo permissions



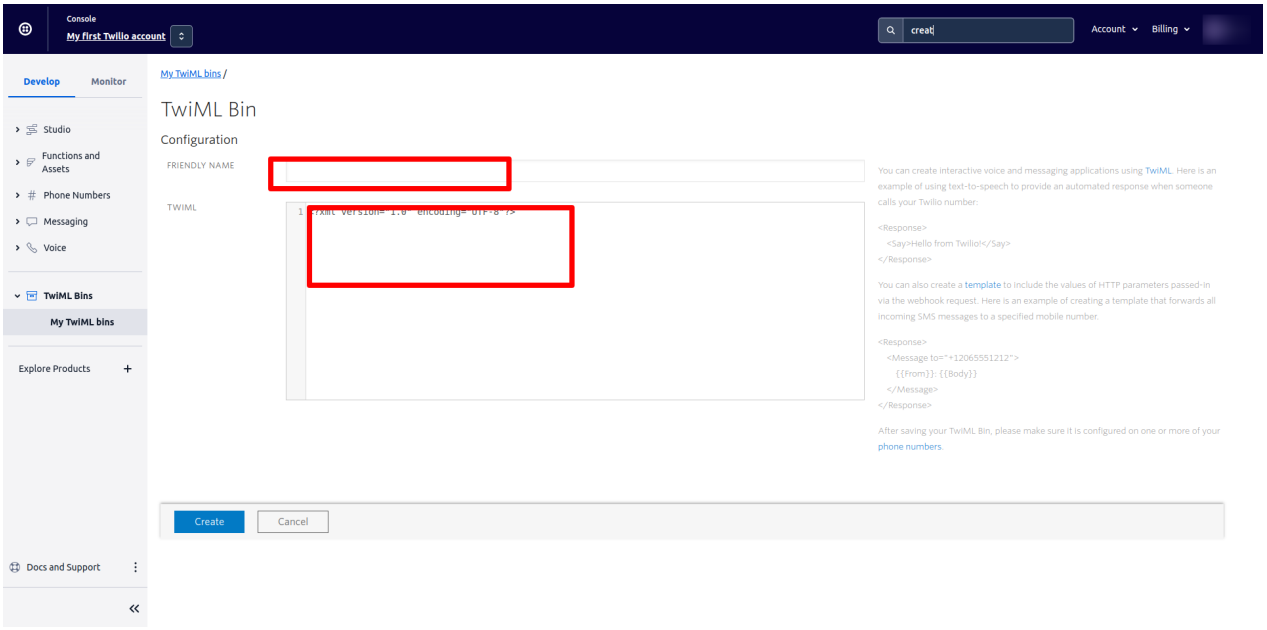
15.5 And save this Account SID,Auth Token & Phone number in Admin Panel Settings.





15.6 Twilio Call Masking

-> Go to search bar and search create twiml bin

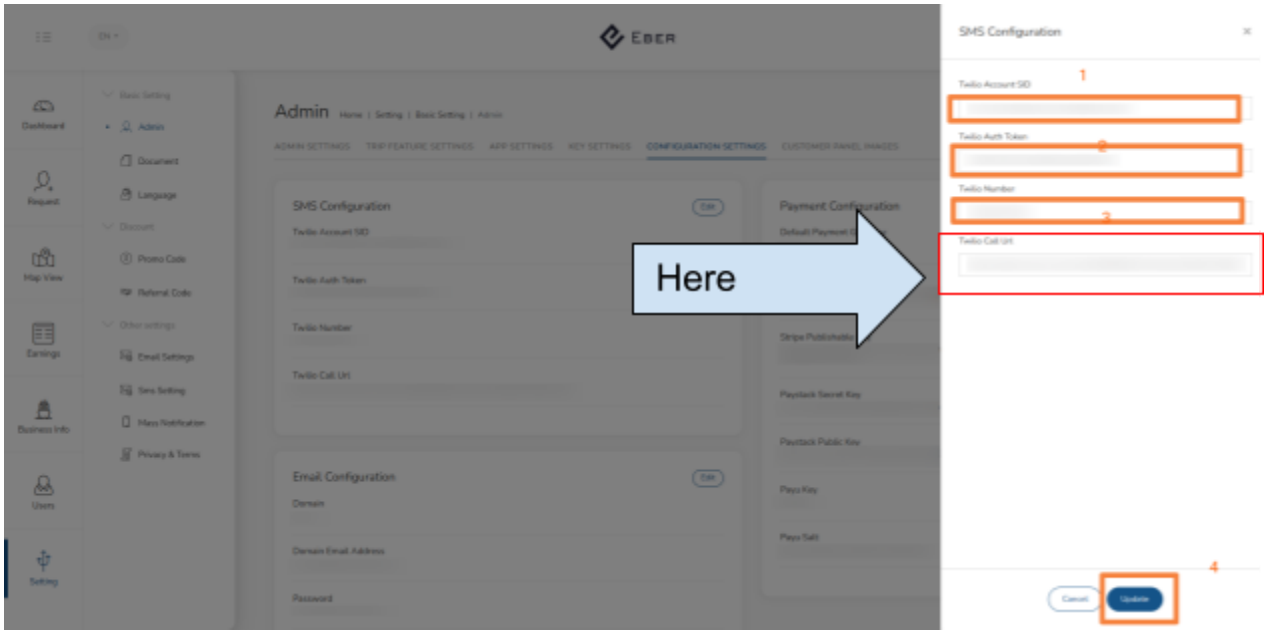


Any friendly name you want. Enter twiml code as below

```
<Response>
<Say>Hello From Twilio!</Say>
<Dial>
<Number>{{to}}</Number>
</Dial>
</Response>
```

Then create

> you got twiml url after this copy and paste in our admin panel settings


Here
SMS Configuration
1
2
3
4

16. Terms & Condition Details.

16.1 Terms and Privacy Document add in to admin panel.

Navigate to, **admin panel** -> ***setting*** -> ***Privacy & Terms*** and paste the terms and policy and save it respectively.