

# Installation of EBER

## 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

steps to purchase instance

### 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.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'EC2 Dashboard', 'Instances', 'Images', 'Elastic Block Store', 'Network & Security', and 'Load Balancing'. The main area has a 'Resources' section displaying metrics for Instances (running), Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, and Volumes. Below this is a 'Launch instance' section with a 'Launch instance' button, and a 'Scheduled events' section stating 'No scheduled events'. To the right, there's a 'Service health' section showing 'This service is operating normally' and a 'Zones' table with three rows: us-east-2a (use2-az1), us-east-2b (use2-az2), and us-east-2c (use2-az3). A large dropdown menu on the right lists various AWS regions, with 'Ohio' highlighted. The menu includes regions like US East (N. Virginia), US East (Ohio), US West (N. California), US West (Oregon), Africa (Cape Town), Asia Pacific (Hong Kong), Asia Pacific (Mumbai), Asia Pacific (Osaka), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), Europe (Frankfurt), Europe (Ireland), Europe (London), Europe (Milan), Europe (Paris), Europe (Stockholm), Middle East (Bahrain), and South America (São Paulo).

# 1.2 Purchasing Instance

## 1.2.1: From AWS services select EC2.

The screenshot shows the AWS Management Console homepage. In the top left, there's a sidebar titled "AWS services" with a section for "Recently visited services" where "EC2" is highlighted with a red box. Below this are sections for "Build a solution" and "Explore AWS". The "Explore AWS" section includes links for "Amazon Lookout for Metrics", "Free AWS Training", and "Calling All Java and Python Developers". At the bottom, there are links for "Feedback", "English (US)", and various AWS links like "Privacy Policy", "Terms of Use", and "Cookie preferences".

## 1.2.2: Now, from the side navigation bar select instances.

The screenshot shows the "Welcome to the new EC2 console" page. The left sidebar has "Instances" selected, which is highlighted with a red box. The main content area shows "Resources" for the Asia Pacific (Mumbai) Region, including tables for Instances (running), Instances, Placement groups, Volumes, Key pairs, Security groups, Load balancers, and Snapshots. Below this are sections for "Launch instance", "Service health", and "Zones". On the right, there's an "Account attributes" panel and an "Explore AWS" panel with links for "Amazon EBS Backup and Restore" and "10 Things You Can Do Today to Reduce AWS Costs". The bottom of the page includes standard AWS footer links.

1.2.3: Select Launch from the top right corner.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Limits, Instances (with sub-links for Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and footer links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences. The main area displays a table titled 'Instances (17) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 ... (partially visible). All 17 instances listed are 'Running' and have 't2.medium' as their instance type. The 'Actions' dropdown menu at the top right has a 'Launch instances' option, which is highlighted with a red box. A tooltip 'Select an instance above' is visible near the bottom left of the table area.

1.2.4: Now we few steps to follow for 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.

Sales Services ▾

Search for services, features, marketplace products, and docs [Alt+S]

Cancel and Exit

**Step 1: Choose an Amazon Machine Image (AMI)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Quick Start

My AMIs

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-011c99152163a87ae (64-bit x86) / ami-0fcc3ea54be0b9c73 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.

Select  
64-bit (x86)  
64-bit (Arm)

AWS Marketplace

Community AMIs

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-06a0b4e3b7eb7a300 (64-bit x86) / ami-0cbe04a3ce796c98e (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Select  
64-bit (x86)  
64-bit (Arm)

Free tier only ⓘ

SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0b3acf3edf2397475 (64-bit x86) / ami-0ab71076ab9b53b0d (64-bit Arm)

SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.

Select  
64-bit (x86)  
64-bit (Arm)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0ca17f89451184c8b (64-bit x86) / ami-0d18acc6e813fd2e0 (64-bit Arm)

Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Select  
64-bit (x86)  
64-bit (Arm)

Free tier eligible

Microsoft Windows Server 2019 Base - ami-0e5d82cae745873b8

Windows

Microsoft Windows 2019 Datacenter edition, [English]

Select  
64-bit (x86)

Free tier eligible

Are you launching a database instance? Try Amazon RDS.

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy **Amazon Aurora**, **MariaDB**, **MySQL**, **Oracle**, **PostgreSQL**, and **SQL Server** databases on AWS. Aurora is a MySQL- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. Learn more about RDS

Feedback English (US) ▾

© 2008 – 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

### 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

	AMI	Instance Type	Cores	Memory (GiB)	Storage	Network	Processor
1.	t4g	t4g.large	2	8	EBS only	Yes	Up to 5 Gigabit
2.	t4g	t4g.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit
3.	t4g	t4g.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit
4.	c4	c4.large	2	3.75	EBS only	Yes	Moderate
5.	c4	c4.xlarge	4	7.5	EBS only	Yes	High
6.	c4	c4.2xlarge	8	15	EBS only	Yes	High
7.	c4	c4.4xlarge	16	30	EBS only	Yes	High
8.	c4	c4.8xlarge	36	60	EBS only	Yes	10 Gigabit
9.	<b>c5</b>	<b>c5.large</b>	<b>2</b>	<b>4</b>	<b>EBS only</b>	<b>Yes</b>	<b>Up to 10 Gigabit</b>
10.	c5	c5.xlarge	4	8	EBS only	Yes	Up to 10 Gigabit
11.	c5	c5.2xlarge	8	16	EBS only	Yes	Up to 10 Gigabit
12.	c5	c5.4xlarge	16	32	EBS only	Yes	Up to 10 Gigabit
13.	c5	c5.9xlarge	36	72	EBS only	Yes	10 Gigabit
14.	c5	c5.12xlarge	48	96	EBS only	Yes	12 Gigabit
15.	c5	c5.18xlarge	72	144	EBS only	Yes	25 Gigabit
16.	c5	c5.24xlarge	96	192	EBS only	Yes	25 Gigabit
17.	c5	c5.metal	96	192	EBS only	Yes	25 Gigabit

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

#### 1.2.4(iii) Configure instance details

- Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.
- Configure your server according to your requirements. Let's head to the next step.

#### 1.2.4(iv) Add storage

- 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.

**Step 4: Add Storage**

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. [Learn more about storage options in Amazon EC2](#).

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0c063602c11839b7c	80	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more about free usage tier eligibility and usage restrictions.](#)

Cancel Previous Review and Launch Next: Add Tags

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

### 1.2.4(v) Add Tags(Optional)

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

### 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.

Sales Services ▾ Search for services, features, marketplace products, and docs [Alt+S]

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

**Step 6: 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. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:  Create a new security group  Select an existing security group

Security group name: launch-wizard-13

Description: launch-wizard-13 created 2021-07-08T11:37:02.756+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

Feedback English (US) ▾ © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

### 1.2.4(vii) Review

- Please review your instance launch details. You can go back to edit changes for each section.
- Click Launch to assign a key pair to your instance and complete the launch process.

### 1.2.4(viii) 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.

**Step 7: Review Instance Launch**

To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about free usage tier eligibility and usage restrictions.

**AMI Details**

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0b1deed  
Free tier eligible  
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Supp...  
Root Device Type: ebs Virtualization type: hvm

**Instance Type**

Instance Type	ECUs	vCPUs	Memory (GiB)
t2.medium	-	2	4

**Security Groups**

Security group name	Description
launch-wizard-1	launch-wizard-1 created 2021-07-09T16:01:20.070+00:00

**Type** **Protocol**

Type	Protocol	Port Range
SSH	TCP	22
HTTP	TCP	80
HTTP	TCP	80
HTTPS	TCP	443
HTTPS	TCP	443

**Select an existing key pair or create a new key pair**

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types. ED25519 keys are smaller and faster while offering the same level of security as RSA keys. Use ED25519 keys to improve the speed of authentication or if you have regulatory requirements that mandate the use of ED25519 keys.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair  
Select a key pair  
No key pairs found

No key pairs found  
You don't have any key pairs. Please create a new key pair by selecting the Create a new key pair option above to continue.

**Launch Instances**

**Edit instance type** **Edit security groups** **Edit instance details** **Edit storage** **Edit tags** **Cancel** **Previous** **Launch**

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

**Step 7: Review Instance Launch**

To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about free usage tier eligibility and usage restrictions.

**AMI Details**

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0b1deed  
Free tier eligible  
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Supp...  
Root Device Type: ebs Virtualization type: hvm

**Instance Type**

Instance Type	ECUs	vCPUs	Memory (GiB)
t2.medium	-	2	4

**Security Groups**

Security group name	Description
launch-wizard-1	launch-wizard-1 created 2021-07-09T16:01:20.070+00:00

**Type** **Protocol**

Type	Protocol	Port Range
SSH	TCP	22
HTTP	TCP	80
HTTP	TCP	80
HTTPS	TCP	443
HTTPS	TCP	443

**Select an existing key pair or create a new key pair**

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types. ED25519 keys are smaller and faster while offering the same level of security as RSA keys. Use ED25519 keys to improve the speed of authentication or if you have regulatory requirements that mandate the use of ED25519 keys.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair  
Key pair name  
example  
**Download Key Pair**

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

**Launch Instances**

**Edit instance type** **Edit security groups** **Edit instance details** **Edit storage** **Edit tags** **Cancel** **Previous** **Launch**

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

## 2). Add Elastic IP address in AWS

### 2.1) What is Elastic Ip and why it is 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.

### 2.2) Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>

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

### 2.3) Choose Allocate Elastic IP address.

The screenshot shows the AWS Management Console interface for managing Elastic IP addresses. The left sidebar lists various AWS services, with 'Elastic IPs' under 'Network & Security' highlighted with a red box. The main content area displays a table titled 'Elastic IP addresses (2)' containing two entries: 'panel' and 'Backend'. Both entries are listed as 'Public IP' type. At the top right of the table, there is a red box around the 'Actions' dropdown menu, specifically highlighting the 'Allocate Elastic IP address' option.

Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record
panel		Public IP	ip-10-0-10-125.ec2.us-east-2.amazonaws.com	-
Backend		Public IP	ip-10-0-10-126.ec2.us-east-2.amazonaws.com	-

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

The screenshot shows the 'Allocate Elastic IP address' page in the AWS Management Console. At the top, there's a search bar with 'Elastic IP addresses'. Below it, the main section is titled 'Elastic IP address settings' with an 'Info' link. It contains three options for the 'Public IPv4 address pool': 'Amazon's pool of IPv4 addresses' (selected), 'Public IPv4 address that you bring to your AWS account', and 'Customer owned pool of IPv4 addresses'. A note says 'Learn more' for each option. Below this is a section for 'Global static IP addresses' with a note about AWS Global Accelerator. A 'Create accelerator' button is available. The next section is 'Tags - optional', which includes a note about tags and a 'Add new tag' button. A note says 'You can add up to 50 more tag'. At the bottom right, there are 'Cancel' and 'Allocate' buttons, with 'Allocate' being highlighted with a red box. The footer includes links for Feedback, English (US), Copyright notice (© 2021, Amazon Web Services, Inc. or its affiliates.), Privacy, Terms, and Cookie preferences.

## 2.5) Now Select IP and click on allocate an elastic IP address.

The screenshot shows the 'Elastic IP addresses (1/2)' list page in the AWS Management Console. On the left, there's a sidebar with 'New EC2 Experience' (Tell us what you think), 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (Instances New, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances New, Dedicated Hosts, Capacity Reservations), 'Images' (AMIs New, AMI Catalog), and 'Elastic Block Store' (Volumes New, Snapshots New, Lifecycle Manager New). The main area shows a table with one row: 'panel' (Backend, Allocated IPv4 address: 13.214.13.212, Type: Public IP). To the right of the table is an 'Actions' menu with several options: 'View details', 'Release Elastic IP addresses', 'Associate Elastic IP address' (which is highlighted with a red box), 'Disassociate Elastic IP address', and 'Update reverse DNS'. Below the table, there's a summary for the selected IP: '13.214.13.212' with tabs for 'Summary' and 'Tags'. The 'Summary' tab displays details like 'Allocated IPv4 address' (13.214.13.212), 'Type' (Public IP), 'Scope' (VPC), 'Allocation ID' (redacted), 'Associated Instance ID' (redacted), 'Reverse DNS record' (redacted), and 'Private IP address' (redacted). The footer includes links for Feedback, English (US), Copyright notice (© 2021, Amazon Web Services, Inc. or its affiliates.), Privacy, Terms, and Cookie preferences.

## 2.6) Select your project Instance and allocate.

The screenshot shows the AWS EC2 'Associate Elastic IP address' interface. At the top, the navigation bar includes 'Services' and a search bar for 'Elastic IP addresses'. Below the navigation, the breadcrumb path shows 'EC2 > Elastic IP addresses > Associate Elastic IP address'. The main title is 'Associate Elastic IP address'. A note at the top says 'Choose the instance or network interface to associate to this Elastic IP address ('. The 'Elastic IP address:' field contains a placeholder. Under 'Resource type', the 'Instance' option is selected. A warning message states: '⚠ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)'.

**Instance**

Search bar: choose an instance

Results: (panel) - running  
(Backend) - running

**Reassociation**

Specify whether the Elastic IP address can be reassigned to a different resource if it's already associated with one.

Allow this Elastic IP address to be reassigned

Buttons: Cancel, Associate

Page footer: Feedback English (US) ▾ © 2021, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### 3). Create s3 Bucket in AWS.

#### 3.1) 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.

#### 3.2) Click on s3 from all services

The screenshot shows the AWS Management Console homepage. The 'Services' menu is open, and the 'Storage' section is selected. Within 'Storage', 'S3' is highlighted with a red box. Other services listed in 'Storage' include EFS, FSx, S3 Glacier, Storage Gateway, AWS Backup, AWS Elastic Disaster Recovery, and AWS CloudWatch Metrics. The main content area displays several training and learning resources: 'Free AWS Training' (Cloud Practitioner Essentials), 'AWS Cloud Training' (Comprehensive training for cloud adoption), and 'AWS Machine Learning Training' (Courses for machine learning pipeline). A 'Have feedback?' section allows users to submit feedback via email. The bottom of the page includes standard footer links for Feedback, English (US), Copyright notice (© 2021, Amazon Web Services, Inc. or its affiliates.), Privacy, Terms, and Cookie preferences.

### 3.3) Click on create a bucket

The screenshot shows the AWS S3 console interface. On the left, there's a navigation sidebar with links like 'Buckets', 'Access Points', 'Object Lambda Access Points', etc. The main area displays an 'Account snapshot' with a 'Storage lens provides visibility into storage usage and activity trends' message. Below it is a table titled 'Buckets (1) info' with one entry. The 'Create bucket' button is highlighted with a red box. At the bottom of the page, there are standard footer links for 'Feedback', 'English (US)', '© 2021, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

### 3.4) 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.

**Object Ownership** Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Object Ownership**

**Bucket owner preferred**  
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

**Object writer**  
The object writer remains the object owner.

If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

### 3.5) Unblock public access.

- We need to disable the block off public access
- Accept the Acknowledge because our bucket gives the public access.

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

**Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

**Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.

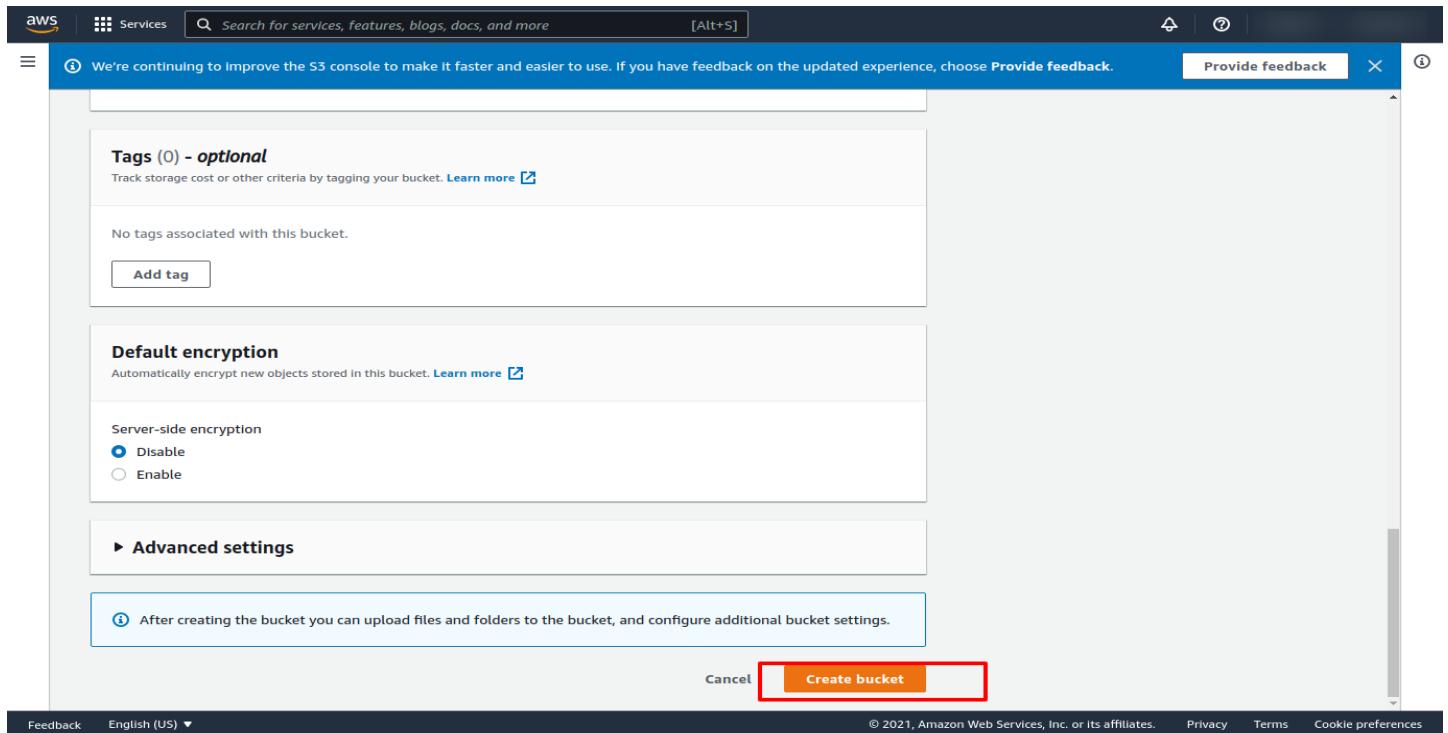
**Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

**Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**⚠ Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

### 3.6) Now Click on Create Bucket.



### 3.7) 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) It will look something like - "http://your-bucket.s3-website-us-east-1.amazonaws.com"

The screenshot shows the AWS S3 console. On the left, there's a sidebar with various options like Buckets, Storage Lens, and AWS Marketplace. The main area is titled 'Object overview' for an object named 'Ovv.jpg'. It displays details such as Owner (developers), AWS Region (redacted), Last modified (December 7, 2021, 16:29:33 (UTC+05:30)), Size (9.3 KB), Type (jpg), and Key (redacted). To the right, there are fields for S3 URI, Amazon Resource Name (ARN), Entity tag (Etag), and Object URL. The 'Object URL' field is highlighted with a red box.

### 3.8) S3 bucket Access key and secret key download and save it.

- 1) Save this key & Add this Acces key & Secret key in the Database..
- 2) Add bucket name in Database.

The screenshot shows the AWS IAM 'Your Security Credentials' page. The left sidebar includes sections for Identity and Access Management (IAM), Dashboard, Access management, Access reports, and Credential report. The main content area is titled 'Your Security Credentials' and lists 'Access keys (access key ID and secret access key)'. A table shows access keys for a user named 'AL' (Access Level). A 'Create New Access Key' button is highlighted with a red box. The top navigation bar has a 'Global' dropdown with a red box around the number '1'. A secondary navigation menu on the right includes links for Account, Organization, Service Quotas, Billing Dashboard, and Security credentials, which is highlighted with a green box. The bottom of the page includes a 'Sign out' button and copyright information.

## 4). Pointing Domain name, and SSL certificate

Now as our code is now 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

### 4.1 Domain Name(DNS)

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 @ and in IPv4 enter the IP address of our server, then click Save. For subdomains click add new record again and in the type section select CNAME and in name section enter admin and in IPv4 enter @ (it indirectly points to our server IP address). Add 3 more subdomains for API, store, and for user.

Generally, for our user panels, we use our domain name instead of sub domains, And now we are done with our Domain pointing.

The screenshot shows the Cloudflare dashboard with the DNS section selected. At the top, there's a message: "A few more steps are required to complete your setup." Below it, a note says: "Some of your DNS only records are exposing IPs that are proxied through Cloudflare. Make sure to proxy all A, AAAA, and CNAME records pointing to proxied records to avoid exposing your origin IP." The main area is titled "DNS management for [REDACTED]" and contains a table of DNS records. The table has columns for Type, Name, IPv4 address, TTL, and Proxy status. One record is visible: Type A, Name @, IPv4 address [REDACTED], TTL Auto, Proxy status Proxied. There are "Cancel" and "Save" buttons at the bottom right of the table. The Cloudflare logo is in the top left corner, and the top navigation bar includes links for Overview, Analytics, DNS, SSL/TLS, Firewall, Access, Speed, Caching, Workers, Rules, Network, Traffic, Stream, Custom Pages, Apps, Scrape Shield, and buttons for "+ Add site", "Support", and "English (US)".

A few more steps are required to complete your setup. Hide

✓ Some of your DNS only records are exposing IPs that are proxied through Cloudflare. Make sure to proxy all A, AAAA, and CNAME records pointing to proxied records to avoid exposing your origin IP.

DNS management for [REDACTED]

Type	Name	Target	TTL	Proxy status
CNAME	admin	[REDACTED]	Auto	Proxied

Cancel Save

## 4.2 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.

CLOUDFLARE [REDACTED]

+ Add site Support English (US)

[Overview](#) [Analytics](#) [DNS](#) [SSL/TLS](#) [Firewall](#) [Access](#) [Speed](#) [Caching](#) [Workers](#) [Rules](#) [Network](#) [Traffic](#) [Stream](#) [Custom Pages](#) [Apps](#) [Scrape Shield](#)

Overview Edge Certificates Client Certificates Origin Server Custom Hostnames

**Your SSL/TLS encryption mode is Full (strict)**

This setting was last changed a few seconds ago

- Off (not secure)  ⓘ No encryption applied
- Flexible  Encrypts traffic between the browser and Cloudflare
- Full  Encrypts end-to-end, using a self signed certificate on the server
- Full (strict)  Encrypts end-to-end, but requires a trusted CA or Cloudflare Origin CA certificate on the server

[Learn more about End-to-end encryption with Cloudflare](#)

[API ▶](#) [Help ▶](#)

**SSL/TLS Recommender** Beta

To check if your website can use a more secure SSL/TLS mode, enable the SSL/TLS Recommender. Receive an email with Cloudflare's recommendation.

[Help ▶](#)

CLOUDFLARE [REDACTED]

+ Add site Support English (US)

[Overview](#) [Analytics](#) [DNS](#) [SSL/TLS](#) [Firewall](#) [Access](#) [Speed](#) [Caching](#) [Workers](#) [Rules](#) [Network](#) [Traffic](#) [Stream](#) [Custom Pages](#) [Apps](#) [Scrape Shield](#)

Overview Edge Certificates Client Certificates [Origin Server](#) Custom Hostnames

**Origin Certificates**

Generate a free TLS certificate signed by Cloudflare to install on your origin server.

Origin Certificates are only valid for encryption between Cloudflare and your origin server.

[Create Certificate](#)

Hosts	Expires On
No Certificates.	

[Help ▶](#)

**Authenticated Origin Pulls**

TLS client certificate presented for authentication on origin pull.

[Help ▶](#)

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

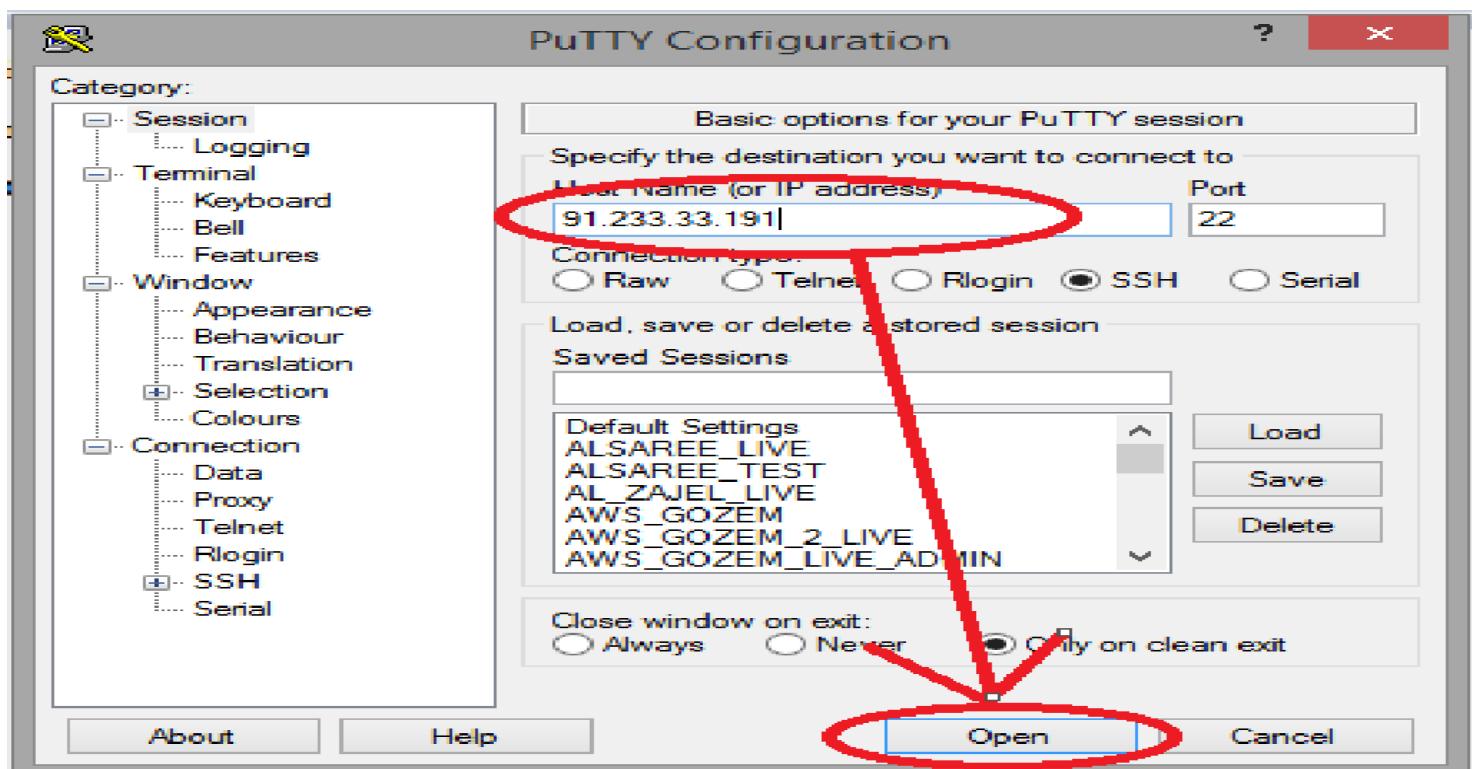
We will be Using putty for establishing SSH connection between our system

and server.

What is PuTTY ?

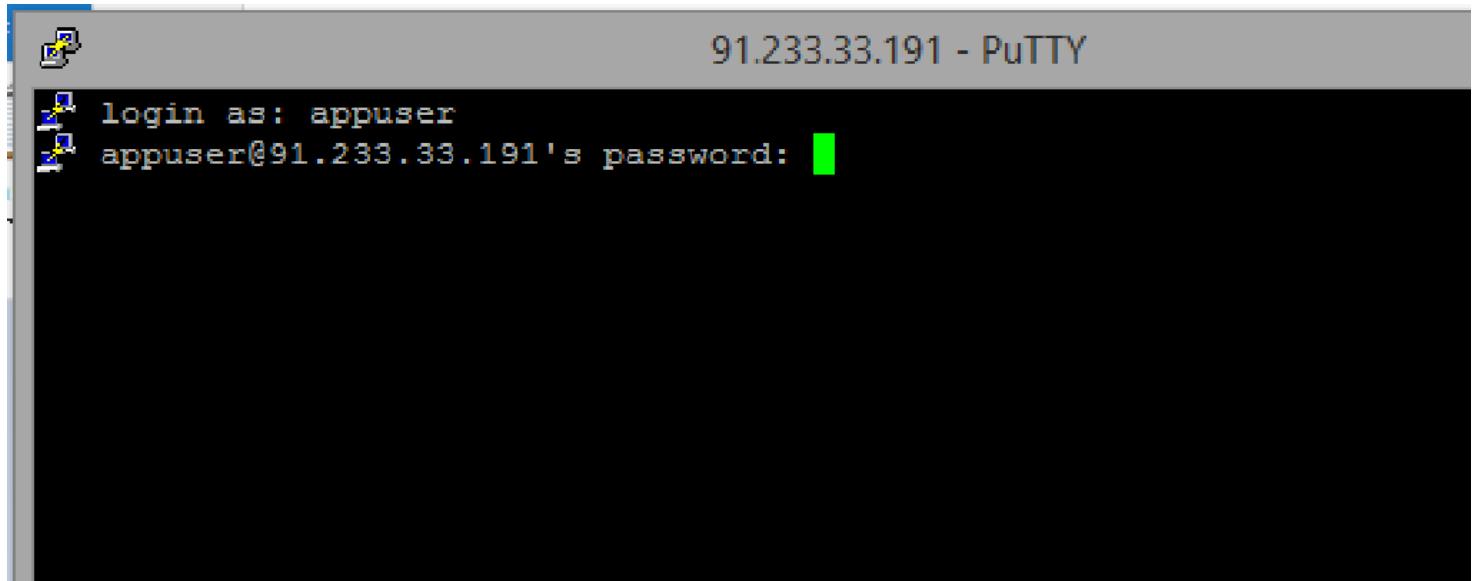
PuTTY is a free and open-source terminal emulator, serial console, and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. The name "PuTTY" has no official meaning.in short, it use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

Step 1 : Open PuTTY using ip



## Step 2 : Enter username and password in the PuTTY console :

(if you connect with the key and ip then no need to provide a user name and password at that time hostname be like : username@ip  
for ex : ubuntu@91.233.33.191 or root@91.233.33.191)



## Step 3 : Perform the below steps to install all dependancies or important server domain:

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

```
sudo chown <hostname>  
  
hostname = root(for digitalocean domain) /  
  
ubuntu(mostly in all) /  
  
ur root name
```

## Step 1 : Installing NodeJS Server by PPA Method (version 16)

```
sudo apt-get install python-software-properties  
curl -sL https://deb.nodesource.com/setup\_16.x | sudo -E bash
```

```
sudo apt-get install nodejs
```

(Reference : <https://tecatadmin.net/install-latest-nodejs-npm-on-ubuntu/> )(always  
Install LTS release)

## Step 2 : Installing NGINX Server

```
sudo apt-get update
```

```
sudo apt-get install nginx
```

```
sudo ufw app list
```

```
sudo ufw allow 'Nginx Full'
```

```
systemctl status nginx
```

```
sudo systemctl start nginx
```

(Reference : <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-16-04> )

## Step 3 : Installing MongoDB Server (4.4)

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add
```

```
sudo apt-get install gnupg
```

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```

---

```
sudo apt-get update
```

Then select the appropriate tab below based on the result:

- `systemd` - select the **systemd (systemctl)** tab below.
- `init` - select the **System V Init (service)** tab below.

```
sudo systemctl start mongod
```

If you receive an error similar to the following when starting mongod:

---

```
Failed to start mongod.service: Unit mongod.service not
Found.
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl status mongod
```

```
sudo systemctl start mongod
```

(Reference :

<https://www.mongodb.com/docs/v4.4/tutorial/install-mongodb-on-ubuntu/>)

## Step-4 : Installing Redis Session Management Server

```
sudo apt-get update
```

```
sudo apt-get install build-essential
```

```
sudo apt-get install tcl 8.6
```

```
wget https://download.redis.io/releases/redis-7.0.0.tar.gz
```

```
tar xzf redis-7.0.0.tar.gz
```

```
cd redis-7.0.0
```

```
make
```

```
make test
```

```
sudo make install
```

```
cd utils
```

```
sudo nano ./install_server.sh
```

**Comment it →**

```
#bail if this system is managed by systemd
#_pid_1_exe=$(readlink -f /proc/1/exe)
#if [ "${_pid_1_exe##*/}" = systemd ]
#then
# echo "This systems seems to use systemd."
# echo "Please take a look at the provided example service unit files in the
directory, and adept and install them. sorry!"
# exit 1
#fi
```

```
sudo ./install_server.sh
then (press enter 5 times)
```

<https://stackoverflow.com/questions/61694459/installing-redis-use-install-server-s>

```
sudo service redis start  
----> if not installed command this 2 line  
1> sudo apt-get install redis-server  
2> sudo service redis-server status
```

## Step-5 : Installing Git & Checkout Code

```
sudo apt-get install git  
(command to give permission)  
sudo chmod -R 777 /var/www/html  
Change path :  
cd /var/www/html (for ubuntu)  
cd /root (for root )
```

```
git clone <Your URL>  
(ex.: git clone https://gitlab.com/elluminatiEber/eberBackend.git)
```

## Step-6 : Installing required modules & Configure database for production environment

```
cd backend (if root, else go to appropriate path)
```

```
sudo chmod -R 777 /var/www/html/ OR  
sudo chmod -R 777 /root
```

```
rm -rf package-lock.json
```

```
sudo npm install
```

```
sudo npm install nodemon -g
```

```
sudo npm i pm2 -g
```

```
goto FileZilla and change DB name
```

```
//CONFIGURE DATABASE NAME IN PRODUCTION AND DEVELOPMENT ENVIRONMENT  
sudo nano /root/eberBackend/config/env/production.js (for root)  
sudo nano /var/www/html/backend/config/env/production.js (for ubuntu)
```

```
//Edit below line with your database name db:  
'mongodb://localhost:27017/<YOUR_DATABASE_NAME>'
```

```
(permission of the folder if needed :)
```

```
sudo chmod -R 777 /var/www/html/web/new_user_panel/node_modules
```

```
sudo npm install socket.io-redis
```

```
nodemon server.js
```

## Step-7 : Configure your NGINX Server config file

```
sudo nano /etc/nginx/sites-available/default
```

(default file is use to final configuration of our domain to ip we are applying ssl in this file for security of our domain.

here, we are adding two server function one for admin panel and another one is for user panel add your domain in this and related port in the below file and copy it in the default file.)

//Add below lines in the file

```
server {  
    listen 80;  
    server_name localhost;  
    root <path to server file>;  
  
    #Load configuration files for the default server block.  
    include /etc/nginx/default.d/*.conf;  
  
    location / {  
        proxy_pass http://localhost:5000;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
    }  
  
    error_page 404 /404.html;  
    location = /errorPage.html {  
    }  
}
```

( **ctrl+o** to write out the file; **ctrl+x** to exit the file; then **y** to save the content and give **enter** )

**( Most IMP : STOP NGINX SERVER  
sudo systemctl stop nginx )**

## Step-8 : Removing SSL comments from NGINX config file and restarting NGINX

```
sudo nano /etc/nginx/sites-available/default
```

//File after removing comments

```
server {
    server_name eberdeveloper.appemporio.net;
    location / {
        proxy_pass http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    listen 443 ssl;
    underscores_in_headers on;
    ssl_certificate /var/www/html/eber/ssl/eber_origin.pem;
    ssl_certificate_key /var/www/html/eber/ssl/eber_private.pem;

    location /gmapsapi {
        proxy_pass https://maps.googleapis.com/;
        proxy_pass_request_body on;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    error_page 404 /404.html;
        location = /errorPage.html {
    }

    if ($scheme = http) {
        return 301 https://$server_name$request_uri;
    }
}

//Save the file above and restart NGINX Server
sudo systemctl restart nginx
**OR**
sudo systemctl start nginx
```

## Step 9 : Start Forever Server(mostly we are using PM2):

change Path :

```
cd /var/www/html/<path to server.js of admin> (for ubuntu)
cd /root /<path to server.js of user> (for root)
```

```
forever start server.js  
**OR**  
if you install pm2 then;  
pm2 restart server.js
```

```
cd /var/www/html/<path to server.js of user> (for ubuntu)  
cd /root /<path to server.js of user> (for root)  
forever start server.js  
**OR**  
if you install pm2 then;  
pm2 restart server.js
```

#### Step 10 : Add database from Setting\_data folder :

```
cd /backend/settingsdata
```

```
node initial_data.js
```

After few second type ctrl + c to exit

```
nodemon server.js
```

#### Step 11: Once its start on the domain then change api keys :

first of all go to the gitlab acc of client and enable the api keys:

for backend ;

1) Road map API key

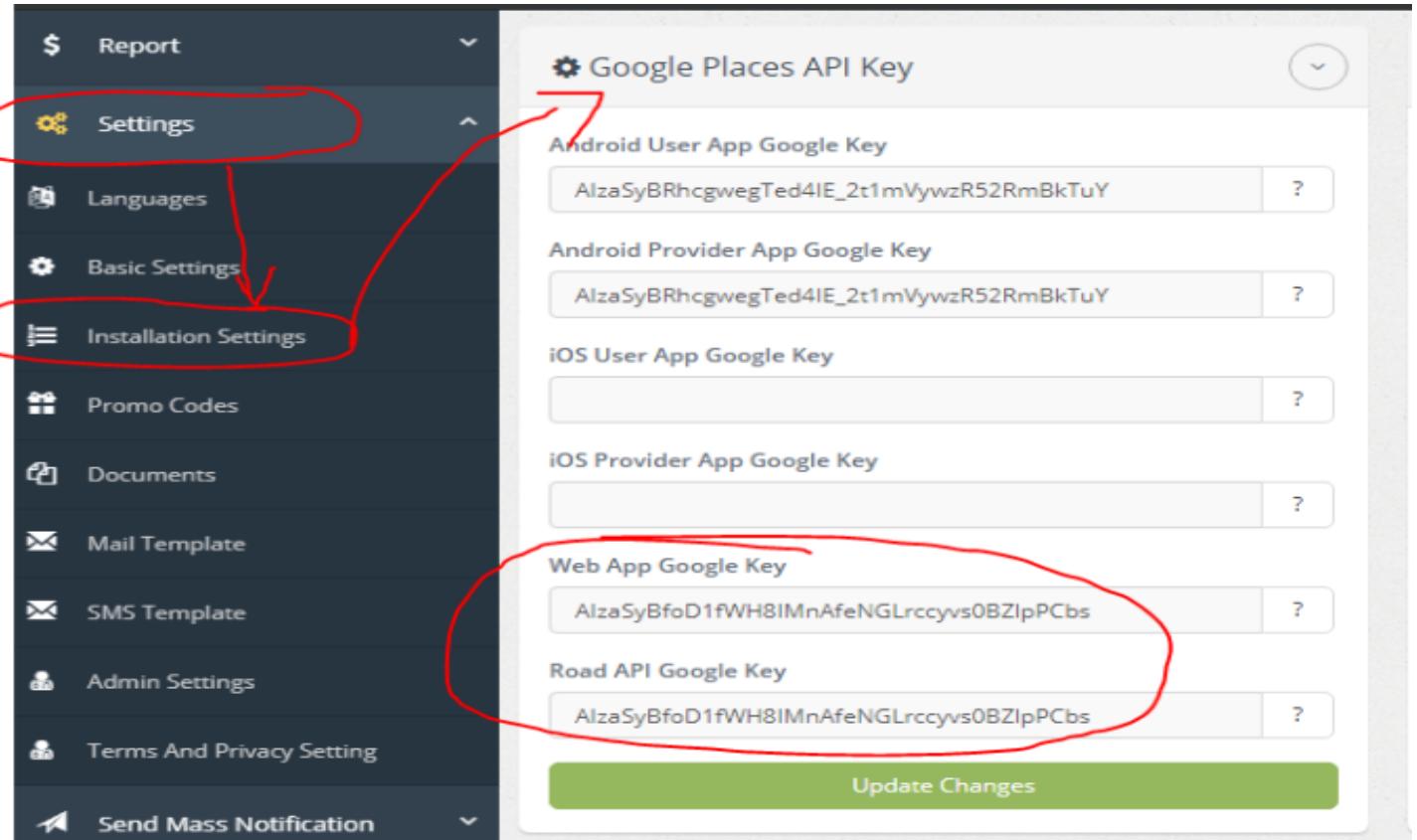
2)JavaScript API key

or you can do all api enable if you want to do ;)

-> Add Browser Key in the;

admin panel -> settings -> installation settings ->

1. Web App google key
2. Road Map google key



# EXTRA :

sudo zip -r web.zip web/

### Step 12 : connecting to db

go in to the puttygen and generate the private key public key and db key to use db.  
go to the robo3t .

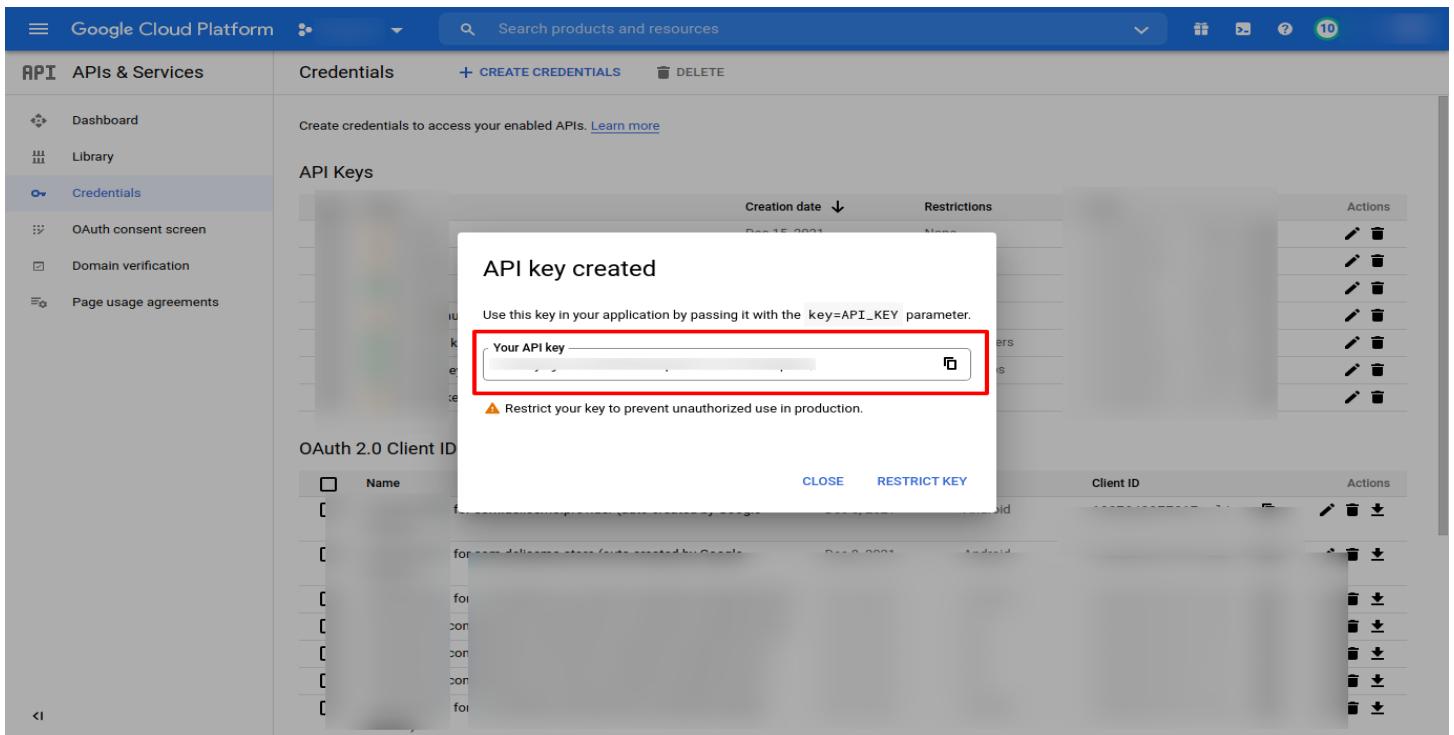
add ssh connection and enter details relate then test and connect.

```
twilio ml :  
twilio bin ->  
<Response>  
<Dial>  
<Number>{to}</Number>  
</Dial>  
</Response>
```

## 6). Google API key

- 1) Open Google cloud console.
- 2) Copy that API key and save it in a text file and use it in Your project
- 3) Set this API Key in the given files.

The screenshot shows the Google Cloud Platform interface for managing credentials. The left sidebar has 'APIs & Services' selected under 'Dashboard'. The main area is titled 'Credentials' with a sub-section 'API Keys'. A red box highlights the '+ CREATE CREDENTIALS' button at the top of the API Keys section. Another red box highlights the 'API key' option in the list of credential types. The 'API key' section includes a description: 'Identifies your project using a simple API key to check quota and access'. Below this are other options: 'OAuth client ID' (with a description 'Requests user consent so your app can access the user's data'), 'Service account' (with a description 'Enables server-to-server, app-level authentication using robot accounts'), and 'Help me choose' (with a description 'Asks a few questions to help you decide which type of credential to use'). To the right, there is a table for managing API keys, showing columns for 'Restrictions', 'Key', and 'Actions'. At the bottom, there is a section for 'OAuth 2.0 Client IDs' with a table showing columns for 'Name', 'Creation date', 'Type', and 'Client ID'.



#### 4) --Google Api key----

-> Turn on this apis

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

\*Please Restrict the Google API Key

## 7). Social login using Gmail

1) Open google console and select your project.

2) Select Web client auto service.

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

The screenshot shows the Google Cloud Platform API & Services Credentials page. The left sidebar has options like Dashboard, Library, Credentials (which is selected and highlighted with a red box), OAuth consent screen, Domain verification, and Page usage agreements. The main area shows three sections: API Keys, OAuth 2.0 Client IDs, and Service Accounts. The OAuth 2.0 Client IDs section contains a table with columns: Name, Creation date (sorted by creation date), Restrictions, Key, and Actions. One row is highlighted with a red box and shows the details: Name is "Web client (auto created by Google Service)", Creation date is "NOV 14, 2021", Type is "Web application", and the Key is partially visible as "10...". The Service Accounts section below it has a table with columns: Email, Name (sorted by name), and Actions. A "Manage service accounts" link is also present.

3) Add your domain name to URI

4) And use Client Id & Secret in your project Database.

5) you have to add client\_id in constant.json collection.

The screenshot shows the Google Cloud Platform interface for managing APIs & Services. A new OAuth 2.0 client is being created. The 'Name' field is set to 'Web client (auto created by Google Service)'. The 'Client ID' and 'Client secret' fields are highlighted with a red box. Below these, a note states: 'The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.' A list of authorized JavaScript origins is shown, also highlighted with a red box, including various local host and custom domain entries. The 'Authorized redirect URIs' section is also visible, with one entry highlighted with a red box. At the bottom right, there are 'SAVE' and 'CANCEL' buttons.

## 8). Facebook Login.

1) Open Facebook developer console.

2) Select your project app.

The screenshot shows the Facebook Developer Console interface. At the top, there's a navigation bar with links for Docs, Tools, Support, My Apps (with 8 notifications), and a search bar. Below the navigation is an orange banner with a message about email communication requirements. The main area is titled 'Admin Apps' and contains a grid of app cards. A specific app card is highlighted with a red box. The card for 'YoVoy' has its details visible: App ID: 716569672127622, Type: Business. Other cards include 'Eber Provider' (App ID: 126933474457184, Type: Consumer) and 'Eber' (App ID: 1861175364114058, Type: Consumer). On the left, there's a sidebar with 'Filter by' options (All Apps selected, Archived, Data Use Checkup with 99+ notifications), 'Required Actions' (none listed), and a 'Create App' button. There are also 'Select All' and 'Start Checkup' buttons at the top right of the app grid.

3) Click on FB Login Setting.

**Meta for Developers**

Docs Tools Support My Apps 8 Search developer documentation Ellu Minati Help

App ID: 4 App Type: Consumer App Mode: Development Live

**Required Actions**

You don't have any required action items to display. If any of your apps need immediate attention in the future, an item will show here.

**Application Rate Limit**

0% of limit used View Details 100% Remaining

**User Rate Limit**

0 Users throttled

**Last Mobile App Installs**

Dec 10 at 1:05 PM Install registered on iOS Dec 8 at 1:19 PM Install registered on Android

**Add Products to Your App**

## 4) Add Domains of your project

**Meta for Developers**

Docs Tools Support My Apps 8 Search developer documentation Ellu Minati Help

App ID: 4 App Type: Consumer App Mode: Development Live

**Client OAuth Settings**

**Client OAuth Login** Yes Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

**Web OAuth Login** Yes Enables web-based Client OAuth Login. [?] Enforce HTTPS Yes Enforce the use of HTTPS for Redirect URIs and the JavaScript SDK. Strongly recommended. [?]

**Force Web OAuth Reauthentication** Yes When on, prompts people to enter their Facebook password in order to log in on the web. [?]

**Embedded Browser OAuth Login** Yes Enable webview Redirect URIs for Client OAuth Login. [?]

**Use Strict Mode for Redirect URIs** Yes Only allow redirects that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]

**Valid OAuth Redirect URIs** A manually specified redirect\_uri used with Login on the web must exactly match one of the URIs listed here. This list is also used by the JavaScript SDK for in-app browsers that suppress popups. [?]

https:// https://s https://n https:// https://localhost:4202/

**Login from Devices** Yes Enables the OAuth client login flow for devices like a smart TV. [?]

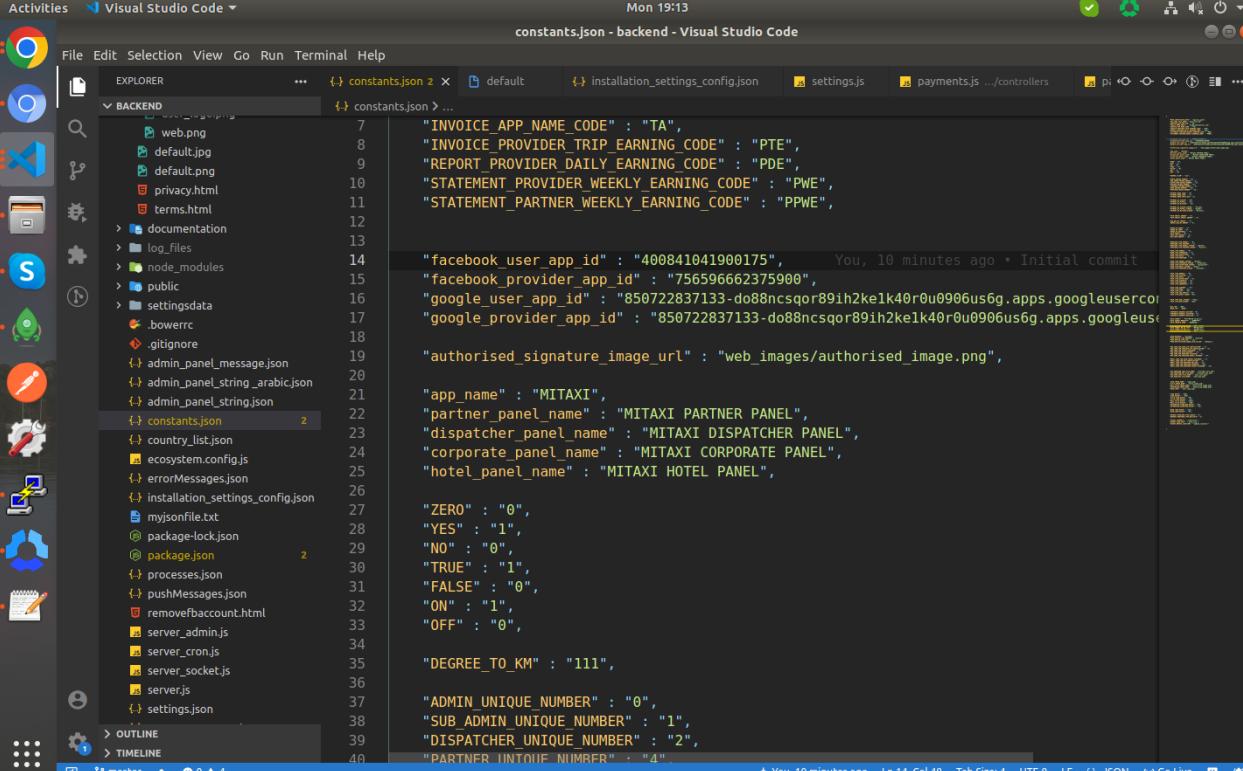
**Login with the JavaScript SDK** Yes Enables Login and signed-in functionality with the JavaScript SDK. [?]

**Allowed Domains for the JavaScript SDK** Login and signed-in functionality of the JavaScript SDK will only be available on these domains. [?]

https://localhost:4201/ https://i https://n https://m/ https://localhost:4202/

**Save changes**

## 5) Add user fb id and driver fb id in constant.json file



```
constants.json - backend - Visual Studio Code
Mon 19:13
File Edit Selection View Go Run Terminal Help
EXPLORER constants.json 2 default installation_settings_config.json settings.js payments.js .../controllers pi ...
BACKEND constants.json ...
    "INVOICE_APP_NAME_CODE" : "TA",
    "INVOICE_PROVIDER_TRIP_EARNING_CODE" : "PTE",
    "REPORT_PROVIDER_DAILY_EARNING_CODE" : "PDE",
    "STATEMENT_PROVIDER_WEEKLY_EARNING_CODE" : "PWE",
    "STATEMENT_PARTNER_WEEKLY_EARNING_CODE" : "PPWE",
    "facebook_user_app_id" : "400841041900175", You, 10 minutes ago * Initial commit
    "facebook_provider_app_id" : "756596662375900",
    "google_user_app_id" : "850722837133-do88ncsqr89ih2ke1k40r0u0906us6g.apps.googleusercontent.com"
    "google_provider_app_id" : "850722837133-do88ncsqr89ih2ke1k40r0u0906us6g.apps.googleusercontent.com
    "authorised_signature_image_url" : "web_images/authorised_image.png",
    "app_name" : "MITAXI",
    "partner_panel_name" : "MITAXI PARTNER PANEL",
    "dispatcher_panel_name" : "MITAXI DISPATCHER PANEL",
    "corporate_panel_name" : "MITAXI CORPORATE PANEL",
    "hotel_panel_name" : "MITAXI HOTEL PANEL",
    "ZERO" : "0",
    "YES" : "1",
    "NO" : "0",
    "TRUE" : "1",
    "FALSE" : "0",
    "ON" : "1",
    "OFF" : "0",
    "DEGREE_TO_KM" : "111",
    "ADMIN_UNIQUE_NUMBER" : "0",
    "SUB_ADMIN_UNIQUE_NUMBER" : "1",
    "DISPATCHER_UNIQUE_NUMBER" : "2",
    "PARTNER_UNIQUE_NUMBER" : "4"
}
You, 10 minutes ago Ln 14, Col 48 Tab Size: 4 UTF-8 LF () JSON Go Live
```

## 9). Security Groups in AWS.

### 1) Why security groups are 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) In the navigation pane, choose Security Groups.

### 3) Choose your instance and create a security group.

The screenshot shows the AWS Management Console with the Services menu open. The 'Network & Security' section is selected, and the 'Security Groups' link is highlighted with a red box. The main content area displays the 'Security Groups (1/3)' page, showing three security groups: 'launch-wizard-1' (selected), 'default', and 'launch-wizard-2'. The 'Create security group' button is also highlighted with a red box.

Name	Security group ID	Security group name	VPC ID	Description
-	-	laun... d-1	-	laun... launch-wizard-1 create...
-	-	def... t	-	def... default VPC security gr...
-	-	laun... -2	-	laun... launch-wizard-2 create...

**Details** | Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer [Run Reachability Analyzer](#)

**Details**

Security group name	Security group ID	Description	VPC ID
laun... d-1	-	-	-
Owner	Inbound rules count	Outbound rules count	
-	17 Permission entries	1 Permission entry	

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

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

### Basic details

Security group name <a href="#">Info</a>	<input type="text" value="MyWebServerGroup"/>
Name cannot be edited after creation.	
Description <a href="#">Info</a>	<input type="text" value="Allows SSH access to developers"/>
VPC <a href="#">Info</a>	<input type="text" value="vpc-0000000000000000"/> <input type="button" value="X"/>

### Inbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Custom TCP	TCP	0	Custom <input type="button" value="Search"/> <input type="text" value="0.0.0.0/8"/>	<input type="button" value="Delete"/>
<input type="button" value="Add rule"/>				

### Outbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Feedback English (US) ▾				© 2021, Amazon Web Services, Inc. or its affiliates. <a href="#">Privacy</a> <a href="#">Terms</a> <a href="#">Cookie preferences</a>

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

### Basic details

<input type="text" value="S"/>	<input type="button" value="Search"/>
<ul style="list-style-type: none"> <li><input type="checkbox"/> SMTP</li> <li><input type="checkbox"/> DNS (UDP)</li> <li><input type="checkbox"/> DNS (TCP)</li> <li><input type="checkbox"/> HTTP</li> <li><input type="checkbox"/> POP3</li> <li><input type="checkbox"/> IMAP</li> <li><input type="checkbox"/> LDAP</li> <li><input checked="" type="checkbox"/> HTTPS</li> <li><input type="checkbox"/> SMB</li> <li><input type="checkbox"/> SMTPS</li> <li><input type="checkbox"/> IMAPS</li> <li><input type="checkbox"/> POP3S</li> <li><input type="checkbox"/> MSSQL</li> <li><input type="checkbox"/> HTTPS</li> </ul>	
<input type="button" value="Add rule"/>	<input type="button" value="Delete"/>

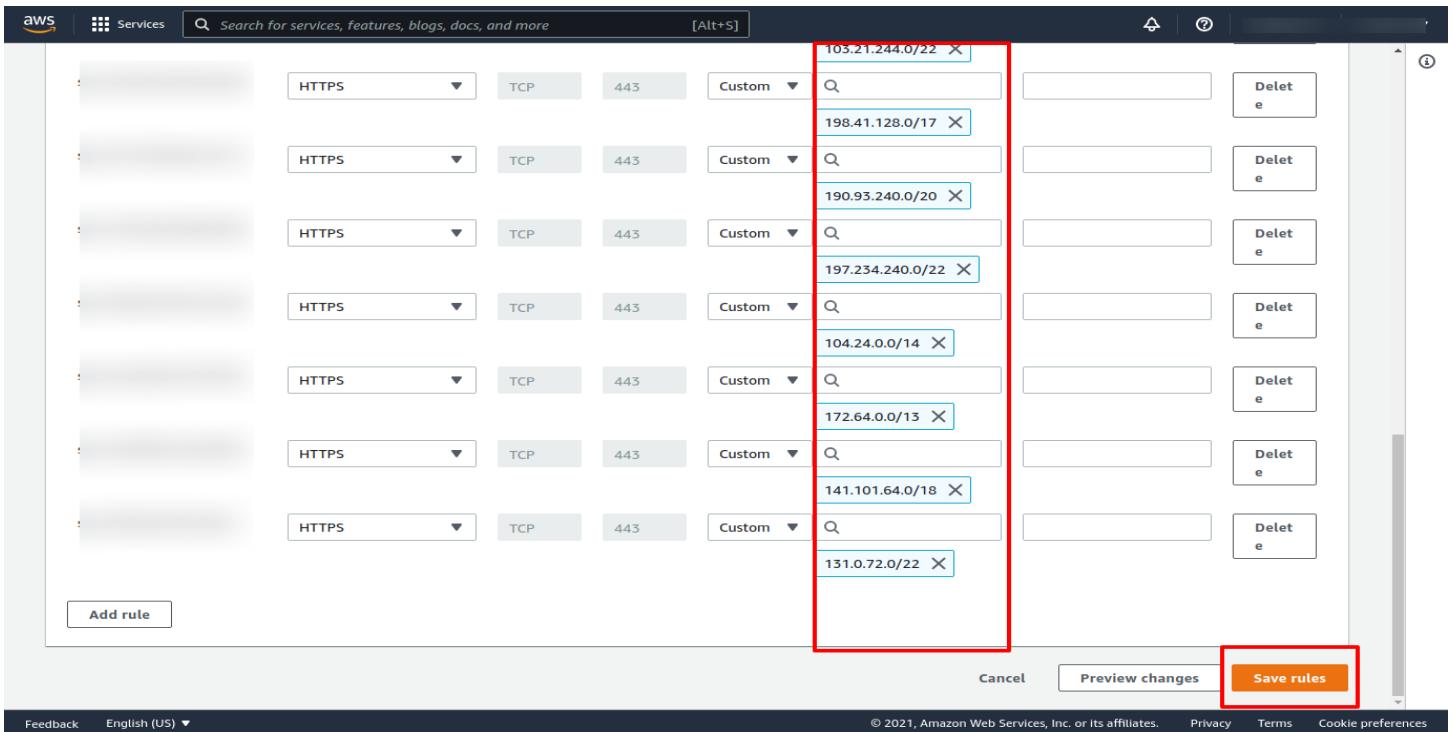
### Inbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
HTTPS	TCP	443	Custom <input type="button" value="Search"/> <input type="text" value="0.0.0.0/8"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>
<input type="button" value="Add rule"/>				

### Outbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Feedback English (US) ▾				© 2021, Amazon Web Services, Inc. or its affiliates. <a href="#">Privacy</a> <a href="#">Terms</a> <a href="#">Cookie preferences</a>

5) Add all custom IPv4 addresses with HTTPS, which are given in this link <https://www.cloudflare.com/en-gb/ips/>, and Save Rules.



## 10). 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.

### 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

2) 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

3) The following keys/file are removed that need to be configured from client account that needs to be push in the repository:

- app/ios\_push/push\_file.p8 - Will be added by admin from client account