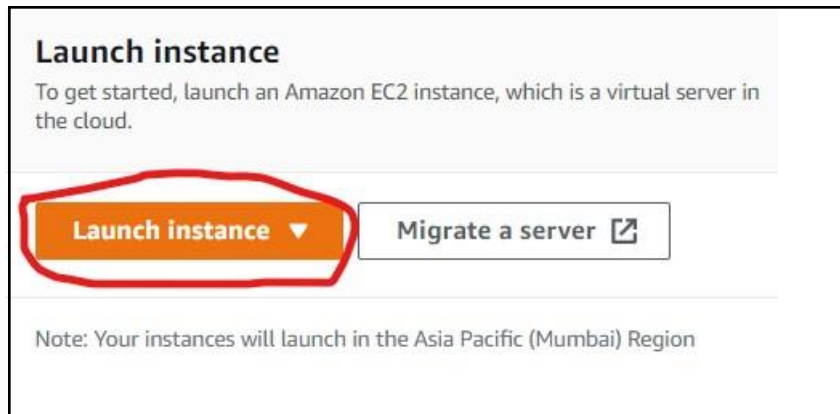
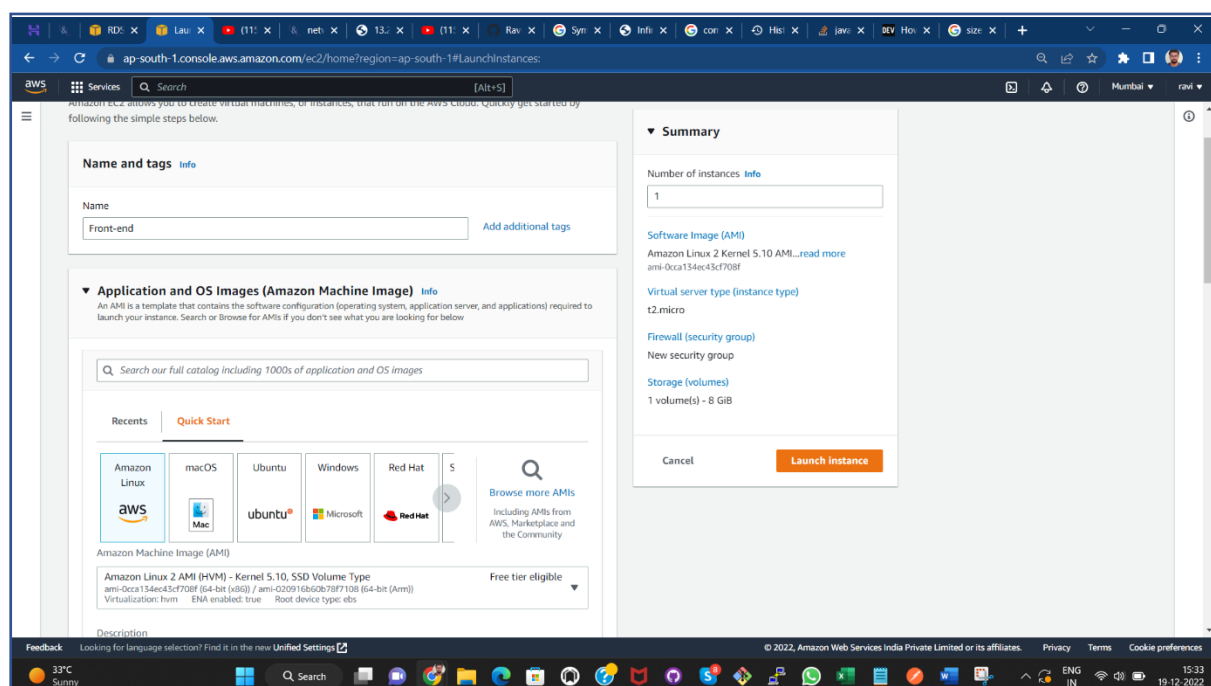


**Title:** Live Laravel back-end api(serve project) project in Aws Ec2-Instance.

Step1: Open Ec2 and launch instance using option “Launch Instance.”



Step2: Give name that instance what you want and select ‘Ubuntu’ OS in “Application and OS Images (Amazon Machine Image)” option.



Step3: Select Instance Type (Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and **give you the flexibility to choose the appropriate mix of resources for your applications.**) as per your website requirement, then select “key pair(login)” if you have one or create new one using “create new key pair” option.

- In some project there is t2.small or t2.medium is required, for now set it on t2.micro and later if there is cmd “php artisan db:seed” is not working and hang the process then just stop the ec2 change the instance from

t2.micro to t2.small or t2.medium and start ec2 then apply that cmd it's works fine.

The screenshot shows the AWS EC2 console configuration page. Under the 'Instance type' section, 't2.micro' is selected, which is 'Free tier eligible'. Details for t2.micro include: Family: t2, 1 vCPU, 1 GiB Memory. On-Demand Linux pricing: 0.0124 USD per Hour. On-Demand Windows pricing: 0.017 USD per Hour. A 'Compare instance types' link is available. The 'Key pair (login)' section explains that a key pair is used to securely connect to the instance. A dropdown menu for 'Key pair name - required' shows 'key2'. A 'Create new key pair' button with a refresh icon is also present.

- For how to create keypair see documentation of “Front-end quasar project live” after step3 in it.

Step4:

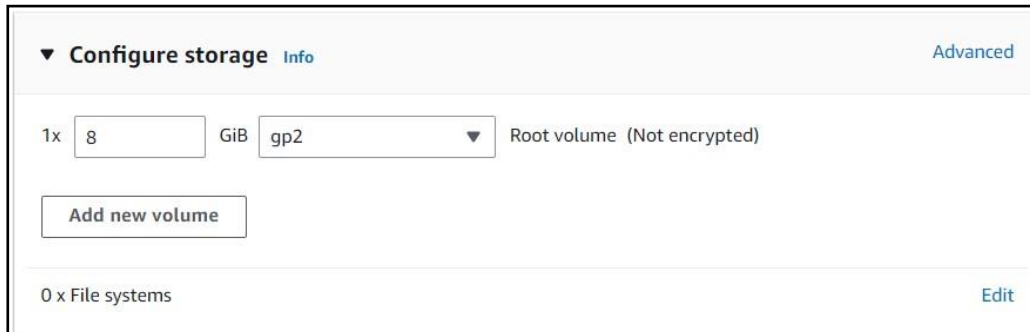
**Network setting> Firewall (security groups)** select “**Create security group**” if you don’t have one or if you already created security group select “**Select existing security group**” and then select the name of security group you created.

For “**Create security group**” option select check-box of “**Allow SSH traffic from, Allow HTTPS traffic from the internet, Allow HTTP traffic from the internet**”.

The screenshot shows the 'Network settings' section of the AWS EC2 console. It includes an 'Edit' button. Under 'Network', the VPC ID is 'vpc-03b5c60ef60ab7bd5'. Under 'Subnet', it says 'No preference (Default subnet in any availability zone)'. The 'Auto-assign public IP' option is 'Enable'. The 'Firewall (security groups)' section explains that a security group is a set of firewall rules. Two options are available: 'Create security group' (selected) and 'Select existing security group'. Below this, it states: 'We'll create a new security group called 'launch-wizard-3' with the following rules:'. Three rules are listed, all with checked checkboxes: 'Allow SSH traffic from' (Helps you connect to your instance, source: Anywhere 0.0.0.0/0), 'Allow HTTPS traffic from the internet' (To set up an endpoint, for example when creating a web server), and 'Allow HTTP traffic from the internet' (To set up an endpoint, for example when creating a web server).

Step5:

In **“Configure Storage”** select size of storage as per limitation of EBS volume type. (Here I take 8gb in **gb2** EBS volume type, which’s limitation is up to 30gb for free tier eligible account.)



▼ **Configure storage** [Info](#) [Advanced](#)

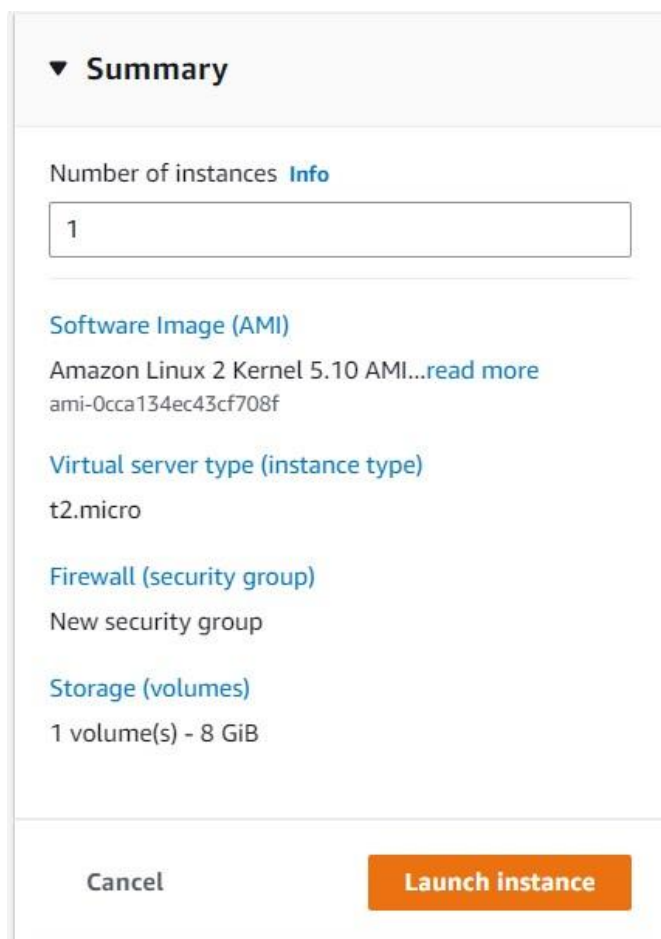
1x  GiB  ▼ Root volume (Not encrypted)

[Add new volume](#)

0 x File systems [Edit](#)

Step6:

Select **“Launch instance”** and launch it.



▼ **Summary**

Number of instances [Info](#)

[Software Image \(AMI\)](#)  
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-0cca134ec43cf708f

[Virtual server type \(instance type\)](#)  
t2.micro

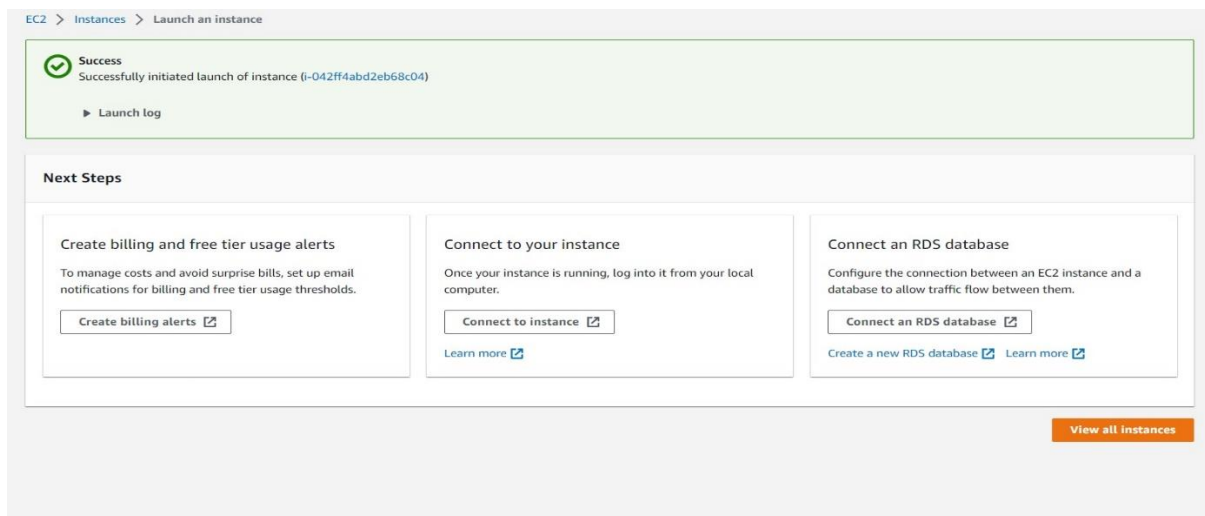
[Firewall \(security group\)](#)  
New security group

[Storage \(volumes\)](#)  
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#)

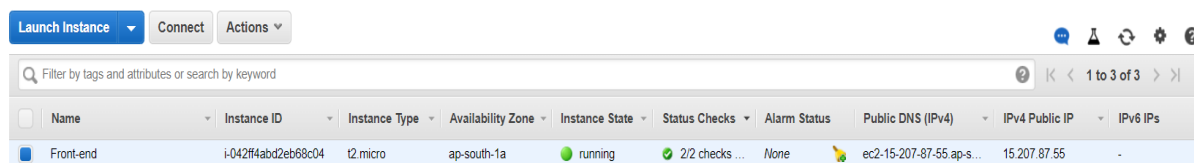
Step7:

Choose “View all Instances”



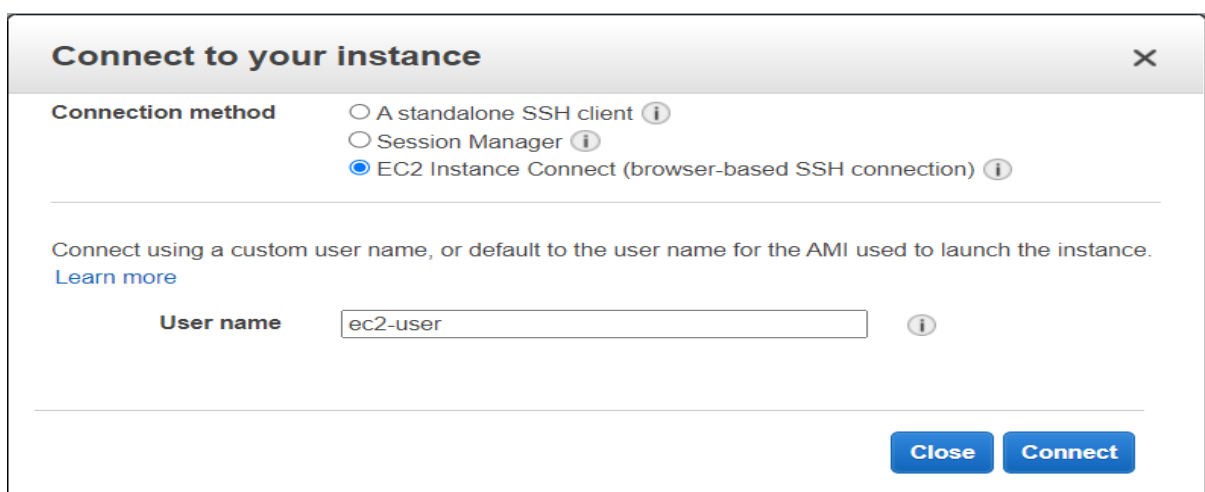
Step8:

Choose Option “Connect” after the “Status Checks” shows “2/2checks”.



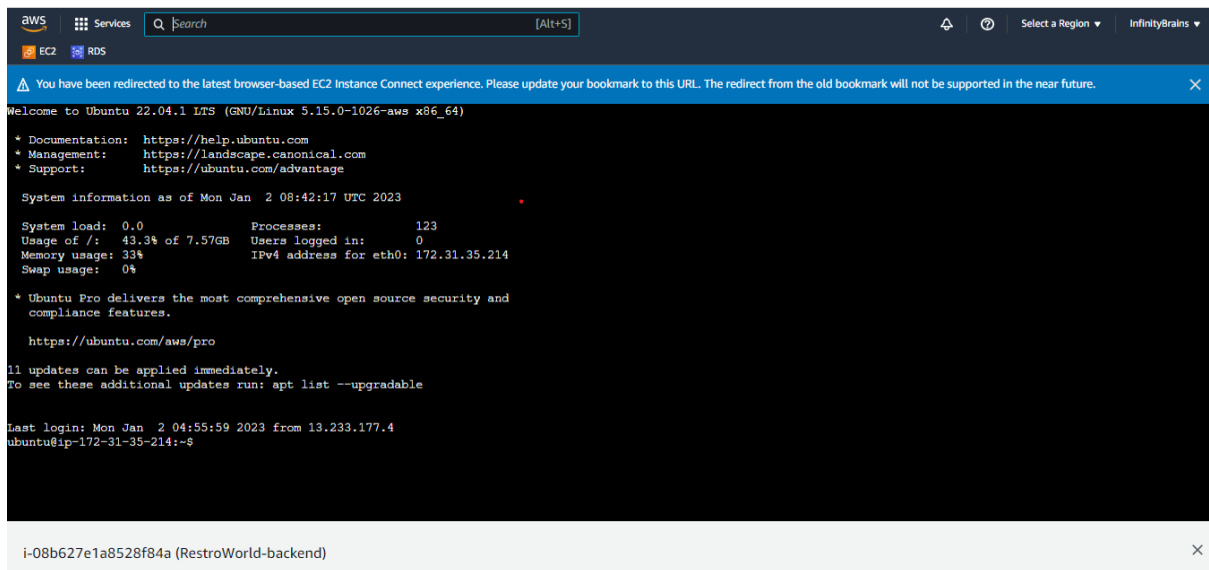
Step9:

Choose “EC2 Instance Connect (browser-based SSH connection)” and select “Connect” option.



Step 10:

This type of interface is open.



```
aws
Services
Q Search [Alt+S]
EC2 RDS
You have been redirected to the latest browser-based EC2 Instance Connect experience. Please update your bookmark to this URL. The redirect from the old bookmark will not be supported in the near future.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Mon Jan 2 08:42:17 UTC 2023
System load: 0.0 Processes: 123
Usage of /: 43.3% of 7.57GB Users logged in: 0
Memory usage: 33% IPv4 address for eth0: 172.31.35.214
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.
https://ubuntu.com/aws/pro
11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Last login: Mon Jan 2 04:55:59 2023 from 13.233.177.4
ubuntu@ip-172-31-35-214:~$
```

Now, first give following cmds :

`sudo su`

`sudo apt-get update`

- Active git using following cmd:

`git init`

- Create ssh key using following cmd in terminal

`ssh-keygen`

or

`ssh-keygen -t ed25519 -C "mailto:youremailaddress@domain.tld DAY-MONTH-YEAR" -f ~/.ssh/my_key`

- Use below shown cmd and then copy “id\_rsa.pub” text data which have been shown below:

`cat ~/.ssh/id_rsa.pub`

```

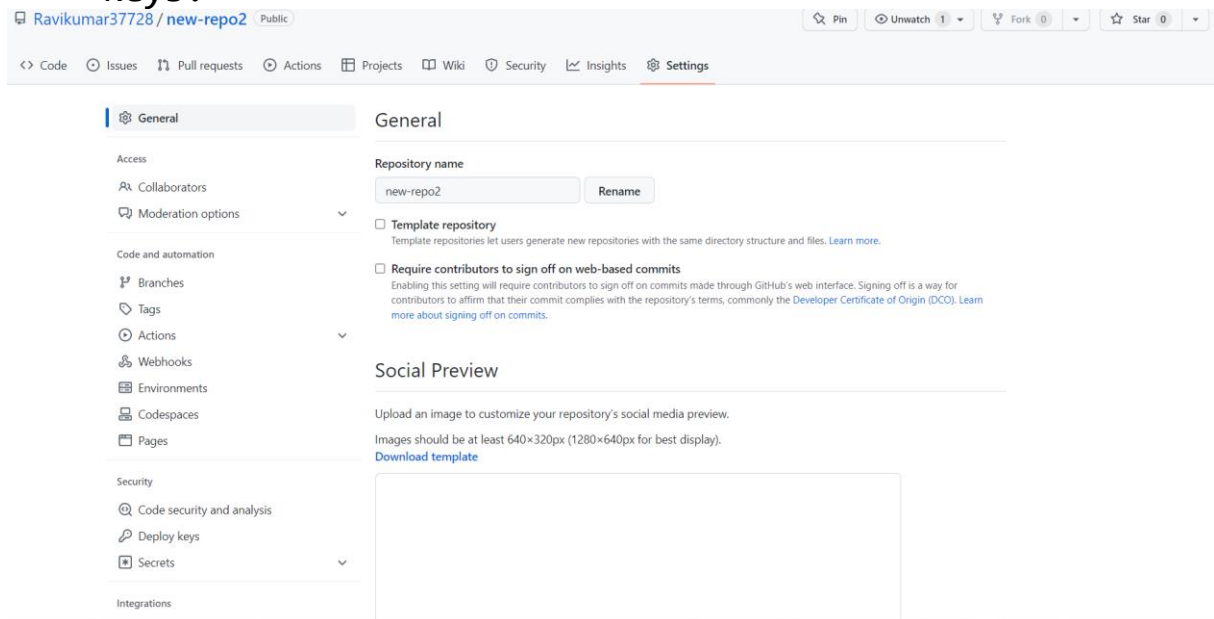
  _ _ _ _ _
 _/   ( _/   \
 _/_   _/   _/

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
ec2-user@ip-172-31-38-63 ~]$ sudo su
root@ip-172-31-38-63 ec2-user# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:gDPbSH2wZqRzrm6wt3vnx/85mAoyfDwRfC3+HXATbDU root@ip-172-31-38-63.ap-south-1.compute.internal
The key's randomart image is:
+-----[RSA 2048]-----+
|      . . . E .|
|      .   o   = .|
|    + o o . o + .|
|      B = oo . .|
|      . = So. . .|
|    . . * o . . .|
|    o+.++   o   |
|      . O+oo o . .|
|    +=+o.o.o.o .|
+-----[SHA256]-----+
root@ip-172-31-38-63 ec2-user# cat ~/.ssh/id_rsa.pub
cat: /root/root/.ssh/id_rsa.pub: No such file or directory
root@ip-172-31-38-63 ec2-user# cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCVwiiI3dckhQY0cB9eIzxnIKBw4pIeAooZDFhLHDA629EP2XzEjW9bLVU3pCbU0BKMmYSobbFUD4CZm9CSXp2hHoyZMLelZRmbxSYKvouRSORvHWkjfmjWNM30yILdYl1NF+CWE8sIZhoE+fidgkJIBjKUPiBcVrStqGlE/G2lXbj9KEzztblPWEzAdhfAuszsU/+1+U6ceASFVxht96v/xBm3r0ROyJyRk52n0o2v1EkeOL4gKj4xsiuMIz3JmI5IvuxBXD1TKRPsgf5SSE+o6o5gyYbEPK/2t1Dr3tqxrvDARtJY+fg69KdPPchNBVCNx9JiVmKSZNBv18gY1zE
root@ip-172-31-38-63.ap-south-1.compute.internal
root@ip-172-31-38-63 ec2-user#

```

- Paste it in git-hub source code repo.>settings>Deploy Keys.



- Paste copied text in there and named it. Then press “Add key”

Ravikumar37728 / new-repo2 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

### Deploy keys / Add new

Title: virtual-pc

Key: `ssh-rsa  
AAAAAB3NzaC1yc2EAAAADAQABAAQACwll3dchQY0cB9elzm1Kbw4pleAooZDFHIDa629EP2XzEjW9bLVU3pCbU0Bk  
MmYSObbPUd4CZm9CSXp2hHOyZMLelzRmxbSYKvourSOAvHWgfmwNM30yldY1INf+CWEr8slZhoE+fiDgkIIBkUPiBcV  
rsYqGIE/G2IXb9KZztb1PwEzAdhlAuszsU/+1+U6ceA5FVxht96v/x8m3r0ROyYjRk52n0o2v1EkeOL4gK4xsiuMlz3JMI5lvux  
BXD1TKrPsgf5S5E+o6o5gyYbZPk/2tldr3tgrVDArTjY+Fg69KbPPchNBvCNb9jVmk5ZNBv18qY1zH root@ip-172-31-38-  
63.ap-south-1.compute.internal`

Begin with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

☐ Allow write access

Can this key be used to push to this repository? Deploy keys always have pull access.

Add key

Your key is ssh key is now added in git hub repo.

Step 11:

- Give following cmd for activate github repo in ec2.

`ssh git@github.com`

- Type “yes” in connecting
- Now git hub is successfully connected to your ec2 insatncs.

Step 12:

- Search “Security Groups” open it.
- Select “Create Security group” give name it, add description for that if you want it.
- Select “Add rule” in Inbound rules, in Type select “MYSQL/Aurora” don’t change “Source” and after it select that security group, which is used by ec2 backend-api

EC2 > Security Groups > Create security group

### Create security group

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:

Description:

VPC:

**Inbound rules**


Type	Protocol	Port range	Source	Description - optional
MYSQL/Aurora	TCP	3306	Custom	sg-034f066bb005c1c62

**Outbound rules**

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

- Then select “Create security group”, your group is created.

- Now give it name for your ease of remembering it.

<input checked="" type="checkbox"/>	Testing backend api 	sg-034f066bb005c1c62	testing back end	vpc-0a1db80de583a
-------------------------------------	---	----------------------	------------------	-------------------

Step13:

- Search “Rds” and open it, select “Create database”.
- Select “Standard create” in first option, then select “MySQL” in Engine Option.


### Choose a database creation method [Info](#)


☒ **Standard create**  
 You set all of the configuration options, including ones for availability, security, backups, and maintenance.


☐ **Easy create**  
 Use recommended best-practice configurations. Some configuration options can be changed after the database is created.


### Engine options


Engine type [Info](#)


☐ Amazon Aurora  


☒ **MySQL**  



☐ MariaDB  


☐ PostgreSQL  


☐ Oracle  


☐ Microsoft SQL Server  


Engine Version

MySQL 8.0.31 

- In Templates select “Free tier”.



## Templates

Choose a sample template to meet your use case.

☐ **Production**  
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**  
This instance is intended for development use outside of a production environment.

☒ **Free tier**  
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.  
[Info](#)

- In “Settings” give name to the database in option “DB instance identifier”.

## Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

Testing Db

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in 'mydbinstance'). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

- Give “Master username” and “Master password” also note it down somewhere because we need it later.

### ▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

testing

1 to 16 alphanumeric characters. First character must be a letter.

Master password [Info](#)

\*\*\*\*\*

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password [Info](#)

\*\*\*\*\*

- Change “DB instance class” if it’s required for project or leave it as it is.
- Change “Storage” according to your project requirement for me its 20Gib.

## Storage

Storage type [Info](#)

General Purpose SSD (gp2)

Baseline performance determined by volume size

Allocated storage

20

GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

- Unchecked “Storage autoscaling”.

### Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

☐ Enable storage autoscaling

Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

- Select security group which one you created on step12 in “Existing VPC Security groups” after removing “default” security group.

Existing VPC security groups

Choose one or more options

testing back end X

- In “Additional configuration” give Initial Database name and also note it down somewhere for later it will be required.

### ▼ Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

#### Database options

Initial database name [Info](#)

testing1

If you do not specify a database name, Amazon RDS does not create a database.

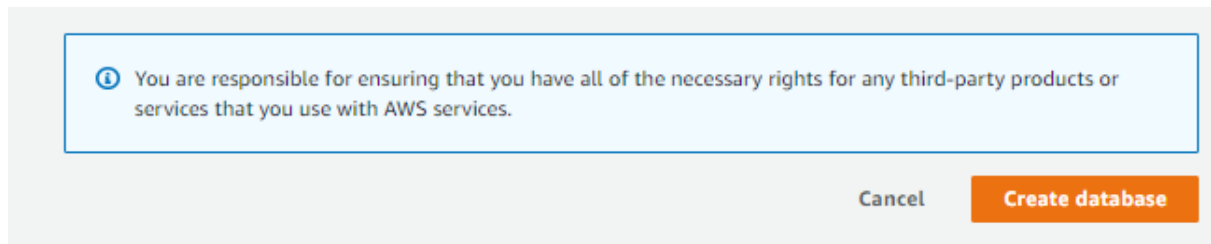
- If you want to store automated backup of your database then don't unchecked “Backup”, in my case I Unchecked it.

#### Backup

☐ Enable automated backups

Creates a point-in-time snapshot of your database

- Now, select “Create database” your data-base in Rds is created.



Step 14: Import Project in ec2 and connect it with Rds.

- Open Your ec2 instance
- Fire following cmds:

```
sudo su
```

```
sudo apt-get update
```

```
sudo apt-get install apache2
```

```
sudo service apache2 restart
```

```
sudo apt-get install mysql-server
```

```
sudo apt-get install php8.1 libapache2-mod-php8.1 php8.1-mysql
```

```
sudo add-apt-repository ppa:ondrej/php8.1
```

```
sudo apt-get install php8.1-mcrypt
```

```
sudo curl -O https://getcomposer.org/composer.phar
```

```
sudo mv composer.phar composer
```

```
sudo chmod +x composer
```

```
sudo mv composer /usr/local/bin
```

```
composer
```

```
sudo apt-get install php8.1-mbstring
```

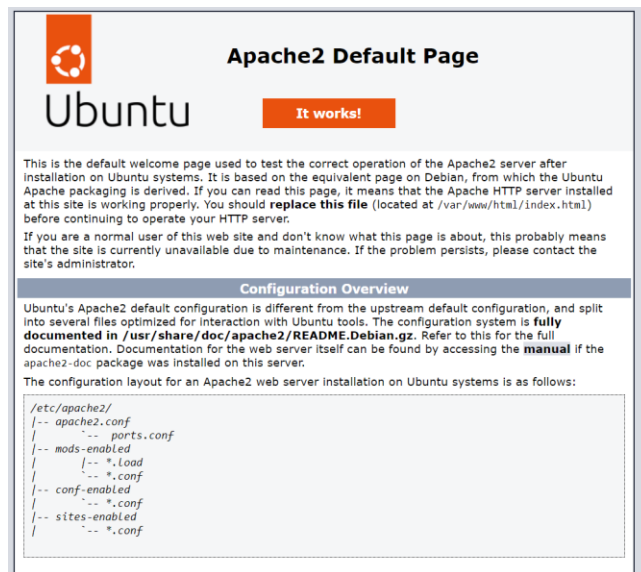
```
sudo apt-get install php8.1-xml
```

```
sudo apt-get install zip unzip
```

```
composer clearcache
```

```
cd /var/www/html
```

- Run your IPv4 address in browser this type of interface is shown.



- Now in your ec2 instance remove index.html file from location /var/www/html and import your working project there by follow following cmds:

`cd /var/www/html`

`rm -rf index.html`

`git init`

`git remote add origin (ssh link)`

e.g.

`git remote add origin git@github.com:Ravikumar37728/new-repo2.git`

- Now use following cmd for pull codes in ec2 on path var/www/html.

`git pull origin Master`

or

`git pull origin main`

**NOTE:** main or master or other branch is depended upon which branch code you want to pull from connected git-hub repo.

- Check your code is there or not by using cmd:

`ls` or `ls -la`

- Now give following cmd for installing and updating composer

`composer install --ignore-platform-reqs`

`composer update --ignore-platform-reqs`

- Change and add data in 000-default.conf for open it, use following cmd:  
`sudo nano /etc/apache2/sites-available/000-default.conf`
- Change "DocumentRoot" /var/www/html to  
"/var/www/html/index.php" or in some project it will be  
"/var/www/html/public" use one of it, by which one of it project is  
running (generally second one path is used when in postman,uploaded  
img's url gives blank screen in browser. )
- Also add given below data just above the </VirtualHost> in 000-default.conf.

<Directory "/var/www/html">

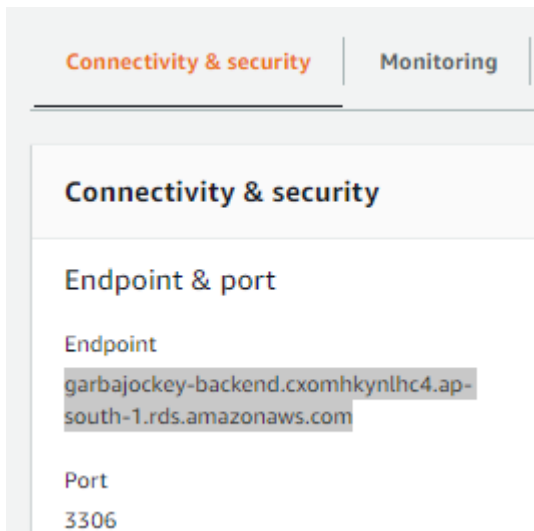
Options Indexes FollowSymLinks MultiViews

AllowOverride All

Require all granted

</Directory>

- After it restart apache 2 using cmd:  
`sudo service apache2 restart`
- Rename .env.example and named it .env using cmd:  
`mv .env.example .env`
- Fire following cmd:  
`php artisan cach:clear`  
`php artisan config:clear`  
`php artisan key:generate`
- Give read-wright permission to the storage and bootstrap  
`sudo chmod 777 -R bootstrap/`  
`sudo chmod 777 -R storage/`  
`sudo service apache2 restart`
- Open .env file and enter DB\_HOST, DB\_DATABASE, DB\_USERNAME,  
DB\_PASSWORD of Rds in there.
- For DB\_HOST ,open rds click on your db and open it init in "Connectivity &  
Security" copy the "Endpoint" link and paste it in DB\_Host



- In DB\_DATABASE paste name of data base which we save previously in step13,in DB\_USERNAME and DB\_PASSWORD give Master Username and Master Password which we save previously in step13,and save it. then again one time restart apache2.
- Fire following cmd:

```
nano /etc/php/8.1/apache2/php.ini
```

- Change three things in there by finding it in file using Ctrl+Q .  
upload\_max\_filesize = 512M  
post\_max\_size = 558M  
max\_execution\_time = 300
- Now fire following cmds:  
php artisan migrate:fresh  
php artisan passport:install  
php artisan key:generate  
php artisan storage:link  
php artisan route:cache  
php artisan route:clear  
php artisan db:seed
- Now, add one line in apache2.conf file for that fire cmd:  
sudo nano /etc/apache2/apache2.conf
- Find .htaccess using Ctrl+Q and add given below line on that place which is shown in given figure.  
SetEnvIf Authorization "(.\*)" HTTP\_AUTHORIZATION=\$1

```
AccessFileName .htaccess

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<FilesMatch "^\.ht">
    Require all denied
</FilesMatch>
SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1
```

- Restart your apache2  
`sudo service apache2 restart`
- Now run your IPv4 address if its redirect to Laravel page it's means it is live and give path it to api (e.g. 65.24.3.16/api or in some case it is 65.24.3.16/index.php/api) if its redirect to api that means it is working fine.