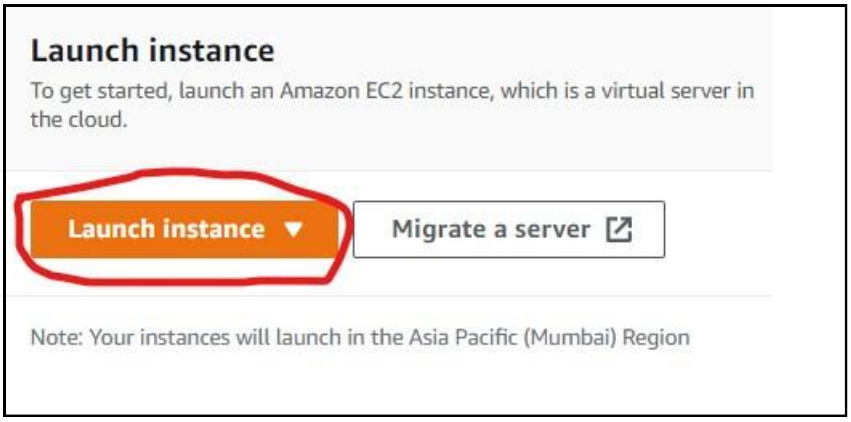
Title: Live Vue.js Front-end (serve project) project in Aws Ec2-Instance.

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

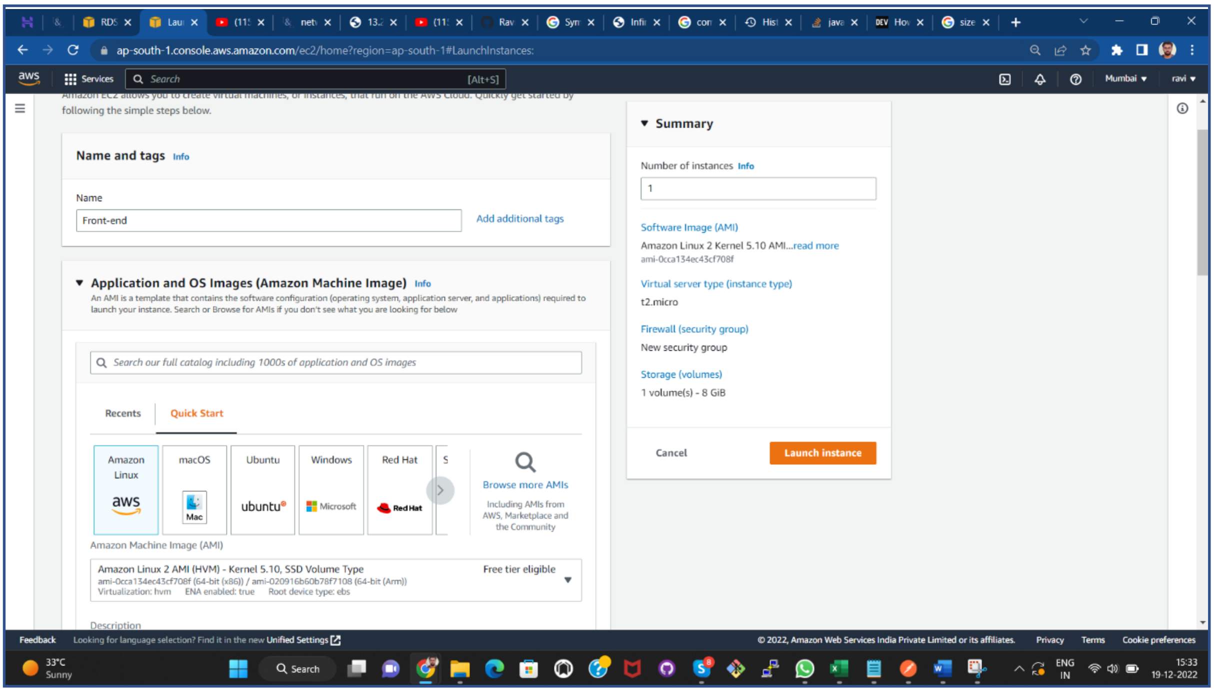


Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Step2: Give name that instance what you want and select 'Ubuntu' OS in "**Application and OS Images (Amazon Machine Image)**" option.



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

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: Front-end

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

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible
ami-0cca134ec43cf708f (64-bit (x86)) / ami-020916b60b78f7108 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances Info: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0cca134ec43cf708f

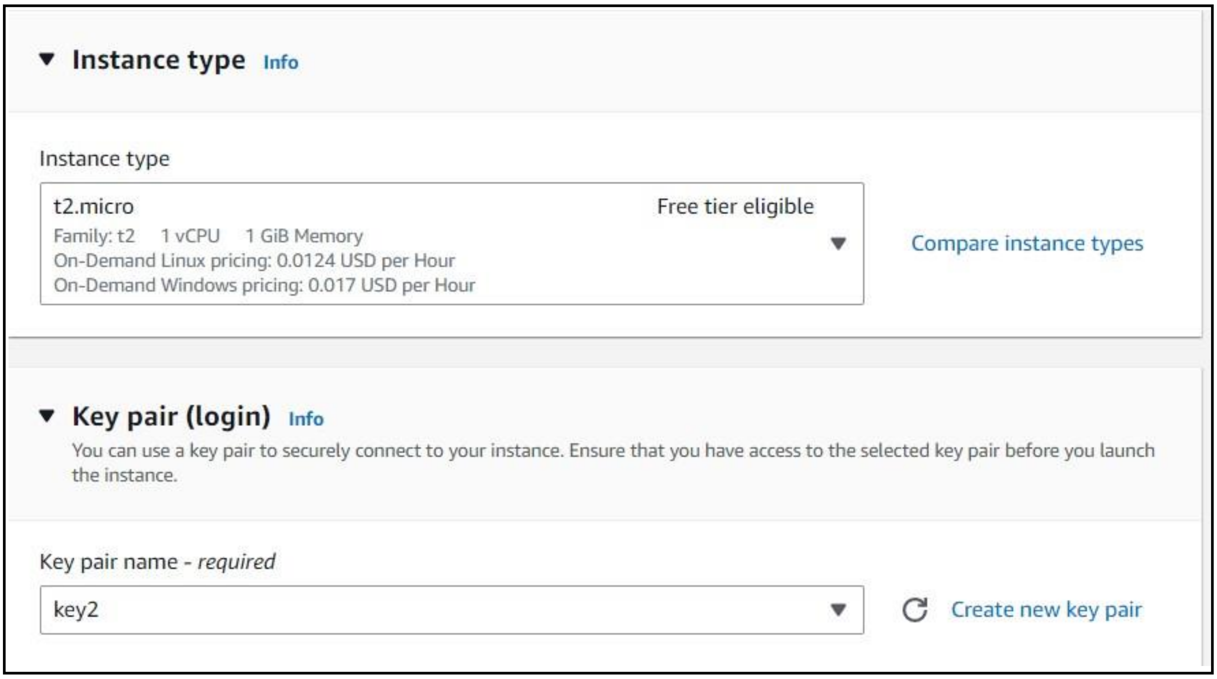
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

aws
Services
Search
[Alt+S]
Mumbai ▼
ravi ▼
Add additional tags
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Description
Cancel
Launch instance
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
33°C Sunny
15:33
19-12-2022

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.



▼ Instance type Info

Instance type

t2.micro Free tier eligible ▼
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

Compare instance types

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

key2 ▼

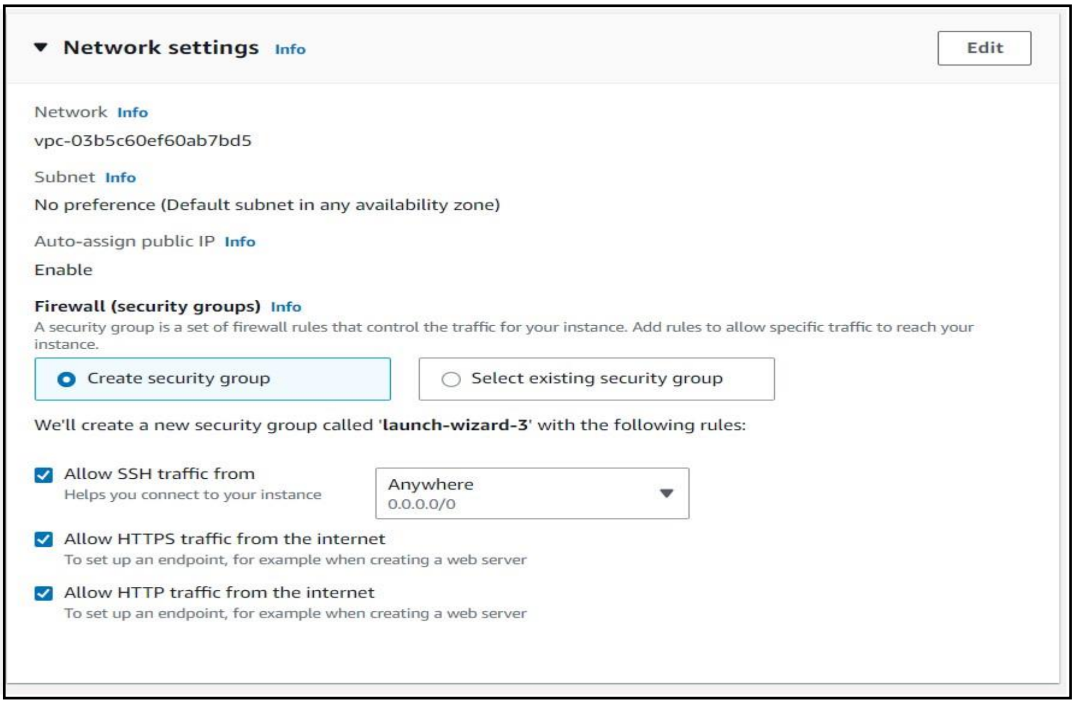
Create new key pair

- 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**”.



▼ Network settings Info Edit

Network Info
vpc-03b5c60ef60ab7bd5

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-3**' with the following rules:

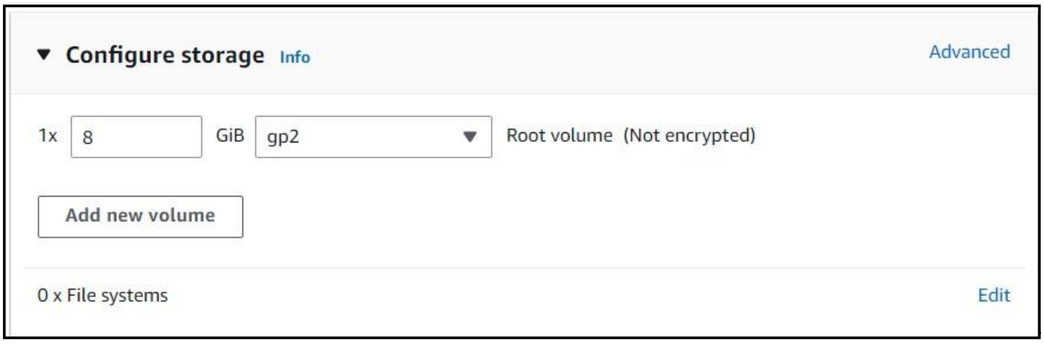
☑ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0 ▼

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ 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.)

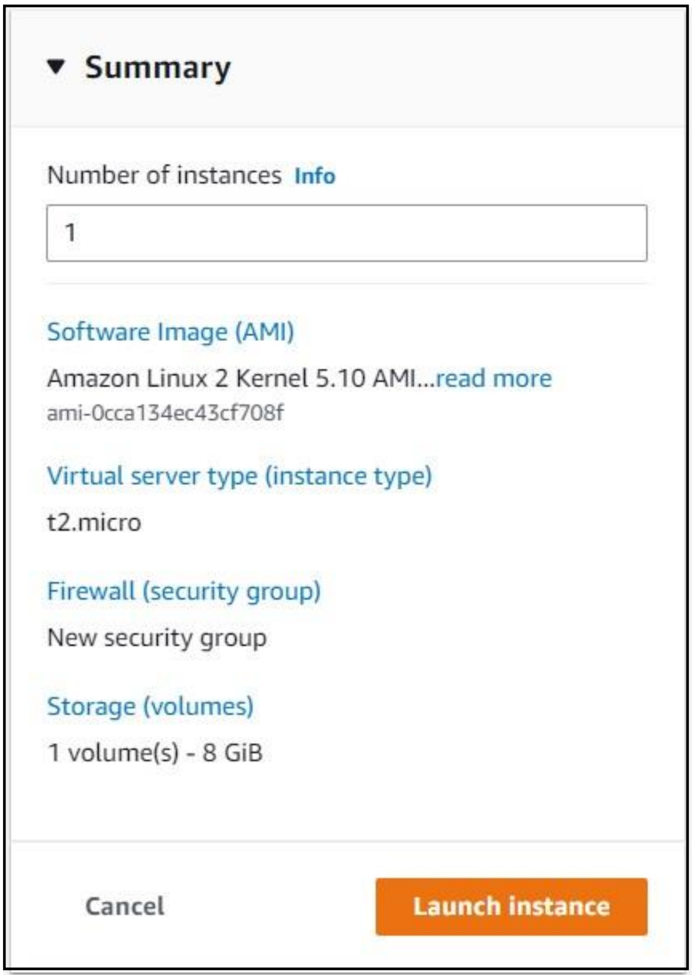


1x 8 GiB gp2 Root volume (Not encrypted)

0 x File systems

Step6:

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



Number of instances Info

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more

ami-0cca134ec43cf708f

Virtual server type (instance type)

t2.micro

Firewall (security group)

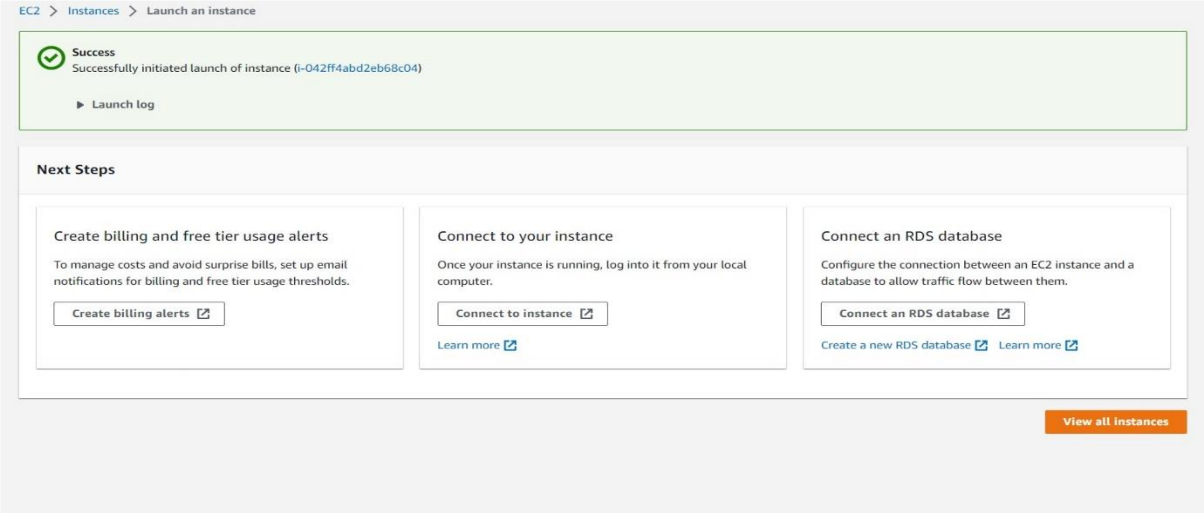
New security group

Storage (volumes)

1 volume(s) - 8 GiB

Step7:

Choose “**View all Instances**”



Success
Successfully initiated launch of instance (i-042ff4abd2eb68c04)

Next Steps

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Connect to your instance

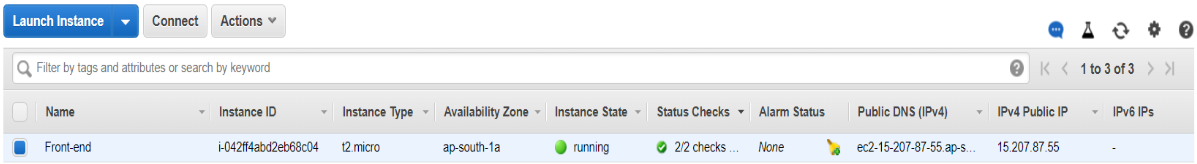
Once your instance is running, log into it from your local computer.

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Step8:

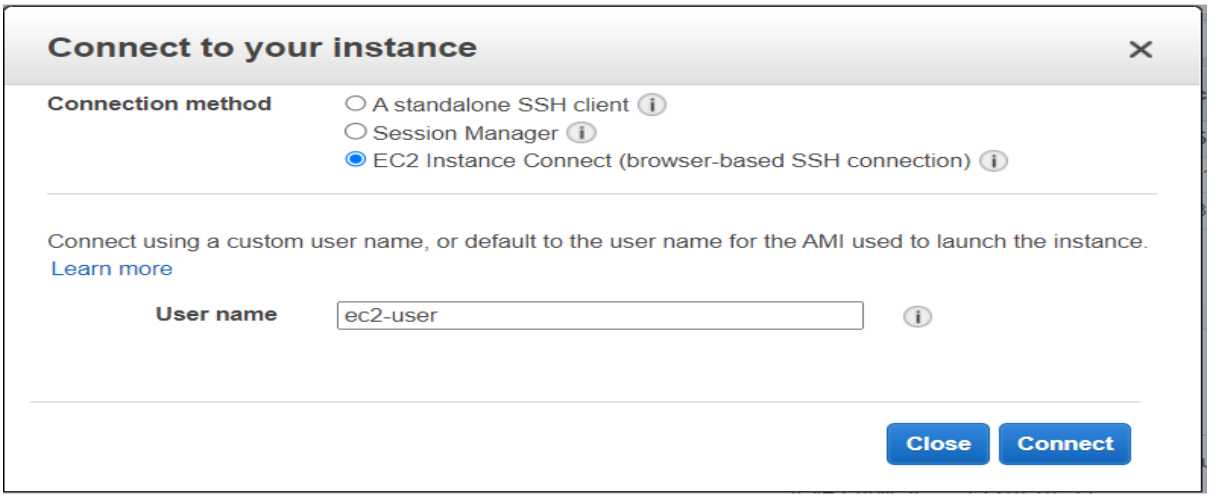
Choose Option “**Connect**” after the “**Status Checks**” shows “**2/2checks**”.



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
■	Front-end	i-042ff4abd2eb68c04	t2.micro	ap-south-1a	● running	✓ 2/2 checks ...	None	ec2-15-207-87-55.ap-s...	15.207.87.55	-

Step9:

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



Connect to your instance

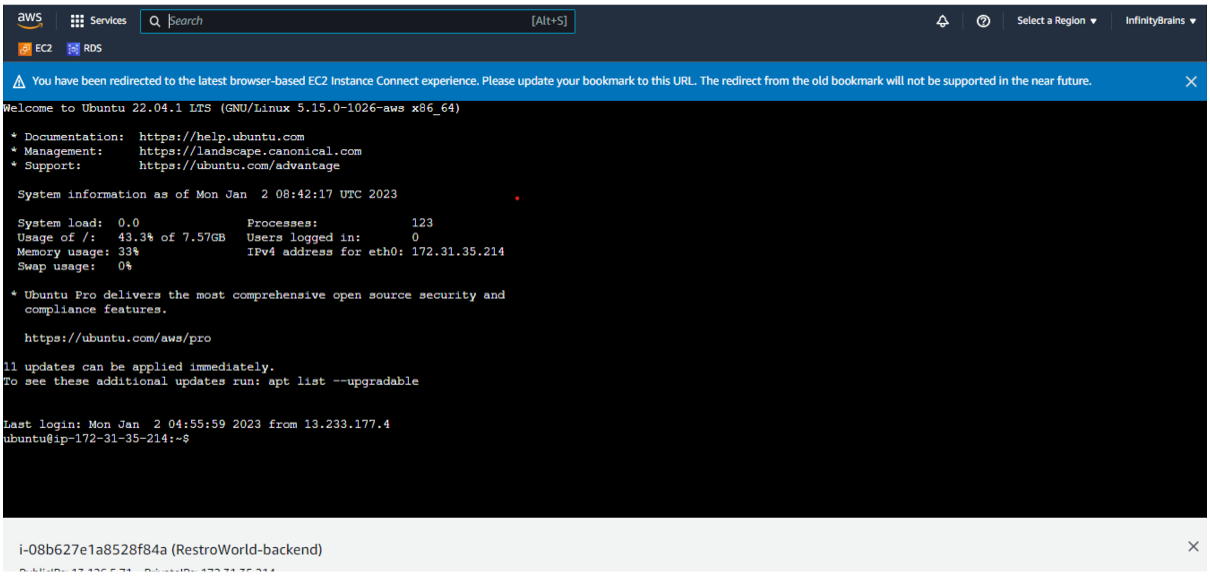
Connection method
○ A standalone SSH client
○ Session Manager
● EC2 Instance Connect (browser-based SSH connection)

Connect using a custom user name, or default to the user name for the AMI used to launch the instance.
Learn more

User name: ec2-user

Step 10:

This type of interface is open.



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:~$
```

i-08b627e1a8528f84a (RestroWorld-backend)

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

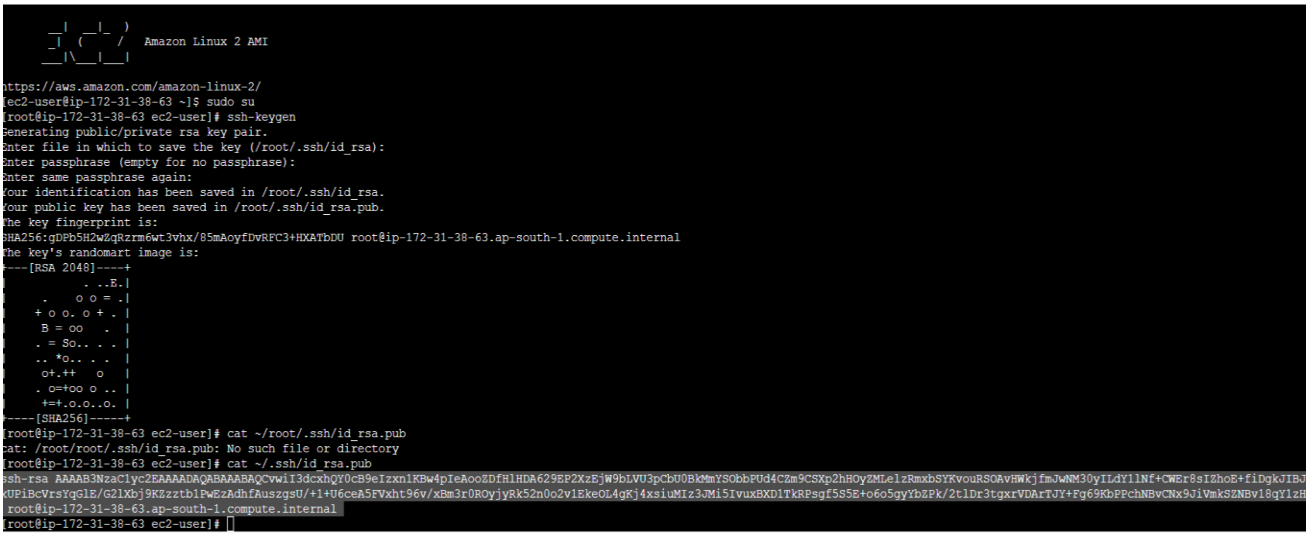
ssh-keygen

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:

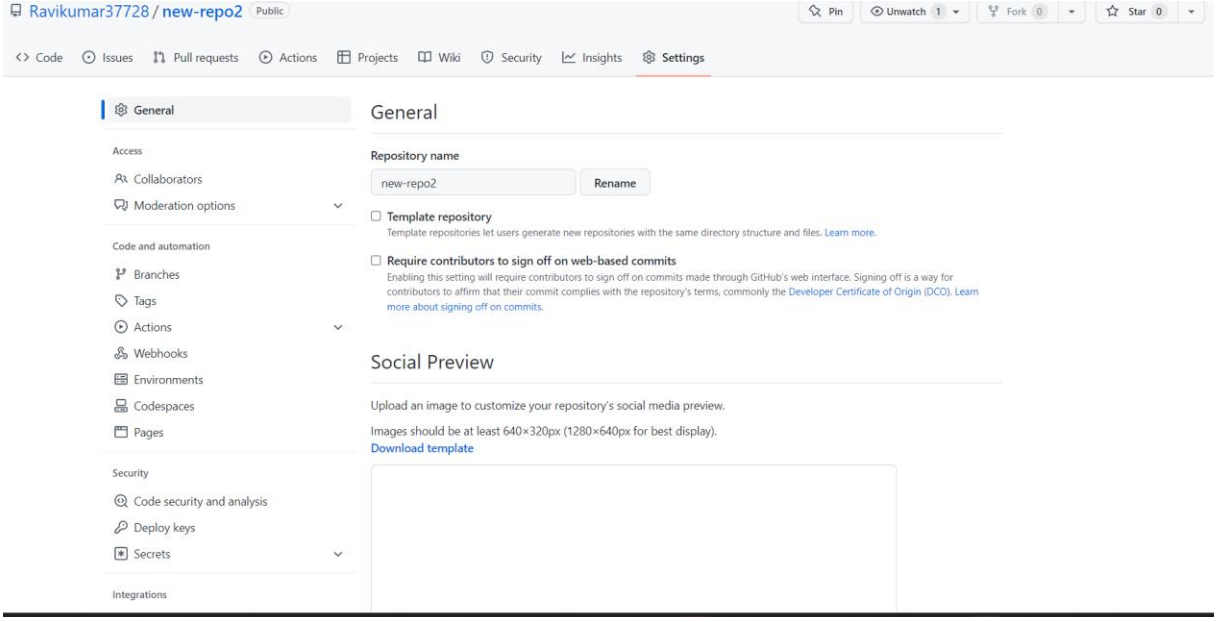
cat ~/.ssh/id_rsa.pub



```
       __|  __|_  )
       _|  (     /   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:gDPb5H2wZqRzrm6wt3vhx/85mAoyfDvRFC3+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. o + . |
|     B = oo    . |
|    . = So.. . . |
|    .. *o.. . .  |
|     o+.++   o   |
|    . o=+oo o .. |
|     +=+.o.o..o. |
+----[SHA256]-----+
[root@ip-172-31-38-63 ec2-user]# cat ~/root/.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 AAAAB3NzaC1yc2EAAAADAQABAAABAQCvwiI3dcxhQY0cB9eIzxn1KBw4pIeAooZDfHlHDA629EP2XzEjW9bLVU3pCbU0BkMmYSObbPUd4CZm9CSXp2hHOyZMLelzRmxbSYKvouRSOAvHWkjfmJwNM30yILdY1lNf+CWEr8sIZhoE+fiDgkJIBJ
kUPiBcVrsYqGlE/G2lXbj9KZzztb1PwEzAdhfAuszgsU/+1+U6ceA5FVxht96v/xBm3r0ROyjyRk52n0o2v1EkeOL4gKj4xsiuMIz3JMi5IvuxBXD1TkRPsgf5S5E+o6o5gyYbZPk/2tlDr3tgxrVDArTJY+Fg69KbPPchNBvCNx9JiVmkSZNBv18qY1zH
 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.



General

Repository name: new-repo2

□ Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more.

□ Require contributors to sign off on web-based commits

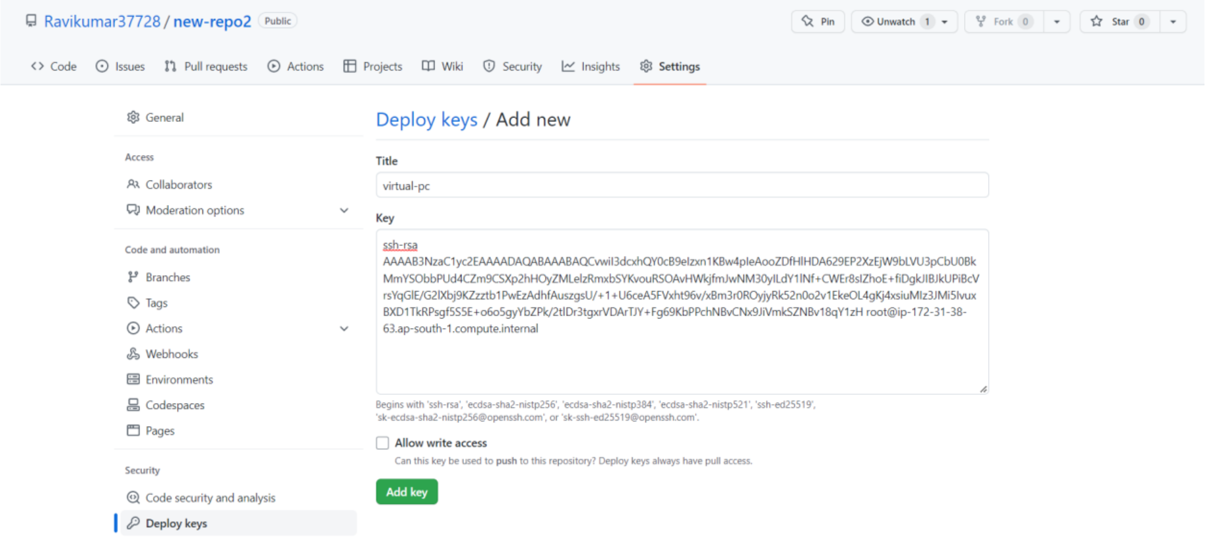
Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origin (DCO). Learn more about signing off on commits.

Social Preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

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



Deploy keys / Add new

Title: virtual-pc

Key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCvwiI3dcxhQY0cB9elzxn1KBw4pleAooZDfHlHDA629EP2XzEjW9bLVU3pCbU0Bk
MmYSObbPUd4CZm9CSXp2hHOyZMLelzRmxbSYKvouRSOAvHWkjfmJwNM30yILdY1lNf+CWEr8sIZhoE+fiDgkJIBJkUPiBcV
rsYqGlE/G2lXbj9KZztb1PwEzAdhfAuszgsU/+1+U6ceA5FVxht96v/xBm3r0ROyjyRk52n0o2v1EkeOL4gKj4xsiuMIz3JMi5lvux
BXD1TkRPsgf5S5E+o6o5gyYbZPk/2tlDr3tgxrVDArTJY+Fg69KbPPchNBvCNx9JiVmkSZNBv18qY1zH root@ip-172-31-38-
63.ap-south-1.compute.internal
```

Begins 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.

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 insatnces.

- Give following cmds:

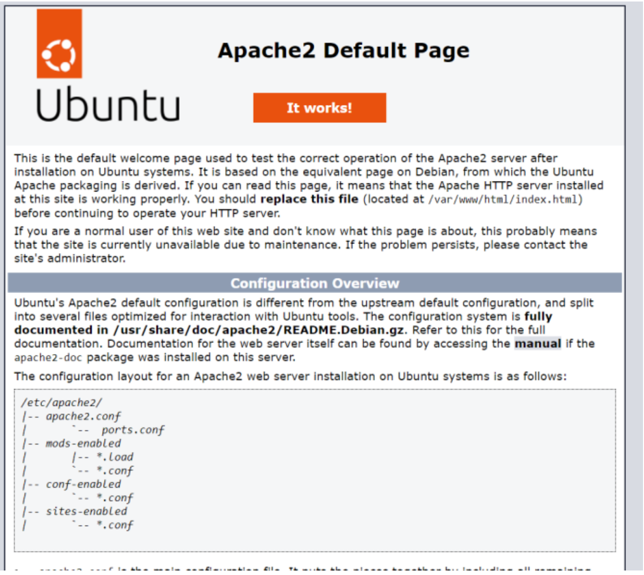
```
sudo apt-get install apache2

sudo systemctl status apache2
```

NOTE: Now, if you're using a firewall, it is necessary to establish a rule in the Firewall so that Apache can run smoothly. If you have no firewall installed, feel free to skip this step

```
sudo ufw allow "Apache Full"
```

- Open IPv4 public Ip on browser, this type of interface is open.



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining

- Fire following cmds:

```
cd /var/www/html

sudo apt install php libapache2-mod-php php-mbstring php-xmlrpc     php-soap php-gd php-xml php-cli php-zip php-bcmath php-tokenizer php-json php-pear

sudo apt-get install php8.1-mysql

apt-get update

sudo apt-get install mysql-server

curl  -o-  https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh  | bash

. ~/.nvm/nvm.sh

nvm install 16
```

- Remove index.html, which is create when we install apache2.

```
rm -rf index.html
```

- Now add ssh link in your following cmd and fire it in terminal

```
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 git-hub repo.

- Now fire following cmds:

```
npm i npm@latest -g

// npm install --legacy-peer-deps

npm i @vue/cli-service --legacy-peer-deps
```

- fire following cmd:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

- add path in “DocumentRoot” /var/www/html/ to /var/www/html/dist
- add given below data just above the </VirtualHost> as we can see in fig.

```
<Directory “/var/www/html”>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Require all granted
</Directory>
```

- fire following cmds

```
sudo a2enmod rewrite
```

sudo service apache2 restart

```
sudo apt update
sudo apt upgrade


sudo apt install software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt update
```

```
sudo apt install php8.1-fpm php8.1-common php8.1-mysql php8.1-xml php8.1-xmlrpc
php8.1-curl php8.1-gd php8.1-imagick php8.1-cli php8.1-dev php8.1-imap php8.1-
mbstring php8.1-soap php8.1-zip php8.1-bcmath -y
```

then fire following cmd:

```
nano /etc/apache2/apache2.conf
```

- go to the very bottom and paste this

```
<Directory "/var/www/html">


        AllowOverride All


</Directory>
```

- restart the apache2 service using below cmd.

service apache2 restart

- now browse ec2 IPv4 and your project is live.