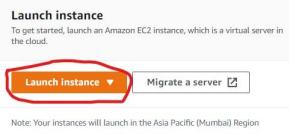
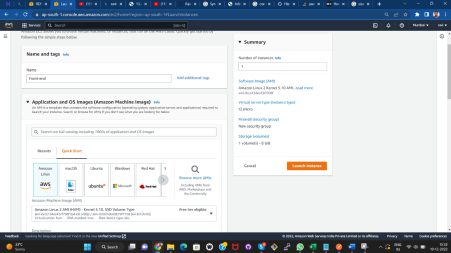
**Title:** Live Vue.js Front-end project in Aws Ec2-Instance.

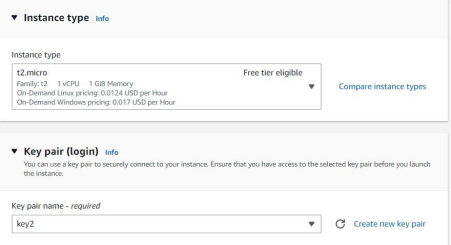
**SKIP UPTO Step 8 if you have already an instance and you want project live in that same one.**

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.

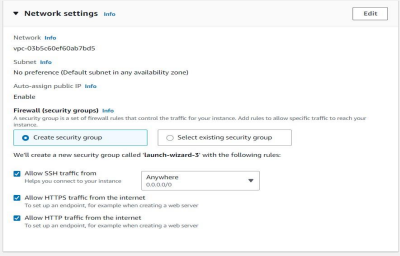


• For how to create keypair see documentation of “Front-end quasar project live” after step3 in it.

Step4:

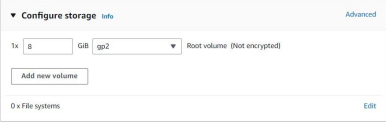
**Network setting> Firewall (security groups)** select “**Create security group**” if you don’t have one or if you already created security group select “**Select existing security group**” and then select the name of security group you created.

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**”.



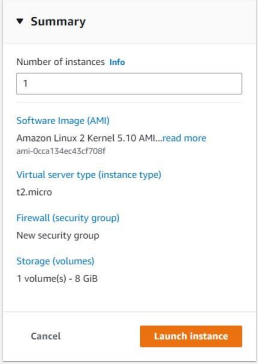
Step5:

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.)



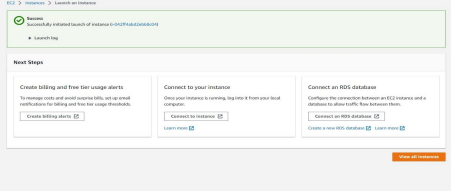
Step6:

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



Step7:

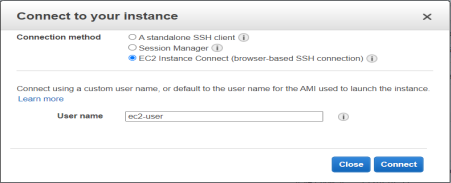
Choose “**View all Instances**”

Step8:

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

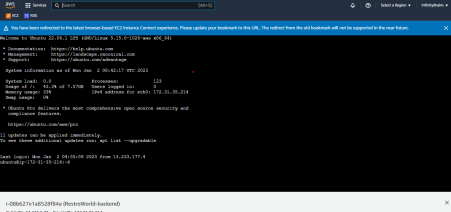
Step9:

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



Step 10:

This type of interface is open.

Now, first give following cmds :

sudo su

sudo apt-get update

• Active git using following cmd:

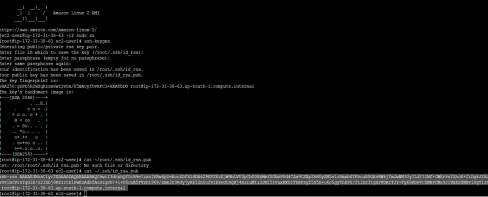
git init

• Create ssh key using following cmd in terminal

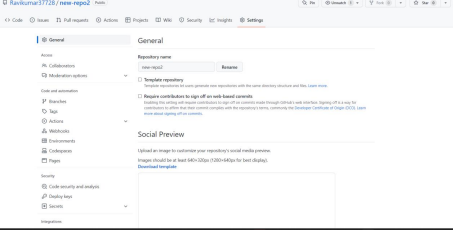
ssh-keygen

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

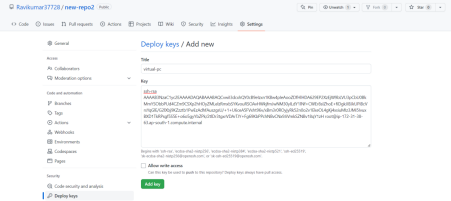
cat ~/.ssh/id\_rsa.pub



• Paste it in git-hub’s >settings>SSH keys[note:: use this step if new instance is launched(it means not launched using AMIs or snapshot).Also remember it is a one time process,once you add ssh key in github you not have to add every time ssh key when you launch any new project.You just connect your ec2 to this github account using this key once for all. ]



• Paste copied text in there and named it. Then press “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 insatnces.

• Give following cmds:

sudo apt-get install apache2 (**don’t use this cmd if apache2 is already there**)

sudo service apache2 status

**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.



• Fire following cmds:

cd /var/www/html

apt-get update

• Remove index.html, which is created by default 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.

• fire following cmds

sudo a2enmod rewrite

sudo service apache2 restart

**EITHER**

• 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>

**OR**

In most case we want to add multiple .conf file for different project for that create new .conf file using cmd:

nano /etc/apache2/sites-available/xyz.conf

Enter following details in that file

<VirtualHost \*:80>

ServerName yourdomainname.com

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html/path of your project folder/public

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

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

Options Indexes FollowSymLinks MultiViews

AllowOverride All

Require all granted

</Directory>

</VirtualHost>

• Then enable that file using “a2ensite” cmd for e.g. your config file name is rw.conf then hit cmd::

a2ensite rw.conf

• Using this command allows domain hosts like cloudflare or route53 that, specifically listen on which domain or subdomain projects from single ec2 instance.

• After it restart apache 2 using cmd:

sudo service apache2 restart

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.