

Assignment: 09

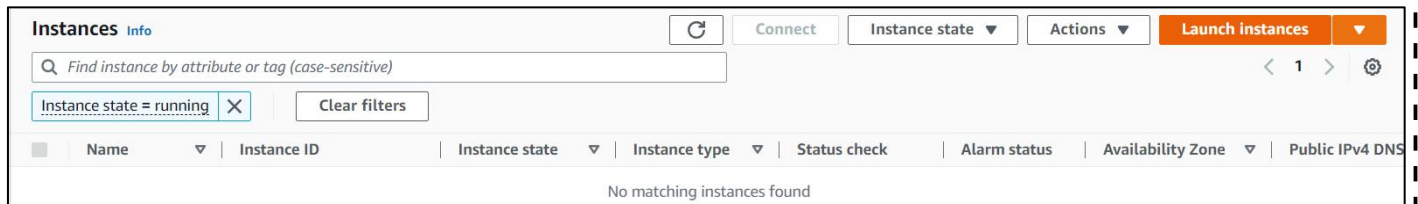
Title : Deploy a project from github to EC2

The following steps to be followed :

- 1-- Create an EC2 instance.
- 2-- Connect to Bitwise SSH and run the commands.
- 3-- Add the port number and security.

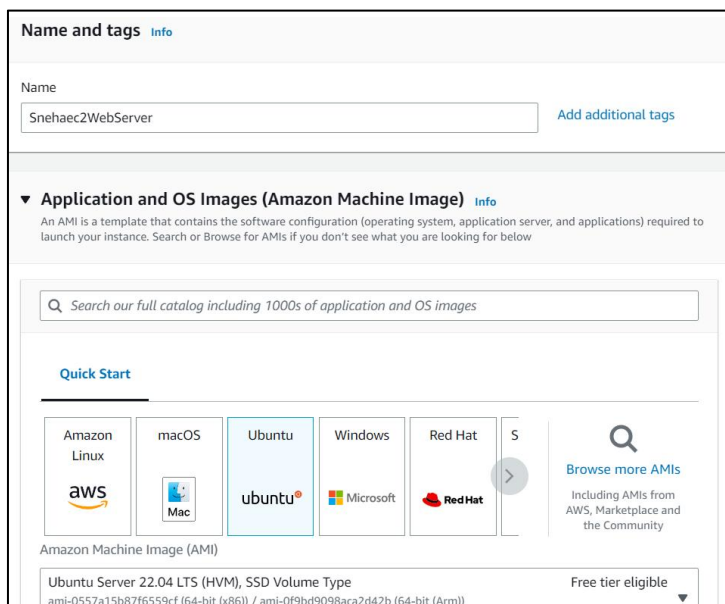
Steps to create an instance on EC2 :

1. Open the Amazon EC2 console.
2. From the EC2 console dashboard, Click on **Instances(Running)**, choose **Launch instance**.

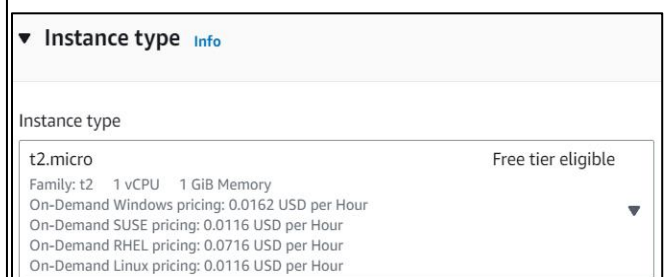


and The **Launch an instance** page opens..

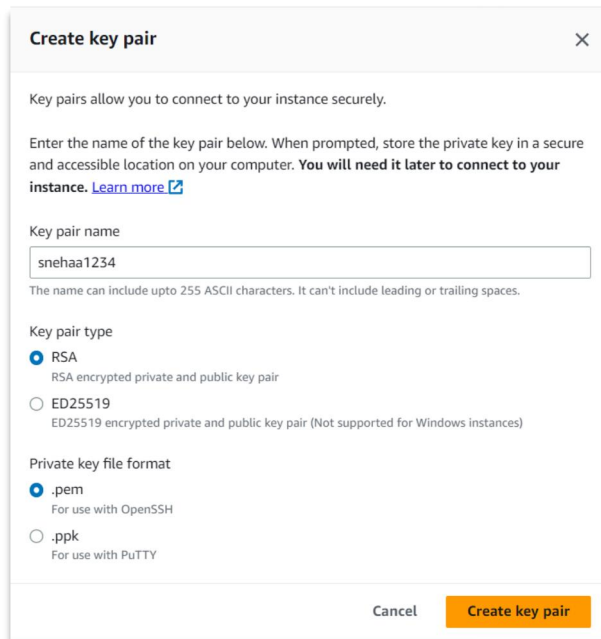
3. Under **Name and tags**, for **Name**, enter a descriptive name for your instance like 'Snehaec2WebServer'.
4. Under **Application and OS Images (Amazon Machine Image)**, do the following: Choose **Quick Start**, and then choose **Ubuntu**. This is the operating system (OS) for your instance, which is **Free Tier Eligible**.



Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the **t2.micro** instance type, which is selected by default. The **t2.micro** instance type is eligible for the free tier.



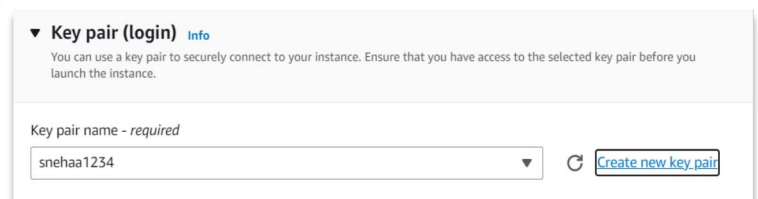
5. Under **Key pair (login)**, for **Key pair name**, choose the key pair that you created already or Choose **Create new key pair**. A dialogue box opens - Give a name to the key pair under the **Key pair name** like **snehaa1234**



The key pair generated is of:

- Type - RSA
- File format - .pem

Click on **Create key pair** and the .pem file of your key pair is automatically downloaded. And is saved for further use.

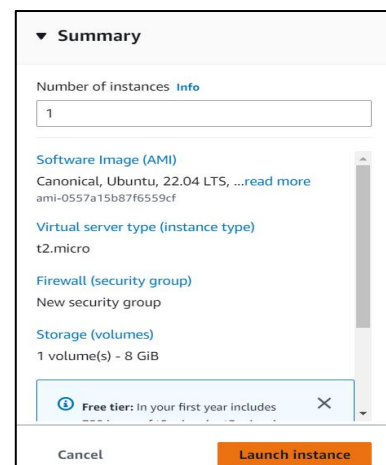
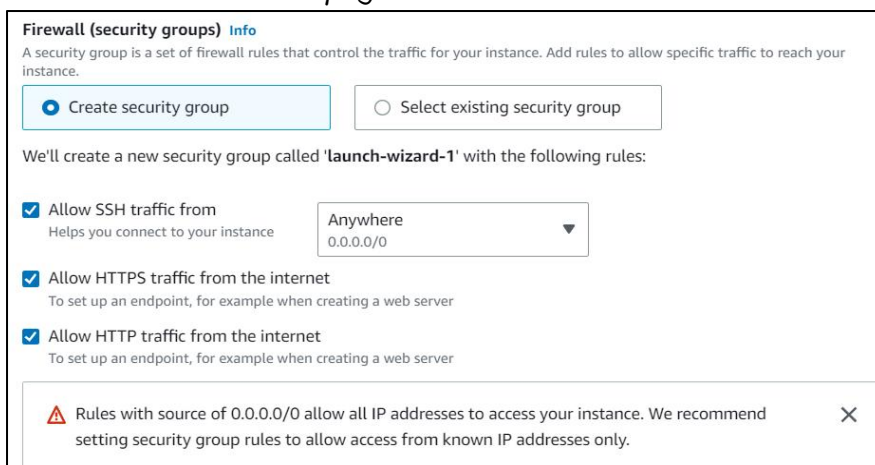


6. In **Network settings**, under the **Firewall (Security groups)** there is a by default selection of **Create security Groups** under which check or select all the three boxes namely :

- ☒ **Allow SSH traffic from** - Helps you connect to your instance
- ☒ **Allow HTTPS traffic from the internet** - To set up an end point.
- ☒ **Allow HTTP traffic from the internet** - To set up an endpoint .

7. Keep the default selections for the other configuration settings for your instance. Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance**.

A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.



Steps to connect client to server (EC2) using Bitrise SSH :

1. Click on the **Instance ID** of the instance you created. The instance summary opens .
2. Copy the **Public IPv4 Address**.

Instance ID

i-0220f6d2677952e20

Public IPv4 address



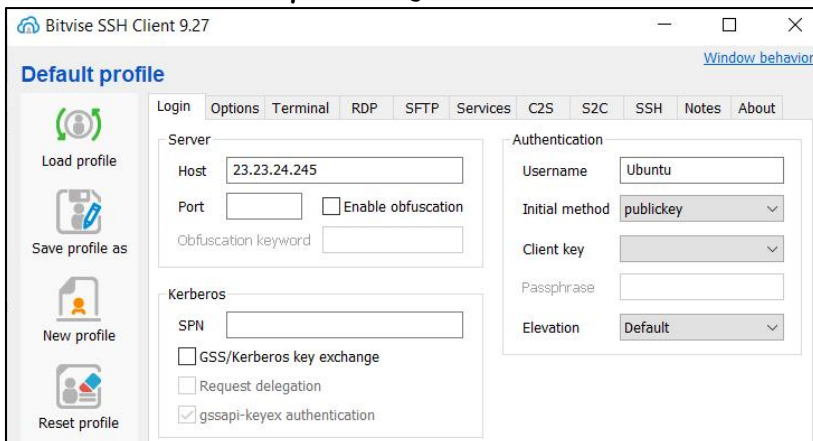
23.23.24.245 | [open address](#)

3. Download the **Bitwise SSH client** from browser - Install it and open the application to move further.

4. Under **Login** section ,In **Server** - Host paste the public IPv4 address of the instance
In the **Authentication** part do as follows:

Username - Ubuntu

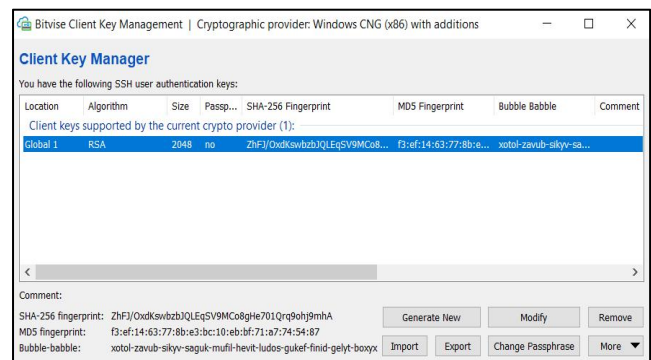
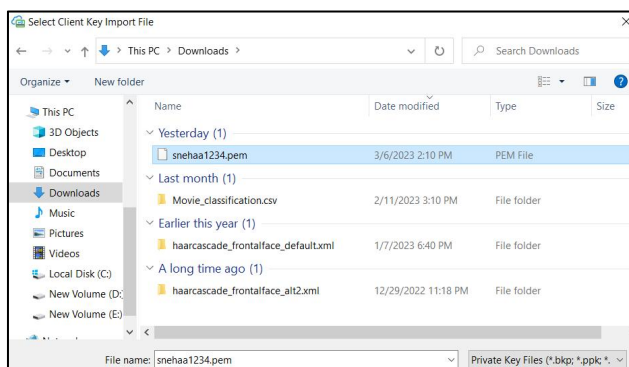
Initial method - **publickey**



Click on **Client key manager** , in the dialogue box Click on **Import** .

Import the key pair generated while making the instance ->choose open
-> **import**.

It is visible in the client key manager as **Global 1**. Return back (close the window)



5. In the **Authentication** section , **Client key** - **Global 1**

Click On **log in** -> **Accept & Save**.

6. Open new terminal console - and type —————>

(i) **Sudo apt-get update && sudo apt-get upgrade**

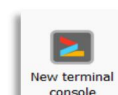
Followed by typing **y** when asked for yes/no and then pressing enter when finished.

(ii) **Sudo apt-get install nginx**

Followed by typing **y** when asked for yes/no and then pressing enter when finished.

```
ubuntu@ip-172-31-31-167:~$ sudo apt-get update && sudo apt-get upgrade
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [941 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [879 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [173 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [119 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [139 kB]
Fetched 2486 kB in 1s (1795 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
```

```
ubuntu@ip-172-31-31-167:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg8 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4
  nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg8 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 ngi
  nginx-common nginx-core
```



(iii) `curl -sL https://deb.nodesource.com/setup_16.x|sudo -E bash -`

-(curl: a command-line tool used to transfer data from or to a server.

-sL: two options for the curl command. -s is used to silence any progress or error messages, and -L tells curl to follow redirects if any.

https://deb.nodesource.com/setup_16.x: the URL of the script that adds the Node.js package source.

|: a pipe character, which redirects the output of the curl command to the input of the next command.

sudo -E bash -: runs the script with elevated privileges using the sudo command. The

-E option preserves the environment variables, and the - option tells bash to read commands from standard input.)

(iv) `sudo apt install nodejs`(server side scripting runtime environment. It allows developers to run JavaScript code outside of a web browser, making it useful for server-side applications and command-line tools)

(v) `git clone https://github.com/itsmesneha/MYNEWREPO.git`(repository name)

after giving repo name, username and password will come .for username we have to give email of github acct and for password we have to give that token(ex-tok2).Then we can see that repo is copied and by typing `ls` we can see that repo-
MYNEWREPO

(vi) `cd MYNEWREPO` : by giving this command we can move to this directory.

(vii) `npm install` by giving this command in repo2 we have to install npm. npm stands for Node Package Manager. It is a package manager for the Node.js runtime environment, and it is used to install, manage, and share packages or modules of JavaScript code that can be used in Node.js projects.

Now, before starting the server we have to add port number as in index.js file the port is 4000.so we need to add that.

Steps to add the port number :

1. go to instances and click instance id(which is used here) and go to security and click security groups.
2. in security groups click *Edit inbound rules*.

The screenshot shows the AWS IAM console. On the left, the 'Security details' section is expanded, showing the IAM Role as '-' and the Security groups as 'sg-064f641660bcdaf9e (launch-wizard-6)'. On the right, the 'Inbound rules' tab is selected, showing a list of 4 inbound rules. The first rule is highlighted, showing its details: Name '-', Security group rule ID 'sgr-0ba4ab810801909ef', IP version 'IPv4', Type 'HTTP', Protocol 'TCP', and Port range '80'.

3. in edit inbound rules click **add rule** and in **type** select **custom TCP**, in **port range** give **4000** and in **Source** select **Anywhere public IPv4**.

Inbound rules Info					
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	
sgr-0ba4ab810801909ef	HTTP	TCP	80	Custom	<input type="text" value="0.0.0.0/0"/>
sgr-03393774891fc9a3a	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0/0"/>
sgr-0a7bf59d1c7fa56db	Custom TCP	TCP	4000	Anywh...	<input type="text" value="0.0.0.0/0"/>
sgr-0b19bd427e52fee02	HTTPS	TCP	443	Custom	<input type="text" value="0.0.0.0/0"/>

4. now in bitwise terminal type `node index.js` and the server is started.

Now, copy that ec2 IPv4 address and paste it in a new tab with **:4000** and by clicking we can run the website.

```
ubuntu@ip-172-31-5-175:~$ ls
MYNEWREPO
ubuntu@ip-172-31-5-175:~$ cd MYNEWREPO
ubuntu@ip-172-31-5-175:~/MYNEWREPO$ node index.js
Started server
```

