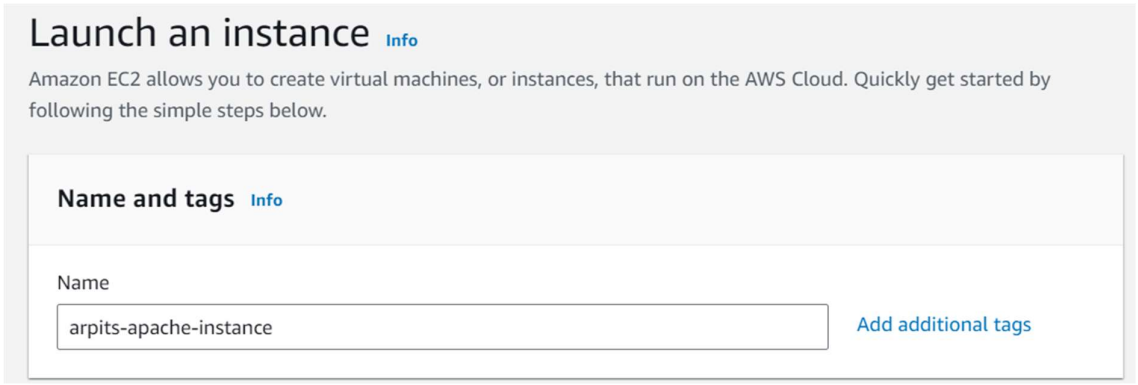
AMAZON WEB SERVICES PROJECT- II

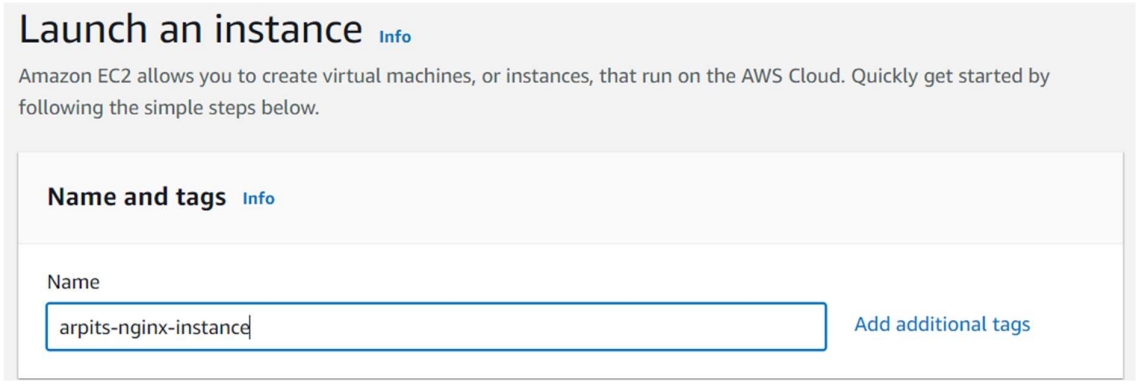
AIM: Create two EC2 web servers one with the apache http server installed on it and one with the nginx server installed on it and do all the installations using the user data section of ec2.

PROCEDURE:

We'll proceed step by step with proper understanding, documentation and demonstration wherever possible

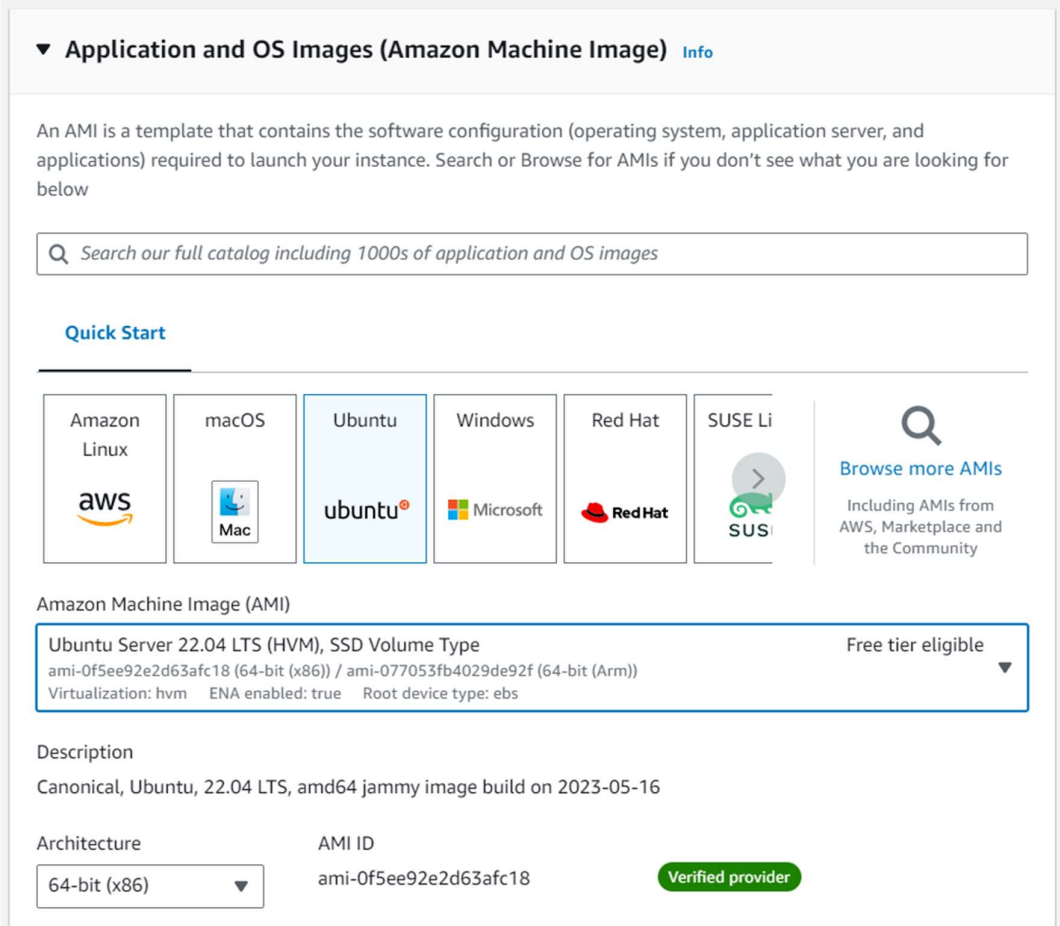
1. Create EC2 Instances: Go to the AWS Management Console and navigate to the EC2 dashboard. Click on "Launch Instance."



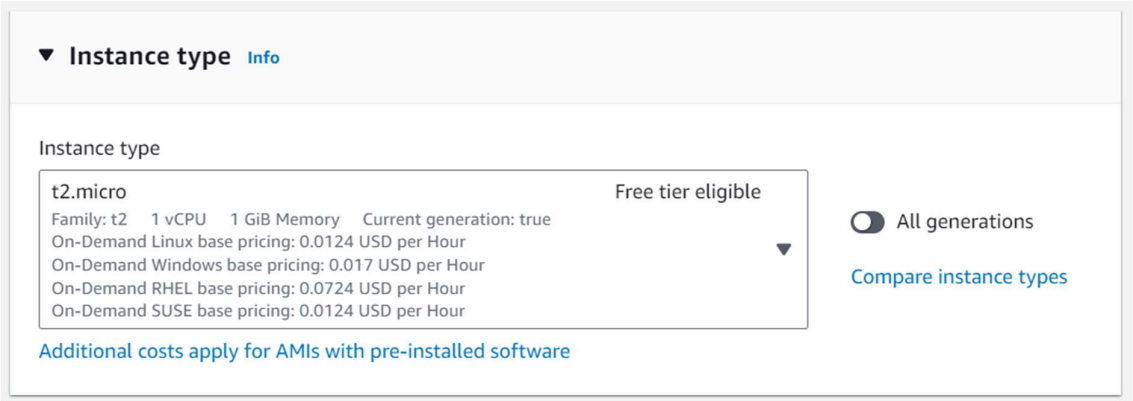


- Instance "arpits-apache-instance" is for Apache server and instance "arpits-nginx-instance" is for Nginx server.

2.Choose an Amazon Machine Image (AMI): Select an Amazon Linux 2 or any other Linux-based AMI of your choice. Here, ubuntu server. We can choose any, only the later command will be different.

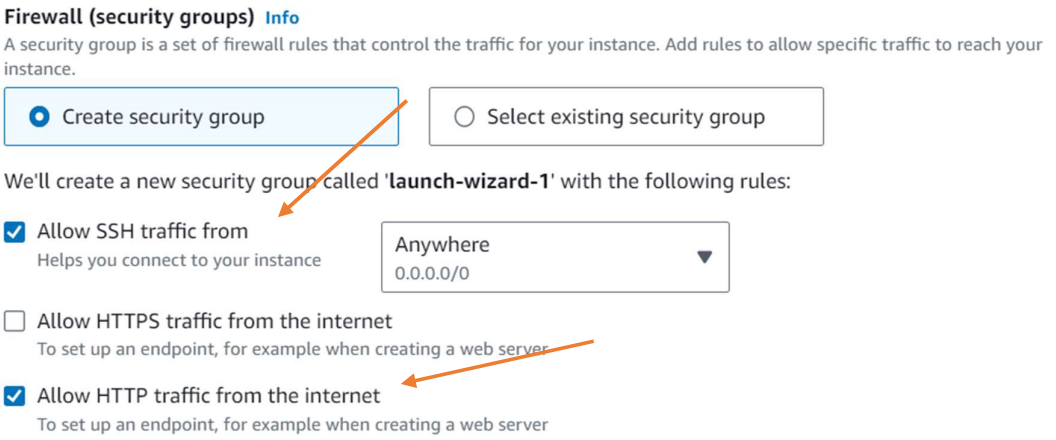


3. Choose an Instance Type: Choose an instance type based on your requirements. Here, we have selected by-default instance type i.e., t2.micro.

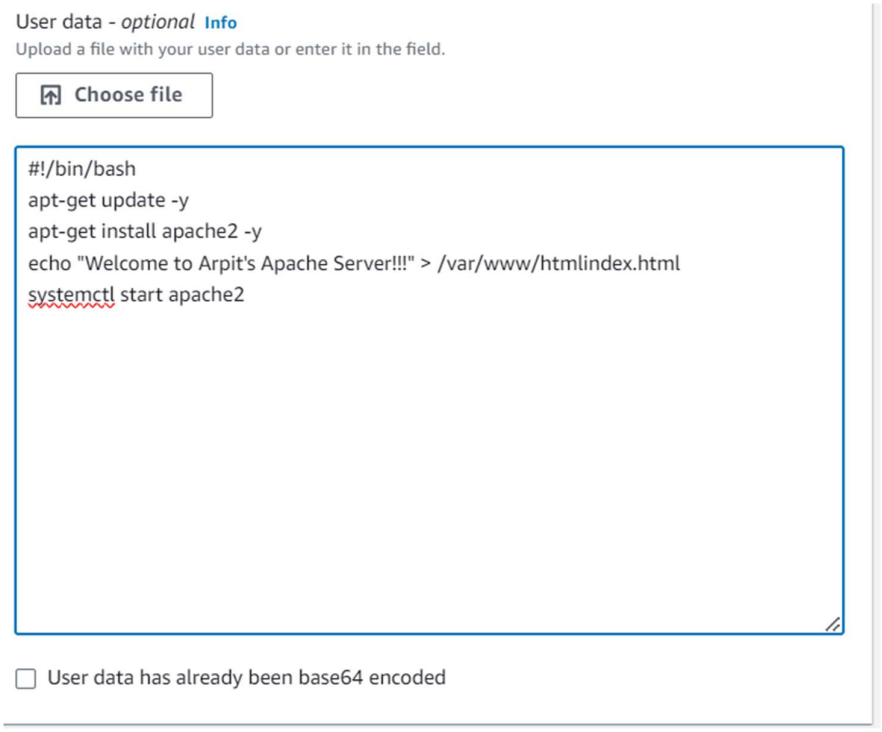


4.Configure Instance:

- Configure the network settings, subnet, and security group to allow HTTP traffic (port 80) and SSH access (port 22).
- In the "Advanced Details" section, add the following user data script for the Apache server:



In the **"Advanced Details"** section, add the following user data script for the Apache server:

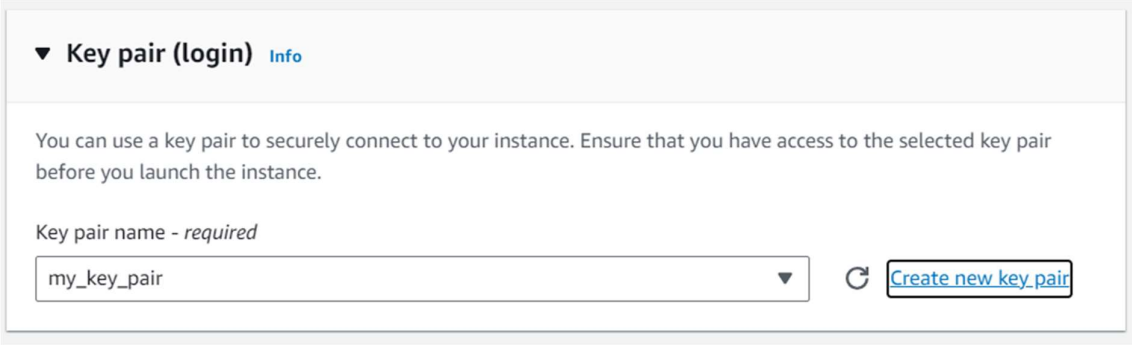


- apt-get update -y: it will do all the updates required.
- apt-get install apache2 -y: it will install all the packages needed.
- echo means print. it is used to print any type of message in the web server as required.
- systemctl start apache2: start the services of apache2.

5. Add Storage, Tags, and Configure Security Group: Configure additional settings such as storage, tags, and security groups as needed. We have selected most of them as default.

6. Review and Launch: Review your configurations, and when you're ready, click "Launch."

7. Key Pair: Choose an existing key pair or create a new one for SSH access to your instances.

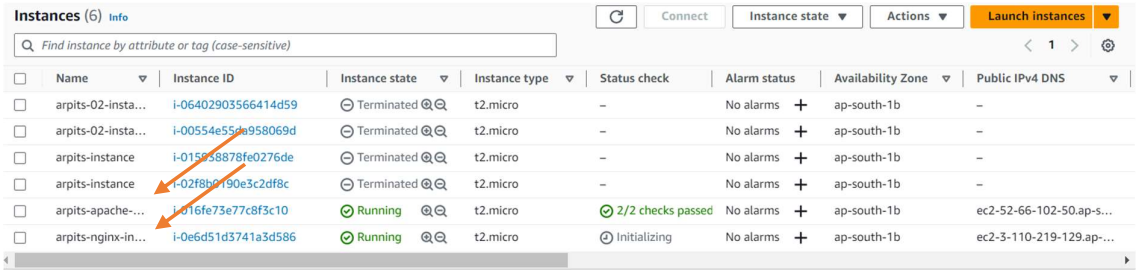


We have created a key pair named my_key_air. We'll use this for both the instances.

8. Launch Instances: Click "Launch Instances" to create your EC2 instance.

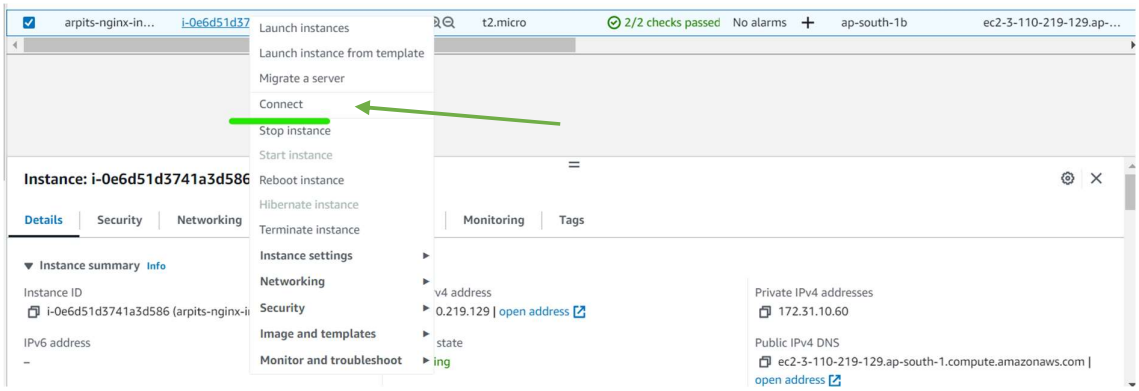
- Now as we can see in the below picture both our instances are now in running state.

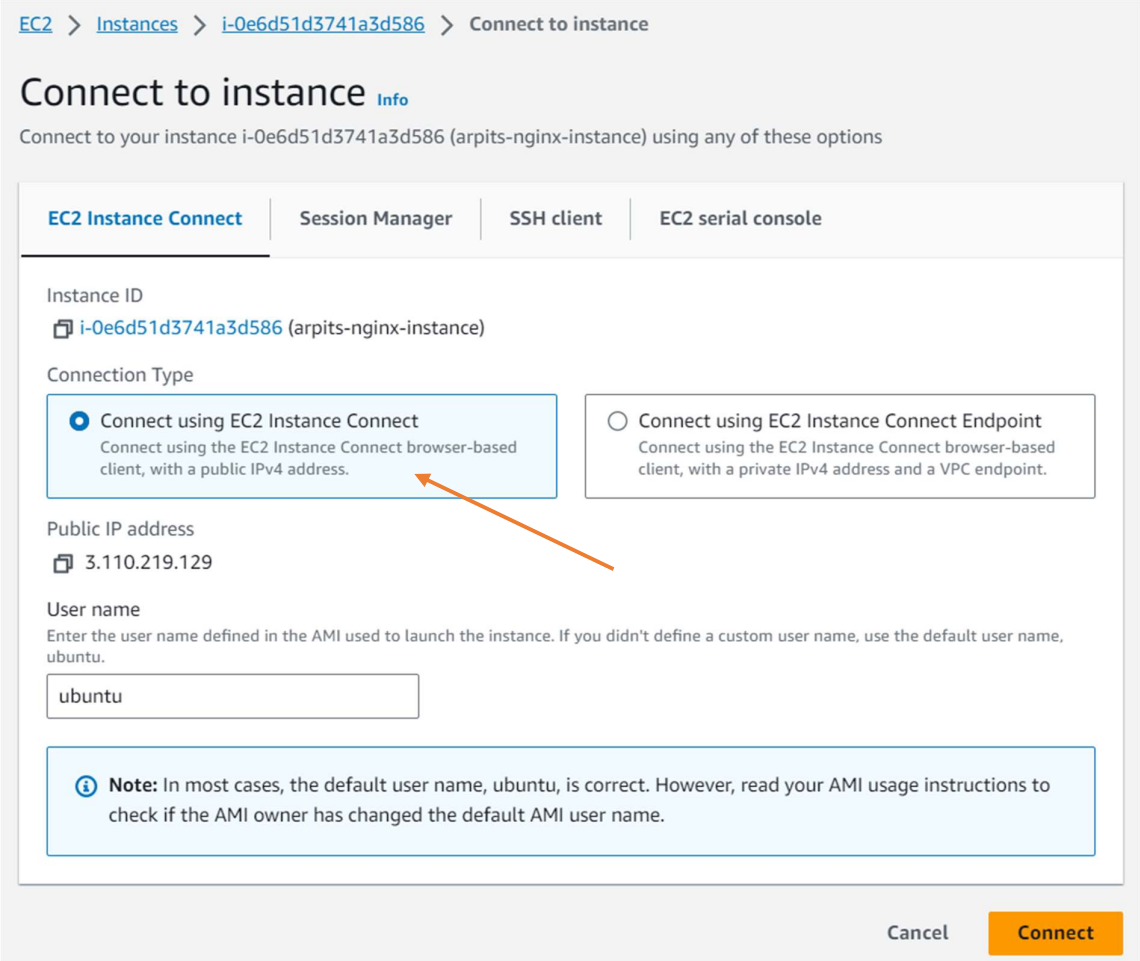
9. Access the Instances: Wait for the instances to be launched and their status to become "running." We can now access your EC2 instances using SSH.



For Nginx installation for arpits-nginx-instance do the following:

- Go to your instance.
- Right click on it. A dropdown list appears as shown in the below picture.
- Click on connect.
- In connection type select Connect using EC2 Instance Connect and connect.





After clicking on Connect, a ubuntu poweshell will appear. Here, we have to write the user data commands.

Now type the following commands:

- sudo apt-get update : It will do all the updates required for the server
- sudo apt-get install nginx : It will install all the necessary packages needed.
- Then it will ask for extra disk space. Type Y.
- sudo systemctl status nginx : It'll show the current status of the server.

In the status section, it'll show active (running). That means it is ready to serve.

sudo apt-get update:

```
ubuntu@ip-172-31-10-60:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [966 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [222 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.6 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [830 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [133 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 B]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [979 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [213 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [21.8 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [476 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.4 kB]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [640 B]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
```

sudo apt-get install nginx:

```
ubuntu@ip-172-31-10-60:~$ 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 libjbig0 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 libjbig0 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 nginx-common nginx-core
0 upgraded, 20 newly installed, 0 to remove and 113 not upgraded.
Need to get 2691 kB of archives.
After this operation, 8335 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-core all 2.37-2build1 [1041 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fontconfig-config all 2.13.1-4.2ubuntu5 [29.1 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libdeflate0 amd64 1.10-2 [70.9 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfontconfig1 amd64 2.13.1-4.2ubuntu5 [131 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libjpeg-turbo8 amd64 2.1.2-0ubuntu1 [134 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libjpeg8 amd64 8c-2ubuntu10 [2264 B]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libjbig0 amd64 2.1-3.1ubuntu0.22.04.1 [29.2 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libwebp7 amd64 1.2.2-2ubuntu0.22.04.1 [206 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtiff5 amd64 4.3.0-6ubuntu0.5 [184 kB]
```

sudo systemctl status nginx:

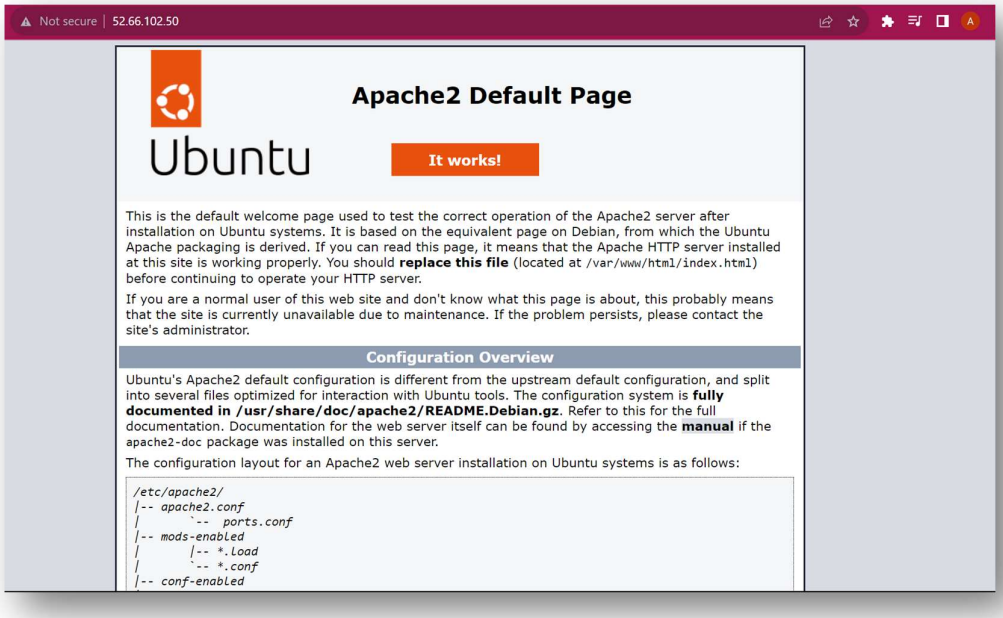
```
ubuntu@ip-172-31-10-60:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-09-07 08:57:05 UTC; 1min 14s ago
       Docs: man:nginx(8)
    Process: 2484 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 2485 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2578 (nginx)
      Tasks: 2 (limit: 1141)
     Memory: 4.4M
        CPU: 22ms
     CGroup: /system.slice/nginx.service
             ├─2578 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2581 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Sep 07 08:57:05 ip-172-31-10-60 systemd[1]: Starting A high performance web server and a reverse proxy server...
Sep 07 08:57:05 ip-172-31-10-60 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-10-60:~$
```

i-0e6d51d3741a3d586 (arpits-nginx-instance)

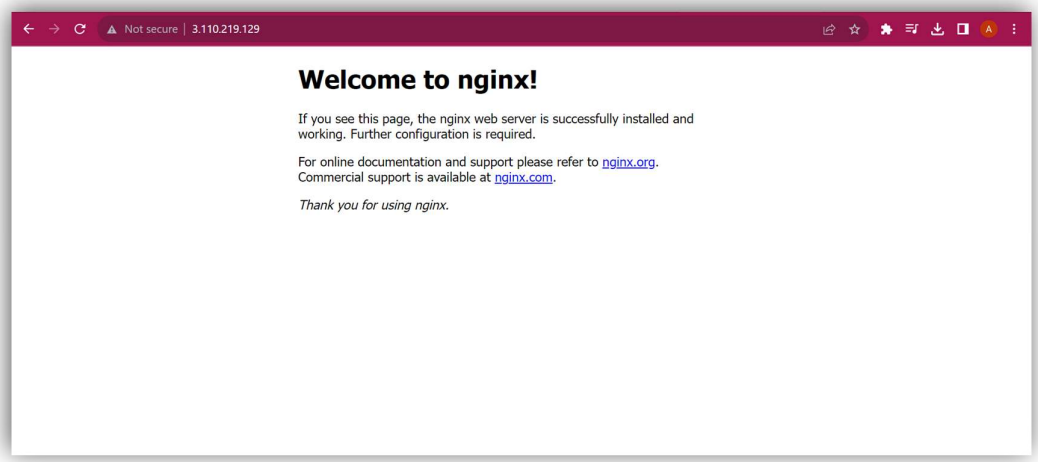
PublicIPs: 3.110.219.129 PrivateIPs: 172.31.10.60

Now, our Nginx installation is done and server is ready to use.

10. Test the Web Servers: For the Apache server, open a web browser and enter the public IP address or DNS name of the instance. For the Nginx server, do the same with its public IP address or DNS name.



This shows that apache server is installed successfully.



This shows that nginx server is installed successfully.

RESULT: Creation of two EC2 web servers one with the apache http server installed on it and one with the nginx server installed on it and all the installations using the user data section of ec2 is done successfully.