

# **Graded Project**

## **on**

# **Building CI-CD Pipeline Tool**

**Course 5: SDLC Project: CI-CD Pipeline | DevOps B4**

## Create a complete CI-CD pipeline using bash, python, and crontabs.

The list of tasks is specified below:

### Task 1: Set Up a Simple HTML Project

Create a simple HTML project and push it to a GitHub repository.

### Task 2: Set Up an AWS EC2/Local Linux Instance with Nginx

### Task 3: Write a Python Script to Check for New Commits

Create a Python script to check for new commits using the GitHub API.

### Task 4: Write a Bash Script to Deploy the Code

Create a bash script to clone the latest code and restart Nginx.

### Task 5: Set Up a Cron Job to Run the Python Script

Create a cron job to run the Python script at regular intervals.

### Task 6: Test the Setup

Make a new commit to the GitHub repository and check that the changes are automatically deployed.

## ANSWER:-

### Task 1 - Set Up a Simple HTML Project

- Create a simple HTML project

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Kinnar Chowdhury - Portfolio</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
      color: #333;
    }
    .container {
      max-width: 800px;
      margin: 20px auto;
      padding: 0 20px;
    }
    .header {
      text-align: center;
      margin-bottom: 30px;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="header">
      <h1>Kinnar Chowdhury</h1>
      <h2>Full Stack Developer</h2>
    </div>
  </div>
</body>
</html>
```

```

        .name {
            font-size: 36px;
            margin-bottom: 10px;
        }

        .contact-details {
            margin-bottom: 20px;
        }

        .contact-details p {
            margin: 5px 0;
        }

        .experience-card {
            background-color: #fff;
            padding: 20px;
            margin-bottom: 20px;
            border-radius: 5px;
            box-shadow: 0 2px 5px rgba(0,0,0,0.1);
        }

        .experience-card h2 {
            margin-top: 0;
        }

        .experience-card ul {
            list-style-type: none;
            padding: 0;
        }

        .experience-card ul li {
            margin-bottom: 5px;
        }

        .likes {
            margin-bottom: 20px;
        }

        footer {
            text-align: center;
            margin-top: 50px;
            padding: 20px 0;
            background-color: #333;
            color: #fff;
        }
    }
</style>
</head>
<body>
    <div class="container">
        <div class="header">
            <h1 class="name">Kinnar Chowdhury</h1>
            <div class="contact-details">
                <p>Phone:+91-8918939197</p>
                <p>Email: kinnar@example.com</p>
                <p>Home Address: Asansol, Paschim Bardhaman, West Bengal,
713301</p>
            </div>

```

```

</div>

<div class="experience-card">
  <h2>Working Experience</h2>
  <p><b>Fulltime Senior Software Developer, Cloud Administrator &
DevOps</b></p>
  <p>MSQUBE Technology Solutions Pvt. Ltd. - Since Dec, 2019</p>
  <p>Office Address - Unit #112, SDF Building, Near RDB Cinemas, GP
Block, Sector V, Bidhannagar, Kolkata, West Bengal 700091</p>
  <ul>
    <br>
    <p><b>Professional Skills</b></p>
    <li>Python (framework - Django, Flask, Fast API)</li>
    <li>Database Server (MySQL, POSTGRES, MONGO)</li>
    <li>Web Servers (NginX, Apache)</li>
    <li>Amazon Web Services (VPC, Subnet, EC2, Security Groups, Route
Table, NAT gateway, Internet Gateway, EBS Volumes, RDS, S3, IAM, Route 53,
Cloudfront, Elastic Load Balancers, ECR, ECS, AWS FARGATE, AWS Recognition,
Appconfig etc.)</li>
    <li>Linux Server Administration (UBUNTU, RHEL)</li>
    <li>Data Visualization Tools (Graphana, Apache Superset,
Metabase)</li>
  </ul>
</div>

<div class="likes">
  <h2>Likes</h2>
  <p>Bike riding, PC gaming, Listening to music</p>
</div>
</div>
<footer>
  &copy; 2024 Kinnar Chowdhury - All Rights Reserved
</footer>
</body>
</html>

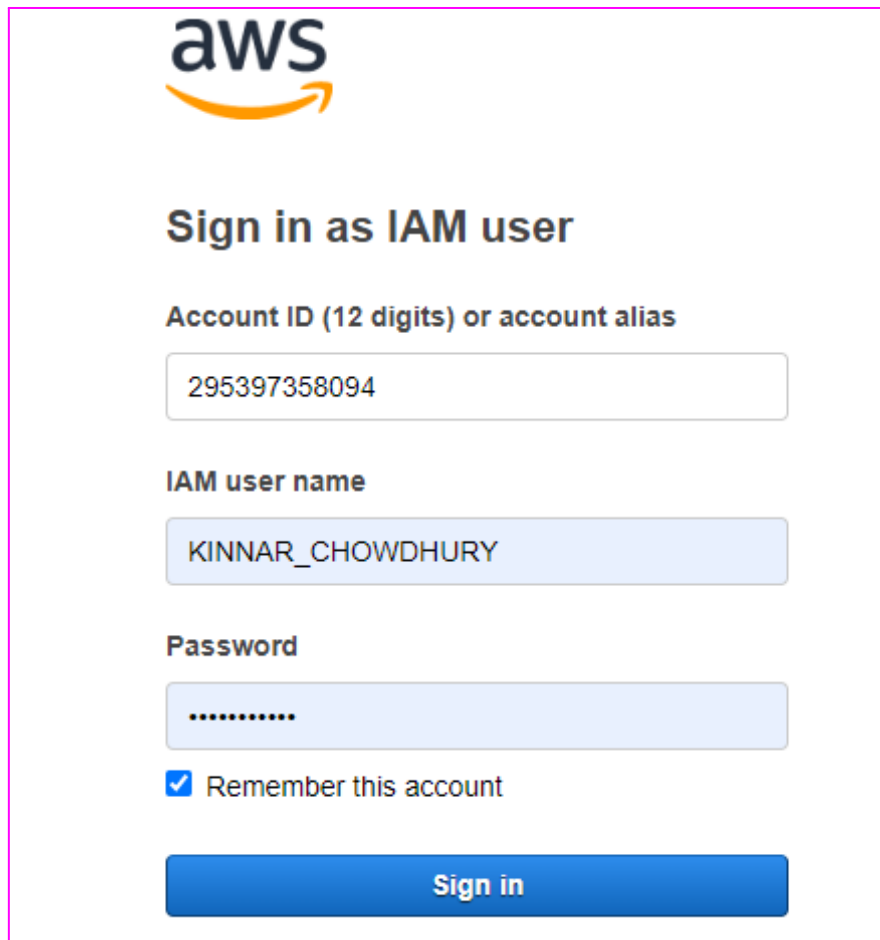
```

- Push it to a GitHub repository

1. Created a git repository - **vlearn\_project\_ci\_cd\_pipeline\_b4**
2. Repository Link: [https://github.com/KinnarChowdhury1994/vlearn\\_project\\_ci\\_cd\\_pipeline\\_b4](https://github.com/KinnarChowdhury1994/vlearn_project_ci_cd_pipeline_b4)

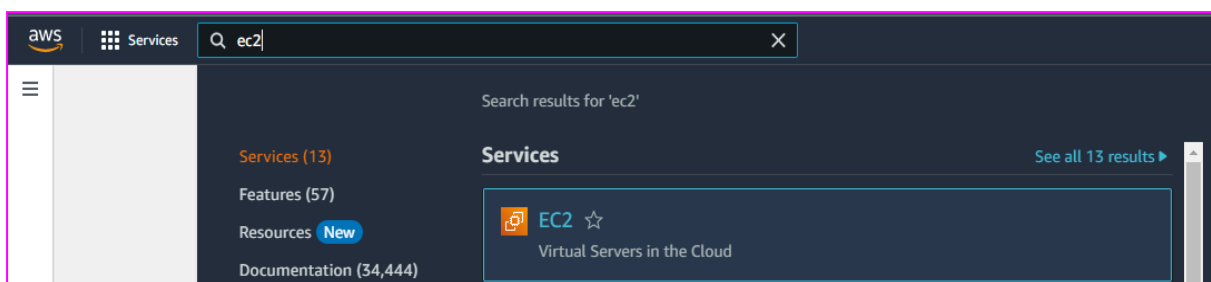
## Task 2 - Set Up an AWS EC2/Local Linux Instance with Nginx

- Set Up an AWS EC2. Login to your AWS Account and Create an EC2 Instance



The image shows the AWS IAM user login interface. At the top is the AWS logo. Below it is the heading "Sign in as IAM user". There are three input fields: "Account ID (12 digits) or account alias" with the value "295397358094", "IAM user name" with the value "KINNAR\_CHOWDHURY", and "Password" with masked characters ".....". Below the password field is a checkbox labeled "Remember this account" which is checked. At the bottom is a blue "Sign in" button.

1. Search for EC2 in the AWS console dashboard and click on it.



## 2. Click on Launch EC2 Instances

**Resources**

EC2 Global view

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running) 1	Auto Scaling Groups API Error	Dedicated Hosts API Error
Elastic IPs API Error	Instances 5	Key pairs API Error
Load balancers API Error	Placement groups API Error	Security groups API Error
Snapshots API Error	Volumes API Error	

**Launch instance**

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

Launch instance

Migrate a server

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

**Service health**

AWS Health Dashboard

Region  
Asia Pacific (Mumbai)

**Zones**

## 3. Provide Name and Tags.

# Launch an instance

Info

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

kinnar\_ec2\_UbuntuLinux\_DevOps\_B4

Add additional tags

#### 4. Choose an OS Image for Amazon Machine Image.

▼ 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

Q Search our full catalog including 1000s of application and OS images

Recents

My AMIs

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUS

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-05fb0b8c1424f266b (64-bit (x86)) / ami-0748d13ffbc370c2b (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-12-07

Architecture

AMI ID

Verified provider

64-bit (x86)

ami-05fb0b8c1424f266b

Verified provider

#### 5. Choose Instance Type.

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0116 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

SUBMITTED BY- KINNAR CHOWDHURY, Hero Vired, DevOps, Batch 4


6. You can use an existing key pair or create an EC2 key pair

▼ 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*

kinnar\_ec2\_key ▼

 Create new key pair

Create key pair

×

Key pair name

Key pairs allow you to connect to your instance securely.

kinnar\_ec2\_key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type



☐ RSA  
RSA encrypted private and public key pair

☒ ED25519  
ED25519 encrypted private and public key pair

Private key file format

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

 When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#) 

Cancel

Create key pair



7. Set up Network settings as per your requirements.

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-02c848fd2d7a8634e

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-14' 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

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

×

8. Allocate storage for your ec2

▼ Configure storage [Info](#)

Advanced

1x  GiB  ▼ Root volume (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

×

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

⊗ Failed to fetch policy coverage

↻

0 x File systems

Edit

9. Now review your setting and click on Launch Instance and wait for some time to be provisioned by AWS.

▼ Summary

Number of instances

Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-05fb0b8c1424f266b

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

Launch instance

Review commands

10. After successfully launching, Check the EC2 running dashboard to get the EC2 IP and other details.

✓ Success

Successfully initiated launch of instance ([i-07979dbaf6cc0a7c5](#))

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Any state

Connect

Instance state

Actions

Launch instances

Instance ID

Clear filters

< 1 >

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
kinrar_ec2_UbuntuLinux_DevOps_...	i-07979dbaf6cc0a7c5	Running	t2.micro	Initializing	User: amaws...	us-east-2c	ec2-3-15-43-39.us-east...	3.15.43.39

Instance: i-07979dbaf6cc0a7c5 (kinrar\_ec2\_UbuntuLinux\_DevOps\_B4)

Details

Status and alarms New

Monitoring

Security

Networking

Storage

Tags

▼ Instance summary Info

Instance ID

i-07979dbaf6cc0a7c5 (kinrar\_ec2\_UbuntuLinux\_DevOps\_B4)

IPv6 address

-

Hostname type

IP name: ip-172-31-32-207.us-east-2.compute.internal

Answer private resource DNS name

IPv4 (A)

Auto-assigned IP address

3.15.43.39 [Public IP]

IAM Role

-

Public IPv4 address

3.15.43.39 [open address]

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-32-207.us-east-2.compute.internal

Instance type

t2.micro

VPC ID

vpc-02c848fd2d7a8634e

Subnet ID

subnet-02f192d52f1c25421

Private IPv4 addresses

172.31.32.207

Public IPv4 DNS

ec2-3-15-43-39.us-east-2.compute.amazonaws.com [open address]

Elastic IP addresses

-

AWS Compute Optimizer finding

User: amaws:iam::295397358094:user/KINNAR\_CHOWDHURY is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: \* because no i identify-based policy allows the compute-optimizer:GetEnrollmentStatus action

Auto Scaling Group name

-

SUBMITTED BY- KINNAR CHOWDHURY, Hero Vired, DevOps, Batch 4

11. Now connect to your aws instance as follows.

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID  
i-07979dbaf6cc0a7c5 (kinnar\_ec2\_UbuntuLinux\_DevOps\_B4)

1. Open an SSH client.

2. Locate your private key file. The key used to launch this instance is kinnar\_ec2\_key.pem

3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 "kinnar\_ec2\_key.pem"

4. Connect to your instance using its Public DNS:  
ec2-3-15-43-39.us-east-2.compute.amazonaws.com

Example:  
ssh -i "kinnar\_ec2\_key.pem" ubuntu@ec2-3-15-43-39.us-east-2.compute.amazonaws.com

Note:

In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

- Deploy an application with Nginx

1. SSH into EC2 to access the console access using the private key downloaded earlier.
2. Locate your private key file. The key used to launch this instance is **kinnar\_ec2\_key.pem**
3. Run this command, if necessary, to ensure your key is not publicly viewable.

```
chmod 400 "kinnar_ec2_key.pem"
```

4. Connect to your instance using its Public DNS:

eg:- ec2-18-188-118-13.us-east-2.compute.amazonaws.com

```
ssh -i "kinnar_ec2_key.pem" ubuntu@ec2-18-188-118-13.us-east-2.compute.amazonaws.com
```

```
kinnar@DESKTOP-TK108QA:~$ cd aws
kinnar@DESKTOP-TK108QA:~/aws$ ls
kinnar_ec2_key.pem  kinnar_ec2_key.pem:Zone.Identifier
kinnar@DESKTOP-TK108QA:~/aws$ sudo chmod 400 "kinnar_ec2_key.pem"
kinnar@DESKTOP-TK108QA:~/aws$ ssh -i "kinnar_ec2_key.pem" ubuntu@ec2-18-188-118-13.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-188-118-13.us-east-2.compute.amazonaws.com (18.188.118.13)' can't be established.
ED25519 key fingerprint is SHA256:c/4278ZK1/eDMzfLwNYJ6f7YGz3GYrsjfYUauZUFvQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-188-118-13.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.2.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Feb 17 12:20:26 UTC 2024

System load:  0.78857421875   Processes:           100
Usage of /:   7.9% of 28.89GB   Users logged in:     0
Memory usage: 20%             IPv4 address for eth0: 172.31.36.196
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
 * compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Feb 17 12:08:11 2024 from 3.16.146.5
ubuntu@ip-172-31-36-196:~$ |
```

```

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Sun Feb 11 15:32:47 UTC 2024

System load:  0.0           Processes:            95
Usage of /:   20.6% of 7.57GB Users logged in:          0
Memory usage: 21%          IPv4 address for eth0: 172.31.32.207
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-32-207:~$

```

1. Update OS using the command **sudo apt update && sudo apt upgrade -y**

```

Last login: Sat Feb 17 12:08:11 2024 from 3.16.146.5
ubuntu@ip-172-31-36-196:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 1s (149 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
ubuntu@ip-172-31-36-196:~$ |

```

2. Install Network Tools using the command **sudo apt install net-tools**

```

ubuntu@ip-172-31-36-196:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
net-tools is already the newest version (1.60+git20181103.0eebece-1ubuntu5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-36-196:~$ |

```

3. Install nginx using the command `sudo apt install -y nginx` and check the version using `nginx -v` to validate your installation.

```
ubuntu@ip-172-31-36-196:~$ sudo apt install -y nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-36-196:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-36-196:~$
```

4. Now use **sudo systemctl status nginx** to check if the nginx service is running as a daemon.

```
ubuntu@ip-172-31-36-196:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-36-196:~$ 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 Sat 2024-02-17 12:27:35 UTC; 2min 53s ago
     Docs: man:nginx(8)
    Process: 1489 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 1490 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 1579 (nginx)
    Tasks: 2 (limit: 1121)
   Memory: 6.5M
      CPU: 29ms
   CGroup: /system.slice/nginx.service
           └─1579 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─1582 "nginx: worker process" .....
```

```
Feb 17 12:27:35 ip-172-31-36-196 systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 17 12:27:35 ip-172-31-36-196 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-36-196:~$
```

5. Goto nginx directory - **cd /etc/nginx/** and make necessary configuration for web server and hosting websites.

```
ubuntu@ip-172-31-36-196:~$ cd /etc/nginx/
conf.d/          modules-available/ modules-enabled/  sites-available/ sites-enabled/    snippets/
ubuntu@ip-172-31-36-196:~$ cd /etc/nginx/
```

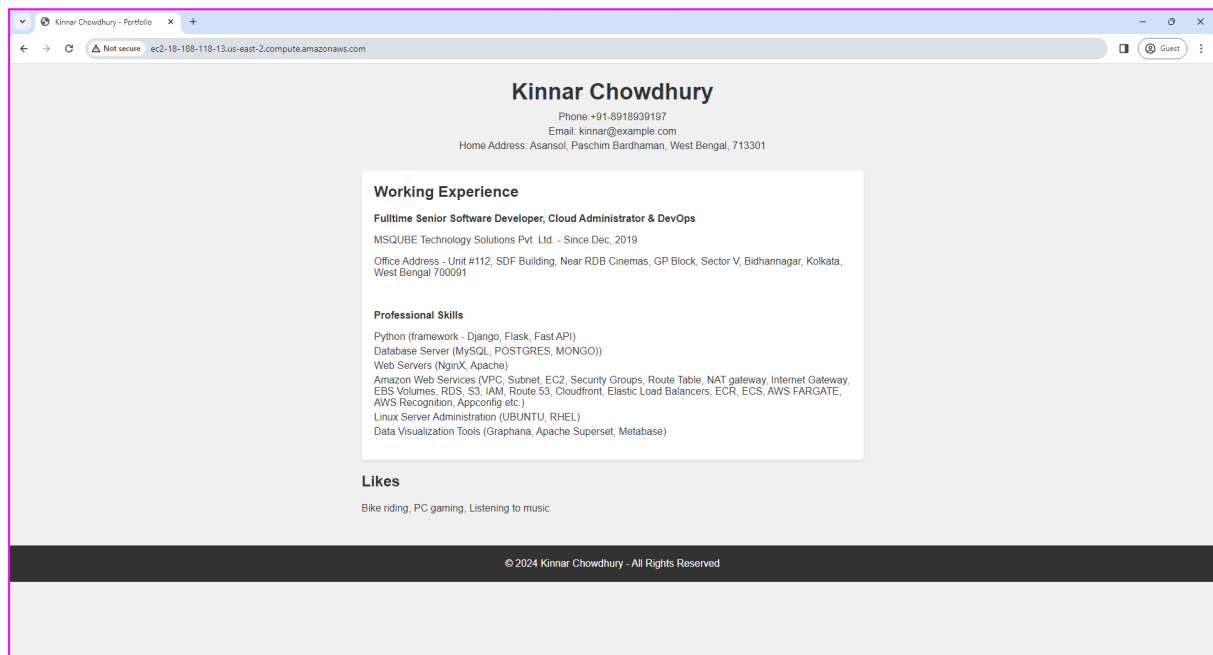
## 6. Go to `/var/www/html/` and put the index.html

```
ubuntu@ip-172-31-36-196:~$ history
1  sudo apt update
2  sudo apt upgrade
3  sudo apt upgrade
4  sudo apt install build-essential
5  sudo apt install net-tools
6  sudo reboot
7  sudo apt update
8  sudo apt install net-tools
9  sudo apt install -y nginx
10 clear
11 sudo apt install -y nginx
12 nginx -v
13 sudo systemctl status nginx
14 ls
15 mkdir hosting_sites
16 mkdir hosting_sites/17022024
17 mv index.html hosting_sites/17022024/
18 cd hosting_sites/17022024/
19 ls
20 cd ..
21 mkdir backup
22 cd backup/
23 sudo cp -r /var/www/html .
24 ls
25 sudo mv html html_originak.bkp
26 sudo mv html_originak.bkp html_default.bkp
27 ls
28 sudo cp -r /etc/nginx . && sudo mv nginx nginx_default.bkp
29 ls -l
30 cd nginx_default.bkp/
31 ls
32 cd ..
33 ls
34 cd 17022024/
35 ls
36 sudo rm -rf /var/www/html/* && cp -r index.html /var/www/html/
37 sudo rm -rf /var/www/html/* && sudo cp -r index.html /var/www/html/
38 systemctl restart nginx
39 sudo systemctl restart nginx
40 sudo systemctl status nginx
41 history
42 cd
43 clear
44 sudo netstat -tulpn | grep LISTEN
45 history
ubuntu@ip-172-31-36-196:~$ |
```

## 7. Check ports

```
ubuntu@ip-172-31-36-196:~$ sudo netstat -tulpn | grep LISTEN
tcp        0      0 127.0.0.53:53         0.0.0.0:*           LISTEN      304/systemd-resolve
tcp        0      0 127.0.0.1:6010        0.0.0.0:*           LISTEN      1790/sshd: ubuntu@p
tcp        0      0 0.0.0.0:80           0.0.0.0:*           LISTEN      6577/nginx: master
tcp        0      0 0.0.0.0:22           0.0.0.0:*           LISTEN      521/sshd: /usr/sbin
tcp6       0      0 :::6010              :::*                LISTEN      1790/sshd: ubuntu@p
tcp6       0      0 :::80                :::*                LISTEN      6577/nginx: master
tcp6       0      0 :::22                :::*                LISTEN      521/sshd: /usr/sbin
ubuntu@ip-172-31-36-196:~$ |
```

8. Go to <http://ec2-18-188-118-13.us-east-2.compute.amazonaws.com/>



### Task 3 - Write a Python Script to Check for New Commits

- Create a Python script to check for new commits using the GitHub API.

**Note:-** Python Script is pushed to the repository.

#### Task 4 - Write a Bash Script to Deploy the Code

- Create a bash script to clone the latest code and restart Nginx.

##### \_deploy.sh

```
#!/bin/bash

cd /home/ubuntu/repo/
ls
cd vlearn_project_ci_cd_pipeline_b4
ls -la
git pull
git branch
ls
git checkout main
git pull
rm -rf /var/www/html/*
cp -r index.html /var/www/html/

sudo systemctl restart nginx

echo "Thank You Deployment Completed"
```

#### Task 5 - Set Up a Cron Job to Run the Python Script

- Create a cron job to run the Python script at regular intervals.

Created a shell file “\_cronScript.sh” in the **/home/ubuntu/automation/** directory.

##### \_cronScript.sh

```
#!/bin/sh
cd /home/ubuntu/repo/
ls
cd vlearn_project_ci_cd_pipeline_b4
ls -la

python automate.py
```

##### **crontab -l**

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
```



```
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command

*/2 * * * *      /home/ubuntu/automation/_cronScript.sh >>
/home/ubuntu/cronjobs/log/success.log 2> /home/ubuntu/cronjobs/log/error.log >>
/home/ubuntu/cronjobs/log/debug.log && sudo cp -r
/home/ubuntu/cronjobs/log/debug.log /home/ubuntu/cronjobs/log/debug.txt

ubuntu@ip-172-31-36-196:~/cronjobs/log$ pwd
/home/ubuntu/cronjobs/log
ubuntu@ip-172-31-36-196:~/cronjobs/log$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 832 Feb 17 13:56 debug.log
-rw-rw-r-- 1 ubuntu ubuntu  0 Feb 17 13:56 error.log
-rw-rw-r-- 1 ubuntu ubuntu  0 Feb 17 13:50 success.log
```

## Task 6 - Test the Setup

- Make a new commit to the GitHub repository and check that the changes are automatically deployed.

## References:

1. <https://www.google.com/>
2. <https://stackoverflow.com/>
3. <https://docs.github.com/en/rest?apiVersion=2022-11-28>
4. <https://chat.openai.com/>

THANK YOU

===== End of File =====