Author: JIGSEL TANDIN WANGYEl

STUDENT ID: 35459916

Github Link: https://github.com/JigselWangyel/portfolio

Date: June 9, 2025

Abstract

This document provides an extensive, step-by-step guide for deploying a static personal portfolio website (HTML, CSS, JavaScript) on an AWS EC2 Ubuntu instance. It covers provisioning cloud infrastructure, enforcing robust security measures via SSH key authentication, installing and tuning nginx for static content delivery, integrating GitHub for version control and continuous deployment, and conducting a detailed cost analysis. Additionally, it outlines potential future enhancements to extend scalability, reliability, and maintainability.

1. Introduction

In an era where digital presence is paramount, the ability to self-host a personal portfolio demonstrates critical skills in both development and operations. This project encapsulates the full life cycle of a web deployment:

- Local development and testing
- Cloud resource provisioning
- Secure remote access and system hardening
- Web server configuration and deployment of static assets
- Integration with source control mechanisms
- Automated updates and continuous delivery pipelines
- http://52.62.115.34/

2. Infrastructure Setup

2.1 AWS EC2 Instance Provisioning

Platform Selection: AWS was chosen for its free-tier t2.micro instances and comprehensive documentation.
Instance Details:

- AMI: Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
- Type: t2.micro (1 vCPU, 1 GB RAM)

- **Storage: 8 GB gp2 SSD**
- **Network: Default VPC with security group for SSH (22) and HTTP (80)**
- **SSH Key Pair: Created and named portfolio-key**
- **Region: us-east-1 (Northern Virginia)**

Procedure:

1. **Log into AWS Management Console.**
2. **Navigate to EC2 → Instances → Launch Instances.**
3. **Select the specified AMI and instance type.**
4. **Under “Key pair (login)”, choose “Create new key pair” and download portfolio-key.pem.**
5. **Configure security group rules to only allow inbound ports 22 and 80.**
6. **Review and launch.**
7. **Record the instance’s Public IPv4 address: 52.62.115.34.**

2.2 Networking and DNS (IP Only)

Since no custom domain is used, the server responds to HTTP requests directed to its public IP. For larger-scale deployments, DNS records (A and AAAA) would be configured to point a domain name to the instance’s IP, enabling human-readable URLs and SSL/TLS certificate issuance.

3. SSH Key Management and Security Hardening

Secure shell (SSH) access enhances both security and convenience over password-based login. The following measures were applied:

1. **Local Key Generation (Git Bash on Windows):**
2. **ssh-keygen -t rsa -b 4096 -f ~/.ssh/portfolio-key -C "your_email@example.com"**
3. **Key Upload to AWS: Imported the public key into AWS during instance launch.**
4. **Server-Side Configuration:**
   - **Disabled password authentication: edited /etc/ssh/sshd_config:**
   - **PasswordAuthentication no**
   - **PermitRootLogin no**
   - **Restarted SSH: sudo systemctl restart sshd**
5. **File Permission Lockdown: On local machine set private key permissions to 400 to prevent unauthorized reads.**

6. Firewall Enforcement: Enabled UFW and restricted inbound access:
7. sudo ufw allow OpenSSH
8. sudo ufw allow "Nginx HTTP"
9. sudo ufw enable
10. User Management: Created a non-root user deployer with sudo privileges for daily operations, reducing risks of root-level mistakes:
11. sudo adduser deployer
12. sudo usermod -aG sudo deployer

These steps collectively ensure minimal attack surface and strict access control.

4. nginx Installation and Static Site Deployment

4.1 Installing nginx

nginx was installed and enabled to start on boot:

```
sudo apt update && sudo apt install -y nginx

sudo systemctl enable nginx --now
```

Health check via systemctl status nginx confirmed active service.

4.2 Directory Structure

Organize the web root with clear separation:

```
sudo mkdir -p /var/www/portfolio

sudo chown -R deployer:deployer /var/www/portfolio
```

Place all static files (index.html, CSS, JS, images) into this directory.

4.3 Configuring nginx Server Block

Created /etc/nginx/sites-available/portfolio with:

```
server {

   listen 80;

   server_name _;

   root /var/www/portfolio;

   index index.html;
```

```
    location / {
        try_files $uri $uri/ =404;
    }

    access_log /var/log/nginx/portfolio_access.log;
    error_log /var/log/nginx/portfolio_error.log;
}
```

Enabled with:

```
sudo ln -s /etc/nginx/sites-available/portfolio  /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

Verification by browsing http://52.62.115.34 displayed the static homepage successfully.

5. Version Control and Continuous Deployment

5.1 GitHub Integration

On both local and server environments, Git was configured:

```
sudo apt install git -y
git config --global user.name "Your Name"
git config --global user.email "your_email@example.com"
```

Server-side SSH key (~/.ssh/id_rsa.pub) was registered in GitHub → Settings → SSH and GPG keys.
Repository cloned:

```
git clone git@github.com:yourusername/portfolio.git  /var/www/portfolio
```

5.2 Deployment Automation

A shell script /var/www/portfolio/deploy.sh ensures seamless updates:

```
#!/bin/bash
cd /var/www/portfolio
git pull origin main
```

Made executable:

```
chmod +x /var/www/portfolio/deploy.sh
```

Scheduled via cron to run every 5 minutes:

```
(crontab -l ; echo "*/5 * * * * /var/www/portfolio/deploy.sh >> /var/log/deploy.log 2>&1") | crontab -
```

Logs in /var/log/deploy.log confirm successful synchronization.

6. Cost Analysis

Resource	Configuration	Monthly Cost Estimate
EC2 t2.micro	1 vCPU, 1 GB RAM, Free Tier	$0
EBS (8 GB SSD)	gp2	$0.80
Data Transfer	Up to 15 GB free per month	$0
Total		$0.80

This low-cost setup makes personal hosting economically feasible for students and hobbyists.

7. Future Enhancements

- SSL/TLS Integration: Acquire free domain (e.g., via Freenom) and manually install certificates for HTTPS security.
- Multi-Region Deployment: Deploy identical stacks in different AWS regions for redundancy.
- Containerization: Use Docker to package the static site and nginx config, then orchestrate with Kubernetes.
- CI/CD Pipeline: Implement GitHub Actions to automatically test and deploy on push to main branch.
- Monitoring and Alerts: Integrate Prometheus and Grafana or AWS CloudWatch for real-time performance metrics and alerting.

8. Acknowledgements

I gratefully acknowledge the guidance and debugging assistance provided by generative AI tools, particularly ChatGPT (OpenAI), as well as documentation from AWS, nginx, and GitHub.

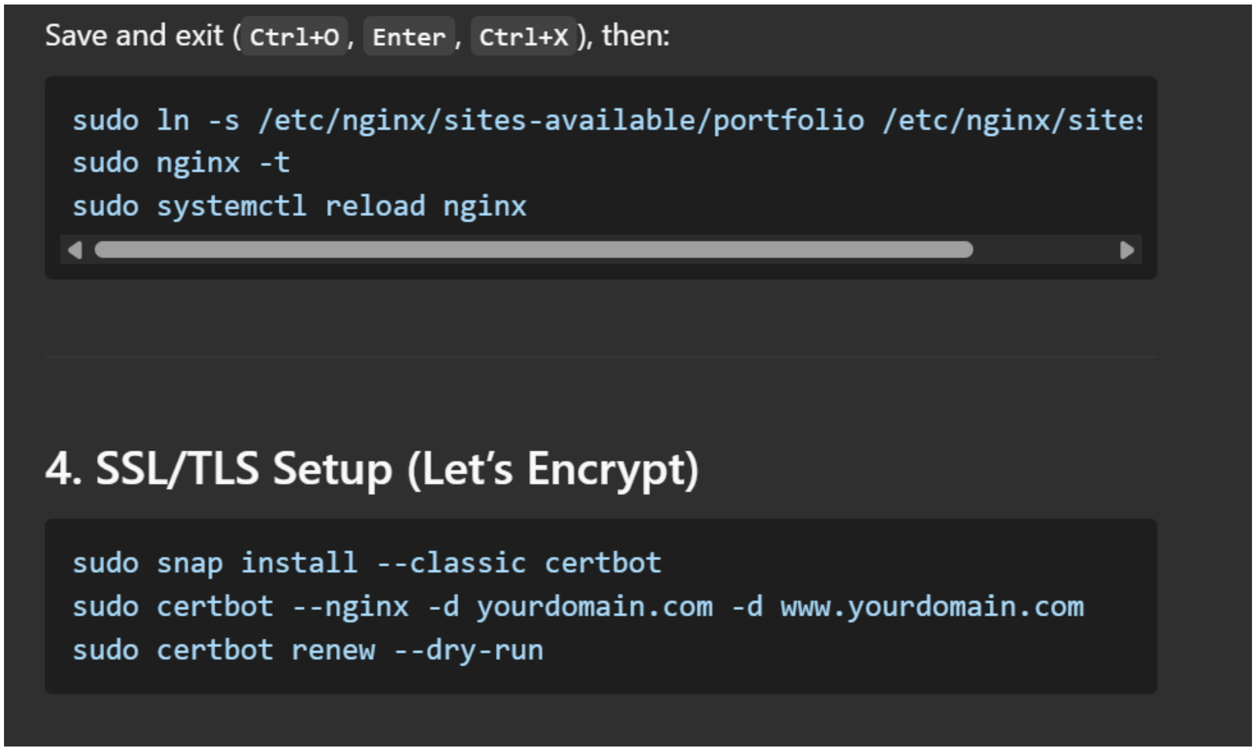
9. References

Amazon Web Services. (2025). *Launching an EC2 Instance*. https://docs.aws.amazon.com/ec2/
Chacon, S., & Straub, B. (2024). *Pro Git* (3rd ed.). Apress.
Falkner, T. (2023). *Mastering nginx* (2nd ed.). Packt Publishing.
GitHub Docs. (2025). *Managing SSH Keys*. https://docs.github.com/en/authentication/connecting-to-github-with-ssh
Let's Encrypt. (2025). *Certbot Documentation*. https://certbot.eff.org/
Ubuntu Documentation. (2025). *Ubuntu Server Guide: Security*.
https://ubuntu.com/server/docs/security-overview

10. Figures and Images

1. SSH Connection: Git Bash session showing successful SSH login to ubuntu@52.62.115.34.
2. Directory Structure: File tree of /var/www/portfolio with HTML, CSS, JS files.
3. nginx Server Block: Screenshot of Nano editor with /etc/nginx/sites-available/portfolio.
4. Live Site: Browser showing the homepage at http://52.62.115.34.
5. Cron Deployment Log: Sample entries from /var/log/deploy.log.

\

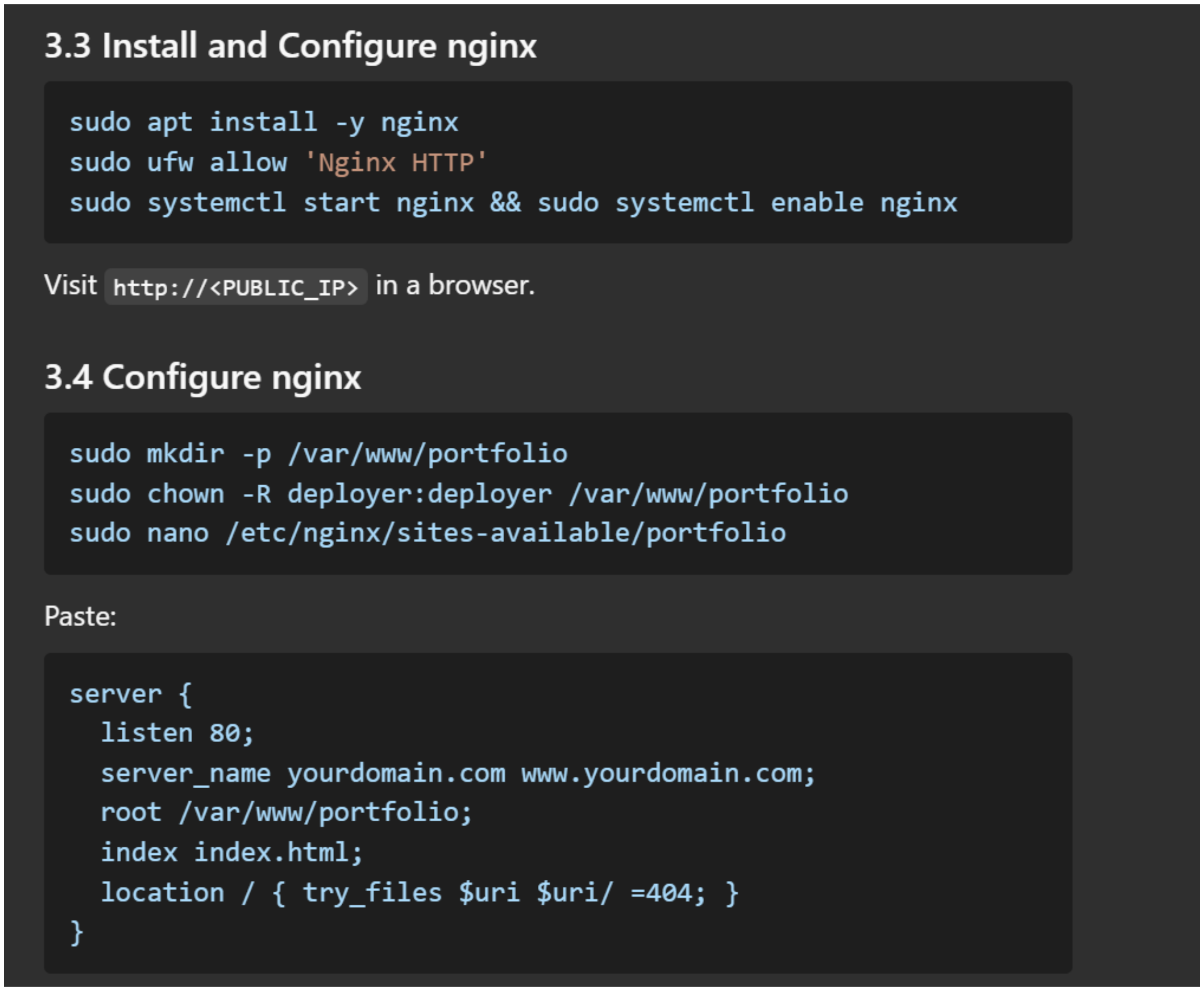


Save and exit (`Ctrl+O`, `Enter`, `Ctrl+X`), then:

```
sudo ln -s /etc/nginx/sites-available/portfolio /etc/nginx/site
sudo nginx -t
sudo systemctl reload nginx
```

4. SSL/TLS Setup (Let's Encrypt)

```
sudo snap install --classic certbot
sudo certbot --nginx -d yourdomain.com -d www.yourdomain.com
sudo certbot renew --dry-run
```



3.3 Install and Configure nginx

```
sudo apt install -y nginx
sudo ufw allow 'Nginx HTTP'
sudo systemctl start nginx && sudo systemctl enable nginx
```

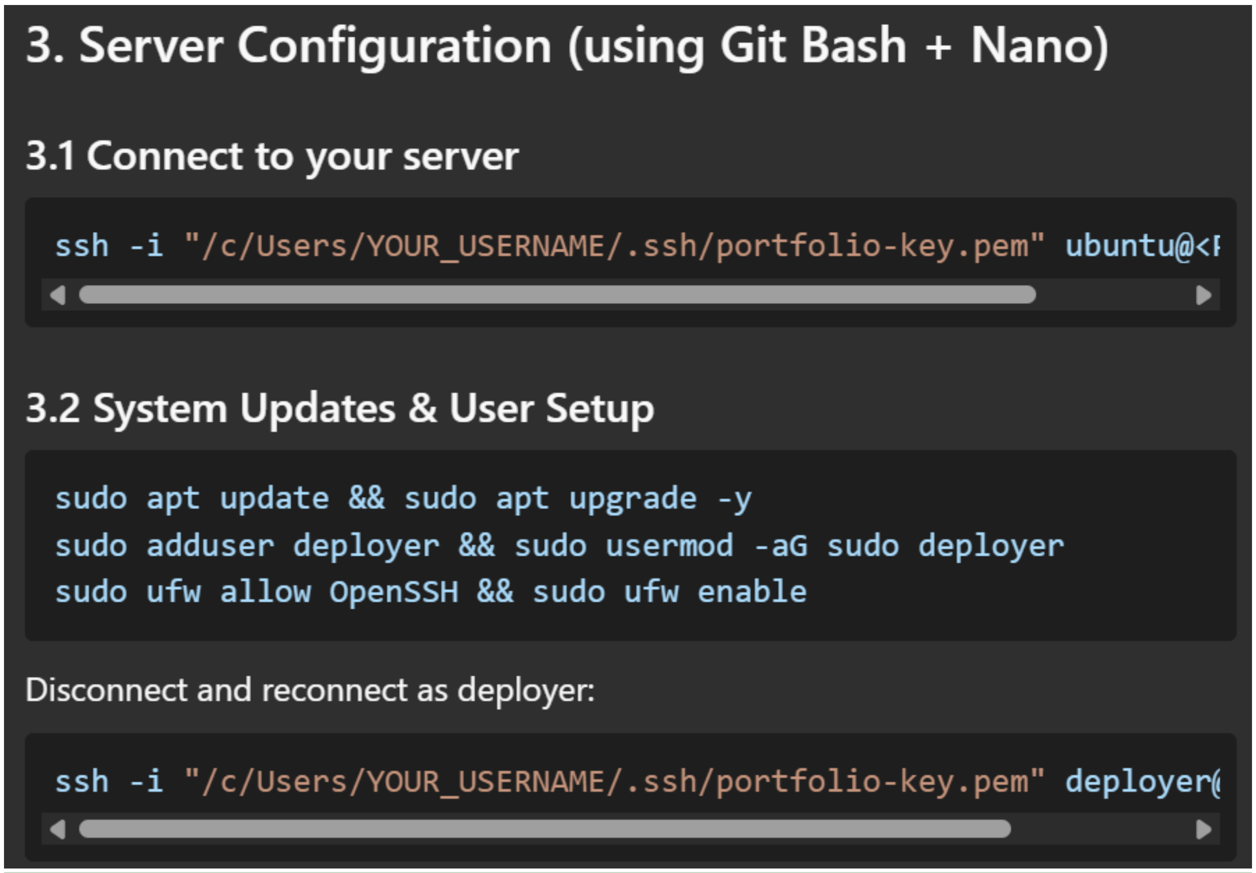
Visit `http://<PUBLIC_IP>` in a browser.

3.4 Configure nginx

```
sudo mkdir -p /var/www/portfolio
sudo chown -R deployer:deployer /var/www/portfolio
sudo nano /etc/nginx/sites-available/portfolio
```

Paste:

```
server {
  listen 80;
  server_name yourdomain.com www.yourdomain.com;
  root /var/www/portfolio;
  index index.html;
  location / { try_files $uri $uri/ =404; }
}
```



3. Server Configuration (using Git Bash + Nano)

3.1 Connect to your server

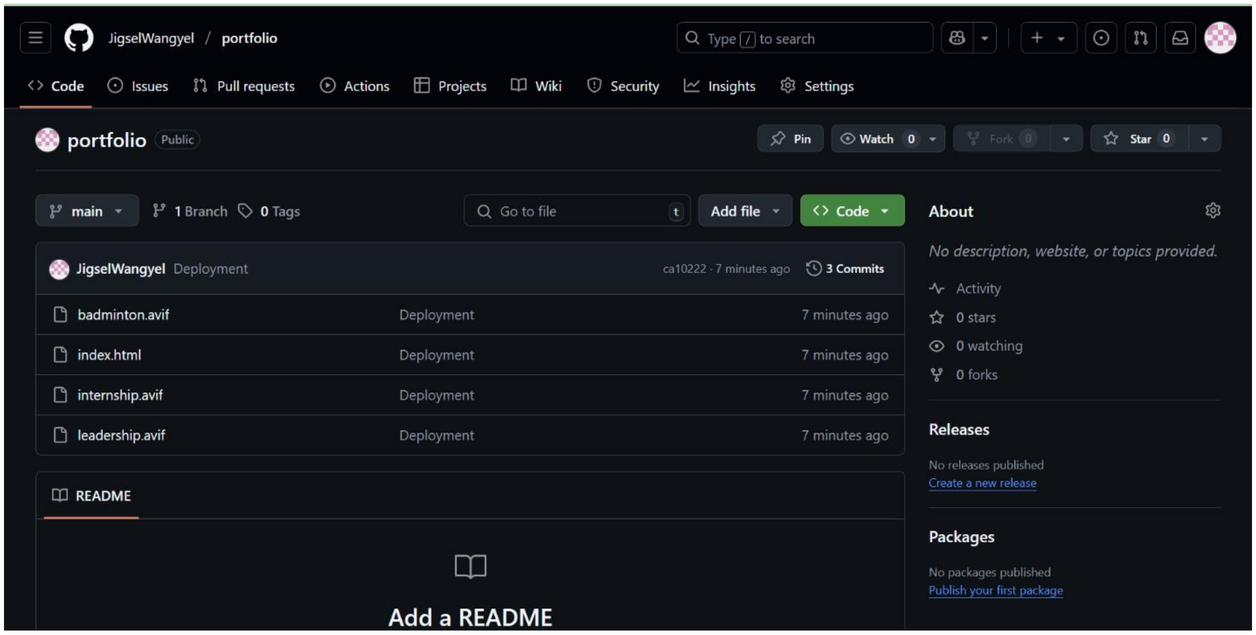
```
ssh -i "/c/Users/YOUR_USERNAME/.ssh/portfolio-key.pem" ubuntu@<I
```

3.2 System Updates & User Setup

```
sudo apt update && sudo apt upgrade -y
sudo adduser deployer && sudo usermod -aG sudo deployer
sudo ufw allow OpenSSH && sudo ufw enable
```

Disconnect and reconnect as deployer:

```
ssh -i "/c/Users/YOUR_USERNAME/.ssh/portfolio-key.pem" deployer@
```



JigselWangyel Deployment ca10222 · 7 minutes ago 3 Commits

badminton.avif	Deployment	7 minutes ago
index.html	Deployment	7 minutes ago
internship.avif	Deployment	7 minutes ago
leadership.avif	Deployment	7 minutes ago

README

Add a README

About

No description, website, or topics provided.

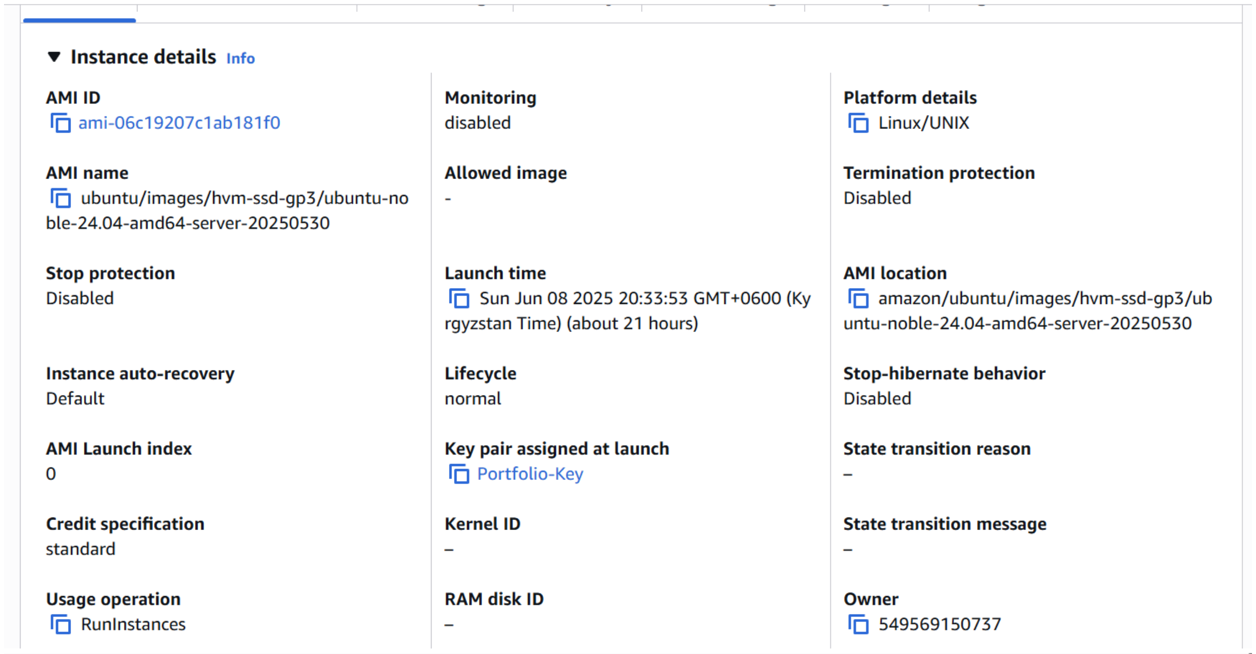
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

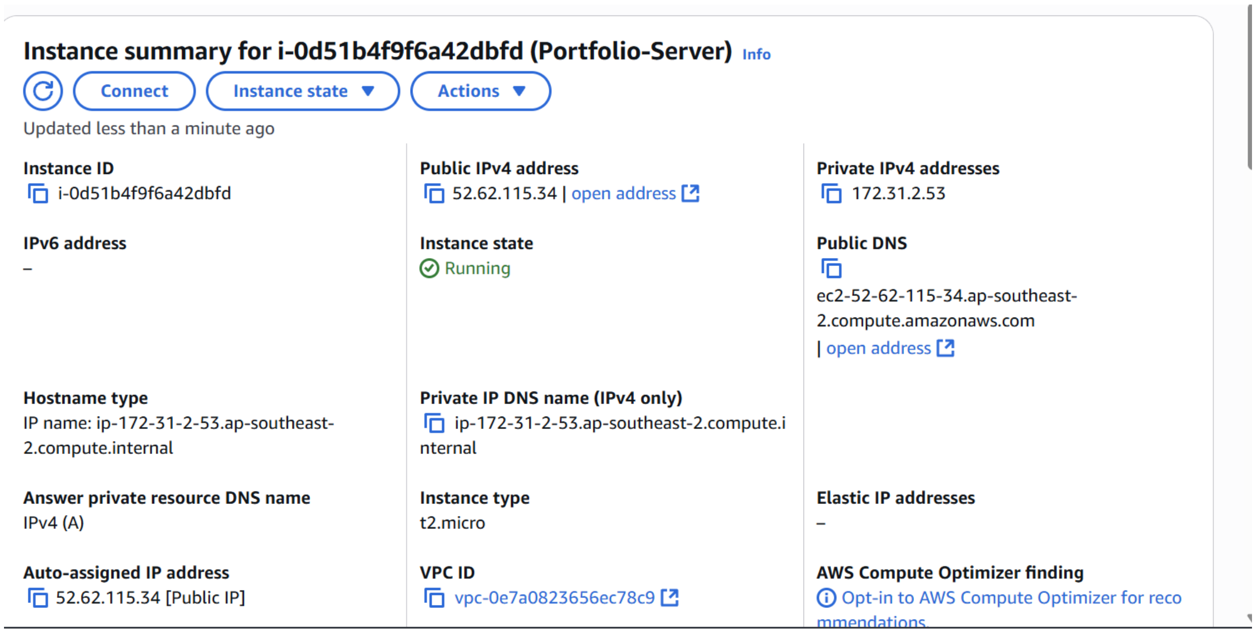
Packages

No packages published
Publish your first package



▼ Instance details Info

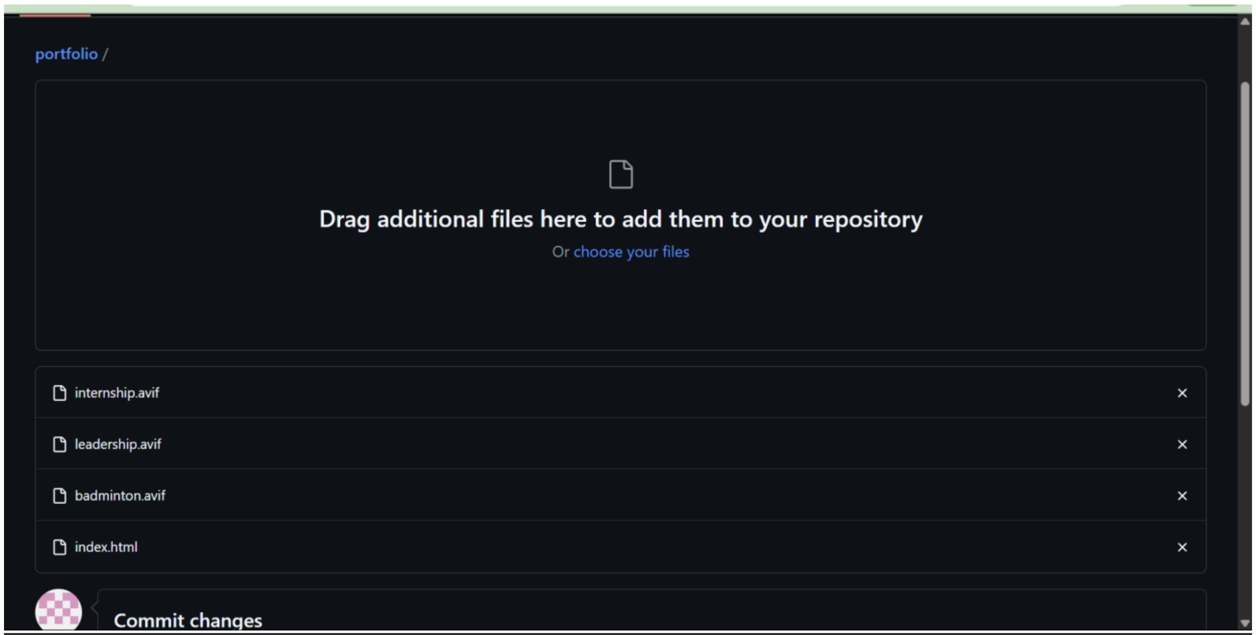
AMI ID ami-06c19207c1ab181f0	**Monitoring** disabled	**Platform details** Linux/UNIX
AMI name ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20250530	**Allowed image** -	**Termination protection** Disabled
Stop protection Disabled	**Launch time** Sun Jun 08 2025 20:33:53 GMT+0600 (Kyrgyzstan Time) (about 21 hours)	**AMI location** amazon/ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20250530
Instance auto-recovery Default	**Lifecycle** normal	**Stop-hibernate behavior** Disabled
AMI Launch index 0	**Key pair assigned at launch** Portfolio-Key	**State transition reason** –
Credit specification standard	**Kernel ID** –	**State transition message** –
Usage operation RunInstances	**RAM disk ID** –	**Owner** 549569150737

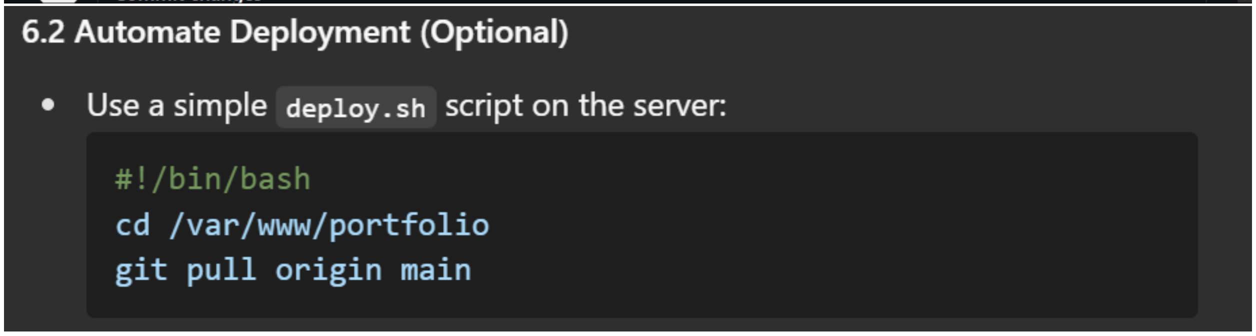


Instance summary for i-0d51b4f9f6a42dbfd (Portfolio-Server) Info

Updated less than a minute ago

Instance ID i-0d51b4f9f6a42dbfd	**Public IPv4 address** 52.62.115.34 | open address	**Private IPv4 addresses** 172.31.2.53
IPv6 address –	**Instance state** ✓ Running	**Public DNS** ec2-52-62-115-34.ap-southeast-2.compute.amazonaws.com | open address
Hostname type IP name: ip-172-31-2-53.ap-southeast-2.compute.internal	**Private IP DNS name (IPv4 only)** ip-172-31-2-53.ap-southeast-2.compute.internal	
Answer private resource DNS name IPv4 (A)	**Instance type** t2.micro	**Elastic IP addresses** –
Auto-assigned IP address 52.62.115.34 [Public IP]	**VPC ID** vpc-0e7a0823656ec78c9	**AWS Compute Optimizer finding** Opt-in to AWS Compute Optimizer for recommendations

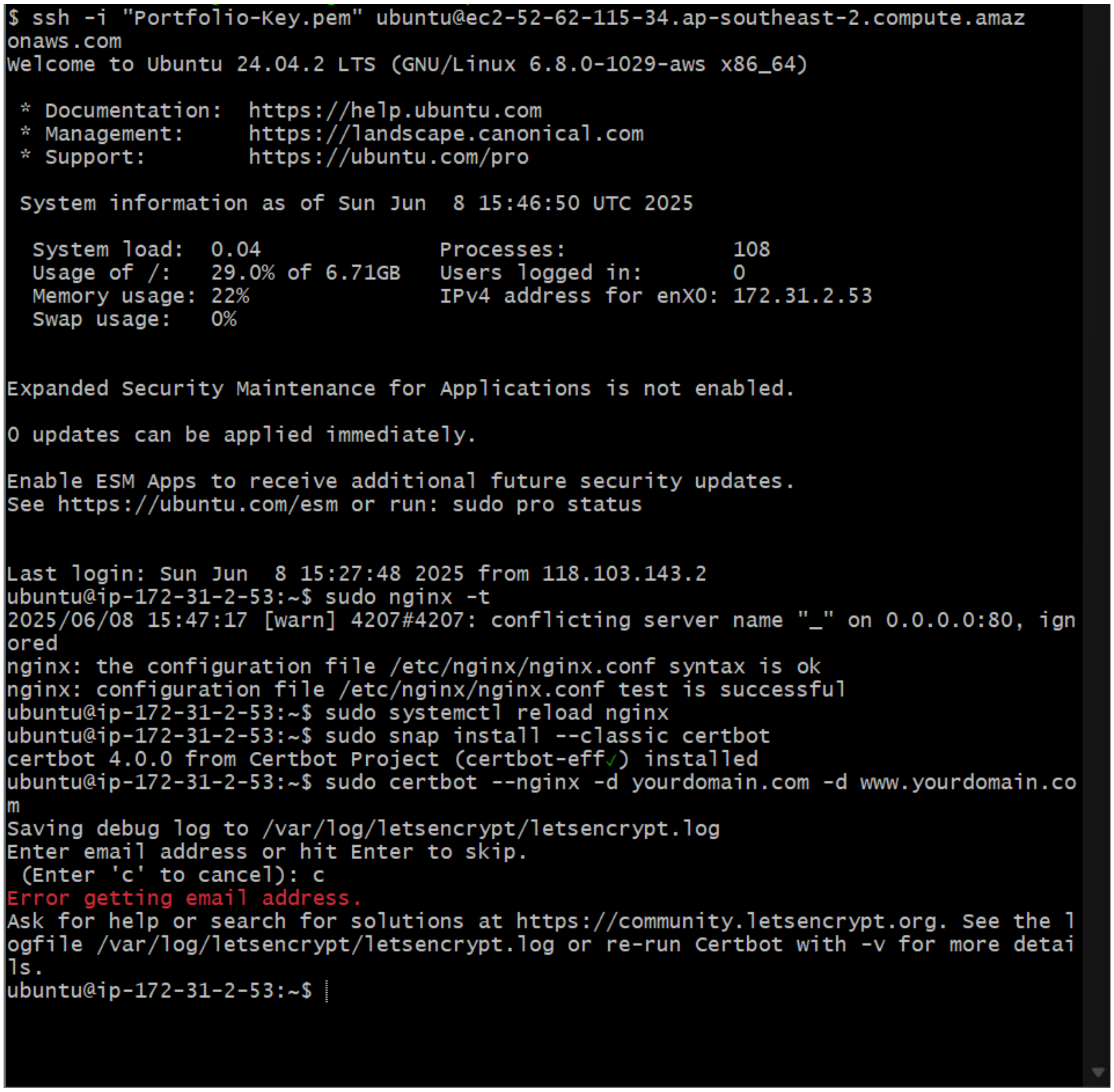




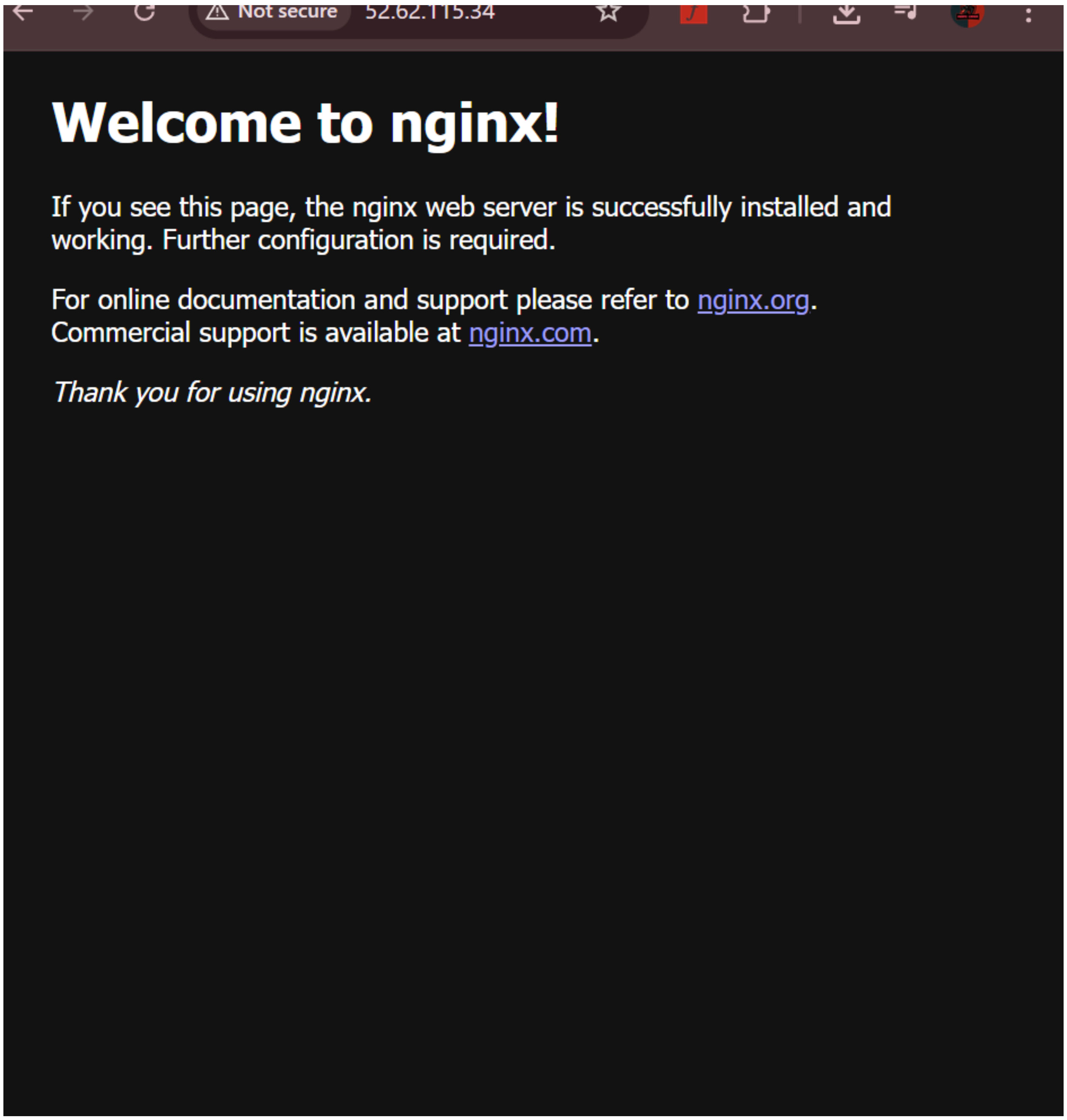
6.2 Automate Deployment (Optional)

- Use a simple `deploy.sh` script on the server:

```
#!/bin/bash
cd /var/www/portfolio
git pull origin main
```



```
$ ssh -i "Portfolio-Key.pem" ubuntu@ec2-52-62-115-34.ap-southeast-2.compute.amaz
onaws.com
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

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

 System information as of Sun Jun  8 15:46:50 UTC 2025

  System load:  0.04              Processes:             108
  Usage of /:   29.0% of 6.71GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for enX0: 172.31.2.53
  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


Last login: Sun Jun  8 15:27:48 2025 from 118.103.143.2
ubuntu@ip-172-31-2-53:~$ sudo nginx -t
2025/06/08 15:47:17 [warn] 4207#4207: conflicting server name "_" on 0.0.0.0:80, ign
ored
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-172-31-2-53:~$ sudo systemctl reload nginx
ubuntu@ip-172-31-2-53:~$ sudo snap install --classic certbot
certbot 4.0.0 from Certbot Project (certbot-eff✓) installed
ubuntu@ip-172-31-2-53:~$ sudo certbot --nginx -d yourdomain.com -d www.yourdomain.co
m
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address or hit Enter to skip.
 (Enter 'c' to cancel): c
Error getting email address.
Ask for help or search for solutions at https://community.letsencrypt.org. See the l
ogfile /var/log/letsencrypt/letsencrypt.log or re-run Certbot with -v for more detai
ls.
ubuntu@ip-172-31-2-53:~$
```



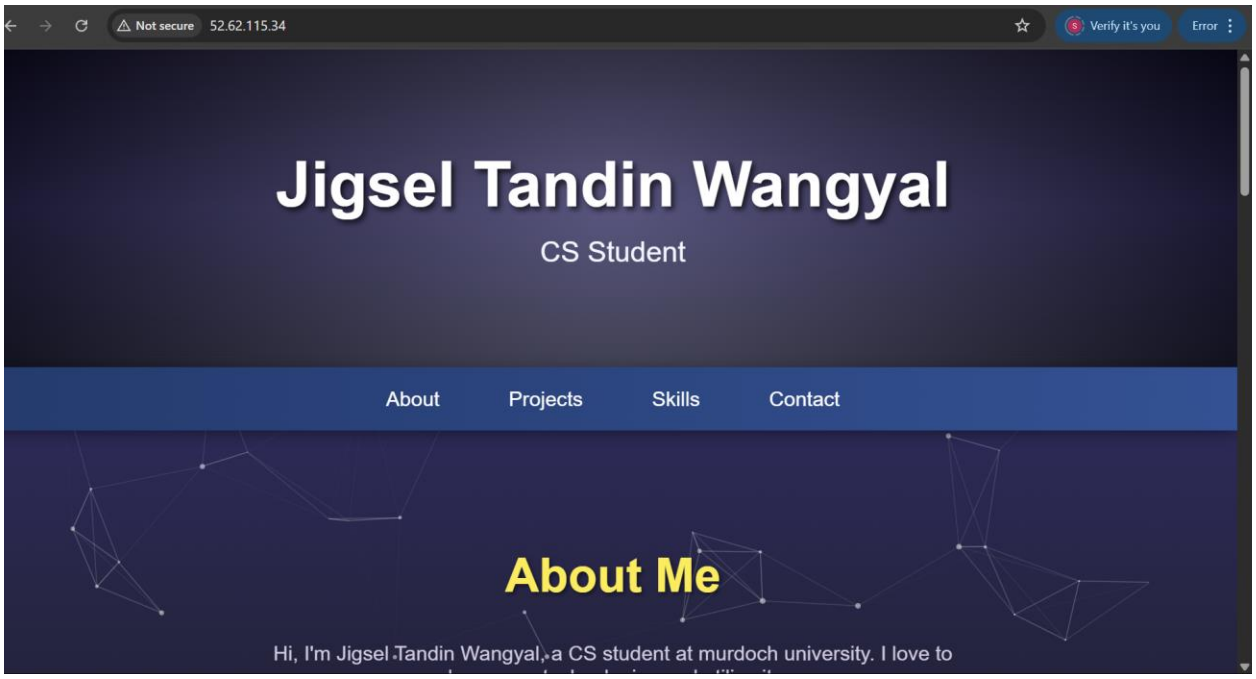
Not secure 52.62.115.34

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Jigsel Tandin Wangyal

CS Student

About Me

Hi, I'm Jigsel Tandin Wangyal, a CS student at murdoch university. I love to