

Step-by-Step Process

1. Install and Configure NGINX

Update the System:

```
sudo apt update && sudo apt upgrade -y
```

Install NGINX:

```
sudo apt install nginx -y
```

Verify Installation:

```
nginx -v
```

2. Install Certbot for SSL

Install Certbot and NGINX Plugin:

```
sudo apt install certbot python3-certbot-nginx -y
```

Generate SSL Certificate:

Replace `example.com` with your domain name.

```
sudo certbot --nginx -d example.com -d www.example.com
```

Test Certificate Renewal:

```
sudo certbot renew --dry-run
```

3. Change Default Branch in GitHub

1. Go to your GitHub repository settings.
 2. Navigate to **Branches**.
 3. Under **Default Branch**, change the default branch to your desired branch.
-

4. Change Ownership to Default User (e.g., ubuntu)

Assign Ownership to **ubuntu**:

```
sudo chown -R ubuntu:ubuntu /path/to/your/code
```

5. Set Up GitHub Actions Runner

Download Runner Package:

```
mkdir actions-runner && cd actions-runner
curl -o actions-runner-linux-x64-2.303.0.tar.gz -L
https://github.com/actions/runner/releases/download/v2.303.0/actions-runner-linux-x64-2.303.0.t
ar.gz
tar xzf actions-runner-linux-x64-2.303.0.tar.gz
```

Configure Runner:

```
./config.sh --url https://github.com/your-repo-owner/your-repo-name --token YOUR_TOKEN
```

Install Runner as a Service:

```
sudo ./svc.sh install
sudo ./svc.sh start
```

6. Add SSH Key

Generate SSH Key:

```
ssh-keygen -t ed25519 -C "your-email@example.com"
```

Press Enter for default file location and passphrase.

Add Key to SSH Agent:

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
```

Copy Public Key to GitHub:

```
cat ~/.ssh/id_ed25519.pub
```

1. Go to GitHub > Settings > SSH and GPG keys > Add new key.
2. Paste the public key.

7. Initialize Git and Configure Settings

Initialize Git:

```
git init
```

Set Username and Email:

```
git config --global user.name "Your Name"  
git config --global user.email "your-email@example.com"
```

8. Configure Runner and Start Service

Verify Runner Service:

```
sudo ./svc.sh status
```

9. Copy CI/CD Code and Modify Paths

1. Copy your previous CI/CD YAML file into `.github/workflows/` in your repository.
 2. Edit the file to match:
 - **Branch:** Update the `on` field to use the correct branch.
 - **Paths:** Adjust paths as necessary for your new setup.
-

10. Set Up NGINX for the Application

Create Site Configuration:

```
sudo nano /etc/nginx/sites-available/your-site
```

Example configuration:

```
server {  
    listen 80;  
    server_name example.com www.example.com;  
  
    location / {  
        proxy_pass http://127.0.0.1:5000;
```

```
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
}
```

Enable Site:

```
sudo ln -s /etc/nginx/sites-available/your-site /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

11. Test and Verify Setup

- Access the application via the domain.
- Check CI/CD pipeline execution in GitHub Actions.
- Verify SSL is working on the domain.