

ภาคผนวก ก

คู่มือการติดตั้ง

การพัฒนาเว็บแอปพลิเคชันบริหารจัดการ ร้านก๋วยเตี๋ยว และ แอปพลิเคชันสามารถติดตั้งบน Amazon Web Services (AWS) นำมาช่วยในการติดตั้งโปรเจกต์ โดยใช้รายละเอียดคำสั่งต่างๆ

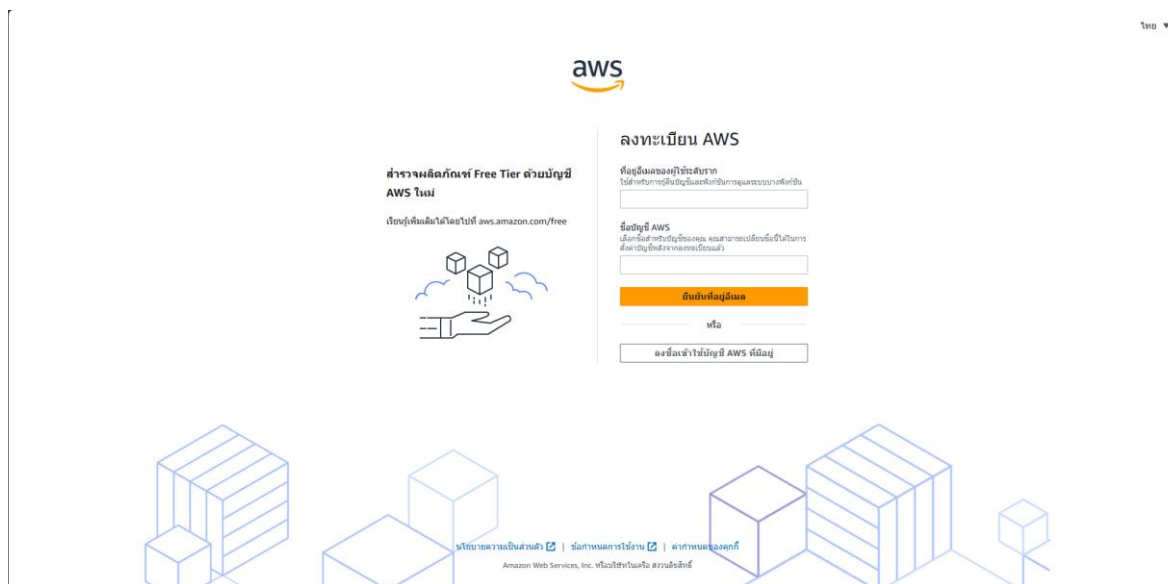
ขั้นตอนที่ 1: เตรียมไฟล์ เว็บแอปพลิเคชัน

สร้างไฟล์ requirements.txt file ประกอบด้วยแพ็คเกจที่จำเป็นสำหรับ Python ที่ใช้ในโปรเจกต์ ด้วยคำสั่ง

```
pip freeze -r requirements.txt
```

ขั้นตอนที่ 2: สมัครบัญชี AWS Account

สมัครบัญชี AWS Account AWS account ที่ aws.amazon.com และเข้าคอนโซล AWS Management Console เพื่อเริ่มต้นในการติดตั้ง



ภาพที่ ก.1 ภาพแสดงหน้าจอสัญสมัครบัญชี AWS

ขั้นตอนที่ 3: เปิดใช้งาน Amazon EC2 Instance

เปิดใช้งาน Amazon EC2 Instance โดยใช้ AWS Management Console หรือ รันโดยคำสั่ง:

```
aws ec2 run-instances --image-id ami-xxxxxxxxxxxxxxxx --instance-type t2.micro --key-name your-key-pair-name --security-group-ids your-security-group-id --subnet-id your-subnet-id
```

ขั้นตอนที่ 4: เชื่อมต่อกับ EC2 Instance

เชื่อมต่อโดยใช้ SSH, เชื่อมต่อกับ EC2 instance: ของคุณ

```
ssh -i /path/to/your-key.pem ec2-user@your-ec2-public-ip
```

ขั้นตอนที่ 5: ติดตั้ง Required Dependencies

ติดตั้ง Python, pip, and virtualenv ลงบน EC2 instance ของคุณ โดยใช้คำสั่ง

```
sudo yum update -y
```

```
sudo yum install python3 python3-pip -y
```

```
pip install virtualenv
```

ขั้นตอนที่ 6: ติดตั้ง Virtual Environment

สร้างโฟลเดอร์ด้วยคำสั่ง `mkdir my-django-project`

ไปที่โฟลเดอร์ด้วยคำสั่ง `cd my-django-project`

สร้าง Virtual Environment ด้วยคำสั่ง `virtualenv venv`

เปิดใช้งาน Virtual Environment ด้วยคำสั่ง `source venv/bin/activate`

ขั้นตอนที่ 7: อัปโหลดไฟล์เว็บไซต์

อัปโหลดไฟล์เว็บไซต์ไปที่ EC2 instance โดยใช้ SCP, Github หรือ วิธีอื่น

ขั้นตอนที่ 8: ติดตั้ง Required Packages

ติดตั้ง Required Packages เจาะจงใน requirements.txt file

```
pip install -r requirements.txt
```

ขั้นตอนที่ 9: Configure Database

ติดตั้ง Amazon RDS instance เพื่อ host ฐานข้อมูลโปรเจกต์ของคุณ, สร้างฐานข้อมูลและอัปเดต settings.py เพื่อเชื่อมต่อกับ RDS instance

ขั้นตอนที่ 10: Set Up Amazon S3

สร้าง Amazon S3 bucket เพื่อเก็บไฟล์ media (e.g., images, videos). อัปเดต settings.py เพื่อใช้งาน S3 bucket เก็บข้อมูล media storage.

ขั้นตอนที่ 11: รัน Migrations และ Collect Static Files

รัน Django migrations เพื่อติดตั้ง database และ collect static files:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

```
python manage.py collectstatic
```

ขั้นตอนที่ 12: ติดตั้งและกำหนดค่า Web Server โดยใช้ Nginx

ติดตั้งและกำหนดค่า Nginx as a reverse proxy server:

```
sudo amazon-linux-extras install nginx1.12 -y
```

```
sudo systemctl start nginx
```

```
sudo systemctl enable nginx
```

ขั้นตอนที่ 13: Deploy Your Django Project ติดตั้ง Gunicorn

รันแอปพลิเคชันโดยใช้ Gunicorn:

```
pip install gunicorn
```

```
gunicorn myproject.wsgi:application
```

ขั้นตอนที่ 14: ติดตั้ง SSL certificate

เปิดใช้งาน HTTPS โดยการตั้งค่า SSL certificate

ขั้นตอนที่ 15: เพิ่ม Custom Domain Name

เชื่อมต่อชื่อโดเมนแบบกำหนดเองกับโปรเจกต์ โดยใช้ port 53 หรือการตั้งค่าของผู้รับจดทะเบียนโดเมน