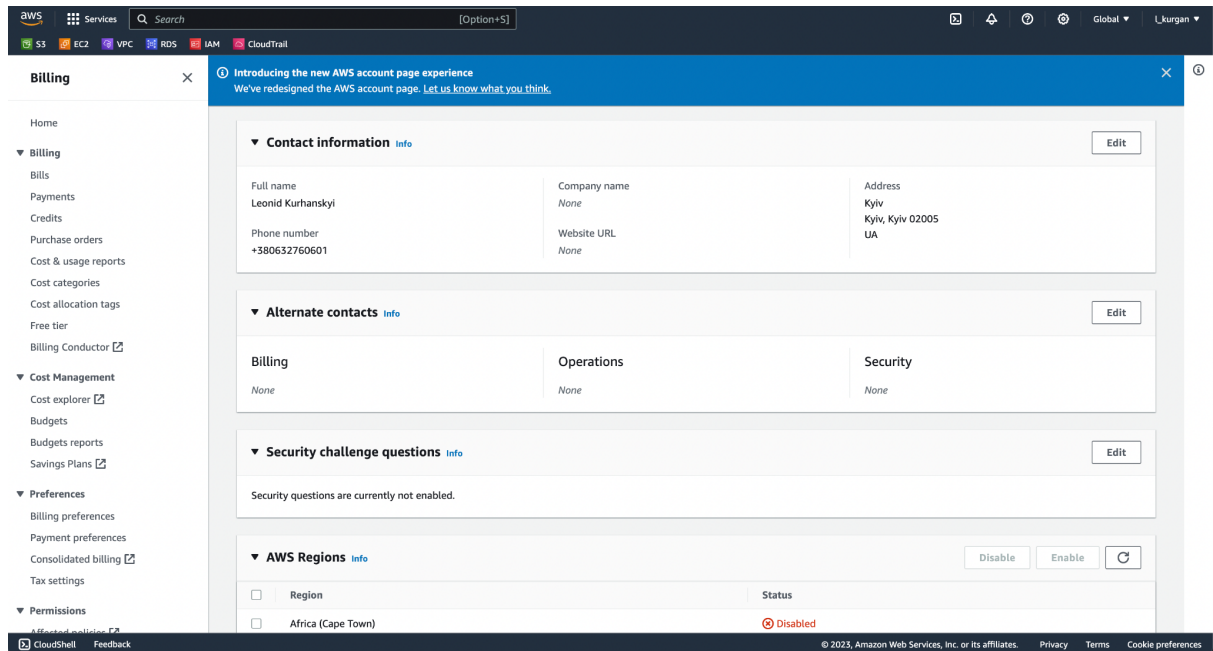
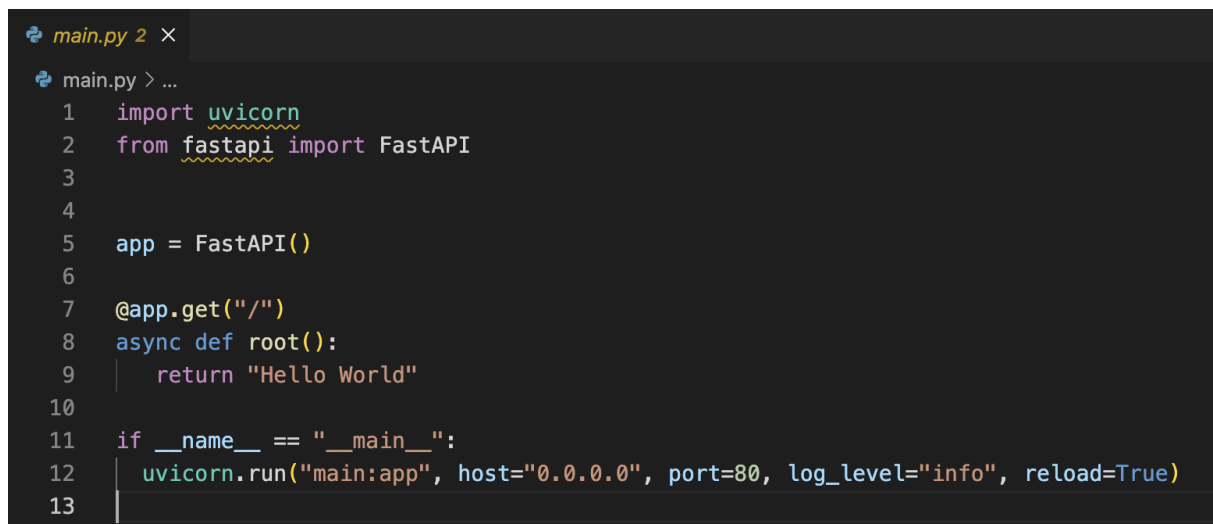


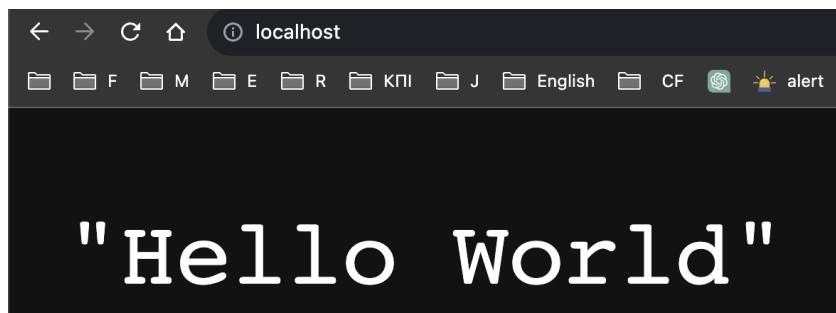
Реєструємося і входимо в AWS:



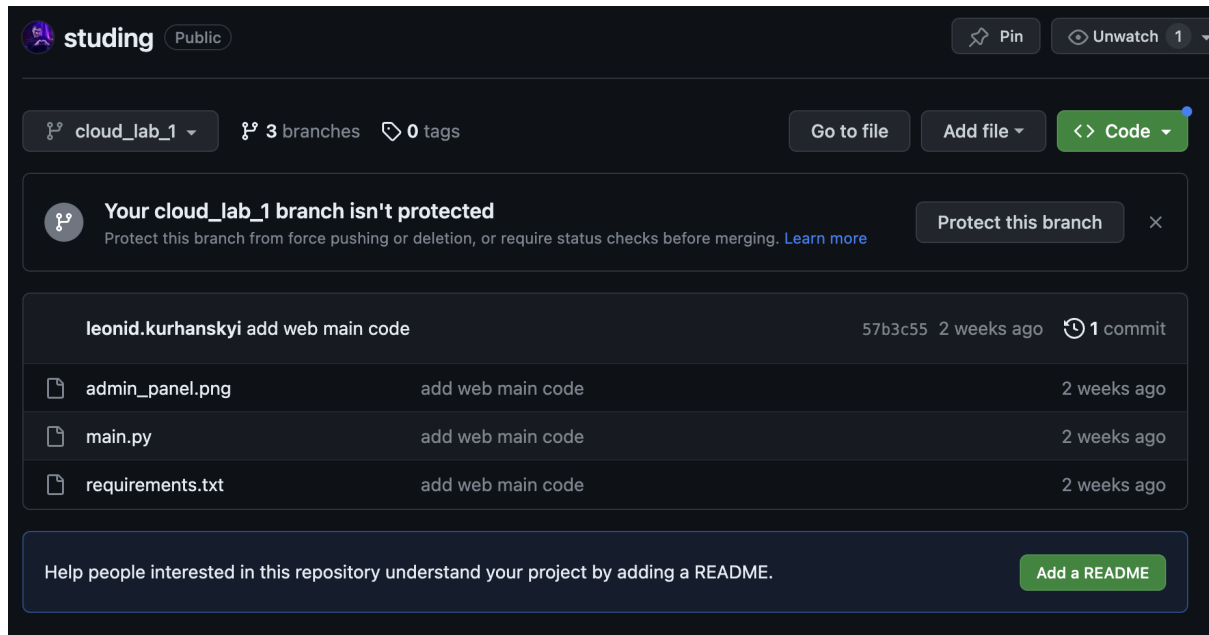
Підготуємо тестовий код веб-додатку:



Перевіримо локальну роботу:



Загрузимо код та залежності на [git](#):



Тепер перейдемо до AWS і зайдемо до сервісу EC2, запустимо інстанс та обираємо налаштування безкоштовного використання:

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-05b5a865c3579bbc4 (64-bit (x86)) / ami-0d031bec3dde37593 (64-bit (Arm))

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

Instance type

t2.micro

Free tier eligible

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

On-Demand RHEL base pricing: 0.0732 USD per Hour

On-Demand SUSE base pricing: 0.0132 USD per Hour

On-Demand Linux base pricing: 0.0132 USD per Hour

On-Demand Windows base pricing: 0.0178 USD per Hour

Обираємо ключ для з'єднання:

▼ 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*

lab_1

[Create new key pair](#)

Дозволимо доступ через протоколи http і https:

We'll create a new security group called '**launch-wizard-2**' 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

У полі advanced обираємо “User data” і передаємо туди скрипт який буде виконаний після старту інстансу.

User data - *optional* [Info](#)

Upload a file with your user data or enter it in the field.

 **Choose file**

```
#!/bin/bash

sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install python3 python3-pip git --yes --force-yes

sudo git clone https://github.com/lkurgan55/studing.git
cd studing
sudo git checkout cloud_lab_1

sudo python3 -m pip install -r ./requirements.txt
sudo crontab -l | { cat; sudo echo "@reboot python3 /studing/main.py"; } | crontab
-
sudo python3 /studing/main.py &
```

Чекаємо на запуск серверу:

Instance summary for i-032f2240443af7b28 (cloud_lab_1) Info		
Updated less than a minute ago		
Instance ID i-032f2240443af7b28 (cloud_lab_1)	Public IPv4 address 35.180.189.75 open address	Private IPv4 addresses 172.31.47.63
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-35-180-189-75.eu-west-3.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-47-63.eu-west-3.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-47-63.eu-west-3.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 35.180.189.75 [Public IP]	VPC ID vpc-07628878135f8e89 open	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-09d82e93b285e80ff open	
IMDSv2 Optional		

Підключаємось до серверу через веб-термінал і перевіримо виконання стартового скрипту:

git склонував проект:

```
ubuntu@ip-172-31-47-63:/studing$ pwd
/studing
ubuntu@ip-172-31-47-63:/studing$ ls
__pycache__  admin_panel.png  lab_1.pdf  main.py  requirements.txt  result.jpg  start_script.sh
ubuntu@ip-172-31-47-63:/studing$
```

crontab додав задачу на запуск веб-додатку при старті:

GNU nano 6.2

```
@reboot python3 /studing/main.py
```

Перевіряємо підключення через http:

