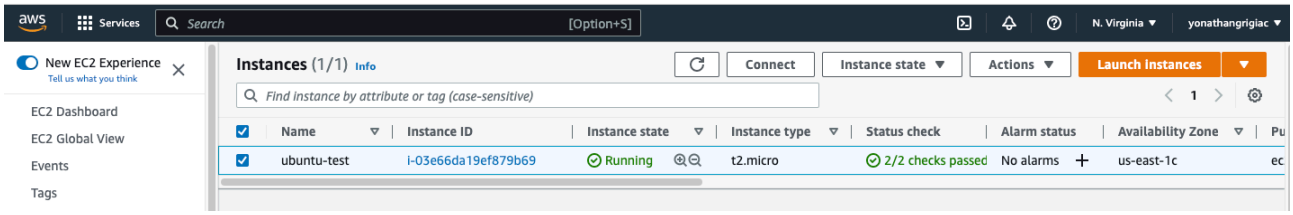
1. Create a t2.micro ec2 made from an Ubuntu AMI.
2. Setup security group.
3. Install docker and docker-compose
4. Deploy this docker-compose.yml

```
version: '3.3'
services:
db:
image: mysql:5.7
volumes:
- db_data:/var/lib/mysql
restart: always
environment:
MYSQL_ROOT_PASSWORD: somewordpress
MYSQL_DATABASE: wordpress
MYSQL_USER: wordpress
MYSQL_PASSWORD: wordpress
wordpress:
depends_on:
- db
image: wordpress:latest
ports:
- "8000:80"
restart: always
environment:
WORDPRESS_DB_HOST: db:3306
WORDPRESS_DB_USER: wordpress
WORDPRESS_DB_PASSWORD: wordpress
WORDPRESS_DB_NAME: wordpress
volumes:
db_data: {}
```




```
ubuntu@ip-172-31-19-148:~/assignment-12$ sudo docker-compose up -d
Starting assignment-12_db_1 ... done
Starting assignment-12_wordpress_1 ... done
```

```
ubuntu@ip-172-31-19-148:~/assignment-12$ sudo docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                   NAMES
6bccb4596d6e   wordpress:latest   "docker-entrypoint.s…"   32 minutes ago   Up 26 minutes   0.0.0.0:8000->80/tcp, :::8000->80/tcp   assignment-12_wordpress_1
2a848991168d   mysql:5.7          "docker-entrypoint.s…"   32 minutes ago   Up 26 minutes   3306/tcp, 33060/tcp                     assignment-12_db_1
ubuntu@ip-172-31-19-148:~/assignment-12$
```

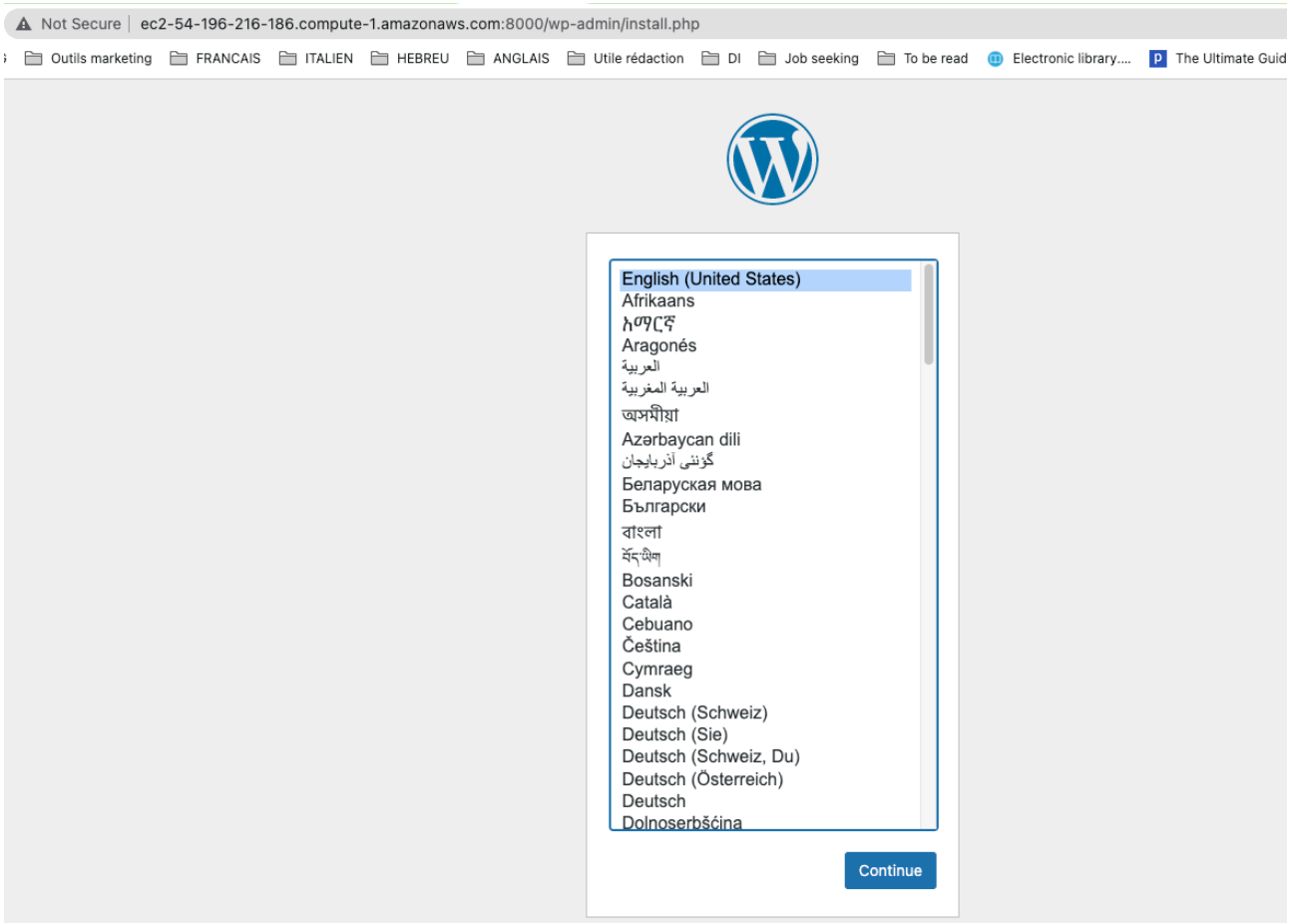
History (for reference):

```
    1  sudo apt-get remove docker docker-engine docker.io containerd runc
    2  sudo apt-get update
    3  sudo apt-get install    ca-certificates    curl    gnupg    lsb-release
    4  sudo mkdir -p /etc/apt/keyrings
    5  curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/
keyrings/docker.gpg
    6  echo   "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
    7    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
    8  sudo apt-get update
```

```
 9  sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
10  sudo docker run hello-world
11  mkdir assignment-12
12  cd assignment-12/
13  vim docker-compose.yml
14  ls
15  docker-compose up
16  sudo apt  install docker-compose
17  docker-compose up
18  ls
19  vim docker-compose.yml
20  docker-compose up
21  vim docker-compose.yml
22  docker-compose up
23  docker images
24  sudo docker-compose up
25  sudo docker-compose up -d
26  docker run -d assignment-12
27  sudo docker run -d assignment-12
28  cd ..
29  sudo docker run -d assignment-12
30  cd assignment-12/
31  docker ps
32  sudo docker ps
33  history
34  docker ps
35  sudo docker ps
36  telnet localhost 80
37  apt-get install telnetd
38  sudo apt-get install telnetd
39  telnet localhost 80
40  docker ps
41  sudo docker ps
42  cd assignment-12/
43  ls
44  vim docker-compose.yml
45  sudo docker logs 69b
46  sudo docker logs
47  sudo docker logs 6bccb4596d6e
48  sudo docker
49  sudo docker exec
50  sudo docker exec 6bccb4596d6e sh
51  sudo docker exec -it 6bccb4596d6e sh
52  telnet localhost 8000
53  history
```

URL: http://ec2-54-196-216-186.compute-1.amazonaws.com:8000/wp-admin/install.php (in the meantime I terminated my instance)

Screenshot of wordpress:




Challenges
1. Setup the EC2 resources with AWS CLI.

History (for reference):
aws ec2 describe-key-pairs
aws ec2 describe-security-groups
aws ec2 describe-vpcs
aws ec2 describe-subnets
aws ec2 describe-instances
aws ec2 run-instances --image-id ami-08c40ec9ead489470 --count 1 --instance-type t2.micro --key-name my-key-pair --security-group-ids sg-0d728afaba4d9057f --subnet-id subnet-0e32d83cb25b78005