

## Labs 6-9

**Student ID:** 23796349

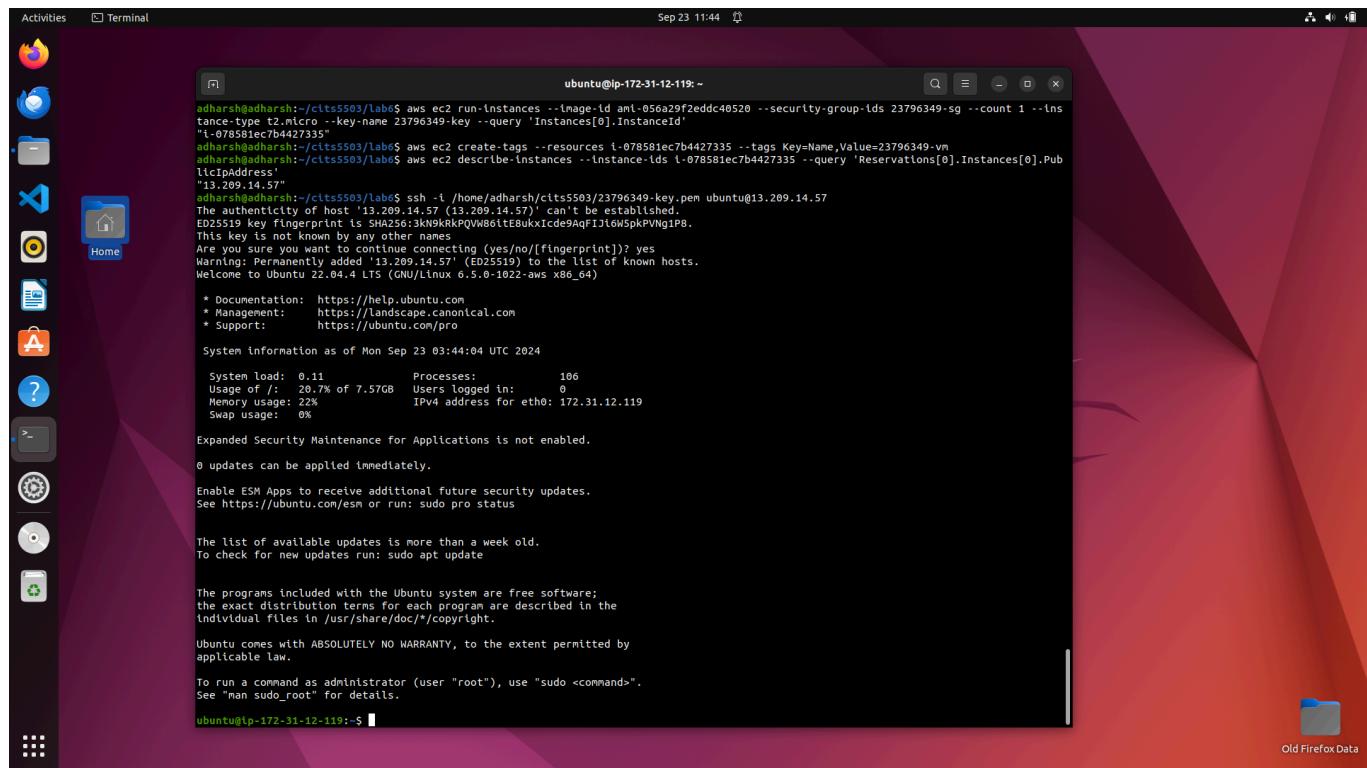
**Student Name:** Adharsh Sundaram Soudakar

# Lab 6

## Set up an EC2 instance

### [1] Create an EC2 micro instance with Ubuntu and SSH into it

- I used AWS CLI to create the EC2 instance required to complete this lab.
- The steps followed is the same steps that were done in [Lab 2](#).
- The screenshots below show the commands and steps followed to create the EC2 instance and connect to it using [SSH](#):



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the date and time are "Sep 23 11:44". The terminal content shows the following commands and output:

```
ubuntu@ip-172-31-12-119:~$ adhars@adhars:~/cttss503/lab6$ aws ec2 run-instances --image-id ami-05a29f2eddc40520 --security-group-ids 23796349-sg --count 1 --instance-type t2.micro --key-name 23796349-key --query 'Instances[0].InstanceId'
"i-078581ec7b442735"
adhars@adhars:~/cttss503/lab6$ aws ec2 create-tags --resources i-078581ec7b442735 --tags Key=Name,Value=23796349-vm
adhars@adhars:~/cttss503/lab6$ aws ec2 describe-instances --instance-ids i-078581ec7b442735 --query 'Reservations[0].Instances[0].PublicIpAddress'
13.209.14.57"
adhars@adhars:~/cttss503/lab6$ ssh -i /home/adhars/citc5502/23796349-key.pem ubuntu@13.209.14.57
The authenticity of host '13.209.14.57' (13.209.14.57) can't be established.
ED25519 key fingerprint is SHA256:3KN9kRkPQW86litEBuKxIcd9qPIj6W5pkPVNgIP8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.209.14.57' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

System information as of Mon Sep 23 03:44:04 UTC 2024

System load: 0.11      Processes:          106
Usage of /: 20.7% of 7.57GB   Users logged in:    0
Memory usage: 22%        IPv4 address for eth0: 172.31.12.119
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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-12-119:~$
```

### [2] Install the Python 3 virtual environment package

- Before installing Python 3 virtual environment, the instance has to be updated and upgraded to the latest version.
- The following commands were used:

```
sudo apt-get update
sudo apt-get upgrade
```

- The screenshot below shows the output:

- Then the following command can be used to install Python 3 virtual environment:

```
sudo apt-get install python3-venv
```

- The screenshot below shows the output:

```
Activities Terminal Sep 23 11:52 ubuntu@ip-172-31-12-119:~  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  python3-pip-whl python3-setuptools-whl python3.10-venv  
The following NEW packages will be installed:  
  python3-pip-whl python3-setuptools-whl python3-venv python3.10-venv  
0 upgraded, 4 newly installed, 0 to remove and 3 not upgraded.  
Need to get 2475 kB of archives.  
After this operation, 2891 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-pip-whl all 22.0.2+dfsg-1ubuntu0.4 [1680 kB]  
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-setuptools-whl all 59.6.0-1.2ubuntu0.22.04.2 [788 kB]  
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3.10-venv amd64 3.10.12-1-22.04.6 [5722 B]  
Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-venv amd64 3.10.6-1-22.04.1 [1042 B]  
Fetched 2475 kB in 0s (4678 kB/s)  
Selecting previously unselected package python3-pip-whl.  
Preparing database ... (65% File and directory currently installed.)  
Preparing to unpack .../python3-pip-whl_22.0.2+dfsg-1ubuntu0.4_all.deb ...  
Unpacking python3-pip-whl (22.0.2+dfsg-1ubuntu0.4) ...  
Selecting previously unselected package python3-setuptools-whl.  
Preparing to unpack .../python3-setuptools-whl_59.6.0-1.2ubuntu0.22.04.2_all.deb ...  
Unpacking python3-setuptools-whl (59.6.0-1.2ubuntu0.22.04.2) ...  
Selecting previously unselected package python3.10-venv.  
Preparing to unpack .../python3.10-venv_3.10.12-1-22.04.6_amd64.deb ...  
Unpacking python3.10-venv (3.10.12-1-22.04.6) ...  
Selecting previously unselected package python3-venv.  
Preparing to unpack .../python3-venv_3.10.6-1-22.04.1_amd64.deb ...  
Unpacking python3-venv (3.10.6-1-22.04.1) ...  
Setting up python3-setuptools-whl (59.6.0-1.2ubuntu0.22.04.2) ...  
Setting up python3-pip-whl (22.0.2+dfsg-1ubuntu0.4) ...  
Setting up python3.10-venv (3.10.12-1-22.04.6) ...  
Setting up python3-venv (3.10.6-1-22.04.1) ...  
Scanning processes...  
Scanning candidates...  
Scanning linux images...  
Running kernel seems to be up-to-date.  
Restarting services...  
Service restarts being deferred:  
/etc/needrestart/restart.d/dbus.service  
systemctl restart networkd-dispatcher.service  
systemctl restart packagekit.service  
systemctl restart polkit.service  
systemctl restart ssh.service  
systemctl restart systemd-journald.service  
/etc/needrestart/restart.d/systemd-manager  
systemctl restart systemd-networkd.service  
systemctl restart systemd-resolved.service  
systemctl restart systemd-udevd.service  
systemctl restart unattended-upgrades.service  
systemctl restart user@1000.service
```

- Since we are going to run commands that require **su** access, it will be easier to persist the command line to be in **su** state for a while.
  - This can be achieved with the following command:

```
sudo bash
```

### [3] Access a directory

- To create the required directories and traverse to that location, the following commands were used:

```
mkdir -p opt/wwc/mysites  
cd opt/wwc/mysites
```

- The screenshot below shows the output:

```
root@ip-172-31-12-119:~# tree opt/  
opt/  
└── wwc  
    └── mysites  
  
2 directories, 0 files  
root@ip-172-31-12-119:~#
```

### [4] Set up a virtual environment

- To set up a virtual environment, the following command can be used:

*NOTE:* Here we create a virtual environment called **myvenv**

```
python3 -m venv myvenv
```

- The screenshot below shows the output:

```
root@ip-172-31-12-119:~/opt/wwc/mysites# tree opt/  
opt/  
└── wwc  
    └── mysites  
  
2 directories, 0 files  
root@ip-172-31-12-119:~/opt/wwc/mysites# cd opt/wwc/mysites/  
root@ip-172-31-12-119:~/opt/wwc/mysites# python3 -m venv myvenv  
root@ip-172-31-12-119:~/opt/wwc/mysites# ls  
myvenv  
root@ip-172-31-12-119:~/opt/wwc/mysites#
```

### [5] Activate the virtual environment

- The virtual environment (myvenv) can be activated using the following command:

```
source myvenv/bin/activate
```

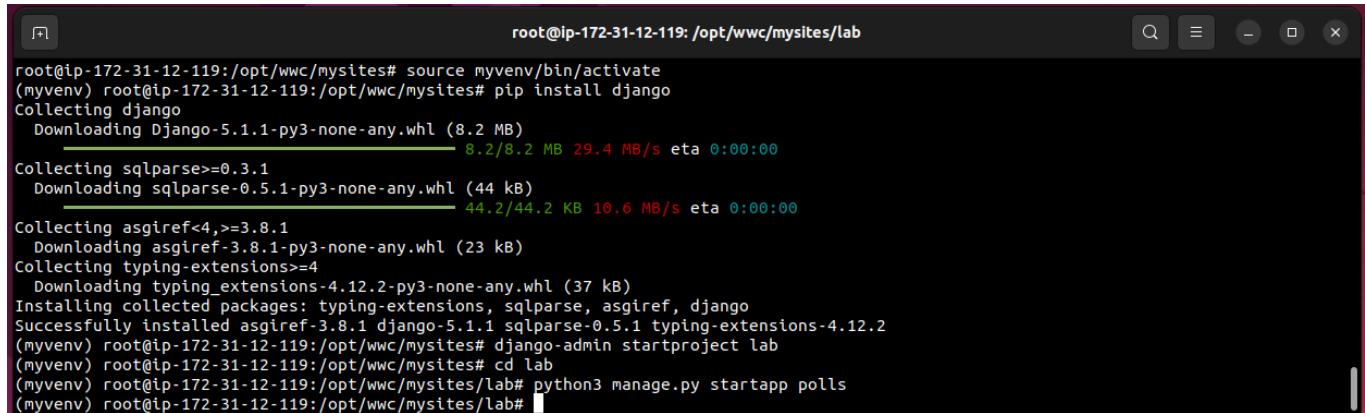
- Now that we are inside the virtual environment, we can install **django** and create a project called **lab** using the following commands:

```
pip install django
django-admin startproject lab
```

- Then we traverse into the project (lab) directory and create an app called **polls** using the following commands:

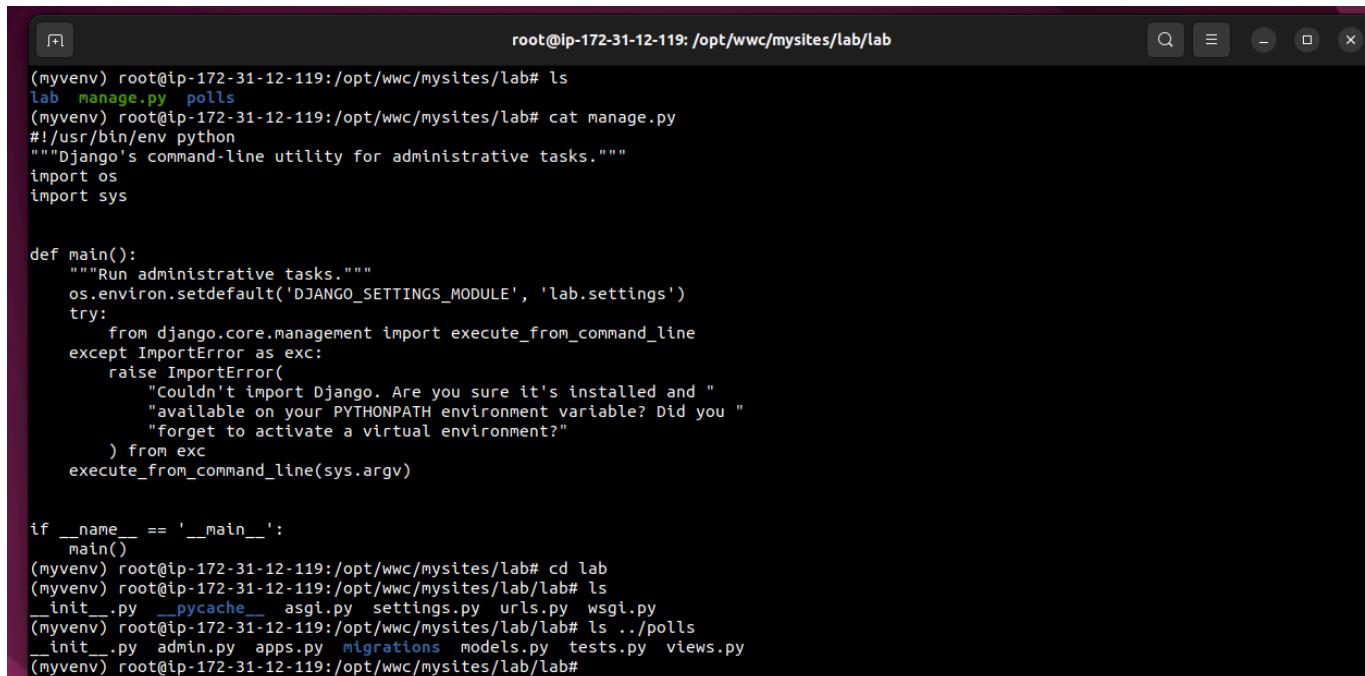
```
cd lab
python3 manage.py startapp polls
```

- The screenshot below shows the output of the above steps:



```
root@ip-172-31-12-119:/opt/wwc/mysites# source myvenv/bin/activate
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites# pip install django
Collecting django
  Downloading Django-5.1.1-py3-none-any.whl (8.2 MB)
    8.2/8.2 MB 29.4 MB/s eta 0:00:00
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.5.1-py3-none-any.whl (44 kB)
    44.2/44.2 KB 10.6 MB/s eta 0:00:00
Collecting asgiref<4,>=3.8.1
  Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Collecting typing-extensions>=4
  Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: typing-extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.1 sqlparse-0.5.1 typing-extensions-4.12.2
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites# django-admin startproject lab
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites# cd lab
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab# python3 manage.py startapp polls
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab#
```

- The screenshot below shows the files and content created after executing the **startproject** and **startapp** commands:



```
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab# ls
lab  manage.py  polls
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab# cat manage.py
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'lab.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab# cd lab
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab/lab# ls
__init__.py  __pycache__  asgi.py  settings.py  urls.py  wsgi.py
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab/lab# ls ..//polls
__init__.py  admin.py  apps.py  migrations  models.py  tests.py  views.py
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab/lab#
```

## [6] Install nginx

- To host our web app, we need a web server. Here, we plan to use **nginx**. It can be installed using the following command:

```
apt install nginx
```

- The screenshot below shows the installation output:

The terminal window shows the command `root@ip-172-31-12-119:/opt/www/mysites/lab# apt install nginx` being run. The output details the package dependencies and the download and installation process for various components like libfontconfig, libjpeg-turbo, libxml2, and libnginx-mod-stream.

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libnginx-mod-http-geolp2
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geolp2
  libtiff5 libwebp7 libxpm4 nginx nginx-common nginx-core
Suggested packages:
  libbd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libnginx-mod-http-geolp2
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geolp2
  libtiff5 libwebp7 libxpm4 nginx nginx-common nginx-core
0 upgraded, 20 newly installed, 0 to remove and 3 not updated.
Need to get 2693 kB of archives.
After this operation, 8358 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-core all 2.37-2build1 [1041 kB]
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fontconfig-config all 2.13.1-4.2ubuntu5 [29.1 kB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libdeflate0 amd64 1.10-2 [70.9 kB]
Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfontconfig1 amd64 2.13.1-4.2ubuntu5 [131 kB]
Get:5 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libjpeg-turbo8 amd64 2.1.2-0ubuntu1 [134 kB]
Get:6 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libjpeg8 amd64 8c-2ubuntu10 [2264 B]
Get:7 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libjbig0 amd64 2.1-3.1ubuntu0.22.04.1 [29.2 kB]
Get:8 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libwebp7 amd64 1.2.2-2ubuntu0.22.04.2 [206 kB]
Get:9 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libxpm4 amd64 1.1.1-5.12ubuntu0.22.04.2 [36.7 kB]
Get:10 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgd3 amd64 2.3.0-2ubuntu2 [129 kB]
Get:11 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libtiff5 amd64 4.1.0-13.5.12ubuntu0.22.04.2 [185 kB]
Get:12 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtiff5 amd64 4.1.0-13.5.12ubuntu0.22.04.2 [185 kB]
Get:13 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-http-geolp2 amd64 1.18.0-0ubuntu14.5 [40.1 kB]
Get:14 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-0ubuntu14.5 [15.5 kB]
Get:15 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-0ubuntu14.5 [13.8 kB]
Get:16 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-mail amd64 1.18.0-0ubuntu14.5 [45.8 kB]
Get:17 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-stream amd64 1.18.0-0ubuntu14.5 [72.8 kB]
Get:18 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-stream-geolp2 amd64 1.18.0-0ubuntu14.5 [10.1 kB]
Get:19 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nginx-core amd64 1.18.0-0ubuntu14.5 [483 kB]
Get:20 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nginx amd64 1.18.0-6ubuntu14.5 [3882 B]
Unpacking fonts-dejavu-core (2.37-2build1) ...
Preconfiguring packages...
Selecting previously unselected package fonts-dejavu-core.
(Reading database ... 65364 files and directories currently installed.)
Preparing to unpack .../00-fonts-dejavu-core_2.37-2build1_all.deb ...
Unpacking fonts-dejavu-core (2.37-2build1) ...
Selecting previously unselected package fontconfig-config.
Preparing to unpack .../01-fontconfig-config_2.13.1-4.2ubuntu5_all.deb ...
Unpacking fontconfig-config (2.13.1-4.2ubuntu5) ...
Selecting previously unselected package libdeflate0:amd64.
Preparing to unpack .../02-libdeflate0.1.10-2_amd64.deb ...
Unpacking libdeflate0:amd64 (1.10-2) ...
Selecting previously unselected package libfontconfig1:amd64.
Preparing to unpack .../03-libfontconfig1_2.13.1-4.2ubuntu5_amd64.deb ...
Unpacking libfontconfig1:amd64 (2.13.1-4.2ubuntu5) ...
Selecting previously unselected package libjpeg-turbo8:amd64.
Preparing to unpack .../04-libjpeg-turbo8_2.1.2-0ubuntu1_amd64.deb ...
Unpacking libjpeg-turbo8:amd64 (2.1.2-0ubuntu1) ...

```

## [7] Configure nginx

- We need to configure nginx, specifying the port to listen to and how requests should be handled (forward requests to localhost on port 8000).
- The following code accomplishes the same:

*NOTE: This code was placed in the configuration file `default` located in the directory `/etc/nginx/sites-enabled`*

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    location / {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_pass http://127.0.0.1:8000;
    }
}
```

- The screenshot below shows the changes made to the `default` file:

```
Activities Terminal Sep 23 12:09
root@ip-172-31-12-119: /
```

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    location / {
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_pass http://127.0.0.1:8000;
    }
}
```

## [8] Restart nginx

- We need to restart nginx to apply the changes made to the configuration file. This can be done using the following command:

```
service nginx restart
```

## [9] Access your EC2 instance

- To run our web application, use the following command:  
*NOTE: This will start the Django development server on port **8000**.*

```
python3 manage.py runserver 8000
```

- The screenshot below shows the output on the terminal when the above command was executed:

```
root@ip-172-31-12-119: /opt/wwc/mysites/lab
```

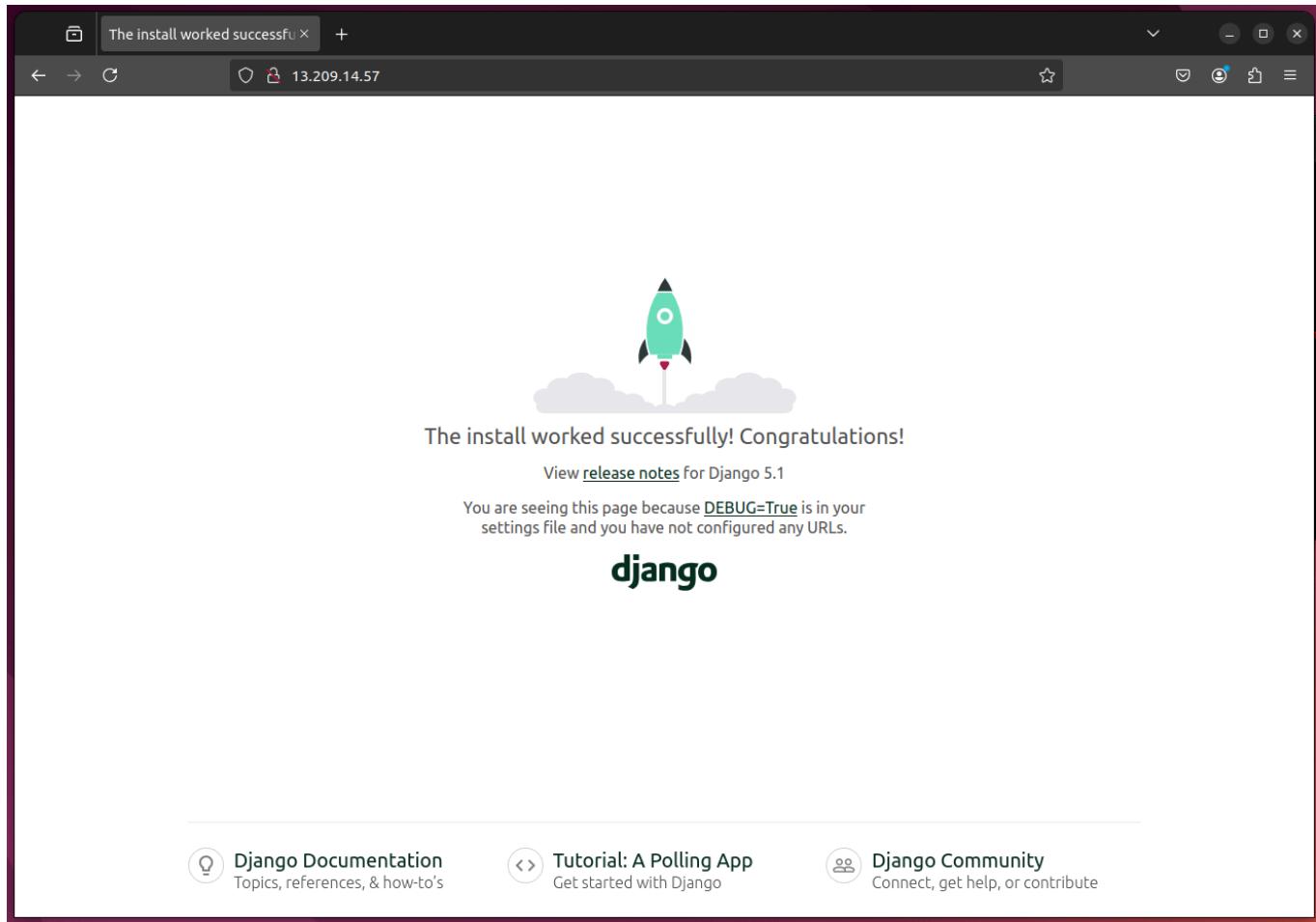
```
(myvenv) root@ip-172-31-12-119:/# service nginx restart
(myenv) root@ip-172-31-12-119:/# cd opt/wwc/mysites/lab/
(myenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab# python3 manage.py runserver 8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
September 23, 2024 - 04:10:51
Django version 5.1.1, using settings 'lab.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

[23/Sep/2024 04:11:47] "GET / HTTP/1.0" 200 12068
Not Found: /favicon.ico
[23/Sep/2024 04:11:47] "GET /favicon.ico HTTP/1.0" 404 2205
```

- To verify if this works correctly, we can enter the IP address of the created EC2 instance in a browser.
- The following screenshot shows the page that was displayed:



## Set up Django inside the created EC2 instance

### [1] Edit the following files (create them if non-existent)

- **Vim** text editor was used to edit all the three files.
- The file `views.py` in the directory `polls` was replaced with the following code:

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, world.")
```

- The file `urls.py` was created in the directory `polls` and the following code was added:

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, world.")
```

- The file `urls.py` in the directory `lab` was replaced with the following code:

```
from django.urls import include, path
from django.contrib import admin

urlpatterns = [
    path('polls/', include('polls.urls')),
    path('admin/', admin.site.urls),
]
```

## [2] Run the web server again

- The same command used in [Access your EC2 instance](#) was used again to start the Django web development server.
- The screenshot below shows the output:

The screenshot shows a terminal window with the following text:

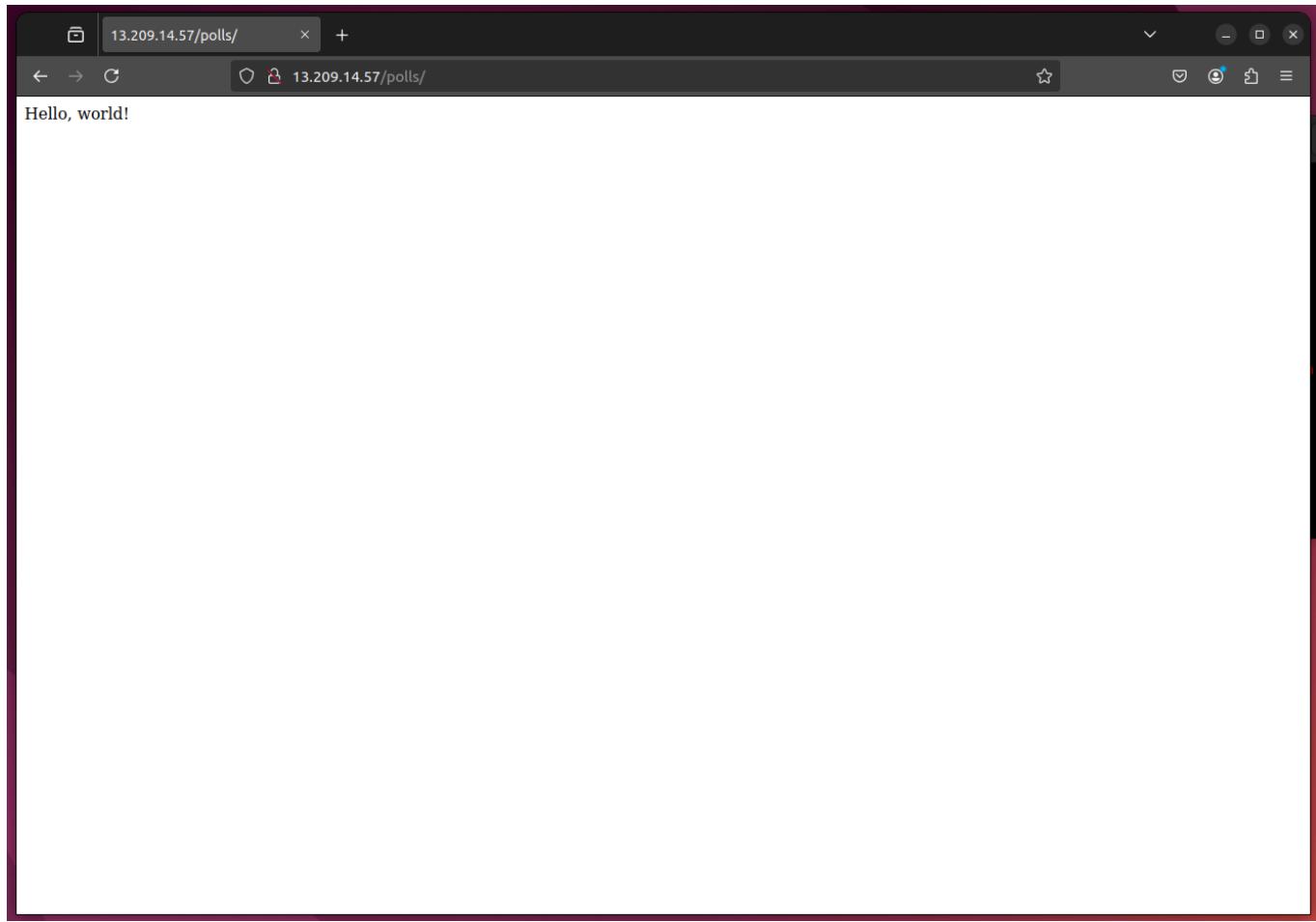
```
root@ip-172-31-12-119:/opt/wwc/mysites/lab
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab/polls# vim urls.py
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab/polls# cd ..
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab# lab
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab/lab# vim urls.py
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab/lab# cd ..
(myvenv) root@ip-172-31-12-119:/opt/wwc/mysites/lab# python3 manage.py runserver 8000
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
September 23, 2024 - 04:18:09
Django version 5.1.1, using settings 'lab.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

[23/Sep/2024 04:18:40] "GET /polls/ HTTP/1.0" 200 13
```

## [3] Access the EC2 instance

- Use the URL:[http://<ip\\_add\\_of\\_your\\_EC2>/polls/](http://<ip_add_of_your_EC2>/polls/) to see the HTTP Response in your browser screen.
- In my case the URL is <http://13.209.14.57/polls/>.
- The screenshot below shows the output that I got:



## Set up an ALB

### [1] Create an application load balancer

To create the ALB, the code in **Lab 5** was reused:

*NOTE:* Refer to comments in the code to understand the implementation.

```
import boto3

# Constants
#The created instance's ID
inst_ID = 'i-078581ec7b4427335'
#The security group created in previous lab (lab5), has both ssh and http
permissions
sec_grp_id = 'sg-0ba69d87ee7933db6'
#The subnets 'a' and 'c' IDs (since my instance is a t2.micro machine)
SUBNETS = [ 'subnet-09f996d1ad81767a9', 'subnet-0b3601189181ee48e' ]
#The default VPC ID
VPC_ID = 'vpc-01f842220d0070f97'

#Initiating client for alb
client = boto3.client('elbv2')

#Application Load Balancer
response = client.create_load_balancer(
    Name='23796349-ALB',
```

```

Subnets=SUBNETS[:2],
SecurityGroups=[sec_grp_id],
Scheme='internet-facing',
Tags=[
    {
        'Key': 'Name',
        'Value': '23796349-ALB'
    },
]
)
lb_arn = response['LoadBalancers'][0]['LoadBalancerArn']
print(f"Load Balancer ARN: {lb_arn}")

#Target group
response = client.create_target_group(
    Name='23796349-TG',
    Protocol='HTTP',
    Port=80,
    HealthCheckPath='/polls/', #Specifying polls as the path for health check
    VpcId=VPC_ID,
    Tags=[
        {
            'Key': 'Name',
            'Value': '23796349-TG'
        },
    ]
)
tg_arn = response['TargetGroups'][0]['TargetGroupArn']
print(f"Target Group ARN: {tg_arn}")

#Register the created instance as a target in the target group
client.register_targets(
    TargetGroupArn=tg_arn,
    Targets=[
        {'Id': inst_ID},
    ]
)

#Listener with HTTP and Port 80 forwarding
client.create_listener(
    LoadBalancerArn=lb_arn,
    Protocol='HTTP',
    Port=80,
    DefaultActions=[
        {
            'Type': 'forward',
            'TargetGroupArn': tg_arn
        },
    ]
)

print("Application Load Balancer and listener successfully created. The instance was registered as a target")

```

- The screenshots below show the ALB that was created:

```
adharsh@adharsh:~/cits5503/lab6$ python3 createAlbForLab6.py
Load Balancer ARN: arn:aws:elasticloadbalancing:ap-northeast-2:489389878001:loadbalancer/app/23796349-ALB/6142640949336830
Target Group ARN: arn:aws:elasticloadbalancing:ap-northeast-2:489389878001:targetgroup/23796349-TG/52113031a432dc23
Application Load Balancer and listener successfully created. The instance was registered as a target
adharsh@adharsh:~/cits5503/lab6$
```

The screenshot shows the AWS CloudWatch terminal output. It starts with the command 'python3 createAlbForLab6.py'. The response indicates that the load balancer and target group were successfully created. The load balancer ARN is 'arn:aws:elasticloadbalancing:ap-northeast-2:489389878001:loadbalancer/app/23796349-ALB/6142640949336830' and the target group ARN is 'arn:aws:elasticloadbalancing:ap-northeast-2:489389878001:targetgroup/23796349-TG/52113031a432dc23'. A message at the end states 'Application Load Balancer and listener successfully created. The instance was registered as a target'.

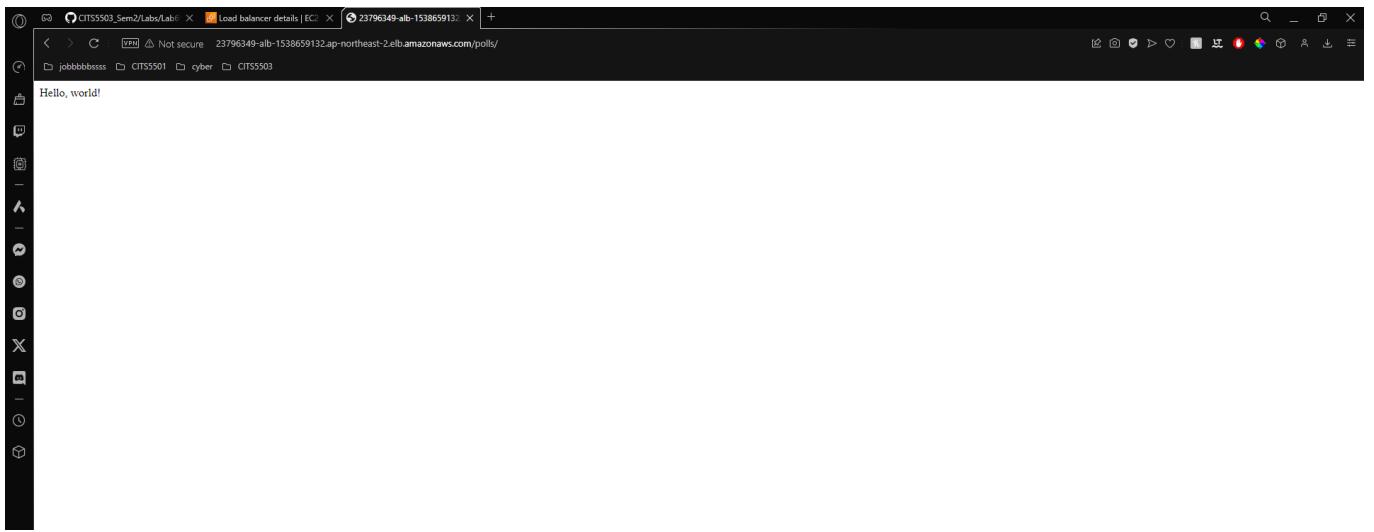
## [2] Health check

- By default health checks are fetched every 30 seconds.
- The screenshot below shows the Target group that was created and its health check settings:

The screenshot shows the AWS CloudWatch terminal output. It starts with the command 'python3 createAlbForLab6.py'. The response indicates that the target group was successfully created. The target group ARN is 'arn:aws:elasticloadbalancing:ap-northeast-2:489389878001:targetgroup/23796349-TG/52113031a432dc23'.

## [3] Access

- To see if the ALB is working as intended, access the URL: [http://<alb\\_dns\\_name>/polls/](http://<alb_dns_name>/polls/). In my case, it is <http://23796349-alb-1538659132.ap-northeast-w.elb.amazonaws.com/polls/>.
- The screenshot below shows the output that I got:



**All resources that were created for the lab (*EC2 instance, ALB and TG*) were deleted using the AWS Console after the lab completion.**

---

# Lab 7

## Create an EC2 instance

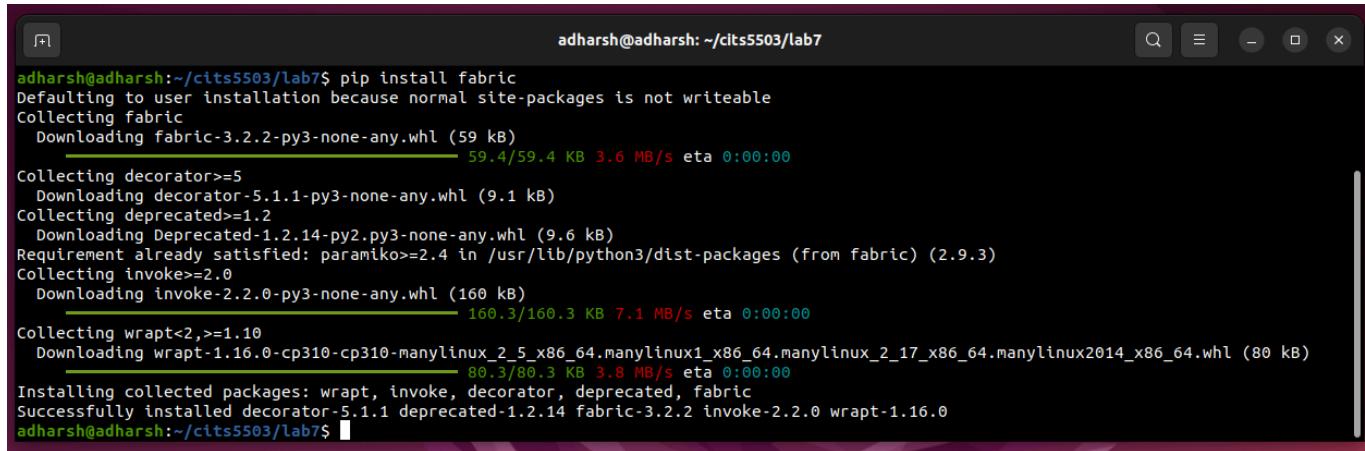
- The EC2 instance required for this lab was created using the same methods used in [Lab 6](#) ([Refer here](#)).

## Install and configure Fabric

- The following command can be used to install **fabric**:

```
pip install fabric
```

- The screenshot below shows the output of the command:



adharsh@adharsh:~/cits5503/lab7\$ pip install fabric  
Defaulting to user installation because normal site-packages is not writeable  
Collecting fabric  
 Downloading fabric-3.2.2-py3-none-any.whl (59 kB)  
 59.4/59.4 KB 3.6 MB/s eta 0:00:00  
Collecting decorator>=5  
 Downloading decorator-5.1.1-py3-none-any.whl (9.1 kB)  
Collecting deprecated>=1.2  
 Downloading Deprecated-1.2.14-py2.py3-none-any.whl (9.6 kB)  
Requirement already satisfied: paramiko>=2.4 in /usr/lib/python3/dist-packages (from fabric) (2.9.3)  
Collecting invoke>=2.0  
 Downloading invoke-2.2.0-py3-none-any.whl (160 kB)  
 160.3/160.3 KB 7.1 MB/s eta 0:00:00  
Collecting wrapt<2,>=1.10  
 Downloading wrapt-1.16.0-cp310-cp310-manylinux\_2\_5\_x86\_64.manylinux1\_x86\_64.manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (80 kB)  
 80.3/80.3 KB 3.8 MB/s eta 0:00:00  
Installing collected packages: wrapt, invoke, decorator, deprecated, fabric  
Successfully installed decorator-5.1.1 deprecated-1.2.14 fabric-3.2.2 invoke-2.2.0 wrapt-1.16.0  
adharsh@adharsh:~/cits5503/lab7\$

- Once **fabric** is installed, we need to configure it. We can do this by creating a **config** file in the **~/.ssh** folder.
- The following content was added to the config file:

```
Host <your EC2 instance name>  
  Hostname <your EC2 instance public IPv4 DNS>  
  User ubuntu  
  UserKnownHostsFile /dev/null  
  StrictHostKeyChecking no  
  PasswordAuthentication no  
  IdentityFile <path to your private key>
```

- The following screenshot shows the content that I added to my config file:

*NOTE: I copied my private key to the **~/.ssh** folder. Hence I haven't specified path for the same. Also I replaced the placeholders with the appropriate content.*

- To see if fabric has been configured the right way, connect to your instance using the following code:

```
python3
>>> from fabric import Connection
>>> c = Connection('<your EC2 instance name>')
>>> result = c.run('uname -s')
Linux
>>>
```

- The screenshot below shows my attempt to connect to my instance:

```
adharsh@adharsh: ~/ssh
adharsh@adharsh:~/ssh$ vim config
adharsh@adharsh:~/ssh$ python3
Python 3.10.12 (main, Sep 11 2024, 15:47:36) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from fabric import Connection
>>> c = Connection('23796349-vm')
>>> result = c.run('uname -s')
Linux
>>>
```

## Use Fabric for automation

- The entirety of Lab 6 steps in Set up an EC2 instance and Set up Django inside the created EC2 instance has been automated with the following code:

Code referenced from [Fabric documentation](#).

**NOTE:**

*Refer to comments in the code to understand the implementation.*

Note the usage of `run` and `sudo` functions, wherever `su` permissions were required `sudo` was used.

Appropriate number of \ were used to escape special characters.

*Full path to a file was always used.*

```
from fabric import Connection, task

def automate_setup_and_run():

    with Connection('23796349-vm') as c:

        print('-----')
        print('----Installing necessary things----+')
        print('-----')

        # Installing python3 virtual environment and nginx
        c.sudo("apt update")
        c.sudo("apt upgrade -y")
        c.sudo("apt install python3-pip -y")
        c.sudo("apt install python3-venv -y")
        c.sudo("apt install nginx -y")

        print('-----')
        print('----Setting up Virtual Environment----+')
        print('-----')

        # Setup virtual environment
        c.run("python3 -m venv myenv")
        c.run("source myenv/bin/activate")

        print('-----')
        print('----Installing django----+')
        print('-----')

        # Installing Django within the virtual environment
        c.sudo("pip install django")

        print('-----')
        print('----Setting up django----+')
        print('-----')

        # Setting up Django project
        c.run("django-admin startproject lab")

        print('-----')
        print('----Creating polls app----+')
        print('-----')

        # Changing directory and creating polls app
        c.run("cd lab && python3 manage.py startapp polls")

        print('-----')
        print('----Configuring nginx----+')
        print('-----')

        # Configure nginx
        c.run("echo 'server {\n      listen 80 default_server;\n      listen [::]:80\n      default_server;\n\n      location / {\n          proxy_set_header X-Forwarded-Host\n          proxy_pass http://127.0.0.1:8000;\n      }\n}' | sudo tee /etc/nginx/sites-available/lab >> /dev/null")
        c.run("sudo ln -s /etc/nginx/sites-available/lab /etc/nginx/sites-enabled/")
        c.run("sudo nginx -t")
        c.run("sudo systemctl restart nginx")
```

```

$host;\\n      proxy_set_header X-Real-IP $remote_addr;\\n      proxy_pass
http://127.0.0.1:8000;\\n    }\\n}' | sudo tee /etc/nginx/sites-enabled/default")
c.sudo("nginx -t")
c.sudo("service nginx restart")

print('-----')
print('----Editing polls/views.py----')
print('-----')

# Edit /polls/views.py
c.run("echo 'from django.http import HttpResponse\\n\\ndef
index(request):\\n    return HttpResponse(\"Hello, world.\")' | sudo tee
/home/ubuntu/lab/polls/views.py")

print('-----')
print('----Editing polls/urls.py----')
print('-----')

# Edit /polls/urls.py
c.run("echo 'from django.urls import path\\nfrom . import
views\\n\\nurlpatterns = [\\n    path(\"\", views.index, name=\"index\"),\\n]' | sudo
tee /home/ubuntu/lab/polls/urls.py")

print('-----')
print('----Editing lab/urls.py----')
print('-----')

# Edit /lab/urls.py
c.run("echo 'from django.urls import include, path\\nfrom django.contrib
import admin\\n\\nurlpatterns = [\\n    path(\"polls/\", include(\"polls.urls\")),\\n
path(\"admin/\", admin.site.urls),\\n]' | sudo tee /home/ubuntu/lab/lab/urls.py")

print('-----')
print('----Starting Development Server----')
print('-----')

# Starting development server
c.run("cd lab && python3 manage.py runserver 0.0.0.0:8000 &")

if __name__ == "__main__":
    automate_setup_and_run()

```

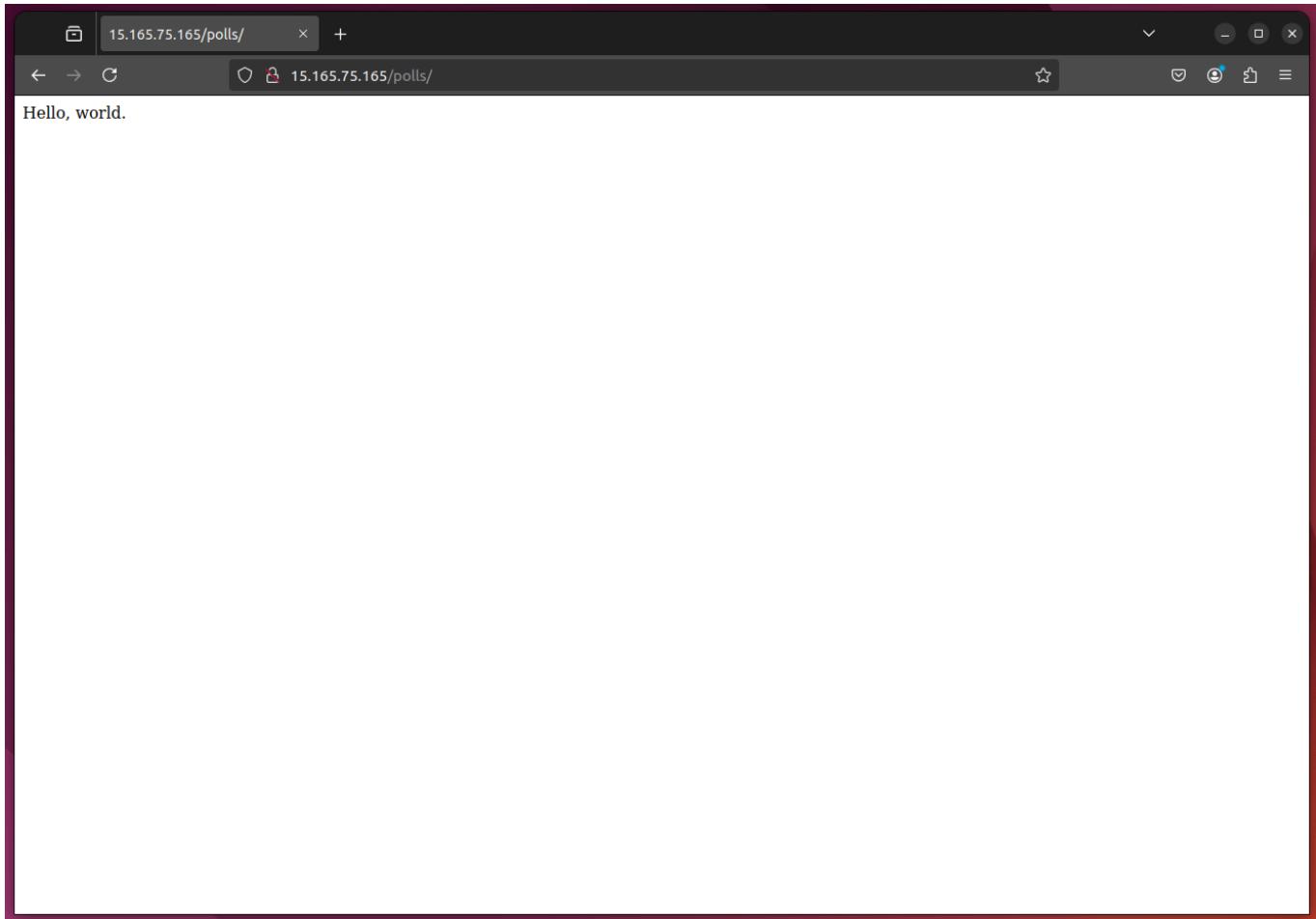
- The screenshot below shows the end result (*web development server on port 8000*) of the code execution:

```
adharsh@adharsh: ~/ssh
-----+-----+
+----Editing polls/urls.py-----+
from django.urls import path
from . import views

urlpatterns = [
    path("", views.index, name="index"),
]
-----+-----+
+----Editing lab/urls.py-----+
from django.urls import include, path
from django.contrib import admin

urlpatterns = [
    path("polls/", include("polls.urls")),
    path("admin/", admin.site.urls),
]
-----+-----+
+----Starting Development Server-----+
-----+
Watching for file changes with StatReloader
[23/Sep/2024 13:12:35] "GET /polls/ HTTP/1.0" 200 13
Not Found: /favicon.ico
[23/Sep/2024 13:12:36] "GET /favicon.ico HTTP/1.0" 404 2354
```

- The screenshot below shows the output when the *IP address of my EC2 instance is accessed through a web browser*:



**All resources that were created for the lab (*EC2 instance*) were deleted using the AWS Console after the lab completion.**

---

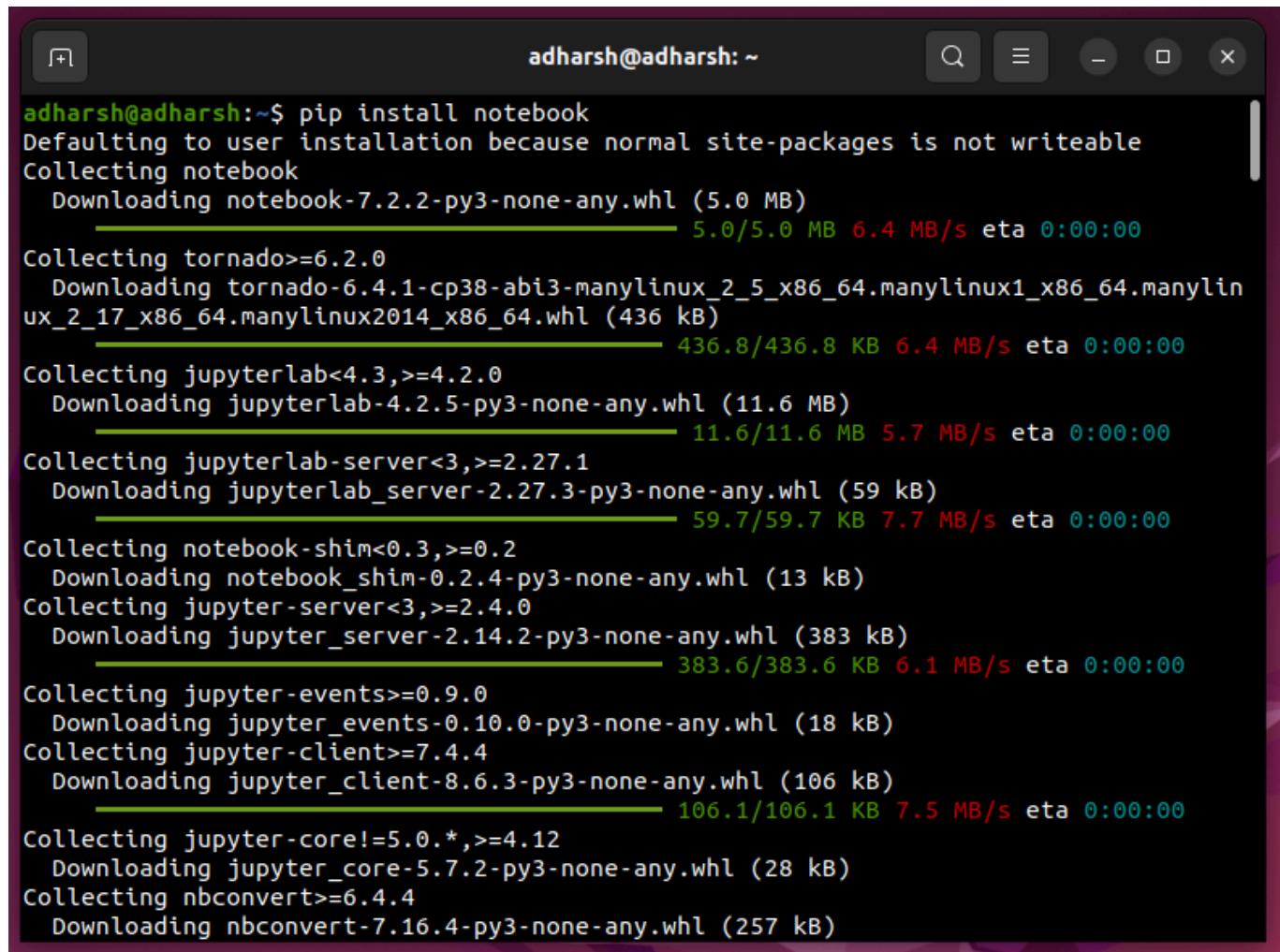
# Lab 8

## Install and run jupyter notebooks

- To install **Jupyter Notebook**, use the following command:

```
pip install notebook
```

- The screenshot below shows the output of the command:



adharsh@adharsh:~\$ pip install notebook  
Defaulting to user installation because normal site-packages is not writeable  
Collecting notebook  
 Downloading notebook-7.2.2-py3-none-any.whl (5.0 MB)  
 5.0/5.0 MB 6.4 MB/s eta 0:00:00  
Collecting tornado>=6.2.0  
 Downloading tornado-6.4.1-cp38-abi3-manylinux\_2\_5\_x86\_64.manylinux1\_x86\_64.manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (436 kB)  
 436.8/436.8 KB 6.4 MB/s eta 0:00:00  
Collecting jupyterlab<4.3,>=4.2.0  
 Downloading jupyterlab-4.2.5-py3-none-any.whl (11.6 MB)  
 11.6/11.6 MB 5.7 MB/s eta 0:00:00  
Collecting jupyterlab-server<3,>=2.27.1  
 Downloading jupyterlab\_server-2.27.3-py3-none-any.whl (59 kB)  
 59.7/59.7 KB 7.7 MB/s eta 0:00:00  
Collecting notebook-shim<0.3,>=0.2  
 Downloading notebook\_shim-0.2.4-py3-none-any.whl (13 kB)  
Collecting jupyter-server<3,>=2.4.0  
 Downloading jupyter\_server-2.14.2-py3-none-any.whl (383 kB)  
 383.6/383.6 KB 6.1 MB/s eta 0:00:00  
Collecting jupyter-events>=0.9.0  
 Downloading jupyter\_events-0.10.0-py3-none-any.whl (18 kB)  
Collecting jupyter-client>=7.4.4  
 Downloading jupyter\_client-8.6.3-py3-none-any.whl (106 kB)  
 106.1/106.1 KB 7.5 MB/s eta 0:00:00  
Collecting jupyter-core!=5.0.\*,>=4.12  
 Downloading jupyter\_core-5.7.2-py3-none-any.whl (28 kB)  
Collecting nbconvert>=6.4.4  
 Downloading nbconvert-7.16.4-py3-none-any.whl (257 kB)

- To run Jupyter Notebook, use the following command:

```
jupyter notebook
```

- The screenshot below shows the command execution and its output:

The terminal window shows the following log output:

```

adharsh@adharsh:~$ jupyter notebook
[I 2024-09-30 18:14:05.297 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-09-30 18:14:05.306 ServerApp] jupyter_server_terminals | extension was successfully linked.
[...]
[I 2024-09-30 18:14:05.303 ServerApp] JupyterLab | extension was successfully linked.
[I 2024-09-30 18:14:05.306 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-09-30 18:14:05.308 ServerApp] Writing Jupyter server cookie secret to /home/adharsh/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2024-09-30 18:14:05.548 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-09-30 18:14:05.565 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-09-30 18:14:05.567 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-09-30 18:14:05.568 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[...]
[I 2024-09-30 18:14:05.569 LabApp] JupyterLab extension loaded from /home/adharsh/.local/lib/python3.10/site-packages/jupyterlab
[...] 2024-09-30 18:14:05.570 LabApp] JupyterLab application directory is /home/adharsh/.local/share/jupyter/lab
[...] 2024-09-30 18:14:05.570 LabApp] Extension Manager is 'pyppi'.
[...] 2024-09-30 18:14:05.599 ServerApp] jupyterlab | extension was successfully loaded.
[...] 2024-09-30 18:14:05.602 ServerApp] notebook | extension was successfully loaded.
[...] 2024-09-30 18:14:05.602 ServerApp] Serving notebooks from local directory: /home/adharsh
[...] 2024-09-30 18:14:05.604 ServerApp] Jupyter Server 2.14.2 is running at:
[...] 2024-09-30 18:14:05.604 ServerApp] http://localhost:8888/tree?token=a3b86c1702e84d03a0ffdb703
edaa35e4f906ab49718ec86
[...] 2024-09-30 18:14:05.604 ServerApp] http://127.0.0.1:8888/tree?token=a3b86c1702e84d03a0ffdb703
b703eda35e4f906ab49718ec86
[...] 2024-09-30 18:14:05.604 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[...] 2024-09-30 18:14:05.670 ServerApp]

To access the server, open this file in a browser:
  file:///home/adharsh/.local/share/jupyter/runtime/jpserver-3762-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/tree?token=a3b86c1702e84d03a0ffdb703eda35e4f906ab49718ec86
  http://127.0.0.1:8888/tree?token=a3b86c1702e84d03a0ffdb703eda35e4f906ab49718ec86
[...] 2024-09-30 18:14:05.696 ServerApp] Skipped non-installed servers(s): bash-language-server, dock-erfile-language-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-lang-usage-server, pyright, python-language-server, python-lsp-server, r-languageserver, sol-language-s-erver, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server

```

The browser window shows the Jupyter interface with a list of files in the current directory.

## Install ipykernel

- To install **ipykernel**, use the following command:

```
pip install ipykernel
```

- The screenshot below shows the output of the command:

The terminal window shows the following output:

```

adharsh@adharsh:~$ pip install ipykernel
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: ipykernel in ./local/lib/python3.10/site-packages (6.29.5)
Requirement already satisfied: ipython>=7.23.1 in ./local/lib/python3.10/site-packages (from ipykernel) (8.27.0)
Requirement already satisfied: packaging in ./local/lib/python3.10/site-packages (from ipykernel) (24.1)
Requirement already satisfied: nest-asyncio in ./local/lib/python3.10/site-packages (from ipykernel) (1.6.0)
Requirement already satisfied: tornado>=6.1 in ./local/lib/python3.10/site-packages (from ipykernel) (6.4.1)
Requirement already satisfied: jupyter-client>=6.1.12 in ./local/lib/python3.10/site-packages (from ipykernel) (8.6.3)
Requirement already satisfied: traitlets>=5.4.0 in ./local/lib/python3.10/site-packages (from ipykernel) (5.14.3)
Requirement already satisfied: comm=0.1.1 in ./local/lib/python3.10/site-packages (from ipykernel) (0.2.2)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in ./local/lib/python3.10/site-packages (from ipykernel) (5.7.2)
Requirement already satisfied: matplotlib-inline!=0.1 in ./local/lib/python3.10/site-packages (from ipykernel) (0.1.7)
Requirement already satisfied: psutil in ./local/lib/python3.10/site-packages (from ipykernel) (6.0.0)
Requirement already satisfied: pyzmq>=24 in ./local/lib/python3.10/site-packages (from ipykernel) (26.2.0)
Requirement already satisfied: debugpy>=1.6.5 in ./local/lib/python3.10/site-packages (from ipykernel) (1.8.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in ./local/lib/python3.10/site-packages (from ipython>=7.23.1->ipykernel) (3.0.48)
Requirement already satisfied: pexpect>=4.3 in /usr/lib/python3/dist-packages (from ipython>=7.23.1->ipykernel) (4.8.0)
Requirement already satisfied: typing-extensions>=4.6 in ./local/lib/python3.10/site-packages (from ipython>=7.23.1->ipykernel) (4.12.2)
Requirement already satisfied: exceptiongroup in ./local/lib/python3.10/site-packages (from ipython>=7.23.1->ipykernel) (1.2.2)
Requirement already satisfied: stack-data in ./local/lib/python3.10/site-packages (from ipython>=7.23.1->ipykernel) (0.6.3)
Requirement already satisfied: pygments>=2.4.0 in /usr/lib/python3/dist-packages (from ipython>=7.23.1->ipykernel) (2.11.2)
Requirement already satisfied: decorator in ./local/lib/python3.10/site-packages (from ipython>=7.23.1->ipykernel) (5.1.1)
Requirement already satisfied: jedi>=0.16 in ./local/lib/python3.10/site-packages (from ipython>=7.23.1->ipykernel) (0.19.1)
Requirement already satisfied: python-dateutil>=2.8.2 in ./local/lib/python3.10/site-packages (from jupyter-client=6.1.12->ipykernel) (2.9.0.post0)
Requirement already satisfied: platformdirs>=2.5 in ./local/lib/python3.10/site-packages (from jupyter-core!=5.0.*,>=4.12->ipykernel) (4.3.6)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in ./local/lib/python3.10/site-packages (from jedi>=0.16->python>=7.23.1->ipykernel) (0.8.4)
Requirement already satisfied: wcwidth in ./local/lib/python3.10/site-packages (from prompt-toolkit<3.1.0,>=3.0.41->ipython>=7.23.1->ipykernel) (0.2.13)
Requirement already satisfied: stix>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.2->jupyter-client=6.1.12->ipykernel) (1.16.0)
Requirement already satisfied: pure-eval in ./local/lib/python3.10/site-packages (from stack-data->ipython>=7.23.1->ipykernel) (0.2.3)
Requirement already satisfied: executing>=1.2.0 in ./local/lib/python3.10/site-packages (from stack-data->ipython>=7.23.1->ipykernel) (2.1.0)
Requirement already satisfied: asttokens>=2.1.0 in ./local/lib/python3.10/site-packages (from stack-data->ipython>=7.23.1->ipykernel) (2.4.1)

```

## Run hyperparameter tuning jobs

- I first downloaded the [LabAI.ipynb](#).
- Then, I opened the file using Jupyter Notebook.
- I followed the instructions in the notebook.

- One of the instructions, asks us to create a S3 bucket and create appropriate objects(folders) in the same.
- I used the code below to create the bucket, which is a modified version of **Lab 3** code:

```
import boto3

#Bucket configuration
name = '23796349-lab8' #The name that was recommended in the notebook
bucket_config = {'LocationConstraint': 'ap-northeast-2'} #My specified region

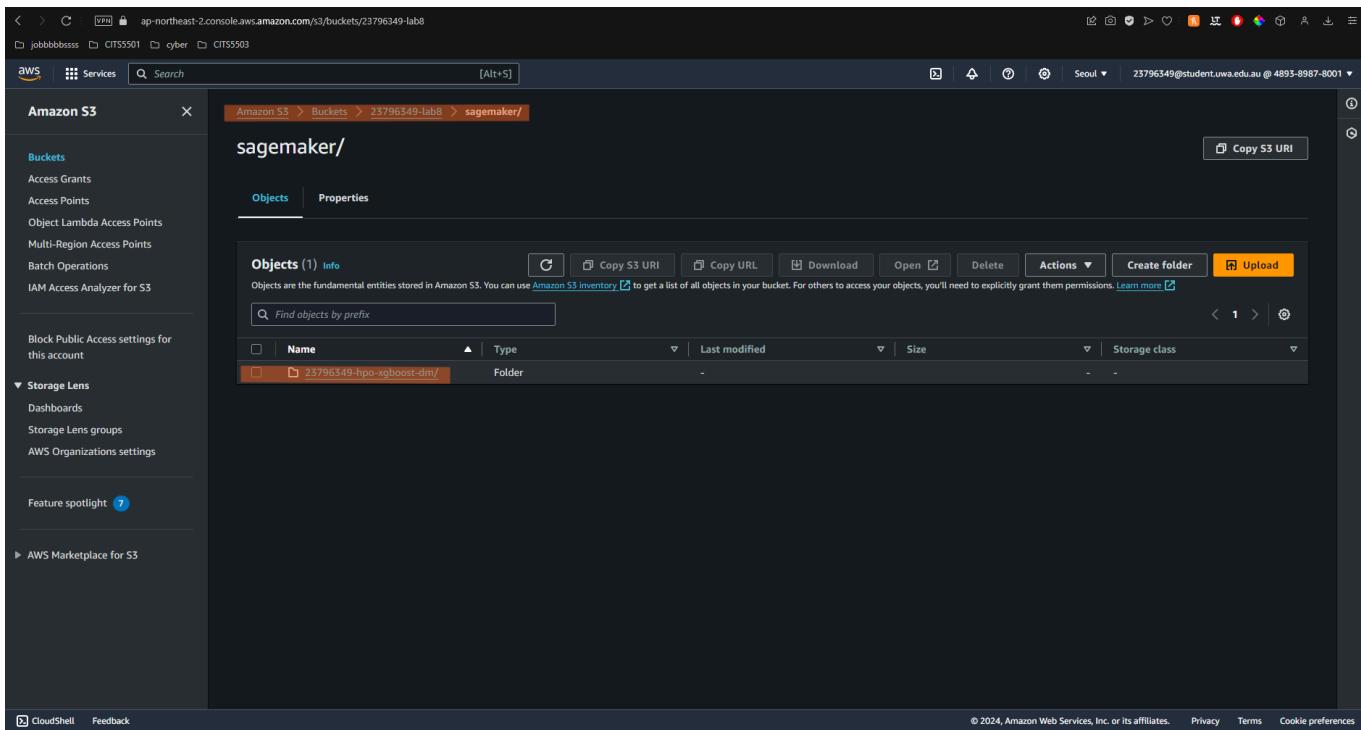
#S3 client
s3 = boto3.client("s3")

try:
    s3.create_bucket(Bucket=name, CreateBucketConfiguration=bucket_config)
#create_bucket function is idempotent.
    response = s3.head_bucket(Bucket=name)
    print(f"Bucket '{name}' created and exists.")
    print(response)
except Exception as error:
    pass
```

- The screenshot below shows the created S3 bucket:

Name	AWS Region	IAM Access Analyzer	Creation date
23796349-lab8	Asia Pacific (Seoul) ap-northeast-2	<a href="#">View analyzer for ap-northeast-2</a>	September 30, 2024, 18:32:45 (UTC+08:00)

- Then, I used AWS Console to create the required objects (**folders - sagemaker/23796349-hpo-xgboost-dm**) in my S3 Bucket.
- The screenshot below shows the objects (folders) created using AWS Console:



- I made the necessary changes to the variables as instructed by the comments in the notebook. The screenshot below shows the changes I made:

## Prepare a SageMaker session

```
[2]: import sagemaker
import boto3

import numpy as np # For matrix operations and numerical processing
import pandas as pd # For munging tabular data
from time import gmtime, strftime
import os

smclient = boto3.Session().client("sagemaker")
iam = boto3.client('iam')
sagemaker_role = iam.get_role(RoleName='SageMakerRole')['Role']['Arn']
region = 'ap-northeast-2' # use the region you are mapped to
student_id = "23796349" # use your student id
bucket = '23796349-lab8' # use <studentid-lab8> as your bucket name
prefix = f"sagemaker/{student_id}-hpo-xgboost-dm"
# Create an S3 bucket using the bucket variable above. The bucket creation is done using the region variable above.
# Create an object into the bucket. The object is a folder and its name is the prefix variable above.

sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/xdg-ubuntu/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/adharsh/.config/sagemaker/config.yaml
```

- I ran all the code blocks and the output of the **final code block** is shown in the screenshot below:

```
[48]: #Launch Hyperparameter Tuning Job
smclient.create_hyper_parameter_tuning_job(
    HyperParameterTuningJobName=tuning_job_name,
    HyperParameterTuningJobConfig=tuning_job_config,
    TrainingJobDefinition=training_job_definition,
)

[48]: {'HyperParameterTuningJobArn': 'arn:aws:sagemaker:ap-northeast-2:489389878001:hyper-parameter-tuning-job/23796349-xgboost-tuningjob-03',
'ResponseMetadata': {'RequestId': '63741a35-5af7-4d98-a857-1ea62ce212ed',
'HTTPStatusCode': 200,
'HTTPHeaders': {'x-amzn-requestid': '63741a35-5af7-4d98-a857-1ea62ce212ed',
'content-type': 'application/x-amz-json-1.1',
'content-length': '135',
'date': 'Mon, 30 Sep 2024 11:46:05 GMT'},
'RetryAttempts': 0}}
```

- The execution of all the code blocks and the final block resulted in the creation of objects(folders and files) in the bucket. This is the result of training the model and validating it.
- The screenshot below shows the content in the folder **23796349-hpo-xgboost-dm**:

The screenshot shows the AWS S3 console interface. The left sidebar has sections for Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main area displays the contents of the bucket '23796349-hpo-xgboost-dm/'. The 'Objects' tab is selected, showing three items: 'output/' (Folder), 'train/' (Folder), and 'validation/' (Folder). A search bar at the top says 'Find objects by prefix'. Below the search bar is a table with columns: Name, Type, Last modified, Size, and Storage class. The 'Actions' button is highlighted in orange.

- The screenshot below shows the content in the folders **train** and **validation**:

Two side-by-side screenshots of the AWS S3 console. The left screenshot shows the 'train/' folder containing a single CSV file named 'train.csv'. The right screenshot shows the 'validation/' folder containing a single CSV file named 'validation.csv'. Both files have a size of 0 bytes and were last modified on September 30, 2024, at 08:00 UTC+08:00. The 'Objects' tab is selected in both panels.

- The screenshot below shows the **Hyperparameter Tuning Jobs**:

The screenshot shows the AWS SageMaker console under the 'Admin configurations' section. The left sidebar includes Admin configurations (Domains, Role manager, Images, Lifecycle configurations), SageMaker dashboard, Search, and JumpStart. The main area shows the 'Hyperparameter tuning jobs' list. There is a search bar at the top. The table has columns: Name, Status, Training completed/total, Creation time, and Duration. The first job, '23796349-xgboost-tuningjob-03', is completed with 2/2 training completed. The second job, '23796349-xgboost-tuningjob-02', and the third job, '23796349-xgboost-tuningjob-01', both failed with 0/2 training failed. The 'Create hyperparameter tuning job' button is highlighted in orange.

**NOTE:**

I encountered the Non-numeric type error, that is why you could see more than one hyperparameter tuning jobs.

To solve this problem, I converted the Non-numeric data to Numeric data (True/False to 1/0 respectively).

The screenshot below shows the code block that contains the code to convert Non-numeric data to Numeric data before uploading to the S3 bucket:

```
[44]: #Converting True/False to 1/0 respectively.

#train data
train_df = pd.read_csv('train.csv', header=None)
train_df = train_df.astype(int)
train_df.to_csv('train.csv', index=False, header=False)

#validation data
valid_df = pd.read_csv('validation.csv', header=None)
valid_df = valid_df.astype(int)
valid_df.to_csv('validation.csv', index=False, header=False)
```

Copy the file to the S3 bucket created earlier for Amazon SageMaker training to pick up.

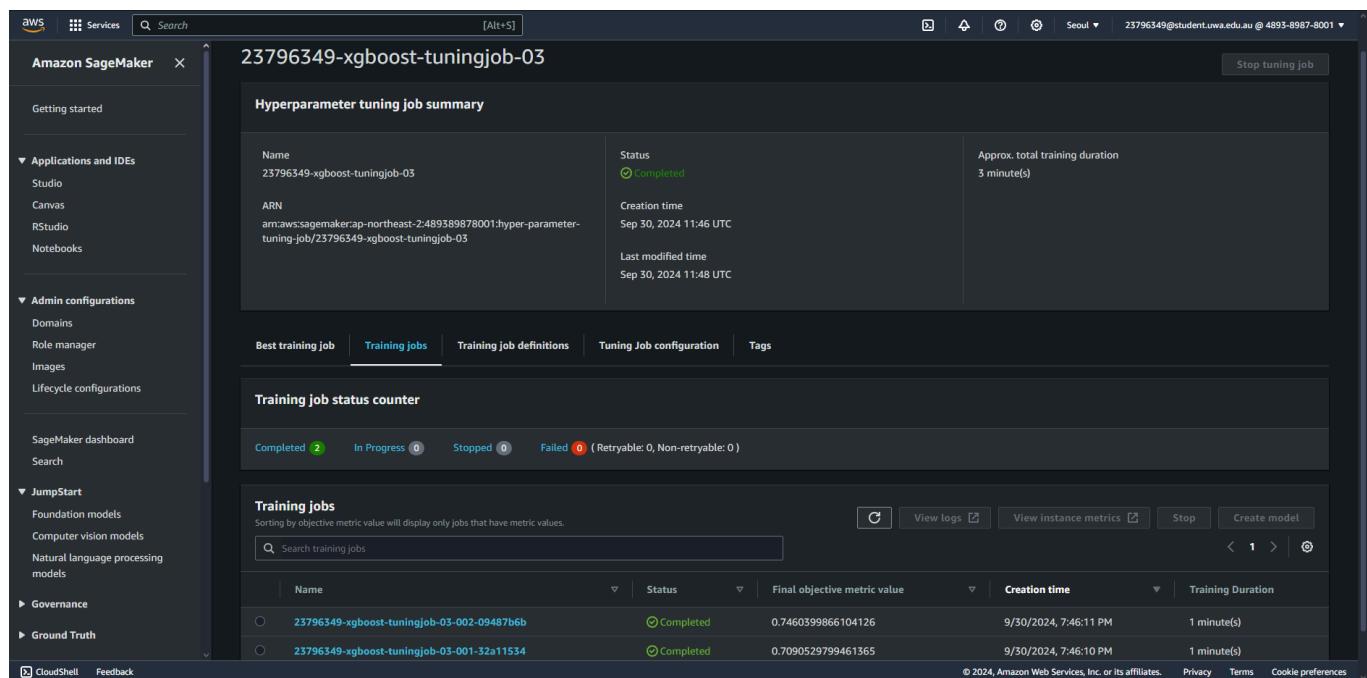
```
[45]: boto3.Session().resource("s3").Bucket(bucket).Object(
    os.path.join(prefix, "train/train.csv")
).upload_file("train.csv")
boto3.Session().resource("s3").Bucket(bucket).Object(
    os.path.join(prefix, "validation/validation.csv")
).upload_file("validation.csv")
```

```
#Converting True/False to 1/0 respectively

#train data
train_df = pd.read_csv('train.csv', header=None)
train_df = train_df.astype(int)
train_df.to_csv('train.csv', index=False, header=False)

#validation data
valid_df = pd.read_csv('validation.csv', header=None)
valid_df = valid_df.astype(int)
valid_df.to_csv('validation.csv', index=False, header=False)
```

- The screenshot below shows the details (the associated training jobs) of **Successful Tuning job**:



- The screenshot below shows the contents of the **output** folder:

The screenshot shows the AWS S3 console interface. The left sidebar includes links for Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main area displays the contents of the 'output/' folder in the '23796349-hpo-xgboost-dm/' bucket. The 'Objects' tab is selected, showing a list of items with columns for Name, Type, Last modified, Size, and Storage class. Two items are listed as 'Folder': '23796349-xgboost-tuningjob-03-001-32a11534/' and '23796349-xgboost-tuningjob-03-002-09487b6b/'. The bottom right corner of the interface shows copyright information: © 2024, Amazon Web Services, Inc. or its affiliates.

- The screenshot below shows the **models** created:

The image displays two side-by-side screenshots of the AWS S3 console. Both screenshots show the same directory structure: 'Amazon S3 > ... > sagemaker/ > 23796349-hpo-x... > output/ > 23796349-xgboost-tu... > output/'. In both cases, the 'Objects' tab is selected, and the list shows a single item: 'model.tar.gz'. The details for this item show it is a 'gz' file, last modified on September 30, 2024, at 08:00 UTC. The bottom right corner of each screenshot shows copyright information: © 2024, Amazon Web Services, Inc. or its affiliates.

- I have linked my Jupyter Notebook [here](#).

**All resources that were created for the lab (*S3 Bucket and the objects in it*) were deleted using the AWS Console after the lab completion.**

# Lab 9

---

## AWS Comprehend

### Detect Languages from text

#### [1] Modify the code above

- The code given in the lab sheet was modified to display output in the specified format.

```
import boto3
import langcodes

#Sample
sentence = "Hi my name is John. Nice to meet you!"

client = boto3.client('comprehend')
response = client.detect_dominant_language(Text=sentence)
lng_code = response[ 'Languages' ][0][ 'LanguageCode' ]
score = response[ 'Languages' ][0][ 'Score' ]
lng_detected = langcodes.get(lng_code).language_name()
print(f"For the sentence:\n\"{sentence}\"\n")
print("Detection")
print("-----")
print(f"{lng_detected} detected with {int(score*100)}% confidence!")
```

- To do this, I had to change the *Language Code* returned by the `detect_dominant_language` method to the respective full form (eg. "en" should be converted to "English").
- I searched and found that there is a python library called `langcodes` that returns the full language name given a language code.
- I used the following command to install the library:

```
pip install langcodes
```

- The screenshot below shows the output of the **modified code**:

```
adharsh@adharsh:~/cits5503/Lab9$ python3 detectlng.py
For the sentence:
"Hi my name is John. Nice to meet you!"

Detection
-----
English detected with 99% confidence!
adharsh@adharsh:~/cits5503/Lab9$
```

#### [2] Test your code with other languages

- The next task was to test the modified code with other lanuages.
- I created a list that contains the text provided in the Lab sheet as its elements.
- The code below shows my implementation:

```

import boto3
import langcodes

#Sample
sentence = "Hi my name is John. Nice to meet you!"

#Test text provided in Lab Sheet
test = ["The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799.",  

        "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha.",  

        "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir",  

        "L'amor che move il sole e l'altre stelle."]
for sentence in test:  

    client = boto3.client('comprehend')
    response = client.detect_dominant_language(Text=sentence)
    lng_code = response['Languages'][0]['LanguageCode']
    score = response['Languages'][0]['Score']
    lng_detected = langcodes.get(lng_code).language_name()
    print(f"For the sentence:\n\"{sentence}\"\n")
    print("Detection")
    print("-----")
    print(f"{lng_detected} detected with {int(score*100)}% confidence!")
    print("\n")

```

- The screenshot below shows the **output**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 detectLng.py
For the sentence:
"The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799."
Detection
-----
English detected with 99% confidence!

For the sentence:
"El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha."
Detection
-----
Spanish detected with 99% confidence!

For the sentence:
"Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir"
Detection
-----
French detected with 99% confidence!

For the sentence:
"L'amor che move il sole e l'altre stelle."
Detection
-----
Italian detected with 99% confidence!

adharsh@adharsh:~/cits5503/lab9$
```

**NOTE:** I used the **detect\_dominant\_language** method to get and feed the appropriate language code in the following sections.

## Analyze sentiment

- The following code shows the implementation of the **detect\_sentiment** method to return the sentiment of a given sentence.

```
import boto3

#Test provided in Lab Sheet
test = ["The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799.",  
        "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra.  
Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha.",  
        "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir",  
        "L'amor che move il sole e l'altre stelle."]

for sentence in test:  
    client = boto3.client('comprehend')  
    response_lng = client.detect_dominant_language(Text=sentence)  
    lng_code = response_lng[ 'Languages'][0][ 'LanguageCode']  
    response_sent = client.detect_sentiment(Text=sentence, LanguageCode=lng_code)  
    sentiment = response_sent[ 'Sentiment']  
    score = response_sent[ 'SentimentScore'][sentiment.capitalize()]  
    print(f"For \"{sentence}\"\n\nThe sentiment detected is  
{sentiment.capitalize()} with {int(score*100)}% confidence!")  
    print("\n")
```

- The screenshot below shows the **output**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 sentanal.py
For "The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799."
The sentiment detected is Neutral with 99% confidence!

For "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha."
The sentiment detected is Neutral with 98% confidence!

For "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir"
The sentiment detected is Positive with 96% confidence!

For "L'amor che move il sole e l'altra stelle."
The sentiment detected is Positive with 99% confidence!

adharsh@adharsh:~/cits5503/lab9$
```

## Detect entities

- The following code shows the implementation of `detect_entities` method to return the entities found in a given sentence.

```

import boto3

#Test provided in Lab Sheet
test = ["The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799.",
        "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha.",
        "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir",
        "L'amor che move il sole e l'altra stelle."]

for sentence in test:
    client = boto3.client('comprehend')
    response_lng = client.detect_dominant_language(Text=sentence)
    lng_code = response_lng['Languages'][0]['LanguageCode']
    reponse_ent = client.detect_entities(Text=sentence, LanguageCode=lng_code)
    entities = reponse_ent['Entities']
    print(f"In the sentence:\n\"{sentence}\"\n")
    print("Detection")
    print("-----")
    if len(entities) != 0:
        for entity in entities:
            print(f"Entity: {entity['Text']} | Type: {entity['Type']} | "
                  f"Confidence: {int(entity['Score']*100)}%")
    else:
```

```

print("None")
print("\n")

```

- The screenshot below shows the **output**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 entdetect.py
In the sentence:
"The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799."
Detection
-----
Entity: French Revolution | Type: EVENT | Confidence: 98%
Entity: France | Type: LOCATION | Confidence: 98%
Entity: 1789 | Type: DATE | Confidence: 99%
Entity: 1799 | Type: DATE | Confidence: 99%

In the sentence:
"El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El Ingenioso caballero don Quijote de la Mancha."
Detection
-----
Entity: El Quijote | Type: TITLE | Confidence: 96%
Entity: Miguel de Cervantes Saavedra | Type: PERSON | Confidence: 99%
Entity: primera parte | Type: QUANTITY | Confidence: 86%
Entity: El ingenioso hidalgo don Quijote de la Mancha | Type: TITLE | Confidence: 87%
Entity: 1605 | Type: DATE | Confidence: 79%
Entity: una de | Type: QUANTITY | Confidence: 59%
Entity: española | Type: OTHER | Confidence: 98%
Entity: una de las más | Type: QUANTITY | Confidence: 63%
Entity: 1615 | Type: DATE | Confidence: 98%
Entity: segunda parte | Type: QUANTITY | Confidence: 88%
Entity: Quijote de Cervantes | Type: TITLE | Confidence: 74%
Entity: El ingenioso caballero don Quijote de la Mancha | Type: TITLE | Confidence: 91%

In the sentence:
"Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir"
Detection
-----
Entity: aujourd'hui | Type: DATE | Confidence: 98%
Entity: Tout ce qu' | Type: QUANTITY | Confidence: 66%

In the sentence:
"L'amor che move il sole e l'altra stelle."
Detection
-----
None

```

- Answer to lab sheet question:** Entities can be described as specific pieces of information that are found and categorized in a given text. (eg. Location, Dates, Events, People, etc.)

## Detect keyphrases

- The following code shows the implementation of `detect_key_phrases` method to return the keyphrases found in a given sentence.

```

import boto3

#Test provided in Lab Sheet
test = ["The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799.",
        "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha.",
        "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir",
        "L'amor che move il sole e l'altra stelle."]

```

```

for sentence in test:
    client = boto3.client('comprehend')
    response_lng = client.detect_dominant_language(Text=sentence)
    lng_code = response_lng[ 'Languages' ][0][ 'LanguageCode' ]
    response_kpd = client.detect_key_phrases(Text=sentence,
    LanguageCode=lng_code)
    keyPhrases = response_kpd[ 'KeyPhrases' ]
    print(f"In the sentence:\n\"{sentence}\"\n")
    print("Detection")
    print("-----")
    for kp in keyPhrases:
        print(f"Keyphrase: {kp[ 'Text' ]} | Confidence: {int(kp[ 'Score' ]*100)}%")
    print("\n")

```

- The screenshots below show the **output**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 kphdetect.py
In the sentence:
"The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799."
Detection
-----
Keyphrase: The French Revolution | Confidence: 99%
Keyphrase: a period | Confidence: 99%
Keyphrase: social and political upheaval | Confidence: 99%
Keyphrase: France | Confidence: 99%
Keyphrase: its colonies | Confidence: 99%
Keyphrase: 1789 | Confidence: 99%
Keyphrase: 1799 | Confidence: 99%

In the sentence:
"El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha."
Detection
-----
Keyphrase: El Quijote | Confidence: 99%
Keyphrase: la obra | Confidence: 99%
Keyphrase: más conocida | Confidence: 99%
Keyphrase: Miguel de Cervantes Saavedra | Confidence: 99%
Keyphrase: su primera parte | Confidence: 99%
Keyphrase: el título | Confidence: 99%
Keyphrase: El ingenioso hidalgo don Quijote de la Mancha | Confidence: 95%
Keyphrase: comienzos | Confidence: 99%
Keyphrase: 1605 | Confidence: 99%
Keyphrase: las obras | Confidence: 99%
Keyphrase: más destacadas | Confidence: 99%
Keyphrase: la literatura española | Confidence: 99%
Keyphrase: la literatura universal | Confidence: 99%
Keyphrase: las más traducidas | Confidence: 99%
Keyphrase: la segunda parte | Confidence: 99%
Keyphrase: Quijote de Cervantes | Confidence: 99%
Keyphrase: el título | Confidence: 99%
Keyphrase: ingenioso caballero don Quijote de la Mancha | Confidence: 93%

In the sentence:
"Mot je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir"
Detection
-----
Keyphrase: Moi | Confidence: 99%
Keyphrase: je | Confidence: 95%
Keyphrase: n'étais rien | Confidence: 95%
Keyphrase: aujourd'hui | Confidence: 94%
Keyphrase: Je suis le gardien Du sommeil de ses nuits | Confidence: 94%
Keyphrase: Je | Confidence: 99%
Keyphrase: l' | Confidence: 99%
Keyphrase: Vous | Confidence: 99%
Keyphrase: Tout ce | Confidence: 98%
Keyphrase: qu' | Confidence: 96%
Keyphrase: il | Confidence: 99%
Keyphrase: vous | Confidence: 99%
Keyphrase: Elle | Confidence: 99%
Keyphrase: L'espace de ses bras | Confidence: 99%
Keyphrase: tout | Confidence: 89%
Keyphrase: tout | Confidence: 96%
Keyphrase: Je | Confidence: 99%
Keyphrase: l' | Confidence: 99%

In the sentence:
"L'amor che move il sole e l'altra stelle."
Detection
-----
Keyphrase: L'amor | Confidence: 99%
Keyphrase: che | Confidence: 99%
Keyphrase: il sole | Confidence: 99%
Keyphrase: l'altra stelle | Confidence: 99%

```

- Answer to lab sheet question:** Keyphrases can be described as crucial expressions that give the main idea of a given text.

## Detect syntaxes

- The following code shows the implementation of `detect_syntax` method to return the syntaxes found in a given sentence.

```
import boto3

#Test provided in Lab Sheet
test = ["The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799.",
        "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha.",
        "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir",
        "L'amor che move il sole e l'altre stelle."]

for sentence in test:
    client = boto3.client('comprehend')
    response_lng = client.detect_dominant_language(Text=sentence)
    lng_code = response_lng['Languages'][0]['LanguageCode']
    response_syntax = client.detect_syntax(Text=sentence, LanguageCode=lng_code)
    syntaxTokens = response_syntax['SyntaxTokens']
    print(f"In the sentence:\n\"{sentence}\"\n")
    print("Detection")
    print("-----")
    for tk in syntaxTokens:
        print(f"Word: {tk['Text']} | Part of Speech: {tk['PartOfSpeech']['Tag']}")
    | Confidence: {int(tk['PartOfSpeech']['Score']*100)}%")
    print("\n")
```

- The screenshots below show the **output**:

```
adharsh@adharsh:~/cits5503/lab9$ python3 syndetect.py
In the sentence:
"The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799."
Detection
-----
Word: The | Part of Speech: DET | Confidence: 100%
Word: French | Part of Speech: PROPN | Confidence: 100%
Word: Revolution | Part of Speech: PROPN | Confidence: 100%
Word: was | Part of Speech: VERB | Confidence: 100%
Word: a | Part of Speech: DET | Confidence: 100%
Word: period | Part of Speech: NOUN | Confidence: 100%
Word: of | Part of Speech: ADP | Confidence: 100%
Word: social | Part of Speech: ADJ | Confidence: 100%
Word: and | Part of Speech: CCONJ | Confidence: 99%
Word: political | Part of Speech: ADJ | Confidence: 100%
Word: upheaval | Part of Speech: NOUN | Confidence: 100%
Word: in | Part of Speech: ADP | Confidence: 100%
Word: France | Part of Speech: PROPN | Confidence: 100%
Word: and | Part of Speech: CCONJ | Confidence: 100%
Word: its | Part of Speech: PRON | Confidence: 100%
Word: colonies | Part of Speech: NOUN | Confidence: 100%
Word: beginning | Part of Speech: VERB | Confidence: 100%
Word: in | Part of Speech: ADP | Confidence: 100%
Word: 1789 | Part of Speech: NUM | Confidence: 100%
Word: and | Part of Speech: CCONJ | Confidence: 100%
Word: ending | Part of Speech: VERB | Confidence: 100%
Word: in | Part of Speech: ADP | Confidence: 100%
Word: 1799 | Part of Speech: NUM | Confidence: 100%
Word: . | Part of Speech: PUNCT | Confidence: 100%
```

In the sentence:  
"El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El Ingenioso caballero don Quijote de la Mancha."

```
Detection
-----
Word: El | Part of Speech: DET | Confidence: 99%
Word: Quijote | Part of Speech: PROPN | Confidence: 99%
Word: es | Part of Speech: VERB | Confidence: 99%
Word: la | Part of Speech: DET | Confidence: 100%
Word: obra | Part of Speech: NOUN | Confidence: 99%
Word: más | Part of Speech: ADV | Confidence: 100%
Word: conocida | Part of Speech: ADJ | Confidence: 69%
Word: de | Part of Speech: ADP | Confidence: 100%
Word: Miguel | Part of Speech: PROPN | Confidence: 99%
Word: de | Part of Speech: ADP | Confidence: 100%
Word: Cervantes | Part of Speech: PROPN | Confidence: 99%
Word: Saavedra | Part of Speech: PROPN | Confidence: 100%
Word: . | Part of Speech: PUNCT | Confidence: 100%
Word: Publicada | Part of Speech: VERB | Confidence: 99%
Word: su | Part of Speech: DET | Confidence: 99%
Word: primera | Part of Speech: ADJ | Confidence: 99%
Word: parte | Part of Speech: NOUN | Confidence: 99%
Word: con | Part of Speech: ADP | Confidence: 100%
Word: el | Part of Speech: DET | Confidence: 100%
Word: título | Part of Speech: NOUN | Confidence: 99%
Word: de | Part of Speech: ADP | Confidence: 100%
Word: El | Part of Speech: DET | Confidence: 99%
Word: Ingenioso | Part of Speech: PROPN | Confidence: 99%
Word: hidalgo | Part of Speech: PROPN | Confidence: 99%
Word: don | Part of Speech: PROPN | Confidence: 99%
Word: Quijote | Part of Speech: PROPN | Confidence: 99%
Word: de | Part of Speech: ADP | Confidence: 100%
Word: la | Part of Speech: DET | Confidence: 99%
Word: Mancha | Part of Speech: PROPN | Confidence: 99%
Word: a | Part of Speech: ADP | Confidence: 100%
Word: comienzos | Part of Speech: NOUN | Confidence: 99%
Word: de | Part of Speech: ADP | Confidence: 100%
Word: 1605 | Part of Speech: NUM | Confidence: 100%
Word: , | Part of Speech: PUNCT | Confidence: 100%
Word: es | Part of Speech: VERB | Confidence: 99%
Word: una | Part of Speech: PRON | Confidence: 100%
Word: de | Part of Speech: ADP | Confidence: 100%
Word: las | Part of Speech: DET | Confidence: 100%
Word: obras | Part of Speech: NOUN | Confidence: 100%
Word: más | Part of Speech: ADV | Confidence: 99%
Word: destacadas | Part of Speech: ADJ | Confidence: 99%
Word: de | Part of Speech: ADP | Confidence: 100%
Word: la | Part of Speech: DET | Confidence: 100%
Word: literatura | Part of Speech: NOUN | Confidence: 99%
Word: española | Part of Speech: ADJ | Confidence: 99%
Word: y | Part of Speech: CCONJ | Confidence: 100%
Word: la | Part of Speech: DET | Confidence: 99%
Word: literatura | Part of Speech: NOUN | Confidence: 99%
```

```
In the sentence:  
"Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir  
L'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir"
```

Detection

```
-----  
Word: Moi | Part of Speech: PRON | Confidence: 99%  
Word: Je | Part of Speech: PRON | Confidence: 100%  
Word: n' | Part of Speech: ADV | Confidence: 99%  
Word: étais | Part of Speech: AUX | Confidence: 99%  
Word: rien | Part of Speech: PRON | Confidence: 99%  
Word: Et | Part of Speech: CCONJ | Confidence: 100%  
Word: voilà | Part of Speech: VERB | Confidence: 100%  
Word: qu' | Part of Speech: SCONJ | Confidence: 99%  
Word: aujourd'hui | Part of Speech: NOUN | Confidence: 99%  
Word: Je | Part of Speech: PRON | Confidence: 100%  
Word: suis | Part of Speech: AUX | Confidence: 99%  
Word: le | Part of Speech: DET | Confidence: 99%  
Word: gardien | Part of Speech: NOUN | Confidence: 99%  
Word: Du | Part of Speech: ADP | Confidence: 99%  
Word: sommeil | Part of Speech: NOUN | Confidence: 99%  
Word: de | Part of Speech: ADP | Confidence: 100%  
Word: ses | Part of Speech: DET | Confidence: 99%  
Word: nuits | Part of Speech: NOUN | Confidence: 99%  
Word: Je | Part of Speech: PRON | Confidence: 100%  
Word: l' | Part of Speech: PRON | Confidence: 99%  
Word: aime | Part of Speech: VERB | Confidence: 100%  
Word: à | Part of Speech: ADP | Confidence: 100%  
Word: mourir | Part of Speech: VERB | Confidence: 99%  
Word: Vous | Part of Speech: PRON | Confidence: 100%  
Word: pouvez | Part of Speech: VERB | Confidence: 99%  
Word: détruire | Part of Speech: VERB | Confidence: 100%  
Word: Tout | Part of Speech: DET | Confidence: 92%  
Word: ce | Part of Speech: PRON | Confidence: 99%  
Word: qu' | Part of Speech: PRON | Confidence: 99%  
Word: il | Part of Speech: PRON | Confidence: 100%  
Word: vous | Part of Speech: PRON | Confidence: 100%  
Word: plaira | Part of Speech: VERB | Confidence: 99%  
Word: Elle | Part of Speech: PRON | confidence: 100%  
Word: n' | Part of Speech: ADV | Confidence: 99%  
Word: a | Part of Speech: VERB | Confidence: 99%  
Word: qu' | Part of Speech: SCONJ | Confidence: 71%  
Word: à | Part of Speech: ADP | Confidence: 100%  
Word: ouvrir | Part of Speech: VERB | Confidence: 99%  
Word: L' | Part of Speech: DET | Confidence: 100%  
Word: espace | Part of Speech: NOUN | Confidence: 99%  
Word: de | Part of Speech: ADP | Confidence: 100%  
Word: ses | Part of Speech: DET | Confidence: 99%  
Word: bras | Part of Speech: NOUN | Confidence: 99%  
Word: Pour | Part of Speech: ADP | Confidence: 99%  
Word: tout | Part of Speech: PRON | Confidence: 99%  
Word: reconstruire | Part of Speech: VERB | Confidence: 100%  
Word: Pour | Part of Speech: ADP | Confidence: 99%
```

```
In the sentence:  
"L'amor che move il sole e l'altra stelle."
```

Detection

```
-----  
Word: L' | Part of Speech: DET | Confidence: 100%  
Word: amor | Part of Speech: NOUN | Confidence: 100%  
Word: che | Part of Speech: PRON | Confidence: 100%  
Word: move | Part of Speech: VERB | Confidence: 100%  
Word: il | Part of Speech: DET | Confidence: 100%  
Word: sole | Part of Speech: NOUN | confidence: 100%  
Word: e | Part of Speech: CCONJ | confidence: 100%  
Word: l' | Part of Speech: DET | Confidence: 100%  
Word: altra | Part of Speech: ADJ | Confidence: 100%  
Word: stelle | Part of Speech: NOUN | Confidence: 100%  
Word: . | Part of Speech: PUNCT | Confidence: 100%
```

```
adharsh@adharsh:~/cits5503/lab9$
```

- **Answer to lab sheet question:** Syntaxes can be described as each individual word in a given text that can be used to identify the grammatical structure of the sentences in the text.

## AWS Rekognition

### Add images

- First, I downloaded the required images to complete the task in this section.
- These are the images that I chose.
- The image **urban.jpg**:



- The image **beach.jpg**:



- The image **faces.jpg**:



Image ID: 2GB11JC  
[www.alamy.com](http://www.alamy.com)

- The image **text.jpg**:

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness...

- The following code creates a S3 bucket named **23796349-lab9** in the region that I am mapped to (ap-northeast-2) and uploads the four images, it is a modification of code from [Lab 5](#):

```
import boto3
import os

#Bucket configuration
name = '23796349-lab9'
bucket_config = {'LocationConstraint': 'ap-northeast-2'}

#S3 client
s3 = boto3.client("s3")

#File paths
```

```

file_paths = [
    '/home/cits5503/lab9/beach.jpg',
    '/home/cits5503/lab9/faces.jpg',
    '/home/cits5503/lab9/urban.jpg',
    '/home/cits5503/lab9/text.jpg'
]

try:
    s3.create_bucket(Bucket=name, CreateBucketConfiguration=bucket_config)
#create_bucket function is idempotent.
    response = s3.head_bucket(Bucket=name)
    print(f"Bucket '{name}' created and exists.")
    print(response)
except Exception as error:
    pass

for file_path in file_paths:
    file_name = os.path.basename(file_path)
    s3.upload_file(file_path, bucket_name, file_name)
    print(f'Uploaded {file_name} to {bucket_name}')

```

## Test AWS rekognition

- The code below shows the implementation of `detect_labels`, `detect_moderation_labels`, `detect_faces`, and `detect_text` methods to return appropriate labels and their confidence level.
- I used argument parser to have the choice to run only the required method.
- The user has to specify two options, the method they want to use and the photo they want to use that is present in their S3 bucket.

```

import boto3,argparse

#Constants
bucket = '23796349-lab9'
client = boto3.client('rekognition')

# Argument Parsing
parser = argparse.ArgumentParser(description='AWS_Rekognition_options')
parser.add_argument('--labelrecog', '-l', action='store_true', help='Label Recognition')
parser.add_argument('--imgmoderation', '-i', action='store_true', help='Image Moderation')
parser.add_argument('--faceanalyse', '-f', action='store_true', help='Facial Analysis')
parser.add_argument('--exttxt', '-e', action='store_true', help='Extract Text')
parser.add_argument('--photo', '-p', required=True, help='Photo name with extension present in your S3 bucket')
args = parser.parse_args()

#Label Recognition
def detectLabels(photo_name):
    reponse = client.detect_labels(Image={'S3Object':
{'Bucket':bucket,'Name':photo_name}})
```

```

print("Detection")
print("-----")
for label in reponse['Labels']:
    print(f"Label: {label[ 'Name']} | Confidence:
{int(label['Confidence'])}%)")

#Image Moderation
def imgMod(photo_name):
    reponse = client.detect_moderation_labels(Image={'S3Object':
{'Bucket':bucket,'Name':photo_name}})

    print("Detection")
    print("-----")
    for label in reponse['ModerationLabels']:
        print(f"Label: {label[ 'Name']} | Confidence:
{int(label['Confidence'])}%)")

#Facial Analysis
def analyFace(photo_name):
    response = client.detect_faces(Image={'S3Object':
{'Bucket':bucket,'Name':photo_name}}, Attributes=['ALL'])
    faceCount = 0
    for face in response['FaceDetails']:
        faceCount+=1
        print("Face Number: "+ str(faceCount))
        print("Detection")
        print("-----")
        print("Gender: "+face[ 'Gender'][ 'Value']+ " | Confidence:
"+str(int(face[ 'Gender'][ 'Confidence'])))+"%")
        print("Age Range (years): "+str(face[ 'AgeRange'][ 'Low'])+ " to
"+str(face[ 'AgeRange'][ 'High']))
        print("Glasses: "+str(face[ 'Eyglasses'][ 'Value']+ " | Confidence:
"+str(int(face[ 'Eyglasses'][ 'Confidence'])))+"%")
        print("-----")
        print("-----")

#Extract Text
def extText(photo_name):
    response = client.detect_text(Image={'S3Object':
{'Bucket':bucket,'Name':photo_name}})

    print("Detection")
    print("-----")
    for text in response['TextDetections']:
        print(f"Text detected: {text[ 'DetectedText']} | Confidence:
{int(text[ 'Confidence'])}%)")

#Calling appropriate function based on the option specified
if args.labelrecog:
    detectLabels(args.photo)

if args.imgmoderation:
    imgMod(args.photo)

if args.faceanalyse:
    analyFace(args.photo)

```

```

if args.exttxt:
    extText(args.photo)

```

- The screenshot below shows the output of **Label Recognition**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 awsrecog.py -l -p urban.jpg
Detection
-----
Label: City | Confidence: 99%
Label: Architecture | Confidence: 99%
Label: Building | Confidence: 99%
Label: Cityscape | Confidence: 99%
Label: Urban | Confidence: 99%
Label: Metropolis | Confidence: 99%
Label: Outdoors | Confidence: 98%
Label: Tower | Confidence: 97%
Label: Office Building | Confidence: 93%
Label: Aerial View | Confidence: 82%
Label: Neighborhood | Confidence: 64%
Label: Intersection | Confidence: 57%
Label: Road | Confidence: 57%
Label: High Rise | Confidence: 57%
Label: Downtown | Confidence: 57%
adharsh@adharsh:~/cits5503/lab9$ 

```

- The screenshot below shows the output of **Image Moderation**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 awsrecog.py -i -p beach.jpg
Detection
-----
Label: Non-Explicit Nudity of Intimate parts and Kissing | Confidence: 78%
Label: Non-Explicit Nudity | Confidence: 78%
Label: Bare Back | Confidence: 78%
adharsh@adharsh:~/cits5503/lab9$ 

```

- The screenshot below shows the output of **Facial Analysis**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 awsrecog.py -f -p faces.jpg
Face Number: 1
Detection
-----
Gender: Male | Confidence: 98%
Age Range (years): 21 to 27
Glasses: False | Confidence: 99%
-----
Face Number: 2
Detection
-----
Gender: Female | Confidence: 92%
Age Range (years): 18 to 22
Glasses: True | Confidence: 99%
-----
Face Number: 3
Detection
-----
Gender: Male | Confidence: 99%
Age Range (years): 23 to 29
Glasses: False | Confidence: 99%
-----
Face Number: 4
Detection
-----
Gender: Female | Confidence: 99%
Age Range (years): 15 to 21
Glasses: False | Confidence: 99%
-----
Face Number: 5
Detection
-----
Gender: Female | Confidence: 99%
Age Range (years): 18 to 22
Glasses: False | Confidence: 99%
adharsh@adharsh:~/cits5503/lab9$ 

```

- The screenshot below shows the output of **Extract Text**:

```

adharsh@adharsh:~/cits5503/lab9$ python3 awsrecog.py -e -p text.jpg
Detection
-----
Text detected: It was the best of | Confidence: 99%
Text detected: times, it was the worst | Confidence: 99%
Text detected: of times, it was the age | Confidence: 99%
Text detected: of wisdom, it was the | Confidence: 99%
Text detected: age of foolishness... | Confidence: 99%
Text detected: It | Confidence: 100%
Text detected: was | Confidence: 99%
Text detected: the | Confidence: 100%
Text detected: best | Confidence: 100%
Text detected: of | Confidence: 100%
Text detected: times, | Confidence: 100%
Text detected: it | Confidence: 100%
Text detected: was | Confidence: 100%
Text detected: the | Confidence: 100%
Text detected: worst | Confidence: 99%
Text detected: of | Confidence: 100%
Text detected: times, | Confidence: 100%
Text detected: it | Confidence: 99%
Text detected: was | Confidence: 99%
Text detected: the | Confidence: 100%
Text detected: age | Confidence: 100%
Text detected: of | Confidence: 99%
Text detected: wisdom, | Confidence: 99%
Text detected: it | Confidence: 100%
Text detected: was | Confidence: 99%
Text detected: the | Confidence: 100%
Text detected: age | Confidence: 100%
Text detected: of | Confidence: 100%
Text detected: foolishness... | Confidence: 98%
adharsh@adharsh:~/cits5503/lab9$ 

```

**All resources that were created for the lab (*S3 Bucket and the objects in it*) were deleted using the AWS Console after the lab completion.**

---