FIT 3162 Computer Science Project 2

User Guides

Automatic Identification of Neuropsychiatric Disorders from Brain Networks Using Deep Learning

Team:

MA_B_2

Members:

Choo Jun Yi

Khoo Teng Ian

Fong Zhong Kern

Supervisor:

A/Prof Ting Chee Ming

Table of Contents

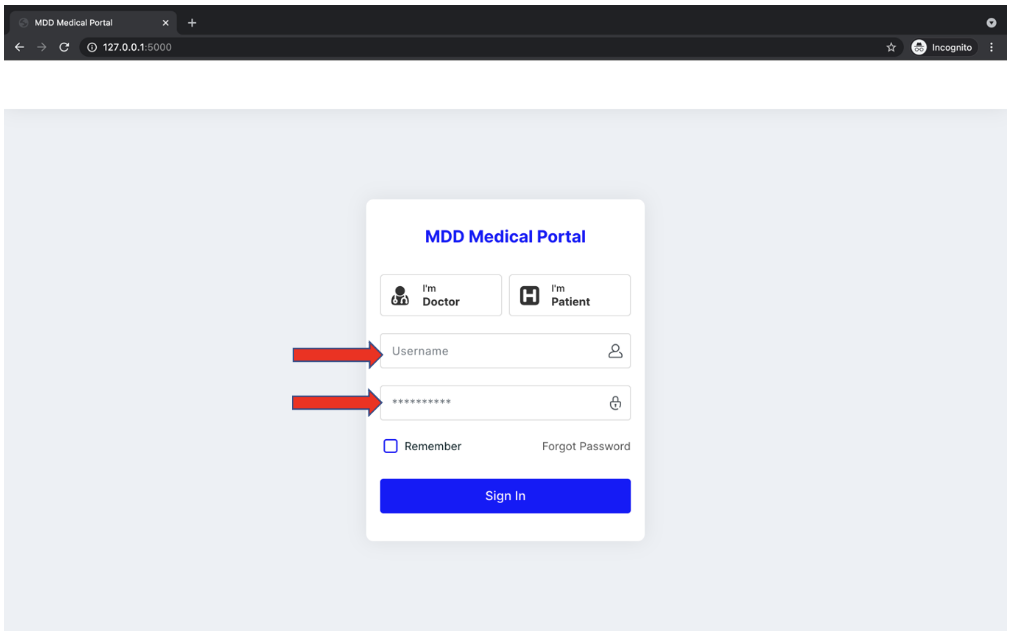
<u>End User Guide</u>

The users of our application will be psychiatrists who are responsible for diagnosing Major Disorder Depression (MDD) patients. This application aims to assist their diagnosis by using the state-of-the-art Deep Learning models to predict how likely a patient will suffer from MDD.

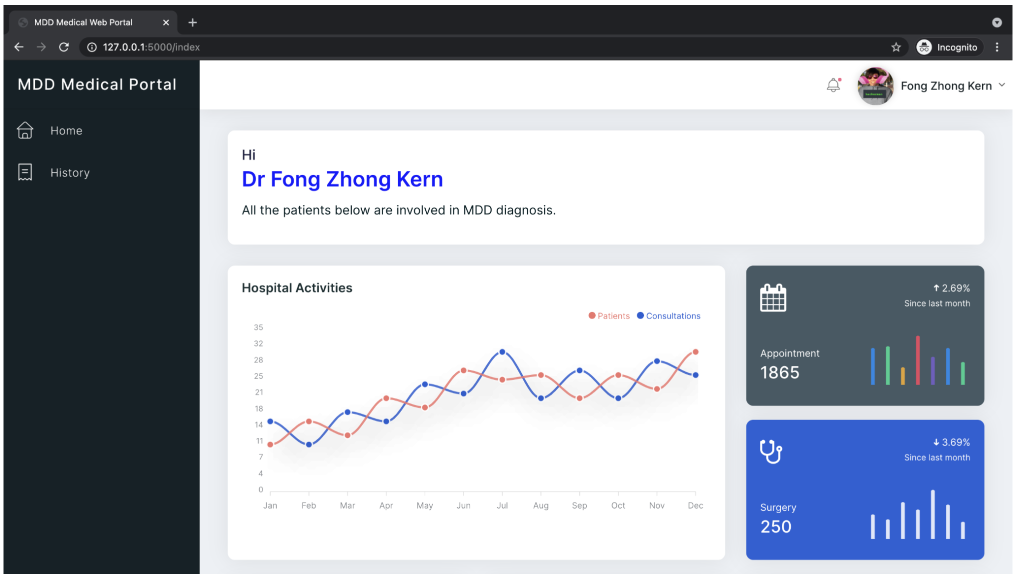
Instructions to use:

Step 1: Navigate to the website portal

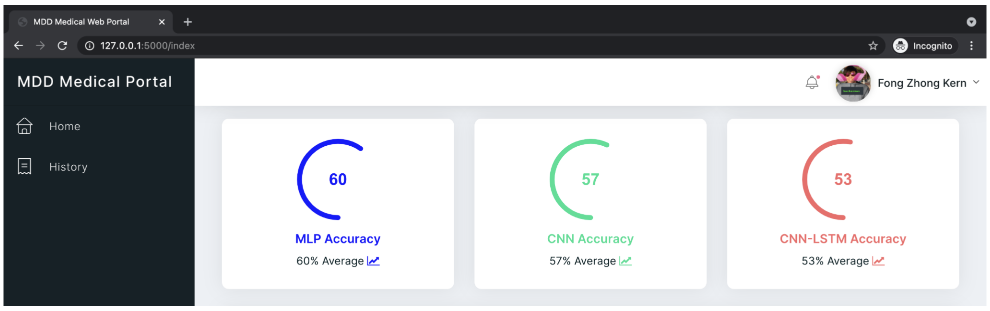
Step 2: Login to the portal using respective credentials



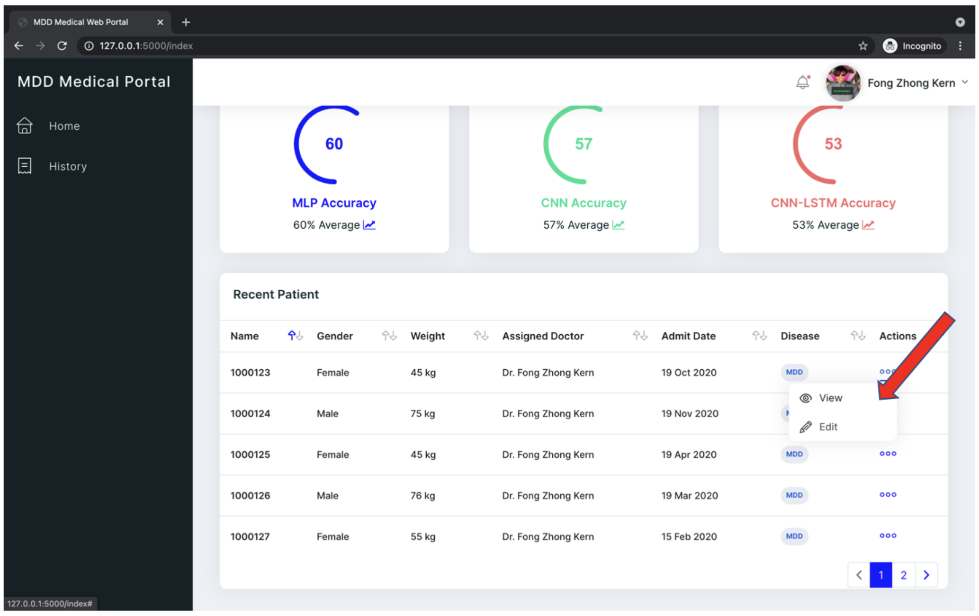
Step 3: Check out the daily reports and updates from the hospital


Hi
Dr Fong Zhong Kern
All the patients below are involved in MDD diagnosis.

Step 4: Check out the system updates on the current accuracy of each Deep Learning model. This is where each psychiatrist will evaluate how much they should trust the model



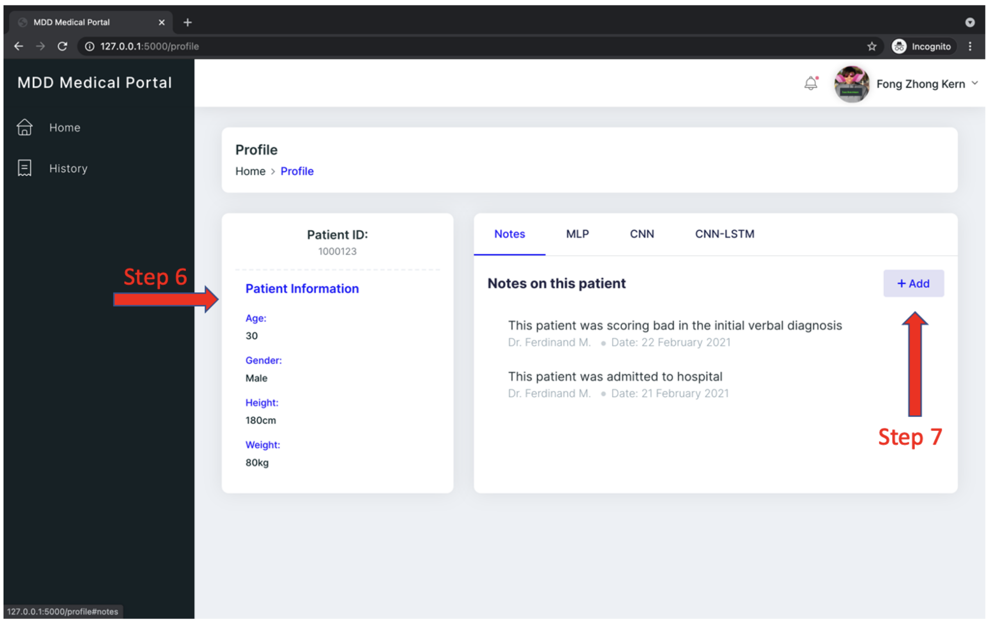
Step 5: Select the patient of interest and click 'View' button



Name	Gender	Weight	Assigned Doctor	Admit Date	Disease	Actions
1000123	Female	45 kg	Dr. Fong Zhong Kern	19 Oct 2020	MDD	ooo
1000124	Male	75 kg	Dr. Fong Zhong Kern	19 Nov 2020		
1000125	Female	45 kg	Dr. Fong Zhong Kern	19 Apr 2020	MDD	ooo
1000126	Male	76 kg	Dr. Fong Zhong Kern	19 Mar 2020	MDD	ooo
1000127	Female	55 kg	Dr. Fong Zhong Kern	15 Feb 2020	MDD	ooo

Step 6: Check the patient information

Step 7: Add any comments about the patients if there is any



Patient ID:
1000123

Patient Information

Age:
30

Gender:
Male

Height:
180cm

Weight:
80kg

Notes on this patient

This patient was scoring bad in the initial verbal diagnosis
Dr. Ferdinand M. • Date: 22 February 2021

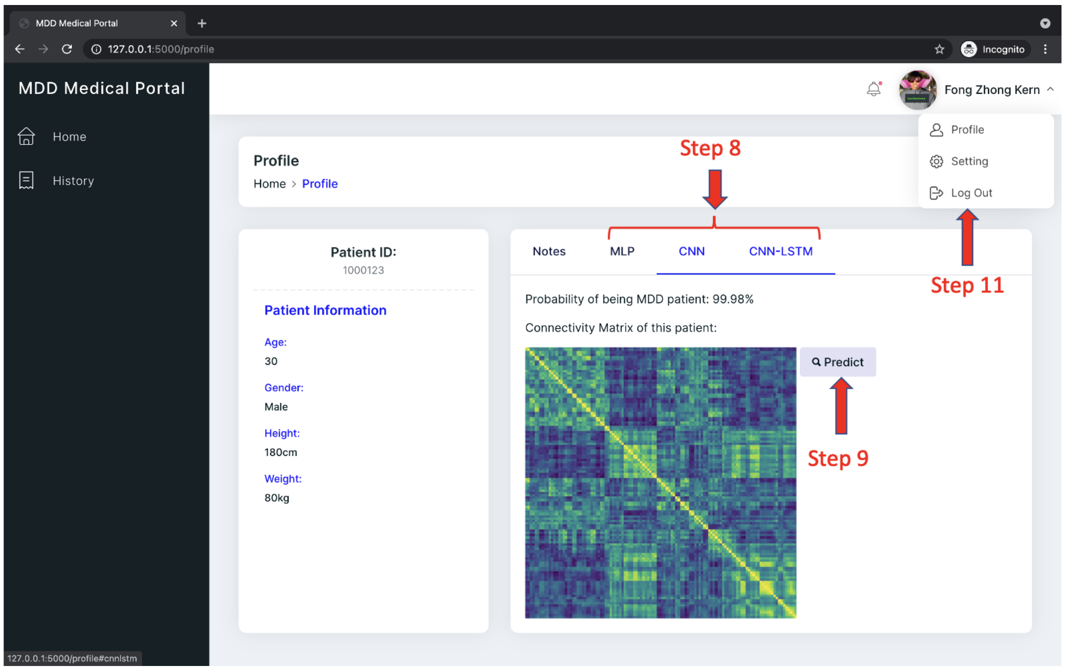
This patient was admitted to hospital
Dr. Ferdinand M. • Date: 21 February 2021

Step 8: Navigate to MLP, CNN, CNN-LSTM tabs to perform prediction using each models

Step 9: For CNN and CNN-LSTM, a connectivity matrix of the brain nodes will be plotted out for the reference of the psychiatrists

Step 10: Repeat step 6 to step 9 above on other patients

Step 11: Log out when the prediction is done


Profile

Home > Profile

Patient ID:
1000123

Patient Information

Age:
30

Gender:
Male

Height:
180cm

Weight:
80kg

Probability of being MDD patient: 99.98%

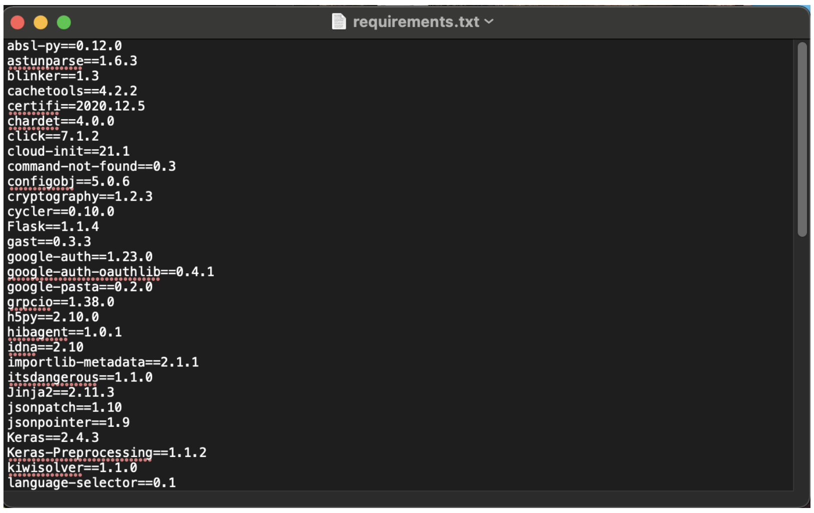
Connectivity Matrix of this patient:

Technical Guide

There are 2 ways to set up a MDD web application as shown in the figures above given that we already have the folder deployment. The first way is to host it using localhost and the second way is to host it on an Amazon ec2 instance. Note that if this web application were to be hosted on the localhost, only the owner of the computer is able to access the web portal. However, if we were to host it on an EC2 instance, then everyone on the Internet will be able to access the web application via an open IP address.

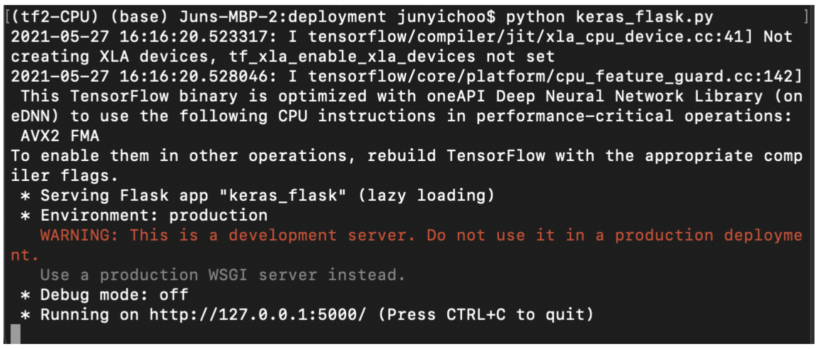
Hosting on localhost

1. The assumption made here is that you will have pip3 and python3 installed on your localhost. If you have not done so please refer to this link to install Python 3 and this link to install pip3.
2. Download the zip folder deployment.zip and unzip it
3. Navigate to the deployment folder in your terminal/command prompt and then enter the command `pip3 install -r requirements.txt`. The requirements.txt file contains the libraries shown below



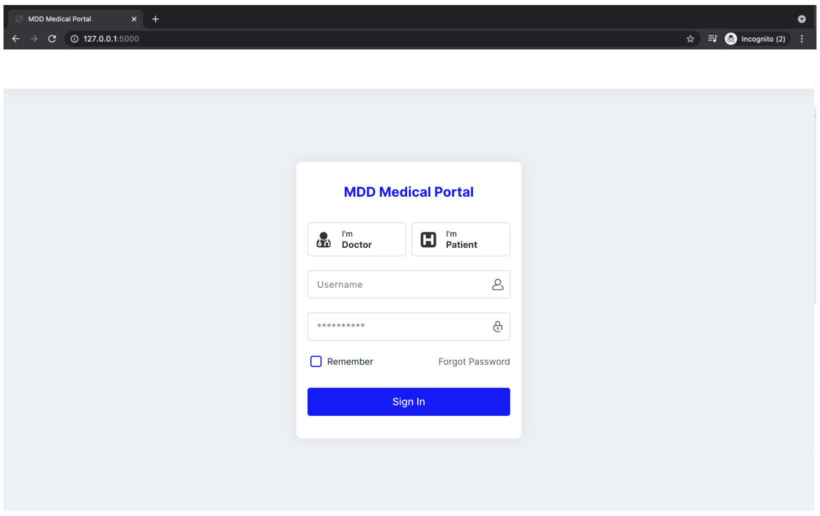
```
absl-py==0.12.0
astunparse==1.6.3
blinker==1.3
cachetools==4.2.2
certifi==2020.12.5
chardet==4.0.0
click==7.1.2
cloud-init==21.1
command-not-found==0.3
configobj==5.0.6
cryptography==1.2.3
cycler==0.10.0
Flask==1.1.4
gast==0.3.3
google-auth==1.23.0
google-auth-oauthlib==0.4.1
google-pasta==0.2.0
grpcio==1.38.0
h5py==2.10.0
hibagent==1.0.1
idna==2.10
importlib-metadata==2.1.1
itsdangerous==1.1.0
Jinja2==2.11.3
jsonpatch==1.10
jsonpointer==1.9
Keras==2.4.3
Keras-Preprocessing==1.1.2
kiwisolver==1.1.0
language-selector==0.1
```

4. Once the libraries are downloaded, you may start the flask server by typing in `python3 keras_flask.py` into your terminal.
5. This is the output you are expected to see



```
[(tf2-CPU) (base) Juns-MBP-2:deployment junyichoo$ python keras_flask.py
2021-05-27 16:16:20.523317: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not
creating XLA devices, tf_xla_enable_xla_devices not set
2021-05-27 16:16:20.528046: I tensorflow/core/platform/cpu_feature_guard.cc:142]
 This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (on
eDNN) to use the following CPU instructions in performance-critical operations:
 AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate comp
iler flags.
 * Serving Flask app "keras_flask" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

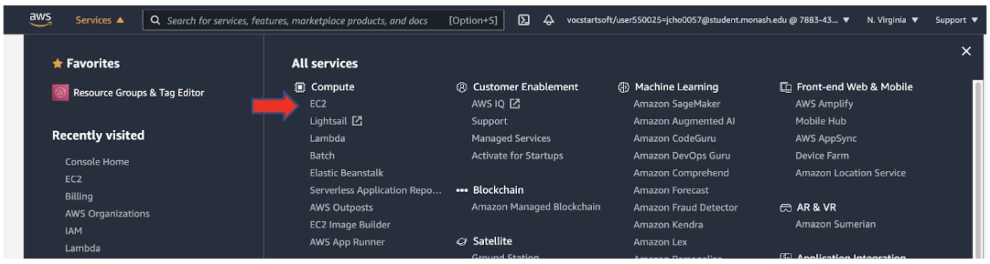
6. In your web browser, paste http://127.0.0.1:5000/ in the search bar and you will be served with the login page of the web application.



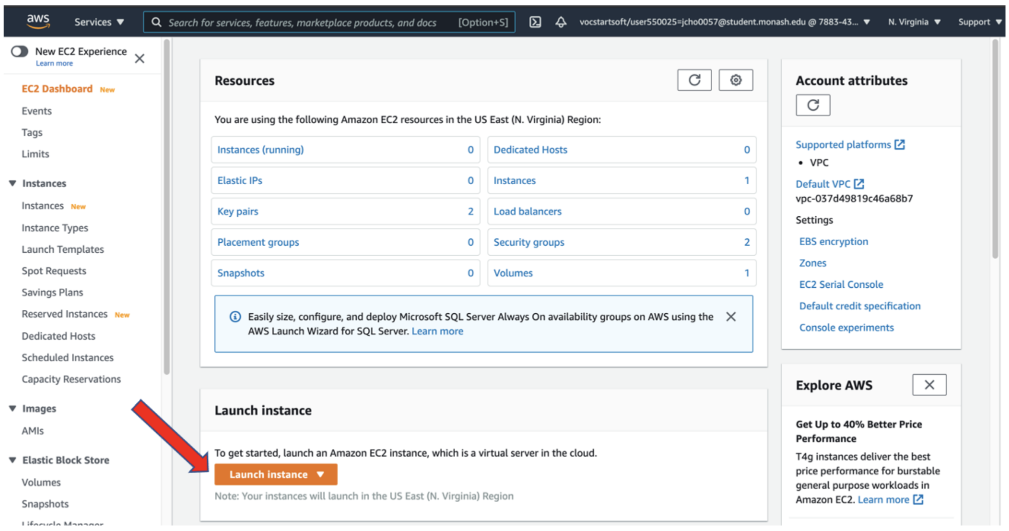
Hosting on Amazon EC2

Setting up an EC2 instance (Phan, 2020):

1. In the Amazon console, go to Services and select EC2 under Compute



2. Select Launch instance to start a new EC2 instance



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1
Key pairs	2	Load balancers	0
Placement groups	0	Security groups	2
Snapshots	0	Volumes	1

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Account attributes

Supported platforms
- VPC

Default VPC
vpc-037d49819c46a68b7

Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

Launch instance

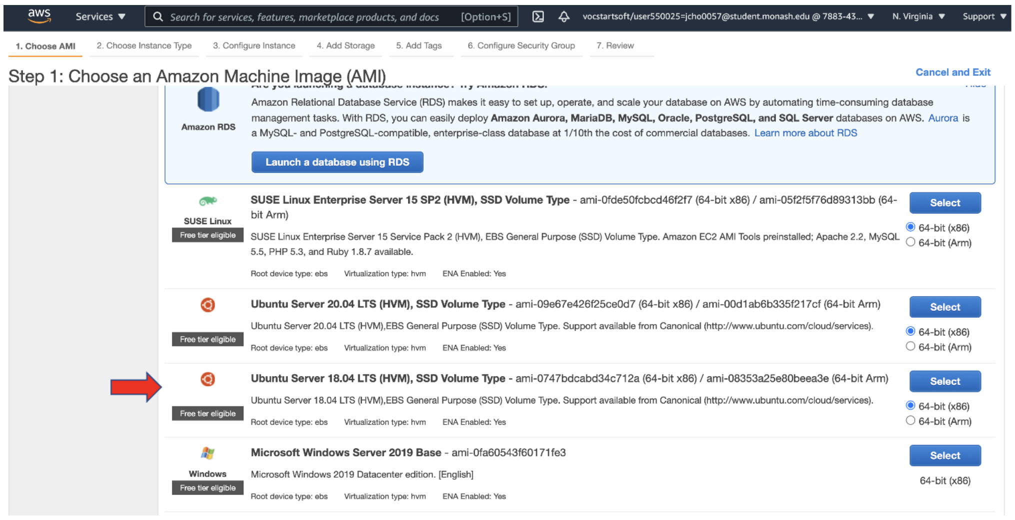
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

Explore AWS

Get Up to 40% Better Price Performance
T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2. Learn more

3. Select the Free Tier Instance, specifically, select the Ubuntu Server 18.04 LTS


aws
Services
Search for services, features, marketplace products, and docs [Option+S]
vocstartsoft/user550025=jcho0057@student.monash.edu @ 7883-43...
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Cancel and Exit
Amazon RDS
Launch a database using RDS
SUSE Linux
Free tier eligible
Windows
Select

Step 1: Choose an Amazon Machine Image (AMI)

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy **Amazon Aurora, MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server** databases on AWS. Aurora is a MySQL- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. Learn more about RDS

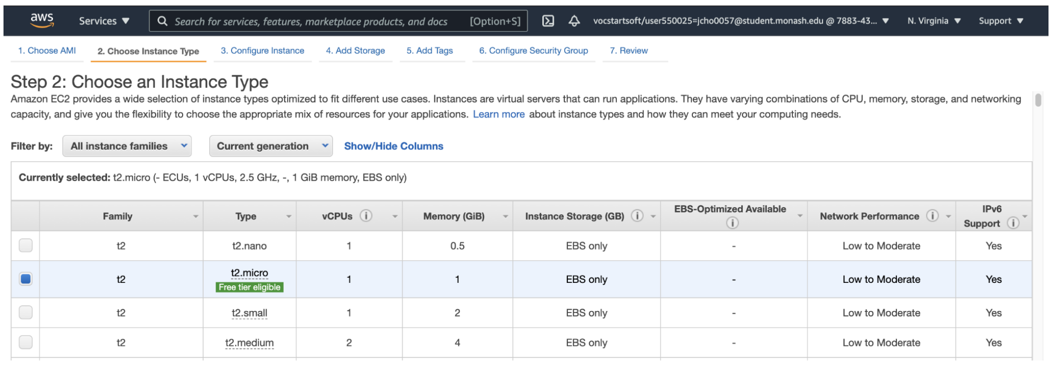
SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0fde50fcbcd46f2f7 (64-bit x86) / ami-05f2f5f76d89313bb (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
[✓ 64-bit (x86)] 64-bit (Arm)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7 (64-bit x86) / ami-00d1ab6b335f217cf (64-bit Arm)
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
[✓ 64-bit (x86)] 64-bit (Arm)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0747bdcabd34c712a (64-bit x86) / ami-08353a25e80beea3e (64-bit Arm)
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
[✓ 64-bit (x86)] 64-bit (Arm)

Microsoft Windows Server 2019 Base - ami-0fa60543f60171fe3
Microsoft Windows 2019 Datacenter edition. [English]
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

4. Select the instance that is labelled with 'Free Tier Eligible'. Family: General purpose, Type: t2.micro, vCPUs: 1, Memory: 1, Network Performance: Low to Moderate


aws
Services
Search for services, features, marketplace products, and docs [Option+S]
vocstartsoft/user550025=jcho0057@student.monash.edu @ 7883-43...
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Show/Hide Columns

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

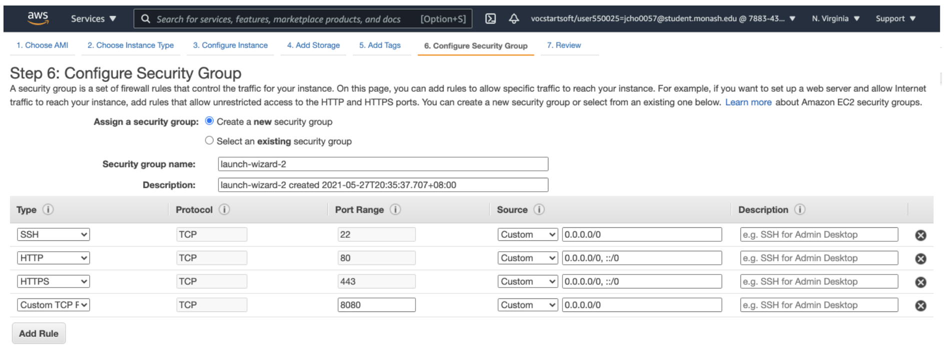
Filter by: All instance families | Current generation

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

5. Select Next: Configure Instance Details and Add Storage, Add Tag and Security Group

6. In *Step 6: Configure Security Group*, enter the relevant information as the screenshot below and press Next.


aws
Services
Search for services, features, marketplace products, and docs [Option+S]
vocstartsoft/user550025=jcho0057@student.monash.edu @ 7883-43...
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Add Rule

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

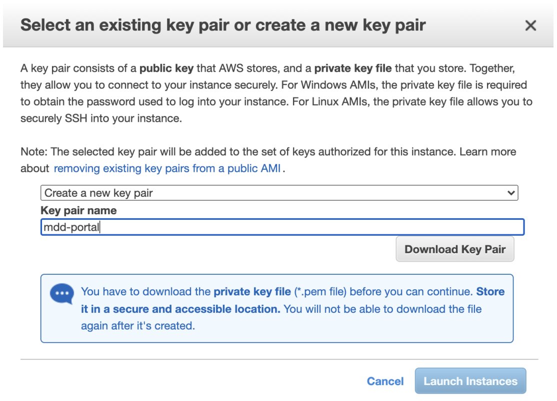
Assign a security group: [✓ Create a **new** security group] Select an **existing** security group

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2021-05-27T20:35:37.707+08:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

7. Select 'Create a new key pair' if you have not had one before, and key your key pair mdd-portal>. Next, download the Key Pair into a directory that you are most comfortable with.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

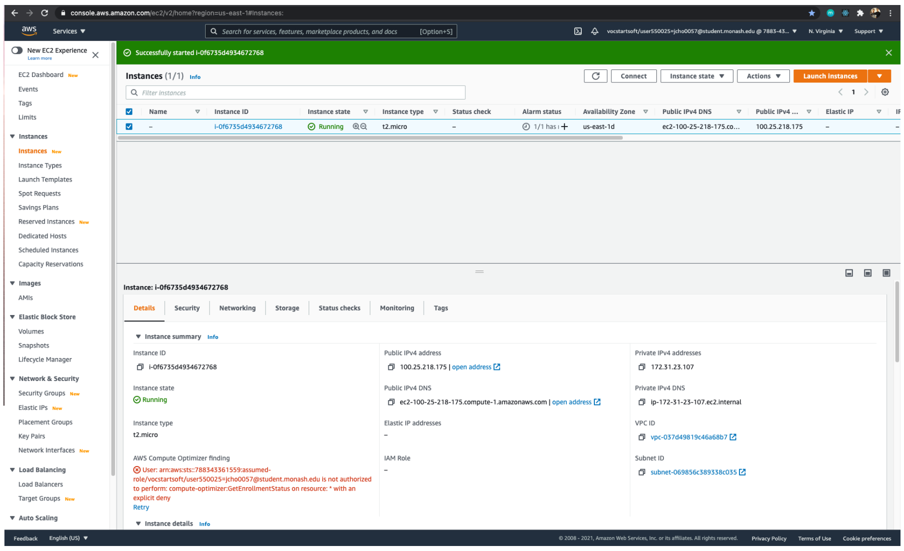
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name: mdd-portal

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

8. Launch the instance by right clicking your instance row. When the Instance State changes to 'Running' and and you can see a public IP, this means your instance is ready



Successfully started i-0f6735d4934672768

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
–	i-0f6735d4934672768	Running	t2.micro	–	1/1 has i	us-east-1d	ec2-100-25-218-175.co...	100.25.218.175	–

Instance: i-0f6735d4934672768

Instance summary Info

Instance ID i-0f6735d4934672768	Public IPv4 address 100.25.218.175 | open address	Private IPv4 addresses 172.31.23.107
Instance state Running	Public IPv4 DNS ec2-100-25-218-175.compute-1.amazonaws.com | open address	Private IPv4 DNS ip-172-31-23-107.ec2.internal
Instance type t2.micro	Elastic IP addresses –	VPC ID vpc-037d49819c46a68b7
AWS Compute Optimizer finding User: arn:aws:sts::788343361559:assumed-role/vocstartsoft/user550025=jcho0057@student.monash.edu is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny Retry	IAM Role –	Subnet ID subnet-069856c389338c035

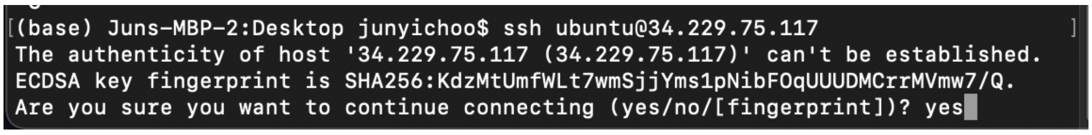
Instance details Info

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

<u>SSH into the EC2 instance:</u>

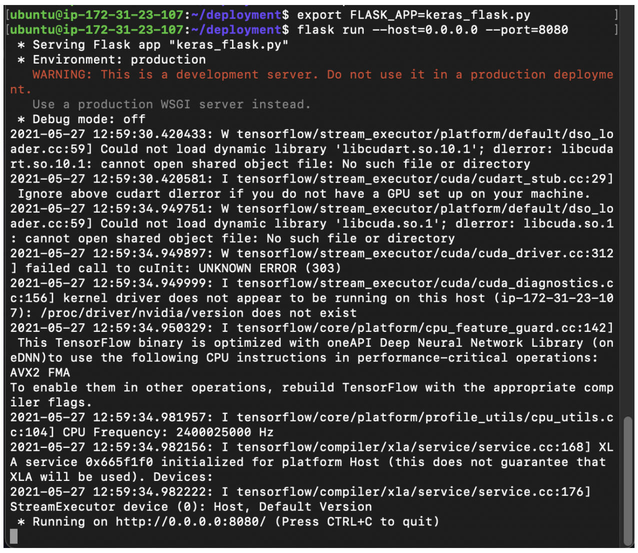
1. Locate the directory where you store your .pem/.cer file and enter `chmod 600 ./<pem file name>.pem.`

2. Next, in the same directory, enter `ssh-add ./<pem file name>.pem` in the command line to configure the ssh environment.
3. Now, we can ssh into the EC2 instance by typing `ssh ubuntu@<public IP address obtained above>` and you will see the output below. Enter `Yes` into the terminal and press enter.



```
[(base) Juns-MBP-2:Desktop junyichoo$ ssh ubuntu@34.229.75.117                  ]
The authenticity of host '34.229.75.117 (34.229.75.117)' can't be established.
ECDSA key fingerprint is SHA256:KdzMtUmfWLt7wmSjjYms1pNibFOqUUUDMCrrMVmw7/Q.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

4. You will now see the Ubuntu shell. Enter command `sudo apt update` and `sudo apt install python3 python3-pip tmux htop`.
5. Once installed, you are ready to transfer the deployment folder from localhost to the EC2 instance.
6. Go to the terminal in your local host and enter this command `sudo rsync -rv <Folder full path>/ ubuntu@<Public IP address>:/home/ubuntu`. This process will take around 3 minutes to complete due to the folder size.
7. Once the folder is transferred, navigate to the folder directory by entering `cd deployment/` and install the required python libraries using pip3 by using `pip3 install -r requirements.txt.`
8. Next, export our python file to a global FLASK_APP variable by entering `export FLASK_APP=keras_flask.py` in the terminal.
9. You may run the flask server now by entering `flask run --host=0.0.0.0 --port=8080` into the terminal. The expected output is shown below.



```
[ubuntu@ip-172-31-23-107:~/deployment$ export FLASK_APP=keras_flask.py         ]
[ubuntu@ip-172-31-23-107:~/deployment$ flask run --host=0.0.0.0 --port=8080    ]
 * Serving Flask app "keras_flask.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
2021-05-27 12:59:30.420433: W tensorflow/stream_executor/platform/default/dso_lo
ader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcuda
rt.so.10.1: cannot open shared object file: No such file or directory
2021-05-27 12:59:30.420581: I tensorflow/stream_executor/cuda/cudart_stub.cc:29]
 Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2021-05-27 12:59:34.949751: W tensorflow/stream_executor/platform/default/dso_lo
ader.cc:59] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1
: cannot open shared object file: No such file or directory
2021-05-27 12:59:34.949897: W tensorflow/stream_executor/cuda/cuda_driver.cc:312
] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-27 12:59:34.949999: I tensorflow/stream_executor/cuda/cuda_diagnostics.c
c:156] kernel driver does not appear to be running on this host (ip-172-31-23-10
7): /proc/driver/nvidia/version does not exist
2021-05-27 12:59:34.950329: I tensorflow/core/platform/cpu_feature_guard.cc:142]
 This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (on
eDNN)to use the following CPU instructions in performance-critical operations:
AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate comp
iler flags.
2021-05-27 12:59:34.981957: I tensorflow/core/platform/profile_utils/cpu_utils.c
c:104] CPU Frequency: 2400025000 Hz
2021-05-27 12:59:34.982156: I tensorflow/compiler/xla/service/service.cc:168] XL
A service 0x665f1f0 initialized for platform Host (this does not guarantee that
XLA will be used). Devices:
2021-05-27 12:59:34.982222: I tensorflow/compiler/xla/service/service.cc:176]
StreamExecutor device (0): Host, Default Version
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

10. Share the IP address of *http://<Public IP address>:8080/* with someone who wants to access the MDD web portal.

References

Phan, D. (2020). *How to Deploy a Flask App on AWS EC2 Instance*. Twilio Blog. https://www.twilio.com/blog/deploy-flask-python-app-aws.