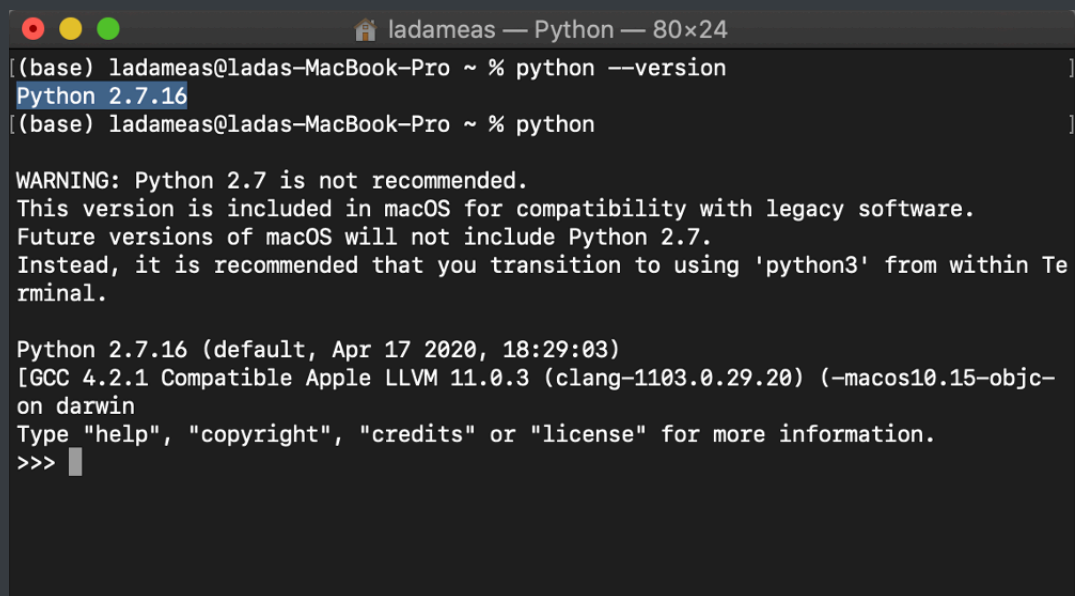


Setting up OpenCV and Python Environment on MacOS

By default, MacOS comes with Python version 2.X.X. To check:

1. Go to terminal, type:

```
python --version
```

A screenshot of a macOS terminal window. The title bar shows 'ladameas — Python — 80x24'. The terminal content shows the command 'python --version' being executed, resulting in 'Python 2.7.16'. Below this, a warning message is displayed: 'WARNING: Python 2.7 is not recommended. This version is included in macOS for compatibility with legacy software. Future versions of macOS will not include Python 2.7. Instead, it is recommended that you transition to using 'python3' from within Terminal.' The prompt then shows 'Python 2.7.16 (default, Apr 17 2020, 18:29:03) [GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc-on darwin)' and the prompt 'Type "help", "copyright", "credits" or "license" for more information.' followed by '>>>' and a cursor.

```
ladameas@ladas-MacBook-Pro ~ % python --version
Python 2.7.16
ladameas@ladas-MacBook-Pro ~ % python

WARNING: Python 2.7 is not recommended.
This version is included in macOS for compatibility with legacy software.
Future versions of macOS will not include Python 2.7.
Instead, it is recommended that you transition to using 'python3' from within Terminal.

Python 2.7.16 (default, Apr 17 2020, 18:29:03)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc-on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

2. To find python path directory, type:

```
which python
```

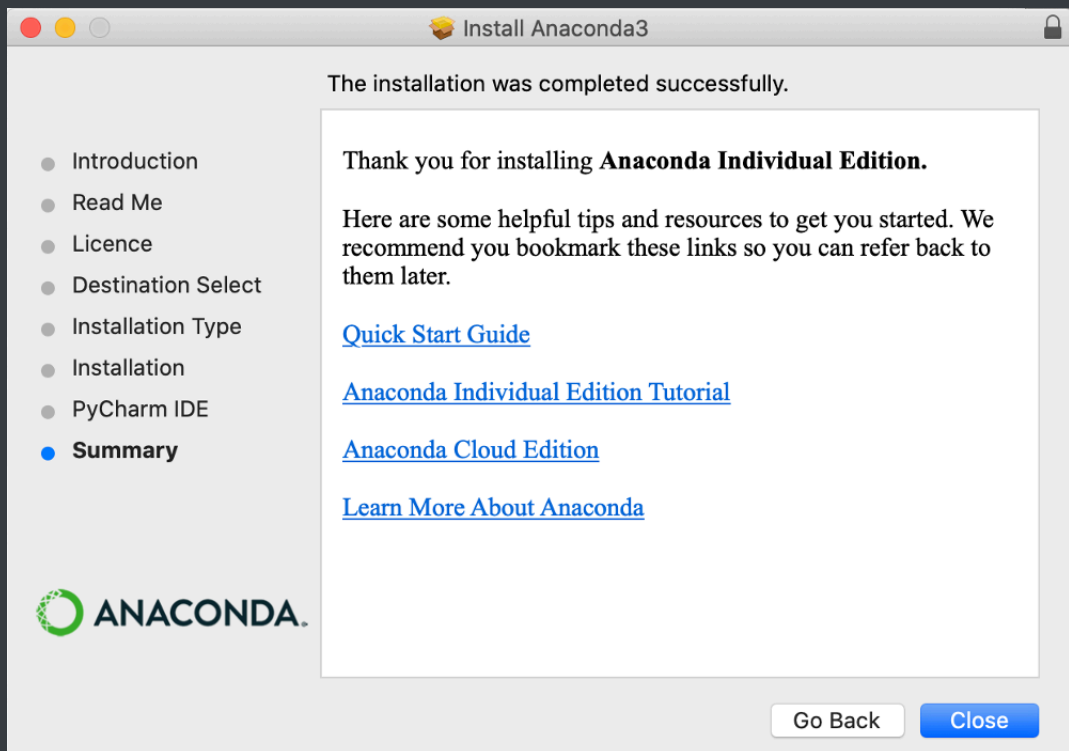
Download and Install Anaconda

In this section, we would like to show you how to install Python3 via Anaconda.

1. Go to <https://www.anaconda.com/products/individual>
2. Under the MacOS, click on "64-Bit Graphical Installer (462 MB)"

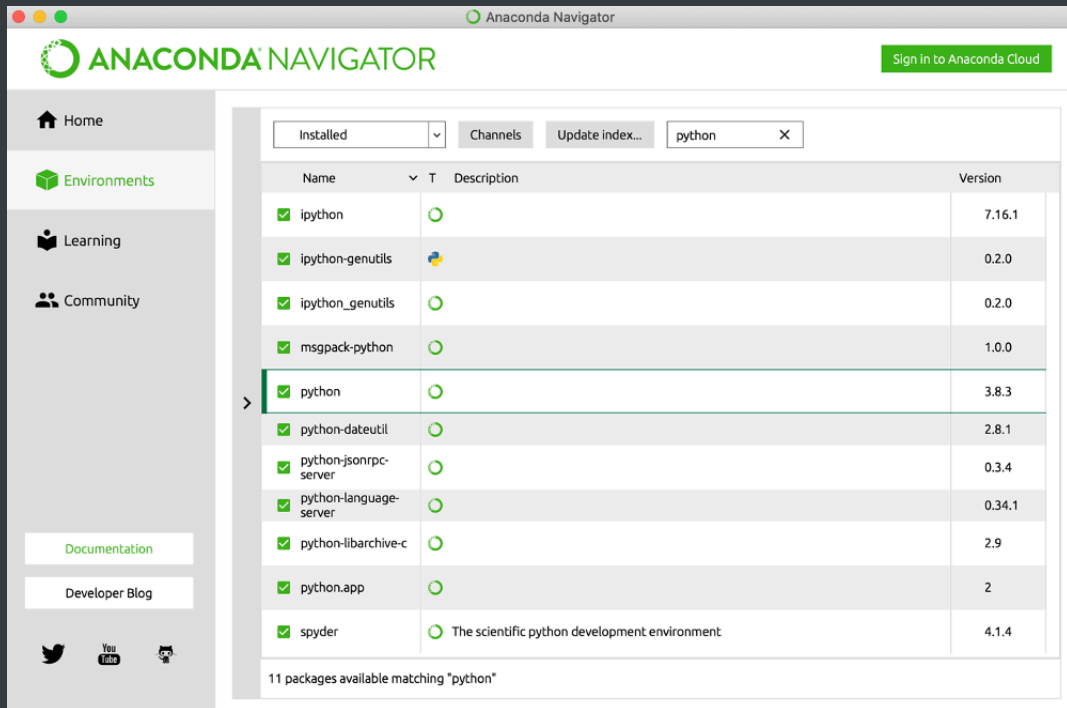


3. Follow the installation instruction to create an Anaconda environment



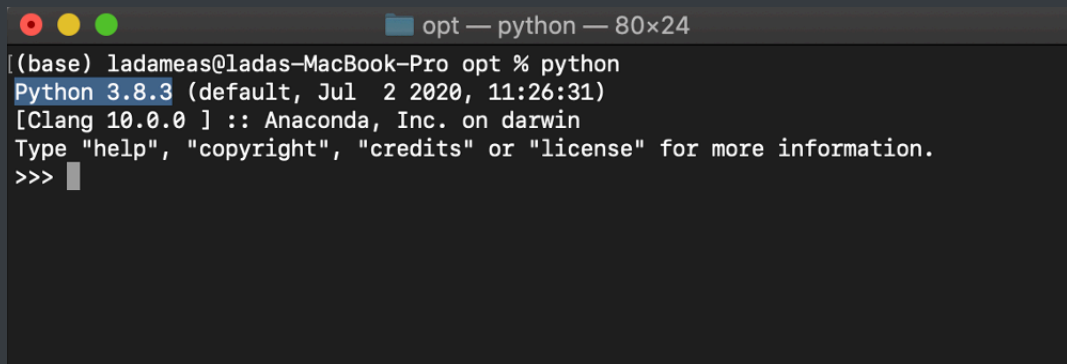
Verify Python version

1. Go to Anaconda-Navigator
2. On the left navigation bar, click on the "Enviroments" tab
3. In the search bar, Search Packages, type "python"
4. You will see python version 3.8.x is already installed



5. Check through the terminal (optional), type:

```
python
```



Install OpenCV

We install OpenCV on MacOS using pip.

1. Go to terminal, type:

```
sudo pip3 install opencv-python opencv-contrib-python
```

```
ladameas — -zsh — 80x24
(base) ladameas@ladas-MacBook-Pro ~ % sudo pip3 install opencv-python opencv-contrib-python
[Password:
WARNING: The directory '/Users/ladameas/Library/Caches/pip' or its parent directory is not owned or is not writable by the current user. The cache has been disabled. Check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting opencv-python
  Downloading opencv_python-4.4.0.44-cp38-cp38-macosx_10_13_x86_64.whl (52.3 MB)
    | 52.3 MB 20.4 MB/s
Requirement already satisfied: opencv-contrib-python in ./opt/anaconda3/lib/python3.8/site-packages (4.4.0.44)
Requirement already satisfied: numpy>=1.17.3 in ./opt/anaconda3/lib/python3.8/site-packages (from opencv-python) (1.18.5)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.4.0.44
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/Users/ladameas/opt/anaconda3/bin/python -m pip install --upgrade pip' command.
(base) ladameas@ladas-MacBook-Pro ~ %
```

2. To verify whether OpenCV has successfully installed

3. Go to terminal, type:

```
python3
```

```
>>> import cv2
```

```
ladameas — python3 — 80x24
(base) ladameas@ladas-MacBook-Pro ~ % python3
Python 3.8.3 (default, Jul 2 2020, 11:26:31)
[Clang 10.0.0] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>>
```

4. To check OpenCV version, type:

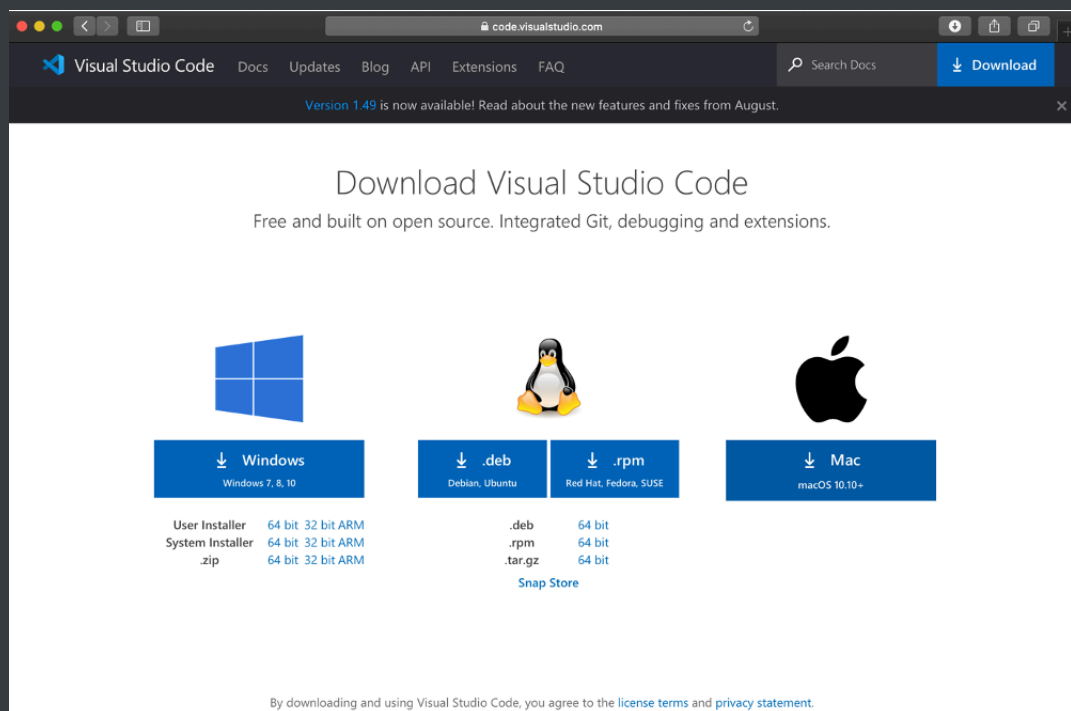
```
cv2.__version__
```

```
Testing Python — python — 80x24
(base) ladameas@ladas-MacBook-Pro Testing Python % python
Python 3.8.3 (default, Jul 2 2020, 11:26:31)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.4.0'
>>>
```

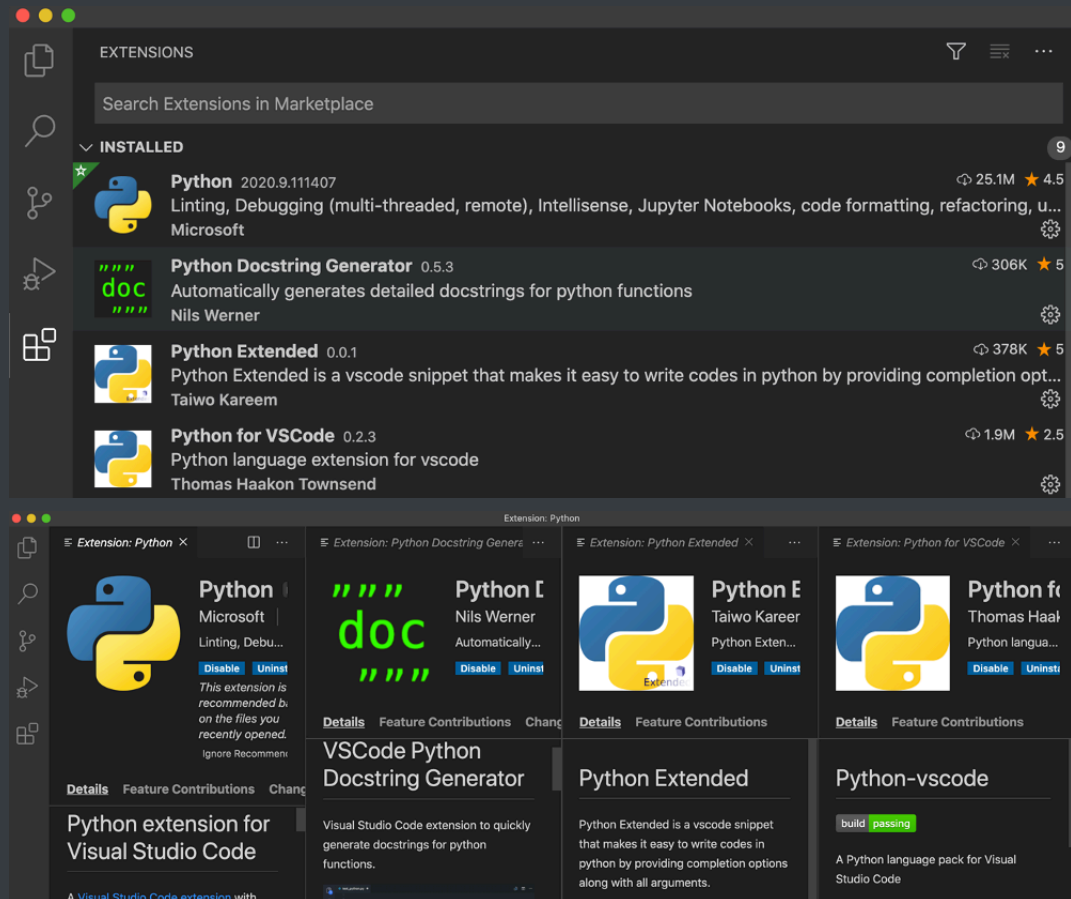
Download Visual Studio Code (VS Code)

In this section, we will show how to download Visual Studio, a code editor used to write OpenCv and Python programming language.

1. Go to <https://code.visualstudio.com/download>
2. Click on Mac (macOS 10.10+) to download



3. After successfully downloading, go to Visual Studio Code
4. On the left navigation bar, click on "Extension"
5. Search and enable four essential packages including
 - (1) Python, (2) Python Docstring Generator, (3) Python Extended, (4) Python for VSCode



How to read and write an image using OpenCV

How to read an image

1. Import OpenCV package

```
import cv2
```

2. Read an image

```
variable_name = cv2.imread('image relative path')
```

3. Example

```
import cv2

# Read an image
# img, a variable to store a given image
img = cv2.imread('/Users/ladameas/Desktop/Testing Python/image_1.png')
```

How to write an image

1. Write an image

```
cv2.imwrite('Output Image Filepath', variable_name)
```

2. Example

```
import cv2

# Read an image
# " img ", a variable to store a given image
img = cv2.imread('/Users/ladameas/Desktop/Testing Python/image_1.png')

# Write an image
# " /Users/ladameas/Desktop/Testing Python/ ", a relative path for image to be stored
# " Original_Image_1.png ", a name given to output image`
cv2.imwrite('/Users/ladameas/Desktop/Testing Python/Original_Image_1.png',img)
```

How to show an image

1. Show an image

```
cv2.imshow('Output Image Filename', variable_name)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

2. Example

```

import cv2

# Read an image
# " img ", a variable to store a given image
img = cv2.imread('/Users/ladameas/Desktop/Testing Python/image_1.png')

# Write an image
# " /Users/ladameas/Desktop/Testing Python/ ", a relative path for image to be stored
# " Original_Image_1.png ", a name given to output image`
cv2.imwrite('/Users/ladameas/Desktop/Testing Python/Original_Image_1.png',img)

# Show an image
# " Original_Image_1 ", a name given to an image when displaying
cv2.imshow('Original_Image_1',img)

# Wait for a keyborad event, to be pressed
cv2.waitKey(0)
# Destroy all created windows
cv2.destroyAllWindows()

```

How to resize an image using IMUTILS

1. To install "imutils", go to terminal, type:

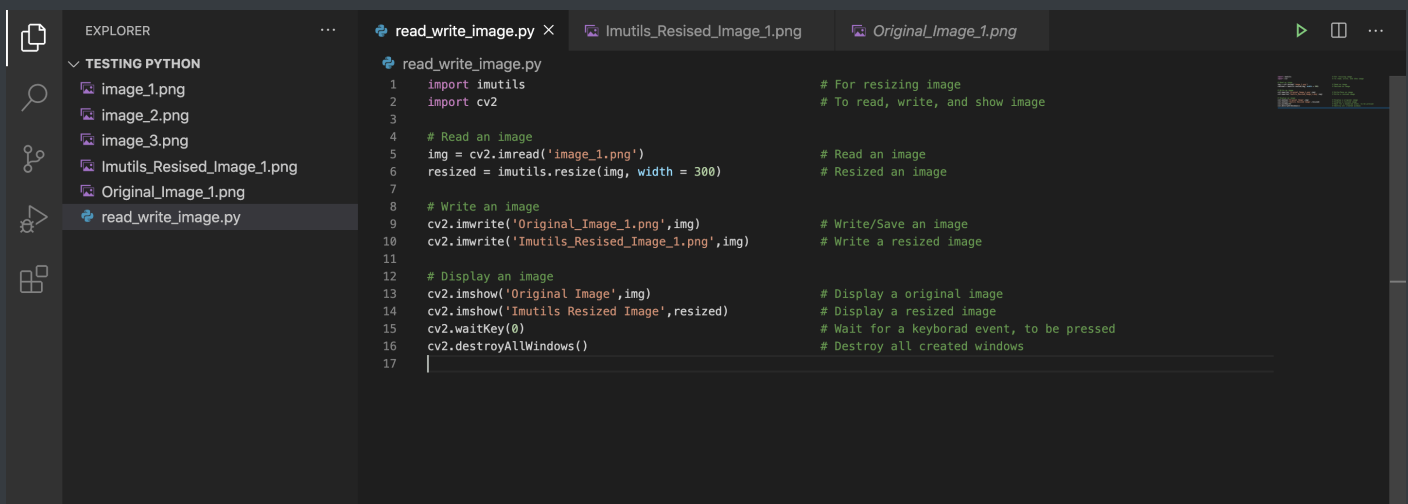
```
pip install imutils
```

2. To use "imutils"

```
import imutils
```

```
resized(variable_name) = imutils.resize(variable_name, width = XXX)
```

Example



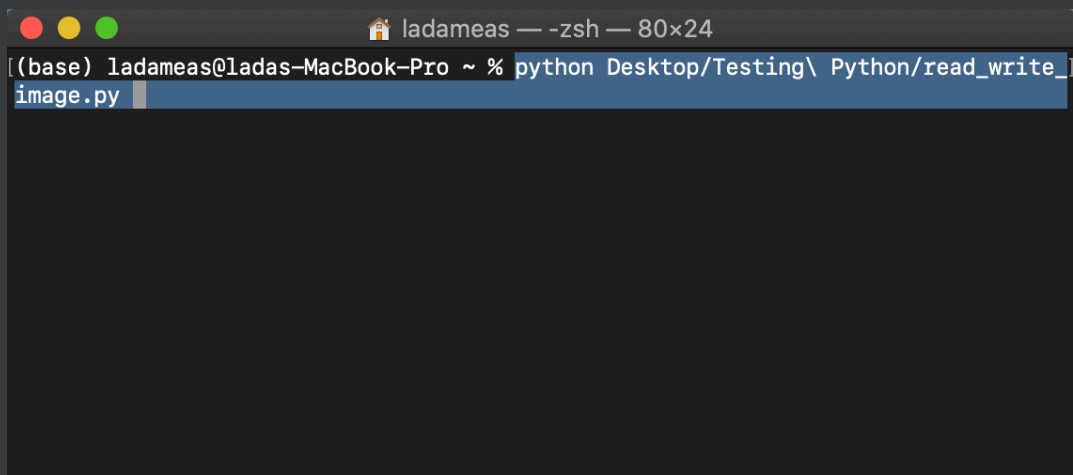
How to execute a python file using command line

1. Go to terminal
2. Go the relative path directory where python file is in, type:

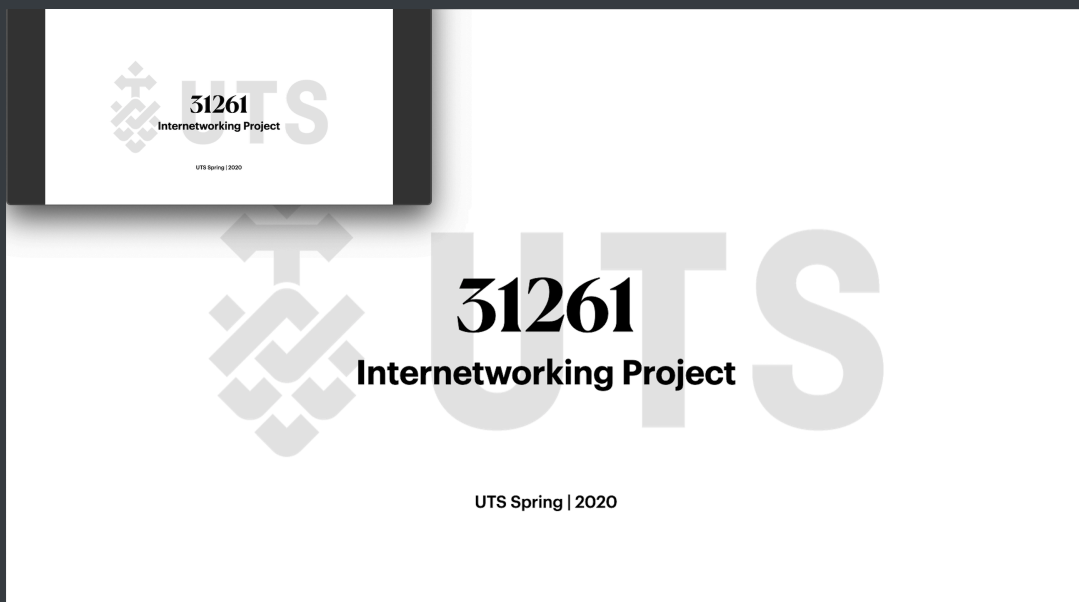
```
python + <python path of file>
```

For example:

```
python Desktop/Testing Python/read_write_imgae.py
```



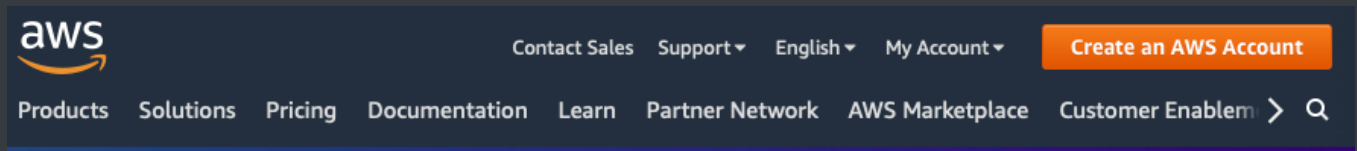
Example of display image:



Deploying a Flask Application on Amazon Web Service (AWS) EC2 Instance

Create an AWS user account

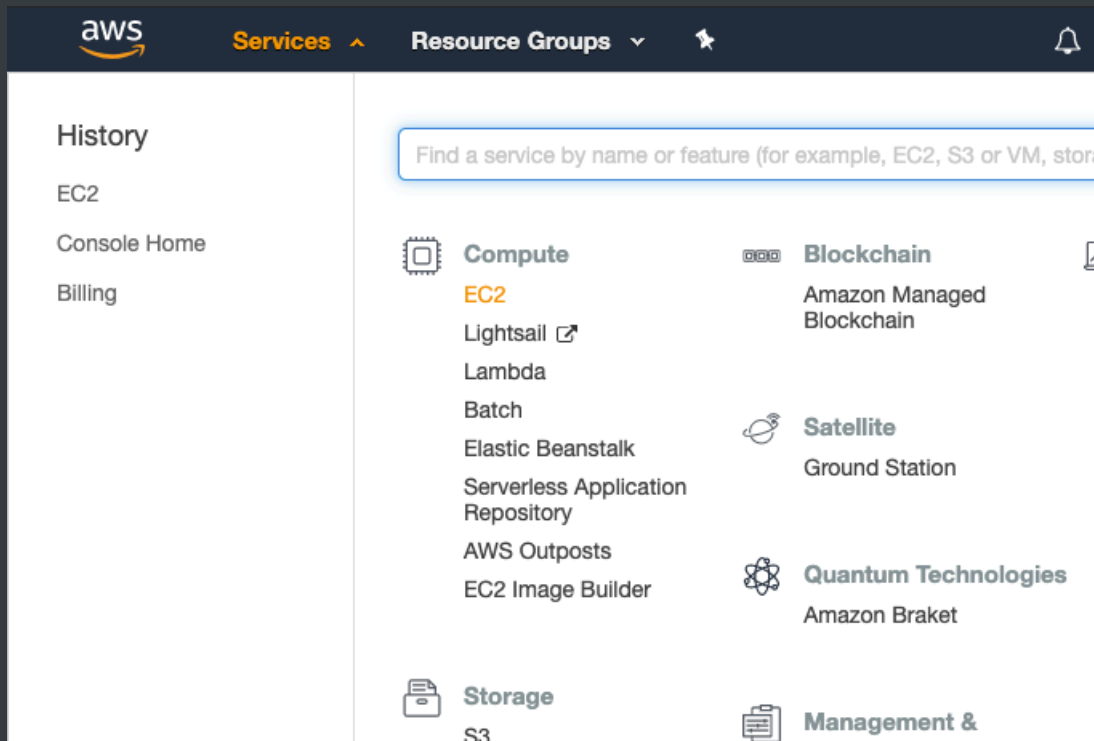
1. Go to <https://aws.amazon.com>, select "Create an AWS Account" and continue to create an account



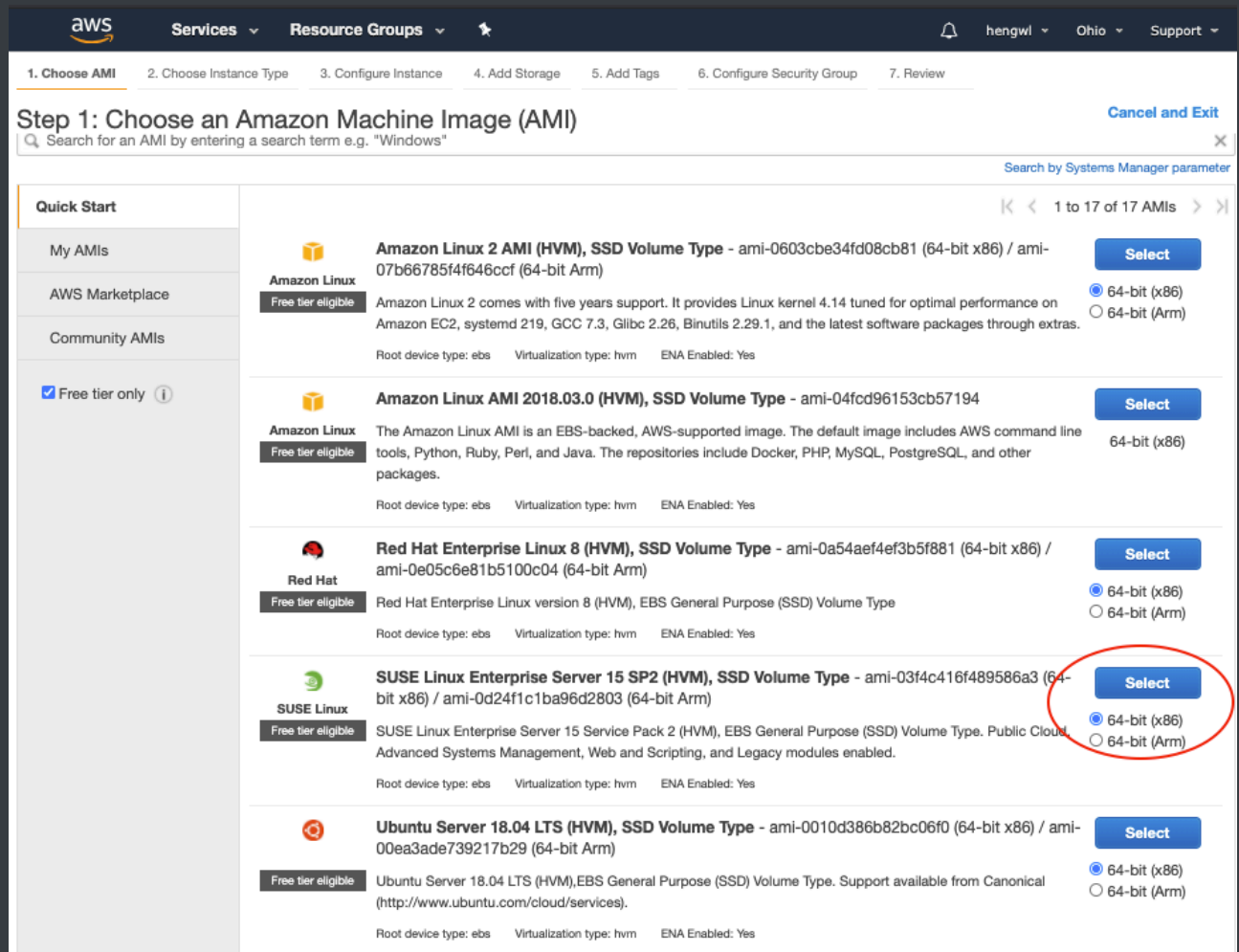
2. Upon logging back into your account, select login option "Root user", enter you email, password and proceed with signing in.

A screenshot of the AWS 'Sign in' page. At the top is the AWS logo. Below it is the heading 'Sign in'. There are two radio button options: 'Root user' (selected) and 'IAM user'. The 'Root user' option includes the text 'Account owner that performs tasks requiring unrestricted access. Learn more'. The 'IAM user' option includes the text 'User within an account that performs daily tasks. Learn more'. Below these options is a section titled 'Root user email address' with a text input field containing 'username@example.com'. A blue 'Next' button is positioned below the email field. At the bottom, there is a link 'New to AWS?' and a button labeled 'Create a new AWS account'.

3. Click on the **Services** tab, under **Compute**, select **EC2** and **Launch Instance**.



4. Under **Quick Start**, check "**Free tier only**" checkbox and select **Ubuntu Server**.



5. Ensure the select type is indicated with "Free tier eligible", click "Next: Configure Instance Details"

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 types** **Current generation** [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

4. Continue by click "Add Storage". Select the amount of storage required by your server. Free tier customer can get up to 30GB of storage. **Caution:** If you application exceed the free BG allocation, you will be charged. After indicating your storage requirement, click "Next: Add Tags"

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

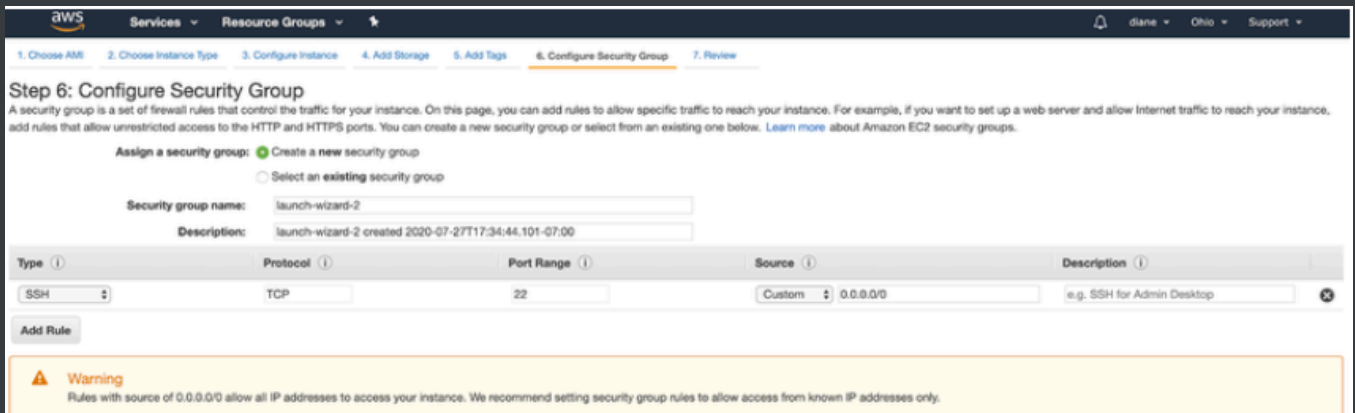
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0ed1b3d50e9f4ec5e	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

4. Tag are used to categorise your AWS resources for different use cases, making it easier to keep track of resources. It might not be necessary if you only plan on using AWS once for this specific application. Click "Configure Security Group" to move on.

5. Configuring security group is the last step before launching the AWS EC2 Instance. Set you security group name and description. Also, you have to add the necessary protocols and ports for your application. Such as port 80 for HTTP, Port 443 for HTTPS or if your flask application is running on custom port of 5000, add "Custom TCP Rule", port range of 5000, Source: 0.0.0.0/0 (indicating all address with port 5000). Once done, click "Review and Launch"



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

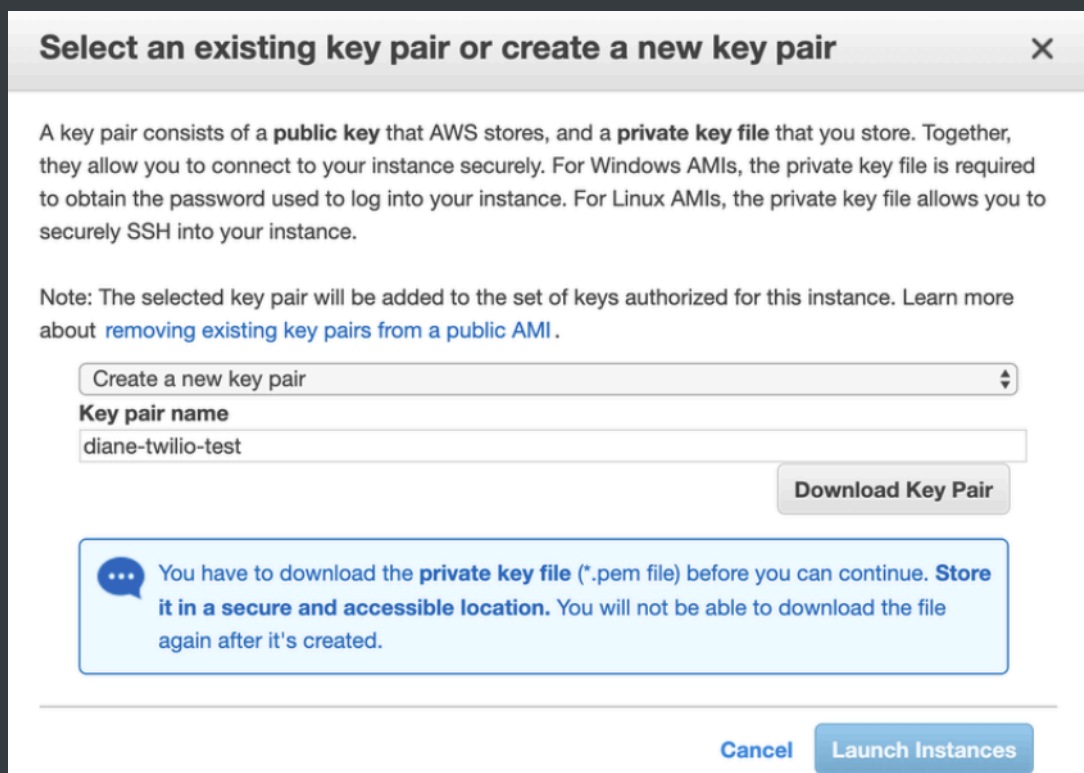
Description: launch-wizard-2 created 2020-07-27T17:34:44.101-07:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

6. After your review your settings, click "Launch" and you will be prompted a pop-up window asking you to select/ create a key pair. Select **Create a new key pair**, give your key pair a name. Click "**Download Key Pair**", save this private key file **.pem** to a secure location. That is the only way you can access your web application. After downloading your private key file, click "Launch Instance".



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

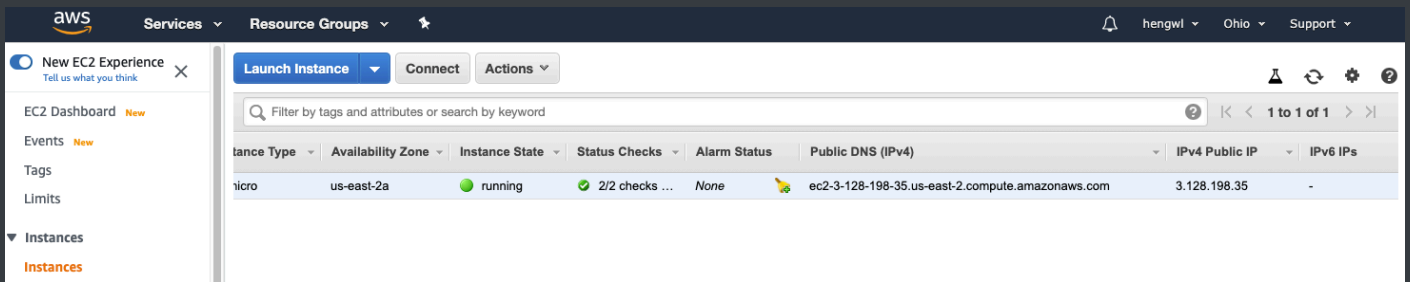
Key pair name
diene-twilio-test

[Download Key Pair](#)

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.

[Cancel](#) [Launch Instances](#)

10. Once instance launch, you will be able to retrieve your public IP address under **IPv4 Public IP** header when the Instance State shows "running".



SSH into your AWS EC2 Instance

1. Open your putty or terminal, locate the directory with the **.pem** file. Change the permission with **R** and **W**.

```
chmod 600 ./<YOUR_PEM_NAME>.pem
```

2. Next, set up the SSH environment by typing

```
ssh-add ./<YOUR_PEM_NAME>.pem
```

You will see the following response text in your terminal:

```
1 Identity added: ./diane-twilio-test.pem (./diane-twilio-test.pem)
```

3. Retrieve your public IP address from the EC2 Instance dashboard, and type

```
ssh ubuntu@<PUBLIC_IP_ADDRESS>
```

Respond with "yes", you will notice that your terminal shell is now your Ubuntu shell. It will look something like

```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.3.0-1032-aws x86_64)

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

System information as of Tue Sep  8 12:19:08 UTC 2020

System load:  0.0                       Processes:            100
Usage of /:   16.0% of 23.20GB          Users logged in:     1
Memory usage: 46%                       IP address for eth0: 172.31.8.130
Swap usage:   0%

* Kubernetes 1.19 is out! Get it in one command with:

    sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

* Canonical Livepatch is available for installation.
  - Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch

26 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Tue Sep  8 04:35:55 2020 from 218.212.113.224
ubuntu@ip-172-31-8-130:~$
```

Installing the required libraries and application

1. In AWS EC2 Instance, Ubuntu

```
$ sudo apt update
$ sudo apt install python3 python3-pip tmux htop flask
```

2. Make a directory of the application you wish to deploy (optional) in AWS EC2 Instance, Ubuntu.

```
$ mkdir deployedapp
```

3. Your local machine, inside your application that your wish to transfer

```
$ pip freeze > requirement.txt
```

4. Copy the full path of your application in your local machine **<FULL_PATH>**

5. In. AWS EC2 Instance, Ubuntu

```
$ sudo rsync -rv <FULL_PATH>/  
ubuntu@<PUBLIC_IP_ADDRESS>:~/home/ubuntu/deployedapp
```

This process will take a few minutes depending on the size of your program.

Deploying your application on AWS EC2 Instance

1. Using **tmux** command to create a new session with any name. This is where your application will run live.

```
$ tmux new -s <NAME_OF_SESSION>
```

2. Once inside your session, install the requirements of your application ie. the libraries required

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

3. If your application runs inside a virtual environment, you may install **venv** and activate it.

```
pip3 install venv
```

```
$ source venv/bin/activate
```

4. Run your application.
5. To detach from the session, press **CTRL B** follow by **D** and you will see the following message. And **CTRL D** will kill the session

```
[detached (from session Server)]  
ubuntu@ip-172-31-8-130:~$ █
```

6. To resume an active session,

```
$ tmux attach -t <SESSION_NAME>
```

7. To list all active session,

```
$ tmux ls
```


Flask server setting

```
from flask import request, Flask

@app.route("/", methods="POST")
def your_app_name():
    ...

if __name__ == "__main__":
    app.run("0.0.0.0", port=<YOUR_APP_PORT>) #an example port number could be 5000
```

Flask client setting

```
import requests

if __name__ == "__main__":
    url = "<YOUR_PUBLIC_IP_ADDRESS>:<YOUR_APP_PORT_NUMBER>"
    ...
    ...

    variable = requests.post(url, ...) #files, data etc.
```