

Remote Environment Setting

- Jupyter notebook Remote environment setting.
- Pure python mode environment setting.
- Kaggle auto submission.

We provide two working environment settings. The first one is jupyter notebook based. And the second one is python source code based. We recommend using Jupyter Notebook Remote Server + AWS EC2 to do the homeworks because it's easy to debug and interactively understand your code.

1) Jupyter notebook + AWS EC2.

- Download the key pair of your EC2, `deeplearning.pem`.
- Run the following code to set the password so that your local machine could access it using the password.

```
jupyter notebook password
```
- On EC2, run the following code to open a jupyter notebook server listening at port 8889. If you close the terminal, then this server would be killed. (trick: You can add nohup to prefix this command line code to let it run in the background. So the server would not be terminated if you close the terminal.)

```
(nohup) jupyter notebook --no-browser --port=8889
```
- On local machine, run the following command to connect 8888 port number on your machine to 8889 port on remote machine.

```
ssh -i deeplearning.pem -N -f -L localhost:8888:localhost:8889 ubuntu@ec2instance
```
- Open your browser and type in "localhost:8888", you can then input the password and access the remote jupyter server like you running a jupyter notebook on a local machine.
- Trick:** Sometimes you could close the terminal and run your model overnight. The second day you wake up and connect to the server. You will find that the output disappears. This happens because each terminal is a process and when you close it, it would be killed and the next time you connect to it, it's a new process. The best way to keep track of your output is to using logging package in python. (<https://docs.python.org/3/howto/logging-cookbook.html>). You can add two handlers so that your output would be printed and also be written to the log file.

2) Pure python mode + AWS EC2

One problem of coding on a remote machine is that you cannot use user-friendly IDE or text edit like you are coding on your macbook. SFTP could help you. SFTP is a file transmission protocol that could automatically synchronize your code with the remote machine. We recommend that you use VScode or Pycharm because it provides such plugins.

- If you are using VScode, you should install SFTP plugin first
 - Type shift+cmd+p to see the pop-up window and choose "SFTP-config".
 - In the configuration file, input the following code

```

{
  "name": "Anything You Like",
  "host": "ec2-instance-public-ip",
  "protocol": "sftp",
  "port": 22,
  "username": "ubuntu",
  "privateKeyPath": "deeplearning.pem",
  "remotePath": "/the-path-of-your-code-on-remote-machine",
  "uploadOnSave": true
}

```

- iii) Now you can edit your code on your local machine and it would automatically populate to the remote machine.
- b) If you are using Pycharm.
 - i) Preference -> Deployment -> Add, choose SFTP as the option. Configure it just like what you need to do if you are using VScode.

3) Kaggle

- 1) Download your kaggle.json file at Kaggle My Profile page.
- 2) Install Kaggle package


```
pip install kaggle
```
- 3) Copy your kaggle.json to .kaggle/ folder so that kaggle package could know who you are.


```
scp -i deep_learning.pem ./kaggle.json ubuntu@instance:~/.kaggle/
```
- 4) You could easily download the data using the following command. `competition-name` could be found on the 'Data' section in the competition main webpage.


```
kaggle competitions download -c competition-name
```
- 5) You can submit using the following code


```
kaggle competitions submit -c competition-name -f your-submission-path -m "Message"
```