# DSC 204A: Scalable Data Systems

Programming Assignment 0

Released: 10 April 2023, Due: 25 April 2023

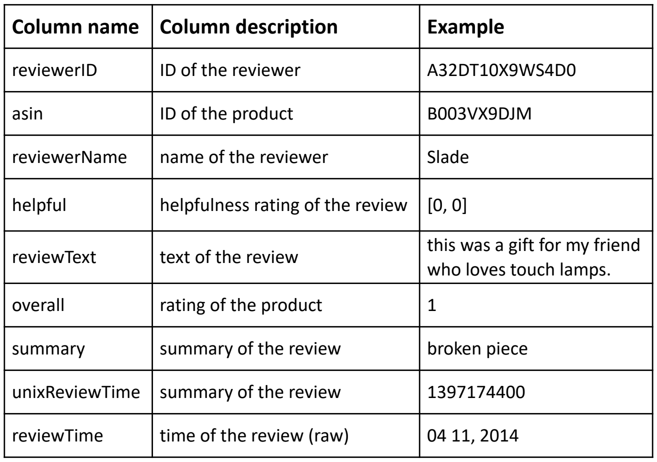
**VERY IMPORTANT: Download your progress to your local machine at regular intervals and terminate your instance when you decide to pause working. You have only $50 for both PA0 and PA1 and so DO NOT leave instances running. If you terminate without downloading, you WILL LOSE all your work. Every time you start a new instance, you must download the dataset from S3 to your instance. Also, start only AWS Spot Instances and NOT On-Demand instances.**

## 1 Introduction

The goal of this programming assignment is to get you comfortable with datasets that do not fit in the RAM of a single machine and hence are not suitable for analysis using packages like Pandas or NumPy. In PA0 and PA1 you will be using the Dask library to explore secondary storage aware data access on a single machine. In this assignment, you will be learn to setup Dask on AWS and compute several descriptive statistics about the data to build intuitions for feature engineering for the final assignment.

## 2 Dataset Description

You are provided with the Amazon Reviews dataset with the *reviews* table as CSV file. The schema is provided in Table 1.



| Column name | Column description | Example |
|---|---|---|
| reviewerID | ID of the reviewer | A32DT10X9WS4D0 |
| asin | ID of the product | B003VX9DJM |
| reviewerName | name of the reviewer | Slade |
| helpful | helpfulness rating of the review | [0, 0] |
| reviewText | text of the review | this was a gift for my friend who loves touch lamps. |
| overall | rating of the product | 1 |
| summary | summary of the review | broken piece |
| unixReviewTime | summary of the review | 1397174400 |
| reviewTime | time of the review (raw) | 04 11, 2014 |

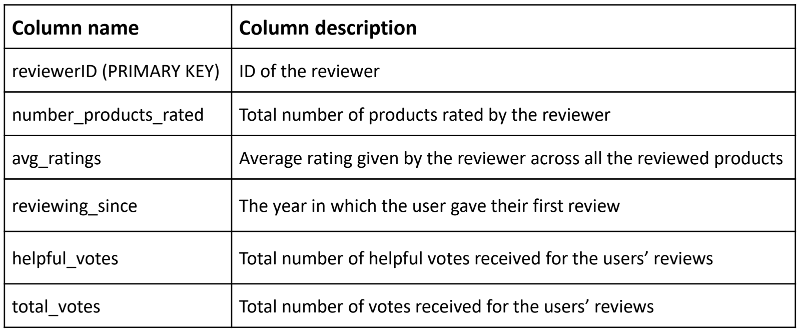
Table 1: Schema of Reviews table

The helpful attribute is a tuple of two integer values. The first value represents the number of people who found the review helpful, and the second value represents the total number of people who voted.

## 3 Tasks

You will use the *reviews* table to explore features related to users. Your task is to create a users table with the schema given in Table 2.

A code stub with the function signature has been provided to you. The input to this function is the path to the reviews CSV file and you will be carrying out a series of transformations to produce the required users table as a DataFrame. Plug in the DataFrame you obtained as a result in <`YOUR_USERS_DATAFRAME`>. The last line converts the dataframe into a json file and writes it to `results_PA0.json` file. Do not remove this line. We will time the execution of the function `PA0`.



| Column name | Column description |
|---|---|
| reviewerID (PRIMARY KEY) | ID of the reviewer |
| number_products_rated | Total number of products rated by the reviewer |
| avg_ratings | Average rating given by the reviewer across all the reviewed products |
| reviewing_since | The year in which the user gave their first review |
| helpful_votes | Total number of helpful votes received for the users' reviews |
| total_votes | Total number of votes received for the users' reviews |

Table 2: Schema of users table

We have shared with you the "development" dataset and our accuracy results. Our code's runtime on 1 node is roughly 615s. You can use this to validate your results and debug your code. The final evaluation will happen on a separate held-out test set. The runtime will be different for the held-out test set.

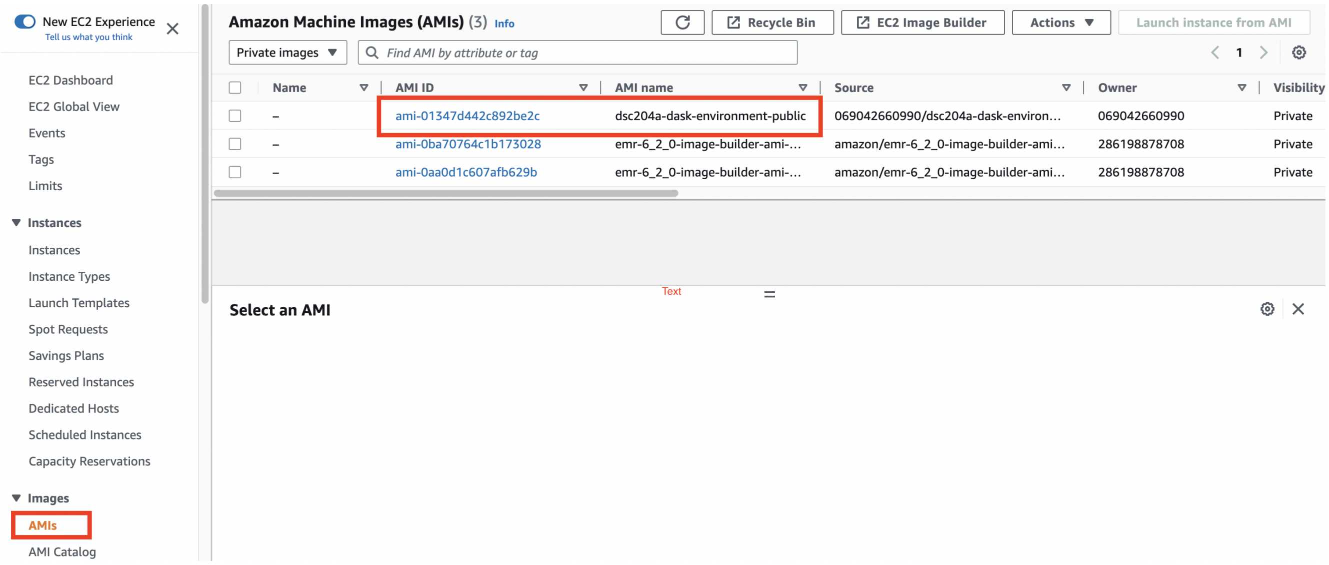
## 4 Deliverables

Submit your source code as `<NAME>_<PID>.py` on Canvas. Your source code must conform to the function signatures provided to you. Make sure that your code is writing results to `results_PA0.json`.

## 5 Getting Started

**1.** Access your ETS account using single sign-on ID: `https://ets-apps.ucsd.edu/individual/DSC204A_SP23_A00/`. To open the AWS console click "Click here to access AWS" at the bottom of the page. To get your AWS credentials for CLI / API usage click "Generate API Keys (for CLI/scripting)".

**2.** We have setup the Dask environment on an AMI with the name "dsc204a-dask-environment-public." Go to "AMIs" (under "Images") in your EC2 dashboard, select private images, and then search by name to find it. Select this AMI. See Figure 1.



| Name | AMI ID | AMI name | Source | Owner | Visibility |
|---|---|---|---|---|---|
| – | ami-01347d442c892be2c | dsc204a-dask-environment-public | 069042660990/dsc204a-dask-environ… | 069042660990 | Private |
| – | ami-0ba70764c1b173028 | emr-6_2_0-image-builder-ami-… | amazon/emr-6_2_0-image-builder-ami… | 286198878708 | Private |
| – | ami-0aa0d1c607afb629b | emr-6_2_0-image-builder-ami-… | amazon/emr-6_2_0-image-builder-ami… | 286198878708 | Private |

Figure 1

**3.** After selecting the AMI, click "Launch Instance from AMI" as shown below.

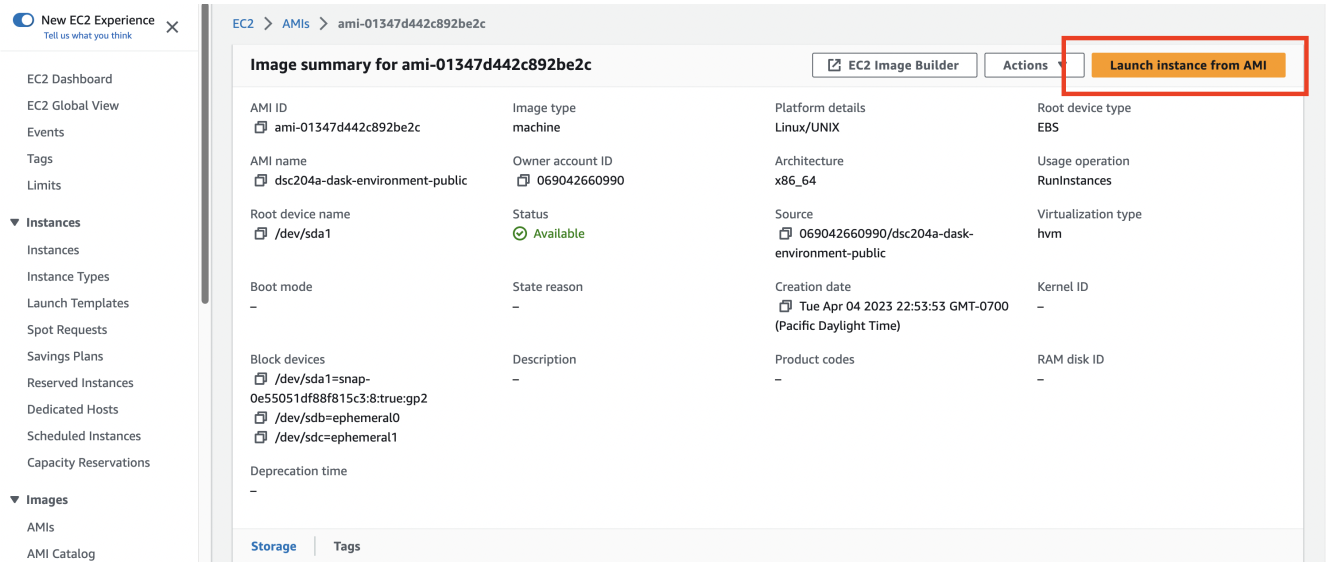


Image summary for ami-01347d442c892be2c

| AMI ID | Image type | Platform details | Root device type |
|---|---|---|---|
| ami-01347d442c892be2c | machine | Linux/UNIX | EBS |

| AMI name | Owner account ID | Architecture | Usage operation |
|---|---|---|---|
| dsc204a-dask-environment-public | 069042660990 | x86_64 | RunInstances |

| Root device name | Status | Source | Virtualization type |
|---|---|---|---|
| /dev/sda1 | Available | 069042660990/dsc204a-dask-environment-public | hvm |

| Boot mode | State reason | Creation date | Kernel ID |
|---|---|---|---|
| – | – | Tue Apr 04 2023 22:53:53 GMT-0700 (Pacific Daylight Time) | – |

| Block devices | Description | Product codes | RAM disk ID |
|---|---|---|---|
| /dev/sda1=snap-0e55051df88f815c3:8:true:gp2<br>/dev/sdb=ephemeral0<br>/dev/sdc=ephemeral1 | – | – | – |

Deprecation time
–

Figure 2

**4.** Now, strictly follow the below instructions to launch one **EC2 Spot** instance in which you will run your code.

**a.** Give any name for your instance.

**b.** Number of instances to launch is 1. The instance type is "t2.xlarge".

**c.** Create a new key pair for SSH'ing to your instance later. The private key will be downloaded to your local machine.

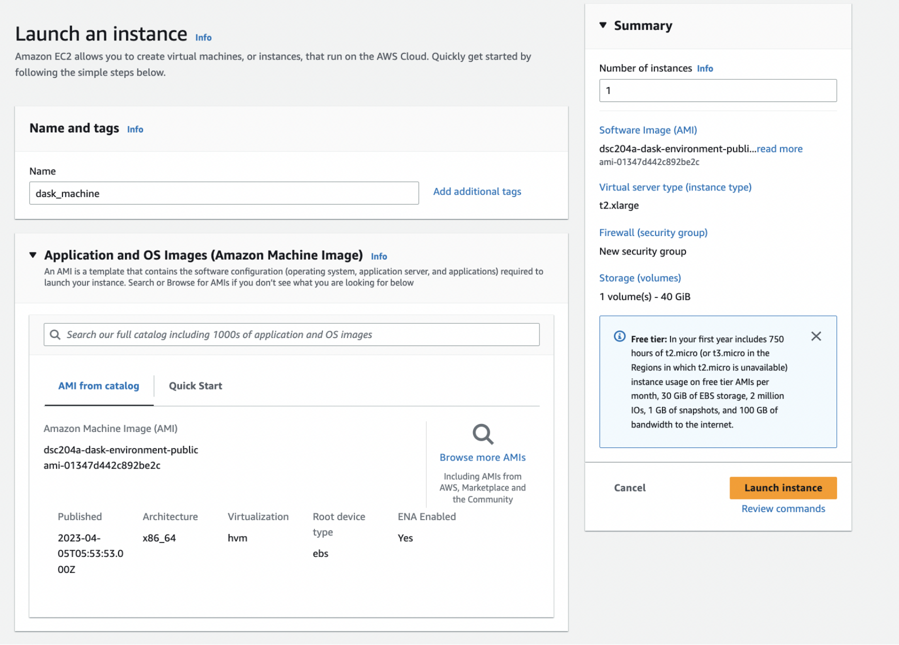
**d.** Let the network settings be same as default.

**e.** Choose 40GB SSD gp2 storage.

**f.** Open advanced details. Select "Request Spot Instances". Then click on "customize" just on the right. Open the dropdown for "Request type" and select "One-time".

**g.** Click "Launch Instance" as shown in Figure 5.

See below figures for the required configuration.



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: dask_machine

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

AMI from catalog | Quick Start

Amazon Machine Image (AMI)
dsc204a-dask-environment-public
ami-01347d442c892be2c

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

| Published | Architecture | Virtualization | Root device type | ENA Enabled |
|---|---|---|---|---|
| 2023-04-05T05:53:53.000Z | x86_64 | hvm | ebs | Yes |

▼ Summary

Number of instances Info: 1

Software Image (AMI)
dsc204a-dask-environment-publi...read more
ami-01347d442c892be2c

Virtual server type (instance type)
t2.xlarge

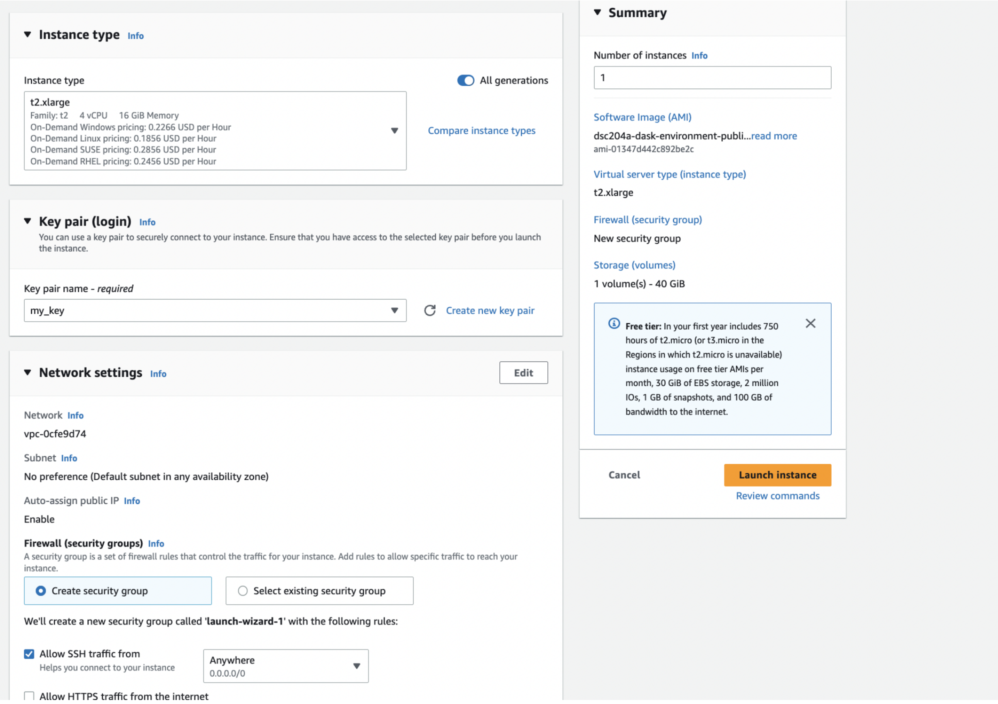
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 40 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance | Review commands

Figure 3



▼ Instance type Info

Instance type

All generations

t2.xlarge
Family: t2 4 vCPU 16 GiB Memory
On-Demand Windows pricing: 0.2266 USD per Hour
On-Demand Linux pricing: 0.1856 USD per Hour
On-Demand SUSE pricing: 0.2856 USD per Hour
On-Demand RHEL pricing: 0.2456 USD per Hour

Compare instance types

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: my_key

Create new key pair

▼ Network settings Info

Edit

Network Info
vpc-0cfe9d74

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

**Firewall (security groups)** Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called **'launch-wizard-1'** with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

□ Allow HTTPS traffic from the internet

▼ Summary

Number of instances Info: 1

Software Image (AMI)
dsc204a-dask-environment-publi...read more
ami-01347d442c892be2c

Virtual server type (instance type)
t2.xlarge

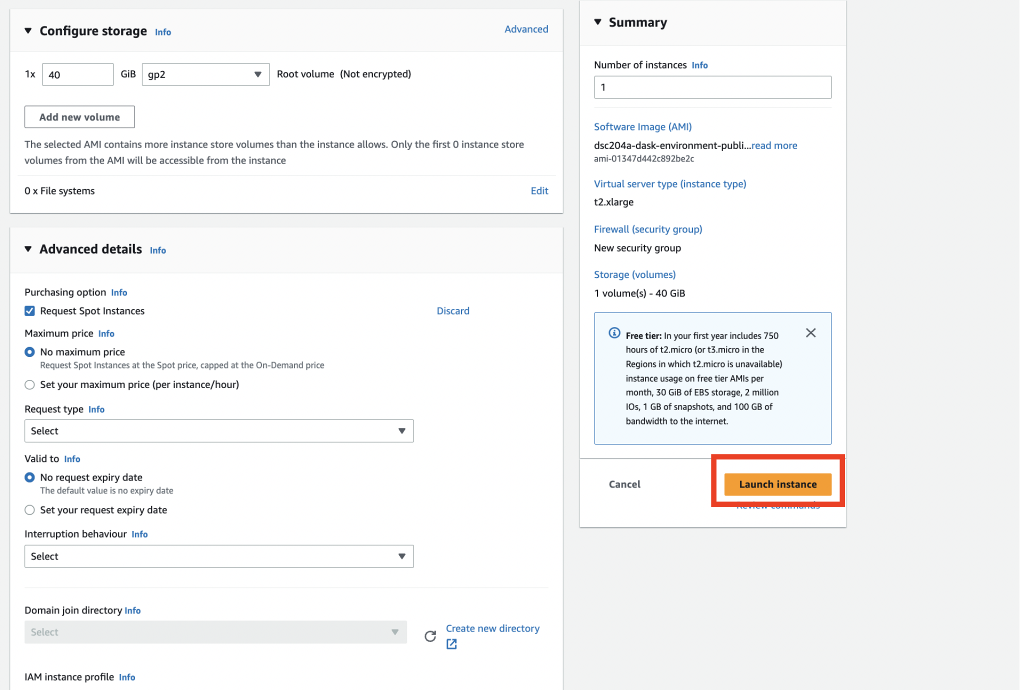
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 40 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance | Review commands

Figure 4



▼ Configure storage

1x 40 GiB gp2 ▼ Root volume (Not encrypted)

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems

▼ Advanced details

Purchasing option

☑ Request Spot Instances

Maximum price

● No maximum price
Request Spot Instances at the Spot price, capped at the On-Demand price

○ Set your maximum price (per instance/hour)

Request type: Select ▼

Valid to

● No request expiry date
The default value is no expiry date

○ Set your request expiry date

Interruption behaviour: Select ▼

Domain join directory: Select ▼

IAM instance profile

▼ Summary

Number of instances: 1

Software Image (AMI)
dsc204a-dask-environment-publi...read more
ami-01347d442c892be2c

Virtual server type (instance type)
t2.xlarge

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 40 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Figure 5

**5.** In these final steps you will SSH into your instance, download the dataset and start jupyter-notebook.

**a.** Change permission of the SSH keyfile to make sure your private key file isn't publicly viewable:

```
chmod 400 <keyfilename>.pem
```

**b.** SSH into one of the nodes using command:

```
ssh -i ‘‘<your_key>.pem’’ ubuntu@<ip-address-of-EC2-instance>
```

Public IP of your instance can be found inside the instance details of your running instance.

**c.** I will use tmux to manage my terminals. You can can use `tmux` or `screen` or whatever works for you, but `tmux` is recommended. Run `tmux new -s dt` to open a new window. We will use AWS CLI for downloading the dataset and code stub from our S3 bucket into our instance. First, export your AWS credentials (see step 1. on where you can find them) into the current shell environment. Then run -

```
aws s3 sync s3://dsc204a-public .
```

This will start the download.

**c.** Now, detach from your tmux session by pressing `Ctrl B + D`. Start another tmux session for running jupyter-notebook. In this session, first activate the Dask environment with the command:

```
source dask_env/bin/activate
```

Then, start jupyter notebook with the command: `jupyter-notebook`.

**d.** We will forward port 8888 on our AWS instance to our local machine so we can access jupyter notebook on our machine. Open a new terminal on your local machine and run this command -

```
ssh -i ‘‘<your_key>.pem’’ ubuntu@<ip-address-of-
EC2-instance> -L 8888:localhost:8888
```

Now, on your browser go to `http://localhost:8888` where you will be prompted to enter a token. You can find this token in the terminal output where you started jupyter notebook.

**e.** Dask also provides us with a Dask Dashboard (like Tensorboard for those who've worked with DL) where we can see the progress of our tasks. This is automatically started on port 8787. So, we will forward this port as well. Run this command in yet another terminal on your local machine -

```
ssh -i ‘‘<your_key>.pem’’ ubuntu@<ip-address-of-
EC2-instance> -L 8787:localhost:8787
```

You can visit `http://localhost:8787` but you will only see output once you have started the Dask Scheduler.

**VERY IMPORTANT: Download your progress to your local machine at regular intervals and terminate your instance when you decide to pause working. You have only $50 for both PA0 and PA1 and so DO NOT leave instances running. If you terminate without downloading, you WILL LOSE all your work. Every time you start a new instance, you must download the dataset from S3 to your instance. Also, start only AWS Spot Instances and NOT On-Demand instances.**