DSC 102: Systems for Scalable Analytics

Programming Assignment 0

Released: 12 Apr 2025, Due: 23 Apr 2025

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, 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 learn to set up 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. Instructions on obtaining the required CSV are later on in this document.

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	unix timestamp of 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. Update this stub with your implementation and import statements. 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 300s. 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 `<YOUR-TEAM-ID>.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/DSC102_SP25_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 "dsc102-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. After selecting the AMI, click "Launch Instance from AMI".

3. Now, strictly follow the below instructions to launch one EC2 Spot instance in which you will run your code (in the cloud, not on your laptop). Note that an AWS spot instance is heavily discounted in price, in exchange for giving AWS permissions to shut down your instance if demand for compute is high. Be mindful about backing up your code and associated artifacts.

   (a) Under 'Name', give your instance a name you will remember.

   (b) Under 'Number of instances' on the right side of the page, leave the value as 1.

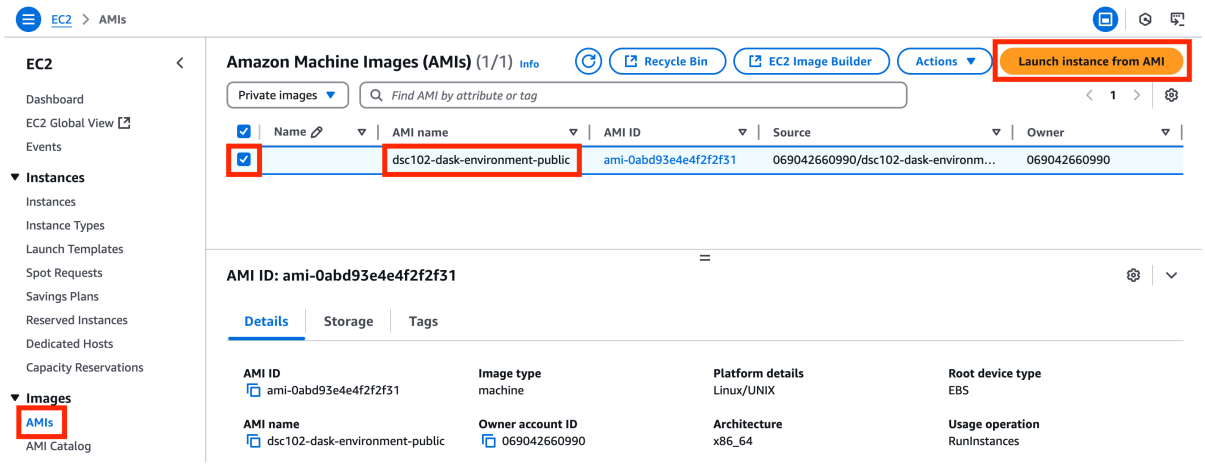

Figure 1

(c) Leave the 'Application and OS Images (Amazon Machine Image)' field as is, as that was pre-populated by your selection to run from the dsc102-dask-environment-public AMI.

(d) Under 'Instance type', select "t2.xlarge".

(e) Under the 'Key pair (login)' heading, click 'Create new key pair', give the key pair a name that you will remember, leave 'Key pair type' RSA checked, and then select the private key file format '.pem'. The private key will be downloaded to your local machine. Store the key in a location you will remember as you will be reusing this each time you want to log in (SSH) to your machine. Here is more info for Mac users: `https://www.youtube.com/watch?v=8UtgMcX_kg0` and for Windows users: `https://www.youtube.com/watch?v=kzLRxVgos2M`.

(f) Let the Network settings be same as default.

(g) Under 'Configure Storage', select "40GB" of storage on a 'general purpose SSD (gp3)'.

(h) 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".

(i) Click "Launch Instance" as shown in Figure 5. Return to the 'Instances' page and wait for your instance's 'Instance state' to be set to 'Running'.

See below figures for the required configuration.

4. In these final steps you will SSH into your instance, download the dataset and start a 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) Open a terminal window on your local machine. SSH into the EC2 instance that you launched in the previous step 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 as shown in Figure 6.

(c) We will use AWS CLI for downloading the dataset and code stub from our S3 bucket into our instance. Go to the UCSD ETS landing page where you clicked the link to access the AWS console in step 1. Click "Generate API Keys (for CLI/scripting)". You will find three export statements there, corresponding to `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, and `AWS_SESSION_TOKEN`. Copy all the text there into your EC2 terminal (where you just ssh-ed

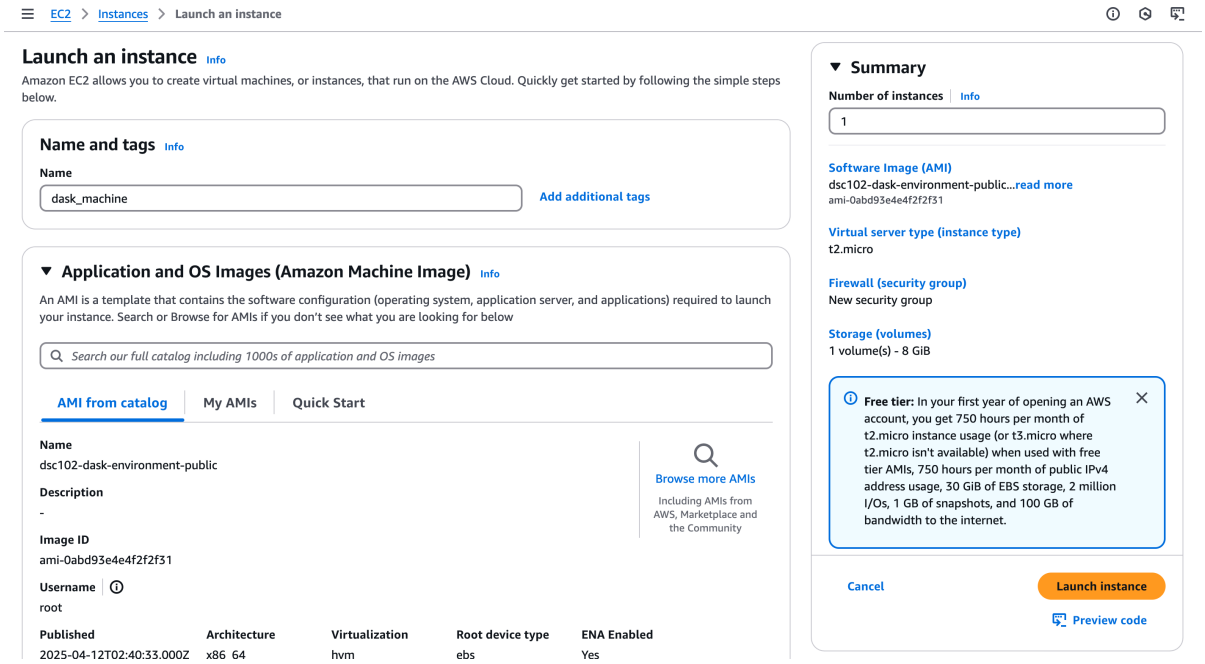

Figure 2

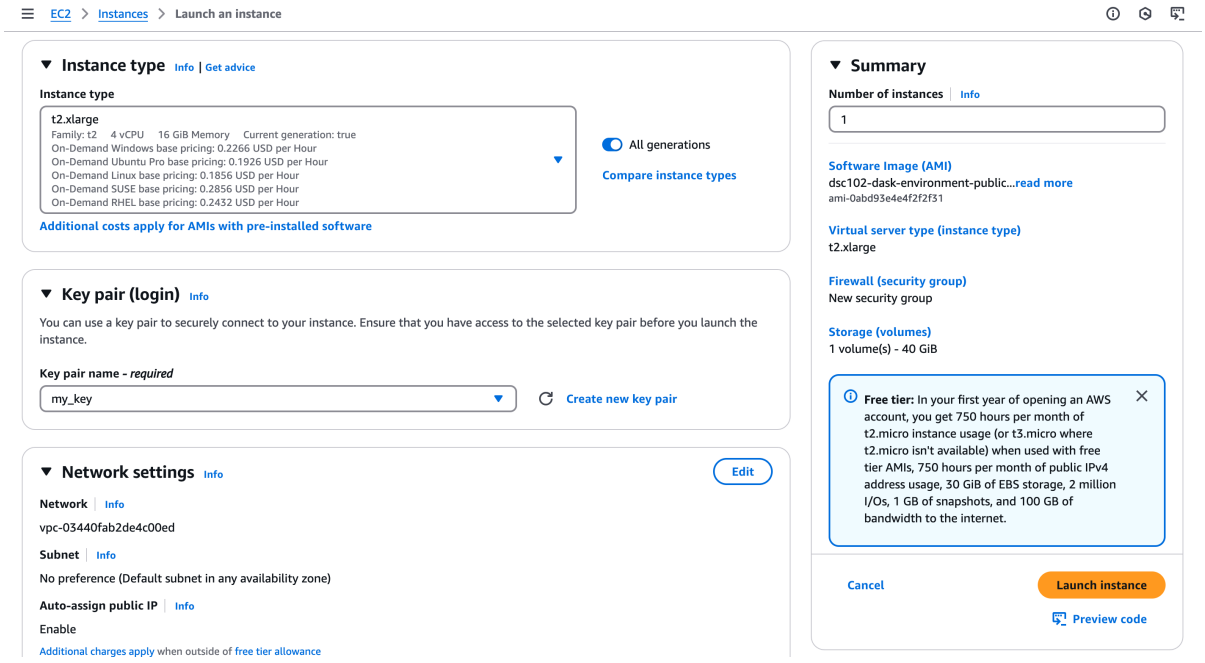

Figure 3

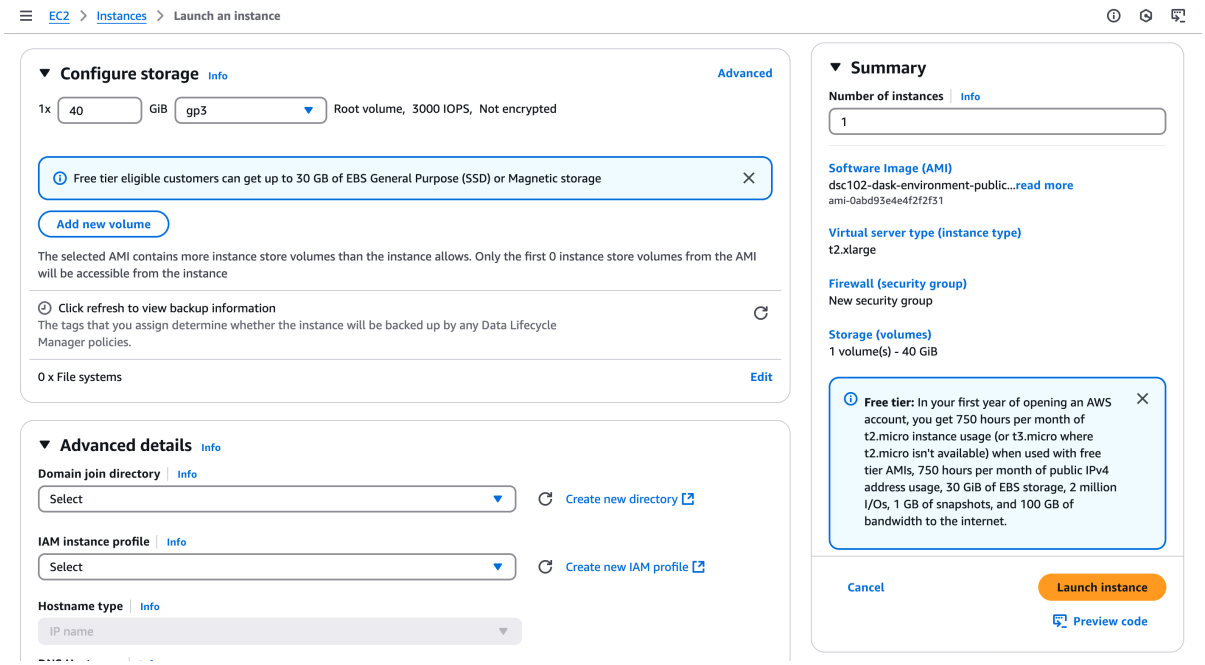

Figure 4

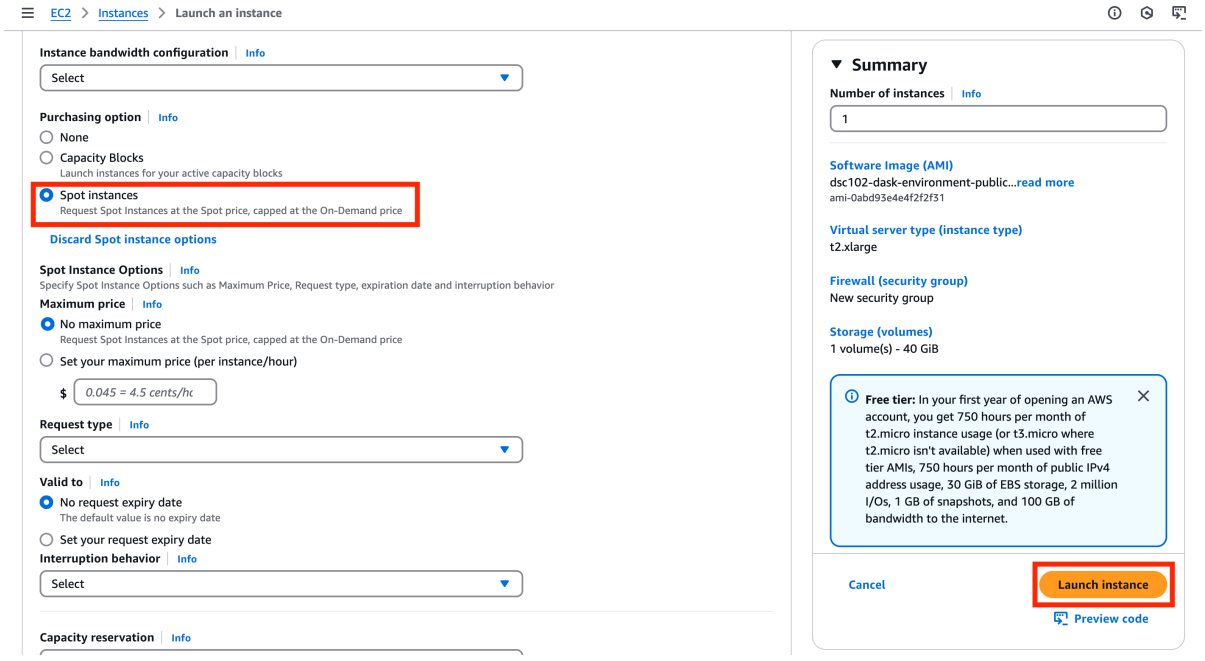

Figure 5

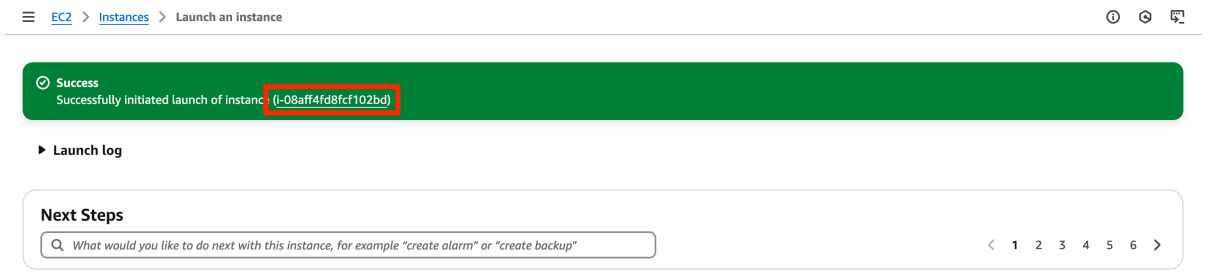

Figure 6

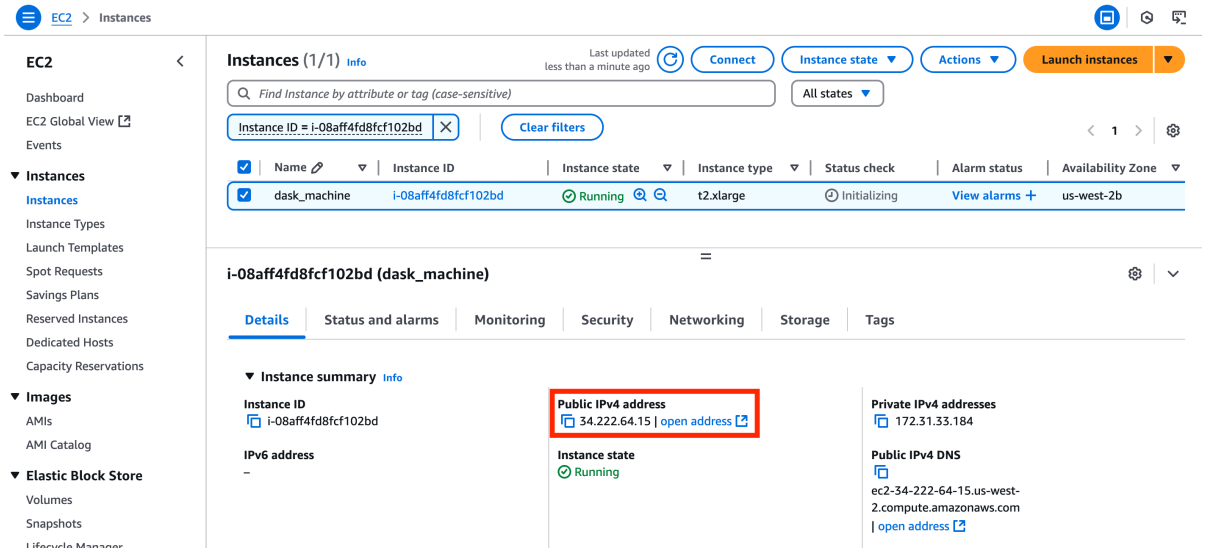

Figure 7

in), and you are now authenticated to copy objects from S3.
Then run -
`aws s3 sync s3://dsc102-public /home/ubuntu/`
This will start the download.

(d) Activate the Dask environment with the command:
`source dask_env/bin/activate`
Then, start Jupyter notebook with the command:
`jupyter-notebook --port=8888`

(e) We will forward port 8888 on our AWS instance to our local machine so we can access Jupyter notebook on our local machine.
Open a new terminal window 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.

(f) Dask also provides us with a Dask Dashboard (like Tensorboard for those who've worked with deep learning) where we can see the progress of our tasks. This is automatically started on port 8787 of the EC2 instance. So, we will forward this port as well so we can access the dashboard on our local machine.
Open a new terminal window on your local machine and run this command:
`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 the output once you have started the Dask Scheduler.

Consider using utilities like `tmux` or `screen` for managing terminals.

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, 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.