

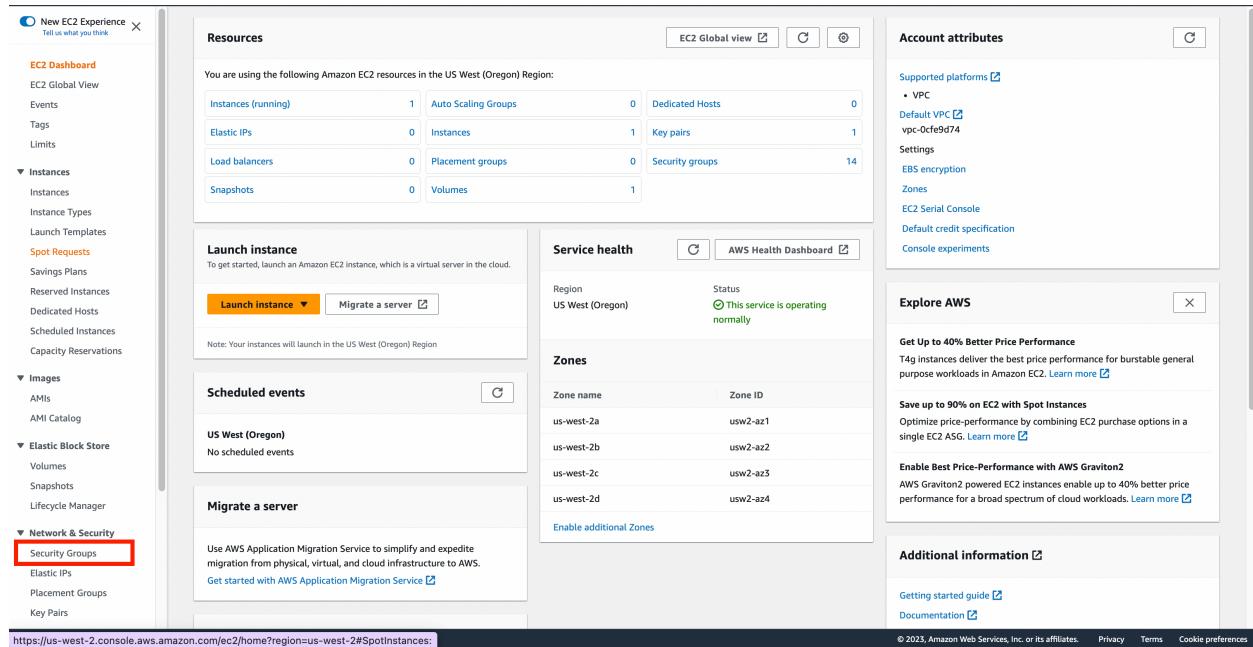
AWS SETUP

The AWS setup is very similar to PA0.

For launching multiple instances, there are three differences.

- Firstly, we will specify 5 instances instead of 1.
- One of these instances will run our Jupyter Notebook and Dask Scheduler, and the remaining 4 instances will run our Dask workers.
- Each of these instances will have 100GB SSD storage instead of 40GB.
- We will create a new security group for our 5 instances that allow each of the instances to communicate with each other. Follow the below steps one by one.

1. Access your ETS account using single sign-on ID: https://ets-apps.ucsd.edu/individual/DSC102_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. Open AWS Dashboard. We will first create a new security group so that we can apply it to all of our instances later. Click on “[Security Groups](#)” on the left menu.



The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Network & Security' section, the 'Security Groups' item is highlighted with a red box. The main content area displays the 'Resources' section, which lists various Amazon EC2 resources in the US West (Oregon) Region. It includes tables for Instances (running), Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. Below this, there are sections for 'Launch instance', 'Service health', 'Zones', 'Scheduled events', and 'Migrate a server'. The 'Service health' section shows the status as 'This service is operating normally'. The 'Zones' section lists four availability zones: us-west-2a, us-west-2b, us-west-2c, and us-west-2d, each associated with a specific Zone ID. The 'Explore AWS' and 'Additional information' sections provide links to various AWS services and documentation. The URL in the browser bar is <https://us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#SpotInstances>.

3. Click on create security group.

The screenshot shows the AWS EC2 Security Groups page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Limits, Instances, Images, and AMIs. The main area displays a table titled 'Security Groups (14) info' with columns for Name, Security group ID, Security group name, VPC ID, Description, Owner, Inbound rules count, and Outbound rules. Each row represents a different security group, such as 'launch-wizard-4' or 'dask_security_group'. At the top right of the table, there's a 'Create security group' button, which is highlighted with a red box.

- Choose any name and description for your security group. Change both the inbound and outbound rules to have "Type=All Traffic", "Source/Destination = Anywhere-IPv4". See image below.

This screenshot shows the 'Create new security group' wizard. It has several steps: 'Basic details', 'Inbound rules', 'Outbound rules', and 'Tags - optional'. The 'Basic details' step is completed with a security group name 'dask_security_group' and a VPC 'vpc-0cfed9d74'. The 'Inbound rules' and 'Outbound rules' sections are both highlighted with red boxes. Each section contains a single rule: 'All traffic' from 'Anywhere-IPv4' to '0.0.0.0/0'. The 'Tags - optional' section at the bottom allows adding up to 50 tags.

- Lastly, click on "Create Security Group" at the bottom right.

- Now, we will create our 5 instances which will use this new security group. 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.

The screenshot shows the AWS EC2 console under the 'New EC2 Experience' tab. On the left, a sidebar navigation includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (selected), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', and 'Capacity Reservations'. Under 'Images', 'AMIs' is selected. The main content area displays 'Amazon Machine Images (AMIs) (1/1)' with a single entry: 'Name: ami-087ab6cf5ee2ab526', 'AMI ID: ami-087ab6cf5ee2ab526', 'Source: dsc102-dask-environment-public', and 'Owner: 035170873046/dsc102-dask-environm...'. A search bar at the top has 'dsc' entered. Below this, a detailed view for 'AMI ID: ami-087ab6cf5ee2ab526' shows fields like 'Image type: machine', 'Platform details: Linux/UNIX', 'Root device type: EBS', 'Architecture: x86_64', 'Usage operation: RunInstances', 'Virtualization type: hvm', and 'Creation date'. The 'Details' tab is active.

7. After selecting the AMI, click "[Launch Instance from AMI](#)" as shown below.
8. Now, strictly follow the below instructions to launch the EC2 Spot instances.
 - a. Give any name for your instance.
 - b. Number of instances to launch is 5.
 - c. The instance type is "t2.xlarge".
 - d. Create a new key or use the key previously used for PA0.
 - e. Under "[Network Settings](#)", click "[Select Existing Security Group](#)" and choose the name of the security group you just created.
 - f. Choose 100GB SSD gp2 storage.
 - g. 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" and limit the "Set your maximum price (per instance/hour) to as low as possible. (Start From 0.07)
9. Lastly, click "[Launch Instance](#)".

SETUP CLIENT AND SCHEDULER :

(For the 1st EC2 Instance)

Step 1 : Setup Client i.e Jupyter Notebook and Port Forwarding for Jupyter Notebook onto localhost.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
✓	pa1	i-08b49961db3e69fc2	Running	t2.xlarge	2/2 checks passed	No alarms	us-west-2c	ec2-34-221-146-31.us...	34.221.146.31	-
✗	pa1	i-03cb9d140d2fe869e	Running	t2.xlarge	2/2 checks passed	No alarms	us-west-2c	ec2-34-217-137-104.us...	34.217.137.104	-
✗	pa1	i-0454ae152445fadcb	Running	t2.xlarge	2/2 checks passed	No alarms	us-west-2c	ec2-35-92-24-86.us-we...	35.92.24.86	-
✗	pa1	i-0c1b1002e1d2f8469	Running	t2.xlarge	2/2 checks passed	No alarms	us-west-2c	ec2-54-203-131-59.us...	54.203.131.59	-
✗	pa1	i-0517c357f8857adad	Running	t2.xlarge	2/2 checks passed	No alarms	us-west-2c	ec2-34-219-174-56.us...	34.219.174.56	-

1. Open a terminal and Do the following:

a. SSH Into the Scheduler EC2 Instance :

```
ssh -i <pem_key_name>.pem ubuntu@34.221.146.31
```

b. Activate the Dask Environment :

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

c. Launch Jupyter Notebook on the EC2 :

```
jupyter notebook --port=8888
```

```
ubuntu@ip-172-31-12-224:~$ source dask_env/bin/activate
(dask_env) ubuntu@ip-172-31-12-224:~$ jupyter notebook --port=8888
[I 19:04:18.101 NotebookApp] Writing notebook server cookie secret to /home/ubuntu/.local/share/jupyter/runtime/notebook_cookie_secret
[I 19:04:20.632 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 19:04:20.632 NotebookApp] Jupyter Notebook 6.4.10 is running at:
[I 19:04:20.632 NotebookApp] http://localhost:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
[I 19:04:20.632 NotebookApp] or http://127.0.0.1:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
[E 19:04:20.632 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 19:04:20.639 NotebookApp] No web browser found: could not locate runnable browser.
[C 19:04:20.639 NotebookApp]

To access the notebook, open this file in a browser:
  file:///home/ubuntu/.local/share/jupyter/runtime/nbserver-12027-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
  or http://127.0.0.1:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
```

2. Open New Terminal And run the following commands –

1. Port Forwarding Jupyter Notebook running on port 8888 on the EC2 to port 8000 on local system (localhost:8000):

```
ssh -i <pem_key_name>.pem ubuntu@34.221.4.2 -L 8000:localhost:8888
```

```
X ubuntu@ip-172-31-12-224: ~ (ssh)
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

6 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Thu Jan 19 06:07:19 2023 from 24.43.123.81
ubuntu@ip-172-31-12-224:~$ source dask_env/bin/activate
(dask_env) ubuntu@ip-172-31-12-224:~$ jupyter notebook --port=8888
[I 19:04:18.101 NotebookApp] Writing notebook server cookie secret to /home/ubuntu/.local/share/jupyter/runtime/notebook_cookie_se
cret
[I 19:04:20.632 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 19:04:20.632 NotebookApp] Jupyter Notebook 6.4.10 is running at:
[I 19:04:20.632 NotebookApp] http://localhost:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
[I 19:04:20.632 NotebookApp] or http://127.0.0.1:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
[E 19:04:20.632 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 19:04:20.639 NotebookApp] No web browser found: could not locate runnable browser.
[C 19:04:20.639 NotebookApp]

To access the notebook, open this file in a browser:
  file:///home/ubuntu/.local/share/jupyter/runtime/nbserver-12027-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
  or http://127.0.0.1:8888/?token=fac098e07f82116c3c0b8c9539cb94eaf45d5e4a14ab773a
[]

X ubuntu@ip-172-31-12-224: ~ (ssh)
(base) golokeshpatra@Golokeshs-MacBook-Air ~ % cd ~/Downloads/Q2/DSC102/Assignments/PA1
(base) golokeshpatra@Golokeshs-MacBook-Air PA1 % ssh -i dsc102-pa1.pem ubuntu@34.221.146.31 -L 8000:localhost:8888
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1092-aws x86_64)

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

System information as of Thu Feb  2 19:06:42 UTC 2023

System load:  0.01          Processes:      124
Usage of /:   2.8% of 96.73GB  Users logged in:     1
Memory usage: 2%            IP address for eth0: 172.31.12.224
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

6 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Thu Feb  2 19:03:52 2023 from 24.43.123.92
ubuntu@ip-172-31-12-224:~$ []
```

Step 2 : Setup Dask Scheduler on the same EC2.

On the same node, we must start the dask-scheduler. To establish this, we need to follow these instructions:

1. Open a New Terminal :

 - a. **SSH Into the EC2 Instance**
 - b. **Activate the Dask Environment :**
source dask_env/bin/activate
 - c. **Activate the dask-scheduler :**
dask scheduler --host 0.0.0.0

```
(dask_env) ubuntu@ip-172-31-2-152:~$ dask scheduler --host=0.0.0.0
2023-04-26 20:20:07,694 - distributed.scheduler - INFO - -----
2023-04-26 20:20:08,220 - distributed.http.proxy - INFO - To route to workers diagnostics web server please install jupyter-server-proxy: python -m pip install jupyter-server-proxy
2023-04-26 20:20:08,265 - distributed.scheduler - INFO - State start
2023-04-26 20:20:08,267 - distributed.scheduler - INFO - -----
2023-04-26 20:20:08,268 - distributed.scheduler - INFO -   Scheduler at:  tcp://172.31.2.152:8786
2023-04-26 20:20:08,268 - distributed.scheduler - INFO -   dashboard at:  http://172.31.2.152:8787/status
```

NOTE: The 'Scheduler at' shown above will be required while setting up workers

Step 3 : Dask UI/Dashboard Port forwarding .

1. Open a New Terminal:
 - a. **Port Forward the dask dashboard UI from the EC2's 8787 port to local systems Port 8001 (localhost:8001) :**
ssh -i dsc102-pa1.pem ubuntu@34.221.4.2 -L 8001:localhost:8787

Step 4 : Downlaod data from S3 :

1. In the same terminal opened or in a new terminal :
 - a. **If New Terminal then activate the dask environment**
 - b. **Copy and paste the AWS_ACCESS_KEY and SECRET ACCESS KEYS**
 - c. **Download all the files from the S3 :**
aws s3 sync s3://dsc102-public /home/ubuntu/

Step 5 : Check Dask UI and Jupyter Notebook.

NOTE : Unlike PA0 , when you go to the Info tab on Dask UI, you SHOULD NOT OBSERVE ANY WORKERS .

DASK UI :

The screenshot shows a web browser window with the URL `localhost:8001/info/main/workers.html`. The title bar says "Scheduler `tcp://172.31.4.65:8786`". Below the title, there are two tabs: "Logs" (selected) and "Bokeh". A horizontal menu bar includes "Worker", "Name", "Cores", "Memory", "Memory use", "Occupancy", "Processing", "In-memory", "Services", "Logs", and "Last seen". The main content area is currently empty, showing only the header row of the table.

Jupyter Notebook :

The screenshot shows the Jupyter Notebook dashboard. At the top, there's a logo and links for "Quit" and "Logout". Below that, a navigation bar has tabs for "Files" (selected), "Running", and "Clusters". A message "Select items to perform actions on them." is displayed above a file list. The file list table has columns for "Name", "Last Modified", and "File size". The files listed are:

	Name	Last Modified	File size
<input type="checkbox"/>	/		
<input type="checkbox"/>	dask_env	10 days ago	
<input type="checkbox"/>	OutputSchema_PA1.json	3 hours ago	417 B
<input type="checkbox"/>	PA1.py	3 hours ago	1.22 kB
<input type="checkbox"/>	products_Release.csv	3 hours ago	9.79 GB
<input type="checkbox"/>	results_PA1.json	2 hours ago	2.44 kB
<input type="checkbox"/>	user_reviews.csv	3 hours ago	28.5 GB

SETUP DASK WORKER:

(For other 4 EC2 Instances)

Step 1 : SSH into the worker node

Step 2: Setup dask worker on the worker node.

1. To get the Dask Workers Up, we need to follow these instructions :
 - a. **After SSH into the worker EC2, Activate the Dask Environment :**
source dask_env/bin/activate
 - b. **Now activate the dask worker with the following command:**
dask worker tcp://172.31.152:8786 --nworkers 4

NOTE : The dask scheduler IP can be obtained from [Step 2 of Setting up Scheduler](#)

```
(dask_env) ubuntu@ip-172-31-1-242:~$ dask worker tcp://172.31.2.152:8786 --nworkers 4
2023-04-26 20:58:54,824 - distributed.nanny - INFO - Start Nanny at: 'tcp://172.31.1.242:40655'
2023-04-26 20:58:54,827 - distributed.nanny - INFO - Start Nanny at: 'tcp://172.31.1.242:36873'
2023-04-26 20:58:54,830 - distributed.nanny - INFO - Start Nanny at: 'tcp://172.31.1.242:45675'
2023-04-26 20:58:54,838 - distributed.nanny - INFO - Start Nanny at: 'tcp://172.31.1.242:40845'
2023-04-26 20:58:55,798 - distributed.worker - INFO - Start worker at: 'tcp://172.31.1.242:35981
2023-04-26 20:58:55,798 - distributed.worker - INFO - Listening to: 'tcp://172.31.1.242:35981
2023-04-26 20:58:55,798 - distributed.worker - INFO - dashboard at: 172.31.1.242:40725
2023-04-26 20:58:55,798 - distributed.worker - INFO - Waiting to connect to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,798 - distributed.worker - INFO - -----
2023-04-26 20:58:55,799 - distributed.worker - INFO - Threads: 1
2023-04-26 20:58:55,799 - distributed.worker - INFO - Memory: 3.91 GiB
2023-04-26 20:58:55,799 - distributed.worker - INFO - Local Directory: '/tmp/dask-worker-space/worker-n68ms144
2023-04-26 20:58:55,799 - distributed.worker - INFO - -----
2023-04-26 20:58:55,803 - distributed.worker - INFO - Start worker at: 'tcp://172.31.1.242:40487
2023-04-26 20:58:55,804 - distributed.worker - INFO - Listening to: 'tcp://172.31.1.242:40487
2023-04-26 20:58:55,804 - distributed.worker - INFO - dashboard at: 172.31.1.242:38999
2023-04-26 20:58:55,804 - distributed.worker - INFO - Waiting to connect to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,804 - distributed.worker - INFO - -----
2023-04-26 20:58:55,804 - distributed.worker - INFO - Threads: 1
2023-04-26 20:58:55,804 - distributed.worker - INFO - Memory: 3.91 GiB
2023-04-26 20:58:55,804 - distributed.worker - INFO - Local Directory: '/tmp/dask-worker-space/worker-x4tdxup6
2023-04-26 20:58:55,804 - distributed.worker - INFO - -----
2023-04-26 20:58:55,808 - distributed.worker - INFO - Registered to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,808 - distributed.worker - INFO - -----
2023-04-26 20:58:55,810 - distributed.core - INFO - Starting established connection to 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,810 - distributed.worker - INFO - Start worker at: 'tcp://172.31.1.242:44943
2023-04-26 20:58:55,810 - distributed.worker - INFO - Listening to: 'tcp://172.31.1.242:44943
2023-04-26 20:58:55,810 - distributed.worker - INFO - dashboard at: 172.31.1.242:37215
2023-04-26 20:58:55,810 - distributed.worker - INFO - Waiting to connect to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,811 - distributed.worker - INFO - -----
2023-04-26 20:58:55,811 - distributed.worker - INFO - Threads: 1
2023-04-26 20:58:55,811 - distributed.worker - INFO - Memory: 3.91 GiB
2023-04-26 20:58:55,811 - distributed.worker - INFO - Local Directory: '/tmp/dask-worker-space/worker-z758171z
2023-04-26 20:58:55,811 - distributed.worker - INFO - -----
2023-04-26 20:58:55,814 - distributed.worker - INFO - Registered to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,814 - distributed.worker - INFO - -----
2023-04-26 20:58:55,815 - distributed.core - INFO - Starting established connection to 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,817 - distributed.worker - INFO - Registered to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,817 - distributed.worker - INFO - -----
2023-04-26 20:58:55,818 - distributed.core - INFO - Starting established connection to 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,826 - distributed.worker - INFO - Start worker at: 'tcp://172.31.1.242:38621
2023-04-26 20:58:55,827 - distributed.worker - INFO - Listening to: 'tcp://172.31.1.242:38621
2023-04-26 20:58:55,827 - distributed.worker - INFO - dashboard at: 172.31.1.242:43297
2023-04-26 20:58:55,827 - distributed.worker - INFO - Waiting to connect to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,827 - distributed.worker - INFO - -----
2023-04-26 20:58:55,827 - distributed.worker - INFO - Threads: 1
2023-04-26 20:58:55,827 - distributed.worker - INFO - Memory: 3.91 GiB
2023-04-26 20:58:55,827 - distributed.worker - INFO - Local Directory: '/tmp/dask-worker-space/worker-9s725y3m
2023-04-26 20:58:55,827 - distributed.worker - INFO - -----
2023-04-26 20:58:55,833 - distributed.worker - INFO - Registered to: 'tcp://172.31.2.152:8786
2023-04-26 20:58:55,833 - distributed.core - INFO - Starting established connection to 'tcp://172.31.2.152:8786
```

Now if we check the Worker Info on the DASK UI then –

Scheduler tcp://172.31.8.102:8786

Worker	Name	Cores	Memory	Memory use	Occupancy	Processing	In-memory	Services	Logs	Last seen
Logs Bokeh										
tcp://172.31.5.152:35911	tcp://172.31.5.152:35911	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	510.11 ms
tcp://172.31.5.152:38641	tcp://172.31.5.152:38641	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	510.62 ms
tcp://172.31.5.152:41199	tcp://172.31.5.152:41199	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	508.22 ms
tcp://172.31.5.152:41637	tcp://172.31.5.152:41637	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	514.77 ms

Step 3: Download the data from s3 to the worker nodes too.

1. Open a new Terminal:

Repeat the [Step 4 in the scheduler setup section](#)

Step 4 : Check the DASK Dashboard/UI for Information of the workers.

We should observe 16 Worker nodes i.e 4 Workers * 4 nodes = 16 Nodes

For Reference -

localhost:8001/info/main/workers.html										
Worker	Name	Cores	Memory	Memory use	Occupancy	Processing	In-memory	Services	Logs	Last seen
Logs Bokeh										
tcp://172.31.12.248:33483	tcp://172.31.12.248:33483	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	766.98 ms
tcp://172.31.12.248:36339	tcp://172.31.12.248:36339	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	771.40 ms
tcp://172.31.12.248:42139	tcp://172.31.12.248:42139	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	770.26 ms
tcp://172.31.12.248:46435	tcp://172.31.12.248:46435	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	769.32 ms
tcp://172.31.12.96:33965	tcp://172.31.12.96:33965	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	719.89 ms
tcp://172.31.12.96:34225	tcp://172.31.12.96:34225	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	720.44 ms
tcp://172.31.12.96:39611	tcp://172.31.12.96:39611	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	721.26 ms
tcp://172.31.12.96:41441	tcp://172.31.12.96:41441	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	722.73 ms
tcp://172.31.13.222:39279	tcp://172.31.13.222:39279	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	316.18 ms
tcp://172.31.13.222:40447	tcp://172.31.13.222:40447	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	316.58 ms
tcp://172.31.13.222:40543	tcp://172.31.13.222:40543	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	318.89 ms
tcp://172.31.13.222:46063	tcp://172.31.13.222:46063	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	318.49 ms
tcp://172.31.2.228:33005	tcp://172.31.2.228:33005	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	714.36 ms
tcp://172.31.2.228:33185	tcp://172.31.2.228:33185	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	717.33 ms
tcp://172.31.2.228:35339	tcp://172.31.2.228:35339	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	717.03 ms
tcp://172.31.2.228:39095	tcp://172.31.2.228:39095	1	4.20 GB	<div style="width: 100px; height: 10px; background-color: #ccc;"></div>	0.00 us	0	0	dashboard	logs	721.38 ms

Step 5 : Repeat All the above steps on each and every Worker EC2 Instance.