

IPYTHON (JUPYTER) NOTEBOOK WITH SPARK ON AWS EC2 INSTANCE

1. CREATE AWS EDUCATE ACCOUNT

You can create a normal AWS account (Credit card required) with your **UF email**.

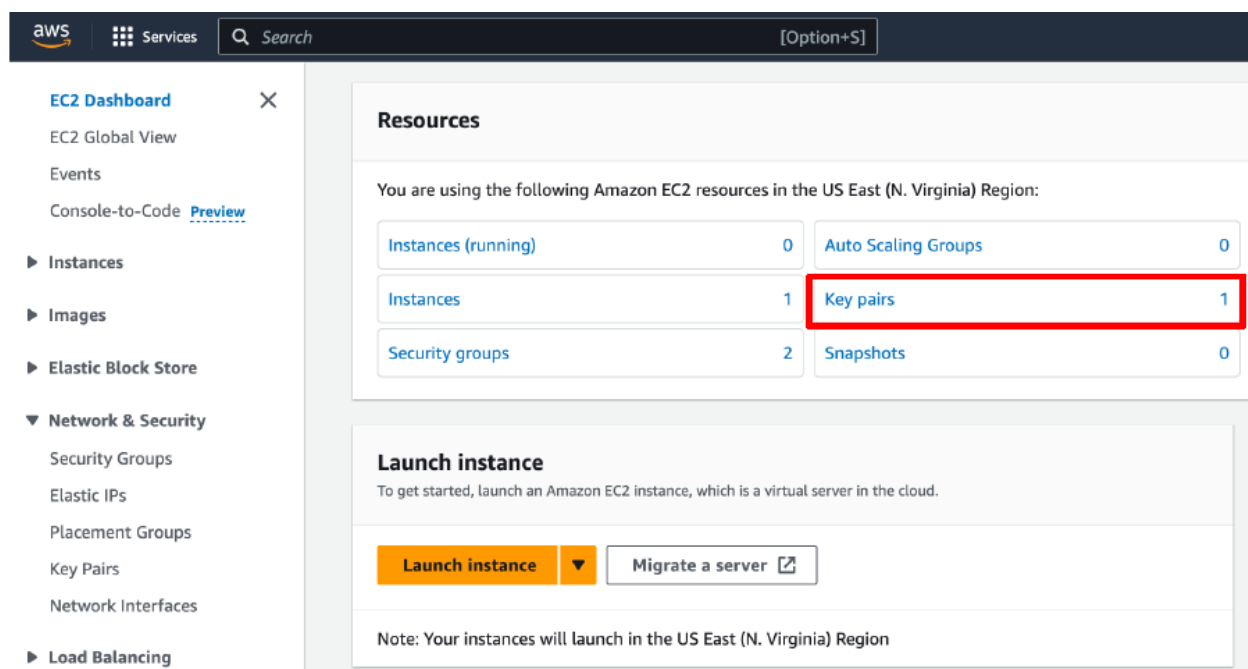
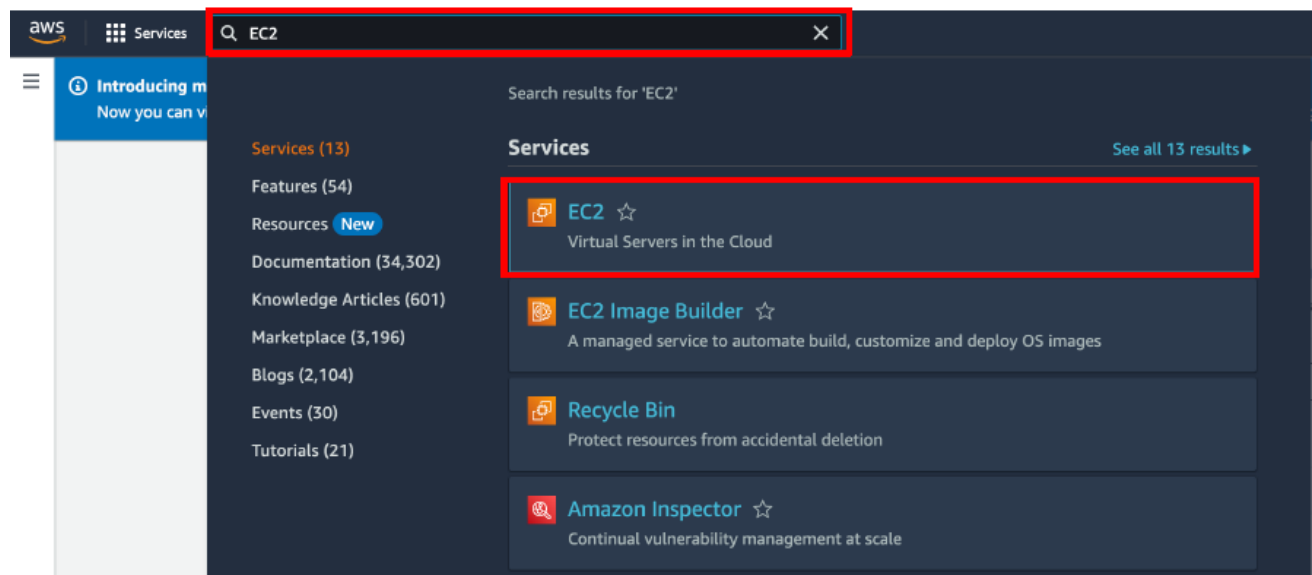
2. LAUNCH AND CONNECT TO AN EC2 INSTANCE

1. Sign into your AWS account console: <https://aws.amazon.com/>

Make sure you are in Region US East (N. Virginia) in the top right corner.

2. Get key pair:

On your console, click 'Services' -> EC2 -> 0 Key Pairs -> Create Key Pair



EC2 > Key pairs > Create key pair

Create key pair [Info](#)

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#) Document was last saved: Just now
☒ RSA ☐ ED25519

Private key file format
☒ .pem
For use with OpenSSH
☐ .ppk
For use with PuTTY

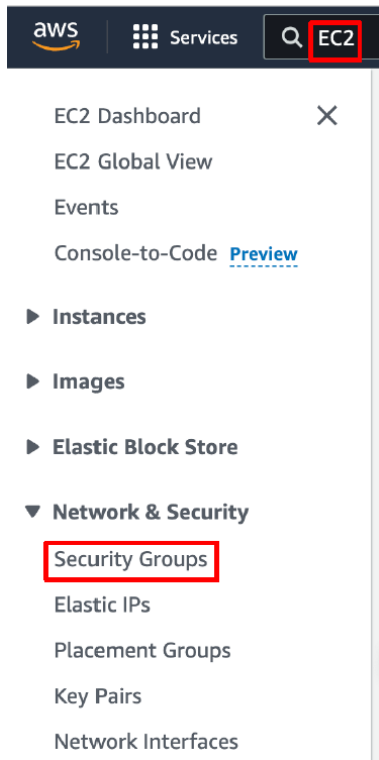
Tags - *optional*
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Create key pair](#)

Download the 'pem' file in a safe space for later use.

Put the 'pem' file in your local lab folder, and open terminal in the local folder,
\$ cd < path to the local lab folder >
type:
\$ chmod 400 ~/your_key.pem
for ssh client to read

3. Create security group by clicking 'Services' -> EC2 -> Security Groups -> Create Security Group
Make sure you are in Region US East (N. Virginia) in the top right corner.



Give the name and description of the security group.

Basic details

Security group name [Info](#)

CAP4770-2024summer-security-group

Name cannot be edited after creation.

Description [Info](#)

Allow ssh access and custom TCP access over certain ports.

Then add the following inbound and outbound rules.

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
HTTPS	TCP	443	Anywhere-IPv4	0.0.0.0/0	Delete
HTTPS	TCP	443	Anywhere-IPv6	::/0	Delete
All TCP	TCP	0 - 65535	Anywhere-IPv4	0.0.0.0/0	Delete
All TCP	TCP	0 - 65535	Anywhere-IPv6	::/0	Delete
Custom TCP	TCP	8888	Anywhere-IPv4	0.0.0.0/0	Delete
Custom TCP	TCP	8888	Anywhere-IPv6	::/0	Delete

[Add rule](#)

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info	
SSH	TCP	22	Anywhere-IPv4	0.0.0.0/0	Delete
SSH	TCP	22	Anywhere-IPv6	::/0	Delete
HTTPS	TCP	443	Anywhere-IPv4	0.0.0.0/0	Delete
HTTPS	TCP	443	Anywhere-IPv6	::/0	Delete
Custom TCP	TCP	8888	Anywhere-IPv4	0.0.0.0/0	Delete
Custom TCP	TCP	8888	Anywhere-IPv6	::/0	Delete
Custom TCP	TCP	4040	Anywhere-IPv4	0.0.0.0/0	Delete
Custom TCP	TCP	4040	Anywhere-IPv6	::/0	Delete
All traffic	All	All	Anywhere-IPv4	0.0.0.0/0	Delete

Then, create the security group.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

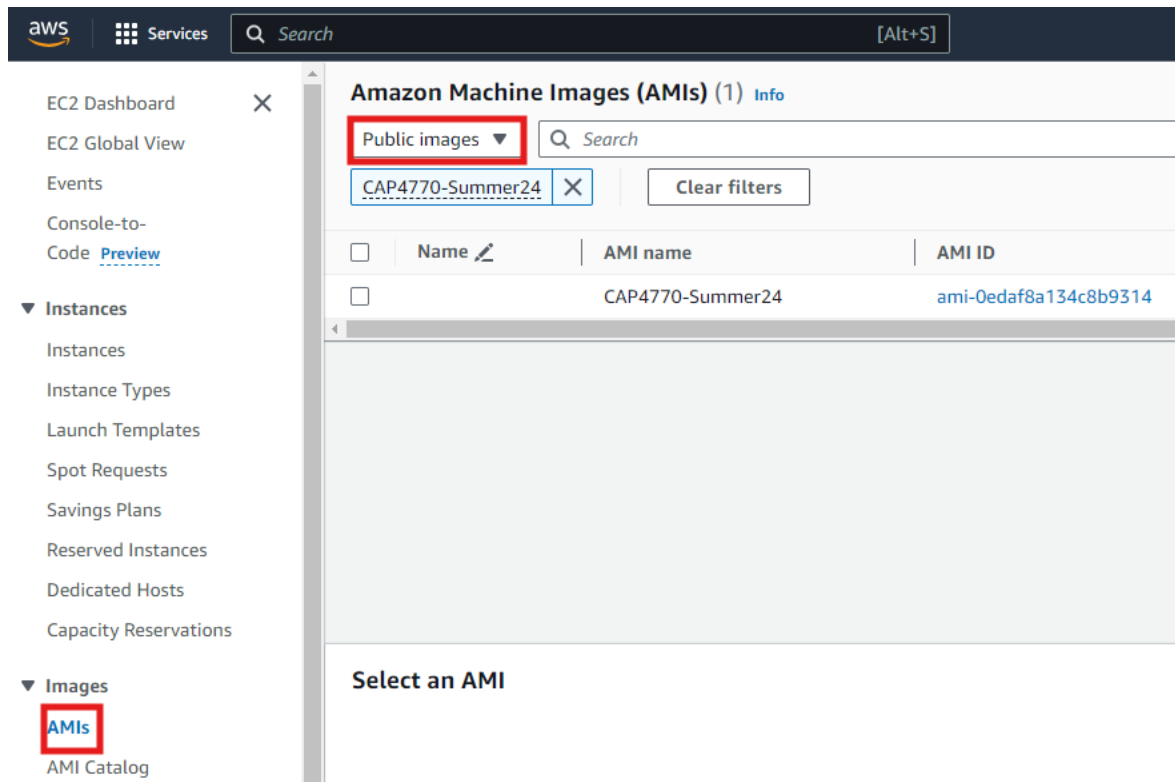
No tags associated with the resource.

[Add new tag](#)

You can add up to 50 new tags.

[Cancel](#) [Create security group](#)

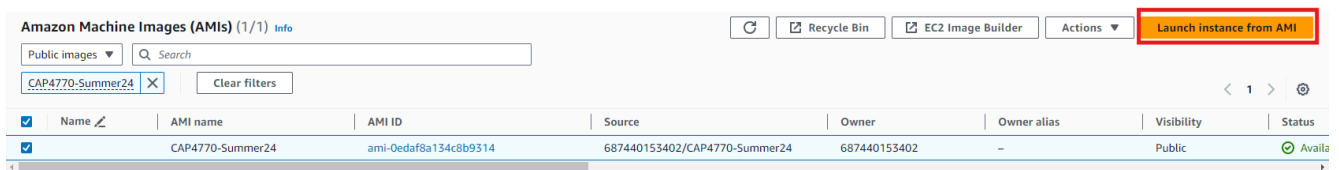
4. Start EC2 instance from preconfigured AMI click 'Services' -> EC2 -> AMIs



Make sure under the launch button, it says public images, and you are in Region US East (N. Virginia) in the top right corner.

In the search bar, search for 'CAP4770-Summer24'

An AMI image with the name and ID 'ami-0edaf8a134c8b9314' should show up.



Select the image, and click 'Launch instance from AMI' button as shown above, and follow the steps below:

Step 1: Choose AMI: Already chosen.

Step 2: Give your instance a name.

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

[Add additional tags](#)

Step 3: Choose Instance Type: t2.micro (free tier), **you can update this to large instance later if you want to process larger dataset.**

Note: Sometimes the micro instance is not sufficient to run pyspark. In case, you experience errors and timeouts while executing the flatmap function or while saving the file to S3, try with t2.medium. Make sure to terminate the instance after capturing the required screenshots to avoid being charged when the instance is not used.

For this lab, t2.medium is recommended over t2.micro to avoid timeouts.

Step 4: Select the key pair created from the dropdown menu.

▼ 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*

[Create new key pair](#)



Proceed without a key pair (Not recommended)

Default value

CAP-4770-IntroDS

Type: rsa



[Edit](#)

Step 5: Network settings: Select existing group and choose the security group created earlier.

▼ Network settings

Info

Edit

Network

Info

vpc-0738d1e6f0cfd05ca

Subnet

Info

No preference (Default subnet in any availability zone)

Auto-assign public IP

Info

Enable

Additional charges apply when outside of free tier allowance

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

Common security groups

Info

Select security groups

CAP4770-2024summer-security-group sg-0387be8072dcd9578

×

VPC: vpc-0738d1e6f0cfd05ca

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Step 6: Config Storage: default, skip

Step 7: Configure Instance (In Advanced details):

In AWS setup, create an IAM role manually, following these steps: (If you leave the "IAM role" as "None", then you will see an error "Unable to load AWS credentials from any provider in the chain" when trying to access S3 in ipython notebook)

1) Click "Create new IAM role."

▼ Advanced details

Info

Domain join directory

Info

Select

Create new directory

IAM instance profile

Info

Select

Create new IAM profile

2) Click "Create role"

IAM

>

Roles

Roles (3)

Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

<

1

>

⚙

☐

Role name

☐

Trusted entities

☐

Last activity

Refresh

Delete

Create role

3) Search and Click "EMR" in "Use case" section:

Select trusted entity [Info](#)

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Q EMR

Other services

Amazon **EMR** Serverless

EMR

EMR Containers

Choose a service or use case

Cancel **Next**

4) Click "EMR Role for EC2" then click "Next: ..." buttons (in lower-right corner) until you see "Create role" button (also in lower-right corner).

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EMR

Choose a use case for the specified service.

Use case

☐ **EMR**
Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

☒ **EMR Role for EC2**
Allows EC2 instances in an Elastic MapReduce cluster to call AWS services such as S3 on your behalf.

☐ **EMR - Cleanup**
Allows EMR to terminate instances and delete resources from EC2 on your behalf.

Cancel **Next**

5) Give it a name and click "Create role".

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

cap-4770-EMR-for-EC2-role

Maximum 64 characters. Use alphanumeric and '+=, @, _' characters.

Description

Add a short explanation for this role.

Allows EC2 instances in an Elastic MapReduce cluster to call AWS services such as S3 on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @, _' characters.

Step 3: Add tags

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create role

Then you will see this role in the "IAM role" section when setting EC2 instance. Just select it like mentioned above.

▼ Advanced details [Info](#)

Domain join directory [Info](#)

Select



[Create new directory](#)

IAM instance profile [Info](#)

cap4770-EMR-for-EC2-role

arn:aws:iam::560132372676:instance-profile/cap4770-EMR-for-EC2-role



[Create new IAM profile](#)



Select

cap4770-EMR-for-EC2-role

arn:aws:iam::560132372676:instance-profile/cap4770-EMR-for-EC2-role



Step8: Review and Click “Launch instance”

Advanced details [Info](#)

Domain join directory [Info](#)

Select [Create new directory](#)

IAM instance profile [Info](#)

cap4770-EMR-for-EC2-role [Create new IAM profile](#)

arn:aws:iam::560132372676:instance-profile/cap4770-EMR-for-EC2-role

Hostname type [Info](#)

IP name

DNS Hostname [Info](#)

☒ Enable IP name IPv4 (A record) DNS requests

☒ Enable resource-based IPv4 (A record) DNS requests

☐ Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery [Info](#)

Select

Shutdown behavior [Info](#)

Stop

Stop - Hibernate behavior [Info](#)

Select

Termination protection [Info](#)

Select

Summary

Number of instances [Info](#)

1

Software Image (AMI)

CAP4770-2024sp

ami-05813e5ab88d26c1d

Virtual server type (instance type)

t2.medium

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 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 I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch instance** [Review commands](#)

Now click on Services -> EC2 -> Instances -> Instances (or you can again click Services -> EC2 -> Instances (running)). You will be able to view the instance created.

aws **Services** **EC2**

EC2 Dashboard [X](#)

EC2 Global View

Events

Console-to-Code [Preview](#)

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Resources

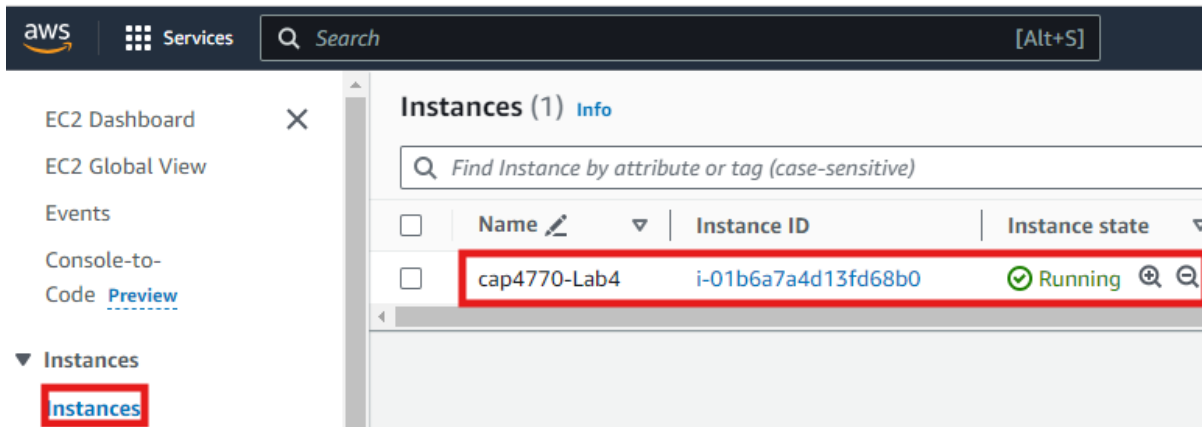
You are using the following Amazon EC2 resources in the

Instances (running) 1

Instances 1

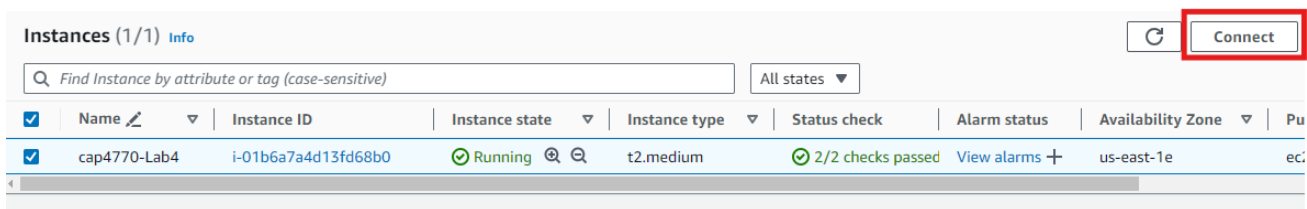
Security groups 2

And you'll see the instance we just created.

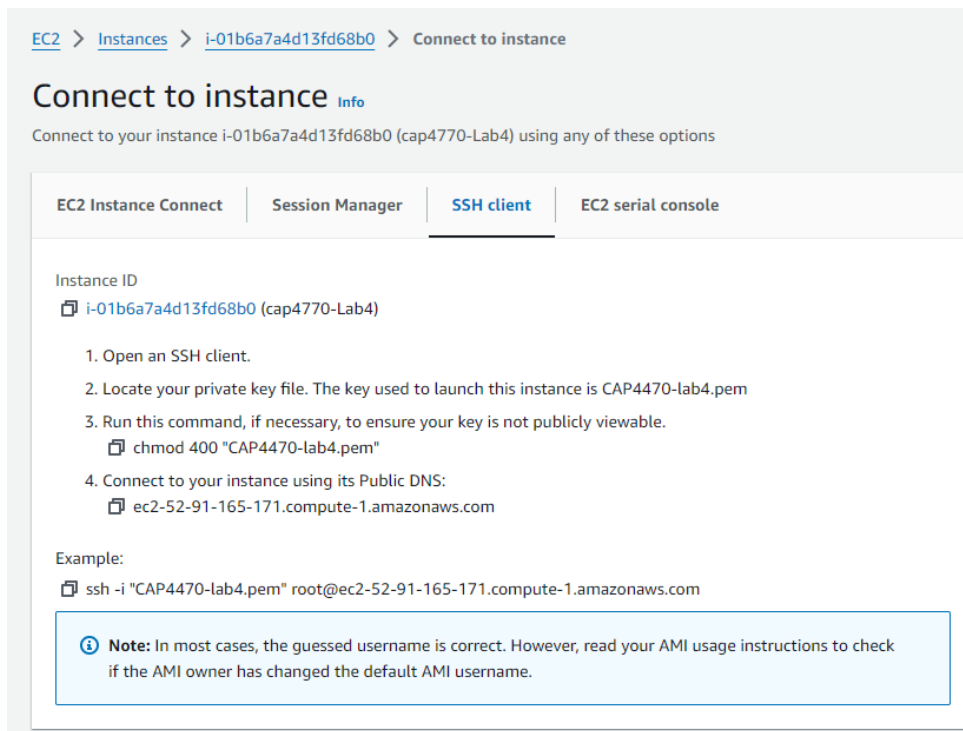


5. Connect to the instance.

Select the running instance and click 'connect'.



Go to the SSH client tab, you'll see a screen like this:



Switch to your local machine, in the terminal, copy and paste the example command (note: update with your actual DNS, and make sure to log in as user 'ubuntu'):

```
$ ssh -i "<path to the .pem file that you downloaded before>" ubuntu@ec2-3-85-136-34.compute-1.amazonaws.com
```

if denied because of bad key permission try:

```
$ chmod 400 "<path to the .pem file that you downloaded before>"
```

Or just use sudo command:

```
$ sudo ssh -i "<path to the .pem file that you downloaded before>"  
ubuntu@ec2-3-85-136-34.compute-1.amazonaws.com
```

```
PS C:\Users\jayet\OneDrive - University of Florida\Documents\summer 2024\Introduction to data science\Summer 2024\labs\cap4778Docker\lab_4_1> ssh -i "C:\Users\jayet\OneDrive - University of Florida\Documents\summer 2024\Introduction to data science\Summer 2024\labs\cap4778Docker\lab_4_1\CAP4778-lab4.pem" ubuntu@ec2-3-85-136-34.compute-1.amazonaws.com
The authenticity of host 'ec2-3-85-136-34.compute-1.amazonaws.com (52.91.165.171)' can't be established.
ED25519 key fingerprint is SHA256:teQJHInGoDx8DkYgkoVnG4v6fyb8NMAgCnJTVVSPDA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-85-136-34.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)

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

System information as of Thu Jun  6 22:03:41 UTC 2024

System load:  0.00      Processes:      114
Usage of /:   17.0% of 22.21GB   Users logged in:  0
Memory usage: 6%          IPv4 address for enx0: 172.31.54.102
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

51 updates can be applied immediately.
34 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Jun  3 01:03:15 2024 from 70.171.32.245
ubuntu@ip-172-31-54-102:~$
```

And you have logged into the instance in the cloud via ssh. (Note: your actual .pem file and public DNS will be different from the example)

Important Notes:

1. Don't forget to stop/terminate your EC2 Instance after using it, otherwise it will cost your extra credits/money!

Also Please note that different regions are independent, so make sure you stop/terminate all running instances in different regions.

2. Your data on the EC2 instance will be lost if you terminate it. You can save your output results on S3, for Jupyter Notebooks you create, you can download them from the browser via File -> Download as Jupyter notebook.

3. RUN SPARK WITH IPYTHON NOTEBOOK IN REMOTE INSTANCE

In the terminal where you ssh into the remote instance from last step above,

type

```
$ pyspark
```

you'll see following:

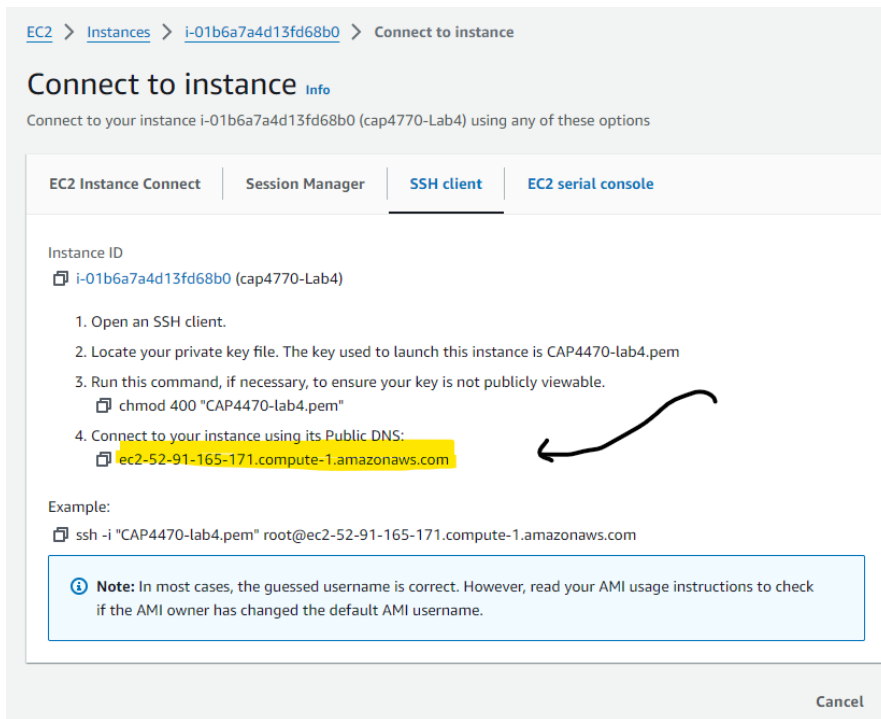
```
ubuntu@ip-172-31-36-251: ~  
File Edit View Search Terminal Help  
ubuntu@ip-172-31-36-251:~$ pyspark  
[TerminalIPythonApp] WARNING | Subcommand 'ipython notebook' is deprecated and will be removed in future versions.  
[TerminalIPythonApp] WARNING | You likely want to use 'jupyter notebook' in the future  
[I 03:28:56.739 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found  
[I 03:28:56.804 NotebookApp] ✓ nbpresent HTML export ENABLED  
[W 03:28:56.805 NotebookApp] ✗ nbpresent PDF export DISABLED: No module named 'nbbrowserpdf'  
[I 03:28:56.843 NotebookApp] [nb_anacondacloud] enabled  
[I 03:28:56.846 NotebookApp] [nb_conda] enabled  
[I 03:28:56.849 NotebookApp] Serving notebooks from local directory: /home/ubuntu  
[I 03:28:56.849 NotebookApp] 0 active kernels  
[I 03:28:56.849 NotebookApp] The Jupyter Notebook is running at: https://[all ip addresses on your system]:8888/  
[I 03:28:56.849 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).  
█
```

Now Open your browser on your laptop (can be the browser in your Host OS), and input the address in the URL (**Note it may start with “https”**):

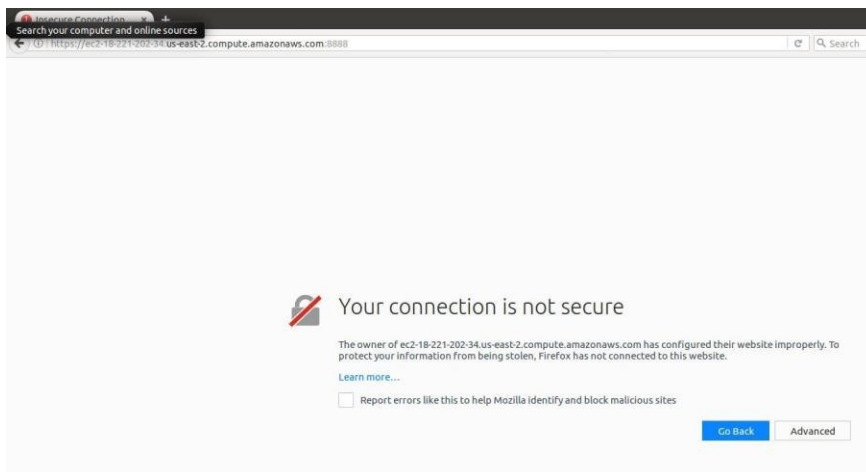
<http://ec2-52-91-165-171.compute-1.amazonaws.com:8888/>

Note your DNS should be different in your case, change accordingly.

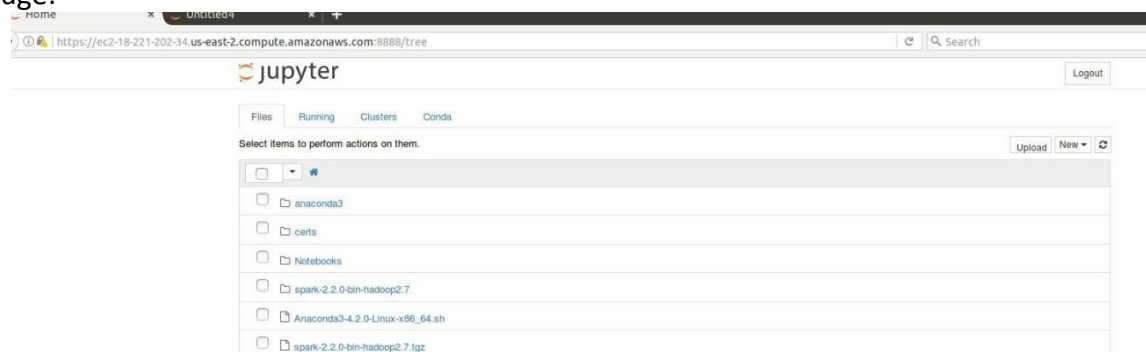
The public IPV4 DNS address can be found in the ‘connect to instance screen’ (under SSH client):



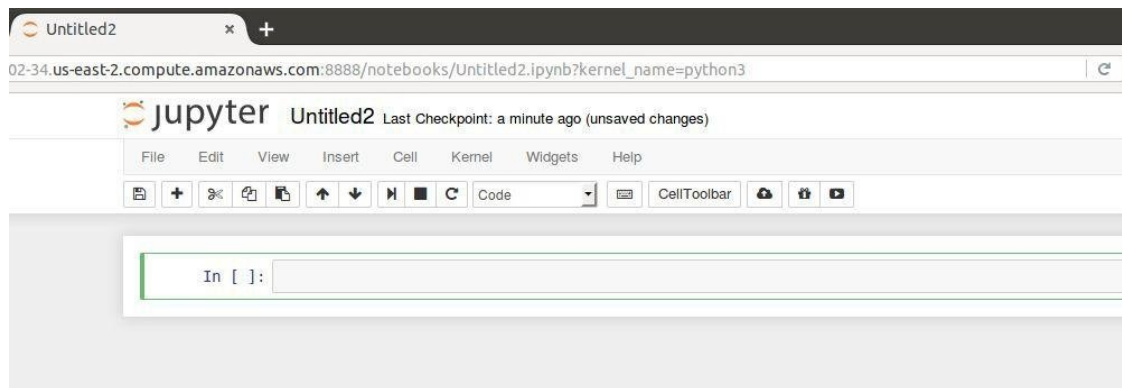
If you see a warning, click advanced and add Exception to continue.



And input your password (password is **123456** by default. You'll see a familiar Jupyter Notebook page:

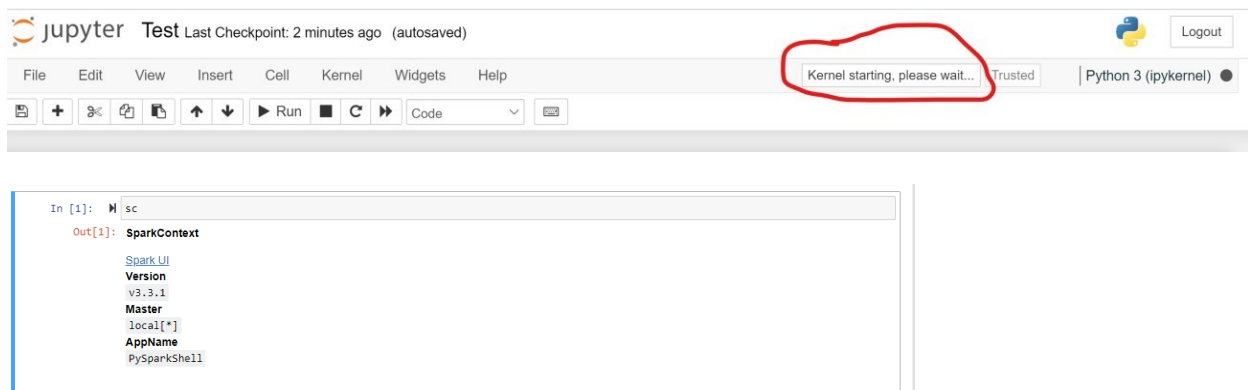


Create a test Ipython notebook by click New -> Python [default], you'll be prompted to a new page:



Type `sc` in the cell and you'll see (in a short while, ignore the warnings)

(Note: Always wait for the kernel to be ready, then run your code blocks)



Now you have a remote server running Spark with Ipython Notebook and you can start coding using PySparkShell interactively!

Optionally, you can monitor the status of the remote Spark via link below (again change to your DNS here too. **Note it must start with “http”, not https**):

<http://ec2-52-91-165-171.compute-1.amazonaws.com:4040/>

4. CONFIGURE ACCESS WITH S3

The remote instance we create does not hold data indefinitely -- any data on the instance will be lost if we terminate the EC2 instance.

AWS offers S3 for permanent data storage. Here is how to use it:

Terminate the pyspark process above (Ctrl + c twice), and re-run with additional dependency to read/write to S3 as below:

```
$ pyspark --packages com.amazonaws:aws-java-sdk:1.12.403,org.apache.hadoop:hadoop-aws:3.3.1
```

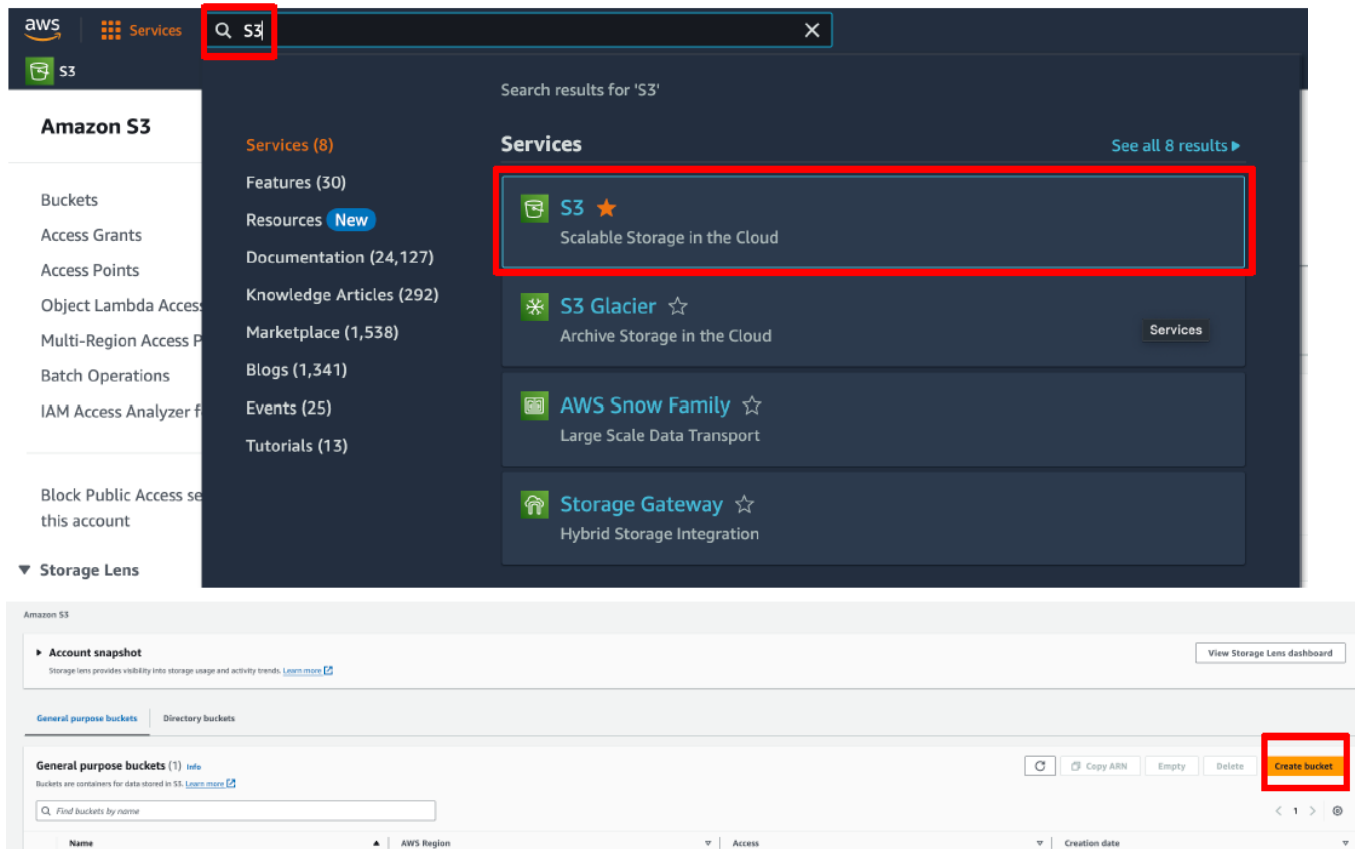
Reload your previous webpage to reconnect to this new Spark PySparkShell instance.

(Note: Always wait for the kernel to be ready, then run your code blocks)

You also need to create bucket(s) on S3 to store your data:

Go to Services -> Search S3 -> click on S3.

1. Upload file to S3
2. Go to Services -> Search S3 -> click on S3 -> Create Bucket



Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type [Info](#)



General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.



Directory - *New*

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

cap4770-2024summer

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#) [↗](#)

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Create a bucket name, e.g., 'cap4770-2024summer'. Click next and use default settings. After creation, a new bucket will show up:

Amazon S3 > Buckets

▶ Account snapshot - updated every 24 hours [All AWS Regions](#) [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#) [↗](#)

[General purpose buckets](#) | [Directory buckets](#)

General purpose buckets (2) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

< 1 > ⚙

	Name ▲	AWS Region ▼	IAM Access Analyzer	Creation date ▼
<input type="radio"/>	cap4770-2024summer	US East (N. Virginia) us-east-1	View analyzer for us-east-1	June 6, 2024, 19:42:42 (UTC-04:00)

Click on the bucket and you can upload files (by clicking on add files)

Create a local test file, say 'test.txt' containing some random sentences, and upload it to the bucket we just created like below.

Upload
[Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 66.0 B)
Remove
Add files
Add folder

All files and folders in this table will be uploaded.

Find by name

☐
Name
Folder

☐
test.txt
-

Destination
[Info](#)

Destination
s3://cap4770-2024summer

▶ Destination details
Bucket settings that impact new objects stored in the specified destination.

▶ Permissions
Grant public access and access to other AWS accounts.

▶ Properties
Specify storage class, encryption settings, tags, and more.

Cancel
Upload

Now in the jupyter notebook file, type something like this to read it (note: change to your bucket name and file name):

In [3]: `sc.textFile('s3a://<name-of-your-bucket>/test.txt').collect()`

```
[5]: sc

[5]: SparkContext
Spark UI

Version      v3.5.1
Master       local[*]
AppName      PySparkShell

[6]: sc.textFile('s3a://cap4770-2024summer/test.txt').collect()

[6]: ['Data science is an interdisciplinary field that uses scientific methods, algorithms, and systems to extract insights from data. It combines domains like data science, statistics, and information science to analyze and interpret complex data.']
```

Now, try executing the following code on Jupyter Notebook:

```
[9]: rdd = sc.textFile('s3a://cap4770-2024summer/test.txt')
    rdd1 = rdd.flatMap(lambda line: line.split(' '))
```

```
[10]: rdd2 = rdd1.map(lambda word: (word, 1))
```

```
[11]: rdd2
```

```
[11]: PythonRDD[10] at RDD at PythonRDD.scala:53
```

```
[12]: rdd3 = rdd2.reduceByKey(lambda x, y: x + y)
```

```
[13]: rdd3.collect()
```

```
[13]: [('science', 2),
      ('an', 1),
      ('field', 1),
      ('uses', 1),
      ('and', 3),
      ('to', 2),
      ('insights', 1),
      ('from', 1),
      ('data.', 2),
      ('It', 1),
      ('like', 1),
      ('science,', 1),
      ('statistics,', 1),
      ('information', 1),
      ('analyze', 1),
      ('Data', 1),
      ('is', 1),
      ('interdisciplinary', 1),
      ('that', 1),
      ('scientific', 1),
      ('methods,', 1),
      ('algorithms,', 1),
      ('systems', 1),
      ('extract', 1),
      ('combines', 1),
      ('domains', 1),
      ('data', 1),
      ('interpret', 1),
      ('complex', 1)]
```

Or more concisely:

```
[16]: rdd = sc.textFile('s3a://cap4770-2024summer/test.txt')
    rdd.flatMap(lambda line: line.split(' ')) \
        .map(lambda word: (word, 1)) \
        .reduceByKey(lambda x, y: x + y) \
        .collect()
```

```
[16]: [('science', 2),
      ('an', 1),
      ('field', 1),
      ('uses', 1),
      ('and', 3),
      ('to', 2),
      ('insights', 1),
      ('from', 1),
      ('data.', 2),
      ('It', 1),
      ('like', 1),
      ('science,', 1),
      ('statistics,', 1),
      ('information', 1),
      ('analyze', 1),
      ('Data', 1),
      ('is', 1),
      ('interdisciplinary', 1),
      ('that', 1),
      ('scientific', 1),
      ('methods,', 1),
      ('algorithms,', 1),
      ('systems', 1),
      ('extract', 1),
      ('combines', 1),
      ('domains', 1),
      ('data', 1),
      ('interpret', 1),
      ('complex', 1)]
```

Now that we have get the word counts, we can save them back to S3 for permanent storage, by replacing `'.collect()'` with `'.saveAsTextFile('s3a://<name-of-your-bucket>/test_rddoutput.txt')'` above.

Now we can see a folder on S3 containing the results

The screenshot shows the Amazon S3 console interface. At the top, the breadcrumb navigation indicates the path: Amazon S3 > Buckets > cap4770-2024summer > test_rddoutput.txt/. The main heading is 'test_rddoutput.txt/'. Below this, there are tabs for 'Objects' and 'Properties'. The 'Objects' tab is active, showing a list of objects. Above the list, there are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. The list of objects has columns for Name, Type, Last modified, Size, and Storage class. The objects listed are '_SUCCESS', 'part-00000', and 'part-00001', all with a storage class of 'Standard'.

Name	Type	Last modified	Size	Storage class
_SUCCESS	-	June 6, 2024, 20:31:07 (UTC-04:00)	0 B	Standard
part-00000	-	June 6, 2024, 20:31:06 (UTC-04:00)	203.0 B	Standard
part-00001	-	June 6, 2024, 20:31:06 (UTC-04:00)	217.0 B	Standard

Please go over the programming guide <https://spark.apache.org/docs/latest/rdd-programming-guide.html> for more operations and detailed explanations.

You can skip the "initializing spark shell" and "linking with spark" parts in the guide.

Experiment with your test.txt and see how it works.