

DS 7200 Distributed Computing

Lab: AWS Services I

Topic: Basics of AWS Identity and Access Management (IAM)

Last Updated: October 7, 2023

Learning Objectives:

- Demonstrate the creation of an S3 bucket and save a file
- Understand how to create an IAM role
- Understand how to create an IAM policy
- Understand how to spin up an EC2 instance
- Explain the purpose of an IAM role
- Explain the purpose of IAM policies
- Explain the purpose of an Amazon Resource Name (ARN)

Submission: Save all of your results and screenshots in a file (Word or PDF doc). Number the steps, keeping things clear and organized.

Total Points: 9

Link to AWS Management Console:

<https://aws.amazon.com/console/>

Assigned Reading

Policies and permissions in IAM (up to but excluding IAM permissions boundaries)

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

Roles

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.

Policies

A policy is an object in AWS that defines the permissions of an identity or resource.

Instructions

This exercise will have you working with various AWS services.

Specifically, you will:

1. Visit S3, create a bucket, and upload a file to it.
2. Create an IAM role and assign policy that allows you to write to and read from the bucket.
3. Spin up an EC2 instance using the role, and write a CLI command to show the contents of the file. Along the way, you will take screenshots of output to submit.

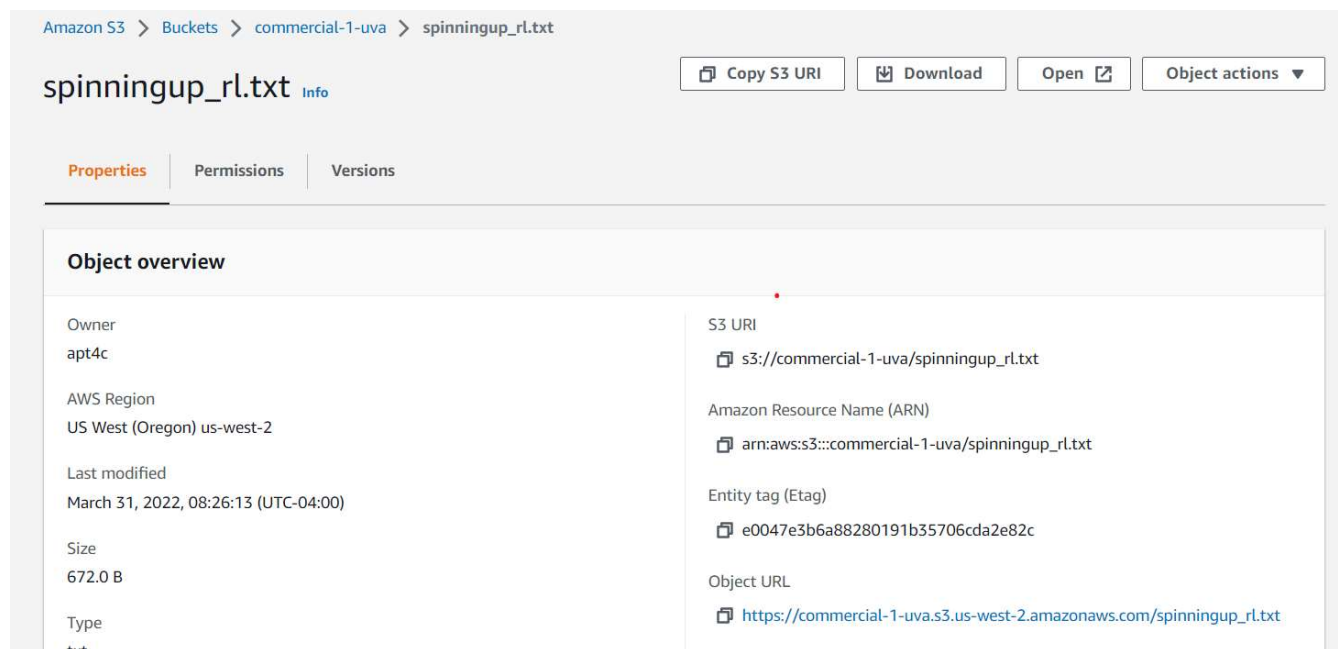
Follow the steps very carefully.

1. **(1 POINT)** Go to the **S3** service. Create a new bucket and upload the file: *spinningup_rl.txt*

FYI: This is a bootstrap file that installs python, some modules, and a repo.

Take a screenshot showing that the file is in the bucket.

2. **(1 POINT)** Take note of the S3 URI which is the path to the file on S3. You will use this in EC2 later to read from the file. Capture this URI in your results file.



3. Go to the **IAM** service. Create a role called *data_scientist_s3_role** where * can be other characters

4. Allow common use case: EC2

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.

Use case

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

Common use cases

- ☐ **EC2**
Allows EC2 instances to call AWS services on your behalf.
- ☐ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

5. Create a policy that can read from/write to an S3 bucket. Specifically it should have this access:

- GetObject (from Read). This grants permission to retrieve objects from S3.
- PutObject (from Write). This grants permission to add an object to a bucket.

Create policy

1 2 3

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor **JSON** [Import managed policy](#)

Expand all | Collapse all

▼ Select a service [Clone](#) [Remove](#)

▼ Service [close](#)

Select a service below

[Access Analyzer ?](#) [Elastic Transcoder ?](#) [Mobile Hub ?](#)

[Enter service manually](#)

You will want to provide permissions to a specific file. To do this, specify an Amazon Resource Name (ARN) for a specific S3 bucket and object. You will need to have at least one bucket with a file created in S3; please create one if needed. When I enter bucket name and object name, the ARN is auto-populated.

Add ARN(s)

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. [Learn more](#)

Specify ARN for object [List ARNs manually](#)

arn:aws:s3:::commercial-1-uva/spinningup_rl.txt

Bucket name *

commercial-1-uva
☐ Any

Object name *

spinningup_rl.txt
☐ Any

Cancel
Add

6. **(1 POINT)** Take a screenshot of the policy summary page that shows the created policy. You'll need to give the policy and name. Take note so you can find it for the next step. You can see mine is called *data_scientist_s3_policy*

[Policies](#) > [data_scientist_s3_policy](#)

Summary

Delete policy

Policy ARN arn:aws:iam::208637696492:policy/data_scientist_s3_policy

Description

Permissions

Policy usage

Tags

Policy versions

Access Advisor

Policy summary

{ } JSON

Edit policy

Q Filter

Service	Access level	Resource	Request condition
Allow (1 of 371 services) <a>Show remaining 370			
S3	Limited: Read, Write	ObjectPath string like spinningup_rl.txt, BucketName string like commercial-1-uva	None

7. Attach the policy to the role.

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions [Info](#)

Permissions policies (Selected 1/826) [Info](#)

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter.

< 1 2 3 4 5 6 7 ... 42 > [Settings](#)

<input type="checkbox"/>	Policy name ↗	Type ▼	Description
--------------------------	-------------------------------	------------------------	-------------

<input checked="" type="checkbox"/>	+ data_scientist_s3_policy	Custom...	
-------------------------------------	--	-----------	--

8. The final role creation step should look like this:

Role name

Enter a meaningful name to identify this role.

data_scientist_s3_role

Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters.

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

Step 1: Select trusted entities

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "sts:AssumeRole"  
8       ],  
9       "Principal": {  
10        "Service": [  
11          "ec2.amazonaws.com"  
12        ]  
13      }  
14    }  
15  ]  
16 }
```

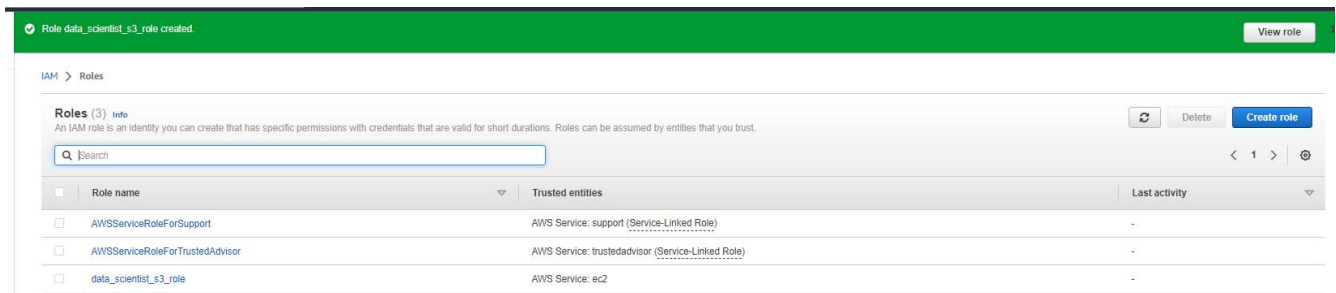
Step 2: Add permissions

Permissions policy summary

Policy name ↗	Type ▼	Attached as
data_scientist_s3_policy	Customer managed	Permissions policy

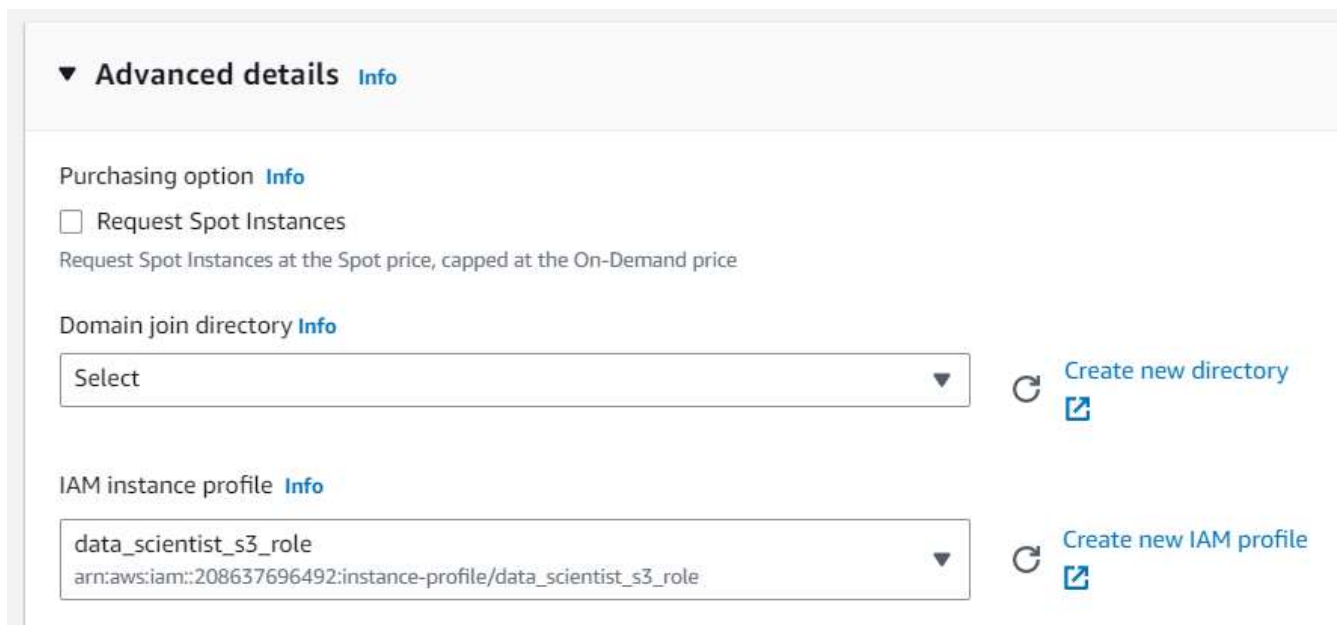
Notice the trusted entities (EC2) is pre-populated in JSON.

9. (1 POINT) Take a screenshot of the Roles page to show the new role.



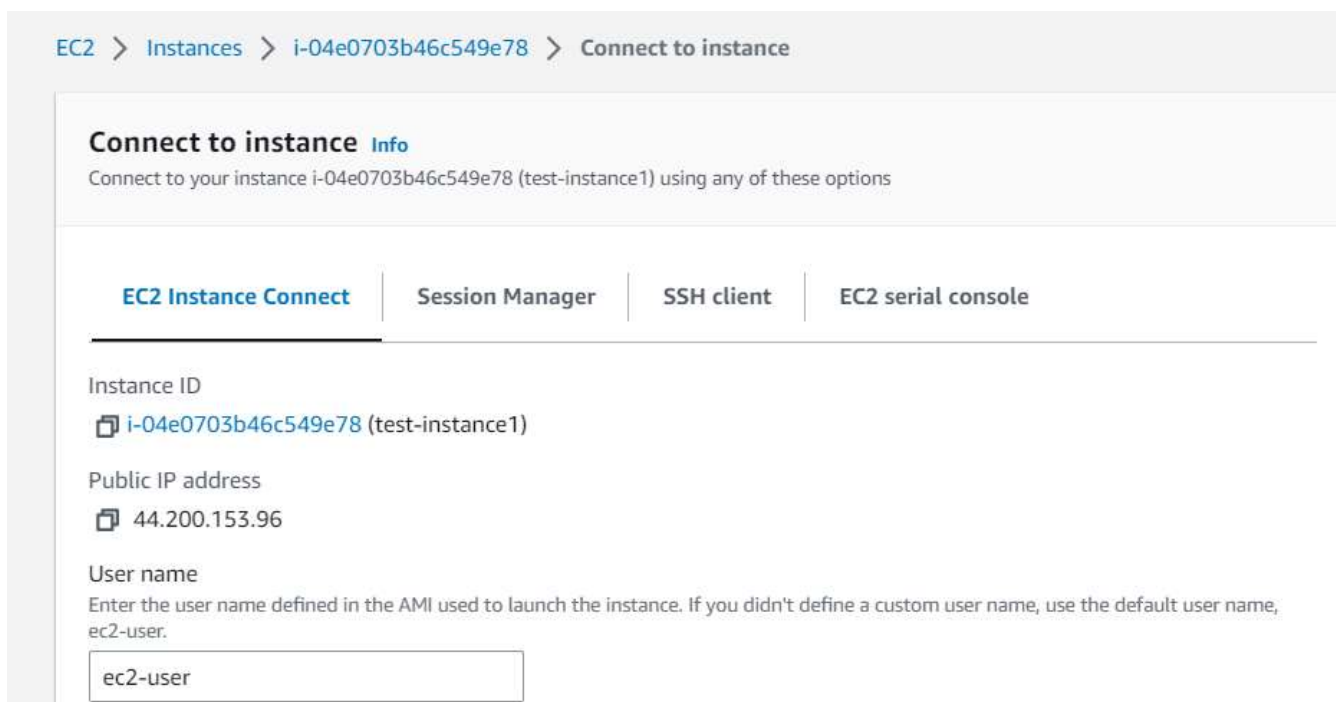
10. (1 POINT) Create an EC2 instance with these parameters:

- size: t2.micro (this is free on the AWS Free Tier)
- OS: Amazon Linux
- create or select a key pair
- under Advanced details, select the role you've created (THIS IS IMPORTANT)

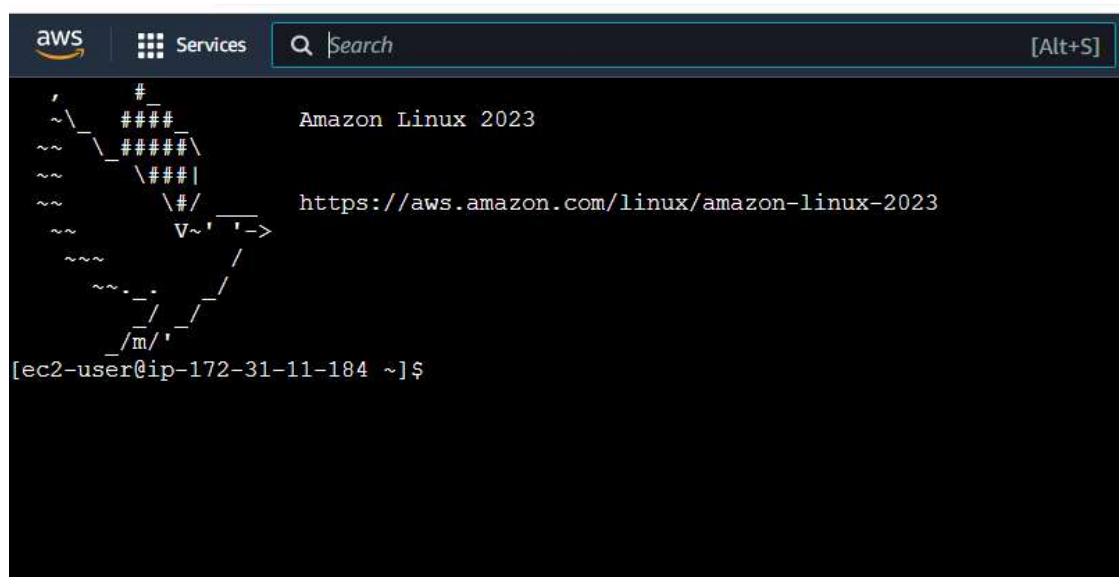


Take a screenshot showing the instance.

11. Launch the instance
 12. Connect to the instance using EC2 Instance Connect
- Note there are other options like using an SSH client



13. (1 POINT) You will see the landing page of your EC2 instance. Take a screenshot.

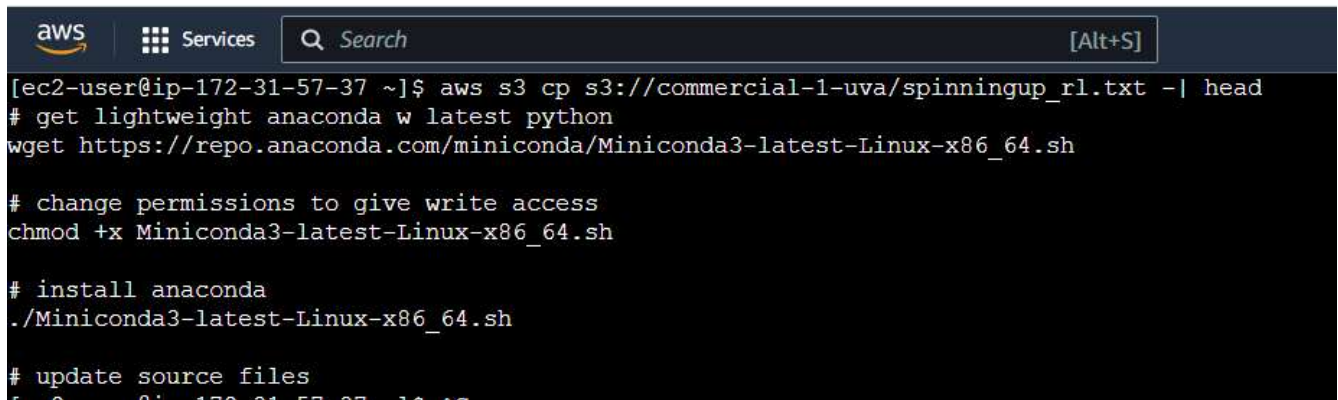


14. (1 POINT)
Run the

command following prompt \$ at the terminal to show the top lines of the file you've uploaded to S3. This uses the aws s3 CLI tool to copy the file contents and pipe to a file that shows the head. The portion `s3://commercial-1-uva/spinningup_rl.txt` is my S3 URI.

```
[ec2-user@ip-172-31-57-37 ~]$ aws s3 cp s3://commercial-1-uva/spinningup_rl.txt -| head
```

Take a screenshot of the command and the output, which should look like this:

A screenshot of an AWS Management Console terminal window. The top bar shows the AWS logo, a 'Services' menu, a search bar, and a '[Alt+S]' shortcut. The terminal content shows a user running a series of commands to install Anaconda on an EC2 instance. The commands include using 'aws s3 cp' to download a script, 'wget' to fetch the script from a URL, 'chmod' to make the script executable, and finally running the script to install Anaconda. The prompt indicates the user is 'ec2-user' on an instance with IP 'ip-172-31-57-37' in the '~' directory.

```
aws Services Search [Alt+S]
[ec2-user@ip-172-31-57-37 ~]$ aws s3 cp s3://commercial-1-uva/spinningup_rl.txt -| head
# get lightweight anaconda w latest python
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

# change permissions to give write access
chmod +x Miniconda3-latest-Linux-x86_64.sh

# install anaconda
./Miniconda3-latest-Linux-x86_64.sh

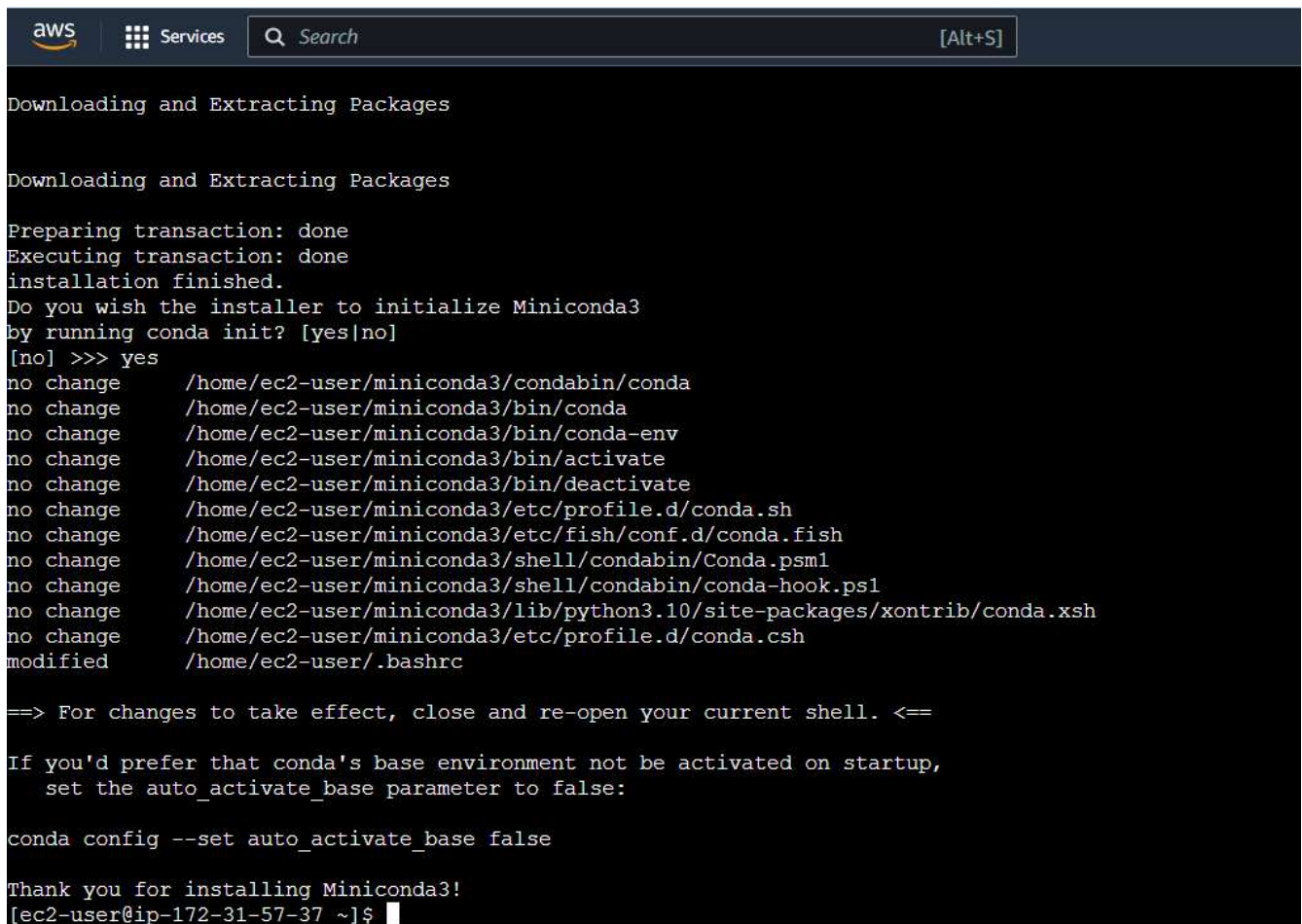
# update source files
```

15. Run the top lines of code at the command line to install Anaconda. You might copy/paste. They are reproduced here. You'll need to agree and press enter a few times to complete installation. When it completes, you'll see the screenshot below.

```
# get lightweight anaconda w latest python
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

# change permissions to give write access
chmod +x Miniconda3-latest-Linux-x86_64.sh

# install anaconda
./Miniconda3-latest-Linux-x86_64.sh
```


The screenshot shows the AWS Management Console interface at the top with the 'aws' logo, 'Services' menu, and a search bar. Below this is a terminal window titled 'Downloading and Extracting Packages'. The terminal output shows the Miniconda3 installer progress, including file downloads and the completion of the installation. It prompts the user to initialize Miniconda3, which is confirmed with 'yes'. A list of files is shown as 'no change' or 'modified'. A final instruction to run 'conda config --set auto_activate_base false' is given. The terminal ends with a thank you message and the prompt '[ec2-user@ip-172-31-57-37 ~]\$'.

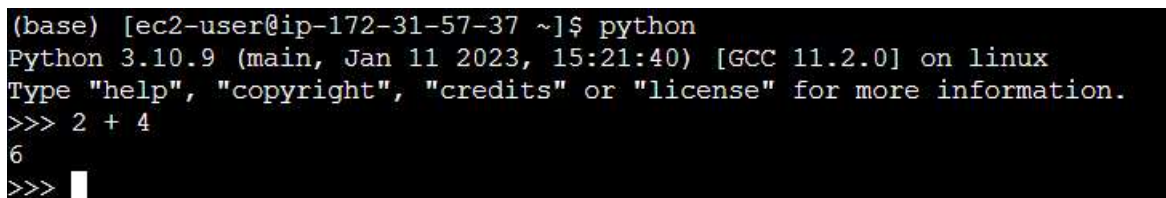
16. Run this line to update your session:

```
source ~/.bashrc
```

17. **(1 POINT)** Next, you'll show that python is installed by running at the command line:

```
$ python
```

This should launch the latest Python. It's now available for the lifetime of this instance. Take a screenshot.

The screenshot shows a terminal window with the prompt '(base) [ec2-user@ip-172-31-57-37 ~]\$ python'. The output shows 'Python 3.10.9 (main, Jan 11 2023, 15:21:40) [GCC 11.2.0] on linux'. It then prompts 'Type "help", "copyright", "credits" or "license" for more information.' followed by a prompt '>>>'. The user enters '2 + 4' and the output is '6'. The prompt '>>>' is shown again.

18. **(1 POINT)** We are finished with this session and instance. Revisit the instances and terminate your EC2 instance. Take a screenshot that shows your instance has been terminated.

Successfully terminated i-063c675f46a1489ca

Instances (2) Info

Find instance by attribute or tag (case-sensitive)

Refresh

Connect

Instance state

Actions

Launch instances

< 1 > ⚙

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
<input type="checkbox"/>	test-instance1	i-063c675f46a1489ca	Terminated	t2.micro	-	No alarms	us-east-1e	-	-

If you were able to complete this assignment, I commend you! There's a lot to learn when getting started with AWS.