DS 7200 Distributed Computing

Lab: AWS Services I

Topic: Basics of AWS Identity and Access Management (IAM)
Last Updated: October 26, 2024

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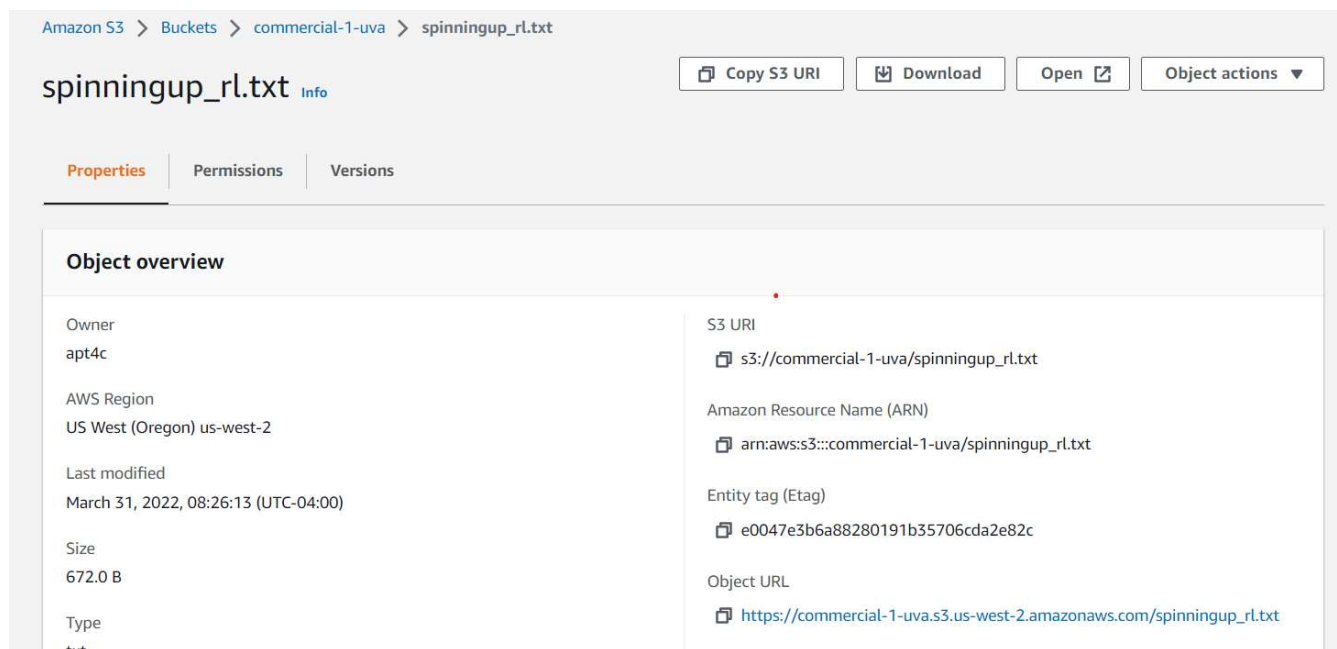
3. Go to the **IAM** service**.** Create a role called *data_scientist_s3_role**
where * can be other characters

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   **Select a service below**
close

🔍 Find a service

Enter service manually

Access Analyzer ⑦   Elastic Transcoder ⑦   Mobile Hub ⑦

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.

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*

7. Attach the policy to the role.

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

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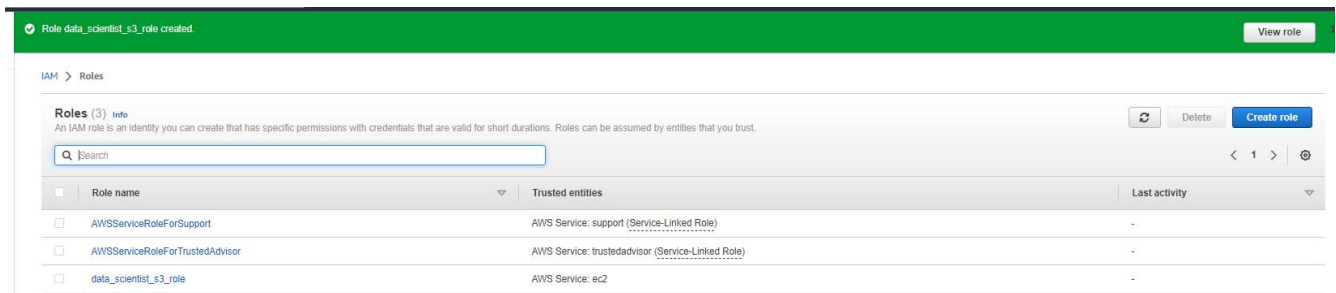
10. **(1 POINT)** Create an **EC2** instance with these parameters:

-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)
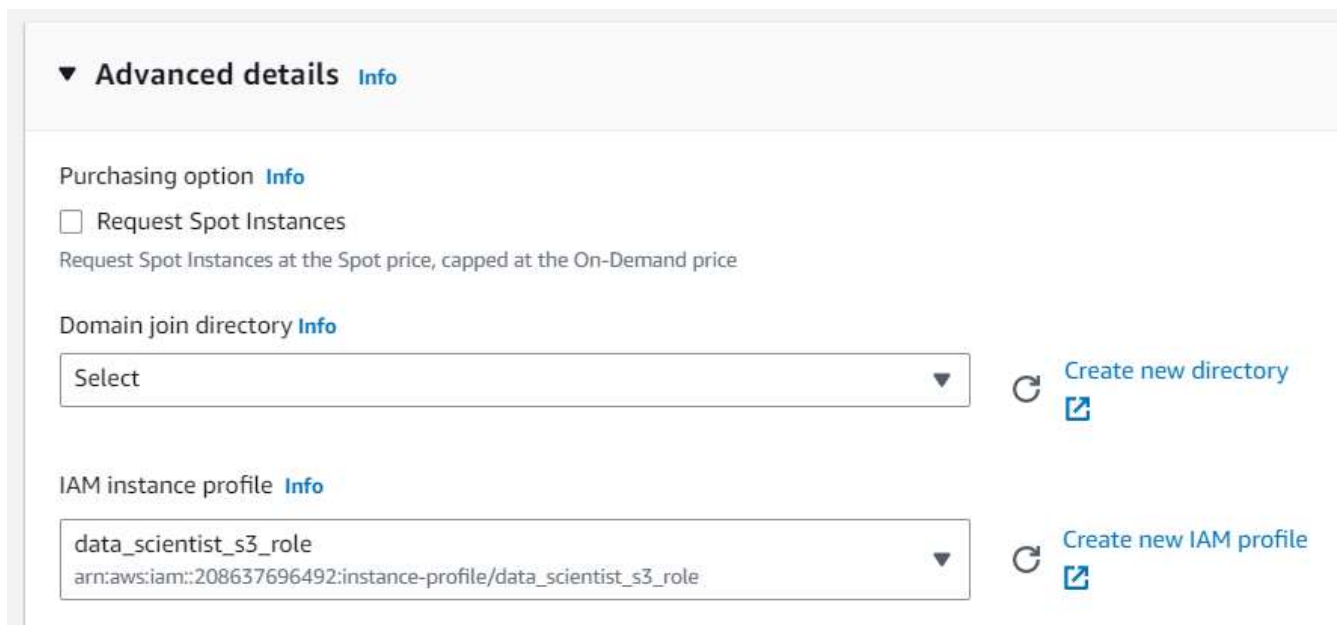
Take a screenshot showing the instance.

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.
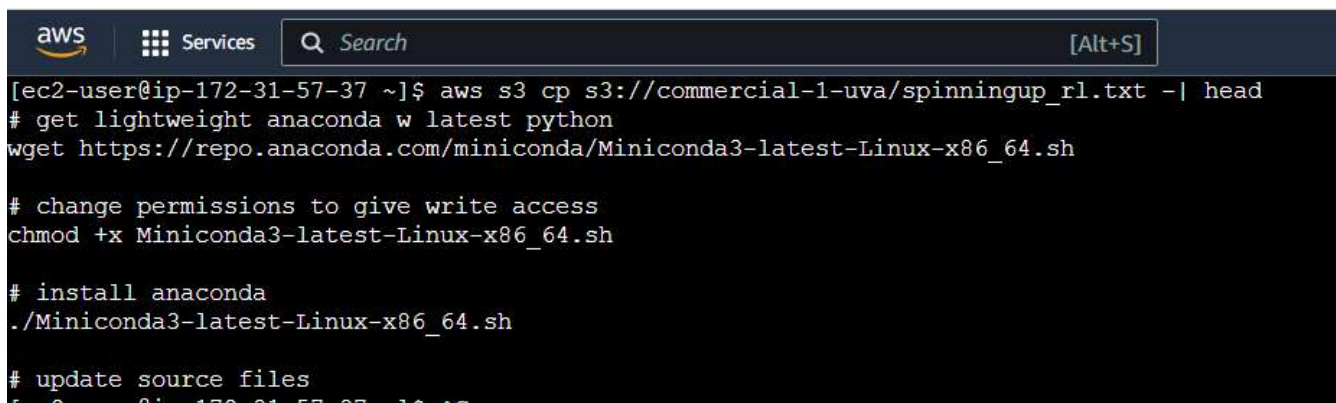
14. **(1 POINT)** Run the command following prompt $ at the terminal to show the top lines of the file you've uploaded to S3. This uses the aws s3 CLI tool to copy the file contents and pipe to a file that shows the head. The portion s3://commercial-1-uva/spinningup_rl.txt is my S3 URI.

[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:

```
[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

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.

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.

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