

DS 7200 Distributed Computing

Lab: AWS Services I

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

Last Updated: October 28, 2025

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.

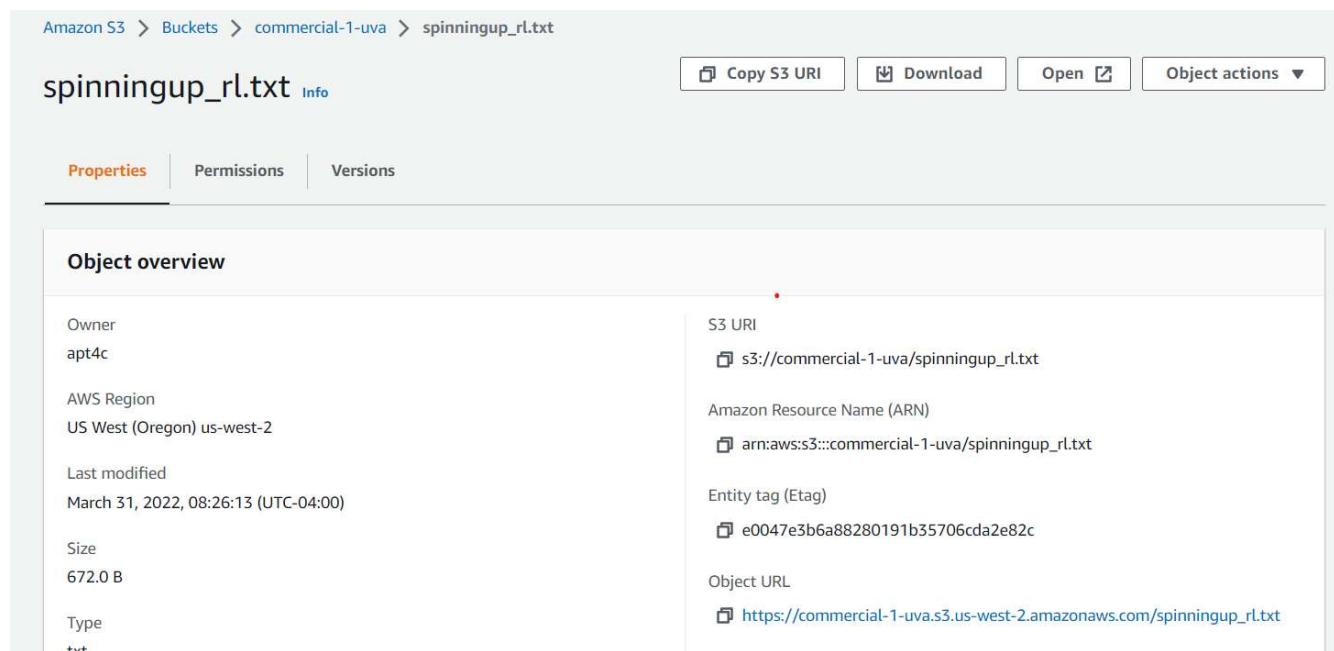
Follow the steps very carefully.

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.



The screenshot shows the AWS S3 Object Overview page for the file 'spinningup_rl.txt' located in the 'commercial-1-uva' bucket. The top navigation bar shows 'Amazon S3 > Buckets > commercial-1-uva > spinningup_rl.txt'. Below the navigation are buttons for 'Copy S3 URI', 'Download', 'Open', and 'Object actions'. The main content area is divided into two columns: 'Properties' and 'Object overview'. The 'Properties' column lists the file's details: Owner (apt4c), AWS Region (US West (Oregon) us-west-2), Last modified (March 31, 2022, 08:26:13 (UTC-04:00)), Size (672.0 B), and Type (txt). The 'Object overview' column provides the S3 URI (s3://commercial-1-uva/spinningup_rl.txt), Amazon Resource Name (ARN) (arn:aws:s3:::commercial-1-uva/spinningup_rl.txt), Entity tag (Etag) (e0047e3b6a88280191b35706cda2e82c), and Object URL (https://commercial-1-uva.s3.us-west-2.amazonaws.com/spinningup_rl.txt).

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

<input checked="" type="radio"/> AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.	<input type="radio"/> AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.	<input type="radio"/> Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
<input type="radio"/> SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.	<input type="radio"/> 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](#)

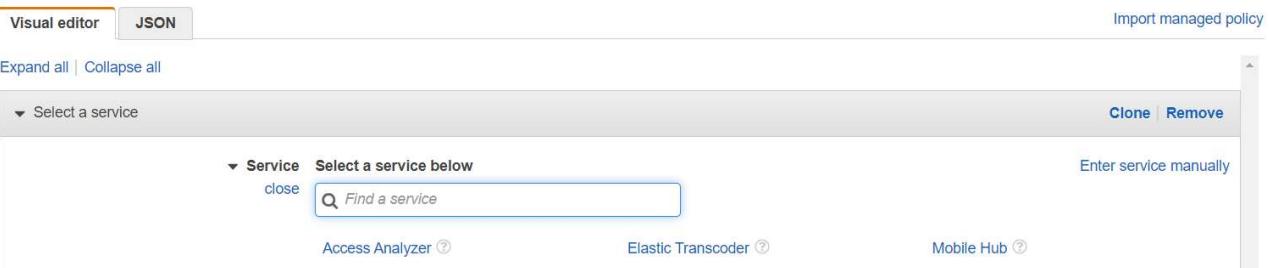
[Clone](#) | [Remove](#)

[Select a service](#) [Enter service manually](#)

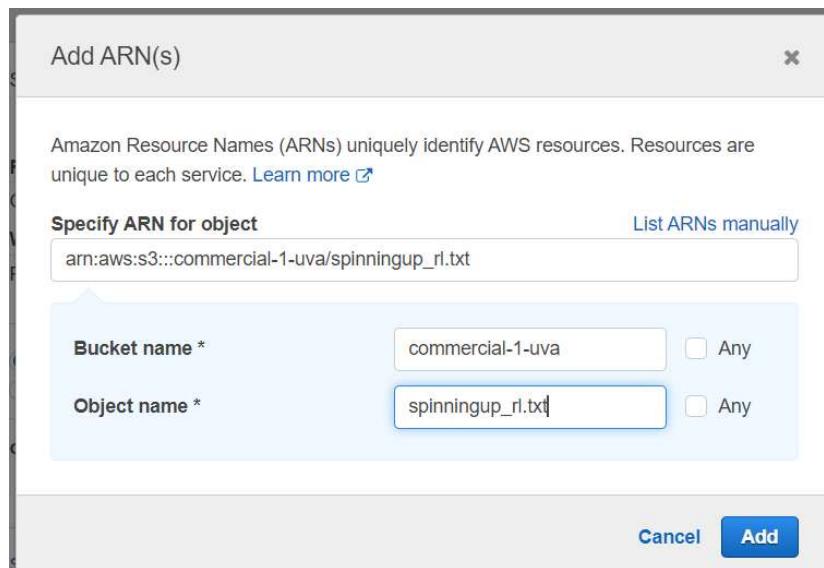
[Service](#) [Select a service below](#)

[close](#)

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



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*

Policies > *data_scientist_s3_policy*

Summary Delete policy

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

Description

Permissions Policy usage Tags Policy versions Access Advisor

Policy summary { } JSON Edit policy ?

Filter

Service	Access level	Resource	Request condition
S3	Limited: Read, Write	ObjectPath string like spinningup_rl.txt, BucketName string like commercial-1-uva	None

7. Attach the policy to the 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.

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.

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

Description
Add a short explanation for this role.

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.

The screenshot shows the AWS IAM Roles page. At the top, a green banner indicates "Role data_scientist_s3_role created." Below the banner, the page title is "IAM > Roles". A sub-header "Roles (3) Info" is followed by a search bar and a "Create role" button. A table lists three roles: "AWS ServiceRoleForSupport" (AWS Service: support), "AWS ServiceRoleForTrustedAdvisor" (AWS Service: trustedadvisor), and "data_scientist_s3_role" (AWS Service: ec2). The table includes columns for "Role name", "Trusted entities", and "Last activity".

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

- size: t3.micro (this is free on the AWS Free Tier)
- OS: Amazon Linux
- create or select a key pair
- under Network settings, select Allow HTTPS traffic from the internet. (THIS WILL ENABLE INSTANCE CONNECT)

The screenshot shows the "Network settings" section of the EC2 instance creation wizard. It includes fields for "Network" (vpc-f19af88c), "Subnet" (No preference), and "Auto-assign public IP" (Enable). A note states "Additional charges apply when outside of free tier allowance". Under "Firewall (security groups)", it says "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." Two options are shown: "Create security group" (selected) and "Select existing security group". Below this, a note says "We'll create a new security group called 'launch-wizard-31' with the following rules:" followed by two checked items: "Allow SSH traffic from Anywhere" and "Allow HTTPS traffic from the internet".

-under Advanced details > IAM instance profile, select the role you've created (THIS IS IMPORTANT)

The screenshot shows the 'Advanced details' section of an AWS EC2 instance configuration. It includes fields for purchasing options (Request Spot Instances), domain join directory (Select dropdown), and IAM instance profile (dropdown showing 'data_scientist_s3_role').

Purchasing option [Info](#)
 Request Spot Instances
Request Spot Instances at the Spot price, capped at the On-Demand price

Domain join directory [Info](#)
Select [Create new directory](#)

IAM instance profile [Info](#)
data_scientist_s3_role [arn:aws:iam::208637696492:instance-profile/data_scientist_s3_role](#) [Create new IAM profile](#)

Take a screenshot showing the instance.

11. Launch the instance

12. Connect to the instance.

There are several options for connection including:

-EC2 Instance Connect (through the AWS Management Console)

-SSH client (on a Mac you can use Terminal; on a Windows machine you can use PuTTY)

Connect to instance Info

Connect to your instance i-04e0703b46c549e78 (test-instance1) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

[i-04e0703b46c549e78 \(test-instance1\)](#)

Public IP address

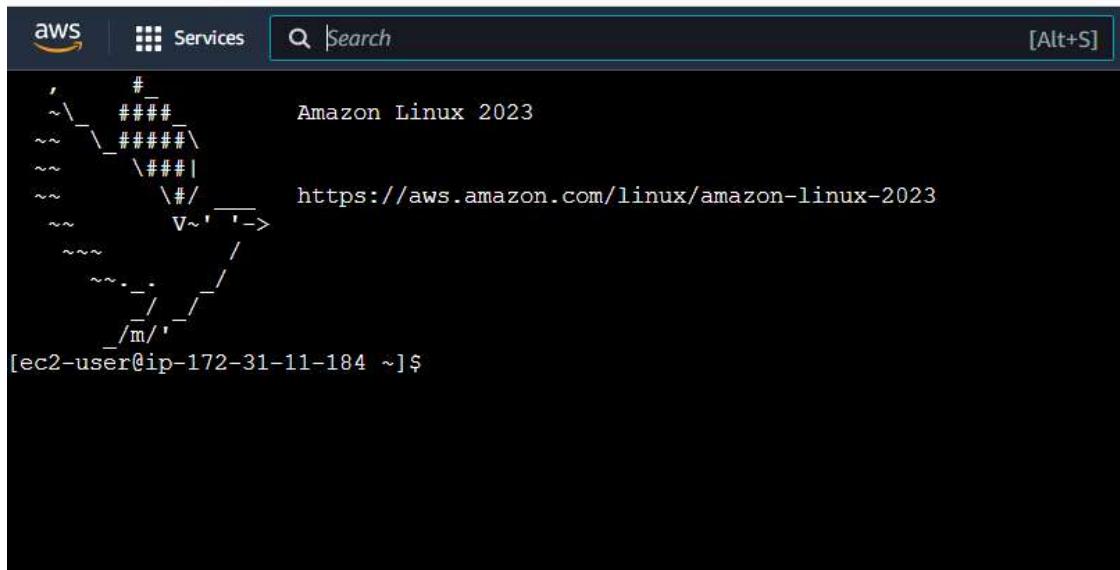
44.200.153.96

User name

Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

ec2-user

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

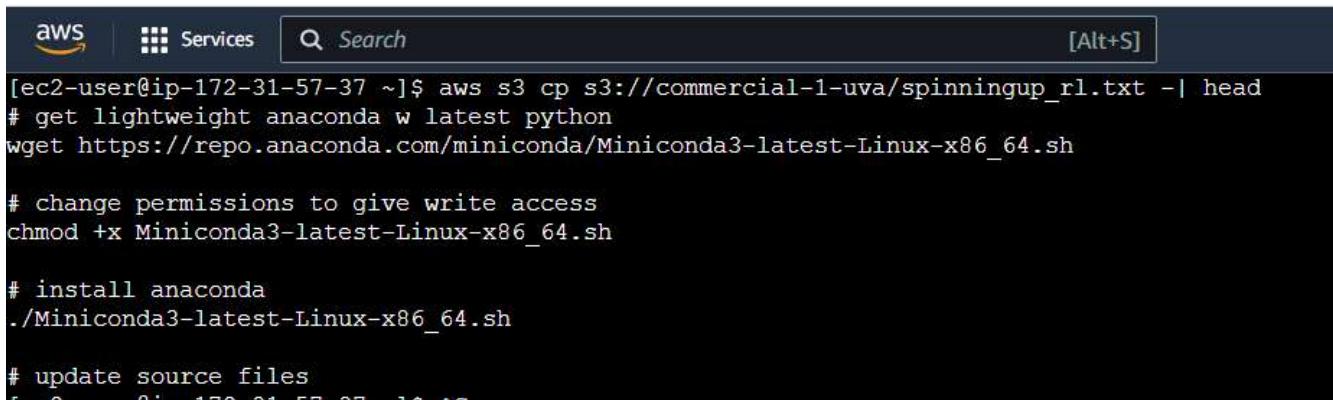


The screenshot shows a terminal window with the AWS Lambda interface. The top bar includes the AWS logo, a services menu, a search bar containing 'Search [Alt+S]', and a user icon. The main area displays a command-line session. The output is a stylized ASCII-art logo for 'Amazon Linux 2023' followed by its URL: <https://aws.amazon.com/linux/amazon-linux-2023>. The prompt at the bottom is '[ec2-user@ip-172-31-11-184 ~]\$'.

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 Lambda function editor. The top navigation bar includes the AWS logo, 'Services' (with a grid icon), a search bar containing 'Search', and a '[Alt+S]' keybinding. The main area shows a terminal window with the following command-line session:

```
[ec2-user@ip-172-31-57-37 ~]$ aws s3 cp s3://commercial-1-uva/spinningup_r1.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
[ec2-user@ip-172-31-57-37 ~]
```

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
```

```
aws | Services | Search | [Alt+S]

Downloading and Extracting Packages

Downloading and Extracting Packages

Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
no change    /home/ec2-user/miniconda3/condabin/conda
no change    /home/ec2-user/miniconda3/bin/conda
no change    /home/ec2-user/miniconda3/bin/conda-env
no change    /home/ec2-user/miniconda3/bin/activate
no change    /home/ec2-user/miniconda3/bin/deactivate
no change    /home/ec2-user/miniconda3/etc/profile.d/conda.sh
no change    /home/ec2-user/miniconda3/etc/fish/conf.d/conda.fish
no change    /home/ec2-user/miniconda3/shell/condabin/Conda.ps1
no change    /home/ec2-user/miniconda3/shell/condabin/conda-hook.ps1
no change    /home/ec2-user/miniconda3/lib/python3.10/site-packages/xontrib/conda.xsh
no change    /home/ec2-user/miniconda3/etc/profile.d/conda.csh
modified     /home/ec2-user/.bashrc

=> For changes to take effect, close and re-open your current shell. <=

If you'd prefer that conda's base environment not be activated on startup,
  set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniconda3!
[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.

```
(base) [ec2-user@ip-172-31-57-37 ~]$ python
Python 3.10.9 (main, Jan 11 2023, 15:21:40) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 2 + 4
6
>>>
```

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										
<input type="text"/> Find instance by attribute or tag (case-sensitive)										
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.