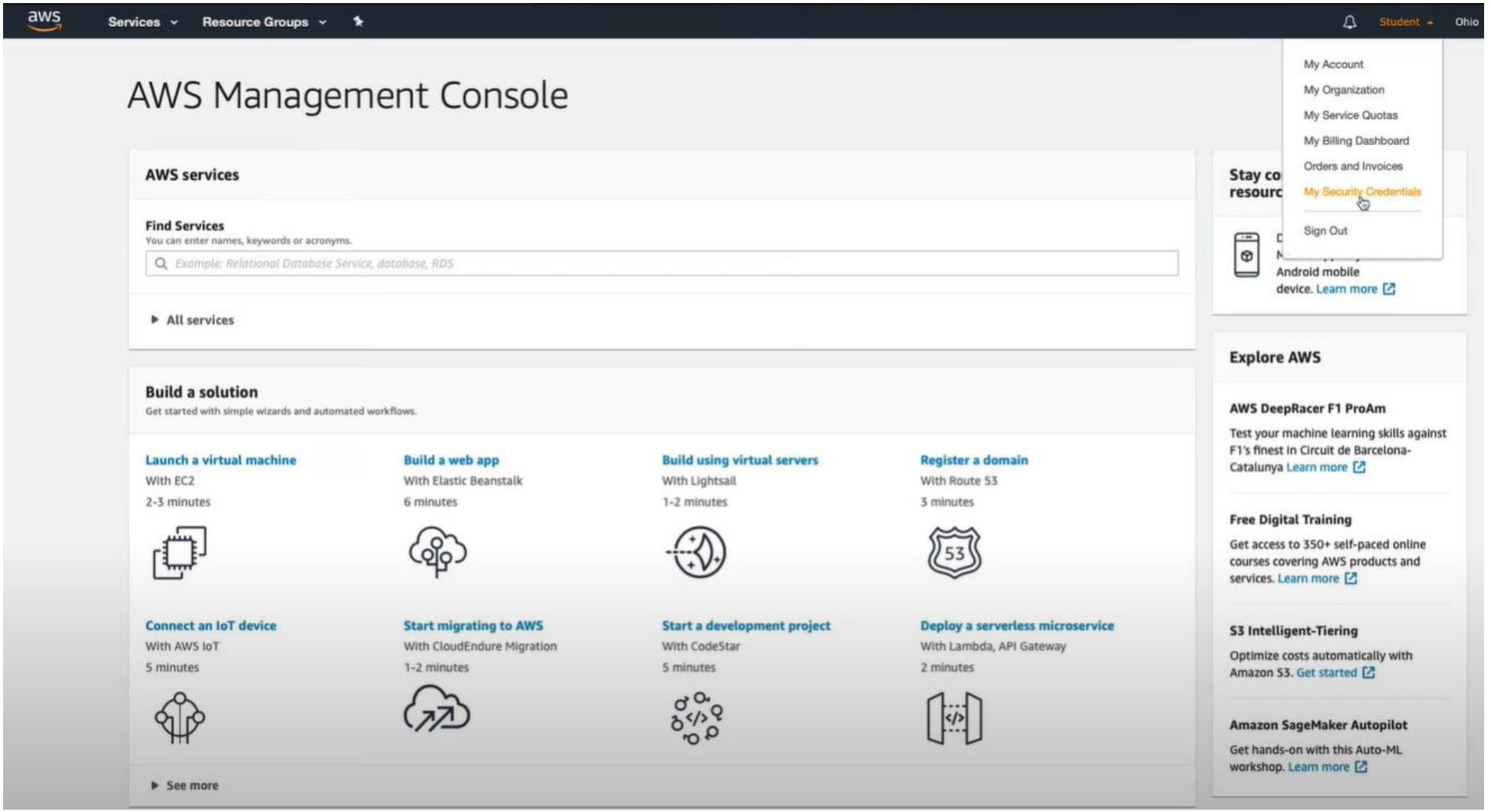
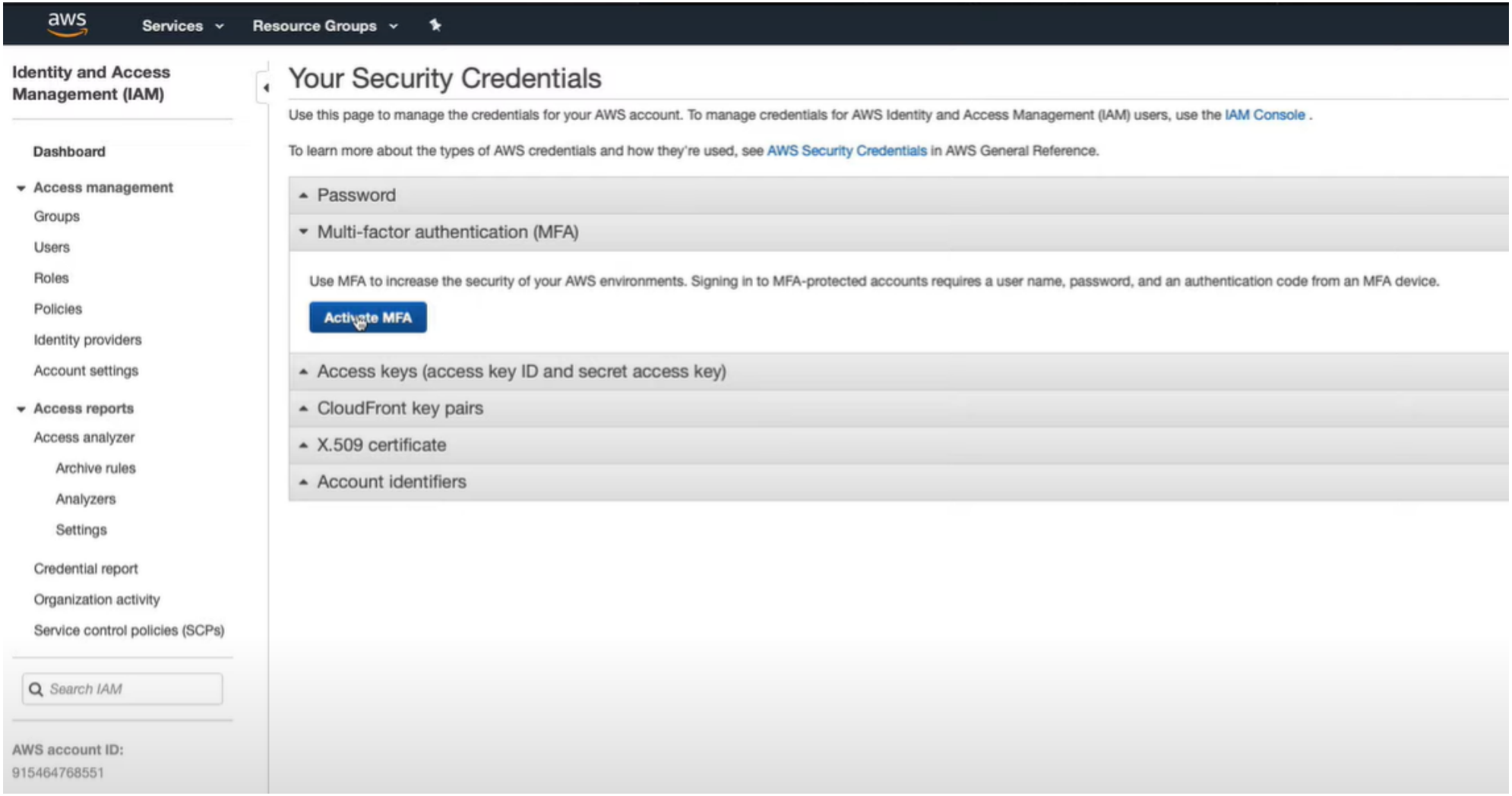
Securing Root User With MFA



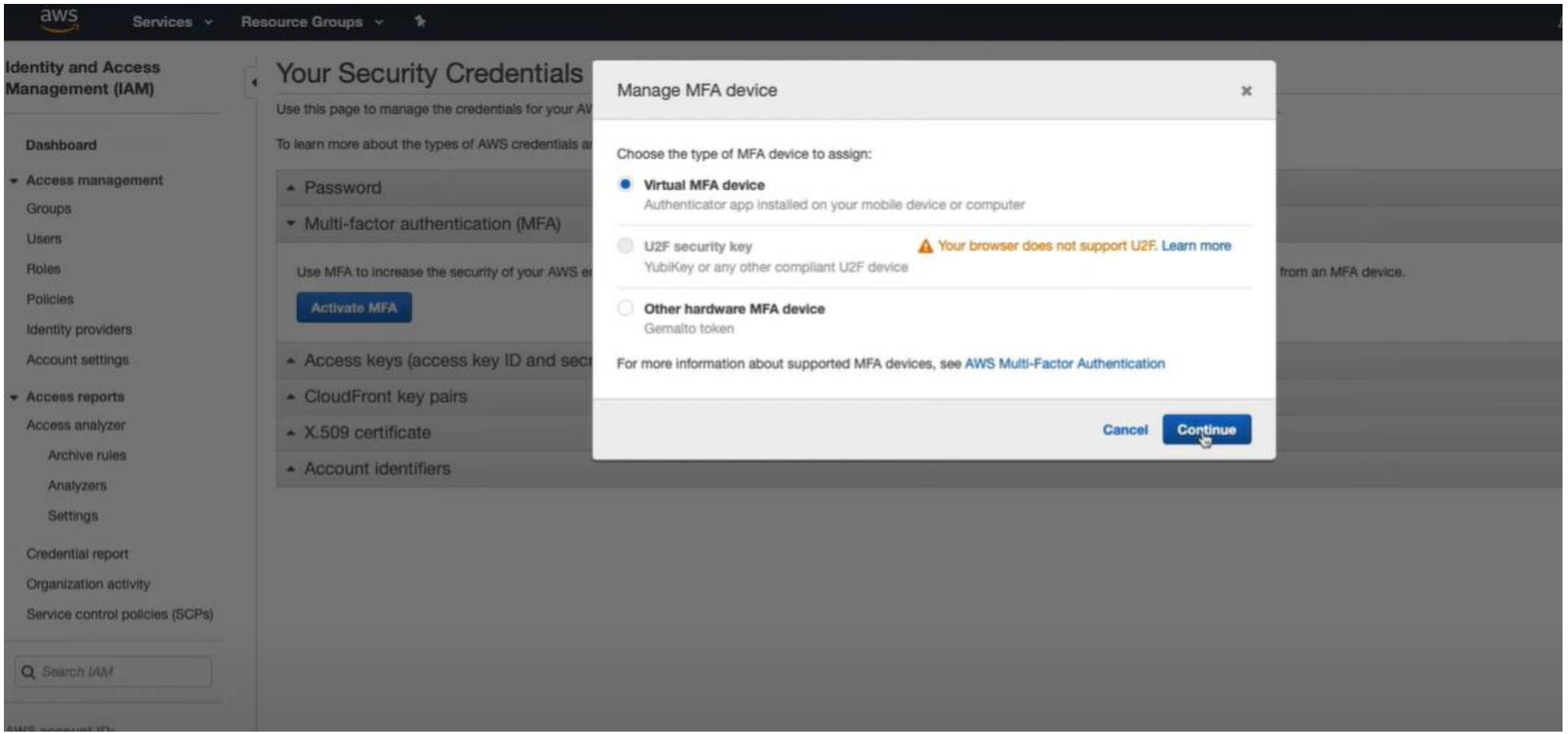
Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Securing Root User With MFA



Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

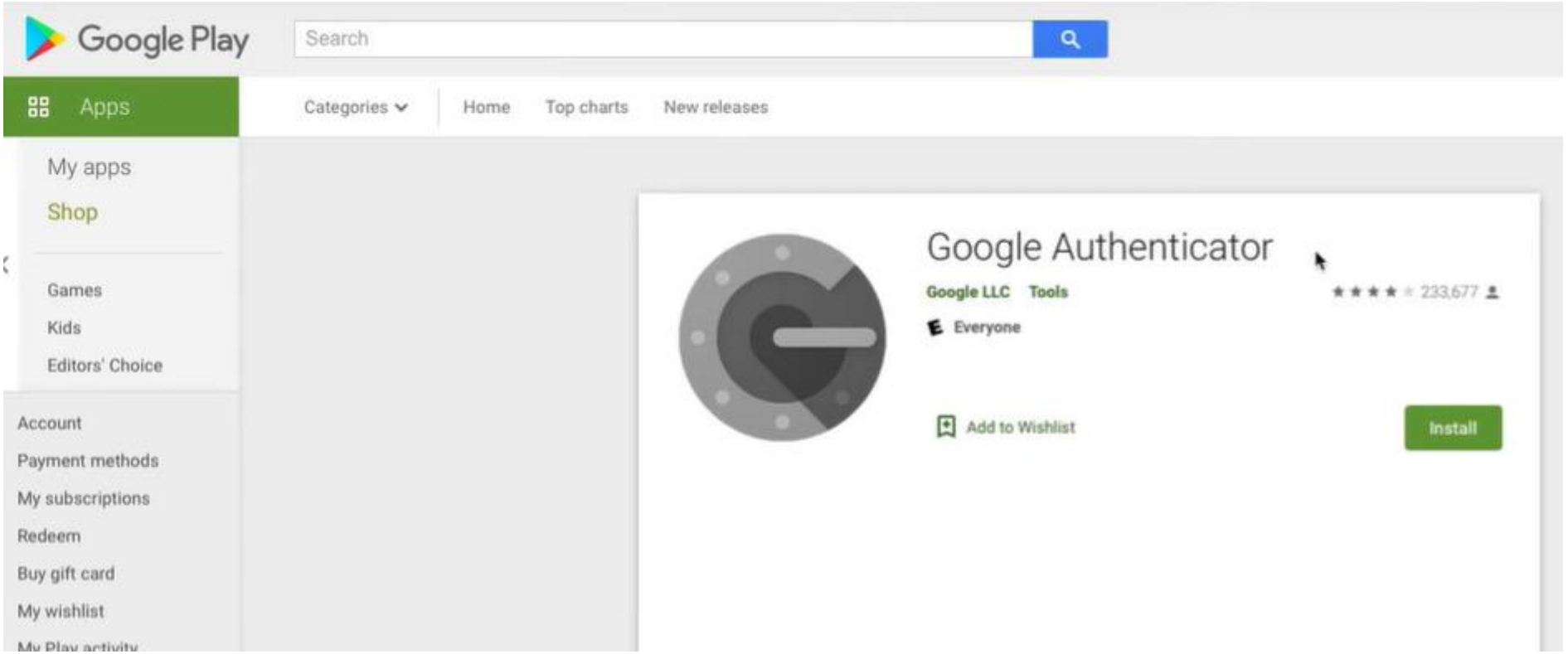
Securing Root User With MFA



Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

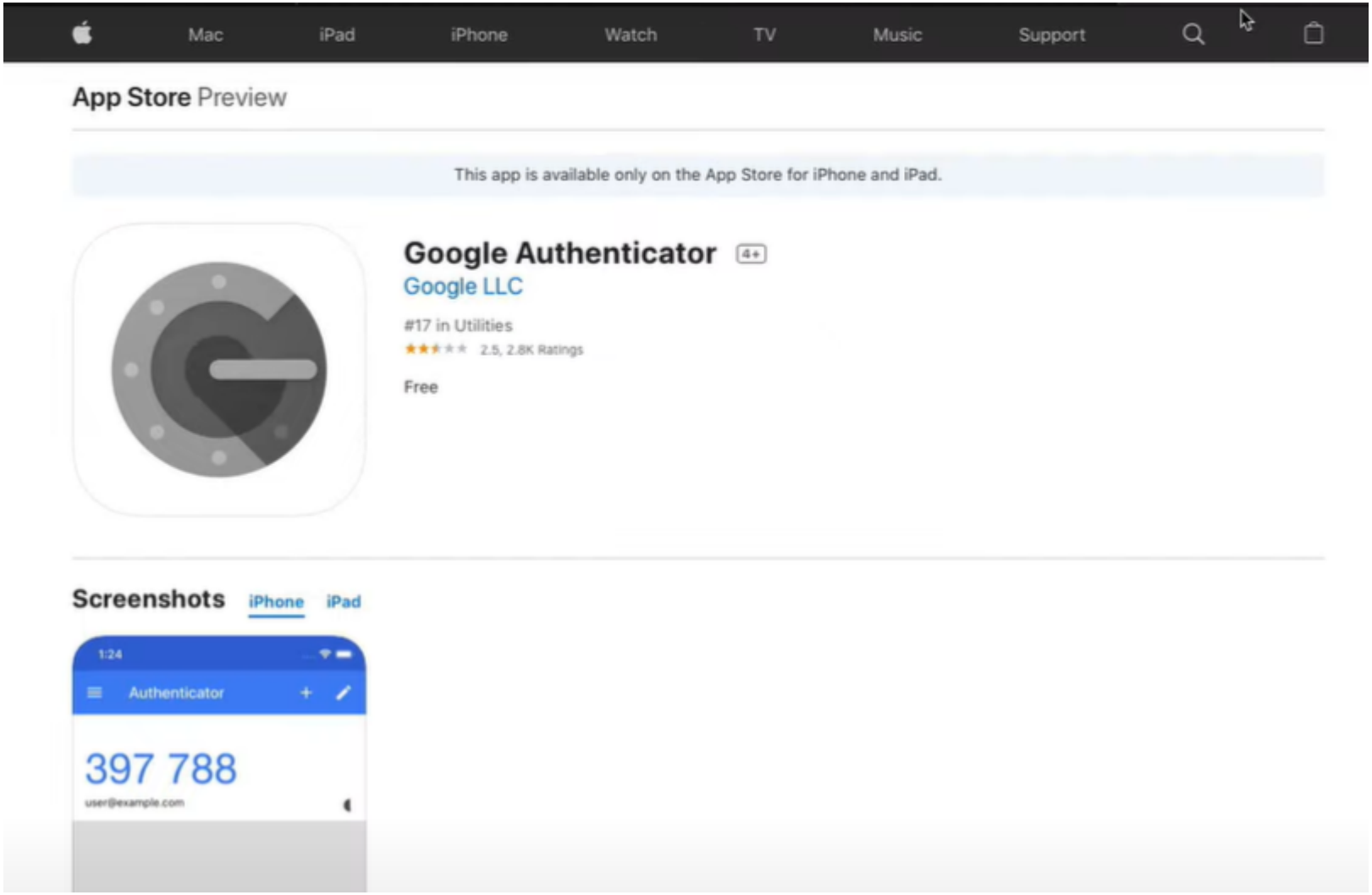
Securing Root User With MFA

Install Google Authenticator app.



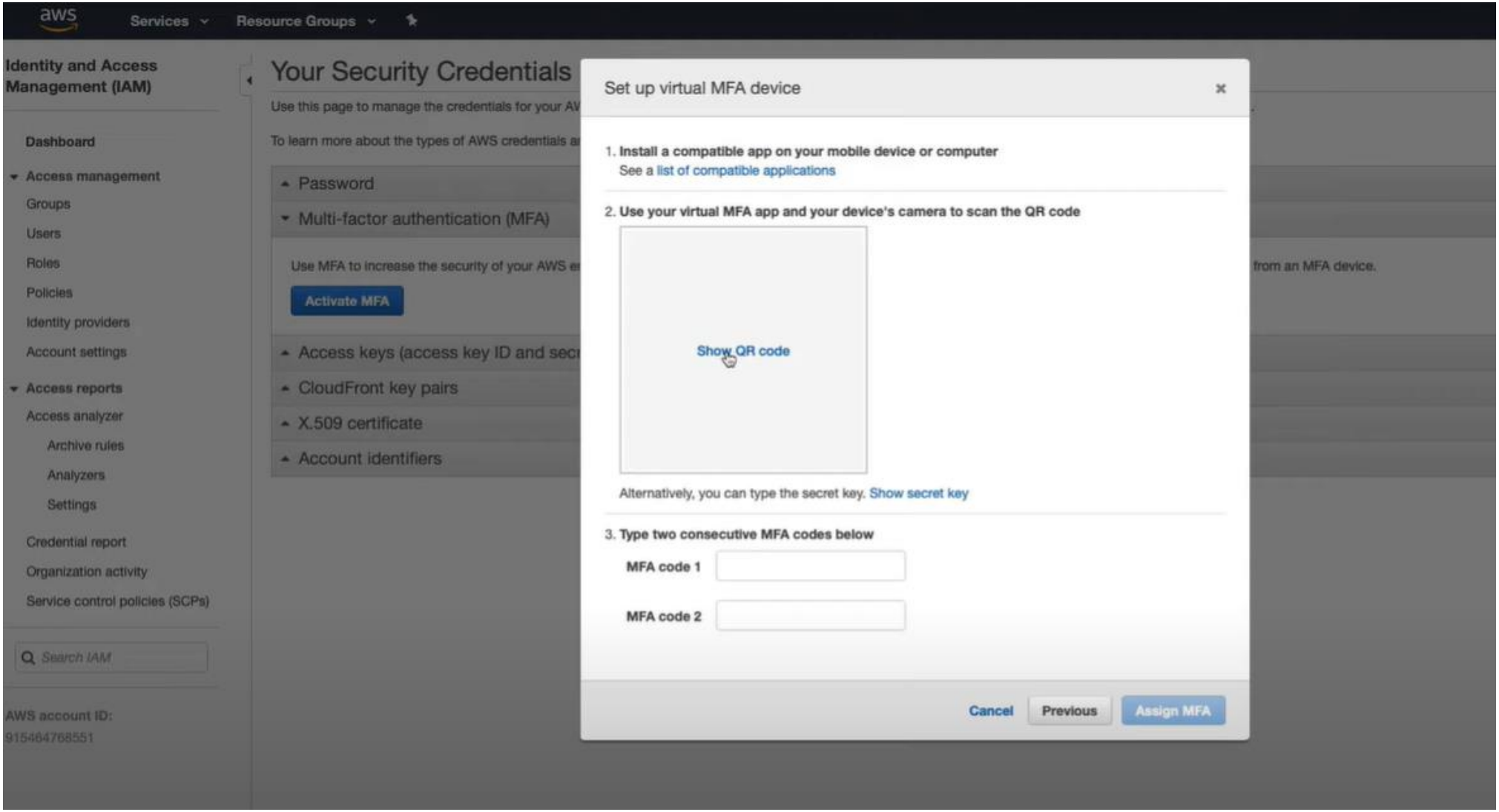
Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Securing Root User With MFA



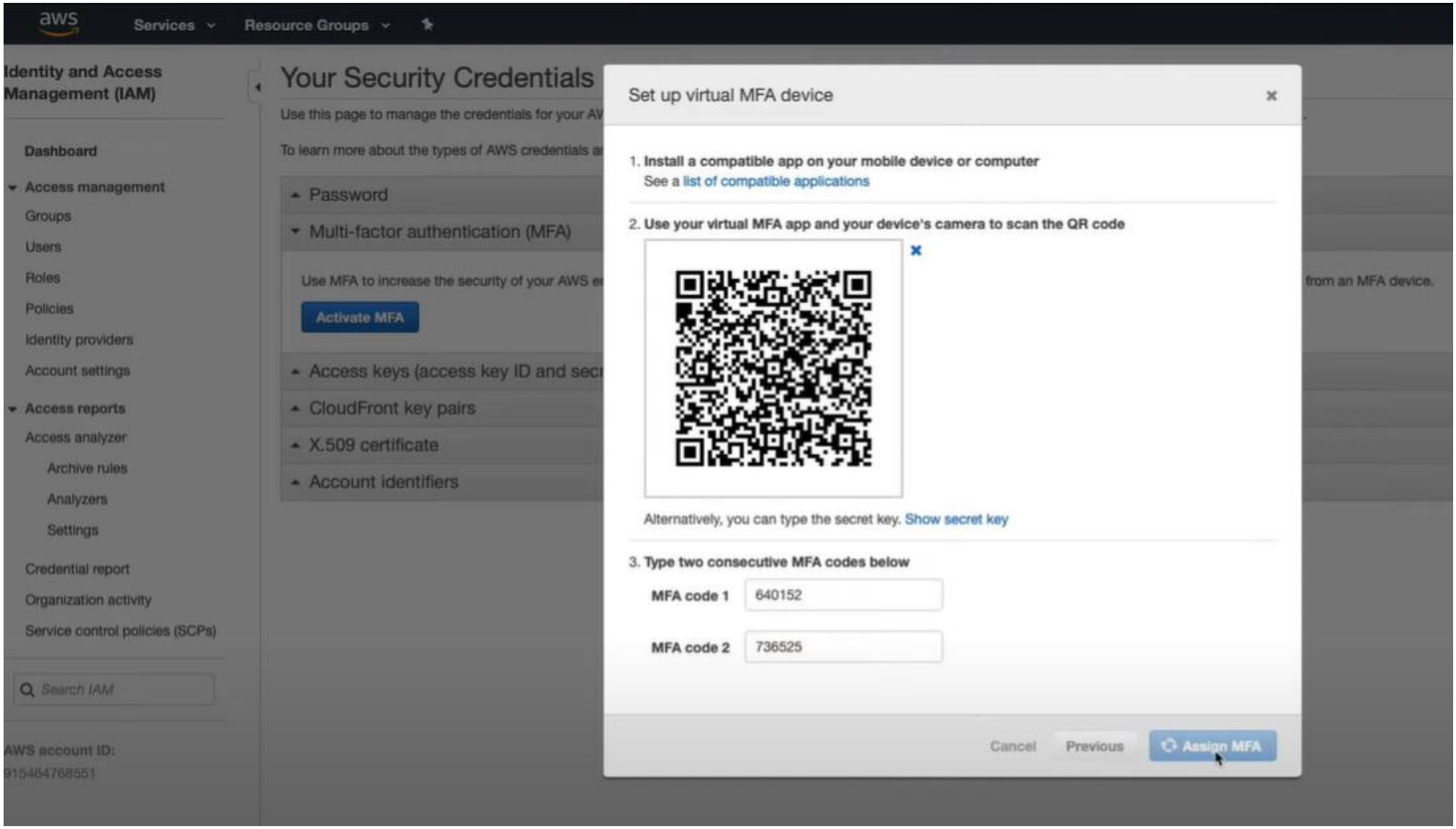
Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Securing Root User With MFA



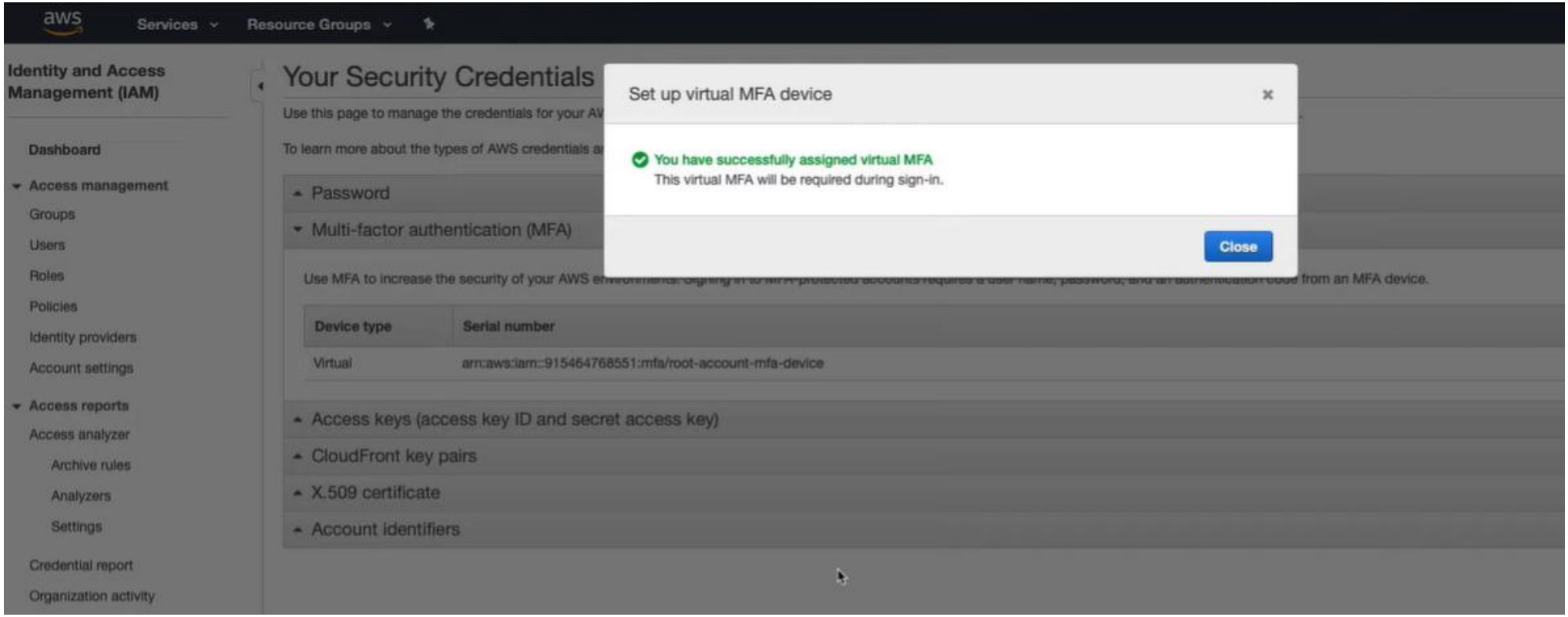
Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Securing Root User With MFA



Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Securing Root User With MFA



Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Exercise: Creating A Manager IAM User With Built-in Policies

In this exercise, you will create an IAM user for a manager, and apply built-in IAM policies.

Instructions

Create an IAM user

Name the user "manager"

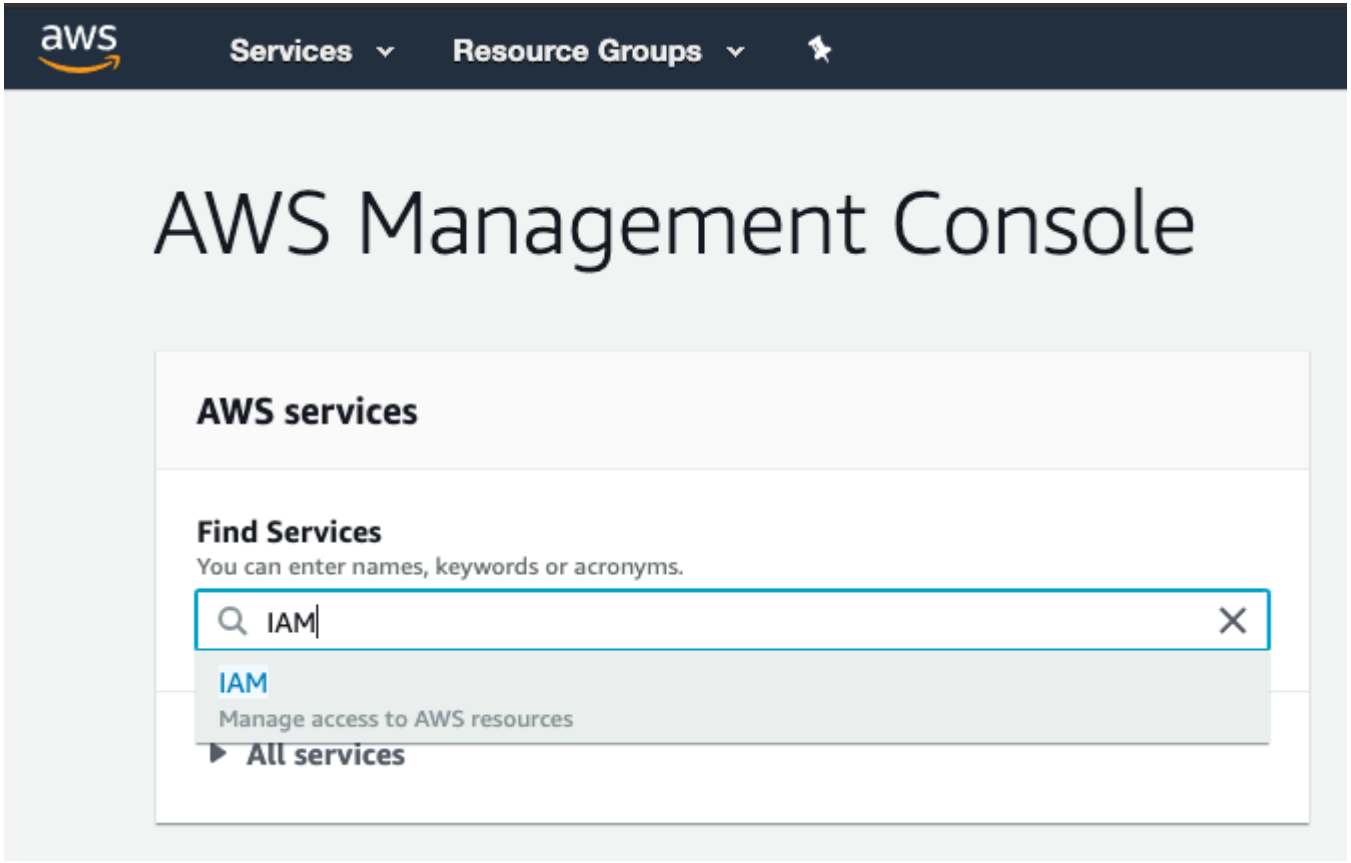
Provide this user with:

- IAM Full Access permissions
- S3 Full Access permissions
- EC2 Full Access permissions

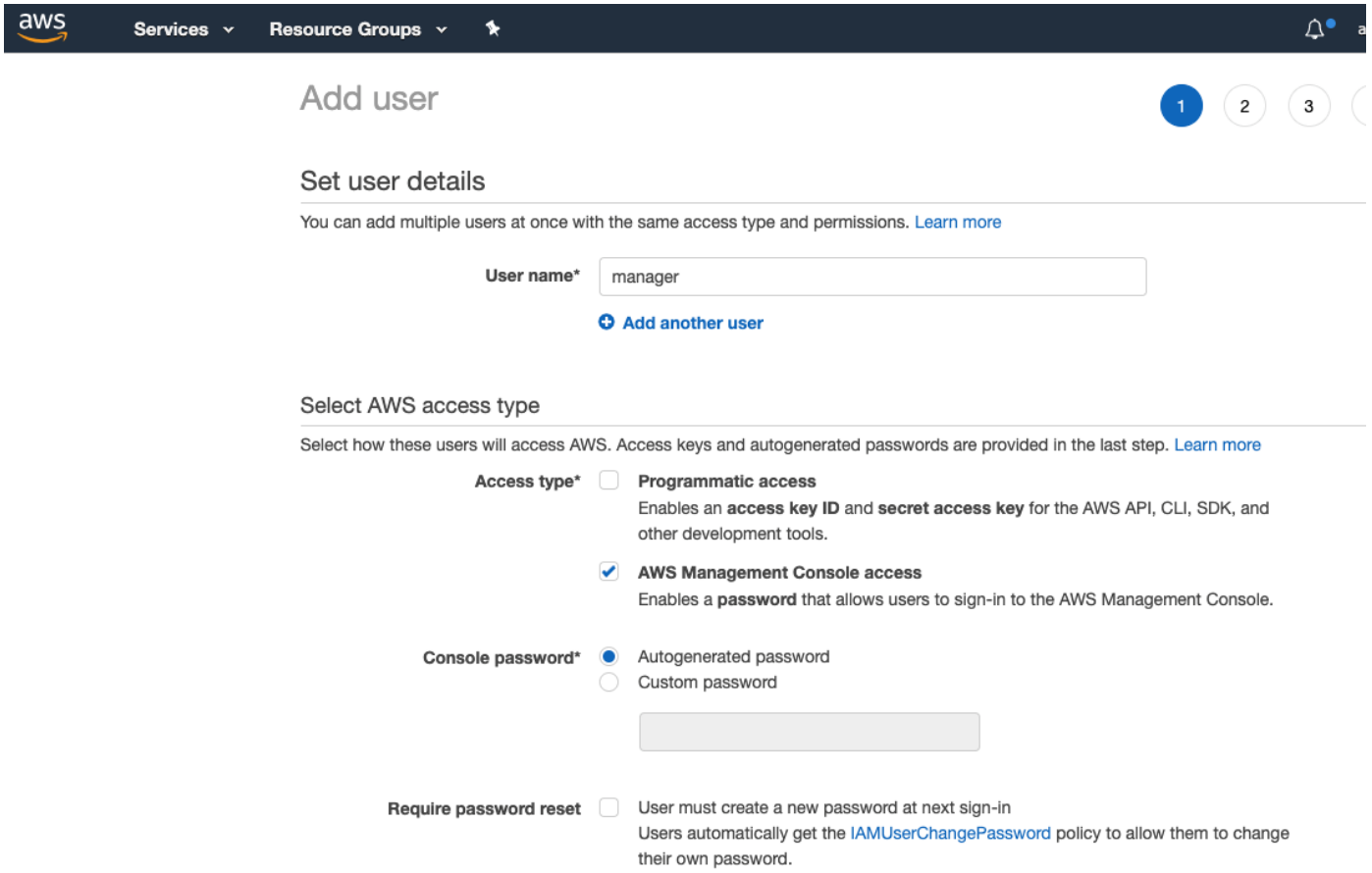
Exercise: Creating A Manager IAM User With Built-in Policies

Creating An IAM User

While logged in as the root account user, navigate to the IAM console.

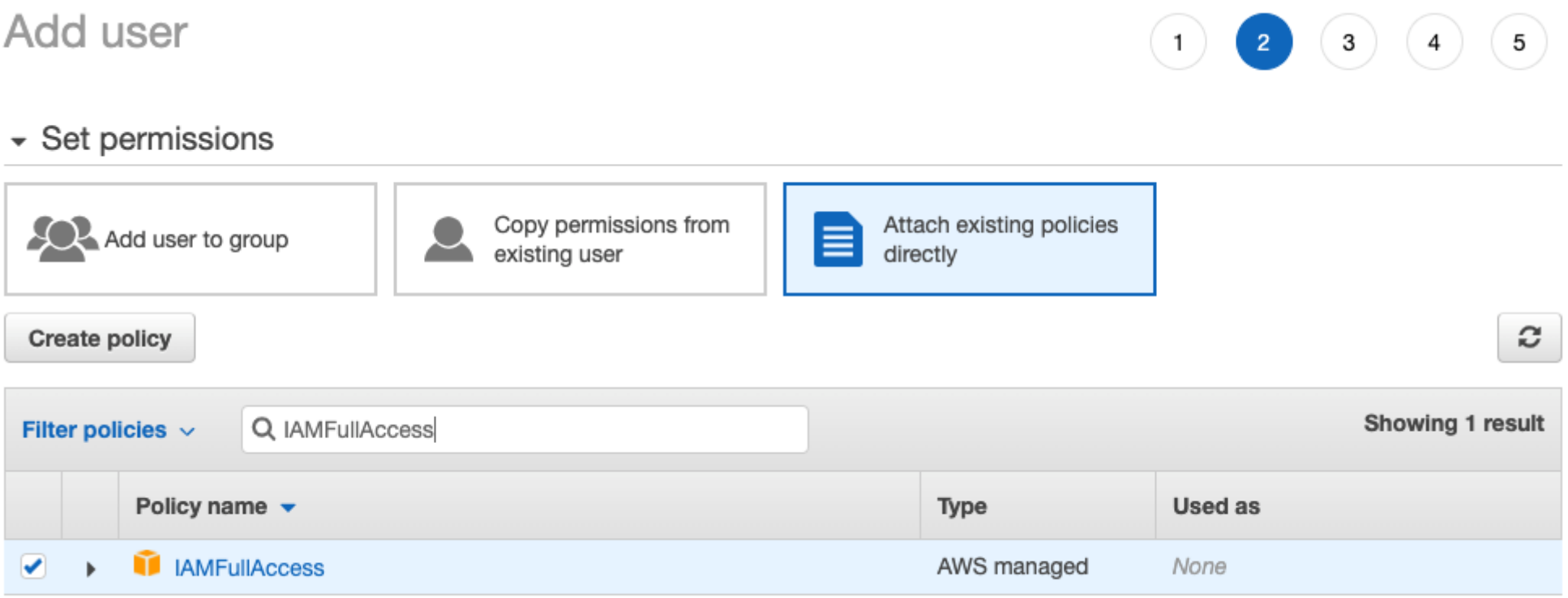


1. From the left sidebar click on **Users** to get into the Users page
2. Click the **Add User** button and set the **User name** to **manager**
3. Select the **AWS Management Console Access** access type
4. Leave the **Autogenerated password** on
5. Uncheck the option for **Require password reset**
6. Click on the **Next: Permissions** button

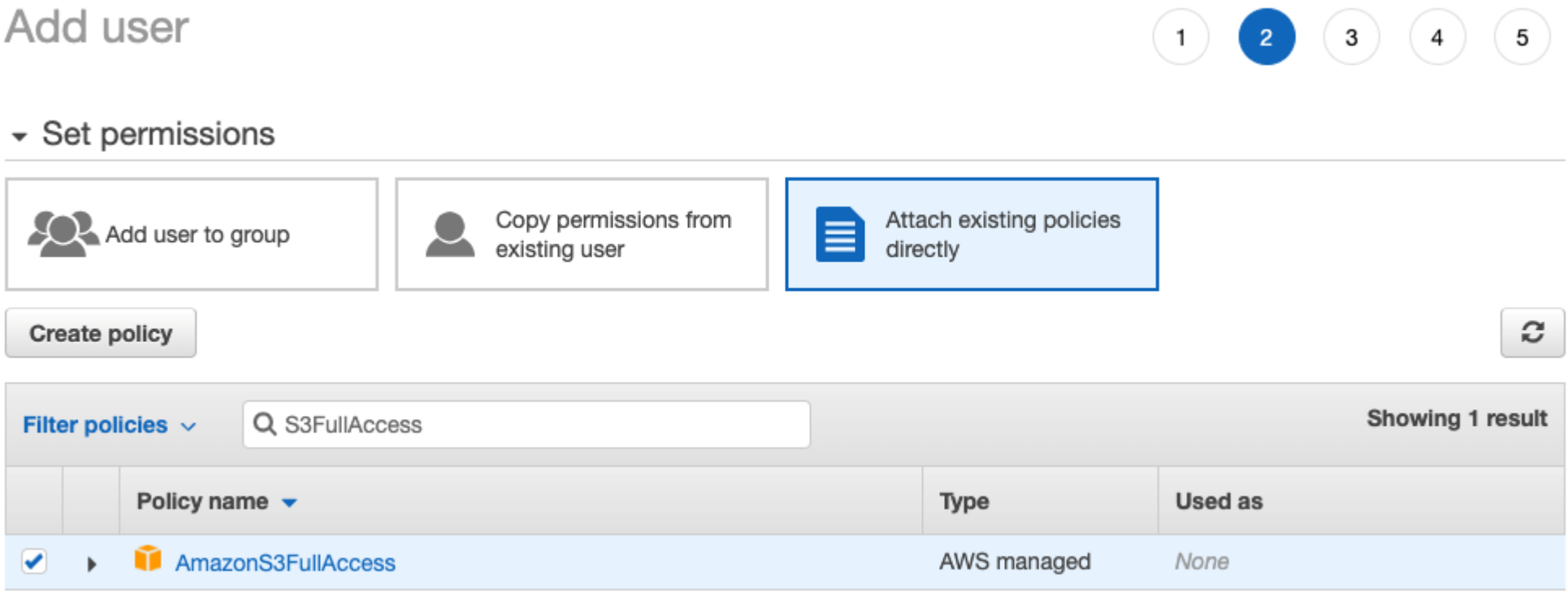


7. Click on the **Attach existing policies directly** button

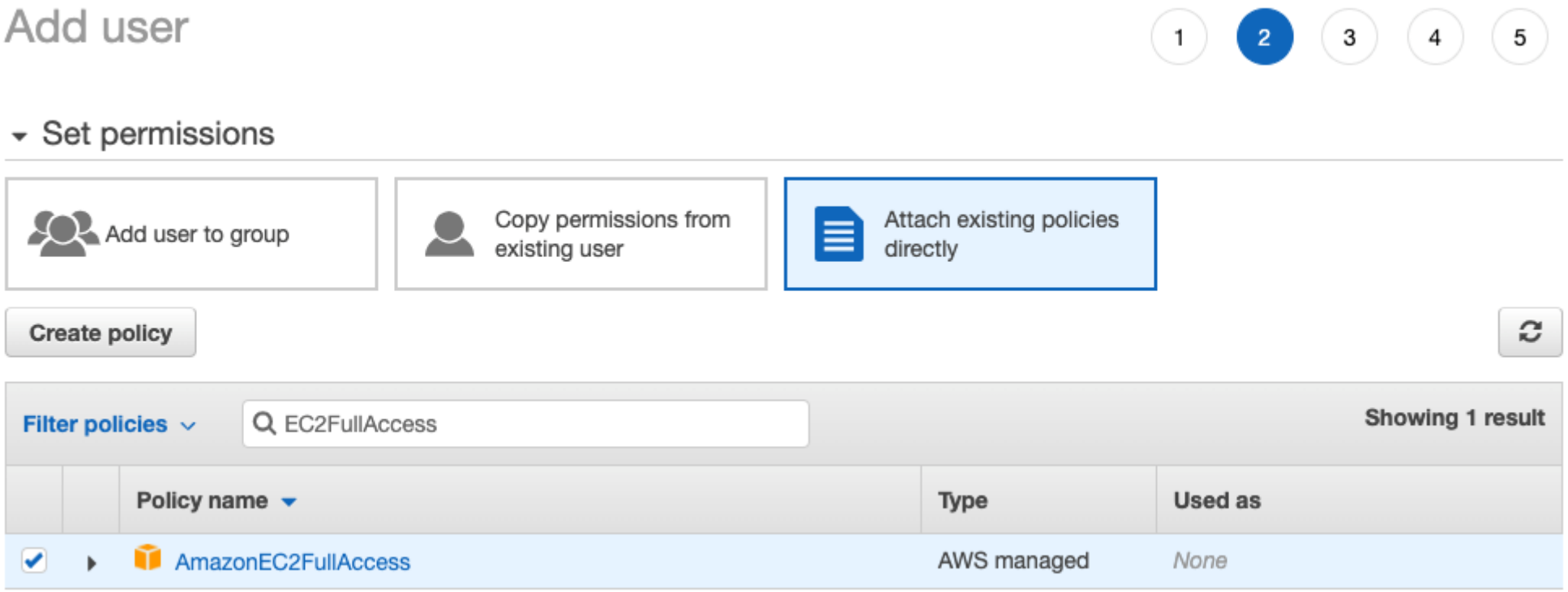
8. Search for **IAMFullAccess** and select it



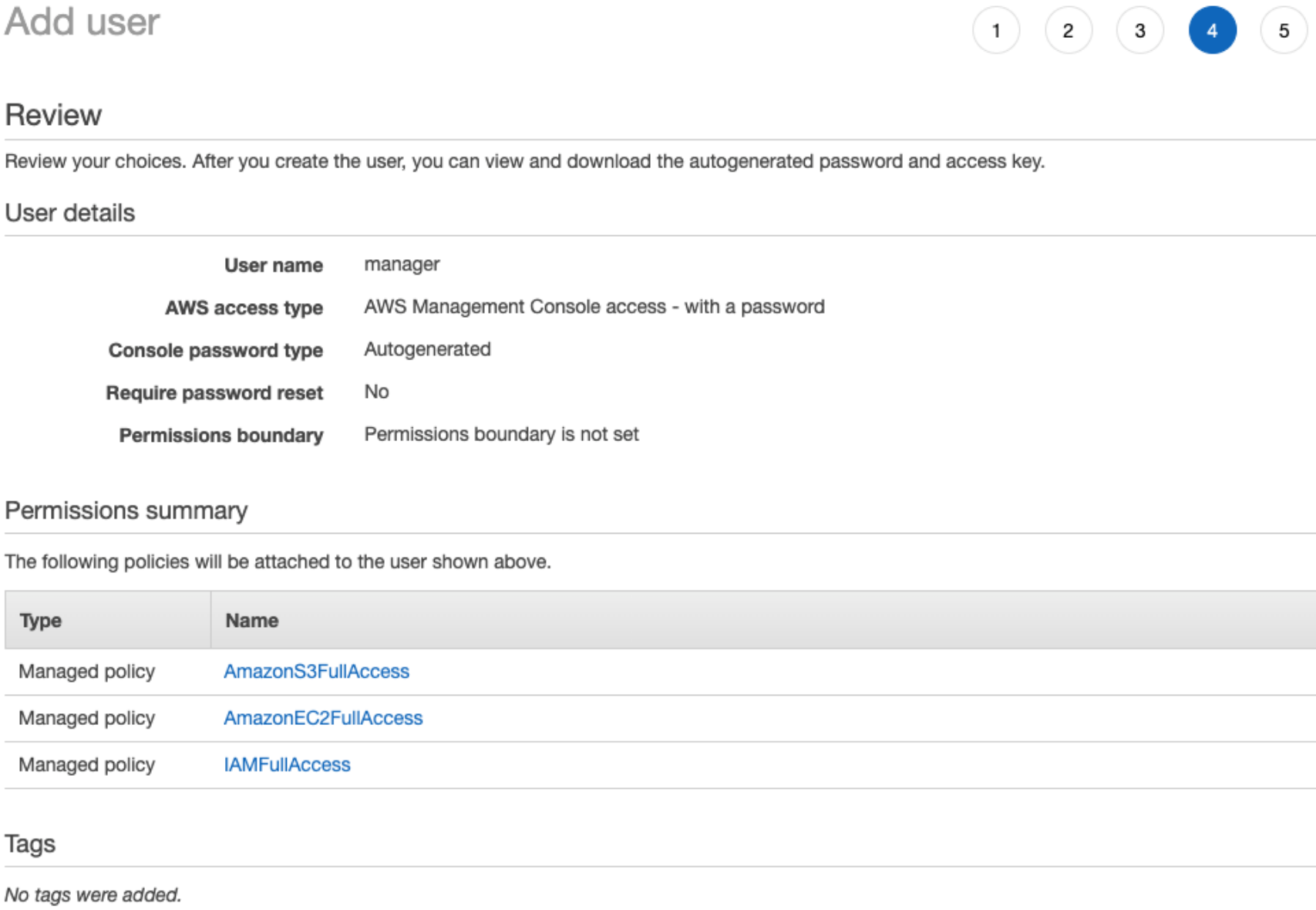
9. Search for **S3FullAccess** and select it



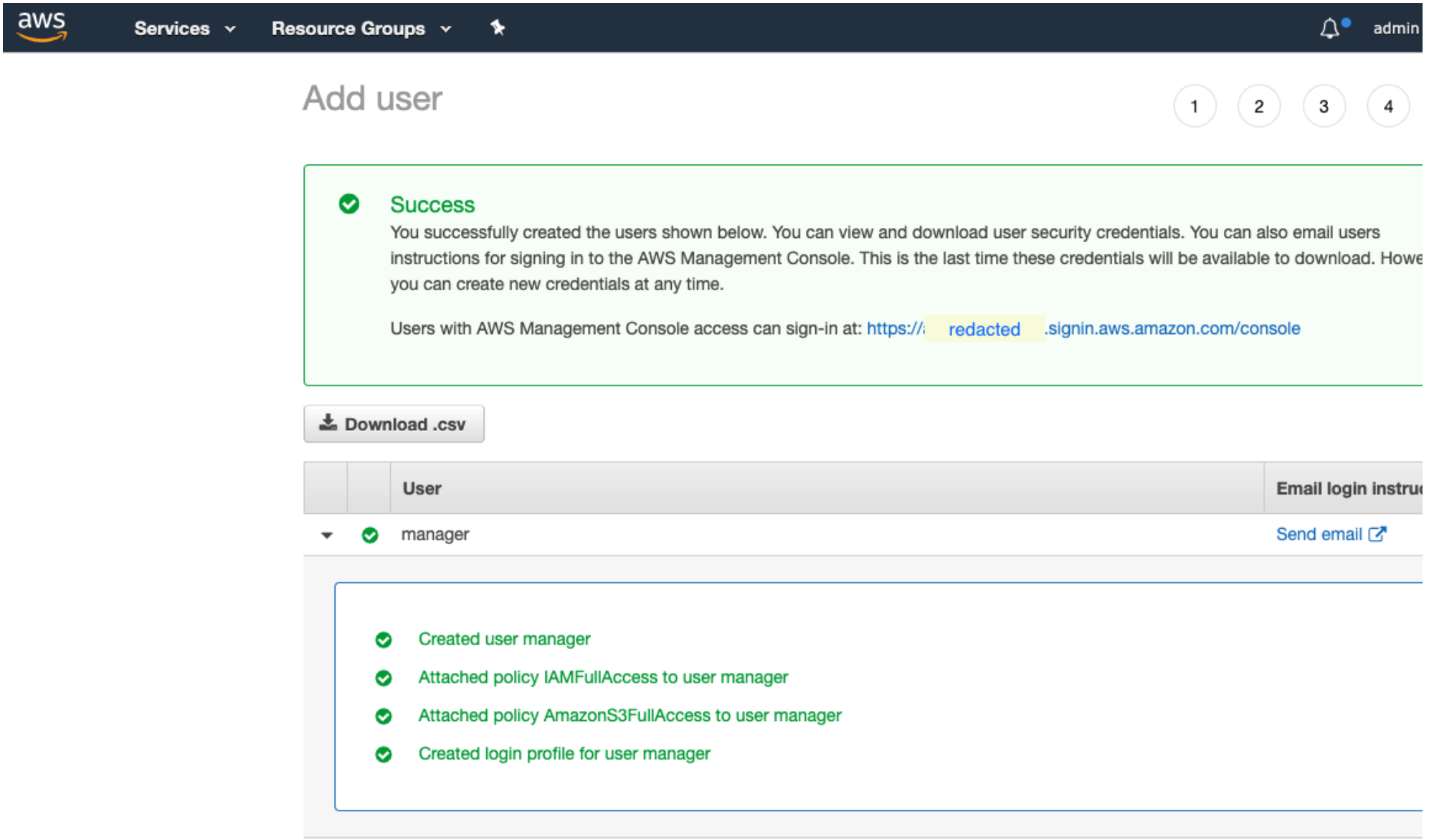
10. Search for **EC2FullAccess** and select it



11. Click on the **Next: Tags** button. We don't need to add any tags at the moment, so skip the tags by clicking on the **Next: Review** button



12. Click on the **Create user** button to create the user



13.Copy and save both the password (show password) and the login URL somewhere you will be able to access it later

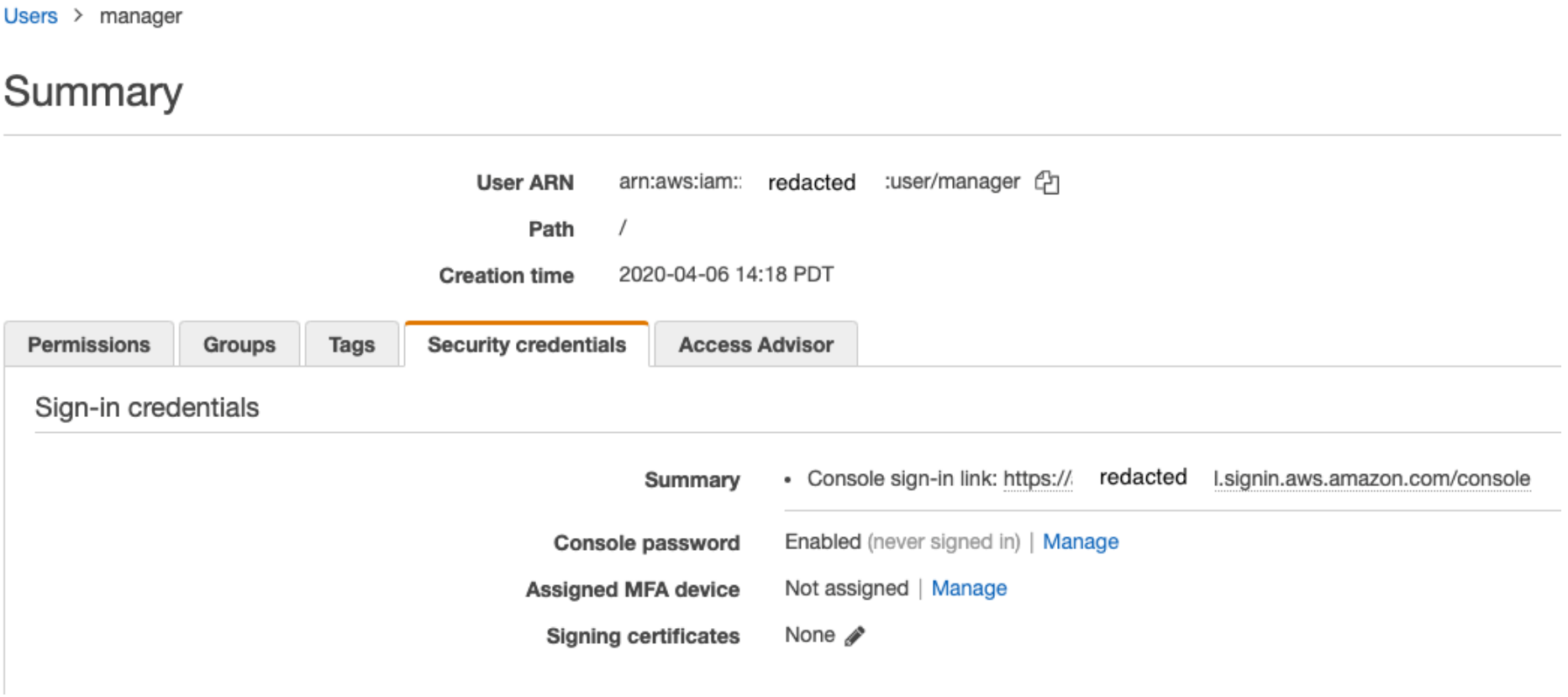
Exercise: Securing IAM User With MFA

Instructions

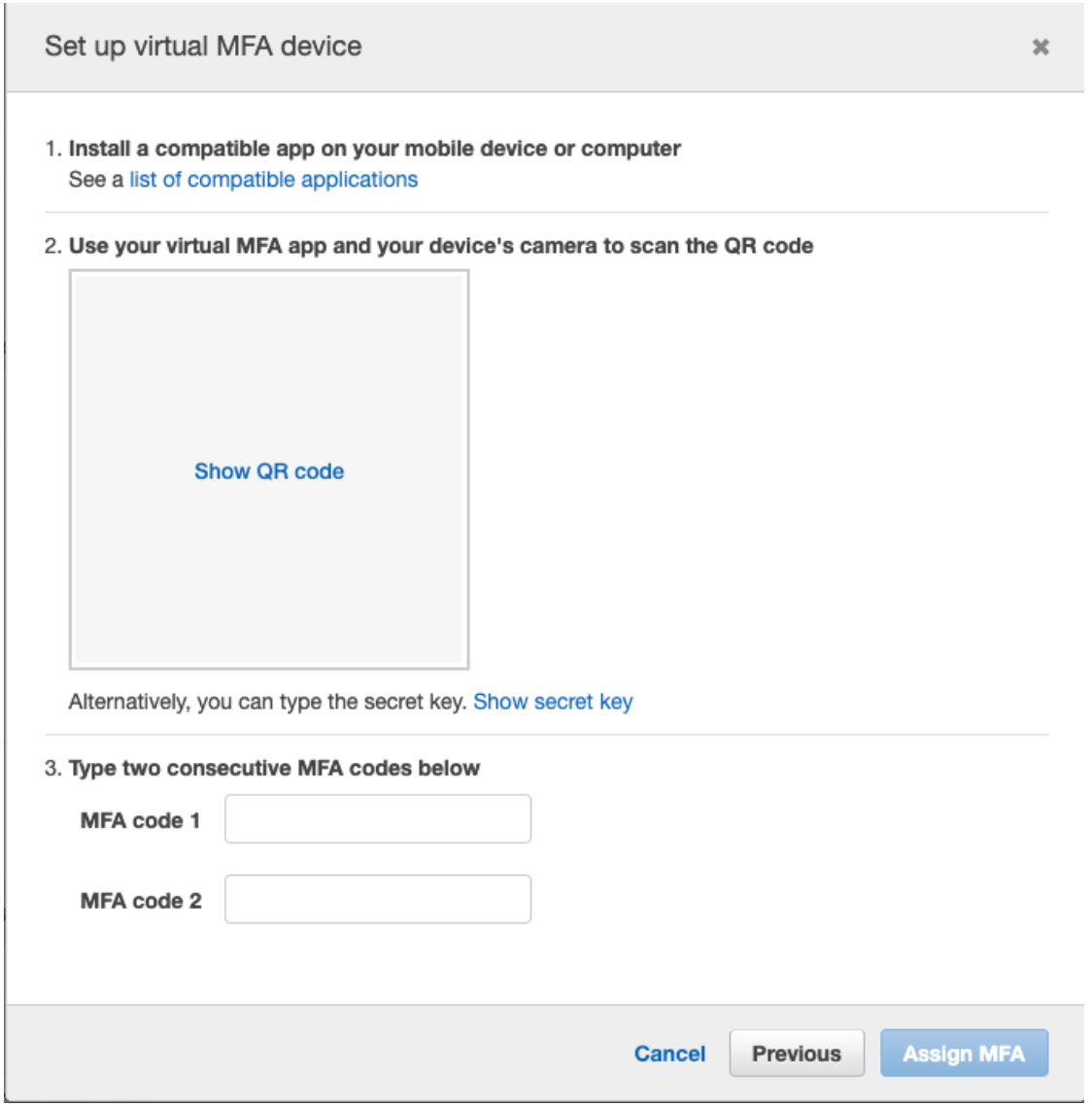
- If you have not yet done so, you must first create a **manager** AWS account
- Install an MFA app on your phone - the **Google Authenticator** app (Android, iOS) is a good choice
- Secure the manager account with MFA
- Log out and log back in to verify that MFA is enabled

Navigate to the IAM console and select the user you want to secure with MFA (in this series of exercises this would be the **manager** user).

1. Click on the user account and open the **Security Credentials** tab



2. Click **Manage** next to the **Assign MFA device**
3. Leave the default **Virtual MFA device** option on and click continue
4. Click the **Show QR code** in the empty square



5. Open the **Google Authenticator** app on your phone and click the **Scan a barcode** camera icon
6. Scan the barcode
7. Type the code you see on your phone into the **MFA code 1** field
8. Wait for the Google Authenticator app to generate the next code
9. Type the next consecutive code into the **MFA code 2** field and click the **Assign MFA** button
10. Using a new incognito or private browser window to login with this user and the MFA code from your phone

If you have configured MFA correctly you should be able to log in.
If not:

1. Log in as the root user
2. Proceed to the IAM service console
3. Click on **Users**
4. Select the **manager** user
5. Visit the **Security credentials** tab
6. Next to **Assigned MFA device**, click **Manage**
7. Remove the assigned MFA device
8. Log out of the root user account
9. Log back in with the manager account
10. Try again to configure MFA

Exercise: Creating S3 Bucket And Uploading Content

Instructions

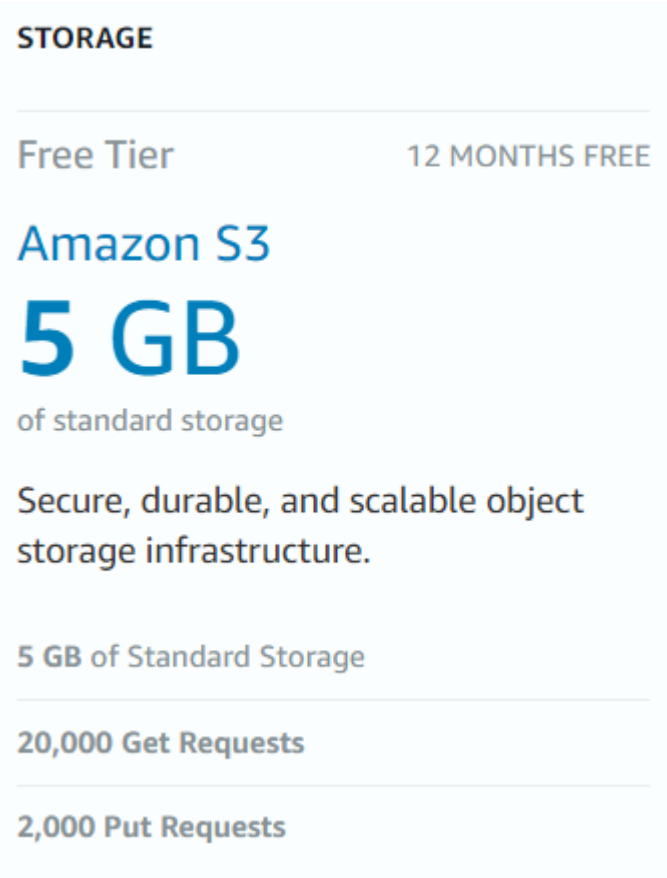
If you have not done so already, register with AWS and authenticate as the **manager** user

Create a bucket in the **us-west-2 Oregon** region

Create 2 folders in the bucket: "exercise" and "manager"

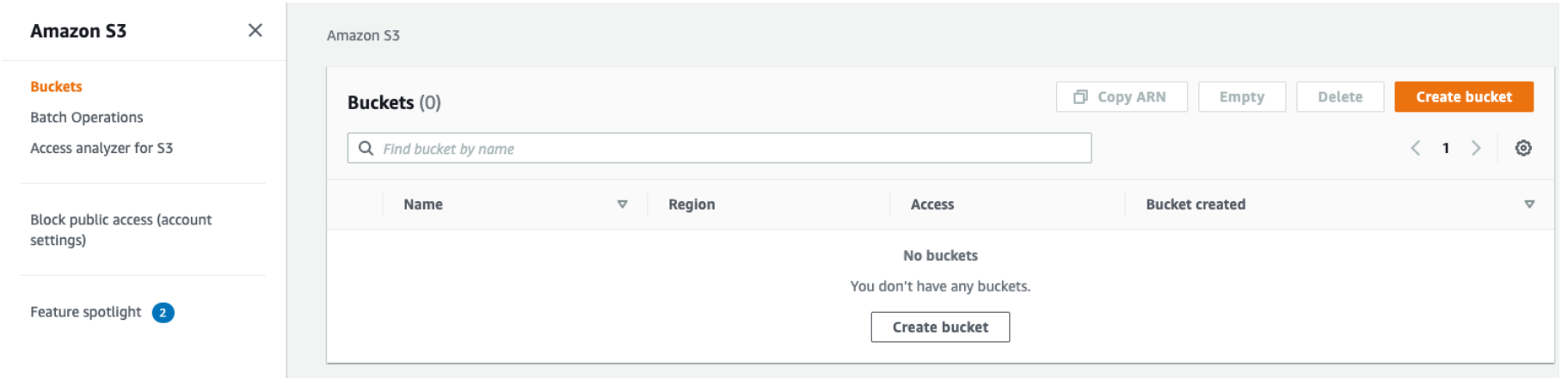
Upload content to both folders (a single file in each folder will suffice)

Note: Please do not delete the bucket until the end of this course. Deleting a bucket takes a long time. If you delete the bucket and then try to create it again later, you may run into errors.



Part-1: Creating The Bucket

1. Log out of the root account user by clicking on the top right email next between the **Bell icon** and the **Global** dropdown menu
2. Log in to the console as the manager, using the information from you've got from the previous step
3. Navigate to the S3 console
4. Create a new bucket (remember that bucket names are globally unique across all AWS accounts)



Note that if a bucket name already exists you will get a warning.

Note:

Deleting a bucket takes time, so creating the same bucket name after a deletion would not allow you to create the bucket and will result in a "Bucket already exists error"

Amazon S3 > Create bucket

Create bucket

General configuration

Bucket name

ami-test

Bucket with the same name already exists

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

Region

US West (Oregon) us-west-2

Bucket settings for Block Public Access

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Block *all* public access

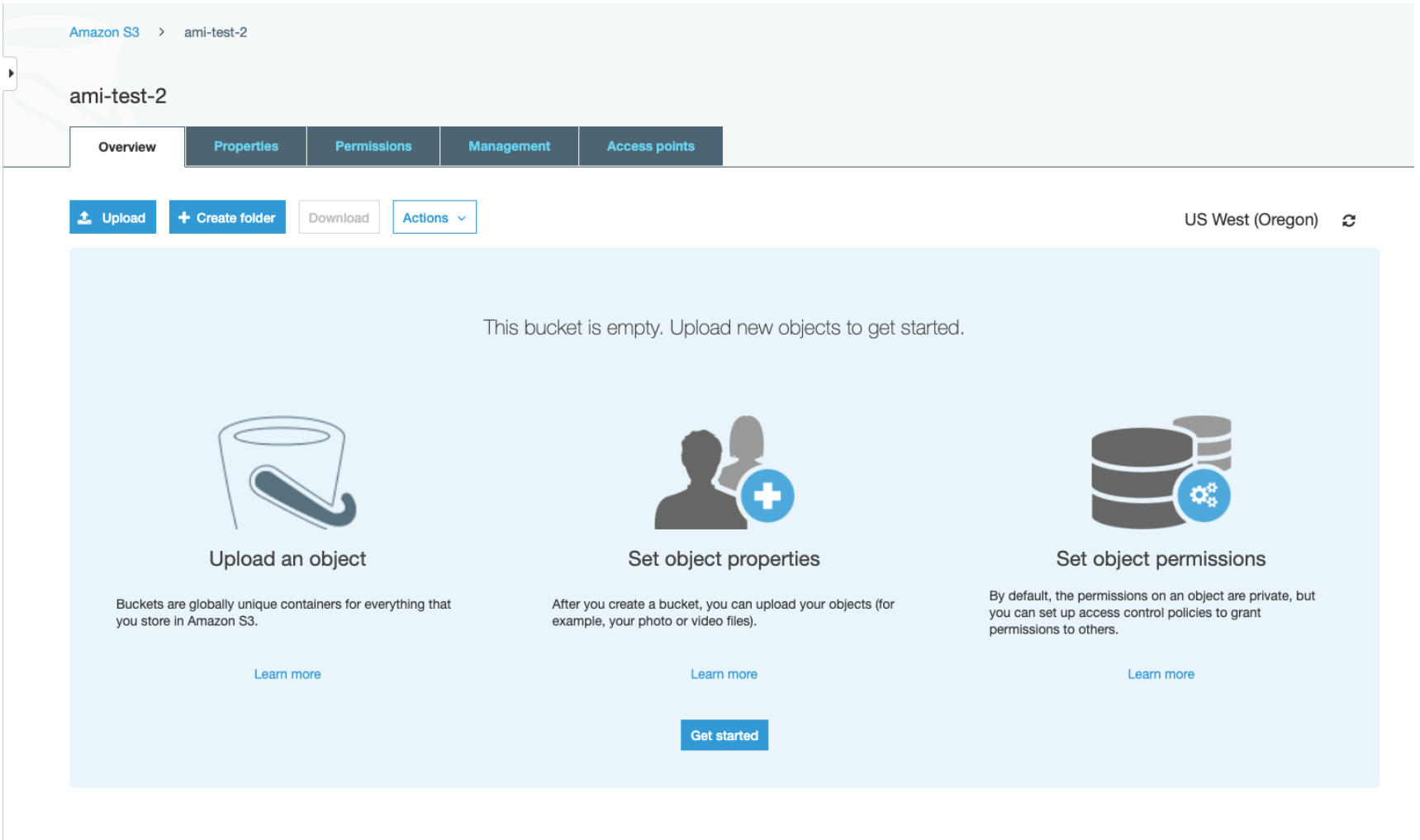
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

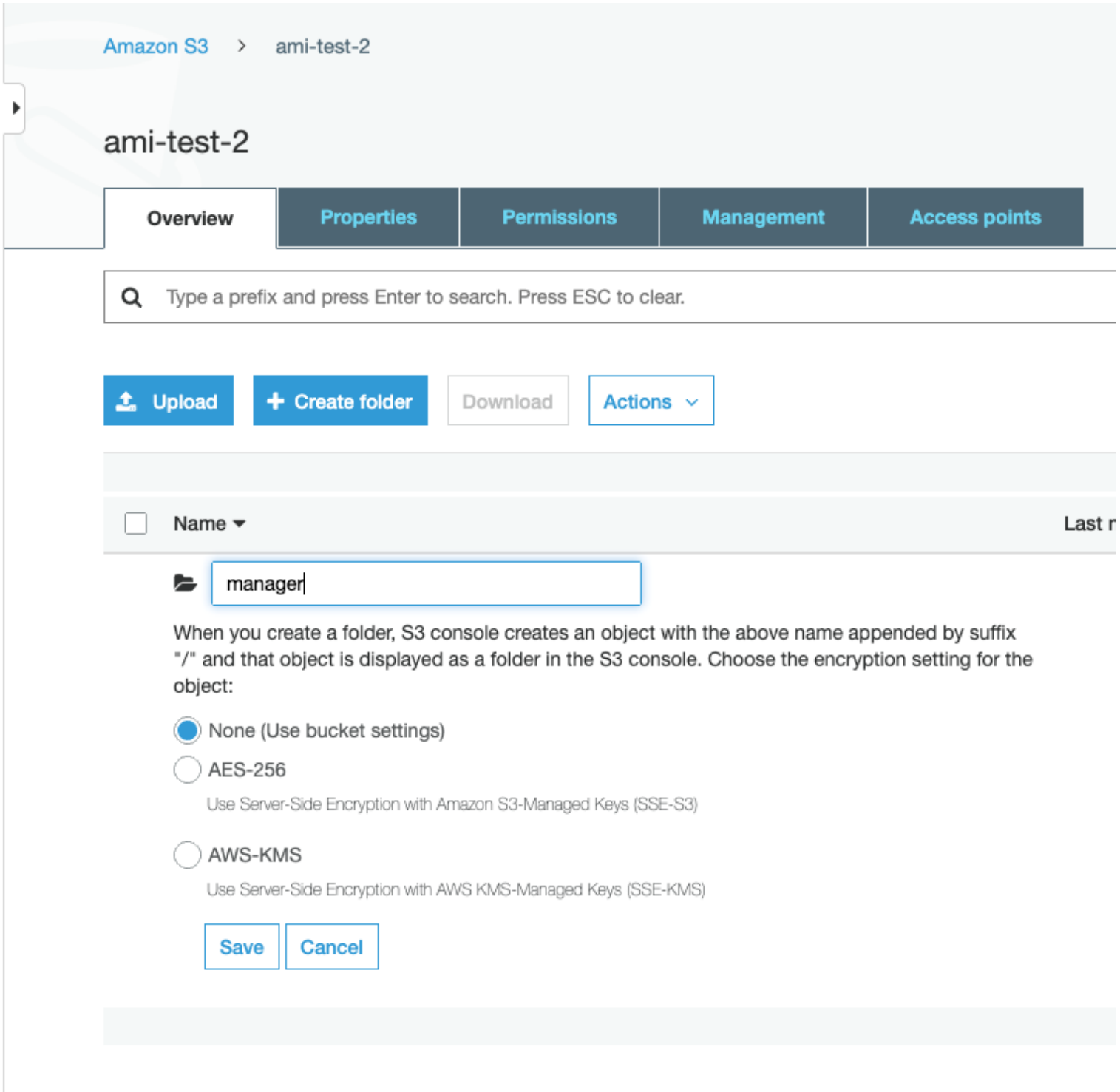
- **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Advanced settings

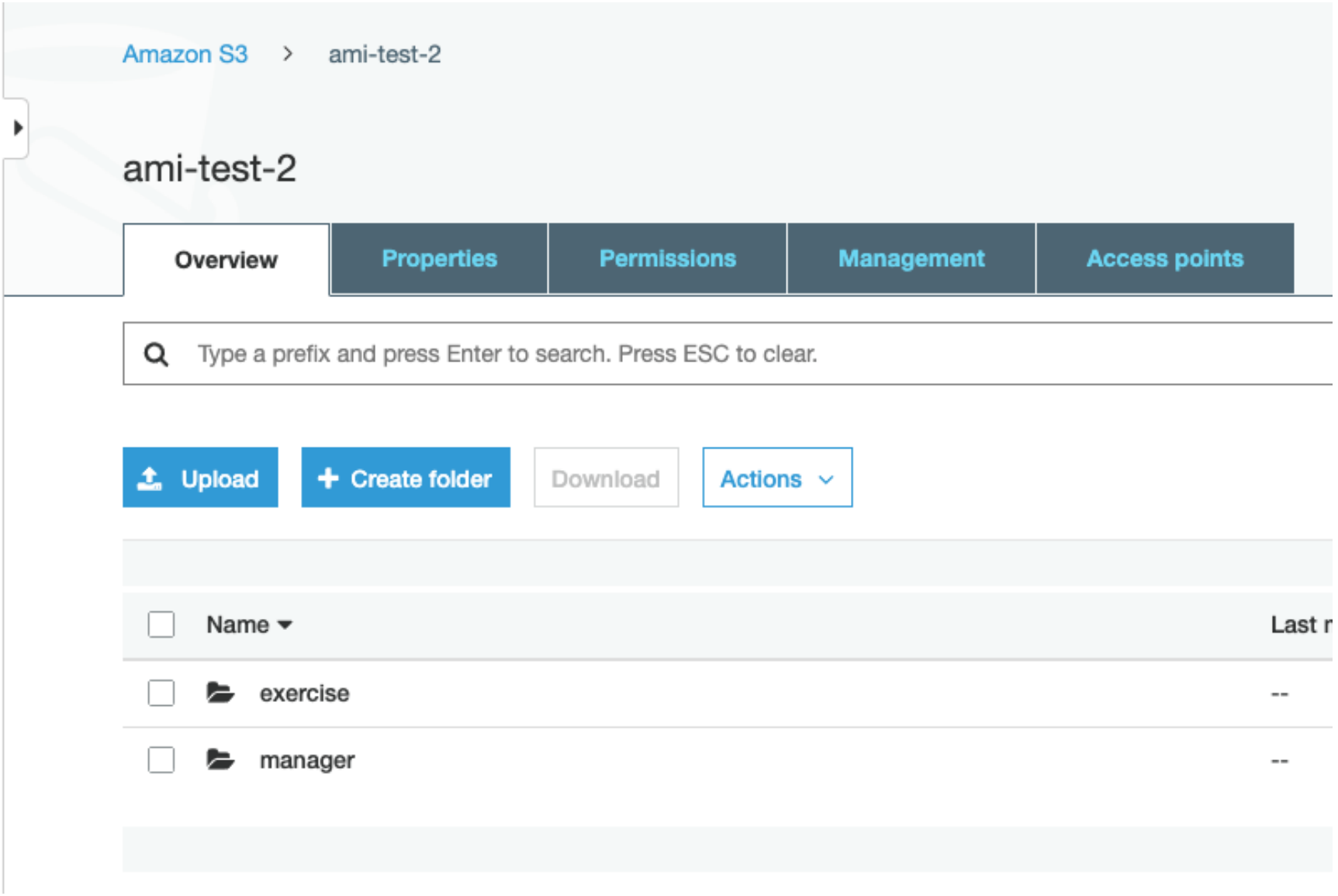
Cancel Create bucket

5. Once the bucket is created, click on the **Create folder** button to create a **manager** folder

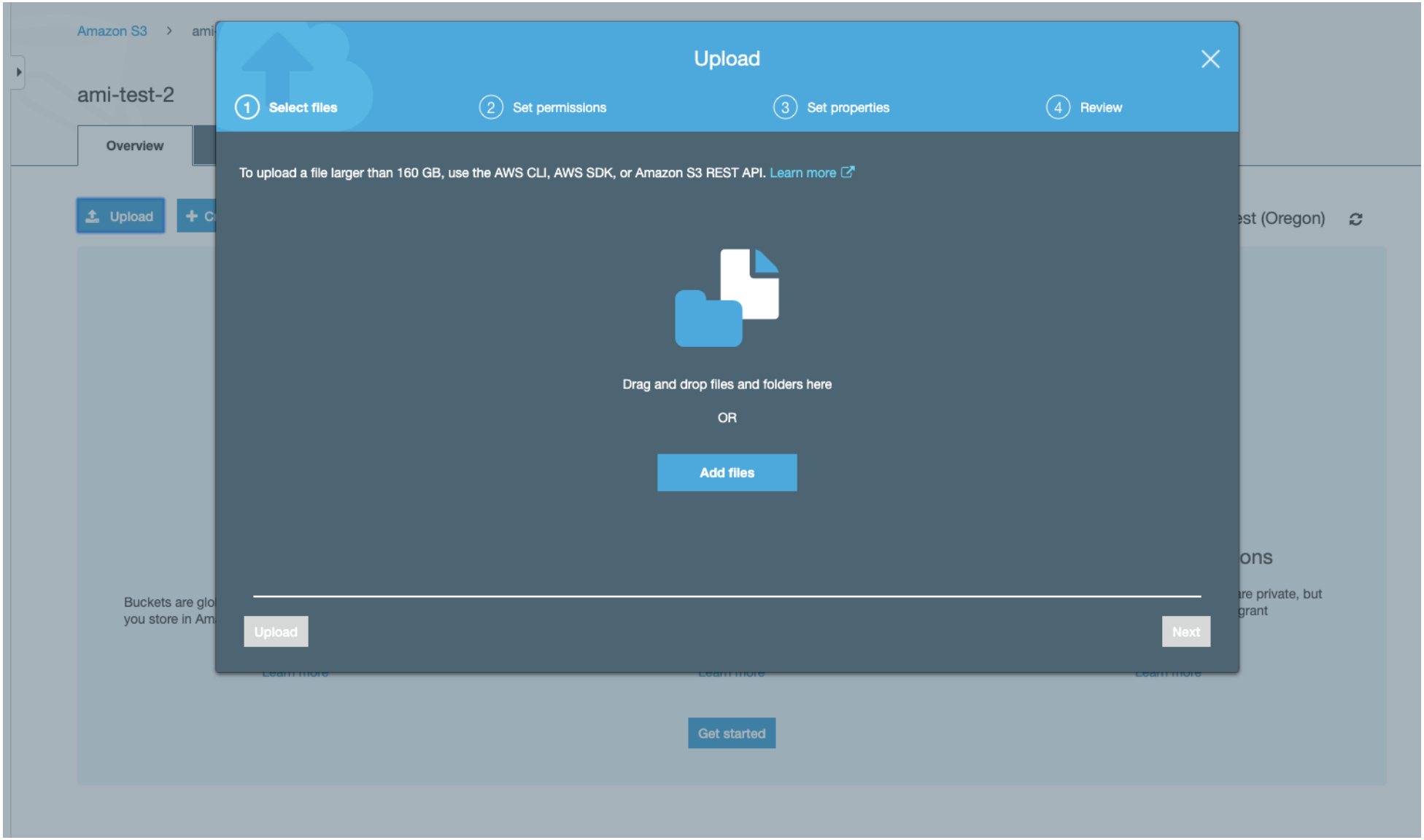



Amazon S3 > ami-test-2
ami-test-2
Overview
Properties
Permissions
Management
Access points
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
Name
Last r
manager
When you create a folder, S3 console creates an object with the above name appended by suffix "/" and that object is displayed as a folder in the S3 console. Choose the encryption setting for the object:
None (Use bucket settings)
AES-256
Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
AWS-KMS
Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)
Save
Cancel

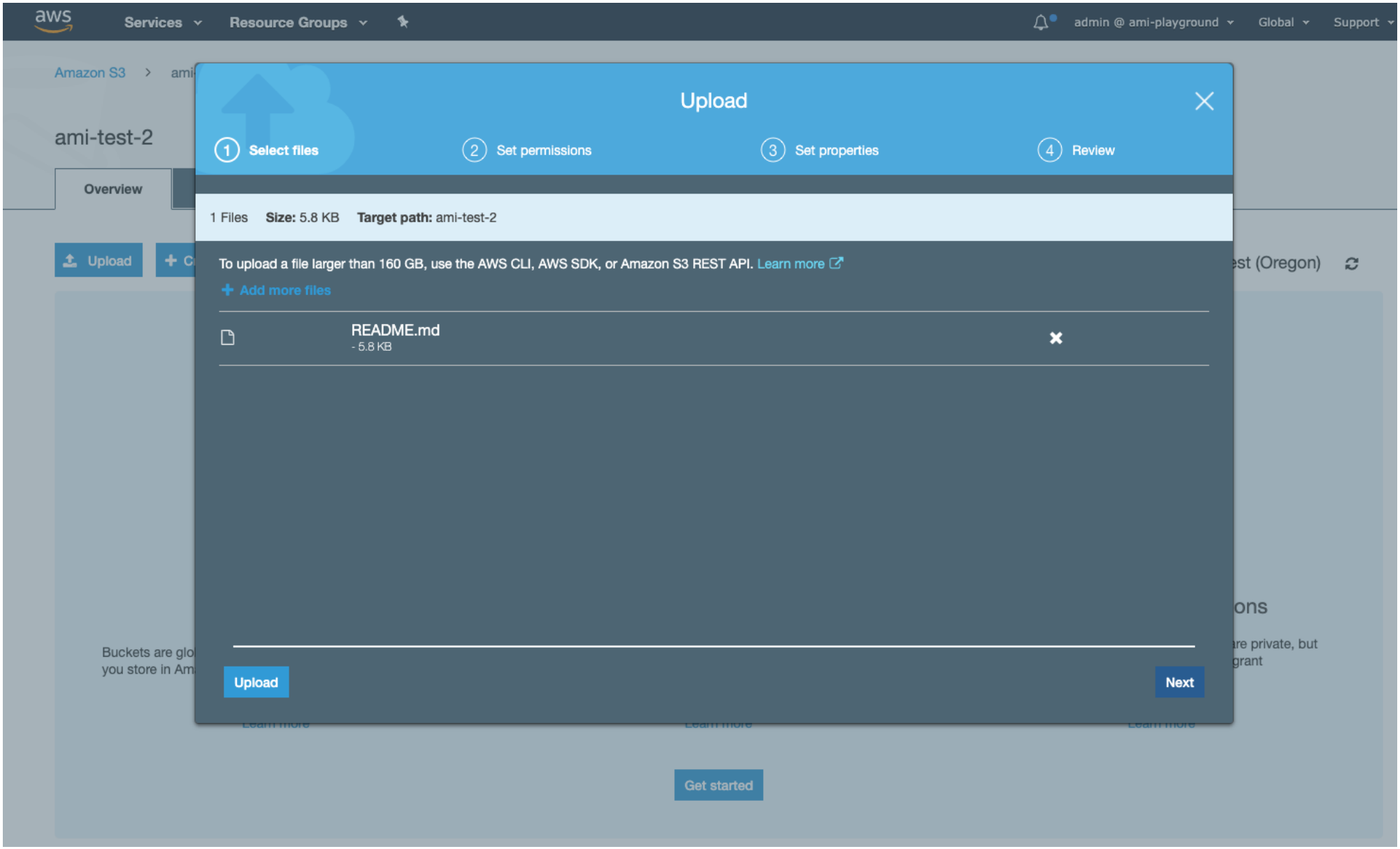
6. Create another folder called exercise



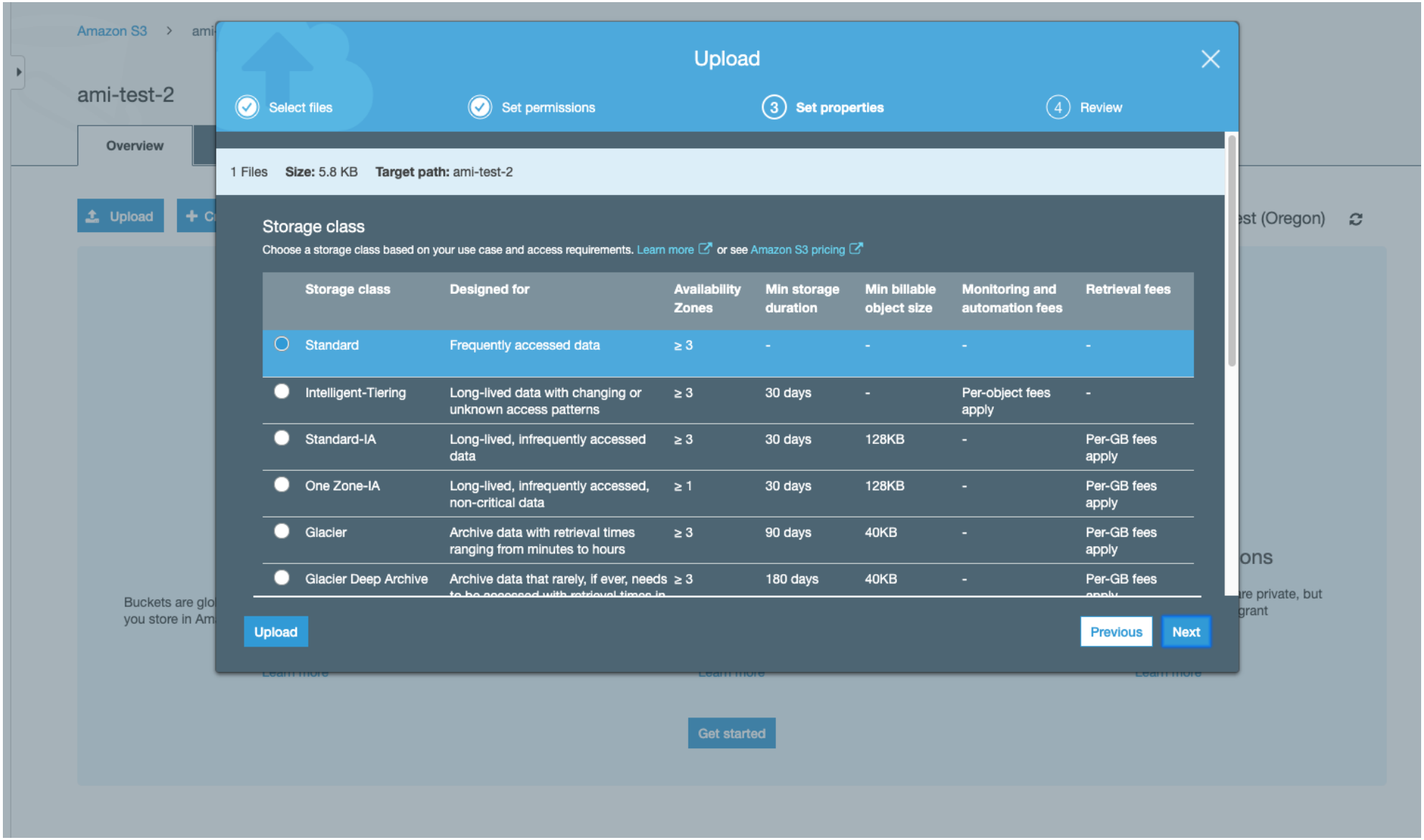
7. Click on the **exercise** folder to get into that folder
8. Click on the **Upload** button to upload a file (any file) into that folder



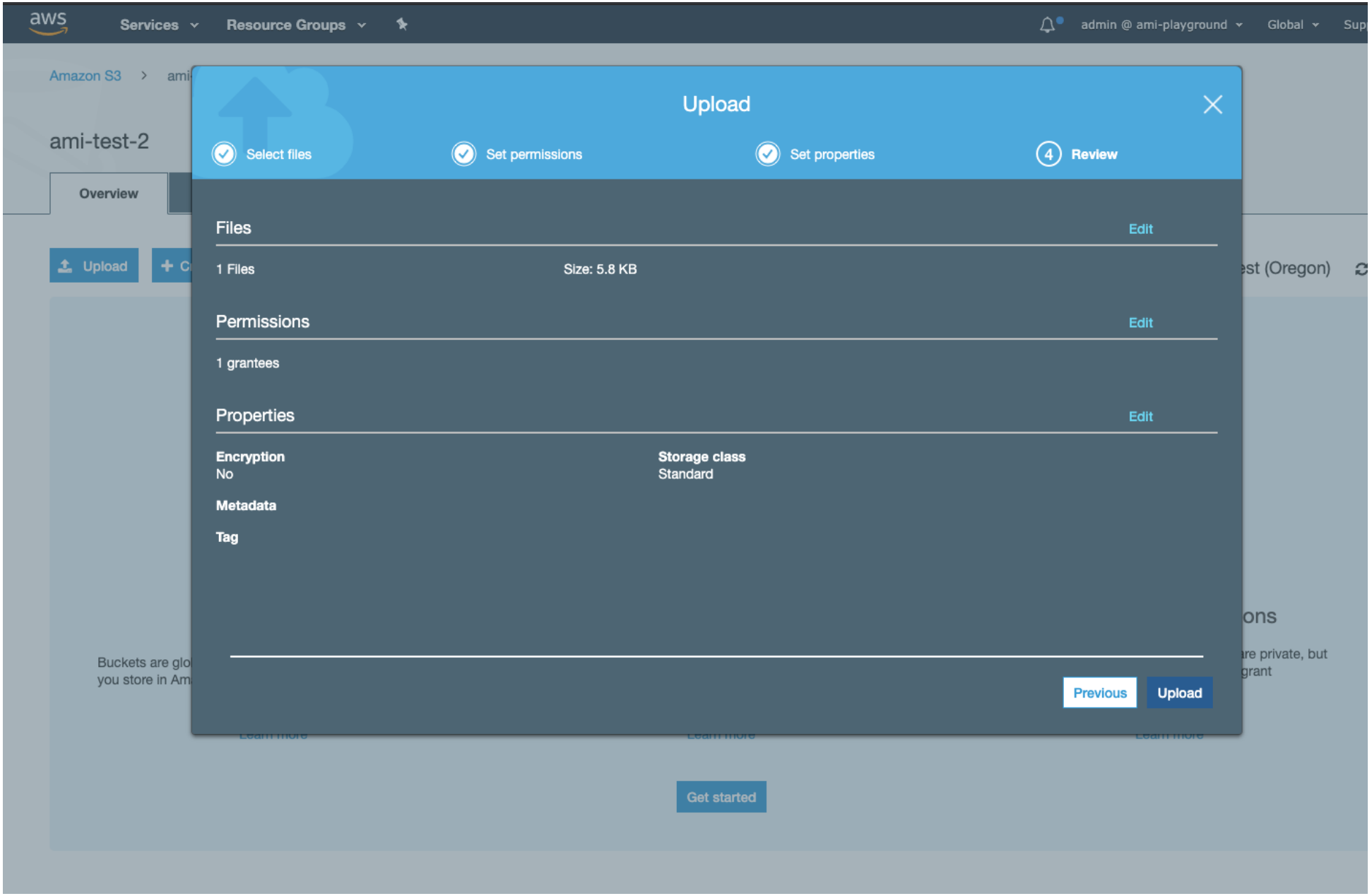
9. Either drag and drop a file onto the window or click the **Add files** button to start the upload process

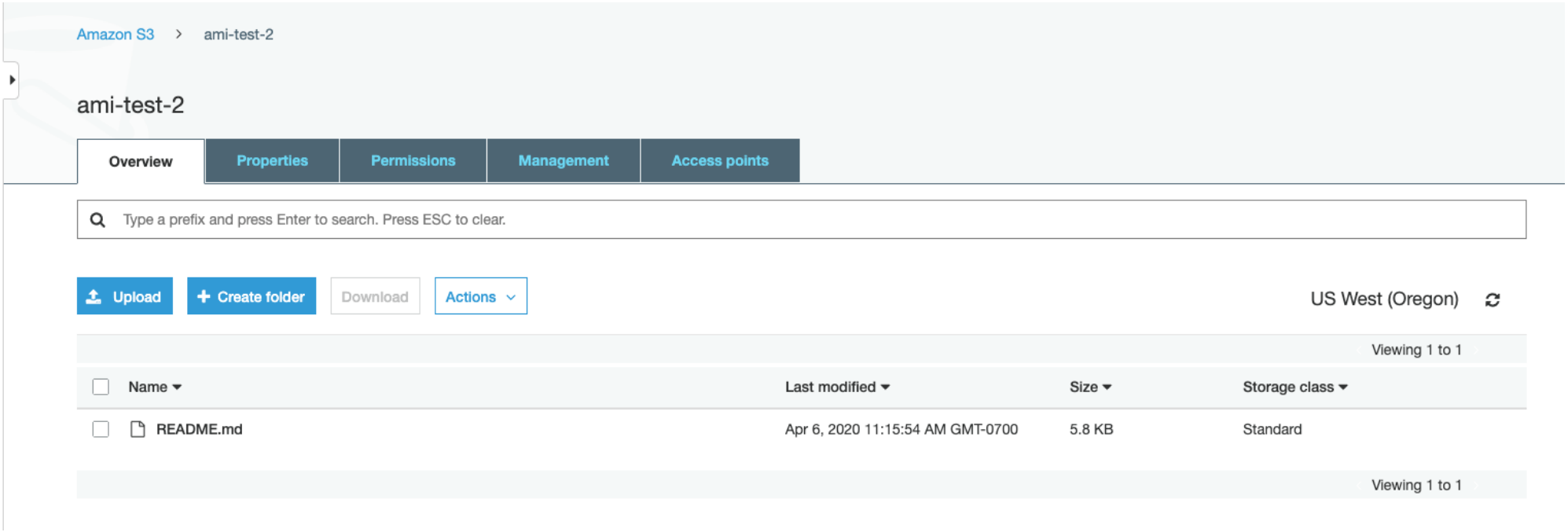


10. Click the **Next** button to see the list of default permissions for this file (we will use IAM permission to access it rather than resource permissions)
11. Click Next to see the S3 storage classes, and leave the default **Standard** class selected



12. Click the **Upload** button to start the upload.




Amazon S3
ami-test-2
ami-test-2
Overview
Properties
Permissions
Management
Access points
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
US West (Oregon)
Viewing 1 to 1
Name
Last modified
Size
Storage class
README.md
Apr 6, 2020 11:15:54 AM GMT-0700
5.8 KB
Standard
Viewing 1 to 1

Monitoring your AWS costs

All AWS services are a pay-as-you-go service, so we urge our students to closely monitor their usage costs and if they have adequate credits available to complete their project/task. Follow the instructions below to do that:

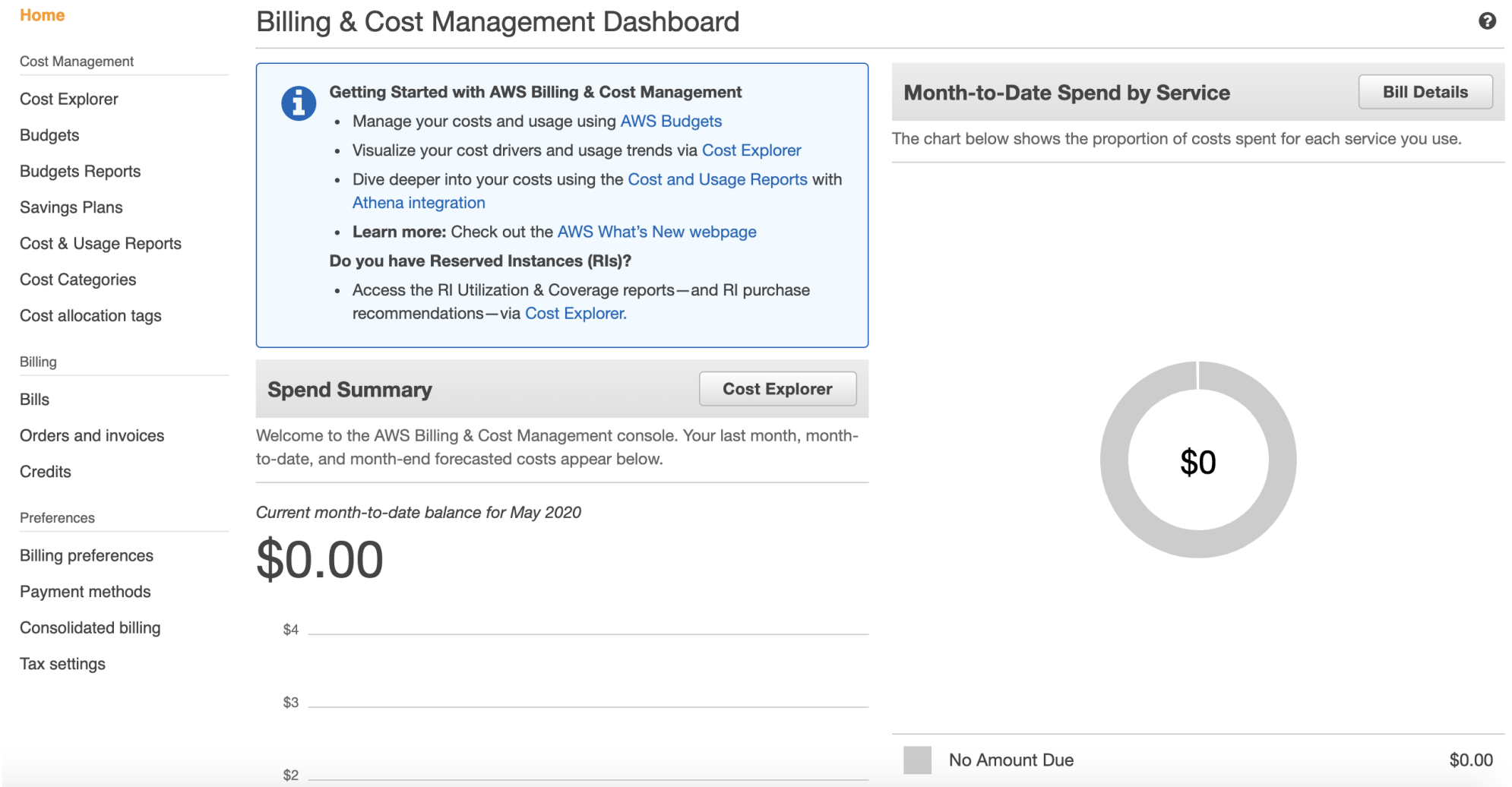
Step 1. Log into your AWS account.

Step 2. Examine your costs

Go to https://console.aws.amazon.com/billing/

You should see the following billing dashboard where it will show your costs.

Monitoring your AWS costs

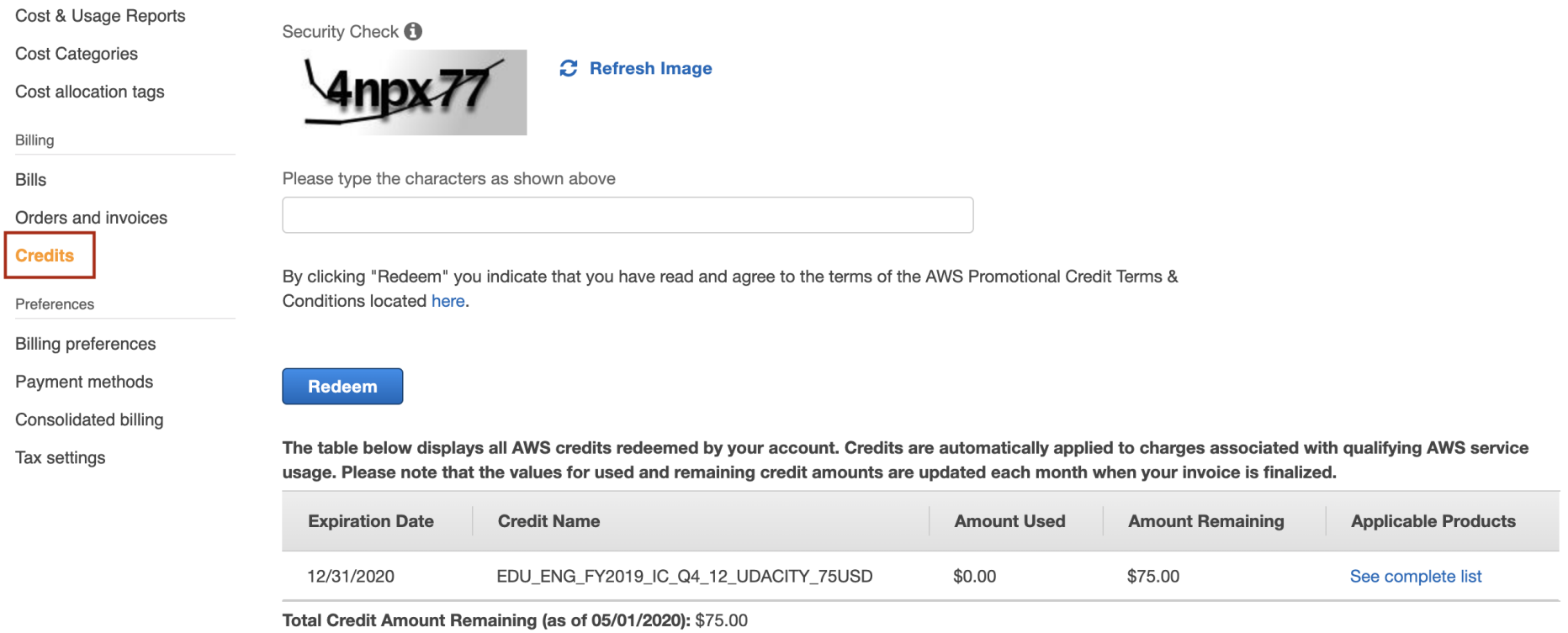


Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Monitoring your AWS costs

Step 3 (optional). Check the value of your credits.

Click on the "Credits" from the left side menu and the following screen will show with your available credits.



AWS Elastic Compute Cloud (EC2)

- Resizeable, scalable compute capacity
- Instances sizes are predefined by instance types
- Configuring an instance can be done within the EC2 Launch Wizard:
    - Security groups (i.e. firewall rules)
    - Networking (public or private)
    - Storage size and type
    - Custom provisioning scripts

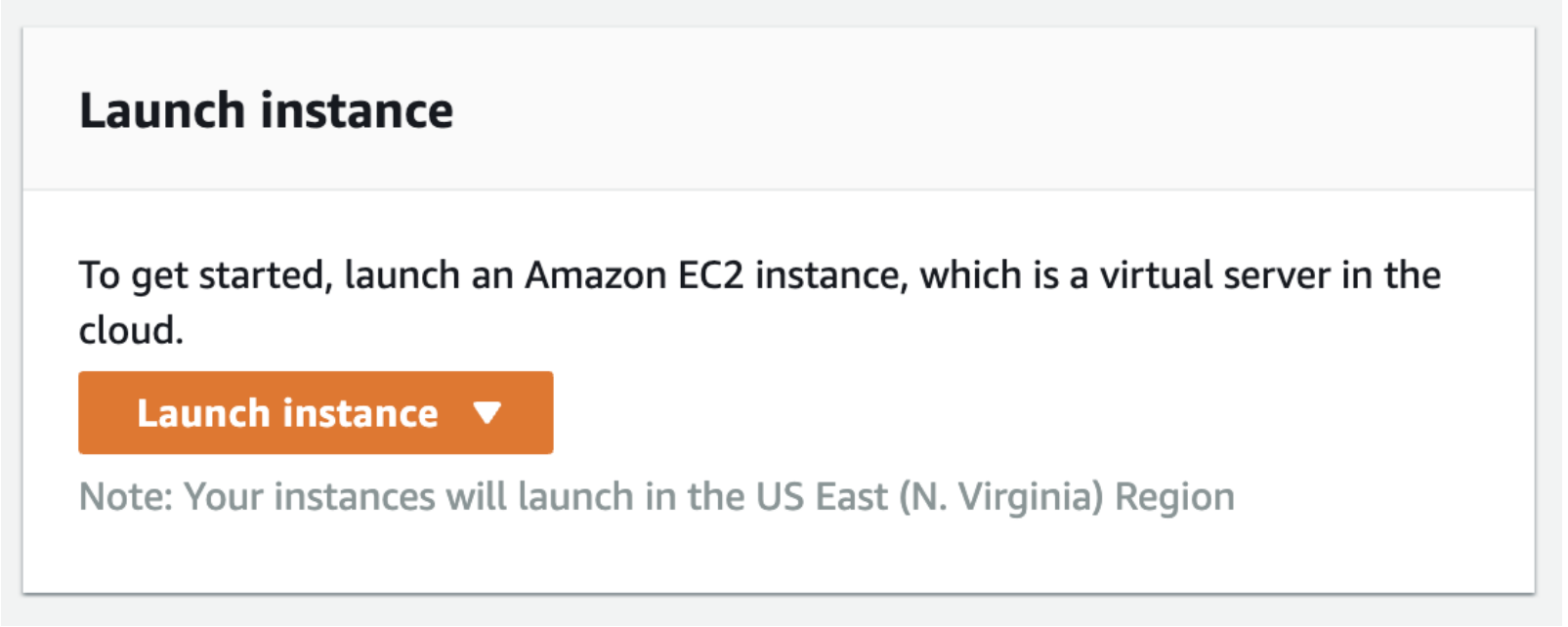
Intro to Cloud Computing https://www.udacity.com/course/intro-to-cloud-computing--ud080

Exercise: Launch An E2 Instance

In this exercise, you will launch an EC2 instance.
No need to connect to this instance using a key pair.

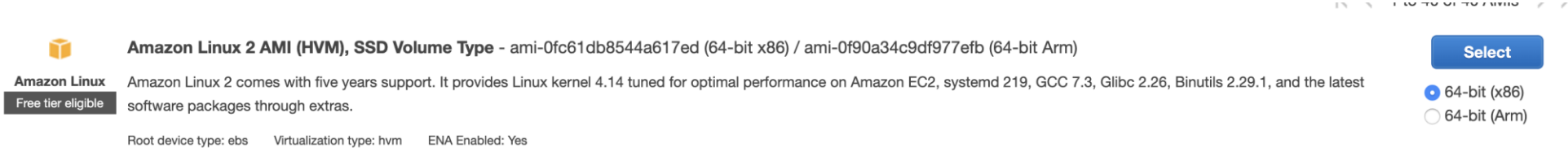
Instructions
Region: **us-east-1**
AMI: **Amazon Linux 2 AMI (HVM), SSD Volume Type 64-bit**
Type: **t2.micro** (Free Tier = 750 hours per month)
Root volume size: **20GB**
IP: **Public**

Use the **us-east-1 N.Virginia** region to launch an EC2 instance with the following configuration:
AMI: Amazon Linux
Instance type: t2.micro
IP: Public
Storage size: 20GB

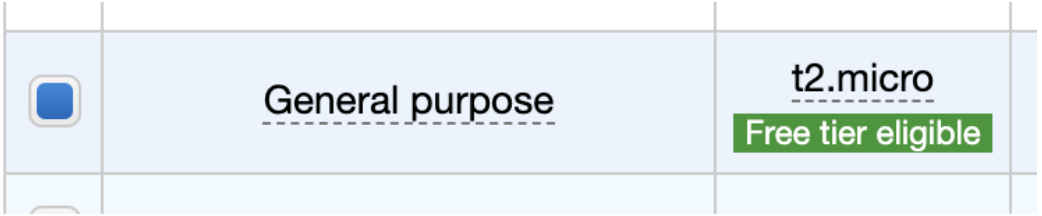


Intro to Cloud Computing https://classroom.udacity.com/courses/ud080/lessons/ac7396e3-43f7-46b2-8f8b-c69d67dbe220/concepts/382c4f04-93fa-4adc-a1ba-03a161b95300

Choose the **Amazon Linux 2 AMI (HVM), SSD Volume Type 64-bit** image.



Select the **t2.micro** option and click the **Next: Configure Instance Details** button.



1. Skip the defaults for VPC and Subnet
2. Change the **Auto-assign Public IP** field from the default **disabled** to **Enable** and then click **Next: Add Storage** button
3. Change the default size from **8GB** to **20GB**

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | **4. Add Storage** | 5. Add Tags | 6. Configure Security Group

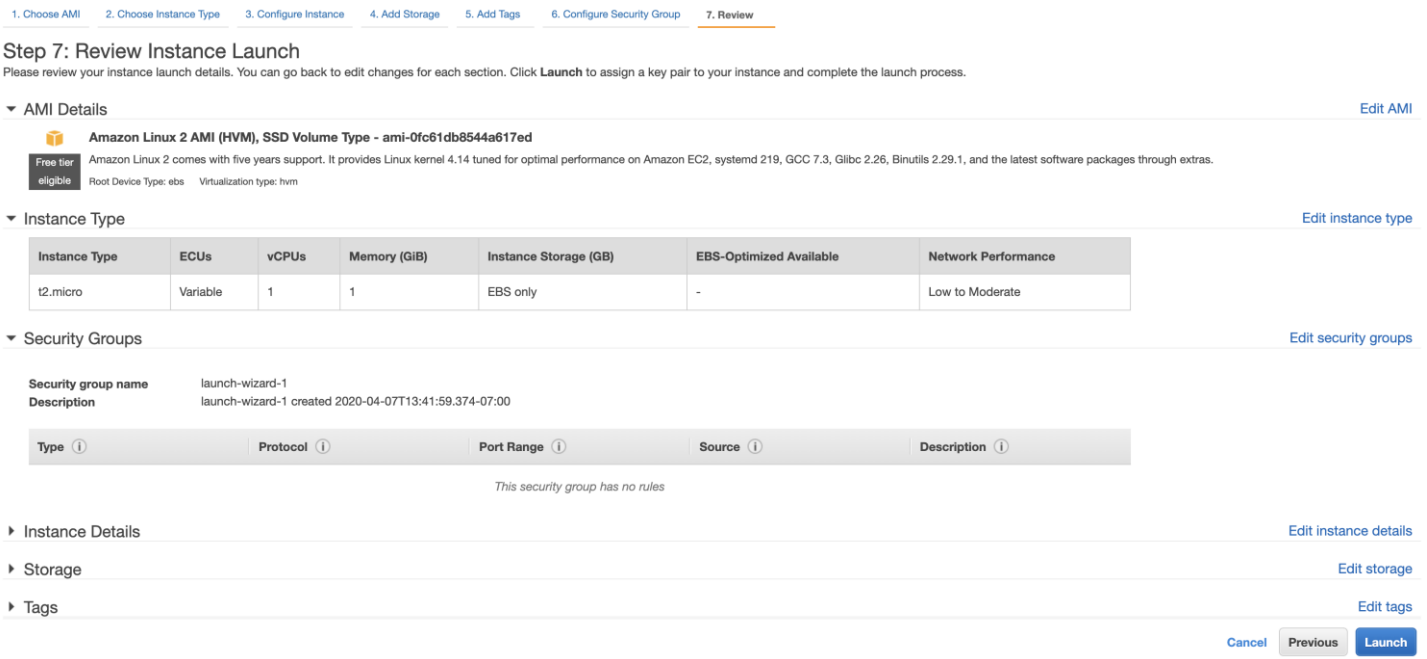
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

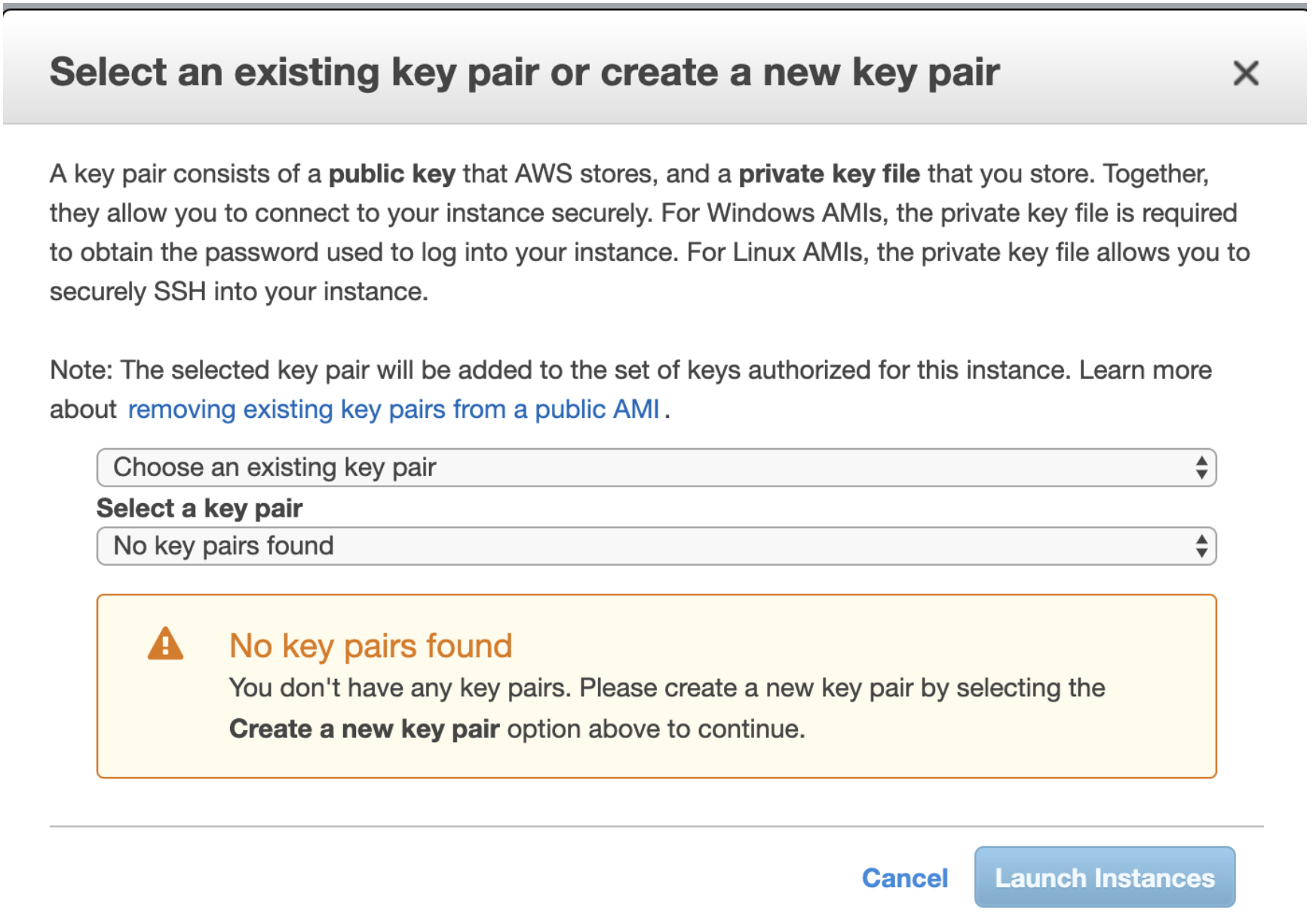
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination
Root	/dev/xvda	snap-0e27a39c6e2f9f079	20	General Purpose S	100 / 3000	N/A	☑

Add New Volume

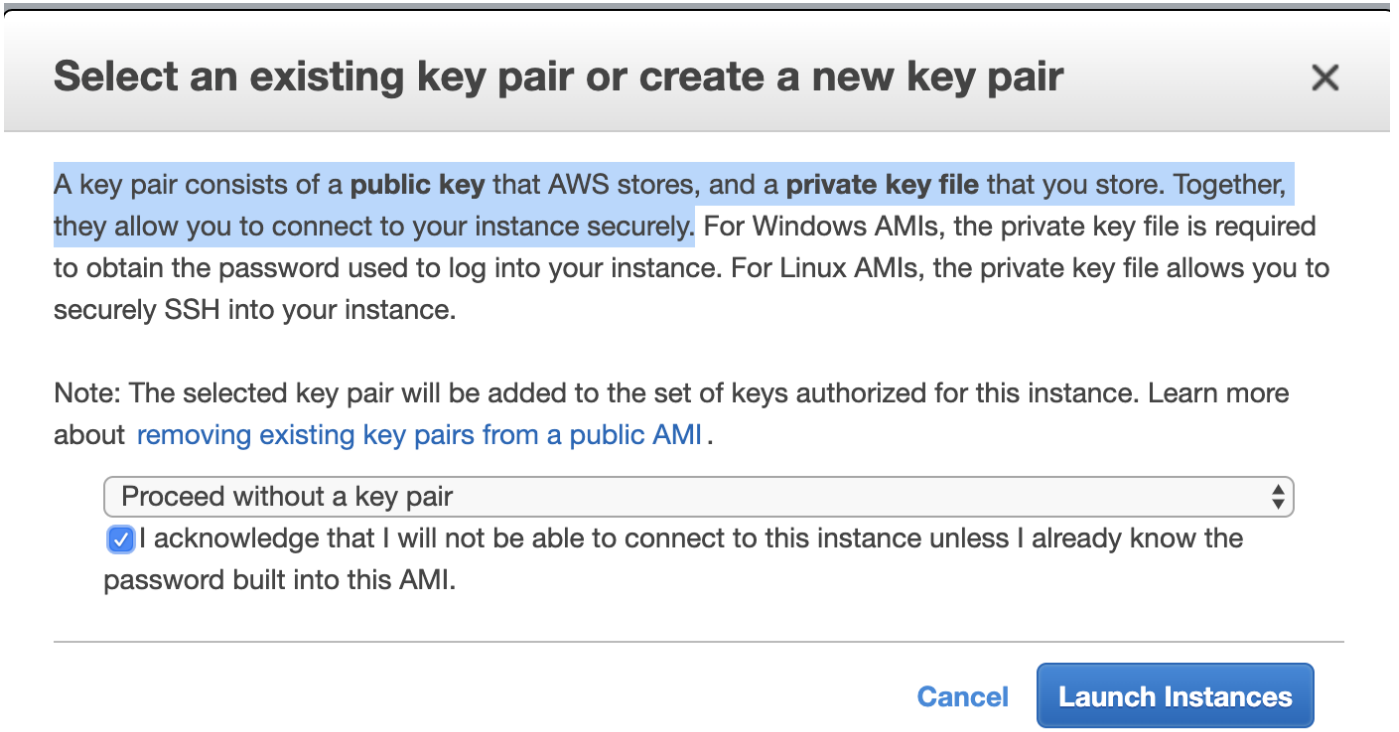
Since we do not need any further configuration we can click on the **Review and Launch** button.



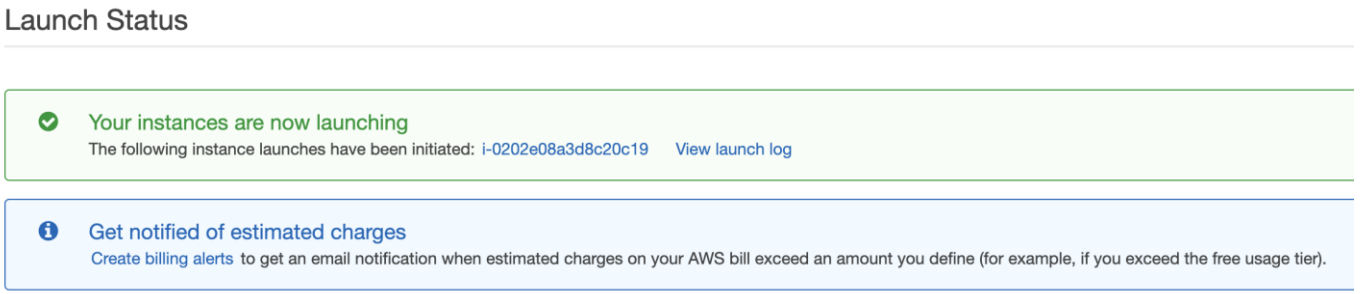
Choose to proceed without an SSH key pair.



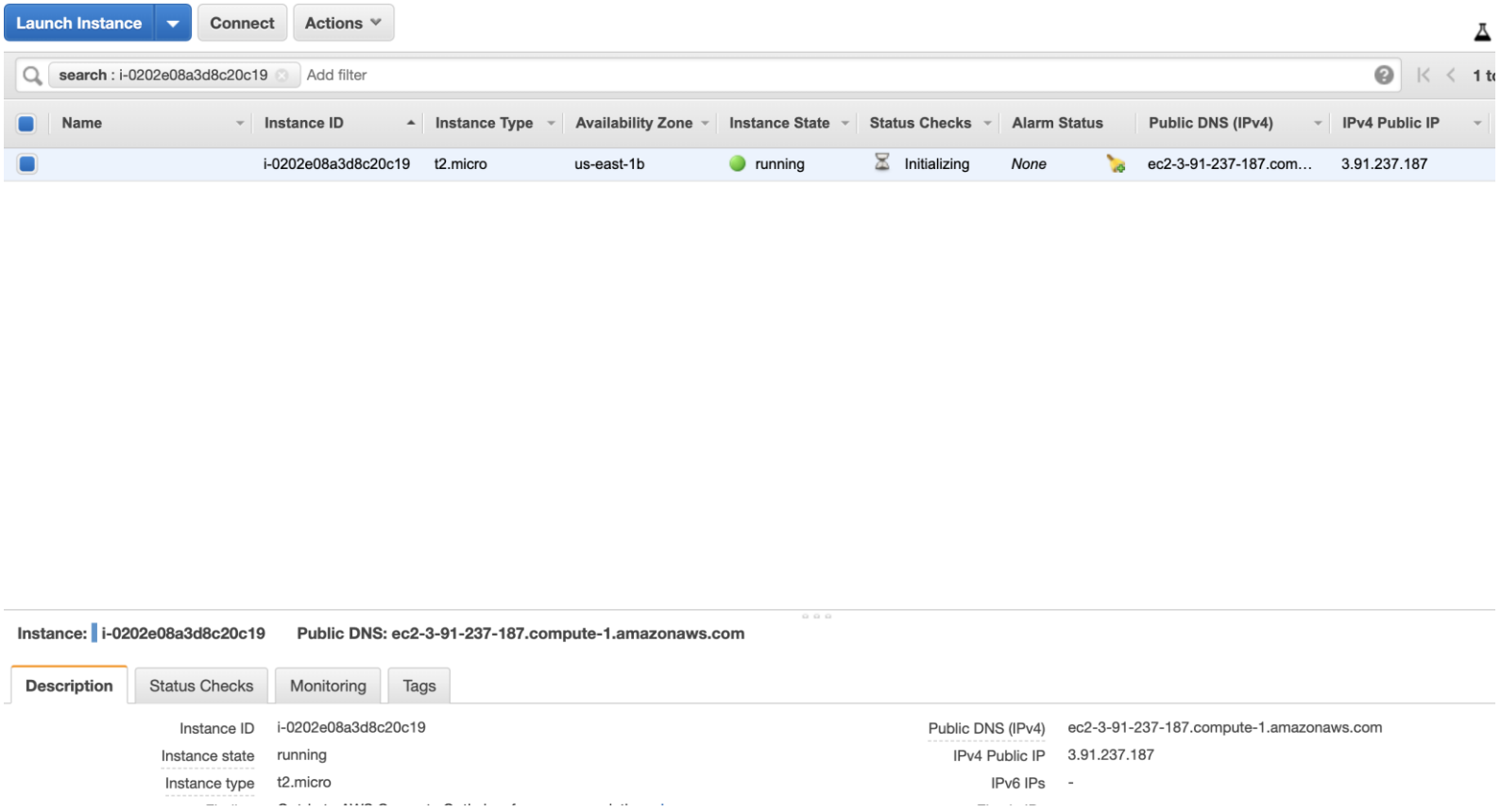
For this exercise, we can skip the creation of a new key pair, and acknowledge that we will not be able to connect to the instance.



Select **Proceed without a key pair** and check the box for **I acknowledge that...**. Then click on the **Launch Instances** button.



Every instance has an ID. If we click on the ID starting with the prefix **i-** we can see the instance.



In this page we can verify that:

- Our instance type is: **t2.micro**
- An **IPv4 Public IP** has been allocated
- The instance is **running**
  - The instance may first take several minutes to boot

Security Group

- Acts as a firewall
- Ingress rules can be added to allow traffic to specific services and ports
- Security groups are scoped to an instance level, not to a network or subnet
- You can specify ALLOW rules but you can't specify DENY rules
- You can control traffic via an IP address range (CIDR), or via security group IDs (for other AWS resources)

Exercise: Security Group

In this exercise, you will create a **security group** to allow traffic to an instance.

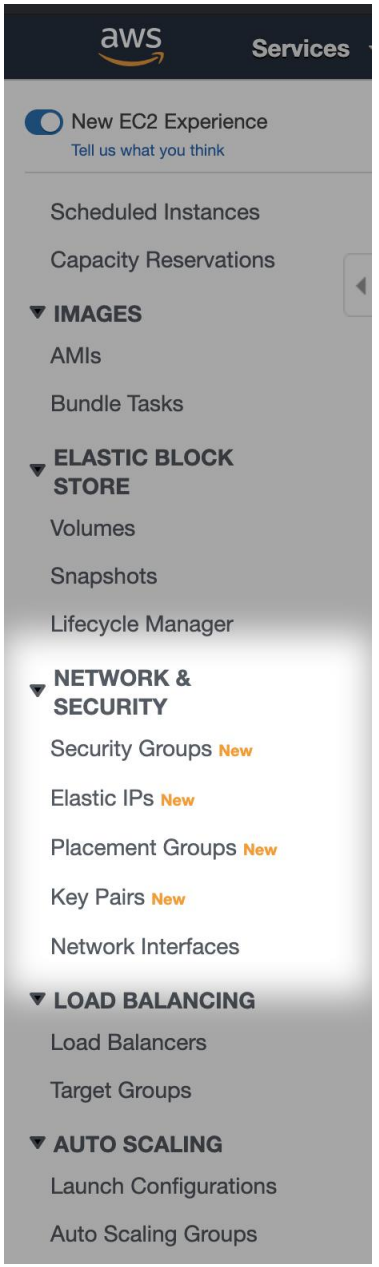
Instructions

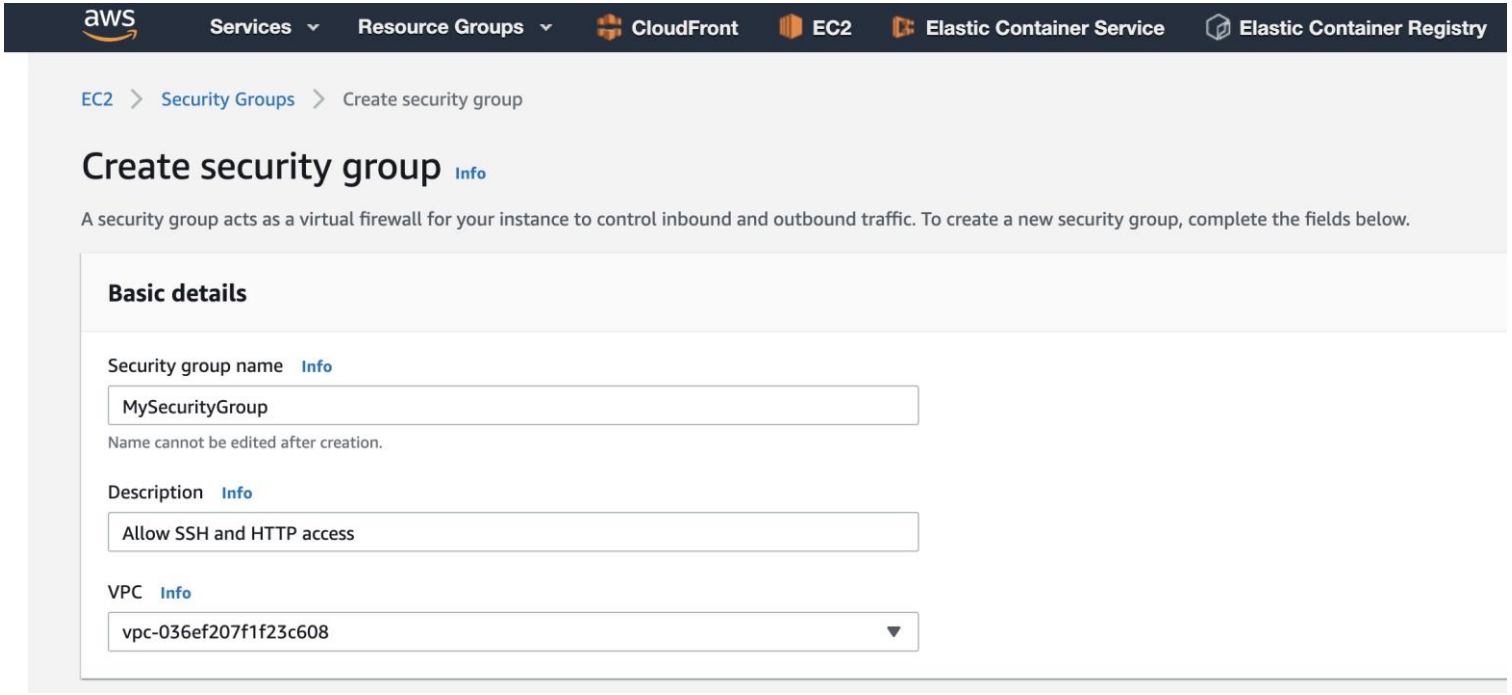
- If you have not done so already, launch an EC2 instance
- Create a new Security Group allowing **Port 80** and **Port 22** Inbound traffic from **anywhere** (any IP address)
- Attach the security group to the instance

Part 1: Create a Security Group

From the EC2 console, scan the sidebar to find the **Network & Security** section. Within that section click on **Security Groups**.

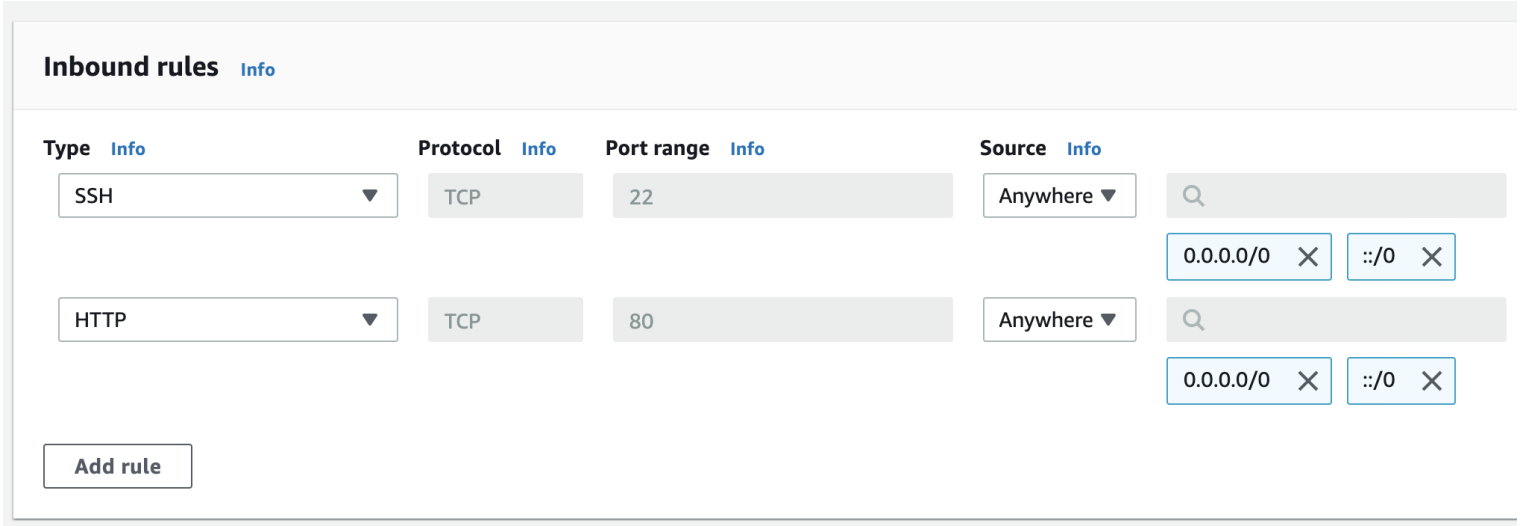
Click on the **Create security group** button
Fill the information for your security group name and description.





Click on the **Add Rule** button under the inbound section and add the following rules:

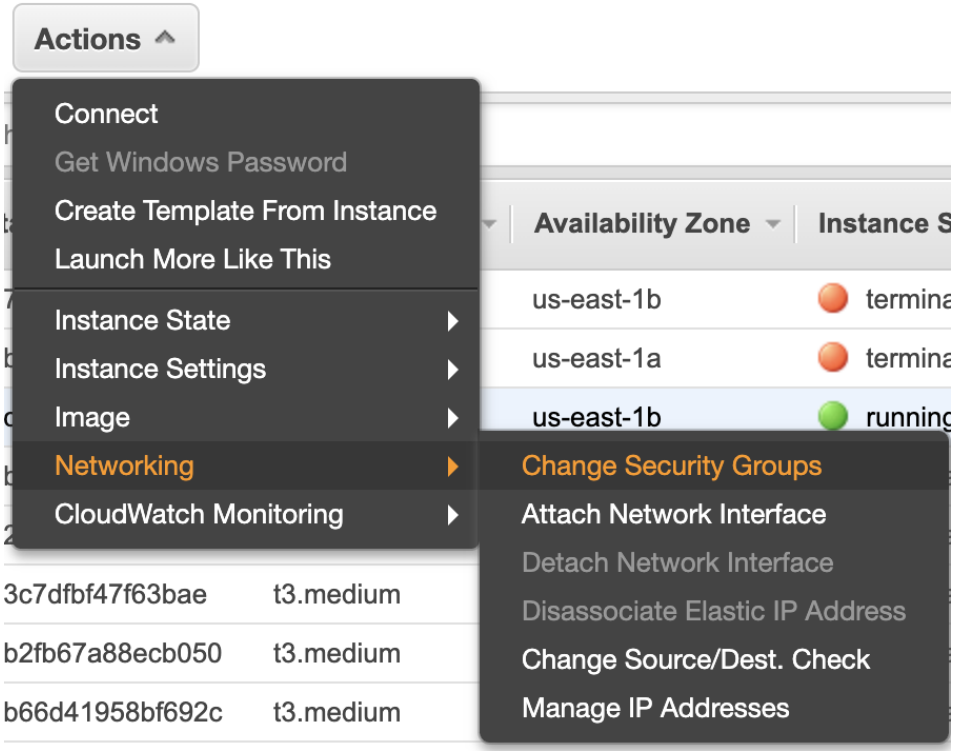
```
SSH TCP 22 Anywhere
HTTP TCP 80 Anywhere
```



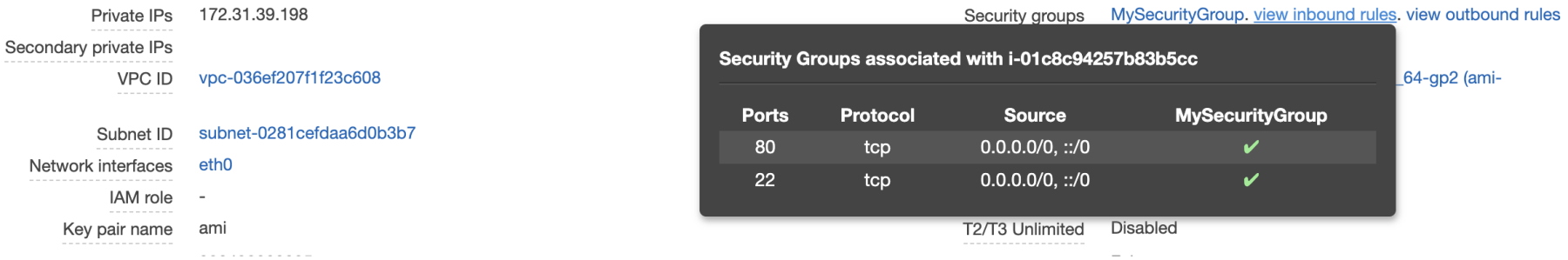
Click on the **Create security group** button to create the security group.

Part 2: Attach The Security Group To An EC2 Instance

Navigate via the sidebar to EC2 Instances Dashboard. Select the instance to which you want to add the security group.
From the **Actions** dropdown menu select **Networking**, followed by **Change Security Groups**.



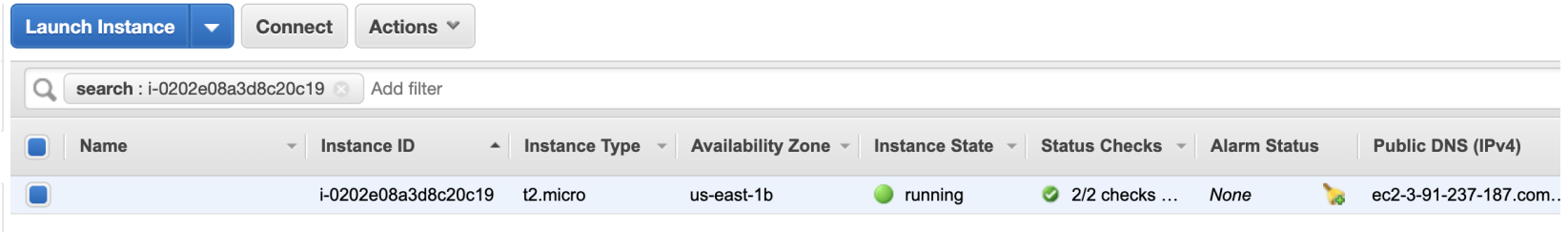
Select the security group you just created in Step 1 and click the **Assign security groups** button.
In the **Description** tab at the bottom of the screen, we can now see the security group.



Exercise: Terminate Instance

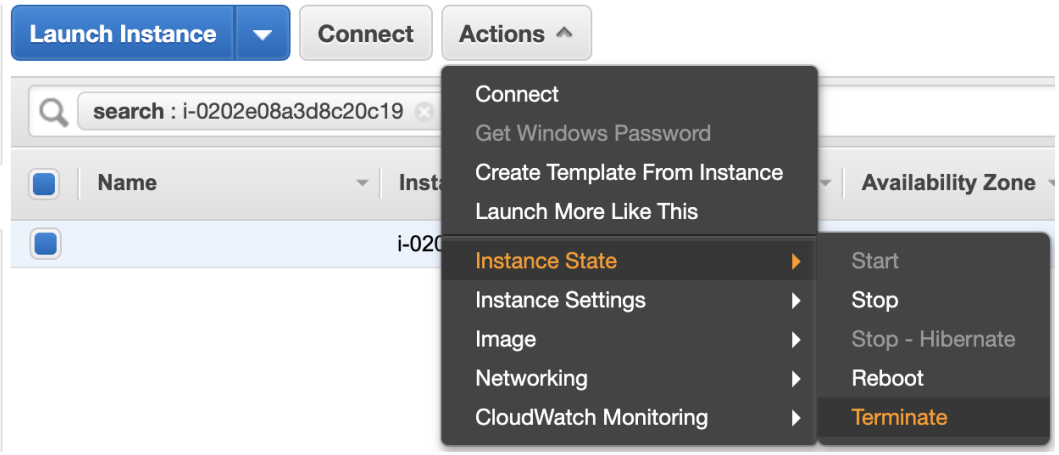
Now that you have practiced launching an EC2 instance, let's practice cleaning up (i.e. terminating) that instance. Cleaning up resources is important so that we don't incur any unexpected charges.

Select the instance ID from the AWS Instances console.



From the **Actions** dropdown menu, select **Instance State**, followed by **Terminate**.

Confirm that the instances reaches the **Terminated** state. This may take a few minutes.



Init Script

- An init script can be configured on instance launch page under userData section
- Can be used to run commands, install and configure an instance only during first boot
- Can be configured to run on every reboot
- Instances must have a public IP to install software from the internet
- Can be written in any language if the instance image AMI has the required software, mainly written in Bash

Exercise: Init Script

In this exercise, you will automatically provision an EC2 Instance using an init script, which runs at first boot.

Instructions
EC2 instance AMI: **Amazon Linux AMI 2**
Instance type: **t2.micro**
Instance should be open for traffic from **anywhere** on **port 80**
Instance must have a **public IP address** to be able to reach the internet and install **NGINX**
For this exercise, we can ignore SSH permissions, as we do not need them.
Use this code as the init-script:

```
#!/bin/bash

 yum update -y
 amazon-linux-extras install nginx1.12 -y
 service nginx start
```

Goal

We should be able to access the instance's public IP address via a web browser. The NGINX default page should appear: **Welcome to NGINX**.

Exercise: Init Script

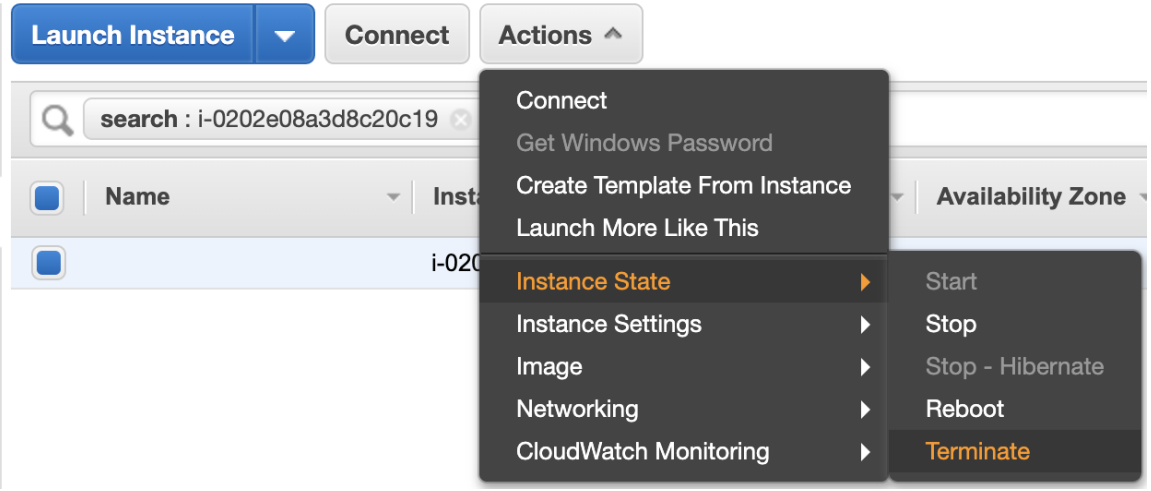
When you launch an EC2 instance, you have the option of passing **user-data** that you can use to configure or install software on your instance.

This is also known as an "Init script". For this exercise solution, we will program a shell script to be our init script.

By default, Init scripts run only on the first boot of the instance. We can configure an existing instance to run its init script on every restart, but we do not need that for this exercise.

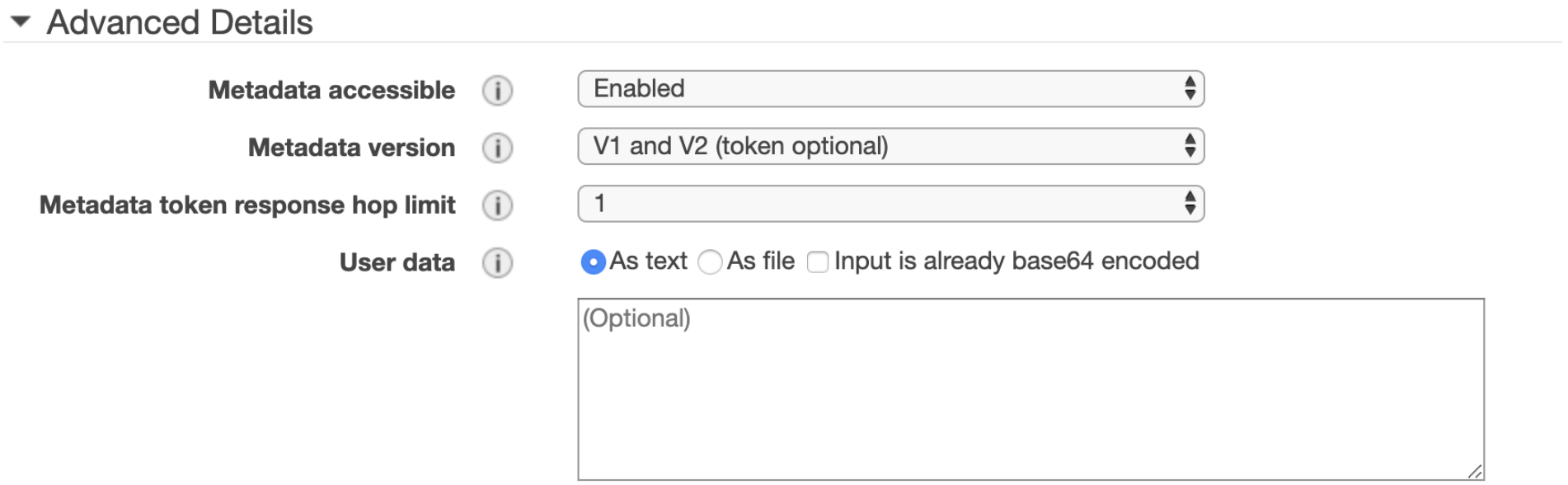
Part 0: Clean Up

If you haven't already, terminate all previous instances by selecting them in the EC2 Instances Dashboard. From the **Actions** dropdown menu select **Instance state**, followed by **Terminate**.



Part :1 Launch an EC2 Instance

1. Using the EC2 Console, select Instances from the sidebar and click **Launch instance**
2. Select **t2.micro** instance type and click the **Next: Configure Instance Details** button
3. Set the **Auto-assign public IP** to **Enabled**, because we need an Internet connection to install the NGINX software
4. Scroll down to the **Advanced Details** section, where we will configure the init script to install the NGINX server



5. Copy the following commands into the **User data** box

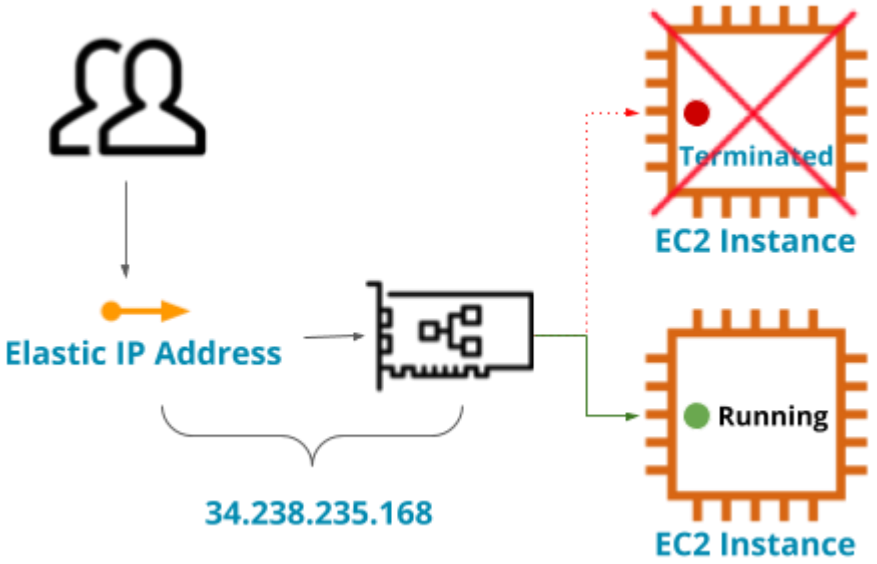
```
#!/bin/bash

yum update -y
amazon-linux-extras install nginx1.12 -y
service nginx start
```

6. Click **Review and Launch**, then edit the **Security Groups**
7. Either select an existing security group that allows HTTP access, or create a new such security group
8. Click **Review and Launch** and then launch the EC2 instance
9. Find the instance's **IPv4 Public IP** address from the EC2 Instance Dashboard
10. Visit the IP address in the browser and you should see the default Nginx page

Elastic IP

- Allocate a public, static IP address that can be associated with an instance
- Reachable from the internet
- Once associated, it becomes the public IP of the instance
- You can not have 2 public IP address per instance
- Can be disassociated from an instance
- You only pay for *unassociated* Elastic IPs



Exercise: Elastic IP

In this exercise, you will allocate an Elastic IP and assign it to your instance.

Instructions

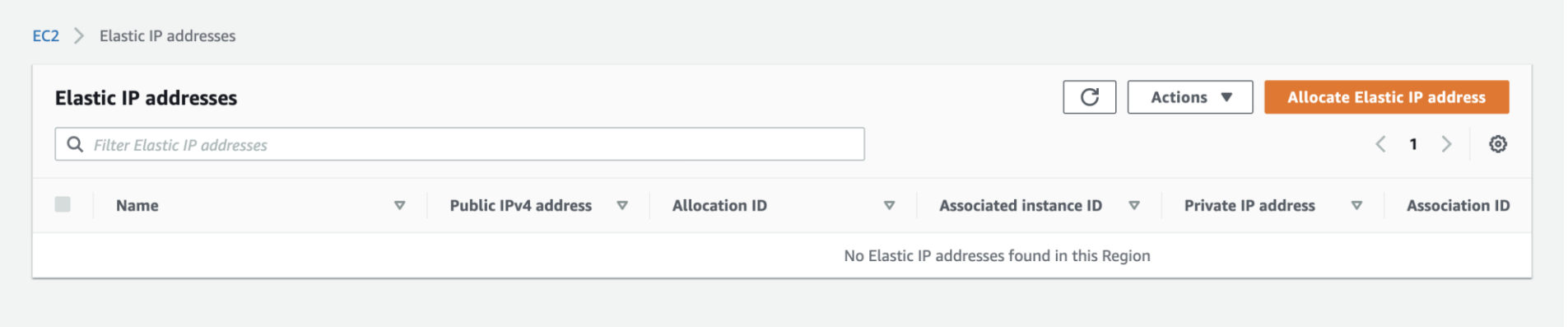
- If you have not done so already, launch an EC2 instance
- Allocate an **Elastic IP**
- Associate the Elastic IP with your EC2 instance
- Verify that the Elastic IP address works

Exercise: Elastic IP

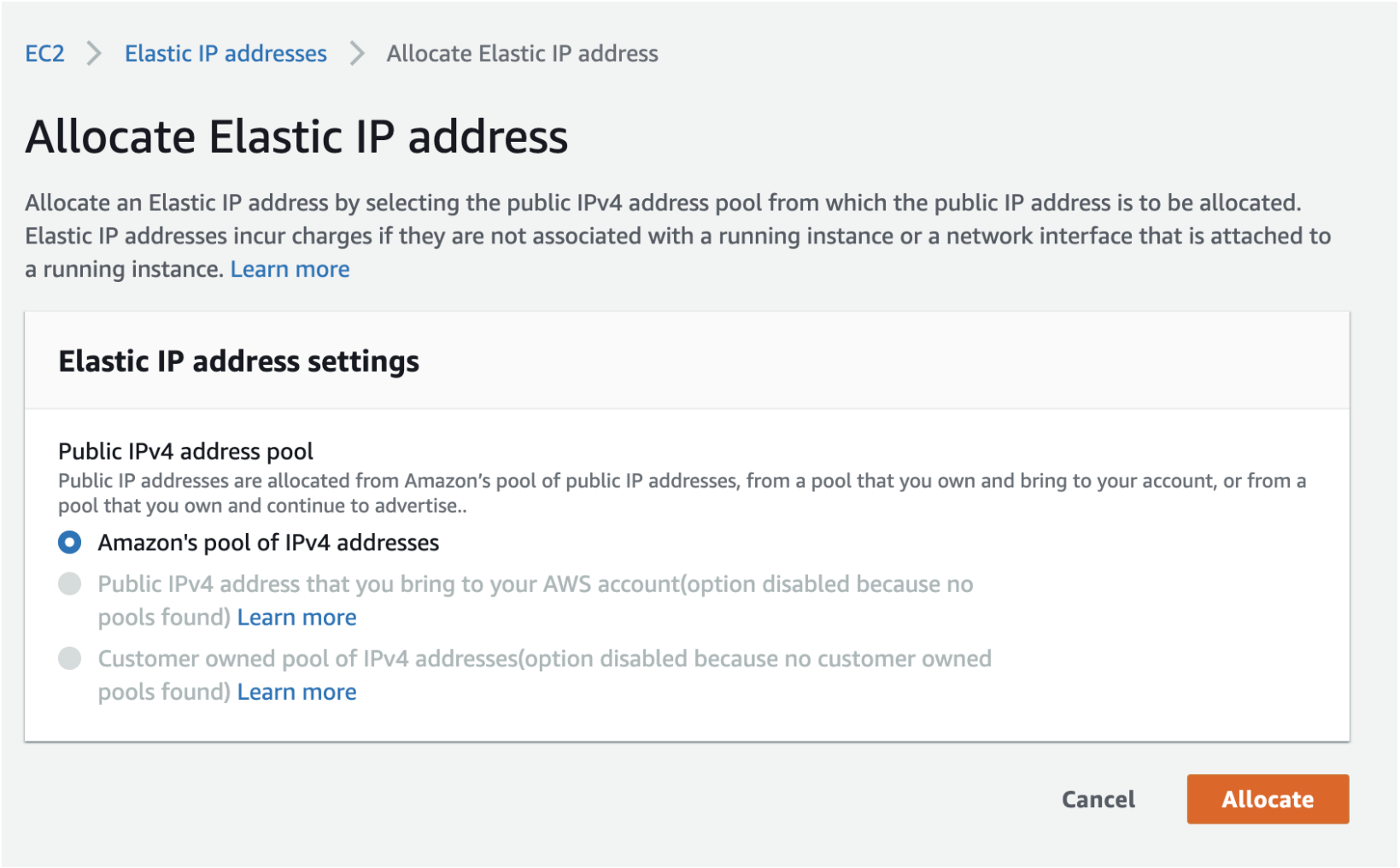
By default, EC2 instances do not get a public IP address. Even If you do assign an automatic public IP address in the Launch Wizard, the address will change on the next restart.

Elastic IP is a way to maintain the same external public IP address for an Instance or application.

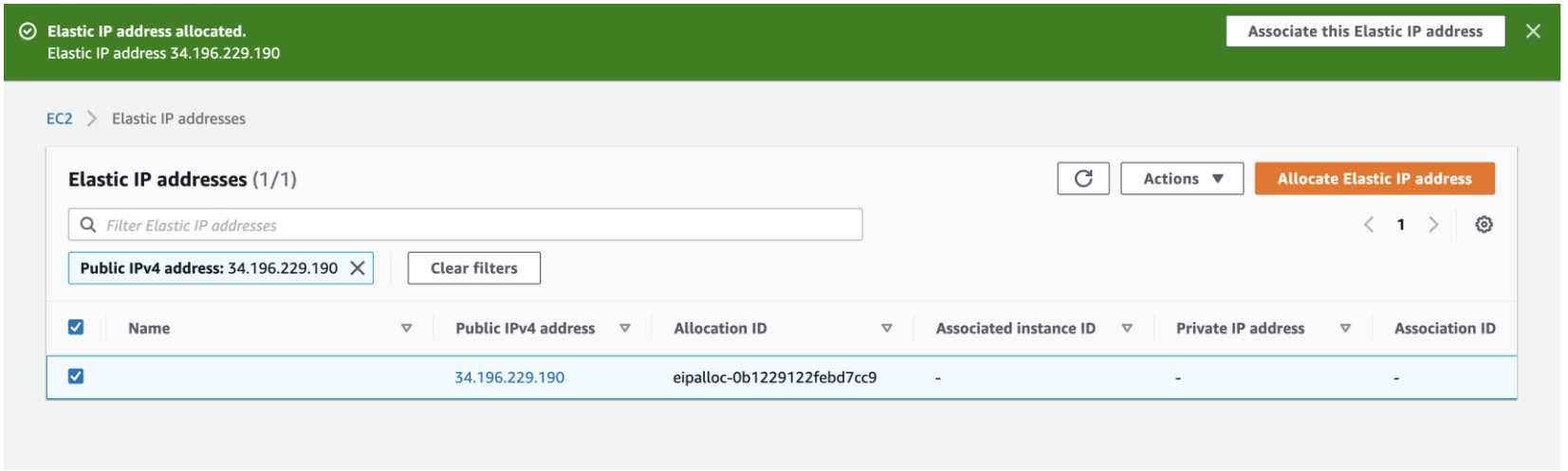
From the EC2 console sidebar menu, under the **Network & Security** section click on **Elastic IPs** .



Click on the **Allocate Elastic IP Address** button to allocate a new IP address out of the AWS IPv4 pool.

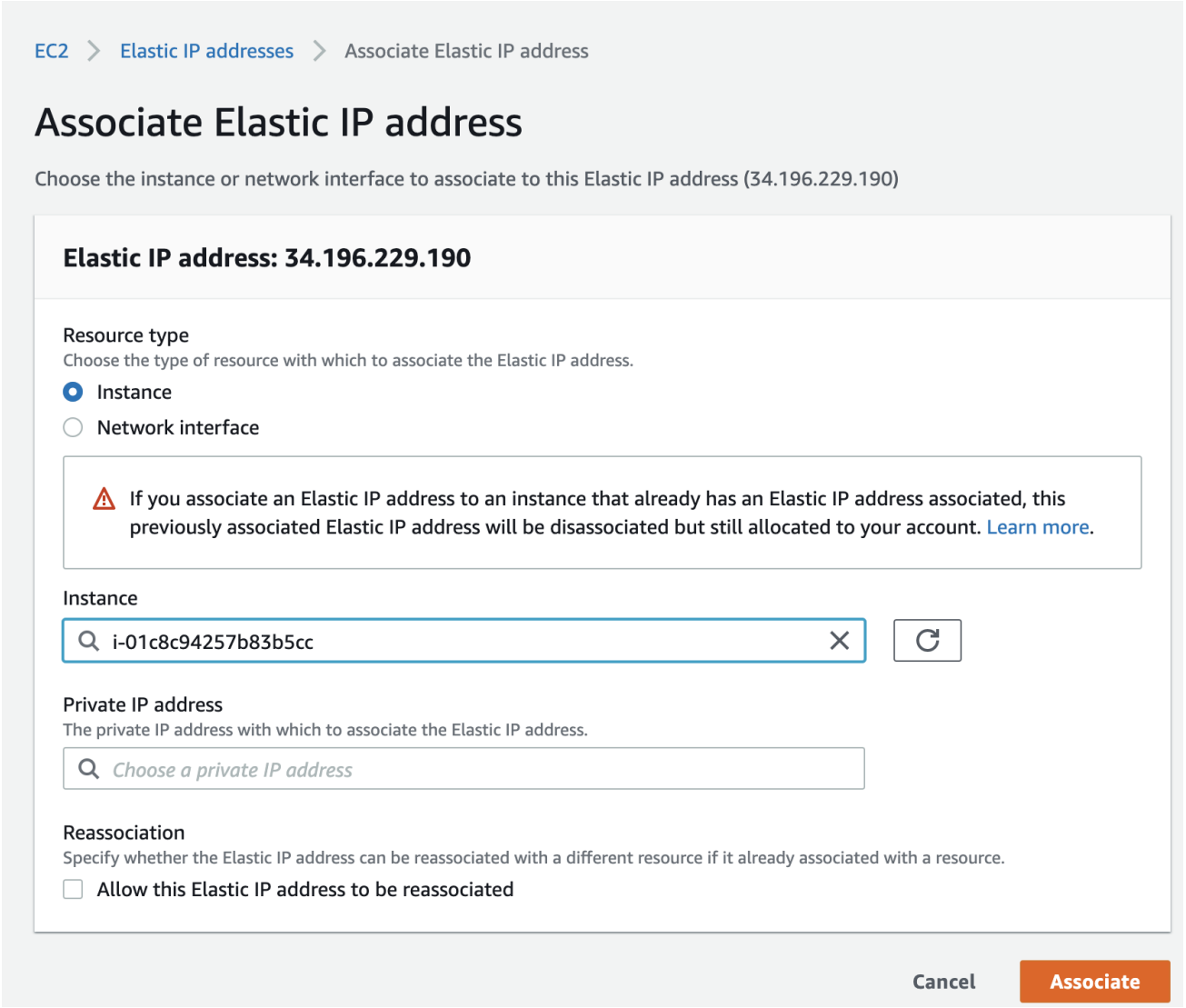


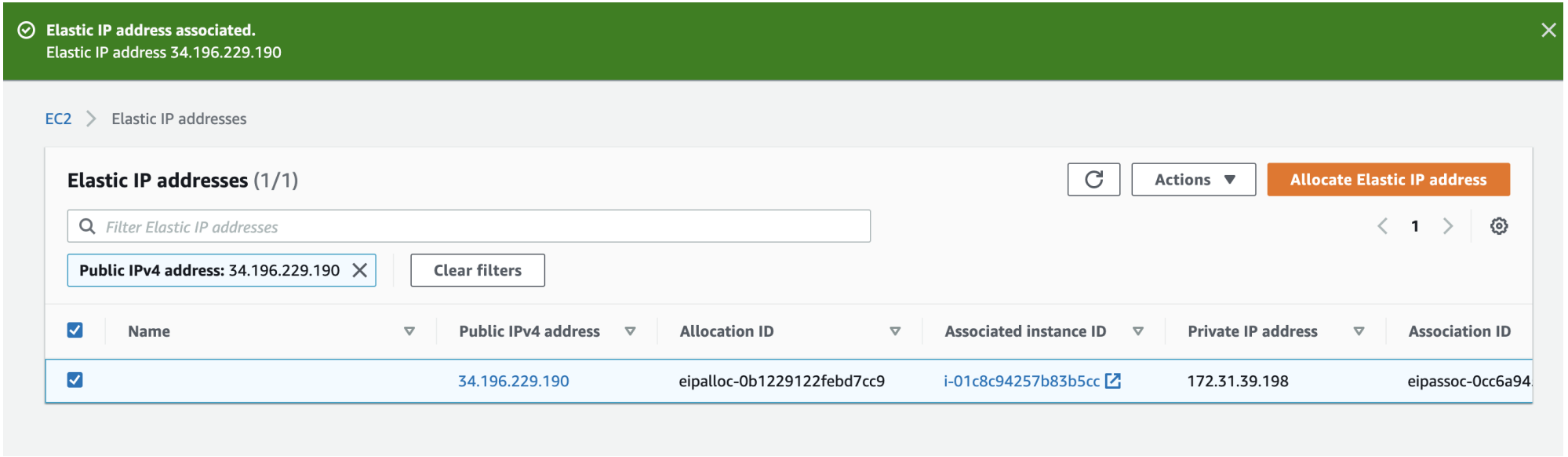
Click on the **Allocate** button.



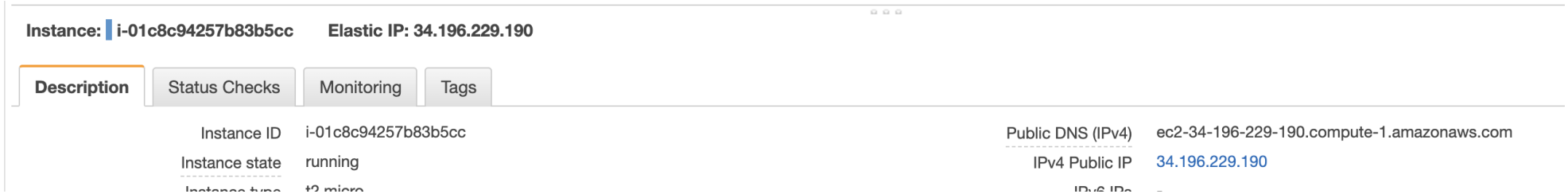
The newly-created Elastic IP address is not associated with any instance. Click on the **Associate this Elastic IP address** button, or select it and from the **Actions** dropdown and choose **Associate Elastic IP Address**.

In the next window, in the instance field, select the instance with which you want to associate the Elastic IP address.



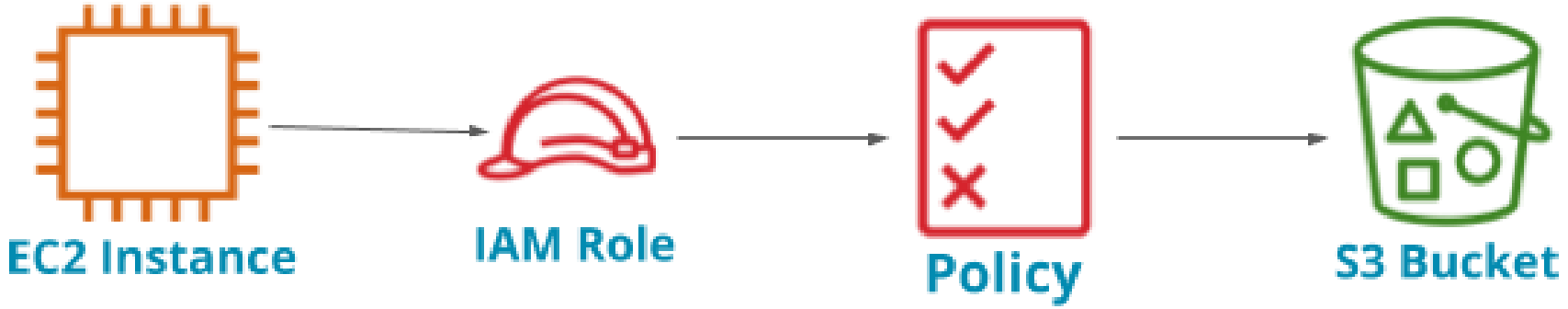


Once the instance is associated, you can click on **Associated Instance ID** to see the instance's details. On that screen, you can see the Elastic IP next to the instance ID. You can also verify that the Elastic IP address is the same as the instance's IPv4 Public IP.



IAM Role

- An IAM identity that does not have long-term credentials
- Can be assumed by a trusted service and operate on your behalf
- Designed for service-to-service communication
- Defines a set of permissions for accessing AWS services
- Is not bound to an identifiable user



Exercise: IAM Role

In this exercise, you will create an IAM role with a read-only policy to an S3 bucket.

Instructions

- If you have not already done so, create an S3 bucket with a folder inside
- Create an **IAM Role**
- Attach an **Inline IAM Policy** allowing **Read-Only** access to that **S3 bucket**

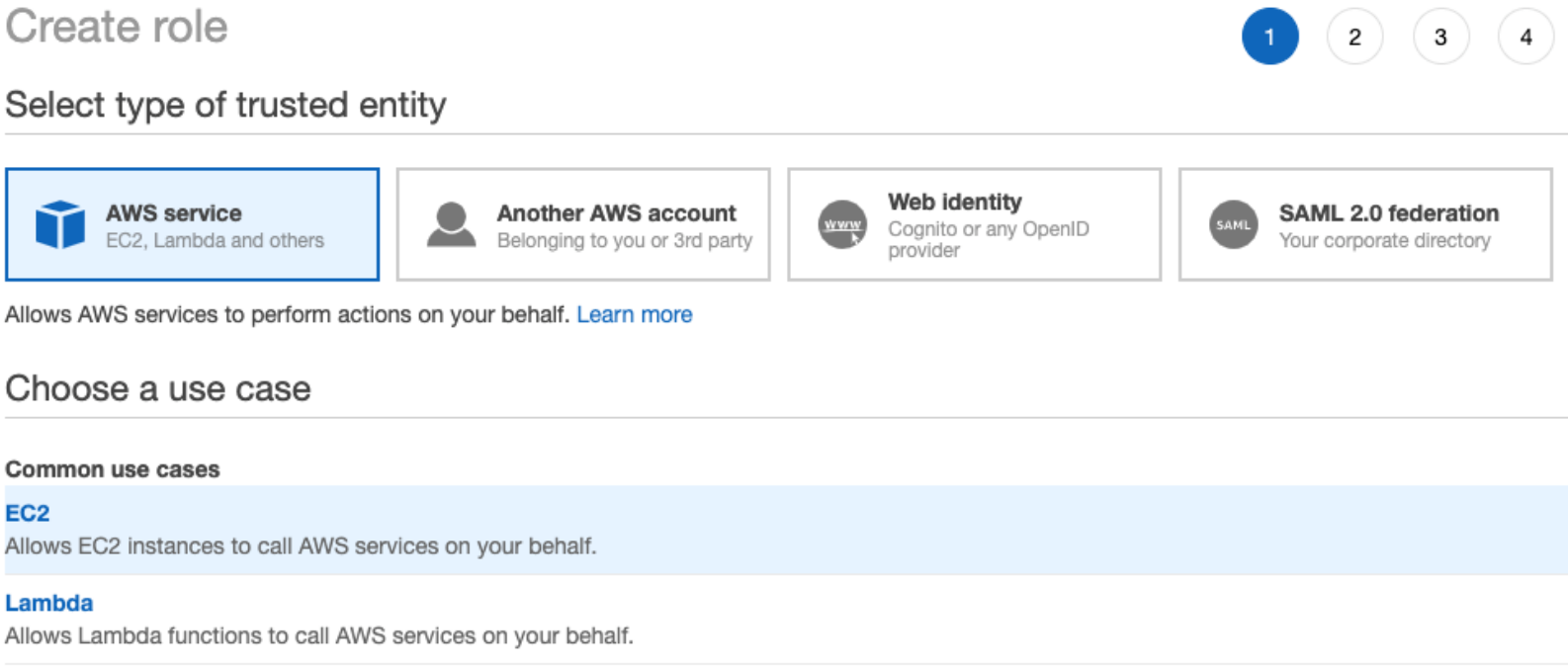
Exercise: IAM Role

An IAM Role is an IAM identity similar to IAM user. Instead of uniquely associating with a person, a role is designed for anyone who needs to assume it. This is an excellent fit for a service or a server-to-server identity.

A role is more secure than the long-lived credentials of an IAM user because when the role is assumed it provides temporary security credentials that expire and renew every 15 minutes

Create the IAM Role

1. Navigate to the **IAM console**
2. Click **Roles** on the sidebar menu
3. Click on the **Create Role** button
4. Since we intended to use the role on an EC2 in the next exercise, choose **EC2** as the use case, and click on **Next: Permissions** button

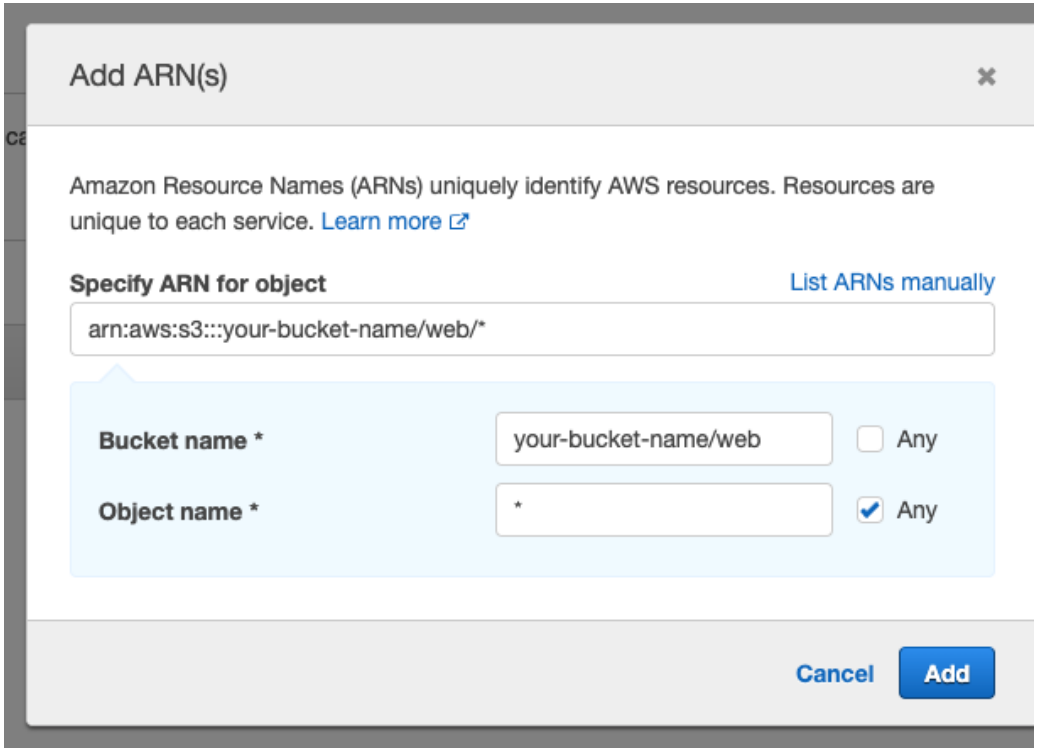


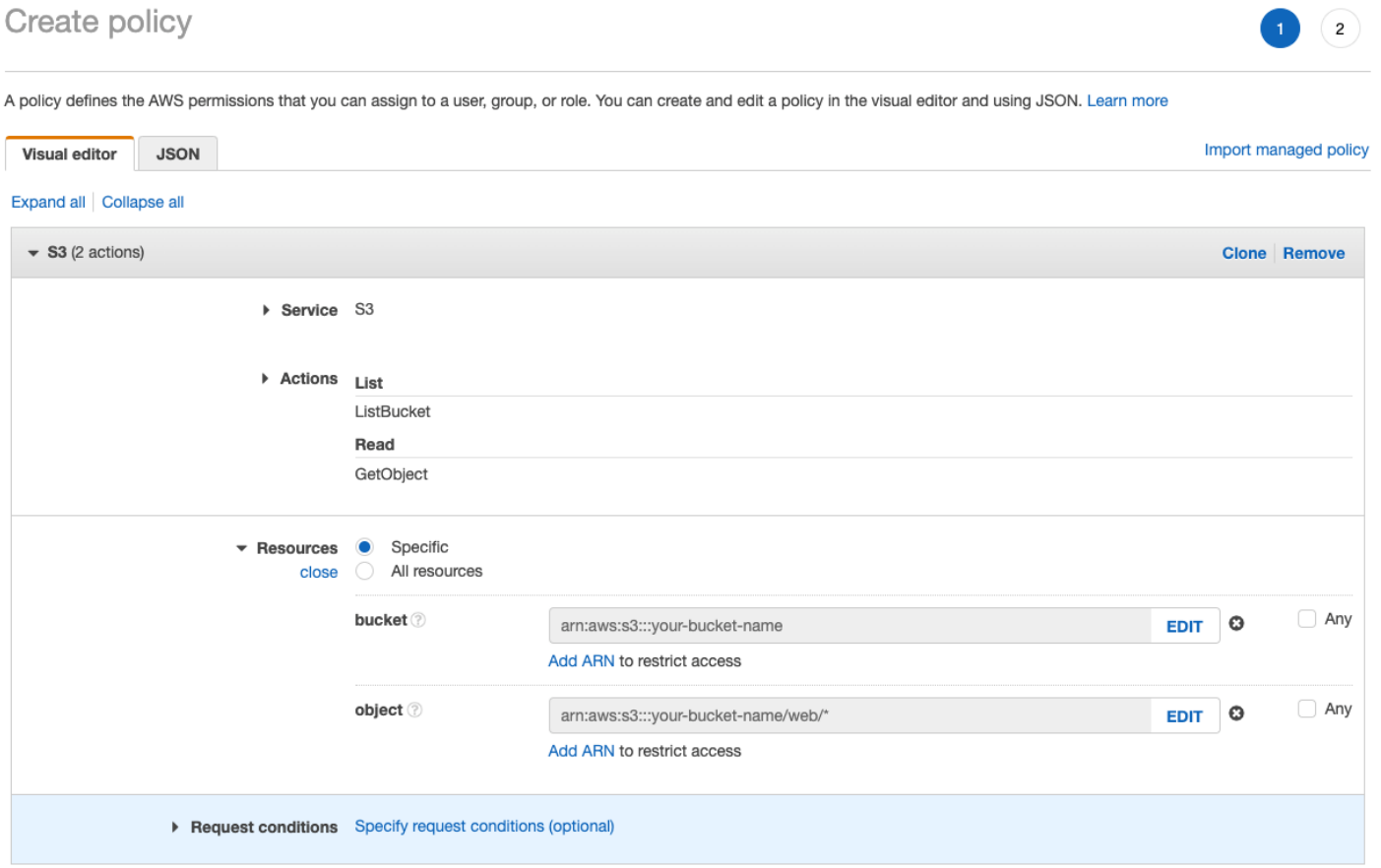
5. If we were to click on the **Create Policy** button, the role creation would be canceled, so instead click on **Next** until you get to the review page
6. Name the role **web** and click on the **Create Role** button

Creating and Attaching an IAM Policy to the Role

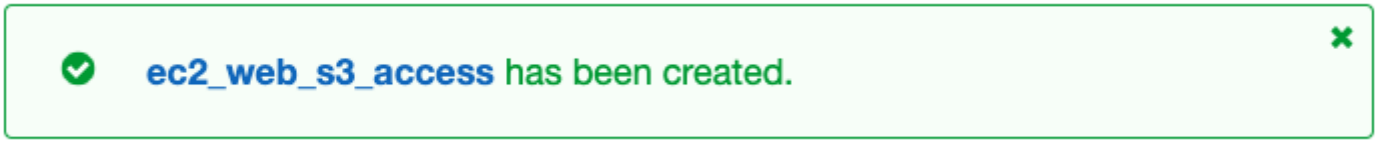
Now that we have a role, let's create an IAM policy so that it can access our S3 bucket.

1. From the sidebar menu click on **Policies** and then click on the **Create Policy** button
2. Select **S3** for the service
3. On the **Actions** field select **ListBucket** and **GetObject**
4. Click on Resources, while it is set to **specific** click the **Add ARN** for the **bucket** and add your bucket name (instead of *<your-bucket-name>* on image below)
5. Click on the **Add ARN** for the object, set the **bucket name** again to your bucket name and set the **object name** to "web/*"





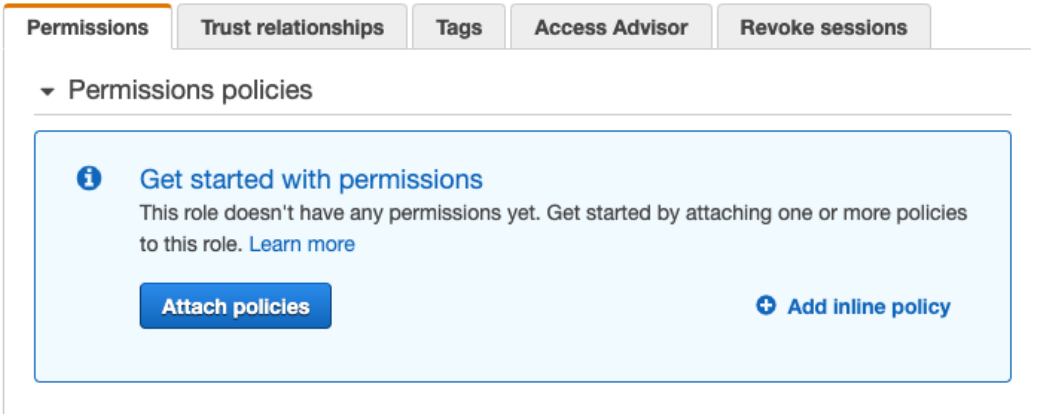
7. Click on the **Review policy** button, set a name (for example, "ec2_web_s3_access") and click the **Create policy** button



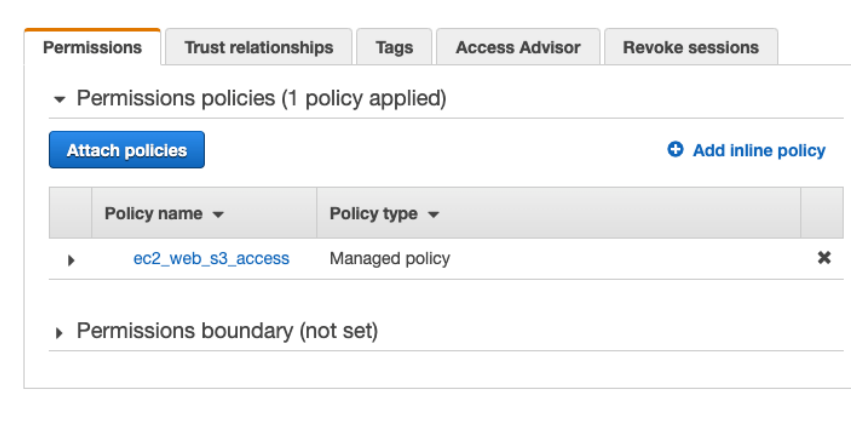
Attaching the Policy to the IAM Role

1. Click on the **Roles** on the sidebar menu of the IAM console
2. Search for the role "web"

3. Click on the role name to edit the role
4. Click on the **Attach policies** button



5. Search for "web" to find the policy you just created
6. Select it and click the **Attach policies** button



Overriding NGINX Defaults

NGINX comes with a default web page to verify that the webserver is working.

Instead of configuring a new site for NGINX, we can copy a website to this default location. Our new website will replace the existing NGINX default.

We will store this content in an S3 bucket under the **web** folder. In order to access this content, the following prerequisite must exist:

- S3 bucket storing the zip file for the SB Admin 2 theme
- IAM role and policy for EC2 to access the bucket

Provisioning Script
Note: Make sure to change "BUCKET_NAME" to the name of the bucket you created

```
#!/bin/bash

   BUCKET_NAME=<s3-bucket-name-place-holder>
   yum update -y
   amazon-linux-extras install nginx1.12 -y
   service nginx start
   aws s3 cp s3://${BUCKET_NAME}/web/startbootstrap-sb-admin-2-gh-pages.zip /tmp/
   cd /tmp
   unzip start*
   mv startbootstrap-sb-admin-2-gh-pages html
   rm -rf /usr/share/nginx/html
   mv /tmp/html /usr/share/nginx/
```

Exercise: Overriding NGINX Defaults

In this exercise, you will launch an EC2 instance with an IAM role and pull static website content from an S3 bucket to overwrite the default NGINX website.

Exercise Requirements

Prerequisites

- Download zip file for Bootstrap SB2 Admin Theme Site
- Upload the zip file to your **S3** bucket under a folder called **web** (do not extract it)

EC2 Requirements

- **AMI**: Amazon Linux 2
- **IP Address**: Public
- **Type**: t2.micro
- **IAM Role**: web (created in the previous exercise)
- **Security group**: accessible from **anywhere** on **port 80**
- **Init Script**: see below

Exercise: Overriding NGINX Defaults

Init Script

Copy the following script into the **User data** text box in the EC2 Instance Launch Wizard.

Note: you must change the to your bucket name (with no spaces after the = sign)

```
#!/bin/bash

   BUCKET_NAME=<s3-bucket-name-place-holder>
   yum update -y
   amazon-linux-extras install nginx1.12 -y
   service nginx start
   aws s3 cp s3://${BUCKET_NAME}/web/startbootstrap-sb-admin-2-gh-pages.zip /tmp/
   cd /tmp
   unzip start*
   mv startbootstrap-sb-admin-2-gh-pages html
   rm -rf /usr/share/nginx/html
   mv /tmp/html /usr/share/nginx/
```

Goal

You should be able to see the SB Admin 2 web page with images displayed in the browser.

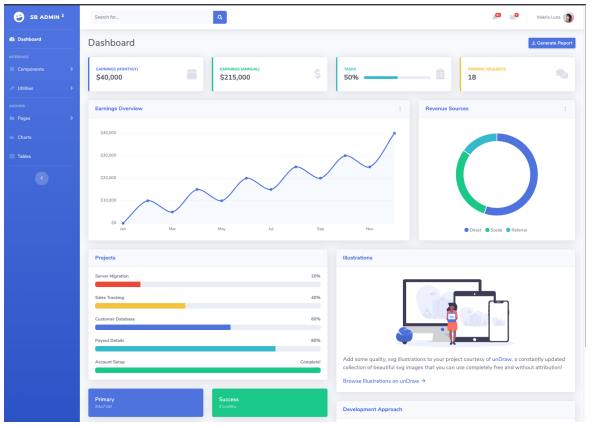
Exercise: Overriding NGINX Defaults

Init Script Explanation
This is not necessary to understand in order to complete the exercise. But in case you are interested, here is what the init script does.

The init script installs Nginx web server.

Then it uses the **aws-cli s3 cp command** to copy the **Bootstrap Theme** zip file from **S3** into the **/tmp** directory on the EC2 instance. Next, the init script extracts the file using the **unzip** command.

Finally, the init script overrides the contents of the **/usr/share/nginx/html** directory with the contents of the unzipped folder.



Upload to S3

1. Navigate to the S3 console and select your bucket
2. Create a folder called web in your bucket
3. Upload the SB Admin 2 bootstrap theme zip file to the web folder

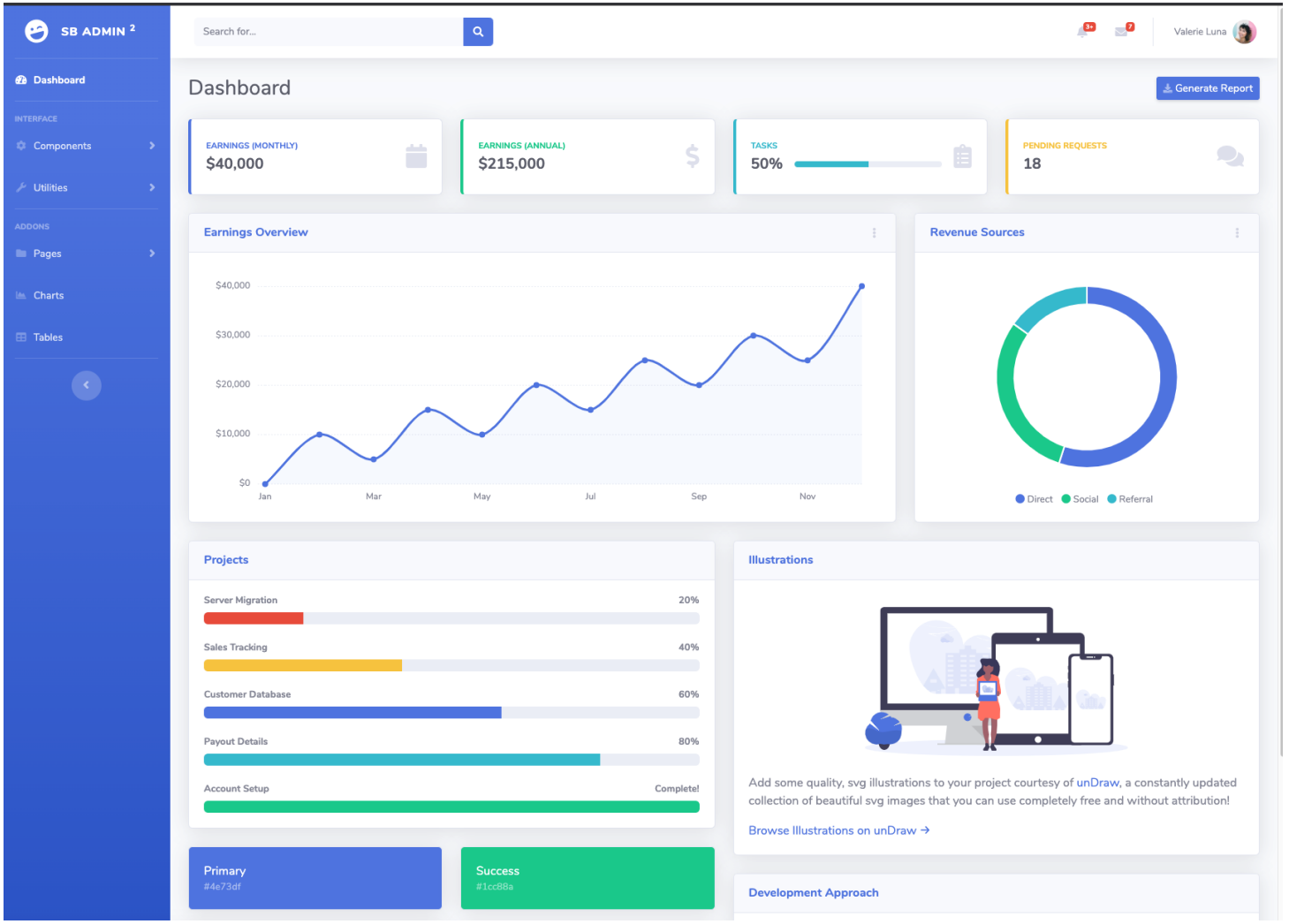
Launching an EC2 instance using the EC2 Console

1. Navigate to the EC2 Instances Dashboard
2. Click on the **Launch instance** button
3. Choose the **Amazon Linux 2 AMI (HVM), SSD Volume Type 64-bit** image
4. Leave the default instance type as **t2.micro** and click **Next: Configure instance details**
5. Set the **Auto-assign Public IP** to **enabled**
6. Set the **IAM role** to be **web**
7. Copy the following script into the **User data** text box (**Note: you must change the to your bucket name (with no spaces after the = sign)**)

```
#!/bin/bash

   BUCKET_NAME=<s3-bucket-name-place-holder>
   yum update -y
   amazon-linux-extras install nginx1.12 -y
   service nginx start
   aws s3 cp s3://${BUCKET_NAME}/web/startbootstrap-sb-admin-2-gh-pages.zip /tmp/
   cd /tmp
   unzip start*
   mv startbootstrap-sb-admin-2-gh-pages html
   rm -rf /usr/share/nginx/html
   mv /tmp/html /usr/share/nginx/
```

8. Click on the **Review and Launch** button and then click on the **Launch** button
9. Wait about 20-30 seconds for the instance to come up and for the script to finish running
10. Using a browser, visit the web page at the public IP address



When Is EC2 Sub-Optimal?

EC2 resources (memory, CPU) are configured by instance types.

If a process is short-lived or does not need to be running constantly, and its footprint is much smaller than the EC2 instance resources. It might be better to use serverless functions rather than EC2, as a serverless approach will take fewer resources and is designed for short-lived processes.

Further Reading

Documentation
IAM Getting Started
IAM roles for EC2
EC2 Getting Started
EC2 best practices
Security Groups

Books
Programming EC2
Research
Best Practices in Evaluating Elastic Load Balancing White Paper
Blog Posts
EC2 for beginners
AWS IAM Summary
EC2 T2 Insatnces
Advanced Topics
AWS Nitro System - the system beyond standard virtualization
Auto Scaling
Elastic Load Balancing