

Securing Root User With MFA

AWS Management Console

AWS services

Find Services
You can enter names, keywords or acronyms.

[All services](#)

Build a solution
Get started with simple wizards and automated workflows.

Launch a virtual machine With EC2 2-3 minutes 	Build a web app With Elastic Beanstalk 6 minutes 	Build using virtual servers With Lightsail 1-2 minutes 	Register a domain With Route 53 3 minutes 
Connect an IoT device With AWS IoT 5 minutes 	Start migrating to AWS With CloudEndure Migration 1-2 minutes 	Start a development project With CodeStar 5 minutes 	Deploy a serverless microservice With Lambda, API Gateway 2 minutes 

[▶ See more](#)

Stay connected
My Account
My Organization
My Service Quotas
My Billing Dashboard
Orders and Invoices
My Security Credentials 
[Sign Out](#)

Android mobile device. [Learn more](#)

Explore AWS

AWS DeepRacer F1 ProAm
Test your machine learning skills against F1's finest in Circuit de Barcelona-Catalunya [Learn more](#)

Free Digital Training
Get access to 350+ self-paced online courses covering AWS products and services. [Learn more](#)

S3 Intelligent-Tiering
Optimize costs automatically with Amazon S3. [Get started](#)

Amazon SageMaker Autopilot
Get hands-on with this Auto-ML workshop. [Learn more](#)

Securing Root User With MFA

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and a star icon. The left sidebar is titled 'Identity and Access Management (IAM)' and contains the following navigation items:

- Dashboard
- Access management
 - Groups
 - Users
 - Roles
 - Policies
 - Identity providers
- Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Below the sidebar is a search bar with the placeholder 'Search IAM' and an AWS account ID field containing '915464768551'. The main content area is titled 'Your Security Credentials' and includes the following text:

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#). To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference.

Use MFA to increase the security of your AWS environments. Signing in to MFA-protected accounts requires a user name, password, and an authentication code from an MFA device.

[Activate MFA](#)

▼ Password

▼ Multi-factor authentication (MFA)

Use MFA to increase the security of your AWS environments. Signing in to MFA-protected accounts requires a user name, password, and an authentication code from an MFA device.

[Activate MFA](#)

▼ Access keys (access key ID and secret access key)

▼ CloudFront key pairs

▼ X.509 certificate

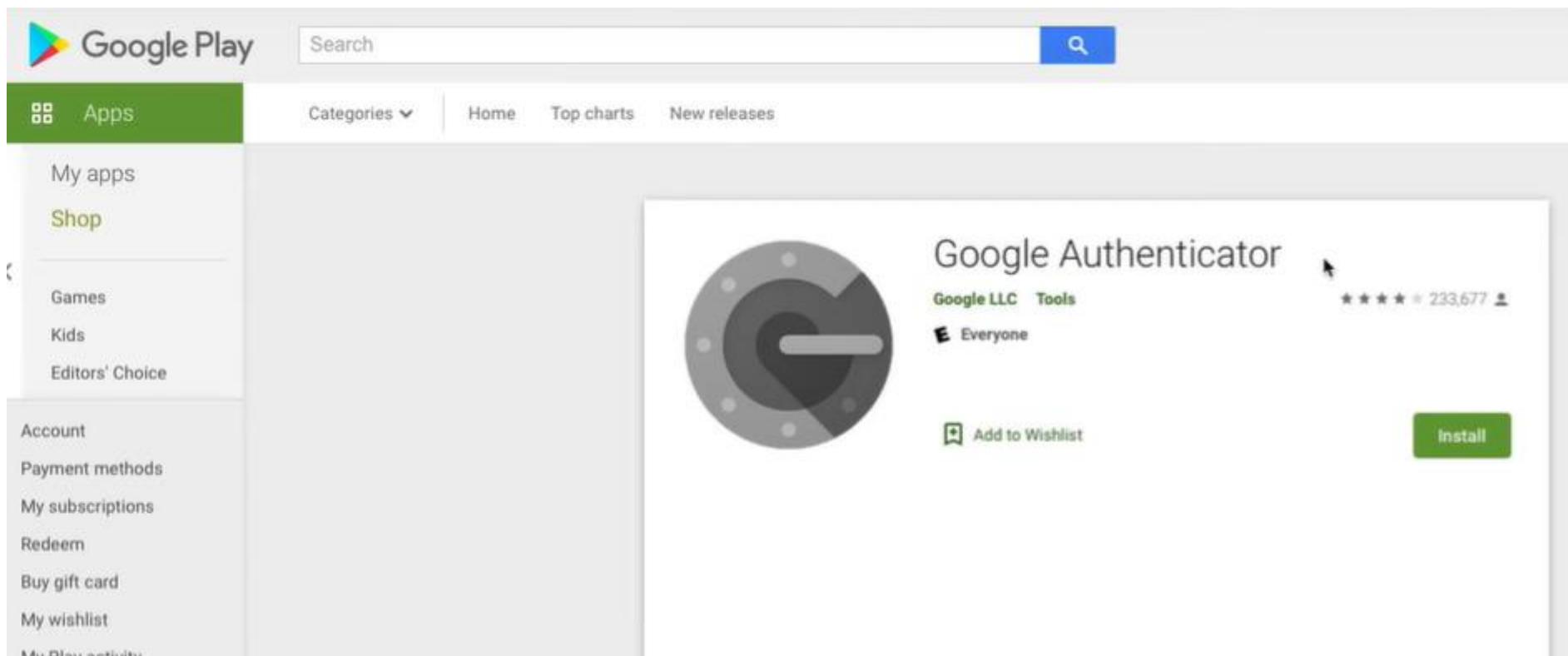
▼ Account identifiers

Securing Root User With MFA

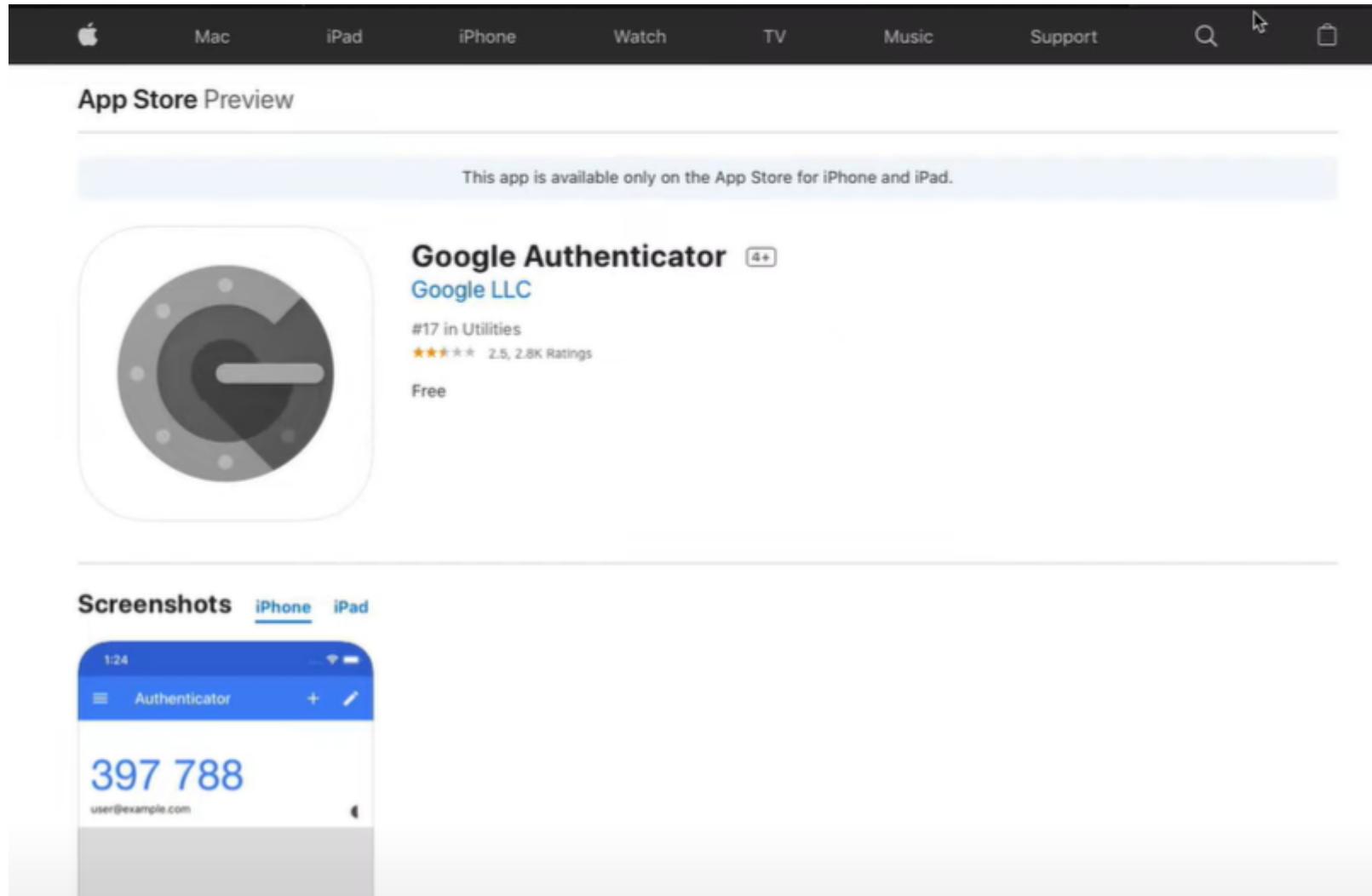
The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar lists various IAM management options like Groups, Users, Policies, and Access reports. The main content area is titled 'Your Security Credentials' and provides instructions for managing credentials. A modal window titled 'Manage MFA device' is open, prompting the user to choose the type of MFA device. The 'Virtual MFA device' option is selected, with a note explaining it's an Authenticator app installed on a mobile device or computer. There is also a warning message stating 'Your browser does not support U2F. Learn more' next to the 'U2F security key' option. The 'Other hardware MFA device' option is also listed. At the bottom of the modal, there are 'Cancel' and 'Continue' buttons.

Securing Root User With MFA

Install Google Authenticator app.



Securing Root User With MFA



The screenshot shows the App Store preview for the "Google Authenticator" app. At the top, there's a navigation bar with links for Mac, iPad, iPhone, Watch, TV, Music, Support, a search icon, and a shopping bag icon. Below the navigation bar, it says "App Store Preview". A message states, "This app is available only on the App Store for iPhone and iPad." The main section features the app's icon (a circular QR code-like pattern), the app's name "Google Authenticator" with a 4+ rating, the developer "Google LLC", its rank "#17 in Utilities", its rating "★★★★★ 2.5, 2.8K Ratings", and its price "Free". Below this, there's a "Screenshots" section with tabs for iPhone and iPad. An iPhone screenshot shows the app's interface with a large blue six-digit code "397 788" and the email "user@example.com".

Securing Root User With MFA

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar lists various IAM management options like Groups, Users, Roles, Policies, and CloudFront key pairs. The main content area is titled 'Your Security Credentials' and provides instructions for managing AWS credentials. A modal window titled 'Set up virtual MFA device' is open, guiding the user through three steps: 1. Install a compatible app on your mobile device or computer (with a link to a list of compatible applications). 2. Use your virtual MFA app and your device's camera to scan the QR code (with a 'Show QR code' button). 3. Type two consecutive MFA codes below (with input fields for 'MFA code 1' and 'MFA code 2'). At the bottom of the modal are 'Cancel', 'Previous', and 'Assign MFA' buttons.

Securing Root User With MFA

Screenshot of the AWS IAM "Your Security Credentials" page showing the "Multi-factor authentication (MFA)" section and an open "Set up virtual MFA device" modal.

The modal provides instructions for setting up a virtual MFA device:

1. Install a compatible app on your mobile device or computer
[See a list of compatible applications](#)
2. Use your virtual MFA app and your device's camera to scan the QR code

A large QR code is displayed for scanning. An alternative method is provided: "Alternatively, you can type the secret key. [Show secret key](#)".

Below the QR code, there are fields for entering two consecutive MFA codes:

- MFA code 1: 640152
- MFA code 2: 736525

At the bottom right of the modal are buttons: "Cancel", "Previous", and "Assign MFA". The "Assign MFA" button is highlighted with a cursor.

Securing Root User With MFA

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar lists various IAM management options like Dashboard, Access management, Policies, and Access reports. The main content area is titled 'Your Security Credentials' and displays a success message: 'Set up virtual MFA device' with a green checkmark and the text 'You have successfully assigned virtual MFA'. It also states 'This virtual MFA will be required during sign-in.' Below this, there's a table with two columns: 'Device type' and 'Serial number'. A single entry is shown: 'Virtual' under Device type and 'arn:aws:iam::915464768551:mfa/root-account-mfa-device' under Serial number. Other sections listed include Access keys, CloudFront key pairs, X.509 certificate, and Account identifiers.

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"

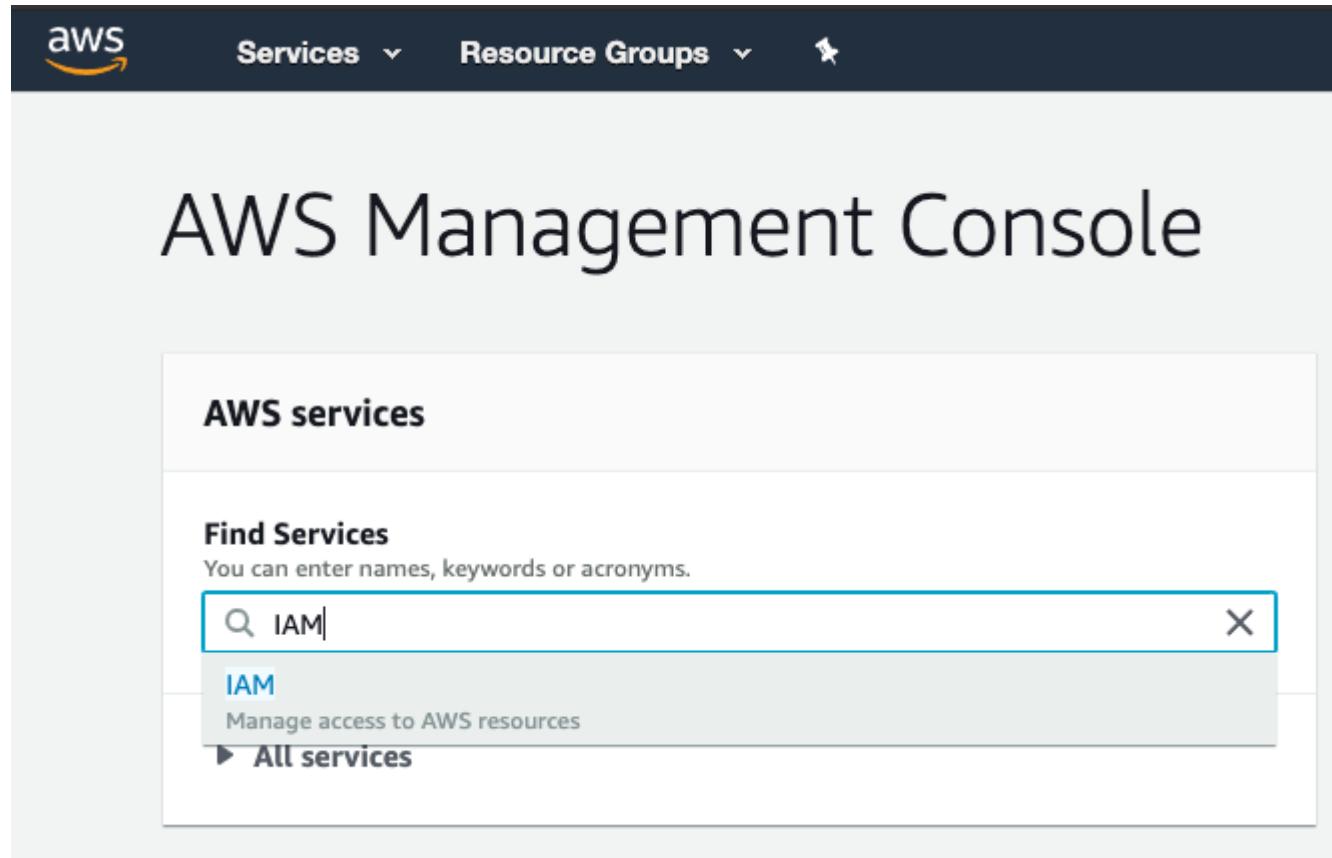
Provide this user with:

- 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

The screenshot shows the 'Add user' wizard in the AWS IAM console. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and a notification bell icon. Below the header, the title 'Add user' is displayed above a progress bar with three steps: 1 (blue circle), 2 (white circle), 3 (white circle), and a right-pointing arrow.

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* [Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* Autogenerated password Custom password

Require password reset User must create a new password at next sign-in
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

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

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

Add user

1 2 3 4 5

Set permissions

The screenshot shows the AWS IAM 'Add user' wizard at step 2. It has five numbered steps at the top right: 1, 2 (highlighted in blue), 3, 4, and 5. Below the steps, there's a section titled 'Set permissions' with three options: 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly'. The third option is highlighted with a blue border. Below this is a 'Create policy' button and a refresh icon. At the top right, there's a search bar with the text 'IAMFullAccess'. Below the search bar is a table with one result, showing a policy named 'IAMFullAccess' which is AWS managed and has no usage. The table has columns for 'Policy name', 'Type', and 'Used as'.

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	IAMFullAccess	AWS managed	None

9. Search for S3FullAccess and select it

Add user

1 2 3 4 5

▼ Set permissions

 Add user to group  Copy permissions from existing user  Attach existing policies directly

[Create policy](#) 

Filter policies  **S3FullAccess** Showing 1 result

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	▶  AmazonS3FullAccess	AWS managed	None

10. Search for EC2FullAccess and select it

Add user

1 2 3 4 5

▼ Set permissions

 Add user to group  Copy permissions from existing user  Attach existing policies directly

[Create policy](#) 

Filter policies		Showing 1 result	
	Policy name	Type	Used as
<input checked="" type="checkbox"/> ▶  AmazonEC2FullAccess	AmazonEC2FullAccess	AWS managed	None

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

Add user



Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	manager
AWS access type	AWS Management Console access - with a password
Console password type	Autogenerated
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonS3FullAccess
Managed policy	AmazonEC2FullAccess
Managed policy	IAMFullAccess

Tags

No tags were added.

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

The screenshot shows the AWS IAM 'Add user' interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and a user 'admin'. Below the header, the title 'Add user' is displayed, along with four numbered steps (1, 2, 3, 4) in circles. A green 'Success' message box contains text about successfully creating users and instructions for signing in. It also includes a link to the AWS Management Console sign-in page. Below the message is a 'Download .csv' button. The main content area shows a table with one row for a user named 'manager'. The table has columns for 'User' and 'Email login instructions'. The 'User' column shows 'manager' with a green checkmark and a dropdown arrow. The 'Email login instructions' column shows a 'Send email' button. At the bottom, a summary box lists four successful actions: 'Created user manager', 'Attached policy IAMFullAccess to user manager', 'Attached policy AmazonS3FullAccess to user manager', and 'Created login profile for user manager'.

User	Email login instructions
manager	Send email

Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.
Users with AWS Management Console access can sign-in at: <https://redacted.signin.aws.amazon.com/console>

[Download .csv](#)

Created user manager
Attached policy IAMFullAccess to user manager
Attached policy AmazonS3FullAccess to user manager
Created login profile for user manager

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

Users > manager

Summary

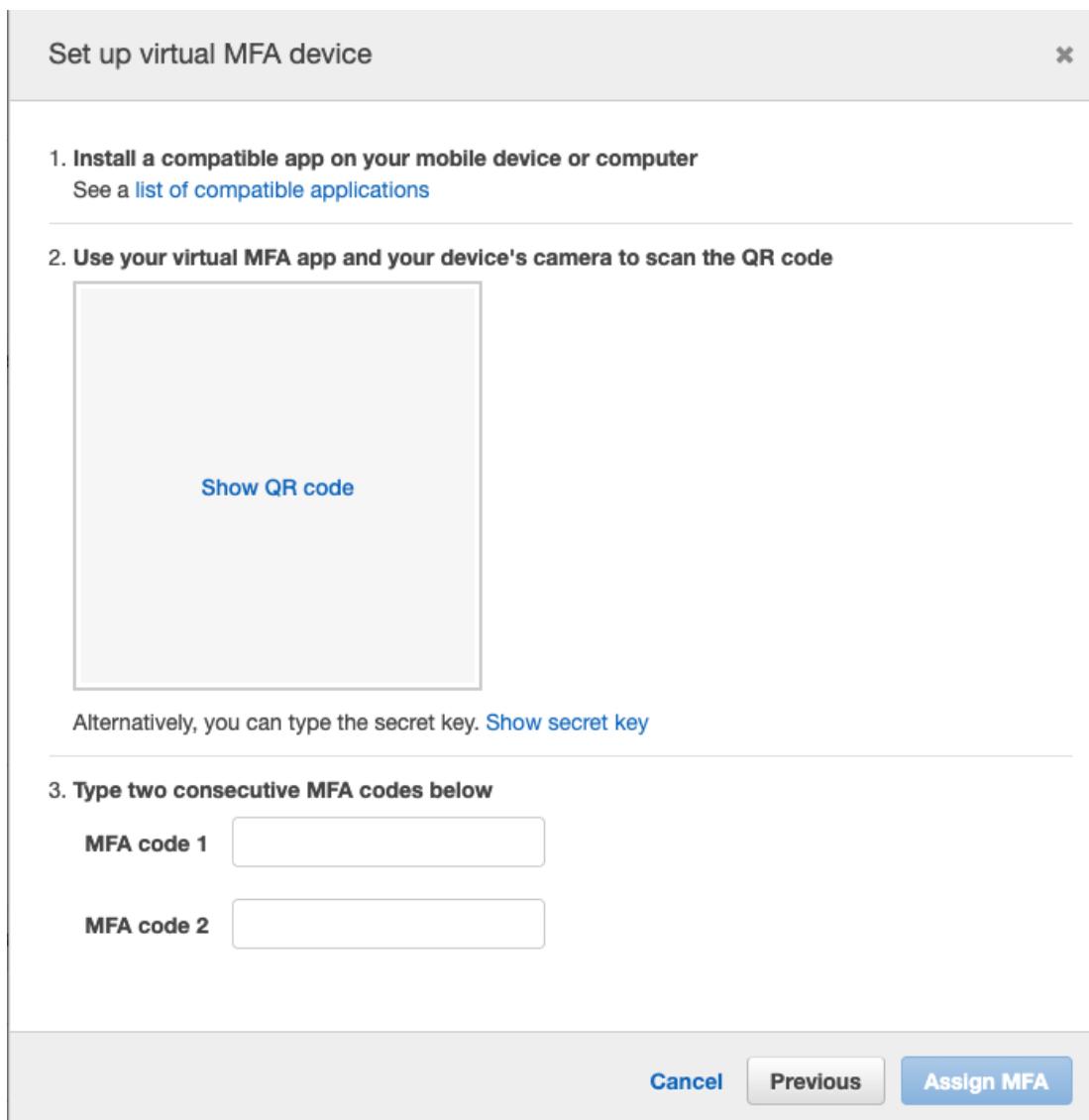
User ARN	arn:aws:iam:: redacted :user/manager	
Path	/	
Creation time	2020-04-06 14:18 PDT	

Permissions **Groups** **Tags** **Security credentials** **Access Advisor**

Sign-in credentials

Summary	<ul style="list-style-type: none">Console sign-in link: https:// redacted .signin.aws.amazon.com/console
Console password	Enabled (never signed in) Manage
Assigned MFA device	Not assigned Manage
Signing certificates	None

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

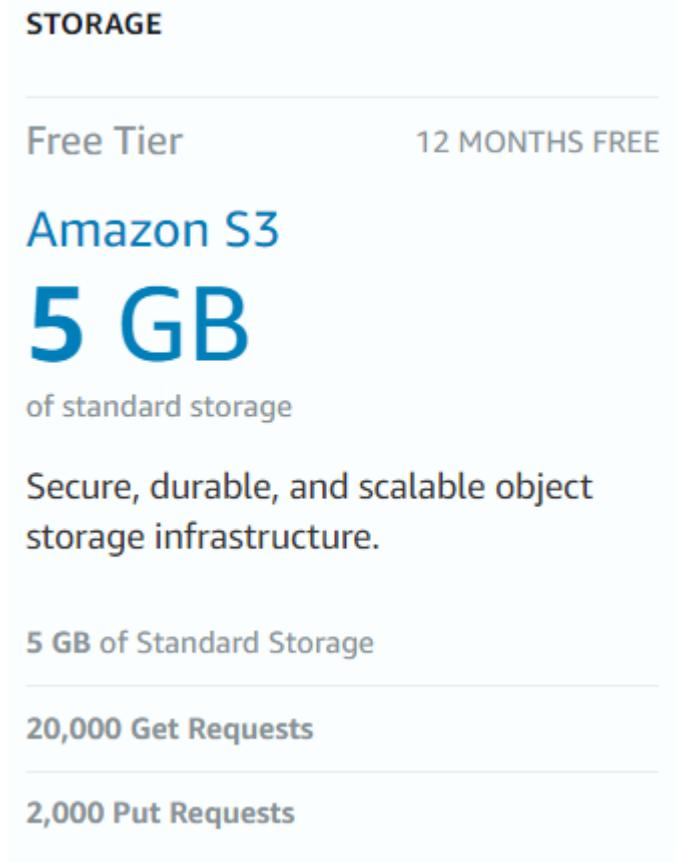
If you have not done so already, register with AWS and authenticate as the **manager** user

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

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

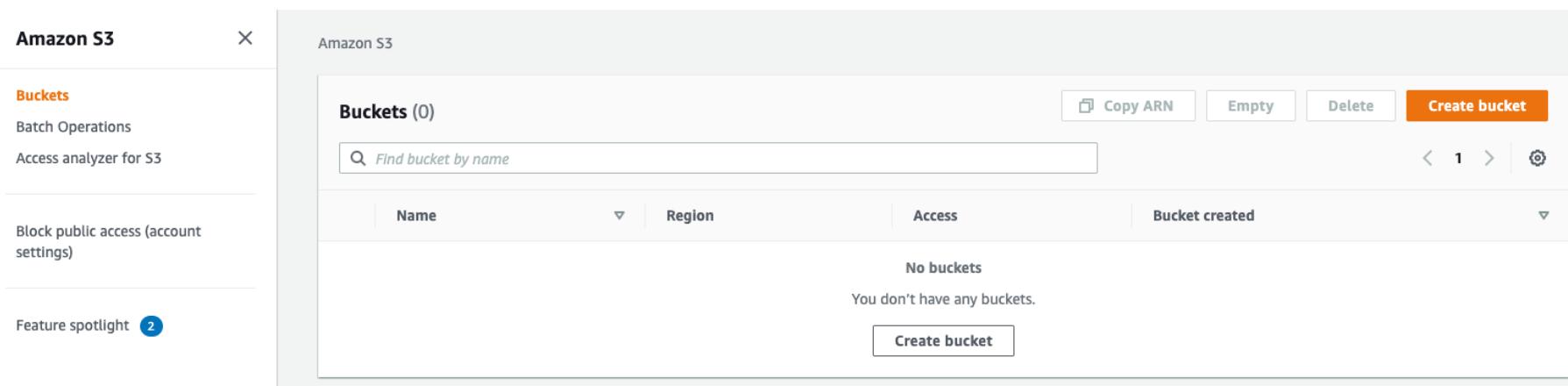
Upload content to both folders (a single file in each folder will suffice)

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"



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

Upload Create folder Download Actions

US West (Oregon)

This bucket is empty. Upload new objects to get started.

 Upload an object

Buckets are globally unique containers for everything that you store in Amazon S3.

[Learn more](#)

 Set object properties

After you create a bucket, you can upload your objects (for example, your photo or video files).

[Learn more](#)

 Set object permissions

By default, the permissions on an object are private, but you can set up access control policies to grant permissions to others.

[Learn more](#)

Get started

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

Amazon S3 > ami-test-2

ami-test-2

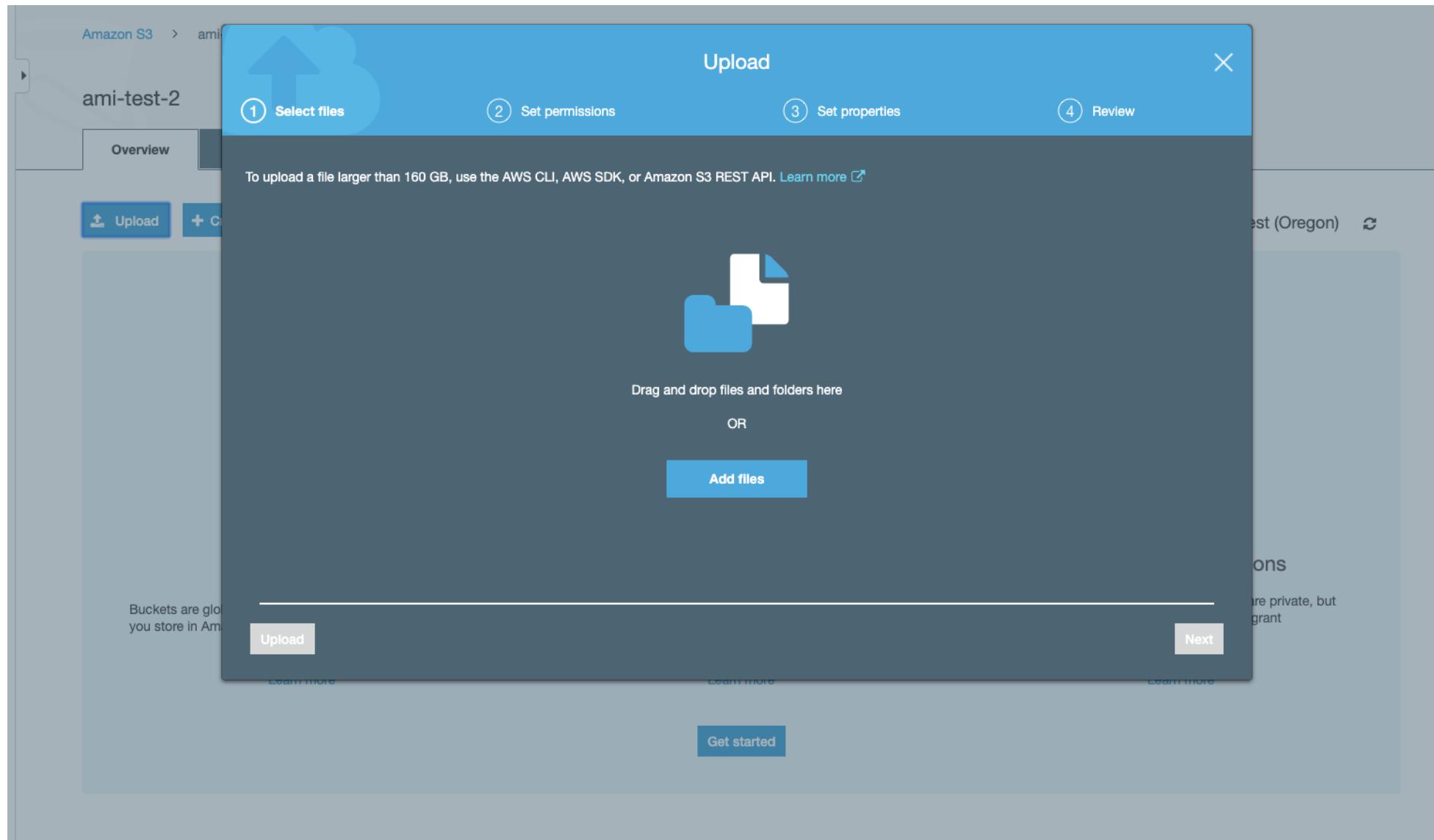
Overview Properties Permissions Management Access points

Type a prefix and press Enter to search. Press ESC to clear.

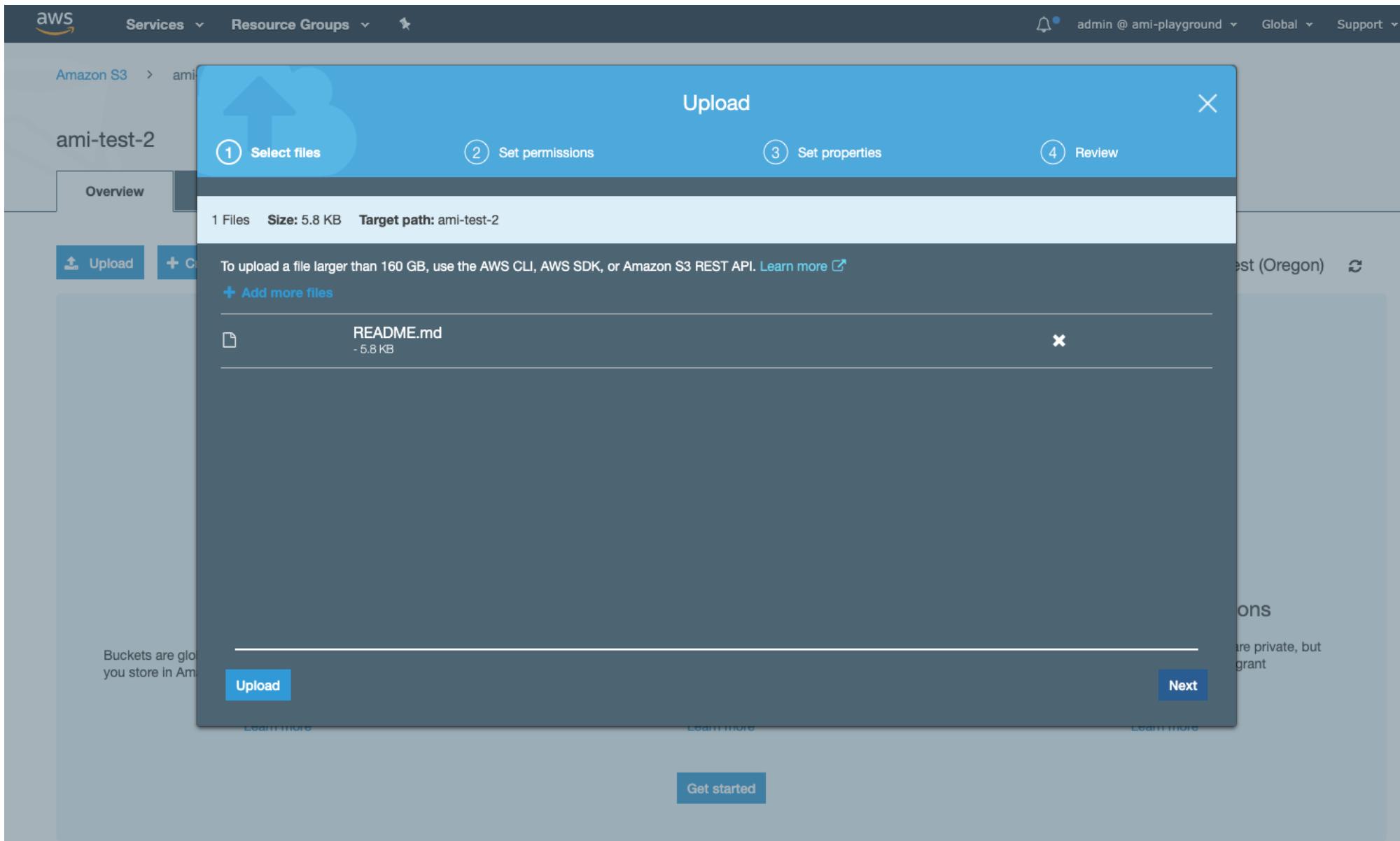
Upload Create folder Download Actions

<input type="checkbox"/>	Name ▾	Last r
<input type="checkbox"/>	exercise	--
<input type="checkbox"/>	manager	--

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



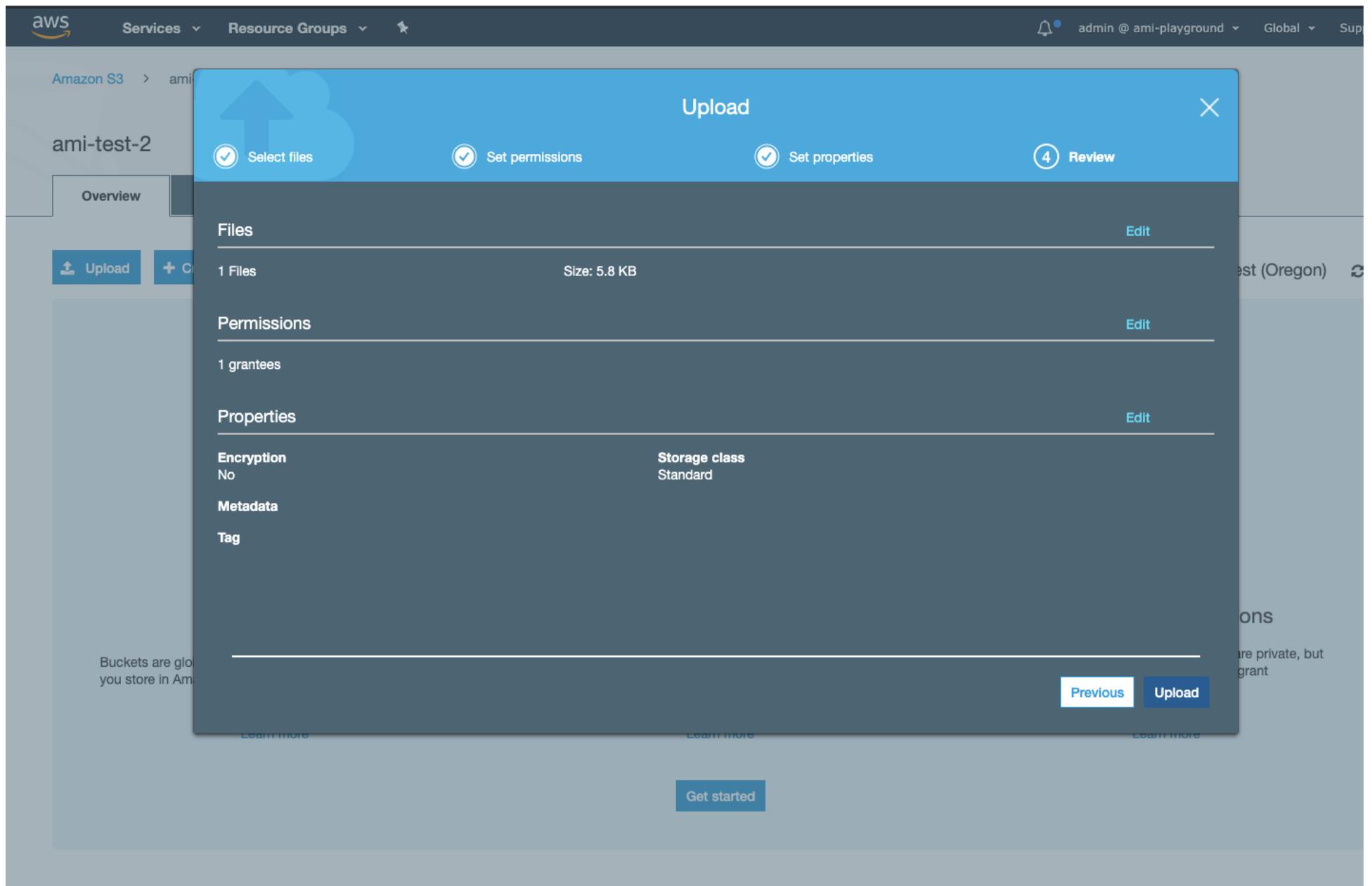
- Click the **Next** button to see the list of default permissions for this file (we will use IAM permission to access it rather than resource permissions)
- Click Next to see the S3 storage classes, and leave the default **Standard** class selected

The screenshot shows the Amazon S3 'Upload' wizard in progress. The current step is '3 Set properties'. The 'Storage class' section is displayed, listing various options with their characteristics and fees:

Storage class	Designed for	Availability Zones	Min storage duration	Min billable object size	Monitoring and automation fees	Retrieval fees
<input checked="" type="radio"/> Standard	Frequently accessed data	≥ 3	-	-	-	-
<input type="radio"/> Intelligent-Tiering	Long-lived data with changing or unknown access patterns	≥ 3	30 days	-	Per-object fees apply	-
<input type="radio"/> Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days	128KB	-	Per-GB fees apply
<input type="radio"/> One Zone-IA	Long-lived, infrequently accessed, non-critical data	≥ 1	30 days	128KB	-	Per-GB fees apply
<input type="radio"/> Glacier	Archive data with retrieval times ranging from minutes to hours	≥ 3	90 days	40KB	-	Per-GB fees apply
<input type="radio"/> Glacier Deep Archive	Archive data that rarely, if ever, needs to be accessed with retrieval times in	≥ 3	180 days	40KB	-	Per-GB fees apply

At the bottom of the wizard, there are 'Upload', 'Previous', and 'Next' buttons, along with a 'Get started' button.

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

Home

Cost Management

Cost Explorer

Budgets

Budgets Reports

Savings Plans

Cost & Usage Reports

Cost Categories

Cost allocation tags

Billing

Bills

Orders and invoices

Credits

Preferences

Billing preferences

Payment methods

Consolidated billing

Tax settings

Billing & Cost Management Dashboard



Getting Started with AWS Billing & Cost Management

- Manage your costs and usage using [AWS Budgets](#)
- Visualize your cost drivers and usage trends via [Cost Explorer](#)
- Dive deeper into your costs using the [Cost and Usage Reports with Athena integration](#)
- **Learn more:** Check out the [AWS What's New webpage](#)

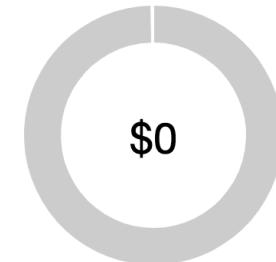
Do you have Reserved Instances (RIs)?

- Access the RI Utilization & Coverage reports—and RI purchase recommendations—via [Cost Explorer](#).

Month-to-Date Spend by Service

[Bill Details](#)

The chart below shows the proportion of costs spent for each service you use.



Spend Summary

[Cost Explorer](#)

Welcome to the AWS Billing & Cost Management console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for May 2020

\$0.00

\$4 _____

\$3 _____

\$2 _____

No Amount Due

\$0.00

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.

Cost & Usage Reports

Cost Categories

Cost allocation tags

Billing

Bills

Orders and invoices

Credits

Preferences

Billing preferences

Payment methods

Consolidated billing

Tax settings

Security Check 

 Refresh Image



Please type the characters as shown above

By clicking "Redeem" you indicate that you have read and agree to the terms of the AWS Promotional Credit Terms & Conditions located [here](#).

Redeem

The table below displays all AWS credits redeemed by your account. Credits are automatically applied to charges associated with qualifying AWS service usage. Please note that the values for used and remaining credit amounts are updated each month when your invoice is finalized.

Expiration Date	Credit Name	Amount Used	Amount Remaining	Applicable Products
12/31/2020	EDU_ENG_FY2019_IC_Q4_12_UDACITY_75USD	\$0.00	\$75.00	See complete list

Total Credit Amount Remaining (as of 05/01/2020): \$75.00

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

COMPUTE

Free Tier

12 MONTHS FREE

Amazon EC2

750 Hours

per month

Resizable compute capacity in the Cloud.

750 hours per month of Linux, RHEL, or SLES t2.micro or t3.micro instance dependent on region

750 hours per month of Windows t2.micro or t3.micro instance dependent on region

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**

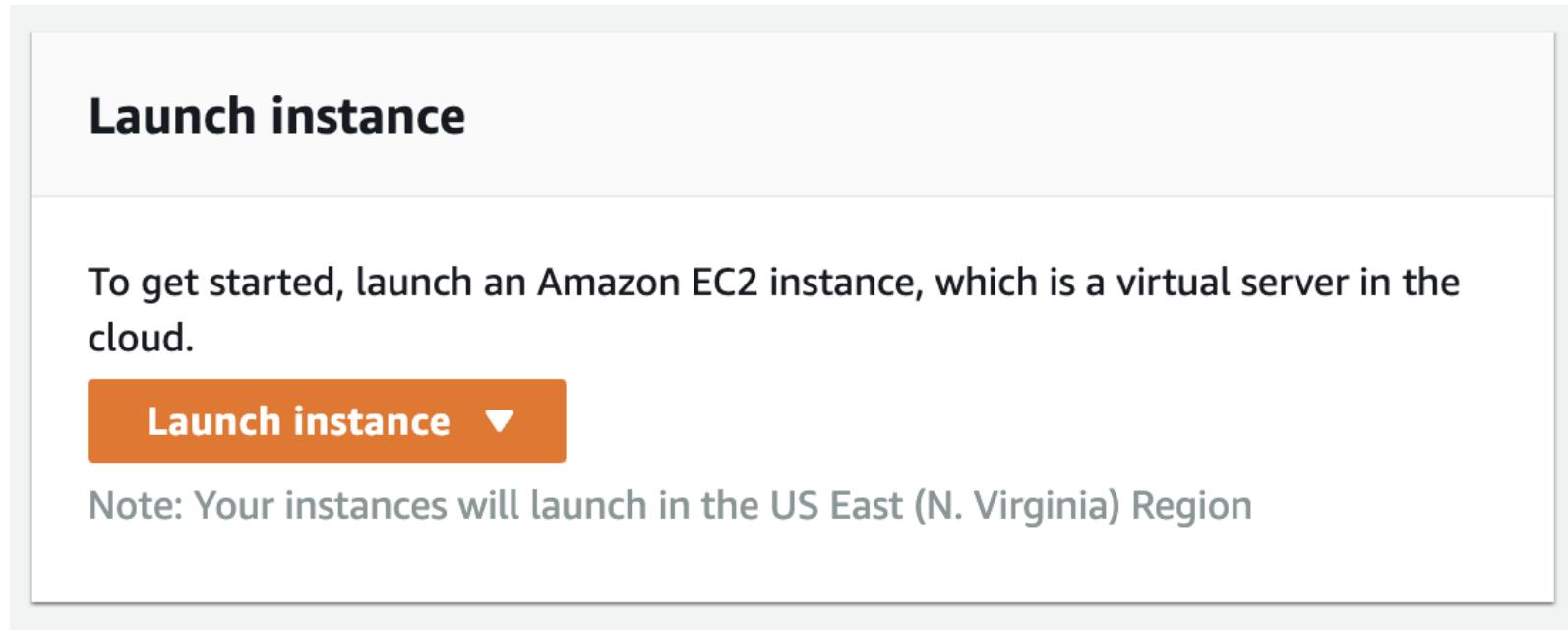
Use the **us-east-1 N.Virginia** region to launch an EC2 instance with the following configuration:

AMI: Amazon Linux

Instance type: t2.micro

IP: Public

Storage size: 20GB



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

The screenshot shows the AWS Cloud9 instance creation interface. At the top, there's a navigation bar with icons for back, forward, and search. Below it, a search bar contains the query "Amazon Linux". A dropdown menu lists "Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0fc61db8544a617ed (64-bit x86) / ami-0f90a34c9df977efb (64-bit Arm)". To the right of the search results, there's a "Select" button and two radio buttons for "64-bit (x86)" and "64-bit (Arm)", with "64-bit (x86)" being selected. The main content area shows details for the chosen AMI: "Amazon Linux" and "Free tier eligible". It also lists "Root device type: ebs", "Virtualization type: hvm", and "ENAv Enabled: Yes".

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	<input checked="" type="checkbox"/>
Add New Volume							

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

[1. Choose AMI](#) [2. Choose Instance Type](#) [3. Configure Instance](#) [4. Add Storage](#) [5. Add Tags](#) [6. Configure Security Group](#) [7. Review](#)

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

[Edit AMI](#)**Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0fc61db8544a617ed**

Free tier eligible Root Device Type: ebs Virtualization type: hvm

Instance Type

[Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

[Edit security groups](#)

Security group name: launch-wizard-1
Description: launch-wizard-1 created 2020-04-07T13:41:59.374-07:00

This security group has no rules

Instance Details

[Edit instance details](#)

Storage

[Edit storage](#)

Tags

[Edit tags](#)[Cancel](#) [Previous](#) [Launch](#)

Choose to proceed without an SSH key pair.

Select an existing key pair or create a new key pair X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

No key pairs found

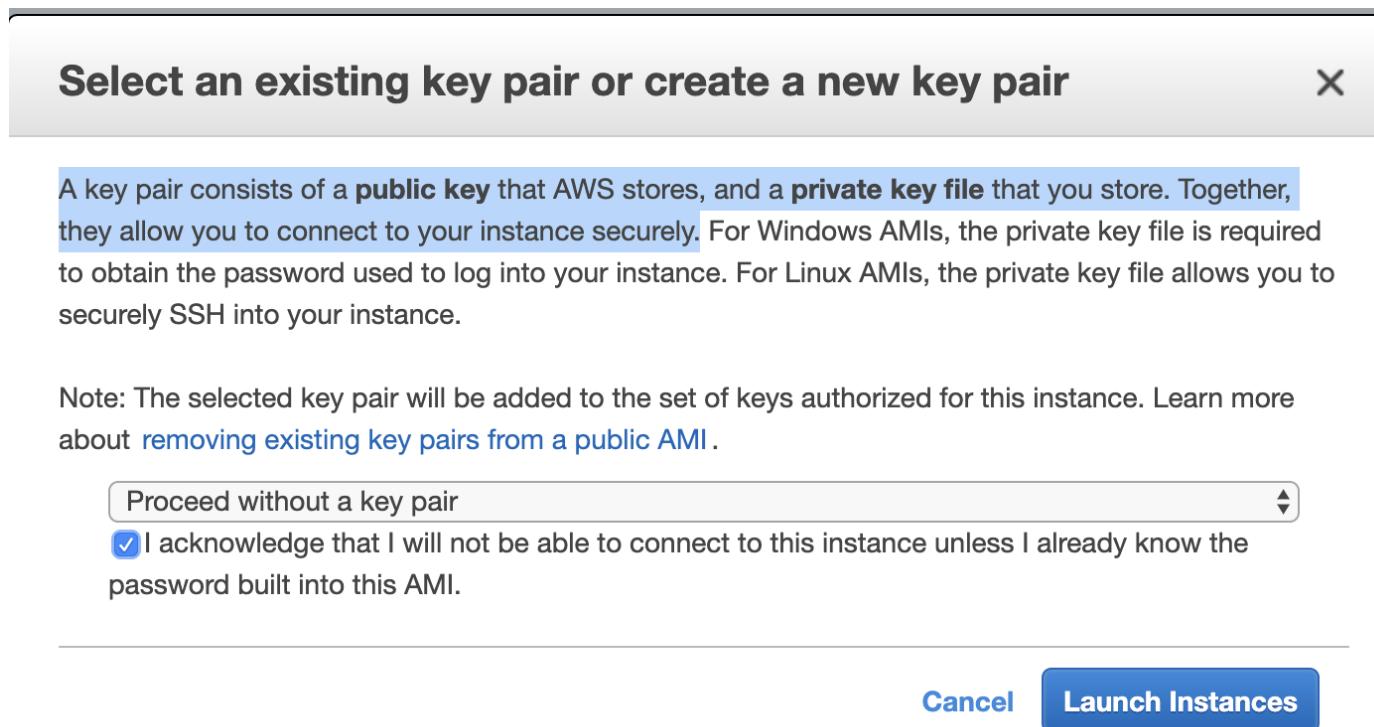
⚠ **No key pairs found**

You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

[Cancel](#)

[Launch Instances](#)

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.

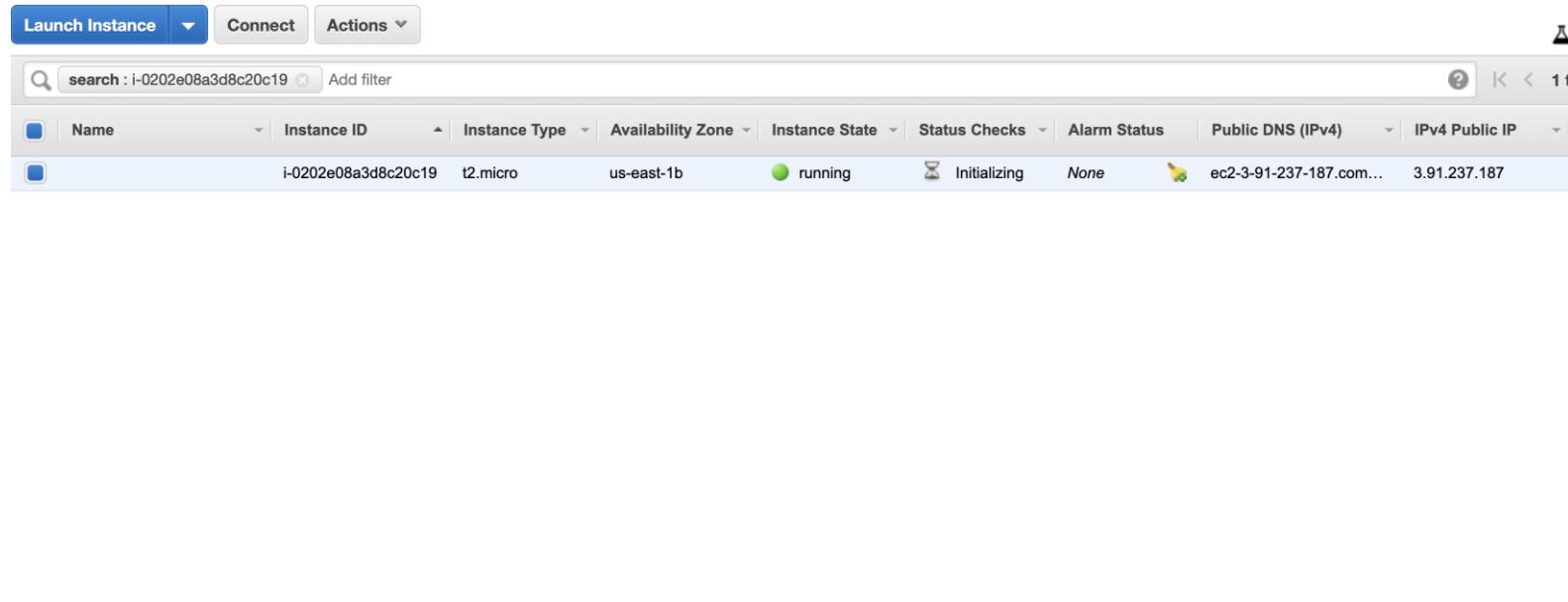


Select **Proceed without a key pair** and check the box for **I acknowledge that...** Then click on the **Launch Instances** button.

Launch Status

- ✓ Your instances are now launching
The following instance launches have been initiated: [i-0202e08a3d8c20c19](#) [View launch log](#)
- i Get notified of estimated charges
[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

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



The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for "Launch Instance", "Connect", and "Actions". A search bar contains the text "search : i-0202e08a3d8c20c19" and an "Add filter" button. Below the search bar is a table header with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Public IP. A single instance row is visible, showing the details: Name (i-0202e08a3d8c20c19), Instance ID (i-0202e08a3d8c20c19), Instance Type (t2.micro), Availability Zone (us-east-1b), Instance State (running), Status Checks (Initializing), Alarm Status (None), Public DNS (IPv4) (ec2-3-91-237-187.compute-1.amazonaws.com), and IPv4 Public IP (3.91.237.187).

Instance: **i-0202e08a3d8c20c19** Public DNS: **ec2-3-91-237-187.compute-1.amazonaws.com**

Description Status Checks Monitoring Tags

Instance ID	i-0202e08a3d8c20c19	Public DNS (IPv4)	ec2-3-91-237-187.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.91.237.187
Instance type	t2.micro	IPv6 IPs	-

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

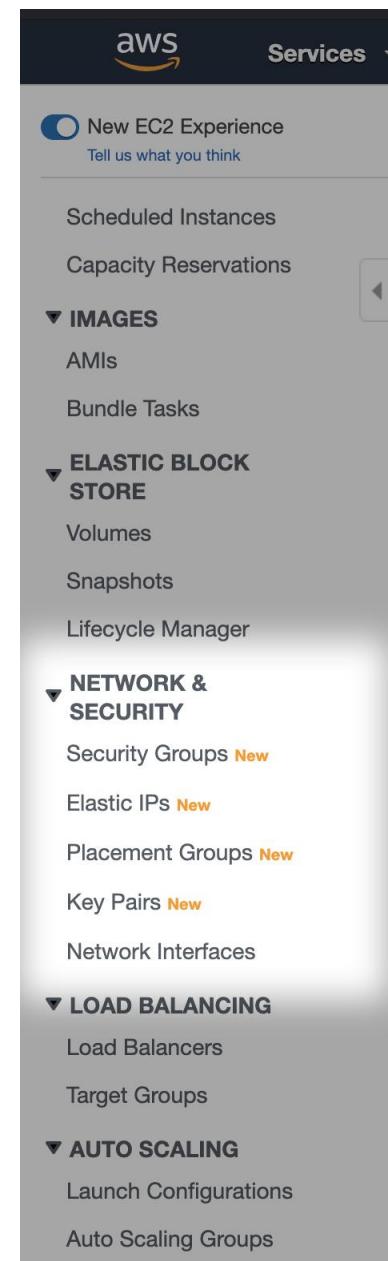
Part 1: Create a Security Group

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.

The screenshot shows the 'Create security group' wizard in the AWS EC2 console. The top navigation bar includes links for AWS Services, Resource Groups, CloudFront, EC2 (selected), Elastic Container Service, and Elastic Container Registry. The breadcrumb trail indicates the user is at EC2 > Security Groups > Create security group. The main form has a 'Basic details' section with fields for 'Security group name' (set to 'MySecurityGroup'), 'Description' (set to 'Allow SSH and HTTP access'), and 'VPC' (set to 'vpc-036ef207f1f23c608'). A note below the name field states 'Name cannot be edited after creation.'



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

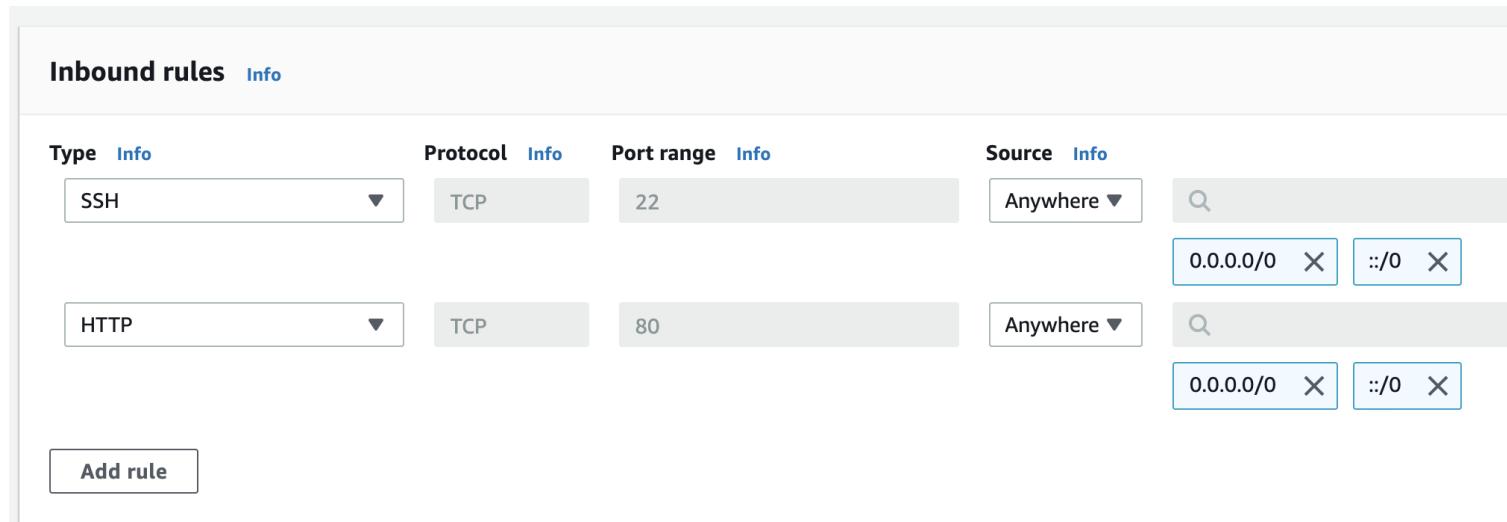
SSH TCP 22 Anywhere

HTTP TCP 80 Anywhere

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info
SSH	TCP	22	Anywhere ▾ 0.0.0.0/0 X ::/0 X
HTTP	TCP	80	Anywhere ▾ 0.0.0.0/0 X ::/0 X

[Add rule](#)

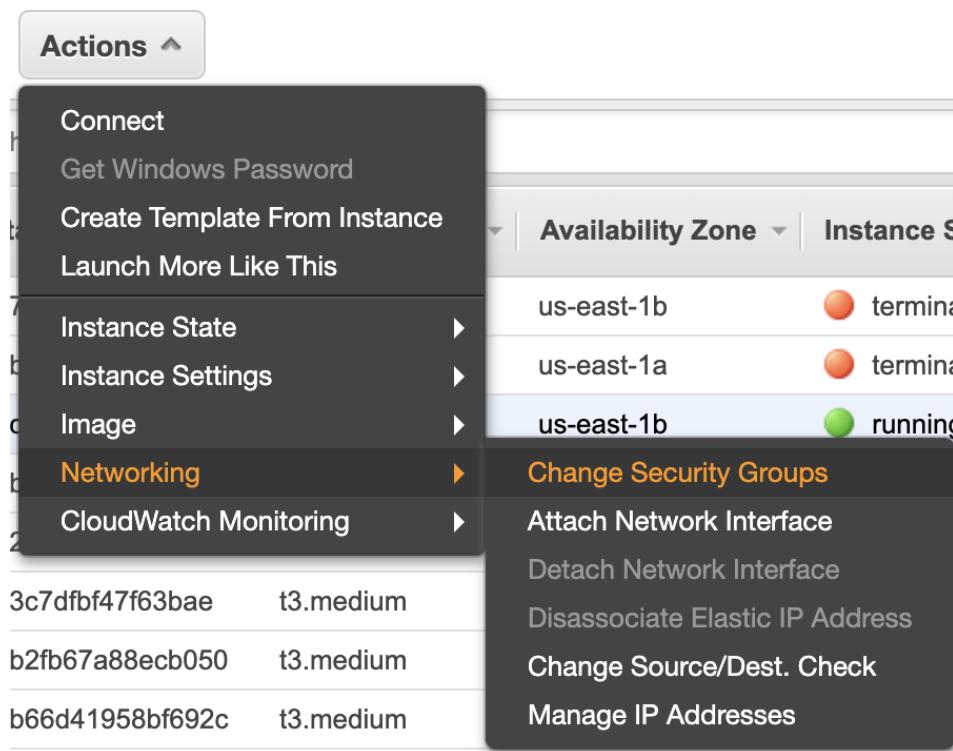


Click on the **Create security group** button to create the security group.

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.

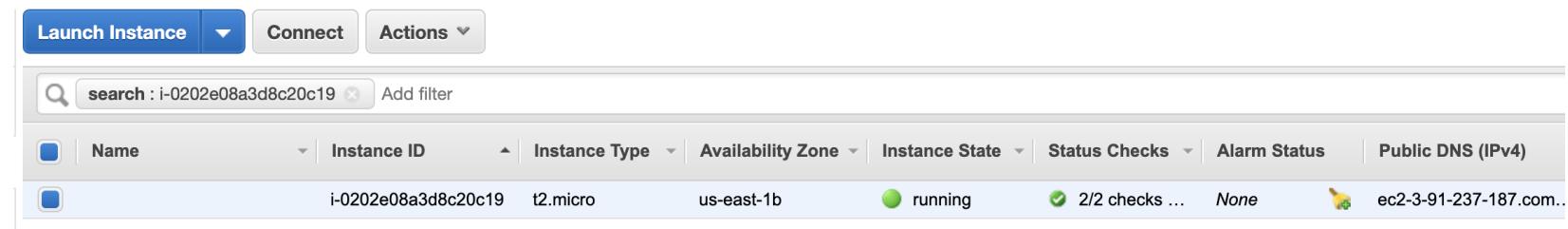
In the **Description** tab at the bottom of the screen, we can now see the security group.

Private IPs	172.31.39.198	Security groups	MySecurityGroup . view inbound rules . view outbound rules
Secondary private IPs			
VPC ID	vpc-036ef207f1f23c608		_64-gp2 (ami-
Subnet ID	subnet-0281cefdaa6d0b3b7		
Network interfaces	eth0		
IAM role	-		
Key pair name	ami	T2/T3 Unlimited	Disabled

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.

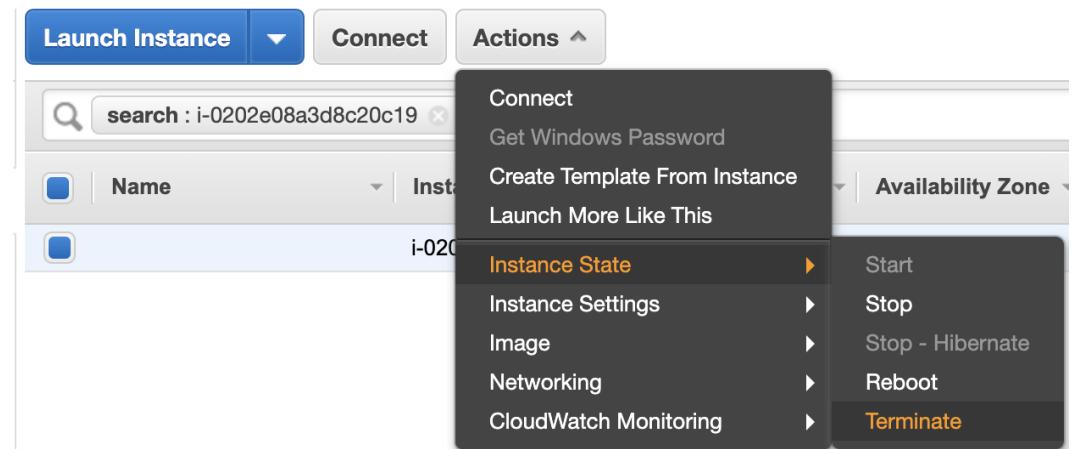
Select the instance ID from the AWS Instances console.



The screenshot shows the AWS Instances console. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. A dropdown menu is open under 'Actions' with the option 'Instance State' highlighted. Below the buttons is a search bar with the text 'search : i-0202e08a3d8c20c19' and a 'Add filter' button. The main table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IPv4). One instance is listed: 'i-0202e08a3d8c20c19' (t2.micro), 'us-east-1b', 'running', '2/2 checks ...', 'None', and 'ec2-3-91-237-187.com..'. The instance state is also shown as green with a circle icon.

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.



The screenshot shows the AWS Instances console with the 'Actions' dropdown menu open. The 'Instance State' option is highlighted. The menu includes options: Connect, Get Windows Password, Create Template From Instance, Launch More Like This, Start, Stop, Stop - Hibernate, Reboot, and Terminate. The 'Terminate' option is at the bottom of the list.

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.

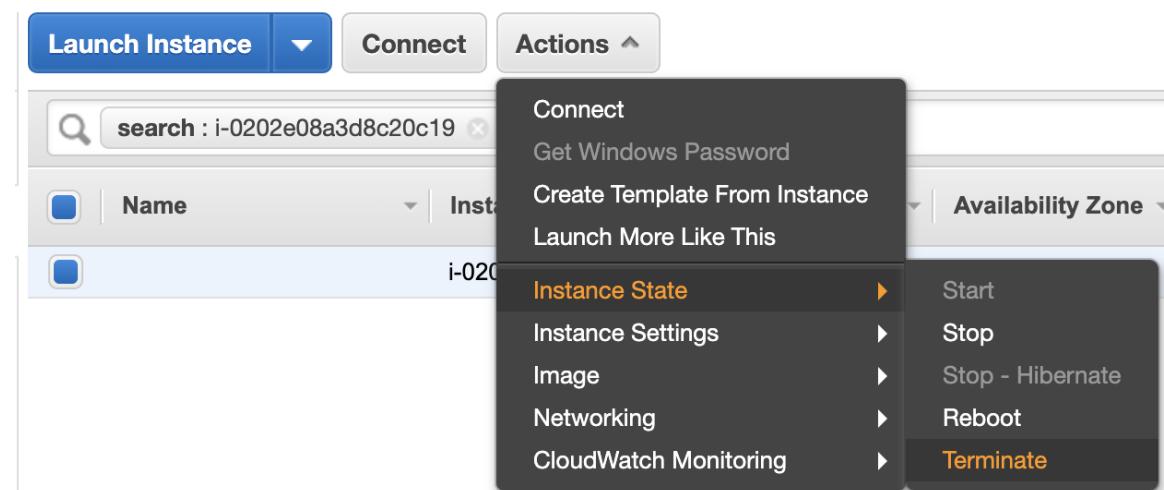
This is also known as an "Init script". For this exercise solution, we will program a shell script to be our init script.

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

▼ Advanced Details

Metadata accessible ⓘ Enabled

Metadata version ⓘ V1 and V2 (token optional)

Metadata token response hop limit ⓘ 1

User data ⓘ As text As file Input is already base64 encoded

(Optional)

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

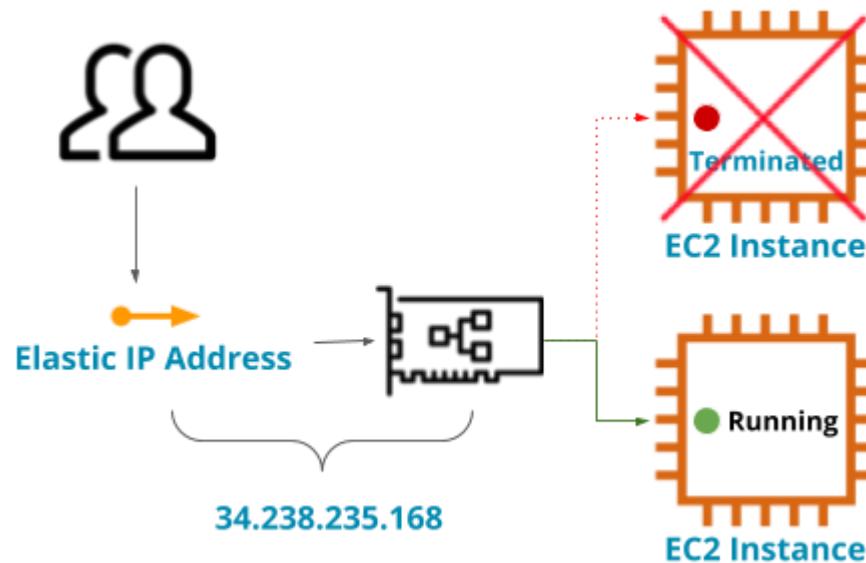
9. Find the instance's **IPv4 Public IP** address from the EC2 Instance Dashboard

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

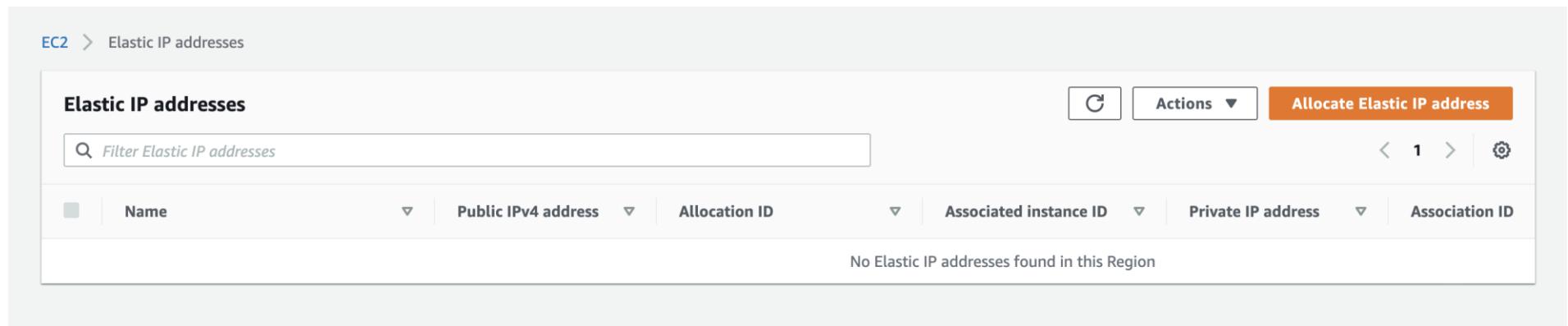
- If you have not done so already, launch an EC2 instance
- Allocate an **Elastic IP**
- Associate the Elastic IP with your EC2 instance
- Verify that the Elastic IP address works

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** .



The screenshot shows the AWS EC2 console interface. The top navigation bar has 'EC2' selected, followed by a right arrow, and then 'Elastic IP addresses'. Below the navigation is a search bar labeled 'Filter Elastic IP addresses' with a magnifying glass icon. To the right of the search bar are three buttons: a white button with a black 'C' icon, a white button with a black 'Actions' dropdown arrow, and an orange button labeled 'Allocate Elastic IP address'. Further to the right are navigation icons for back, forward, and refresh, along with a gear icon for settings. The main content area has a header 'Elastic IP addresses' with a table of columns: Name, Public IPv4 address, Allocation ID, Associated instance ID, Private IP address, and Association ID. A message at the bottom of the table says 'No Elastic IP addresses found in this Region'.

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

The screenshot shows the 'Allocate Elastic IP address' configuration page. At the top, the breadcrumb navigation indicates 'EC2 > Elastic IP addresses > Allocate Elastic IP address'. The main title 'Allocate Elastic IP address' is displayed prominently. Below it, a descriptive text states: 'Allocate an Elastic IP address by selecting the public IPv4 address pool from which the public IP address is to be allocated. Elastic IP addresses incur charges if they are not associated with a running instance or a network interface that is attached to a running instance.' A 'Learn more' link is provided for additional information. A section titled 'Elastic IP address settings' contains a 'Public IPv4 address pool' configuration. It lists three options: 'Amazon's pool of IPv4 addresses' (selected), 'Public IPv4 address that you bring to your AWS account (option disabled because no pools found)', and 'Customer owned pool of IPv4 addresses (option disabled because no customer owned pools found)'. Each option has a 'Learn more' link. At the bottom right, there are 'Cancel' and 'Allocate' buttons, with 'Allocate' being highlighted in orange.

EC2 > Elastic IP addresses > Allocate Elastic IP address

Allocate Elastic IP address

Allocate an Elastic IP address by selecting the public IPv4 address pool from which the public IP address is to be allocated. Elastic IP addresses incur charges if they are not associated with a running instance or a network interface that is attached to a running instance. [Learn more](#)

Elastic IP address settings

Public IPv4 address pool

Public IP addresses are allocated from Amazon's pool of public IP addresses, from a pool that you own and bring to your account, or from a pool that you own and continue to advertise..

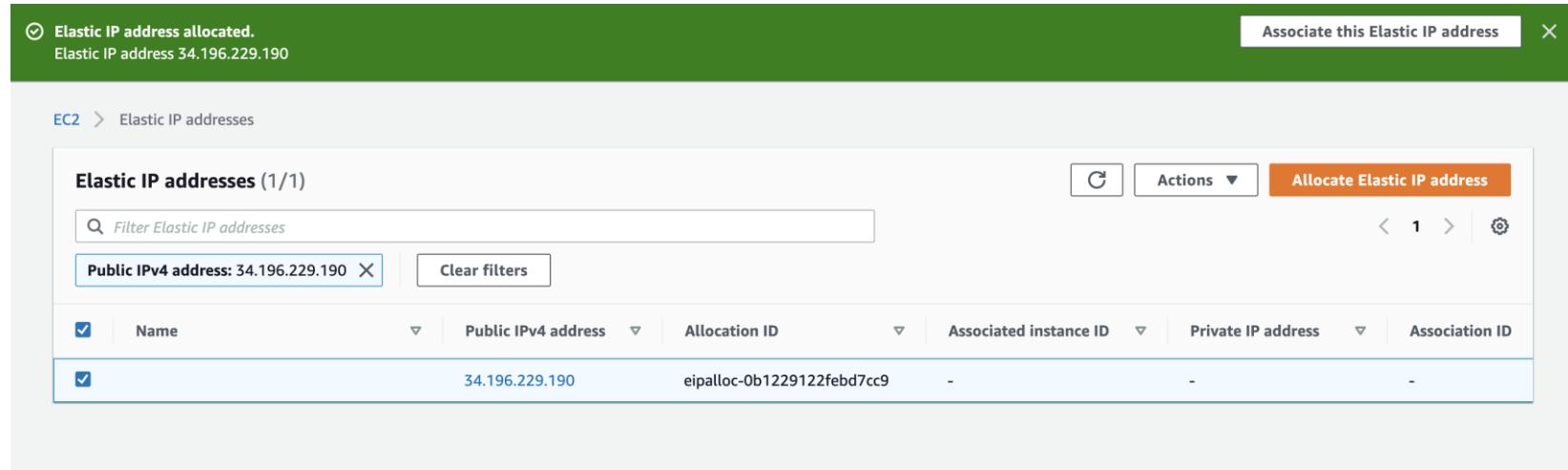
Amazon's pool of IPv4 addresses

Public IPv4 address that you bring to your AWS account(option disabled because no pools found) [Learn more](#)

Customer owned pool of IPv4 addresses(option disabled because no customer owned pools found) [Learn more](#)

Cancel **Allocate**

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.

EC2 > [Elastic IP addresses](#) > Associate Elastic IP address

Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (34.196.229.190)

Elastic IP address: 34.196.229.190

Resource type
Choose the type of resource with which to associate the Elastic IP address.

Instance
 Network interface

⚠️ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more.](#)

Instance
 X C

Private IP address
The private IP address with which to associate the Elastic IP address.

Reassociation
Specify whether the Elastic IP address can be reassigned to a different resource if it's already associated with a resource.
 Allow this Elastic IP address to be reassigned

Cancel Associate

ⓘ Elastic IP address associated.
Elastic IP address 34.196.229.190

EC2 > Elastic IP addresses

Elastic IP addresses (1/1)

Filter Elastic IP addresses

Actions ▾ Allocate Elastic IP address

Public IPv4 address: 34.196.229.190 X Clear filters

<input checked="" type="checkbox"/>	Name	Public IPv4 address	Allocation ID	Associated instance ID	Private IP address	Association ID
<input checked="" type="checkbox"/>		34.196.229.190	eipalloc-0b1229122febd7cc9	i-01c8c94257b83b5cc	172.31.39.198	eipassoc-0cc6a94

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.

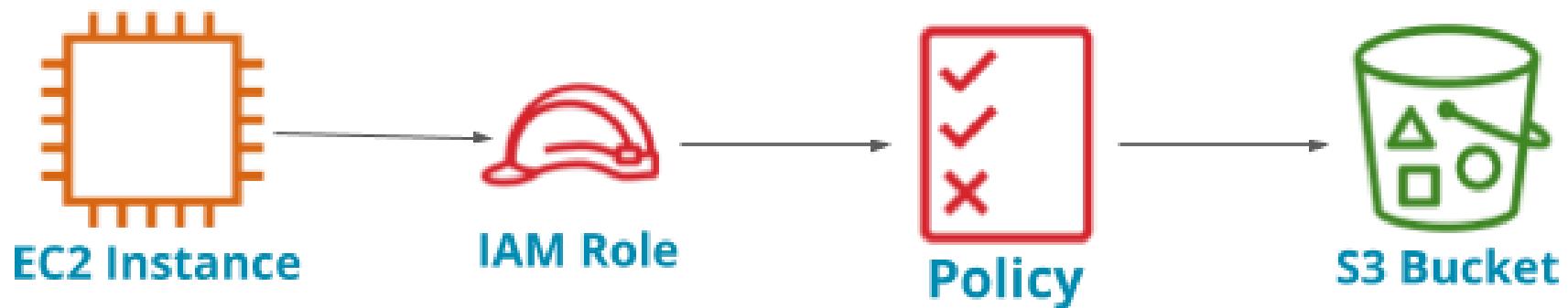
Instance: i-01c8c94257b83b5cc Elastic IP: 34.196.229.190

Description Status Checks Monitoring Tags

Instance ID	i-01c8c94257b83b5cc	Public DNS (IPv4)	ec2-34-196-229-190.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	34.196.229.190
Instance type	t2.micro	IPv6 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

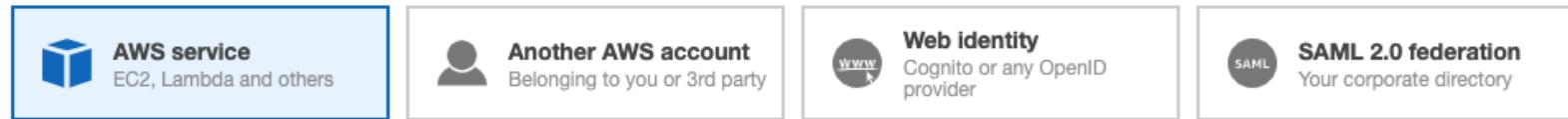
Create the IAM Role

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

Create role

1 2 3 4

Select type of trusted entity



Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

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. 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

Create role

Review

Provide the required information below and review this role before you create it.

Role name* Use alphanumeric and '+=, @-' characters. Maximum 64 characters.

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

Trusted entities AWS service: ec2.amazonaws.com

Policies Policies not attached

Permissions boundary Permissions boundary is not set

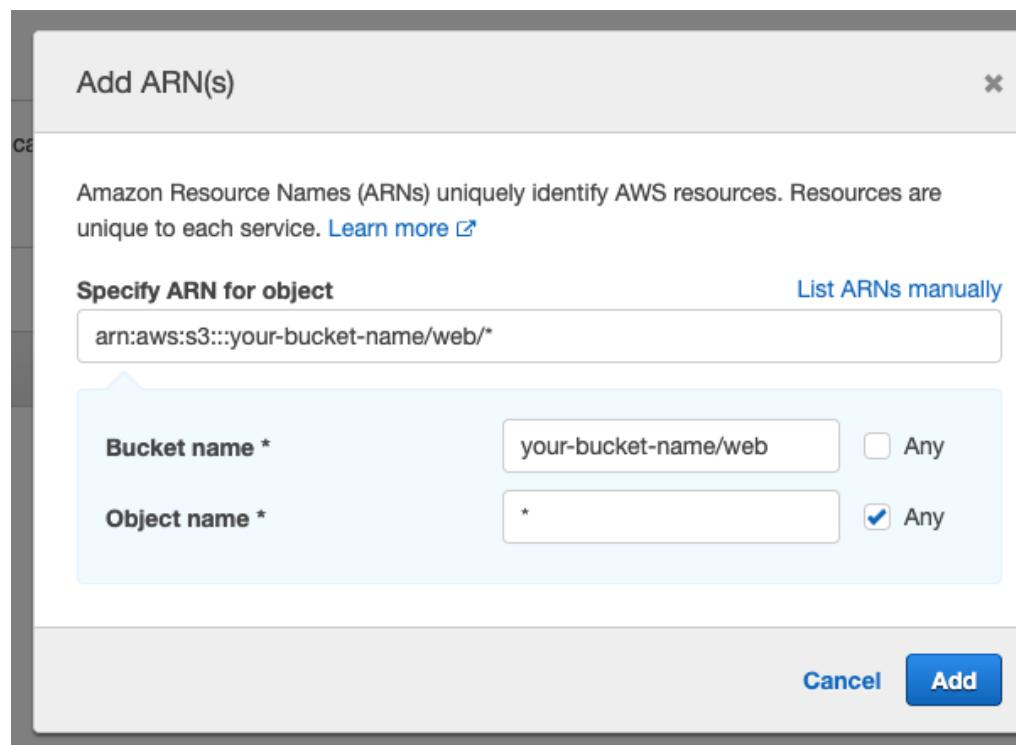
No tags were added.

 The role **web** has been created.

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/*"



Create policy

1

2

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](#)

S3 (2 actions)		Clone Remove
▶ Service S3		
▶ Actions List		
ListBucket		
Read		
GetObject		
▼ Resources <input checked="" type="radio"/> Specific <input type="radio"/> All resources Close		
bucket ?	arn:aws:s3:::your-bucket-name	EDIT <input type="checkbox"/> Any
Add ARN to restrict access		
object ?	arn:aws:s3:::your-bucket-name/web/*	EDIT <input type="checkbox"/> Any
Add ARN to restrict access		
▶ Request conditions Specify request conditions (optional)		

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

[ec2_web_s3_access has been created.](#)

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"

	Role name	Trusted entities	Last activity
<input type="checkbox"/>	web	AWS service: ec2	None

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

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

▼ Permissions policies

Get started with permissions

This role doesn't have any permissions yet. Get started by attaching one or more policies to this role. [Learn more](#)

Attach policies **Add inline policy**

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

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

▼ Permissions policies (1 policy applied)

Attach policies **Add inline policy**

Policy name	Policy type
ec2_web_s3_access	Managed policy

Permissions boundary (not set)

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

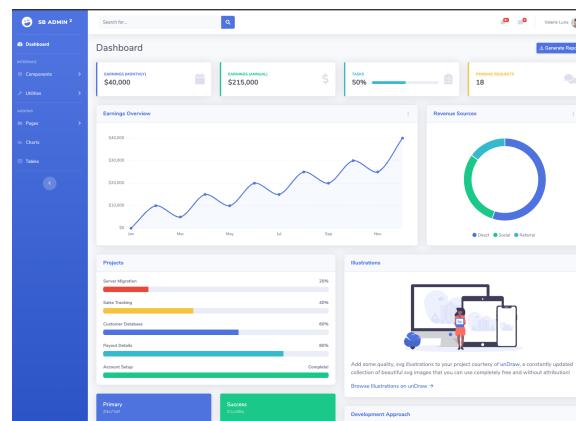
Init Script Explanation

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

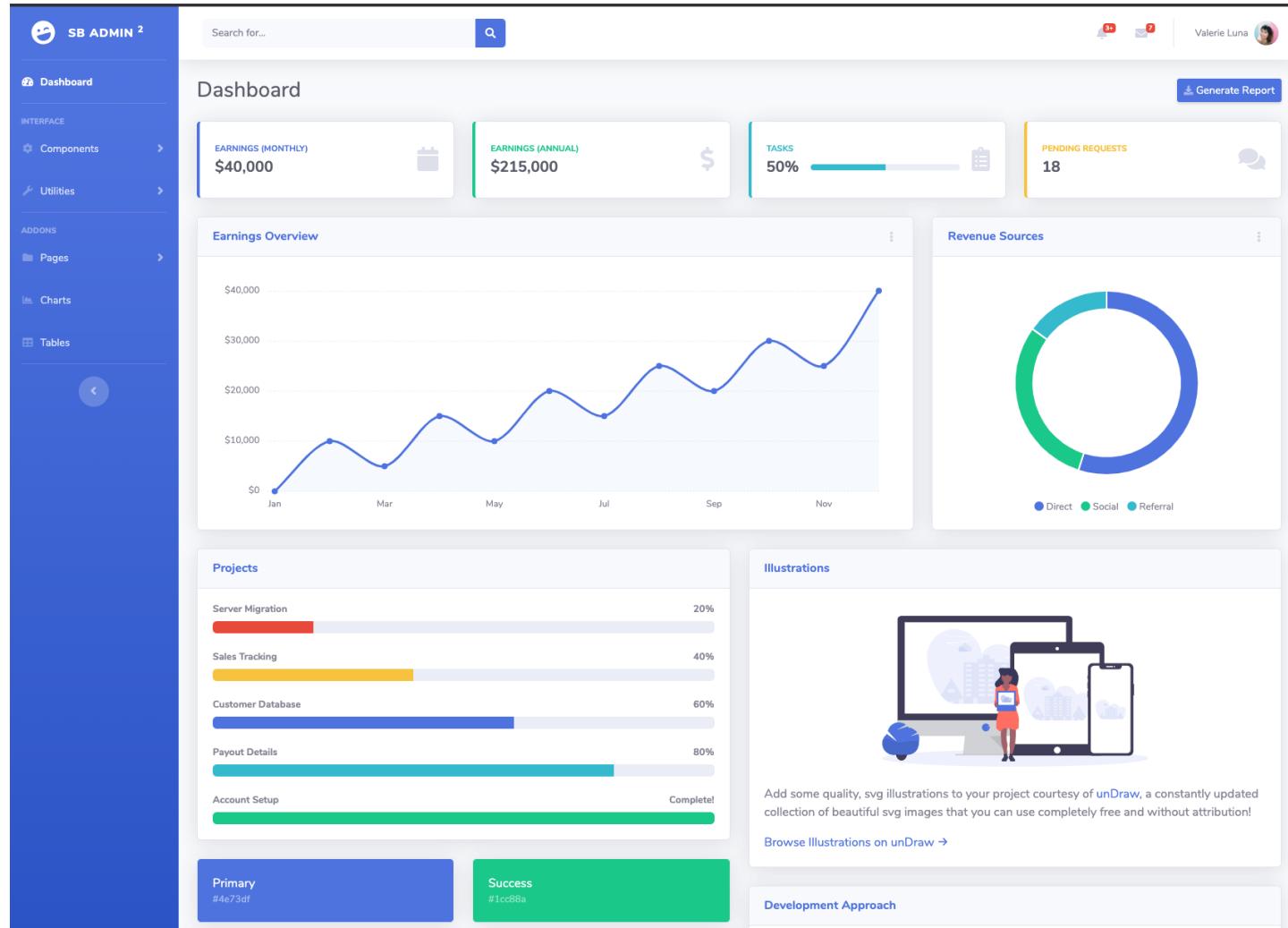
Launching an EC2 instance using the EC2 Console

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 Instances](#)

Advanced Topics

[AWS Nitro System - the system beyond standard virtualization](#)

[Auto Scaling](#)

[Elastic Load Balancing](#)