

Maria Valencia

Csc 154

Lab 13

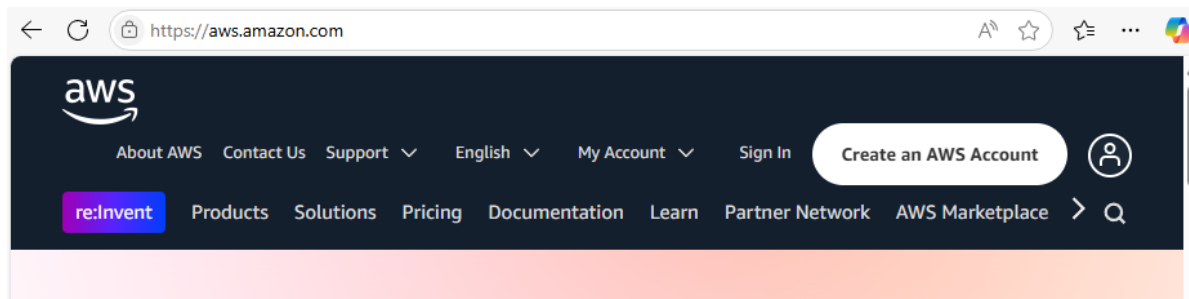
Cloud Security

Exercise 13.1 - Create and Setup AWS Account

In this task I will create an AWS account, secure the root user, and create an IAM administrator.

Step 1: Create AWS Account

From my host computer, I opened a browser and navigated to <https://aws.amazon.com/>. I pressed the "Create an AWS Account" button in the top right corner, entered my email address for the "Root user email address" and entered my name under the "AWS account name". Lastly, pressed "Verify email address".



Sign up for AWS

Root user email address

Used for account recovery and some administrative functions

⊗ An email address is required.

AWS account name

Choose a name for your account. You can change this name in your account settings after you sign up.

Verify email address

OR

Sign in to an existing AWS account

I entered the verification code that was sent to my email and pressed "Verify". Enter a "Root user password" and confirm the value then continue to the next step. The next step required my name, contact, address and use information. I select “personal” for the type of account. The next step required me to enter my billing information.

Sign up for AWS

Contact Information

How do you plan to use AWS?


☐ Business - for your work, school, or organization

☐ Personal - for your own projects

Who should we contact about this account?

Full Name

Country Code Phone Number

 +1

222-333-4444

Country or Region

United States

Address line 1

Address line 2

Apartment, suite, unit, building, floor, etc.

City

State, Province, or Region

Postal Code

☐ I have read and agree to the terms of the [AWS Customer Agreement](#).

Continue (step 2 of 5)

I then entered my payment information and press "Verify and Continue"

Sign up for AWS





Billing Information

Billing country

Your billing country determines the payment methods available to you to pay for AWS services.

United States

Credit or Debit card number



AWS accepts most major credit and debit cards. To learn more about payment options, review our [FAQ](#).

Expiration date

Month

Year

Security code ⓘ

CVV/CVC

Cardholder's name

Billing address

☒ Use my contact address

2327 volpi dr
Stockton California 95206
US

☐ Use a new address

Verify and Continue (step 3 of 5)

You might be redirected to your bank's website to authorize the verification charge.

The next step required me to confirm my identity. So, I entered my phone number and completed the CAPTCHA. Upon submission I received a text message with a numeric code and used that code to verify my identity. In the final step, I select "Basic support -Free" and "Complete sign up".

Sign up for AWS

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

☒ Text message (SMS)

☐ Voice call

Country or region code

United States (+1)

Mobile phone number

Send SMS (step 4 of 5)

Sign up for AWS

Confirm your identity

Verify code

Continue (step 4 of 5)

Having trouble? Sometimes it takes up to 10 minutes to retrieve a verification code. If it's been longer than that, [return to the previous page](#) and try again.



Congratulations !

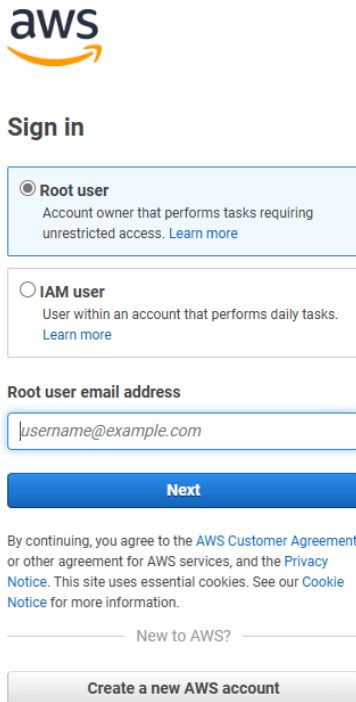
Thank you for signing up with AWS.

We are activating your account, which should take a few minutes. You will receive an email when this is complete.

Go to the AWS Management Console

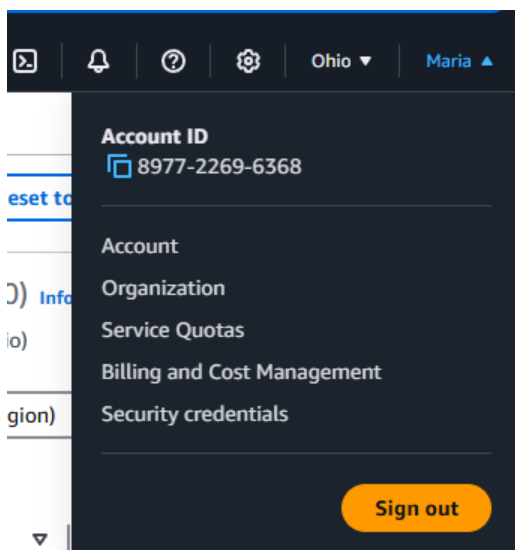
Step 2: Setup Root User MFA

With my AWS account setup, I sign in to the management console by pressing the "Sign In to the Console". I selected "Root user" and entered the email address I used to setup the account. Then I entered my password, and I was in as the root user.



The image shows the AWS Sign in page. At the top is the AWS logo. Below it is the "Sign in" heading. There are two selection boxes: "Root user" (selected) and "IAM user". The "Root user" box contains the text "Account owner that performs tasks requiring unrestricted access. [Learn more](#)". The "IAM user" box contains the text "User within an account that performs daily tasks. [Learn more](#)". Below these is a text input field labeled "Root user email address" containing the placeholder text "username@example.com". A blue "Next" button is below the input field. At the bottom, there is a link for "New to AWS?" and a button labeled "Create a new AWS account".

Once I logged in, I navigated to the user (root) settings selecting the account drop down menu in the upper right corner and pressing "Security credentials".



Then, I pressed "Assign MFA device" in the Multi-factor authentication table.

Multi-factor authentication (MFA) (0)

RemoveResyncAssign MFA device

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

Type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment			

Assign MFA device

I select a "Device name" and choose an MFA device. I used duo mobile as that is an authenticator app I already have because of school.

Step 1 of 2

Select MFA device [Info](#)

MFA device name


Device name
This name will be used within the identifying ARN for this device.

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

MFA device


Device options
In addition to username and password, you will use this device to authenticate into your account.

☐




Passkey or security key
Authenticate using your fingerprint, face, or screen lock. Create a passkey on this device or use another device, like a FIDO2 security key.

☒



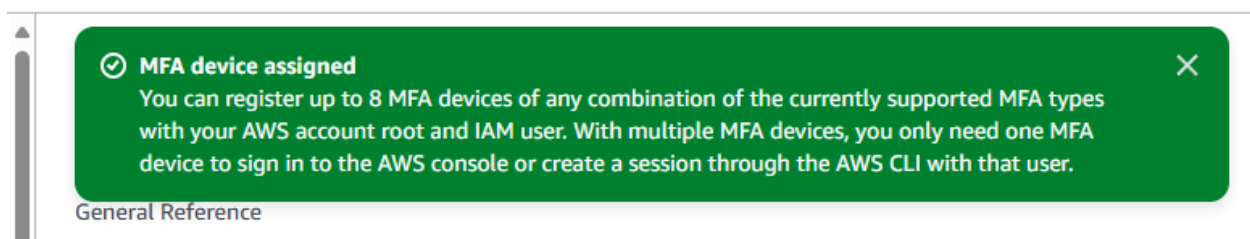
Authenticator app
Authenticate using a code generated by an app installed on your mobile device or computer.

☐



Hardware TOTP token
Authenticate using a code generated by Hardware TOTP token or other hardware devices.

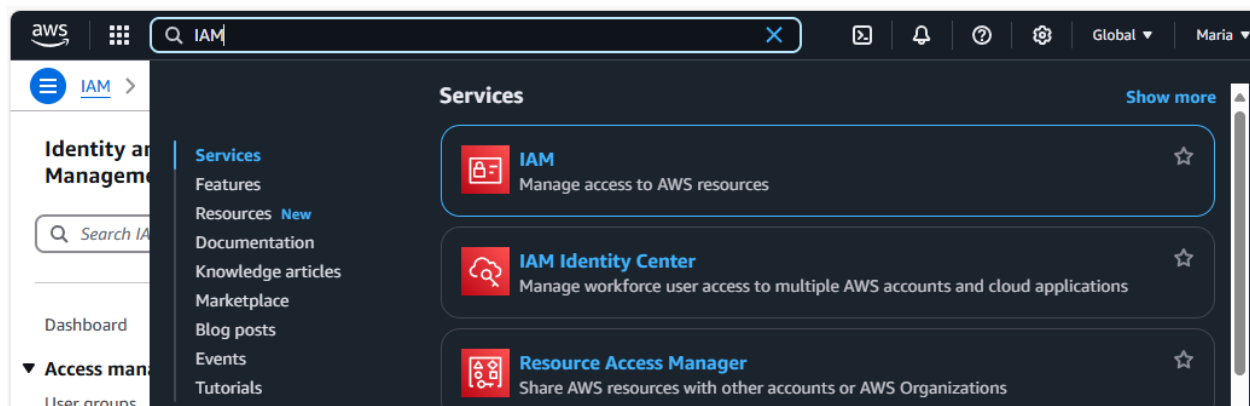
In duo mobile, I added AWS.



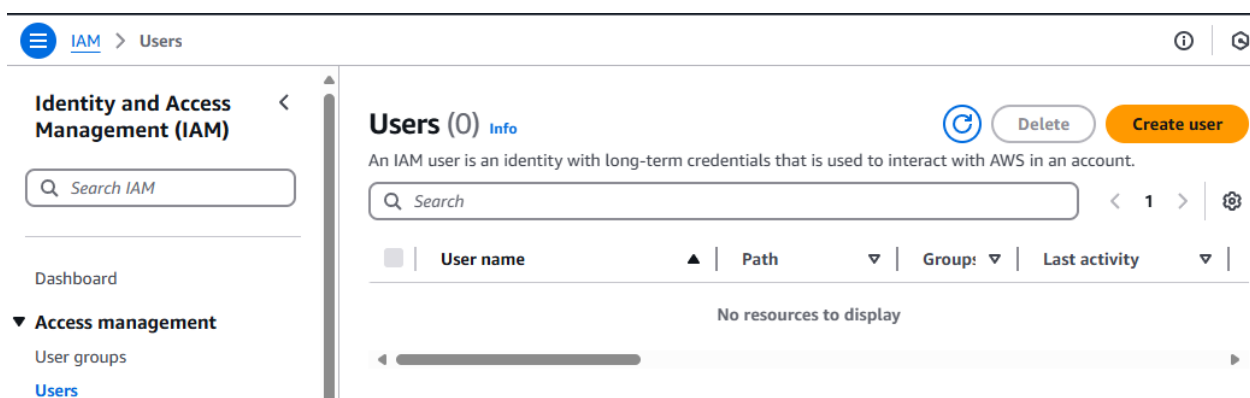
Step 3: Setup IAM User

Using the root user for administrative activities is considered a bad practice. So, I will create a non-root administrator IAM user for all activities in this AWS lab.

I searched for "IAM" in the services search bar at the top of the screen and followed the link for the IAM service.



Then, I selected "Users" in the left navigation pane to begin the process of creating a user.



I pressed the "Create user" button on the Users page. Then, entered my name as the username, checked the box "Provide user access to the AWS Management Console" (so the user can log into the console). Then in the bluebox, I selected the "I want to create an IAM user" option. Entered a strong password and unselected the "Users must create a new password at next sign-in" checkbox.

[illegible]

After pressing next, I set the IAM user permissions to administrator by selecting the "Attach policies directly" option and marking the " AdministratorAccess " policy checkbox. Then pressed "Next".

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- ☐ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- ☒ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1288)

 [Create policy](#)





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

Filter by Type

All types ▾

< 1 2 3 4 5 6 7 ... 65 >

⚙

	Policy name 	Type	Attached entities
<input type="checkbox"/>	 AccessAnalyzerSer...	AWS managed	0
<input checked="" type="checkbox"/>	 AdministratorAccess	AWS managed - job fun...	0

I reviewed the IAM creation settings and pressed "Create user". To create a new user.

Review and create

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

User details

User name
maria

Console password type
Custom password

Require password reset
No

Permissions summary

< 1 >

Name 	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user

After that, I signed out and then signed in as the IAM user. To sign in, I used my AWS account number, IAM username and password.

IAM user sign in

Account ID (12 digits) or account alias

897722696368

IAM username

maria

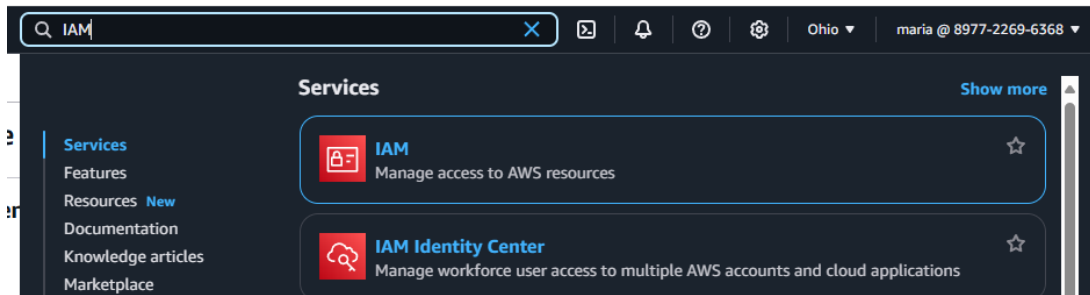
Password

☐ Show Password

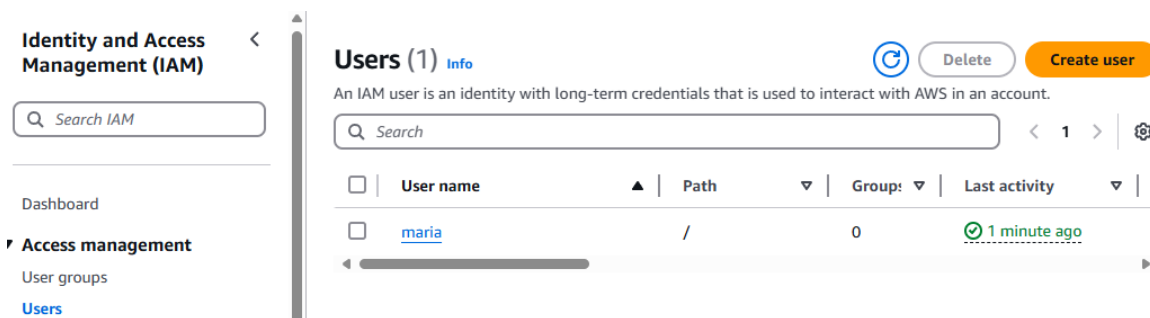
[Having trouble?](#)

Sign in

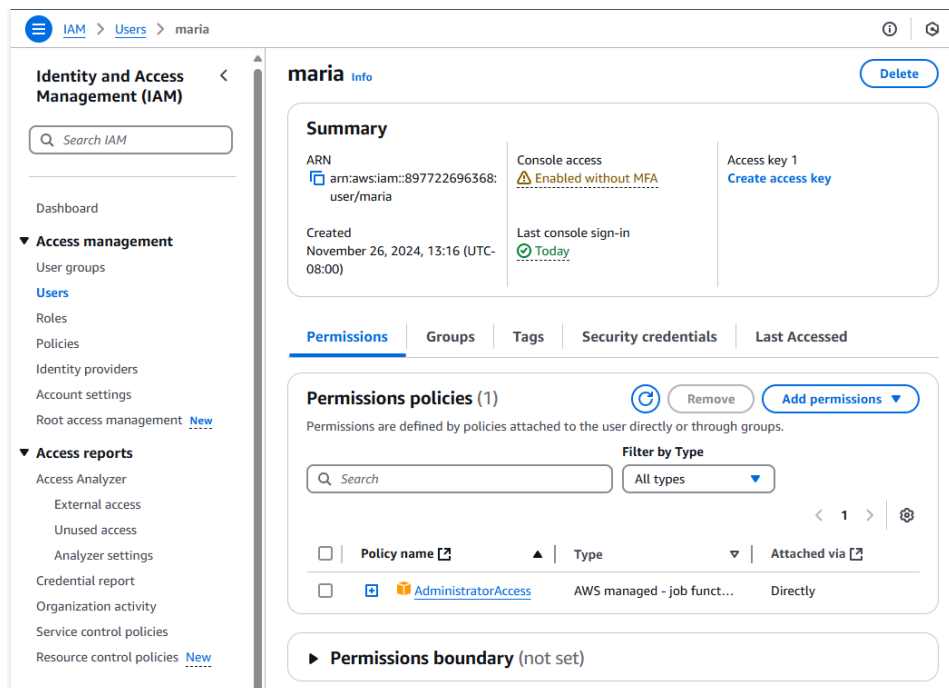
While logged in as my IAM administrator user (not root), I navigated to the IAM service by searching IAM in the search bar (top bar) and selecting the IAM service link.



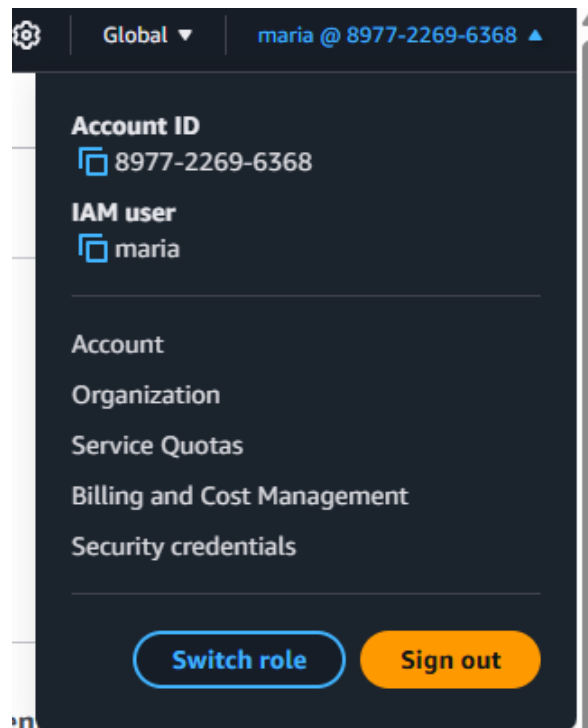
I selected "Users" in the left navigation menu to observe all IAM users.



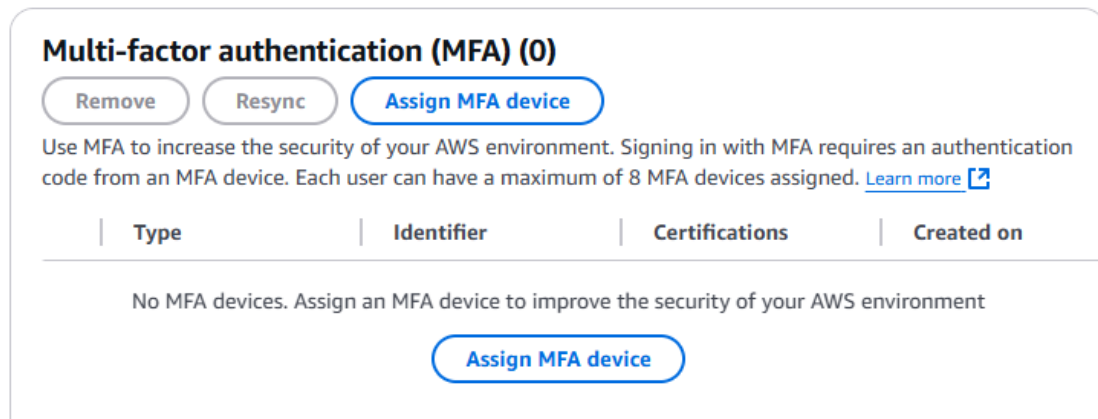
Then, I selected the IAM user I created (and are logged in as) to view its settings.



After viewing its settings, I selected the user "Security credentials" tab.




Then, I set up an MFA device following the same procedure as the root MFA device (using duo mobile). I named the device “marsnotroot” to distinguish it from the root account.





MFA device name
Device name
This name will be used within the identifying ARN for this device.

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

MFA device
Device options
In addition to username and password, you will use this device to authenticate into your account.



☐**Passkey or security key**
Authenticate using your fingerprint, face, or screen lock. Create a passkey on this device or use another device, like a FIDO2 security key.

☒**Authenticator app**
Authenticate using a code generated by an app installed on your mobile device or computer.

☐**Hardware TOTP token**
Authenticate using a code generated by Hardware TOTP token or other hardware devices.

[Cancel](#) [Next](#)

In duo mobile, I added AWS.

 **MFA device assigned** 

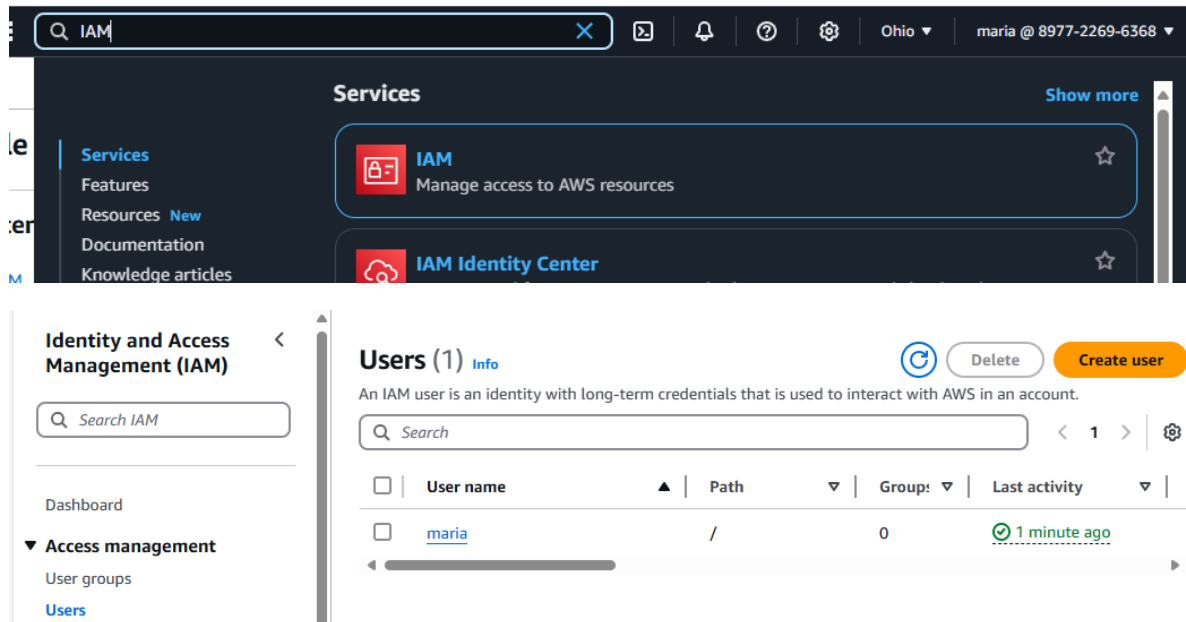
You can register up to 8 MFA devices of any combination of the currently supported MFA types with your AWS account root and IAM user. With multiple MFA devices, you only need one MFA device to sign in to the AWS console or create a session through the AWS CLI with that user.

Exercise 13.2 - Scout Suite CSPM

In this task I will create an IAM user with limited permissions and scan my AWS account to detect security misconfigurations. I will then identify and remediate security issues raised by the scanning tool.

Step 1: Create IAM User

I logged into my AWS account using my administrator IAM user (not root). Then, navigated to the IAM service and the Users page.



I pressed the "Create user" button to start the user creation process and entered "auditor" as the username. I left "Provide user access to the AWS Management Console" unchecked. Then pressed Next.

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Info If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

In the next step of the wizard, I chose "Attach policies directly". Then, I used the filter "AWS managed - job function" and selected "ReadOnlyAccess" and "SecurityAudit" permission policies.

● Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (2/1288)

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

Search

Filter by Type

AWS managed - ...

11 matches

< 1 >

⚙

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AdministratorAccess	AWS managed - job func...	1
<input type="checkbox"/>	Billing	AWS managed - job func...	0
<input type="checkbox"/>	DatabaseAdministr...	AWS managed - job func...	0
<input type="checkbox"/>	DataScientist	AWS managed - job func...	0
<input type="checkbox"/>	NetworkAdministr...	AWS managed - job func...	0
<input type="checkbox"/>	PowerUserAccess	AWS managed - job func...	0
<input checked="" type="checkbox"/>	ReadOnlyAccess	AWS managed - job func...	0
<input checked="" type="checkbox"/>	SecurityAudit	AWS managed - job func...	0

Then pressed Next and then "Create user".

Review and create

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

User details

User name

auditor

Console password type

None

Require password reset

No

Permissions summary

< 1 >

Name	Type	Used as
ReadOnlyAccess	AWS managed - job function	Permissions policy
SecurityAudit	AWS managed - job function	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user

Now, I see it in my users. I selected the created user "auditor" from the Users page.

Users (2) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

< 1 >

<input type="checkbox"/>	User name ▲	Path ▼	Group: ▼	Last activity ▼
<input type="checkbox"/>	auditor	/	0	-
<input type="checkbox"/>	maria	/	0	6 minutes ago

After, I navigated to the "Security credentials" tab and scroll down to the "Access keys" section.

Access keys (0) [Create access key](#)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

[Create access key](#)

Then, I pressed the "Create access key" button to create a user token that can be used within the command line interface.

Access key best practices & alternatives [Info](#)

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

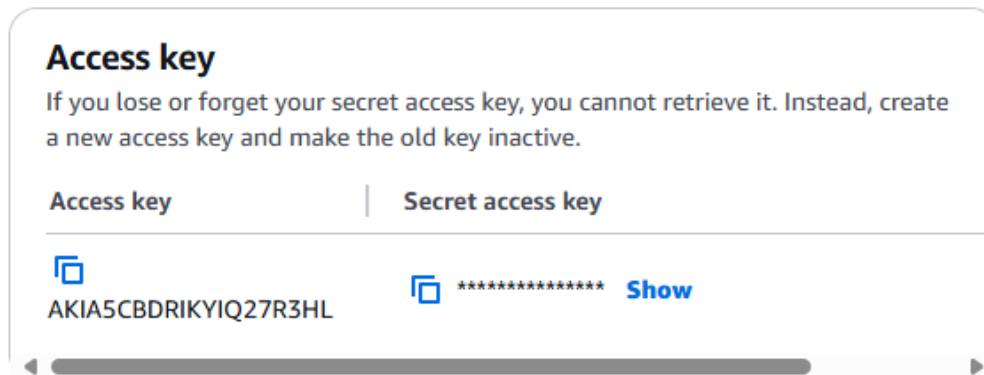
Use case

☒ **Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.

☐ Local code

With the create access key wizard launched, I chose the "Command Line Interface (CLI)" option and agreed to the confirmation. Then press Next and then "Create access key".

Retrieve access keys [Info](#)



Step 2: Install and Configure AWS CLI

I launched my Ubuntu VM with Bridge Adapter network mode and opened a terminal. I made sure to update my system and then install the AWS CLI tool using the commands below.

```
maria@ubuntu:~/Desktop$ sudo apt install -y
[sudo] password for maria:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
maria@ubuntu:~/Desktop$ sudo apt install awscli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

After a couple of minutes, the AWS CLI is installed and can be configured using the AWS version command.

```
maria@ubuntu:~/Desktop$ aws --version
aws-cli/1.22.34 Python/3.10.12 Linux/6.8.0-49-generic botocore/1.23.34
maria@ubuntu:~/Desktop$
```

Once the AWS CLI is installed, I configured the tool to use the "auditor" IAM credentials created in the previous step. After using the `aws configure --profile auditor` command, I entered my access key, secret key, region as "us-west-2", and output format as "json".


```
maria@ubuntu:~/Desktop$ aws configure --profile auditor
AWS Access Key ID [None]: AKIA5CBDRIKIQ27R3HL
AWS Secret Access Key [None]:
```

```
Default region name [None]: us-west-2
Default output format [None]: json
maria@ubuntu:~/Desktop$
```

Step 3: Setup and Run Scout Suite

I updated sudo and installed Python virtual environment using the commands below.

```
maria@ubuntu:~/Desktop$ sudo apt update -y
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
```

```
maria@ubuntu:~/Desktop$ sudo apt install python3-virtualenv -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

Here, I created a python virtual environment to run Scout Suite from to avoid any Python library conflicts. Using the commands below I:

- I created the virtual environment using the command below

```
maria@ubuntu:~/Desktop$ virtualenv -p python3 venv
created virtual environment CPython3.10.12.final.0-64 in 396ms
creator CPython3Posix(dest=/home/maria/Desktop/venv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/maria/.local/share/virtualenv)
added seed packages: pip==22.0.2, setuptools==59.6.0, wheel==0.37.1
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
maria@ubuntu:~/Desktop$ source venv/bin/activate
```

- installed scout using the command below

```
(venv) maria@ubuntu:~/Desktop$ pip install scoutsuite
Collecting scoutsuite
  Downloading ScoutSuite-5.14.0-py3-none-any.whl (3.5 MB)
    3.5/3.5 MB 16.0 MB/s eta 0:00:00
Collecting google-cloud-kms==1.3.0
  Downloading google_cloud_kms-1.3.0-py2.py3-none-any.whl (65 kB)
    65.3/65.3 KB 3.7 MB/s eta 0:00:00
Collecting azure-mgmt-keyvault==8.0.0
  Downloading azure_mgmt_keyvault-8.0.0-py2.py3-none-any.whl (197 kB)
```

- verified its installation using the command below

```
(venv) maria@ubuntu:~/Desktop$ scout --help
usage: scout [-h] [-v] {aws,gcp,azure,aliyun,oci,kubernetes,do} ...

Files
  -h, --help            show this help message and exit
  -v, --version          show program's version number and exit

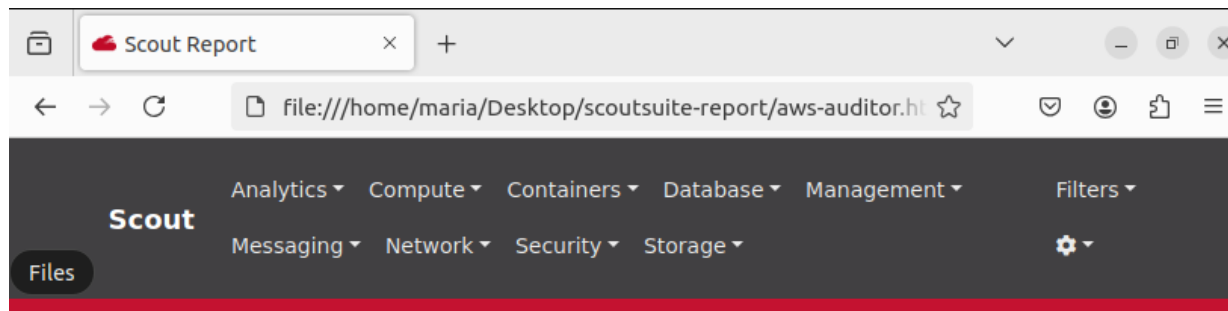
The provider you want to run scout against:
{aws,gcp,azure,aliyun,oci,kubernetes,do}
  aws                  Run Scout against an Amazon Web Services account
  gcp                  Run Scout against a Google Cloud Platform account
  azure                Run Scout against a Microsoft Azure account
  aliyun               Run Scout against an Alibaba Cloud account
  oci                  Run Scout against an Oracle Cloud Infrastructure
                      account
  kubernetes           Run Scout against a Kubernetes cluster
  do                   Run Scout against an DigitalOcean account

To get additional help on a specific provider run: scout.py {provider} -h
(venv) maria@ubuntu:~/Desktop$
```









After, I ran a posture scan to discover any potential security misconfigurations. Scout will make API calls to all AWS services using the auditor account and compare results to a rules engine that identifies any potential security flaws.

```
(venv) maria@ubuntu:~/Desktop$ scout aws --profile auditor
2024-11-26 14:44:24 ubuntu scout[7163] INFO Launching Scout
2024-11-26 14:44:24 ubuntu scout[7163] INFO Authenticating to cloud provider
2024-11-26 14:44:27 ubuntu scout[7163] INFO Gathering data from APIs
2024-11-26 14:44:27 ubuntu scout[7163] INFO Fetching resources for the ACM service
2024-11-26 14:44:27 ubuntu scout[7163] INFO Fetching resources for the Lambda service
2024-11-26 14:44:28 ubuntu scout[7163] INFO Fetching resources for the CloudFormation service
2024-11-26 14:44:28 ubuntu scout[7163] INFO Fetching resources for the CloudTrail service
2024-11-26 14:44:28 ubuntu scout[7163] INFO Fetching resources for the CloudWatch service
```

Once the scan was completed, an HTML report was generated and automatically opened in my VM's browser.



Amazon Web Services > 897722696368

Dashboard				
Service	Resources	Rules	Findings	Checks
 ACM	0	2	0	0
 Lambda	0	0	0	0
 CloudFormation	0	1	0	0
 CloudFront	0	3	0	0
 CloudTrail	0	9	17	17
 CloudWatch	0	1	0	0
 Codebuild	0	0	0	0
 Config	0	1	17	17

Step 4: Analyze and Fix

In this step, I will be selecting a vulnerability that was identified by scout suite, research its vulnerability and try to “cure” the vulnerability.

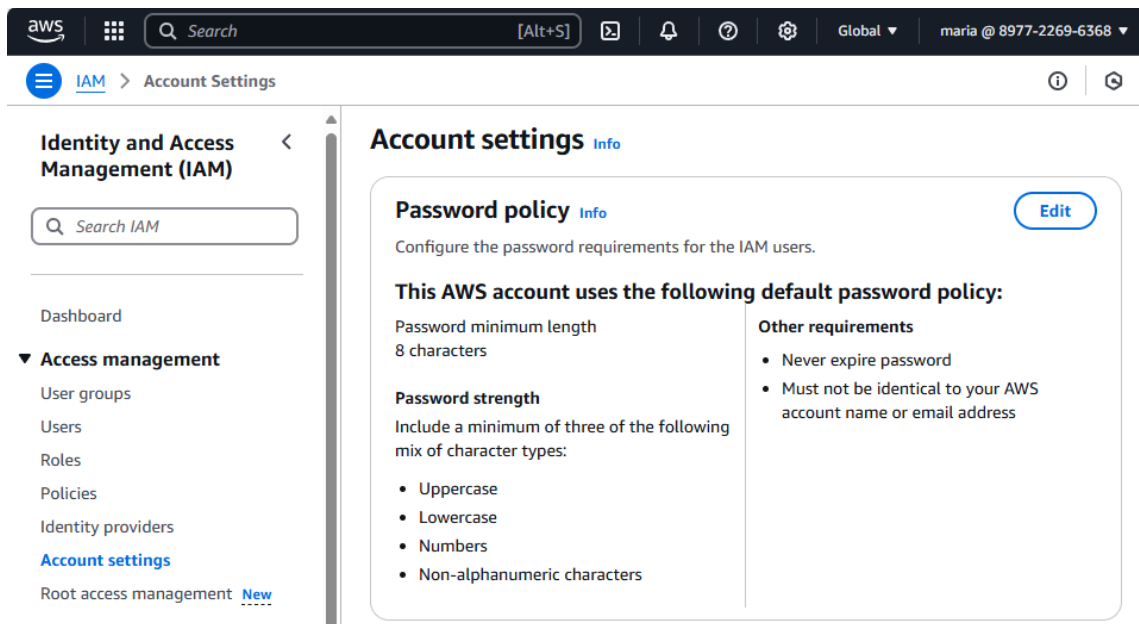
I chose the IAM service, specifically the password policy that allows the reuse of passwords.

1. Description of the vulnerable AWS service:

- a. The password policy allowed password reuse. As a result, password complexity requirements were not in line with security best practice.
2. How is security impacted by the vulnerability/misconfiguration:
 - a. It undermines the principle of credential hygiene, increasing the risk of unauthorized access and attackers can exploit reused passwords from previous breaches.
3. How can the service be fixed (what steps are needed):
 - a. Ensure the password policy is configured to prevent password reuse.
 - b. In order to do that, I log into the AWS console, navigate to IAM > Account settings > password policy. Enable prevent password reuse. Save changes .

Next, cure the vulnerability/misconfiguration and re-run Scout to confirm the issue no longer exists in the report. Please keep in mind that some fixes may incur a cost and you should:

- I log into the AWS console, navigate to IAM > Account settings



- In edit password policy, I check the “prevent password reuse” box and added “12” to the “remember x passwords” and clicked save changes.

Edit password policy [Info](#)

Password policy

☐ IAM default
Apply default password requirements.

☒ Custom
Apply customized password requirements.

Password minimum length.
Enforce a minimum length of characters.
 characters
Needs to be between 6 and 128.

Password strength
☐ Require at least one uppercase letter from the Latin alphabet (A-Z)
☐ Require at least one lowercase letter from the Latin alphabet (a-z)
☐ Require at least one number
☐ Require at least one non-alphanumeric character (!@#\$%^&*()_+-=[]{}|')

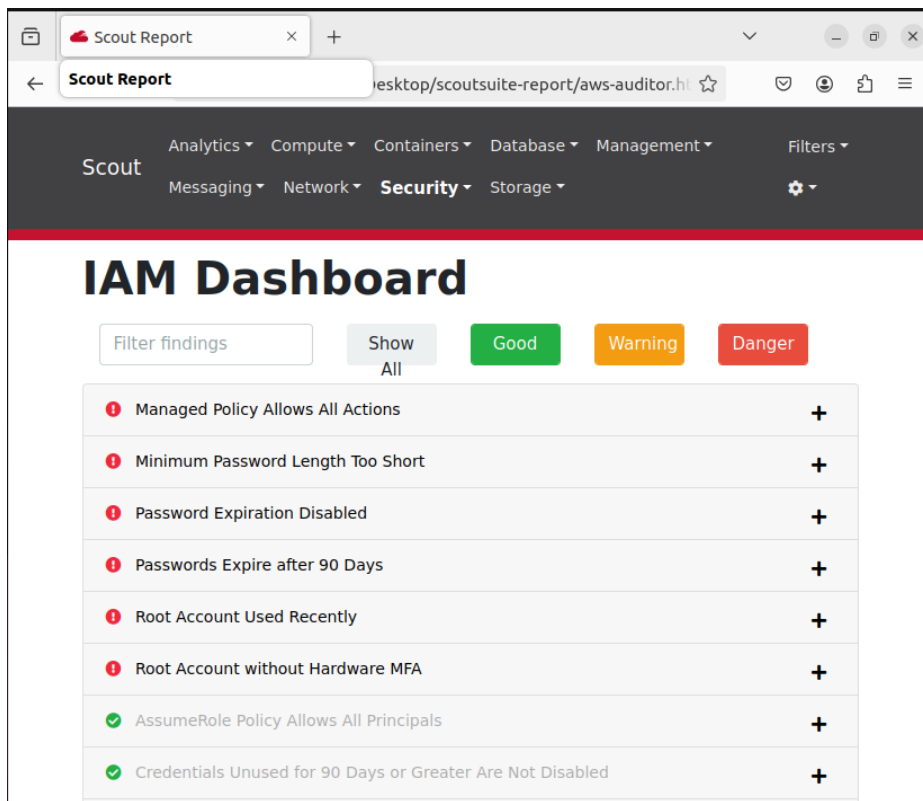
Other requirements
☐ Turn on password expiration
☐ Password expiration requires administrator reset
☐ Allow users to change their own password
☒ Prevent password reuse
Remember password(s)
Needs to be between 1 and 24.

[Cancel](#)[Save changes](#)

- Here I ran scout again on the auditor profile like previously and waited for an HTML report to pop up on my VM.

```
To get additional help on a specific provider run: scout.py {provider} -h
(venv) maria@ubuntu:~/Desktop$ scout aws --profile auditor
2024-11-26 15:02:04 ubuntu scout[9335] INFO Launching Scout
2024-11-26 15:02:04 ubuntu scout[9335] INFO Authenticating to cloud provider
2024-11-26 15:02:06 ubuntu scout[9335] INFO Gathering data from APIs
2024-11-26 15:02:06 ubuntu scout[9335] INFO Fetching resources for the ACM service
2024-11-26 15:02:06 ubuntu scout[9335] INFO Fetching resources for the Lambda service
2024-11-26 15:02:07 ubuntu scout[9335] INFO Fetching resources for the CloudFormation service
2024-11-26 15:02:07 ubuntu scout[9335] INFO Fetching resources for the CloudTrail service
```

- The HTML report popped up and the Password policy allows the reuse of passwords is fixed! It no longer pops up



For reference, this is what the IAM dashboard looked like before I cured the vulnerability.

