Immersion Day

IAM Hands-On Lab

Getting Started with Identity & Access Management

Identity & Access Management (IAM) Overview

AWS Identity and Access Management (IAM) is a free service that enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

In the following lab, you will create one of each of the three types of AWS IAM entities: groups, users, and roles. You will also learn about IAM policy documents, which you can attach to entities in order to give them permissions to access AWS resources and services.

Create an IAM Group

1. To get started, go to the **IAM dashboard** in the AWS console.

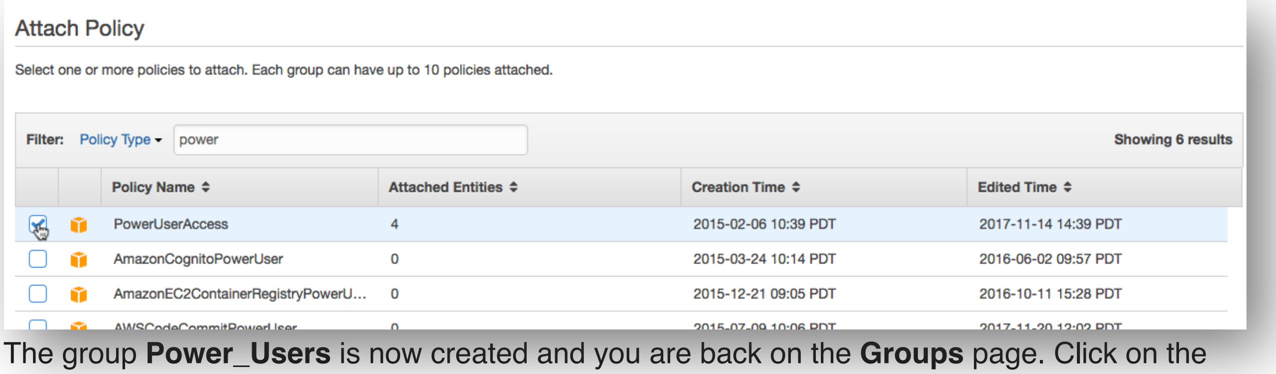
An IAM group is a collection of users. Groups are often based on job function or role. They allow you to manage permissions by applying policies to each group rather than individual users.

2. Go to **Groups** in the sidebar and click on **Create New Group.**
3. Type in Power_Users as the Group Name. Go to the Next Step.

Now we will attach a policy to the group. Policies are what give IAM entities permissions. AWS provides managed policies for many common access needs, but you can also write your own custom policies if needed.

For this lab, we are attaching an AWS managed policy called **PowerUserAccess**. This policy allows all AWS actions except for most IAM actions. This will prevent **Power_Users** from giving themselves more permissions.

4. Attach this policy to your group by typing **power** into the search bar, and then selecting the correct policy. Go to the **Next Step** and then **Create Group**.



Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Filter: Policy Type ▾ power   Showing 6 results

		Policy Name ⬍	Attached Entities ⬍	Creation Time ⬍	Edited Time ⬍
☑		PowerUserAccess	4	2015-02-06 10:39 PDT	2017-11-14 14:39 PDT
□		AmazonCognitoPowerUser	0	2015-03-24 10:14 PDT	2016-06-02 09:57 PDT
□		AmazonEC2ContainerRegistryPowerU…	0	2015-12-21 09:05 PDT	2016-10-11 15:28 PDT
□		AWSCodeCommitPowerUser	0	2015-07-09 10:06 PDT	2017-11-20 12:02 PDT

The group **Power_Users** is now created and you are back on the **Groups** page. Click on the **Power_Users** group and go to the **Permissions** tab.

We're not going in depth on IAM policy documents in this lab, but if you're curious, you should view the JSON content of the **PowerUserAccess** policy by clicking on **Show Policy**.

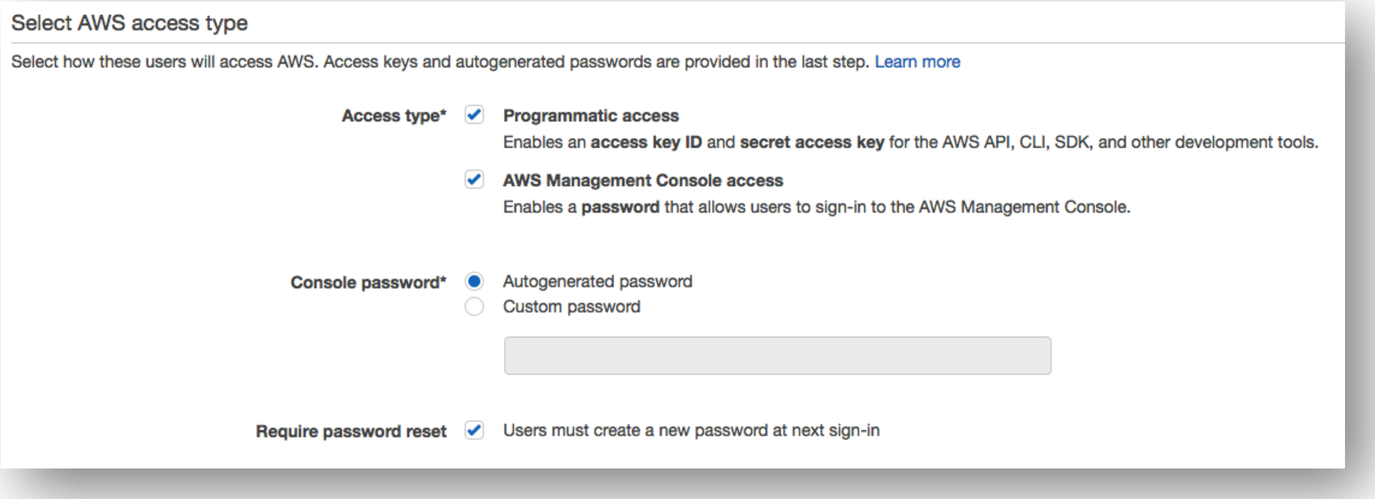
Awesome! You just created an IAM group with a policy attached which will give any user in this group permissions to do anything except for most IAM actions, such as giving themselves or others additional permissions. Next, we will create a user to put into this group.

Create an IAM User

It is best practice to create a different user for each person who needs AWS access. A user is associated with permanent credentials and has permissions based off of policies attached directly to the user or policies attached to a group to which the user belongs.

1. Go to **Users** in the sidebar and click on **Add user.**

Enter a unique name as the **User name**. In this walkthrough, I chose **Emily** as the username.
For this lab, we will choose to give this user both **Programmatic access** and **AWS Management Console access**. In practice, follow the principle of **least privilege** and only give users the type(s) of access necessary for their work. This prevents unnecessary extra credentials from being created.



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.

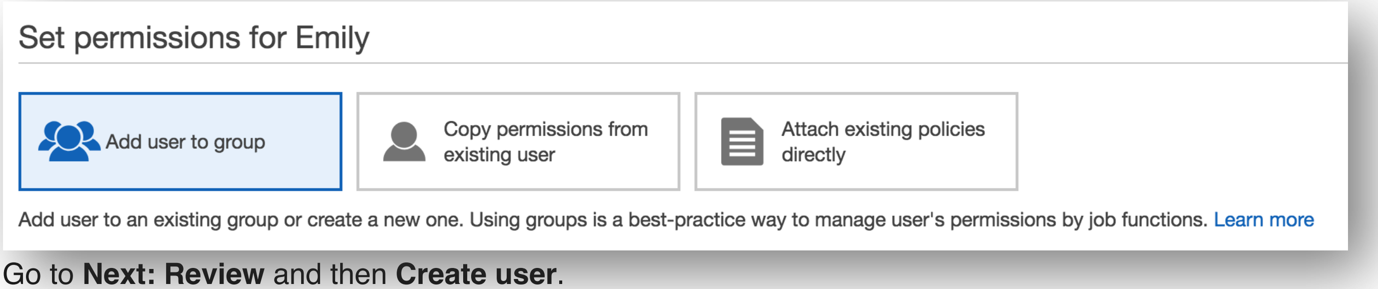
Console password* ● Autogenerated password
○ Custom password

Require password reset ☑ Users must create a new password at next sign-in

For **console access**, you can choose to have AWS **autogenerate** a password or you can type in a **custom** password here. Either way, you have the option of emailing this password to the user at the end of this process. Click **Next.**

Now we will **Add user** to the **group** we made in the previous step, **Power_Users**. There are other options which would attach policies directly to this user, but as mentioned before, using IAM groups is a best practice in order to simplify policy management.

If the top of your page doesn't look like the screenshot below, you need to scroll up.



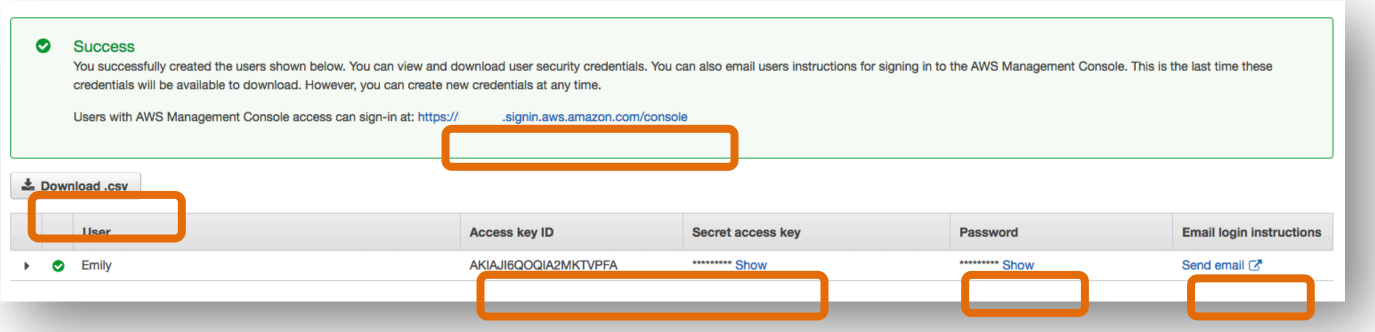
Set permissions for Emily

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

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

Go to **Next: Review** and then **Create user**.

After clicking create user, you will be presented with all of the information you will need to get that user started working in AWS.



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://   .signin.aws.amazon.com/console

User	Access key ID	Secret access key	Password	Email login instructions
Emily	AKIAJI6QOQIA2MKTVPFA	********* Show	********* Show	Send email

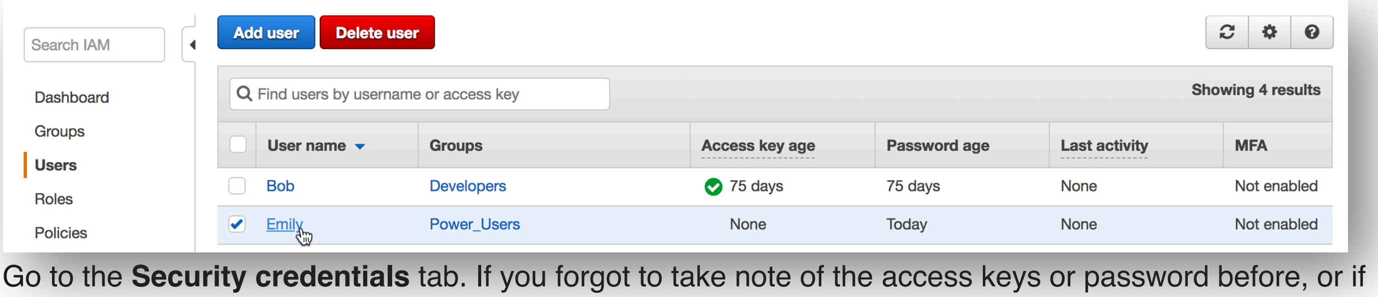
For **programmatic access**, the user will need their **Access key ID** and **Secret access key**. Take note of the Secret access key now or **Download .csv.** *You will never be able to retrieve the secret key once you leave this page.*

For **console access**, the user will need the **IAM users sign-in link:** https://[account ID or alias].signin.aws.amazon.com/console. Or, they can go to https://console.aws.amazon.com and login as long as they know the **account ID or alias**.

They will also need to know their **username** and **password**. Take note of the password, **Download .csv**, or **Send** an **email** to the user with their login information. If you decide to go down the email route, remember to fill in the username in the pre-written email and send the password in a separate email. You will not be able to see the password once you leave this page.

For this lab, you will need to have the account ID/alias, username, and password.
Once you're done, **Close** out of this page.

Back on the **Users** page, click on the **User name** of the user that you just created.



Showing 4 results

☐	User name	Groups	Access key age	Password age	Last activity	MFA
☐	Bob	Developers	✓ 75 days	75 days	None	Not enabled
☑	Emily	Power_Users	None	Today	None	Not enabled

Go to the **Security credentials** tab. If you forgot to take note of the access keys or password before, or if credentials have been compromised, here is where you can reset the password (click on **Manage password**), change access keys to **inactive**, and create new access keys.
Great! You have just created an IAM user with both programmatic and console access. You didn't have to attach policy permissions directly to the user since you had a group created already with the correct permissions.

Test IAM User Access

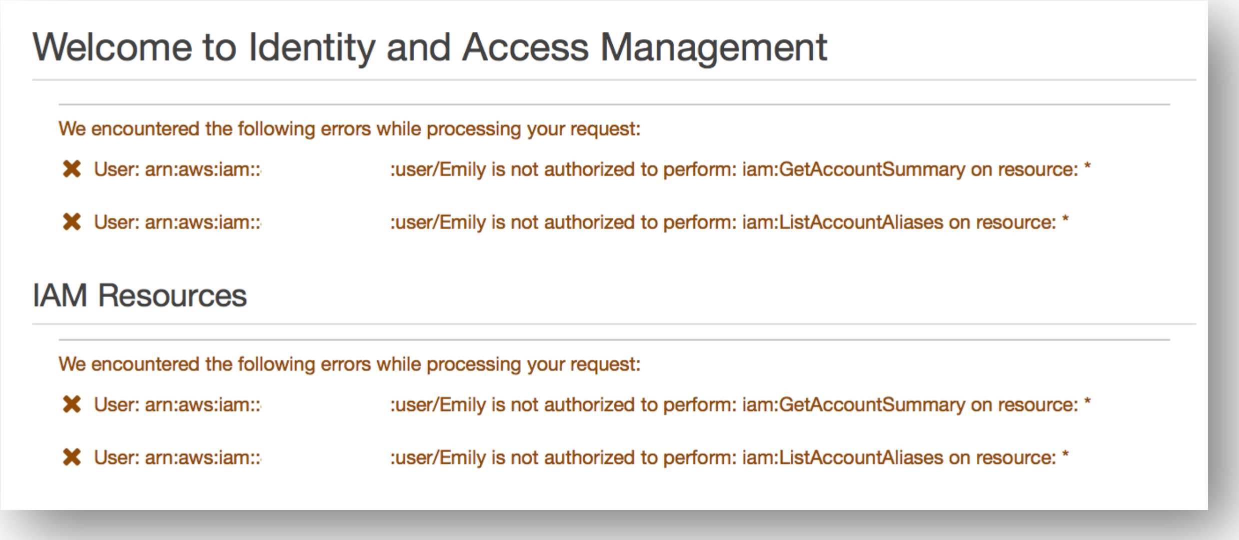
In this section, we will put your IAM User to the test. You will want to stay logged in as you currently are while at the same time testing the user's access. To do this, you will need to either open a **different browser** or open **private browsing** in your current browser. You will use this new window to sign in as the user you just created. If you didn't take note of the password or login link previously, now is the time to do so.

Go to your account's IAM user sign-in page. You could also just go to console.aws.amazon.com and enter your account ID or alias. Sign in using the IAM User **username** and **password**.

You can tell which user you are signed in as by looking at the top right corner of the console. The dropdown menu will now be titled **User @ Account ID/Alias.**

We know from the IAM Group section that this user has access to everything except most IAM permissions. Go to the **IAM dashboard** to see this in action.

Now instead of seeing a summary of IAM users, groups, and roles, you will see multiple errors.



Welcome to Identity and Access Management

We encountered the following errors while processing your request:

✖ User: arn:aws:iam:: :user/Emily is not authorized to perform: iam:GetAccountSummary on resource: *

✖ User: arn:aws:iam:: :user/Emily is not authorized to perform: iam:ListAccountAliases on resource: *

IAM Resources

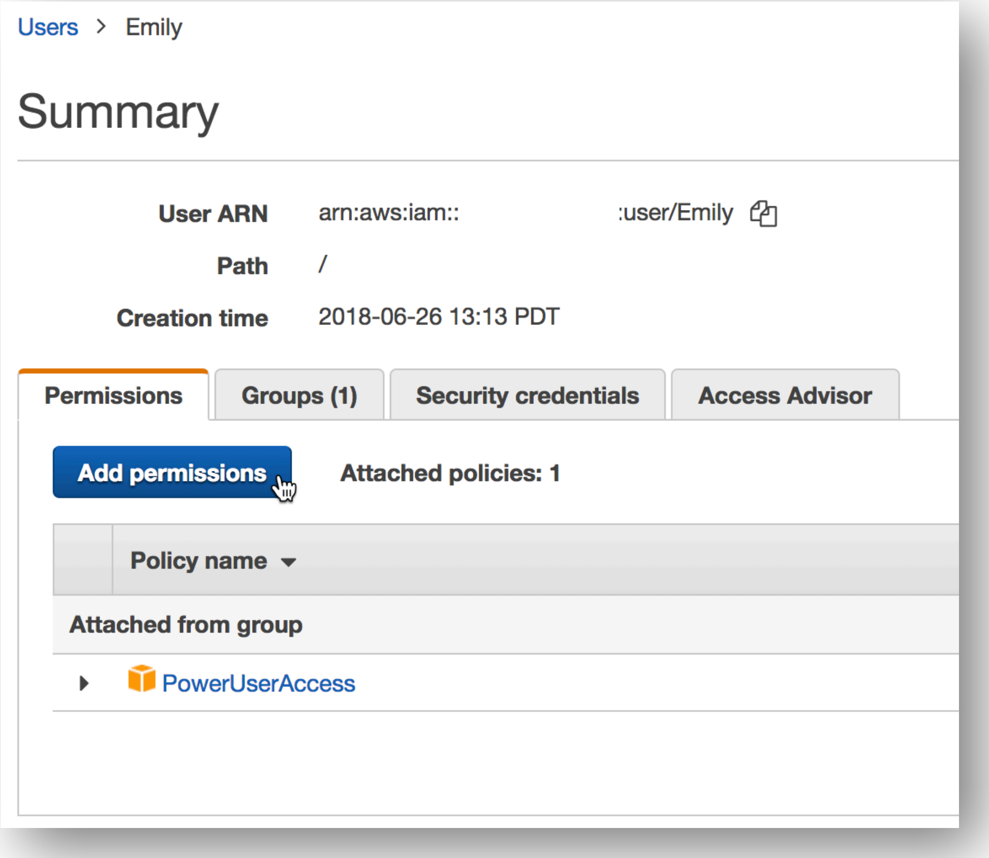
We encountered the following errors while processing your request:

✖ User: arn:aws:iam:: :user/Emily is not authorized to perform: iam:GetAccountSummary on resource: *

✖ User: arn:aws:iam:: :user/Emily is not authorized to perform: iam:ListAccountAliases on resource: *

Now we will give the user extra permissions in order to allow access to IAM.

Go to your other browser. You should still be in the **Users** dashboard on this user's page. Go to the **Permissions** tab and click on **Add permissions**. Adding permissions this way will attach policies directly to the user rather than to the entire group.



Summary

User ARN arn:aws:iam::   :user/Emily

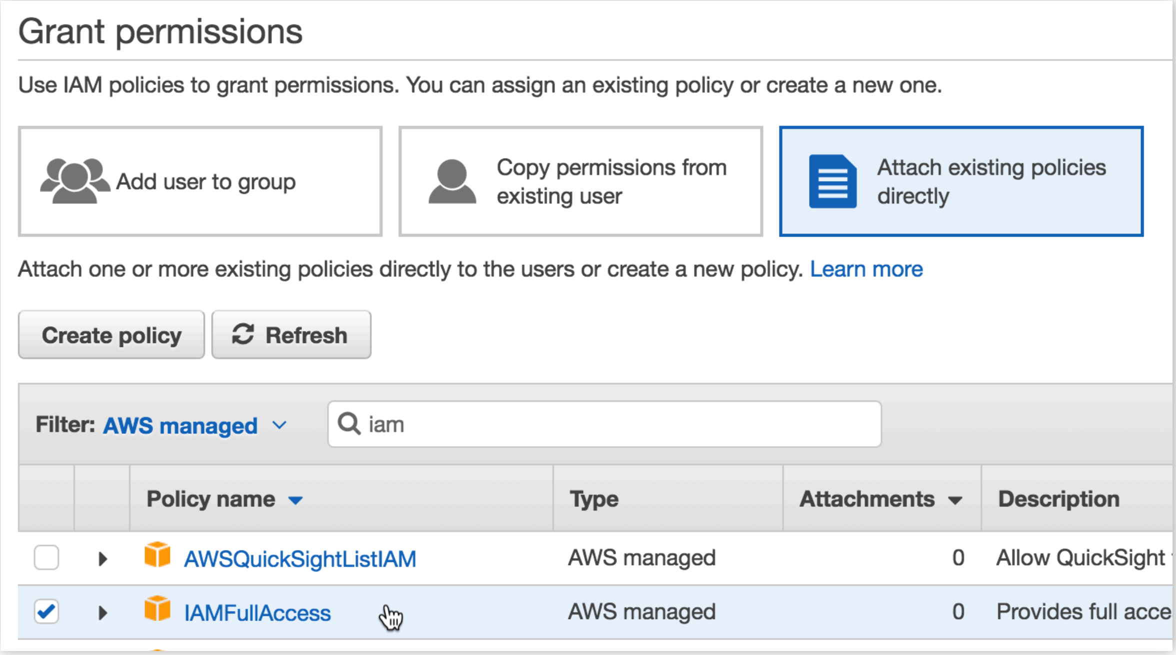
Path /

Creation time 2018-06-26 13:13 PDT

Attached policies: 1

	Policy name ▾
Attached from group	
▸	PowerUserAccess

This time, we will **Attach existing policies directly**. Type IAM into the search bar to find a managed policy which will give this user access to IAM. Select the **IAMFullAccess** policy and click on **Next: Review** and then **Add permissions**.



Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

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

Attach one or more existing policies directly to the users or create a new policy. Learn more

Filter: AWS managed ⌄ iam

		Policy name	Type	Attachments	Description
□	▸	AWSQuickSightListIAM	AWS managed	0	Allow QuickSight
☑	▸	IAMFullAccess	AWS managed	0	Provides full acce

Go back to the browser where you are signed in as the user. IAM policies are evaluated every time an action is attempted, so we just need to **refresh** the page to see the changes we made take effect. The IAM dashboard should now show content rather than errors.

Congrats! You just tested an IAM user's access to the console and modified their privileges as needed.

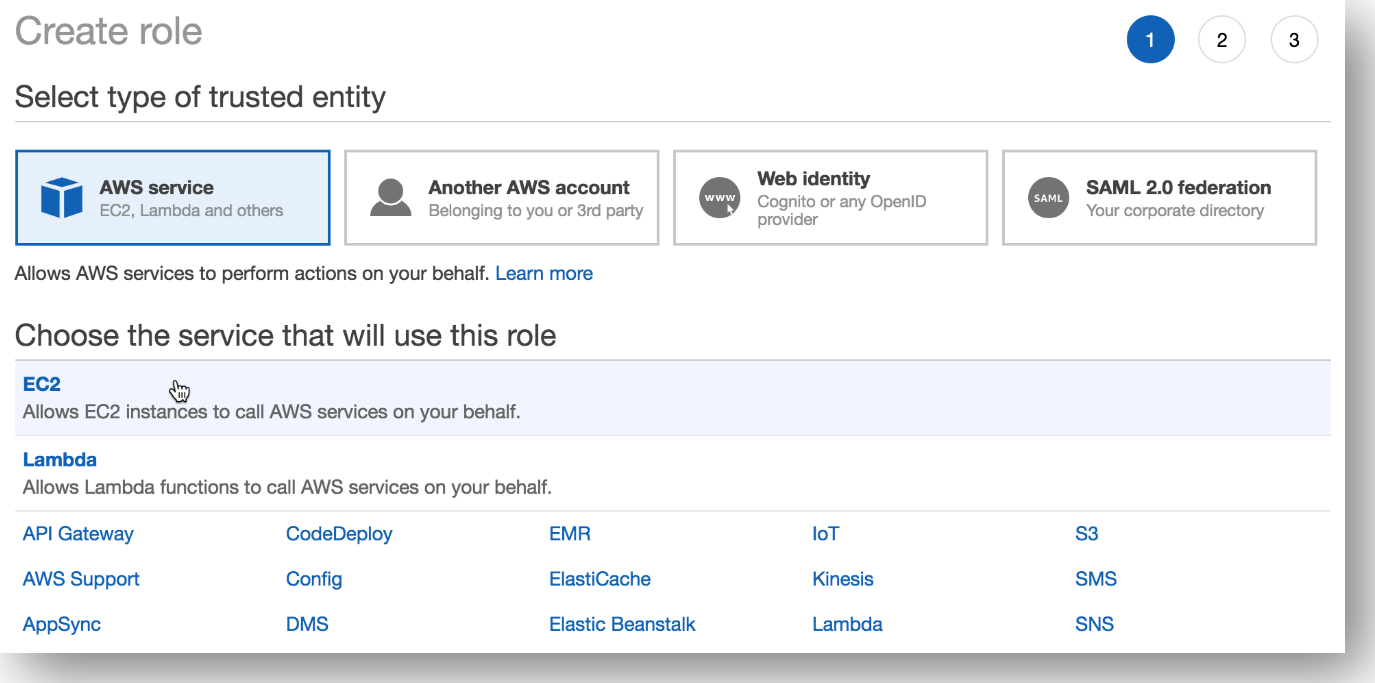
Create an IAM Role

IAM Roles can be assumed by AWS services, IAM users, or applications. They are assigned temporary rather than permanent credentials whenever assumed. Using roles for privileged permissions sets (such as the **PowerUserAccess** policy) can help improve your security posture since credential exposure is minimized.

1. Go to **Roles** in the sidebar and click on **Create Role.**

2. On the **Select type of trusted identity** page, you decide who or what will be able to assume this role. For this lab, we will create a role that allows an EC2 instance to read files in S3. Therefore,

we will stay on the **AWS service** tab and select **EC2**. Go to **Next: Permissions**.



Create role

Select type of trusted entity

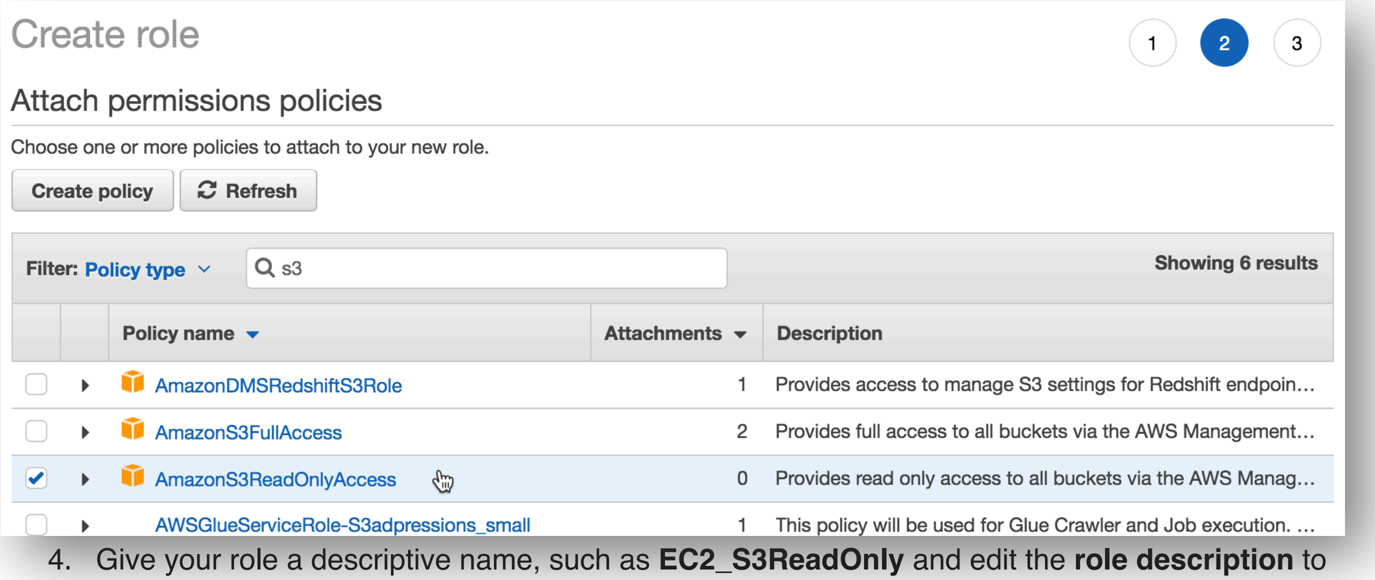
Allows AWS services to perform actions on your behalf. Learn more

Choose the service that will use this role

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

3. Attach a managed policy with S3 Read Only access to the role by typing **s3** into the search bar, and then selecting the **AmazonS3ReadOnlyAccess** policy. Go to the **Next: Review**.



Create role

Attach permissions policies

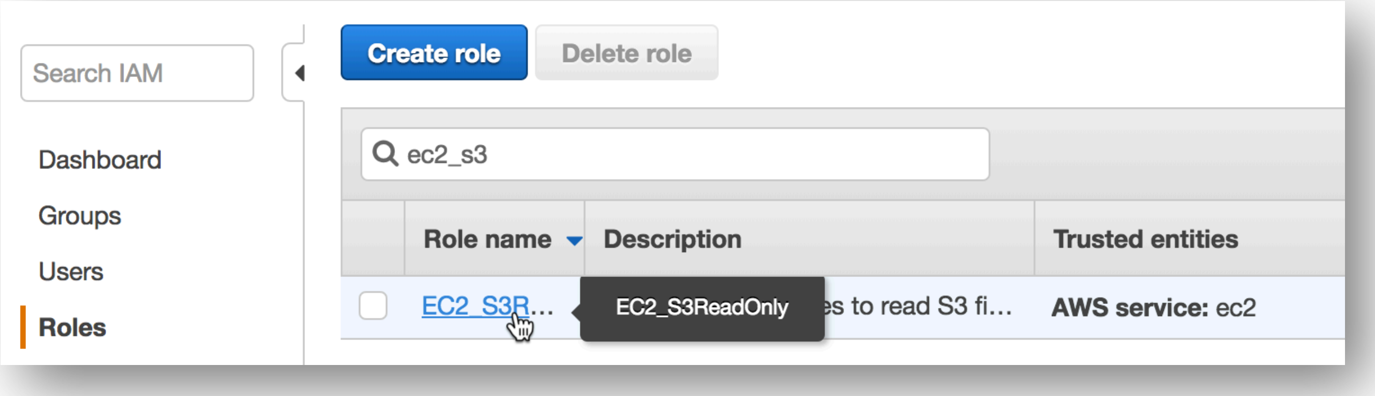
Choose one or more policies to attach to your new role.

Filter: Policy type | s3 | Showing 6 results

	Policy name	Attachments	Description
□	AmazonDMSRedshiftS3Role	1	Provides access to manage S3 settings for Redshift endpoin…
□	AmazonS3FullAccess	2	Provides full access to all buckets via the AWS Management…
☑	AmazonS3ReadOnlyAccess	0	Provides read only access to all buckets via the AWS Manag…
□	AWSGlueServiceRole-S3adpressions_small	1	This policy will be used for Glue Crawler and Job execution. …

4. Give your role a descriptive name, such as **EC2_S3ReadOnly** and edit the **role description** to be a helpful summary of what this role is. When you're done, **Create Role**.

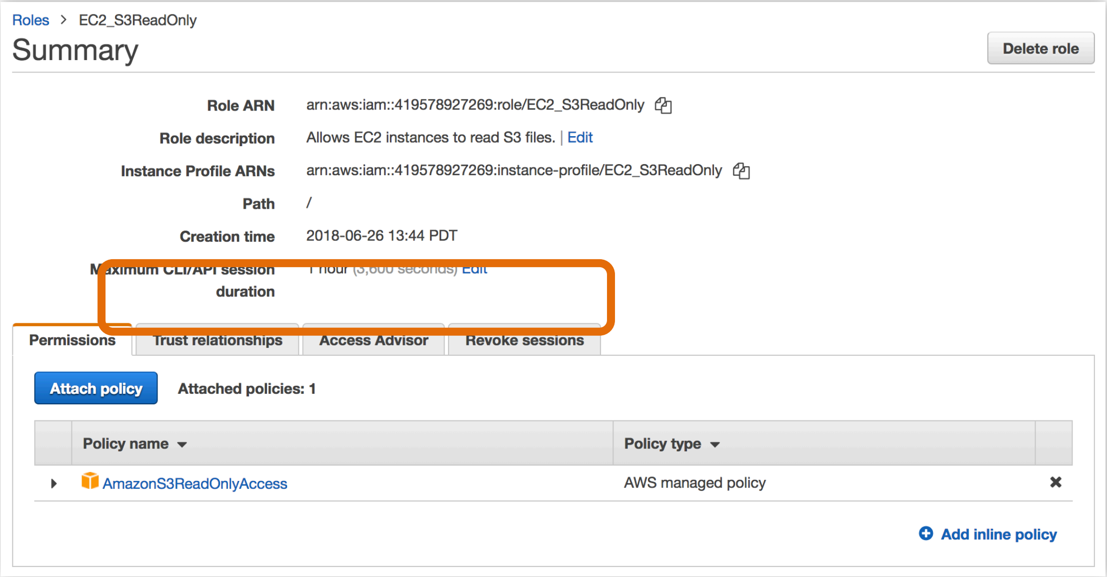
5. You are now back on the **Roles** page. Enter the name of the role you just created into the search bar and click on the role name.



ec2_s3

	Role name	Description	Trusted entities
□	EC2_S3R…	EC2_S3ReadOnly …s to read S3 fi…	**AWS service:** ec2

6. You are now on the **Summary** page of the role you just created. Here you can view and edit attributes of the role, such as how long the role's temporary credentials last. The default value as you can see below is 1 hour but can be as long as 12 hours.



Role ARN: arn:aws:iam::419578927269:role/EC2_S3ReadOnly

Role description: Allows EC2 instances to read S3 files. | Edit

Instance Profile ARNs: arn:aws:iam::419578927269:instance-profile/EC2_S3ReadOnly

Path: /

Creation time: 2018-06-26 13:44 PDT

Maximum CLI/API session duration: 1 hour (3,600 seconds) Edit

Attached policies: 1

Policy name	Policy type
AmazonS3ReadOnlyAccess	AWS managed policy

Awesome! You've just created an IAM role which will allow EC2 instances to read objects in S3.

Clean Up Lab Resources

If you want to clean up your account to get rid of everything we created during this lab, follow the instructions in this section. You can also leave your lab environment running if you want to test other AWS IAM concepts.

Go to **Groups** in the sidebar. Select the **Power_Users** group and go to the **Group Actions** dropdown. Select **Delete Group** and then **Yes, Delete**.

Go to **Users** in the sidebar. Select the user you created and select **Delete user**. Select the **checkbox** and then **Yes, delete**.

Go to **Roles** in the sidebar. Type in **EC2_S3ReadOnly** in the search bar. Select the role and then click on **Delete role**. Since we didn't actually use this role and there was no recent activity, the pop-up window will not ask for a confirmation. Click **Yes, delete**.

Additional Resources

IAM Introduction: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html
IAM Best Practices: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
IAM Policies: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html
IAM Tutorials: https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorials.html