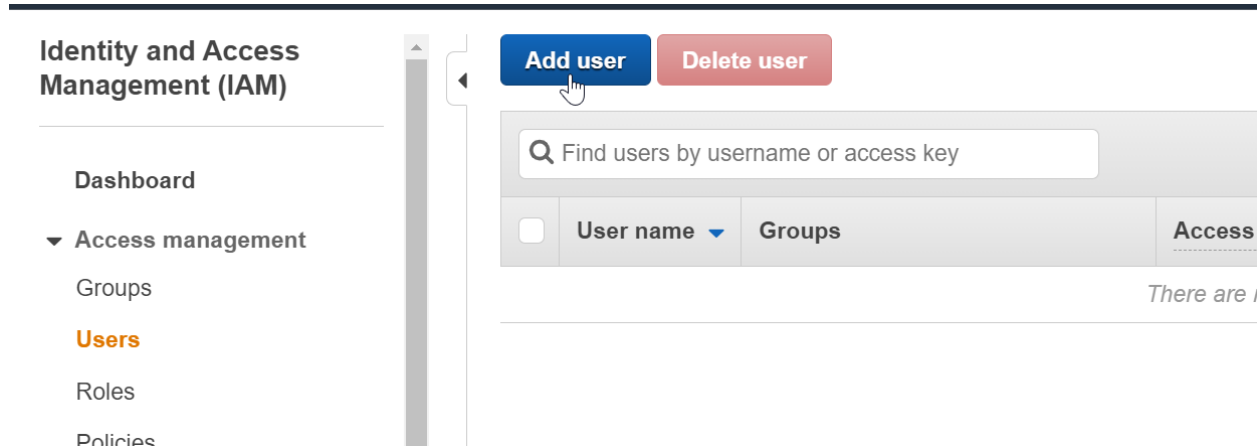


Identity Access Management (IAM) Lab

****Recommended – Add Multi factor authentication to your account before starting****

Step 1: Creating User Accounts

- 1.) Go to the IAM Dashboard
- 2.) Click: **Users**
- 3.) Click: **Add user**



- 4.) User name: **Jade** (or any name you want)
- 5.) Check: **Programmatic access**
- 6.) Check: **AWS Management Console access**
- 7.) Console password: **Autogenerated Password**
- 8.) Required password reset: **User must create a new password at next sign-in**

Set user details

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

User name* Jade

[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

* Required

[Cancel](#)


[Next: Permissions](#)


Step 2: Creating groups


- 1.) In the add user section click: **Create group**

Add user

▼ Set permissions

 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.

Add user to group

Create group Refresh

Group ▼	Attached policies
---------	-------------------






- 2.) Group name: **Developers**
- 3.) Add: **AdministratorAccess – Job function**
- 4.) Click: Create group

Group name

Create policy Refresh

Filter policies ▼

Showing 646 results

	Policy name ▼	Type	Used as	Description
<input checked="" type="checkbox"/>	 AdministratorAccess	Job function	Permissions policy (3)	Provides full access to AWS services and resources.
<input type="checkbox"/>	 AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissions while explicitly allowing direct acces...
<input type="checkbox"/>	 AdministratorAccess-AWSElasticB...	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and ad...
<input type="checkbox"/>	 AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/>	 AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to related AWS S...

Cancel Create group

Step 3: Creating user account (Cont.)

1.) Add Tags: **Do not add**

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	<input type="button" value="Remove"/>

You can add 50 more tags.

[Cancel](#)[Previous](#)[Next: Review](#)

2.) Review

3.) Click: **Create user**

Add user

1 2 3 **4** 5

Review

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

User details

User name	Jade
AWS access type	Programmatic access and AWS Management Console access
Console password type	Autogenerated
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

[Cancel](#)[Previous](#)[Create user](#)

***WARNING – Never allow anyone access to use/see your Access key ID and Secret Access key ID ***

Step 4: Programmatic and Secret access keys

1.) Click: **Download .csv**

- Downloading the .csv file saves the below information in a spreadsheet
- Click the Show button to show the Secret access key and password.
- Access key id - username to programatically access the aws console
- secret access key - programatically access the AWS infrastructure)
- Send e-mail sends the users the information to log into the console.



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://npoweraws1.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key	Password	Email login instructions
▶	✓ Jade	AKIARB4IONDYVAKPLF47	***** Show	***** Show	Send email

Close

2.) Open the CSV file and you should see your users information.

A	B	C	D	E
User name	Password	Access key ID	Secret access key	Console login link
Jade	9kX{F8'f5}8X+6E	AKIARB4IONDYVAKPLF47	PhsCcrLMFDt18jNA8zLrAf4FSD2EzoUv0R4gWg4l	https://npoweraws1.signin.aws.amazon.com/console

3.) Click: **Close**

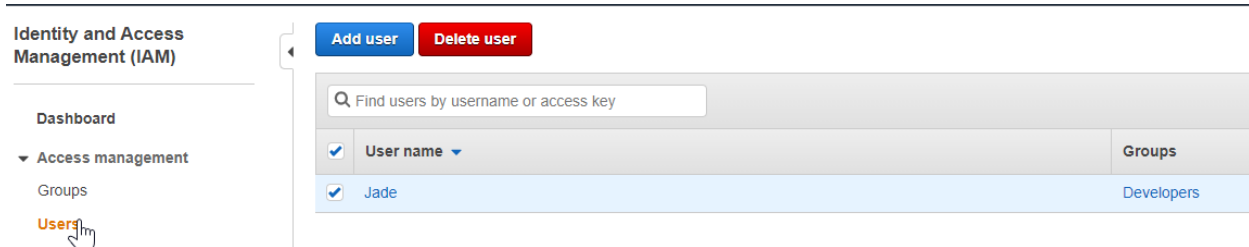
Download .csv

	User	Access key ID	Secret access key	Password	Email login instructions
▶	✓ Jade	AKIARB4IONDYVAKPLF47	***** Show	***** Show	Send email

Close

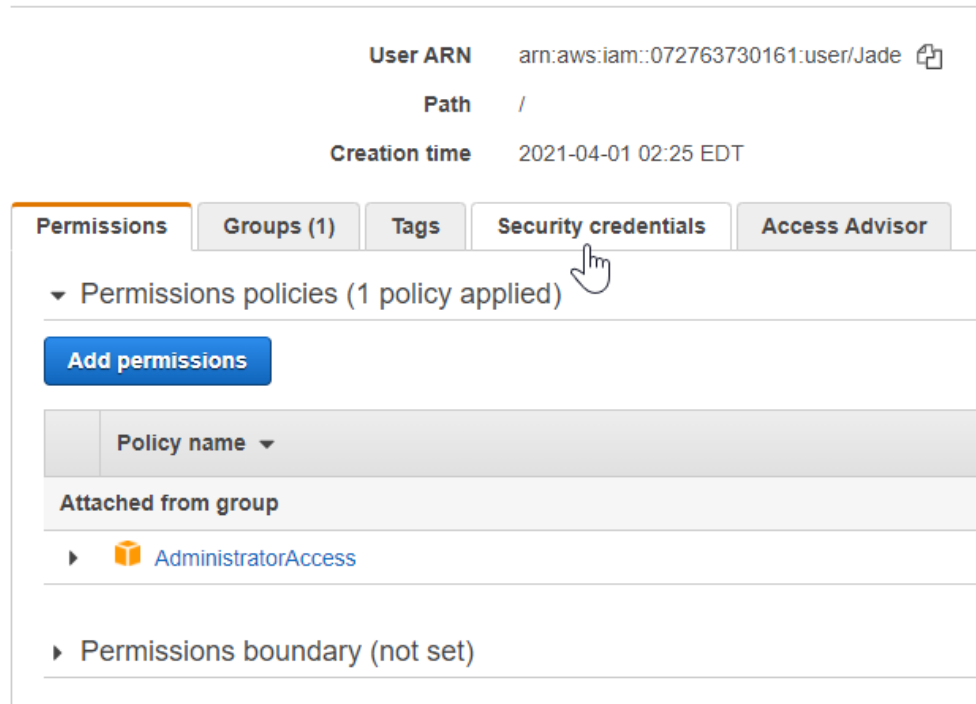
Step 5: If the user information is lost

- 1.) In IAM: Click users
- 2.) Click: **the user (Jade)**



- 3.) Click: **Security Credentials tab**

Summary



4.) Console password: **Manage**

Path /

Creation time 2021-04-01 02:25 EDT

Permissions Groups (1) Tags **Security credentials** Access Advisor

Sign-in credentials

Summary	• Console sign-in link: https://npoweraws1.signin.aws.amazon.com/console
Console password	Enabled (never signed in) Manage
Assigned MFA device	Not assigned Manage
Signing certificates	None

Access keys

5.) You can change the users set password or console access

6.) **Do not change anything and press cancel**

Manage console access

Manage Jade's AWS console access and password.

Console access ☒ Enable
☐ Disable
Disabling will remove pre-existing password.

Set password* ☐ Keep existing password
☐ Autogenerated password
☒ Custom password

☐ Show password

Require password reset ☐ User must create a new password at next sign-in

Cancel Apply

7.) Access keys

- Anyone that has the access key and secret access key Id, could use your aws account
- You can choose to make an access key inactive as well
- Change nothing

Access keys

Use access keys to make programmatic calls to AWS from the AWS CLI, 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.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. [Learn more](#)

Create access key

Access key ID	Created	Last used	Status
AKIARBAIONDYVAKPLF47	2021-04-01 02:25 EDT	N/A	Active Make inactive ✕

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. [Learn more](#)

Upload SSH public key

Step 6: Understanding JSON

1.) From IAM Dashboard: **Click Policies**

2.) Click: **AdministratorAccess** arrow

- Look at the javascript object notation (JSON)
- Policies are written in JSON
- Effects: allow

Action - *

Resource - *,

*(Star) is a wildcard allowing anything to do anything with any resources in aws.

Identity and Access Management (IAM)

Dashboard

Access management

- Groups
- Users
- Roles
- Policies**
- Identity providers
- Account settings

Access reports

- Access analyzer
 - Archive rules
 - Analizers
 - Settings
- Credential report
- Organization activity

Create policy Policy actions

Filter policies Search

	Policy name	Type	Used as	Description
<input type="radio"/>	AccessAnalyzerServiceRolePolicy	AWS managed	None	Allow Access Analyzer to analyze resource metadata
<input checked="" type="radio"/>	AdministratorAccess	Job function	Permissions policy (4)	Provides full access to AWS services and resources.

AdministratorAccess

Provides full access to AWS services and resources.

Policy summary {} JSON

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "*",
7       "Resource": "*"
8     }
9   ]
10 }
```

Step 7: Creating Roles

- 1.) From IAM Dashboard Click: **Roles**
- 2.) Click: **Create role**

Identity and Access Management (IAM)

Dashboard

▼ Access management

Groups

Users

Roles

Create role

Delete role

Search

Role name ▼

☐ [AWSServiceRoleForSupport](#)

- 3.) Click: **EC2**

Create role

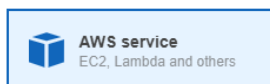
1

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.

4.) Click: **AmazonS3FullAccess**

Create role 1

▼ Attach permissions policies

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

Create policy

Filter policies ▼

	Policy name ▼	Used as
<input type="checkbox"/>	▶ AmazonDMSRedshiftS3Role	None
<input checked="" type="checkbox"/>	▶ AmazonS3FullAccess	None

5.) Tags: **Leave Empty**

6.) Review - Role name: **S3_Admin_Access**

7.) Click: **Create Role**

Create role 1 2 3 4

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 AmazonS3FullAccess [↗](#)

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

Cancel

Previous

Create role

8.) Now our EC2 has full access to S3 resources.

Roles > S3_Admin_Access

Summary Delete role

Role ARN	arn:aws:iam::072763730161:role/S3_Admin_Access
Role description	Allows EC2 instances to call AWS services on your behalf. Edit
Instance Profile ARNs	arn:aws:iam::072763730161:instance-profile/S3_Admin_Access
Path	/
Creation time	2021-04-01 03:32 EDT
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions Trust relationships Tags Access Advisor Revoke sessions

▼ Permissions policies (1 policy applied)

[Attach policies](#) [Add inline policy](#)

Policy name ▼	Policy type ▼	
▶ AmazonS3FullAccess	AWS managed policy	

▶ Permissions boundary (not set)

Congratulations on completing the LAB!!