Name: Saurabh. Chaudhari.

Roll no: A005

PAGE NO.

DATE:

Cloud Computing - Practical 3

Q1]. Users and Groups :-

USERS: In Cloud Computing users are individuals entities that require access to cloud resources and services. Users can be human individual or non human entities (such as application or services). Each user has a unique identity and is authenticated through credential like usernames, passwords or certificates. Users are assigned permission that define what actions they can perform with the cloud environment.

Key aspects of Users :

1) Idendity Management - Ensures each user has a unique identity

2) Authentication - Verifies users identity through methods like passwords or multilayer/factor authentication.

3) Authorisation - Determines what action a user case perform based on permission.

4) Types - End Users, Services accounts, administrator and external users.

GROUPS :

→ Groups is cloud computing are collection of users who shares similar roles or access needs.
Groups simplify permission management by allowing administrator to assign permission and policies collectively rather than individually.
This approach 8 is managing individual user permission can be complex.

Key Aspects of groups :-

(a) Role Based Access Control (RBAC) : Assigns permission based on roles to enhance security and reduce administrative tasks

(b) Policy Enforcement : Ensure consistent application of security policies across all groups members.

(c) Stalability : Facilitate management of permissions for large number of users

(d) Types : Security groups, service groups and user groups q2] Identity and Access Management (IAM).

→ Identity and access management (IAM) ensures that the right people and job roles in your organization (identities) can access the tool they need to do their jobs.

→ Companies need IAM to provide online security and to increase employee productivity.

→ The has been a burst in market with new application and requirement for an organization to use these application has increased drastically.

→ The services and resources you want to access can be specified in IAM.

→ IAM doesn't provide any replica or backup.
IAM can be used for many purpose such as if one wants to control access of individual & groups access for your AWS Resources.

→ With IAM policies managing permission to your workforce and system to ensure least-priviledge permission become eaiser.
The AWS IAM is a global services.

→ Components of identity and Access Management (IAM)
1. Roles
2. Groups
3. Policies.

→ IAM identies classified as
1. IAM Users
2. IAM Groups
3. IAM Roles

PAGE NO.
DATE :

3] IAM Roles :

→ A Role is set of permission that great access to action and reomas in Aus. These permission are attached to the role not to an IAM Users or a group.

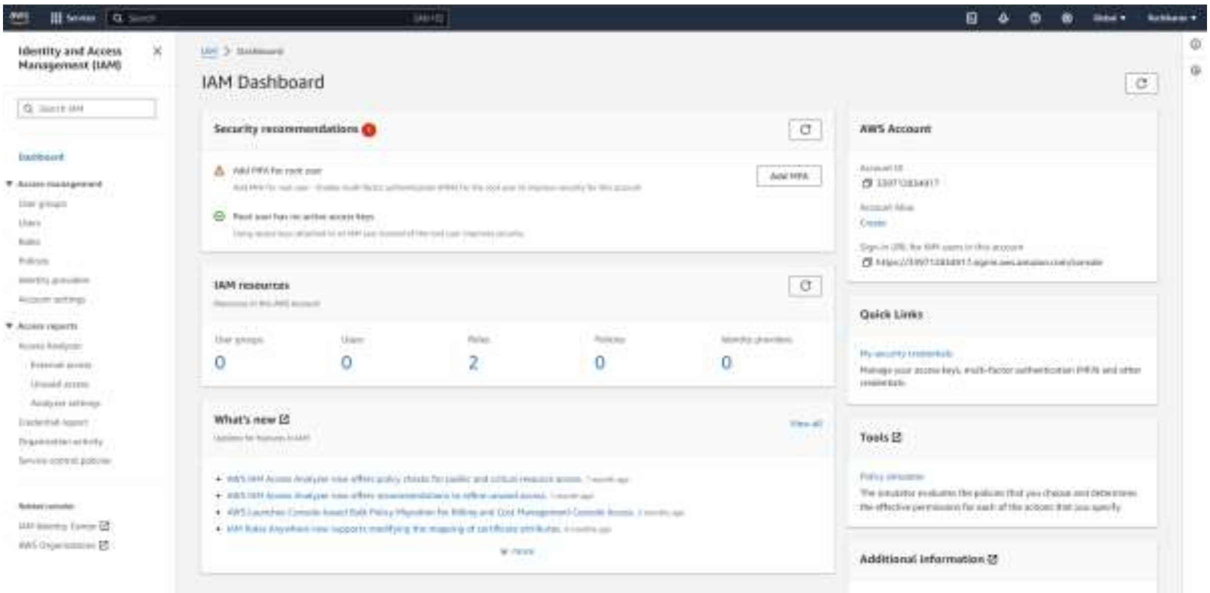
→ An IAM user can use a role in the same Aus account or diff account.

→ An IAM User is simiar to as IAM User role is also an Aus Identity with permissio policies that determis what the identity can & cannot do in Aus

→ A role does not have long term security Credential password or secrity key Insted if the user uses a role tempurarly security credential are created & provide the user.

You can use the roles to delegate acces to users application or services that generaly do not have access to your Aus account resource.

IAM User CC - A3


Identity and Access Management (IAM)

IAM Dashboard

Security recommendations

Add MFA for root user

Root user has no active access keys

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	0	2	0	0

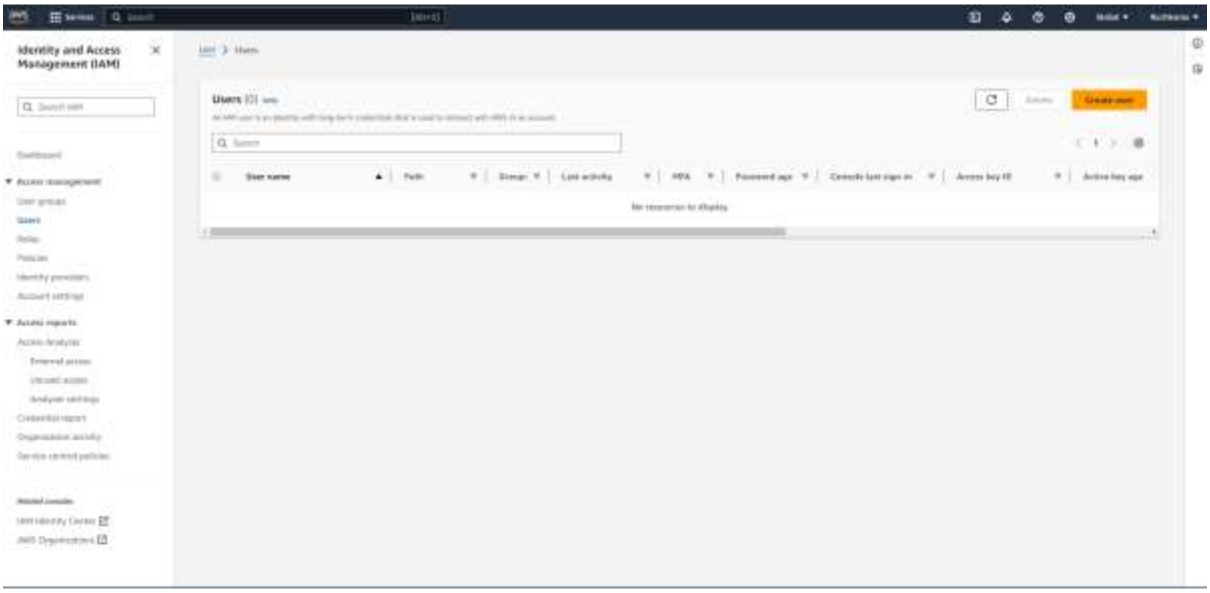
What's new

AWS Account

Quick Links

Tools

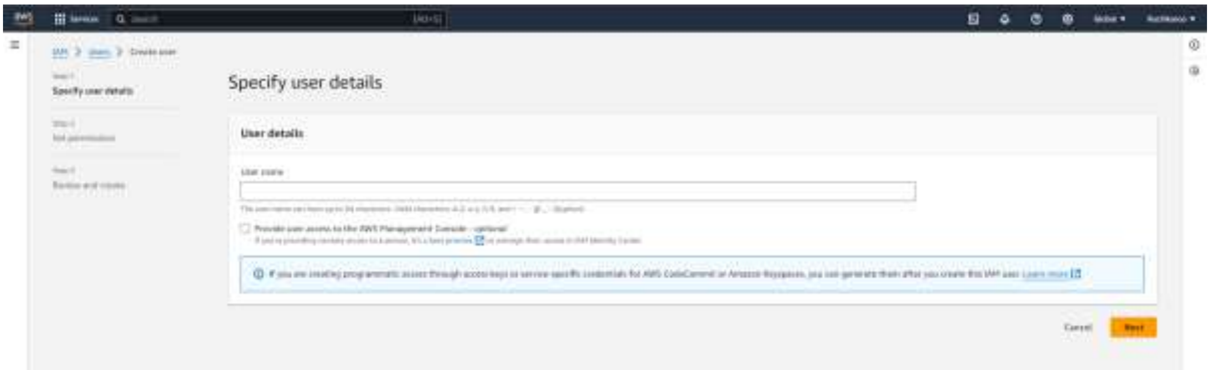
Additional information


Identity and Access Management (IAM)

Users (0)

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age

No resources to display

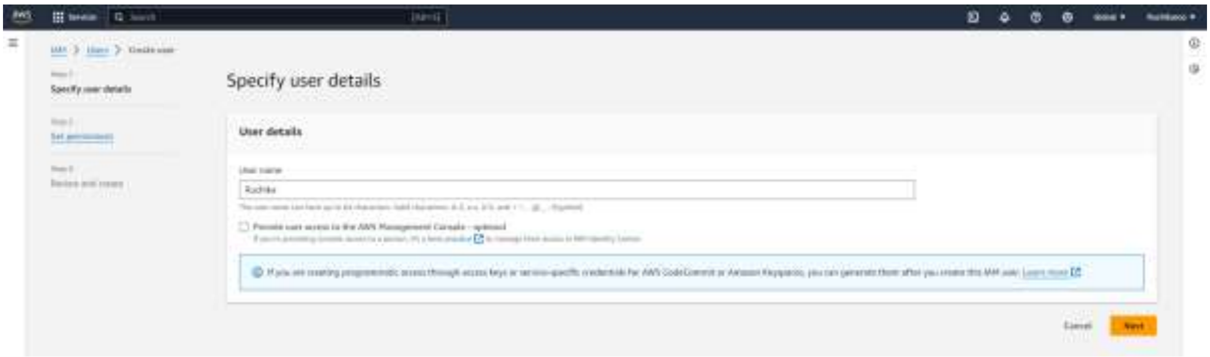

Specify user details

User details

User name

□ Provide user access to the AWS Management Console - optional

Cancel Next



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

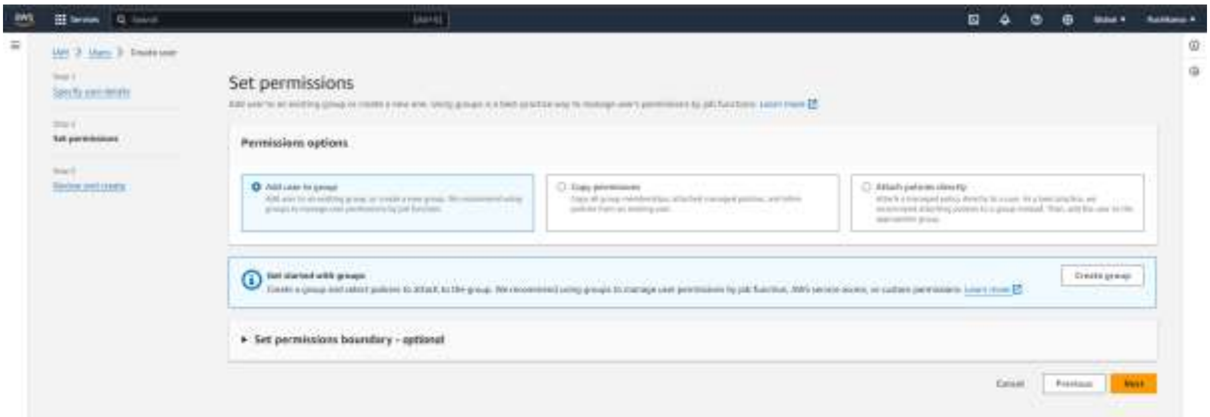
User details

User name
Ruchika

□ Provide user access to the AWS Management Console - optional

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



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Set permissions

Permissions options

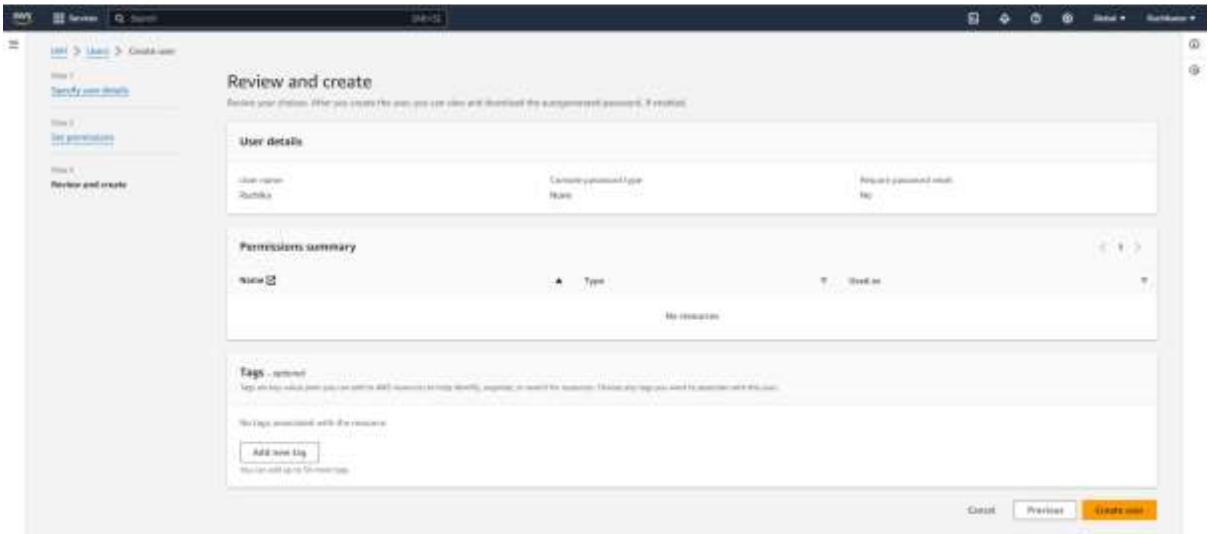
● Add user to group
○ Copy permissions
○ Attach policies directly

Get started with groups

Create group

▸ Set permissions boundary - optional

Cancel Previous Next



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Review and create

User details

User name	Console password type	Require password reset
Ruchika	None	No

Permissions summary

Name	Type	Used as

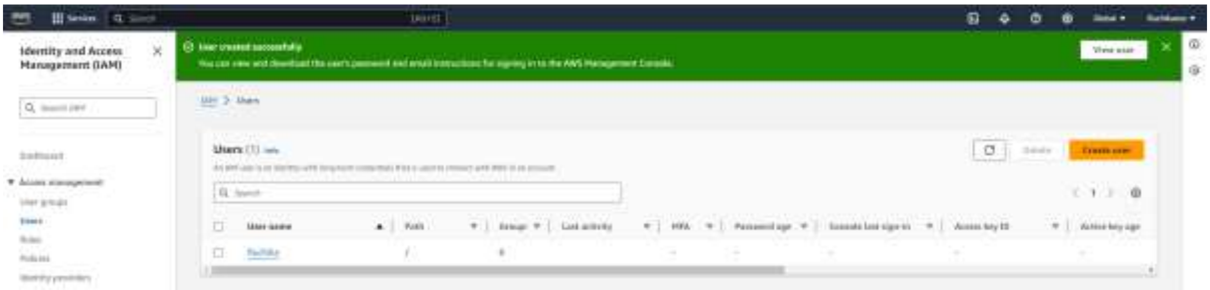
No resources

Tags - optional

No tags associated with the resource.

Add new tag

Cancel Previous Create user



Identity and Access Management (IAM)

User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

View user

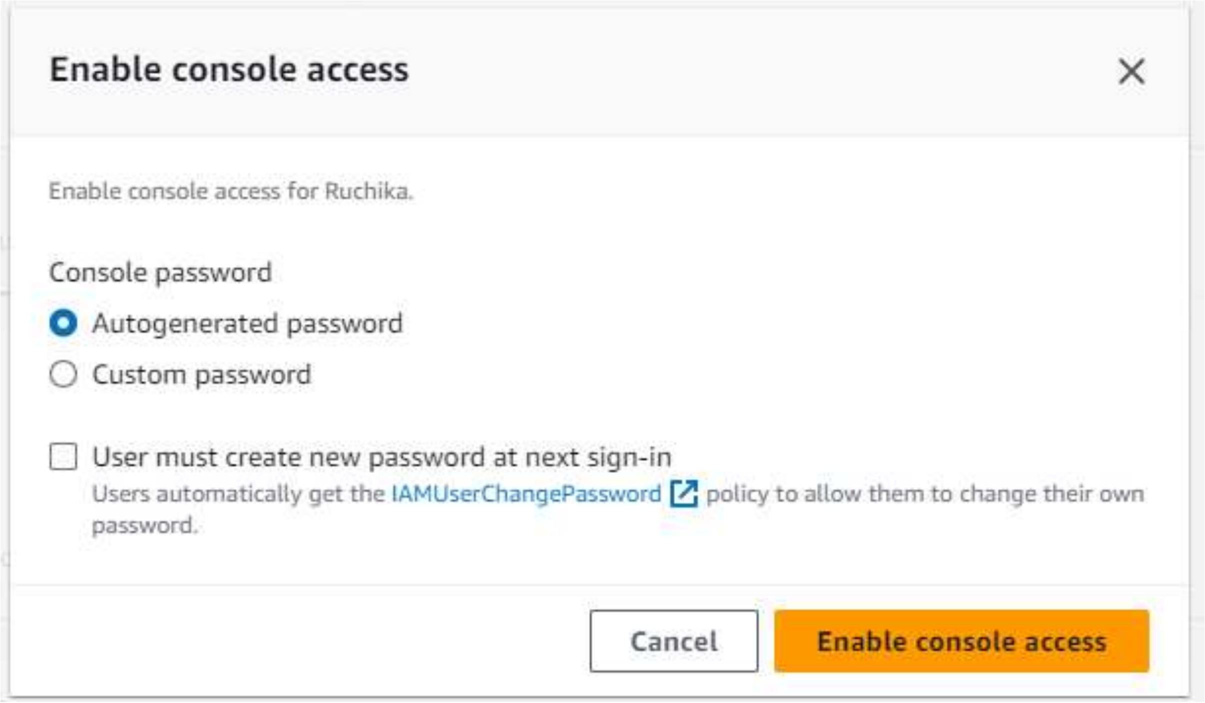
IAM > Users

Dashboard
▾ Access management
User groups
Users
Roles
Policies
Identity providers

Users (1)

Delete Create user

□	User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age
□	Ruchika	/	0	-	-	-	-	-	-



Enable console access

Enable console access for Ruchika.

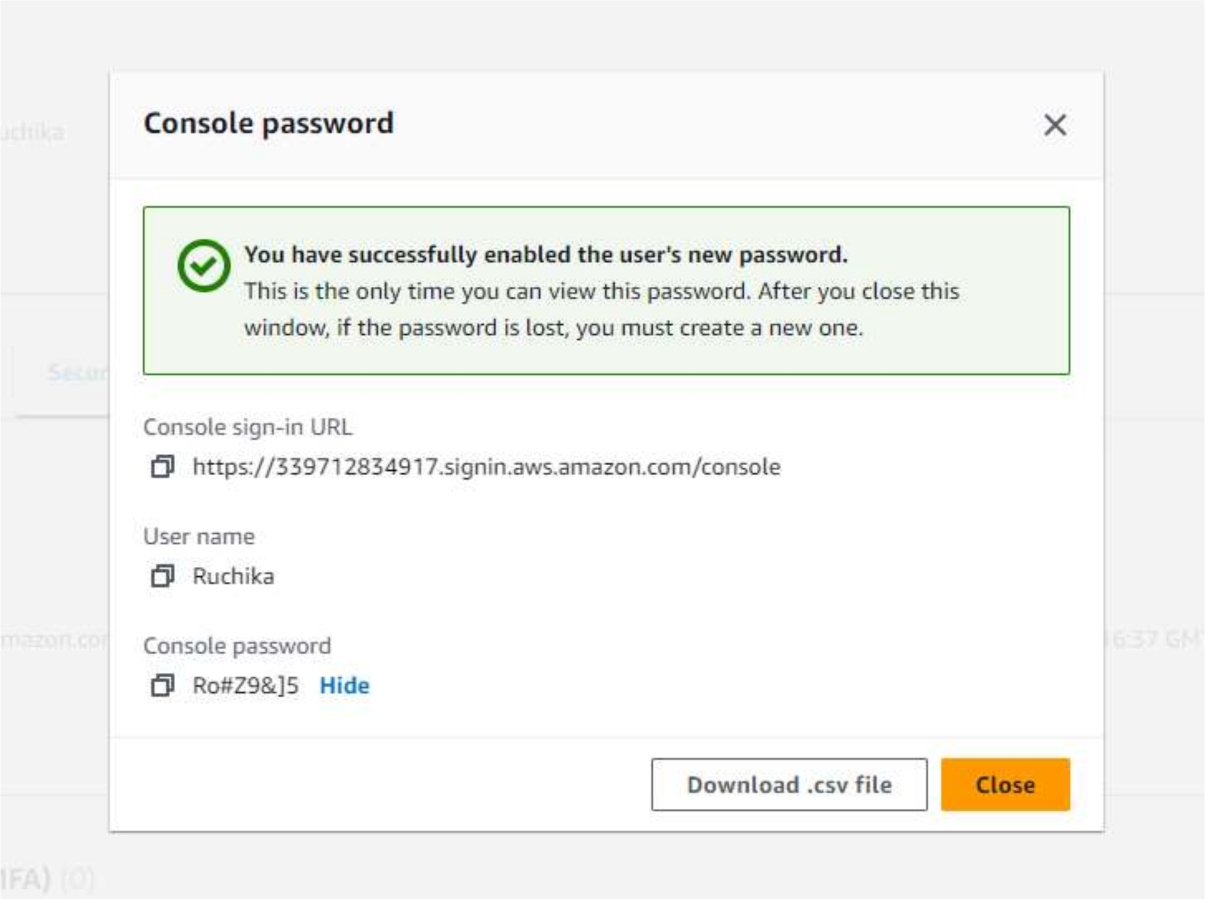
Console password

● Autogenerated password

○ Custom password

□ User must create new password at next sign-in

Users automatically get the IAMUserChangePassword policy to allow them to change their own password.



Console password

You have successfully enabled the user's new password.
This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.

Console sign-in URL

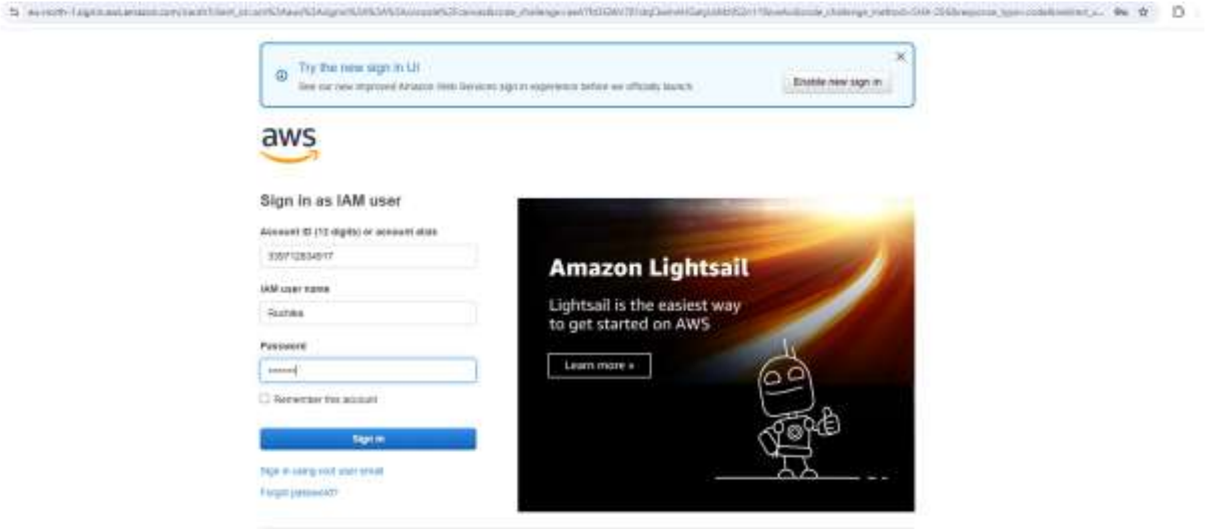
https://339712834917.signin.aws.amazon.com/console

User name

Ruchika

Console password

Ro#Z9&]5 Hide



Try the new sign in UI

See our new improved Amazon Web Services sign in experience before we officially launch

Enable new sign in

Sign in as IAM user

Account ID (12 digits) or account alias: 339712834917

IAM user name: Ruchika

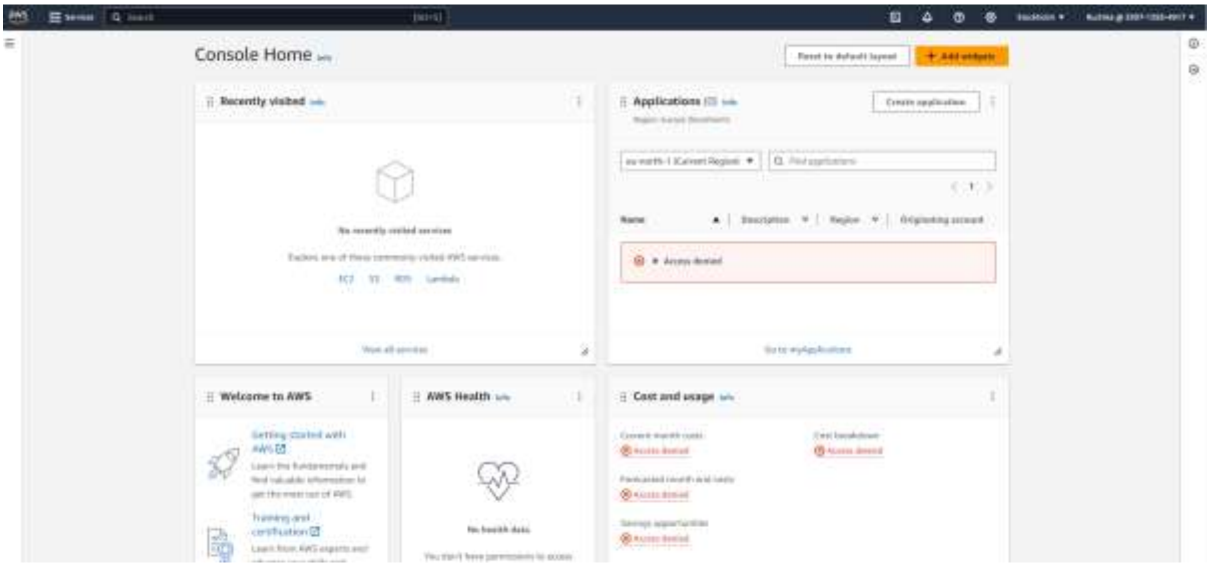
Password: ••••••

☐ Remember this account

Sign in

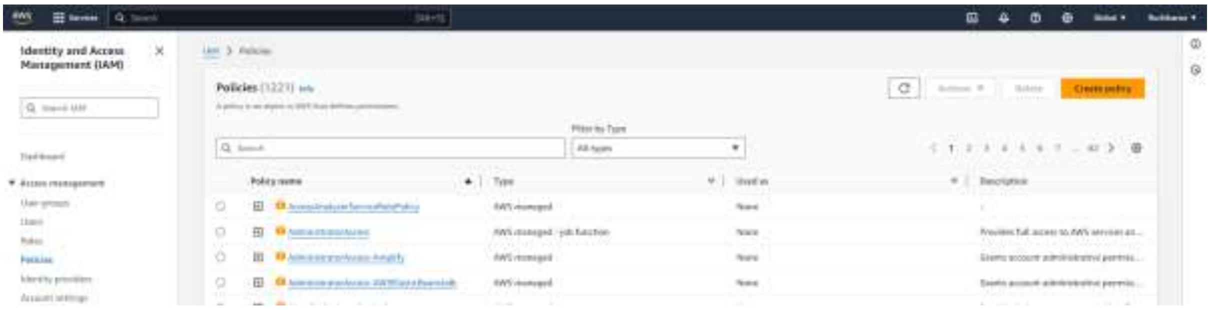
Sign in using root user email

Forgot password?



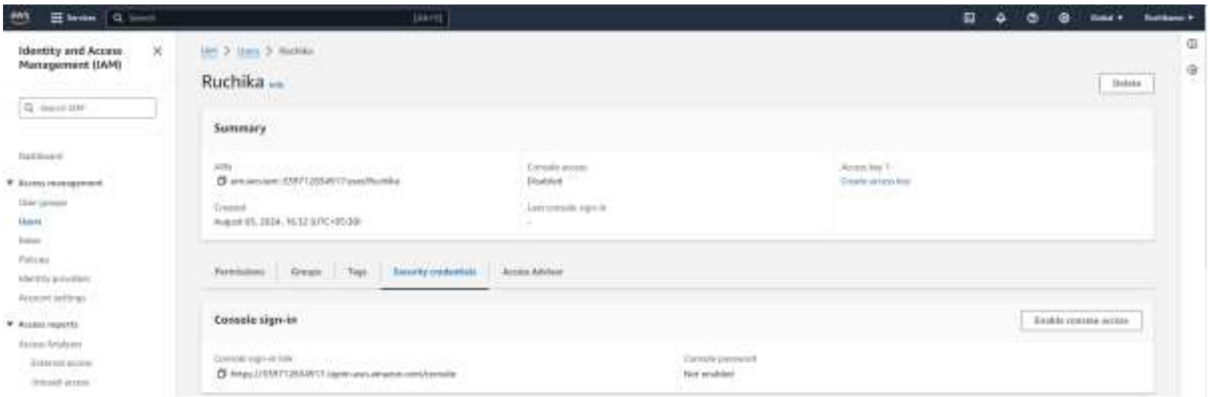
Console Home

Access denied



Policies (1221)

	Policy name	Type	Used as	Description
○	AccessAnalyzerServiceRolePolicy	AWS managed	None	-
○	AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
○	AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permis...
○	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants account administrative permis...



Ruchika

Summary

ARN: arn:aws:iam::339712834917:user/Ruchika

Console access: Disabled

Access key 1: Create access key

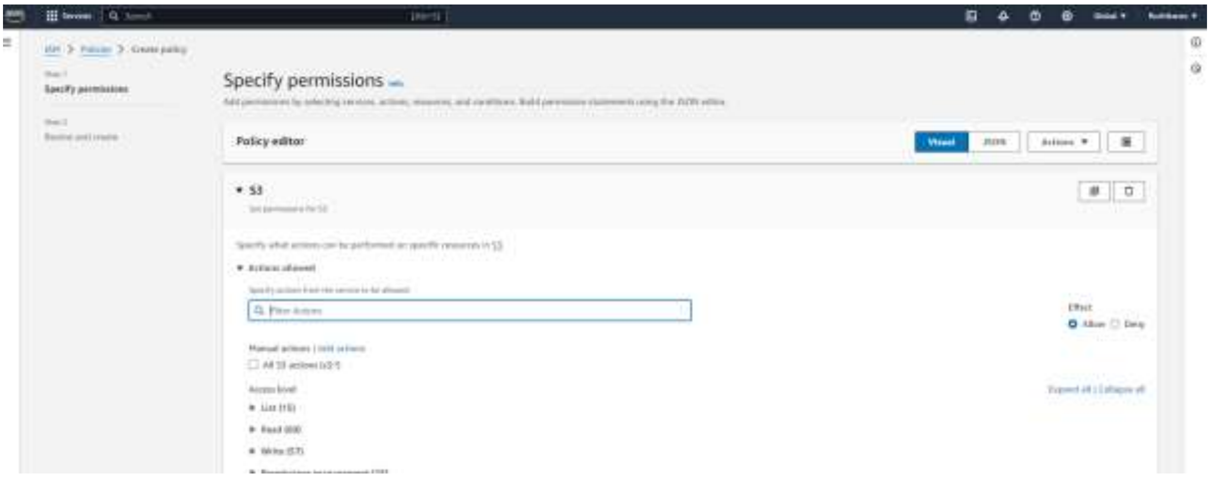
Created: August 05, 2024, 16:12 (UTC+05:30)

Last console sign-in: -

Console sign-in

Console sign-in link: https://339712834917.signin.aws.amazon.com/console

Console password: Not enabled



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

▼ **S3**

Set permissions for S3

Specify what actions can be performed on specific resources in S3.

▼ **Actions allowed**

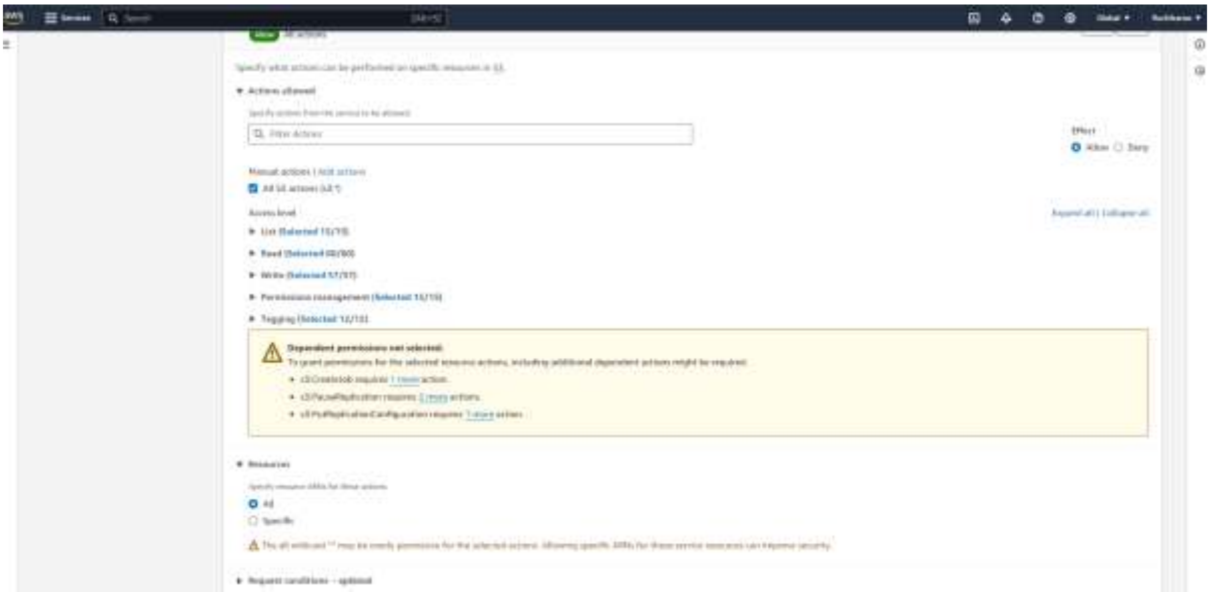
Specify actions from the service to be allowed.

Filter Actions

Effect
● Allow ○ Deny

Manual actions | Add actions
□ All S3 actions (s3:*)

Access level
▶ List (15)
▶ Read (60)
▶ Write (57)



Allow All actions

Specify what actions can be performed on specific resources in S3.

▼ **Actions allowed**

Specify actions from the service to be allowed.

Filter Actions

Effect
● Allow ○ Deny

Manual actions | Add actions
☑ All S3 actions (s3:*)

Access level
▶ List (Selected 15/15)
▶ Read (Selected 60/60)
▶ Write (Selected 57/57)
▶ Permissions management (Selected 15/15)
▶ Tagging (Selected 12/12)

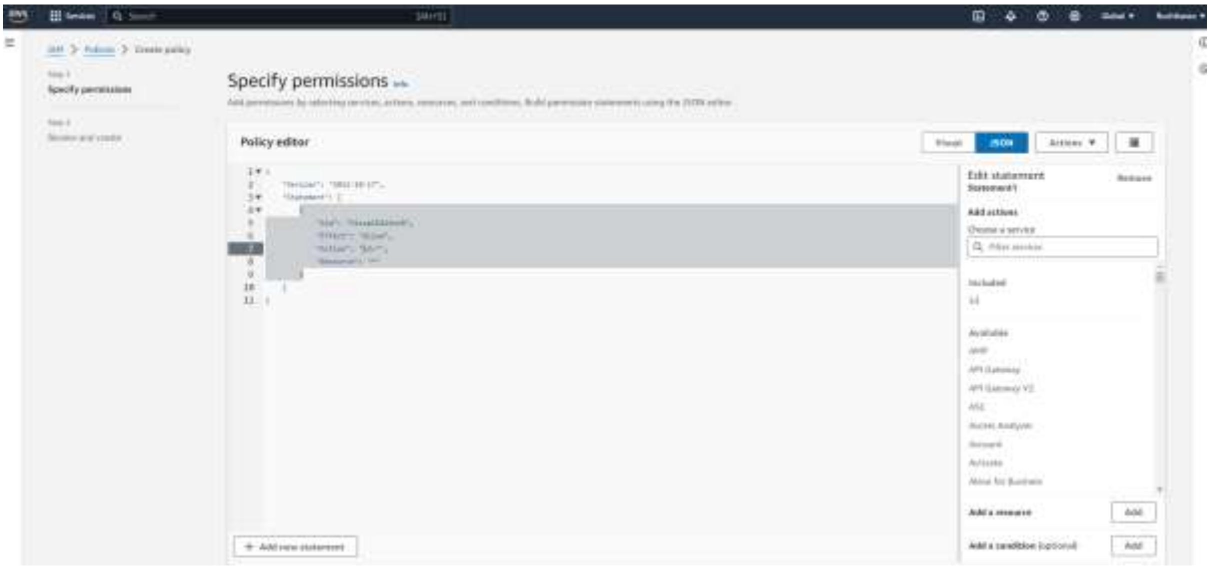
Dependent permissions not selected
To grant permissions for the selected resource actions, including additional dependent actions might be required.
- s3:CreateJob requires 1 more action.
- s3:PauseReplication requires 2 more actions.
- s3:PutReplicationConfiguration requires 1 more action.

▼ **Resources**

Specify resource ARNs for these actions.
● All
○ Specific

⚠ The all wildcard '*' may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.

▶ **Request conditions** - optional



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        }
    ]
}
```

Edit statement
Statement1

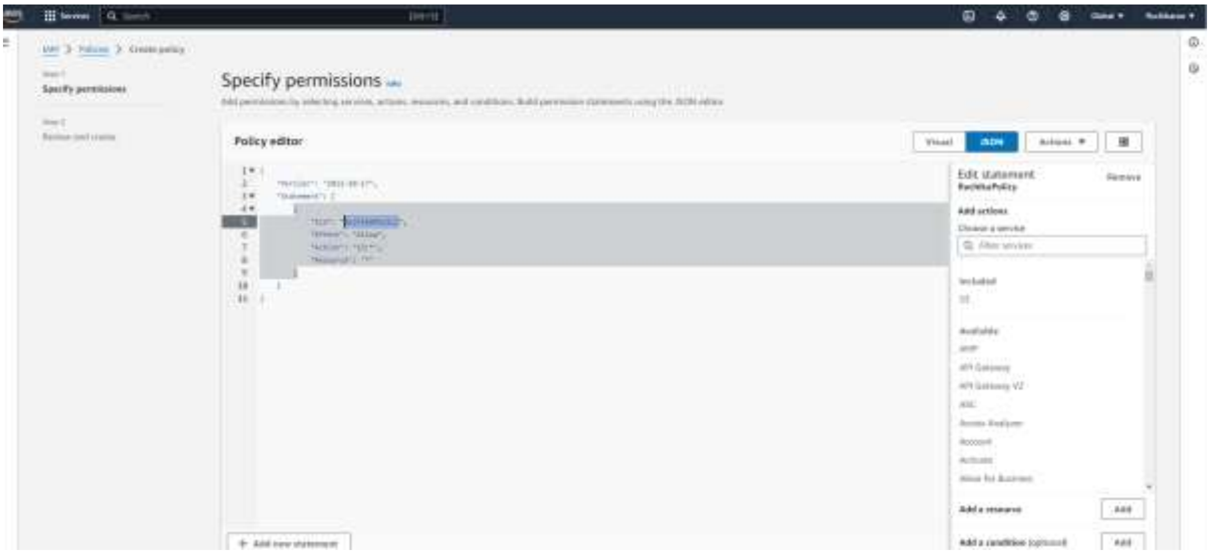
Add actions
Choose a service
Filter services

Included
S3

Available
AMP
API Gateway
API Gateway V2
ARC
Access Analyzer
Account
Activate
Alexa for Business

Add a resource

Add a condition (optional)



IAM > Policies > Create policy

Step 1
Specify permissions

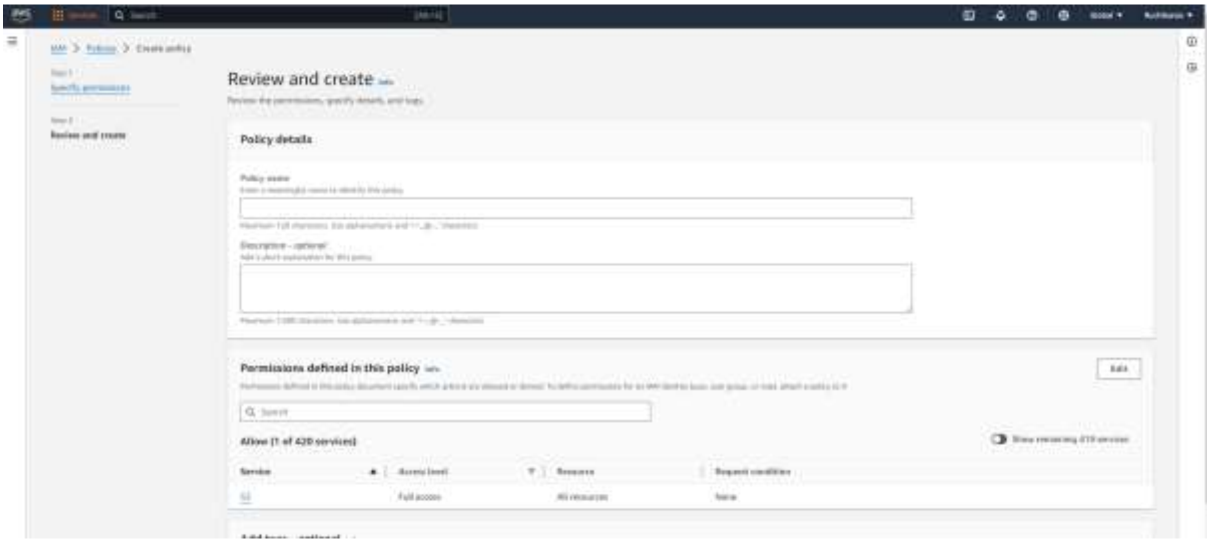
Step 2
Review and create

Specify permissions

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "[illegible]",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        }
    ]
}
```



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Description - optional
Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.

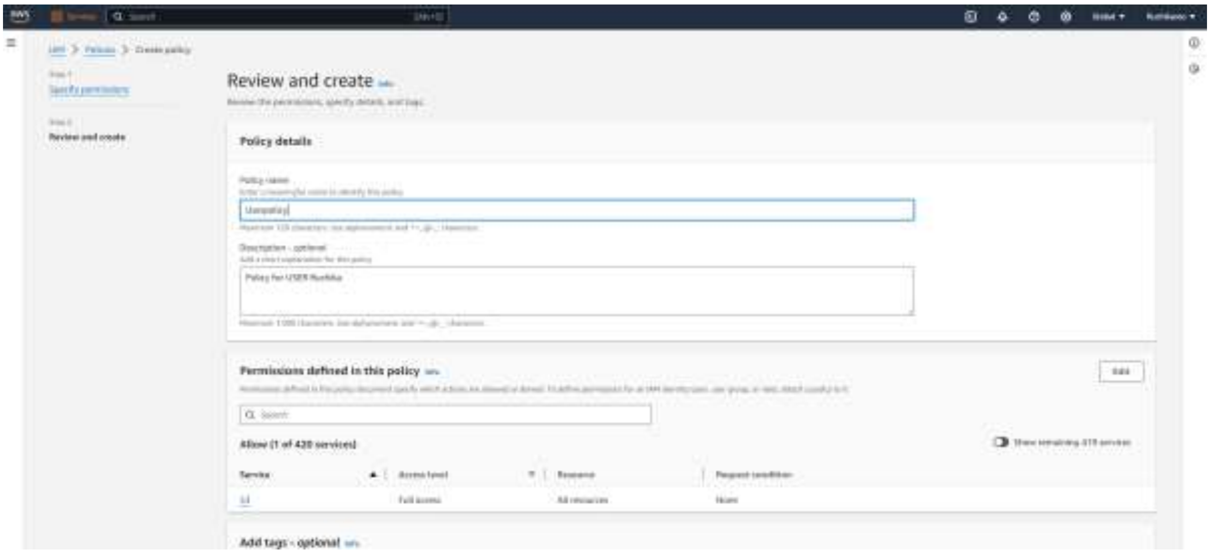
Permissions defined in this policy

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 420 services)

Service	Access level	Resource	Request condition
S3	Full access	All resources	None

Add tags - optional



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
Userpolicy

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Description - optional
Add a short explanation for this policy.
Policy for USER Ruchika

Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.

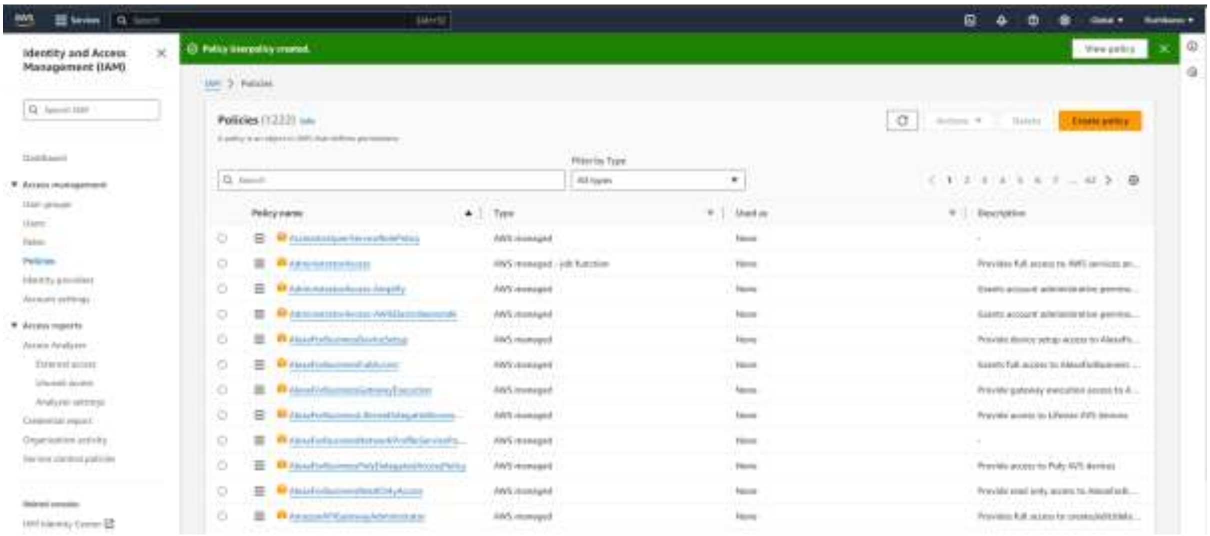
Permissions defined in this policy

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 420 services)

Service	Access level	Resource	Request condition
S3	Full access	All resources	None

Add tags - optional



Policy Userpolicy created.

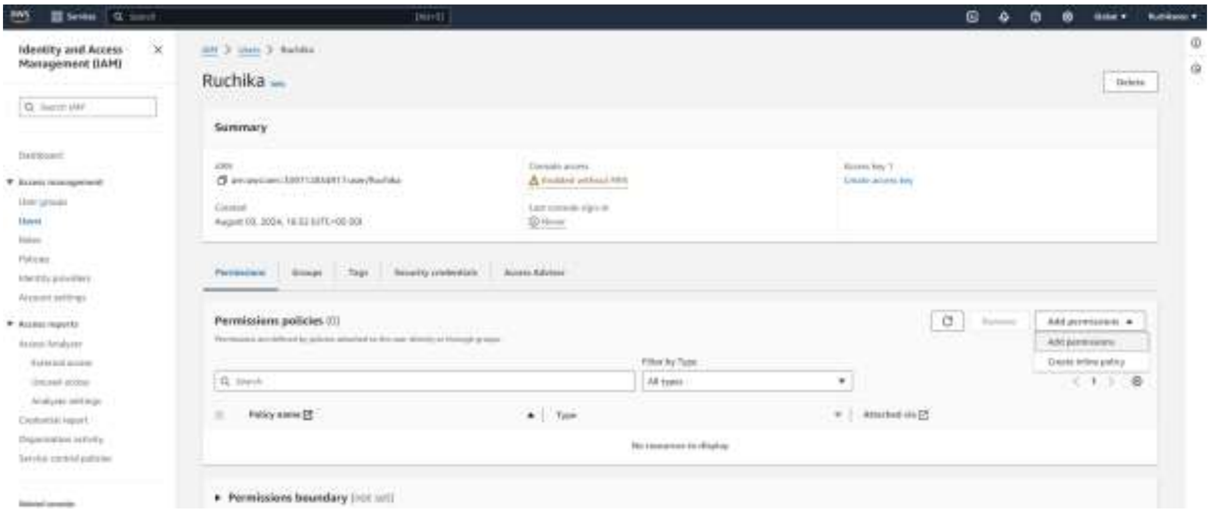
IAM > Policies

Policies (1222)

A policy is an object in AWS that defines permissions.

Filter by Type: All types

Policy name	Type	Used as	Description
AccessAnalyzerServiceRolePolicy	AWS managed	None	-
AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permis...
AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants account administrative permis...
AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaFo...
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness ...
AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution access to A...
AlexaForBusinessLifesizeDelegatedAccess...	AWS managed	None	Provide access to Lifesize AVS devices
AlexaForBusinessNetworkProfileServicePo...	AWS managed	None	-
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None	Provide access to Poly AVS devices
AlexaForBusinessReadOnlyAccess	AWS managed	None	Provide read only access to AlexaForB...
AmazonAPIGatewayAdministrator	AWS managed	None	Provides full access to create/edit/dele...



IAM > Users > Ruchika

Ruchika

Summary

ARN: arn:aws:iam::[illegible]:user/Ruchika

Created: August 03, 2024, 16:32 (UTC+05:30)

Console access: Enabled without MFA

Last console sign-in: Never

Access key 1: Create access key

Permissions policies (0)

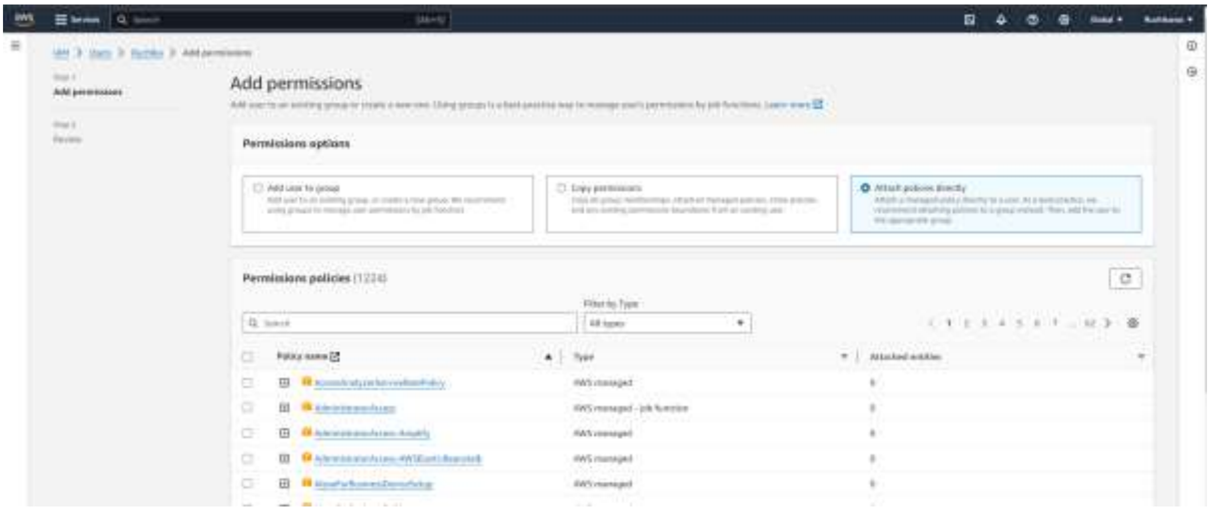
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via

No resources to display

Permissions boundary (not set)



IAM > Users > Ruchika > Add permissions

Step 1
Add permissions

Step 2
Review

Add 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.

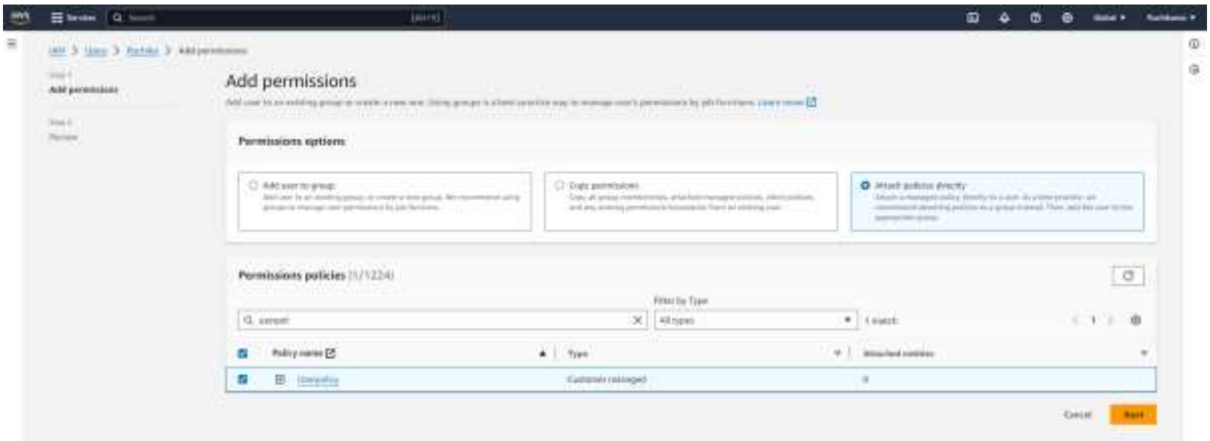
○ Copy permissions
Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

● 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 (1224)

Filter by Type: All types

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AdministratorAccess	AWS managed - job function	0
AdministratorAccess-Amplify	AWS managed	0
AdministratorAccess-AWSElasticBeanstalk	AWS managed	0
AlexaForBusinessDeviceSetup	AWS managed	0



IAM > Users > Ruchika > Add permissions

Step 1
Add permissions

Step 2
Review

Add permissions

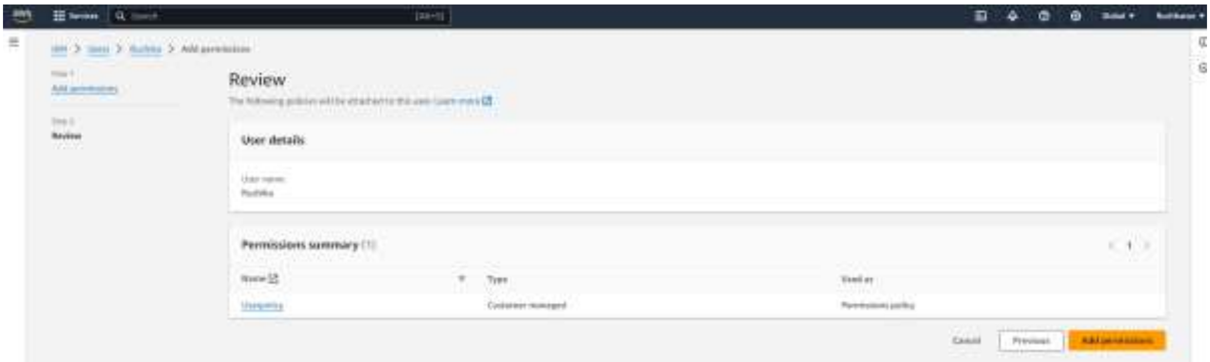
Permissions options

- ○ Add user to group
- ○ Copy permissions
- ● Attach policies directly

Permissions policies (1/1224)

Filter by Type: All types — 1 match

☑	Policy name	Type	Attached entities
☑	Userpolicy	Customer managed	0



IAM > Users > Ruchika > Add permissions

Step 1
Add permissions

Step 2
Review

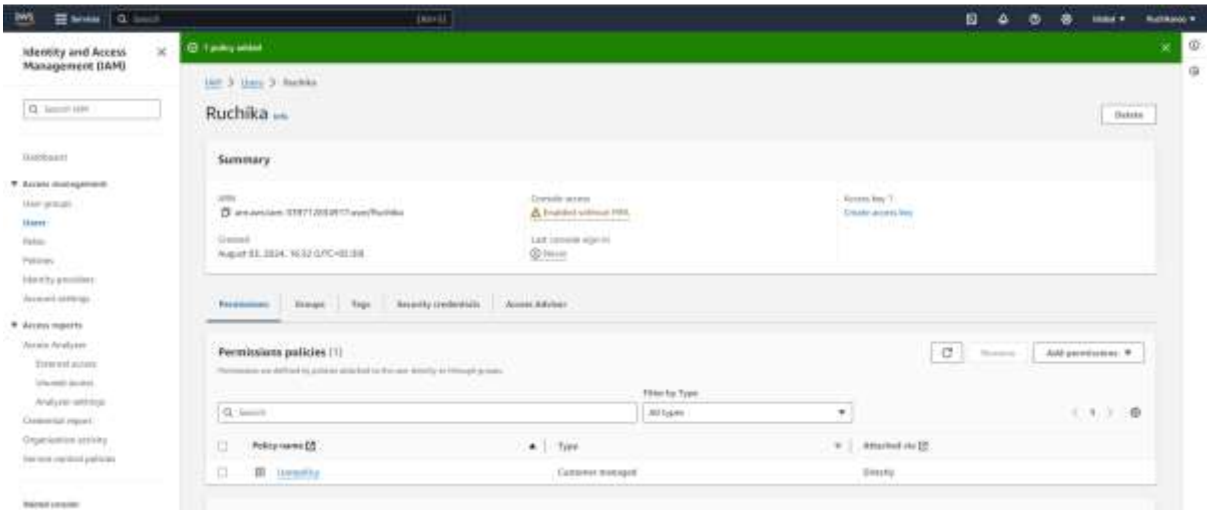
Review

User details

User name
Ruchika

Permissions summary (1)

Name	Type	Used as
Userpolicy	Customer managed	Permissions policy



1 policy added

IAM > Users > Ruchika

Ruchika

Summary

Created
August 03, 2024, 16:32 (UTC+01:00)

Console access
Enabled without MFA

Last console sign-in
Never

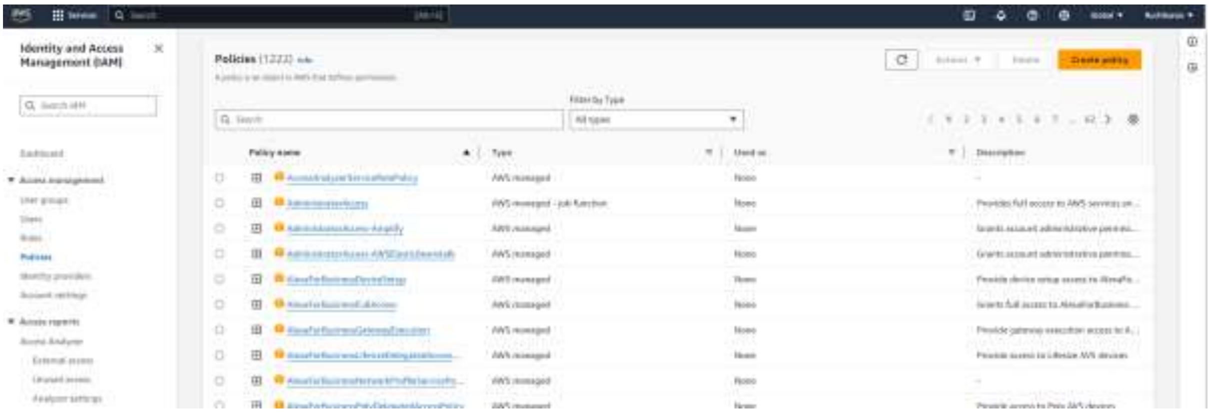
Access key 1
Create access key

Permissions policies (1)

Filter by Type: All types

	Policy name	Type	Attached via
□	Userpolicy	Customer managed	Directly

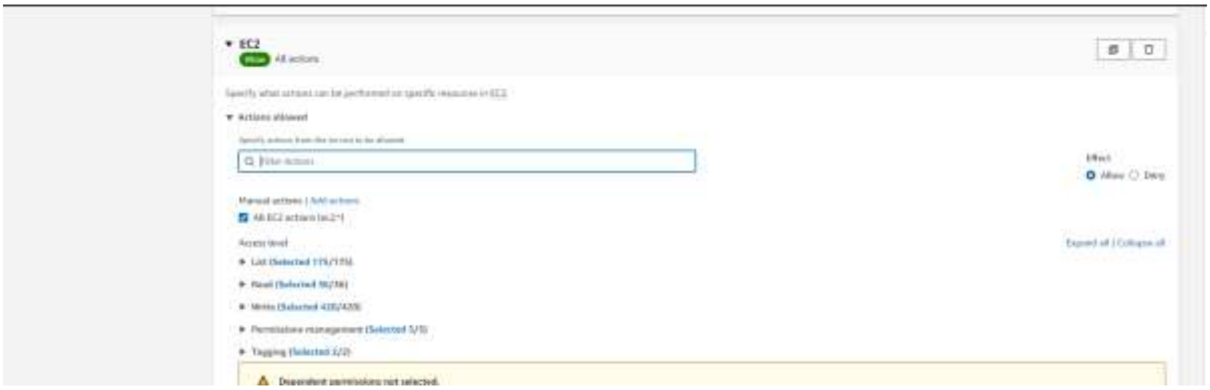
Policy changed !!



Policies (1223) Info

Filter by Type: All types

		Policy name	Type	Used as	Description
○	⊞	AccessAnalyzerServiceRolePolicy	AWS managed	None	-
○	⊞	AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
○	⊞	AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permiss...
○	⊞	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants account administrative permiss...
○	⊞	AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaFo...
○	⊞	AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness ...
○	⊞	AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution access to A...
○	⊞	AlexaForBusinessLifesizeDelegatedAccess...	AWS managed	None	Provide access to Lifesize AVS devices
○	⊞	AlexaForBusinessNetworkProfileServicePo...	AWS managed	None	-
○	⊞	AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None	Provide access to Poly AVS devices



▾ EC2
Allow All actions

Specify what actions can be performed on specific resources in EC2

▾ Actions allowed

Specify actions from the service to be allowed

Filter Actions

Effect
● Allow ○ Deny

Manual actions | Add actions

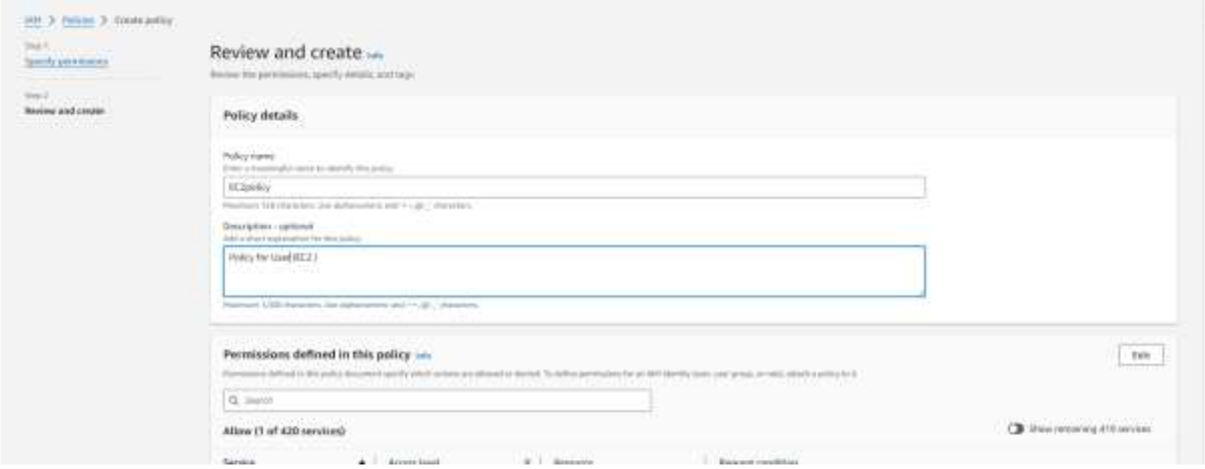
☑ All EC2 actions (ec2:*)

Access level

Expand all | Collapse all

- ▸ List (Selected 175/175)
- ▸ Read (Selected 18/18)
- ▸ Write (Selected 420/420)
- ▸ Permissions management (Selected 5/5)
- ▸ Tagging (Selected 2/2)

⚠ Dependent permissions not selected.



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details and tags.

Policy details

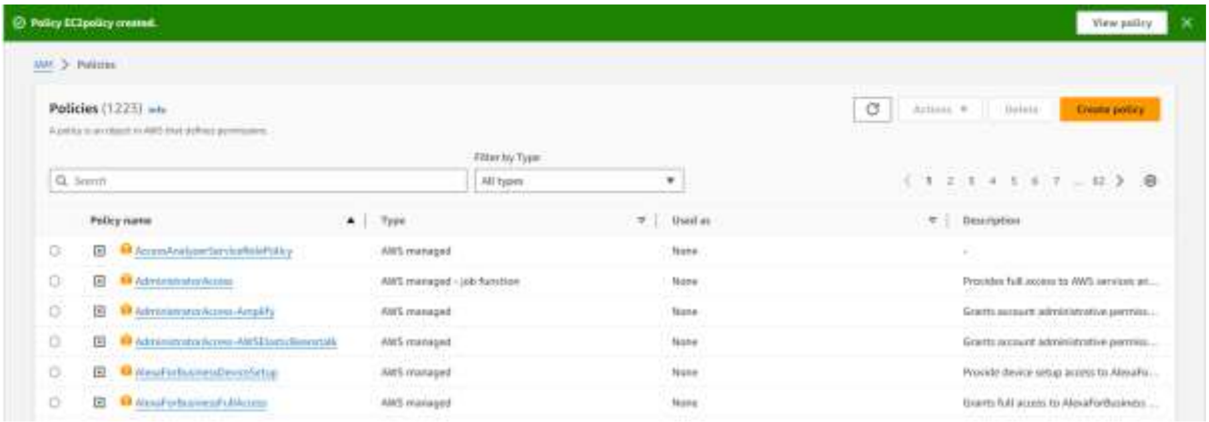
Policy name
EC2policy

Description - optional
Policy for User(EC2)

Permissions defined in this policy Info

Allow (1 of 420 services)

Service | Access level | Resource | Request condition



Policy EC2policy created.

IAM > Policies

Policies (1223) Info

A policy is an object in AWS that defines permissions.

Filter by Type: All types

		Policy name	Type	Used as	Description
○	⊞	AccessAnalyzerServiceRolePolicy	AWS managed	None	-
○	⊞	AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
○	⊞	AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permiss...
○	⊞	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants account administrative permiss...
○	⊞	AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaFo...
○	⊞	AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness ...