SVKM'S NMIM'S Nilkamal School of Mathematics, Applied Statistics & Analytics

Master of Science (Data Science)

Practical-5 Identity Access Management.

Date:-11/03/2024 **Submission Date:- 18/03/2024**

Name : Upasana Yadav **Roll no : A015**

Writeup :-

- **Users and groups**

Root user

The root user will automatically be created and granted unrestricted rights. We can create an admin user with fewer powers to control the entire Amazon account.

IAM Users

We can utilize IAM users to access the AWS Console and their administrative permissions differ from those of the Root user and if we can keep track of their login information.

Example

With the aid of IAM users, we can accomplish our goal of giving a specific person access to every service available in the Amazon dashboard with only a limited set of permissions, such as read-only access. Let's say user-1 is a user that I want to have read-only access to the EC2 instance and no additional permissions, such as create, delete, or update. By creating an IAM user and attaching user-1 to that IAM user, we may allow the user access to the EC2 instance with the required permissions.

IAM Groups

A group is a collection of users, and a single person can be a member of several groups. With the aid of groups, we can manage permissions for many users quickly and efficiently.

Example

Consider two users named user-1 and user-2. If we want to grant user-1 specific permissions, such as the ability to delete, create, and update the auto-calling group only, and if we want to grant user-2 all the necessary permissions to maintain the auto-scaling group as well as the ability to maintain EC2, we can create groups and add this user to them. If a new user is added, we can add that user to the required group with the necessary permissions.

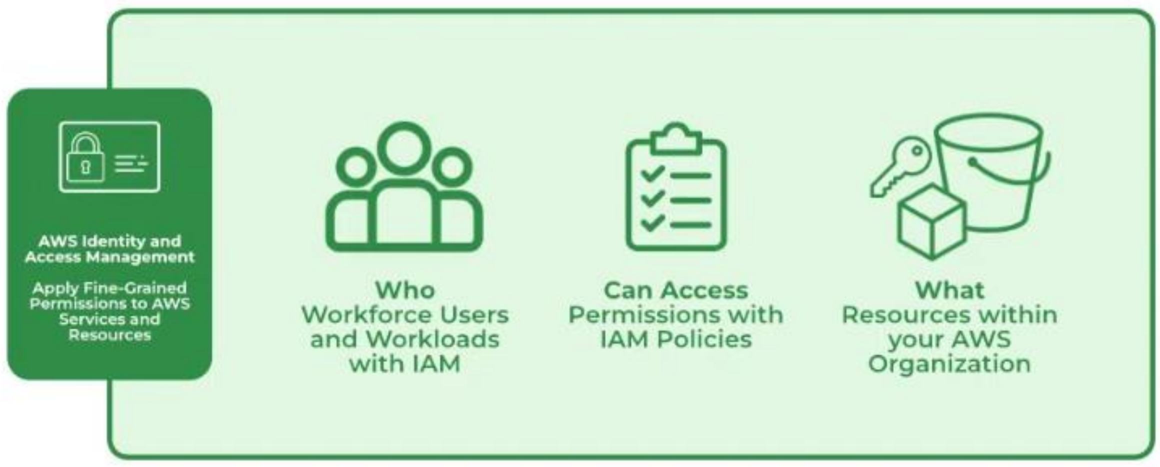
IAM Roles

While policies cannot be directly given to any of the services accessible through the Amazon dashboard, IAM roles are similar to IAM users in that they may be assumed by anybody who requires them. By using roles, we can provide AWS Services access rights to other AWS Services.

- **IAM**

Identity and Access Management (IAM) manages Amazon Web Services (AWS) users and their access to AWS accounts and services. It controls the level of access a user can have over an AWS account & set users, grant permission, and allows a user to use different features of an AWS account. Identity and access management is mainly used to manage users, groups, roles, and Access policies The account we created to sign in to Amazon web services is known as the root account and it holds all the administrative rights and has access to all parts of the account. The new user created an AWS account, by default they have no access to any services in the account & it is done with the help of IAM that the root account holder can implement access policies and grant permission to the user to access certain services.

How IAM Works?
IAM verifies that a user or service has the necessary authorization to access a particular service in the AWS cloud. We can also use IAM to grant the right level of access to specific users, groups, or services. For example, we can use IAM to enable an EC2 instance to access S3 buckets by requesting fine-grained permissions.


AWS Identity and Access Management
Apply Fine-Grained Permissions to AWS Services and Resources
Who Workforce Users and Workloads with IAM
Can Access Permissions with IAM Policies
What Resources within your AWS Organization

- **Role of IAM**

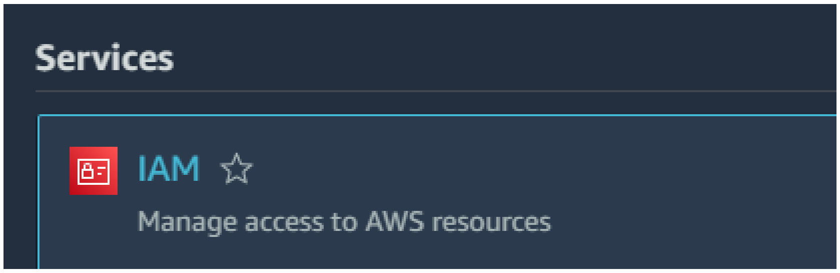
An IAM role is an identity within your AWS account that has specific permissions. It's similar to an IAM user, but isn't associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles.

IAM roles with temporary credentials are used in the following situations:

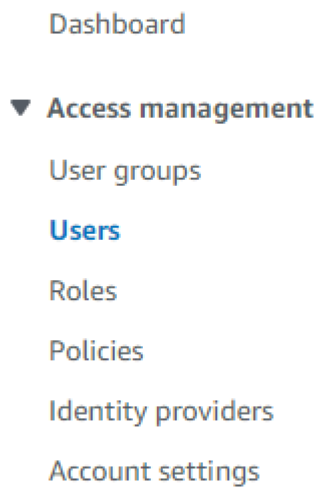
- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

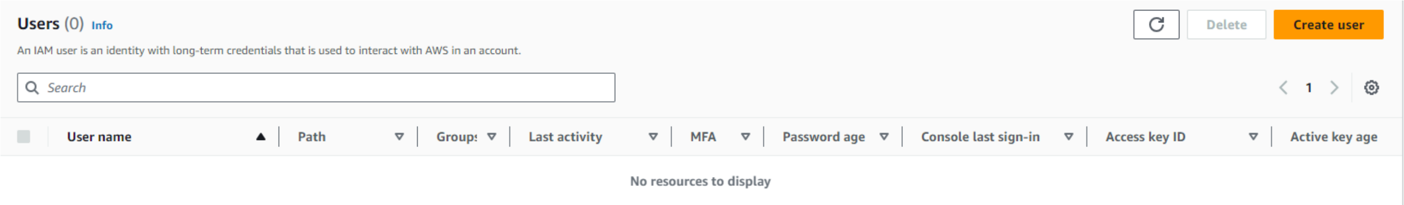
Create and Implement policies IAM user for accessing any 4 services from the aws user and group.

GO TO IAM



CLICK ON USERS



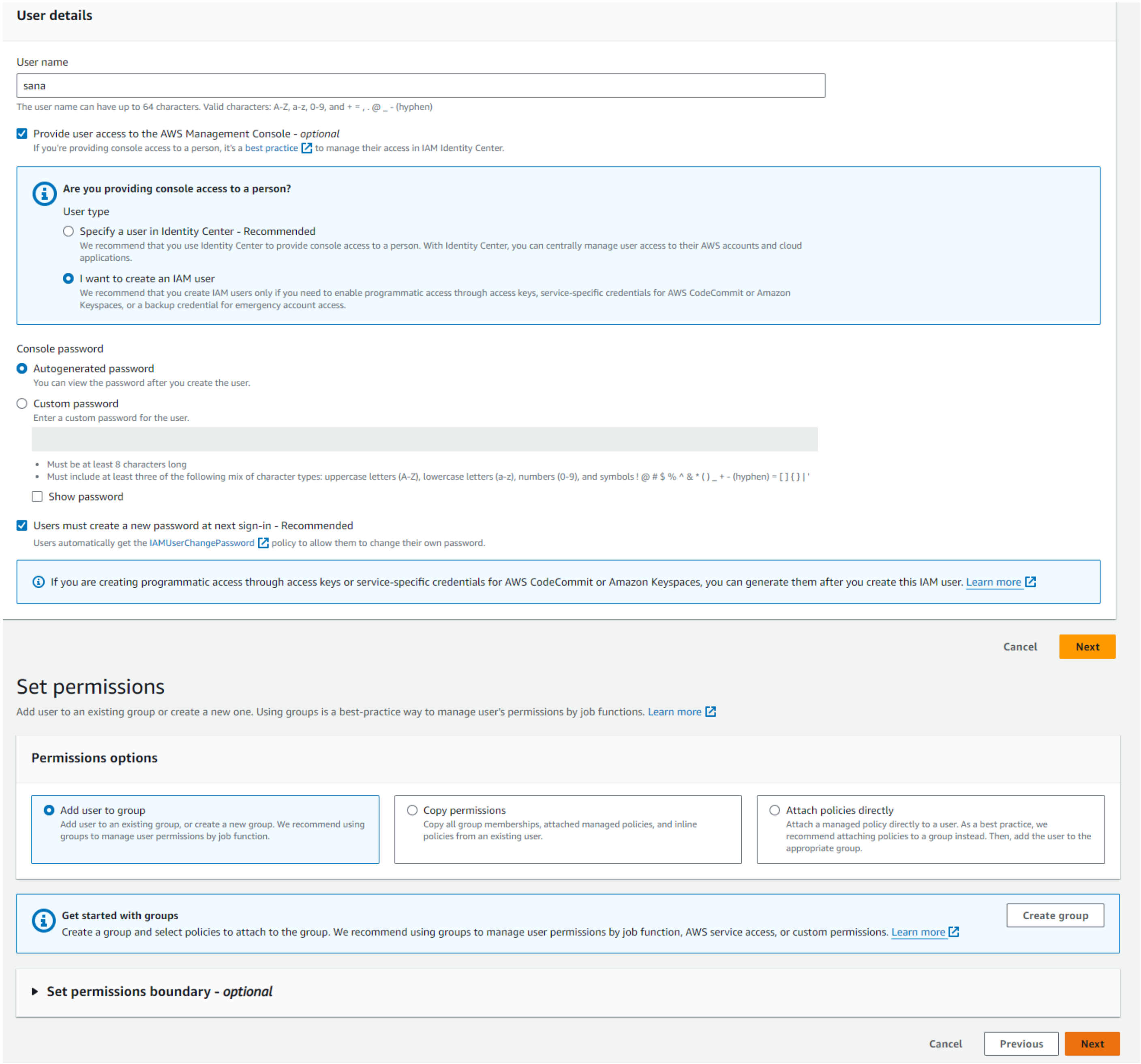


An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

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

No resources to display

FULL ALL THE DETAILS AS FOLLOWS



User details

User name

sana

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.

Are you providing console access to a person?

User type

○ Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

● I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

● Autogenerated password
You can view the password after you create the user.

○ Custom password
Enter a custom password for the user.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '

□ Show password

☑ Users must create a new password at next sign-in - Recommended
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

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

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

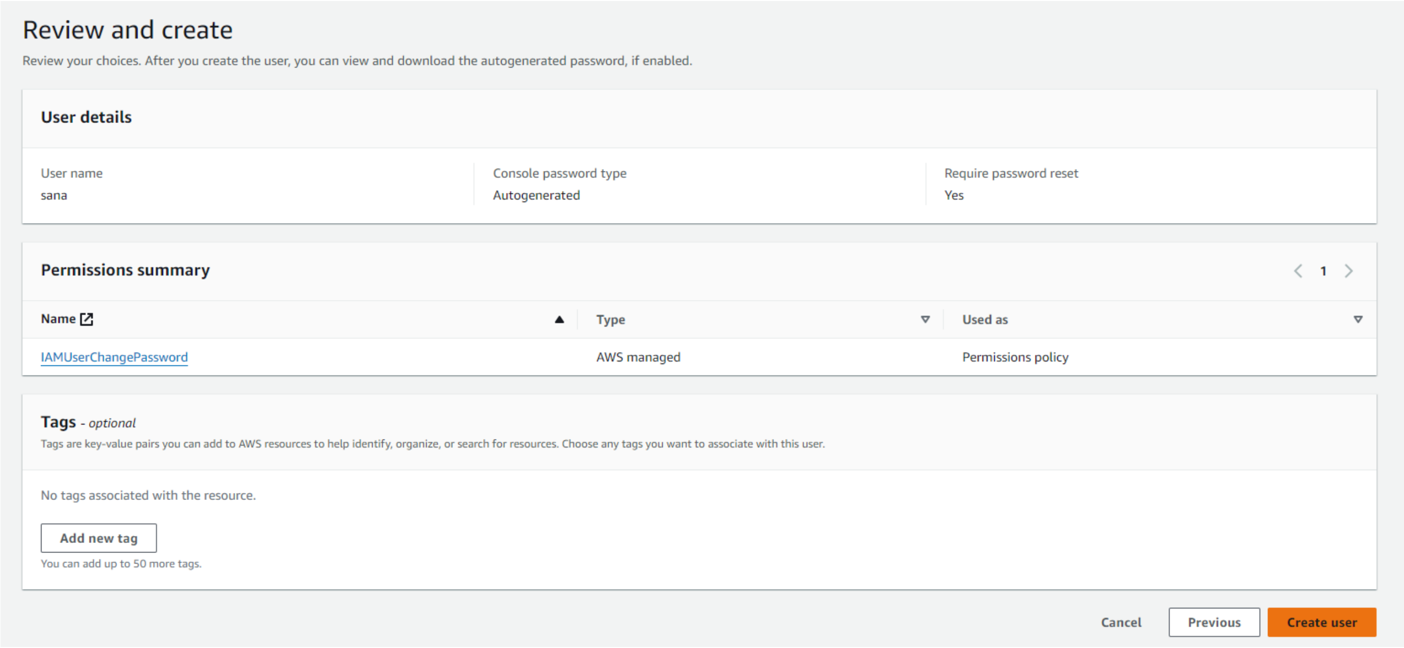
● Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ 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.

Get started with groups
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more

▸ **Set permissions boundary - *optional***



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	Console password type	Require password reset
sana	Autogenerated	Yes

Permissions summary

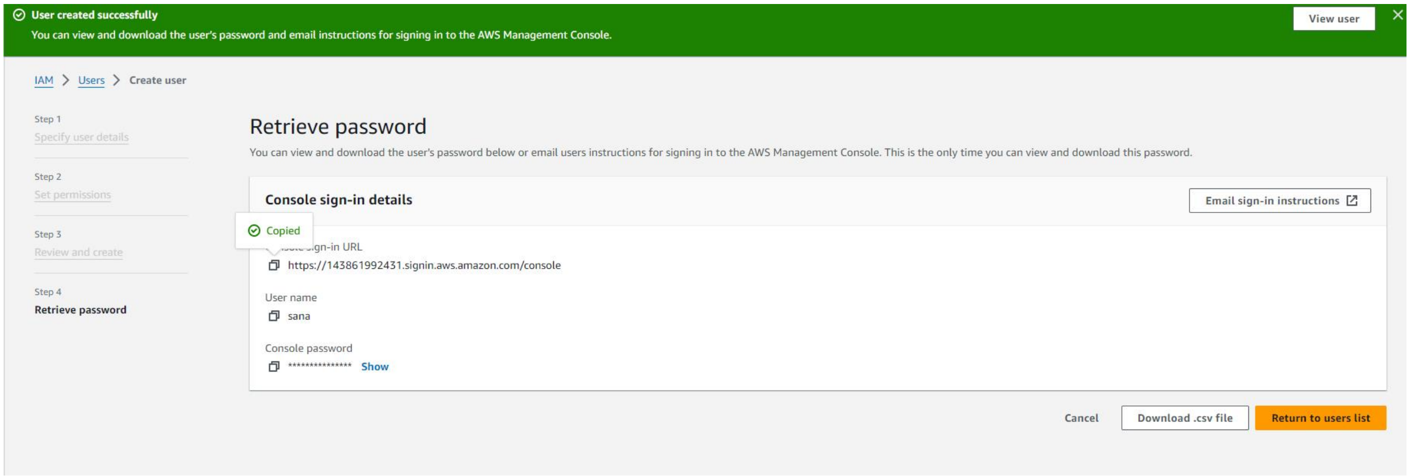
Name	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy

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.

You can add up to 50 more tags.



User created successfully

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

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

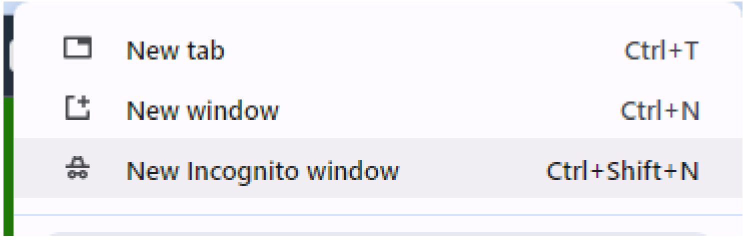
Copied

sign-in URL
https://143861992431.signin.aws.amazon.com/console

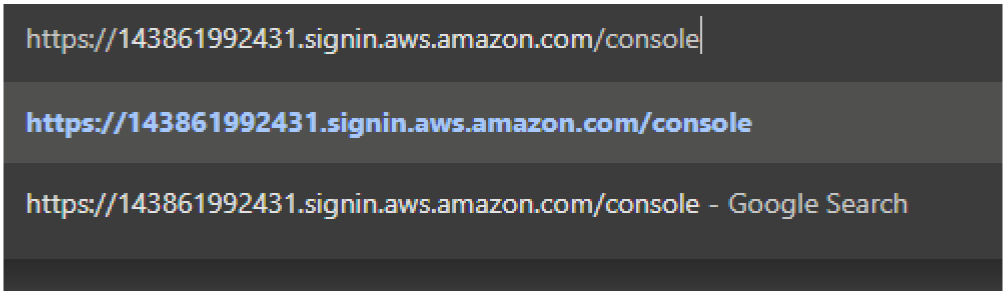
User name
sana

Console password
************** Show

GO TO INCOGNITO MODE



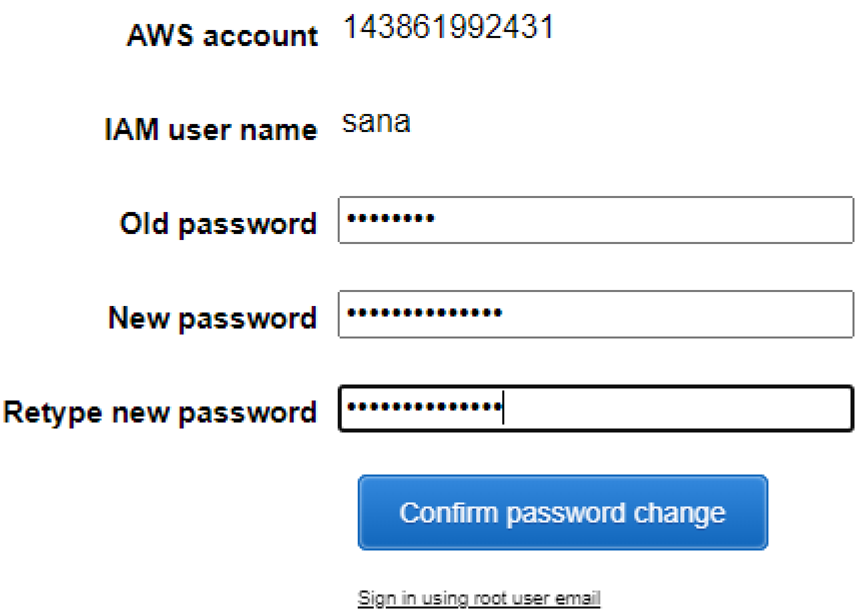
PASTE THE URL AND PROCEED



https://143861992431.signin.aws.amazon.com/console

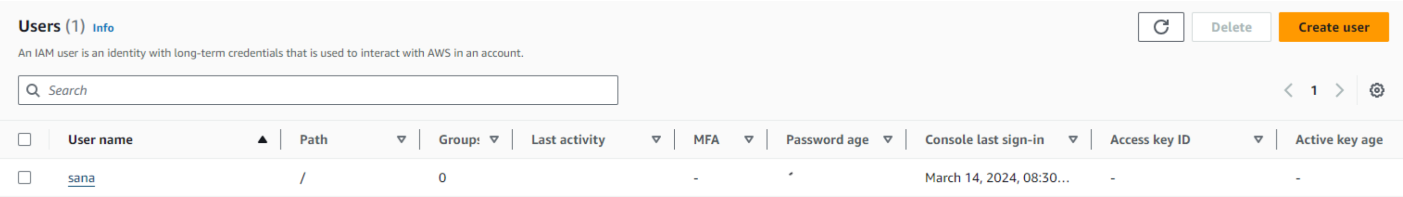
https://143861992431.signin.aws.amazon.com/console

https://143861992431.signin.aws.amazon.com/console - Google Search



AWS account	143861992431
IAM user name	sana
Old password	••••••••
New password	••••••••••••••
Retype new password	••••••••••••••

GO BACK TO MAIN ROOT CONSOLE GO TO USERS

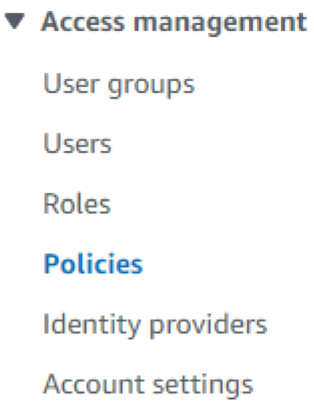


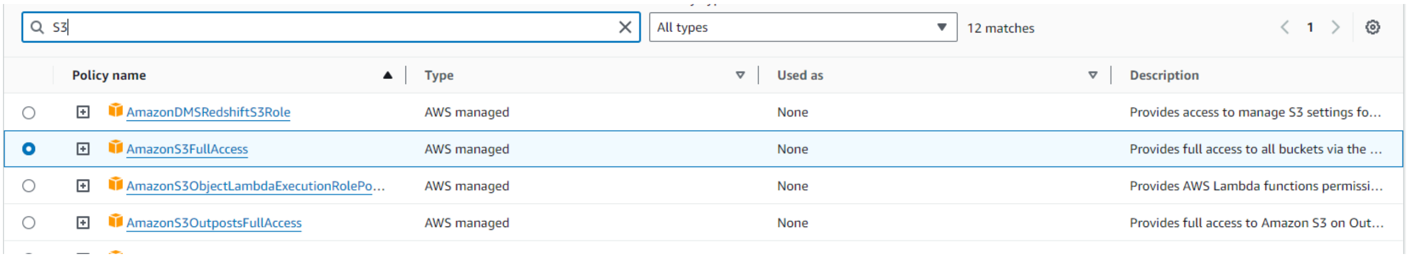
Users (1) Info

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

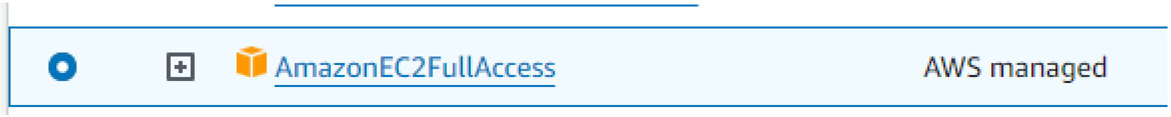
□	User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age
□	sana	/	0		-	'	March 14, 2024, 08:30...	-	-

CLICK ON SANA
AND GO TO POLICIES

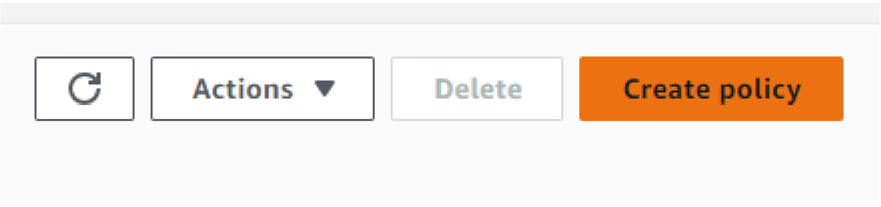




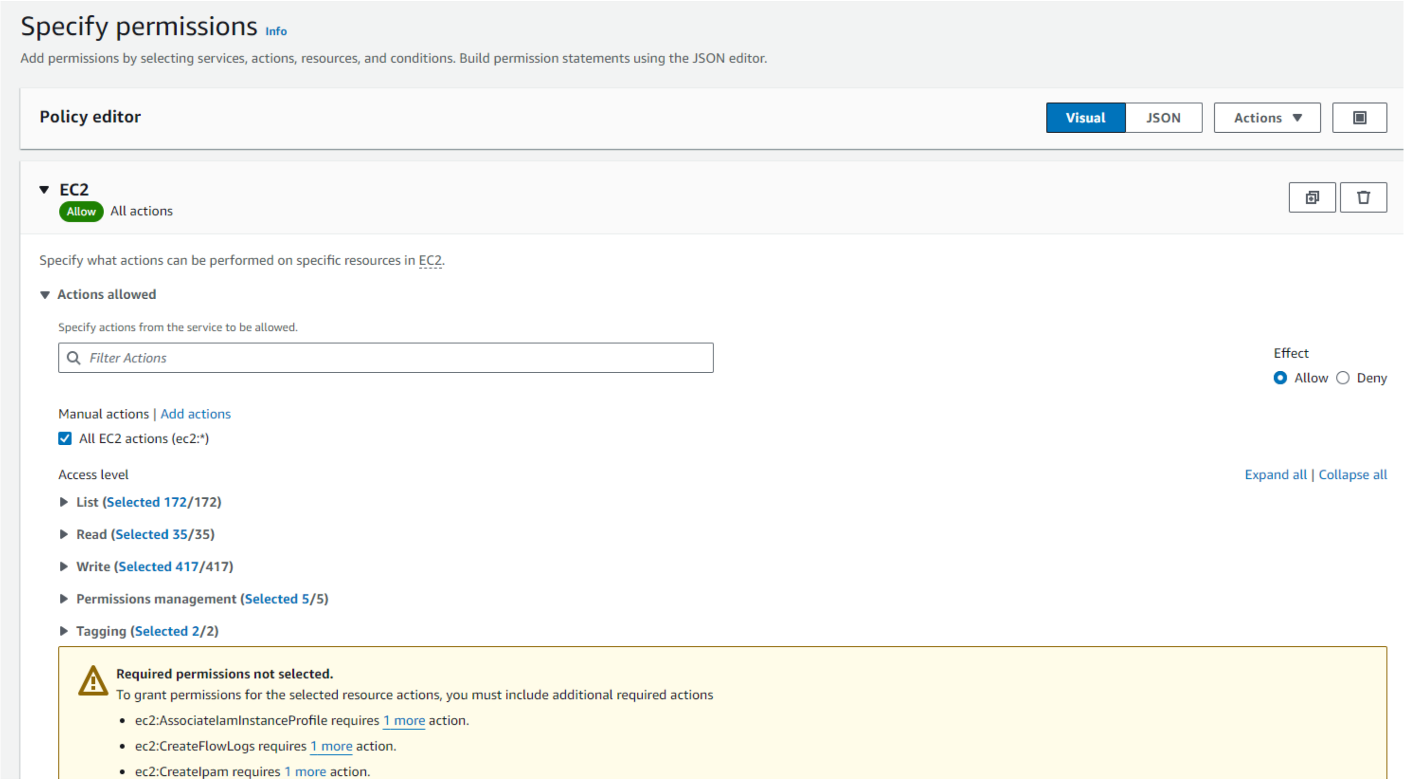
		Policy name ▲	Type	Used as	Description
○	⊞	AmazonDMSRedshiftS3Role	AWS managed	None	Provides access to manage S3 settings fo…
●	⊞	AmazonS3FullAccess	AWS managed	None	Provides full access to all buckets via the …
○	⊞	AmazonS3ObjectLambdaExecutionRolePo…	AWS managed	None	Provides AWS Lambda functions permissi…
○	⊞	AmazonS3OutpostsFullAccess	AWS managed	None	Provides full access to Amazon S3 on Out…



●	⊞	AmazonEC2FullAccess	AWS managed



CREATE POLICY



Specify permissions Info

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

Policy editor

▼ 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

▶ List (Selected 172/172)

▶ Read (Selected 35/35)

▶ Write (Selected 417/417)

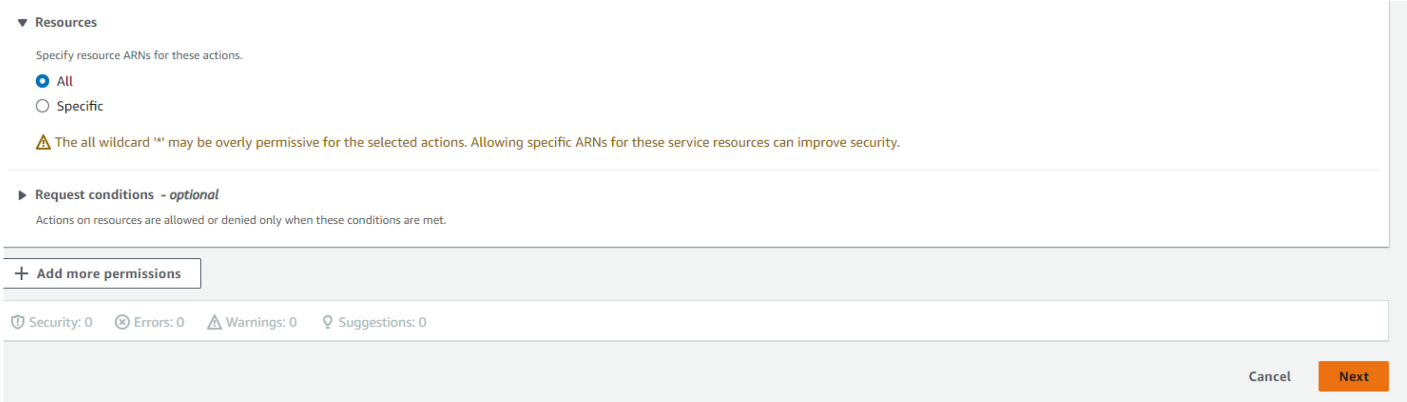
▶ Permissions management (Selected 5/5)

▶ Tagging (Selected 2/2)

Required permissions not selected.

To grant permissions for the selected resource actions, you must include additional required actions

- ec2:AssociateIamInstanceProfile requires 1 more action.
- ec2:CreateFlowLogs requires 1 more action.
- ec2:CreateIpam requires 1 more action.



▼ Resources

Specify resource ARNs for these actions.

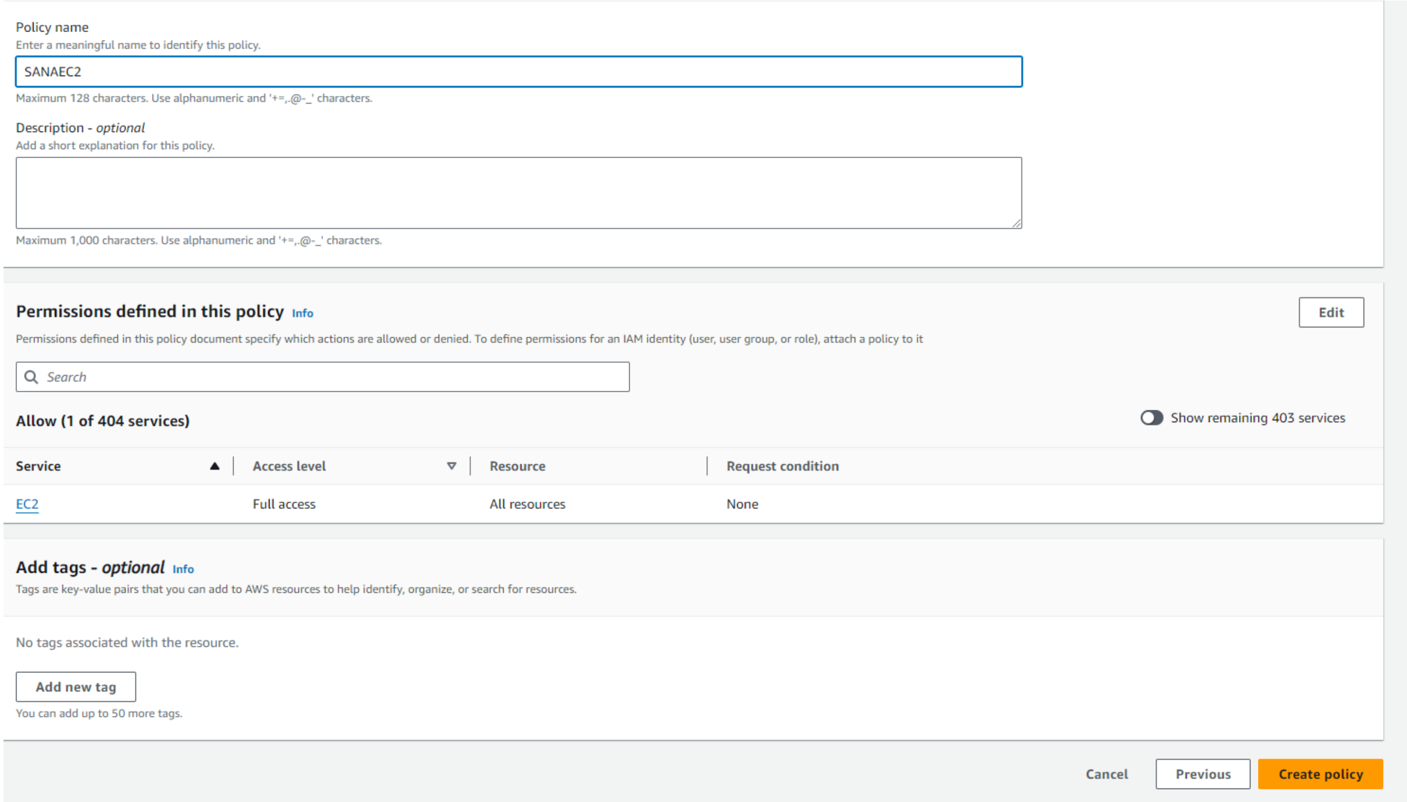
● All

○ 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*

Actions on resources are allowed or denied only when these conditions are met.



Policy name

Enter a meaningful name to identify this policy.

SANAEC2

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 Info

Edit

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

Search

Allow (1 of 404 services)

Show remaining 403 services

Service	Access level	Resource	Request condition
EC2	Full access	All resources	None

Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

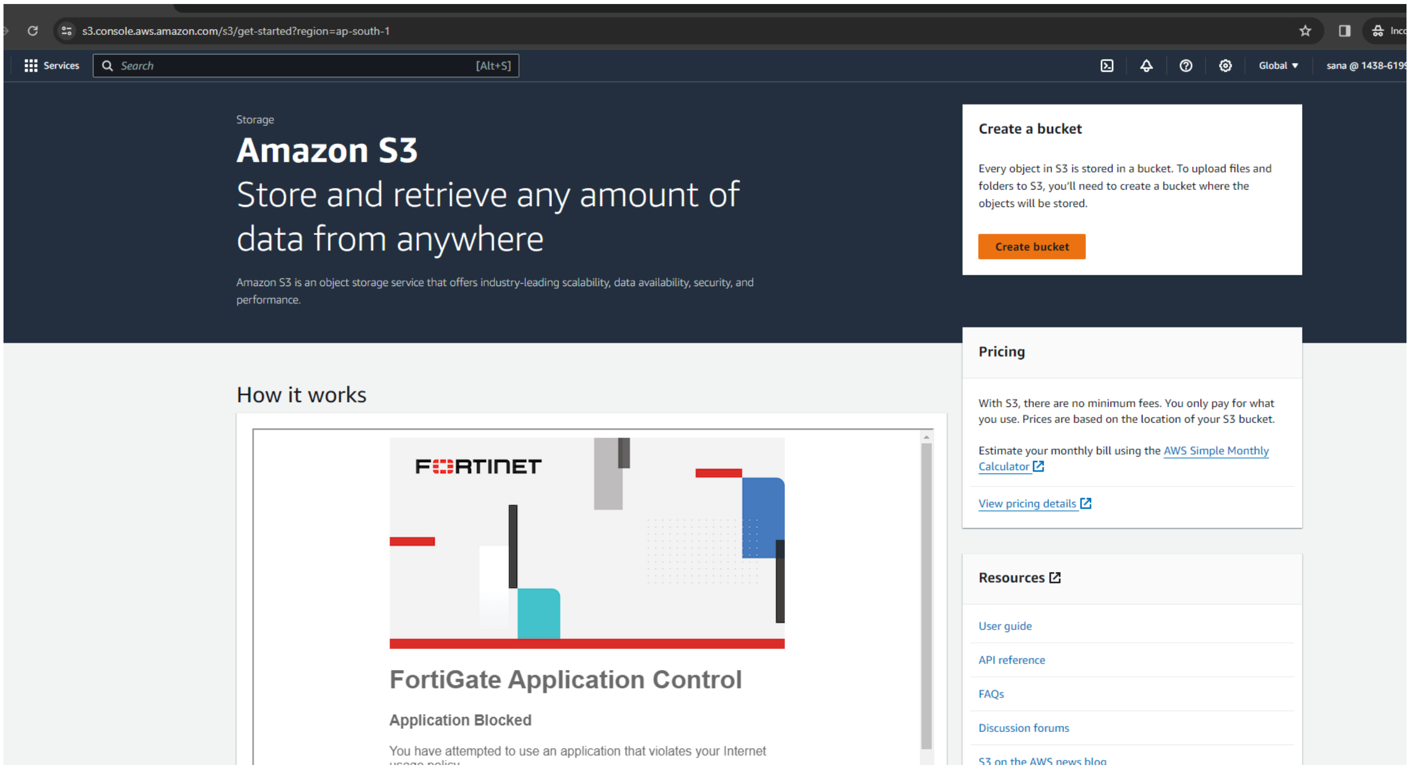
Add new tag

You can add up to 50 more tags.

Cancel | Previous | Create policy

SIMILARLY DO THIS FOR S3

NOW GO TO INCOGNITO MODE AND ACCESS EC2 AND S3 SERVICES



s3.console.aws.amazon.com/s3/get-started?region=ap-south-1

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the AWS Simple Monthly Calculator

View pricing details

Resources

User guide

API reference

FAQs

Discussion forums

S3 on the AWS news blog

How it works

FORTINET

FortiGate Application Control

Application Blocked

You have attempted to use an application that violates your Internet usage policy.