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**

Writeup:-

- **Users and groups**
- **IAM**
- **Role of IAM**

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

Purva Burundkar A018

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 theAmazon 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 specificusers, groups, or services. For example, we can use IAM to enable an EC2 instance to access S3 buckets by requesting fine-grained permissions.

- **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

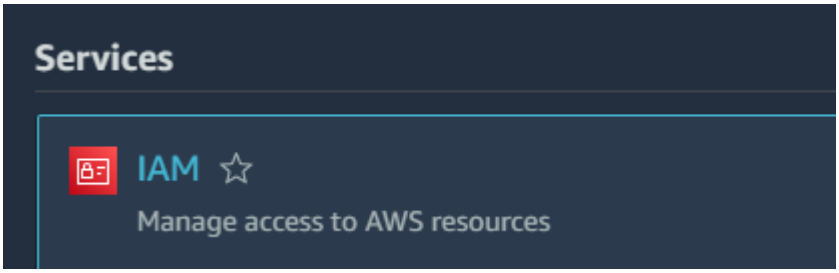
or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles.

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

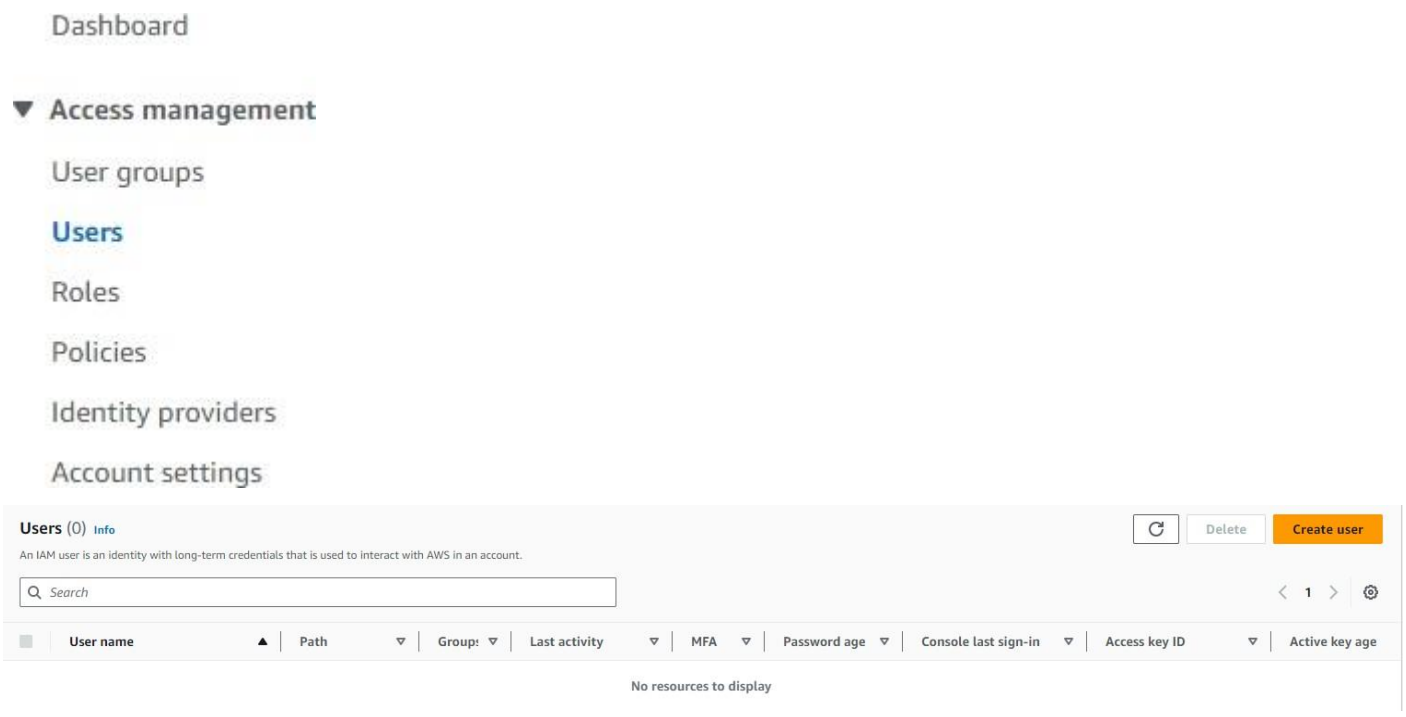
- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles 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 userand group.

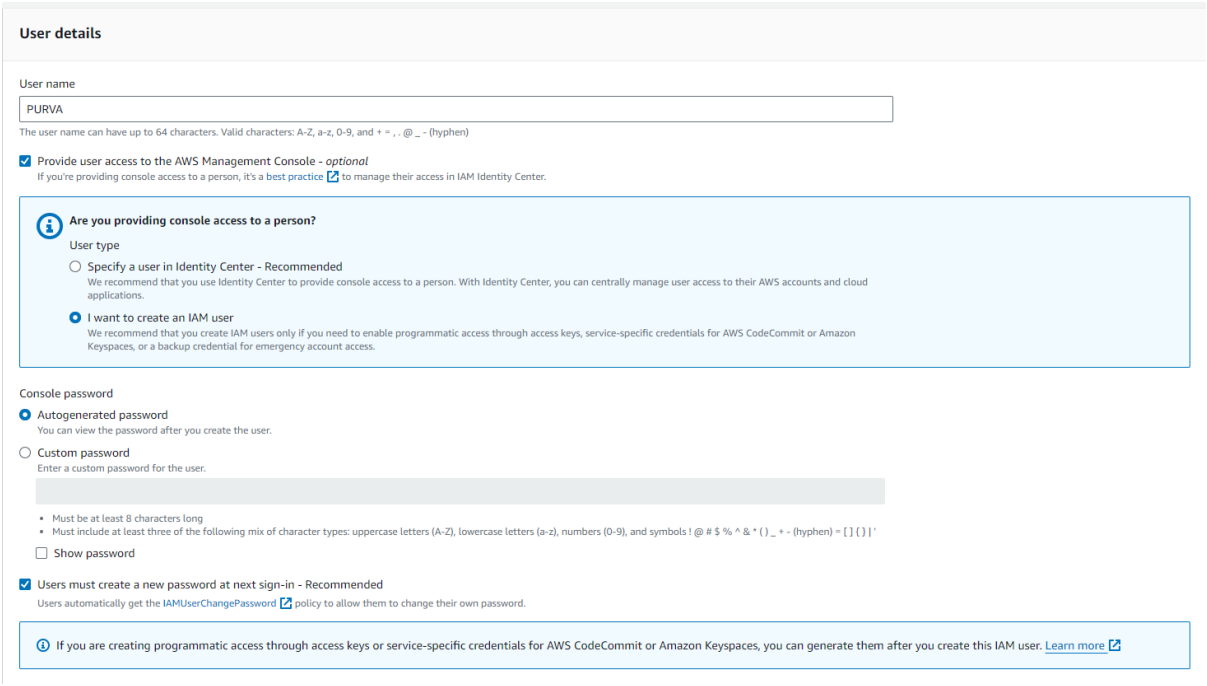
GO TO IAM

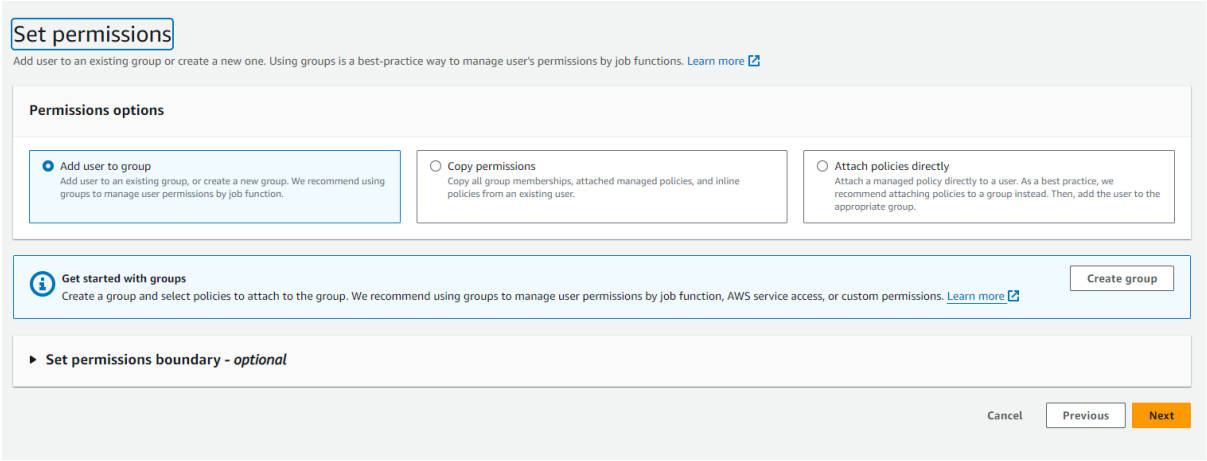


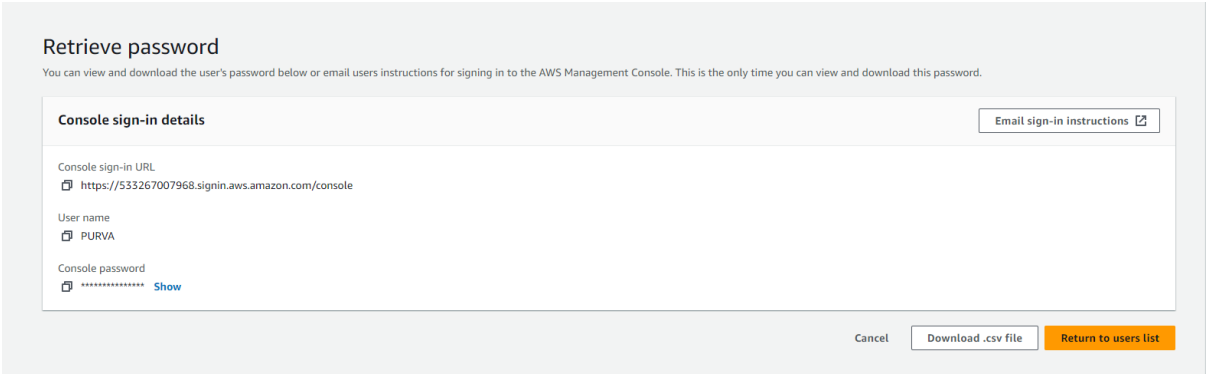
CLICK ON USERS



FULL ALL THE DETAILS AS FOLLOWS







GO TO INCOGNITO MODE

Sign in as IAM user

Account ID (12 digits) or account alias

533267007968

IAM user name

PURVA

Password

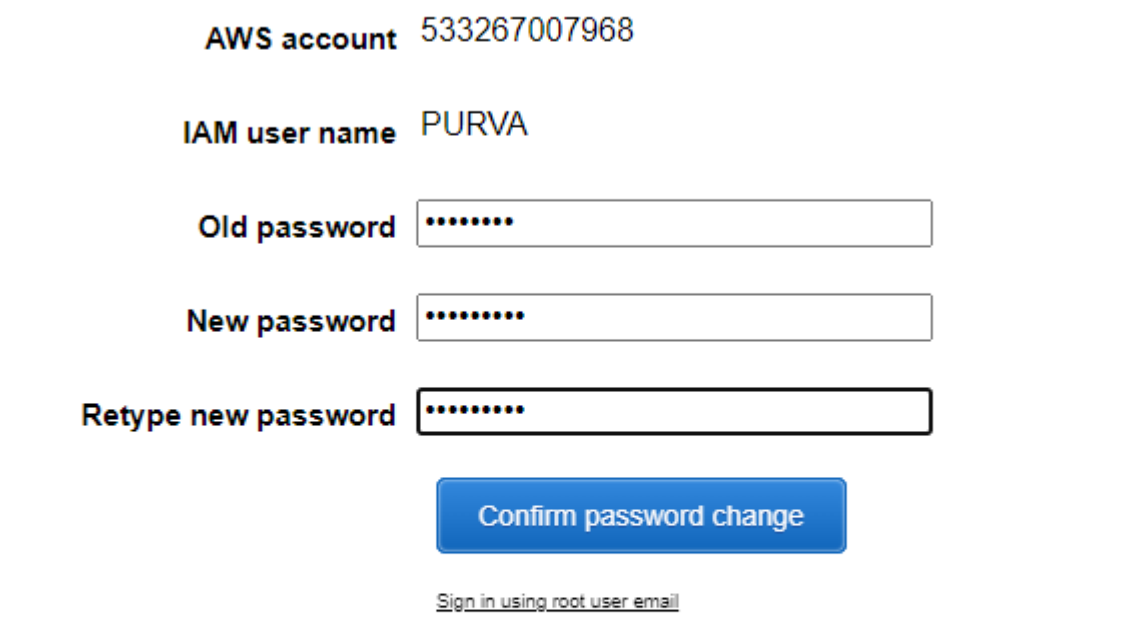
••••••••

☐ Remember this account

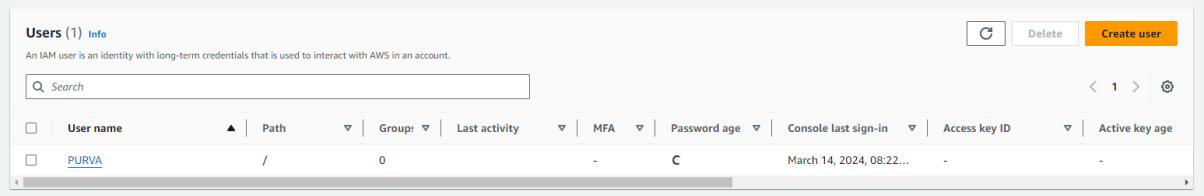
Sign in

Sign in using root user email

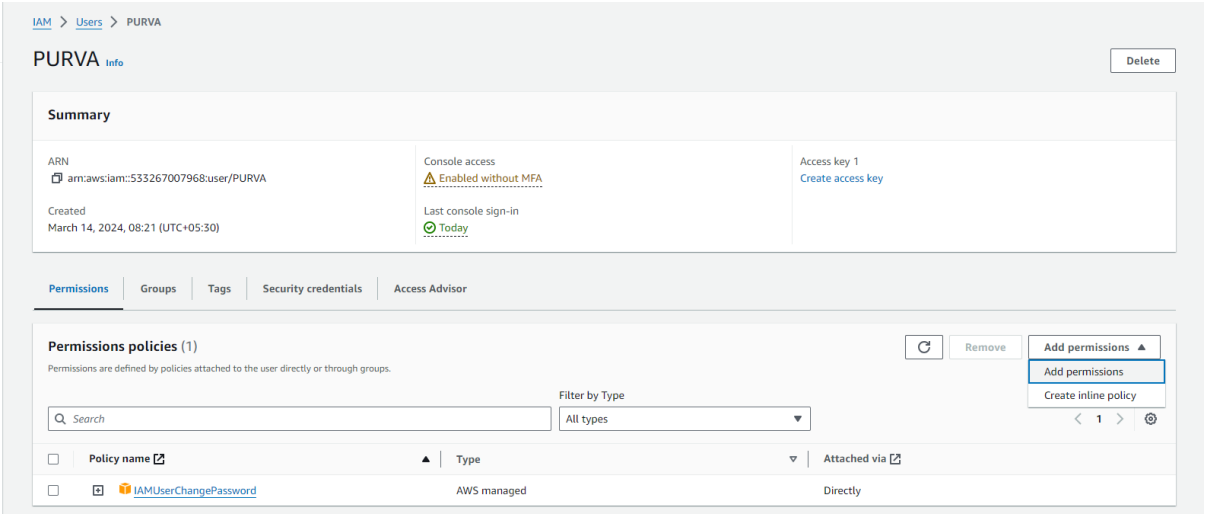
Forgot password?

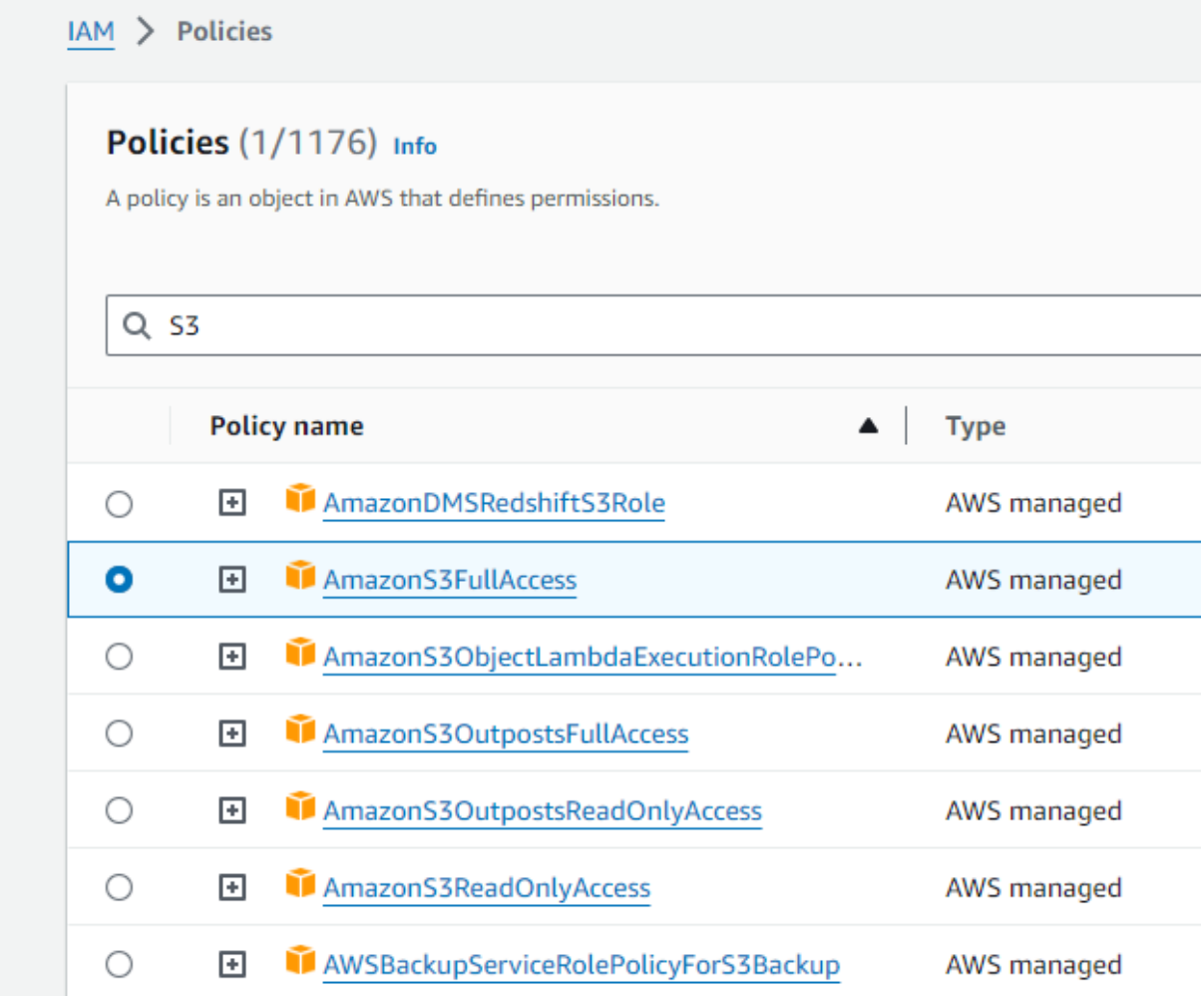


GO BACK TO MAIN ROOT CONSOLE GO TO USERS

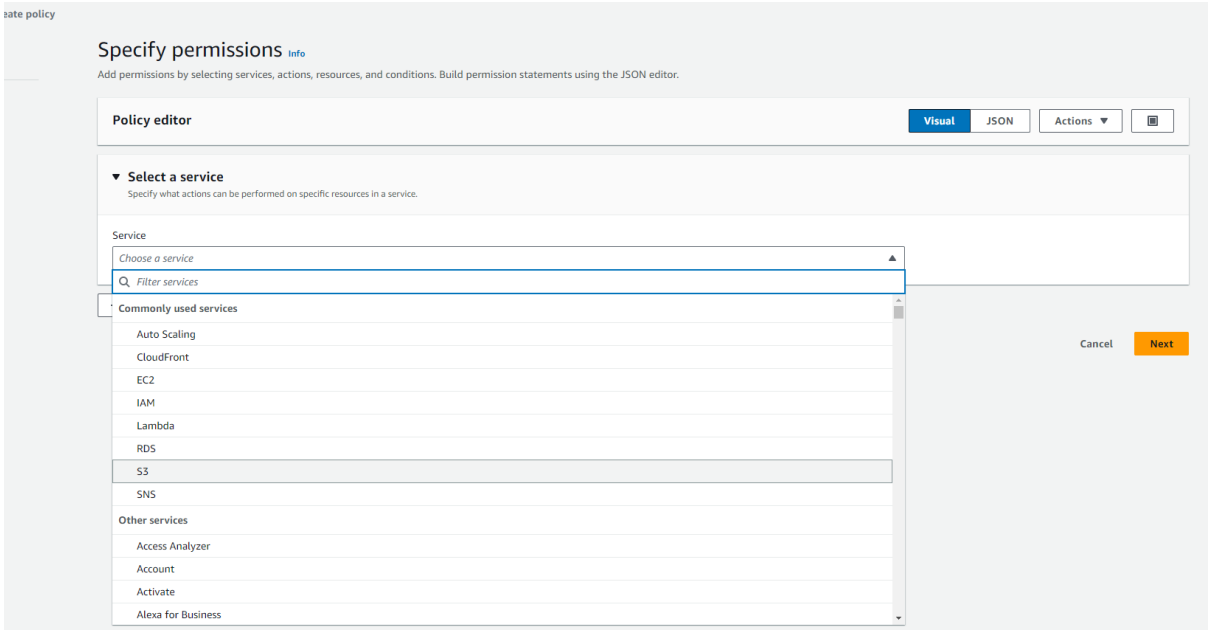


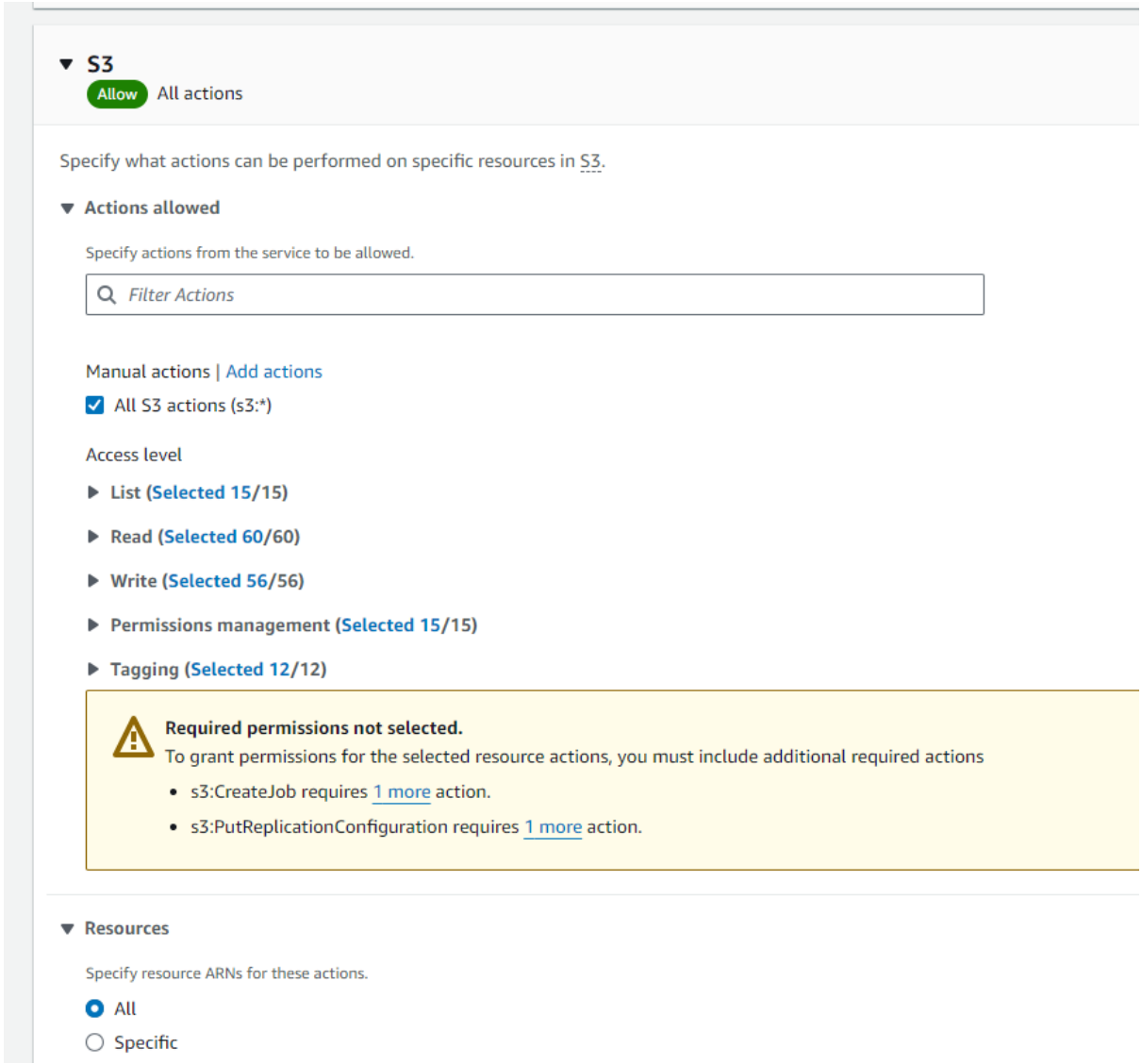
Click on purva and go to policies

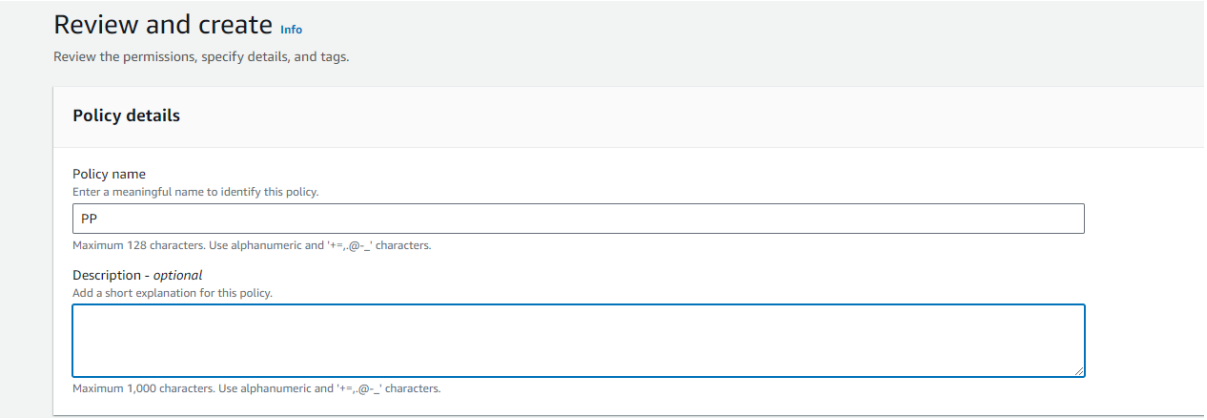




CREATE POLICY

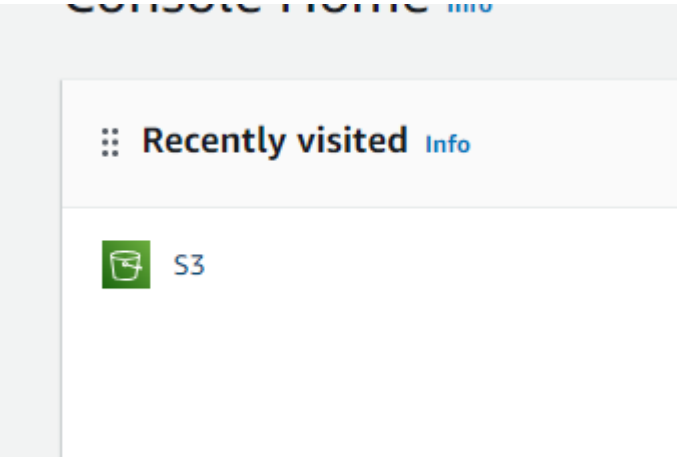

eate policy
Specify permissions Info
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.
Policy editor
Visual
JSON
Actions
Select a service
Specify what actions can be performed on specific resources in a service.
Service
Choose a service
Filter services
Commonly used services
Auto Scaling
CloudFront
EC2
IAM
Lambda
RDS
S3
SNS
Other services
Access Analyzer
Account
Activate
Alexa for Business
Cancel
Next

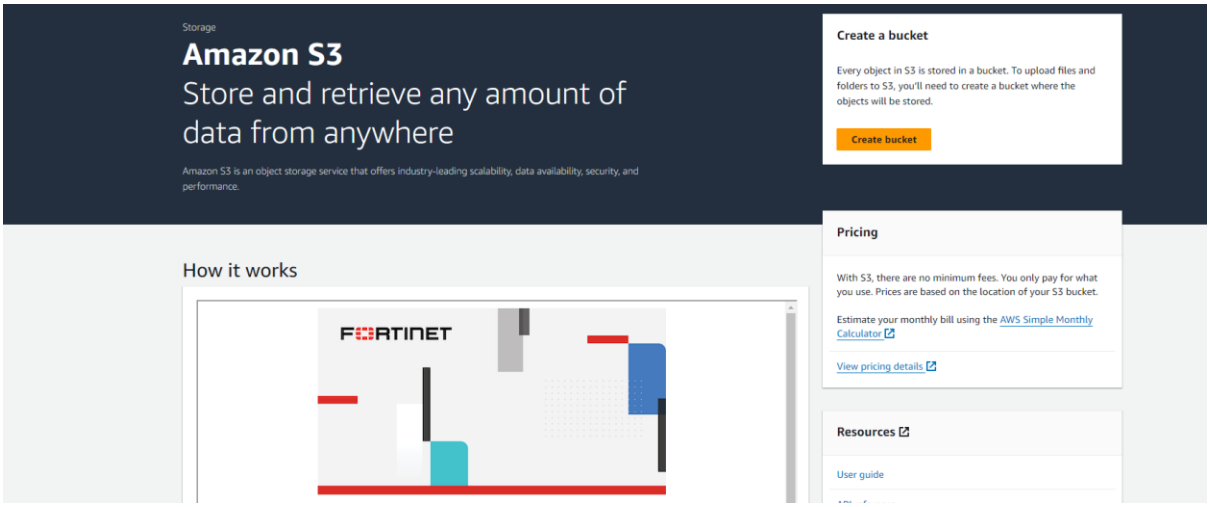

S3
Allow
All actions
Specify what actions can be performed on specific resources in S3.
Actions allowed
Specify actions from the service to be allowed.
Filter Actions
Manual actions | Add actions
All S3 actions (s3:*)
Access level
List (Selected 15/15)
Read (Selected 60/60)
Write (Selected 56/56)
Permissions management (Selected 15/15)
Tagging (Selected 12/12)
Required permissions not selected.
To grant permissions for the selected resource actions, you must include additional required actions
s3:CreateJob requires 1 more action.
s3:PutReplicationConfiguration requires 1 more action.
Resources
Specify resource ARNs for these actions.
All
Specific

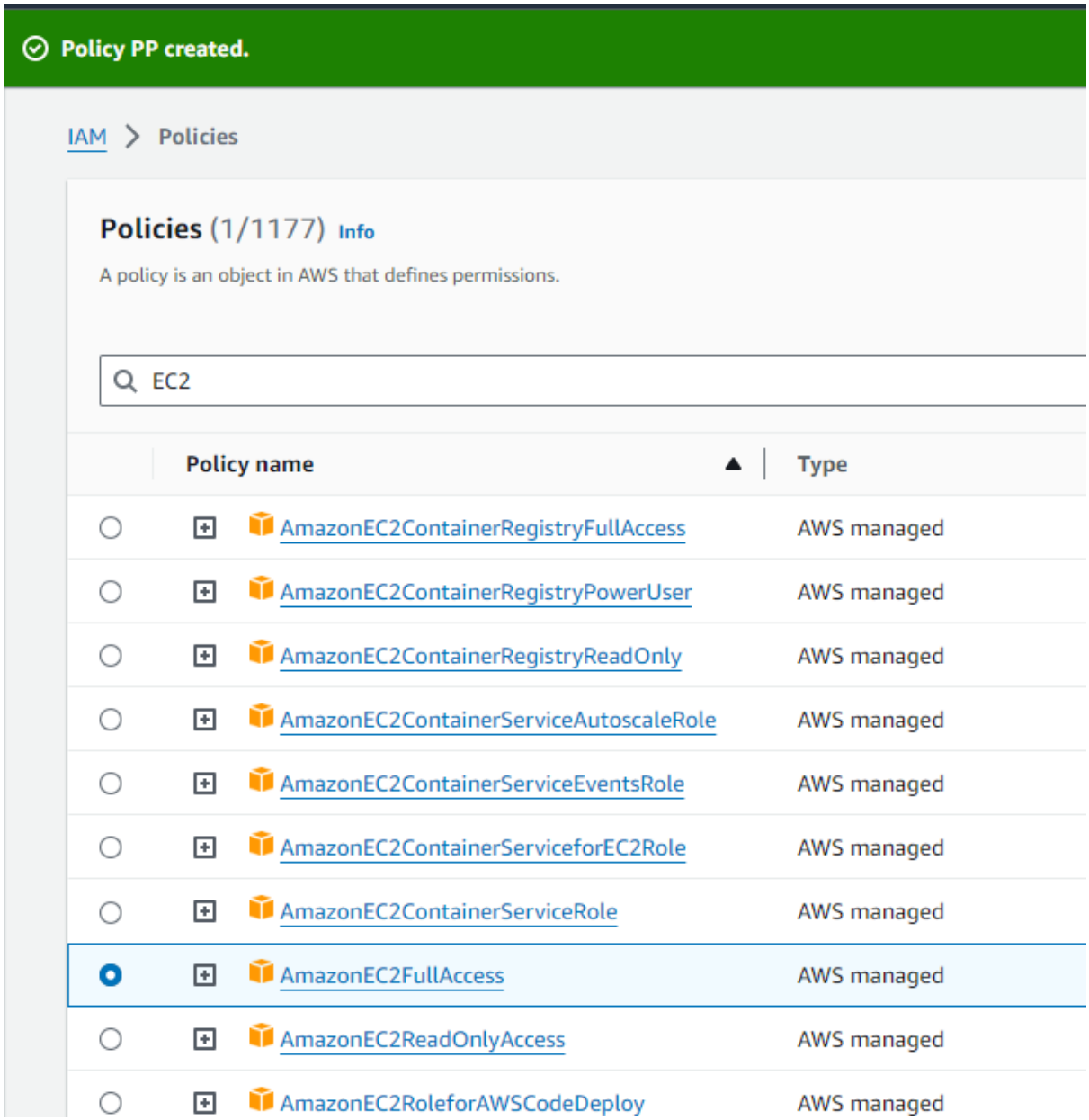


SIMILARLY DO THIS FOR S3

NOW GO TO INCOGNITO MODE AND ACCESS EC2 AND S3 SERVICES






Policy PP created.
IAM
Policies
Policies (1/1177) Info
A policy is an object in AWS that defines permissions.
EC2
Policy name
Type
AmazonEC2ContainerRegistryFullAccess
AWS managed
AmazonEC2ContainerRegistryPowerUser
AWS managed
AmazonEC2ContainerRegistryReadOnly
AWS managed
AmazonEC2ContainerServiceAutoscaleRole
AWS managed
AmazonEC2ContainerServiceEventsRole
AWS managed
AmazonEC2ContainerServiceforEC2Role
AWS managed
AmazonEC2ContainerServiceRole
AWS managed
AmazonEC2FullAccess
AWS managed
AmazonEC2ReadOnlyAccess
AWS managed
AmazonEC2RoleforAWSCodeDeploy
AWS managed

Specify permissions Info

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

Policy editor

▼ **Select a service**

Specify what actions can be performed on specific resources in a service.

Service

Choose a service

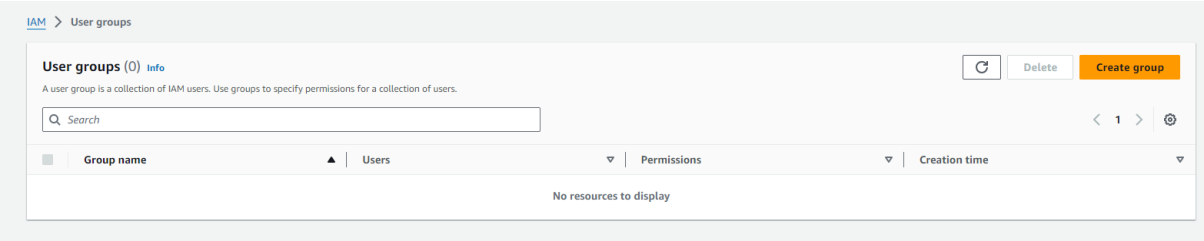
Filter services

Commonly used services

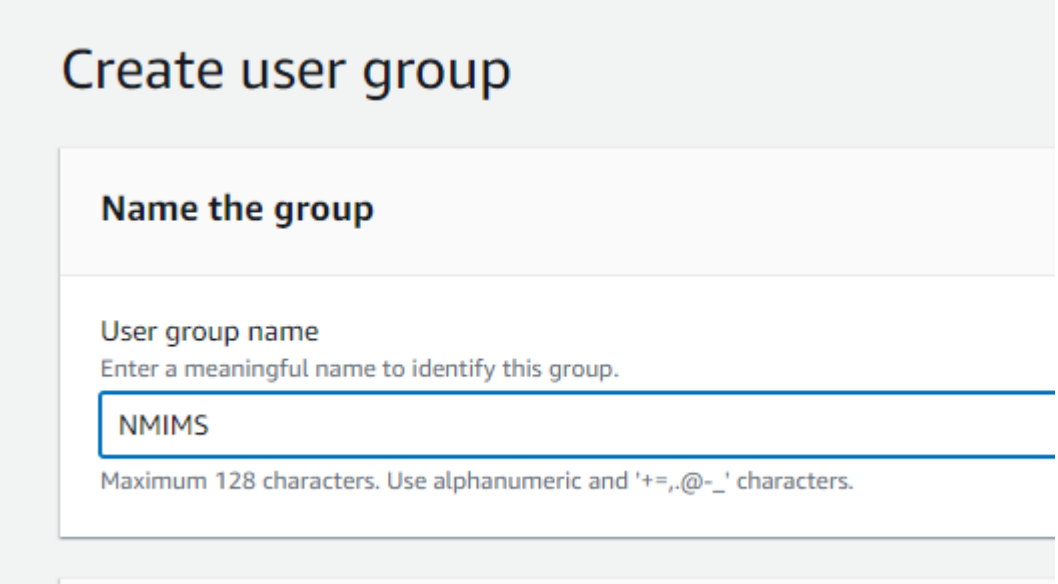
- Auto Scaling
- CloudFront
- EC2
- IAM
- Lambda
- RDS
- S3
- SNS

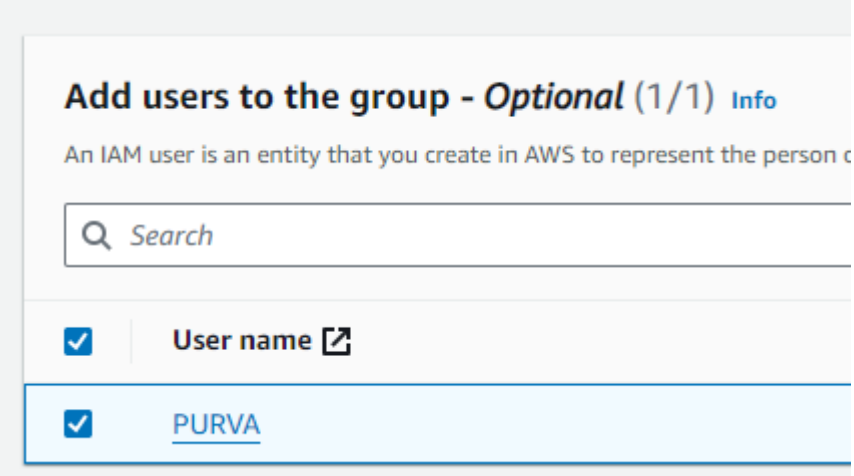
Other services

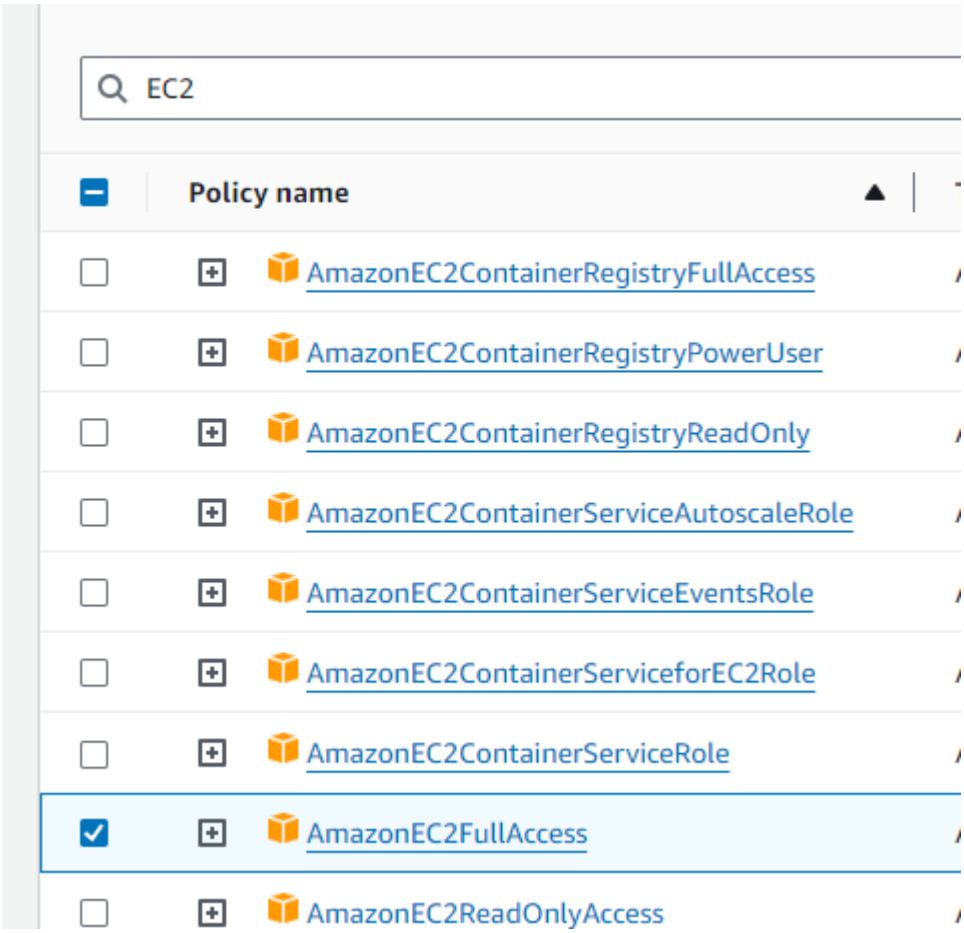
- Access Analyzer



Creating a group






EC2
Policy name
AmazonEC2ContainerRegistryFullAccess
AmazonEC2ContainerRegistryPowerUser
AmazonEC2ContainerRegistryReadOnly
AmazonEC2ContainerServiceAutoscaleRole
AmazonEC2ContainerServiceEventsRole
AmazonEC2ContainerServiceforEC2Role
AmazonEC2ContainerServiceRole
AmazonEC2FullAccess
AmazonEC2ReadOnlyAccess