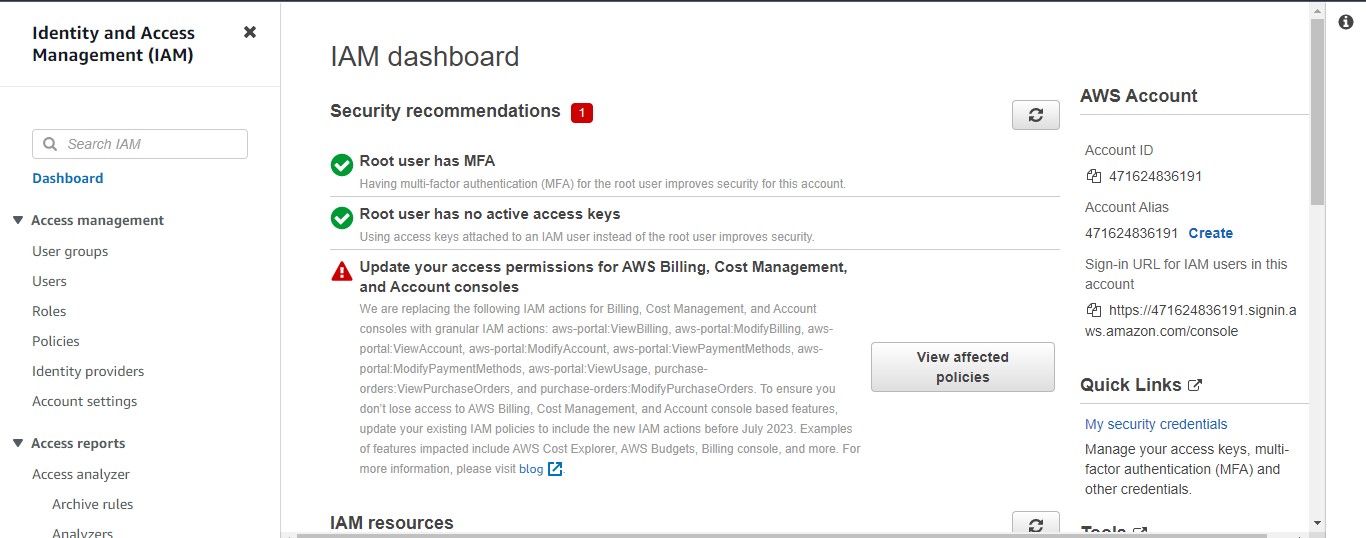
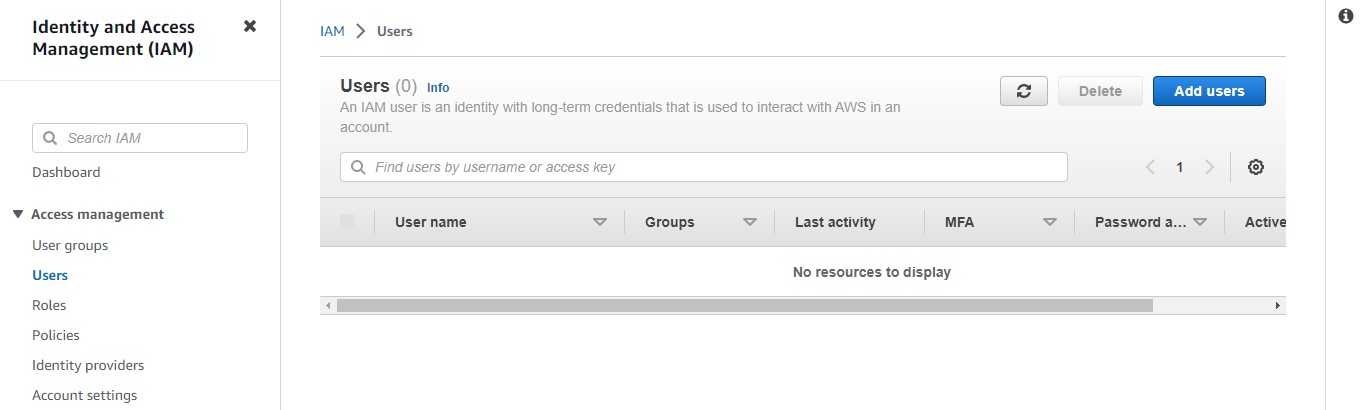
**Experiment – 7: Identity Access Management in AWS**

**i) Create: user, role and policies and access it through Management console and CLI**

**Step 1:-** Login into the aws console as the root user and goto IAM dashboard



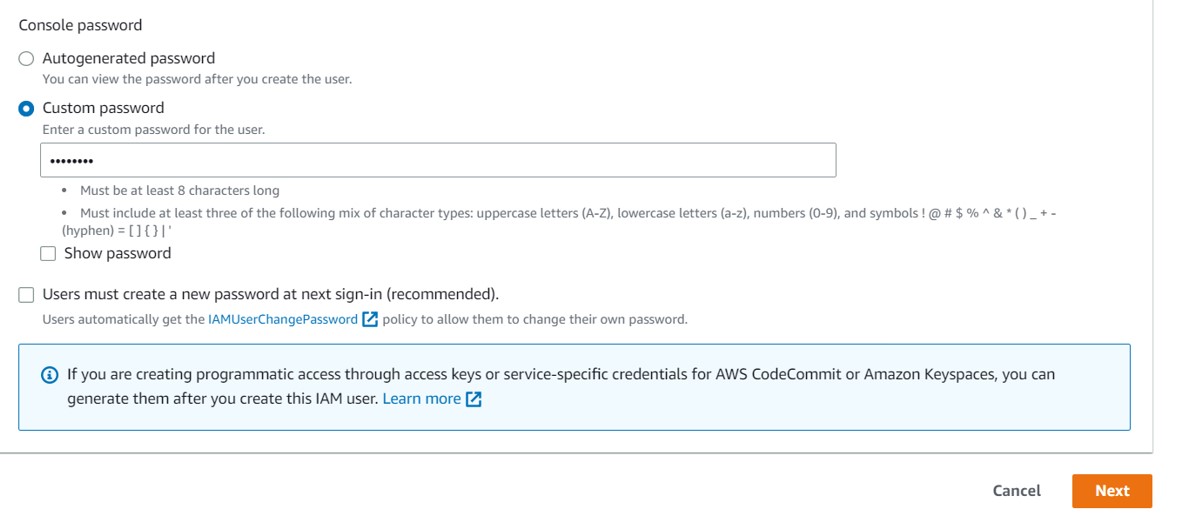
**Step 2:** Goto Access management -> users and click on “Add users”



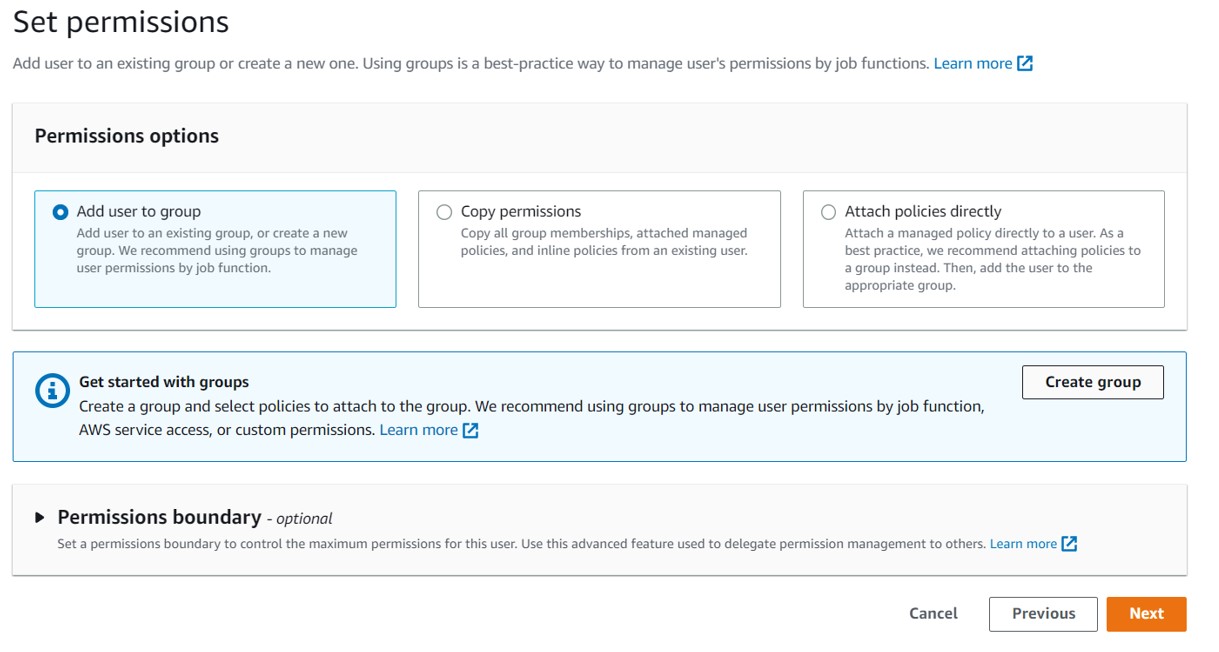
**Step 3:** Specify user details and create an IAM user



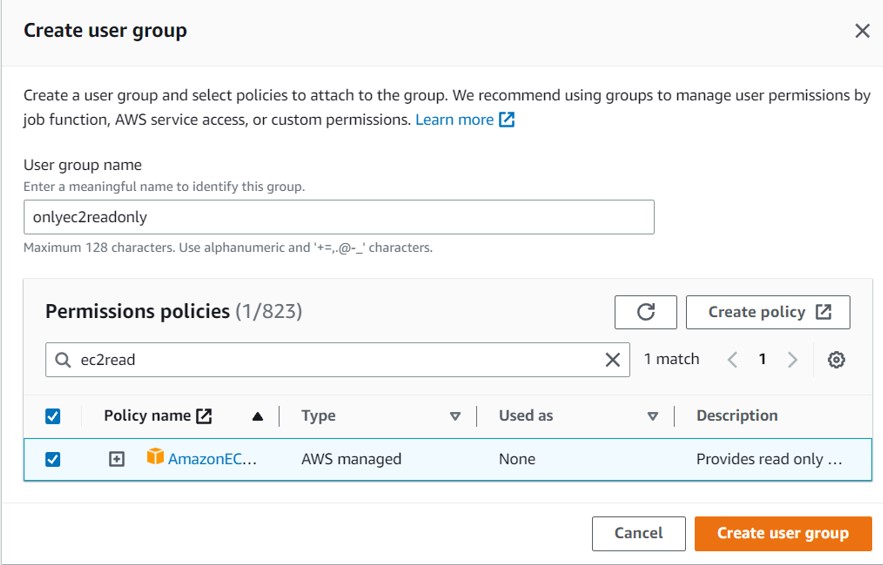
**Step 4:** Set up a custom password



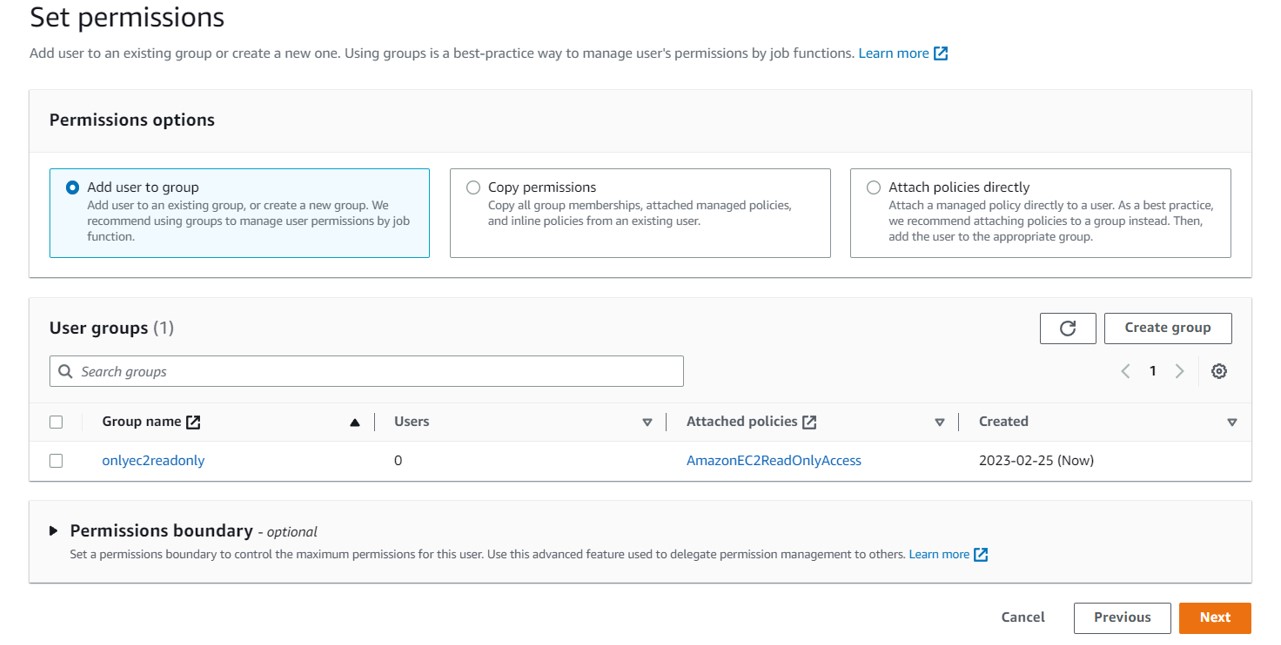
**Step 5:** In the „Set permissions‟ column create a new group to attach the policy



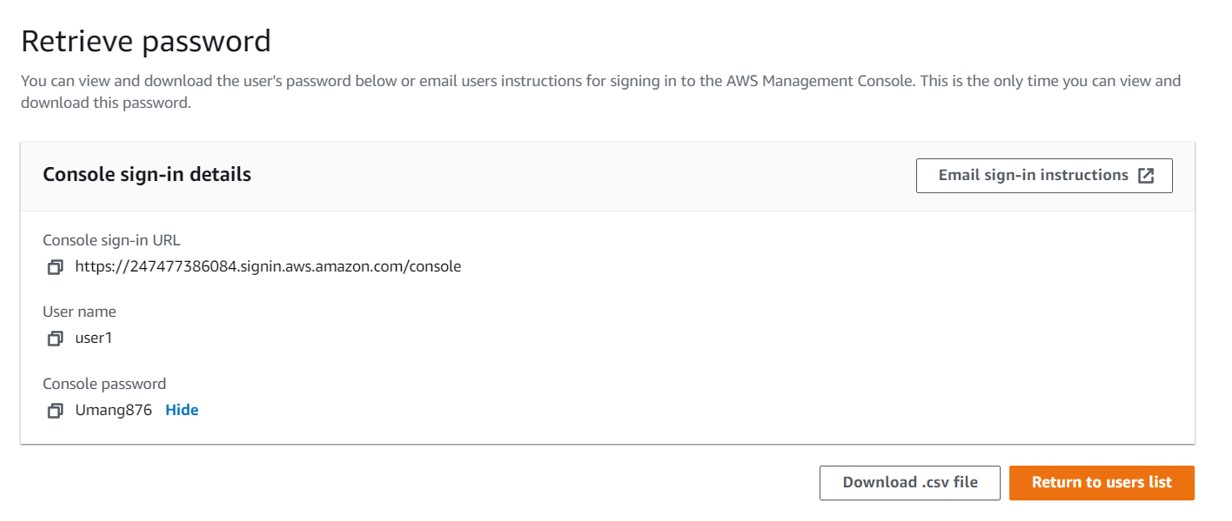
**Step 6:** Assign the group a name and a policy [here ec2readonly]



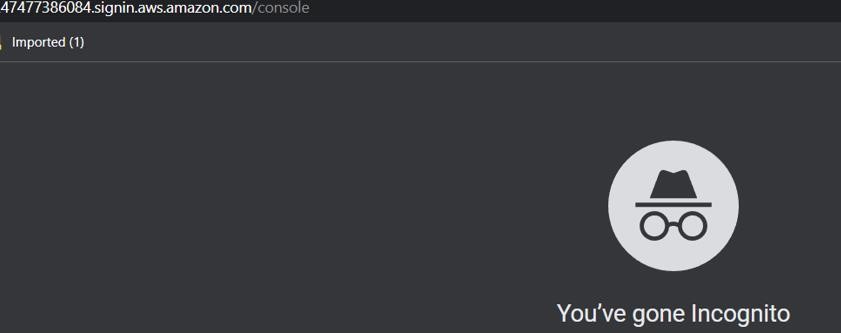
**Step 7:** Review and create the user group

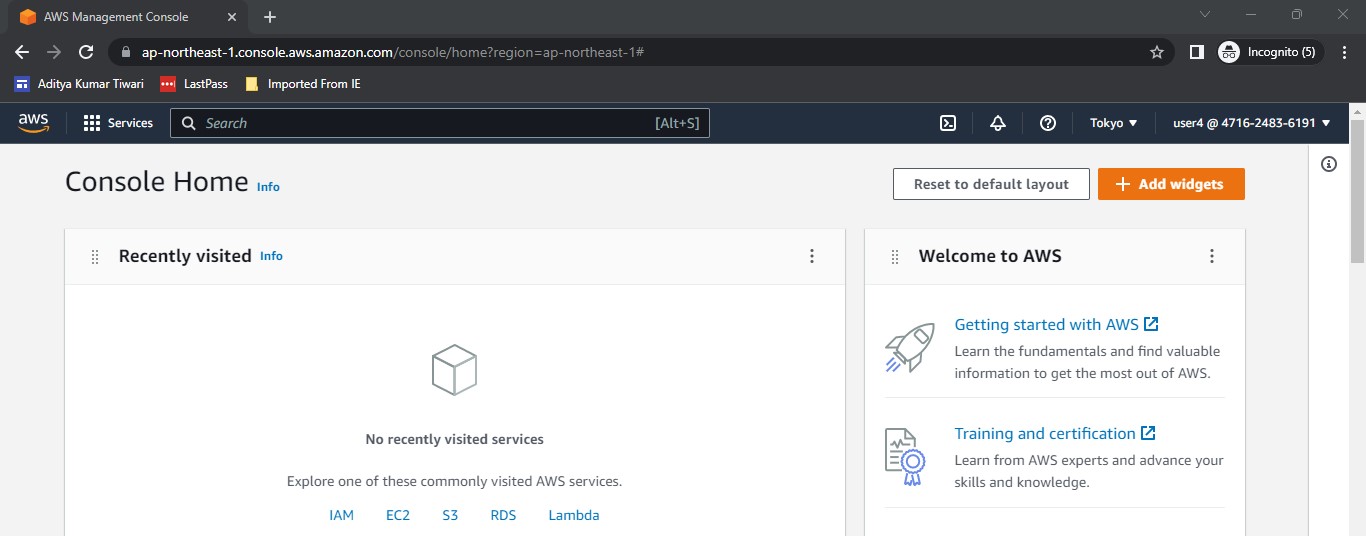
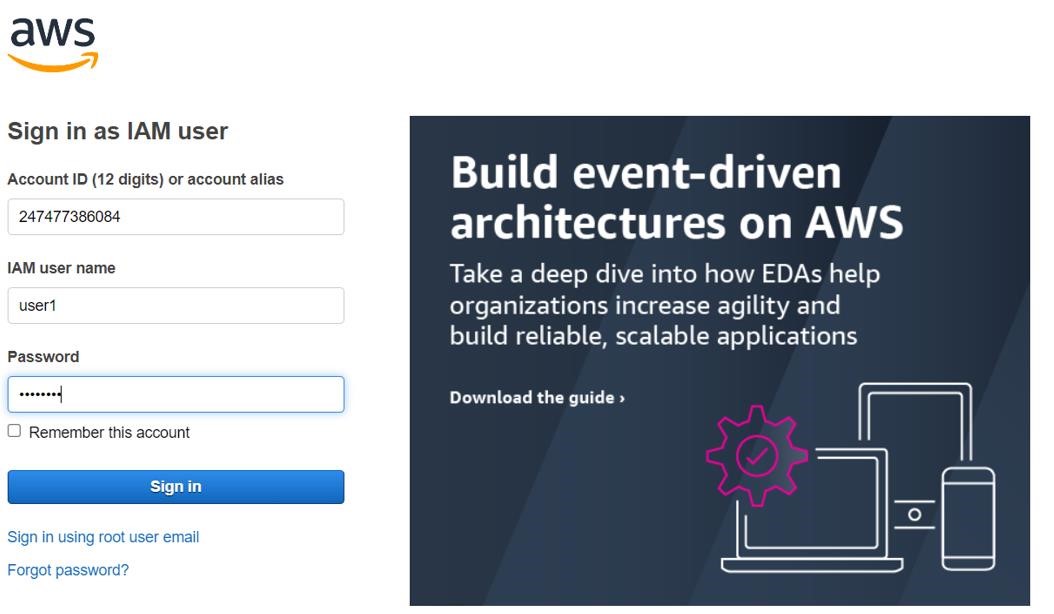


**Step 8:** After creation of the user , pop shows a link used to login as the user



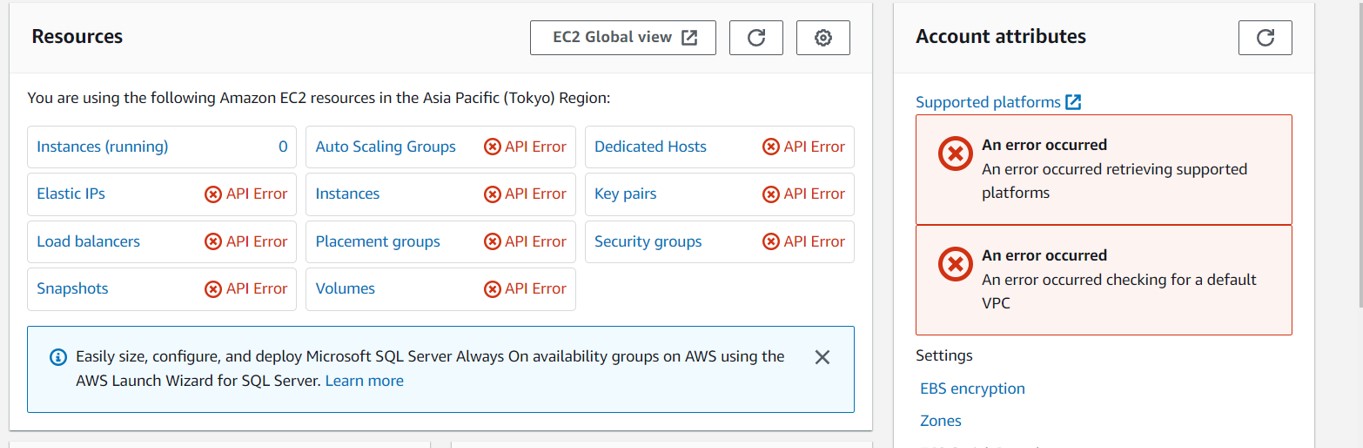
**Step 9:** Open an incognito tab and login as the new user





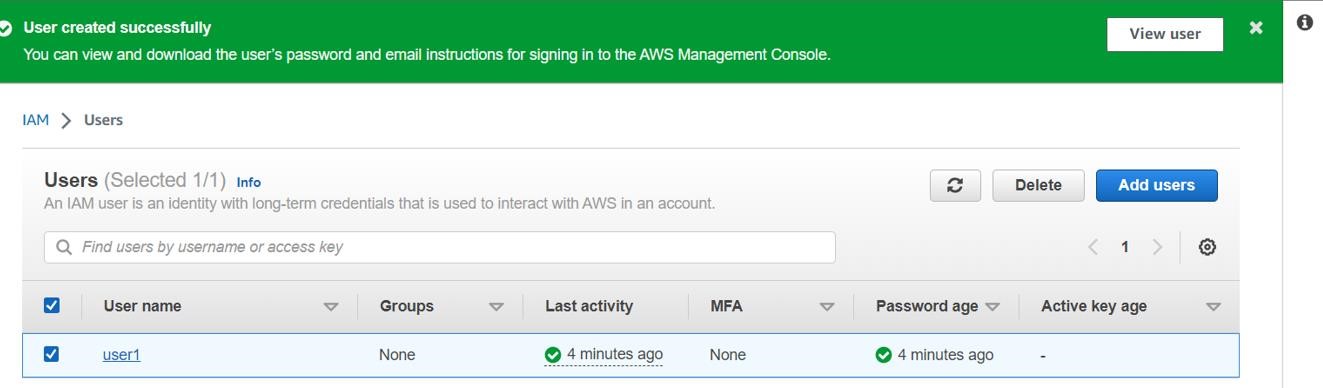
**Step 10:** To ensure that the user only has readonly policies, we check S3 & ec2



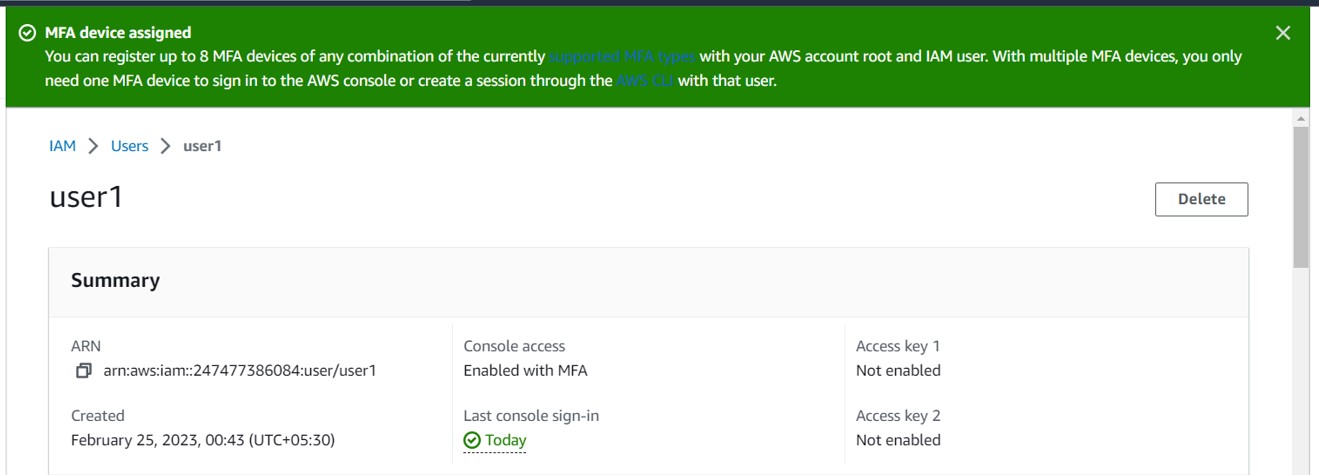
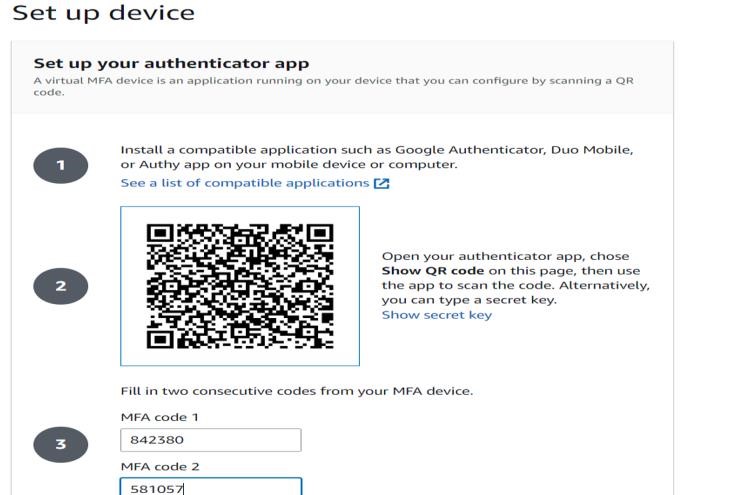


**ii) Show how to use MFA in IAM?**

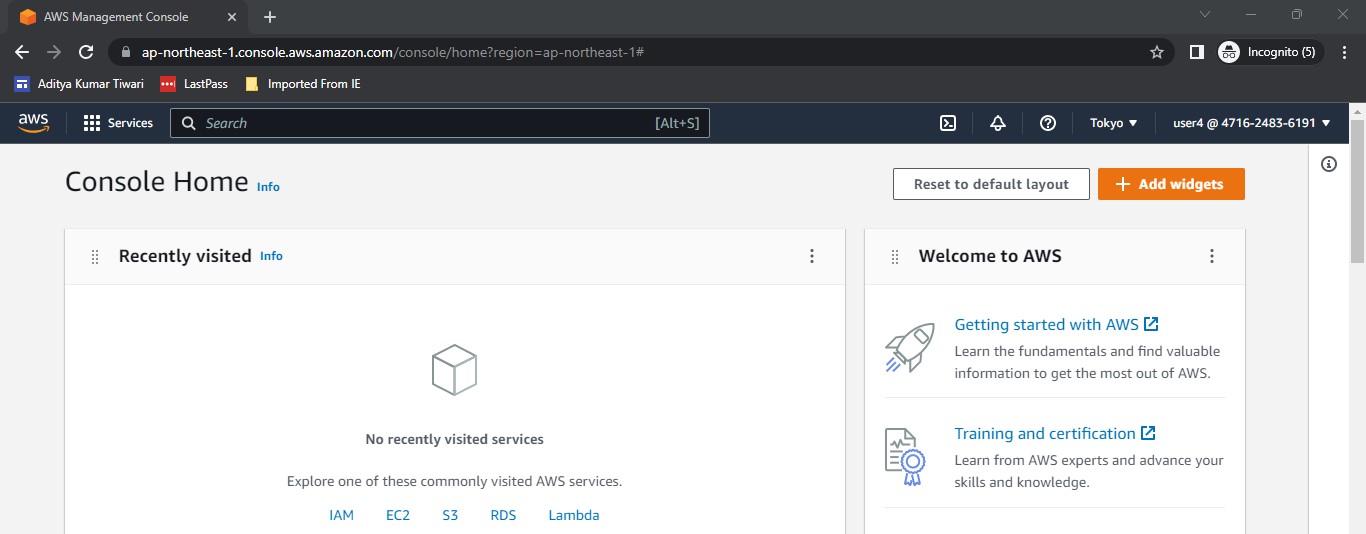
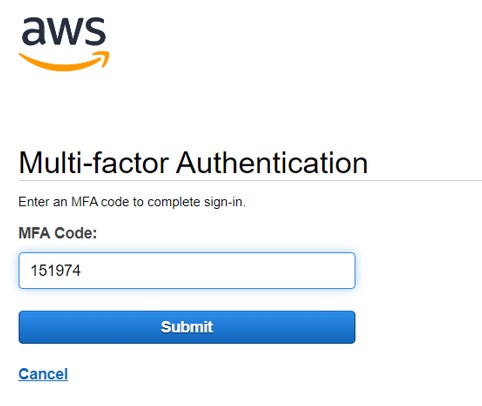
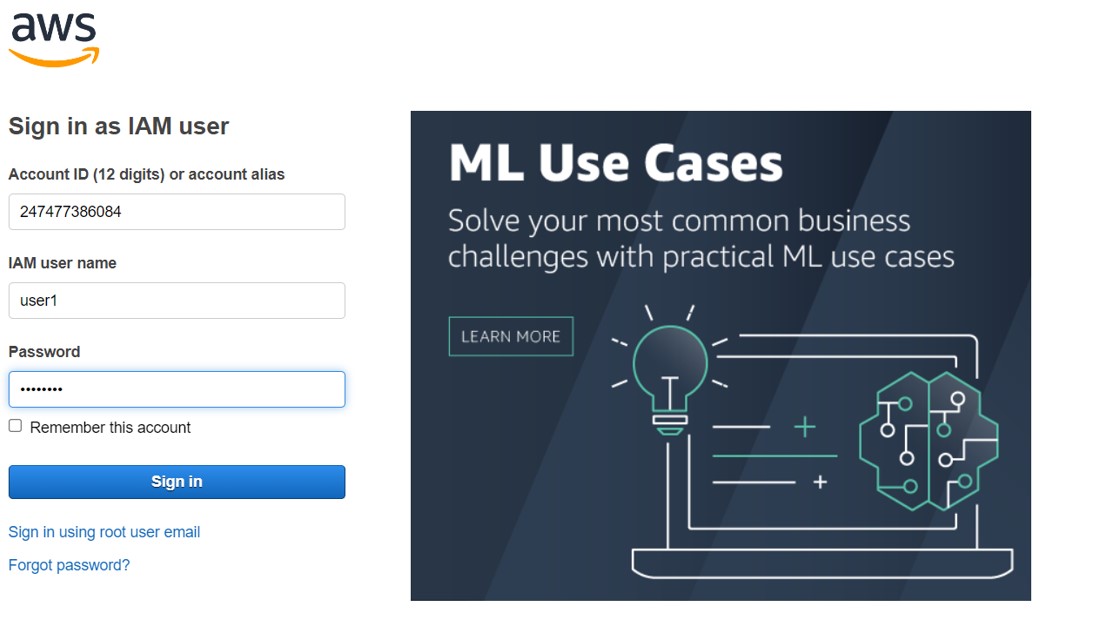
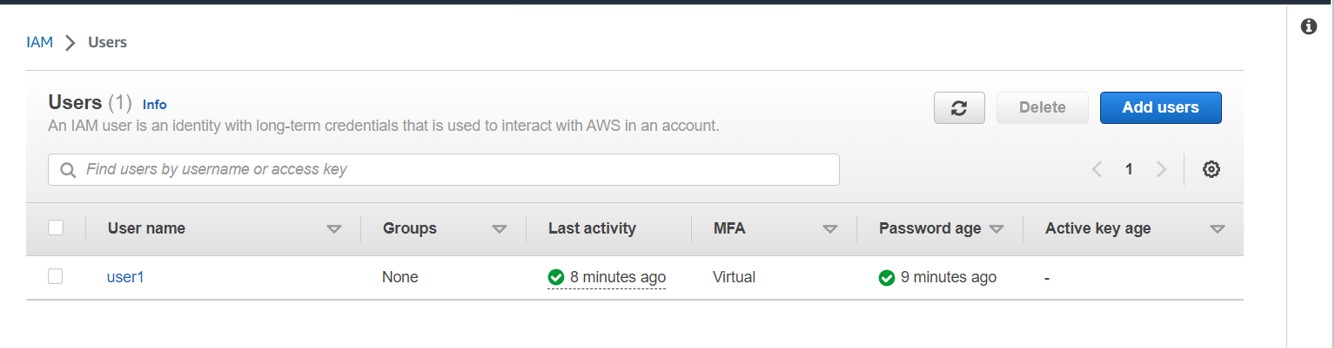
**Step 11:** Now to enable MFA on the user created we login as the root user & view user & goto add MFA



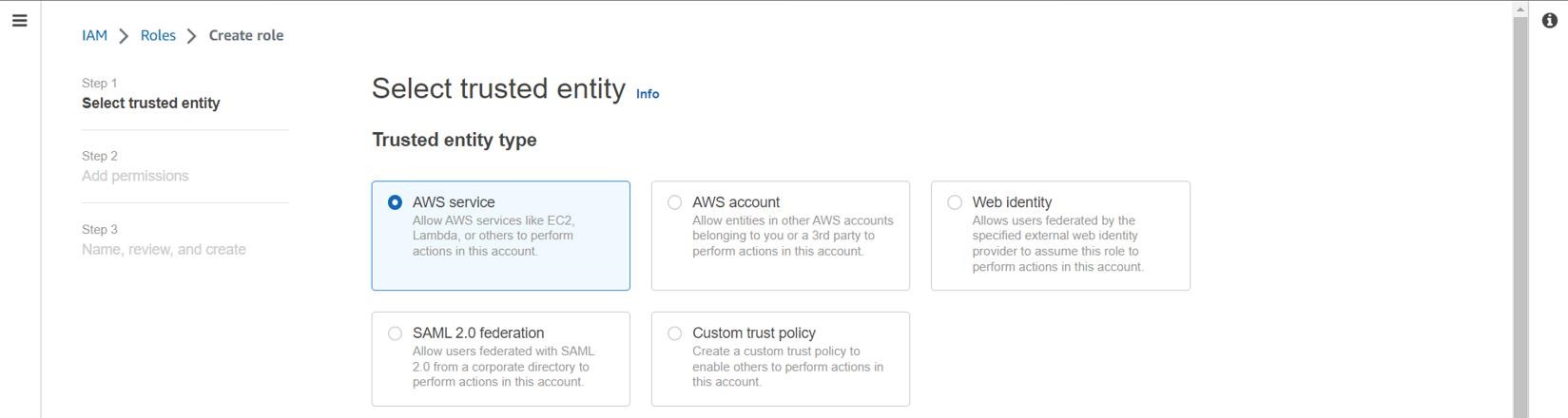
**Step 12:** Add the MFA device and enter the generated codes



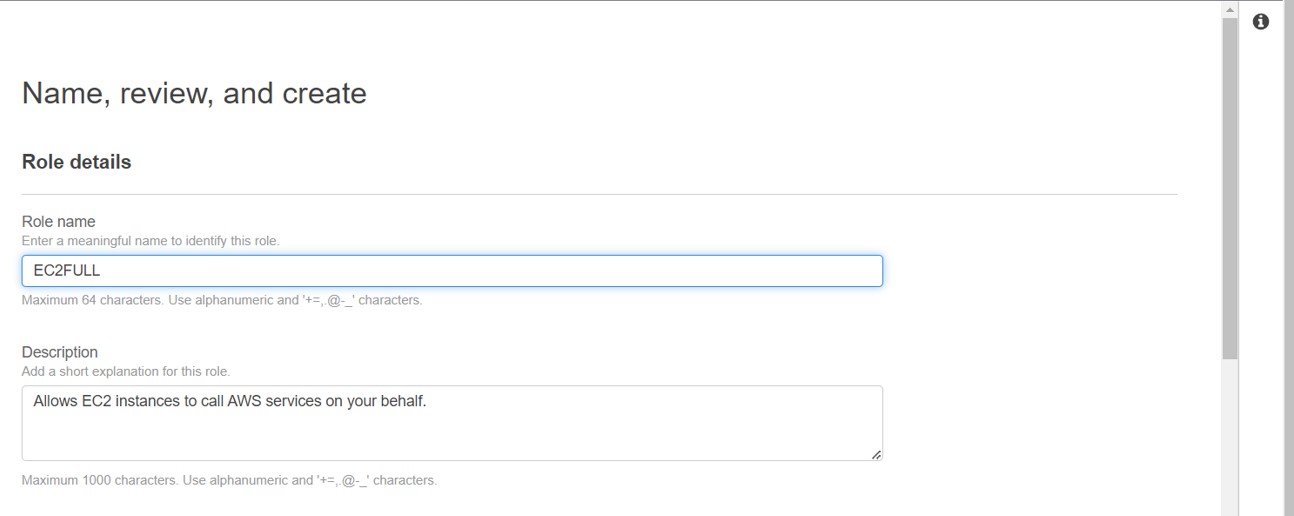
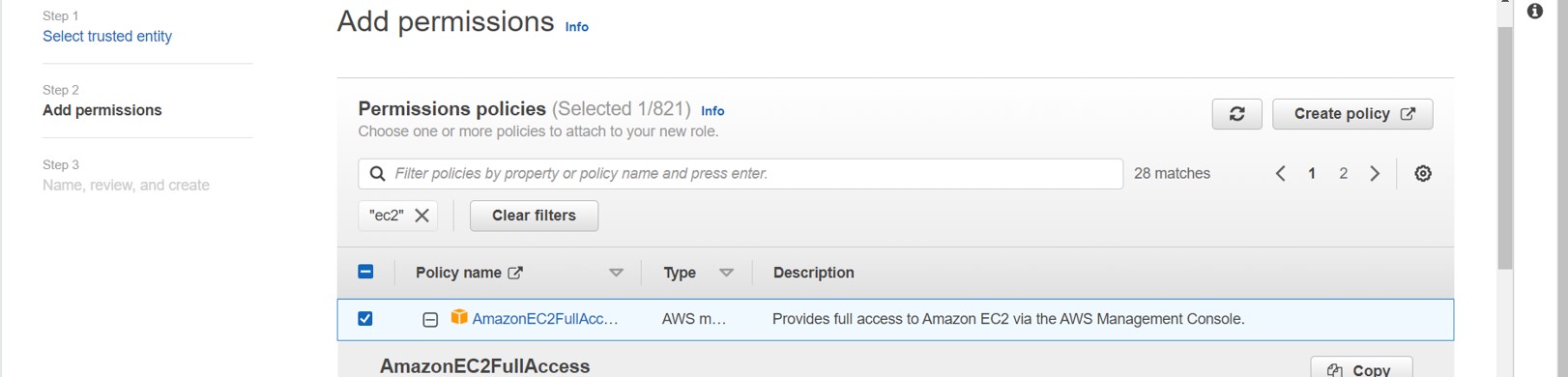
**Step 13:** Login as the IAM user again



**Step 14:** For creating roles , goto IAM->Roles->Create roles



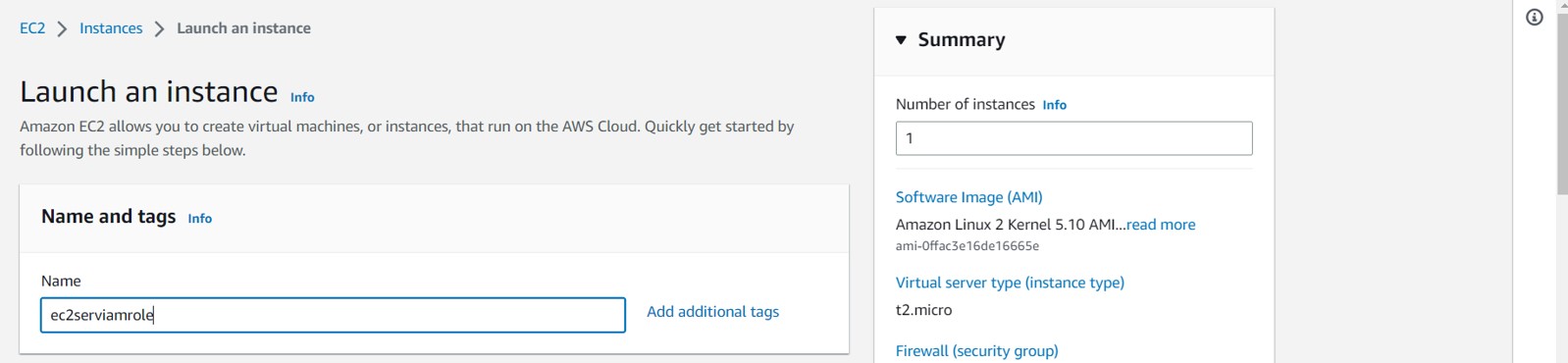
**Step 15:** In add permissions, add a fullec2 access permission



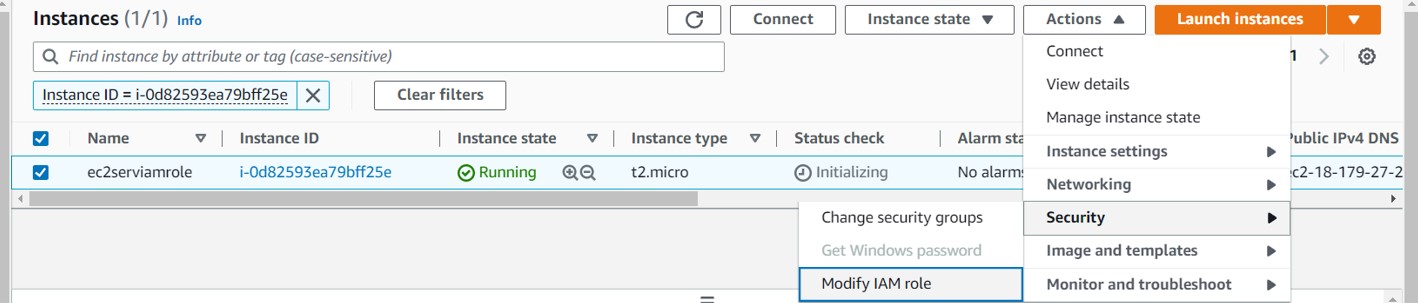
**Step 16:** Click on create role



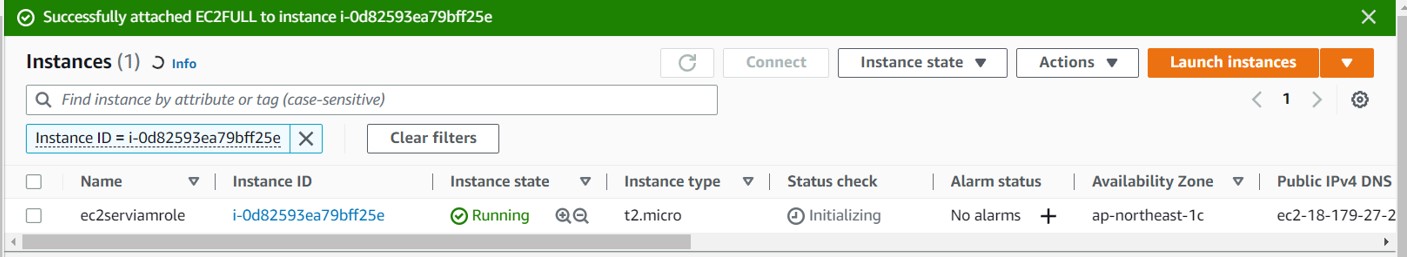
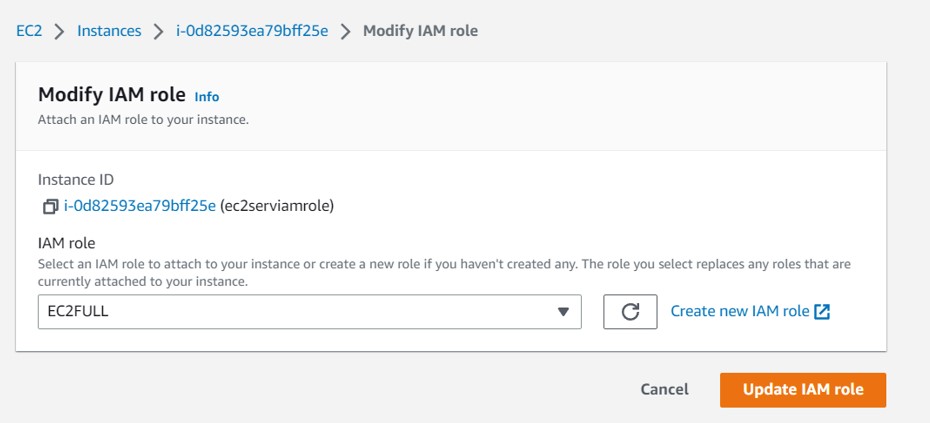
**Step 17:** To attach the IAM role , we create an ec2 instance



**Step 18:** Select the instance and goto actions->security->modify IAM role

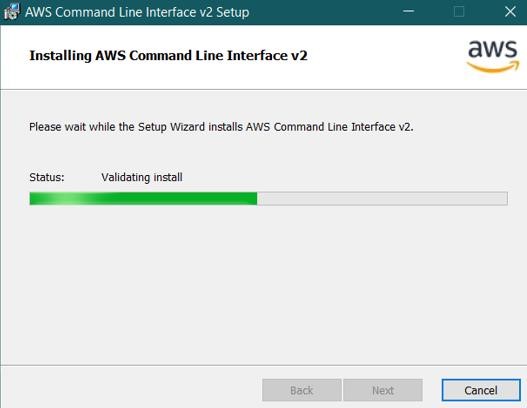
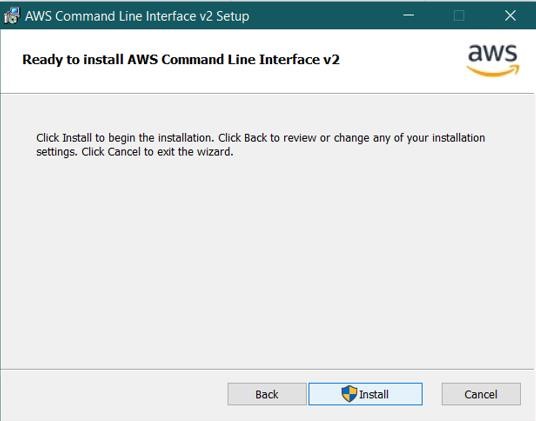
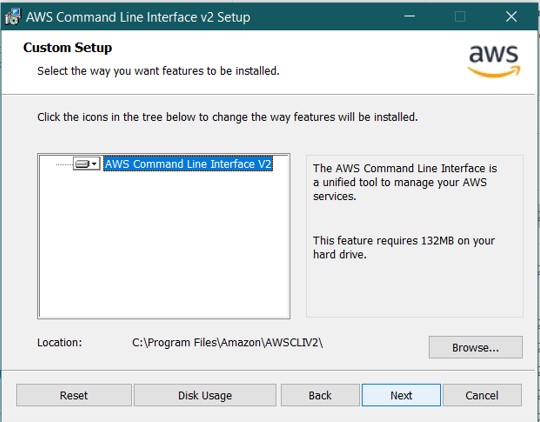
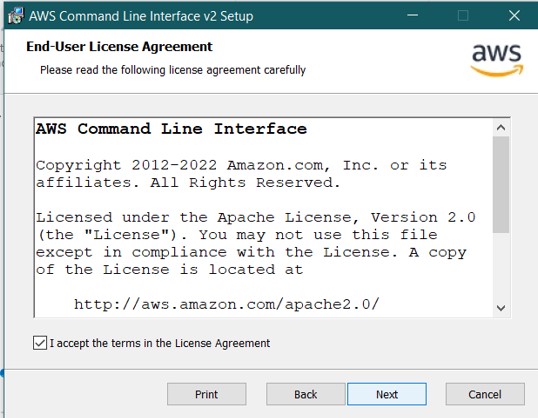
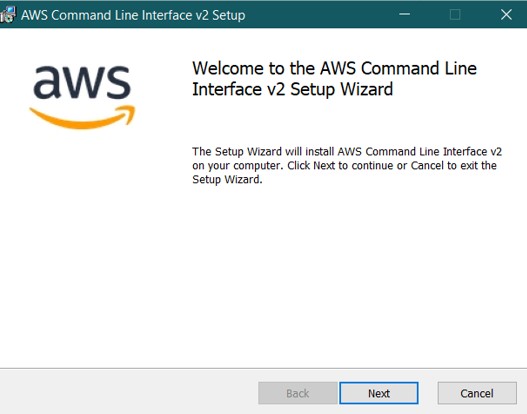
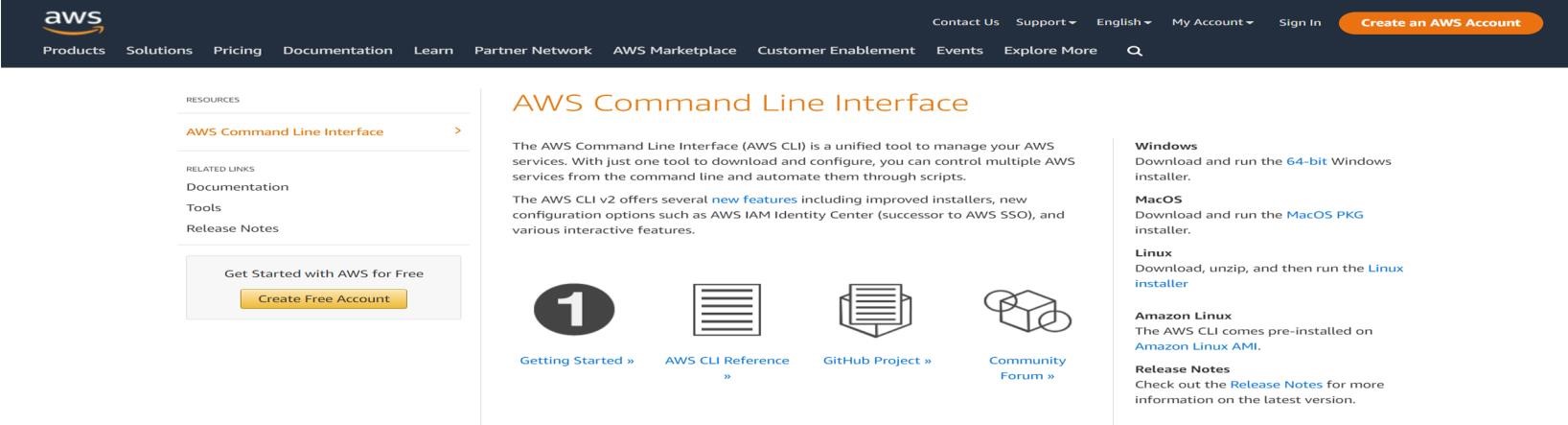


**Step 19:** Select the IAM role and update

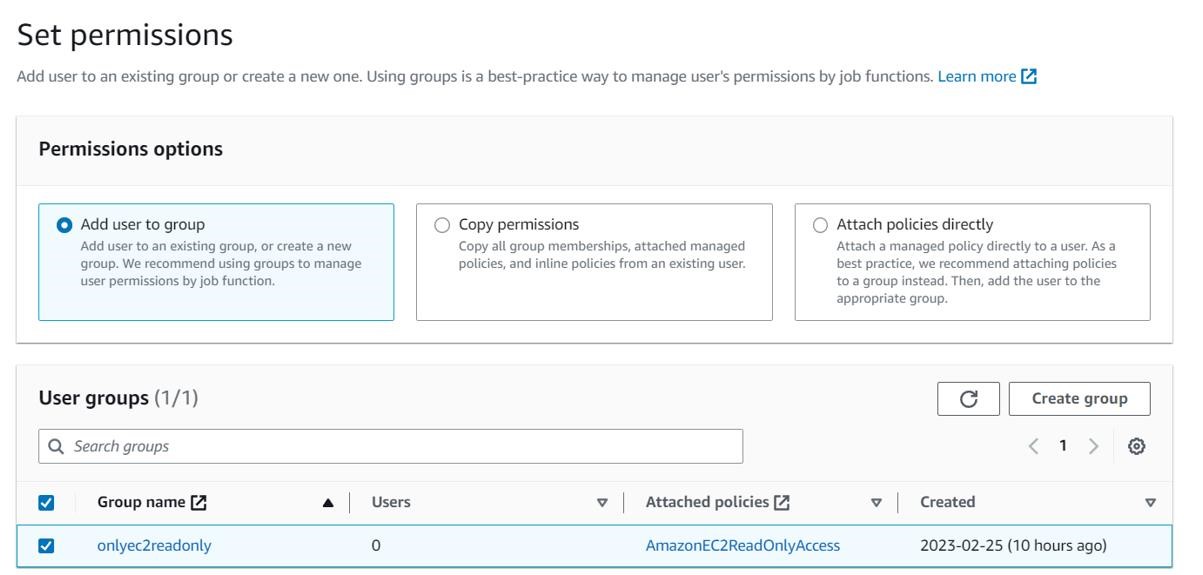
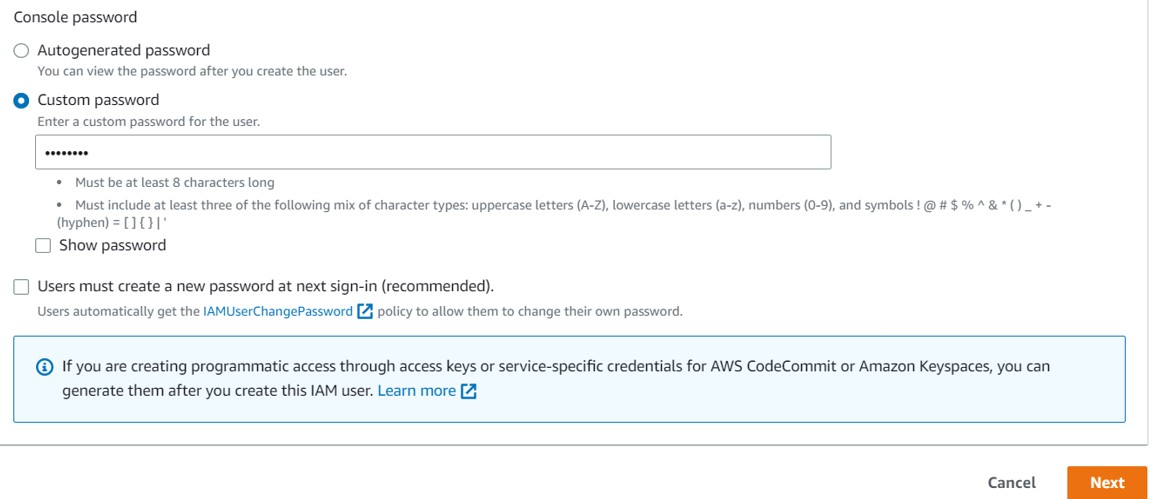
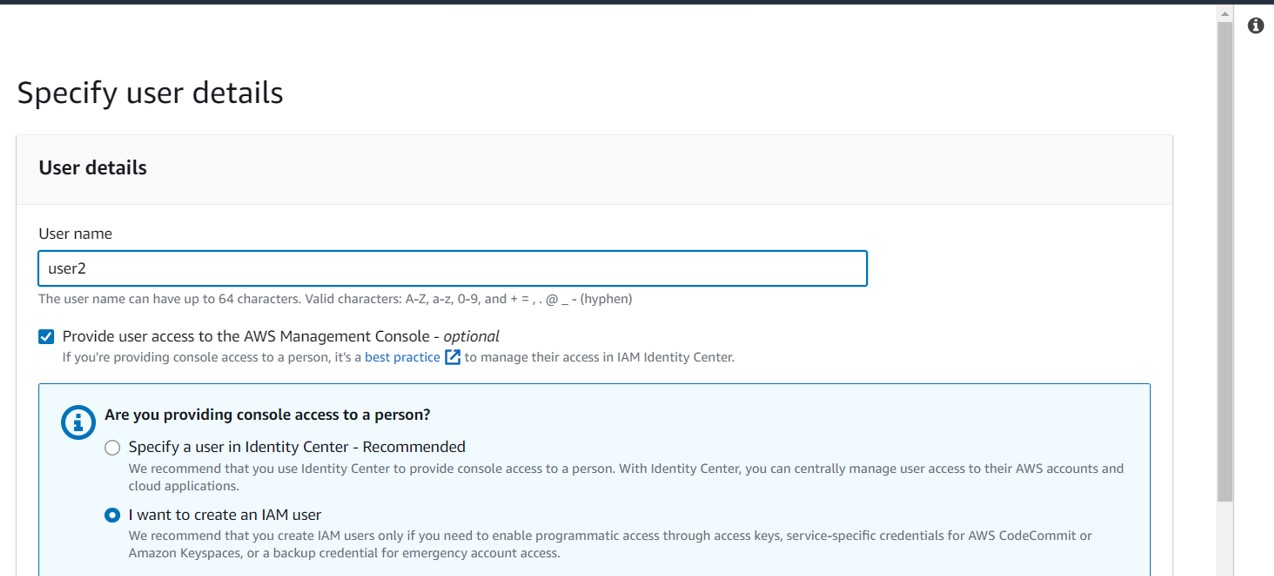


**ii)**

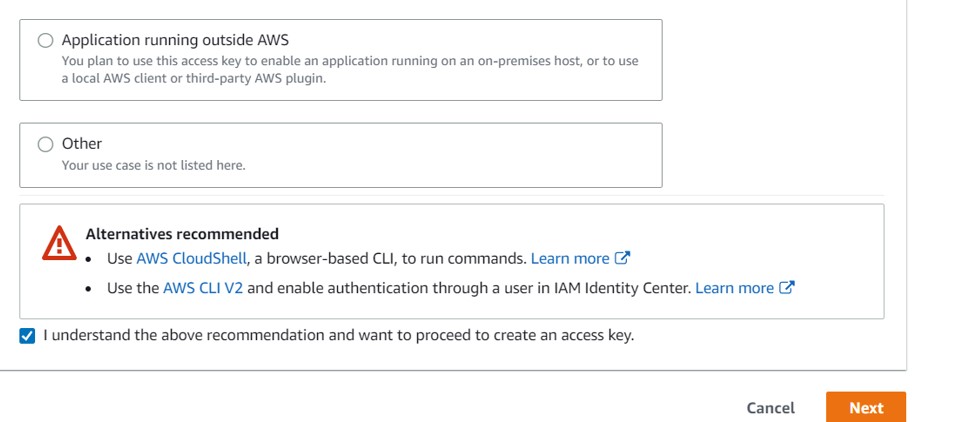
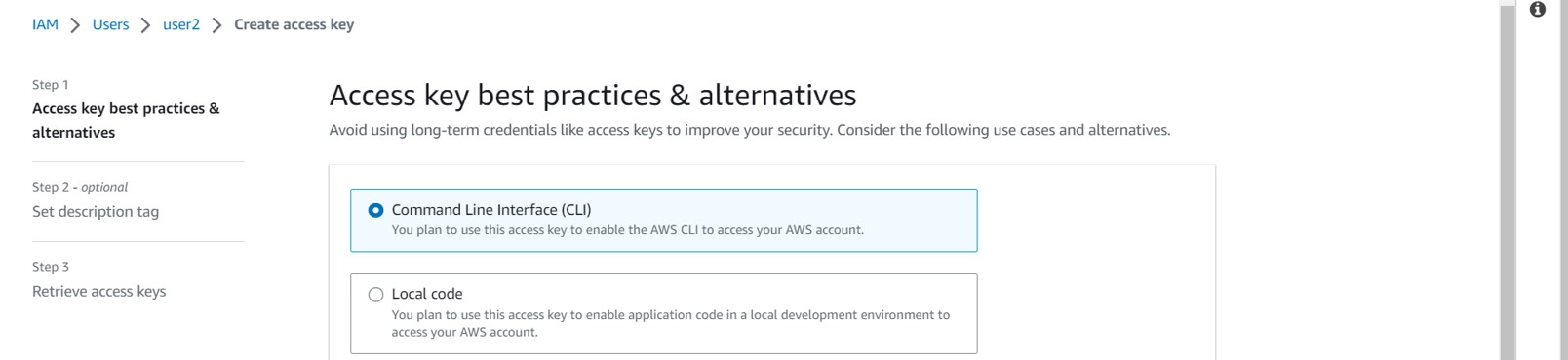
**Step 1:** Install the Amazon CLI v2 from aws website



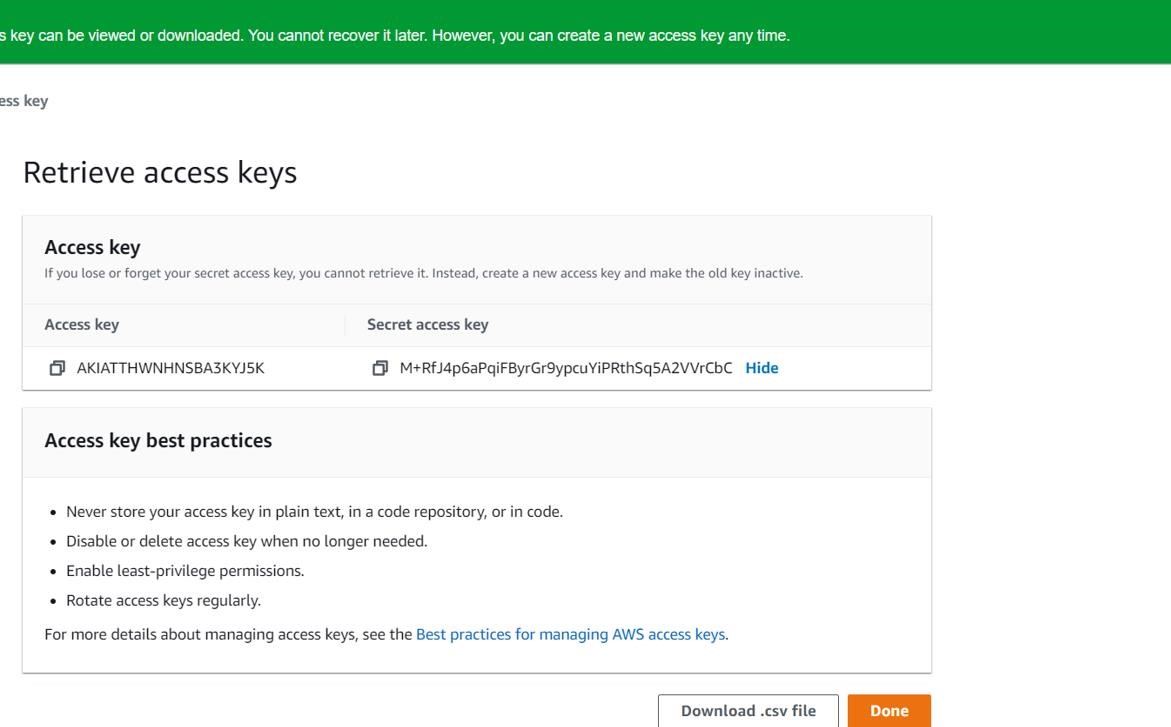
**Step 2:** Goto the root user and create a new user [here user2]

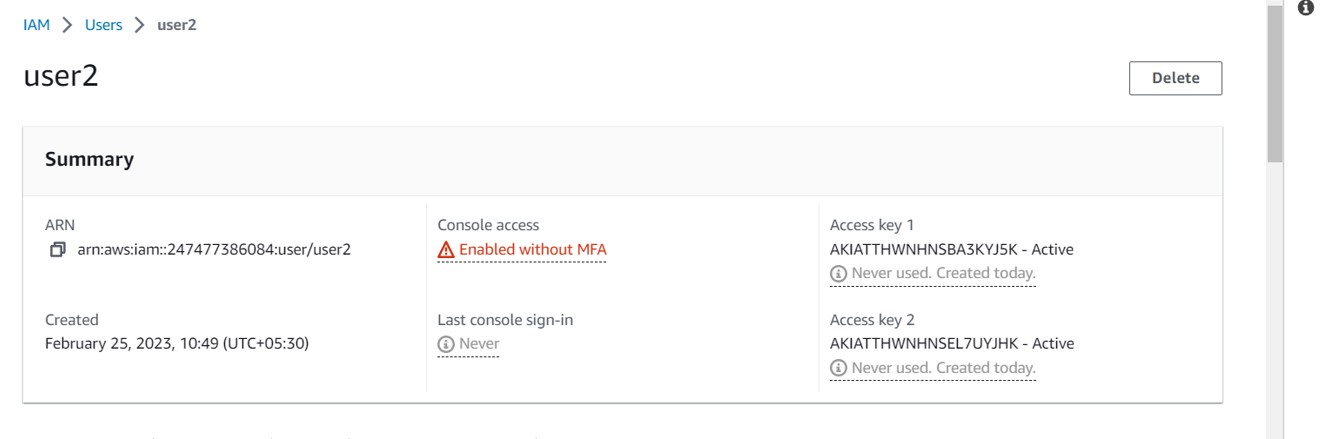


**Step 3:** Goto user2 and click on “Access key” under “Security credentials”

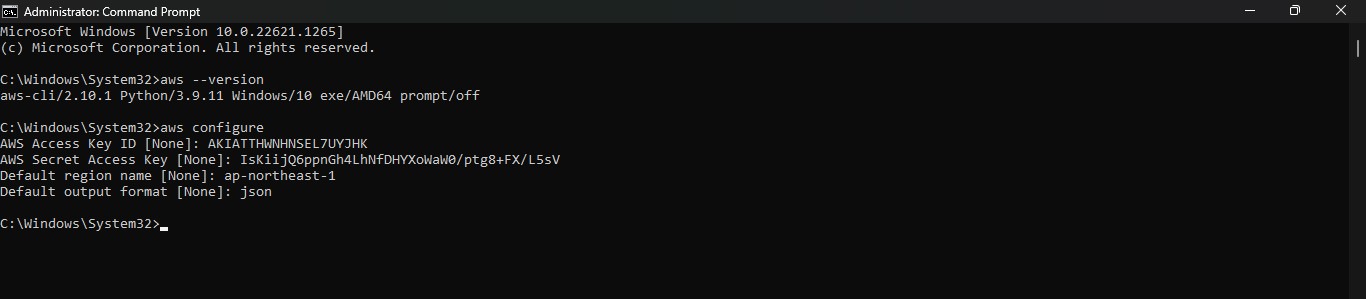


**Step 4:** Create the access key



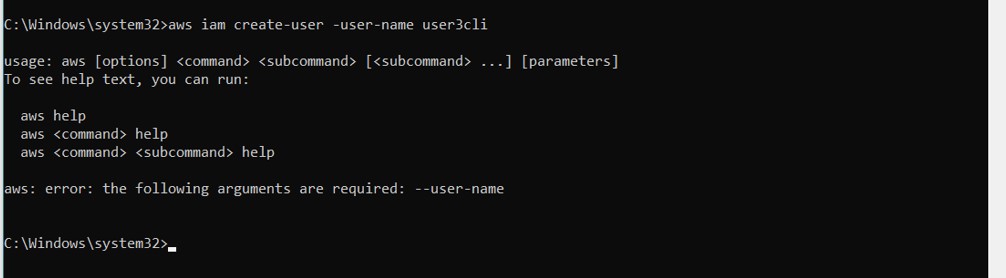


Step 5: In the command line , use the command aws configure

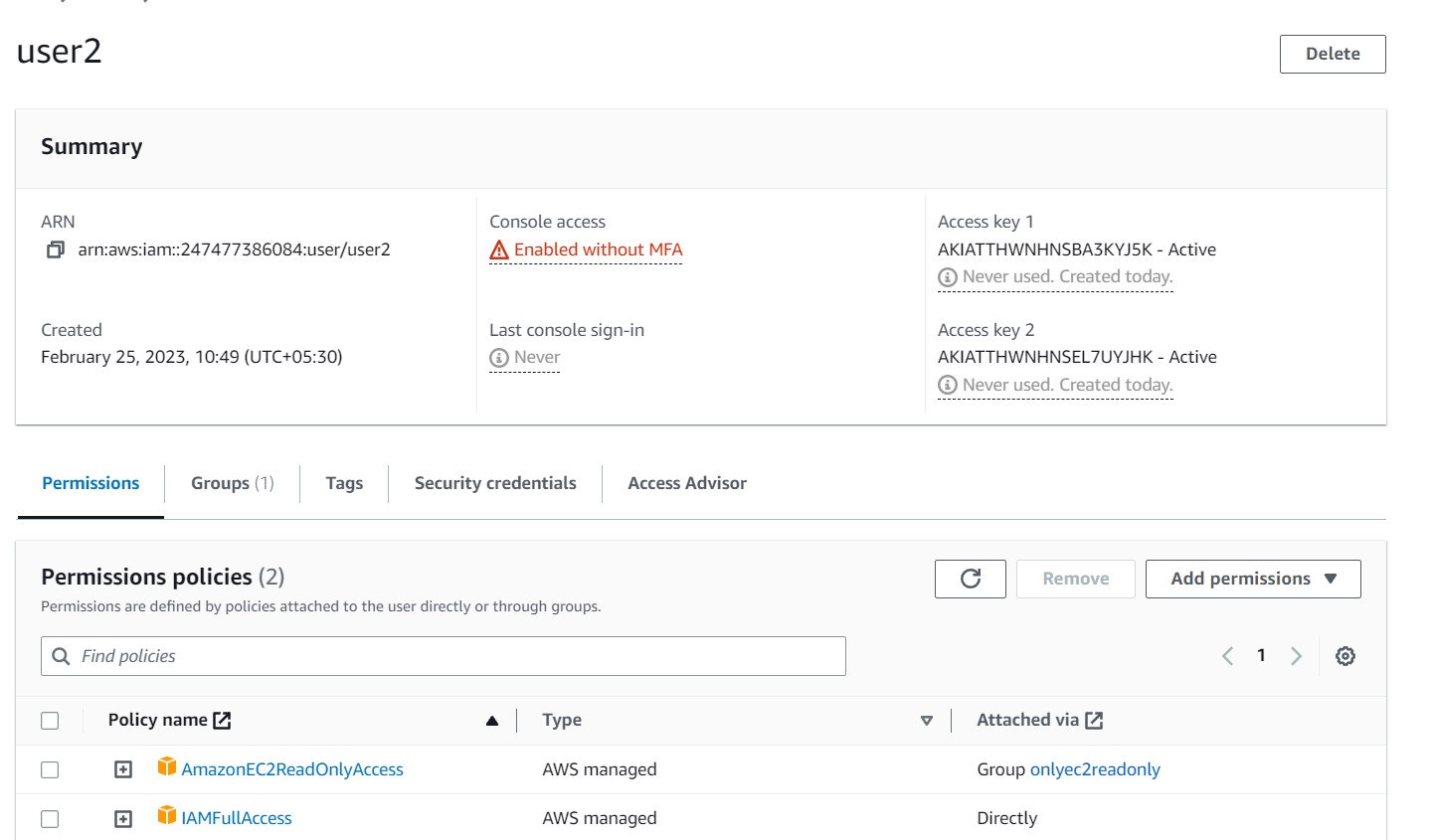


Provide the access key, secret access key ,region and output format as shown

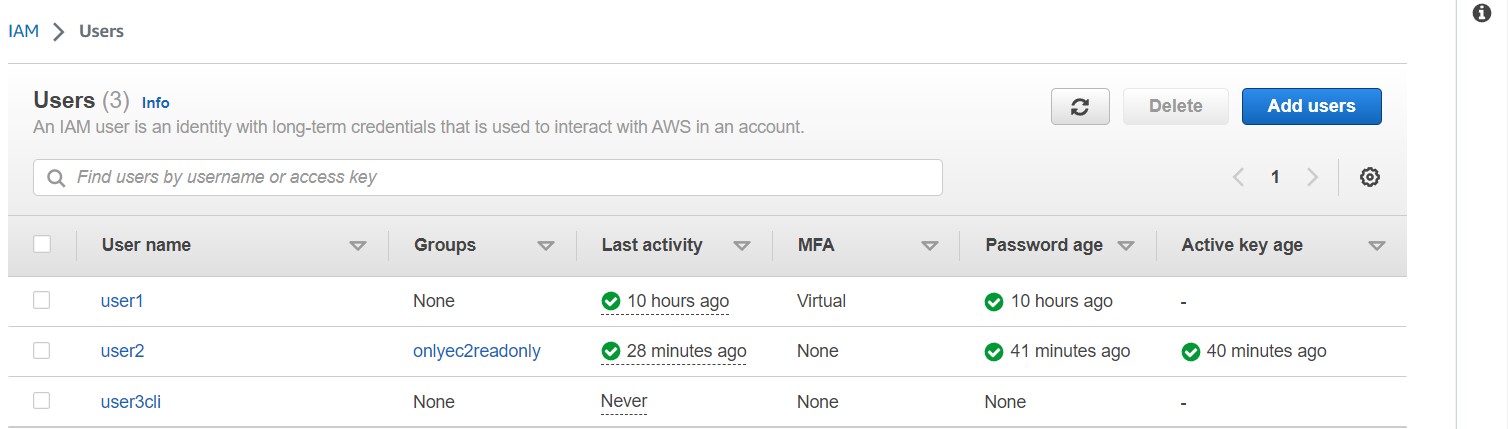
**Step 6:** Use the command to create the iam user aws iam create-user –user-name user3cli



Note :- If you encounter an error in the previous step ,goto root user and the user 2 and allow all IAM permissions by directly attaching the policy



**Step 7:**Goto Iam-> users , and user3 would have been created



**QUESTIONS**

**Ques1:** Explain the Features of IAM in detail?

**Ans:** IAM (Identity and Access Management) is a service provided by AWS (Amazon Web Services) that enables users to manage access to AWS resources securely. Some of the key features of IAM are:

* User Management: IAM allows users to create and manage user accounts, including assigning permissions and access levels to each user.
* Access Management: IAM allows users to define and manage policies that control access to AWS resources based on user roles, groups, or specific permissions.
* Integration with AWS Services: IAM integrates with various AWS services, allowing users to control access to these services based on IAM policies.
* Multi-Factor Authentication (MFA): IAM provides support for MFA, which provides an extra layer of security to user accounts by requiring a second form of authentication in addition to a password.
* Identity Federation: IAM allows users to set up identity federation, which enables users to access AWS resources using their existing credentials from other identity providers such as Active Directory, LDAP, or Facebook.
* Audit and Monitoring: IAM provides detailed audit logs and monitoring features, allowing users to track access and changes to AWS resources and policies.

Overall, IAM provides users with a comprehensive set of features for managing access to AWS resources in a secure and efficient manner.

**Ques 2:** What is the difference between Root account, User account, Roles and Policies?

**Ans:** In AWS (Amazon Web Services), Root accounts, User accounts, Roles, and Policies are different ways to manage access to AWS resources, with some key differences:

* Root account: The Root account is the initial account created when signing up for an AWS account. It has complete access to all AWS services and resources, including the ability to create and delete AWS accounts. The Root account is typically used to set up the initial configuration of an AWS account, and it should be used sparingly.

* User account: User accounts are created by the Root account and are used to manage access to AWS resources. Each User account is associated with an IAM (Identity and Access Management) User, which has its own username and password. User accounts can be assigned permissions to access specific AWS resources based on policies defined in IAM.

* Roles: Roles are similar to User accounts but are not associated with a specific user. Instead, Roles are created and assigned to AWS services or resources, allowing those services or resources to access other AWS resources or services on behalf of users or applications. Roles are often used in conjunction with services such as EC2 or Lambda, where applications running on those services require access to other AWS resources.

* Policies: Policies are used to define permissions and access levels for User accounts and Roles. Policies are created in IAM and can be attached to User accounts or Roles to grant specific permissions to access AWS resources. Policies are defined in JSON format and can be customized to provide granular access control to AWS resources.

Overall, the key difference between Root accounts, User accounts, Roles, and Policies is the level of access and control they provide over AWS resources. Root accounts have full access and control over all AWS resources, while User accounts, Roles, and Policies provide more granular access control and can be customized to meet specific access requirements.

**Ques 3:** What is the difference between the Management Console and Programmatic access?

**Ans:** The Management Console and Programmatic access are two ways to access and manage AWS (Amazon Web Services) resources, with some key differences:

* Management Console: The Management Console is a web-based interface that provides users with a graphical user interface (GUI) for managing AWS resources. It allows users to interact with AWS services using a point-and-click interface, making it easy to perform tasks such as creating EC2 instances, managing S3 buckets, or configuring security settings. The Management Console is primarily used by administrators and developers who are more comfortable with a GUI interface.

* Programmatic Access: Programmatic access to AWS involves using APIs (Application Programming Interfaces) to interact with AWS resources. This type of access allows users to automate tasks and manage AWS resources programmatically using scripts or applications. Programmatic access requires knowledge of programming languages, such as Python, Java, or Ruby, and is often used by developers and DevOps engineers to manage AWS resources at scale.

Overall, the key difference between the Management Console and

Programmatic access is that the former is a graphical interface that provides an intuitive way to manage AWS resources, while the latter provides more control and flexibility, allowing users to automate and manage AWS resources programmatically. Depending on the use case, users may choose to use one or both methods of accessing and managing AWS resources.

**Ques 4:** What is MFA in AWS?

**Ans:** MFA stands for Multi-Factor Authentication, which is an extra layer of security that AWS (Amazon Web Services) provides to its users. MFA requires users to provide two or more authentication factors to access their AWS accounts.

In AWS, the first factor is usually a password or access key, and the second factor is a unique one-time code generated by a physical or virtual MFA device.

When MFA is enabled, users are required to enter the code generated by the MFA device after entering their password or access key, thus providing an additional layer of protection against unauthorized access.

MFA can be applied to various levels of access within an AWS account, including access to the AWS Management Console, AWS CLI, or API calls. AWS supports various types of MFA devices, including hardware devices such as YubiKey, software-based virtual MFA devices, and SMS-based MFA.

Enabling MFA on your AWS account is highly recommended as it greatly increases the security of your AWS resources and data.