**Task Description**

The task includes -

* Creating AWS S3 bucket and bucket policy
* Creating IAM users, user groups and policies and their working and implementation

About **AWS IAM** :

AWS Identity and Access Management (IAM) safeguards access to AWS services and resources and it creates and manages AWS users and groups and uses permissions to grant or deny access to AWS services.

**Benefits of IAM :**

● It can scale securely over your data with superior visibility and control.

● Automates and reduces risks like human config errors with deeply integrated solutions/services.

● Inherit the most comprehensive security and compliance controls.

● Build with the highest standards for privacy and data security.

**Components of IAM :**

● **IAM Users**- Users are AWS entities that represent the person or application that interacts with AWS, AWS users have credentials (for example, name). The root user of an AWS account is not the same as an IAM user with administrator access.

● **IAM Groups**- A set of IAM users is referred to as an IAM user group. A user group allows you to specify permissions for many users, making it easier to manage those users’ permissions. In a resource-based policy, a user group can’t be designated as a principal. A user group is a technique to apply policies to a

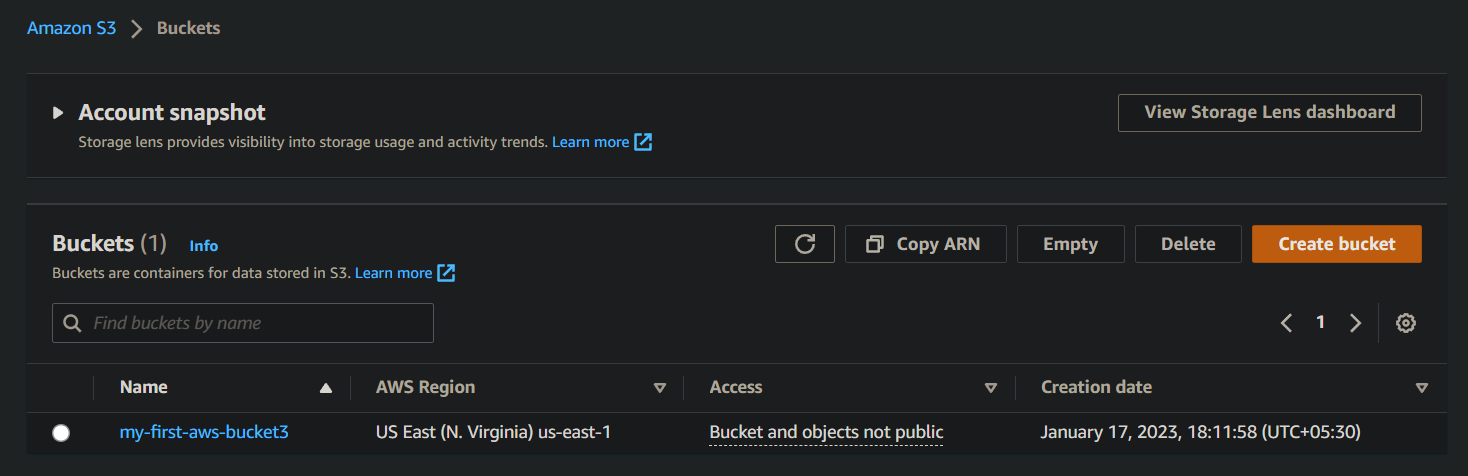
group of people all at once.

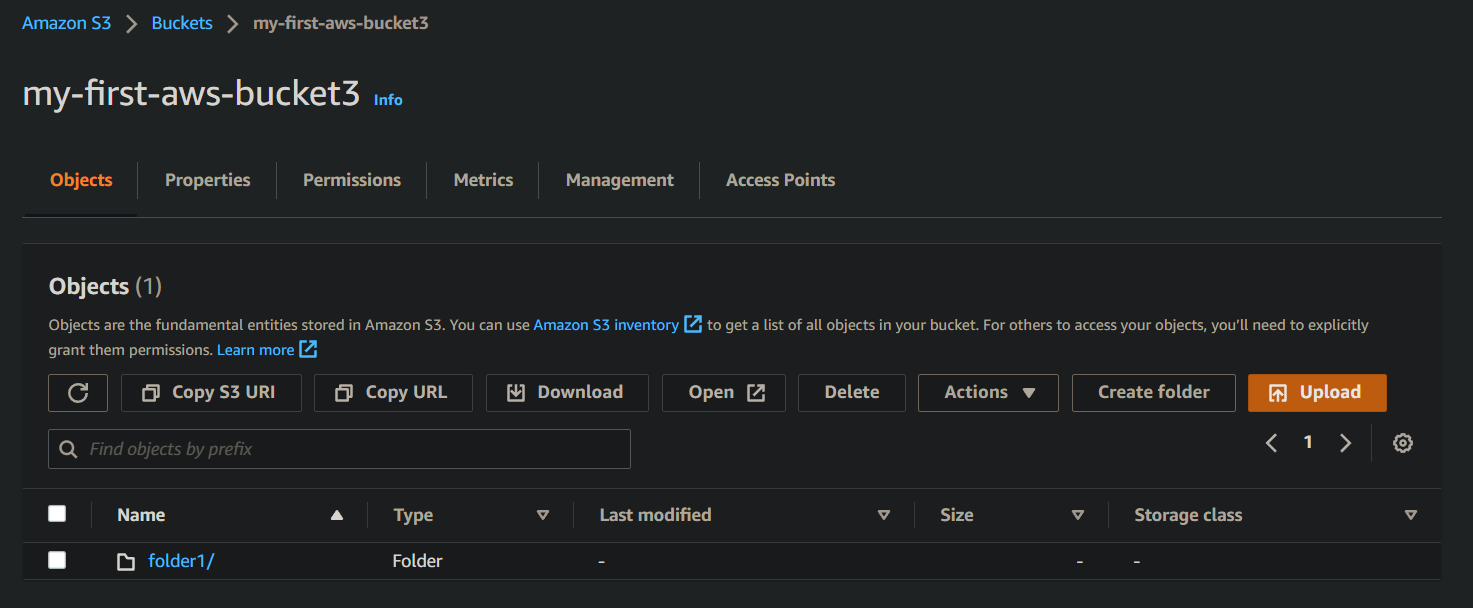
● **IAM Roles**- An IAM role, like an IAM user, is an AWS identity with permission policies that govern what the identity can and cannot do in AWS. There are no long-term credentials connected with a role. Instead, when you take on a role,you’re given temporary security credentials for the duration of your role session. Users, applications, and services that don't ordinarily have access to your AWS resources can be given access through roles.

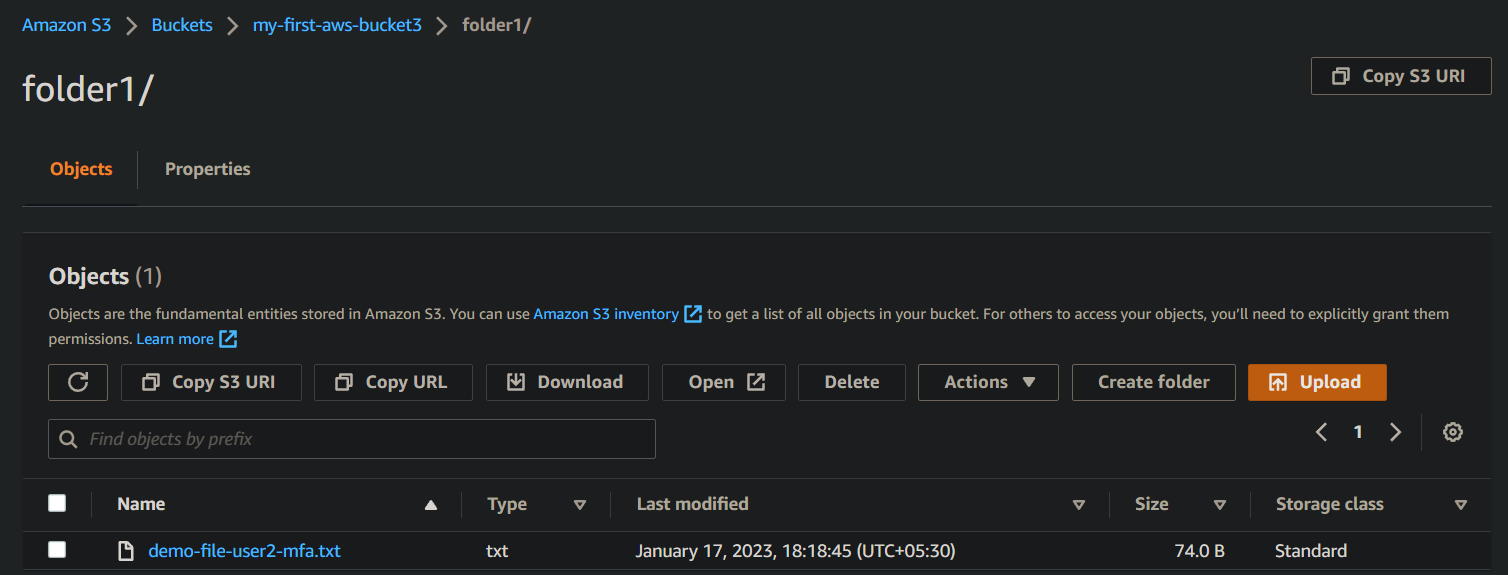
● **IAM Policies**- Policies are created and linked to IAM identities (like users, groups, or roles) or AWS resources to manage access in AWS. A policy is an object that defines the rights of an identity or resource when it is associated with it. AWS examines them when an IAM principal submits a request. The policy permission decides whether the request is approved or rejected. Most policies are saved as JSON documents on AWS

**Screenshots of the tasks :**

* Creating AWS S3 bucket and creating folder “**folder1**” inside this bucket and uploading one text document “**demo-file-user2-mfa”**

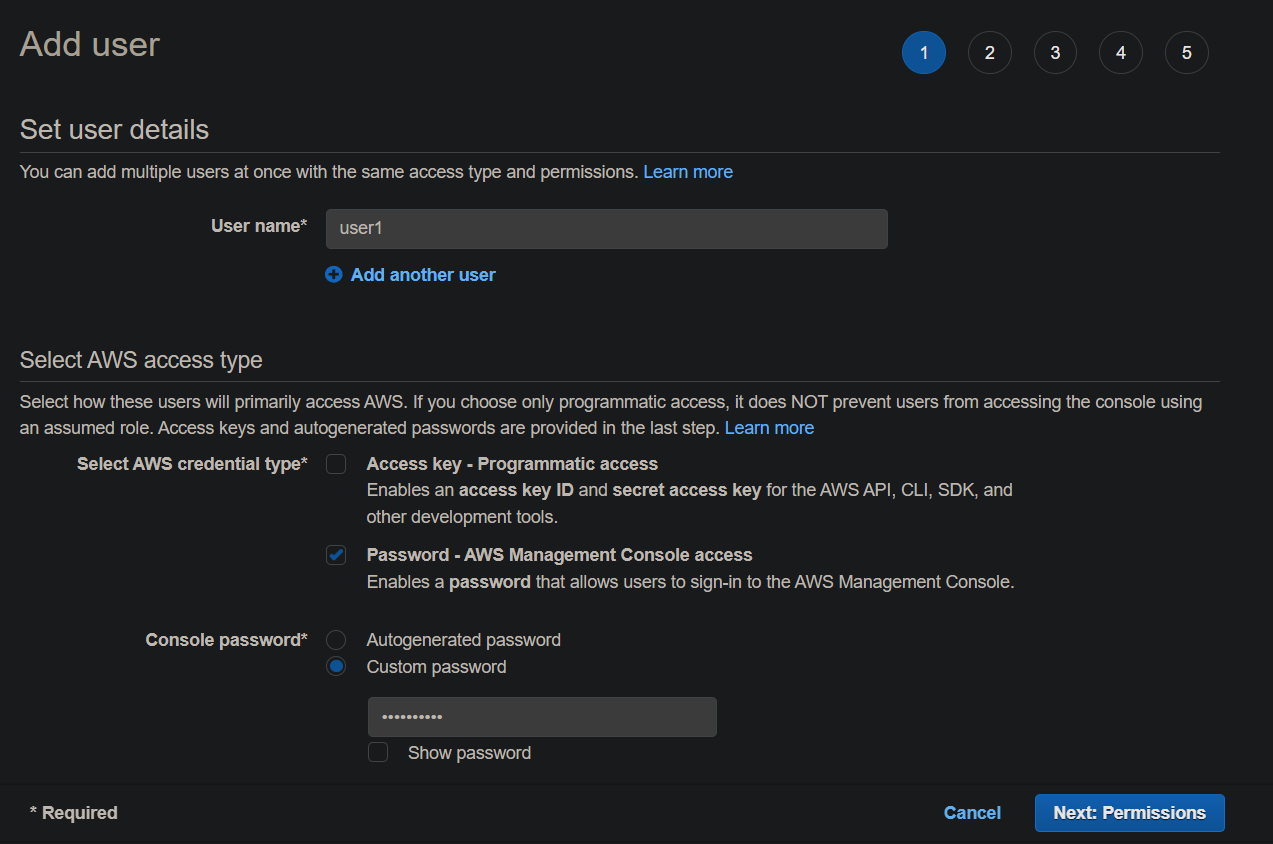


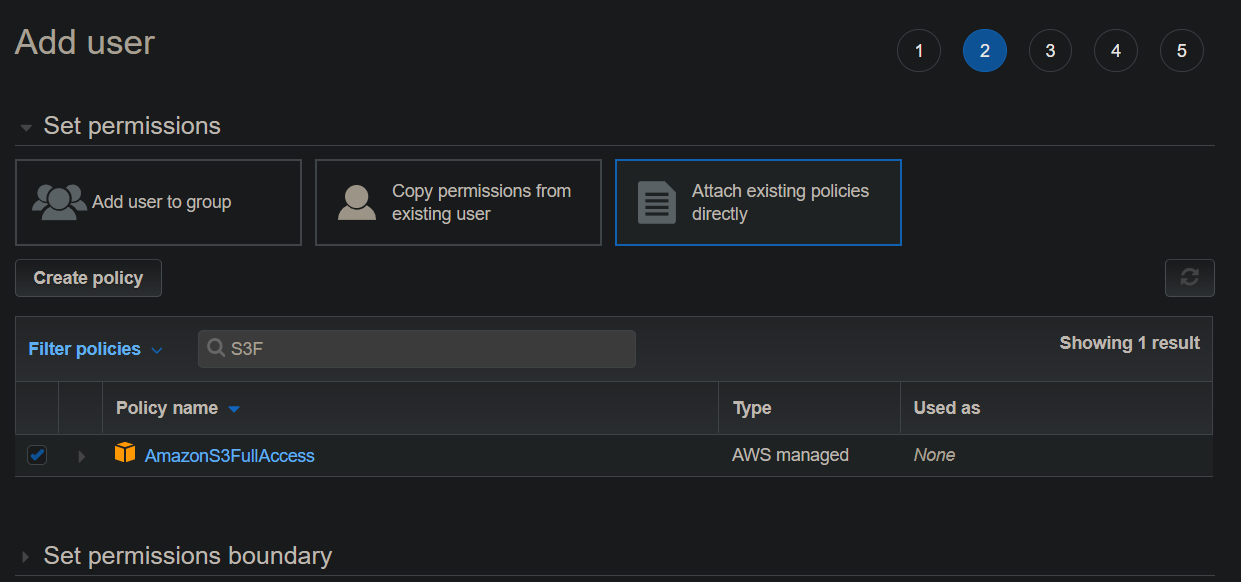


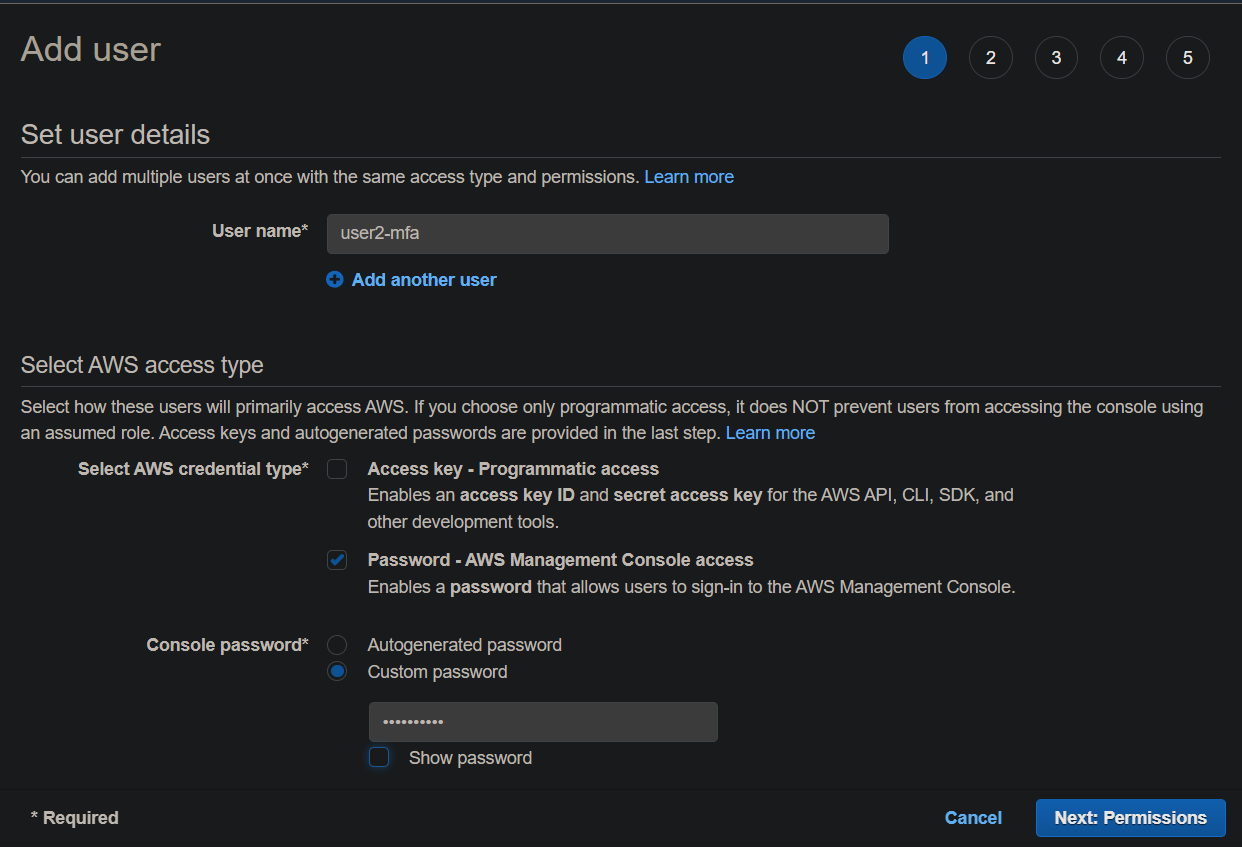


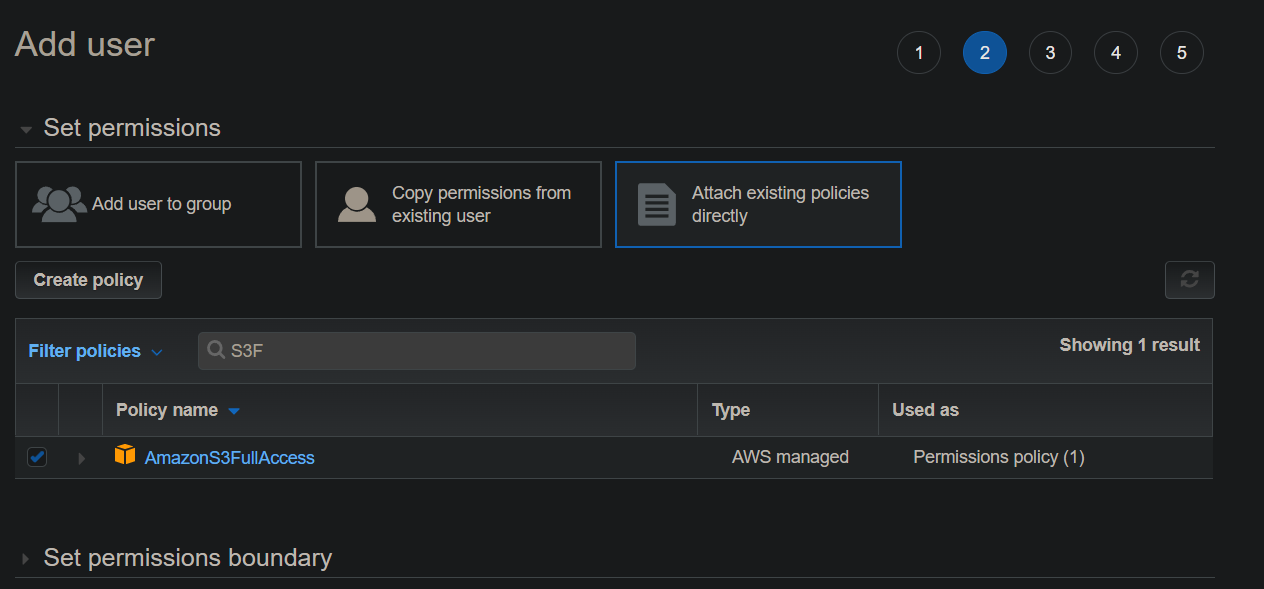
* Creating IAM users using AWS IAM management console :

Creating user “**user1”** by clicking on Add users and Then assign existing policies directly. Similarly creating another user “**user2-mfa”** with the same configuration as user1.

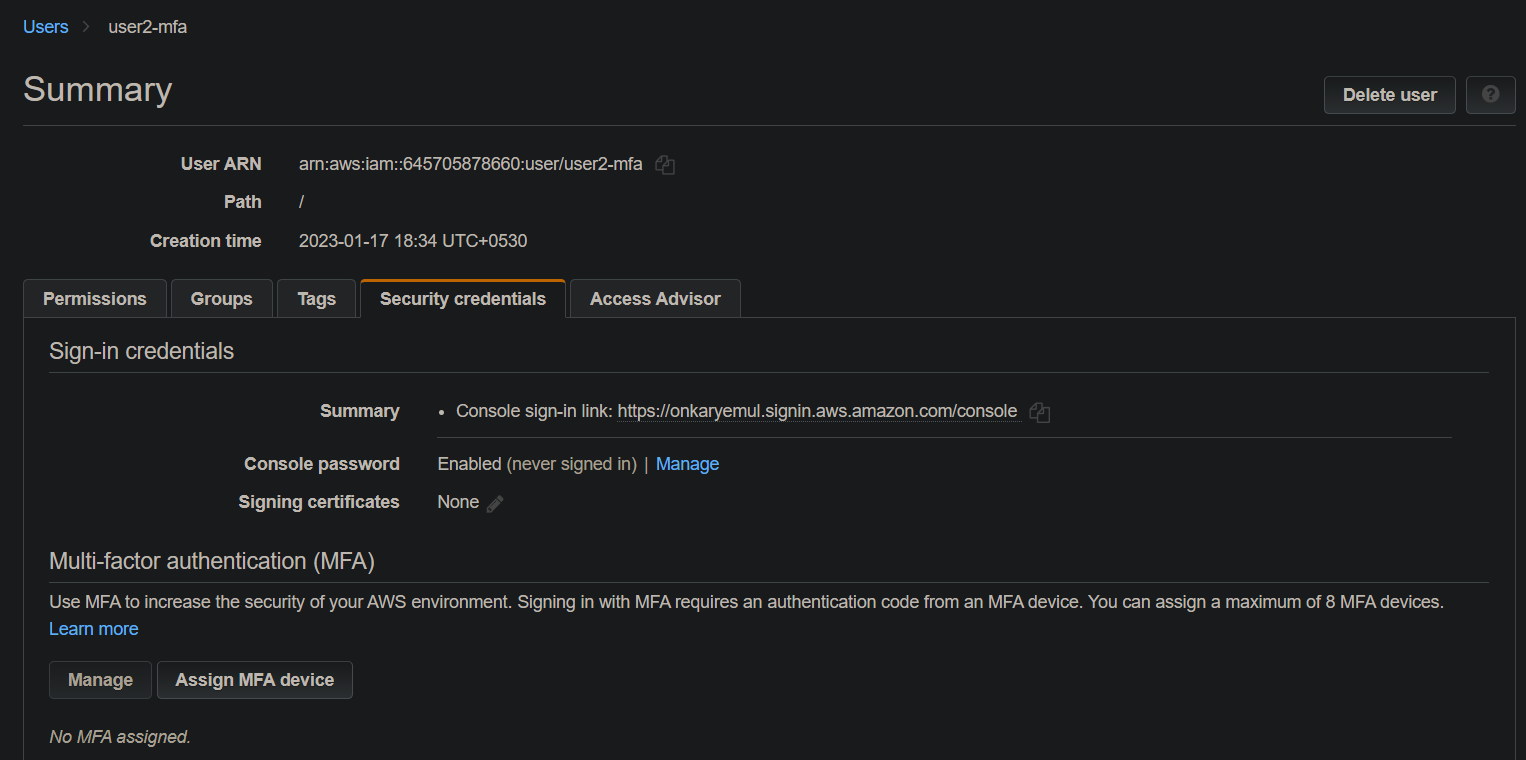


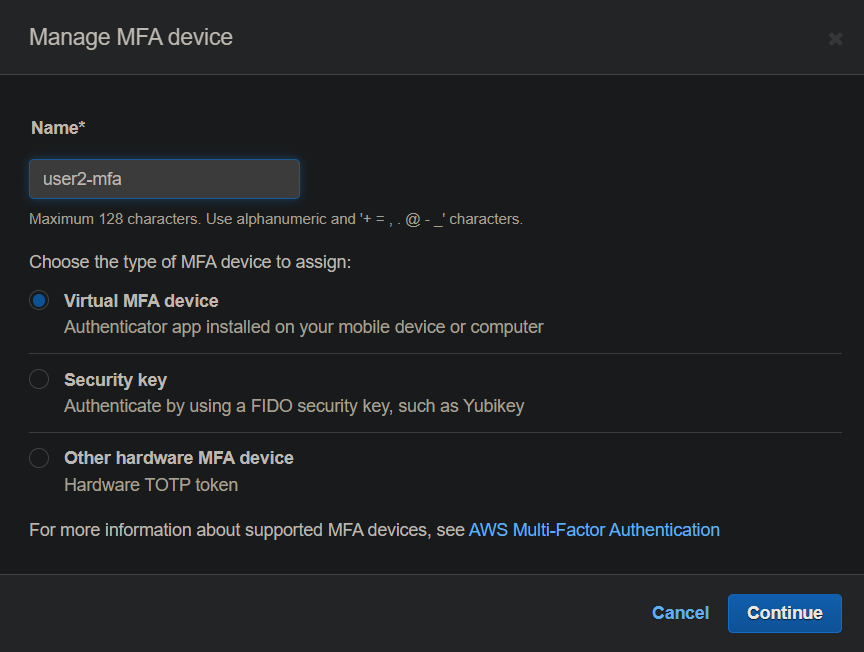




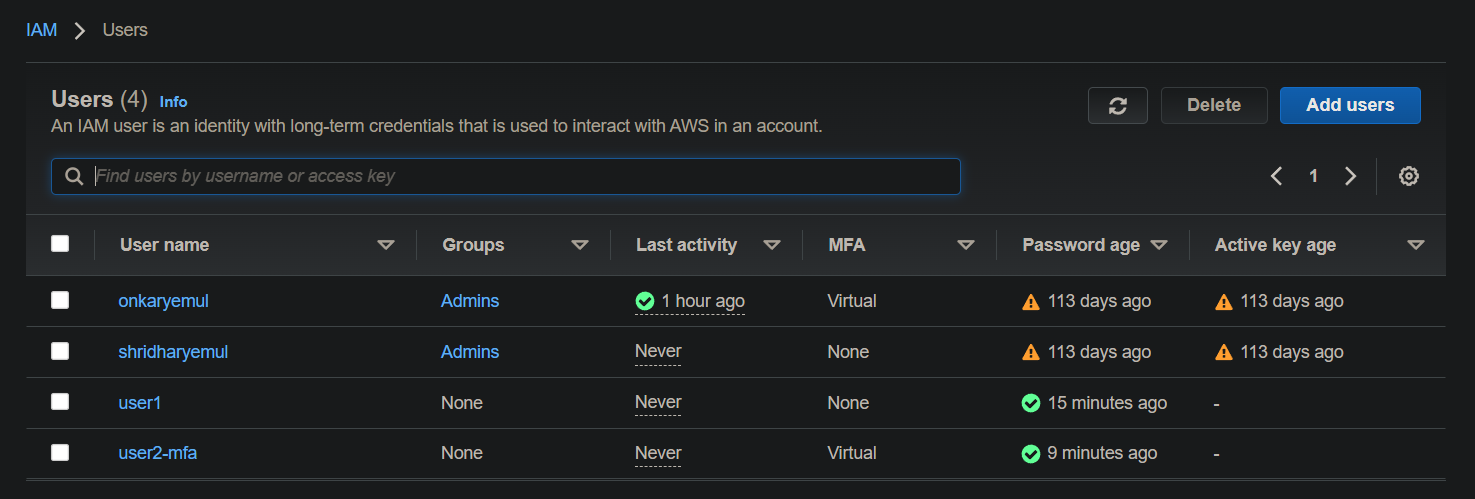


* Now, setting MFA(Multi-factor authentication) for user2-mfa using Virtual MFA device like Google Authenticator

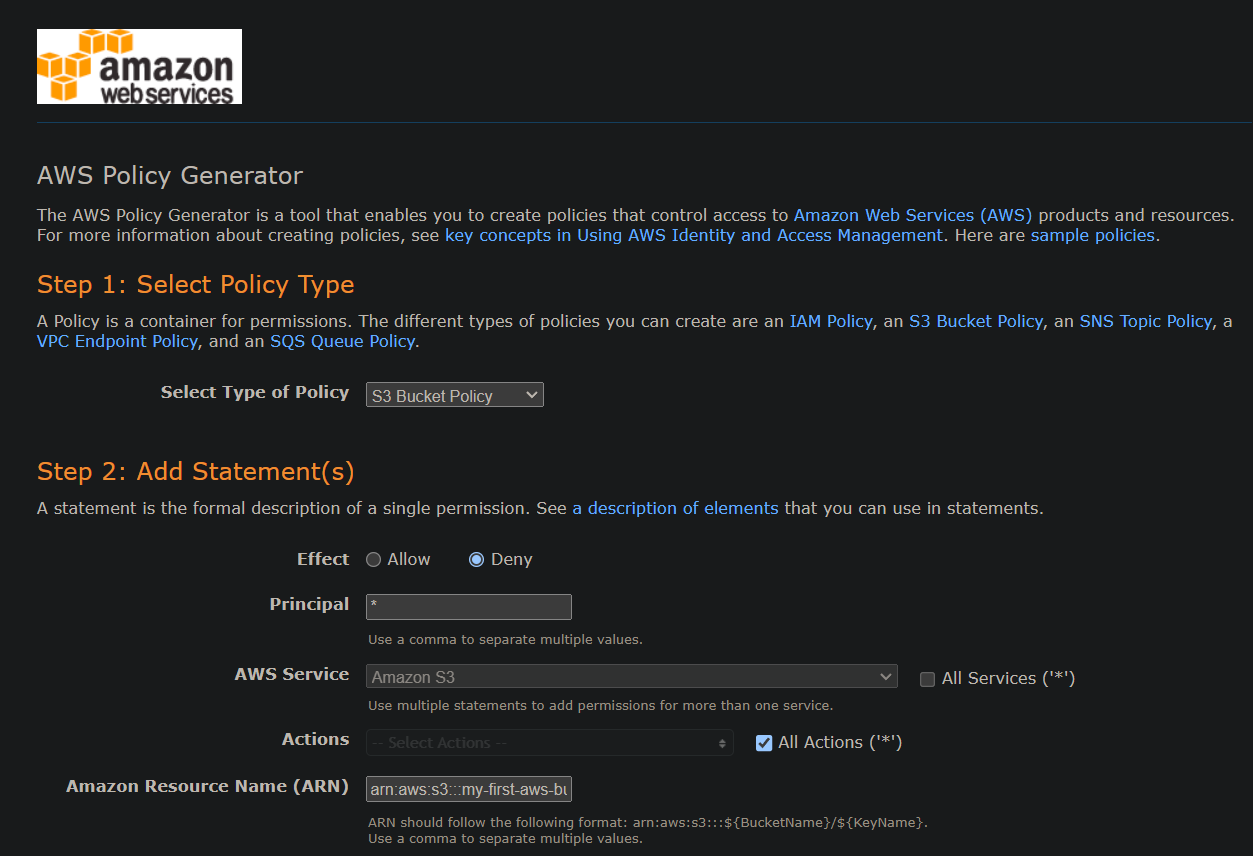


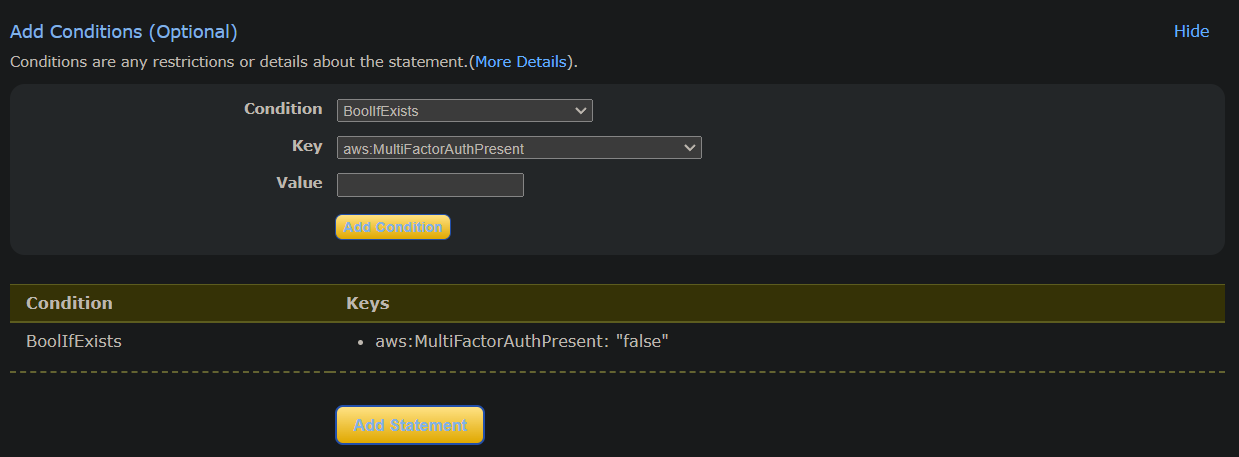


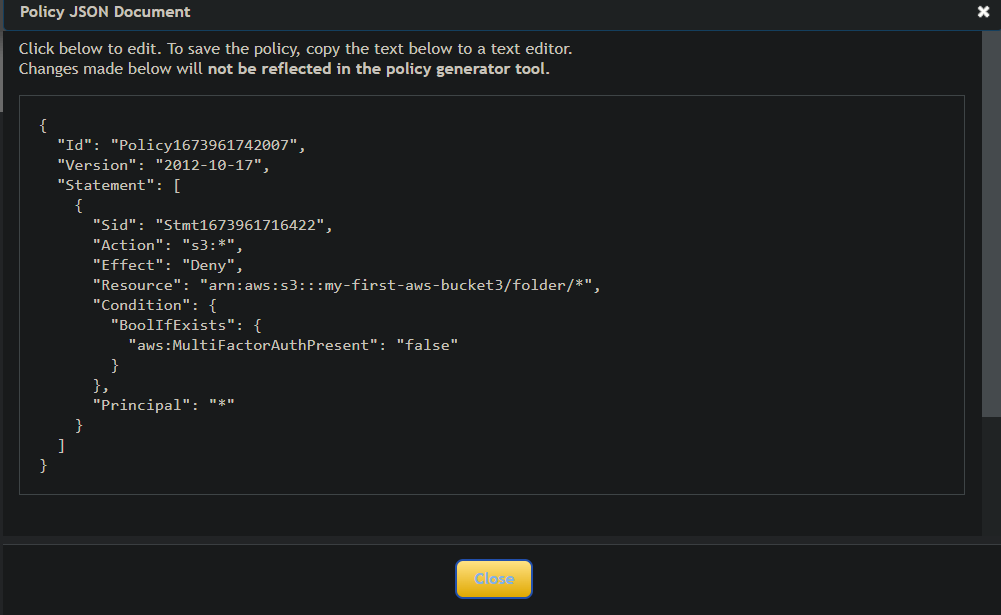


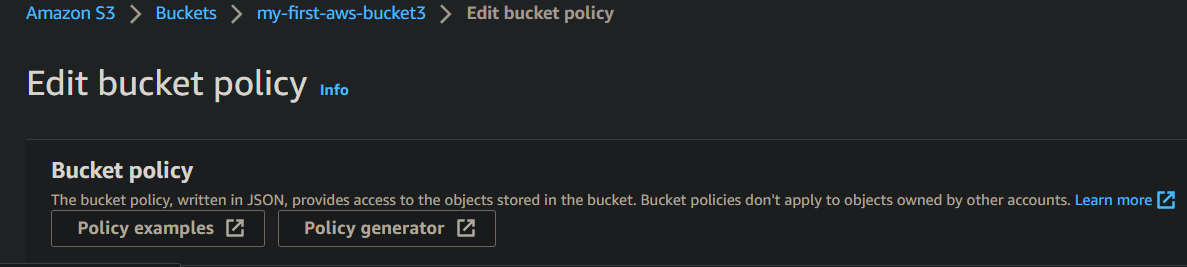


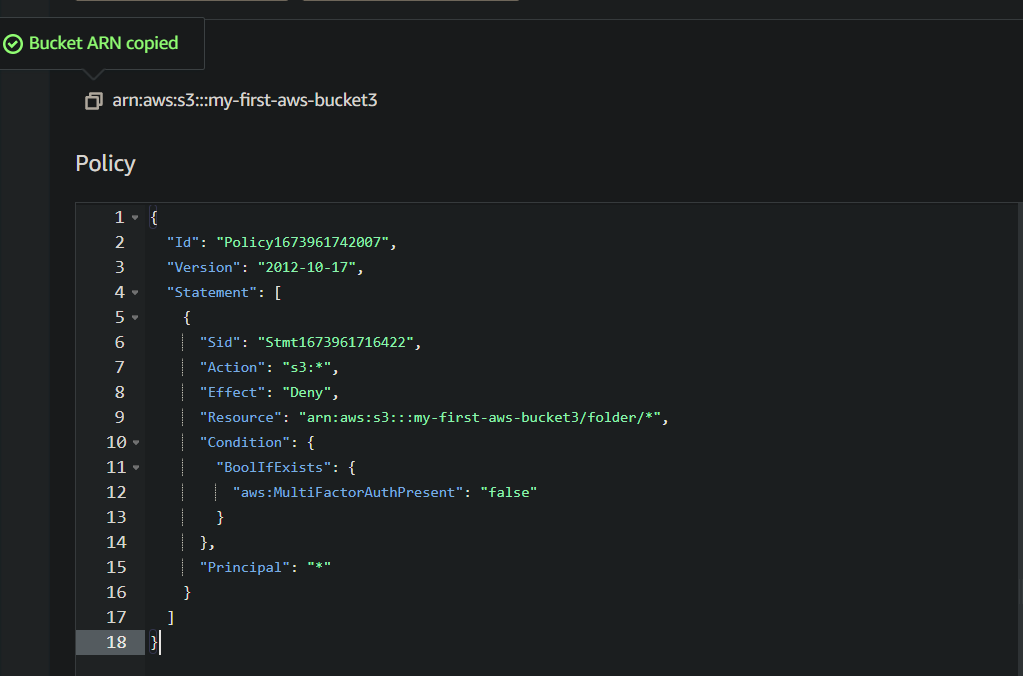
* Now, creating **Bucket policies** for previously created bucket using **Policy generator** .



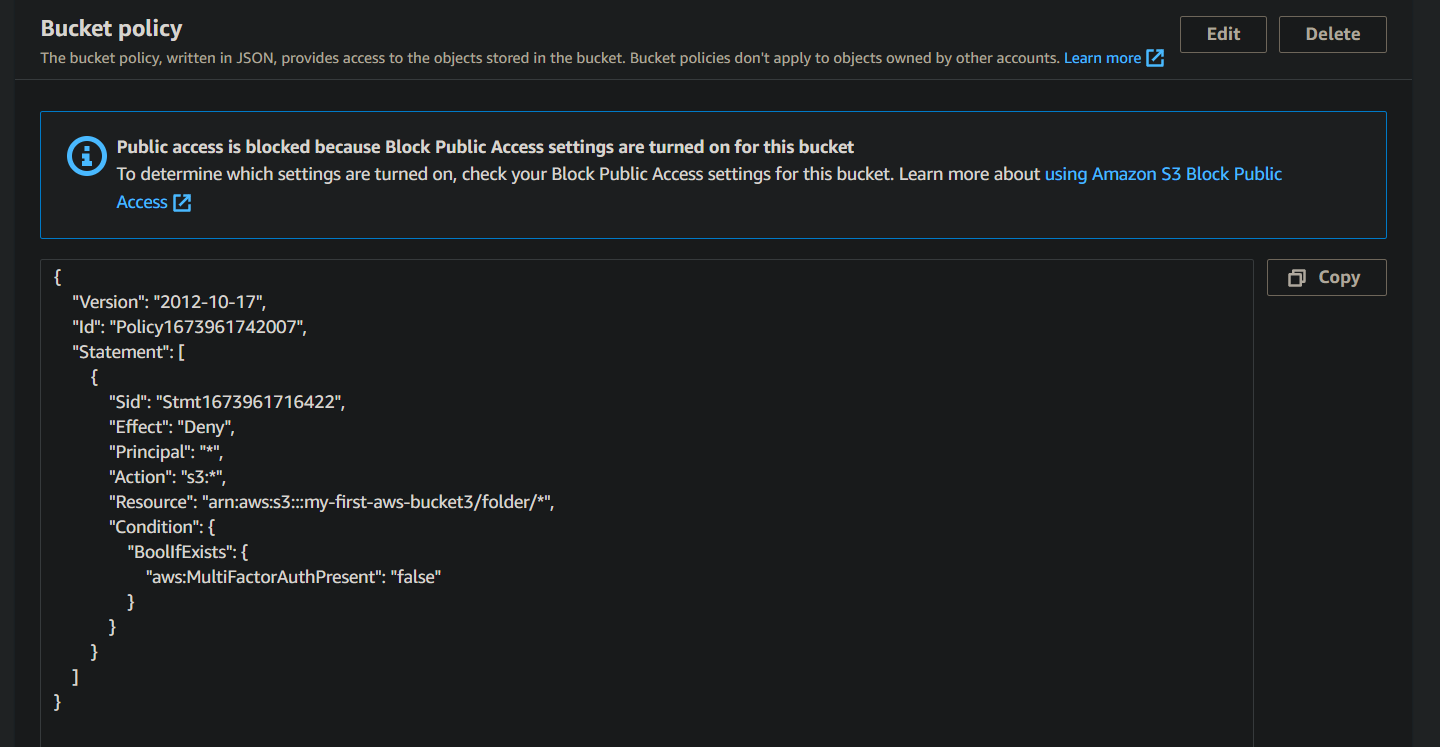








Copy and paste the **JSON file** into the console tab and click on save changes

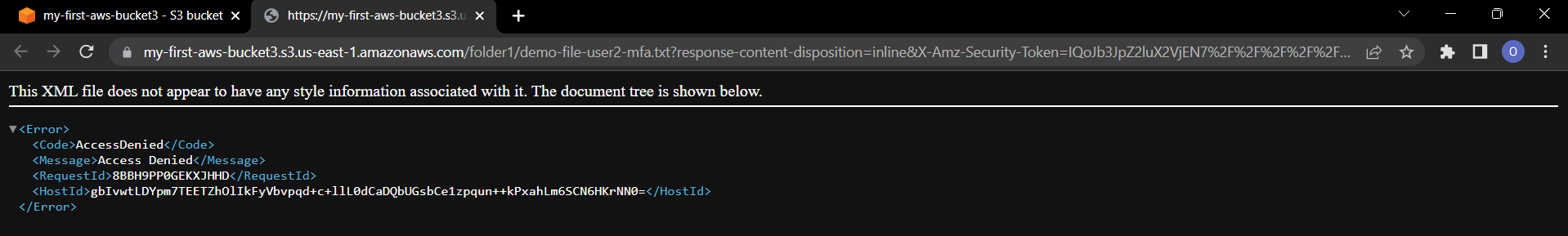


Until now created an AWS S3 bucket with a folder in it containing any simple text file and two users named user1 and user2-mfa where user2-mfa is having MFA authentication in its security configuration whereas user1 doesn’t and we also just

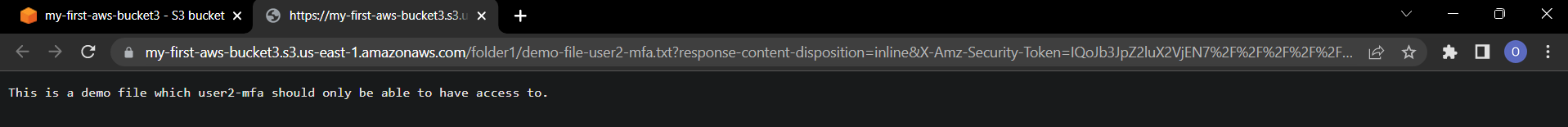
created a **policy that says that only users having MFA authentication factor can access any object in folder1 of our S3 bucket.**

Now it’s time to check our policy.

* Logging in as an IAM user user1 and now let’s try to open the file inside the folder1. Since user1 isn’t having the MFA , so it is not able to view the contents of the file and show following as output -

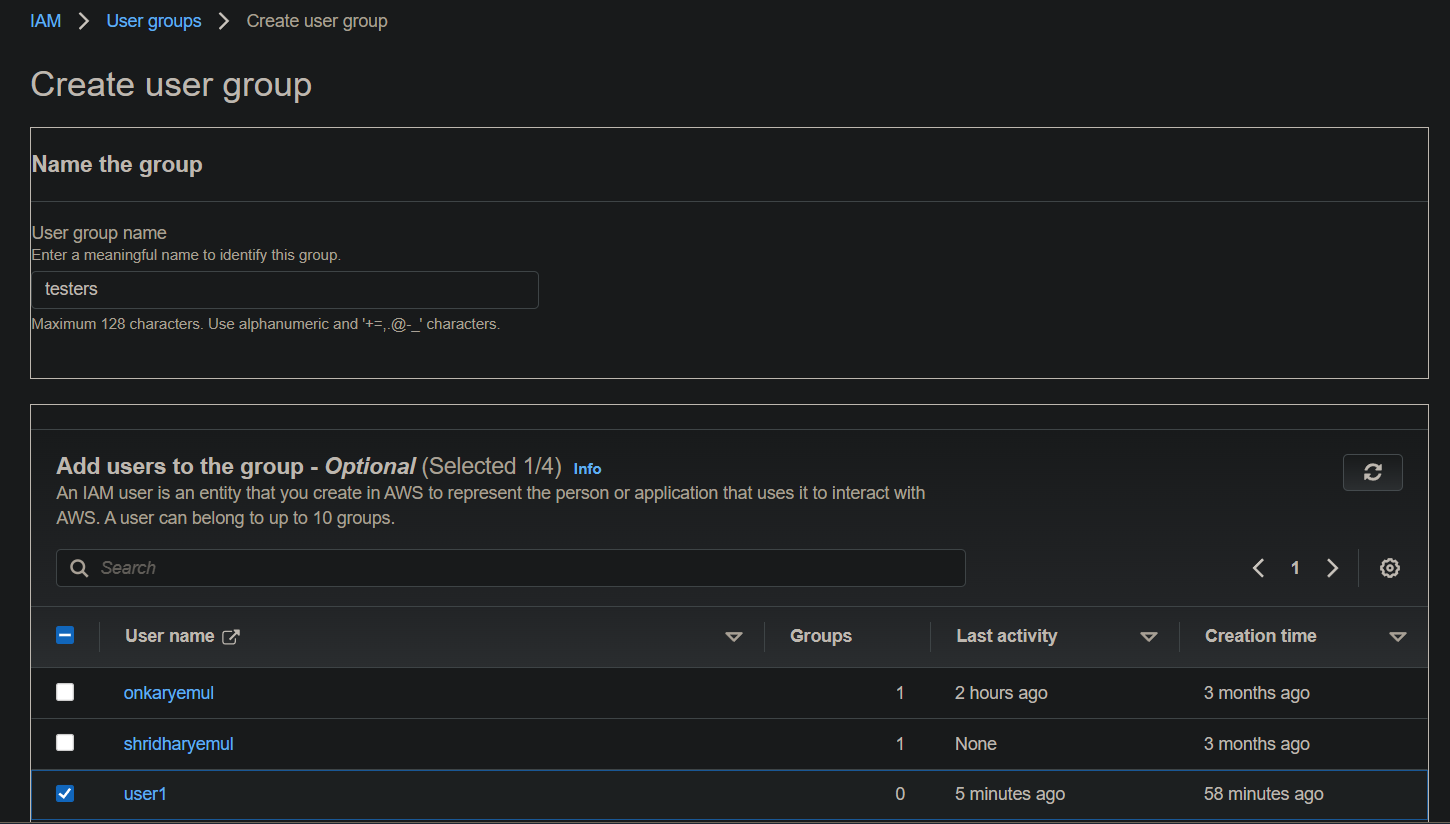


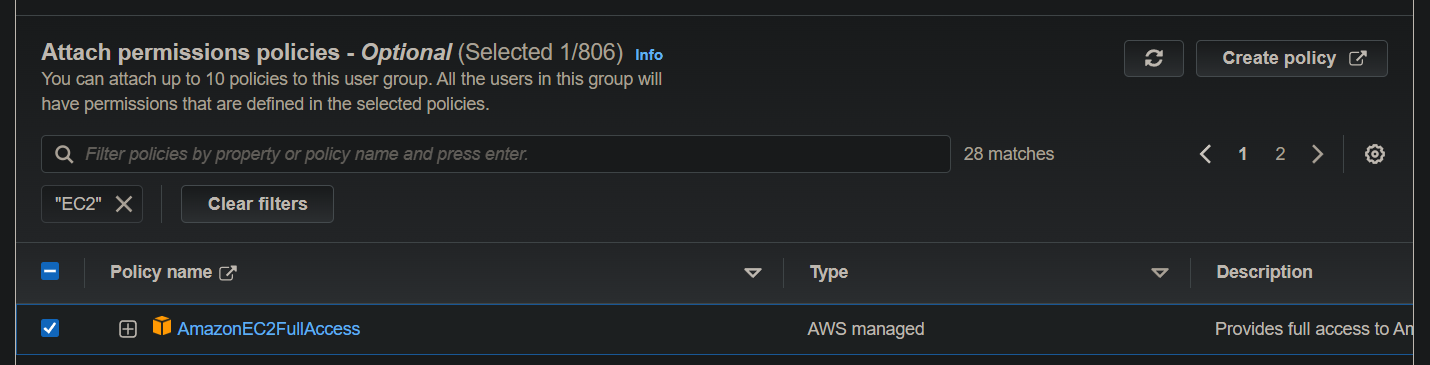
* Now, trying do the same with user2-mfa and it is working perfectly fine for the user2-mfa.

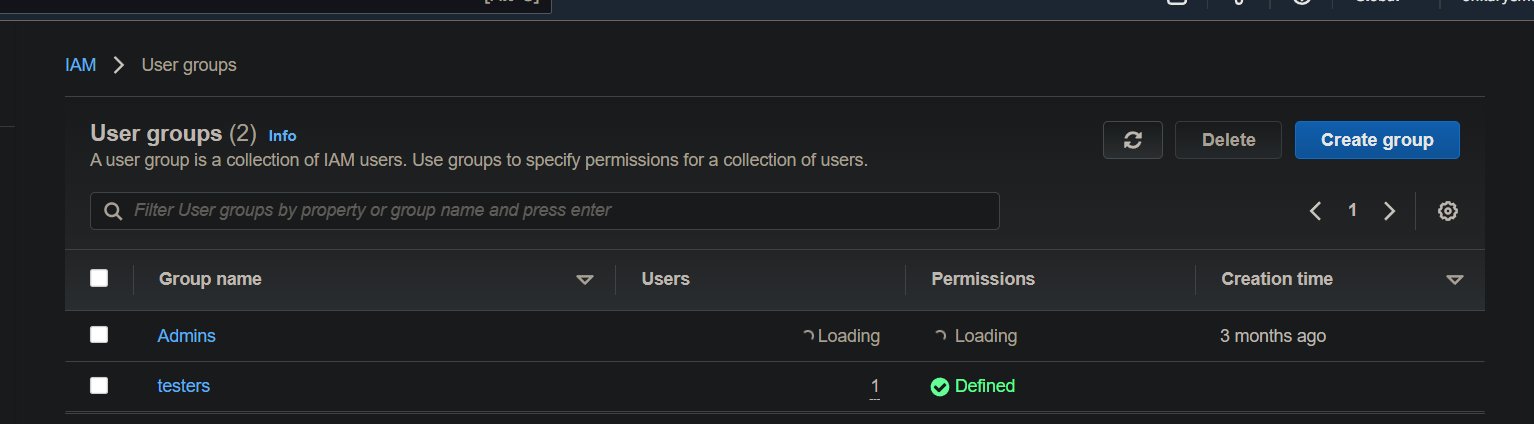


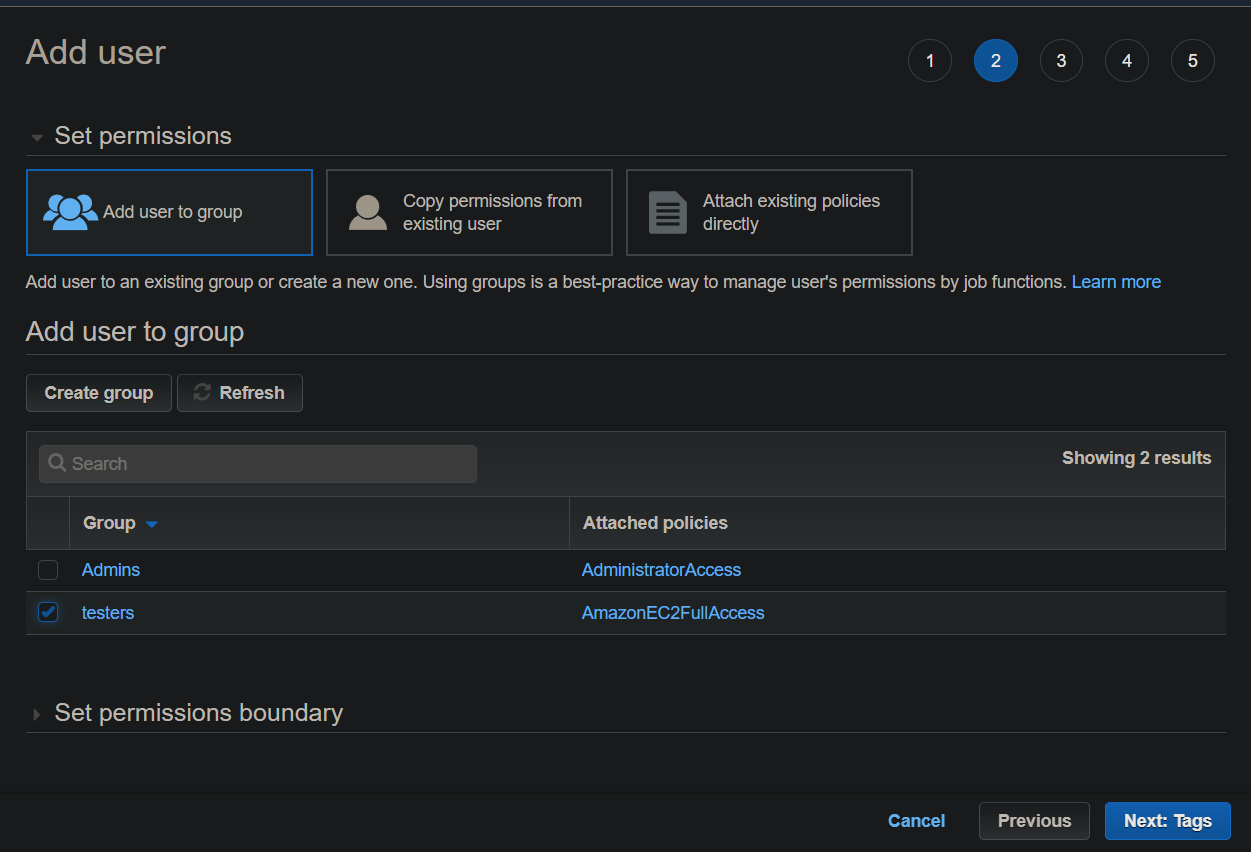
* Now, Creating **IAM User Groups :**

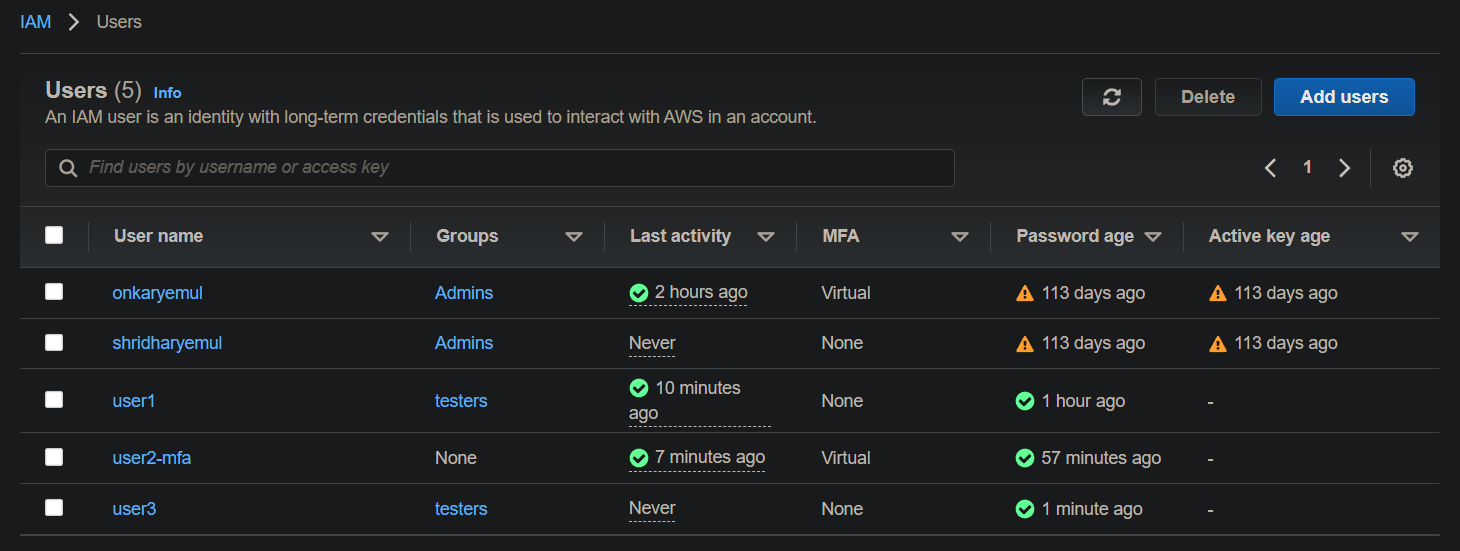
Creating one new group testers and then adding two users user1 and user3 to the group and by attaching existing policies directly and by adding more permission to the group.

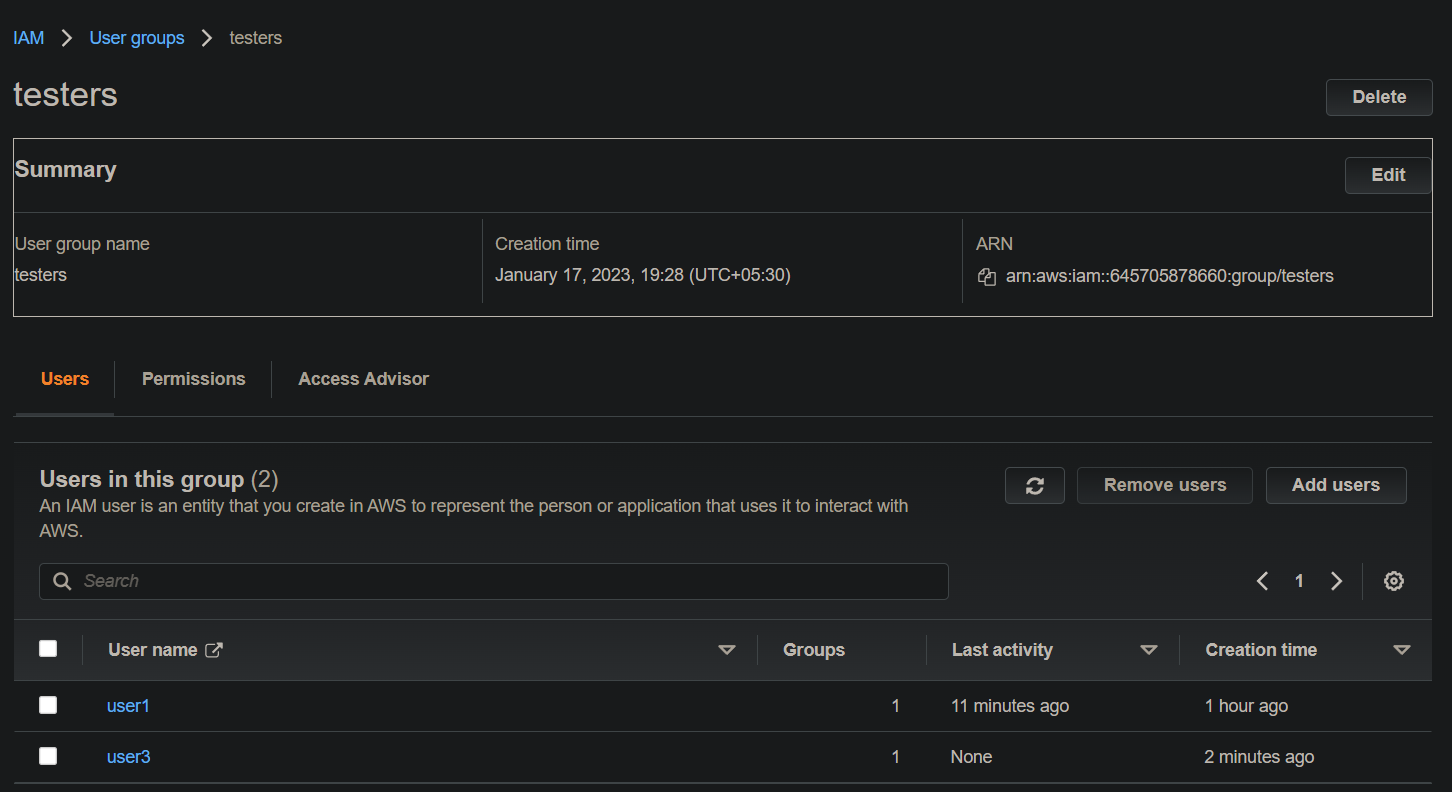


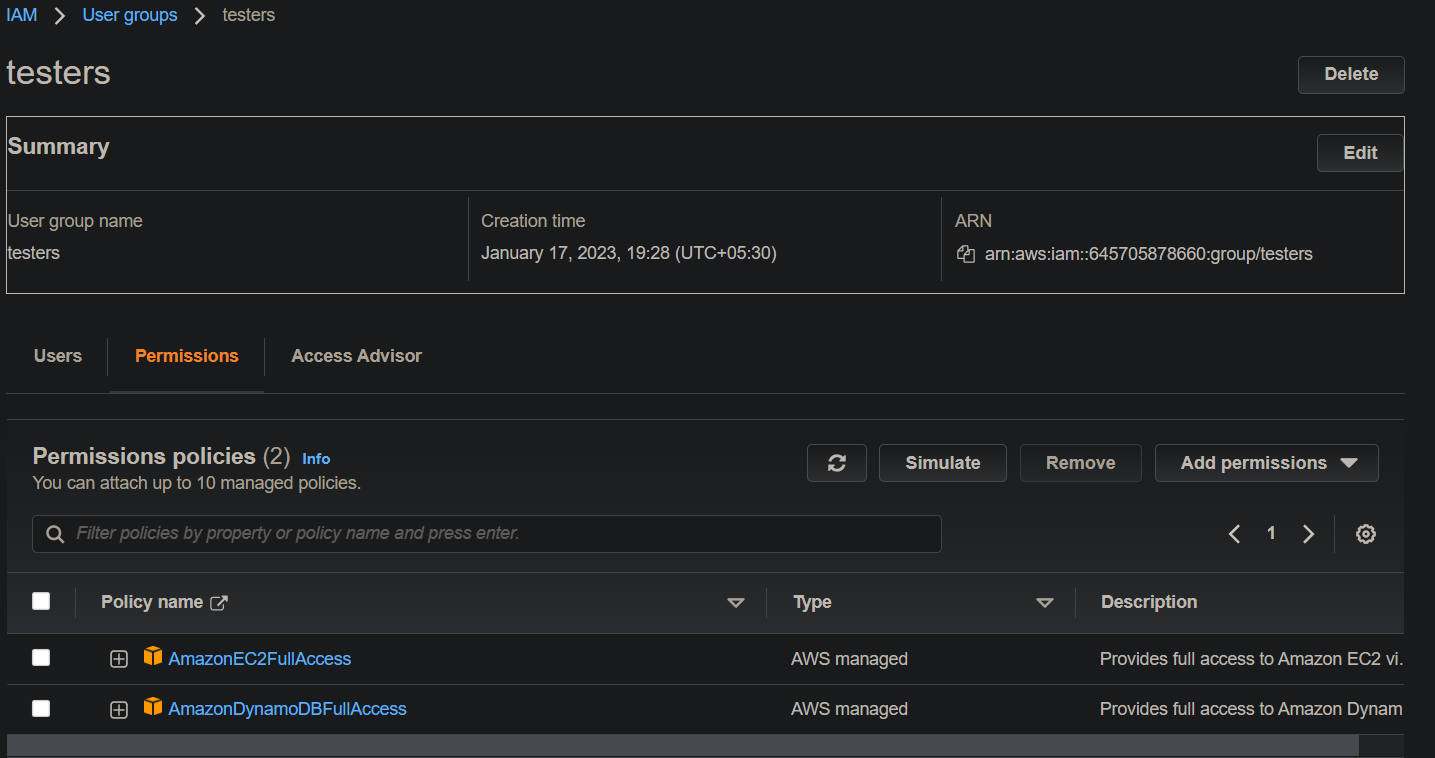


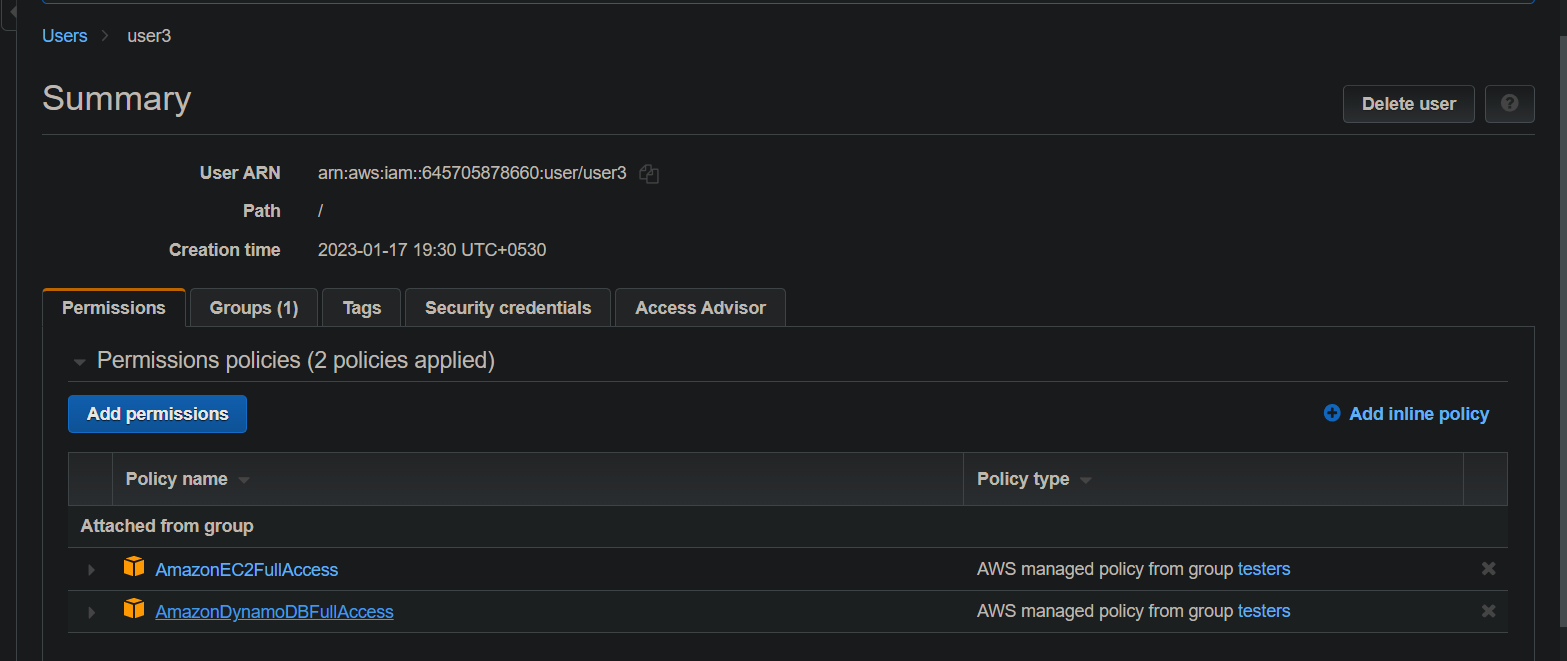


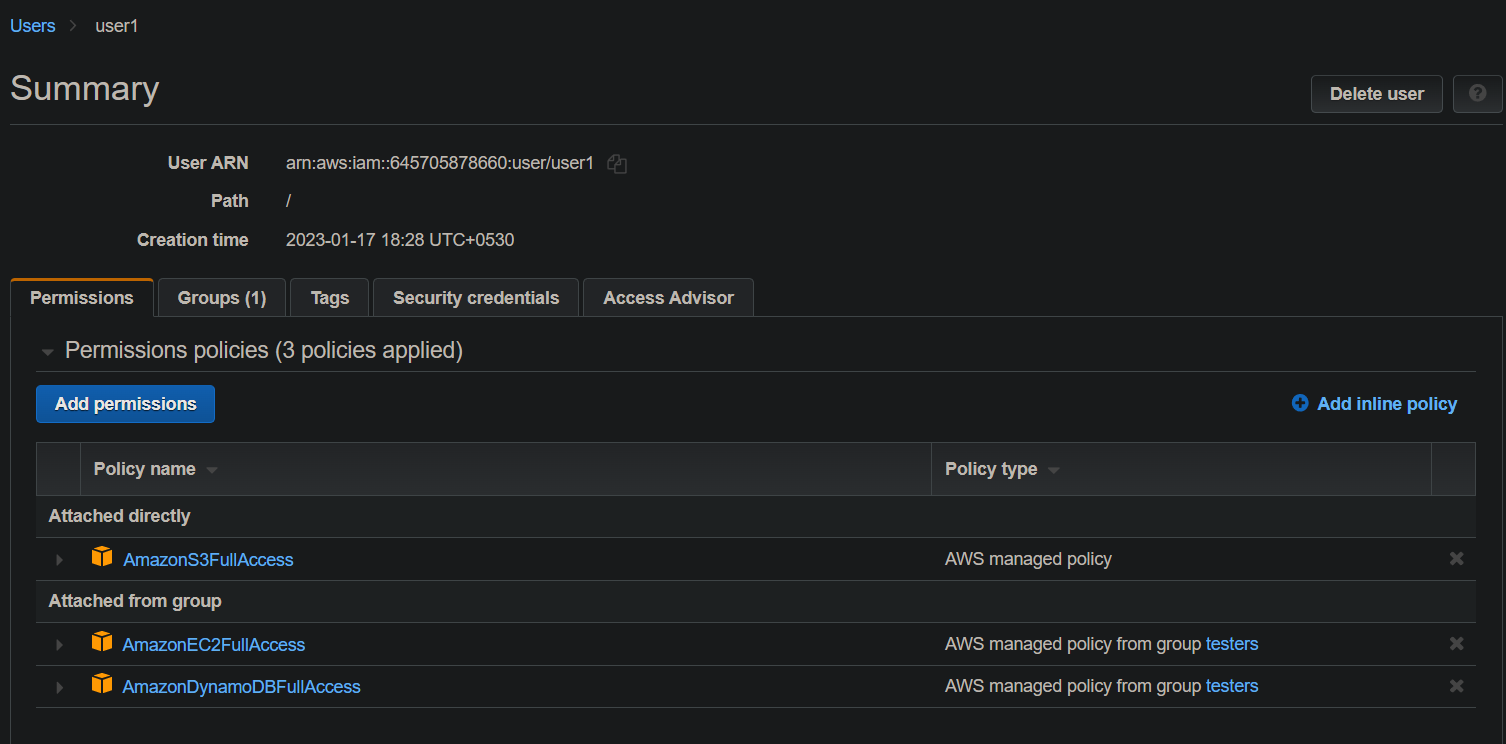












So this was all about AWS IAM users,groups and policies.