**Authentication in IAM**

* Authentication is about making sure you are who you say your are. IAM offers the following authentication features
  + Managing users and their access:
    - You can create & manage users and their security such as access keys, passwords and multifactor authentication
  + Managing federated users and their access:
    - Using IAM you can manage federated users. (Federation is about checking with trusted third party to confirm your identity)
    - Using federation you can use single sign-on to access the aws resources using the credentials of your corporate directory
    - AWS supports Security Assertion Markup Language (SAML) as well as non-SAML options such as AWS Directory services for Microsoft Active Directory to exchange identity and security information between an Identity Provider (IdP) and an Application.
    - Using IdP, you don’t have to manage your own identities or create custom sign-in code, your external users sign in through a well know identity provide such as Login with Google, facebook , Amazon etc.

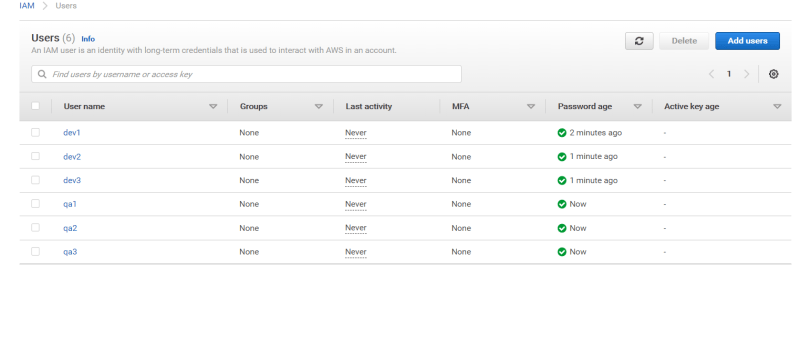
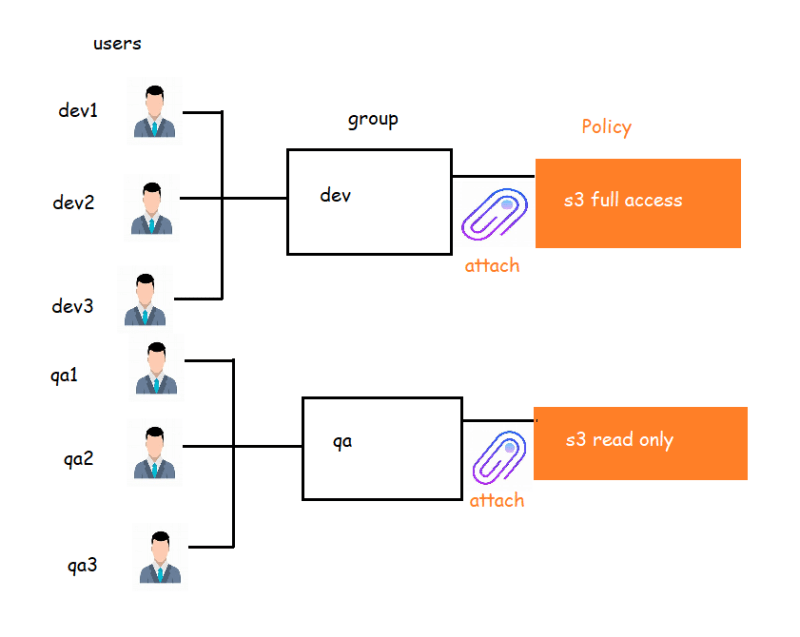
**Authorization**

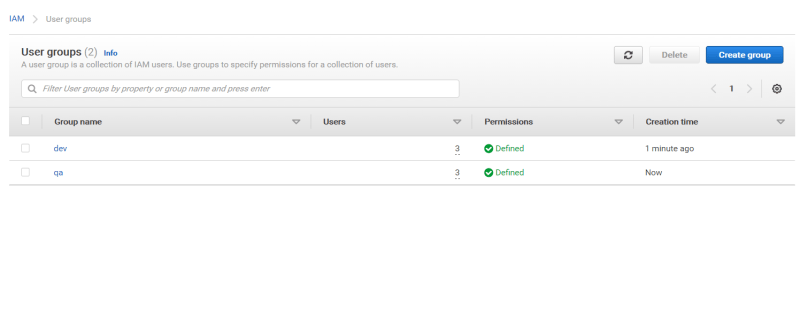
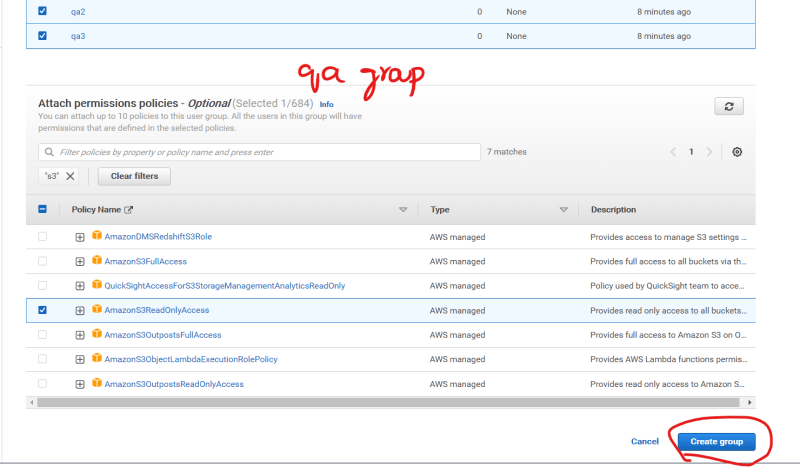
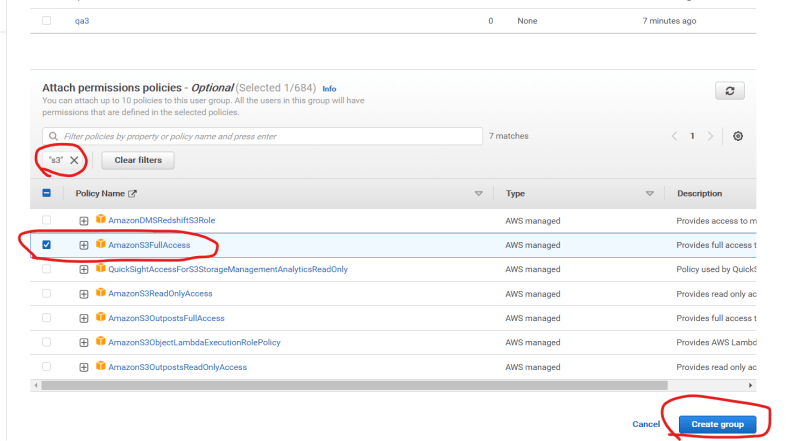
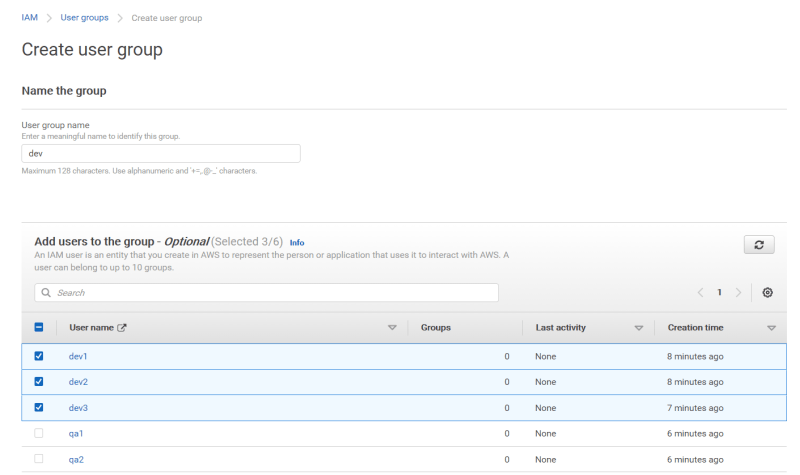
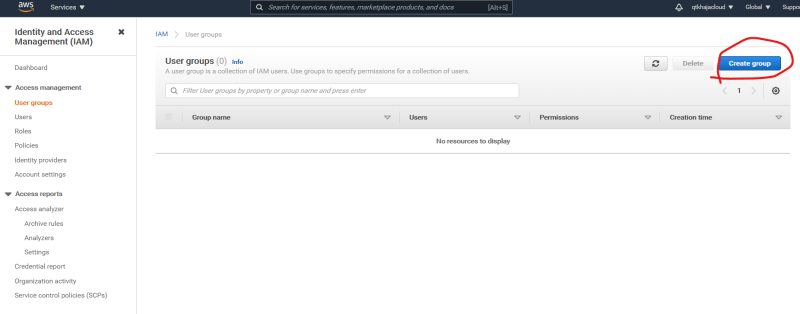
* In AWS authorization is mainly done using IAM Policies
* IAM Policy is written in JSON where you define permissions
* AWS has built in IAM Policies which we can use for standard use cases, for any custom scenario, we can create our own IAM policies
* IAM Policy can be attached to any IAM Entity (user, group or role)

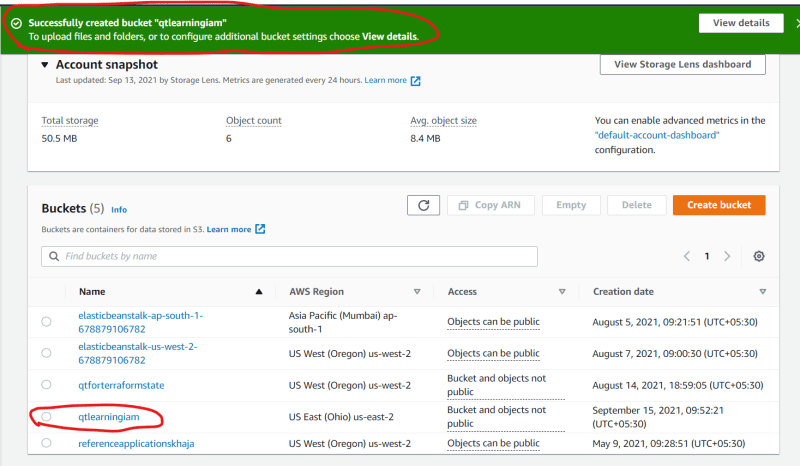
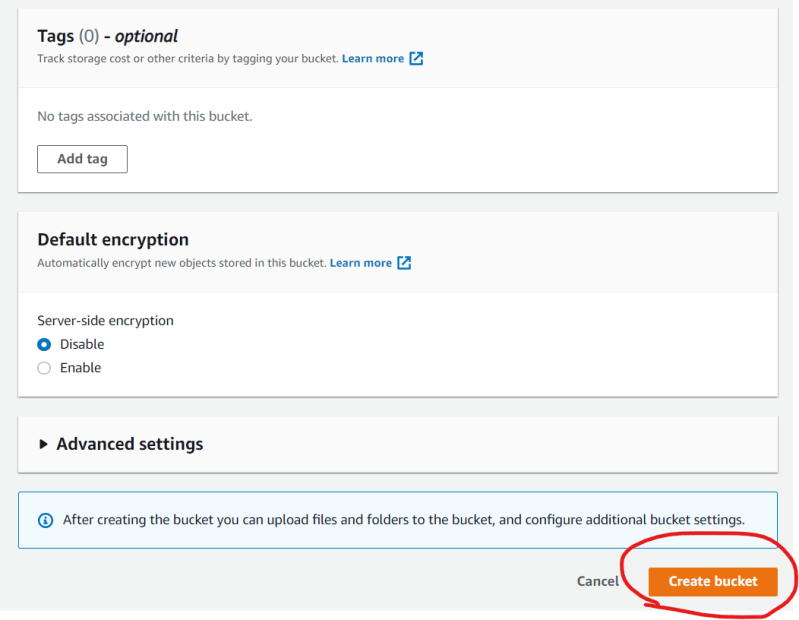
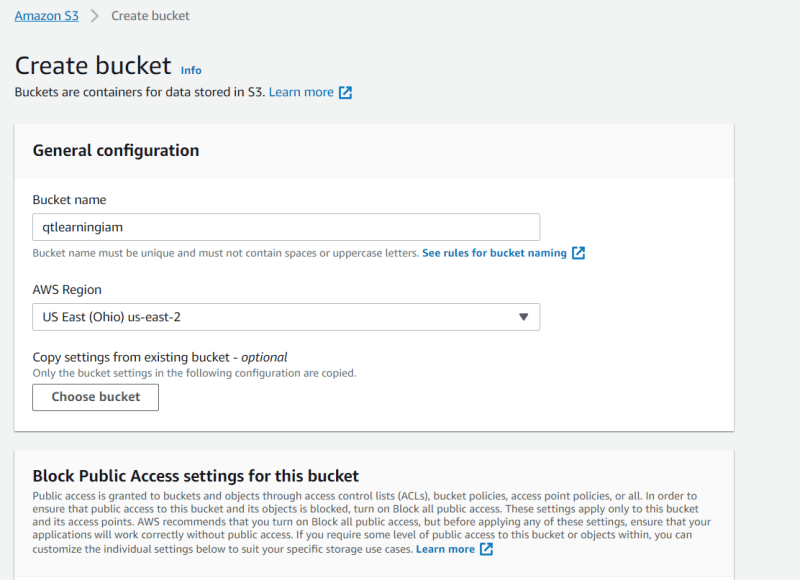
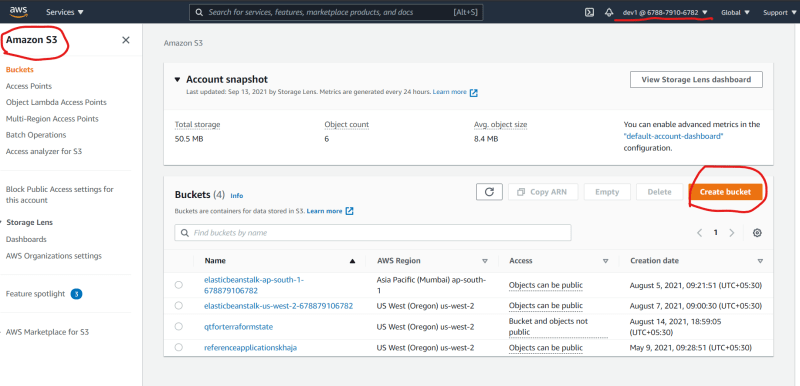
**IAM Entities**

* In AWS we have 3 IAM Entities
  + User: This is an user trying to access AWS from Console/Programatically. When we create an user with console access we would have username and password, whereas when we create an user with programatic access we would have username, access key id and secret access key
  + Group: Group is collection of users so that you can manage permissions at group level rather than at individual user level.
  + Role: A role represents an AWS Resource (for eg ec2) trying to access other AWS Resource (S3 bucket)

**Scenario-1:**

* We have six users. 3 of them are developers and 3 of them are testers
* Create six iam users in your account without any permissions 
* Now AWS has managed policies (Policies created and managed by AWS) which we can use for assigning permissions
* Now lets create the groups and attach policies as mentioned below 



* Now lets try to login as developer (any dev user) and verify the s3 access 
* Now lets login in as qa (any qa user) and try to view and delete the bucket created by developer 