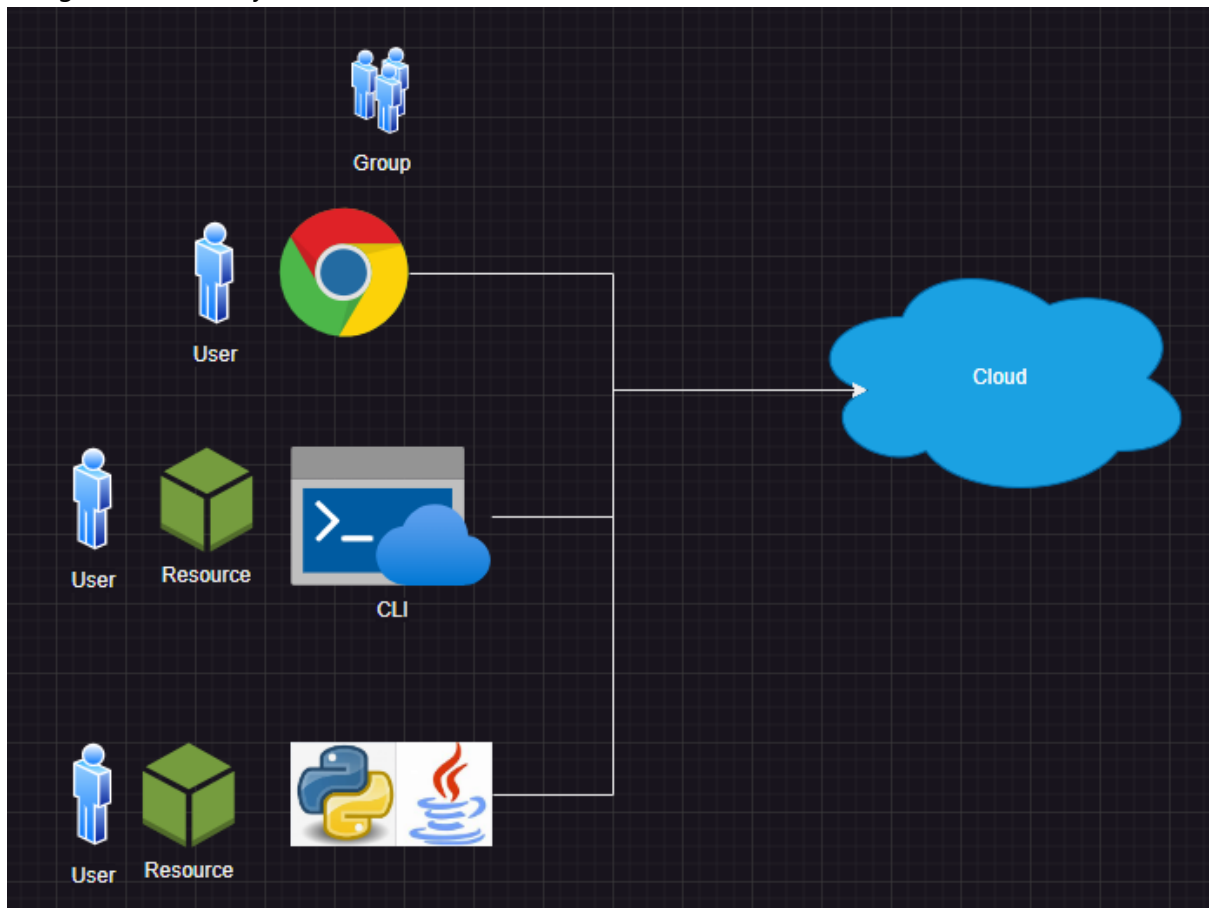


Users of Cloud

- Generally we have two categories of usage
 - user (employee)
 - program/tool/a cloud resource
- Cloud is accessible in 3 ways
 - using browsers (Console/Portal)
 - using command line (AWS CLI/Azure CLI/Azure Powershell)
 - using code (Azure Python SDK, AWS Boto3)

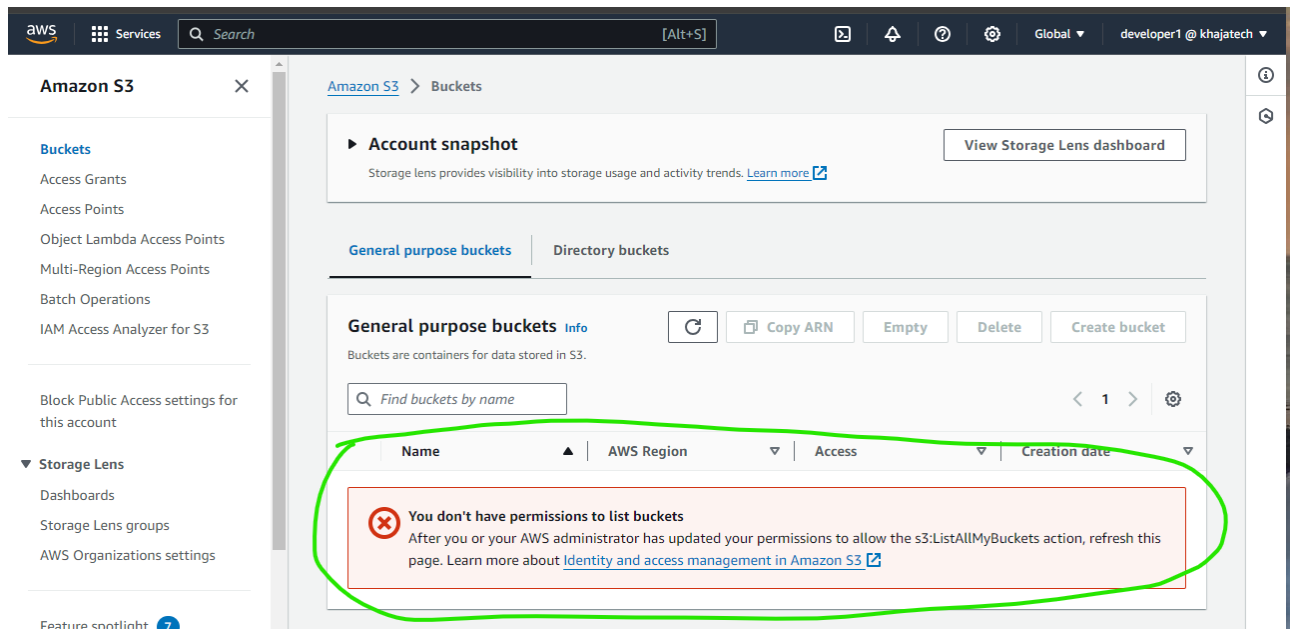


Authentication and Authorization

- We need to give access to users who can perform login operation manually and we also need give acces to programs where manual login is not an option.
- Generally for programs we have two options
 - Temporary access
 - Permanent Access
- For every user/program we need to give well defined set of permissions.

AWS - user creation

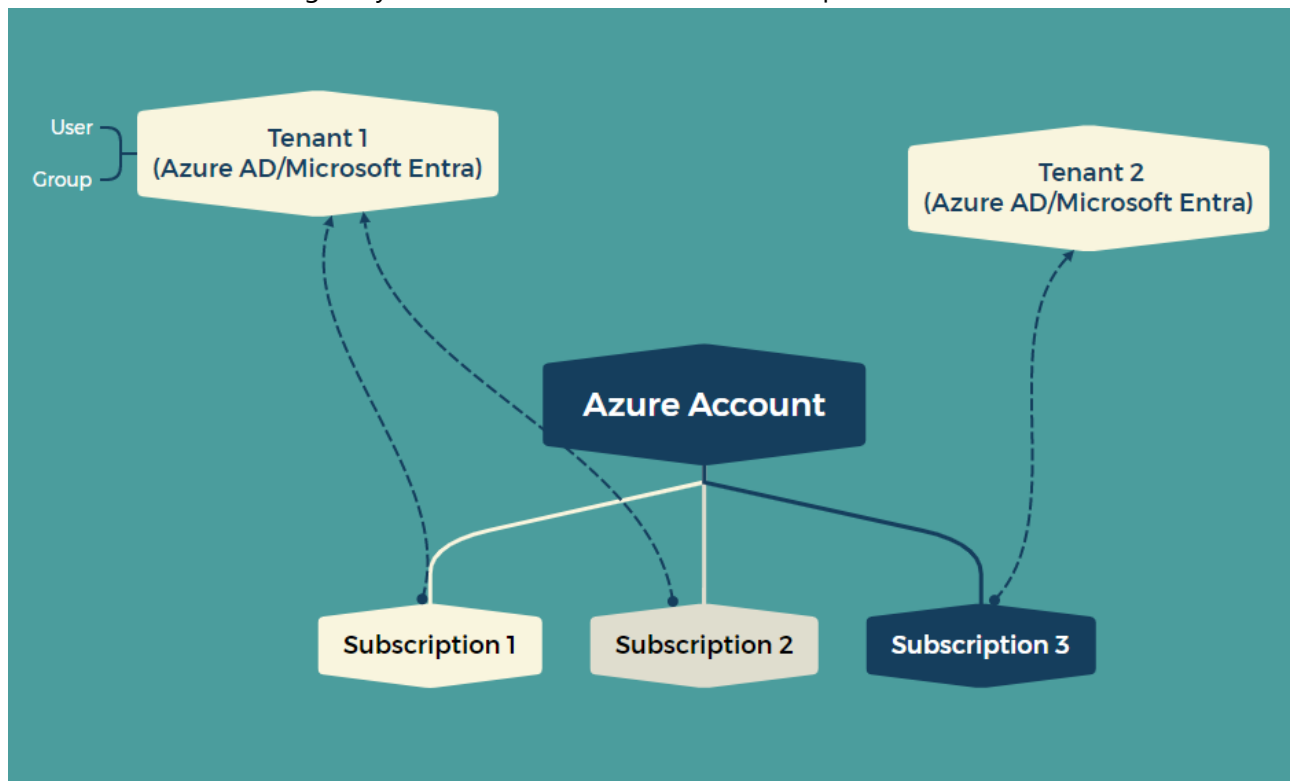
- The user with which you have create AWS Account is considered as root account and this account has all permissions.
- When an account is created in AWS, we get unique account-id for which alias can be set
- Create a user without selecting any group and then login int aws account with IAM user credentials, Navigate to any service



- Users can set Multifactor Authentications with Google Authenticator or Microsoft Authenticator for free
- In AWS Permission is defined by Policies. AWS provides lots of managed policies
- components:
 - root account
 - user
 - policy
- Generally in organizations, policies provided by aws are not enough and it is expected to create custom policies

Azure - user creation

- In Azure users are managed by tenants which are linked to subscriptions.



- Steps:
 - Create a user in tenant (Microsoft Entra/Azure AD)

- Add a role (Reader) to created user at subscription level
- Azure user Roles that needs to assigned at subscription/resource group/resource level to attach permissions to user which it refers as RBAC (Role Based Access Control)
- In Azure also, we need to create custom roles.

JSON

- JSON (Javascript Object Notation) is used to represent data
- JSON helps in representing data which is easy for machines and humans to read data and it is light weight.
- JSON uses name value pairs to design data.

```
"name": <value>
```

- Value can be of following types
 - text/string
 - number
 - boolean
 - array or list
 - object or map or dictionary
- [Refer Here](#) for video to understand yaml and json