Aws Task Lab 1

1- Create aws account and set billing alarm

2 - A - Create 2 groups one admin and one for development

B - in the admin group it has admin permission , and in the development only access to s3

C- create admin-1 user console access and mfa enabled in admin group

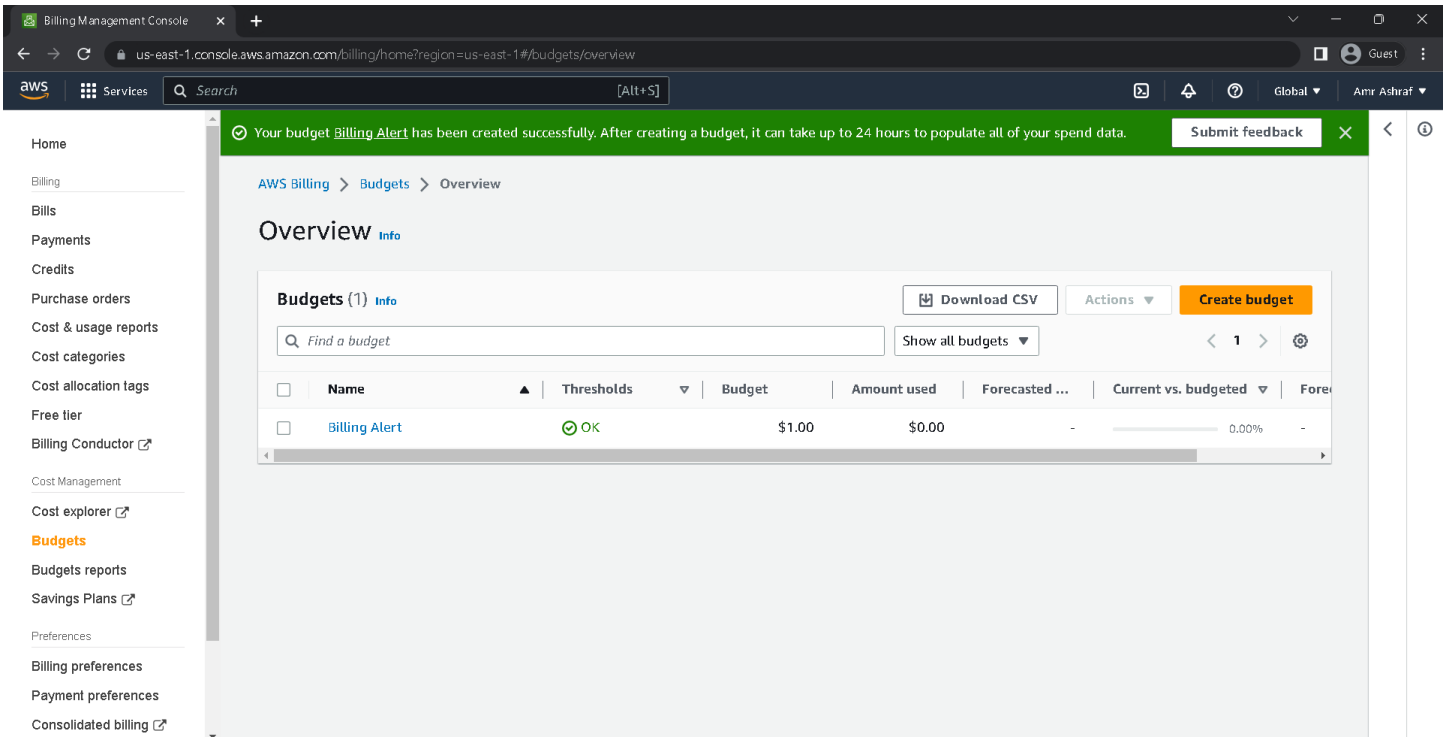
D - and admin2-prog with cli access only then list all users and groups using commands not console

E - in the development group create user with name dev-user with programmatic and console access then try to access aws using it (take a screenshot from accessing ec2 and s3 console)

F - Also access cli using dev-user and try to get all users and groups using it

The Solutions:

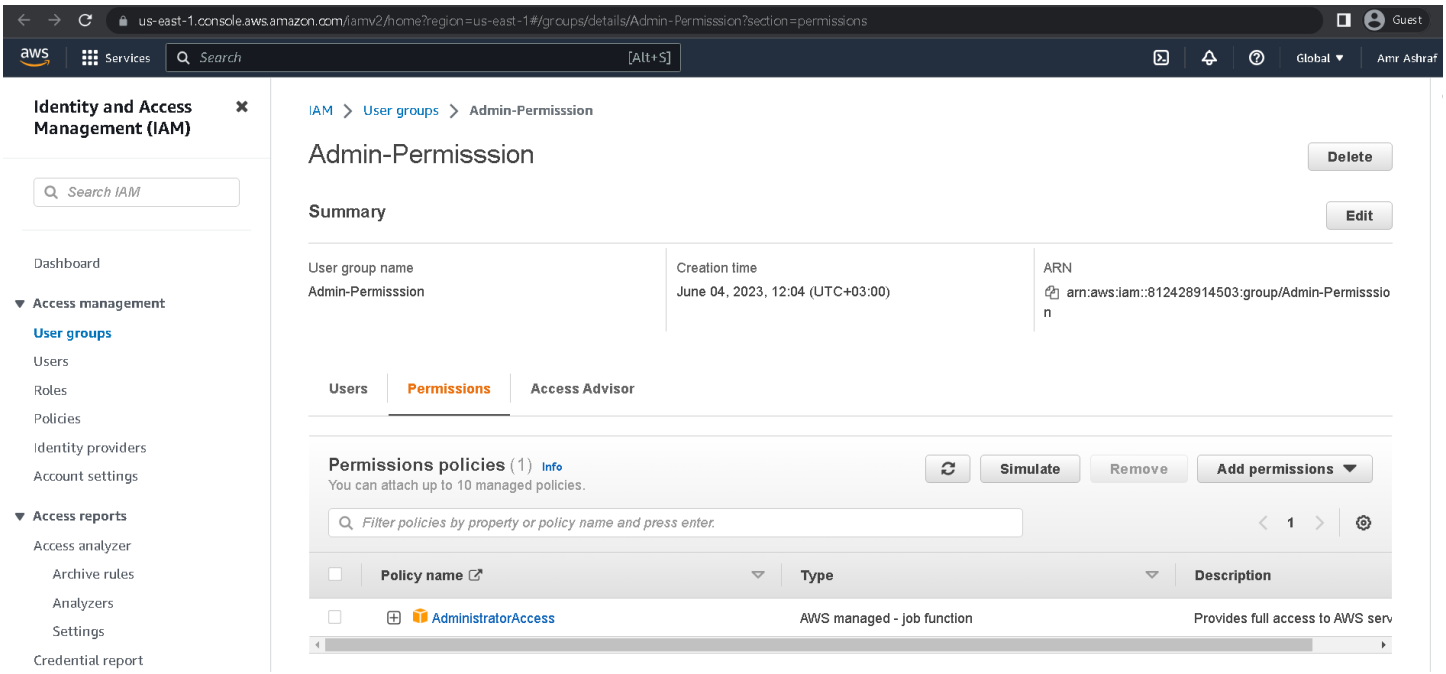
1 - Create an AWS account and set up billing alarm



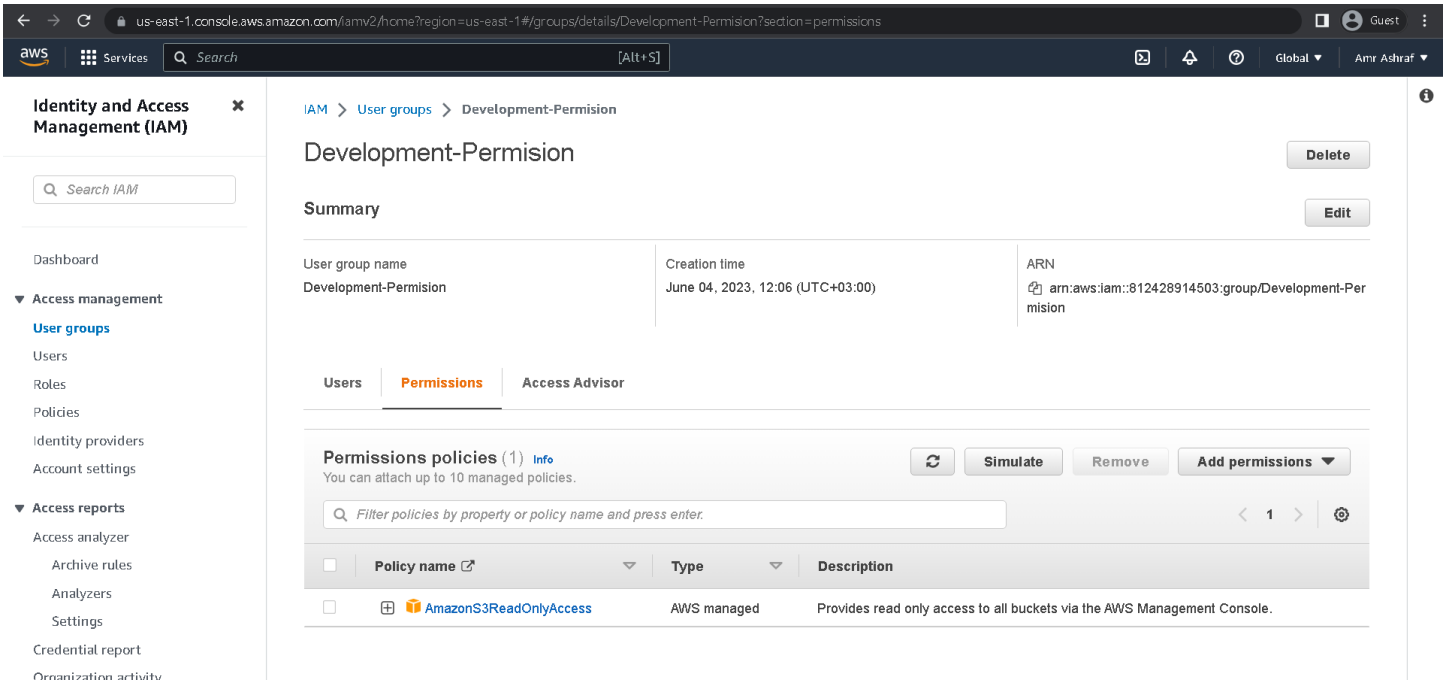
2 –

A - Create two groups, one admin and one for development

For Admin



For Development



To Check Them From AWS CLI

```
[cloudshell-user@ip-10-8-96-127 ~]$ aws iam list-users
{
    "Users": []
}
[cloudshell-user@ip-10-8-96-127 ~]$ aws iam list-groups
[cloudshell-user@ip-10-8-96-127 ~]$ aws iam list-users
{
    "Users": []
}
[cloudshell-user@ip-10-8-96-127 ~]$ aws iam list-groups
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "Admin-Permisssion",
            "GroupId": "AGPA32KEJN5D3FL26HDUR",
            "Arn": "arn:aws:iam::812428914503:group/Admin-Permisssion",
            "CreateDate": "2023-06-04T09:04:39+00:00"
        },
        {
            "Path": "/",
            "GroupName": "Development-Permision",
            "GroupId": "AGPA32KEJN5D6QM6ODCRR",
            "Arn": "arn:aws:iam::812428914503:group/Development-Permision",
            "CreateDate": "2023-06-04T09:06:33+00:00"
        }
    ]
}
```

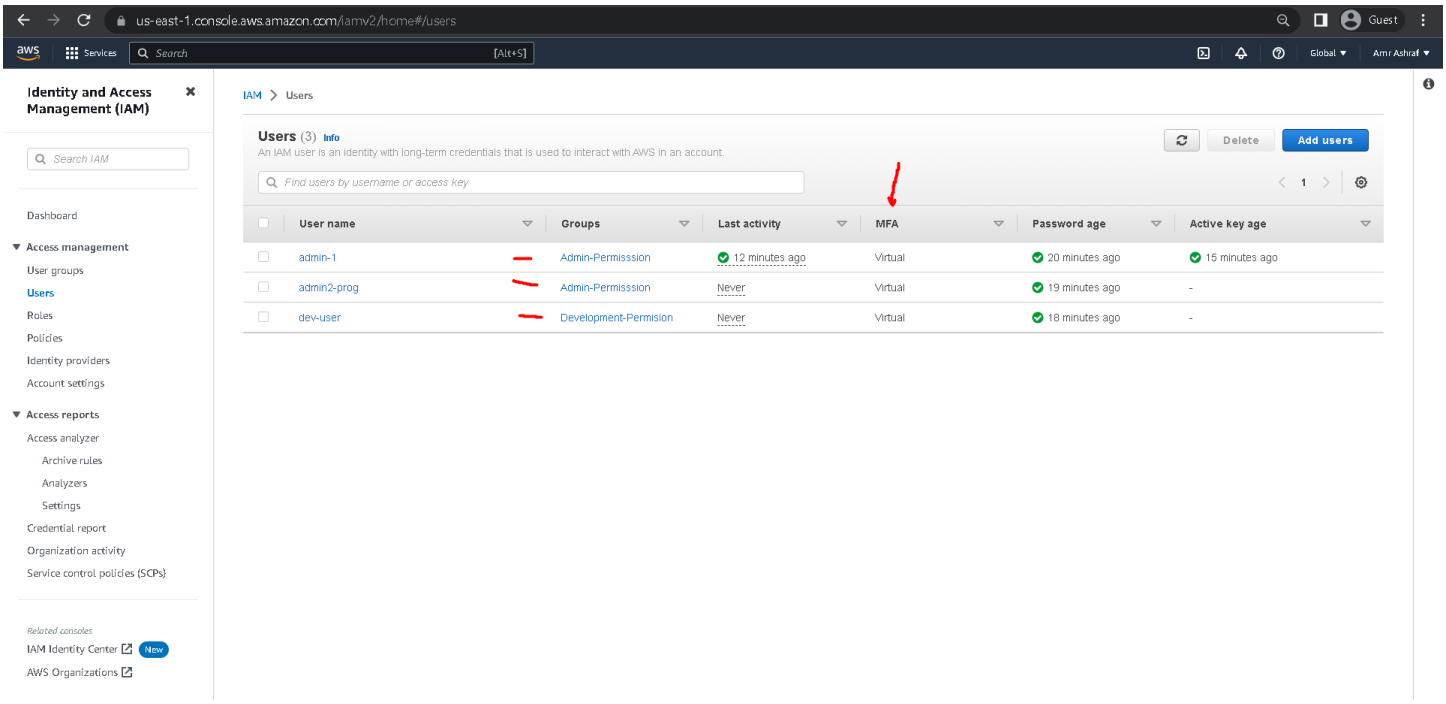
B –

Create 3 users

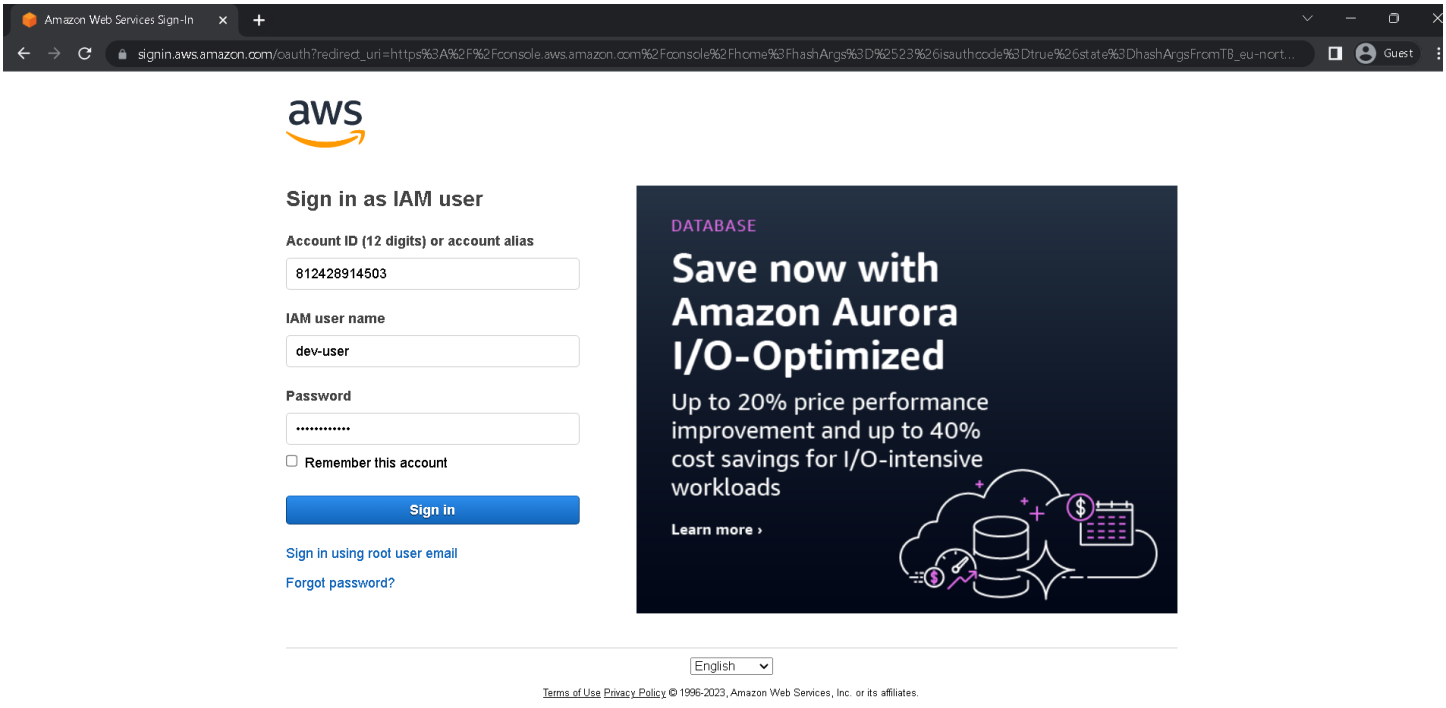
1 – User : admin-1 [Enabled MFA – Added To Group Admin-Permission]

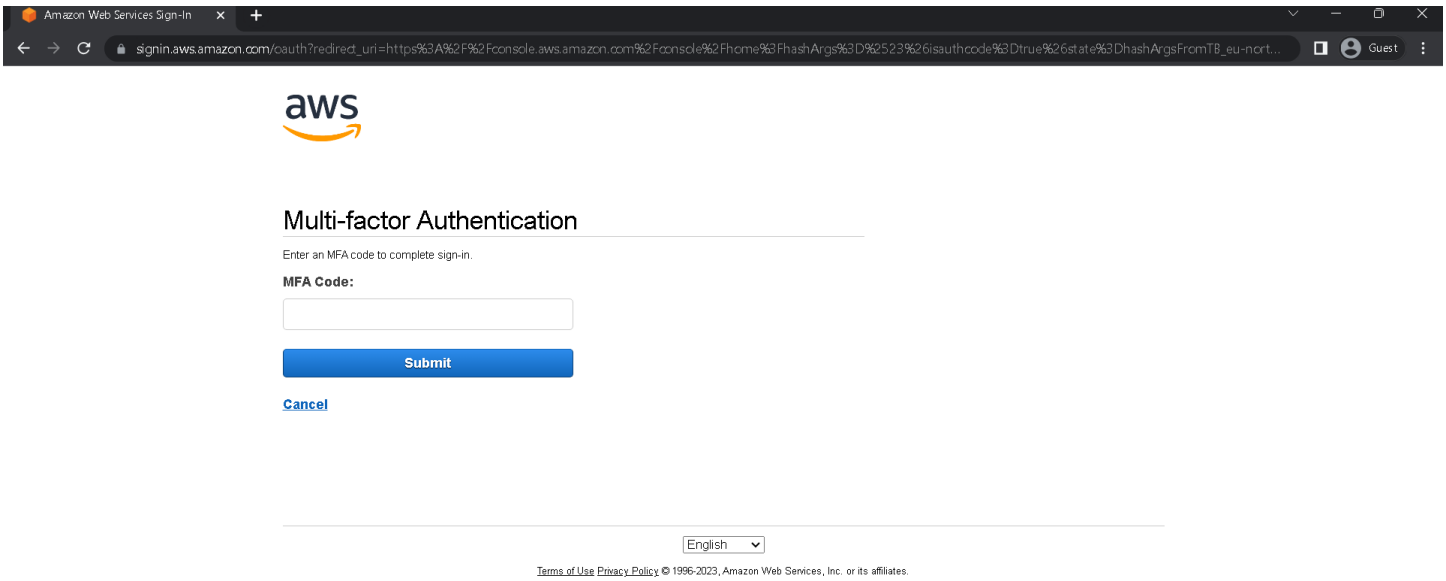
2 – User : admin2-prog [Enabled MFA – Added To Group Admin-Permission]

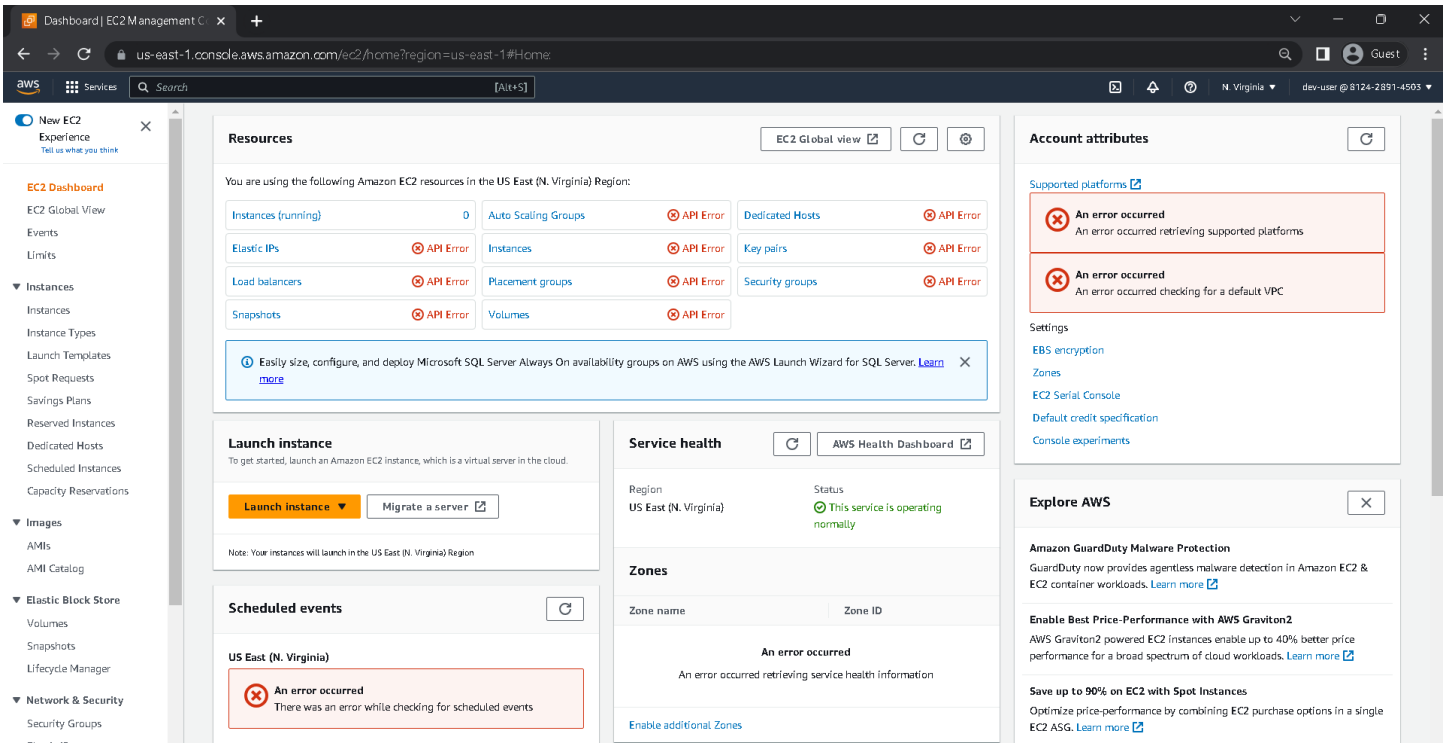
3 – User : dev-user [Enabled MFA – Added To Group Admin-Permission]

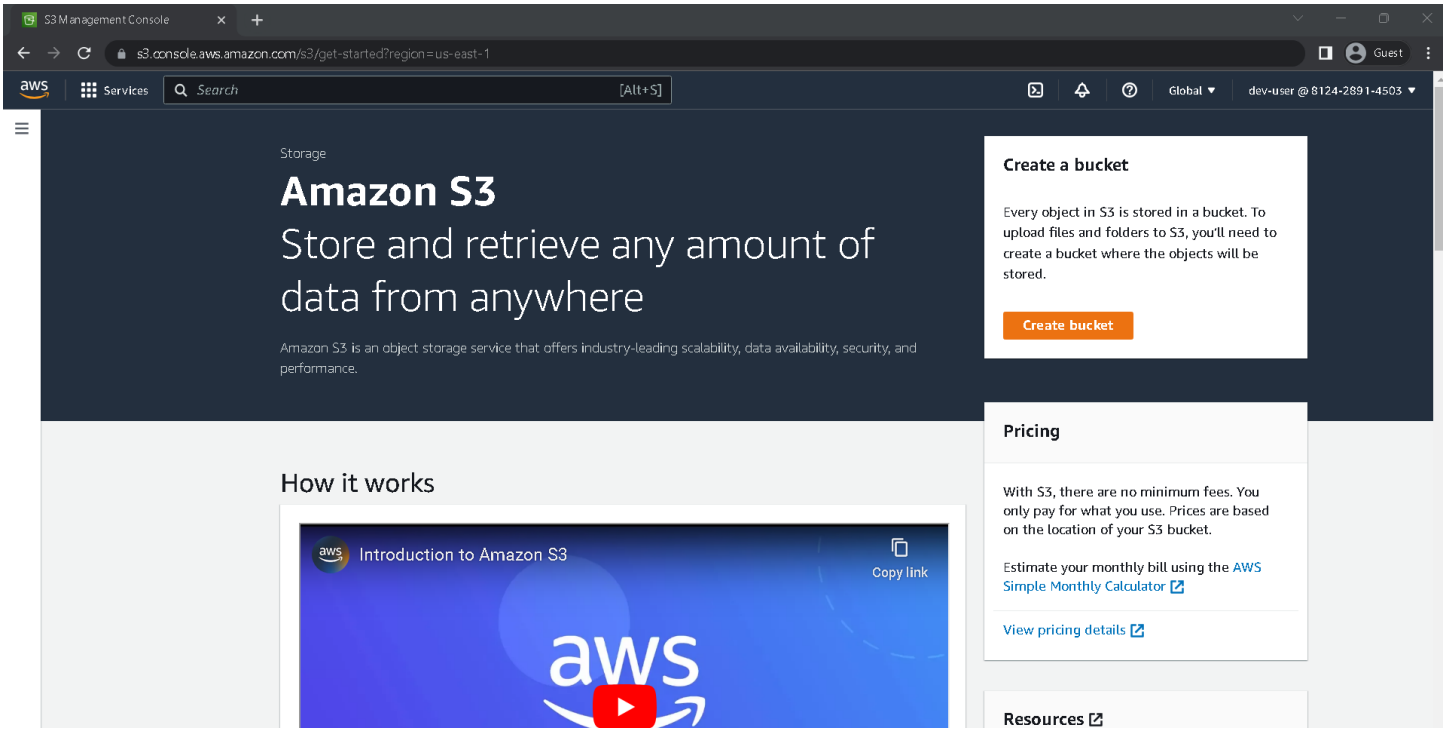


C – Access AWS using dev-user:






Dashboard | EC2 Management C
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:
Guest
Services
Search
[Alt+S]
N. Virginia
dev-user @ 8124-2891-4503
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Resources
EC2 Global view
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:
Instances (running) 0
Auto Scaling Groups API Error
Dedicated Hosts API Error
Elastic IPs API Error
Instances API Error
Key pairs API Error
Load balancers API Error
Placement groups API Error
Security groups API Error
Snapshots API Error
Volumes API Error
Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Migrate a server
Note: Your instances will launch in the US East (N. Virginia) Region
Scheduled events
US East (N. Virginia)
An error occurred
There was an error while checking for scheduled events
Service health
AWS Health Dashboard
Region
US East (N. Virginia)
Status
This service is operating normally
Zones
Zone name
Zone ID
An error occurred
An error occurred retrieving service health information
Enable additional Zones
Account attributes
Supported platforms
An error occurred
An error occurred retrieving supported platforms
An error occurred
An error occurred checking for a default VPC
Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments
Explore AWS
Amazon GuardDuty Malware Protection
GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. Learn more
Enable Best Price-Performance with AWS Graviton2
AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. Learn more
Save up to 90% on EC2 with Spot Instances
Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. Learn more


S3 Management Console
s3.console.aws.amazon.com/s3/get-started?region=us-east-1
Guest
Services
Search
[Alt+S]
Global
dev-user @ 8124-2891-4503
Storage
Amazon S3
Store and retrieve any amount of data from anywhere
Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.
Create a bucket
Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.
Create bucket
Pricing
With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.
Estimate your monthly bill using the AWS Simple Monthly Calculator
View pricing details
How it works
Introduction to Amazon S3
Copy link
Resources

D –

First I will install AWS CLI in my Ubuntu

1 -

```
root@DESKTOP-D5VVHN0:/home/amr# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 55.0M  100 55.0M    0     0  2893k      0  0:00:19  0:00:19 --:--:-- 2840k
root@DESKTOP-D5VVHN0:/home/amr# unzip awscliv2.zip
Archive:  awscliv2.zip
   creating: aws/
   creating: aws/dist/
  inflating: aws/README.md
  inflating: aws/install
  inflating: aws/THIRD_PARTY_LICENSES
```

2 –

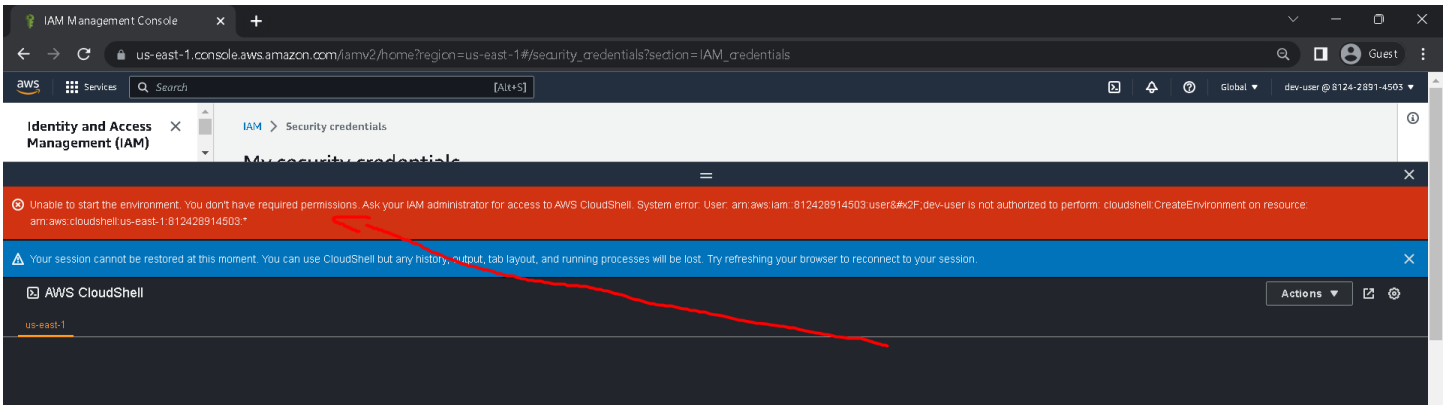
```
root@DESKTOP-D5VVHN0:/home/amr# aws --version
aws-cli/2.11.25 Python/3.11.3 Linux/5.10.102.1-microsoft-standard-WSL2 exe/x86_64.ubuntu.22 prompt/off
root@DESKTOP-D5VVHN0:/home/amr# aws configure
AWS Access Key ID [None]: AKIA32KEJN5DVYTP7FJD
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

3 - list all users From User admin-1

```
root@DESKTOP-D5VVHN0:/home/amr# aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "admin-1",
            "UserId": "AIDA32KEJN5DRVQ2S5IQ2",
            "Arn": "arn:aws:iam::812428914503:user/admin-1",
            "CreateDate": "2023-06-04T09:28:51+00:00"
        },
        {
            "Path": "/",
            "UserName": "admin2-prog",
            "UserId": "AIDA32KEJN5DS2NVCCIUI",
            "Arn": "arn:aws:iam::812428914503:user/admin2-prog",
            "CreateDate": "2023-06-04T09:30:05+00:00"
        },
        {
            "Path": "/",
            "UserName": "dev-user",
            "UserId": "AIDA32KEJN5DS2MZJ3FJT",
            "Arn": "arn:aws:iam::812428914503:user/dev-user",
            "CreateDate": "2023-06-04T09:30:46+00:00"
        }
    ]
}
(END)
```

4 - list all groups From User admin-1

```
root@DESKTOP-D5VVHN0:/home/amr# aws iam list-groups
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "Admin-Permisssion",
            "GroupId": "AGPA32KEJN5D3FL26HDUR",
            "Arn": "arn:aws:iam::812428914503:group/Admin-Permisssion",
            "CreateDate": "2023-06-04T09:04:39+00:00"
        },
        {
            "Path": "/",
            "GroupName": "Development-Permision",
            "GroupId": "AGPA32KEJN5D6QM6ODCRR",
            "Arn": "arn:aws:iam::812428914503:group/Development-Permision",
            "CreateDate": "2023-06-04T09:06:33+00:00"
        }
    ]
}
root@DESKTOP-D5VVHN0:/home/amr#
```

E - Trying to listing users and groups from "dev-user" user

1 – login to user : dev-user