

AWS LAB1:

1. Create aws account and set billing alarm
2. create 2 groups one admin and one for development
 1. in the admin group it has admin permission , and in the development only access to s3
 2. create admin-1 user console access and mfa enabled in admin group
 3. and admin2-prog with cli access only and list all users and groups using commands not console
 4. 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)
 5. Also access cli using dev-user and try to get all users and groups using it
3. Create ec2 and install apache2 on it.

4. Required:

1. Screenshot from each group with users and permissions attached to it

The screenshot shows the AWS IAM console for the 'admin-group'. The breadcrumb navigation is 'IAM > User groups > admin-group'. The group name is 'admin-group'. The summary section shows the group name, creation time (May 09, 2023, 13:58 (UTC+02:00)), and ARN (arn:aws:iam::385582076770:group/admin-group). The 'Users' tab is selected, showing two users: 'admin-2' and 'admin-1'. Both users have a status of '1' and 'None' for last activity, and were created '1 hour ago'.

User name	Groups	Last activity	Creation time
admin-2	1	None	1 hour ago
admin-1	1	None	1 hour ago

The screenshot shows the AWS IAM console for the 'admin-group'. The breadcrumb navigation is 'IAM > User groups > admin-group'. The group name is 'admin-group'. The summary section shows the group name, creation time (May 09, 2023, 13:58 (UTC+02:00)), and ARN (arn:aws:iam::385582076770:group/admin-group). The 'Permissions' tab is selected, showing one policy: 'AdministratorAccess'. The policy is an 'AWS managed - job function' and 'Provides full access to AWS se'.

Policy name	Type	Description
AdministratorAccess	AWS managed - job function	Provides full access to AWS se

[Alt+S]

🔍

🔔

🔗

Global ▾

Hadi Al-Atally ▾

IAM > User groups > development_group

development_group

Delete

Summary

Edit

User group name development_group	Creation time May 09, 2023, 14:07 (UTC+02:00)	ARN arn:aws:iam::385582076770:group/development_group
--------------------------------------	--	--

Users

Permissions

Access Advisor

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

🔄

Simulate

Remove

Add permissions ▾

🔍

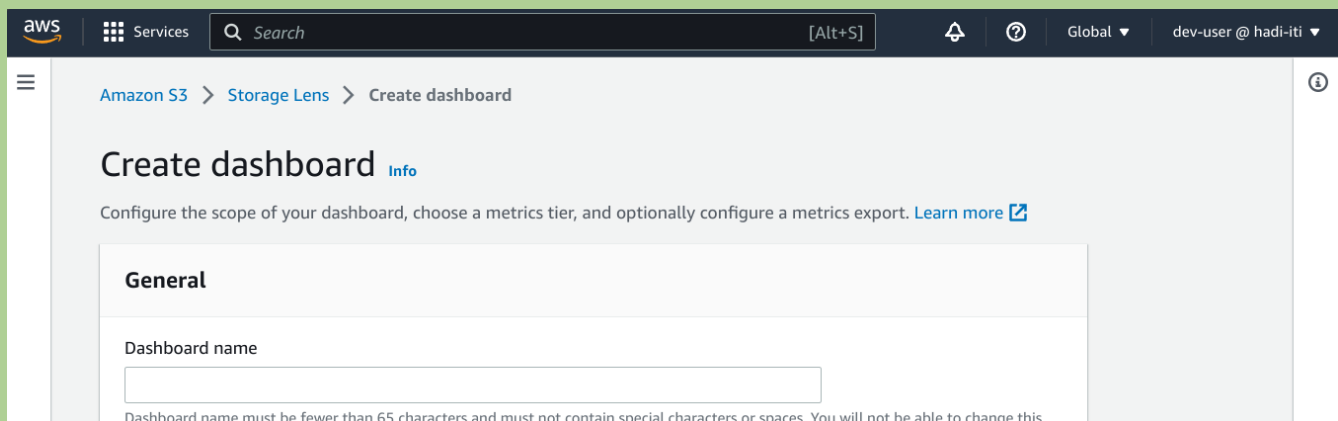
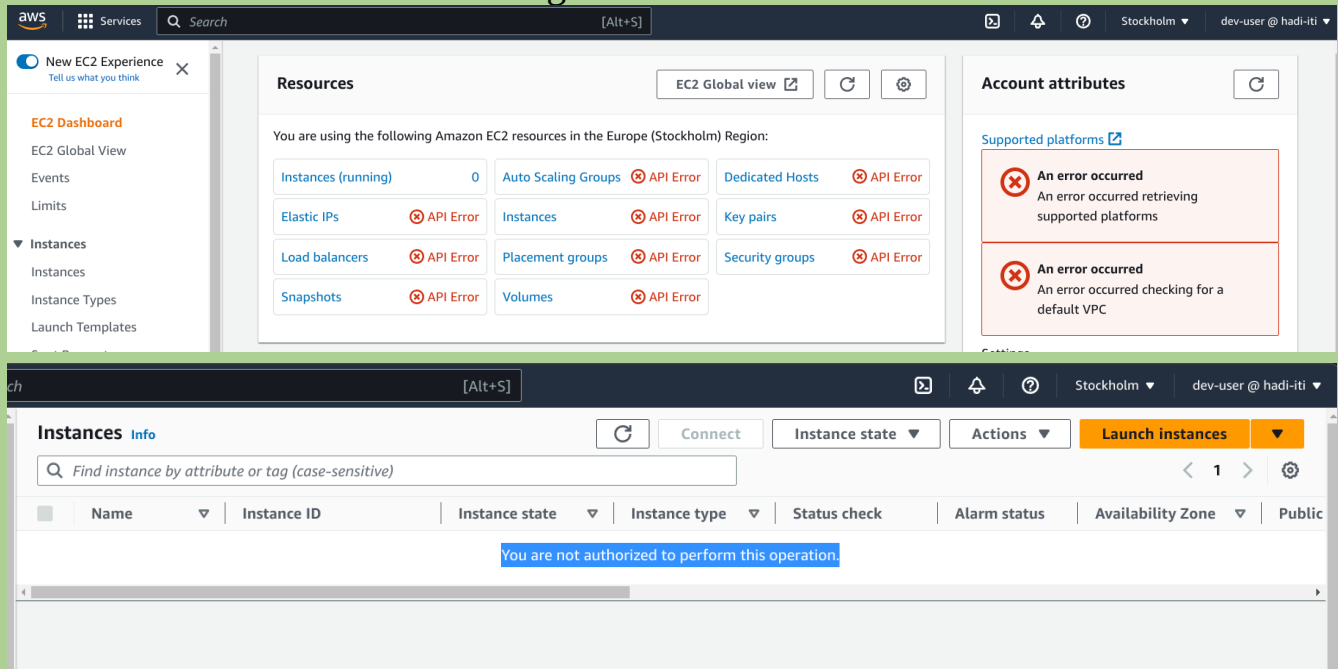
Filter policies by property or policy name and press enter.

< 1 >

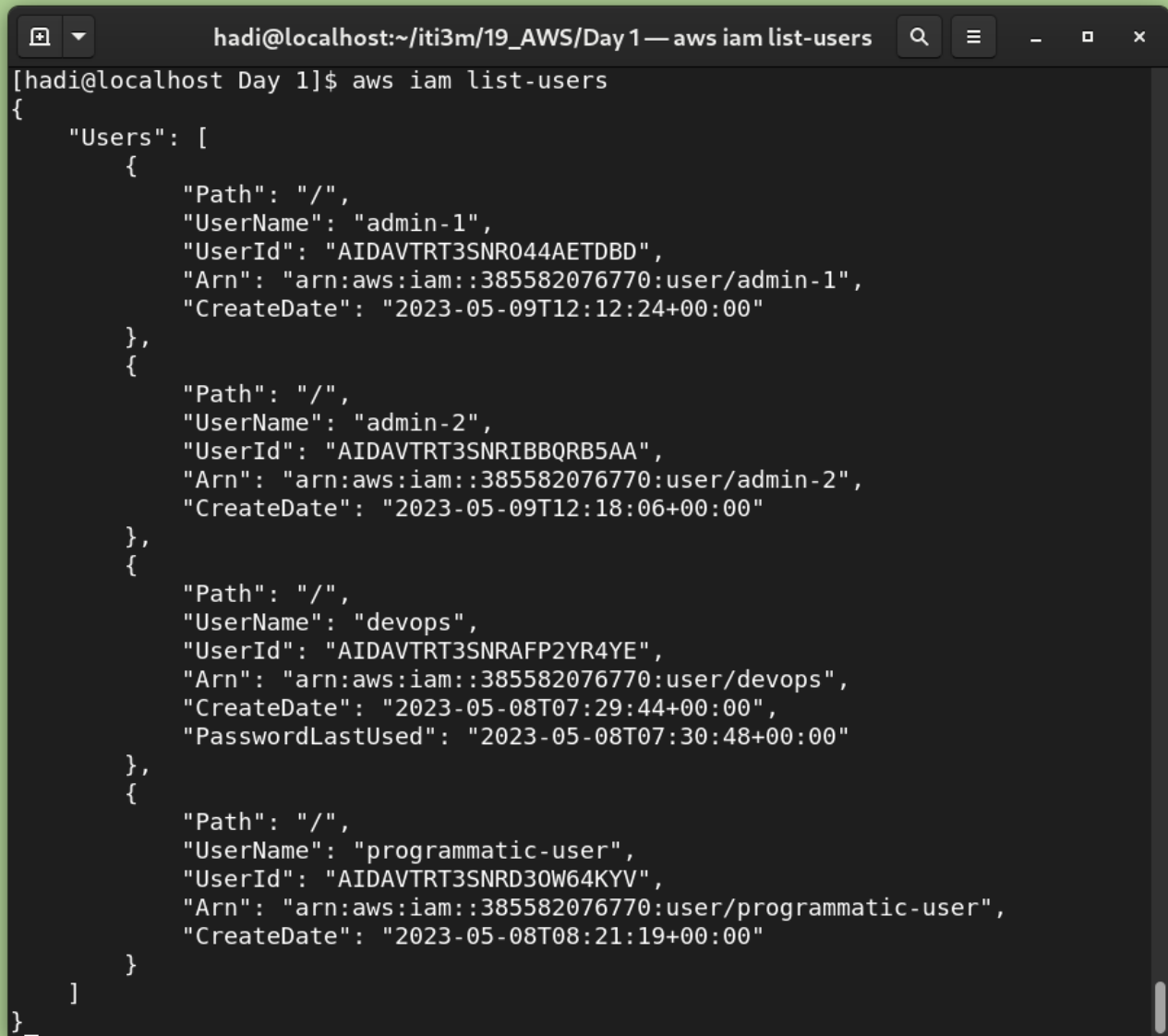
⚙️

<input type="checkbox"/>	Policy name 🔗	Type ▾	Description
<input type="checkbox"/>	🔗 s3_full_access_policy	Customer managed	

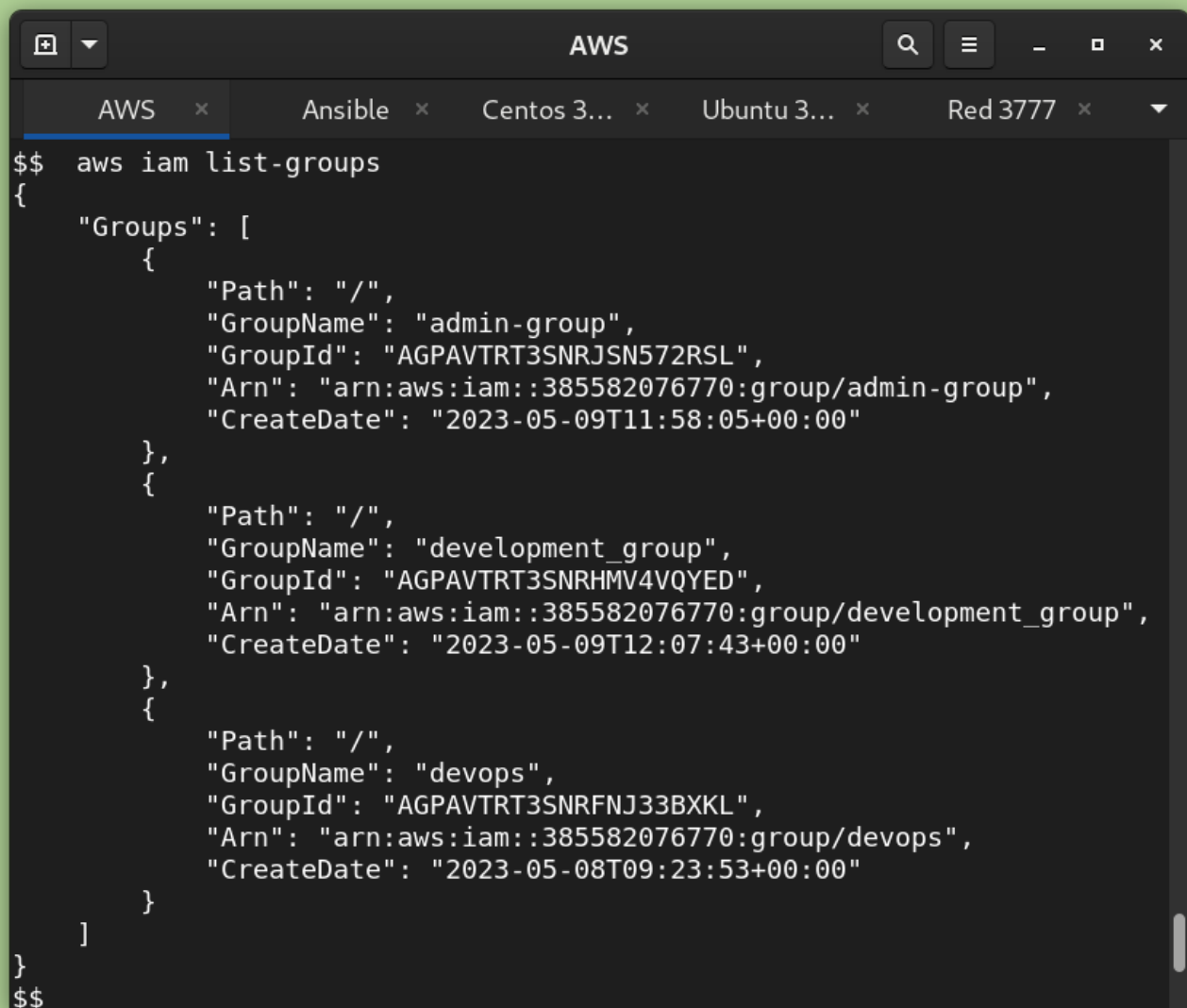
2. Screenshot from using dev-user to access ec2 and s3 from console



3. Screenshot from listing users and groups using admin2-prog

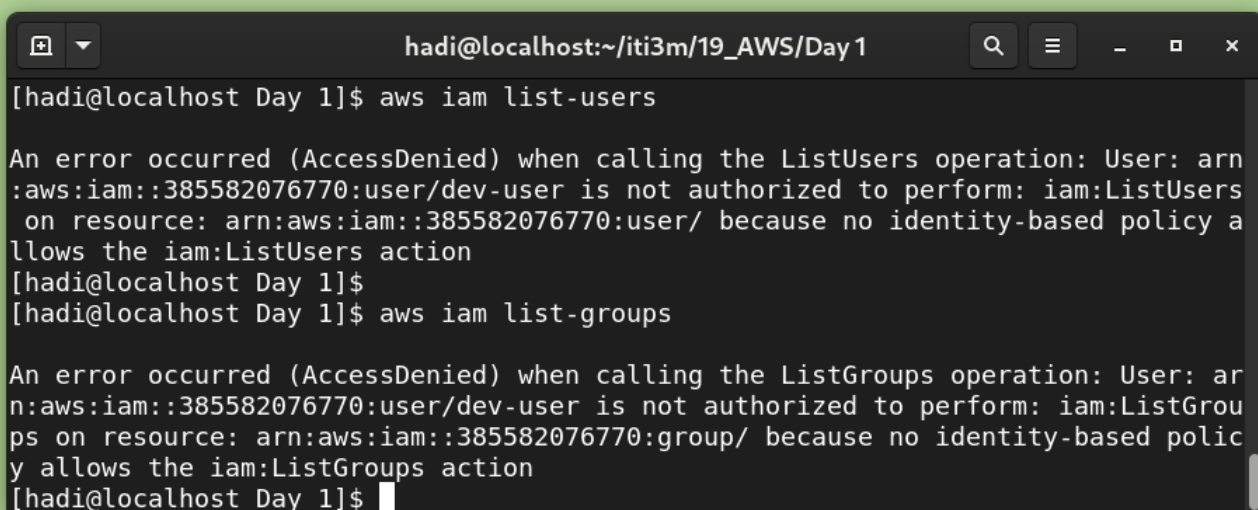


```
hadi@localhost:~/iti3m/19_AWS/Day 1 — aws iam list-users
[hadi@localhost Day 1]$ aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "admin-1",
      "UserId": "AIDAVTRT3SNR044AETDBD",
      "Arn": "arn:aws:iam::385582076770:user/admin-1",
      "CreateDate": "2023-05-09T12:12:24+00:00"
    },
    {
      "Path": "/",
      "UserName": "admin-2",
      "UserId": "AIDAVTRT3SNRIBBQRB5AA",
      "Arn": "arn:aws:iam::385582076770:user/admin-2",
      "CreateDate": "2023-05-09T12:18:06+00:00"
    },
    {
      "Path": "/",
      "UserName": "devops",
      "UserId": "AIDAVTRT3SNRAFP2YR4YE",
      "Arn": "arn:aws:iam::385582076770:user/devops",
      "CreateDate": "2023-05-08T07:29:44+00:00",
      "PasswordLastUsed": "2023-05-08T07:30:48+00:00"
    },
    {
      "Path": "/",
      "UserName": "programmatic-user",
      "UserId": "AIDAVTRT3SNRD30W64KYV",
      "Arn": "arn:aws:iam::385582076770:user/programmatic-user",
      "CreateDate": "2023-05-08T08:21:19+00:00"
    }
  ]
}
```



```
aws iam list-groups
{
  "Groups": [
    {
      "Path": "/",
      "GroupName": "admin-group",
      "GroupId": "AGPAVTRT3SNRJSN572RSL",
      "Arn": "arn:aws:iam::385582076770:group/admin-group",
      "CreateDate": "2023-05-09T11:58:05+00:00"
    },
    {
      "Path": "/",
      "GroupName": "development_group",
      "GroupId": "AGPAVTRT3SNRHMV4VQYED",
      "Arn": "arn:aws:iam::385582076770:group/development_group",
      "CreateDate": "2023-05-09T12:07:43+00:00"
    },
    {
      "Path": "/",
      "GroupName": "devops",
      "GroupId": "AGPAVTRT3SNRFNJ33BXKL",
      "Arn": "arn:aws:iam::385582076770:group/devops",
      "CreateDate": "2023-05-08T09:23:53+00:00"
    }
  ]
}
```


4. Screenshot from listing users and groups using dev-users




```
hadi@localhost:~/iti3m/19_AWS/Day1
[hadi@localhost Day 1]$ aws iam list-users
An error occurred (AccessDenied) when calling the ListUsers operation: User: arn:aws:iam::385582076770:user/dev-user is not authorized to perform: iam:ListUsers on resource: arn:aws:iam::385582076770:user/ because no identity-based policy allows the iam:ListUsers action
[hadi@localhost Day 1]$
[hadi@localhost Day 1]$ aws iam list-groups
An error occurred (AccessDenied) when calling the ListGroups operation: User: arn:aws:iam::385582076770:user/dev-user is not authorized to perform: iam:ListGroups on resource: arn:aws:iam::385582076770:group/ because no identity-based policy allows the iam:ListGroups action
[hadi@localhost Day 1]$
```

5. Required: screenshot From accessing the machine public ip from the browser C2 General

4. Connect to your instance using its Public IP:

 54.86.174.79

Example:

 `ssh -i "aws_key.pem" ec2-user@54.86.174.79`

