
ATTACK
DEFENSE
by PentesterAcademy

Name	IAM Enumeration
URL	https://attackdefense.com/challengedetails?cid=2245
Type	AWS Cloud Security : IAM

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Solution:

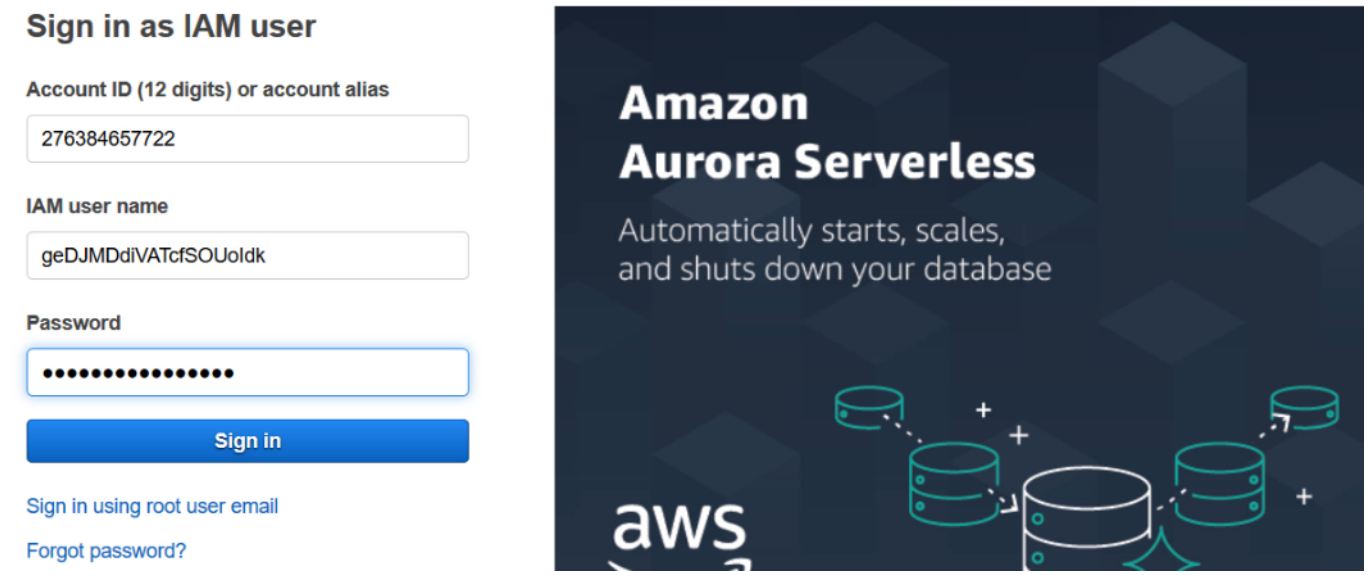
Console Based Enumeration

Step 1: Click on the lab link button to get access to the AWS lab credentials.

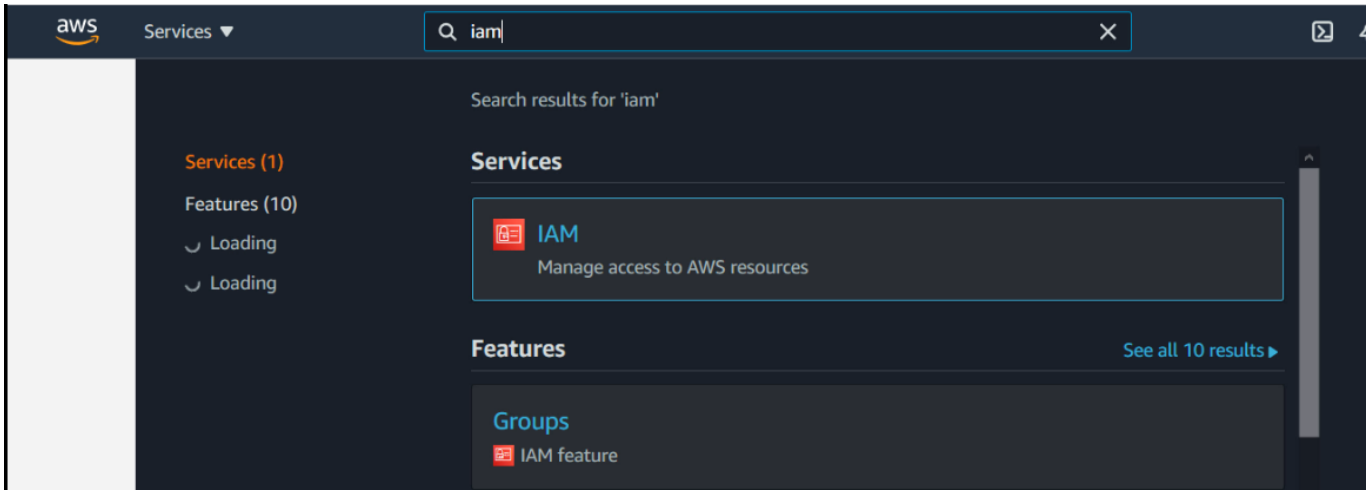
Access Credentials to your AWS lab Account

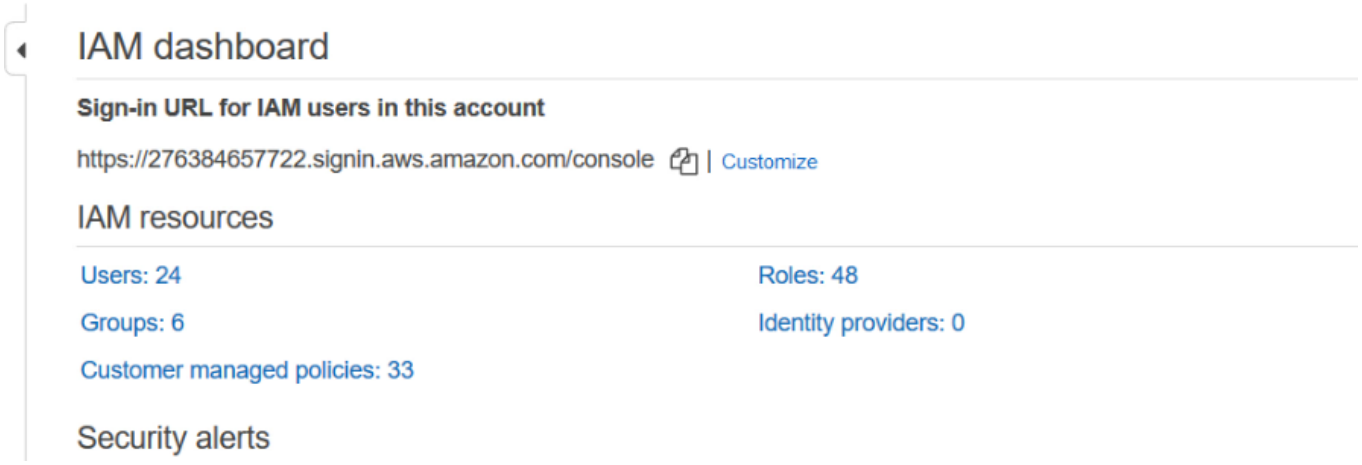
Login URL	https://276384657722.signin.aws.amazon.com/console
Region	Asia Pacific (Singapore) ap-southeast-1
Username	geDJMDdiVATcfSOUoIdk
Password	dG3v6mFrNFY2Tcir
Access Key ID	AKIAUAWOPGE5DA2GTYSQ
Secret Access Key	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

Step 2: Sign-in into the AWS console.

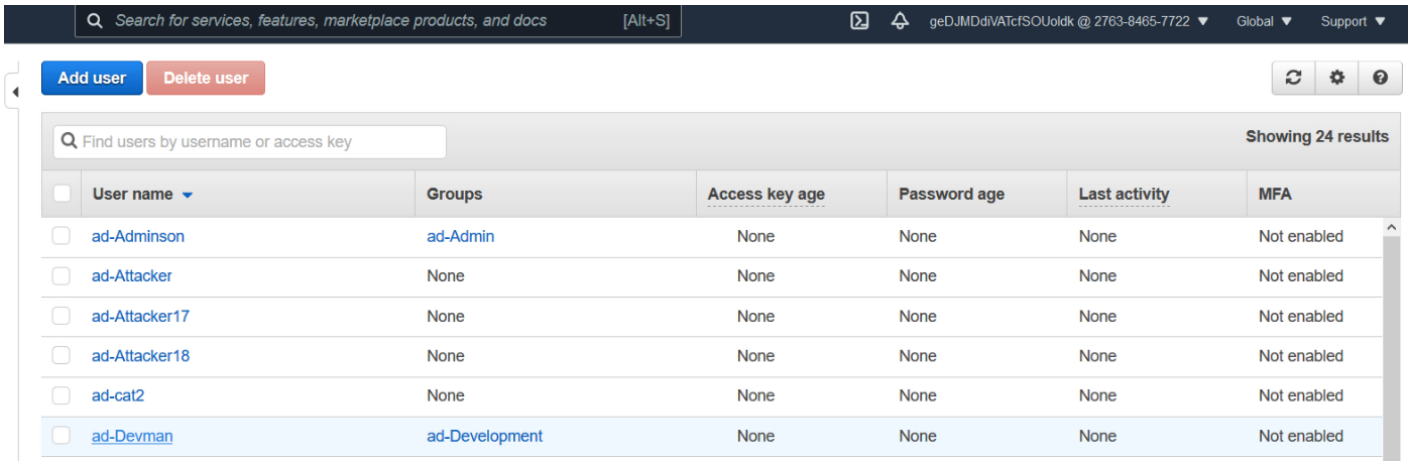


Step 3: Search for the IAM dashboard and navigate to it.

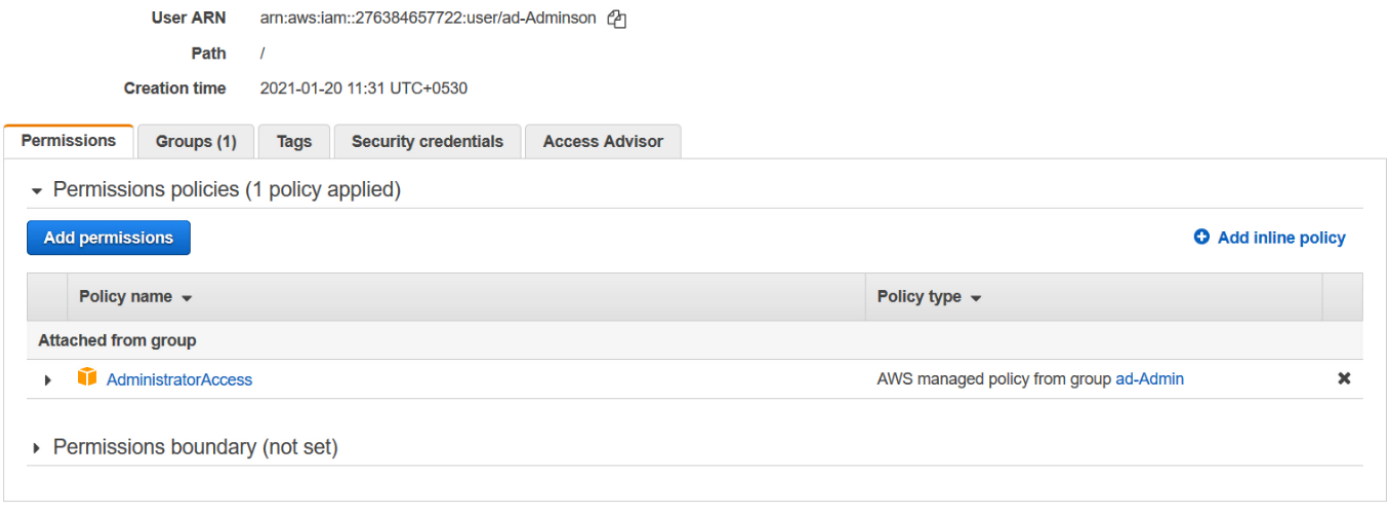




Step 4: Click on users.

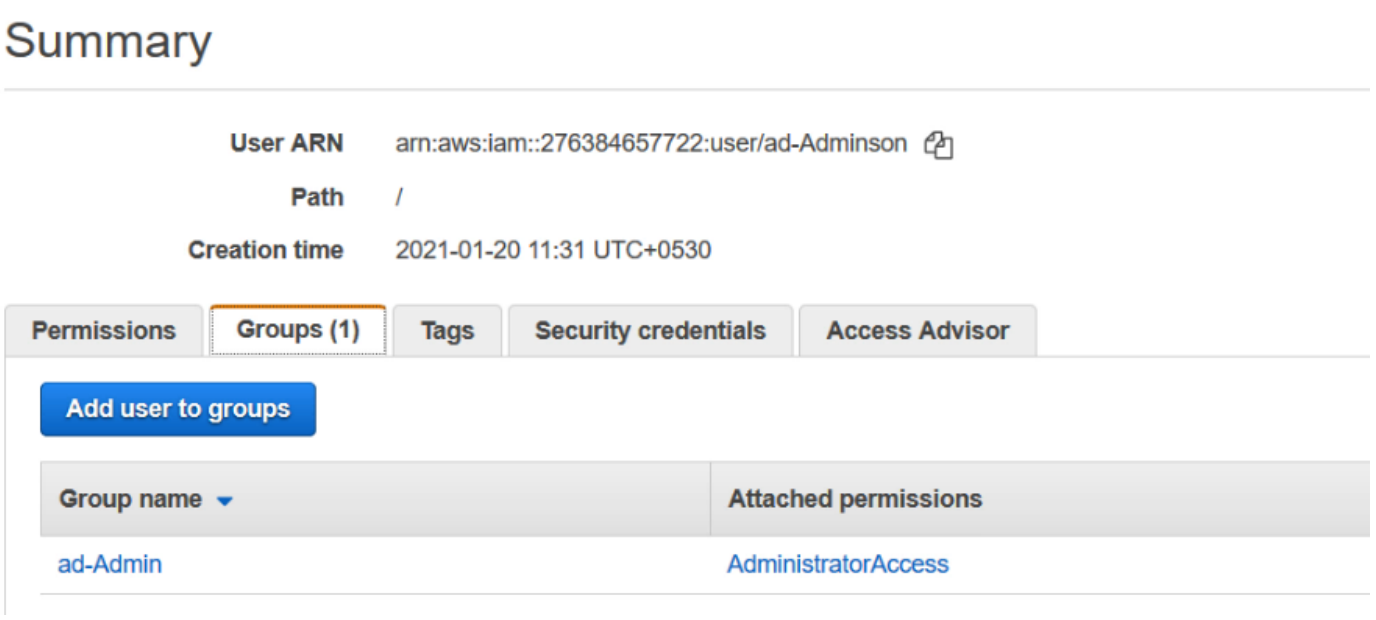


Step 5: To enumerate the user click on the username.

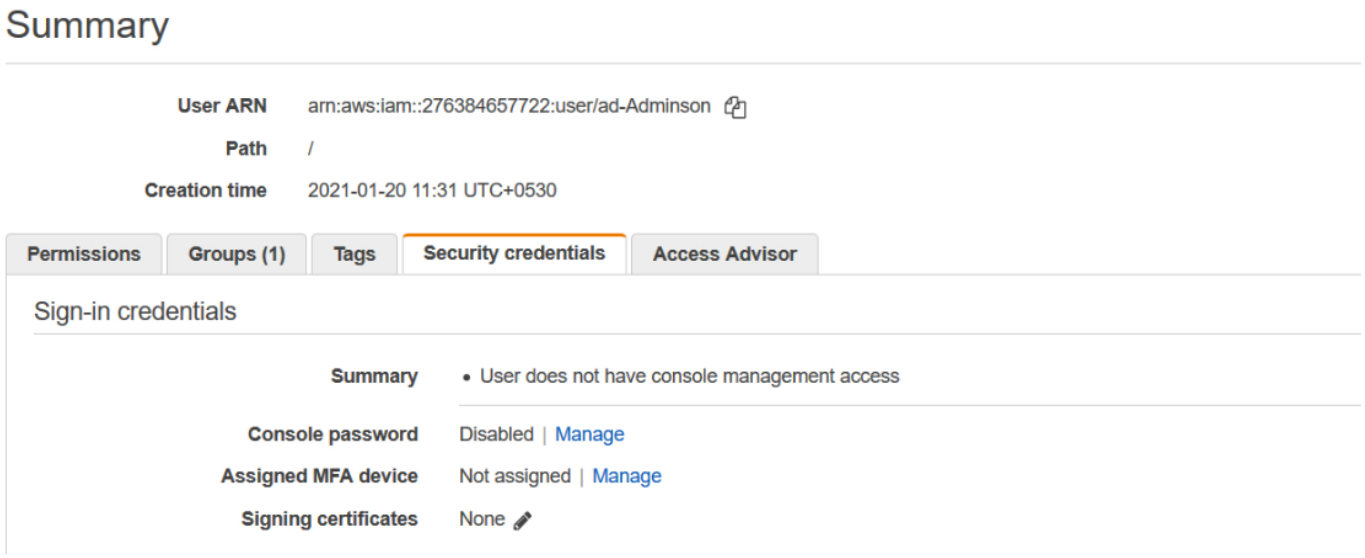


Check user permission and policies.

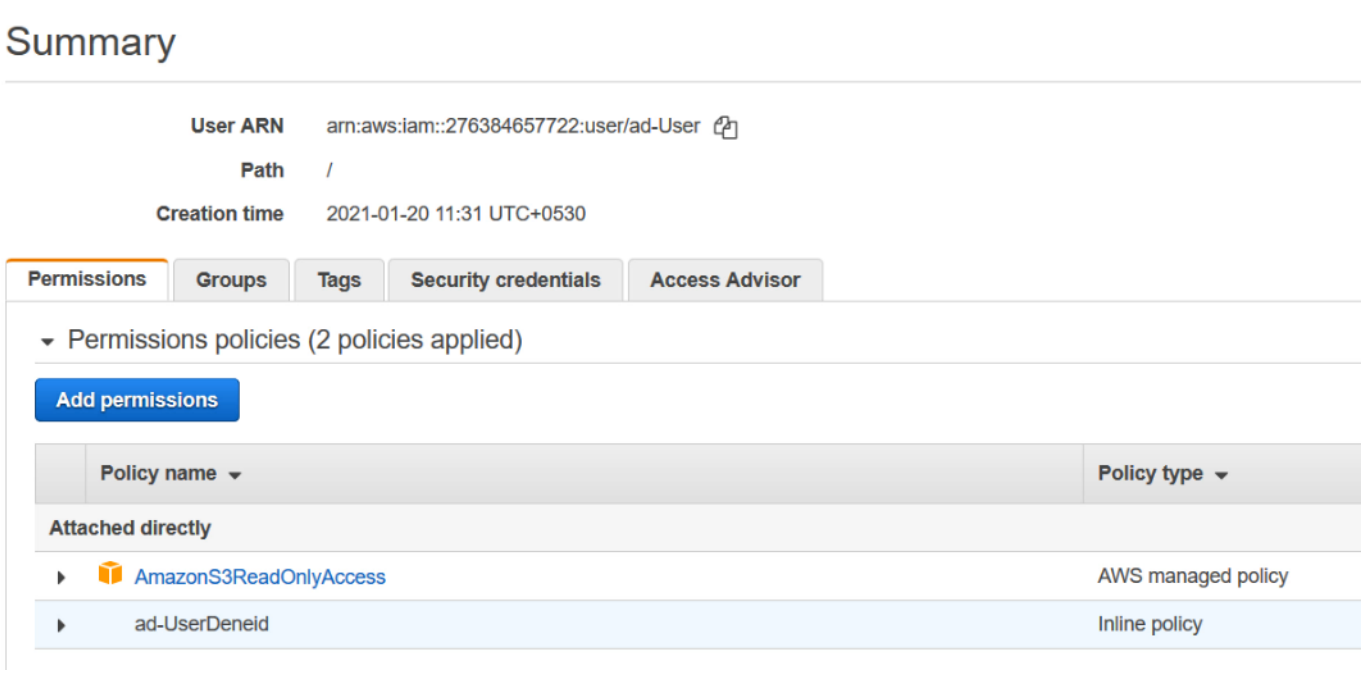
Step 6: Check groups for the user.



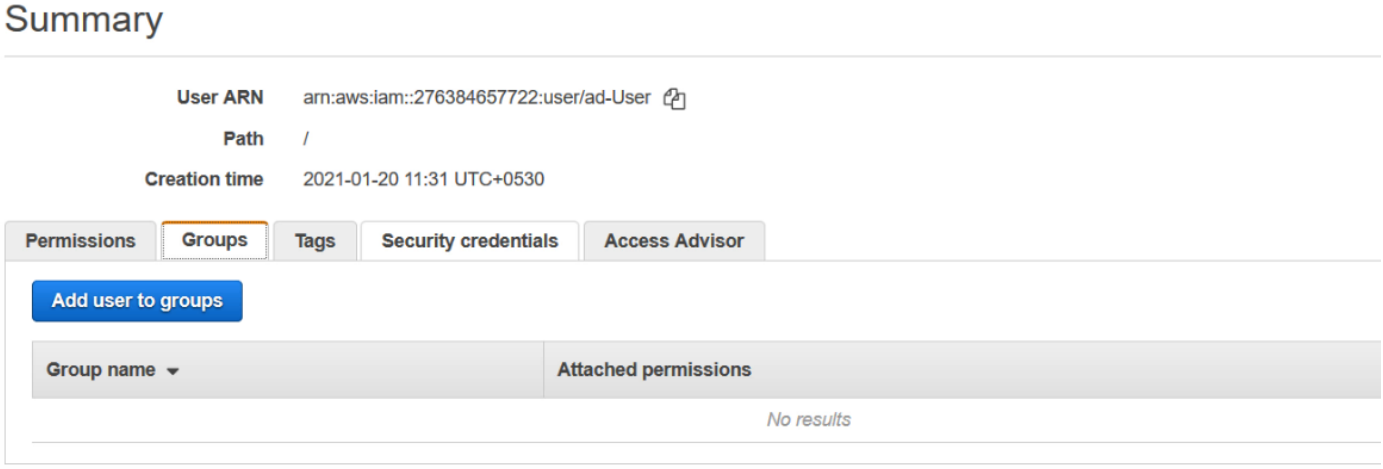
Step 7: Check the user's security credentials.



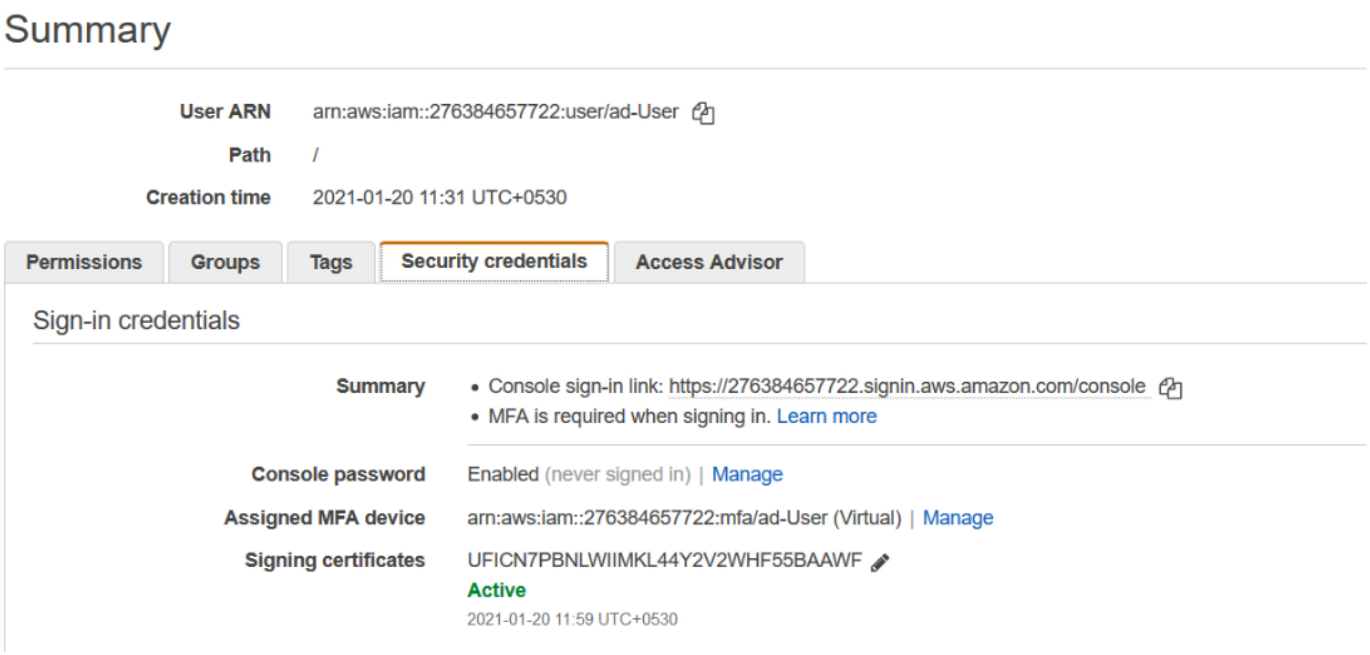
Step 8: Similarly check for other user's permissions.



Step 9: Check groups for the user.



Step 10: Check the user's security credentials.



Step 11: Check the user's access keys and ssh keys.

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

Create access key

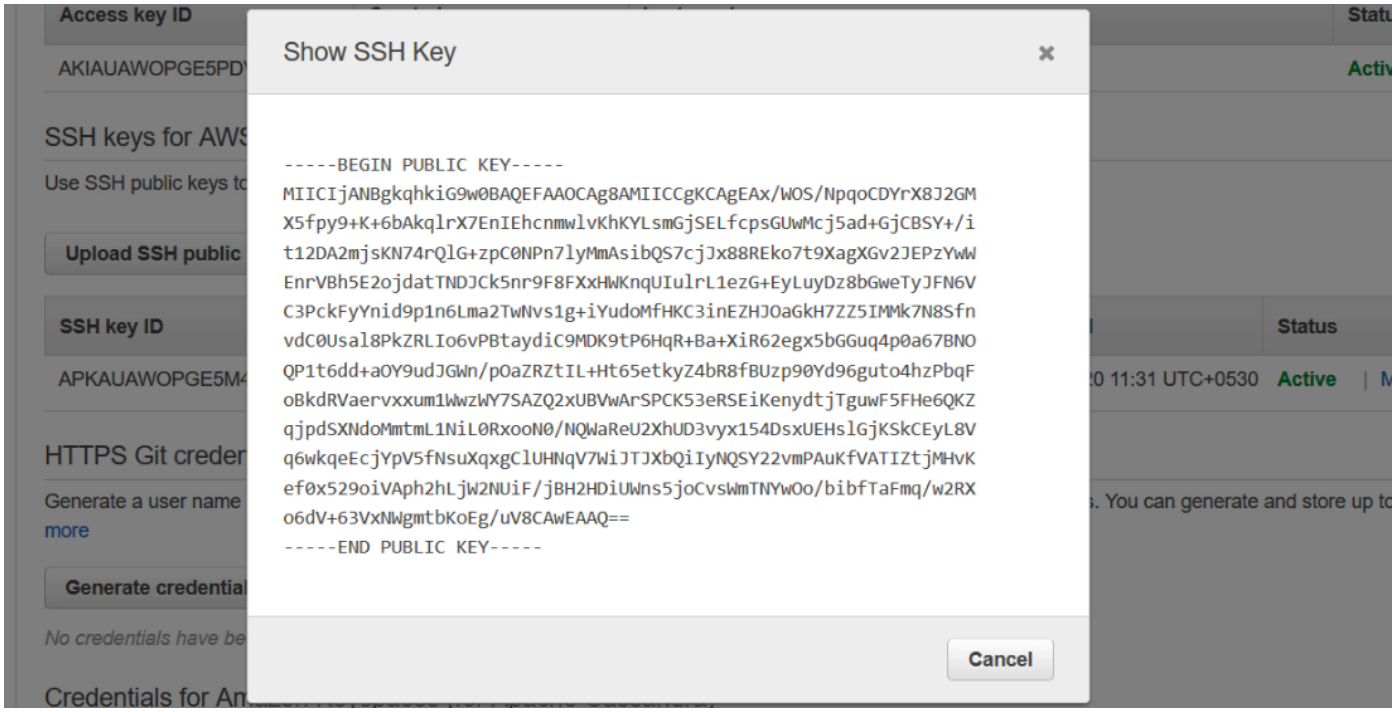
Access key ID	Created	Last used	Status
AKIAUAWOPGE5PDVCCLFW	2021-01-20 11:31 UTC+0530	N/A	Active | Make inactive

SSH keys for AWS CodeCommit

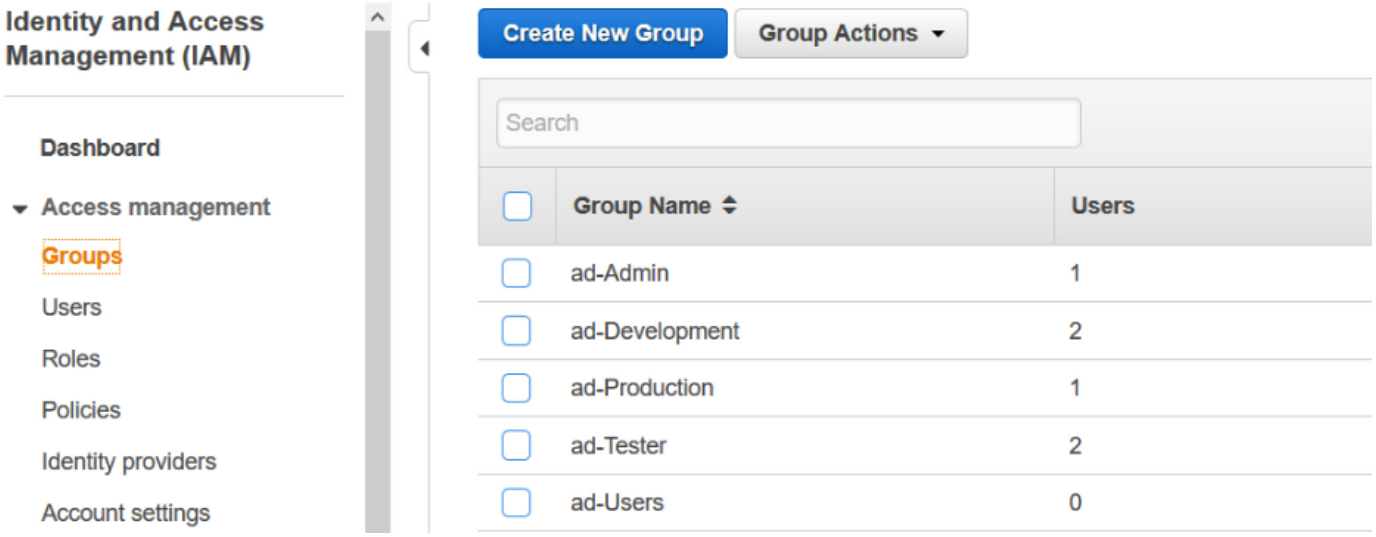
Use SSH public keys to authenticate access to AWS CodeCommit repositories. Learn more

Upload SSH public key

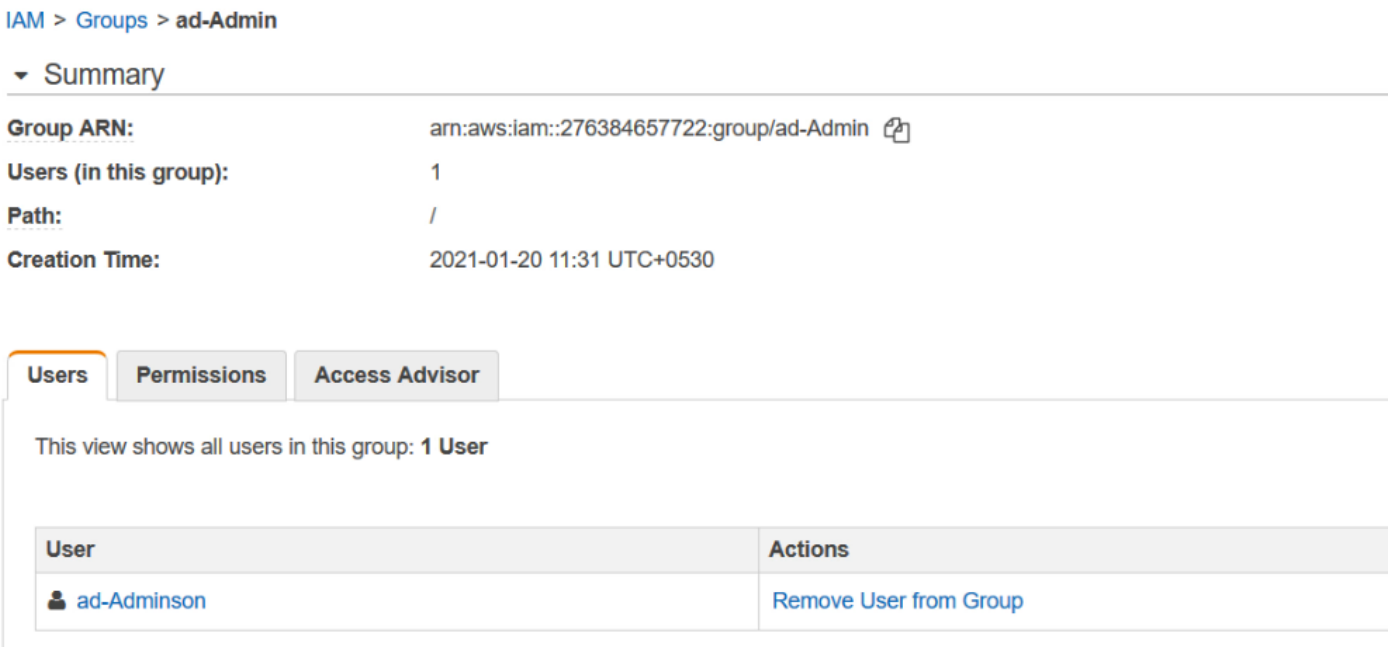
SSH key ID	Uploaded	Status
APKAUAWOPGE5M47NZEIT | Show SSH key	2021-01-20 11:31 UTC+0530	Active | Make inactive



Step 12: Click on groups on the left panel to enumerate groups.

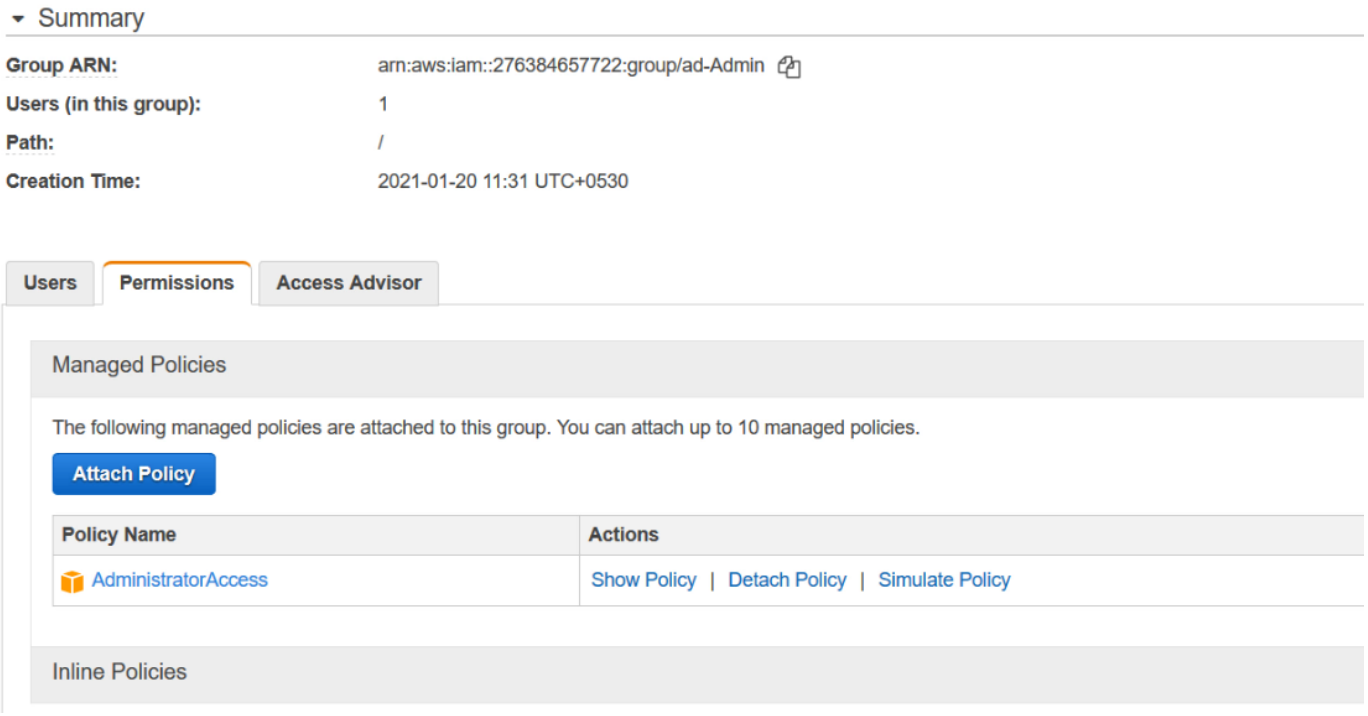


Step 13: Click on the group name to open group details.

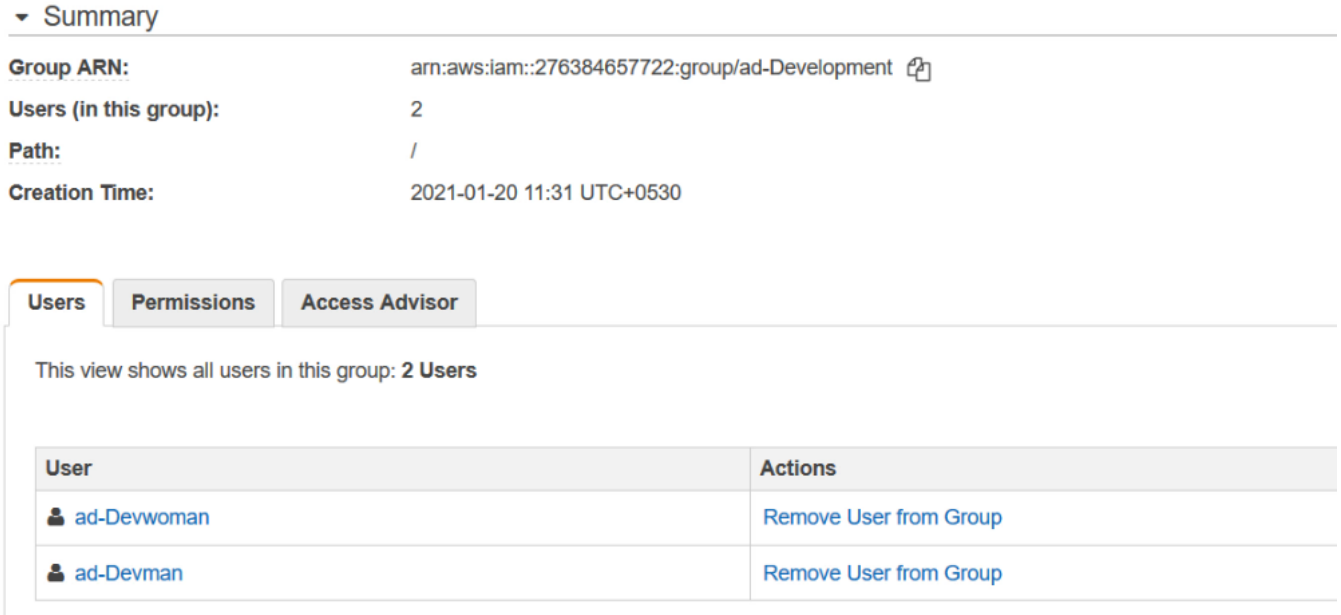


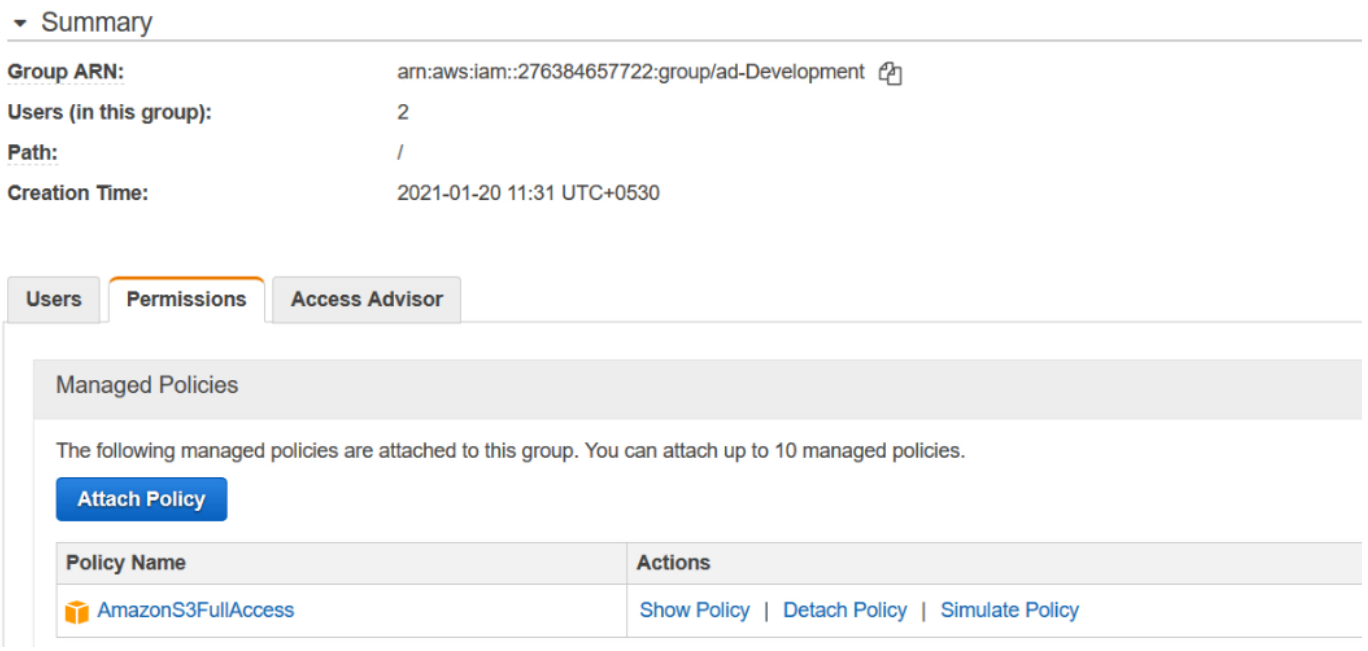
Check for the users that are part of the group.

Step 14: Check for the group attached policies.

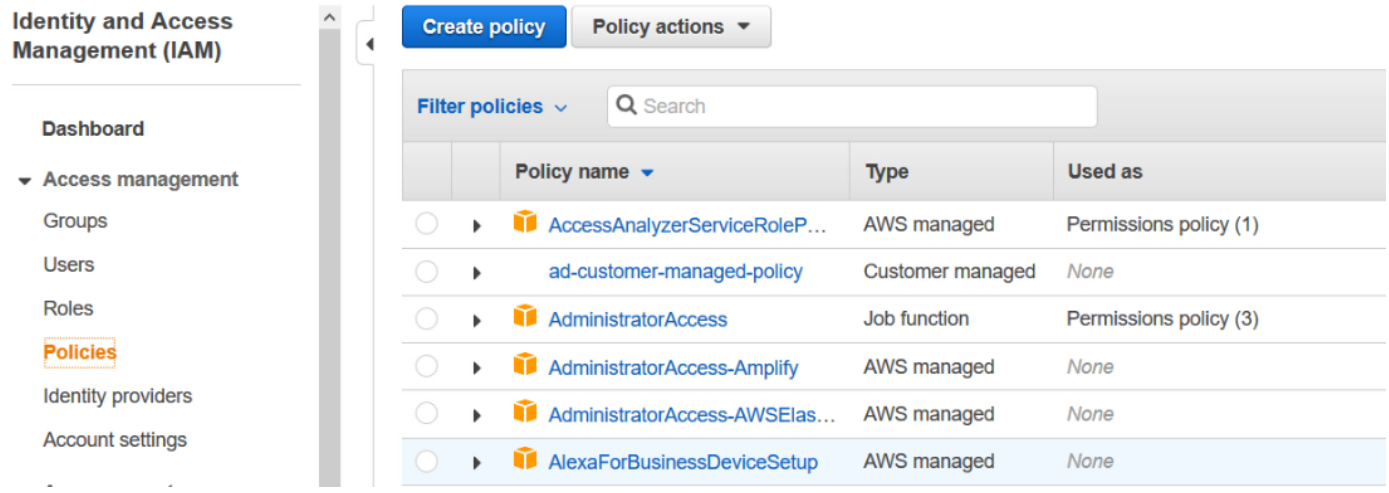


Step 15: Similarly check for the other groups.

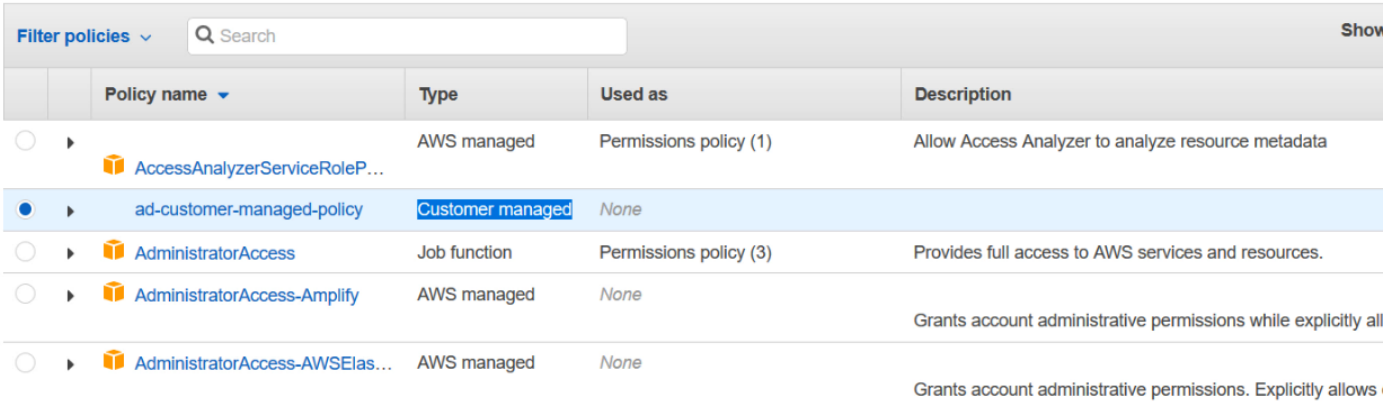




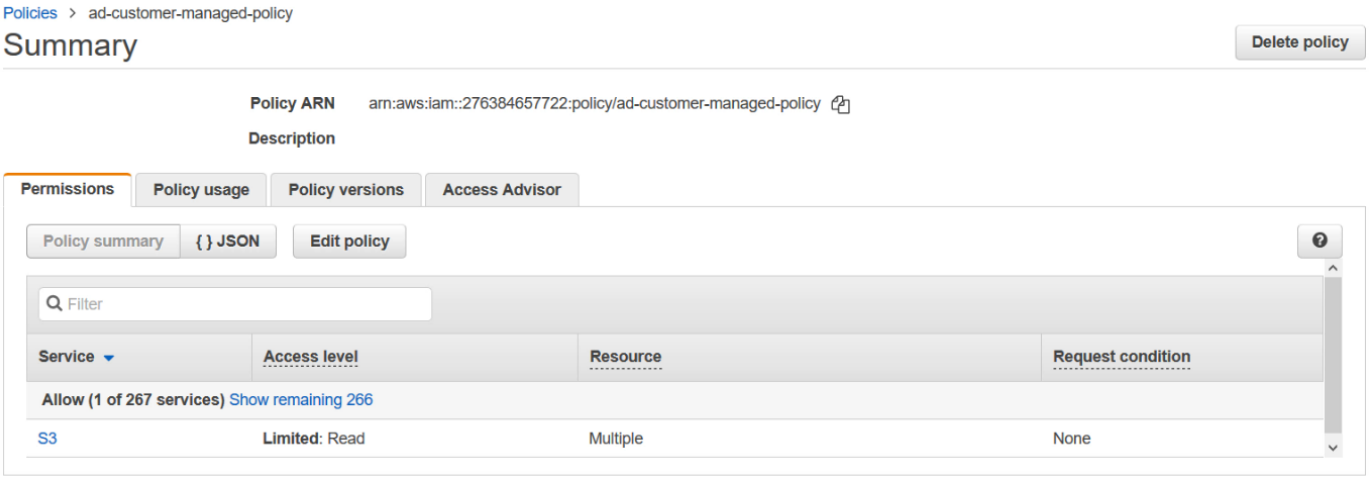
Step 16: Click on policies in the left pane to enumerate policies.



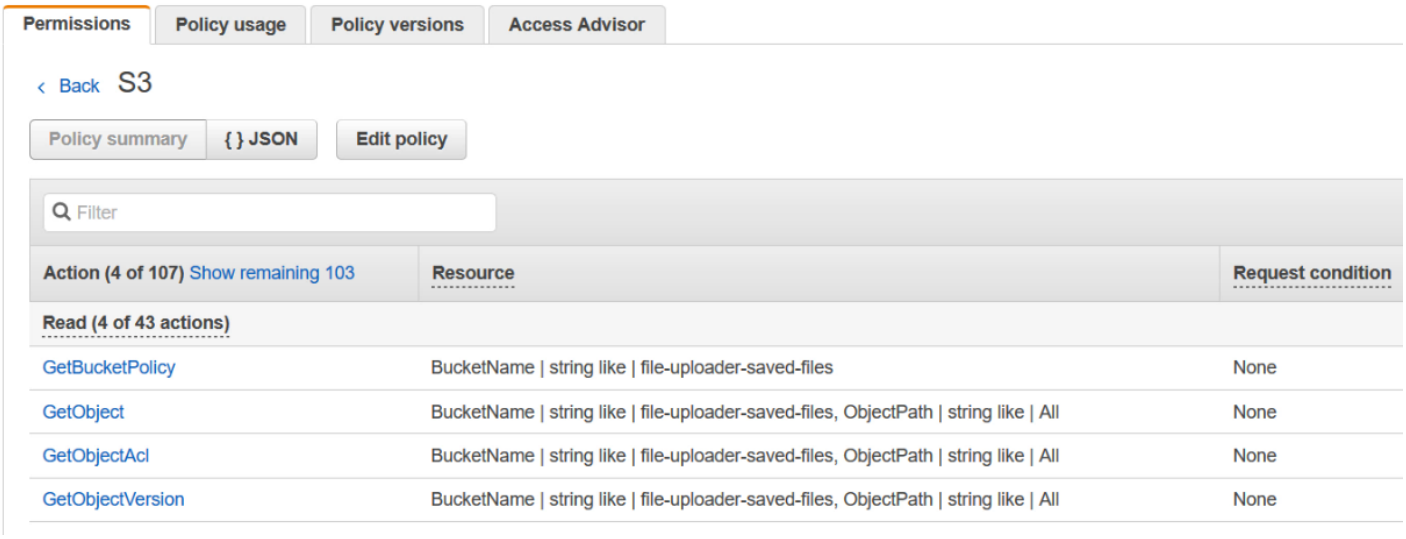
Steps 17: Check for customer-managed and AWS managed policies.

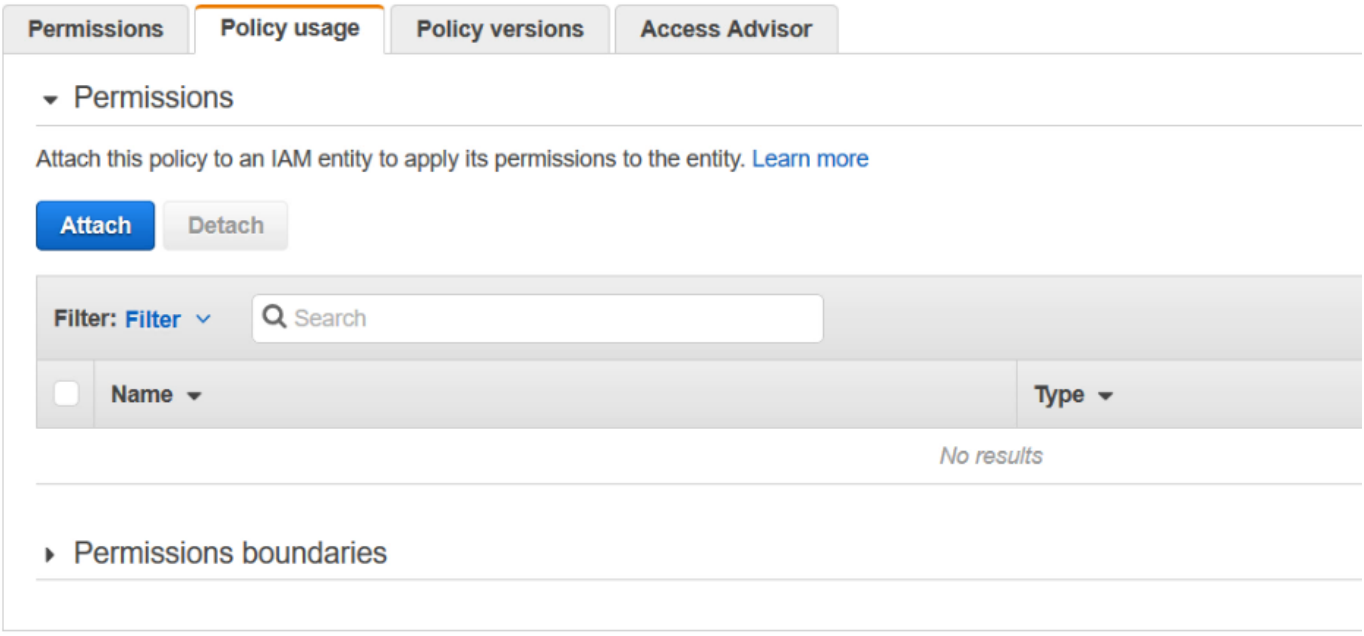


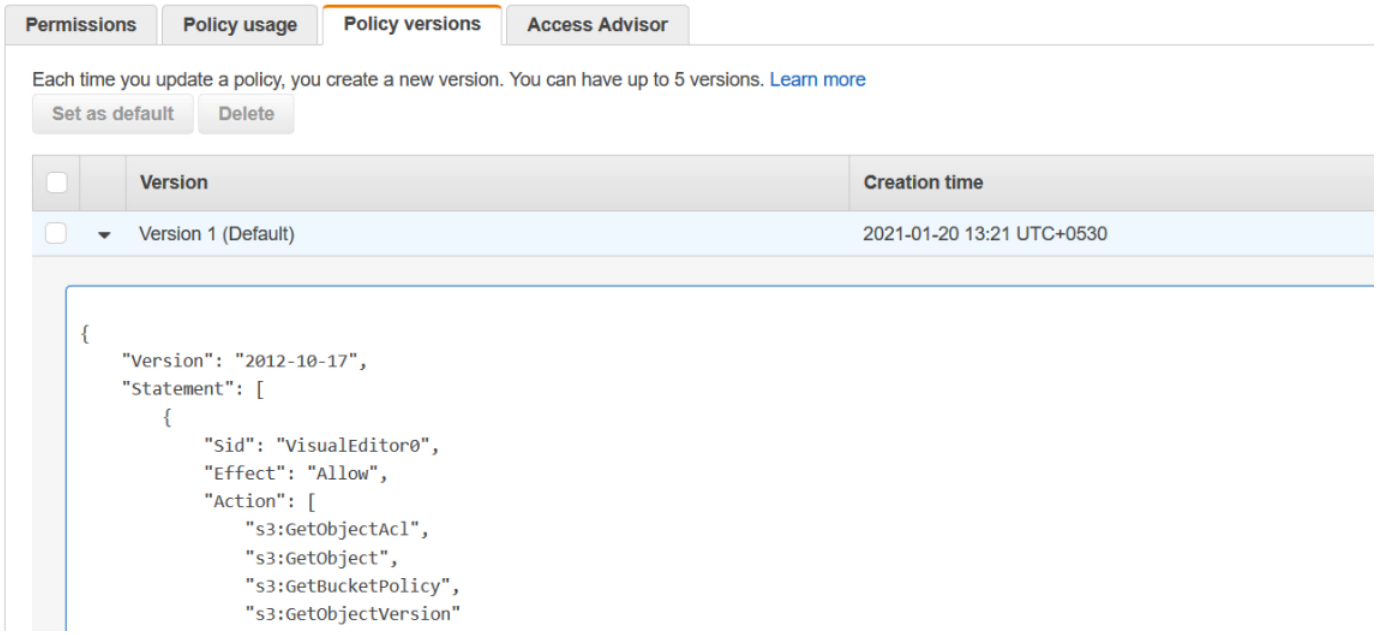
Step 18: Click on the policy name to enumerate the policy.



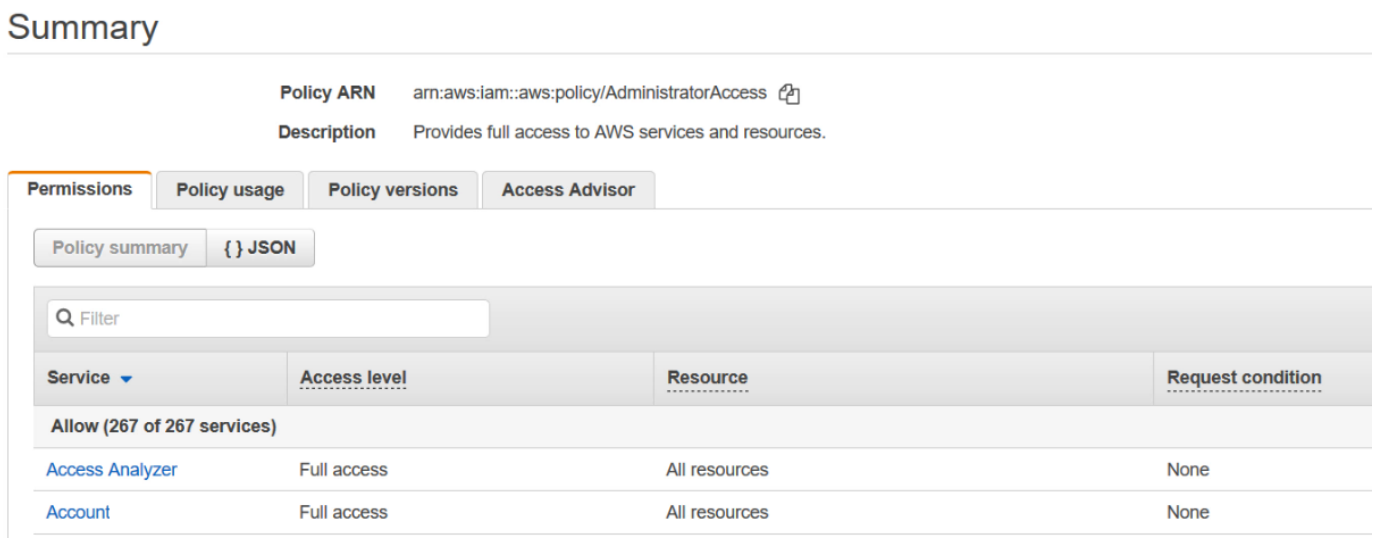
Step 19: Check policy document, policy usages and policy permissions.


Permissions
Policy usage
Policy versions
Access Advisor
Back S3
Policy summary
{ } JSON
Edit policy
Filter
Action (4 of 107) Show remaining 103
Resource
Request condition
Read (4 of 43 actions)
GetBucketPolicy
BucketName | string like | file-uploader-saved-files
None
GetObject
BucketName | string like | file-uploader-saved-files, ObjectPath | string like | All
None
GetObjectAcl
BucketName | string like | file-uploader-saved-files, ObjectPath | string like | All
None
GetObjectVersion
BucketName | string like | file-uploader-saved-files, ObjectPath | string like | All
None

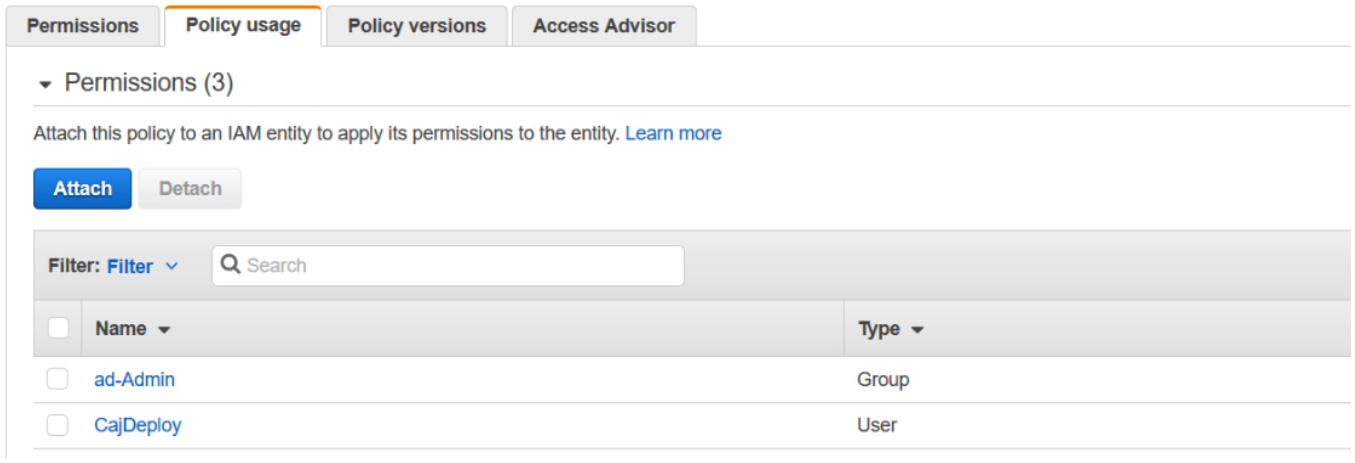

Permissions
Policy usage
Policy versions
Access Advisor
Permissions
Attach this policy to an IAM entity to apply its permissions to the entity. Learn more
Attach
Detach
Filter: Filter
Search
Name
Type
No results
Permissions boundaries

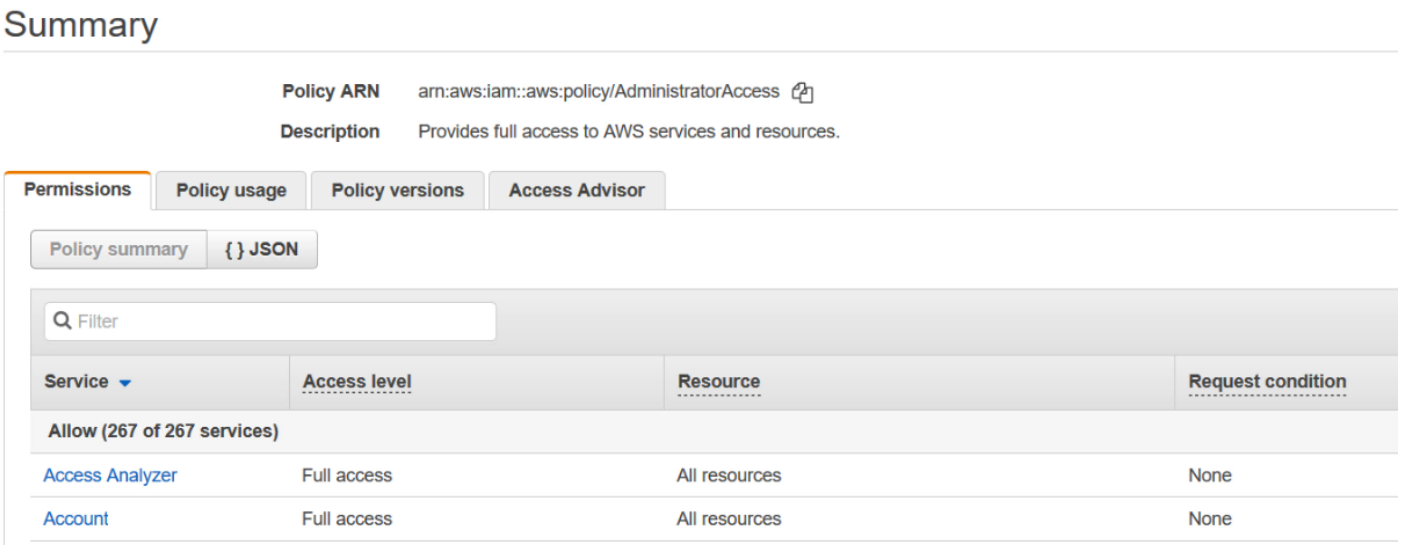


Step 20: Similarly enumerate other AWS managed policies.

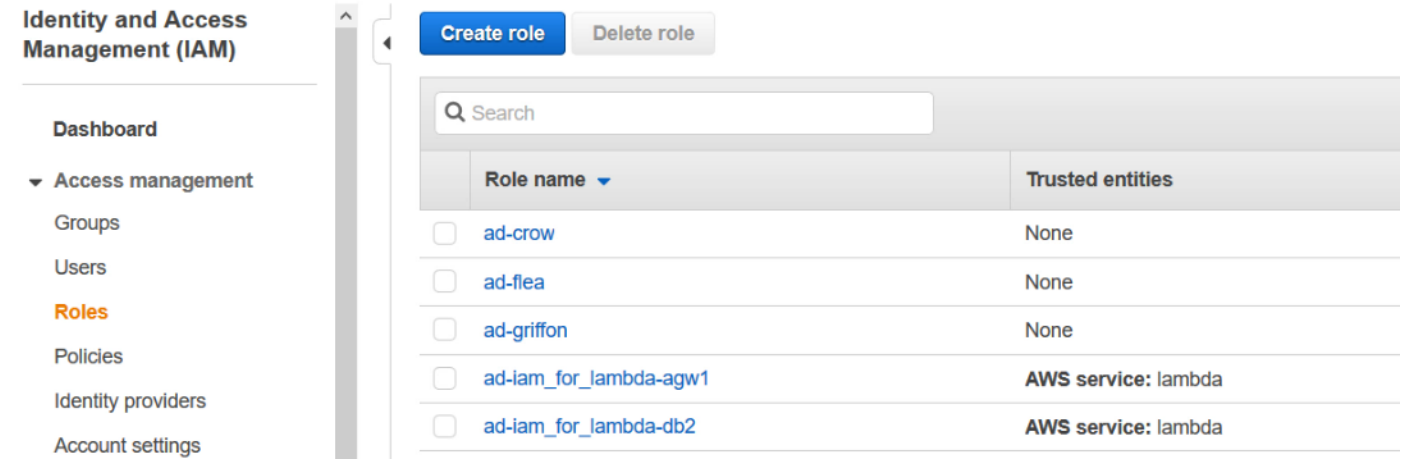


Step 21: Check policy document, policy usages and policy permissions.

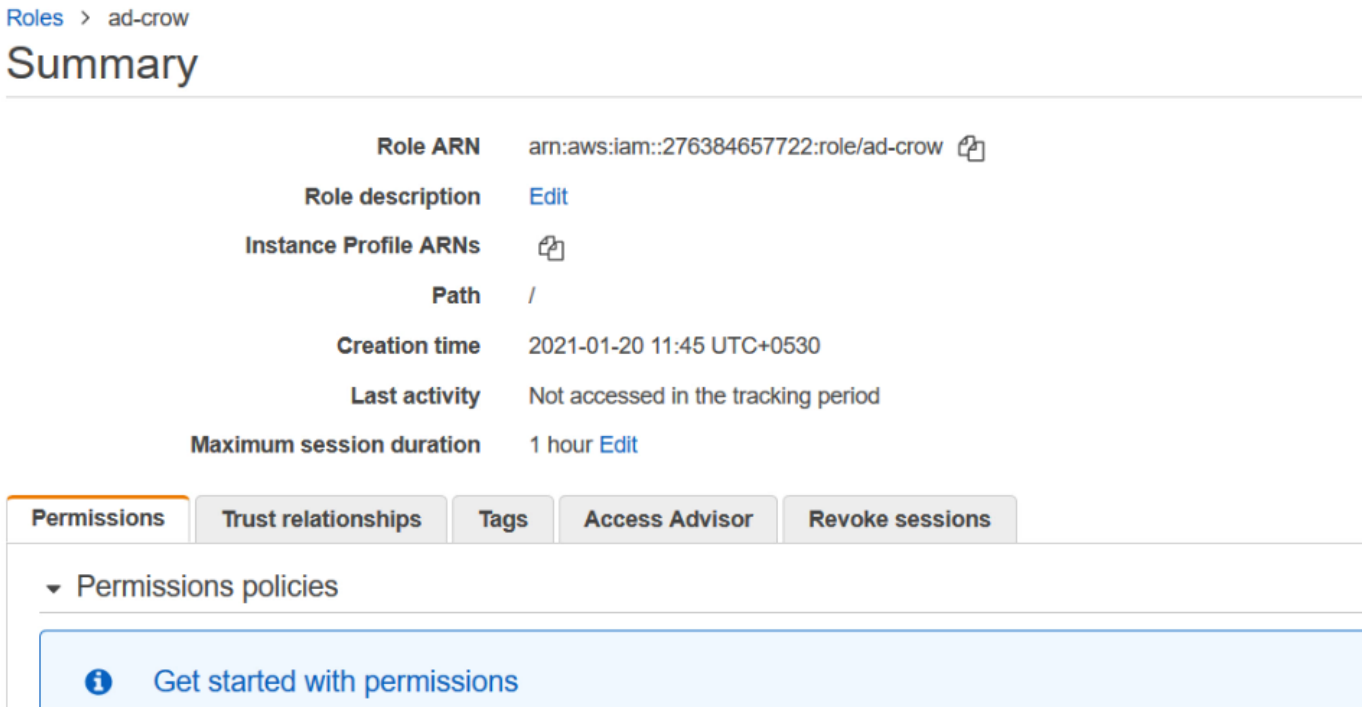




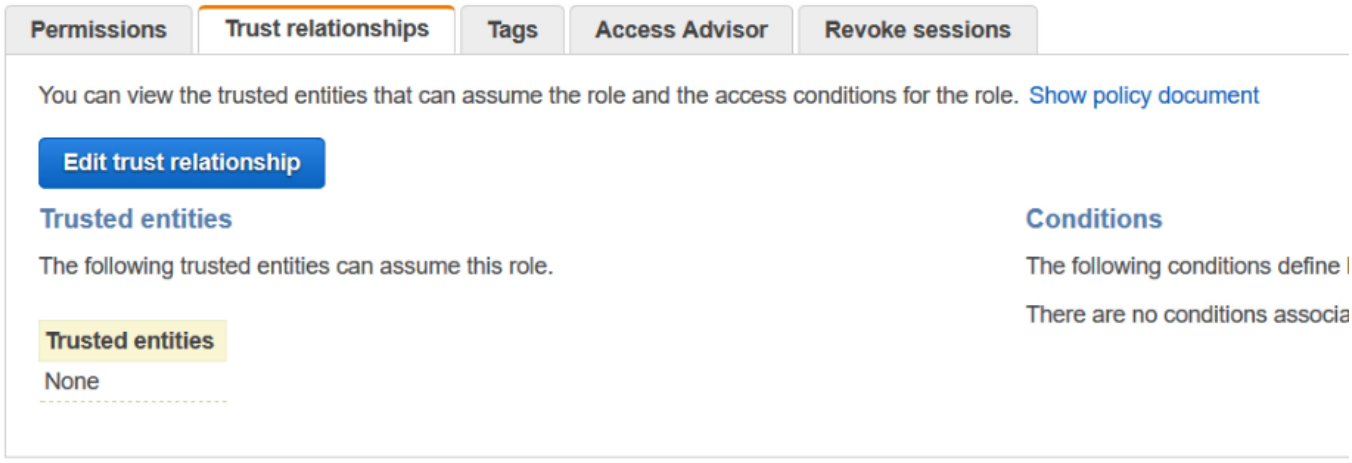
Step 22: Click on roles on the left panels to enumerate roles.



Step 23: Click on role name to enumerate roles for the AWS account.

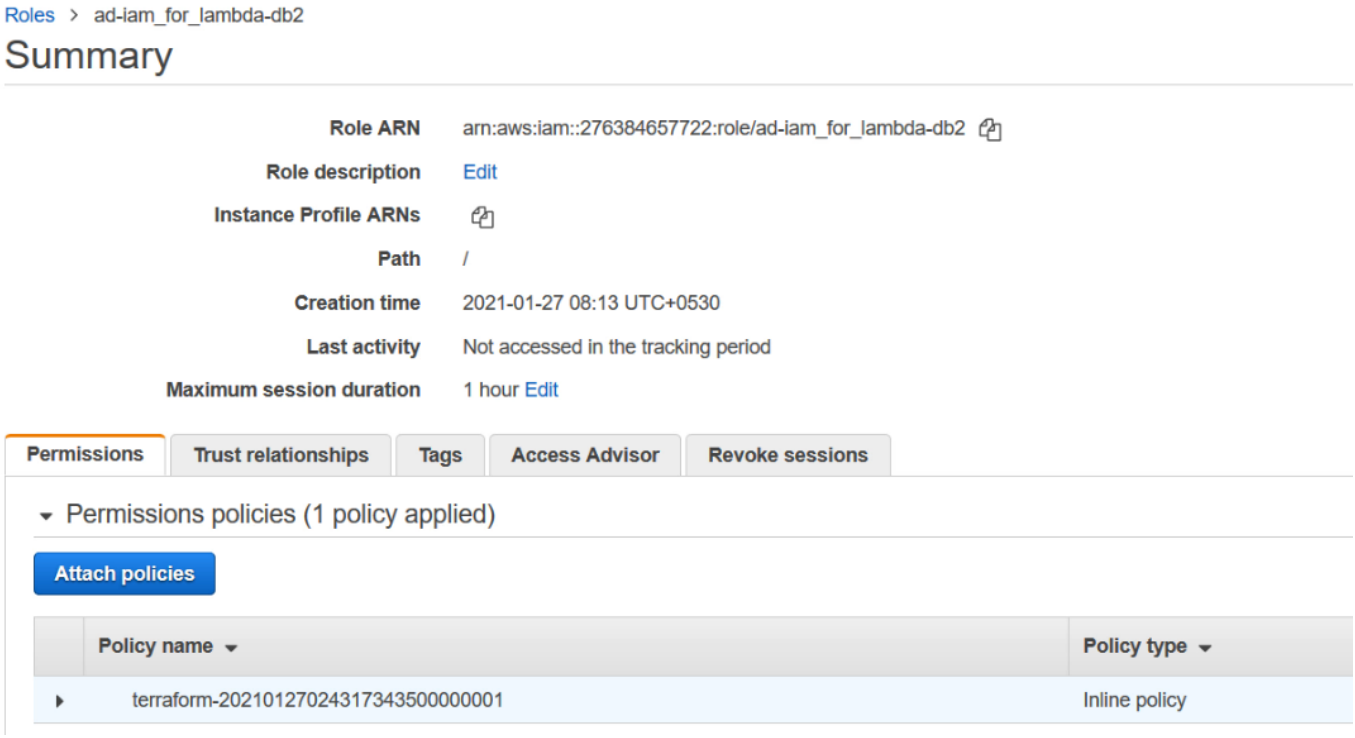


Step 24: Check for the role trust policy.

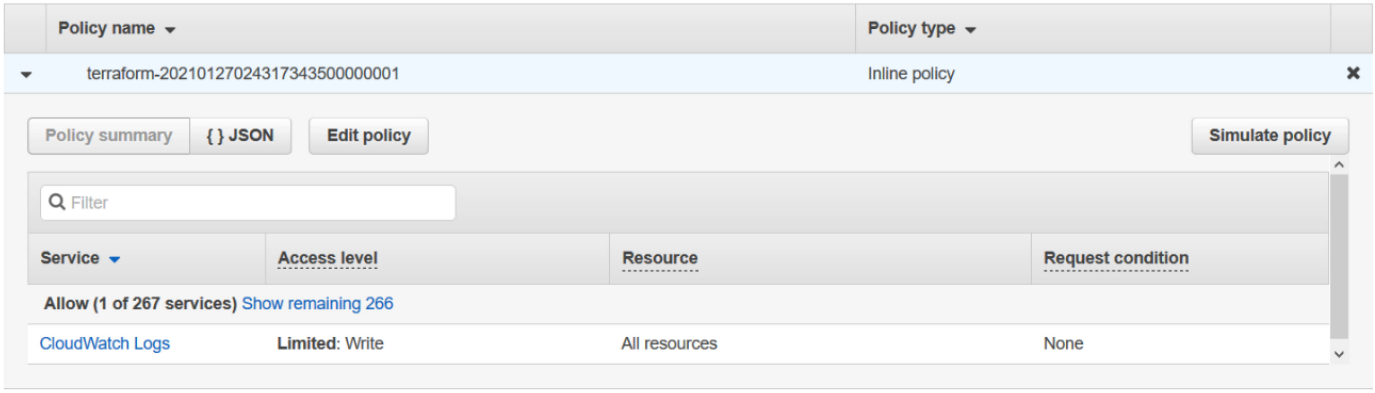


Step 25: Check role's trusted entities.

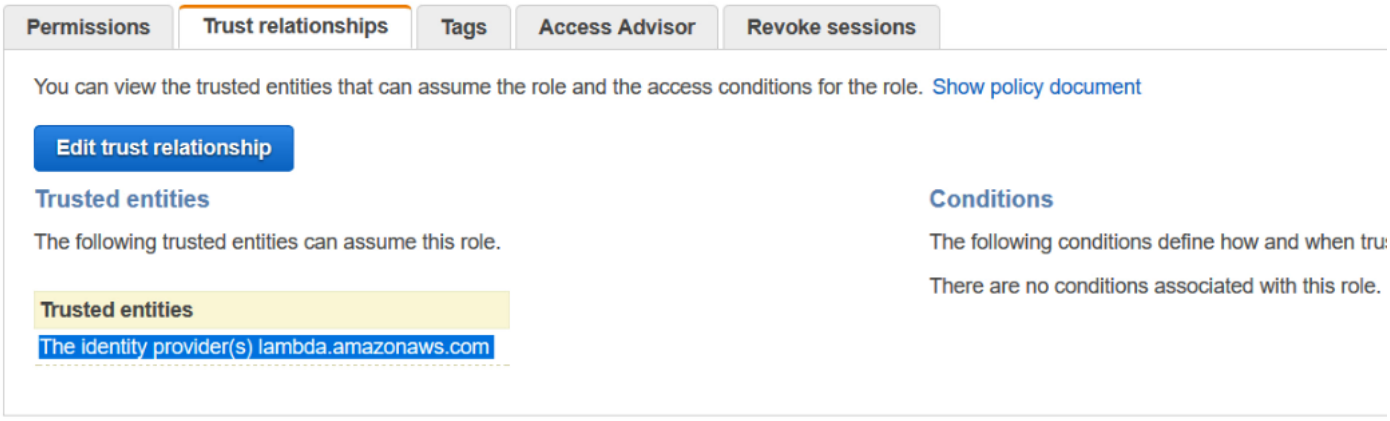
	Role name	Trusted entities	Last activity
☐	ad-crow	None	None
☐	ad-flea	None	None
☐	ad-griffon	None	None
☑	ad-iam_for_lambda-agw1	**AWS service:** lambda	None
☐	ad-iam_for_lambda-db2	**AWS service:** lambda	None
☐	ad-ibex	None	None
☐	ad-LoggingRole	**Account:** *	8 days



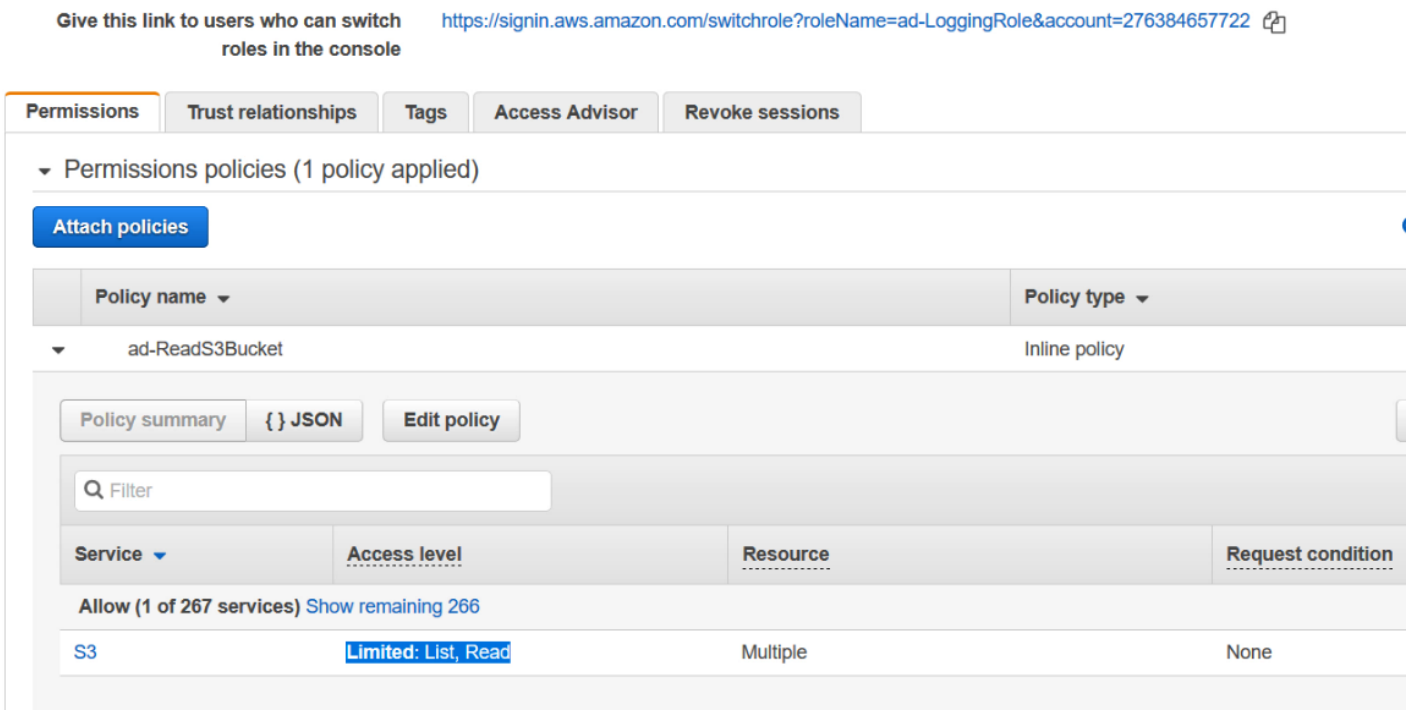
Step 26: Check for the role's inline policies.

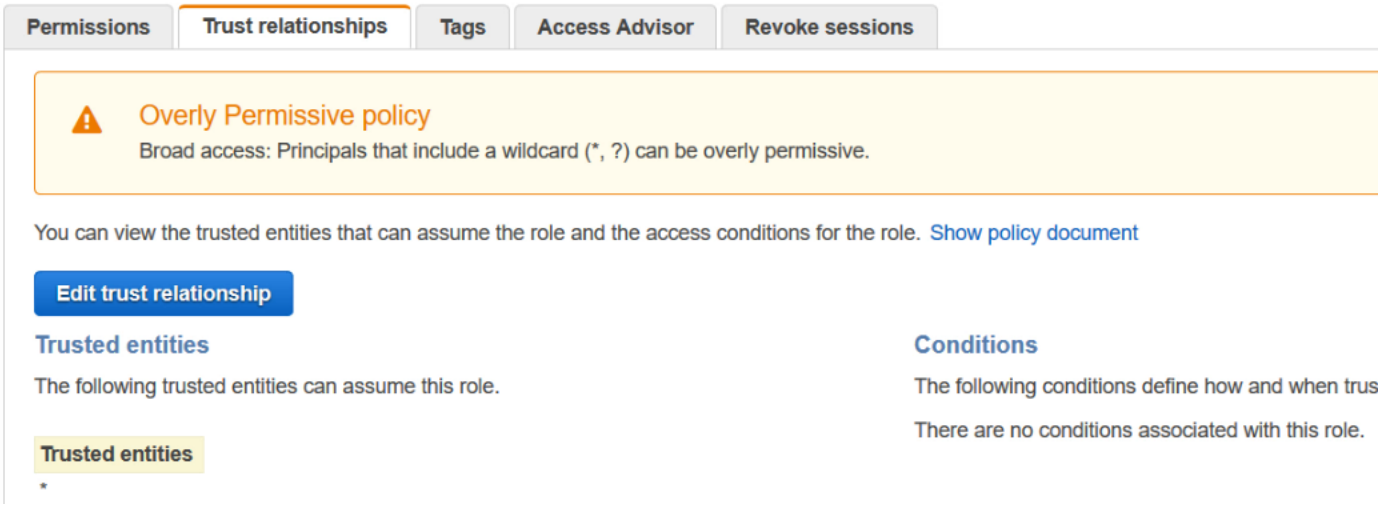


Step 27: Check for the role's trusted entities.



Step 28: Similarly enumerate other roles.





Note: Roles with * as trusted entities can be assumed by any AWS resource which can be inside or outside an organisation.

CLI Based Enumeration

Step 1: Click on the lab link button to get access to the AWS lab credentials.

Access Credentials to your AWS lab Account

Login URL	https://276384657722.signin.aws.amazon.com/console
Region	**Asia Pacific (Singapore) ap-southeast-1**
Username	**qiVMHVacHfwkveFcsASM**
Password	**Um7uYTlKeMjU35Jc**
Access Key ID	**AKIAUAWOPGE5KB43QAEB**
Secret Access Key	**w1OqQUFmc4uQaeJ4iHresM78bTFimXboQpSBMfuw**

Step 2: Configure AWS CLI to use the provided credentials.

Command: aws configure

```
 ⚡ root@Kali ~ aws configure
AWS Access Key ID [****************K2JI]: AKIAUAWOPGE5KB43QAEB
AWS Secret Access Key [****************+Ing]: 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
Default region name [us-east-1]: ap-southeast-1
Default output format [None]:
 ⚡ root@Kali ~
```

Step 3: Get a list of the users on the AWS account.

Command: aws iam list-users

```
        {
            "Path": "/",
            "UserName": "LabAdmin",
            "UserId": "AIDAUAWOPGE5NDH7SQWF4",
            "Arn": "arn:aws:iam::276384657722:user/LabAdmin",
            "CreateDate": "2020-09-10T00:40:21Z",
            "PasswordLastUsed": "2021-02-09T06:18:32Z"
        },
        {
            "Path": "/",
            "UserName": "qiVMHVacHfwkveFcsASM",
            "UserId": "AIDAUAWOPGE5N33FFBIDY",
            "Arn": "arn:aws:iam::276384657722:user/qiVMHVacHfwkveFcsASM",
            "CreateDate": "2021-02-13T02:58:49Z"
        }
    ]
}
 ⚡ root@Kali ~
```

Briefly check user name, user id and ARNs.

Step 4: Check groups for users.

Command: aws iam list-groups-for-user --user-name ad-adminson

```
root@Kali ~ aws iam list-groups-for-user --user-name ad-adminson
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "ad-Admin",
            "GroupId": "AGPAUAWOPGE5CRFDP2XCR",
            "Arn": "arn:aws:iam::276384657722:group/ad-Admin",
            "CreateDate": "2021-01-20T06:01:45Z"
        }
    ]
}
```

Step 5: Check policies attached to the user.

Command: aws iam list-attached-user-policies --user-name ad-user

```
root@Kali ~ aws iam list-attached-user-policies --user-name ad-user
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonS3ReadOnlyAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"
        }
    ]
}
```

Step 6: Check for any signing certificates for the user.

Command: aws iam list-signing-certificates --user-name ad-user

```
root@Kali ~ aws iam list-signing-certificates --user-name ad-user
{
    "Certificates": [
        {
            "UserName": "ad-User",
            "CertificateId": "UFICN7PBNLWIIMKL44Y2V2WHF55BAAWF",
            "CertificateBody": "-----BEGIN CERTIFICATE-----\nMIIDETCCAfkCFDWs8Ss6Cjinu2B/2zNYM/yAKePKMA0GCSq
5ldCBXaWRnaXRzIFB0eSBMdGQwHhcNMjEwMTIwMDYyOTI3WhcNMjIwMTIwMDYy\nOTI3WjBFMQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29t
nAQ8AMIIBCgKCAQEA4CkwJtCjmLNZrGReKKK+xSbEmXIJDrH6a5JyRAegXKh5Xke/\nagupZJwIgraKZ9tXeEeIiASVlyL/1bOTRRVHAcK2x
XO\n8Yg0Kz/3f8wRYighGbyMIPP4/OP+XiweBWpzioQ1Y1TBhLzSbWfsu3w6C97+V99D\nzTGNexMgbNJ5w6eBUWgjcMpblzvCK6c+QyOnmJ
BCwUA\nA4IBAQAAye2ttrt8T0PICawSooLIr+DD7/9ZP3yiCcyqlBR+tHl4u5OYcZaae0e0\nSjSRe2uUiMGxd8kaaoe1t8uCdRfHTP0RnEL
yJSnwTzO\n7SXZK2/inWE9HpwTb6ujozZ8A6hDh6VgvcjZp1lam3gJn1csqb3NvcSMZsMwt4dv\nyD0BJQKUALKQ/8/txppEwQCBOtsgR1×6
--\n",
            "Status": "Active",
            "UploadDate": "2021-01-20T06:29:43Z"
        }
    ]
}
```

Step 7: Check for any public ssh keys for the user.

Command: aws iam list-ssh-public-keys --user-name ad-user

```
root@Kali ~ aws iam list-ssh-public-keys --user-name ad-user
{
    "SSHPublicKeys": [
        {
            "UserName": "ad-User",
            "SSHPublicKeyId": "APKAUAWOPGE5M47NZEIT",
            "Status": "Active",
            "UploadDate": "2021-01-20T06:01:51Z"
        }
    ]
}
```

Step 8: Get ssh key details.

Command: aws iam get-ssh-public-key --user-name ad-user --encoding PEM --ssh-public-key-id APKAUAWOPGE5M47NZEIT

```
root@Kali ~ aws iam get-ssh-public-key --user-name ad-user --encoding PEM --ssh-public-key-id APKAUAWOPGE5M47NZEIT
{
    "SSHPublicKey": {
        "UserName": "ad-user",
        "SSHPublicKeyId": "APKAUAWOPGE5M47NZEIT",
        "Fingerprint": "ba:1f:d9:94:62:7a:78:70:fc:ef:6b:72:9a:29:06:5b",
        "SSHPublicKeyBody": "-----BEGIN PUBLIC KEY-----\nMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAx/WOS/NpqoCDYrX8J2GM\nX
mjsKN74rQlG+zpC0NPn7lyMmAsibQS7cjJx88REko7t9XagXGv2JEPzYwW\nEnrVBh5E2ojdatTNDJCk5nr9F8FXxHWKnqUIulrL1ezG+EyLuyDz8bGweTyJFN6V
0Usal8PkZRLIo6vPBtaydiC9MDK9tP6HqR+Ba+XiR62egx5bGGuq4p0a67BNO\nQP1t6dd+aOY9udJGWn/pOaZRZtIL+Ht65etkyZ4bR8fBUzp90Yd96guto4hzP
qjpdSXNdoMmtmL1NiL0RxooN0/NQWaReU2XhUD3vyx154DsxUEHslGjKSkCEyL8V\nq6wkqeEcjYpV5fNsuXqxgClUHNqV7WiJTJXbQiIyNQSY22vmPAuKfVATIZ
X\no6dV+63VxNWgmtbKoEg/uV8CAwEAAQ==\n-----END PUBLIC KEY-----\n",
        "Status": "Active",
        "UploadDate": "2021-01-20T06:01:51Z"
    }
}
```

Step 9: Check for MFA devices for users.

Command: aws iam list-virtual-mfa-devices

```
root@Kali ~ aws iam list-virtual-mfa-devices
{
    "VirtualMFADevices": [
        {
            "SerialNumber": "arn:aws:iam::276384657722:mfa/ad-User",
            "User": {
                "Path": "/",
                "UserName": "ad-User",
                "UserId": "AIDAUAWOPGE5E7QHAWL43",
                "Arn": "arn:aws:iam::276384657722:user/ad-User",
                "CreateDate": "2021-01-20T06:01:45Z"
            },
            "EnableDate": "2021-01-20T06:47:34Z"
        }
    ]
}
```

Step 10: Check for user login profile.

Command: aws iam get-login-profile --user-name ad-user

```
root@Kali ~ aws iam get-login-profile --user-name ad-user
{
    "LoginProfile": {
        "UserName": "ad-user",
        "CreateDate": "2021-01-20T06:01:51Z",
        "PasswordResetRequired": false
    }
}
root@Kali ~
```

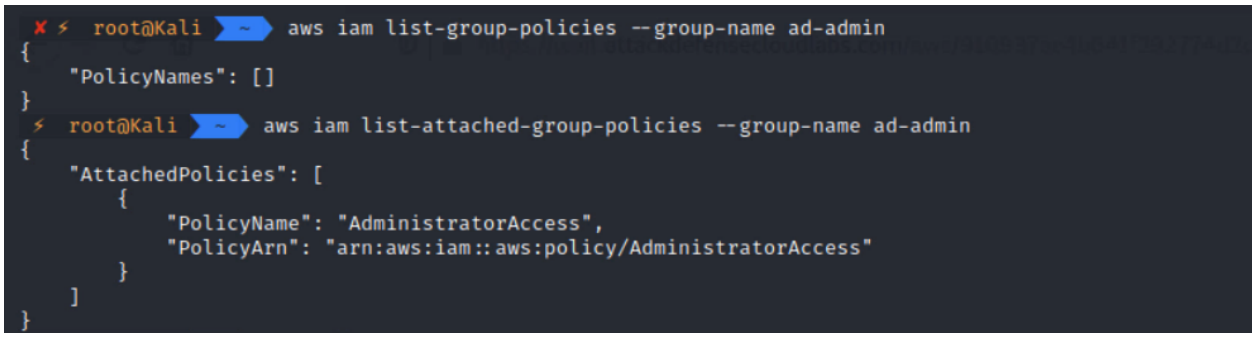
Step 11: Enumerate groups for the AWS account.

Command: aws iam list-groups

```
root@Kali ~ aws iam list-groups
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "ad-Admin",
            "GroupId": "AGPAUAWOPGE5CRFDP2XCR",
            "Arn": "arn:aws:iam::276384657722:group/ad-Admin",
            "CreateDate": "2021-01-20T06:01:45Z"
        },
```

Step 12: Check which policies are attached to the group to enumerate permissions.

Commands:
aws iam list-group-policies --group-name ad-admin
aws iam list-attached-group-policies --group-name ad-admin



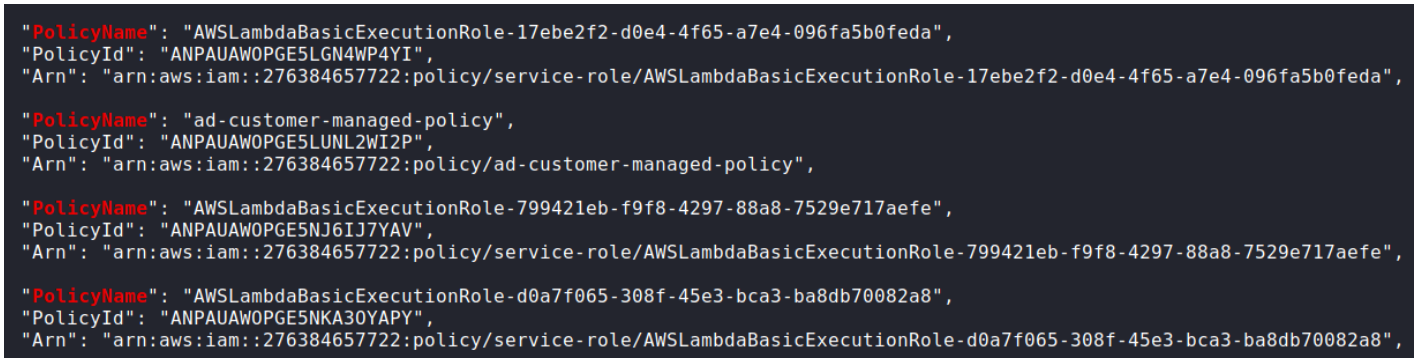
Step 13: Enumerate policies for the AWS account.

Command: aws iam list-policies

Without any filter the command will return all the IAM managed policies (AWS Managed Policy + Customer Managed Policies)

Step 14: Searching for customer managed policies

Command: aws iam list-policies --scope Local | grep -A2 PolicyName



The command “aws iam list-policies --scope Local” will return the customer managed policies. “ grep -A2 PolicyName” is used to search for the PolicyName string and the next two lines after the string match.

Step 15: Check for policy details of ad-customer-managed-policy.

Command: aws iam get-policy --policy-arn arn:aws:iam::276384657722:policy/ad-customer-managed-policy

```
root@Kali ~ aws iam get-policy --policy-arn arn:aws:iam::276384657722:policy/ad-customer-managed-policy
{
    "Policy": {
        "PolicyName": "ad-customer-managed-policy",
        "PolicyId": "ANPAUAWOPGE5LUNL2WI2P",
        "Arn": "arn:aws:iam::276384657722:policy/ad-customer-managed-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2021-01-20T07:51:42Z",
        "UpdateDate": "2021-01-20T07:51:42Z"
    }
}
```

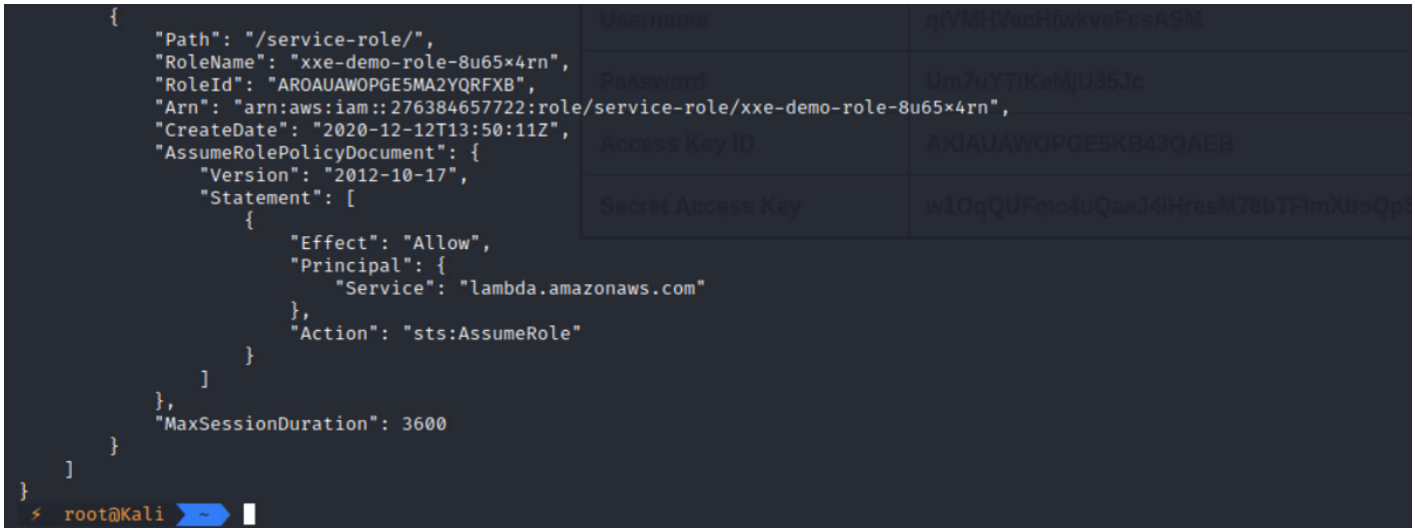
Step 16: Get the policy version document to check permissions that the policy grants.

Command: aws iam get-policy-version --policy-arn arn:aws:iam::276384657722:policy/ad-customer-managed-policy --version-id v1

```
root@Kali ~ aws iam get-policy-version --policy-arn arn:aws:iam::276384657722:policy/ad-customer-managed-policy --version-id v1
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "VisualEditor0",
                    "Effect": "Allow",
                    "Action": [
                        "s3:GetObjectAcl",
                        "s3:GetObject",
                        "s3:GetBucketPolicy",
                        "s3:GetObjectVersion"
                    ],
                    "Resource": [
                        "arn:aws:s3:::file-uploader-saved-files",
                        "arn:aws:s3:::file-uploader-saved-files/*"
                    ]
                }
            ]
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2021-01-20T07:51:42Z"
    }
}
```

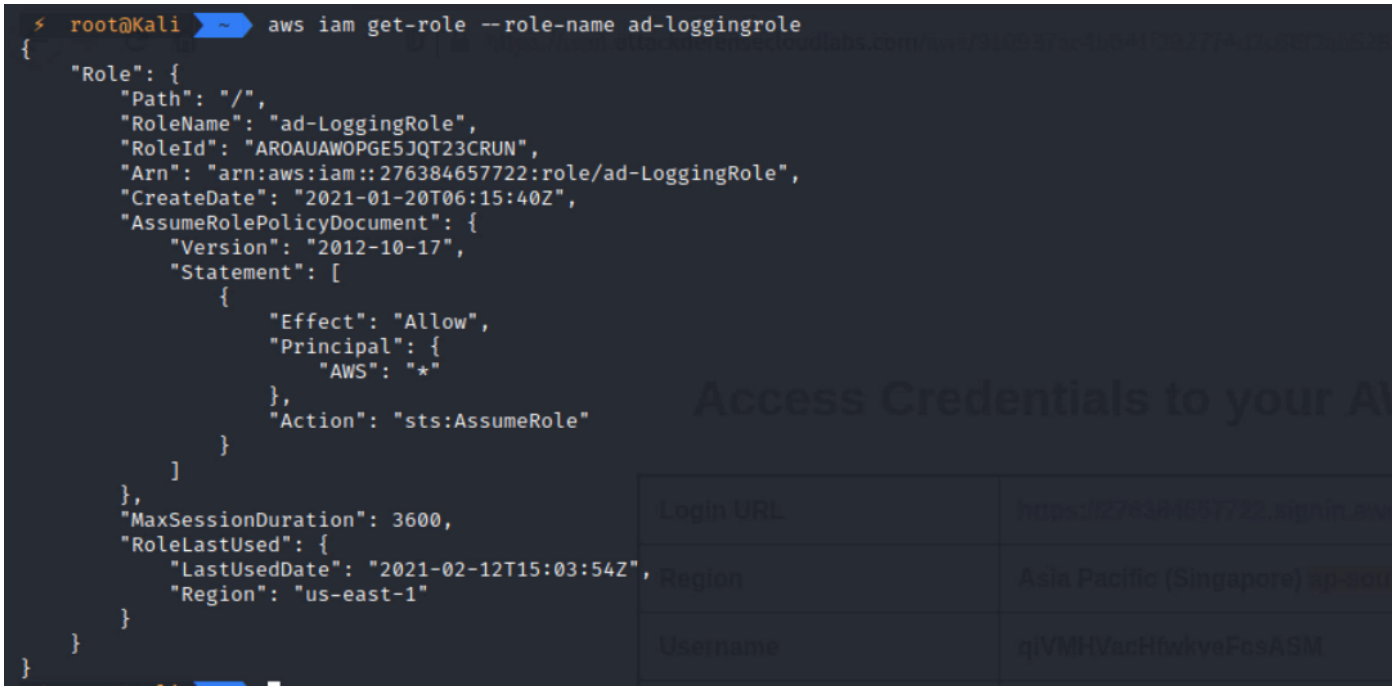
Step 17: Enumerate roles on the AWS account.

Command: aws iam list-roles



Step 18: Check details for roles.

Command: aws iam get-role --role-name ad-loggingrole



Note: This role can be assumed by any AWS resource.

Step 19: Check for policies attached to roles.

Commands:
aws iam list-attached-role-policies --role-name ad-loggingrole
aws iam list-role-policies --role-name ad-loggingrole

```
 ✘ ⚡  root@Kali  ~  aws iam list-role-policies --role-name ad-loggingrole
{
    "PolicyNames": []
}
 ⚡  root@Kali  ~  aws iam list-attached-role-policies --role-name ad-loggingrole
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonS3ReadOnlyAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"
        },
        {
            "PolicyName": "IAMReadOnlyAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/IAMReadOnlyAccess"
        }
    ]
}
```

References:

1. AWS CLI (https://docs.aws.amazon.com/cli/latest/reference/)