

AWS Cloud Infrastructure Overview

In this cloud environment, the **AWS Management Console** serves as the central control panel for managing all cloud-based resources.

At the highest level, a **Root User** is responsible for administrative oversight managing all assets, permissions and configurations across the organization's AWS account.

To simulate an enterprise structure, multiple **departments** such as Production and Deployment are represented through isolated environments, each managing its own set of cloud assets. These assets include **virtual servers** and **workstations** provisioned as **Amazon EC2 instances**.

EC2 Instances – Virtual Infrastructure

Each **EC2 instance** functions like a dedicated virtual computer or server hosted in the AWS cloud. These instances vary in CPU cores, memory (RAM), and performance capacity, defined by the **Instance Type** (for example `t2.micro`, `t3.medium`).

Within this setup, I will be deploying two key environments:

- **Deployment Server** – Used for staging, testing, and preparing updates before release.
- **Production Server** – Hosts the live environment used by end users or departments.

Each instance is built from an **Amazon Machine Image (AMI)**, which acts as the base operating system and configuration blueprint for the virtual machine.

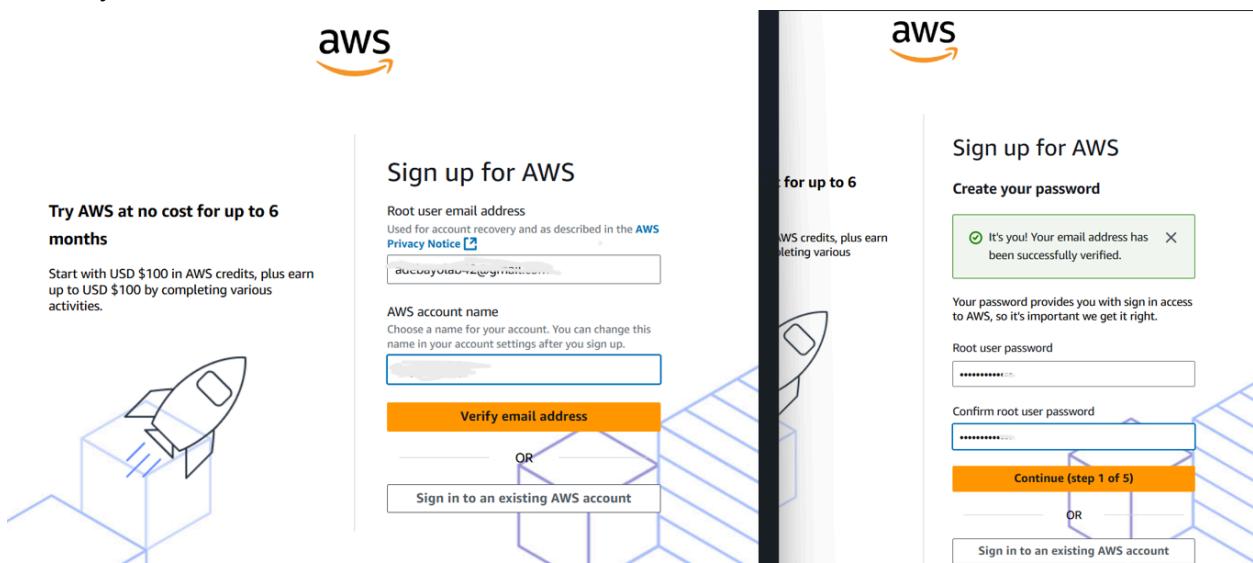
Cloud Security Implementation Steps (IAM)

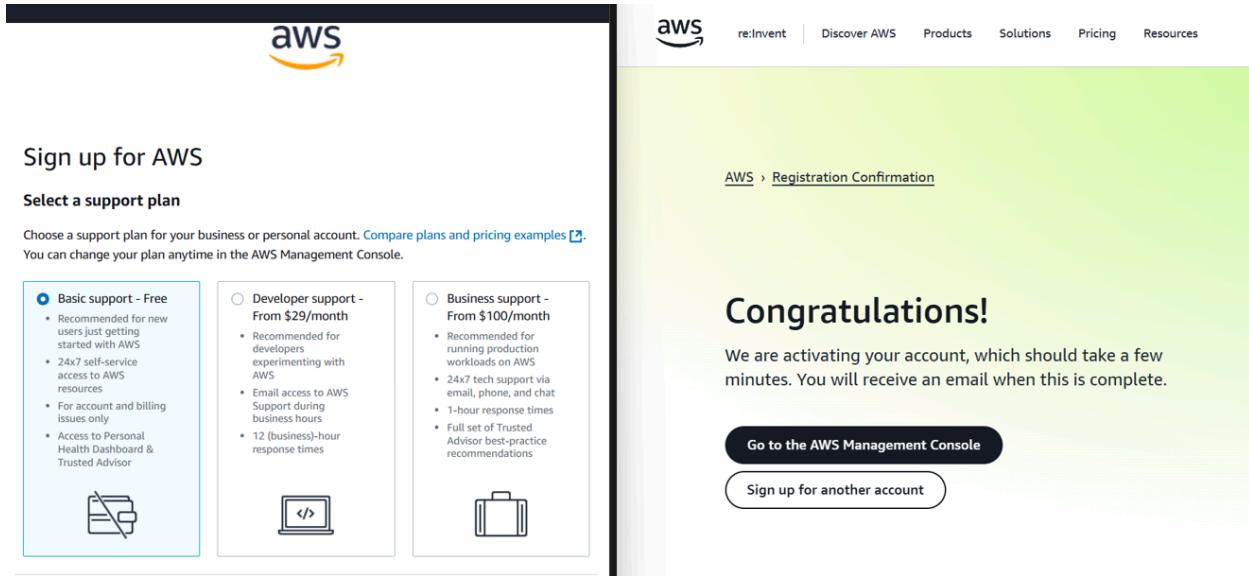
- **Set up the AWS Management Console** → Access and configure the AWS Management Console to manage cloud resources and permissions.
- **Launch Two EC2 Instances** → Deploy two virtual machines to simulate separate environments (e.g., *Production* and *Deployment* servers).
- **Create an IAM Policy** → Define a custom JSON-based policy to control permissions and enforce least-privilege access for users and groups.
- **Create an AWS Account Alias** → Customize the AWS account alias for easier identification and secure access management.
- **Create an IAM Group and User** → Establish user groups based on roles or departments and create IAM users with appropriate permissions.

- **Test IAM User Access**→ Log in as the newly created IAM user to verify that permissions and restrictions are working as intended.
- **CloudTrail & Config**→ This ensures every user action, configuration change and policy update in the AWS environment is tracked.

STEP 1 → CREATING AWS ACCOUNT

- Open your preferred web browser and navigate to <https://aws.amazon.com>
- Click on the link titled “**Amazon Web Services (AWS)**” to access the official AWS homepage.
- On the homepage, click “**Create an AWS Account.**”
- You’ll be redirected to the AWS sign-up console where you can begin the account creation process.
- Register using a **root user email address** (this will serve as the main administrative account).
- When prompted, select “**Personal use**” for the account type.
- Choose the **Basic Support Plan**, then click “**Complete Sign Up.**”
- Once registration is complete, click “**Go to the AWS Management Console**” to access your AWS environment.

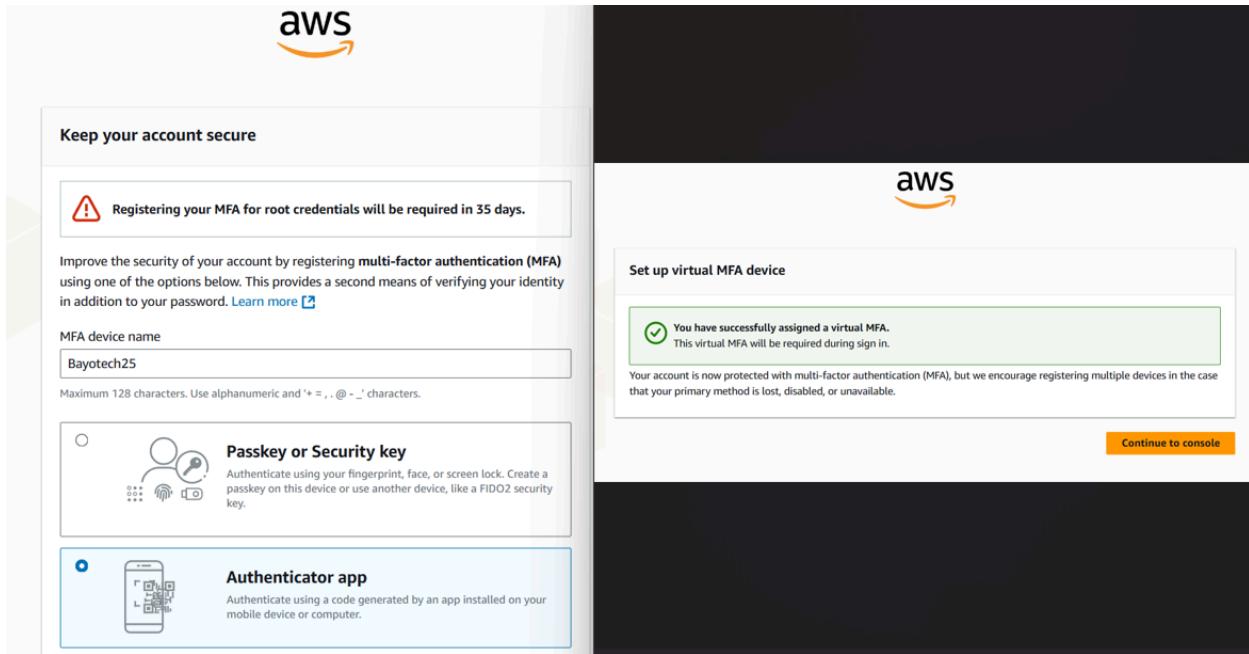




Once logged into the **AWS Management Console**, you'll see the Services Menu, which provides access to all available AWS services such as **EC2, S3, IAM**, and more.

From the top-right corner of the console, select the **region closest to your geographical location**. Choosing a nearby region helps ensure lower latency, faster performance, and compliance with data residency requirements.

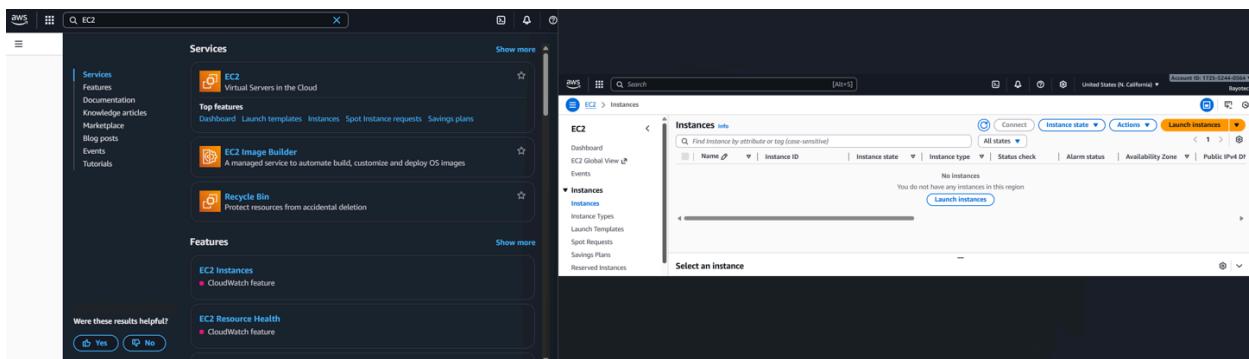
As best practice ,I will be enabling MFA for the Root user, as this is one of AWS top security recommendations.



STEP 2 → Creating EC2 Instances (Deployment Server)

To create a Deployment EC2 instance, follow these steps:

- In the AWS Management Console, use the search bar at the top and type “EC2.”
 - Click on EC2 this is where you can create and manage virtual servers (instances) in the AWS cloud.
 - In the EC2 dashboard, select “**Instances**” from the left-hand menu.
- Note:** By default, no instances will appear if you haven’t created any in the current region.
- Click “**Launch Instance**” to begin creating a new virtual server.
 - Under the Name and Tags section:
 - In the Name field, enter:
 1. **Bayotech-Deployment-John**
 2. Click “**Add additional tag.**”
 - **Key: Env** (short for Environment) → **Value: Deployment** → Leave **Resource Type** as default (blank).



It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

[Take a walkthrough](#) [Do not show me this message again.](#)

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

Bayotech-Deployment-John

Add additional tags

Manage tags Info

A tag is a custom label that you assign to an AWS resource. You can use tags to help organize and identify your instances.

Key	Value - optional	Action
Env	Deployment	Remove
Name	Bayotech-Deployment-John	Remove

Add new tag

You can add up to 48 more tags.

Cancel [Save](#)

Configuring the EC2 Instance

1) Select the Application and OS Image (AMI)

- Scroll down to the “Application and OS Images (Amazon Machine Image)” section.
- From the dropdown menu, select the **Operating System** you want to use.
 - For this lab, I chose **Amazon Linux** (free tier eligible).

NOTE→ You can also choose other OS options such as **Ubuntu**, **Windows Server**, or **Red Hat Enterprise Linux**, depending on your project requirements.

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Quick Start

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-0e6be795b21969e1d (64-bit (x86), uefi-preferred) / ami-0e3605f8a6c0853e5 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.9.20251027.0 x86_64 HVM kernel-6.1

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.9.2...[read more](#)
ami-0e6be795b21969e1d

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

[Cancel](#)

2) Choose Instance Type

- Select the **Instance Type**, which determines the virtual hardware (CPU and memory) of your EC2 instance. (For example: **t2.micro** (1 vCPU, 1 GiB RAM) — suitable for testing and free-tier environments). → Once selected, click **Next** to proceed.

Instance type [Info](#) | Get advice

Instance type

t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0138 USD per Hour On-Demand Linux base pricing: 0.0138 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0156 USD per Hour
On-Demand RHEL base pricing: 0.0282 USD per Hour On-Demand Windows base pricing: 0.0184 USD per Hour

All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

3. Key Pair (Login)

- Under Key pair (login), select “Proceed without a key pair.”

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Bayotech-dep-john ▼  [Create new key pair](#)

4) Under Network Settings, For this project I will check on ‘**Create security group**’ (select existing security group can be checked if there is an existing one) → Check “**Allow SSH traffic from**” and select Anywhere (I will be using ‘myip’) to enable remote access.

NOTE → (For enhanced security, restrict this to a specific IP range in real-world deployments.)

▼ Network settings [Info](#) [Edit](#)

Network | [Info](#)
vpc-0d52155d5fec30136

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable

Firewall (security groups) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#) [Select existing security group](#)

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from
Helps you connect to your instance My IP
76.89.89.150/32 ▼

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

5. Configure Storage

- The default storage size is 8 GiB (General Purpose SSD).
- You may adjust this based on your use case, but 8 GiB is sufficient for basic testing.

Configure storage [Info](#)

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

6. Launch the Instance

- Review your configuration and click “Launch Instance.”
- Once launched, navigate back to EC2 → Instances to view your newly created server.
- If it doesn’t appear immediately, click Refresh to update the instance list.

Instances (1) [Info](#)

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 IP
Bayotech-Depl...	i-0501b9d91a2279c26	Running	t3.micro	Initializing	View alarms +	us-west-1a	ec2-54-183-1

***REPEAT THE SAME PROCESS TO CREATE PRODUCTION INSTANCE

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name Add additional tags

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.9.2... [read more](#)

ami-0e6be795b21969e1d

aws | Search [Alt+S] Account ID: 1725-5244-0564 ▾ United States (N. California) ▾ Bayotech

EC2 > Instances > i-0fc8f175ad8661cdb > Manage tags

Manage tags Info

A tag is a custom label that you assign to an AWS resource. You can use tags to help organize and identify your instances.

Key	Value - optional
<input type="text" value="Env"/>	<input type="text" value="Production"/> X Remove
<input type="text" value="Name"/>	<input type="text" value="Bayotech-Production-Brian"/> X Remove

[Add new tag](#) You can add up to 48 more tags.

Cancel Save

aws | Search [Alt+S] Account ID: 1725-5244-0564 ▾ United States (N. California) ▾ Bayotech

EC2 > Instances > Launch an instance

▼ Instance type Info | Get advice

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory Current generation: true
 On-Demand SUSE base pricing: 0.0124 USD per Hour On-Demand Linux base pricing: 0.0124 USD per Hour
 On-Demand Ubuntu Pro base pricing: 0.0159 USD per Hour
 On-Demand Windows base pricing: 0.0216 USD per Hour On-Demand RHEL base pricing: 0.0412 USD per Hour

[All generations](#) [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

ProductionKeybayotech ▼ [Create new key pair](#)

aws | Search [Alt+S] Account ID: 1725-5244-0564 ▾ United States (N. California) ▾ Bayotech

EC2 > Instances > Launch an instance

▼ Network settings Info

Network [Info](#)
 vpc-0d52155d5fec30136

Subnet [Info](#)
 No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
 Enable

Firewall (security groups) [Info](#)
 A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#) [Select existing security group](#)

We'll create a new security group called 'launch-wizard-2' with the following rules:

- Allow SSH traffic from**
Helps you connect to your instance
- Allow HTTPS traffic from the internet**
To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet**
To set up an endpoint, for example when creating a web server

Edit

▼ Summary

Number of instances [Info](#)
 1

Software Image (AMI)
 Amazon Linux 2023 AMI 2023.9.2... [read more](#)
 ami-0e6be795b21969e1d

Virtual server type (instance type)
 t3.micro

Firewall (security group)
 New security group

Storage (volumes)
 1 volume(s) - 8 GiB

Cancel Launch instance Preview code

Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Bayotech-Depl...	i-0501b9d91a2279c26	Running	t3.micro	3/3 checks passed	View alarms +	us-west-1a	ec2-54-183-1:
Bayotech-Prod...	i-0fc8f175ad8661cdb	Running	t3.micro	Initializing	View alarms +	us-west-1a	ec2-54-193-1:

i-0501b9d91a2279c26 (Bayotech-Deployment-John)

Details	Status and alarms	Monitoring	Security	Networking	Storage	Tags
VPC ID	Subnet ID			Availability zone		
vpc-0d52155d5fec30136	subnet-03116c8c8745b7c8d			us-west-1a		
Availability zone ID	Outpost ID					
usw1-az1	-					
IP addresses <small>Info</small>						
Public IPv4 address	Private IPv4 addresses			IPv6 addresses		
54.183.156.147 open address	172.31.29.20			-		
Secondary private IPv4 addresses	Carrier IP addresses (ephemeral)					
-	-					

i-0fc8f175ad8661cdb (Bayotech-Production-Brian)

Details	Status and alarms	Monitoring	Security	Networking	Storage	Tags
Instance summary <small>Info</small>						
Instance ID	Public IPv4 address			Private IPv4 addresses		
i-0fc8f175ad8661cdb	54.193.178.203 open address			172.31.26.99		
IPv6 address	Instance state			Public DNS		
-	Running			ec2-54-193-178-203.us-west-1.compute.amazonaws.com open address		
Hostname type	Private IP DNS name (IPv4 only)					
IP name: ip-172-31-26-99.us-west-1.compute.internal	ip-172-31-26-99.us-west-1.compute.internal					

STEP 3→ Create an IAM Policy

IAM policies define the rules that determine who has access to what within the AWS environment. These policies specify which actions users, groups, or roles can perform on specific

AWS resources. There are available policies but I will be creating a custom IAM policy, follow these steps:

- Navigate back to the AWS Management Console.
- In the search bar, type “IAM” and select IAM from the results.

- In the IAM dashboard, click on “Policies” in the left-hand menu. You’ll see a list of existing AWS-managed and customer-managed policies.
- To create a new custom policy, click “Create policy.”

Policy name	Type	Used as	Description
AccessAnalyzerServiceRole...	AWS managed	None	Allow Access Analyzer to analyze resou...
AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissi...
AdministratorAccess-AWSE...	AWS managed	None	Grants account administrative permissi...
AIOpsAssistantIncidentRep...	AWS managed	None	Provides permissions required by the A...
AIOpsAssistantPolicy	AWS managed	None	Provides ReadOnly permissions requir...
AIOpsConsoleAdminPolicy	AWS managed	None	Grants full access to Amazon AI Opera...
AIOpsOperatorAccess	AWS managed	None	Grants access to the Amazon AI Opera...
AIOpsReadOnlyAccess	AWS managed	None	Grants ReadOnly permissions to the A...
AlexaForBusinessDeviceSet...	AWS managed	None	Provide device setup access to AlexaFo...

- The Policy Editor will open this interface and allow me to define custom permissions.

The screenshot shows the 'Specify permissions' step of the AWS IAM 'Create policy' wizard. It includes a navigation bar with the AWS logo, search bar, and account information. The main area has a title 'Specify permissions' with an info link, a description about adding permissions, and a 'Policy editor' section. The 'Service' dropdown is set to 'Choose a service'. Below it is a 'Visual' tab, a 'JSON' tab, and a 'Actions' dropdown. A 'Next' button is at the bottom right.

- Switch to the JSON tab to enter my policy code. Input the custom JSON policy script (for example, the one granting EC2 access to specific tagged resources).
- Review the policy structure it consists of:
 - i. Root permissions: Define which AWS services and actions are allowed or denied.
 - ii. Group permissions: Apply policies to multiple users within a group.
 - iii. Identity permissions: Attach directly to a specific IAM user or role.

The screenshot shows the 'Specify permissions' step of the AWS IAM 'Create policy' wizard, focusing on the JSON editor. The visual representation of the policy is as follows:

```
1▼ {
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Sid": "AllowFullAccessToDeploymentTaggedResources",
6      "Effect": "Allow",
7      "Action": "ec2:*",
8      "Resource": "*",
9      "Condition": {
10        "StringEquals": {
11          "ec2:ResourceTag/Env": "Deployment"
12        }
13      }
14    },
15  ]
```

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
BayotechDeploymentPolicy

Maximum 128 characters. Use alphanumeric and '+-=._@-' characters.

Description - optional
Add a short explanation for this policy.
IAM policy for users in the Deployment environment.

Maximum 1,000 characters. Use alphanumeric and '+-=._@-' characters.

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Explicit deny (1 of 450 services)

Service	Access level	Resource	Request condition
EC2	Full - Trusted	All resources	None

Once your policy is complete → click “Next Tags” → “Next: Review”, give the policy a clear name (e.g., EC2DeploymentAccessPolicy) → then click “Create policy”.

Identity and Access Management (IAM)

Policies (1400) Info

A policy is an object in AWS that defines permissions.

Filter by Type

View policy **X**

Actions **Delete** **Create policy**

Policy name	Type	Used as	Description
BayotechDeploymentPolicy	Customer managed	None	IAM policy for users in the Deployment...

REPEAT THE-SAME PROCESS TO CREATE A PRODUCTION POLICY.

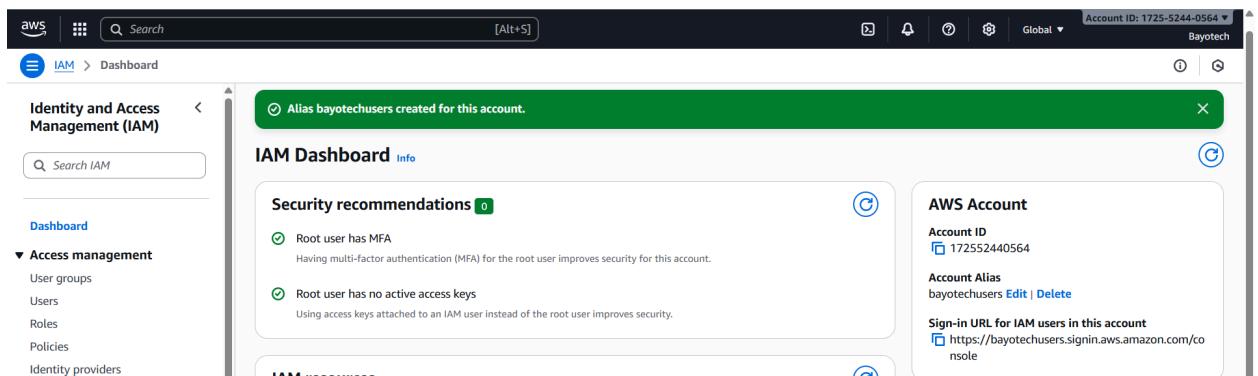
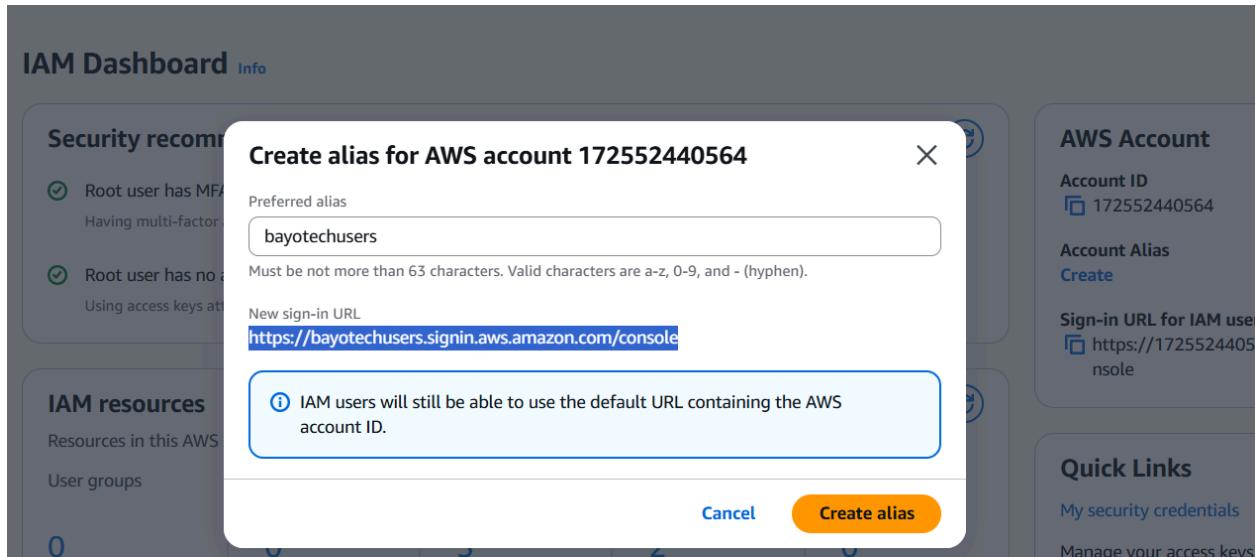
The screenshot shows the AWS IAM Policies page. At the top, a green banner displays the message "Policy BayotechProductionPolicy created." Below this, the title "Policies (1401) Info" is shown, with a note that "A policy is an object in AWS that defines permissions." A search bar contains the text "bay". A filter bar at the top right says "Filter by Type All types 2 matches". The main table lists two policies:

Policy name	Type	Used as	Description
BayotechDeploymentPolicy	Customer managed	None	IAM policy for users in the Deployment...
BayotechProductionPolicy	Customer managed	None	IAM policy for users in the production ...

STEP 4→ Create AWS ALIAS

On IAM Dashboard → Click on ‘create’ under ‘Account Alias’ . Next, to customize your AWS sign-in URL. Enter your ‘preferred alias’, for example: Bayotechusers → Click ‘create Alias’

The screenshot shows the AWS IAM Dashboard. On the left, a sidebar lists navigation items: Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and Access reports (Access Analyzer, Resource analysis). The main area has three sections: "Security recommendations" (Root user has MFA, Root user has no active access keys), "IAM resources" (User groups: 0, Users: 0, Roles: 3, Policies: 2, Identity providers: 0), and "AWS Account" (Account ID: 172552440564, Account alias: Create, Sign-in URL: https://172552440564.siginn.aws.amazon.com/console). A "Quick Links" section includes "My security credentials" and a note about managing access keys, MFA, and other credentials.



This alias creates a more user-friendly login URL(e.g.,<https://Bayotechusers.signin.aws.amazon.com/console>) making it easier and more secure for team members to access the AWS environment.

STEP 5→ CREATING IAM GROUP AND USERS

CREATING IAM GROUP

An IAM Group is a collection of multiple IAM users. The main advantage of using groups is simplified permission management ,instead of assigning policies to each user individually, you can apply a policy once to the group, and all members automatically inherit those permissions.

For example, in an organization with 20 employees, managing access for each user separately would be inefficient. By creating a single group (e.g., DeploymentTeam), you can attach one or more policies to that group, and every member of the group will receive the same access privileges.

Each IAM User is a member of one or more groups, allowing for structured, scalable, and secure access management within the AWS environment.

Steps To create Groups;

- Navigate to the IAM service by typing “IAM” into the search bar at the top of the AWS Management Console.
- In the IAM Dashboard, select “**User groups**” from the left-hand menu → Click “Create group.”
- In the Group name field, enter:**Bayotech-Deployment-Group**
- Under **Attach permissions policies (optional)**, search for the custom policy you previously created for example:**BayotechDeploymentPolicy**
- Select the checkbox next to the policy name to attach it to the group → Finally, click “**Create user group.**”

The screenshot shows the AWS IAM User groups page. On the left, there's a navigation sidebar with 'Identity and Access Management (IAM)' selected under 'Access management'. The main area is titled 'User groups (0) Info' and contains a search bar and a table header with columns for 'Group name', 'Users', 'Permissions', and 'Creation time'. A large orange arrow points to the 'Create group' button in the top right corner of the page.

The screenshot shows the 'Create user group' wizard. The title is 'Create user group'. The first step, 'Name the group', has a 'User group name' input field containing 'bayotech-Deployment-group'. An orange arrow points to this input field. Below the input field, there's a note: 'Enter a meaningful name to identify this group.' and 'Maximum 128 characters. Use alphanumeric and '+=_,@-' characters.'

Add users to the group - *Optional* (0) [Info](#)
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

User name	Group:	Last activity	Creation time
No resources to display			

Attach permissions policies - *Optional* (1/1079) [Info](#)
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Policy name	Type	Used as	Description
<input checked="" type="checkbox"/> BayotechDeploymentPolicy	Customer managed	None	IAM policy for users in the Deployment...
<input type="checkbox"/> BayotechProductionPolicy	Customer managed	None	IAM policy for users in the production ...

[Cancel](#) [Create user group](#)

bayotech-Deployment-group user group created. [View group](#) [X](#)

User groups (1) [Info](#)

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time
bayotech-Deployment-group	⚠ 0	Defined	Now

[Delete](#) [Create group](#)

REPEAT THE SAME STEPS TO CREATE A PRODUCTION GROUP .

bayotech-Production-group user group created. [View group](#) [X](#)

User groups (2) [Info](#)

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time
bayotech-Deployment-group	⚠ 0	Defined	1 minute ago
bayotech-Production-group	⚠ 0	Defined	Now

[Delete](#) [Create group](#)

Creating an IAM User and Assigning Group Access

- In the **IAM Dashboard**, click on “**Users**.” → Select “**create user**”
- Under Specify user details, enter a username, for example:**Bayotech-Deployment-John**
- Check the box for “Provide user access to the AWS Management Console”.

- (This allows the user to access the AWS Console via a web interface instead of only through the CLI or terminal.)
- Create a custom password, and (optional) uncheck the box labeled “User must create a new password at next sign-in.” → Click “Next.”

Specify user details

User details

User name: Bayotech-Deployment-John

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ _ - (hyphen)

Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Console password

Autogenerated password
You can view the password after you create the user.

Custom password
Enter a custom password for the user.

Must be at least 8 characters long
Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | `

Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

- On the ‘Set permission’ screen check ‘**Add user to group**’, select the group you created earlier (e.g., **Bayotech-Deployment-Group**). → Click “Next.” → On ‘Review and create’ leave as default → Click “Create user.”

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1/2)

Group name	Users	Attached policies	Created
<input checked="" type="checkbox"/> bayotech-Deployment-group	0	BayotechDeploymentPolicy	2025-11-02 (16 minutes ago)
<input type="checkbox"/> bayotech-Production-group	0	BayotechProductionPolicy	2025-11-02 (14 minutes ago)

Set permissions boundary - optional

Cancel Previous Next

Step 1

- Specify user details
- Step 2
- Set permissions
- Step 3
- Review and create**
- Step 4
- Retrieve password

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details		
User name	Console password type	Require password reset
Bayotech-Deployment-John	Custom password	No

Permissions summary

Name	Type	Used as
bayotech-Deployment-group	Group	Permissions group

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user

- On the Retrieve password page, download the generated .csv file this file contains the user's login credentials (username, password, and console sign-in URL).
- Copy the AWS Console Sign-In URL, which will reflect your custom alias (e.g., <https://Bayotechusers.signin.aws.amazon.com/console>).

Step 1

- Specify user details
- Step 2
- Set permissions
- Step 3
- Review and create
- Retrieve password**

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[View user](#)

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL <https://bayotechusers.signin.aws.amazon.com/console>

User name [Bayotech-Deployment-John](#)

Console password [***** Show](#)

Email sign-in instructions

Cancel Download .csv file Return to users list

Repeat the same process to create a user assigned to the Distribution group . Return to the Users tab ,you'll now see the newly created users along with the assigned group and attached policies.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management

Access reports

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[View user](#)

Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Acc
Bayotech-Deployment-John	/	1	-	-	12 minutes	-	
Bayotech-Production-Brian	/	1	-	-	6 minutes	-	

Delete Create user

STEP 6→ Test IAM User Access

- Open a new browser window→ Go to the AWS Console (I will be implementing the Sign-In URL i copied earlier, for example:(<https://Bayotechusers.signin.aws.amazon.com/console>)
- Log in using the IAM user credentials from the downloaded .csv file:
 - Username: Bayotech-Deployment-John
 - Password: (xxxxxxx)

The screenshot shows the AWS IAM user sign-in interface. At the top right is the AWS logo. The main form has the title "IAM user sign in" with an info icon. It contains fields for "Account ID or alias" (with a "Don't have?" link) containing "bayotechusers", a "Remember this account" checkbox, "IAM username" (containing "Bayotech-Production-Brian"), "Password" (represented by a redacted box), a "Show Password" checkbox, and a "Having trouble?" link. A large orange "Sign in" button is at the bottom. Below it is a link "Sign in using root user email". At the bottom is a link "Create a new AWS account". To the right of the form is a blurred vertical banner with the text "AWS re:Invent catalog" and "Browse the catalog explore all and experience" followed by a "View catalog >" link.

- Once logged in, you'll see the AWS Management Console ,but with limited access based on the policy attached to the user's group.
- I tried performing actions such as stopping EC2 instances and view IAM.
- If the IAM policy restricts certain actions (e.g.,Creating user group, launching untagged instances or modifying tags), AWS will display an “Access Denied” message.

This confirms that the IAM policy and group permissions are working as intended, enforcing the Principle of Least Privilege.

The screenshot shows the AWS Management Console home page. In the top right, there is a red box highlighting an 'Access denied' message for the 'servicecatalog>ListApplications' action. Below this, the 'Instances (1/2)' section shows a single EC2 instance named 'Bayotech-Deployment-John' (i-0501b9d91a2279c26) in the 'Running' state. A red box highlights this instance.

Console Home

Recently visited

No recently visited services

Explore one of these commonly visited AWS services.

[EC2](#) [S3](#) [Aurora and RDS](#) [Lambda](#)

[View all services](#)

Applications (0)

Region: US West (N. California)

Select Region: us-west-1 (Current Region)

[Find applications](#)

Access denied to servicecatalog>ListApplications

[Diagnose with Amazon Q](#)

Welcome to AWS

Getting started with

AWS Health

Cost and usage

Current month Cost breakdown

[Go to myApplications](#)

Instances (1/2)

Find Instance by attribute or tag (case-sensitive)

Instance state: All states

Actions

Launch instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Bayotech-Depl...	i-0501b9d91a2279c26	Running	t3.micro	3/3 checks passed	User: arn:aws:	us-west-1a	ec2-54-183-1

i-0501b9d91a2279c26 (Bayotech-Deployment-John)

Details Status and alarms Monitoring Security Networking Storage Tags

The screenshot shows the AWS IAM Dashboard with the following details:

- Identity and Access Management (IAM)** sidebar with options like Dashboard, Access management, Access reports, and User groups.
- IAM Dashboard Info** section with **Security recommendations** and **AWS Account** sections.
- Security recommendations**:
 - Access denied to iam>ListMFADevices**: You don't have permission to `iam>ListMFADevices`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)
 - Access denied to iam>ListAccessKeys**: You don't have permission to `iam>ListAccessKeys`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)
- AWS Account**:
 - Access denied to iam>ListAccountAliases**: You don't have permission to `iamListAccountAliases`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)
 - Access denied to iam:ListGroups**: You don't have permission to `iam:ListGroups`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)
- Quick Links** section with [My security credentials](#).

User groups (0) Info section:

- Search bar: `Q Search`
- Buttons: `Group name`, `Users`, `Permissions`, `Creation time`, `Delete`, `Create group`.
- Access denied to iam>ListGroups**: You don't have permission to `iamListGroups`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)
- Details: `User: arn:aws:iam::172552440564:user/Bayotech-Production-Brian`, `Action: iam:ListGroups`, `On resource(s): arn:aws:iam::172552440564:group/`, `Context: no identity-based policy allows the action`.

SUMMARY

This IAM configuration demonstrates how to:

- Securely manage user identities and permissions in AWS.
- Enforce policy-based access control using JSON configurations.
- Maintain account security and compliance with AWS best practices.

By combining IAM policies, groups, and user testing, I've established a scalable and secure structure the foundation for any enterprise-level Cloud Security and Governance strategy.