

AWS Cloud Security Project - IAM Implementation

Project Overview

This project demonstrates the implementation of AWS Identity and Access Management (IAM) security best practices. The goal is to set up a secure AWS environment with proper user access controls, policies, and EC2 instance management.

Table of Contents

- Prerequisites
- Architecture
- Step-by-Step Implementation
- Screenshots
- Security Best Practices
- Troubleshooting
- Conclusion

Prerequisites

- An active AWS account
- Basic understanding of cloud computing concepts
- Familiarity with AWS console navigation

Architecture

This project implements a basic IAM security structure with:

- AWS Management Console access
- EC2 instances for testing
- Custom IAM policies
- AWS CLI aliases for quick access
- IAM groups and users with specific permissions

Step-by-Step Implementation

Step 1: Setup an AWS Management Console

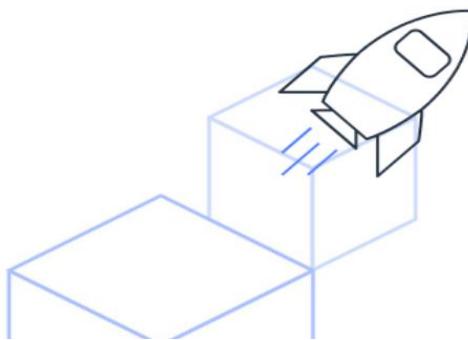
1. Navigate to [AWS Console](#)
2. Sign in with your root account credentials
3. Enable Multi-Factor Authentication (MFA) on your root account for security

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with the AWS logo, links for Discover AWS, Products, Solutions, Pricing, and Resources, a search bar, a 'Sign in to console' button, and a 'Create account' button. Below the navigation bar, a secondary navigation bar includes links for AWS Management Console, Overview, Features, Mobile Application, and FAQs. A breadcrumb trail at the top left indicates the path: Products > Management and Governance > Management Console. The main content area features a large heading 'AWS Management Console' with the subtext 'Everything you need to access and manage the AWS Cloud — in one web interface'. A 'Sign in' button is located on the left, and a dark sidebar is visible on the right.



Try AWS at no cost for up to 6 months

Start with USD \$100 in AWS credits, plus earn up to USD \$100 by completing various activities.



Sign up for AWS

Root user email address

Used for account recovery and as described in the [AWS Privacy Notice](#)

AWS account name

Choose a name for your account. You can change this name in your account settings after you sign up.

Verify email address

OR

Sign in to an existing AWS account

This site uses essential cookies. See our [Cookie Notice](#) for more information.



EC2

- Dashboard
- EC2 Global View
- Events
- Instances**
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
 - Capacity Manager New
- Images

Compute

Amazon Elastic Compute Cloud (EC2)

Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

Benefits and features

Launch a virtual machine

- Launch instance
- View dashboard
- Get started walkthru
- Get started tutorial

Additional actions

View running instances

The screenshot shows the AWS EC2 Resources page. On the left, a sidebar menu includes 'EC2' (selected), 'Dashboard', 'EC2 Global View', 'Events', and a collapsed section 'Instances' containing 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', and 'Savings Plans'. The main content area is titled 'Resources' and displays the following table:

You are using the following Amazon EC2 resources in the United States (Ohio) Region:		
Instances (running)	0	Auto Scaling Groups
Capacity Reservations	0	Dedicated Hosts
Elastic IPs	0	Instances
Key pairs	0	Load balancers
Placement groups	0	Security groups

At the top right of the main content area are two buttons: 'EC2 Global View' and a gear icon.

- Capture the MFA configuration screen

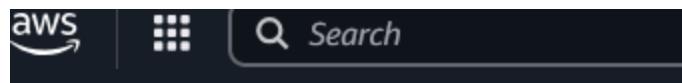
Key Points:

- Never use root account for daily operations
- Always enable MFA on root account
- Note your AWS Account ID for future reference

Step 2: Launch 2 EC2 Instances

1. In the AWS Console, navigate to **EC2 Dashboard**
2. Click **Launch Instance**
3. Configure First Instance:
 - o **Name:** IAM-Test-Instance-1
 - o **AMI:** Amazon Linux 2023 (or latest)
 - o **Instance Type:** t2.micro (Free tier eligible)
 - o **Key Pair:** Create new key pair or select existing
 - o **Network Settings:** Default VPC
 - o **Security Group:** Allow SSH (Port 22) from your IP
 - o Click **Launch Instance**
4. Repeat for Second Instance:
 - o **Name:** IAM-Test-Instance-2
 - o Use same configuration as Instance 1

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The current step is 'Name and tags'. A blue header bar at the top contains a message: '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'. Below this, the 'Launch an instance' section title is followed by a sub-section titled '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.' The 'Name and tags' section is expanded, showing two sets of tags being configured. Each set consists of a 'Key' (e.g., 'Name' or 'Env') and a 'Value' (e.g., 'cybersecurity-Ec2' or 'Production'). Each tag entry has a 'Remove' button and an 'Instances' button. Below the tags, there is a 'Select resource types' dropdown and a 'Remove' button. At the bottom of the 'Name and tags' section, there is a 'Add new tag' button and a note: 'You can add up to 48 more tags.'



☰ [EC2](#) > [Instances](#) > Launch an instance

Success

Successfully initiated launch of instance i-0703f6052c9f2f066

▶ [Launch log](#)

Next Steps

🔍 What would you like to do next with this instance?

Create billing usage alerts

To manage costs and avoid surprise bills, set notifications for billing usage thresholds.

[Create billing alerts ↗](#)

☰ [EC2](#) > [Instances](#)

Instances (1/1) [Info](#)

less than

🔍 Find Instance by attribute or tag (case-sensitive)

All states ▾

<input checked="" type="checkbox"/>	Name ↕	Instance ID	Instance state	Instance type	Status check	Alarm status
<input checked="" type="checkbox"/>	cybersecurity-...	i-0703f6052c9f2f066	Running	t2.micro	Initializing	View alarms +

The screenshot shows the AWS EC2 Instances dashboard. At the top, there's a search bar and navigation links for 'EC2' and 'Instances'. Below the header, it says 'Instances (1/2) Info'. There's a search input field and a dropdown for 'All states'. A table lists two instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input checked="" type="checkbox"/>	cybersecurity-...	i-0703f6052c9f2f066	Running	t2.micro	2/2 checks passed	View alarms +
<input type="checkbox"/>	CyberSecurity-...	i-0d6941670f8966b16	Running	t2.micro	Initializing	View alarms +

- Capture the EC2 dashboard showing both instances in "running" state

Important Notes:

- Save your key pair (.pem file) securely
- Note the Instance IDs for both instances
- Verify both instances are in "running" state

Step 3: Create an IAM Policy

1. Navigate to **IAM Dashboard**
2. Click **Policies** in the left sidebar
3. Click **Create Policy**
4. Select **JSON** tab
5. Enter the following policy (example - EC2 read-only access):

```
json
```

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:Describe*",  
                "ec2:Get*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

6. Click **Next: Tags** (optional)

7. Click **Next: Review**

8. **Policy Name:** EC2-ReadOnly-Policy

9. **Description:** "Allows read-only access to EC2 resources"

10. Click **Create Policy**

Policy editor

```
1  "Version": "2012-10-17",
2  "Statement": [
3  {
4    "Effect": "Allow",
5    "Action": "ec2:*",
6    "Resource": "*",
7    "Condition": {
8      "StringEquals": {
9        "ec2:ResourceTag/Env": "Audit"
10       }
11     }
12   },
13 },
14 {
15   "Effect": "Allow",
16   "Action": "ec2:Describe*",
17   "Resource": "*"
18 },
19 {
20   "Effect": "Deny",
21   "Action": [
22     "ec2>DeleteTags",
23     "ec2>CreateTags"
24   ],
25   "Resource": "*"
26 }
27 ]
28 }
29 ||
```

[+ Add new statement](#)

JSON Ln 29, Col 0

! Security: 0 ✖ Errors: 0 ⚠ Warnings: 0 💡 Suggestions: 0

The screenshot shows the AWS IAM Policies page. At the top, a green banner displays the message: "Policy CyberSecurity-Audit-Env-Policy created." Below this, the main title is "Policies (1441) Info". A sub-instruction states: "A policy is an object in AWS that defines permissions." On the right, there are "Filter by Type" options set to "All types". A search bar at the top right contains the placeholder "[Alt+S] Search". The main content area lists policies, with one entry highlighted: "CyberSecurity-Audit-Env-Policy" (Customer managed). A search bar on the left side of the main content area contains the text "cyber".

- Capture the success message after policy creation

Policy Explanation:

- ec2:Describe* - Allows listing all EC2 resources
- ec2:Get* - Allows retrieving EC2 resource details
- Resource: "*" - Applies to all EC2 resources

Step 4: Create an AWS Alias

1. In the IAM Dashboard, look for **AWS Account** section (top right)
2. Click on your **Account ID**
3. Click **Create** next to Account Alias
4. Enter a unique alias (e.g., mycompany-aws-prod)
5. Click **Create Alias**

The screenshot shows the AWS IAM Dashboard. At the top, there are two notifications: one about an alias being created and another about new access analyzers. Below the notifications, the 'IAM Dashboard' section is visible, featuring a 'Security recommendations' card with a count of 0 and a refresh button. On the left sidebar, under 'Access Management', 'User groups' is selected. The main content area displays a user summary card for the root user, indicating it has no active access keys. To the right, there are links for 'Account Alias' (bjnetworksolution) and a 'Sign-in URL for IAM users in this account'. The bottom of the dashboard shows navigation links for 'Users', 'Roles', and 'Policies'.

Benefits of Account Alias:

- Easier sign-in URL: <https://bjnet.signin.aws.amazon.com/console>
- More user-friendly than using account ID
- Professional appearance for team members

Step 5: Create IAM Group and User

Create IAM Group

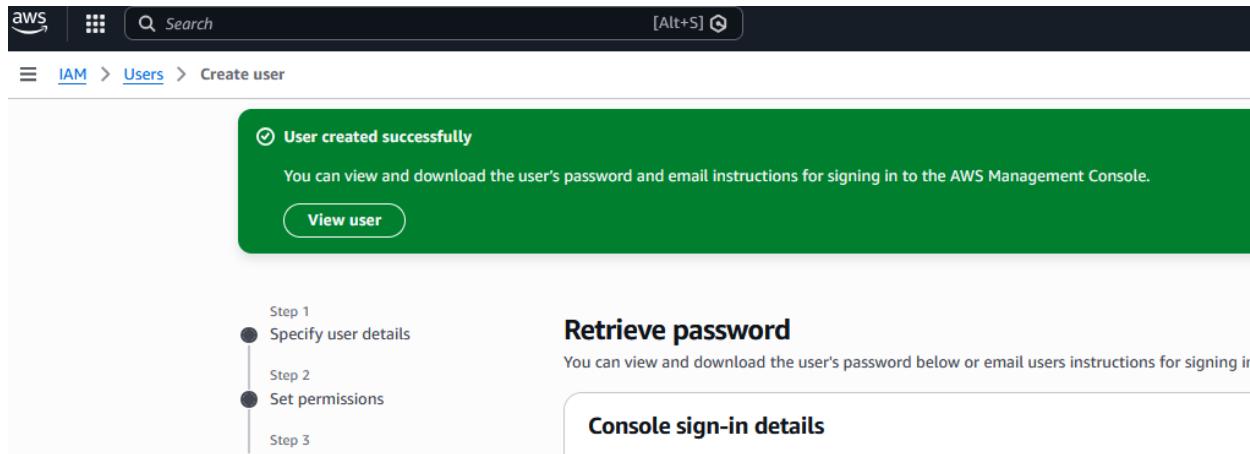
1. In IAM Dashboard, click **User groups**
2. Click **Create Group**
3. **Group Name:** EC2-Administrators
4. Attach the policy created in Step 3: EC2-ReadOnly-Policy
5. Click **Create Group**

The screenshot shows the 'User groups' page in the AWS IAM Dashboard. A success message at the top indicates a user group was created. The main table lists a single user group named 'cybersecurity-Audit-Group'. The table includes columns for 'Group name' and 'Users'. There is also a warning icon with a count of 0. The left sidebar shows the 'User groups' section is selected under 'Access Management'.

Create IAM User

1. Click **Users** in IAM Dashboard
2. Click **Add Users**
3. **User Name:** test-user-1
4. **Access Type:** Select both:
 - AWS Management Console access
 - Programmatic access (for API/CLI)
5. **Console Password:** Choose "Autogenerated" or "Custom"
6. **Require Password Reset:** (Recommended)
7. Click **Next: Permissions**
8. Select **Add user to group**
9. Select EC2-Administrators group
10. Click **Next: Tags** (optional)
11. Click **Next: Review**
12. Click **Create User**
13. **IMPORTANT:** Download the credentials CSV file

The screenshot shows the 'Set permissions' step of the IAM user creation wizard. On the left, a sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions, which is selected and highlighted in blue), Step 3 (Review and create), and Step 4 (Retrieve password). The main area is titled 'Set permissions' with the sub-instruction: '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.' Below this is a 'Permissions options' section containing three radio button choices: 'Add user to group' (selected), 'Copy permissions', and 'Attach policy'. Under 'User groups (1/2)', there is a table showing one group assignment: 'Group name: cybersecurity-Audit-Group' and 'Attached policies: CyberSecurity-Audit-Env-Policy'.



Security Notes:

- Store credentials securely
- Never commit credentials to Git
- Enable MFA for all IAM users in production

Step 6: Test The IAM User Access

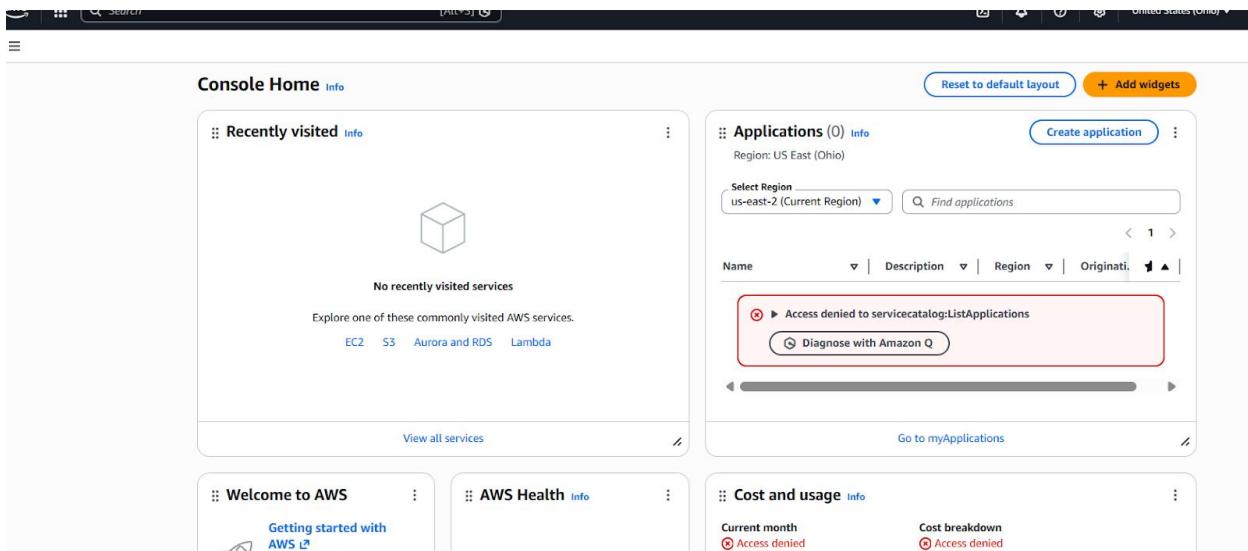
Test Console Access

1. Sign out of the root account
2. Navigate to the sign-in URL with your alias
3. Log in with the IAM user credentials:
 - **Account ID/Alias:** Your account alias
 - **Username:** test-user-1
 - **Password:** The password from Step 5
4. If prompted, change the password

5. Navigate to EC2 Dashboard
6. Verify you can **view** EC2 instances (read-only)
7. Try to **stop** an instance - this should fail with "Access Denied"



The image shows the 'Console Home' dashboard. At the top, there's a search bar and navigation links for 'Search', 'Dashboard', and 'Region: United States (Ohio)'. Below the search bar are three main cards: 'Recently visited' (empty), 'Applications (0)' (empty, with a note about access denied to 'servicecatalog>ListApplications'), and 'Cost and usage' (empty, with a note about access denied). At the bottom, there are three smaller cards: 'Welcome to AWS' (with a 'Getting started with AWS' link), 'AWS Health' (empty), and 'Cost and usage' (empty).



Security Best Practices

Implemented in This Project

- Root account MFA enabled
- IAM users instead of root for daily operations
- Least privilege principle (read-only policy)
- IAM groups for permission management
- Password policy enforcement

Additional Recommendations

1. **Enable CloudTrail:** Log all API calls for auditing
2. **Set up billing alerts:** Monitor unexpected charges
3. **Rotate access keys regularly:** Every 90 days minimum
4. **Use IAM roles for EC2:** Instead of embedding credentials
5. **Implement password policy:** Require complex passwords
6. **Regular access reviews:** Remove unused users/permissions
7. **Enable AWS Organizations:** For multi-account management

Troubleshooting

Issue: Cannot launch EC2 instances

Solution:

- Check if you've reached the instance limit (default is 20 per region)
- Verify you're in a region that supports t2.micro instances
- Ensure your account has EC2 service permissions

Issue: IAM user cannot sign in

Solution:

- Verify the sign-in URL includes your account alias or ID
- Check username and password are correct
- Ensure the user has console access enabled

Issue: Policy not working as expected

Solution:

- Review the JSON syntax for errors
- Check the policy simulator: IAM → Policies → Simulate
- Verify the policy is attached to the correct group/user

Issue: Access denied errors

Solution:

- Verify the user is in the correct IAM group
- Check that the policy is attached to the group
- Wait a few minutes for IAM changes to propagate

Conclusion

This project demonstrates fundamental AWS IAM security concepts:

- Proper account setup and security
- EC2 instance management
- Custom IAM policy creation
- Group-based permission management
- User access testing and validation

Author

Your Name -Bolaji Bakare

Acknowledgments

- AWS Documentation Team
- Cloud Security Community