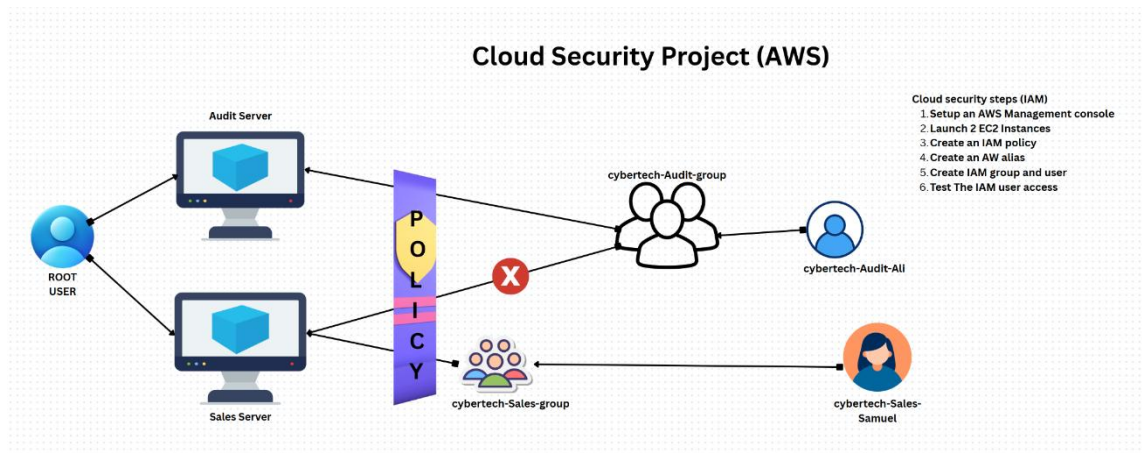


AWS IAM Cloud Security Project

1. Project Overview

I completed this project on cloud security controls in Amazon Web Services (AWS), focusing on Identity and Access Management (IAM). The goal was to create a least-privilege policy, attach it to a user group, and verify that the policy correctly restricts actions on two Amazon EC2 instances (audit and sales).



2. Tools & Concepts

- AWS IAM – users, groups, policies, account alias
- Amazon EC2 – instance tagging and lifecycle actions
- JSON policy syntax – Effect, Action, Resource
- Principle of least privilege and policy testing

3. Tagging Strategy

I applied a descriptive tag to each EC2 instance:

Instance | Tag Key | Tag Value

Audit | Environment | Audit

Sales | Environment | Sales

Instances (1/2) Info								
Last updated 4 minutes ago Refresh Connect Instance state Actions Launch instances								
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> All states < 1 > Settings								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
<input checked="" type="checkbox"/>	Cybertech-Au...	i-00b975b44258ef783	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2a	ec2-3
<input type="checkbox"/>	Cybertech-Sal...	i-0eb4d3209a7e36b52	Running	t2.micro	Initializing	View alarms +	us-east-2a	ec2-3

4. Creating the IAM Policy

I authored the following JSON policy to deny instance stop/start actions on the Sale Server but allow those actions on the Audit Server. I named the policy **CybertechAuditEnvPolicy**:

Permissions defined in this policy [info](#)

[Copy](#)[Edit](#)[Summary](#)[JSON](#)

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


```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "ec2:*",  
7       "Resource": "*",  
8       "Condition": {  
9         "StringEquals": {  
10          "ec2:ResourceTag/Env": "Audit"  
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 }
```

5. Account Alias

I set a memorable account alias to replace the default numeric URL, making signing in easier for team members.

AWS Account


Account ID

 356062354220

Account Alias



cybertech-users [Edit](#) | [Delete](#)

Sign-in URL for IAM users in this account

 <https://cybertech-users.signin.aws.amazon.com/console>



6. IAM Users & Groups

1. Created an IAM user group called cybertech-Audit-group.
2. Attached the **CybertechAuditEnvPolicy** policy to the group.
3. Added individual IAM users who require controlled EC2 access.

 cybertech-Audit-group user group created. [View group](#) 

User groups (1) [info](#)

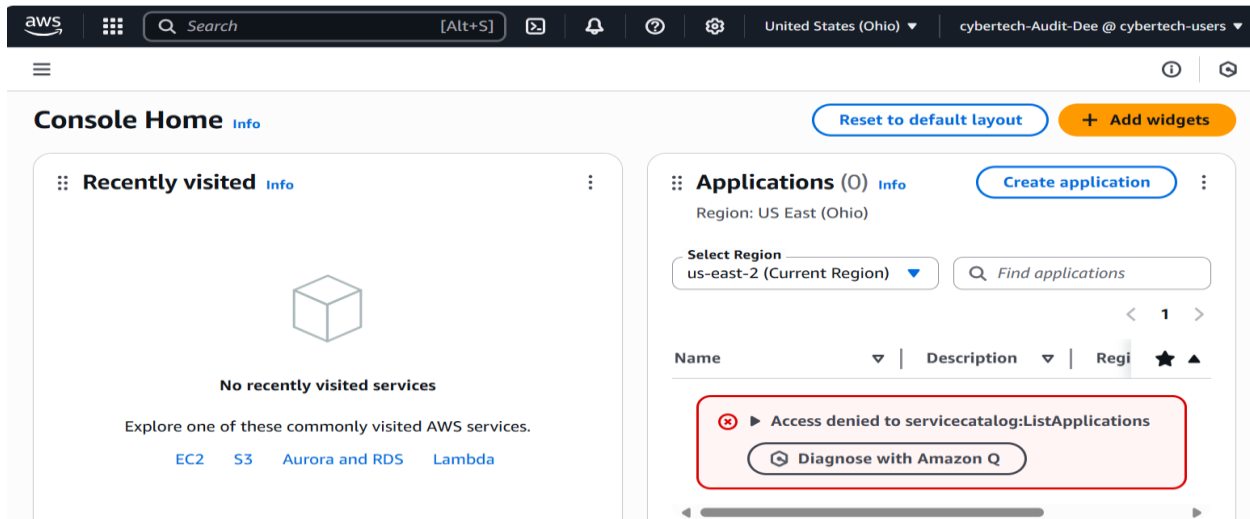
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

<input type="checkbox"/>	Group name	Users	Permissions	Creation time
<input type="checkbox"/>	cybertech-Audit-group	 0	 Defined	Now

7. Logging in as an IAM User

IAM users can sign in through:

- AWS Management Console (using the new alias URL)
- AWS CLI via programmatic keys



8. Testing the Policy

Test Action | Expected Result | Actual Result

Stop sale instance | Denied | Access denied error displayed

Stop audit instance | Allowed | Instance stopped successfully

Start sale instance | Denied | Access denied error displayed

Start audit instance | Allowed | Instance started successfully

