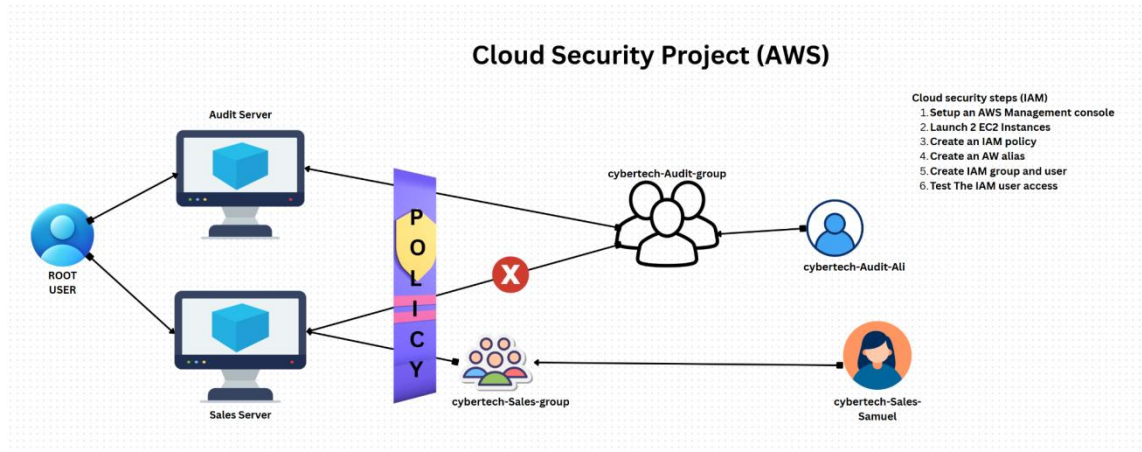


# AWS IAM Cloud Security Project

## 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

Instance	Tag Key	Tag Value
audit	Environment	Audit
sales	Environment	Sales

Instances (2) <a href="#">Info</a>								
<div> <div>Find Instance by attribute or tag (case-sensitive)</div> <div>All states</div> </div>			<div> <div>Last updated less than a minute ago</div> <div>Connect</div> <div>Instance state</div> <div>Actions</div> <div>Launch instances</div> </div>					
<input type="checkbox"/>	Name <a href="#">🔗</a>	Instance ID	Instance state <a href="#">▼</a>	Instance type <a href="#">▼</a>	Status check	Alarm status	Availability Zone <a href="#">▼</a>	Public IPv4
<input type="checkbox"/>	cybertec-sales...	i-007648ef12df1e56d	Running <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	Initializing	<a href="#">View alarms +</a>	eu-north-1b	ec2-13-60
<input type="checkbox"/>	cybertec-audit...	i-0463424610ce6ed8f	Running <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	Initializing	<a href="#">View alarms +</a>	eu-north-1b	ec2-13-60

#### 4. Creating the IAM Policy

I created an IAM policy in JSON format that explicitly prevents stop and start operations on the audit server, while permitting those same actions on the sales server. This policy enforces operational restrictions by using instance-level conditions, ensuring the audit server remains continuously available while allowing controlled management of the sales server.

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 sign-in easier for team members.

## AWS Account


### Account ID

 525184038043

### Account Alias

cybertecusers [Edit](#) | [Delete](#)

### Sign-in URL for IAM users in this account

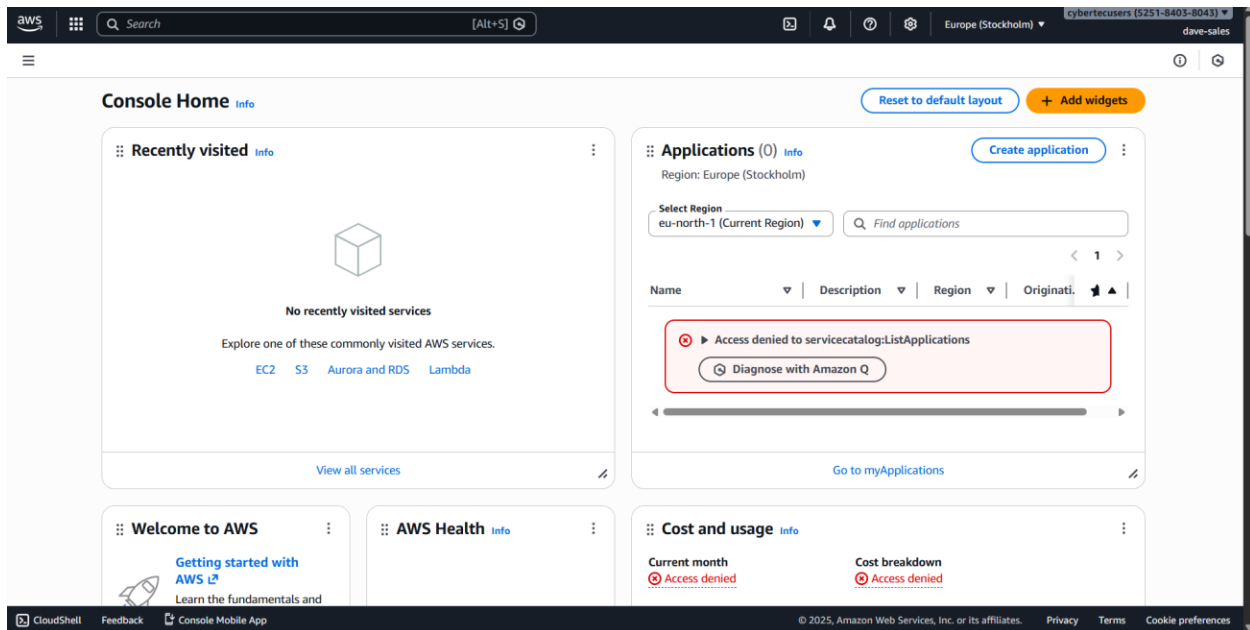
 <https://cybertecusers.signin.aws.amazon.com/console>

## 6. IAM Users & Groups

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

## 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	audit instance	Denied	Access denied	error	displayed
Stop	sales instance	Allowed	Instance	stopped	successfully
Start	audit instance	Denied	Access denied	error	displayed
Start	sales instance	Allowed	Instance started		successfully

