AWS Identity and Access Management (IAM) Security Implementation

A Comprehensive Guide to Tag-Based Access Control and Least-Privilege Policy Design

Summary

This technical implementation guide documents the deployment of a robust identity and access management security framework within Amazon Web Services (AWS). The project demonstrates enterprise-grade security controls through the implementation of tag-based resource access policies, enabling granular permissions management while maintaining strict compliance and audit capabilities.

Project Scope: Design and implementation of conditional IAM policies restricting EC2 instance management based on environment classification tags.

Key Deliverables:

- Custom IAM policy with tag-based conditional access
- Segregated user group architecture
- Comprehensive audit trail via CloudTrail integration
- Validated security controls through systematic testing

Technical Stack: AWS IAM, Amazon EC2, AWS CloudTrail, JSON Policy Language

Part I: Foundation & Architecture

1.1 Business Context

Modern cloud environments require sophisticated access control mechanisms that balance operational flexibility with security compliance. This implementation addresses a common enterprise scenario: enabling junior administrators to manage non-production workloads while protecting critical audit infrastructure from unauthorized modifications.

Problem Statement:
How can organizations grant EC2 management permissions to operations teams while ensuring audit and compliance servers remain protected from accidental or unauthorized state changes?

Solution Architecture:
Implement attribute-based access control (ABAC) using resource tags as the conditional element in IAM policies, combined with explicit deny rules to prevent privilege escalation through tag manipulation.

1.2 Core Technologies

Technology	Purpose	Implementation Role
AWS IAM	Identity & Access Management	Policy creation, user/group management, authentication
Amazon EC2	Elastic Compute Cloud	Target resources for access control policies
AWS CloudTrail	Audit & Compliance Logging	Activity monitoring and forensic analysis
JSON Policy Syntax	Policy Definition Language	Declarative security rule expression

1.3 Security Principles Applied

Principle of Least Privilege (PoLP)
Users receive only the minimum permissions necessary to perform their job functions. Access is explicitly granted rather than implicitly allowed.

Defense in Depth
Multiple security layers: IAM authentication, policy-based authorization, tag-based resource boundaries, explicit deny overrides, and comprehensive audit logging.

Separation of Duties
Root account used only for initial setup; administrative functions delegated to IAM users; operational access managed through groups.

Part II: Environment Preparation

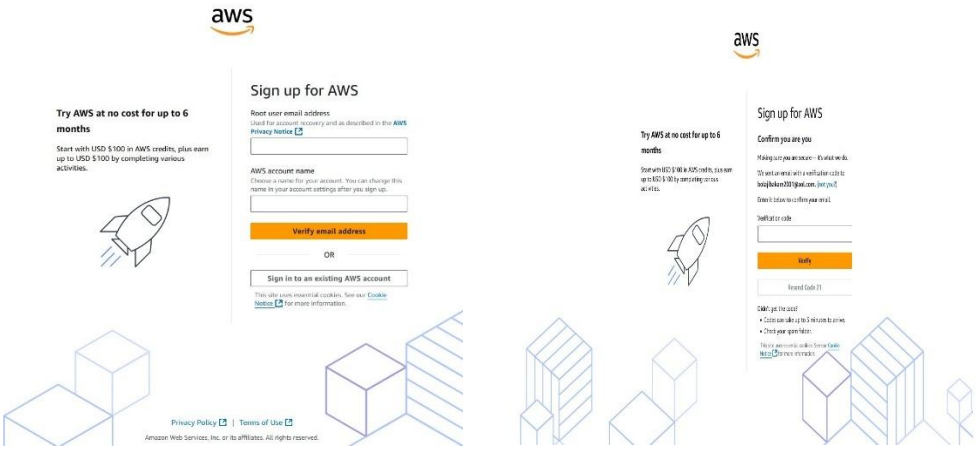
2.1 AWS Account Initialization

Objective: Establish a new AWS account with proper foundational security configuration.

Step 1: Account Creation

1. Navigate to the AWS sign-up portal: https://aws.amazon.com/console/
2. Select **"Create an AWS Account."**
3. Complete registration workflow:
    - Email verification
    - Account name designation
    - Billing information (Free Tier eligible)
    - Identity verification via phone/SMS
    - Support plan selection: **Basic Support (Free)**

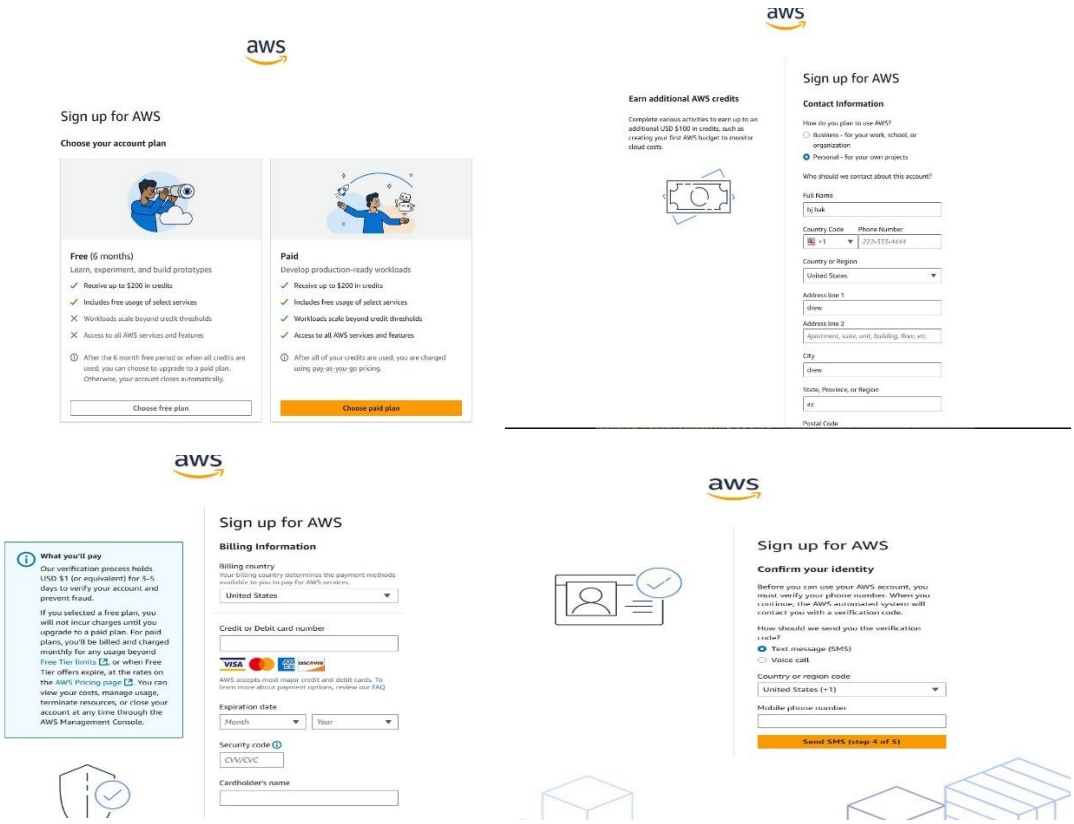
📸 **Visual Reference:** AWS registration page with account creation form

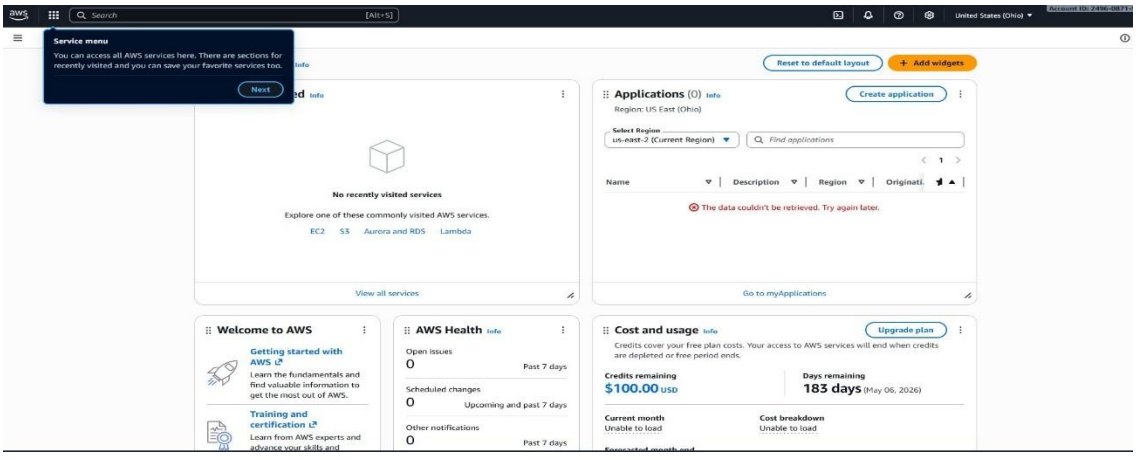


Step 2: Initial Root Account Access

1. Return to https://aws.amazon.com/console/
2. Click **"Sign in to the Console"**
3. Select **"Root user"** authentication method
4. Authenticate using registered email and password

📸 **Visual Reference:** AWS Management Console sign-in page showing root user option





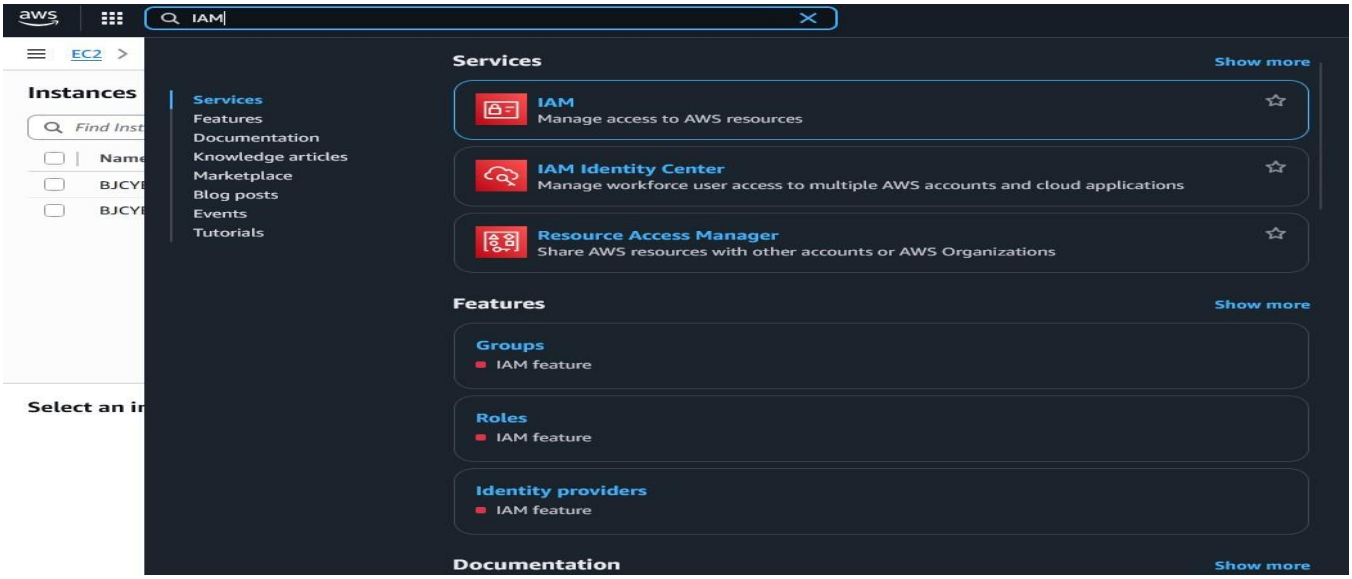
⚠️ **Critical Security Note:**
The root account possesses unrestricted access to all AWS resources and billing information. Best practice dictates that the root account be used only for account-level configurations (billing, account closure, root access keys). All operational activities should utilize IAM users with appropriate permissions.

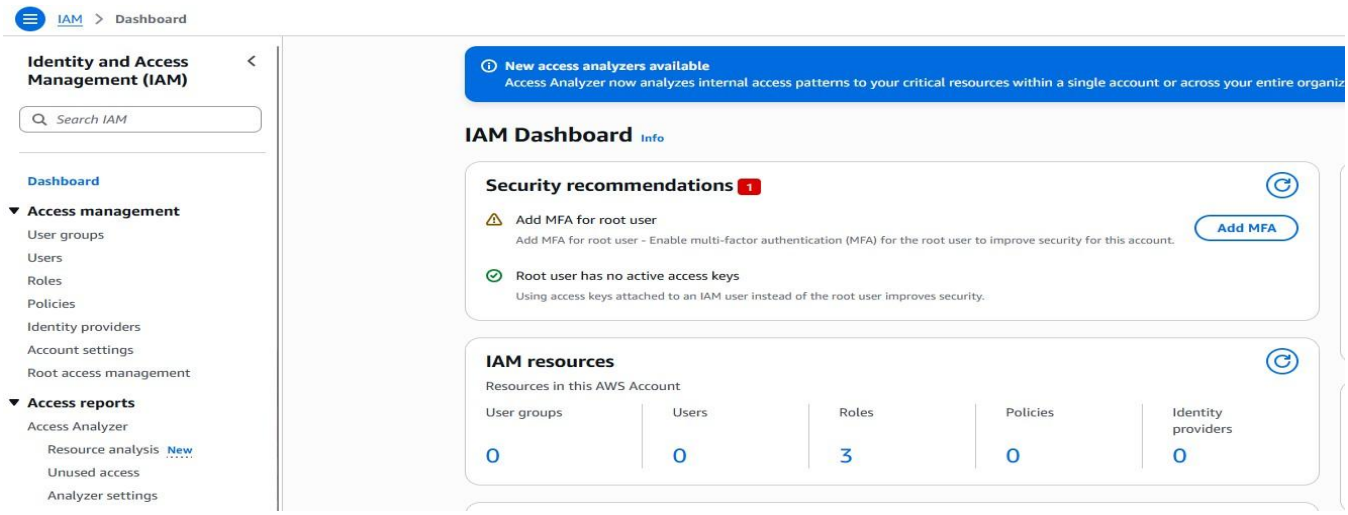
2.2 IAM Administrative User Configuration

Objective: Create a dedicated IAM user with administrative privileges for daily operations, eliminating the need for root account access.

Step 3: Access IAM Service

1. From the AWS Management Console dashboard, utilize the universal search bar
2. Enter **"IAM"** (Identity and Access Management)
3. Select the IAM service from the search results





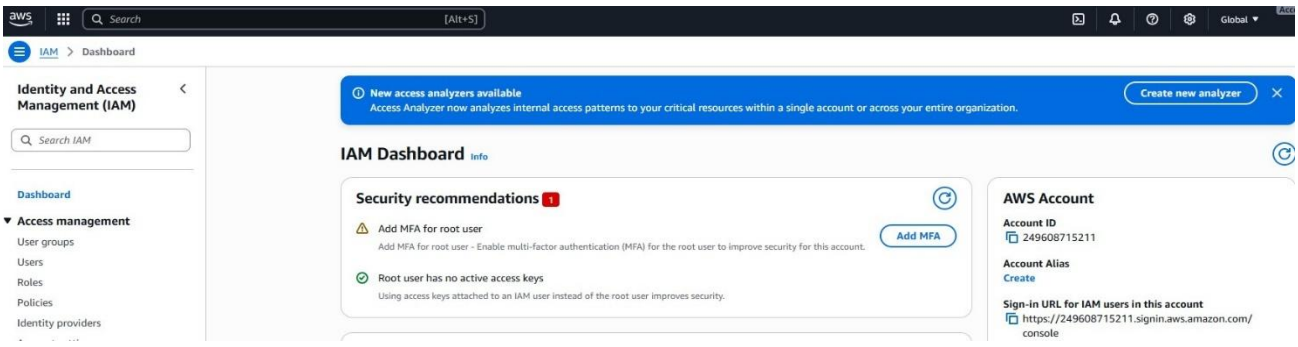
Step 4: Configure Account Alias (Optional - Recommended)

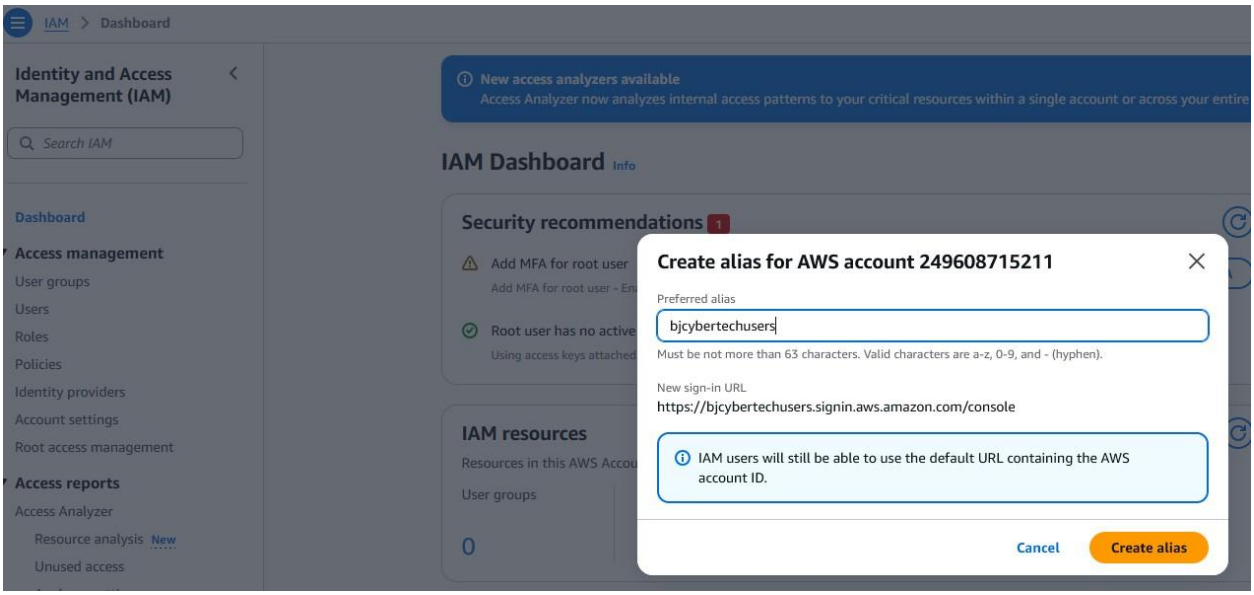
Account aliases create user-friendly sign-in URLs, improving user experience and organizational branding.

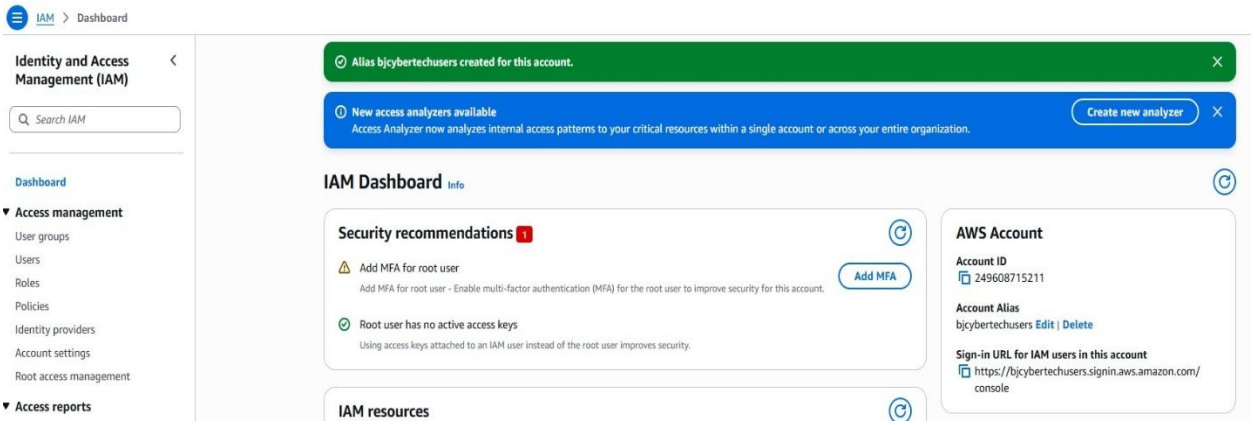
Configuration Process:

1. Locate the **"AWS Account"** information panel (right sidebar)
2. Identify the **"Account Alias"** section
3. Click **"Create"** or **"Edit."**
4. Enter unique alias (format: organization-environment-purpose)
    - Example: acme-corp-production or techcompany-dev
5. Click **"Save changes."**

Result: Custom IAM sign-in URL generated
https://[your-alias]. signin.aws.amazon.com/console





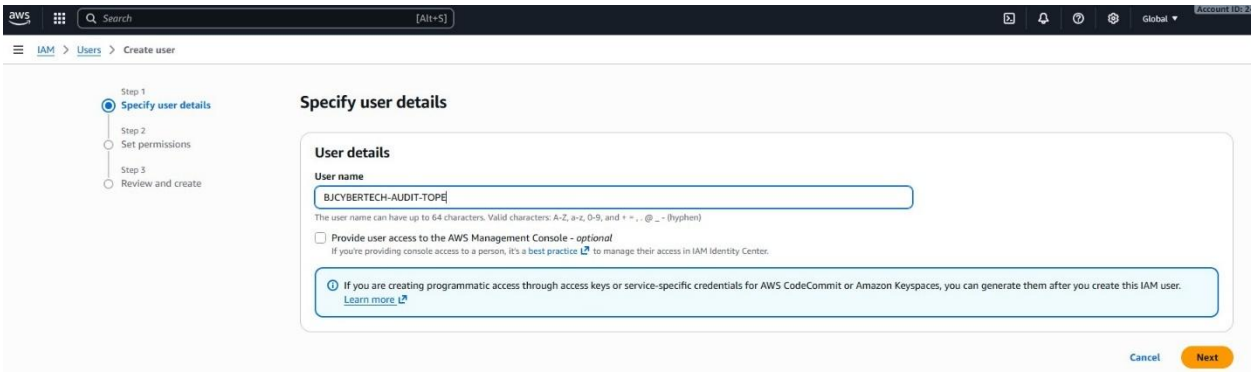


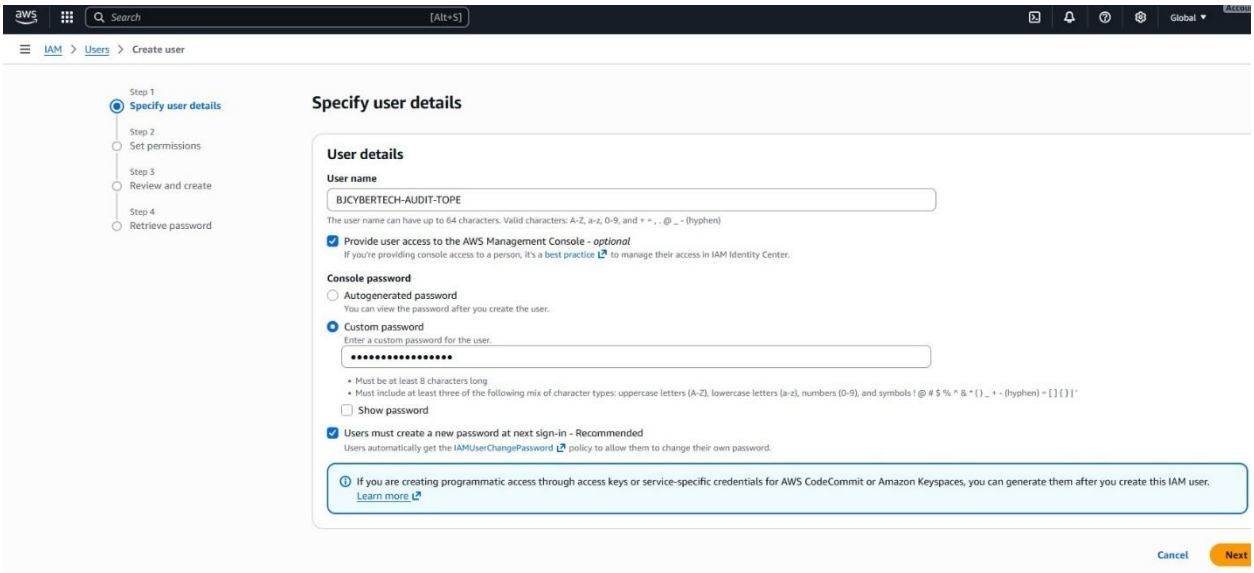
Step 5: Create Administrative IAM User

1. Navigate to **IAM Dashboard → Users**
2. Click the **"Create user"** button
3. **User Configuration:**
    - **Username:** admin-user (or follow organization naming convention)
    - **Access Type:** Enable **"Provide user access to the AWS Management Console."**

- **User Type:** Select **"I want to create an IAM user."**
- **Console Password:** Choose **"Custom password."**
- Enter a strong password (minimum 14 characters, mixed case, numbers, symbols)
- **Password Reset:** Uncheck **"Users must create a new password at next sign-in"** (optional, based on security policy)

4. Click **"Next"** to proceed to permissions



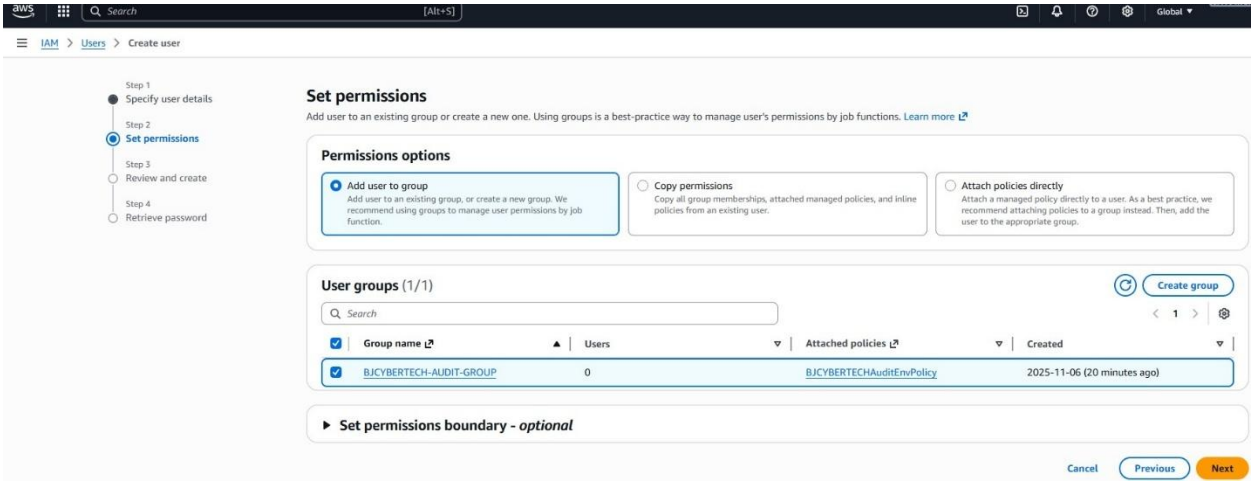


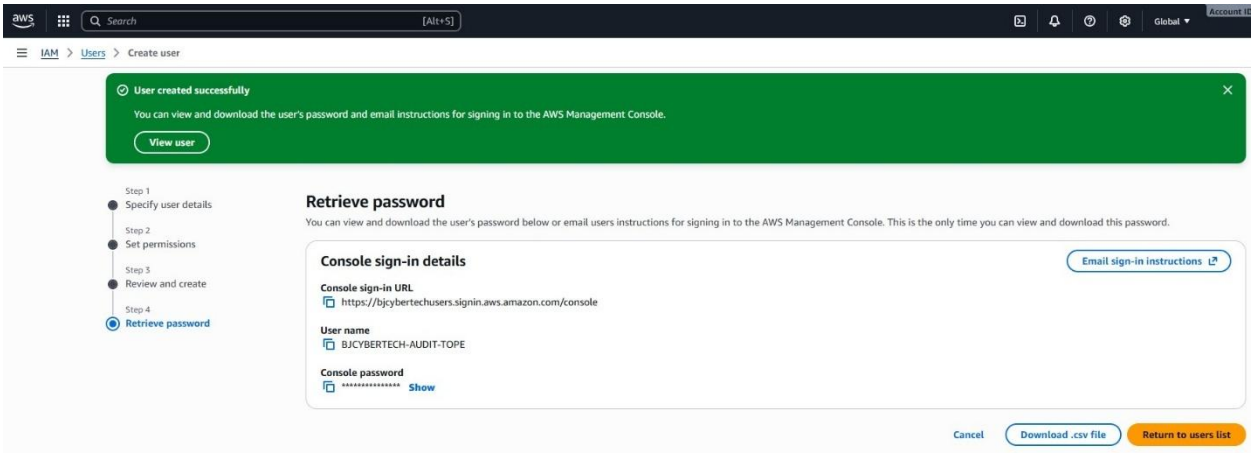
Step 6: Grant Administrative Permissions

Permission Assignment Strategy:

1. Select the **"Attach policies directly"** method
2. Search for **"AdministratorAccess"** in the policy filter
3. Select the **AdministratorAccess** managed policy checkbox
4. Click **"Next"** to review
5. Verify configuration and click **"Create user"**

Policy Details:
AdministratorAccess provides full access to all AWS services and resources. This is appropriate for administrative users but should be restricted to a minimal number of trusted personnel.



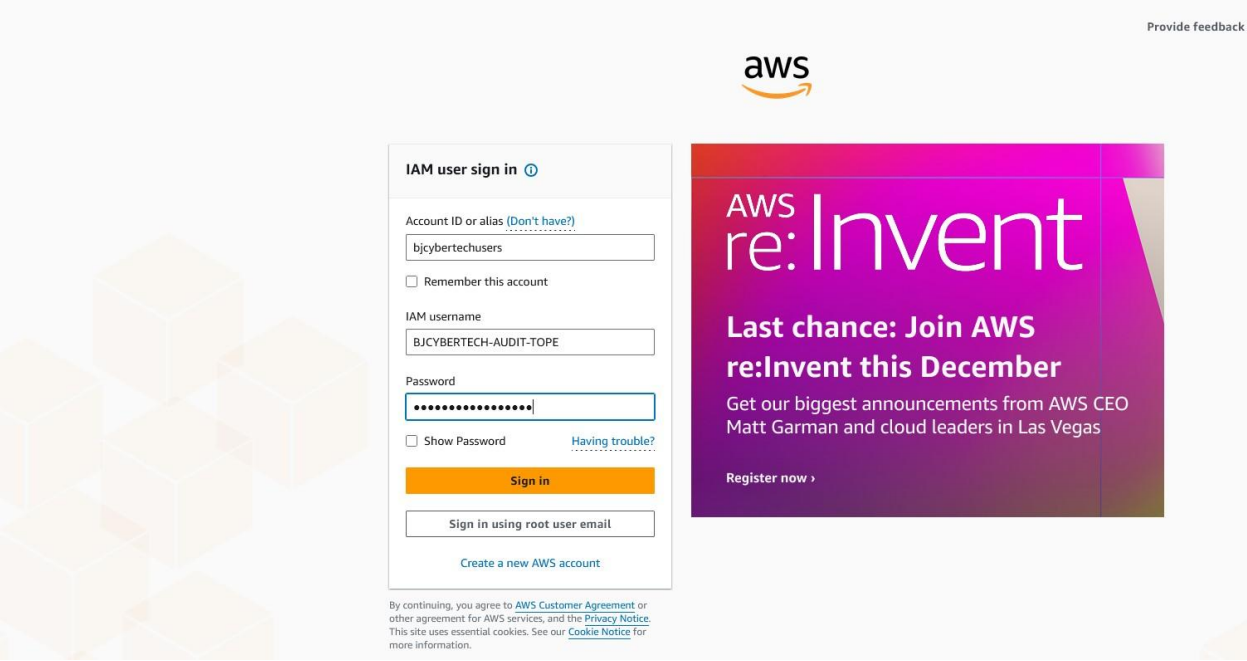


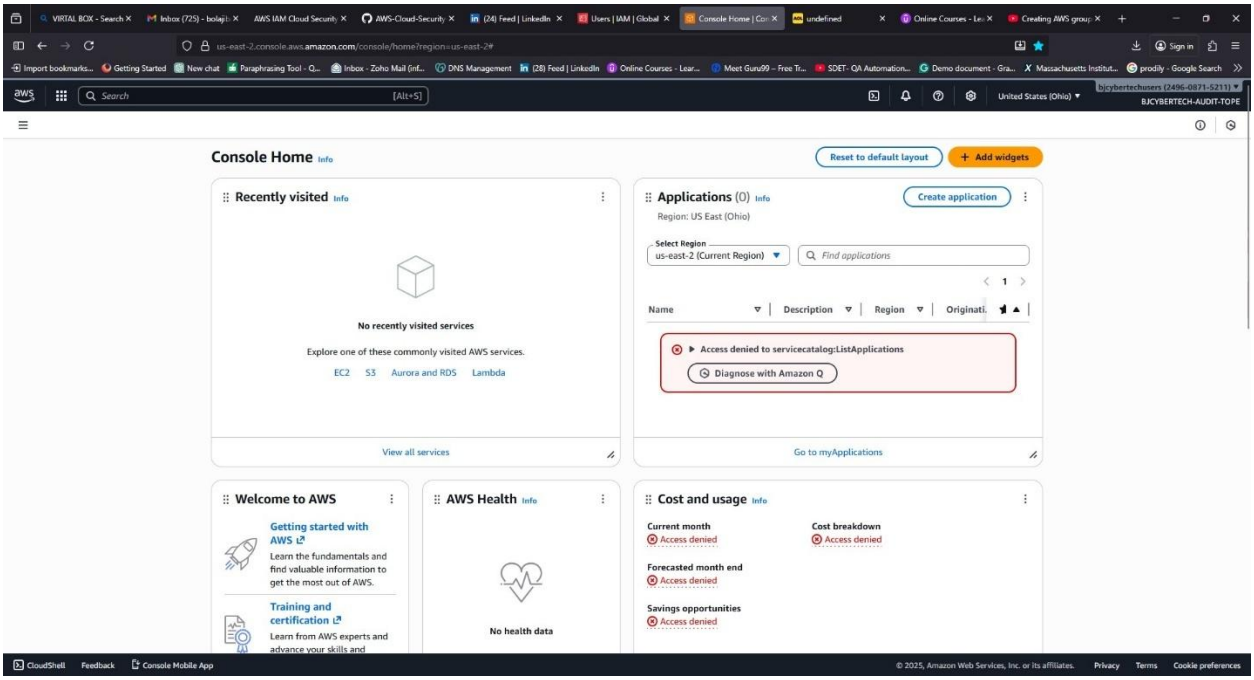
Step 7: Validate Administrative Access

Testing Process:

1. Sign out of the root account session
2. Navigate to custom alias URL: https://[your-alias].signin.aws.amazon.com/console
3. Authenticate using newly created IAM credentials:
    - **Account ID or alias:** Pre-populated or enter alias
    - **IAM username:** admin-user
    - **Password:** As configured
4. Verify successful authentication and console access

✅ **Checkpoint:** From this point forward, all configuration activities should be performed using the IAM administrator account, not the root account.





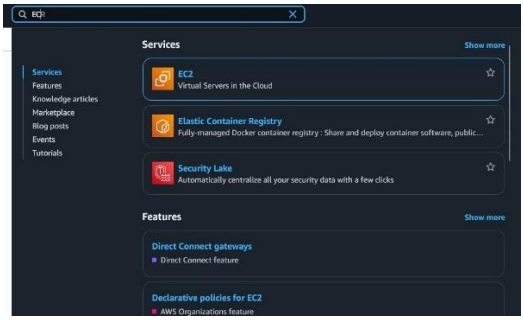
Part III: Infrastructure Deployment

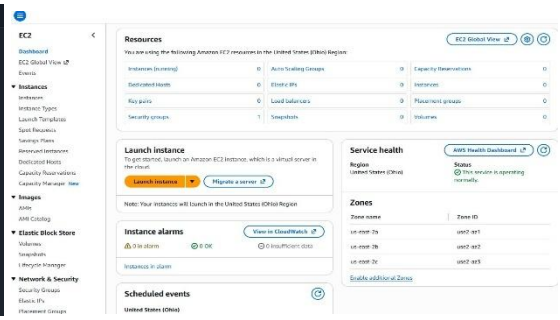
3.1 EC2 Instance Provisioning

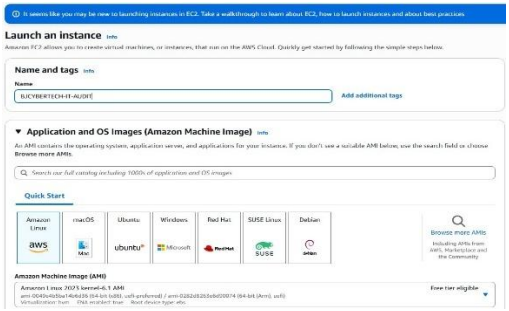
Objective: Deploy two EC2 instances representing different environmental classifications (Audit and Sales) to serve as test subjects for policy validation.

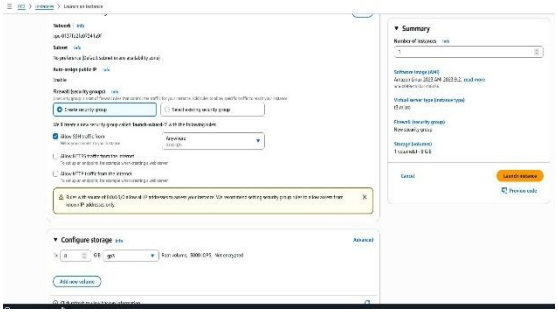
Step 8: Navigate to EC2 Service

1. Access the AWS Management Console as an IAM admin user
2. Search for **"EC2"** using the service search bar
3. Select **EC2** to open the Elastic Compute Cloud dashboard
4. **Region Verification:** Confirm appropriate AWS region selection (top-right corner)
    - Recommended: us-east-1 (N. Virginia) for initial testing
    - All resources in this project must reside in the same region









Step 9: Launch Audit Environment Instance

1. Click the **"Launch instance"** button
2. **Instance Configuration Parameters:**

Name and Tags:

- Name: audit-server

Application and OS Images (Amazon Machine Image):

- AMI: **Amazon Linux 2023 AMI**
- Architecture: **64-bit (x86)**
- Free tier eligible: ✅

Instance Type:

- Type: **t2.micro**
- vCPUs: 1
- Memory: 1 GiB

- Free tier eligible: ✅

Key Pair (Login Credentials):

- Action: **Create new key pair**
- Key pair name: my-ec2-key
- Key pair type: **RSA**
- Private key file format:
  - **.pem** (for macOS, Linux, or Windows 10+)
  - **.ppk** (for Windows with PuTTY)
- Click **"Create key pair."**
- **Important:** Download and securely store the key file—this is the only opportunity to obtain it

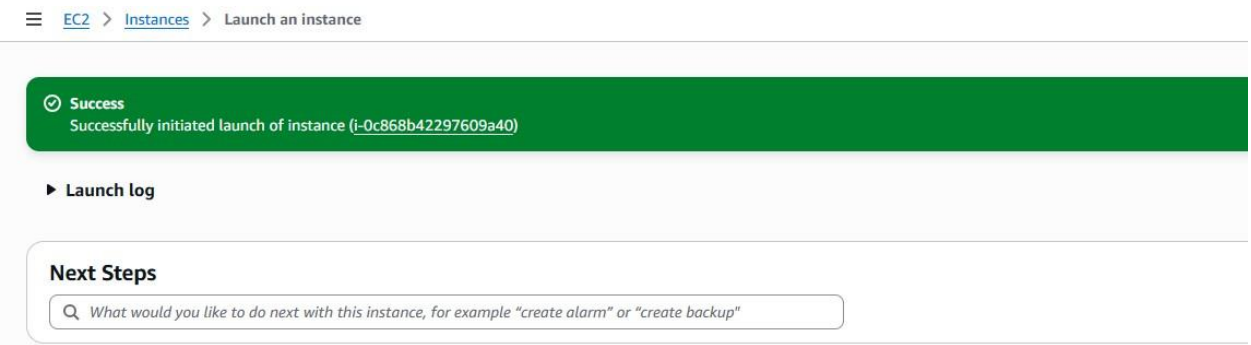
Network Settings:

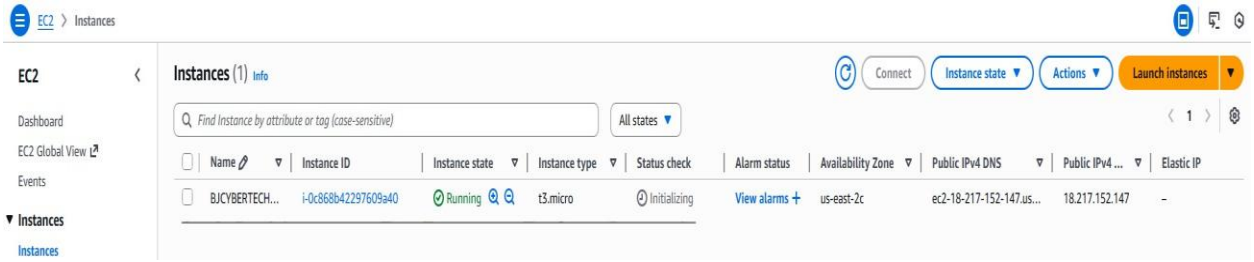
- Accept default VPC configuration
- Auto-assign public IP: **Enable**
- Security group: Create new (default rules)

Configure Storage:

- Volume type: **gp3** (General Purpose SSD)
- Size: **8 GiB** (Free Tier eligible)

3. Click **"Launch instance."**

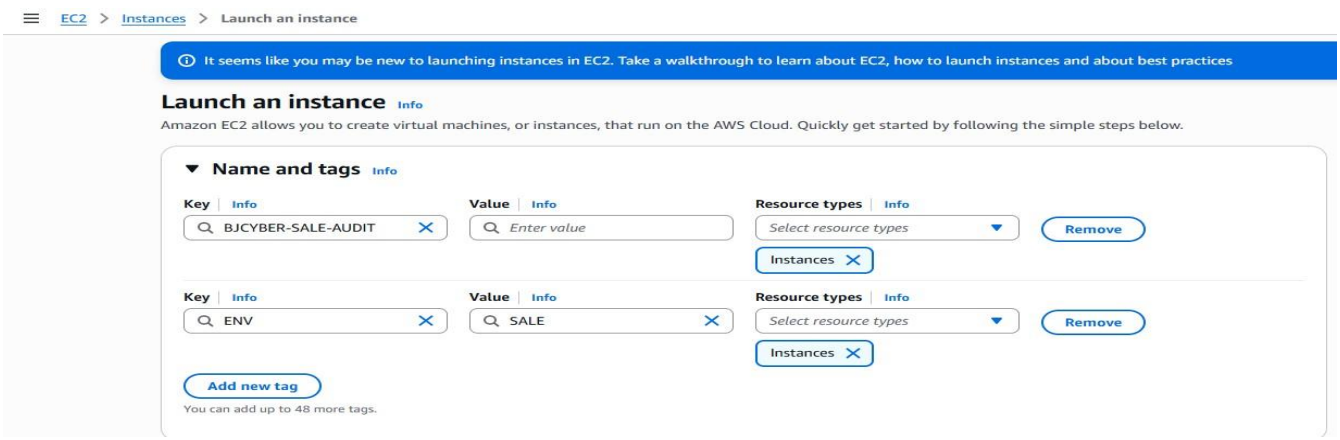


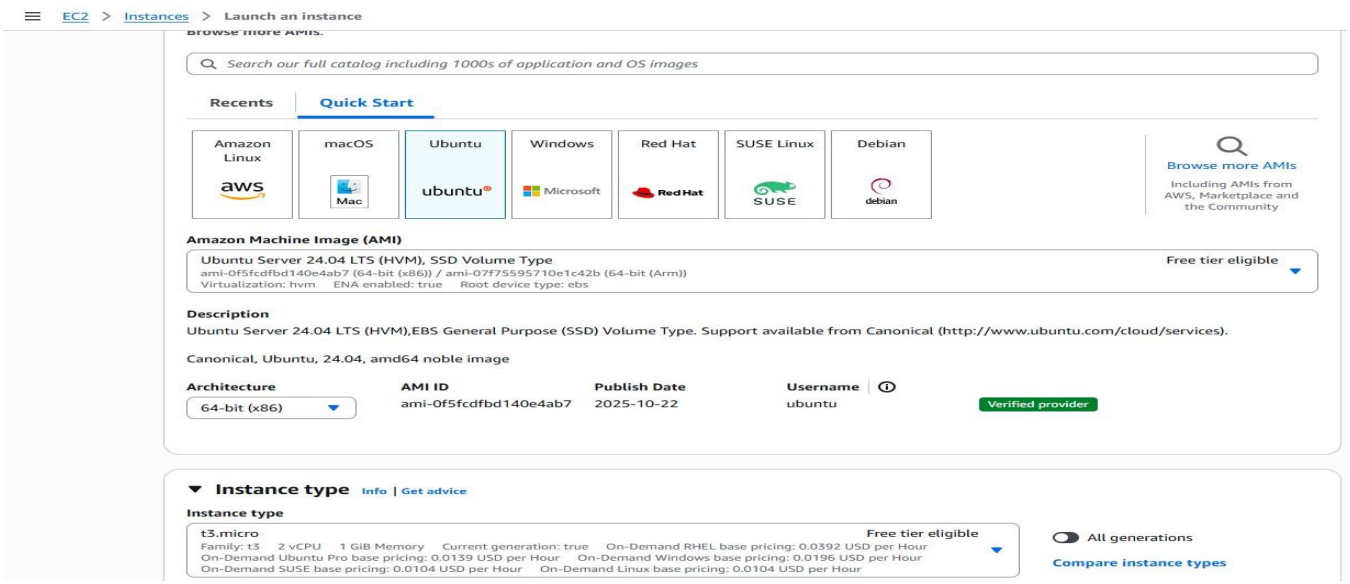


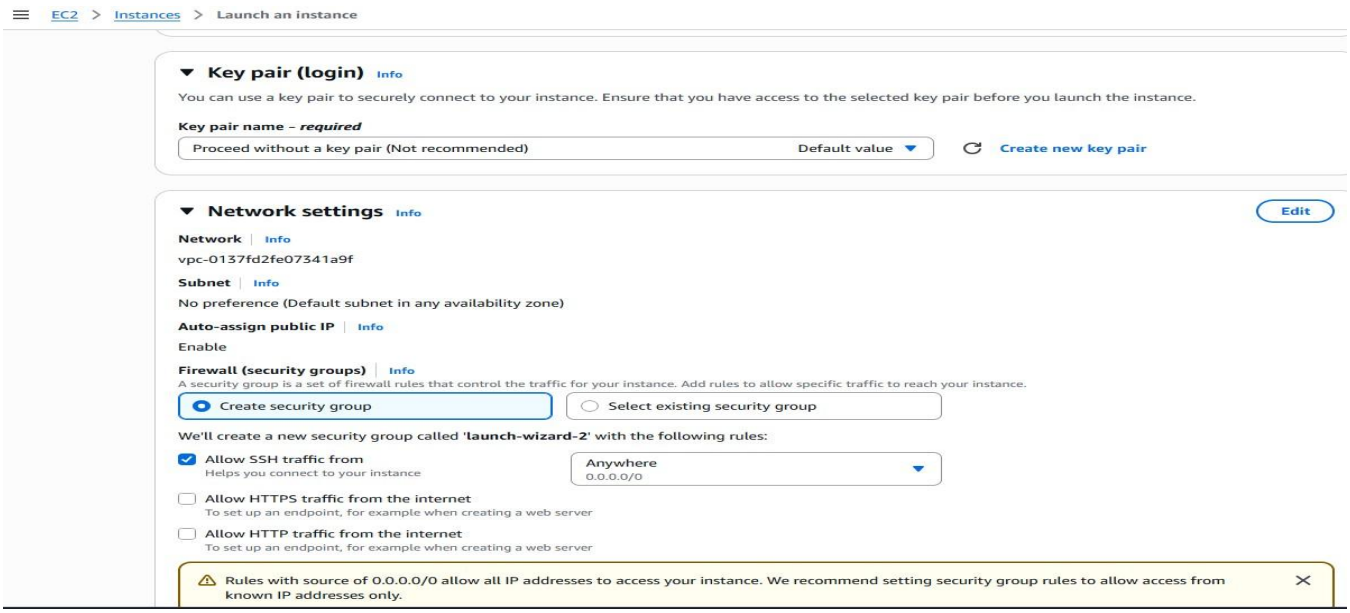
Step 10: Launch Sales Environment Instance

Repeat the instance launch process with the following modifications:

- **Name:** sales-server
- **Key pair:** Select existing **my-ec2-key** (no need to create new)
- **All other settings:** Identical to audit-server configuration



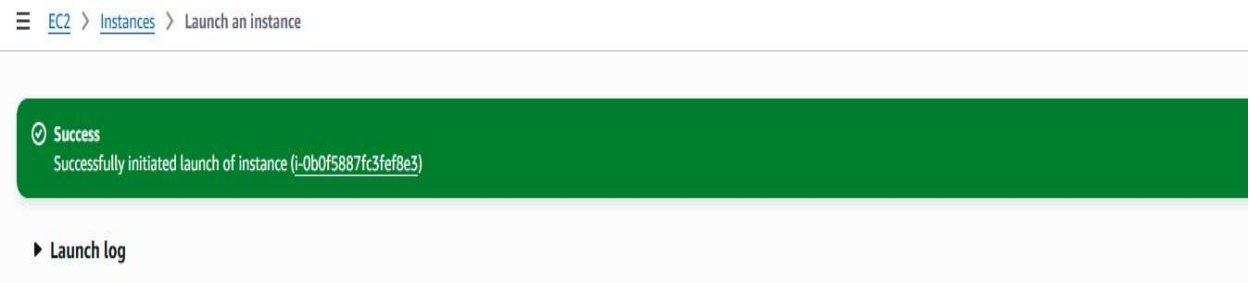


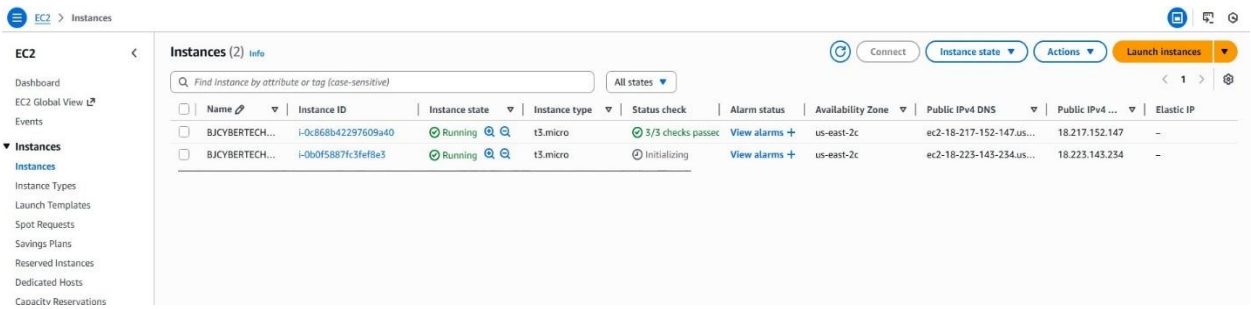


Step 11: Verify Instance Deployment

1. Navigate to **EC2 → Instances** (left sidebar)
2. Confirm both instances display **"Running"** state
3. Document instance details:

Instance Name	Instance ID	Instance State	Availability Zone	Public IPv4
audit-server	i-0abc1234...	Running	us-east-1a	XX.XX.XX.XX
sales-server	i-0xyz5678...	Running	us-east-1b	XX.XX.XX.XX





3.2 Resource Tagging Implementation

Objective: Apply environment classification tags to enable attribute-based access control (ABAC).

Tagging Strategy Rationale:
Tags function as metadata attributes attached to AWS resources. By conditionalizing IAM policies on tag values, we create flexible, scalable access boundaries without hard-coding resource identifiers (ARNs) into policies.

Step 12: Tag Audit Server

1. In EC2 Instances view, select **audit-server**
2. Navigate to **"Tags"** tab (lower panel)
3. Click **"Manage tags"**
4. Click **"Add tag"**

5. Enter tag key-value pair:
    - **Key:** Env
    - **Value:** Audit
6. Click **"Save."**

Step 13: Tag Sales Server

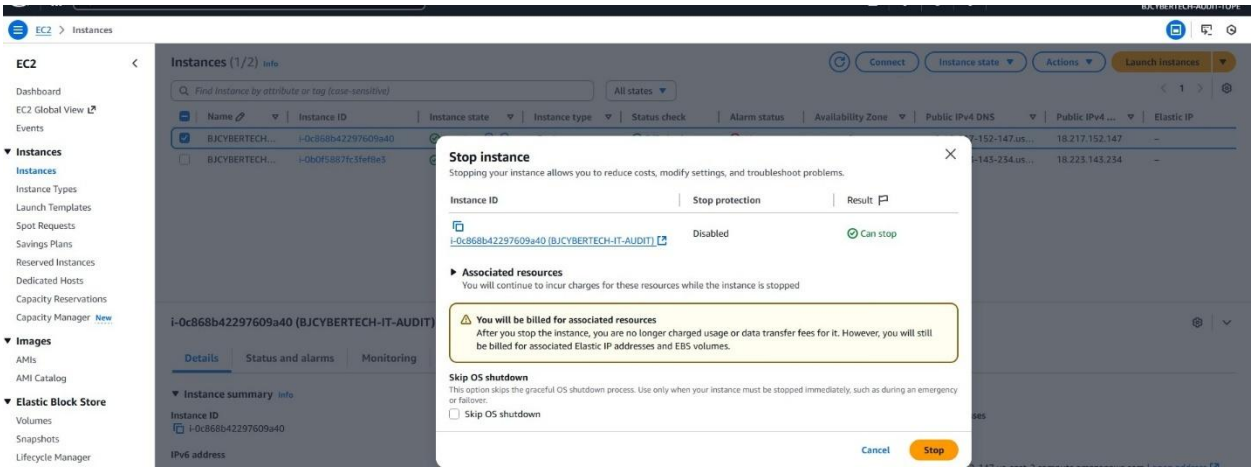
1. Select **sales-server**
2. Click the **"Tags"** tab
3. Click **"Manage tags."**
4. Click **"Add tag."**
5. Enter tag key-value pair:
    - **Key:** Env
    - **Value:** Sales
6. Click **"Save"**

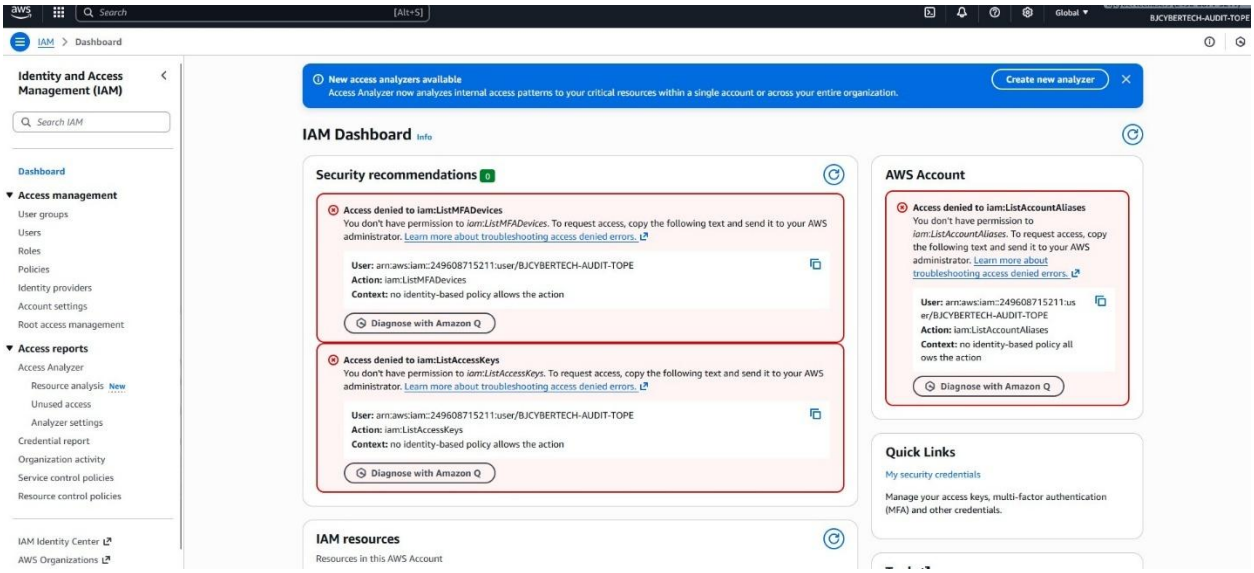
📸 **Visual Reference:** Tag management interface showing Env=Sales tag configuration

Step 14: Validate Tagging Configuration

Return to the EC2 Instances overview and verify tag visibility:

Instance Name	Instance ID	Env Tag	Accessible By Policy
audit-server	i-0abc1234...	Audit	✅ Full Management
sales-server	i-0xyz5678...	Sales	❌ Read-Only Access


EC2 > Instances
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
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
BJCYBERTECH...
i-0c868b42297609a40
18.217.152.147
BJCYBERTECH...
i-0b0f5887fc3fef8e3
18.223.143.234
Stop instance
Stopping your instance allows you to reduce costs, modify settings, and troubleshoot problems.
Instance ID
Stop protection
Result
i-0c868b42297609a40 (BJCYBERTECH-IT-AUDIT)
Disabled
Can stop
Associated resources
You will continue to incur charges for these resources while the instance is stopped
You will be billed for associated resources
After you stop the instance, you are no longer charged usage or data transfer fees for it. However, you will still be billed for associated Elastic IP addresses and EBS volumes.
Skip OS shutdown
This option skips the graceful OS shutdown process. Use only when your instance must be stopped immediately, such as during an emergency or failover.
Skip OS shutdown
Cancel
Stop
i-0c868b42297609a40 (BJCYBERTECH-IT-AUDIT)
Details
Status and alarms
Monitoring
Instance summary Info
Instance ID
i-0c868b42297609a40
IPv6 address


aws
Search
[Alt+S]
Global
BJCYBERTECH-AUDIT-TOPE
IAM > Dashboard
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management
Access reports
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
Resource control policies
IAM Identity Center
AWS Organizations
New access analyzers available
Access Analyzer now analyzes internal access patterns to your critical resources within a single account or across your entire organization.
Create new analyzer
IAM Dashboard Info
Security recommendations 0
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.
User: arn:aws:iam::249608715211:user/BJCYBERTECH-AUDIT-TOPE
Action: iam:ListMFADevices
Context: no identity-based policy allows the action
Diagnose with Amazon Q
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.
User: arn:aws:iam::249608715211:user/BJCYBERTECH-AUDIT-TOPE
Action: iam:ListAccessKeys
Context: no identity-based policy allows the action
Diagnose with Amazon Q
IAM resources
Resources in this AWS Account
AWS Account
Access denied to iam:ListAccountAliases
You don't have permission to iam:ListAccountAliases. To request access, copy the following text and send it to your AWS administrator. Learn more about troubleshooting access denied errors.
User: arn:aws:iam::249608715211:user/BJCYBERTECH-AUDIT-TOPE
Action: iam:ListAccountAliases
Context: no identity-based policy allows the action
Diagnose with Amazon Q
Quick Links
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

Part IV: Access Control Implementation

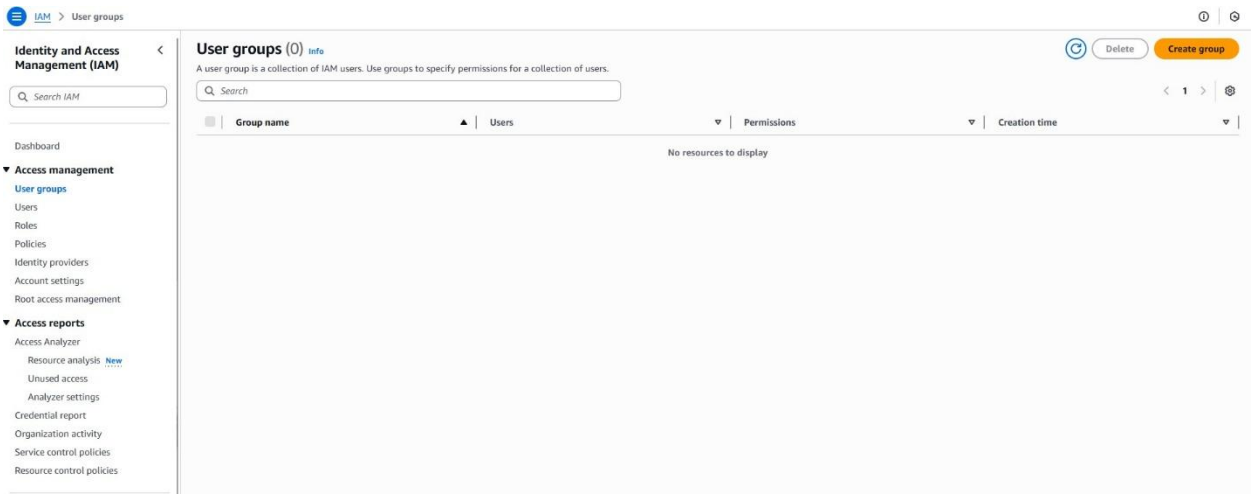
4.1 IAM User Group Architecture

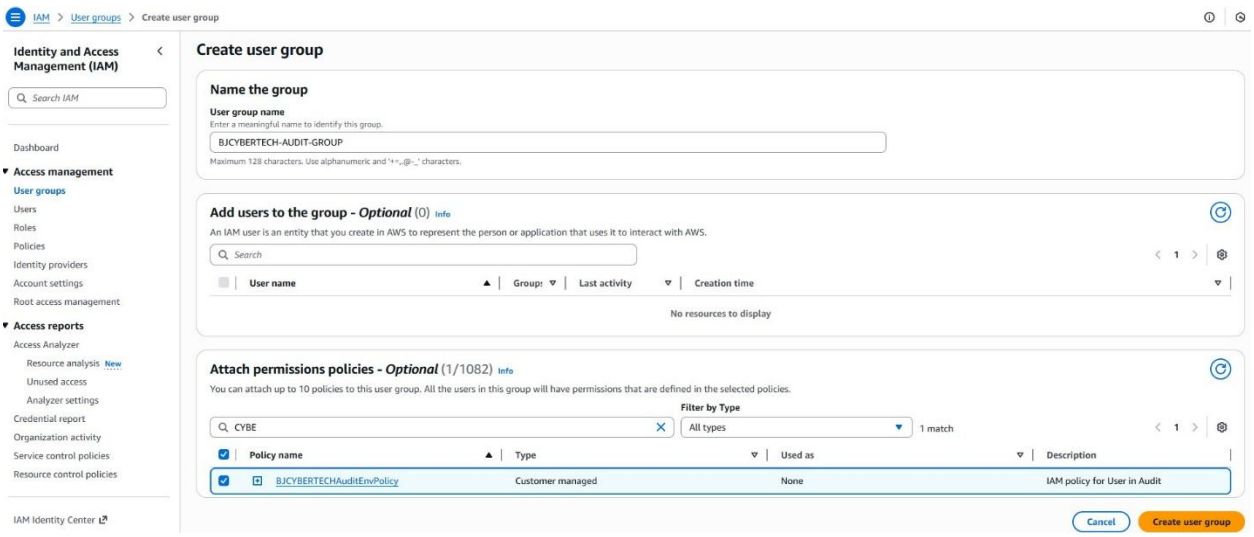
Objective: Establish a user group that will inherit the custom tag-based access policy, enabling centralized permission management.

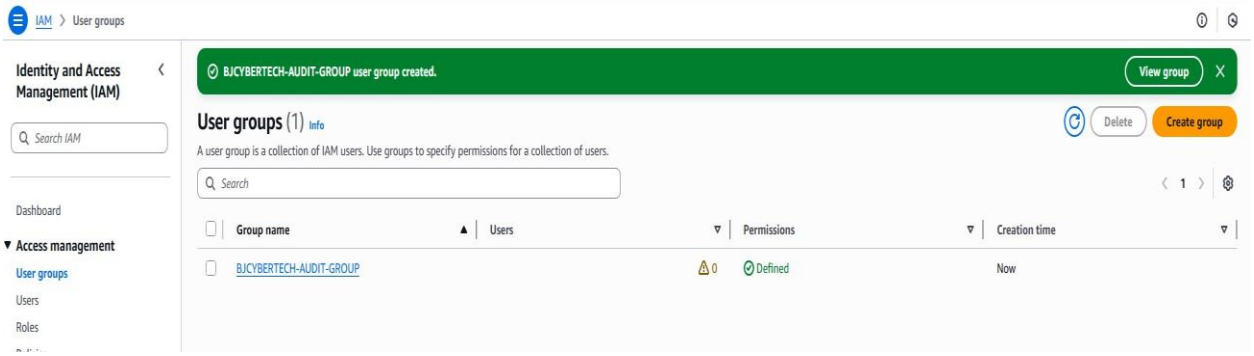
Step 15: Create User Group

1. Navigate to **IAM Dashboard → User groups**
2. Click **"Create group."**
3. **Group Configuration:**
    - **User group name:** Ltechng-Audit-Webapp
    - **Naming Convention:** Organization-Environment-Function
4. **User Assignment:** Skip for now (users will be added post-policy creation)
5. **Permissions:** Skip for now (custom policy will be attached next)
6. Click **"Create group."**

✅ **Milestone:** User group established and ready for policy attachment.







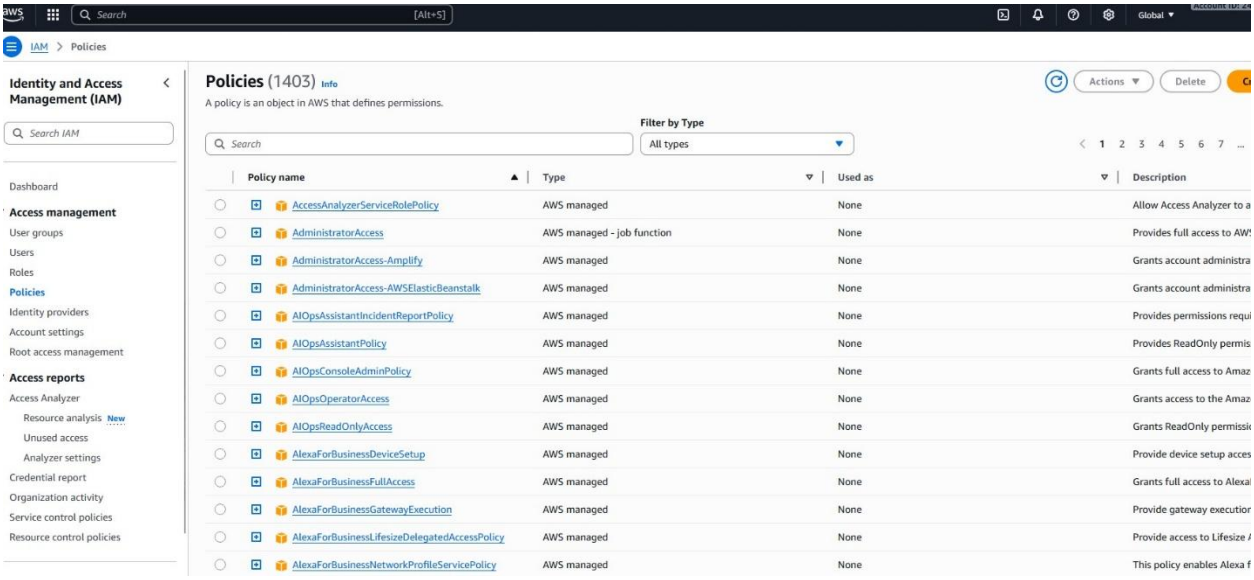
4.2 Custom IAM Policy Development

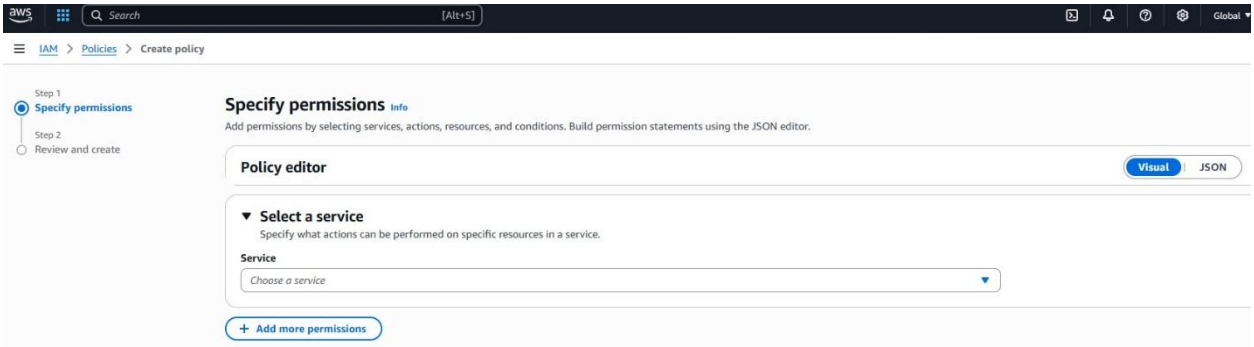
Objective: Author a JSON-formatted IAM policy implementing tag-based conditional access with tag modification protection.

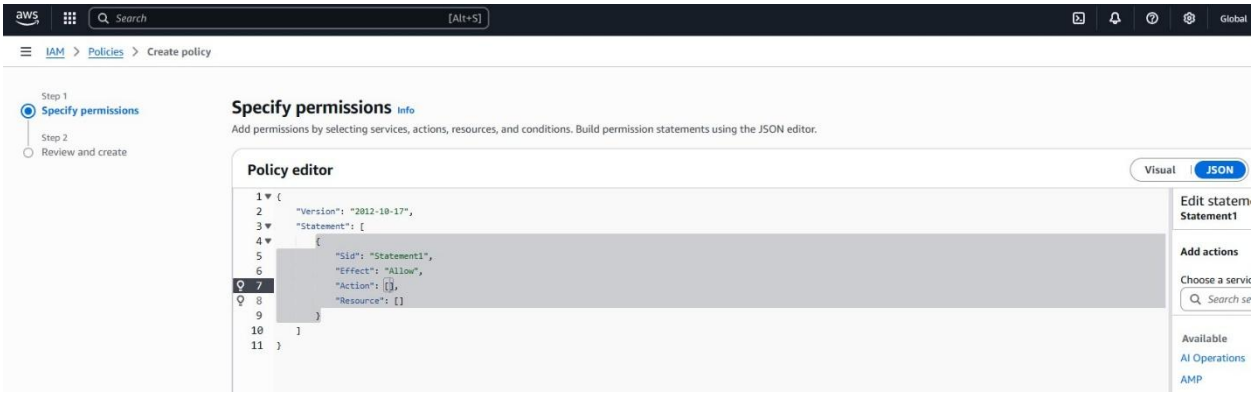
Step 16: Initiate Policy Creation

1. Navigate to **IAM Dashboard → Policies**
2. Click **"Create policy."**
3. Select **"JSON"** editor tab (vs. Visual editor)

📸 **Visual Reference:** Policy creation interface with JSON editor selected


Search
[Alt+S]
Global
IAM > 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
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
Resource control policies
Policies (1403) Info
A policy is an object in AWS that defines permissions.
Actions
Delete
Filter by Type
Search
All types
1 2 3 4 5 6 7 ...
Policy name
Type
Used as
Description
AccessAnalyzerServiceRolePolicy AWS managed None Allow Access Analyzer to a
AdministratorAccess AWS managed - job function None Provides full access to AW
AdministratorAccess-Amplify AWS managed None Grants account administra
AdministratorAccess-AWSElasticBeanstalk AWS managed None Grants account administra
AIOpsAssistantIncidentReportPolicy AWS managed None Provides permissions requi
AIOpsAssistantPolicy AWS managed None Provides ReadOnly permis
AIOpsConsoleAdminPolicy AWS managed None Grants full access to Amaz
AIOpsOperatorAccess AWS managed None Grants access to the Amaz
AIOpsReadOnlyAccess AWS managed None Grants ReadOnly permissi
AlexaForBusinessDeviceSetup AWS managed None Provide device setup acces
AlexaForBusinessFullAccess AWS managed None Grants full access to Alexa
AlexaForBusinessGatewayExecution AWS managed None Provide gateway executior
AlexaForBusinessLifesizeDelegatedAccessPolicy AWS managed None Provide access to Lifesize
AlexaForBusinessNetworkProfileServicePolicy AWS managed None This policy enables Alexa f


Search
[Alt+S]
Global
IAM > Policies > Create policy
Step 1
Specify permissions
Step 2
Review and create
Specify permissions Info
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.
Policy editor
Visual
JSON
Select a service
Specify what actions can be performed on specific resources in a service.
Service
Choose a service
+ Add more permissions


Search
[Alt+S]
Global
IAM > Policies > Create policy
Step 1
Specify permissions
Step 2
Review and create
Specify permissions Info
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.
Policy editor
Visual
JSON
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Statement1",
"Effect": "Allow",
"Action": [],
"Resource": []
}
]
}
Edit statem
Statement1
Add actions
Choose a servic
Search se
Available
AI Operations
AMP

Step 17: Define Policy Logic

Replace the default JSON with the following policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "Audit"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteTags",
        "ec2:CreateTags"
      ],
      "Resource": "*"
```

```
    }
  ]
}
```

📸 **Visual Reference:** JSON policy editor with complete policy code

4.3 Policy Analysis and Interpretation

Statement 1: Conditional EC2 Management Access

```
{
  "Effect": "Allow",
  "Action": "ec2:*",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "ec2:ResourceTag/Env": "Audit"
    }
  }
}
```

Function: Grants comprehensive EC2 permissions (ec2:*) but exclusively for resources tagged with Env=Audit.

Permitted Actions:

- ec2:StartInstances - Start stopped instances
- ec2:StopInstances - Stop running instances
- ec2:RebootInstances - Reboot instances
- ec2:TerminateInstances - Permanently delete instances
- ec2:ModifyInstanceAttribute - Change instance configuration
- And ~200+ other EC2 actions

Scope Limitation: The StringEquals condition ensures these permissions apply exclusively to resources where the Env tag equals Audit.

Statement 2: Universal Read Access

```
{
  "Effect": "Allow",
  "Action": "ec2:Describe*",
  "Resource": "*"
}
```

Function: Grants read-only access to all EC2 resources regardless of tags.

Permitted Actions:

- ec2:DescribeInstances - View instance details
- ec2:DescribeSecurityGroups - View security configurations
- ec2:DescribeVolumes - View storage information
- ec2:DescribeTags - View all resource tags
- And ~60+ other describe operations

Rationale: Users need visibility into all infrastructure to understand the environment context, even if they cannot modify non-Audit resources.

Statement 3: Tag Modification Protection (Explicit Deny)

```
{
  "Effect": "Deny",
  "Action": [
    "ec2:DeleteTags",
    "ec2:CreateTags"
  ],
  "Resource": "*"
}
```

Function: Explicitly prevents tag creation or deletion on all resources.

Security Rationale: Without this statement, users could:

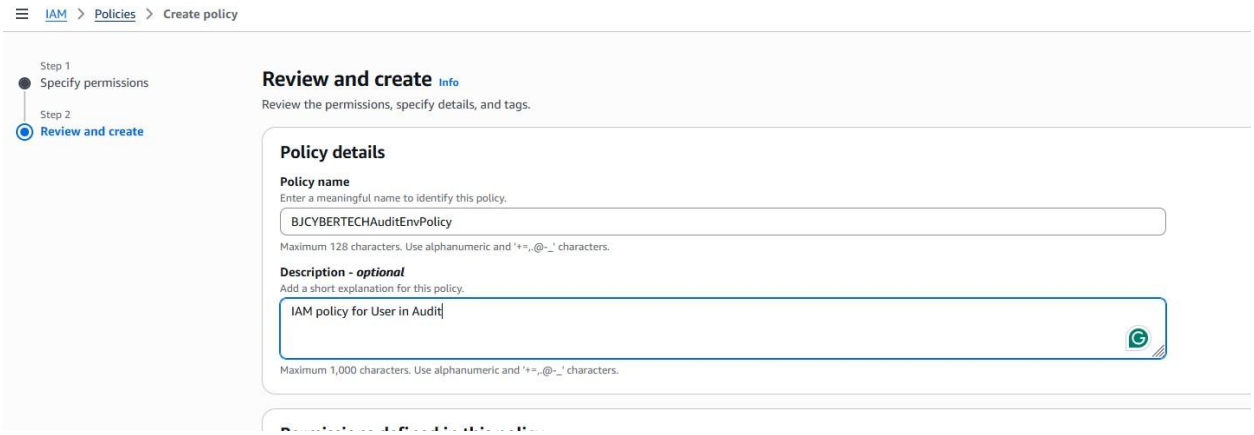
1. Create Env=Audit tags on sales-server
2. Gain full management access to sales-server
3. Bypass the intended security boundary

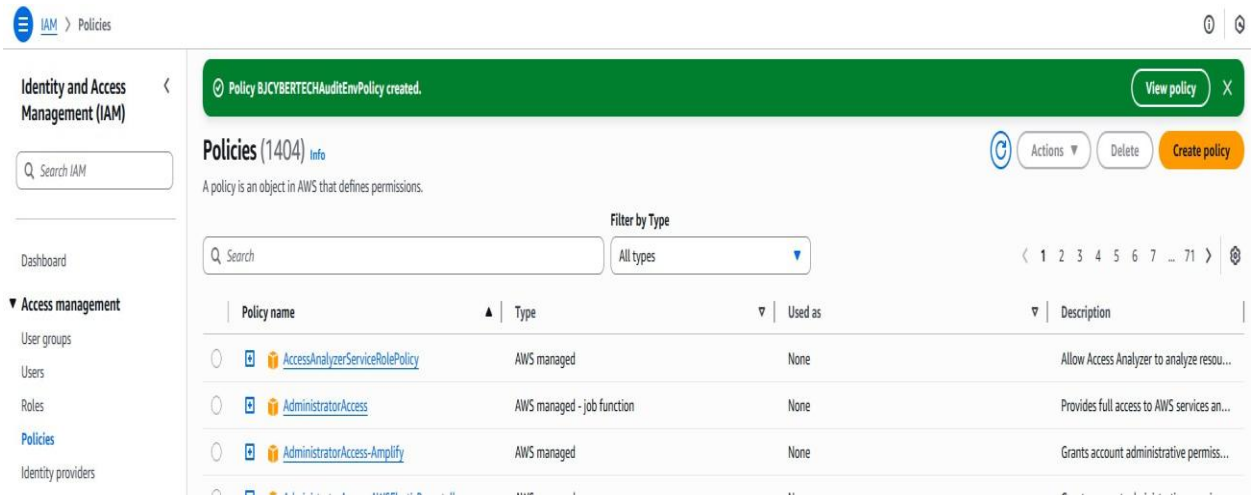
IAM Evaluation Logic: Explicit Deny statements always override Allow statements, regardless of policy order or scope. This creates unbreakable security control.

Step 18: Finalize Policy Creation

1. Click **"Next"** to proceed to review
2. **Policy Metadata:**
    - **Policy name:** AuditEnvironmentEC2Policy
    - **Description:** Grants EC2 management permissions exclusively for Audit-tagged instances with tag modification prevention
3. Review policy summary showing:
    - Service: EC2
    - Access level: Limited writing, List, Read
    - Resources: All resources (with conditions)
4. Click **"Create policy."**

✅ **Checkpoint:** Custom policy successfully created and available for attachment.


IAM > Policies > Create policy
Step 1
Specify permissions
Step 2
Review and create
Review and create Info
Review the permissions, specify details, and tags.
Policy details
Policy name
Enter a meaningful name to identify this policy.
BJCYBERTECHAuditEnvPolicy
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.
Description - optional
Add a short explanation for this policy.
IAM policy for User in Audit
Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.


IAM > Policies
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Policy BJCYBERTECHAuditEnvPolicy created.
View policy
Policies (1404) Info
A policy is an object in AWS that defines permissions.
Actions
Delete
Create policy
Filter by Type
Search
All types
1 2 3 4 5 6 7 ... 71
Policy name
Type
Used as
Description
AccessAnalyzerServiceRolePolicy
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 permiss...

4.4 Policy-Group Association

Step 19: Attach Policy to User Group

1. Navigate to **IAM → User groups**
2. Click on **Ltechng-Audit-Webapp** group name
3. Select **"Permissions"** tab
4. Click the **"Add permissions"** dropdown
5. Select **"Attach policies."**
6. Search filter: AuditEnvironmentEC2Policy
7. Check the policy checkbox
8. Click **"Attach policies."**

Result: All users added to this group will automatically inherit the tag-based access restrictions.

Part V: User Provisioning

5.1 IAM User Creation with Group Membership

Objective: Create a test user and assign it to the policy-enabled group for validation purposes.

Step 20: Create Test User

1. Navigate to **IAM → Users**
2. Click **"Create user."**
3. **User Details:**
    - **User name:** audit-webapp-user
    - **Console access:** Enable **"Provide user access to the AWS Management Console."**
    - **User type:** Select **"I want to create an IAM user."**
    - **Password Configuration:**
        - Type: **Custom password**
        - Password: [Enter secure password]
        - Password reset: Uncheck (for testing convenience)
4. Click **"Next."**

Step 21: Assign Group Membership

1. **Permission Options:** Select **"Add user to group"** (recommended approach)
2. **Group Selection:** Check **Ltechng-Audit-Webapp**
3. Click **"Next."**
4. **Review Configuration:**
    - User name: audit-webapp-user
    - Group memberships: Ltechng-Audit-Webapp
    - Permissions: AuditEnvironmentEC2Policy (inherited from group)
5. Click **"Create user."**

Step 22: Document User Credentials

Success Screen Information:

- **Console sign-in URL:** https://[your-alias].signin.aws.amazon.com/console
- **Username:** audit-webapp-user
- **Password:** [As configured]

Secure Storage: Document these credentials in a password manager or secure location.

📸 **Visual Reference:** User creation success page with sign-in details

5.2 Permission Assignment Methodology Comparison

Method	Use Case	Management Overhead	Scalability	Recommendation
Add user to group	Standard operational use	Low - modify group policy affects all members	High - easily add/remove users	✅ **Preferred**
Copy permissions	Replicating exact permissions from existing user	Medium - must update each user individually	Medium - useful for similar roles	Situational
Attach policies directly	Emergency temporary access, unique requirements	High - each user managed separately	Low - doesn't scale	❌ **Avoid**

Best Practice: Manage permissions at the group level. Individual policy attachments should be reserved for exceptional circumstances requiring unique access patterns.

Part VI: Audit and Compliance Infrastructure

6.1 CloudTrail Configuration

Objective: Implement comprehensive logging of all API calls and management console actions for security monitoring, compliance, and forensic analysis.

AWS CloudTrail Overview

Service Function:
CloudTrail provides governance, compliance, operational auditing, and risk auditing of your AWS account by logging, continuously monitoring, and retaining account activity.

Audit Capabilities:

- **Identity Tracking:** Records which IAM user, role, or service performed actions
- **Action Logging:** Captures API calls, console actions, and automated service operations
- **Temporal Context:** Timestamps with millisecond precision
- **Source Attribution:** Logs originating IP addresses and user agents
- **Resource Association:** Links actions to specific AWS resources affected

Storage and Analysis:

- Primary storage: Amazon S3 buckets (scalable, durable)
- Optional integration: CloudWatch Logs (real-time monitoring)
- Optional integration: EventBridge (automated response triggers)

Step 23: Initiate Trail Creation

1. Search for **"CloudTrail"** in the AWS Console
2. Navigate to the CloudTrail service dashboard
3. Click **"Create trail"**

Step 24: Configure Trail Parameters

General Details:

- **Trail name:** my-security-audit-trail
- **Trail log bucket and folder:**
    - Selection: **"Create new S3 bucket."**
    - Bucket name: Auto-generated (format: aws-cloudtrail-logs-[account-id]-[random-string])
    - Bucket region: Same as trail region
- **Log file SSE-KMS encryption:** Disabled (for simplicity; enable in production)
- **Log file validation: Enabled** ✅
    - Function: Creates digital signature for each log file
    - Purpose: Detects tampering or unauthorized modifications
    - Compliance: Required for many regulatory frameworks
- **SNS notification delivery:** Disabled (optional feature)

Click **"Next"**

Step 25: Select Event Types

Management Events:

- **Read events:** Enabled ✅ (e.g., DescribeInstances, ListUsers)
- **Write events:** Enabled ✅ (e.g., StopInstances, CreateUser)
- **Exclude AWS KMS events:** Enabled (reduces log volume)
- **Exclude Amazon RDS Data API events:** Enabled

Data Events: Disabled (optional - logs S3 object-level operations and Lambda invocations)

Insights Events: Disabled (optional - ML-based unusual activity detection)

Click **"Next"**

📸 **Visual Reference:** Event selection interface with management events configured

Step 26: Review and Create Trail

1. Review all configuration parameters
2. Verify the S3 bucket will be created automatically
3. Click **"Create trail."**

Expected Result: Trail status shows **"Logging"** with a green indicator

6.2 CloudTrail Log Structure and Analysis

Sample CloudTrail Event Record


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAEXAMPLE123456",
    "arn": "arn:aws:iam::123456789012:user/audit-webapp-user",
    "accountId": "123456789012",
    "userName": "audit-webapp-user"
  },
  "eventTime": "2025-11-05T14:32:18Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "StopInstances",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.42",
  "userAgent": "console.ec2.amazonaws.com",
  "requestParameters": {
    "instancesSet": {
      "items": [{"instanceId": "i-0abc1234efgh5678"}]
    }
  },
  "responseElements": {
    "requestId": "example-request-id",
    "instancesSet": {
      "items": [{
        "instanceId": "i-0abc1234efgh5678",
```

```
      "currentState": {"code": 64, "name": "stopping"}
    }]
  }
 },
 "errorCode": "Client.UnauthorizedOperation",
 "errorMessage": "You are not authorized to perform this operation."
}
```


Key Fields for Security Analysis:

- **userIdentity:** Identifies who performed the action
- **eventName:** Specific API call or action
- **errorCode:** Indicates authorization failures (security events)
- **sourceIPAddress:** Geographic and network context
- **requestParameters:** Details of what was requested
- **responseElements:** Outcome of the request

Part VII: Security Validation and Testing

7.1 Policy Effectiveness Testing

Objective: Systematically verify that the IAM policy correctly permits actions on Audit-tagged instances while denying actions on Sales-tagged instances.

Step 27: Authenticate as Test User

1. Sign out of IAM admin user session
2. Navigate to IAM sign-in URL: https://[your-alias].signin.aws.amazon.com/console
3. Enter credentials:
    - **IAM user name:** audit-webapp-user
    - **Password:** [As configured]
4. Click **"Sign in"**

Step 28: Verify Read Access

1. Search for and navigate to the **EC2 service**
2. Click **"Instances (running)"** in the left sidebar
3. **Expected Result:** Both instances (audit-server and sales-server) are visible

Analysis: Statement 2 of the policy grants ec2:Describe* permissions on all resources, enabling visibility.

7.2 Positive Test Case: Audit Server Management

Hypothesis: User can successfully stop the audit-server instance due to Env=Audit tag.

Step 29: Execute Permitted Action

1. Select the **audit-server** checkbox
2. Click the **"Instance state"** dropdown menu
3. Select **"Stop instance."**
4. Confirm the action in the dialog box

Expected Result: ✅ **Success**

- Instance state transitions: Running → Stopping → Stopped
- No error messages displayed
- Action completes within 30-60 seconds

📸 **Visual Reference:** Successfully stopped audit-server with state showing "Stopped."

Policy Evaluation:

Statement 1: ALLOW (ec2:StopInstances on resources with Env=Audit)

Statement 2: ALLOW (ec2:Describe* on all resources)

Statement 3: N/A (tag modification not attempted)

Final Result: ALLOW

7.3 Negative Test Case: Sales Server Management

Hypothesis: The User cannot stop the sales-server instance due to the Env=Sales tag mismatch.

Step 30: Execute Denied Action

1. Select the **sales-server** checkbox
2. Click the **"Instance state"** dropdown menu
3. Select **"Stop instance."**

Expected Result: ❌ Failure

Error Message:

Failed to stop the instance i-0xyz5678...

You are not authorized to perform this operation.

Encoded authorization failure message: [truncated]

Policy Evaluation:

Statement 1: NO MATCH (resource has Env=Sales, not Env=Audit)

Statement 2: ALLOW (ec2:Describe* only - does not include ec2:StopInstances)

Statement 3: N/A (tag modification not attempted)

Final Result: IMPLICIT DENY (no explicit allow = deny by default)

Analysis: The conditional access in Statement 1 does not match the sales-server tag, and Statement 2 only grants describe permissions. Result: Access denied.

7.4 Tag Modification Protection Test

Hypothesis: User cannot modify tags on any instance due to explicit deny in Statement 3.

Step 31: Attempt Tag Modification

1. Select either instance (audit-server or sales-server)
2. Navigate to the **"Tags"** tab
3. Click **"Manage tags."**

Expected Result: ❌ Failure

Observed Behavior:

- Either the "Manage tags" button is grayed out/disabled
- Or clicking produces an authorization error:
- You are not authorized to perform this operation. Required permissions: ec2:CreateTags

Policy Evaluation:

Statement 1: Would normally ALLOW ec2:CreateTags for Env=Audit resources

Statement 2: N/A (does not include tag operations)

Statement 3: EXPLICIT DENY on ec2:CreateTags and ec2:DeleteTags

Final Result: DENY (explicit deny always wins)

Security Validation: ✅ Users cannot circumvent policy by adding Env=Audit tags to non-Audit resources.

7.5 CloudTrail Verification

Step 32: Audit Log Analysis

1. Sign back in as **IAM admin user**
2. Navigate to **CloudTrail → Event history**
3. Apply filters:
    - **User name:** audit-webapp-user
    - **Time range:** Last hour
4. Locate relevant events:

Successful Action (audit-server stop):

Event name: StopInstances

Event source: ec2.amazonaws.com

User name: audit-webapp-user

Resource type: AWS::EC2::Instance

Resource name: i-0abc1234efgh5678

Event time: 2025-11-05 14:32:18 UTC

Error code: (none)

Failed Action (sales-server stop):

Event name: StopInstances

Event source: ec2.amazonaws.com

User name: audit-webapp-user

Resource type: AWS::EC2::Instance

Resource name: i-0xyz5678...

Event time: 2025-11-05 14:35:42 UTC

Error code: Client.UnauthorizedOperation

Error message: You are not authorized to perform this operation.

Forensic Value: CloudTrail provides a complete audit trail for:

- Compliance reporting
- Security incident investigation
- User behavior analysis
- Policy effectiveness validation

Part VIII: Project Outcomes and Best Practices

8.1 Implementation Summary

Achievements

✅ **Least-Privilege Access Control**
Implemented granular permissions enabling users to manage only designated resources while maintaining visibility across the entire infrastructure.

✅ **Tag-Based Resource Boundaries**
Leveraged AWS tags as policy conditions, creating flexible and maintainable access controls that scale with infrastructure growth.

✅ **Privilege Escalation Prevention**
Explicit deny rules prevent users from modifying resource tags to bypass security boundaries.

✅ **Comprehensive Audit Trail**
CloudTrail logging captures all user actions, providing forensic evidence for security investigations and compliance reporting.

✅ **Defense in Depth**
Multiple security layers: authentication (IAM users), authorization (policies), resource boundaries (tags), and monitoring (CloudTrail).

8.2 Technical Competencies Demonstrated

Domain	Skills Validated
Cloud Security	IAM policy design, least-privilege implementation, ABAC (attribute-based access control)
Policy Engineering	JSON syntax, conditional logic, effect precedence (allow vs deny), resource ARN patterns
Identity Management	User/group architecture, permission inheritance, and authentication methods
Compliance & Audit	Activity logging, event analysis, forensic investigation, and evidence preservation
Infrastructure as Code	Declarative security policies, reproducible configurations

8.3 Enterprise Security Best Practices

1. Root Account Protection

- ✅ Minimize root account usage (account-level changes only)
- ✅ Enable MFA on root account
- ✅ Never create access keys for root account
- ✅ Use IAM users for all operational activities

2. Group-Based Permission Management

- ✅ Assign permissions to groups, not individual users
- ✅ Add users to appropriate groups based on