CLOUD SECURITY PROJECT (AWS)

Implementing IAM Least Privilege for EC2 Using Tags

Created by: Emmanuel Oladeinde
Role: Cyber Security Analyst
Date: November 2025

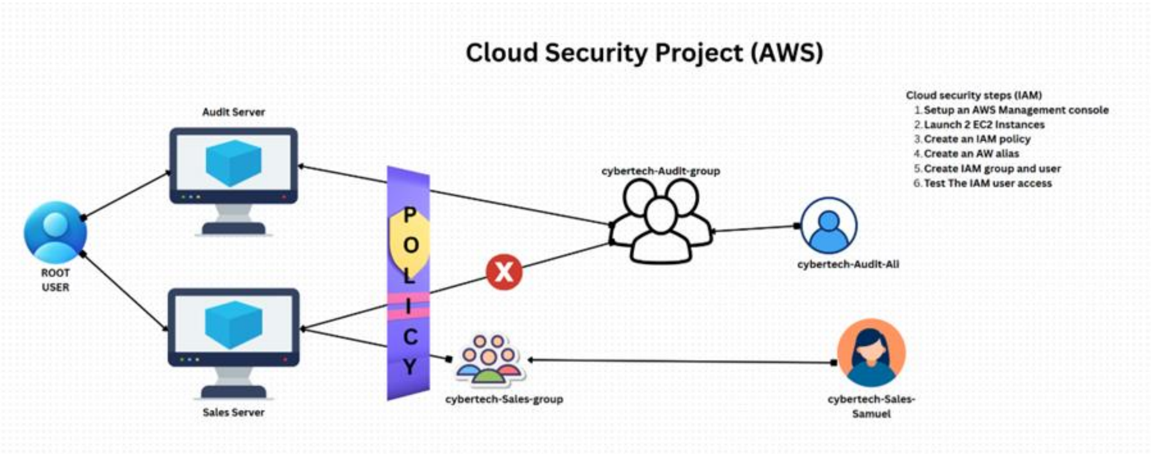
1. Project Overview

This project focuses on implementing **least privilege access control** for Amazon EC2 instances using **AWS Identity and Access Management (IAM)** and **resource tags**.

The scenario:

- The organization runs multiple EC2 instances.
- Two of them are key:
  - `audit` – a critical instance that must not be stopped or started by standard users.
  - `sales` – a business application instance that standard users are allowed to start and stop.
- We want IAM users to:
  - **Start/stop only the `sales` instance.**
  - **Be blocked from starting/stopping the `audit` instance.**

This is achieved by combining **IAM policies** with **EC2 tags**, rather than giving broad EC2 permissions or hard-coding instance IDs.



Cloud Security Project (AWS)

Cloud security steps (IAM)
1. Setup an AWS Management console
2. Launch 2 EC2 Instances
3. Create an IAM policy
4. Create an AW alias
5. Create IAM group and user
6. Test The IAM user access

2. Objectives

The goals of this project were to:

1. Demonstrate how to enforce **least privilege** for EC2 operations using IAM.
2. Restrict EC2 lifecycle actions (start/stop) based on **tags** instead of granting blanket permissions.
3. Separate **sensitive** and **non-sensitive** workloads (`audit` vs `sales`) at the access-control level.
4. Validate the configuration by testing **expected vs actual behavior** using an IAM user account.

3. AWS Services & Tools Used

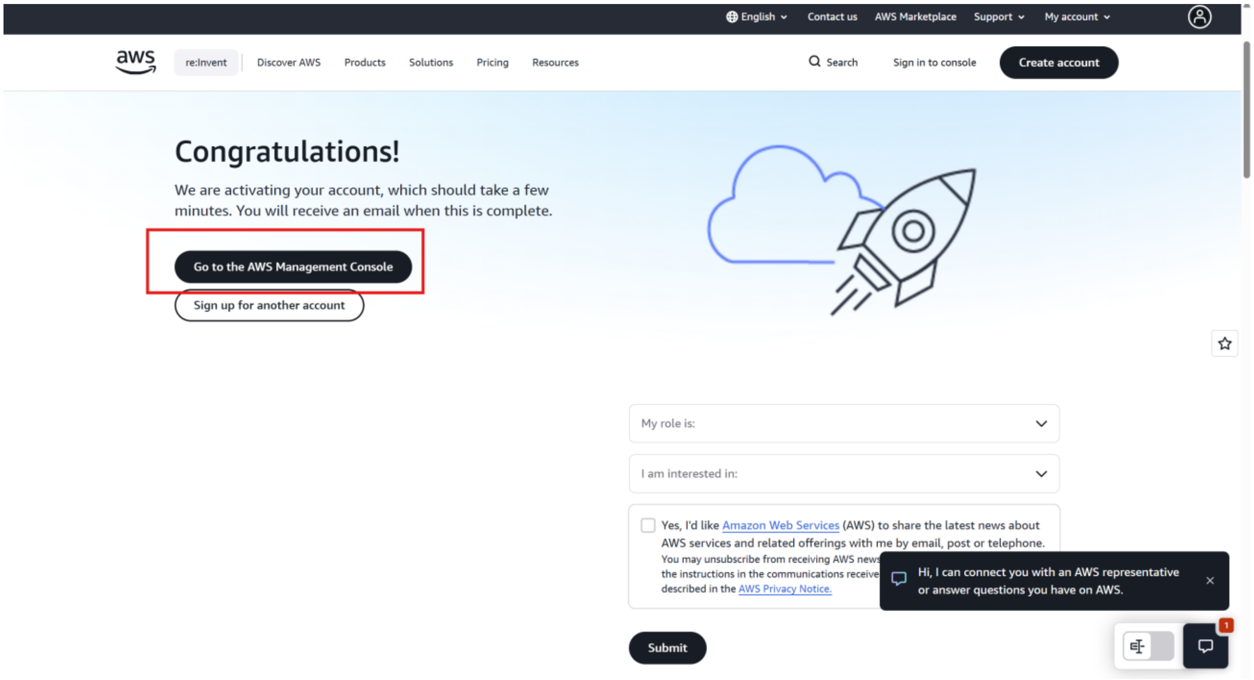
- **AWS Management Console**
  For managing IAM, EC2, and testing user access.
- **AWS Identity and Access Management (IAM)**
  - Users
  - Groups
  - Customer-managed policies
  - Account alias (friendly sign-in URL)
- **Amazon EC2 (Elastic Compute Cloud)**
  - Launching Linux instances
  - Adding and managing tags

  - Starting and stopping instances
- **IAM Policy Language (JSON)**
  - `Effect`, `Action`, `Resource`, `Condition` blocks
  - Tag-based conditions using `ec2:ResourceTag/*`

4. Environment Setup

1. **AWS Account**
   a. Logged into an AWS account via the **Management Console**.
   b. Confirmed that the root user is reserved only for high-level administration and not used for daily tasks.
2. **IAM Account Alias**
   a. Navigated to **IAM → Dashboard**.
   b. Created a **custom account alias** to replace the long numeric sign-in URL with a human-readable URL for IAM users.

This setup allows IAM users to log in with a simple URL while keeping the root account minimally used, which is a best practice in cloud security.



Congratulations!

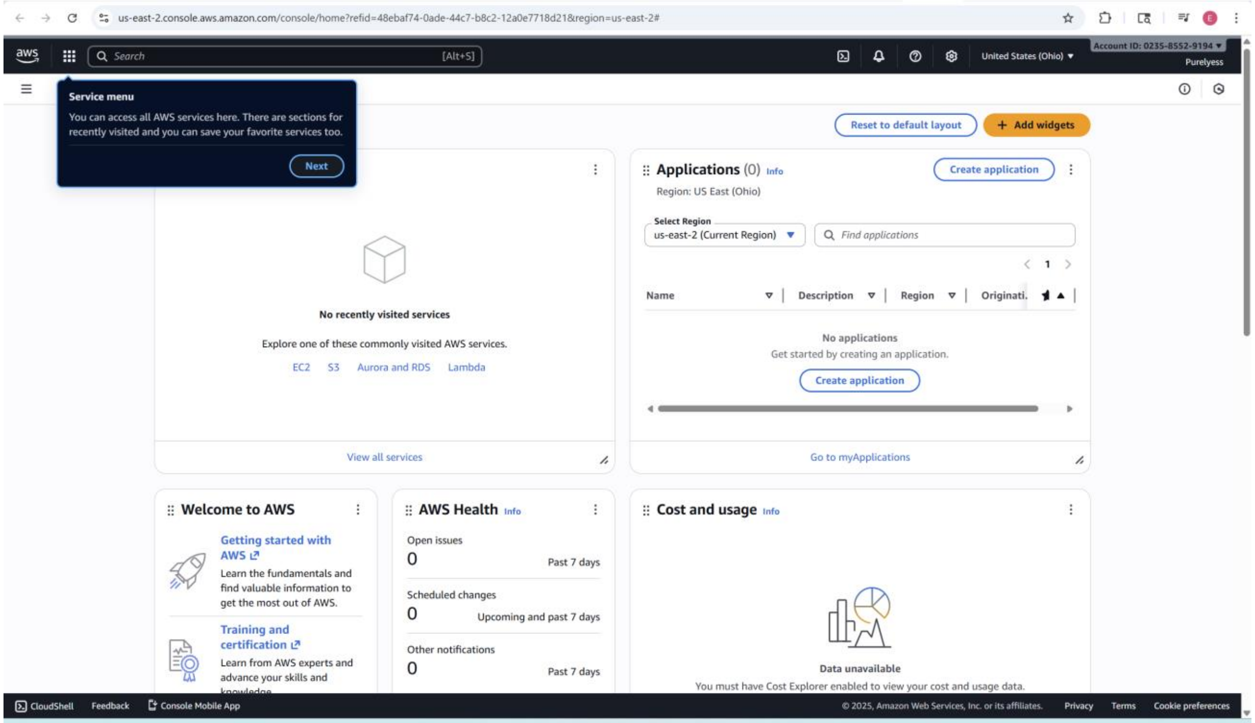
We are activating your account, which should take a few minutes. You will receive an email when this is complete.

My role is:

I am interested in:

□ Yes, I'd like Amazon Web Services (AWS) to share the latest news about AWS services and related offerings with me by email, post or telephone. You may unsubscribe from receiving AWS news the instructions in the communications receive described in the AWS Privacy Notice.

Hi, I can connect you with an AWS representative or answer questions you have on AWS.



Service menu

You can access all AWS services here. There are sections for recently visited and you can save your favorite services too.

Region: US East (Ohio)

No recently visited services

Explore one of these commonly visited AWS services.

No applications

Get started by creating an application.

Learn the fundamentals and find valuable information to get the most out of AWS.

Learn from AWS experts and advance your skills and knowledge

Open issues
0 Past 7 days

Scheduled changes
0 Upcoming and past 7 days

Other notifications
0 Past 7 days

Data unavailable

You must have Cost Explorer enabled to view your cost and usage data.

© 2025, Amazon Web Services, Inc. or its affiliates.

5. EC2 Instance Configuration

Two EC2 instances were launched to represent different environments:

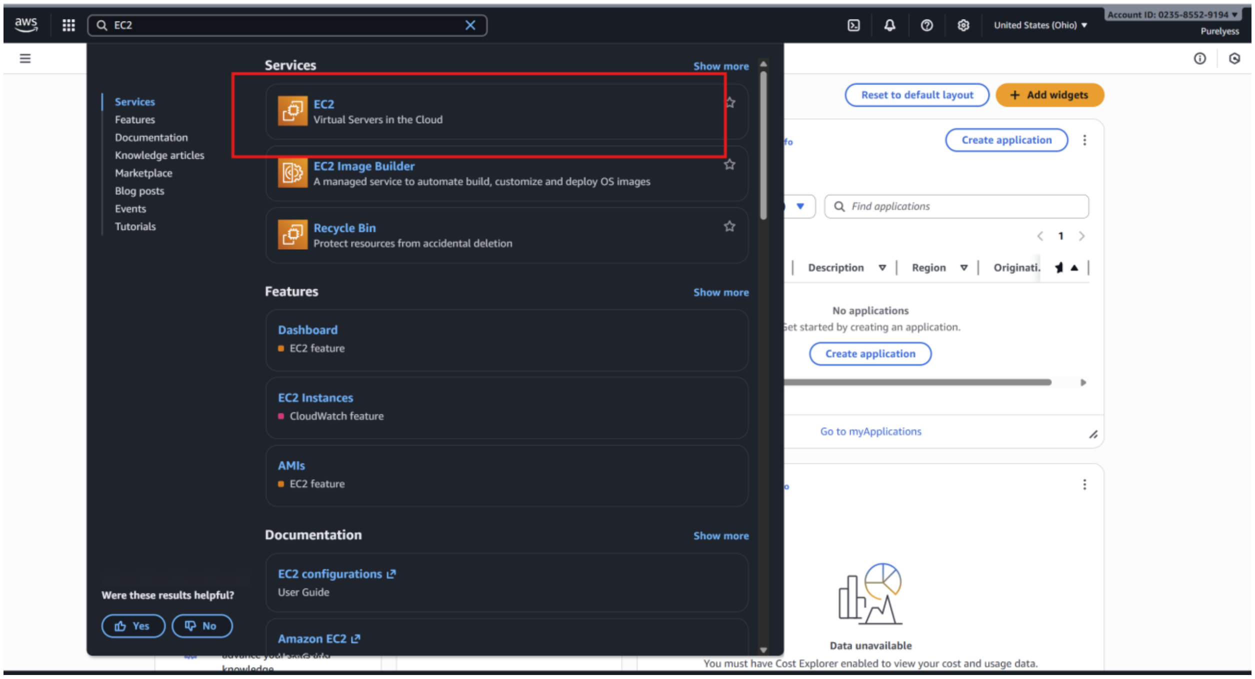
1. **audit instance**
   a. Purpose: Sensitive auditing/logging system.
   b. Configuration:
      i. Launched via **EC2 → Instances → Launch instance**.
      ii. Free-tier eligible AMI (e.g., Amazon Linux 2).
      iii. Instance tagged with:
         1. `Key = Environment`
         2. `Value = Audit`
2. **sales instance**
   a. Purpose: Sales or application server that standard users may manage.
   b. Configuration:
      i. Launched in the same region as `audit`.
      ii. Tagged with:
         1. `Key = Environment`
         2. `Value = Sales`

Tagging Strategy

The same tag key is used for both instances, but the values are different:

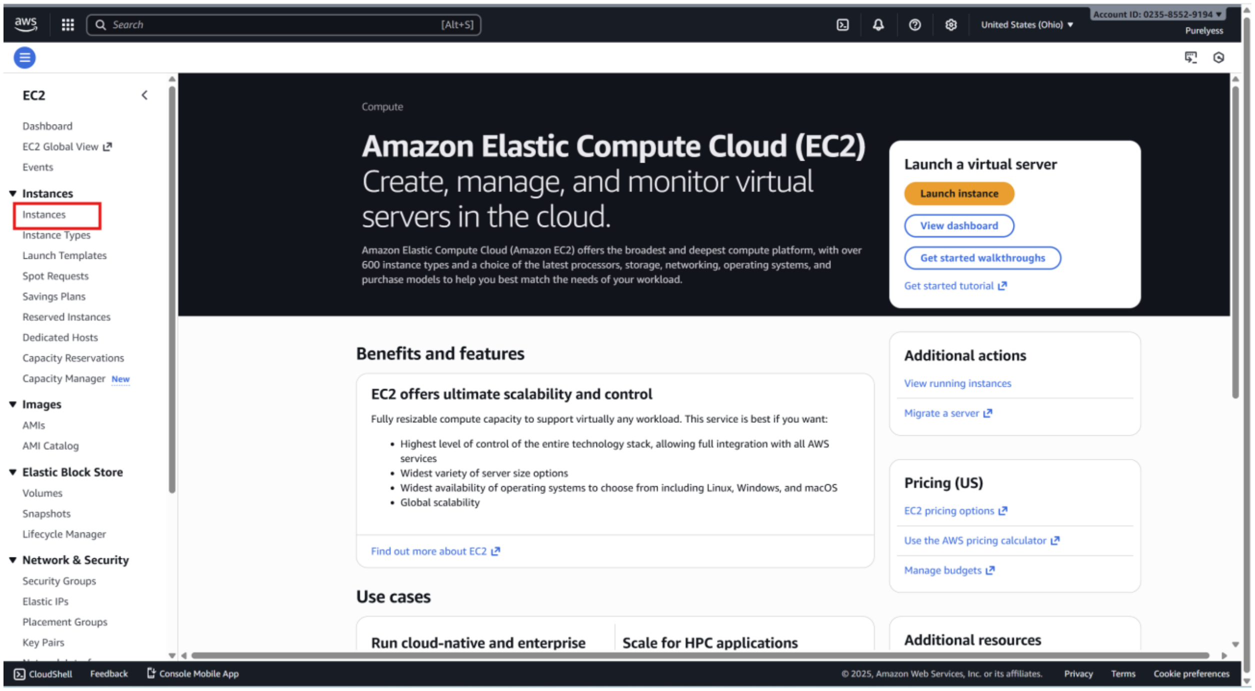
Instance	Tag Key	Tag Value
audit	Environment	Audit
sales	Environment	Sales

These tags are the basis for the IAM policy’s conditional logic.



No applications
Get started by creating an application.

You must have Cost Explorer enabled to view your cost and usage data.



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

EC2 offers ultimate scalability and control

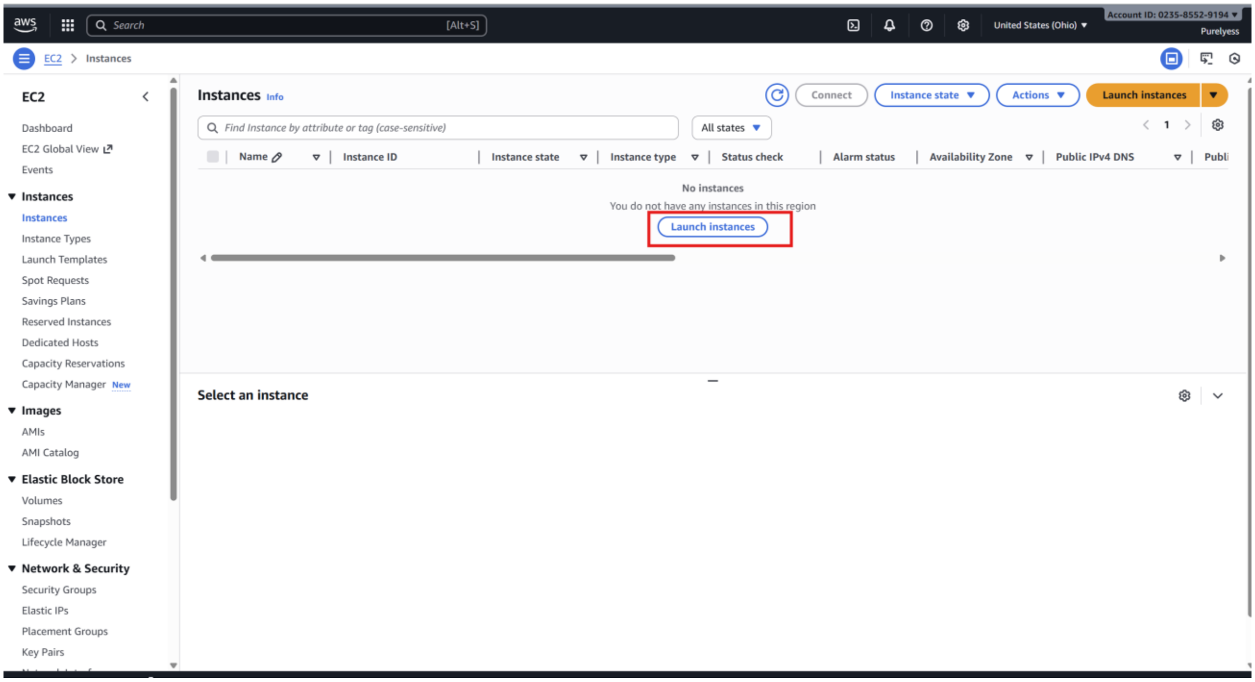
Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Widest variety of server size options
- Widest availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

Use cases

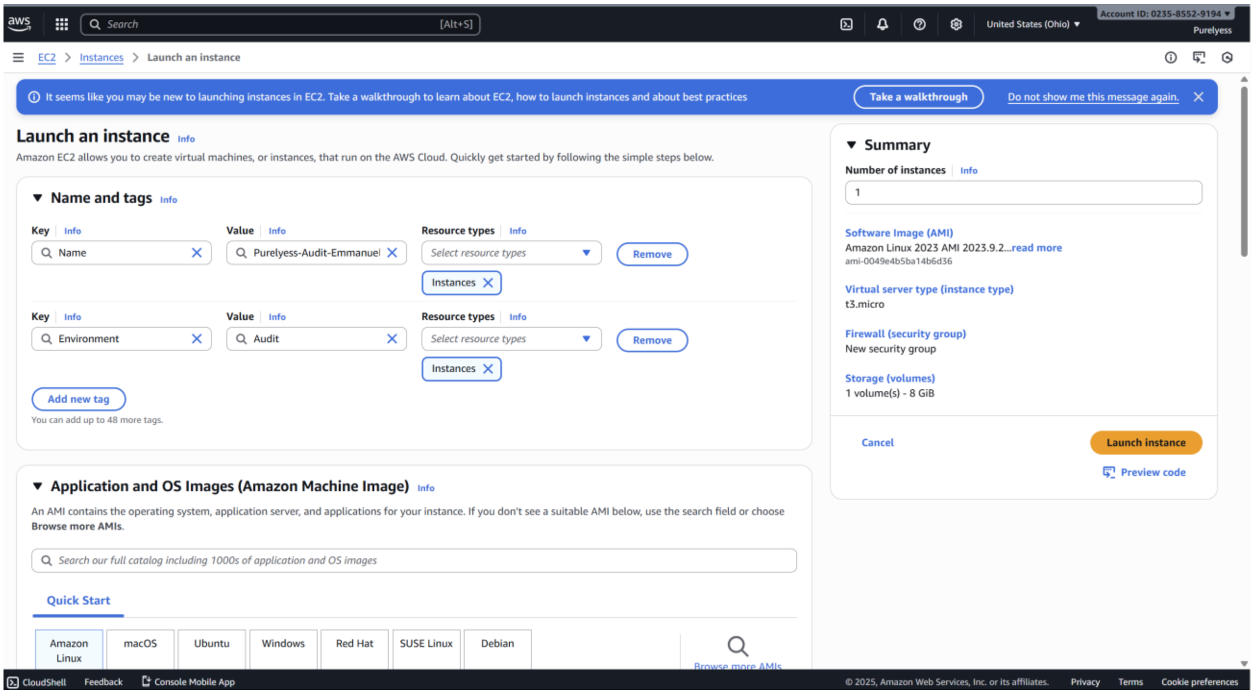
Run cloud-native and enterprise

Scale for HPC applications



No instances

You do not have any instances in this region



aws
Search [Alt+S]
United States (Ohio) ▾
Account ID: 0235-8552-9194 ▾
Purelyess
EC2 > Instances > Launch an instance

ⓘ 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

Key	Value	Resource types	
Name	Purelyess-Audit-Emmanuel	Select resource types Instances	Remove
Environment	Audit	Select resource types Instances	Remove

Add new tag

You can add up to 48 more tags.

▼ 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**.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs

▼ Summary

Number of instances Info

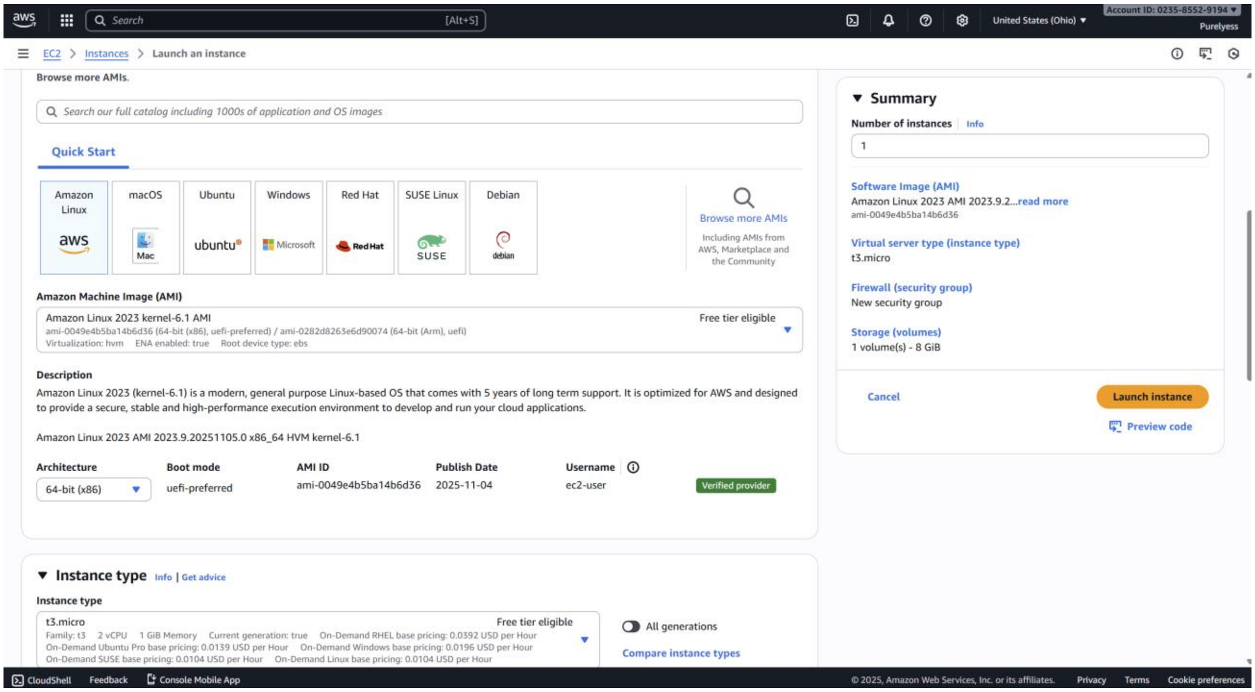
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.9.2...read more
ami-0049e4b5ba14b6d36

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

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



aws
Search [Alt+S]
United States (Ohio) ▾
Account ID: 0235-8552-9194 ▾
Purelyess
EC2 > Instances > Launch an instance

Browse more AMIs.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI — Free tier eligible
ami-0049e4b5ba14b6d36 (64-bit (x86), uefi-preferred) / ami-0282d8263e6d90074 (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 2023.9.20251105.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username ⓘ	
64-bit (x86)	uefi-preferred	ami-0049e4b5ba14b6d36	2025-11-04	ec2-user	Verified provider

▼ Instance type Info | Get advice

Instance type

t3.micro — Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour

▼ Summary

Number of instances Info

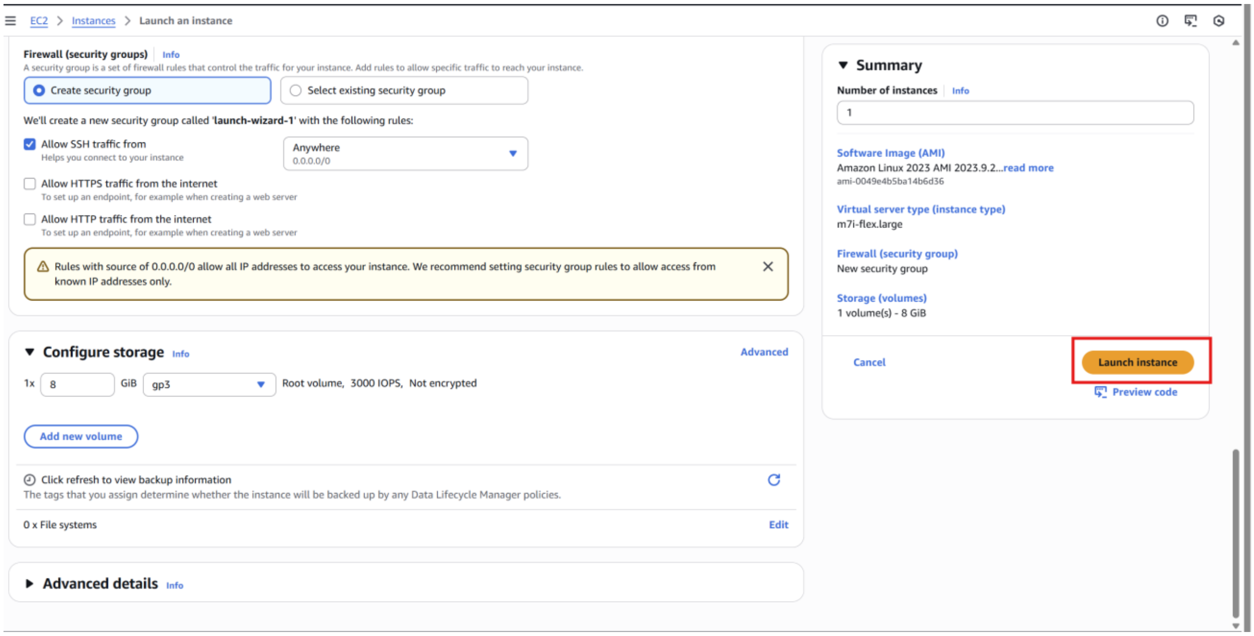
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.9.2...read more
ami-0049e4b5ba14b6d36

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

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



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 — Anywhere 0.0.0.0/0
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

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info — Advanced

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

Advanced details Info

Summary

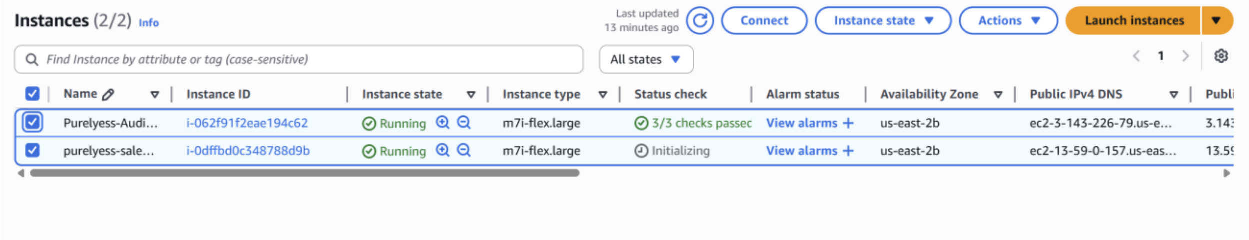
Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.9.2...read more
ami-0049e4b5ba14b6d36

Virtual server type (instance type)
m7i-flex.large

Firewall (security group)
New security group

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



Instances (2/2) Info

Last updated 13 minutes ago
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find Instance by attribute or tag (case-sensitive)
All states ▼
1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Publi
☑	Purelyess-Audi...	i-062f91f2eae194c62	Running	m7i-flex.large	3/3 checks passec	View alarms +	us-east-2b	ec2-3-143-226-79.us-e...	3.143
☑	purelyess-sale...	i-0dffbd0c348788d9b	Running	m7i-flex.large	Initializing	View alarms +	us-east-2b	ec2-13-59-0-157.us-eas...	13.59

6. IAM Policy Design

The core of this project is the IAM policy that:

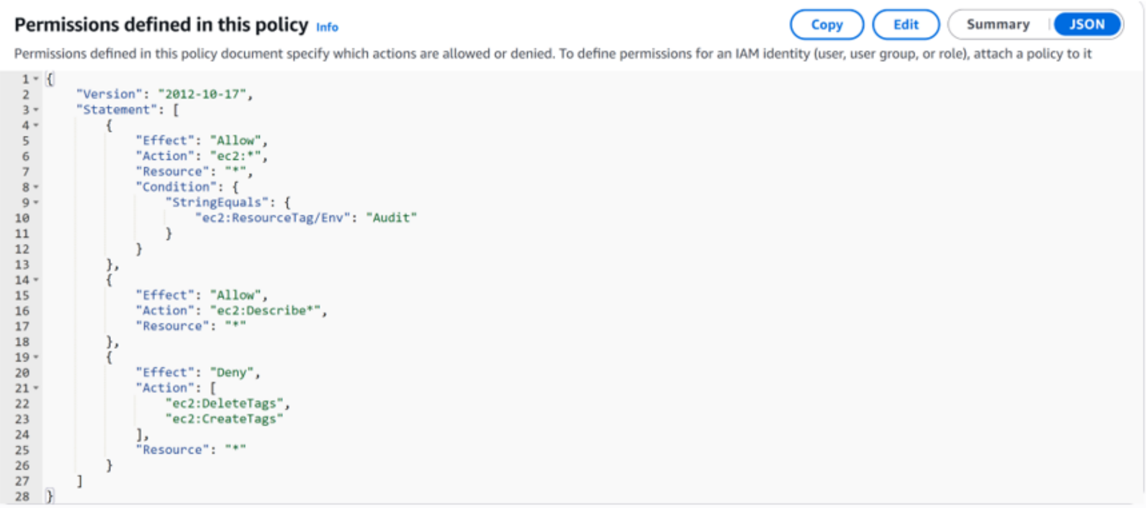
- Allows **read-only** access to EC2 resources.
- Allows **start/stop** actions only on instances tagged `Environment=Sales`.
- Does **not** grant start/stop permissions for instances tagged `Environment=Audit`.

Policy Logic

- **Read-only:**
  All instances can be described so the user can see what exists.
- **Start/Stop allowed:**
  Only when the target EC2 instance has the tag `Environment=Sales`.
- **Implicit deny:**
  Because there is no matching allow for `Environment=Audit`, start/stop on the `audit` instance is denied.

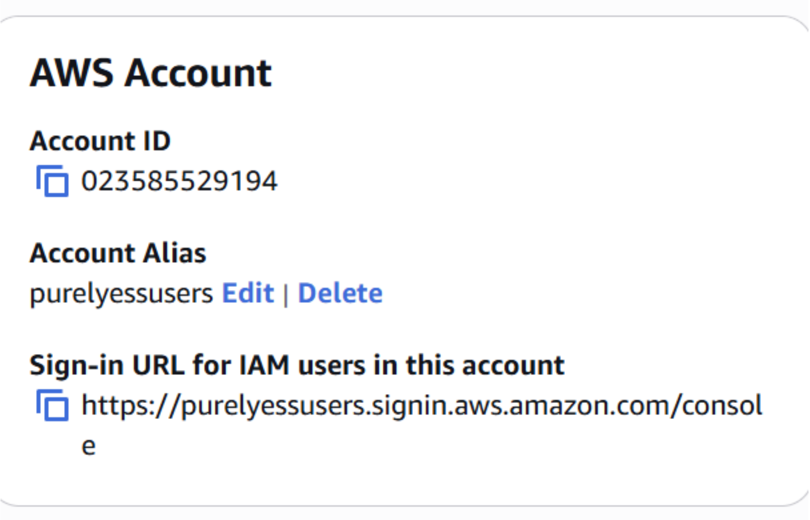
Example IAM Policy (JSON)

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllowReadOnlyEC2",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeInstanceStatus",
    "ec2:DescribeTags"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AllowStartStopSalesOnly",
   "Effect": "Allow",
   "Action": [
    "ec2:StartInstances",
    "ec2:StopInstances"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "ec2:ResourceTag/Environment": "Sales"
    }
   }
  }
 ]
}
```



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

```
{
    "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": "*"
        }
    ]
}
```



AWS Account

Account ID

023585529194

Account Alias

purelyessusers Edit | Delete

Sign-in URL for IAM users in this account

https://purelyessusers.signin.aws.amazon.com/console

7. IAM Group and User Setup

1. **Create IAM Group**
   a. Path: **IAM → User groups → Create group**.
   b. Group name: `Audit`.
   c. Attached policy: `PurelyessAuditEnvPolicy`.
2. **Create IAM Users**
   a. Path: **IAM → Users → Add users**.

b. Created standard IAM user accounts (e.g., `audit-operator1`).

c. Granted **console access** with a login password.

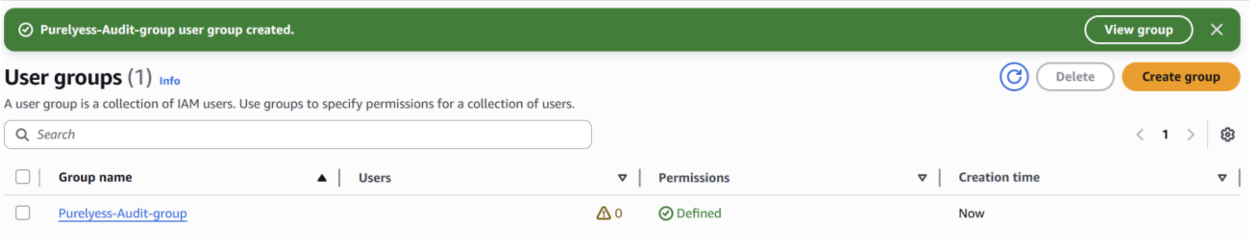
d. Added users to the `Audit` group so they inherit the group policy.

3. **Sign-In Options for IAM Users**

a. Via the **custom account alias URL** in the browser.

b. Optionally, via **AWS CLI** if programmatic access keys are created.

This structure keeps permissions centralized at the **group** level, which is easier to manage and audit than assigning policies directly to users.

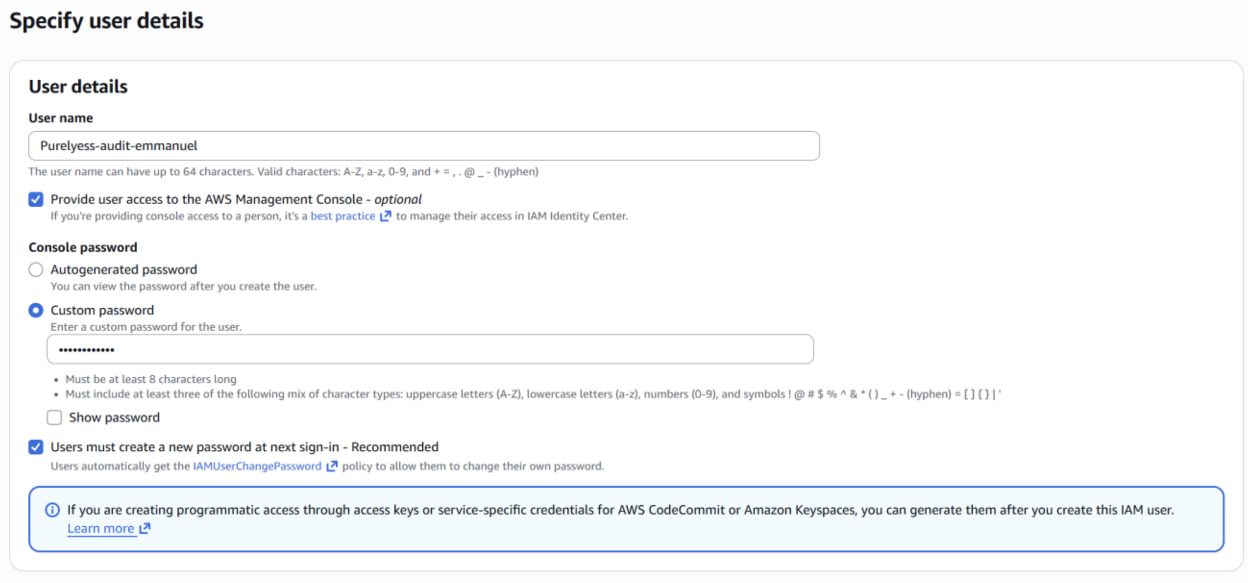


Purelyess-Audit-group user group created.

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
□	Purelyess-Audit-group	⚠ 0	✓ Defined	Now



Specify user details

User details

User name

Purelyess-audit-emmanuel

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.

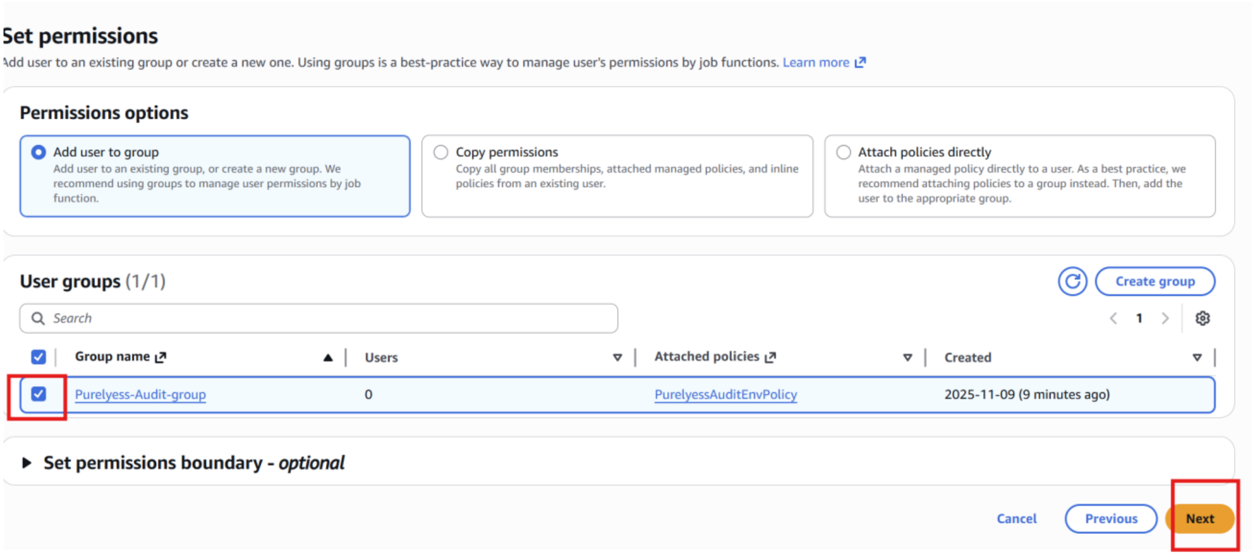
••••••••••••

- 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) = [ ] { } | '

□ Show password

☑ 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



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.

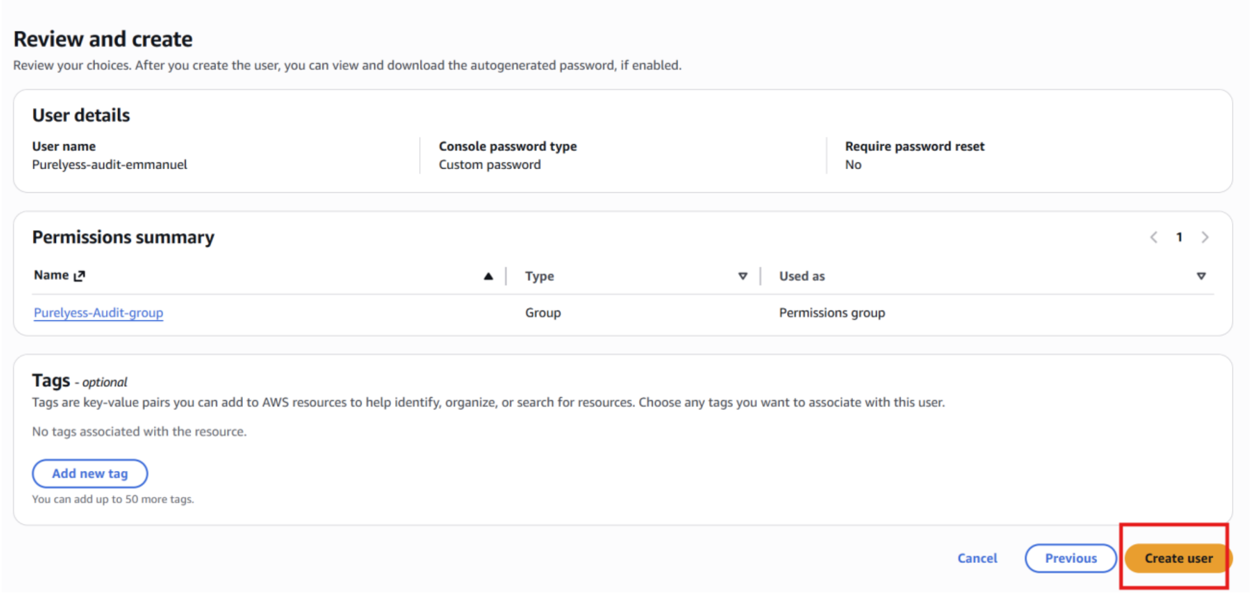
○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

○ 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 (1/1)

☑	Group name	Users	Attached policies	Created
☑	Purelyess-Audit-group	0	PurelyessAuditEnvPolicy	2025-11-09 (9 minutes ago)

▶ Set permissions boundary - *optional*



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
Purelyess-audit-emmanuel	Custom password	No

Permissions summary

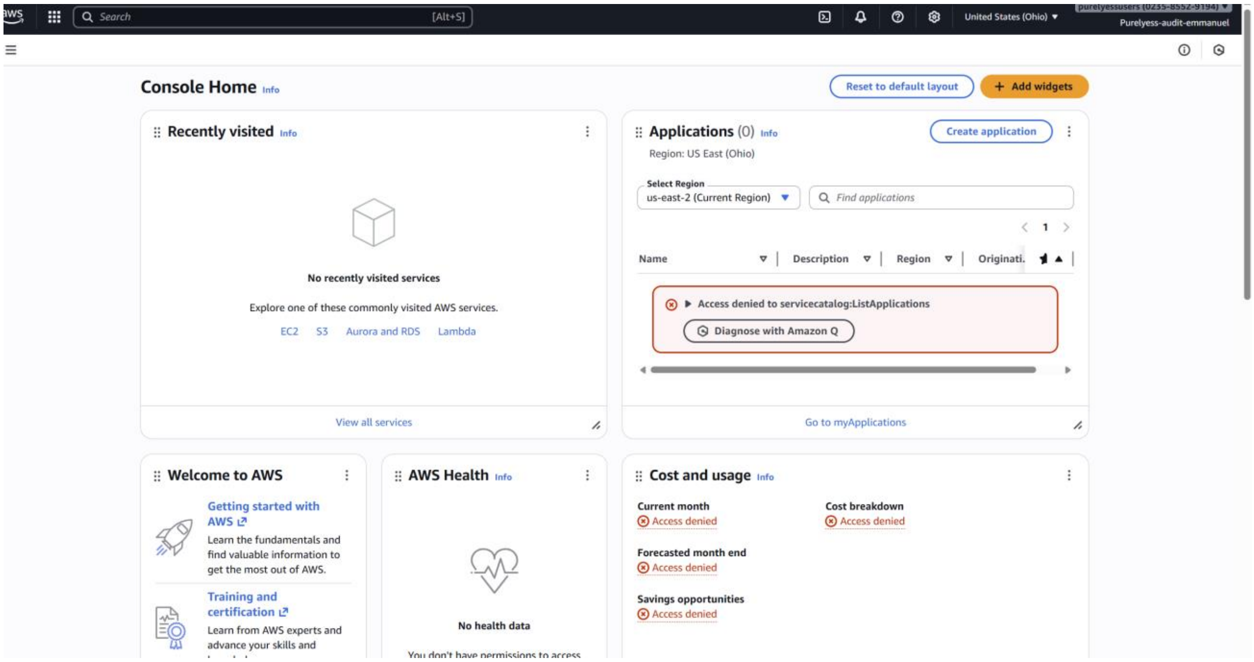
Name	Type	Used as
Purelyess-Audit-group	Group	Permissions group

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.

You can add up to 50 more tags.



No recently visited services

Explore one of these commonly visited AWS services.

Access denied to servicecatalog:ListApplications

Learn the fundamentals and find valuable information to get the most out of AWS.

Learn from AWS experts and advance your skills and

No health data

You don't have permissions to access

Current month: Access denied

Cost breakdown: Access denied

Forecasted month end: Access denied

Savings opportunities: Access denied

8. Testing the Access Controls

To verify the policy, I performed the following tests while logged in as the IAM user (not as root):

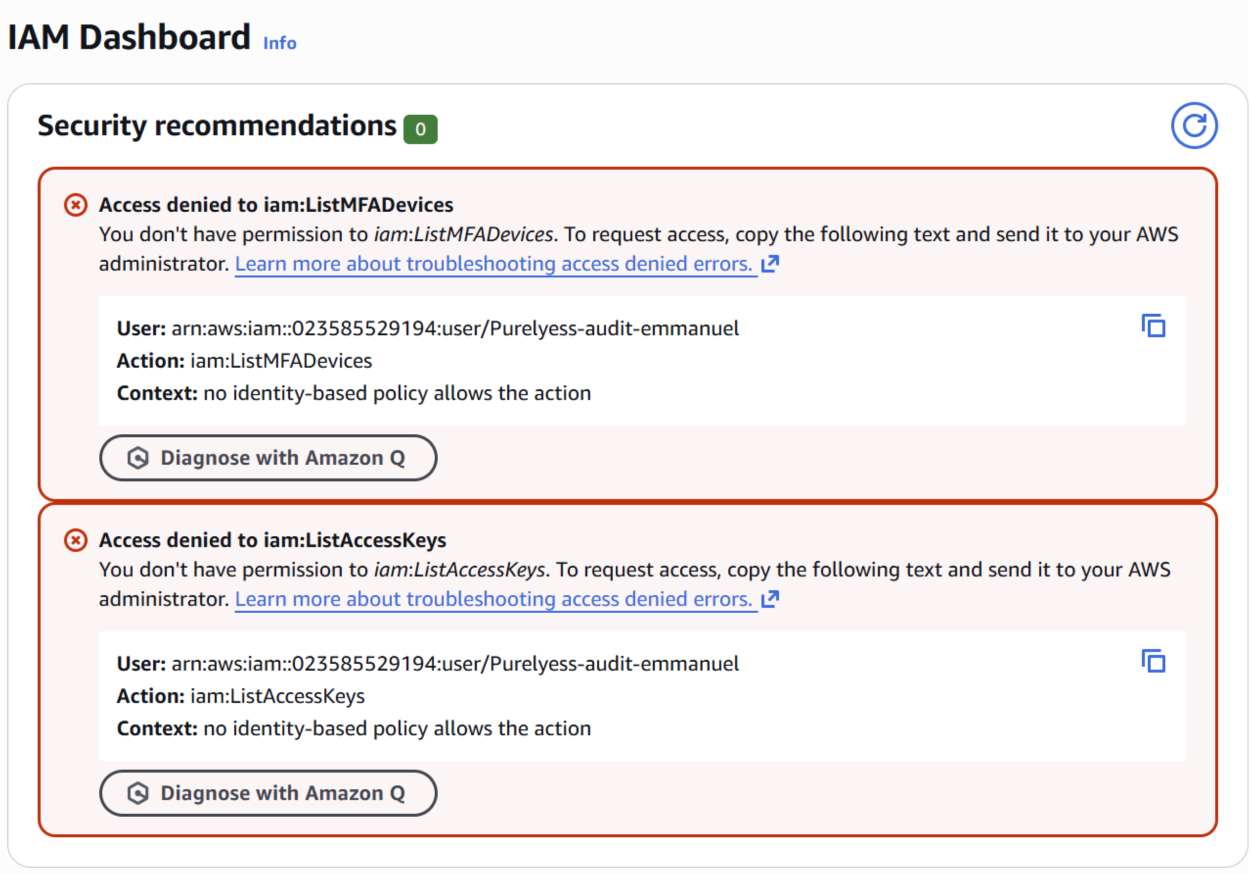
Test Matrix

Test Action	Target Instance	Expected Result	Actual Result
Stop instance	`audit`	Denied	Access denied (IAM policy enforced)
Stop instance	`sales`	Allowed	Instance stopped successfully
Start instance	`audit`	Denied	Access denied (IAM policy enforced)
Start instance	`sales`	Allowed	Instance started successfully

The results matched the design:

- Attempts to start/stop the **`audit`** instance failed with an **"Access denied"** error due to missing permissions for resources tagged `Environment=Audit`.

- Start/stop operations on the **`sales`** instance succeeded because the condition on `Environment=Sales` was satisfied.



IAM Dashboard

Security recommendations 0

Access denied to iam:ListMFADevices

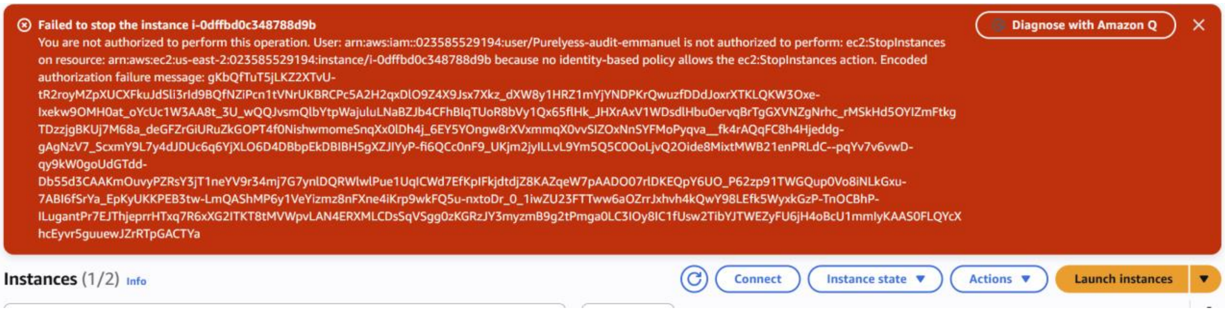
You don't have permission to *iam:ListMFADevices*. To request access, copy the following text and send it to your AWS administrator. Learn more about troubleshooting access denied errors.

User: arn:aws:iam::023585529194:user/Purelyess-audit-emmanuel
Action: iam:ListMFADevices
Context: no identity-based policy allows the action

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::023585529194:user/Purelyess-audit-emmanuel
Action: iam:ListAccessKeys
Context: no identity-based policy allows the action



Failed to stop the instance i-0dffbd0c348788d9b

```
You are not authorized to perform this operation. User: arn:aws:iam::023585529194:user/Purelyess-audit-emmanuel is not authorized to perform: ec2:StopInstances
on resource: arn:aws:ec2:us-east-2:023585529194:instance/i-0dffbd0c348788d9b because no identity-based policy allows the ec2:StopInstances action. Encoded
authorization failure message: gKbQfTuT5jLKZ2XTvU-
tR2royMZpXUCXFkuJdSli3rId9BQfNZiPcn1tVNrUKBRCPc5A2H2qxDlO9Z4X9Jsx7Xkz_dXW8y1HRZ1mYjYNDPKrQwuzfDDdJoxrXTKLQKW3Oxe-
lxekw9OMH0at_oYcUc1W3AA8t_3U_wQQJvsmQlbYtpWajuluLNaBZJb4CFhBlqTUoR8bVy1Qx65flHk_JHXrAxV1WDsdlHbu0ervqBrTgGXVNZgNrhc_rMSkHd5OYIZmFtkg
TDzzjgBKUj7M68a_deGFZrGiURuZkGOPT4f0NishwmomeSnqXx0lDh4j_6EY5YOngw8rXVxmmqX0vvSIZOxNnSYFMoPyqva__fk4rAQqFC8h4Hjeddg-
gAgNzV7_ScxmY9L7y4dJDUc6q6YjXLO6D4DBbpEkDBIBH5gXZJIYyP-fi6QCc0nF9_UKjm2jyILLvL9Ym5Q5C0OoLjvQ2Oide8MixtMWB21enPRLdC--pqYv7v6vwD-
qy9kW0goUdGTdd-
Db55d3CAAKmOuvyPZRsY3jT1neYV9r34mj7G7ynlDQRWlwlPue1UqICWd7EfKpIFkjdtdjZ8KAZqeW7pAADO07rlDKEQpY6UO_P62zp91TWGQup0Vo8iNLkGxu-
7ABI6fSrYa_EpKyUKKPEB3tw-LmQAShMP6y1VeYizmz8nFXne4iKrp9wkFQ5u-nxtoDr_0_1iwZU23FTTww6aOZrrJxhvh4kQwY98LEfk5WyxkGzP-TnOCBhP-
ILugantPr7EJThjeprrHTxq7R6xXG2ITKT8tMVWpvLAN4ERXMLCDsSqVSgg0zKGRzJY3myzmB9g2tPmga0LC3IOy8IC1fUsw2TibYJTWEZyFU6jH4oBcU1mmlyKAAS0FLQYcX
hcEyvr5guuewJZrRTpGACTYa
```

9. Security Impact

This configuration demonstrates several important security principles:

- **Least Privilege:**
  Users receive exactly the permissions needed to manage the `sales` instance, and nothing more.
- **Tag-Based Access Control:**
  Access is controlled by **resource tags**, which makes it easy to scale this design:
  - Any new instance tagged `Environment=Sales` automatically falls under the same policy.
  - Sensitive resources (tagged differently) remain protected without editing the policy.
- **Separation of Duties:**
  Critical systems such as `audit` can be reserved for administrators, while operational users can manage non-critical workloads.

10. Lessons Learned & Next Steps

Lessons Learned

- Tags are not just for organization; they are powerful **security controls** when combined with IAM conditions.
- Designing IAM policies correctly requires:
  - Clear understanding of resources and actions.
  - Careful use of conditions to avoid over-privileged access.
- Testing with a **non-admin IAM user** is essential to confirm that policies behave as expected.

Possible Enhancements

- Enable **AWS CloudTrail** to log all EC2 start/stop actions and review them for unauthorized attempts.
- Add **CloudWatch Alarms** to alert when critical instances (like `audit`) are stopped or when a denied action occurs.

- Extend the design using **Infrastructure as Code** (e.g., Terraform or CloudFormation) to automate instance creation, tagging, and IAM policy deployment.
- Introduce **Service Control Policies (SCPs)** under AWS Organizations for stricter guardrails at the account or OU level.