Secure Cloud Foundation Deployment (Production-Minded Mini Architecture)

Introduction

This project implements a secure, access-controlled, and cost-governed AWS cloud environment that simulates a small company workload.

The objective is to demonstrate foundational cloud architecture, security controls, and financial governance aligned with AWS best practices.

The deployment focuses on secure configuration rather than feature complexity.

Project Objective

The main objectives of this project are:

1. Establish secure AWS account baseline configuration
2. Implement identity and access control using IAM
3. Deploy compute resource with restricted network exposure
4. Protect storage using secure configuration and encryption
5. Implement cost monitoring to prevent financial risk
6. Demonstrate understanding of AWS Shared Responsibility Model

---

Scope of the Project

Included:

- AWS account hardening
- IAM user and group implementation
- EC2 deployment with restricted Security Group

- S3 bucket with Block Public Access enabled
- AWS Budget configuration

Not Included:

- Multi-tier architecture
- Load balancers
- Auto scaling
- Infrastructure as Code
- Advanced networking

This project focuses strictly on foundational security architecture.

---

5. Architecture Overview

Core Components:

- **AWS Account (Hardened)**
- **IAM (Identity & Access Management)**
- **Default VPC**
- **Security Group (Restricted SSH)**
- **EC2 Instance (Compute Layer)**
- **S3 Bucket (Private Storage)**
- **AWS Budget (Cost Monitoring)**

How the Architecture Works

1. The AWS Account provides the cloud control plane.
2. IAM controls who can access resources and what actions they can perform.
3. The Default VPC provides isolated networking.
4. EC2 instance runs inside the VPC.
5. Security Group acts as a stateful firewall controlling inbound traffic.
6. S3 bucket stores company data securely with public access blocked.
7. AWS Budget monitors spending and triggers alerts.

Data flow:

User → IAM Authentication → EC2 via Security Group → S3 (private storage)

Main Security Controls Implemented

- Root MFA enabled
- No daily root usage
- Least privilege IAM model
- SSH restricted to specific IP
- S3 Block Public Access enabled
- Default encryption for S3
- Budget alert to prevent financial abuse

Shared Responsibility Context

AWS Responsible For:

- Physical data centers
- Hardware
- Network infrastructure
- Hypervisor

Customer Responsible For:

- IAM configuration
- Security Group rules
- Data encryption
- OS-level security inside EC2
- Cost control

This project validates understanding of “security in the cloud.”

Risk Mitigation Strategy

Risk	Control Implemented
Account takeover	Root MFA
Over-privileged access	IAM least privilege
SSH brute force	Restricted IP
Data leakage	S3 Block Public Access
Unexpected cost spike	AWS Budget

Expected Learning Outcomes

After completion, the implementer will be able to:

- Explain Shared Responsibility Model
- Implement IAM securely
- Configure network access controls
- Secure S3 storage
- Understand AWS pricing behavior
- Navigate AWS Console confidently

Success Criteria

Project is successful if:

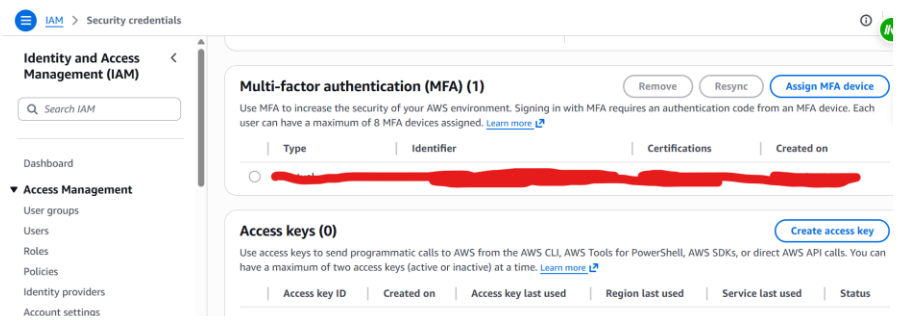
- No resource is publicly exposed
- IAM follows least privilege
- Budget alert is functional
- EC2 accessible only from approved IP
- S3 objects are not publicly readable

Implementation Procedure (Step-by-Step Execution)

Phase 1 — Account Hardening (Non-Negotiable)

Enable Root MFA

1. Login as root
2. Go to: IAM → Dashboard> security credentials
3. Enable MFA on root account



Multi-factor authentication (MFA) (1)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

Type	Identifier	Certifications	Created on
[redacted]			

Access keys (0)

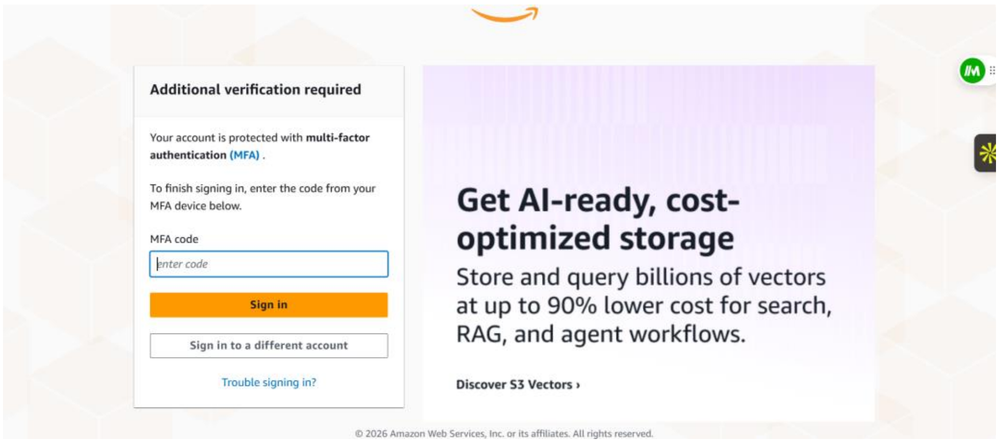
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

Access key ID	Created on	Access key last used	Region last used	Service last used	Status

4. Use authenticator app
5. Verify login with MFA

Validation:

- Root shows "MFA Enabled"


Additional verification required
MFA code
enter code
Sign in
Sign in to a different account
Trouble signing in?
Get AI-ready, cost-optimized storage
Discover S3 Vectors ›

Your account is protected with multi-factor authentication (MFA) .

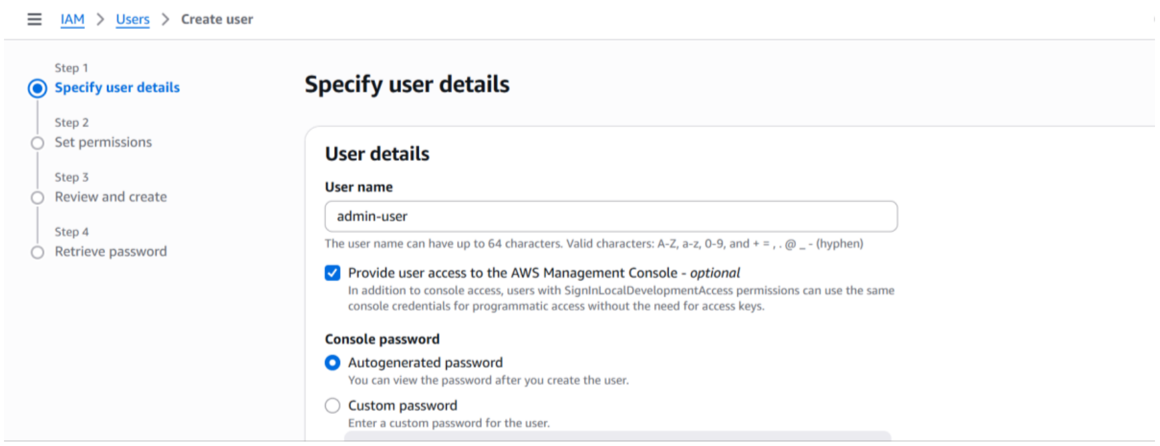
To finish signing in, enter the code from your MFA device below.

Store and query billions of vectors at up to 90% lower cost for search, RAG, and agent workflows.

© 2026 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Create IAM Admin User (Daily Use Account)

1. IAM → Users → Create user
2. Username: admin-user



Specify user details

User details

User name

admin-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

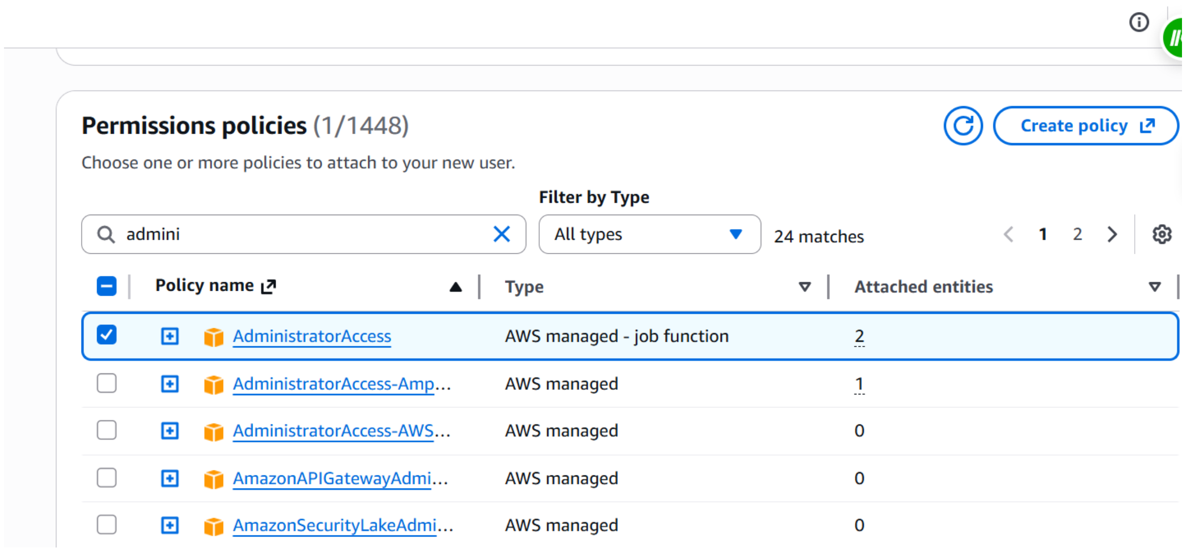
[✓] Provide user access to the AWS Management Console - *optional*
In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

Console password

(●) Autogenerated password
You can view the password after you create the user.

( ) Custom password
Enter a custom password for the user.

3. Enable console access
4. Attach policy: AdministratorAccess



Permissions policies (1/1448)

Choose one or more policies to attach to your new user.

Filter by Type

admini | All types | 24 matches | < 1 2 >

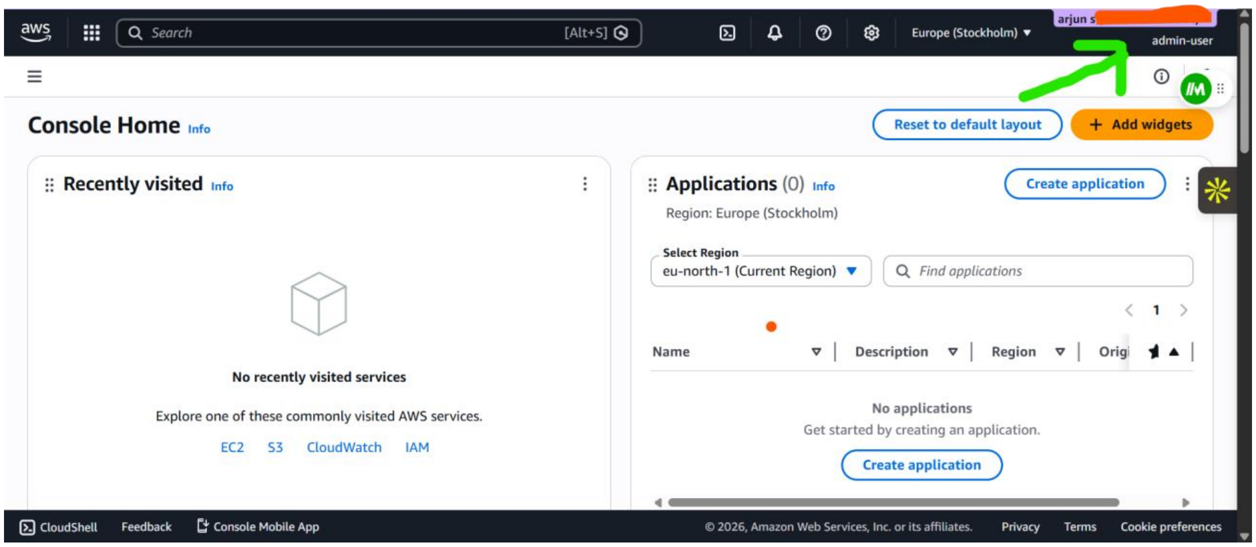
	Policy name	Type	Attached entities
✓	AdministratorAccess	AWS managed - job function	2
	AdministratorAccess-Amp...	AWS managed	1
	AdministratorAccess-AWS...	AWS managed	0
	AmazonAPIGatewayAdmi...	AWS managed	0
	AmazonSecurityLakeAdmi...	AWS managed	0

5. Download credentials

Logout root.
Login using admin-user.

Validation:

- Admin can access services
- Root not used again



arjun s[redacted]
admin-user

Europe (Stockholm)

Console Home Info

Reset to default layout | + Add widgets

Recently visited Info

No recently visited services

Explore one of these commonly visited AWS services.

EC2 S3 CloudWatch IAM

Applications (0) Info

Create application

Region: Europe (Stockholm)

Select Region: eu-north-1 (Current Region)

Find applications

Name	Description	Region	Orig

No applications
Get started by creating an application.

Create application

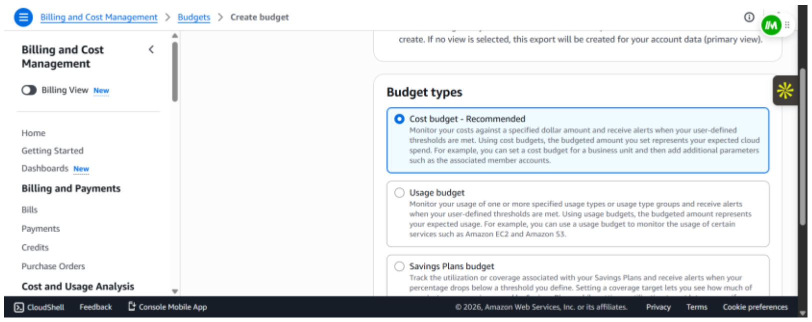
CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

---

Phase 2 — Cost Governance Setup

Create AWS Budget

1. Billing → Budgets → Create Budget
2. Type: Cost Budget



Billing and Cost Management > Budgets > Create budget

Billing and Cost Management

Billing View New

Home
Getting Started
Dashboards New
Billing and Payments
Bills
Payments
Credits
Purchase Orders
Cost and Usage Analysis

create. If no view is selected, this export will be created for your account data (primary view).

Budget types

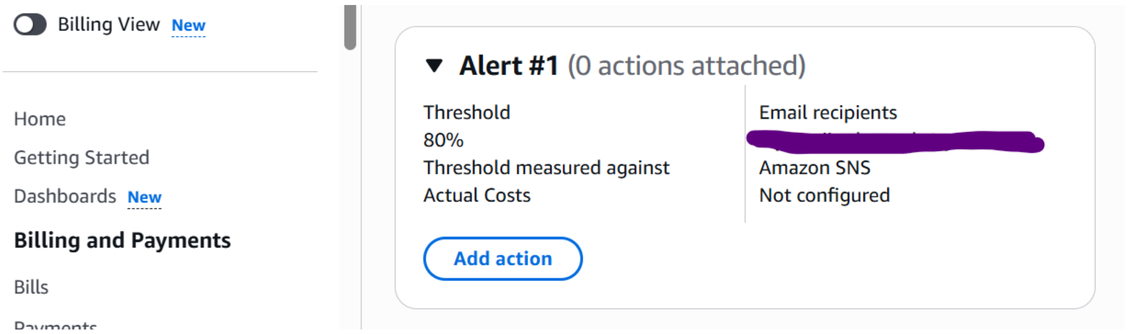
[✓ Cost budget - Recommended]
Monitor your costs against a specified dollar amount and receive alerts when your user-defined thresholds are met. Using cost budgets, the budgeted amount you set represents your expected cloud spend. For example, you can set a cost budget for a business unit and then add additional parameters such as the associated member accounts.

Usage budget
Monitor your usage of one or more specified usage types or usage type groups and receive alerts when your user-defined thresholds are met. Using usage budgets, the budgeted amount represents your expected usage. For example, you can use a usage budget to monitor the usage of certain services such as Amazon EC2 and Amazon S3.

Savings Plans budget
Track the utilization or coverage associated with your Savings Plans and receive alerts when your percentage drops below a threshold you define. Setting a coverage target lets you see how much of

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

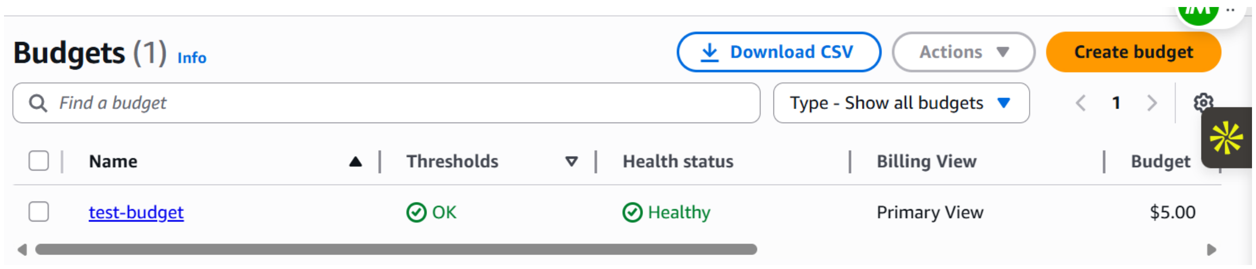
3. Monthly
4. Amount: $5
5. Add email alert at 80%



Alert #1 (0 actions attached)

Threshold
80%
Threshold measured against
Actual Costs

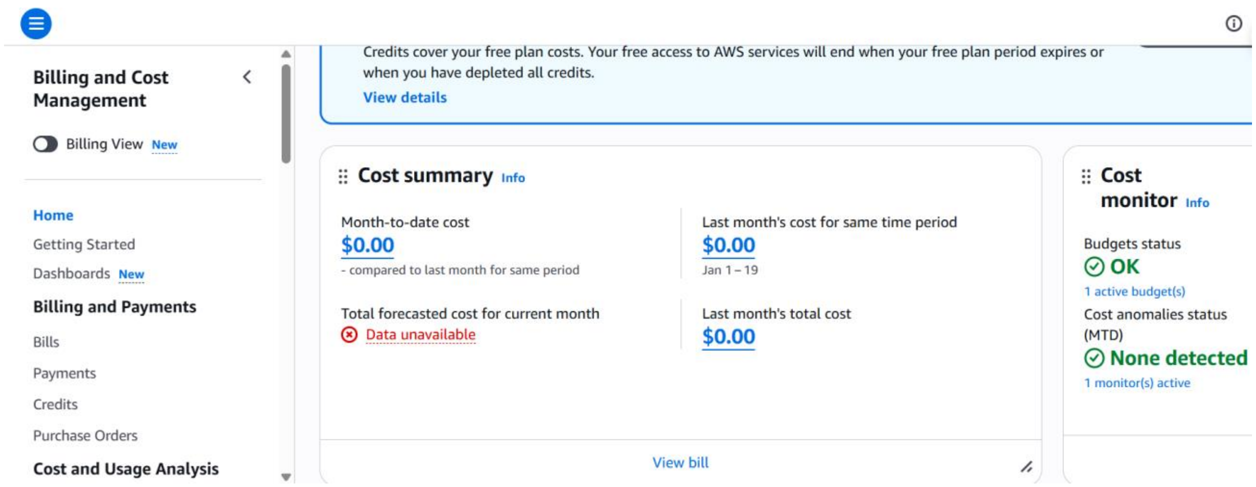
Email recipients
[redacted]
Amazon SNS
Not configured


Budgets (1) Info

Name	Thresholds	Health status	Billing View	Budget
test-budget	✓ OK	✓ Healthy	Primary View	$5.00

Validation:

- Budget visible in dashboard
- Email alert configured



Credits cover your free plan costs. Your free access to AWS services will end when your free plan period expires or when you have depleted all credits.
View details

Cost summary Info

Month-to-date cost
$0.00
- compared to last month for same period

Last month's cost for same time period
$0.00
Jan 1 – 19

Total forecasted cost for current month
Data unavailable

Last month's total cost
$0.00

View bill

Cost monitor Info

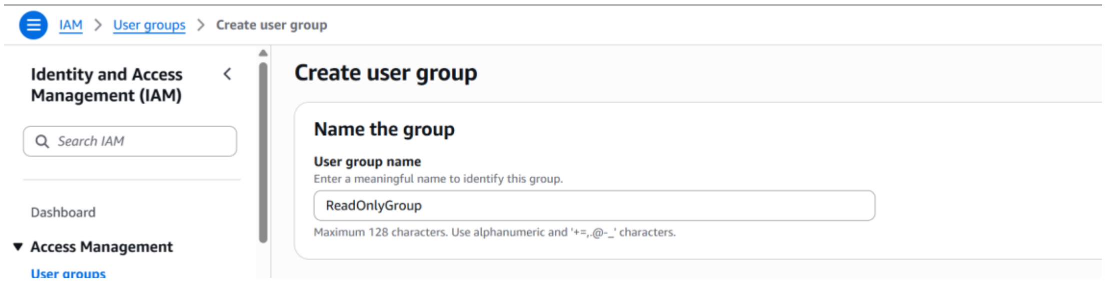
Budgets status
✓ OK
1 active budget(s)

Cost anomalies status (MTD)
✓ None detected
1 monitor(s) active

Phase 3 — IAM Least Privilege Model

Create ReadOnly Group

1. IAM → User Groups → Create group
2. Name: ReadOnlyGroup



IAM > User groups > Create user group

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access Management

User groups

Create user group

Name the group

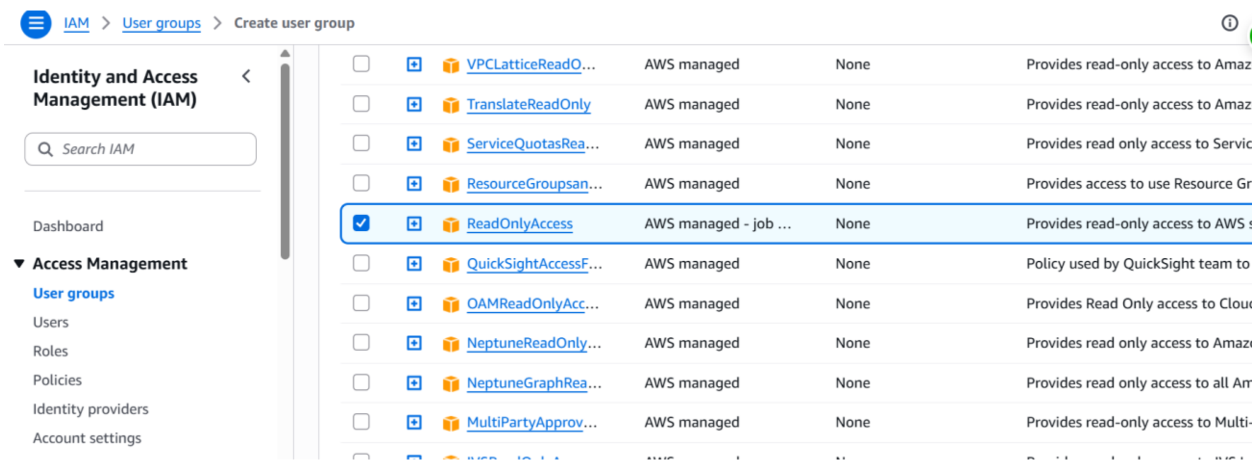
User group name

Enter a meaningful name to identify this group.

ReadOnlyGroup

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

3. Attach policy: ReadOnlyAccess



IAM > User groups > Create user group

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access Management

User groups

Users

Roles

Policies

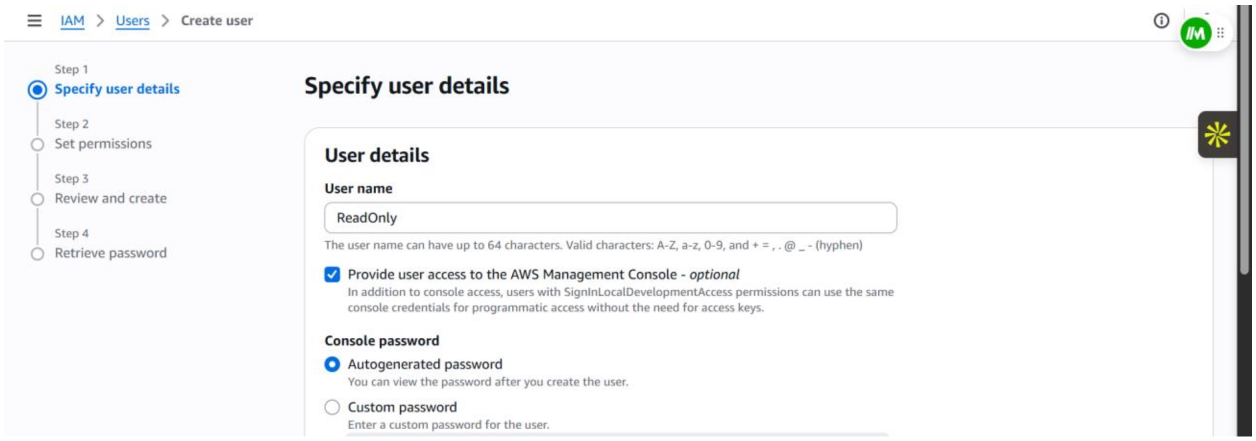
Identity providers

Account settings

☐ VPCLatticeReadO…	AWS managed	None	Provides read-only access to Amaz
☐ TranslateReadOnly	AWS managed	None	Provides read-only access to Amaz
☐ ServiceQuotasRea…	AWS managed	None	Provides read only access to Servic
☐ ResourceGroupsan…	AWS managed	None	Provides access to use Resource Gr
✓ ReadOnlyAccess	AWS managed - job …	None	Provides read-only access to AWS s
☐ QuickSightAccessF…	AWS managed	None	Policy used by QuickSight team to
☐ OAMReadOnlyAcc…	AWS managed	None	Provides Read Only access to Clouc
☐ NeptuneReadOnly…	AWS managed	None	Provides read only access to Amaz
☐ NeptuneGraphRea…	AWS managed	None	Provides read only access to all Am
☐ MultiPartyApprov…	AWS managed	None	Provides read-only access to Multi-

Create ReadOnly User

1. IAM → Users → Create user


IAM > Users > Create user

Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password

Specify user details

User details

User name

ReadOnly

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

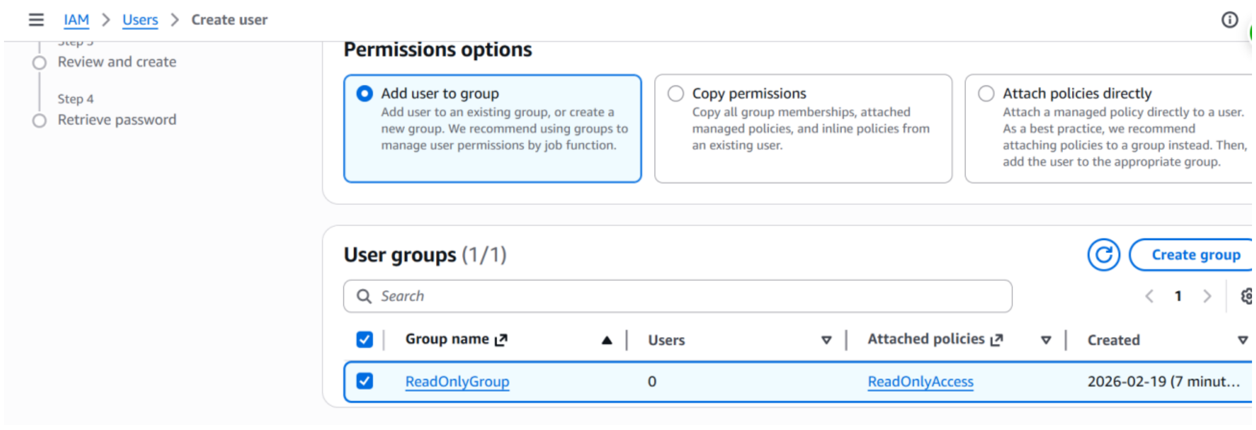
[✓] Provide user access to the AWS Management Console - *optional*
In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

Console password

[✓ Autogenerated password]
You can view the password after you create the user.

Custom password
Enter a custom password for the user.

2. Add to ReadOnlyGroup
3. Enable console login


IAM > Users > Create user

Review and create
Step 4
Retrieve password

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.

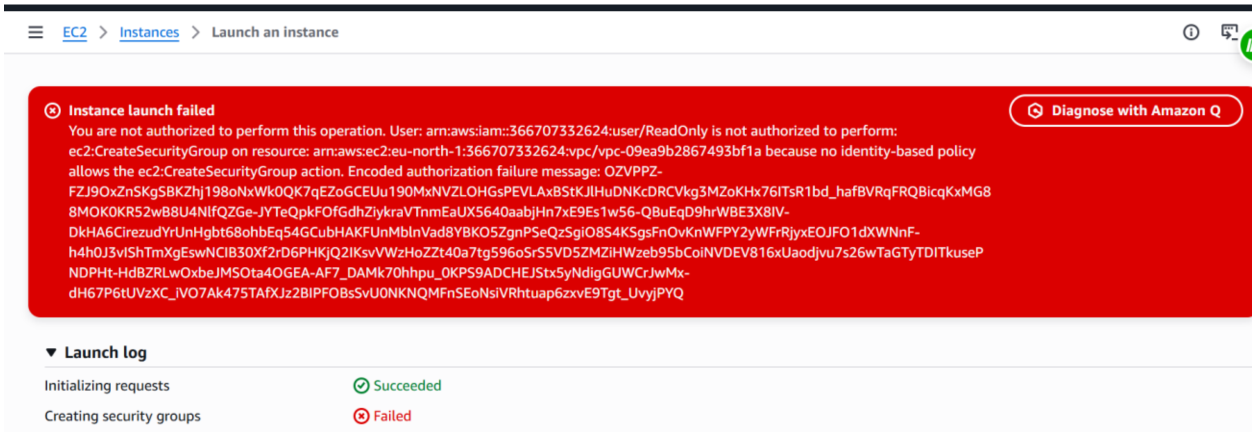
User groups (1/1)

	Group name	Users	Attached policies	Created
✓	ReadOnlyGroup	0	ReadOnlyAccess	2026-02-19 (7 minut…

4. Test login

Validation:

- ReadOnly user cannot launch EC2
- Can only view resources



EC2 > Instances > Launch an instance

Instance launch failed

```
You are not authorized to perform this operation. User: arn:aws:iam::366707332624:user/ReadOnly is not authorized to perform:
ec2:CreateSecurityGroup on resource: arn:aws:ec2:eu-north-1:366707332624:vpc/vpc-09ea9b2867493bf1a because no identity-based policy
allows the ec2:CreateSecurityGroup action. Encoded authorization failure message: OZVPPZ-
FZJ9OxZnSKgSBKZhj198oNxWk0QK7qEZoGCEUu190MxNVZLOHGsPEVLAxBStKJlHuDNKcDRCVkg3MZoKHx76ITsR1bd_hafBVRqFRQBicqKxMG8
8MOK0KR52wB8U4NlfQZGe-JYTeQpkFOfGdhZiykraVTnmEaUX5640aabjHn7xE9Es1w56-QBuEqD9hrWBE3X8IV-
DkHA6CirezudYrUnHgbt68ohbEq54GCubHAKFUnMblnVad8YBKO5ZgnPSeQzSgiO8S4KSgsFnOvKnWFPY2yWFrRjyxEOJFO1dXWNnF-
h4h0J3vIShTmXgEswNCIB30Xf2rD6PHKjQ2IKsvVWzHoZZt40a7tg596oSrS5VD5ZMZiHWzeb95bCoiNVDEV816xUaodjvu7s26wTaGTyTDITkuseP
NDPHt-HdBZRLwOxbeJMSOta4OGEA-AF7_DAMk70hhpu_0KPS9ADCHEJStx5yNdigGUWCrJwMx-
dH67P6tUVzXC_iVO7Ak475TAfXJz2BIPFOBsSvU0NKNQMFnSEoNsiVRhtuap6zxvE9Tgt_UvyjPYQ
```

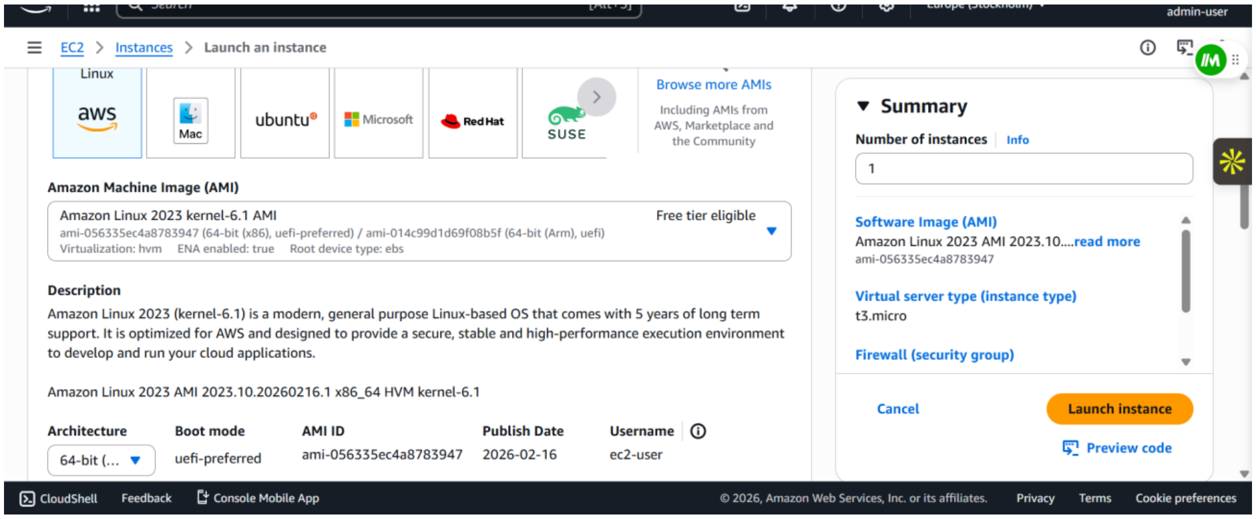
▼ Launch log

Initializing requests	✓ Succeeded
Creating security groups	✗ Failed

Phase 4 — Compute + Network Security

Launch EC2 Instance

1. EC2 → Launch Instance
2. Choose Amazon Linux 2023
3. Instance type: t3.micro (Free tier)



EC2 > Instances > Launch an instance

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI — Free tier eligible
ami-056335ec4a8783947 (64-bit (x86), uefi-preferred) / ami-014c99d1d69f08b5f (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 2023.10.20260216.1 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username
64-bit (...	uefi-preferred	ami-056335ec4a8783947	2026-02-16	ec2-user

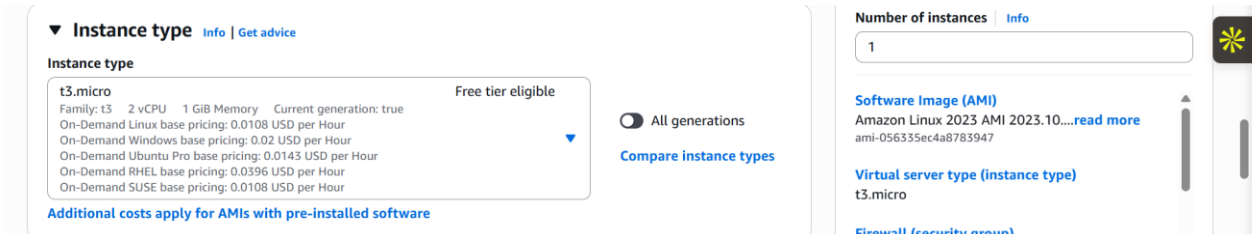
▼ **Summary**

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.10....read more
ami-056335ec4a8783947

Virtual server type (instance type): t3.micro

Firewall (security group)



▼ Instance type Info | Get advice

Instance type

t3.micro Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0108 USD per Hour
On-Demand Windows base pricing: 0.02 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0143 USD per Hour
On-Demand RHEL base pricing: 0.0396 USD per Hour
On-Demand SUSE base pricing: 0.0108 USD per Hour

Additional costs apply for AMIs with pre-installed software

All generations

Compare instance types

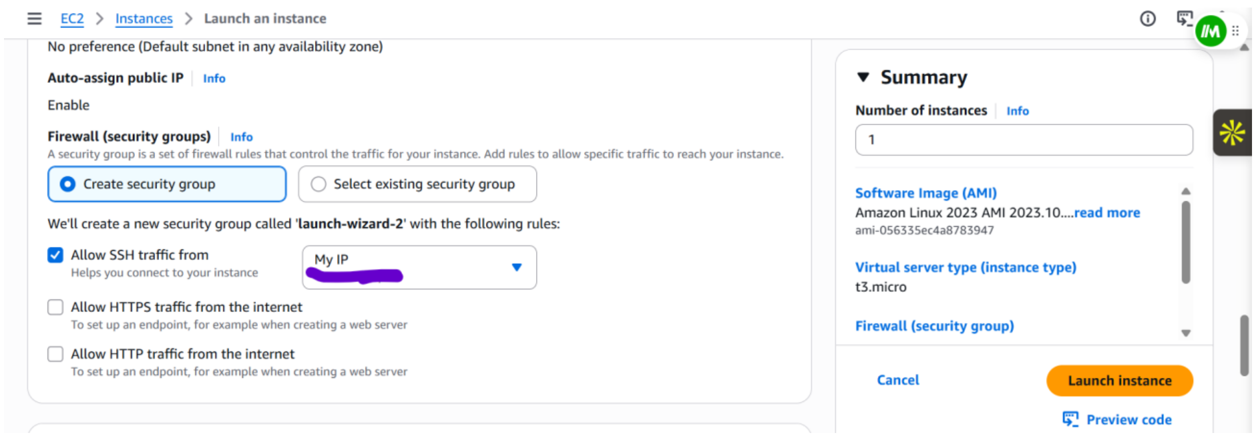
Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.10....read more
ami-056335ec4a8783947

Virtual server type (instance type)
t3.micro

4. Use default VPC
5. Create key pair
6. Configure Security Group:
   - SSH (port 22)
   - Source: My IP only

DO NOT use 0.0.0.0/0.



EC2 > Instances > Launch an instance

No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

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-2' with the following rules:

☑ Allow SSH traffic from — My IP
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

▼ Summary

Number of instances Info
1

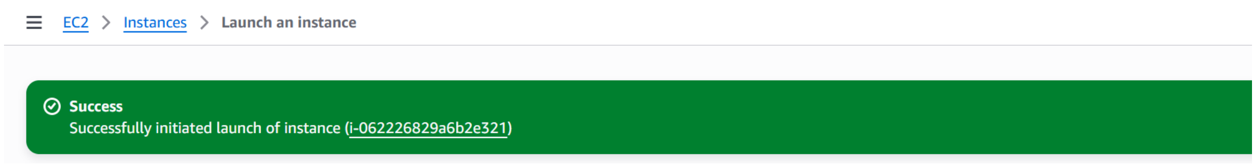
Software Image (AMI)
Amazon Linux 2023 AMI 2023.10....read more
ami-056335ec4a8783947

Virtual server type (instance type)
t3.micro

Firewall (security group)

Cancel Launch instance Preview code

Launch instance.



EC2 > Instances > Launch an instance

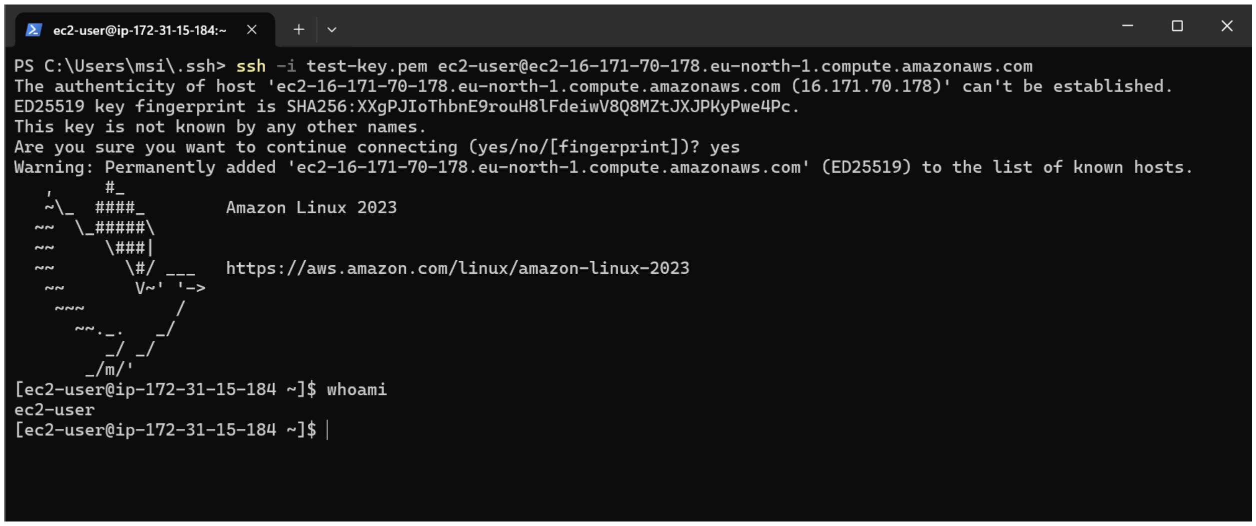
Success
Successfully initiated launch of instance (i-062226829a6b2e321)

SSH Test

From terminal:

ssh -i key.pem ec2-user@public-ip

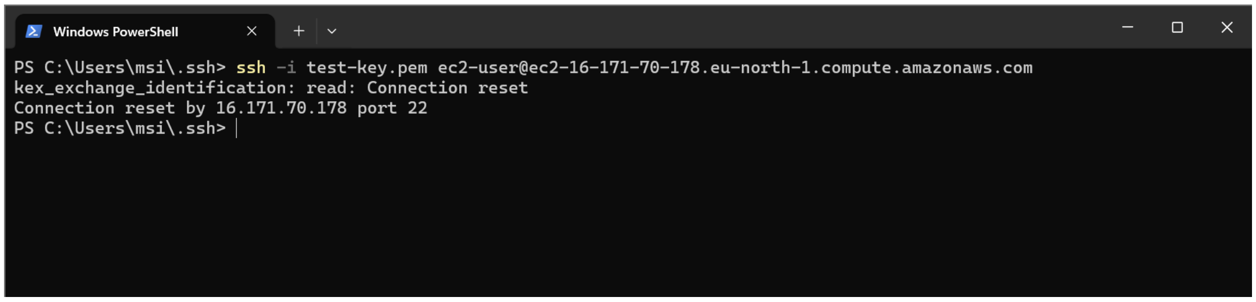
Confirm connection works.



```
PS C:\Users\msi\.ssh> ssh -i test-key.pem ec2-user@ec2-16-171-70-178.eu-north-1.compute.amazonaws.com
The authenticity of host 'ec2-16-171-70-178.eu-north-1.compute.amazonaws.com (16.171.70.178)' can't be established.
ED25519 key fingerprint is SHA256:XXgPJIoThbnE9rouH8lFdeiwV8Q8MZtJXJPKyPwe4Pc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-16-171-70-178.eu-north-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-15-184 ~]$ whoami
ec2-user
[ec2-user@ip-172-31-15-184 ~]$
```

Security Group Verification

- Try connecting from another IP (optional test)
- Confirm connection fails : I turn on the vpn therefore the IP has changed



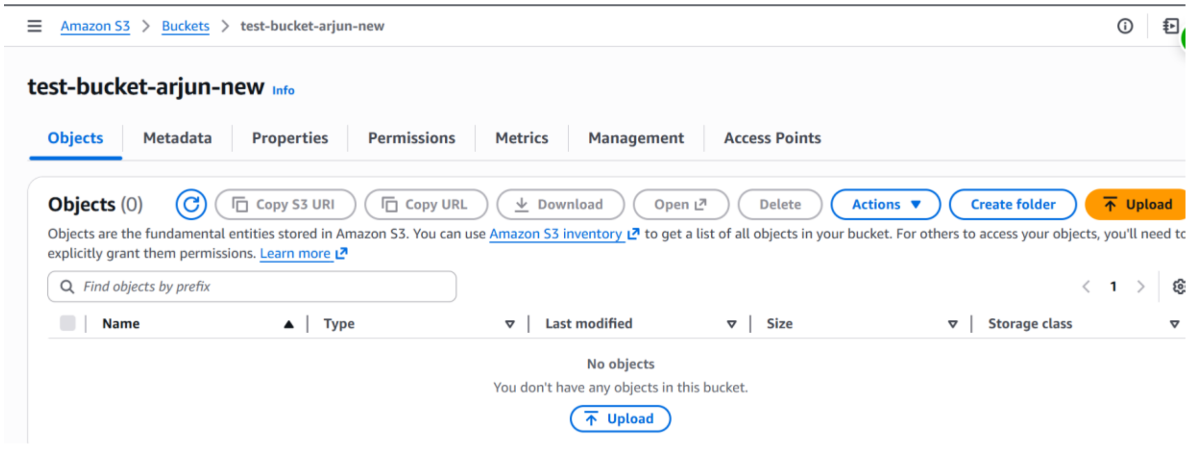
```
PS C:\Users\msi\.ssh> ssh -i test-key.pem ec2-user@ec2-16-171-70-178.eu-north-1.compute.amazonaws.com
kex_exchange_identification: read: Connection reset
Connection reset by 16.171.70.178 port 22
PS C:\Users\msi\.ssh>
```

Phase 5 — Storage Hardening

Create Secure S3 Bucket

1. S3 → Create bucket
2. Unique name
3. Keep Block Public Access = ON
4. Enable default encryption (SSE-S3)

Create bucket.



Amazon S3 > Buckets > test-bucket-arjun-new

test-bucket-arjun-new Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

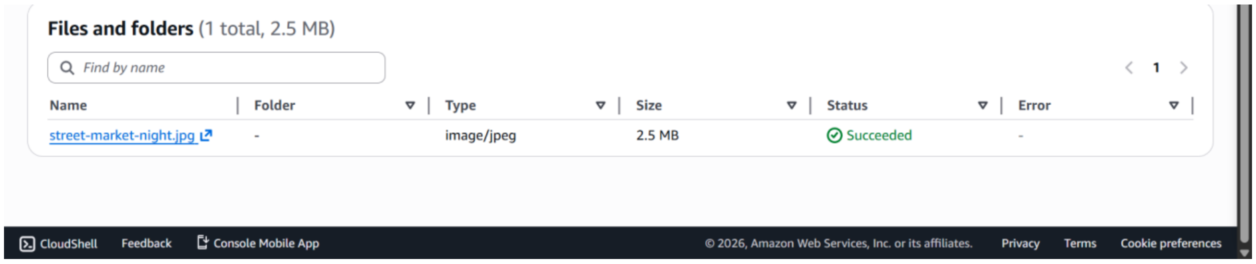
Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

Upload Test Object

Upload a small text file/img.



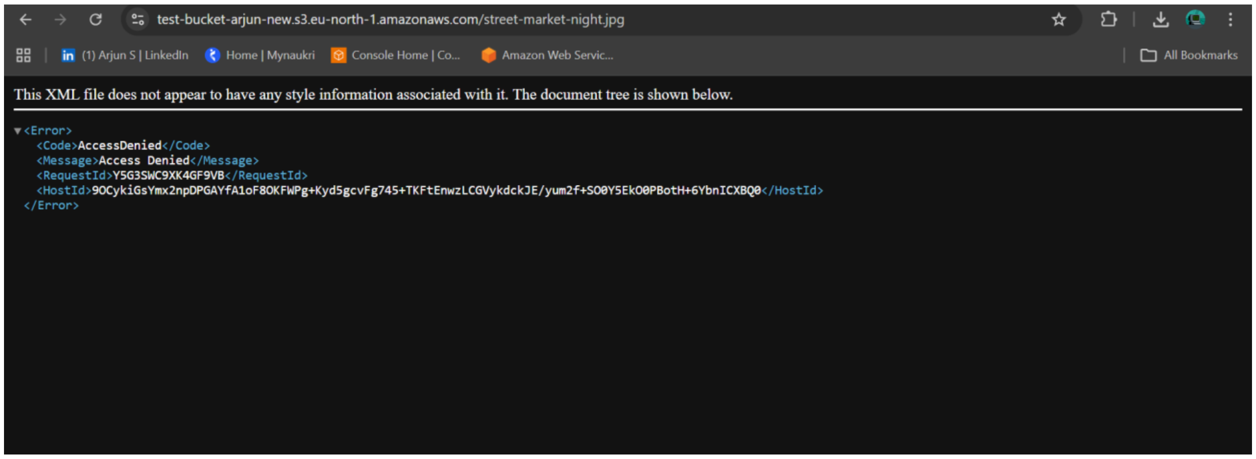
Files and folders (1 total, 2.5 MB)

Name	Folder	Type	Size	Status	Error
street-market-night.jpg	-	image/jpeg	2.5 MB	Succeeded	-

© 2026, Amazon Web Services, Inc. or its affiliates.

Try accessing via public URL.

Expected: Access Denied.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>Y5G3SWC9XK4GF9VB</RequestId>
   <HostId>9OCykiGsYmx2npDPGAYfA1oF8OKFWPg+Kyd5gcvFg745+TKFtEnwzLCGVykdckJE/yum2f+SO0Y5EkO0PBotH+6YbnICXBQ0</HostId>
 </Error>
```

Phase 6 — Architecture Validation

Verify Security Controls

Check:

- Root MFA enabled
- No public SSH
- ReadOnly user restricted
- S3 not public
- Budget active

Phase 7 — Cleanup (Cost Control Discipline)

Terminate EC2

1. EC2 → Instance → Terminate
2. Delete unused EBS volumes
3. Keep S3 bucket only if needed

Failure to cleanup = cost leak.

Final Statement

This Secure Cloud Foundation Deployment demonstrates practical implementation of core AWS security principles, identity management, network restriction, storage protection, and cost governance.

It establishes a validated baseline required before advancing to higher-level AWS architecture or cloud security engineering topics.

Project Completed Successfully.