PROJECT 4

AWS Config Compliance Enforcement

Project: AWS Config Managed Rules Practical Validation

Department: DevOPs

Team: Cloud Infrastructure Engineering

Team Leader: Chinazor Nwode

Co-Leader: Ifunanya Benedicta

As part of our **continuous compliance validation program**, we, the cloud security engineering team, conducted a **controlled misconfiguration test** in our AWS development environment.

Our objective was to:

1. **Deliberately introduce** specific misconfigurations into various AWS resources.
2. **Observe AWS Config's detection** of these misconfigurations in real time.
3. **Remediate** each issue and confirm compliance was restored.

We executed each scenario using the **STAR method** so that the process is clear, reproducible, and aligned with our organizational compliance playbooks.

---

1. EC2 Launch Template Policy – IMDSv2 Enforcement

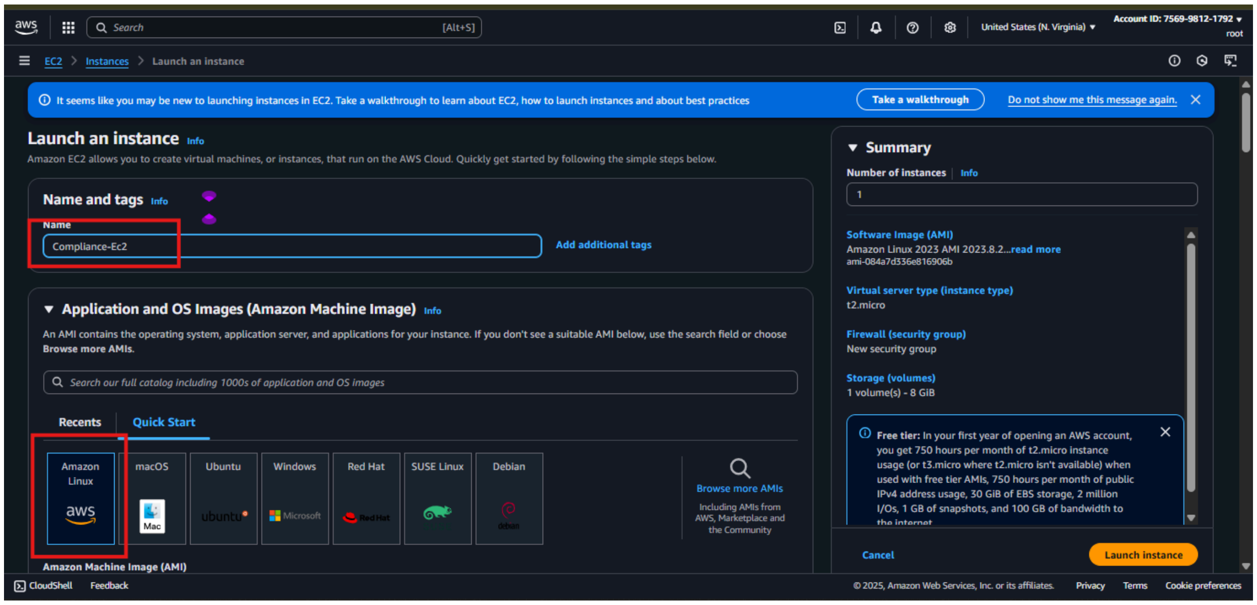
Rule: `ec2-launch-template-imdsv2-check`

EC2 recommends using metadata version 2 you explicitly requires metadat version 1"

We identified that enforcing IMDSv2 is critical to prevent metadata service exploitation. IMDSv1 leaves us open to SSRF-style credential theft.

Launch an EC2 instance using a launch template that does not enforce IMDSv2.

- We launched an EC2 instance from a template with IMDSv1 enabled.


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 7569-9812-1792
root
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

Name: Compliance-Ec2

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
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-084a7d336e816906b

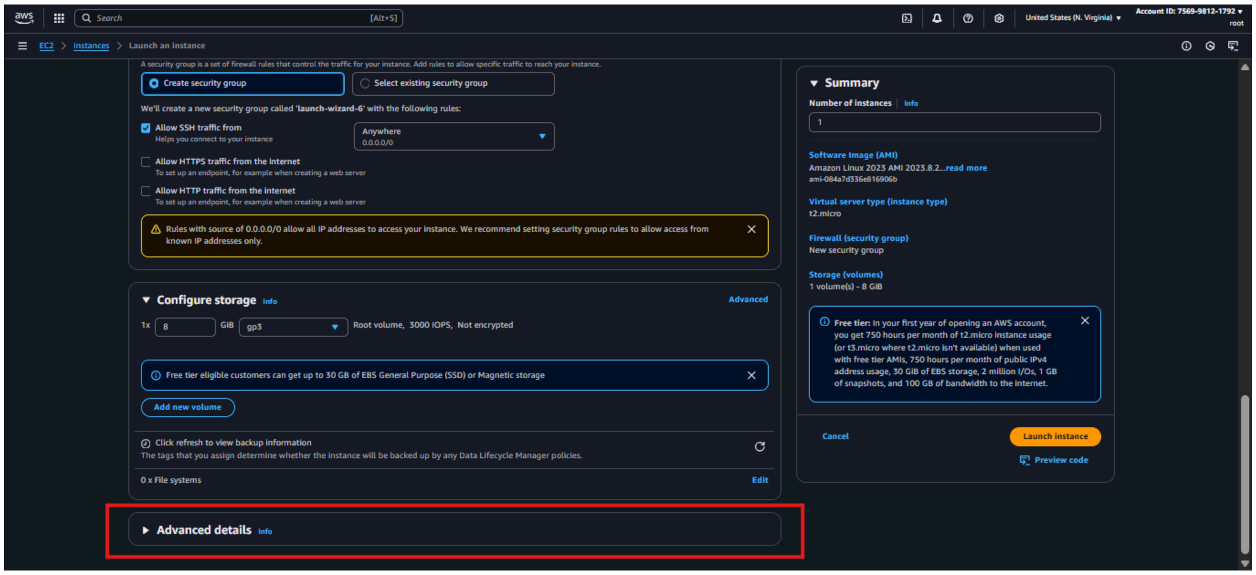
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet

- Click "Advanced Details"


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 7569-9812-1792
root
EC2 > Instances > Launch an instance

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-6' 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

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

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

Summary

Number of instances: 1

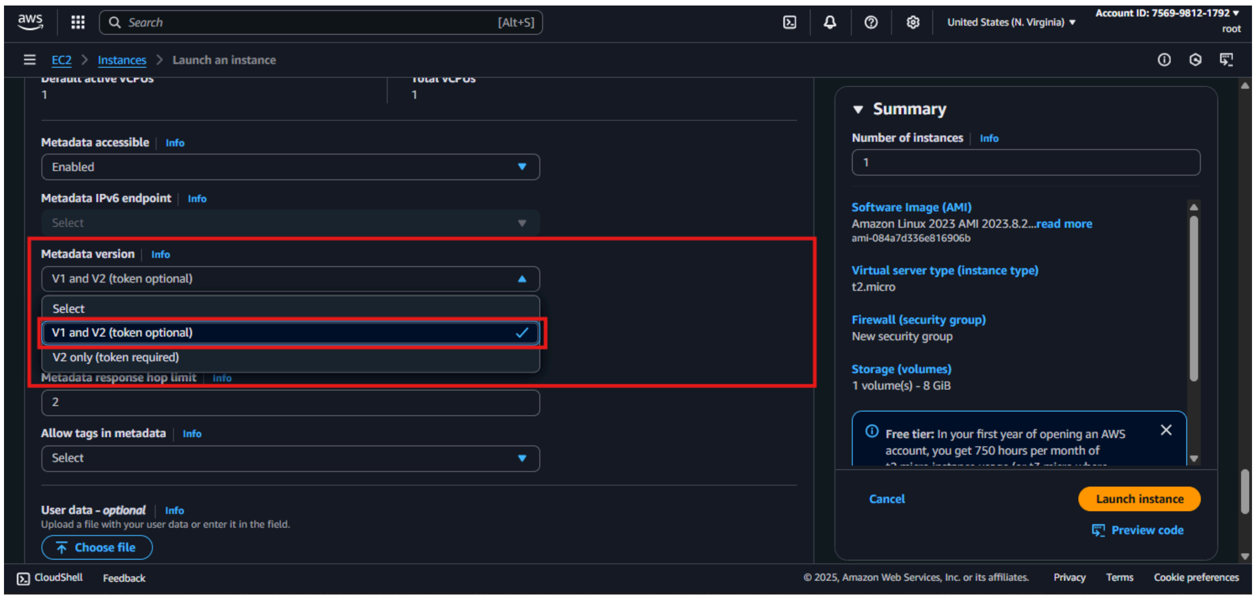
Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-084a7d336e816906b

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 7569-9812-1792
root
EC2 > Instances > Launch an instance
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Default active vCPUs: 1
Total vCPUs: 1

Metadata accessible | Info: Enabled

Metadata IPv6 endpoint | Info: Select

Metadata version | Info: V1 and V2 (token optional)
- Select
- V1 and V2 (token optional) ✓
- V2 only (token required)

Metadata response hop limit | Info: 2

Allow tags in metadata | Info: Select

User data - *optional* | Info
Upload a file with your user data or enter it in the field.
Choose file

Summary

Number of instances | Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-084a7d336e816906b

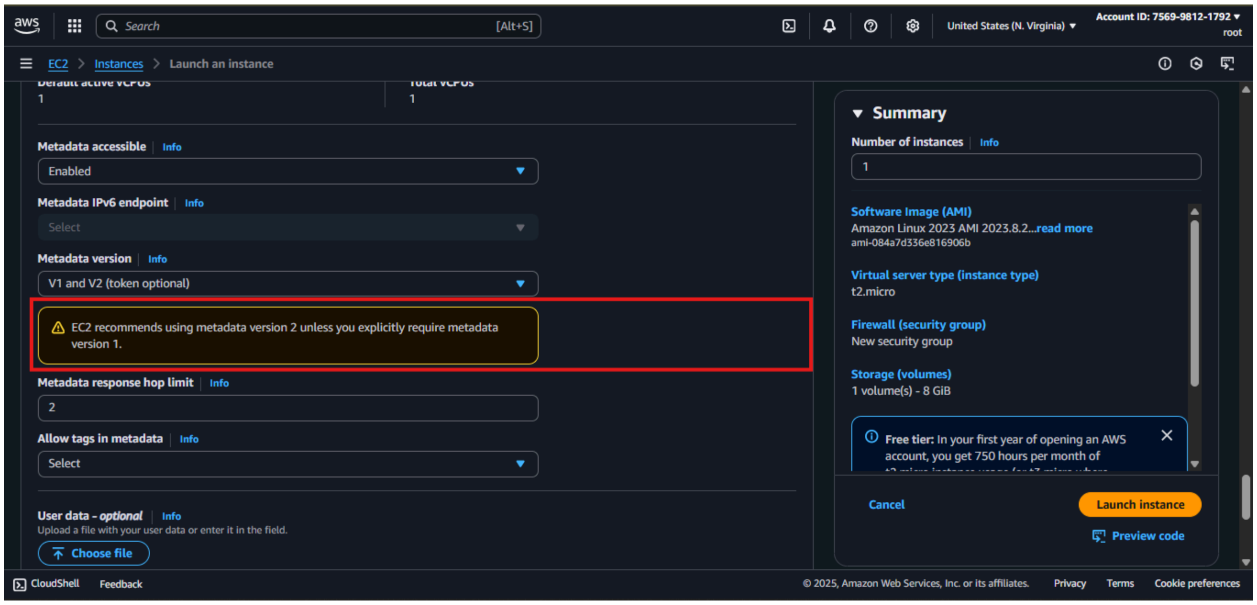
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

- AWS Tried to warn us `EC2 recommends using metadata version 2, unless we explicitly wants the version 1`


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 7569-9812-1792
root
EC2 > Instances > Launch an instance
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Default active vCPUs: 1
Total vCPUs: 1

Metadata accessible | Info: Enabled

Metadata IPv6 endpoint | Info: Select

Metadata version | Info: V1 and V2 (token optional)

⚠ EC2 recommends using metadata version 2 unless you explicitly require metadata version 1.

Metadata response hop limit | Info: 2

Allow tags in metadata | Info: Select

User data - *optional* | Info
Upload a file with your user data or enter it in the field.
Choose file

Summary

Number of instances | Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-084a7d336e816906b

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

- AWS Config flagged the instance as `COMPLIANT`. Which means this rule has an issue. so we created Custom Lambda rule for it using Python code.

> Please Refer to PART 4 and get how this was achieved

Lambda Function Python Code

▼ **Click to view python code**

```
import json
import boto3
import os
import datetime

# Initialize the AWS Config client
CONFIG_CLIENT = boto3.client('config')

def lambda_handler(event, context):
    """
    The main handler for the AWS Lambda function.
    It receives an event from AWS Config and evaluates the compliance of a
n
    EC2 Launch Template by checking if it requires IMDSv2.

    Args:
        event (dict): The event object from AWS Config.
        context (object): The context object for the Lambda function.
    """
    print("Received event: " + json.dumps(event, indent=2))

    # Initializing variables to ensure they exist for the final evaluation
    resource_type = 'Unknown'
    resource_id = 'Unknown'
    ordering_timestamp = str(datetime.datetime.now())
    result_token = ''

    compliance_type = 'NOT_APPLICABLE'
    annotation = 'This resource type is not applicable for this rule.'
    try:
        invoking_event = json.loads(event['invokingEvent'])
        configuration_item = invoking_event['configurationItem']
        result_token = event['resultToken']

        resource_type = configuration_item.get('resourceType')
        resource_id = configuration_item.get('resourceId')
        ordering_timestamp = configuration_item.get('configurationItemCaptu
reTime')

        # Check if the resource is an EC2 Launch Template.
        if resource_type != 'AWS::EC2::LaunchTemplate':
            print(f"Skipping evaluation for non-EC2 Launch Template resource t
ype: {resource_type}")
            compliance_type = 'NOT_APPLICABLE'
            annotation = 'This rule only applies to EC2 Launch Templates.'

        else:
            # --- Evaluate the Launch Template for IMDSv2 requirement ---
            # The launch template data is stored in the 'configuration' key.
            config_data = configuration_item.get('configuration', {})

            # We need to navigate through the LaunchTemplateData structure.
            launch_template_data = config_data.get('LaunchTemplateData', {})
            metadata_options = launch_template_data.get('MetadataOptions',
{})

            # Check if ImdsSupport is set to 'required'.
            imds_support = metadata_options.get('ImdsSupport')

            if imds_support == 'required':
                compliance_type = 'COMPLIANT'
                annotation = "Launch template is compliant: IMDSv2 is required."
            else:
                compliance_type = 'NON_COMPLIANT'
                annotation = "Launch template is non-compliant: IMDSv2 is not r
equired."

    except KeyError as e:
        # Gracefully handle cases where a key is missing from the event.
        print(f"KeyError: Missing key {e} in event. Skipping evaluation for this
malformed event.")
        return {
            'statusCode': 200,
            'body': json.dumps(f"Missing key: {e}")
        }
    except Exception as e:
        # Catch any other unexpected errors and set the compliance status to
NON_COMPLIANT
        # This will provide visibility into the error in your AWS Config dashboa
rd.
        print(f"Error evaluating resource {resource_id}: {e}")
        compliance_type = 'NON_COMPLIANT'
        annotation = f"Failed to evaluate Launch Template: {e}"

    # Submit the evaluation results to AWS Config
    evaluations = [{
        'ComplianceResourceType': resource_type,
        'ComplianceResourceId': resource_id,
        'ComplianceType': compliance_type,
        'Annotation': annotation,
        'OrderingTimestamp': ordering_timestamp
    }]

    try:
        CONFIG_CLIENT.put_evaluations(
            Evaluations=evaluations,
            ResultToken=result_token
        )
        print(f"Submitted evaluation for {resource_id} with compliance status:
{compliance_type}")
```

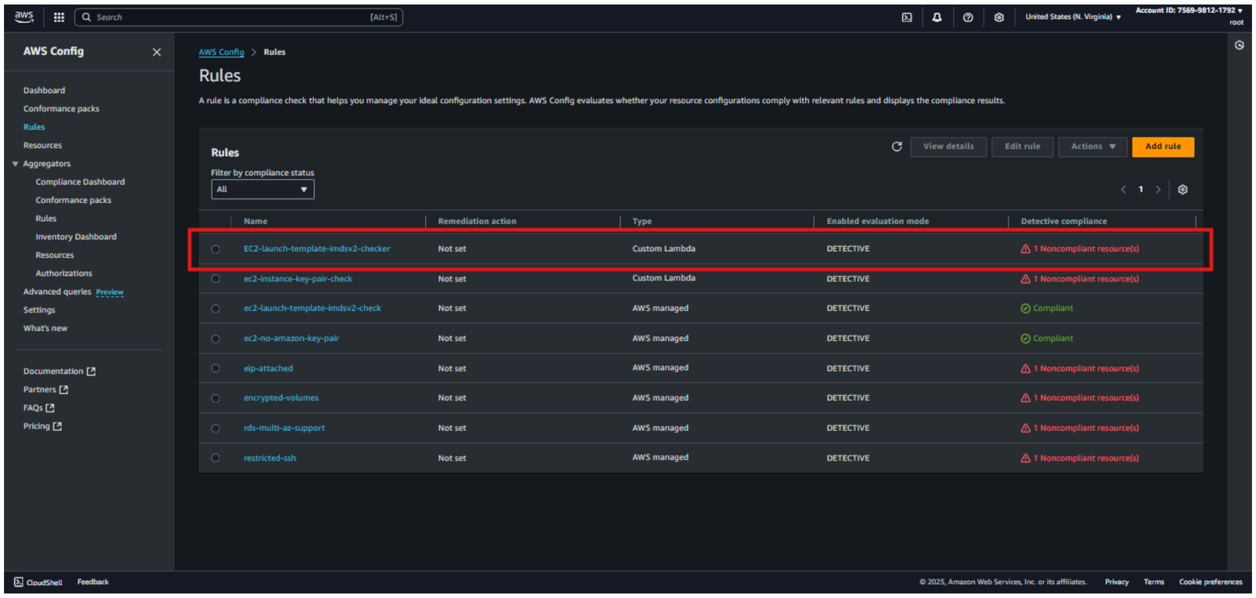
```
    except Exception as e:
        print(f"Error submitting evaluation to AWS Config: {e}")
        # Re-raise the exception to signal an unrecoverable failure
        raise e

    return {
        'statusCode': 200,
        'body': json.dumps('Evaluation submitted successfully')
    }
```

Inline Policy for Permission Role Json Code

▼ Click to view JSON code

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": "config:PutEvaluations",
            "Resource": "*"
        }
    ]
}
```

- So after creating the Lambda Function and applying it to Custom Lambda Rule in AWS Config, It immediatly flagged the instance as `NON-COMPLIANT`.



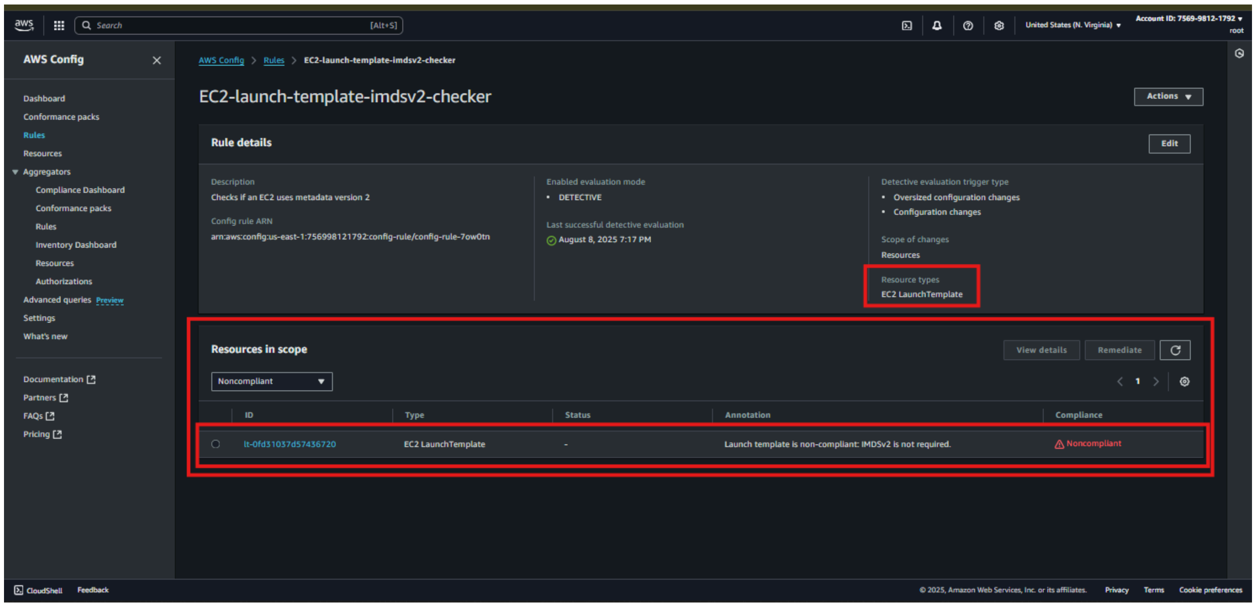
AWS Config > Rules

Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	Compliant
eip-attached	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)



AWS Config > Rules > EC2-launch-template-imdsv2-checker

EC2-launch-template-imdsv2-checker

Rule details

Description: Checks if an EC2 uses metadata version 2

Config rule ARN: arn:aws:config:us-east-1:756998121792:config-rule/config-rule-7ow0tn

Enabled evaluation mode:
- DETECTIVE

Last successful detective evaluation: August 8, 2025 7:17 PM

Detective evaluation trigger type:
- Oversized configuration changes
- Configuration changes

Scope of changes: Resources

Resource types: EC2 LaunchTemplate

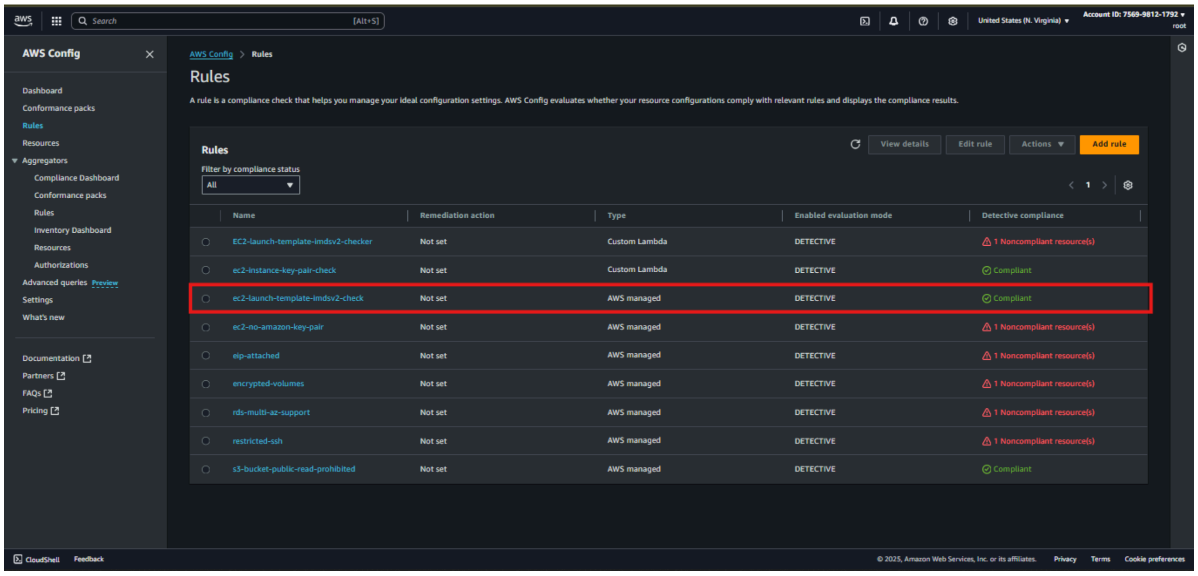
Resources in scope

Noncompliant

ID	Type	Status	Annotation	Compliance
lt-0fd31037d57436720	EC2 LaunchTemplate	-	Launch template is non-compliant: IMDSv2 is not required.	Noncompliant

- We updated the launch template to require IMDSv2, then relaunched the instance using the updated template.

  The instance was marked `COMPLIANT`. This ensures that all new instances follow secure metadata access practices.



AWS Config > Rules

Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	Compliant
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
eip-attached	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
s3-bucket-public-read-prohibited	Not set	AWS managed	DETECTIVE	Compliant

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

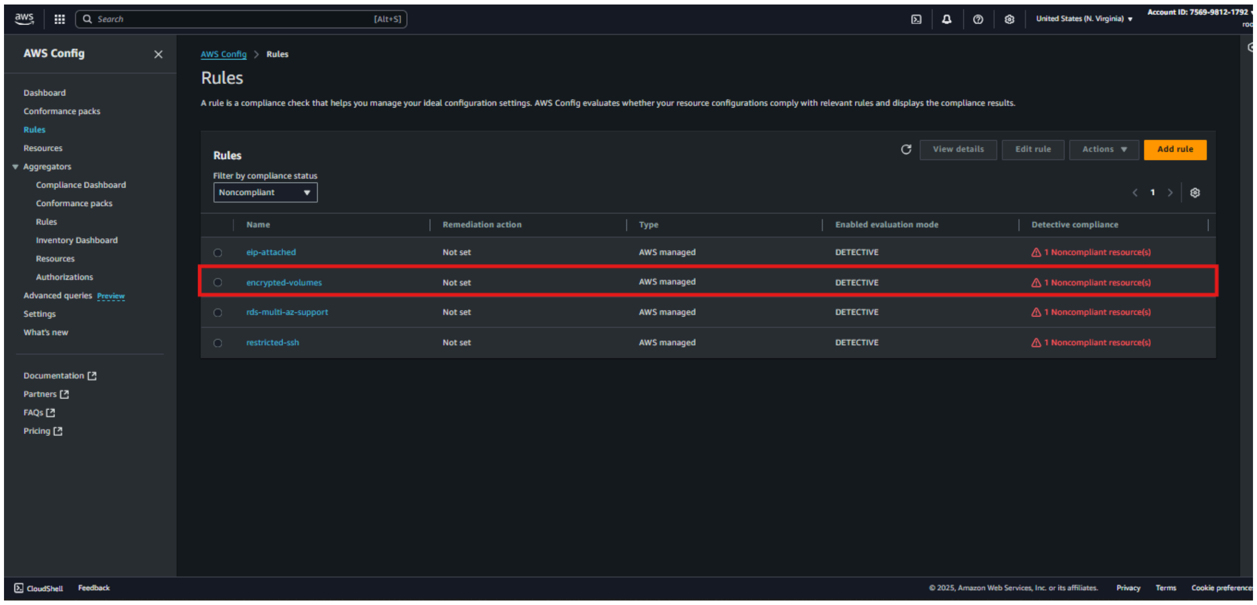
2. EBS Volume Encryption Enforcement

Rule: `encrypted-volumes`

Unencrypted EBS volumes violate our encryption-at-rest policy and introduce compliance risks under standards like HIPAA and PCI-DSS.

We Launched an EC2 instance with an unencrypted root EBS volume.

- We created an EC2 instance with an unencrypted volume.



AWS Config > Rules

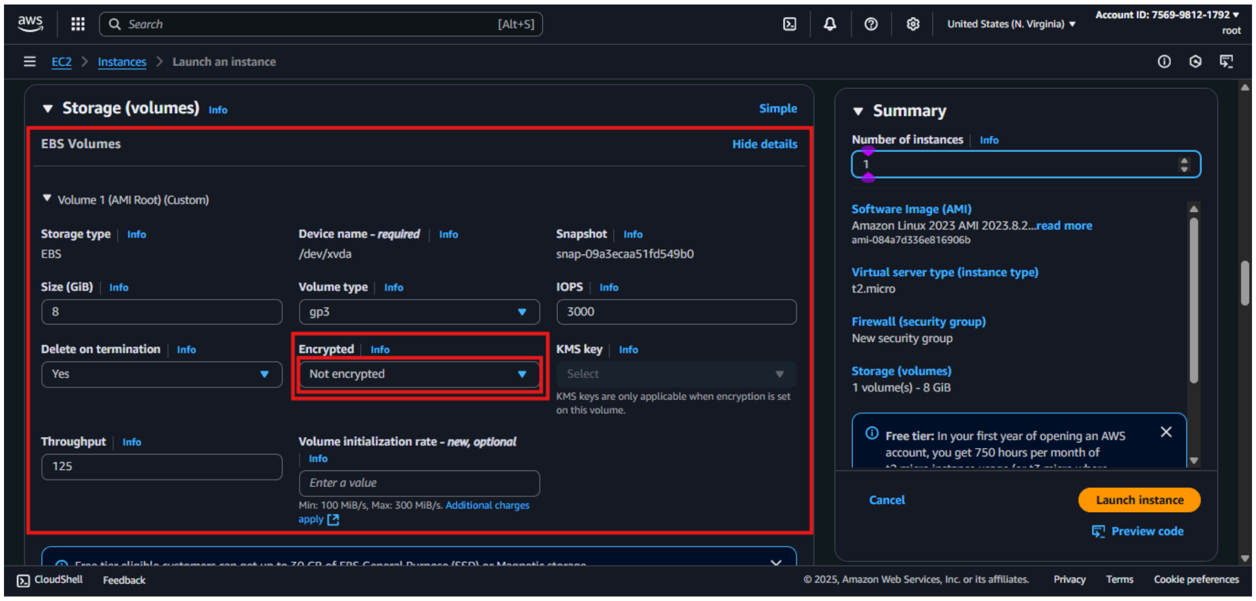
Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: Noncompliant

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
eip-attached	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)



EC2 > Instances > Launch an instance

Storage (volumes)

EBS Volumes

Volume 1 (AMI Root) (Custom)

Storage type: EBS	Device name - *required*: /dev/xvda	Snapshot: snap-09a3ecaa51fd549b0
Size (GiB): 8	Volume type: gp3	IOPS: 3000
Delete on termination: Yes	Encrypted: Not encrypted	KMS key: Select
Throughput: 125	Volume initialization rate - *new, optional*: Enter a value	

KMS keys are only applicable when encryption is set on this volume.

Min: 100 MiB/s, Max: 300 MiB/s. Additional charges apply

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-084a7d336e816906b

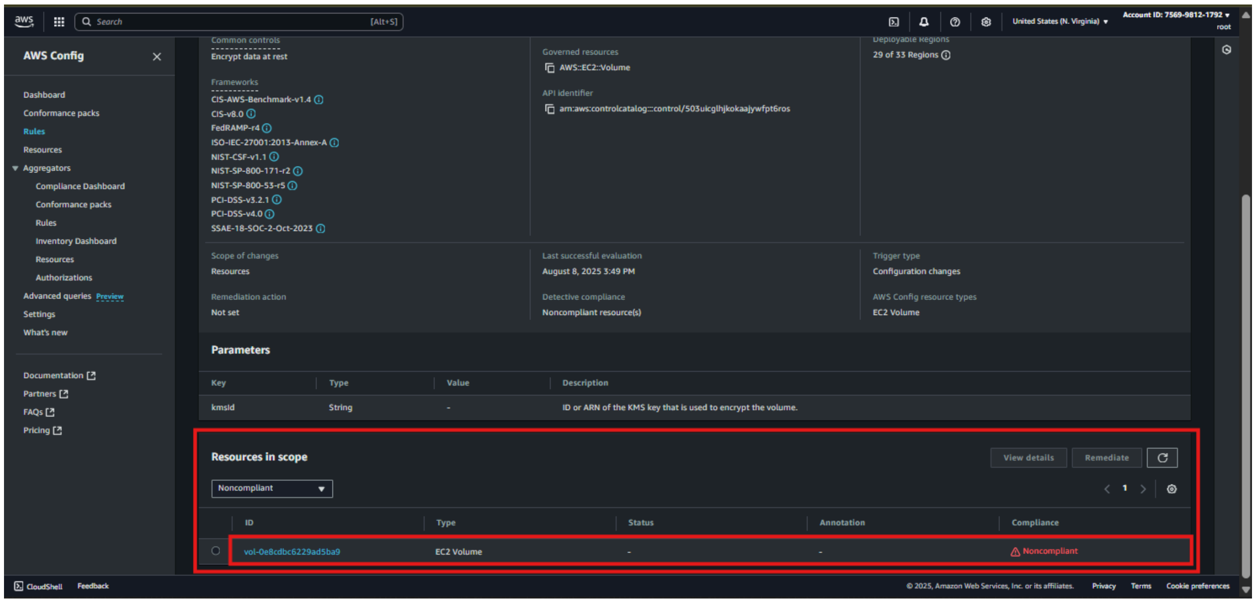
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

- AWS Config immediately flagged it as `NON_COMPLIANT`.



Common controls
Encrypt data at rest

Frameworks
CIS-AWS-Benchmark-v1.4
CIS-v8.0
FedRAMP-r4
ISO-IEC-27001:2013-Annex-A
NIST-CSF-v1.1
NIST-SP-800-171-r2
NIST-SP-800-53-r5
PCI-DSS-v3.2.1
PCI-DSS-v4.0
SSAE-18-SOC-2-Oct-2023

Governed resources
AWS::EC2::Volume

API identifier
arn:aws:controlcatalog:::control/503uicglhjkokaajywfpt6ros

Deployable Regions
29 of 33 Regions

Scope of changes: Resources	Last successful evaluation: August 8, 2025 3:49 PM	Trigger type: Configuration changes
Remediation action: Not set	Detective compliance: Noncompliant resource(s)	AWS Config resource types: EC2 Volume

Parameters

Key	Type	Value	Description
kmsId	String	-	ID or ARN of the KMS key that is used to encrypt the volume.

Resources in scope

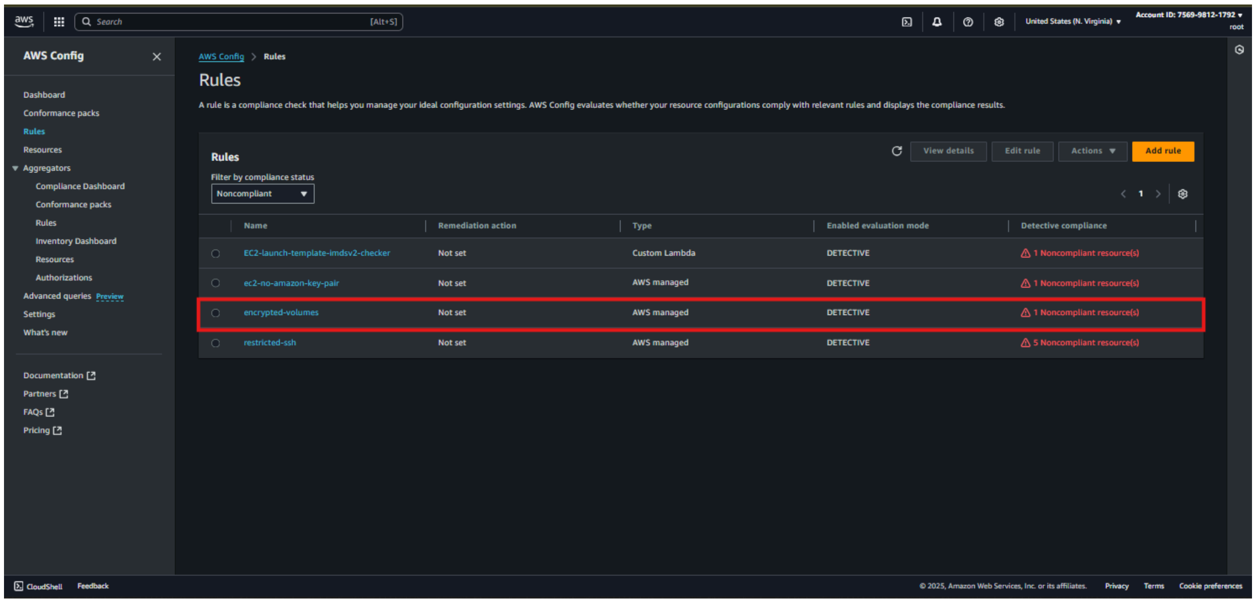
Noncompliant

ID	Type	Status	Annotation	Compliance
vol-0e8cdbc6229ad5ba9	EC2 Volume	-	-	Noncompliant

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- So We created an EC2 instance with an unencrypted volume.
- AWS Config immediately flagged it

AWS Config updated the resource to `COMPLIANT`. We confirmed our ability to enforce encrption on all persistent storage.



AWS Config > Rules

Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: Noncompliant

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	5 Noncompliant resource(s)

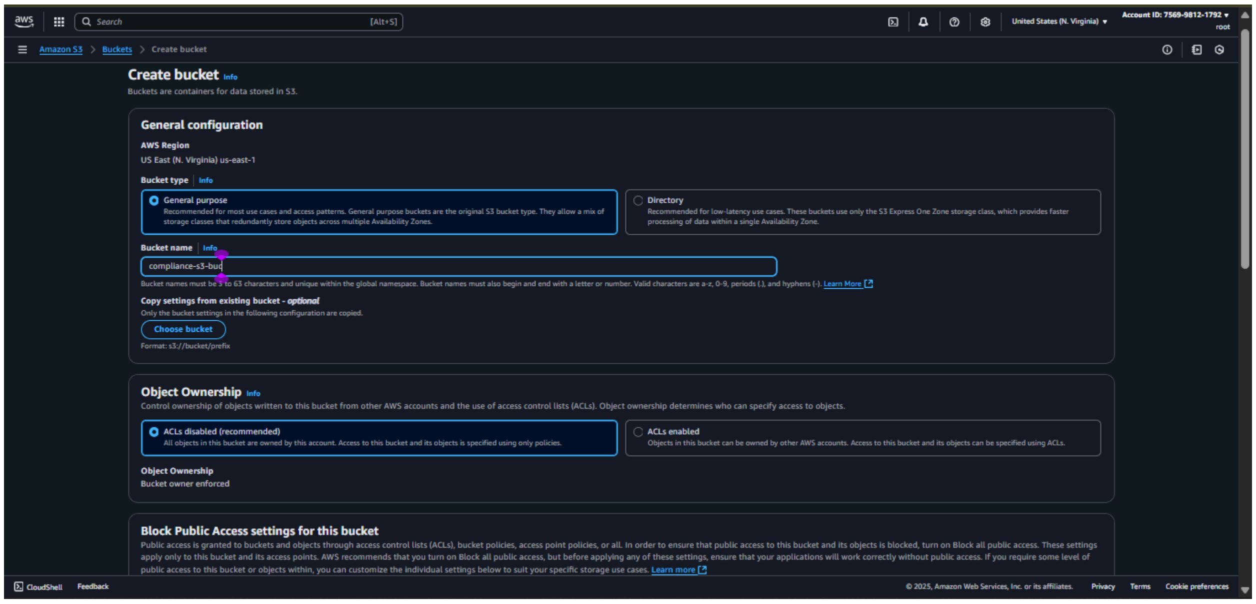
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. S3 Bucket Public Access Control

Rule: `s3-bucket-public-read-prohibited`

Public S3 buckets risk accidental or malicious exposure of sensitive data.

- We created a new S3 bucket and granted public read permissions.



Amazon S3 > Buckets > Create bucket

Account ID: 7569-9812-1792 root · United States (N. Virginia)

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type | Info

- (●) General purpose — Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
- ( ) Directory — Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name | Info
compliance-s3-buc

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

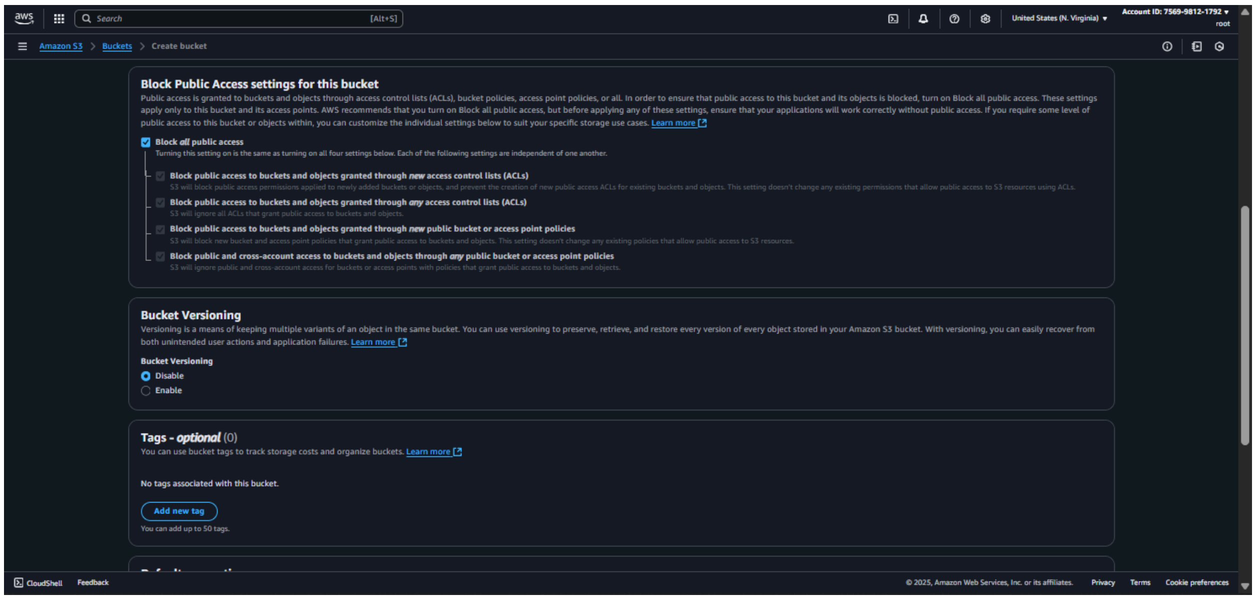
- (●) ACLs disabled (recommended) — All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
- ( ) ACLs enabled — Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

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



Amazon S3 > Buckets > Create bucket

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

- [✓] Block *all* public access
  Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
  - [✓] Block public access to buckets and objects granted through *new* access control lists (ACLs)
    S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - [✓] Block public access to buckets and objects granted through *any* access control lists (ACLs)
    S3 will ignore all ACLs that grant public access to buckets and objects.
  - [✓] Block public access to buckets and objects granted through *new* public bucket or access point policies
    S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - [✓] Block public and cross-account access to buckets and objects through *any* public bucket or access point policies
    S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
- (●) Disable
- ( ) Enable

Tags - *optional* (0)

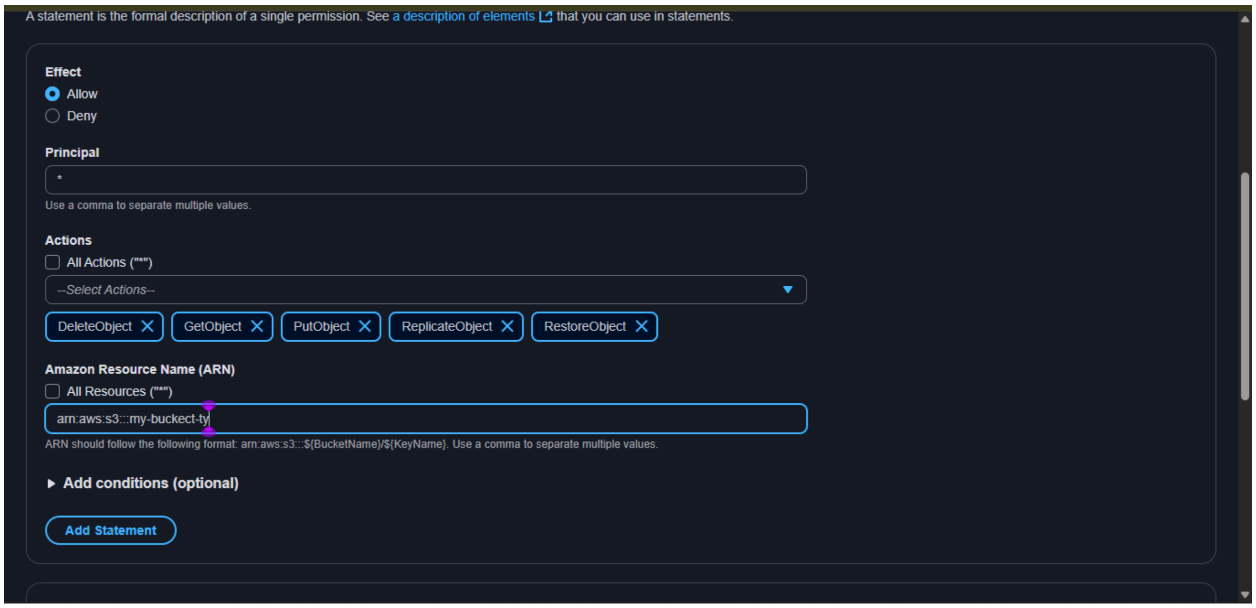
You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Add new tag
You can add up to 50 tags.

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

- Created Policy for our S3 using policy generator



A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect
Allow [✓]
Deny

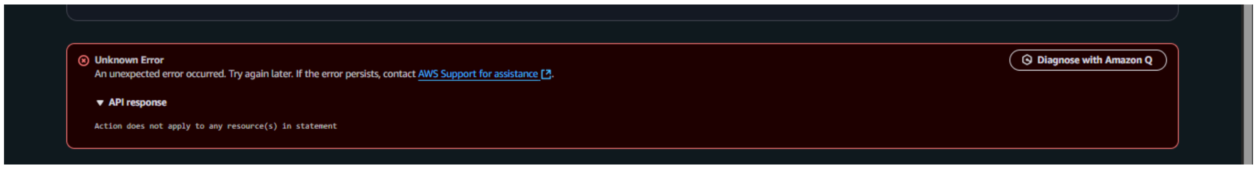
Principal
*
Use a comma to separate multiple values.

Actions
All Actions ("*")
--Select Actions--
DeleteObject GetObject PutObject ReplicateObject RestoreObject

Amazon Resource Name (ARN)
All Resources ("*")
arn:aws:s3:::my-buckect-ty
ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}. Use a comma to separate multiple values.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ReplicateObject",
                "s3:RestoreObject"
            ],
            "Resource": "arn:aws:s3:::my-buckect-ty/*"
        }
    ]
}
```

So we got issue here because of trying to add the S3 buckect as the direct resource,



Unknown Error

An unexpected error occurred. Try again later. If the error persists, contact AWS Support for assistance.

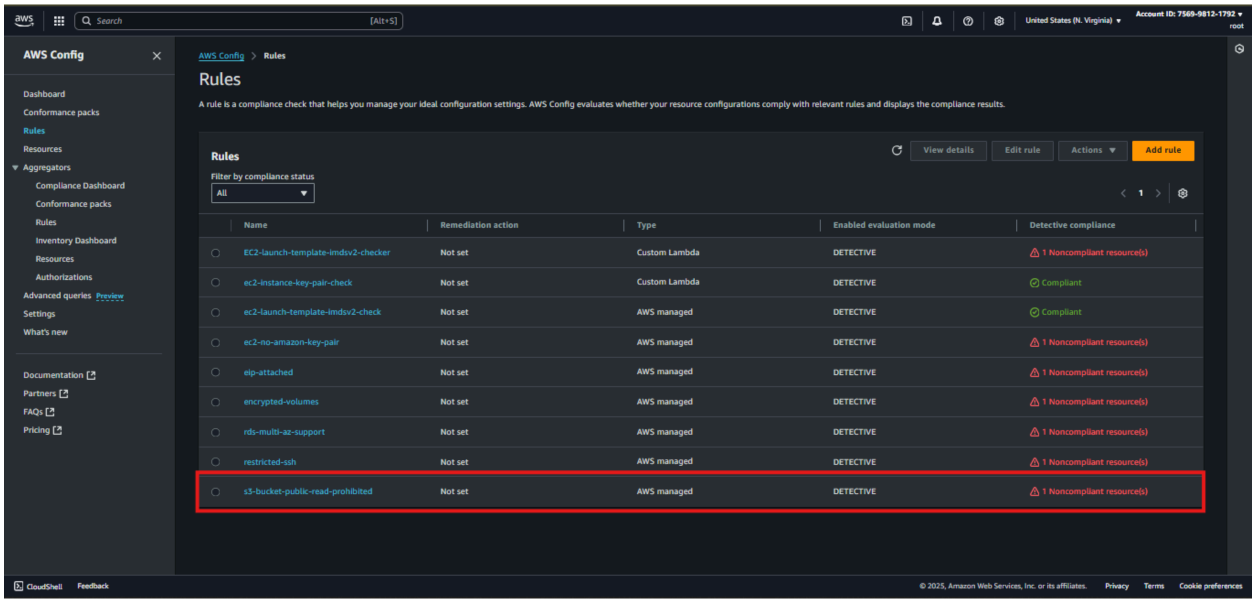
API response

```
Action does not apply to any resource(s) in statement
```

> S3 Buckect itself can’t be assesed as a rosource but what inside

> So we added “/*” at the end of the ARN of the S3 bucket and the error was raised off

- AWS Config detected and marked it as `NON-OMPLIANT`.



Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

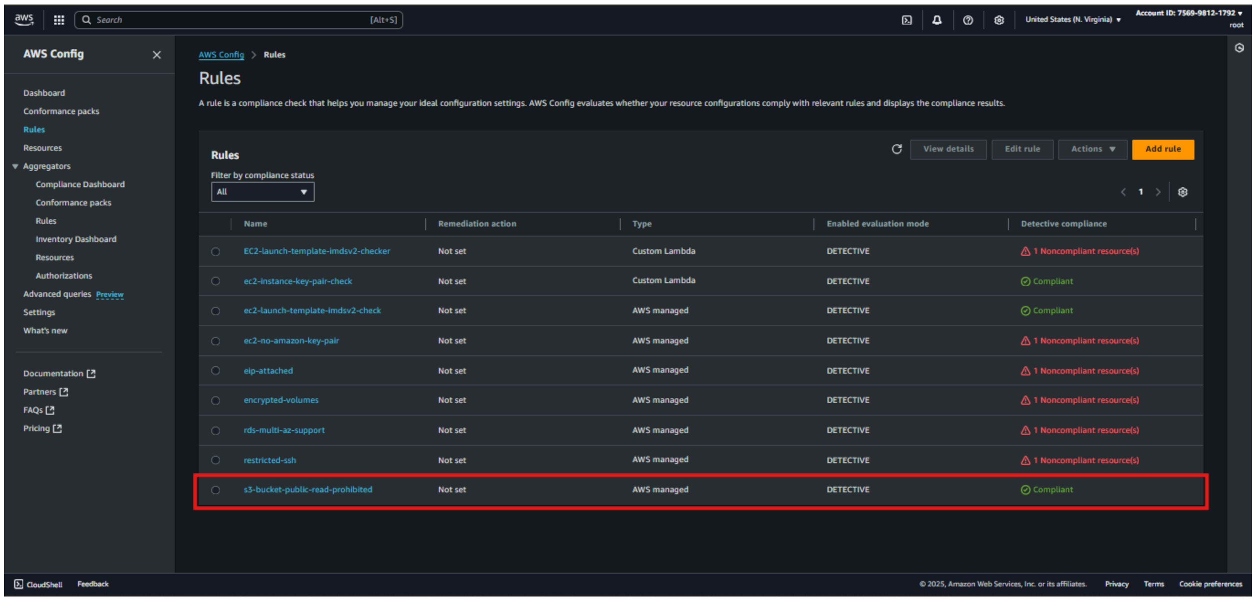
Rules

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	Compliant
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
eip-attached	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
s3-bucket-public-read-prohibited	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)

- We blocked public access using the S3 **Block Public Access** feature and removed all public ACLs.

AWS Config marked the bucket as `COMPLIANT`. This reinforced our access control posture for object storage.



AWS Config > Rules

Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	Compliant
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
eip-attached	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
s3-bucket-public-read-prohibited	Not set	AWS managed	DETECTIVE	Compliant

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

---

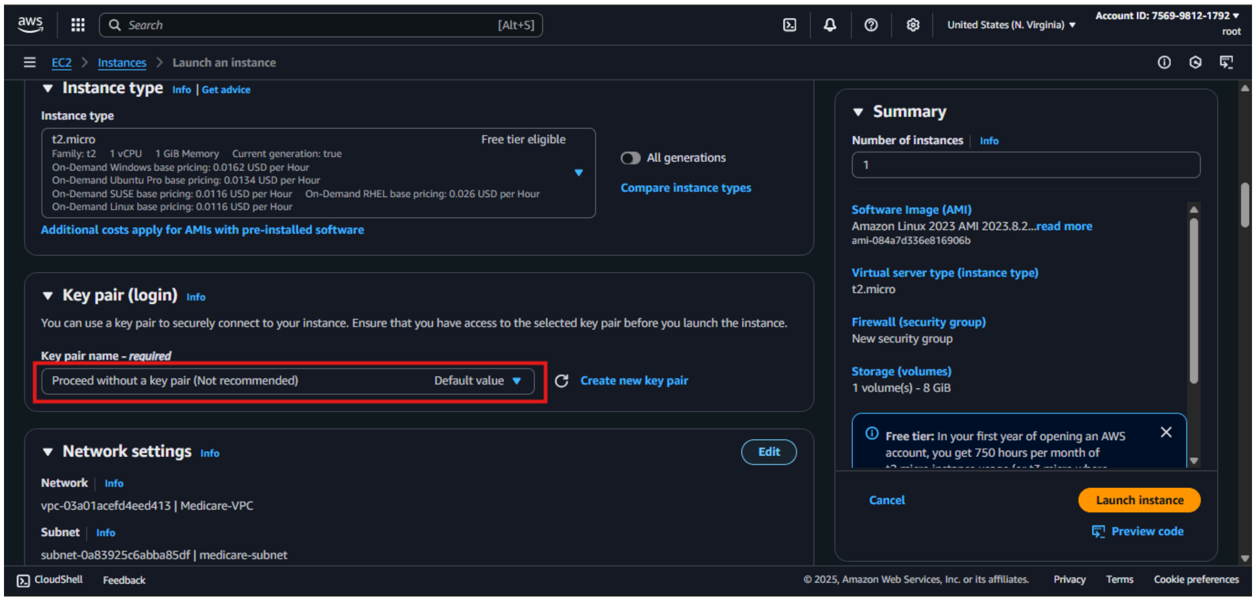
4. EC2 Without Key Pair

Rule: `ec2-no-amazon-key-pair`

Launching Instances without key pairs prevent secure SSH access and risk operational lockouts.

We Launched an EC2 instance without associating a key pair.

- We launched the instance without any key pair.



Instance type

Instance type: t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: Proceed without a key pair (Not recommended) — Default value

Network settings

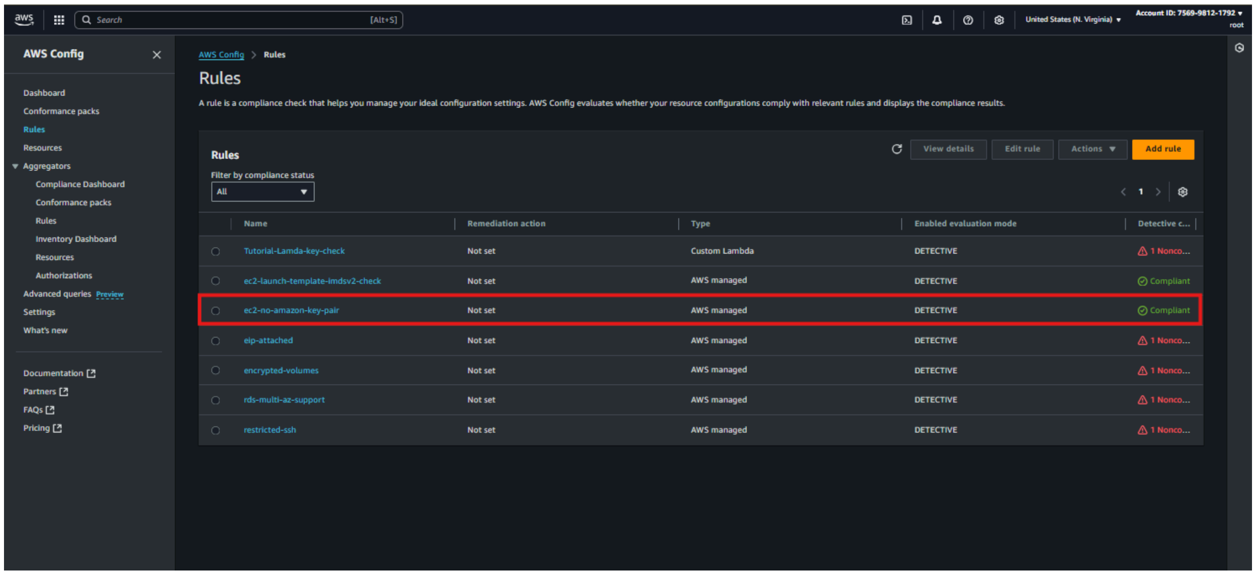
Network: vpc-03a01acefd4eed413 | Medicare-VPC
Subnet: subnet-0a83925c6abba85df | medicare-subnet

Summary

Number of instances: 1
Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more — ami-084a7d336e816906b
Virtual server type (instance type): t2.micro
Firewall (security group): New security group
Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

- AWS Config flagged the resource as `COMPLIANT` . We noticed that the rule said it `"Checks if running Amazon Elastic Compute Cloud (EC2) instances are launched using amazon key pairs. The rule is NON_COMPLIANT if a running EC2 instance is launched with a key pair.`



Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

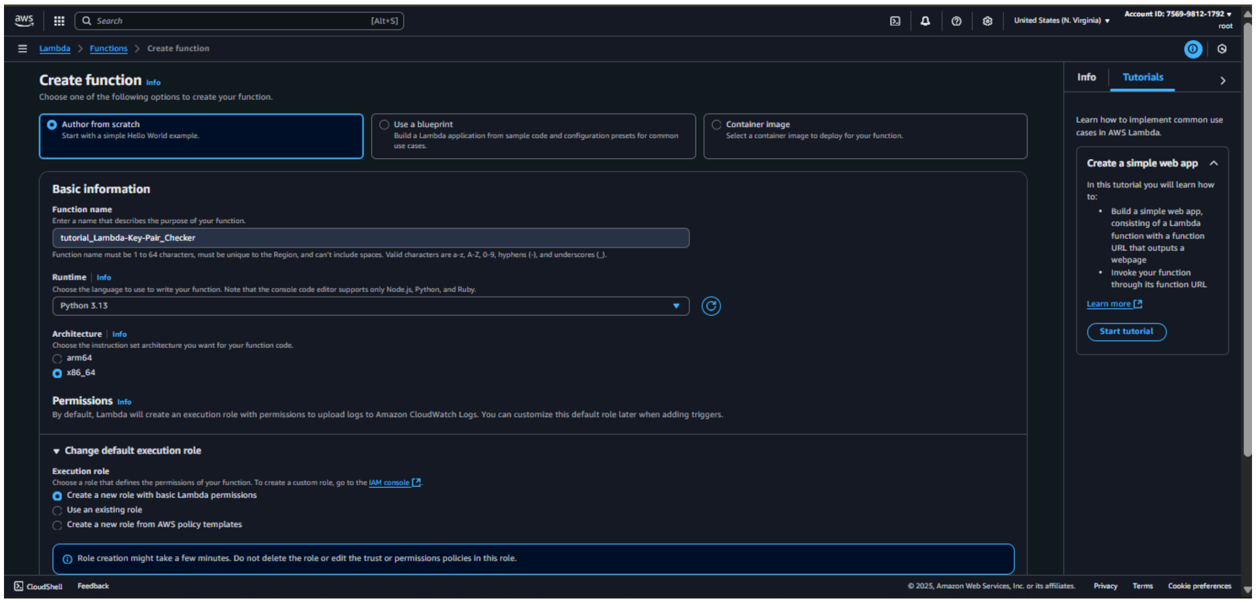
Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective c...
Tutorial-Lamda-key-check	Not set	Custom Lambda	DETECTIVE	1 Nonco...
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	Compliant
eip-attached	Not set	AWS managed	DETECTIVE	1 Nonco...
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Nonco...
rds-multi-az-support	Not set	AWS managed	DETECTIVE	1 Nonco...
restricted-ssh	Not set	AWS managed	DETECTIVE	1 Nonco...

- But this is very confusing and if by chance another people is to be in our position. They not find out about the rule description soon.

- So we tried to create a new rule using lambda and IAM role.

- created a function and set the "**Change default execution role"** to Create a new role with basic Lambda permissions


Search [Alt+S]
United States (N. Virginia)
Account ID: 7569-9812-1792
root
Lambda > Functions > Create function
Info | Tutorials

Create function Info

Choose one of the following options to create your function.

- [✓ Author from scratch] Start with a simple Hello World example.
- Use a blueprint — Build a Lambda application from sample code and configuration presets for common use cases.
- Container image — Select a container image to deploy for your function.

Basic information

Function name: tutorial_Lambda-Key-Pair_Checker

Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime: Python 3.13

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture: arm64 [✓ x86_64]

Choose the instruction set architecture you want for your function code.

Permissions

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

- [✓ Create a new role with basic Lambda permissions]
- Use an existing role
- Create a new role from AWS policy templates

Role creation might take a few minutes. Do not delete the role or edit the trust or permissions policies in this role.

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial

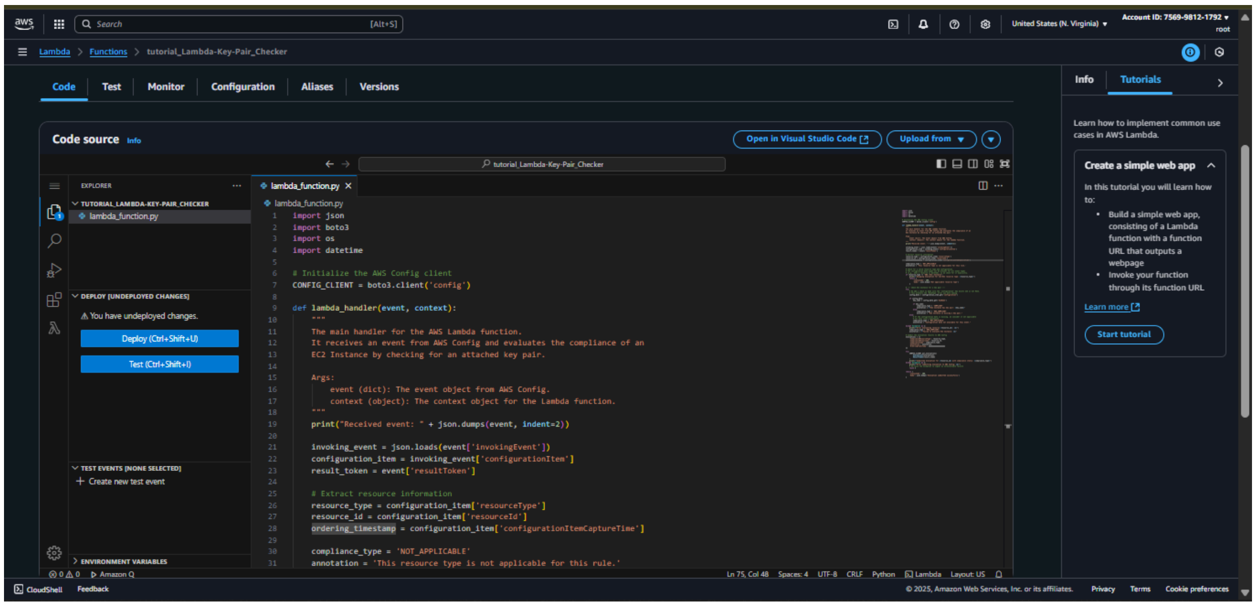
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- open the new lambda function and inserted the python code which we wrote basically to obtain

resource properties of our EC2 instance which our target was "key-pair".

▼ Click to view Python code

```
import json
import boto3
import os
import datetime

# Initialize the AWS Config client
CONFIG_CLIENT = boto3.client('config')

def lambda_handler(event, context):
    """
    The main handler for the AWS Lambda function.
    It receives an event from AWS Config and evaluates the compliance of a
n
    EC2 Instance by checking for an attached key pair.

    Args:
        event (dict): The event object from AWS Config.
        context (object): The context object for the Lambda function.
    """
    print("Received event: " + json.dumps(event, indent=2))

    invoking_event = json.loads(event['invokingEvent'])
    configuration_item = invoking_event['configurationItem']
    result_token = event['resultToken']

    # Extract resource information
    resource_type = configuration_item['resourceType']
    resource_id = configuration_item['resourceId']
    ordering_timestamp = configuration_item['configurationItemCaptureTim
e']

    compliance_type = 'NOT_APPLICABLE'
    annotation = 'This resource type is not applicable for this rule.'

    # Check for a valid resource type and configuration.
    # The configuration data might not be present for all event types
    # (e.g., when a resource is deleted), so we check for it explicitly.
    if resource_type != 'AWS::EC2::Instance':
        print(f"Skipping evaluation for non-EC2 resource type: {resource_typ
e}")
        return {
            'statusCode': 200,
            'body': json.dumps('Not applicable resource type')
        }
    # --- Check EC2 Instance for a key pair ---
    try:
        # We add a check to make sure the 'configuration' key exists and is n
ot None.
        # This prevents the 'NoneType' error you were seeing.
        config_data = configuration_item.get('configuration')

        if config_data:
            key_name = config_data.get('keyName')

            if key_name:
                compliance_type = 'COMPLIANT'
                annotation = f"EC2 Instance has key pair: {key_name}"
            else:
                compliance_type = 'NON_COMPLIANT'
                annotation = "EC2 Instance is missing a key pair."
        else:
            # If the configuration data is missing, we consider it not applicable
            # for this rule's evaluation.
            compliance_type = 'NOT_APPLICABLE'
            annotation = "Configuration data not available for this event."

    except Exception as e:
        print(f"Error evaluating resource {resource_id}: {e}")
        compliance_type = 'NON_COMPLIANT'
        annotation = f"Failed to evaluate EC2 Instance: {e}"

    # Submit the evaluation results to AWS Config
    evaluations = [{
        'ComplianceResourceType': resource_type,
        'ComplianceResourceId': resource_id,
        'ComplianceType': compliance_type,
        'Annotation': annotation,
        'OrderingTimestamp': ordering_timestamp
    }]
    try:
        CONFIG_CLIENT.put_evaluations(
            Evaluations=evaluations,
            ResultToken=result_token
        )
        print(f"Submitted evaluation for {resource_id} with compliance status: {compliance_type}")
    except Exception as e:
        print(f"Error submitting evaluation to AWS Config: {e}")
        # Re-raise the exception to signal an unrecoverable failure
        raise e

    return {
        'statusCode': 200,
        'body': json.dumps('Evaluation submitted successfully')
    }
```

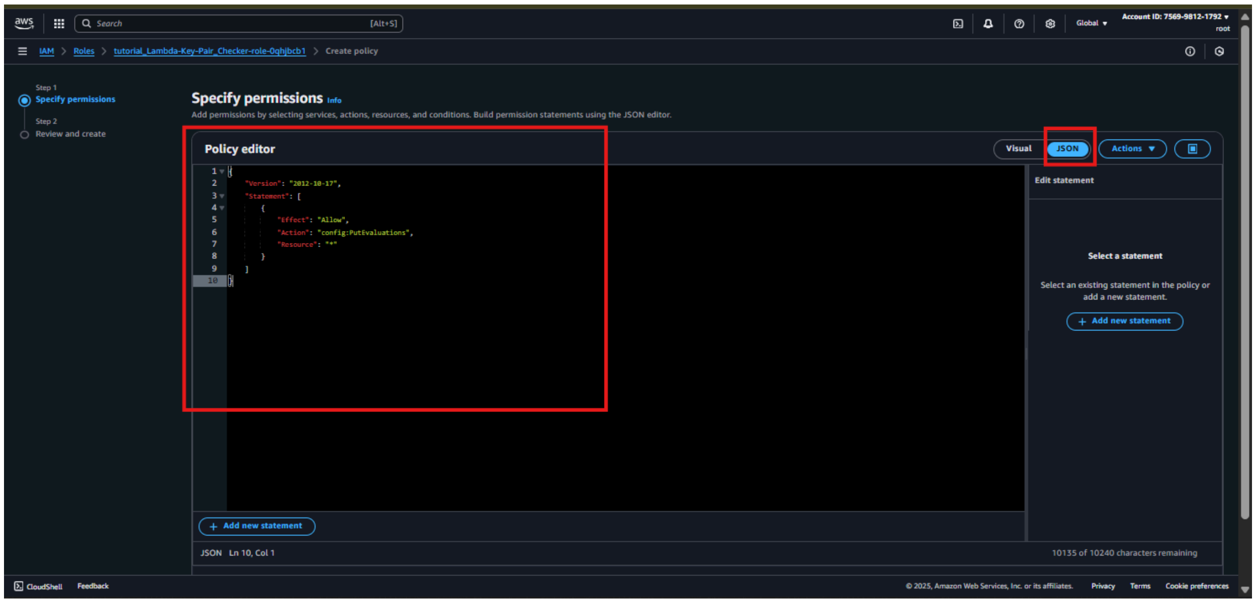


```
import json
import boto3
import os
import datetime

# Initialize the AWS Config client
CONFIG_CLIENT = boto3.client('config')

def lambda_handler(event, context):
    """
    The main handler for the AWS Lambda function.
    It receives an event from AWS Config and evaluates the compliance of an
    EC2 Instance by checking for an attached key pair.

    Args:
        event (dict): The event object from AWS Config.
        context (object): The context object for the Lambda function.
    """
    print("Received event: " + json.dumps(event, indent=2))

    invoking_event = json.loads(event['invokingEvent'])
    configuration_item = invoking_event['configurationItem']
    result_token = event['resultToken']

    # Extract resource information
    resource_type = configuration_item['resourceType']
    resource_id = configuration_item['resourceId']
    ordering_timestamp = configuration_item['configurationItemCaptureTime']

    compliance_type = 'NOT_APPLICABLE'
    annotation = 'This resource type is not applicable for this rule.'
```

- Then We Attached an inline policy to the new role created by the lambda function using JSON code

▼ Click to view JSON code

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "config:PutEvaluations",
         "Resource": "*"
      }
   ]
}
```


IAM > Roles > tutorial_Lambda-Key-Pair_Checker-role-0qhjbcb1 > 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.

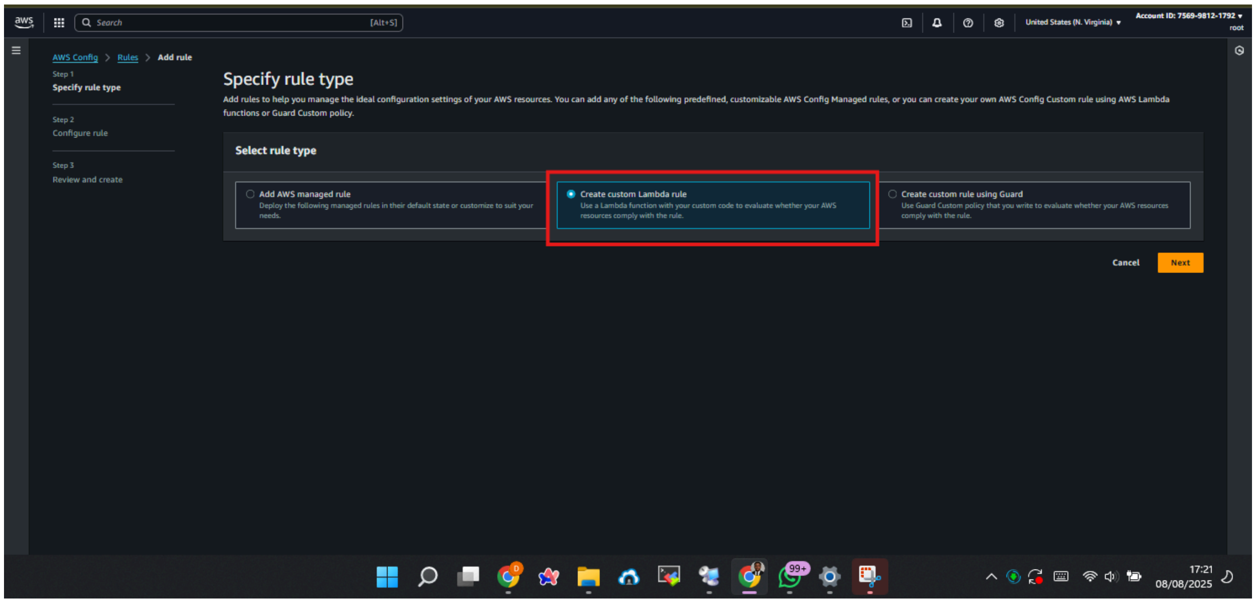
Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "config:PutEvaluations",
            "Resource": "*"
        }
    ]
}
```

JSON Ln 10, Col 1

10135 of 10240 characters remaining

- Created a new Rule using "Create custom Lambda rule" Option



AWS Config > Rules > Add rule

Step 1
Specify rule type

Step 2
Configure rule

Step 3
Review and create

Specify rule type

Add rules to help you manage the ideal configuration settings of your AWS resources. You can add any of the following predefined, customizable AWS Config Managed rules, or you can create your own AWS Config Custom rule using AWS Lambda functions or Guard Custom policy.

Select rule type

- Add AWS managed rule — Deploy the following managed rules in their default state or customize to suit your needs.
- [✓ Create custom Lambda rule] — Use a Lambda function with your custom code to evaluate whether your AWS resources comply with the rule.
- Create custom rule using Guard — Use Guard Custom policy that you write to evaluate whether your AWS resources comply with the rule.

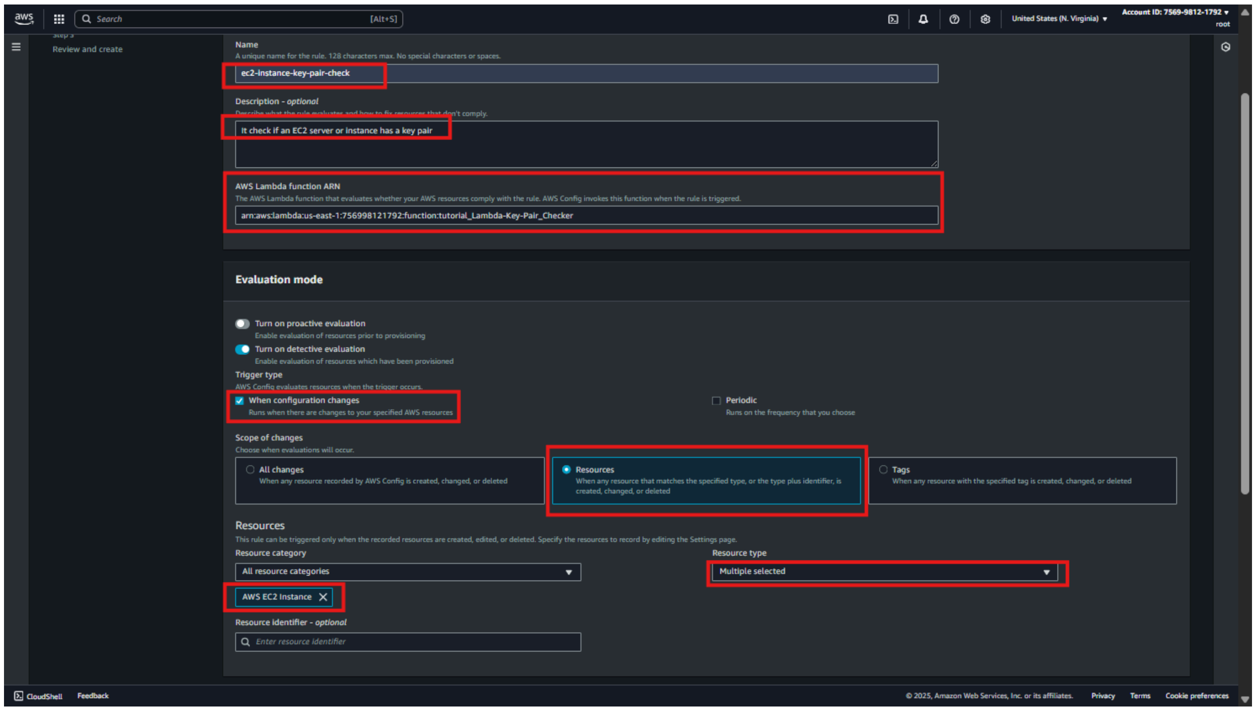
We named our rule

Added a description

Added the ARN of our Lambda Function

Selected trigger type

And then choosed resource type the Lambda rule should to fetch, which is “AWS EC2 instance”


Name
A unique name for the rule. 128 characters max. No special characters or spaces.
ec2-instance-key-pair-check

Description - *optional*
Describe what the rule evaluates and how to fix resources that don't comply.
It check if an EC2 server or instance has a key pair

AWS Lambda function ARN
The AWS Lambda function that evaluates whether your AWS resources comply with the rule. AWS Config invokes this function when the rule is triggered.
arn:aws:lambda:us-east-1:756998121792:function:tutorial_Lambda-Key-Pair_Checker

Evaluation mode

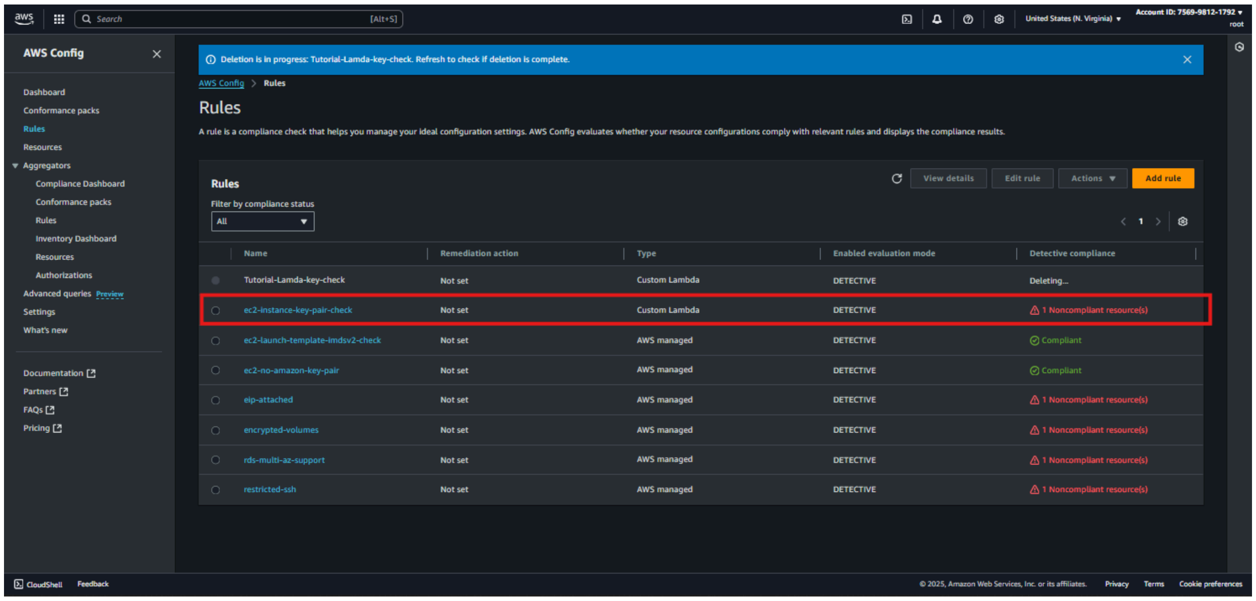
Turn on proactive evaluation
Enable evaluation of resources prior to provisioning
Turn on detective evaluation
Enable evaluation of resources which have been provisioned

Trigger type
AWS Config evaluates resources when the trigger occurs.
[✓ When configuration changes] — Runs when there are changes to your specified AWS resources
Periodic — Runs on the frequency that you choose

Scope of changes
Choose when evaluations will occur.
All changes — When any resource recorded by AWS Config is created, changed, or deleted
[✓ Resources] — When any resource that matches the specified type, or the type plus identifier, is created, changed, or deleted
Tags — When any resource with the specified tag is created, changed, or deleted

Resources
This rule can be triggered only when the recorded resources are created, edited, or deleted. Specify the resources to record by editing the Settings page.
Resource category: All resource categories
Resource type: Multiple selected
AWS EC2 Instance
Resource identifier - *optional*

- The Lambda AWS Config Rule flagged the resource as NON-COMPLIANT

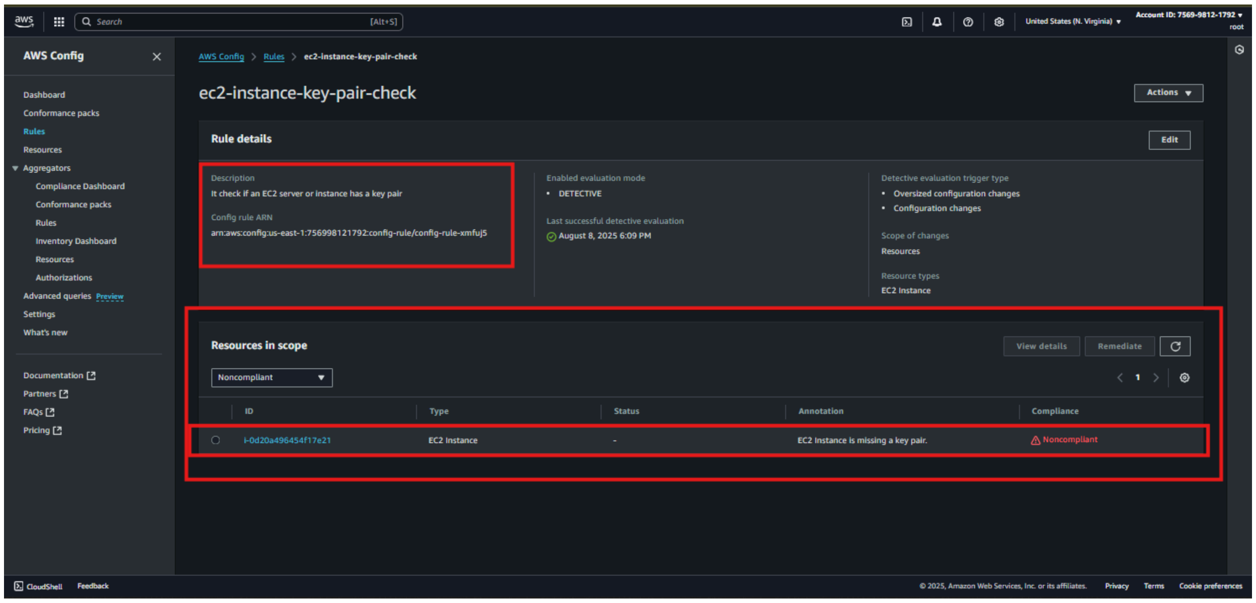

Deletion is in progress: Tutorial-Lamda-key-check. Refresh to check if deletion is complete.

Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
Tutorial-Lamda-key-check	Not set	Custom Lambda	DETECTIVE	Deleting...
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	Compliant
eip-attached	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)


AWS Config > Rules > ec2-instance-key-pair-check

ec2-instance-key-pair-check

Rule details

Description: It check if an EC2 server or instance has a key pair

Config rule ARN: arn:aws:config:us-east-1:756998121792:config-rule/config-rule-xmfuj5

Enabled evaluation mode:
- DETECTIVE

Last successful detective evaluation: August 8, 2025 6:09 PM

Detective evaluation trigger type:
- Oversized configuration changes
- Configuration changes

Scope of changes: Resources

Resource types: EC2 Instance

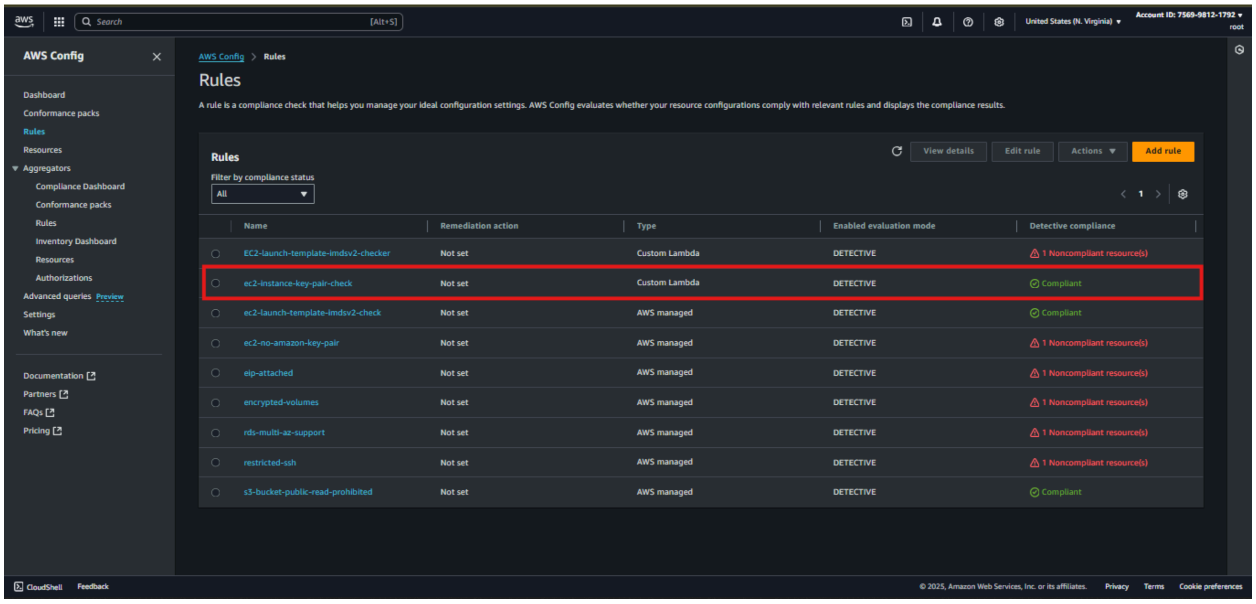
Resources in scope

Noncompliant

ID	Type	Status	Annotation	Compliance
i-0d20a496454f17e21	EC2 Instance	-	EC2 Instance is missing a key pair.	Noncompliant

- So We terminated and relaunched the instance with a valid key pair attached.

AWS Config marked the instance as `COMPLIANT`, ensuring operational access security.


AWS Config > Rules

Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	Compliant
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
eip-attached	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
s3-bucket-public-read-prohibited	Not set	AWS managed	DETECTIVE	Compliant

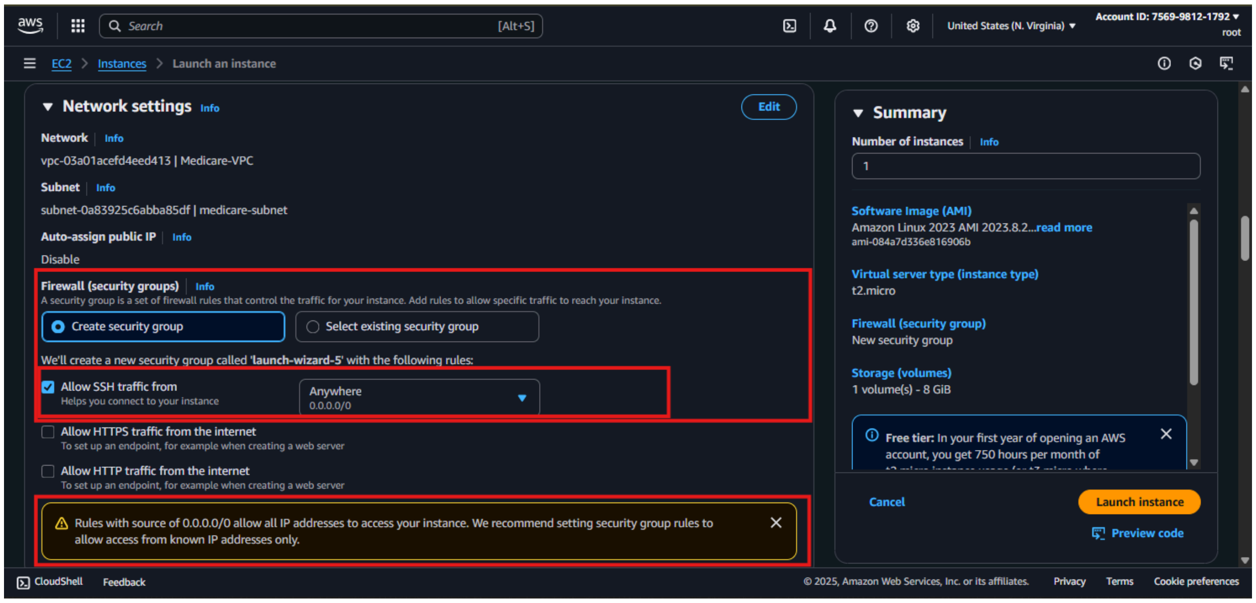
5. Public SSH Restriction

Rule: `restricted-ssh`

Allowing SSH from `0.0.0.0/0` exposes instances to potential brute-force attacks.

Open SSH access to all IPs.

- We updated a Security Group to allow inbound port 22 from all IPs.



Search [Alt+S] United States (N. Virginia) ▾ Account ID: 7569-9812-1792 ▾ root

EC2 > Instances > Launch an instance

Network settings Info Edit

Network Info
vpc-03a01acefd4eed413 | Medicare-VPC

Subnet Info
subnet-0a83925c6abba85df | medicare-subnet

Auto-assign public IP Info
Disable

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-5**' 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.

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-084a7d336e816906b

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

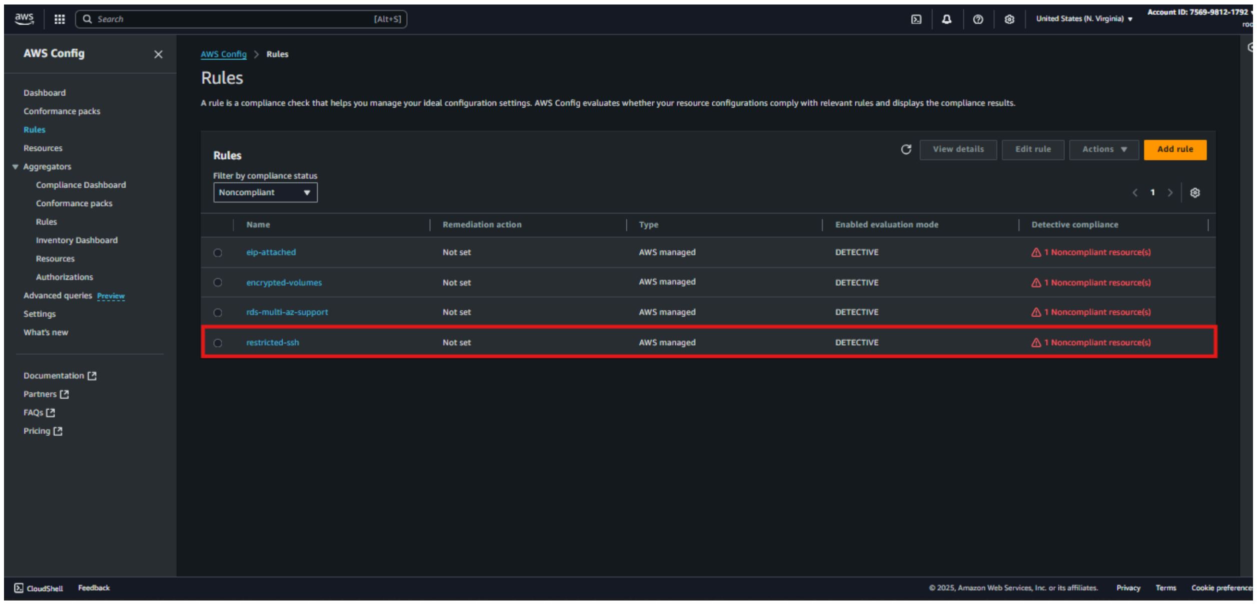
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

Cancel Launch instance Preview code

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- AWS Config flagged it as `NON_COMPLIANT`.



AWS Config > Rules

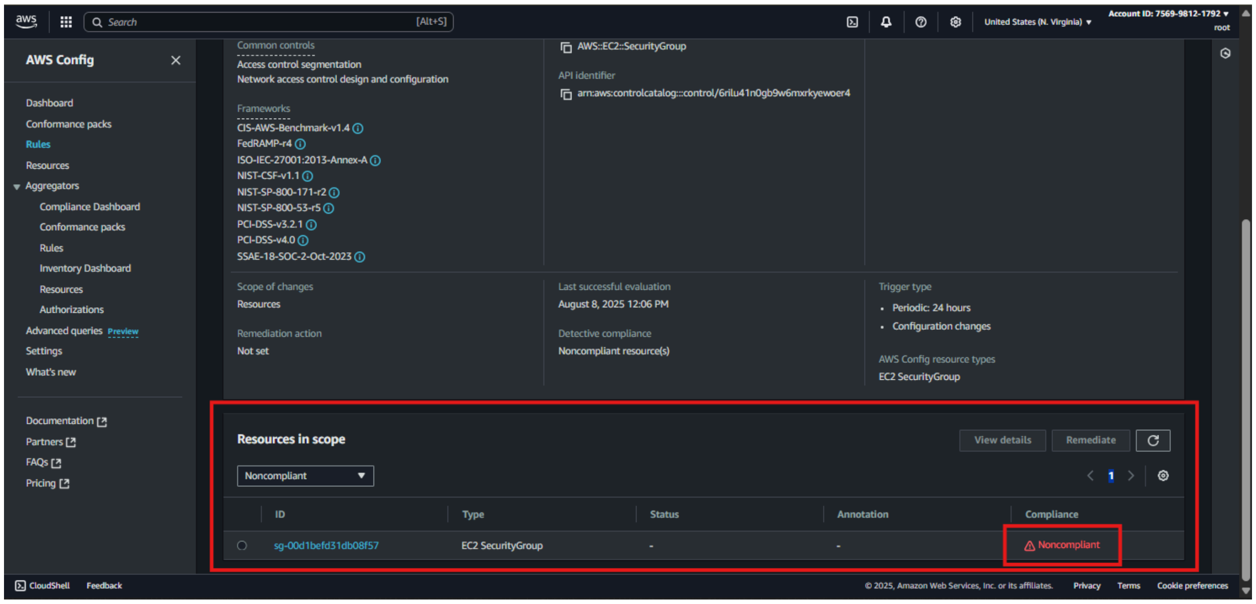
Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: Noncompliant

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
eip-attached	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)



Common controls
Access control segmentation
Network access control design and configuration

Frameworks
CIS-AWS-Benchmark-v1.4
FedRAMP-r4
ISO-IEC-27001:2013-Annex-A
NIST-CSF-v1.1
NIST-SP-800-171-r2
NIST-SP-800-53-r5
PCI-DSS-v3.2.1
PCI-DSS-v4.0
SSAE-18-SOC-2-Oct-2023

AWS::EC2::SecurityGroup

API identifier
arn:aws:controlcatalog:::control/6rilu41n0gb9w6mxrkyewoer4

Scope of changes
Resources

Remediation action
Not set

Last successful evaluation
August 8, 2025 12:06 PM

Detective compliance
Noncompliant resource(s)

Trigger type
- Periodic: 24 hours
- Configuration changes

AWS Config resource types
EC2 SecurityGroup

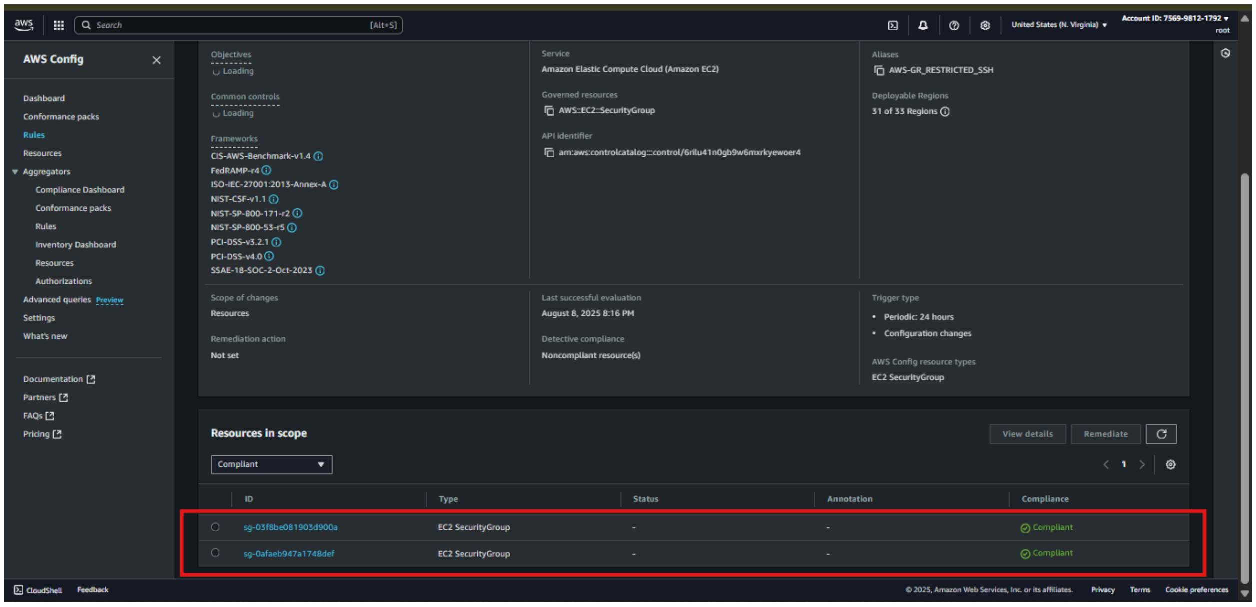
Resources in scope

Noncompliant

ID	Type	Status	Annotation	Compliance
sg-00d1befd31db08f57	EC2 SecurityGroup	-	-	⚠ Noncompliant

- We restricted SSH to our approved office IP ranges.

  The resource returned to `COMPLIANT`, maintaining our least-privilege network access policy.



Account ID: 7569-9812-1792
root
United States (N. Virginia)

Objectives
Loading

Common controls
Loading

Frameworks
CIS-AWS-Benchmark-v1.4
FedRAMP-r4
ISO-IEC-27001:2013-Annex-A
NIST-CSF-v1.1
NIST-SP-800-171-r2
NIST-SP-800-53-r5
PCI-DSS-v3.2.1
PCI-DSS-v4.0
SSAE-18-SOC-2-Oct-2023

Service
Amazon Elastic Compute Cloud (Amazon EC2)

Governed resources
AWS::EC2::SecurityGroup

API identifier
arn:aws:controlcatalog:::control/6rilu41n0gb9w6mxrkyewoer4

Aliases
AWS-GR_RESTRICTED_SSH

Deployable Regions
31 of 33 Regions

Scope of changes
Resources

Remediation action
Not set

Last successful evaluation
August 8, 2025 8:16 PM

Detective compliance
Noncompliant resource(s)

Trigger type
- Periodic: 24 hours
- Configuration changes

AWS Config resource types
EC2 SecurityGroup

Resources in scope

Compliant

ID	Type	Status	Annotation	Compliance
sg-03f8be081903d900a	EC2 SecurityGroup	-	-	✓ Compliant
sg-0afaeb947a1748def	EC2 SecurityGroup	-	-	✓ Compliant

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

6. Root Account MFA

Rule: `root-account-mfa-enabled`

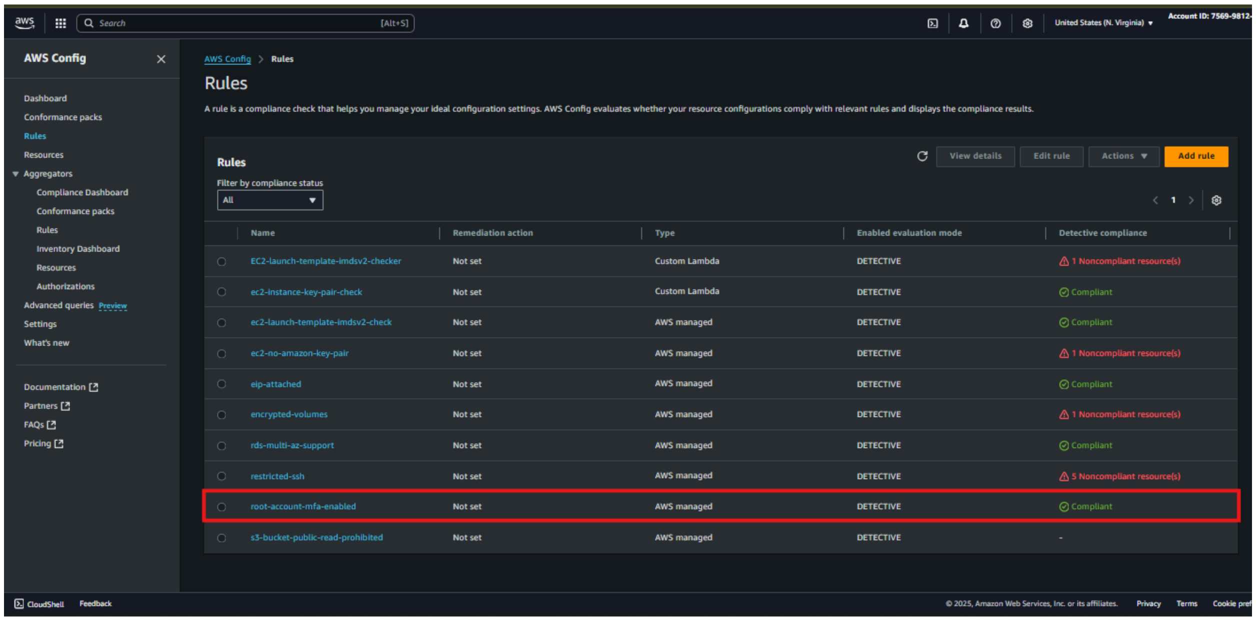
The root account has full privileges; without MFA, it is highly vulnerable.

Leave MFA disabled on the root account.

- We verified the account was operating without MFA.
- AWS Config flagged it as `COMPLIANT`. So we waited and for it to flag `NON-COMPLIANT` But did not get any result

- We set up a virtual MFA device and activated MFA protection.

The account is now `COMPLIANT`, aligning with our account hardening standards.



AWS Config > Rules

Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	Compliant
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
eip-attached	Not set	AWS managed	DETECTIVE	Compliant
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	Compliant
restricted-ssh	Not set	AWS managed	DETECTIVE	5 Noncompliant resource(s)
root-account-mfa-enabled	Not set	AWS managed	DETECTIVE	Compliant
s3-bucket-public-read-prohibited	Not set	AWS managed	DETECTIVE	-

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

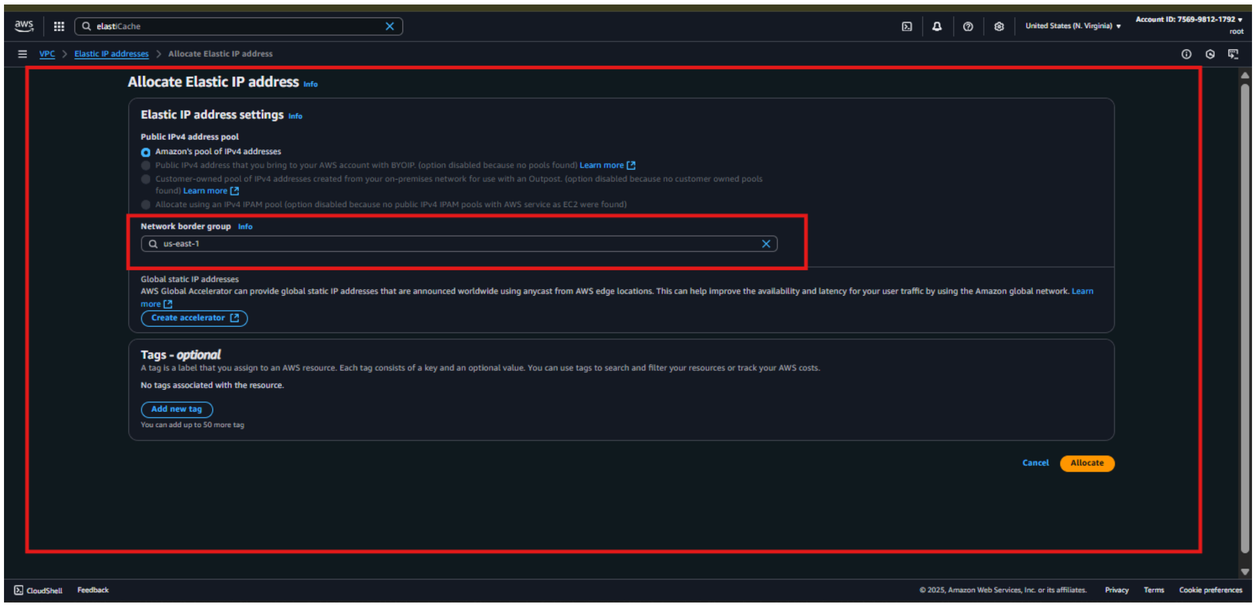
7. Elastic IP Attachment Check

Rule: `eip-attached`

Unattached Elastic IPs incur costs and waste network resources.

Allocated an Elastic IP without attaching it.

- We allocated an EIP and left it unattached.


Allocate Elastic IP address

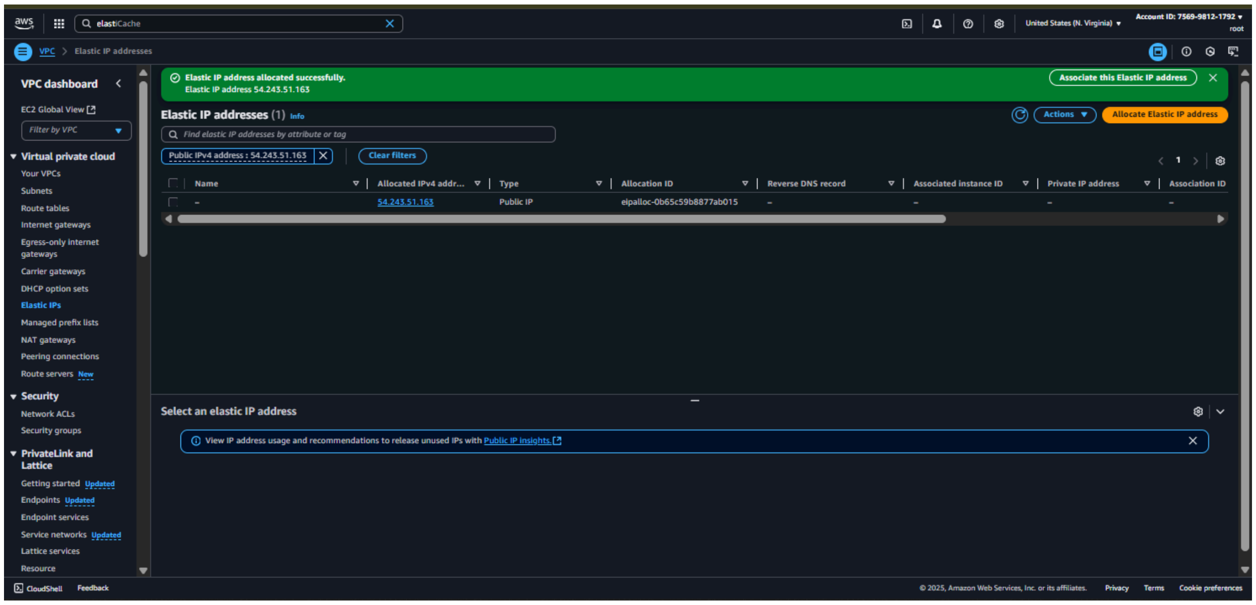
Elastic IP address settings

Public IPv4 address pool
- [✓ Amazon's pool of IPv4 addresses]
- Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found)
- Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found)
- Allocate using an IPv4 IPAM pool (option disabled because no public IPv4 IPAM pools with AWS service as EC2 were found)

Network border group: us-east-1

Global static IP addresses
AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network.

Tags - *optional*
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource.
You can add up to 50 more tag

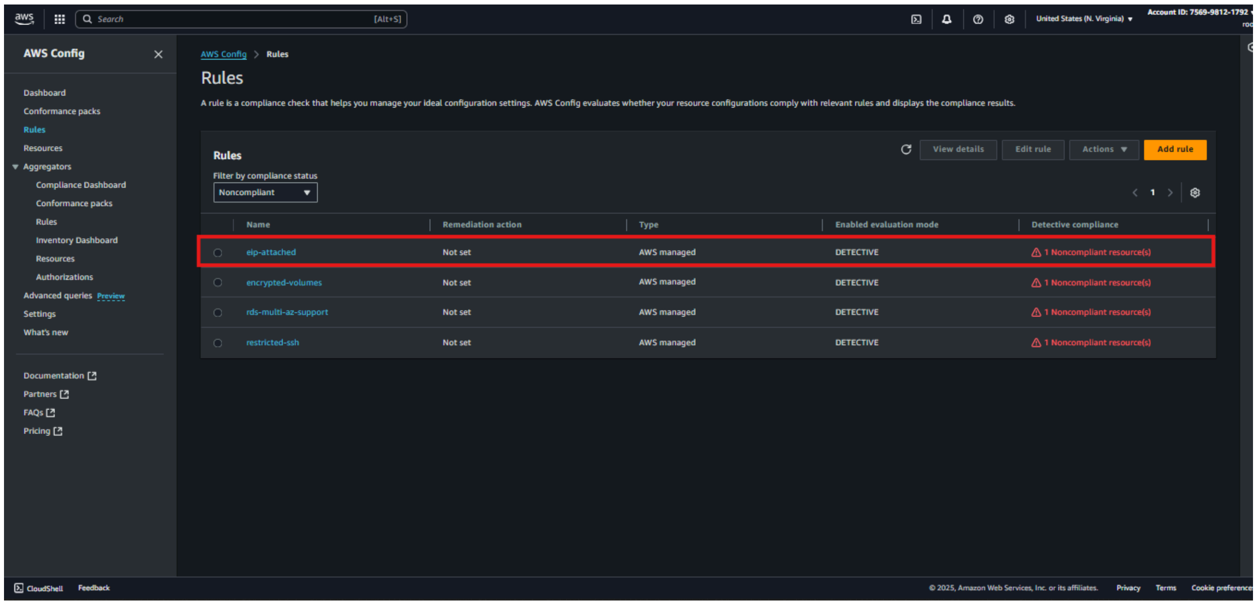

Elastic IP address allocated successfully.
Elastic IP address 54.243.51.163

Elastic IP addresses (1)

Public IPv4 address : 54.243.51.163

Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP address	Association ID
–	54.243.51.163	Public IP	eipalloc-0b65c59b8877ab015	–	–	–	–

- AWS Config flagged it as `NON_COMPLIANT`.



AWS Config > Rules

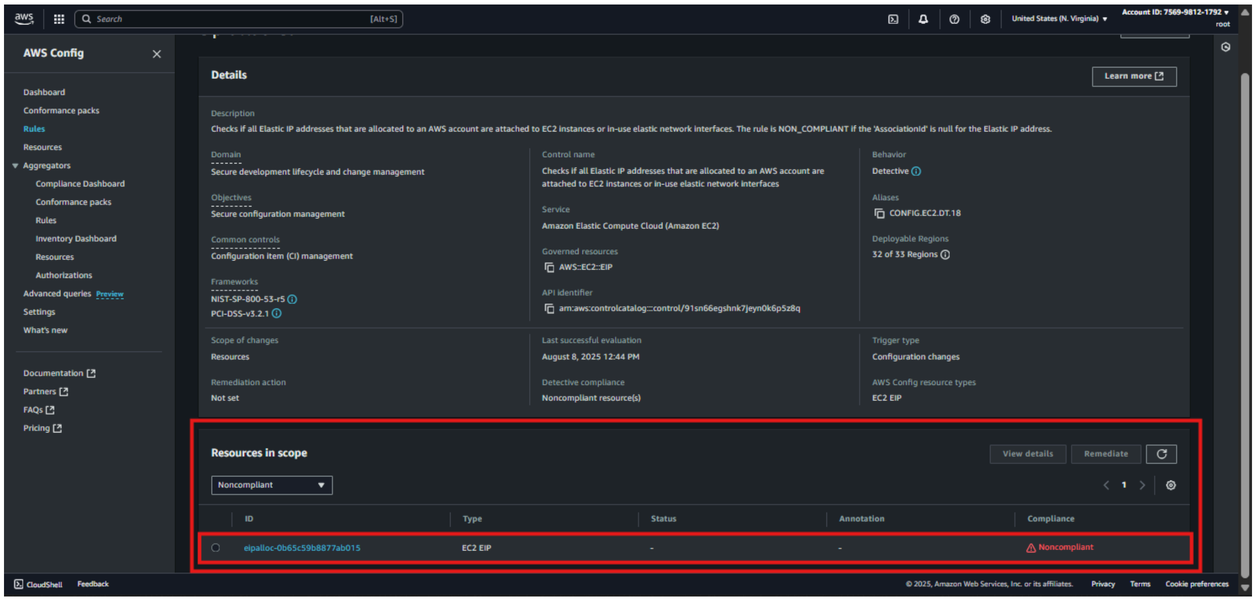
Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: Noncompliant

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
eip-attached	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)



Details

Description
Checks if all Elastic IP addresses that are allocated to an AWS account are attached to EC2 instances or in-use elastic network interfaces. The rule is NON_COMPLIANT if the 'AssociationId' is null for the Elastic IP address.

Domain: Secure development lifecycle and change management
Objectives: Secure configuration management
Common controls: Configuration item (CI) management
Frameworks: NIST-SP-800-53-r5, PCI-DSS-v3.2.1

Control name: Checks if all Elastic IP addresses that are allocated to an AWS account are attached to EC2 instances or in-use elastic network interfaces
Service: Amazon Elastic Compute Cloud (Amazon EC2)
Governed resources: AWS::EC2::EIP
API identifier: arn:aws:controlcatalog:::control/91sn66egshnk7jeyn0k6p5z8q

Behavior: Detective
Aliases: CONFIG.EC2.DT.18
Deployable Regions: 32 of 33 Regions

Scope of changes: Resources
Remediation action: Not set
Last successful evaluation: August 8, 2025 12:44 PM
Detective compliance: Noncompliant resource(s)
Trigger type: Configuration changes
AWS Config resource types: EC2 EIP

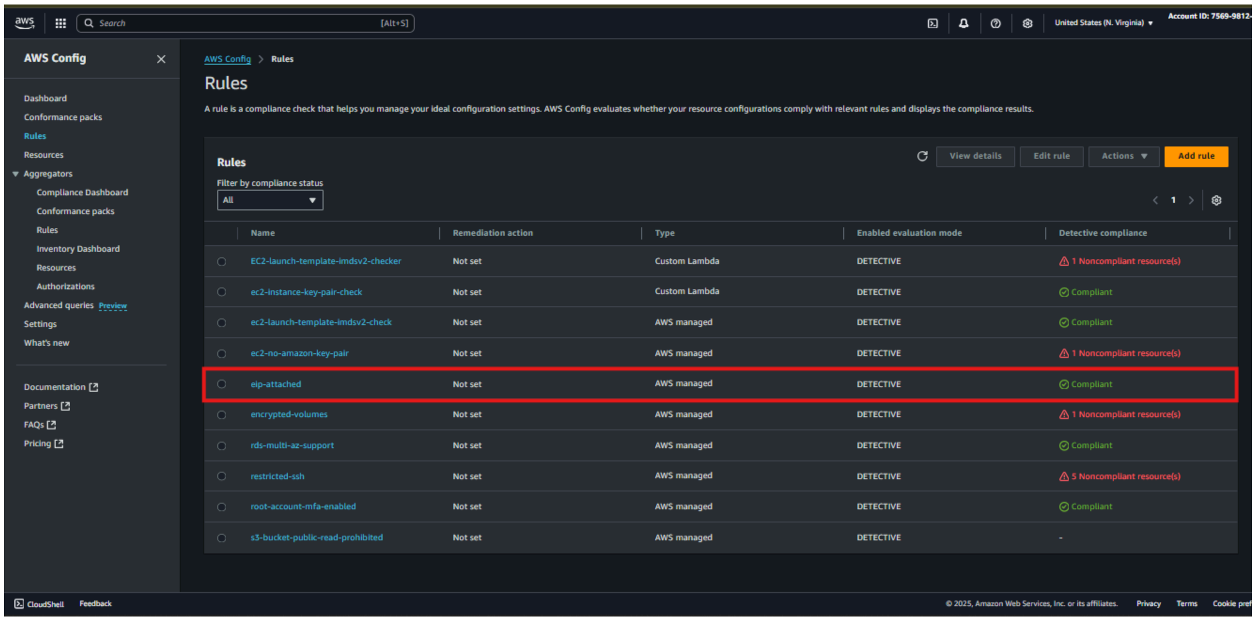
Resources in scope

Noncompliant

ID	Type	Status	Annotation	Compliance
eipalloc-0b65c59b8877ab015	EC2 EIP	-	-	⚠ Noncompliant

- We attached the EIP to a running instance.

The resource became `COMPLIANT`, ensuring optimal IP resource use.



Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	Compliant
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
eip-attached	Not set	AWS managed	DETECTIVE	Compliant
encrypted-volumes	Not set	AWS managed	DETECTIVE	1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	Compliant
restricted-ssh	Not set	AWS managed	DETECTIVE	5 Noncompliant resource(s)
root-account-mfa-enabled	Not set	AWS managed	DETECTIVE	Compliant
s3-bucket-public-read-prohibited	Not set	AWS managed	DETECTIVE	-

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

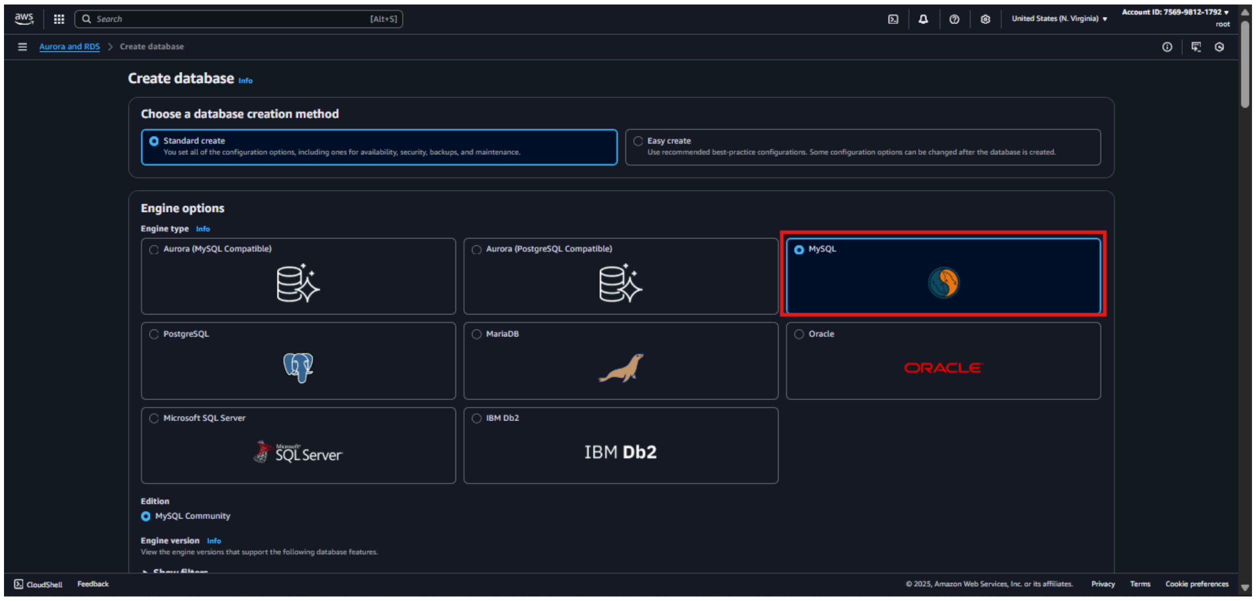
8. RDS Multi-AZ Enforcement

Rule: `rds-multi-az-support`

Single-AZ RDS deployments can cause outages during AZ failures.

So We Launched RDS instance in Single-AZ mode.

- We created the RDS instance without Multi-AZ.


Aurora and RDS > Create database

Create database Info

Choose a database creation method

[✓ Standard create] You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create — Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

Aurora (MySQL Compatible) | Aurora (PostgreSQL Compatible) | [✓ MySQL]

PostgreSQL | MariaDB | Oracle

Microsoft SQL Server | IBM Db2

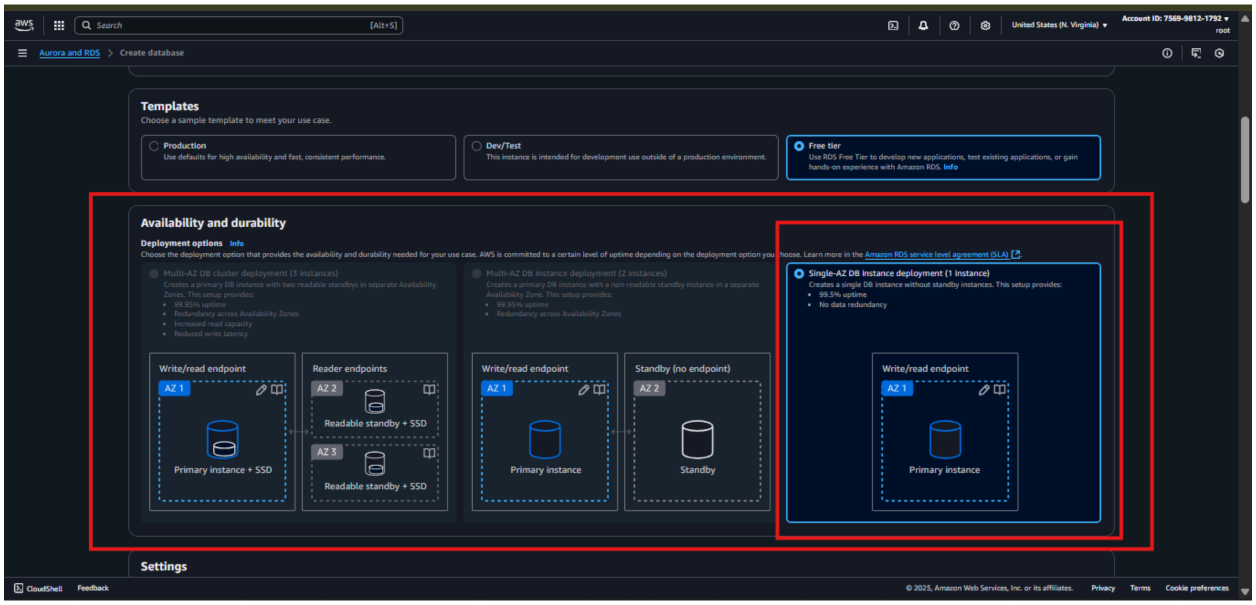
Edition

[✓ MySQL Community]

Engine version Info

View the engine versions that support the following database features.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Aurora and RDS > Create database

Templates

Choose a sample template to meet your use case.

Production — Use defaults for high availability and fast, consistent performance.

Dev/Test — This instance is intended for development use outside of a production environment.

[✓ Free tier] Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

Availability and durability

Deployment options Info

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the Amazon RDS service level agreement (SLA).

Multi-AZ DB cluster deployment (3 instances) — Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

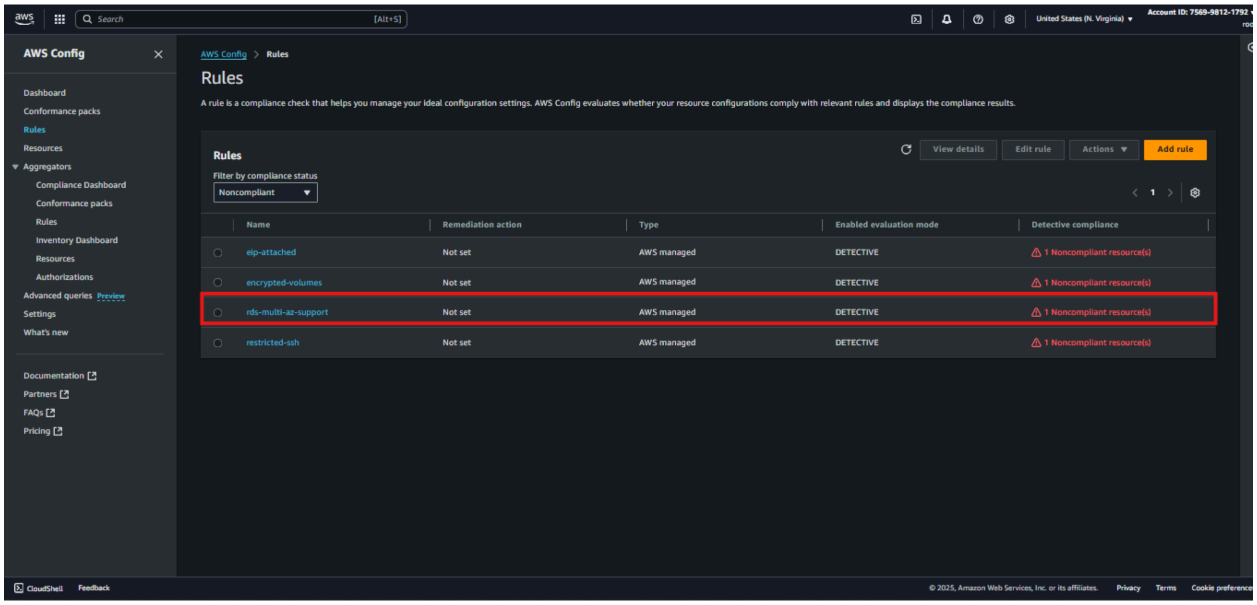
Multi-AZ DB instance deployment (2 instances) — Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones

[✓ Single-AZ DB instance deployment (1 instance)] Creates a single DB instance without standby instances. This setup provides:
- 99.5% uptime
- No data redundancy

Settings

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- AWS Config flagged it as `NON_COMPLIANT`.



AWS Config > Rules

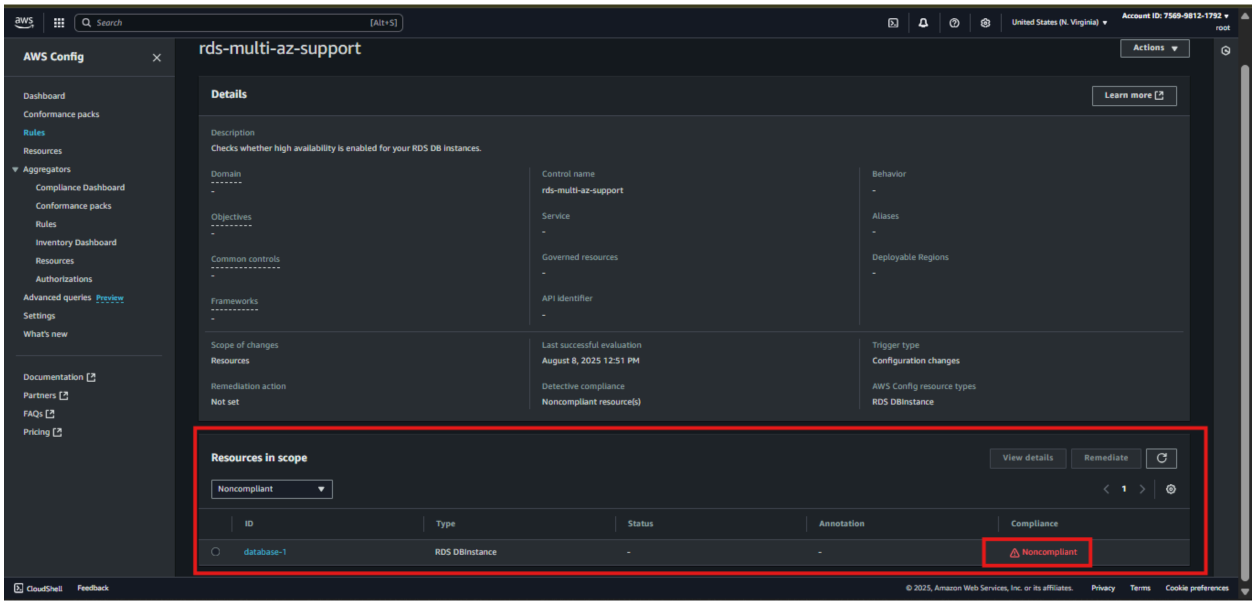
Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: Noncompliant

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
eip-attached	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
restricted-ssh	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)



rds-multi-az-support

Details

Description
Checks whether high availability is enabled for your RDS DB instances.

Domain: -	Control name: rds-multi-az-support	Behavior: -
Objectives: -	Service: -	Aliases: -
Common controls: -	Governed resources: -	Deployable Regions: -
Frameworks: -	API identifier: -	
Scope of changes: Resources	Last successful evaluation: August 8, 2025 12:51 PM	Trigger type: Configuration changes
Remediation action: Not set	Detective compliance: Noncompliant resource(s)	AWS Config resource types: RDS DBInstance

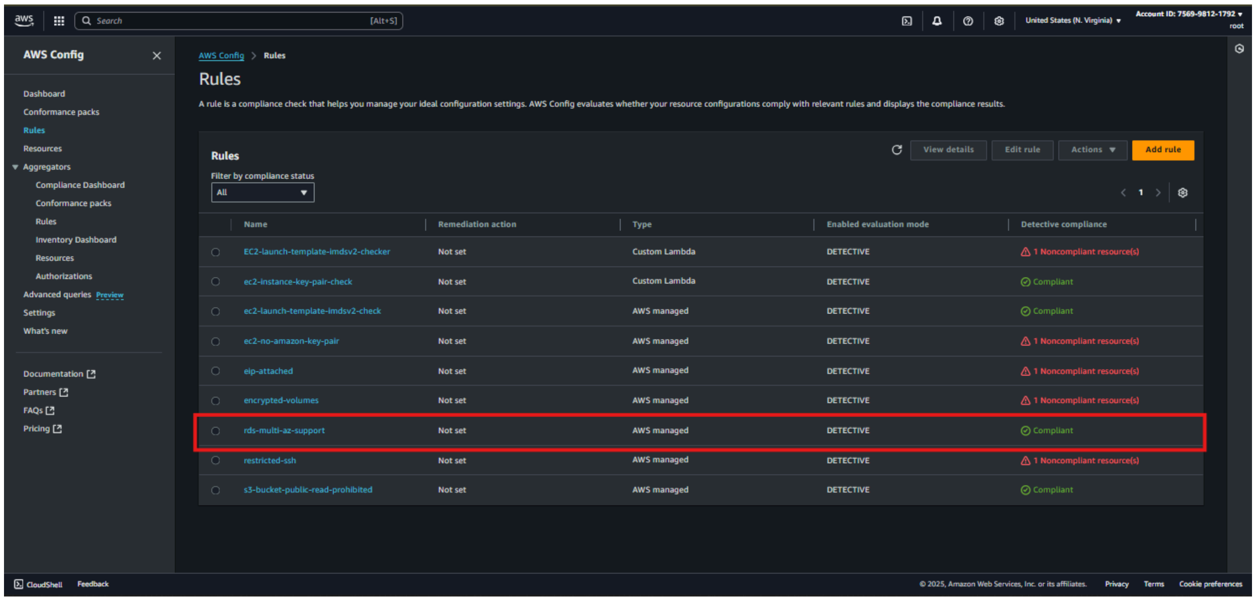
Resources in scope

Noncompliant

ID	Type	Status	Annotation	Compliance
database-1	RDS DBInstance	-	-	⚠ Noncompliant

- We modified it to enable Multi-AZ replication.

AWS Config marked the instance `COMPLIANT`, improving resilience and availability.


Rules

A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.

Rules

Filter by compliance status: All

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
EC2-launch-template-imdsv2-checker	Not set	Custom Lambda	DETECTIVE	⚠ 1 Noncompliant resource(s)
ec2-instance-key-pair-check	Not set	Custom Lambda	DETECTIVE	✓ Compliant
ec2-launch-template-imdsv2-check	Not set	AWS managed	DETECTIVE	✓ Compliant
ec2-no-amazon-key-pair	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
eip-attached	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
encrypted-volumes	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
rds-multi-az-support	Not set	AWS managed	DETECTIVE	✓ Compliant
restricted-ssh	Not set	AWS managed	DETECTIVE	⚠ 1 Noncompliant resource(s)
s3-bucket-public-read-prohibited	Not set	AWS managed	DETECTIVE	✓ Compliant

---

Challenges Encountered

1. **S3 Bucket Policy Conflicts** – Pre-existing bucket policies conflicted with the project's compliance rules, causing repeated non-compliance alerts until corrected.
2. **Rule Evaluation Delay** – AWS Config's evaluation latency caused a lag between remediation and updated compliance status, which slowed verification.
3. **Cross-Region Configuration Gaps** – Some compliance rules did not propagate across all targeted regions, requiring manual synchronization.

Despite these issues, We effectively applied troubleshooting measures, revised IAM policies, updated resource configurations, and achieved **full compliance restoration** across all tested rules.

▼ This project's success was made possible by the dedication, expertise, and collaborative effort of the **Cloud Infrastructure Engineering Team.**

Every member contributed to the planning, execution, validation, and documentation of the AWS Config compliance enforcement scenarios. The following individuals are recognized for their active participation and commitment to operational excellence:

#	Name
1	Nwode Chinazor
2	Ifunanya Benedicta
3	Samuel Bardi
4	Nwakile Ebere Maryrose
5	Emmanuel Chiakasiemobi
6	Nkem Nwawume
7	Eberechi Mercy Uzor
8	Folarin Veronica
9	Clementina Ohaeri
10	Ugonna Ochuba
11	Patience Ibe
12	Gift Ukporo
13	Bona Ogochukwu
14	Uzordi Gideon
15	Owajimimin Gogo John
16	Yusuff Oluwagbamila
17	Obiano Lilian U.
18	Oreva Oputeh
19	Peter Onota
20	Ogbein Gracious
21	Isreal Okeke
22	Tefue Favour Onajite
23	Hephzibah Tobin
24	Olatoyin Pratt

#	Name
25	Gabriel Philip Offiong
26	Yemisi Folaranmi
27	Oguguo Ekene Justice
28	Miracle Chinasa Emuerhi
29	Valentine Chinedu Onuoka
30	Sule Waheed Oladimeji
31	Rahimah Omananyi
32	Duru Laura Kelechi

Through their joint effort, we achieved **100% compliance restoration** across all tested AWS Config rules, proving our team’s capability to detect, respond to, and remediate misconfigurations in real time while strengthening our governance posture.