AWS CodeDeploy

AWS CodeDeploy automates deployments to EC2 / on-prem servers, serverless Lambda functions, and Amazon ECS services. It coordinates installing revisions, running lifecycle hooks, shifting traffic (for Lambda/ECS blue/green), and can perform automatic rollbacks when things go wrong.

CodeDeploy can deploy application content that runs on a server and is stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. CodeDeploy can also deploy a serverless Lambda function. You do not need to make changes to your existing code before you can use CodeDeploy.

Overview of CodeDeploy deployment types

CodeDeploy provides two deployment type options:

- **In-place deployment**: The application on each instance in the deployment group is stopped, the latest application revision is installed, and the new version of the application is started and validated. You can use a load balancer so that each instance is deregistered during its deployment and then restored to service after the deployment is complete. Only deployments that use the EC2/On-Premises compute platform can use in-place deployments.

Note AWS Lambda and Amazon ECS deployments cannot use an in-place deployment type.

- **Blue/green deployment**: The behavior of your deployment depends on which compute platform you use:
    - **Blue/green on an EC2/On-Premises compute platform**: The instances in a deployment group (the original environment) are replaced by a different set of instances (the replacement environment) using these steps:
        - Instances are provisioned for the replacement environment.
        - The latest application revision is installed on the replacement instances.
        - An optional wait time occurs for activities such as application testing and system verification.
        - Instances in the replacement environment are registered with one or more Elastic Load Balancing load balancers, causing traffic to be rerouted to them. Instances in the original environment are deregistered and can be terminated or kept running for other uses.

Note If you use an EC2/On-Premises compute platform, be aware that blue/green deployments work with Amazon EC2 instances only.

-
    - **Blue/green on an AWS Lambda or Amazon ECS compute platform**: Traffic is shifted in increments according to a **canary**, **linear**, or **all-at-once** deployment configuration.

    - **Blue/green deployments through AWS CloudFormation**: Traffic is shifted from your current resources to your updated resources as part of an AWS CloudFormation stack update. Currently, only ECS blue/green deployments are supported.

Key building blocks

- **Application** – logical container for revisions and deployment groups.
- **Revision** – your deployable package + AppSpec file (tells CodeDeploy what to do on target).
- **Deployment group** – the *targets* for a deployment (instance tags, Auto Scaling groups, ECS services, or Lambda configuration), plus the group's deployment style, service role, alarms and rollback settings.
- **Deployment configuration** – rules that dictate speed/health: minimum healthy hosts (EC2) or traffic-shift policy (ECS/Lambda). You can use AWS predefined configs or create custom ones.

Resources What is CodeDeploy? - AWS CodeDeploy

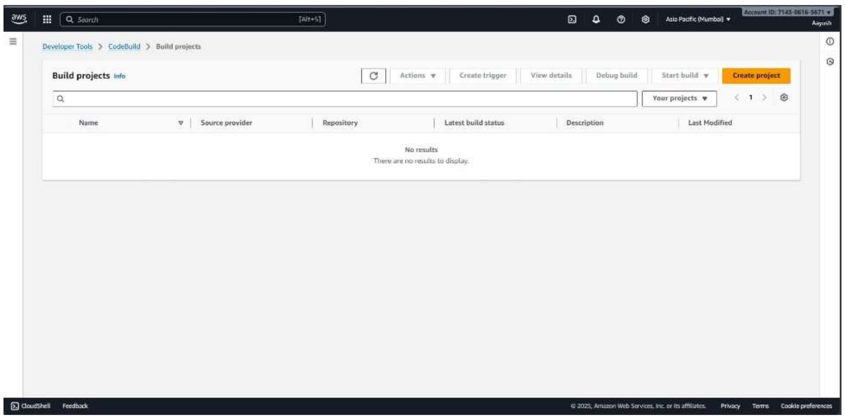
Getting started with CodeDeploy - AWS CodeDeploy

Project

Configure and run CodeBuild for a project, including defining build specifications and integrating with other AWS services.

(Continues Integration)

Step 1: - Goto AWS console Dashboard and Open CodeBuild and click in create project.



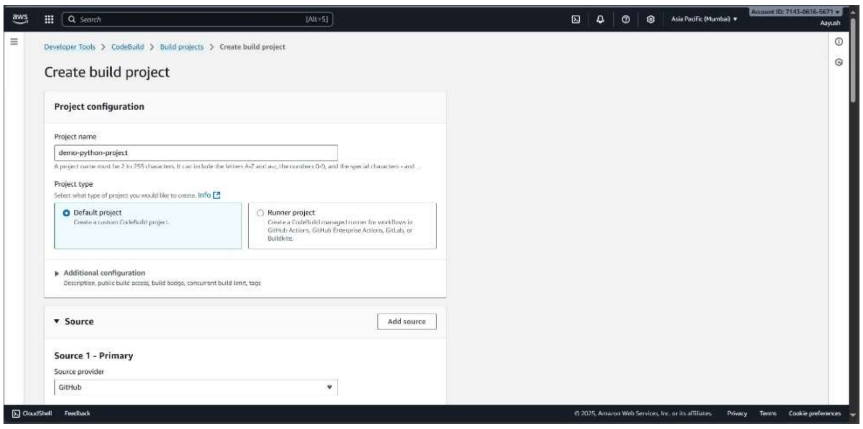
Developer Tools > CodeBuild > Build projects

Build projects Info

Name	Source provider	Repository	Latest build status	Description	Last Modified

No results
There are no results to display.

Step 2: - Enter project name and select options as seen in picture.



Developer Tools > CodeBuild > Build projects > Create build project

Create build project

Project configuration

Project name
demo-python-project
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

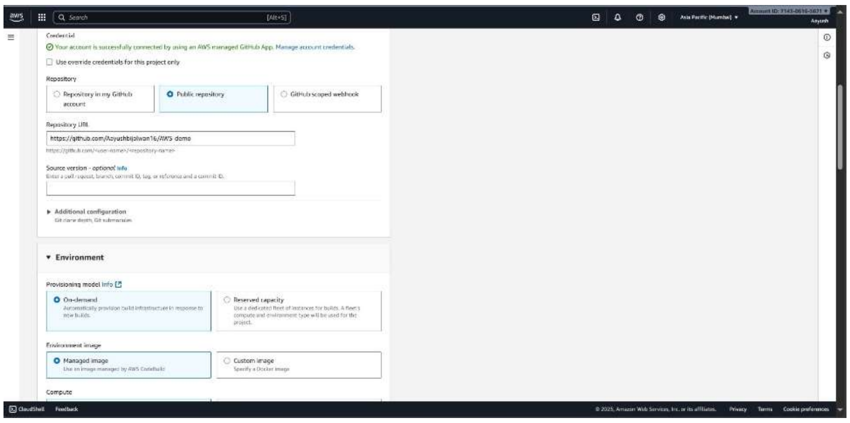
Project type
Select what type of project you would like to create. Info
[✓ Default project] Create a custom CodeBuild project.
Runner project: Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.

Additional configuration
Description, public build access, build badge, concurrent build limit, tags

Source

Source 1 - Primary
Source provider
GitHub

Step 3: - Select public repo and configure it with your GitHub.



Credential
Your account is successfully connected by using an AWS managed GitHub App. Manage account credentials.
Use override credentials for this project only

Repository
Repository in my GitHub account [✓ Public repository] GitHub scoped webhook

Repository URL
https://github.com/Aayushbijalwan16/AWS-demo
https://github.com/<user-name>/<repository-name>

Source version - optional Info
Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

Additional configuration
Git clone depth, Git submodules

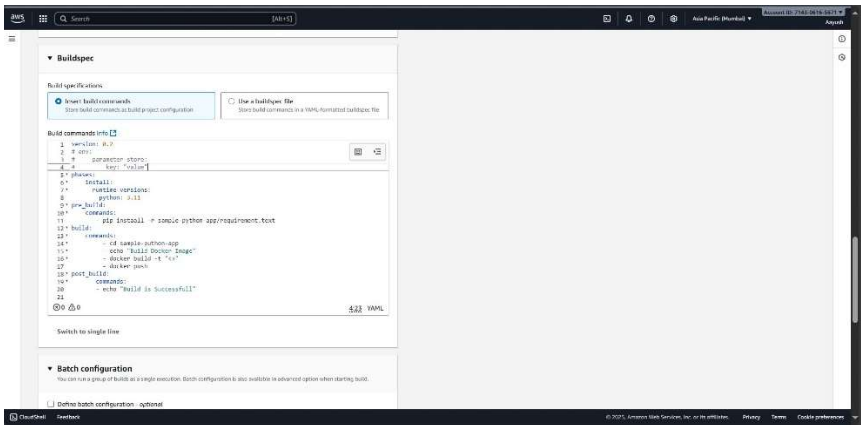
Environment

Provisioning model Info
[✓ On-demand] Automatically provision build infrastructure in response to new builds.
Reserved capacity: Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

Environment image
[✓ Managed image] Use an image managed by AWS CodeBuild
Custom image: Specify a Docker image

Compute

Step 4: - Now create a Buildspec file as shown in picture.



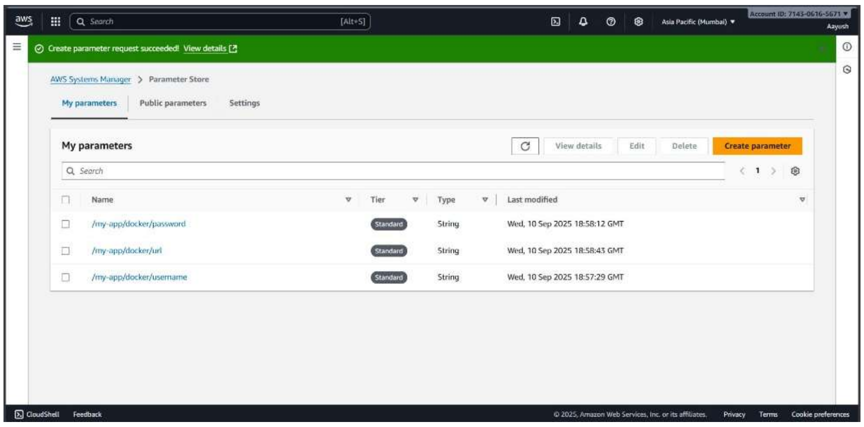
Buildspec

Build specifications: Insert build commands (Store build commands as build project configuration) / Use a buildspec file (Store build commands in a YAML-formatted buildspec file)

Build commands

```
version: 0.2
# env:
#     parameter store:
#         key: "value"
phases:
    install:
      runtime versions:
        python: 3.11
pre_build:
    commands:
         pip instaall -r sample python app/requirement.text
build:
    commands:
         - cd sample-puthon-app
           echo "Build Docker Image"
         - docker build -t "<>"
         - docker push
post_build:
      commands:
      - echo "Build is Successfull"
```

Step 5: - Now Create parameters in AWS System Manager where you store you credentials of Docker.



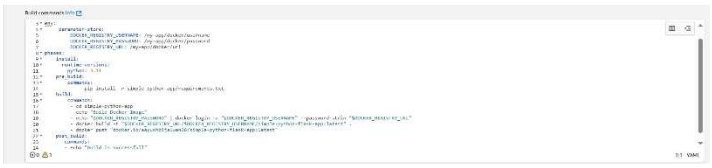
Create parameter request succeeded! View details

AWS Systems Manager > Parameter Store

My parameters

Name	Tier	Type	Last modified
/my-app/docker/password	Standard	String	Wed, 10 Sep 2025 18:58:12 GMT
/my-app/docker/url	Standard	String	Wed, 10 Sep 2025 18:58:43 GMT
/my-app/docker/username	Standard	String	Wed, 10 Sep 2025 18:57:29 GMT

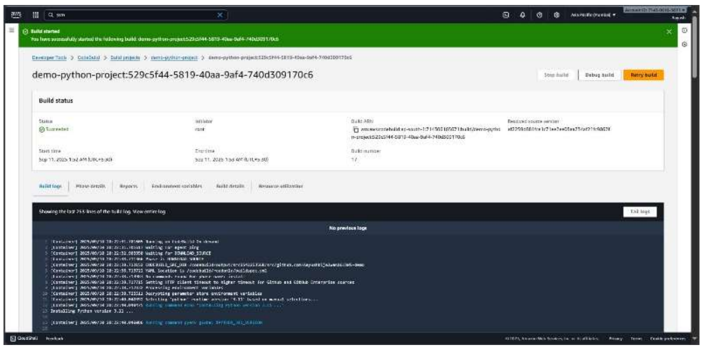
Step 7: - Now edit your BuildSpec file.



Build commands

```
env:
    parameter-store:
        DOCKER_REGISTRY_USERNAME: /my-app/docker/username
        DOCKER_REGISTRY_PASSWORD: /my-app/docker/password
        DOCKER_REGISTRY_URL: /my-app/docker/url
phases:
  install:
    runtime-versions:
      python: 3.11
  pre_build:
      commands:
           pip install -r simple-python-app/requirements.txt
  build:
      commands:
       - cd simple-python-app
         echo "Build Docker Image"
       - echo "$DOCKER_REGISTRY_PASSWORD" | docker login -u "$DOCKER_REGISTRY_USERNAME" --password-stdin "$DOCKER_REGISTRY_URL"
       - docker build -t "$DOCKER_REGISTRY_URL/$DOCKER_REGISTRY_USERNAME/simple-python-flask-app:latest" .
       - docker push "docker.io/[illegible]/simple-python-flask-app:latest"
  post_build:
     commands:
     - echo "Build is successfull"
```

Step 8: - Now click on start build, you might face some issue try to resolve them.



Build started
You have successfully started the following build: demo-python-project:529c5f44-5819-40aa-9af4-740d309170c6

Developer Tools > CodeBuild > Build projects > demo-python-project > demo-python-project:529c5f44-5819-40aa-9af4-740d309170c6

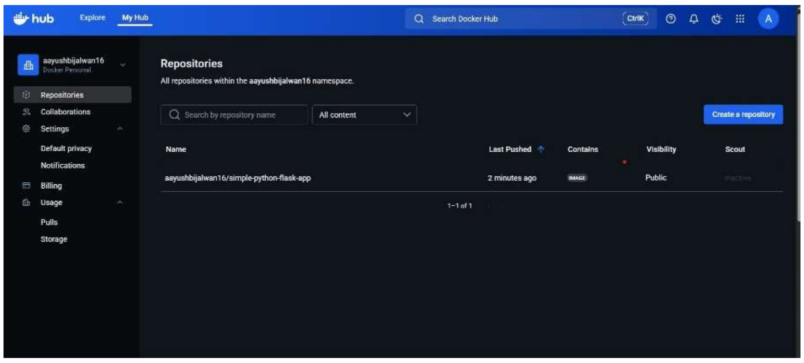
demo-python-project:529c5f44-5819-40aa-9af4-740d309170c6

Build status

Status: Succeeded
Start time: Sep 11, 2025 1:52 AM (UTC+5:30)
End time: Sep 11, 2025 1:53 AM (UTC+5:30)
Build number: 17

```
[Container] 2025/09/10 20:22:31 Waiting for DOWNLOAD_SOURCE
[Container] 2025/09/10 20:22:38 YAML location is /codebuild/readonly/buildspec.yml
[Container] 2025/09/10 20:22:38 Setting HTTP client timeout to higher timeout for Github and GitHub Enterprise sources
[Container] 2025/09/10 20:22:38 Decrypting parameter store environment variables
Installing Python version 3.11 ...
```

Step 9: - Now project is build successfully you can check on Docker hub.



Repositories

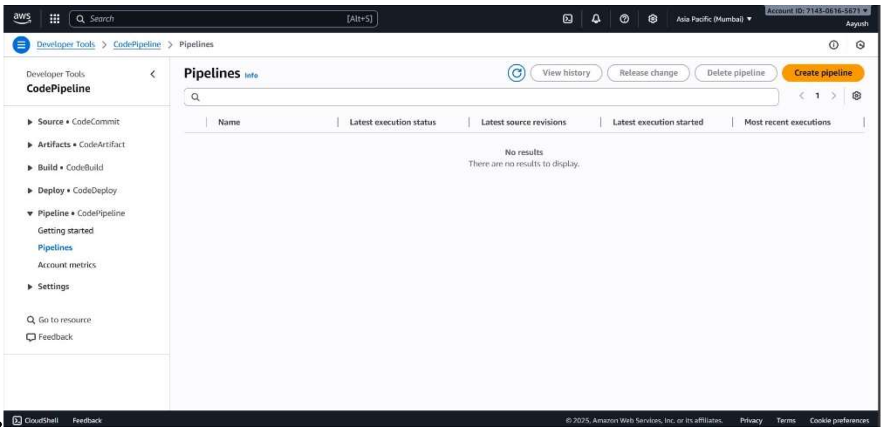
All repositories within the **aayushbijalwan16** namespace.

Name	Last Pushed	Contains	Visibility	Scout
aayushbijalwan16/simple-python-flask-app	2 minutes ago	IMAGE	Public	Inactive

1–1 of 1

(Pipelining Start)

Step 10: - Now add pipelining, open CodePipeline and click on create pipeline

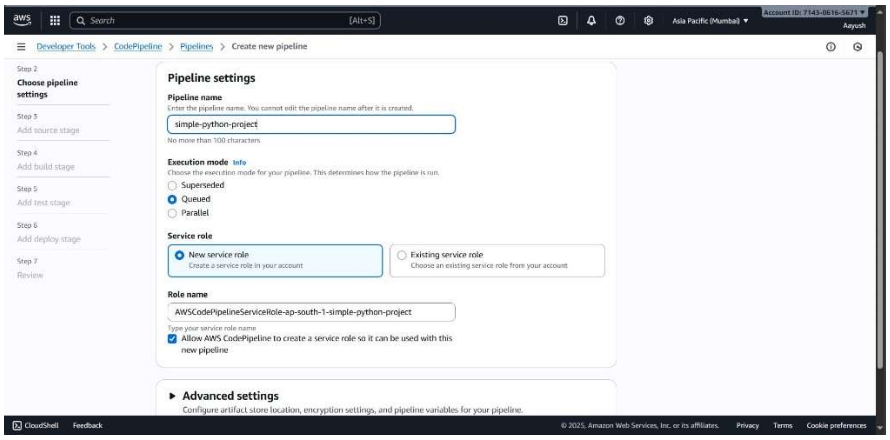


Pipelines Info

Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions

No results
There are no results to display.

Step 11: - Enter Pipeline name and configure accordingly.



Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
simple-python-project
No more than 100 characters

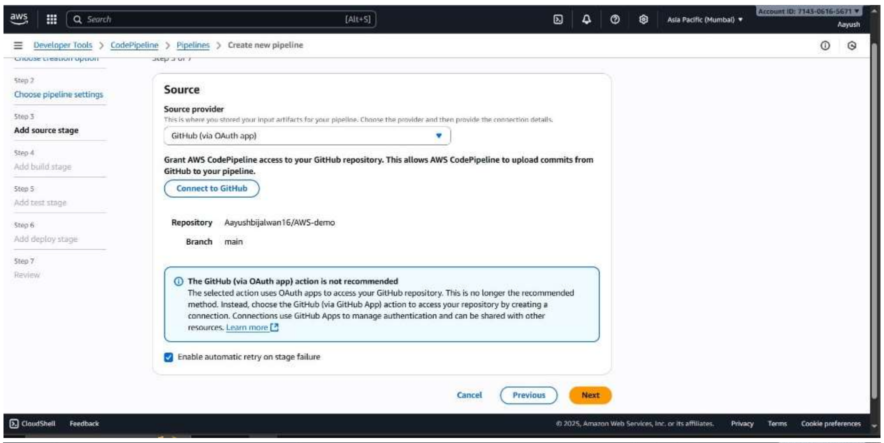
Execution mode Info
Choose the execution mode for your pipeline. This determines how the pipeline is run.
- Superseded
- ● Queued
- Parallel

Service role
- ● New service role — Create a service role in your account
- Existing service role — Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-ap-south-1-simple-python-project
Type your service role name
☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Advanced settings
Configure artifact store location, encryption settings, and pipeline variables for your pipeline.

Step 12: - Again, configure with GitHub and add build stage and click next and final submit.


Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (via OAuth app)

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connect to GitHub

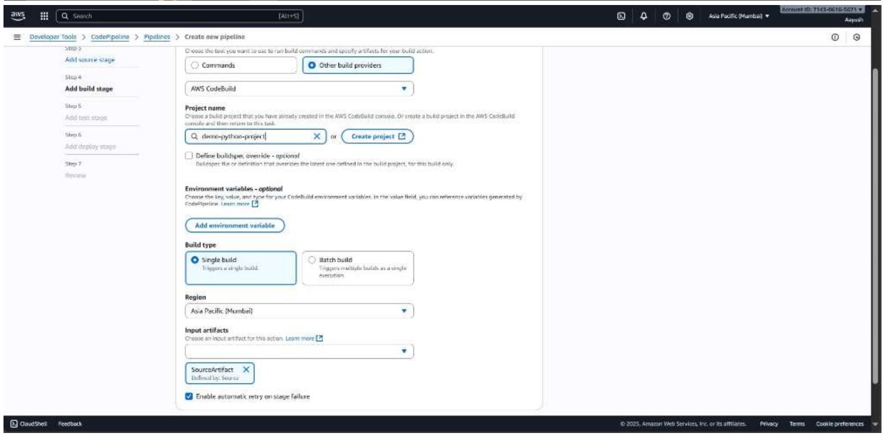
Repository Aayushbijalwan16/AWS-demo

Branch main

The GitHub (via OAuth app) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (via GitHub App) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more

☑ Enable automatic retry on stage failure


Choose the tool you want to use to run build commands and specify artifacts for your build action.

Commands [✓ Other build providers]

AWS CodeBuild

Project name

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

demo-python-project

☐ Define buildspec override - optional

Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

Environment variables - optional

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

Build type

[✓ Single build] Triggers a single build.

Batch build Triggers multiple builds as a single execution.

Region

Asia Pacific (Mumbai)

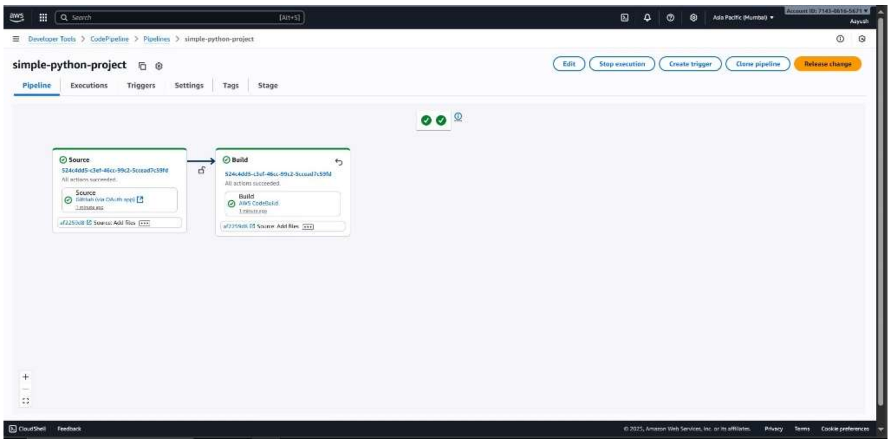
Input artifacts

Choose an input artifact for this action. Learn more

SourceArtifact Defined by: Source

☑ Enable automatic retry on stage failure

Step 13: - Now pipeline is created and you can test by editing you code file in GitHub and see updates.


simple-python-project

Source 524c4dd5-c3ef-46cc-99c2-5cccad7c59fd All actions succeeded.

Source GitHub (via OAuth app) 1 minute ago

af2259d8 Source: Add files

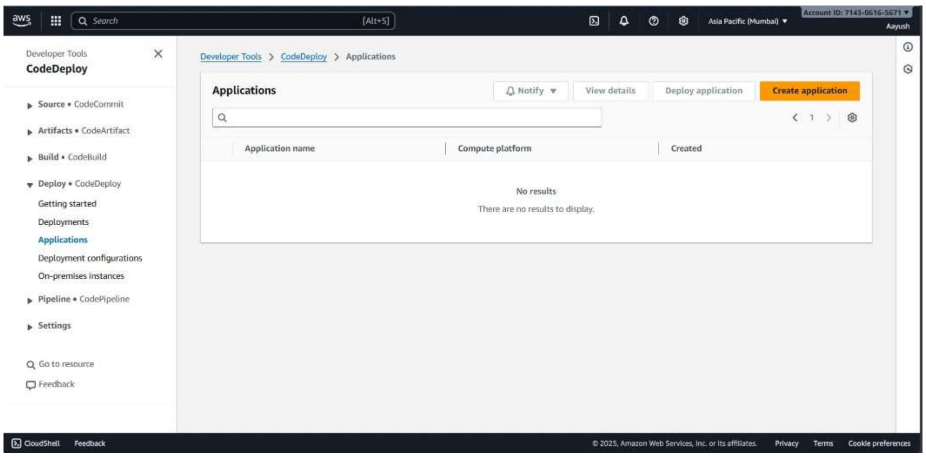
Build 524c4dd5-c3ef-46cc-99c2-5cccad7c59fd All actions succeeded.

Build AWS CodeBuild 1 minute ago

af2259d8 Source: Add files

(Continuous Deployment)

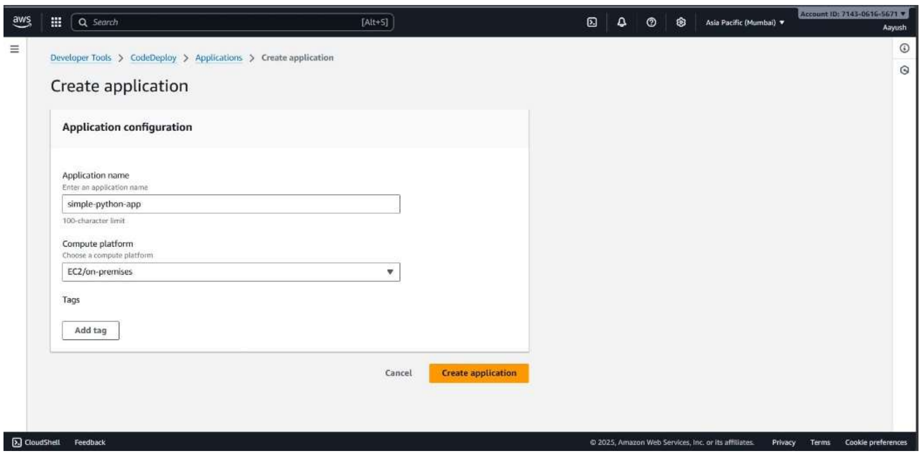
Step 14: - Goto AWS CodeDeploy and click on Application and Click on Create Application.



No results

There are no results to display.

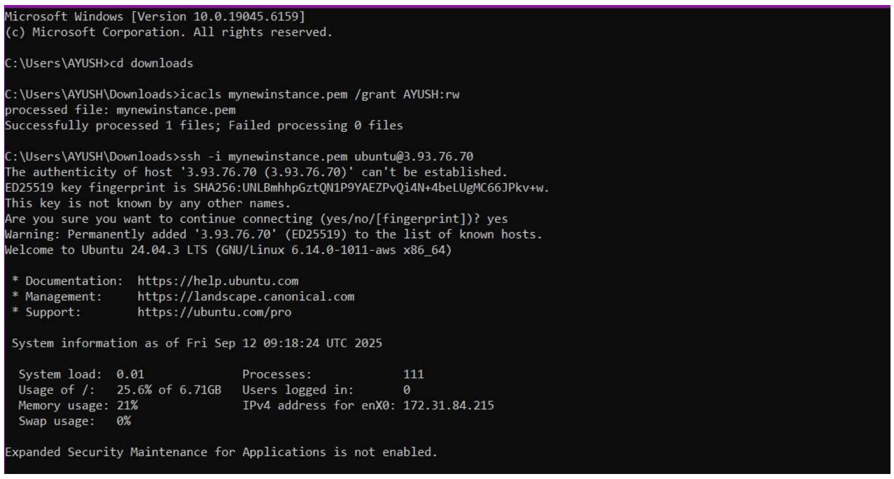
Step 15: - Now enter application name and select Compute Platform and click on create appliction.


Search
[Alt+S]
Asia Pacific (Mumbai) ▾
Account ID: 7143-0616-5671 ▾
Aayush
Developer Tools > CodeDeploy > Applications > Create application
Create application
Application configuration
Tags
Add tag
Cancel
Create application
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Application name
Enter an application name
simple-python-app
100-character limit

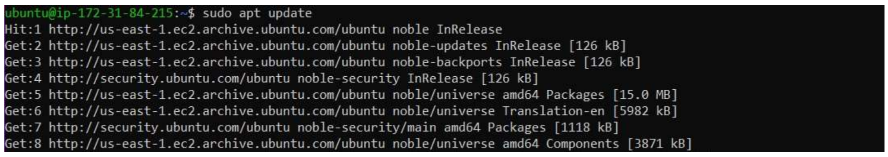
Compute platform
Choose a compute platform
EC2/on-premises

Step 17: -Now create a EC2 instance, IAM role in which primarily attach policy Full access(for demo purpose only) of CodeDepoly and put it into EC2 instance through security section and connect it through terminal.

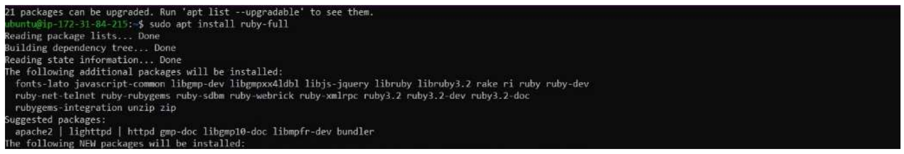


```
Microsoft Windows [Version 10.0.19045.6159]
(c) Microsoft Corporation. All rights reserved.

C:\Users\AYUSH>cd downloads

C:\Users\AYUSH\Downloads>icacls mynewinstance.pem /grant AYUSH:rw
processed file: mynewinstance.pem
Successfully processed 1 files; Failed processing 0 files

C:\Users\AYUSH\Downloads>ssh -i mynewinstance.pem ubuntu@3.93.76.70
The authenticity of host '3.93.76.70 (3.93.76.70)' can't be established.
ED25519 key fingerprint is SHA256:UNLBmhhpGztQN1P9YAEZPvQi4N+4beLUgMC66JPkv+w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.93.76.70' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri Sep 12 09:18:24 UTC 2025

  System load:  0.01              Processes:             111
  Usage of /:   25.6% of 6.71GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for enX0: 172.31.84.215
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.
```

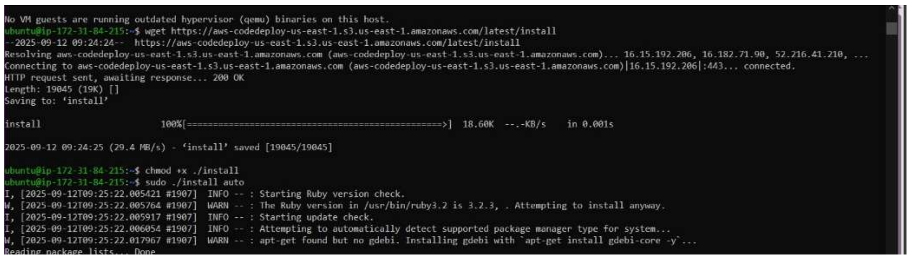
Step 17: - Now install CodeDeploy Agent into EC2 instance and follow this document [Install the CodeDeploy agent for Ubuntu Server - AWS CodeDeploy]



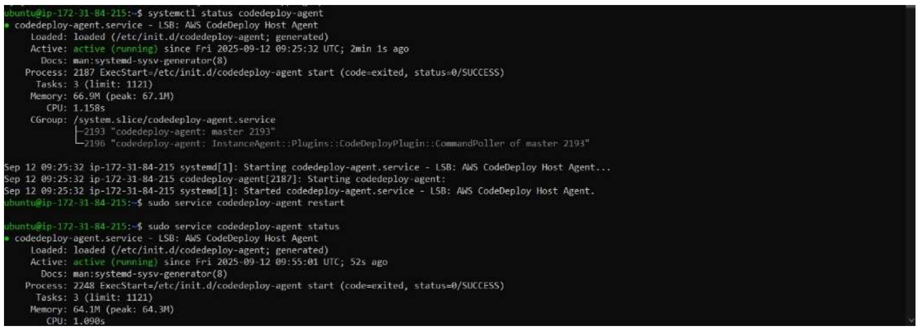
```
ubuntu@ip-172-31-84-215:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1118 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
```



```
21 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-84-215:~$ sudo apt install ruby-full
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fonts-lato javascript-common libgmp-dev libgmpxx4ldbl libjs-jquery libruby libruby3.2 rake ri ruby ruby-dev
  ruby-net-telnet ruby-rubygems ruby-sdbm ruby-webrick ruby-xmlrpc ruby3.2 ruby3.2-dev ruby3.2-doc
  rubygems-integration unzip zip
Suggested packages:
  apache2 | lighttpd | httpd gmp-doc libgmp10-doc libmpfr-dev bundler
The following NEW packages will be installed:
```

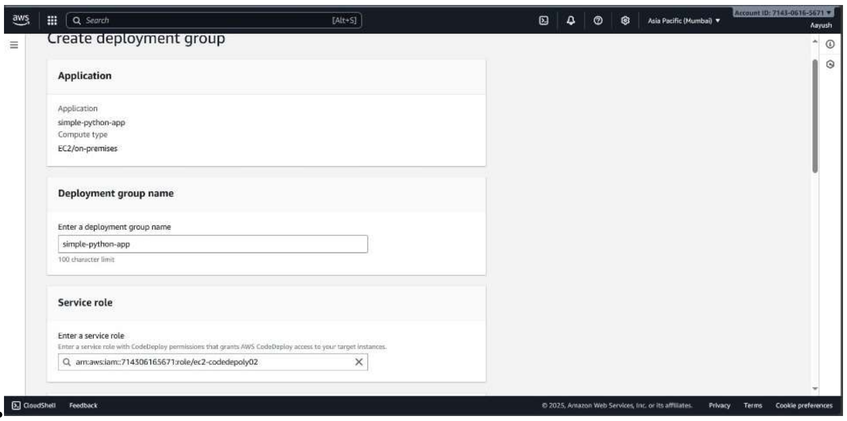


```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-84-215:~$ wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
--2025-09-12 09:24:24--  https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com)... 16.15.192.206, 16.182.71.90, 52.216.41.210, ...
Connecting to aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com)|16.15.192.206|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19045 (19K) []
Saving to: 'install'

install                 100%[=================================================>]  18.60K  --.-KB/s    in 0.001s

2025-09-12 09:24:25 (29.4 MB/s) - 'install' saved [19045/19045]

ubuntu@ip-172-31-84-215:~$ chmod +x ./install
ubuntu@ip-172-31-84-215:~$ sudo ./install auto
I, [2025-09-12T09:25:22.005421 #1907]  INFO -- : Starting Ruby version check.
W, [2025-09-12T09:25:22.005764 #1907]  WARN -- : The Ruby version in /usr/bin/ruby3.2 is 3.2.3, . Attempting to install anyway.
I, [2025-09-12T09:25:22.005917 #1907]  INFO -- : Starting update check.
I, [2025-09-12T09:25:22.006054 #1907]  INFO -- : Attempting to automatically detect supported package manager type for system...
W, [2025-09-12T09:25:22.017967 #1907]  WARN -- : apt-get found but no gdebi. Installing gdebi with `apt-get install gdebi-core -y`...
Reading package lists... Done
```



```
ubuntu@ip-172-31-84-215:~$ systemctl status codedeploy-agent
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
     Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
     Active: active (running) since Fri 2025-09-12 09:25:32 UTC; 2min 1s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 2187 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
      Tasks: 3 (limit: 1121)
     Memory: 66.9M (peak: 67.1M)
        CPU: 1.158s
     CGroup: /system.slice/codedeploy-agent.service
             ├─2193 "codedeploy-agent: master 2193"
             └─2196 "codedeploy-agent: InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller of master 2193"

Sep 12 09:25:32 ip-172-31-84-215 systemd[1]: Starting codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent...
Sep 12 09:25:32 ip-172-31-84-215 codedeploy-agent[2187]: Starting codedeploy-agent:
Sep 12 09:25:32 ip-172-31-84-215 systemd[1]: Started codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent.
ubuntu@ip-172-31-84-215:~$ sudo service codedeploy-agent restart

ubuntu@ip-172-31-84-215:~$ sudo service codedeploy-agent status
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
     Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
     Active: active (running) since Fri 2025-09-12 09:55:01 UTC; 52s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 2248 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
      Tasks: 3 (limit: 1121)
     Memory: 64.1M (peak: 64.3M)
        CPU: 1.090s
```

Step 18: - Now Create Deployment group inside the application by entering deployment group name, service role and other field as shown in picture below


Create deployment group

Application

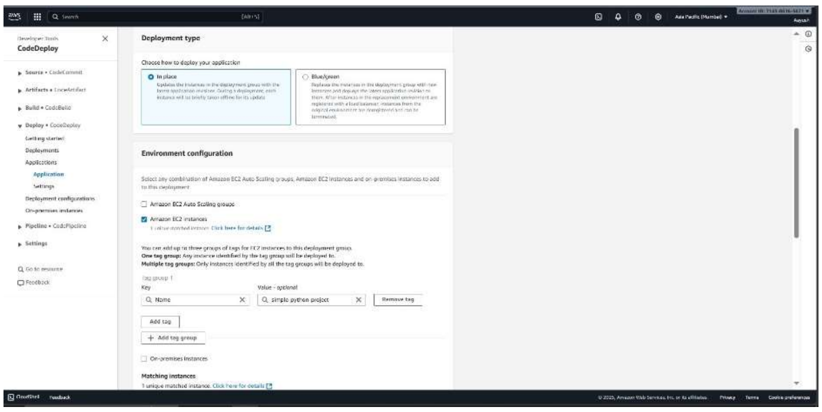
Application
simple-python-app
Compute type
EC2/on-premises

Deployment group name

Enter a deployment group name
simple-python-app
100 character limit

Service role

Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.
arn:aws:iam::714306165671:role/ec2-codedepoly02


Deployment type

Choose how to deploy your application
[✓ In place] Updates the instances in the deployment group with the latest application revision. During a deployment, each instance will be briefly taken offline for its update
Blue/green

Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances and on-premises instances to add to this deployment

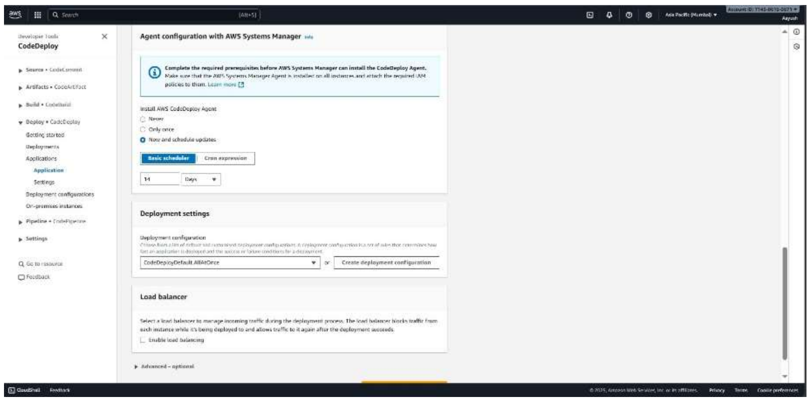
☐ Amazon EC2 Auto Scaling groups
☑ Amazon EC2 instances
1 unique matched instance. Click here for details

You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

Tag group 1
Key: Name
Value - optional: simple python project

☐ On-premises instances

Matching instances
1 unique matched instance. Click here for details


Agent configuration with AWS Systems Manager

Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent. Make sure that the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. Learn more

Install AWS CodeDeploy Agent
Never
Only once
[✓ Now and schedule updates]

Basic scheduler
14 Days

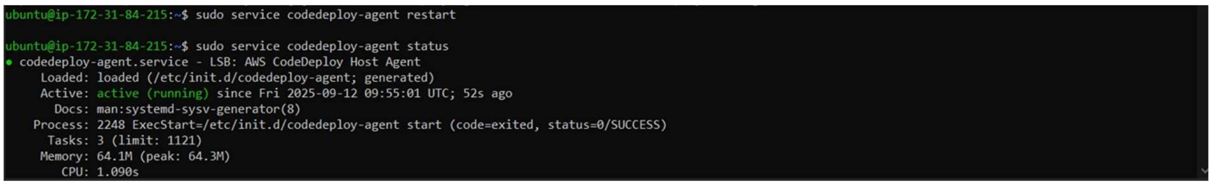
Deployment settings

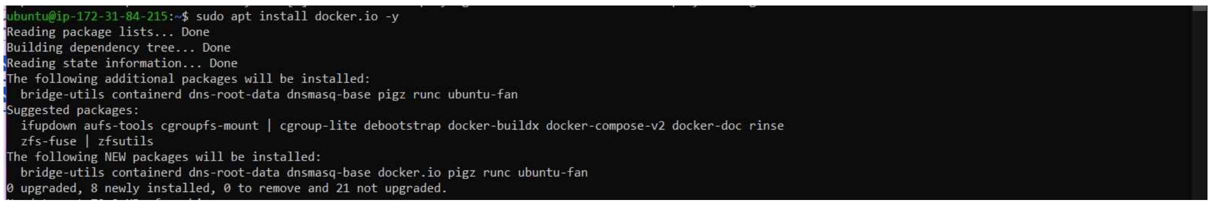
Deployment configuration
CodeDeployDefault.AllAtOnce

Load balancer

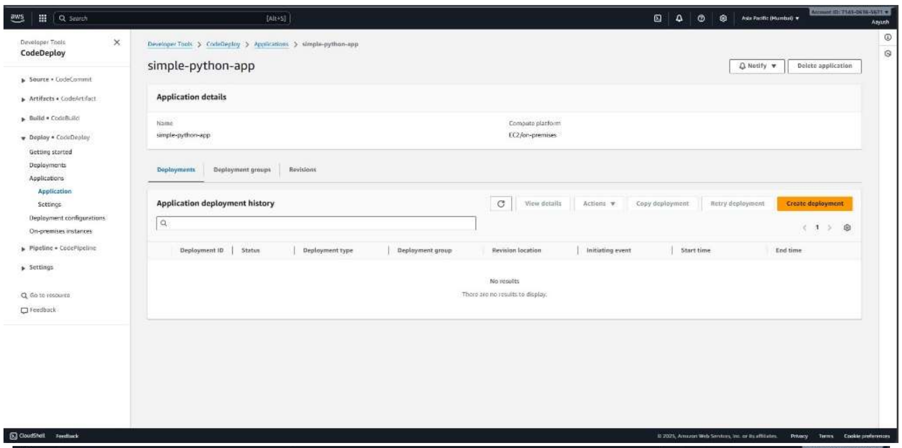
Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.
☐ Enable load balancing

Step 19: - Now add policy EC2FullAccess in same IAM role which is attach to EC2 Instance and run command on terminal and also install Docker in EC2 instance as shown in image below.


```
ubuntu@ip-172-31-84-215:~$ sudo service codedeploy-agent restart

ubuntu@ip-172-31-84-215:~$ sudo service codedeploy-agent status
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
     Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
     Active: active (running) since Fri 2025-09-12 09:55:01 UTC; 52s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 2248 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
      Tasks: 3 (limit: 1121)
     Memory: 64.1M (peak: 64.3M)
        CPU: 1.090s
```


```
ubuntu@ip-172-31-84-215:~$ sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse
  zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 21 not upgraded.
```

Step 20: - Now create deployment inside deployment group.



Developer Tools > CodeDeploy > Applications > simple-python-app

simple-python-app

Application details

Name: simple-python-app

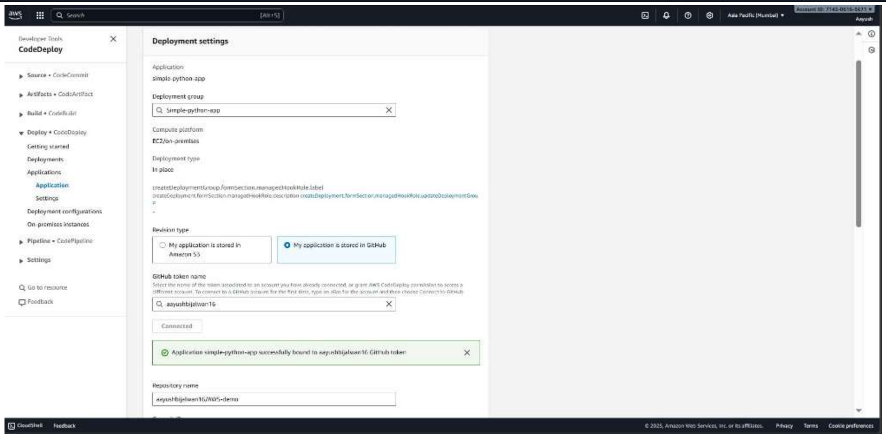
Compute platform: EC2/on-premises

Application deployment history

Deployment ID	Status	Deployment type	Deployment group	Revision location	Initiating event	Start time	End time

No results

There are no results to display.



Deployment settings

Application: simple-python-app

Deployment group: Simple-python-app

Compute platform: EC2/on-premises

Deployment type: In place

createDeploymentGroup.formSection.managedHookRole.label

createDeployment.formSection.managedHookRole.description createDeployment.formSection.managedHookRole.updateDeploymentGroup

Revision type: My application is stored in Amazon S3 [✓ My application is stored in GitHub]

GitHub token name

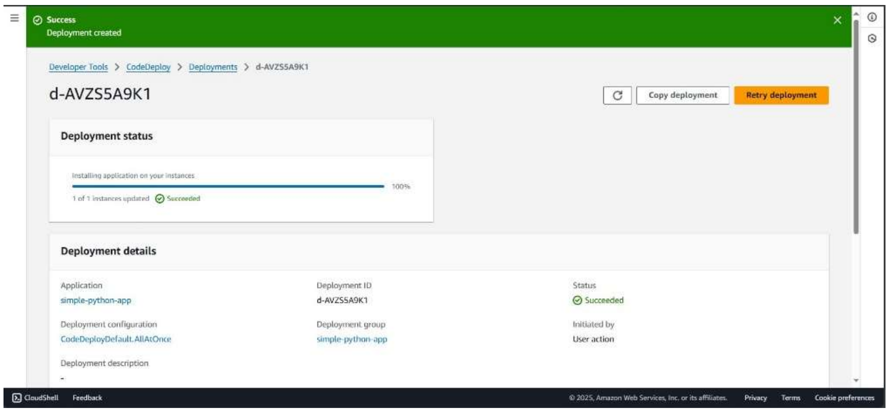
Select the name of the token associated to an account you have already connected, or grant AWS CodeDeploy permission to access a different account. To connect to a GitHub account for the first time, type an alias for the account and then choose Connect to GitHub.

aayushbijalwan16

Connected

Application simple-python-app successfully bound to aayushbijalwan16 GitHub token

Repository name: aayushbijalwan16/AWS-demo



Success
Deployment created

Developer Tools > CodeDeploy > Deployments > d-AVZS5A9K1

d-AVZS5A9K1

Deployment status

Installing application on your instances 100%

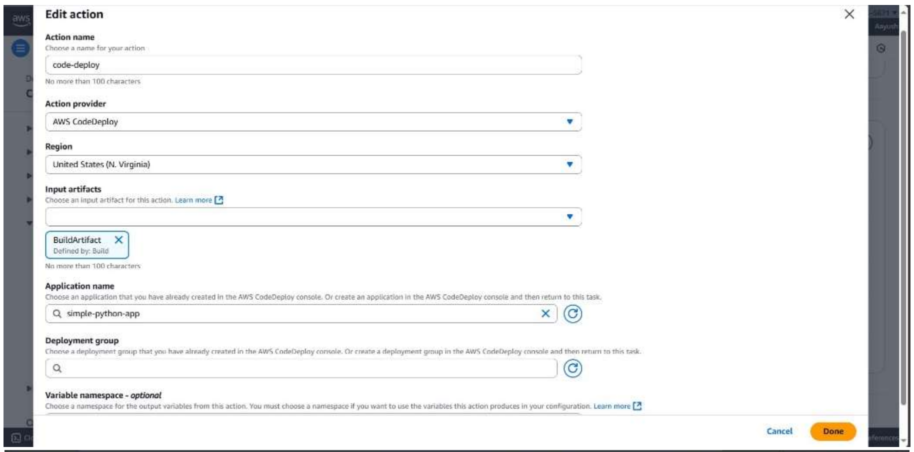
1 of 1 instances updated Succeeded

Deployment details

Application	Deployment ID	Status
simple-python-app	d-AVZS5A9K1	Succeeded
Deployment configuration	**Deployment group**	**Initiated by**
CodeDeployDefault.AllAtOnce	simple-python-app	User action

Deployment description: -

Step 21: - After successful Deployment Go to CodePipeline section and Click on Pipelines and click on pipeline which is created for application and click on edit and click on add stage after the build section and fill the details as shown in image and also select deployment group.



Edit action

Action name
Choose a name for your action
code-deploy
No more than 100 characters

Action provider
AWS CodeDeploy

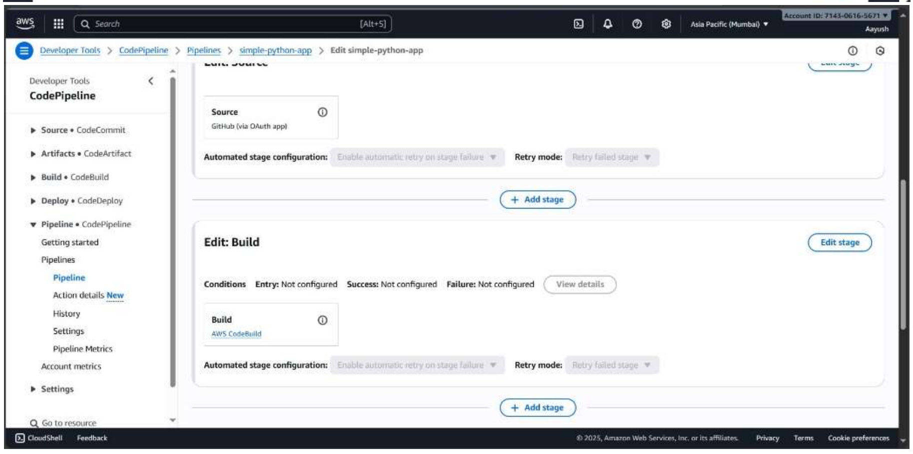
Region
United States (N. Virginia)

Input artifacts
Choose an input artifact for this action. Learn more
BuildArtifact
Defined by: Build
No more than 100 characters

Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.
simple-python-app

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

Variable namespace - *optional*
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. Learn more



Developer Tools > CodePipeline > Pipelines > simple-python-app > Edit simple-python-app

Source
GitHub (via OAuth app)

Automated stage configuration: Enable automatic retry on stage failure Retry mode: Retry failed stage

Edit: Build

Conditions Entry: Not configured Success: Not configured Failure: Not configured

Build
AWS CodeBuild

Automated stage configuration: Enable automatic retry on stage failure Retry mode: Retry failed stage

Step 22: - Now just go to you application and click on release change and here your basic CI/CD complete.

Refer this doc for taking help in CodeDeploy to deploy using GitHub:- Tutorial: Use CodeDeploy to deploy an application from GitHub - AWS CodeDeploy

For other: - CodeDeploy tutorials - AWS CodeDeploy

GitHub repo link of project: - Aayushbijalwan16/AWS-demo: Demo projects of AWS

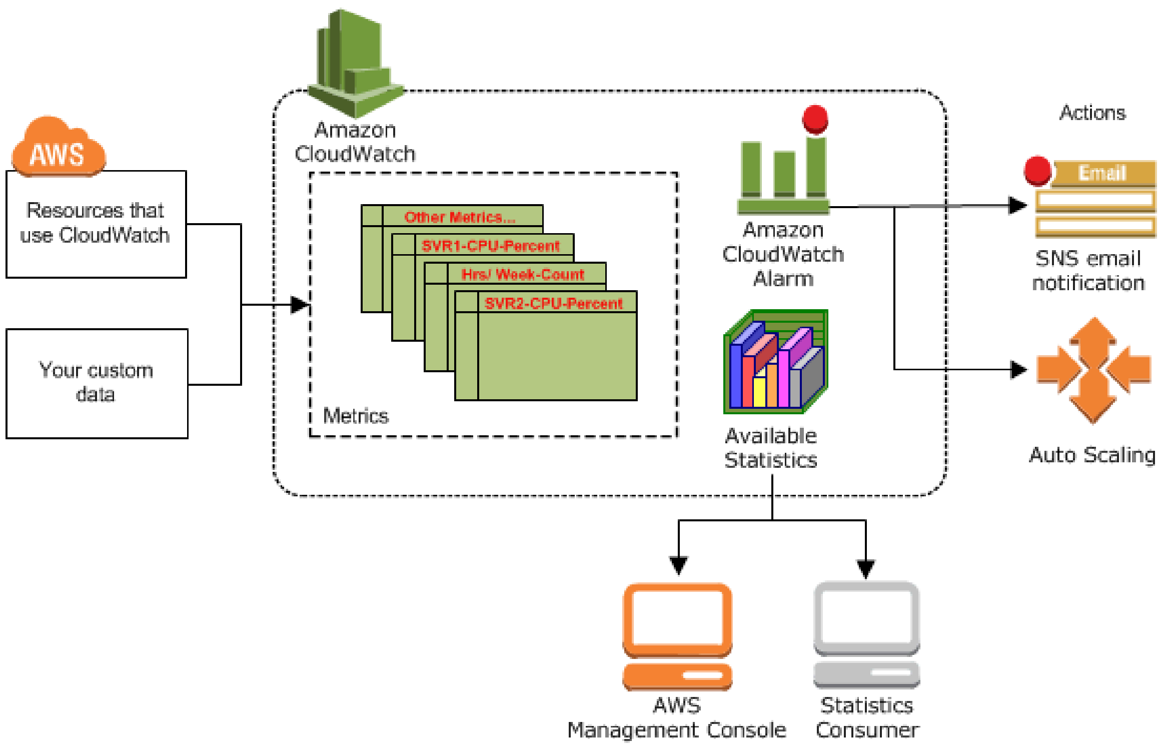
AWS CloudWatch Basic

Concept Overview:

Inntroduction of CloudWatch:

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time, and offers many tools to give you system-wide observability of your application performance, operational health, and resource utilization.

Simple Monitoring Architechture



Benefits of CloudWatch:

Centralized Monitoring: View all AWS resources, apps, and on-premises servers in one place.

Real-Time Metrics & Logs: Get near real-time visibility into performance, errors, and usage.

Automated Alerts & Scaling: Use alarms to send notifications or trigger Auto Scaling actions.

Improves Reliability: Detect failures quickly and troubleshoot using logs, metrics, and dashboards.

Cost Optimization: Analyze usage patterns to optimize resource allocation and reduce costs.

CloudWatch Features:

Metrics: Collects performance metrics from AWS services (e.g., CPU utilization, network traffic) and provides a central repository for them.

- **Custom Metrics:** You can publish your own application-specific metrics to CloudWatch.
- **Metric Streams:** Allows for continuous, near real-time streaming of metrics to a destination of your choice.

Logs: Enables you to centralize, monitor, and analyze log files from various AWS resources, applications, and on-premises servers.

- **Log Groups and Streams:** Logs are organized into groups and streams for easier management and analysis.
- **Metric Filters:** Extracts numerical data from logs and transforms it into CloudWatch metrics.

Alarms: Watches a single metric over a specified time period and takes one or more actions based on the value of the metric.

- **Actions:** Can trigger notifications (via Amazon SNS), and automated actions like scaling resources with Auto Scaling or stopping an EC2 instance.
- **Composite Alarms:** Combines multiple alarms to reduce noise and provide a more holistic view.

CloudWatch Features:

Dashboards: Provides customizable visualizations of your metrics and alarms in a single view.

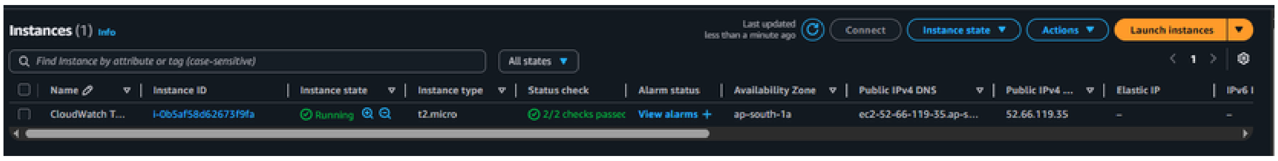
- **Widgets:** You can add various widgets (e.g., line graphs, numbers, gauges) to display key performance indicators (KPIs) and monitor the health of your environment at a glance.
- **Cross-Account Observability:** Allows you to view metrics and logs from multiple AWS accounts on a single dashboard.

Events and Insights:

- **Events (Amazon EventBridge):** Delivers a near real-time stream of system events that describe changes in your AWS resources. You can create rules to automatically route these events to a target.
- **Logs Insights:** A powerful query service for interactively analyzing log data in near real-time, helping you troubleshoot operational issues and identify trends.
- **Container Insights:** Collects, aggregates, and summarizes metrics and logs from containerized applications and microservices (e.g., on Amazon EKS, Amazon ECS, and Kubernetes).
- **Contributor Insights:** Analyzes log data to find the top "contributors" to a metric, helping you identify what is causing a problem.

Hands-on EC2 instance monitoring using CloudWatch Alarms:

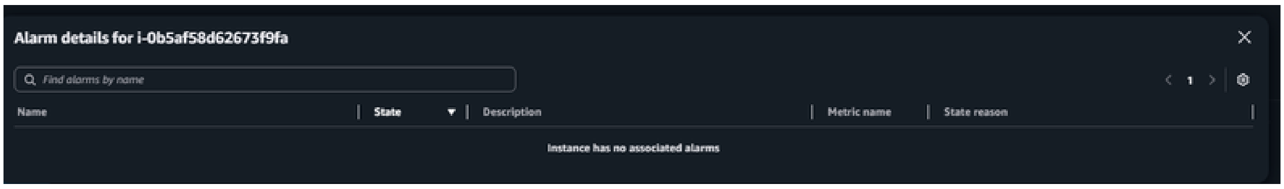
To attach CloudWatch Alarms, you first need to create an EC2 instance.


Instances (1) Info

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 I
CloudWatch T...	i-0b5af58d62673f9fa	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-52-66-119-35.ap-s...	52.66.119.35	–	–

By default, an instance doesn't have any alarms.

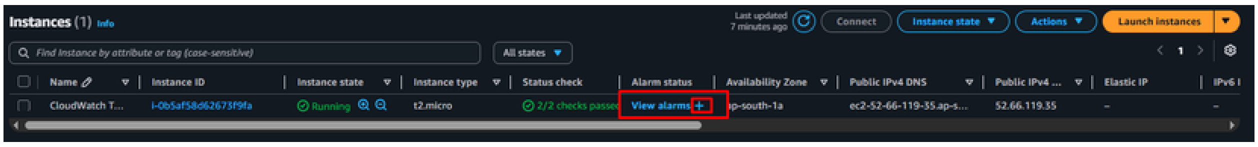

Alarm details for i-0b5af58d62673f9fa

Name	State	Description	Metric name	State reason

Instance has no associated alarms

Now it's time to create an Alarm. You can create it from the instance or from the Alarms section in CloudWatch. In my case, I will create it from CloudWatch.

Create directly from instance:


Instances (1) Info

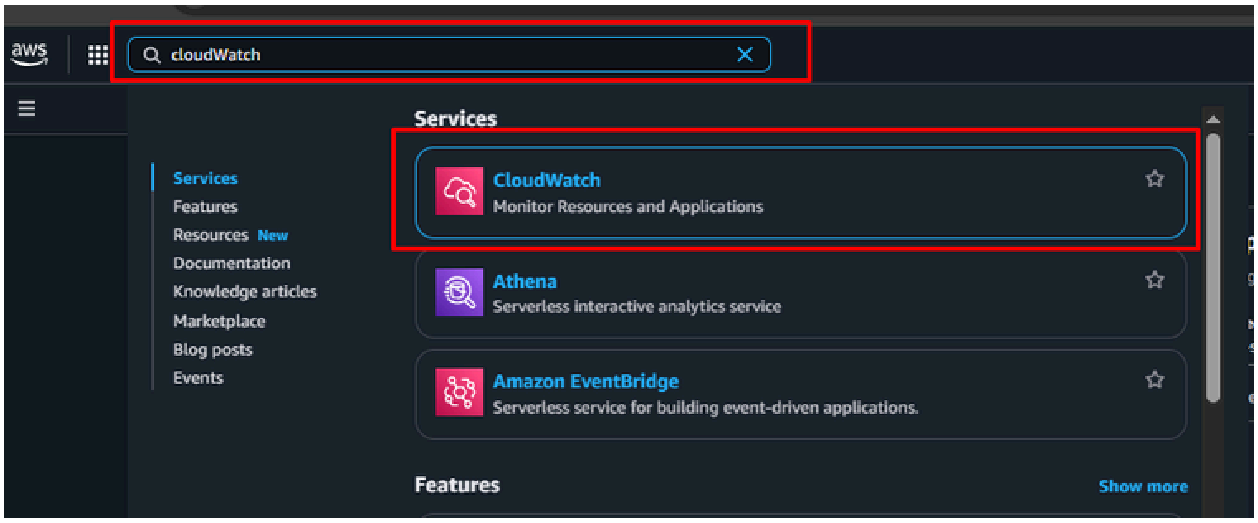
Last updated 7 minutes ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 I
CloudWatch T...	i-0b5af58d62673f9fa	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-52-66-119-35.ap-s...	52.66.119.35	–	–

/in/alamgirweb11

Hands-on EC2 instance monitoring using CloudWatch Alarms:

Create from CloudWatch:
First, search for CloudWatch in your console and click on it.

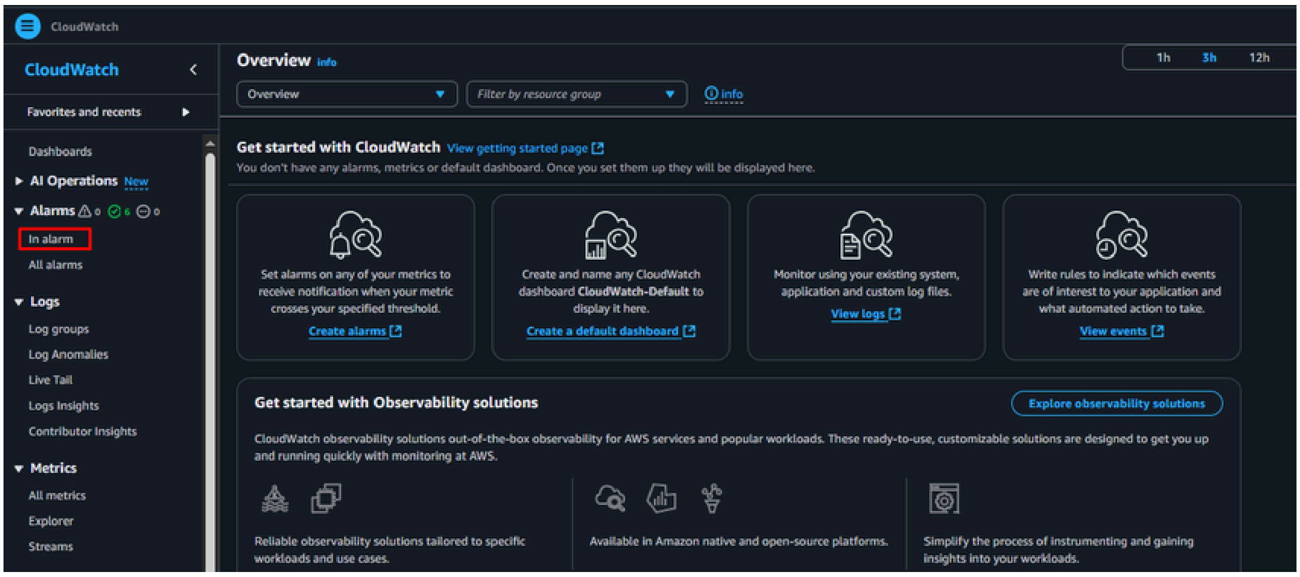


Monitor Resources and Applications

Serverless interactive analytics service

Serverless service for building event-driven applications.

Now click In alarm section.



Get started with CloudWatch

You don't have any alarms, metrics or default dashboard. Once you set them up they will be displayed here.

Set alarms on any of your metrics to receive notification when your metric crosses your specified threshold.

Create and name any CloudWatch dashboard **CloudWatch-Default** to display it here.

Monitor using your existing system, application and custom log files.

Write rules to indicate which events are of interest to your application and what automated action to take.

Get started with Observability solutions

CloudWatch observability solutions out-of-the-box observability for AWS services and popular workloads. These ready-to-use, customizable solutions are designed to get you up and running quickly with monitoring at AWS.

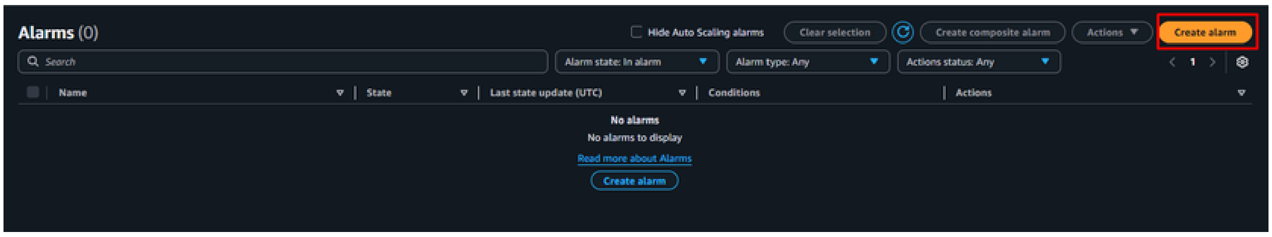
Reliable observability solutions tailored to specific workloads and use cases.

Available in Amazon native and open-source platforms.

Simplify the process of instrumenting and gaining insights into your workloads.

Hands-on EC2 instance monitoring using CloudWatch Alarm:

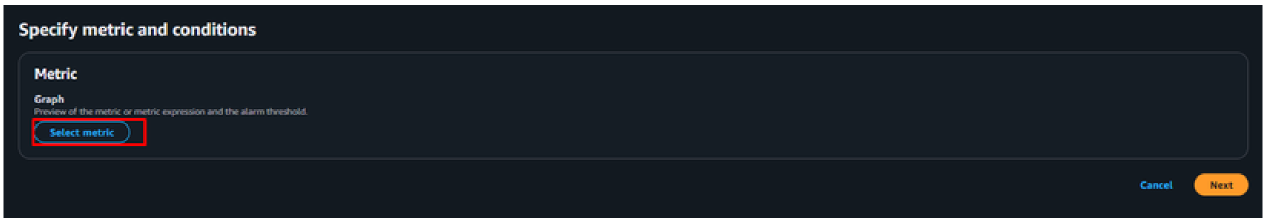
Get this page, now click on create alarms.



Alarm state: In alarm
Alarm type: Any
Actions status: Any

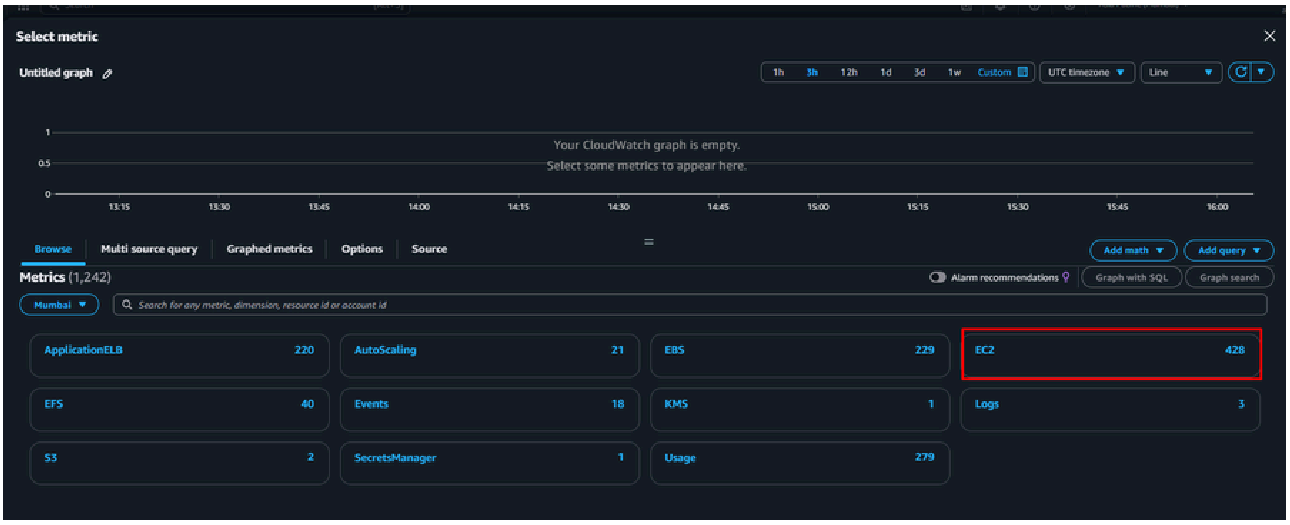
No alarms
No alarms to display

Step 1: Select a metric, click on it.



Preview of the metric or metric expression and the alarm threshold.

Step 2: I'm creating alarms for EC2 that's why I select EC2.



Your CloudWatch graph is empty.
Select some metrics to appear here.

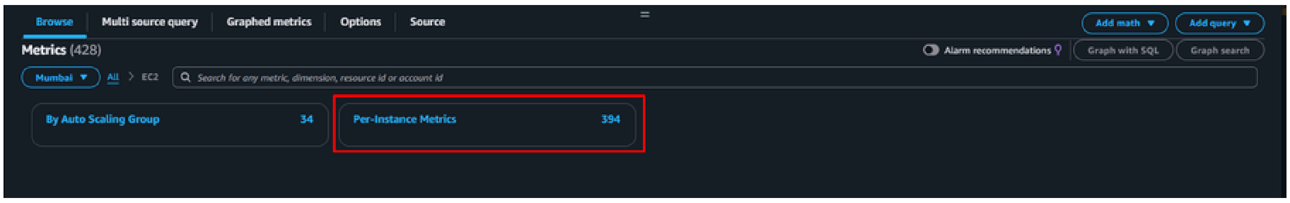
Metrics (1,242)

ApplicationELB	220	AutoScaling	21	EBS	229	EC2	428
EFS	40	Events	18	KMS	1	Logs	3
S3	2	SecretsManager	1	Usage	279		

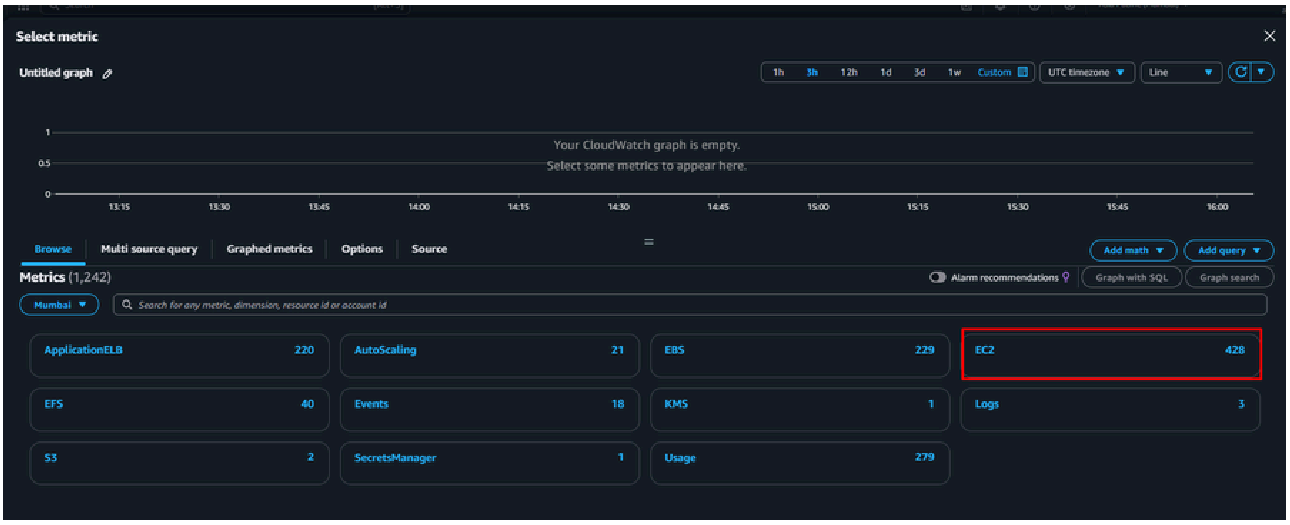
/in/alamgirweb11

Hands-on EC2 instance monitoring using CloudWatch Alarm:

Step 3: Choose based on your needs. In my case, I selected Per-instance metrics, now click on it.



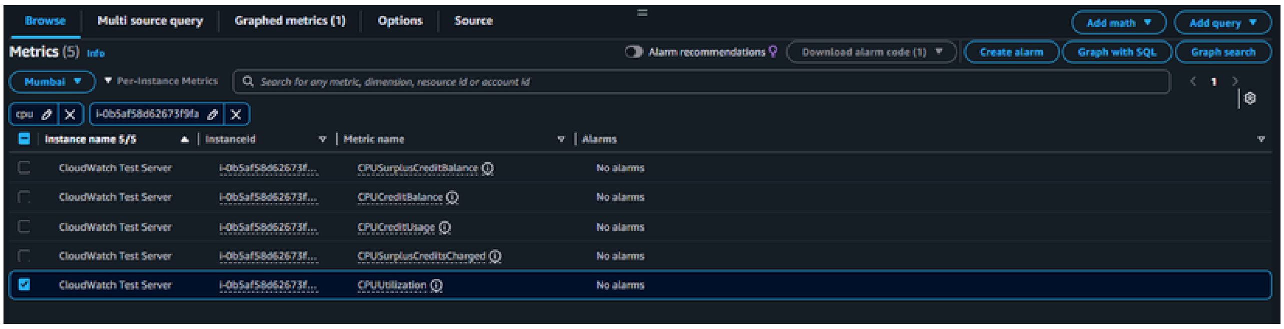
Step 4: Shows this section now choose the service for your metrics i choose EC2.



/in/alamgirweb11

Hands-on EC2 instance monitoring using CloudWatch Alarm:

Step 5: Now, search for your EC2 instance by its instance ID and select the metric you want to add. I'm creating this metric for CPU Utilization monitoring, so I selected that.



Metrics (5) Info

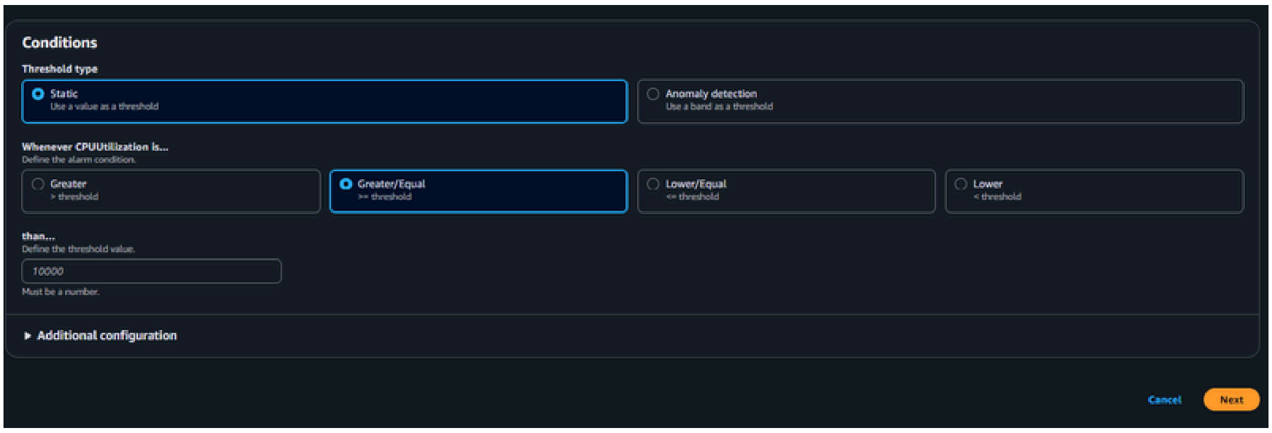
Mumbai ▼ Per-Instance Metrics

Search for any metric, dimension, resource id or account id

cpu | i-0b5af58d62673f9fa

	Instance name 5/5	InstanceId	Metric name	Alarms
☐	CloudWatch Test Server	i-0b5af58d62673f...	CPUSurplusCreditBalance	No alarms
☐	CloudWatch Test Server	i-0b5af58d62673f...	CPUCreditBalance	No alarms
☐	CloudWatch Test Server	i-0b5af58d62673f...	CPUCreditUsage	No alarms
☐	CloudWatch Test Server	i-0b5af58d62673f...	CPUSurplusCreditsCharged	No alarms
✓	CloudWatch Test Server	i-0b5af58d62673f...	CPUUtilization	No alarms

Step 6: In condition section, select Threshold type and Whenever CPUUtilization condition. Don't forget to set threshold value is required.



Conditions

Threshold type

[✓ Static — Use a value as a threshold] Anomaly detection — Use a band as a threshold

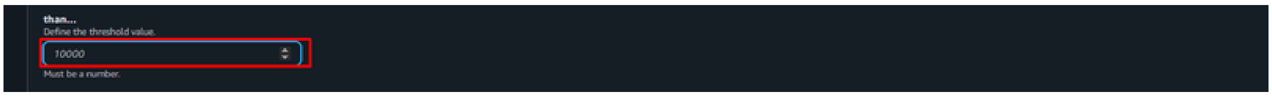
Whenever CPUUtilization is...
Define the alarm condition.

Greater — > threshold | [✓ Greater/Equal — >= threshold] | Lower/Equal — <= threshold | Lower — < threshold

than...
Define the threshold value.

10000

Must be a number.



than...
Define the threshold value.

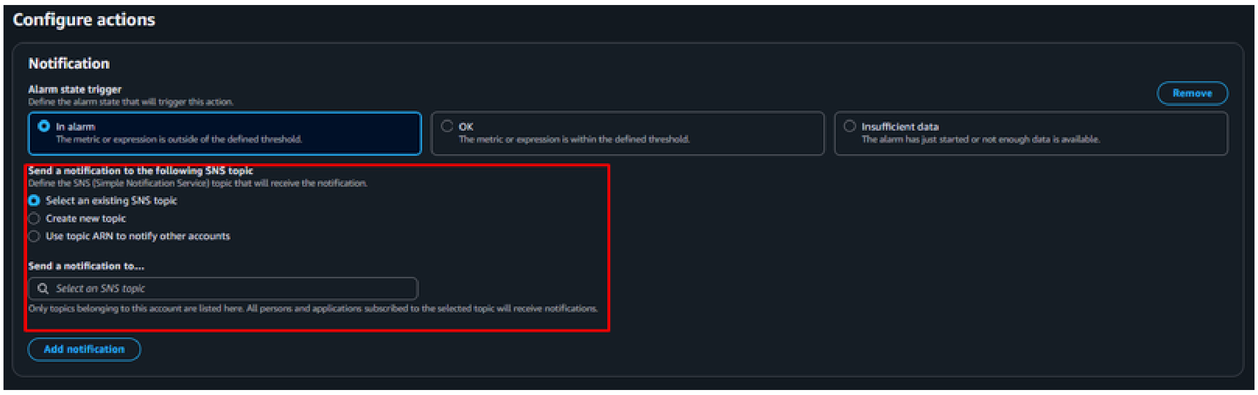
10000

Must be a number.

/in/alamgirweb11

Hands-on EC2 instance monitoring using CloudWatch Alarm:

Step 7: Choose or create a SNS for getting notification. And Select action.



Configure actions

Notification

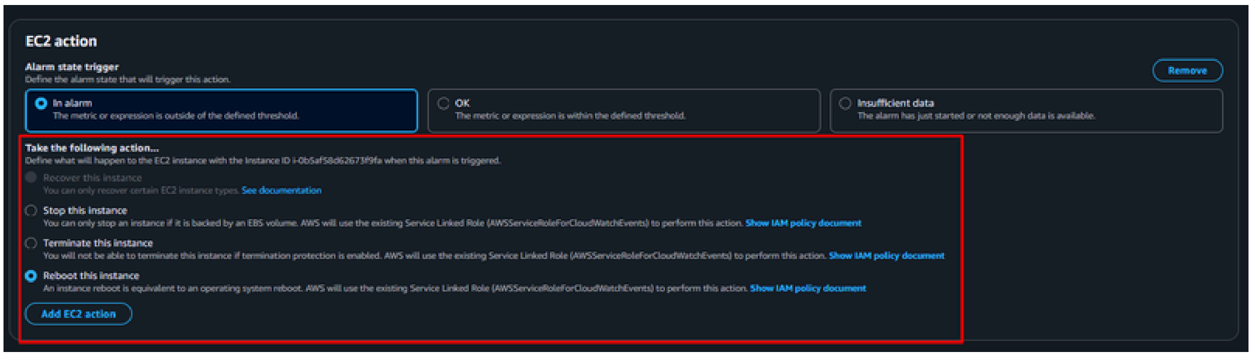
Alarm state trigger
Define the alarm state that will trigger this action.

- [✓ In alarm] The metric or expression is outside of the defined threshold.
- OK — The metric or expression is within the defined threshold.
- Insufficient data — The alarm has just started or not enough data is available.

Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

- [✓ Select an existing SNS topic]
- Create new topic
- Use topic ARN to notify other accounts

Send a notification to...
Select an SNS topic
Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.



EC2 action

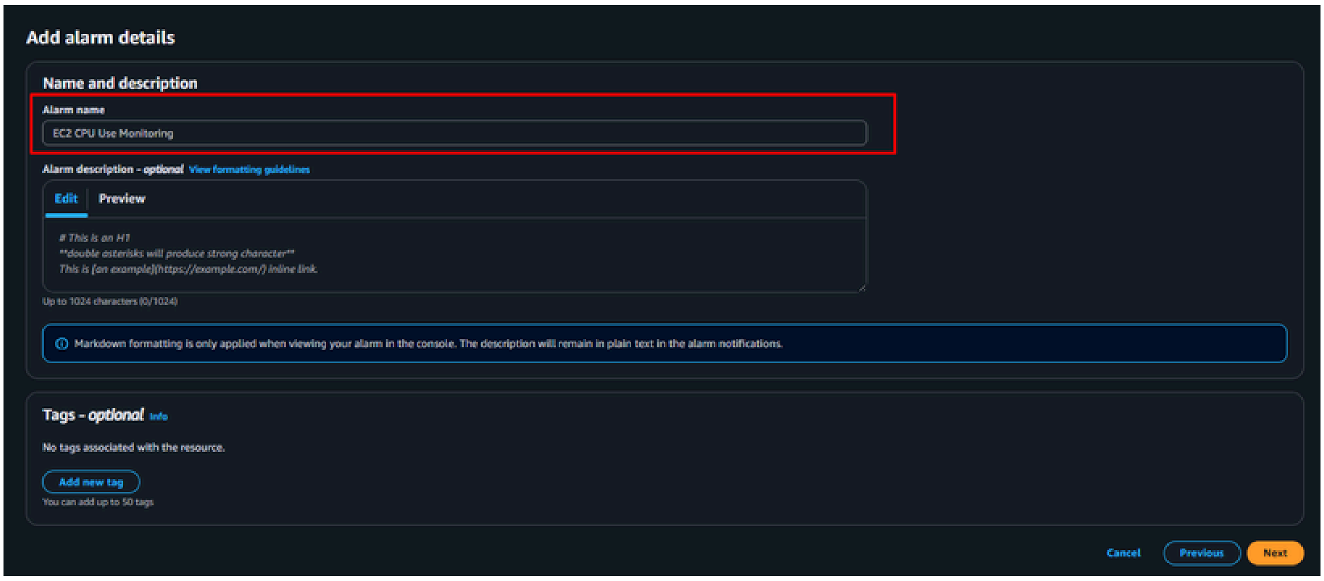
Alarm state trigger
Define the alarm state that will trigger this action.

- [✓ In alarm] The metric or expression is outside of the defined threshold.
- OK — The metric or expression is within the defined threshold.
- Insufficient data — The alarm has just started or not enough data is available.

Take the following action...
Define what will happen to the EC2 instance with the Instance ID i-0b5af58d62673f9fa when this alarm is triggered.

- Recover this instance — You can only recover certain EC2 instance types. See documentation
- Stop this instance — You can only stop an instance if it is backed by an EBS volume. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. Show IAM policy document
- Terminate this instance — You will not be able to terminate this instance if termination protection is enabled. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. Show IAM policy document
- [✓ Reboot this instance] An instance reboot is equivalent to an operating system reboot. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. Show IAM policy document

Step 8: In this step, type a name for your alarms.



Add alarm details

Name and description

Alarm name: EC2 CPU Use Monitoring

Alarm description - *optional* View formatting guidelines

```
# This is an H1
**double asterisks will produce strong character**
This is [an example](https://example.com/) inline link.
```

Up to 1024 characters (0/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Tags - *optional* Info

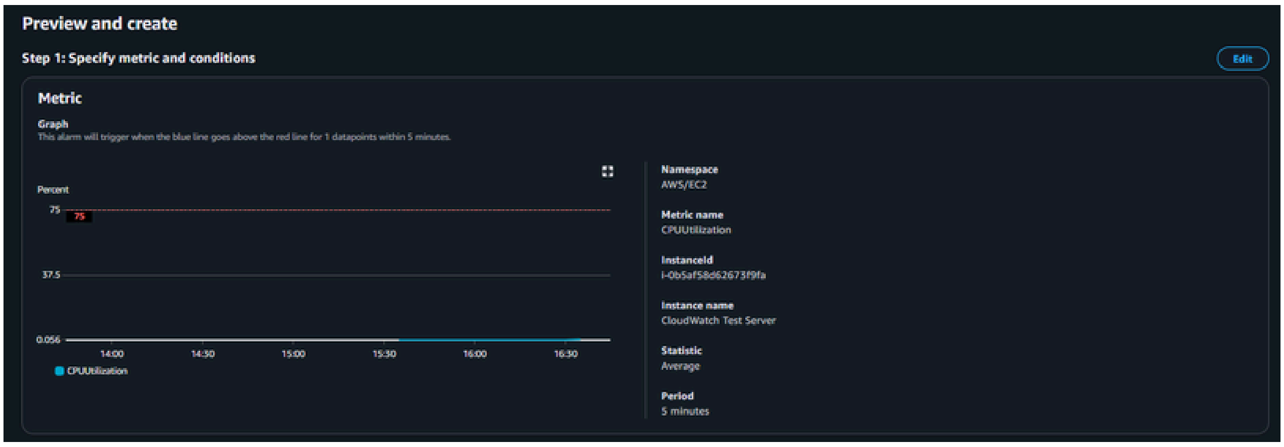
No tags associated with the resource.

You can add up to 50 tags

/in/alamgirweb11

Hands-on EC2 instance monitoring using CloudWatch Alarm:

Step 9: Now, review your configuration and create it.



Preview and create

Step 1: Specify metric and conditions

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

Namespace
AWS/EC2

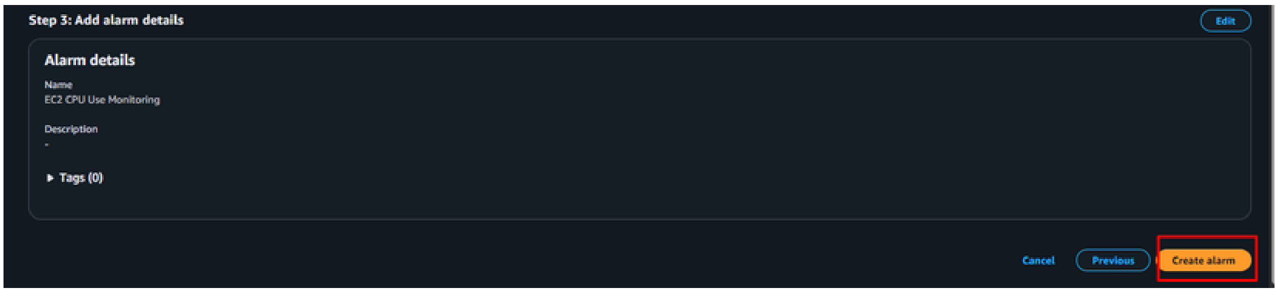
Metric name
CPUUtilization

InstanceId
i-0b5af58d62673f9fa

Instance name
CloudWatch Test Server

Statistic
Average

Period
5 minutes



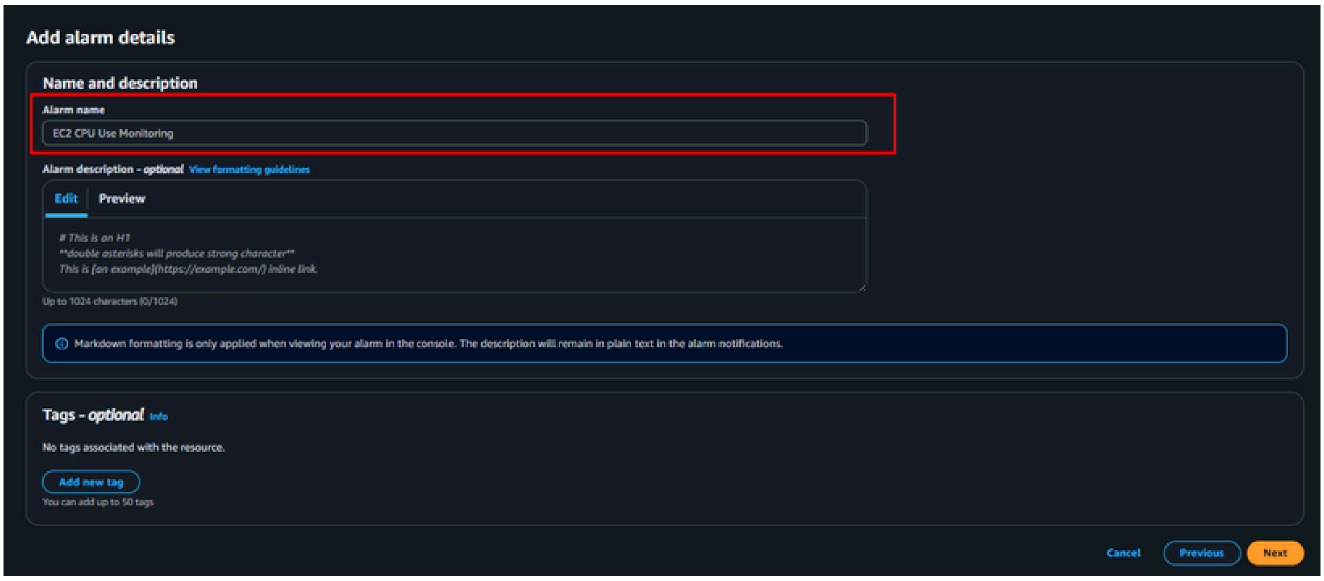
Step 3: Add alarm details

Alarm details

Name
EC2 CPU Use Monitoring

Description
-

Step 8: In this step, type a name for your alarms.



Add alarm details

Name and description

Alarm name
EC2 CPU Use Monitoring

Alarm description - *optional* View formatting guidelines

This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.

Up to 1024 characters (0/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

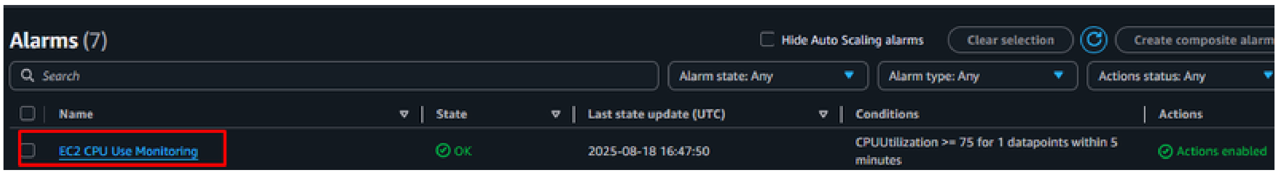
Tags - *optional* Info

No tags associated with the resource.

You can add up to 50 tags

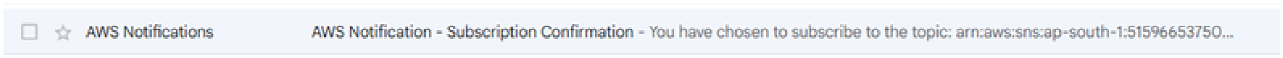
Hands-on EC2 instance monitoring using CloudWatch Alarm:

Step 10: Our alarm is created successfully.

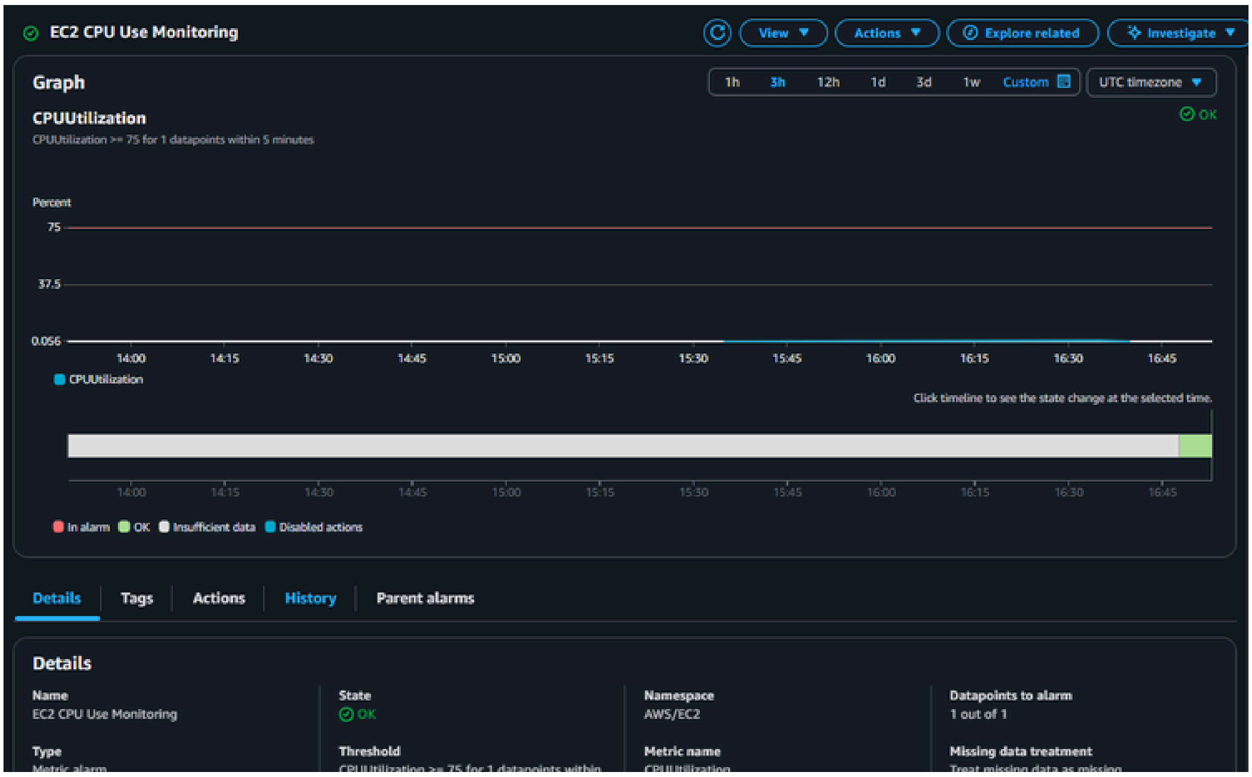


Name	State	Last state update (UTC)	Conditions	Actions
EC2 CPU Use Monitoring	OK	2025-08-18 16:47:50	CPUUtilization >= 75 for 1 datapoints within 5 minutes	Actions enabled

You also get a email which you use for SNS.


AWS Notifications AWS Notification - Subscription Confirmation - You have chosen to subscribe to the topic: arn:aws:sns:ap-south-1:51596653750...

Now, you can view the details of your alarm.

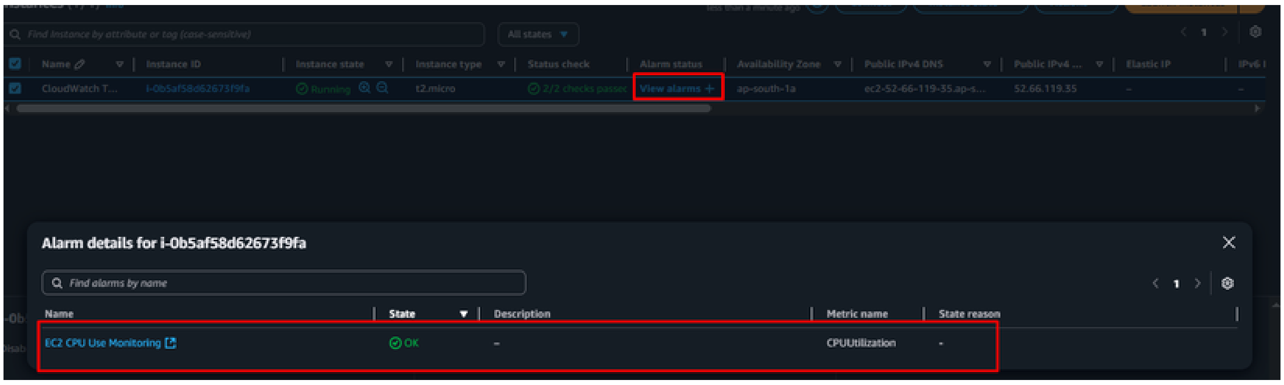


Details

Name	State	Namespace	Datapoints to alarm
EC2 CPU Use Monitoring	OK	AWS/EC2	1 out of 1
Type	**Threshold**	**Metric name**	**Missing data treatment**
Metric alarm	CPUUtilization >= 75 for 1 datapoints within	CPUUtilization	Treat missing data as missing

Hands-on EC2 instance monitoring using CloudWatch Alarm:

Go back to your EC2 instance and verify that it is attached successfully.



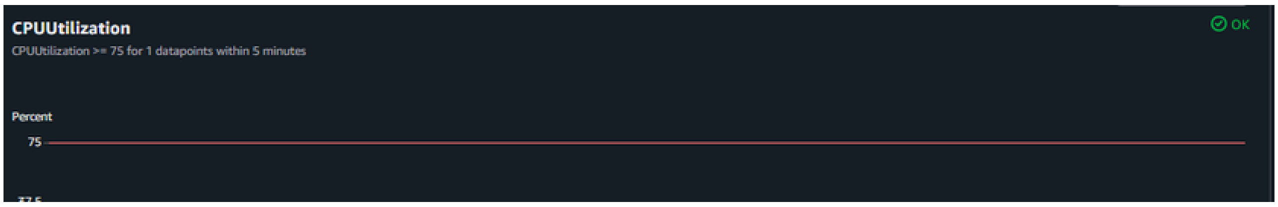
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 I
CloudWatch T...	i-0b5af58d62673f9fa	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-52-66-119-35.ap-s...	52.66.119.35	–	–

Alarm details for i-0b5af58d62673f9fa

Name	State	Description	Metric name	State reason
EC2 CPU Use Monitoring	OK	–	CPUUtilization	-

In my case, it shows the newly created Alarm, which means we successfully created it. Now it is monitoring CPU utilization based on the Alarm configuration. If any issue occurs, it will send an email and reboot the instance, as I set earlier.

I set it to 75% CPU utilization — when this threshold is reached, it will reboot the instance and send me an email via SNS(Simple Notification Service).



CPUUtilization

CPUUtilization >= 75 for 1 datapoints within 5 minutes

Thank You

Stay Connect:

/in/alamgirweb11

/alamgirweb11