

CI/CD Pipeline Setup with AWS Services

By Tushar Gupta

Objective:

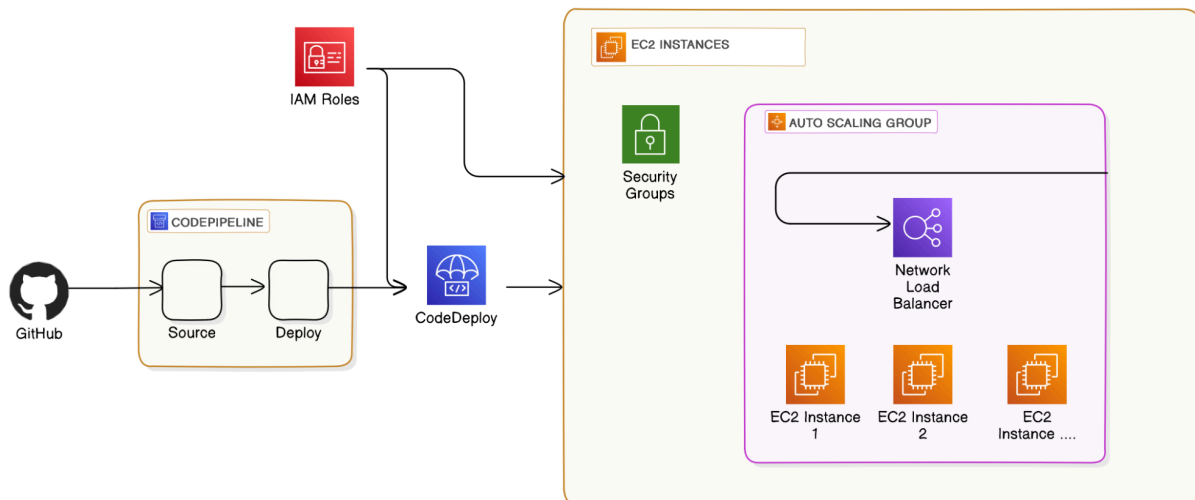
The goal is to set up a CI/CD pipeline that automates the deployment of a simple web application to a server. The pipeline should trigger on code pushes to a repository and deploy the application in a scalable manner. When load increases, the number of servers should automatically scale up or down, ensuring that new servers have the latest code.

Github Repo : https://github.com/cstushar/ci_cd_repo.git

Tools Used:

1. **AWS CodeDeploy**
2. **AWS Auto Scaling**
3. **AWS Load Balancing**
4. **AWS CodePipeline**
5. **AWS EC2**
6. **Git/Bitbucket**

Basic CI/CD Pipeline



Step-by-Step Process

1. Create a Launch Instance Template in EC2

- **Operating System:** Ubuntu (Free tier)
- **Instance Type:** T2 Micro
- **Key Pair:** Enabled
- **Security Group:**
 - Allow SSH and HTTP access from anywhere.

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

MyTemplate

Must be unique to this account. Max 128 chars. No spaces or special characters like @, %, ^, &.

Template version description

A prod webserver for MyApp

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags

► Source template

Summary

Software Image (AMI)

-

Virtual server type (instance type)

-

Firewall (security group)

-

Storage (volumes)

-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million

Cancel **Create launch template**

Quick Start

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

Recents **Quick Start**

Don't include in launch template

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-003932de22c285676 (64-bit (x86)) / ami-03772d93fb1879bbe (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 22.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Architecture

64-bit (x86)

AMI ID

ami-003932de22c285676 **Verified provider**

Summary

Firewall (security group)

-

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Create launch template**

aws

Services

Search

[All+5]

Ohio

Tushar Gupta

▼ Instance type

Info | Get advice

Advanced

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0116 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand RHEL base pricing: 0.026 USD per Hour

Free tier eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login)

Info

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

Tushar_Key

Create new key pair

▼ Network settings

Info

▼ Summary

t2.micro

Firewall (security group)

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Create launch template

CloudShell

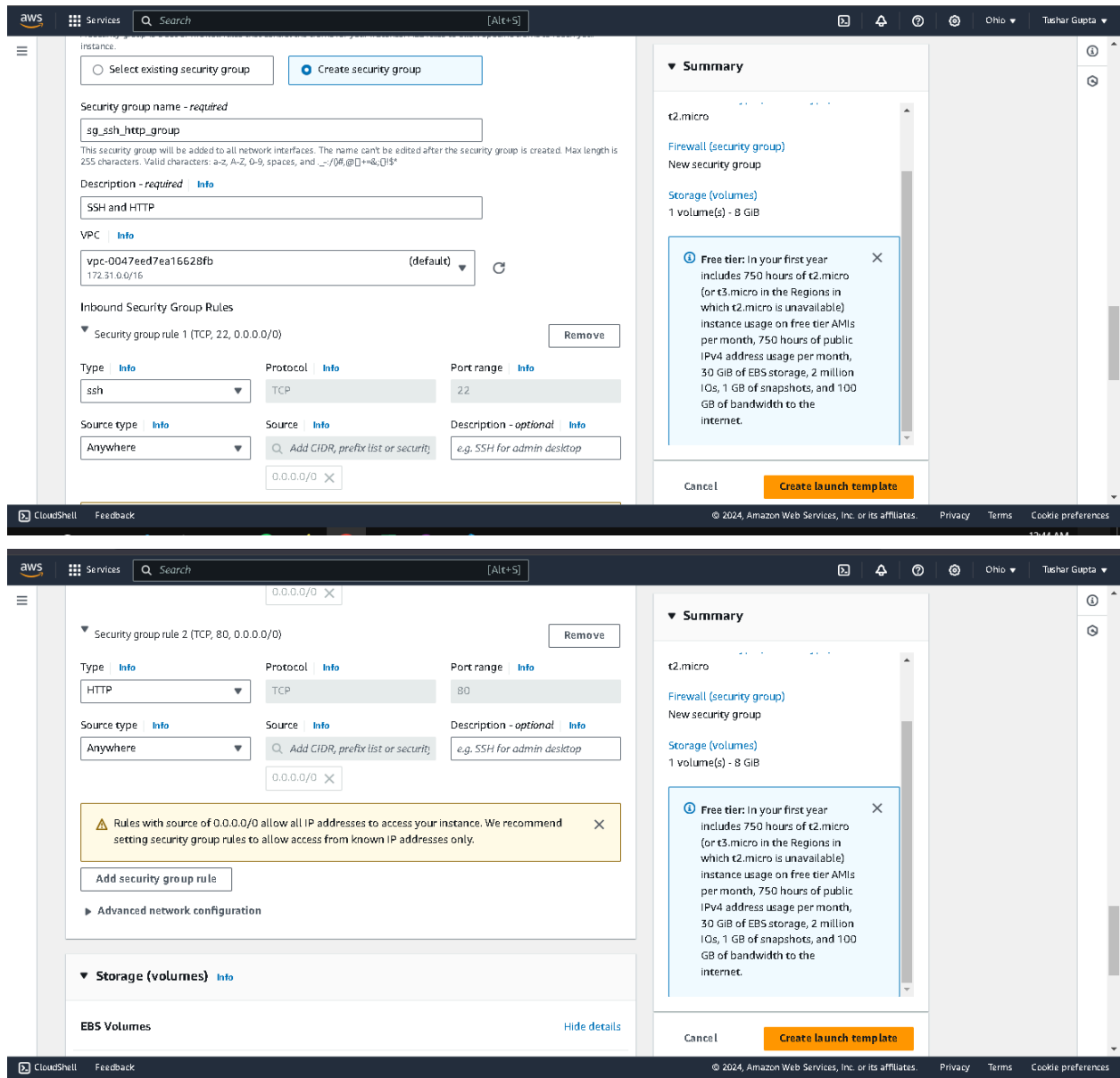
Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences



2. Create an IAM Role for EC2

- **IAM Role Type:** EC2
- **Permissions:** AmazonEC2RoleforAWSCodeDeploy
- **Assign the Role:** Select this IAM role when creating the EC2 launch template.

us-east-1.console.aws.amazon.com/iam/home?region=us-east-2#/roles

Services

Search

[Alt+S]

Global

Tushar Gupta

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

- Access Analyzer
- External access
- Unused access

IAM > Roles

Roles (5) Info

Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

< 1 >

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Rol	1 hour ago
<input type="checkbox"/>	AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (S	32 minutes
<input type="checkbox"/>	AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Servi	-
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linked	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-	-

Roles Anywhere Info

Manage

Authenticate your non AWS workloads and securely provide access to AWS services.

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Services

Search

[Alt+S]

Global

Tushar Gupta

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Select trusted entity Info

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EC2

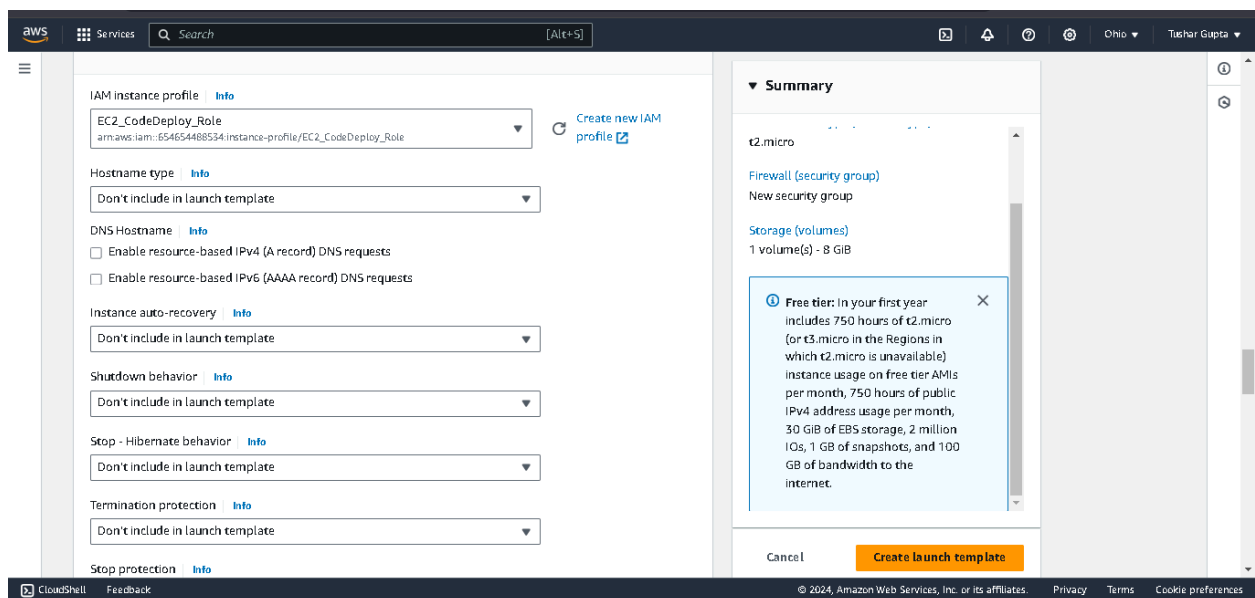
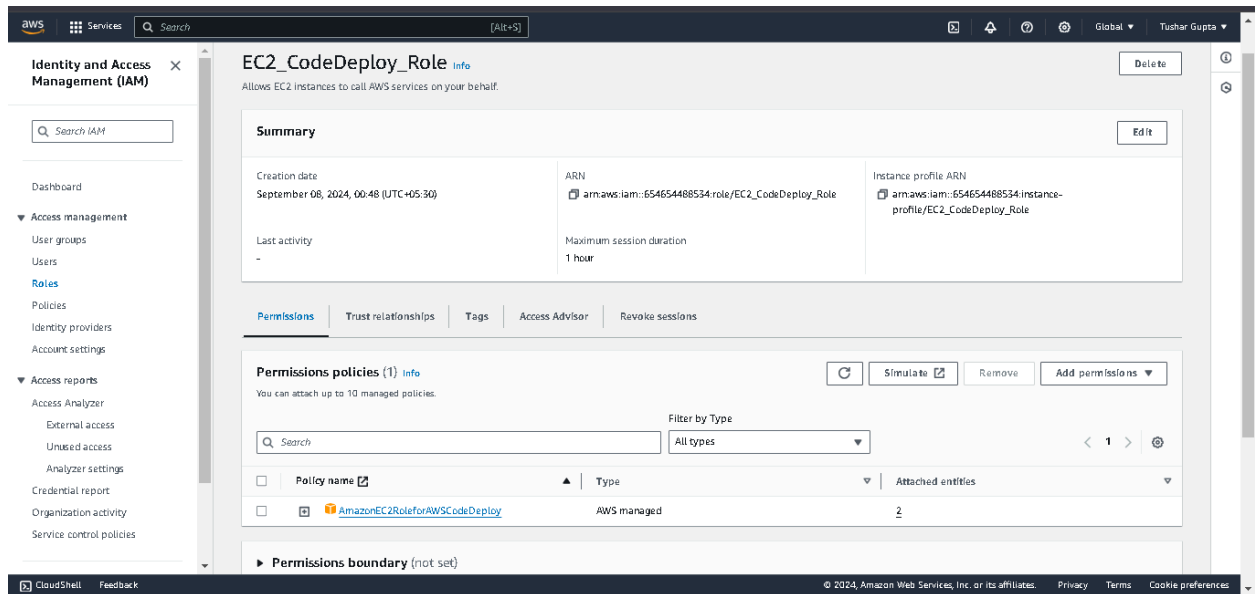
Choose a use case for the specified service.

Use case

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



3. User Data for EC2 Instance

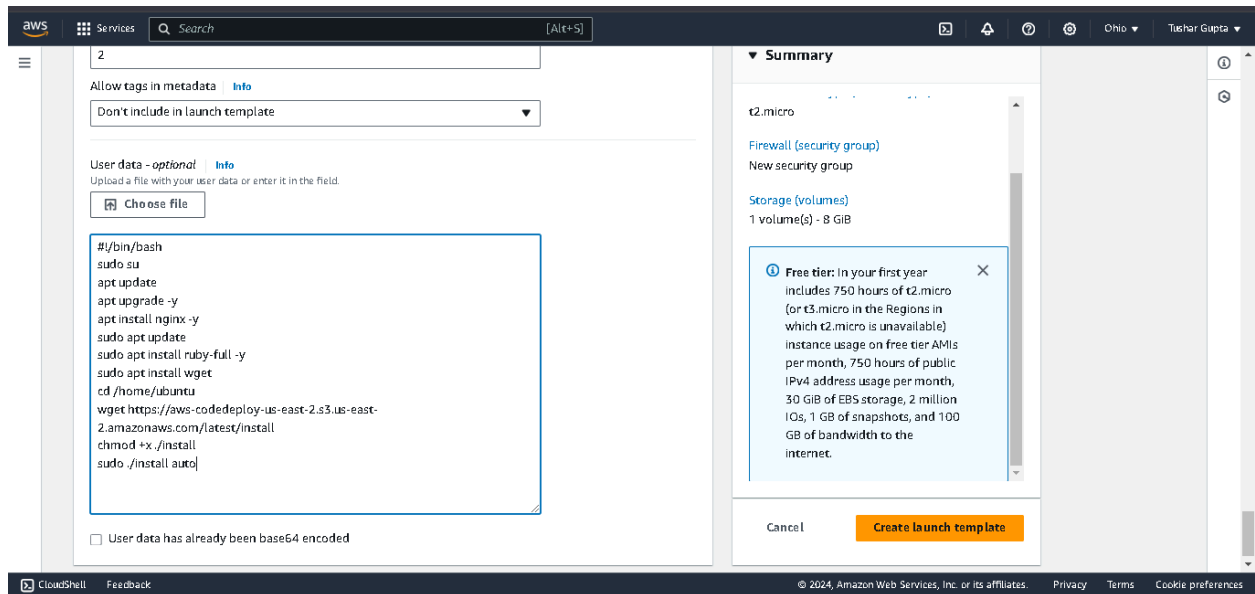
- In the User Data section, provide a script that automatically installs the CodeDeploy agent when the instance is launched. This ensures that the CodeDeploy agent is set up without manual intervention.

```
#!/bin/bash
sudo su
apt update
apt upgrade -y
apt install nginx -y
sudo apt update
sudo apt install ruby-full -y
```

```

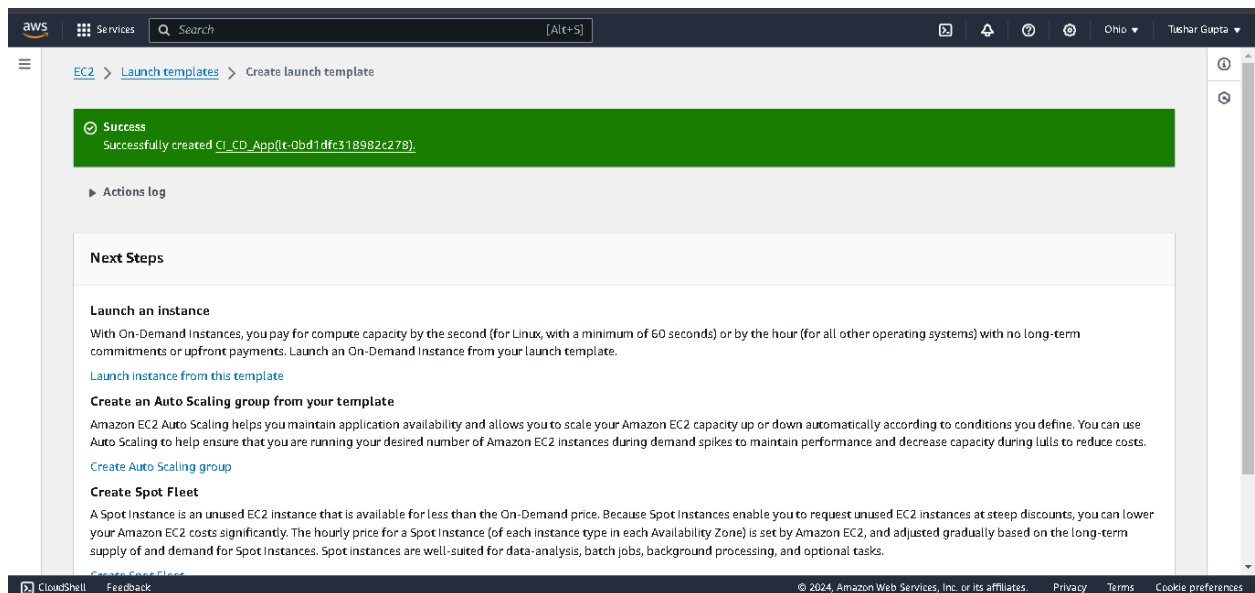
sudo apt install wget
cd /home/ubuntu
wget https://aws-coddeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto

```



4. Create the Launch Template

- After configuring the above settings, create the EC2 launch template.



Auto Scaling Configuration

1. Create an Auto Scaling Group

- **Name:** CICDApp (or another preferred name)
- **Launch Template:** Select the template created earlier.
- **Availability Zones:** Choose all available zones for your region.

aws

Services

Search

[Alt+S]

Ohio

Tushar Gupta

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

Load Balancers

Target Groups

Trust Stores [New](#)

▼ Auto Scaling

Auto Scaling Groups

Settings

Amazon EC2 Auto Scaling

helps maintain the availability of your applications

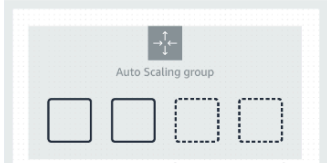
Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

Create Auto Scaling group

How it works



Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

Search

[Alt+S]

Ohio

Tushar Gupta

Step 3 – optional

Configure advanced options

Step 4 – optional

Configure group size and scaling

Step 5 – optional

Add notifications

Step 6 – optional

Add tags

Step 7

Review

Auto Scaling group name

Enter a name to identify the group.

EC2_Auto_Scaling_Group

Must be unique to this account in the current Region and no more than 255 characters.

Launch template [info](#)

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

CL_CD_App

Create a launch template

Version

Default (1)

Create a launch template version

Description	Launch template	Instance type
-	CL_CD_App	t2.micro
	lt-0bd1dfc318982c278	
AMI ID	Security groups	Request Spot Instances
ami-005952de22c285676	-	No

CloudShell

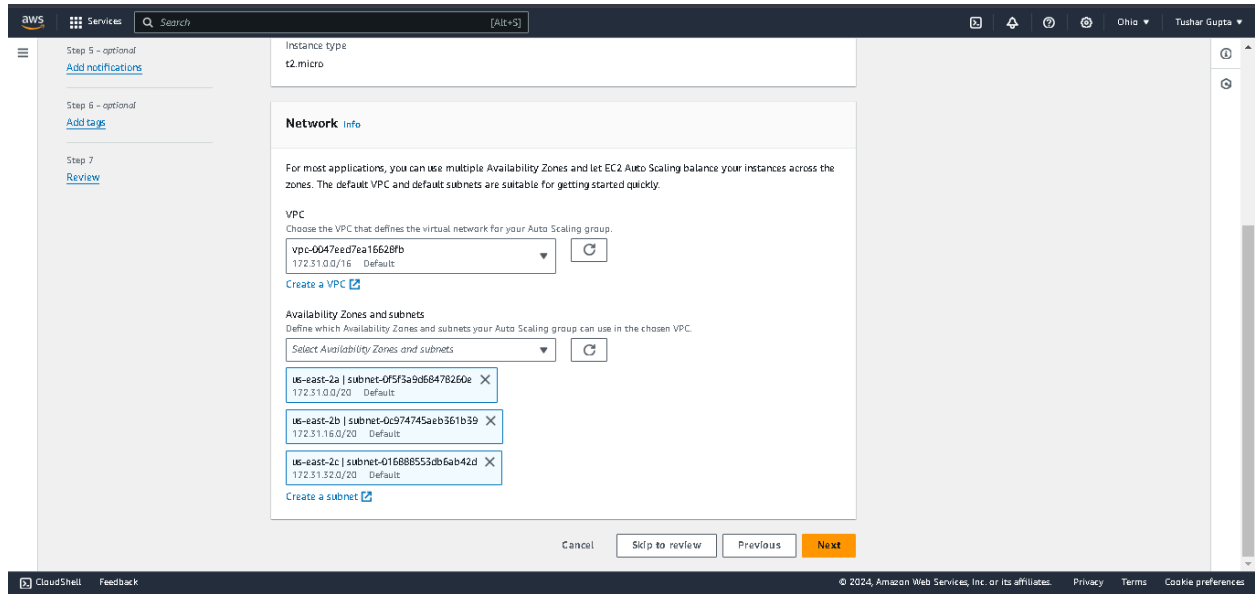
Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences



2. Load Balancer Setup

- **Type:** Network Load Balancer
- **Visibility:** Internet-facing
- **Listeners:** Create a target group and assign it a name.

aws

Services

Search

[Alt+S]

Ohio

Step 3 - optional

Configure advanced options

Step 4 - optional

Configure group size and scaling

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

Load balancing info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer

Traffic to your Auto Scaling group will not be fronted by a load balancer.

☐ Attach to an existing load balancer

Choose from your existing load balancers.

☒ Attach to a new load balancer

Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

☐ Application Load Balancer

HTTP, HTTPS

☒ Network Load Balancer

TCP, UDP, TLS

Load balancer name

Name cannot be changed after the load balancer is created.

EC2_Auto_Scaling_Group-1

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

☐ Internal

☒ Internet-facing

Network mapping

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

aws

Services

Search

[Alt+S]

VPC-LAP/EEU/ea16b2bd

Availability Zones and subnets

You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

☒ us-east-2c

subnet-01688553db6ab42d

☒ us-east-2b

subnet-0c974745aeb361b39

☒ us-east-2a

subnet-0f5f3a9d684782e0e

Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol

Port

Default routing (forward to)

TCP

80

Create a target group

New target group name

An instance target group with default settings will be created.

EC2AutoScalingTargetGroup

Tags - optional

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add tag

50 remaining

VPC Lattice integration options info

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

P

aws Services Search [Alt+S]

Choose instance launch options

Step 3 - optional
[Configure advanced options](#)

Step 4 - optional
Configure group size and scaling

Step 5 - optional
[Add notifications](#)

Step 6 - optional
[Add tags](#)

Step 7
[Review](#)

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances) ▼

Desired capacity
Specify your group size.
0

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity
0
Equal or less than desired capacity

Max desired capacity
0
Equal or greater than desired capacity

Automatic scaling - optional
Choose whether to use a [target tracking policy](#) [Info](#)
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☒ **No scaling policies**
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

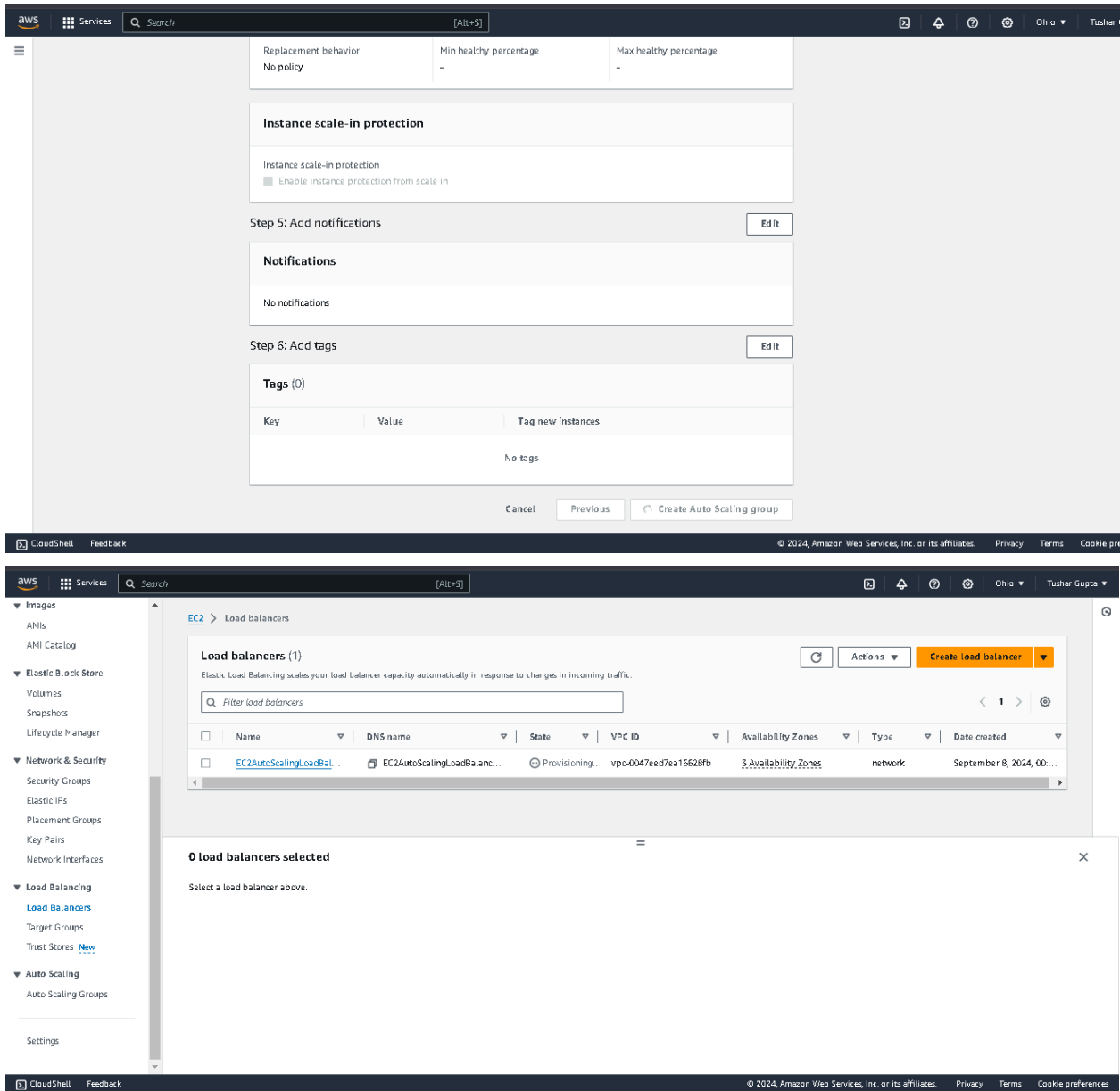
☐ **Target tracking scaling policy**
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms

Note: Initially set the desired capacity to 0, as you don't need to start instances yet.

3. Verify the Load Balancer

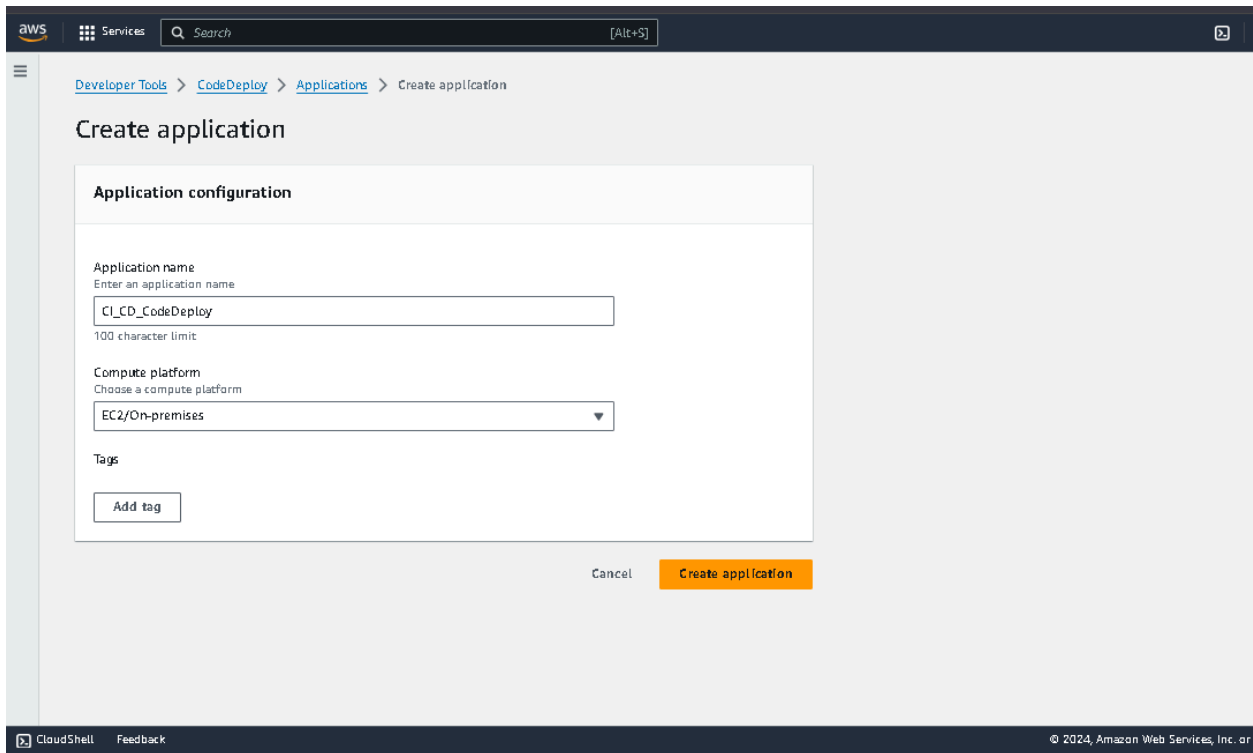
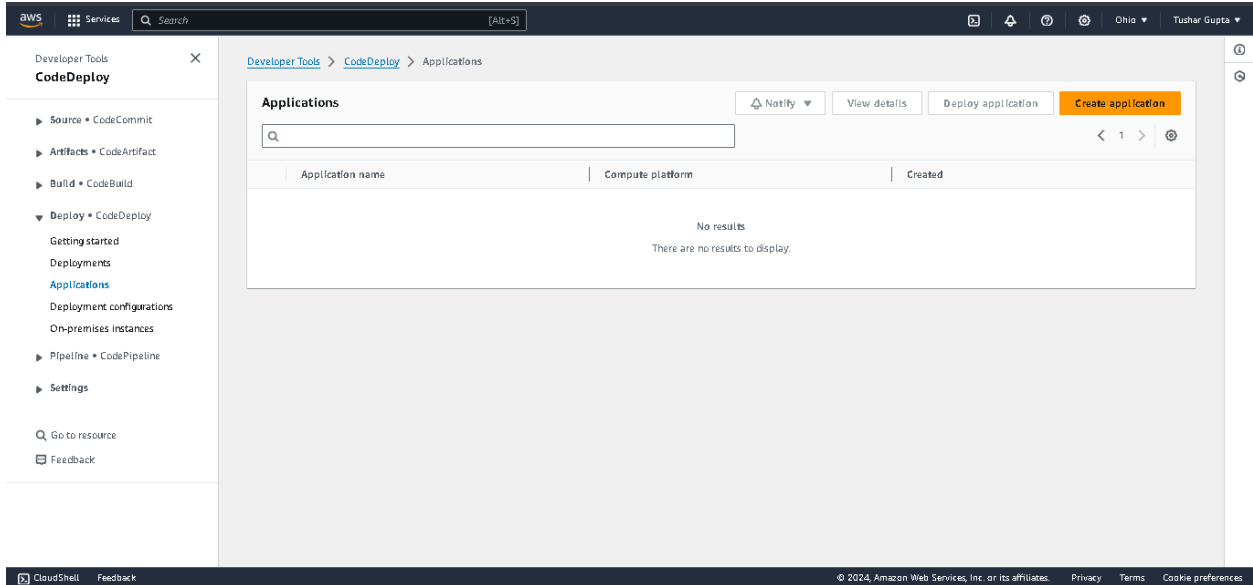
- Check the Load Balancer dashboard to ensure the Load Balancer is created and is in the provisioning state. It will eventually transition to an active state.

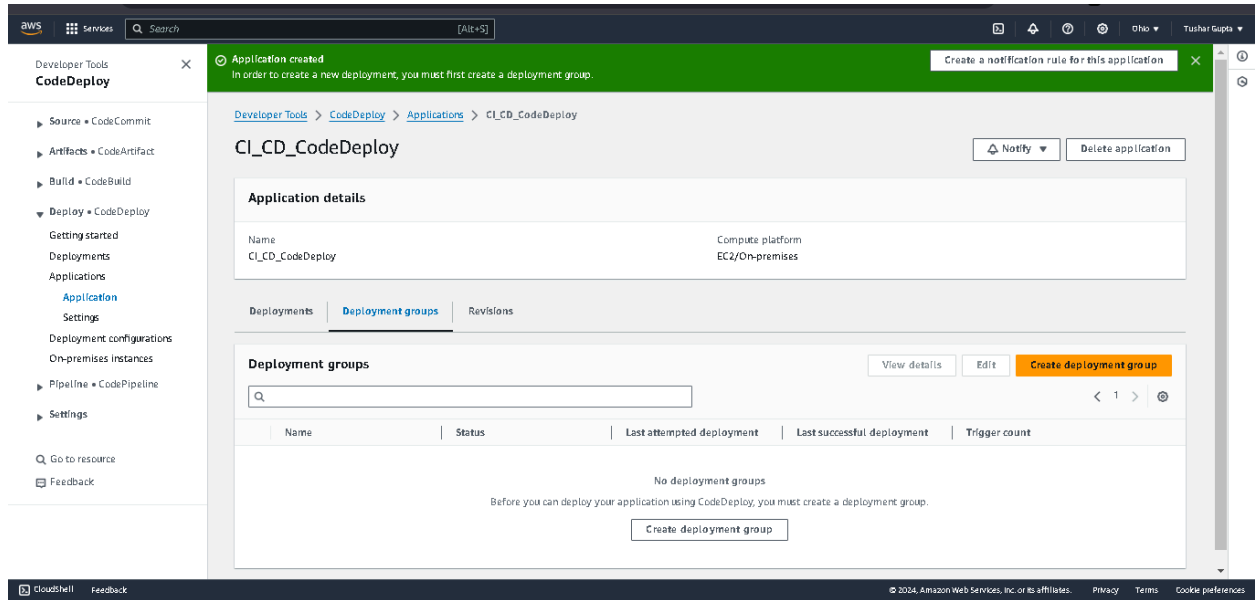


CodeDeploy Setup

1. Create a CodeDeploy Application

- **Application Name:** Choose a name.
- **Compute Platform:** EC2/on-premises





2. Create an IAM Role for CodeDeploy

- **Service:** CodeDeploy
- **Permissions:** AWSCodeDeployRole

The top screenshot shows the 'Select trusted entity' step in the AWS IAM console. The 'Trusted entity type' section has 'AWS service' selected. The 'Use case' section has 'CodeDeploy' selected. The bottom screenshot shows the 'Summary' and 'Permissions policies' for the newly created 'CI_CD_CodeDeployRole'. The summary shows the creation date as September 08, 2024, 01:06 (UTC+05:30) and the ARN as arn:aws:iam::654654486534:role/CI_CD_CodeDeployRole. The permissions policies section shows one policy attached: 'AWSCodeDeployRole'.

3. Create a Deployment Group

- **Deployment Group Name:** Choose a name.
- **Service Role:** CodeDeploy role with the necessary permissions.
- **Amazon EC2 Auto Scaling Group:** Select the group created earlier.
- **Deployment Settings:** Use the `CodeDeployDefault.OneAtATime` deployment configuration to ensure that instances remain healthy.
- **Load Balancer:** Enable load balancing and select the appropriate Load Balancer target group.

aws

Services

Search

[Alt+S]

Developer Tools > CodeDeploy > Applications > CI_CD_CodeDeploy > Create deployment group

Create deployment group

Application

Application
CI_CD_CodeDeploy
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name

CI_CD_CodeDeployGroup

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::654654488534:role/CI_CD_CodeDeployRole

Deployment type

CloudShell

Feedback

© 2024, Amazon

aws

Services

Search

[Alt+S]

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
0 unique matched instances. [Click here for details](#)

You can select up to 10 Amazon EC2 Auto Scaling groups to deploy your application revision to.

EC2_Auto_Scaling_Group

► Termination hook

☐ Amazon EC2 instances
☐ On-premises instances

Deployment settings

Deployment configuration
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

CodeDeployDefault:OneAtATime or Create deployment configuration

CloudShell

Feedback

© 2024, Amazon

The image shows two screenshots from the AWS CodeDeploy console. The top screenshot displays the 'Deployment settings' section where 'CodeDeployDefault:OneAtATime' is selected as the deployment configuration. The 'Load balancer' section shows 'Enable load balancing' checked, with 'Application Load Balancer or Network Load Balancer' selected as the load balancer type and 'EC2AutoScalingTargetGroup' as the target group. The bottom screenshot shows a green success banner stating 'Deployment group created' for 'CI_CD_CodeDeployGroup'. Below the banner, the 'Deployment group details' table lists the group name, application name, compute platform, deployment type, service role ARN, and agent update scheduler. The 'Environment configuration: Amazon EC2 Auto Scaling groups' section shows the name 'EC2_Auto_Scaling_Group' and a disabled termination hook. The left sidebar shows the navigation menu with 'CodeDeploy' selected.

Deployment settings

Deployment configuration
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

CodeDeployDefault:OneAtATime or Create deployment configuration

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

Load balancer type
☒ Application Load Balancer or Network Load Balancer

Choose target groups

EC2AutoScalingTargetGroup X

☐ Classic Load Balancer

▼ Advanced - optional

Triggers Delete trigger Create trigger

Success
Deployment group created

Developer Tools > CodeDeploy > Applications > CI_CD_CodeDeploy > CI_CD_CodeDeployGroup

CI_CD_CodeDeployGroup Edit Delete Create deployment

Deployment group details

Deployment group name CI_CD_CodeDeployGroup	Application name CI_CD_CodeDeploy	Compute platform EC2/On-premises
Deployment type In-place	Service role ARN arn:aws:iam::654654488534:role/CI_CD_CodeDeployRole	Deployment configuration CodeDeployDefault:OneAtATime
Rollback enabled False	Agent update scheduler Learn to schedule update in AWS Systems Manager	

Environment configuration: Amazon EC2 Auto Scaling groups

Name EC2_Auto_Scaling_Group
Termination hook Disabled

Triggers

© 2024 Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CodePipeline Setup

1. Create a Pipeline

- **Pipeline Name:** Choose a name.

- **Source Provider:** GitHub (Version 2)
 - Connect to GitHub by logging in with your credentials and selecting the repository and branch to trigger the pipeline.

aws

Services

Search

[Alt+S]

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Choose pipeline settings [info](#)

Step 1 of 5

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Pipeline type

You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.

☐ Superseded
A more recent execution can overtake an older one. This is the default.

☒ Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

☐ Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

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

CloudShell Feedback © 2024, Amazon

aws

Services

Search

[Alt+S]

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add source stage [info](#)

Step 2 of 5

Source

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

GitHub (Version 2)

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one.
[Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

 or

Connect to GitHub

Repository name
Choose a repository in your GitHub account.

You can type or paste the group path to any project that the provided credentials can access. Use the format "group/subgroup/project".

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

Output artifact format

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy

Step 3
Add build stage

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

GitHub (Version 2)

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one.
[Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

armaws:codeconnections:eu-central-1:654654488534:connection/cc036e2 or [Connect to GitHub](#)

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

cstuehar/ci_cd_repo

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

main

Output artifact format
Choose the output artifact format.

☒ **CodePipeline default**
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

☐ **Full clone**
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

CloudShell Feedback

2. Skip the Build Stage (if not needed)

- If your project does not require a build step (e.g., if it's just an HTML page), skip this stage.

Developer Tools > **CodePipeline** > **Pipelines** > Create new pipeline

Step 1
[Choose pipeline settings](#)

Step 2
[Add source stage](#)

Step 3
Add build stage

Step 4
[Add deploy stage](#)

Step 5
[Review](#)

Add build stage [Info](#)
Step 3 of 5

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, buildspec file, and output file names.

[Cancel](#) [Previous](#) [Skip build stage](#) [Next](#)

CloudShell Feedback

3. Deploy Using CodeDeploy

- **Region:** Select your AWS region (e.g., US East).
- **Application Name:** Choose the application created in CodeDeploy.
- **Deployment Group:** Select the deployment group created earlier.

The screenshot shows the AWS CodePipeline console interface. On the left, a sidebar lists the steps: Step 3 (Add build stage), Step 4 (Add deployment stage), and Step 5 (Review). The main content area is titled 'Deploy' and contains the following configuration fields:

- Deploy provider:** A dropdown menu set to 'AWS CodeDeploy'.
- Region:** A dropdown menu set to 'US East (Ohio)'.
- Input artifacts:** A dropdown menu set to 'SourceArtifact'.
- Application name:** A text input field containing 'CI_CD_CodeDeploy'.
- Deployment group:** A text input field containing 'CI_CD_CodeDeployGroup'.
- Configure automatic rollback on stage failure:** An unchecked checkbox.

At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

4. Verify Pipeline Creation

- Once the pipeline is set up, it will attempt to deploy the code. However, it will fail initially if no EC2 instances are available. This is expected behavior.

aws

Services

Search

[Alt+S]

Trigger type

No filter

Step 3: Add build stage

Build action provider

Build stage

No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS CodeDeploy

ApplicationName

CI_CD_CodeDeploy

DeploymentGroupName

CI_CD_CodeDeployGroup

Configure automatic rollback on stage failure

Disabled

Cancel

Previous

Create pipeline

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates

aws

Services

Search

[Alt+S]

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates

Privacy

Terms

Cookie preferences

Developer Tools

CodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Go to resource

Feedback

Success

Create a notification rule for this pipeline

Congratulations! The pipeline CI_CD_CodePipeline has been created.

Developer Tools > CodePipeline > Pipelines > CI_CD_CodePipeline

CI_CD_CodePipeline

Pipeline type: V2 Execution mode: QUEUED

Source

in progress

Pipeline execution ID: 2fbee64-5a27-4811-86d8-55564bcb8825

Source

in progress - just now

View details

Disable transition

Deploy

Didn't Run

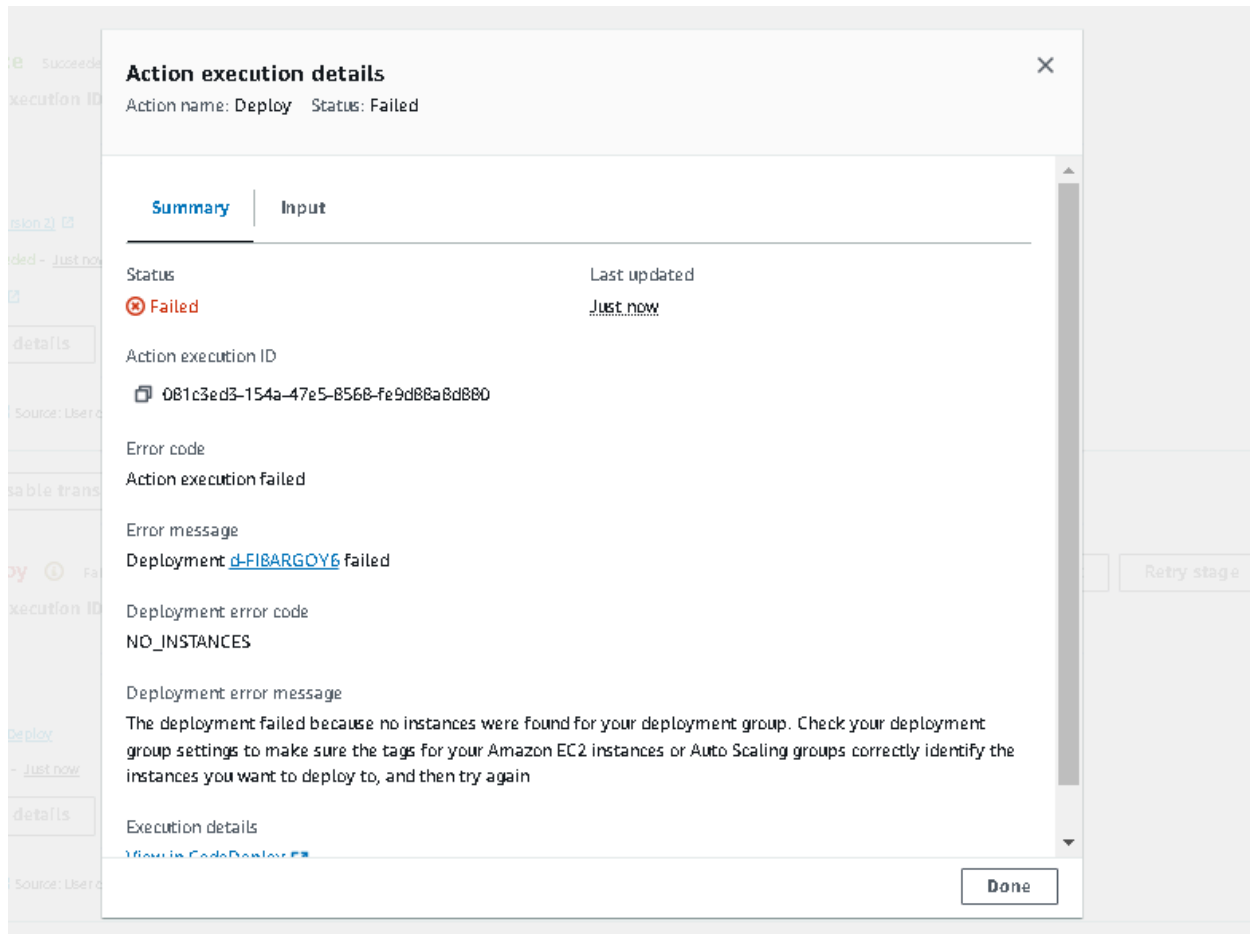
Start rollback

Deploy

AWS CodeDeploy

Didn't Run

No executions yet



Deploying the Application

- Edit Auto Scaling Group Size:**
Increase the desired capacity to 1 to initiate the creation of an EC2 instance. The instance will automatically install the CodeDeploy agent and be prepared for deployment.
- Monitor Deployment Events:**
Check CodeDeploy for event logs. The deployment process may get stuck at the "Block Traffic" event for around 5 minutes. This happens as the Load Balancer registers and deregisters the instance to verify its health status.
- Access the Deployed Application:**
Once the deployment is complete and the instance is healthy, copy the DNS from the Load Balancer dashboard and paste it into a web browser. The application should now be live.
- Test Auto Scaling:**
Increase the number of instances in the Auto Scaling group to test automatic scaling. As

load increases, more instances will be created, and the application will be replicated across these instances.

The screenshot displays the AWS Management Console interface for an EC2 Auto Scaling Group. The left sidebar shows navigation options like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area shows the 'EC2_Auto_Scaling_Group' details, including group name, date created, and launch template. A modal dialog titled 'Group size' is open, allowing configuration of the desired capacity type (Units), desired capacity (1), and scaling limits (Min: 1, Max: 1). The bottom section of the console shows the 'Instances (1)' table with one instance in a 'Pending' state, and 'Lifecycle hooks (1)' and 'Warm pool' sections.

EC2_Auto_Scaling_Group Details:

- Auto Scaling group name: EC2_Auto_Scaling_Group
- Date created: Sun Sep 08 2024 00:59:35 GMT+0530 (India Standard Time)
- Launch template: lt-d9d1dfc318982a278 (CL_CD_App)
- Version: Default
- Security groups: sg-08bcb95a6a683d8000
- Storage (root): Key pair name

Group size Modal:

- Desired capacity type: Units (number of instances)
- Desired capacity: 1
- Scaling limits: Min desired capacity: 1, Max desired capacity: 1

Instances (1):

Instance ID	Lifecycle	Instance type	Weighted ca...	Launch temp...	Availability Z...	Health status	Protected from
i-0bcb9b67f8e7cb29a	Pending	t2.micro	-	CL_CD_App Versi	us-east-2b	Healthy	

Lifecycle hooks (1):

Name	Lifecycle transition	Default result	Heartbeat timeout (sec...	Notification target ARN	Role ARN
CodeDeploy-managed-aut...	autoscaling:EC2_INSTANC...	ABANDON	600	arnaws.sqs:us-east-2.626...	

Warm pool: No warm pool currently configured. Decrease scale-out latency by pre-initializing EC2 instances and save money by reducing the number of continuously running instances.

- **Terminate Instances:** After testing, make sure to either terminate the instances manually or reduce the Auto Scaling group size to 0. This will prevent unnecessary charges.

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Search

[Alt+S]

Ohio

Tushar Gupta

EC2 > Auto Scaling groups > EC2_Auto_Scaling_Group

EC2_Auto_Scaling_Group

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Instances (2)

Filter instances

< 1 >

Instance ID	Lifecycle	Instance type	Weighted ca...	Launch temp...	Availability Z...	Health status	Protected from
i-03c0390e68068d468	Terminating	t2.micro	-	CI_CD_App Versi	us-east-2c	Healthy	
i-0f36fa57a5b5e12cc	Terminating	t2.micro	-	CI_CD_App Versi	us-east-2a	Healthy	

Lifecycle hooks (1)

Filter lifecycle hooks

< 1 >

Name	Lifecycle transition	Default result	Heartbeat timeout (sec...	Notification target ARN	Role ARN
CodeDeploy-managed-aut...	autoscaling:EC2_INSTANC...	ABANDON	600	arn:aws:sqs:us-east-2:626...	

Warm pool

No warm pool currently configured.

Search

[Alt+S]

Ohio

Tushar Gupta

EC2 > Auto Scaling groups > EC2_Auto_Scaling_Group

EC2_Auto_Scaling_Group

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details

Auto Scaling group name

EC2_Auto_Scaling_Group

Desired capacity

0

Desired capacity type

Units (number of instances)

Amazon Resource Name (ARN)

arn:aws:autoscaling:us-east-2:654654488534:autoScalingGroup:37a419d5-0aee-4103-836b-cfe78b10e6d:autoScalingGroupName/EC2_Auto_Scalin

Date created

Sun Sep 08 2024 00:59:35 GMT+0530 (India Standard Time)

Minimum capacity

0

Status

Updating capacity

Maximum capacity

0

Launch template

Launch template

lt-0bd1dfc318982c278

AMI ID

ami-003932de22c285676

Instance type

t2.micro

Owner

arn:aws:iam::654654488534:root

Version

Default

Security groups

-

Security group IDs

sg-08ed986a6063ddb0b

Create time

Sun Sep 08 2024 00:51:20 GMT+0530 (India Standard Time)

Description

Storage (volumes)

Key pair name

Request Spot Instances

Conclusion:

By following the above steps, you can successfully set up a CI/CD pipeline that deploys a web application to AWS. The pipeline is designed to scale automatically with load, ensuring high availability and consistency in deployment.