CI/CD Pipeline Setup with AWS Services

By Tushar Gupta

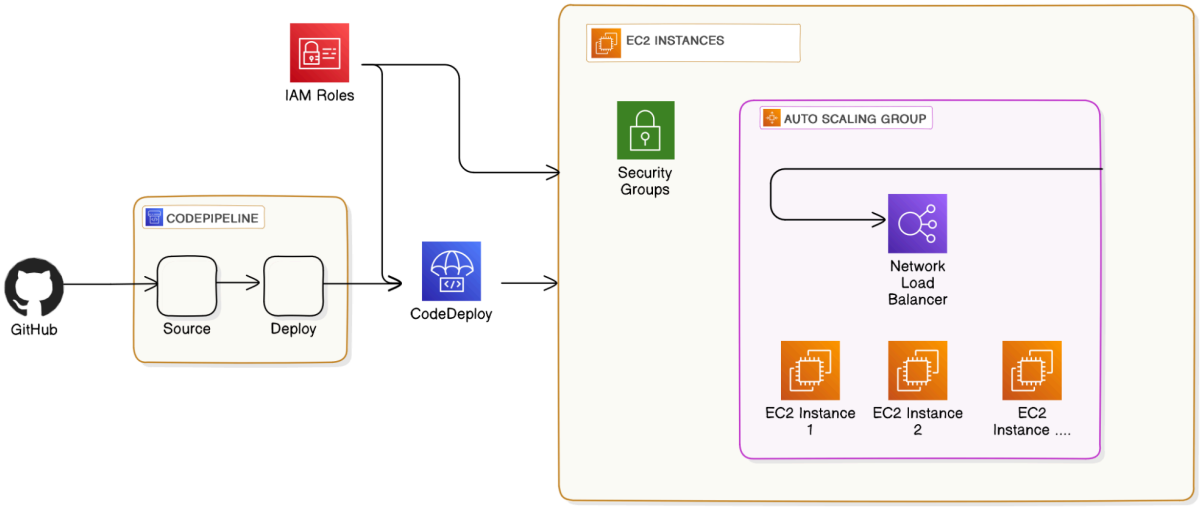
Objective:
The goal is to set up a CI/CD pipeline that automates the deployment of a simple web application to a server. The pipeline should trigger on code pushes to a repository and deploy the application in a scalable manner. When load increases, the number of servers should automatically scale up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

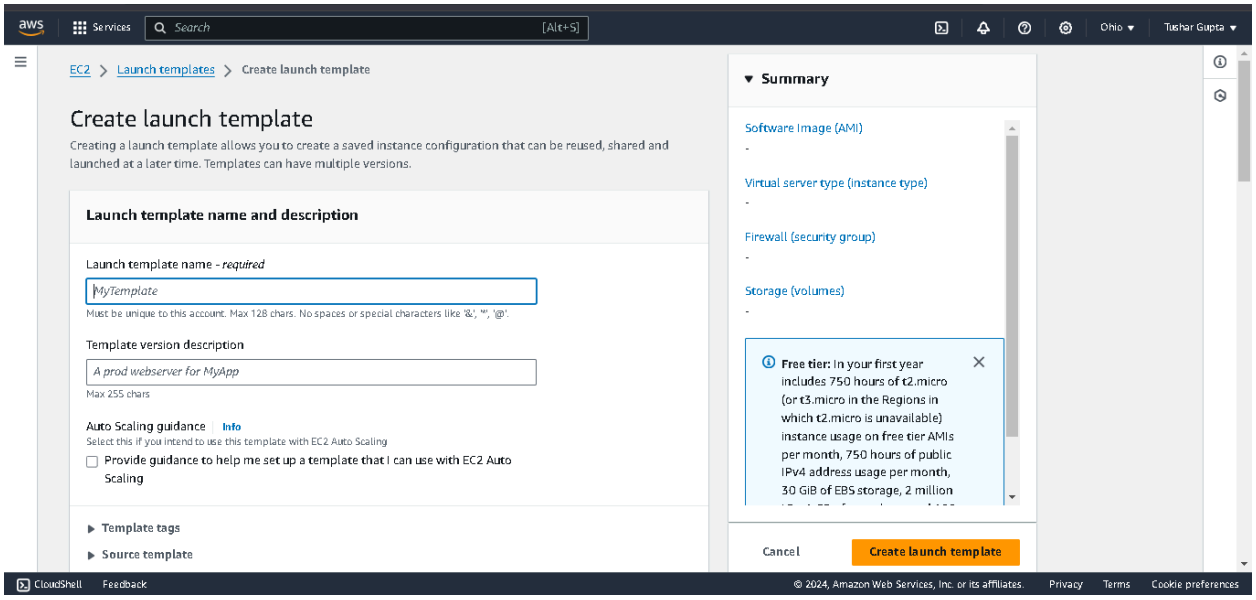
Basic CI/CD Pipeline

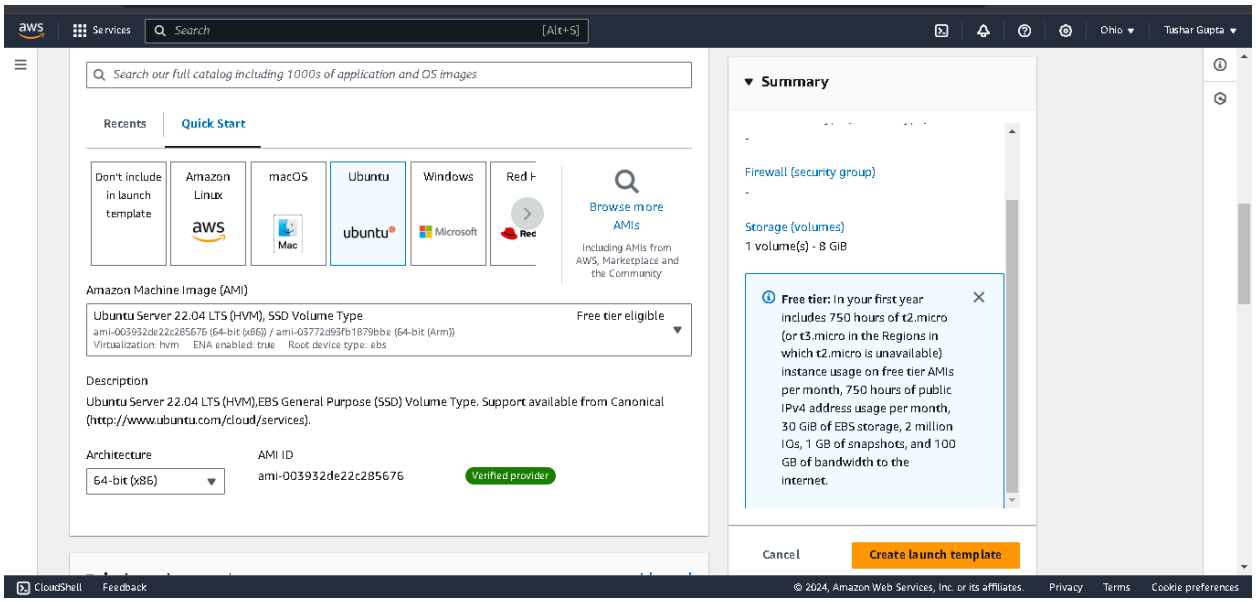


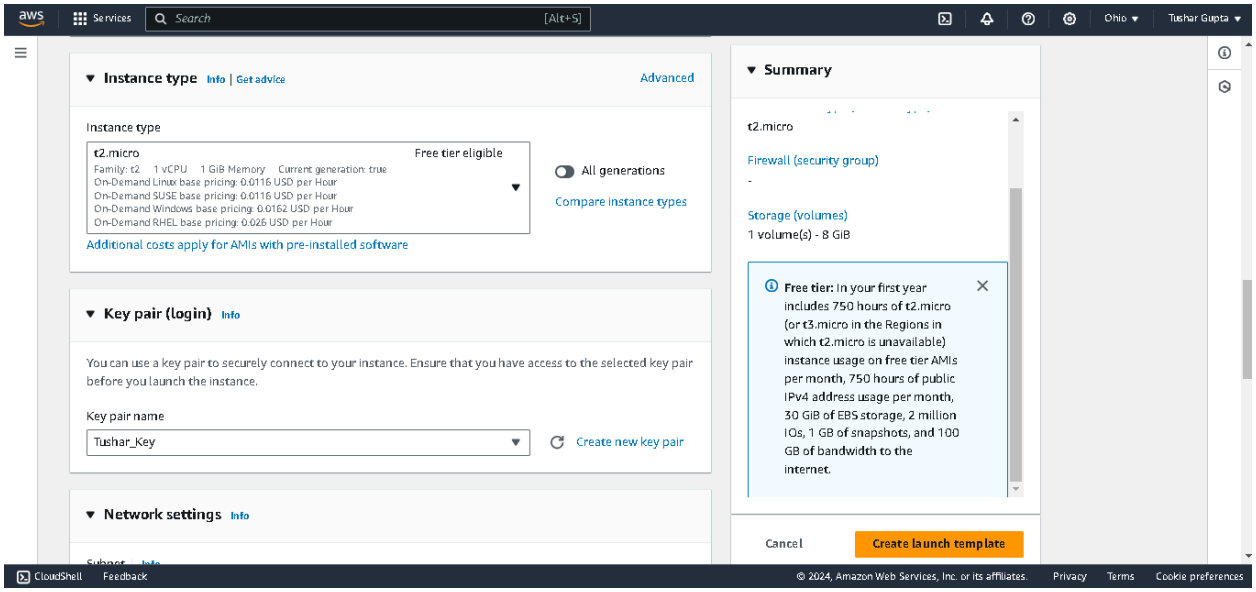
Step-by-Step Process

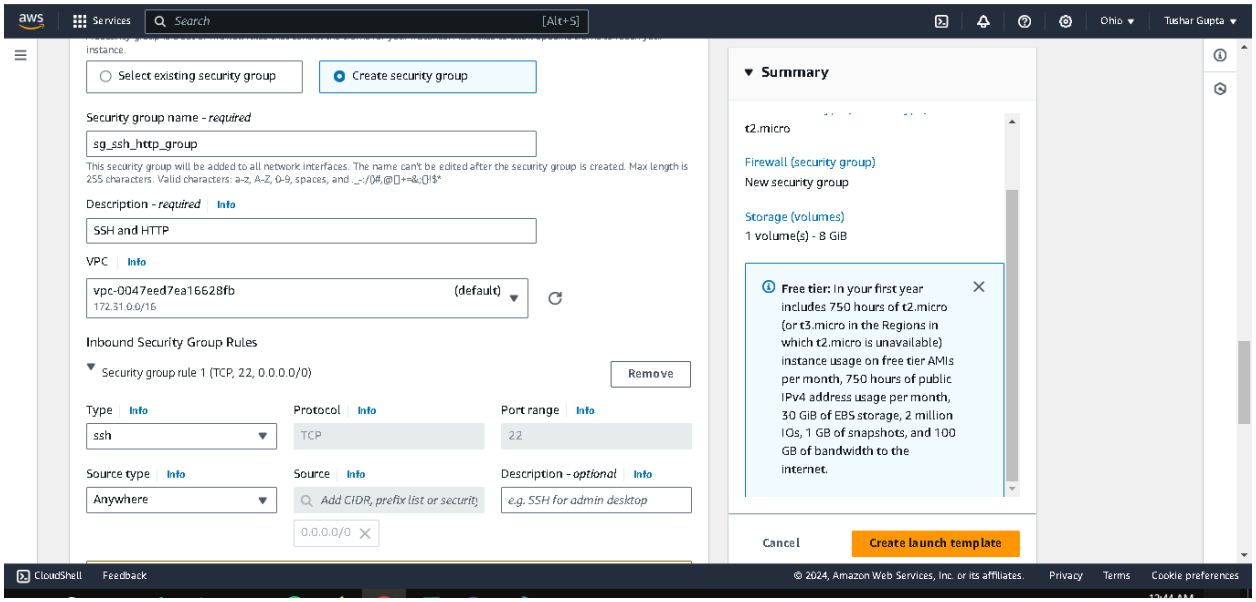
1. Create a Launch Instance Template in EC2

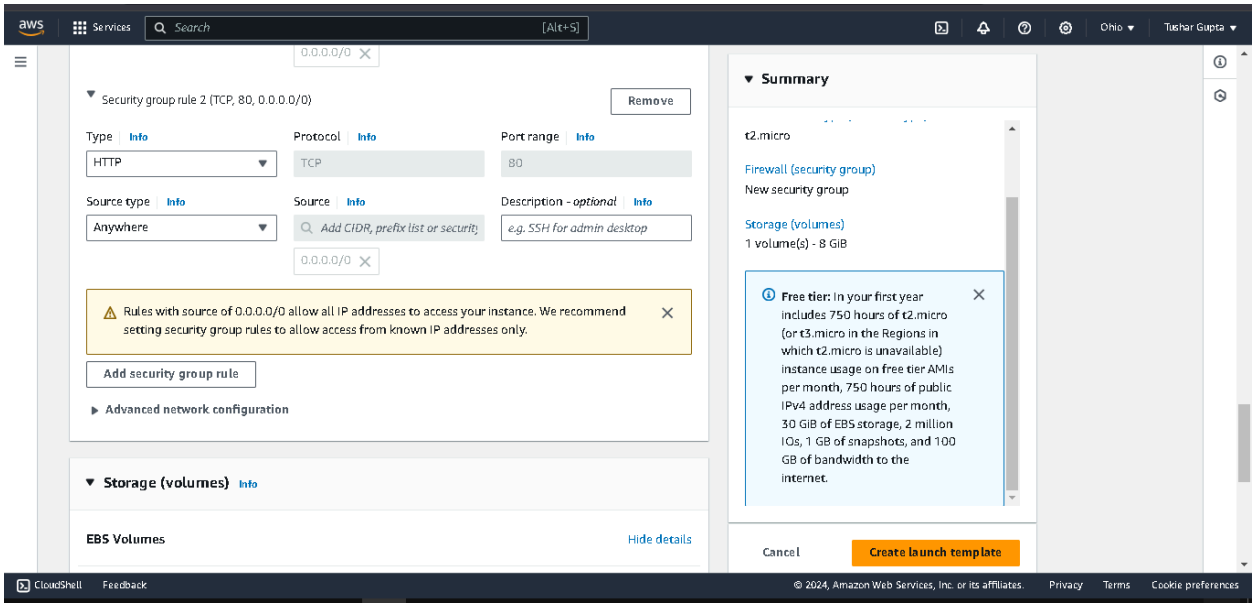
- **Operating System:** Ubuntu (Free tier)
- **Instance Type:** T2 Micro
- **Key Pair:** Enabled
- **Security Group:**
    - Allow SSH and HTTP access from anywhere.





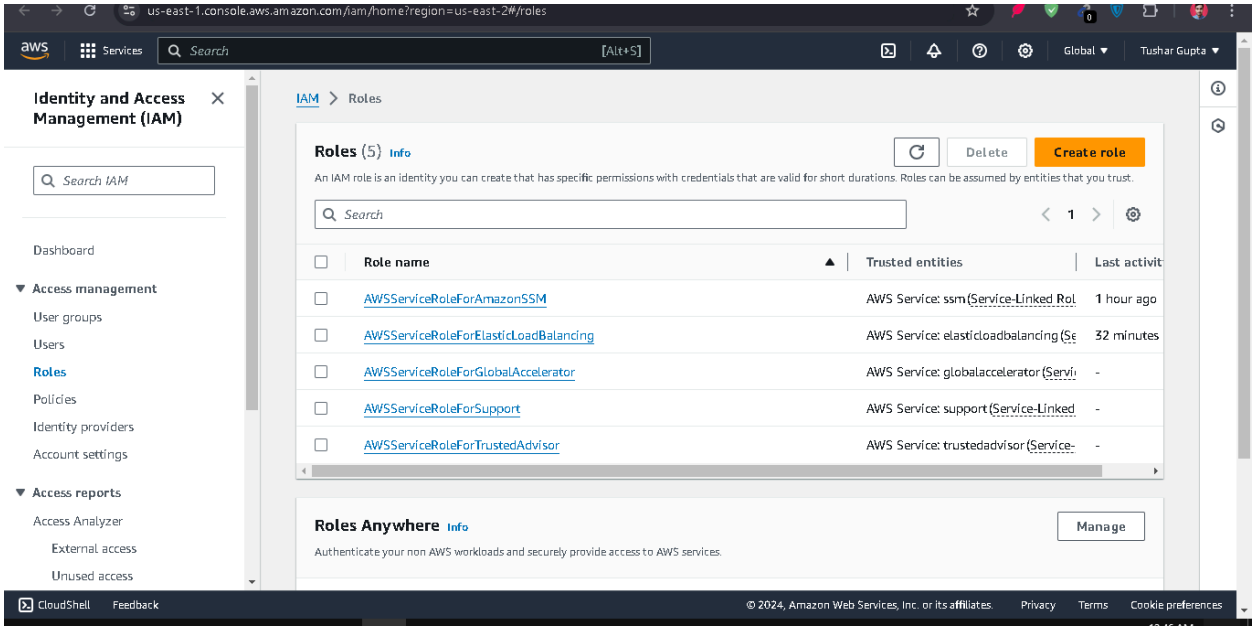

aws
Services
Search
[Alt+S]
Ohio
Tushar Gupta
Instance type Info | Get advice
Advanced
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
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 IOs, 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

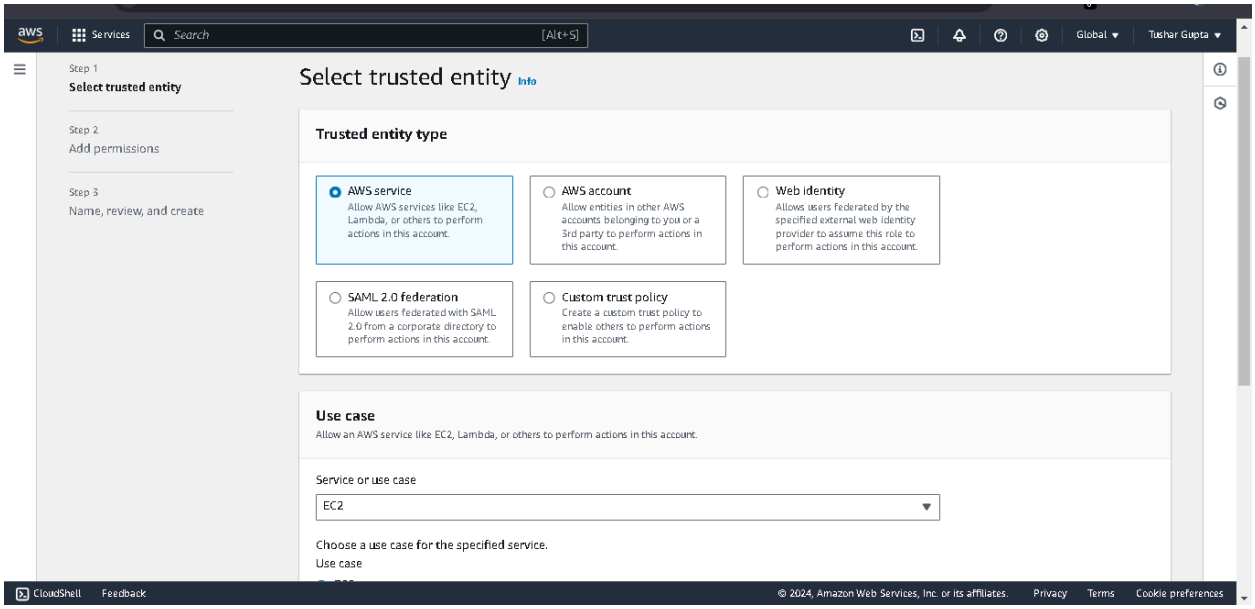


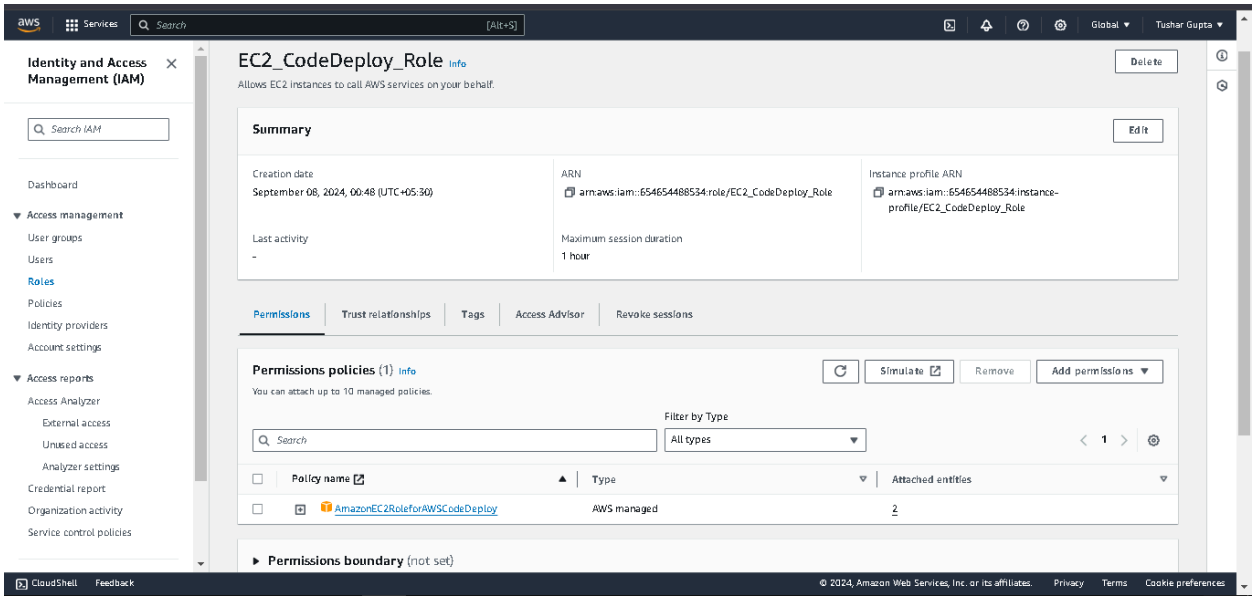


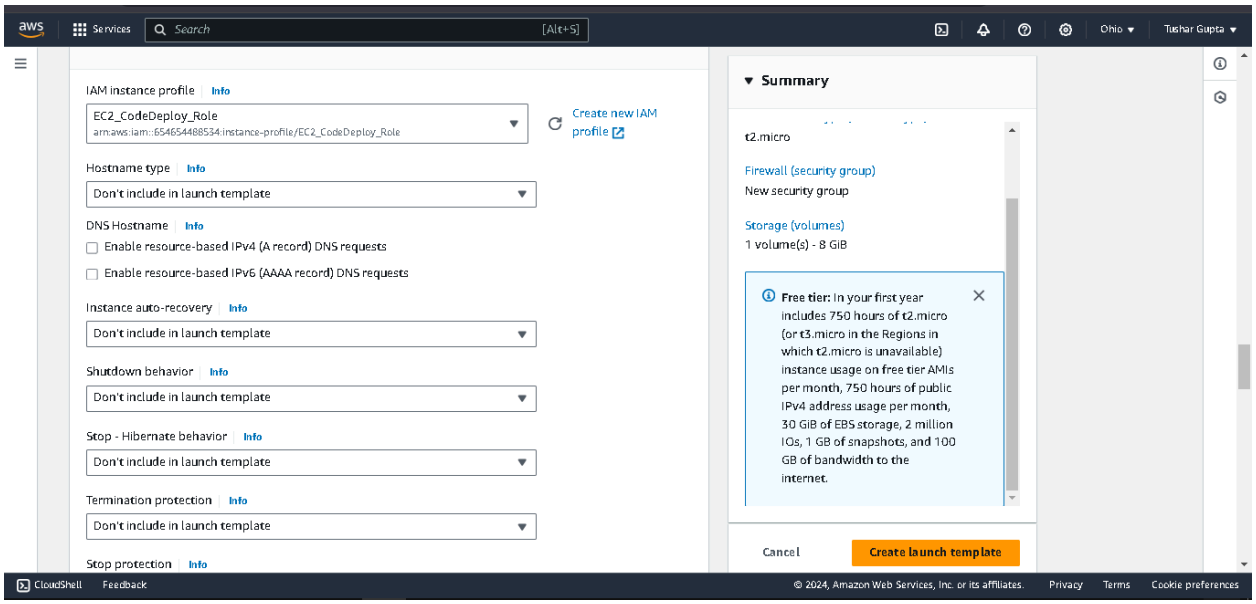
2. Create an IAM Role for EC2

- **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
aws
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
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.
Delete
Create role
Search
1
Role name
Trusted entities
Last activit
AWSServiceRoleForAmazonSSM
AWS Service: ssm (Service-Linked Rol
1 hour ago
AWSServiceRoleForElasticLoadBalancing
AWS Service: elasticloadbalancing (Se
32 minutes
AWSServiceRoleForGlobalAccelerator
AWS Service: globalaccelerator (Servi
-
AWSServiceRoleForSupport
AWS Service: support (Service-Linked
-
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service-
-
Roles Anywhere Info
Authenticate your non AWS workloads and securely provide access to AWS services.
Manage
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
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



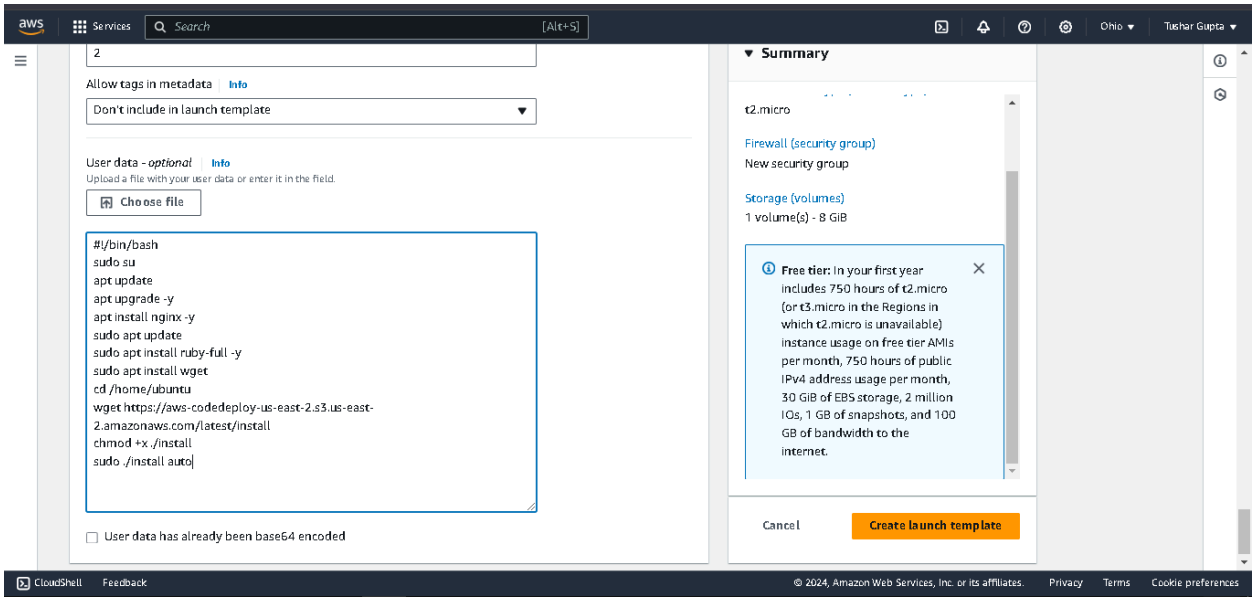


3. User Data for EC2 Instance

- 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.

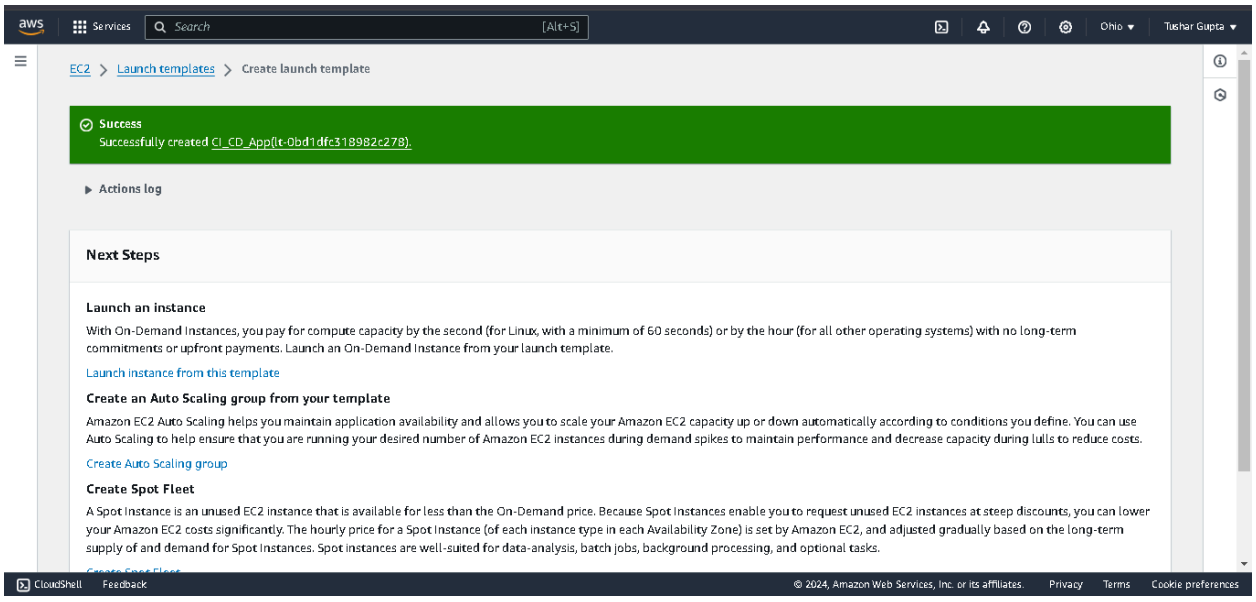
```
#!/bin/bash
sudo su
apt update
apt upgrade -y
apt install nginx -y
sudo apt update
sudo apt install ruby-full -y
```

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



4. Create the Launch Template

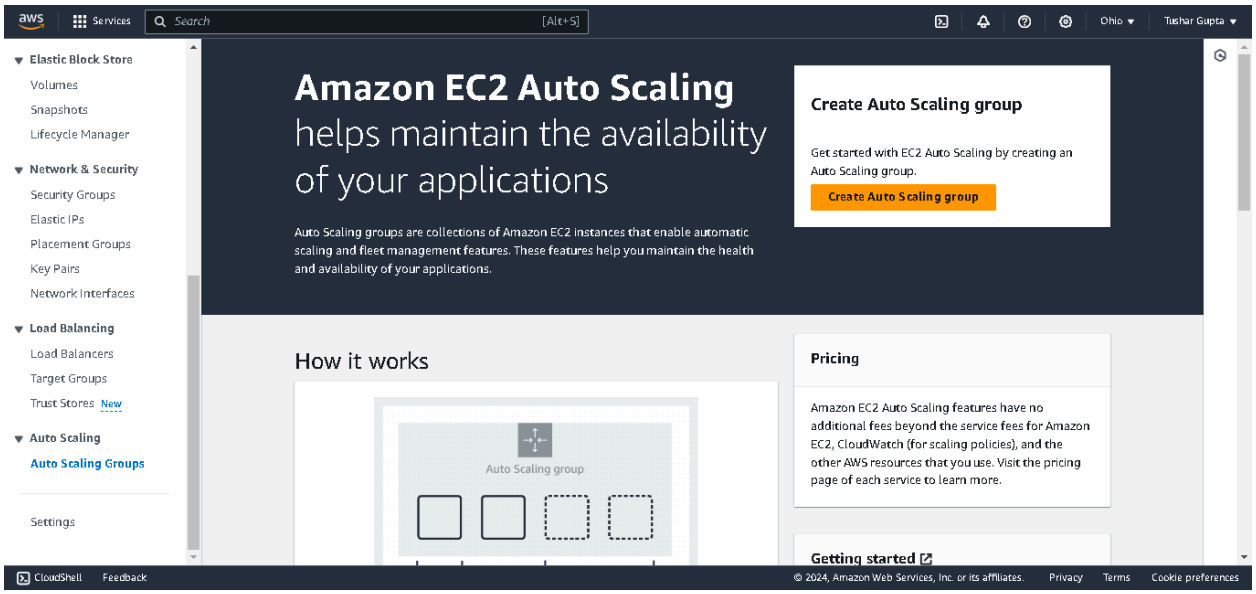
- After configuring the above settings, create the EC2 launch template.

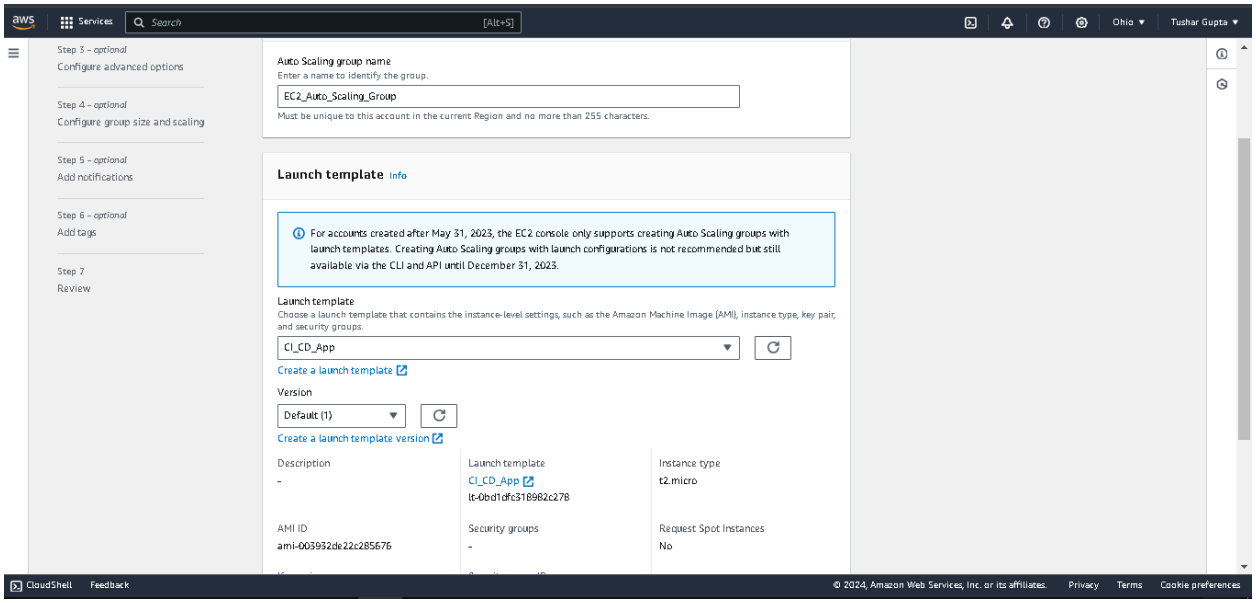


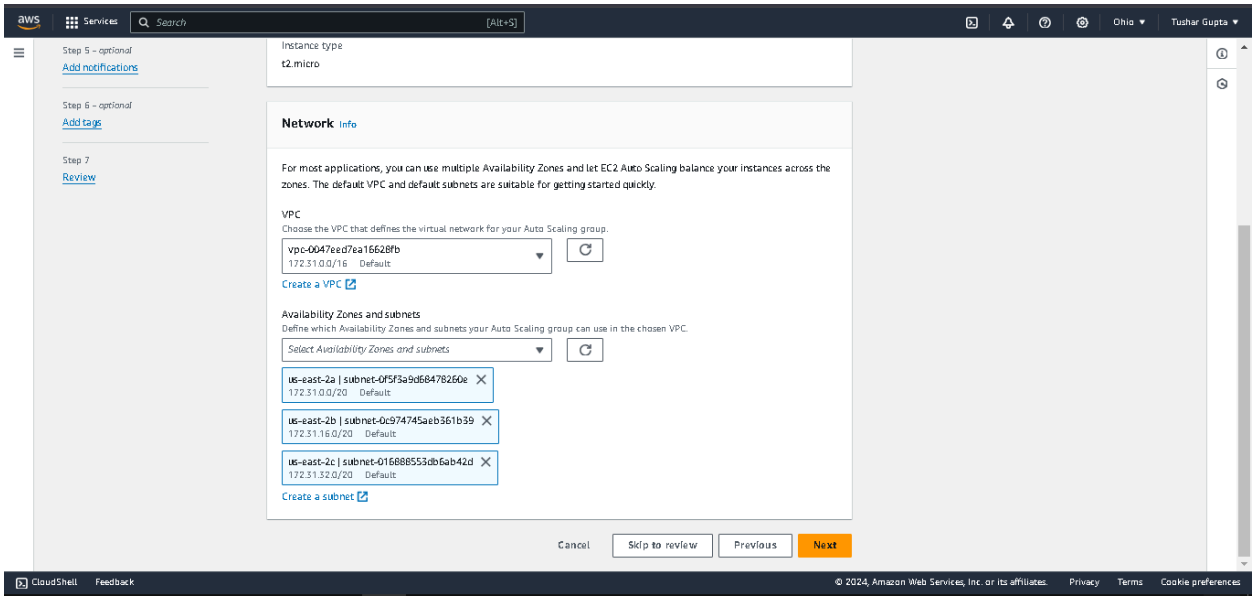
Auto Scaling Configuration

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.

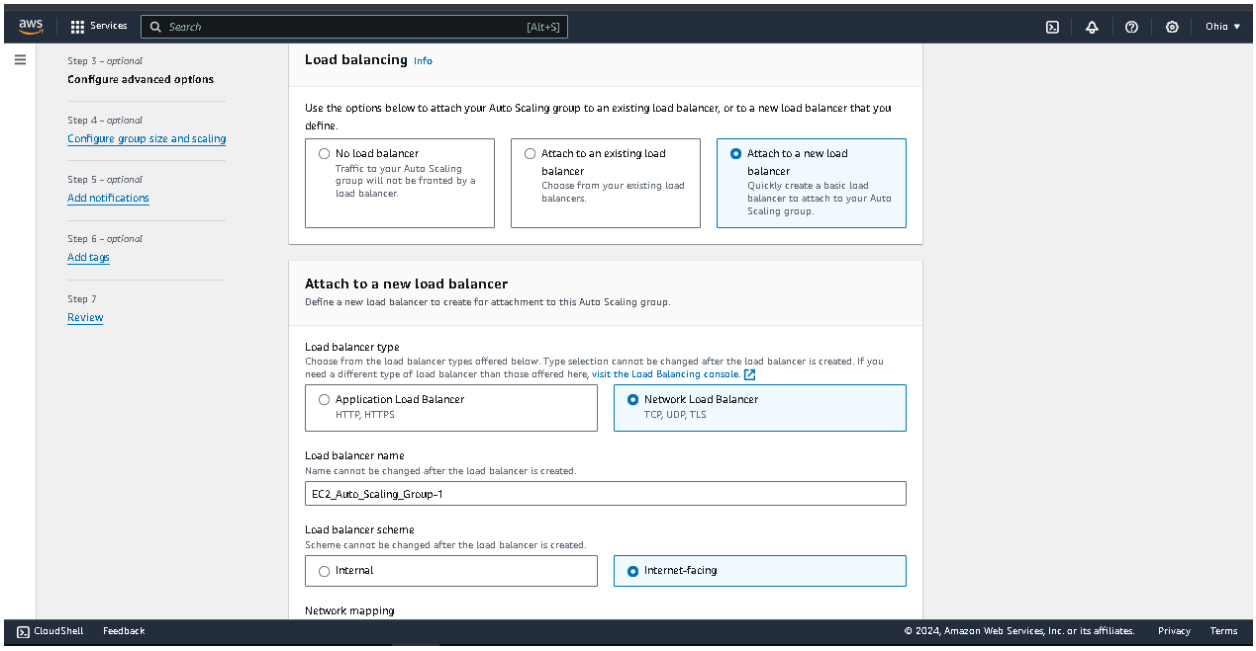

Services
Search
[Alt+S]
Ohio
Tushar Gupta
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores New
Auto Scaling
Auto Scaling Groups
Settings
Amazon 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
Auto Scaling group
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

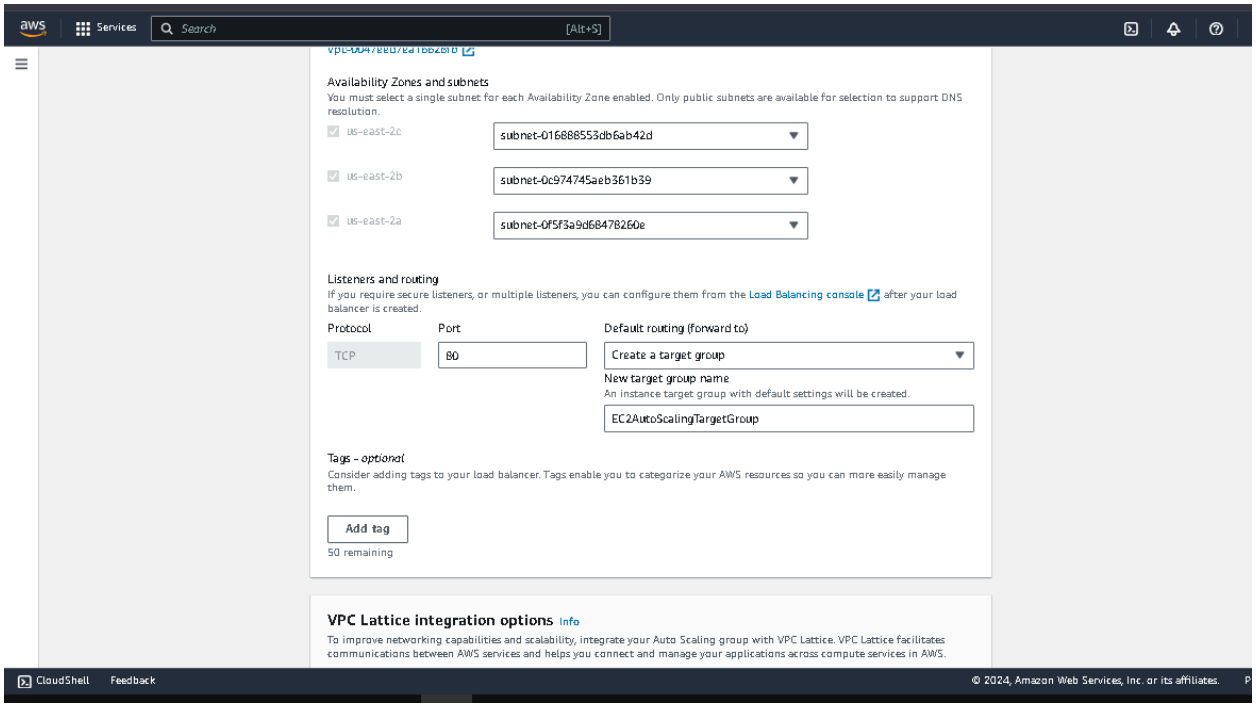

Services
Search
[Alt+S]
Ohio
Tushar Gupta
Step 3 – optional
Configure advanced options
Step 4 – optional
Configure group size and scaling
Step 5 – optional
Add notifications
Step 6 – optional
Add tags
Step 7
Review
Auto Scaling group name
Enter a name to identify the group.
EC2_Auto_Scaling_Group
Must 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.
CI_CD_App
Create a launch template
Version
Default (1)
Create a launch template version
Description
-
Launch template
CI_CD_App
lt-0bd1dfc318982c278
Instance type
t2.micro
AMI ID
ami-003932de22c285676
Security groups
-
Request Spot Instances
No
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

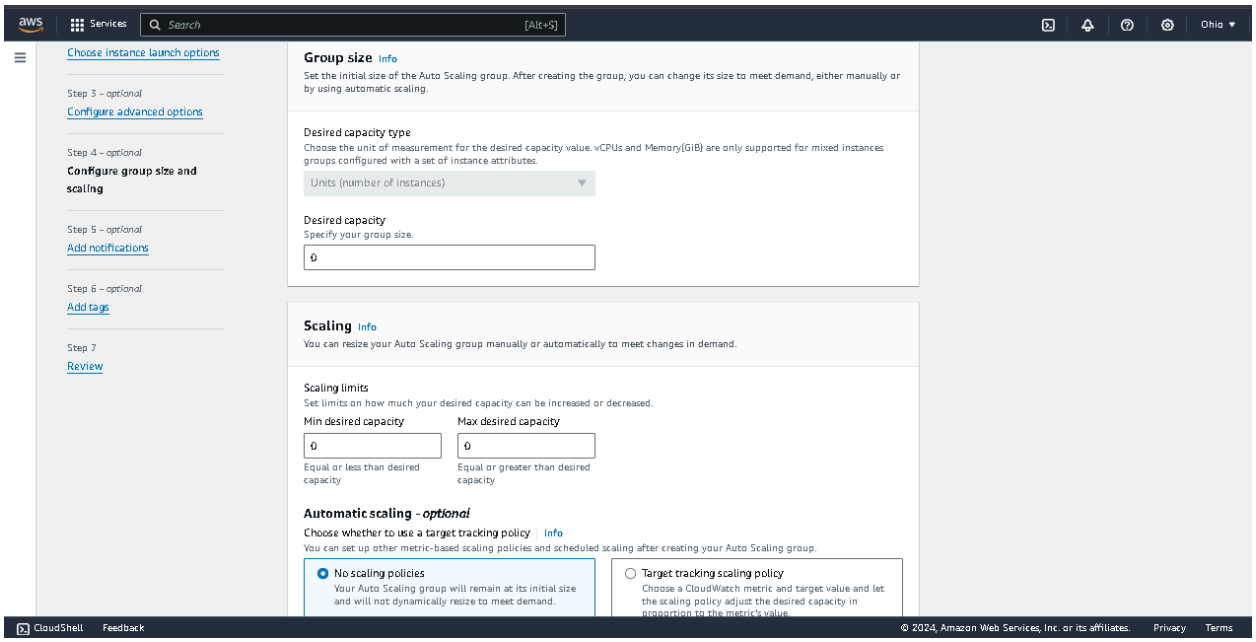


2. Load Balancer Setup

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


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

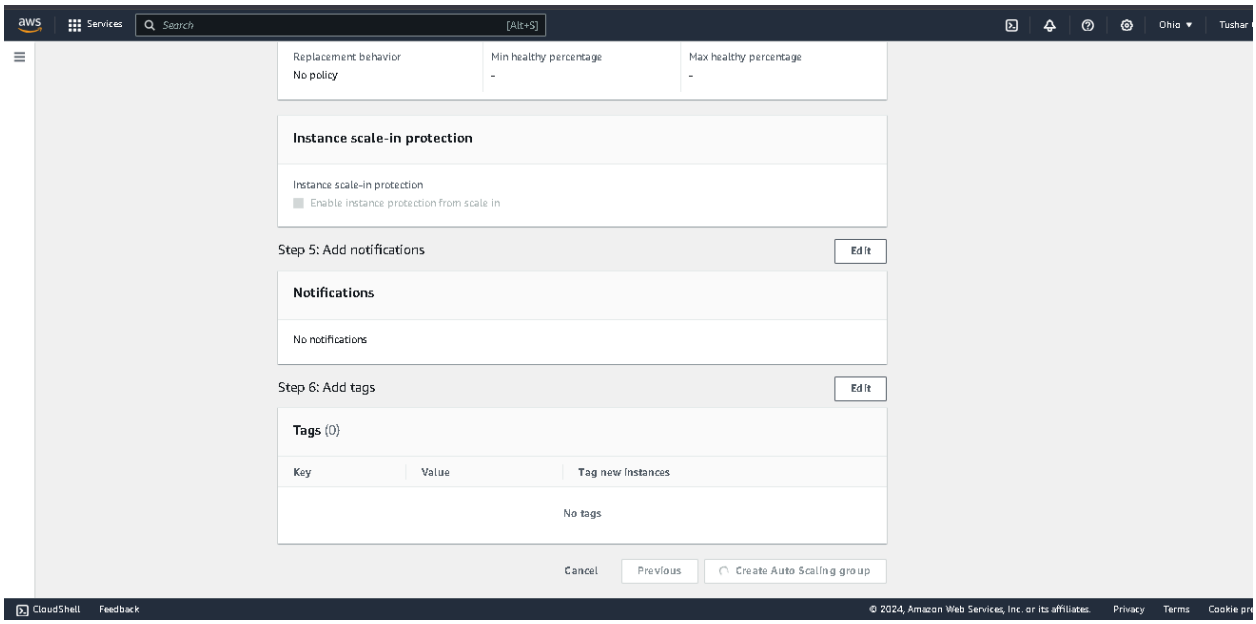

Services
Search
[Alt+S]
Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.
us-east-2c
subnet-016888553db6ab42d
us-east-2b
subnet-0c974745aeb361b39
us-east-2a
subnet-0f5f3a9d68478260e
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.

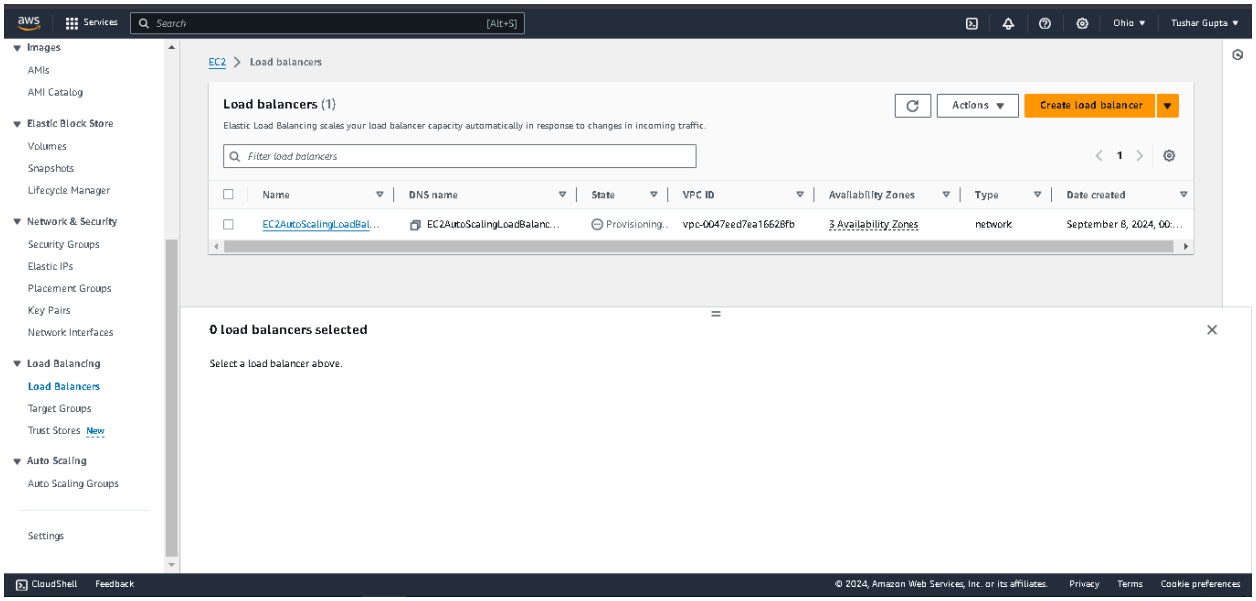


Note: Initially set the desired capacity to 0, as you don't need to start instances yet.

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.

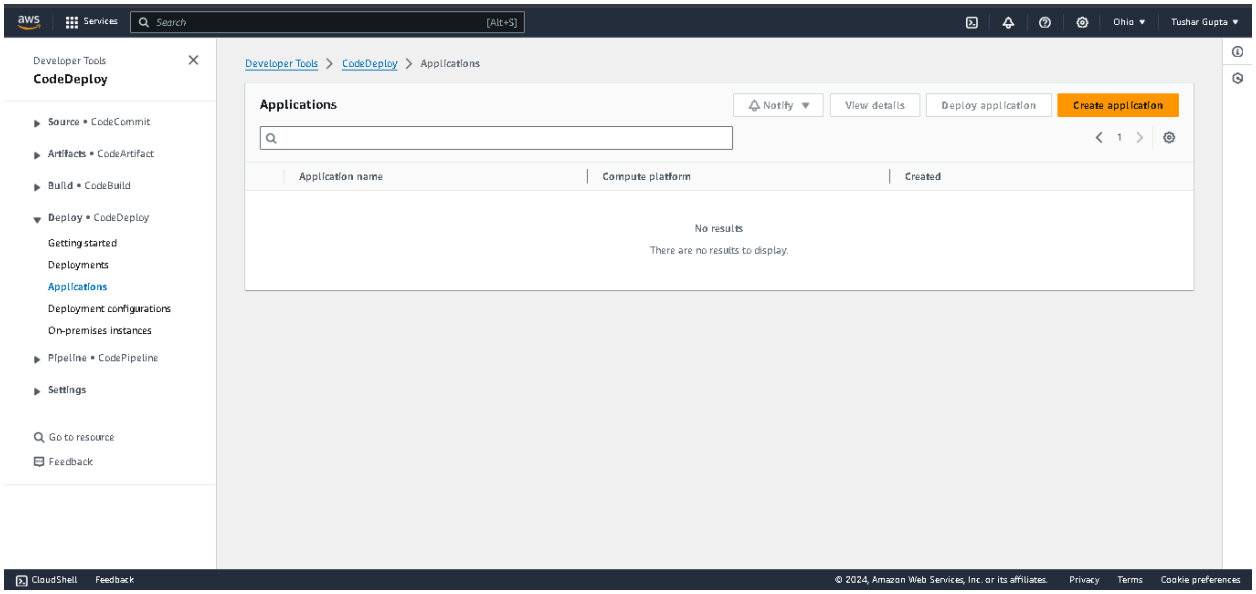


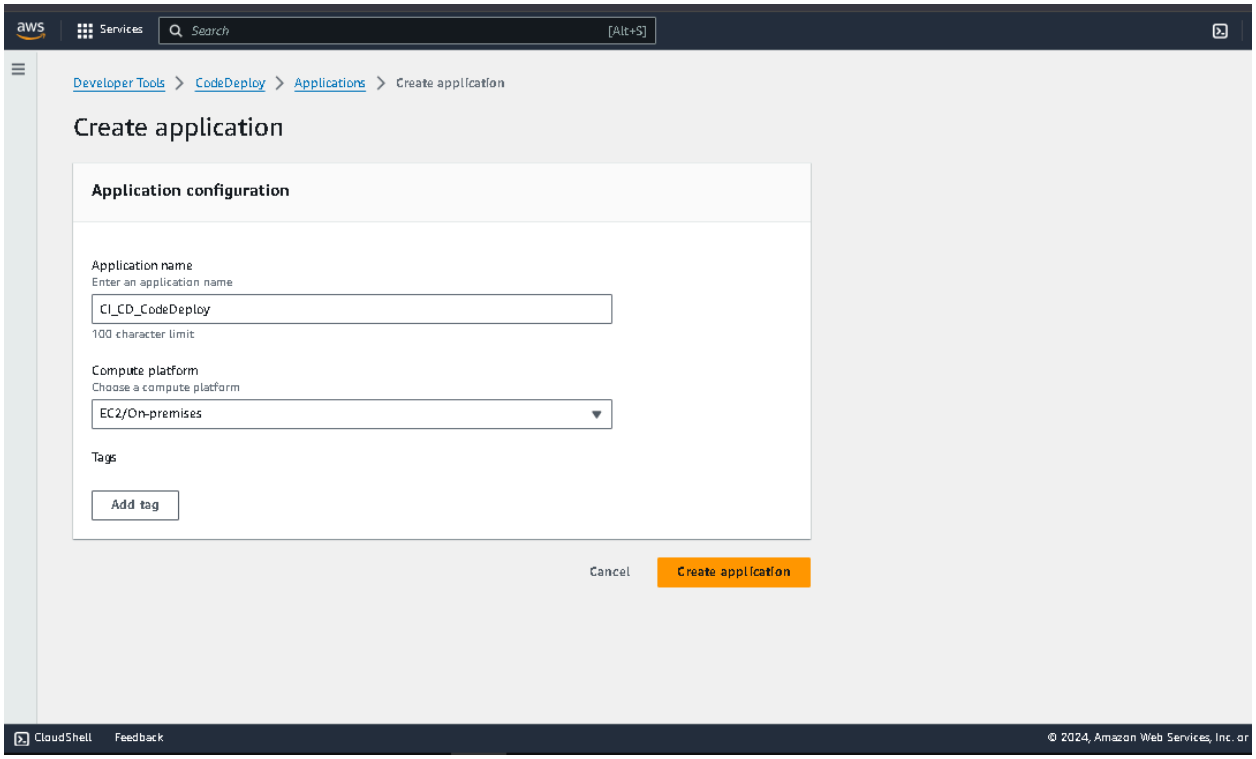


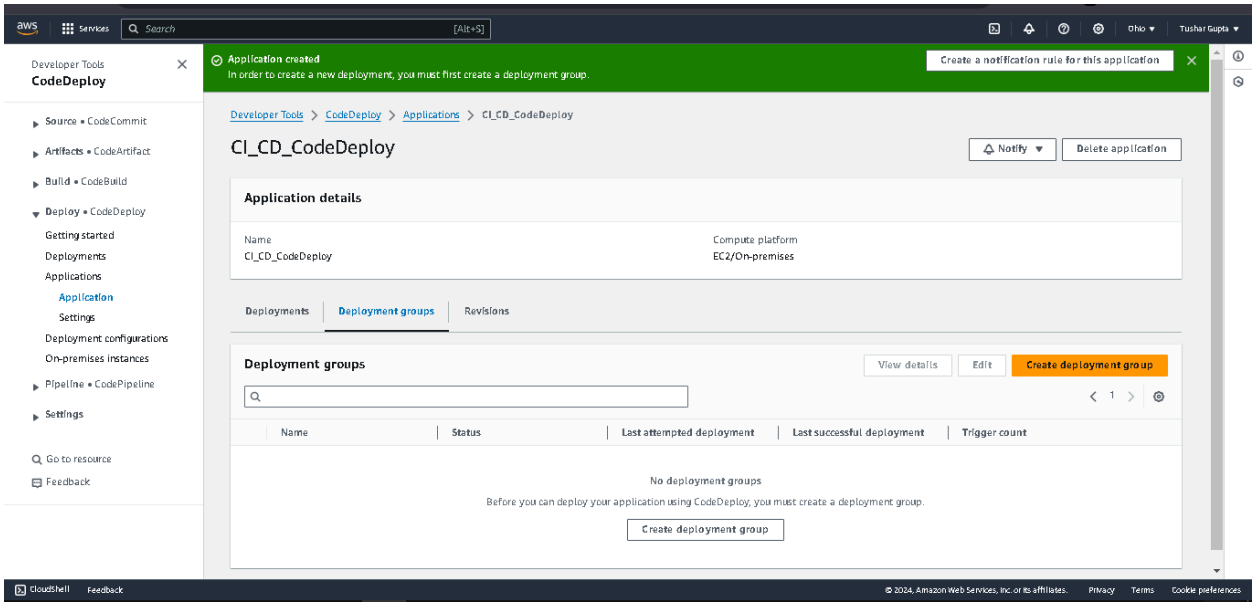
CodeDeploy Setup

1. Create a CodeDeploy Application

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

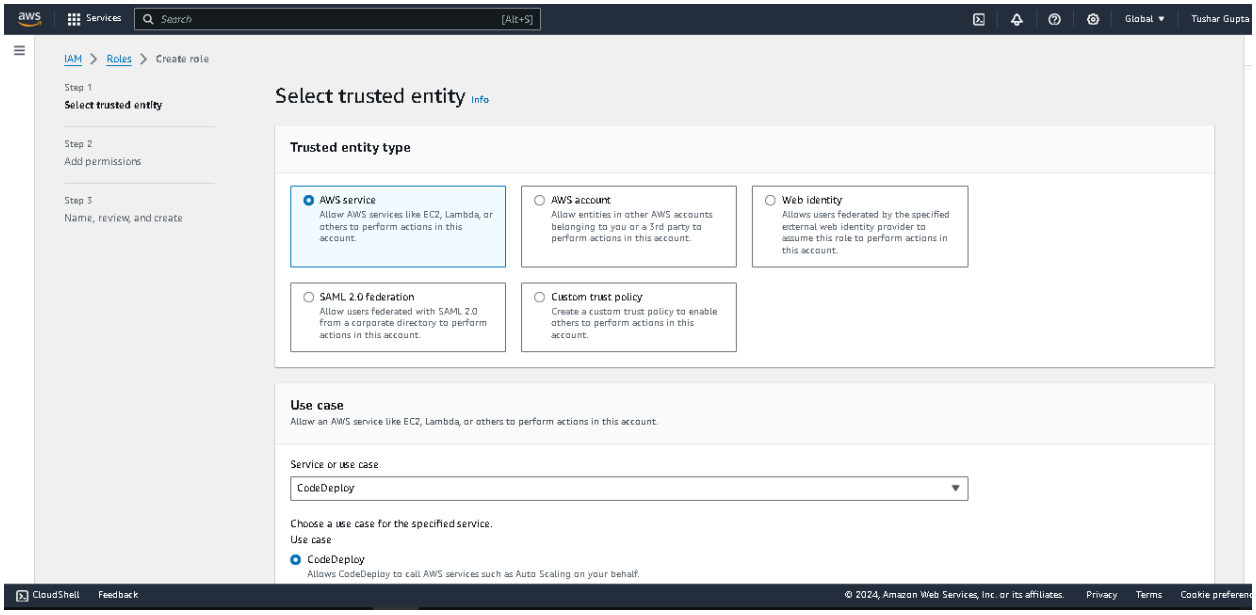

aws
Services
Search
[Alt+S]
Ohio
Tushar Gupta
Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Applications
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Settings
Go to resource
Feedback
Developer Tools > CodeDeploy > Applications
Applications
Notify
View details
Deploy application
Create application
Application name
Compute platform
Created
No results
There are no results to display.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

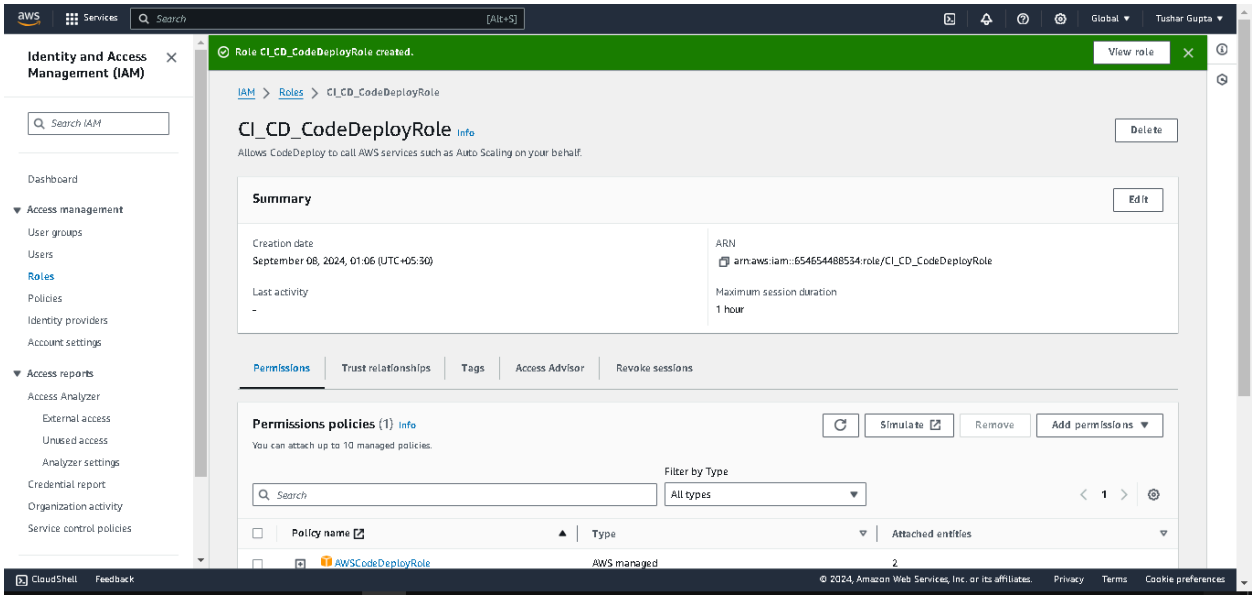

aws
Services
Search
[Alt+S]
Developer Tools > CodeDeploy > Applications > Create application
Create application
Application configuration
Application name
Enter an application name
CI_CD_CodeDeploy
100 character limit
Compute platform
Choose a compute platform
EC2/On-premises
Tags
Add tag
Cancel
Create application
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or



2. Create an IAM Role for CodeDeploy

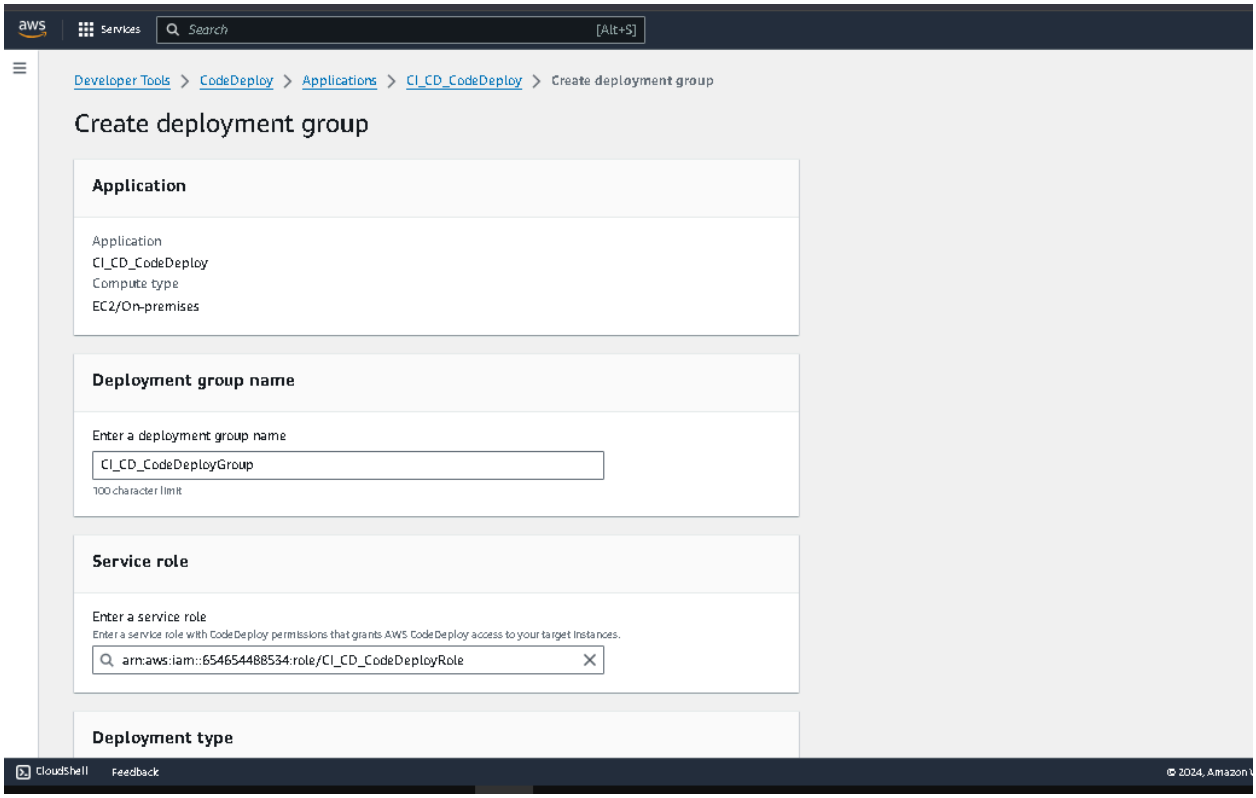
- **Service:** CodeDeploy
- **Permissions:** AWSCodeDeployRole

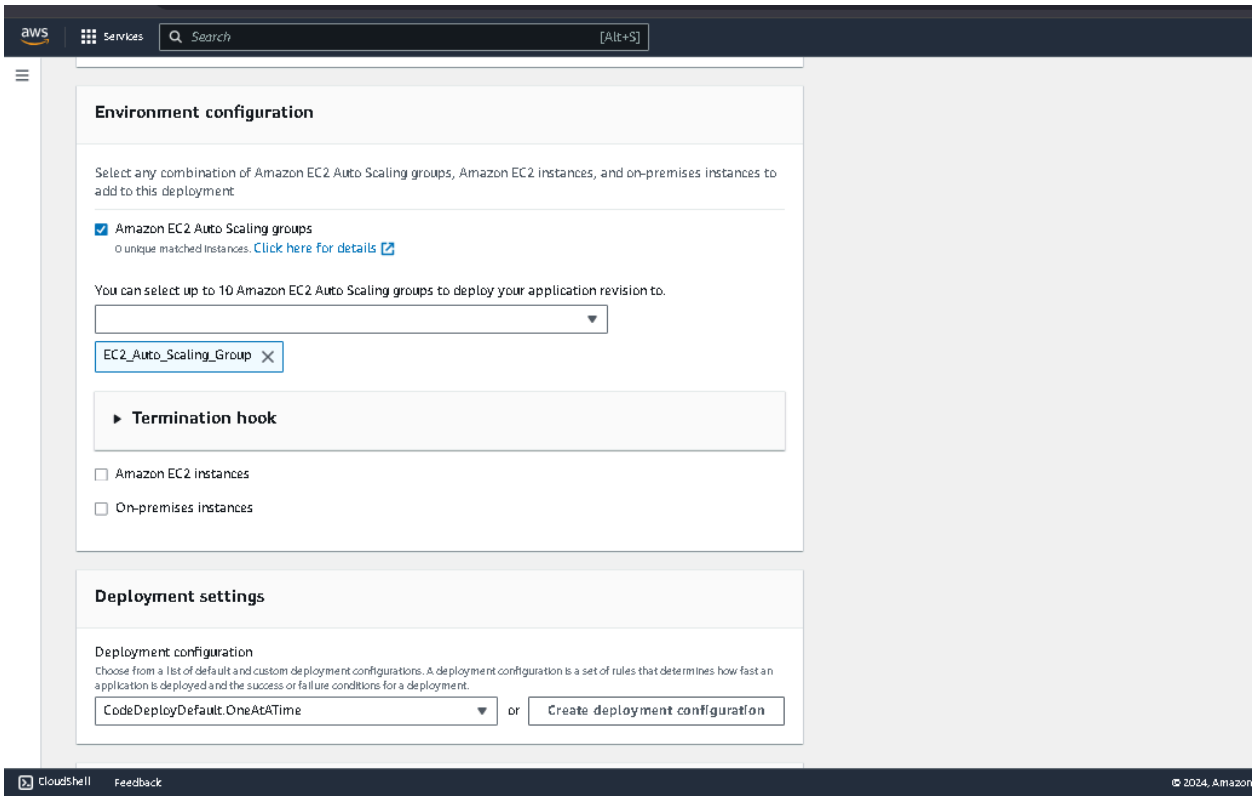


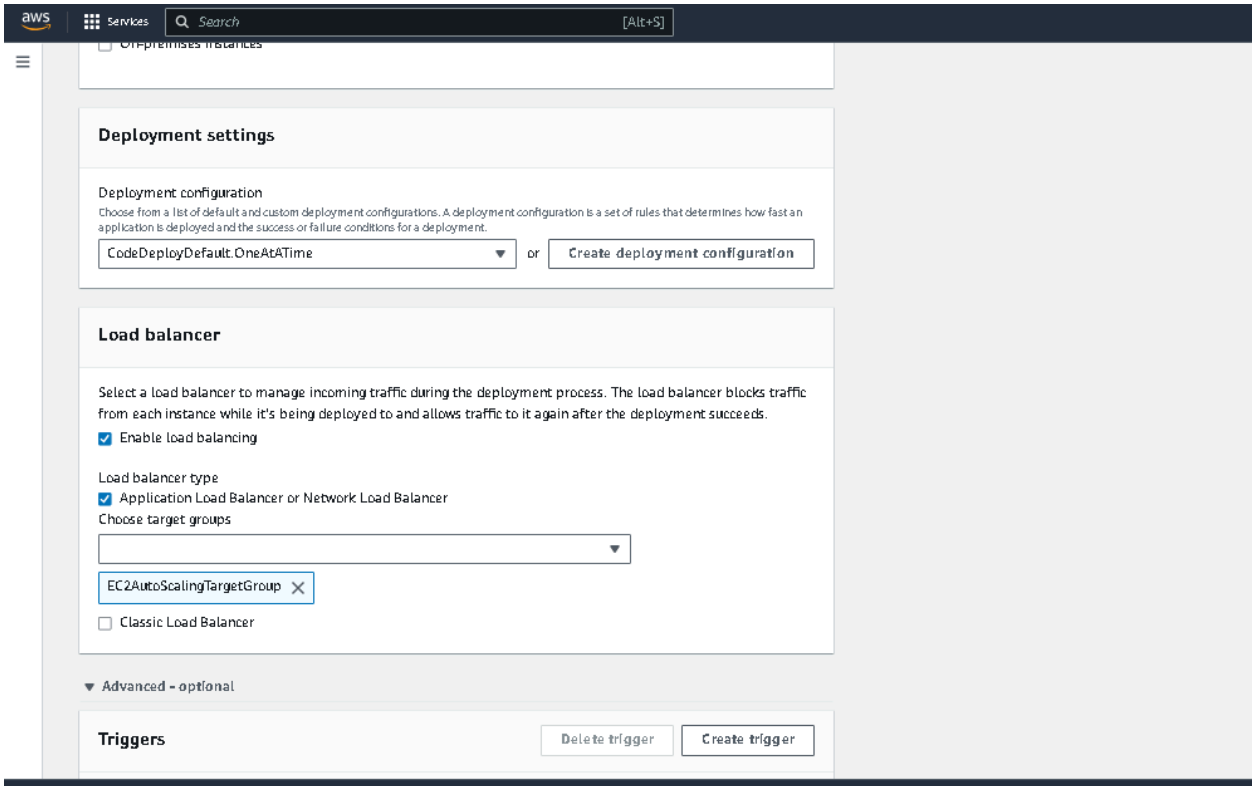


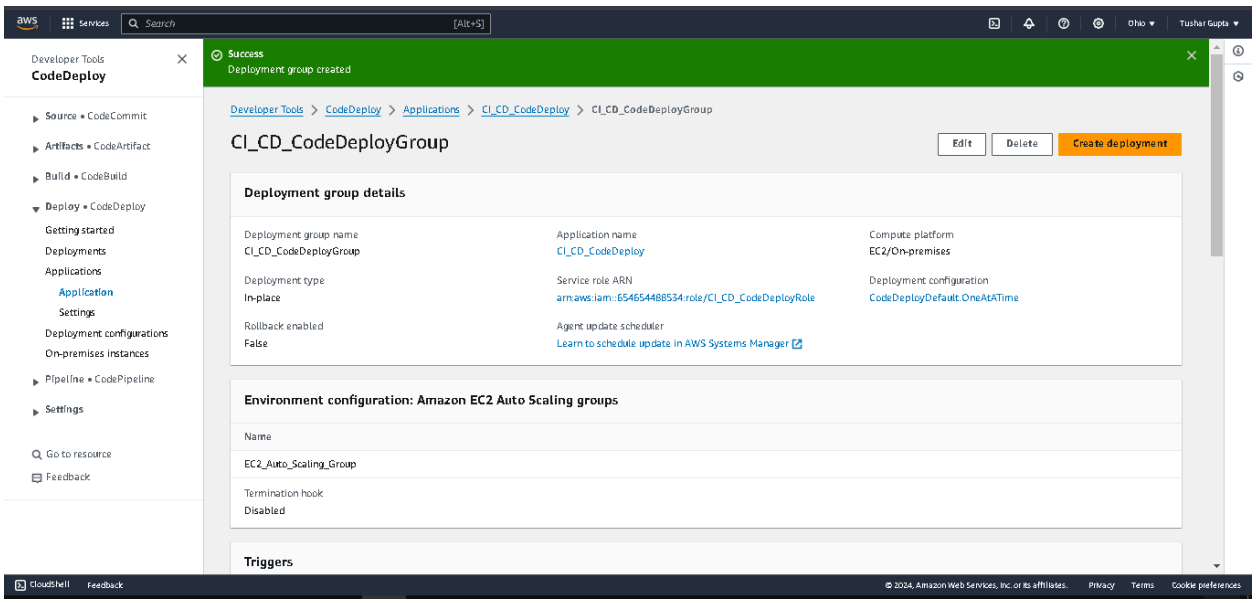
3. Create a Deployment Group

- **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


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



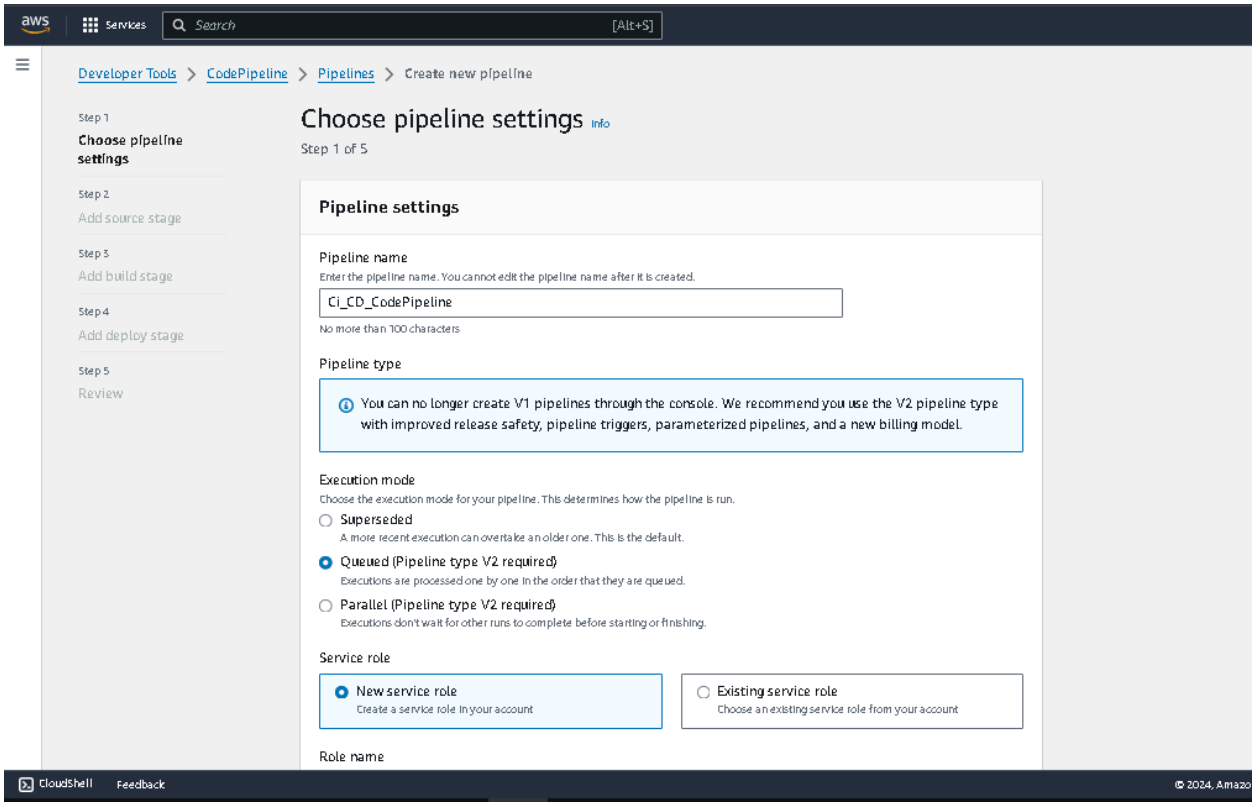


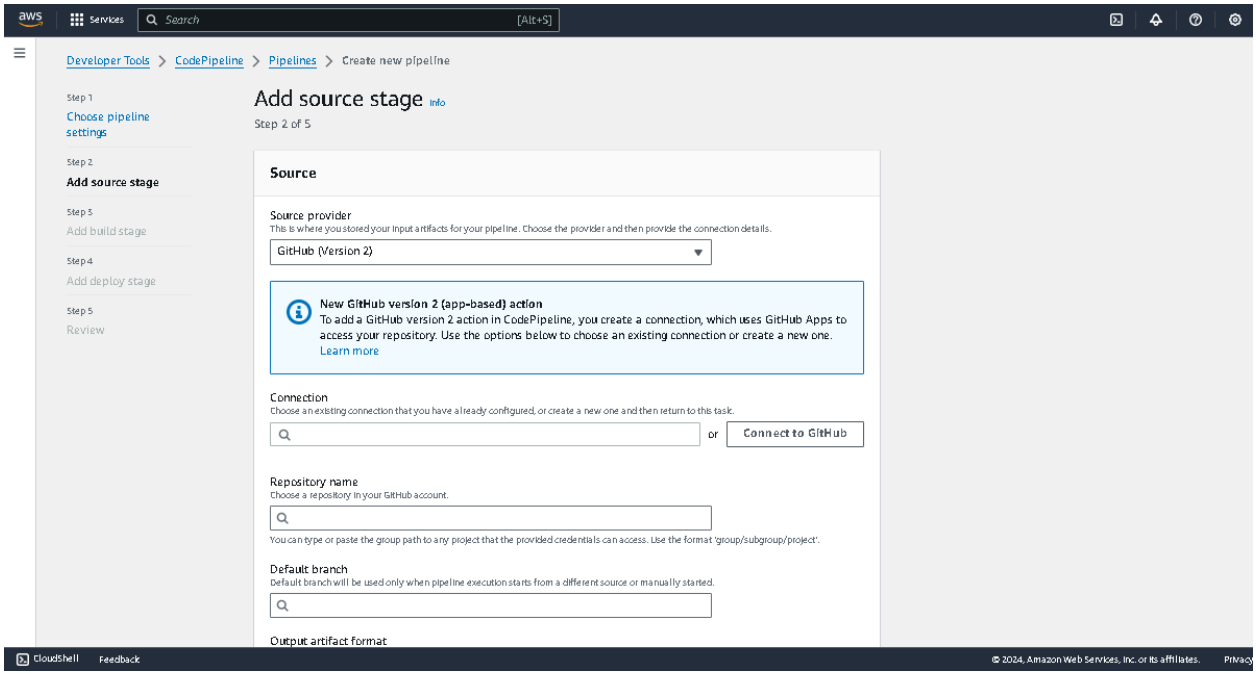
CodePipeline Setup

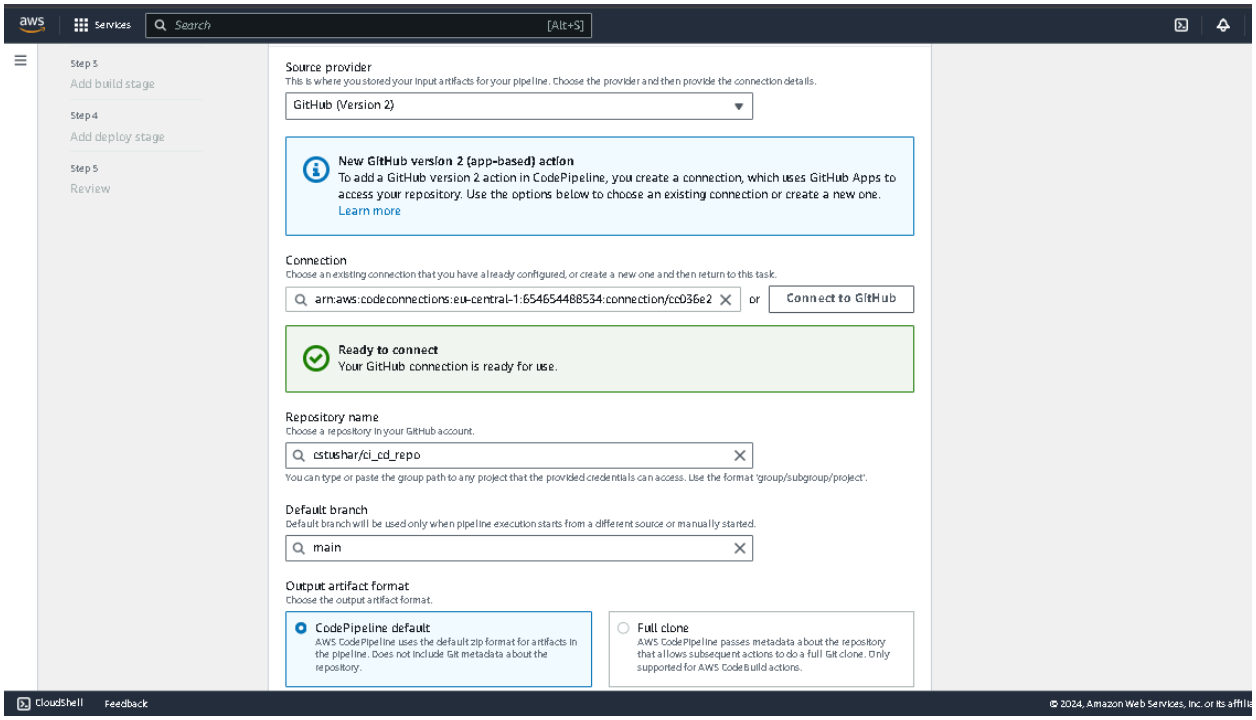
1. Create a Pipeline

- **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.

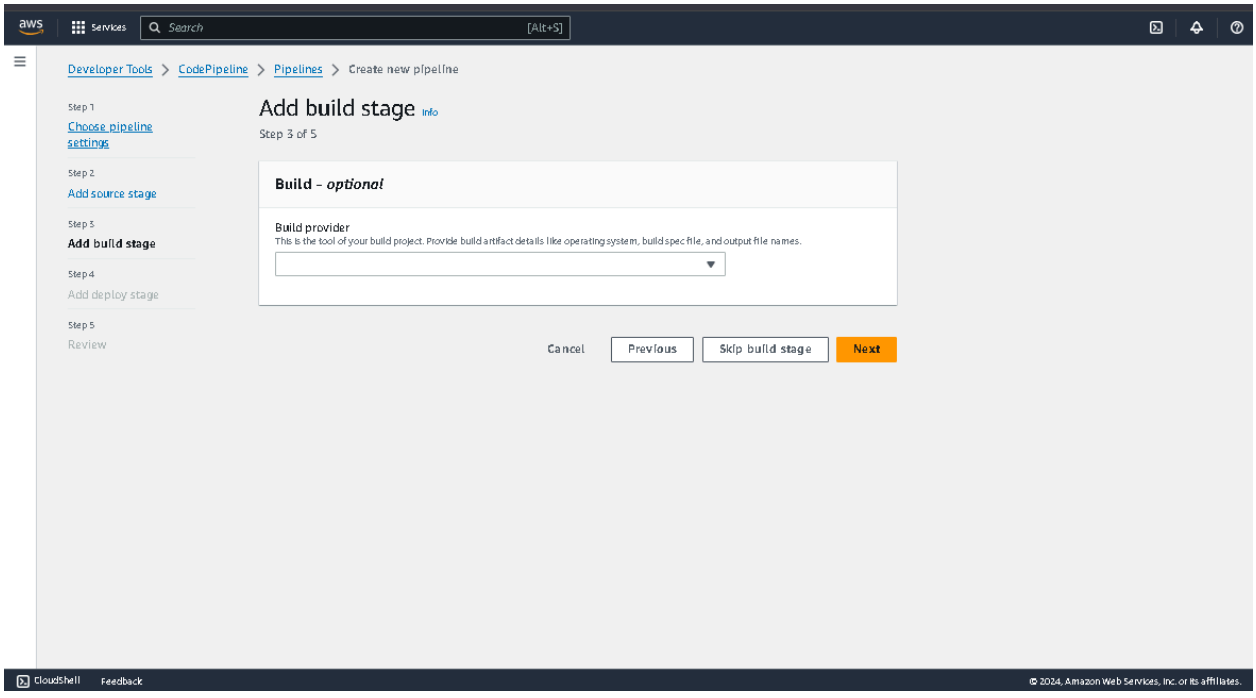

Services
Search
[Alt+S]
Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Choose pipeline settings Info
Step 1 of 5
Pipeline settings
Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
Ci_CD_CodePipeline
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


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



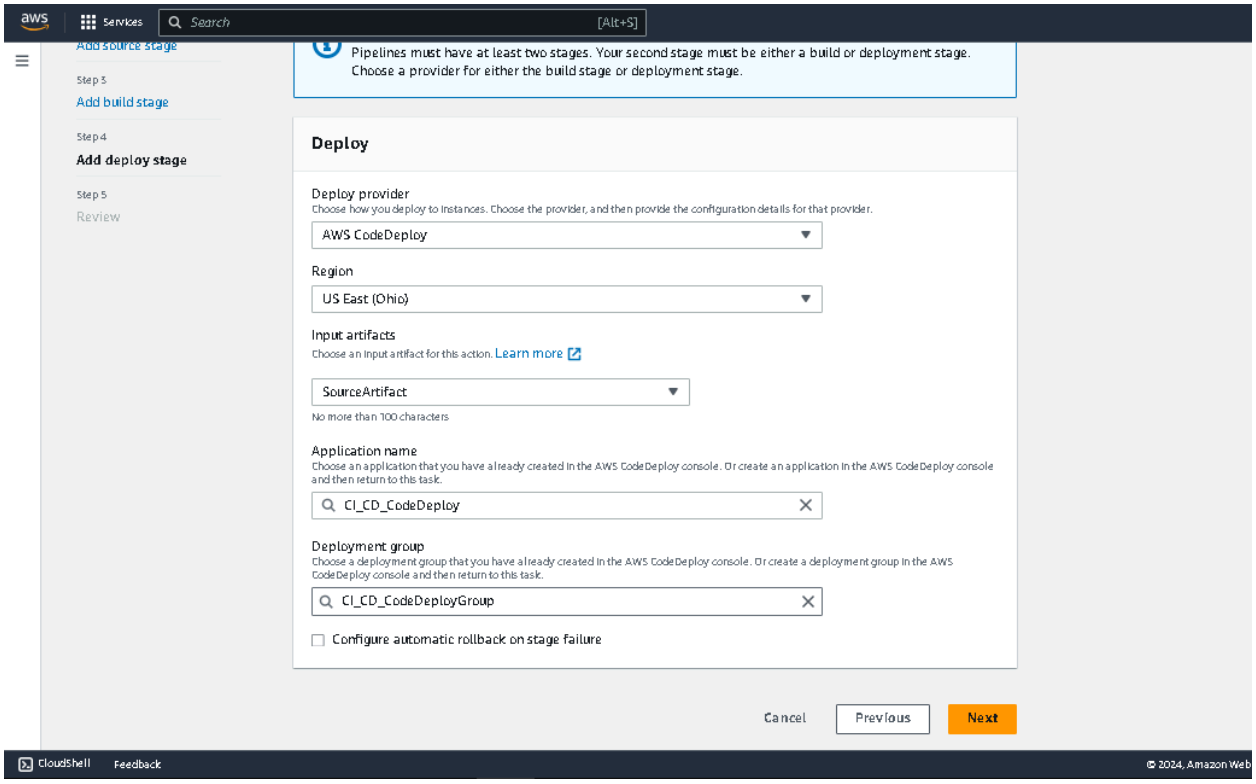
2. Skip the Build Stage (if not needed)

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



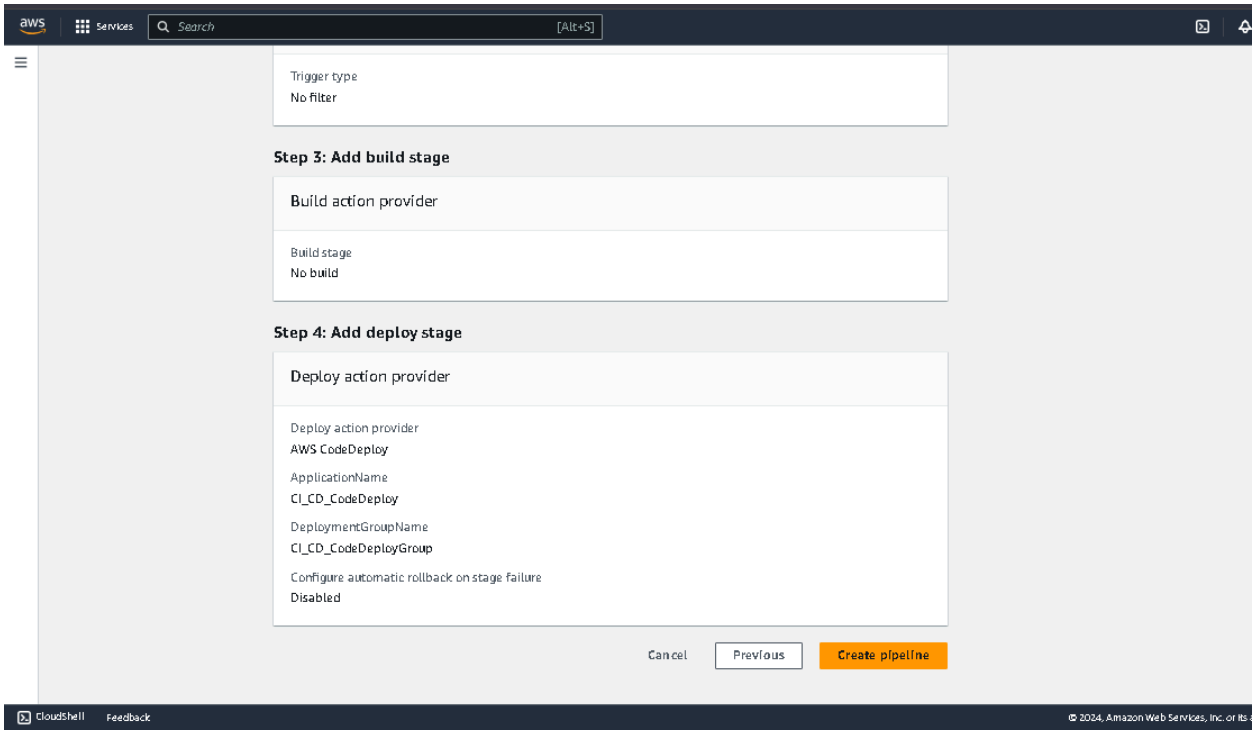
3. Deploy Using CodeDeploy

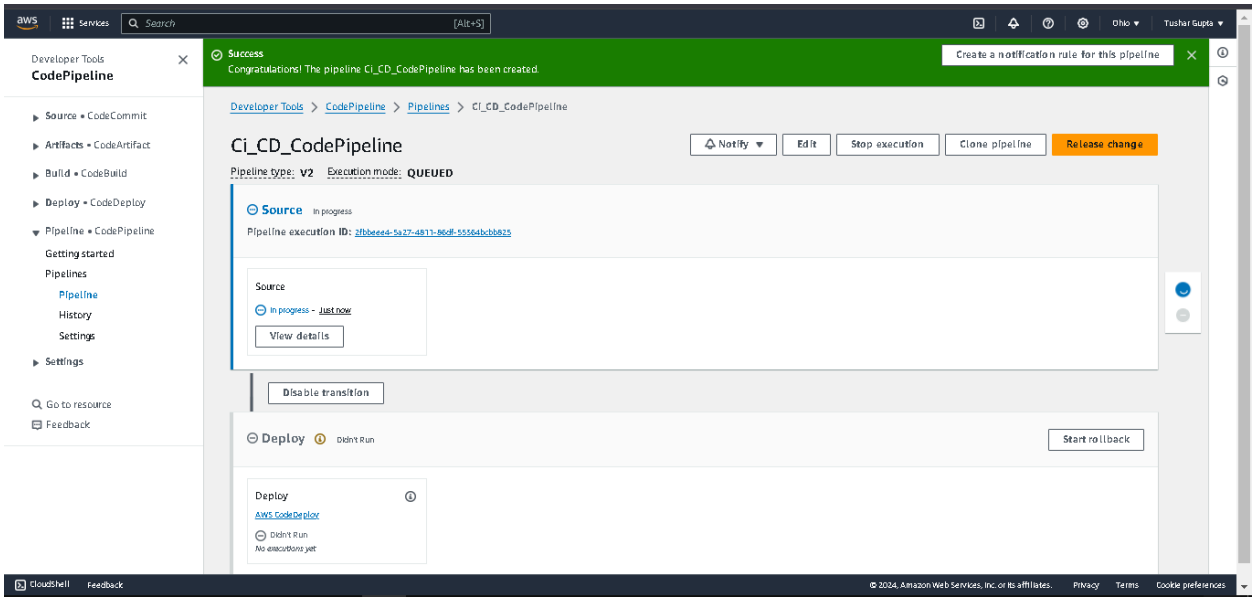
- **Region:** Select your AWS region (e.g., US East).
- **Application Name:** Choose the application created in CodeDeploy.
- **Deployment Group:** Select the deployment group created earlier.

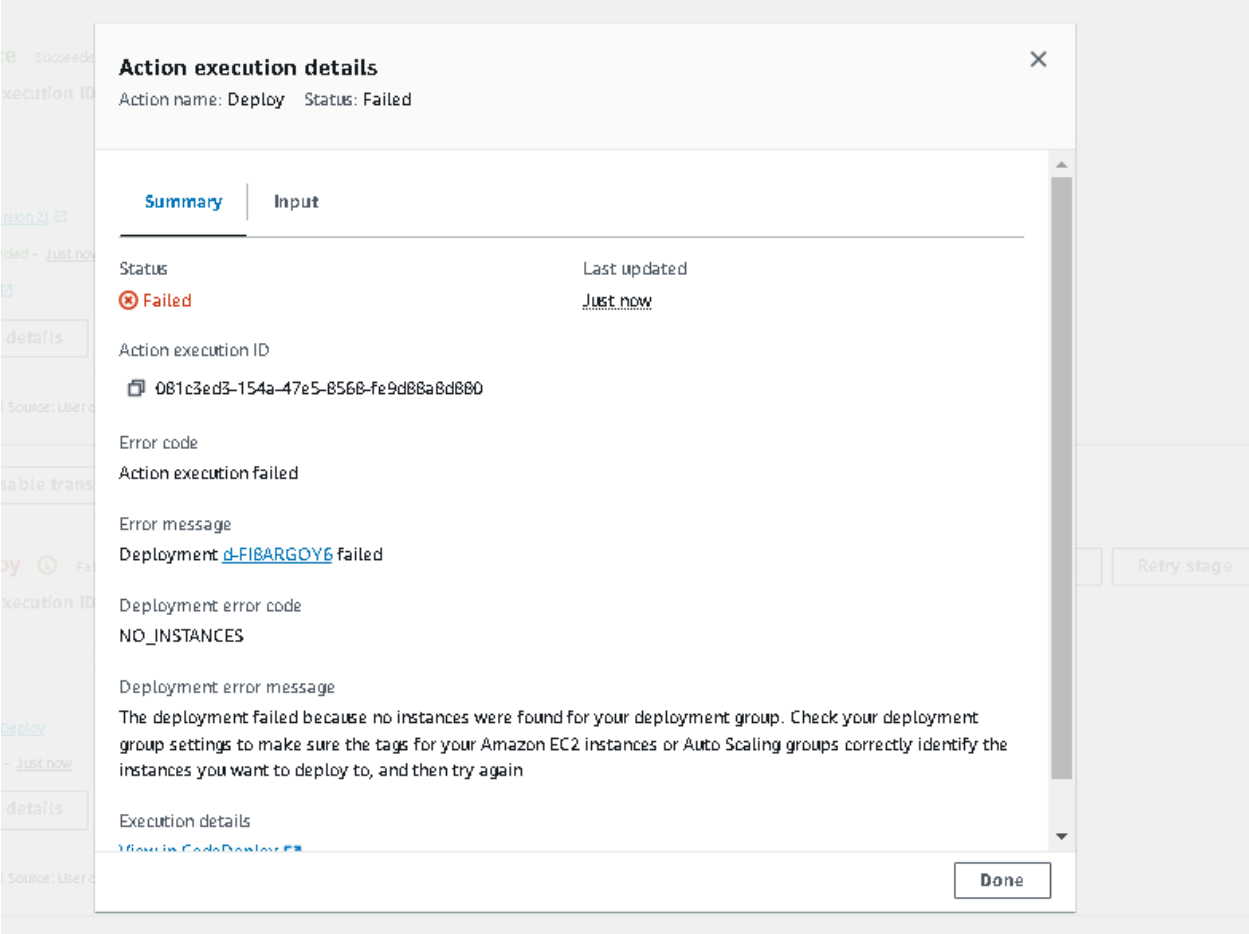


4. Verify Pipeline Creation

- 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

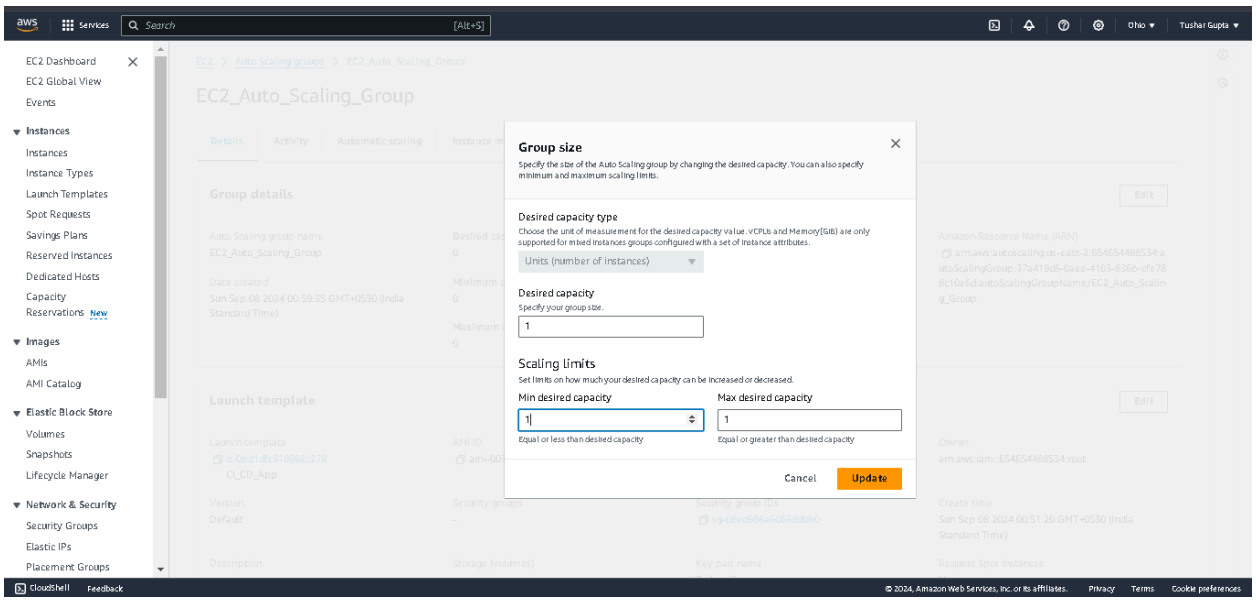

aws
Services
Search
[Alt+S]
Ohio
Tushar Gupta
Developer Tools
CodePipeline
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
History
Settings
Settings
Go to resource
Feedback
Success
Congratulations! The pipeline CI_CD_CodePipeline has been created.
Create a notification rule for this pipeline
Developer Tools > CodePipeline > Pipelines > CI_CD_CodePipeline
Ci_CD_CodePipeline
Notify
Edit
Stop execution
Clone pipeline
Release change
Pipeline type: V2
Execution mode: QUEUED
Source
In progress
Pipeline execution ID: 2fbbeee4-5a27-4811-86df-55584cbb825
Source
In progress - Just now
View details
Disable transition
Deploy
Didn't Run
Start rollback
Deploy
AWS CodeDeploy
Didn't Run
No executions yet
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

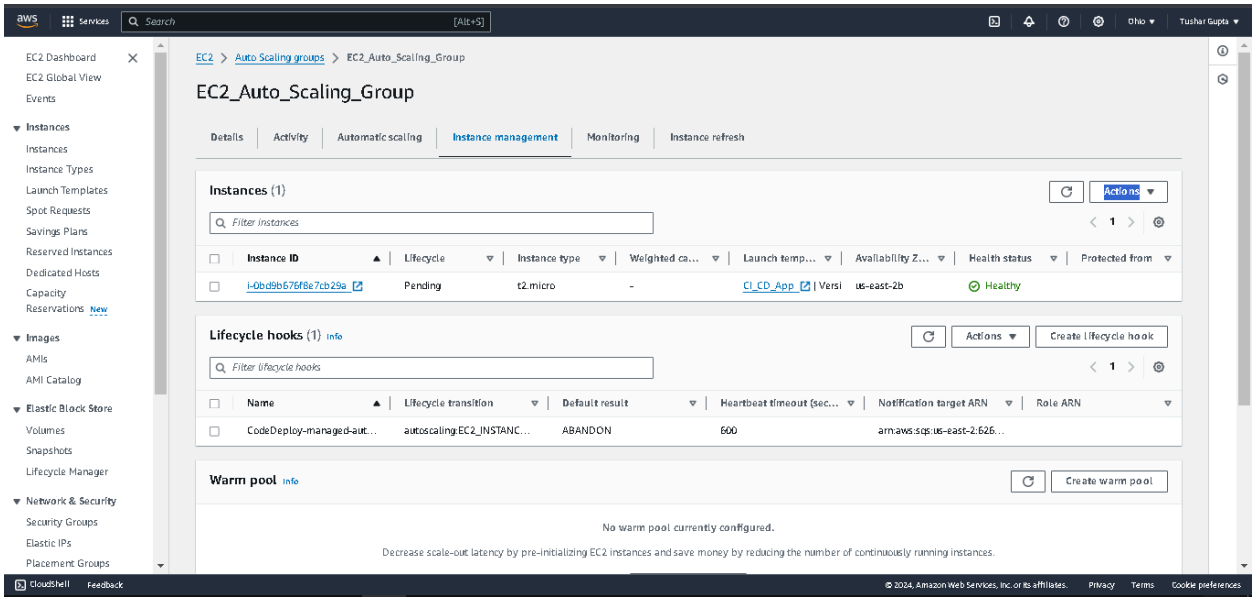


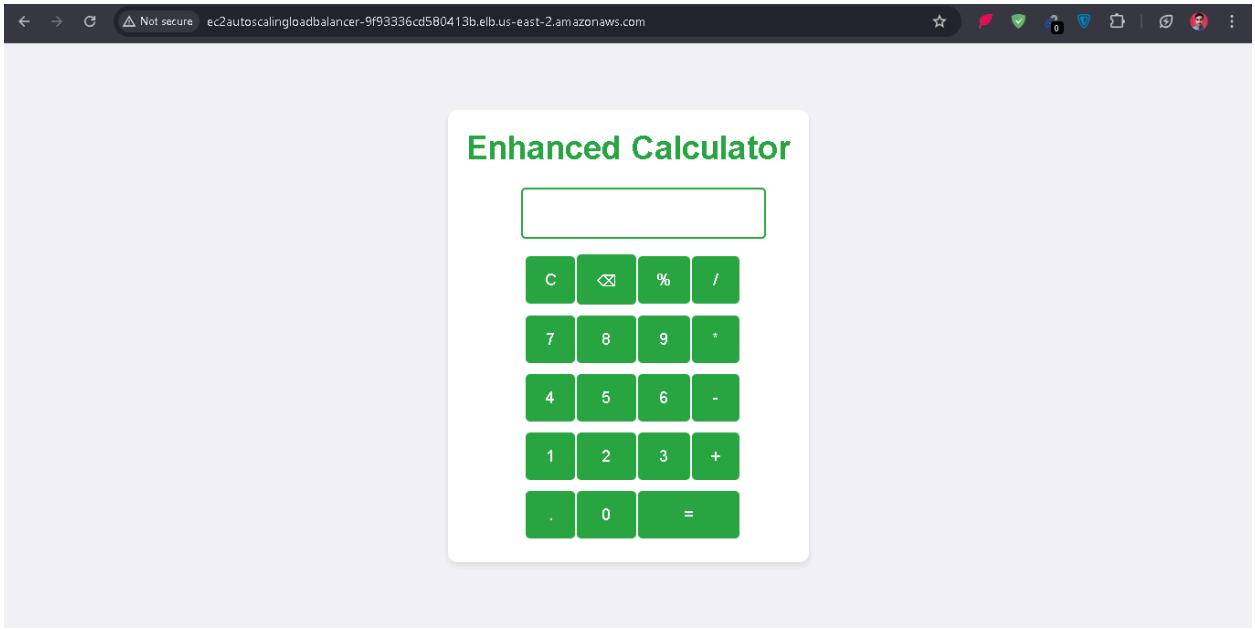
Deploying the Application

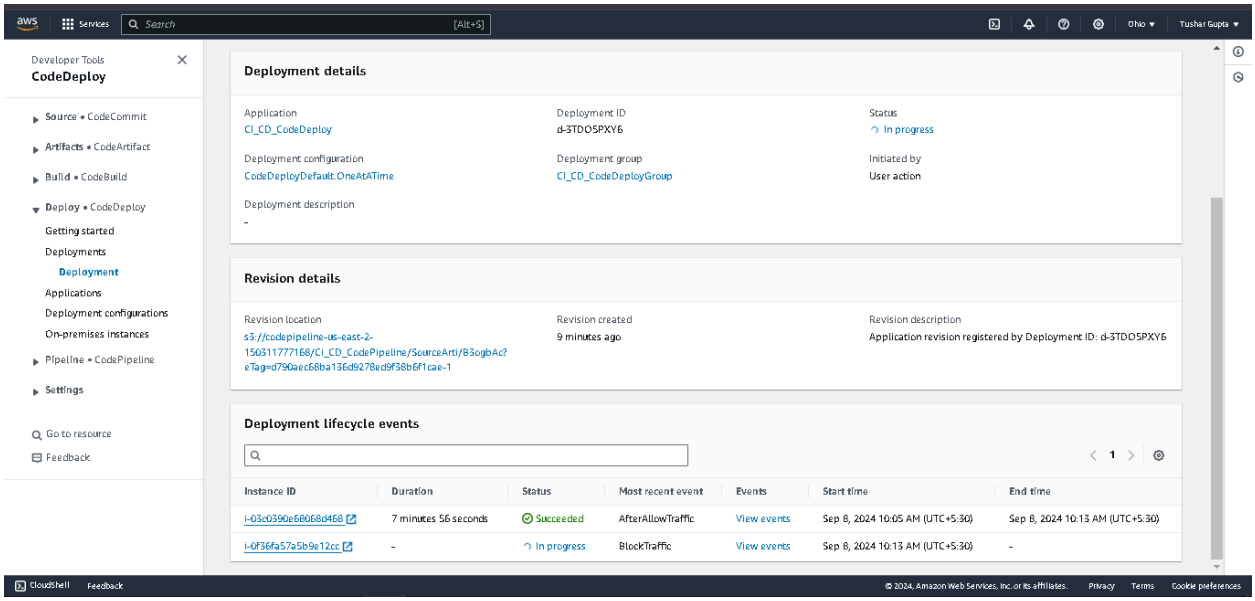
1. **Edit Auto Scaling Group Size:**
   Increase the desired capacity to 1 to initiate the creation of an EC2 instance. The instance will automatically install the CodeDeploy agent and be prepared for deployment.
2. **Monitor Deployment Events:**
   Check CodeDeploy for event logs. The deployment process 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.
3. **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.
4. **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.



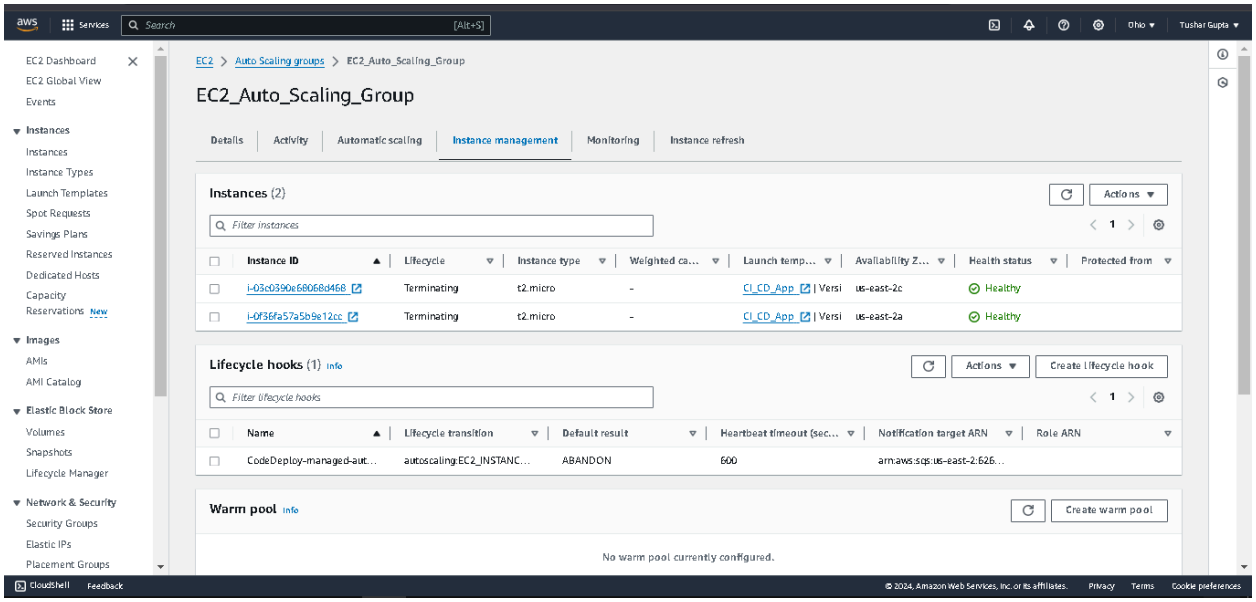


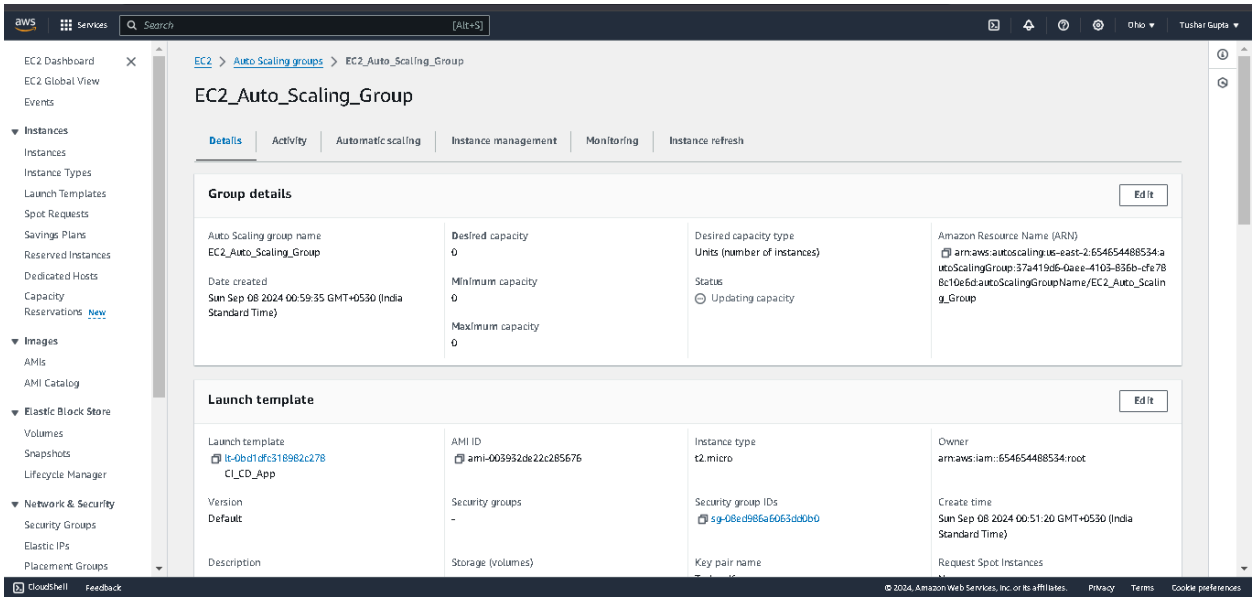




Clean-Up:

- **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.





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.