

BASIC - CI/CD SETUP

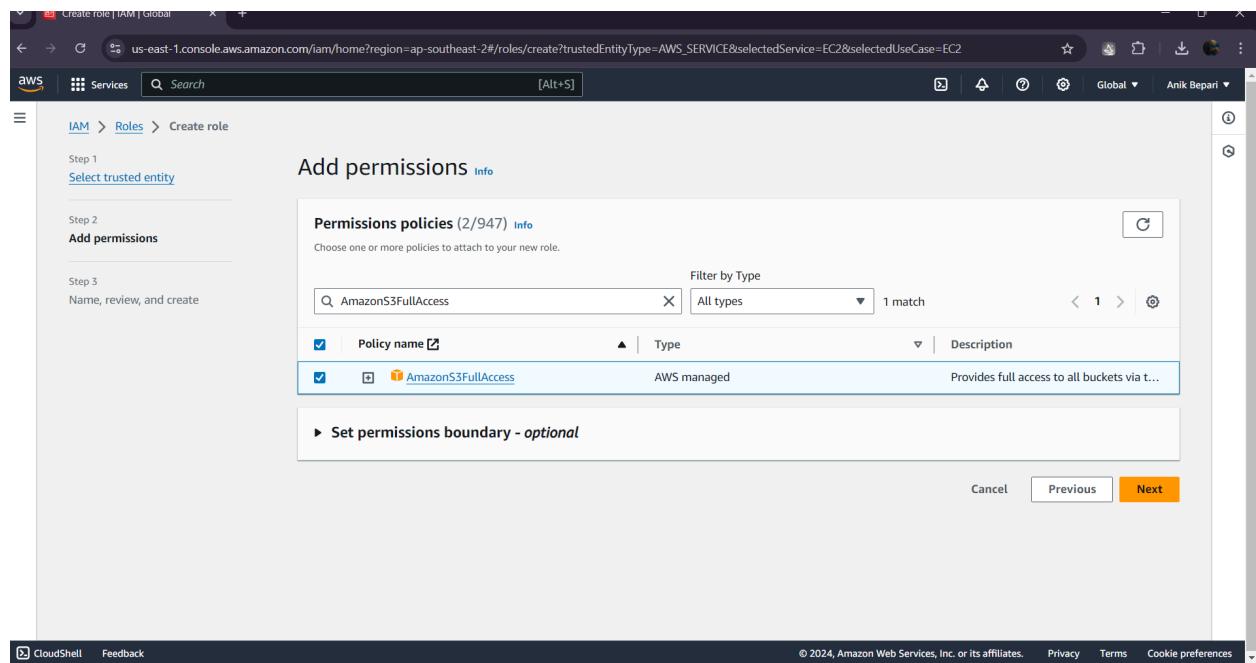
Assignment - Junior Devops Engineer Role

Done By : Anik Bepari

Step 1 - Create Two IAM roles as -

- 1. In first provide AmazonEC2RoleforAWSCodeDeploy and AmazonS3FullAccess**
- 2. In second provide codedeplyrole role. This roles will be used further.**

1



Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2RoleforAWSCodeDeploy	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Step 3: Add tags

```
2 "Version": "2012-10-17",  
3 "Statement": [  
4     {  
5         "Effect": "Allow",  
6         "Action": [  
7             "sts:AssumeRole"  
8         ],  
9         "Principal": {  
10             "Service": [  
11                 "ec2.amazonaws.com"  
12             ]  
13         }  
14     }  
15 ]  
16 }
```

2

Step 1: Select trusted entities

Trust policy

```
1 [ {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Sid": "",  
6             "Effect": "Allow",  
7             "Principal": {  
8                 "Service": [  
9                     "codedeploy.amazonaws.com"  
10                 ]  
11             },  
12             "Action": [  
13                 "sts:AssumeRole"  
14             ]  
15         }  
16     ]  
17 } ]
```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSCodeDeployRole	AWS managed	Permissions policy

Step 2 - Create 4 ec2 instances and put user data in it -

ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#LaunchInstances:

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: cicd-assignment Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

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

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux Enterprise Server

Quick Start: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux Enterprise Server

Summary

Number of instances Info: 4

When launching more than 1 instance, consider EC2 Auto Scaling

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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel **Launch instance** Review commands

Summary

Number of instances Info: 4

When launching more than 1 instance, consider EC2 Auto Scaling

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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel **Launch instance** Review commands

Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro	Free tier eligible		
Family: t2	1 vCPU	1 GiB Memory	Current generation: true
On-Demand SUSE base pricing: 0.0146 USD per Hour			
On-Demand Linux base pricing: 0.0146 USD per Hour			
On-Demand Windows base pricing: 0.0192 USD per Hour			
On-Demand RHEL base pricing: 0.029 USD per Hour			

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 - required

[Create new key pair](#)

Network settings [Info](#)

Summary

Number of instances [Info](#)

When launching more than 1 instance, consider [EC2 Auto Scaling](#)

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](#) [Launch instance](#)

ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#LaunchInstances:

[Services](#) [Search](#) [Alt+S]

Description - required [Info](#)

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type [Info](#) Protocol [Info](#) Port range [Info](#)

ssh	TCP	22
-----	-----	----

Source type [Info](#) Source [Info](#) Description - optional [Info](#)

Anywhere	Add CIDR, prefix list or security	e.g. SSH for admin desktop
0.0.0.0/0		

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type [Info](#) Protocol [Info](#) Port range [Info](#)

HTTP	TCP	80
------	-----	----

Source type [Info](#) Source [Info](#) Description - optional [Info](#)

Anywhere	Add CIDR, prefix list or security	e.g. SSH for admin desktop
0.0.0.0/0		

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting [X](#)

Summary

Number of instances [Info](#)

When launching more than 1 instance, consider [EC2 Auto Scaling](#)

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](#) [Launch instance](#) [Review commands](#)

Advanced details

- Domain join directory: Select
- IAM instance profile: EC2CodeDeployRole (arn:aws:iam:023810699384:instance-profile/EC2CodeDeployRole)
- Hostname type: IP name
- DNS Hostname: Enable IP name IPv4 (A record) DNS requests Enable resource-based IPv4 (A record) DNS requests Enable resource-based IPv6 (AAAA record) DNS requests
- Instance auto-recovery: Select
- Shutdown behavior: Stop

Summary

Number of instances: 4

When launching more than 1 instance, consider EC2 Auto Scaling

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.

User data - optional

```
#!/bin/bash
sudo yum -y update
sudo yum -y install ruby wget
cd /home/ec2-user
wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install
sudo chmod +x ./install
sudo ./install auto
sudo yum install -y python-pip
sudo pip install awscli
```

User data has already been base64 encoded

Step 3 - Create an of any of the instance which we will use in AutoScalingGroup.

The screenshot shows the AWS EC2 Instances page with five instances listed. The instance **i-08e1470226500d424 (cicd-assignment)** is selected. A modal window titled **Create image** is open, showing the instance ID and an image name **Autoscaling-image**.

Step 4 - Create Application Load Balancer.

1. Create target group than load balancer , in target group give path to /index.html , which is the app file containing code over git.

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level.	Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and	Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable

Step 1
Specify group details

Step 2
Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

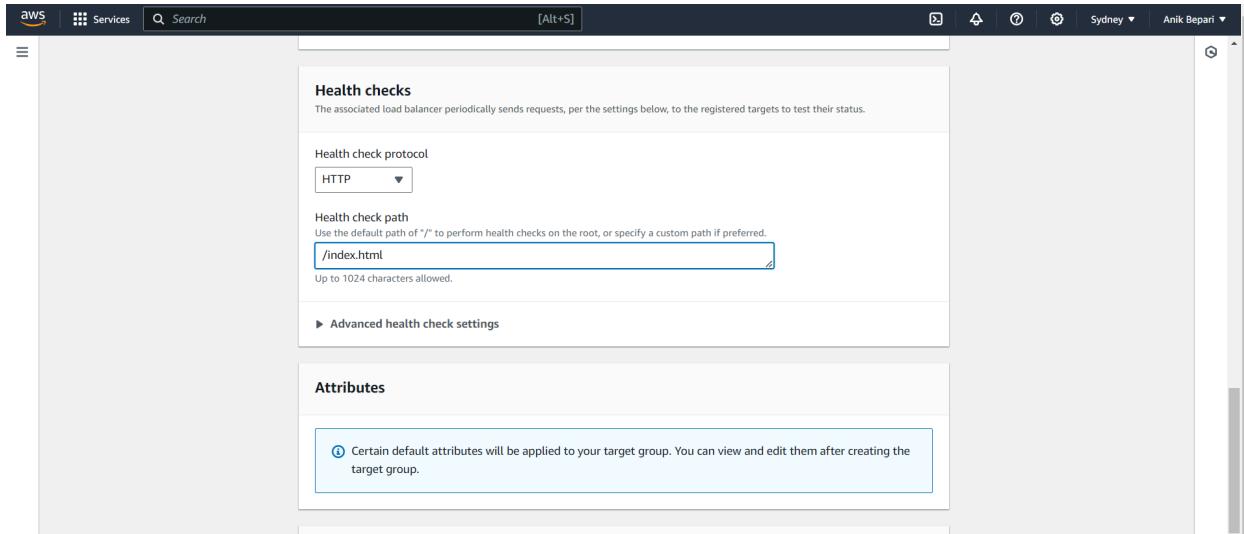
Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function



Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol
HTTP

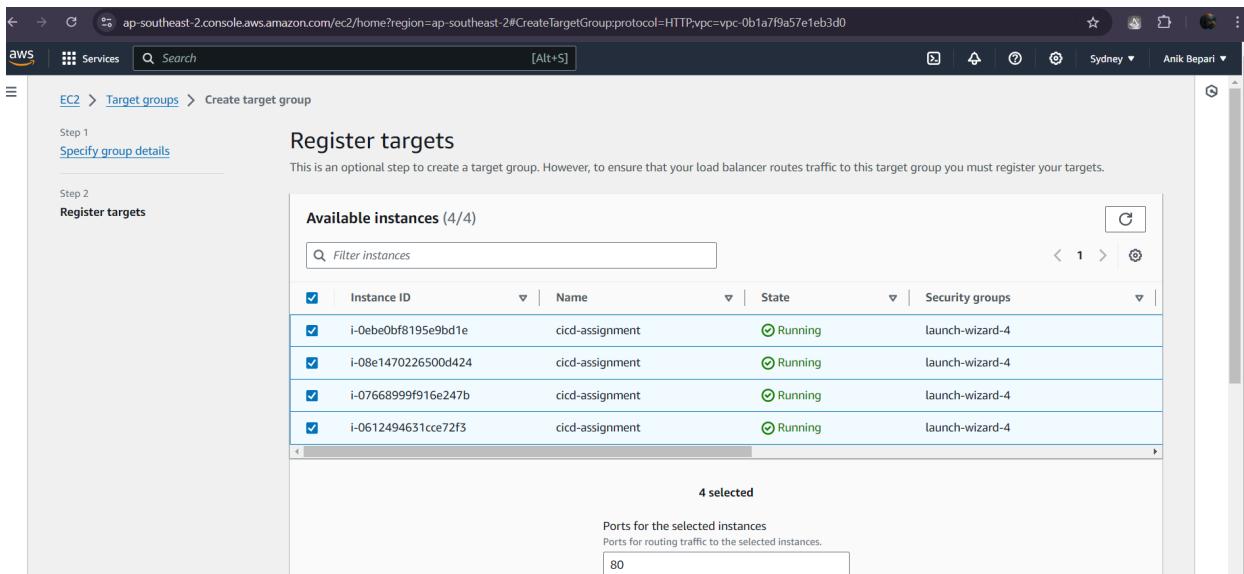
Health check path
/index.html
Up to 1024 characters allowed.

► Advanced health check settings

Attributes

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Now Create Application Load Balancer -



EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (4/4)

Instance ID	Name	State	Security groups
i-0ebe0bf8195e9bd1e	cicd-assignment	Running	launch-wizard-4
i-08e1470226500d424	cicd-assignment	Running	launch-wizard-4
i-07668999f916e247b	cicd-assignment	Running	launch-wizard-4
i-0612494631cce72f3	cicd-assignment	Running	launch-wizard-4

4 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
80

aws Services Search [Alt+S] Sydney Anik Bepari

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.

Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal
An internal load balancer routes requests from clients to targets using private IP addresses. Compatible with the IPv4 and Dualstack IP address types.

aws Services Search [Alt+S] Sydney Anik Bepari

launch-wizard-4 X
sg-078567a6067cc256e VPC: vpc-0b1a7f9a57e1eb5d0

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

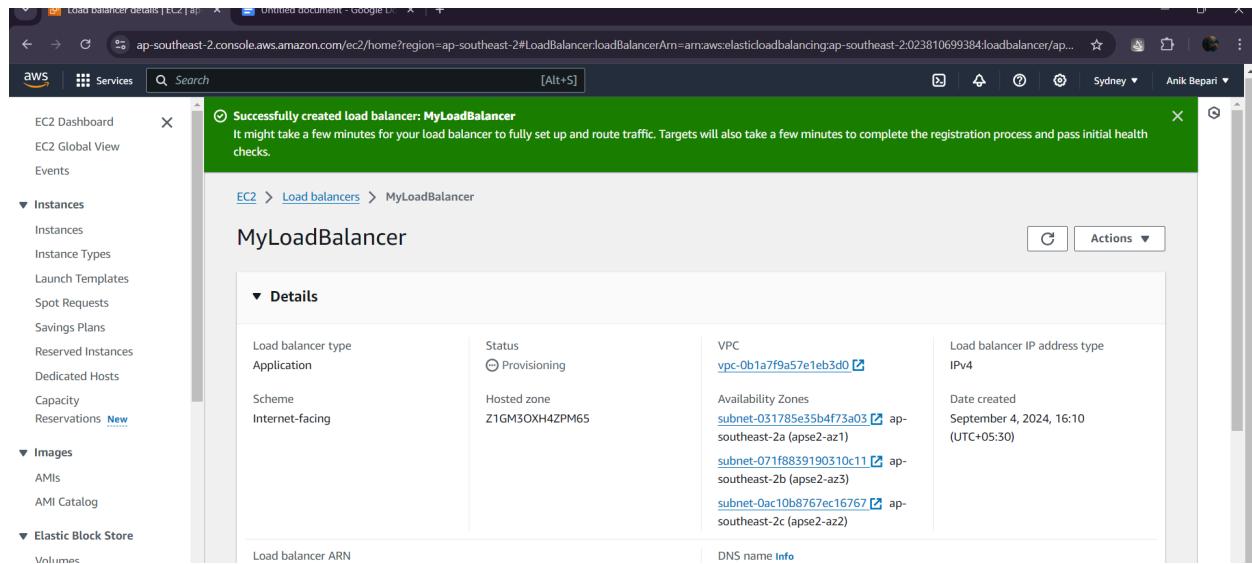
▼ Listener HTTP:80 Remove

Protocol	Port	Default action <small>Info</small>
HTTP	: 80	Forward to code-deployTG Target type: Instance, IPv4 1-65535 Create target group

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

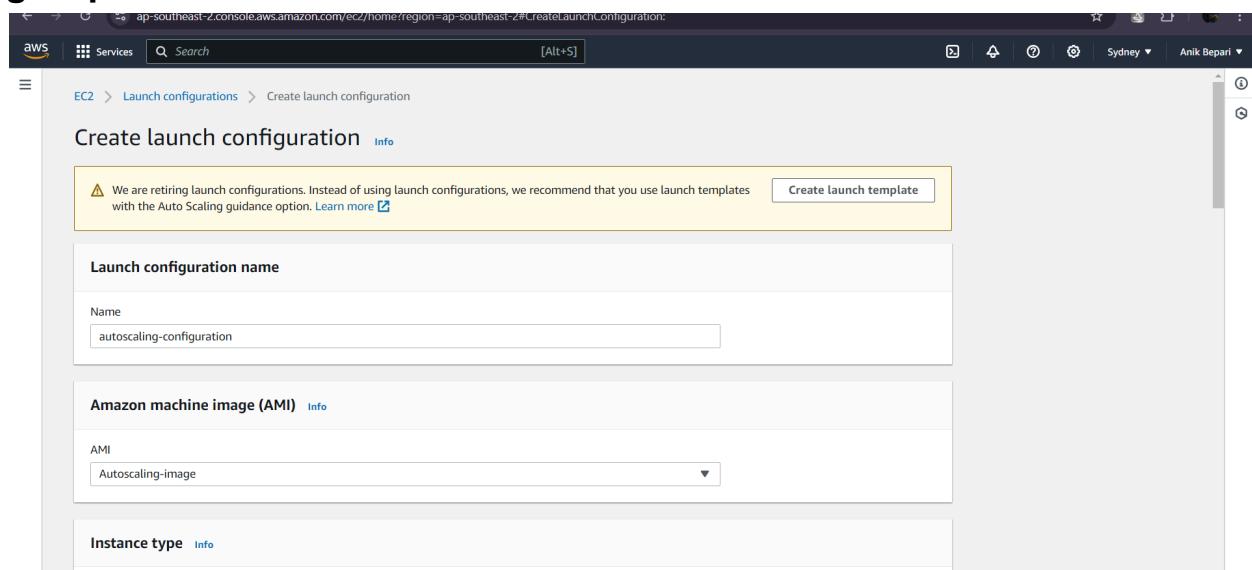
[Add listener](#)



Successfully created load balancer: MyLoadBalancer
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

Details	
Load balancer type	Application
Status	Provisioning
VPC	vpc-0b1a7f9a57e1eb3d0
Scheme	Internet-facing
Hosted zone	Z1GM3OXH4ZPM65
Availability Zones	subnet-031785e35b4f73a03 ap-southeast-2a (apse2-az1) subnet-071f8839190310c11 ap-southeast-2b (apse2-az3) subnet-0ac10b8767ec16767 ap-southeast-2c (apse2-az2)
Load balancer ARN	DNS name Info
Load balancer IP address type	IPv4
Date created	September 4, 2024, 16:10 (UTC+05:30)

Step 5 - Create Launch Configurations and than create Auto scaling group.



We are retiring launch configurations. Instead of using launch configurations, we recommend that you use launch templates with the Auto Scaling guidance option. [Learn more](#)

Launch configuration name

Name: autoscaling-configuration

Amazon machine image (AMI)

AMI: Autoscaling-image

Instance type

aws Services Search [Alt+S] Sydney Anik Bepari

Instance type
t2.micro (1 vCPUs, 1 GiB, EBS Only) Choose instance type

Additional configuration - optional

Purchasing option [Info](#)
 Request Spot Instances

IAM instance profile [Info](#)
EC2CodeDeployRole

Monitoring [Info](#)
 Enable EC2 instance detailed monitoring within CloudWatch

EBS-optimized instance
 Launch as EBS-optimized instance

▶ Advanced details

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#CreateLaunchConfiguration:

aws Services Search [Alt+S] Sydney Anik Bepari

<input type="checkbox"/> sg-0cd6b59911e140260	launch-wizard-3	vpc-0b1a7f9a57e1eb3d0	launch-wizard-3 created 2024-07-19T12:23:39.891Z
<input type="checkbox"/> sg-00231d6ad5def6204	launch-wizard-1	vpc-0b1a7f9a57e1eb3d0	launch-wizard-1 created 2024-03-26T16:54:33.614Z
<input checked="" type="checkbox"/> sg-078567a6067cc256e	launch-wizard-4	vpc-0b1a7f9a57e1eb3d0	launch-wizard-4 created 2024-09-04T09:30:43.279Z
<input type="checkbox"/> sg-0ad4d42081d5cfaeb	default	vpc-0b1a7f9a57e1eb3d0	default VPC security group

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Key pair (login) [Info](#)

Key pair options
Choose an existing key pair

Existing key pair
Choose a key pair

I acknowledge that I have access to the selected private key file (.pem), and that without this file, I won't be able to log into my instance.

Cancel **Create launch configuration**

create Auto scaling group.

Services Search [Alt+S] Sydney Anik Bepari

Step 1 Choose launch template or configuration

Step 2 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

Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name
Enter a name to identify the group.

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

Launch configuration Info [Switch to launch template](#)

⚠ Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, [see the documentation](#).

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

[Create a launch configuration](#) [C](#)

Launch configuration	AMI ID	Date created
----------------------	--------	--------------

Step 1 Choose launch template or configuration

Step 2 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

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

[Create a VPC](#) [C](#)

172.31.0.0/16 Default

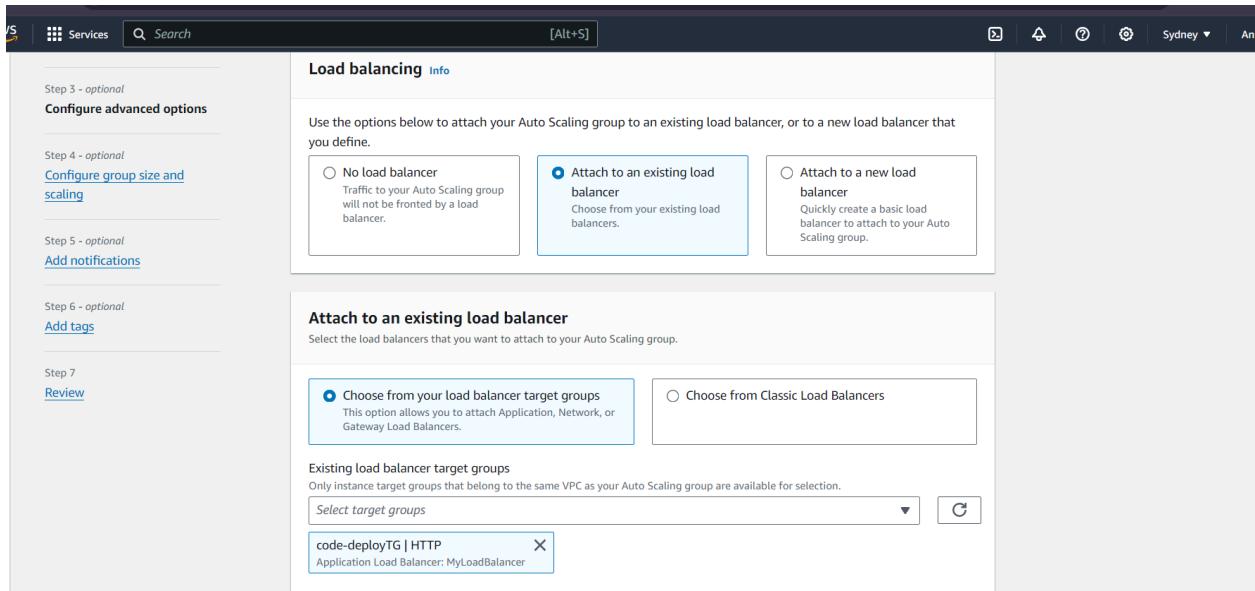
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

[C](#)

ap-southeast-2b | subnet-071f8839190310c11 X
172.31.32.0/20 Default

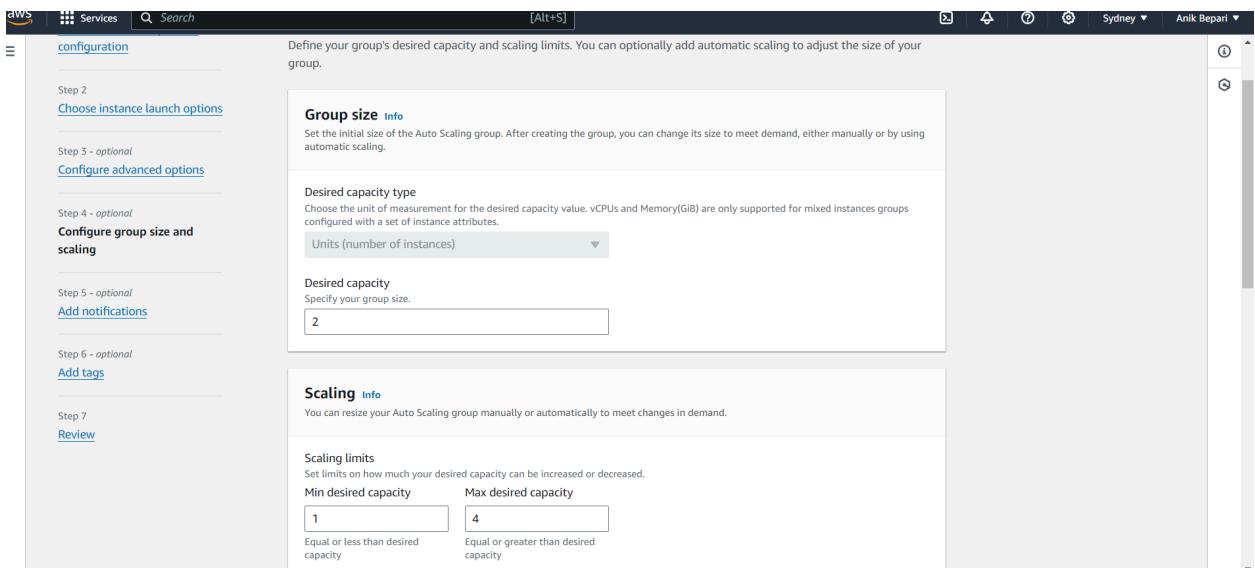
ap-southeast-2a | subnet-031785e35b4f73a03 X
172.31.0.0/20 Default

ap-southeast-2c | subnet-0ac10b8767ec16767 X
172.31.16.0/20 Default

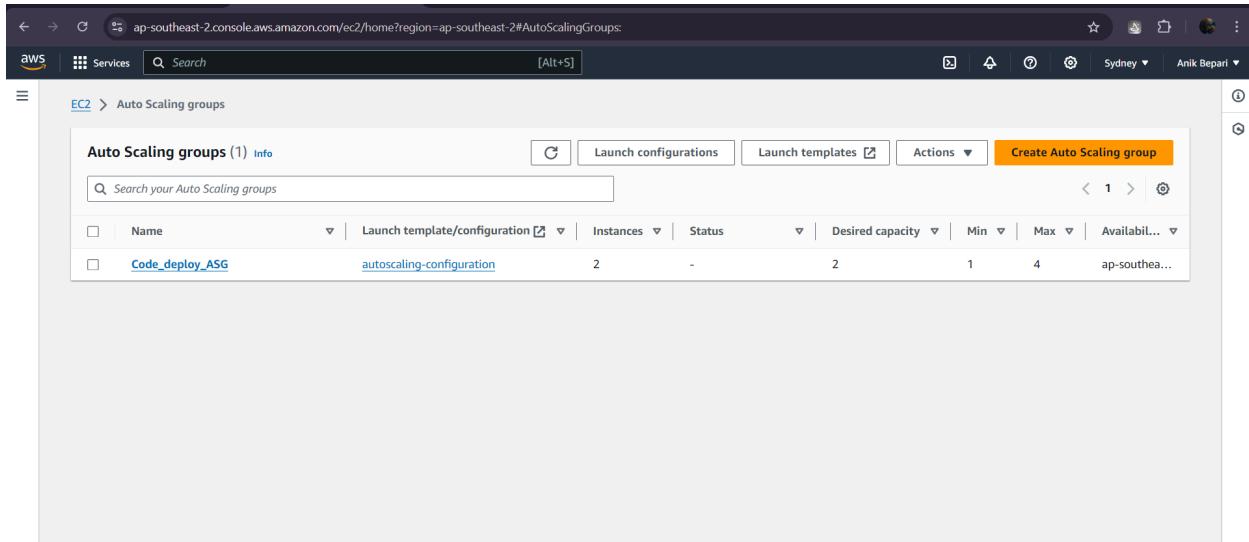


The screenshot shows the AWS Auto Scaling configuration interface. The left sidebar lists steps from 3 to 7. Step 6 is currently selected, titled 'Configure advanced options'. The main content area is titled 'Load balancing' and contains three options: 'No load balancer' (radio button not selected), 'Attach to an existing load balancer' (radio button selected), and 'Attach to a new load balancer' (radio button not selected). The 'Attach to an existing load balancer' section is expanded, showing a sub-section 'Attach to an existing load balancer' with a note to select load balancers. Below this is a 'Choose from your load balancer target groups' section, which is also expanded, showing a dropdown menu with 'code-deployTG | HTTP' and 'Application Load Balancer: MyLoadBalancer' selected. Other options like 'Choose from Classic Load Balancers' are shown but not selected.

I have attached Application Load Balancer to it which is already Created



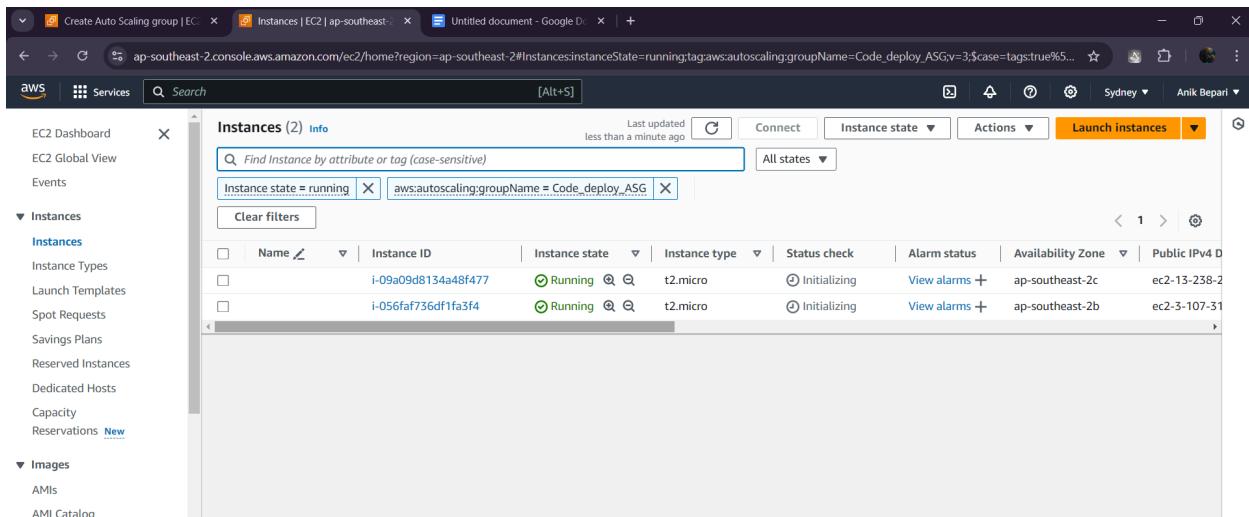
The screenshot shows the AWS Auto Scaling configuration interface. The left sidebar lists steps from 2 to 7. Step 4 is selected, titled 'Configure group size and scaling'. The main content area is divided into two sections: 'Group size' and 'Scaling'. The 'Group size' section is expanded, showing a note to define desired capacity and scaling limits. It includes a 'Desired capacity type' dropdown set to 'Units (number of instances)', a 'Desired capacity' input field with the value '2', and a note that vCPUs and Memory(GiB) are only supported for mixed instances groups. The 'Scaling' section is also expanded, showing a note to resize the group manually or automatically. It includes 'Scaling limits' settings with 'Min desired capacity' set to '1' and 'Max desired capacity' set to '4', with a note that the range is 'Equal or less than desired capacity' to 'Equal or greater than desired capacity'.



The screenshot shows the AWS Auto Scaling Groups page. At the top, there is a search bar and a navigation bar with tabs for 'Launch configurations', 'Launch templates', 'Actions', and 'Create Auto Scaling group'. Below the search bar is a table with one row, showing the ASG details:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zone
Code_deploy_ASG	autoscaling-configuration	2	-	2	1	4	ap-southeast-...

Auto Scaling Group is created. We can see as soon as ASG created , the two instances are up.



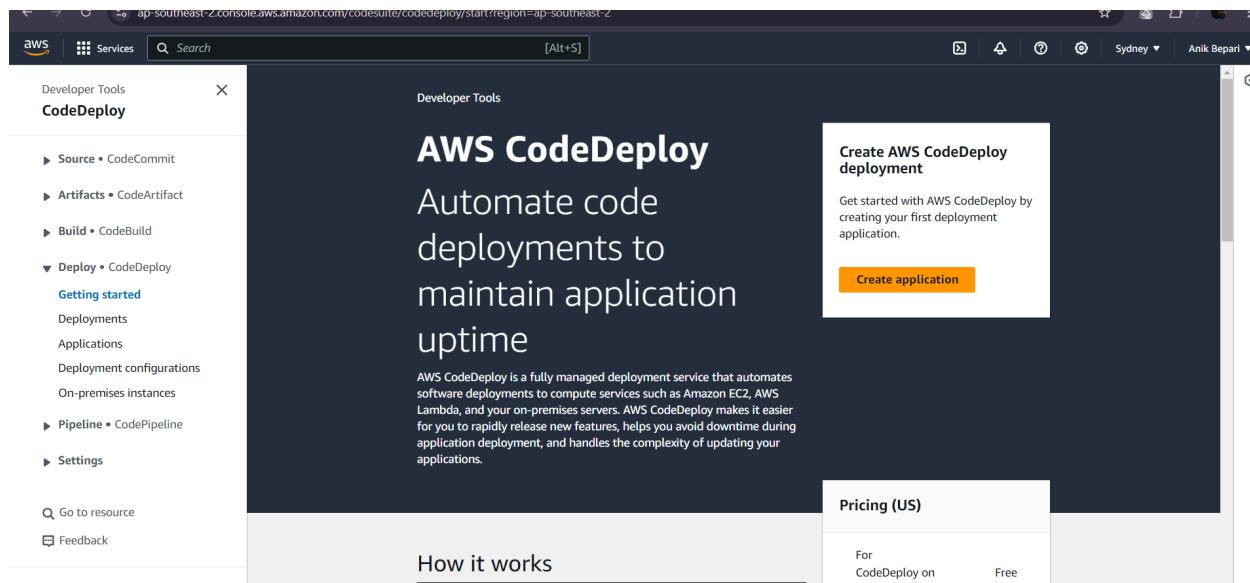
The screenshot shows the AWS Instances page. The left sidebar has a 'Instances' section with options like 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity', and 'Reservations'. The main area shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
	i-09a09d8134a48f477	Running	t2.micro	Initializing	View alarms	ap-southeast-2c	ec2-13-238-2
	i-056faf736df1fa3f4	Running	t2.micro	Initializing	View alarms	ap-southeast-2b	ec2-3-107-31

```
[ec2-user@ip-172-31-9-9 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-9-9 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
  Active: active (running) since Wed 2024-09-04 13:14:21 UTC; 1min 31s ago
    Main PID: 26792 (java)
      Tasks: 39 (Limit: 1112)
     Memory: 214.3M
        CPU: 1min 32.186s
      CGroup: /system.slice/jenkins.service
              └─26792 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war

Sep 04 13:13:43 ip-172-31-9-9.ap-southeast-2.compute.internal jenkins[26792]: Please use the following password to proc>
Sep 04 13:13:43 ip-172-31-9-9.ap-southeast-2.compute.internal jenkins[26792]: 140e6e1b1ba94c09a7d64dc99e49bd90>
Sep 04 13:13:43 ip-172-31-9-9.ap-southeast-2.compute.internal jenkins[26792]: This may also be found at: /var/lib/jenki>
Sep 04 13:13:43 ip-172-31-9-9.ap-southeast-2.compute.internal jenkins[26792]: ****=>
Sep 04 13:14:21 ip-172-31-9-9.ap-southeast-2.compute.internal jenkins[26792]: 2024-09-04 13:14:21.404+0000 [id=31]>
Sep 04 13:14:21 ip-172-31-9-9.ap-southeast-2.compute.internal jenkins[26792]: 2024-09-04 13:14:21.646+0000 [id=24]>
Sep 04 13:14:21 ip-172-31-9-9.ap-southeast-2.compute.internal systemd[1]: Started jenkins.service - Jenkins Continuous>
Sep 04 13:14:26 ip-172-31-9-9.ap-southeast-2.compute.internal jenkins[26792]: 2024-09-04 13:14:26.736+0000 [id=58]>
lines 1-20/20 (END)client_loop: send disconnect: Connection reset
```

Step 7 - Create CodeDeploy



The screenshot shows the AWS CodeDeploy console in a web browser. The URL is `ap-southeast-2.console.aws.amazon.com/codesuite/codedeploy/start?region=ap-southeast-2`. The left sidebar is titled 'Developer Tools' and has a 'CodeDeploy' section with the following sub-options: 'Source • CodeCommit', 'Artifacts • CodeArtifact', 'Build • CodeBuild', 'Deploy • CodeDeploy' (which is expanded to show 'Getting started', 'Deployments', 'Applications', 'Deployment configurations', 'On-premises instances', 'Pipeline • CodePipeline', and 'Settings'), 'Go to resource', and 'Feedback'. The main content area has a dark background with the title 'AWS CodeDeploy' and the sub-headline 'Automate code deployments to maintain application uptime'. It includes a description of AWS CodeDeploy as a fully managed deployment service and a 'Create AWS CodeDeploy deployment' button. Below this, there's a 'How it works' section and a 'Pricing (US)' table. The table shows that for CodeDeploy on AWS Lambda, the cost is 'Free'. The top of the browser window shows the AWS logo, a search bar, and the region 'Sydney'.

Create application

AWS Services Search [Alt+S] Sydney Anil

Developer Tools > CodeDeploy > Applications > Create application

Create application

Application configuration

Application name
Enter an application name
 100 character limit

Compute platform
Choose a compute platform

Tags

Now Create Deployment Group

AWS Services Search [Alt+S] Sydney Anil

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

Create deployment group

Application

Application
MyApp
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name
 100 character limit

Service role

Now Choose Amazon Ec2 ASG which we have Created earlier

Choose how to deploy your application

In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

Blue/green
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

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

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

Now choose Application Load balancer which we have created earlier.

an application is deployed and the success or failure conditions for a deployment.

CodeDeployDefault.AllAtOnce

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

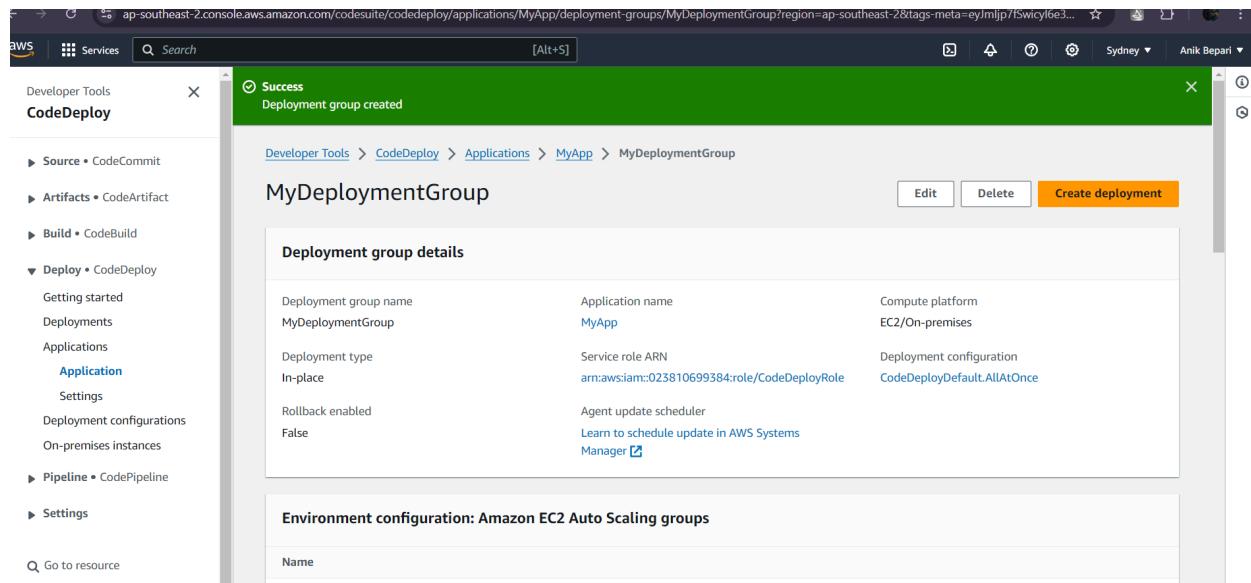
Load balancer type

Application Load Balancer or Network Load Balancer

Choose target groups

Classic Load Balancer

Deployment Group Is Created. Now we will use Jenkins for Creating a pipeline that will use git as a source and will run the CodeDeploy.

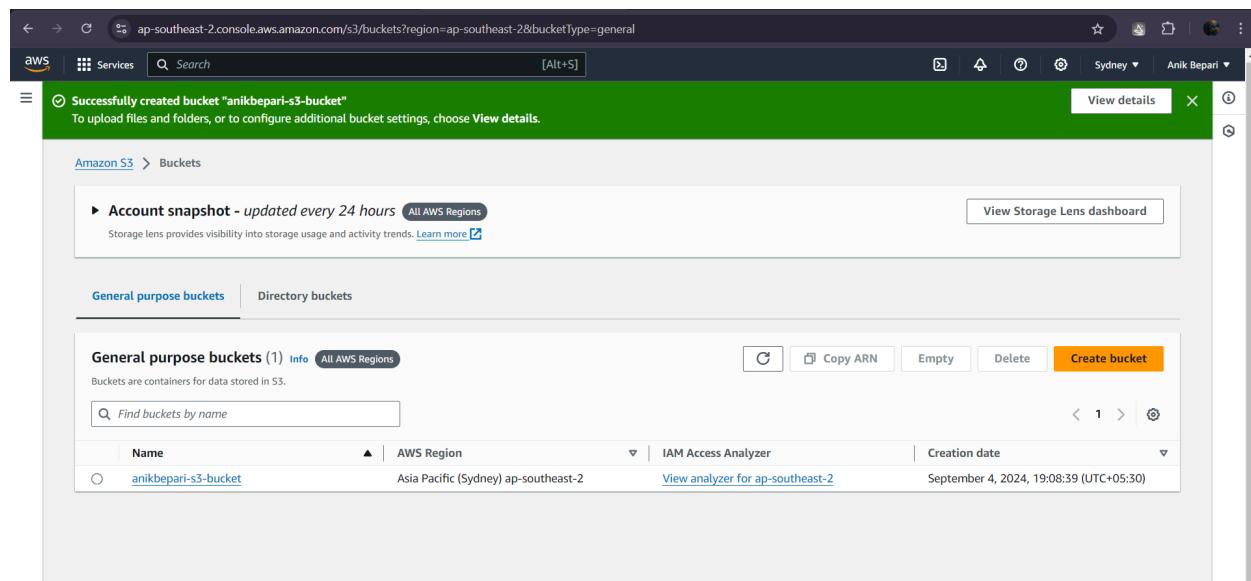


The screenshot shows the AWS CodeDeploy console with a success message: "Deployment group created". The navigation path is: Developer Tools > CodeDeploy > Applications > MyApp > MyDeploymentGroup. The main section displays "Deployment group details" for "MyDeploymentGroup". Key details include:

Deployment group name	Application name	Compute platform
MyDeploymentGroup	MyApp	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::023810699384:role/CodeDeployRole	CodeDeployDefault.AllAtOnce
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager	

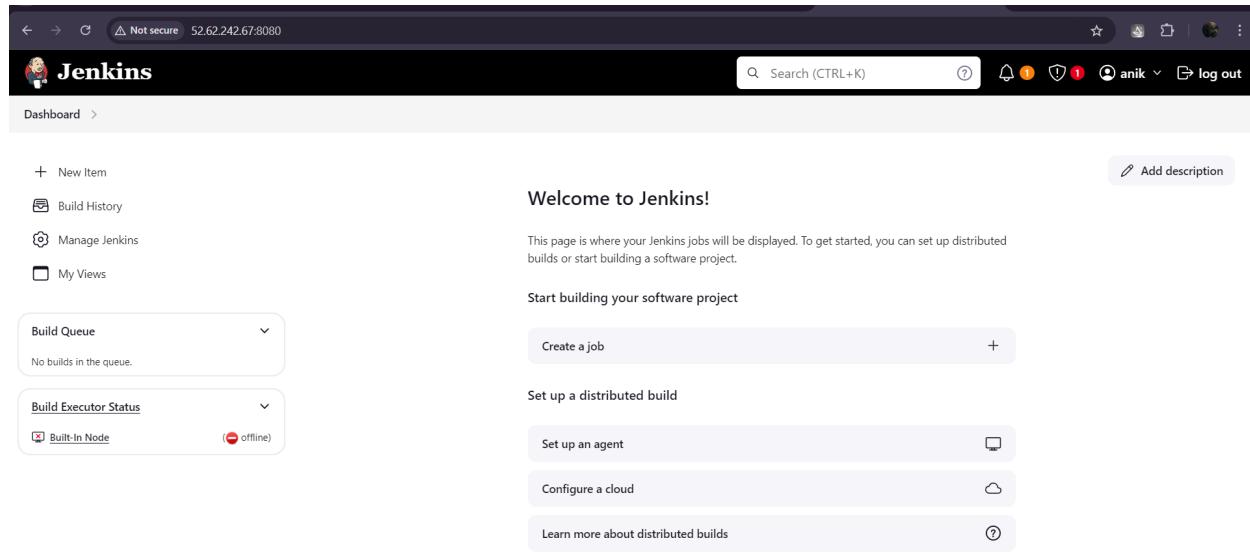
Below this, there is a section for "Environment configuration: Amazon EC2 Auto Scaling groups" which is currently empty.

Step 8 - Create an S3 bucket.



The screenshot shows the AWS S3 console with a success message: "Successfully created bucket 'anikbepari-s3-bucket'". The navigation path is: Amazon S3 > Buckets. The main section displays "General purpose buckets" with one item:

Name	AWS Region	IAM Access Analyzer	Creation date
anikbepari-s3-bucket	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	September 4, 2024, 19:08:39 (UTC+05:30)



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node (offline)

Create a job

Set up a distributed build

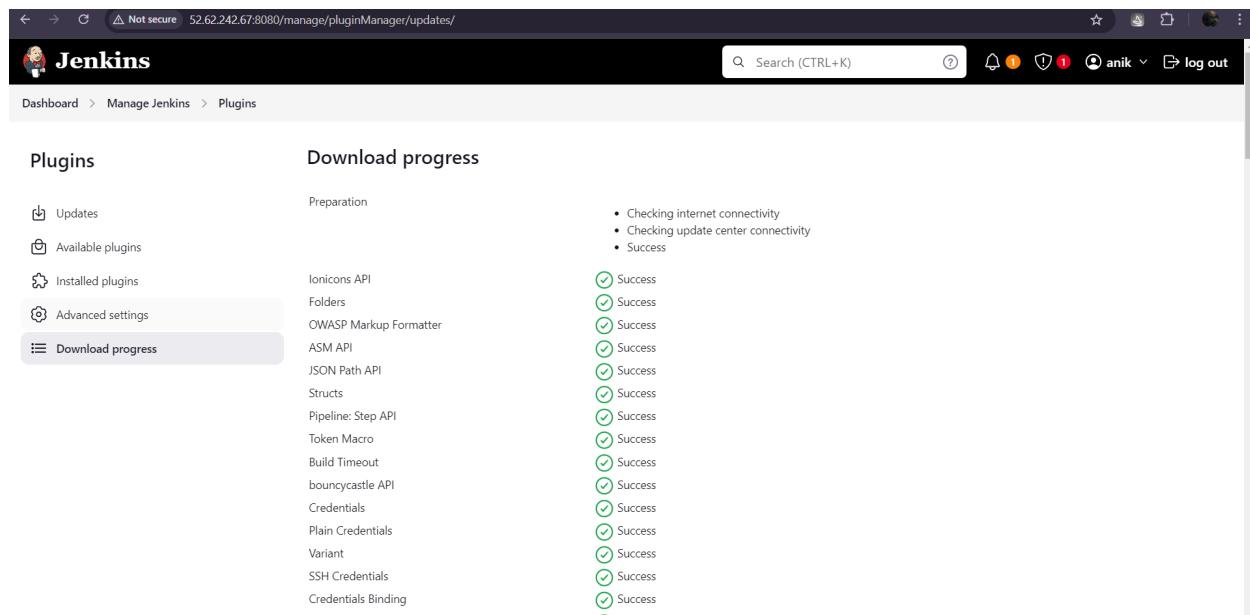
Set up an agent

Configure a cloud

Learn more about distributed builds

REST API Jenkins 2.462.2

- Install AWS Code Deploy plugin in it



Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Plugin	Status
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success

The screenshot shows the Jenkins Dashboard. On the left, there are links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are two expandable sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (Built-In Node, offline). The main area displays a table of builds. The columns are: S (Status), W (Work), Name (Assignment, MyPipeline), Last Success (N/A, 13 min #4), Last Failure (N/A), and Last Duration (N/A, 82 ms). There are 'Edit' and 'Delete' icons for each build entry.

The screenshot shows the 'Assignment' configuration page. The left sidebar has tabs for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is selected. The main area shows the 'General' configuration with an 'Enabled' toggle switch (checked). The 'Description' field is empty. Under 'Source Code Management', the 'GitHub project' option is selected, with 'Project url' set to 'https://github.com/Anikbepari/cicd-assignment.git'. There is an 'Advanced' button and a checkbox for 'This project is parameterized'.

The screenshot shows the 'Assignment' configuration page, specifically the 'Build Triggers' section. The left sidebar has tabs for 'General', 'Source Code Management', 'Build Triggers' (selected), 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'Build Triggers' section contains several checkboxes: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling', and 'Poll SCM'. The 'Poll SCM' checkbox is checked. Below it is a 'Schedule' field with the value '*****'. A note says 'No schedules so will only run due to SCM changes if triggered by a post-commit hook'. There is also a checkbox for 'Ignore post-commit hooks'.

Choose post-build-action as deploy an application AWS CodeDeploy

Configure

Post-build Actions

Deploy an application to AWS CodeDeploy

AWS CodeDeploy Application Name: MyApp

AWS CodeDeploy Deployment Group: MyDeploymentGroup

AWS CodeDeploy Deployment Config:

AWS Region: AP_SOUTHEAST_2

S3 Bucket: anikbepari-s3-bucket

S3 Prefix:

Configure

Post-build Actions

VERSION FILE

Appspec.yml per Deployment Group

Register Revision

Deploy Revision

Use Access/Secret keys

If these keys are left blank, the plugin will attempt to use credentials from the default provider chain. That is: Environment Variables, Java System properties, credentials profile file, and finally, EC2 Instance profile.

AWS Access Key: AKIAQLCZ2OB4KVAAWMR

AWS Secret Key: [REDACTED]

Use temporary credentials

Add post-build action

Save Apply

Step 10 - Build Jenkins Pipeline

Not secure 3.25.64.173:8080/job/newproject/6

Jenkins

Dashboard > newproject > #6

Status #6 (05-Sep-2024, 4:42:49 am)

Started by user **anik**

This run spent:

- 7 ms waiting;
- 1.4 sec build duration;
- 1.4 sec total from scheduled to completion.

git Revision: cdf2e05c7b0b853e2b6e757bd288677fd1c301b3
Repository: <https://github.com/Anikbepari/cicd-assignment.git>
refs/remotes/origin/main

</> No changes.

Changes
Console Output
Edit Build Information
Delete build #6
Timings
Git Build Data
Previous Build

Started 2 min 31 sec ago
Took 1.4 sec

REST API Jenkins 2.462.2

Check console output.

Not secure 3.25.64.173:8080/job/newproject/6/console

Jenkins

Dashboard > newproject > #6 > Console Output

Console Output

Started by user **anik**

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/newproject

The recommended git tool is: NONE

using credential jenkins-assignment

```
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/newproject/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Anikbepari/cicd-assignment.git # timeout=10
Fetching upstream changes from https://github.com/Anikbepari/cicd-assignment.git
> git --version # timeout=10
> git --version # 'git' version 2.40.1
using GIT_SSH to set credentials
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
> git fetch --tags --force --progress -- https://github.com/Anikbepari/cicd-assignment.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision cdf2e05c7b0b853e2b6e757bd288677fd1c301b3 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f cdf2e05c7b0b853e2b6e757bd288677fd1c301b3 # timeout=10
Commit message: "files are added"
> git rev-list --no-walk cdf2e05c7b0b853e2b6e757bd288677fd1c301b3 # timeout=10
Zipping files into /tmp/#6-12398490877262525696.zip
```

Changes
Console Output
Edit Build Information
Delete build #6
Timings
Git Build Data
Previous Build

Download Copy View as plain text

As soon as we will hit the Build Now button the job will run and trigger for code deploy deployment . we can see here it is in progress.

Developer Tools > CodeDeploy > Deployments > d-YTKE8H567

d-YTKE8H567

Deployment status

Installing application on your instances

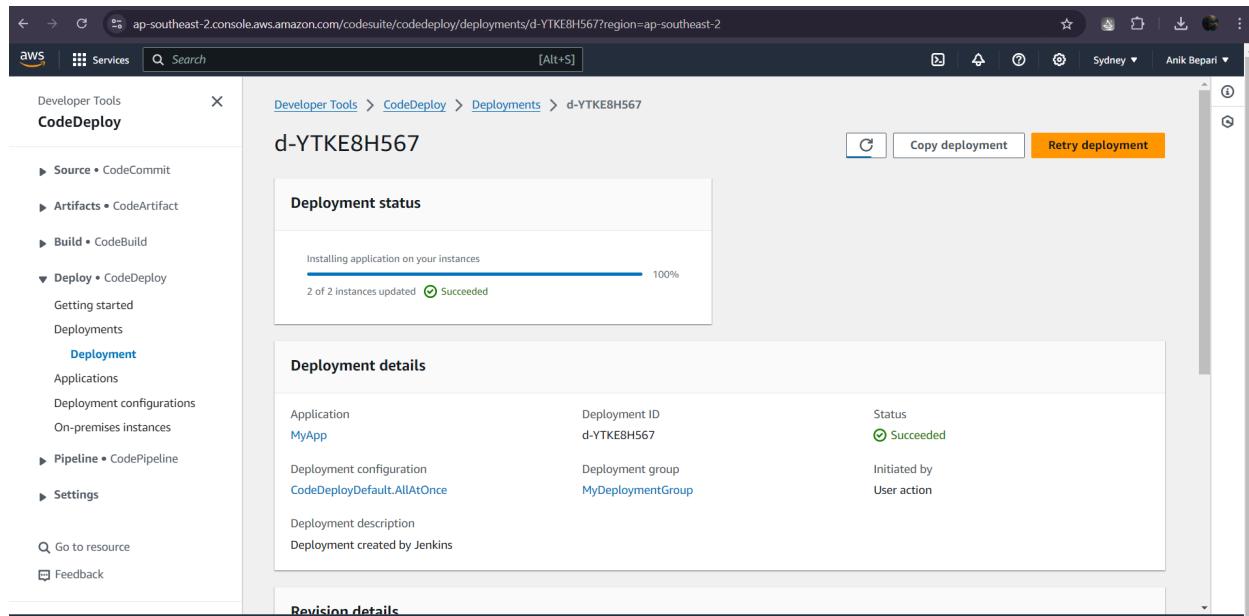
2 of 2 instances updated Succeeded 100%

Deployment details

Application	Deployment ID	Status
MyApp	d-YTKE8H567	Succeeded
Deployment configuration	Deployment group	Initiated by
CodeDeployDefault.AllAtOnce	MyDeploymentGroup	User action
Deployment description	Deployment created by Jenkins	

Revision details

Deployment description
Deployment created by Jenkins



Developer Tools > CodeDeploy > Deployments > d-YTKE8H567

d-YTKE8H567

Deployment details

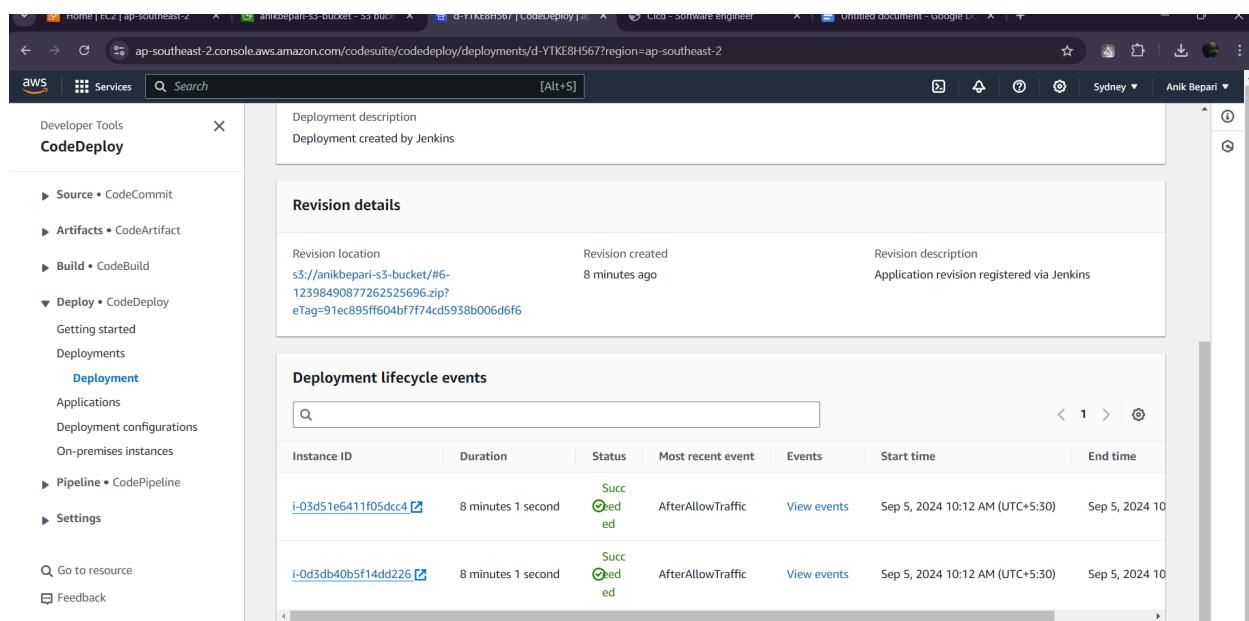
Deployment description	Deployment created by Jenkins
------------------------	-------------------------------

Revision details

Revision location
s3://anikbepari-s3-bucket/#6-12398490877262525696.zip?
eTag=91ec895f604bf7f74cd5938b006d6f6

Deployment lifecycle events

Instance ID	Duration	Status	Most recent event	Events	Start time	End time
i-03d51e6411f05dcc4	8 minutes 1 second	Succeeded	AfterAllowTraffic	View events	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10
i-0d5db40b5f14dd226	8 minutes 1 second	Succeeded	AfterAllowTraffic	View events	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10



Revision details

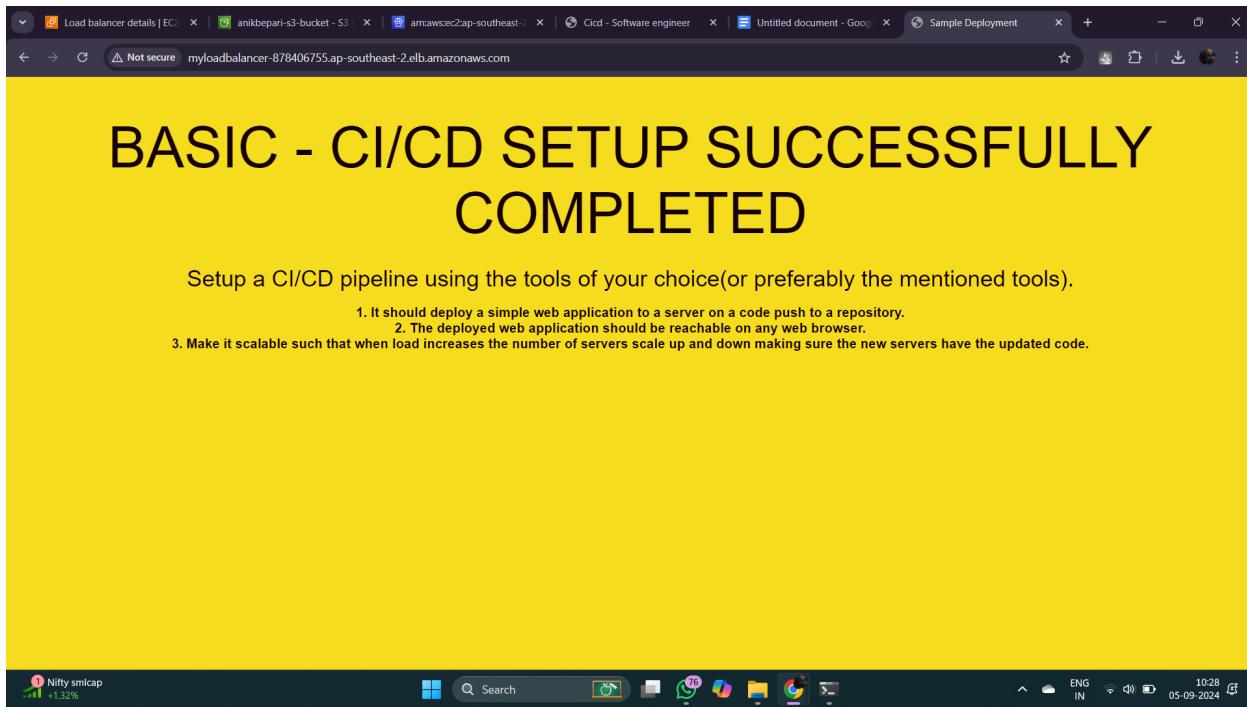
Event	Duration	Status	Error code	Start time	End time
BeforeBlockTraffic	less than one second	✓ Succeeded	-	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10:12 AM (UTC+5:30)
BlockTraffic	5 minutes 11 seconds	✓ Succeeded	-	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
AfterBlockTraffic	less than one second	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
ApplicationStop	less than one second	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
DownloadBundle	3 seconds	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
BeforeInstall	3 seconds	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
Install	less than one second	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
AfterInstall	less than one second	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
ApplicationStart	less than one second	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
ValidateService	less than one second	✓ Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)

Step 11 - Go to AWS Load balancer portal and go to that particular ELB you have Created and copy the DNS and paste it on browser , your application will be running .

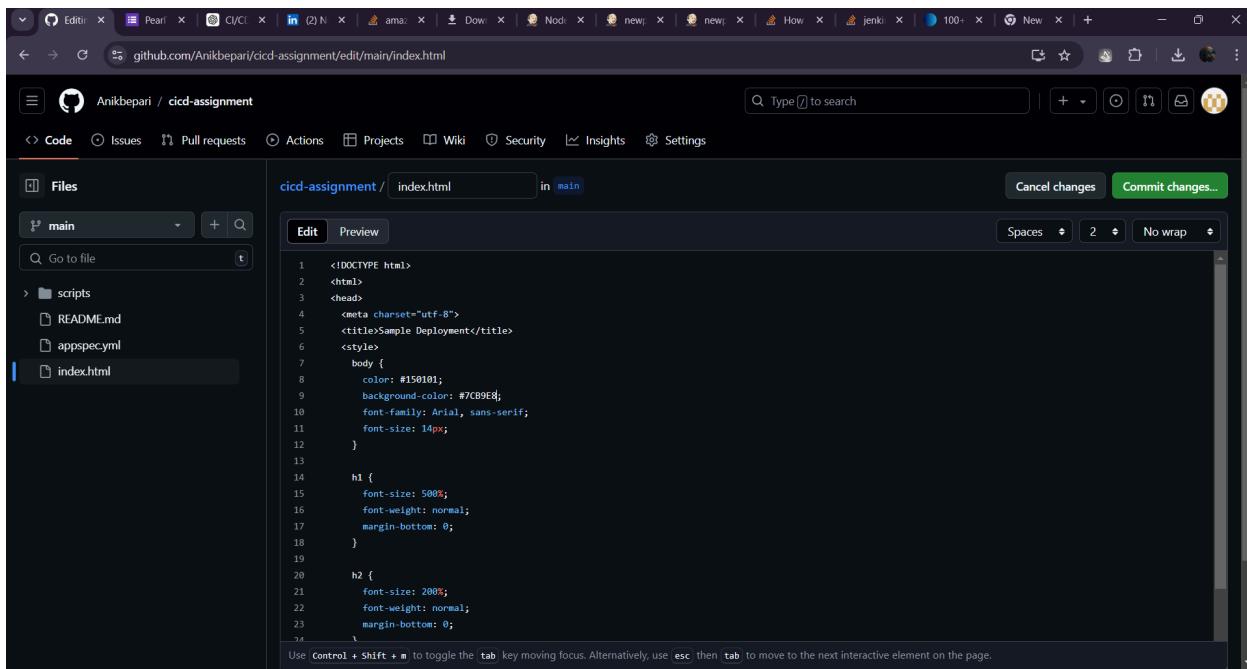
MyLoadBalancer

Details

Load balancer type	Status	VPC	Load balancer IP address type
Application	✓ Active	vpc-0b1a7f9a57e1eb3d0	IPv4
Scheme	Hosted zone	Availability Zones	Date created
Internet-facing	Z1GM3OXH4ZPM65	subnet-031785e35b4f73a03 ap-southeast-2a (apse2-az1)	September 4, 2024, 16:10 (UTC+05:30)
		subnet-071f839190510c11 ap-southeast-2b (apse2-az3)	
		subnet-0ac10b8767ec16767 ap-southeast-2c (apse2-az2)	
Load balancer ARN	DNS name		
arn:aws:elasticloadbalancing:ap-southeast-2:023810699384:loadbalancer/app/MyLoadBalancer/7cf9cccc6cc99ca7	MyLoadBalancer-878406755.ap-southeast-2.elb.amazonaws.com (A Record)		



We can check that as soon as we run the jenkins job it run the deployment the data will get stored in s3 bucket.



← → G Not secure 3.25.64.173:8080/job/newproject/7/console

Dashboard > newproject > #7 > Console Output

Edit Build Information
Delete build '#7'
Polling Log
Timings
Git Build Data
Previous Build

```

Building in workspace /var/lib/jenkins/workspace/newproject
The recommended git tool is: NONE
using credential jenkins-assignment
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/newproject/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Anikbepari/cicd-assignment.git # timeout=10
Fetching upstream changes from https://github.com/Anikbepari/cicd-assignment.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
using GIT_SSH to set credentials
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
> git fetch -tags --force --progress -- https://github.com/Anikbepari/cicd-assignment.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^(commit) # timeout=10
Checking out Revision 4ac5b1f417c1670152923a9a968c5f8e6c3bf6f (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4ac5b1f417c1670152923a9a968c5f8e6c3bf6f # timeout=10
Commit message: "Update index.html"
> git rev-list --no-walk cdfe2e05c70b0853e2bde757bd288677fd1c301b3 # timeout=10
Zipping files into /tmp/#7-5757832227779262015.zip
Uploading zip to s3://anikbepari-s3-bucket/#7-5757832227779262015.zip
Registering revision for application 'MyApp'
Creating deployment with revision at {RevisionType: S3, S3Location: {Bucket: anikbepari-s3-bucket, Key: #7-5757832227779262015.zip, BundleType: zip, ETag: 8fc0ccc6819732608aa5ba4ae1c11db12},}
Finished: SUCCESS

```

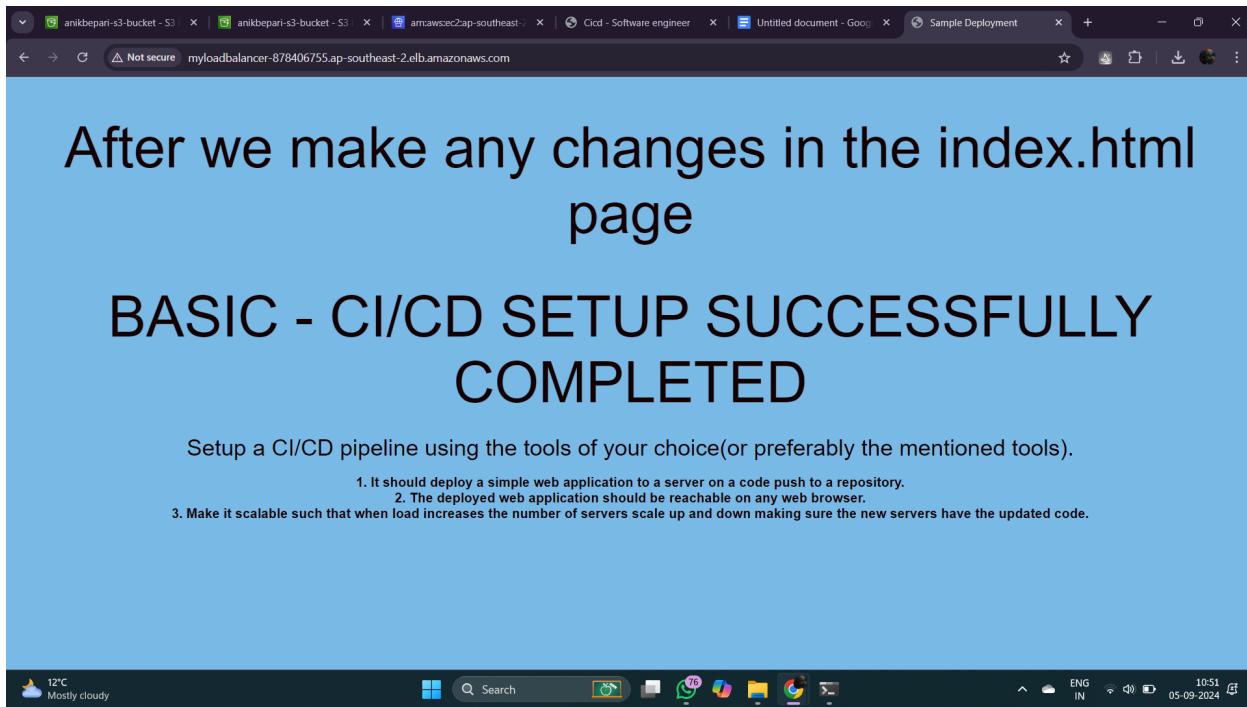
Commit message: "Update index.html"

REST API | Jenkins 2.462.2

Developer Tools **CodeDeploy** X

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
 - Getting started
 - Deployments
 - Deployment**
 - Applications
 - Deployment configurations
 - On-premises instances
- Pipeline • CodePipeline
- Settings
- Go to resource

Deployment configuration CodeDeployDefault.AllAtOnce	Deployment group MyDeploymentGroup	Initiated by User action																														
Deployment description Deployment created by Jenkins																																
Revision details																																
Revision location s3://anikbepari-s3-bucket/#7-12398490877262525696.zip?	Revision created 38 minutes ago	Revision description Application revision registered via Jenkins																														
<table border="1"> <thead> <tr> <th>Event</th> <th>Duration</th> <th>Status</th> <th>Error code</th> <th>Start time</th> <th>End time</th> </tr> </thead> <tbody> <tr> <td>BeforeBlockTraffic</td> <td>less than one second</td> <td>Succeeded</td> <td>-</td> <td>Sep 5, 2024 10:12 AM (UTC+5:30)</td> <td>Sep 5, 2024 10:12 AM (UTC+5:30)</td> </tr> <tr> <td>BlockTraffic</td> <td>5 minutes 11 seconds</td> <td>Succeeded</td> <td>-</td> <td>Sep 5, 2024 10:12 AM (UTC+5:30)</td> <td>Sep 5, 2024 10:18 AM (UTC+5:30)</td> </tr> <tr> <td>AfterBlockTraffic</td> <td>less than one second</td> <td>Succeeded</td> <td>-</td> <td>Sep 5, 2024 10:18 AM (UTC+5:30)</td> <td>Sep 5, 2024 10:18 AM (UTC+5:30)</td> </tr> <tr> <td>ApplicationStop</td> <td>less than one second</td> <td>Succeeded</td> <td>-</td> <td>Sep 5, 2024 10:18 AM (UTC+5:30)</td> <td>Sep 5, 2024 10:18 AM (UTC+5:30)</td> </tr> </tbody> </table>			Event	Duration	Status	Error code	Start time	End time	BeforeBlockTraffic	less than one second	Succeeded	-	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10:12 AM (UTC+5:30)	BlockTraffic	5 minutes 11 seconds	Succeeded	-	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)	AfterBlockTraffic	less than one second	Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)	ApplicationStop	less than one second	Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)
Event	Duration	Status	Error code	Start time	End time																											
BeforeBlockTraffic	less than one second	Succeeded	-	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10:12 AM (UTC+5:30)																											
BlockTraffic	5 minutes 11 seconds	Succeeded	-	Sep 5, 2024 10:12 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)																											
AfterBlockTraffic	less than one second	Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)																											
ApplicationStop	less than one second	Succeeded	-	Sep 5, 2024 10:18 AM (UTC+5:30)	Sep 5, 2024 10:18 AM (UTC+5:30)																											



And here is the Our Final Output . We can see that it shows the recent changed done by us in the html file. Since we have used Auto Scaling Group and using ELB and its url to access the application , we can confirm that it scalable such that when load increases the number of servers scale up and down making sure the new servers have the updated code.

Github repo used - <https://github.com/Anikbepari/cicd-assignment>

Steps To complete the assignment are following :

1.Createated IAM Roles

Created a role for EC2 instances with necessary policies.

Created a role for CodeDeploy with its specific policy.

2.Launched EC2 Instances

Launched instances with the required AMI and instance type.

Attached the appropriate IAM role and configure user data for setup.

Set up security groups and launch the instances.

3.Created an AMI for Auto Scaling

Create an AMI from one of the configured EC2 instances for use in Auto Scaling.

4. Made Application Load Balancer (ALB)

Created an Application Load Balancer, configure listeners, and set up a target group.

Registered the EC2 instances with the target group.

5. Created Launch Configuration and Auto Scaling Group

Created a launch configuration using the AMI and instance settings.

Created an Auto Scaling Group with desired capacity and attach the Load Balancer.

6. Set Up Jenkins

Launched an EC2 instance for Jenkins and install Jenkins.

Accessed Jenkins through its public IP.

7. Set Up CodeDeploy

Created a CodeDeploy application and deployment group.

Configured the deployment group with the Auto Scaling Group and Load Balancer.

8. Create an S3 Bucket

Created an S3 bucket for storing application files.

9. Configured Jenkins Pipeline

Install the AWS CodeDeploy plugin in Jenkins.

Create a Jenkins pipeline job with Git as the source and configure post-build actions for CodeDeploy.

10.Built the Jenkins Pipeline

Trigger a build in Jenkins to deploy the application.

11.Accessed the Web Application

Accessed the application via the Load Balancer's DNS name.

Problems that i have encountered while completing the assignment are how did i resolved those are following :

1) I was unable to connect via SSH to EC2 instance for Jenkins due to an incorrect key pair.

Solution: Ensured the correct key pair was selected during EC2 instance launch and verified the private key file during the SSH connection.

2) I have incorrectly selected the Auto Scaling Group during CodeDeploy deployment group creation.

Solution: Reviewed and selected the correct Auto Scaling Group from the dropdown menu when setting up the deployment group.

3)The S3 bucket name was invalid when configuring Jenkins to upload files to S3.

Solution: Renamed the bucket to adhere to S3 naming conventions and updated the pipeline configuration in Jenkins.

4) Build failure in Jenkins due to missing AWS credentials for CodeDeploy.

Solution: Configured AWS Access Key and Secret Key in Jenkins under "Manage Credentials" to ensure CodeDeploy can access AWS resources.

5) Application was not accessible through the Load Balancer.

Solution : Realized the security group didn't allow HTTP (port 80) traffic. Added the rule to the security group, enabling access to the application via the Load Balancer DNS.

Problem Statement

Write a shell script to create a file in multiple folder which contains a folder name oriserve only.
Explanation is below.

```
$ Shell_script.sh X
$ Shell_script.sh
1  #!/bin/bash
2  base_dir="projects"
3  if [ ! -d "$base_dir" ]; then
4      echo "Directory '$base_dir' does not exist."
5      exit 1
6  fi
7  for project in "$base_dir"/*; do
8      if [ -d "$project" ]; then
9          oriserve_dir="$project/oriserve"
10
11         if [ -d "$oriserve_dir" ]; then
12             touch "$oriserve_dir/test.txt"
13             echo "Created test.txt in $oriserve_dir"
14         fi
15     fi
16 done
17
18 echo "Script execution completed."
19 |
```