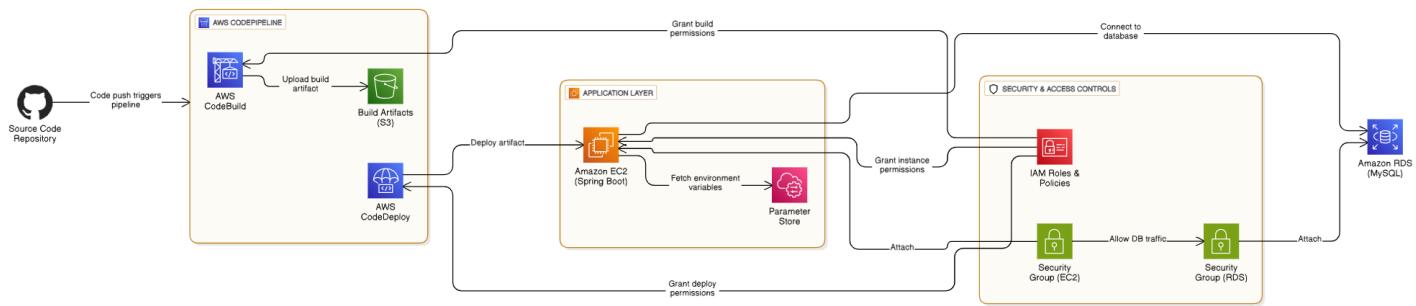


# End-to-End CI/CD Deployment of Java Spring Boot Application on AWS using CodePipeline, CodeBuild, CodeDeploy, EC2, RDS, S3, IAM, and Systems Manager Parameter Store



eraser

Dhruv N Patel (dnp176)  
<https://emergencecloudsolutions.cloud/>

## ❖ Step-by-Step: Create Security Group

- Go to **AWS Console**.
- Search for **EC2** service and open it.
- From the left side menu, click on **Security Groups** under **Network & Security**.
- Click on **Create security group**.
- In **Security group name**, type a suitable name (example: AllTrafficSupport).
- Add a description if you want.
- In **Inbound rules**, click **Add rule**.
  - Set **Type** as All Traffic.
  - Set **Source** as Anywhere-IPv4 (0.0.0.0/0).
- Leave **Outbound rules** as default (Allow all traffic).
- Click **Create security group**.

The screenshot shows the AWS EC2 Security Groups page. A new security group named "sg-0f53481d882b72879 - AllTrafficSupport" has been created. The "Inbound rules" section is highlighted with a red box, displaying seven rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0b285364270e8adb0	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-03d464f3046bbff18	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0
-	sgr-05db1e4b838adb44	IPv4	All traffic	All	All	0.0.0.0/0
-	sgr-07ed05a0883085b29	IPv4	All TCP	TCP	0 - 65535	0.0.0.0/0
-	sgr-037ed3b457b2c31a4	IPv4	Custom TCP	TCP	8080	0.0.0.0/0
-	sgr-0e50892e742dc5c41	IPv4	HTTPS	TCP	443	0.0.0.0/0
-	sgr-0bc6a3fc7b2601445	IPv4	HTTP	TCP	80	0.0.0.0/0

## Step-by-Step: Create RDS (MySQL)

- Go to **AWS Console**.
- Search for **RDS** service and open it.
- From the left side menu, click on **Databases**.
- Click on **Create database**.
- In **Choose a database creation method**, select **Standard Create**.
- In **Engine options**, select **MySQL**.
- In **Templates**, select **Free tier**.
- Set a **DB instance identifier** (example: ecomrdstest).
- Set **Master username** (example: admin) and **Master password** (example: Admin\$123).
- In **DB instance size**, select db.t3.micro (or t2.micro if available under free tier).
- In **Connectivity** section:
  - Select your existing **VPC**.
  - In **Additional connectivity configuration**, choose your **security group** (example: AllTrafficSupport).
- Leave other settings as default.
- Click on **Create database**.

AWS CloudShell Feedback

Search [Alt+S]

Account ID: 7466-6922-2741 Admin

Aura and RDS Create database

**Create database** [Info](#)

**Choose a database creation method**

- Standard create You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- Easy create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

**Engine type** [Info](#)

<input type="radio"/> Aurora (MySQL Compatible)	<input type="radio"/> Aurora (PostgreSQL Compatible)	<input checked="" type="radio"/> MySQL
<input type="radio"/> PostgreSQL	<input type="radio"/> MariaDB	<input type="radio"/> Oracle
<input type="radio"/> Microsoft SQL Server	<input type="radio"/> IBM Db2	

**Edition**

MySQL Community

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

AWS CloudShell Feedback

Search [Alt+S]

Account ID: 7466-6922-2741 Admin

Aura and RDS Create database

**Edition**

MySQL Community

**Engine version** [Info](#)

View the engine versions that support the following database features.

**Hide filters**

- Show only versions that support the Multi-AZ DB cluster [Info](#) Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.
- Show only versions that support the Amazon RDS Optimized Writes [Info](#) Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

**Engine version**

MySQL 8.0.41

Enable RDS Extended Support [Info](#) Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

**Templates**

Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.	<input checked="" type="radio"/> Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. <a href="#">Info</a>
---	---	---

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

**Settings**

**DB instance identifier** Info  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.  
`ecommerceproject`  
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**Credentials Settings**

**Master username** Info  
Type a login ID for the master user of your DB instance.  
`admin`  
1 to 15 alphanumeric characters. The first character must be a letter.

**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.

**Managed in AWS Secrets Manager - most secure**  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

**Self managed**  
Create your own password or have RDS create a password that you manage.

**Auto generate password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** Info  
`Admin$123`  
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ \* @

**Confirm master password** Info  
`Admin$123`

**Instance configuration**

**DB instance class** Info  
 **db.t4g.micro**  
2 vCPUs | 1 GB RAM | Network: Up to 2,085 Mbps

**Storage**

**Storage type** Info  
Amazon EBS SSD (io2) storage volumes are now available.  
 **General Purpose SSD (gp2)**  
Baseline performance determined by volume size

**Allocated storage** Info  
`10` GB  
Allocated storage value must be 20 GB to 6,144 GB.

**Additional storage configuration**

**Storage autoscaling** Info  
Helps you automatically scale your database's storage based on your application's needs.  
 **Enable storage autoscaling**  
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

**Connectivity** Info

**Compute resource**  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

**Don't connect to an EC2 compute resource**  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

**Connect to an EC2 compute resource**  
Set up a connection to an EC2 compute resource for this database.

**Network type**  
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

**IPv4**  
Your resources can communicate only over the IPv4 addressing protocol.

**Dual-stack mode**  
Your resources can communicate over IPv4, IPv6, or both.

**Virtual private cloud (VPC)** Info  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

**Default VPC** `(rpo-011f0092f2d1e43e0)`  
2 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

**After a database is created, you can't change its VPC.**

**DB subnet group** Info  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

`db.grp`  
2 Subnets, 2 Availability Zones

**Public access** Info  
 **Yes**  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

The screenshot shows the 'Create database' configuration page in the AWS RDS console. The 'Additional configuration' section is expanded, highlighting several key settings:

- VPC security group (firewall)**: Shows options to 'Choose existing' or 'Create new' VPC security groups.
- Existing VPC security groups**: Displays 'Choose one or more options' and lists 'default' and 'AllTrafficSupport'.
- Availability Zone**: Set to 'No preference'.
- RDS Proxy**: Includes a checkbox for 'Create an RDS Proxy'.
- Certificate authority (optional)**: Shows a dropdown menu with 'rds-ca-20248-1 (default)' selected.
- Database port**: Set to '3306'.
- Backup**: Contains a checkbox for 'Enable automated backups'.
- Encryption**: Contains a checkbox for 'Enable encryption'.
- Maintenance**: Contains a checkbox for 'Enable auto minor version upgrade'.

## 📌 Step-by-Step: Create an IAM Policy

- Go to **AWS Console**.
- Search for **IAM** service and open it.
- From the left side menu, click on **Policies**.
- Click on **Create policy**.
- Select the **JSON** tab.

- Paste the required policy JSON code (as per your requirement, for example to access Parameter Store).
- Click **Next** to review.
- Give a suitable **Policy name** (example: EC2ParameterStoreReadAccess).
- Click **Create policy**

Paste this code there (replace **YOUR\_ACCOUNT\_ID** with your AWS account ID):

**JSON:**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameter",
        "ssm:GetParameters"
      ],
      "Resource": "arn:aws:ssm:ap-south-1:YOUR_ACCOUNT_ID:parameter/ecomapp/*"
    }
  ]
}
```

**EC2ParameterStoreReadAccess**

**Policy details**

Type: Customer managed  
Creation time: June 14, 2025, 11:16 (UTC+05:30)  
Edited time: June 14, 2025, 11:16 (UTC+05:30)  
ARN: arn:aws:iam:746669222741:policy/EC2ParameterStoreReadAccess

**Permissions**   **Entities attached**   **Tags**   **Policy versions**   **Last Accessed**

**Permissions defined in this policy**

```

1 - {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "ssm:GetParameter",
8         "ssm:GetParameters"
9       ],
10      "Resource": "arn:aws:ssm:ap-south-1:746669222741:parameter/econapp/*"
11    }
12  ]
13 }
```

## 📌 Step-by-Step: Create IAM Role and Attach Policies [Go to AWS Console.](#)

1. Search for **IAM** service and open it.
2. From the left side menu, click on **Roles**.
3. Click on **Create role**.
4. In **Trusted entity type**, select **AWS service**.
5. In **Use case**, select **EC2**.
6. Click on **Next**.
7. Now, in the policies list, select these policies:
  - AmazonEC2FullAccess (AWS managed)
  - AmazonS3FullAccess (AWS managed)
  - AWSCodeDeployFullAccess (AWS managed)
  - EC2ParameterStoreReadAccess (Customer managed)
8. Click on **Next**.
9. Give a role name like **ec2-code-deploy**.

## 10. Click on **Create role**.

The screenshot shows the AWS IAM Roles page. A red box highlights the 'ec2-code-deploy' role name. The 'Permissions' tab is selected, showing a table of managed policies attached to the role. The table includes:

Policy name	Type	Attached entities
AmazonEC2FullAccess	AWS managed	2
AmazonS3FullAccess	AWS managed	3
AWSCodeDeployFullAccess	AWS managed	2
EC2ParameterStoreReadAccess	Customer managed	2

## 📌 Step-by-Step: Create EC2 Instance with Required Settings

1. Go to **AWS Console**.
2. Search for **EC2** service and open it.
3. Click on **Instances** from the left menu.
4. Click on **Launch Instance**.
5. In **Name**, type E-Commerce-Server.
6. In **Instance type**, select t2.micro.
7. In **Key pair (login)** section:
  - If you already have a key pair, select **Choose existing key pair**.
  - Otherwise, click on **Create new key pair**, give it a name, download the .pem file, and then select it.
8. In **Network settings**:
  - Select **Existing security group**.
  - Choose **AllTrafficSupport** from the list.
9. In **Advanced details**, under **IAM instance profile**, select **ec2-code-deploy**.

10. Leave other settings as default.

11. Click on **Launch instance**.

## 📌 Install CodeDeploy Agent on EC2

**Paste below script in EC2 machine**

```
#!/bin/bash

# Update packages
sudo apt-get update -y

# Install dependencies
sudo apt-get install -y ruby wget

# Download the CodeDeploy agent installer (replace region if
needed)

cd /home/ubuntu

wget https://aws-codedeploy-us-east-1.s3.us-east-
1.amazonaws.com/latest/install

# Make installer executable
chmod +x ./install

# Run the installer
sudo ./install auto
```

```

# Start the CodeDeploy agent service
sudo systemctl start codedeploy-agent
# Enable the service to start on boot
sudo systemctl enable codedeploy-agent
# Check the agent status
sudo systemctl status codedeploy-agent

```

## 📌 Step-by-Step: Create Environment Variables in AWS Systems Manager Parameter Store

- Go to **AWS Console**.
- Search for **Systems Manager** service and open it.
- From the left side menu, click on **Parameter Store** under **Application Management**.
- Click on **Create parameter**.
- In **Name**, type the parameter name (example: /ecomapp/DOMAIN).
- In **Tier**, select **Standard**.
- In **Type**, select **String**.
- In **Value**, type the required value (like EC2 public IP, RDS endpoint, username, password etc.).
- Click **Create parameter**.
- Repeat the same process for all required parameters, for example:

Name	Value
/ecomapp/DOMAIN	http://<EC2-Public-IP>:9090

<b>/ecomapp/MYSQLHOST</b>	<RDS Endpoint>
<b>/ecomapp/MYSQL_DATABASE</b>	ecomrdstest
<b>/ecomapp/MYSQLUSER</b>	admin
<b>/ecomapp/MYSQLPASSWORD</b>	'Admin\$123'

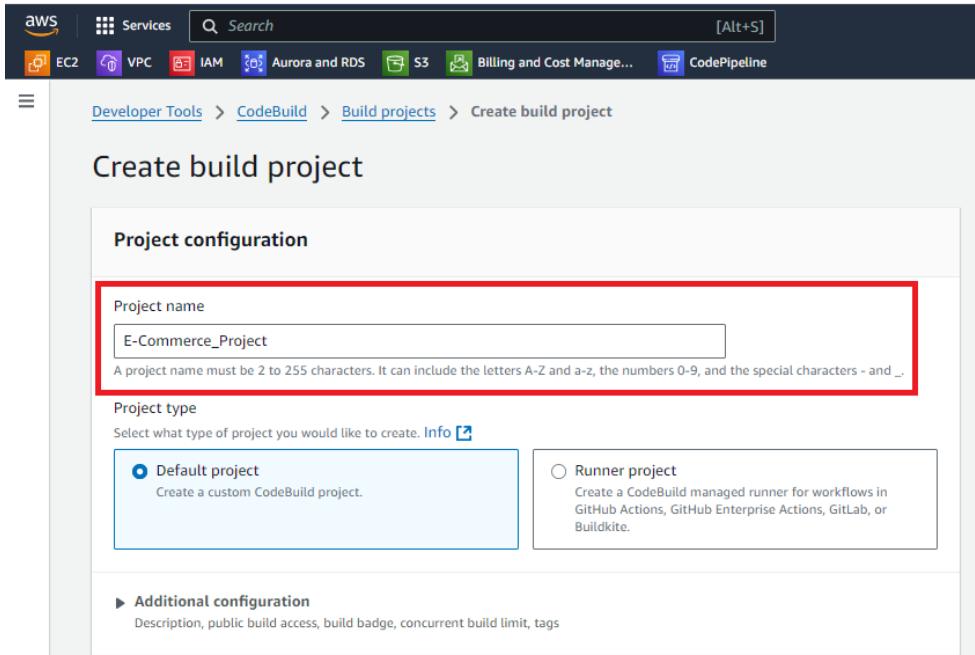
### Step-by-Step: Create S3 Bucket

- Go to **AWS Console**.
- Search for **S3** service and open it.
- Click on **Create bucket**.
- In **Bucket name**, type a name (example: ecom-app-bucket).
- In **Region**, select Asia Pacific (Mumbai) ap-south-1 or your preferred region.
- Leave other settings as default.
- Scroll down and click **Create bucket**.

### 👉 Step-by-Step: Create CodeBuild Project

- Go to **AWS Console**.
- Search for **CodeBuild** service and open it.
- Click on **Build projects** from the left side menu.
- Click on **Create build project**.
- In **Project name**, enter a suitable name (example: EcomAppBuildProject).
- In **Source provider**, select your source (like **GitHub**, **CodeCommit**, or **S3**).
- Configure **Environment** settings:
  - Select **Managed image**.
  - Choose your runtime (like **Ubuntu**, standard image version).

- In **Service role**, either select an existing role or create a new one with required permissions.
- Optionally, provide **buildspec.yml** configuration or let CodeBuild pick it from your source repo.
- Click **Create build project**



▼ Source Add source

**Source 1 - Primary**

Source provider GitHub

Credential  
Your account is successfully connected through PAT using CodeBuild managed token. [Manage account credentials](#).

Use override credentials for this project only

Repository  
 Repository in my GitHub account  Public repository  GitHub scoped webhook

https://github.com/dnp176/E-Commerce\_REST-API\_SwaggerProject\_LIVE X G

Source version - *optional* [Info](#)  
Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

► Additional configuration  
Git clone depth, Git submodules, Build status config

**▼ Environment**

Provisioning model [Info](#)

<input checked="" type="radio"/> On-demand Automatically provision build infrastructure in response to new builds.	<input type="radio"/> Reserved capacity Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.
---	--

Environment image

<input checked="" type="radio"/> Managed image Use an image managed by AWS CodeBuild	<input type="radio"/> Custom image Specify a Docker image
---	--

Compute

<input checked="" type="radio"/> EC2 Optimized for flexibility during action runs	<input type="radio"/> Lambda Optimized for speed and minimizes the start up time of workflow actions
--	---

Running mode

<input checked="" type="radio"/> Container Running on Docker container	<input type="radio"/> Instance Running on EC2 instance directly
---	--

Operating system

Ubuntu	▼
--------	---

Runtime(s)

Standard	▼
----------	---

Image

aws/codebuild/standard:7.0	▼
----------------------------	---

Image version

Always use the latest image for this runtime version	▼
--	---

► Additional configuration  
Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, registry credential

**▼ Buildspec**

Build specifications

<input type="radio"/> Insert build commands Store build commands as build project configuration	<input checked="" type="radio"/> Use a buildspec file Store build commands in a YAML-formatted buildspec file
--	--

Buildspec name - *optional*  
By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

buildspec.yml
---------------

▼ Artifacts
Add artifact

---

**Artifact 1 - Primary**

Type

Amazon S3

You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.

Bucket name

Name

The name of the folder or compressed file in the bucket that will contain your output artifacts. Use Artifacts packaging under Additional configuration to choose whether to use a folder or compressed file. If the name is not provided, defaults to project name.

Enable semantic versioning  
Use the artifact name specified in the buildspec file

**Path - optional**  
The path to the build output ZIP file or folder.

Example: MyPath/MyArtifact.zip.

Namespace type - optional

None

Choose Build ID to insert the build ID into the path to the build output ZIP file or folder, e.g. MyPath/MyBuildID/MyArtifact.zip. Otherwise, choose None.

Artifacts packaging

None  
The artifact files will be uploaded to the bucket.

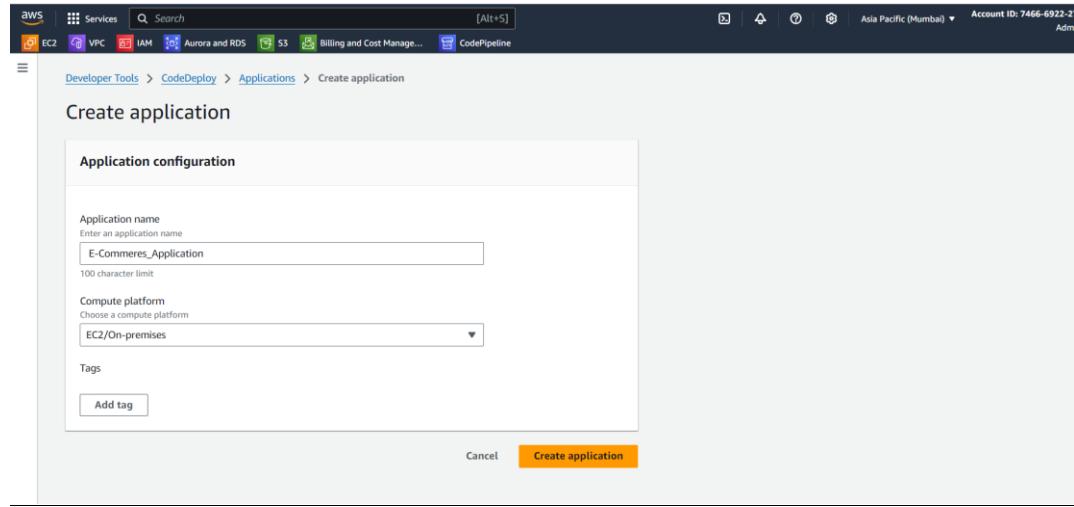
Zip  
AWS CodeBuild will upload artifacts into a compressed file that is put into the specified bucket.

Disable artifact encryption  
Disable encryption if using the artifact to publish a static website or sharing content with others

► Additional configuration  
Cache, encryption key

## Step-by-Step: Create CodeDeploy Application and Deployment Group

- Go to **AWS Console**.
- Search for **CodeDeploy** service and open it.
- From the left side menu, click on **Applications**.
- Click on **Create application**.
- In **Application name**, enter a suitable name (example: EcomAppDeploy).
- In **Compute platform**, select **EC2/On-premises**.
- Click **Create application**.
- Now, open the created application and click on **Create deployment group**.
- In **Deployment group name**, enter a suitable name (example: EcomAppDeployGroup).
- In **Service role**, select the IAM role which has CodeDeploy permissions (example: ec2-code-deploy).
- In **Deployment type**, select **In-place deployment**.
- In **Environment configuration**, select **Amazon EC2 instances**.
- Choose your **EC2 instance tag** or **Auto Scaling Group** to target.
- Click **Create deployment group**.



**Edit deployment group**

**Application**  
E-Commerce\_Application  
Compute type  
EC2/On-premises

**Deployment group name**  
Enter a deployment group name  
E-Commerce\_App\_Grp  
100 character limit

**Service role**  
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.  
arn:aws:iam:746669222741:role/code-deploy-service-role

**Deployment type**

**Deployment type**  
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.

**Environment configuration**  
Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 Instances, and on-premises instances to add to this deployment  
 Amazon EC2 Auto Scaling groups  
 Amazon EC2 Instances  
 1 unique matched instance. Click here for details

**Tag group**  
Key Value - optional  
Name E-Commerce-Server Remove tag  
Add tag + Add tag group

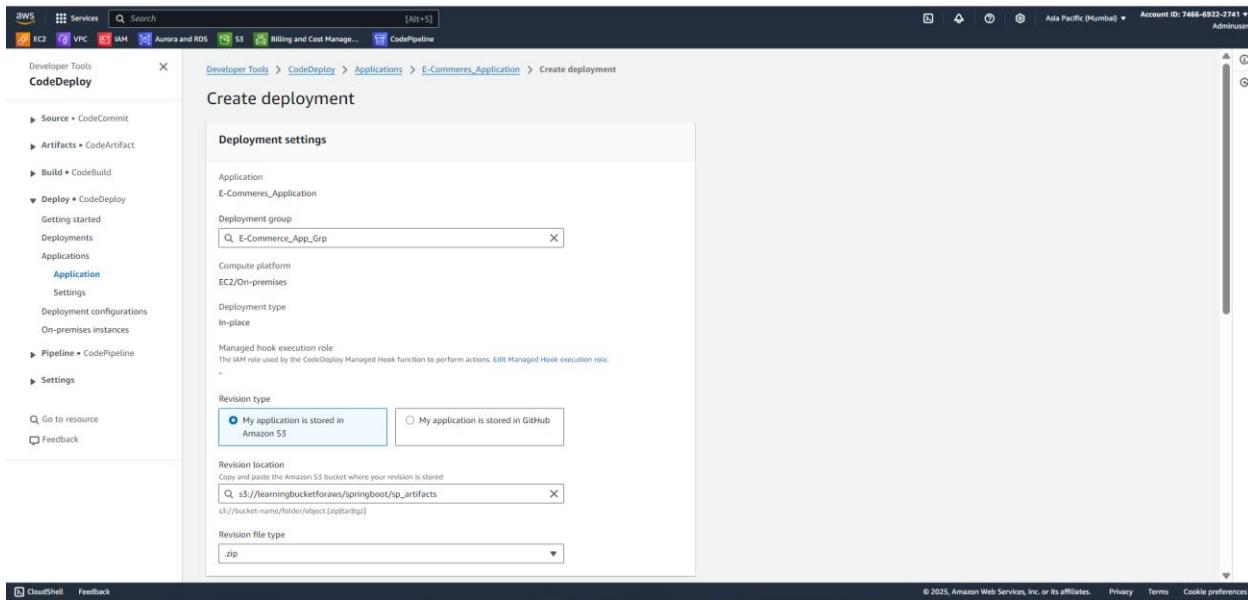
**E-Commerce\_App\_Grp**

**Deployment group details**

Deployment group name	Application name	Compute platform
E-Commerce_App_Grp	E-Commerce_Application	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam:746669222741:role/code-deploy-service-role	CodeDeployDefault.AllAtOnce
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager	

**Environment configuration: Amazon EC2 instances**

Key	Value
Name	E-Commerce-Server



## 📌 Step-by-Step: Create AWS CI/CD Pipeline

- Go to **AWS Console**.
- Search for **CodePipeline** service and open it.
- From the left side menu, click on **Pipelines**.
- Click on **Create pipeline**.
- In **Pipeline name**, enter a suitable name (example: EcomAppPipeline).
- In **Service role**, either create a new role or use an existing one.
- Click **Next**.
- In **Source stage**, select your **Source provider** (like GitHub, S3, or CodeCommit).
- In **Build stage**, select your existing **CodeBuild project**.
- In **Deploy stage**, select your **CodeDeploy application** and **Deployment group**.
- You can skip the **Test stage** if not required.
- Click **Next** and then **Create pipeline**.

The screenshot shows the AWS CodePipeline 'Create new pipeline' wizard at Step 1: 'Choose creation option'. The left sidebar lists steps from 1 to 7. The main area is titled 'Choose creation option' with a sub-section 'Category'. It contains three radio buttons: 'Deployment', 'Continuous Integration', and 'Automation', with 'Build custom pipeline' selected. A 'Next' button is at the bottom right.

The screenshot shows the AWS CodePipeline 'Create new pipeline' wizard at Step 2: 'Choose pipeline settings'. The left sidebar lists steps from 1 to 7. The main area is titled 'Choose pipeline settings' with a sub-section 'Pipeline settings'. It includes fields for 'Pipeline name' (set to 'E-Commeres\_CICD\_Pipeline'), 'Execution mode' (set to 'Queued'), and 'Service role' (set to 'New service role'). A 'Role name' field contains 'AWSCodePipelineServiceRole-ap-south-1-E-Commeres\_CICD\_Pipeline'. A checkbox 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline' is checked. An 'Advanced settings' section is partially visible at the bottom. A 'Next' button is at the bottom right.

Screenshot of the AWS CodePipeline 'Advanced settings' configuration page:

**Artifact store**

- Default location: Create a default S3 bucket in your account.
- Custom location: Choose an existing S3 location from your account in the same region and account as your pipeline.

**Bucket**: learningbucketforaws

**Encryption key**

- Default AWS Managed Key: Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.
- Customer Managed Key: To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.

**Variables**

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

[Add variable](#)

**Cancel** **Previous** **Next**

Screenshot of the AWS CodePipeline 'Create new pipeline' 'Add source stage' configuration page:

**Step 1: Choose creation option**

**Step 2: Choose pipeline settings**

**Step 3: Add source stage** [Info](#) Step 3 of 7

**Source**

**Source provider**: GitHub (via GitHub App)

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

armawocodeconnections:ap-south-1:746669222741:connection/acct [X](#) [C](#) or [Connect to GitHub](#)

**Repository name**: Q\_dnp176/E-Commerce\_REST-API\_SwaggerProject\_LIVE [X](#)

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**: main [X](#)

**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. [Learn more](#)

Enable automatic retry on stage failure

**CloudShell** **Feedback** © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Screenshot of the AWS CodePipeline 'Add build stage' configuration page:

- Step 1: Choose creation option**
- Step 2: Choose pipeline settings**
- Step 3: Add source stage**
- Step 4: Add build stage** (Selected)
- Step 5: Add test stage**
- Step 6: Add deploy stage**
- Step 7: Review**

**Add build stage** Step 4 of 7

**Build - optional**

**Build provider**: AWS CodeBuild

**Project name**: Springboot\_Ans

**Environment variables - optional**: Add environment variable

**Build type**: Single build

**Region**: Asia Pacific (Mumbai)

**Input artifacts**: SourceArtifact

**SourceArtifact**: Defined by: Build

Enable automatic retry on stage failure

## Skip test stage & click next, add deploy stage

Screenshot of the AWS CodePipeline 'Add deploy stage' configuration page:

- Step 1: Choose creation option**
- Step 2: Choose pipeline settings**
- Step 3: Add source stage**
- Step 4: Add build stage**
- Step 5: Add test stage**
- Step 6: Add deploy stage** (Selected)
- Step 7: Review**

**Add deploy stage** Step 6 of 7

**Deploy - optional**

**Deploy provider**: AWS CodeDeploy

**Region**: Asia Pacific (Mumbai)

**Input artifacts**: BuildArtifact

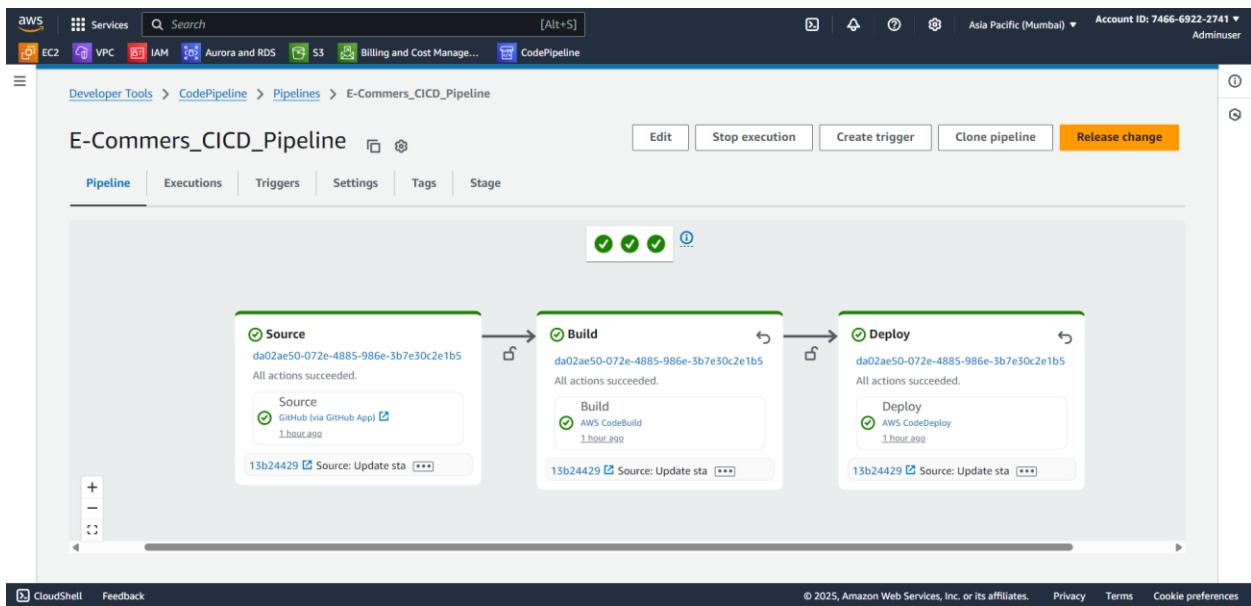
**Application name**: E-Commerce\_Application

**Deployment group**: E-Commerce\_App\_Grp

Configure automatic rollback on stage failure

Enable automatic retry on stage failure

**Next**



## Application Screenshot

### Project use case video

[https://www.linkedin.com/posts/dhruv-n-patel176\\_springbootapi-ecommercedevelopment-javadeveloper-activity-7258722934792351746-HpT7?utm\\_source=share&utm\\_medium=member\\_desktop&rcm=ACoAAChsjTkBF3A5FonzxEYHvAz9c7elr8bc0X8](https://www.linkedin.com/posts/dhruv-n-patel176_springbootapi-ecommercedevelopment-javadeveloper-activity-7258722934792351746-HpT7?utm_source=share&utm_medium=member_desktop&rcm=ACoAAChsjTkBF3A5FonzxEYHvAz9c7elr8bc0X8)

The screenshot shows the Swagger UI for the E-Comers REST APIs Projects. The main page displays the title "E-Comers REST APIs Projects" and a brief description: "This project is represent for real time e-commers projects." Below this, there is a "Contact Demo UserName" button. The "Servers" dropdown is set to "http://52.66.211.57:9090". On the right side, there is an "Authorize" button with a lock icon. The "Authentication Controller" section is expanded, showing four endpoints: "POST /api/auth/register" (Register User), "POST /api/auth/logout" (Logout User), "POST /api/auth/login" (Login User), and "GET /api/auth/getAllUsers" (Get All Users). Each endpoint has a "▼" icon to its right.

Not secure 52.66.211.57:9090/swagger-ui/index.html#/Cart%20or%20Checkout%20Information/getOrderInformation

**GET /api/orderInfo**

**Parameters**

Name	Description
<b>orderId</b> * required	Integer(\$int32) (query)

**Responses**

**Curl**

```
curl -X GET "http://52.66.211.57:9090/api/orderInfo?orderId=37263" \
-H "accept: */*" \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiMzMjQwQS4WV1ubG0vY29tIiJmKQGOTmNTkzLC3pYXQiOjE3NDk40Tg10TN9.-3zvcENCLvpjdEtRHM16sKY1LPvc1JNvLDbfmcckpQI"
```

**Request URL**

<http://52.66.211.57:9090/api/orderInfo?orderId=37263>

**Server response**

**Code** **Details**

**200 Response body**

```
{
  "id": 37263,
  "userId": {
    "id": 1,
    "username": "dip",
    "email": "dip@example.com",
    "role": 1
  },
  "totalAmount": 30387,
  "totalQuantity": 1,
  "createdAt": "2025-06-07T11:52:39.387947Z",
  "orderItems": [
    {
      "id": 1,
      "productImage": "https://via.placeholder.com/150",
      "productName": "Wireless Bluetooth Headphones",
      "productQuantity": 1,
      "productPrice": 2499
    }
  ],
  "id": 3
}
```

Not secure 52.66.211.57:9090/swagger-ui/index.html#/

**Responses**

**Curl**

```
curl -X 'GET' \
http://52.66.211.57:9090/api/products \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJ0ZXN0QGV4YV1wbGUuY29tIiJmKQGOTmNTkzLC3pYXQiOjE3NDk40Tg10TN9.-3zvcENCLvpjdEtRHM16sKY1LPvc1JNvLDbfmcckpQI'
```

**Request URL**

<http://52.66.211.57:9090/api/products>

**Server response**

**Code** **Details**

**200 Response body**

```
[
  {
    "id": 1,
    "productImg": "https://via.placeholder.com/150",
    "productName": "Wireless Bluetooth Headphones",
    "productDescription": "High-quality wireless headphones with noise cancellation and long battery life.",
    "productPrice": 2499,
    "productStock": 400,
    "productCategory": 1,
    "createdDate": "2025-06-07T11:46:29.138549Z"
  },
  {
    "id": 2,
    "productImg": "https://via.placeholder.com/150",
    "productName": "Gaming Mechanical Keyboard",
    "productDescription": "RGB backlit mechanical keyboard with customizable keys and durable design.",
    "productPrice": 3599,
    "productStock": 200,
    "productCategory": 2,
    "createdDate": "2025-06-07T11:46:48.313166Z"
  },
  {
    "id": 3,
    "productImg": "https://via.placeholder.com/150",
    "productName": "Smart LED TV 55-inch",
    "productDescription": "4K Ultra HD Smart TV with built-in AI voice control and HDR support."
  }
]
```