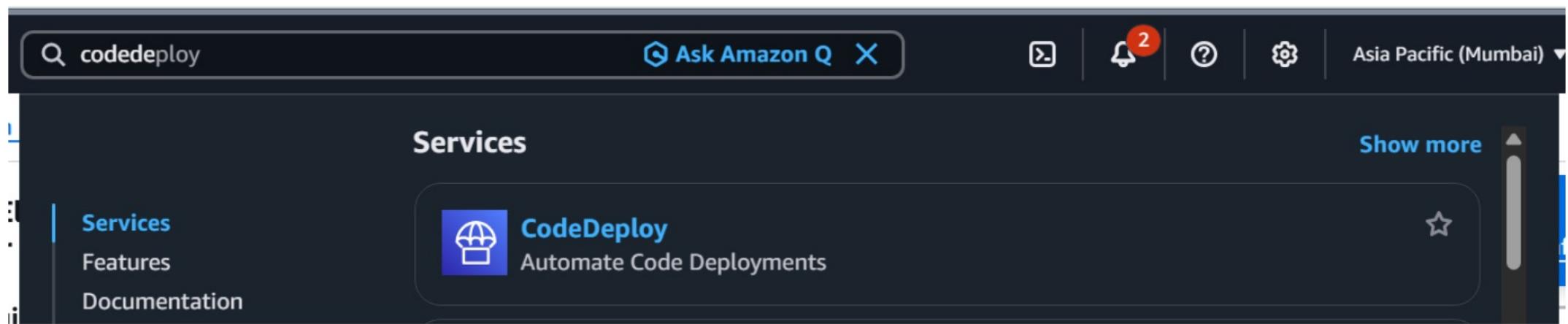


5. Creation of codedeploy application

1. Go to codedeploy page



2. Click on create application button



Create application

3. Name the application and choose ec2 as a compute platform

Application configuration

Application name
Enter an application name
 100 character limit

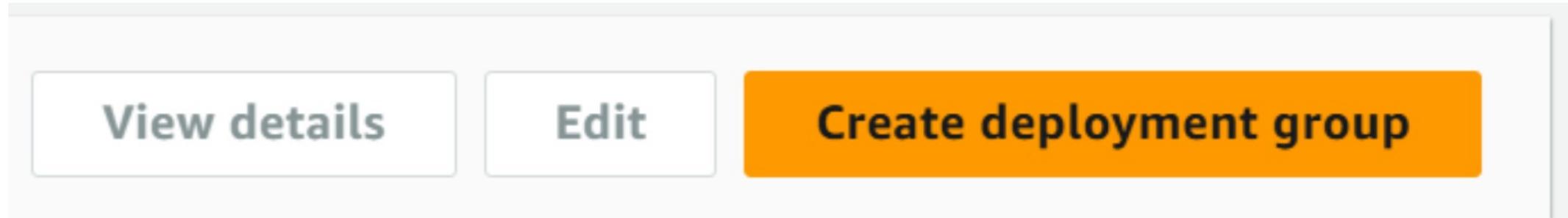
Compute platform
Choose a compute platform
 ▾

4. Now click on create option

Cancel

Create application

5. Now create deployment group



6. Give name to deployment group

Deployment group name

Enter a deployment group name

brain-tasks-eks-dg

100 character limit

7. Now create service role



Delete

Create role

8. Select aws service as trusted identity

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.

9. Choose use case "codedeploy"

Use case

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

Service or use case

CodeDeploy



Choose a use case for the specified service.

Use case

CodeDeploy

Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

10. Add required permissions

Add permissions Info

Permissions policies (1) Info

The type of role that you selected requires the following policy.

Policy name ↗

▲ | Type

▼ |

  [AWSCodeDeployRole](#)

AWS managed

► Set permissions boundary - *optional*

Cancel

Previous

Next

11. Give name to role

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+=,.@-' characters.

Description

Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/[\[]!#\$%^*(0;";'"`

12. Add role to codedeploy

Service role

Enter a service role

Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

X

13. Create appspec file

```
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ nano appsepc.yml
```

14. Enter after install configs in the file

```
GNU nano 7.2                                         appsepc.yml *
```

```
version: 0.0
os: linux

hooks:
  AfterInstall:
    - location: scripts/deploy.sh
      timeout: 300
      runas: root
```

15. Commit and Push the file to github

```
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 678 bytes | 678.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/deepeshchandnani/Brain-Tasks-App.git
  96f71c3..7454a99  main -> main
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ █
```

16. Create mkdir script in repo

```
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ mkdir scripts  
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ █
```

17 create deploy file in the script directory

```
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ nano scripts/deploy.sh
```

18. Enter kubernetes deployment commands in the file

```
GNU nano 7.2                                     scripts/deploy.sh *
```

```
#!/bin/bash
set -e

echo "Updating kubeconfig..."
aws eks update-kubeconfig \
--region ap-south-1 \
--name brain-eks

echo "Deploying to EKS..."
kubectl apply -f k8s/deployment.yaml
kubectl apply -f k8s/service.yaml

echo "Deployment completed"
```

19. Make the file executable

```
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ chmod +x scripts/deploy.sh  
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ █
```

20. Commit and push the file to github

```
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 492 bytes | 492.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/deepeshchandnani/Brain-Tasks-App.git
  7454a99..e58ad4d  main -> main
ubuntu@ip-172-31-3-167:~/Brain-Tasks-App$ □
```

21. Launch an instance

Launch instances



22. Give name to instance

Name and tags [Info](#)

Name

[Add additional tags](#)

23. Add tags in the instance

Manage tags Info

A tag is a custom label that you assign to an AWS resource. You can use tags to help organize and identify your instances.

Key	Value - optional	
<input type="text" value="Name"/> X	<input type="text" value="codedeploy-runner"/> X	Remove
<input type="text" value="Environment"/> X	<input type="text" value="dev"/> X	Remove

[Add new tag](#)

You can add up to 48 more tags.

[Cancel](#) Save

Activate Windows

24. Update packages

```
ubuntu@ip-172-31-10-69:~$ sudo apt update -y
```

25. Install aws codedeploy agent

```
ubuntu@ip-172-31-10-69:~$ sudo apt install ruby wget -y
```

```
untu@ip-172-31-10-69:~$ wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install
```

```
ubuntu@ip-172-31-10-69:~$ chmod +x install  
ubuntu@ip-172-31-10-69:~$ 
```

```
buntu@ip-172-31-10-69:~$ sudo ./install auto  
, [2025-12-18T17:03:42.173185 #2126] INFO -- : Starting Ruby version check.  
, [2025-12-18T17:03:42.173264 #2126] WARN -- : The Ruby version in /usr/bin/ruby3.2 is 3.2.3, . Attempting to install anyway.  
, [2025-12-18T17:03:42.173296 #2126] INFO -- : Starting update check.  
, [2025-12-18T17:03:42.173320 #2126] INFO -- : Attempting to automatically detect supported package manager type for system...  
, [2025-12-18T17:03:42.188651 #2126] WARN -- : apt-get found but no gdebi. Installing gdebi with `apt-get install gdebi-core -y`...
```

26. Start the agent

```
ubuntu@ip-172-31-10-69:~$ sudo systemctl start codedeploy-agent
ubuntu@ip-172-31-10-69:~$ █
ubuntu@ip-172-31-10-69:~$ sudo systemctl status codedeploy-agent
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
    Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
    Active: active (running) since Thu 2025-12-18 17:03:52 UTC; 1min 25s ago
      Docs: man:systemd-sysv-generator(8)
   Process: 2409 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
```

27. Add the same tags in the deployment group page

Tag group 1

Key	Value - optional	
<input type="text"/> Name X	<input type="text"/> codedeploy-runner X	Remove tag
<input type="text"/> Environment X	<input type="text"/> dev X	Remove tag

28. Click on Create deployment group and here deployment group successfully created

The screenshot shows a screenshot of the AWS CodeDeploy console. At the top, there is a modal window with a grey background. Inside the modal, on the left, is a "Cancel" button. In the center is a large orange button labeled "Create deployment group". Below the modal, the main page has a green success notification bar. The notification contains a checkmark icon, the word "Success", and the message "Deployment group created". To the right of the message is a small "X" icon for closing the notification. Below the notification bar, the breadcrumb navigation shows the path: "Developer Tools > CodeDeploy > Applications > brain-tasks-codedeploy > brain-tasks-eks-dg". The main content area displays the deployment group "brain-tasks-eks-dg". To the right of the deployment group name are three buttons: "Edit", "Delete", and a large orange "Create deployment" button. At the bottom of the page, there is a section titled "Deployment group details".

29. Now create deployment

The screenshot shows a user interface for managing application deployments. At the top, there are three navigation tabs: 'Deployments' (which is active and highlighted in blue), 'Deployment groups', and 'Revisions'. Below the tabs, the title 'Application deployment history' is displayed. Underneath the title are several action buttons: a white button with a circular arrow icon, a white button labeled 'View details', a white button labeled 'Actions ▾', a white button labeled 'Copy deployment', and a white button labeled 'Retry deployment'. At the bottom of the interface is a prominent orange button labeled 'Create deployment'.

30. Add deployment group name

Deployment settings

Application

brain-tasks-codeddeploy

Deployment group



brain-tasks-eks-dg



31. Choose revision type "my application is stored in github"

Revision type

My application is stored in
Amazon S3

My application is stored in GitHub

GitHub token name

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



Connect to GitHub

32. Connect to github

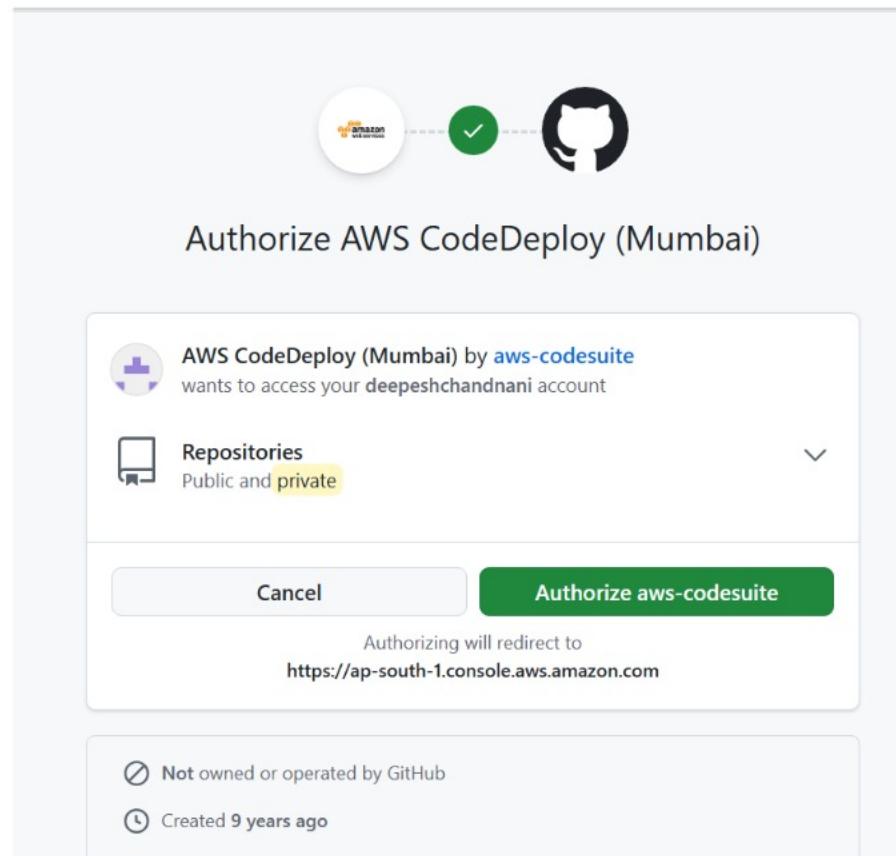
GitHub token alias

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



Connect to GitHub

33. Authorize aws codedeploy



34. Add repo name

Repository name

Commit ID

35. Now click on create deployment

Cancel

Create deployment

36. Watch logs it shows deployment completed successfully

```
1  #!/bin/bash
2  set -e
3
4  echo "Finding latest deployment archive..."
5  LATEST_DIR=$(ls -td /opt/codedeploy-agent/deployment-root/*/*/*deployment-archive | head -1)
6
7  echo "Moving to deployment archive: $LATEST_DIR"
8  cd "$LATEST_DIR"
9
10 echo "Updating kubeconfig..."
11 aws eks update-kubeconfig \
12   --region ap-south-1 \
13   --name brain-eks
14
15 echo "Deploying to EKS..."
16 kubectl apply -f k8s/deployment.yaml
17 kubectl apply -f k8s/service.yaml
18
19 echo "Deployment completed successfully"
20
```



37. The all stages are executed successfully

Event	Duration	Status	Error code
ApplicationStop	less than one second	Succeeded	-
DownloadBundle	less than one second	Succeeded	-
BeforeInstall	less than one second	Succeeded	-
Install	less than one second	Succeeded	-
AfterInstall	4 seconds	Succeeded	-
ApplicationStart	less than one second	Succeeded	-
ValidateService	less than one second	Succeeded	-