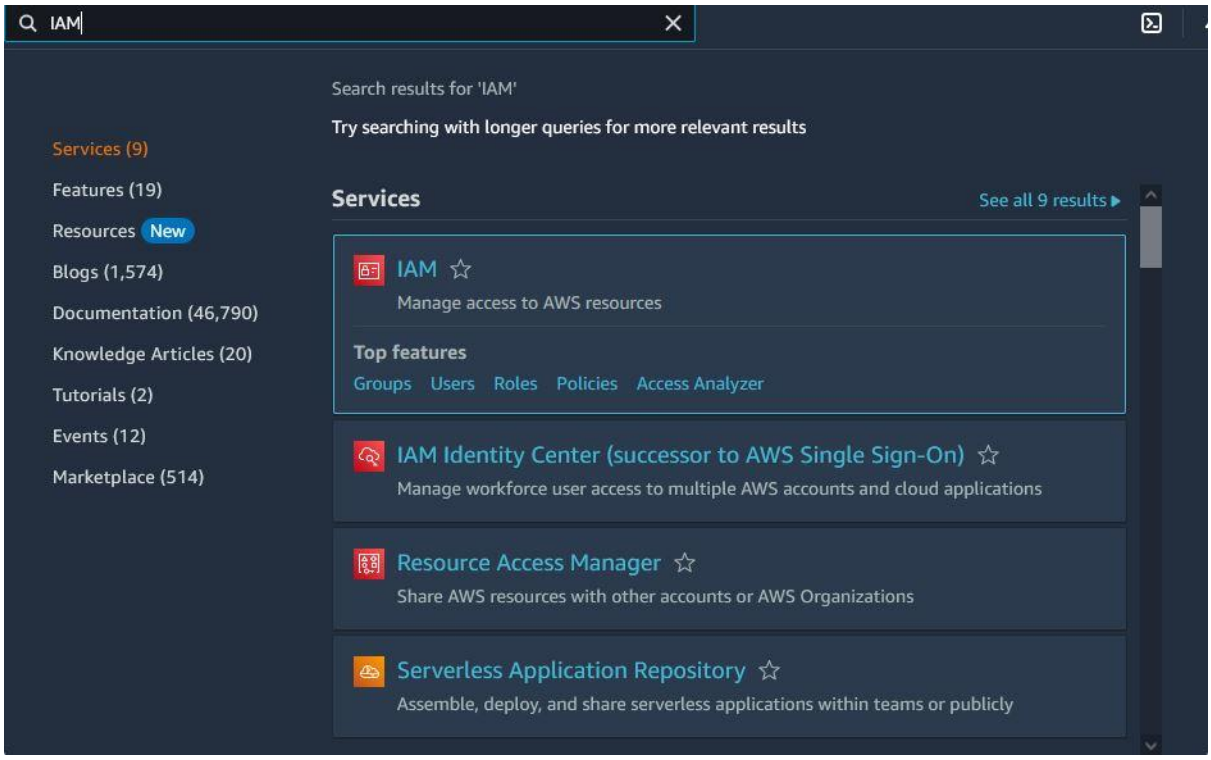
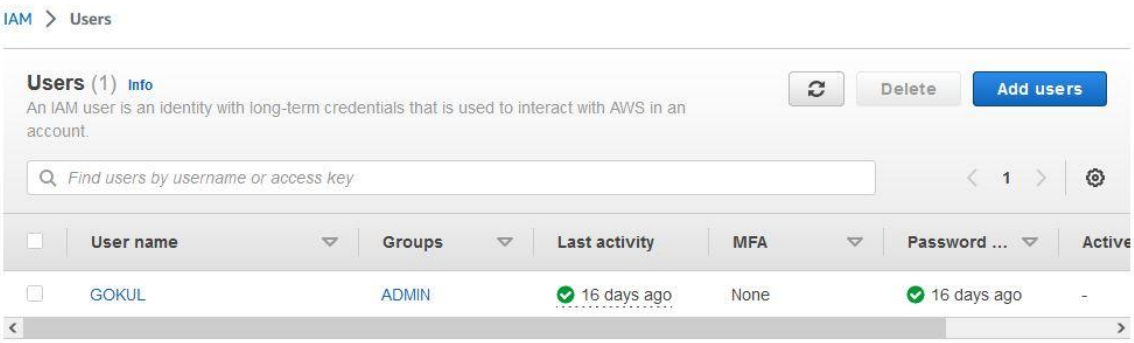
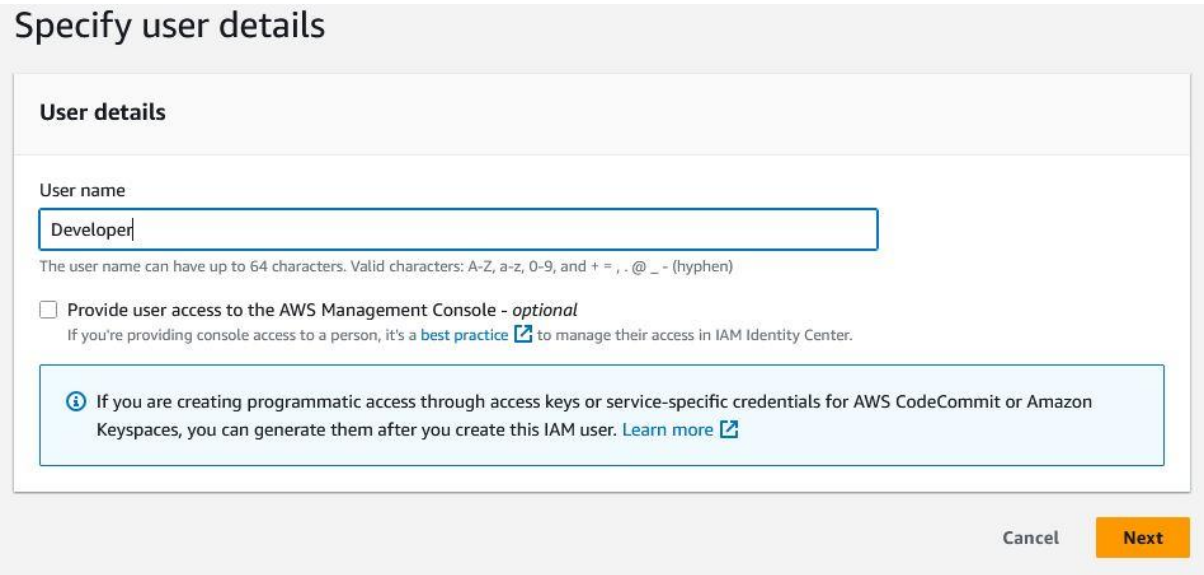
CODE DEPLOY AND CODE PIPELINE

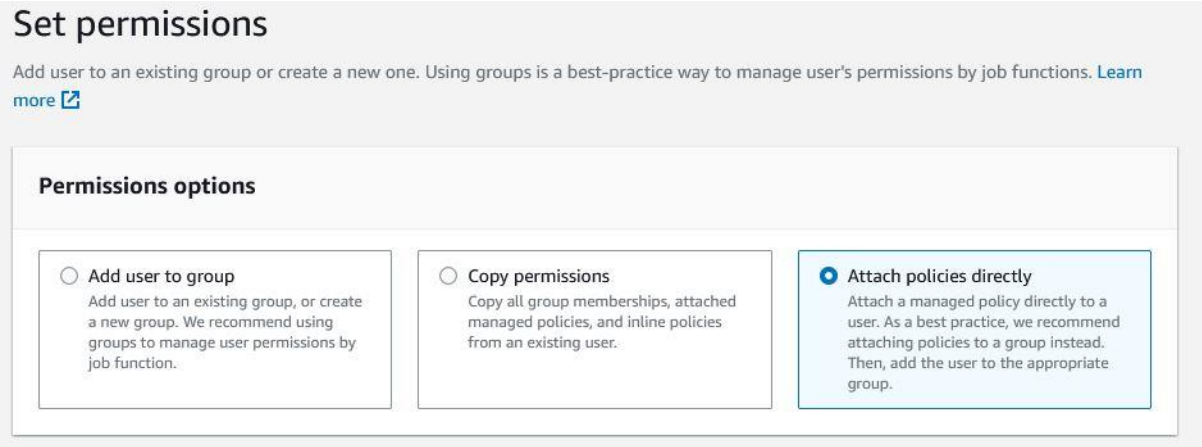
Step 1: Create an IAM user account for 'DEVELOPER'

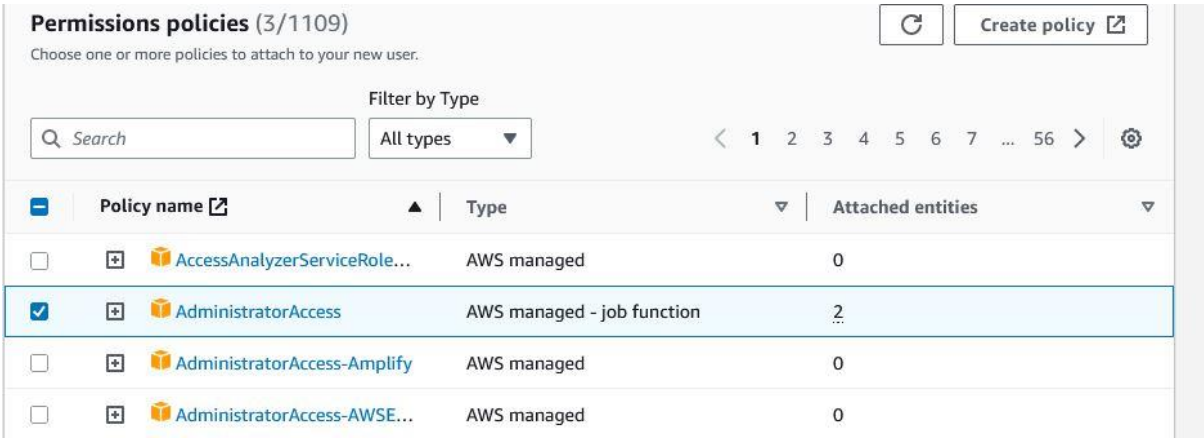
Go to IAM → select user → add user with CLI access and required permission.

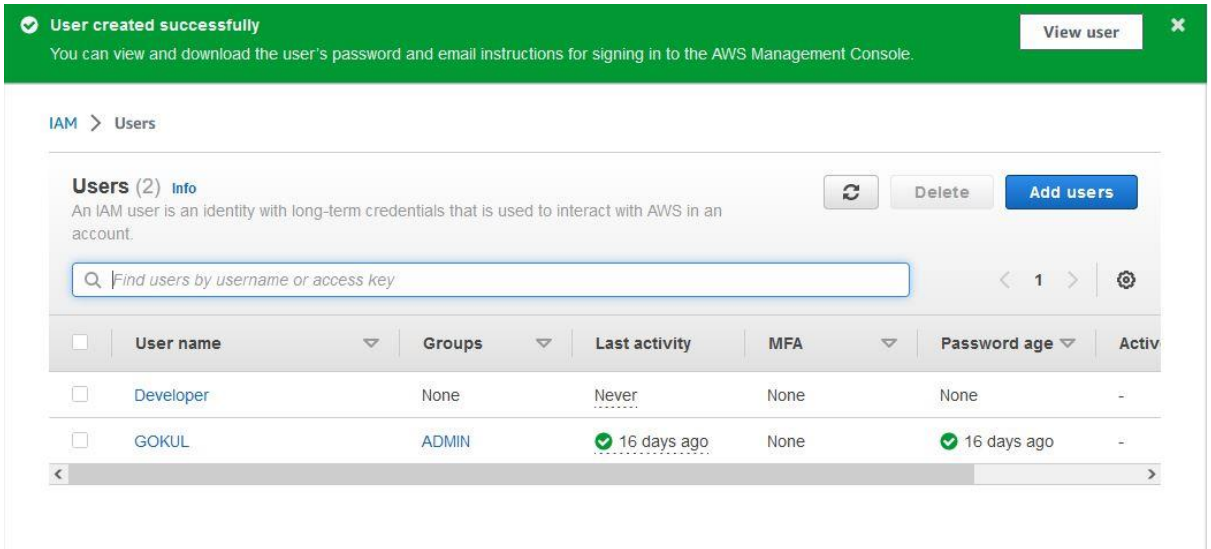












Next to give CLI access for user

Click username → security credentials → scroll down click create access key and copy paste the access key and secrete access key

IAM > Users > Developer

Developer Info

Delete

Summary

ARN	Console access	Access key 1
arn:aws:iam::207146887812:user/Developer	Disabled	Not enabled
Created	**Last console sign-in**	**Access key 2**
June 28, 2023, 11:50 (UTC+05:30)	-	Not enabled

Permissions | Groups | Tags | Security credentials | Access Advisor

Console sign-in

Enable console access

Console sign-in link	Console password
https://gokul0311.signin.aws.amazon.com/console	Not enabled

Multi-factor authentication (MFA) (0)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

Remove | Resync | Assign MFA device

Device type	Identifier	Certifications	Created on

No MFA devices. Assign an MFA device to improve the security of your AWS environment

Assign MFA device

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

Create access key

No access keys

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more

Create access key

IAM > Users > Developer > Create access key

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

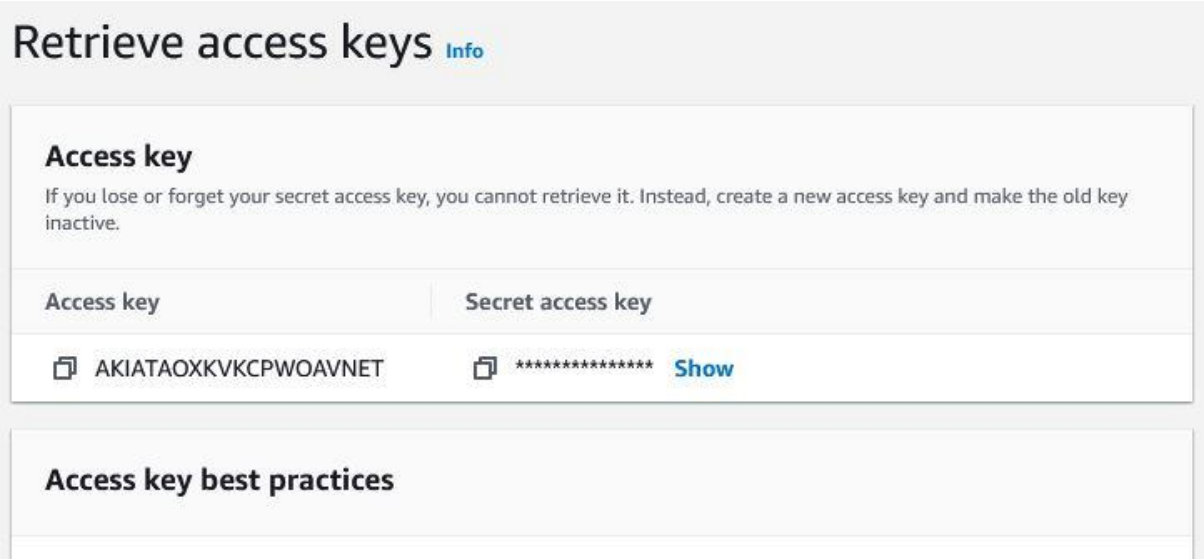
Step 3
Retrieve access keys

Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

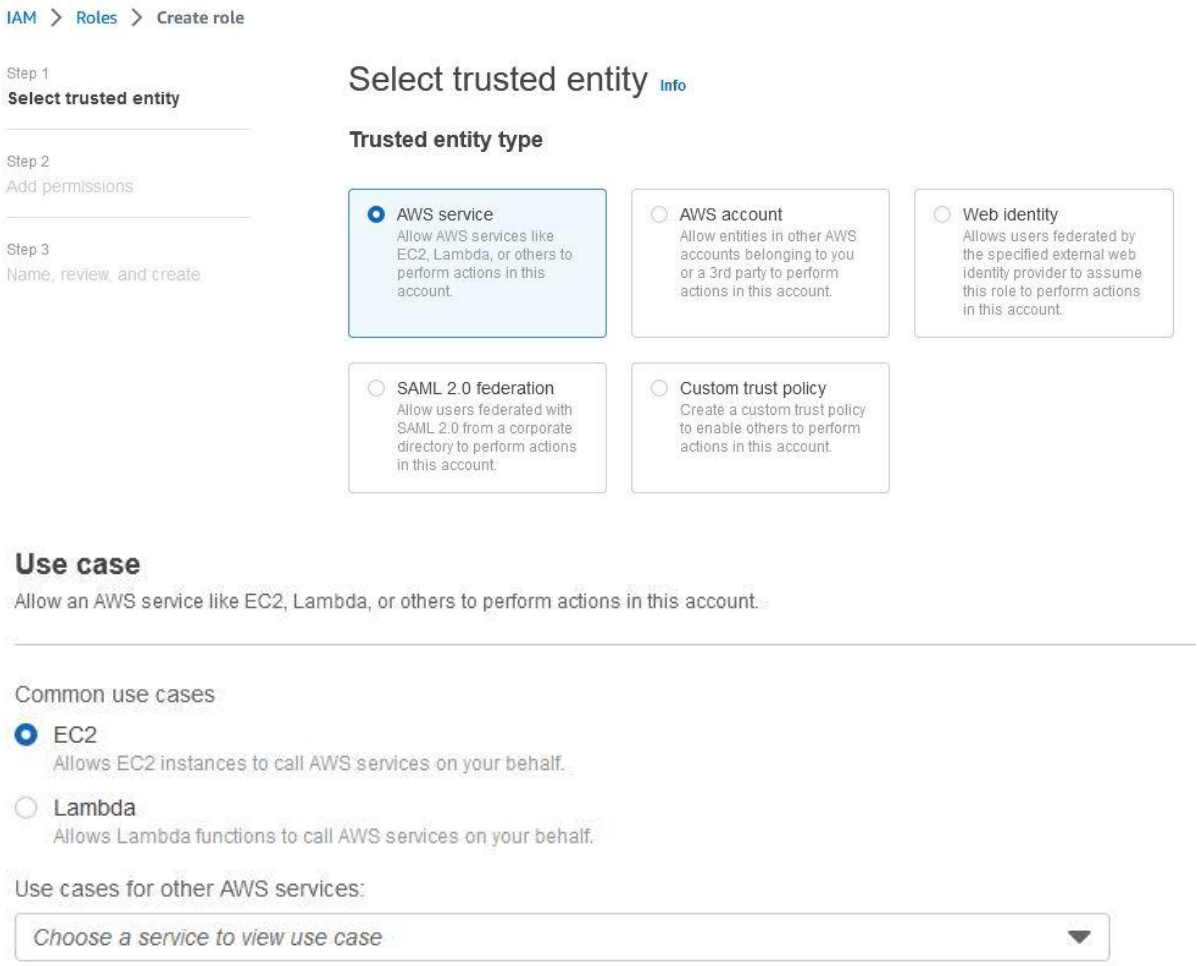
Use case

- (●) **Command Line Interface (CLI)**
  You plan to use this access key to enable the AWS CLI to access your AWS account.
- ( ) **Local code**
  You plan to use this access key to enable application code in a local development environment to access your AWS account.



Step 2: Create IAM role [ec2 and S3 full access] for production server

Select role → create role → aws service → select ec2 → attach policy ec2 and S3 full access → create role



Add permissions Info

Permissions policies (Selected 2/862) Info
Choose one or more policies to attach to your new role.

Create policy

Filter policies by property or policy name and press enter.

9 matches 1

"s3" Clear filters

Policy name	Type	Description
AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.
AmazonS3ReadOnl...	AWS m...	Provides read only access to all buckets via the AWS Management Cons...

Add permissions Info

Permissions policies (Selected 2/862) Info
Choose one or more policies to attach to your new role.

Create policy

Filter policies by property or policy name and press enter.

29 matches 1 2

Properties
Type
Path
Used as

Clear filters

name	Type	Description
LAMDAEC2POLICY	Custom...	
AmazonEC2FullAcc...	AWS m...	Provides full access to Amazon EC2 via the AWS Management Console.
AmazonEC2ReadO...	AWS m...	Provides read only access to Amazon EC2 via the AWS Management Co...

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

EC2-CDCP-ROLE

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

Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

IAM > Roles > EC2-CDCP-ROLE

EC2-CDCP-ROLE

Delete

Allows EC2 instances to call AWS services on your behalf.

Summary

Edit

Creation date	ARN	Instance profile ARN
June 28, 2023, 12:02 (UTC+05:30)	arn:aws:iam::207146887812:role/EC2-CDCP-ROLE	arn:aws:iam::207146887812:instance-profile/EC2-CDCP-ROLE
Last activity	Maximum session duration	
None	1 hour	

Step 3: Create IAM role for code deploy

Role → create role → aws service → select other use search Code deploy → create role

Select trusted entity Info

Trusted entity type

- (●) **AWS service** — Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ( ) **AWS account** — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ( ) **Web identity** — Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ( ) **SAML 2.0 federation** — Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ( ) **Custom trust policy** — Create a custom trust policy to enable others to perform actions in this account.

Use case

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

Common use cases

- ( ) **EC2** — Allows EC2 instances to call AWS services on your behalf.
- ( ) **Lambda** — Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

CodeDeploy

- (●) **CodeDeploy** — Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.
- ( ) **CodeDeploy for Lambda** — Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.
- ( ) **CodeDeploy - ECS** — Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.

IAM > Roles > CDCP-ROLE

CDCP-ROLE

Delete

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

Summary

Edit

Creation date June 28, 2023, 12:07 (UTC+05:30)	ARN arn:aws:iam::207146887812:role/CDCP-ROLE
Last activity None	Maximum session duration 1 hour

Step 4: Create a two ec2 instance in Linux i.e, developer and production server

Instances (2) Info

Connect | Instance state | Actions | Launch instances

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon
DEVELOPER S...	i-0440ac83097bbae36	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a
PRODUCTION ...	i-0476930f4ea890aed	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a

Step 5: Create an S3 bucket for copy the code from developer machine

S3 → select bucket → create bucket with all public access, ACL enable and bucket versioning is enabled.

s3

Search results for 's3'

Try searching with longer queries for more relevant results

Services (7)
Features (19)
Resources New
Blogs (1,256)
Documentation (20,703)
Knowledge Articles (20)
Tutorials (12)
Events (26)
Marketplace (1,173)

Services See all 7 results

S3
Scalable Storage in the Cloud

S3 Glacier
Archive Storage in the Cloud

AWS Snow Family
Large Scale Data Transport

AWS Transfer Family
Fully managed support for SFTP, FTPS and FTP

Features See all 19 results

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

mycdcpbucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

AWS Region

Asia Pacific (Mumbai) ap-south-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

- ACLs disabled (recommended)
  All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
- ACLs enabled
  Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

- Bucket owner preferred
  If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.
- Object writer
  The object writer remains the object owner.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning

- Disable
- Enable

Step 6: To login the developer server through putty with access the CLI commands in developer server

i) Login server with ppk key
ii) To configure CLI → commands given below

"aws configure" → press enter

Next enter the access key, secrete access key, region, format →press enter

```
ec2-user@ip-172-31-33-166:~
login as: ec2-user
Authenticating with public key "linuxkey"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-33-166 ~]$
```

```
ec2-user@ip-172-31-33-166:~
login as: ec2-user
Authenticating with public key "linuxkey"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-33-166 ~]$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]: ^C
[ec2-user@ip-172-31-33-166 ~]$ aws configure
AWS Access Key ID [None]: AKIATAOXKVKCPWOAVNET
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
[ec2-user@ip-172-31-33-166 ~]$
```

Step 7: Create a code for developer machine, commands given below

i) mkdir deploy_dir → create directory
ii) cd deploy_dir
iii) mkdir sampleapp → create directory in deploy_dir
iv) cd sampleapp
v) vi index.html → write html code and save

```
[root@ip-172-31-33-166 ~]# mkdir deploy_dir
[root@ip-172-31-33-166 ~]# cd deploy_dir
[root@ip-172-31-33-166 deploy_dir]# mkdir sampleapp
[root@ip-172-31-33-166 deploy_dir]# cd sampleapp
[root@ip-172-31-33-166 sampleapp]# vi index.html
```

root@ip-172-31-33-166:~/deploy_dir/sampleapp

```
<!DOCTYPE html>
<html>
<head>
    <title>Registration Form</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Registration Form</h1>
        <form>
            <label for="name">Name</label>
            <input type="text" id="name" name="name" required>

            <label for="email">Email</label>
            <input type="email" id="email" name="email" required>

            <label for="phone">Phone</label>
            <input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required>

            <label for="password">Password</label>
            <input type="password" id="password" name="password" required>

            <input type="submit" value="Submit" onclick="submitForm()">
        </form>
    </div>
    <script src="script.js"></script>
</body>
</html>
~
"index.html" 28L, 834B                                                    28,7
```

vi) vi appspec.yml → to write yml file and save it

root@ip-172-31-33-166:~/deploy_dir/sampleapp/scripts

```
version: 0.0
os: linux
files:
 - source: /index.html
   destination: /var/www/html/
hooks:
 BeforeInstall:
  - location: scripts/httpd_install.sh
    timeout: 300
    runas: root
  - location: scripts/httpd_start.sh
    timeout: 300
    runas: root
 ApplicationStop:
  - location: scripts/httpd_stop.sh
    timeout: 300
    runas: root
~
~
~
~
```

vii) mkdir scripts → to create directory in sampleapp
viii) cd scripts
ix) vi httpd_install.sh → to write a bash command in install httpd

```
root@ip-172-31-33-166:~/deploy_dir/sampleapp/scripts
#!/bin/bash
yum install -y httpd
~
~
~
~
~
```

x) vi httpd_start.sh → to write a bash command in start httpd

```
root@ip-172-31-33-166:~/deploy_dir/sampleapp/scripts
#!/bin/bash
systemctl start httpd
systemctl enable httpd
~
~
~
~
~
~
~
~
~
```

xi) vi httpd_stop.sh → to write a bash command in start httpd

```
root@ip-172-31-33-166:~/deploy_dir/sampleapp/scripts
#!/bin/bash
systemctl stop httpd
~
~
~
~
~
~
```

xii) chmod 777 * → to given full permission scripts

Step 8: To install a code deploy agent in production server, commands given below

i) yum install ruby -y

ii)wget https://aws-codedeploy-us-east1.s3.amazonaws.com/latest/install

ii) chmod +x install
iii) ./install auto
iv) service codedeploy-agent status

```
root@ip-172-31-41-30:~

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-41-30 ~]$ sudo -i
[root@ip-172-31-41-30 ~]# yum install ruby -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package ruby.x86_64 0:2.0.0.648-36.amzn2.0.4 will be installed
--> Processing Dependency: ruby-libs(x86-64) = 2.0.0.648-36.amzn2.0.4 for package: ruby-2.0.0.648-36.a
zn2.0.4.x86_64
--> Processing Dependency: rubygem(bigdecimal) >= 1.2.0 for package: ruby-2.0.0.648-36.amzn2.0.4.x86_6
--> Processing Dependency: ruby(rubygems) >= 2.0.14.1 for package: ruby-2.0.0.648-36.amzn2.0.4.x86_64
--> Processing Dependency: libruby.so.2.0()(64bit) for package: ruby-2.0.0.648-36.amzn2.0.4.x86_64
--> Running transaction check
---> Package ruby-libs.x86_64 0:2.0.0.648-36.amzn2.0.4 will be installed
---> Package rubygem-bigdecimal.x86_64 0:1.2.0-36.amzn2.0.4 will be installed
---> Package rubygems.noarch 0:2.0.14.1-36.amzn2.0.4 will be installed
```

```
[root@ip-172-31-41-30 ~]# wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
--2023-06-28 07:46:13--  https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)... 52.2
16.152.36, 52.217.48.36, 52.217.100.188, ...
Connecting to aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)|52.
216.152.36|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17892 (17K) []
Saving to: 'install'

100%[===============================================================>] 17,892      93.7KB/s   in 0.2s

2023-06-28 07:46:14 (93.7 KB/s) - 'install' saved [17892/17892]
```

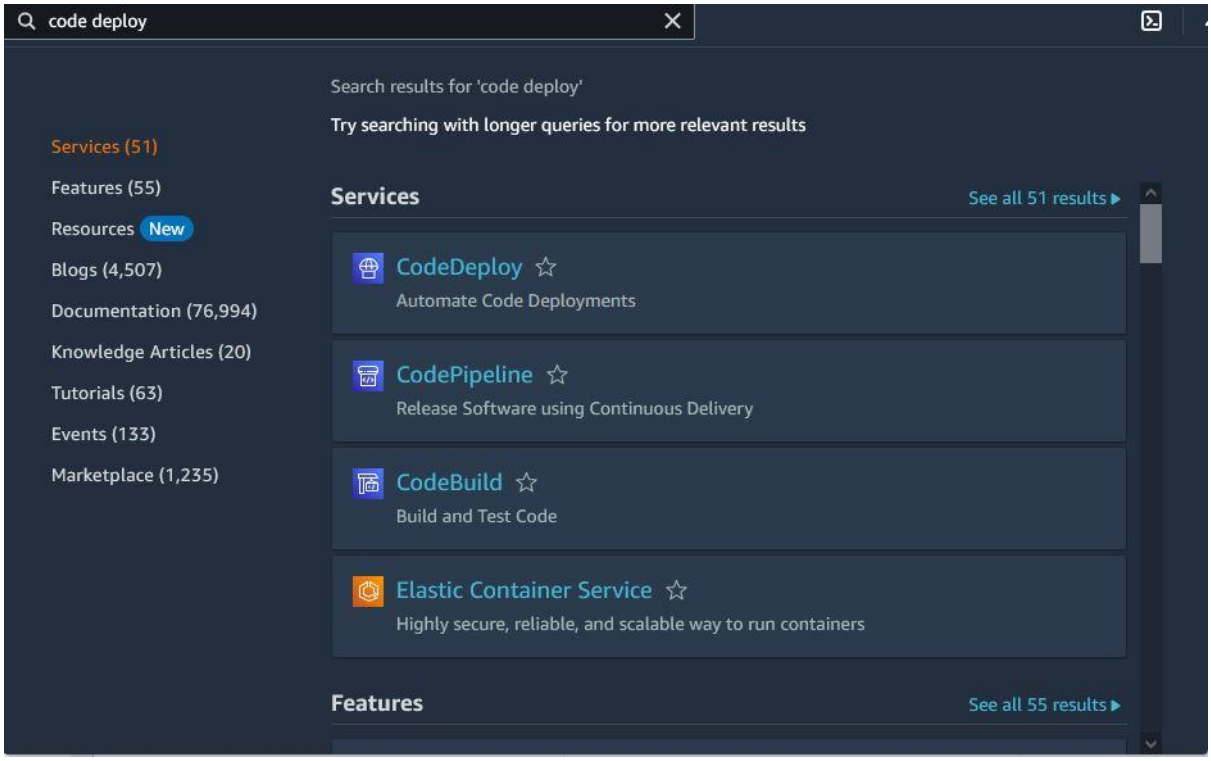
```
[root@ip-172-31-41-30 ~]# service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 32546
[root@ip-172-31-41-30 ~]#
```

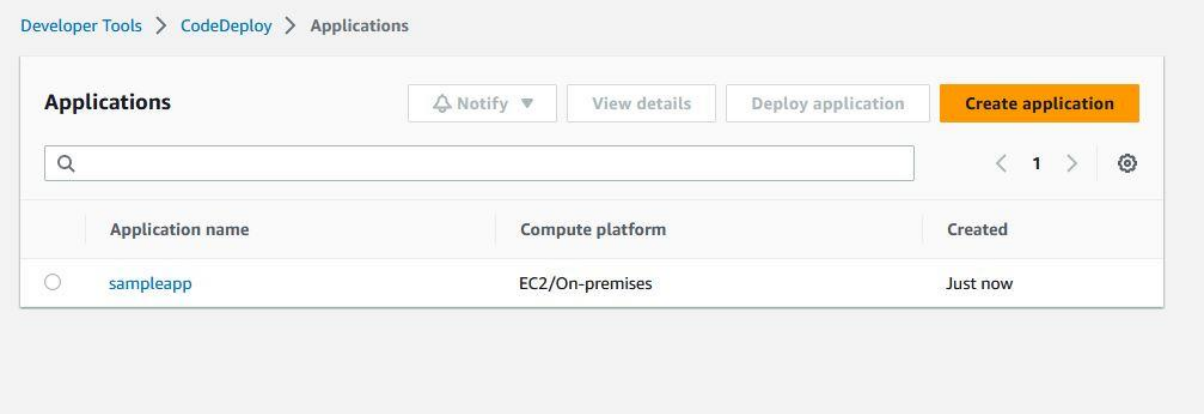
Step 9: Create Code deploy Application and Push the code to S3 bucket from Developer machine, commands given below

i) cd deploy_dir/sampleapp → to open the sampleapp directory

ii) aws deploy create-application --application-name sampleapp
then to check code deploy application is created

```
[root@ip-172-31-33-166 ~]# cd deploy_dir
[root@ip-172-31-33-166 deploy_dir]# cd sampleapp
[root@ip-172-31-33-166 sampleapp]# cd scripts
[root@ip-172-31-33-166 scripts]# aws deploy create-application --application-name sampleapp
{
    "applicationId": "8339bdd6-ce5c-4239-8950-2ace27ad6265"
}
[root@ip-172-31-33-166 scripts]#
```





iii) aws deploy push --application-name sampleapp --s3-location s3://aws24092021/sampleapp.zip → to copy the code for s3 from developer machine

```
[root@ip-172-31-33-166 scripts]# aws deploy push --application-name sampleapp --s3-location s3://mycdcp
bucket2806/sampleapp.zip
To deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=mycdcpbucket2806,key=sam
pleapp.zip,bundleType=zip,eTag=52d236819e5bb024b07cf0cb1507e69f,version=.Uk8M5k.GFx1EdzWpx_PIIKXfdgaX3y
9 --deployment-group-name <deployment-group-name> --deployment-config-name <deployment-config-name> --d
escription <description>
```

Step 10: To create a code deployment group and create deployment

Select code deploy application name → create deployment group → create deployment

Developer Tools > CodeDeploy > Applications > sampleapp

sampleapp

Notify ▼ | Delete application

Application details

Name	Compute platform
sampleapp	EC2/On-premises

Deployments | Deployment groups | Revisions

Deployment groups

View details | Edit | Create deployment group

1

Name	Status	Last attempted deployment	Last successful deployment	Trigger count

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

Create deployment group

Application

Application

sampleapp

Compute type

EC2/On-premises

Deployment group name

Enter a deployment group name

mydeploymentgroup

100 character limit

Service role

Enter a service role

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

arn:aws:iam::207146887812:role/CDCP-ROLE

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
- ( ) **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
- [x] Amazon EC2 instances
  1 unique matched instance. Click here for details

You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key	Value - *optional*	
Name	PRODUCTION SERVER	Remove tag

Add tag

+ Add tag group

Success
Deployment group created

Developer Tools > CodeDeploy > Applications > sampleapp > mydeploymentgroup

mydeploymentgroup

Edit | Delete | Create deployment

Deployment group details

Deployment group name mydeploymentgroup	Application name sampleapp	Compute platform EC2/On-premises
Deployment type In-place	Service role ARN arn:aws:iam::207146887812:role/CDCP-ROLE	Deployment configuration CodeDeployDefault.AllAtOnce
Rollback enabled False	Agent update scheduler Learn to schedule update in AWS Systems Manager	

Developer Tools > CodeDeploy > Applications > sampleapp > Create deployment

Create deployment

Deployment settings

Application
sampleapp

Deployment group
mydeploymentgroup

Compute platform
EC2/On-premises

Deployment type
In-place

Deployment type
In-place

Revision type

- (●) My application is stored in Amazon S3
- ( ) My application is stored in GitHub

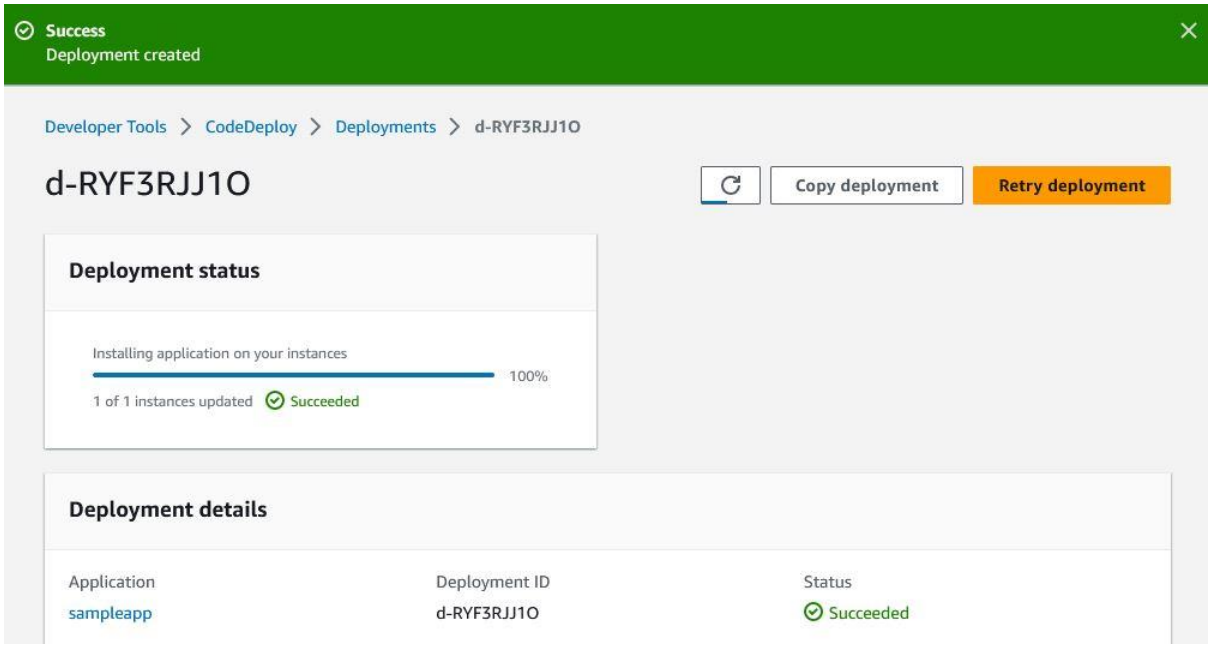
Revision location
Copy and paste the Amazon S3 bucket where your revision is stored

s3://mycdcpbucket2806/sampleapp.zip?versionId=.Uk8M5k.GFx1EdzWpx_PIIK

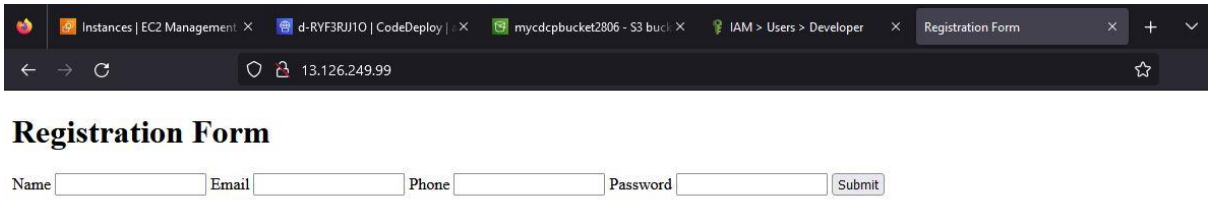
Use "s3://mycdcpbucket2806
/sampleapp.zip?versionId=.Uk8M5k.GFx1EdzWpx_PIIKXfdgaX3y9&
eTag=52d236819e5bb024b07cf0cb1507e69f"

**s3://mycdcpbucket2806
/sampleapp.zip?versionId=.Uk8M5k.GFx1EdzWpx_PIIKXfdgaX3y9&
eTag=52d236819e5bb024b07cf0cb1507e69f**

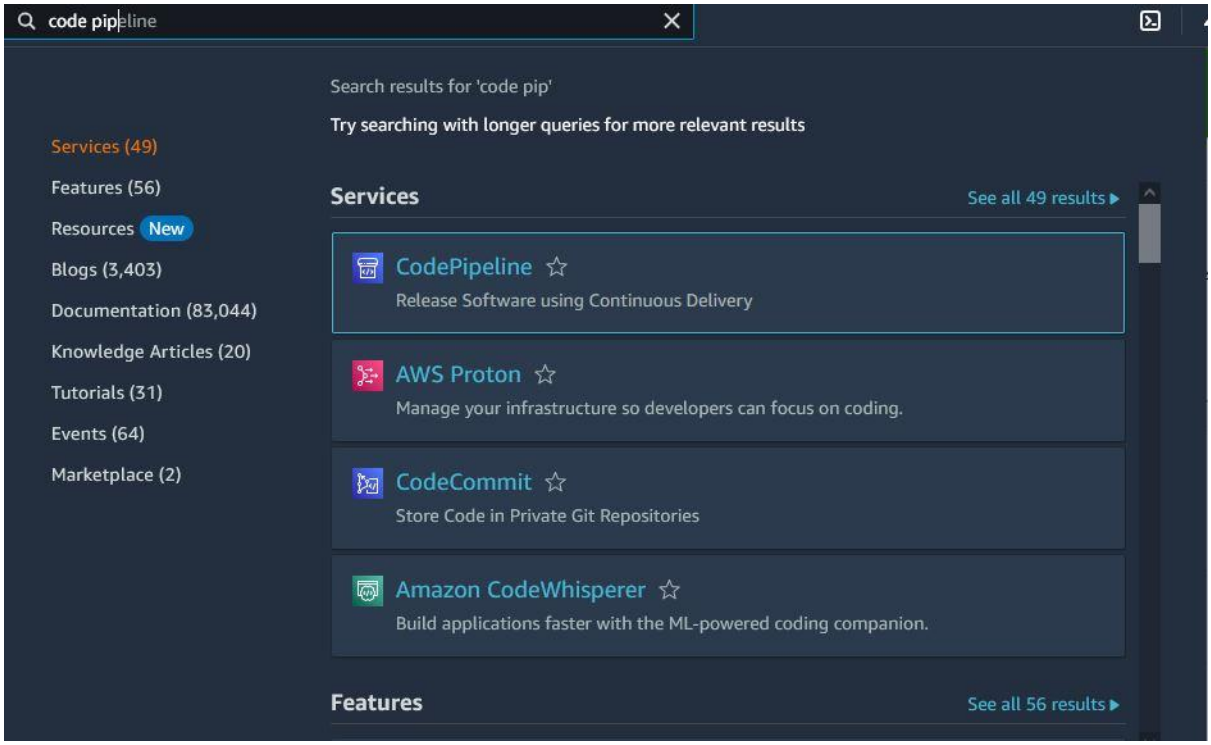
Deployment description

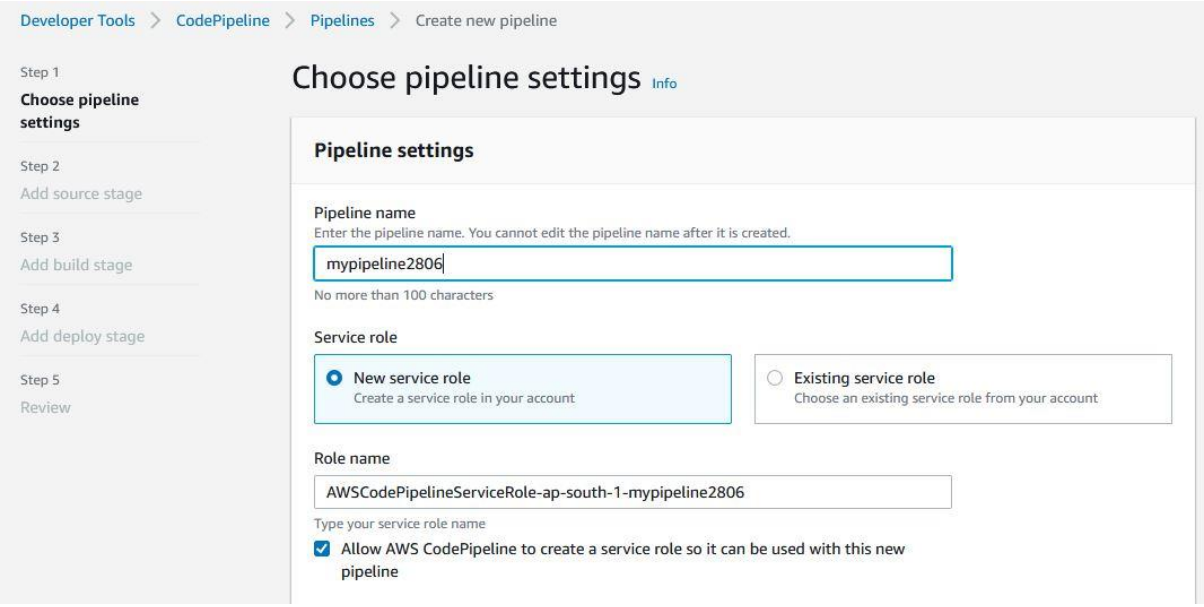


Step 11: Once deployment is succeeded to check browser with public ip address



Step 12: Create code pipeline for new code will be updated in deployment group


code pipeline
Search results for 'code pip'
Try searching with longer queries for more relevant results
Services (49)
Features (56)
Resources New
Blogs (3,403)
Documentation (83,044)
Knowledge Articles (20)
Tutorials (31)
Events (64)
Marketplace (2)
Services
See all 49 results
CodePipeline
Release Software using Continuous Delivery
AWS Proton
Manage your infrastructure so developers can focus on coding.
CodeCommit
Store Code in Private Git Repositories
Amazon CodeWhisperer
Build applications faster with the ML-powered coding companion.
Features
See all 56 results


Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Choose pipeline settings Info
Pipeline settings
Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
mypipeline2806
No more than 100 characters
Service role
New service role
Create a service role in your account
Existing service role
Choose an existing service role from your account
Role name
AWSCodePipelineServiceRole-ap-south-1-mypipeline2806
Type your service role name
Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Source

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

Amazon S3

Bucket

mycdcpbucket2806

S3 object key

sampleapp.zip

Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

- (●) Amazon CloudWatch Events (recommended)
  Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs
- ( ) AWS CodePipeline
  Use AWS CodePipeline to check periodically for changes

Cancel | Previous | Next

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

Region

Asia Pacific (Mumbai)

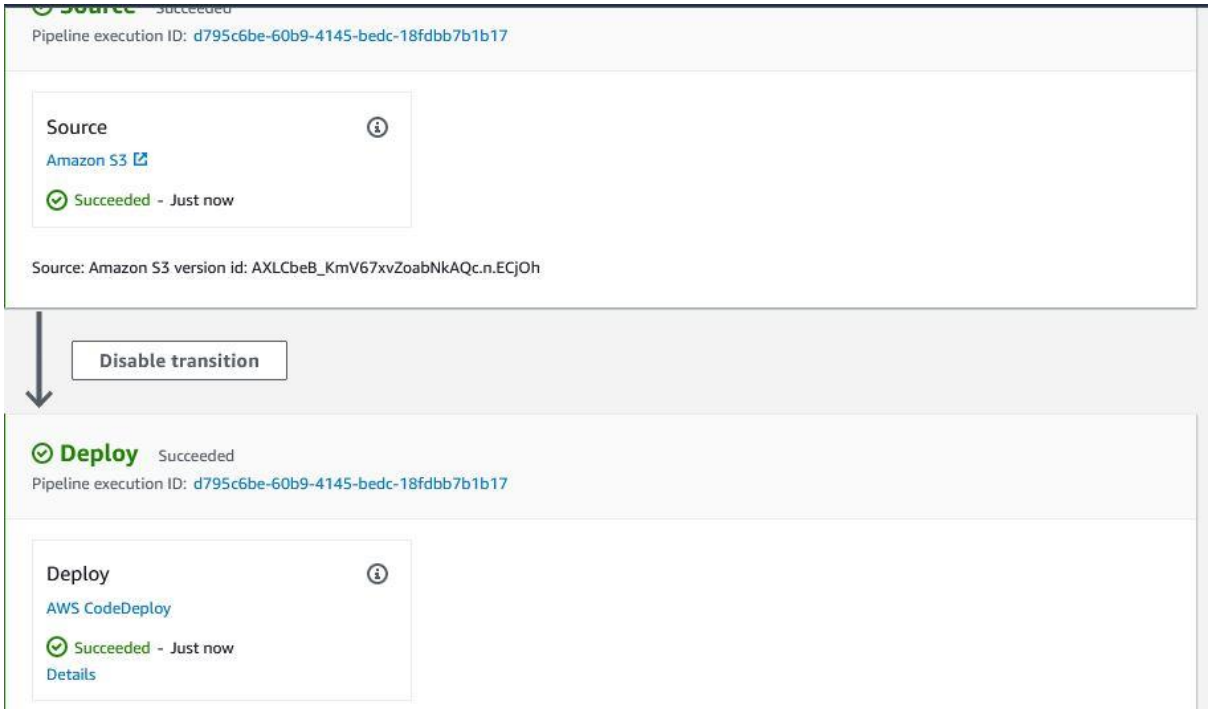
Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

sampleapp

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

mydeploymentgroup

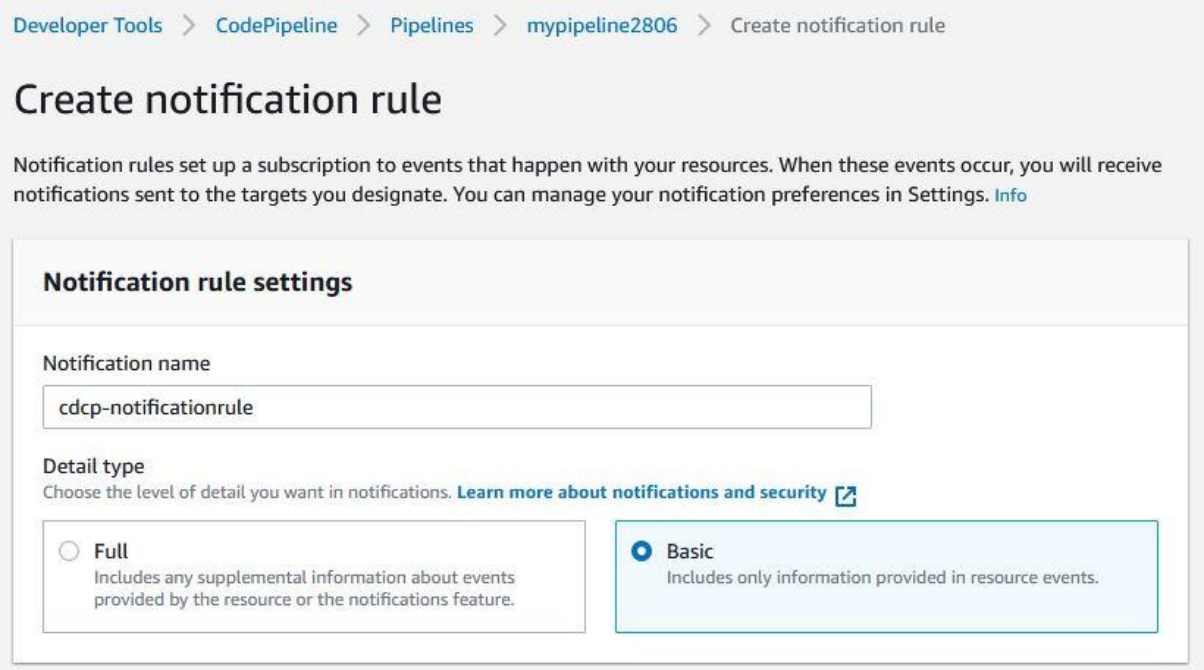
Cancel | Previous | Next



Step 13: Create a Notification for code pipeline action execution state change

Code pipeline → setting → create a notification rule → to fill the details → create a SNS target → create notification.

Then go to SNS → topic is created → subscription is create → confirm the subscription



Events that trigger notifications

Select none | Select all

Action execution	Stage execution	Pipeline execution	Manual approval
☑ Succeeded	☑ Started	☑ Failed	☑ Failed
☑ Failed	☑ Succeeded	☑ Canceled	☑ Needed
☑ Canceled	☑ Resumed	☑ Started	☑ Succeeded
☑ Started	☑ Canceled	☑ Resumed	
	☑ Failed	☑ Succeeded	
		☑ Superseded	

Targets

Create a target to use specifically for this notification rule. SNS topics created as targets have no subscribers but have all policies applied to act as a target for notifications. If you choose AWS Chatbot, you will be redirected to create a client in the AWS Chatbot console. Learn more

Create target

Configured targets

Choose target type	Choose target	
SNS topic	arn:aws:sns:ap-south-1:207146	Remove row

Add row

Notification rule created

cdcp-notificationrule notification rule

View all notification rules | Edit | Delete

Notification rule settings

Notification name
cdcp-notificationrule

Notification ARN
arn:aws:codestar-notifications:ap-south-1:207146887812:notificationrule/e2fd29574501b21983e34a991b8c32f804cb1281

Notification status
Sending notifications

Resource ARN
arn:aws:codepipeline:ap-south-1:207146887812:mypipeline2806

Amazon SNS > Topics > codestar-notifications-

codestar-notifications-

Edit | Delete | Publish message

Details

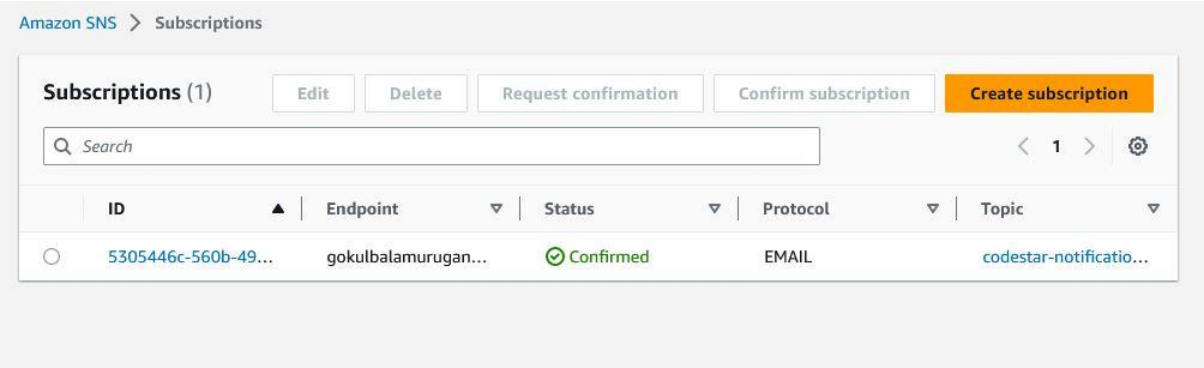
Name
codestar-notifications-

Display name
-

ARN
arn:aws:sns:ap-south-1:207146887812:codestar-notifications-

Topic owner
207146887812

Type
Standard



Step 14: To change the html code and copy the s3 bucket, commands given below

i) zip -r ../sampleapp.zip .
ii) aws s3 cp sampleapp.zip s3://aws280921

once code deployed to check the browser with public ip address and will get sns notification through subscription Email.