

Project Code Deployment Code Pipeline in AWS

Code Deploy is a deployment service from AWS which can automate application deployments to Amazon EC2 instances, on-premises instances or Lambda functions. This does a onetime deployment,

Code Pipeline for scheduling of deployment you may have to use AWS Code Pipeline also.

Scope of the Project :

1. Developer Developing Application code & Stored the source code file into Storage container (S3 Bucket)., the code Repository

2. The **CODE DEPLOYMENT Tool** Fetch the file from the code Repository S3, and deploy to the Web server or deployment Machine by Serive using **Code Deploy Agent**. This is one time Deployment Process.

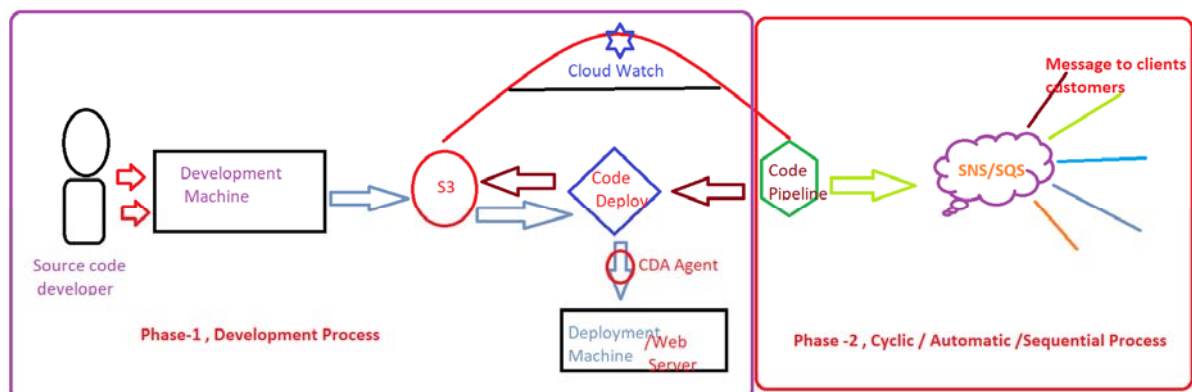
3. This process 1 & 2 may have happened again and again, and the repeated process has watch / monitored by a monitoring tool **Cloud WATCH**. If The code refreshed / reinitiate /newcode update like process , the monitoring tool will check & inform to **the continues / scheduled deployment tool Code PIPELINE** about the changes that happened in the code Repository S3.

Here the **the continues / scheduled deployment tool Code PIPELINE** acts a **Manager**,

4. **To manages the new code availability / modification happened / any chages occured**
The Code Pipeline instruct to Code Deploy about the modifications happened in Storage container and ask the Code Deployer to Deploy the new changes to Webserver.

5. Deployment Success , the **Code PIPELINE** send notifications about the changes to clients/customers through SNS messaging / mailing service.

Process – Phase 1:



Phase 1: Code Dev → Stoage S3 →Deploy → CDA → Dep. Machine / Web Server

Phase 2: Cyclic / Automatic / Sequential Process.

CD-CP Project:

1. EC2 --> VM
2. S3 --> Code Storage
3. IAM --> Service to Service Comm
4. SNS --> Notification
5. Code Deploy --> Deployment
6. Code Pipeline --> Seq Deploy
7. Cloud Watch --> Monitor

Implementation:

1. Create IAM – For Roles, 1.EC2-S3
2.Code Deploy
2. Create IAM – User Management- User Creation for Developer
3. EC2 - For
 1. Create Development Machine
 2. Create Deployment Machine / Webserver
Code deployment Agent creation
4. Code Development- Developer Machine
Create / develop application code
<dir> deploy_dir
 - >sampleapp\index.html
 - Appsec.yml
 - sampleapp\scripts\httpd.install.sh
 - httpd.start.sh
 - httpd.stop.sh

5 -Create Application & Push the code to S3 bucket

a- Create S3 bucket for uploading the code, I have named it as sampleapp
enable – version controller – use s3 as version controller tool

b- Change directory to **sampleapp developer machine** and create a code deploy application.
Execute the command below

```
# aws deploy create-application --application-name sampleapp
```

c- Now upload the code to S3 by the executing the command below. Directory of execution is important.

```
# aws deploy push --application-name sampleapp --s3-location s3://  
sampleapp/sampleapp.zip
```

d- Now browse the s3 bucket to see that sampleapp.zip is present.

6. Code Deploy Code Deployment

a- Login to Codedeploy AWS web console

b- Select sampleapp and click *Create Deployment Group* from *Deployment Groups* tab.

Service role

Code deployment configuration

Code Deployment : #1. Where to get Deployment

#2. When Will deployed

#3. CDA is perfect

#4. Developed code in S3 bucket

Single Code Deployment is Done.

7. Cloud Watch: Monitoring Tool / Updation/Modification/Changes

8. Code Pipeline: Automatic / Sequential Process

- **Create AWS CodePipeline**

1.Using code pipeline

Role of pipeline

Service role -> new service role / existing service role

2.Provide – S3

Browse – object key – sampleapp.zip

Change index.html content

Load & Push the new index.html to S3

9. SNS Messaging Service

Process #1 :

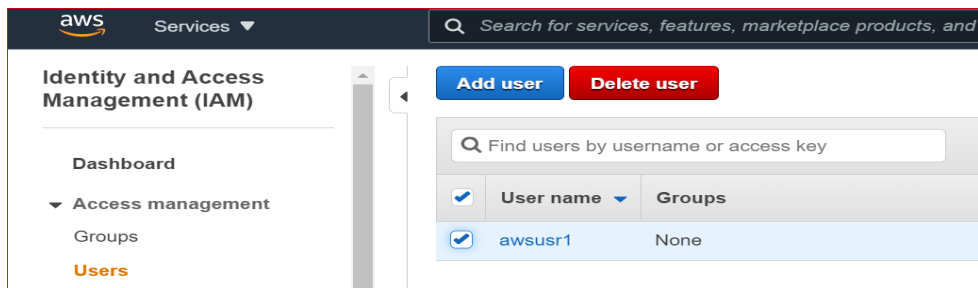
1. Create IAM – For Roles, 1.EC2-S3

2.Code Deploy

The screenshot shows the AWS IAM console interface. On the left, the 'Identity and Access Management (IAM)' menu is visible, with 'Roles' selected. The main content area displays a green notification: 'The role cdrole has been created.' Below this, there are 'Create role' and 'Delete role' buttons. A table lists the roles in the account:

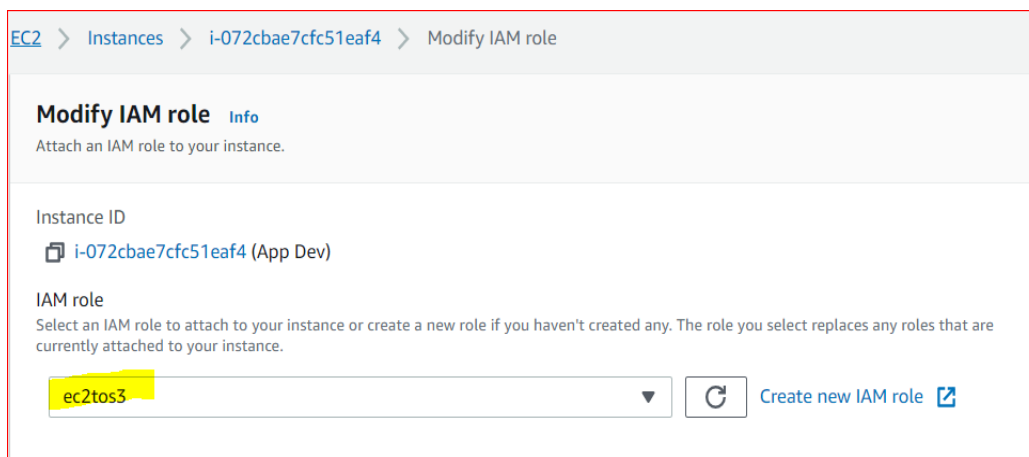
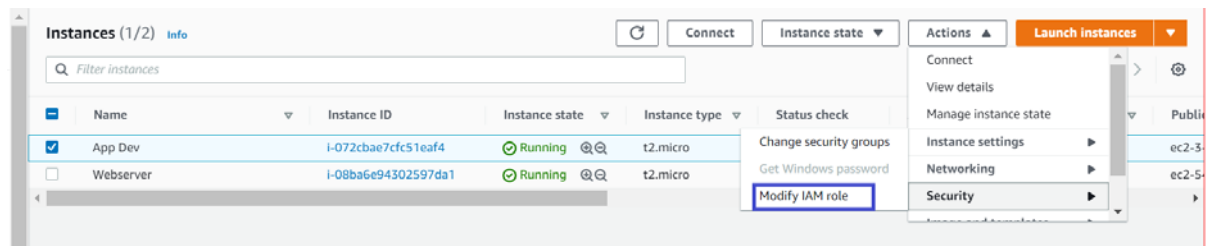
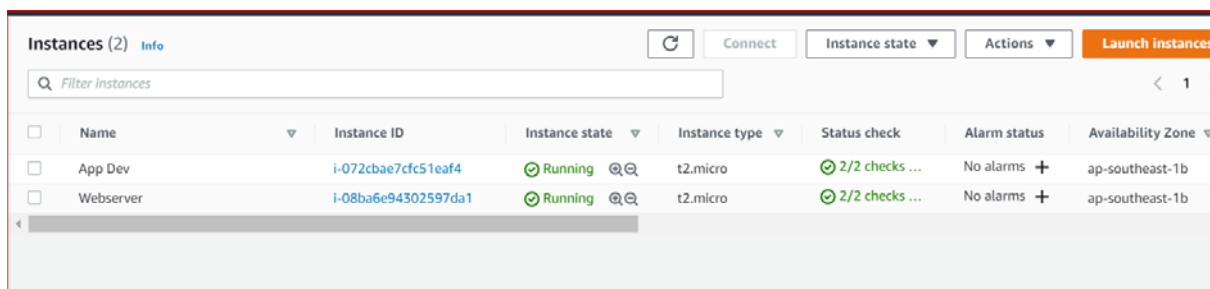
Role name	Trusted entities	Last activity
ec2tos3	AWS service: ec2	49 days
cdrole	AWS service: codedeploy	None
AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked ...)	None
AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	None
AWSServiceRoleForRDS	AWS service: rds (Service-Linked role)	49 days
AWSServiceRoleForMigrationHub	AWS service: migrationhub (Service-Linked r...)	None

2.Create IAM – User Management- User Creation for Developer

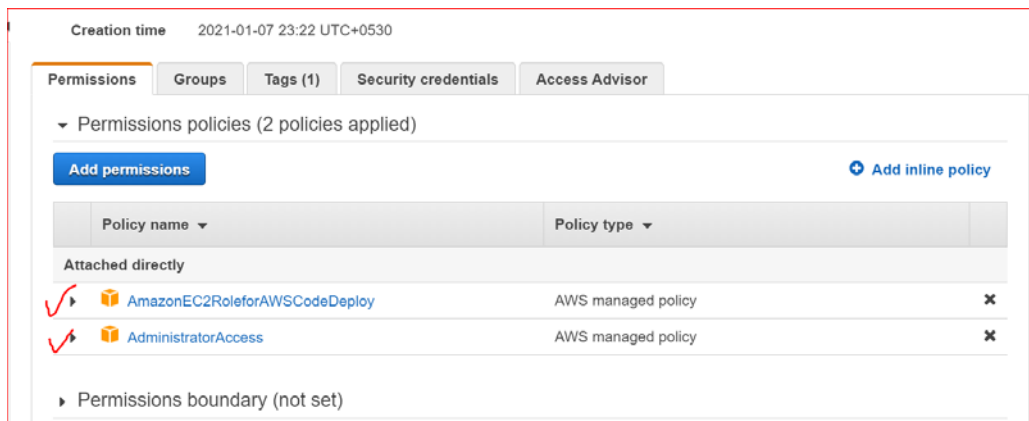


3 . EC2 - For

1. Create Development Machine
2. Create Deployment Machine / Webserver



Assign Roles



4. User Connect with App Developer Machine in Putty

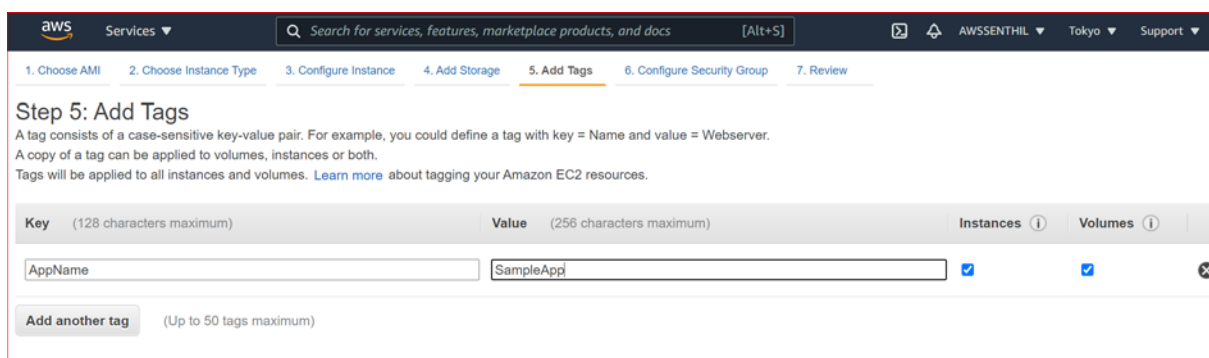
```
[root@ip-172-31-47-100 ~]# aws configure
AWS Access Key ID [*****F3F2]: AKIA5E5ABKTCXOVHF3F2
AWS Secret Access Key [*****blCP]: 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
Default region name [ap-northeast-1]: ap-northeast-1
Default output format [json]: json
[root@ip-172-31-47-100 ~]#
```

```
[root@ip-172-31-47-100 ~]# aws iam list-users
{
  "Users": [
    {
      "UserName": "awsusr1",
      "Path": "/",
      "CreateDate": "2021-01-08T01:15:32Z",
      "UserId": "AIDA5E5ABKTCZDHKZST3M",
      "Arn": "arn:aws:iam::903889376453:user/awsusr1"
    }
  ]
}
```

User Machine Created and ready

5. Development Machine / Code deployment Agent creation

Tag as The deployment group membership for the EC2 instance is decided by this Tag. We have used AppName Tag with value SampleApp
Role : ec2-s3 full access, SSH port -80



Login root in ssh

-Check updates - # yum update

```
Dependency Installed:
  nettle.x86_64 0:2.7.1-8.amzn2.0.2

Updated:
  chrony.x86_64 0:3.5.1-1.amzn2.0.1    cloud-init.noarch 0:19.3-4.amzn2    p11-kit.x86_64 0:0.23.22-1.amzn2.0.1    p11-kit-trust.x86_64 0:0.23.22-1.amzn2.0.1
  tzdata.noarch 0:2020d-2.amzn2

Completed!
```

-Install Ruby - # yum install ruby -y

```
Installed:
  ruby.x86_64 0:2.0.0.648-36.amzn2.0.1

Dependency Installed:
  ruby-irb.noarch 0:2.0.0.648-36.amzn2.0.1    ruby-libs.x86_64 0:2.0.0.648-36.amzn2.0.1    rubygem-bigdecimal.x86_64 0:1.2.0-36.amzn2.0.1
  rubygem-io-console.x86_64 0:0.4.2-36.amzn2.0.1    rubygem-json.x86_64 0:1.7.7-36.amzn2.0.1    rubygem-psych.x86_64 0:2.0.0-36.amzn2.0.1
  rubygem-rdoc.noarch 0:4.0.0-36.amzn2.0.1    rubygems.noarch 0:2.0.14.1-36.amzn2.0.1
```

Note : The Code Deploy Agent tool is developed in Ruby Language . so its needs to run CDA

- # yum install wget -y

```
[root@ip-172-31-46-114 ~]# yum install wget -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package wget-1.14-18.amzn2.1.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-46-114 ~]#
```

Down load CDA from aws

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

```
[root@ip-172-31-46-114 ~]# wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
--2021-01-08 03:51:18-- 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)... 52.216.226.176
Connecting to aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)|52.216.226.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17231 (17K) []
Saving to: 'install'

100%[=====>] 17,231    98.1KB/s  in 0.2s

2021-01-08 03:51:19 (98.1 KB/s) - 'install' saved [17231/17231]
```

Installing CDA

```
[root@ip-172-31-46-114 ~]# ./install auto
-bash: ./install: Permission denied
[root@ip-172-31-46-114 ~]# chmod +x install
[root@ip-172-31-46-114 ~]# ./install auto
```

```
dependencies Resolved

Package Arch Version Repository Size
-----
coddeploy-agent noarch 1.3.1-1880 /coddeploy-agent-1.3.1-1880.noarch.tmp-20210108-32438-ktncw7 11 M

Transaction Summary
Install 1 Package

Total size: 11 M
Installed size: 11 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Pre hook : 1
Checking the ruby version.
Checking if there is already a process named coddeploy-agent running.
Installing : coddeploy-agent-1.3.1-1880.noarch 1/1
Post hook : 1
Check if there is a coddeployagent config file.
Start coddeploy-agent in post hook if this is a first install.
Verifying : coddeploy-agent-1.3.1-1880.noarch 1/1

Installed:
  coddeploy-agent.noarch 0:1.3.1-1880
```

```
[root@ip-172-31-46-114 ~]# service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 32524
[root@ip-172-31-46-114 ~]#
```

6.Code Development

-Application Machine

```
[root@ip-172-31-47-100 ~]# mkdir deploy_dir
[root@ip-172-31-47-100 ~]# cd deploy_dir/
[root@ip-172-31-47-100 deploy_dir]# mkdir sampleapp
[root@ip-172-31-47-100 deploy_dir]# cd sampleapp/
[root@ip-172-31-47-100 sampleapp]# vi index.html
[root@ip-172-31-47-100 sampleapp]# vi appspec.yml
[root@ip-172-31-47-100 sampleapp]# touch index.html
[root@ip-172-31-47-100 sampleapp]# cat index.html
<html>
<h2> Sample App Version 1 </h2>
</html>
```

```
[root@ip-172-31-47-100 sampleapp]# cat appspec.yml
```

```
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
```

```
[root@ip-172-31-47-100 sampleapp]# mkdir scripts
[root@ip-172-31-47-100 sampleapp]# cd scripts/
[root@ip-172-31-47-100 scripts]# touch httpd_install.sh
[root@ip-172-31-47-100 scripts]# touch httpd_start.sh
[root@ip-172-31-47-100 scripts]# touch httpd_stop.sh
[root@ip-172-31-47-100 scripts]# vi httpd_install.sh
```

```
#!/bin/bash
yum install -y httpd
```

```
[root@ip-172-31-47-100 scripts]# vi httpd_start.sh
```

```
#!/bin/bash
systemctl start httpd
systemctl enable httpd
```

```
[root@ip-172-31-47-100 scripts]# vi httpd_stop.sh
```

```
#!/bin/bash
systemctl stop httpd
systemctl disable httpd
```

```
[root@ip-172-31-47-100 scripts]# cat httpd_install.sh
#!/bin/bash
yum install -y httpd
[root@ip-172-31-47-100 scripts]# cat httpd_start.sh
#!/bin/bash
systemctl start httpd
systemctl enable httpd

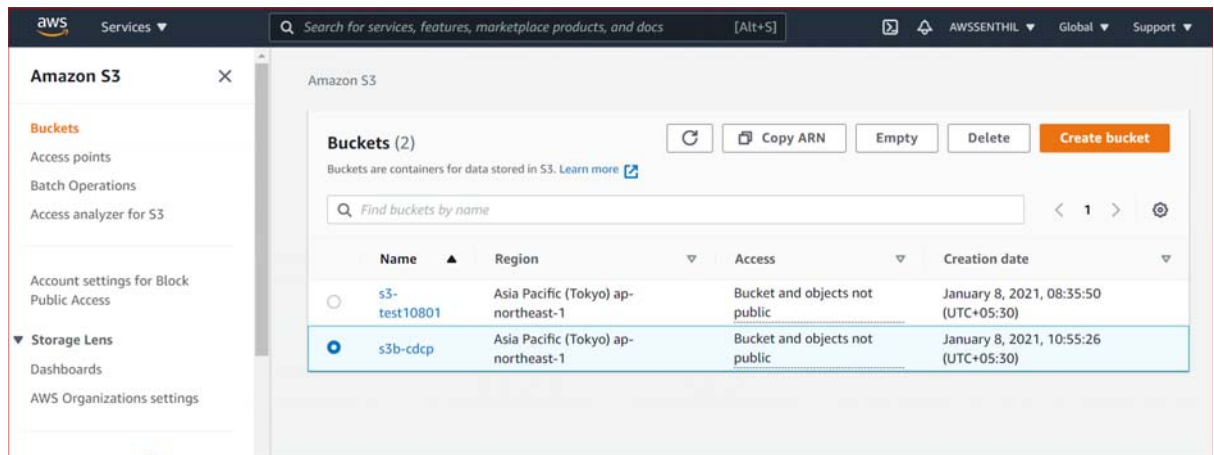
[root@ip-172-31-47-100 scripts]# cat httpd_stop.sh
#!/bin/bash
systemctl stop httpd
systemctl disable httpd

[root@ip-172-31-47-100 scripts]# █
```

Change execute permission to all files

```
[root@ip-172-31-47-100 sampleapp]# cd scripts/
[root@ip-172-31-47-100 scripts]# ls -lr
total 12
-rw-r--r-- 1 root root 58 Jan  8 05:18 httpd_stop.sh
-rw-r--r-- 1 root root 58 Jan  8 05:17 httpd_start.sh
-rw-r--r-- 1 root root 33 Jan  8 05:14 httpd_install.sh
[root@ip-172-31-47-100 scripts]# chmod 755 *
[root@ip-172-31-47-100 scripts]# ls -lr
total 12
-rwxr-xr-x 1 root root 58 Jan  8 05:18 httpd_stop.sh
-rwxr-xr-x 1 root root 58 Jan  8 05:17 httpd_start.sh
-rwxr-xr-x 1 root root 33 Jan  8 05:14 httpd_install.sh
[root@ip-172-31-47-100 scripts]# █
```

7.S3 Bucket Created- Storage container (S3 Bucket), the code Repository



-Block all Public Access – Enabled
Version control Disabled.*****

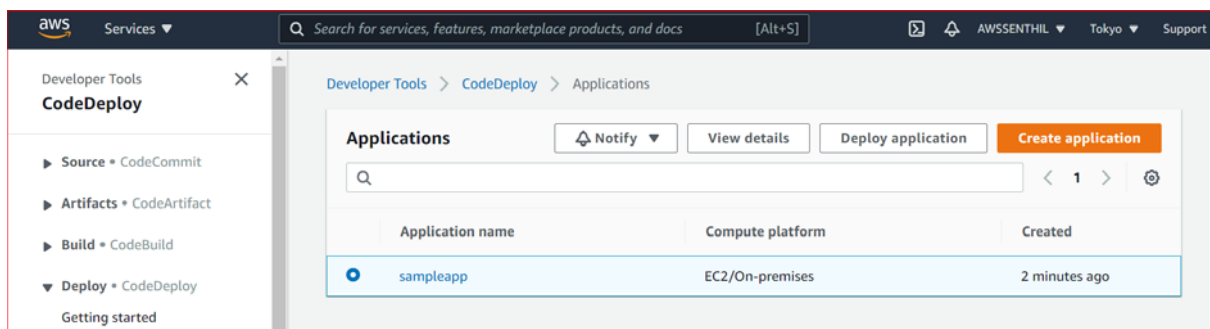
8.Using Code Deploy Create Application & Push the code to S3 bucket -Application Machine

The application creation command should be execute from **\sampleapp** folder.
 the **appspec** file, the **codedeploy process execution** file located here only.

```
[root@ip-172-31-47-100 sampleapp]# pwd
/root/deploy_dir/sampleapp
[root@ip-172-31-47-100 sampleapp]#
```

```
[root@ip-172-31-47-100 sampleapp]# aws deploy create-application --application-name sampleapp
{
  "applicationId": "09fec7cb-f082-47e2-9b63-04aeaa5283f0"
}
[root@ip-172-31-47-100 sampleapp]#
```

Sampleapp application created in CODE DEPLOY service :



```
# aws deploy create-application --application-name sampleapp
```

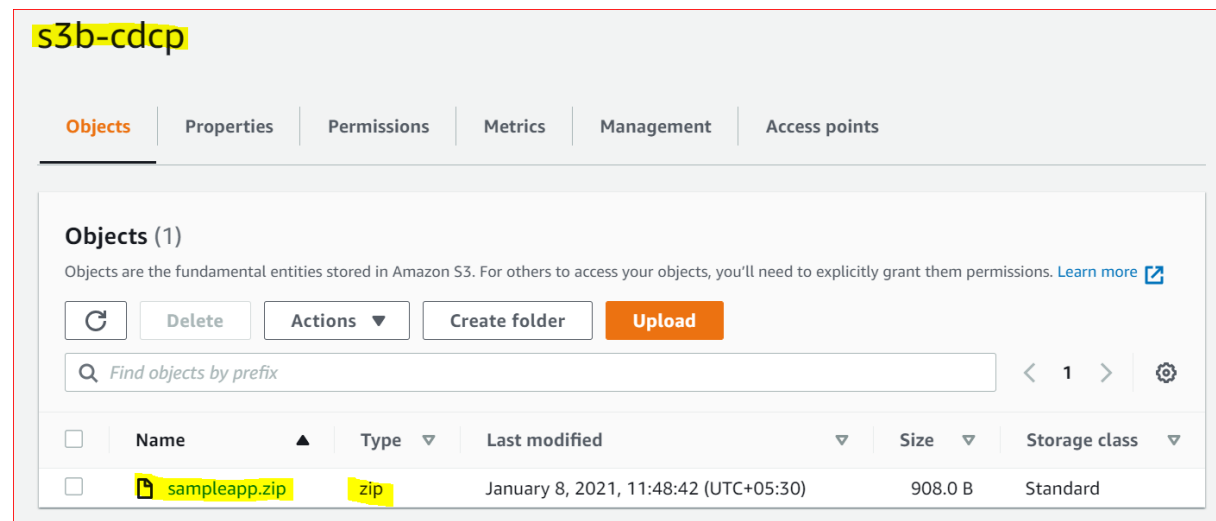
```
Aws <command> deploy <process> create-application <process to-do> --
application-name sampleapp<name of the process>
```

Now upload the code to S3 by the executing the command below. Directory of execution is important

The files what are present in the sampleapp folder will be zipped and stored into S3 bucket what we created for store the deployment process.

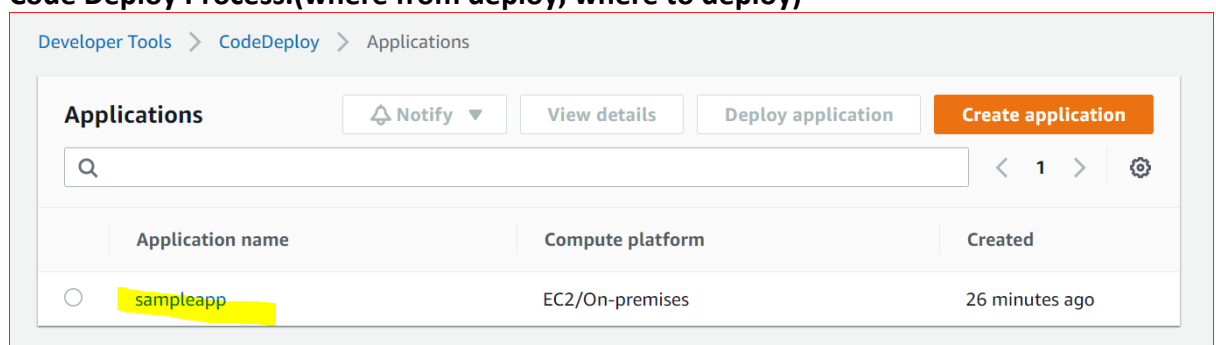
```
# aws deploy push --application-name sampleapp --s3-location s3://s3b-cdcp/sampleapp.zip
```

```
[root@ip-172-31-47-100 sampleapp]# aws deploy push --application-name sampleapp --s3-location s3://s3b-cdcp/sampleapp.zip
To deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=s3b-cdcp,key=sampleapp.zip,bundleType=zip,eTag=cedd80476ee2b0ba0f020fc0c1518d9b --deployment-group-name <deployment-group-name> --deployment-config-name <deployment-config-name> --description <description>
[root@ip-172-31-47-100 sampleapp]#
```



The files are successfully moved into S3 code repository.

10. Code Deploy Process.(where from deploy, where to deploy)



Create Deployment Group → instruct Where to deploy

Deployment group name

Enter a deployment group name

CDeploy-Grp

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.

Q

arn:aws:iam::903889376453:role/cdrole

X

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

☒ 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	
<div><div>Q</div><div>AppName</div><div>X</div></div>	<div><div>Q</div><div>SampleApp</div><div>X</div></div>	<div>Remove tag</div>

Load Balancer – Disable

Create Deployment – Where from to take for deployment. Source location instruction

Create deployment

Deployment settings

Application

sampleapp

Deployment group

CDeploy-Grp

Compute platform

EC2/On-premises

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://s3b-cdcp/sampleapp.zip?eTag=cedd80476ee2b0ba0f020fc0c1518d9b

s3://bucket-name/folder/object.[zip|tar|tgz]

Revision file type

.zip

S3 URI

s3://s3b-cdcp/sampleapp.zip

Amazon resource name (ARN)

arn:aws:s3::s3b-cdcp/sampleapp.zip

Entity tag (Etag)

cedd80476ee2b0ba0f020fc0c1518d9b

Deployment Created.

Success
Deployment created

d-GP7O92Y78 Stop deployment

Deployment status

Installing application on your instances

0 of 1 instances updated in progress 0%

Deployment details

Application sampleapp	Deployment ID d-GP7O92Y78	Status In progress
Deployment configuration CodeDeployDefault.AllAtOnce	Deployment group CDeploy-Grp	Initiated by User action
Deployment description -		

Revision details

Revision location s3://s3b-cdcp/sampleapp.zip?eTag=cedd80476ee2b0ba0f020fc0c1518d9b	Revision created 1 hour ago	Revision description Uploaded by AWS CLI 2021-01-08T06:18:40.874445 UTC
--	--------------------------------	--

11. Deploymnet

Deployment status

Installing application on your instances

100%

1 of 1 instances updated ✓ Succeeded

Deployment details

Application	Deployment ID	Status
sampleapp	d-ADMHSU788	✓ Succeeded
Deployment configuration	Deployment group	Initiated by
CodeDeployDefault.AllAtOnce	mygrp	User action
Deployment description		
-		

Revision details

Revision location	Revision created	Revision description
s3://s3b-cdcp/sampleapp.zip?versionId=vk5ioece7jnZu1oUZ5LNE3Oai9Du2iuS&eTag=4361ef3ee3ccec0cf126202edf686d2f	7 minutes ago	Uploaded by AWS CLI 2021-01-08T17:38:02.728739 UTC

Deployment lifecycle events

< 1 > ⚙

Instance ID	Duration	Status	Most recent event	Events	Start time	End time
i-0e3b448d8cabcccb1	8 seconds	✓ Succeeded	ValidateService	View events	Jan 8, 2021 11:13 PM (UTC+5:30)	Jan 8, 2021 11:13 PM (UTC+5:30)

Testing the Configuration

Now access the public Ip address fo the webserver from the browser and see that it is working

← → ↻ Not secure | 54.238.214.233

Apps AWS DEV-OPPS DC FREE Udemy DevO... AWS Tutorial | Ama...

Sample App Version 1

Complete the Deployment. Once the Deployment process is completed

AWS Code Pipeline

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

Pipeline1

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-northeast-1-Pipeline1

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

s3b-cdcp

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

In the Add Build Stage, Click *Skip Build Stage* button and then confirm the skip.

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 (Tokyo)

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.

mygrp

In the Deploy stage Enter the value like below and Click Next.

Success
Congratulations! The pipeline Pipeline-1 has been created.

Developer Tools > CodePipeline > Pipelines > Pipeline-1

Pipeline-1 Notify Edit Stop execution Clone pipeline Release change

Source Succeeded
Pipeline execution ID: 14139b2f-1d06-45b3-b48d-5543ae5cfc98

Source Amazon S3
Succeeded
~ 1 minute ago

Source: Amazon S3 version id: vk5ioece7jnZu1oUZ5LNE3Oai9Du2luS

✓
✓

[root@ip-172-31-42-160 sampleapp]# zip -r ../sampleapp.zip .

The zip file is available under deploy_dir

```
root@ip-172-31-42-160 deploy_dir]# pwd
root/deploy_dir
root@ip-172-31-42-160 deploy_dir]# ls -la
total 4
-rwxr-xr-x 3 root root 44 Jan 8 18:24 .
-r-xr-x--- 5 root root 133 Jan 8 17:07 ..
-rwxr-xr-x 3 root root 58 Jan 8 18:09 sampleapp
-rw-r--r-- 1 root root 1337 Jan 8 18:24 sampleapp.zip
root@ip-172-31-42-160 deploy_dir]#
```

Copy the sampleapp.zip to s3

```
[root@ip-172-31-42-160 deploy_dir]# aws s3 cp sampleapp.zip s3://s3b-cdcp
upload: ./sampleapp.zip to s3://s3b-cdcp/sampleapp.zip
[root@ip-172-31-42-160 deploy_dir]#
```

s3b-cdcp

Objects Properties Permissions Metrics Management Access points

Objects (1)
Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

☐ List versions

< 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	sampleapp.zip	zip	January 9, 2021, 00:00:58 (UTC+05:30)	1.3 KB	Standard

Pipeline

Deployment history

< 1 >

	Deployment Id	Status	Deployment type	Compute platform	Application	Deployment group	Revision location	Initiating event	Start time	End time
<input type="radio"/>	d-IXUVXW888	Succeeded	In-place	EC2/On-premises	sampleapp	mygrp	s3://code...	User action	Jan 9, 2021 12:05 AM (UTC+5:30)	Jan 9, 2021 12:05 AM (UTC+5:30)
<input type="radio"/>	d-0TDGIA988	Succeeded	In-place	EC2/On-premises	sampleapp	mygrp	s3://code...	User action	Jan 9, 2021 12:01 AM (UTC+5:30)	Jan 9, 2021 12:01 AM (UTC+5:30)
<input type="radio"/>	d-6YMY70988	Succeeded	In-place	EC2/On-premises	sampleapp	mygrp	s3://code...	User action	Jan 8, 2021 11:48 PM (UTC+5:30)	Jan 8, 2021 11:48 PM (UTC+5:30)
<input type="radio"/>	d-ADMH5U788	Succeeded	In-place	EC2/On-premises	sampleapp	mygrp	s3://s3b-c...	User action	Jan 8, 2021 11:12 PM (UTC+5:30)	Jan 8, 2021 11:13 PM (UTC+5:30)

Changes Happened Successfully.....

← → ↻ ⚠ Not secure | 54.238.214.233

Apps AWS DEV-OPPS DC FREE Udemy DevO... AWS Tutorial

Sample App Version 1

Sample App Version 2 - Pipeline Modified