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 DEPLOY**MENT 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.

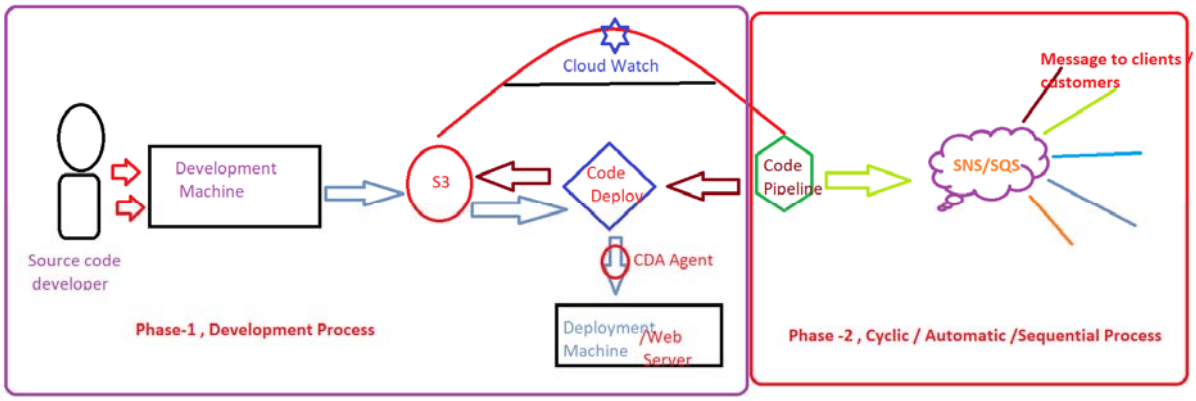
3. This process 1 & 2 may have happened again and again, and the repeated process has watch / monitored by a monitoring tool **Cloud WATCH**. If The code refreshed / reinitiate /newcode update like process , the monitoring tool will check & inform to **the continues / scheduled deployment tool Code PIPELINE**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

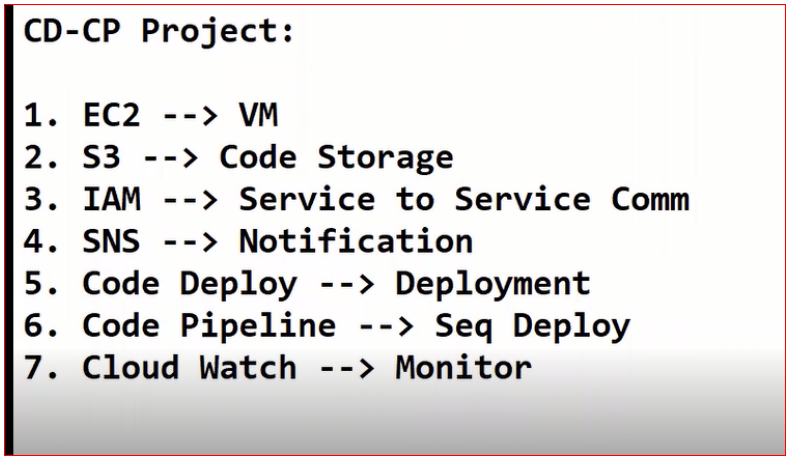
5.Deployment Success , the **Code PIPELINE** send notifications about the changes to clients/customers through SNS messaging / mailing service.

Process – Phase 1:



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

Phase 2: Cyclic / Automatic / Sequential Process.



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 perfet
   #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

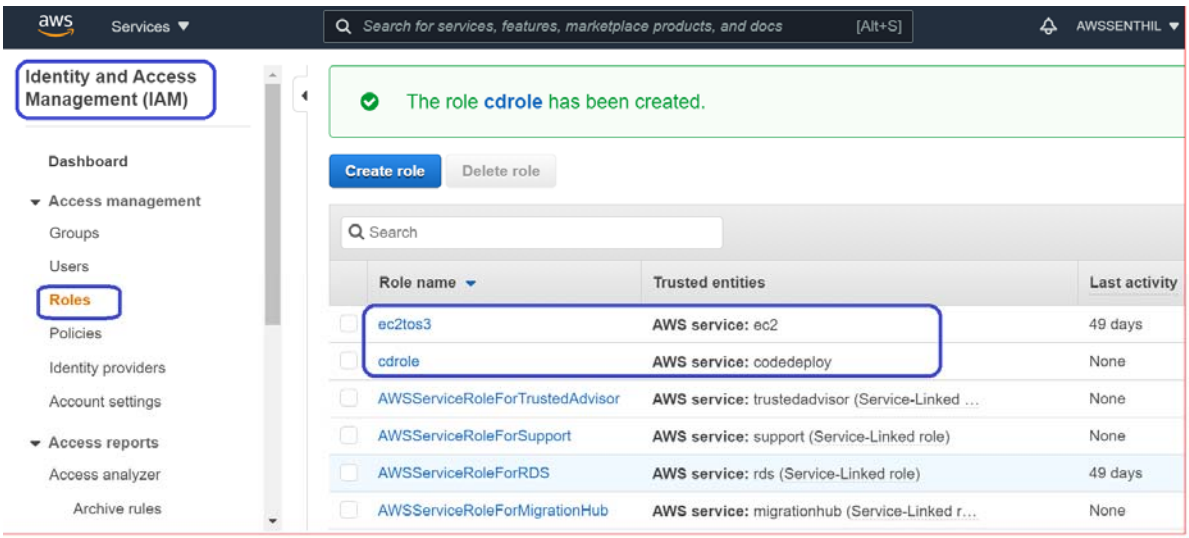
   Change index.html content
   Load & Push the new index.html to S3

9. SNS Messaging Service

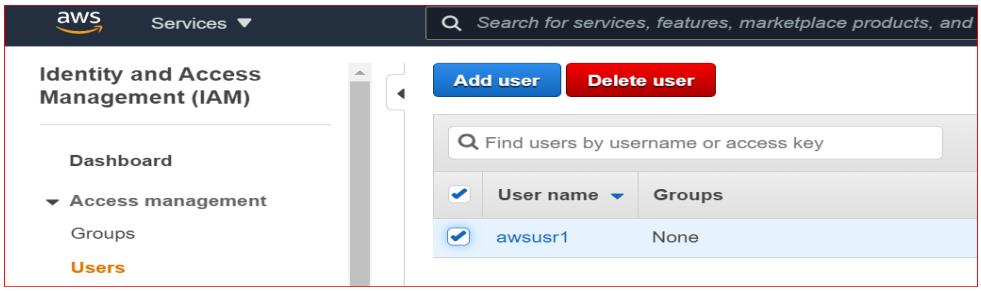
Process #1 :

1. Create IAM – For Roles, 1.EC2-S3

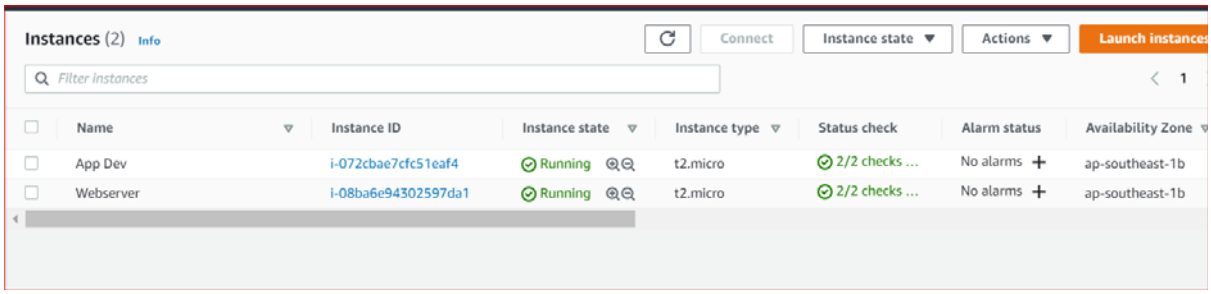
   2.Code Deploy

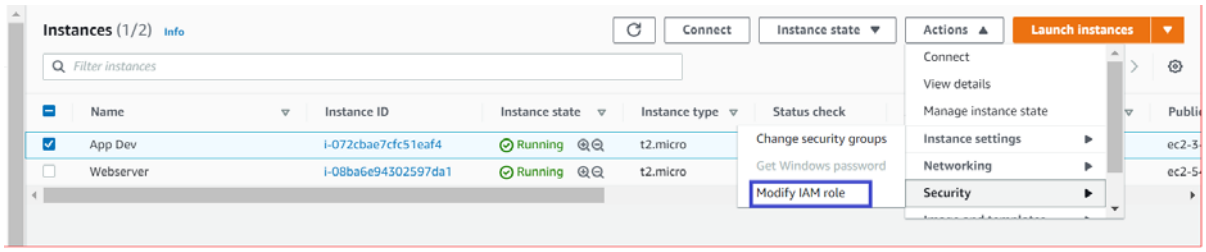


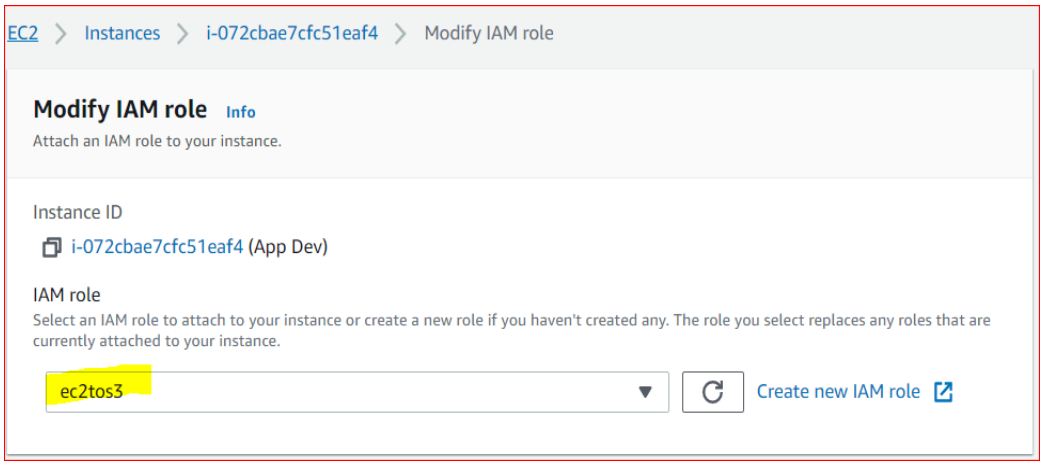
2.Create IAM – User Management- User Creation for Developer



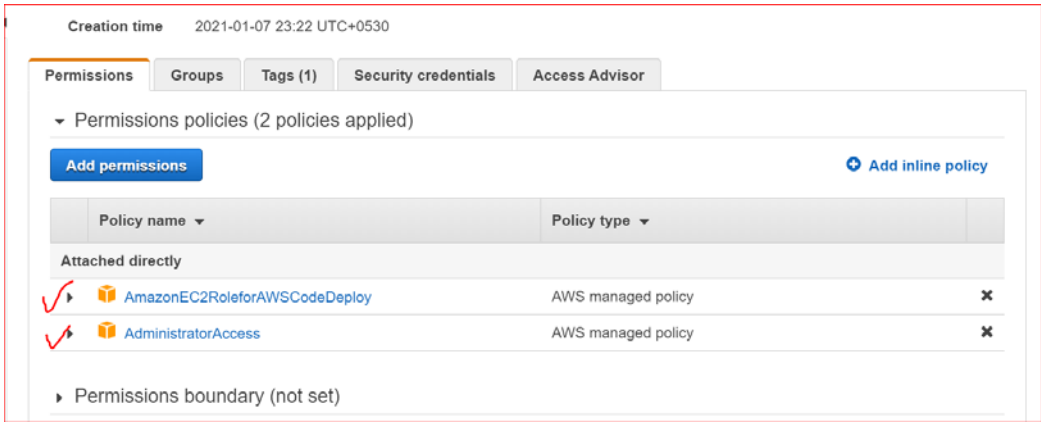
3 . EC2 - For 1. Create Development Machine
2. Create Deployment Machine / Webserver







Assign Roles



4.User Connect with App Developer Machine in Putty

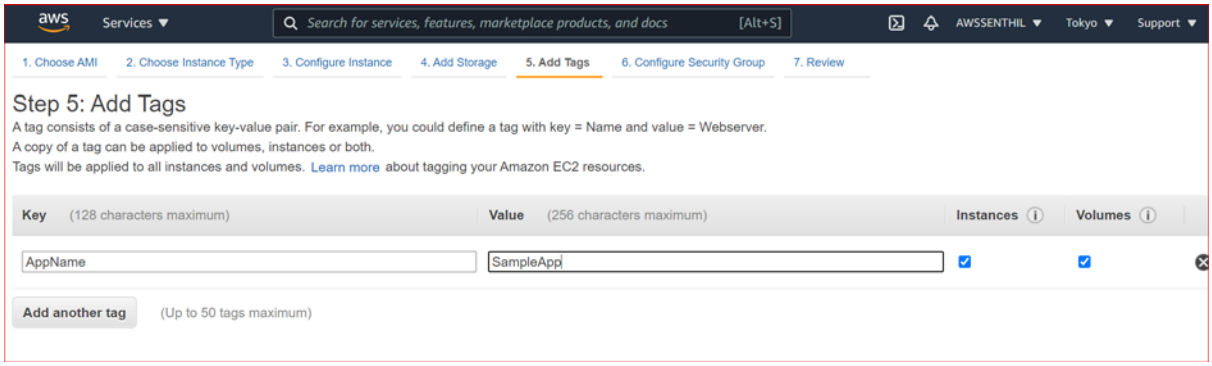
```
[root@ip-172-31-47-100 ~]# aws configure
AWS Access Key ID [****************F3F2]: AKIA5E5ABKTCXOVHF3F2
AWS Secret Access Key [****************blCP]: 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
Default region name [ap-northeast-1]: ap-northeast-1
Default output format [json]: json
[root@ip-172-31-47-100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

```
[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 member ship for the EC2 instance is decided by this Tag. We have used AppName Tag with value SampleApp
Role : ec2-s3 full acess, 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
```

-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
```

```
ependencies Resolved

================================================================================
Package              Arch        Version        Repository                                                     Size
================================================================================
nstalling:
codedeploy-agent     noarch      1.3.1-1880     /codedeploy-agent-1.3.1-1880.noarch.tmp-20210108-32438-ktncw7   11 M
ransaction Summary
================================================================================
nstall  1 Package

otal size: 11 M
nstalled size: 11 M
ownloading packages:
unning transaction check
unning transaction test
ransaction test succeeded
unning transaction

re hook : 1
hecking the ruby version.
hecking if there is already a process named codedeploy-agent running.
 Installing : codedeploy-agent-1.3.1-1880.noarch                                  1/1

ost hook : 1
heck if there is a codedeployagent config file.
tart codedeploy-agent in post hook if this is a first install.
 Verifying  : codedeploy-agent-1.3.1-1880.noarch                                  1/1

nstalled:
 codedeploy-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

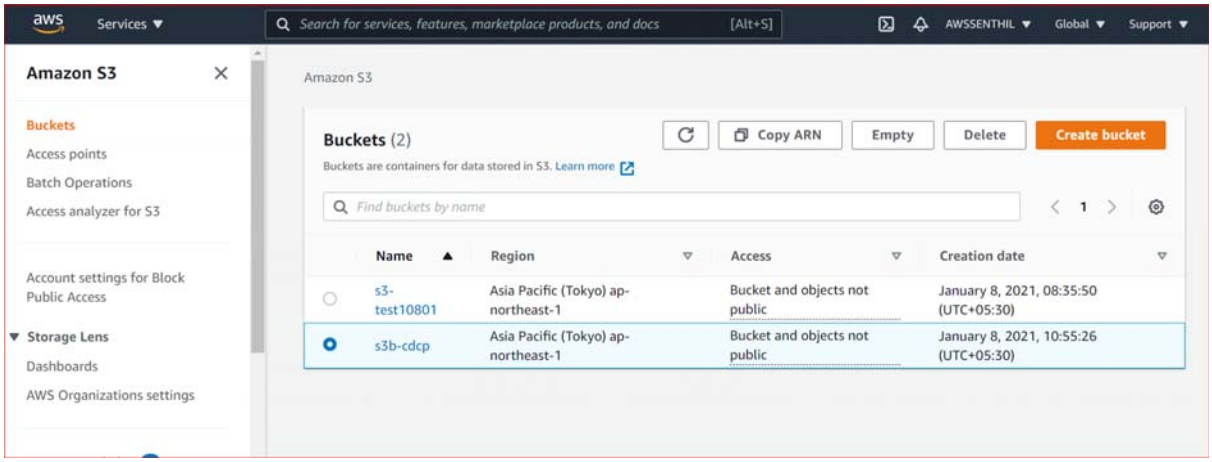
```
#!/bin/bash
systemctl stop httpd
systemctl disable httpd
```

```
[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*

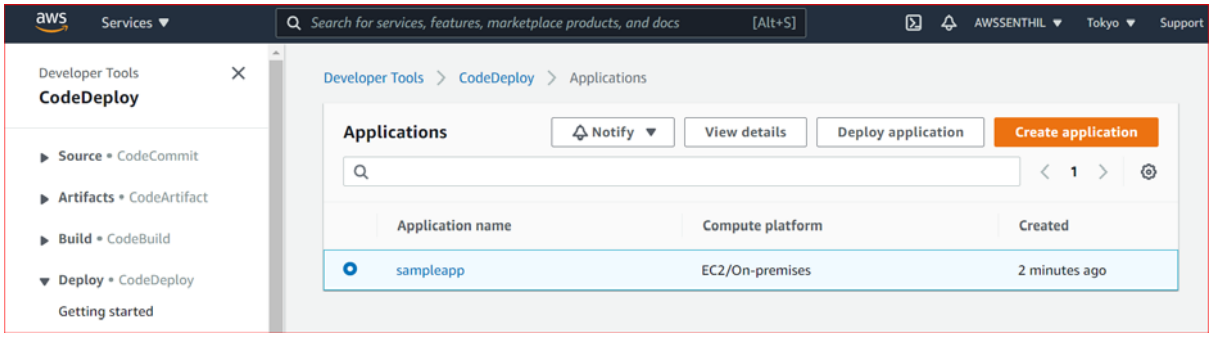
-Application Machine

The application creation command shoud be execute from **\sampleap** folder.
the **appspec file, the codedepoy 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 :



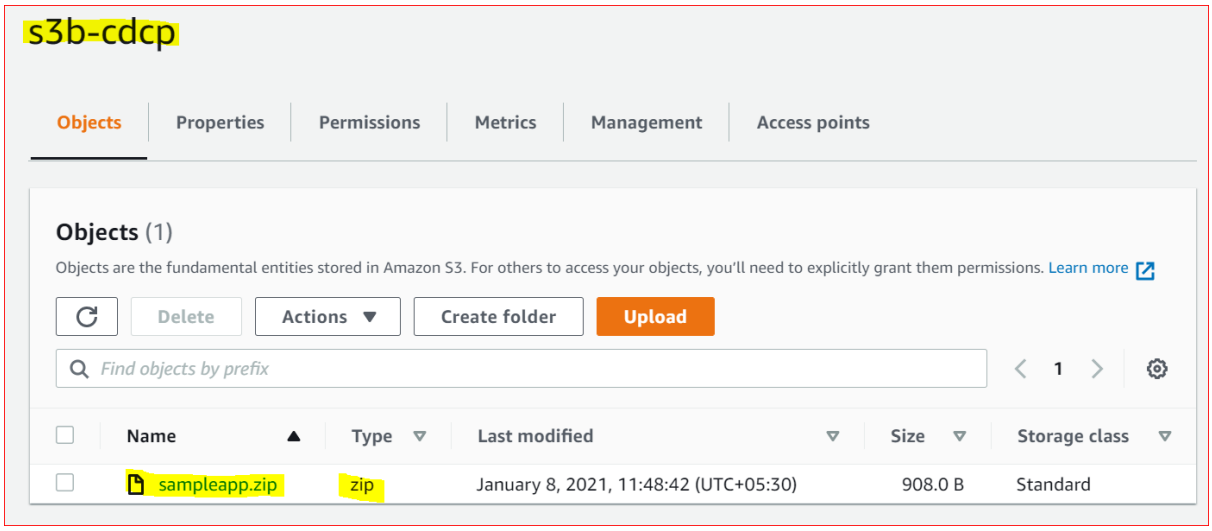
 # aws deploy create-application --application-name sampleapp

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 foler will be zipped and stored into S3 bucket what we created for store the deployment process.

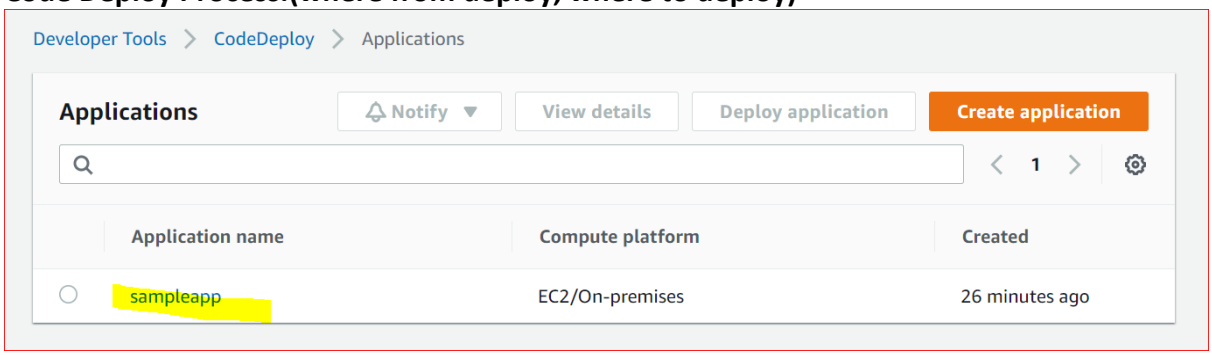
aws deploy push --application-name sampleapp --s3-location s3://**s3b-cdcp/sampleapp.zip**

```
[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]#
```

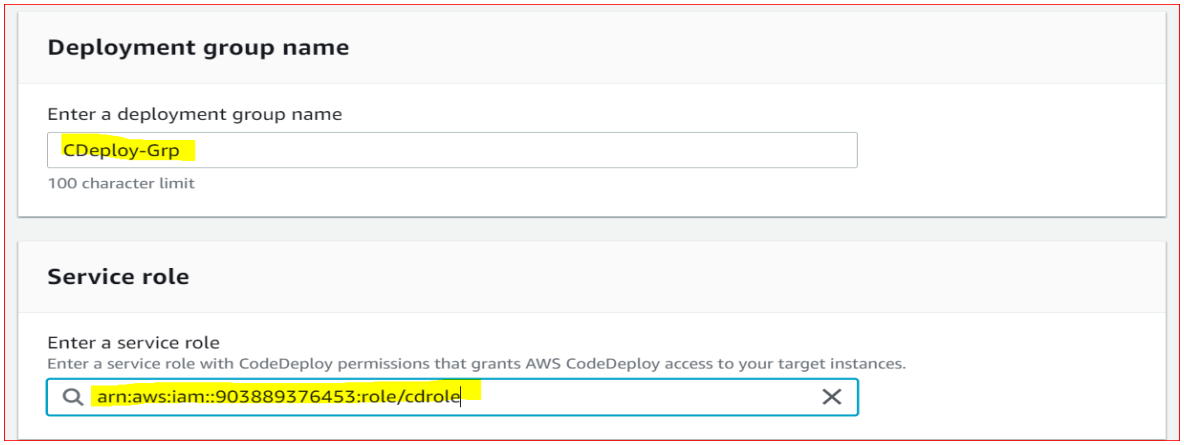


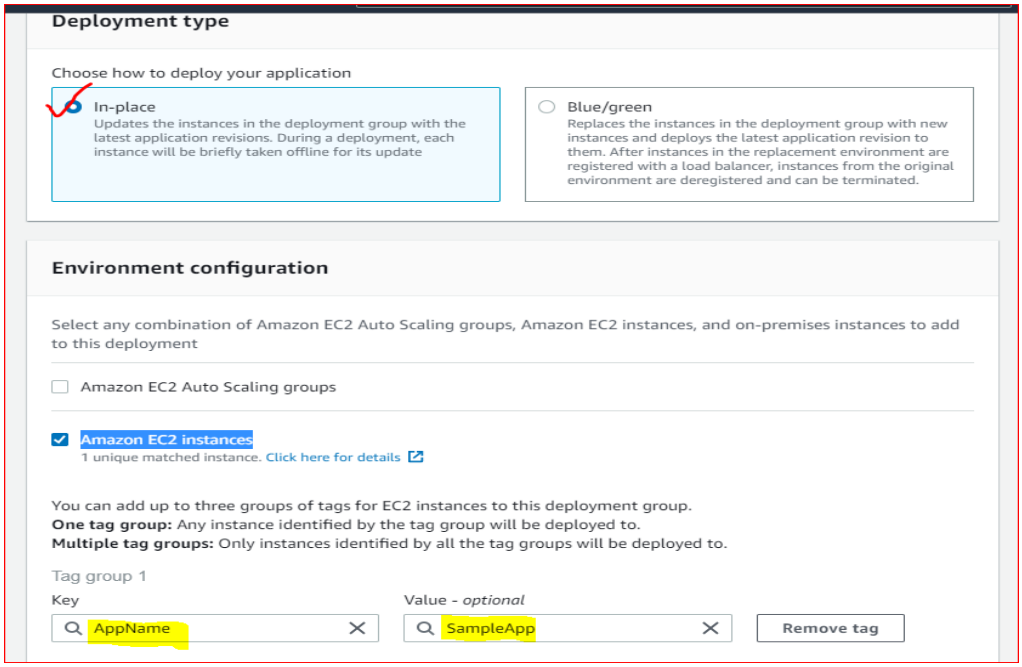
The files are successfully moved into S3 code repository**.**

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



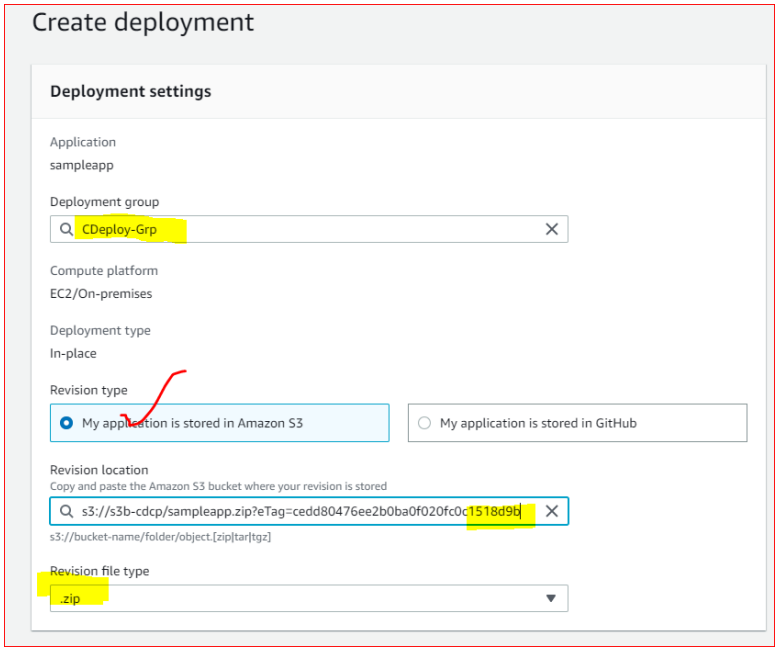
Create Deployment Group →instruct Where to deploy

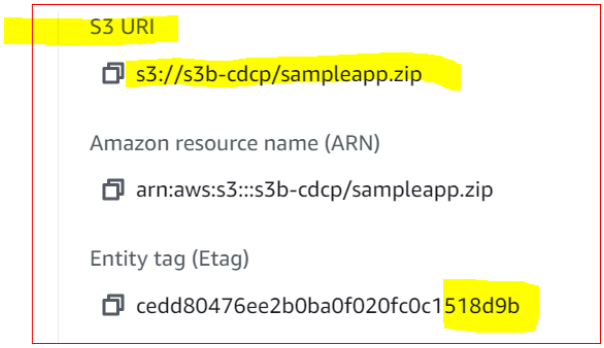




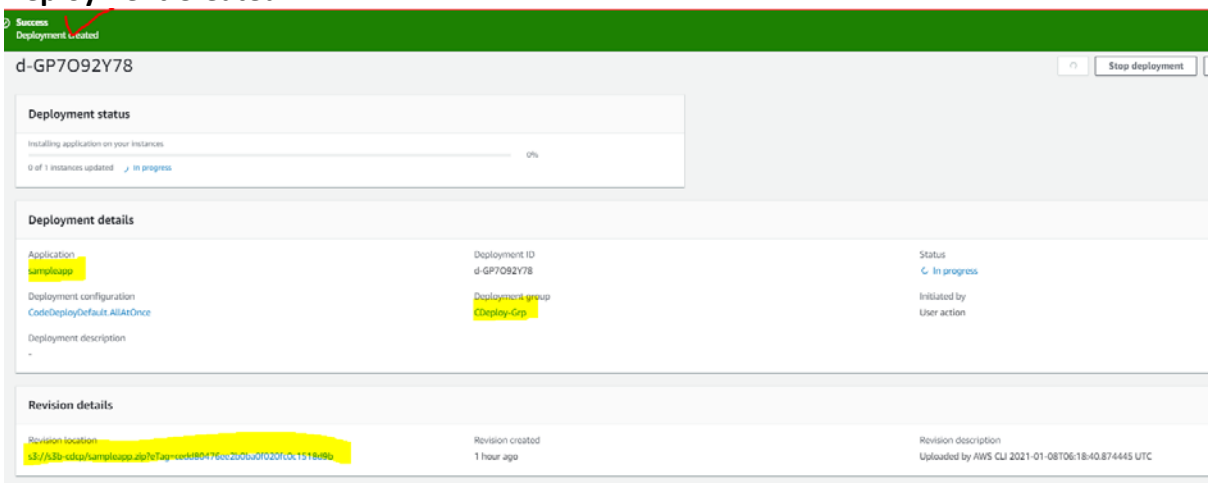
Load Balancer – Disable

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

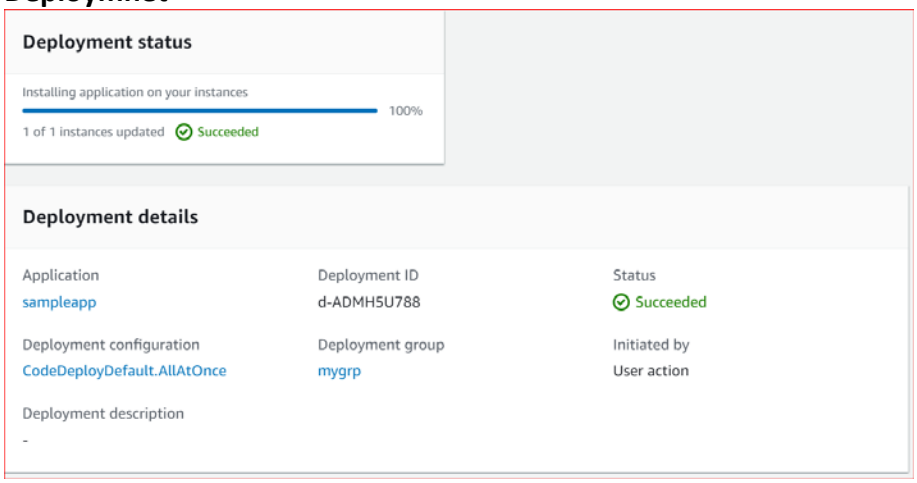


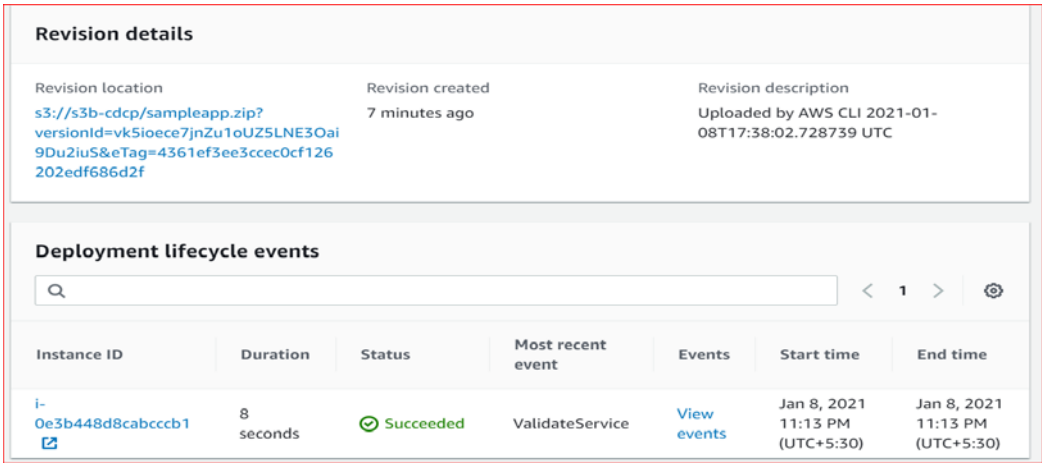


Deployment Created.



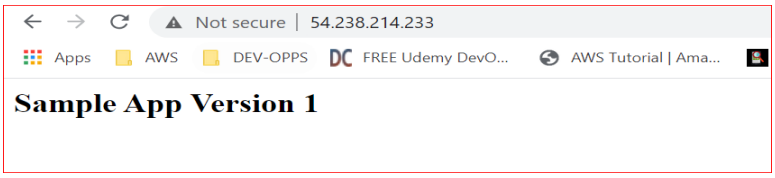
11. Deploymnet





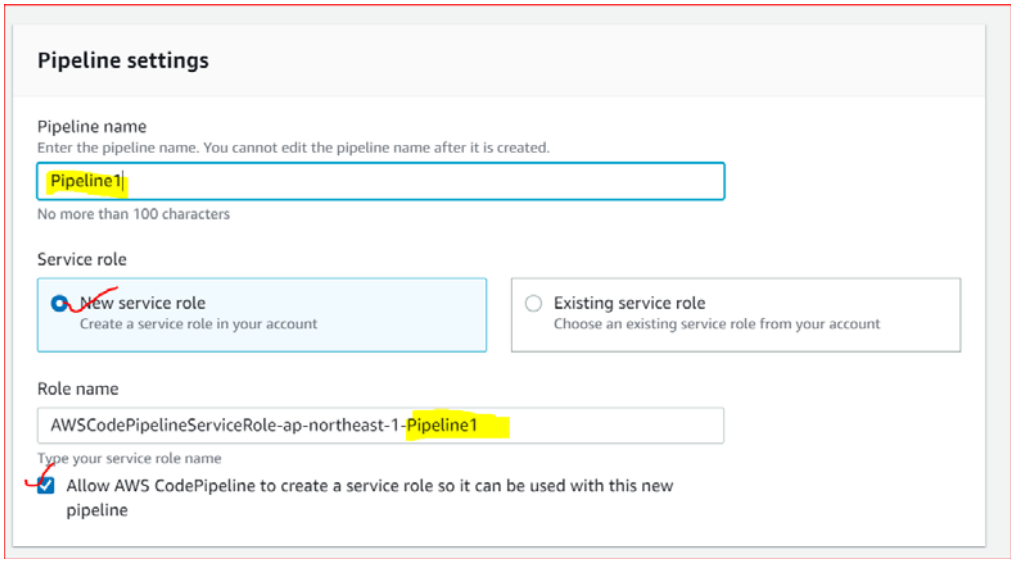
Testing the Configuration

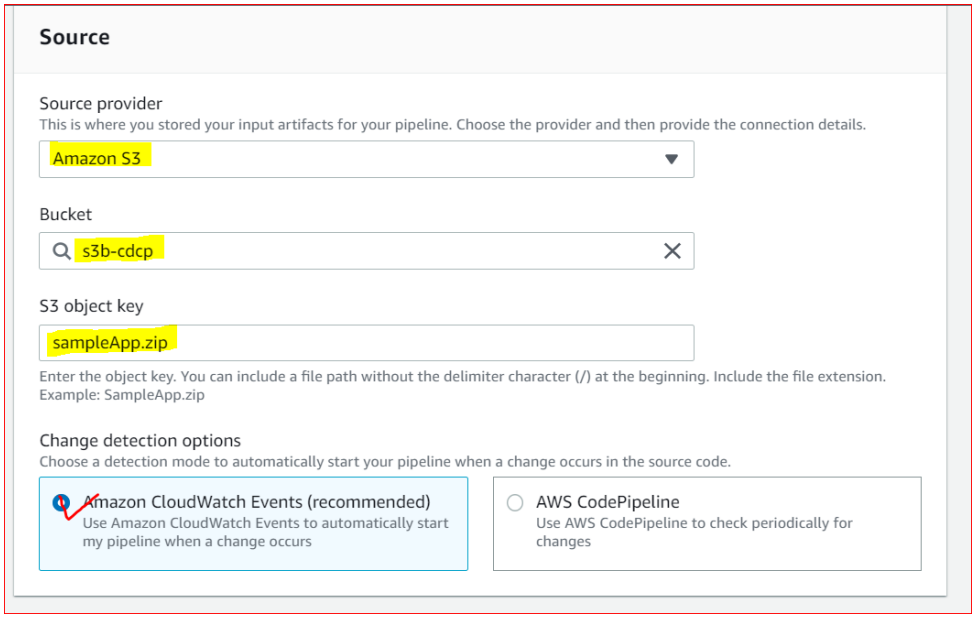
Now access the public Ip address fo the webserver from the browser and see that it is working



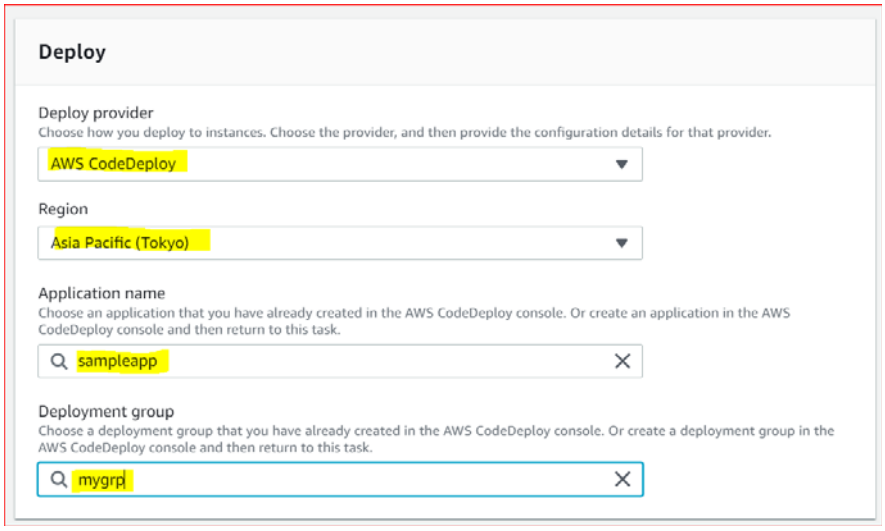
Complete the Deployment. Once the Deployment process is completed …..

AWS Code Pipeline

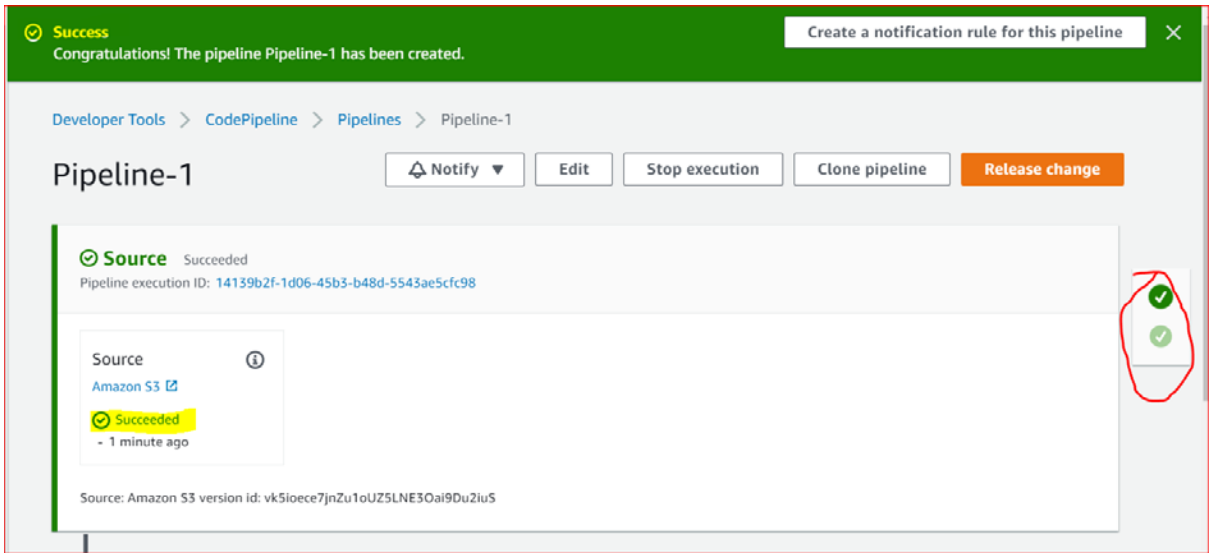




In the Add Build Stage, Click *Skip Build Stage* button and then confirm the skip.



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



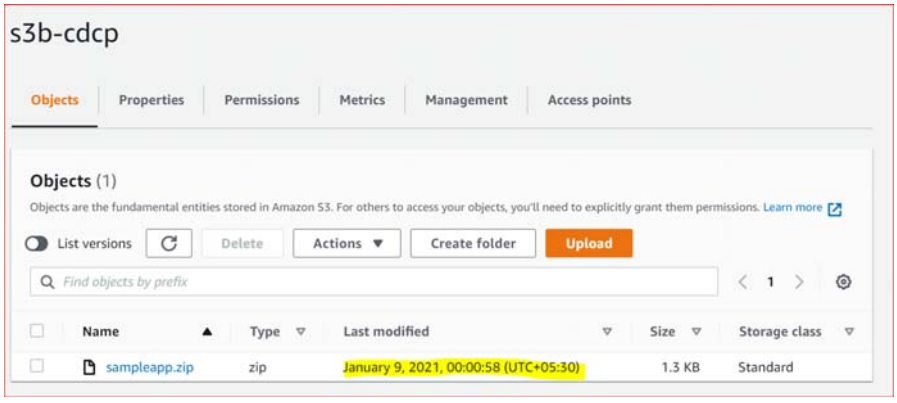
[root@ip-172-31-42-160 sampleapp]# zip -r ../sampleapp.zip .

The zip file is available under deploy_dir

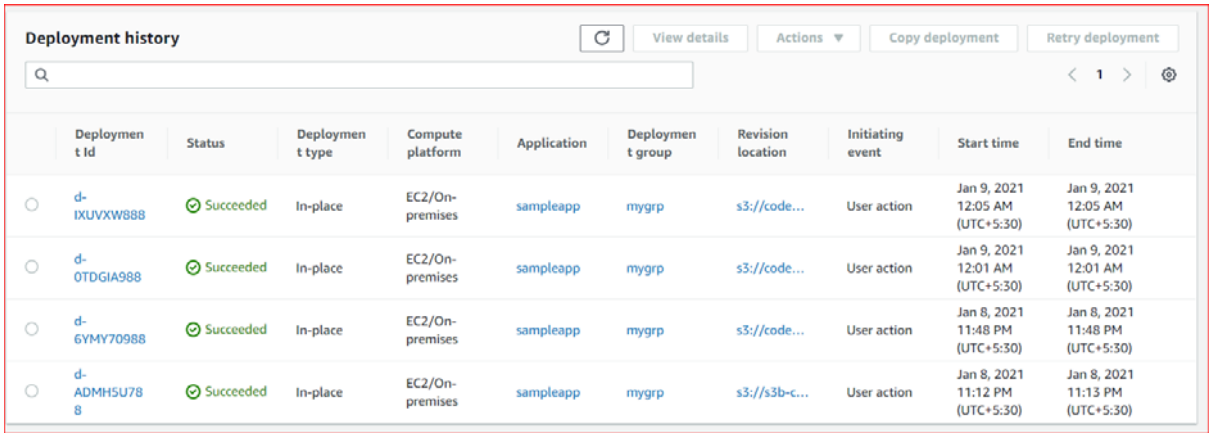
```
root@ip-172-31-42-160 deploy_dir]# pwd
root/deploy_dir
root@ip-172-31-42-160 deploy_dir]# ls -la
otal 4
rwxr-xr-x 3 root root   44 Jan  8 18:24 .
r-xr-x--- 5 root root  133 Jan  8 17:07 ..
rwxr-xr-x 3 root root   58 Jan  8 18:09 sampleapp
rw-r--r-- 1 root root 1337 Jan  8 18:24 sampleapp.zip
root@ip-172-31-42-160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



Pipeline



Changes Happened Successfully……