# CODE PIPELINE:

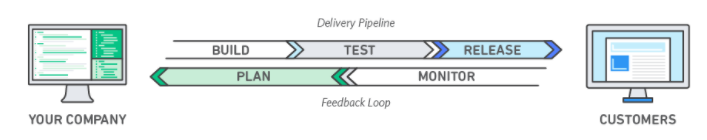
**AWS CodePipeline is a continuous delivery service that enables you to model, visualize, and automate the steps required to release your software. ... AWS CodePipeline then builds, tests, and deploys your application according to the defined workflow every time there is a code change**

# CODE DEPLOY:

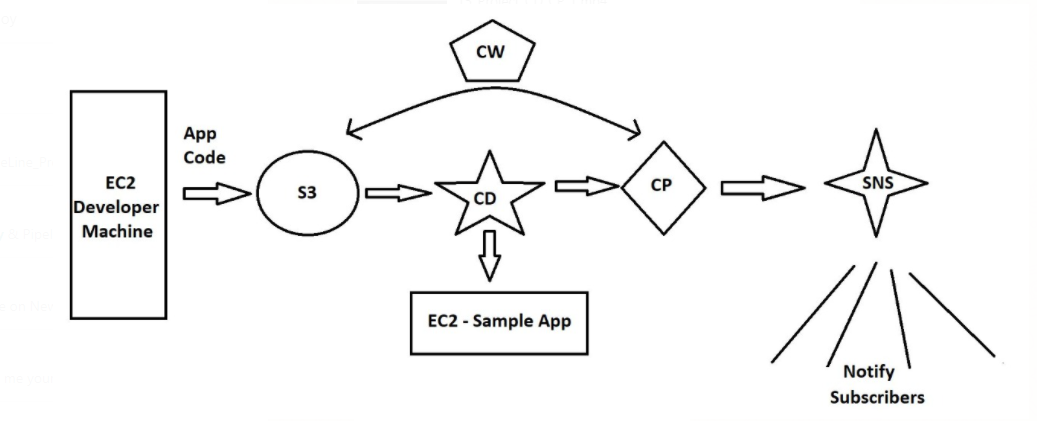
**AWS CodeDeploy is a service that automates code deployments to Elastic Compute Cloud (EC2) and on-premises servers. Accelerating how fast a developer can release code allows him to release new features for an application faster and avoid deployment errors in complex applications.**

# OBJECTIVE TO THE CODE DEPLOY AND CODE PIPELINE:

**Complete CI/CD with AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline**

****

# CODE DEPLOY AND CODE PIPELINE PROJECT:

****

# CODE DEPLOY AND CODE PIPELINE PROJECT STEPS:

**CodeDeploy 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, for scheduling of deployment you may have to use AWS CodePipeline also.**

***Application:* A CodeDeploy application can be defined from AWS CodeDeploy web console.**

***Revision:* Represents the code need to be deployed on EC2 instance.**

***Appspec file*: This contains the instruction to CodeDeploy, like copying of files, executing the scripts etc during the code deployment process. It is present in the root directory of unzipped code with name appspec.yml.**

***Deployment Group:*Represent set of machines of Lambda function where code has to be deployed.**

***Deployment:* The process of deployment.**

**Setup in Brief:**

**I have used two EC2 instance of AMZ2 Linux. First one is the web server we will be configuring, also called CodeDeploy agent. Second EC2 machine is supposed to use by developer where the codes are programmed. The names of the resources in the experiment are arbitrary and may name the resources your own.**

1. **Create IAM Roles for EC2-S3-CodeDeploy access.**
2. **Create IAM user account for developer**
3. **Install and prepare the CodeDeploy agent on webserver.**
4. **Create the code from Developer machine**
5. **Create Codedeploy Application and Push the code to S3 bucket from Developer machine**
6. **Create Deployment Group to include web server**
7. **Create Deployment to push the code to the Webserver**
8. **Test the website configuration**

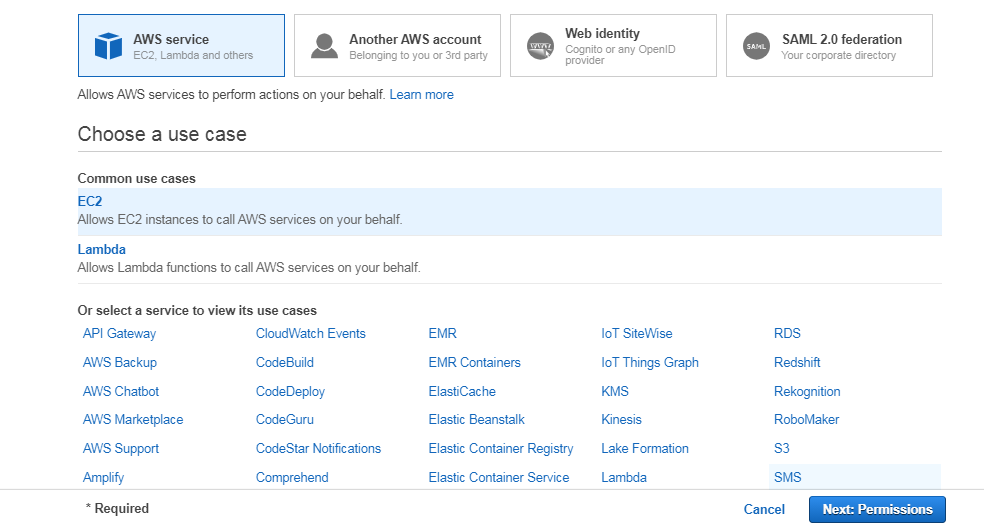
**SERVICES REQUIRED: EC2,S3,CD,CP,CW,IAM,SNS**

**Step 1: Create IAM Roles for EC2-S3-CodeDeploy access:**

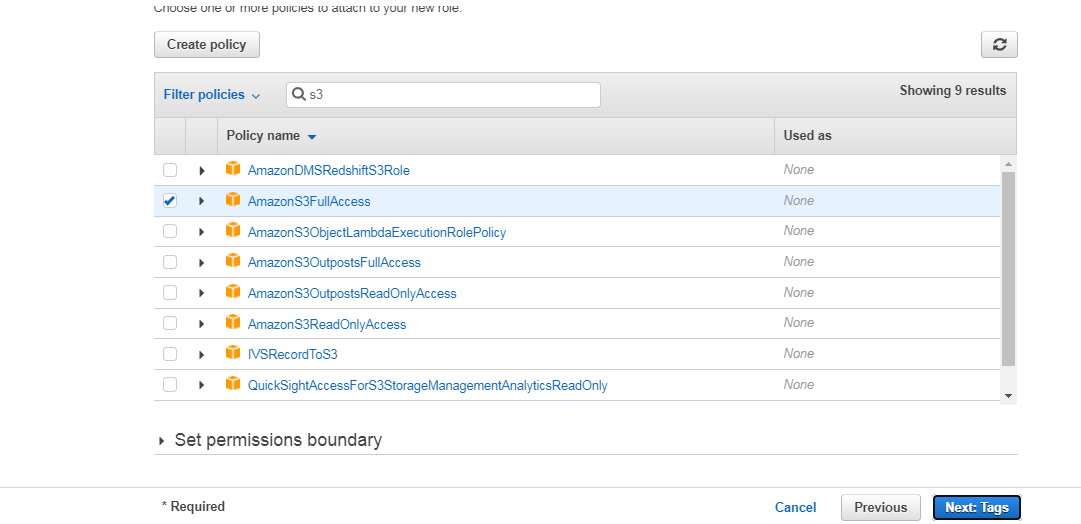
**a) s3 to ec2 full access: s3-ec2-full**

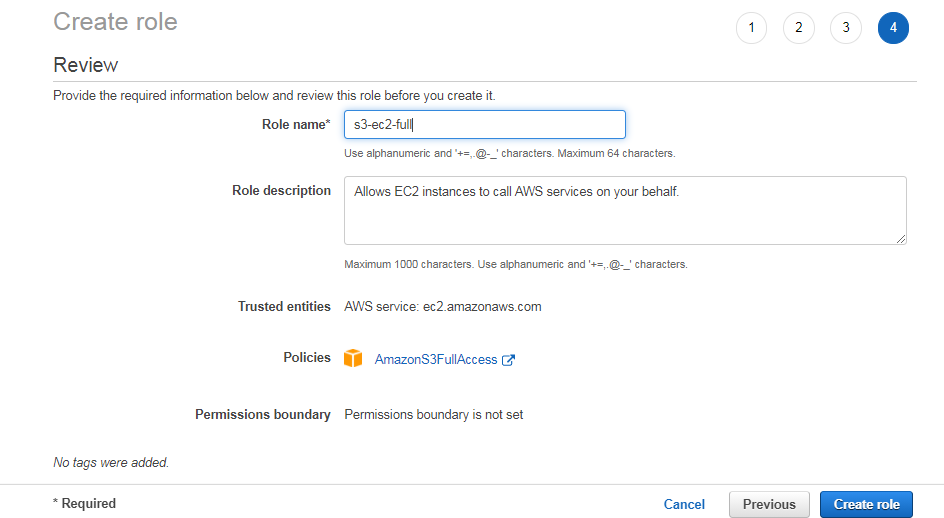
**select IAM service**

**goto Roles: select EC2 and click next Permissions**

****

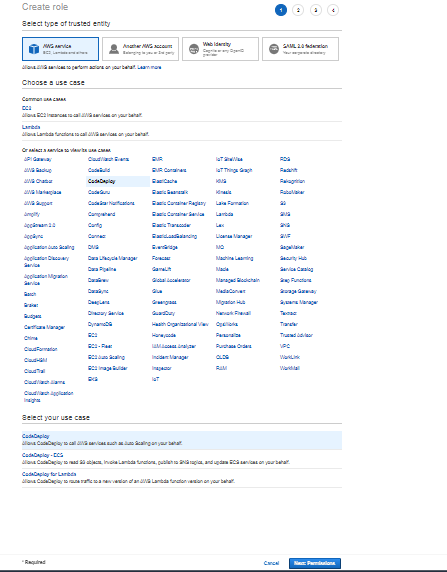
**Then select S3fullaccess policy and click Next: Tags**

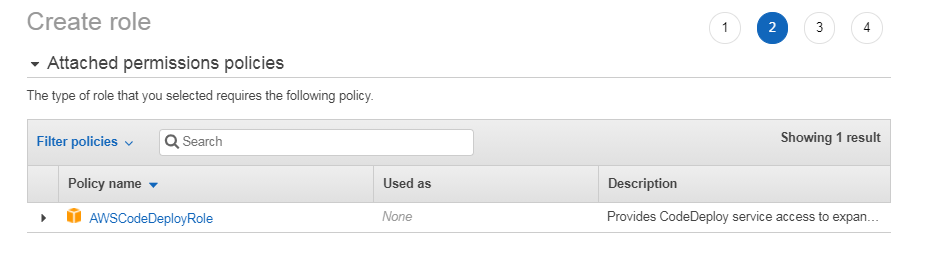
****

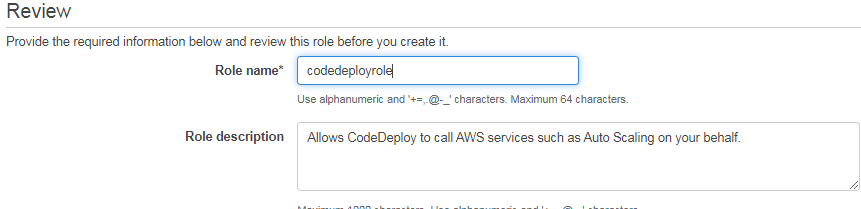
****

**b) Create an Code-Deploy role**

**Select service CodeDeploy and use case CodeDeploy**

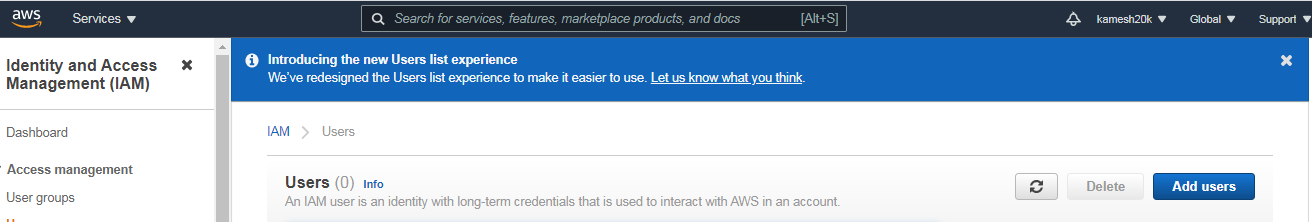
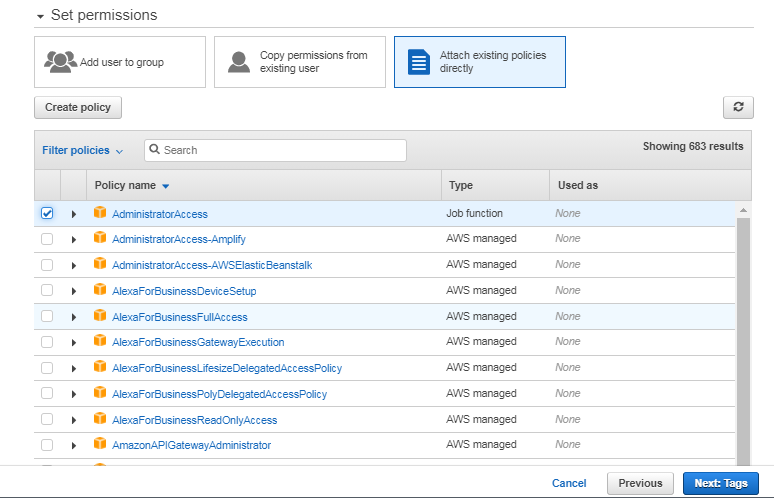
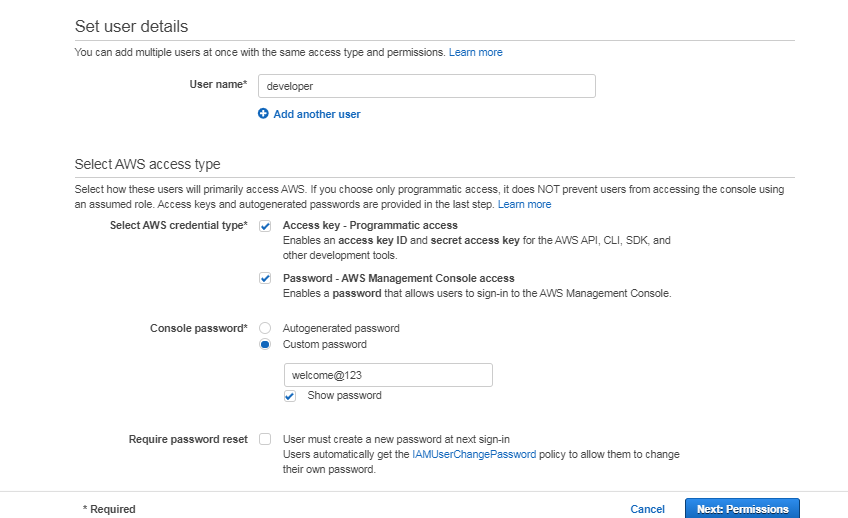
****

****

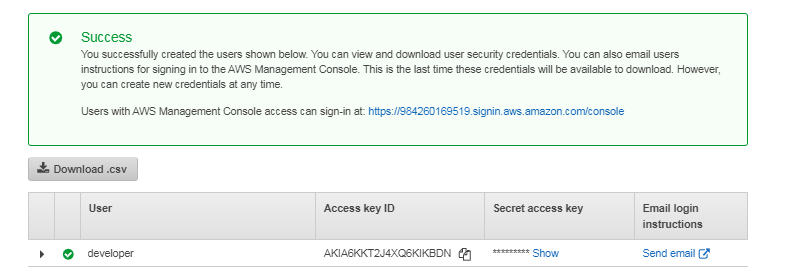
****

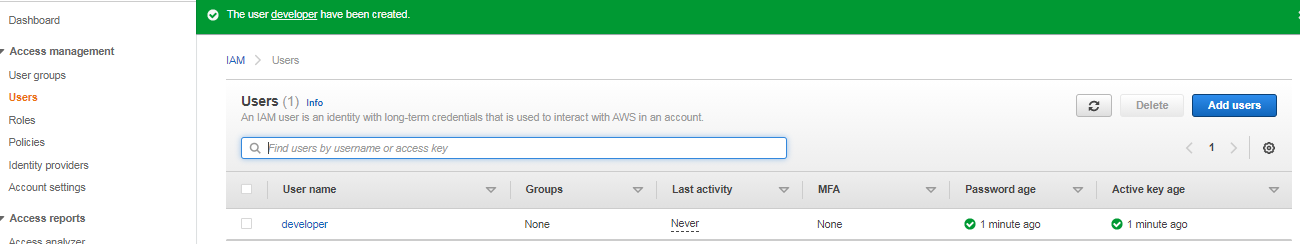
**Step 2: Create IAM user account for developer**

1. **Goto users and click add users**

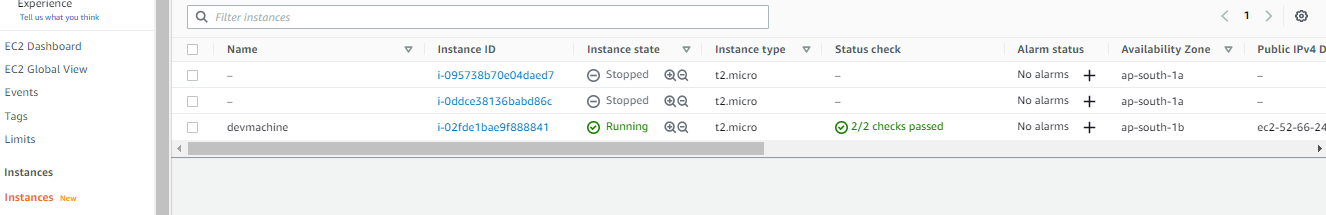
**  **

**Download .csv file**

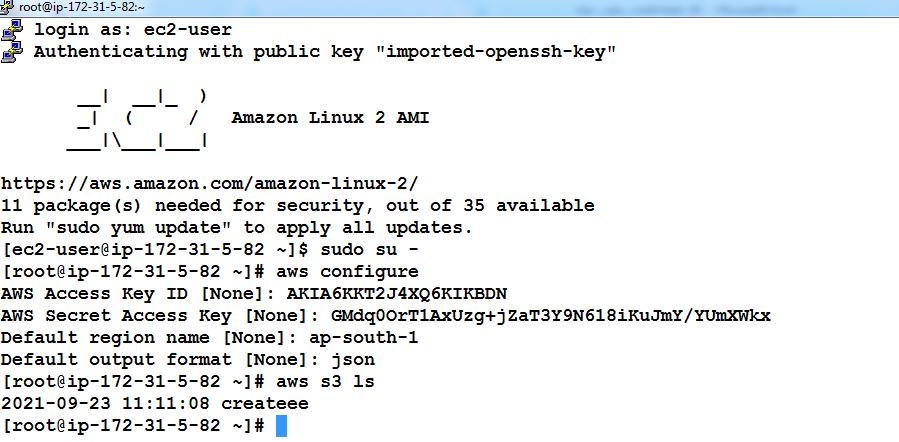
****

****

**Create Amazon Linux EC2 Server with t2.micro instance type … open all tcp.**

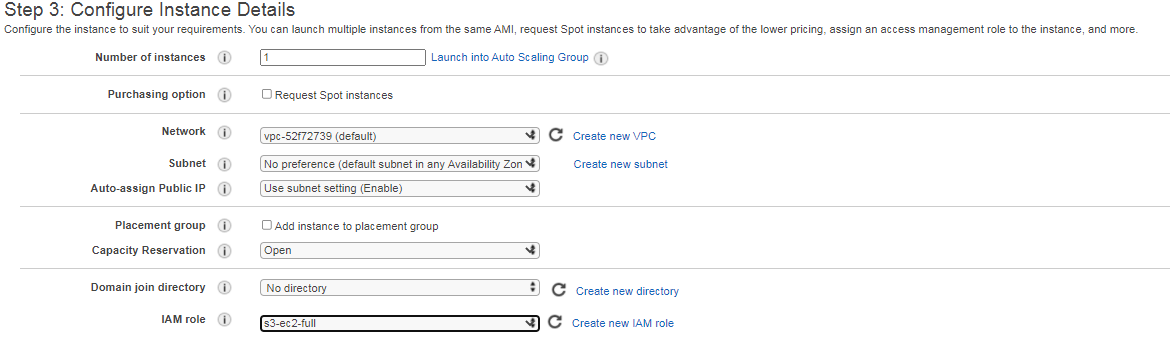
****

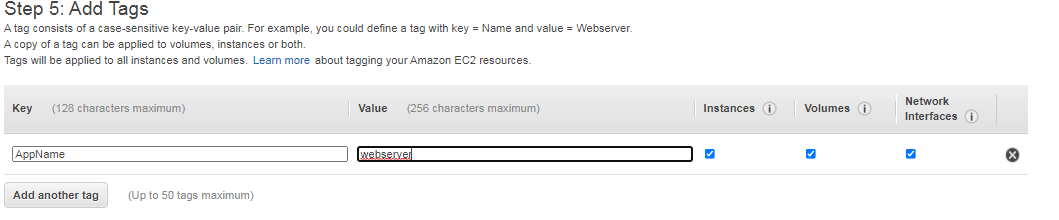
**Make this devmachine as a developer machine by configuring user called “developer”.**

****

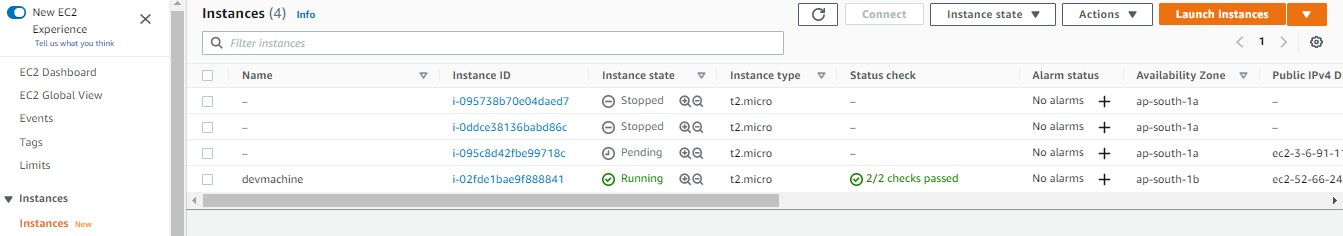
**Step 3: Install and prepare the CodeDeploy agent on webserver.-black screen**

**Create an Amazon Linux Ec2 server and associate s3-ec2-full role**

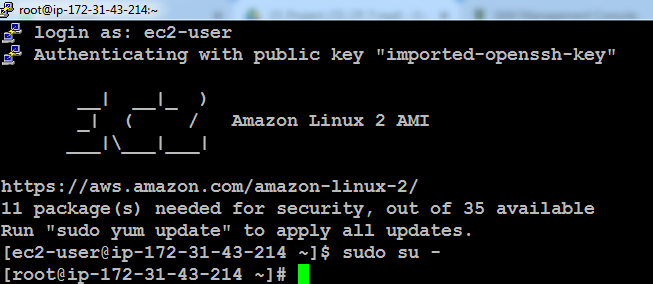
****

****

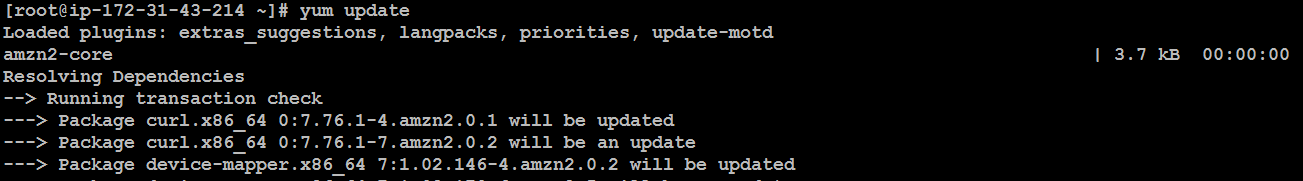
**Name doesnot appear becase we named tag as a “AppName”**

****

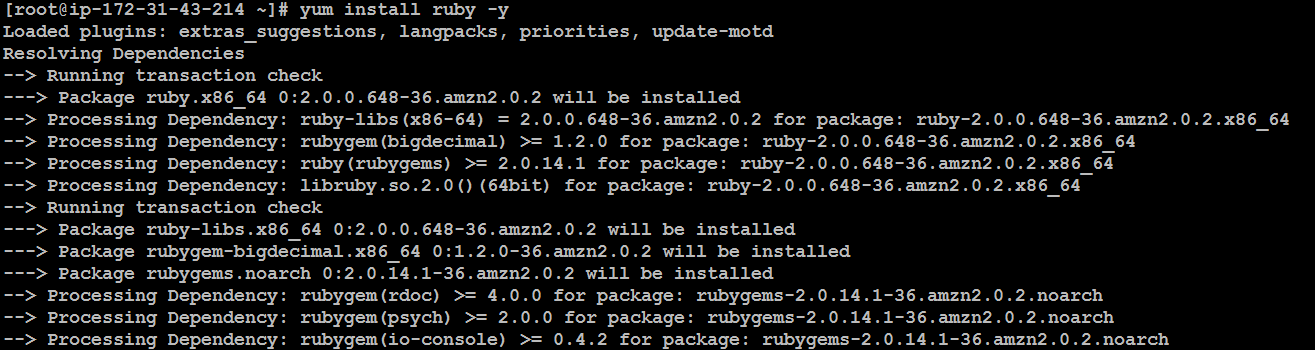
**e)**

****

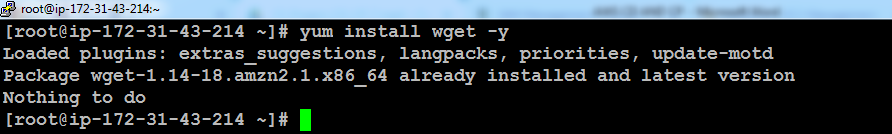
**Refresh Environment: yum update**

****

**yum install ruby -y**

****

**yum install wget -y**

****

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

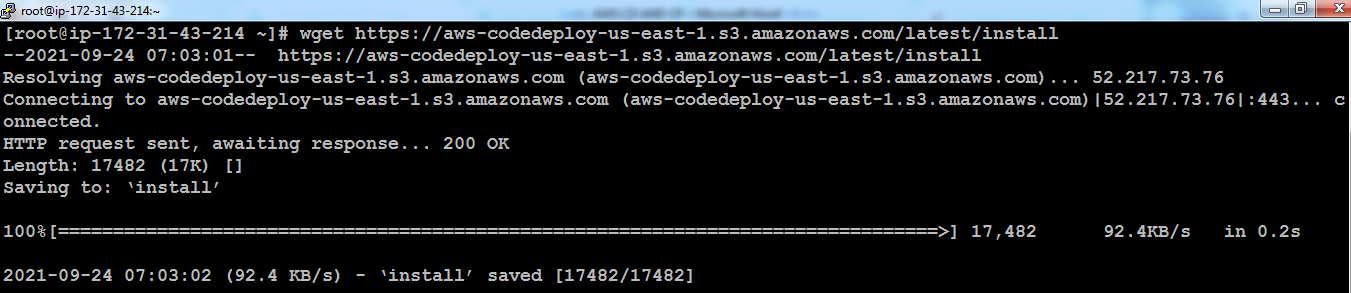
# chmod +x install

#./install auto

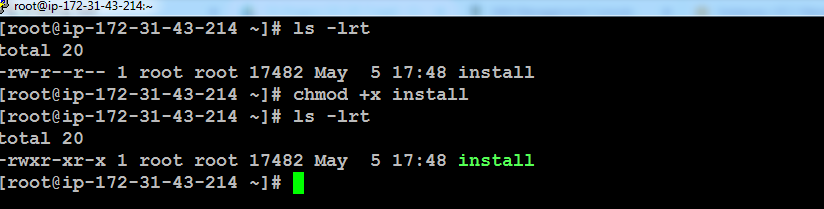
# service codedeploy-agent status

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

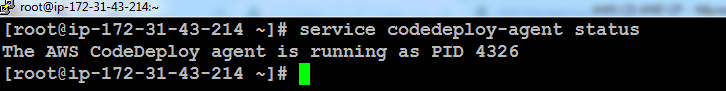
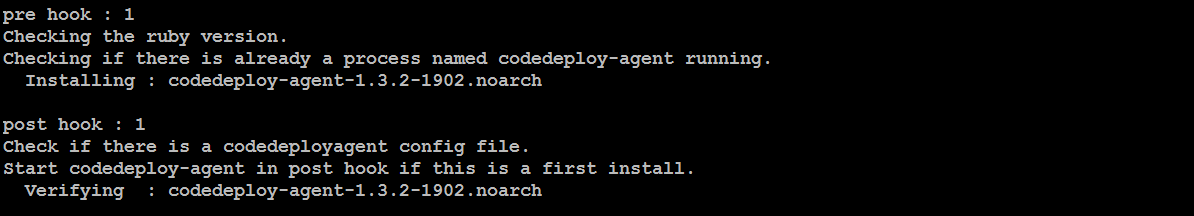
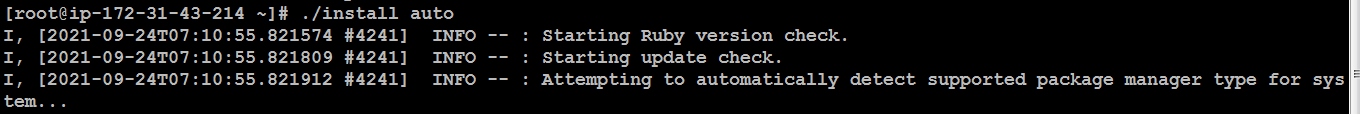
download install script to install the code deploy agent

****

**Chmod +x install**

****

**Install the script: By Default it checks on the ruby version.**

****

**Step 4: Create the code from Developer machine – white machine**

**Work in root**

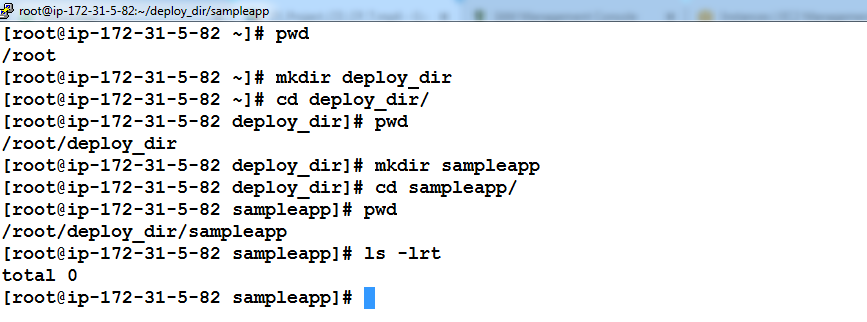
**a)mkdir deploy\_dir**

**cd deploy\_dir**

**mkdir sampleapp**

**cd sampleapp**

**ls –lrt**

****

1. **vi index.html**

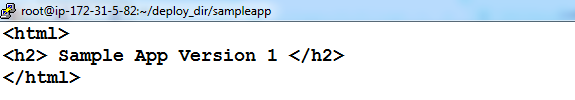
****

**Enter below content and save using wq!**

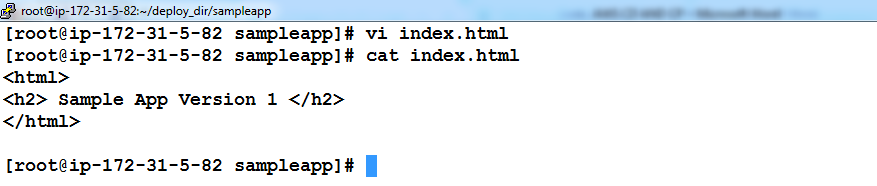
<html>

<h2> Sample App Version 1 </h2>

</html>

****

**:wq! Then enter**

****

1. **Create appspec.yml file – This helps to deploy the source code into webserver automatically**

****

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

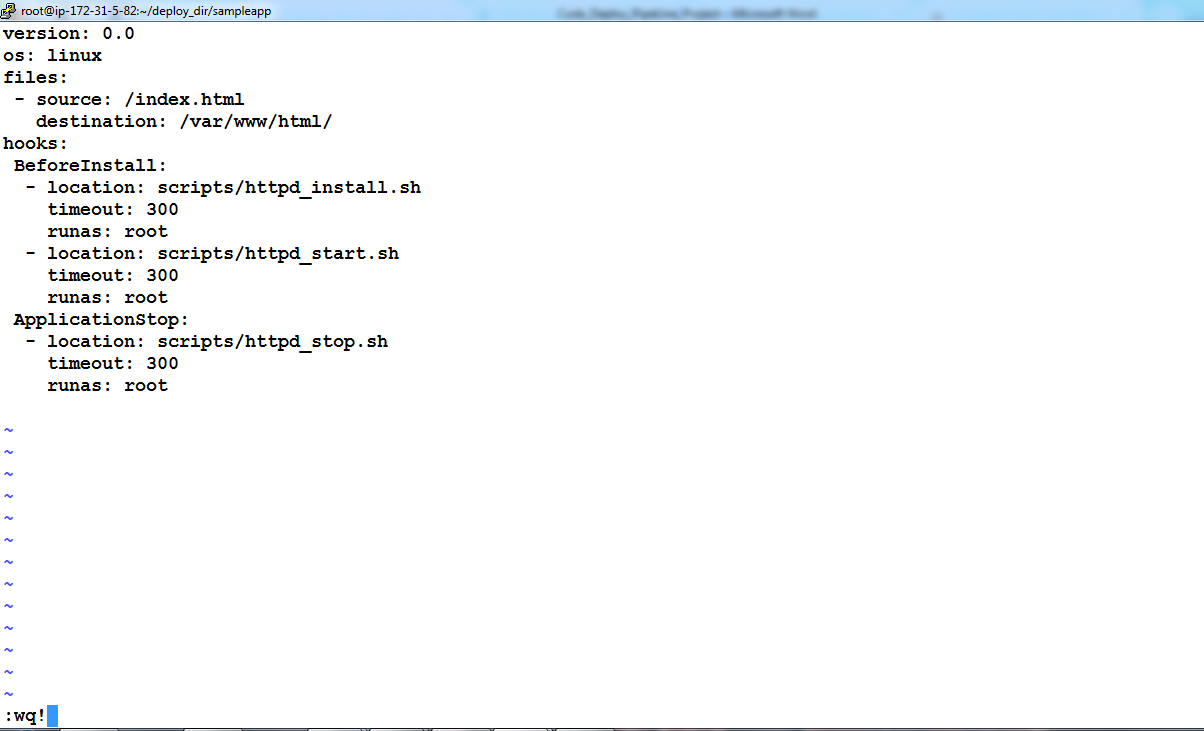
runas: root

ApplicationStop:

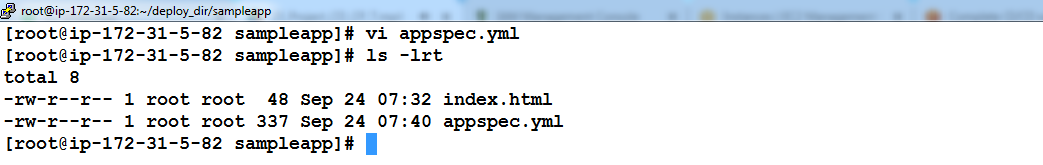
- location: scripts/httpd\_stop.sh

timeout: 300

runas: root

****

**Check: two files has been ready**

****

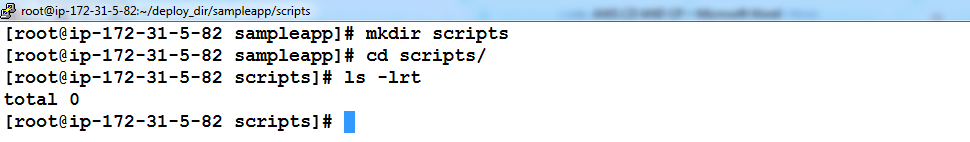
**mkdir scripts**

**cd scripts**

**vi** httpd\_install.sh

**vi** httpd\_start.sh

vi httpd\_stop.sh

****

[root@ip-172-30-0-178 deploy\_dir]# cat sampleapp/scripts/httpd\_install.sh

#!/bin/bash

yum install -y httpd

[root@ip-172-30-0-178 deploy\_dir]# cat sampleapp/scripts/httpd\_start.sh

#!/bin/bash

systemctl start httpd

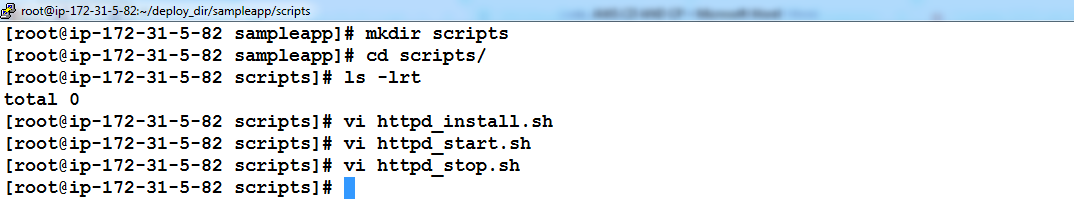
systemctl enable httpd

[root@ip-172-30-0-178 deploy\_dir]# cat sampleapp/scripts/httpd\_stop.sh

#!/bin/bash

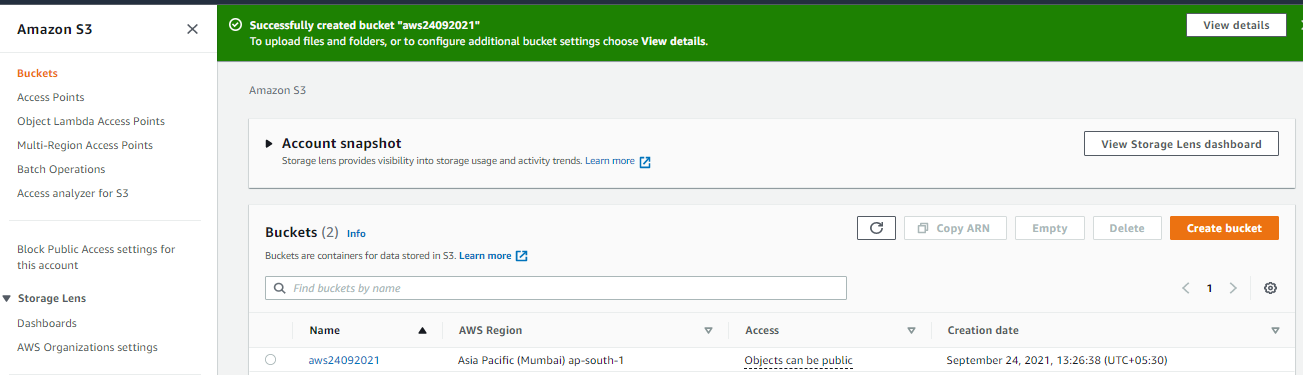
systemctl stop httpd

systemctl disable httpd

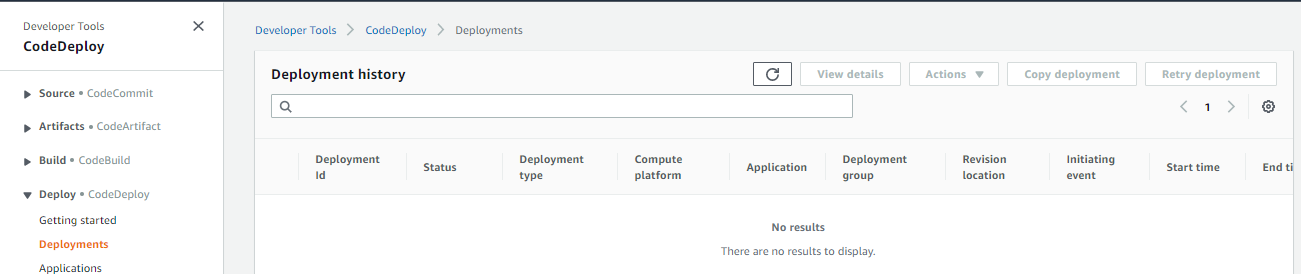
****

**Step 5: Create Codedeploy Application and Push the code to S3 bucket from Developer machine**

1. **Create bucket name called aws24092021 – make it public policy**

****

**Check the codedeploy in the Console level in GUI**

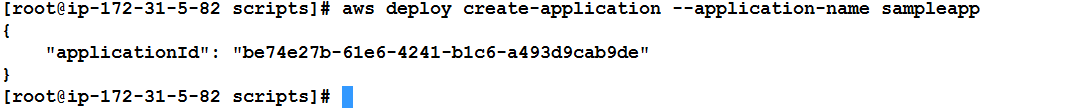
****

1. **Create bucket name called aws24092021 – make it public policy**

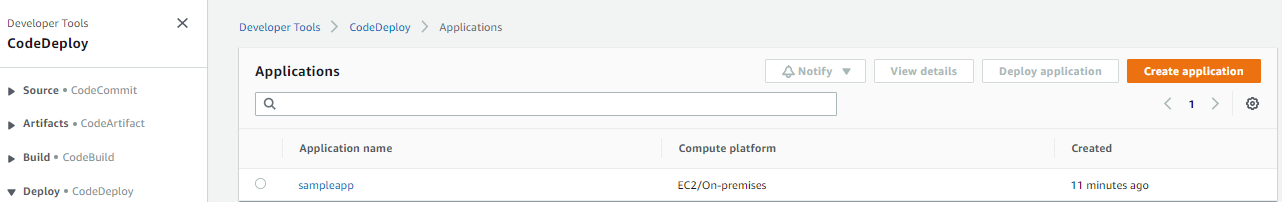
**Run the cli and create an application in the code deploy**

**Note: We might run the command where the appspec.yml kept**

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

****

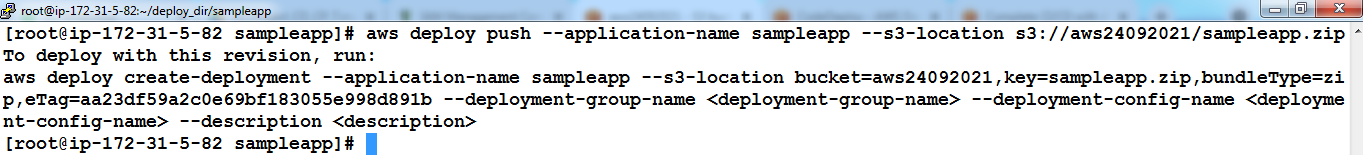
**Check and confirm whether the sample application been created successful created in the aws console:**

****

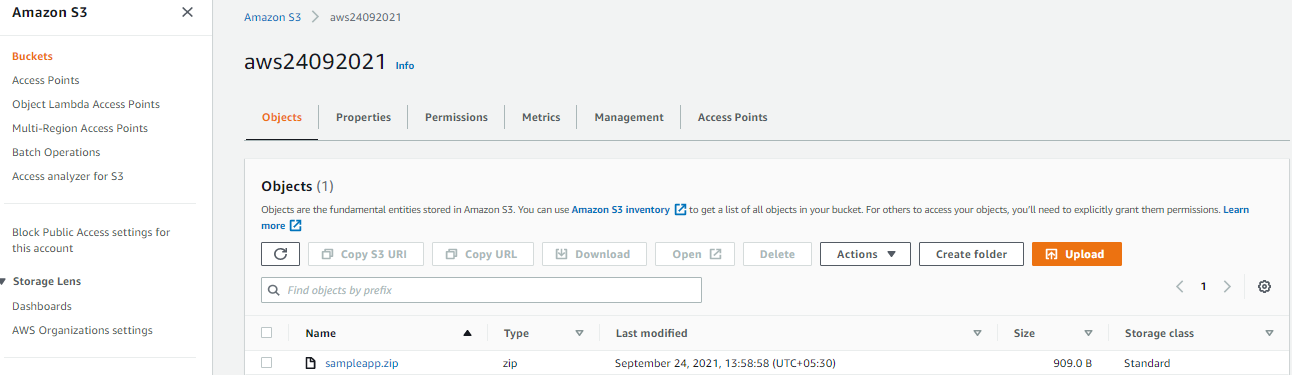
**c- Now upload the code to S3 by executing the command below. Directory of execution is important.**

**aws24092021**

**# aws deploy push --application-name sampleapp --s3-location s3://aws24092021/sampleapp.zip**

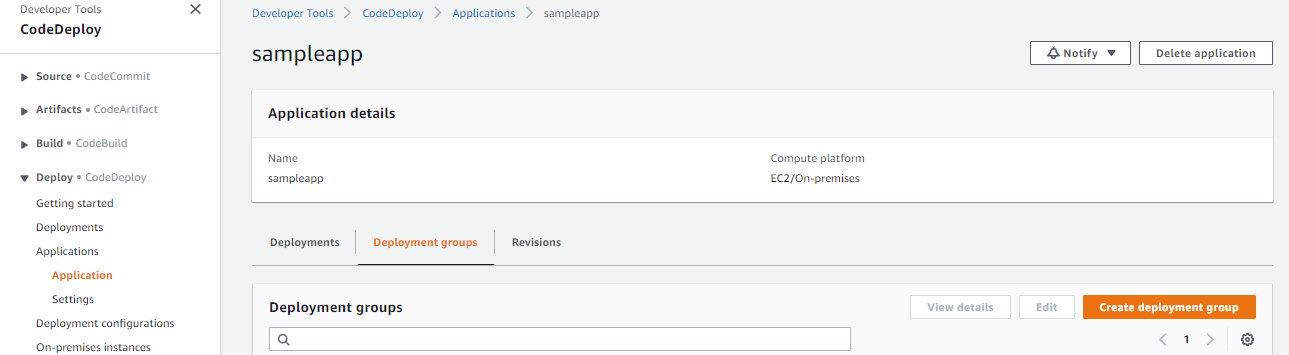
****

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

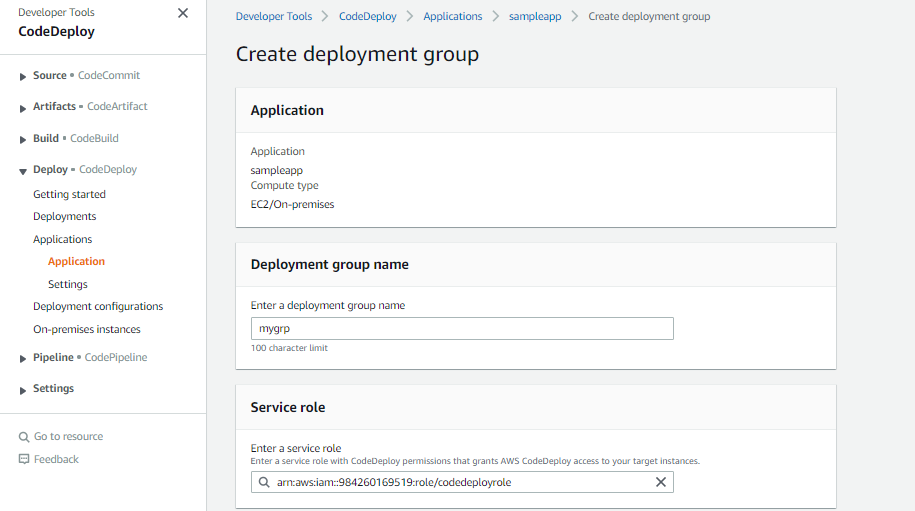
****

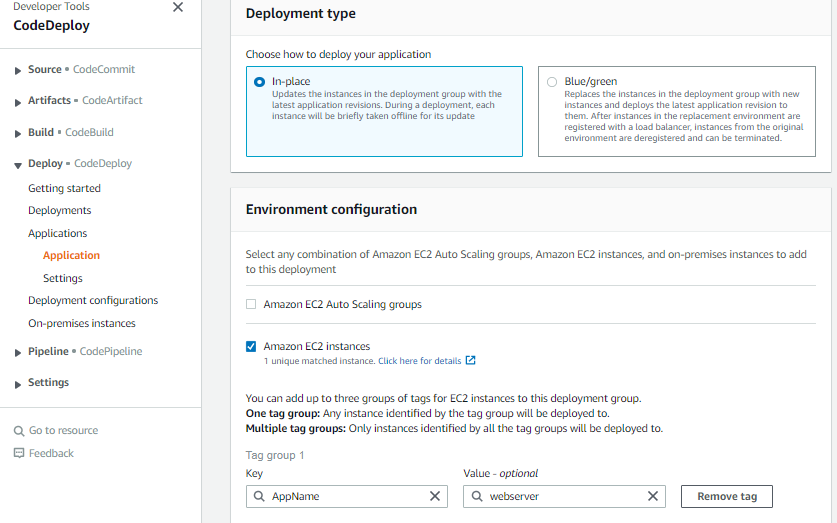
**Step 6: Create Deployment Group to include web server**

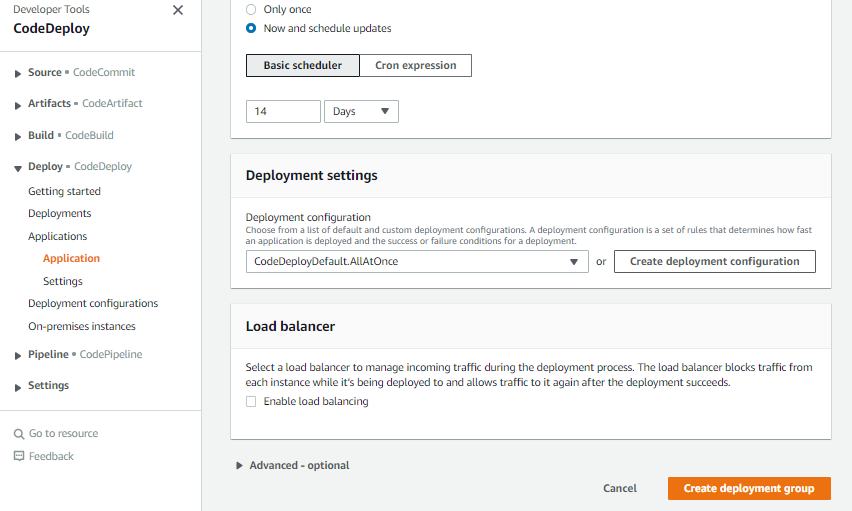
1. Login to Codedeply AWS web console



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







c- Enter the values like below and leave the other parameters default

*Enter a deployment group name*: mygrp

*Choose a service role*:  cdrole

*Deployment type*: in-place

*Environment configuration:* choose Amazon EC2 instances

Key as *AppName* Value as S*ampleApp*

*Load balancer:*  uncheck Enable load balancing

Click *Create Deployment Group* button to finish creation of deployment group

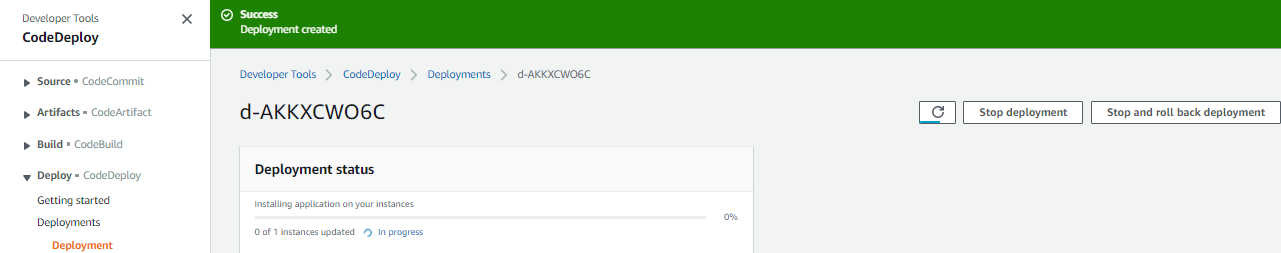
**7-Create Deployment which pushes code to the webserver**

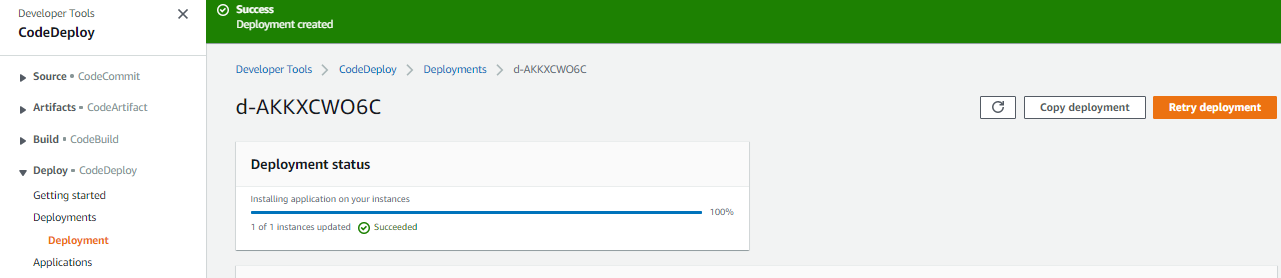
In the sampleapp click *Create Deployment.*Enter values like below. Other parameter can can be kept default

*Deployment group :* mygrp

*Revision type:* My application is stored in Amazon S3

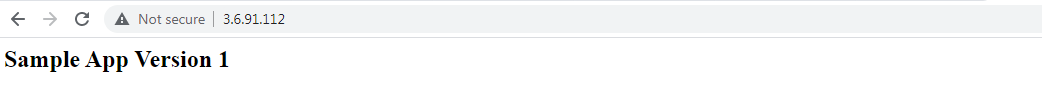
R*evision location* : s3://select\_location\_from\_list



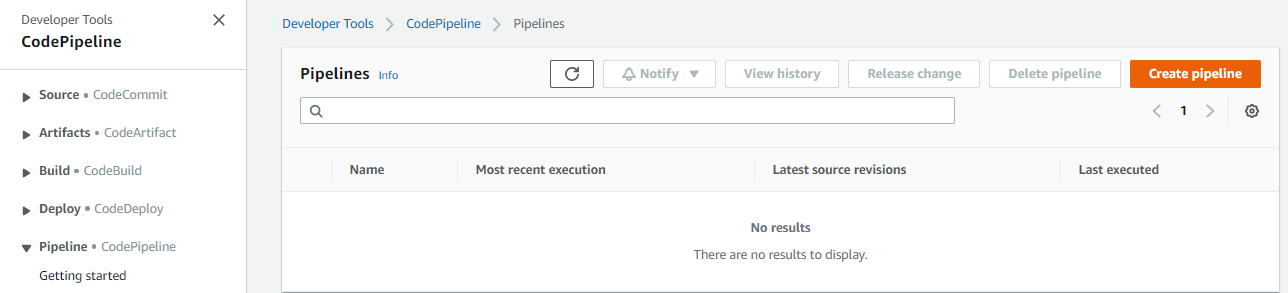


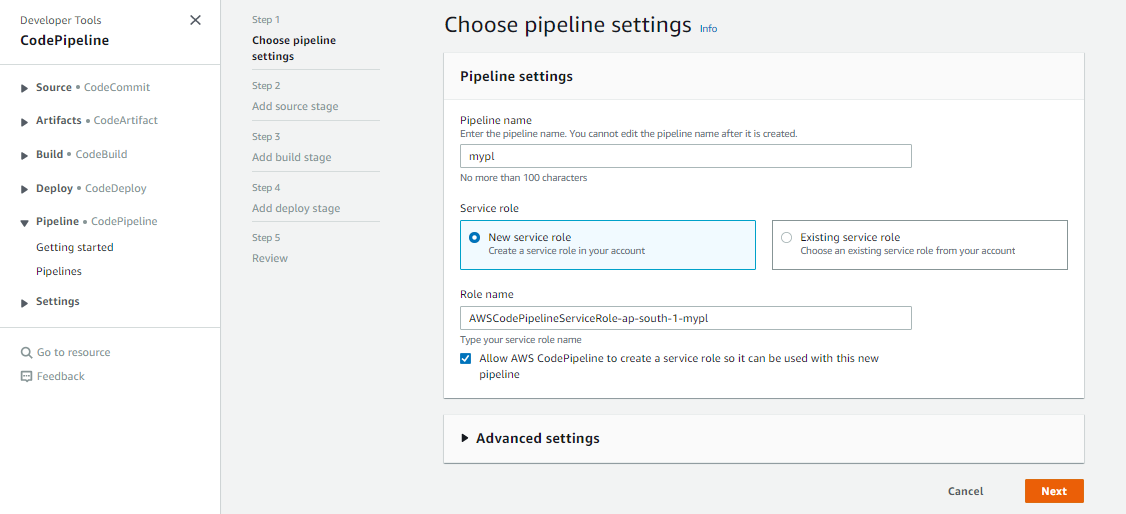
Click *Create Deployment* to finish

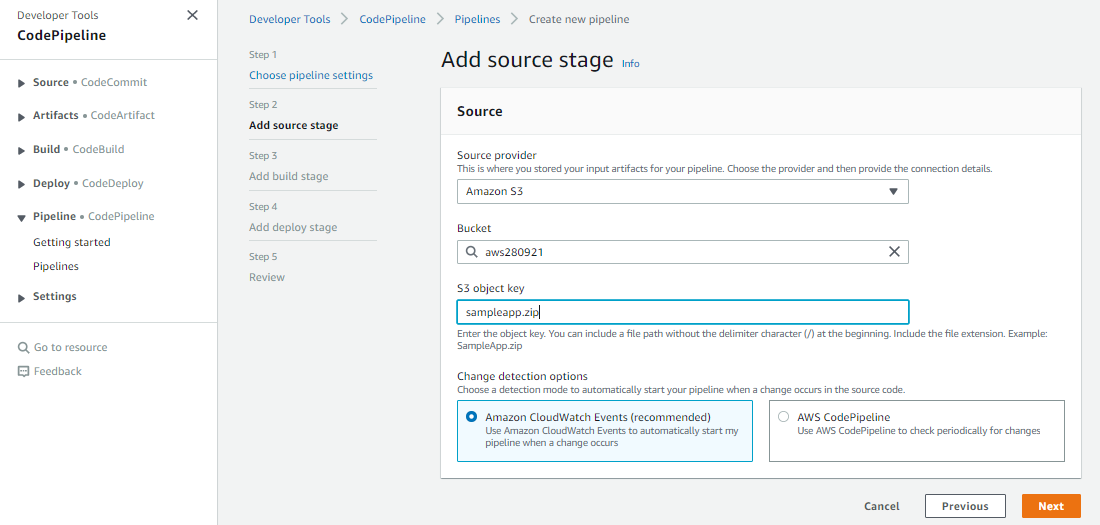
**Successfully done the deployment**

****

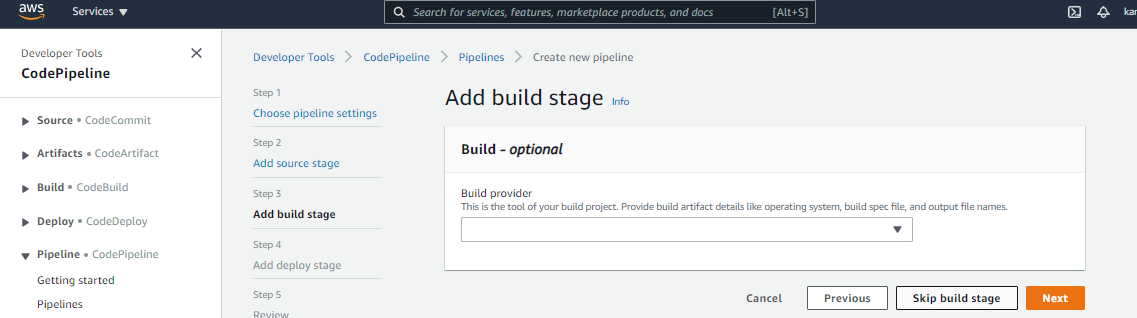
**CREATE PIPELINE TO ENABLE AUTOMATIC DEPLOYMENT MOMENT THE NEW VERSION REACHES THE REPOSITORY CALLED S3**

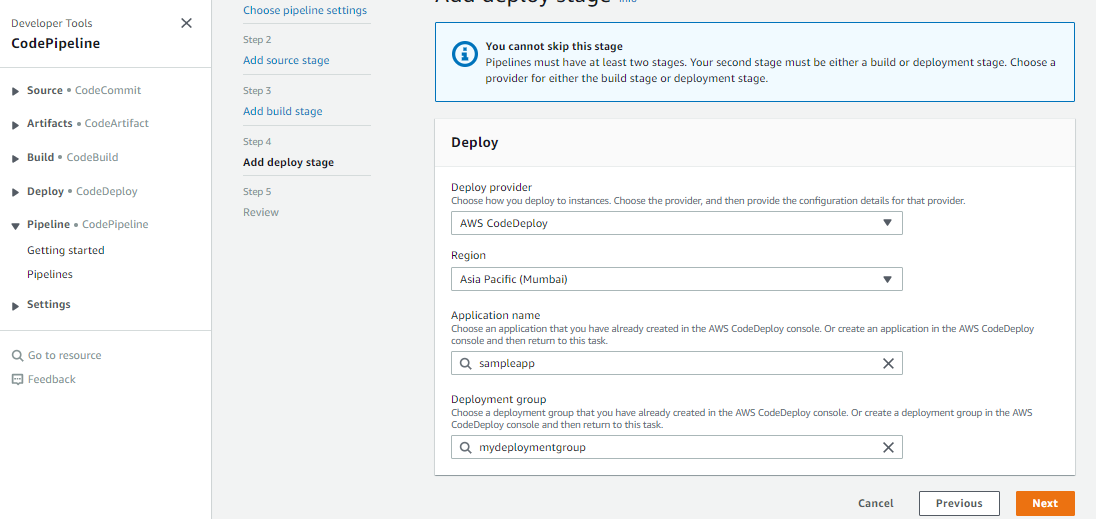
****

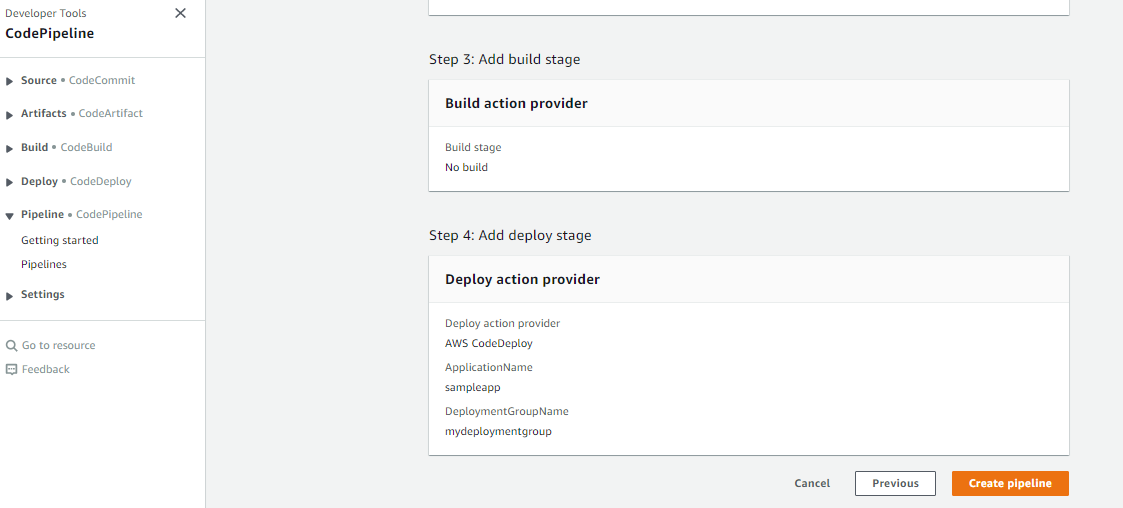
****

****

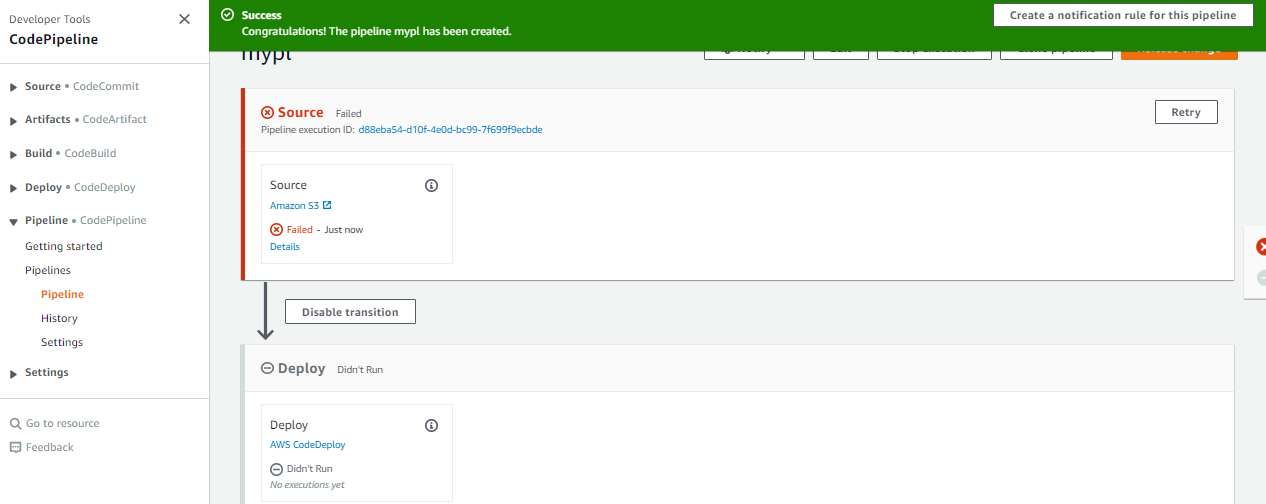
**SKIP THE BUILD STAGE AND MOVE ON..**

****

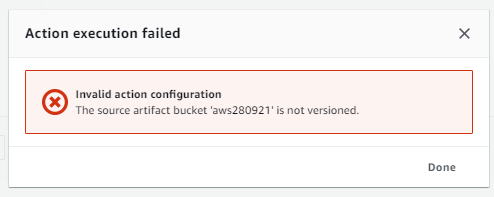
****

****

**EXCEPTION OCCURES LIKE BELOW,**

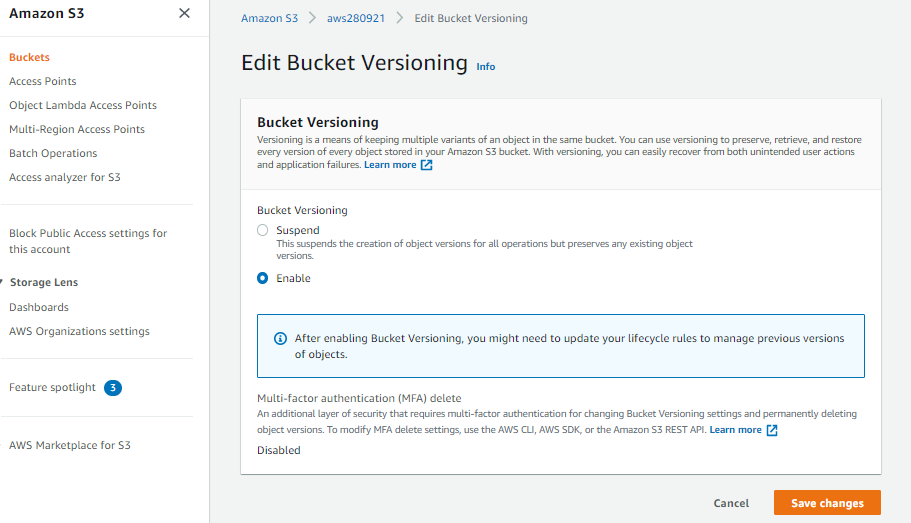
****

**REASON FOR THE FAILURE:**

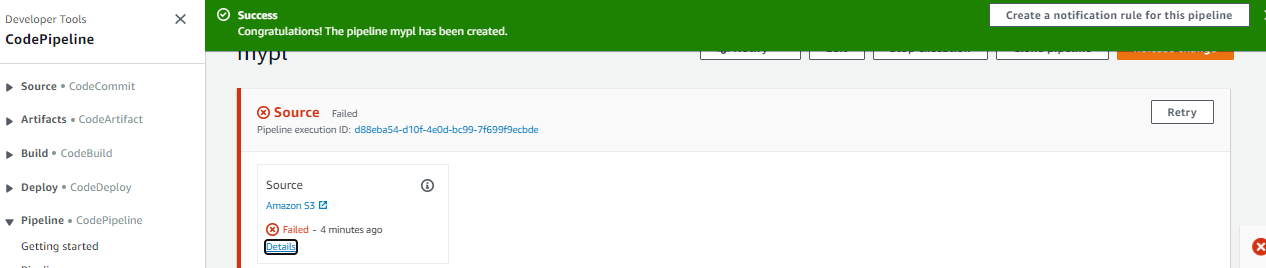
****

**RESOLUTION:**

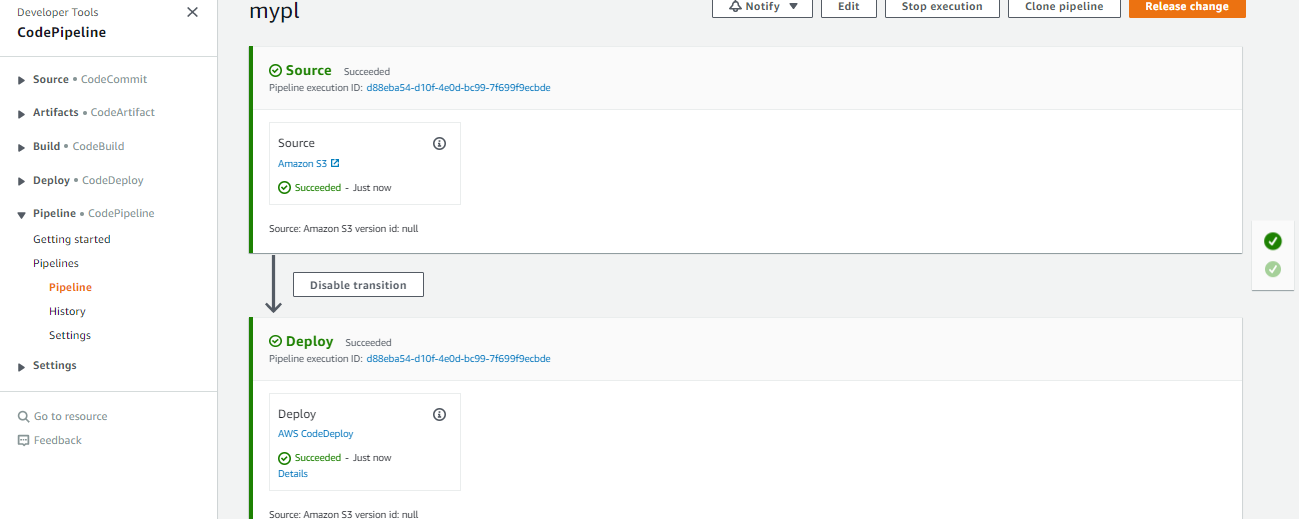
**ENABLE BUCKET VERSIONING AND SAVE**

****

**CLICK RETRY….**

****

**BOTH SOURCE AND DEPLOYMENT GOT SUCCEEDED**

****

**TESTING THE PIPELINE PROCESS:**

**GOTO DEVELOPER MACHINE AND PUSH NEW VERSION OF THE sampleapp.zip**

**Edit the content:**

**[root@ip-172-31-13-20 sampleapp]# vi index.html**

**<html>**

**<h2> Sample App Version 2 </h2>**

**</html>**

**EXECUTE THE BELOW COMMAND AND CREATE ZIP FILE FIRST**

**zip -r ../sampleapp.zip .**

**[root@ip-172-31-13-20 sampleapp]# zip -r ../sampleapp.zip .**

**adding: appspec.yml (deflated 53%)**

**adding: scripts/ (stored 0%)**

**adding: scripts/httpd\_install.sh (stored 0%)**

**adding: scripts/httpd\_start.sh (deflated 21%)**

**adding: scripts/httpd\_stop.sh (deflated 21%)**

**adding: index.html (deflated 9%)**

**CLICK RETRY….**

**[root@ip-172-31-13-20 sampleapp]# cd ..**

**[root@ip-172-31-13-20 deploy\_dir]# ls -lrt**

**total 4**

**drwxr-xr-x 3 root root 58 Sep 28 07:00 sampleapp**

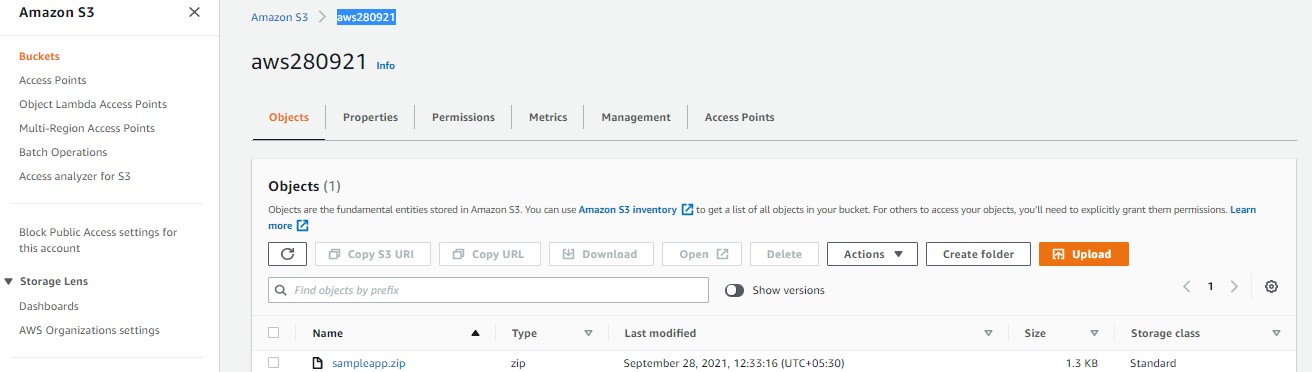
**-rw-r--r-- 1 root root 1306 Sep 28 07:01 sampleapp.zip**

**[root@ip-172-31-13-20 deploy\_dir]# aws s3 cp sampleapp.zip s3://aws280921**

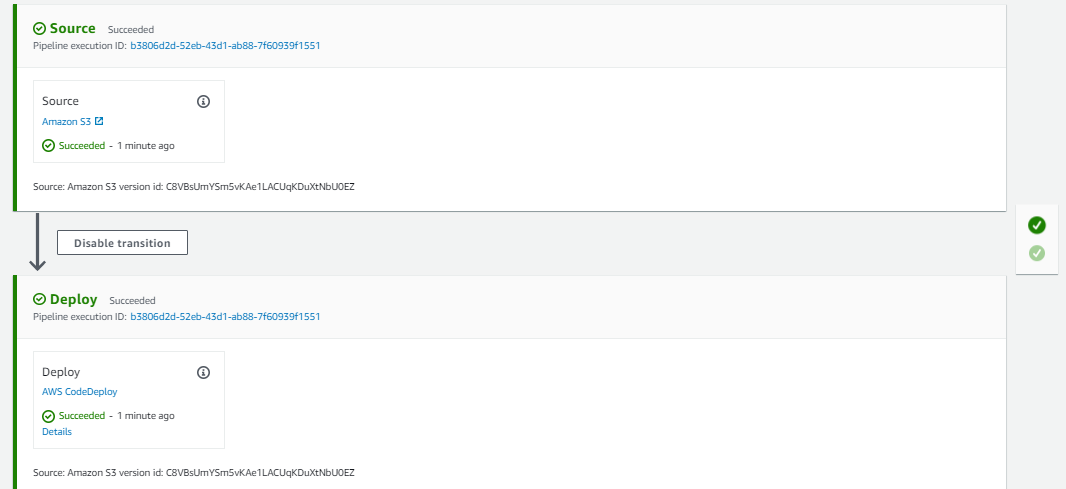
**upload: ./sampleapp.zip to s3://aws280921/sampleapp.zip**

**[root@ip-172-31-13-20 deploy\_dir]#**

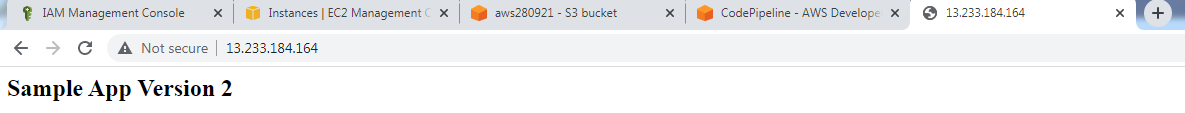
**NEW TIMESTAMP HAS BEEN UPDATED**

****

**NEW SOURCE CODE HAS BEEN SUCCESSFULLY TRIGGERED FROM S3 BY CODE PIPELINE AND DEPLOYMENT NEW VERSION INTO WEBSERVER**

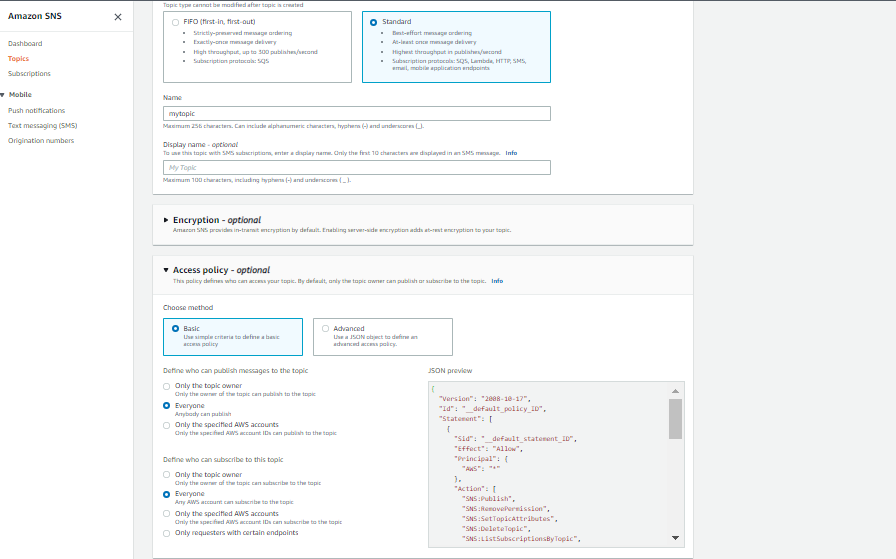
****

**VERIFY THE SAME BY HITTING THE PUBLIC IP OF THE WEBSER MACHINE IN THE BROWSER AND CHECK**

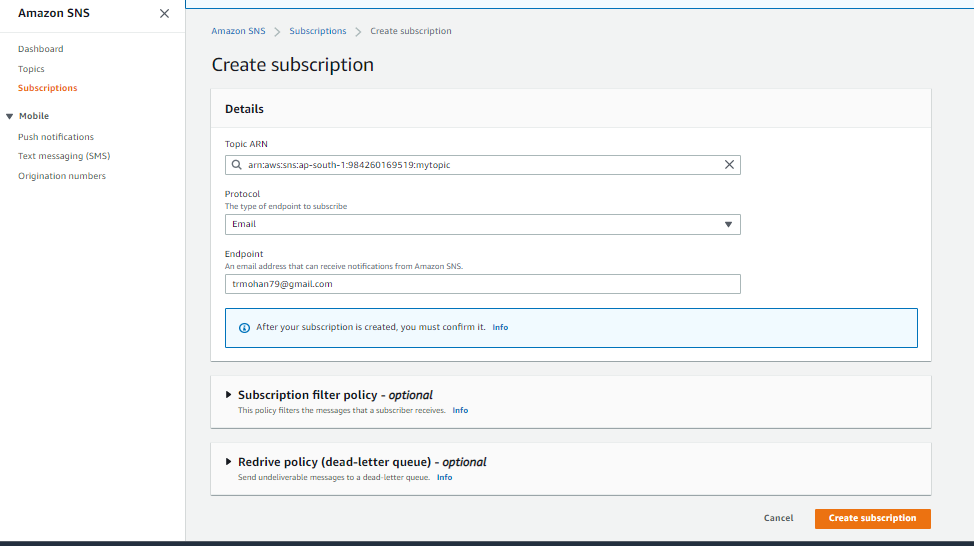
****

**ENABLE THE NOTIFICATION SERVICE: ALERTING TO THE USER IN CASE IF THE DEPLOYMENT SUCCEED, FAILED, START ETC**

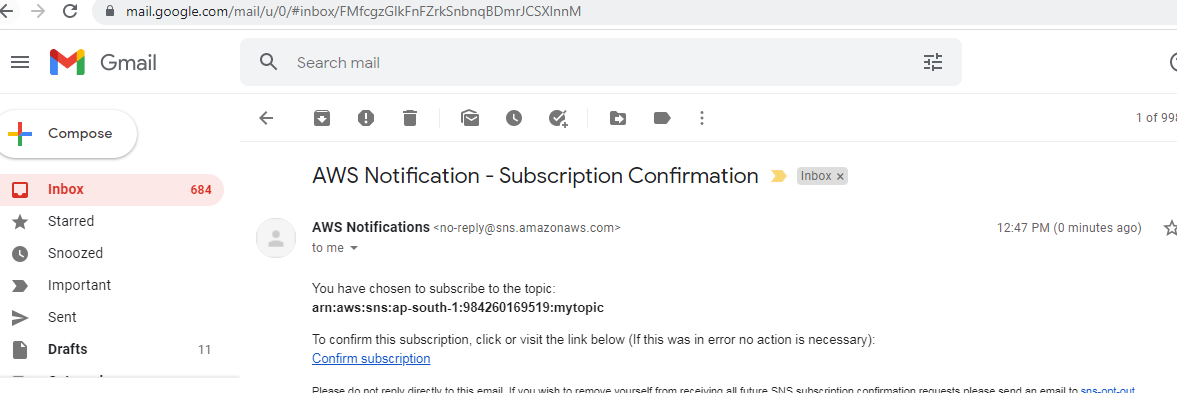
**Create SNS name called mytopic and make it as a public**

****

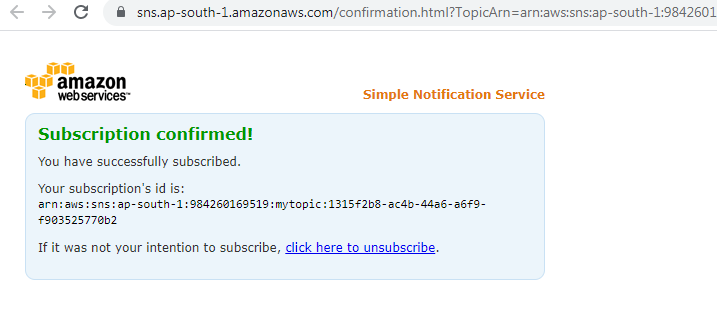
**CREATE SUBSCRIPTION1 –USING EMAIL PROTOCOL**

****

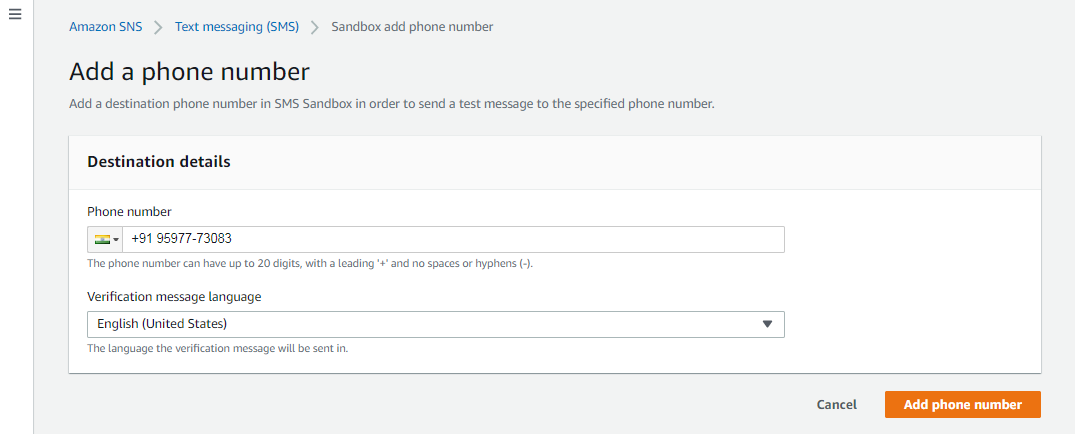
**Confirm the subscription since gmail is an third party server for the aws**

****

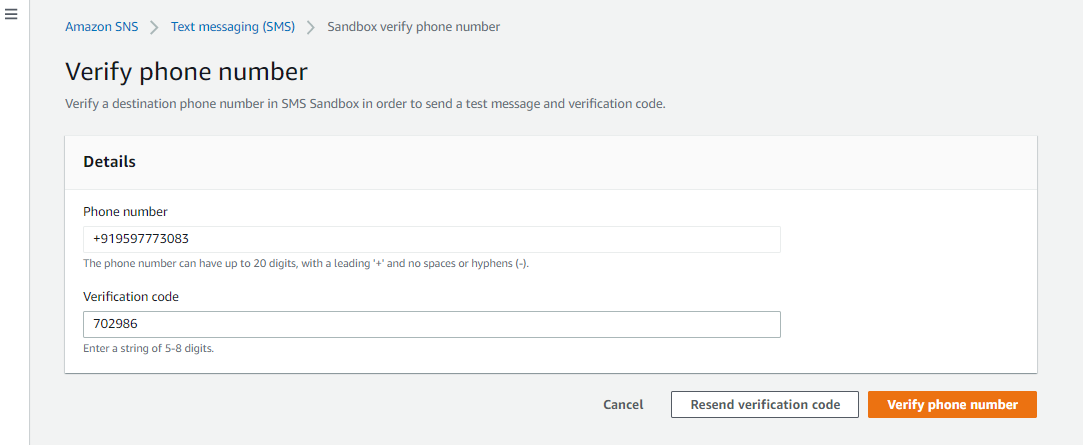
**SUBSCRIPTION CONFIRMED**

****

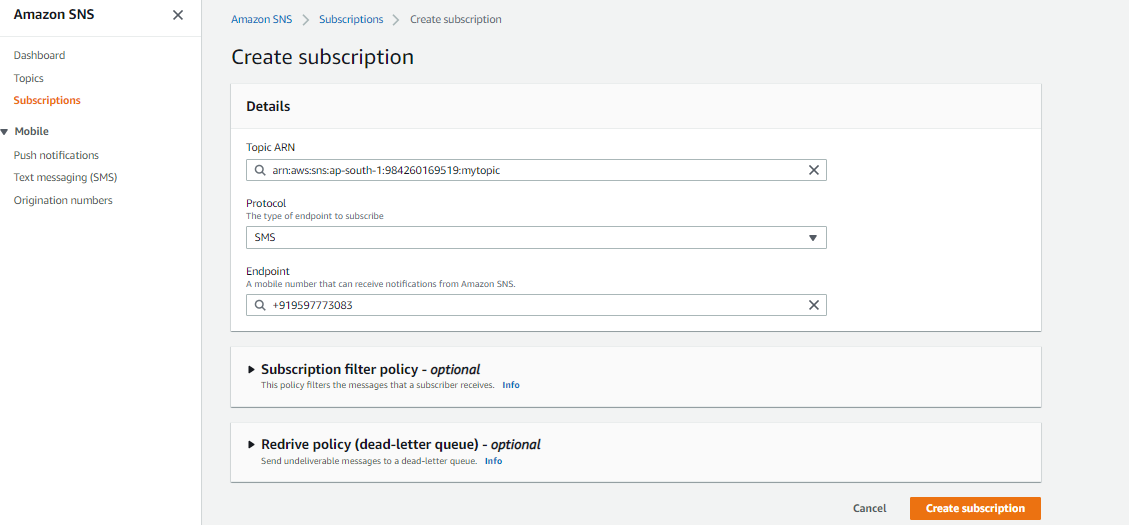
**ADD ANOTHER PROTOCOL CALLED SMS**

****

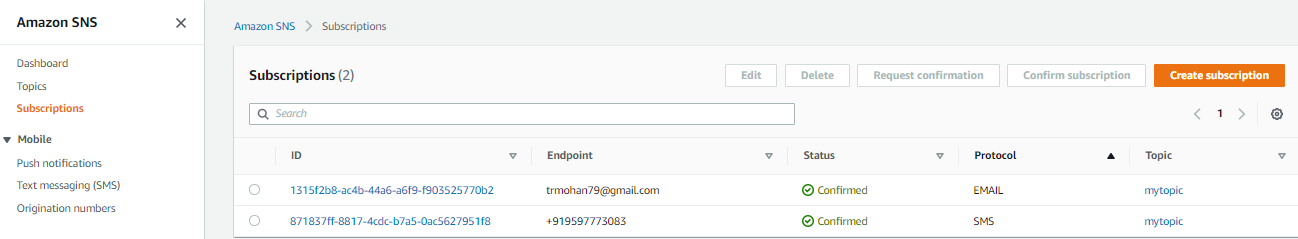
**VERIFY PHONE NUMBER**

****

**CREATE SUBSCRIPTION**

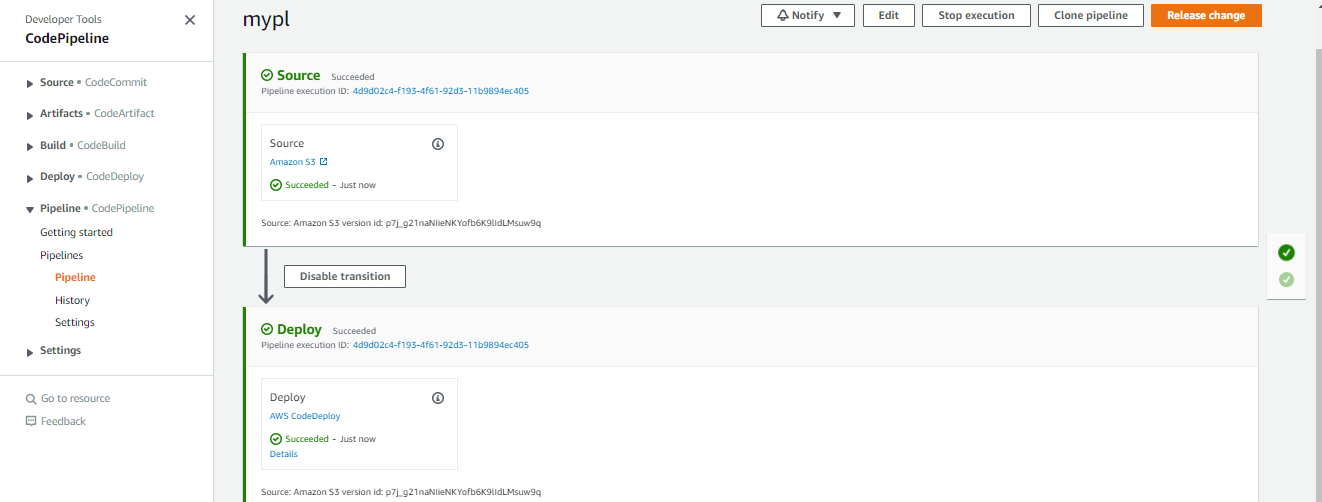
****

**TWO SUBCRIPTIONS HAS BEEN CREATED:**

****

**TEST THE DEPLOYMENT ONCE AGAIN WITH SAME STEPS HOW WE FOLLOWED EARLIER AND**

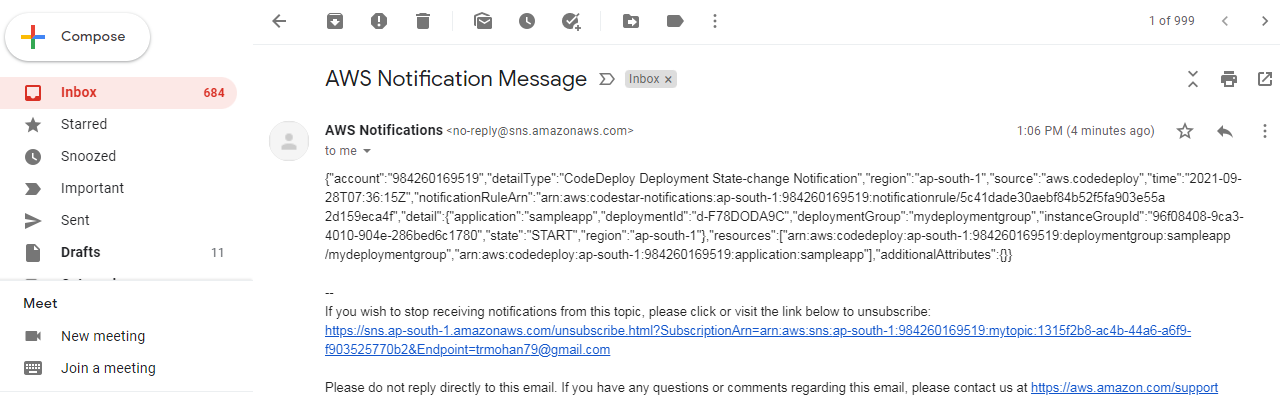
1. **CHANGE THE CONTENT IN THE index.html**
2. **CREATE AN sampleapp.zip**
3. **PUSH THE SOURCE CODE THE S3 BUCKET**
4. **CHECK THE PIPELINE STATUS**
5. **ALSO CHECK THE ALERTING**

****

**VERIFY THE NEW SOURCE CODE DEPLOYMENT WITH CODE PIPELINE**

**SUCCESSFULLY DONE**

**EMAIL NOTIFICATION:**

****

**SMS NOTIFICATION:**

{"account":"984260169519","detailType":"CodeDeploy Deployment State-change Notification","region":"ap-south-1","source":"aws.codedeploy","time":"2021-09-28T07:36:25Z","notificationRuleArn":"arn:aws:codestar-notifications:ap-south-1:984260169519:notificationrule/5c41dade30aebf84b52f5fa903e55a2d159eca4f","detail":{"application":"sampleapp","deploymentId":"d-F78DODA9C","deploymentGroup":"mydeploymentgroup","instanceGroupId":"96f08408-9ca3-4010-904e-286bed6c1780","state":"SUCCESS","region":"ap-south-1"},"resources":["arn:aws:codedeploy:ap-south-1:984260169519:deploymentgroup:sampleapp/mydeploymentgroup","arn:aws:codedeploy:ap-south-1:984260169519:application:sampleapp"],"additionalAttributes":{}}

**DEMOLISH:**

* 1. **DELETE INSTANCES**
  2. **DELETE CODE DEPLOY**
  3. **DELETE CODE PIPELINE**
  4. **DELETE BUCKET AND OBJECT**
  5. **DELETE SNS TOPIC AND SUBSCRIPTIONS**
  6. **DELETE IAM ROLES AND USERS**