

Day 52: Your CI/CD pipeline on AWS - Part 3

On your journey of making a CI/CD pipeline on AWS with these tools, you completed AWS CodeCommit & CodeBuild.

Next few days you'll learn these tools/services:

- CodeDeploy
- CodePipeline
- S3

What is CodeDeploy ?

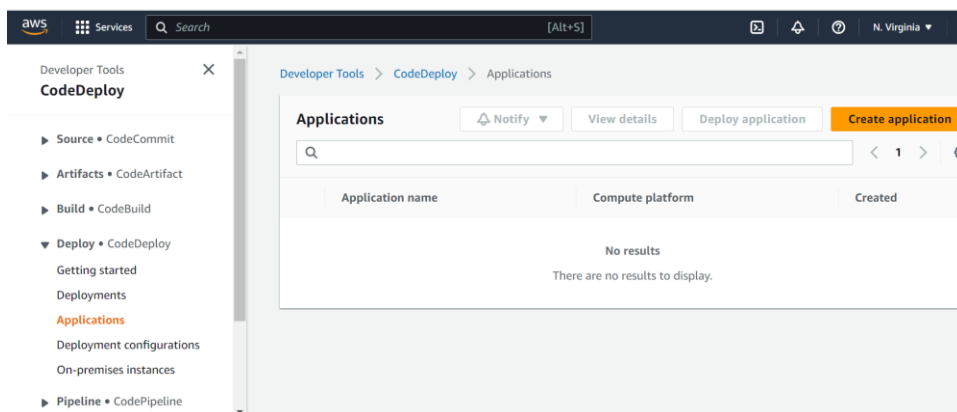
- AWS CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

CodeDeploy can deploy application content that runs on a server and is stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. CodeDeploy can also deploy a serverless Lambda function. You do not need to make changes to your existing code before you can use CodeDeploy.

Task-01:

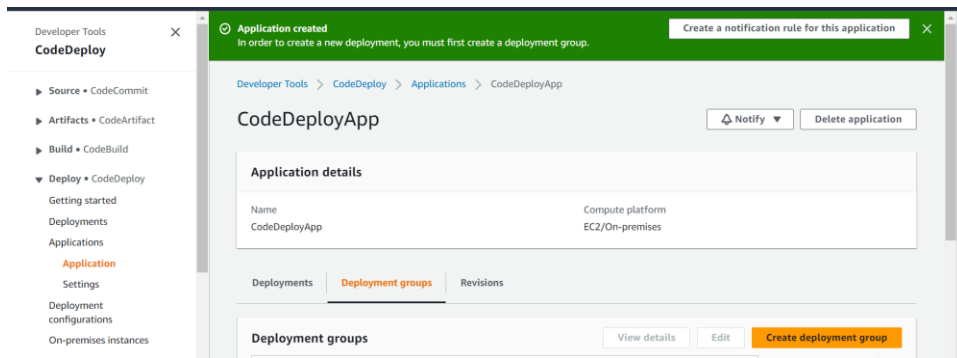
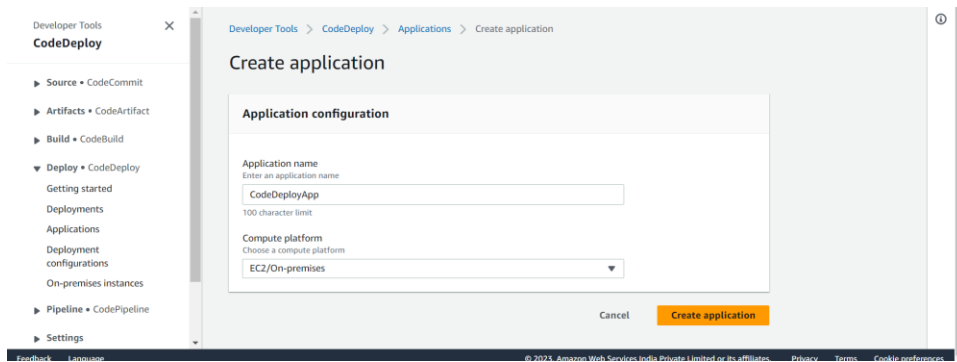
- Read about Appspec.yaml file for CodeDeploy.

Goto CodeDeploy Tab and First Create an application

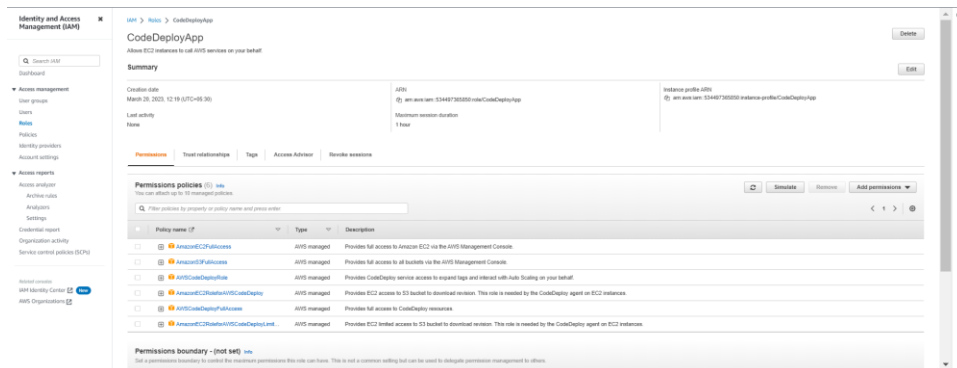


Give Application Name and Select Compute Platform.

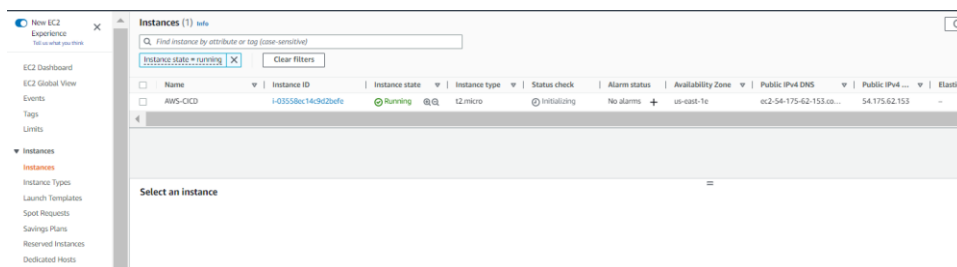
Click on Create Application.



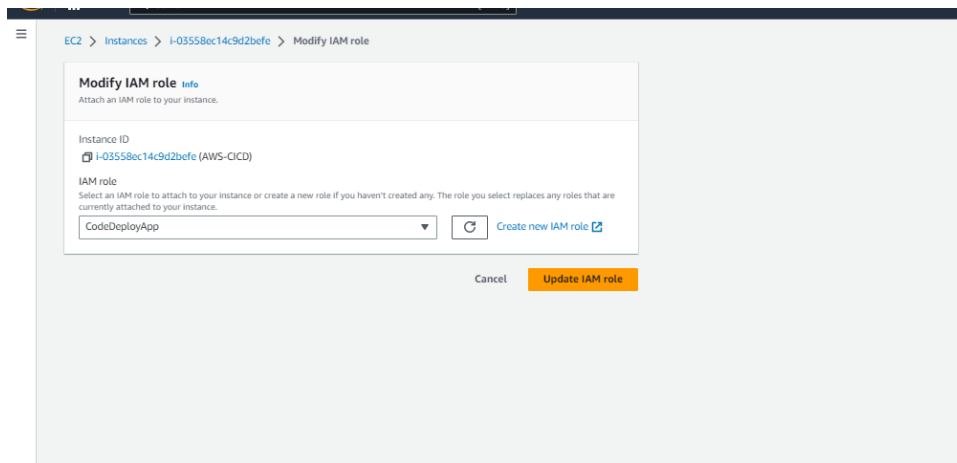
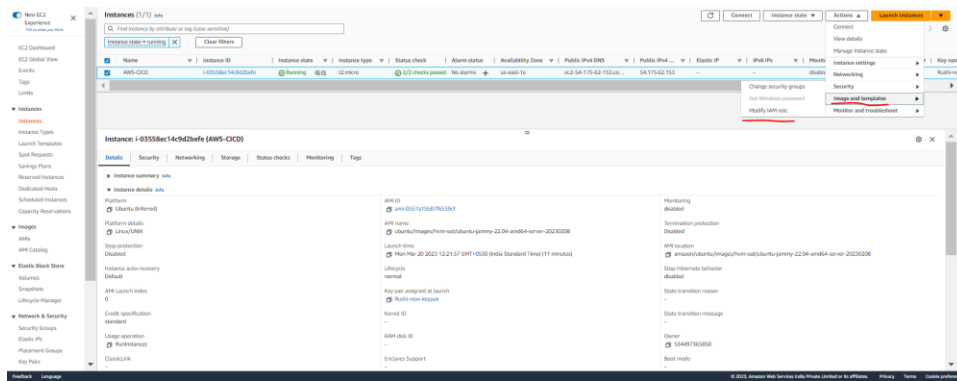
For Deployment, we need an IAM Role, Using This role we can establish a connection between CodeDeploy and EC2 Instance.



Create a one Instance

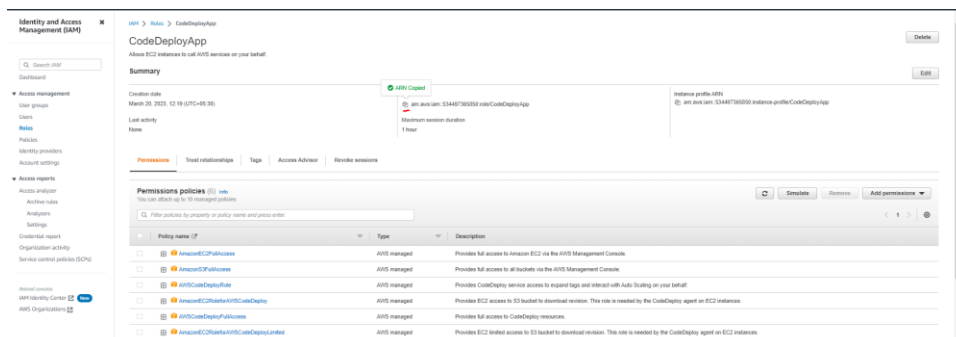
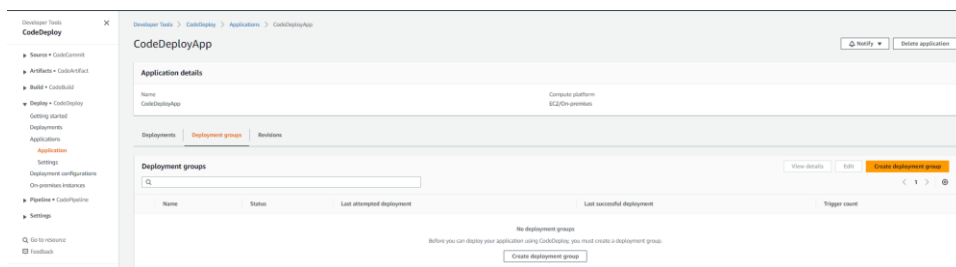


Attache newly created IAM Role to this Instance.



Ok, Now the Second Step is to Create a Deployment group.

Click on create Deployment group button



Give Deployment group name and IAM Service Role ARN.

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

Create deployment group

Application

Application
CodeDeployApp
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name

CodeDeployGroup

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::534497365850:role/CodeDeployApp

Enter a service role ARN

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.

Feedback Language

Choose **In-place** as a Deployment type and In Environment Configuration click on **Amazon EC2 Instance** checkbox and add Running EC2 instance name.

arn:aws:iam::534497365850:role/CodeDeployApp

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	
Q Name	Q AWS-CICD	Remove tag
Add tag		
+ Add tag group		

☐ On-premises instances

Here in this step, we don't want to install CodeDeploy Agent.

We can directly install the CodeDeploy agent on EC2 instance later.

Note: - if you are getting this type of error, when creating a deployment group, follow below Reference document to solve this type of issue.

AWS CodeDeploy does not have the permissions required to assume the role

Here in this Step, we are setting up the CodeDeploy Agent.

Refer Below Document, to get the All commands to Setup agent on EC2 instance.

[Setting Up AWS CodeDeploy Agent on Ubuntu EC2](#)

```
ubuntu@ip-172-31-49-160:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-49-160:~$ vi install.sh
ubuntu@ip-172-31-49-160:~$ vi install.sh
ubuntu@ip-172-31-49-160:~$ cat install.sh ↵
#!/bin/bash
# This installs the CodeDeploy agent and its prerequisites on Ubuntu 22.04.
sudo apt-get update
sudo apt-get install ruby-full ruby-webrick wget -y
cd /tmp
wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/releases/codedeploy-agent_1.3.2-1902_all.deb
mkdir codedeploy-agent_1.3.2-1902_ubuntu22
dpkg-deb -R codedeploy-agent_1.3.2-1902_all.deb codedeploy-agent_1.3.2-1902_ubuntu22
sed 's/Depends:.*/Depends:ruby3.0/' -i ./codedeploy-agent_1.3.2-1902_ubuntu22/DEBIAN/control
dpkg-deb -b codedeploy-agent_1.3.2-1902_ubuntu22/
sudo dpkg -i codedeploy-agent_1.3.2-1902_ubuntu22.deb
systemctl list-units --type=service | grep codedeploy
sudo service codedeploy-agent status
ubuntu@ip-172-31-49-160:~$
```

```
ubuntu@ip-172-31-49-160:~$ bash install.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [948 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [205 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.7 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [684 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [107 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [584 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [890 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [177 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.1 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [24.1 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [6312 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [19.5 kB]
```

Your Code Deploy Agent is Running.

```
Selecting previously unselected package codedeploy-agent.
(Reading database ... 80024 files and directories currently installed.)
Preparing to unpack codedeploy-agent_1.3.2-1902_ubuntu22.deb ...
Unpacking codedeploy-agent (1.3.2-1902) ...
Setting up codedeploy-agent (1.3.2-1902) ...
codedeploy-agent.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable codedeploy-agent
   codedeploy-agent.service                                loaded active running LSB: AWS CodeDeploy Host Agent
* codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
   Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
   Active: active (running) since Tue 2023-03-21 04:21:42 UTC; 71ms ago
     Docs: man:systemd-sysv-generator(8)
   Process: 2233 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
    Tasks: 2 (limit: 1143)
   Memory: 37.6M
      CPU: 528ms
   CGroup: /system.slice/codedeploy-agent.service
           └─2260 *codedeploy-agent: master 2260*
"" "" ""
           └─2262 *codedeploy-agent: booting child* "" "" ""
"" "" ""

Mar 21 04:21:41 ip-172-31-49-160 systemd[1]: Starting LSB: AWS CodeDeploy Host Agent...
Mar 21 04:21:42 ip-172-31-49-160 codedeploy-agent[2233]: Starting codedeploy-agent:
Mar 21 04:21:42 ip-172-31-49-160 systemd[1]: Started LSB: AWS CodeDeploy Host Agent.
ubuntu@ip-172-31-49-160:~$
```

Create a Basic index page on local

```

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi index.html

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cat index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Basic Web Page</title>
  </head>
  <body>
    Hello Rushikesh!
    Day 52 Task is Completed
  </body>
</html>

```

For Run the application, we need **appspec.yaml** file

Version: specifying version
 OS: specifying Required OS
 Files: Specifying Remote file path
 Source: current Remote location
 Destination: Default Nginx index path
 hooks:
 AfterInstall:
 location: specify the nginx install script path
 timeout: after 300 sec, it terminates automatically.
 runas: Run as a Root user.
 ApplicationStart:
 location: Specify the nginx start script path.

```

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi appspec.yaml

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cat appspec.yaml
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
hooks:
  AfterInstall:
    - location: scripts/install_nginx.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_nginx.sh
      timeout: 300
      runas: root

```



```

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ mkdir scripts

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi ^C

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi install_nginx.sh

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cat install_nginx.sh
#!/bin/bash
sudo apt-get update
sudo apt-get install nginx -y

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ mv install_nginx.sh scripts/

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cd scripts/

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo/scripts (master)
$ vi start_nginx.sh

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo/scripts (master)
$ cat start_nginx.sh
#!/bin/bash
sudo service nginx start

```

Add and commit to all local changes.

```

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        appspec.yaml
        scripts/

no changes added to commit (use "git add" and/or "git commit -a")

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'appspec.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/install_nginx.sh', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/start_nginx.sh', LF will be replaced by CRLF the next time Git touches it

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   appspec.yaml
        modified:   index.html
        new file:   scripts/install_nginx.sh
        new file:   scripts/start_nginx.sh

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git commit -m "Added Appspec and Scripts"
[master 7618e98] Added Appspec and Scripts
4 files changed, 21 insertions(+)
create mode 100644 appspec.yaml
create mode 100644 scripts/install_nginx.sh
create mode 100644 scripts/start_nginx.sh

```

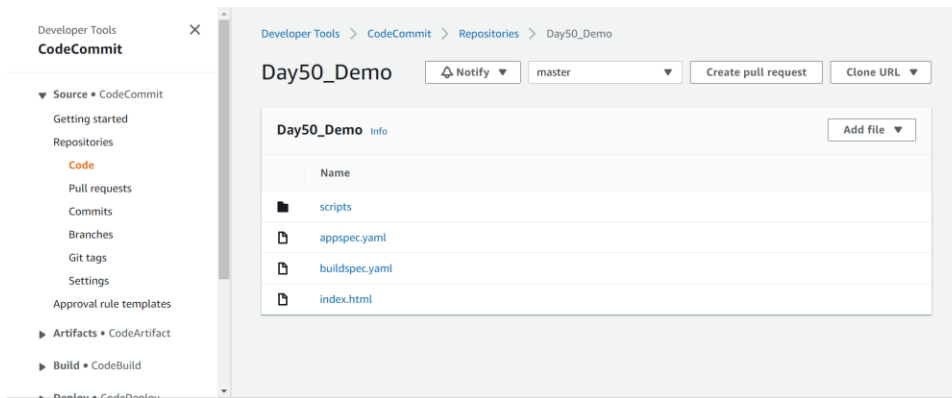
Push Committed Changes into Remote Repository.

```

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 755 bytes | 53.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Day50_Demo
4620fbb..7618e98  master -> master

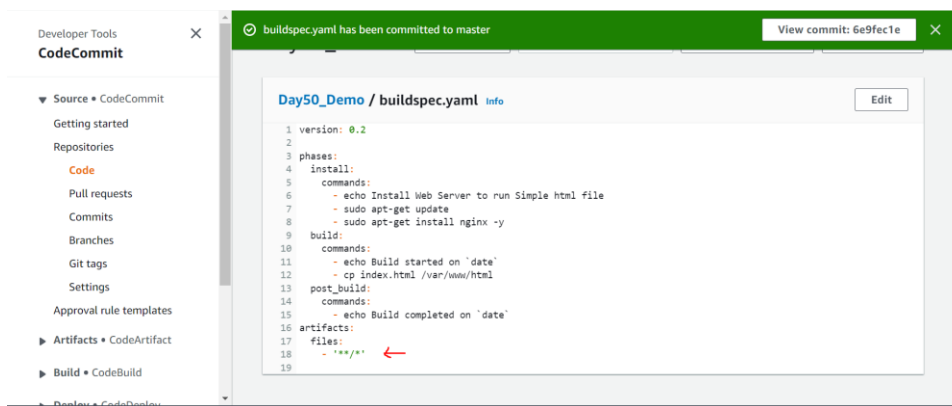
```

All local files are successfully pushed to Code Commit Repo.

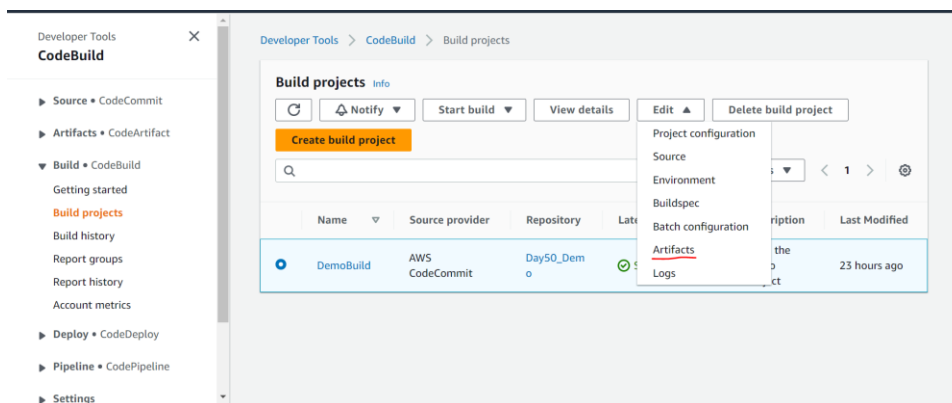


Change the **buildSpec** files.

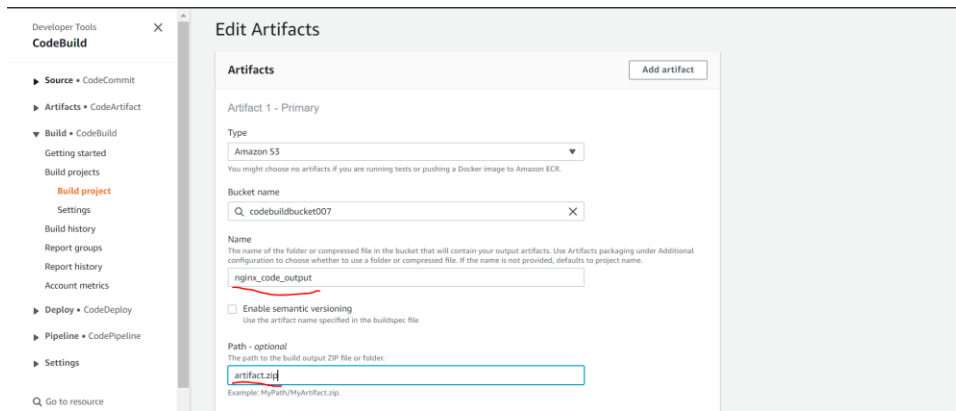
“**/*” means add all files in an artifact.



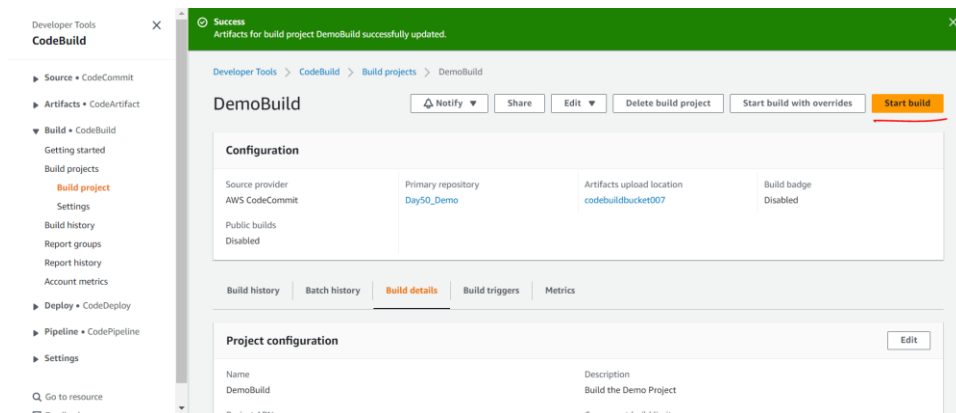
Edit the Artifacts



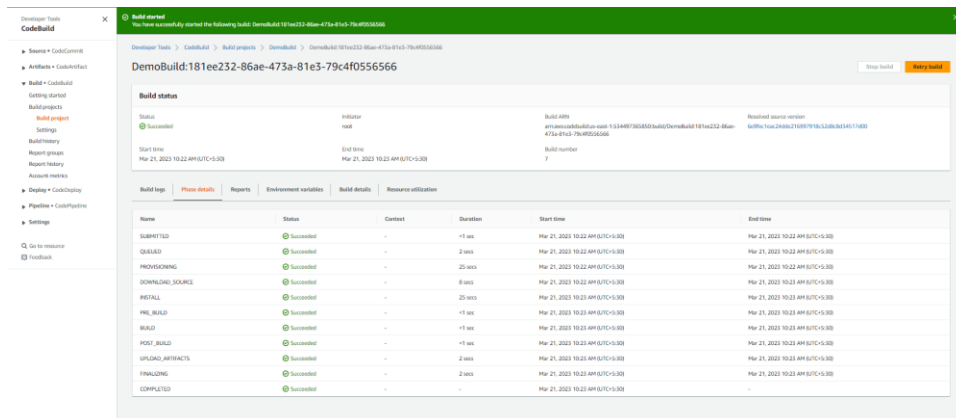
Add Artifact Name and path



Click on Start Build



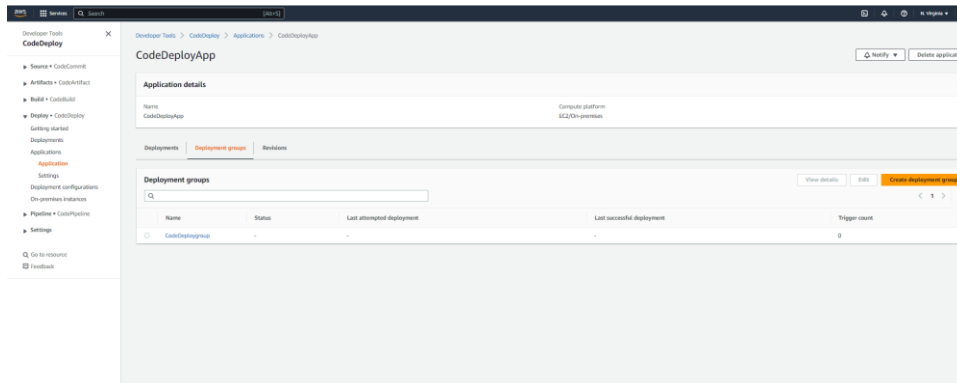
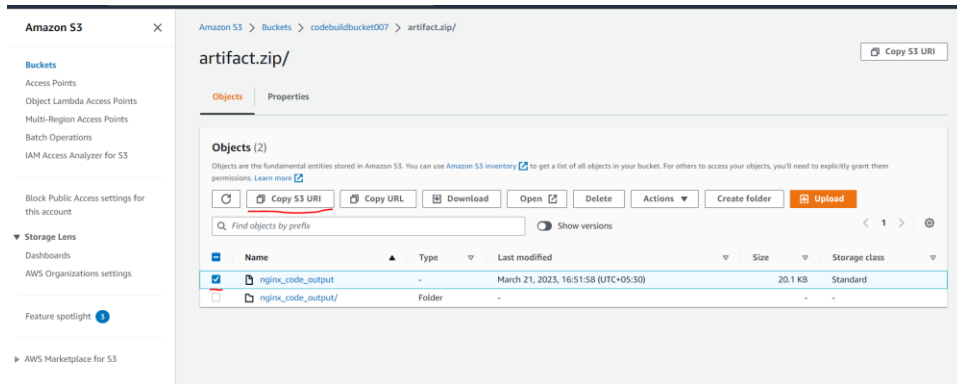
Build Successful



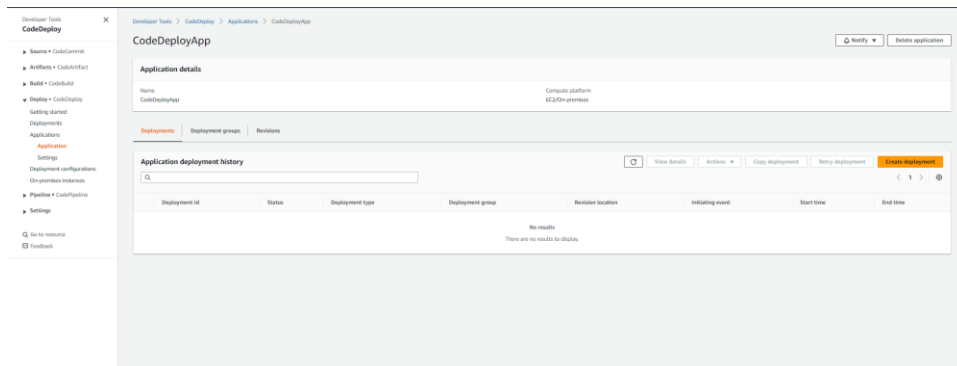
The Third step is creating a Deployment.

For this deployment, need Artifact URI.

Goto s3 and copy the S3 URI.



Click on Create Deployment.



Add Revision location and Revision type.

Developer Tools > CodeDeploy > Applications > CodeDeployApp > Create deployment

Create deployment

Deployment settings

Application
CodeDeployApp

Deployment group
demo-app-deploygrp

Compute platform
EC2/On-premises

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://codebuildbucket007/artifact.zip/
s3://bucket-name/folder/object.[zip|tar|tgz]

Revision file type
.zip

Deployment description

Deployment description - optional
Add a brief description about the deployment

Additional deployment behavior settings

ApplicationStop lifecycle event failure - optional
Type a deployment group name
☐ Don't fail the deployment to an instance if this lifecycle event on the instance fails

Click on Create Deployment.

Developer Tools

CodeDeploy

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Getting started

Deployments

Deployment

Applications

Deployment configurations

On-premises instances

Pipeline • CodePipeline

Settings

Q Go to resource

Feedback

CodeDeploy console

Deployment description

Revision details

Revision location

Revision created

Revision description

Revision location

Revision created

Revision description

Event	Duration	Status	Error code	Start time	End time
ApplicationStop	less than one second	Succeeded	-	Mar 6, 2023 6:40 AM (UTC+0:00)	Mar 6, 2023 6:40 AM (UTC+0:00)
DownloadBundle	less than one second	Succeeded	-	Mar 6, 2023 6:40 AM (UTC+0:00)	Mar 6, 2023 6:40 AM (UTC+0:00)
BeforeInstall	less than one second	Succeeded	-	Mar 6, 2023 6:40 AM (UTC+0:00)	Mar 6, 2023 6:40 AM (UTC+0:00)
Install	less than one second	Succeeded	-	Mar 6, 2023 6:40 AM (UTC+0:00)	Mar 6, 2023 6:40 AM (UTC+0:00)
AfterInstall	16 seconds	Succeeded	-	Mar 6, 2023 6:40 AM (UTC+0:00)	Mar 6, 2023 6:40 AM (UTC+0:00)
ApplicationStart	1 second	Succeeded	-	Mar 6, 2023 6:40 AM (UTC+0:00)	Mar 6, 2023 6:40 AM (UTC+0:00)
ValidateService	less than one second	Succeeded	-	Mar 6, 2023 6:40 AM (UTC+0:00)	Mar 6, 2023 6:40 AM (UTC+0:00)

Happy Learning :)