Practical – 6

Aim: Web-application deployment on EC2 using AWS CodeDeploy & S3 services. (With CI/CD)

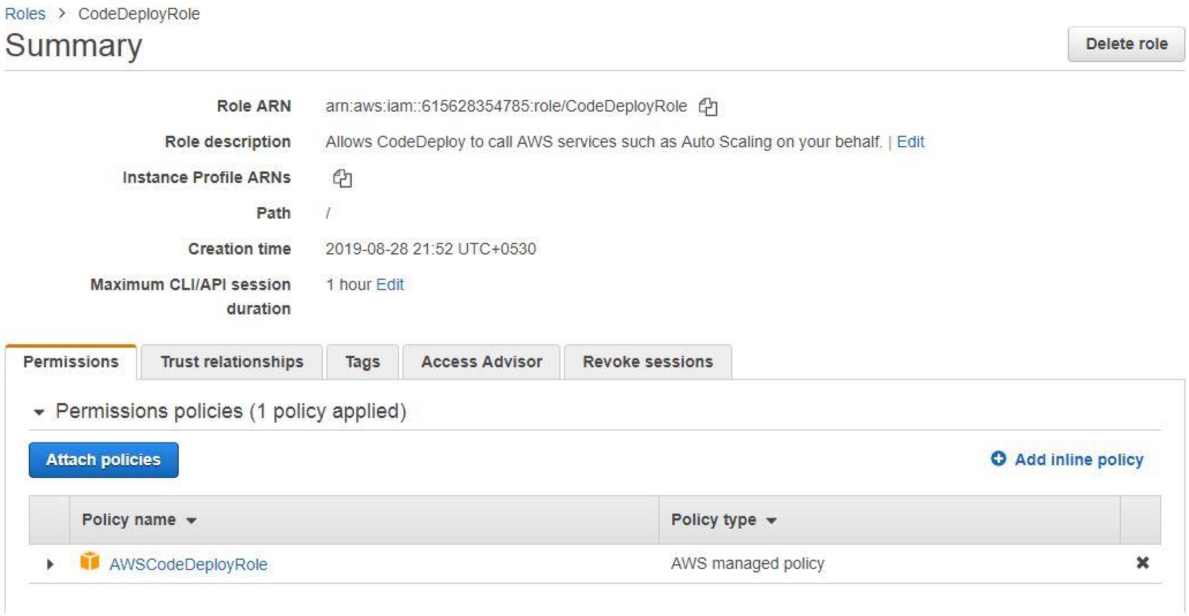
Theory:

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.
CodeDeploy makes it easier for you to:

- Rapidly release new features.
- Update AWS Lambda function versions.
- Avoid downtime during application deployment.
- Handle the complexity of updating your applications, without many of the risks associated with error-prone manual deployments.

The service scales with your infrastructure so you can easily deploy to one instance or thousands.

Output:



Roles > CodeDeployRole

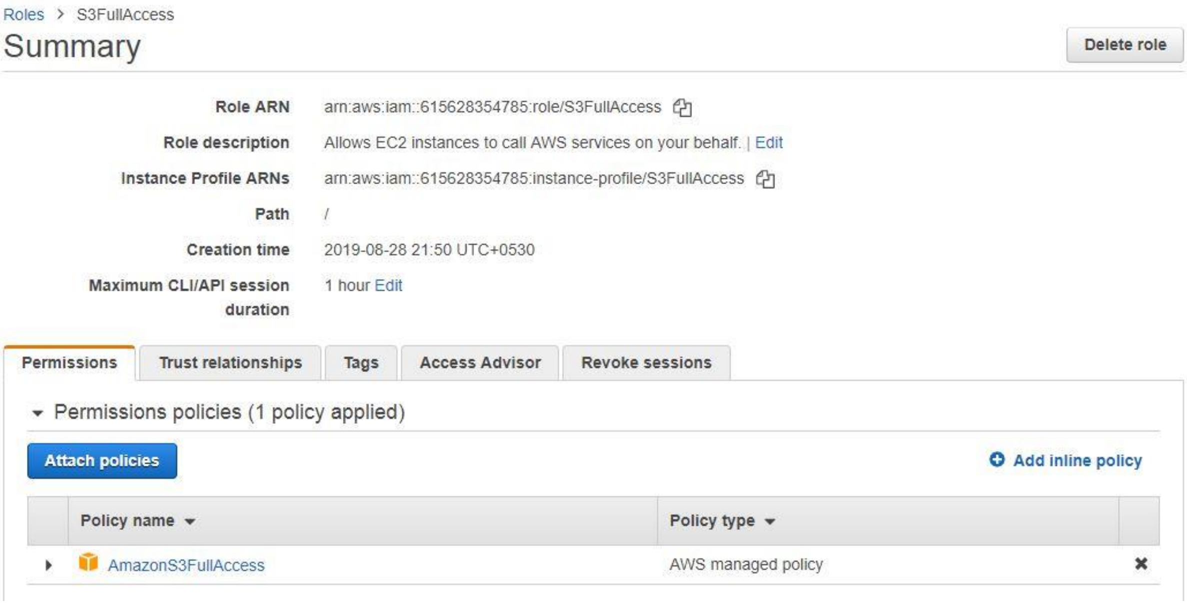
Summary

Role ARN: arn:aws:iam::615628354785:role/CodeDeployRole
Role description: Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf. | Edit
Instance Profile ARNs:
Path: /
Creation time: 2019-08-28 21:52 UTC+0530
Maximum CLI/API session duration: 1 hour Edit

Permissions policies (1 policy applied)

Policy name	Policy type
AWSCodeDeployRole	AWS managed policy

Figure 1 Code Deploy Role Summary



Roles > S3FullAccess

Summary

Delete role

Role ARN arn:aws:iam::615628354785:role/S3FullAccess

Role description Allows EC2 instances to call AWS services on your behalf. | Edit

Instance Profile ARNs arn:aws:iam::615628354785:instance-profile/S3FullAccess

Path /

Creation time 2019-08-28 21:50 UTC+0530

Maximum CLI/API session duration 1 hour Edit

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

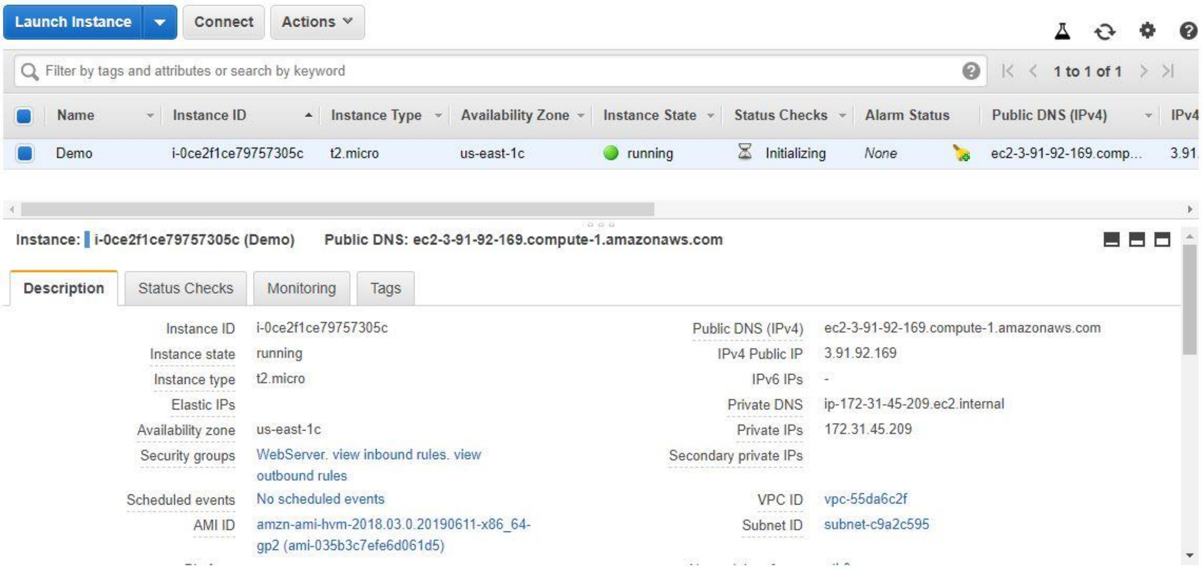
Permissions policies (1 policy applied)

Attach policies

Add inline policy

Policy name	Policy type
AmazonS3FullAccess	AWS managed policy

Figure 2 EC2 Role Summary



Launch Instance | Connect | Actions

Filter by tags and attributes or search by keyword

1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
Demo	i-0ce2f1ce79757305c	t2.micro	us-east-1c	running	Initializing	None	ec2-3-91-92-169.comp...	3.91

Instance: i-0ce2f1ce79757305c (Demo) Public DNS: ec2-3-91-92-169.compute-1.amazonaws.com

Description | Status Checks | Monitoring | Tags

Instance ID i-0ce2f1ce79757305c

Instance state running

Instance type t2.micro

Elastic IPs

Availability zone us-east-1c

Security groups WebServer. view inbound rules. view outbound rules

Scheduled events No scheduled events

AMI ID amzn-ami-hvm-2018.03.0.20190611-x86_64-gp2 (ami-035b3c7efe6d061d5)

Public DNS (IPv4) ec2-3-91-92-169.compute-1.amazonaws.com

IPv4 Public IP 3.91.92.169

IPv6 IPs -

Private DNS ip-172-31-45-209.ec2.internal

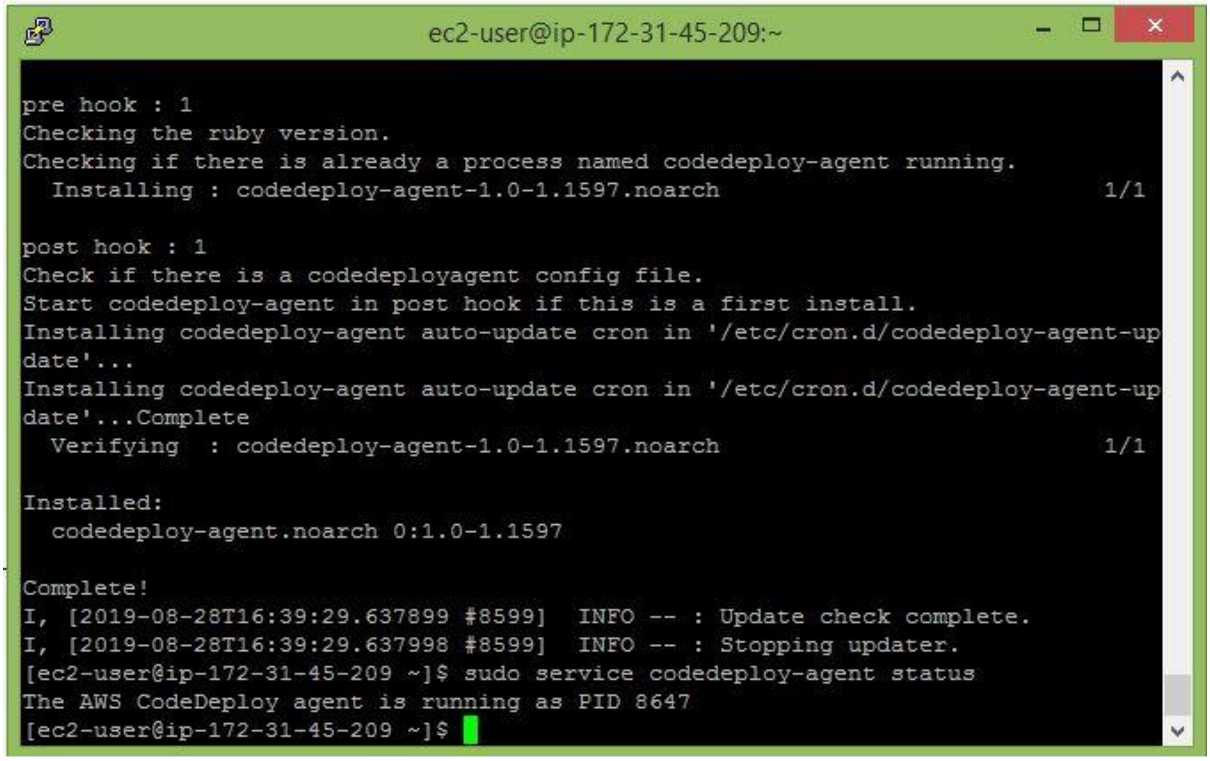
Private IPs 172.31.45.209

Secondary private IPs

VPC ID vpc-55da6c2f

Subnet ID subnet-c9a2c595

Figure 3 Launch Instance



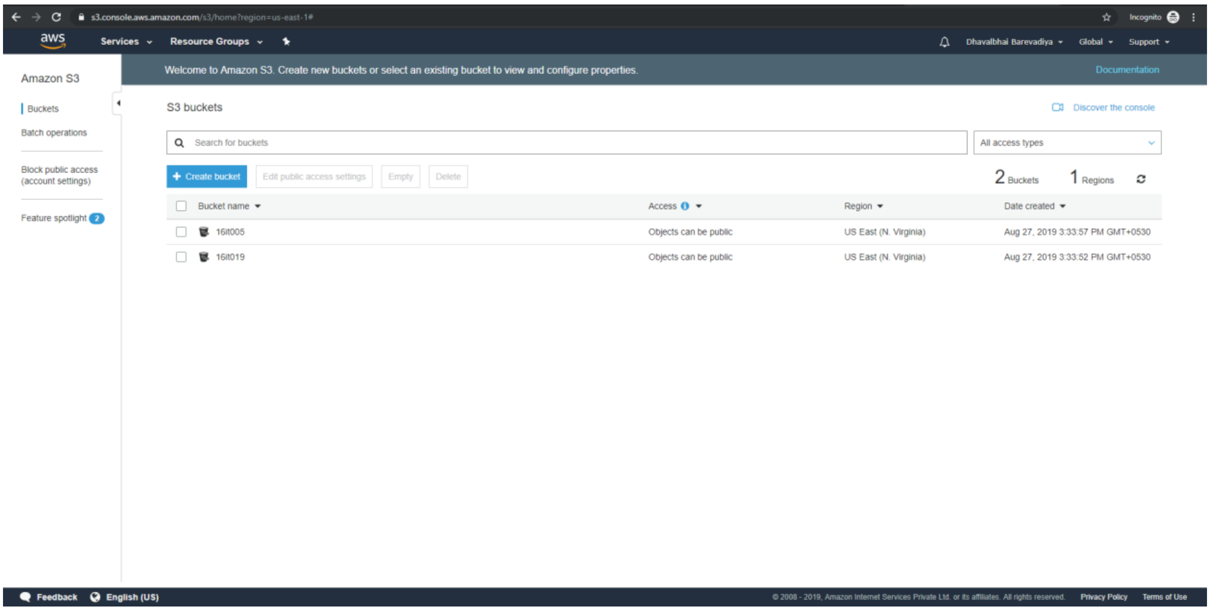
```
pre hook : 1
Checking the ruby version.
Checking if there is already a process named codedeploy-agent running.
  Installing : codedeploy-agent-1.0-1.1597.noarch                         1/1

post hook : 1
Check if there is a codedeployagent config file.
Start codedeploy-agent in post hook if this is a first install.
Installing codedeploy-agent auto-update cron in '/etc/cron.d/codedeploy-agent-up
date'...
Installing codedeploy-agent auto-update cron in '/etc/cron.d/codedeploy-agent-up
date'...Complete
  Verifying  : codedeploy-agent-1.0-1.1597.noarch                         1/1

Installed:
  codedeploy-agent.noarch 0:1.0-1.1597

Complete!
I, [2019-08-28T16:39:29.637899 #8599]  INFO -- : Update check complete.
I, [2019-08-28T16:39:29.637998 #8599]  INFO -- : Stopping updater.
[ec2-user@ip-172-31-45-209 ~]$ sudo service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 8647
[ec2-user@ip-172-31-45-209 ~]$
```

Figure 4 Instance login through PUTTY



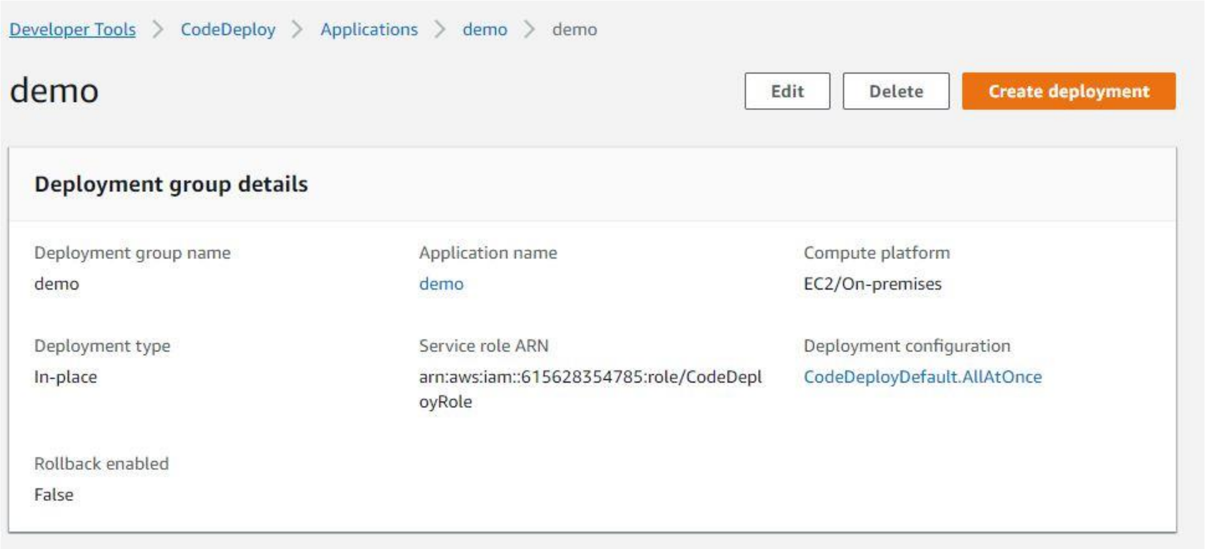
Welcome to Amazon S3. Create new buckets or select an existing bucket to view and configure properties.

S3 buckets

2 Buckets 1 Regions

□	Bucket name	Access	Region	Date created
□	16it005	Objects can be public	US East (N. Virginia)	Aug 27, 2019 3:33:57 PM GMT+0530
□	16it019	Objects can be public	US East (N. Virginia)	Aug 27, 2019 3:33:52 PM GMT+0530

Figure 5 S3 Bucket



demo

Deployment group details

Deployment group name
demo

Application name
demo

Compute platform
EC2/On-premises

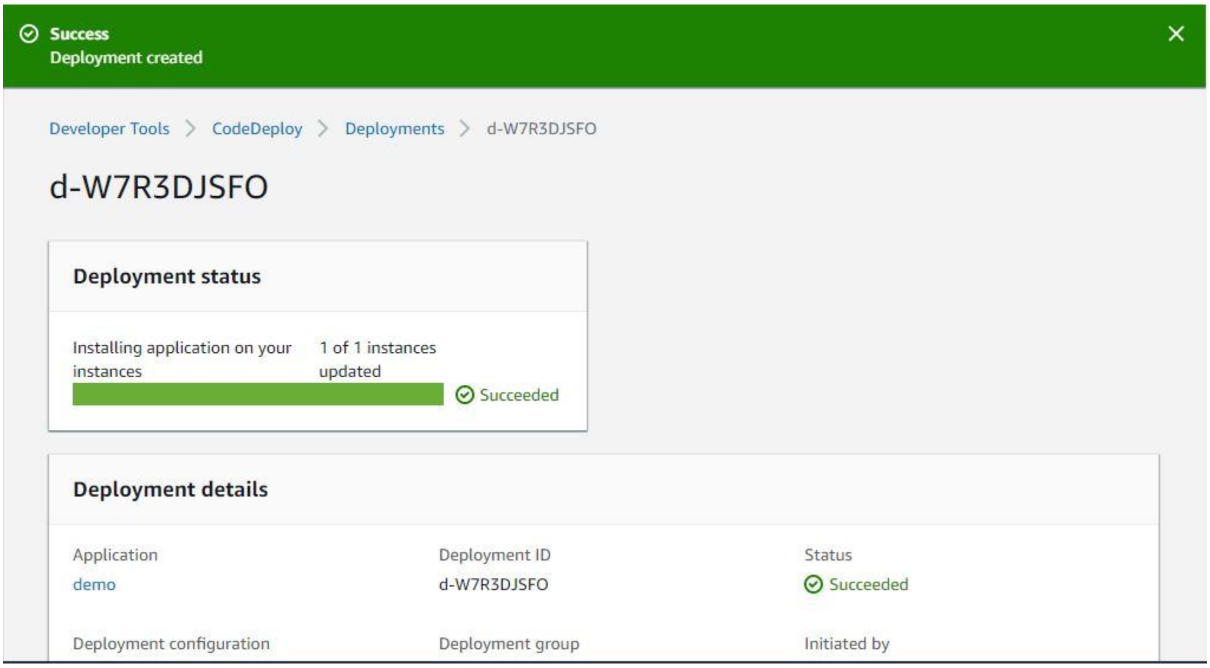
Deployment type
In-place

Service role ARN
arn:aws:iam::615628354785:role/CodeDeployRole

Deployment configuration
CodeDeployDefault.AllAtOnce

Rollback enabled
False

Figure 6 Code Deploy Created



Success
Deployment created

d-W7R3DJSFO

Deployment status

Installing application on your instances

1 of 1 instances updated

✓ Succeeded

Deployment details

Application
demo

Deployment ID
d-W7R3DJSFO

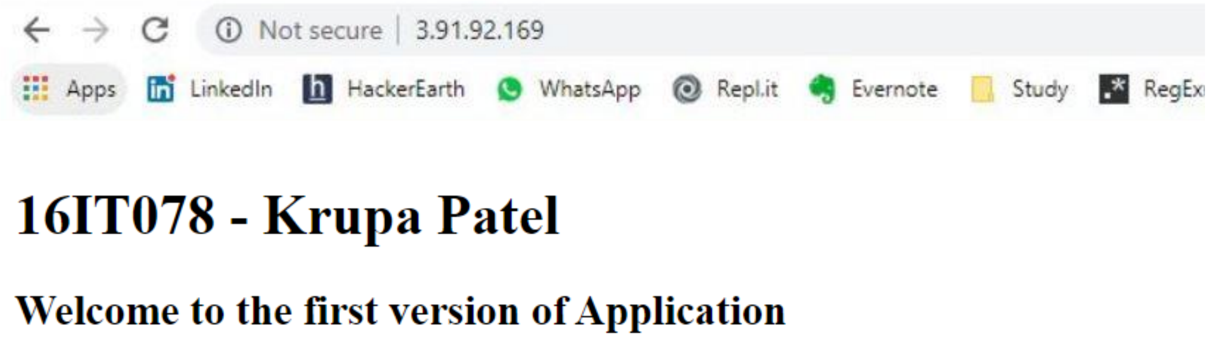
Status
✓ Succeeded

Deployment configuration

Deployment group

Initiated by

Figure 7 Success Message of deployment



16IT078 - Krupa Patel

Welcome to the first version of Application

Figure 8 Final Output

Conclusion:

We have learned the concept of CI/CD and implemented on AWS Platform with Codedeploy Service.