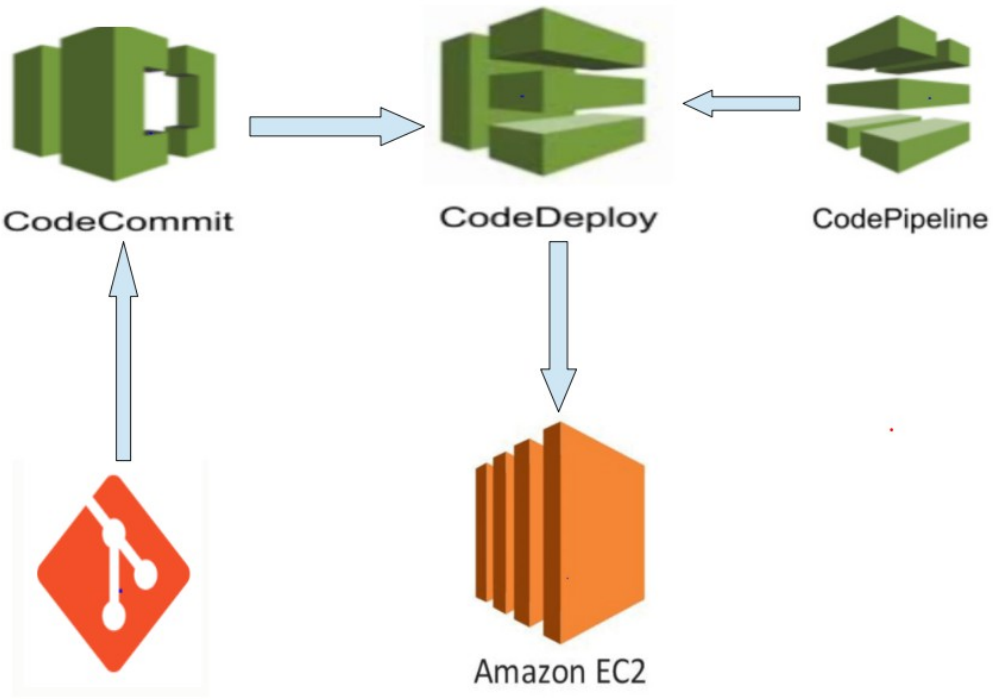
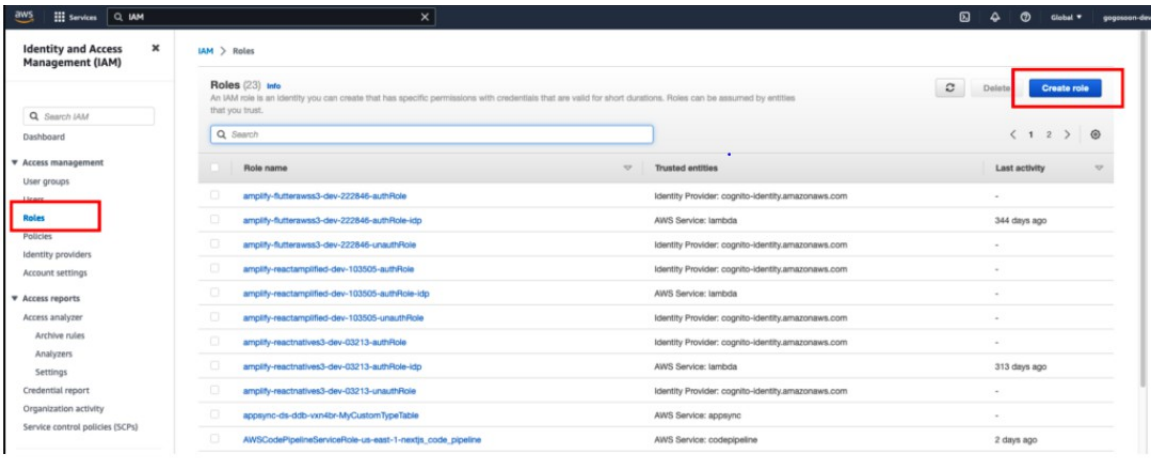
How to Setup Auto-Deployment using CodePipeline and CodeDeploy



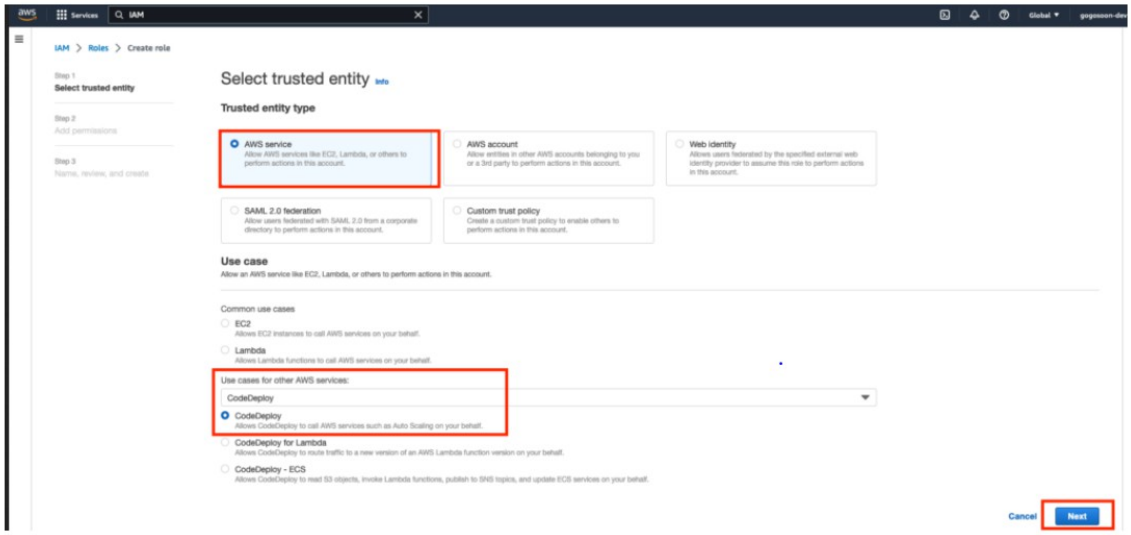
Workflow:-

Create an IAM Role for CodeDeploy:



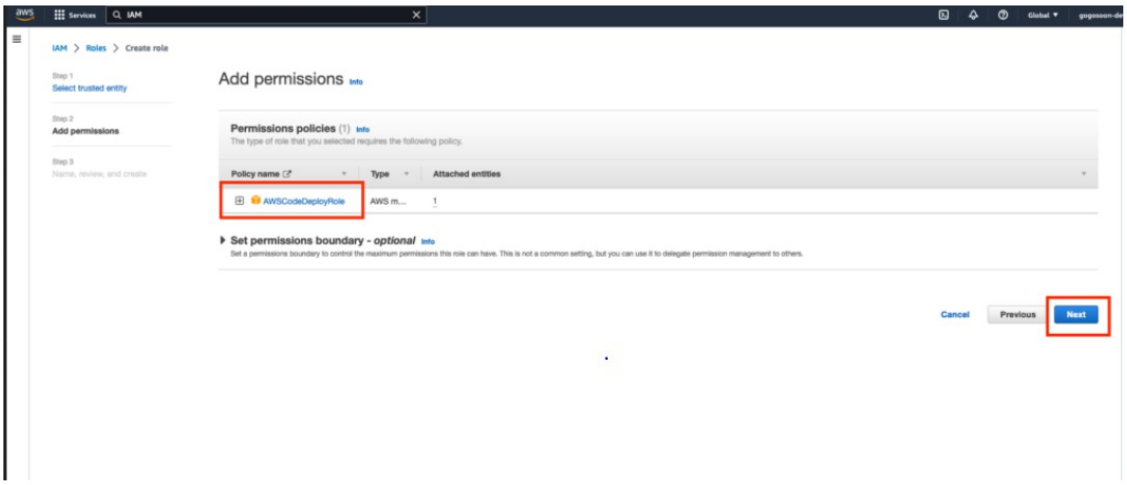
Create IAM role

Choose AWS service in Trusted entity types and choose CodeDeploy in the Use cases section and proceed to the next step.



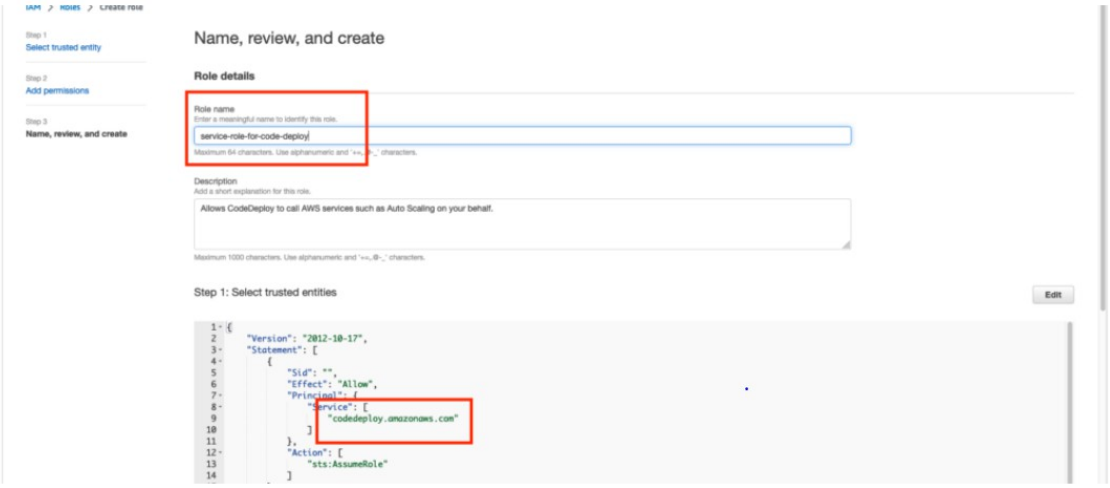
IAM role for CodeDeploy

Now, you can see that the AWSCodeDeployRole policy is the only policy available, and it'll be chosen by default in this (Permissions) step.



AWSCodeDeploy Permission

Enter a name for your IAM role. You should choose a meaningful name to identify this in the future. I'm calling it service-role-for-code-deploy.

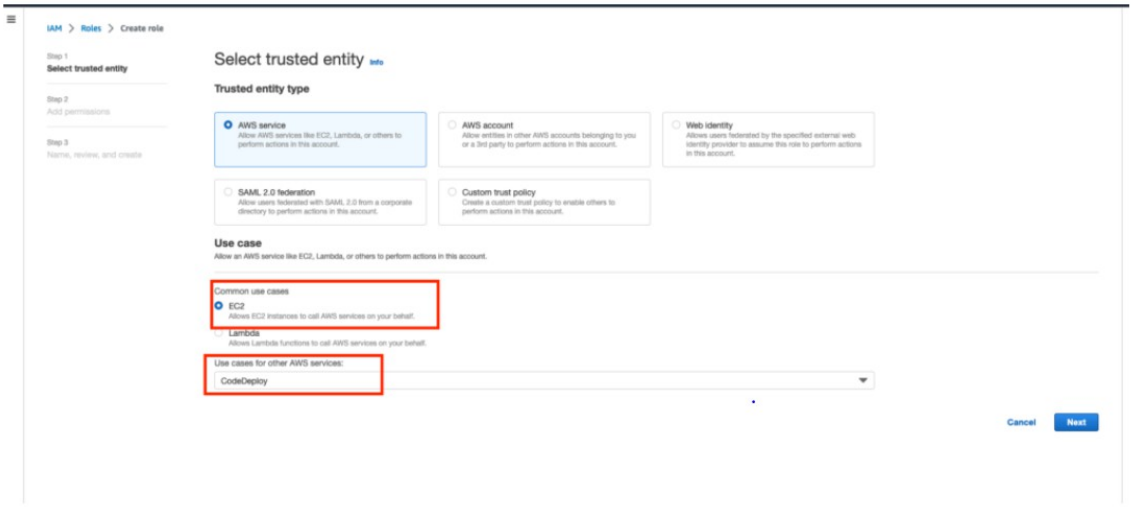


AWSCodeDeploy Permission Review

Create an IAM role for EC2

Let's create the next role. This role is for EC2. Choose AWS service in the Trusted entity type, EC2 in the Common use cases section, and choose CodeDeploy in Use cases for

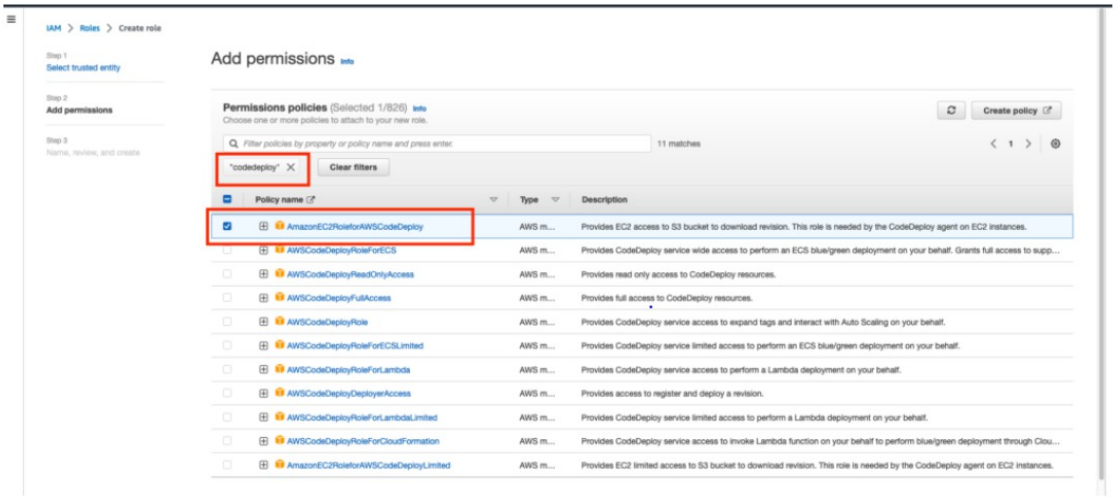
other AWS services.



IAM role for EC2

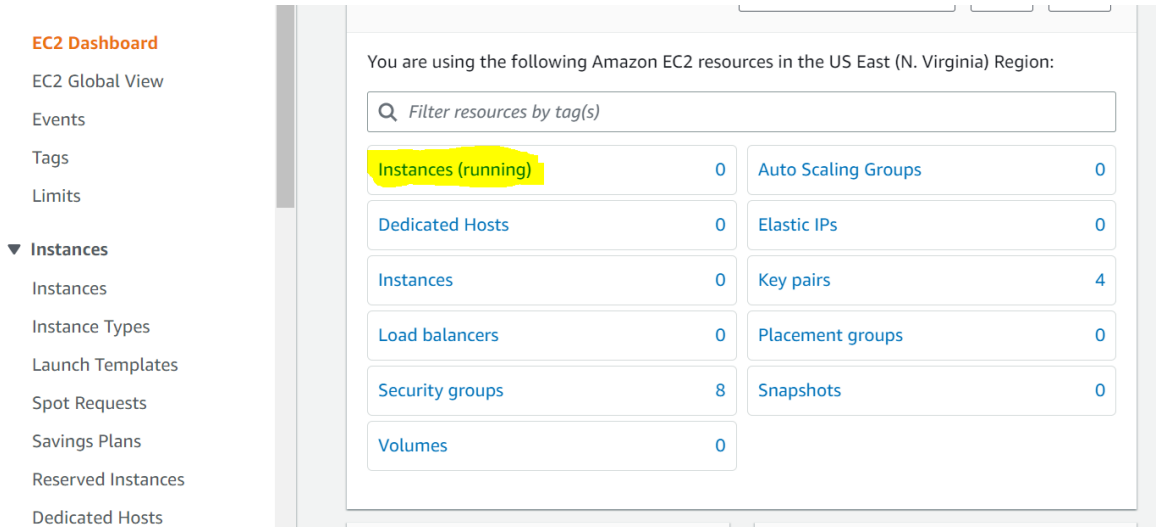
search for codedeploy

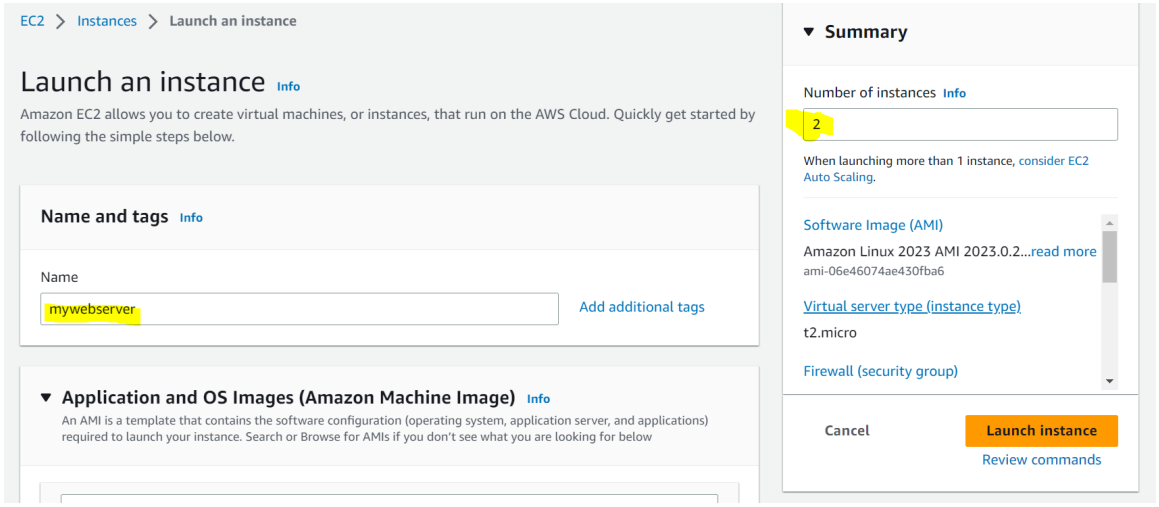
select "AmazonEC2RoleForCodeDeploy"



Adding AmazonEC2RoleForCodeDeploy permission

Launch Ec2 Instance


EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:
Filter resources by tag(s)
Instances (running) 0
Auto Scaling Groups 0
Dedicated Hosts 0
Elastic IPs 0
Instances 0
Key pairs 4
Load balancers 0
Placement groups 0
Security groups 8
Snapshots 0
Volumes 0


EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
mywebserver
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Summary
Number of instances Info
2
When launching more than 1 instance, consider EC2 Auto Scaling.
Software Image (AMI)
Amazon Linux 2023 AMI 2023.0.2...read more
ami-06e46074ae430fba6
Virtual server type (instance type)
t2.micro
Firewall (security group)
Cancel
Launch instance
Review commands

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible

ami-06e46074ae430fba6 (64-bit (x86), uefi-preferred) / ami-085a3abb84068d568 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230329.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	
64-bit (x86)	uefi-preferred	ami-06e46074ae430fba6	Verified provider

Instance type Info

Instance type — All generations

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Windows pricing: 0.0162 USD per Hour

On-Demand SUSE pricing: 0.0116 USD per Hour

On-Demand RHEL pricing: 0.0716 USD per Hour

On-Demand Linux pricing: 0.0116 USD per Hour

Compare instance types

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

key_pem

Create new key pair

Network settings Info

Edit

Network Info

vpc-0d3d739e277994ee8

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

Network Info

vpc-0d3d739e277994ee8

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

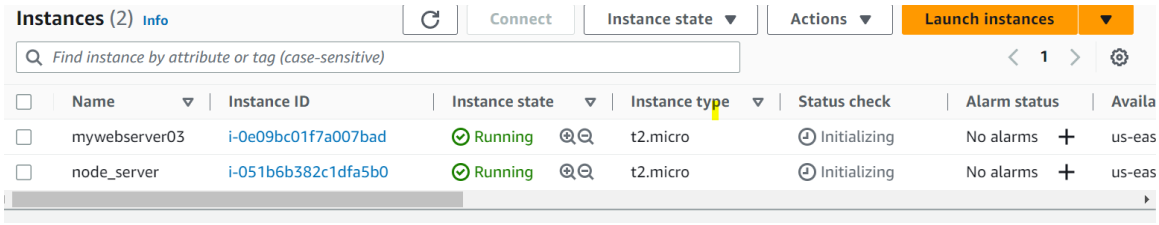
Security groups Info

Select security groups

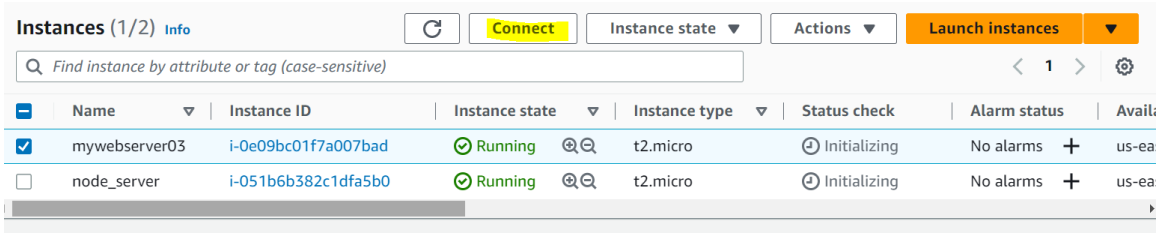
default sg-0557104f995061b47

VPC: vpc-0d3d739e277994ee8

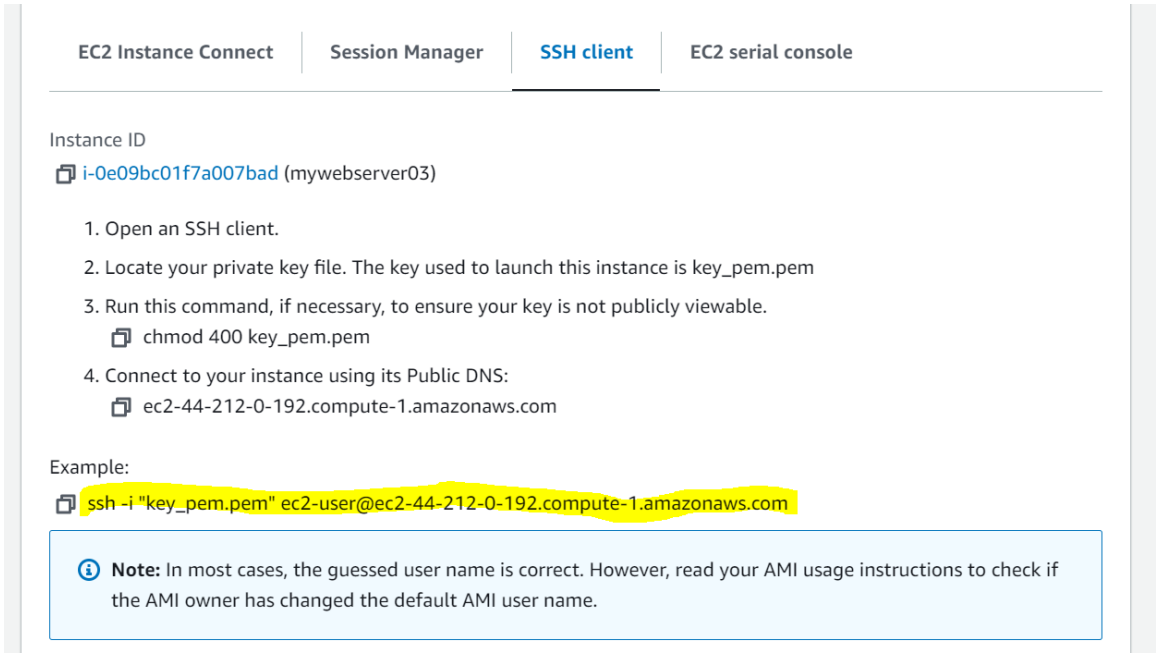
Compare security group rules



Connect Ec2 instance



copy ssh key



open git bash , putty

```
gali.radhika@AHM-LL-Radhagal MINGW64 ~
$ cd Downloads/

gali.radhika@AHM-LL-Radhagal MINGW64 ~/Downloads
$ ssh -i "key_pem.pem" ec2-user@ec2-44-212-0-192.compute-1.amazonaws.com
The authenticity of host 'ec2-44-212-0-192.compute-1.amazonaws.com (44.212.0.192)' can't be established.
ED25519 key fingerprint is SHA256:IRu952h8z+oym4iTih8f6aLnLdxpPXXwYH4IREOOkKO.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-212-0-192.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-95-194 ~]$
```

set hostname

$sudo yum install hostnamectl set-hostname localmachine

$sudo -i

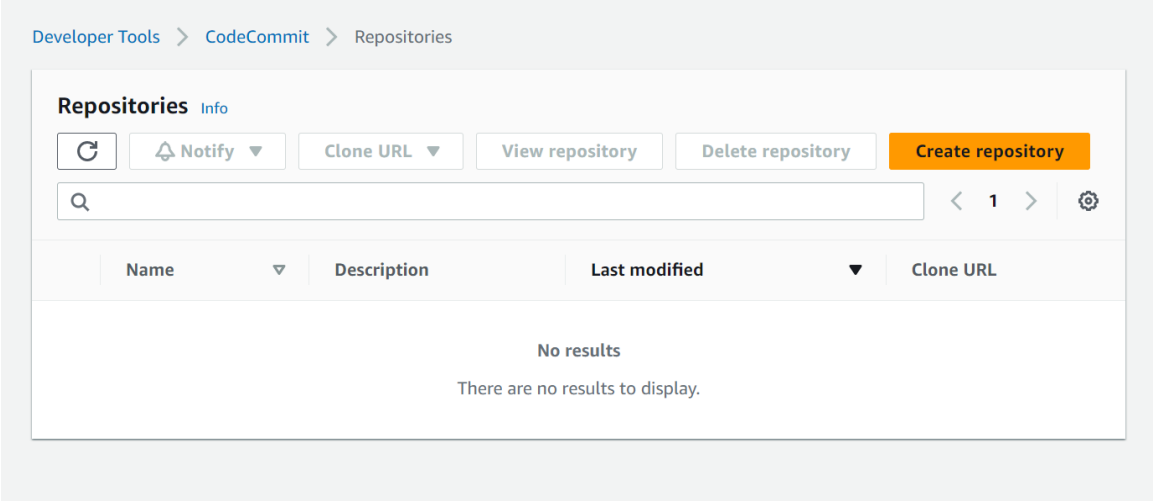
```
gali.radhika@AHM-LL-Radhagal MINGW64 ~
$ cd Downloads/

gali.radhika@AHM-LL-Radhagal MINGW64 ~/Downloads
$ ssh -i "key_pem.pem" ec2-user@ec2-44-212-0-192.compute-1.amazonaws.com
The authenticity of host 'ec2-44-212-0-192.compute-1.amazonaws.com (44.212.0.192)' can't be established.
ED25519 key fingerprint is SHA256:IRu952h8z+oym4iTih8f6aLnLdxpPXXwYH4IREOOkKO.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-212-0-192.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-95-194 ~]$ sudo hostnamectl set-hostname localmachine
[ec2-user@ip-172-31-95-194 ~]$ sudo -i
[root@localmachine ~]#
```

install git

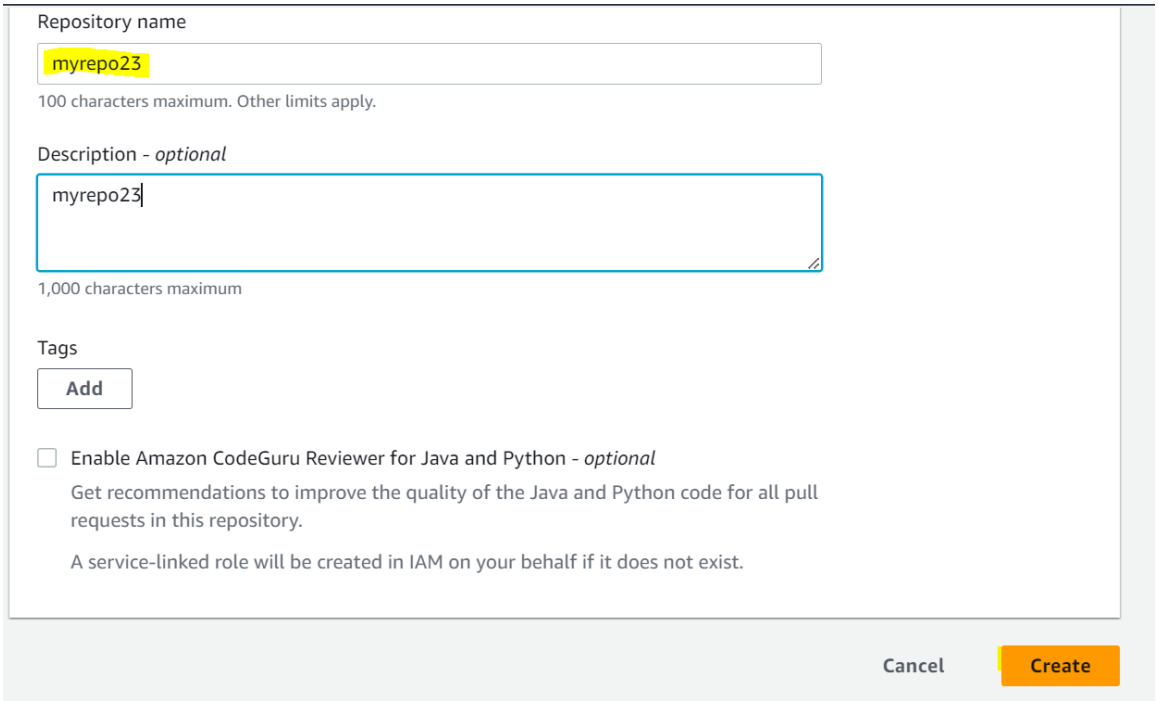
$yum install git -y

How to create Repository:

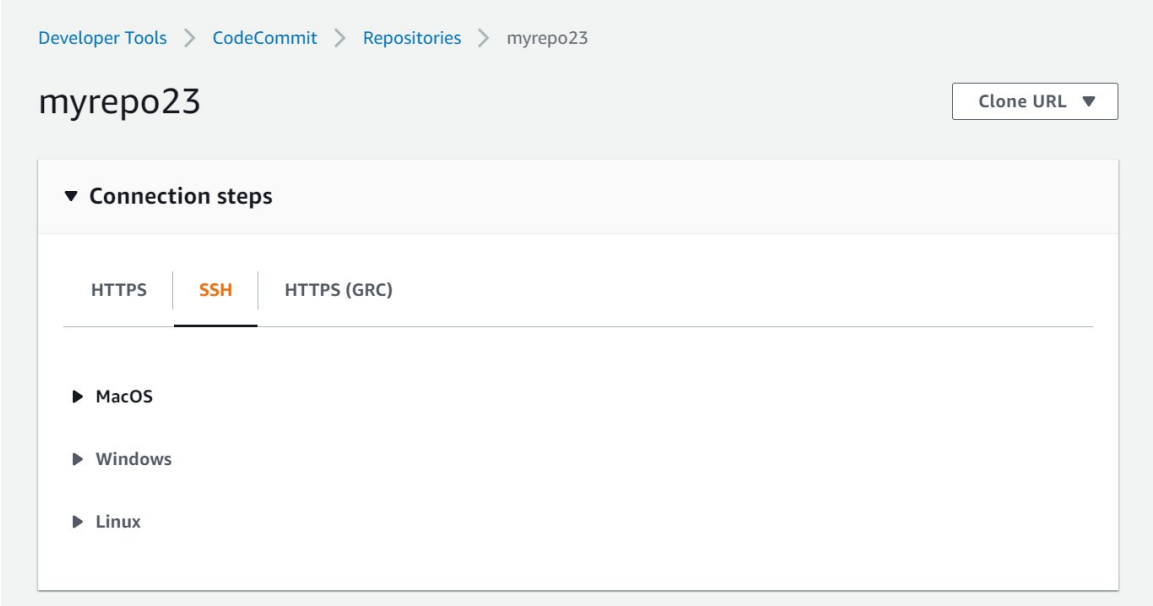
Open Aws CodeCommit



select create repository



Next conncetion steps



open Linux

Step 2: Register SSH Public Key

Upload your SSH public key to your IAM user. Learn how to upload your SSH public key

Once you have uploaded your SSH public key, copy the SSH Key ID. You will need it in the next step.

Step 3: Edit Local SSH Configuration

Edit your SSH configuration file named "config" in your local ~/.ssh directory. Add the following lines to the file, where the value for User is the SSH Key ID you copied in Step 2.

```
Host git-codecommit.*.amazonaws.com
User Your-IAM-SSH-Key-ID-Here
IdentityFile ~/.ssh/Your-Private-Key-File-Name-Here
```

Once you have saved the file, make sure it has the right permissions by running the following command in the ~/.ssh directory.

```
chmod 600 config
```

Step 4: Clone the repository

Clone your repository to your local computer and start working on code. Run the following command:

```
git clone ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myrepo23
```

Copy

open git bash check id_rsa.pub key

```
[root@localmachine ~]# cd .ssh
[root@localmachine .ssh]# ls -al
total 4
drwx------. 2 root root  29 Apr 12 11:34 .
dr-xr-x---. 4 root root 118 Apr 12 11:54 ..
-rw-------. 1 root root 555 Apr 12 11:34 authorized_keys
[root@localmachine .ssh]#
```

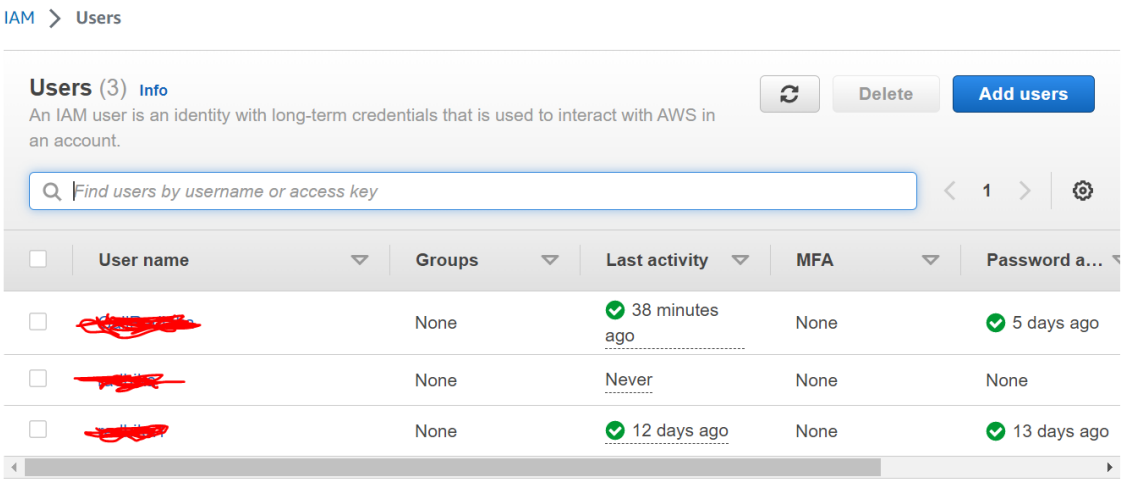
generate ssh-key

Step 2: Register SSH Public Key

$ssh-keygen

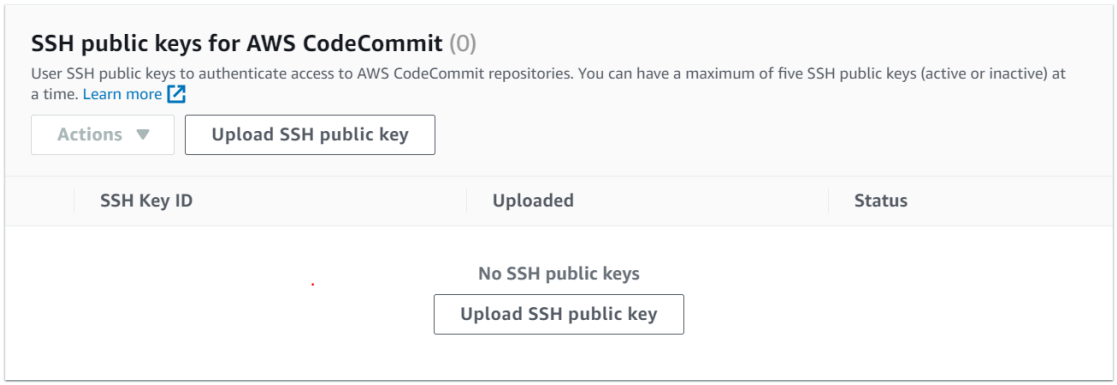
```
[root@localmachine .ssh]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:7Lk+FIOhIHeiPIJ8nDcooNa7ZeHvxoKIiC9ZOmdDU9o root@localmachine
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|o + . o          |
|*+o+o= o         |
|=* *=+E.o        |
|o.+.+.o So       |
|. o.++ ...       |
|++.o=.o.o        |
|*. o . +..       |
| o.   +oo.       |
+----[SHA256]-----+
[root@localmachine .ssh]# ls -al
total 12
drwx------. 2 root root   61 Apr 12 11:59 .
dr-xr-x---. 4 root root  118 Apr 12 11:54 ..
-rw-------. 1 root root  555 Apr 12 11:34 authorized_keys
-rw-------. 1 root root 2602 Apr 12 11:59 id_rsa
-rw-r--r--. 1 root root  571 Apr 12 11:59 id_rsa.pub
[root@localmachine .ssh]#
```

copy id_rsa.pub key

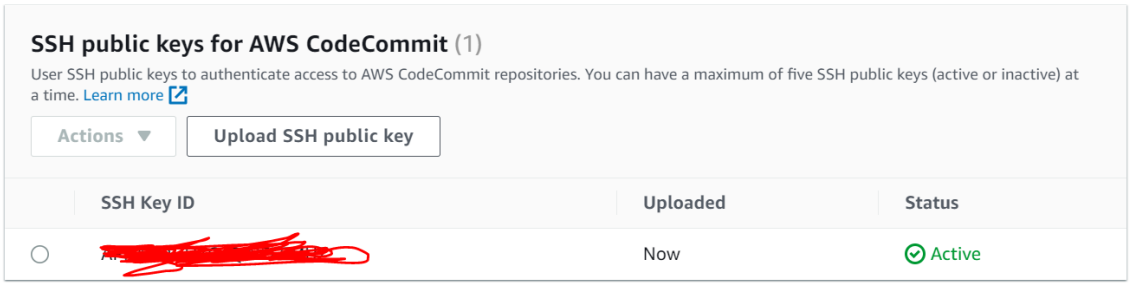


Go to IAM select user

select security credentials



upload ssh public key(id_rsa.pub key)



Step 3: Edit Local SSH Configuration

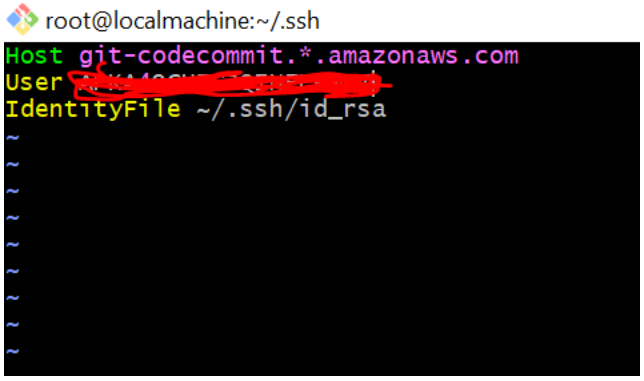
Edit your SSH configuration file named "config" in your local ~/.ssh directory. Add the following lines to the file, where the value for User is the SSH Key ID you copied in Step 2.

$vi ~/.ssh/config

Host git-codecommit.*.amazonaws.com

User Your-IAM-SSH-Key-ID-Here

IdentityFile ~/.ssh/Your-Private-Key-File-Name-Here



Step 4: Clone the repository

$mkdir project

$cd project

$mkdir codecommit

$cd Codecommit

```
drwxr-xr-x.  3 root root  24 Apr 12 11:54 project
[root@localmachine ~]# cd project
[root@localmachine project]# cd codecommit
[root@localmachine codecommit]# ls -al
total 0
drwxr-xr-x. 2 root root  6 Apr 12 11:54 .
drwxr-xr-x. 3 root root 24 Apr 12 11:54 ..
[root@localmachine codecommit]#
```

$git clone ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myrepo23

```
[root@localmachine codecommit]# git clone ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myrepo23
Cloning into 'myrepo23'...
The authenticity of host 'git-codecommit.us-east-1.amazonaws.com (52.94.229.29)' can't be established.
RSA key fingerprint is SHA256:eLMY1jODKA4uvDZcl/KgtIayZANwX6t8+8isPtotBoY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'git-codecommit.us-east-1.amazonaws.com' (RSA) to the list of known hosts.
warning: You appear to have cloned an empty repository.
[root@localmachine codecommit]#
```

$ls -al

$cd myrepo23

```
[root@localmachine codecommit]# ls -al
total 0
drwxr-xr-x. 3 root root 22 Apr 12 12:15 .
drwxr-xr-x. 3 root root 24 Apr 12 11:54 ..
drwxr-xr-x. 3 root root 18 Apr 12 12:15 myrepo23
[root@localmachine codecommit]# cd myrepo23
[root@localmachine myrepo23]# ls -al
total 0
drwxr-xr-x. 3 root root  18 Apr 12 12:15 .
drwxr-xr-x. 3 root root  22 Apr 12 12:15 ..
drwxr-xr-x. 7 root root 119 Apr 12 12:15 .git
[root@localmachine myrepo23]#
```

$vi appspec.yml

root@localmachine:~/project/codecommit/myrepo23

```
version: 0.0
os: linux
files:
  - source: /index.html
    destination: /var/www/html/
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies
      timeout: 300
      runas: root
    - location: scripts/start_server
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server
      timeout: 300
      runas: root
```

$vi index.html

```
welcome to web applicatio
```

$mkdir scripts

$vi scripts/install_dependencies

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

$vi scripts/start_server

```
#!/bin/bash
service httpd start
```

$vi scripts/stop_server

```
#!/bin/bash
isExistApp = 'pgrep httpd'
if [[ -n $isExistApp ]]; then
        service httpd stop
fi
```

$ls -al

```
root@localmachine myrepo23]# ls -al
otal 8
rwxr-xr-x. 4 root root  70 Apr 12 12:25 .
rwxr-xr-x. 3 root root  22 Apr 12 12:15 ..
rwxr-xr-x. 7 root root 119 Apr 12 12:15 .git
rw-r--r--. 1 root root 358 Apr 12 12:19 appsepc.yml
rw-r--r--. 1 root root  27 Apr 12 12:22 index.html
rwxr-xr-x. 2 root root  73 Apr 12 12:29 scripts
root@localmachine myrepo23]#
```

$git add .

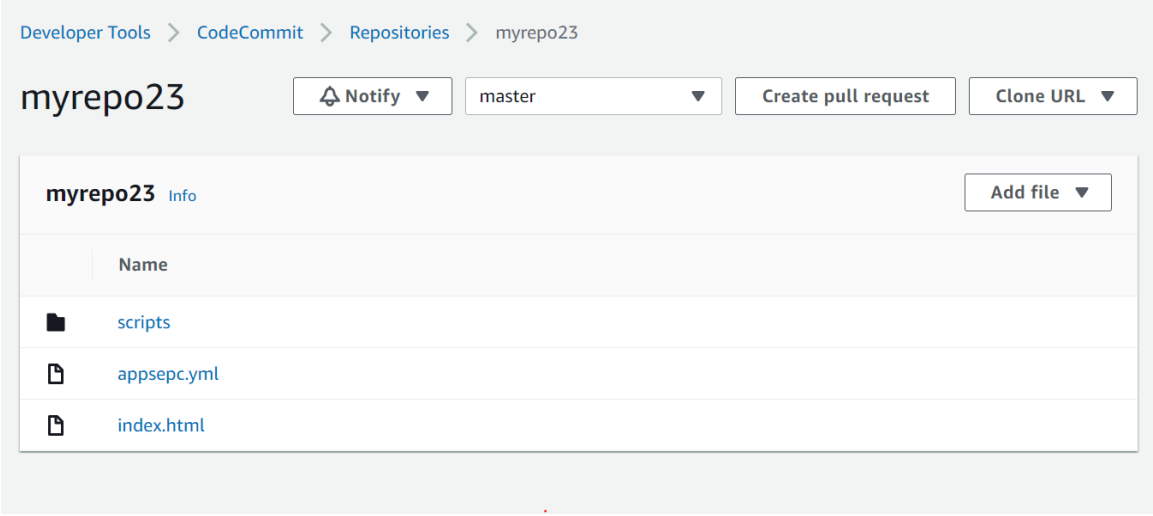
$git config --global user.email " @gmail.com"

$git config --global user.name "your name"

$git commit -m "first commit" -a

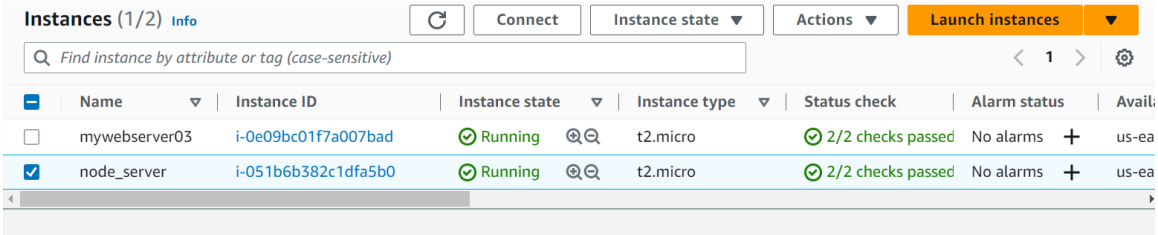
$git push origin master

```
root@localmachine myrepo23]# git add .
root@localmachine myrepo23]# git config --global user.email "[redacted]"
root@localmachine myrepo23]# git config --global user.name "[redacted]"
root@localmachine myrepo23]# git commit -m "first commit" -a
master (root-commit) b4be07d] first commit
5 files changed, 27 insertions(+)
create mode 100644 appsepc.yml
create mode 100644 index.html
create mode 100644 scripts/install_dependencies
create mode 100644 scripts/start_server
create mode 100644 scripts/stop_server
root@localmachine myrepo23]# git push origin master
numerating objects: 8, done.
ounting objects: 100% (8/8), done.
ompressing objects: 100% (5/5), done.
riting objects: 100% (8/8), 785 bytes | 785.00 KiB/s, done.
otal 8 (delta 0), reused 0 (delta 0), pack-reused 0
emote: Validating objects: 100%
o ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myrepo23
 * [new branch]      master -> master
root@localmachine myrepo23]#
```

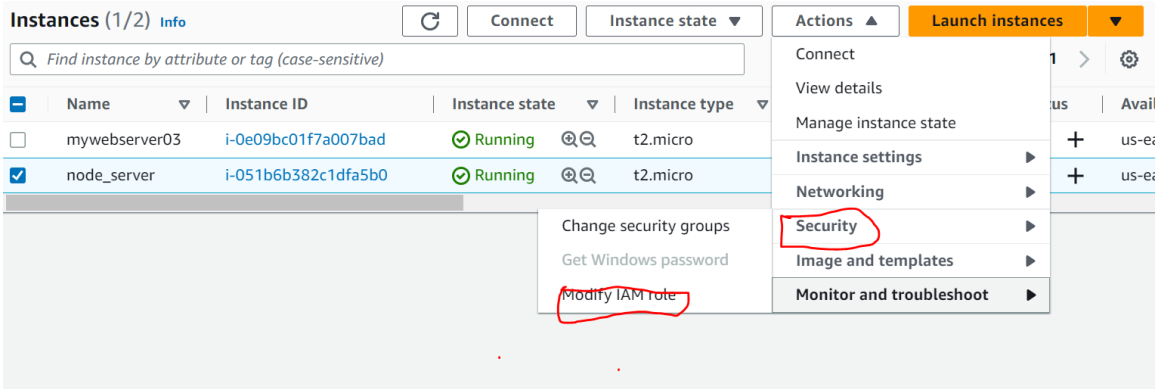


Login into another aws Ec2 instance (for application deploy):-

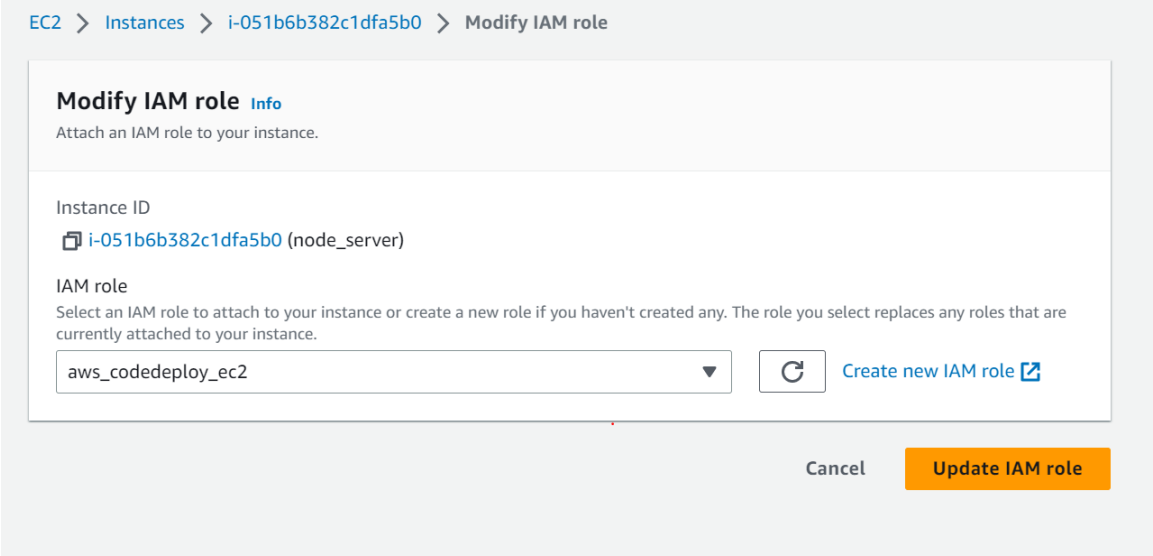
consider as a node



Attach the IAM Role to EC2



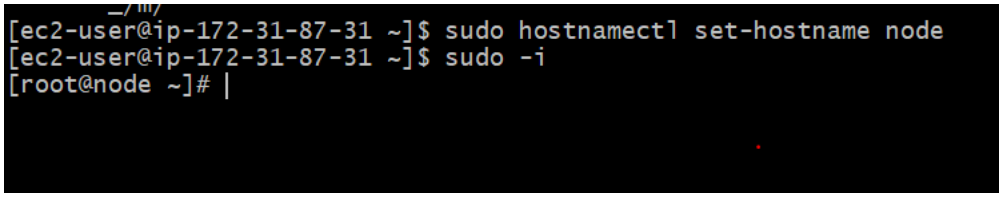
Update IAM role



Connect node

$sudo hostnamectl set-hostname node

$sudo -i



Install codedeploy-agent:-

$ sudo yum update

$sudo yum install ruby

$sudo yum install wget

$cd /home/ec2-user

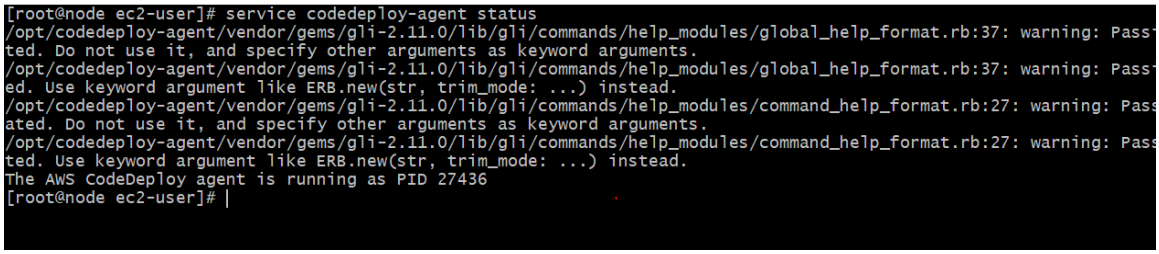
$wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install

$chmod +x ./install

$sudo ./install auto

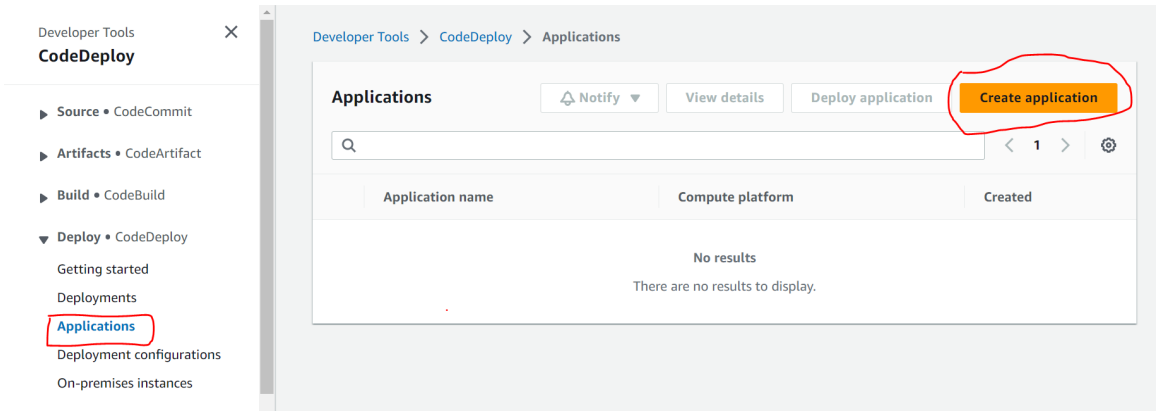
$sudo service codedeploy-agent start

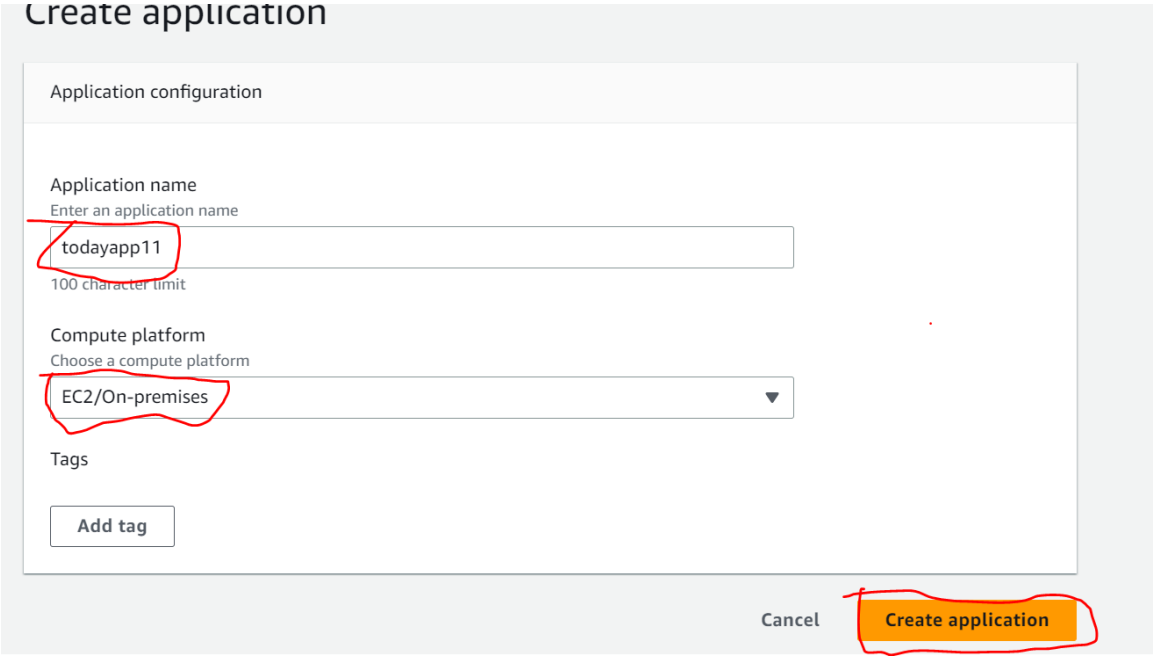
$sudo service codedeploy-gent status



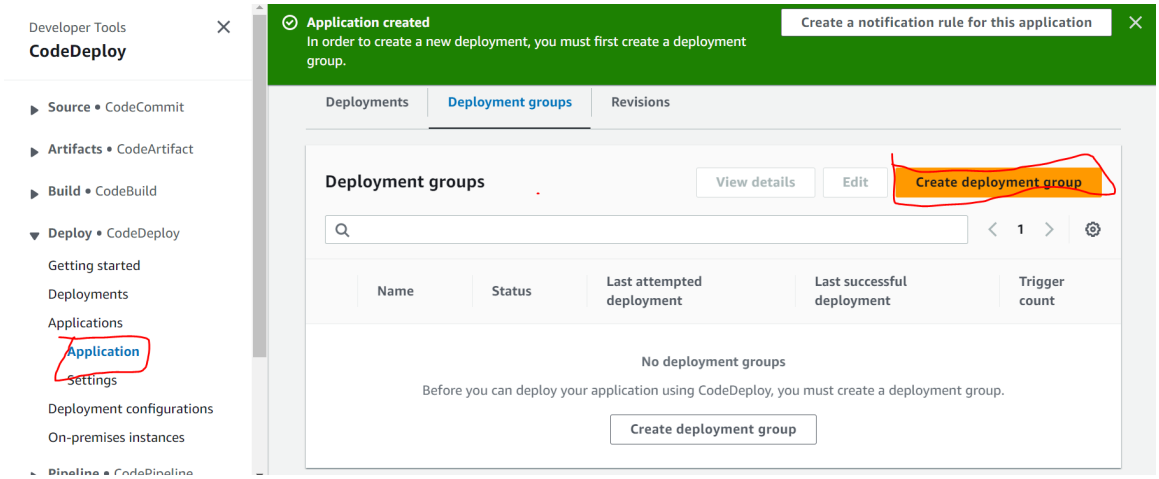
How to Create the CodeDeploy Application:-

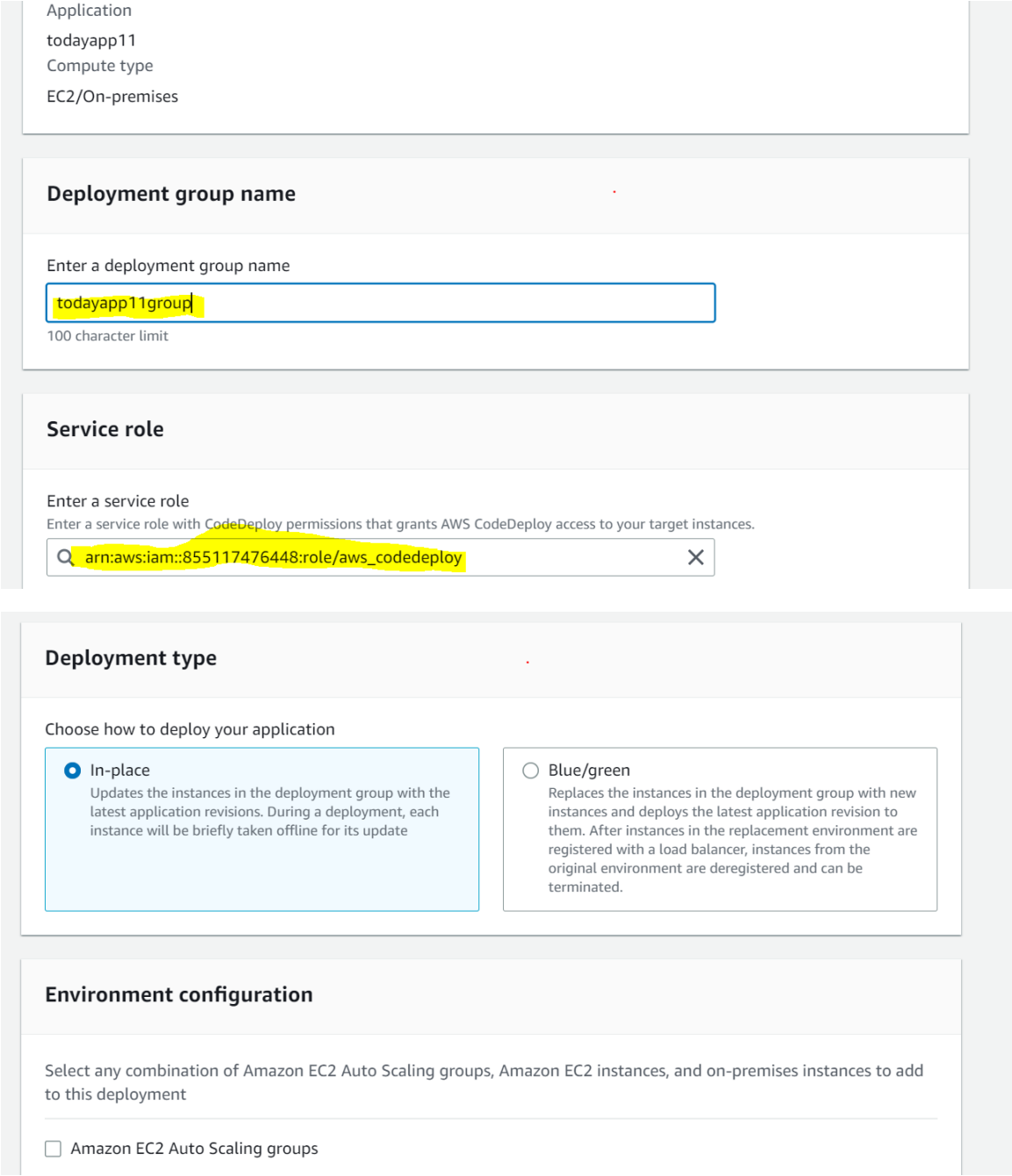
In the AWS Console, search "CodeDeploy" in the search bar at the top. Select "Applications" in the left pane. Click on the "Create application" button on the top right.





create deployment group





In the Environment configuration section, select "Amazon EC2 instances" and select the key as Name. Enter your EC2 instance name in the value.

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
- [x] 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*	
Name	node_server	Remove tag

Agent configuration with AWS Systems Manager Info

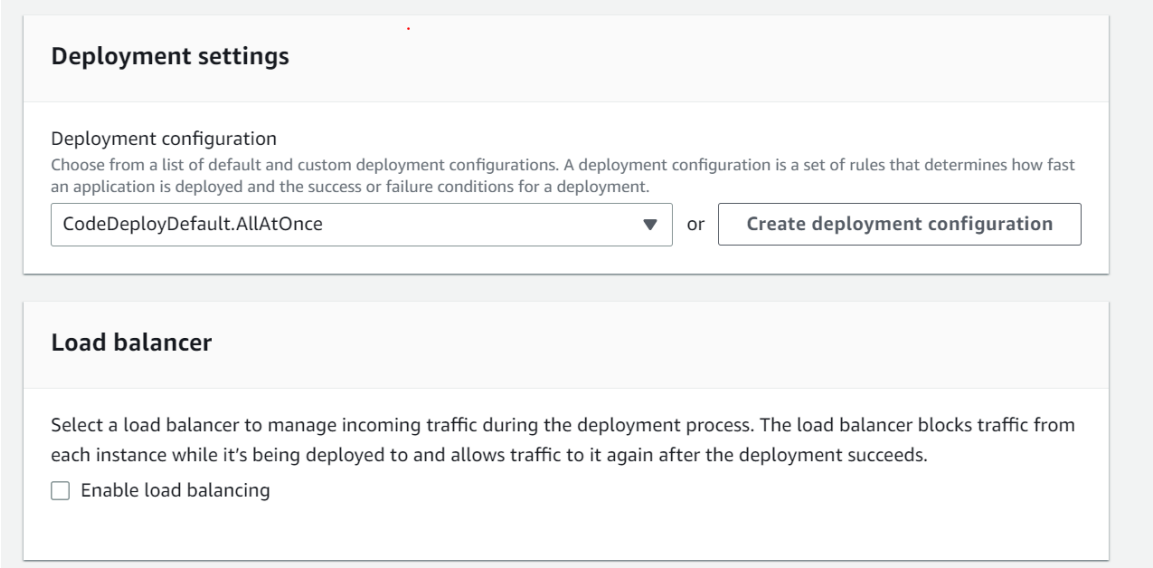
Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent. Make sure the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. Learn more

Install AWS CodeDeploy Agent

- ( ) Never
- ( ) Only once
- (•) Now and schedule updates

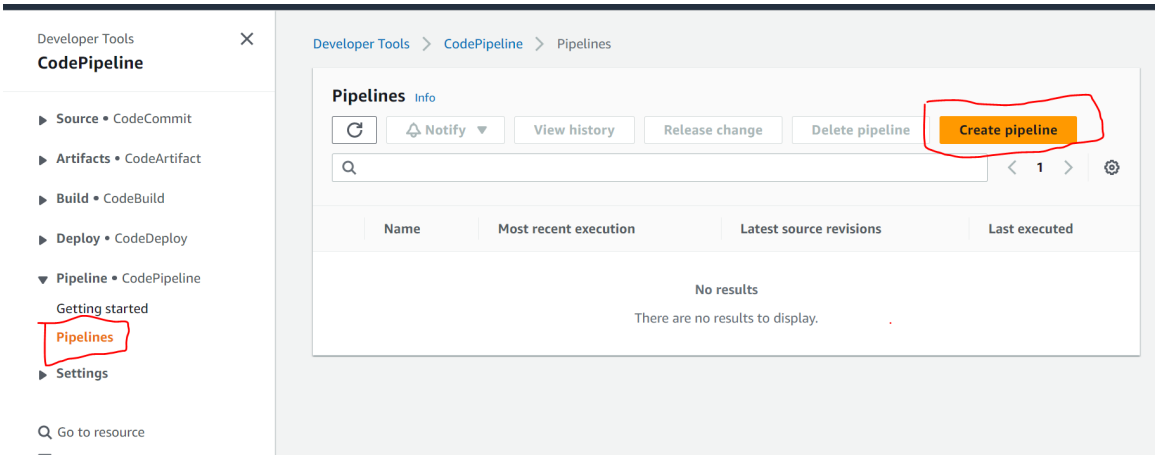
Basic scheduler | Cron expression

14 Days

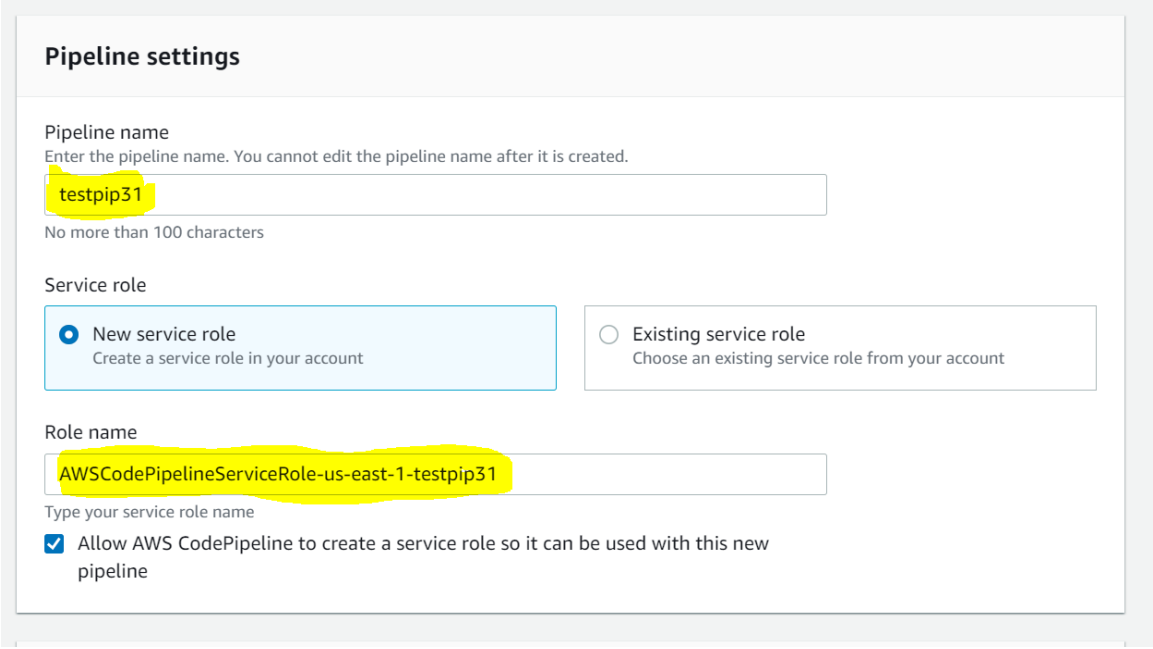


select Create deployment group

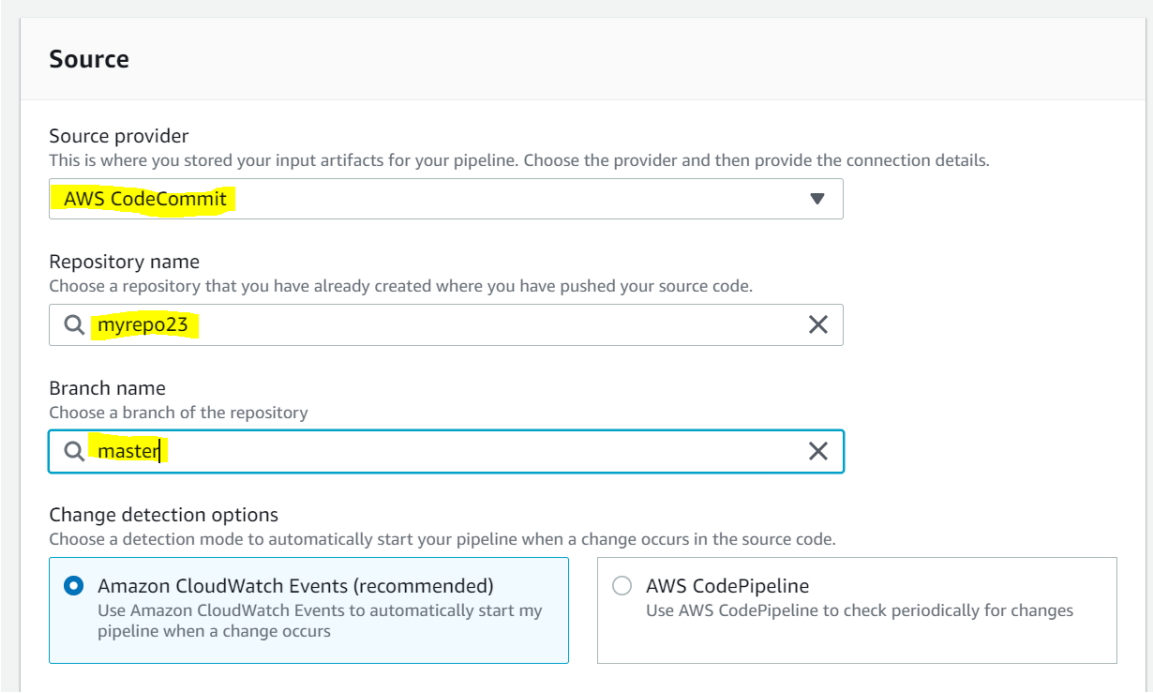
How to Create the CodePipeline:-



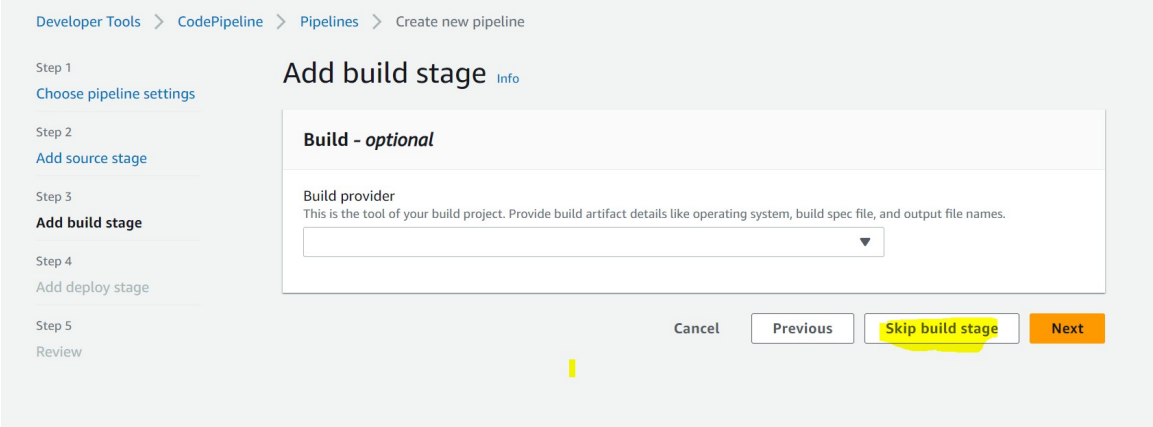
Enter the Pipeline name, and Role name. Remember, we created roles for EC2 and CodeDeploy, but not for CodePipeline. AWS by default creates it from here.

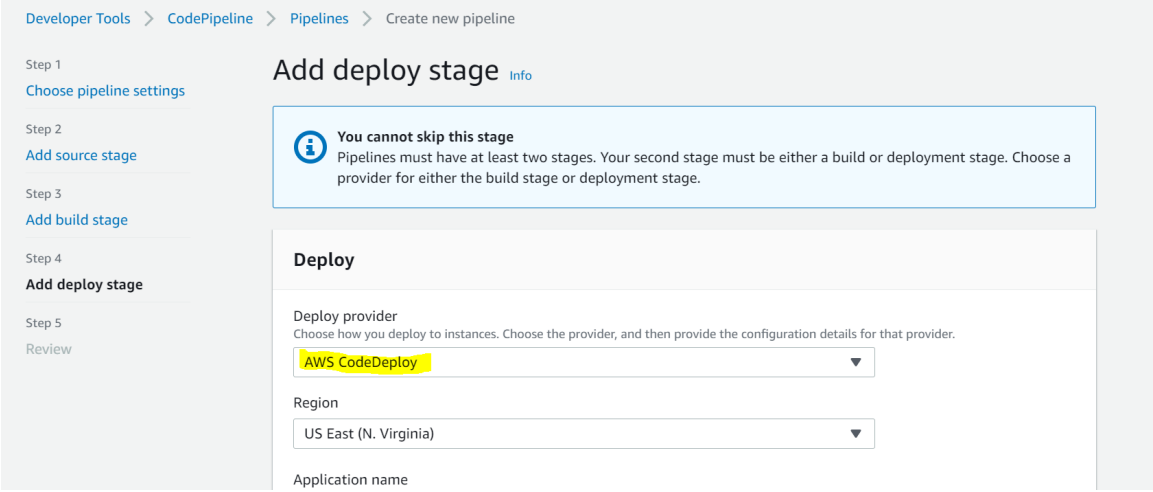


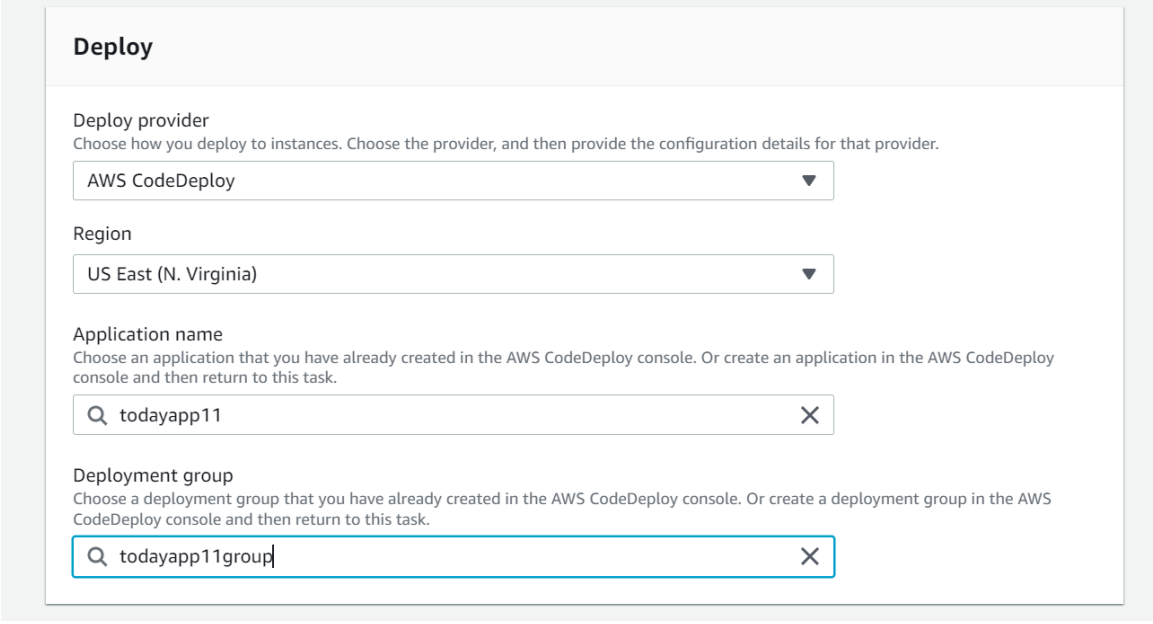
Select Next



Select Next


Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add build stage Info
Build - optional
Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
Cancel
Previous
Skip build stage
Next


Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add deploy stage Info
You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.
Deploy
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy
Region
US East (N. Virginia)
Application name


Deploy
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy
Region
US East (N. Virginia)
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.
todayapp11
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.
todayapp11group

select next

Step 4: Add deploy stage

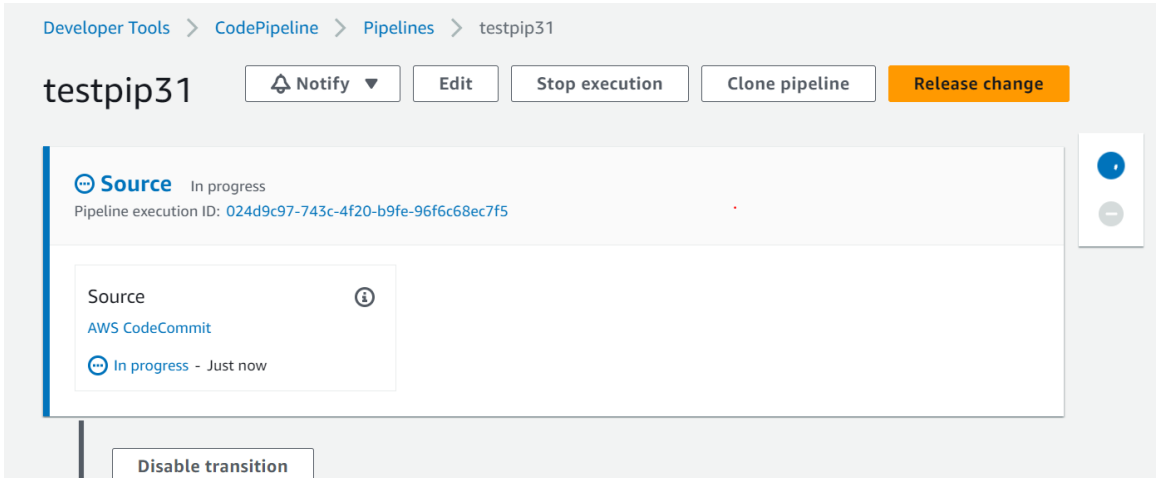
Deploy action provider

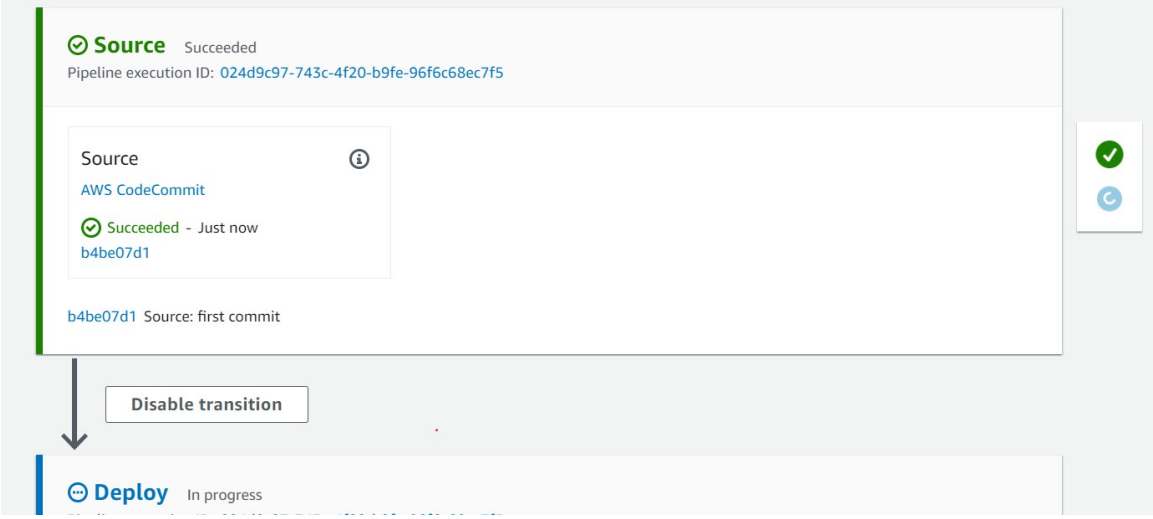
Deploy action provider
AWS CodeDeploy

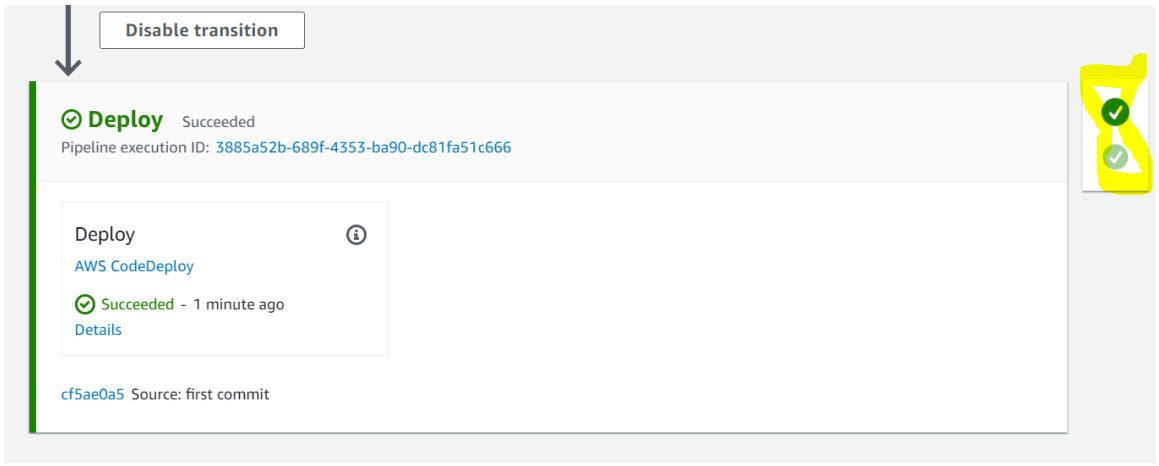
ApplicationName
todayapp11

DeploymentGroupName
todayapp11group

Cancel | Previous | Create pipeline




Source
Succeeded
Pipeline execution ID: 024d9c97-743c-4f20-b9fe-96f6c68ec7f5
Source
AWS CodeCommit
Succeeded - Just now
b4be07d1
b4be07d1 Source: first commit
Disable transition
Deploy
In progress


Disable transition
Deploy Succeeded
Pipeline execution ID: 3885a52b-689f-4353-ba90-dc81fa51c666
Deploy
AWS CodeDeploy
Succeeded - 1 minute ago
Details
cf5ae0a5 Source: first commit