**Configuration Guide for AWS EKS integration with AWS code pipeline.:**

**Required tools and Services:**

1. AWS account, GitHub Account (code commit).
2. Docker based application source code.

https://github.com/vamsi-devops/Insecure-banks.

1. Services used in AWS is (ECS, ECR, IAM, Code Build, Code Deploy, Code Pipeline).

Contents

[**1** **EKS Setup:**](#_Toc22735428)

[1.1 Create IAM User & VPC:](#_Toc22735429) 7

[1.2 Create repository in ECR.](#_Toc22735431) 8

[**2** **CI/CD implementation:**](#_Toc22735437)

[2.1 Code Build](#_Toc22735438) 12

2.1.1 Push image to ECR.

2.2 Create Ec2 instance …………………………………………………………………………………………………….13

[2.3 Code Deploy.](#_Toc22735439) 17

2.4 Code Pipeline ……………………………………………………………………………………………………………21

2.5 EKS Cluster formation………………………………………………………………………………………………….22

2.6 Output…………………………………………………………………………………………………………………………23

* 1. **VPC & IAM User Creation:**

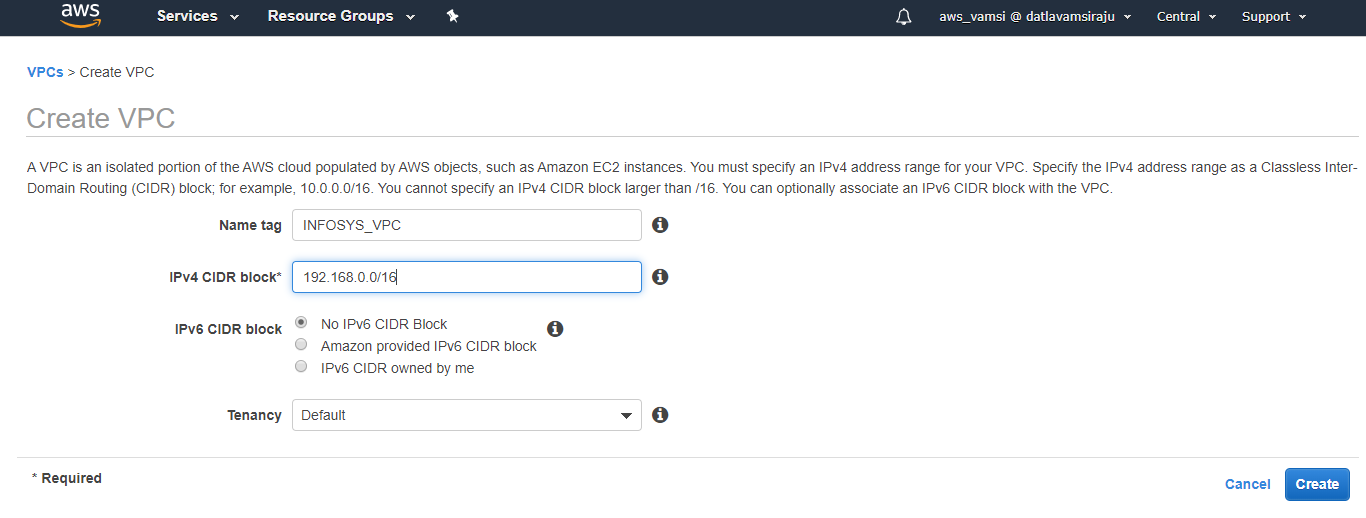
🡪 **VPC CREATION STEP BY STEP**

By default, if we create the VPC we will get

1. Route Table
2. NACL
3. SG (security group)

STEP1:

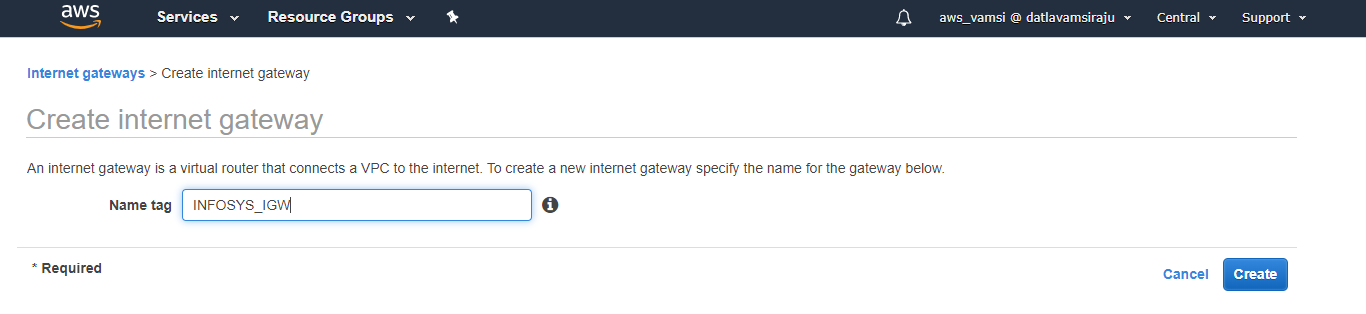
* Assign a tag name for VPC
* Take CIDR value as per your requirement (ex: 192.168.0.0/16)
* VPC has been created as per the requirement





STEP2:

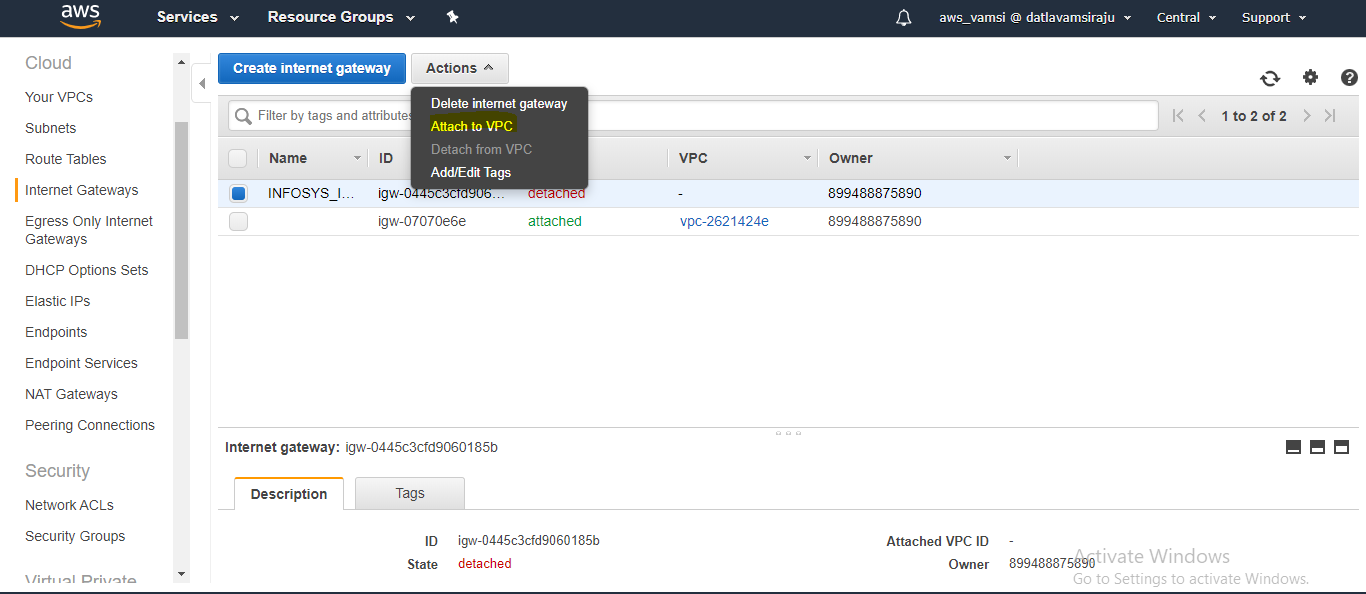
* Create IGW and assign a tag name.





STEP3:

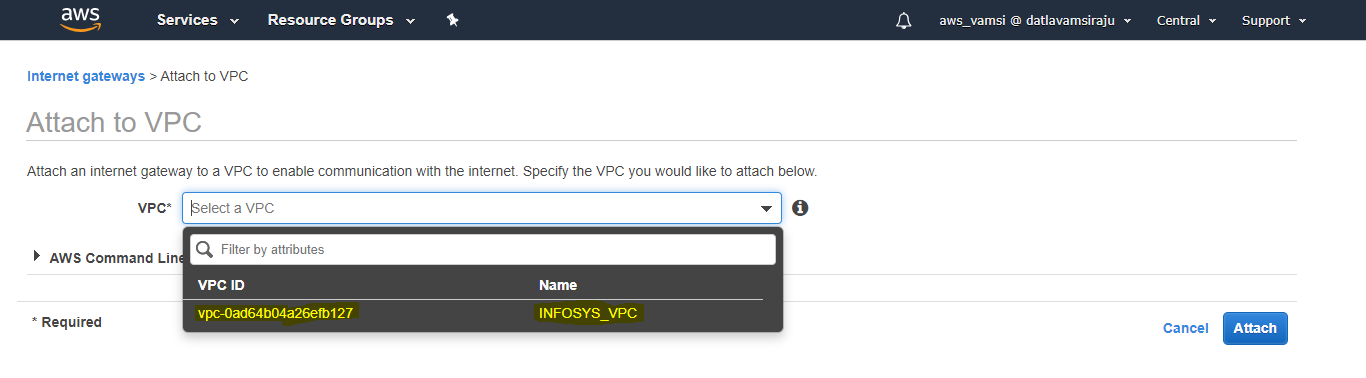
* Attach IGW to VPC to give public access to the VPC.





STEP4:

* IGW has been attached to the VPC.
* Now the isolated VPC has been exposed to public.





STEP5:

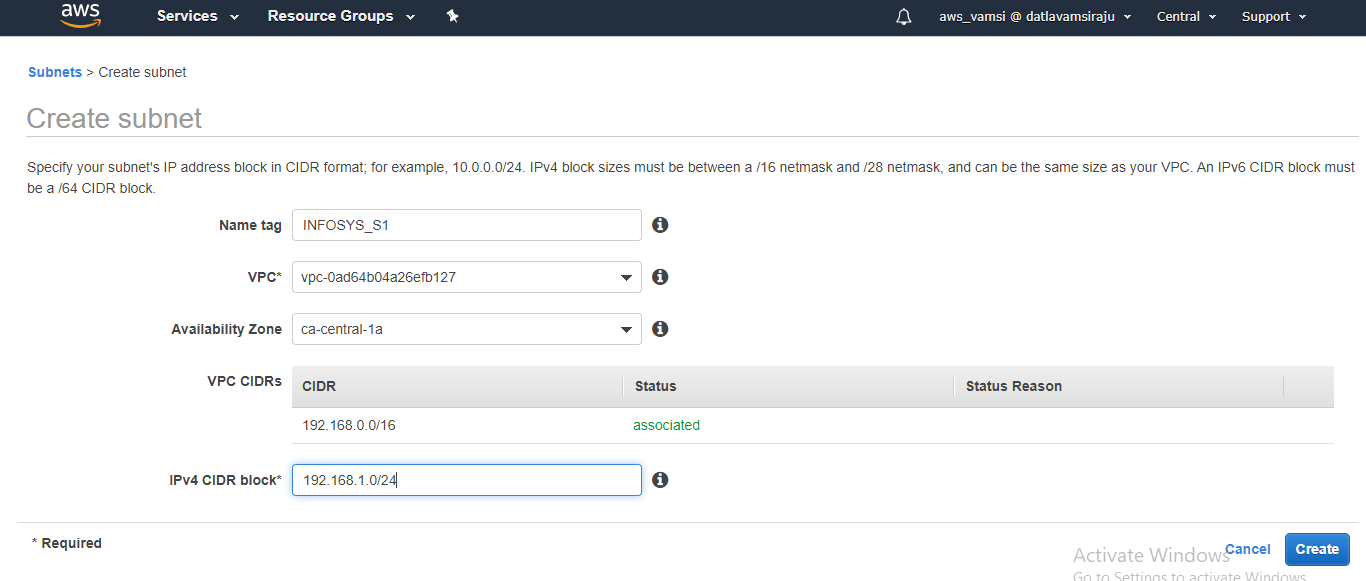
* After VPC creation and IGW attached to the VPC we have to create the SUBNET’S

1st subnet is for public

2nd subnet is for private

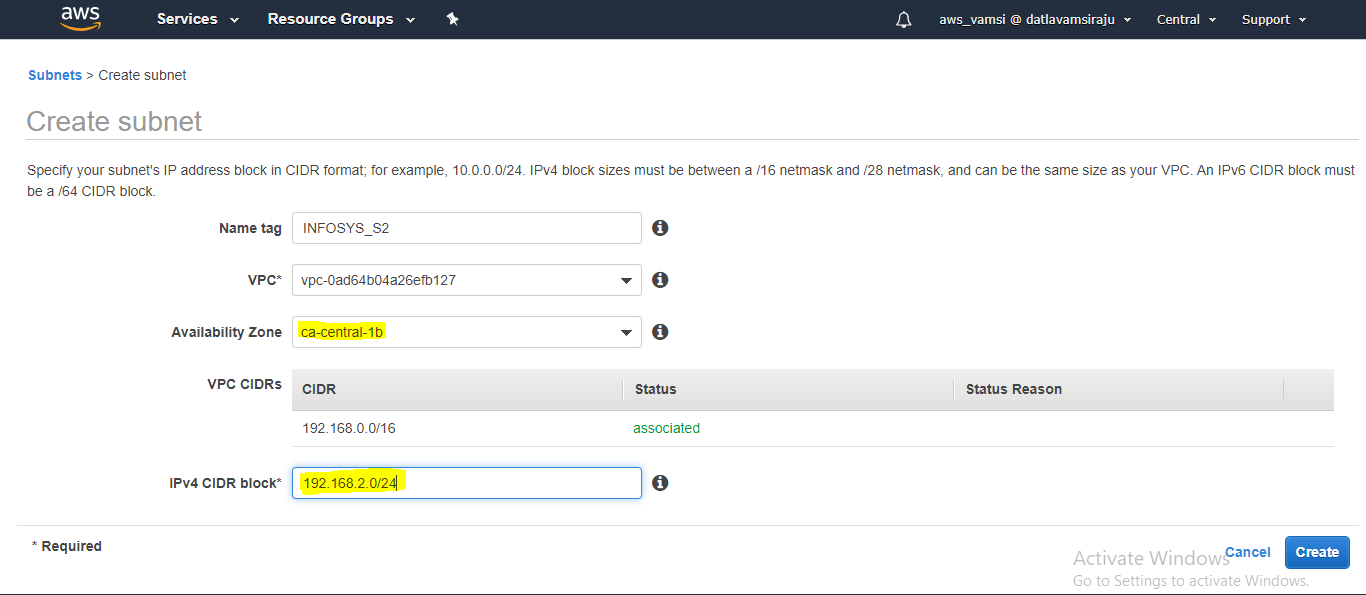
* The below are the required details for creating SUBNET1 well as SUBNET2

SUBNET1:





SUBNET2:



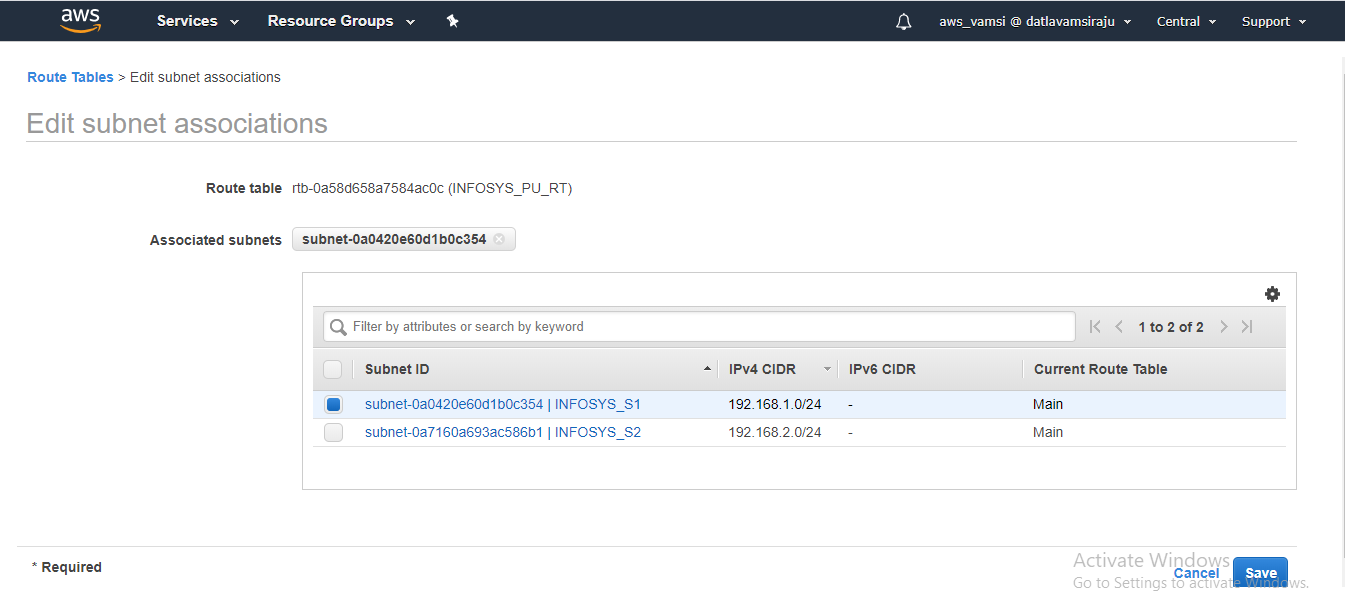


STEP6:

* Assign SUBNET to the Route table.

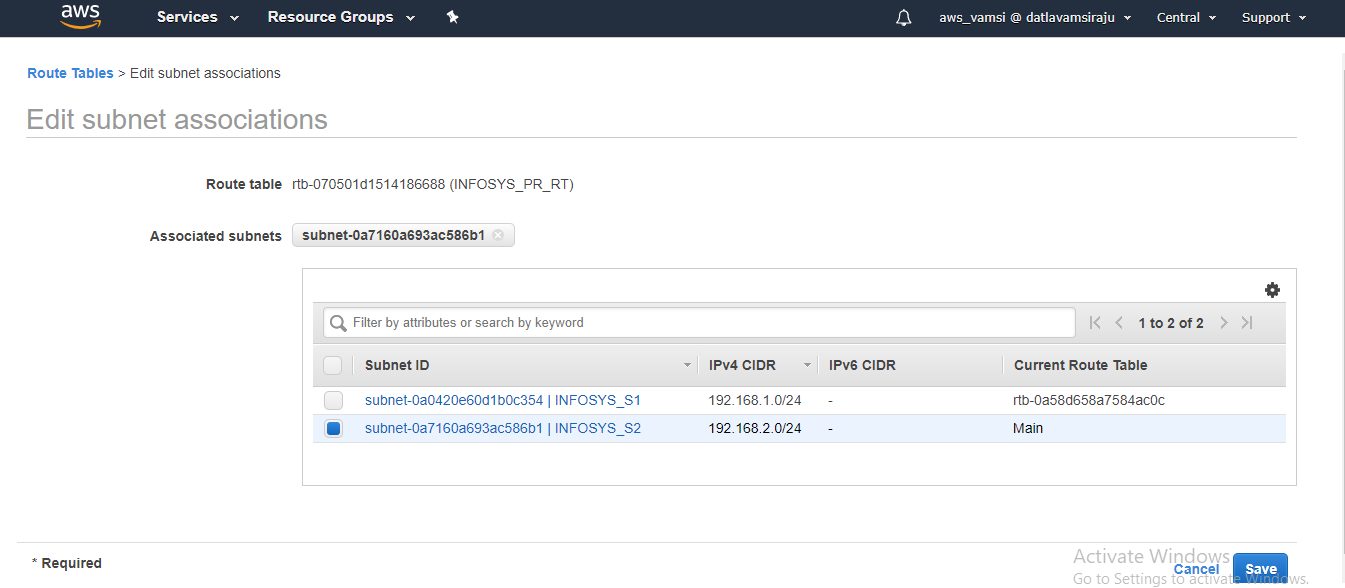
1. Subnet1 is assigned to public route table.
2. Subnet2 is assigned to private route table.

Public route table assigned to public subnet:





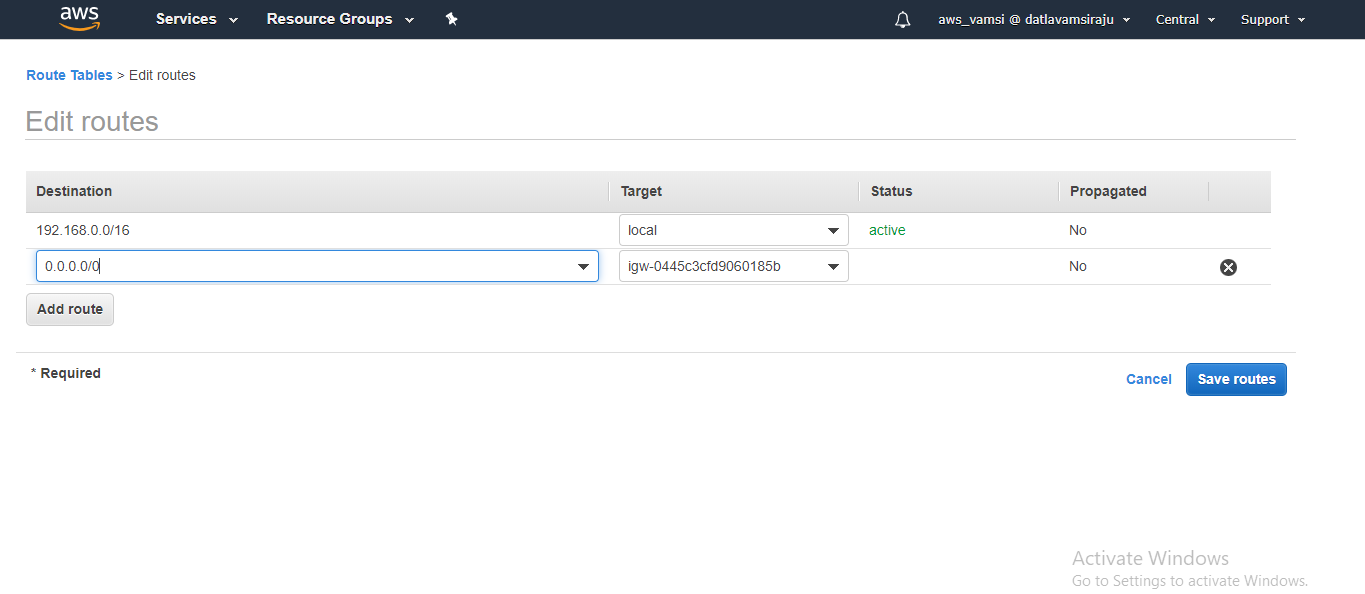
Private route table assigned to private subnet:





STEP7:

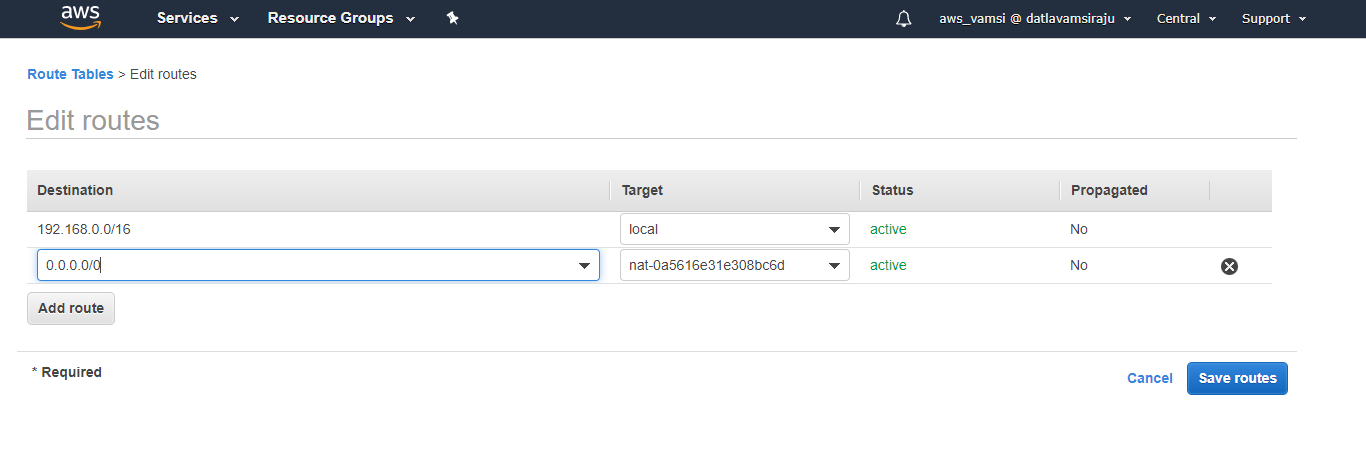
* Need to attach IGW to Route table.
* Only when you attach IGW then the Route table and attached Subnet will act as public.





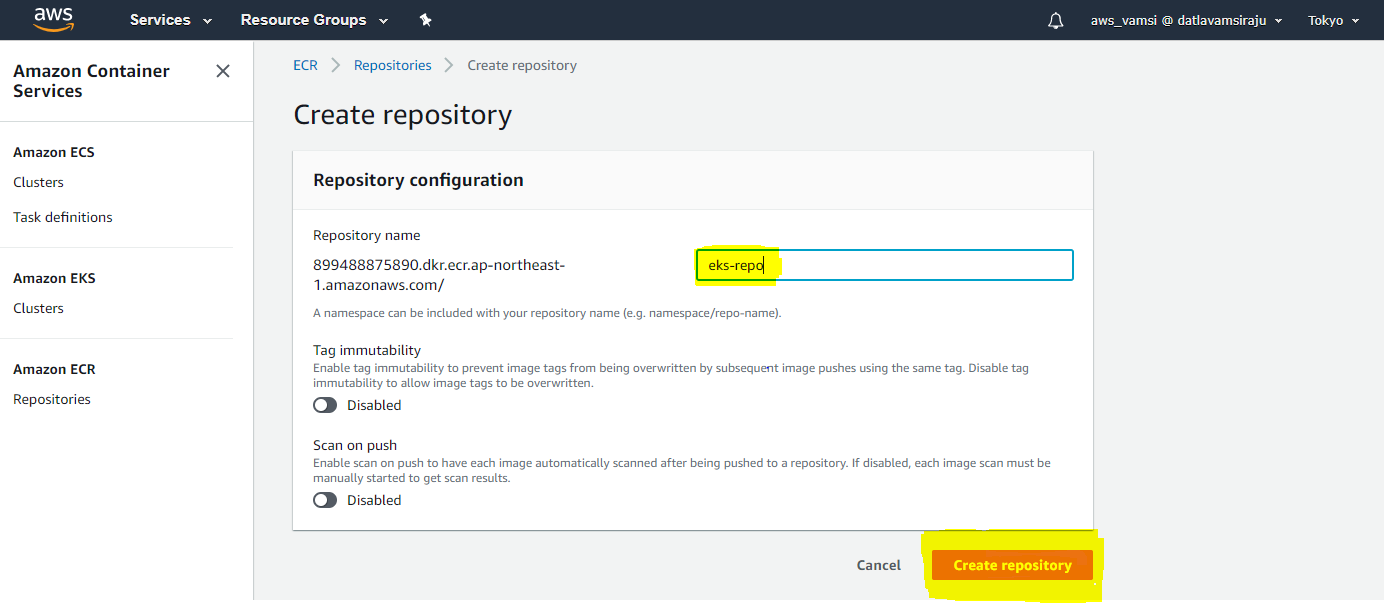
STEP8:

* NAT gate way is a bridge between Private Route table and internet.
* Once NAT is connected to the route table the private route table will be connected to internet.



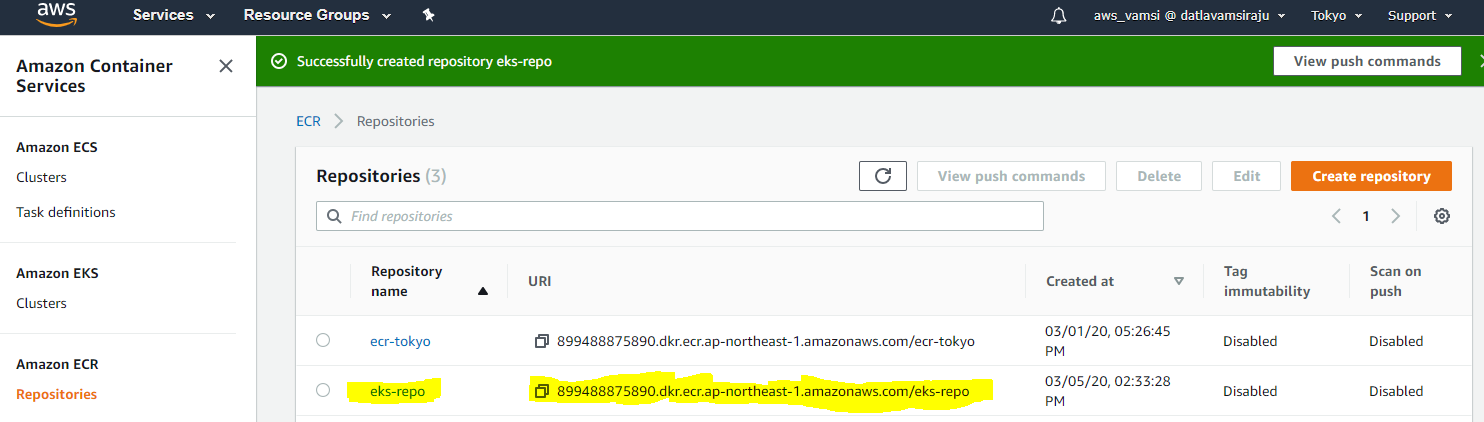


* 1. **Create an ECR repository:**





Give a name to ECR repository and create it in my case I have given EKS-repo.

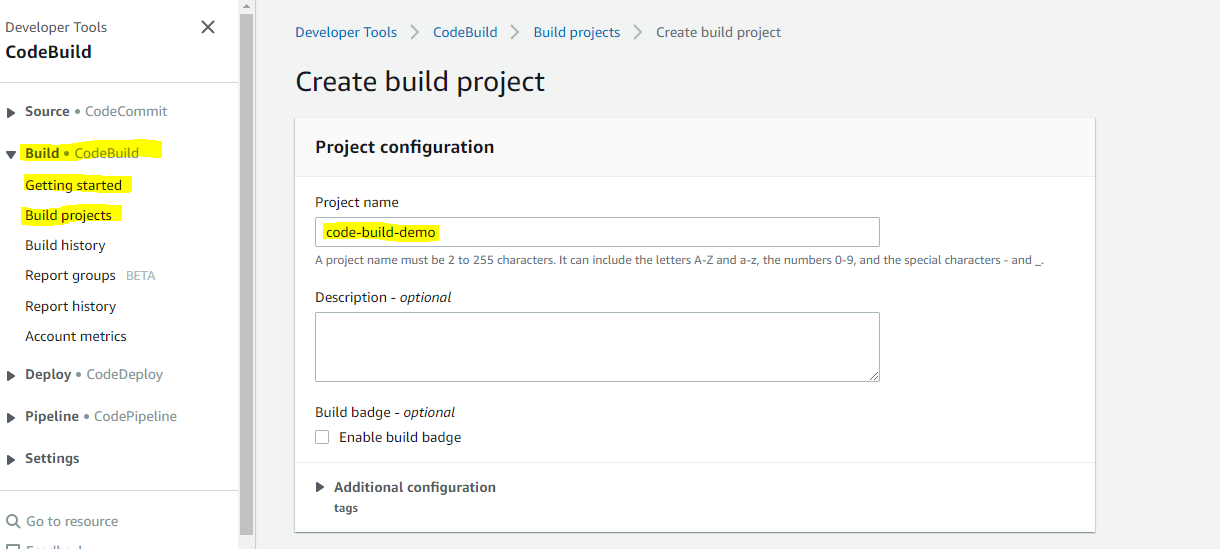


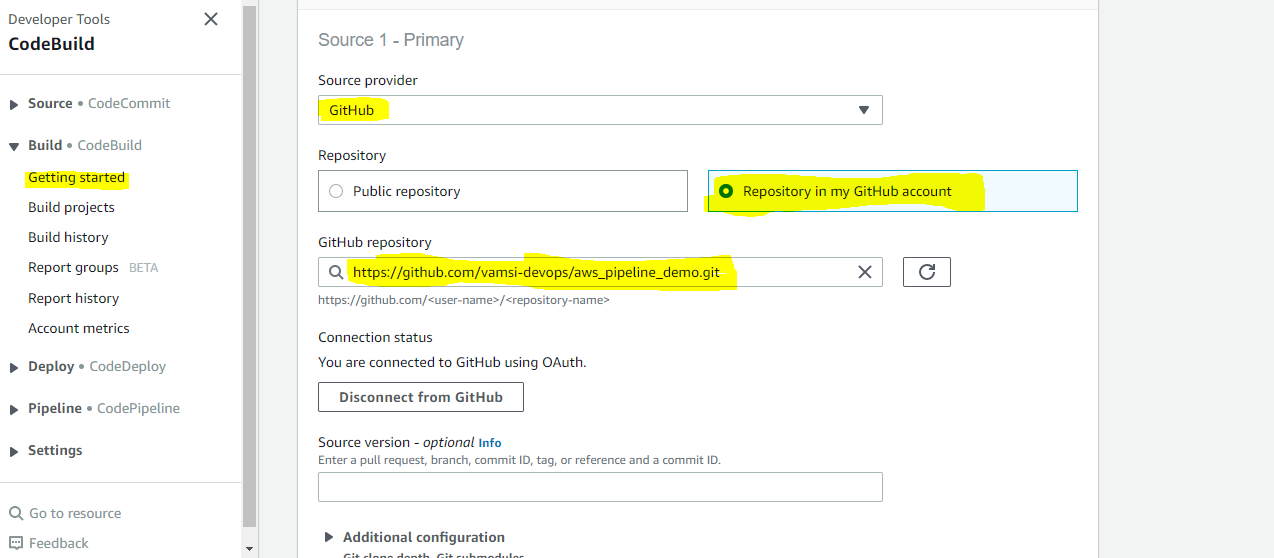


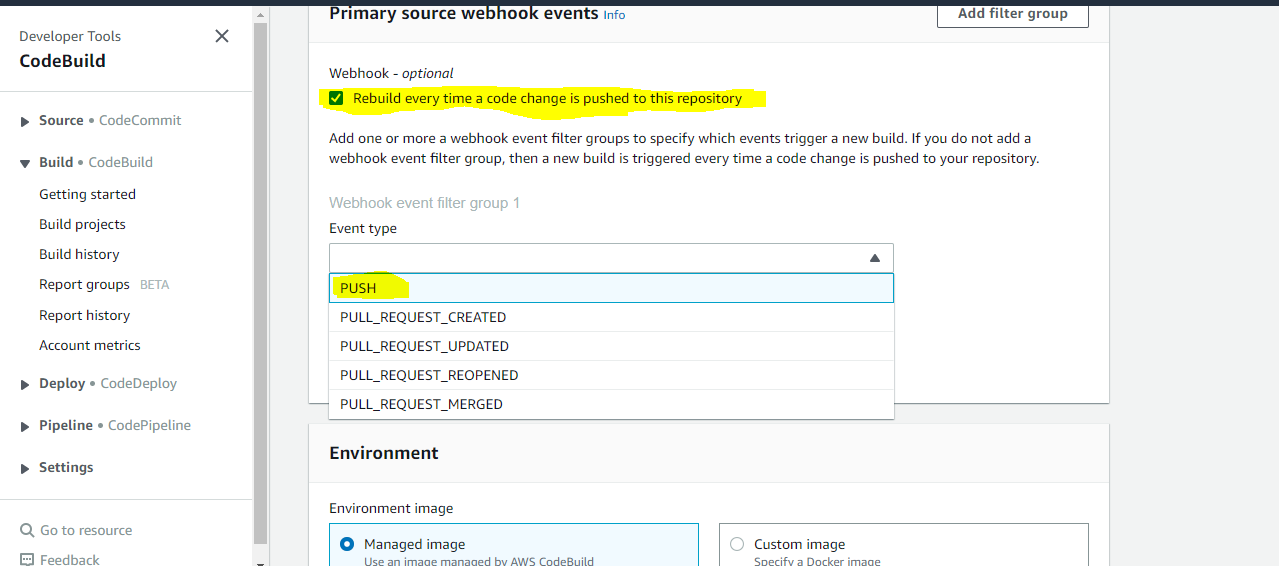
* After creating ECR Repository create a Build project in Code Build

Create build project.

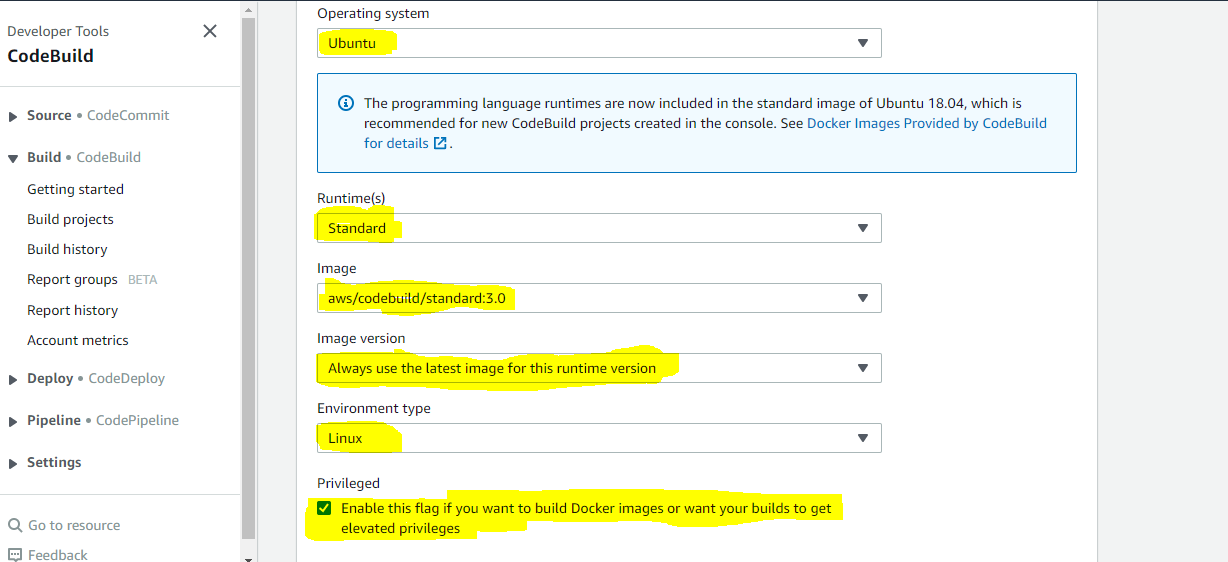
**2.1 Code Build:**

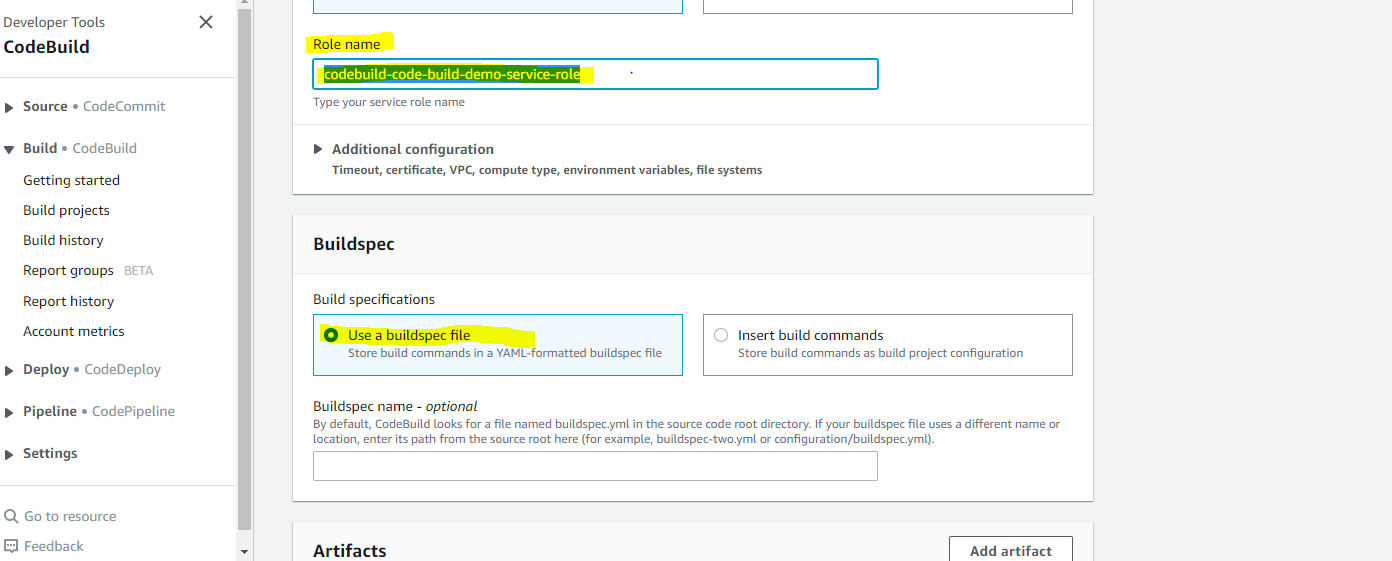


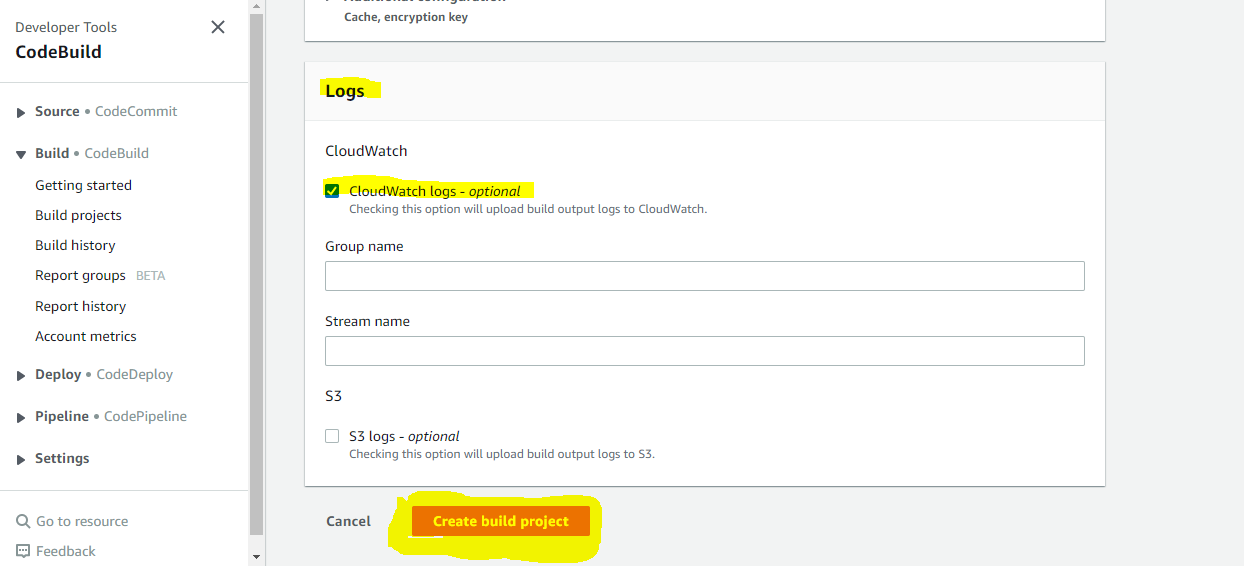




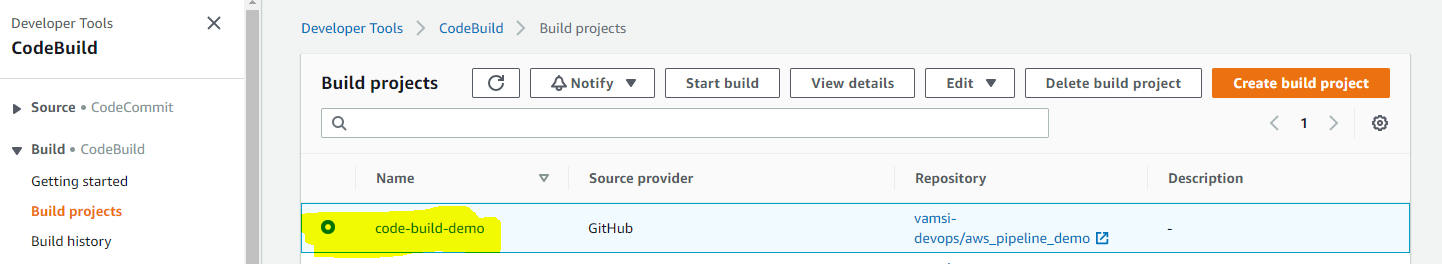
Here we will give a Debian/RedHat instance for building the image.



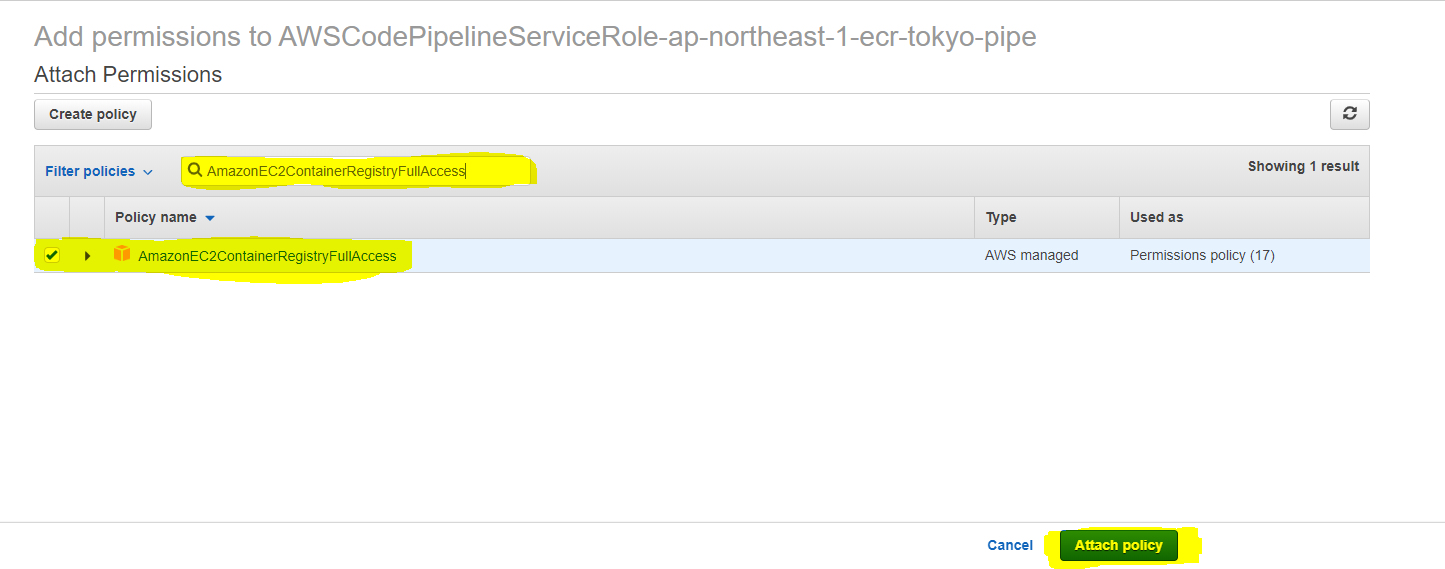




Successfully created a Code Build.

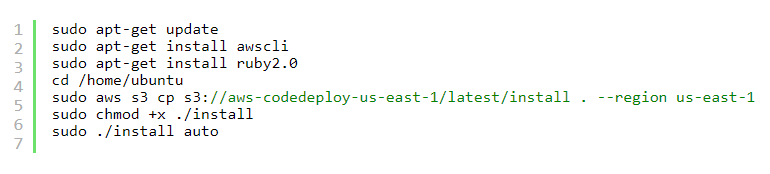


* Create a same IAM role which you given in Code build and attach a AmazonEC2ContainerRegistryFullAccess Policy to the service role for pushing the images into the ECR registry.



**2.2 Create an Ec2 Ubuntu Instance and make sure to Install AWS CLI 2.0 & Ruby:**

 Install the Code Deploy agent on the node now by following the below steps:

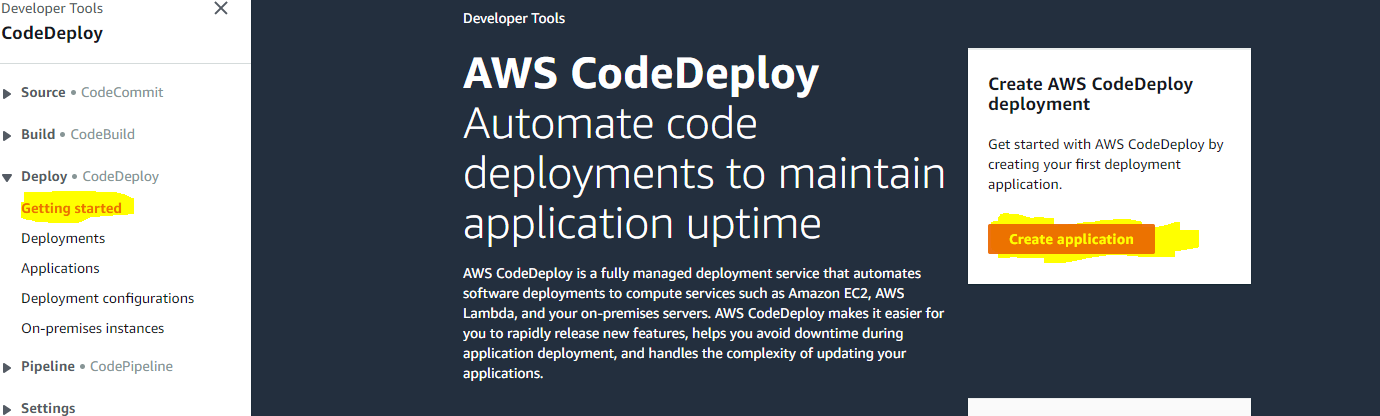


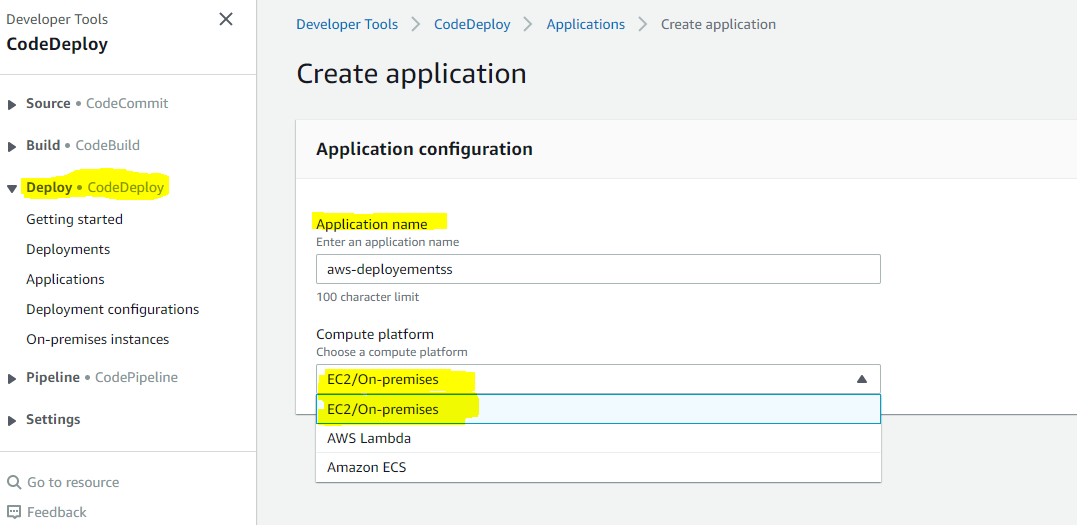
* And create IAM role and attach full Administrator access policy to the role and attach that role to the instance.

🡪 To check whether the Code Deploy agent working or not.

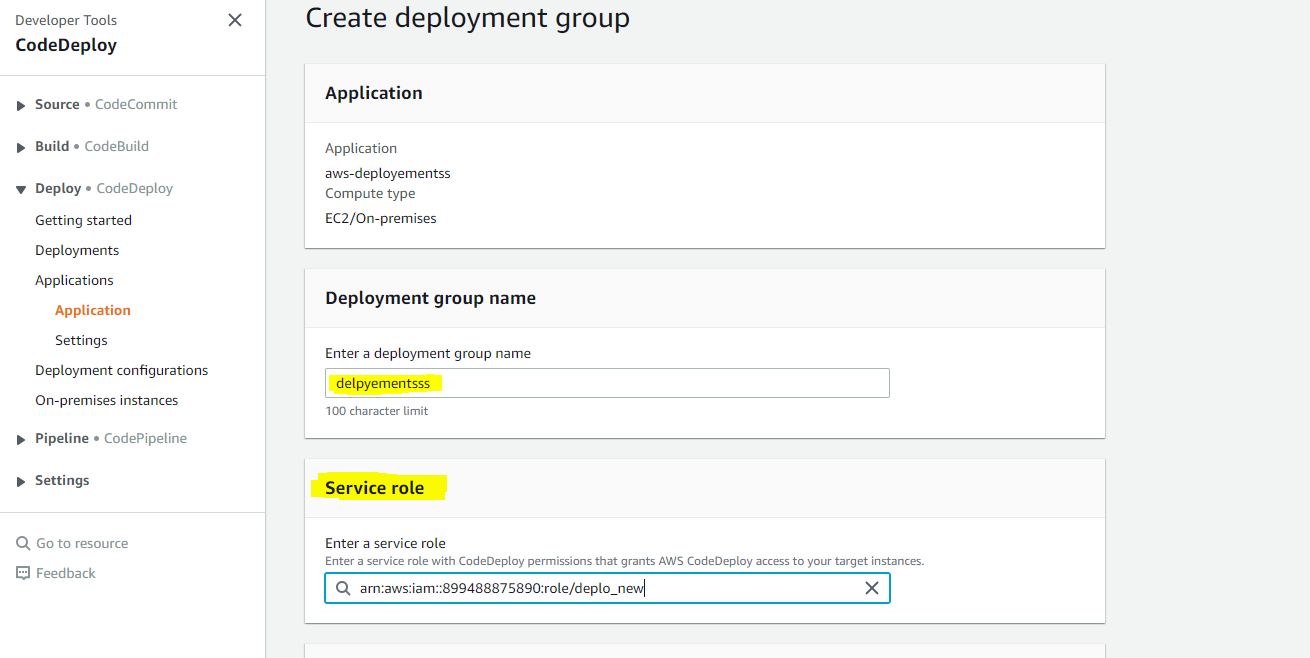
# sudo service codedeploy-agent status

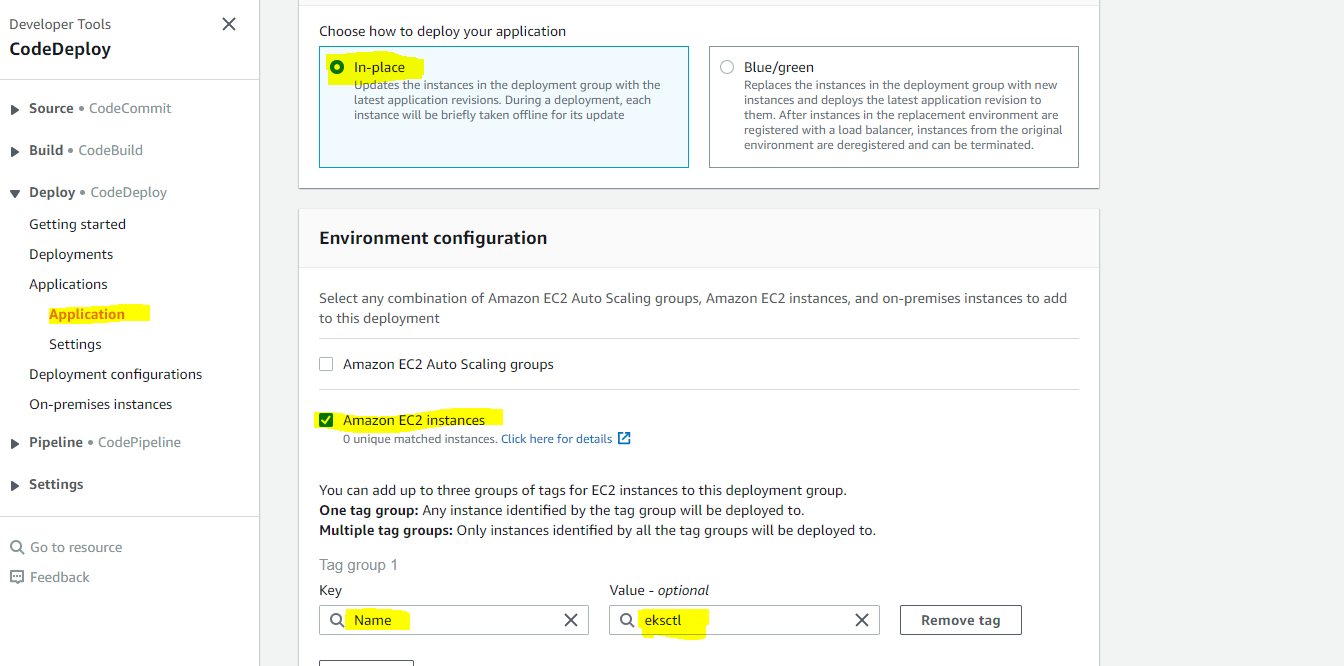
**2.3 Create a Code Deploy:**

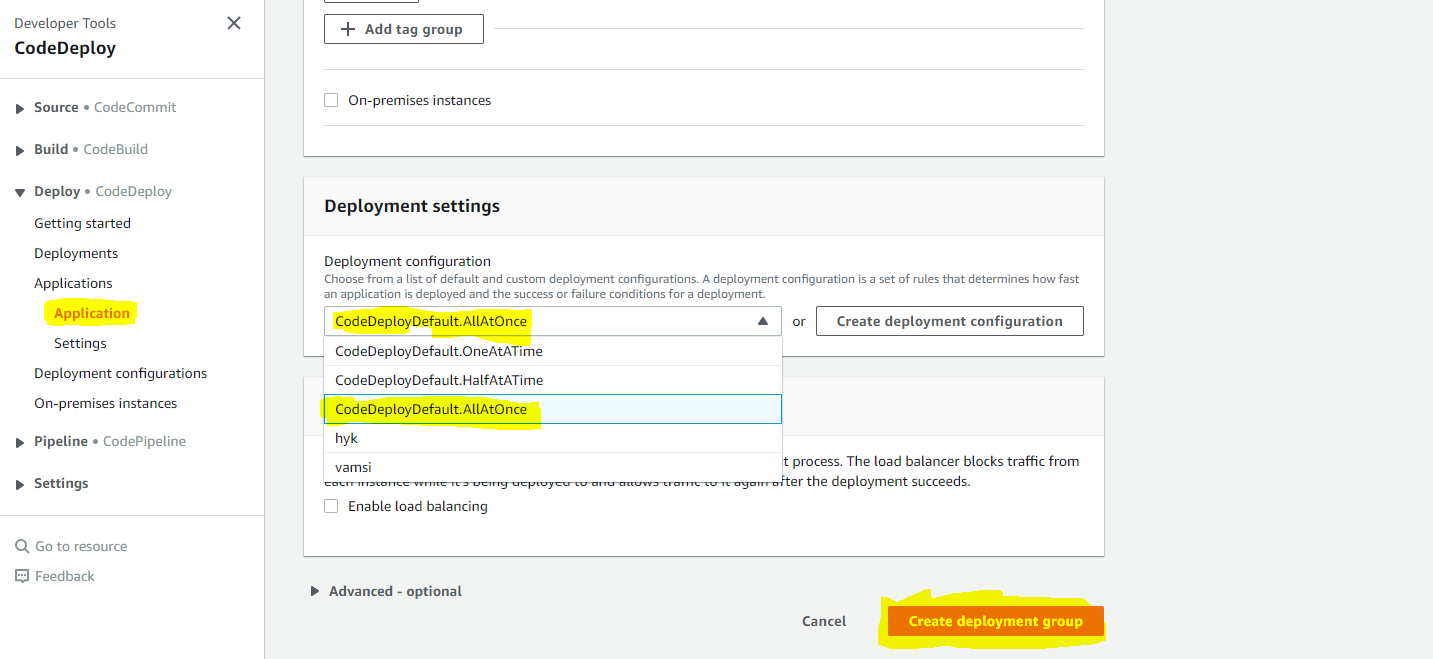


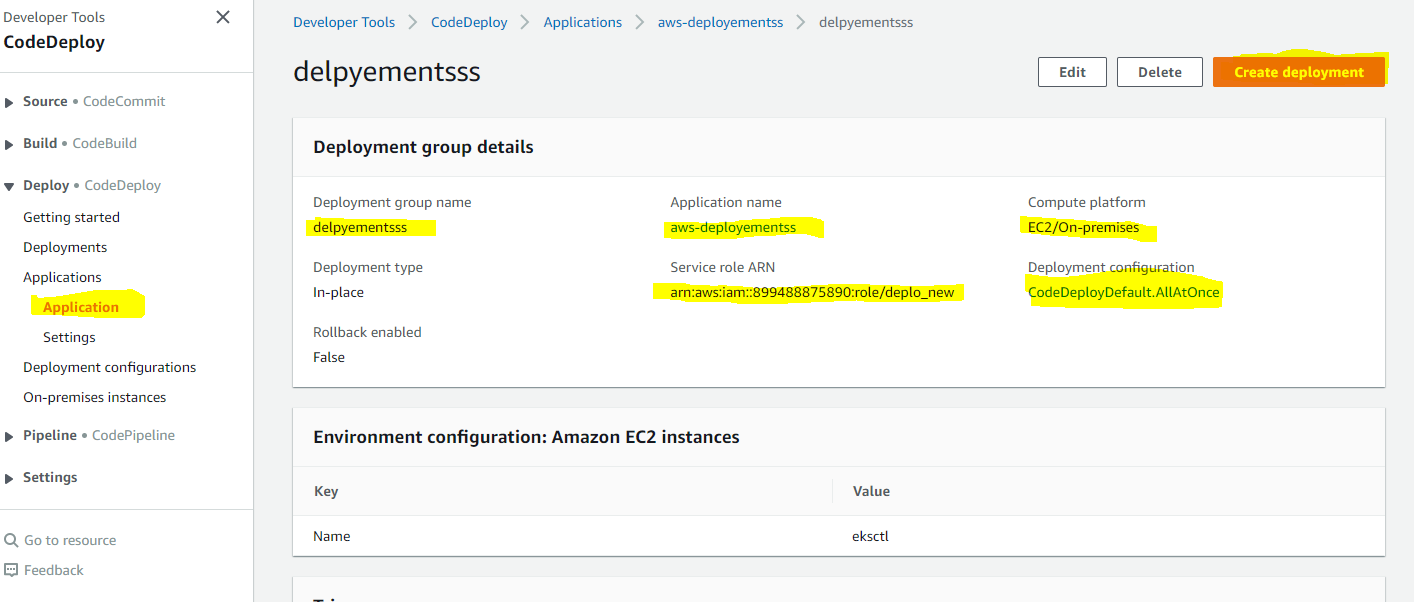


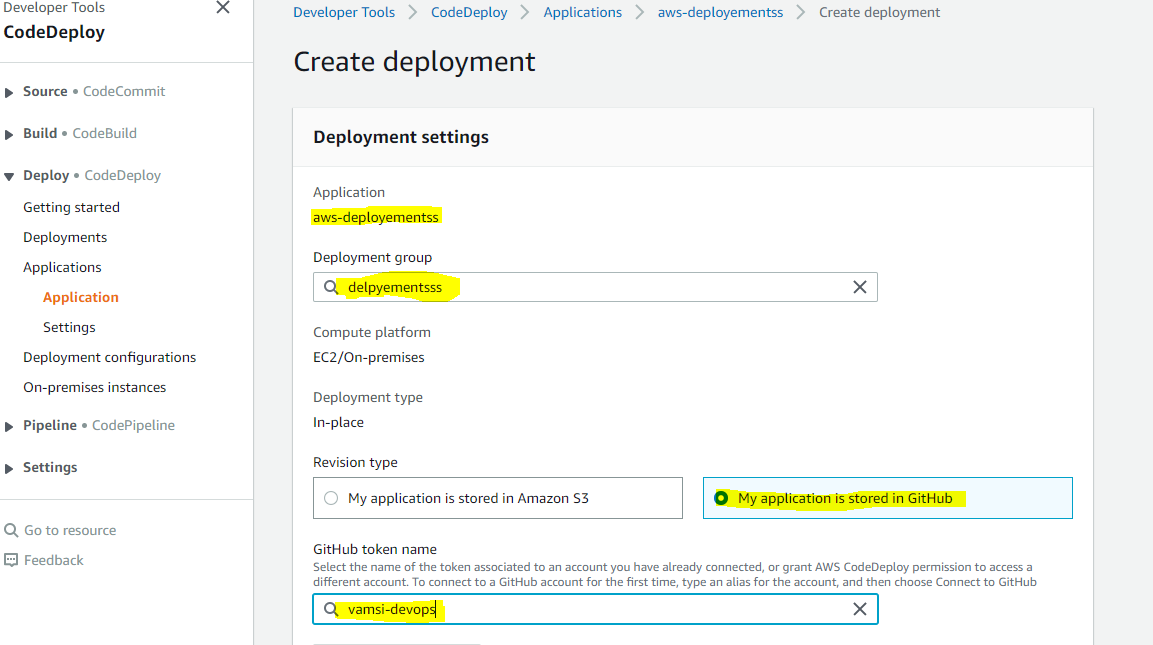
* After creating Application create the deployment group.

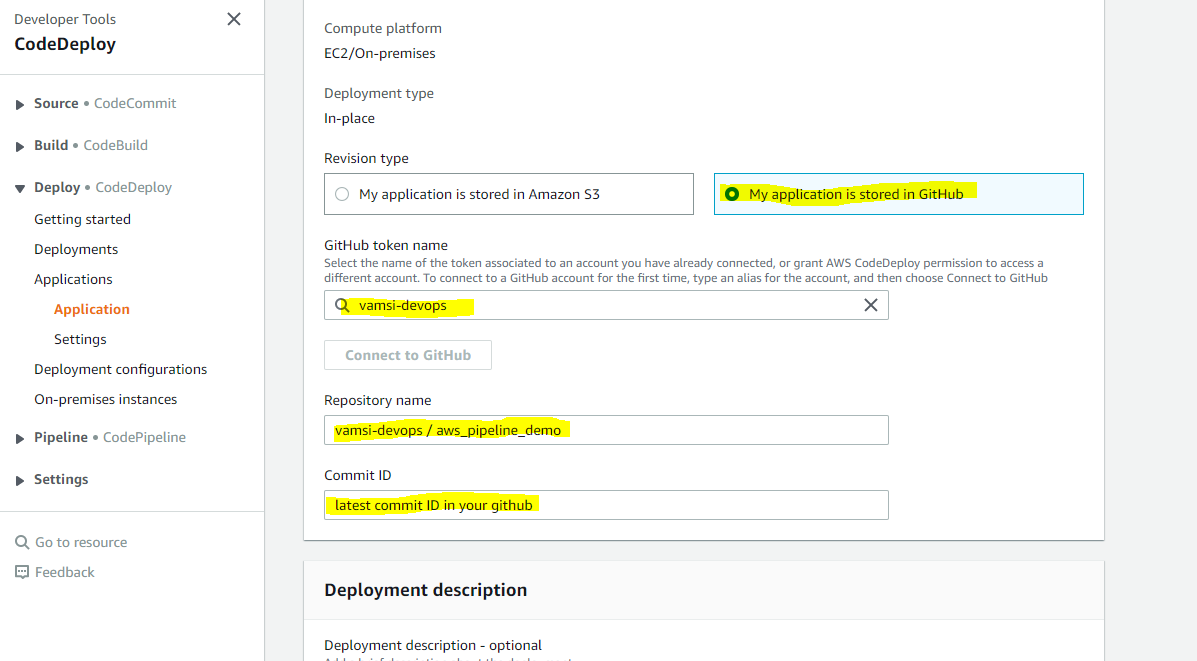


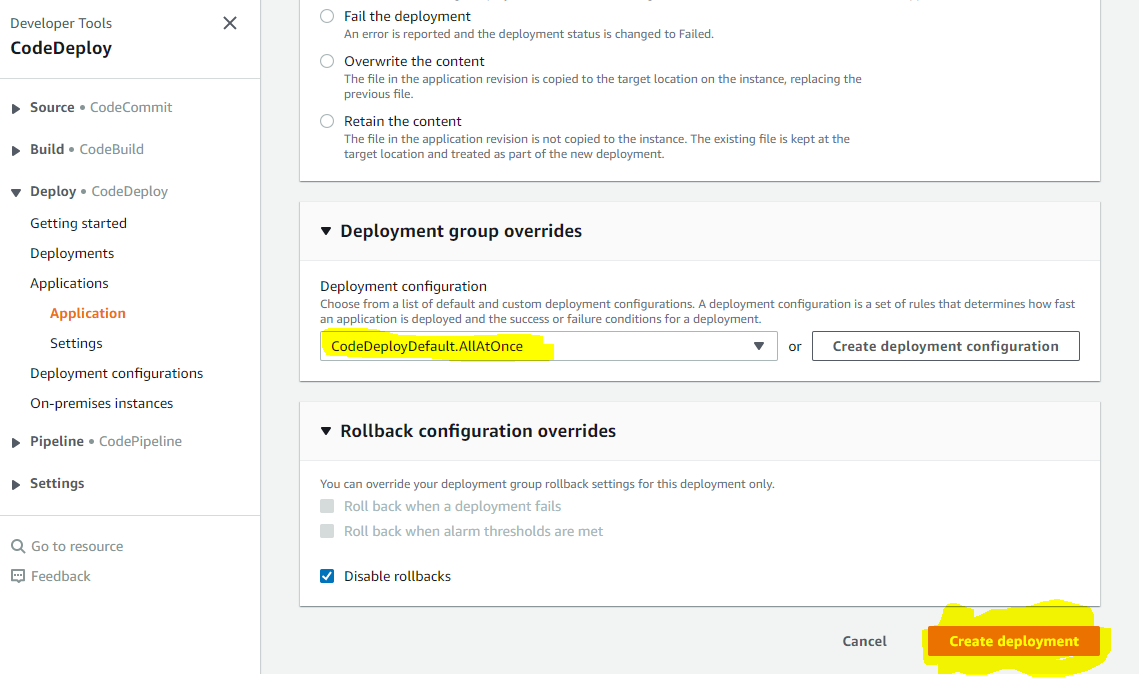












**2.4 Code Pipeline:**

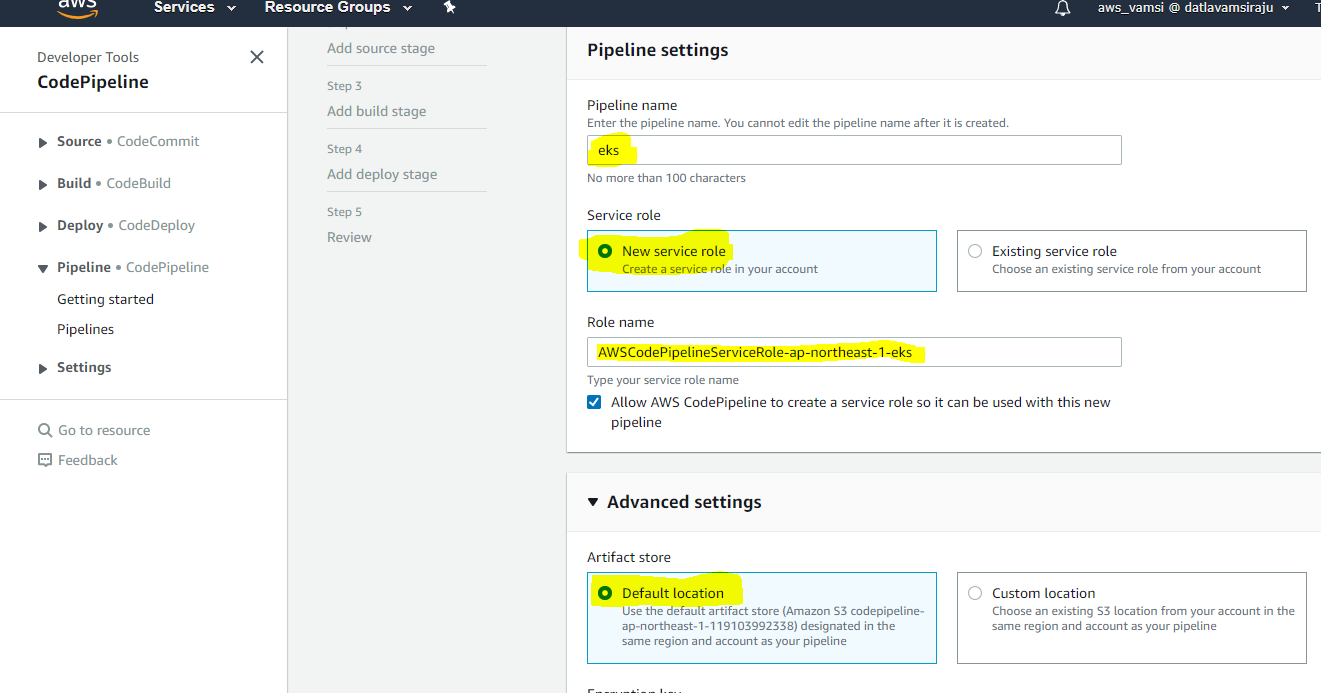
* Now Create a CodePipeline for one click deployment.
* There are three things in Code pipeline.

1) Source 🡪 GitHub is linked to Source

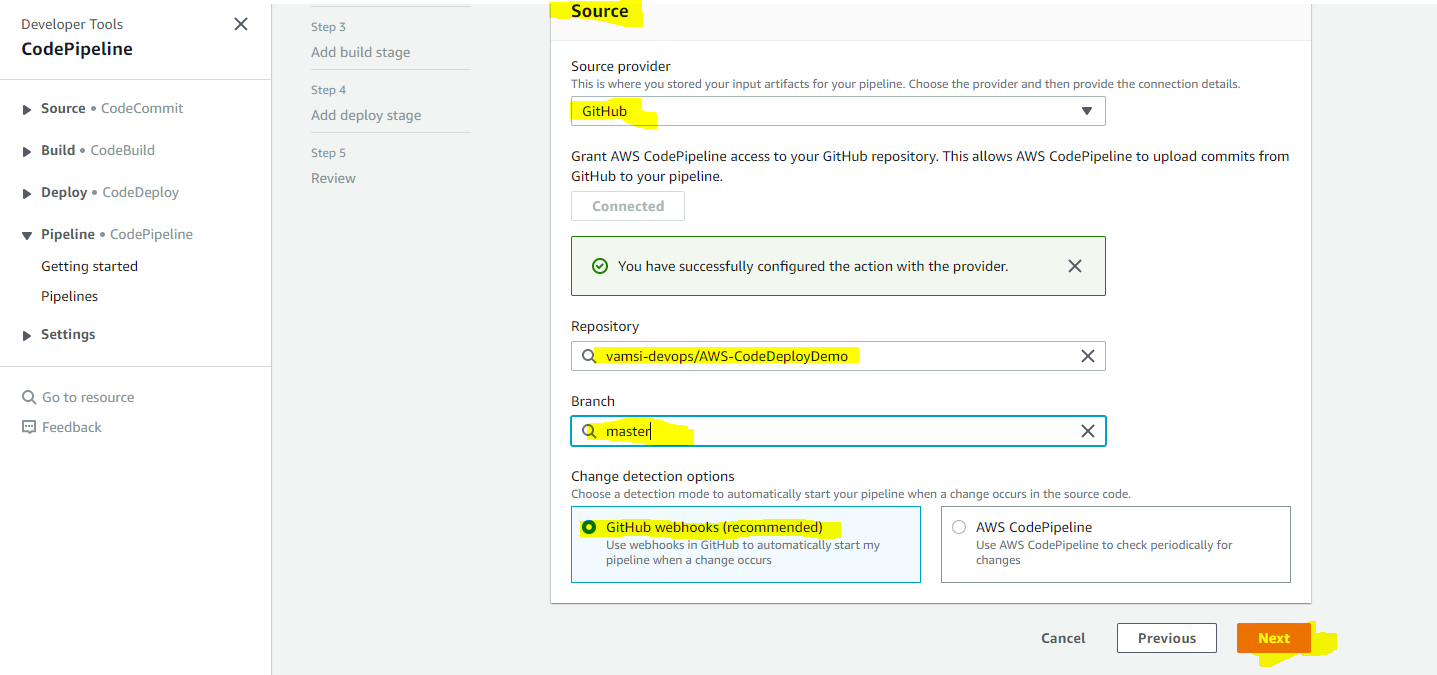
2) Code Build 🡪 buildspec.yml

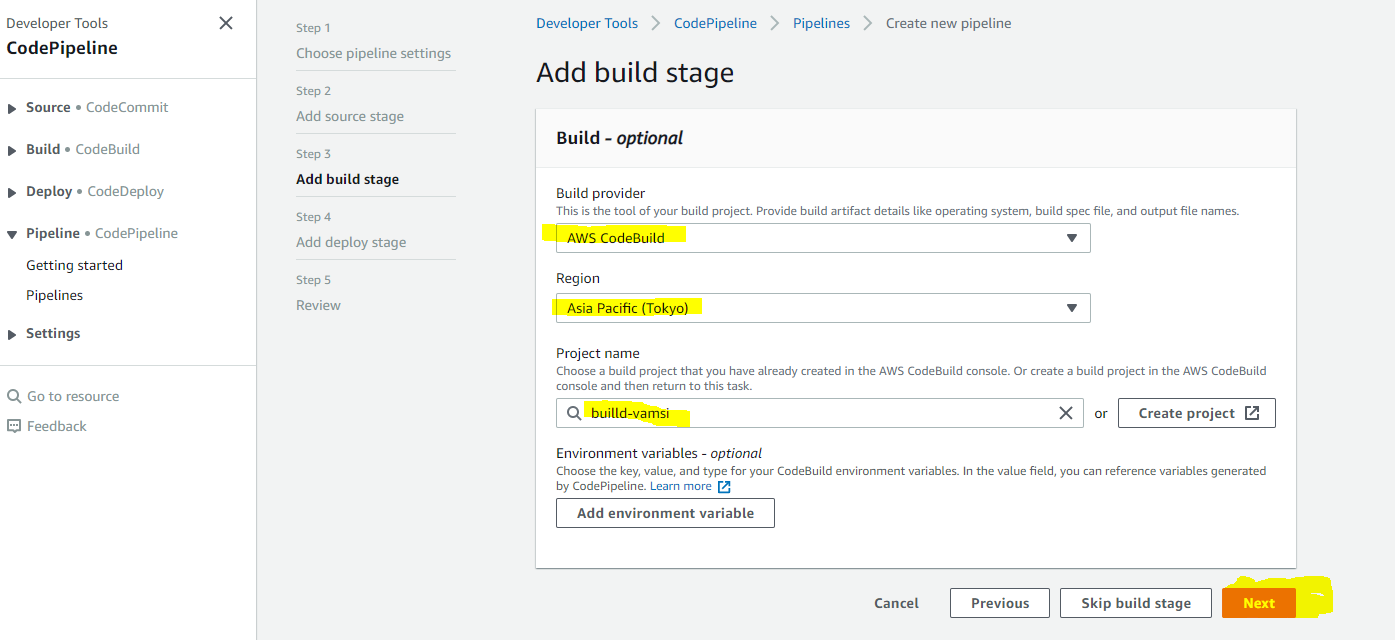
3) Code Deploy 🡪 appspec.yml

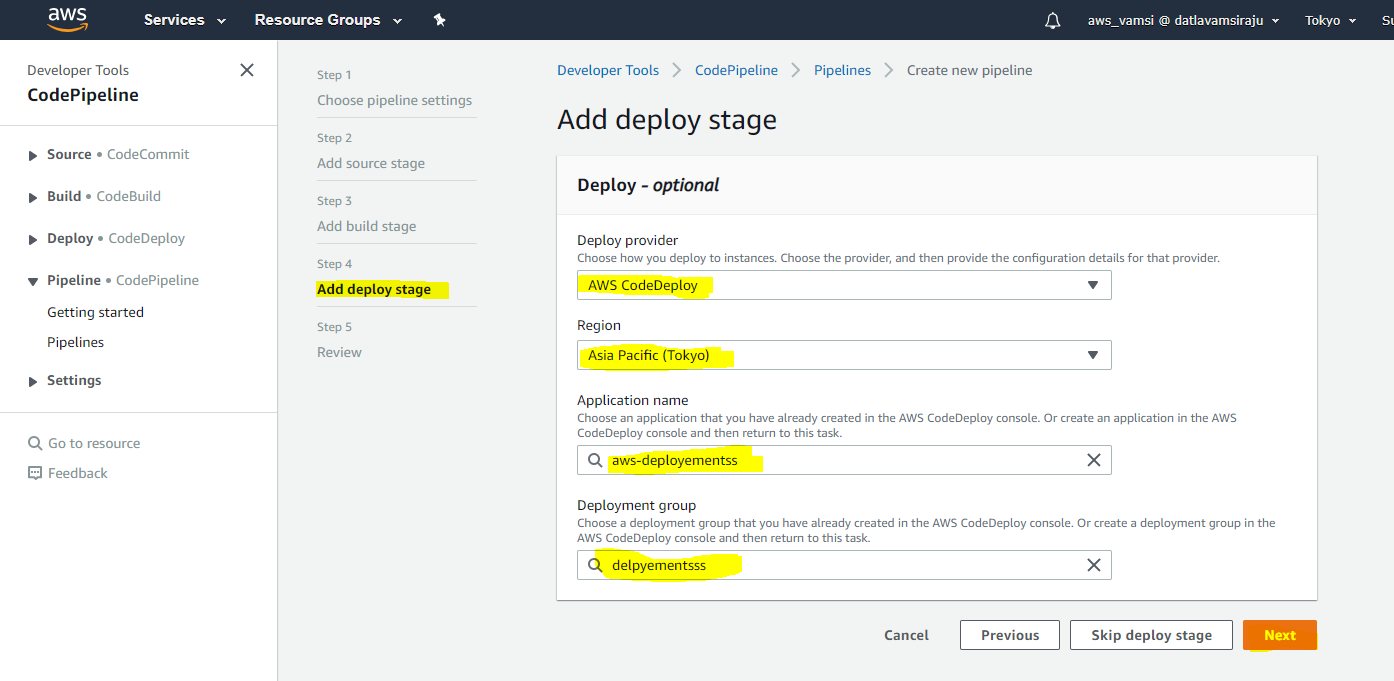




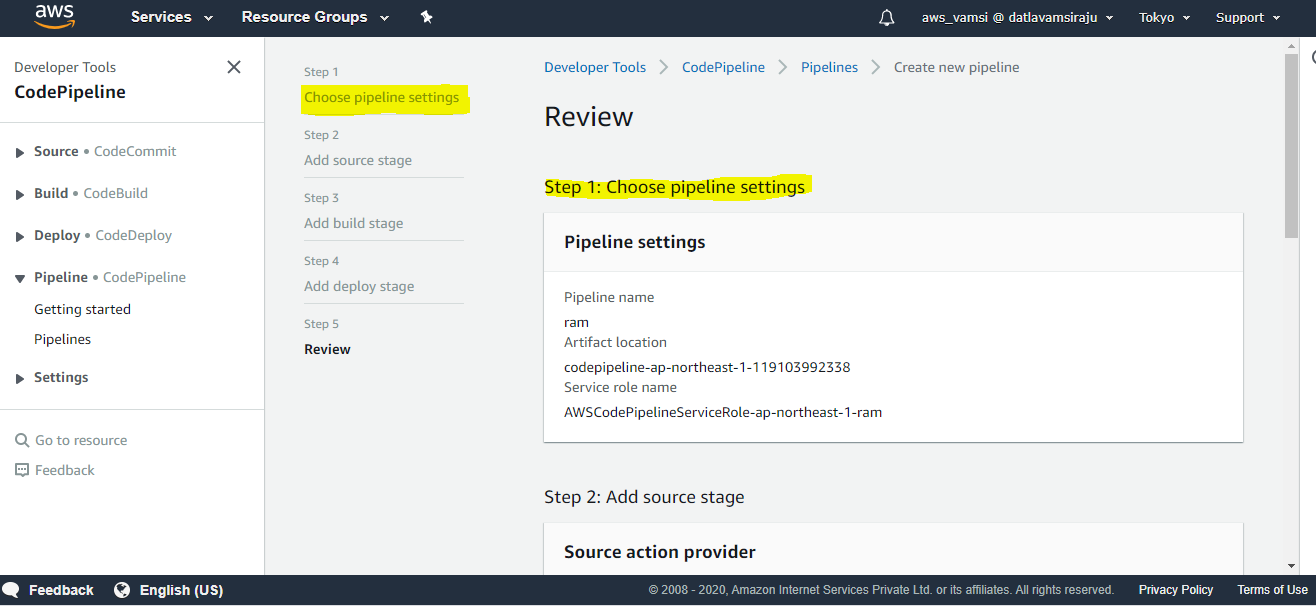




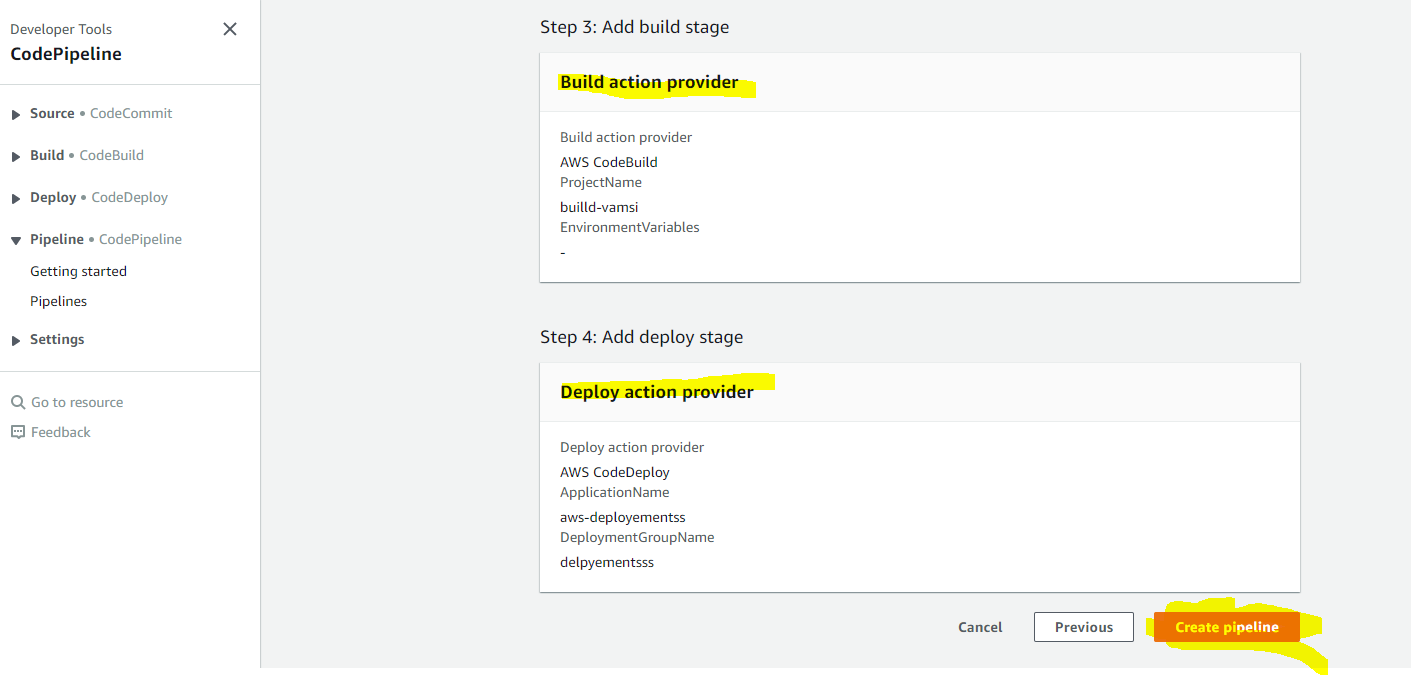


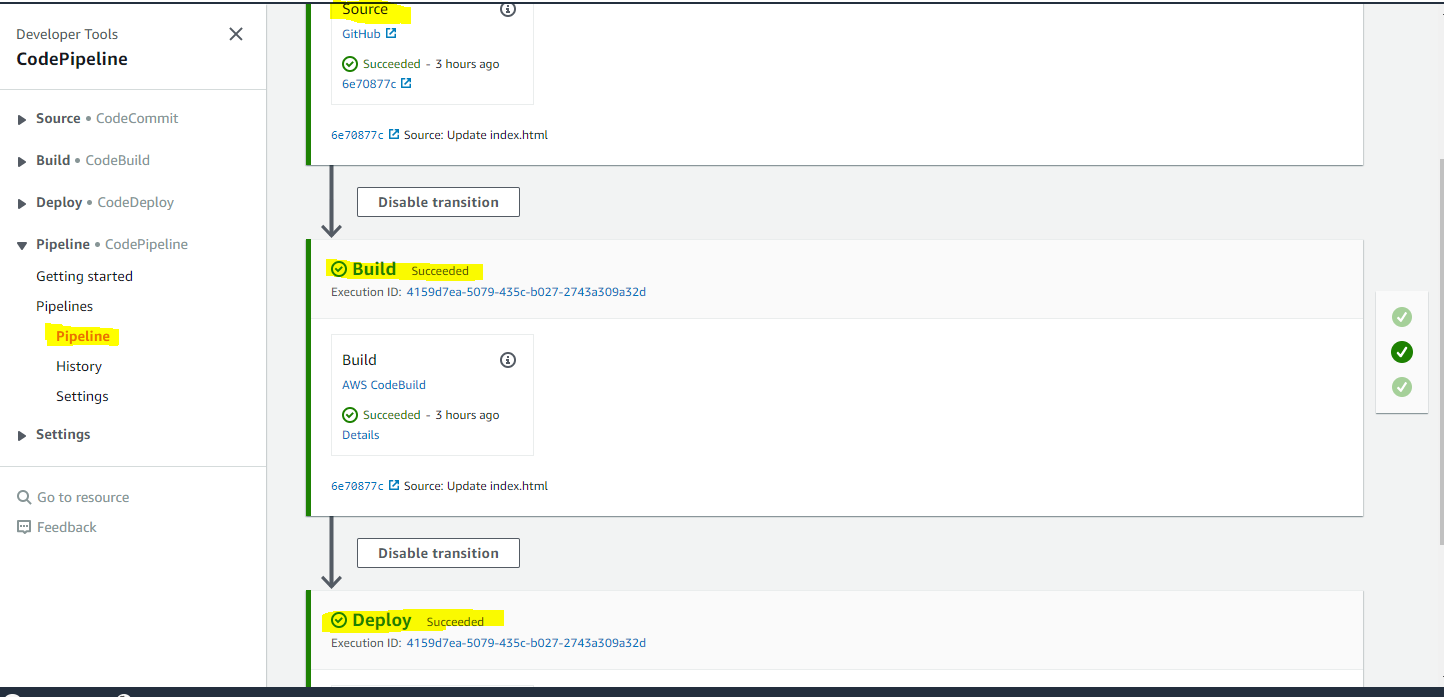










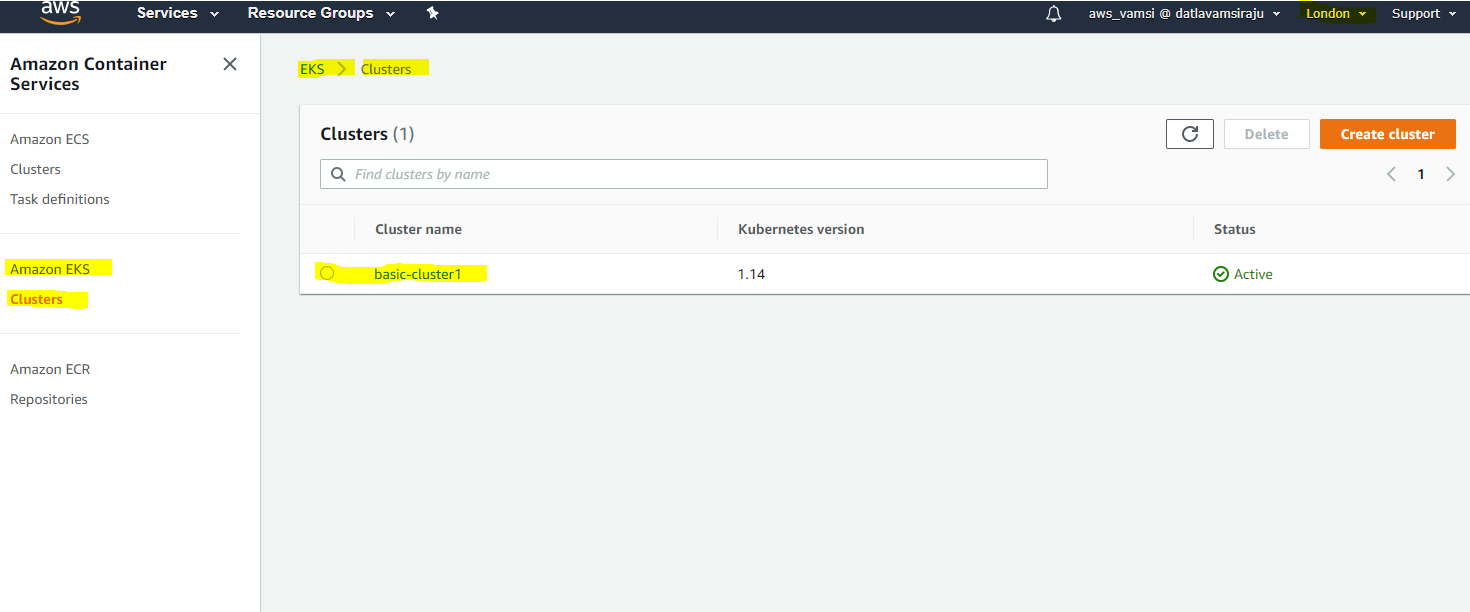


* With the help of below commands we can create a basic-cluster1 these commands are given in appspec.yml.

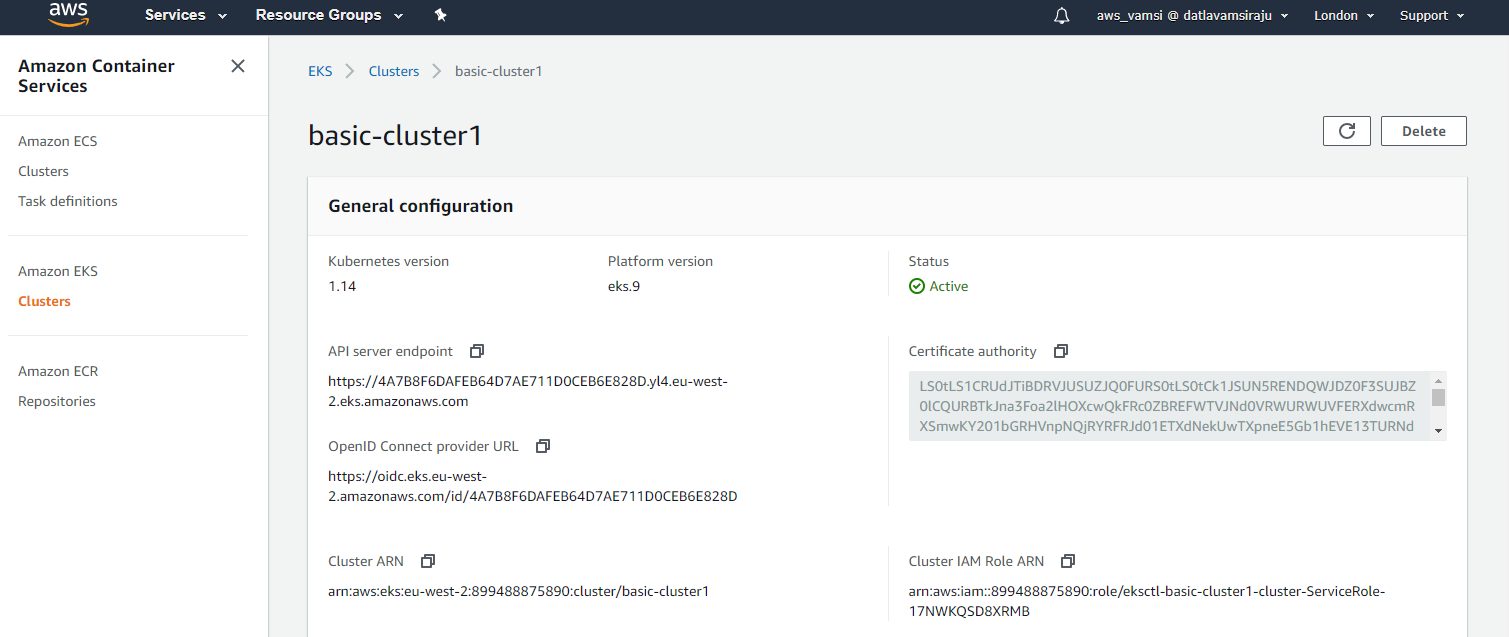
|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |

The above commands with the help of appspec.yml it will create an EKS cluster.

**2.5 EKS Cluster Formation:**

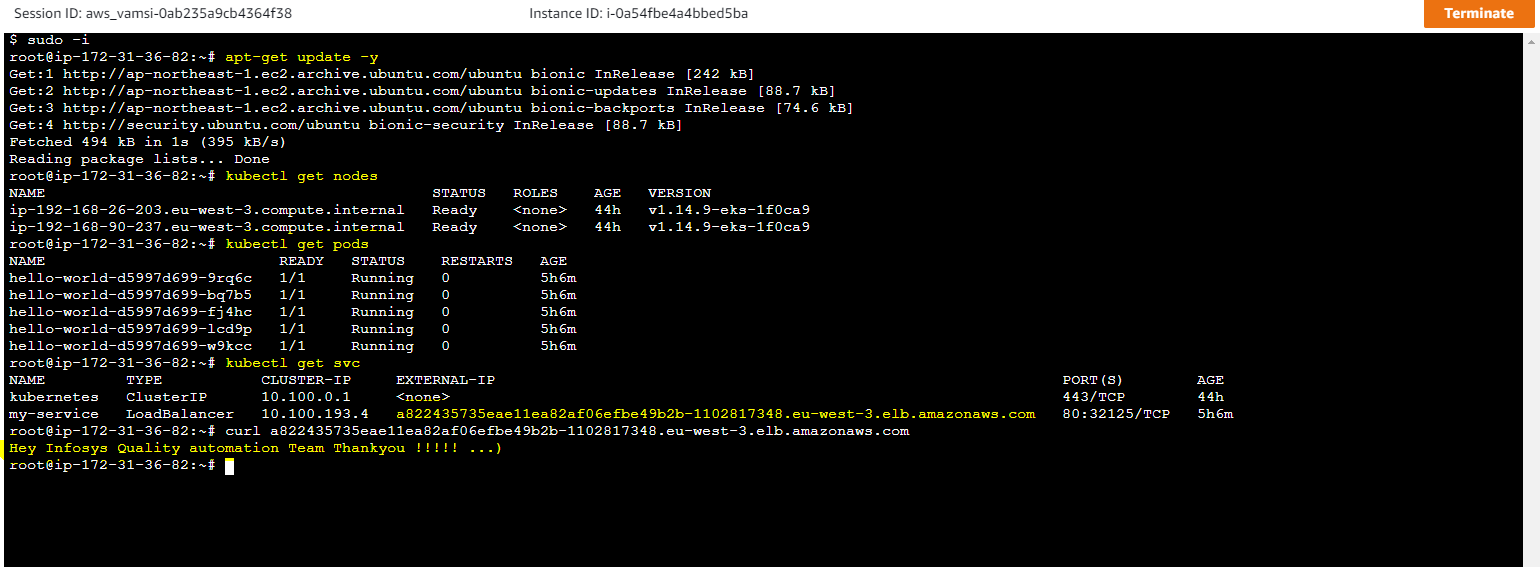








**2.6 Output:**





**Reference:**

[https://www.tothenew.com/blog/aws-codedeploy-and-github/#](https://www.tothenew.com/blog/aws-codedeploy-and-github/)

[https://docs.aws.amazon.com/cli/latest/userguide//install-cliv2-linux.html](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html)

<https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-verify.html>

<https://docs.aws.amazon.com/eks/latest/userguide/getting-started-eksctl.html>