**09. Create CodePipeline**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/11-DevOps-with-AWS-Developer-Tools>

**CodePipeline Introduction**

--- Get a high-level overview about CodePipeline Service

**Create CodePipeline**

--- Create CodePipeline

--- Go to Services -> CodePipeline -> Create Pipeline

--- Pipeline Settings

* Pipeline Name: eks-devops-pipe
* Service Role: New Service Role (leave to defaults)
* Role Name: Auto-populated
* Rest all leave to defaults and click Next

--- Source

* Source Provider: AWS CodeCommit
* Repository Name: eks-devops-nginx
* Branch Name: master
* Change Detection Options: CloudWatch Events (leave to defaults)

--- Build

* Build Provider: AWS CodeBuild
* Region: US East (N.Virginia)
* Project Name: Click on Create Project

--- Create Build Project

1. Project Configuration

* Project Name: eks-devops-cb-for-pipe
* Description: CodeBuild Project for EKS DevOps Pipeline

1. Environment

* Environment Image: Managed Image
* Operating System: Amazon Linux 2
* Runtimes: Standard
* Image: aws/codebuild/amazonlinux2-x86\_64-standard:3.0
* Image Version: Always use the latest version for this runtime
* Environment Type: Linux
* Privileged: Enable
* Role Name: Auto-populated
* Additional Configurations
  + All leave to defaults except Environment Variables
  + Add Environment Variables
  + REPOSITORY\_URI = 180789647333.dkr.ecr.us-east-1.amazonaws.com/eks-devops-nginx
  + EKS\_KUBECTL\_ROLE\_ARN = arn:aws:iam::180789647333:role/EksCodeBuildKubectlRole
  + EKS\_CLUSTER\_NAME = eksdemo1

1. Buildspec

* leave to defaults

1. Logs

* Group Name: eks-deveops-cb-pipe
* Stream Name:

--- Click on Continue to CodePipeline

--- We should see a message Successfully created eks-devops-cb-for-pipe in CodeBuild.

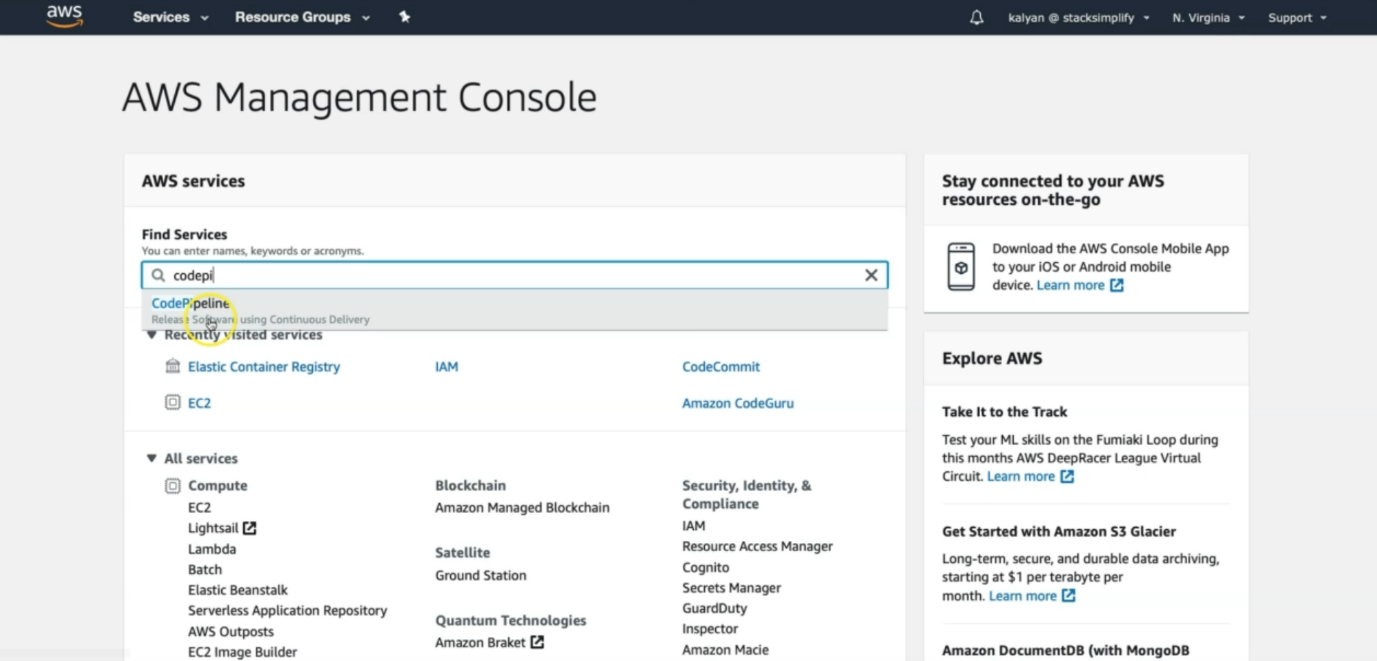
--- Click Next

--- Deploy

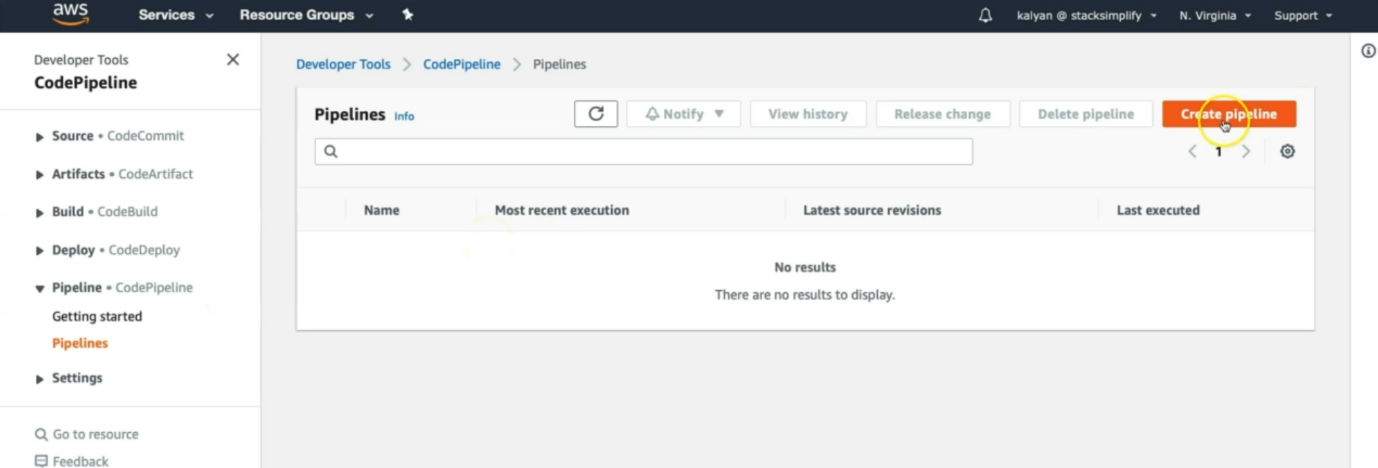
* Click on Skip Deploy Stage

--- Review

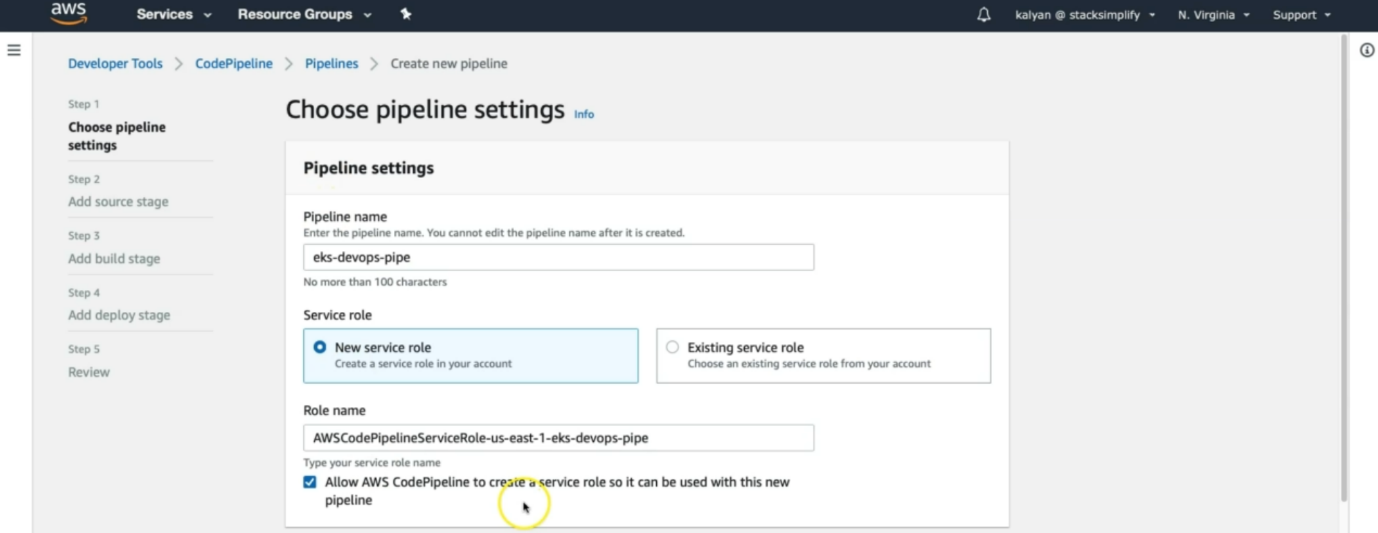
* Review and click on Create Pipeline

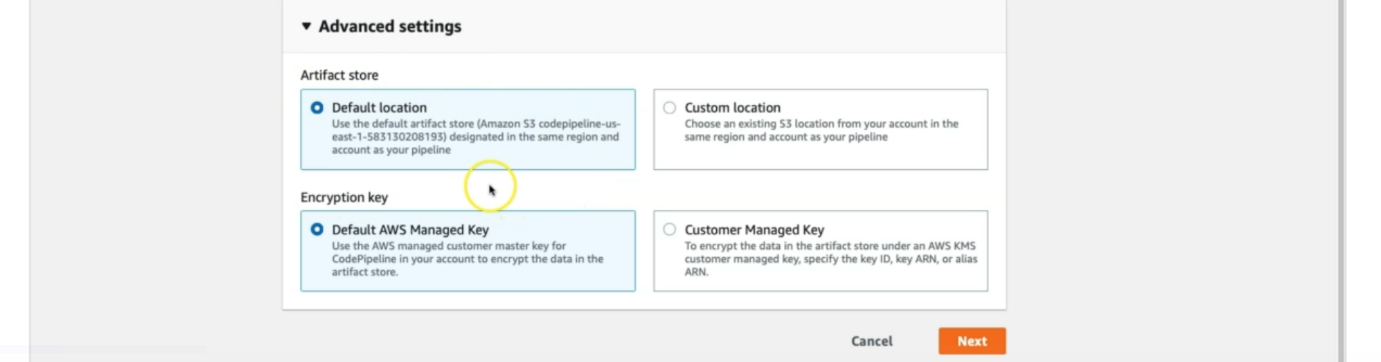


--- click on codepipeline.



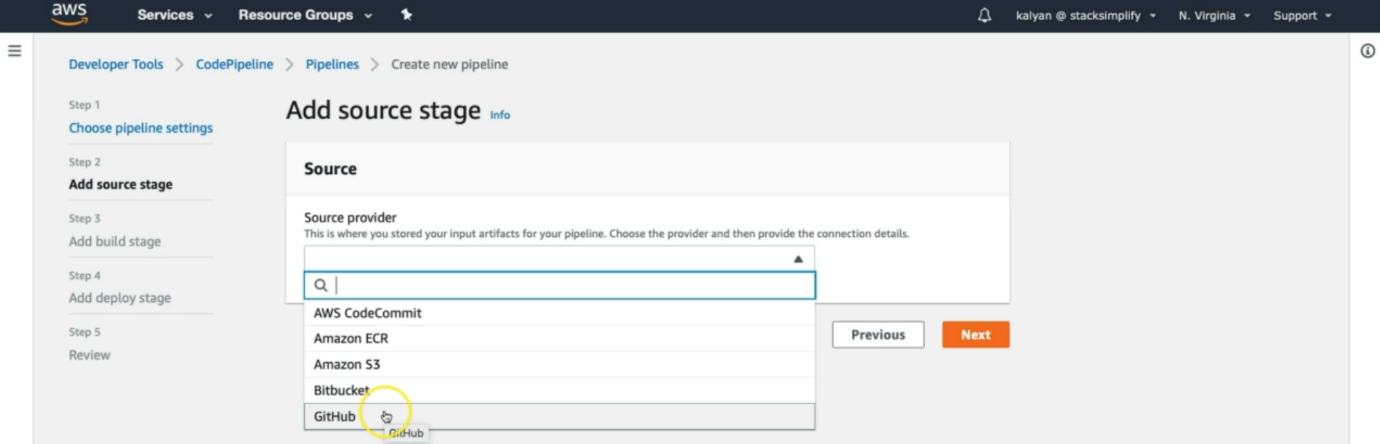
--- click on create pipeline.



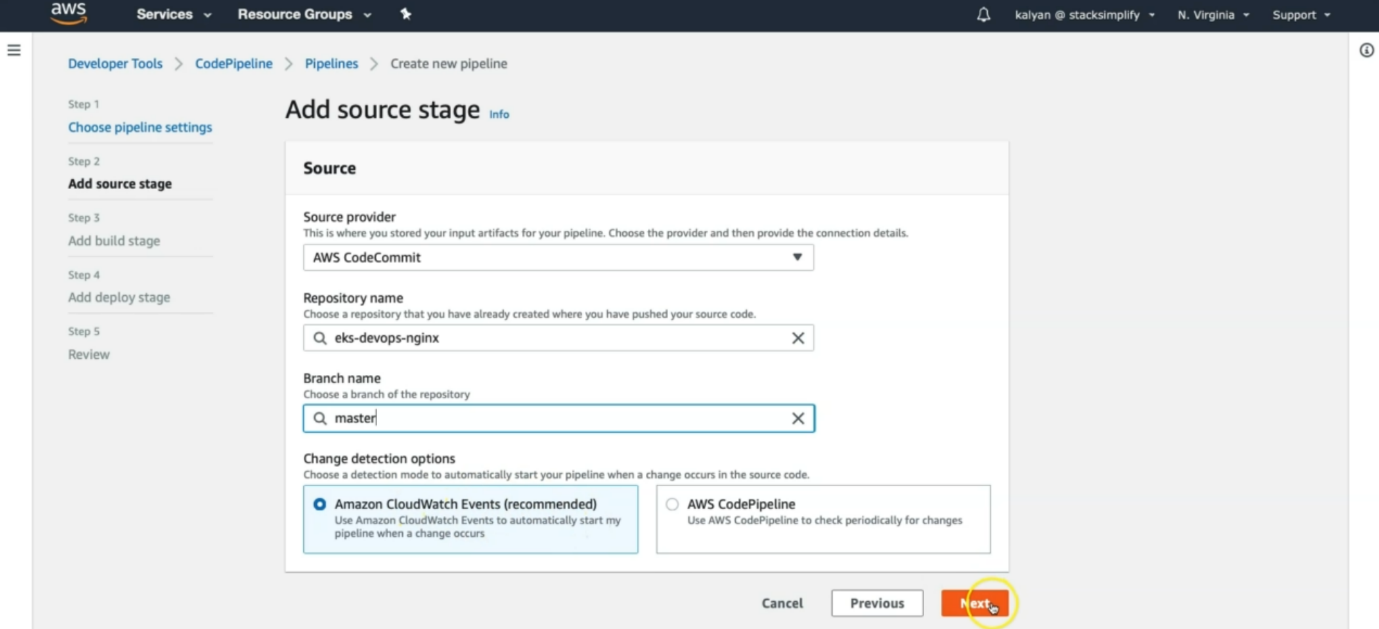


--- except for pipeline name, remaining all leave it as default.

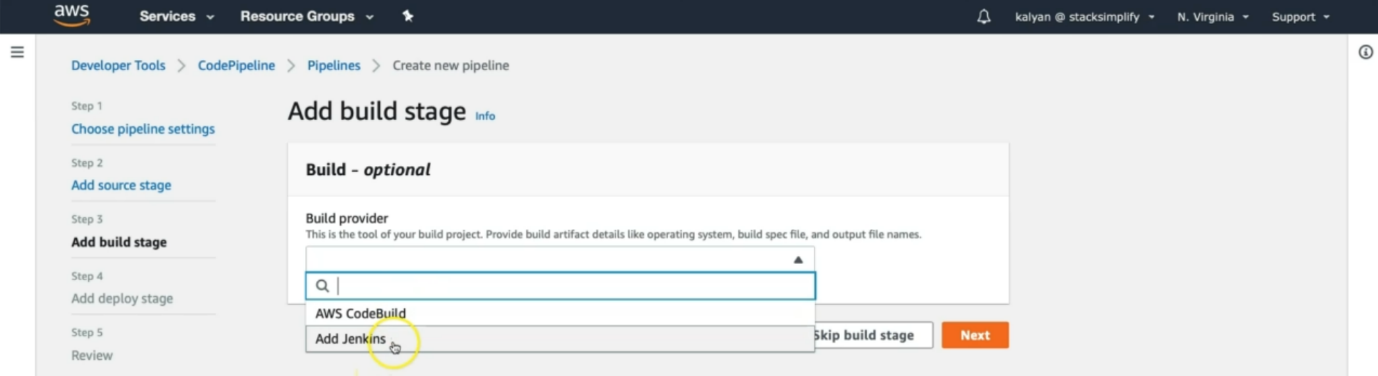
--- click on next.



--- **note** – we can use any of those options. In our case, our source code in aws codecommit. So, I will select that option.



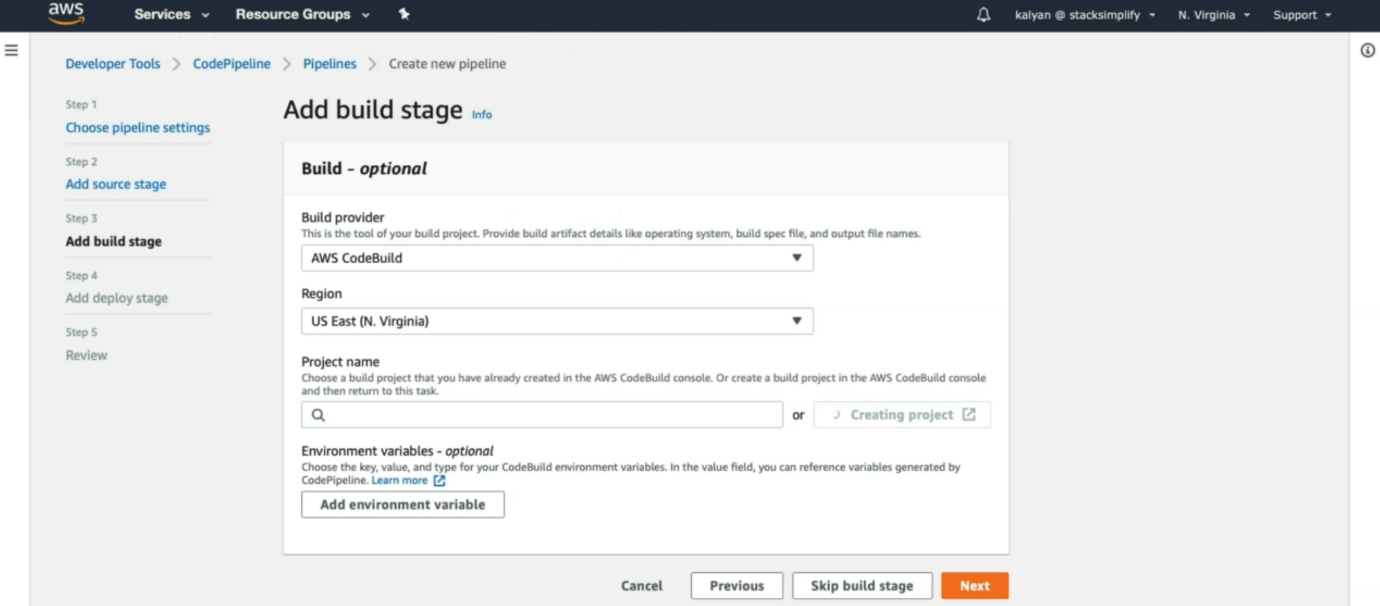
--- I selected our repository name, branch name and for change detection option, I will selected amazon cloudwatch events because it found any change in the source code, it will trigger the pipeline.



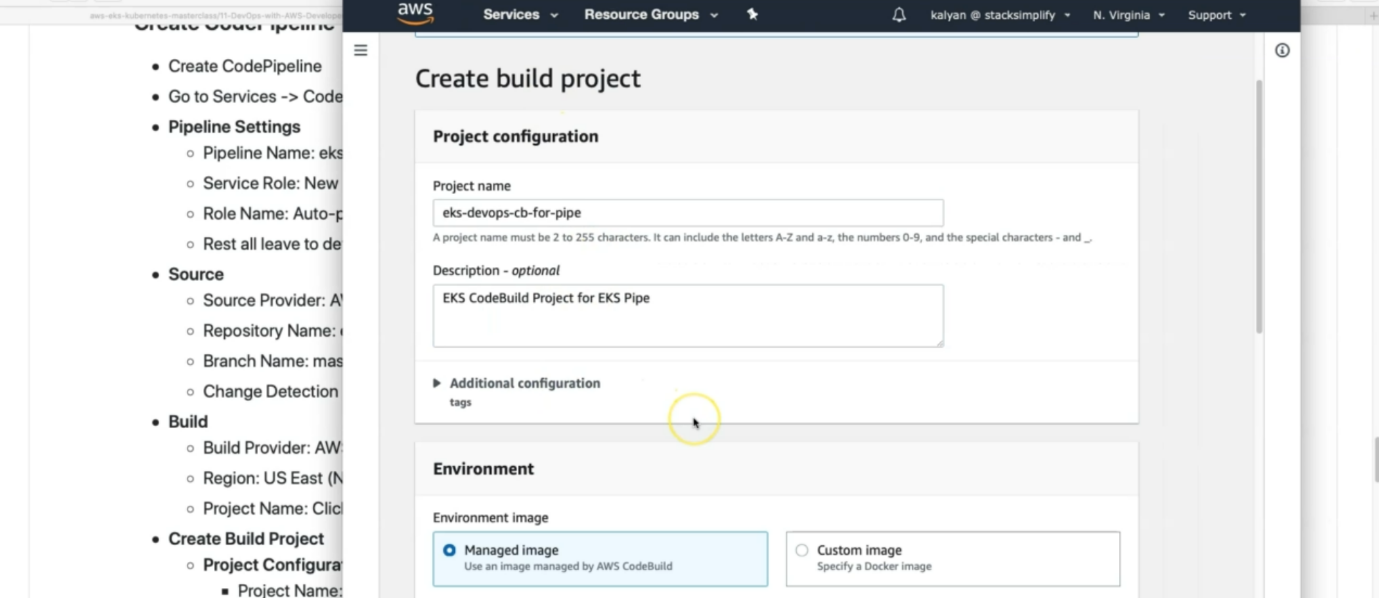
--- for build provider, we have 2 options

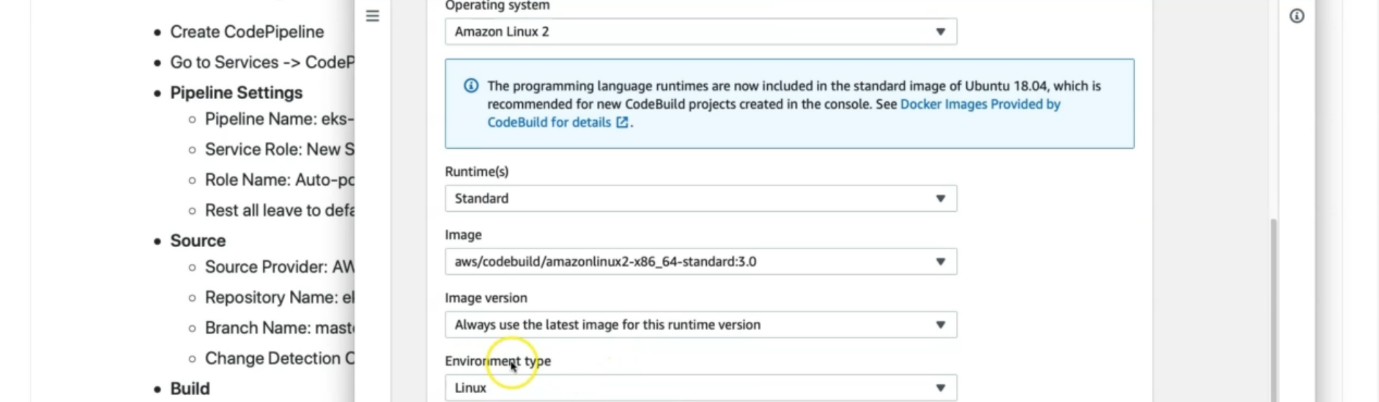
1. Aws codebuild
2. Jenkins

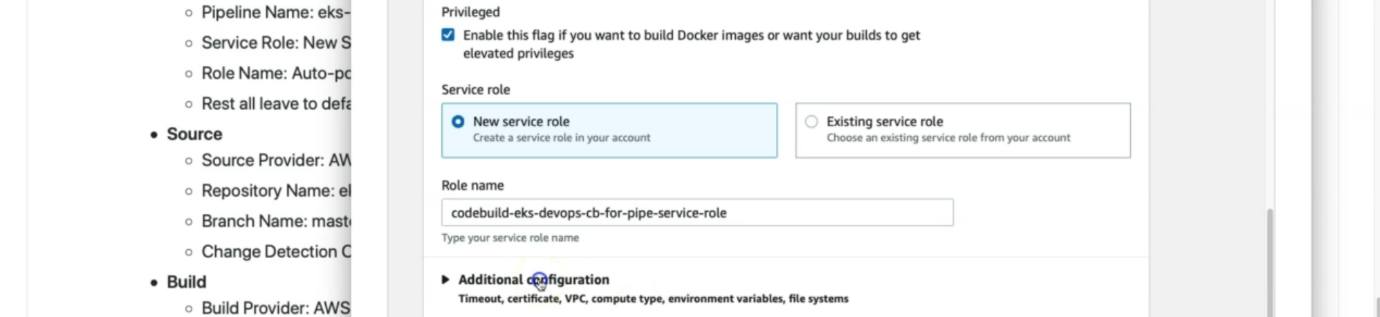
--- I will choose aws codebuild option.



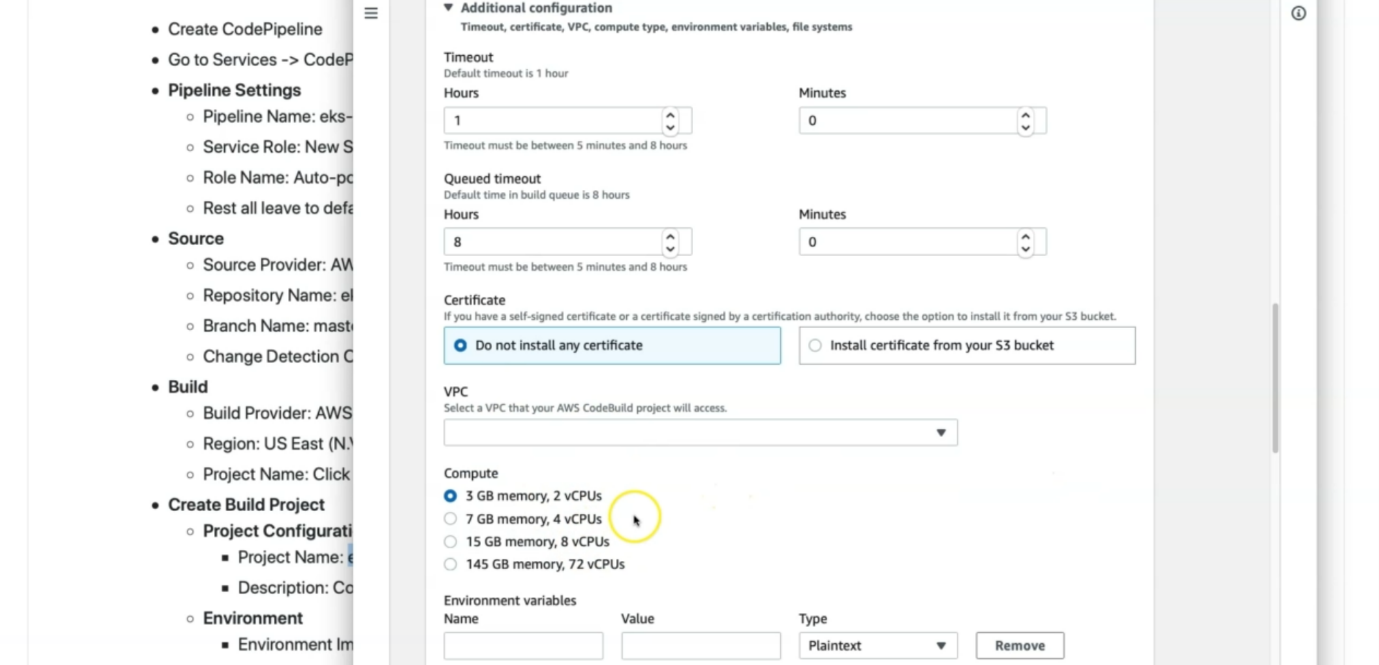
--- click on create project.

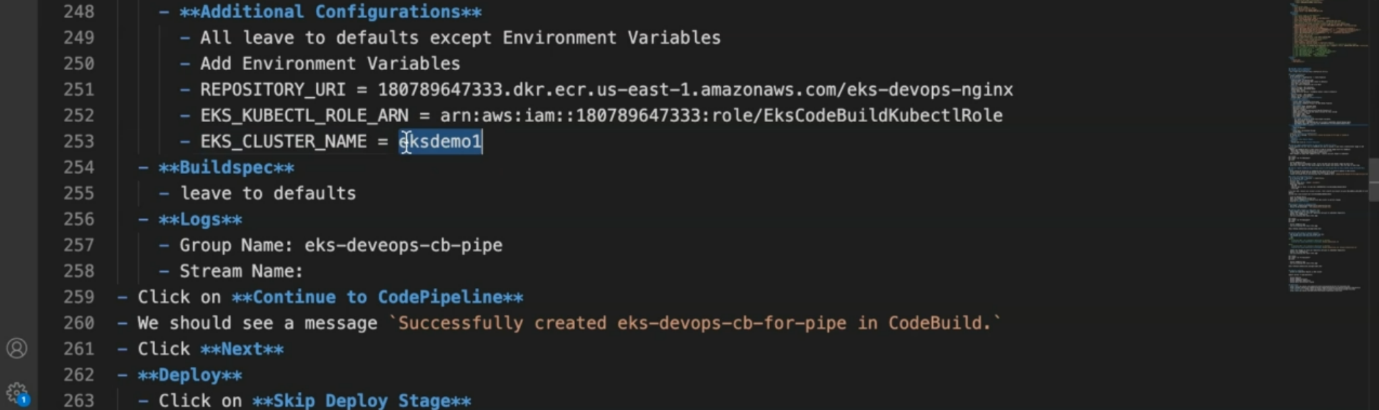




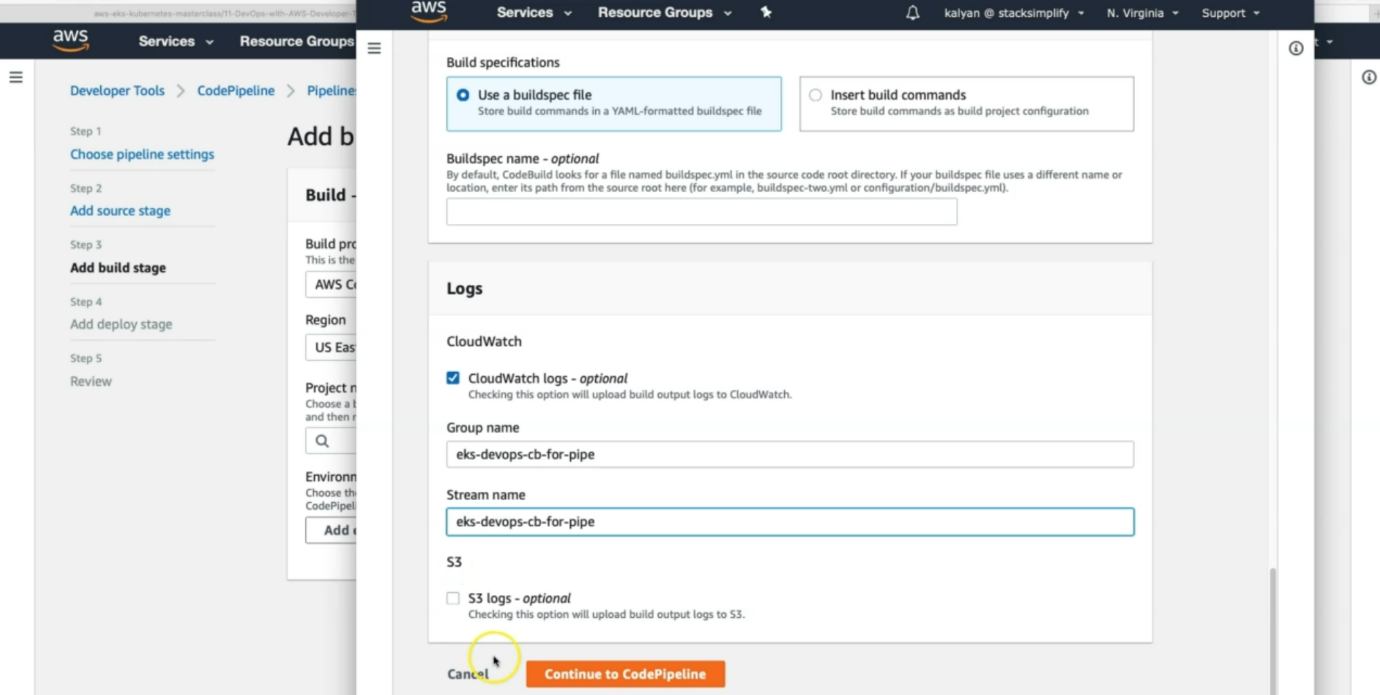


--- click on additional configuration.

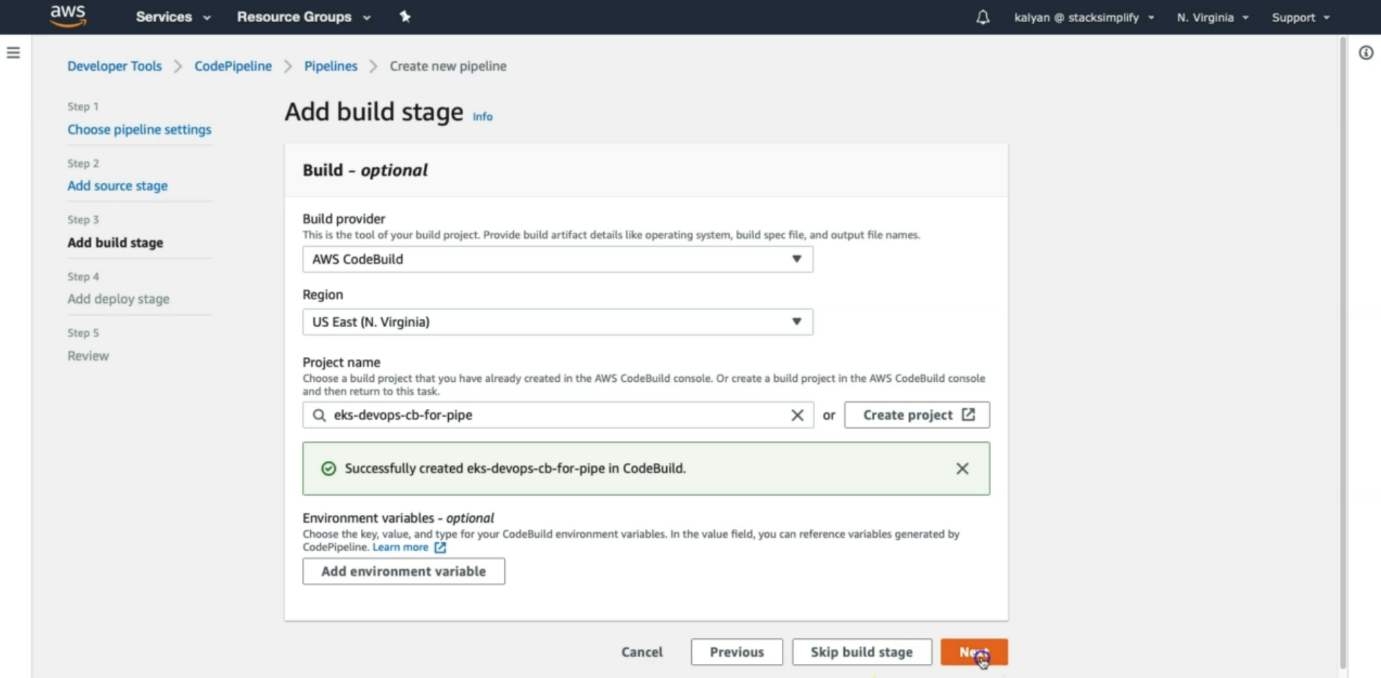




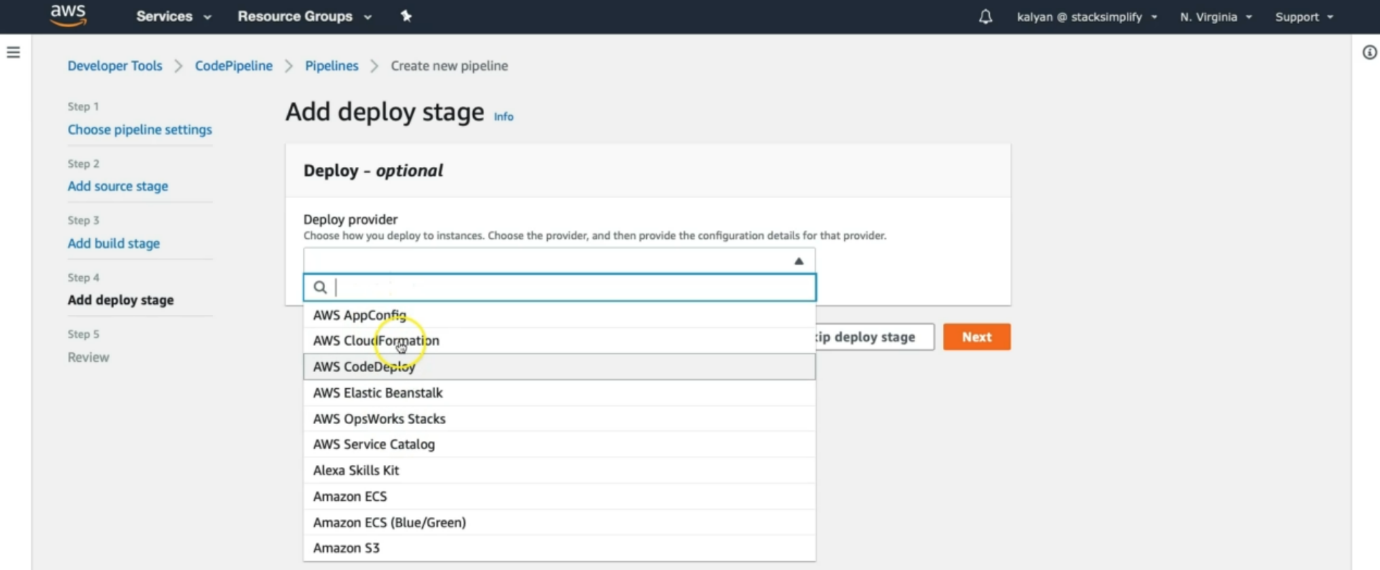
--- note – save those environmental variables in aws environment section.



--- this part is over. Click on continue to codepipeline.



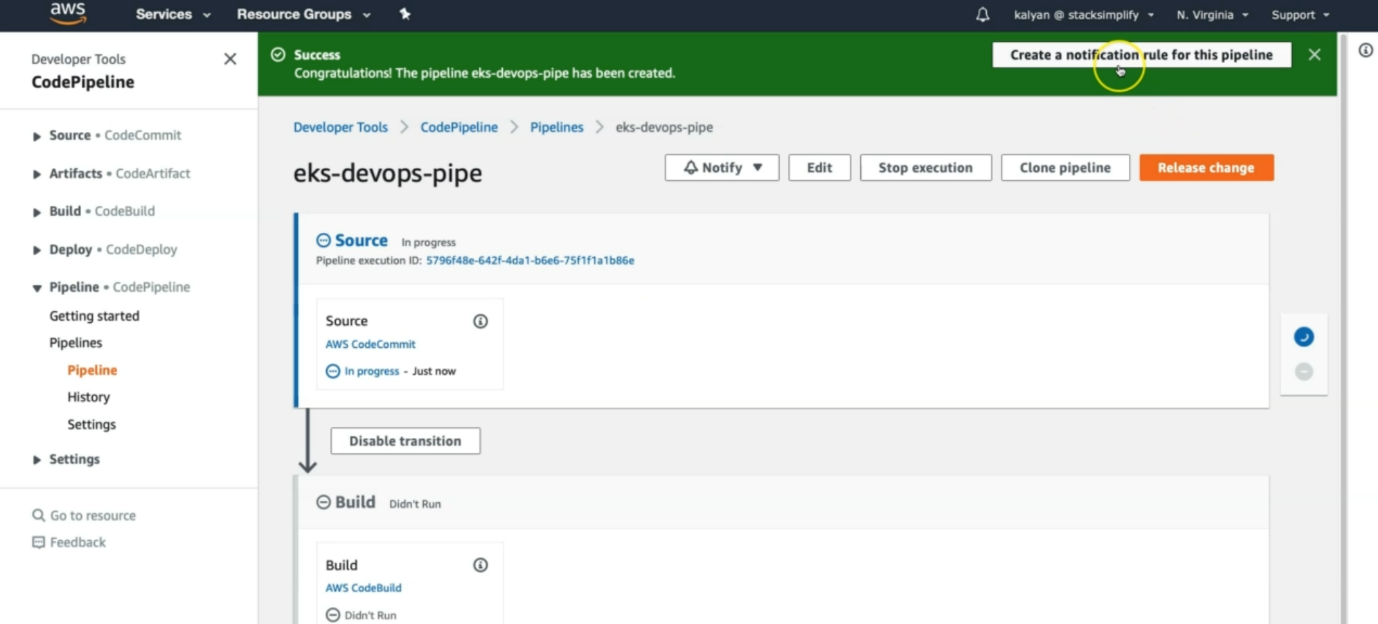
--- click on next.



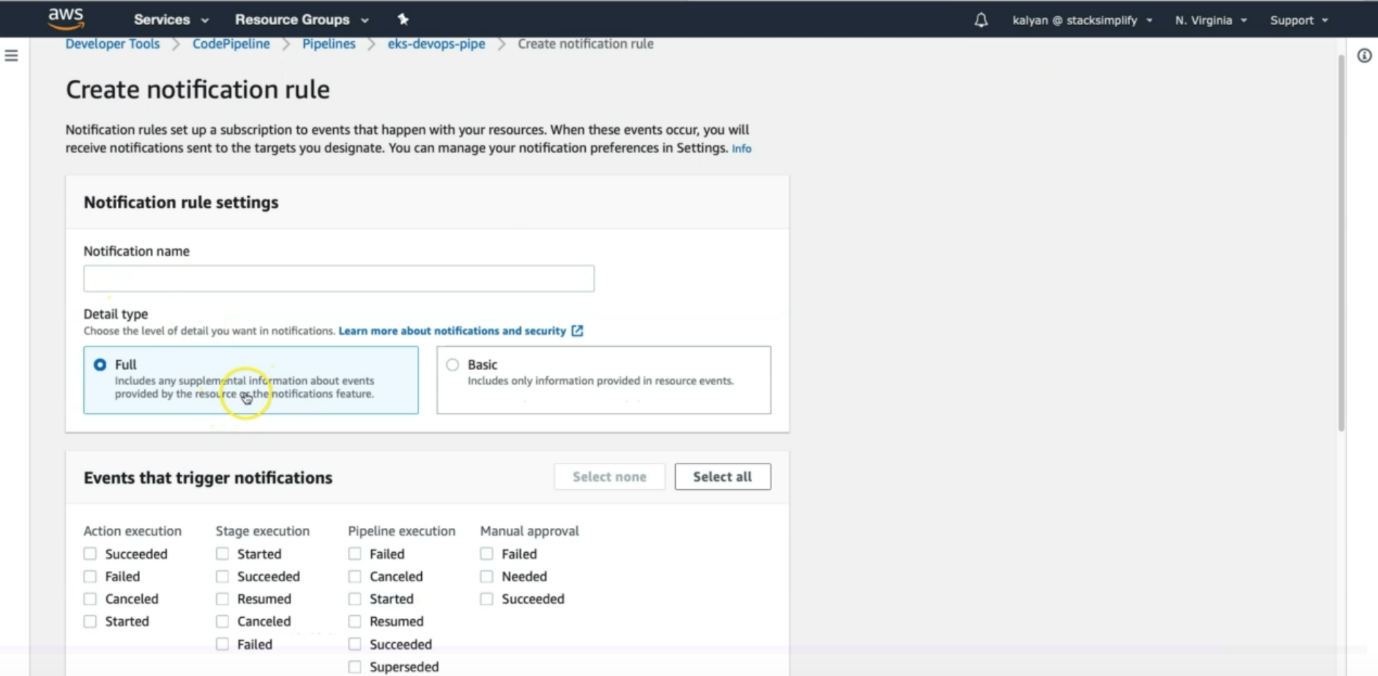
--- we don’t have anything to our respective EKS. We leave it as is. Click on next. Skip deploy stage.

--- this pipeline going to fail, we will see, how we going to trouble shoot it.

**Create a notification rule for this pipeline**



--- click on create a notification rule for this pipeline.



--- you can send notification for above events.

