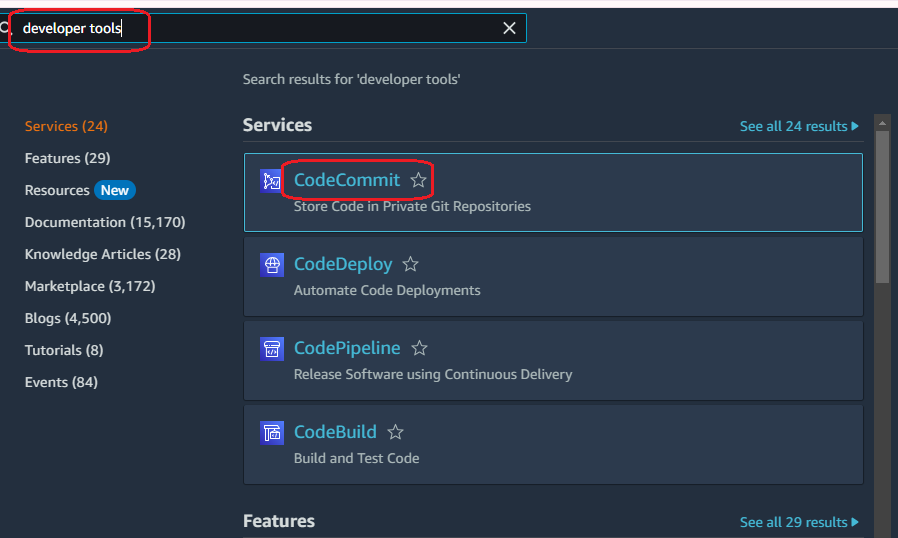
1. Prerequisites:
   1. AWS Account
2. Getting files:
   1. You can find the necessary elements on the next repository:

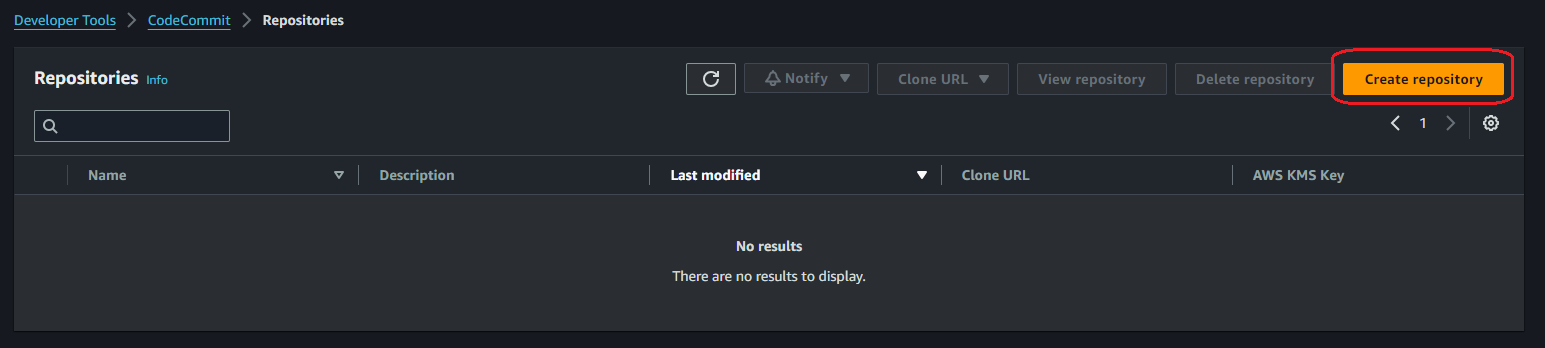
<https://github.com/IvanPdrk/Aws-cicd/tree/master>

* 1. Download the files.

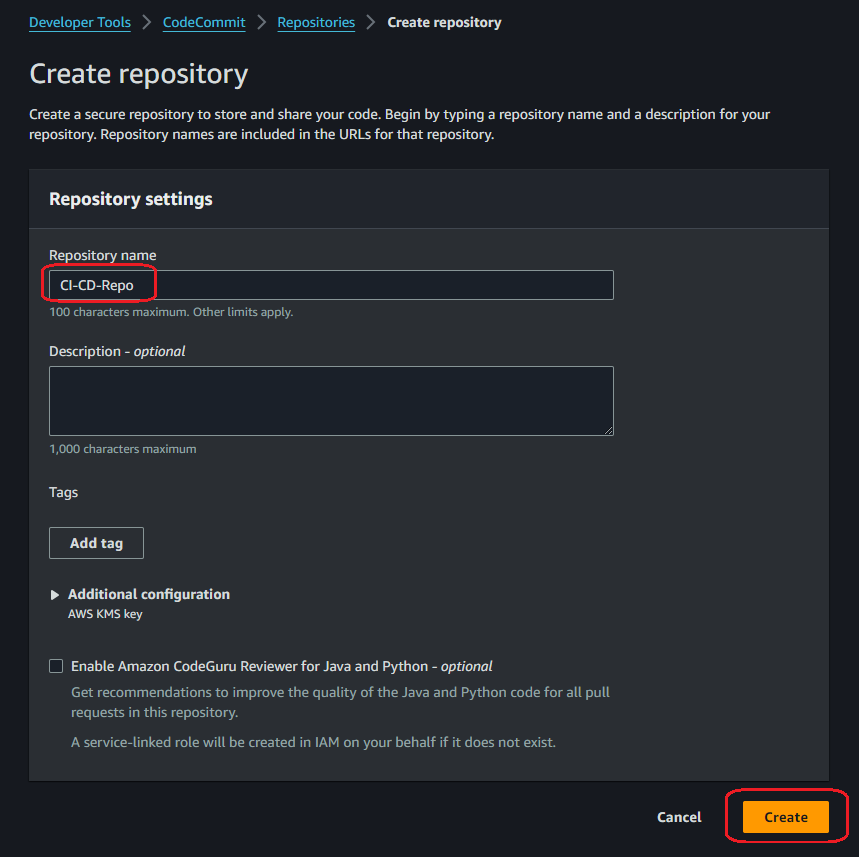
1. Creating repository:
   1. Go to your AWS Console
   2. Look for “Developer tools”
   3. Click on “CodeCommit”:



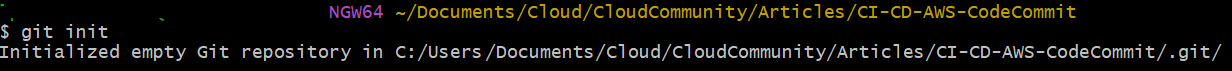
* 1. Click on “Create repository”



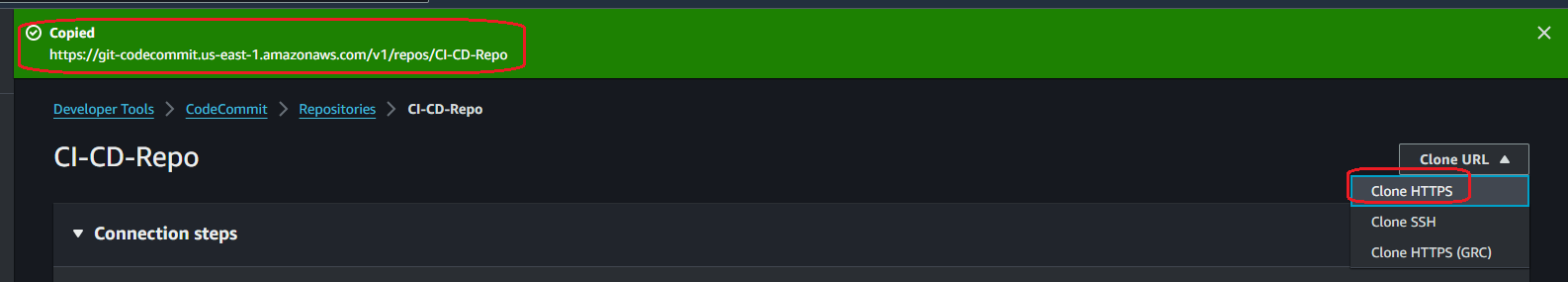
* 1. Assign it a name and click on “Create”:



1. Adding the files to our repository:
   1. Open a console (I prefer to use Git bash, but you can use PowerShell or windows console).
   2. Go to the path where you download the files.
   3. Execute the next command: git init

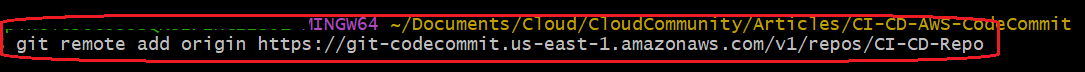


* 1. Go to your repository, click on “Clone URL”, you will receive a confirmation message:

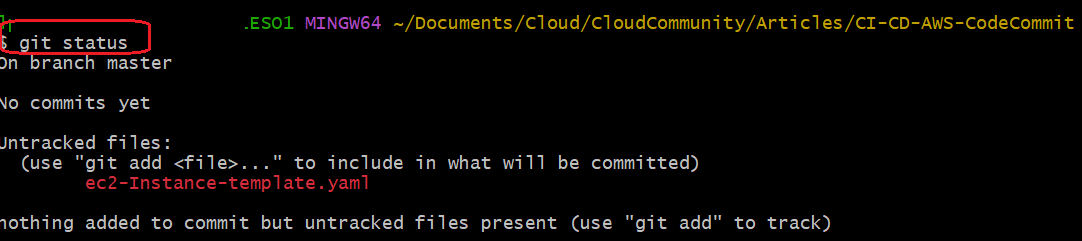


* 1. Go back to your console, execute the next command:

git remote add origin YOUR\_REPO\_URL

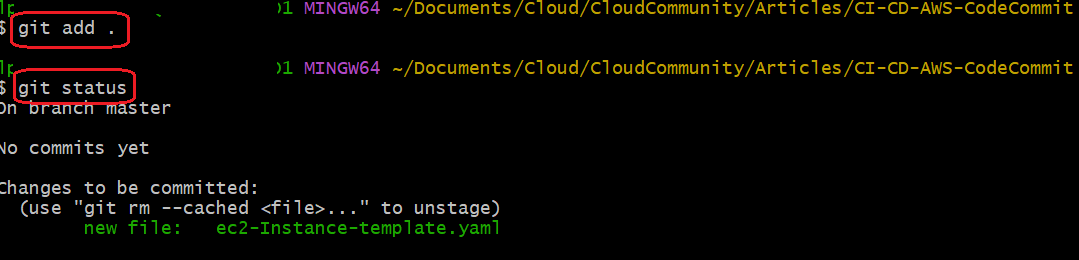


* 1. You can verify the files’ status with the command: git status



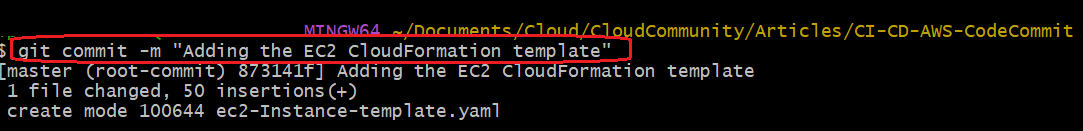
* 1. You must add the files to be consider for the commit, use the command: git add .

If you validate the status, it changed:

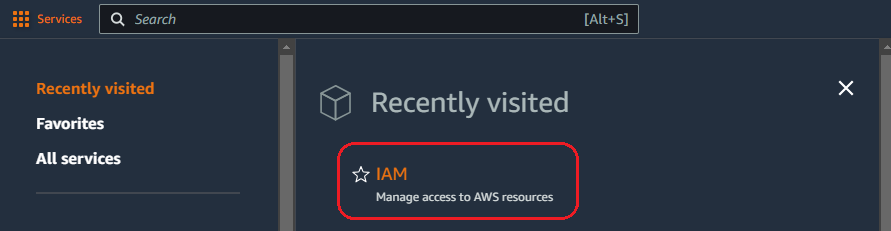


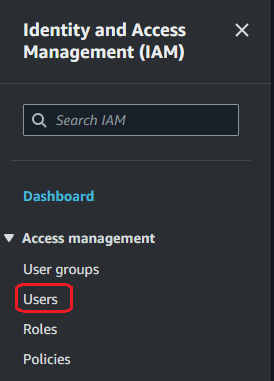
* 1. Now, we can do the commit: git commit -m “Message”

In the “Message” you can specify whatever you want.

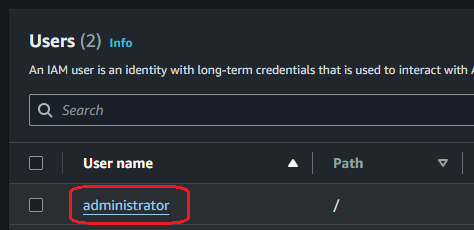


* 1. Go back to your AWS Console, look for IAM, click on “Users”:

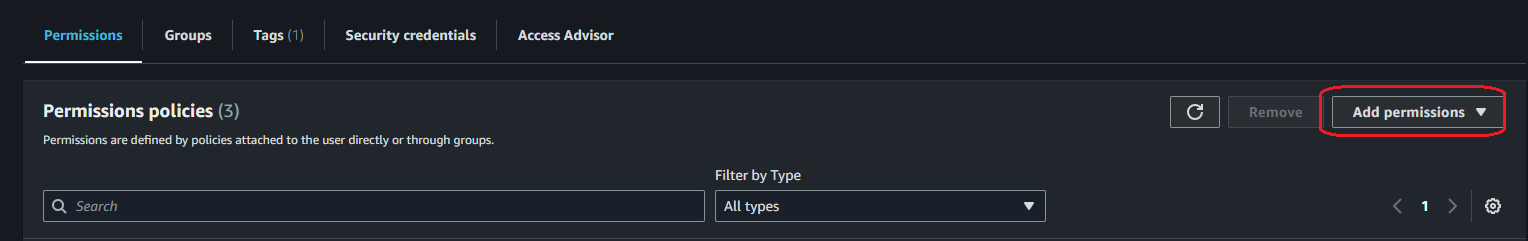




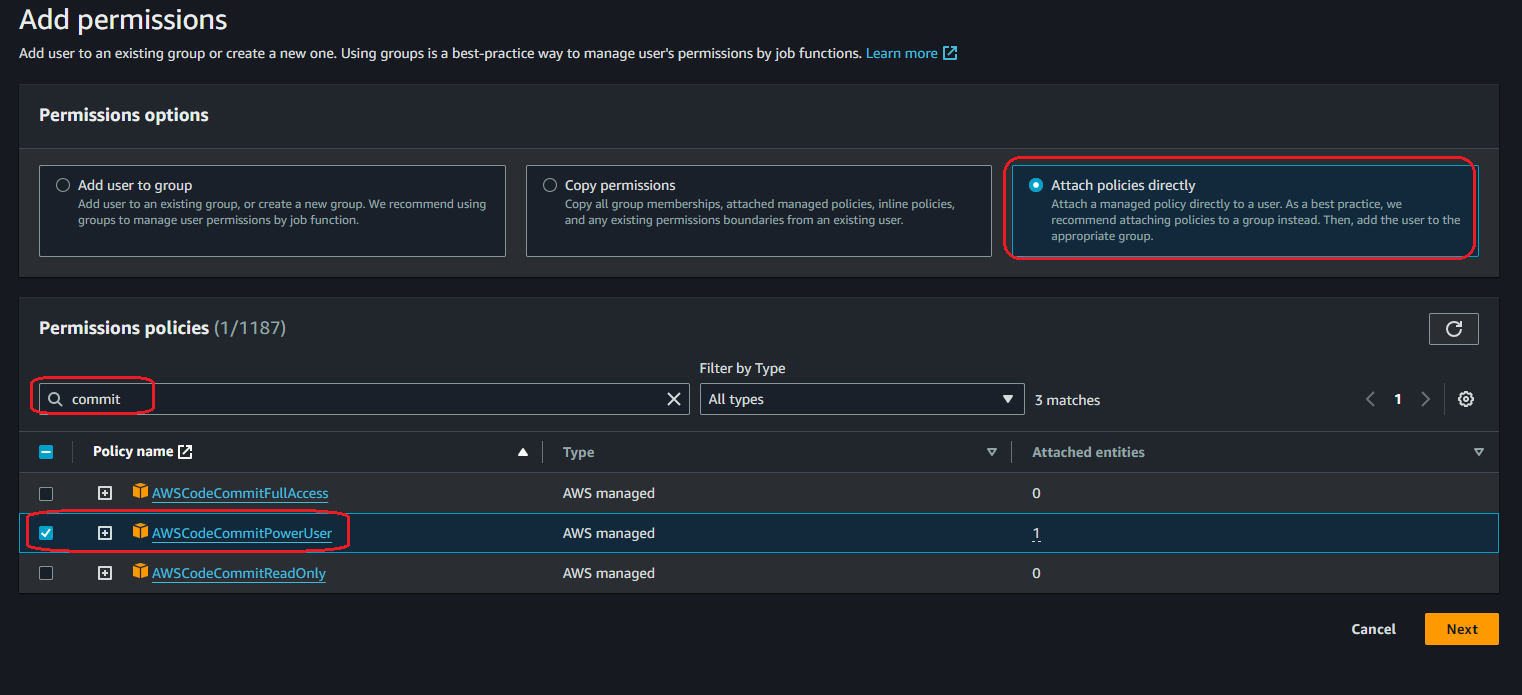
* 1. Click on the user you are using for this article:



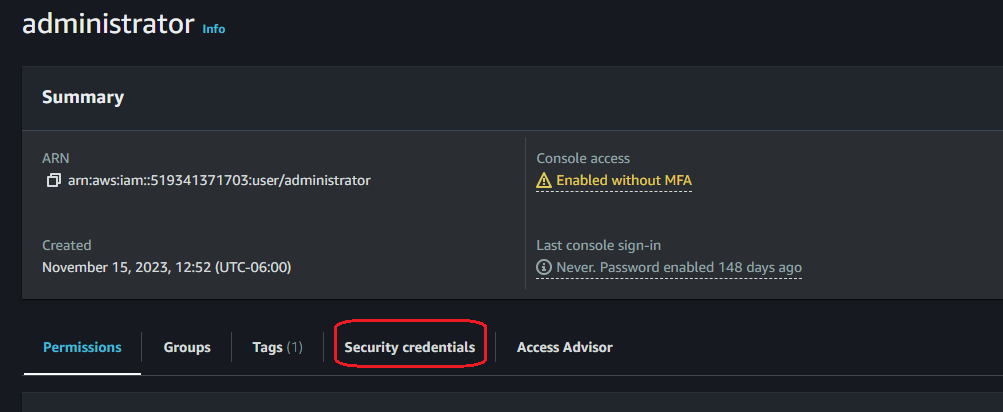
* 1. Click on “Add Permissions”:



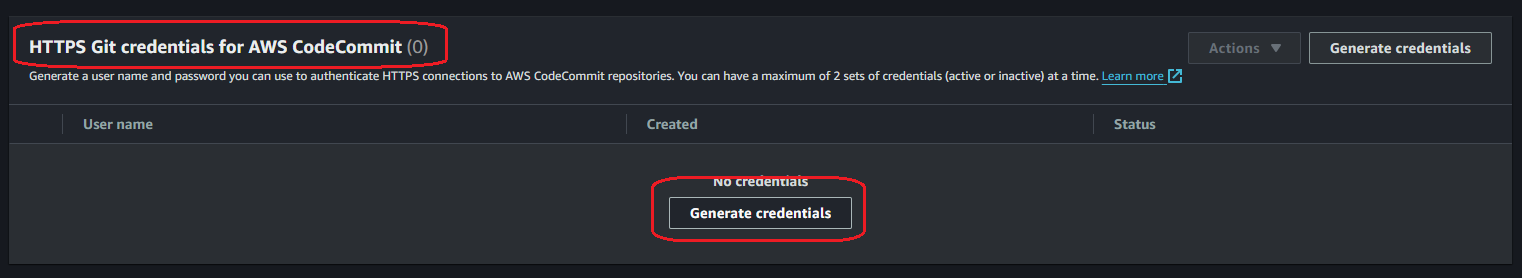
* 1. Select “Attach policies directly”, look for commit, select “AWSCodeCommitPowerUser”, click on next, and in the next page click on “Add permissions” :



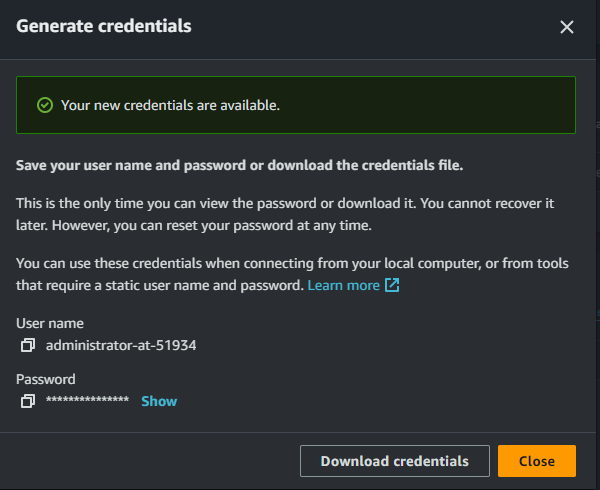
* 1. In the User tab, click on “Security Credencials”:



* 1. Look for “HTTPS Git credentials for AWS CodeCommit”, click on “Generate credencials”:

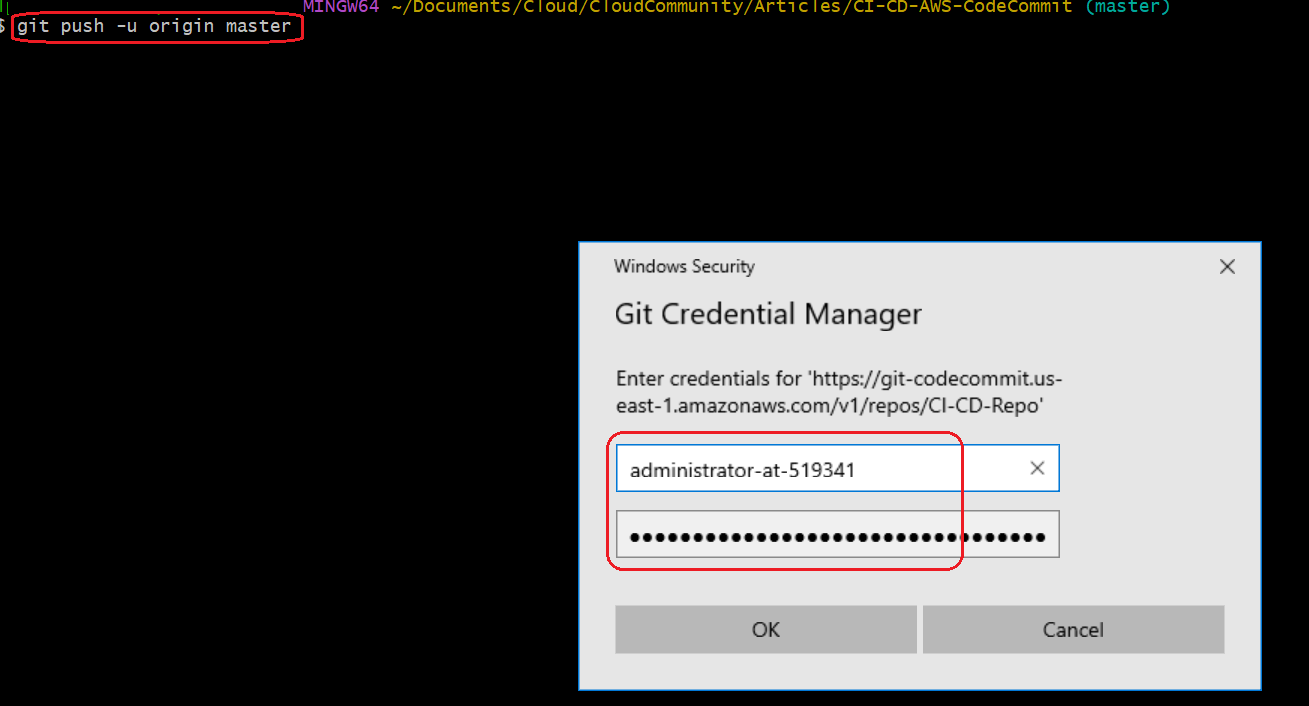


* 1. Click on “Download credentials”, save the file in a secure path:

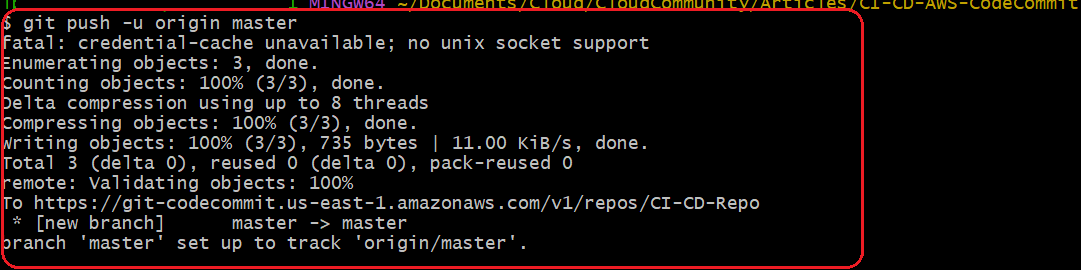


* 1. Go back to the console, execute the command: git push -u origin master

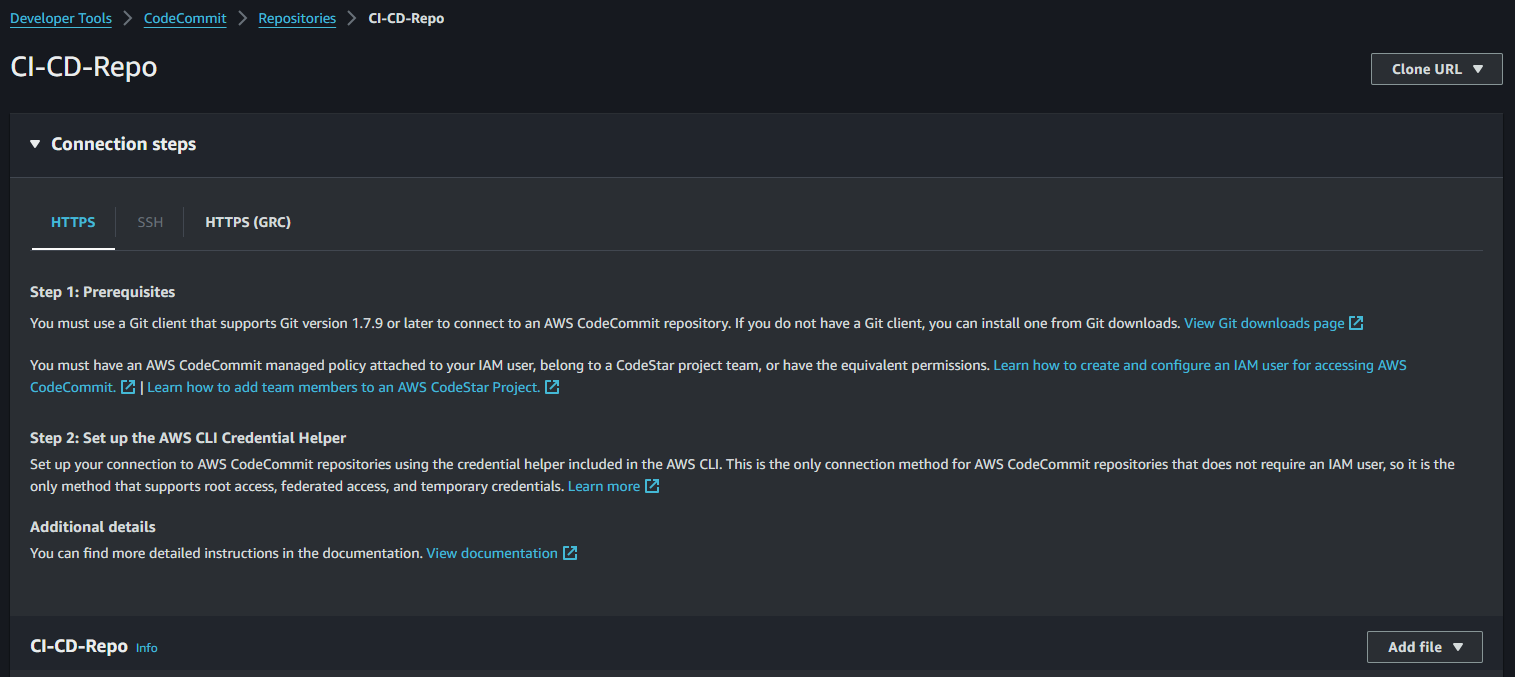
In the “Git Credential Manager” window, copy and paste the credentials generated in the last step.



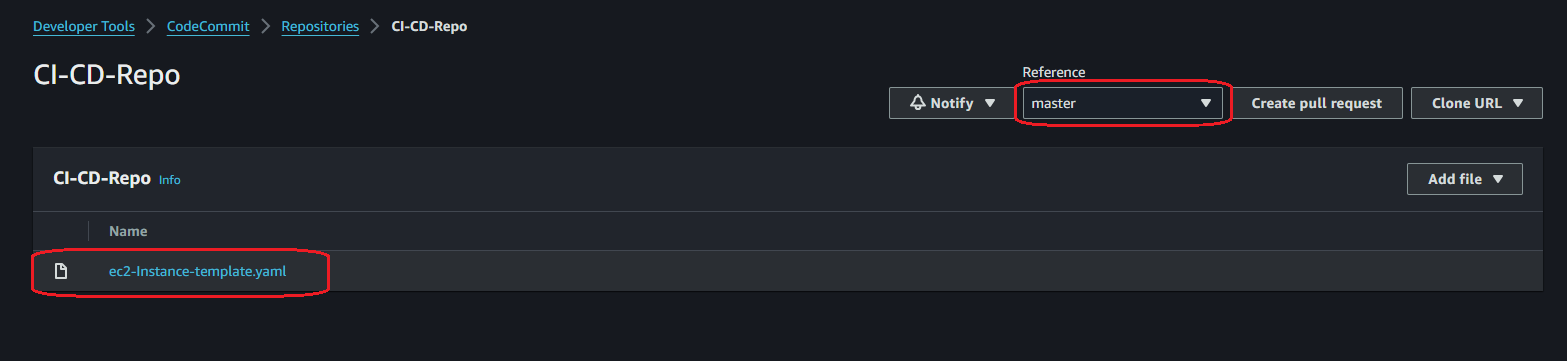
* 1. You will see the next result:



* 1. Your repository was as the next image:

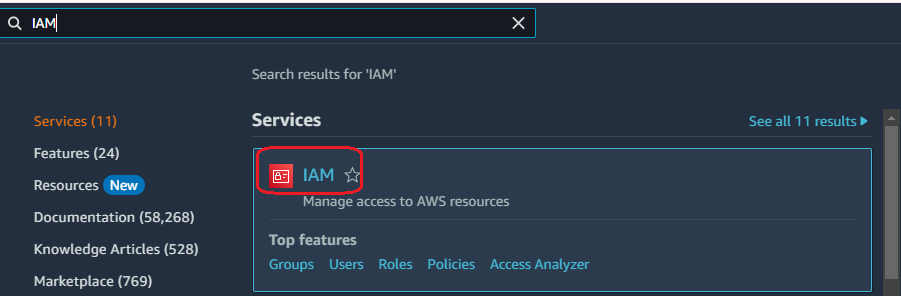


Once you refresh, you will see your repository as the next image:

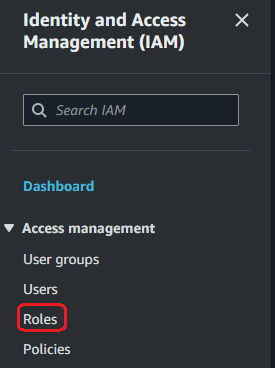


We have our file in our CodeCommit repository, in the “master” branch.

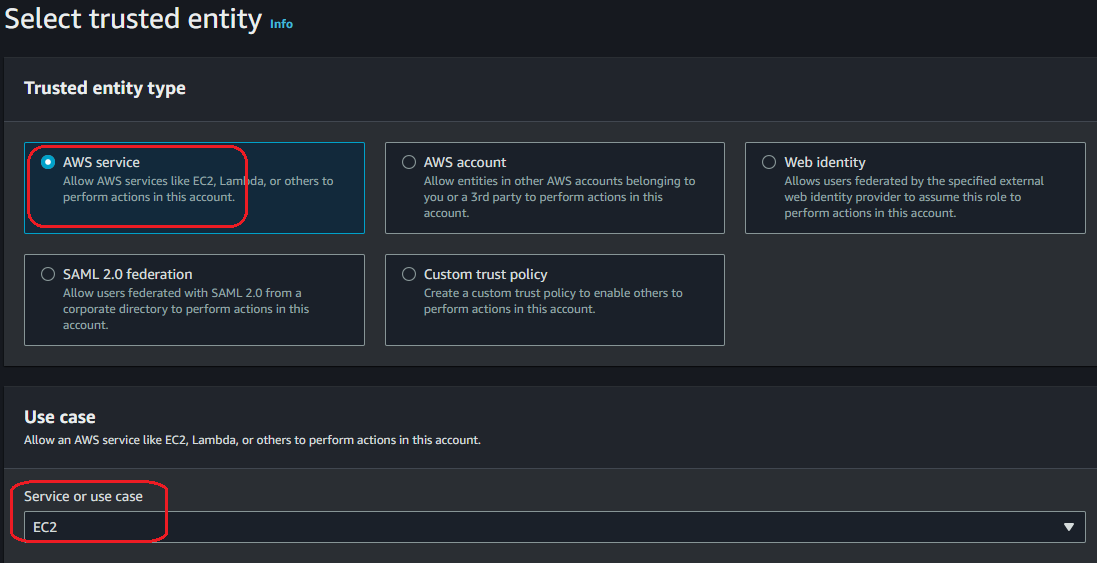
1. Role for our Pipeline
   1. Look for “IAM” service:



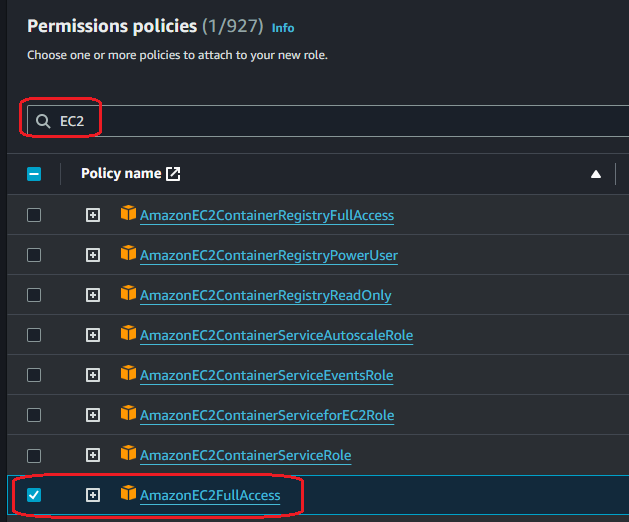
* 1. Click on “Roles”



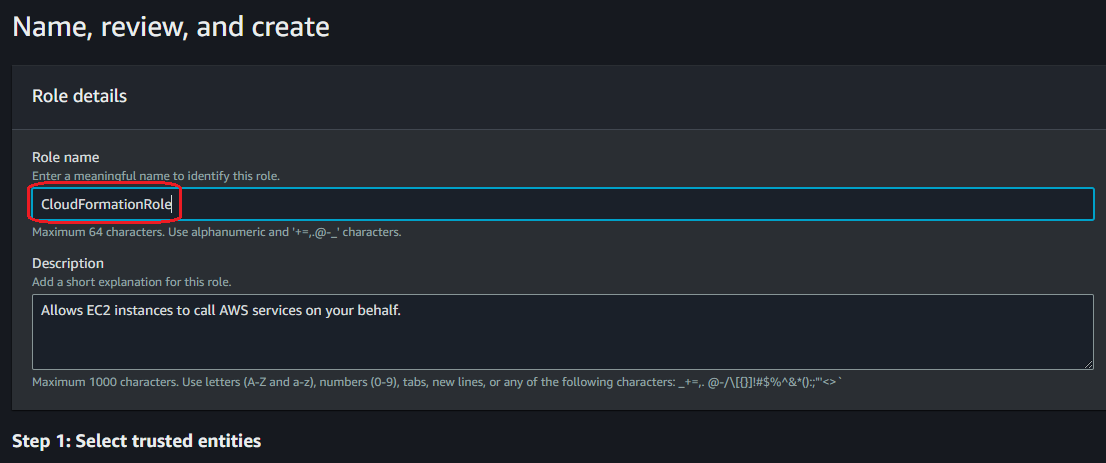
* 1. Check “AWS service” option, select “EC2”, and click on “Next”:



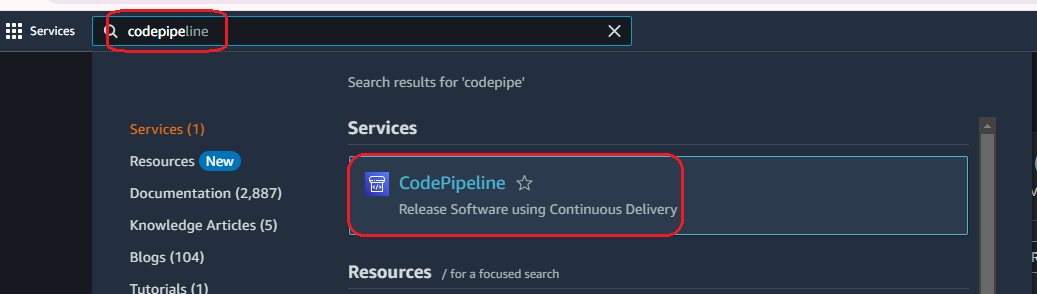
* 1. Look for “EC2”, select the “AmazonEC2FullAccess” policy:



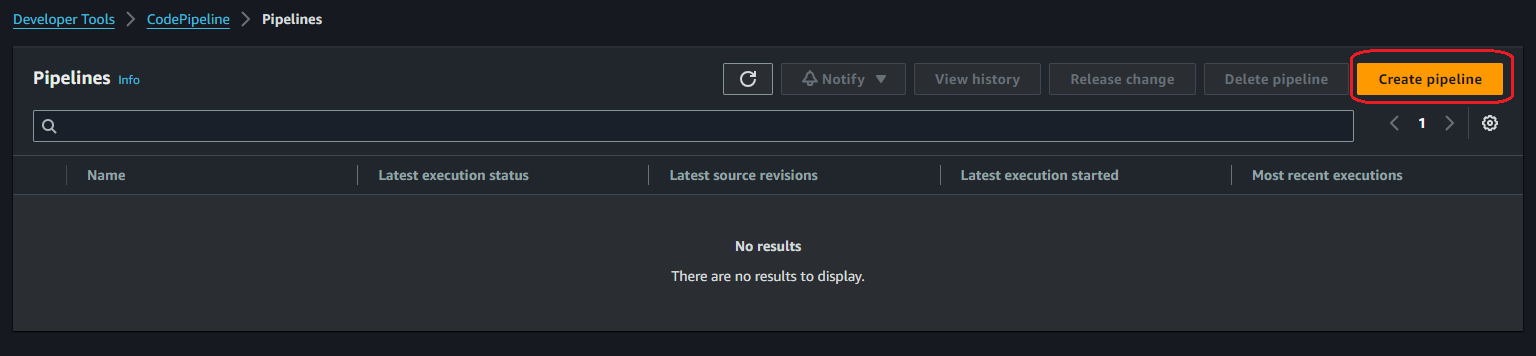
* 1. Assign it a name, and click on “Create Role”:



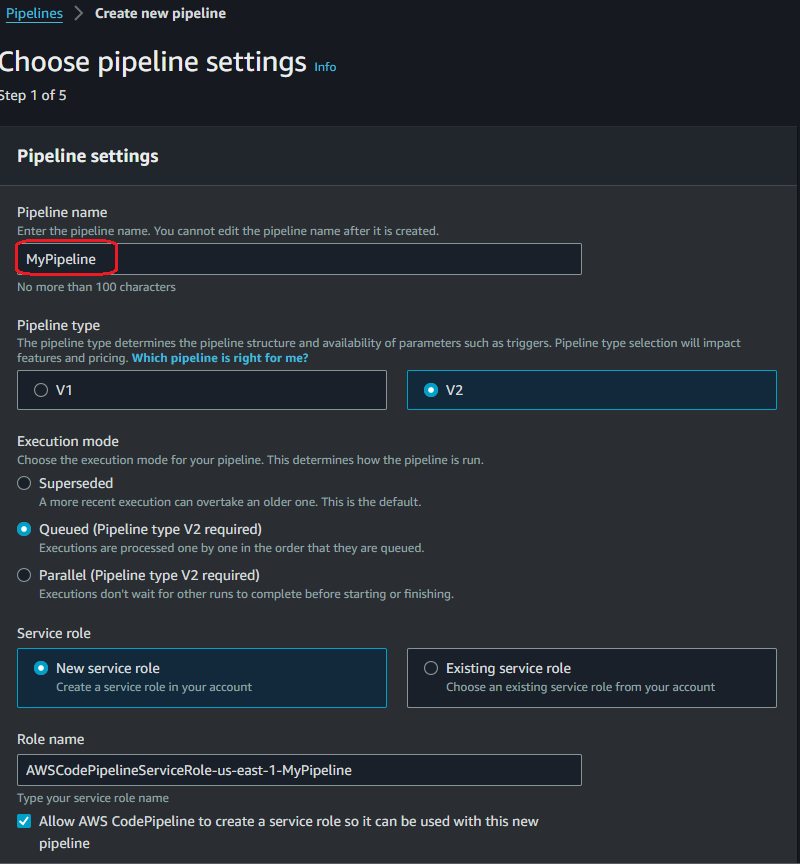
1. Creating our Pipeline
   1. Look for “CodePipeline”:



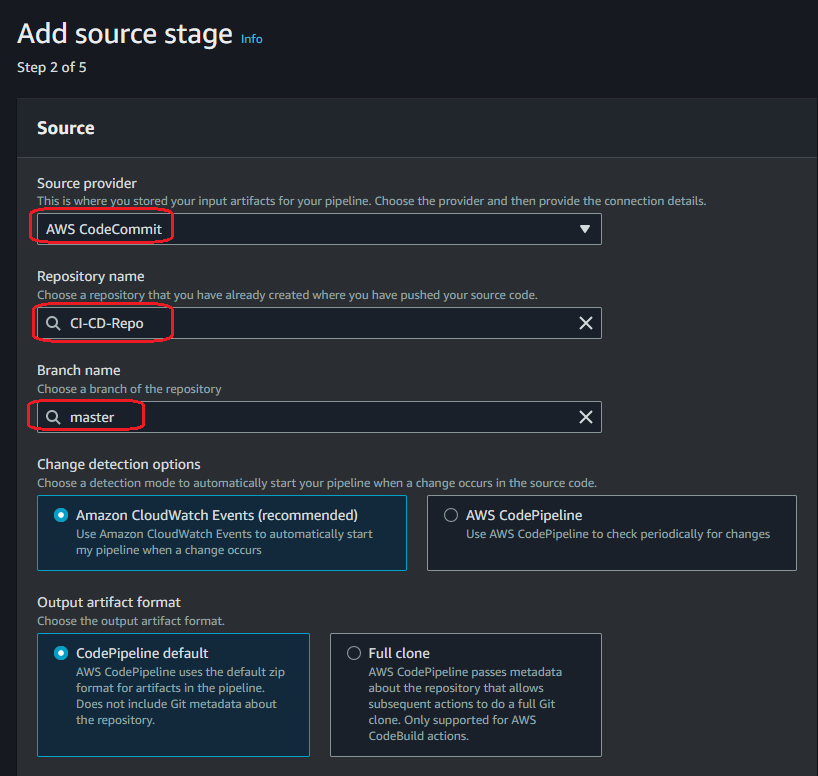
* 1. Click on “Create pipeline”



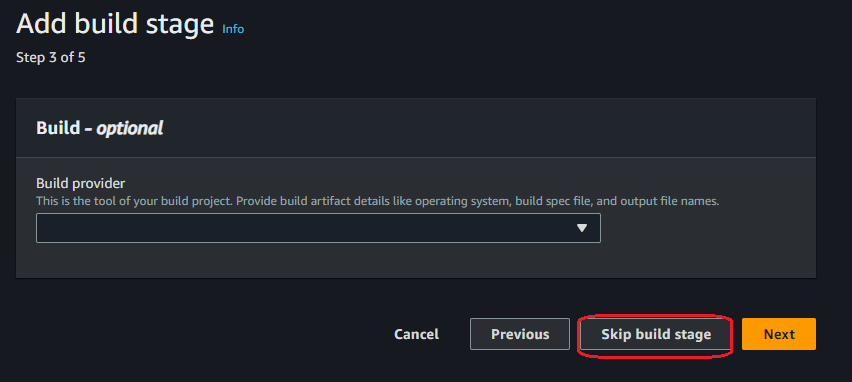
* 1. We need to assign a name, the other properties can stay in the default way:



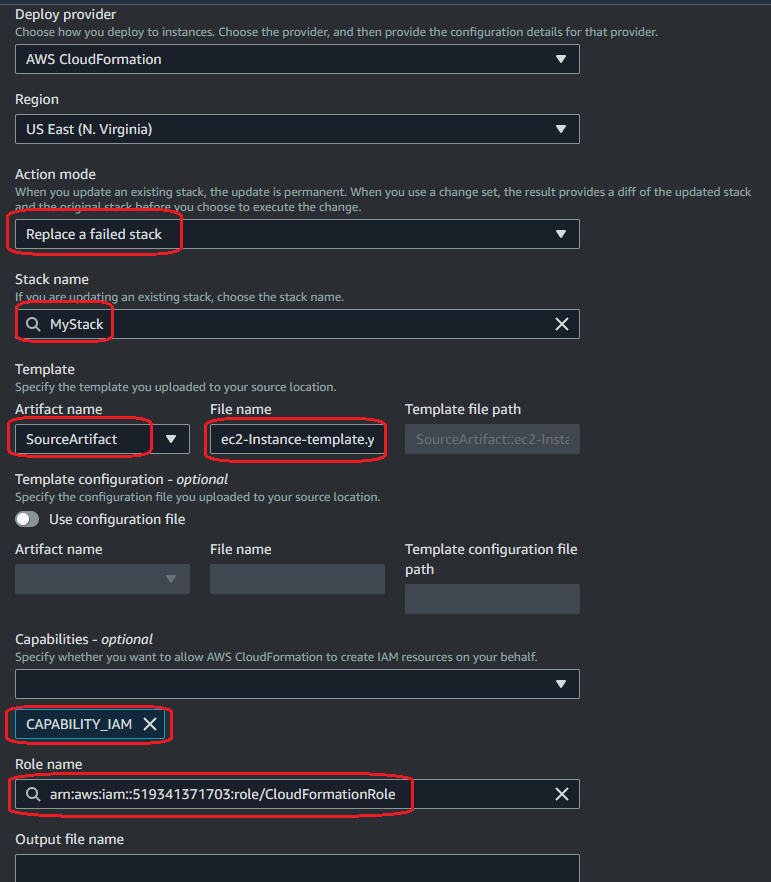
* 1. In the “Add source stage” option, we need to select “AWS CodeCommit”, look for your repository, select your “master” branch, and click “Next”:



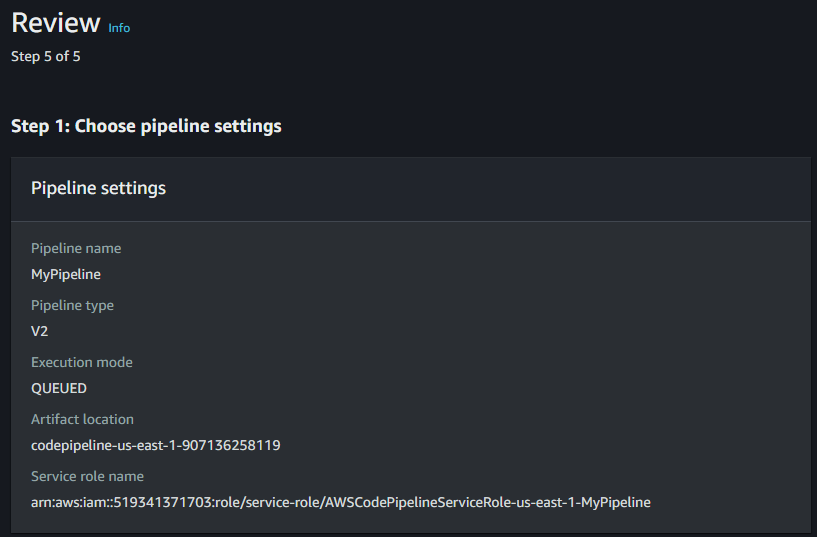
* 1. We can skip the “Add build stage” option:



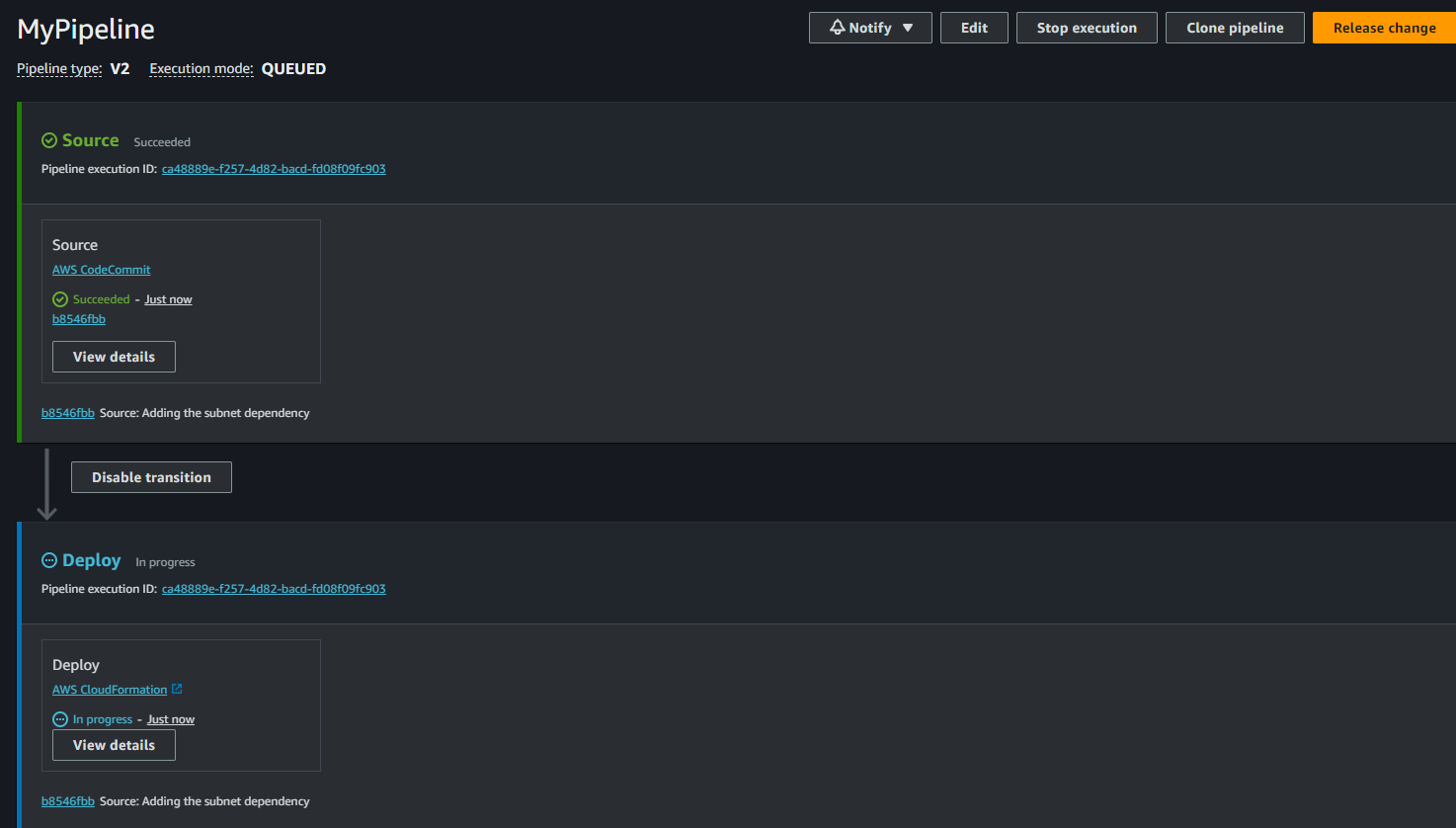
* 1. In the “Add deploy stage” option, we need to define “AWS CloudFormation” as Deploy provider, you must define the next configuration, click on “Next”:



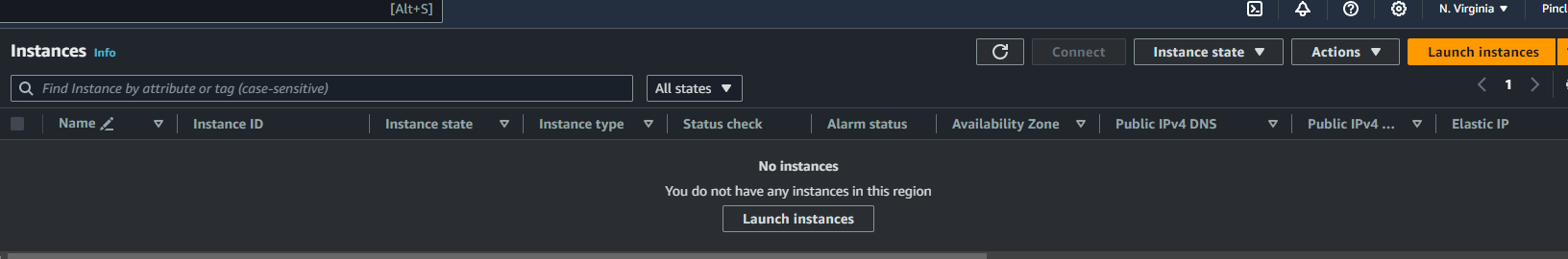
* 1. In the next stage, you will see the review about your configuration, click on “Create Pipeline”

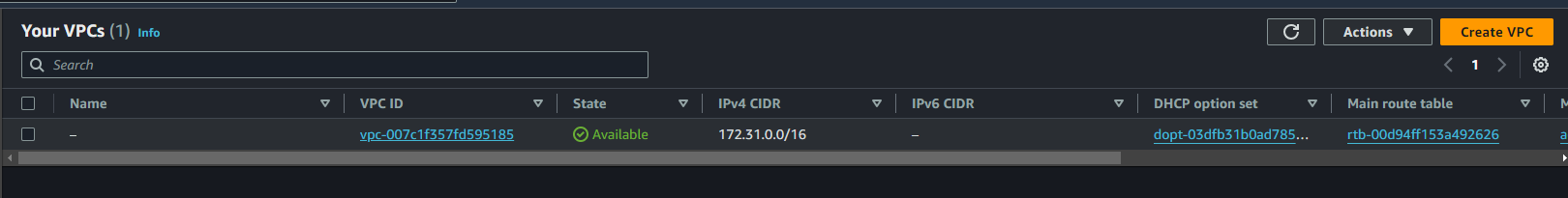


* 1. You will see the next result, the Deploy will take time, you must wait until it finishes:

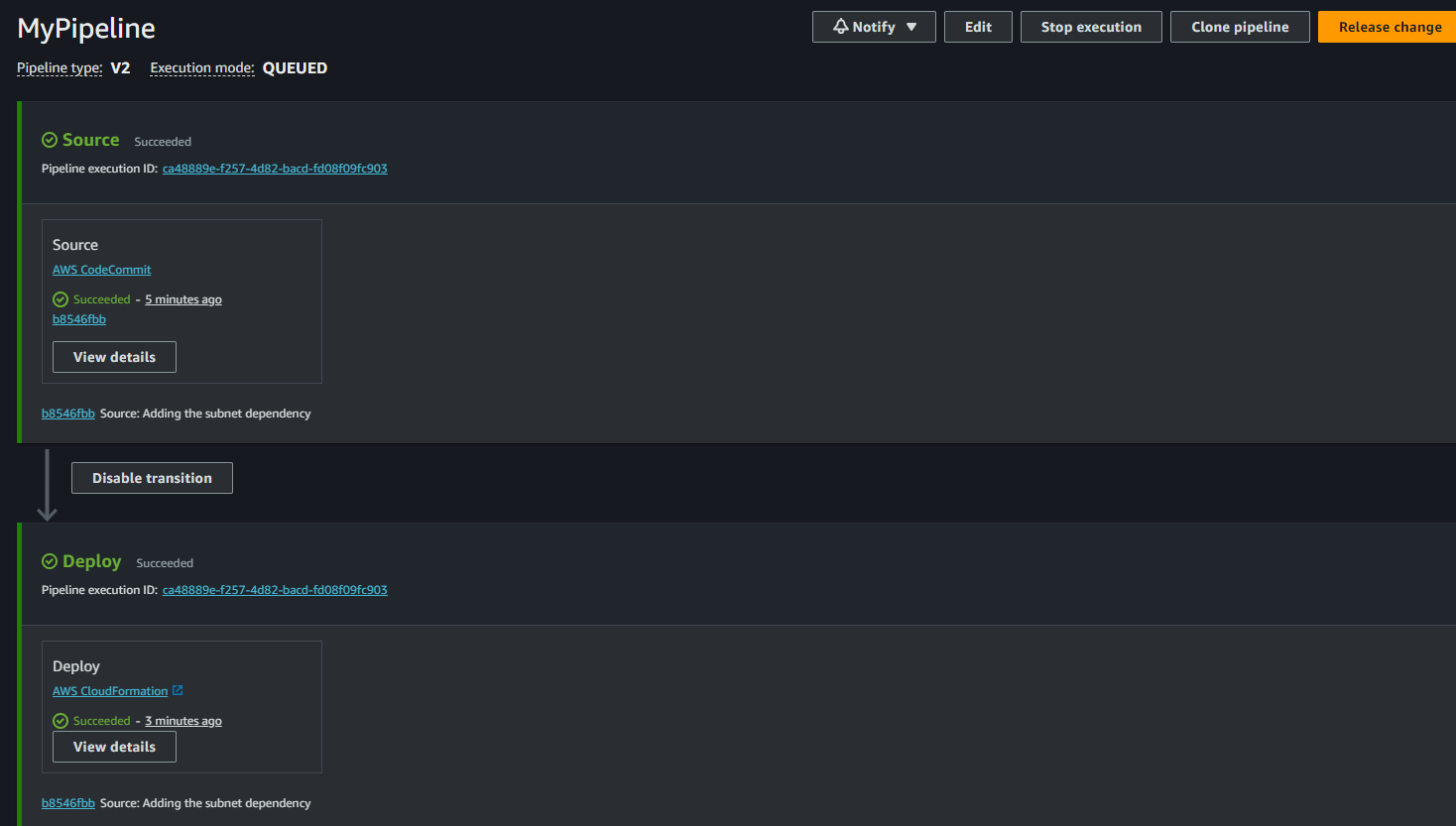


In the middle time let me explain you some details, before creating your pipeline, the vpc and EC2 dashboards do not have resources (only the default VPC):

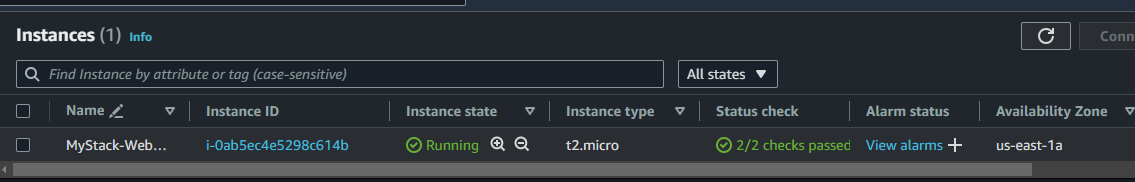




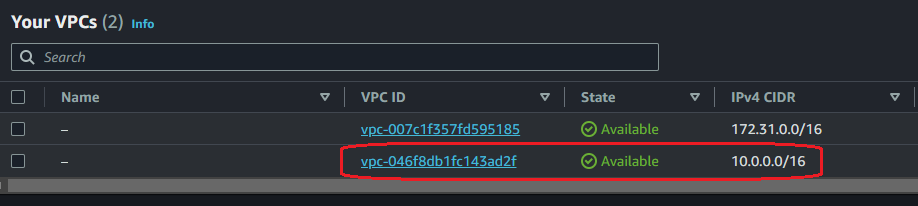
Once the Pipeline finishes, you will see the next result:



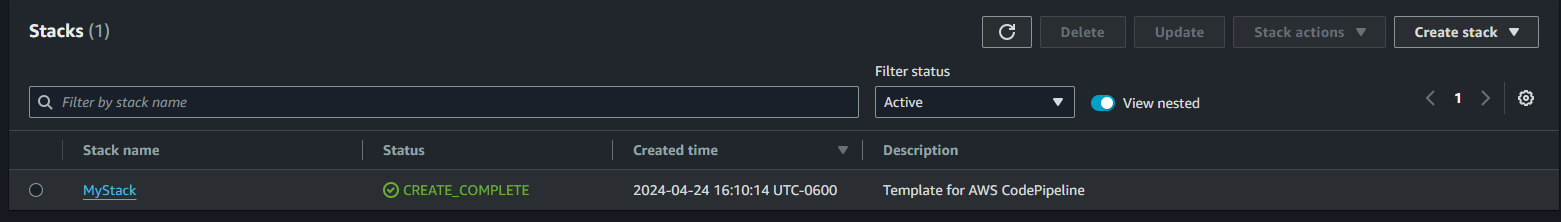
In the EC2 dashboard you can find your instance:



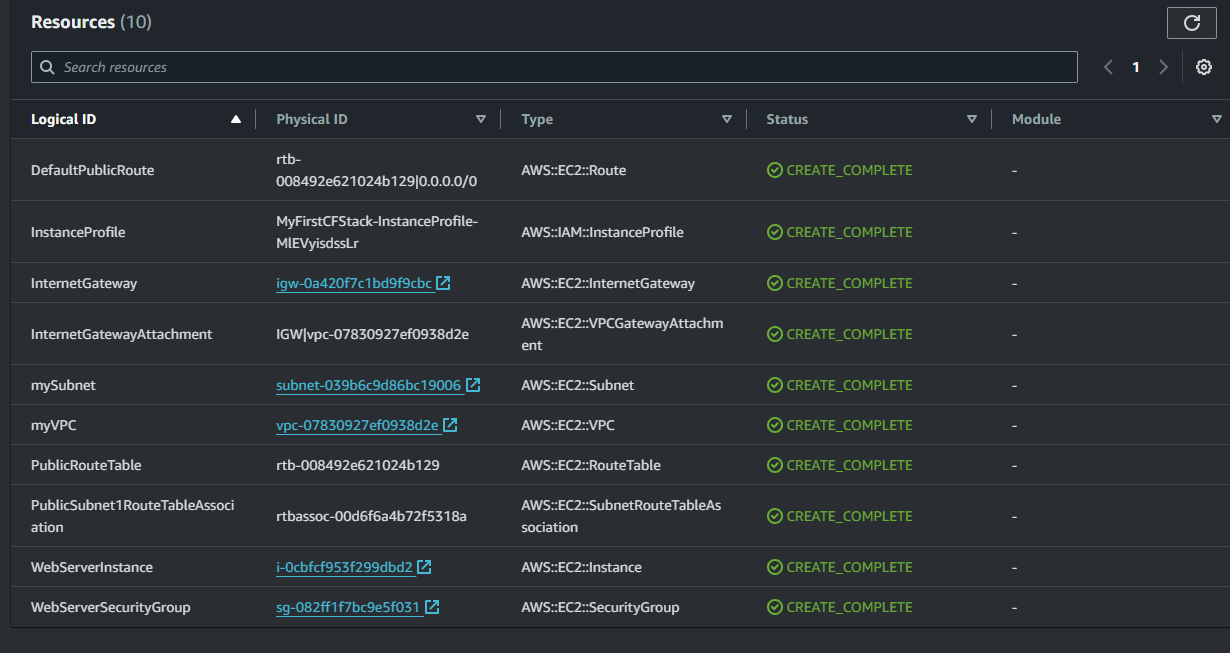
In the VPC dashboard you will see your new VPC:



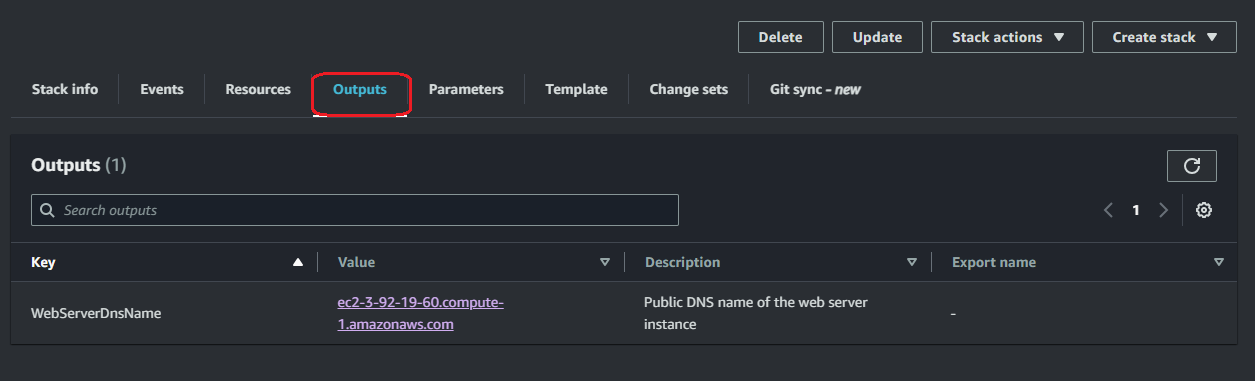
And if you look for the “CloudFormation” service, you can find your Stack:



We can see our resources:



In the Outputs tab, we will find the instance DNS:



If you click on the “Value”, we will open the NGINX page that we configured in the CloudFormation template file:

