Cloud Plus Plus Services

Create an automated software release pipeline (CI/CD pipeline) using AWS CodePipeline and CodeCommit.

Tutorial Objectives:

1. Learn to deploy your code every time there is a code change using AWS CodePipeline.

2. **Learn to use an Amazon CodeCommit as a source code repository**.

3. Learn to automate the software release process by continuous deployment of application code revisions.

Step 1: Create your Elastic Beanstalk Application

Log on to your AWS Management Console and search for **Elastic Beanstalk**.

Once the page opens, click on Create Application and mention the rest as follows:

- **Application Name: CodeCommitCICD**
- **Platform: PHP**

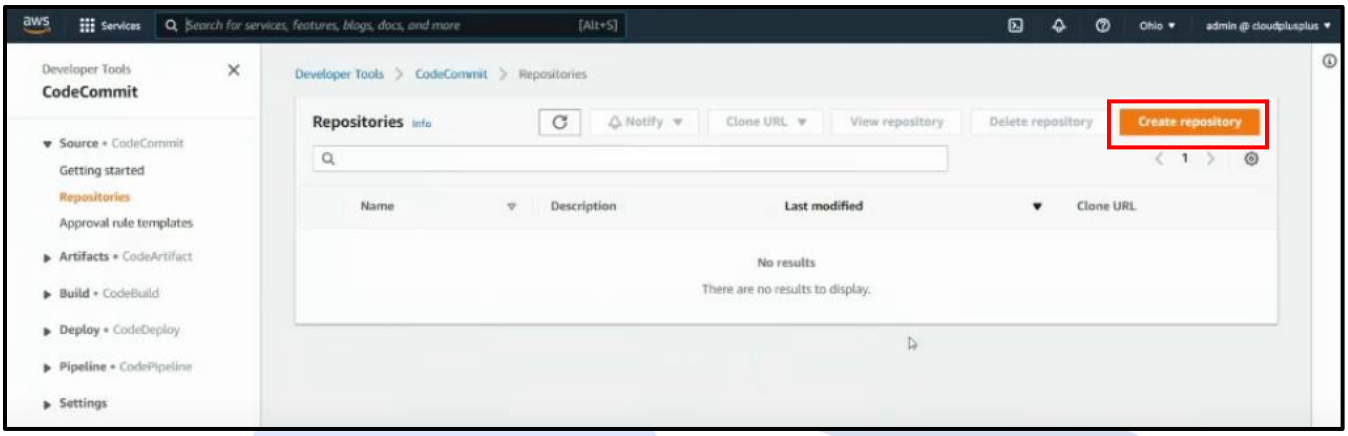
Upload your own code by downloading it from this link.

Note: In the zip file, you will find 3 folders within it. Make sure you upload 'V1' (Version 1) first.

Now, click on **Create Application**. You will find version-1 of the application deployed.

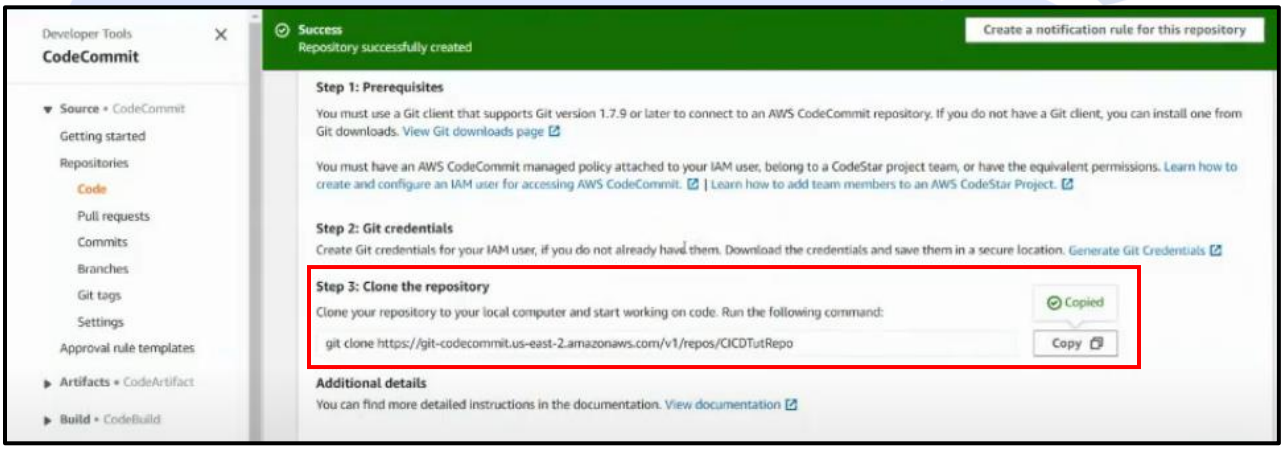
Step 2: Create a CodeCommit Repository

Open AWS Management Console on another tab and search for **CodeCommit**.



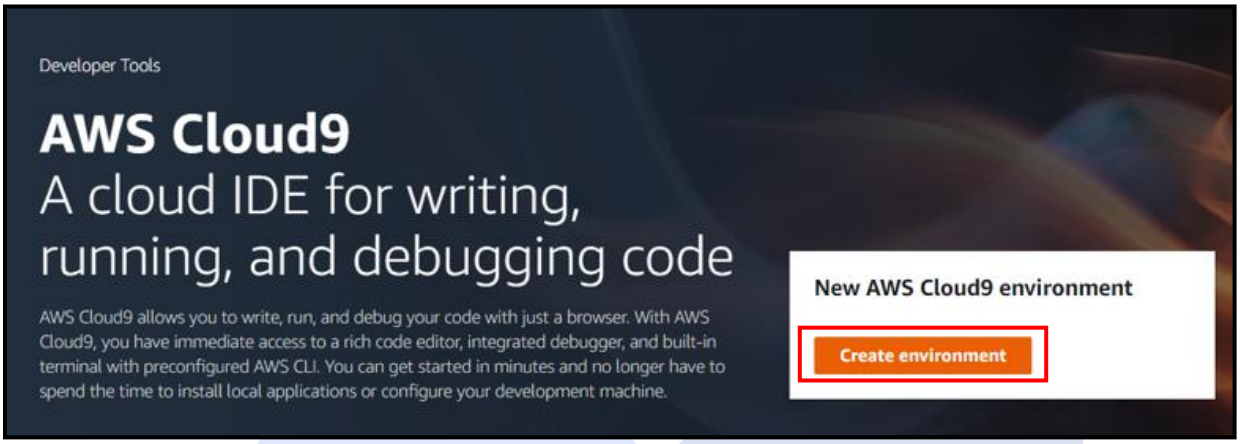
Fill in the Repository name as **CICDRepo** and by leaving the rest as default, click on Create.

Go ahead and Copy the command to clone your repository.



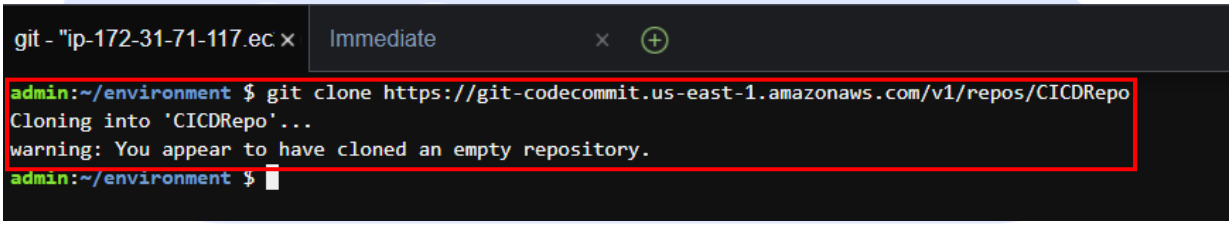
Step 3: Now, open your AWS management console in another new tab and navigate to Cloud9.

Click on **Create environment**



Give a name to your Environment, We'll call it **CodeCommit_Playground**

Once the environment is up and running paste the previously copied git clone command in the terminal.

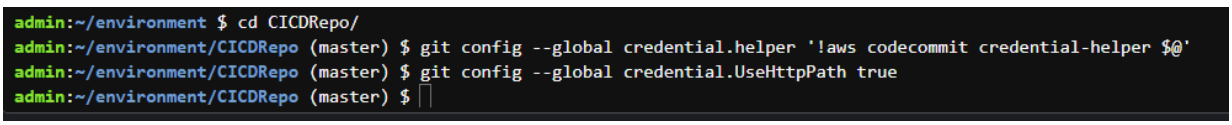


For remote accessing the CodeCommit bucket we need to provide our user's http credentials to the AWS via the terminal. For that, run the following commands.

cd CICDRepo/

git config --global credential.helper '!aws codecommit credential-helper $@'

git config --global credential.UseHttpPath true



Now, we need to specify the user who is going to commit the changes in our CodeCommit repository.

For that, we need to execute the following commands to specify the user.

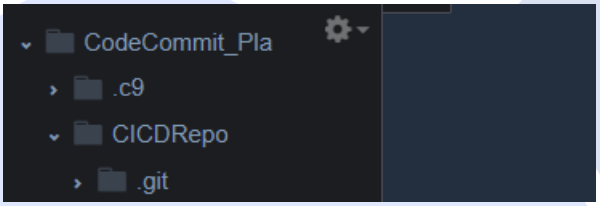
git config --global user.name "<Your Name>"

git config --global user.email <your.email@email.com>

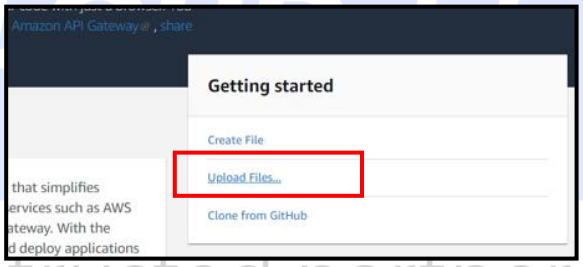
```
admin:~/environment $ cd CICDRepo/
admin:~/environment/CICDRepo (master) $ git config --global credential.helper '!aws codecommit credential-helper $@'
admin:~/environment/CICDRepo (master) $ git config --global credential.UseHttpPath true
admin:~/environment/CICDRepo (master) $ git config --global user.name "CloudPlusPlus"
admin:~/environment/CICDRepo (master) $ git config --global user.email admin@cloudplusplus.com
admin:~/environment/CICDRepo (master) $
```

Now, In the left navigation pane, where you can see the folder structure of your environment click on your repository name. For now, it is **CICDRepo.**

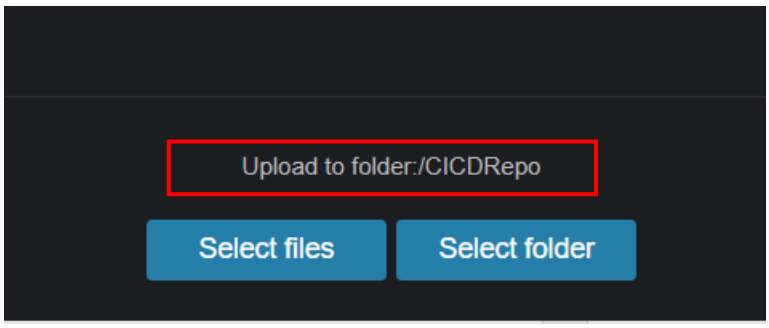
This will ensure that when we upload our Version 2 files, they will be uploaded in the folder of the repository.



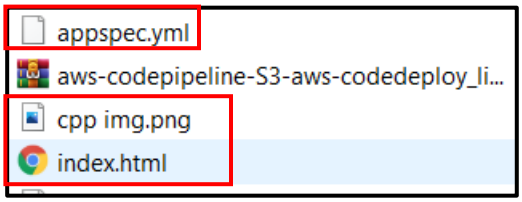
From the **Welcome** tab click on **Upload Files...**



When the **Upload Files** pop-up appears make sure the path specifies Upload to folder: /CICDRepo and click on **Select Files** to upload the version 2 files of our Web Application



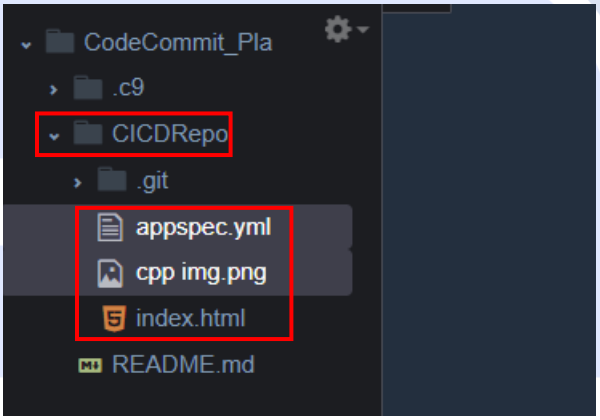
From the extracted CodeCommit folder, Go into the **Version 2** folder and select the following files to upload:



appspec.yml
index.html
cpp img.png

Click on Open and these files will be uploaded in our repository folder.

As you can see in the left folder structure pane, all our files are uploaded inside our CICDRepo folder



Run the following commands to push these files to our CodeCommit repository.

git add .

git commit -m "Version 2 of the Web Application"

git push

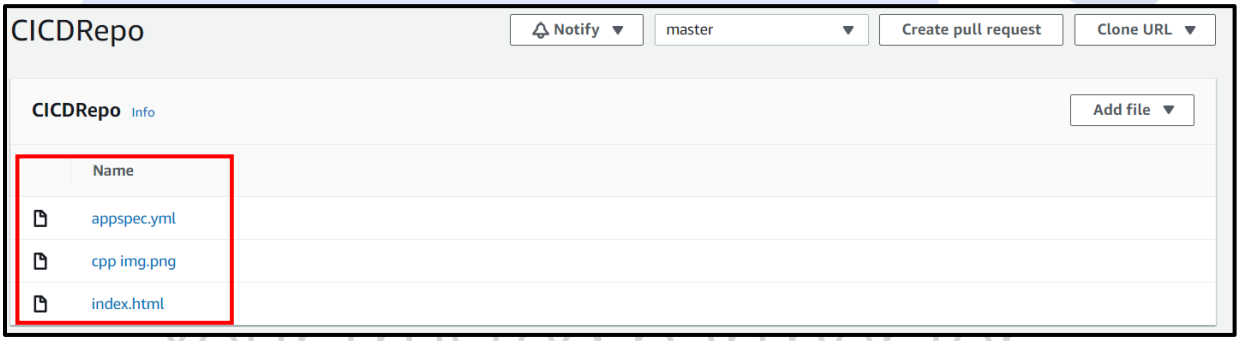
git add . : This command will add all the uploaded files to the staging area to commit the change.

git commit -m “Version 2 of the Web Application” : This command will commit our new changes with the message “Version 2 of the Web Application”.

git push : This command will push all these changes to the master branch of our CodeCommit repository.

```
admin:~/environment $ cd CICDRepo
admin:~/environment/CICDRepo (master) $ git add .
admin:~/environment/CICDRepo (master) $ git commit -m "Version 2 of the Web Application"
[master (root-commit) 0b1e981] Version 2 of the Web Application
 3 files changed, 61 insertions(+)
 create mode 100644 appspec.yml
 create mode 100644 cpp img.png
 create mode 100644 index.html
admin:~/environment/CICDRepo (master) $ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 736.90 KiB | 18.89 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/CICDRepo
 * [new branch]      master -> master
admin:~/environment/CICDRepo (master) $
```

Now, go back to your CodeCommit tab and refresh the page. And you'll be able to see the new Version 2 files are uploaded inside the CodeCommit repository



Step 4: Create a Code Pipeline

In the AWS Management Console, search for **Code Pipeline** and open it.

Click on Create pipeline and do as follows.

- **Pipeline Name: CodeCommitCICD**
- **Service Role: New Service role**

Keep the rest as default and click on next.

Give the **source provider** as **AWS CodeCommit**.

Choose the Repository name as CICDRepo and branch name as master.

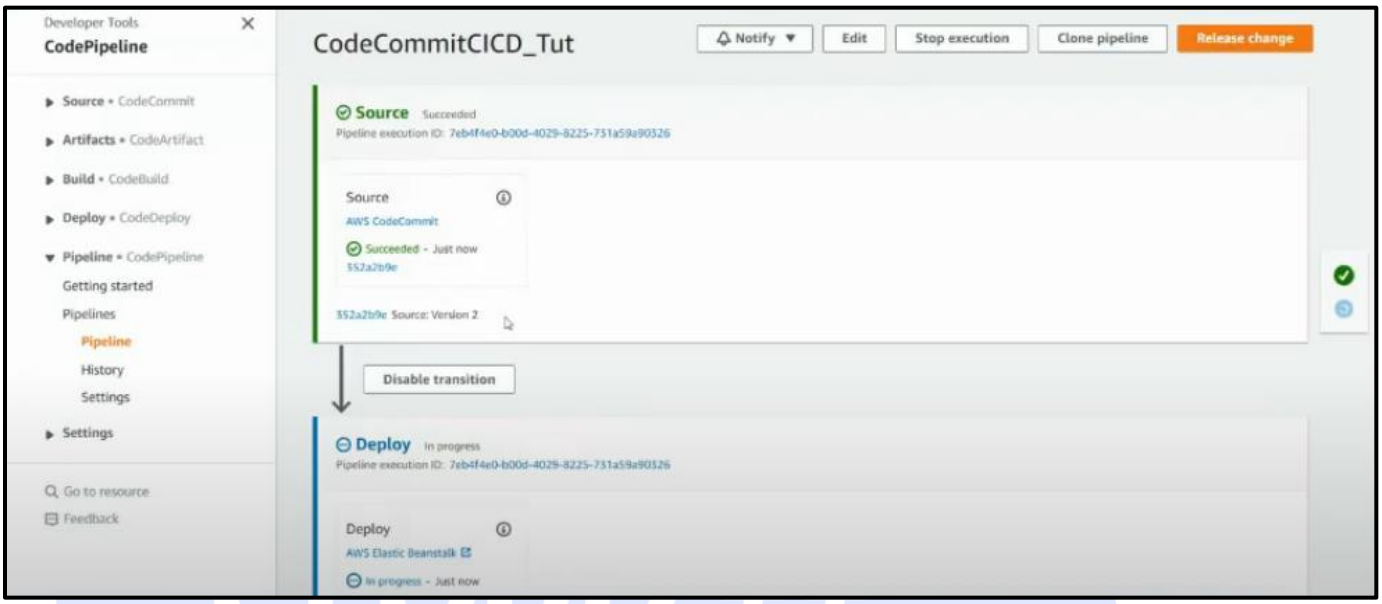
Keep the rest as default and click on Next.

Skip the build stage.

Under **deployment provider**, choose **AWS Elastic Beanstalk** and choose the correct application and environment name.

Click on Next.

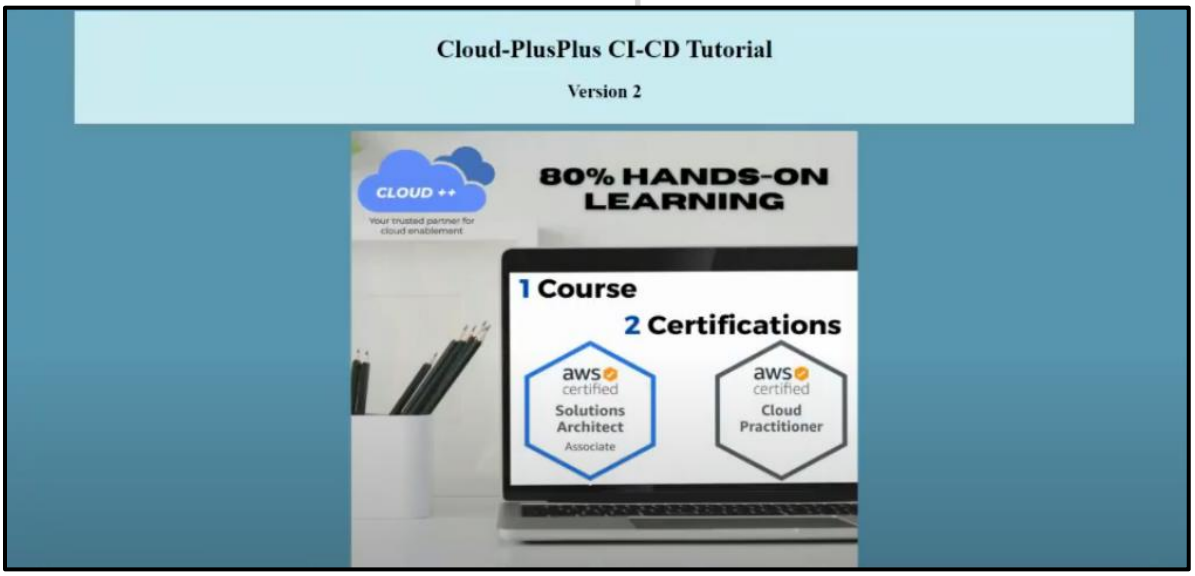
Confirm the settings and click on **Create Pipeline**.



Now wait for the Source and Deploy section to complete its action.

Once this is done, go back to the page where our application is hosted and refresh the page.

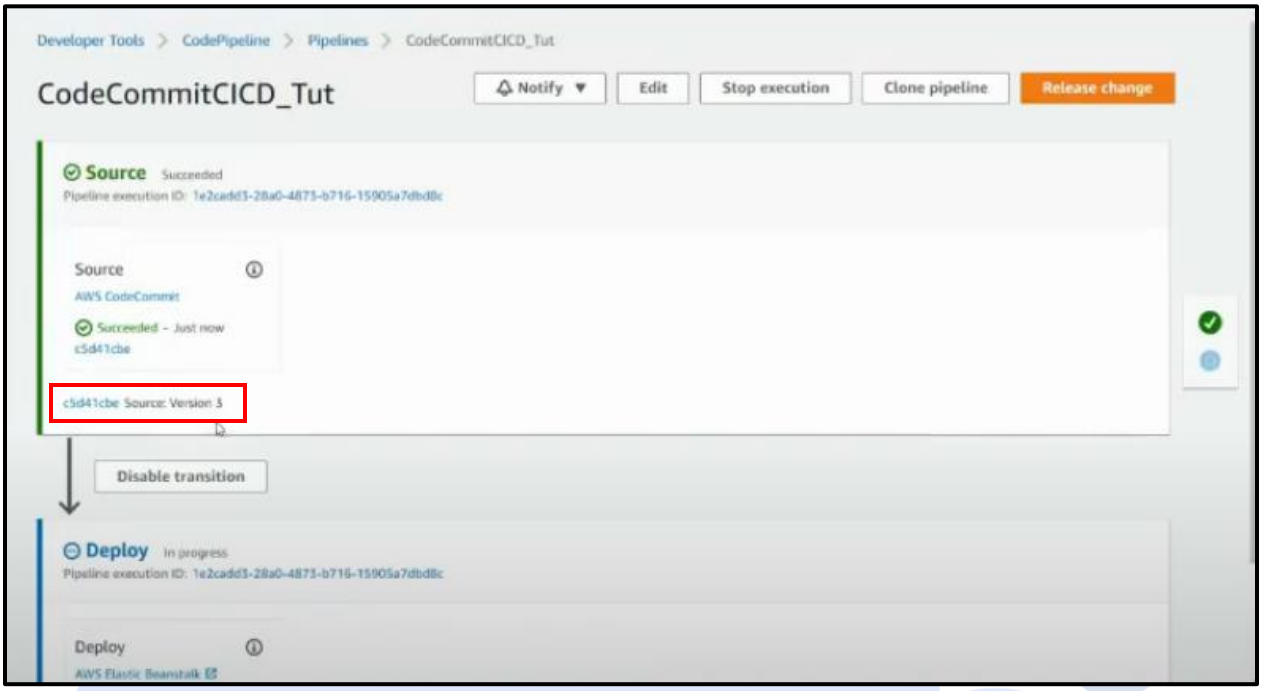
You will find that the Version-2 has been deployed.



www.cloud-plusplus.com/aws-training

Now to deploy Version 3, repeat Step 3.

Simultaneously, the CodePipeline section will be update automatically for Version 3.



Once this is done, go back to the page where our application is hosted and refresh the page.

You will find that the Version-3 has been deployed.

Note: If you no longer need the resources, delete code pipeline, codecommit repository as well as Elastic Beanstalk Application and Environment.

Document Created by	Version
Bavyaa R	23-November-2021
Shruti Dhongade	24-November-2021