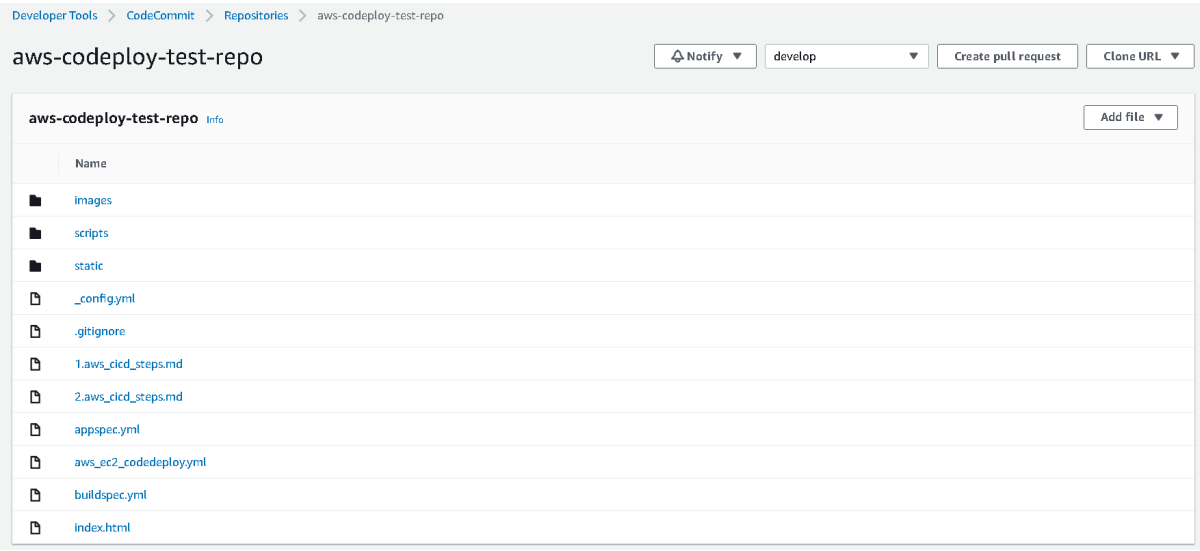
AWS CICD

Orchestrate pipeline using CodeCommit,Codebuild,CodeDeploy and CodePipeline

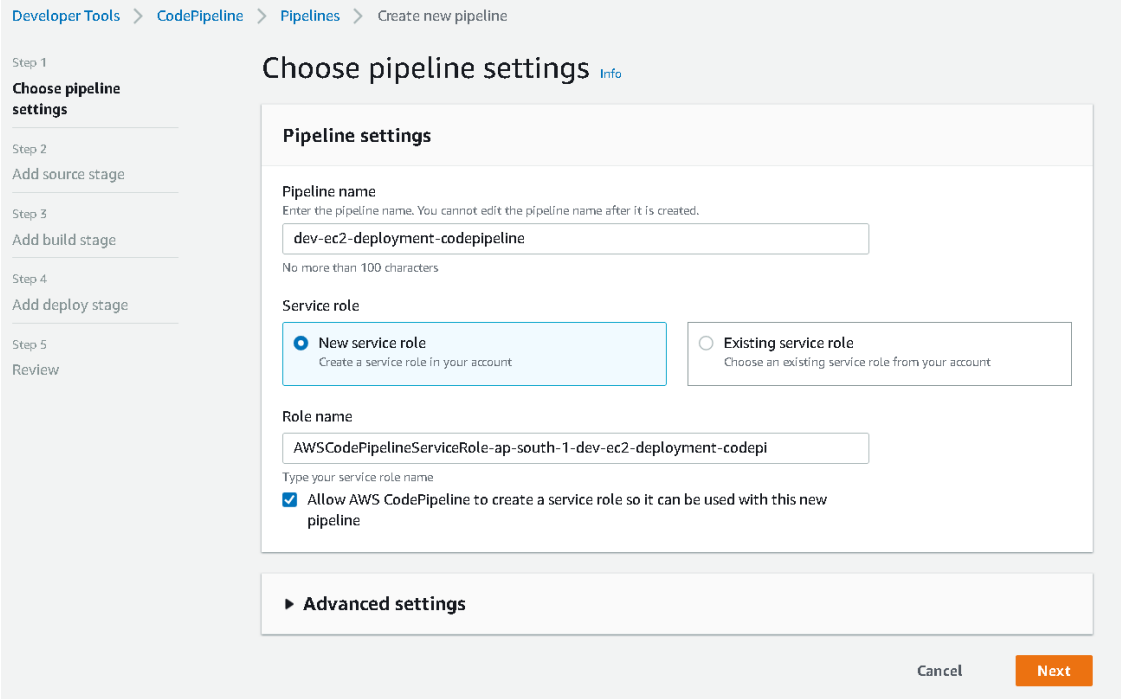
Creating CICD Resources (Continuous Deployment)

Creating Dev Deployment Pipeline

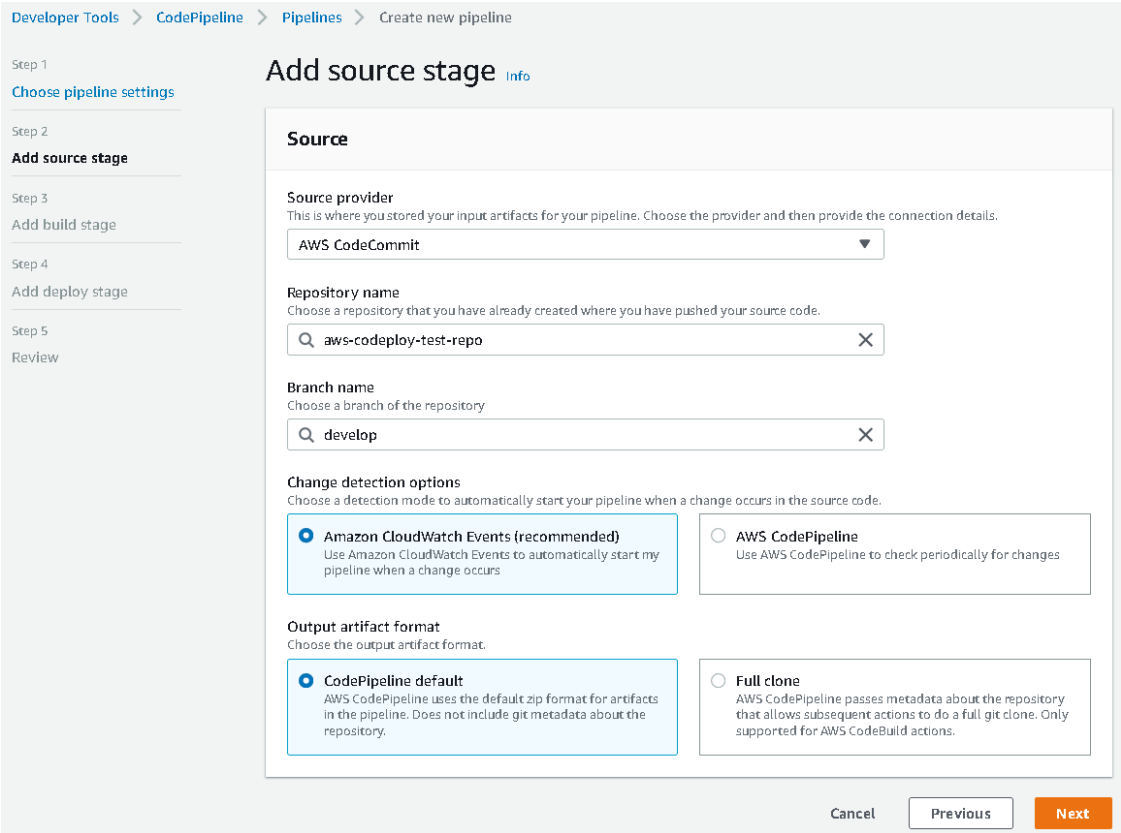
- Create below resources:
  - Configure CodeCommit IAM `https` credentials.
  - Download the Source Code from `https://aws-codedeploy-test-bucket.s3.ap-south-1.amazonaws.com/aws-codedeploy-repo.zip` and unzip the file and Create/Use exising CodeCommit Repo, push the source code files to Codecommit Repo into **master** branch.



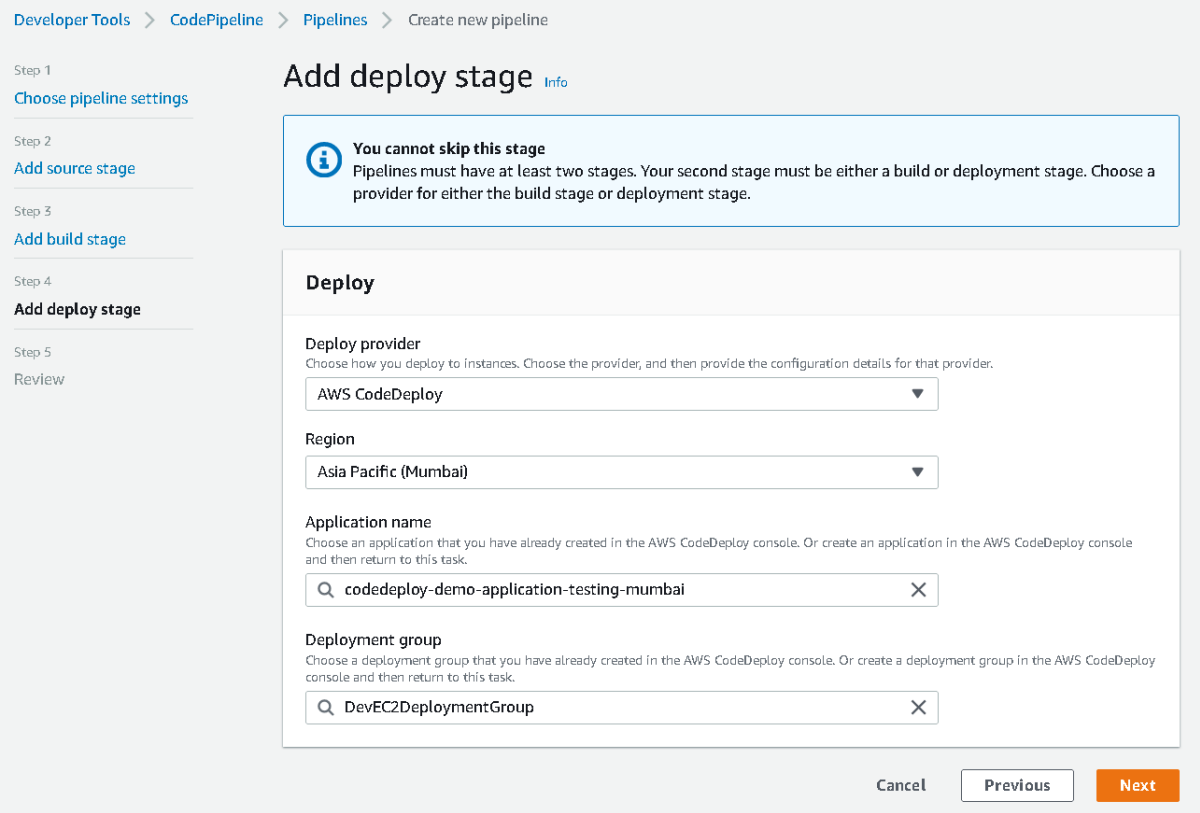
  - First for above setup, using CodeDeploy, we will first have our Code in CodeCommit.
  - CodeBuild Execuion and CodeDeploy Application is started manually, so CodePipeline trigger/start existing Build Project and CodeDeploy Application.
  - Create a new AWS CodePipeline:
    - Navigate to `AWS CodePipeline` > `Pipeline` and click `Create pipeline` > `dev-deployment-pipeline`
    - CodePipeline will create a `IAM Service Role` which will have access `CodeCommit Pull, CodeBuild Execution, start deploy access`



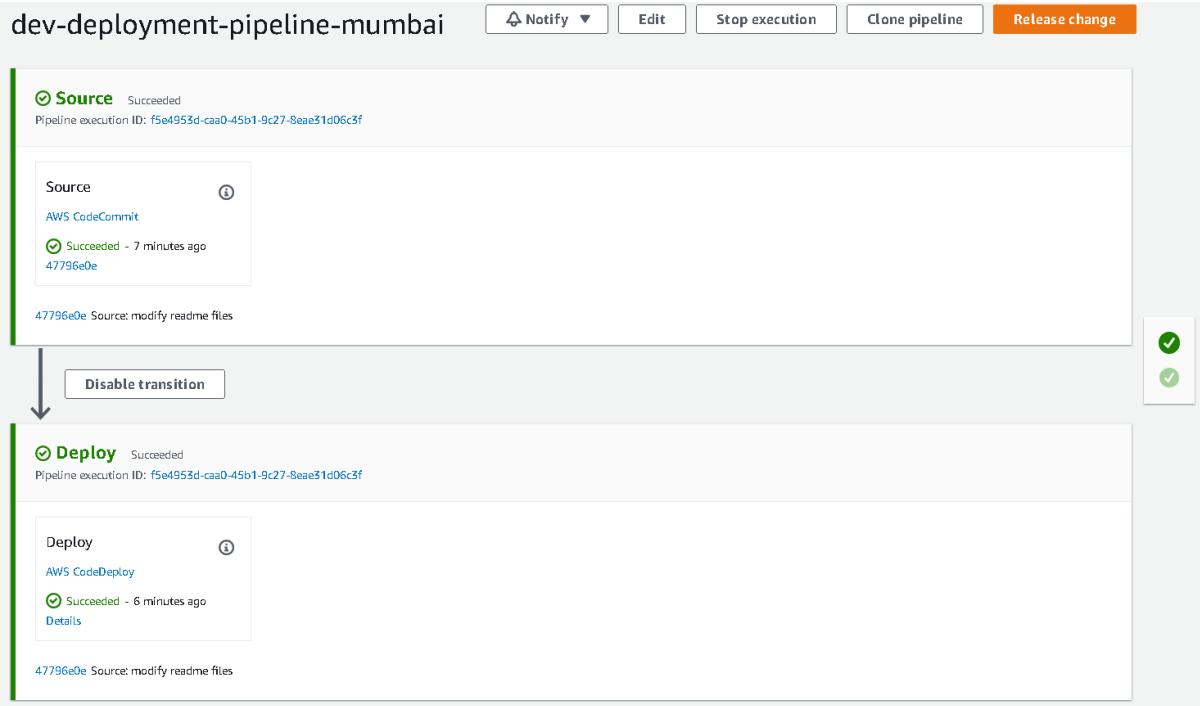
- Input the name for the service for CodePipeline and the S3 bucket that holds the artifacts for this pipeline.
- Add a source stage such as `CodeCommit repository` and `branch` on which code version you want to deploy.
- For `dev` pipeline, `develop` branch can be used.



- Skip the CodeBuild as of now.
- Add a deploy stage using `AWS CodeDeploy`. Select the CodeDeploy Application name that we have already created, and select the `DevEC2DeploymentGroup`



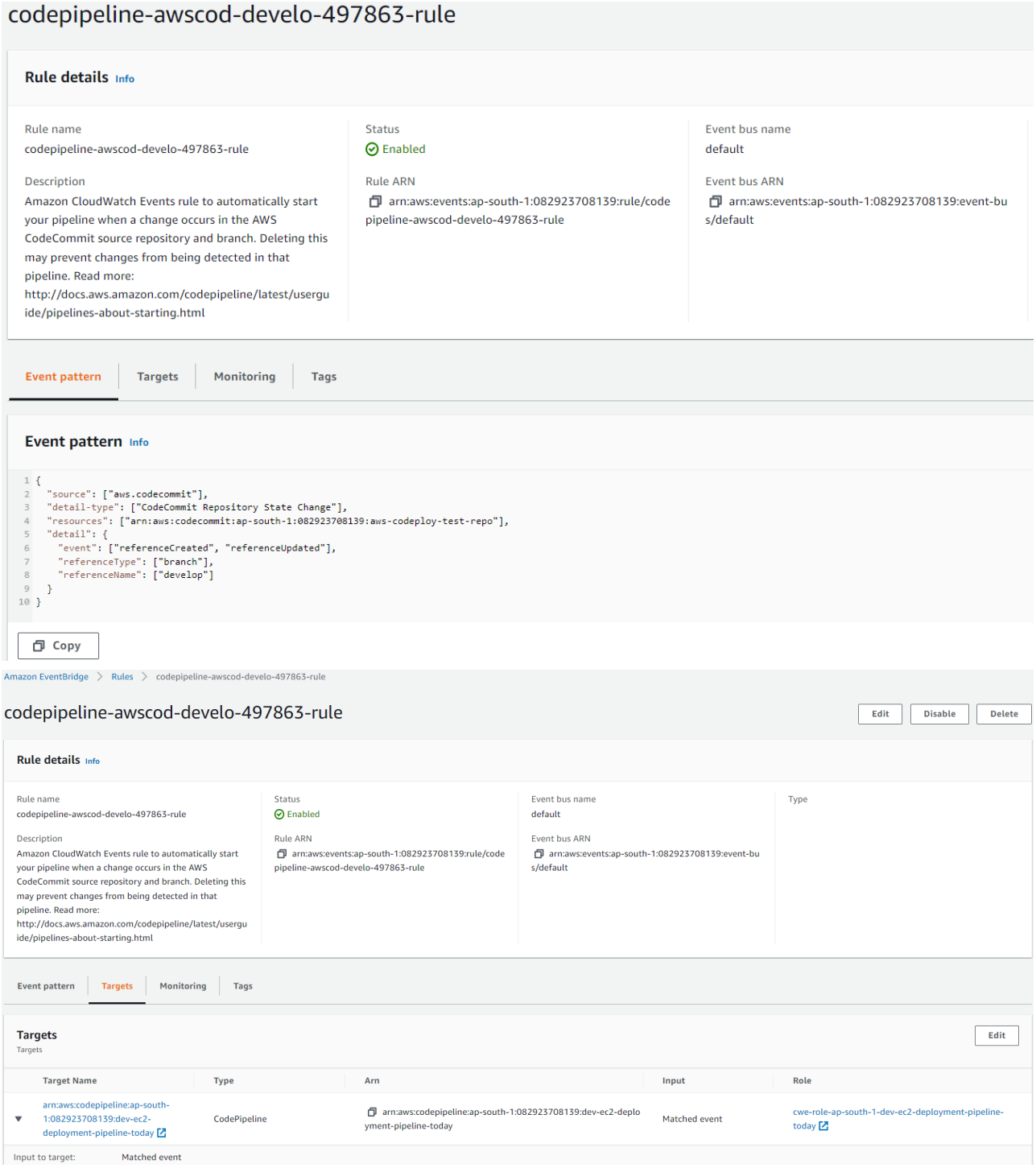
- Review the Stages and select `Create Pipeline`.
- First time when pipeline is created , it will run automatically.



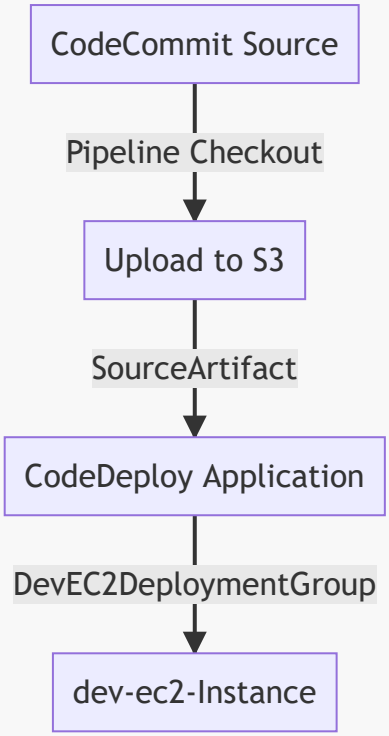
- Pipeline can be triggered manually by selecting the **Release Change**.
    - All Stages in a CodePipeline will be executed with new Execution ID
- A **Cloudwatch Event Rule** with similar name `codepipeline-awscod-develo-497863-rule` is created in the background that will trigger the CodePipeline automatically when there is any change in specified branch of CodeCommit Repo. - Navigate and view the CloudWatch Event.

- Event Pattern will be as below:

```
{
  "source": ["aws.codecommit"],
  "detail-type": ["CodeCommit Repository State Change"],
  "resources": ["arn:aws:codecommit:ap-south-
1:ACCOUNT_ID:CODECOMMIT_REPO"],
  "detail": {
    "event": ["referenceCreated", "referenceUpdated"],
    "referenceType": ["branch"],
    "referenceName": ["develop"]
  }
}
```

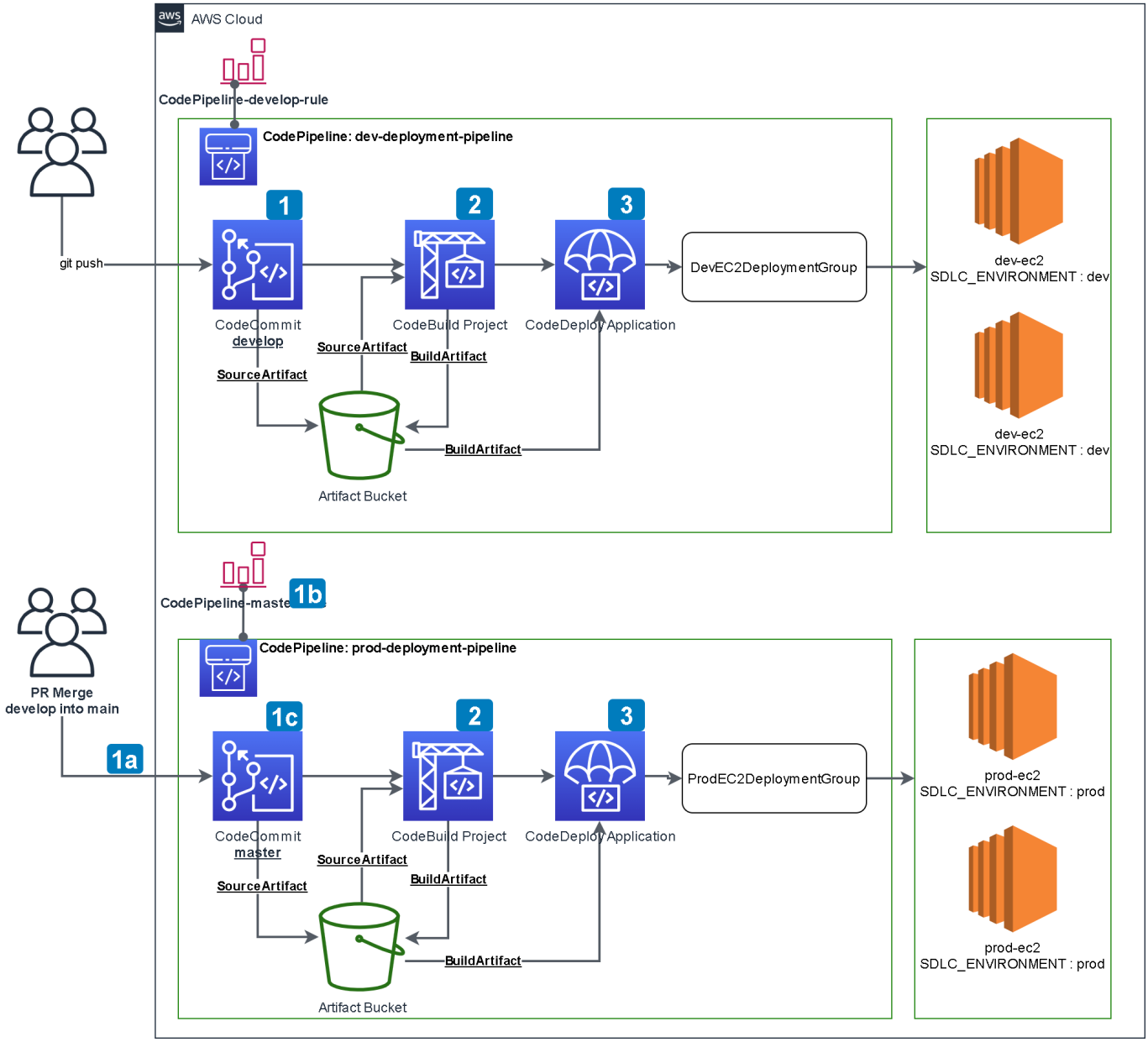


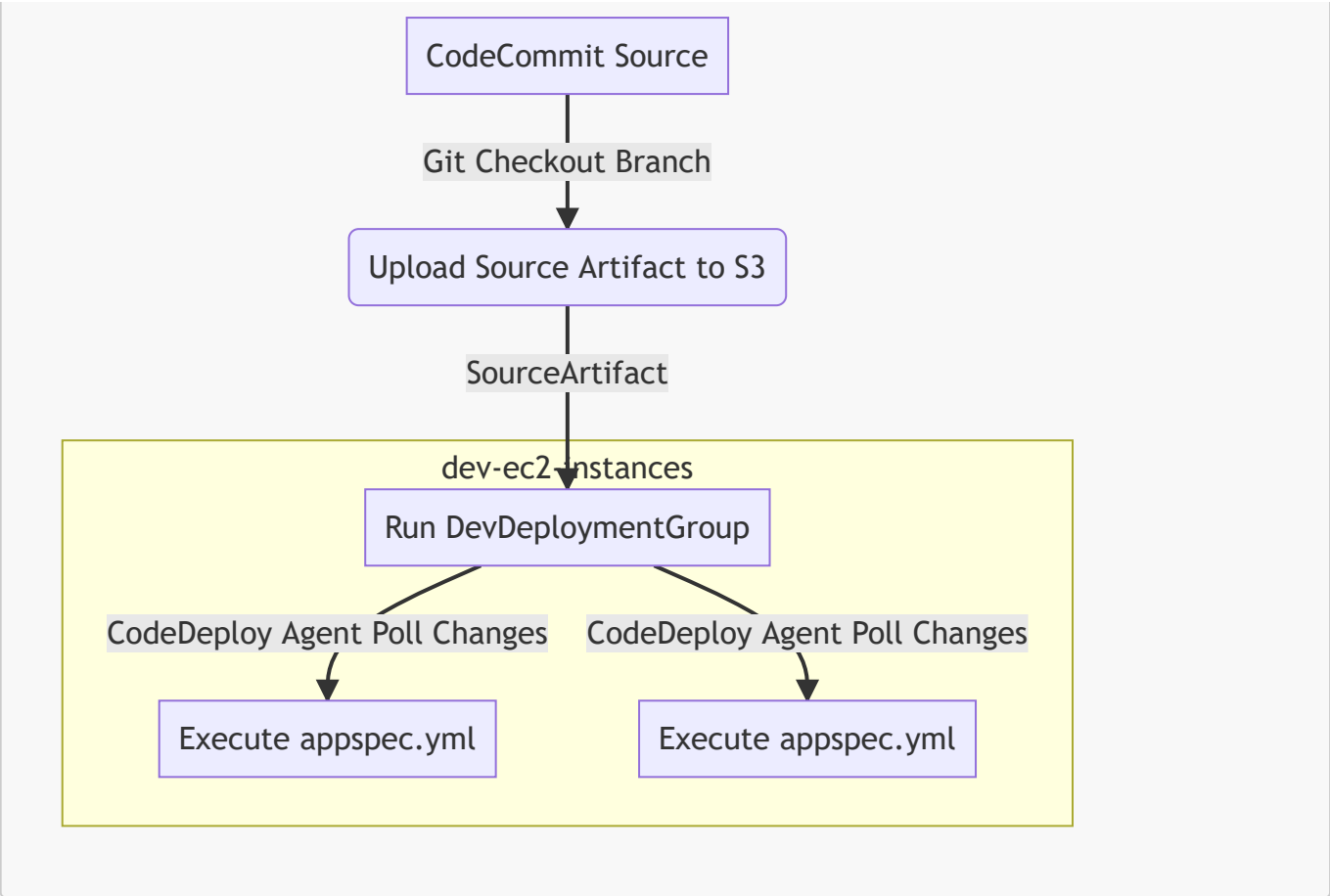
- If you make changes in the branch that is configured in cloudwatch event, the codepipeline will automatically start.


CodeCommit Source
Pipeline Checkout
Upload to S3
SourceArtifact
CodeDeploy Application
DevEC2DeploymentGroup
dev-ec2-Instance

AWS CodePipeline

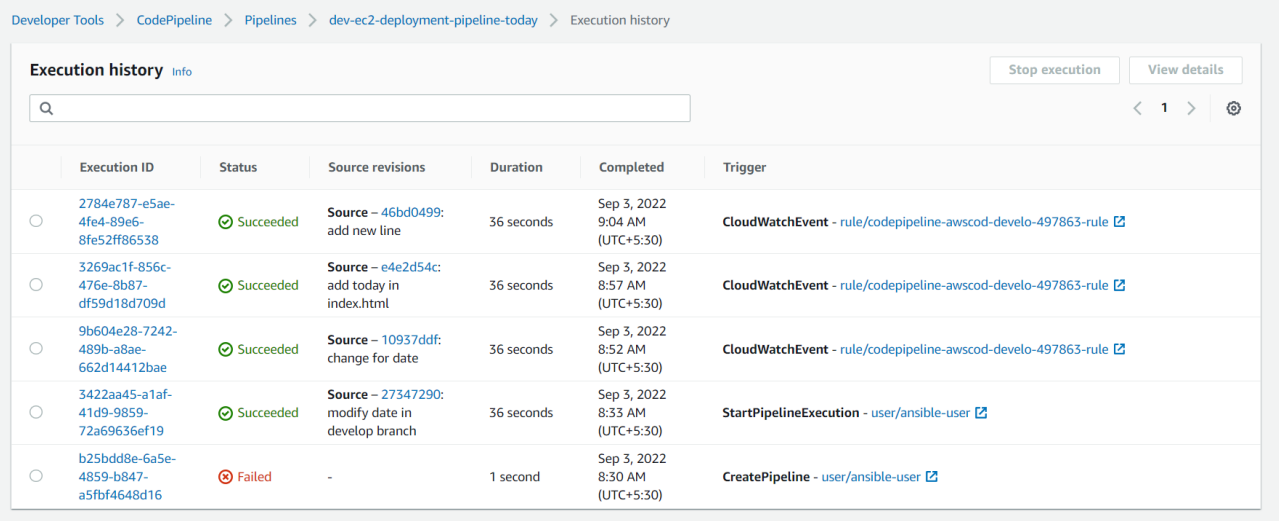
CodeCommit -> CodeBuild -> CodeDeploy





Testing the Pipeline Executions

- Change some content in the source code file in the branch configured above, and push the changes in the CodeCommit Repo.
- This will trigger the CodePipeline with (Codecommit and CodeDeploy stage).
- Above CodePipeline Project will directly deploy anything that is pushed to Repo in the specified branch that is configured in CloudWatch Event Rule.
- Once Pipeline is executed successfully for stages configured, validate all the steps from commits made in repo to application code that is deployed on WebServers.
- Validate the CodeDeploy Agent Execution Logs on any of the Webserver.
- CodePipeline Execution History will be maintained for each pipeline run as below.



Launching Dev and Prod EC2 instances

- Launch EC2 instances for Prod Environment using TF/CF/Manually.
- Launch more two EC2 instances with CodeDeploy Agent installation Script in the UserData during Launch.
- Make sure new instances are having Tags as `SDLC_ENVIRONMENT` : `prod`
- Go to the same CodeDeploy Application -> Create a new Deployment Group with name `ProdEC2DeploymentGroup` for the newly launched EC2 instances Tag Key:Value pair as `SDLC_ENVIRONMENT` : `prod` Tag.

Creating Prod Deployment Pipeline

- Create another Pipeline/Clone above pipeline, with similar above stages to configure the same for prod environment deployment i.e `prod-deployment-pipeline`.
- Configure the Source Stage for CodeCommit Repository with `main/master` branch.
- Configure the Deploy Stage with CodeDeploy Applicaton with `ProdEC2DeploymentGroup`.
- Raise a PR to merge changes from `develop` into `master/main`.
  - `Fast forward merge`: All source branch commits will be available in Destination branch.
  - `Squash and Merge`: All source branch commits will be squashed into a single commit and single commit will be merged into Destination branch.
- Once Changes are present in `master/main` branch, `prod-deployment-pipeline` should get started automatically due to CloudWatch Event Rule.

Reference

- Follow steps CodeDeploy Logs into Cloudwatch to add CodeDeploy EC2 Logs into CloudWatch.