CodePipeline definition over cloudformation

Workflow Modeling

A pipeline defines your release process workflow, and describes how a new code change progresses through your release process. A pipeline comprises a series of stages (e.g., build, test, and deploy), which act as logical divisions in your workflow. Each stage is made up of a sequence of actions, which are tasks such as building code or deploying to test environments. AWS CodePipeline provides you with a graphical user interface to create, configure, and manage your pipeline and its various stages and actions, allowing you to easily visualize and model your release process workflow.

So we understand the whole idea behind Codepipeline, and we want this part of our Infrastructure as a Code templates.

Lets go now to the Main architecture template file we already built, we want our pipeline to be on this file as a module. So under DatabaseCluster we add the resource Deployment Pipeline.

This one will be an AWS::CloudFormation::Stack type.

Properties. Here we will have the TemplateURL lets type in https://s3.amazonaws.com/cicdoveraws/deployment-pipeline.yaml .

Next the parameters we are going to need for this module. Because Codepipeline will need to change and deploy to different parts in the app we will need:

The ECS Cluster name:
The codeCommit repository:
The repository branch:
The LoadBalancer targetgroup:
The ECR, or the Continainer Repository:
The Stackname
and finally The template bucket

With this we finished the parameters needed for the Deployment Pipeline module, in the next lecture we are going to build the module itself with 3 stages on it to deliver automatically from source to production.