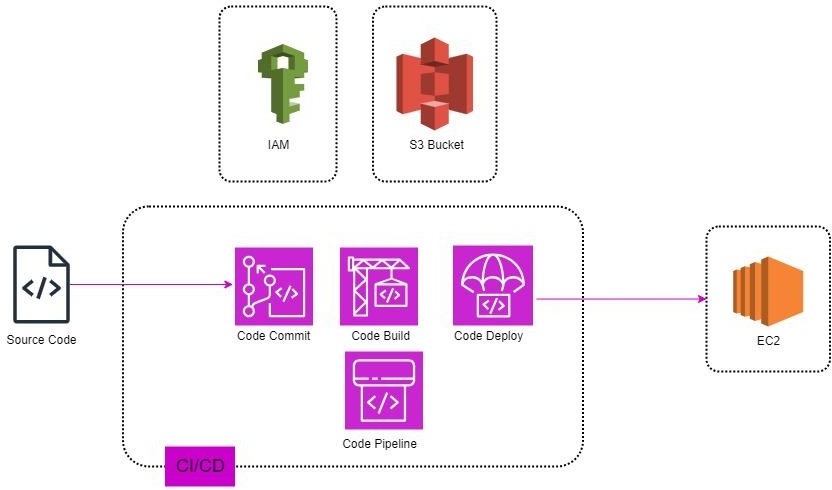
-AWS CI/CD pipeline deployment on EC2 instances



# What is a CI/CD pipeline?

Stating in its name itself ‘CICD’ is an abbreviation used for continuous integration and delivery of resources, where developers can take advantage of this pipeline to fully automate the process of committing the code to repositories to deploying it on resources where it can be hosted. The pipeline Includes various AWS services like CodeCommit, CodeBuild, CodeDeploy, and CodePipline.

# Steps for Deployment -

1. The CICD pipeline itself is a sophisticated form of technology but a real-time saver it generally consists of three CORE services, Starting from CodeCommit where developers usually connect their local machine with the help of code workspaces like VSCode to CodeCommit repositories by making use of Roles created by IAM in AWS.

We Commit all our code files including our static or dynamic web page file and configuration files like Buildspec.yml and appspec.yml into a master branch in the AWS code commit

Furthermore, these files get permanently stored in these repositories unless you delete them from the AWS console or with the help of Sdk’s.

1. Moving to the next and one of the main stages of the pipeline where the files are built in the AWS CodeBuild, as the name pronounces it, it is designed for the purpose of building the whole code into an S3 bucket, where we initially define the path to a folder in an S3 bucket and later the build takes places, throughout these operations, we again use IAM roles created for specific purposes to tighten the security of the architecture.
2. The last and final step of Deployment takes place in AWS CodeDeploy, where we configure the deployment of resources to an EC2 instance up to a maximum of 10 instances or a fleet of instances. The service also offers the integration of Auto-Scaling Groups(ASG) and Application load balancers(ALB) or Classic Load Balancers(CLB).

The load balancers and auto-scaling groups are created for the purpose of distributing traffic to a group of instances and attaching a track policy for scaling in & out during maximum and minimum hours of traffic on a website.

Now the CodeDeploy agent comes in place where it is specifically used for the purpose of connecting your CodeDeploy service to an EC2 instance, we have the choice of either manually setting up the CodeDeploy agent onto the EC2 instance or instead using the codeDeploy agent through AWS SystemManager Service.

Once the Build is deployed through the CodeDeploy service, it gets hosted on the ec2 instance and can be accessed through the Public IPv4 DNS Address.

1. Counting the last AWS service of the CICD architecture, also known as CodePipline, the most important and by far the core structure of the whole pipeline. The CICD itself was created to serve the purpose of Automation from committing the code to the pipeline to seeing its automatic deployment on EC2 instances.

Once a pipeline gets established in the CI/CD Architecture. Congrats !! Now your pipeline will be automatically triggered and Ingest the new Changes in real-time and will automatically build the code in the Specified S3 bucket. Leading to its deployment on the EC2 instances.