

AWS Certified DevOps Engineer Training - Project

You are now a part of the DevOps team in Logicworks company, which is a leading AWS consulting and Managed Services Provider with 25 years of experience in enterprise IT. Your first project is to replicate the current environment on AWS as per the following requirements:

i. **Adoption of IAC (Infrastructure as Code) to facilitate faster replication of environments for different customers who may have a need for similar environments.**

Infrastructure as Code provides the ability to turn complex systems and environments into a few lines of code, which can be deployed at the click of a button. This enables automated dev/test pipelines which provide a rapid feedback loop for developers, and rapid deployment of new features for end-users.

IaC tools make it possible to define a software stack using a simple text-based configuration file, and launch that stack automatically, ensuring it runs exactly the same each time and in any environment. Configuration files are stored in version control, meaning that an entire system, from the operating system to application configuration, can be treated as code. It can be tweaked, tested, and deployed at will whenever required.

Using **AWS CloudFormation** tool to enable teams to define templates representing software stacks and automatically deploy them to environments. CloudFormation lets you automatically spin up anything from a single EC2 machine to a complex application using multiple AWS services.

ii. **Infrastructure must include multi-region architecture and needs to run in two different regions for High Availability and Disaster Recovery purposes. (Hint: ensure proper networking system)**

1. Using **Router 53** - All user traffic will go via Route 53, which will make routing decisions regarding which region the users requests should be served from and configure the below policies accordingly: -
 - **Failover routing policy**
 - **Geolocation routing policy**
 - **Geoproximity routing policy**
 - **Latency routing policy**
 - **Multivalue answer routing policy**
2. Services should be stateless, and state should be shared between regions.
3. **CloudFront** for Global Content Delivery Network (CDN)

- iii. The applications should be run in a container to prevent errors and inconsistencies. The container images are supposed to be stored in a docker image repository. Additionally, a container management system has to be implemented to manage growth in the number of containers in the future

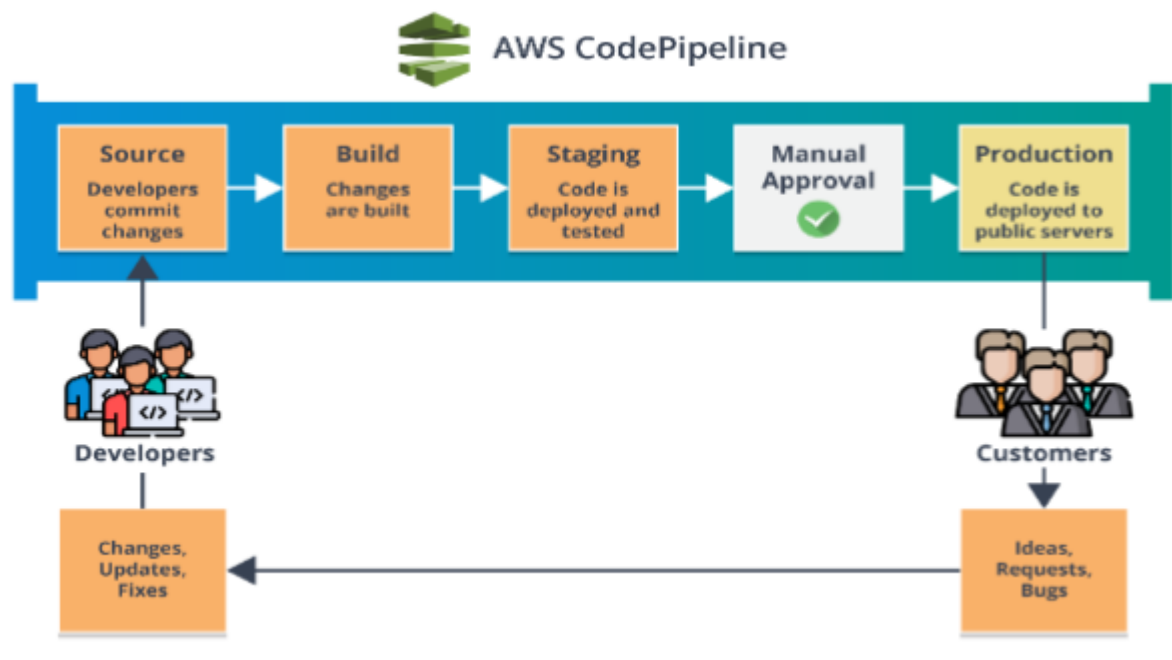
Using:-

1. **AWS Elastic Container Service (ECS)** - Container orchestration service that supports Docker containers and allows you to easily run and scale containerized applications on AWS.
2. **AWS Elastic Container Registry (ECR)** - Managed Docker container registry that will make it easy to store, manage, and deploy Docker container images
3. **AWS Fargate** – to compute engine for Amazon ECS that allows you to run containers without having to manage servers or clusters

- iv. Manual CI/CD pipeline deployments did not fulfill the requirements. Hence there is a need to automate the CI/CD pipeline. The CI/CD pipeline should include source, build, stage, manual approval and production environment.

Using:-

AWS CodePipeline: to fully manage CI/CD service to automate release pipelines for fast and reliable application and infrastructure updates. CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. You can easily integrate AWS CodePipeline with third-party services such as GitHub or with your own custom plugin.



- v. **The application will be running in two different regions, so the source code repository needs to be replicated in the other region as well, in order to pull the code locally within the regions and avoid any latency issues.**

Using:- AWS Lambda and AWS Fargate to replicating an AWS CodeCommit repository from one AWS region to another AWS region.

- vi. **The infrastructure and the application need to be under continuous monitoring with notification mechanism.**

Using:- AWS CloudWatch

CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

- vii. **It is necessary to have an approved CI/CD pipeline before the code gets pushed to production.**

Using:- AWS CodePipeline