DEVOPS ASSIGNMENT 1

By: Nithyashree Arunachalam(4NI19IS058)
Section - A

Introduction

DevOps is an amalgamation of the terms development and operations which represents a shared approach to the tasks performed by a company's application development and IT operations teams.DevOps vouches for better communication and collaboration among teams and others in the organisation. It describes the adoption of iterative software development, automation and programmable infrastructure deployment and maintenance. DevOpshas the ability to change the software delivery chain, services, jobs,IT tools and best practices.

Its not a technology but an environment and includes the following:

1. Continuous integration and continuous delivery or continuous deployment (CI/CD) tools,with an emphasis on task automation.
2. Systems and tools that support DevOps adoption, include real-time monitoring and collaboration platform.
3. Cloud computing

A Devops approach is one of the many techniques IT staff use to execute IT projects that meet business needs. It generally coexists with Agile Software development process. One can visualise a DevOps process as an infinite loop,compromisingthesestep:plan,code,build,test,release,deploy,operate,monitor and through feedback –plan which resets the loop.

What is CI/CD?

CI/CD essentially stands for Continuous Integration (CI) and Continuous Delivery (CD) or Continuous Deployment (CD). It is the method which creates a faster and more accurate way of combining the work of different people into one combined product. Within DevOps, CI/CD is a set of practices or tools that streamlines the product development process. The ultimate goal is to deliver quality products.

- CI(Continuous Integration): always refers to continuous integration, which is an automation process for developers. Successful CI means new code changes to an app are regularly built, tested, and merged to a shared repository. It's resolves the issues that might show up due to having too many branches of an app in development at once that might conflict with each other.

- CD(Continuous Delivery): continuous delivery and/or continuous deployment, which are related concepts that sometimes get used interchangeably. Both are about automating further stages of the pipeline, but they're sometimes used separately to illustrate just how much automation is happening.Continuous delivery usually means a developer's changes to an application are automatically bug tested and uploaded to a repository (like GitHub or a container registry), where they can then be deployed to a live production environment by the operations team.

- Continuous deployment (the other possible "CD") can refer to automatically releasing a developer's changes from the repository to production, where it is usable by customers. It addresses the problem of overloading operations teams with manual processes that slow down app delivery.

A pipeline is a process that drives software development through a path of building, testing, and deploying code, also known as CI/CD. The objective is to minimize human error and maintain a consistent process for how software is released. Tools that are included in the pipeline could include compiling code, unit tests, code analysis, security, and binaries creation.

Stages :

1.Build : The build stage involves the code being written. This is typically done by multiple people in a team, and for larger projects, multiple teams. Code is held in a version control system (VCS).

2. Test: testing code leads to greater confidence that the code will perform as expected. Testing of your code can be automated. In general, this is normally a repetitive, complex and sometimes tedious process to perform manually.

3.Deliver: After the code has been tested, the code is packaged up as an artifact and committed to a repository.

4.Deploy: The deploy stage allows the orchestration of the artifact release. Usually, teams will deploy to multiple environments, including environments for internal use such as development and staging, and Production for end-user consumption.

Benefits of CI/CD Pipeline:

1. Higher Efficiency
2. Reduced risks of defects
3. Faster product delivery
4. Log generation
5. Better Planning

What is Feature Flags?

A feature flag is a software development process used to enable or disable functionality remotely without deploying code.New features can be deployed witout making them visible to users.Feature flags help decouple deployment from release letteing you manage the full lifecycle of a feature.

Feature flagging allows companies to continuously deliver and deploy software to their users in a faster way. The use of feature flags allows companies to perform regular feature rollouts, fix bugs in the code without redeploying, experience a more streamlined development cycle and do rollbacks of code more easily.

Why do we need Feature flags?

Feature flagging enhances CI/CD by making continuous more achievable. The feature flags and feature toggles are a key component of the implementation of continuous delivery, allowing the separation of feature rollout from code deployment. Incomplete features can be merged into the production codebase but hidden behind feature flags.

References

1. **https://www.redhat.com/en/topics/devops/what-cicd-pipeline**
2. **https://www.guru99.com/ci-cd-pipeline.html**
3. **https://launchdarkly.com/blog/what-are-feature-flags/**
4. **https://www.atlassian.com/continuous-delivery/principles/feature-flags**