DevOps Certification Training

Study Material – DevOps Overview

Waterfall Model :

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Explain Agile Methodology :

Agile software development is an iterative approach to creating software products based on quickly releasing a minimum viable product (MVP) and then adjusting it and adding features and functionalities in stages based on user behaviour and feedback. The methodology is designed to address the fact it can be difficult to accurately predict the most intuitive user journeys, features and functionalities users need, prefer and desire from software.

As a methodology, agile software development stands in contrast to the once dominant Waterfall approach. When building software to the Waterfall methodology, software development teams create highly detailed specifications and functionality requirements upfront. The software is then built to that blueprint and released as a ‘completed’ product.

Overview of DevOps :

The word "DevOps" is a mashup of "development’ and "operations" but it represents a set of ideas and practices much larger than those two terms alone, or together. DevOps includes security, collaborative ways of working together, Automation and much more. DevOps is an approach to culture, automation, and platform design intended to deliver increased business value and responsiveness through rapid, high-quality service delivery.

Features of DevOps :

1) Automation

Automation helps to reduce time during the testing and deployment phase. With automation, productivity increases, and releases are made faster. This helps in catching bugs quickly so that they can be fixed easily. For continuous delivery, each code drop is defined through automated tests, cloud-based services, and builds. The builds are deployed in production using automation, to reduce human errors caused due to manual deployment.

2) Collaboration

DevOps is a collaboration of the Development and Operations team. It improves the working model of the teams and they become more productive with their productivity, which strengthens accountability and ownership. The teams work in close collaboration sharing responsibilities, which in turn makes the deployment to production faster.

3) Integration

Software Applications need to be integrated with other components in the environment. The integration phase is where the existing code is combined/merged with new features/functionality and then tested. Continuous integration and testing enable continuous development. A significant operational challenge is faced in how frequently the releases are to be made. To cater to such problems, continuous integration and delivery are implemented. It ensures quicker, safer, and reliable deliveries.

4) Configuration management

Configuration management helps in building robust and stable systems for the engineering teams using tools that can automatically manage and monitor updates to the configuration data. The configuration file can be written during deployment(for the automated environment), or it can be loaded at the run time(manually), depending on the environment in which it is running.

Benefits of DevOps ?

- Faster, better product delivery.
- Faster issue resolution and reduced complexity.
- Greater scalability and availability.
- More stable operating environments.

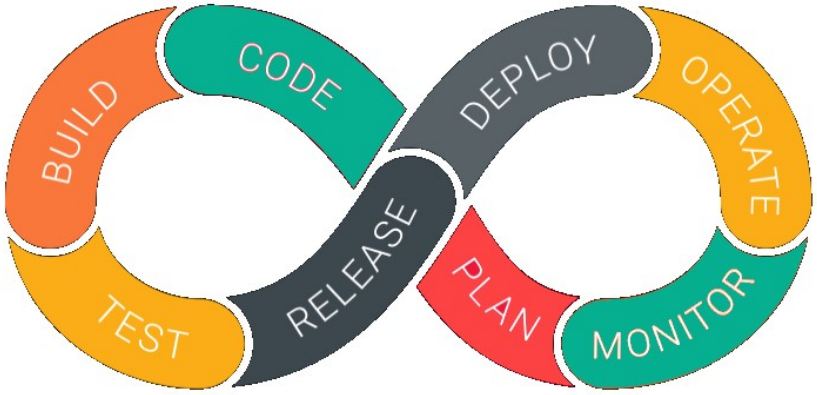
- Better resource utilization.
- Greater automation.
- Greater visibility into system outcomes.
- Greater innovation.

Why DevOps :

DevOps is important because it's a software development and operations approach that enables faster development of new products and easier maintenance of existing deployments.

DevOps Stages :

The 8 devops stages includes Planning, Coding, Building, Testing, Releasing, Deploying, Operating and Monitoring.



What is meant by build automation?

Build automation is the process of automating the retrieval of source code, compiling it into binary code, executing automated tests, and publishing it into a shared, centralized repository. Build automation is critical to successful DevOps processes.

What is meant by test automation ?

Test automation is the process of using automation tools to maintain test data, execute tests, and analyze test results to improve software quality. Automated testing is also

called test automation or automated QA testing. When executed well, it relieves much of the manual requirements of the testing lifecycle.

What is meant by continuous integration?

Continuous integration refers to the build and unit testing stages of the software release process. Every revision that is committed triggers an automated build and test. With continuous delivery, code changes are automatically built, tested, and prepared for a release to production.

What is meant by continuous delivery in DevOps?

Continuous delivery (CD) is the process of automating build, test, configuration, and deployment from a build to a production environment. A release pipeline can create multiple testing or staging environments to automate infrastructure creation and deploy new builds.

What is meant by continuous monitoring ?

Continuous monitoring is the process and technology used to detect compliance and risk issues associated with an organization's financial and operational environment. The financial and operational environment consists of people, processes, and systems working together to support efficient and effective operations.

What is CICD Pipeline ?

A CI/CD pipeline automates your software delivery process. The pipeline builds code, runs tests (CI), and safely deploys a new version of the application (CD). Automated pipelines remove manual errors, provide standardized feedback loops to developers, and enable fast product iterations.