**JAVA DEVOPS / CICD**

The DevOps is the combination of two words, one is Development and other is Operations. It is a culture to promote the development and operation process collectively.

The DevOps tutorial will help you to learn DevOps basics and provide depth knowledge of various DevOps tools such as Git, Ansible, Docker, Puppet, Jenkins, Chef, Nagios, and Kubernetes.

**What is DevOps?**

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to testing, deployment, and operations. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

DevOps helps to increase organization speed to deliver applications and services. It also allows organizations to serve their customers better and compete more strongly in the market.

DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.

DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, quality, and speed of the application delivery has improved to a great extent.

DevOps is nothing but a practice or methodology of making "Developers" and "Operations" folks work together. DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices in the context of a system-oriented approach.

DevOps is all about the integration of the operations and development process. Organizations that have adopted DevOps noticed a 22% improvement in software quality and a 17% improvement in application deployment frequency and achieve a 22% hike in customer satisfaction. 19% of revenue hikes as a result of the successful DevOps implementation.

Why DevOps?

Before going further, we need to understand why we need the DevOps over the other methods.

The operation and development team worked in complete isolation.

After the design-build, the testing and deployment are performed respectively. That's why they consumed more time than actual build cycles.

Without the use of DevOps, the team members are spending a large amount of time on designing, testing, and deploying instead of building the project.

Manual code deployment leads to human errors in production.

Coding and operation teams have their separate timelines and are not in synch, causing further delays.

**DevOps History**

In 2009, the first conference named DevOpsdays was held in Ghent Belgium. Belgian consultant and Patrick Debois founded the conference.

In 2012, the state of DevOps report was launched and conceived by Alanna Brown at Puppet.

In 2014, the annual State of DevOps report was published by Nicole Forsgren, Jez Humble, Gene Kim, and others. They found DevOps adoption was accelerating in 2014 also.

In 2015, Nicole Forsgren, Gene Kim, and Jez Humble founded DORA (DevOps Research and Assignment).

In 2017, Nicole Forsgren, Gene Kim, and Jez Humble published "Accelerate: Building and Scaling High Performing Technology Organizations".

DevOps Architecture Features

Here are some key features of DevOps architecture, such as:

DevOps Tutorial 4

1) Automation

Automation can reduce time consumption, especially during the testing and deployment phase. The productivity increases, and releases are made quicker by automation. This will lead in catching bugs quickly so that it can be fixed easily. For contiguous delivery, each code is defined through automated tests, cloud-based services, and builds. This promotes production using automated deploys.

2) Collaboration

The Development and Operations team collaborates as a DevOps team, which improves the cultural model as the teams become more productive with their productivity, which strengthens accountability and ownership. The teams share their responsibilities and work closely in sync, which in turn makes the deployment to production faster.

3) Integration

Applications need to be integrated with other components in the environment. The integration phase is where the existing code is combined with new functionality and then tested. Continuous integration and testing enable continuous development. The frequency in the releases and micro-services leads to significant operational challenges. To overcome such problems, continuous integration and delivery are implemented to deliver in a quicker, safer, and reliable manner.

4) Configuration management

It ensures the application to interact with only those resources that are concerned with the environment in which it runs. The configuration files are not created where the external configuration to the application is separated from the source code. The configuration file can be written during deployment, or they can be loaded at the run time, depending on the environment in which it is running.

DevOps Advantages and Disadvantages

Here are some advantages and disadvantages that DevOps can have for business, such as:

Advantages

DevOps is an excellent approach for quick development and deployment of applications.

It responds faster to the market changes to improve business growth.

DevOps escalate business profit by decreasing software delivery time and transportation costs.

DevOps clears the descriptive process, which gives clarity on product development and delivery.

It improves customer experience and satisfaction.

DevOps simplifies collaboration and places all tools in the cloud for customers to access.

DevOps means collective responsibility, which leads to better team engagement and productivity.

Disadvantages

DevOps professional or expert's developers are less available.

Developing with DevOps is so expensive.

Adopting new DevOps technology into the industries is hard to manage in short time.

Lack of DevOps knowledge can be a problem in the continuous integration of automation projects.

Prerequisite

To learn DevOps, you should have basic knowledge of Linux, and at least one Scripting language.

Audience

Our DevOps tutorial is designed to help beginners and professionals.

Problem

We assure you that you will not find any issue with this DevOps tutorial. But if there is any mistake or error, please post the error in the contact form.

Difference between CI/CD and DevOps

1. Continuous Integration/Continuous Delivery (CI/CD) :

CI/CD, as name suggests, is basically combined practices of CI and either CD method. It directly facilitates agile development because software change reaches production more frequently. It focuses in incorporation of automation tools that help to define software life-cycles.

2. DevOps :

DevOps, as name suggests, is basically combination of practices and tools especially designed to shorten systems development life cycle and increase organization’s ability to deliver applications and services quicker. It mainly emphasizes people and aim to improve collaboration between operations and development teams. It simply allows development team to achieve business requirements, maintain high code quality, etc.

**Difference between CI/CD and DevOps :**

CI/CD

It focuses more on software defined lifecycle highlighting tools that emphasize automation.

It provides set of activities carried out along whole software delivery process right from developing to final deployment.

It helps put together all code changes into single repository and run automated tests.

It helps development team to deliver frequent code that changes reliably.

Its main aim is to allow team to release constant flow of software updates into production quickly and get faster feedback from end user.

It generally increases speed of innovation and ability to compete in marketplace, provide additional business values for organizations, simple and quicker fault isolation, etc.

Its core operations include building phase, testing phase, deployment phase, and automated testing phase.

DevOps

. It focuses more on culture highlighting roles that emphasize responsiveness.

It provides set of ideal practices for quality software development.

It helps put together more streamlined, agile and efficient process of software production.

It helps organizations to develop software and their production teams in way that allows continuous rapid deployment.

Its main aim is to better combine roles of dev and ops to achieve same shared business goal.

It generally improves operational support and faster fixes, increase team flexibility and agility, cross-skilling and self-improvement, good processes across IT and team, etc.

Its core operations include application development, code development, code coverage, unit testing, packaging, deployment with infrastructure, configuration, orchestration, provisioning, and deployment.