**DEVOPS NOTES**

* **Introduction To Devops**

1. **What is Devops?**

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.

1. **History of Devops?**

* 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
* 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: Buildingand Scaling High Performing Technology Organizations".

1. **Why Do we need Devops?**

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.

1. **What are the Principles of Devops ?**

**Collaboration**

The key premise behind DevOps is collaboration. Development and operations teams coalesce into a functional team that communicates, shares feedback, and collaborates throughout the entire development and deployment cycle.

**Automation**

An essential practice of DevOps is to automate as much of the software development lifecycle as possible. This gives developers more time to write code and develop new features.

**Continuous Improvement**

Continuous improvement was established as a staple of agile practices, as well as lean manufacturing and Improvement Kata. It’s the practice of focusing on experimentation, minimizing waste, and optimizing for speed, cost, and ease of delivery.

**Customer-centric action**

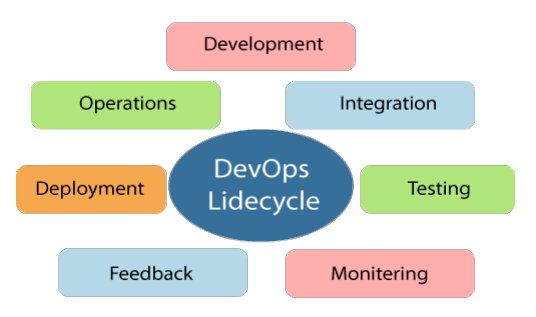
DevOps teams use short feedback loops with customers and end users to develop products and services centered around user needs. DevOps practices enable rapid collection and response to user feedback through use of real-time live monitoring and rapid deployment.

**Create with the end in mind**

This principle involves understanding the needs of customers and creating products or services that solve real problems.

1. **DevOps Lifecycle**

DevOps defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from beginning to the final stage of the product.



The DevOps lifecycle includes seven phases as given below:

1. **Continuous Development :**

This phase involves the planning and coding of the software. The vision of the project is decided during the planning phase. And the developers begin developing the code for the application. There are no DevOps tools that are required for planning, but there are several tools for maintaining the code.

1. **Continuous Integration:**

This stage is the heart of the entire DevOps lifecycle. It is a software development practice in which the developers require to commit changes to the source code more frequently. This may be on a daily or weekly basis. Then every commit is built, and this allows early detection of problems if they are present. Building code is not only involved compilation, but it also includes unit testing, integration testing, code review, and packaging.

The code supporting new functionality is continuously integrated with the existing code. Therefore, there is continuous development of software. The updated code needs to be integrated continuously and smoothly with the systems to reflect changes to the end-users.

Jenkins is a popular tool used in this phase. Whenever there is a change in the Git repository, then Jenkins fetches the updated code and prepares a build of that code, which is an executable file in the form of war or jar. Then this build is forwarded to the test server or the production server

1. **Continuous Testing:**

This phase, where the developed software is continuously testing for bugs. For constant testing, automation testing tools such as TestNG, JUnit, Selenium, etc are used. These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there is no flaw in the functionality. In this phase, Docker Containers can be used for simulating the test environment. Selenium does the automation testing, and TestNG generates the reports.

This entire testing phase can automate with the help of a Continuous Integration tool called Jenkins. Automation testing saves a lot of time and effort for executing the tests instead of doing this manually.

1. **Continuous Monitoring:**

Monitoring is a phase that involves all the operational factors of the entire DevOps process, where important information about the use of the software is recorded and carefully processed to find out trends and identify problem areas. Usually, the monitoring is integrated within the operational capabilities of the software application.

1. **Continuous Feedback:**

The application development is consistently improved by analyzing the results from the operations of the software. This is carried out by placing the critical phase of constant feedback between the operations and the development of the next version of the current software application.

The continuity is the essential factor in the DevOps as it removes the unnecessary steps which are required to take a software application from development, using it to find out its issues and then producing a better version.

1. **Continuous Deployment:**

In this phase, the code is deployed to the production servers. Also, it is essential to ensure that the code is correctly used on all the servers.

The new code is deployed continuously, and configuration management tools play an essential role in executing tasks frequently and quickly. Here are some popular tools which are used in this phase, such as Chef, Puppet, Ansible, and SaltStack.