## Deploying on Cloud

? What is Devops?

? DevOps Adoption Process

? DevOps VS Agile

? DevOps Building blocks

DevOps is the bridge between the development(devlopers and testers) and operations excellence

It is an extension of the lean and Agile principles, which streamlines and helps rapid deployments

conversation:

Hi, I am Mark

and I work in the

Development

team

Hi, My name is

Hi, I am Linda

and I work in the

Operations

team

Development and Operations in Conflict!!

developer:

I want change

to deploy new

things

ops:

I want stability

and do not want

interruptions in

the infrastructure

-----------------------------------------

There are two opposite

camps: Developers and

Operations

They have to work together

and make the transition

Definition:

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.

------------------------------------------

Applications of devops:

DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way.

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 Architecture Features

Automation,Collaboration,Integration,Configuration management

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 whore the existing code is combined with new functionality and then tested. Continuous integration and testing enable continuous development. To overcome such problems, continuous integration and delivery are implemented to deliver in a quicker, safer, and reliable manner.

4)) Configuration management

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.

---------------------------------------------------------------------------

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.

2) 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.

DevOps Lifecycle

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.

3) 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.

DevOps Lifecycle

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. Apart from that, report generation is a big plus. The task of evaluating the test cases that failed in a test suite gets simpler. Also, we can schedule the execution of the test cases at predefined times. After testing, the code is continuously integrated with the existing code.

4) 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.

It may occur in the form of documentation files or maybe produce large-scale data about the application parameters when it is in a continuous use position. The system errors such as server not reachable, low memory, etc are resolved in this phase. It maintains the security and availability of the service.

5) 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. It kills the efficiency that may be possible with the app and reduce the number of interested customers.

6) 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.

DevOps Lifecycle

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.

Containerization tools are also playing an essential role in the deployment phase. Vagrant and Docker are popular tools that are used for this purpose. These tools help to produce consistency across development, staging, testing, and production environment. They also help in scaling up and scaling down instances softly.

Containerization tools help to maintain consistency across the environments where the application is tested, developed, and deployed. There is no chance of errors or failure in the production environment as they package and replicate the same dependencies and packages used in the testing, development, and staging environment. It makes the application easy to run on different computers.

7) Continuous Operations

All DevOps operations are based on the continuity with complete automation of the release process and allow the organization to accelerate the overall time to market continuingly.

It is clear from the discussion that continuity is the critical factor in the DevOps in removing steps that often distract the development, take it longer to detect issues and produce a better version of the product after several months. With DevOps, we can make any software product more efficient and increase the overall count of interested customers in your product.

## Types of Deployment Patterns

<https://docs.microsoft.com/en-us/training/modules/introduction-to-deployment-patterns/>

<https://docs.microsoft.com/en-us/training/modules/introduction-to-deployment-patterns/1-introduction>

<https://docs.microsoft.com/en-us/training/modules/introduction-to-deployment-patterns/2-explore-microservices-architecture>

<https://docs.microsoft.com/en-us/training/modules/introduction-to-deployment-patterns/3-examine-classical-deployment-patterns>