Date: 20/07/2021

Batch: T11

Roll no.: 03

Experiment No.1

Aim: To understand DevOps Principles, Practices, and DevOps Engineer Role and Responsibilities

Theory: DevOps is a collaboration between Development and IT Operations to make software production and deployment in an automated & repeatable way. DevOps helps to increase the organization's speed to deliver software applications and services. The word 'DevOps' is a combination of two words, 'Development' and 'Operations.'

It allows organizations to serve their customers better and compete more strongly in the market. In simple words, DevOps can be defined as an alignment of development and IT operations with better communication and collaboration.

Need for DevOps:

- Before DevOps, the development and operation team worked in complete isolation.
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production
- Coding & operation teams have their separate timelines and are not in synch causing further delays.

DevOps Different from Traditional IT approach

Let's compare traditional software waterfall model with DevOps to understand the changes DevOps bring. We assume the application is scheduled to go live in 2 weeks and coding is 80% done. We assume the application is a fresh launch and the process of buying servers to ship the code has just begun

Old Process	DevOps
After placing an order for new servers, the Development team works on testing. The Operations team works on extensive paperwork as required in enterprises to deploy the infrastructure.	After placing an order for new servers Development and Operations team work together on the paperwork to set-up the new servers. This results in better visibility of infrastructure requirement.
Projection about failover, redundancy, data center locations, and storage requirements are skewed as no inputs are available from developers who have deep knowledge of the application.	Projection about failover, redundancy, disaster recovery, data center locations, and storage requirements are pretty accurate due to the inputs from the developers.
Operations team has no clue on the progress of the Development team. Operations team develop a monitoring plan as per their understanding.	In DevOps, the Operations team is completely aware of the progress the developers are making. Operations team interact with developers and jointly develop a monitoring plan that caters to the IT and business needs. They also use advance Application Performance Monitoring (APM) Tools
Before go-live, the load testing crashes the application. The release is delayed.	Before go-live, the load testing makes the application a bit slow. The development team quickly fixes the bottlenecks. The application is released on time.

DevOps Preferred

DevOps allows Agile Development Teams to implement Continuous Integration and Continuous Delivery. This helps them to launch products faster into the market.

Other Important reasons are:

1. Predictability: DevOps offers significantly lower failure rate of new releases

2. Reproducibility: Version everything so that earlier version can be restored anytime.

3. Maintainability: Effortless process of recovery in the event of a new release crashing or disabling the current system.

4. Time to market: DevOps reduces the time to market up to 50% through streamlined software delivery. This is particularly the case for digital and mobile applications.

5. Greater Quality: DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues.

6. Reduced Risk: DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle.

7. Resiliency: The Operational state of the software system is more stable, secure, and changes are auditable.

8. Cost Efficiency: DevOps offers cost efficiency in the software development process which is always an aspiration of IT companies' management.

9. Breaks larger code base into small pieces: DevOps is based on the agile programming method. Therefore, it allows breaking larger code bases into smaller and manageable chunks.

DevOps should be used for large distributed applications such as eCommerce sites or applications hosted on a cloud platform. It should not be used in a mission-critical application like bank, power and other sensitive data sites. Such applications need strict access controls on

the production environment, a detailed change management policy, access control policy to the data centres.

DevOps Life Cycle

DevOps is deep integration between development and operations. Understanding DevOps Concepts is not possible without knowing DevOps lifecycle.
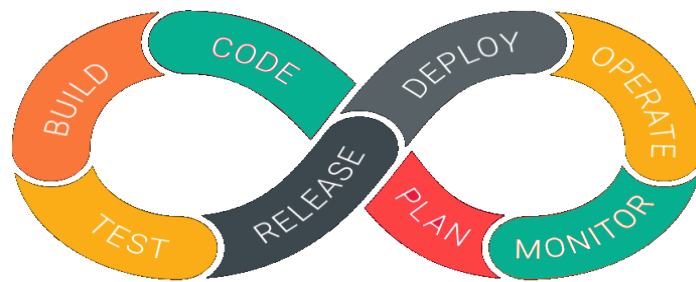
1. Development: In this DevOps stage the development of software takes place constantly. In this phase, the entire development process is separated into small development cycles. This benefits DevOps team to speed up software development and delivery process.

2. Testing: QA team use tools like Selenium to identify and fix bugs in the new piece of code.

3. Integration: In this stage, new functionality is integrated with the prevailing code, and testing takes place. Continuous development is only possible due to continuous integration and testing.

4. Deployment: In this phase, the deployment process takes place continuously. It is performed in such a manner that any changes made any time in the code, should not affect the functioning of high traffic website.

5. Monitoring: In this phase, operation team will take care of the inappropriate system behaviour or bugs which are found in production.

DevOps vs Agile

Stakeholders and communication chain a typical IT process

Agile addresses gaps in Customer and Developer communications

DevOps addresses gaps in Developer and IT Operations communications

Comparison Between DevOps and Agile

Agile	DevOps
Emphasize breaking down barriers between developers and management.	DevOps is about software deployment and operation teams.
Addresses gap between customer requirements and development teams.	Addresses the gap between development and Operation team
Focuses more on functional and non-functional readiness	It focuses operational and business readiness.
Agile development pertains mainly to the way development is thought out by the company.	DevOps emphases on deploying software in the most reliable and safest ways which aren't necessarily always the fastest.
Agile development puts a huge emphasis on training all team members to have varieties of similar and equal skills. So that, when	DevOps, likes to divide and conquer, spreading the skill set between the

something goes wrong, any team member can get assistance from any member in the absence of the team leader.	development and operation teams. It also maintains consistent communication.
Agile development manages on "sprints. It means that the time table is much shorter (less than a month) and several features are to be produced and released in that period.	DevOps strives for consolidated deadlines and benchmarks with major releases, rather than smaller and more frequent ones.

DevOps Principle

1. Customer-Centric Action: DevOps team must take customer-centric action for that they should constantly invest in products and services.

2. End-To-End Responsibility: The DevOps team need to provide performance support until they become end-of-life. This enhances the level of responsibility and the quality of the products engineered.

3. Continuous Improvement: DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the improvement of product or services offered.

4. Automate everything: Automation is a vital principle of DevOps process. This is not only for the software development but also for the entire infrastructure landscape.

5. Work as one team: In the DevOps culture role of the designer, developer, and tester are already defined. All they needed to do is work as one team with complete collaboration.

6. Monitor and test everything: It is very important for DevOps team to have a robust monitoring and testing procedures.

DevOps Engineer : Skill, Role and Responsibility

A DevOps Engineer is an IT professional who works with software developers, system operators, and other production IT staff to administer code releases. DevOps should have hard as well as soft skills to communicate and collaborate with development, testing, and operations teams.

THADOMAL SHAHANI ENGG COLLEGE
INFORMATION TECHNOLOGY DEPARTMENT

DevOps approach needs frequent, incremental changes to code versions, which means frequent deployment and testing regimens. Although DevOps engineers need to code occasionally from scratch, it is important that they should have the basics of software development languages.

A DevOps engineer will work with development team staff to tackle the coding and scripting needed to connect elements of code, like libraries or software development kits.

DevOps engineers work full-time. They are responsible for the production and ongoing maintenance of a software application's platform.

Following are some expected Roles, Responsibilities, and Skills that is expected from DevOps engineer:

- Able to perform system troubleshooting and problem-solving across platform and application domains.
- Manage project effectively through open, standards-based platforms
- Increase project visibility thought traceability
- Improve quality and reduce development cost with collaboration
- Analyse, design and evaluate automation scripts & systems
- Ensuring critical resolution of system issues by using the best cloud security solutions services
- DevOps engineer should have the soft skill of problem-solver and quick-learner

DevOps Automation Tools :

It is vital to automate all the testing processes and configure them to achieve speed and agility. This process is known as DevOps automation. The difficulty faced in large DevOps Team that maintain large huge IT infrastructure can be classified briefly into six different categories.

1. Infrastructure Automation
2. Configuration Management
3. Deployment Automation
4. Performance Management
5. Log Management
6. Monitoring.

DevOps Lab T.E SEM V (2021-2022)

Infrastructure Automation

Amazon Web Services (AWS): Being cloud service you do not need to be physically present in the data center. Also, they are easy to scale on-demand. There are no up-front hardware costs. It can be configured to provision more servers based on traffic automatically.

Configuration Management

Chef: It is a useful DevOps tool for achieving speed, scale, and consistency. It can be used to ease out complex tasks and perform configuration management. With this tool, DevOps team can avoid making changes across ten thousand servers. Instead, they need to make changes in one place which is automatically reflected in other servers.

Deployment Automation

Jenkins: This tool facilitates continuous integration and testing. It helps to integrate project changes more easily by quickly finding issues as soon as a built is deployed.

Log Management

Splunk: This is a tool solves the issues like aggregating, storing, and analyzing all logs in one place.

Performance Management

App Dynamic: It is DevOps tool which offers real-time performance monitoring. The data collected by this tool helps developers to debug when issues occur.

Monitoring

Nagios: It is also important to make sure people are notified when infrastructure and related services go down. Nagios is one such tool for this purpose which helps DevOps teams to find and correct problems.
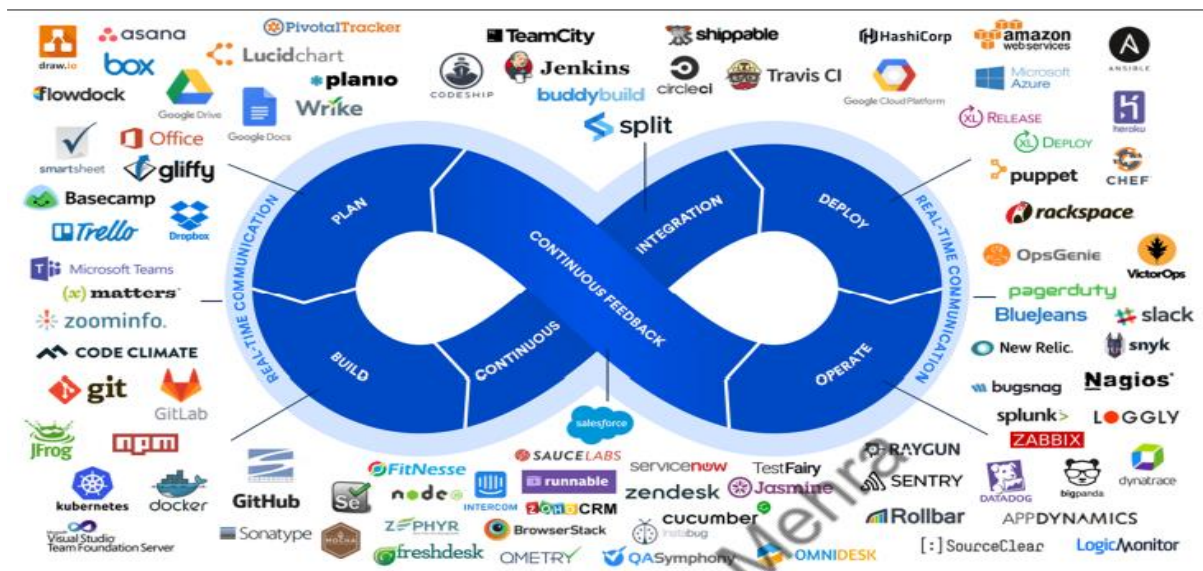
Some of the popular tools for DevOps are

- Jenkins
- Nagios
- Monit
- ELK (Elasticsearch, Logstash, Kibana)
- Jenkins
- Docker
- Ansible
- Git

Conclusion : The basic knowledge about the DevOps was explored