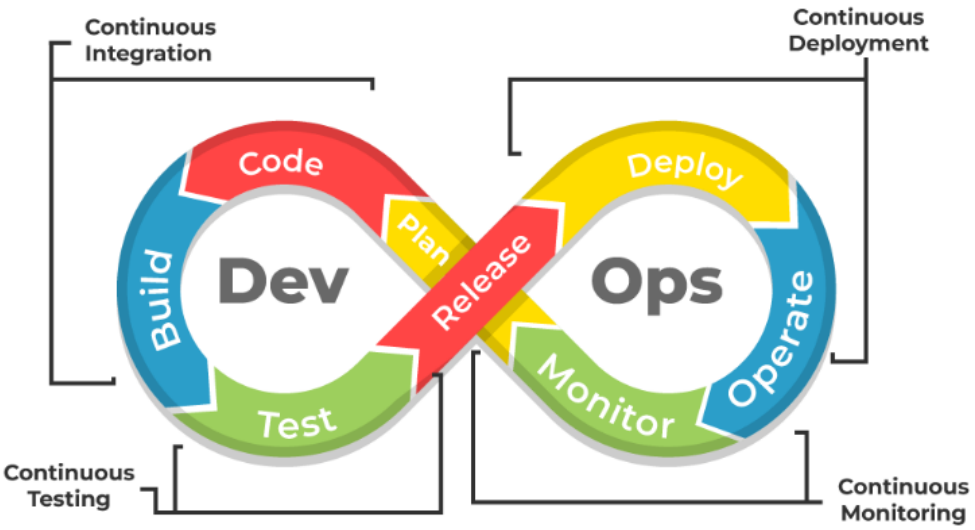
Lesson Plan

Definition & Goals of DevOps

PW SKILLS

Definition & Goals of DevOps

Definition: Devops is a software development methodology which improves the collaboration between developers and operations team using various automation tools. These automation tools are implemented using various stages which are a part of the devops lifecycle.



Key Components and Practices of DevOps:

Continuous Integration (CI):
Definition: Continuous Integration involves regularly merging code changes into a central repository where automated builds and tests run.

Process:
- Developers commit code changes to a shared repository several times a day.
- Automated build tools (like Jenkins, Travis CI) compile the code and run unit tests.
- Issues are detected early, making it easier to fix them.

Continuous Delivery (CD):
Definition: Continuous Delivery ensures that code changes are automatically prepared for a release to production.

Process:
- After CI, automated testing (integration tests, system tests, etc.) is conducted.
- The code is automatically packaged and made ready for deployment.
- While CD ensures the code is ready for production, the actual deployment can still be a manual decision.

Continuous Deployment:
Definition: Continuous Deployment goes a step further than Continuous Delivery by automatically deploying every change that passes all stages of the production pipeline.

Process:
- Successful changes from the CD pipeline are automatically deployed to production.
- This process requires a very robust automated testing setup to ensure only stable changes are deployed.

Infrastructure as Code (IaC):
Definition: IaC involves managing and provisioning computing infrastructure through machine-readable configuration files, rather than through physical hardware configuration or interactive configuration tools.
Tools: Terraform, AWS CloudFormation, Ansible.

Process:
- Define infrastructure configurations in code.
- Use version control to manage these configurations.
- Deploy and manage infrastructure in a consistent and repeatable manner.

Monitoring and Logging:
Definition: Monitoring and logging involve tracking the performance and health of applications and infrastructure, providing insights and alerts on the status of the system.
Tools: Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana).

Process:
- Collect data from applications and infrastructure.
- Analyze data to detect anomalies and performance issues.
- Use alerts and dashboards to monitor system health in real time.

Collaboration and Communication:
Definition: Encourages a culture where development, operations, and other stakeholders work together seamlessly.

Tools: Slack, Microsoft Teams, Jira, Confluence.

Process:
- Use collaboration tools to maintain open and continuous communication.
- Foster a culture of shared responsibility and transparency.
- Conduct regular meetings and reviews to discuss progress, challenges, and improvements.

Goals of DevOps:

1. **Improve Deployment Frequency:**
   - Increase the speed and frequency of software releases to deliver value to customers more quickly.

2. **Faster Time to Market:**
   - Accelerate the process of bringing new features and updates to market.

3. **Lower Failure Rate of New Releases:**
   - Reduce the number of failures and bugs in production by improving testing and quality assurance processes.

4. **Shortened Lead Time for Changes:**
   - Decrease the time it takes from code being committed to being deployed in production.

5. **Faster Mean Time to Recovery:**
   - Reduce the time it takes to recover from failures or issues in production, ensuring higher system reliability and availability.

1. Continuous Integration and Continuous Delivery (CI/CD):
- Automate the integration and delivery process to ensure that code changes are tested and deployed rapidly and reliably.
1. Improved Collaboration and Communication:
- Foster better communication and collaboration between development and operations teams to break down silos and ensure smoother workflows.
1. Increased Efficiency and Productivity:
- Optimize and automate repetitive tasks, freeing up time for more valuable work and innovation.
1. Enhanced Security:
- Integrate security practices into the DevOps process (DevSecOps) to ensure that security is considered throughout the development lifecycle.
1. Customer Satisfaction:
- Deliver higher-quality products faster and more reliably, ultimately leading to increased customer satisfaction and business success.