DevOps

By

Anand Kulkarni

What is DevOps?

Developers & Testers

+

IT Operations

What is DevOps?

- DevOps is a compound of development (Dev) and operations (Ops), DevOps is the union of people, process and technology to continually provide value to customers.
- DevOps is a set of practices that automates the processes between software development(Dev) and IT operations teams(Ops), in order that they can build, test, and release software faster and more reliably.
- DevOps helps is building the culture of collaboration between teams that historically functioned in relative areas.
- At its essence, DevOps is a culture, a movement, a philosophy.

DevOps benefits

- **Predictability:** DevOps offers significantly lower failure rate of new releases
- **Reproducibility:** Version everything so that earlier version can be restored anytime.
- **Maintainability:** Effortless process of recovery in the event of a new release crashing or disabling the current system.
- **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.
- **Greater Quality:** DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues.

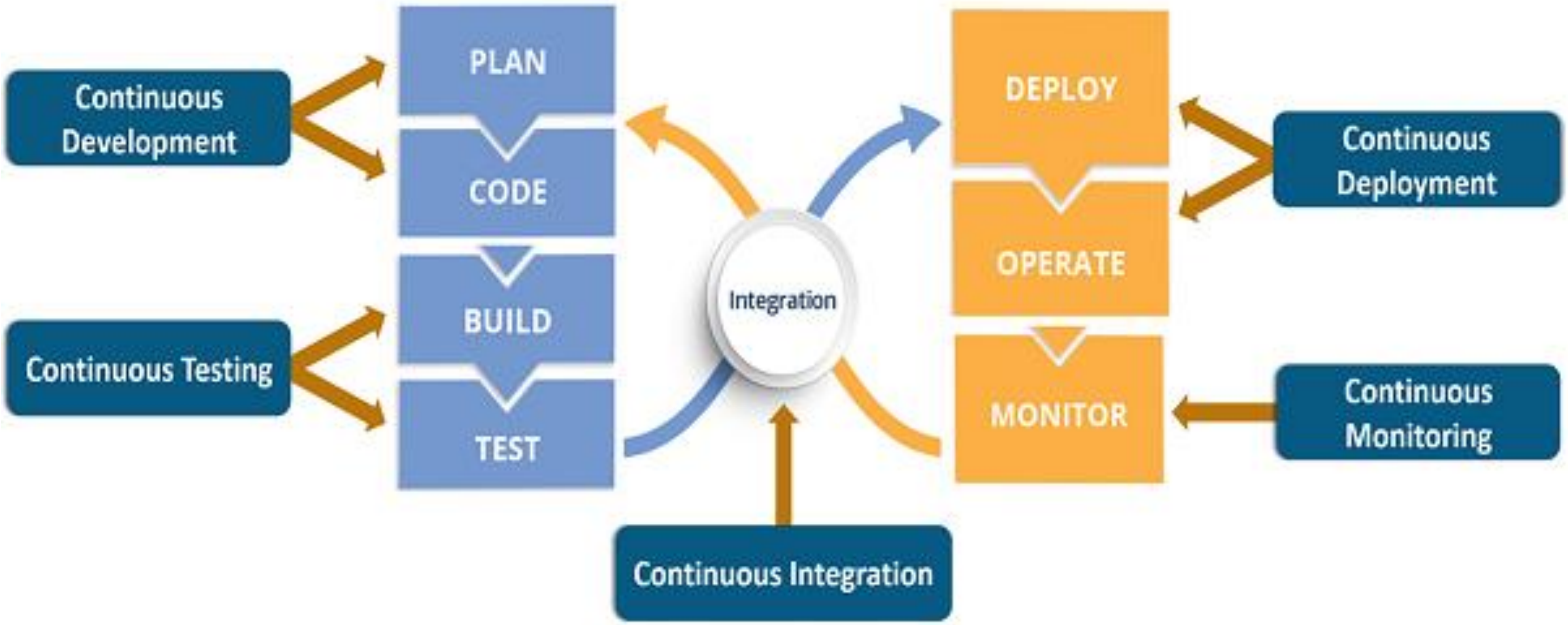
DevOps benefits

- **Reduced Risk:** DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle.
- **Resiliency:** The Operational state of the software system is more stable, secure, and changes are auditable.
- **Cost Efficiency:** DevOps offers cost efficiency in the software development process which is always an aspiration of IT companies' management.
- **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.

Need of 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 Life Cycle



DevOps Life Cycle

- **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.
- **Testing:** QA team use tools like Selenium to identify & fix bugs in the new piece of code.
- **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.
- **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.
- **Monitoring:** In this phase, operation team will take care of the inappropriate system behavior or bugs which are found in production.

DevOps Principles

- **Customer-Centric Action:** DevOps team must take customer-centric action for that they should constantly invest in products and services.
- **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.
- **Continuous Improvement:** DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the improvement of product/services offered.
- **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.
- **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.
- **Monitor and test everything:** It is very important for DevOps team to have a robust monitoring and testing procedures.

Who is DevOps Engineer?

- 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.

DevOps Engineer: Roles & Responsibilities

- 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 Certification

- DevOps training certification helps anyone who aspires to make a career as a DevOps Engineer.
- Certifications are available from Amazon web services, Red Hat, Microsoft Academy, DevOps Institute.

DevOps Automation Tools

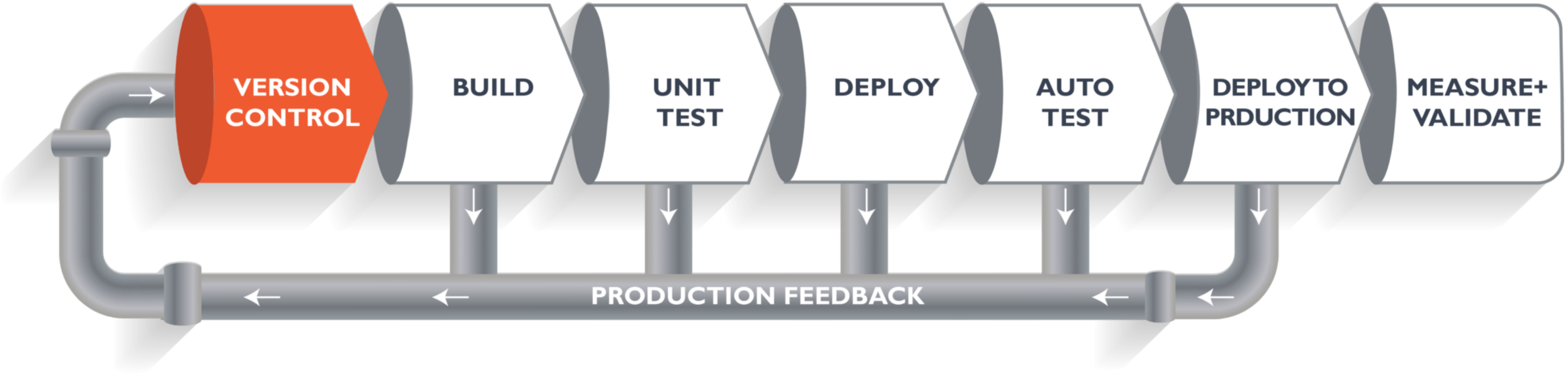
- DevOps automation tools are used to automate all the testing processes and configure them correctly.
- Automation is generally classified in six categories:
  1. Infrastructure Automation
  2. Configuration Management
  3. Deployment Automation
  4. Performance Management
  5. Log Management
  6. Monitoring

DevOps Automation Tools

Automation type	Automation tool
Infrastructure Automation	Amazon Web Services (AWS) cloud service
Configuration Management	Chef – allows to make configuration changes at once place which will be automatically replicated across many production servers.
Deployment Automation	Jenkins – It is used for continuous integration (CI) & testing.
Performance Management	App Dynamics – It is a real time performance monitoring tool. The data collected by AppD helps developers to debug the error.
Log Management	Splunk – It is used for log management like aggregating, storing & analyzing logs in one place.
Monitoring	Nagios – It notifies DevOps team if infrastructure or related services are down.

CI & CD

- CI stands for Continuous Integration and CD stands for Continuous Delivery/Continuous Deployment.
- You can think of it as a process similar to a software development lifecycle.
- Below are the phases in CI/CD pipeline:



CI/CD pipeline phases

- **Version control:** Developers continuously write the code & commit it into the repository using version control tool like GIT, SVN etc. This is a manual phase. It means developer manually needs to commit his code into repository.
- **Build:** It is the first phase of the pipeline, where developers put in their code and then again the code goes to the version control system with a proper version tag. The committed code is compiled again during build phase.
- **Unit Testing:** Once the code committed is built successfully, it executes all unit test cases to check whether the updated code is not breaking any existing functionality.

CI/CD pipeline phases

- **Deploy:** Once unit testing is successful, the code will be deployed into staging or test server.
- **Auto Test:** Once code is deployed, you can run another sanity check during ‘Auto Test’ phase.
- **Deploy to production:** Once sanity testing is successful, the code is deployed on production server.

All above phases can be automated by tool like Jenkins.