What is DevOps?

What Is DevOps?

let's talk a little bit about DevOps.
DevOps is a collaboration between developers and IT operations. It describes a continuous operation whereby everyone involved in the product development lifecycle work together to make the procedure faster, better, and more consistent.

· DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams. It emphasizes team empowerment, cross-team communication and collaboration, and technology automation.

· The DevOps movement began around 2007 when the software development and IT operations communities raised concerns about the traditional software development model, where developers who wrote code worked apart from operations who deployed and supported the code. The term DevOps, a combination of the words development and operations, reflects the process of integrating these disciplines into one, continuous process.

· The term 'DevOps' is a methodology that allows a single team to manage the entire application development life cycle i.e. Development, testing, deployment and operation.

How does DevOps work?

A DevOps team includes developers and IT operations working collaboratively throughout the product lifecycle, in order to increase the speed and quality of software deployment. It's a new way of working, a cultural shift, that has significant implications for teams and the organizations they work for.
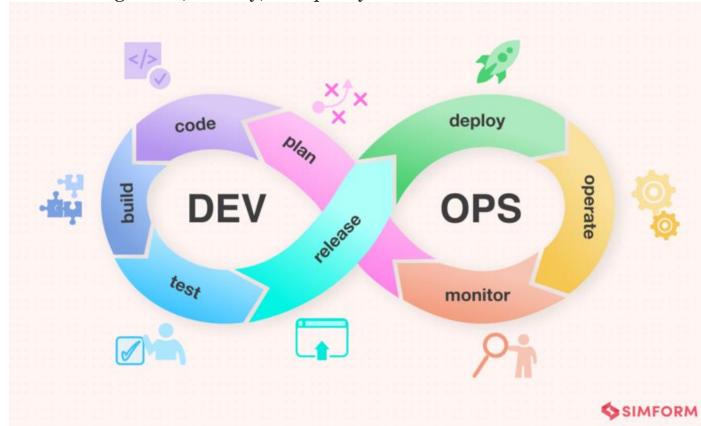
Under a DevOps model, development and operations teams are no longer "siloed." Sometimes, these two teams merge into a single team where the engineers work across the entire application lifecycle — from development and test to deployment and operations — and have a range of multidisciplinary skills.

DevOps teams use tools to automate and accelerate processes, which helps to increase reliability. A DevOps toolchain helps teams tackle important DevOps fundamentals including continuous integration, continuous delivery, automation, and collaboration.
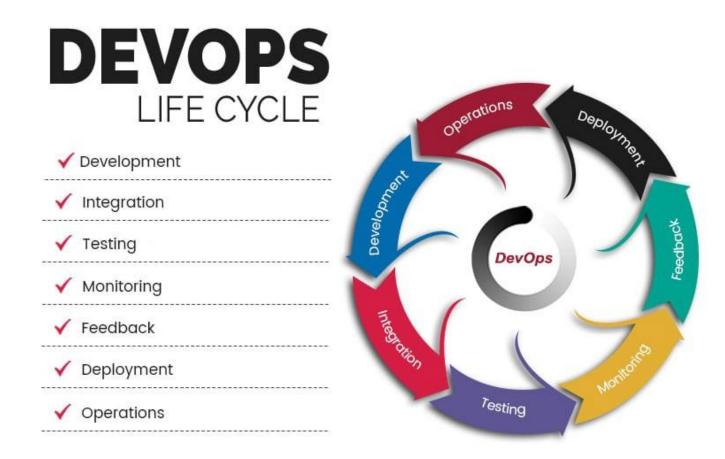
DevOps values are sometimes applied to teams other than development. When security teams adopt a DevOps approach, security is an active and integrated part of the development process. This is called [DevSecOps](#).

The DevOps Lifecycle

The DevOps lifecycle consists of eight phases representing the processes, capabilities, and tools needed for development (on the left side of the loop) and operations (on the right side of the loop). Throughout each phase, teams collaborate and communicate to maintain alignment, velocity, and quality.

The DevOps Lifecycle

DevOps practices

Continuous integration

in continuous integration, new code that adds more features is added to existing code. This code comes from multiple contributors and goes into a central repository that packages it and sends it to either the production or testing server. This helps DevOps teams address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.

Continuous delivery

Continuous delivery expands upon continuous integration by automatically deploying code changes to a testing/production environment. It follows a continuous delivery pipeline, where automated builds, tests, and deployments are orchestrated as one release workflow.

Situational awareness

It is vital for every member of the organization to have access to the data they need to do their job as effectively and quickly as possible. Team members need to be alerted of failures in the deployment pipeline — whether systemic or due to failed tests — and receive timely updates on the health and performance of applications running in production. Metrics, logs, traces, monitoring, and alerts are all essential sources of feedback teams need to inform their work.

Automation

Automation is one of the most important DevOps practices because it enables teams to move much more quickly through the process of developing and deploying high-quality software. With automation the simple act of pushing code changes to a source code repository can trigger a build, test, and deployment process that significantly reduces the time these steps take.

Infrastructure as Code

Whether your organization has an on-premise data center or is completely in the cloud, having the ability to quickly and

consistently provision, configure, and manage infrastructure is key to successful DevOps adoption. IaC goes beyond simply scripting infrastructure configuration to treating your infrastructure definitions as actual code: using source control, code reviews, tests, etc.

Microservices

Microservices is an architectural technique where an application is built as a collection of smaller services that can be deployed and operated independently from each other. Each service has its own processes and communicates with other services through an interface. This separation of concerns and decoupled independent function allows for DevOps practices like continuous delivery and continuous integration.

Monitoring

DevOps teams monitor the entire development lifecycle — from planning, development, integration and testing, deployment, and operations. This allows teams to respond to any degradation in the customer experience, quickly and automatically. More importantly, it allows teams to "shift left" to earlier stages in development and minimize broken production changes.

Why DevOps is important?

DevOps — the amalgamation of development (Dev) and operations (Ops) teams — is an organizational approach that enables faster

development of applications and easier maintenance of existing deployments. By enabling organizations to create stronger bonds between Dev, Ops and other stakeholders in the company, DevOps promotes shorter, more controllable iterations through the adoption of best practices, automation and new tools. DevOps is not a technology per se, but it covers everything from the organisation to culture, processes and tooling. Initial steps usually include CICD, real-time monitoring, incident response systems and collaboration platforms.