

Lecture Notes: Introduction to DevOps

2.1 What is the solution to conflicts among Dev and Ops?

As organisations have become more agile, they leverage their developers' creativity to satisfy their customers' needs. In their quest of agility, the developers push new code frequently. The code changes can lead to downtime of application in production servers as the required environment and dependencies might be missing. Thus, frequent deployments can cause things to break multiple times.

The operation (**Ops**) team is responsible for always keeping the applications available in working condition to customers with no downtimes (ideally ensure 99.999% availability).

Frequent changes are nightmares for the Ops team as they need to ensure a proper environment and installation of the required dependencies in production servers to keep things up and running. Hence, they introduce a lot of formalities before deploying, which in turn slows down the process.

Also, if an application is working in the Development (**Dev**) environment but not in the production environment, the teams blame each other, and this causes a conflict.

Thus, DevOps is a solution to the multiple deployment issues and conflicts between the teams.

2.2 What is DevOps?

Dev team: Consists of professionals who can develop, test, and deploy a feature, or component. The Dev team typically includes software developers and testers.

Ops team: This team handles the infrastructure that is required for the deployment of your application. Provisioning, managing, scaling the servers, load balancers, databases and making applications available for end users are the Ops team's responsibilities.

The uptime of your website is the responsibility of the Ops team.

Hence, DevOps is a methodology that encourages close collaboration between the Dev and Ops teams by using culture and tools. It breaks the virtual wall between the operations and development teams.



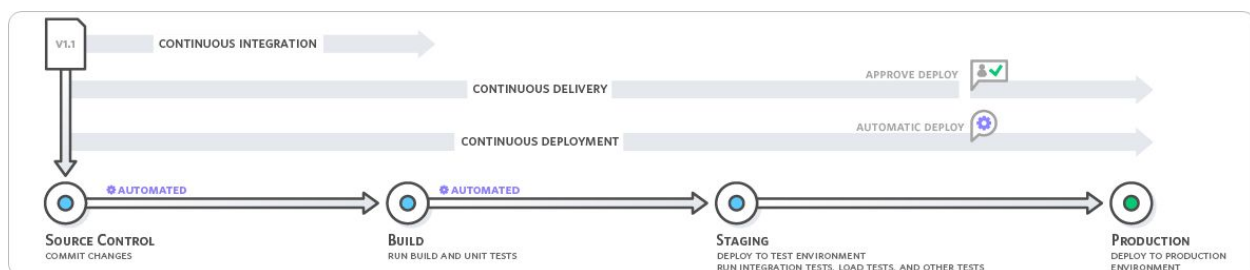
DevOps is a combination of cultural philosophies, practices, and tools that increases an organisation's ability to deliver good quality applications and services at a high velocity- [Amazon.com](https://www.amazon.com)

2.3 CI and CD Pipeline

Continuous integration: A continuous integration server such as Jenkins continuously polls the code repositories like Github, to check if there is any change. Every time a developer commits a code change, the continuous integration server pulls the code, compiles it and prepares a build. Then, the automated test cases are run (automatic unit testing and integration testing is done).

If there is an error in the code, the developer is instantly notified and the issue is then resolved quickly. Thus, this ensures that the repository always has a working and deployable code. The possibility of the wrong code being pushed to further stages gets reduced, making the process cleaner and faster.

Continuous deployment or delivery: It is an extended version of the continuous integration, where the build artefacts are deployed on staging servers and then on the production servers automatically. In continuous delivery, the code is compiled, built, tested and deployed onto a testing environment. But, the manual approval is needed to deploy on the production servers.



2.4 Principles of DevOps

“The Three Ways” to implement DevOps in an organisation are as follows:

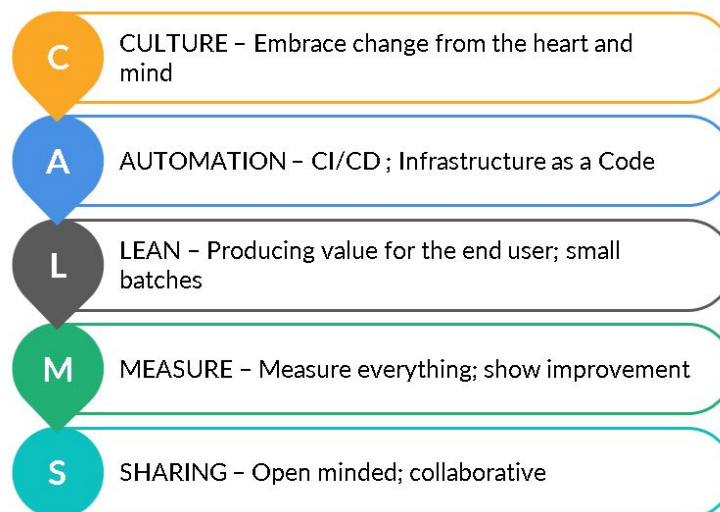
- **Systems thinking:** The emphasis is on the performance of the whole systems rather than its part.
- **Amplifying feedback loops:** If some code is broken, it should return to Dev as soon as possible, and issues should be resolved at the earliest.
- **Culture of continual experimentation and learning:** Create a culture to encourage risk-taking and learning.

The following is an interesting five minute read by Gene Kim, the founder of “**IT revolution**” and “**DevOps Research and Assessment LLC** (Parent Organisation - Alphabet Inc)”.

[The Principles Underpinning DevOps](#)

2.5 CALMS

CALMS is a framework that helps to assess a company's ability to adapt the DevOps processes and measure its success during a DevOps transformation. This acronym was coined by Jez Humble, co-author of *The DevOps Handbook*, and it stands for **culture, automation, lean, measurement, and sharing**.



2.6 Case Study

- At Netflix, a culture of continuous learning and experimentation is encouraged. They built a tool called 'CHAOS monkey' that will randomly kill their own servers. This forces the developers and operators creating the systems to engineer resiliency into their services, instead of being lulled into making the mistake of thinking that their infrastructure is always on.
- Pokemon Go, A virtual-reality based game, deals with a lot of data in realtime. They need to scale in and scale out regularly to optimise their resources without impacting the user experience. Besides this, they need to deploy the new features and bug fixes frequently, without causing a significant downtime. Automated CI or CD pipelines to implement end-to-end agility helped **Pokemon Go** attract millions of active users.

2.7 Key DevOps Practices in Industry

- Version control
- CI or CD pipeline
- Containerisation
- Infrastructure as a code
- Continuous monitoring
- Communication and collaboration

Version Control

Version controlling is the process of tracking and managing changes made to a file present in a repository. Tools like Git are used to track changes made in a code and manage different versions of code. They also help to revert to previous versions if a developer checks a buggy code into the repository.

Version control systems make it easy for different developers to work on the same code base in parallel. They help to resolve conflict if it arises when developers merge their code into a common repository.

Containerisation

Containers allow a developer to package an application with all of its needs, such as libraries and other dependencies and ship it all out as one package.

By doing so, the developer can be assured that the application will run on any other Linux machine, regardless of any customised settings that the machine might have different from the machine used to write and test the code.

Infrastructure as a Code (IaaC)

With IaaC, the engineers write code (generally text files such as YAML and JSON) to describe the servers, load balancers and databases.

It is generally a declarative code used by engineers to write the desired state of infrastructure in a file, which is then used to provision infrastructure repeatedly and consistently without any error.