# **What is DevOps (Development + Operations)?**

DevOps is a software development practice that combines people, processes, and technology to improve the speed and reliability of software delivery. It's a combination of the words "**development**" (Dev) and "**operations**" (Ops).

**DevOps is the combination of culture, practices, tools that increases the ability of the organization to delivery application and services at high velocity.**

# What Is DevOps? Complete Guide to Best Practices - Orange Matter

**Benefits of DevOps**

* Speed
* Rapid Delivery
* Reliability
* Scale
* Improved Collaboration
* Security

**Speed**

Teams that practice DevOps release deliverables more frequently, with higher quality and stability. In fact, the DORA 2019 State of DevOps report found that elite teams deploy 208 times more frequently and 106 times faster than low-performing teams. Continuous delivery allows teams to build, test, and deliver software with automated tools.

**Improved collaboration**

The foundation of DevOps is a culture of collaboration between developers and operations teams, who share responsibilities and combine work. This makes teams more efficient and saves time related to work handoffs and creating code that is designed for the environment where it runs.

**Rapid deployment**

By increasing the frequency and velocity of releases, DevOps teams improve products rapidly. A competitive advantage can be gained by quickly releasing new features and repairing bugs.

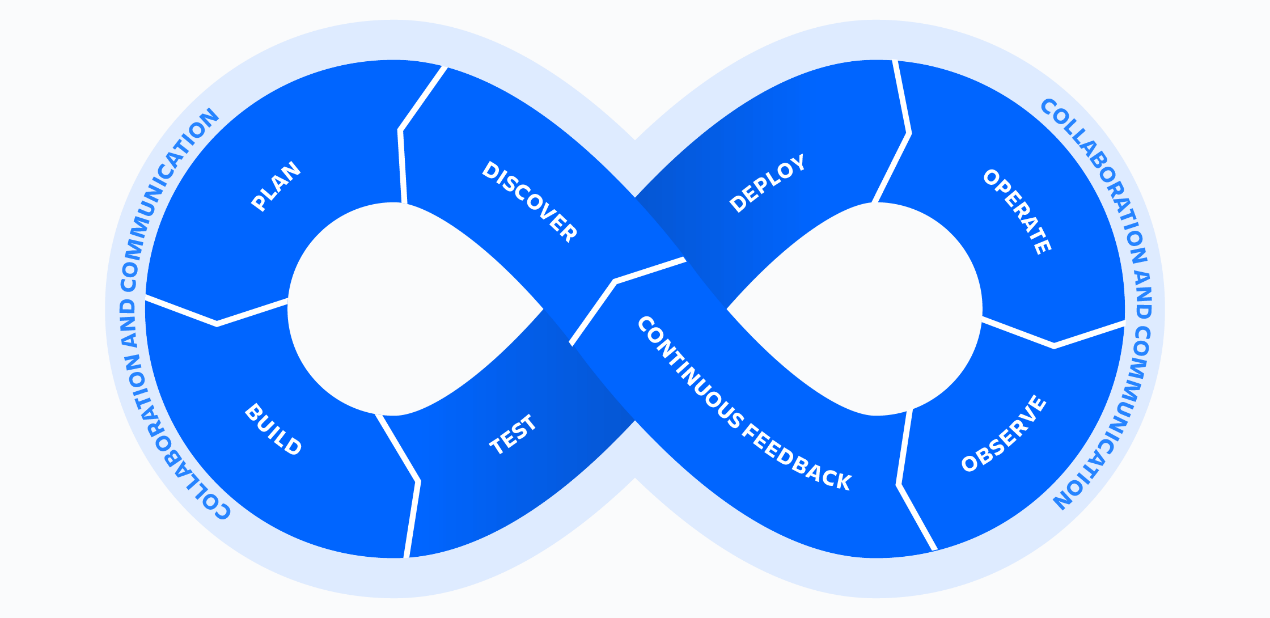
**Quality and reliability**

Practices like continuous integration and continuous delivery ensure changes are functional and safe, which improves the quality of a software product. Monitoring helps teams keep informed of performance in real-time.

**Security**

By integrating security into a continuous integration, continuous delivery, and continuous deployment pipeline, DevSecOps is an active, integrated part of the development process. Security is built into the product by integrating active security audits and security testing into agile development and DevOps workflows.

**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.

**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.**

# **GIT & GITHUB**

What is GIT?

Git is a free, open-source version control system (VCS) that helps developers manage their source code

What is GITHUB?

GitHub is a web-based platform that allows developers to store, share, and collaborate on code

**GIT Commands:**

1. Linux commands:
   1. pwd:- present working directory
   2. ls: - list out the content / files in current directory
   3. cd: - change directory
   4. cd ..: go to previous directory/ one step back
   5. date :- show the current date
   6. touch <file name> :- create new file
   7. mkdir < name> :- create new directory/folder
   8. date <file name> :- to delete / remove the file
   9. rm -rf <directory name> :- to remove / delete the directory / folder
   10. touch <file1> <file2>….<filen> :- to create multiple files at a time
   11. mkdir -p <directory1> <directory2> …. :- to create multiple directory at a time..
   12. my <source> <destination> :- move file from source to destination
   13. cp <source> <destination> : copy the file from source to destination
   14. cat <file name> : to view the content of respective file
   15. vim <file name> :- to edit the respective file
   16. clear :-