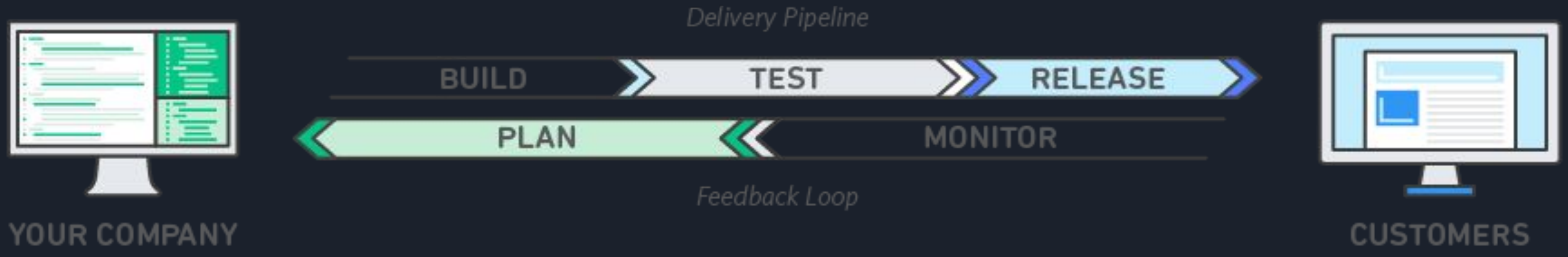
DevOps

By : Bilal Mazhar

What is DevOps

DevOps is the combination of ***cultural philosophies***, ***practices***, and ***tools*** that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.



How DevOps Works

Under a DevOps model, development and operations teams are no longer “siloed.” Sometimes, these two teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function.

In some DevOps models, ***quality assurance*** and ***security*** teams may also become more tightly integrated with development and operations and throughout the application lifecycle. When security is the focus of everyone on a DevOps team, this is sometimes referred to as ***DevSecOps***.

Benefits of DevOps

Speed

Rapid Delivery

Reliability

Scale

Improved Collaboration

Security

DevOps Practices

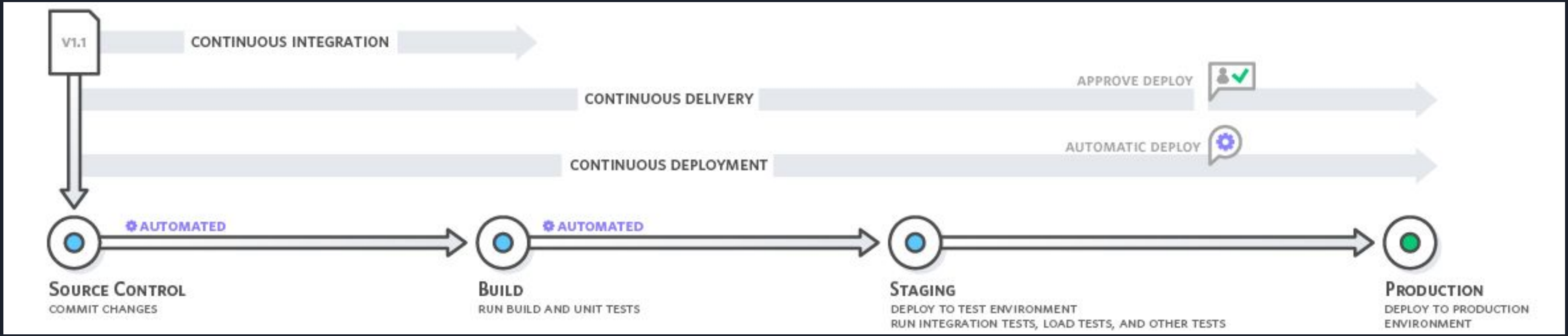
The following are DevOps best practices :

- Continuous Integration
- Continuous Delivery
- Microservices
- Infrastructure as Code
- Monitoring and Logging
- Communication and Collaboration

Continuous integration

Continuous integration is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.

The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.

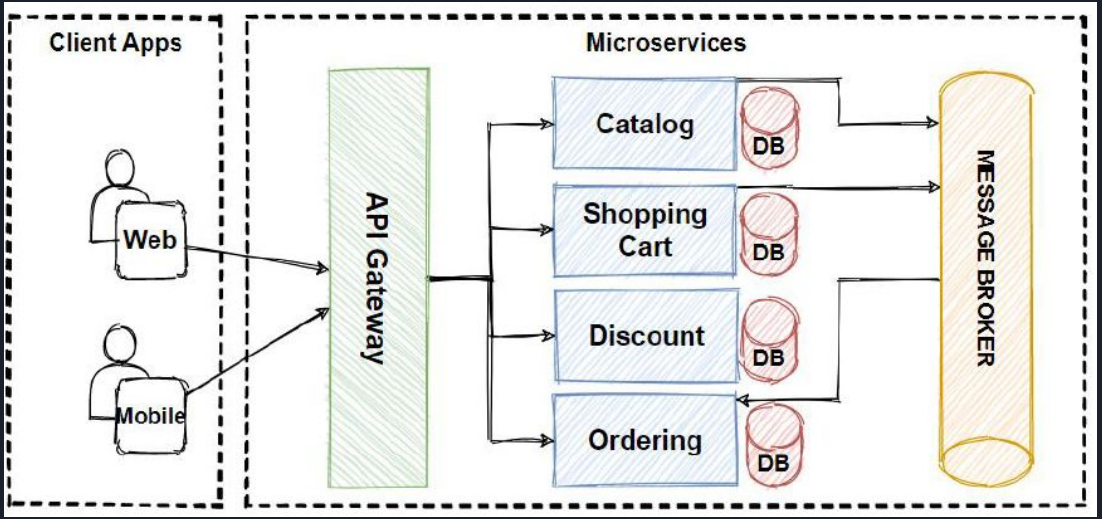


Continuous Delivery

Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process.

Microservices

The microservices architecture is a design approach to build a single application as a set of small services. Each service runs in its own process and communicates with other services through a well-defined interface using a lightweight mechanism, typically an HTTP-based application programming interface (API). Microservices are built around business capabilities; each service is scoped to a single purpose.



Infrastructure as Code

Infrastructure as code is a practice in which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration.

For Example : The cloud’s API-driven model enables developers and system administrators to interact with infrastructure programmatically, and at scale, instead of needing to manually set up and configure resources.

- Configuration Management
- Policy as Code

Monitoring and Logging

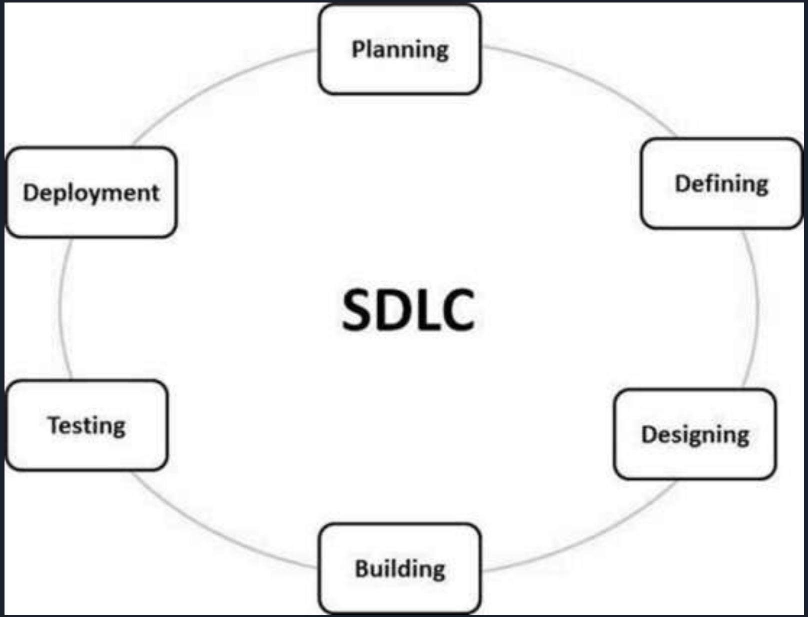
Organizations monitor metrics and logs to see how application and infrastructure performance impacts the experience of their product’s end user. By capturing, categorizing, and then analyzing data and logs generated by applications and infrastructure, organizations understand how changes or updates impact users, shedding insights into the root causes of problems or unexpected changes

Communication and Collaboration

Increased communication and collaboration in an organization is one of the key cultural aspects of DevOps. The use of DevOps tooling and automation of the software delivery process establishes collaboration by physically bringing together the workflows and responsibilities of development and operations. Building on top of that, these teams set strong cultural norms around information sharing and facilitating communication through the use of chat applications, issue or project tracking systems, and wikis. This helps speed up communication across developers, operations, and even other teams like marketing or sales, allowing all parts of the organization to align more closely on goals and projects.

SDLC : Software Development Life Cycle

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process



Agile

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Prerequisite for DevOps

1. Understanding of SDLC Methodology
2. Understanding of Networking
3. Understanding Project Management [agile , scrum etc]
4. Understanding of Cloud
5. Understanding of Cyber Security [OS , Network and Programming]
6. Understanding of Server Administration [Windows server and linux]

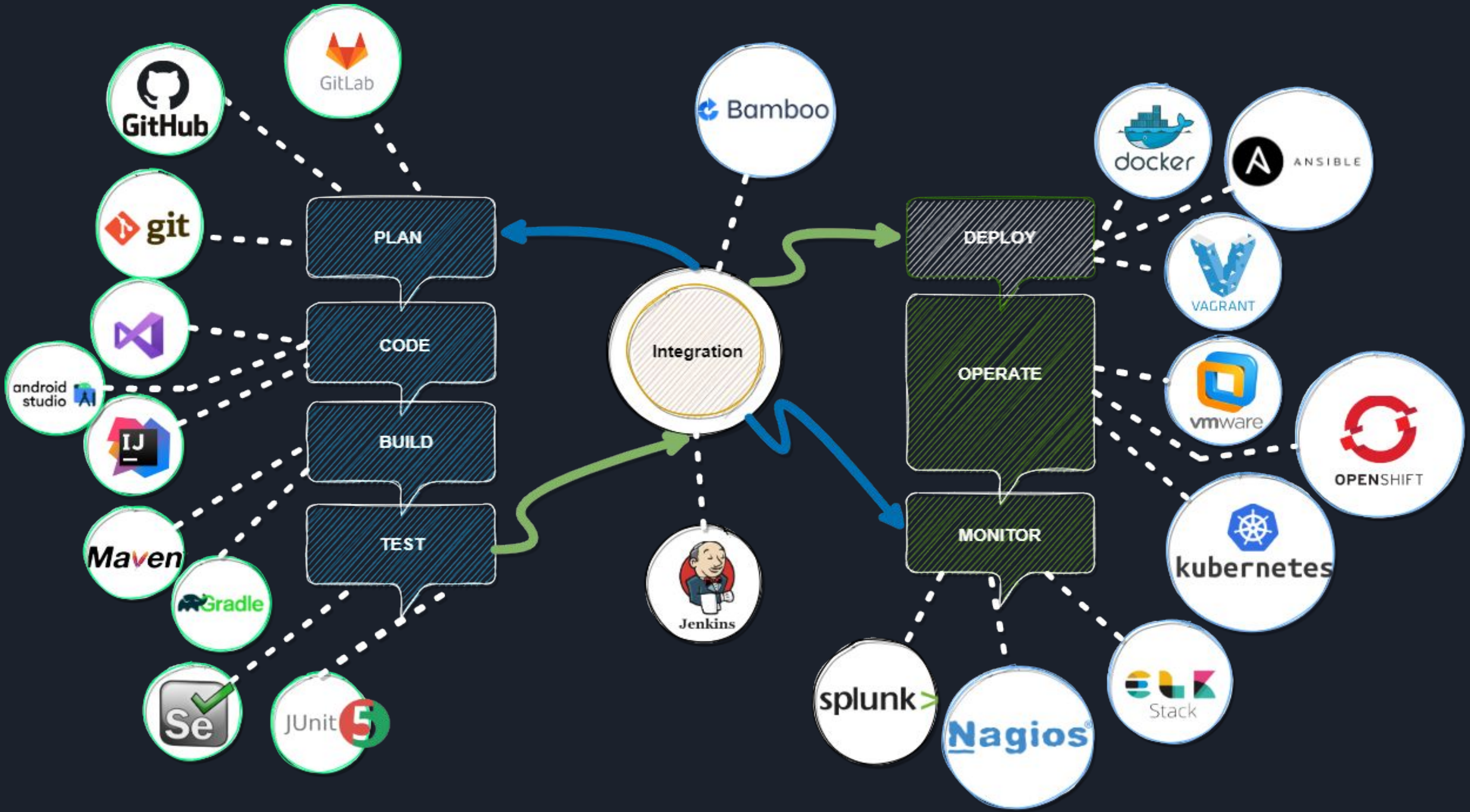
DevOps - Life Cycle

DevOps Lifecycle is the set of phases that involves in DevOps for collaborating development and operation team tasks for faster software delivery. DevOps follows certain processes that include code, build, test, release, deploy, operate, monitor and plan.

DevOps lifecycle follows various phases such as continuous development, integration, testing, continuous monitoring, and continuous feedback.

- Design
- Develop
- Build
- Test
- Deploy
- Operate
- Monitor

Tools in DevOps


GitHub
GitLab
git
android studio
Maven
Gradle
JUnit 5
PLAN
CODE
BUILD
TEST
Bamboo
Integration
Jenkins
docker
ANSIBLE
VAGRANT
vmware
OPENSHIFT
kubernetes
DEPLOY
OPERATE
MONITOR
splunk>
Nagios
Stack