greatlearning
Power Ahead

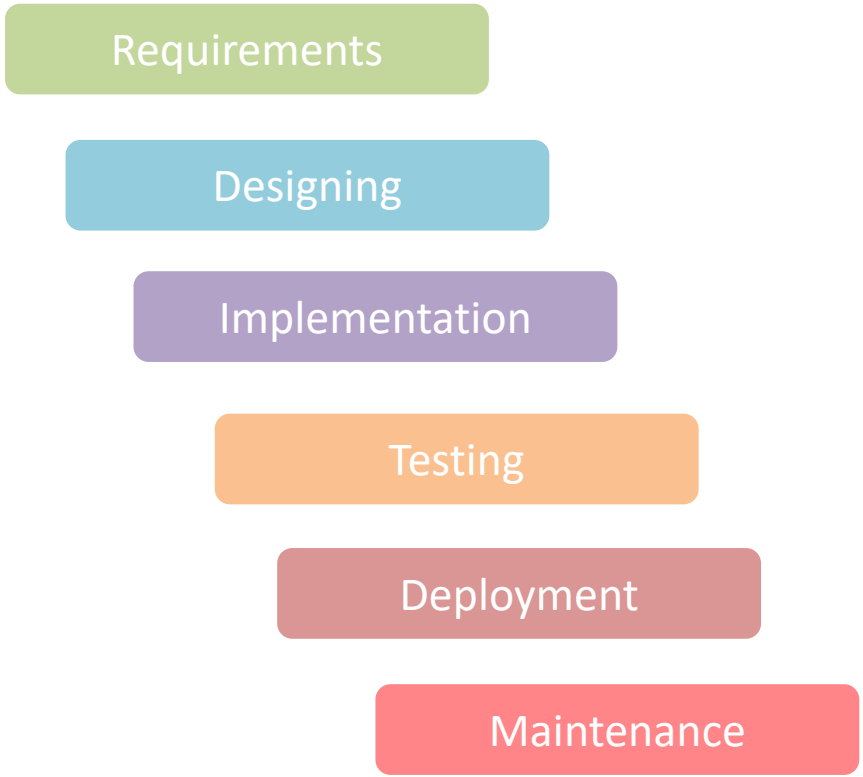
DevOps Tutorial

greatlearning
Power Ahead

Module 1:
DevOps Overview

Why DevOps?

Waterfall Model

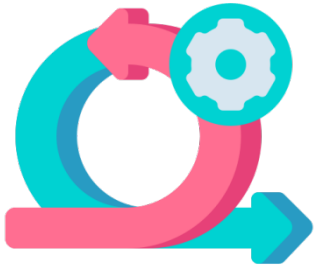


In the early year of software development when it was just coming up, waterfall model was used.

Waterfall model is a very standard model used in many different fields, not just software development.

It was very useful when the requirements were concrete and the development cycles were long.

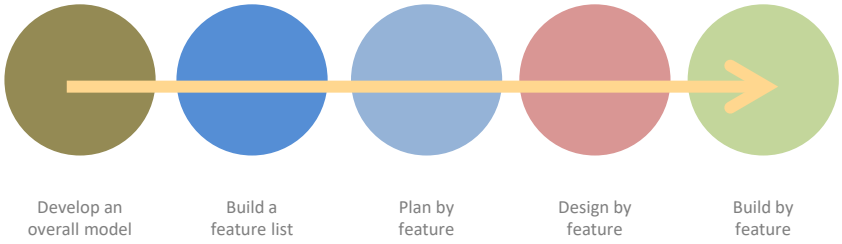
New Methodologies



Agile

Spiral Model

Extreme Programming



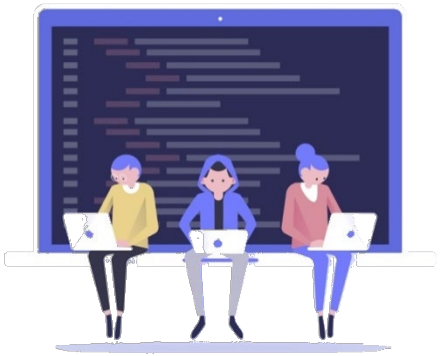
Feature Driven Development

Lean Methodology

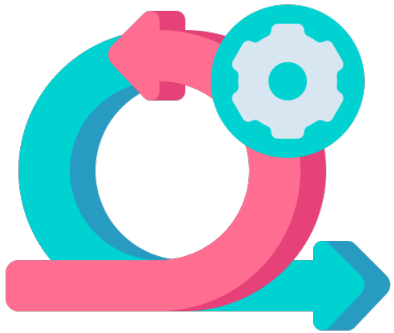
What is DevOps?

What is DevOps?

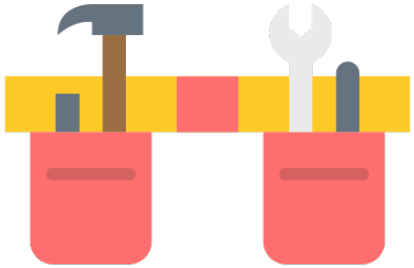
Way of software development



Values & Principle



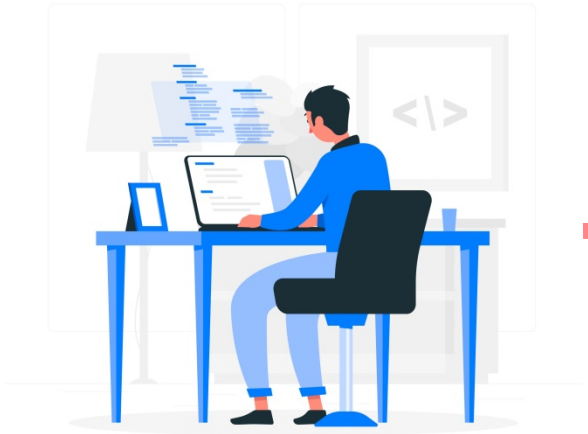
Methodologies



Tools

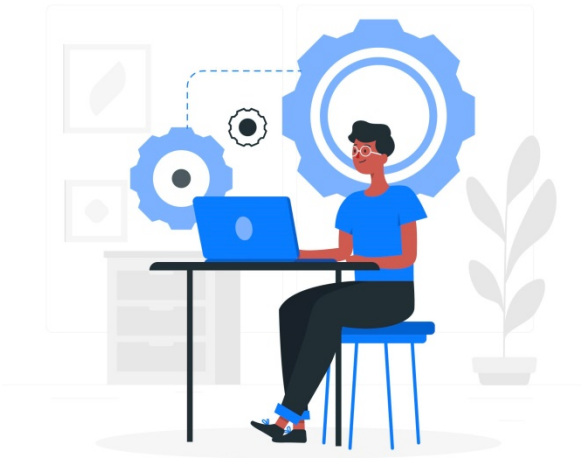
What is DevOps?

The name DevOps comes from Developer and Operations. DevOps bridges the communication gap between the software developers and the It operation teams.



Developers

DevOps



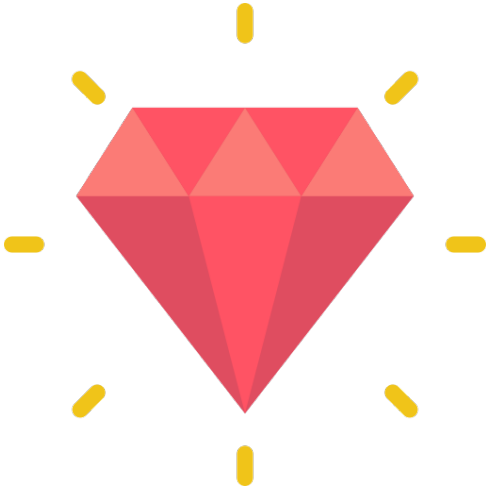
Operations

What is DevOps?

"DevOps is a set of practices that combines software development (*Dev*) and IT operations (*Ops*). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from Agile methodology."

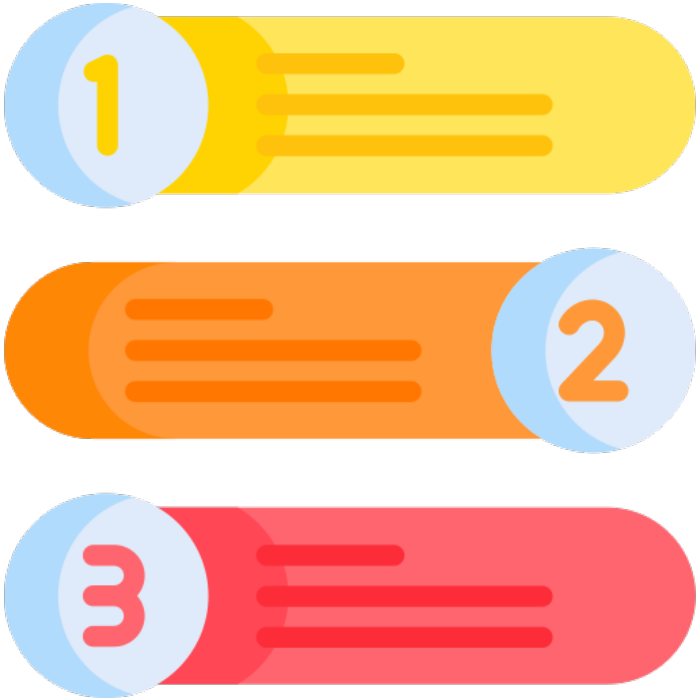
Benefits of DevOps

Productivity



Excellence

Returns



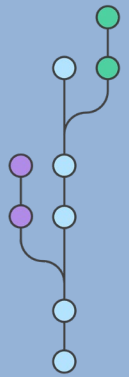
DevOps Phases

DevOps Phases

Continuous Development

Continuous Testing

Continuous Integration



Continuous Deployment

Continuous Monitoring

Who is a DevOps Engineer?

DevOps Engineer

- Basically a DevOps engineer is responsible for designing, maintaining the software development pipeline. He also makes sure that a software is deployed properly without any issues. A DevOps engineer knows how to automate processes and writing automation scripts. He also knows how to keep the whole infrastructure secure and robust.

- A DevOps engineer has excellent communication skills that help him to convey ideas and exchange information between different teams.

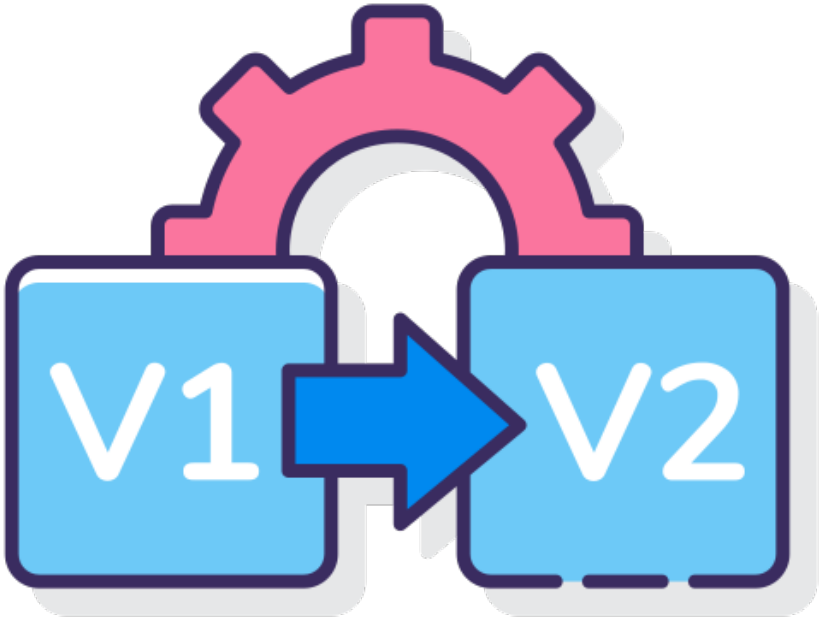
- On daily level, a DevOps engineer is responsible for makes sure scaling needs of cloud are fulfilled, optimisation, managing permissions, documentations.

DevOps Engineer

A DevOps engineer has a wealth of knowledge such as:

- Knowledge of Linux, Windows Operating Systems
- Database Tech
- Server Tech
- Orchestration
- Cloud tech
- Source Control
- Continuous integration & Deployment
- Automation
- Scripting
- Networking
- Excellent Communication

Version Control - GIT



What is Version Control?

What is Version Control?

These days when software is developed, It is not developed with the mind-set that there will only be one piece of code that will be deployed and that's it. These days smaller snippets of code are deployed in regular successions with regular feedbacks. This leads to many different versions of the code.

And that creates a need to organise the code and all of its different version of it. This is where Version Control comes in. It is a practice of managing and storing different version of a source code.

This is especially the case with Larger companies that have multiple projects and multiple teams working within it.

A1, A2, A3

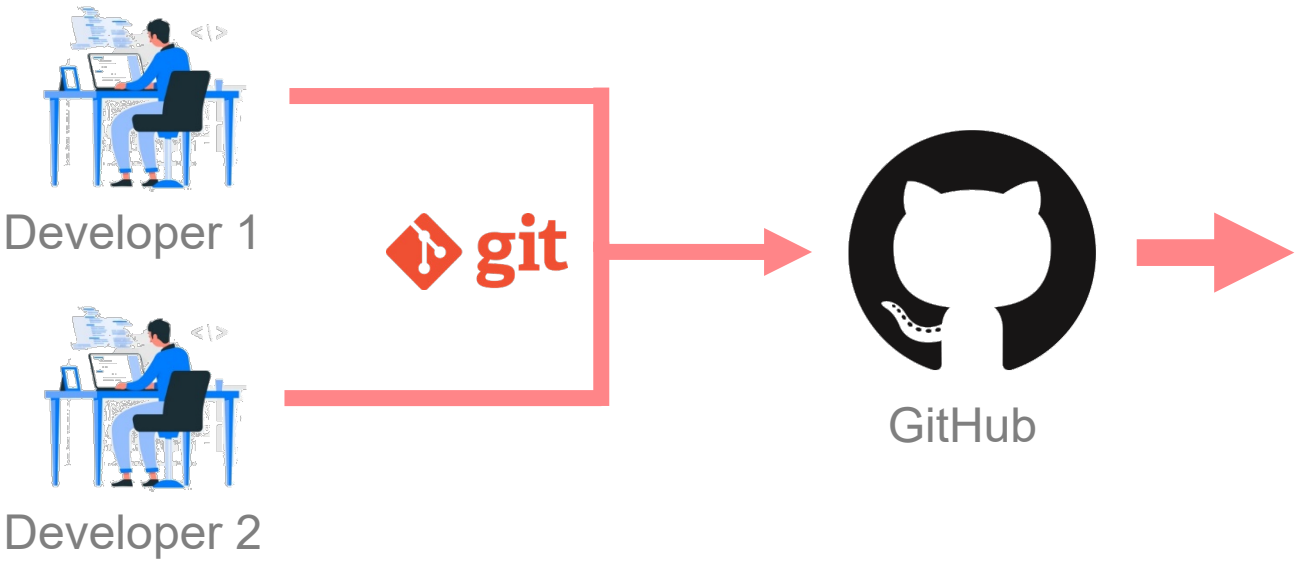
B1, B2, B3

C1,C2,C3

What is Git

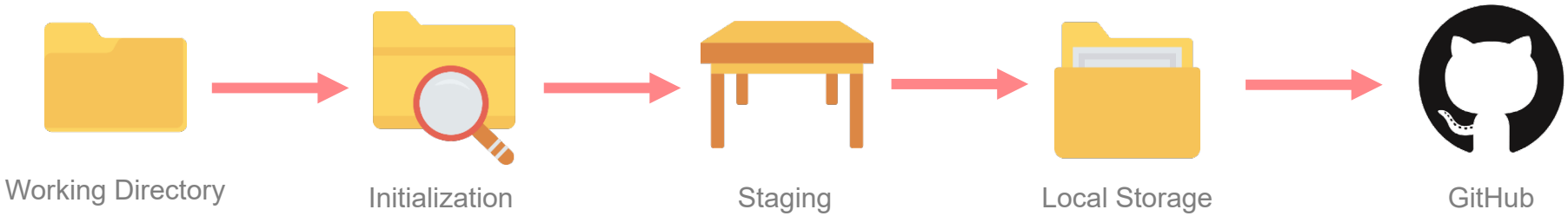
What is Git?

Git is an open source version control system that allows the user to keep track of all the changes that have been made to the source code of the software.



What is Git?

The lifecycle of the code within Git.



greatlearning
Power Ahead

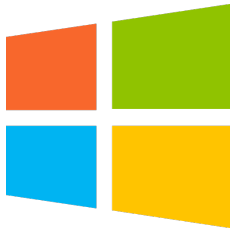
Getting started with GitHub

Installing & Configuring Git

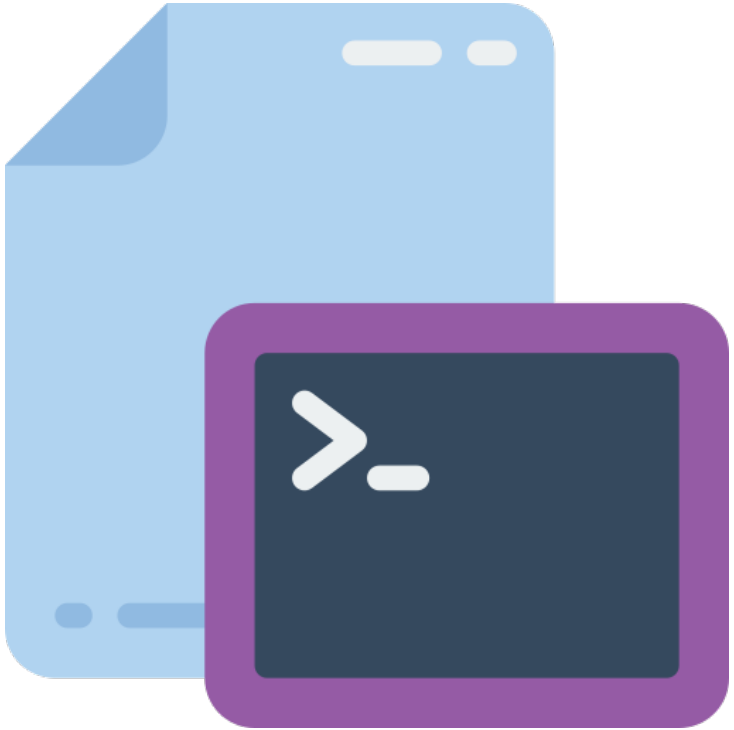
Git Installation & Setup

Linux

Windows



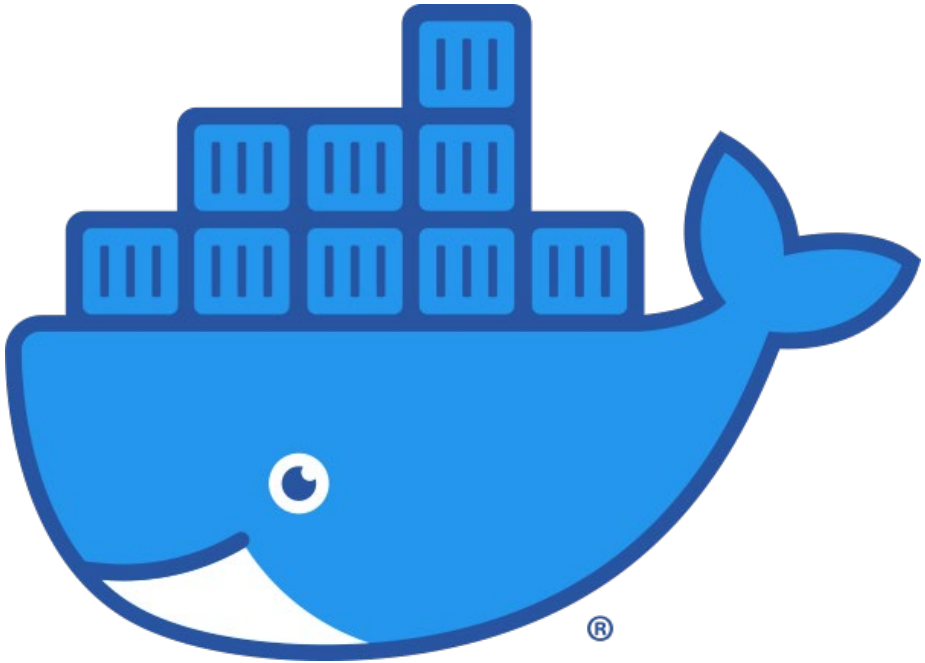
MAC



Git Common Commands

Branches

greatlearning
Power Ahead



Microsystems & Containerization – Docker

Issues we faced before Containerization

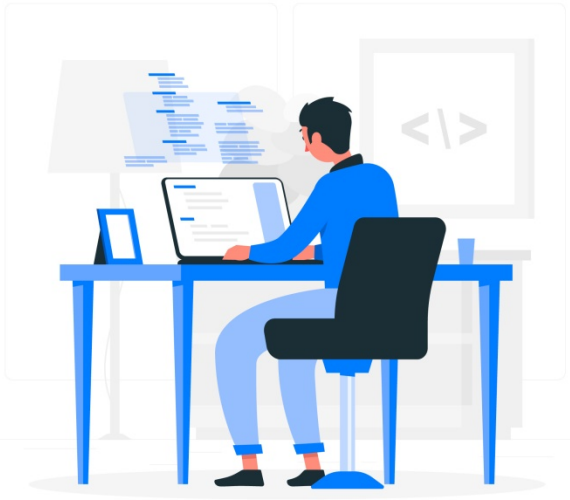
Issues we faced before Containerization

Developer

Tester

Standard Delivery Pipeline

Issues we faced before Containerization



Backend Developer

Operating System

Pytest 5.4.3

Pycharm IDE

Backend Code

Dependencies

Developer's Environment

Issues we faced before Containerization

Tester

Operating System

Pytest 5.3.0

Spyder IDE

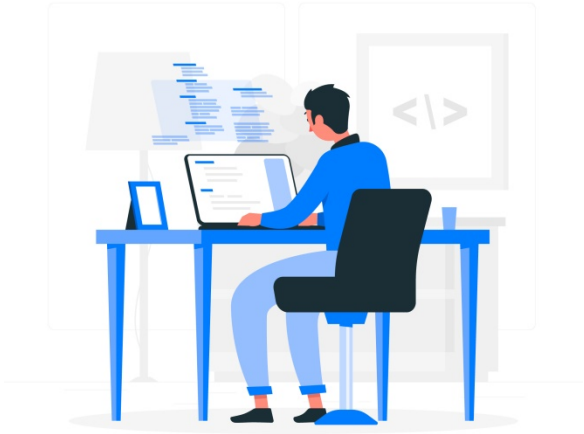
Backend Code

Dependencies

Tester's Environment

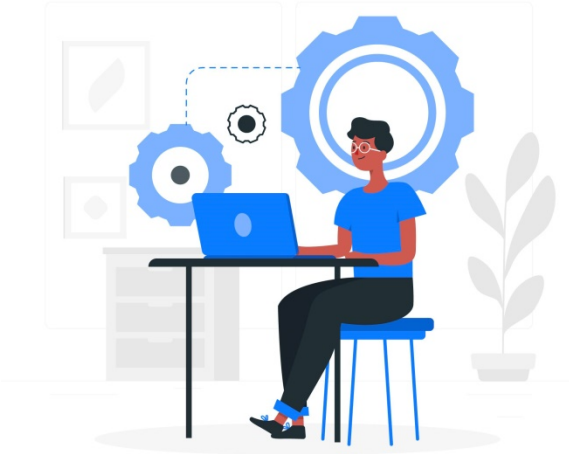
Issues we faced before Containerization

Developer



- This Code runs fine on my computer

Tester



- This Code does not run on my system

Issues we faced before Containerization

Developer

Operating System

Pytest 5.4.3

Pycharm IDE

Backend Code

Dependencies

Tester

Operating System

Pytest 5.3.0

Spyder IDE

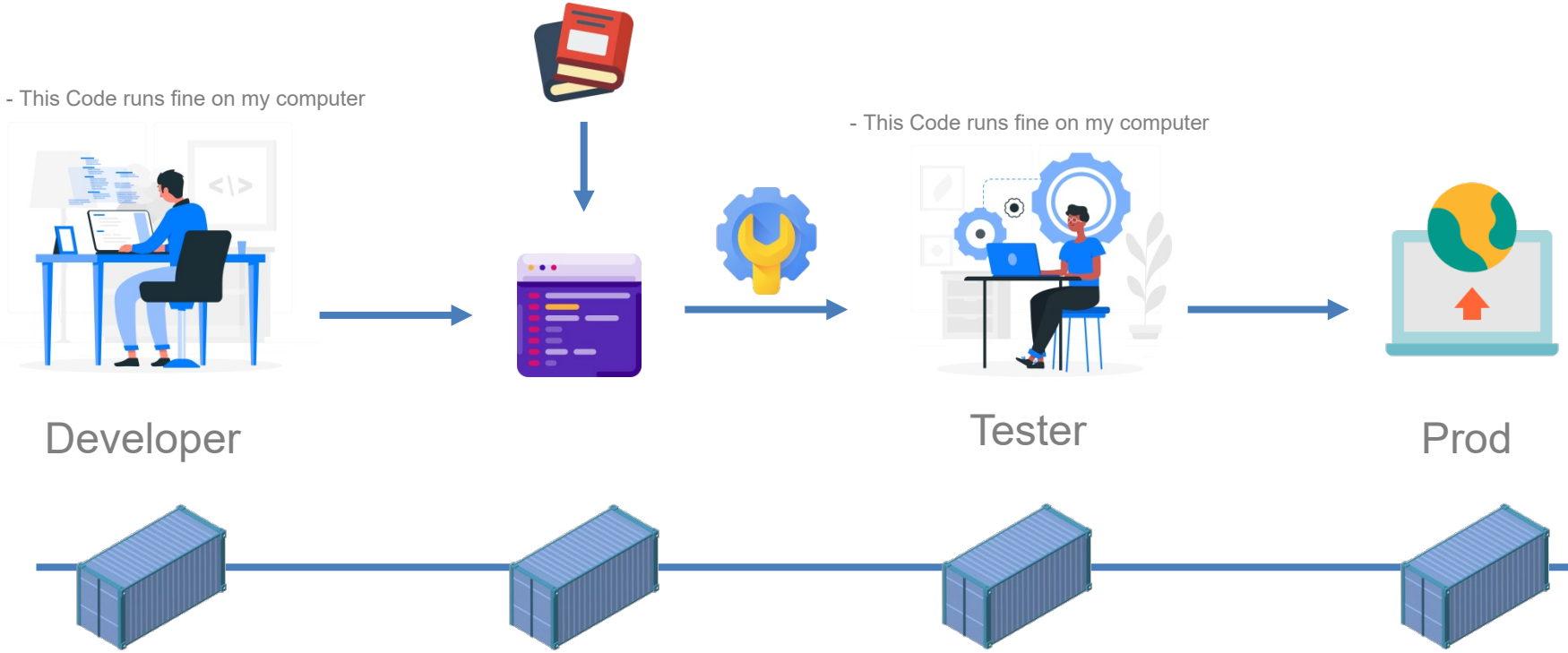
Backend Code

Dependencies

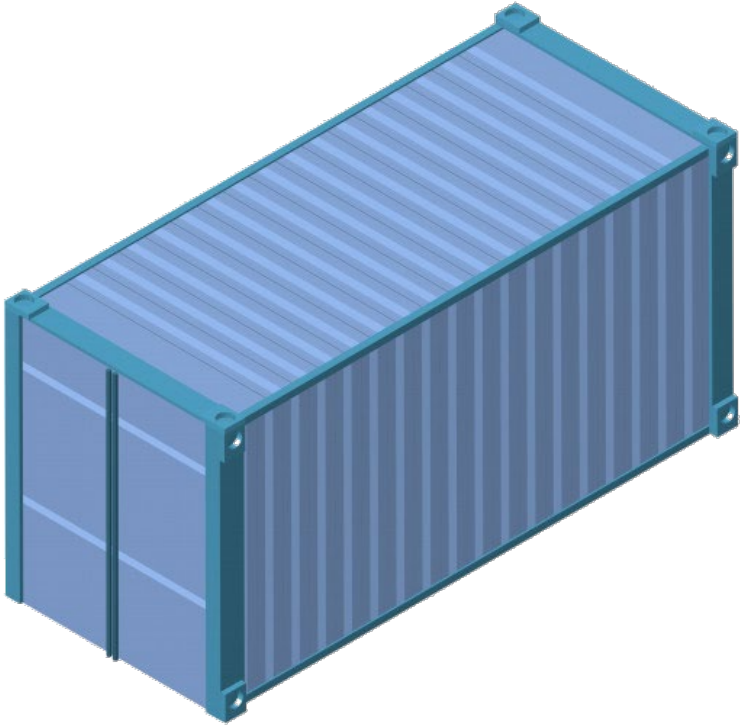
Issues we faced before Containerization

- Now Let's see what happens when we introduce Containers.

Issues we faced before Containerization



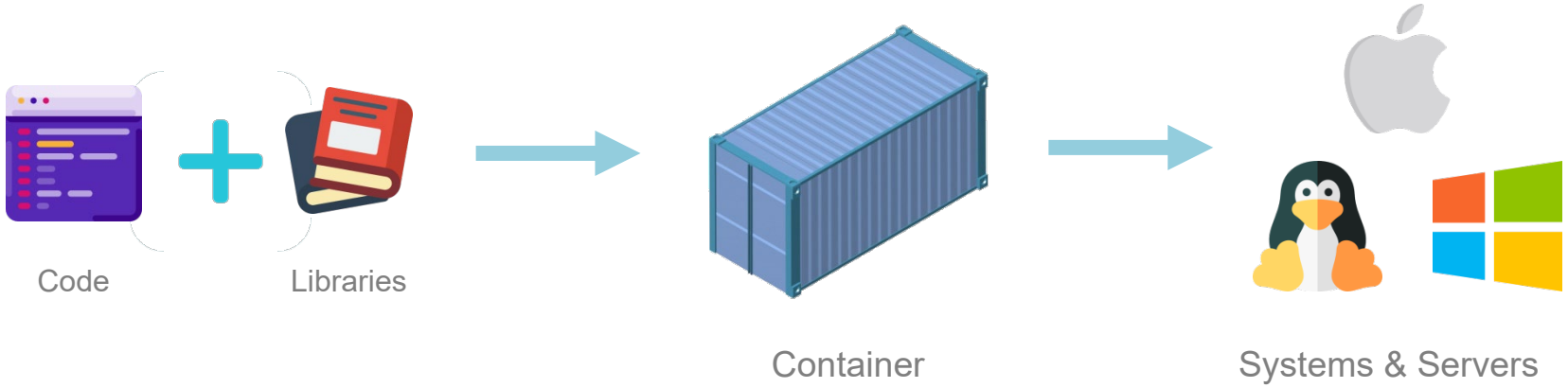
Similar Delivery Pipeline but with containers

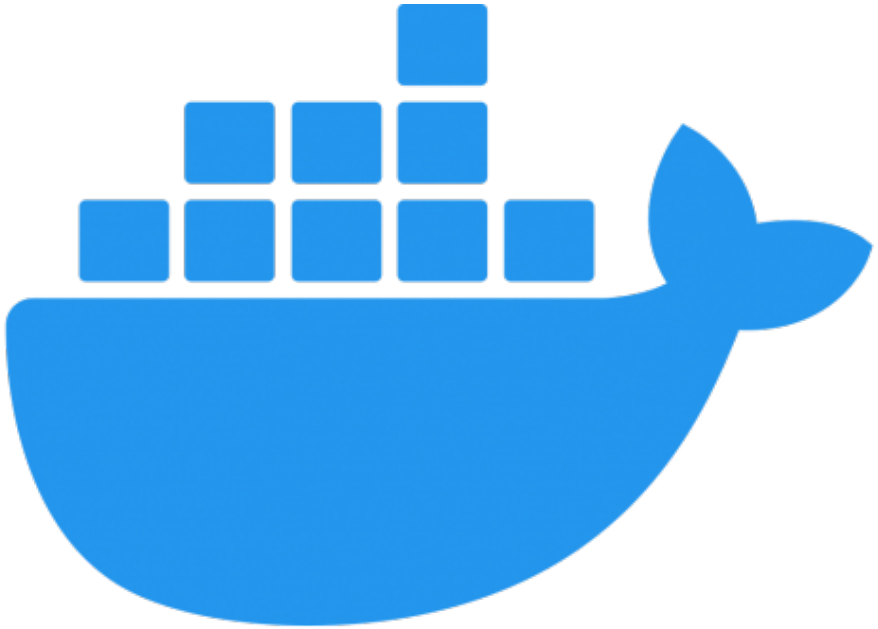


What is a Container?

What is a Container?

Containers are software that wrap up all the parts of a code and all its dependencies into a single deployable unit that can be used on different systems and servers.

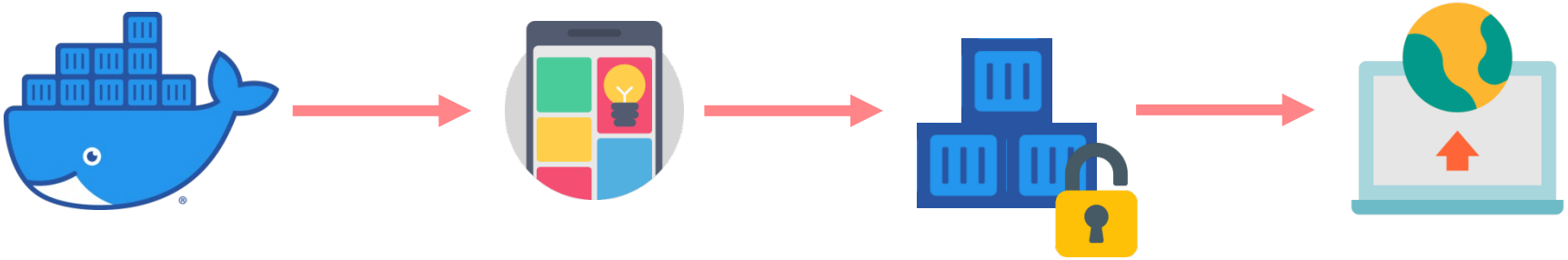




What is Docker?

What is Docker?

Docker is a tool that helps in developing, building, deploying and executing software in isolation. It does so by creating containers that completely wrap a software.



The Isolation provided by container gives a layer of security to the containers.

Why Docker?

Simple

Fast

Easy Collaboration

Built for Developers, by Developers

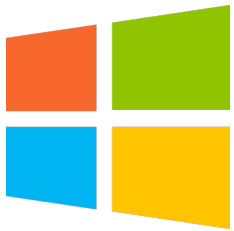
Docker Community

Docker Installation & Setup

Docker Installation & Setup

Linux

Windows



MAC

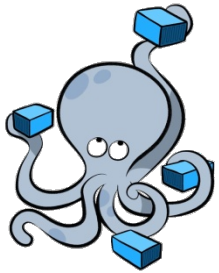
Docker Environment

Docker Environment

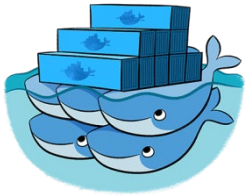
Docker Engine

Docker Objects

Docker Registry



Docker Compose



Docker Swarm

Docker Engine

Docker Engine

Docker engine is as the name suggests, its technology that allows for the creation and management of all the Docker Processes. It has three major parts to it.

Docker CLI

Docker API

Docker Daemon

Docker Objects

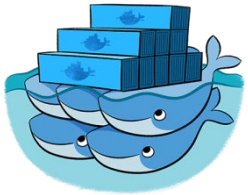
Docker Objects

Docker Images

Docker Containers

Docker Volumes

Docker Networks



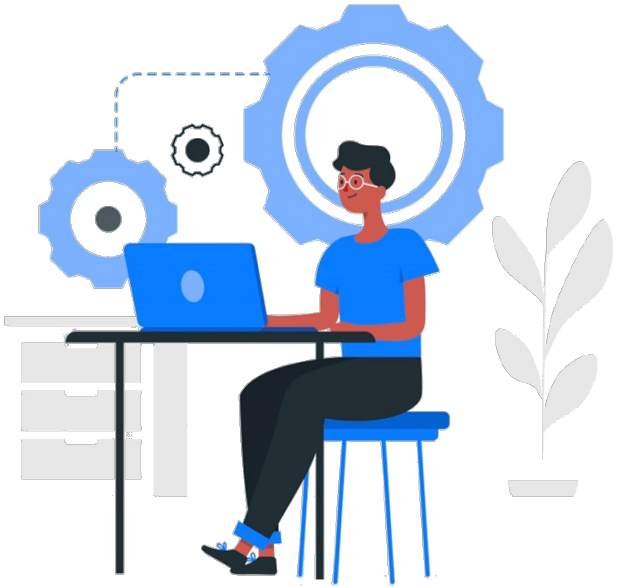
Docker Swarm Nodes & Services

Docker Objects - Images

Docker images are sets of instructions that are used to create containers and execute code inside it.

Docker Objects - Images

Let’s pull an image and work with it.



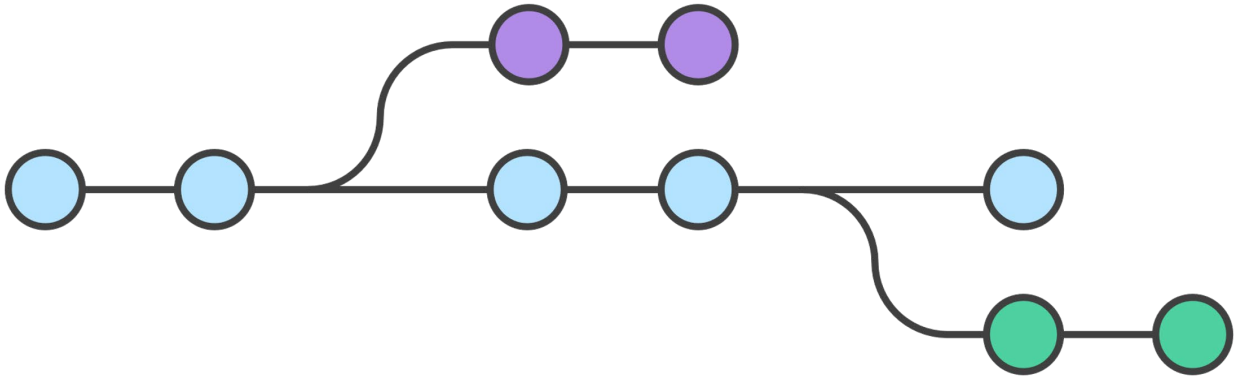
Docker Common Commands

Continuous Integration - Jenkins

greatlearning
Power Ahead

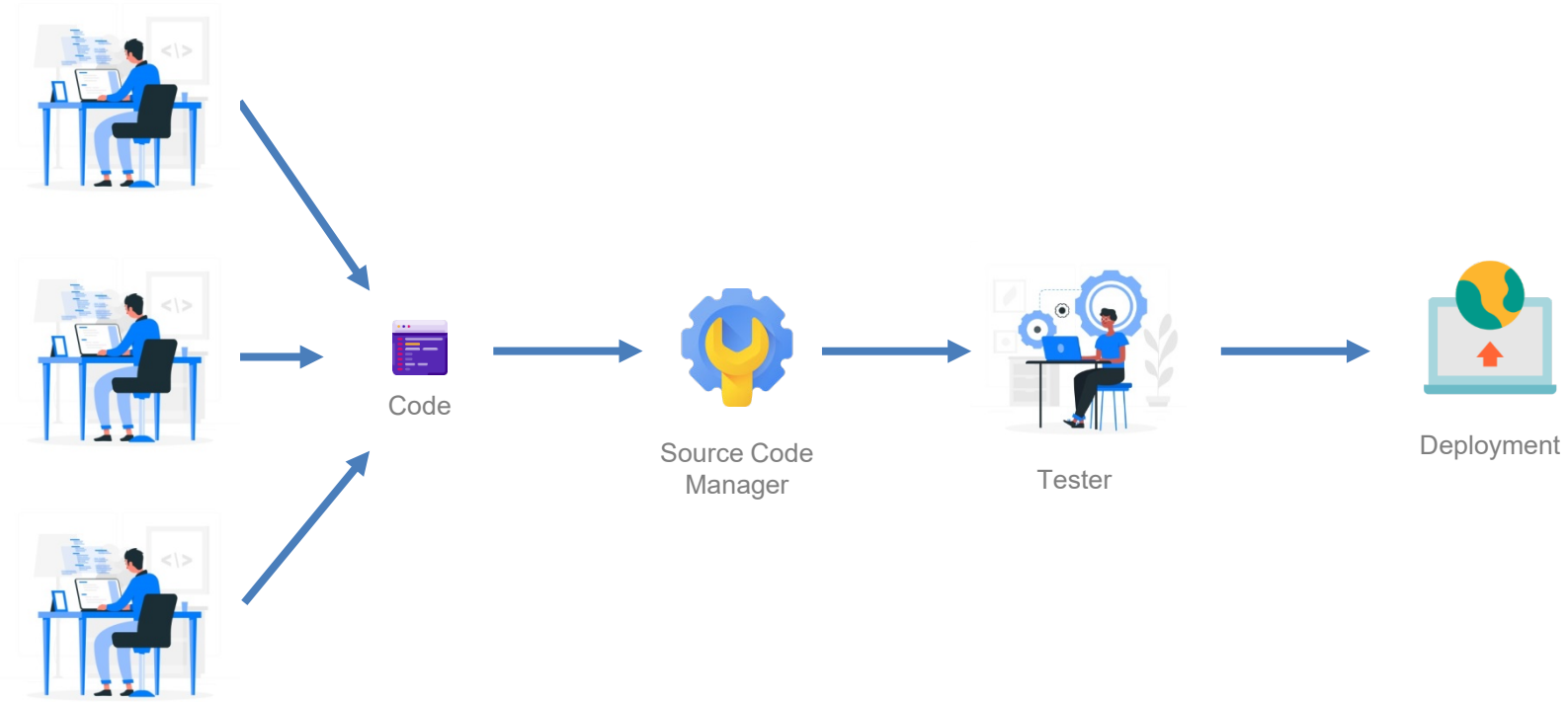
Agenda

1. Why Continuous integration?
2. What is Jenkins?
3. What are Jenkins Features?
4. Jenkins architecture
5. Jenkins installation & setup
6. Setting up a CI/CD pipeline



Why continuous integration?

Why continuous integration?



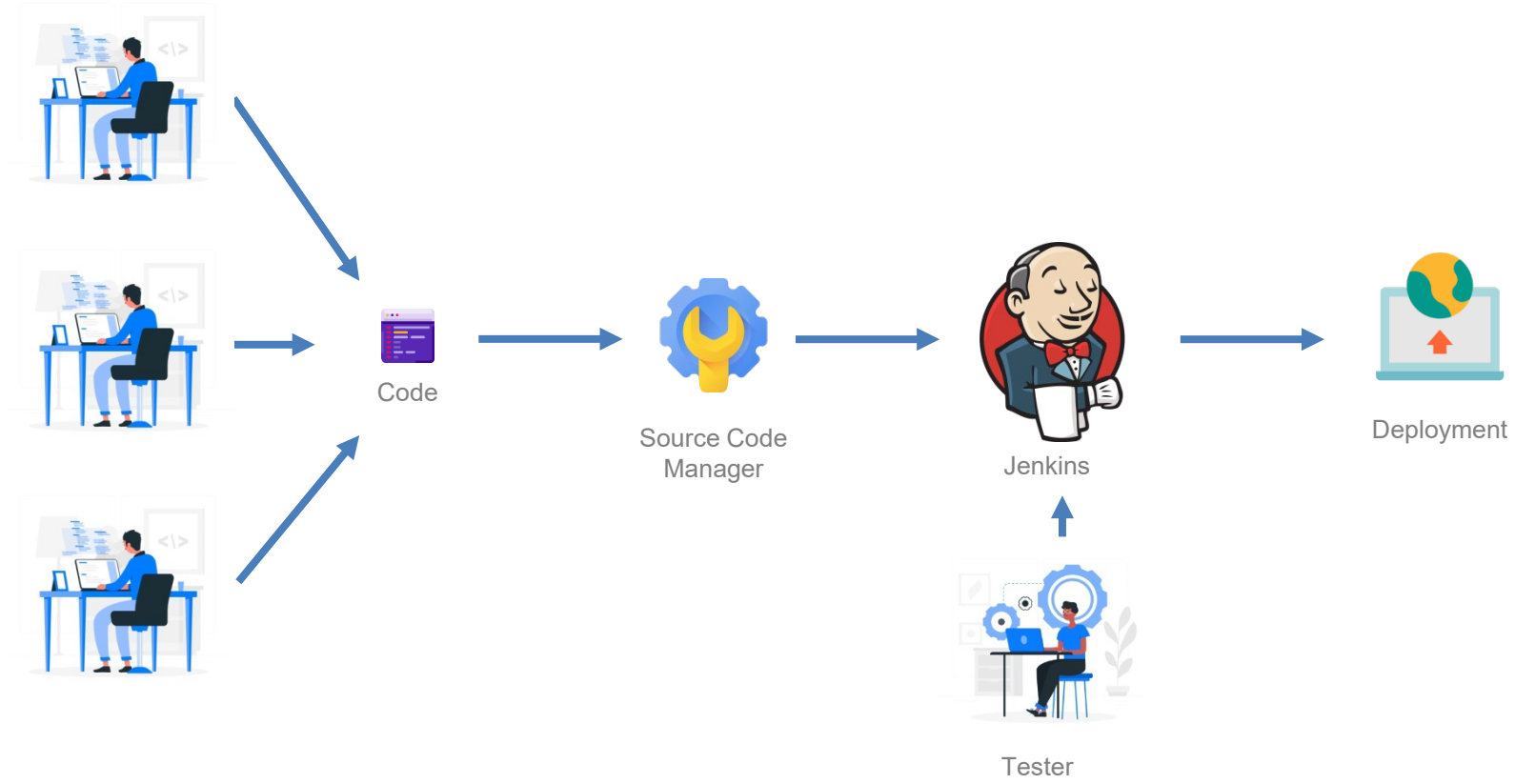
Standard Delivery Pipeline

What is Jenkins?

What is Jenkins?

Jenkins is an Open Source continuous integration tool written in java that allows us to automate the software development process, making sure that there is minimum involvement from us. It also integrates all the different parts of the development.

What is Jenkins?



Benefits of Jenkins

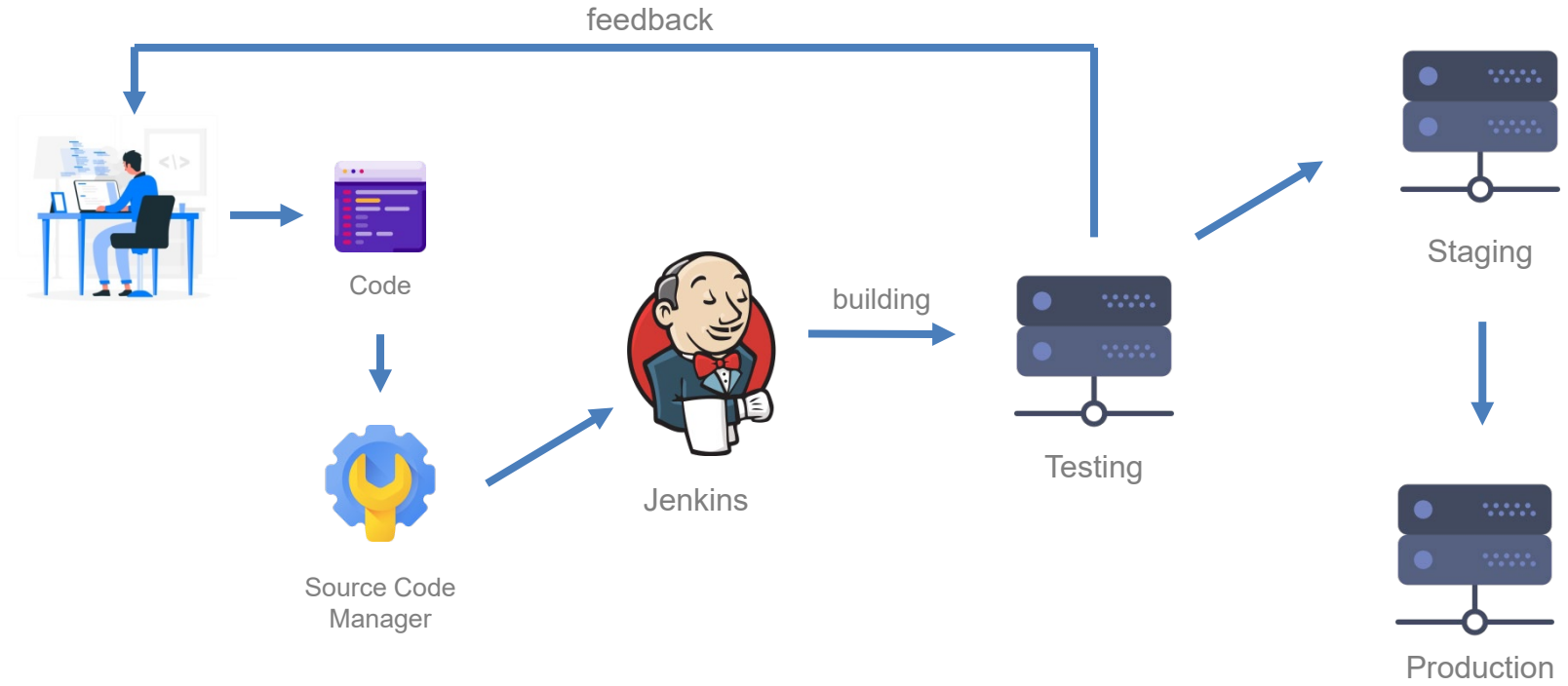
- Jenkins is a community tool – Its open source and has a very very large community that contributes to it regularly.

- It's free, so you don't have to accrue anymore subscription costs.

- It is highly modifiable and adaptable and has a support for a very large number of plugins.

- It can run on any major platform without any compatibility issues.

Jenkins Pipeline



Jenkins Installation & Setup

Continuous Monitoring – ELK & Prometheus & grafana

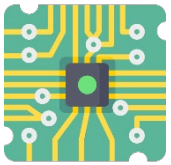
Why Continuous Monitoring?

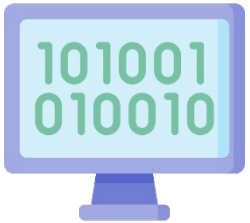
Why Continuous Monitoring?

Continuous Monitoring is an important part of Software Development. It is something we take up as measure to maintain the health of a software and to improve the quality of the software, and this is based on the feed back we get from the insights gained from monitoring.

Why Continuous Monitoring?

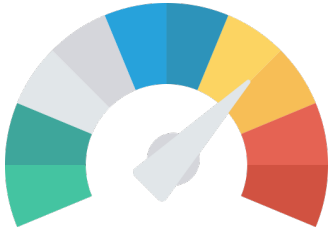
Continuous Monitoring is an important part of Software Development. It is something we take up as measure to maintain the health of a software and to improve the quality of the software, and this is based on the feed back we get from the insights gained from monitoring.





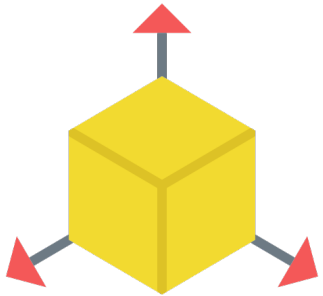
Why Continuous Monitoring?

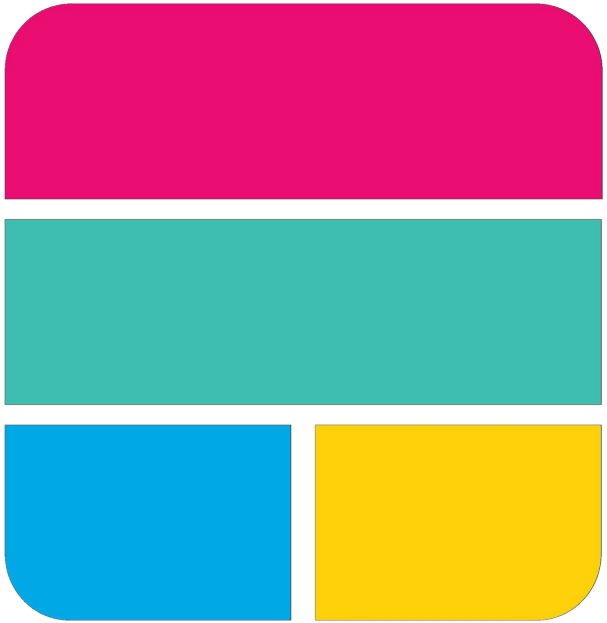
Continuous Monitoring is an important part of Software Development. It is something we take up as measure to maintain the health of a software and to improve the quality of the software, and this is based on the feed back we get from the insights gained from monitoring.



Why Continuous Monitoring?

Continuous Monitoring is an important part of Software Development. It is something we take up as measure to maintain the health of a software and to improve the quality of the software, and this is based on the feed back we get from the insights gained from monitoring.





What is ELK stack?

What is ELK stack?

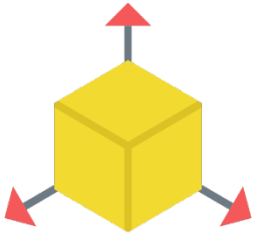
ELK stack is a set of open-source tools that allow us to monitor, collect, process analyze & visualize data, this data can be of different types and formats and from almost any source. It was developed by Elastic co. iteratively. They started with Elastic search and kept on adding more tools to the stack. The primary purpose of ELK stack is log management.

ELK features

System & Application Performance

Logging

Stack Security & Alerting



Scalability and Resiliency

Dashboards & Visualisations

ELK Stack