

# ***My Personal devops course for beginner***

Author:Fouenang Miguel Bruce

Address:[miguelfouenanf@gmail.com](mailto:miguelfouenanf@gmail.com)

This course will be a resumer of all what i will learn from a devops beginner course from udey the course given been given by the MTF institute of management and technology This course is just an introductive course so it will only last for an hour.

## **Devops fundamental concepts,tools and practices**

In this course I'm gonna dive and explore the key concepts for the mastering of devops.

### **Course Overview from udey:**

1. Introduction to devops
2. version control with git
3. Basics of **CI/CD pipelines**
4. Introduction to containers (There use of Docker)
5. Configuration management of Basics
6. Monitoring and logging
7. Cloud basics
8. Devops practices and cultures
9. Mini capstone projects

To this document i will attach other doc from the course for better understanding

Now let dive into learning 😊

### **1.What is Devops??**

Devops represents the culture and practices of integrating and automating development and operations.

There main of devops is for the collaboration,continuous delivery and automation of tasks and process (I had already seen that with used the used of github action).So devops is not about writing code any how but mainly processing the code, correcting it and ensure automation for it processing it .When we talk about continuous delivery it means while an app in already is production and been used we can continue to work on it while been used by every user without any problem that is the continuous delivery process.

So it's deploying services while working along with the software.

To better understand devops we need to know about how software development has been going on throughout the time and how it has evolute with time.The term devops had been introduced in the early 2000s before that developer used to collaborate with each other but did not really have that many tools like today the devops tools where been build in other to facilitate the collaboration between developers and make the building of apps to become something very flexible.

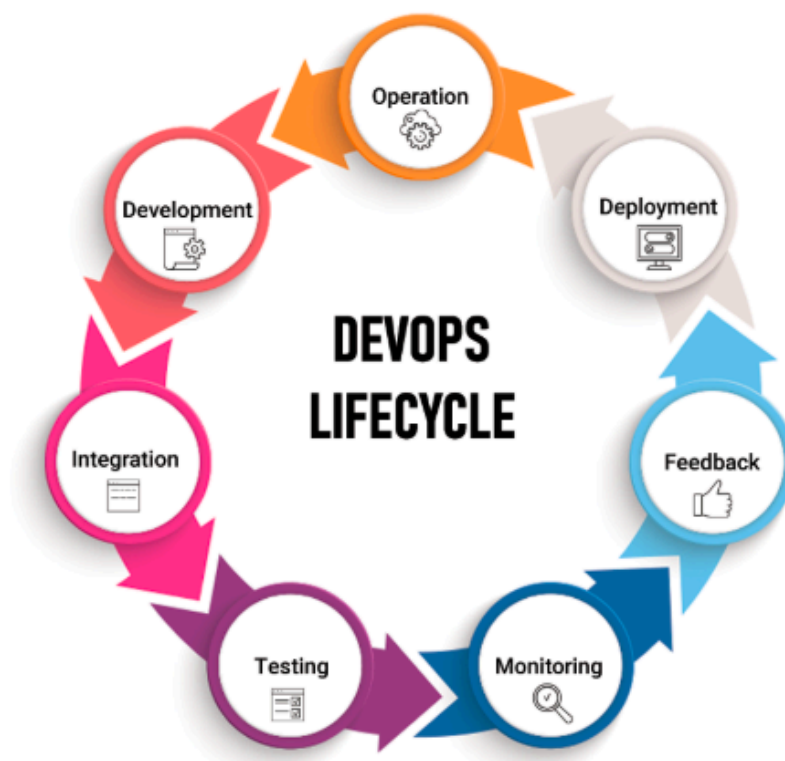
And with all this new tools and technologies were created in order to respond to the working rate of developers and so teams were now able to automate,build,run,and deploy apps faster and easier.

## ***Benefits of devops***

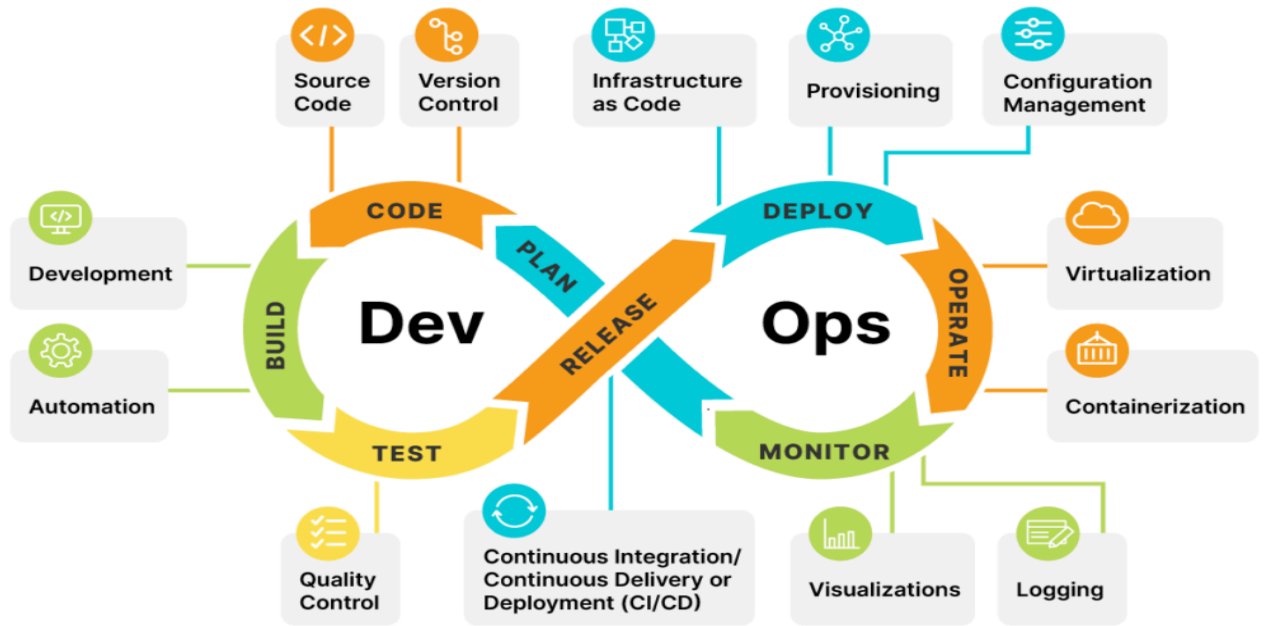
Some benefits of devops are

1. Faster delivery of software
2. Improved collaboration and communication
3. Enhanced reliability through automation

## **The Devops Life Cycle**



**DevOps Lifecycle**



From the pictures above we can now know exactly the devops lifecycle.

### Exercise 1

There aim of this exercise is to

1. Create a github account
2. Create a new repository with name '**Devops fundamentals**'
3. Clone that repo into your pc

For me with my use and master of git and github is was mainly easy

In step 3 we will only need to use the command **git clone** and the git clone command is used with the repository url so that is '**git clone repo-url**'

## Version Control with Git

### What is Git ?

Git is a version control system used for the tracking of code changes. Today Git is a modern tool which is mainly used for collaboration between developers.

Git is a distribution software that means all the dev can have the project on their PC. With the system of branch in github a project can be modified at any time and so we can move from versions to versions for the same project.

For an efficient use of git and github we need to know some crucial commands during working which are

- **git clone**: Use to clone or copy a repository to our Pc
- **git commit**: Use for committing the modifications being made on our work.
- **git push**: Use to upload or push our project or work to github which is mainly online.
- **git pull**: Use to ensure that our local repo is correctly initialise.
- **git branch**: Use to manage and create new branches

## ***Remote collaboration with github***

For the collaboration we gonna need the used of github and the following features are been used for the collaboration with the use of github

- **Pull request**
- **issue Tracking**
- **Team collaboration**

Pull requests are used principally during branch merging while working while issue tracking is used for the correction of errors and for maintaining our code clean by preventing errors from occurring.

### ***Example***

If we have a repository with i have a index.html file we will add a **Readme file** so we gonna add a [readme.md](#) with the command

**'git branch add-readme' or 'git checkout add-readme'**

Within our [Readme.md](#) to add a modification we gonna used the “#” to clearly describe what we have to modify to change on .

To switch back to the main branch we are going to use the command **'git checkout main'** and to merge the branches we are going to use the command **'git merge add-readme'** and make sure that the branches are in the same history.

To delete the branch we are going to use the command **'git push origin main'**.

## ***Basics of CI/CD***

### **what is CI/CD ?**

***CI:Automating the integration of code changes***

***CD:Streamlining deployment processes.***

When a developer makes a change on its content code.The CI has the tasks to ensure that the piece of code that has been added is functional.

**CD** build on the CI for the streamlining deployment processes. So **CI/CD** reduces the time and effort for deploying and automating tasks. some Tools that can be used for the pipeline with CI/CD are

- **Jenkins**
- **Github Actions**
- **GitLab**

And many other tools



Note that

- Jenkins requires a manual configurations
- Github action is suitable for teams work and collaboration
- gitlab CI/CD is mainly for gitlab users that is very similar to github.

### ***Benefits of CI/CD***

- Faster testing and deployment cycles
- Reduced human errors

So with the used of CI/CD i will learn how to build pipelines

### **What is YAML ?**

**YAML** stands for YAML is not a mark up language. YAML is a human-modifiable data serialisation [format](#). It is commonly used in configuration of files for task automation in github

Using associated to in a folder name as '**.github/workflows/main.yml**' the main.yml file is the file that will contain all the automation scripts for the working and github will automatically understand that while we are working.

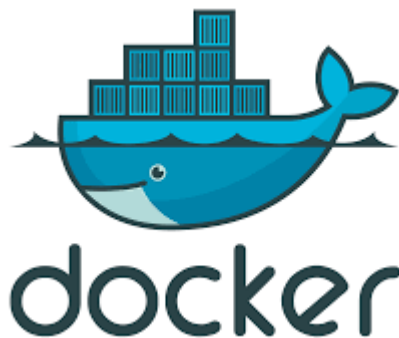
# ***Introduction to containers***

## ***What are containers ?***

Containers can be defined as a lightweight virtualized environment for running applications and this has revolutionised the way applications can be deployed and manage. With containers all apps run in their own personal [environment](#). On their personal environment.

Like virtual machines, containers isolate the app while keeping connection with the host. Containers have come to overcome a serious problem that faces developers that it works on my machine. Containers can be considered as flexible boxes containing everything an application has to have to function properly.

To most popular container build today is known as docker. Docker is container platform used to container.



## ***Installation (Step-by-Step):***

### **1. Download Docker Desktop:**

- Visit [Docker official site](#).
- Download Docker Desktop for your operating system (Windows/Mac).

### **2. Install Docker:**

- Run the installer and follow on-screen instructions.
- Restart your computer if prompted.

### **3. Verify Installation:**

- Open your terminal or command prompt.
- Run **docker --version** to check if Docker is installed.

## ***Docker Basics***

### **Key Concepts**

- ❖ **Image**
- ❖ **Containers**
- ❖ **Docker files**

Docker image is a standardized packet that contains everything an app needs to run with all its dependencies and code for running out.

Containers function like VM (Virtual machines)

Dockerfile is simply a text file that content instructions for building docker image.

## ***Basics Docker Commands***

Some basics docker commands:

- **docker build**: Create an image from a Dockerfile
- **docker run**: Start a container from an image
- **docker stop**: Stops a running container
- **docker ps**: list all running containers

## ***Benefits of Containers***

- Portability
- Consistent environment
- Scalability
- Resource efficiency
- Improve deployment speed

## ***Configuration management Basics***

### **What is configuration management ?**

This is the automating system setup and maintenance to ensure consistency and reliability over our working environment. Configuration management automate and control processes.

## ***Introduction of ansible***

Ansible is one of the most popular configuration management tool and it is a beginner friendly tool. Ansible uses **YAML** based playbook for configuration automation. YAML has 3 key concepts which are **playbooks, modules and inventory**



### ***Installation***

***(Step-by-Step):***

#### ***1. Install Python and pip:***

- **Linux/macOS:** *sudo apt install python3-pip or brew install python.*
- **Windows:** *Download Python from python.org.*

#### ***2. Install Ansible:***

- **Run:** *pip install ansible*

#### ***3. Verify Installation:***

- **Run:** *ansible --version*

## ***Ansible Workflow***

It is mainly a simple process.

- Write a playbook defining and desired configurations
- Specify the target systems in the inventory
- Run the playbook and ansible applies the changes to the specific system

## ***Benefits of Ansible***

- Consistency
- Efficiency
- Reduced errors
- Version control and collaboration
- Faster Recovery

## ***Monitoring and logging***

Monitoring and logging ensures systems are running smoothly, while logging helps to track and diagnose issue.

A popular tool used for the monitoring is **prometheus** and virtualisation **grafana** is been used.

## ***Importance of monitoring***

- Uptime
- performance
- issue détection



### ***Prometheus (Monitoring)***

#### ***Installation (Step-by-Step):***

##### ***1. Download Prometheus:***

- Visit [Prometheus Download Page](#).

##### ***2. Run Prometheus:***

- Extract the downloaded files.
- In the terminal, run: `./prometheus`

##### ***3. Access Prometheus:***

- Open <http://localhost:9090> in your browser.

### ***Grafana (Visualization)***

#### ***Installation (Step-by-Step):***



**1. Download Grafana:**

- Visit [Grafana Official Website](#).

**2. Install and Run Grafana:**

- Follow installation instructions for your OS.
- Start Grafana with: `grafana-server`

**3. Access Grafana:**

- Open <http://localhost:3000> in your browser