We create **innovative software products** that appeal to **global audiences**

Sergio Andres Meneses

SysAdmin Sr. | CloudOps

Gabriel Felipe Diaz

DevOps Sr. | CloudOps

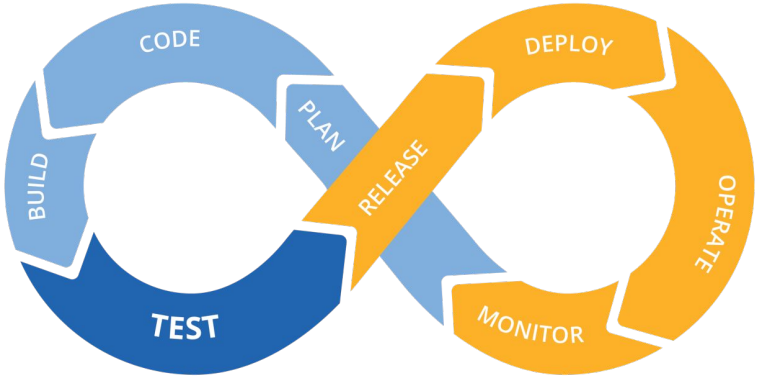
INTRODUCCIÓN A PIPELINES Y A LA CULTURA DEVOPS

WHAT IS DEVOPS?

DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.

Another definition:

DevOps is a set of software development practices that combines software development (**Dev**) and information technology operations (**Ops**) to shorten the systems development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives.



CI & CD

Continuous Integration

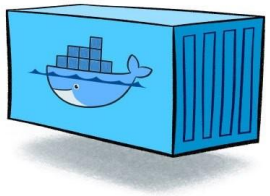
Continuous integration is a coding philosophy and set of practices that drive development teams to implement small changes and check in code to version control repositories frequently. Because most modern applications require developing code in different platforms and tools, the team needs a mechanism to integrate and validate its changes.

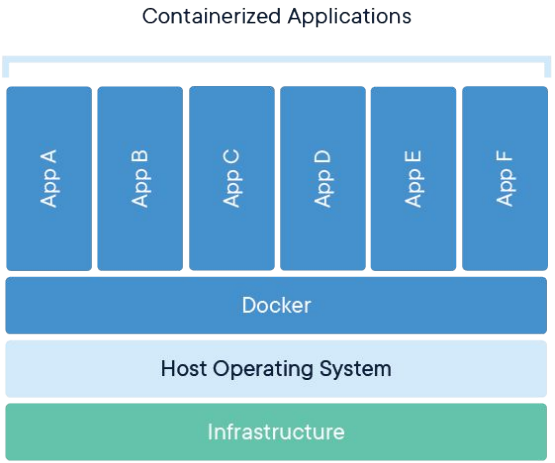
Continuous Delivery

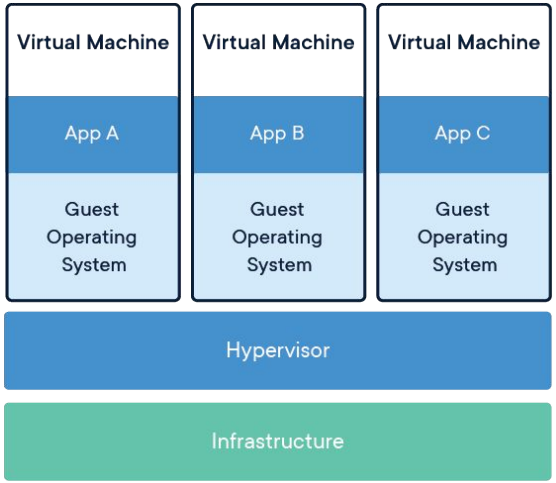
Continuous delivery picks up where continuous integration ends. CD automates the delivery of applications to selected infrastructure environments. Most teams work with multiple environments other than the production, such as development and testing environments, and CD ensures there is an automated way to push code changes to them. CD automation then performs any necessary service calls to web servers, databases, and other services that may need to be restarted or follow other procedures when applications are deployed.

DOCKER

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

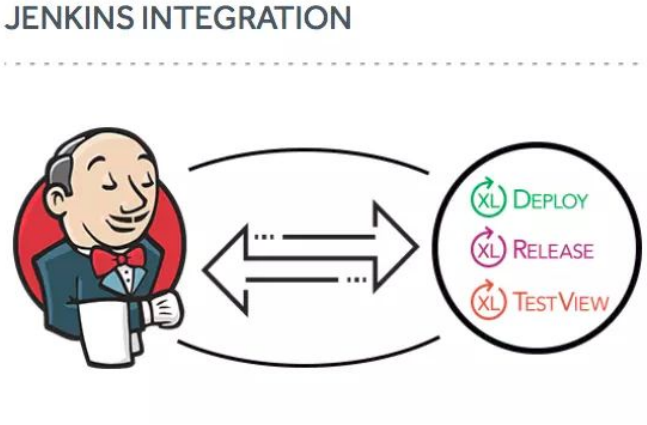






JENKINS

Jenkins is an open source implementation of a **CI** server written in **Java** that can be used as a self-hosted option automating the build cycle for any project. It works with any programming language and for multiple platforms including Windows, Linux and macOS.



WORKSHOP CONTEXT

Our team is working on an automatization process for a local app.

- We need to get a pipeline working for the whole process.
- The app is a static site.
- Containerize the app.
- The main goal is to achieve the pipeline and run update into our app.

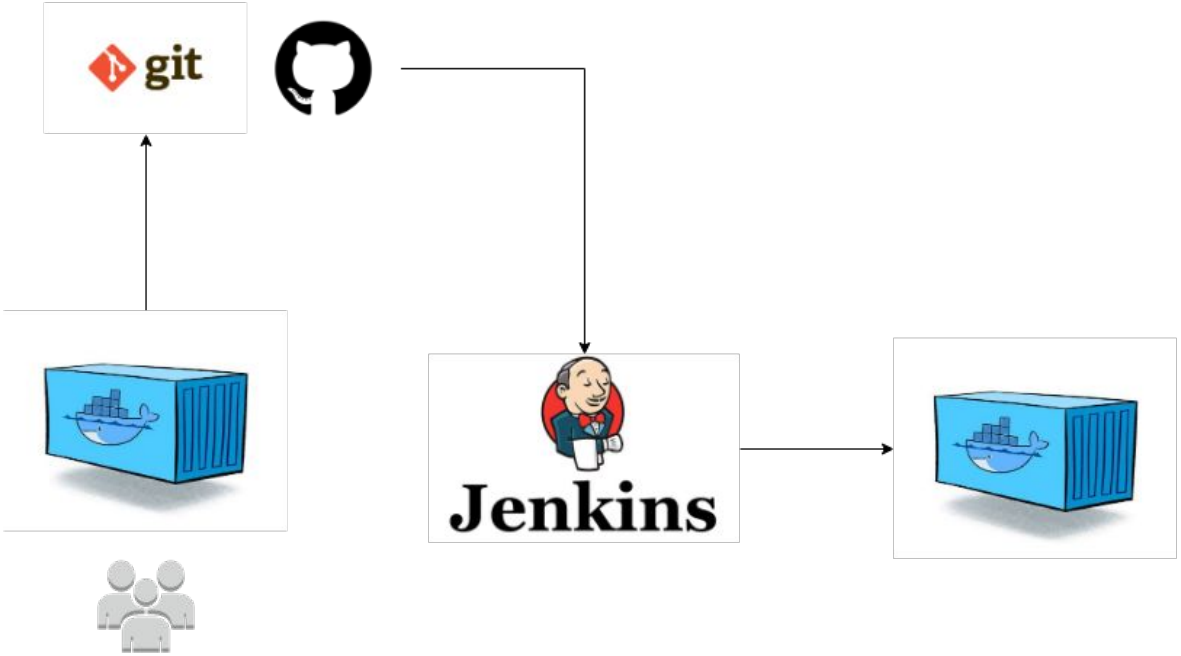
Technologies

Git | Docker | Jenkins

Repos

https://github.com/sergioandresmeneses/DevOps-Workshop-App
https://github.com/sergioandresmeneses/DevOps-Workshop-Infra

WORKSHOP CONTEXT



WORKSHOP CONTEXT

```
\SaMe @ \Globant ~/SandBox/final/DevOps-Workshop-Infra (master)
└─ $ 🐳 ▶ tree
.
├── bootstrap.sh
├── Data
│   └── Dockerfile
├── DevOps-Workshop-App
│   ├── app
│   │   ├── index.html
│   │   ├── js
│   │   │   └── index.js
│   │   └── styles
│   │       └── styles.css
│   ├── default.conf
│   ├── docker-compose.yml
│   ├── Dockerfile
│   └── README.md
├── docker-compose.yml
├── Jenkins
│   └── Dockerfile
├── log-file.txt
├── README.md
├── ubuntu-bionic-18.04-cloudimg-console.log
├── vagrant
└── Vagrantfile
```

```
\SaMe @ \Globant ~/SandBox/final/DevOps-Workshop-App (master)
└─ $ 🐳 ▶ tree
.
├── app
│   ├── index.html
│   ├── js
│   │   └── index.js
│   └── styles
│       └── styles.css
├── default.conf
├── docker-compose.yml
├── Dockerfile
└── README.md
```

LINKS LIBRARY

- DevOps [1]
- CI/CD [1][2].
- Infrastructure as Code [1][2].
- Docker [1][2][3].
- Jenkins[1][2].

Thank You