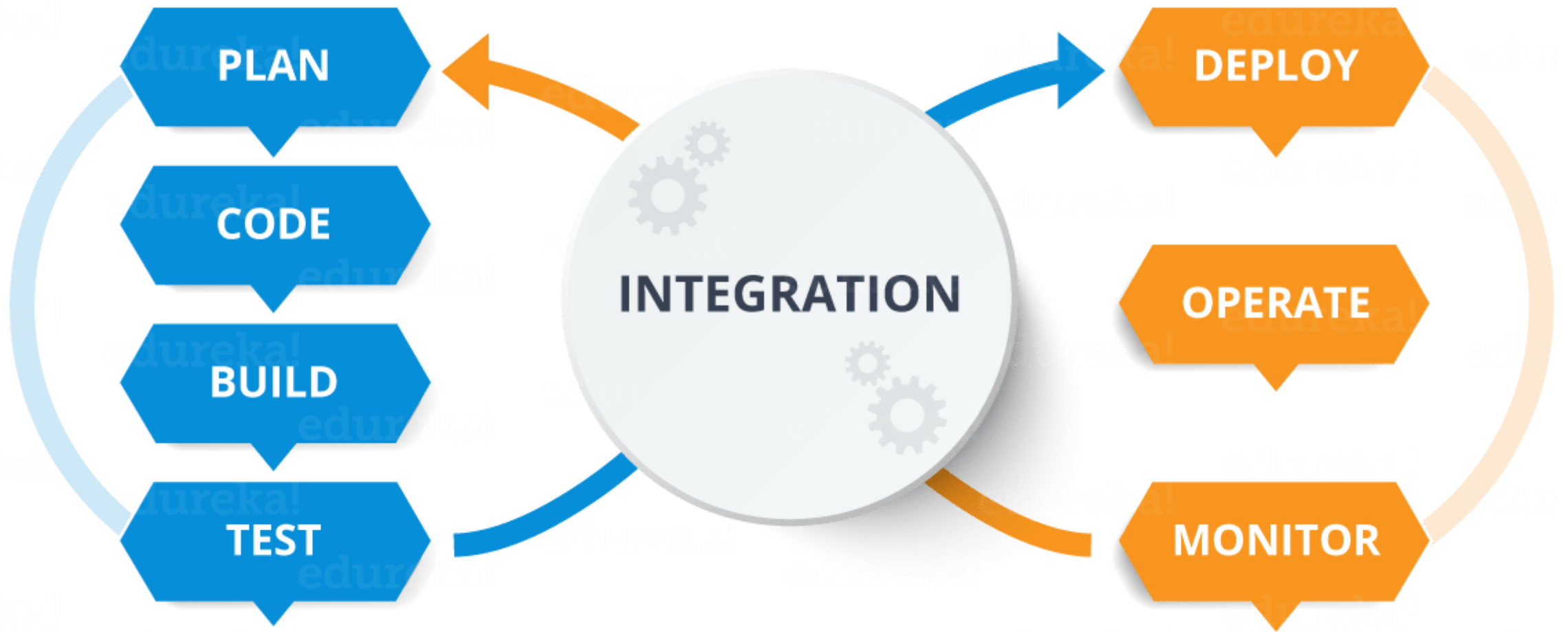
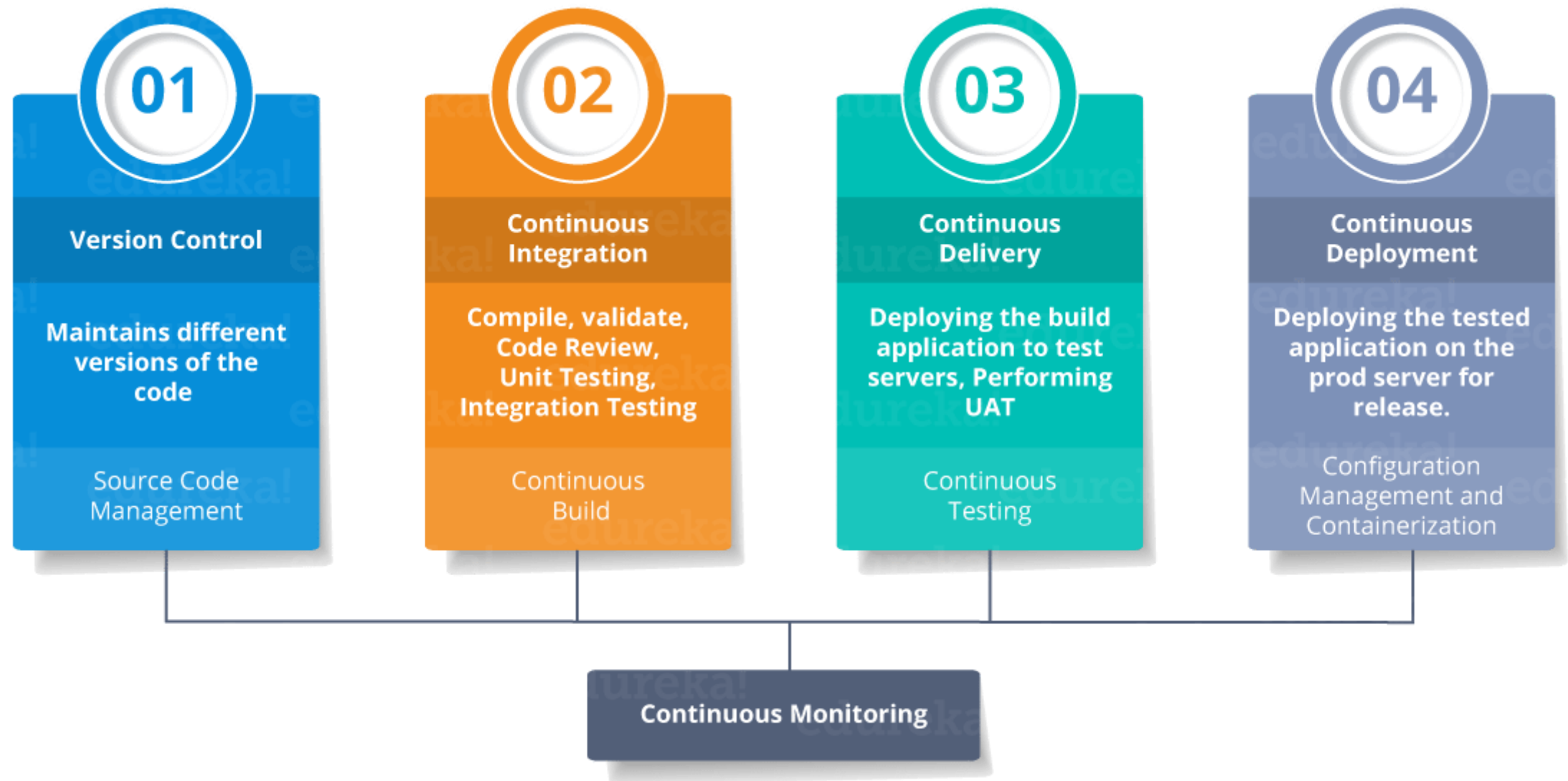
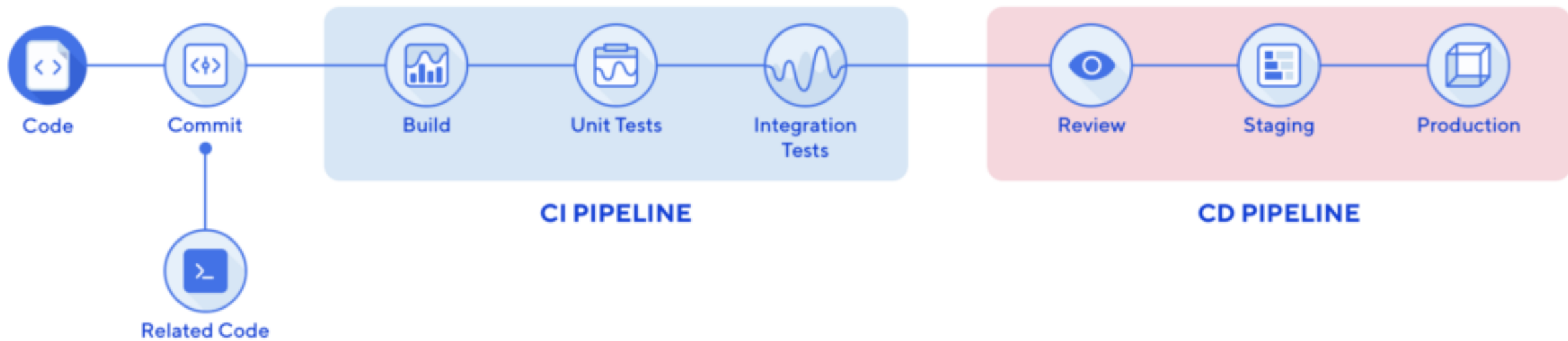


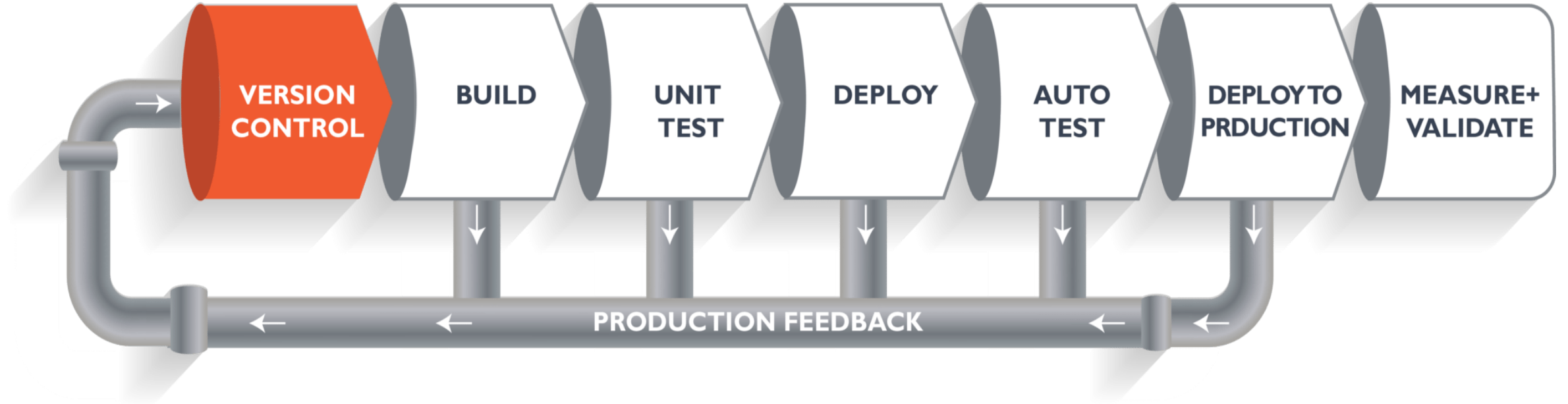
DevOps ~ CI/CD

Docker ~ Kubernetes









Introduction à Git

- Initialement dédiés à la gestion de code source pour les projets logiciels
- mais également :
 - Documentation
 - site web
- travail collaboratif :
 - facilité d'échange
 - traçabilité
 - gestion des conflits

Evolution

- **Systèmes centralisés**

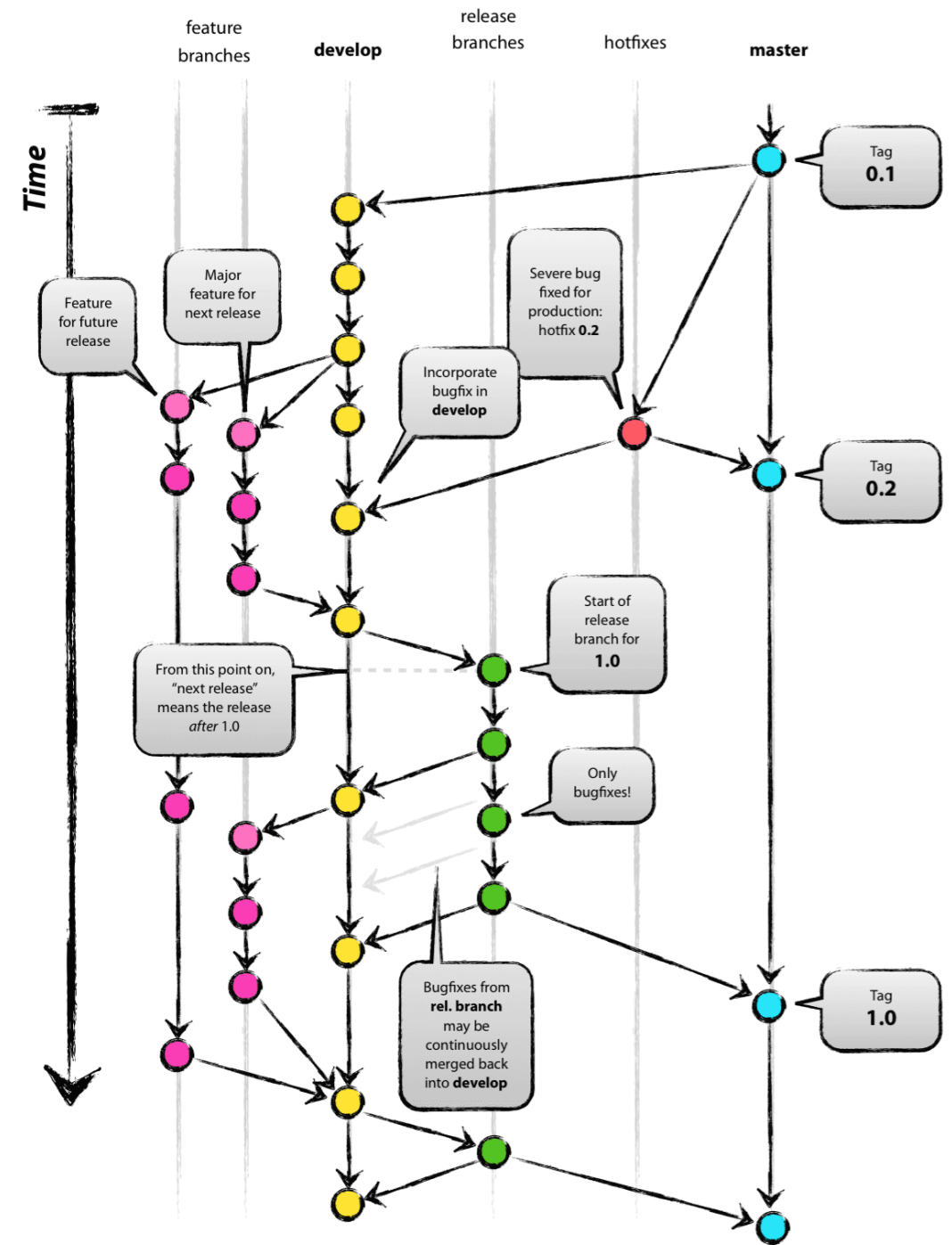
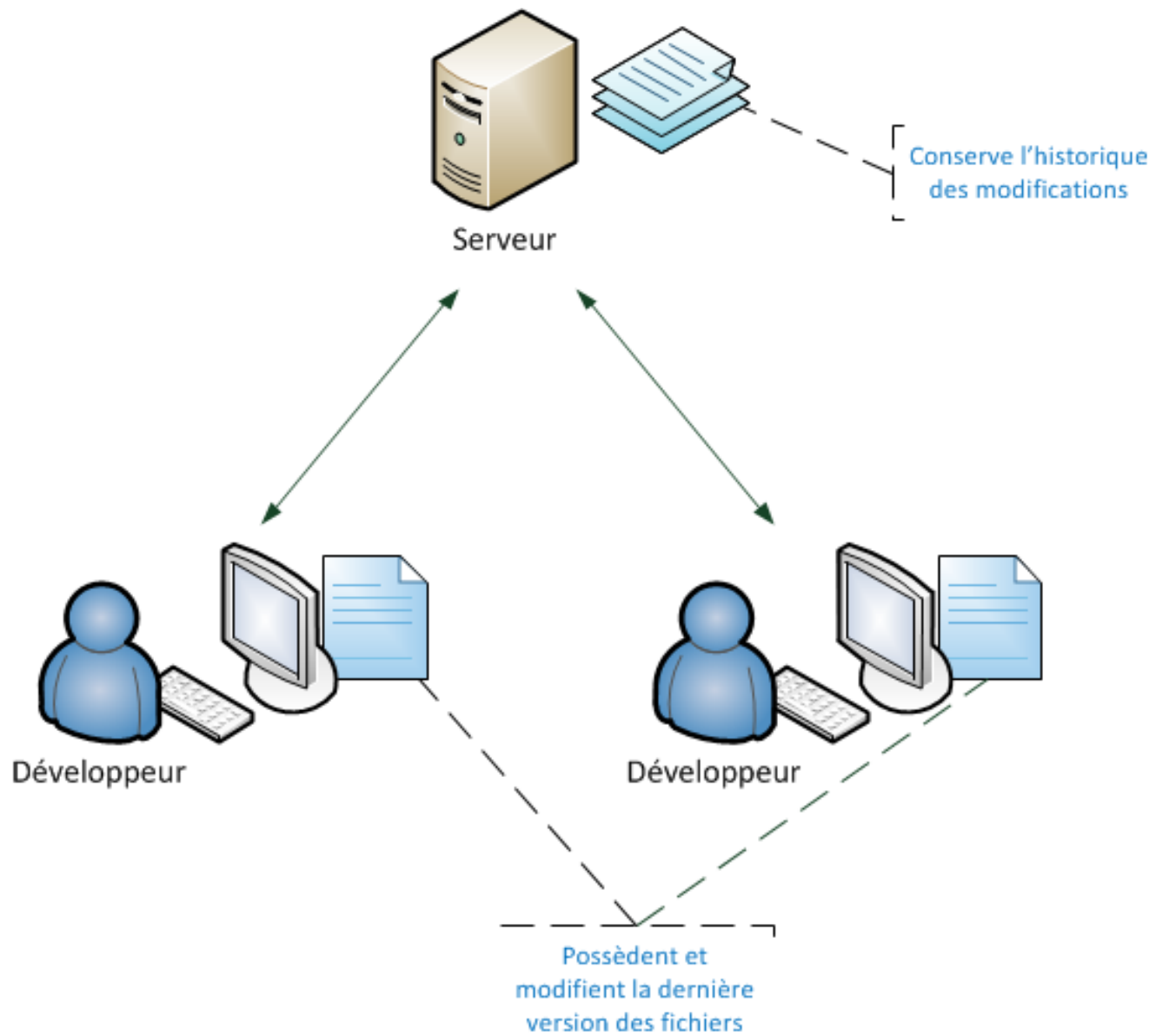
- [CVS](#) (Concurrent Versioning System, vieillissant)
- [SVN](#) (Subversion, très populaire, mais c'est en train de changer)

- **Systèmes décentralisés**

- [GIT](#)
- [Mercurial](#) (Hg)
- [Bazaar](#) (bzd)

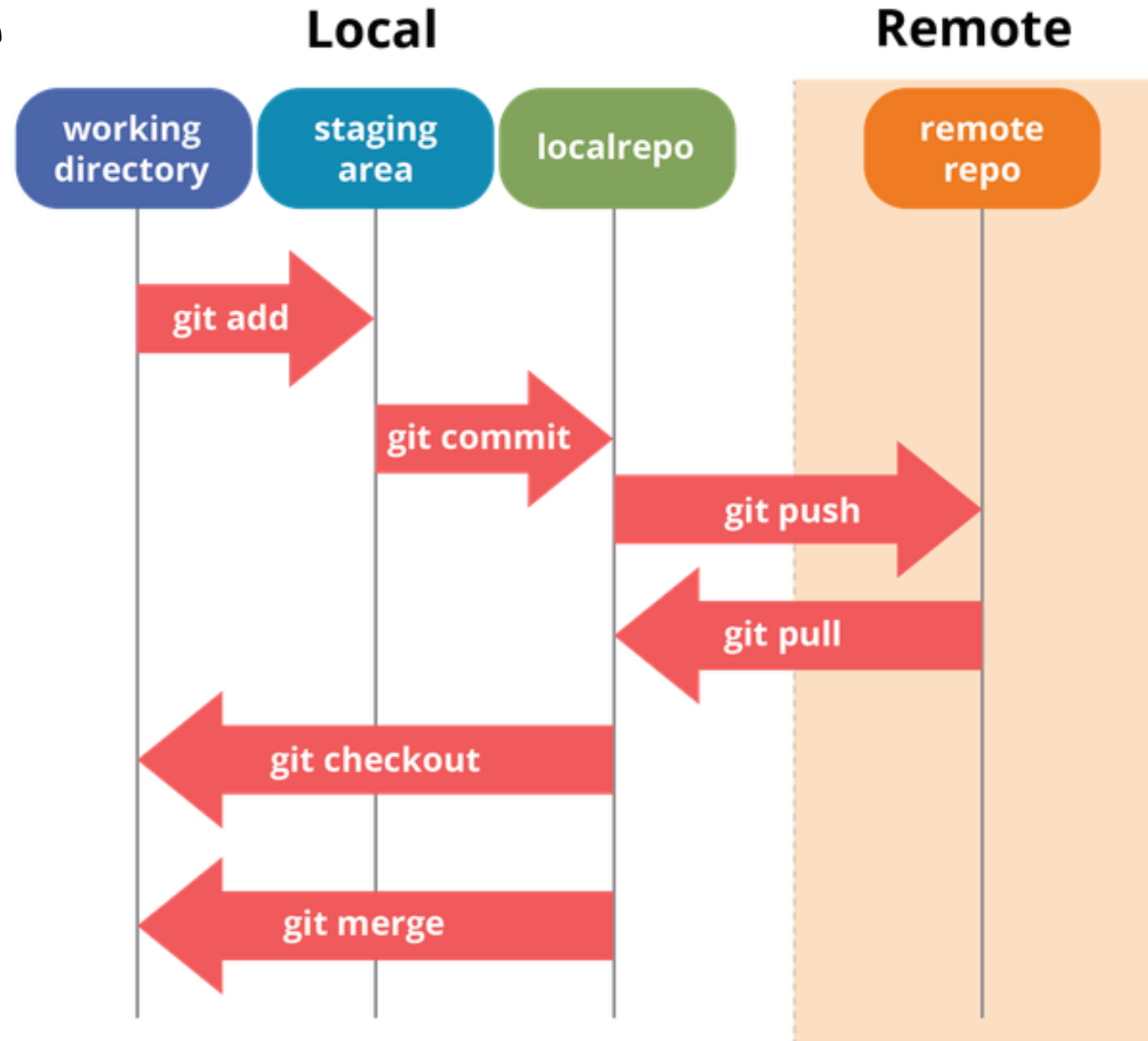
Les avantages de la gestion de versions

- Sauvegarde (modulo la synchronisation avec un serveur distant)
- Conservation de l'historique (nominatif) des fichiers (qui a fait quoi ?)
- Possibilité de retour en arrière
- Fusion des modifications lors du travail collaboratif
- Visualiser les changements au cours du temps



Commandes de base

- git init
- git status
- git add .
- git commit
- git push
- git diff --cached
- git log
- git branch
- git checkout
- git pull
- git merge
- git clone



Search Pull requests Issues Marketplace Explore

poychang / github-dark-theme
vwareia/github-dark-theme

Code Issues 5 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Branches master * github-dark-theme / src / background.ts Find file Copy path

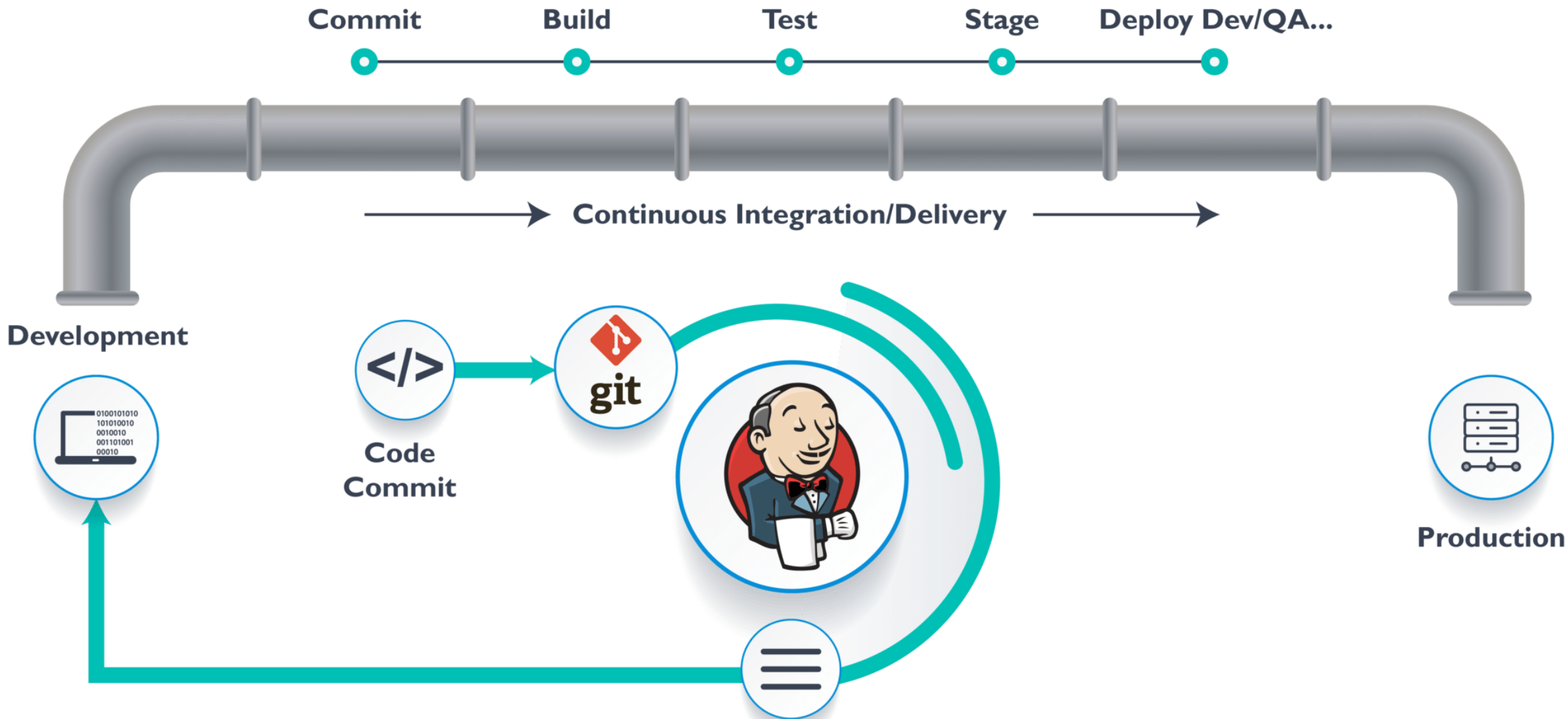
poychang - use rigorous way to compare domain #38 a94fb8c · 25 days ago

1 contributor

56 lines (48 sloc) | 1.72 KB Raw Blame History

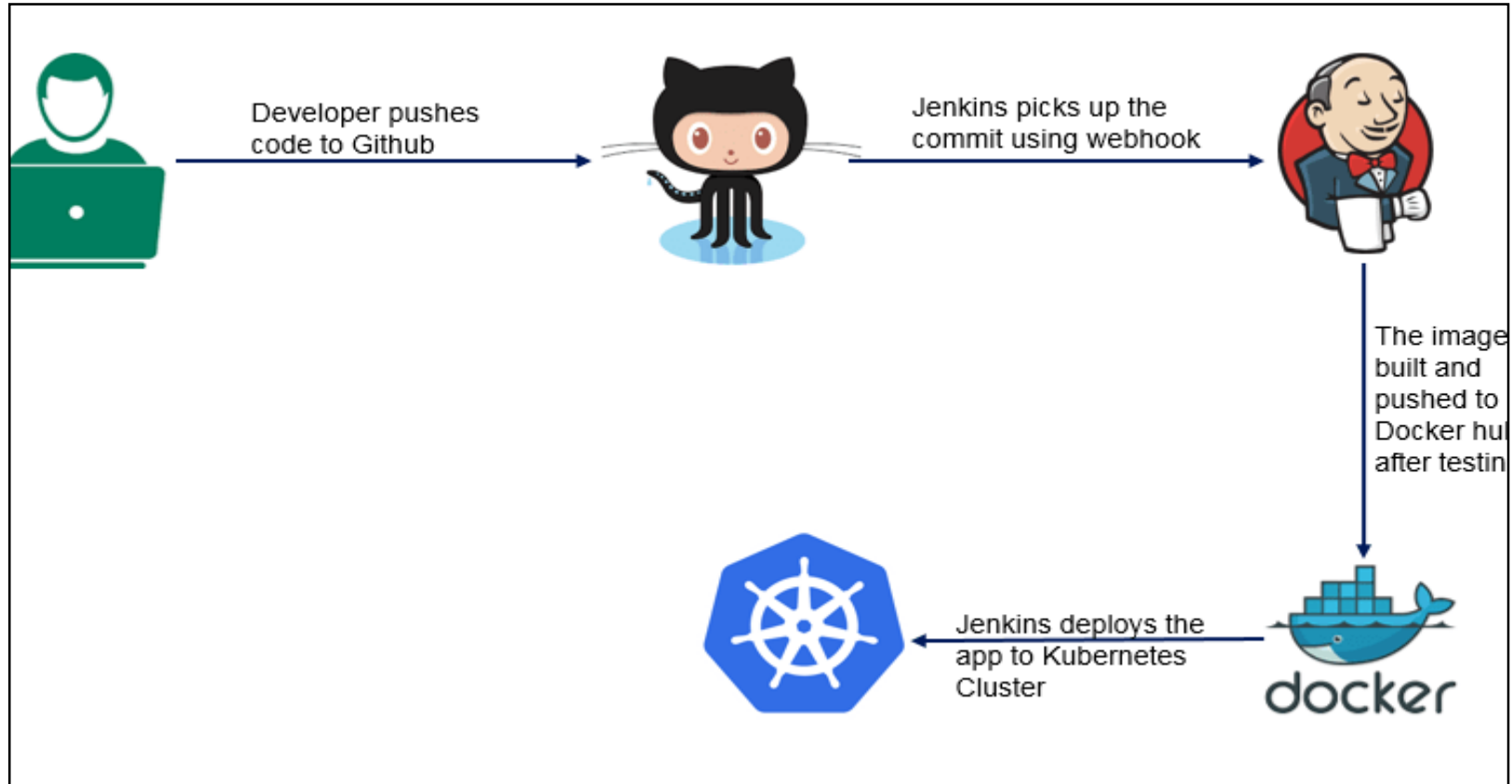
```
1 import { config } from './config';
2 import { isEmpty, fetchDomainString, fetchUrlString, isUrlInList, storage } from './libs';
3
4 const initGithubDarkTheme = (domainList: string[]) => {
5   console.log(domainList);
6   chrome.tabs.getCurrent(tab => {
7     if (!tab) return;
8     if (!tab.url) return;
9     if (isUrlInList(fetchUrlString(tab.url), config.excludeUrlList)) return;
10    if (isUrlInList(fetchDomainString(tab.url), domainList)) insertCSS(tab.id);
11  });
12 };
13
14 const addDomainListener = () => {
15   chrome.tabs.onUpdated.addListener(({tabId, changeInfo, tab}) => {
16     if (!tab) return;
17     if (!tab.url) return;
18     if (isUrlInList(fetchUrlString(tab.url), config.excludeUrlList)) return;
19
20     storage.sync.get(config.storage.nameOfDomainList).then(data => {
21       console.table('Domain list', data.domainList);
22     });
23   });
24 }
```















Jenkins

Jenkins est un serveur d'automatisation gratuit et open source. Il aide à automatiser les parties du développement logiciel liées à la construction, aux tests et au déploiement, facilitant l'intégration continue et la livraison continue. Il s'agit d'un système basé sur un serveur qui s'exécute dans des conteneurs de servlet tels qu'Apache Tomcat.



Jenkins ▶

-  New Item
-  People
-  Build History
-  Manage Jenkins
-  My Views
-  Open Blue Ocean
-  Lockable Resources
-  Credentials
-  Bitfile releases
-  New View

Build Queue

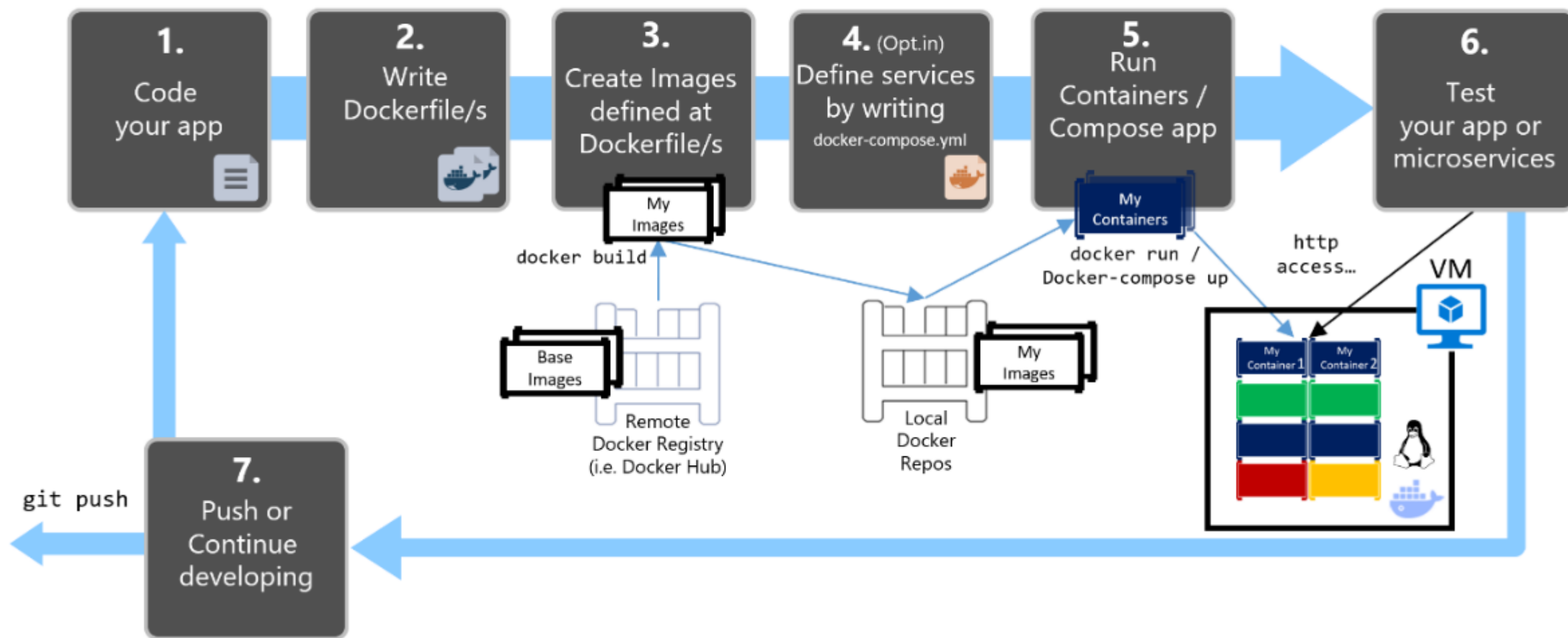
No builds in the queue.

All

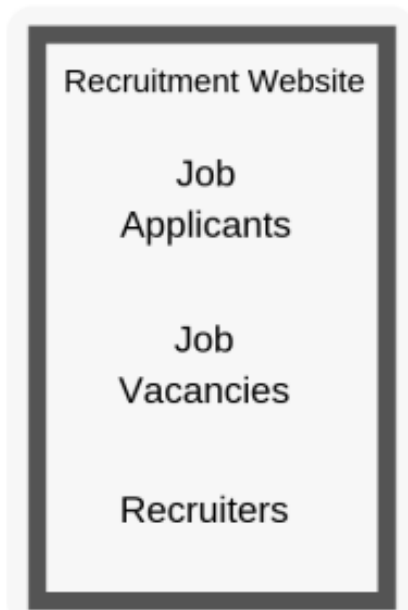
+

S	W	Name ↓	Last Success	Last Failure	Last Duration
		bcd_encoder	23 hr - #15	23 hr - #14	57 sec
		counter	22 hr - #3	22 hr - #2	51 sec
		digit_selector	22 hr - #7	22 hr - #6	52 sec
		output_mux	21 hr - #5	22 hr - #4	53 sec
		packages	3 days 20 hr - #30	3 days 20 hr - #29	44 sec
		reset	20 hr - #3	20 hr - #2	51 sec
		seg7_encoder	20 hr - #3	20 hr - #2	57 sec
		seg7_top	11 hr - #5	11 hr - #4	4 min 28 sec

Inner-Loop development workflow for Docker apps



Monolithic Application



Transition to Microservices



Docker

Create containers for your application



Kubernetes

Launch your containerised application in K8s



Docker

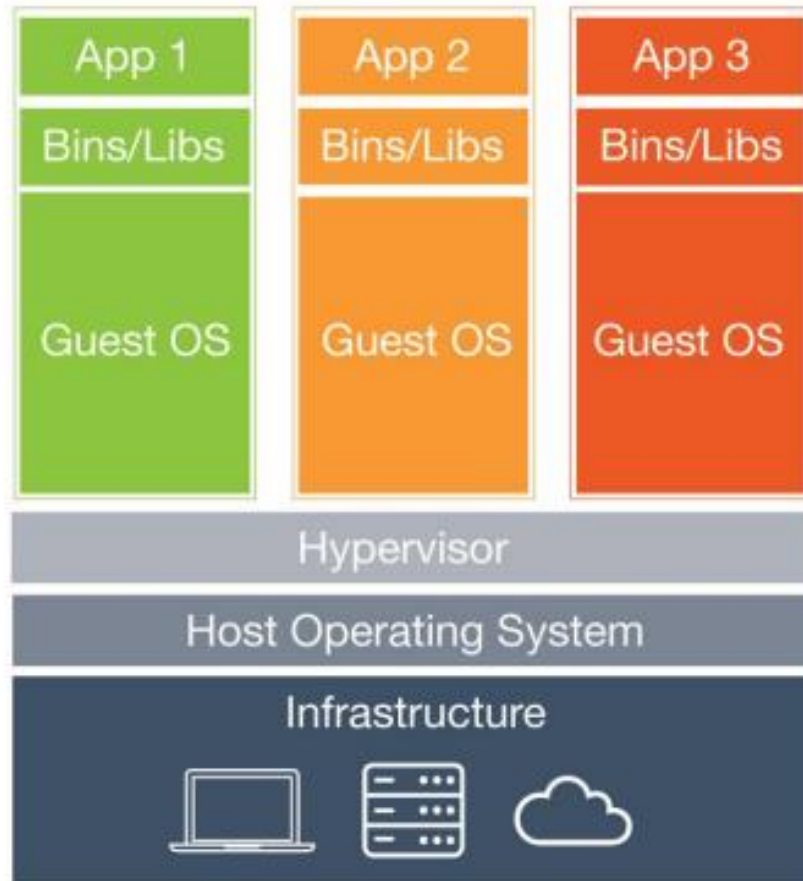
But de la Conteneurisation

- Isoler un processus et ses dépendances dans une unité auto-contenue (conteneur)
- Imposer des limites de ressources (CPU, RAM, I/O, etc) à ce processus
- Isolation des processus: diminution de la surface d'attaque (sécurité accrue)

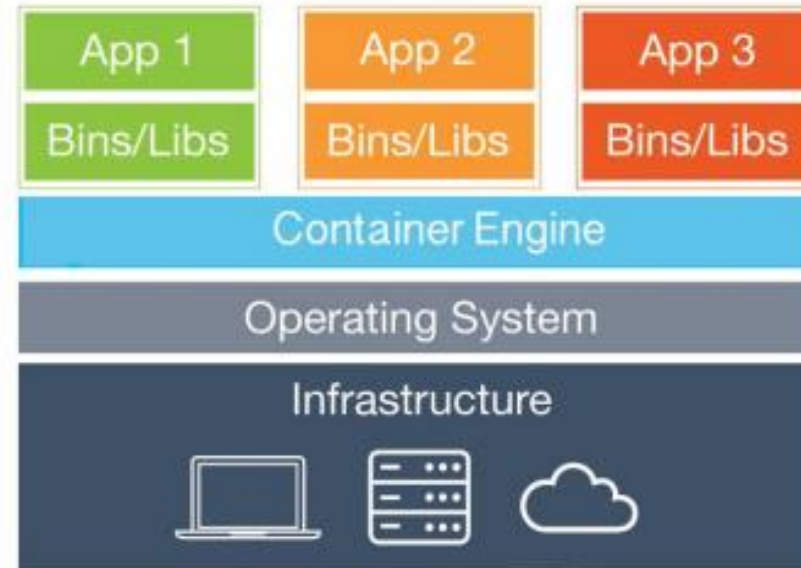
• Isolation

- Contrôle et limitation de la consommation de ressources (CPU, RAM, I/O)
- Isolation réseau (IP, routage, firewall unique par conteneur)
- Isolation filesystem
- Isolation des utilisateurs/groupes
- Isolation des process

Conteneur != VM: architecture



Hypervisor-based Virtualization



Container-based isolation

Architecture micro-service & Docker

- Un micro-service = une image, 1+ conteneur
- Langage de programmation (potentiellement) différent par service
- Equipe (potentiellement) différente par service
- Cycle de vie et de déploiement différent par service
- Scaling par service, en fonction de la charge
- Communication inter-service via des API (REST/SOAP) ou protocoles tels que gRPC

FROM ubuntu

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/web/log/apache2
ENV APACHE_PID_FILE /var/run/apache2.pid
ENV APACHE_RUN_DIR /var/run/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2

RUN export DEBIAN_FRONTEND=noninteractive && apt-get
update && apt-get -y -q upgrade && apt-get -y -q install
apache2

EXPOSE 80 443

CMD ["apache2ctl","-D","FOREGROUND"]

FROM python:3.7

RUN mkdir -p /var/docker-example

WORKDIR /var/docker-example

COPY ./ /var/docker-example

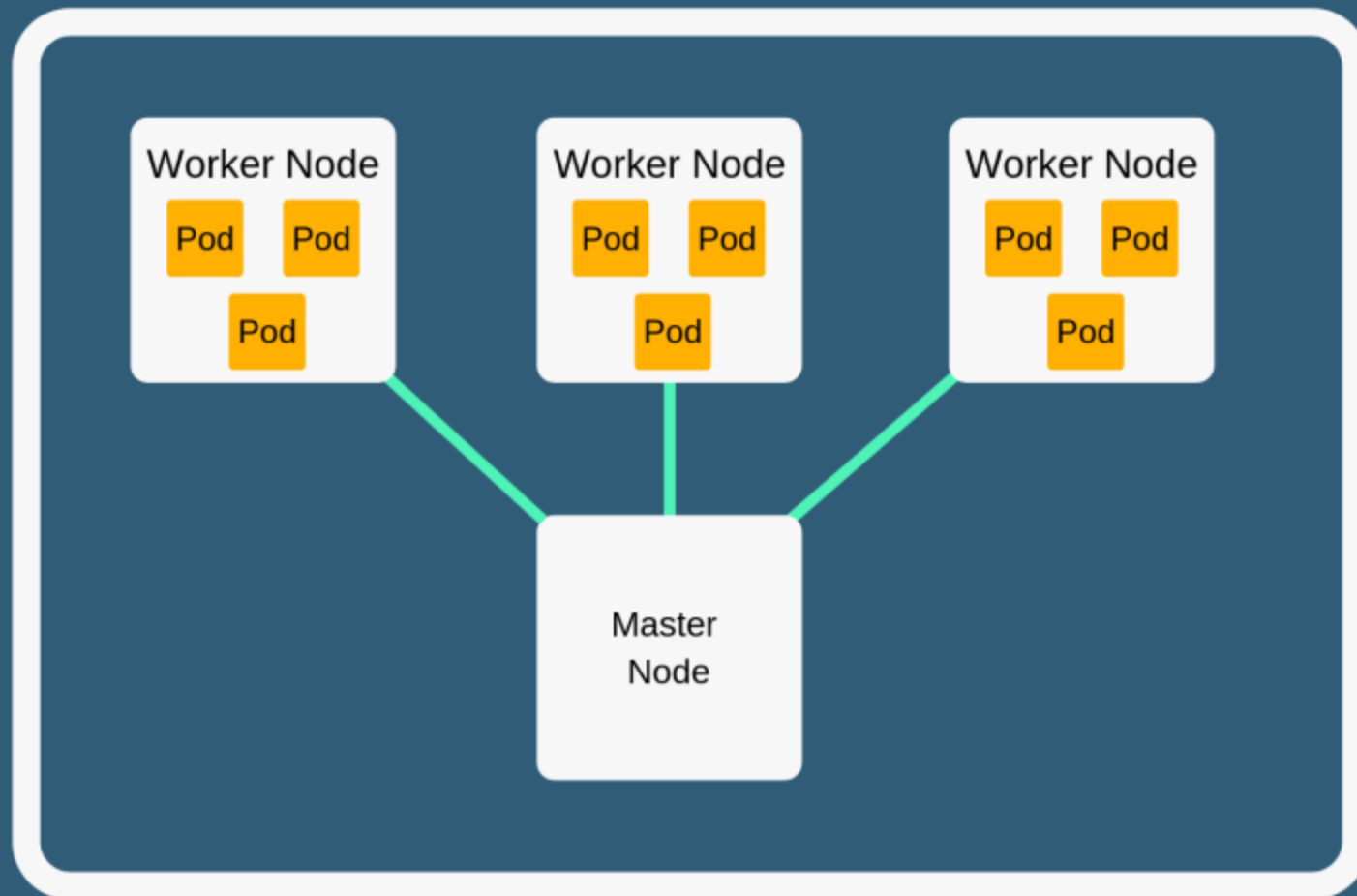
ENTRYPOINT python /var/docker-example/main.py

```
1 FROM ubuntu:latest
2 RUN apt-get update -y
3 RUN apt-get install -y python-pip python-dev build-essential
4 COPY . /app
5 WORKDIR /app
6 RUN pip install -r requirements.txt
7 ENTRYPOINT ["python"]
8 CMD ["app.py"]
```

Orchestrators open source

- Kubernetes (k8s, Google)
- Docker Swarm (Docker Inc)
- Marathon (Mesosphere)
- Rancher (Rancher Labs)

Basic Setup of a Kubernetes Cluster



Sujets de recherche

- ~~Micro service~~
- ~~Machine learning distribué~~
- ~~Calcul distribué~~
- ~~IOT et le Micro service pour la gestion de la mobilité dans les villes intelligentes~~
- ~~Application multitenant~~

Projets

- ~~Création d'une application de gestion de location véhicule~~
 - ~~Client web ou mobile~~
 - ~~Serveur RESTful API connecté à la BD~~
- **Création d'un tableau de bord domotique pour maison intelligentes (Temperature, lumière, capteur présence, ouverture porte)**
 - Client web ou mobile
 - Broker(Mosquito)
 - Un programme python pour chaque capteur
 - Un programme serveur RESTful API connecté à la BD pour l'historique