Organisation

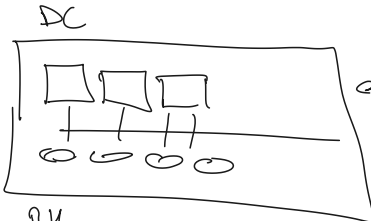
9h — 17h

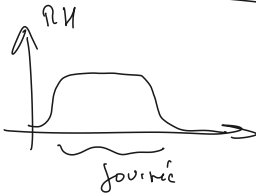
Pause 12h30 à 13h30

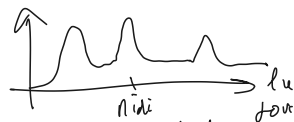
Pause 10h30 / 15min
15h30

Cloud Privé (On Prem)

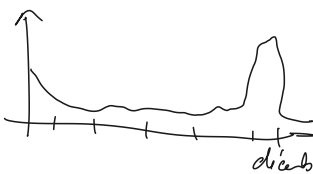
Yellow Cloud → Projet Raison

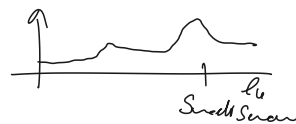






Commerce électronique



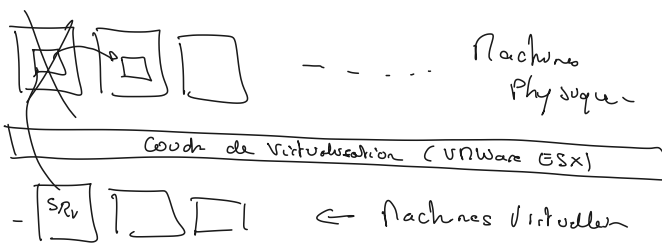


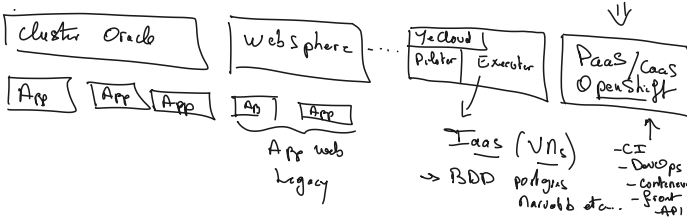
AWS
Azure
GCP
} → Portails, Systèmes "Maison"

Tiers : Orange / OVH } Solution "OpenStack"

La Poste → Yellow Cloud → Solution "Maison"
→ VM → 1er Sujet → automatisé
- autres — automatisé
- manuelle ou façade

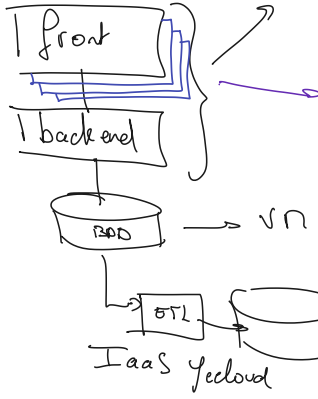
Récap





Archi's nTier

Legacy = WebSphere (Web Archive)



Modern = PaaS

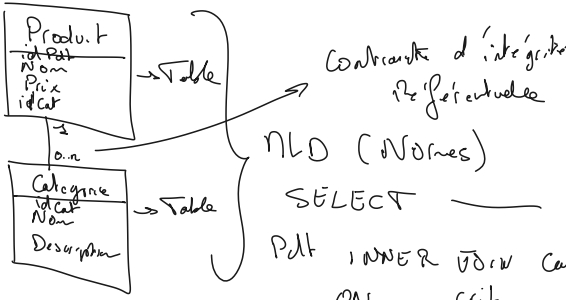
OpenShift (Spring Boot)

VM yellow cloud

Oracle → Solution Technique

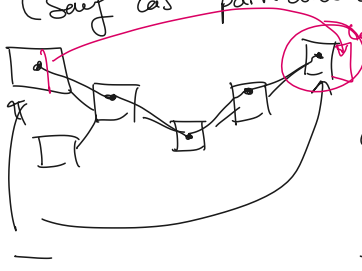
Methode Conceptuelle (Merise)

MCD → MLD → MPD -



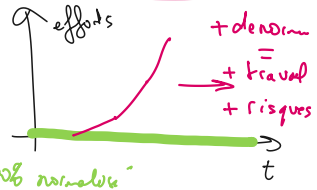
Normalisation est un pattern en relationnel

(Sauf cas particuliers)



Copies (dénormaliser)

Décision



A hauteur d'internet

→ Perf. interdit l'usage de bases SQL

Des spécifiques des stockages

= Bases NoSQL

Objectif = HA

- CouchBase
- MongoDB
- Redis
- Cassandra
- Kafka

Types de charge de Travail

- Stockage en colonnes
- clef/valeur (cache, ...)
- Document
- Graphe
- Messages -

SQL

MA

cluster Cassandra

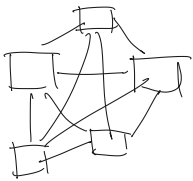
R - W

Collection

P0

R1

R2



interconn

noeuds

SQL

50 noeuds

Risque

130

Report

B1

ETL

Datawarehouse

marts

Wbs

DI

Traditionnelle

(podcast)

front

Redirecte

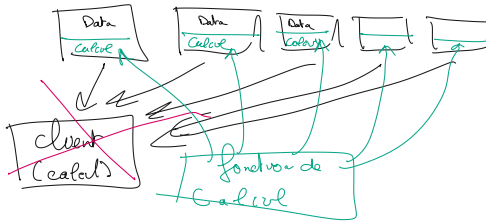
SQL

Batch

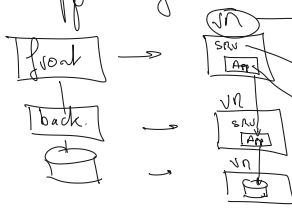
elastic search

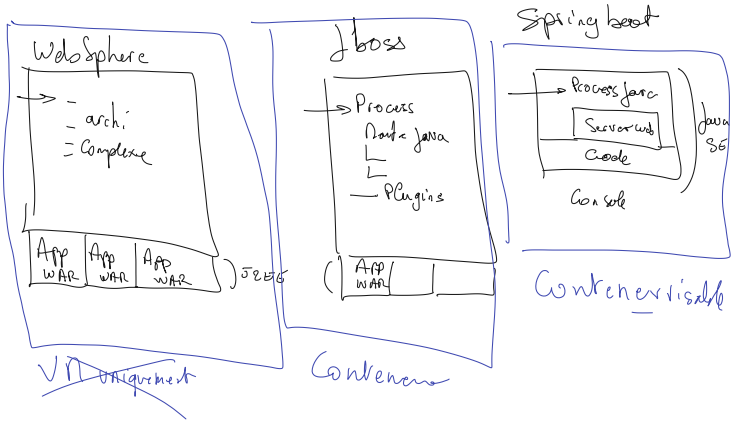
(NoSql)

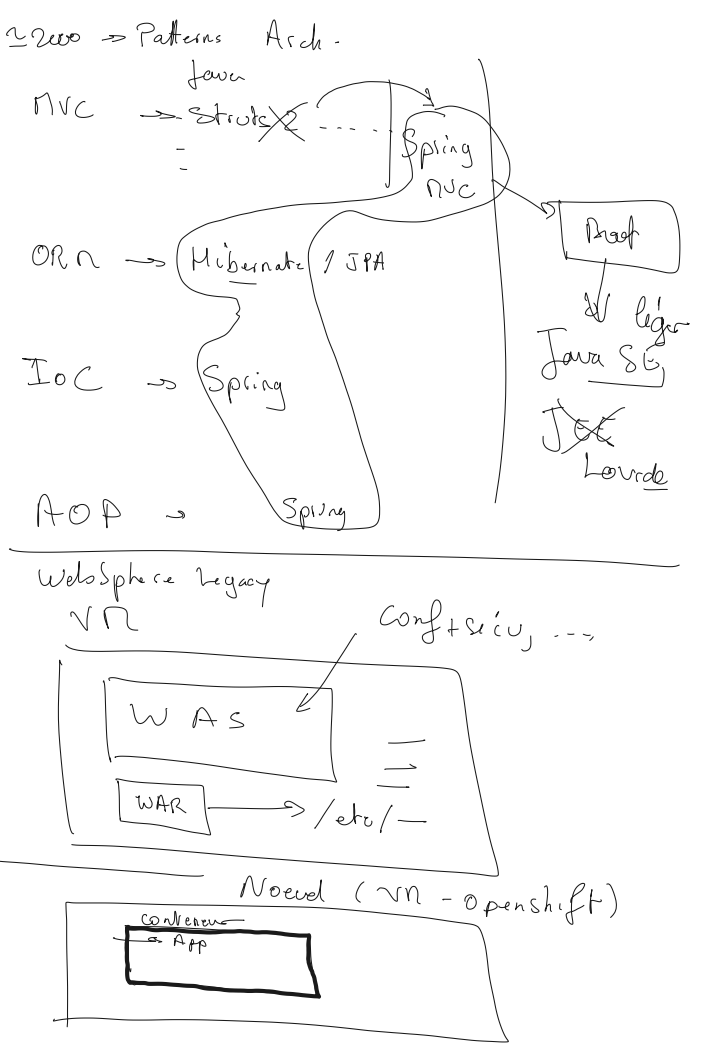
cluster
CouchBase



1 Application = n Couches



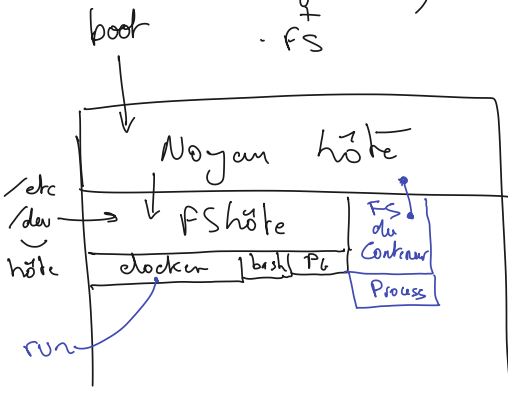


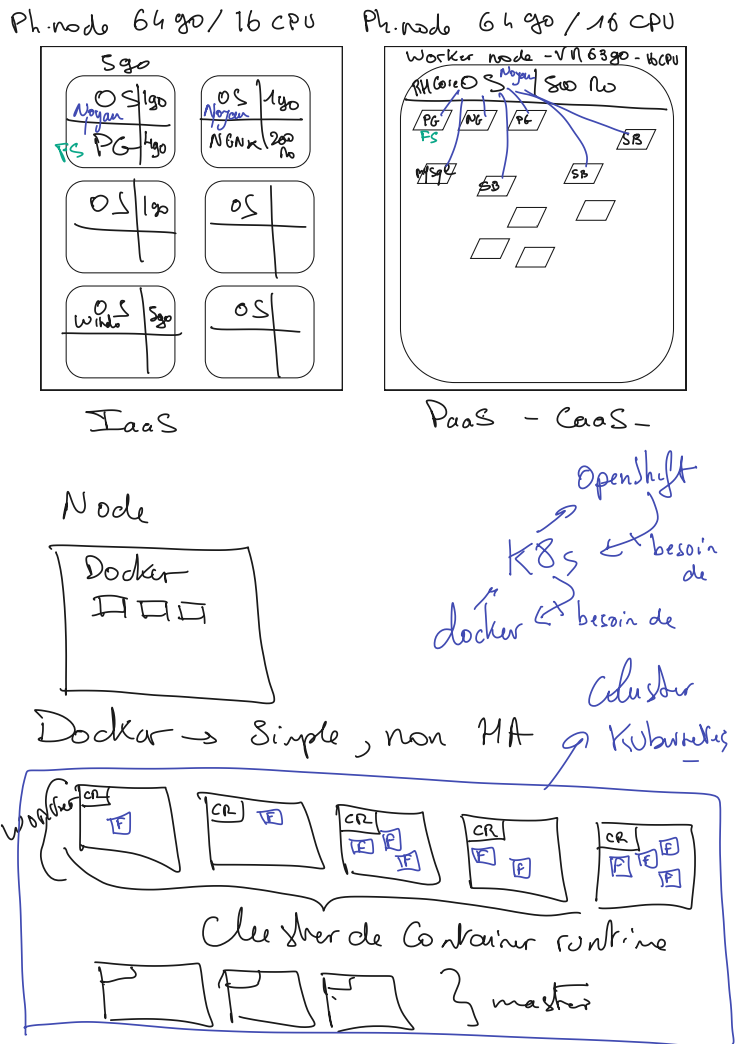

~2000 → Patterns Arch.
Java
MVC
struts
Spring MVC
ORM
Hibernate / JPA
IoC
Spring
AOP
Spring
Proof
W Eclipse
Java SE
JEE
Lourde
WebSphere Legacy
VM
Conf + Scripts ...
WAS
WAR
/etc/...
Novel (VM - Openshift)
container
App

OS = Noyau + Filesystem

Ub 22	RHEL
Kernel 5.10	5.10
FS – Ubuntu	FS – RH

Windows-
- Noyau (DLL)
+
- FS

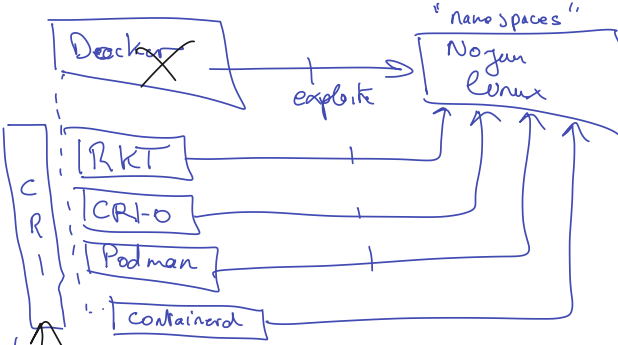



Ph. node 64 go / 16 CPU
Ph. node 64 go / 16 CPU
5go
OS 1go
Noyau
PG 4go
FS
OS 1go
Noyau
NGNX 200 Mo
OS 1go
OS
OS
Windows 5go
OS
Worker node - VM 63go - 16CPU
Noyau
500 Mo
PG
FS
PG
PG
SB
SB
SB
IaaS
PaaS - CaaS -
Node
Docker
OpenShyft
K8s
besoin de
docker
besoin de
Cluster
Kubernetes
Docker → Simple, non HA
Worker
CR
IF
Cluster de Container runtime
master

Docker → ≃ 2008 - 2010

→ Réputé

→ Pas standard



K8s → Projet Google

Open source

utilisé à ≃ 95%

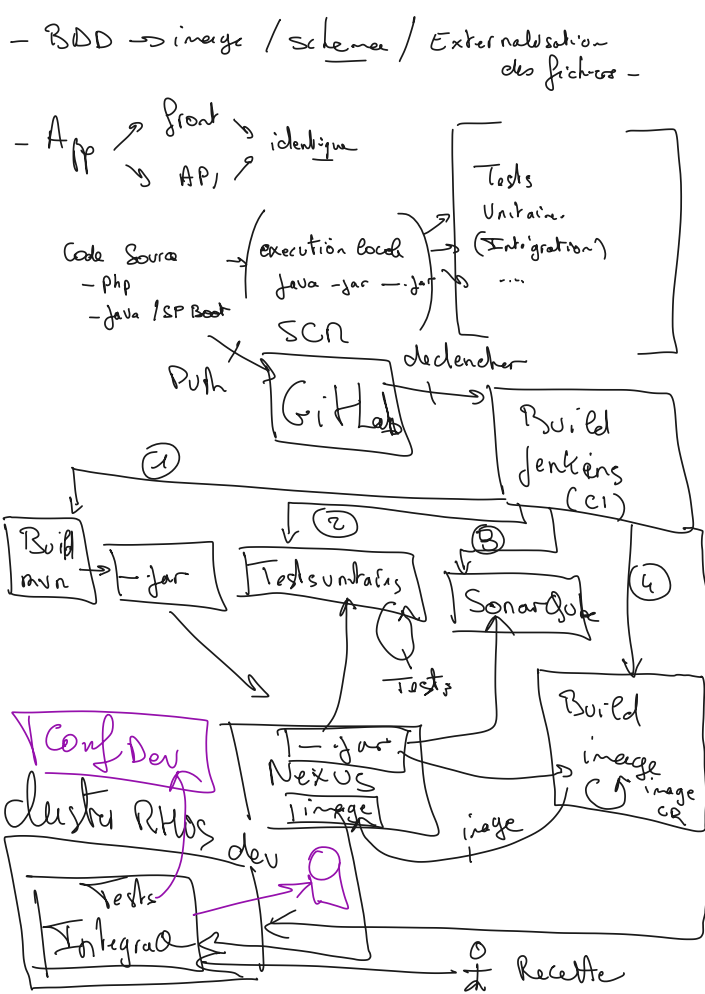
depuis ≃ 2015

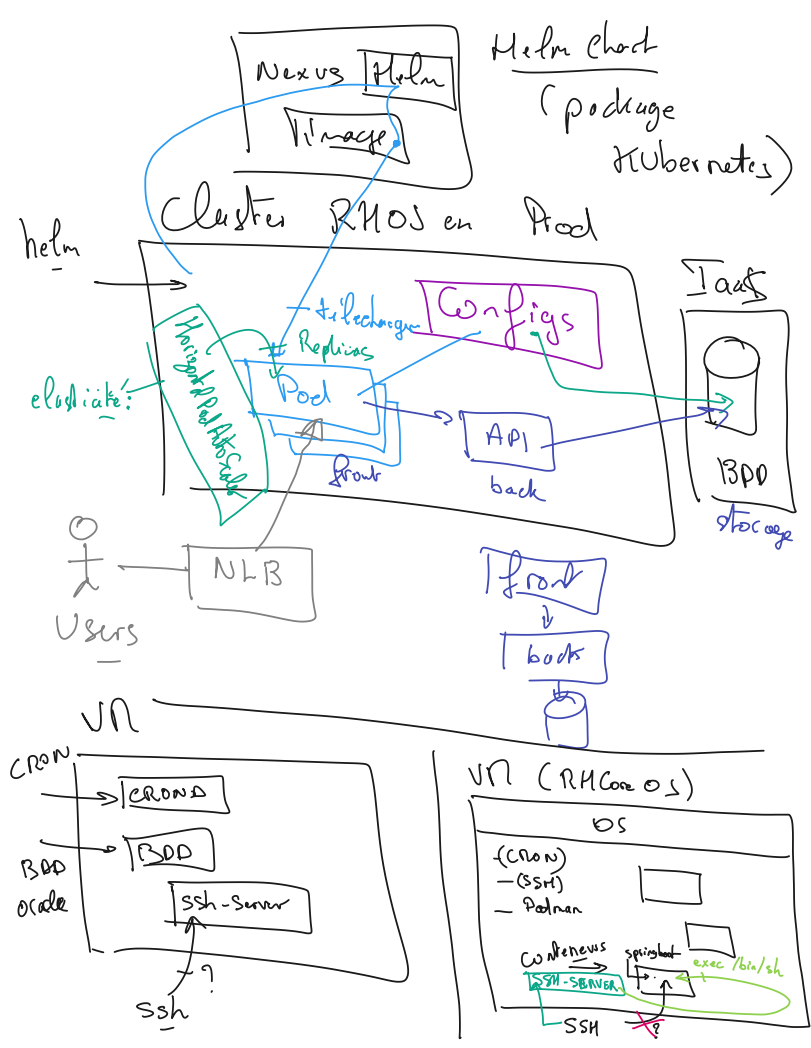
IBM

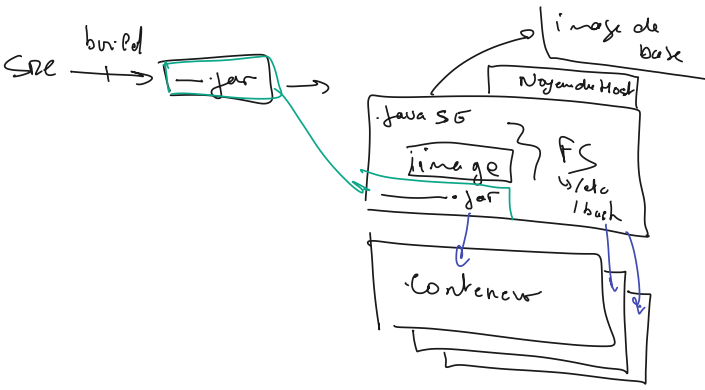
Red Hat

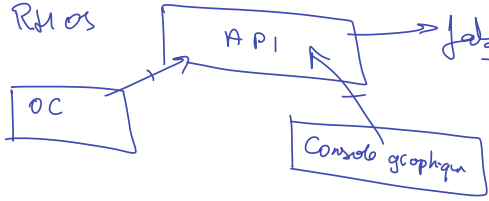
openshift → ~~VM~~

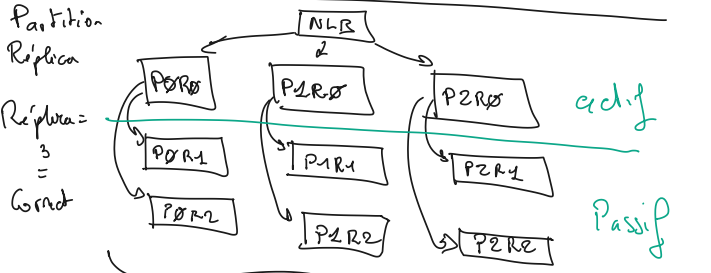
→ Conteneurs


- BDD -> image / schema / Externalisation des fichiers -
- App
front
API
identique
Tests
Unitaire
(Intégration)
....
Code Source
- php
- java / SP Boot
execution locale
java -jar --.jar
SCM
Push
GitLab
declencher
Build
Jenkins
(CI)
1
Build
mvn
-jar
2
Tests unitaires
tests
3
SonarQube
4
Build
image
image
CR
Conf Dev
cluster RHOS dev
-.jar
Nexus
image
image
Tests
Integral
Recette

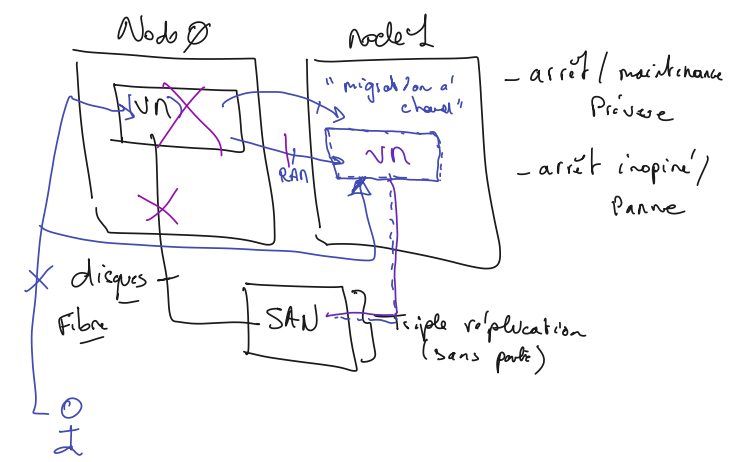

Nexus
Helm
Image
Helm chart
(package Kubernetes)
Cluster RHOS en Prod
helm
Haproxy HPA AutoScale
élasticité
télécharge
Replicas
Pod
Configs
front
API
back
IaaS
BDD
stockage
Users
NLB
front
back
VM
CRON
CROND
BDD
BDD oracle
ssh-server
ssh
?
VM (RHCos OS)
OS
(CRON)
(SSH)
Podman
CoreNews
spring boot
exec /bin/sh
SSH-SERVER
SSH
?

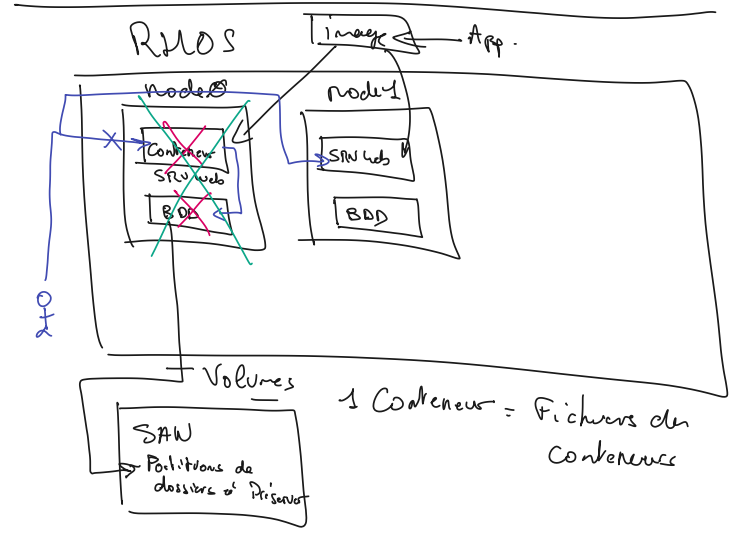


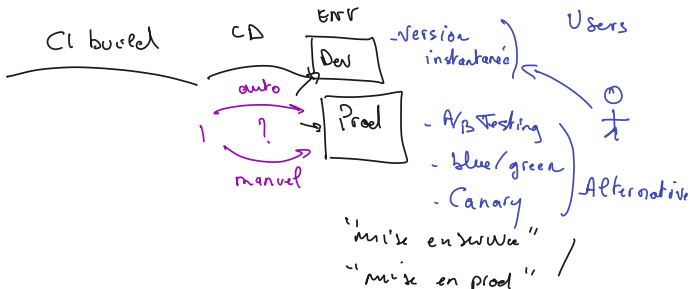


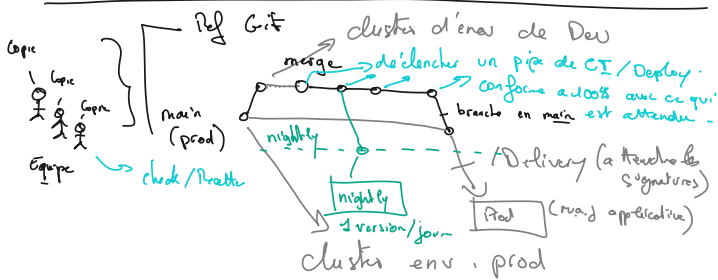


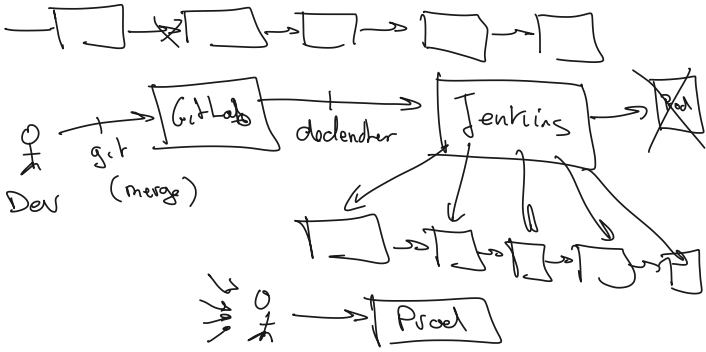
- Synchrone : Pas de perte - implique la disponibilité
- HA de Base de données
- 3 Partitions -
- Asynchrone - Perte - Pas de dépendance -


Node 0
node 1
VM
"migration à chaud"
VM
RAM
- arrêt / maintenance Prévue
- arrêt inopiné / Panne
disques
Fibre
SAN
triple réplication (sans perte)


RMOS
image
App.
node 0
node 1
Conteneur
SRV web
BDD
SRV web
BDD
Volumes
SAN
Partitions de dossiers à préserver
1 Conteneur = Fichiers des conteneurs


CI buscel
CD
Env
Dev
Prod
auto
manuel
?
- version instantanée
- A/B Testing
- blue/green
- Canary
Alternatives
Users
"mise en service"
"mise en prod"


Copie
Copie
Copie
Equipe
Ref Git
main (prod)
check / Rustfix
cluster d'env de Dev
merge
déclencher un pipe de CI / Deploy.
conforme à 100% avec ce qui
branche en main est attendue.
nightly
nightly
1 version/jour
CD Delivery (à travers les signatures)
Prod
(vrai application)
cluster env : prod


GitLab
déclencher
Jenkins
Prod
git
(merge)
Dev
Prod

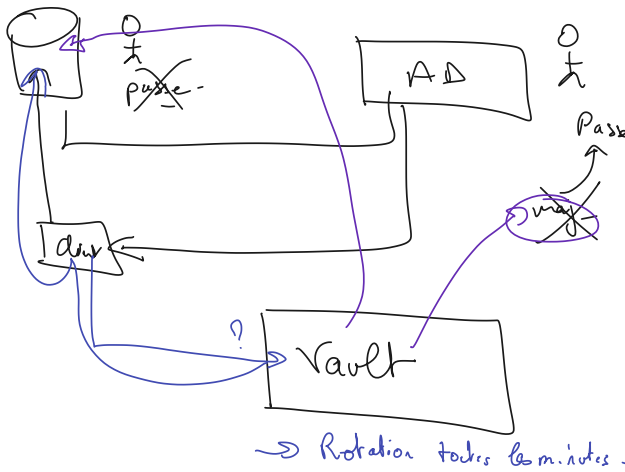
- (Non) Qualité

- (Règles d'entreprise)

analyse statique

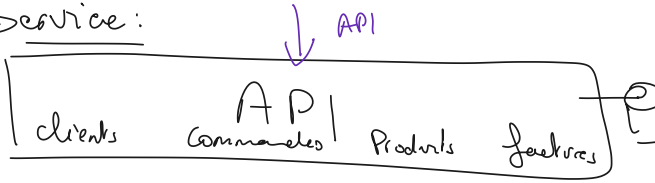
- Risques de bugs / mauvaises pratiques / failles, ...
- Dépendances avec des Vulnérabilités (CVEs)

(- Tests = analyse dynamique)

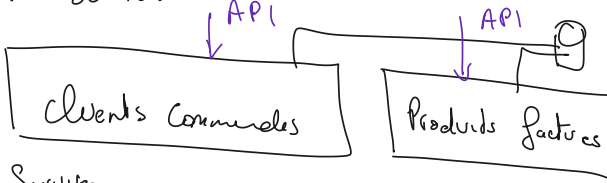


→ Rotation toutes les minutes -

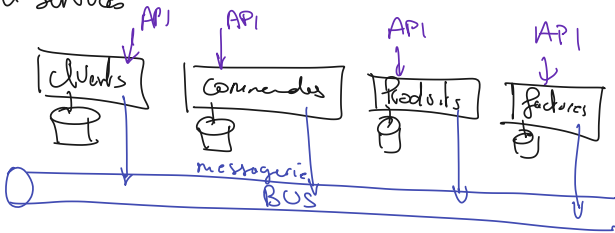
Service :



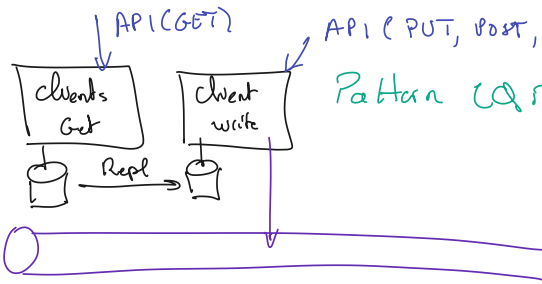
"Mini Services"



Micro Services



Nano Services



Pattern CQRS