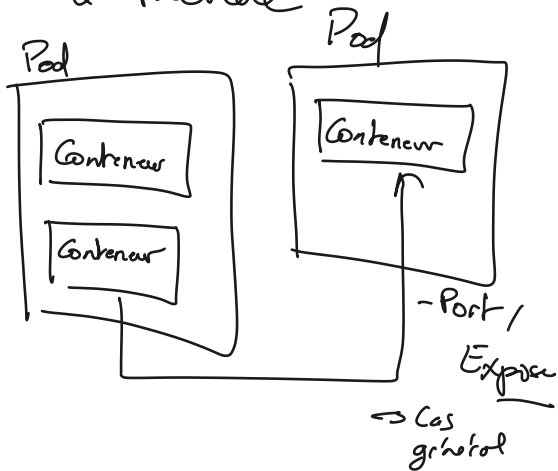
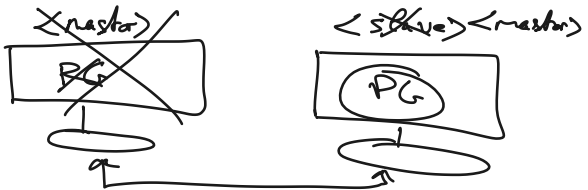
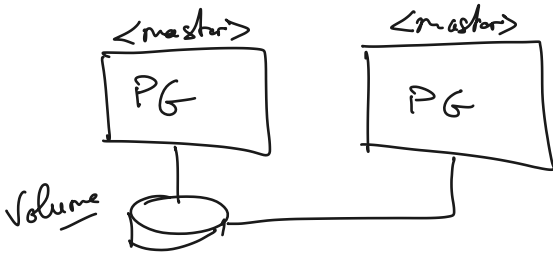
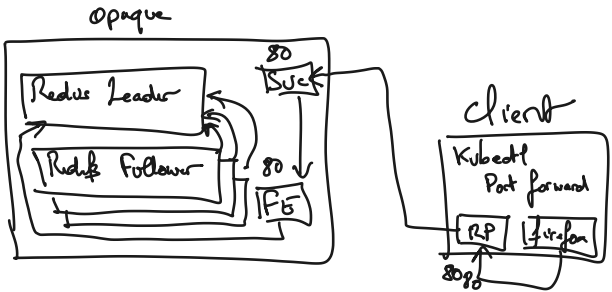


Bonjour tout
le monde






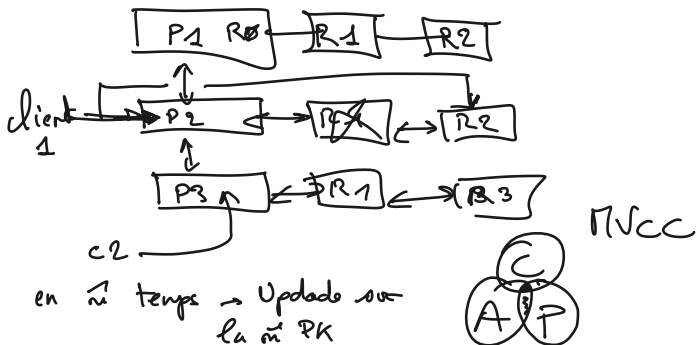
Replica

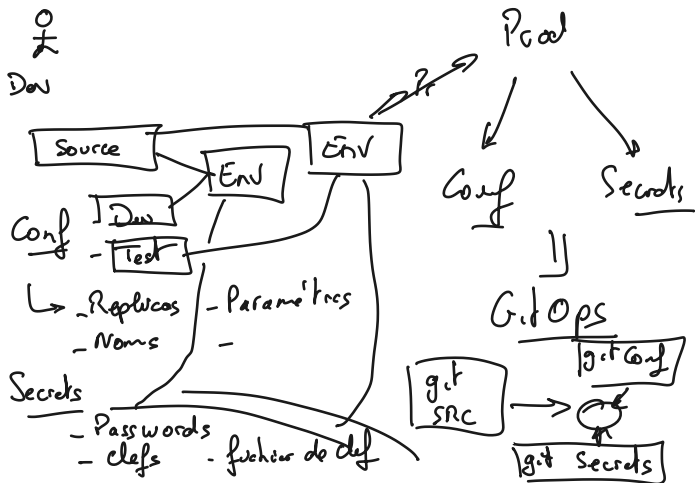
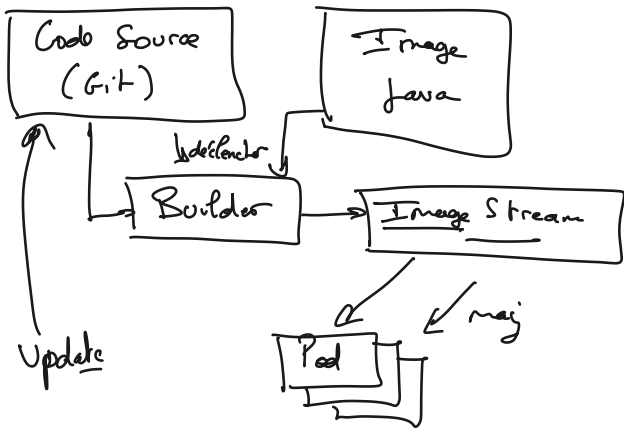
REPLICASET

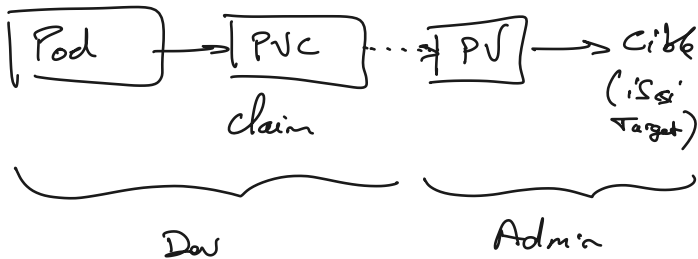


→  Théorème de CAP

cluster NougoDB







Pipeline CI

Helm Kont

CI → Integration Continue

CD → Continuous Deployment
Delivery

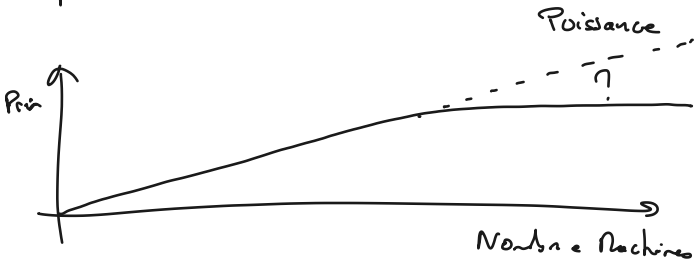
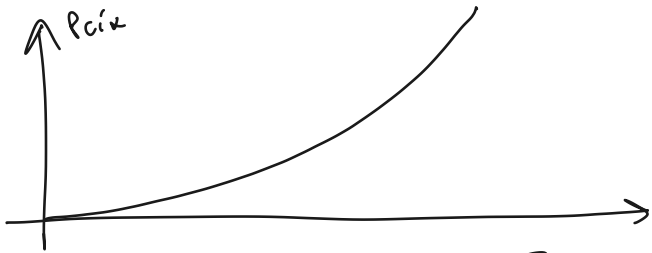
CI → Integration

CD → Deployment
Delivery

Rise on
Prod

CD

→ git / schub / Helm / Tests → /.../image₃ ^{OK} → auto Deployment Delivery -manuelle



↑ Scale Up = machine + puissante

S. In =

Retarder

Scale Out

= ajouter des machines

↓ Scale Down.

Recommando!

→ HA (Partition + Tolérance)

→ moins cher avec volume

Rolling Update

v1.0 → v1.0 → v1.0 → v1.0 → v2.0

v1.0 → v1.0 → v1.0 → α → v1.1

v1.0 → α → v1.1 → v1.1 → v1.1

Rollout Update

v1.0 → α → v1.1

v1.0 → α → v1.1

↓

Coopure de
Service

Worker Nodes



name=n0

name=n1

n2

n3

n4

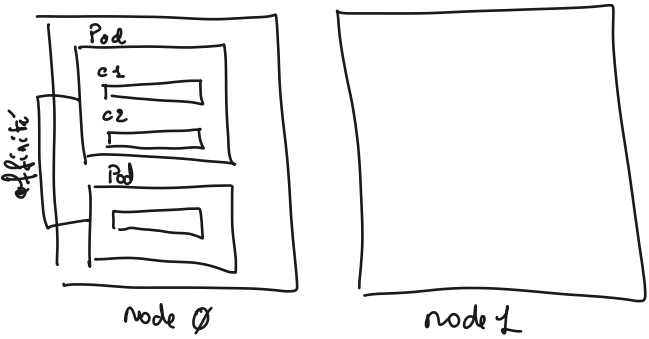
disk=ssd

disk=hd

gpu=0

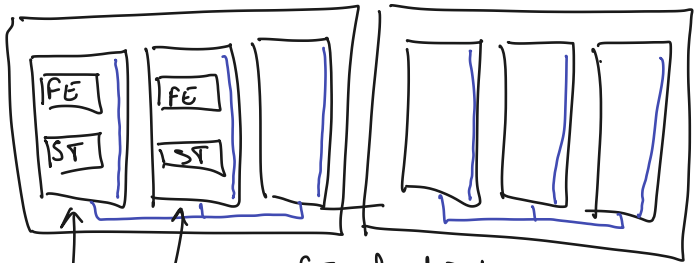
gpu=1

gpu=0



DC 0

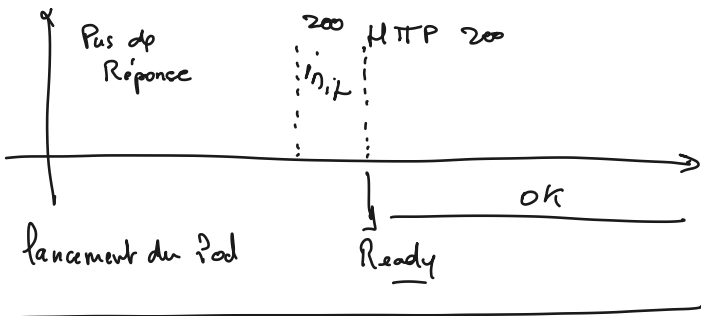
DC 1



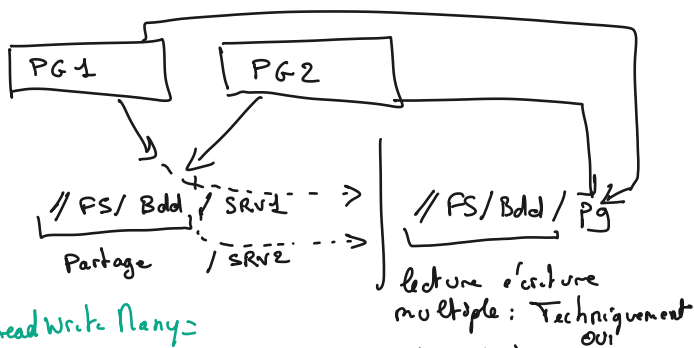
FE frontEnd
ST storage (NoSql)

Recommandation:

- Positionner couches dans le rack → *Affinité*
- Deployer au maximum inter Rack.



Niveau file system

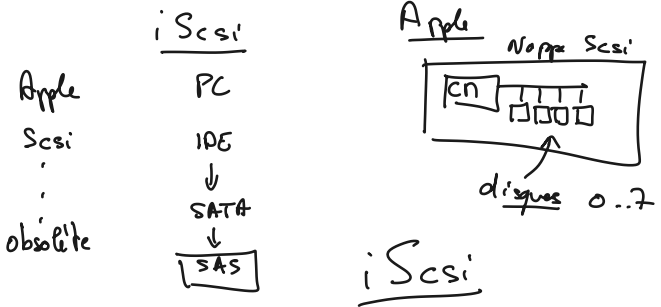
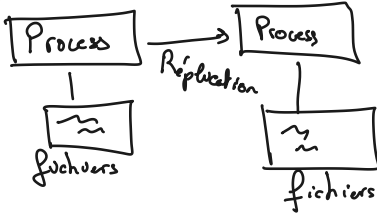


ReadWriteMany:
cas très spécifique -

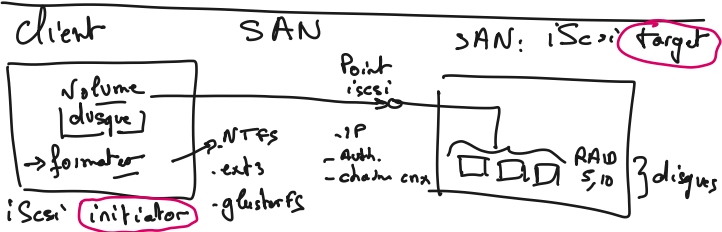
Pour postgres (ex): Par contre
erreurs de lecture/écriture.

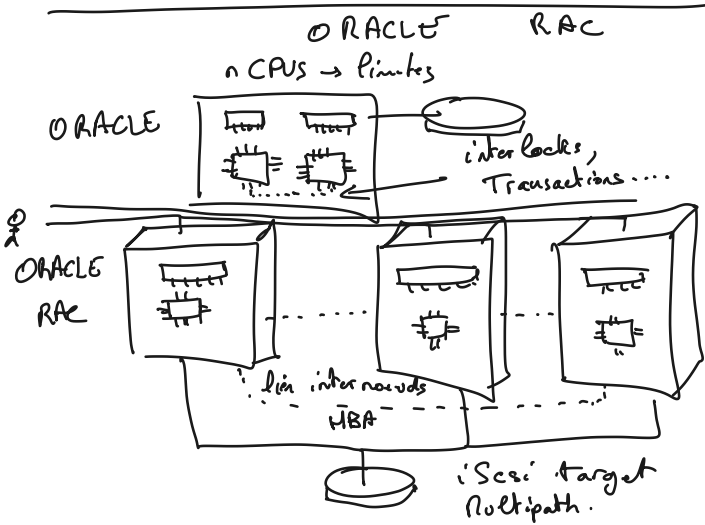
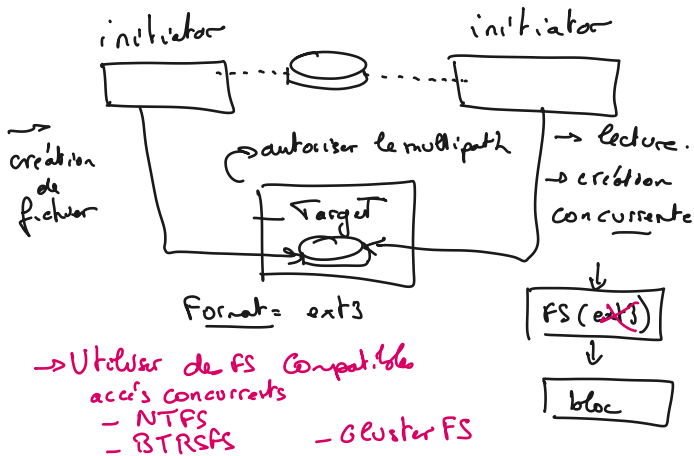
Cas de Oracle RAC (Real Application Cluster)

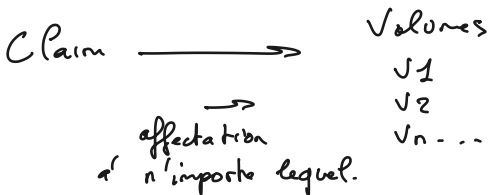
Oracle Simple:



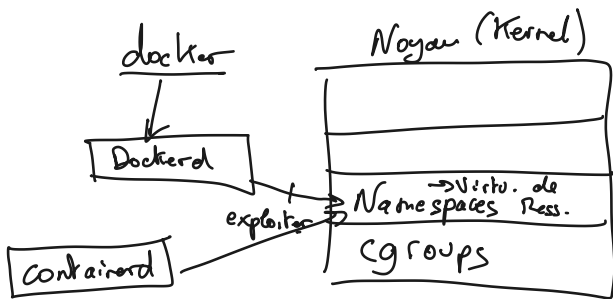
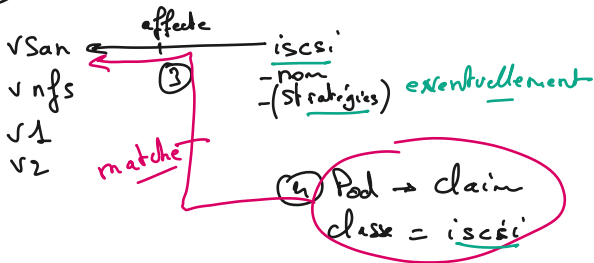
Scsi over Internet



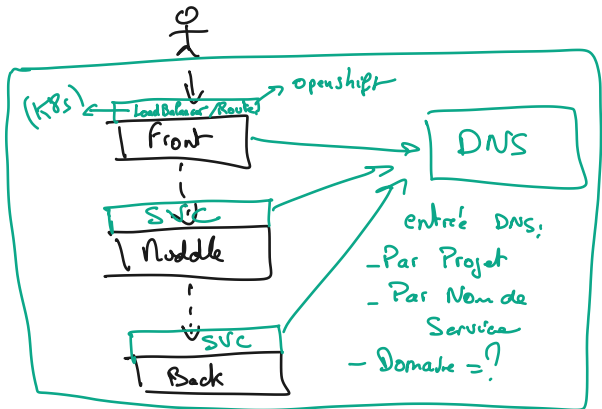
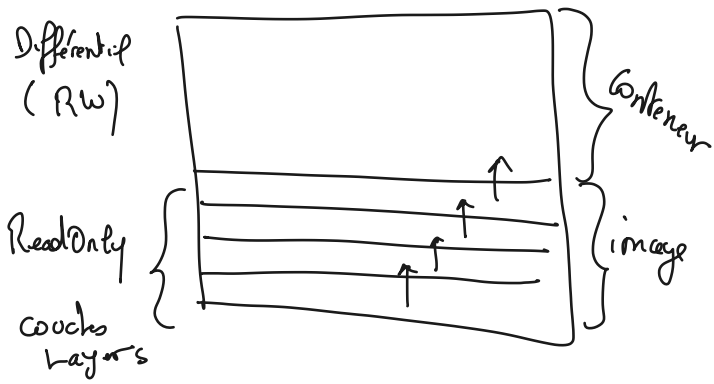




- ① Volumes ② classe.



cgroups \rightarrow Pour tout le système,
 dont les conteneurs



IPv4 : w.x.y.z

0 < element < 255 Réseau Internet

0.0.0.0 → 255.255.255.255

10.x.y.z
172.16.0.0
172.31.255.255
192.168.y.z } adresse ip privées

127.x.y.z → local host

169.254.y.z → ip privé automatique.

192.168.0.1 → 0.2
0.255

~~*~~
192.168.1.13

@ IP. - @ Réseau - Passerelle

ex : 192.168.0.1
Portée du réseau

192.168.0.0
(255 → broadcast)

192.168.0.254

@ IP 192.168.0.1 b8

Masque 255.255.255.0 → 1-1-1-1-0-0

Réseau 192.168.0.0 ↑

Dest = 192.168.1.12 taille du réseau Curseur =

$$0 \leq A \leq 126$$

$$128 \leq B \leq 191$$

$$192 \leq C \leq 223$$

$$224 \leq D \leq ? \quad \rightarrow \text{Multicast}$$

0.0.255 Masque \rightarrow obsolete (pas assez précis)

$$255.255.0.0 \rightarrow 1-1.1-1.0-0.0-0$$

$\xrightarrow{16 \text{ bits a' 1}}$

$$1-1.1-1.1-1-1-111000$$

$\xrightarrow{28 \text{ bits a' 1}}$

4 bits = 0

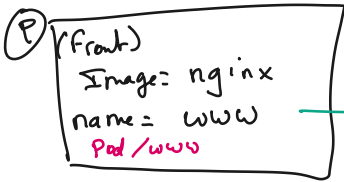
4 bits à 0 \rightarrow 16 poss. b. à 1 -
1; dernier 0 = 164 ports

~~255.255.255.242~~

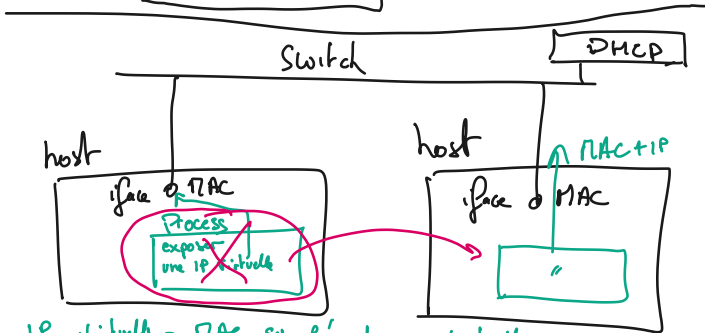
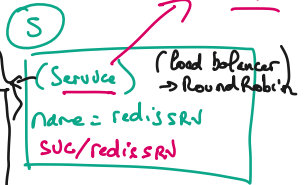
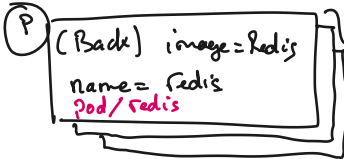
$192.168.0.1/28$

Classless Inter Domain Routing

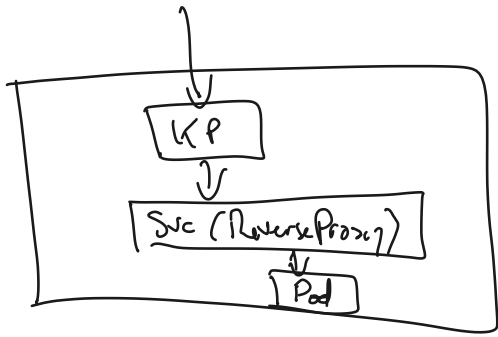
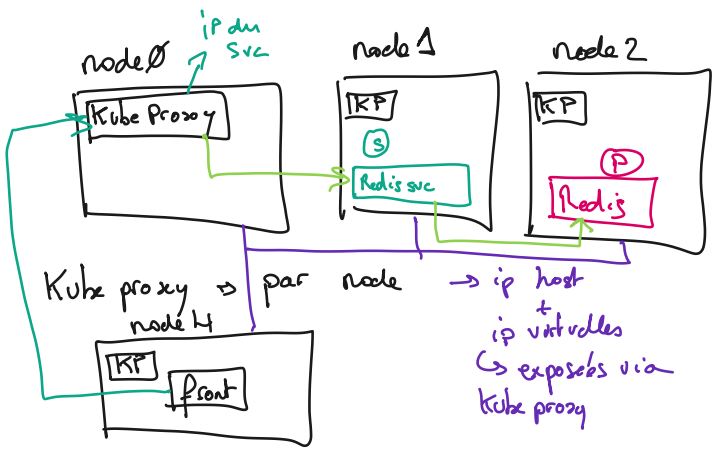
CIDR

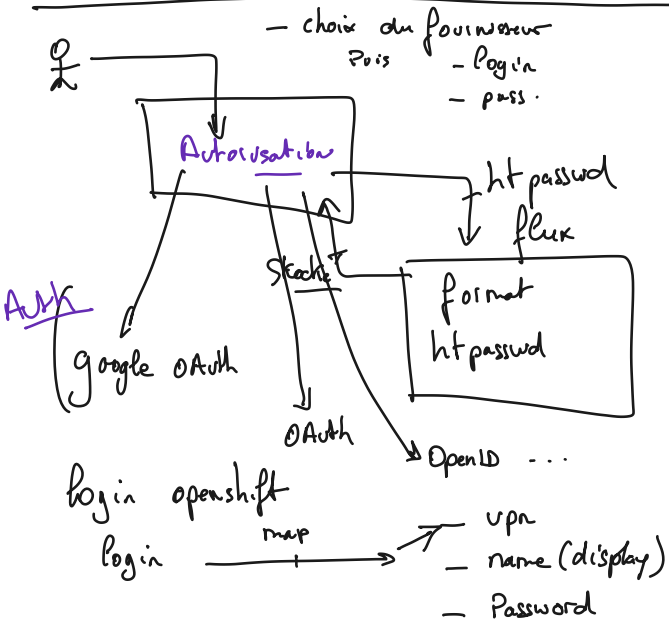
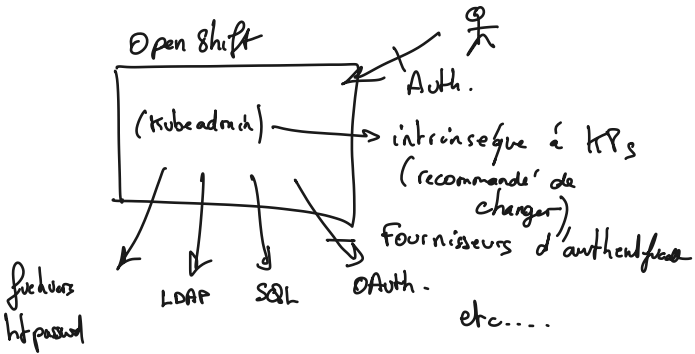


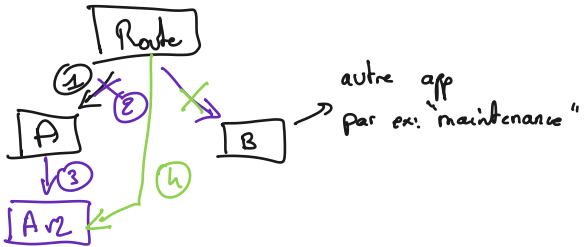
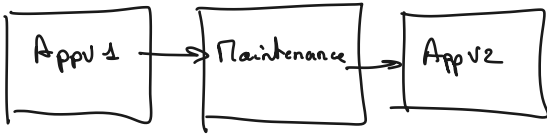
IP = redis SRV
 IP virtuelle
 fixe
 DNS



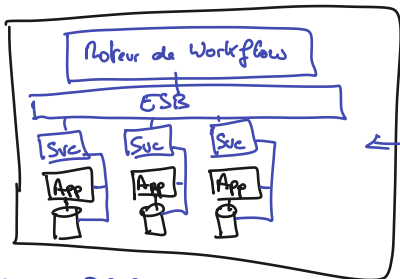
IP virtuelle = MAC supplémentaire virtuelle
 associée à une IP fixe ou dynamique
 SDN → Software Defined Networking







SOA



Web Service
SOAP

ESB = Bus
= Comm

- Synchrones : WS

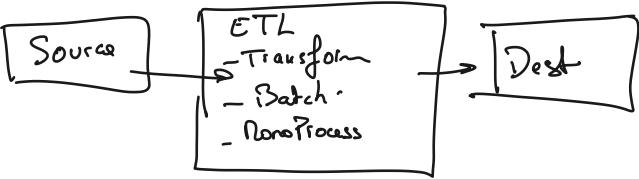
- Asynch : MQ (ESB)

IBM MQSeries
Workflow = orchestrateur -

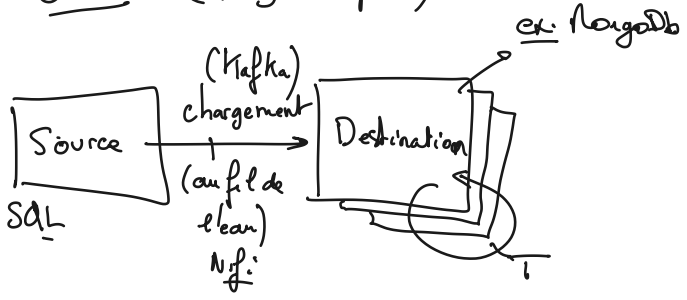
Micro Services \sim - ~~Extension des SOA~~
(subjectif)

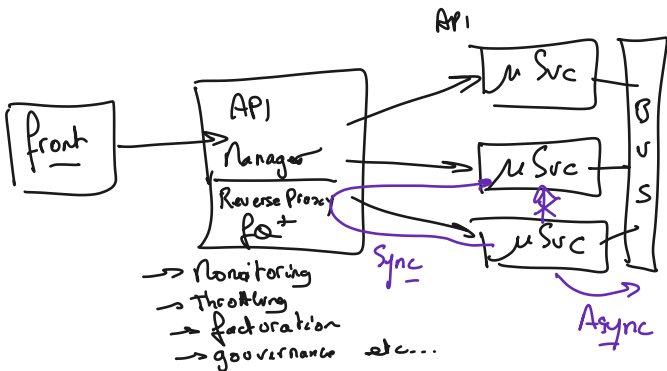
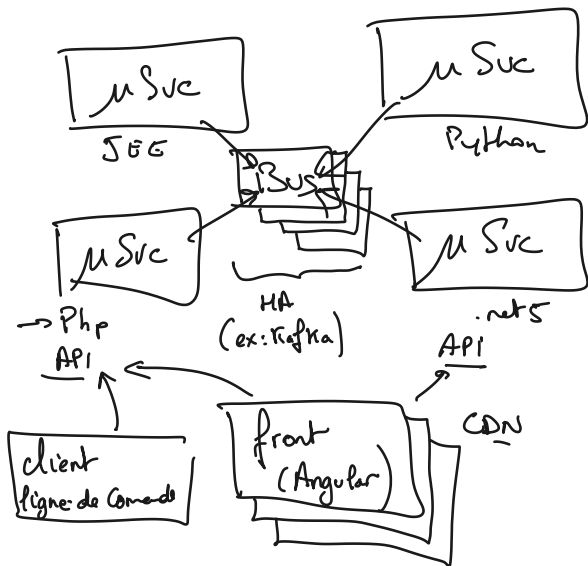
- Éclatement en n Brigues autonomes et découplées et non dépendantes d'un gros projet.
- Chaque Brigue est un projet logiciel spécifique
 - techno (JEE, .net, Php)
 - Bdad : Spl, NoSql
 - Bus
 - Process (batch) etc

ETL:



ELT: (Big Compute)





SQL(Ref) → Norm
est un pattern

Categories	
idc	<u>name</u>

Produits		
idp	idc	name

est un anti-pattern → Normalisation
NoSql

Keyspace

Categories

Key space

Produits

1 SQL NoSql

<u>Categories</u>

Produits	
claf	de l'af

Projet categories



Projet Produits

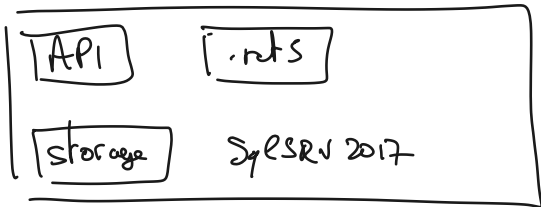
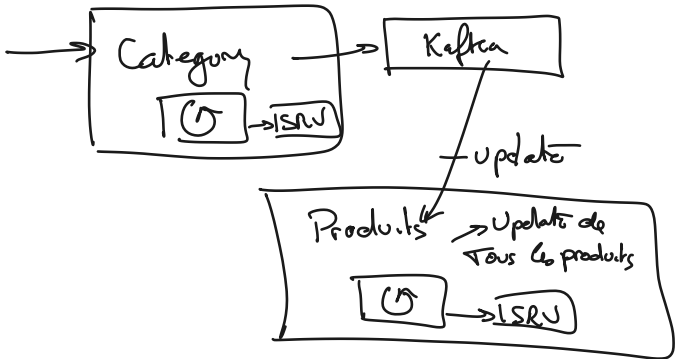


schéma de Produits

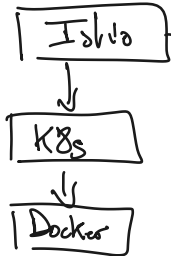
- id
- name
- category name

Plan de source Net = a'
pour un nom de
Categorie

/cat/1/ → POST
name = av2



100% open source



extension RH

