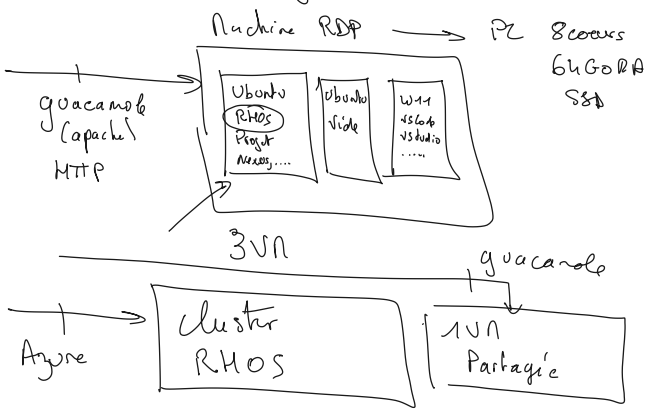
Bonjour tout le monde

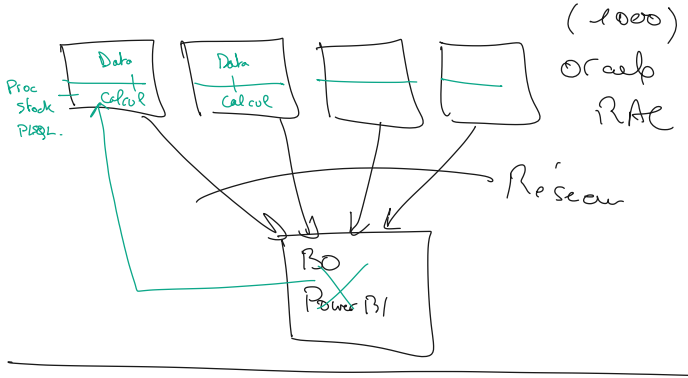
9h - 17h.
12h30 → 13h30

10h30 ± 15 min
15h30 ± 15 min.

→ Bcp de choses -
→ Pratiquer → - partage d'expérience
→ Communiquez ! → Questions !!

Moyens → Tableau blanc.
→ VM "La Defense / Guacamole"

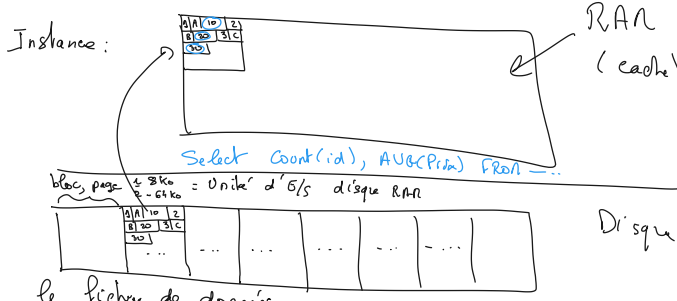




Produits

1	TV	4K	60Hz	HDMI	–
2	TV	4K	60Hz	HDMI	HDR

Row Store (conventionnel)



Column Store

Select AVG(Prix) .... FROM ....

RAM

10	20	30

Disque

id

1	2	3

Nom

A	B	C
A	B	B	B
AA	AB		

algo compression
RLE
Repeat Length Encoding

Prix

10	20	30

Ordre d'idée :

≃ 10x + Rapide pour l'analytique que le row store

≃ 100x

fact Commandes

date	Nom Produit id	Nb Cdé	Prix Cdé
...	A 1	2	10
...	B 2	1	20
...	B 2	3	22
...	B 2	4	20

Colonne = pas remplissable (Pas de 2nde table)

Non relationnel

dimNoms

id	Nom
1	A
2	B

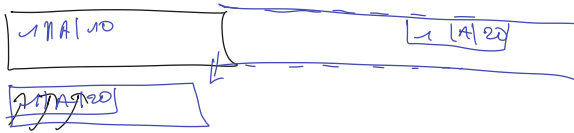
Append Only (Cassandra)

1	A	10	.....	1	A	20

Colonne

Update Prix=20 where id=1

MVCC

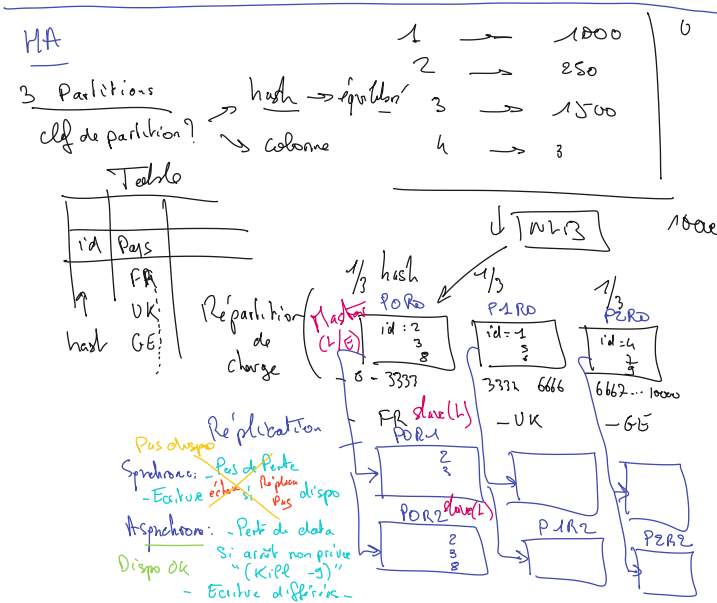


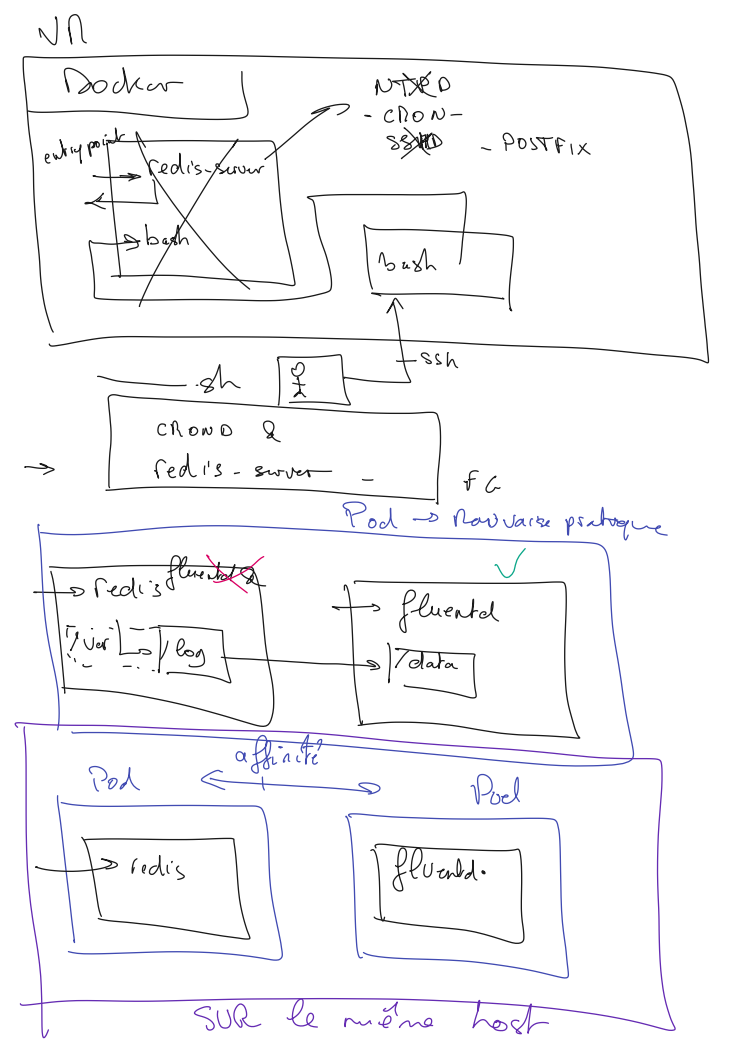
Update (SQL)

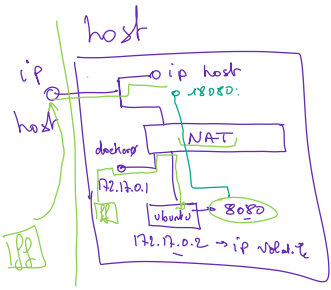
Cassandra (NoSQL)

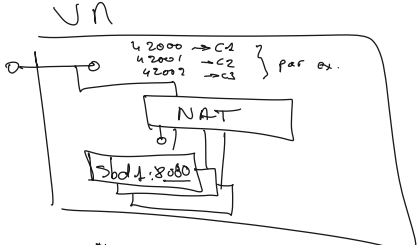
Instance → Down → GC du Disque → Up

BDD NoSql → Conçues pour être en cluster



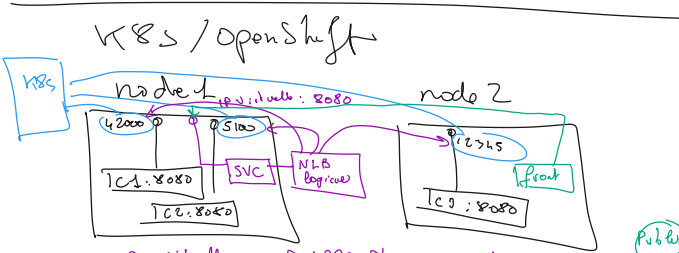

VM
Docker
entrypoint
redis-server
bash
NTPD
- CRON -
- POSTFIX
bash
ssh
CROND &
redis-server
FC
Pod → mauvaise pratique
redis
fluentd
/var
/log
fluentd
/data
affinité
Pod
Pod
redis
fluentd
SUR le même host





[< port du host >] : < port du Ctr >

— p 8080.

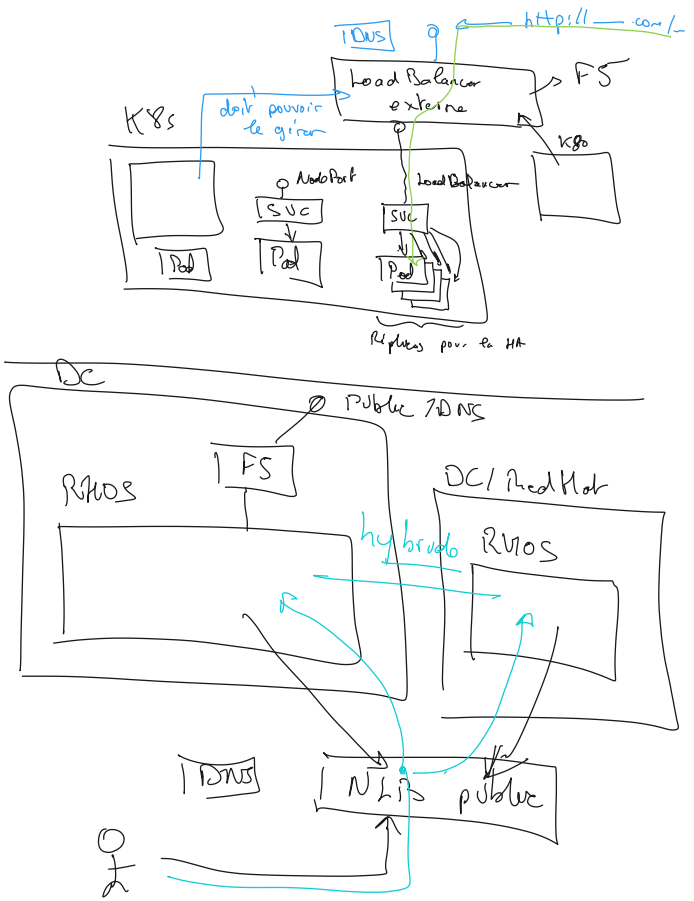


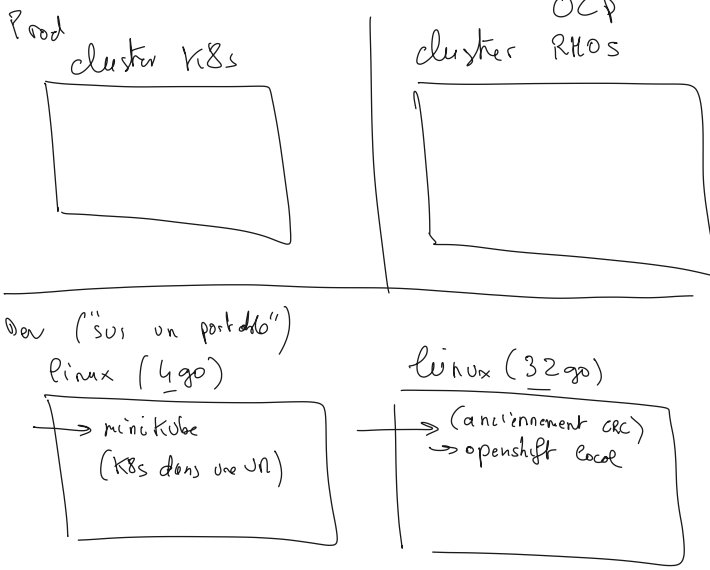
IP Virtuelle : IP / MAC Physique du node

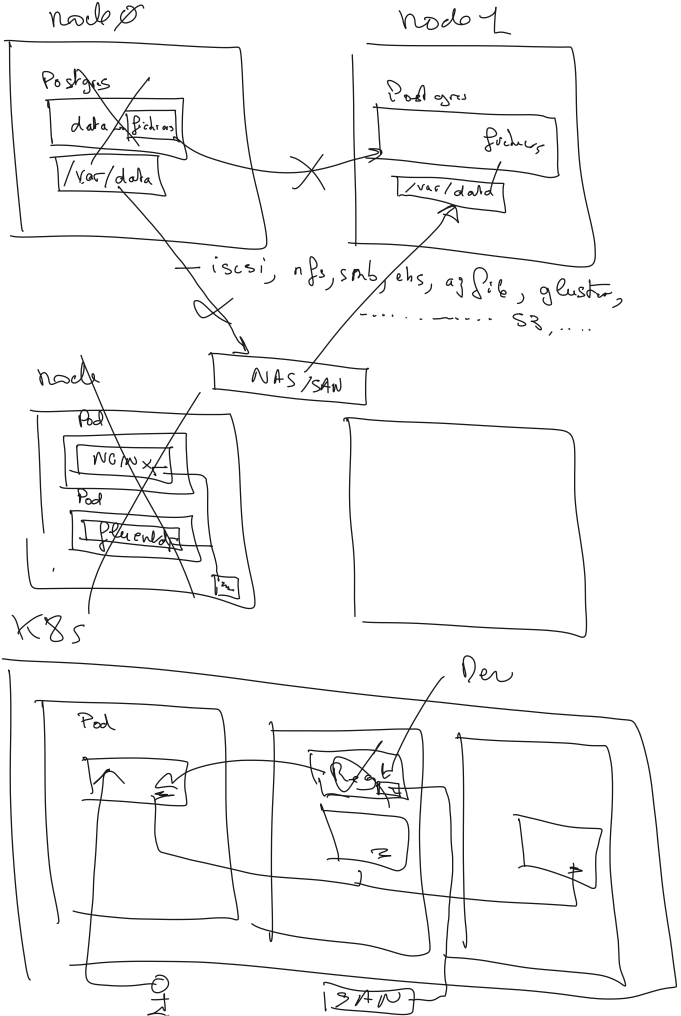
→ Type Load Balancer (externe) / Route (Openshift) → URL

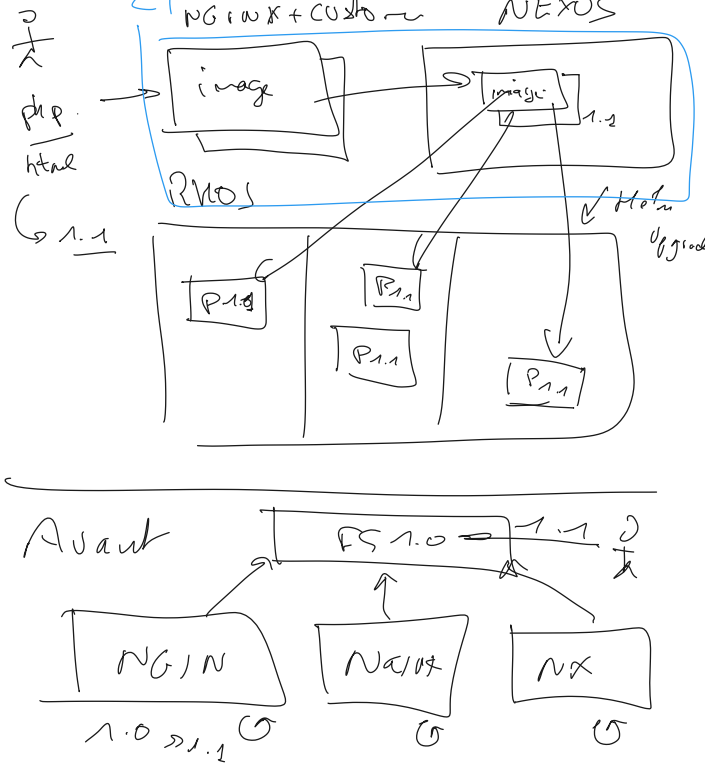
→ Service type Node Port : accès interne uniquement

K8s : — Pod (non accessible) → Tache ETL (BDD)


DNS
http://___.com/_
Load Balancer externe
F5
K8s
doit pouvoir le gérer
K8s
NodePort
Load Balancer
SVC
SVC
Pod
Pod
Pod
Pod
Replicas pour la HA
DC
Public DNS
F5
RHOS
DC/Prod Hat
hybride
RHOS
DNS
NLB
public

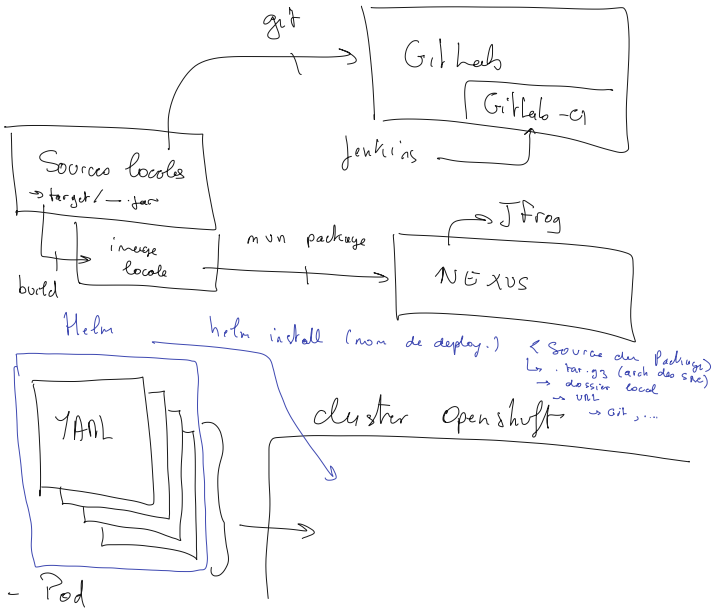

Prod
cluster K8s
OCP
cluster RHOS
Dev ("sur un portable")
linux (4go)
minikube
(K8s dans une VM)
linux (32go)
(ancien environnement CRC)
→ openshift local


node Ø
node 1
Postgres
data
ficheros
/var/data
Postgres
ficheros
/var/data
iscsi, nfs, smb, ebs, azfile, glusterfs,
S3, ...
NAS/SAN
node
Pod
NGINX
Pod
fluentd
K8s
Dev
Pod
SAN


CI
NGINX + Custom
NEXUS
image
image:
1.1
php
html
1.1
PROD
Helm
Upgrade
P1.0
P1.1
P1.1
P1.1

Avant
FS 1.0
NGINX
Nginx
NX
1.0 >> 1.1

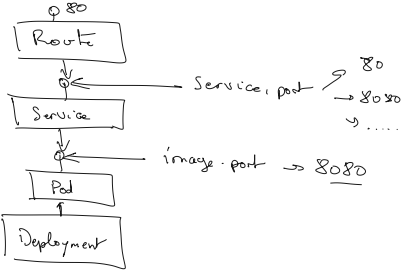
Cluster OpenShift d'essai en ligne :
https://developers.redhat.com/learn/openshift

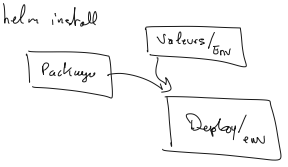


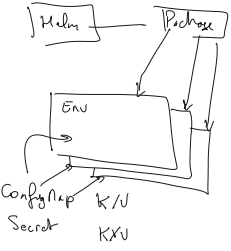
- Pod
- Service
- Route
- Deployment (config)
  - ReplicaSet
  - Horizontal Pod Autoscaler ...

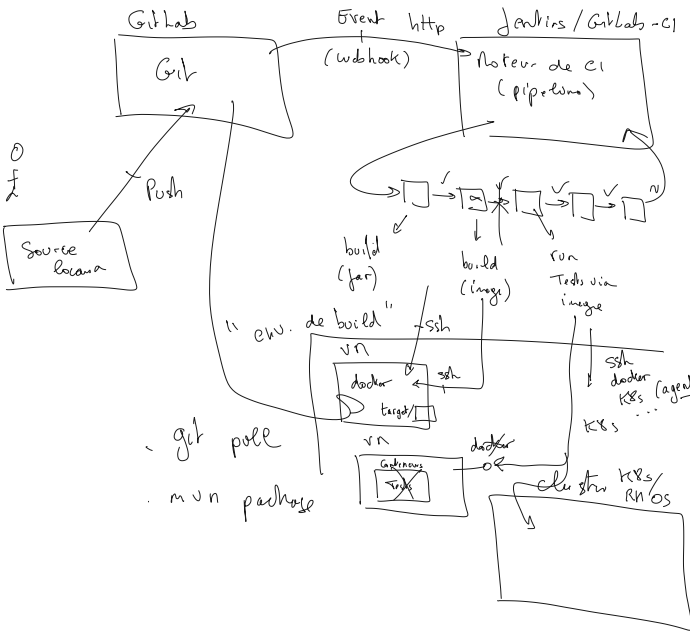
- helm create <nom de votre chart>
- oc new-project <un projet>
- helm install <nom de déploiement> <dossier du chart>
- Contrôler graphiquement le déploiement (dev / Topology / choisir le projet)

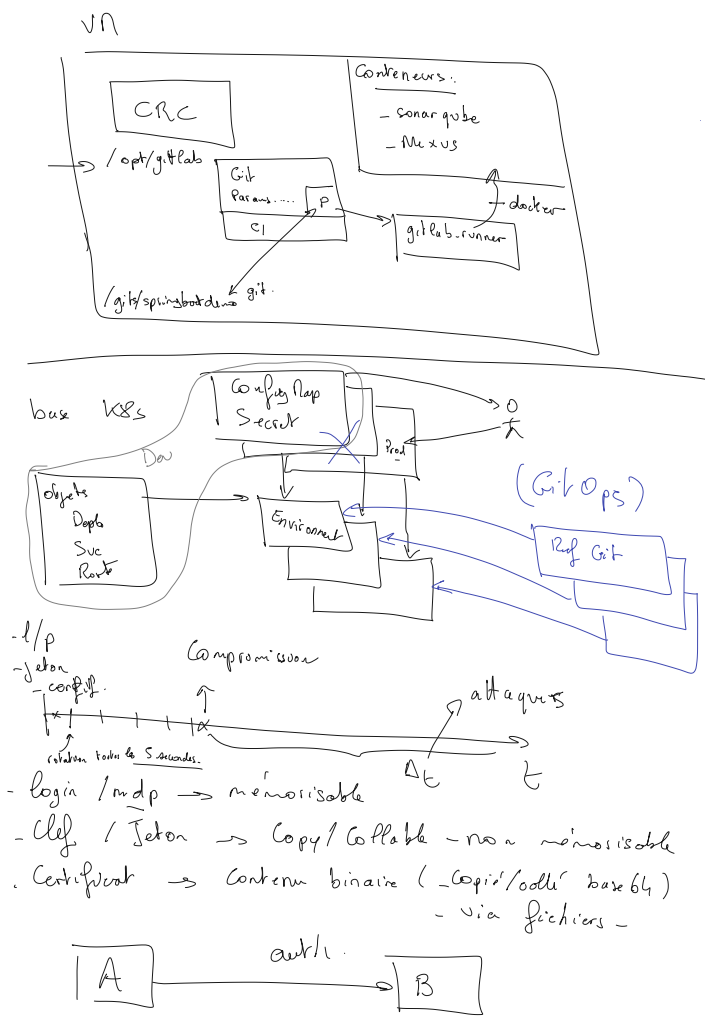
- Exercice :
  - Fabriquez un projet vide
  - oc status : vous devez etre dans ce projet
  - Créez un package helm par défaut
  - Déployez ce package : vous obtenez un serveur nginx non fonctionnel car root est interdit
  - retirez ce déploiement
  - modifiez le contenu du package pour faire référence à :
    - image : nboost/springbootdemo
    - tag : latest
    - port : 8080
  - Redéployez
  - ça devrait être fonctionnel.

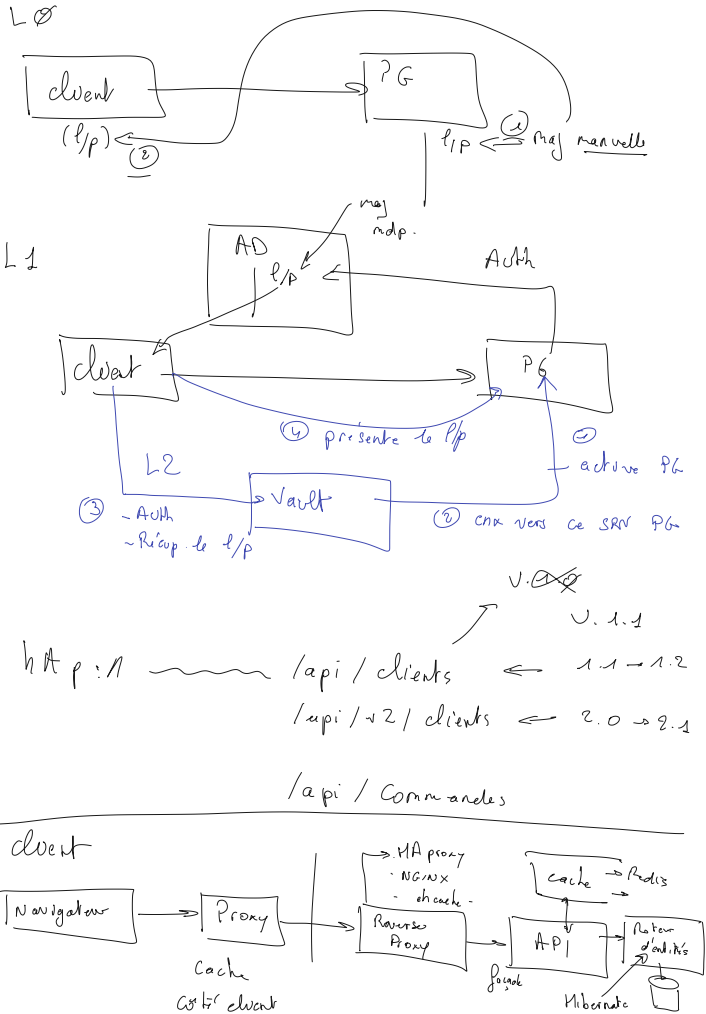


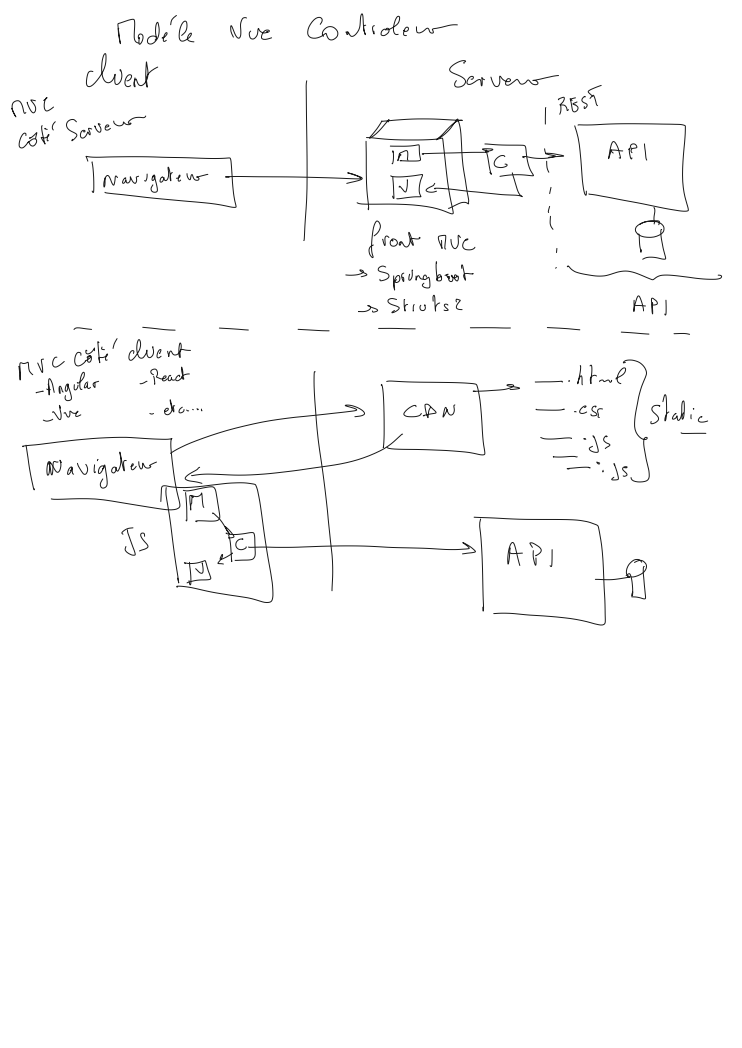


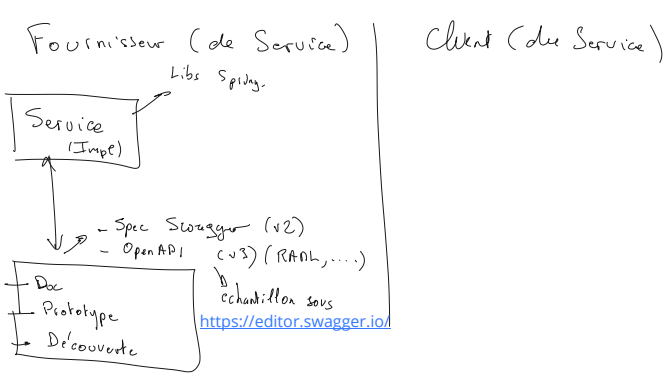


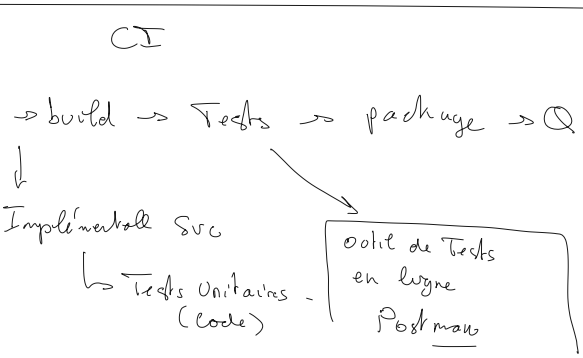

Gitlab
Event
http
Jenkins / Gitlab -CI
Git
(webhook)
Moteur de CI
(pipelines)
Push
Source
Locaux
build
(jar)
build
(image)
run
Tests via
image
"env. de build"
ssh
VM
docker
ssh
target
ssh
docker
K8s (agent)
K8s
git pull
mvn package
VM
docker
Gatlingrunner
Tests
cluster K8s/
RH OS

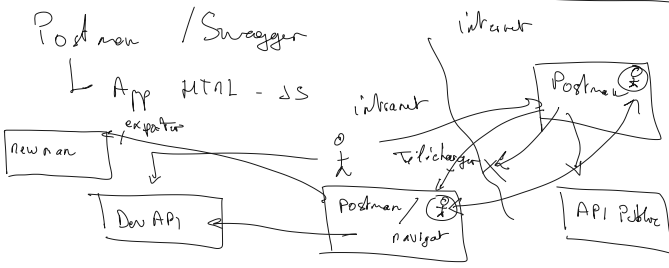

VM
CRC
Conteneurs :
- sonarqube
- Nexus
/opt/gitlab
Git
Params ....
P
CI
docker
gitlab-runner
/gits/springbootdemo
git
base K8s
Config Map
Secret
Dev
Prod
0
objets
Depls
Svc
Route
Environnement
(GitOps)
Ref Git
-l/p
-jeton
-config.
Compromission
attaques
rotation toutes les 5 secondes.
Δt
t
- login / mdp → mémorisable
- Clef / Jeton → Copy / Collable - non mémorisable
- Certificat → Contenu binaire (Copié/collé base 64)
- via fichiers -
auth
A
B

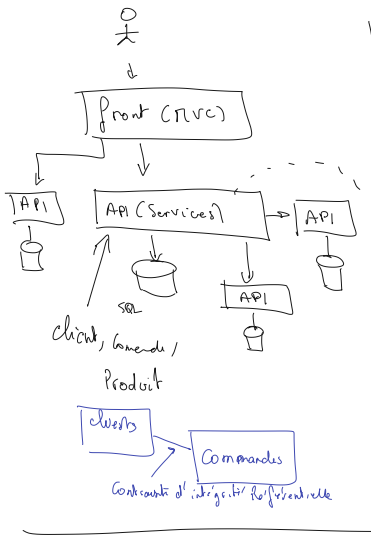

L∅
client
PG
(l/p)
2
l/IP
1
maj manuelle
maj ndp.
L1
AD
l/p
Auth
client
PG
4 présente le l/p
3
L2
Vault
- Auth
- Récup. le l/p
2 cnx vers ce SRV PG
1 active PG
V.∅
V. 1. 1
http://
/api/clients
1.1 → 1.2
/api/v2/clients
2.0 → 2.1
/api/Commandes
client
Navigateur
Proxy
Cache
Cote client
MA proxy
- NGINX
- ehcache -
Reverse Proxy
cache
Redis
API
Fonction
Retour d'entités
Hibernate


Modèle Vue Contrôleur
client
Serveur
MVC
coté Serveur
navigateur
M
V
C
REST
API
front MVC
→ Springboot
→ Struts2
API
MVC coté client
-Angular
-Vue
-React
- etc...
CDN
.html
.css
.js
.js
Statique
navigateur
M
V
C
JS
API


Fournisseur (de Service)
Client (du Service)
Libs Spring.
Service
(Impl)
- Spec Swagger (v2)
- OpenAPI (v3) (RAML, ...)
échantillon sous
https://editor.swagger.io/
Doc
Prototype
Découverte


CI
→ build → Tests → package → Q.
Implémentation Svc
Tests Unitaires
(code)
Outil de Tests
en ligne
Postman

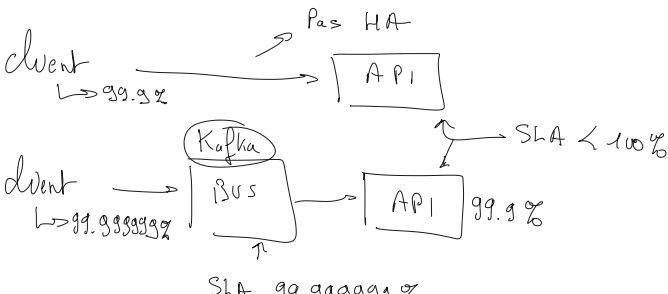

Postman / Swagger
App HTML - JS
exporter
Newman
Dev API
Postman
naviguer
intranet
intranet
Postman
télécharger
API Publique

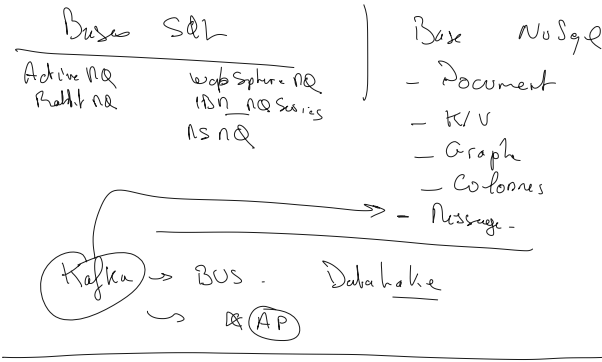


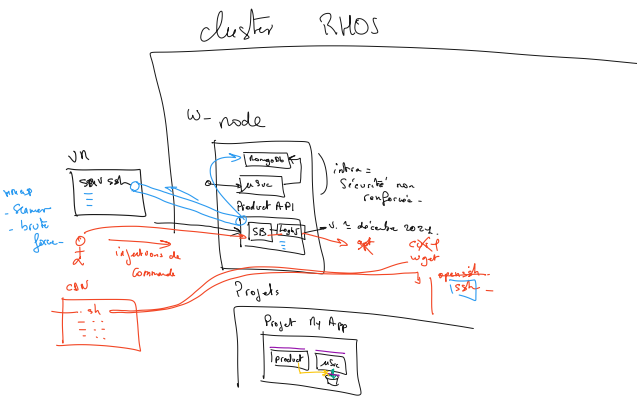
Appel Synchrone → Simple à implémenter

ASynchrone → + Complexe

- Hautement Disponible





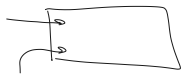


Design pattern SAGA pour les transactions distribuées :
https://microservices.io/patterns/data/saga.html

Desing pattern CQRS pour optimiser les API avec Read/Write dans une base sous jacente :
https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs

Disque / Stockage

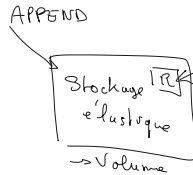
APPEND (DW)



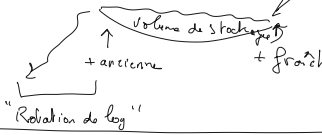
Purger selon des règles

Data Lake

APPEND

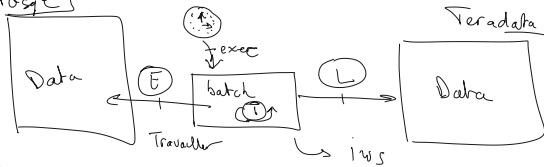


Source



Traitement en batch

ex: Postgres



→ Iws

→ Talend Data Studio

→ Informatica

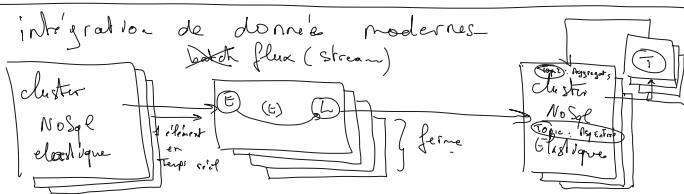
Batch:

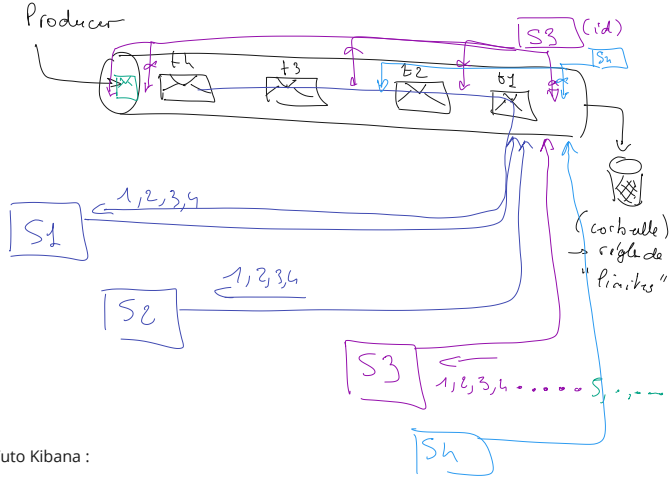
- Pas d'exécution //
- Pas de Temps réel
- un batch voit le n lignes à traiter

par ex: Toutes les nuits, intégrer les activités comm. de la veille dans le DW.

intégration de données modernes

~~batch~~ flux (stream)





Tuto Kibana :

https://grafana.com/tutorials/grafana-fundamentals/?utm_source=grafana_gettingstarted