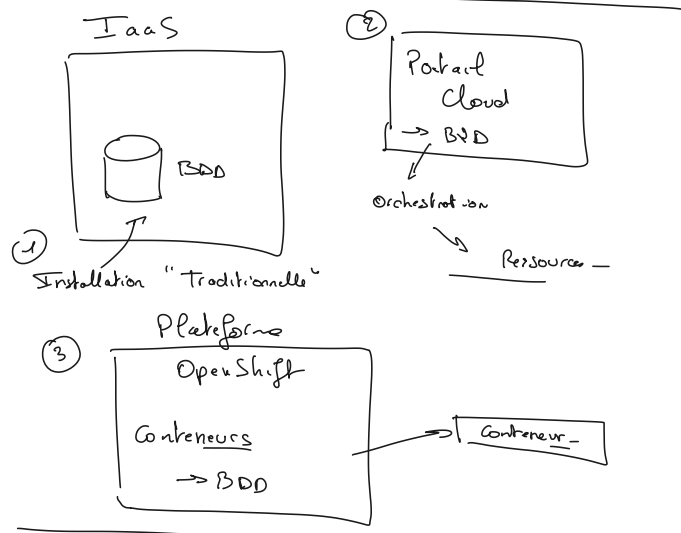
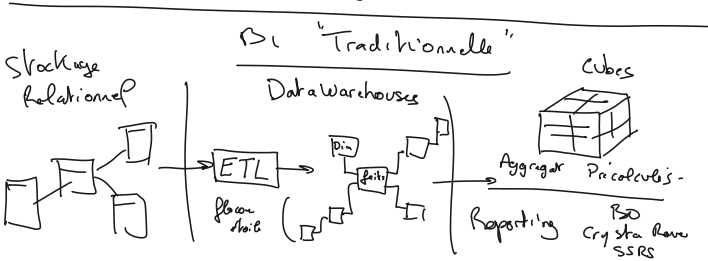
Bonjour tout le Monde

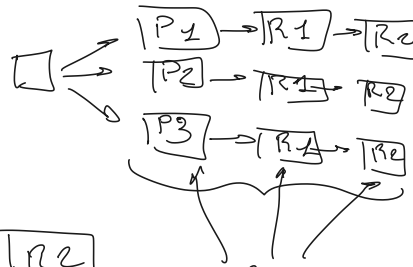


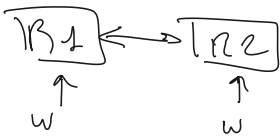


HA High Availability =

Performance + Disponibilité = Partitions + Réplication

Idéal :





Ecriture simultanée du même id = ?

→ Verrou

→ ou Pbm de
- Réplication
- conflit
- Cohérence

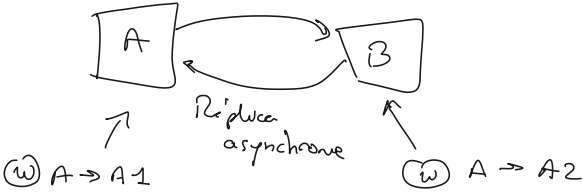
MVCC

Version -> 1 2 3 ...

database <moment>

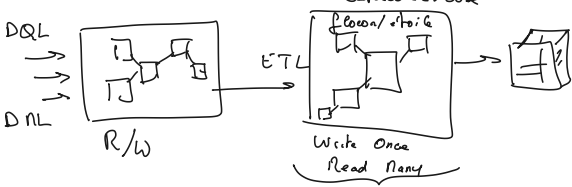
PK

PK id	Version	data
1	1	A
1	2	A2
1	3	A3
2	2	B
3	3	C



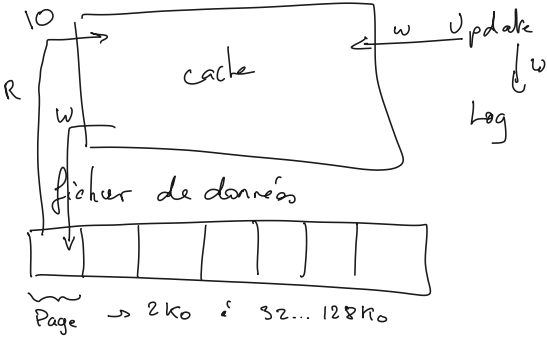
Row Store / Column Store

Referential "in forge"



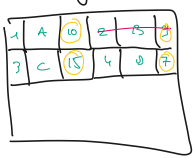
Row Store

Instance de BDD / Cache



Row

Page :

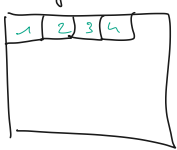


id (int)
nom (char)
Prix (int)

→ Nb d'unités en stock ?
→ Prix } total moyen ?

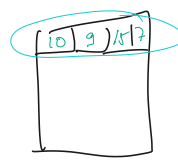
Column

Page Colonne id



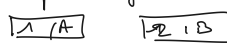
Page col Nom

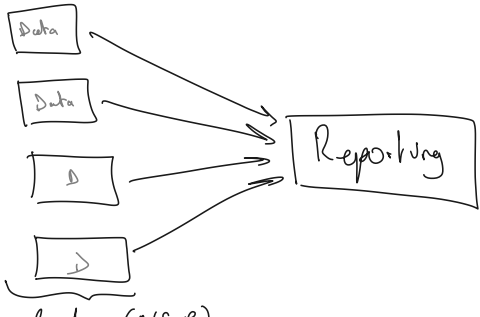
Col. Prix

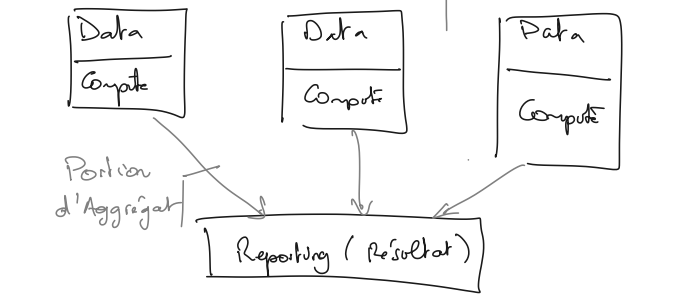


Prix Moyen = ?

Compression RLE Repeat Length Encoding







C / C++

```
int funct( int i) { }

int *pf() = funct( int i) { };
int *pf() = funct;

int rs = pf(10);
```

pf → funct()
pf → other funct()

funct2(* pf(int))

Prog fonctionnelle = passer une fonction en argument à une fonction

operation1(operation2)

→ C++, python, java, C# → ON fonctionnel

C# → Délégués → fonctions anonymes → Lambda expr –

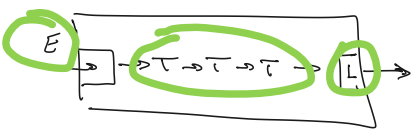
Java → fonctions anonymes → API Stream (map, reduce, ...)

Langages "fonctionnels" → Nlle langages simplifié en fonctionnels

- .net F# produit du .net
- Java Scala produit du bytecode Java

NIFI / ETL Batch

Extract
Transform
load



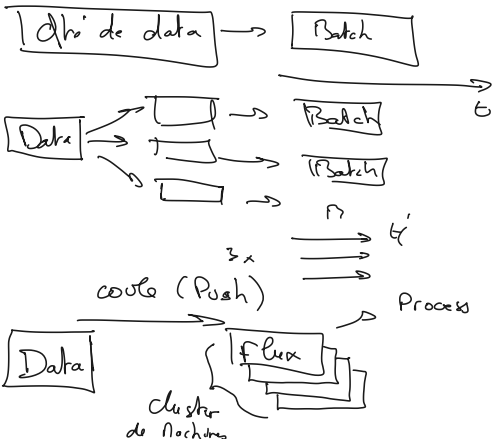
Solutions ETL
Microsoft SSIS
Oracle ODI
~~Talend~~ (open source)
Informatica / SDI gérés

Batch (Pull) → - Execution planifiée d'un processus
- Traitement par batch d'un lot de données.

Flux (Push) vs → - Exécution au fil de l'eau
- Traitement par donnée entrante

Pull → scruter, aller chercher la/les donnée(s)
Push → attendre, recevoir la donnée

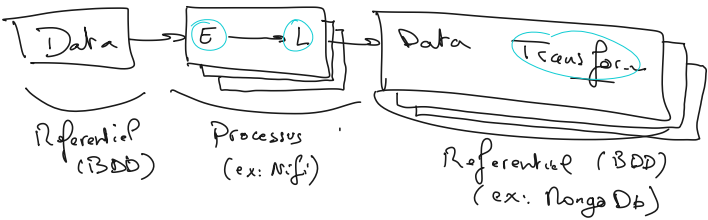
NIFI → - Node flux (Push)
- Cluster NiFi



ETL → Batch, relationnels

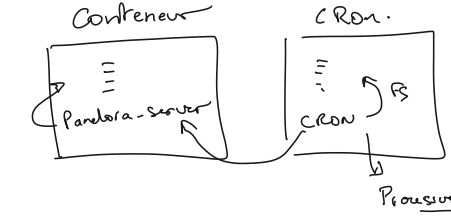
ELT → Flux, Big Data

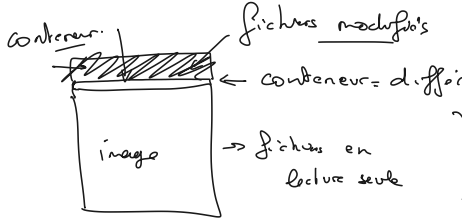
Extract Load Transform

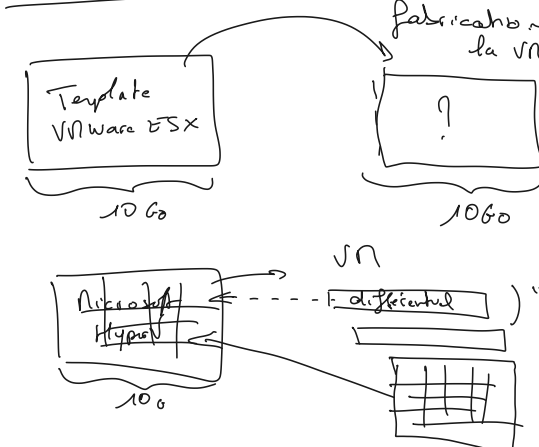


Nifi est un ELT et ETL

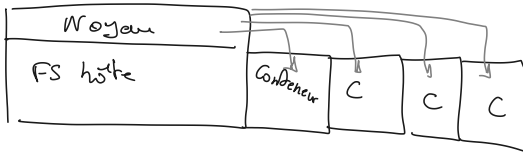
T est moins fonctionnel que le Batch (moins puissant)


Conteneur
CRON
Pandora-server
CRON
fs
Processus


conteneur
fichiers modifiés
conteneur = différentiel) Ø octets
image
→ fichiers en lecture seule
) 1 Go


fabrication de la VM
Template VMware ESX
?
10 Go
10Go
VM
Microsoft Hyper-V
différentiel
) "Ø octets"
10 Go

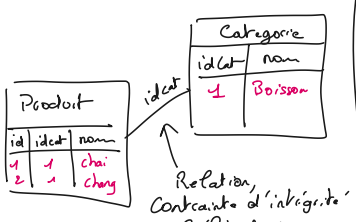
Kernel.org → Noyau Linux + Distribution → Fichiers = 1 Instance Linux.



Normalisation / Dénormalisation

Bases Référentielles

NS SQL Server, Oracle, ...



Dénormalisé :

Produits		
id	nomCat	nomProduit
1	Boisson	chai
2	Boisson	chang

Mauvais en relationnel

OK
- On DataMart
- NoSql

Data : Relationnel Normalisé	Cohérent Lent	Big Data Non Relationnel / dénormalisé Rapide	Eventuellement cohérent

Table - Produits
- Commandes

Normalisé :

Produits

idP	Nom
1	chai

Commandes

idc	date Commande
1	Today
2	hier

Commande Produits

idP	idc	Prix	Qte
1	1	10	3
1	2	9	5

Big Data / Dénormalisé

Produits

id	Nom
1	chai

Commandes

idc	date	WomProd	Px	Qt
1	today	chai	10	3
2	hier	chai	9	5

Cassandra

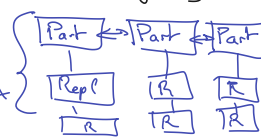
MongoDb

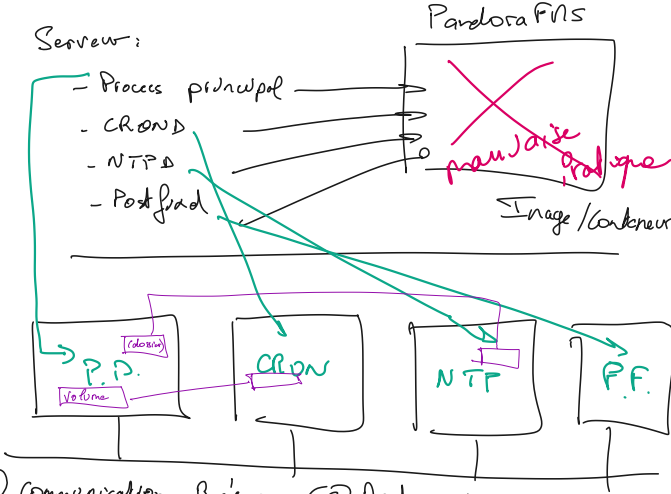
injection de contenu

cluster Cassandra

Servers

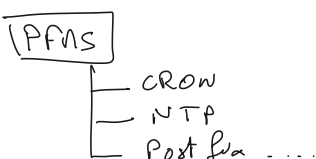
cluster MongoDb





→ Ex 4 Images → Lancer 4 Conteneurs

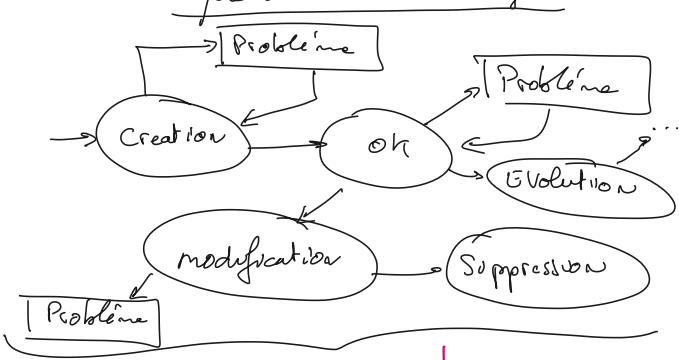
- → Volumes
- → Ports
- → ordre de démarrage
- → etc....



Encapsuler les ordres

docker container RUN ....

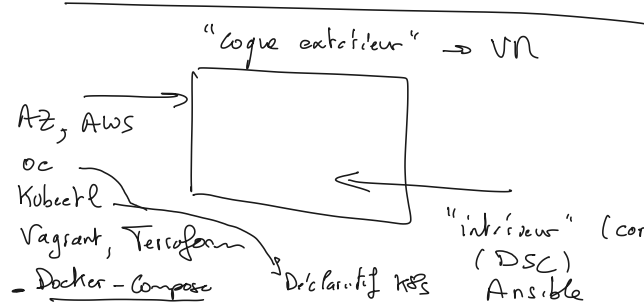
Dans un script shell → mauvaise Pratique !



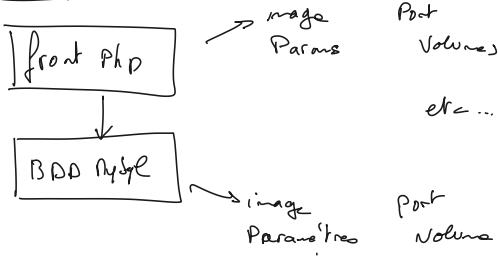
Créer un script pour chaque cas d'erreur et alternatif !

Impératif - (lignes de code) -> OK pour des "algorithmes"
- inutilisable pour de simples infrastructures

Déclaratif: - -> OK pour de l'infrastructure



Word press



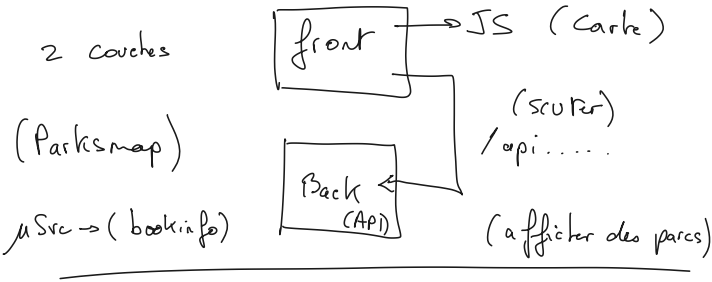
Côté Dev / Tests

Développer au format Docker-Compose (yaml)

Côté Prod

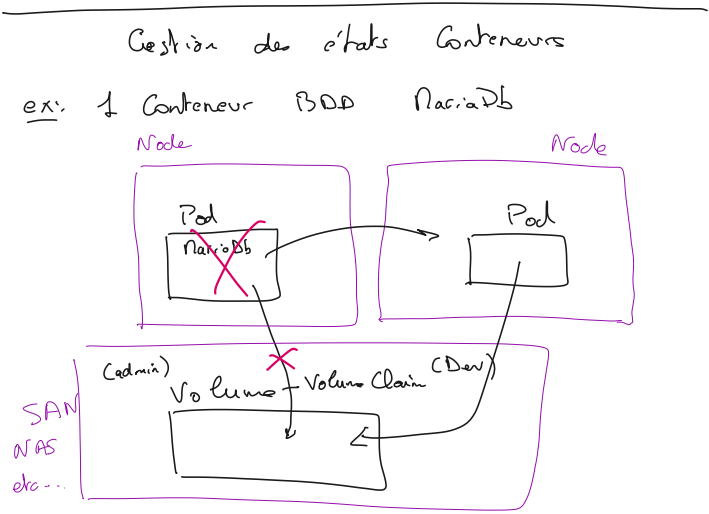
Dev. format Kubernetes (yaml)

- Ecrire l'App
  - Ecrire le Dockerfile
    (Ecrire le docker-compose si tests locaux)
  - Ecrire le déclaratif Kubernetes



Tous les tutos OpenShift : (Il y a 2 pages, cliquer en bas à droite pour basculer)
https://developers.redhat.com/courses?page=0

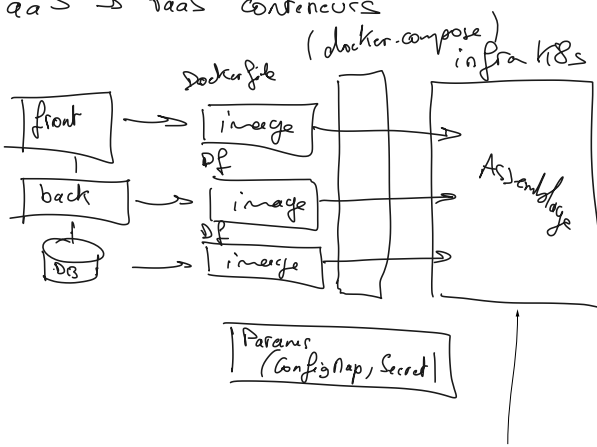
Tutos Kubernetes :
Katacoda Tutos



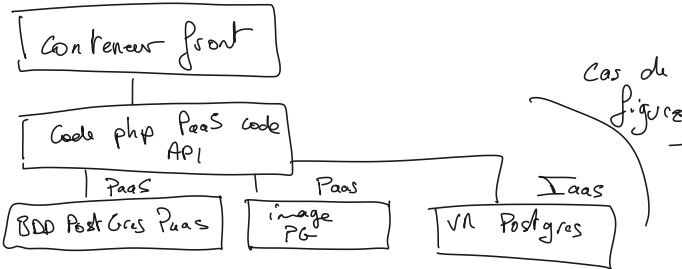
Sécurité des développements :
https://cve.mitre.org/ (les failles)
https://nvd.nist.gov/ (autre base)
https://owasp.org/www-project-top-ten/ (base de travail pour le dev sécurisé)
https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html Erreur de conception toutes technos confondues.

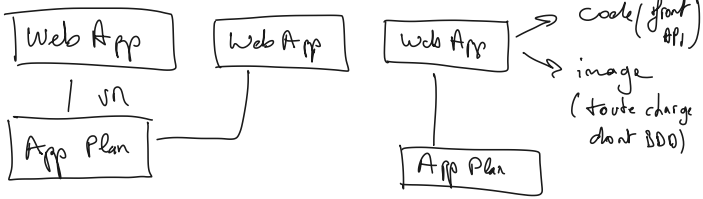
Conteneuriser une App (toutes les couches nTier)

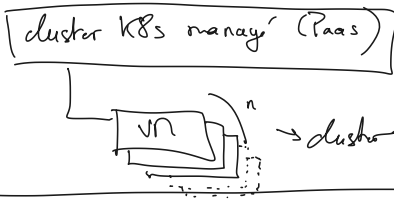
IaaS → PaaS conteneurs

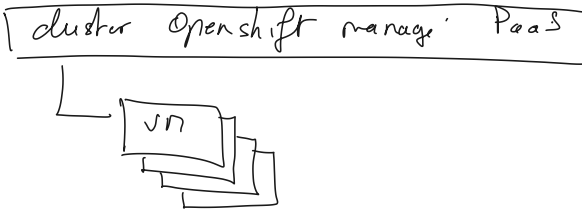


https://kubernetes.io/fr/docs/concepts/workloads/controllers/deployment/

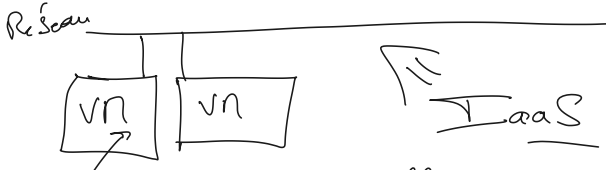






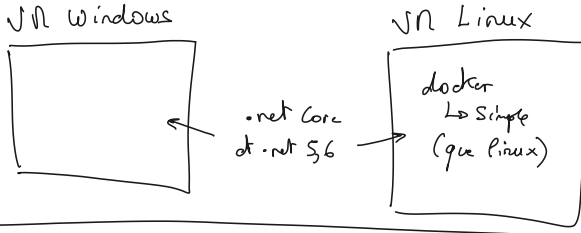


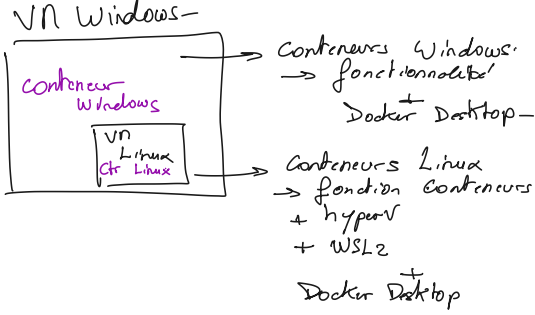
Monter mon cluster K8s ou Openshift



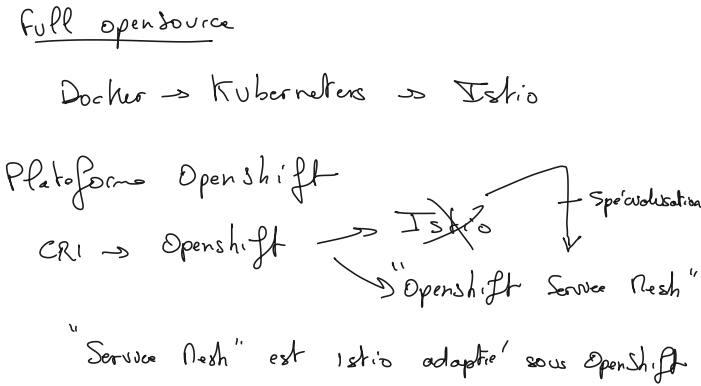
injecte le contenu (installation, paramètres...)

IaaS Conteneurs





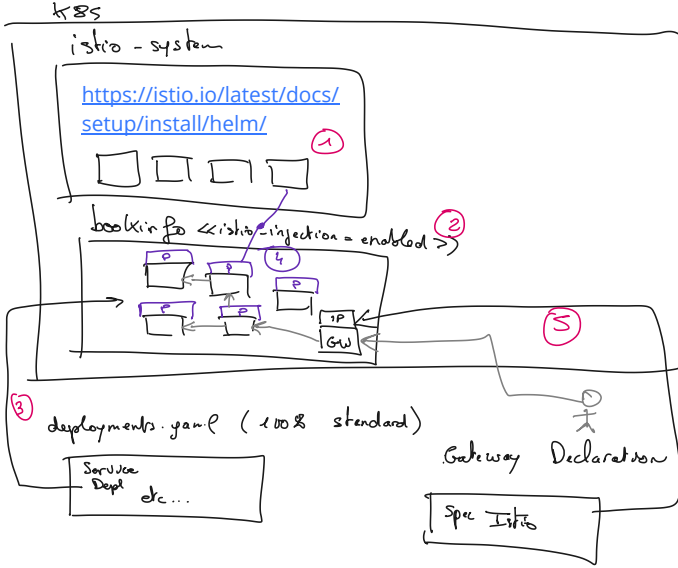
Microservices :
https://istio.io/ (base du framework en opensource)



Tuto Istio :
https://istio.io/latest/docs/examples/bookinfo/

Tuto OpenShift Service Mesh :
https://docs.openshift.com/container-platform/4.8/service_mesh/v2x/preparing-ossm-installation.html

Film :
https://www.youtube.com/watch?v=GHYdOpGUJ3w


k8s
istio - system
https://istio.io/latest/docs/setup/install/helm/
1
bookinfo <istio-injection = enabled>
2
4
P
IP
GW
5
3
deployments.yaml (100% standard)
Service
Depl
etc...
Gateway Declaration
Spec Istio