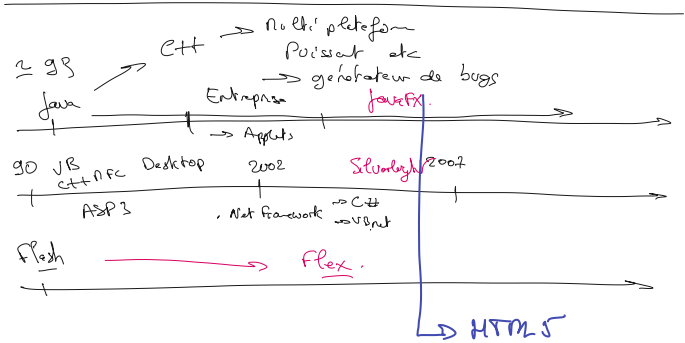
Bonjour Tout le Monde



environnements de Dev

Microsoft → Visual Studio

Java → ≃ 95 IBM Rational software

IBM → "Eclipse" → env de Dev

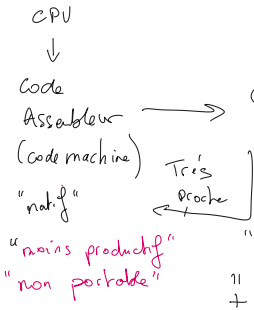
Unix → IBM

→ SUN → NetBeans

Plateforme RCP Eclipse

Plugins

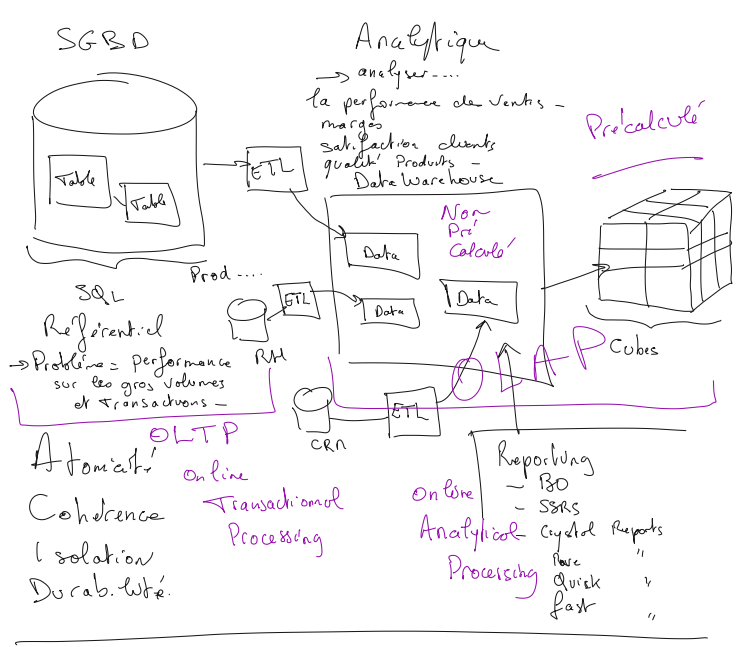
Techniques "natives"

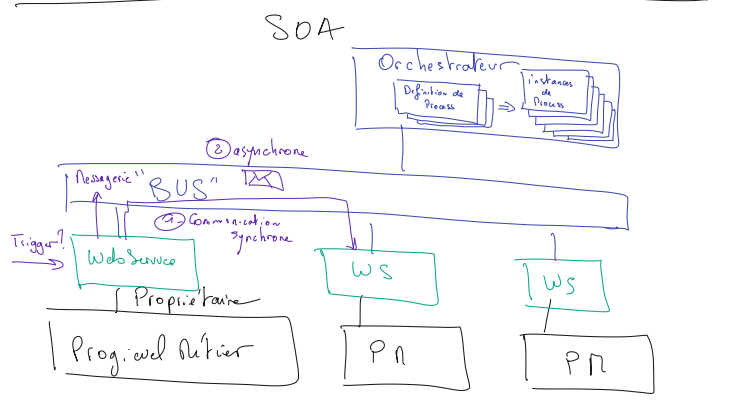


Besoins web

"Productivité"
"Fiabilité"
"Haute Disponibilité"

- "runtime de machine virtuelle"
  - Ne pas confondre avec la virtualisation
  - → Java RE } gérées
  - → .net }
  - → interprété
    - → V8 moteur Javascript
      - Node JS
        - JS est mono thread
    - → Python → le + Connu
      - → Multi plateforme
      - → accessible à tous
      - → "tout faire"
    - → PhP
      - → Accessible
      - → Scriptable
      - − Pas de Desktop
      - → Bien Connu -


SGBD
Analytique
→ analyser ....
la performance des ventes – marges
sat. action clients
qualité Produits –
Data Warehouse
Précalculé
Table
Table
ETL
Data
Data
Data
Non Pré Calculé
Prod ....
SQL
Référentiel
→ Problème = performance sur les gros volumes et transactions –
ETL
ETL
RDM
CRM
OLAP
Cubes
OLTP
Atomicité
Cohérence
Isolation
Durab. Isté.
Online Transactional Processing
Online Analytical Processing
Reporting
– BO
– SSRS
Crystal Reports
Rave
Quick
Fast


SOA
Orchestrateur
Définition de Process
⇒
instances de Process
② asynchrone
Messagerie "BUS"
① Communication synchrone
Trigger ?
Web Service
Propriétaire
Progiciel Métier
WS
PM
WS
PM

Serveurs web

"avant"

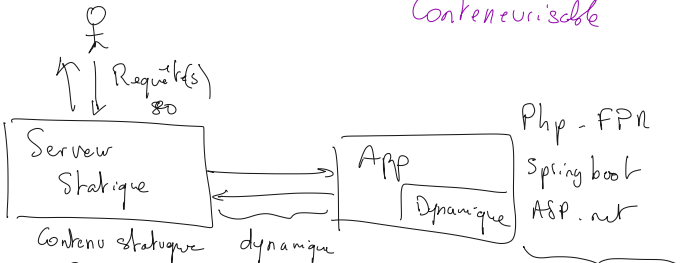
WebSphere
Jboss
Wildfly
NB: IIS
etc...

"maintenant"

- Java Spring boot
- ASP.net MVC Core 6,7.) Kestrel

Pas Conteneurisable

Conteneurisable



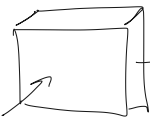
- Performance
- fonctionnalités
  - Cache
  - ReverseProxy
  - Load balancer etc...

Bonne Pratique actuelle

Années 80 +

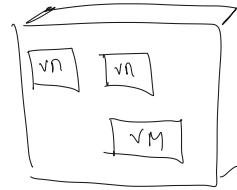
90

Serveurs (Physique)



Rôles – Fichiers
– Authentification
– Bases de données - ...

Serveur (de virtualisation)



Srv VM – VMWare ESX
– Microsoft HyperV
"à la mano" – KVM – (open source)

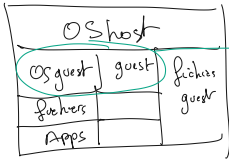
90 + ⟶ Virtualisation

Problèmes – Lourd

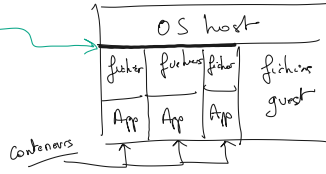
OS host
+ OS guest
⇒ Applications ⇐

2010+ ⟶ Conteneurisation

Virtu



Conteneur



"On Prem" Sur Site

- Maîtrise des coûts
- Risques de sécurité
- Réplica physique Région distincte

Hosting

- Maîtrise des coûts
- Renforcement de sécurité
- Eventuel Réplica hors site

Cloud

- Non Maîtrise de coûts (souvent mauvaise surprises)
- Sécurité extrême
- Conçu pour les réplicas
- Mutualisé
- Confiance des données ?

IaaS
Infra VM

PaaS
Plateforme = Serveur (Web, BDD, ...)

SaaS
Logiciel

wordpress

hebergement VM

Conteneurs
- Docker
- Kubernetes

Cloud providers:

- Azure MS
- Amazon AWS
- Google Cloud Platform
- IBM Cloud
- Digital Ocean
- OVH cloud / orange cloud } open Stack

Ressources pour Azure :
Docs : https://docs.microsoft.com/fr-FR/azure/?product=popular
Learn : https://docs.microsoft.com/en-us/learn/azure/

Cloud

Public (Azure, AWS, ...)

- → Paye les ressources
- → choix de la localisation pour la réglementation
- → Conso à la Demande

Cloud Hybride

Jonction ressources (cloud ou Tradition) Privées et cloud public

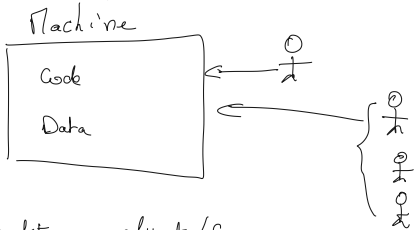
Cloud Privé

- "Logiciel" de cloud
- Portail
- outil scripting
- orchestrateur
- etc ....

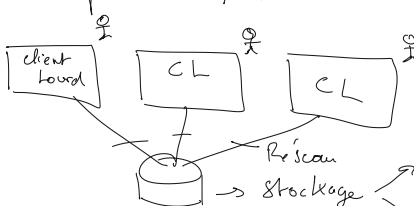
⇒ Livraison de Ressource simplifiée

Architectures des Applications

1) Monolithique ≈ Débuts de l'informatique



2) Desktop client / Serveur



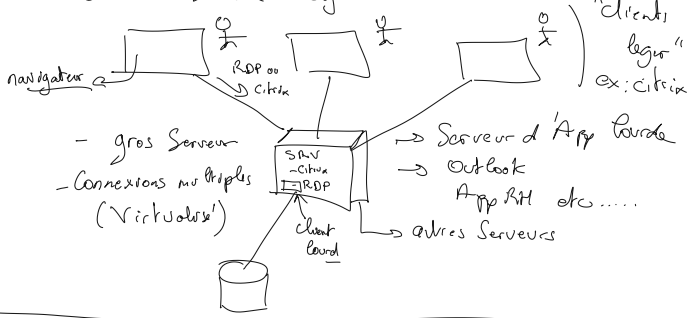
Problème : diffusion et maj des Apps

- Fichiers
- Bases de données
  - Oracle
  - SqlSRV
  - MySql
  - PG ....

Solution (± fin 1990 – 2000)

→ Serveur d'Applications –

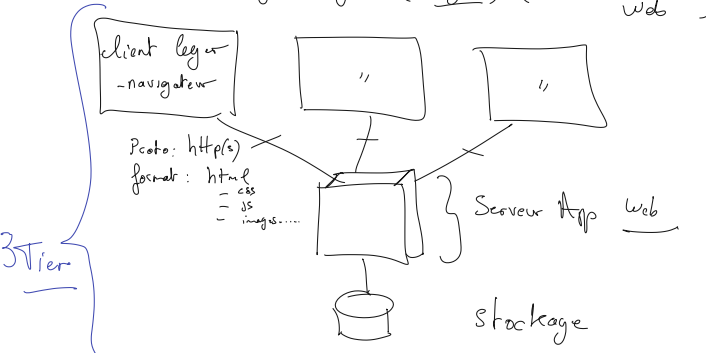
Client Lourd – leger



Migration App lourde → App web

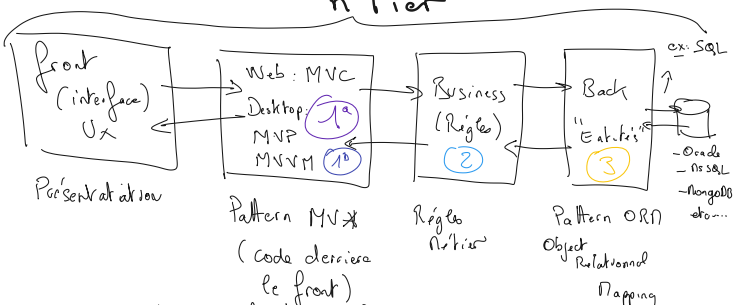
client lourd leger (ex Citrix) →

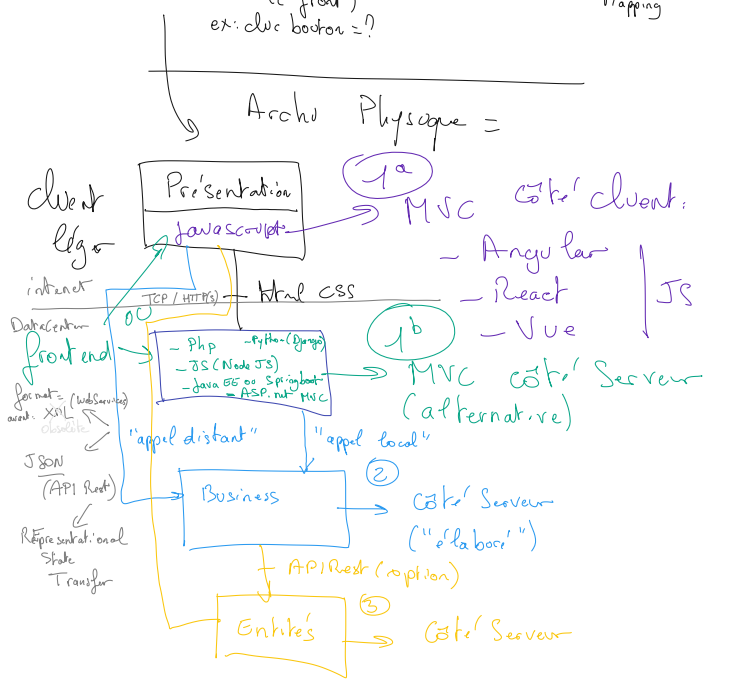
client leger leger (leger) (ex: navigateur web



Couches → Tier

Architecture des Applications nTier





Aujourd'hui < 2010 → SOA, EAI, ESB

"grosses App d'ent."

depuis ≃ 2010

→ "Simple" → archis n-Tier (Majorité)

→ "Complexes" → archis en μ Services

Critère Simple / Complexe :

- Beaucoup de Services
  ex: facturation, logistique, gouvernance, dashboard, analytique, ..., admin
- Découplage Complet (modèle couplé, couplage fort, faible, découplé - dépendances)
- Multiplication des Techno (polyglottes)
  - ≃ 2008 → JEE, .net
  - ≃ 2015 → Python, Node
  - > 2020 → Serverless, No Code, Low Code, ...
    - Work flow
    - front + fonctions.

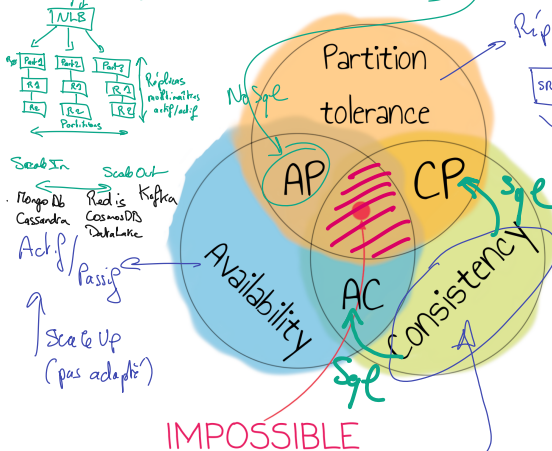
n Tier	μ Svc
- "Simple" - socle technique Simple Sql/NoSql → choix Cohérence ou haute dispo	Découplage Polyglotte Haut niveau Disponible
"monolithique" → dépendances susceptible de devoir être réécrit	infrastructure Complexe - NoSql: écrire la cohérence (conteneurisation) Plusieurs Projets / équipes - Nécessite un framework / infra pour être conforme entre μ Svc

nTier → Cohérent (Sql)
ou
Mouvement Disponible (NoSql)

µSvc → MOT → NoSql ?

→ NoSql / CAP → archi µSvc

NoSql = MA = Availability + Partionning (AP)



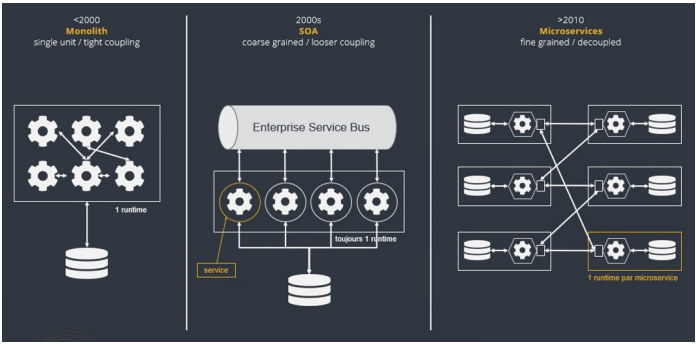
SGBDR
Base NoSql

→ Pas Mouvement Dispo
→ Panne
→ Saturation de Charge

NoSql de BDD
Serveurs "maison"

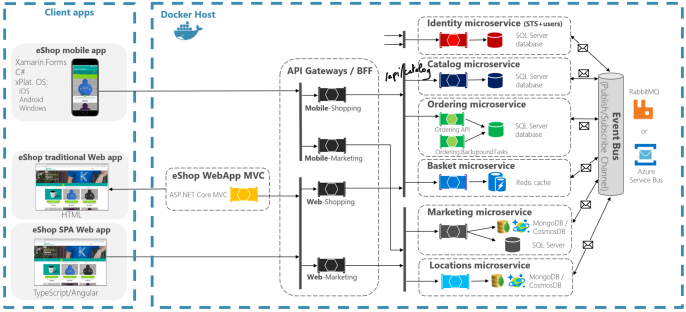
→ Cohérent
(Pas de perte de données)

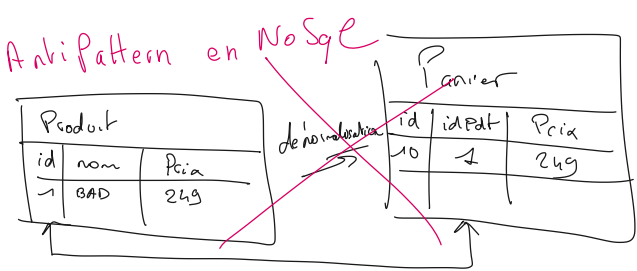
≈ 2000 explosion Trafic / Requêtes internet = Besoin d'un stockage + simple, + performant, NoSql éventuellement Cohérent



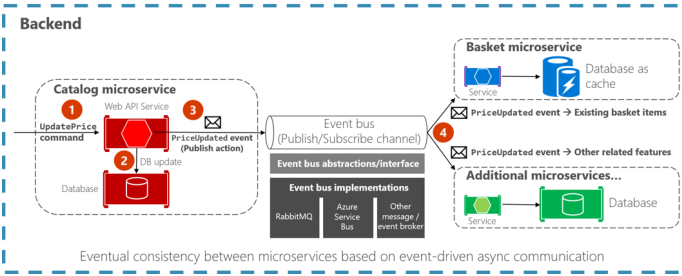
eShopOnContainers reference application

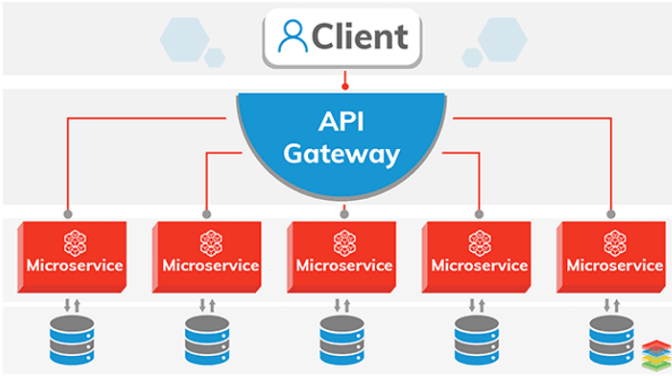
(Development environment architecture)





Implementing asynchronous event-driven communication with an event bus

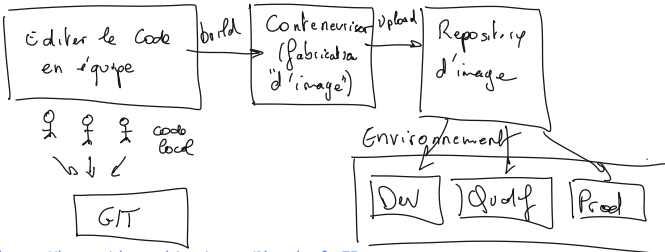




Lien vers l'implémentation du framework Istio (Microservices sous K8s)
https://www.youtube.com/watch?v=GHYdOpGUJ3w

Création d'image
Le cluster Kubernetes
La plateforme OpenShift (mise en pratique de votre côté)
BigData / plateforme Hadoop
DevOps

Le Processus de dev.



https://learngitbranching.js.org/?locale=fr_FR

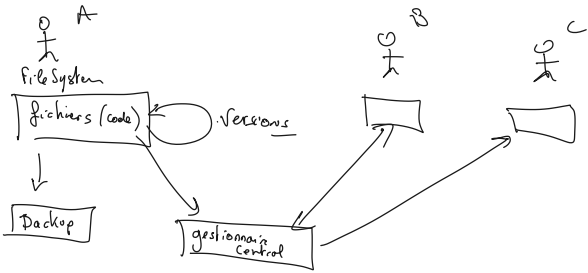
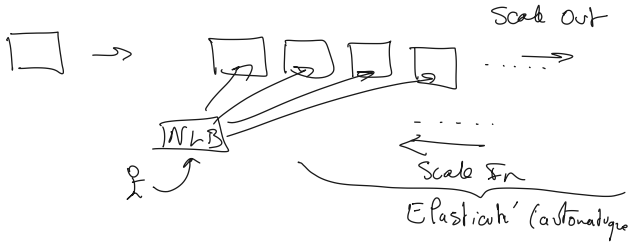


Image → flux binaire qui représente une distribution conteneurisée

1 Conteneur = SLA 99,9% → Pas assez -
et saturation sur la montée en charge

Solution : - Scale Up (pas adapté, limité)
- Scale Out



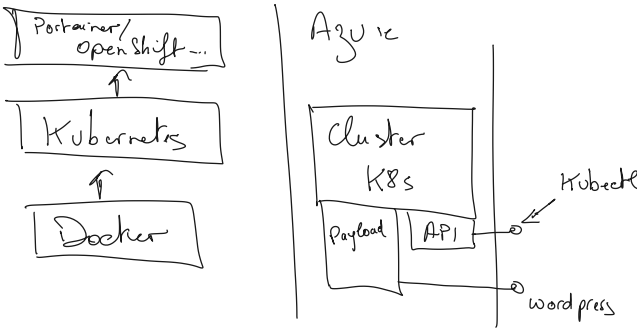
Solution Cluster de Conteneurs

Solutions : - Docker Swarm → gratuit
limité fonctionnellement
- Pas idéal en Prod

- Docker EE → Payant puissant
Interface outils
Enterprise class -

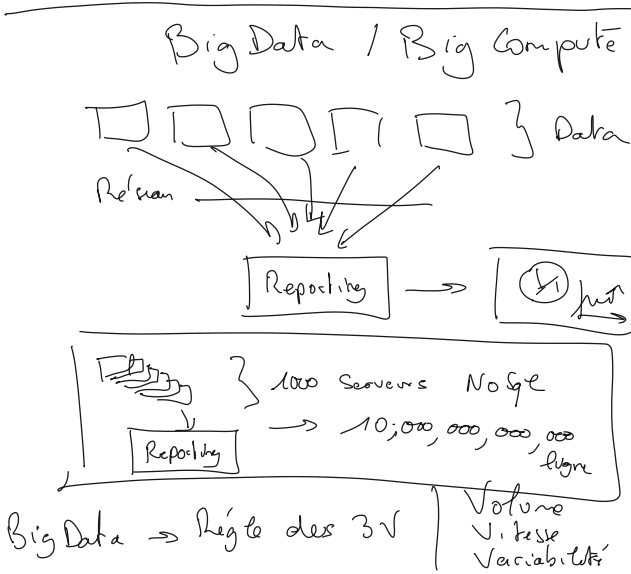
- Portainer / Rancher / ...

- Retenue généralement - Kubernetes
K8s
- Projet Open Source piloté par
Google - Natif, fiable, gratuit
Outil Technique



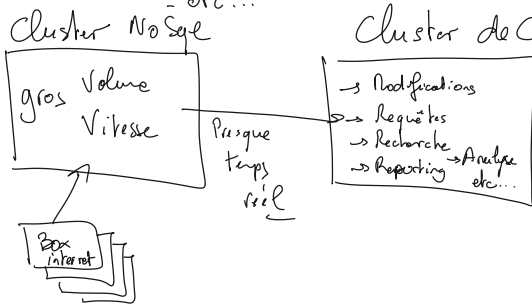
Exercice :
Implémentez le déploiement d'une application conteneurisée dans un cluster Kubernetes dans une plateforme OpenShift.
Tuto : https://developers.redhat.com/courses/openshift/getting-started

---

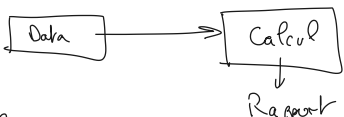


Big Data = implémentation de (cluster de) base de donnée NoSql

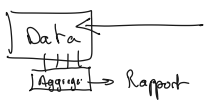
Catégories
- clef-valeur (ex : Redis)
- Document (ex. Mongo DB)
- Column Store (ex. Cassandra)
- Graph (ex Neo 4J)
- etc...



Référentiel



Big Compute



Programation (Niveaux)

N1 Base: Assembleur

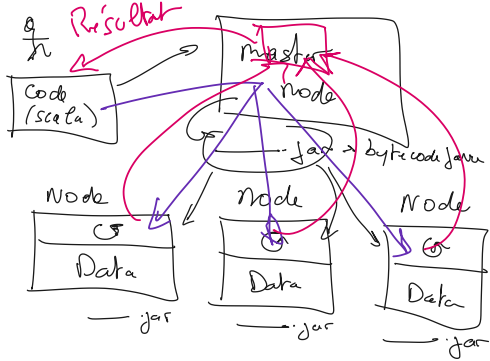
N2 Naturelle: Scripting, Basic, Cobol

N3 Prog structuré C, pascal, php, ...

N4 Prog Orienté objet C++, C#, java
Python Php objet

N5 Prog fonctionnelle fonction ({ code })
fonction → f° → f°
→ F#, scala, etc...
(java, C#, ...)

Hadoop



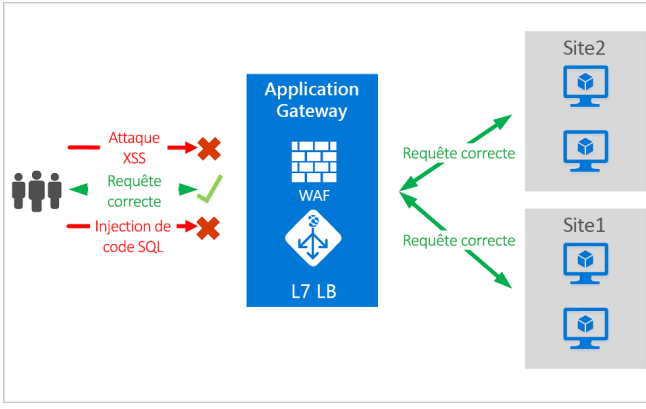
Un cluster Hadoop Gratuit dans le cloud sans installation :
Datbricks Community Edition
https://community.cloud.databricks.com/login.html

WAF - Web Application Firewall

Code → _ bugs
_ failles → exploit

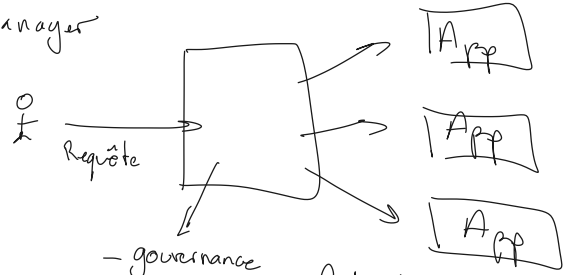
Se prémunir → failles Tierces (CVE)
ex: Log4shell
https://cve.mitre.org/

→ failles de codage (OWASP)
https://owasp.org/www-project-top-ten/

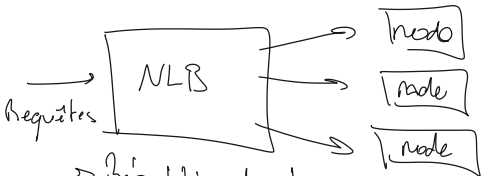


_ ReverseProxy → Routeur Simple de bas Niveau (L3 - L7)
ex: apache httpd, nginx, F5 ... gratuit/Payant

- App Manager API



- Network Load Balancer

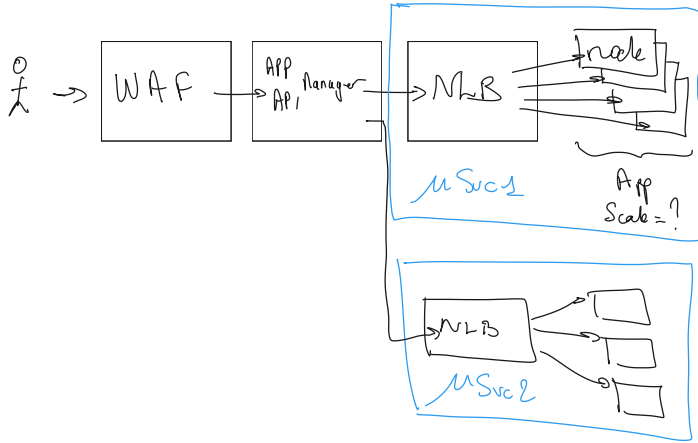


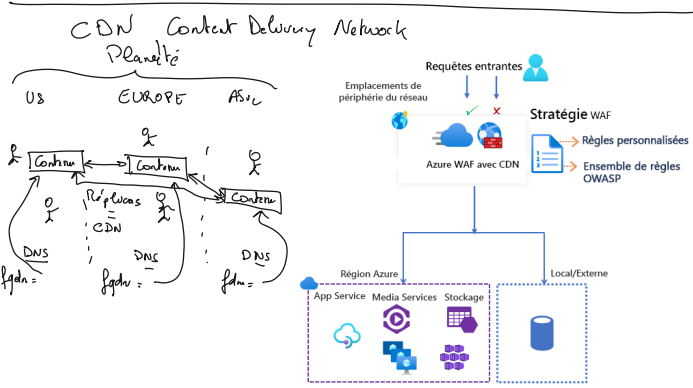
→ Répartition de charge en fonction de critères -

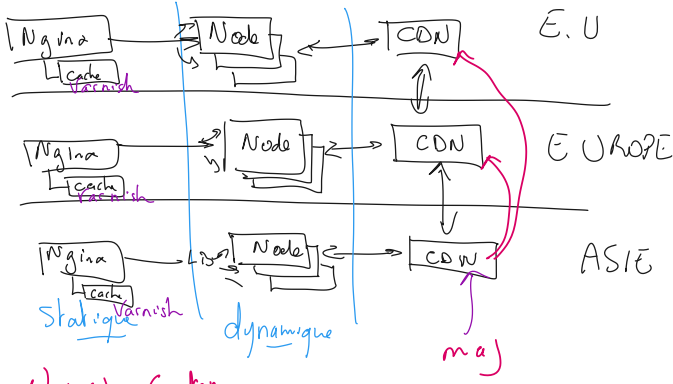
- Web App firewall

⇒ Puyant

⇒ CPU / disque


WAF
APP
API
Manager
NLB
node
μSvc1
App
Scale = ?
NLB
μSvc2


CDN Content Delivery Network
Planète
US
EUROPE
ASIE
Content
Content
Content
Réplicas
CDN
DNS
DNS
DNS
fqdn =
fqdn =
fqdn =
Requêtes entrantes
Emplacements de périphérie du réseau
Stratégie WAF
Azure WAF avec CDN
Règles personnalisées
Ensemble de règles OWASP
Région Azure
Local/Externe
App Service
Media Services
Stockage



1) maj Contenu
2) attente fin Réplica CDN
3) Purge des cache Varnish -

Virtualisation | de bureau
d'application

Remote Desktop