Introduction à Docker

Balthazar Rouberol - OVH

https://ovh.to/pvkQ5Kp

Présentation rapide

- Début à OVH en 2015
- Tech lead Docker chez OVH depuis avril 2016

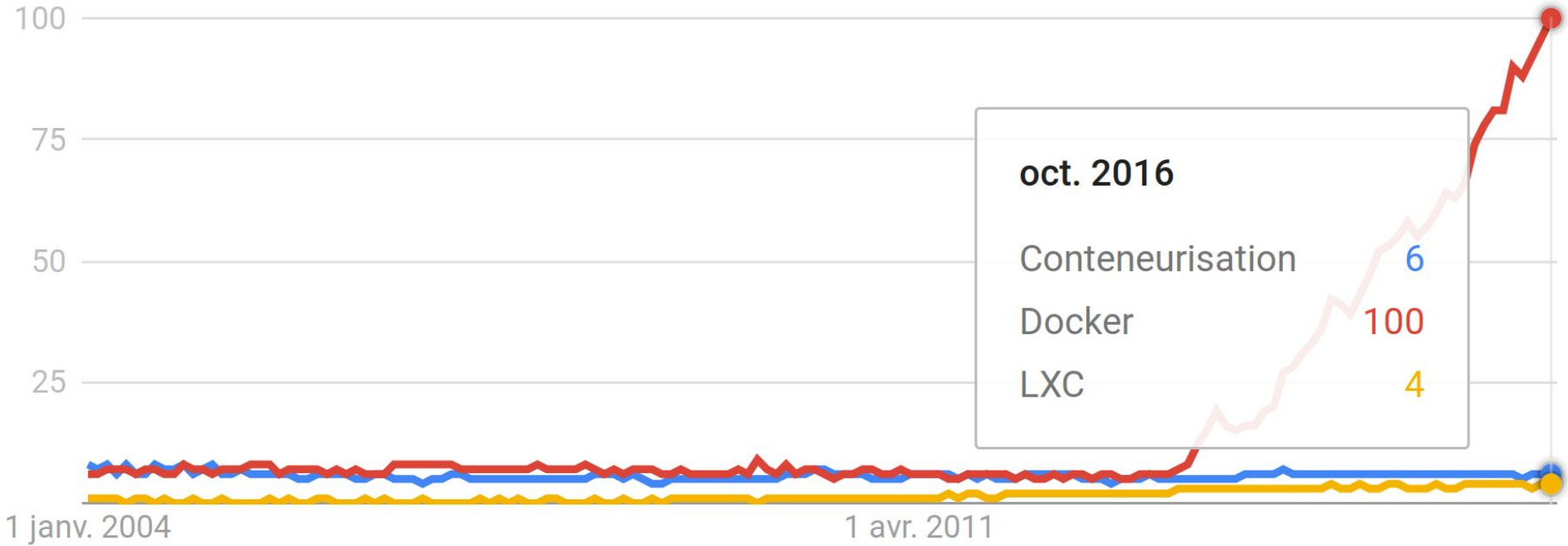
Conteneurisation

But de la Conteneurisation

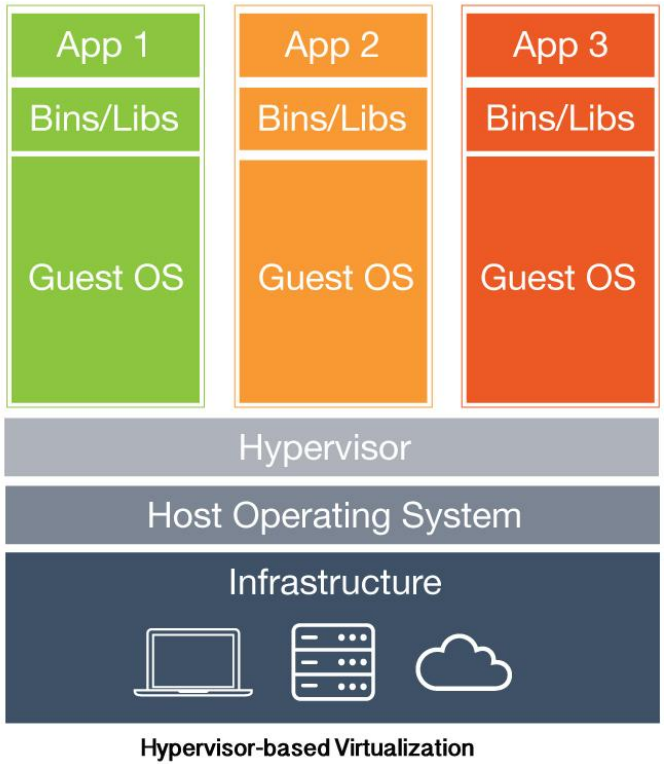
- **Isoler** un processus et ses dépendances dans une unité auto-contenue (conteneur)
- Imposer des **limites de ressources** (CPU, RAM, I/O, etc) à ce processus
- Isolation des processus: diminution de la surface d'attaque (sécurité accrue)

Isolation

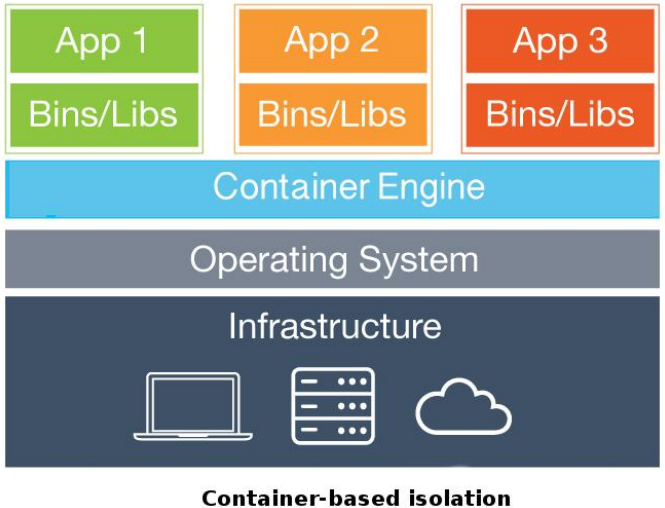
- Contrôle et limitation de la consommation de ressources (CPU, RAM, I/O)
- Isolation réseau (IP, routage, firewall unique par conteneur)
- Isolation filesystem
- Isolation des utilisateurs/groupes
- Isolation des process


100
75
50
25
oct. 2016
Conteneurisation 6
Docker 100
LXC 4
1 janv. 2004
1 avr. 2011

Conteneur != VM: architecture



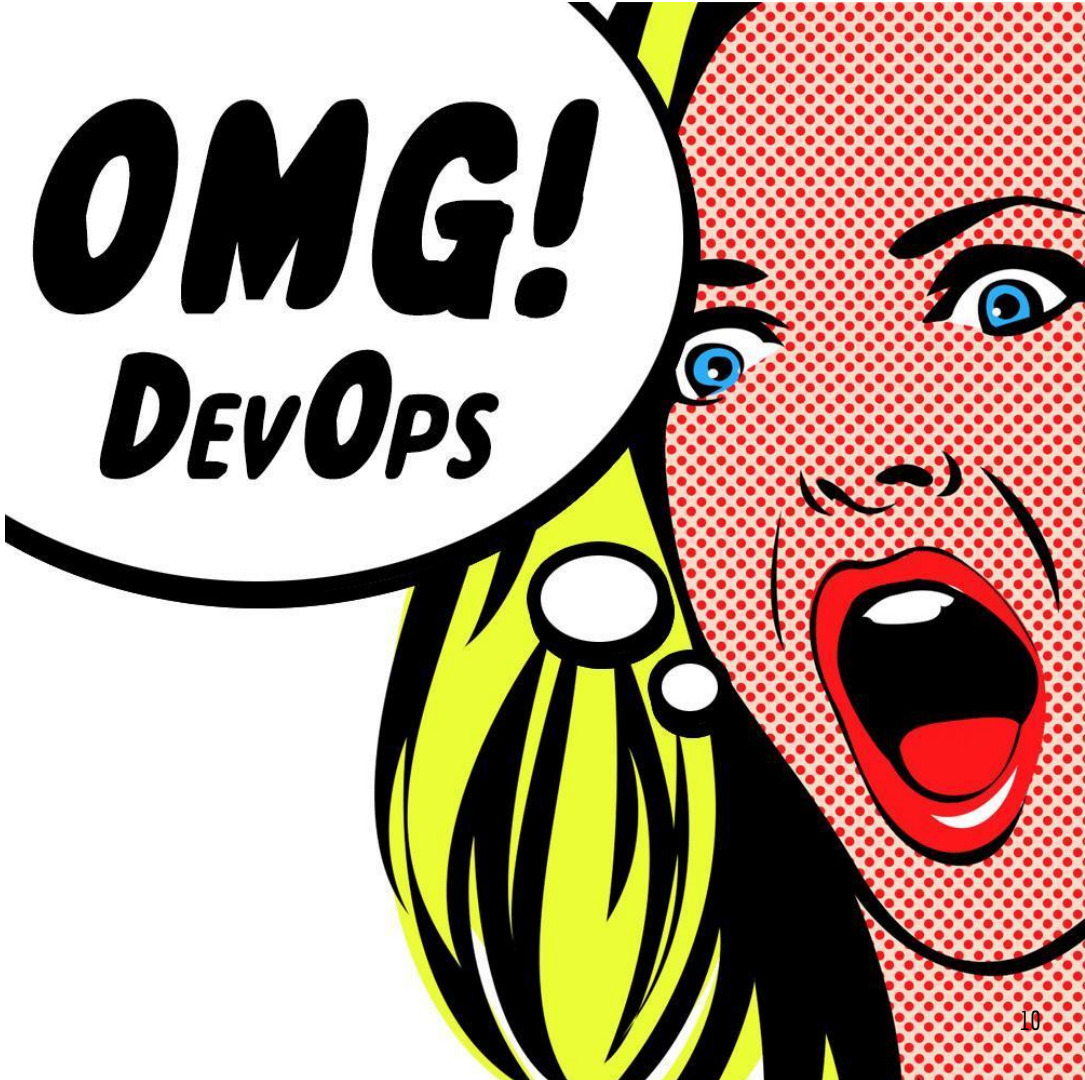
Hypervisor-based Virtualization



Container-based isolation

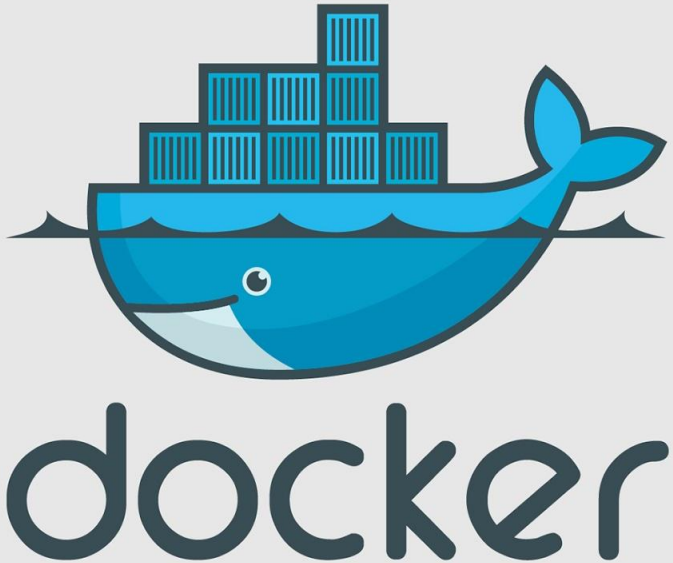
Conteneur != VM: provisionnement

- Une VM est provisionnée en secondes/minutes (le temps de booter l'OS *guest*)
- Un conteneur est lance en **millisecondes**


OMG!
DevOps

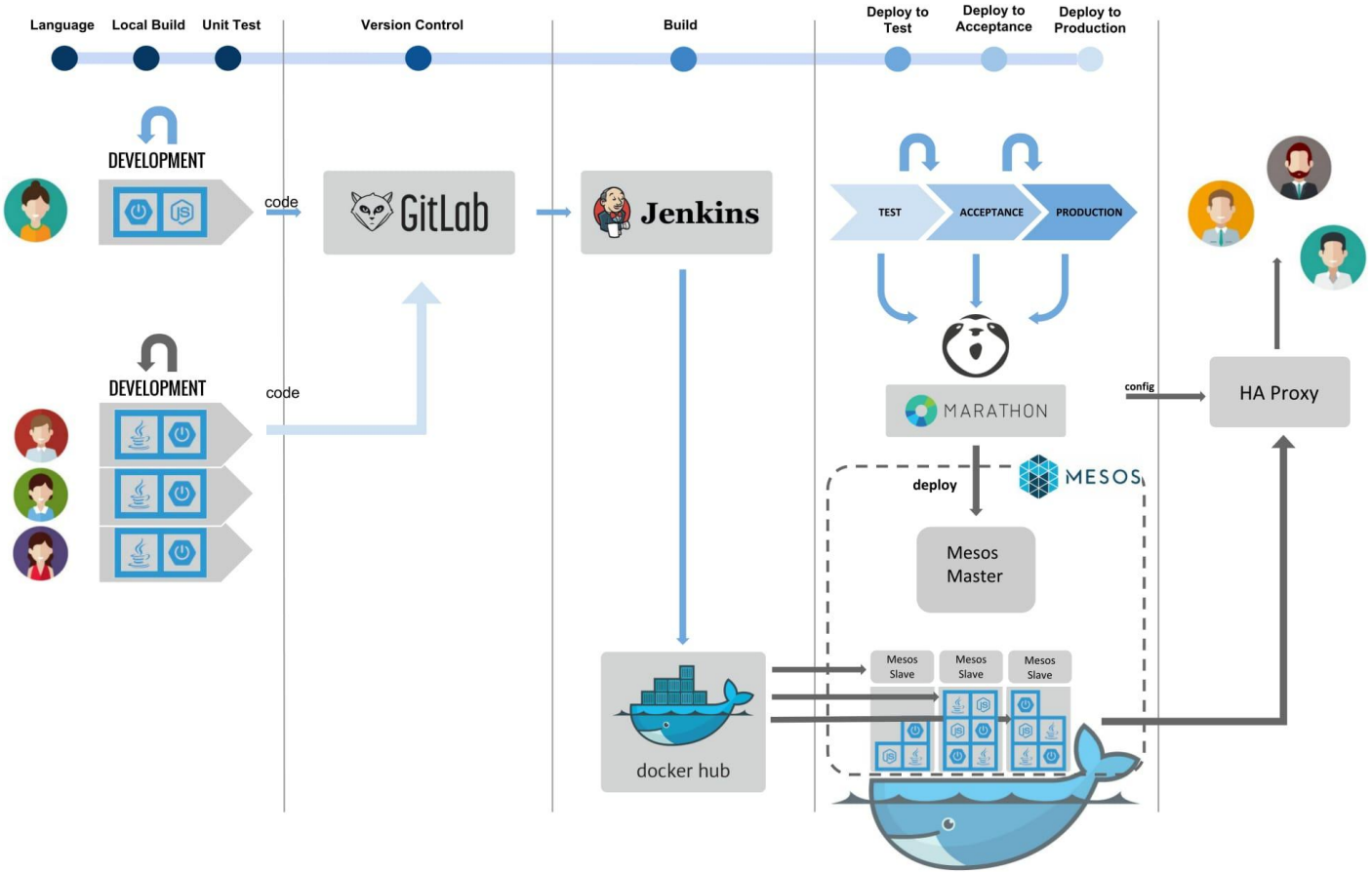
«DevOps promotes a set of processes and methods for thinking about communication and collaboration between development, QA, and IT operations.»

https://en.wikipedia.org/wiki/DevOps


docker

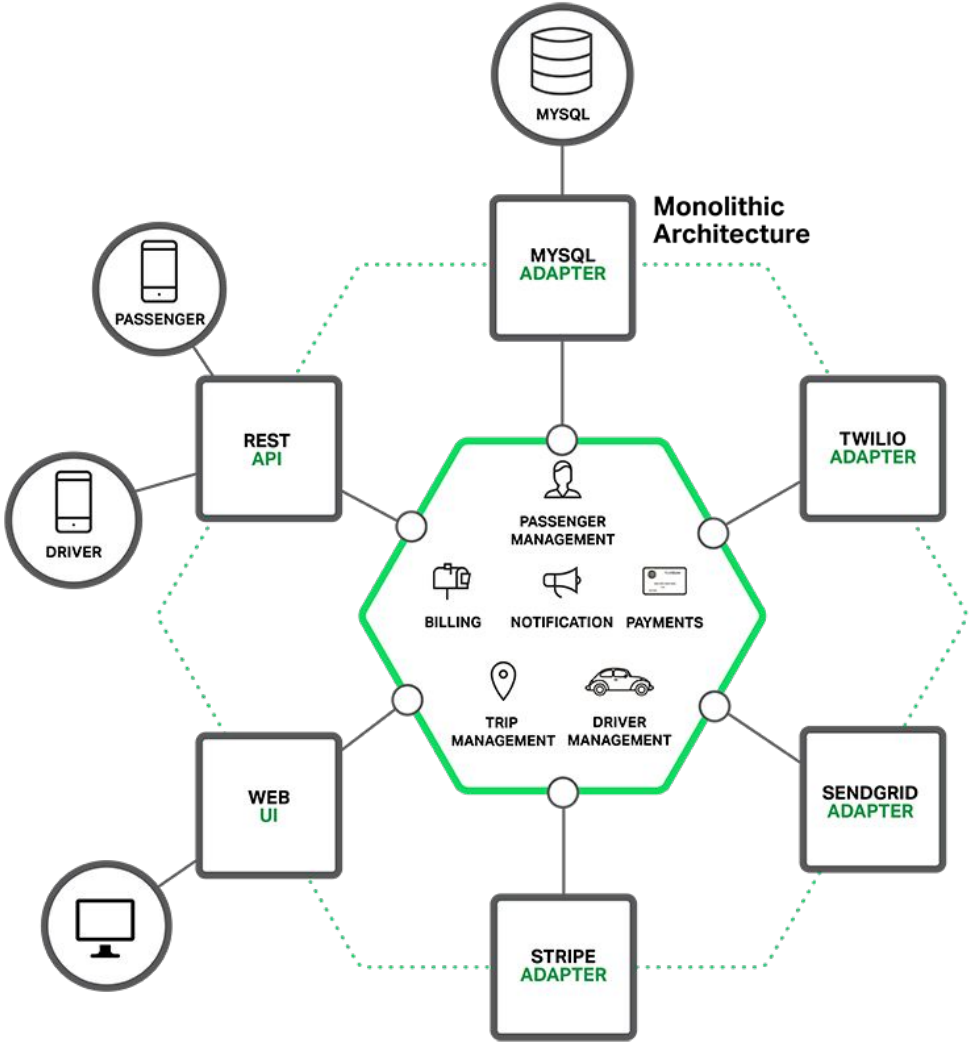
Cas d'utilisation #1
Intégration et déploiement continu

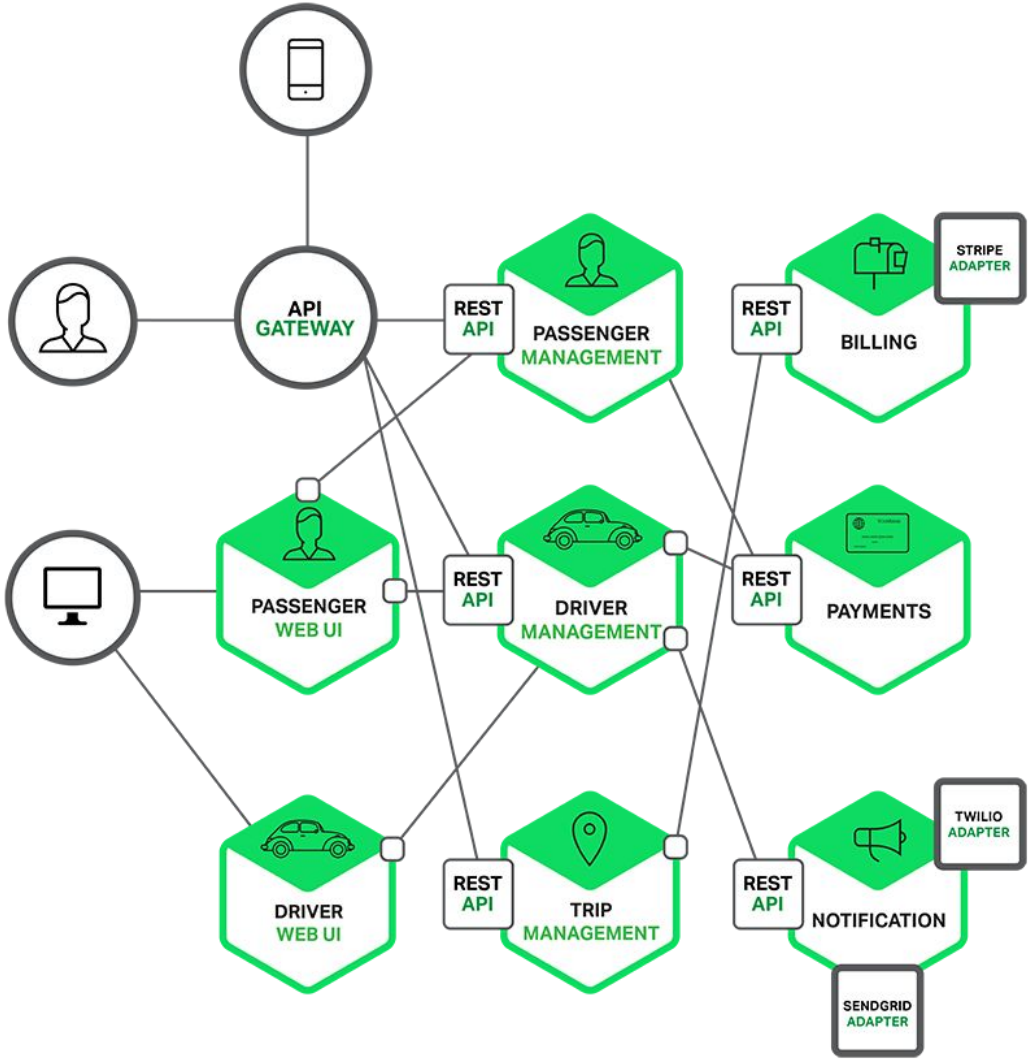
CONTINUOUS DELIVERY WITH MESOS & DOCKER



Cycle de déploiement **rapide** et **automatique**

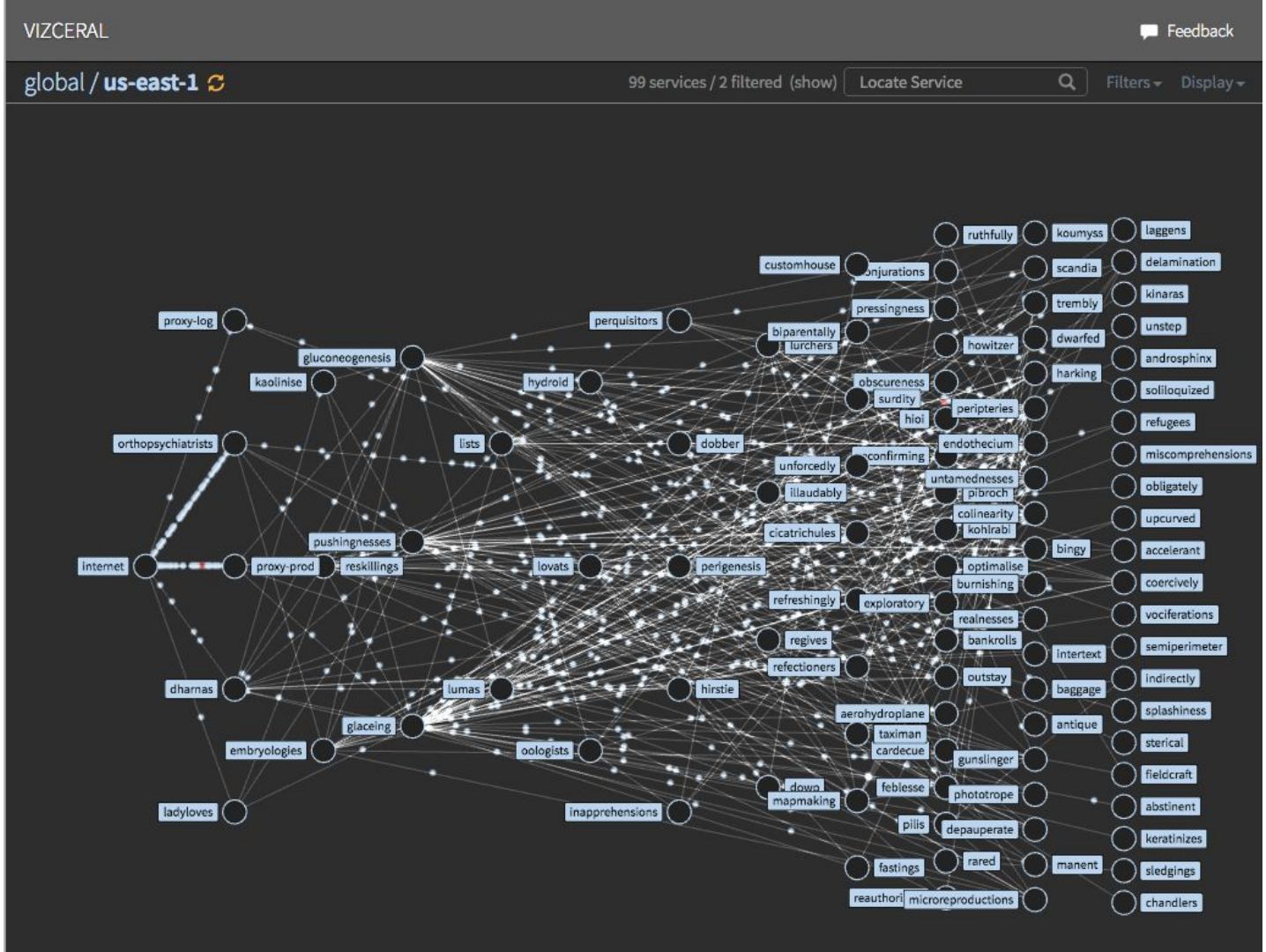
Cas d'utilisation #2
Architecture monolithique VS micro-services


MYSQL
Monolithic Architecture
MYSQL ADAPTER
PASSENGER
REST API
DRIVER
TWILIO ADAPTER
PASSENGER MANAGEMENT
BILLING
NOTIFICATION
PAYMENTS
TRIP MANAGEMENT
DRIVER MANAGEMENT
WEB UI
SENDGRID ADAPTER
STRIPE ADAPTER


API
GATEWAY
REST
API
PASSENGER
MANAGEMENT
REST
API
BILLING
STRIPE
ADAPTER
PASSENGER
WEB UI
REST
API
DRIVER
MANAGEMENT
REST
API
PAYMENTS
DRIVER
WEB UI
REST
API
TRIP
MANAGEMENT
REST
API
NOTIFICATION
TWILIO
ADAPTER
SENDGRID
ADAPTER

Architecture micro-service & Docker

- Un micro-service = une image, 1+ conteneur
- Langage de programmation (potentiellement) différent par service
- Equipe (potentiellement) différente par service
- Cycle de vie et de déploiement différent par service
- Scaling par service, en fonction de la charge
- Communication inter-service via des API (REST/SOAP) ou protocoles tels que gRPC


VIZCERAL
Feedback
global / us-east-1
99 services / 2 filtered (show)
Locate Service
Filters
Display
proxy-log
gluconeogenesis
kaolinise
orthopsychiatrists
internet
proxy-prod
pushingnesses
reskillings
dharmas
embryologies
ladyloves
glaceing
perquisitors
hydroid
lists
lovats
lumas
oologists
inapprehensions
customhouse
biparentally
lurchers
pressingness
onjurations
ruthfully
howitzer
obscureness
surdity
hloi
peripteries
dobber
endothecium
unforcedly
confirming
illaudably
untamednesses
pibroch
cicatrichules
colinearity
kohlrabi
perigenesis
optimalise
burnishing
refreshingly
exploratory
realnesses
regives
bankrolls
refectioners
outstay
hirstie
aerohydroplane
taximan
cardecue
feblesse
mapmaking
phototrope
pilis
depauperate
rared
fastings
reauthori
microreproductions
koumyss
scandia
trembly
dwarfed
harking
bingy
intertext
baggage
antique
gunslinger
manent
laggens
delamination
kinaras
unstep
androsphinx
soliloquized
refugees
miscomprehensions
obligately
upcurved
accelerant
coercively
vociferations
semiperimeter
indirectly
splashiness
sterical
fieldcraft
abstinent
keratinizes
sledgings
chandlers

Orchestrateur?

- Déploiement multi-nodes
- Dépendances de déploiement
- Contraintes de répartition
- Réseaux privés
- Monitoring de santé des conteneurs
- Redéploiement des conteneurs “malades”
- Load balancing du trafic
- etc

Orchestrateurs open source

- Kubernetes (k8s, Google)
- Docker Swarm (Docker Inc)
- Marathon (Mesosphere)
- Rancher (Rancher Labs)

Utilisation de Docker chez OVH

L’infrastructure en quelques chiffres

- 7 clusters Mesos/Marathon/Docker
  - Production interne: 3
  - Production externe: 2
  - Beta/Gamma externe: 2
- 800 hosts
- 3000 cores
- 12TB RAM
- 200 TB disque
- 3 registry Docker (externe/interne/équipe)
- Des centaines de déploiements par jour
- 5 personnes

Merci!
Questions?