

**Bla Bla Car**

**Monitoring  
Host & Container**

**Karles**

14 juin 2017



# Karles

Site Reliability Engineer

France

@BlaBlaCar since 2015

# Host / Pod / Container

---

## Host

Un serveur physique  
Bare Metal Restore

## Conteneur Logiciel

Un espace logique mono ou  
multi-service

## Pod

Un contenant comprenant plusieurs  
conteneurs logiques distincts

# Stateful / Stateless

---

## Stateless

Sans donnée conservée entre deux version du conteneur

## Stateful

Avec des données conservées entre deux version du conteneur

# Host centric ?

---

**Comment  
monitorer des  
services en  
container avec un  
outil host centric  
comme zabbix ?**

- 1 Monitoring des hosts physiques via un container / pod dédié
- 2 Monitorer les pod multi-conteneur / stateful comme des hosts physiques
- 3 Monitorer les conteneurs solo / stateless via l'ordonnanceur ou le discovery
- 4 Alerting classique

# Host

---

## Conteneur agent de monitoring

- ✔ Contenant l'agent Zabbix patché ( pour accès au /dev, /sys et /proc du host sous-jacent )
- ✔ Comprenant le contrôle du hardware via les outils Dell Open Manage
- ✔ Relevant les métriques systèmes classiques

# Containers

---

## Via l'ordonnanceur (fleetd) ou le discovery (zookeeper)

- ✔ Conteneur agent zabbix interrogeant ordonnanceur ou le discovery
- ✔ Découvrant les conteneurs présent dans l'ordonnanceur ou les services dans le discovery.
- ✔ Auto-configuration des métriques et des seuil d'alarmes
- ✔ Alerting sur que chacun des X conteneurs de type A sont bien présent et en service
- ✔ Alerting que le 50% du pool de X conteneurs de type A est bien présent et en service

# Pod

---

## Conteneur agent de monitoring

- ✔ Un conteneur contenant l'agent Zabbix patché (pour rien, juste avoir un type unique)
- ✔ Qui interroge et relève les métriques des services présents dans les autres conteneurs adjacents dans le pod
- ✔ Alerting sur métriques services : Statut du cluster, Nb de node, Nb de processus, Port ouvert, Nb de req, Nb de message, Heartbeat, etc.



# Alerting Zabbix

---

## Classiquement

- ✓ Un message envoyé dans un channel slack
- ✓ Un email envoyé sur les boites email d'astreinte
- ✓ Un téléphone qui sonne à la réception d'un email

# Météologie

---

## Prometheus

“time serie”, “cloud native”, “service discovery”, “scalable”, “alerting”

## Discovery

Patch Prometheus pour le format  
Json de nerve (registration  
daemon) stocké dans zookeeper

## Graphique

Grafana

# Service Centric

---

**Ne plus surveiller  
les applicatifs  
mais le service  
rendu (métrologie  
business)**

1

Monitoring backend / Host : Continuer à relever les métriques systeme, service



2

Monitoring business : compter des événements depuis l'application



# Monitoring backend / host

---

## Exporter Prometheus

- ✔ Comme les hosts, check des services et de leur environnements : Statut du cluster, Nb de node, Nb de processus, Port ouvert, Nb de req, Nb de message, Heartbeat, etc.
- ✔ Relevé toutes les 20 secondes
- ✔ Graphique par grafana

# Monitoring business

---

## Heka-Exporter

- ✓ Recevoir et compter des événements (Req, Call API, Publication, Paiement, Réservation etc ..)
- ✓ Ceci n'est pas de l'instrumentation du code car il n'y a pas de relevé de durée de traitement (loop, fonction, objet, appel backend)
- ✓ Un collecteur de log (heka) reçoit et envoie l'événement à un exporter Prometheus
- ✓ L'heka-exporter compte les événements
- ✓ L'exporter est interrogé par Prometheus (remise à zero du compteur)

# Alerting Prometheus

---

## Exporter Prometheus

- ✔ Prometheus alert rules pour l'enrichissement des évènements (géré par les dev)
- ✔ Alerting par Alerte Manager avec routing vers l'équipe responsable (géré les architectes via des PR)
- ✔ Prometheus alert rules pour l'enrichissement des évènements (géré par les architectes)
- ✔ Alerting par Alerte Manager avec routing vers l'équipe responsable (géré les architectes via des PR)
- ✔ Envois de message slack et d'email. Un téléphone qui sonne à la réception d'un email.