TX Picasoft A16

Mise en place d'une architecture expérimentale pour héberger des services dans le cadre du mouvement de redécentralisation d'Internet

Grégoire Martinache / Antoine Barbare

utc
SORBONNE UNIVERSITES

Plan

1. Architecture globale
2. Docker Engine/Swarm
3. Architecture des services
4. Cluster Proxmox
5. Opérations de maintenance
6. Sauvegarde
7. Monitoring

Architecture Globale

utc
SORBONNE UNIVERSITES

Architecture globale

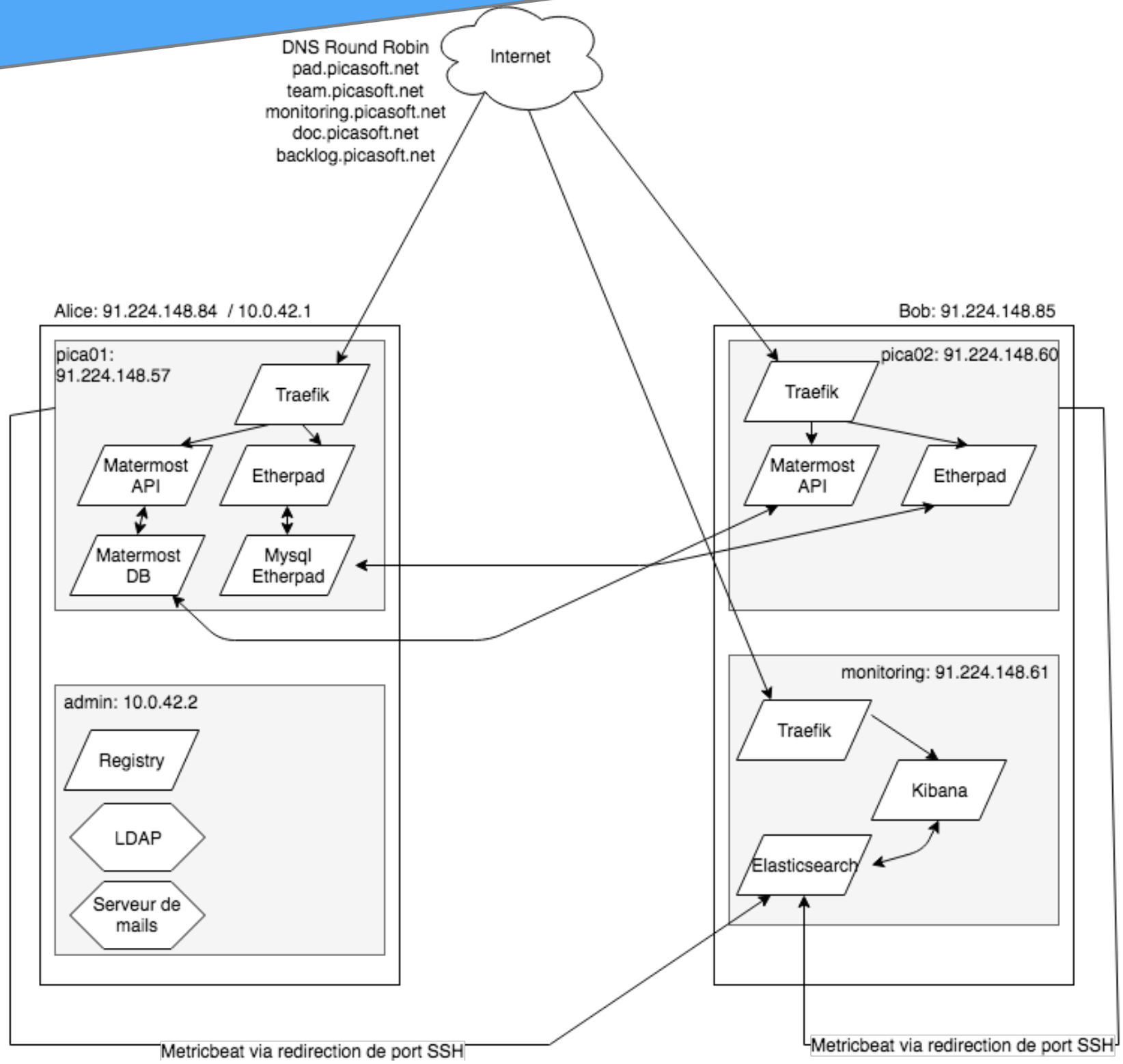
Deux machines physiques au format tour
Hébergées à Toulouse

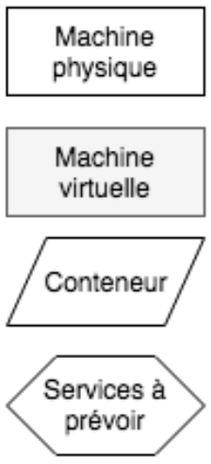
tetaneutral.net

- Carte mère ASRock Q170M vPro
- Intel Core i5-6500 (3.2 GHz)
- 32Go Crucial DDR4
- 2x WD Red 2 To en RAID 1
- 2x SanDisk SSD Ultra II 480G en RAID 1

Architecture globale

2 hyperviseurs Proxmox
4 machines virtuelles
- 2 orientée services
- 2 orientés administration





Architecture globale

Routage des adresses IP :

5 adresses IPV4 flottantes en plus des IPV4 des hyperviseurs

3 IP routées sur Alice
- 91.224.148.57 (pica01 services)
- 91.224.148.58 (en attente)
- 91.224.148.59 (en attente)

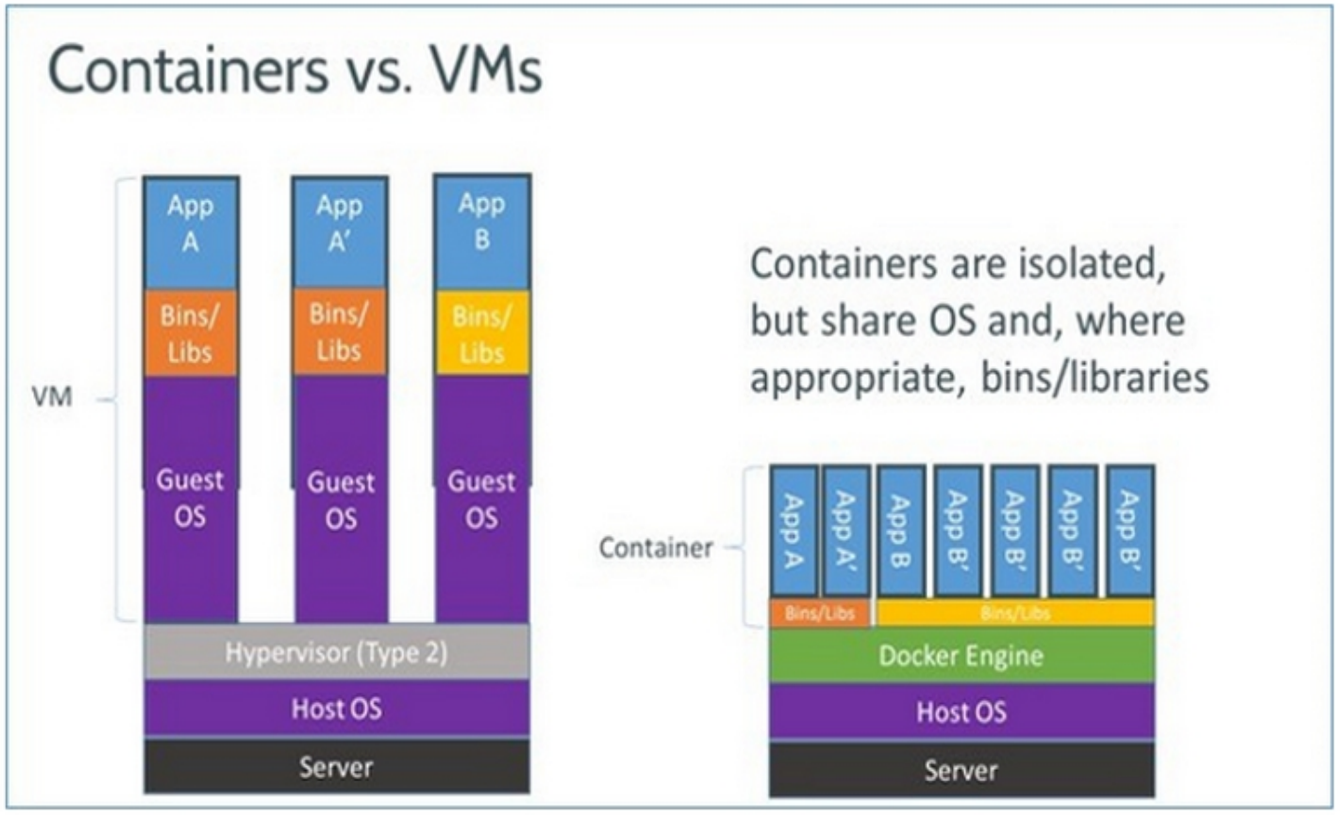
2 IP routées sur Bob
- 91.224.148.60 (pica02 services)
- 91.224.148.61 monitoring

Pour les services internes, on privilégiera l'utilisation d'IP privées et de NAT. Les IPV4 flottantes sont à réserver pour des services publiques ayant vocation à être exposées.

Docker Engine/Swarm

utc
SORBONNE UNIVERSITES

C’est quoi Docker ?



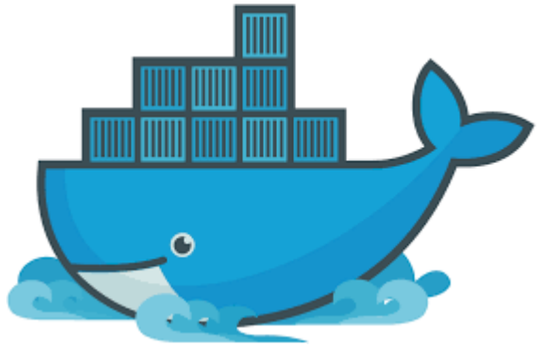
De nombreux avantages :

Une grande légèreté

Une migration facile

Portable (Registry)

Swarm



utc
SORBONNE UNIVERSITES

Comment ca marche ?

Création d'une image Docker : → Le Dockerfile

```
1  # explicitly use Debian for maximum cross-architecture compatibility
2  FROM debian:jessie-slim
3
4  RUN apt-get update && apt-get install -y --no-install-recommends \
5                  gcc \
6                  libc6-dev \
7                  make \
8          && rm -rf /var/lib/apt/lists/*
9
10 WORKDIR /usr/src/hello
11 COPY . .
12
13 RUN set -ex; \
14         make clean all test; \
15         find \( -name 'hello' -or -name 'hello.txt' \) -exec ls -l '{}' +
16
17 CMD ["./hello-world/hello"]
```

docker build -t nomImage .

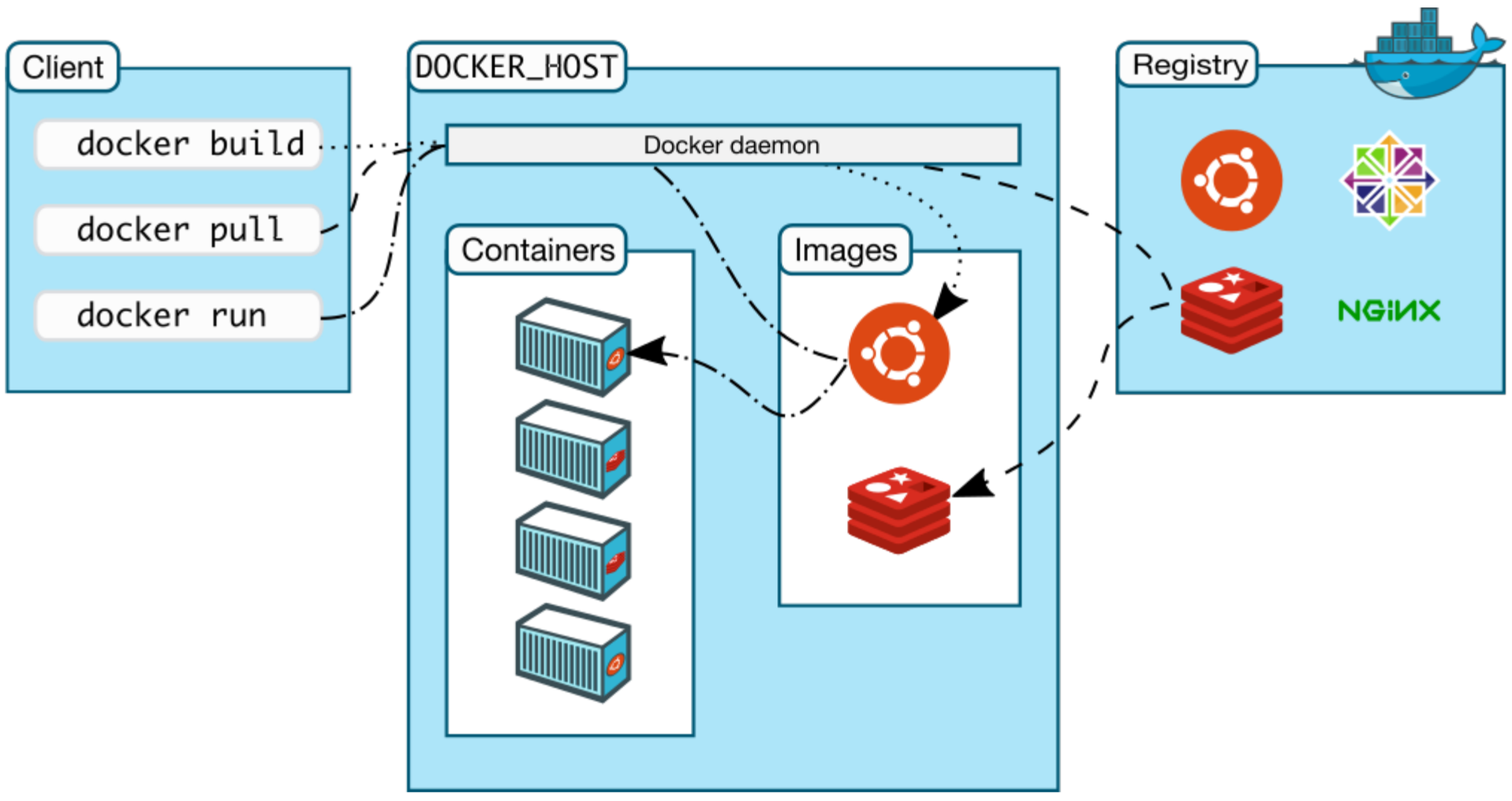
utc
SORBONNE UNIVERSITES

Création d’un container :

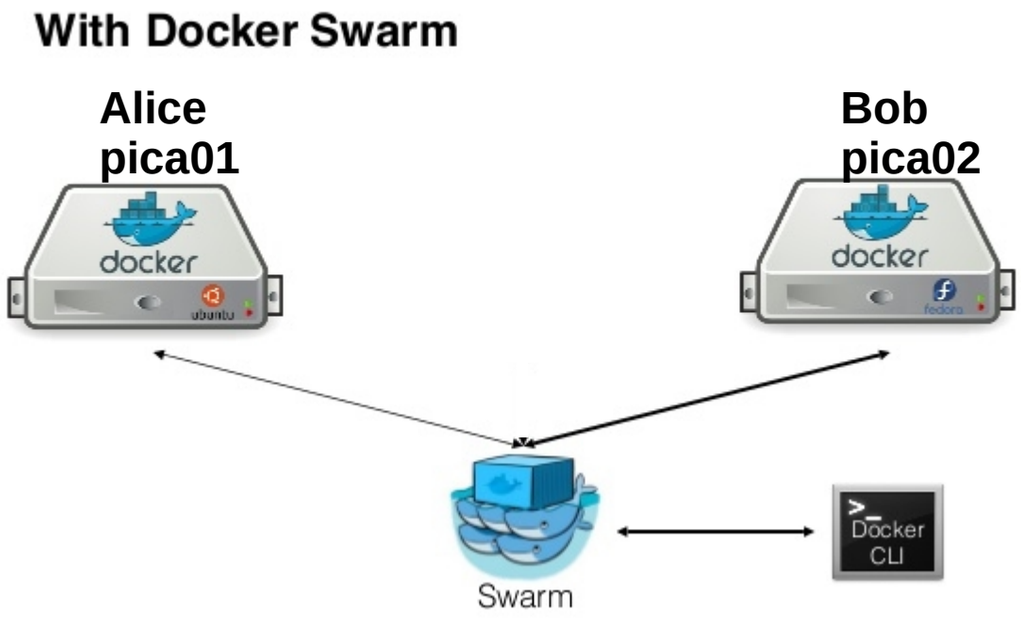
docker run --name superContainer -p 80:80 monImage

```
root@pica01:~# docker ps
CONTAINER ID        IMAGE                                                COMMAND                  CREATED             STATUS              PORTS               NAMES
983bfab21081        traefik:latest                                       "/traefik --docker --"   6 days ago          Up 6 days           80/tcp              traefi
03440aea61c5        registry.picasoft.net:5000/pica-kanboard:latest      "/bin/sh -c /start.sh"   6 days ago          Up 6 days           80/tcp              kanboa
befdc71f0aab        registry.picasoft.net:5000/pica-dokuwiki:latest      "/bin/sh -c /start.sh"   6 days ago          Up 6 days           80/tcp              doc.2.
ed088f8db8dd        registry.picasoft.net:5000/pica-nginx:latest         "/bin/sh -c /start.sh"   6 days ago          Up 6 days           80/tcp              picaso
42fe63411ba9        registry.picasoft.net:5000/postgres-backup:latest    "/run.sh"                6 days ago          Up 6 days                               postgr
29ef97460983        registry.picasoft.net:5000/pica-mattermost-db:latest "/docker-entrypoint1."   6 days ago          Up 6 days           5432/tcp            matter
c565f2be88bc        registry.picasoft.net:5000/mysql-backup:latest       "/run.sh"                6 days ago          Up 6 days                               mysql-
```

Les registry :



Docker Swarm




```
root@pica01:~# docker node ls
ID                            HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS
9c3xnjjo1u6mbol5qtbeh92cb     pica02    Ready   Active        Leader
alpfdvrzjrbiuvya7nyiwhbjg *   pica01    Ready   Active        Reachable
```

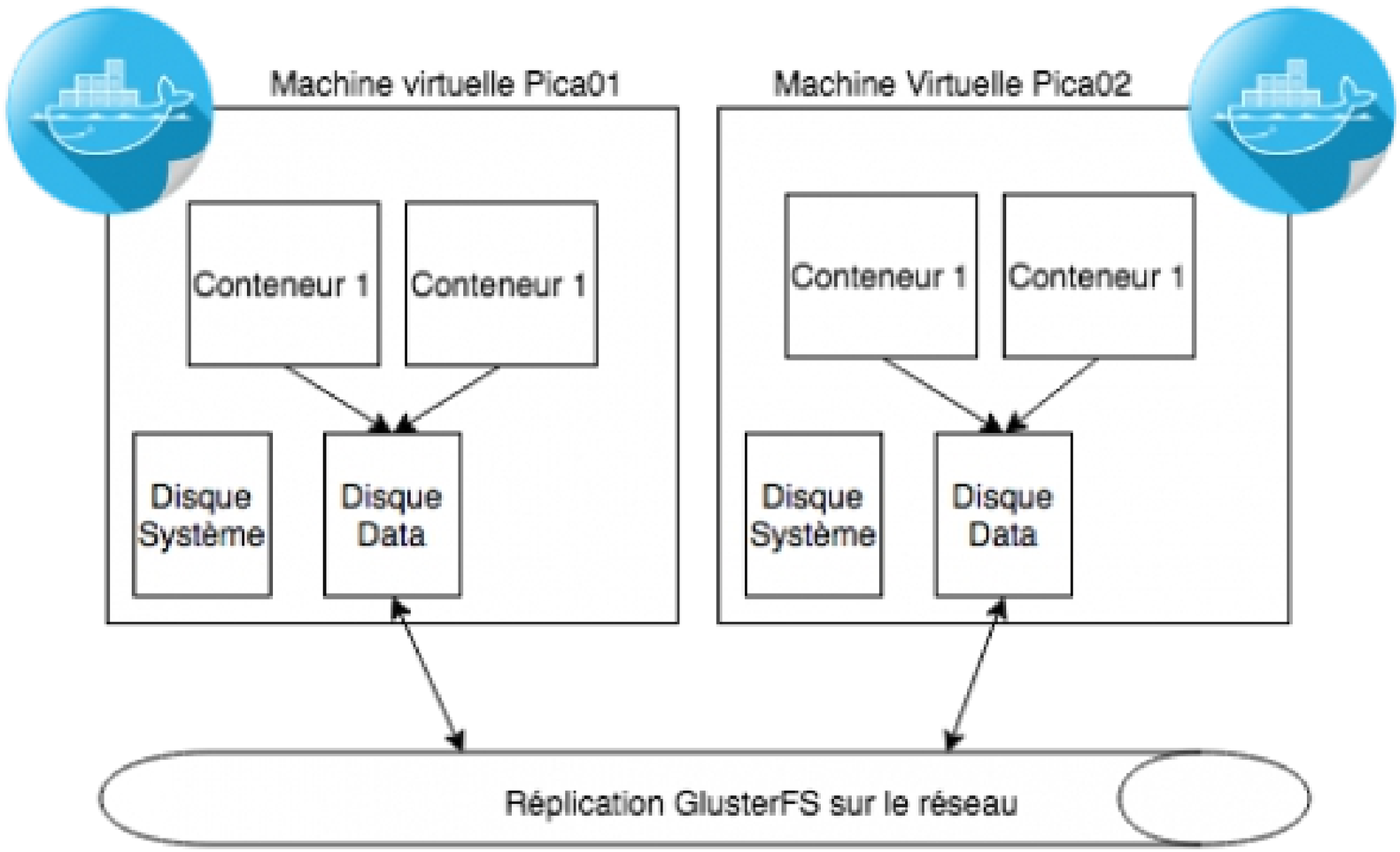
=> Répartition des containers

Docker Swarm

```
#!/bin/bash
docker service create \
--name mattermost \
--replicas 1 \
--label traefik.port=80 \
--label traefik.frontend.rule=Host:team.picasoft.net \
--mount type=bind,source=/DATA/MIDDLE/mattermost/config,target=/mattermost/config \
--mount type=bind,source=/DATA/MIDDLE/mattermost/data,target=/mattermost/data \
--mount type=bind,source=/etc/localtime,target=/etc/localtime \
-e DB_HOST=[redacted] \
-e MM_USERNAME=[redacted] \
-e MM_PASSWORD=[redacted] \
-e MM_DBNAME=mattermost \
--network pica-net \
registry.picasoft.net:5000/pica-mattermost
```

utc
SORBONNE UNIVERSITES

GlusterFS :

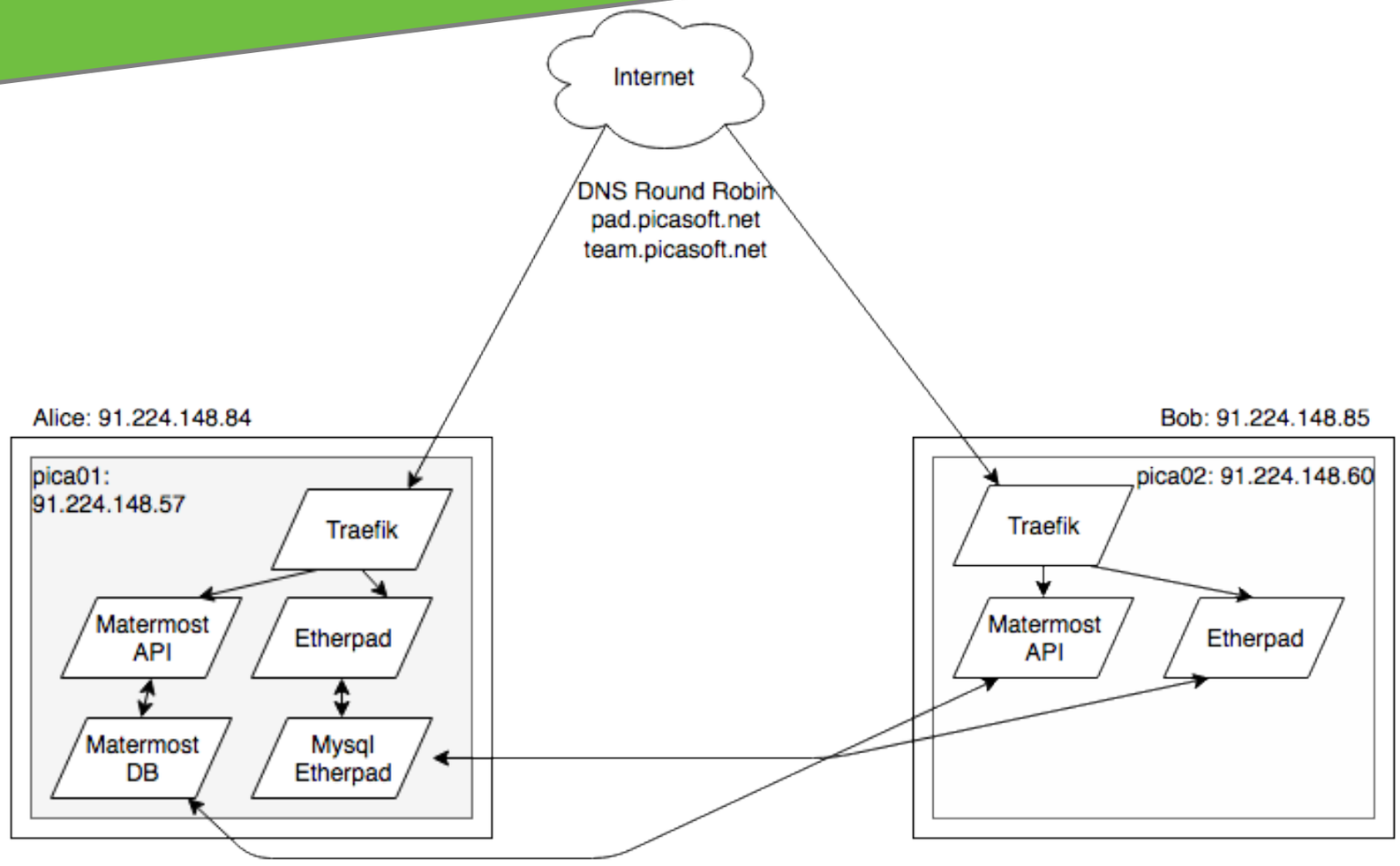


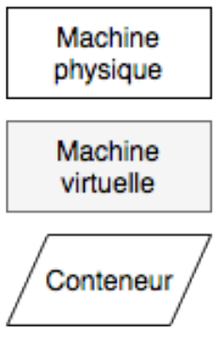
utc
SORBONNE UNIVERSITES

Architecture des Services

utc
SORBONNE UNIVERSITES

Architecture des services





Round Robin DNS :
team.picasoft.net IN A 91.224.148.57
team.picasoft.net IN A 91.224.57.60

Répartir la charge dans un premier temps.
Attention, en cas d'arrêt d'une machine, les services basculent mais les DNS pointent toujours sur la machine stoppée.

utc
SORBONNE UNIVERSITES

Architecture des services

Traefik : Load balancer et reverse proxy automatique

En écoute sur le port :80 et :443 de chaque machine
Mapping des services en fonction de la requête

team.picasoft.net → conteneur 1
pad.picasoft.net → conteneur 2 ou conteneur 3 (Load balancing)

En écoute sur le démon Docker
Génération de la configuration à la volée et configuration automatique du HTTPS

```
--label traefik.port=80
--label traefik.frontend.rule=Host:team.picasoft.net
```

utc
SORBONNE UNIVERSITES

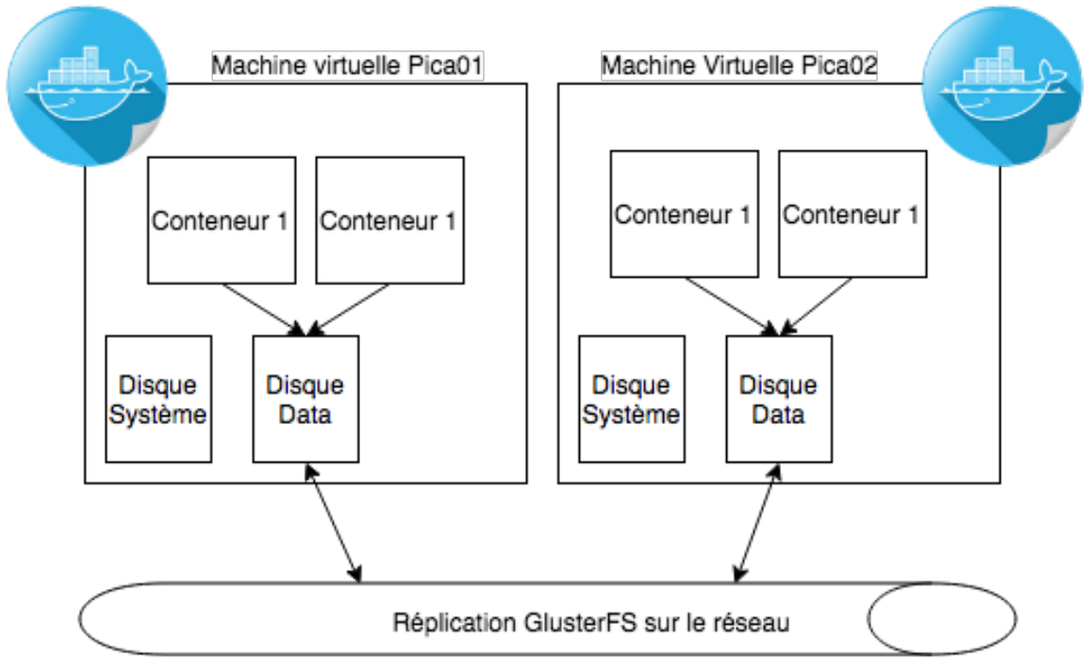
Architecture des services

GlusterFS : synchronisation des fichiers par le réseau

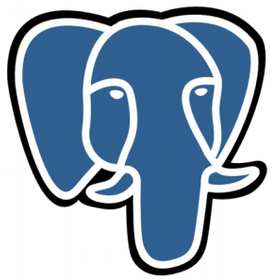
Partition /DATA des machines synchronisée et répliquée

Permet aux services sur pica01 et pica02 d'accéder à la même données

En cas de perte d'une machine, les données sont accessibles depuis l'autre machine sans interruption de service ou presque



Architecture des services

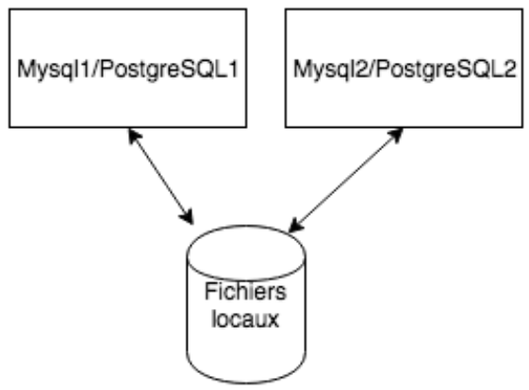


PostgreSQL

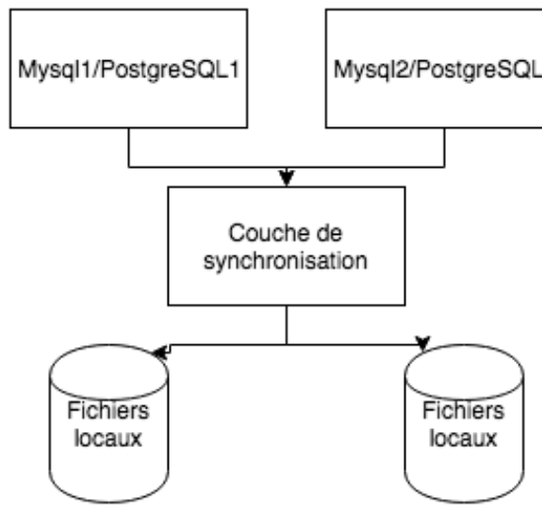
MySQL

Base de données standalone

Deux bases de données ne peuvent pas lire en même temps dans le même système de fichier



Impossible



Architecture à adopter

Si Picasoft venait à prendre de l'ampleur, c'est un point à corriger en priorité

utc
SORBONNE UNIVERSITES

Cluster Proxmox

utc
SORBONNE UNIVERSITES

PROXMOX

Une solution de virtualisation libre :
- Gestion centrale des machines virtuelles
- Solution de backup
- High Availability Cluster

utc
SORBONNE UNIVERSITES

PROXMOX Virtual Environment 4.4-5/c43015a5 Search

Server View

- Datacenter
  - alice
  - bob

Datacenter

- Search
- Summary
- Options
- Storage
- Backup
- Permissions
  - Users
  - Groups
  - Pools
  - Roles
  - Authentication
- HA
- Firewall
- Support

Type	Description	Disk usage...	Memory us...	CPU usage	Uptime
node	alice	2.3 %	21.1 %	1.6% of 4C...	8 days 17:02...
node	bob	2.2 %	40.7 %	1.6% of 4C...	8 days 17:16...
qemu	100 (template)				-
qemu	101 (pica01)		41.8 %	0.9% of 4C...	8 days 17:02...
qemu	103 (admin)		61.6 %	0.3% of 2C...	8 days 17:01...
qemu	102 (pica02)		41.1 %	0.8% of 4C...	8 days 17:16...
qemu	105 (monitoring)		95.7 %	1.2% of 4C...	8 days 17:15...
storage	hdd (alice)	12.9 %			-
storage	local (alice)	7.3 %			-
storage	save (alice)	98.0 %			-
storage	hdd (bob)	12.7 %			-
storage	local (bob)	17.8 %			-
storage	save (bob)	82.0 %			-

Tasks | Cluster log

Start Time	End Time	Node	User name	Description
Jan 08 03:00:02	Jan 08 03:24:41	bob	root@pam	Backup
Jan 08 03:00:01	Jan 08 03:08:59	alice	root@pam	Backup
Jan 07 03:00:02	Jan 07 03:09:01	alice	root@pam	Backup
Jan 07 03:00:01	Jan 07 03:24:49	bob	root@pam	Backup
Jan 06 03:00:01	Jan 06 03:08:57	alice	root@pam	Backup

utc
SORBONNE UNIVERSITES

Opérations de maintenance

utc
SORBONNE UNIVERSITES

Carte mère VPro : Accès VNC à la machine dès son démarrage via rebond chez tetraneutral.net

```
ssh -p 2222 nagios.tetaneutral.net -l ttnn -N \
-L 8081:192.168.128.151:16992 -L 5911:192.168.128.151:5900 \
-L 8082:192.168.128.152:16992 -L 5912:192.168.128.152:5900
```

Port 8081 & 8082 → Inventaire et lancement des commandes de power On/Off

```
http://localhost:8081
```

Port 5911 et 5912 → Accès VNC

```
remote-viewer vnc://localhost:5911
```

Sauvegarde

utc
SORBONNE UNIVERSITES

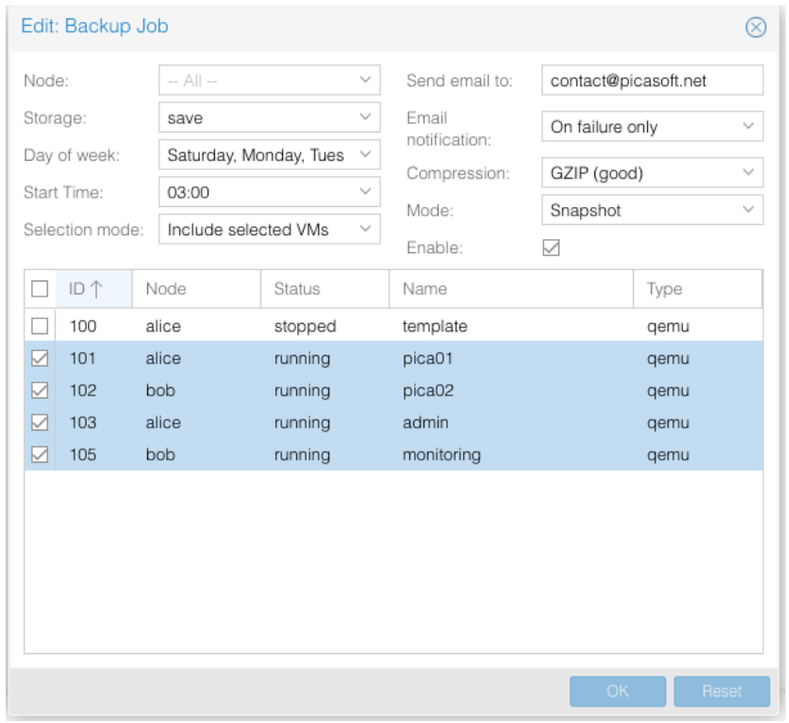
Sauvegarde

Sauvegarde des Machines

1 disque de 200G sur Alice et Bob pour les sauvegardes

1 snapshot de chaque VM à 3h chaque jour et 7 jours d'historique

Synchronisation des snapshots sur les 2 hyperviseurs



Sauvegarde des Bases de données

1 conteneur par type de base de données

1 sauvegarde toute les 6h et 7 jours d'historique

Réplication des sauvegardes sur les deux VM via le volume GlusterFS

```
root@pica01:/DATA/docker# ll -R /DATA/BACKUP/mysql/
/DATA/BACKUP/mysql/:
total 313338
-rw-r--r-- 1 root root 10611701 déc.  30 13:00 2016.12.30.120001.sql
-rw-r--r-- 1 root root 10611845 déc.  30 19:00 2016.12.30.180001.sql
-rw-r--r-- 1 root root 10611845 déc.  30 19:53 2016.12.30.185259.sql
-rw-r--r-- 1 root root 10611989 déc.  31 01:00 2016.12.31.000001.sql
-rw-r--r-- 1 root root 10611989 déc.  31 07:00 2016.12.31.060001.sql
-rw-r--r-- 1 root root 10611989 déc.  31 13:00 2016.12.31.120001.sql
-rw-r--r-- 1 root root 10616102 déc.  31 19:00 2016.12.31.180001.sql
-rw-r--r-- 1 root root 10616102 janv.  1 01:00 2017.01.01.000001.sql
-rw-r--r-- 1 root root 10616390 janv.  1 07:00 2017.01.01.060001.sql
-rw-r--r-- 1 root root 10616678 janv.  1 13:00 2017.01.01.120001.sql
-rw-r--r-- 1 root root 10616966 janv.  1 19:00 2017.01.01.180001.sql
-rw-r--r-- 1 root root 10616966 janv.  2 01:00 2017.01.02.000001.sql
-rw-r--r-- 1 root root 10617398 janv.  2 07:00 2017.01.02.060001.sql
-rw-r--r-- 1 root root 10617398 janv.  2 13:00 2017.01.02.120001.sql
-rw-r--r-- 1 root root 10617686 janv.  2 19:00 2017.01.02.180001.sql
-rw-r--r-- 1 root root 10617686 janv.  3 01:00 2017.01.03.000001.sql
-rw-r--r-- 1 root root 10617686 janv.  3 07:00 2017.01.03.060001.sql
-rw-r--r-- 1 root root 10617686 janv.  3 13:00 2017.01.03.120001.sql
-rw-r--r-- 1 root root 10617686 janv.  3 19:00 2017.01.03.180001.sql
-rw-r--r-- 1 root root 10777895 janv.  4 01:00 2017.01.04.000001.sql
-rw-r--r-- 1 root root 10782359 janv.  4 07:00 2017.01.04.060001.sql
-rw-r--r-- 1 root root 10783257 janv.  4 13:00 2017.01.04.120001.sql
-rw-r--r-- 1 root root 10783851 janv.  4 19:00 2017.01.04.180001.sql
-rw-r--r-- 1 root root 10784589 janv.  5 01:00 2017.01.05.000001.sql
-rw-r--r-- 1 root root 10784877 janv.  5 07:00 2017.01.05.060001.sql
-rw-r--r-- 1 root root 10784877 janv.  5 13:00 2017.01.05.120001.sql
-rw-r--r-- 1 root root 10808231 janv.  5 19:00 2017.01.05.180001.sql
-rw-r--r-- 1 root root 10954345 janv.  6 01:00 2017.01.06.000001.sql
-rw-r--r-- 1 root root 10954345 janv.  6 07:00 2017.01.06.060001.sql
-rw-r--r-- 1 root root 10956989 janv.  6 13:00 2017.01.06.120001.sql
root@pica01:/DATA/docker#
```

utc
SORBONNE UNIVERSITES

Monitoring

utc
SORBONNE UNIVERSITES

Monitoring

Utilisation d'outils open source développés par Elastic dans des conteneurs

Beats est un agent à installer sur les machines clients et qui pousse les métriques dans l'Elasticsearch

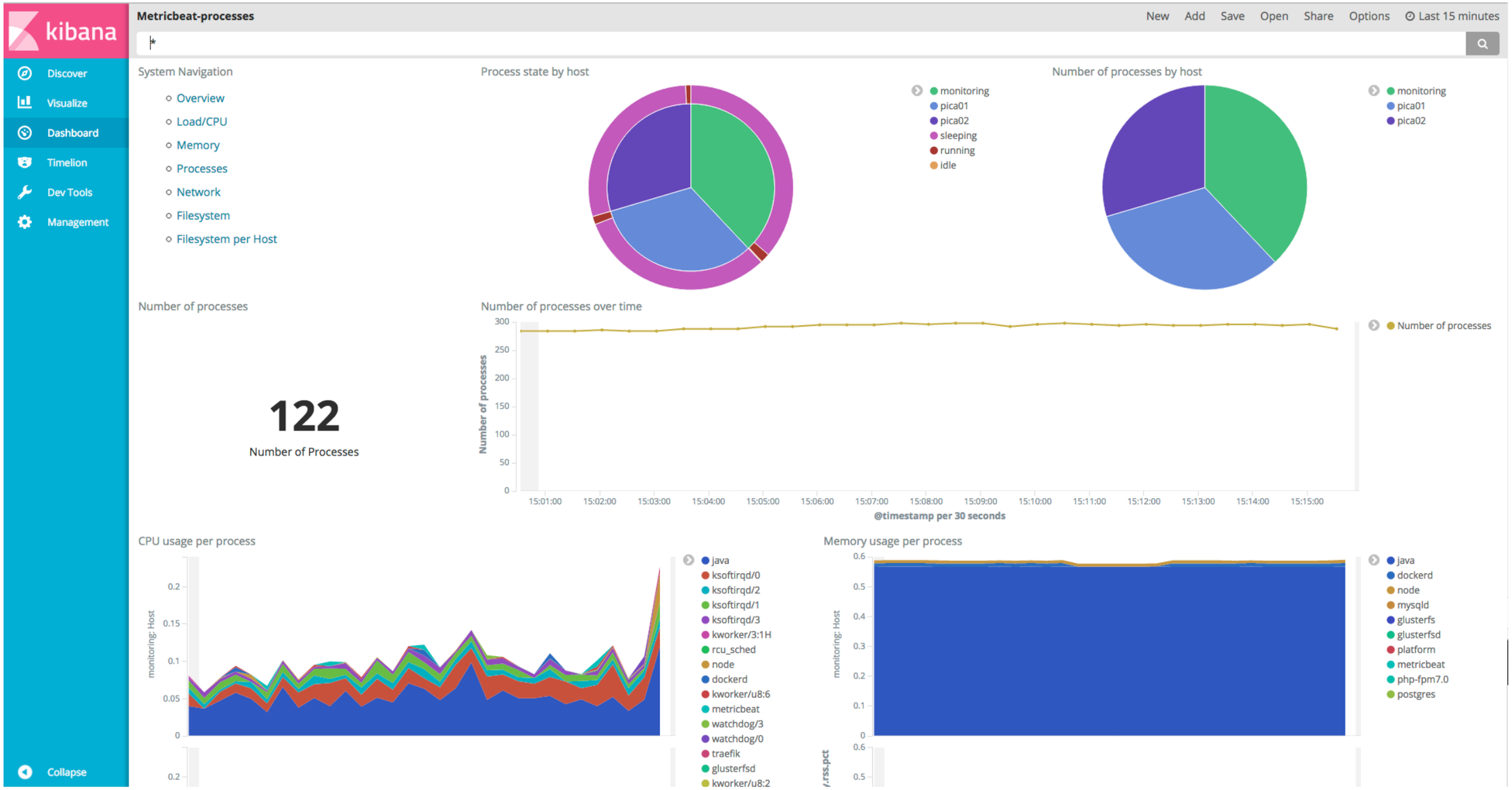
Elasticsearch sert de moteur de recherche et stocke les données

Kibana permet la visualisation graphique de ces données sous forme de dashboard

Avantages : Déploiement simple et rapide, possibilité d'avoir les métriques dans conteneurs dans le futur

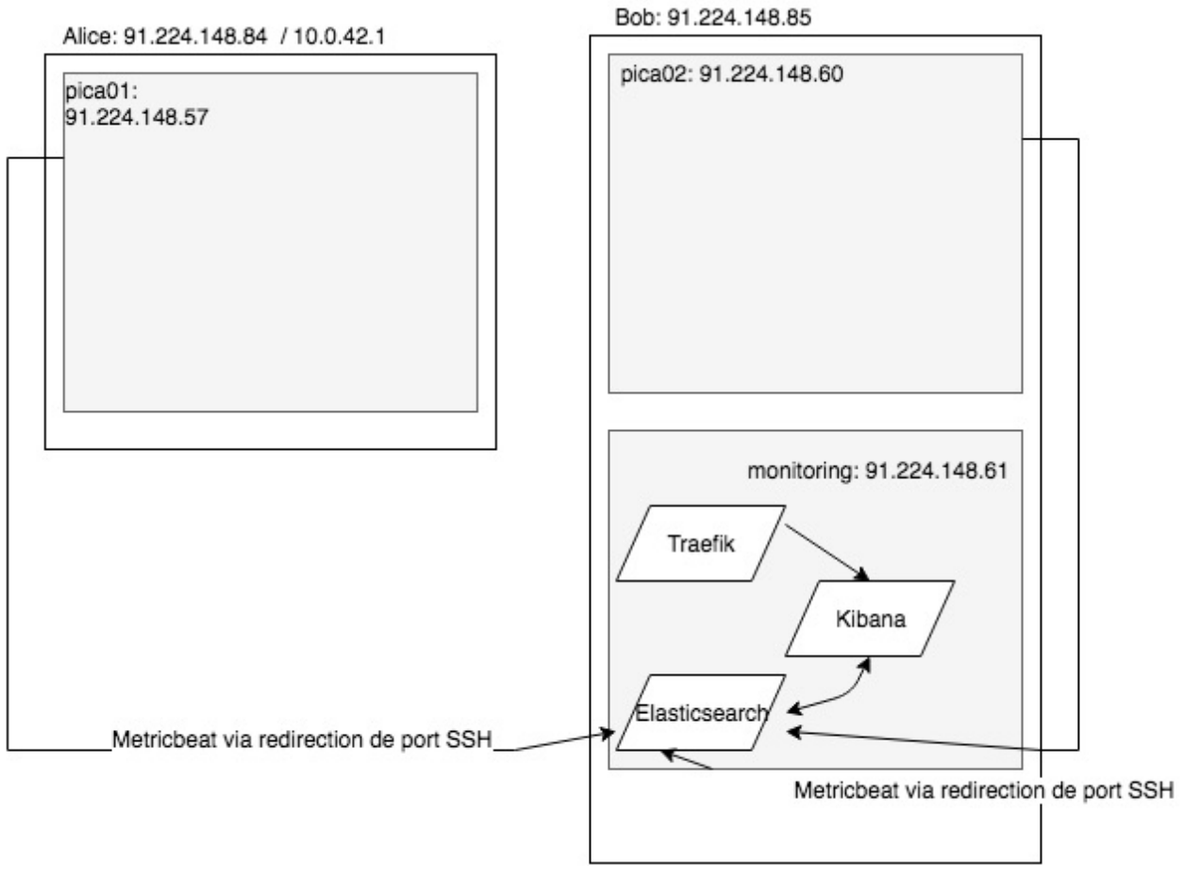
Inconvénients : Pas d'alerting possible, consommation de ressources qui peut être importante

Monitoring



utc
SORBONNE UNIVERSITES

Monitoring



Redirection de port via SSH sur l'Elasticsearch via autossh

Ne pas exposer l'Elasticsearch au public

Pica01, Pica02, Monitoring qui sont suivis pour le moment

Possibilité de migrer le monitoring sur une IP privée avec du NAT

Avez-vous des questions

utc
SORBONNE UNIVERSITES