

# ***Docker / Ansible : convergence avec Ansible Container***

Retour d'expérience sur la dockérisation d'une  
application naguère déployée avec Ansible

*Auteurs :*

*Ernest CHIARELLO, IR*

*Alexandre MAINDRON, IE-CDD*



MAISON DES SCIENCES DE L'HOMME  
ET DE L'ENVIRONNEMENT

CLAUDE NICOLAS LEDOUX

USR 3124

ThéMA

Théoriser & Modéliser pour Aménager  
UMR 6049 CNRS - Université Bourgogne Franche-Comté

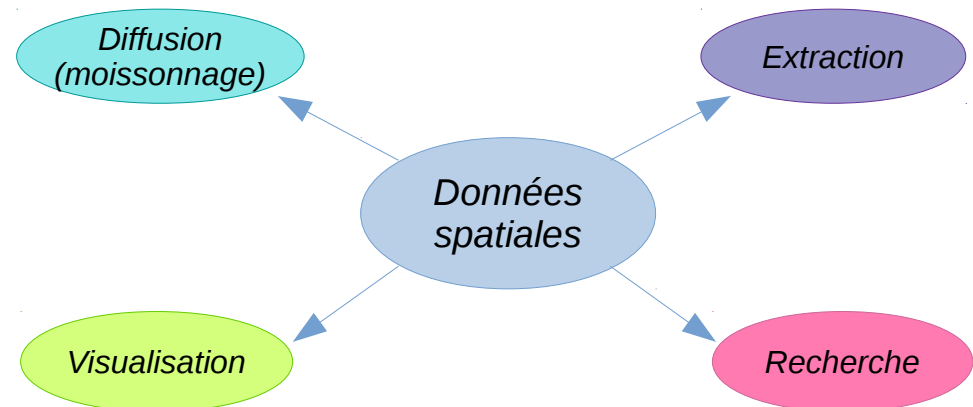
## Donnée spatiale :

- toute donnée pouvant être localisée : routes, bâtiments, relief, image satellitaire, etc.
- utilisée dans des disciplines de recherche variées
- nombreux chercheurs producteurs de données spatiales

## Infrastructure de données spatiales (IDS) :

- ensemble de services informatiques pour la gestion et la diffusion des données cartographiques
- regroupement des acteurs (producteurs, utilisateurs) autour du projet
- peut être définie à l'échelle régionale comme supranationale
- souvent basée sur la technologie web

## Services classiques d'une IDS :

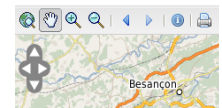


## Infrastructure de données spatiales (IDS) :

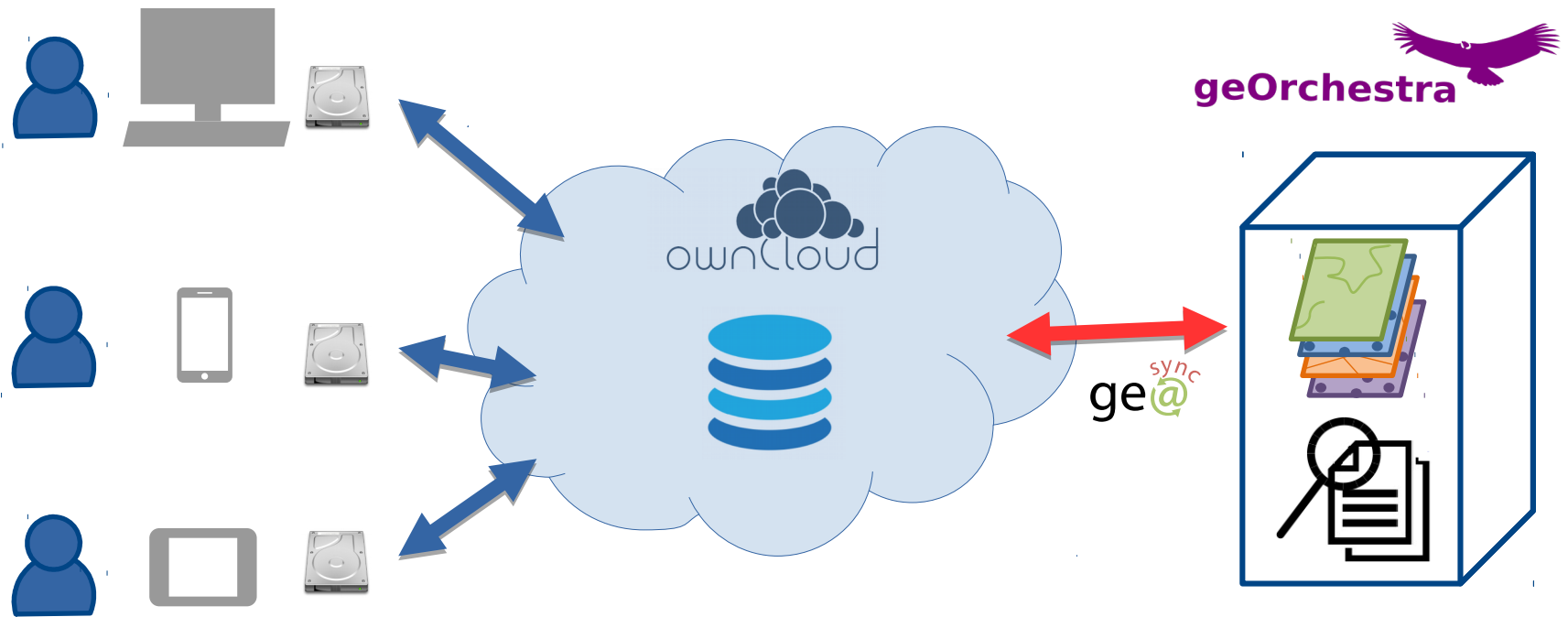
- gestion de l'information géographique
- assemblage d'applications sous Tomcat/Jetty
- respect de conditions d'interopérabilité



| Brique fonctionnelle    | Nom          | Fonction   |
|-------------------------|--------------|--|
| Serveur de cartes       | GeoServer    | Diffusion de données vecteur et raster de manière standardisée |
| Géocatalogue            | GeoNetwork   | Stockage des métadonnées et gestion de leur cycle de vie       |
| Visualiseur             | MapfishApp   | Affichage, édition et requêtage des données spatiales          |
| Extracteur              | ExtractorApp | Extraction et récupération des données spatiales               |
| Gestionnaires des rôles | LdapAdmin    | Gestion des utilisateurs et des groupes                        |



Les données sont déposées/partagées dans Owncloud, puis publiées sur l'IDS geOrchestra via ge@sync

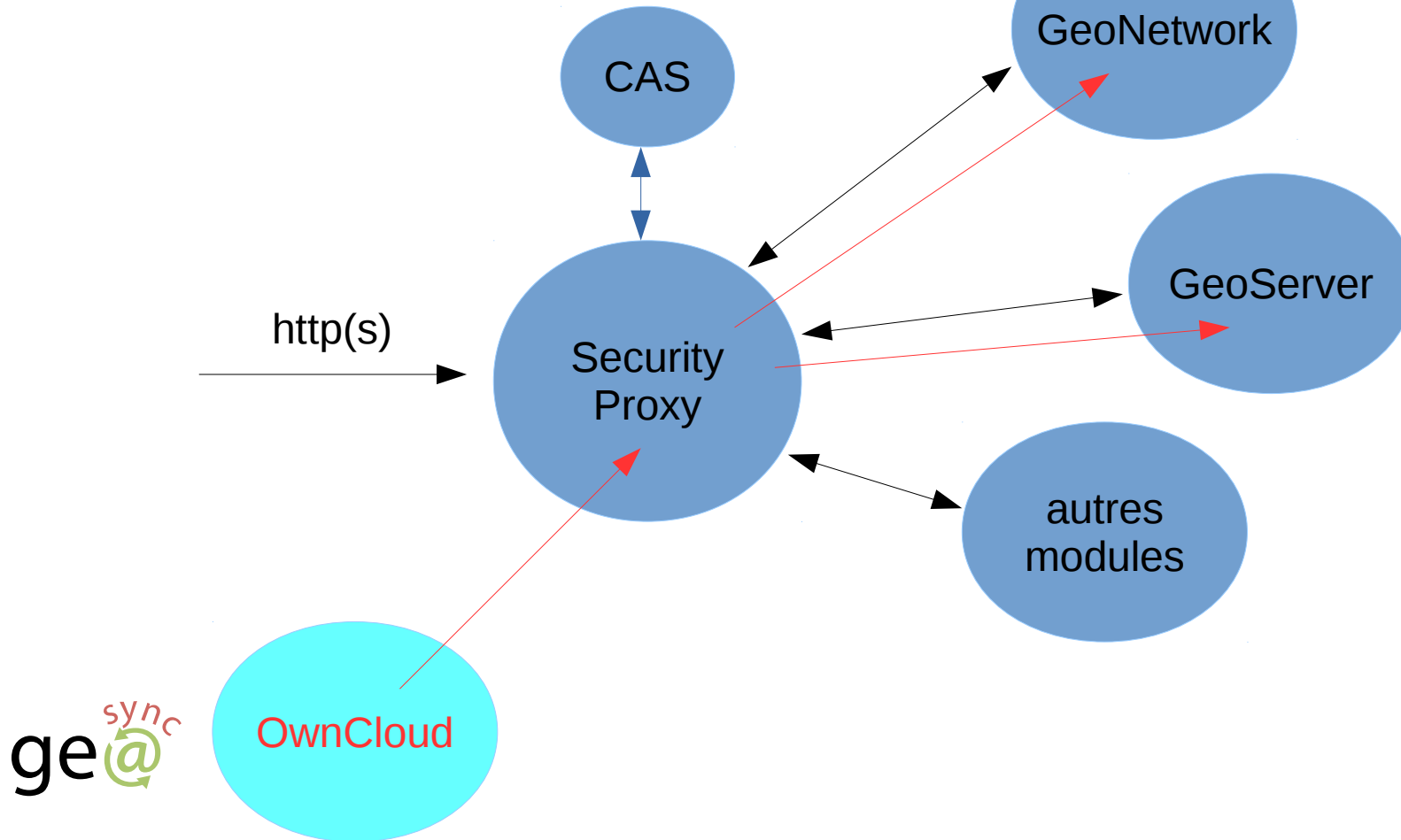


*Utilisateurs*

*Nuage*

*Infrastructure de données spatiales*

Rôle de **ge@sync**  
dans geOrchestra





ANSIBLE est un outil open-source de gestion de configuration écrit en Python

Un système déclaratif : syntaxe YAML facilement lisible

Génération dynamique de fichiers de configuration

Le langage de template par défaut est Jinja2

Quasiment rien à installer :

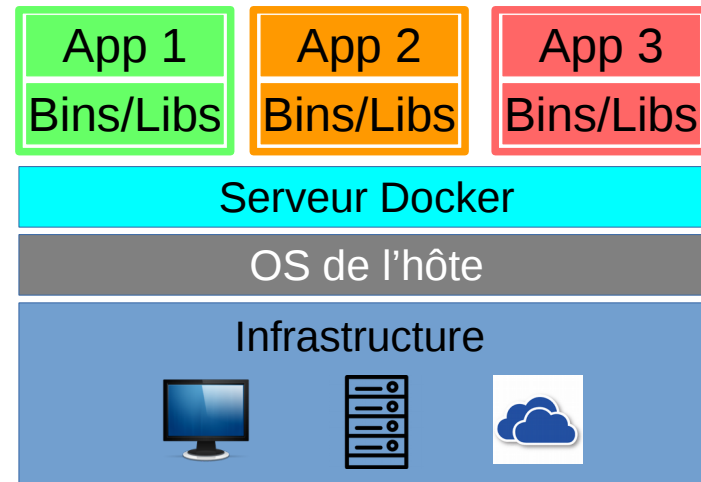
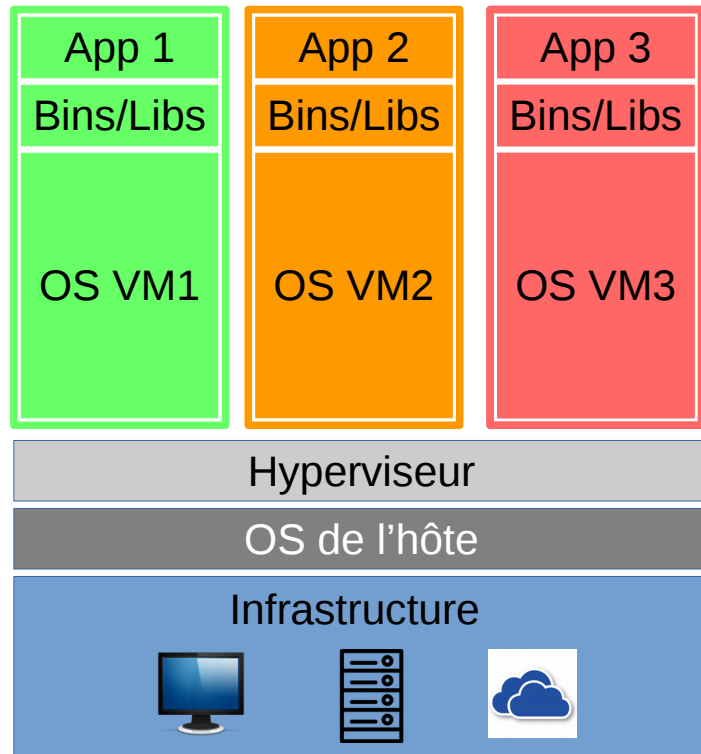
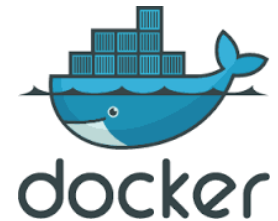
- Ansible sur votre machine hôte
- accès root via SSH sur vos serveurs cibles

Ansible sert à déployer votre infrastructure

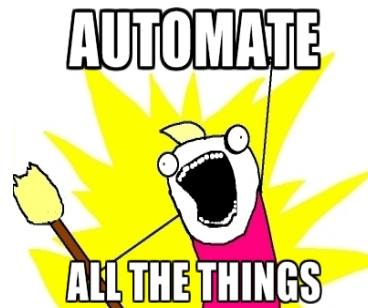
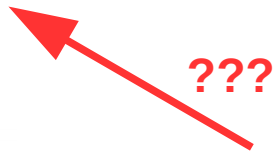
Ansible permet de vérifier : - que les services sont actifs

- que les fichiers de configuration sont bien à jour.

Meilleure utilisation des ressources matérielles  
Déploiement simplifié des applications  
Un standard qui change le paysage du DevOps



Objectif de Docker : partager le système d'exploitation



Ansible est efficace pour configurer des VM.

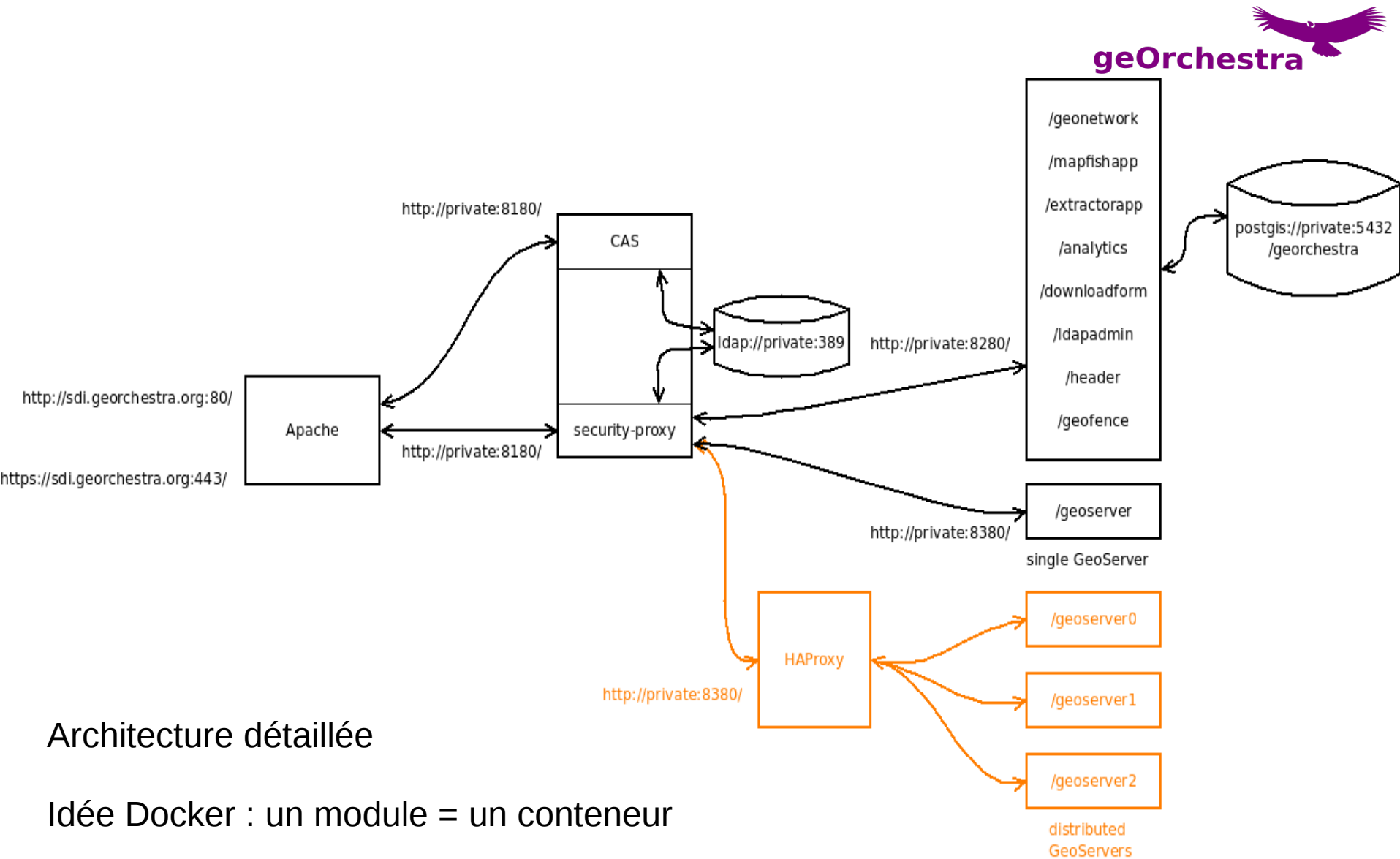
La philosophie Docker impose une approche différente.

Nos partenaires livrent aujourd'hui des conteneurs Docker.

Comment les personnaliser pour les adapter à nos besoins ?

- comment dockériser une application ?
- est-il possible de réutiliser nos playbooks Ansible ?







Explore Help

Search

Log In

<https://hub.docker.com/>

## Build, Ship, & Run Any App, Anywhere

Dev-test pipeline automation, 100,000+ free apps, public and private registries

### New to Docker?

Create your free Docker ID to get started.

Choose a Docker Hub ID

Enter your email address

Choose a password

Tous les conteneurs dont on peut avoir besoin :

GeoNetwork

GeoServer

PostgreSQL

Jetty

Apache

SSH

...

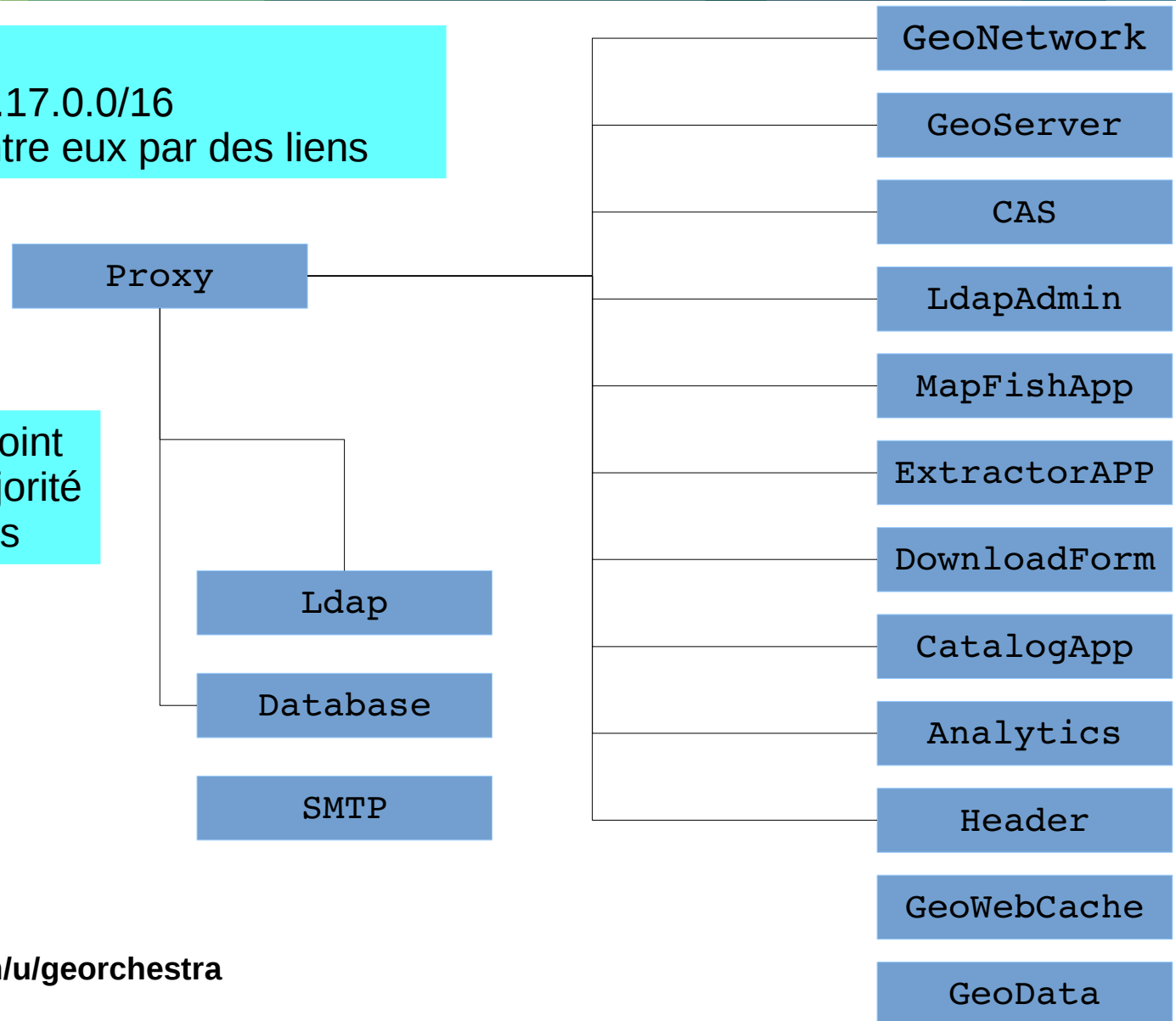
Basés sur :

- des distributions minimales
- avec les outils de base

16 conteneurs

- sur le réseau 172.17.0.0/16
- communiquant entre eux par des liens

Le proxy est le point d'entrée de la majorité des conteneurs



L'orchestration des conteneurs est gérée dans un fichier nommé docker-compose.yml

```
$ docker-compose up
```

Télécharge les images, lance les conteneurs, exécute l'orchestration.

...

```
$ docker ps
```

|                                   |                        |                            |
|-----------------------------------|------------------------|----------------------------|
| georchestra/security-proxy:latest | 0.0.0.0:8080->8080/tcp | georchestra_proxy_1        |
| georchestra/ldapadmin:latest      | 8080/tcp               | georchestra_ldapadmin_1    |
| georchestra/cas:latest            | 8080/tcp               | georchestra_cas_1          |
| georchestra/geonetwork:3-latest   | 8080/tcp               | georchestra_geonetwork_1   |
| georchestra/geoserver:latest      | 8080/tcp               | georchestra_geoserver_1    |
| georchestra/extractorapp:latest   | 8080/tcp               | georchestra_extractorapp_1 |
| georchestra/analytics:latest      | 8080/tcp               | georchestra_analytics_1    |
| georchestra/ssh_data              | 0.0.0.0:2222->22/tcp   | georchestra_geodata_1      |
| georchestra/mapfishapp:latest     | 8080/tcp               | georchestra_mapfishapp_1   |
| georchestra/catalogapp:latest     | 8080/tcp               | georchestra_catalogapp_1   |
| camptocamp/smtp-sink              | 25/tcp                 | georchestra_smtp_1         |
| georchestra/ldap                  | 389/tcp                | georchestra_ldap_1         |
| georchestra/downloadform:latest   | 8080/tcp               | georchestra_downloadform_1 |
| georchestra/database              | 5432/tcp               | georchestra_database_1     |
| georchestra/header:latest         | 8080/tcp               | georchestra_header_1       |
| georchestra/geowebcache:latest    | 8080/tcp               | georchestra_geowebcache_1  |

docker-compose.yml décrit les liens entre les conteneurs

Extrait concernant le conteneur proxy :

```
proxy:
  image: georchestra/security-proxy:latest
  ports:
    - "8080:8080"
  volumes:
    - /etc/georchestra:/etc/georchestra
  links:
    - database
    - cas
    - extractorapp
    - geonetwork
    - geoserver
    - catalogapp
    - ldap
    - mapfishapp
    - ldapadmin
    - header
    - analytics
    - downloadform
  environment:
    - XMS=512M
    - XMX=1G
```

Nom de l'image  
téléchargée depuis  
docker-hub

Le port 8080 de l'hôte  
correspond au port  
8080 du conteneur

Le répertoire /etc/georchestra  
est partagé entre l'hôte et le  
conteneur

Le conteneur proxy est lié à  
presque tous les autres  
conteneurs

Variables d'environnement  
pour Jetty

La base de données est liée à 8 conteneurs.  
Exemple : GeoData est utilisé par Database

`docker-compose.yml`

```
geodata:  
  image: georchestra/ssh_data  
  ports:  
    - "5432:5432"  
  links:  
    - database
```

Proxy

Ldap

Database

172.17.0.2/16  
5432/tcp open postgresql

SMTP

GeoNetwork

GeoServer

CAS

LdapAdmin

MapFishApp

ExtractorAPP

DownloadForm

CatalogApp

Analytics

Header

GeoWebCache

GeoData

## docker-compose.yml

```
database:
  image: georchestra/database
  environment:
    - POSTGRES_USER=georchestra
    - POSTGRES_PASSWORD=georchestra
  volumes:
    - postgresql_data:/var/lib/postgresql/data
```

Le volume  
postgresql\_data  
garantit la pérennité  
des données.

Il est accessible pour  
l'hôte et le conteneur.

```
# ls /var/lib/docker/volumes/postgresql_data/_data/
base          pg_hba.conf    pg_notify      pg_stat        pg_twophase    postgresql.conf
global        pg_ident.conf  pg_replslot    pg_stat_tmp    PG_VERSION     postmaster.opts
pg_clog       pg_logical     pg_serial      pg_subtrans    pg_xlog        postmaster.pid
pg_dynshmem   pg_multixact   pg_snapshots   pg_tblspc      postgresql.auto.conf
```

```
$ docker exec -it georchestra_database_1 bash
root@a1d8e60f250c:/# ls /var/lib/postgresql/data/
```

```
base          pg_hba.conf    pg_notify      pg_stat        pg_twophase    postgresql.conf
global        pg_ident.conf  pg_replslot    pg_stat_tmp    PG_VERSION     postmaster.opts
pg_clog       pg_logical     pg_serial      pg_subtrans    pg_xlog        postmaster.pid
pg_dynshmem   pg_multixact   pg_snapshots   pg_tblspc      postgresql.auto.conf
```

## Dockerfile permettant de créer georchestra/ldap

```
FROM dinkel/openldap

RUN apt-get update && \
    apt-get install -y ldap-utils

USER openldap
ADD groupofmembers.ldif /etc/ldap.dist/modules/
ADD docker/memberof.ldif /tmp/
ADD georchestra.ldif /tmp/

USER root
RUN mkdir /docker-entrypoint.d
ADD docker-entrypoint.d/* /docker-entrypoint.d/
ADD docker-entrypoint.sh /
RUN chmod +x /docker-entrypoint.d/*

ENTRYPOINT [ "/docker-entrypoint.sh" ]

CMD [ "slapd", "-d", "32768", "-u", "openldap",
      "-g", "openldap" ]
```

### Dockerfile

Installation de  
paquets spécifiques

Ajout de fichiers  
pour alimenter  
l'annuaire LDAP

Préparation du  
point d'entrée

Le point d'entrée

La commande au  
démarrage



## Dockerfile

```
FROM georchestra/ldap

ENV SLAPD_DOMAIN georchestra.univ-fcomte.fr

RUN apt-get install -y python-ldap cron sasl2-bin

ADD 01-ldap_sasl /docker-entrypoint.d/
ADD *.ldif /tmp/
ADD saslauthd.conf /etc/
ADD slapd.conf /tmp/
ADD root_crontab /var/spool/cron/crontabs/root

ADD start.sh /root/

ENTRYPOINT [ "/docker-entrypoint.sh" ]

CMD [ "/root/start.sh" ]
```

Objectif : utiliser les mots de passe de l'AD de Théma.

Ajout de :

- saslauthd
- cron

## start.sh

```
/etc/init.d/cron start
/etc/init.d/saslauthd start
slapd -d 32768 -u openldap -g openldap
```

3 processus au démarrage du conteneur

```
FROM postgres:9.4

ENV DEBIAN_FRONTEND noninteractive

RUN apt-get update && \
    apt-get install -y postgresql-9.4-postgis-2.1

ADD *.sql          /docker-entrypoint-initdb.d/
ADD license.txt    /docker-entrypoint-initdb.d/
ADD logo.png       /docker-entrypoint-initdb.d/

CMD ["postgres"]
```

Dockerfile

Dockerfile permettant de créer le conteneur georchestra/database

```
$ ls database
00-create-user.sql          05-ldapadmin-data.sql
01-create-extension.sql    06-ogc-server-statistics.sql
02-mapfishapp.sql          07-atlas.sql
03-downloadform.sql        07-ogc-server-statistics-data.sql
04-ldapadmin.sql           08-geofence.sql
Dockerfile                  fix-owner.sql
logo.png                    license.txt
```

Se connecter sur le hub de docker pour en savoir plus sur l'image postgres:9.4

docker-library / postgres

Watch 41

41

Star 432

432

Fork 223

223

Code

Issues 53

Pull requests 10

Projects 0

Pulse

Graphs

Branch: master

postgres / 9.4 /

Create new file

Upload files

Find file

History

tianon committed on GitHub Merge pull request #226 from lockie/master

Latest commit d2b3854 6 days ago

..

|                      |   |             |
|----------------------|---|-------------|
| alpine               | Merge pull request #226 from lockie/master            | 6 days ago  |
| Dockerfile           | Update to 9.4.10-1.pgdg80+1                           | a month ago |
| docker-entrypoint.sh | Add "file_env" support, especially for Docker secrets | 7 days ago  |

## Le point d'entrée : docker-entrypoint.sh

```
for f in /docker-entrypoint-initdb.d/*; do
  case "$f" in
    *.sh) echo "$0: running $f"; . "$f" ;;
    *.sql) echo "$0: running $f"; "${psql[@]}" < "$f"; echo ;;
    *) echo "$0: ignoring $f" ;;
  esac
done
```

```
FROM georchestra/database
```

Dockerfile

Image du projet

```
ENV DEBIAN_FRONTEND noninteractive
```

```
ENV TERM linux
```

```
RUN echo Europe/Paris | tee /etc/timezone
```

```
RUN dpkg-reconfigure --frontend noninteractive tzdata
```

Timezone à Paris

```
ADD 09-geosync.sql /docker-entrypoint-initdb.d/
```

```
ADD pg_hba.conf /docker-entrypoint-initdb.d/
```

```
ADD setting_pg.sh /docker-entrypoint-initdb.d/
```

Ajout de 3 fichiers

```
CMD ["postgres"]
```

09-geosync.sql

```
CREATE USER geosync WITH PASSWORD 'secret';
```

```
CREATE DATABASE geosync_data_open with owner geosync;
```

```
CREATE EXTENSION postgis;
```

setting\_pg.sh

```
echo "setting pg_hba.conf"
```

```
cp /docker-entrypoint-initdb.d/pg_hba.conf /var/lib/postgresql/data/.
```

Ansible Container permet de créer des conteneurs Docker, en utilisant les technologies Ansible.

```
$ ansible-container init
```

```
$ tree -L 2
```

```
.
├── 09-geosync.sql.j2
├── ansible
│   ├── ansible.cfg
│   ├── container.yml
│   ├── main.yml
│   ├── requirements.txt
│   └── requirements.yml
├── pg_hba.conf
└── setting.sh
```

container.yml

```
services:

  database:
    image: georchestra/database
    entrypoint: /docker-entrypoint.sh
    command: postgres
```

main.yml

```
- hosts: database
  gather_facts: no

  vars:
    geosync: {
      db: {
        name: geosync_data,
        user: geosync,
        pass: secret,
      }
    }
}
```

.../...

tasks:

main.yml

```
- name: setting timezone
  raw: echo Europe/Paris | tee /etc/timezone
- name: setting timezone
  raw: dpkg-reconfigure --frontend noninteractive tzdata
- name: copy stetting_pg.sh
  copy: src=../setting_pg.sh
      dest=/docker-entrypoint-initdb.d/setting_pg.sh
      owner=root group=root mode=0775
- name: copy pg_hba.conf
  copy: src=../pg_hba.conf
      dest=/docker-entrypoint-initdb.d/pg_hba.conf
      owner=root group=root mode=0775
- name: template 09-geosync.sql
  template: src=../09-geosync.sql.j2
            dest=/docker-entrypoint-initdb.d/09-geosync.sql
            owner=root group=root mode=0664
```

```
CREATE USER geosync WITH PASSWORD '{{geosync.db.pass}}';
```

```
CREATE DATABASE {{geosync.db.name}} with owner {{geosync.db.user}};
\connect {{geosync.db.name}}
CREATE EXTENSION postgis;
```

09-geosync.sql.j2

4/16 conteneurs ont été adaptés, soit avec Docker, soit avec Ansible Container.

|                              |                    |                          |
|------------------------------|--------------------|--------------------------|
| georchestra/geoserver:latest | a été remplacé par | <b>geosync_geoserver</b> |
| georchestra/ssh_data         | a été remplacé par | <b>geosync_ssh_data</b>  |
| georchestra/ldap             | a été remplacé par | <b>geosync_ldap</b>      |
| georchestra/database         | a été remplacé par | <b>geosync_database</b>  |

Il suffit de modifier docker-compose.yml pour obtenir notre geOrchestra / ge@sync

```
$ docker PS
```

```
georchestra/security-proxy:latest    0.0.0.0:8080->8080/tcp    georchestra_proxy_1
georchestra/ldapadmin:latest         8080/tcp                  georchestra_ldapadmin_1
georchestra/extractorapp:latest      8080/tcp                  georchestra_extractorapp_1
geosync_geoserver                  8080/tcp                 georchestra_geoserver_1
georchestra/geonetwork:3-latest     8080/tcp                  georchestra_geonetwork_1
georchestra/mapfishapp:latest       8080/tcp                  georchestra_mapfishapp_1
georchestra/analytics:latest        8080/tcp                  georchestra_analytics_1
geosync_ssh_data                  0.0.0.0:2222->22/tcp    georchestra_geodata_1
georchestra/cas:latest               8080/tcp                  georchestra_cas_1
georchestra/downloadform:latest     8080/tcp                  georchestra_downloadform_1
geosync_ldap                      389/tcp                 georchestra_ldap_1
geosync_database                 0.0.0.0:5432->5432/tcp georchestra_database_1
georchestra/geowebcache:latest      8080/tcp                  georchestra_geowebcache_1
camptocamp/smtp-sink                 25/tcp                    georchestra_smtp_1
georchestra/catalogapp:latest       8080/tcp                  georchestra_catalogapp_1
georchestra/header:latest           8080/tcp                  georchestra_header_1
```

| UID   | PID | PPID | COMMAND                                     |
|-------|-----|------|---|
| jetty | 1   | 0    | java -Dgeorchestra.datadir=/etc/georchestra |
| root  | 858 | 0    | bash  |

geoserver

| UID  | PID  | PPID | CMD                       |
|------|------|------|---------------------------|
| root | 1    | 0    | /bin/sh -c /root/start.sh |
| root | 7    | 1    | /usr/sbin/sshd -D         |
| Root | 1060 | 1    | /usr/sbin/cron            |

ssh\_data

| UID      | PID | PPID | COMMAND  |
|----------|-----|------|--|
| root     | 1   | 0    | /bin/bash /root/start.sh                                     |
| root     | 31  | 1    | /usr/sbin/cron   |
| root     | 52  | 1    | /usr/sbin/saslauthd -a ldap -r -c -m /var/run/saslauthd -n 5 |
| openldap | 57  | 1    | slapd -d 32768 -u openldap -g openldap                       |

ldap

| UID      | PID | PPID | COMMAND                                      |
|----------|-----|------|--|
| postgres | 1   | 0    | postgres                                     |
| postgres | 18  | 1    | postgres: checkpointer process               |
| postgres | 19  | 1    | postgres: writer process                     |
| postgres | 20  | 1    | postgres: wal writer process                 |
| postgres | 21  | 1    | postgres: autovacuum launcher process        |
| postgres | 22  | 1    | postgres: stats collector process            |
| postgres | 31  | 1    | postgres: ldapadmin georchestra 172.17.0.14  |
| postgres | 65  | 1    | postgres: geonetwork georchestra 172.17.0.13 |

database



Docker est une technologie séduisante et efficace.

La courbe d'apprentissage est longue, mais le jeu en vaut la chandelle.

On pensait oublier Bash et consorts... mais le contraire s'impose.

Ansible Container est un concept intéressant, mais il n'enlève pas la complexité de Docker.

Il manquait un serveur proxy à notre geOrchestra : c'est Traefik.

Travail à venir : la répartition des volumes entre différents hôtes.  
=> stockage distribué avec GlusterFS

Exit à terme les architectures habituelles de virtualisation.

geOrchestra catalogue view extra services administration

My GeoNetwork catalogue Search + Contribute Admin console

Back to search

Haies de Franche-Comté en 2010. La couche a été créée au 1/5000. L'échelle d'utilisation recommandée est au 1/10000.

Updated: a few seconds ago

Cette couche représente l'ensemble des haies de Franche-Comté en 2010-2011 qui ont été vectorisées par photo-interprétation.

Download and links

enquete\_ar\_em\_pourcent

This dataset is published in the view service (WMS) available at <https://georchestra-dev.umh.fr/maps/arcgis/rest/info?f=application/javascript&layer=arcgis/rest/info?f=application/javascript&layer=arcgis/rest/info?f=application/javascript>

Default Polygon (LegendURL)  Open link

About this resource

Categories

Keywords

- France
- Franche-Comté
- Haies
- Agriculture
- Zones naturelles
- Paysages
- Écologiques
- Downloadable Data
- Environnement

Spatial extent

Layers

Legend

- BATI\_REMARQUABLE
- BATI\_INDUSTRIEL
- BATI\_INDIFFERENCIE

Base Layer

OpenStreetMap

Street, City admin

Fichiers

Tous les fichiers

Favoris

Partagés avec vous

Partagés avec d'autres

Partagés par lien

Fichiers supprimés

Paramètres

Nouveau

160428\_Cesblo

archives

arrond\_bfc

baiesAnthonySophie

clc2\_test

Sophie Mollard

# Merci de votre attention...

# Avez-vous des questions ?

Haies de Franche-Comté en 2010. La couche a été créée au 1/5000. L'échelle d'utilisation recommandée est au 1/10000.

test\_style\_shape\_shape\_avec\_style\_sld\_style7

haiesophie2\_haies\_sophie2

Hauts du Chazal

Planoise

Châteaufaraine

Avanne-

geOrchestra catalogue visualiser services

443.8 MB il y a 2 mois

Couches disponibles

- Bornes\_bornes\_etat
- OpenStreetMap
- GeoImage/OSM

Localités

Rechercher

Allez à :

MSHE Ledoux

BD ORTHO® Franche-Comté

La BD ORTHO® V1 est une collection de mosaïques numériques d'orthophotographies en couleur rectifiées en projection Lambert II étendu et couvrant chacune un département métropolitain. Le produit BD ORTHO® V1 est livrable au format TIFF (Tagged Image File Format) en dalles de 1 km x 1km, ou au format ECW (Enhanced Compressed Wavelet) en dalles de...

ORTHO-IMAGERIE, FRANCHE-COMTE, GEOGRAPHIE, IMAGERYBASEMAPSEARTHCOVER