ANDRY ANDRIATIANA

Portage d'une architecture SOA sous Docker

exemple du système d'information du réseau d'observation ReefTEMPS

- Coordinateur réseau: Bernard Pelletier

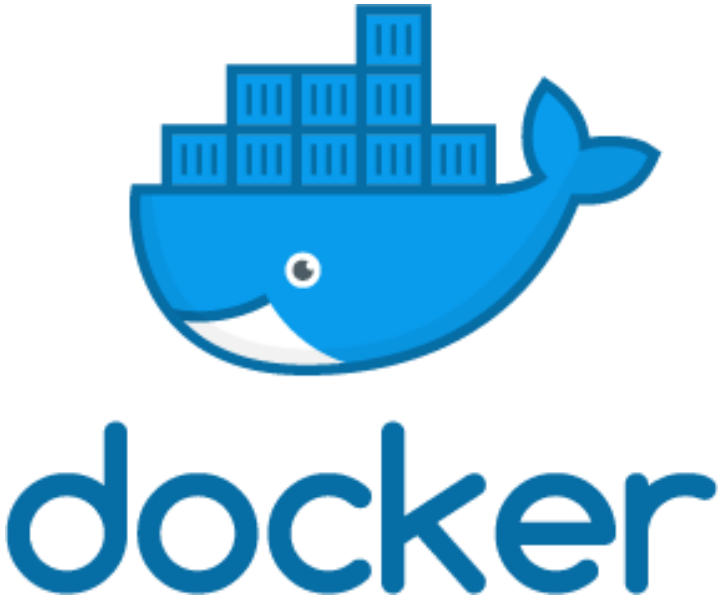
- Chef de projet SI: Régis Hocdé

- Responsable exploitation SI: Sylvie Fiat

- Responsable données: David Varillon

- Equipe projet : Jacques Grelet, Adrien Cheype, Guillaume Brissebrat, Andry Andriatiana & Collaborateurs prestataires

QU'EST CE QUE DOCKER



• Gestionnaire de conteneur virtuel

- Projet open source qui automatise et simplifie le déploiement d'applications dans des conteneurs virtuels.

• Proposé depuis mars 2013

- Développé en Californie par un ingénieur français, Solomon Hykes.

QU'EST CE QUE DOCKER

• Conteneur

- Environnement isolé qui repose directement sur le noyau de l'hôte.

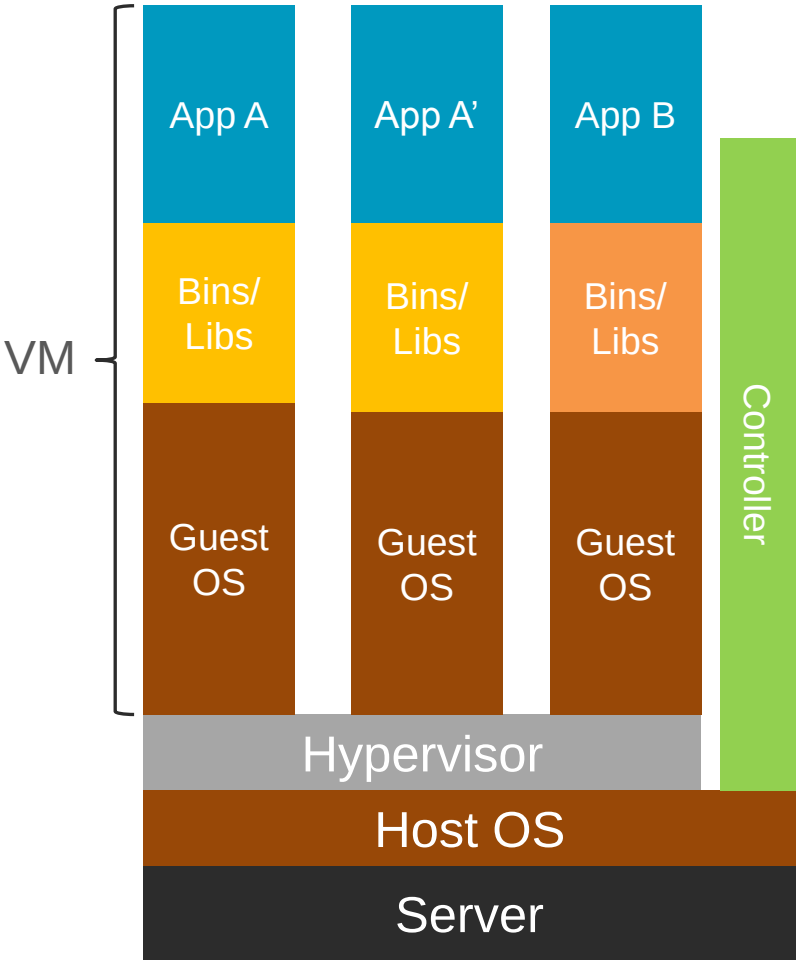
- Docker en gère le cycle de vie.

• Conteneurisation

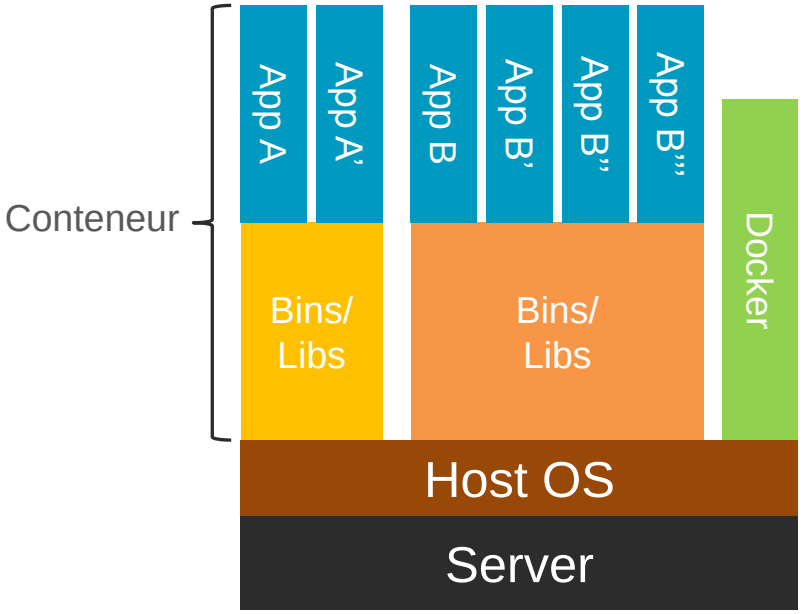
- A l'origine basée sur Linux container (surcouche de LXC).

- Couche d'abstraction Libcontainer permettant de gérer d'autres technologies d'isolation.

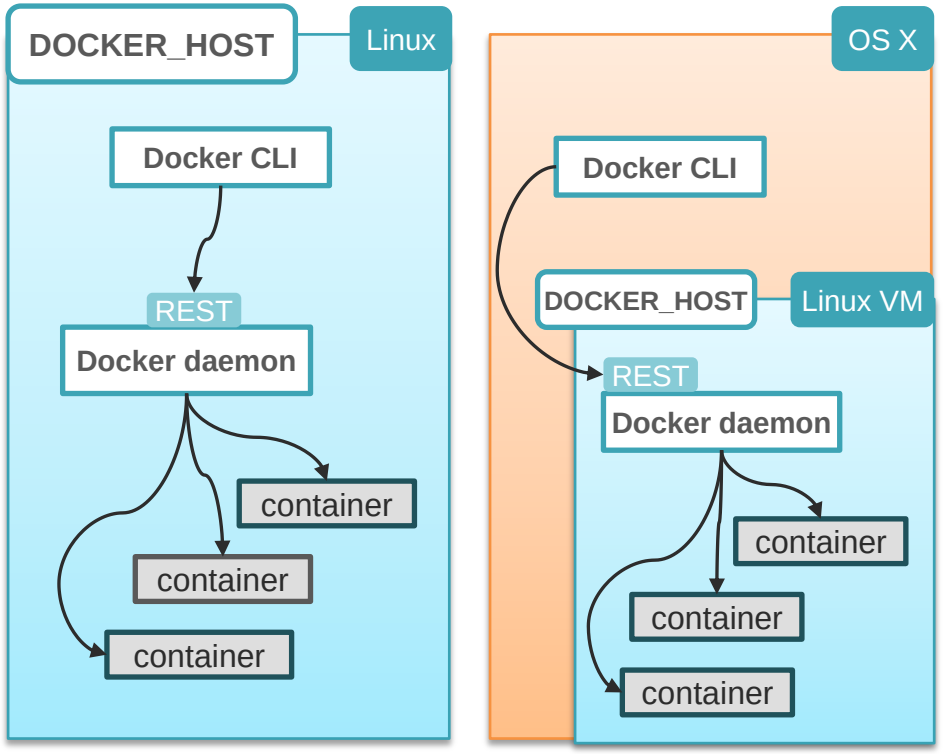
VM ET CONTENEUR



Les conteneurs sont isolés mais partagent leur OS avec l'hôte et, dans certains cas, leurs binaires et librairies.



QU'EST CE QUE DOCKER



• Docker engine

- Interrogeable via une api RESTFull.

- Un client « docker-cli » a été développé pour interagir directement via le terminal de commande.

• Kernel Linux

- Initialement utilisable seulement sous linux. Sur les autres OS via l'intermédiaire d'un émulateur linux.

- Depuis aout 2016, version stable pour tourner nativement sur Mac OS et Windows.

• QUOI ?

- Les conteneurs sont instanciés à partir des images.

- Peuvent être vu comme des templates de conteneur ou un conteneur statique (photographies d'environnement à un instant t).

- Obtention via dépôt docker hub (ou registry privée) ou build.

- Les images Docker fonctionnent grâce à de l'héritage d'autres images.

- L’image est immuable contrairement au conteneur.

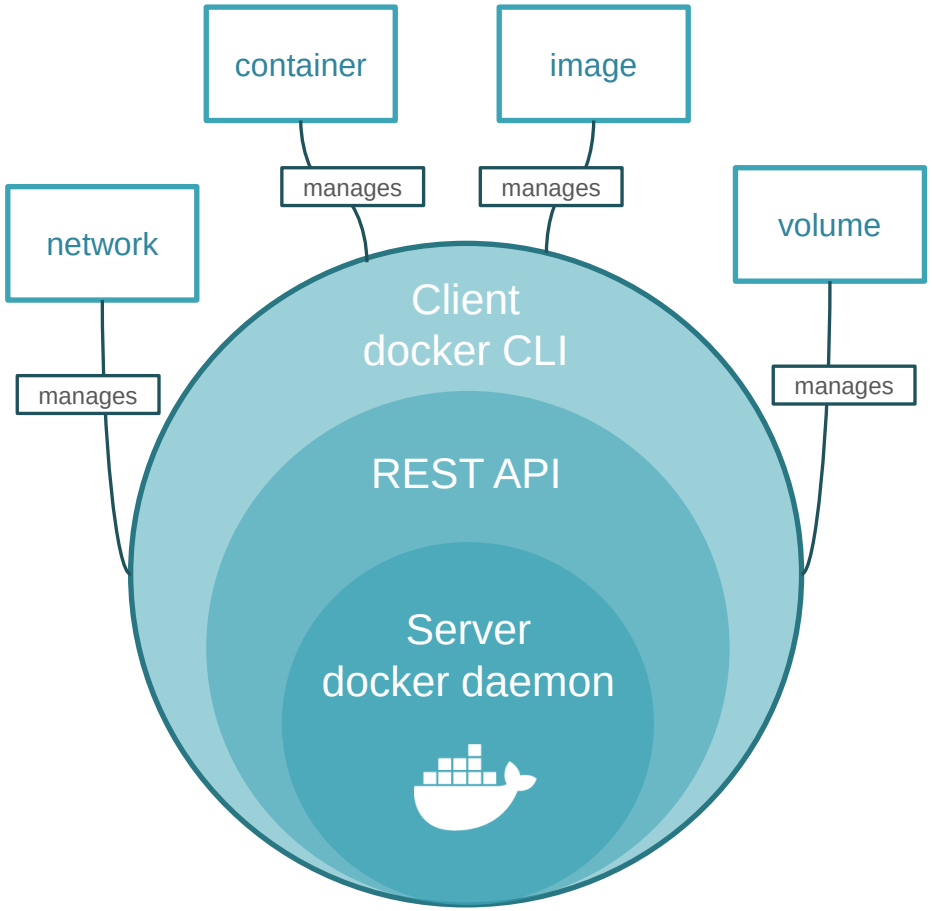
• QUOI ?

- Montage de répertoire/fichier de l’hôte vers le conteneur.

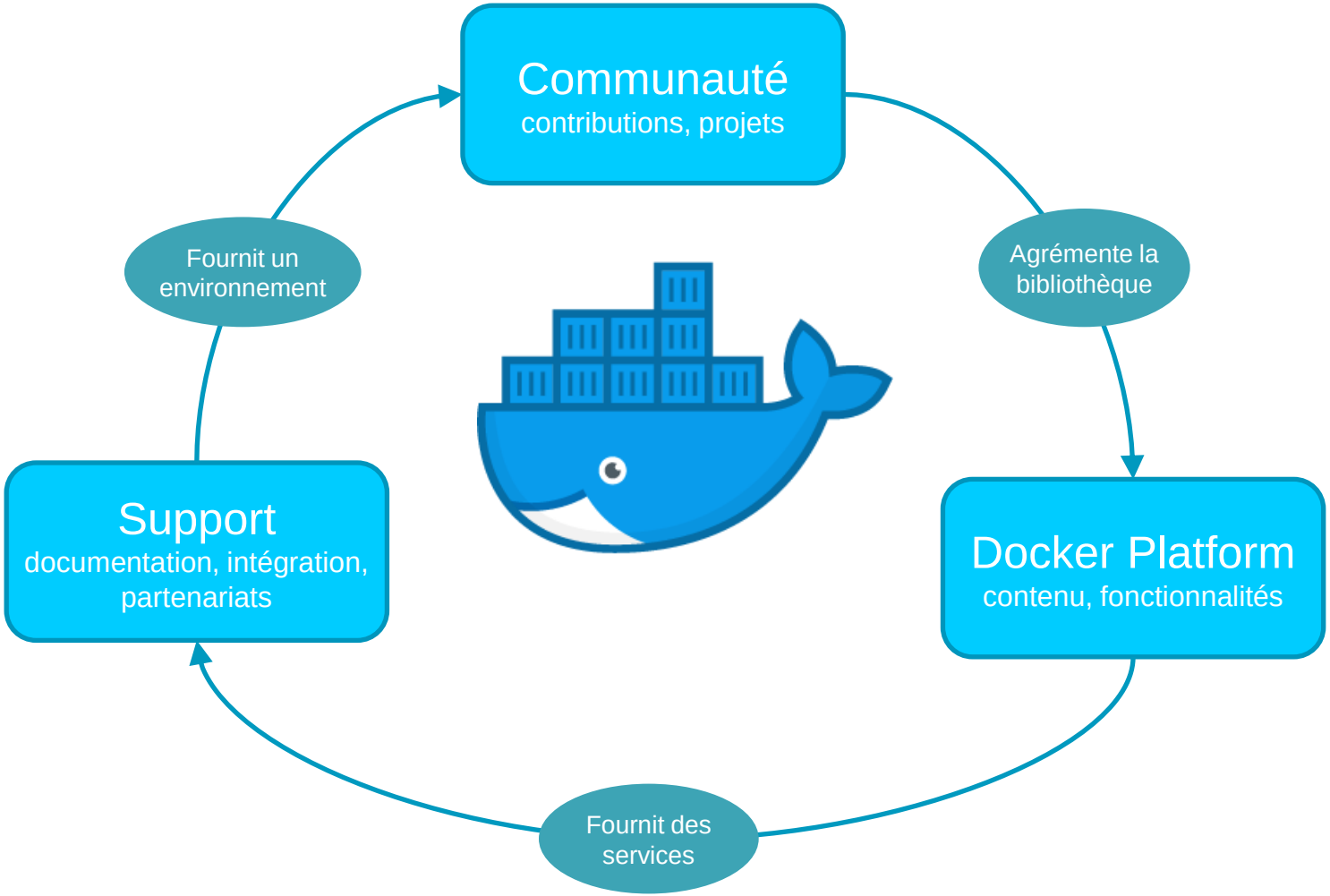
- Moyen de persister des données hors du conteneur et/ou de les partager entre les conteneurs.

- Moyen de fournir des ressources à l’initialisation d’un conteneur.

QU'EST CE QUE DOCKER



ECOSYSTEME


Communauté
contributions, projets
Agrémente la bibliothèque
Docker Platform
contenu, fonctionnalités
Fournit des services
Support
documentation, intégration, partenariats
Fournit un environnement

INSTALLATION

Platform	Docker CE x86_64	Docker CE ARM	Docker EE
Ubuntu	✓	✓	✓
Debian	✓	✓	
Red Hat Enterprise Linux			✓
CentOS	✓		✓
Fedora	✓		
Oracle Linux			✓
SUSE Linux Enterprise Server			✓
Microsoft Windows Server 2016			✓
Microsoft Windows 10	✓		
macOS	✓		
Microsoft Azure	✓		✓
Amazon Web Services	✓		✓

https://docs.docker.com/engine/installation

• Linux

- Packages et documentation disponibles pour différentes distributions.

• Windows et Mac OS

- Passer par l'installation de boot2docker ou docker Machine (configurer l'hôte pour Docker).

- Installer la version native.

• Docker-compose

- Outil de déploiement.

• Orchestration de conteneurs

- Déploiement multi-conteneur.
- Simple d'utilisation, décrit par le fichier docker-compose.yml.
- Définition des options permettant de lier les conteneurs, définir les volumes, les ports…
- Gestion des dépendances entre conteneurs.
- Architecture microservices.

Réseau ReefTEMPS

- Réseau de capteurs de température, pression et salinité dans le domaine côtier du Pacifique Sud, Ouest et Sud Ouest.

- Des instruments déployés pour certains depuis plus de 40 ans.

- 5 gestionnaires: IRD Noumea, USP Fidji, UNC, CPS-SOPAC, SO-Corail-CRIOBE.

- 14 pays : Nouvelle-Calédonie, Polynésie Française, Wallis & Futuna, Vanuatu, Miconésie, Iles Marshall, Papouasie Nouvelle-Guinnée, Tuvalu, Kiribati, Palau, Tokelau, Nauru, Samoa, Fidji.

- Le système d'information a été créé en 2011. La nouvelle version est disponible depuis juin 2017.

OBJECTIF DU SI

- Gestion et diffusion des données issues du réseau de capteurs.

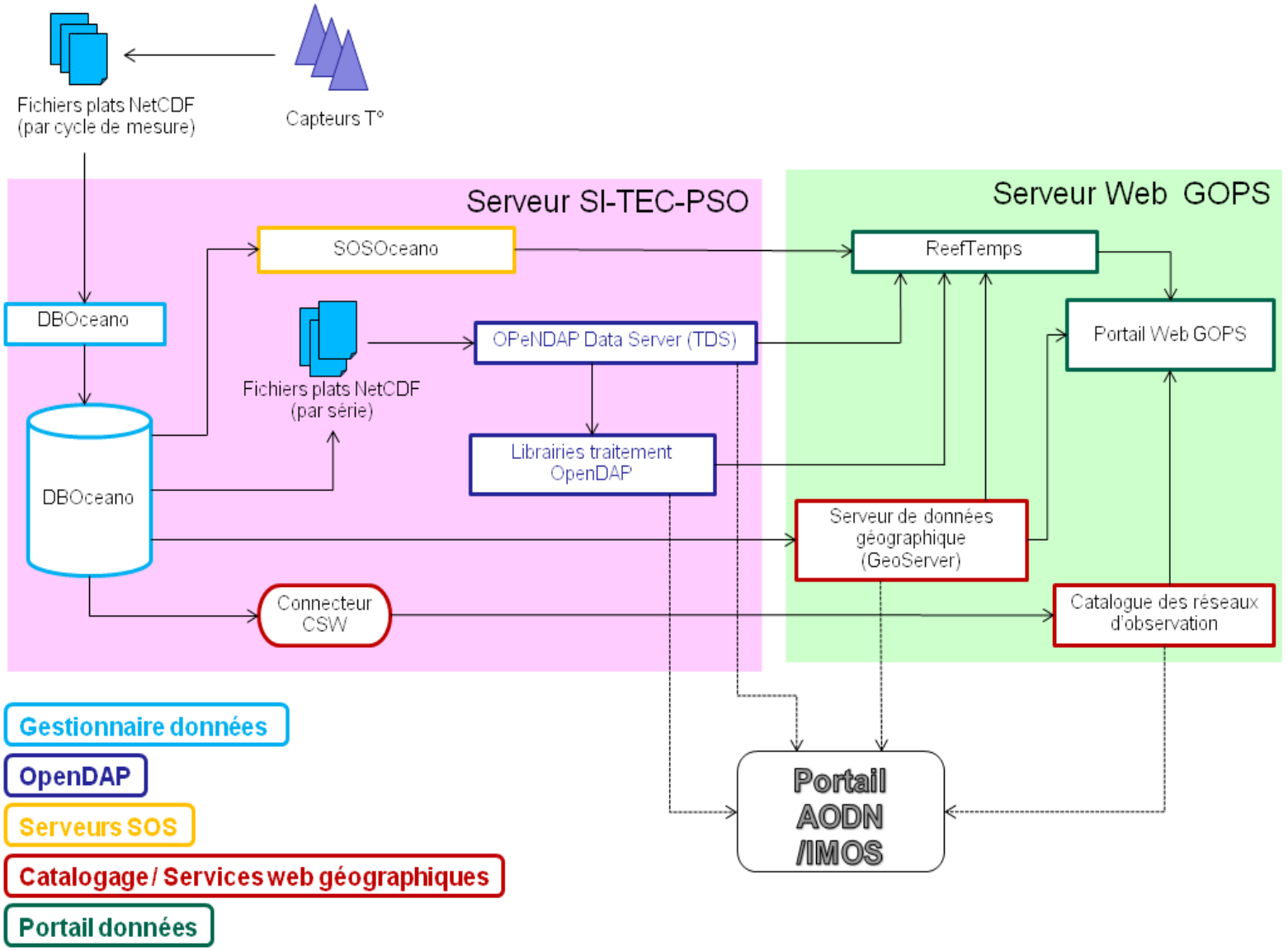
- Rendre les données accessibles à la communauté le plus rapidement possible, avec un libre accès.

- Assurer la pérennité des données dans une logique d'entrepôt ou de centre de données virtuel.

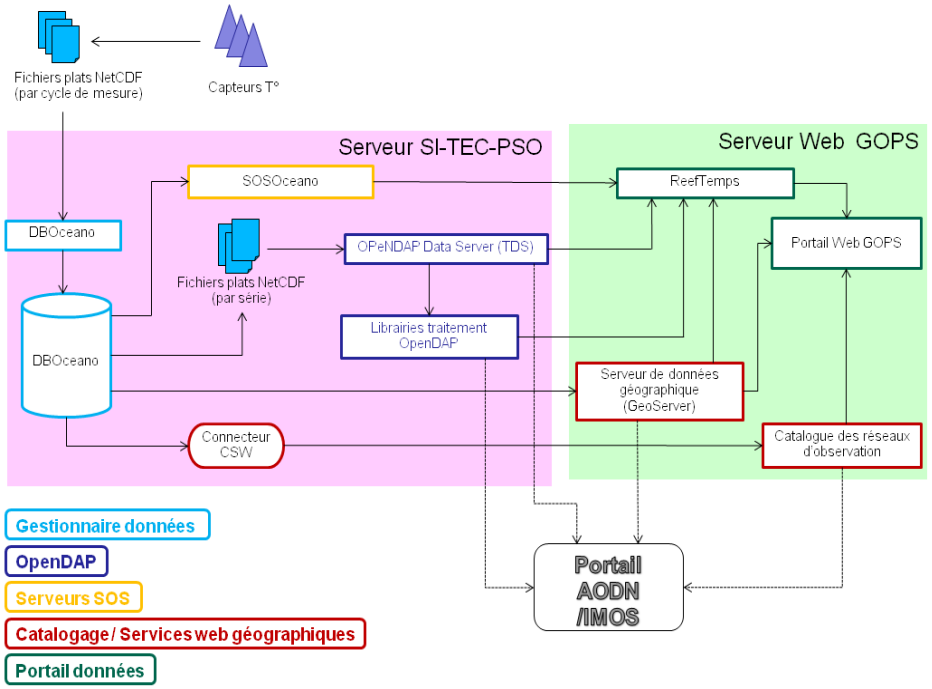
- Produire et diffuser des cartes interactives.

- Etre interopérable et alimenter en données (ou s'interfacer avec) les banques de données.

LE SI EXISTANT



L E P O R T A G E



• Points importants

- Identifier les composants applicatifs isolables (ici ils sont plutôt bien définis).

- Rechercher les images existantes et créer celles manquantes (privilégier les images officielles).

- Identifier les liens de dépendance, le partage, l'externalisation et la persistance des ressources (volumes).

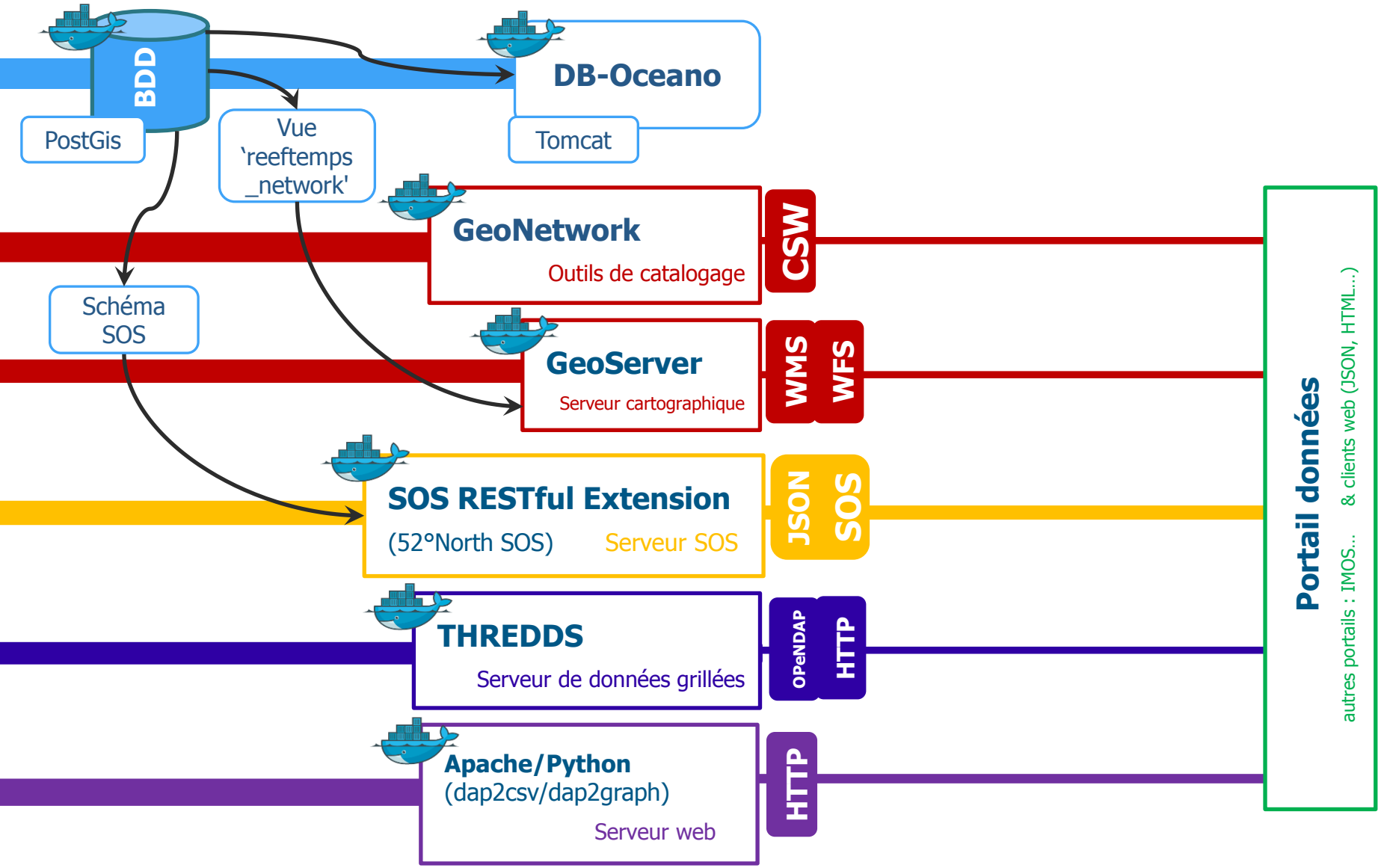
SI L'IMAGE N'EXISTE PAS

- Dockerfile + build

```
# L'image hérite de postgis qui hérite de postgres 9.6...
FROM mdillon/postgis:9.6

# Copie du script d'initialisation du serveur ssh
ADD init-ssh.sh /init-ssh-service.sh

# Installation du serveur ssh
RUN apt-get update \
        && apt-get install -y openssh-server \
        && rm -rf /var/lib/apt/lists/* \
        && chmod +x /init-ssh-service.sh

# Commande executée au démarrage du conteneur
ENTRYPOINT  ./init-ssh-service.sh ; ./docker-entrypoint.sh postgres
```

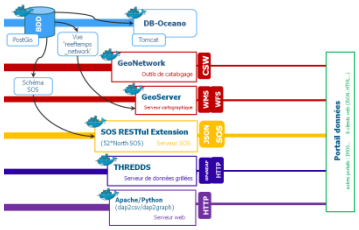
(ajout d'un serveur ssh à une image de base de données postgis)

SOUS DOCKER


BDD
PostGis
Vue 'reeftemps _network'
DB-Oceano
Tomcat
Schéma SOS
GeoNetwork
Outils de catalogage
CSW
GeoServer
Serveur cartographique
WMS
WFS
SOS RESTful Extension
(52°North SOS)
Serveur SOS
JSON
SOS
THREDDS
Serveur de données grillées
OPeNDAP
HTTP
Apache/Python
(dap2csv/dap2graph)
Serveur web
HTTP
Portail données
autres portails : IMOS...
& clients web (JSON, HTML...)

DOCKER COMPOSE

```
version: '2'
services:
  bdd:
    build:
      context: ./bdd
    expose:
      - 5432
      - 22
    volumes:
      - ./bdd/dump_folder:/data/dump_folder
      - ./bdd/data:/var/lib/postgresql/data
    environment:
      - UNIX_USER=user
      - UNIX_PASSWORD=userpassword
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=pgpassword
      - POSTGRES_DB=db

  dboceano:
    build:
      context: ./dboceano
    ports:
      - "8080:8080"
    volumes:
      - ./dboceano/import:/data/dboceano/import/
        ...
      - ./dboceano/log:/data/dboceano/log
    links:
      - bdd:bdd

  geonetwork:
    image: geocat/geonetwork:3.2.0-postgres
    ports:
      - "8081:8080"
    volumes:
       ...
```

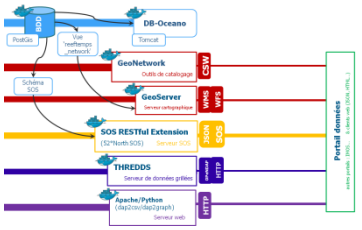


DOCKER COMPOSE : IMAGES

Images construites

Image récupérée

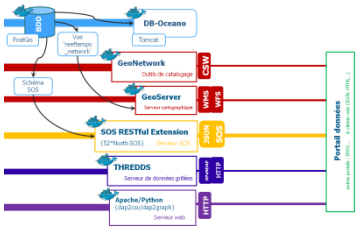
```
version: '2'
services:
  bdd:
    build:
      context: ./bdd
    expose:
      - 5432
      - 22
    volumes:
      - ./bdd/dump_folder:/data/dump_folder
      - ./bdd/data:/var/lib/postgresql/data
    environment:
      - UNIX_USER=user
      - UNIX_PASSWORD=userpassword
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=pgpassword
      - POSTGRES_DB=db

  dboceano:
    build:
      context: ./dboceano
    ports:
      - "8080:8080"
    volumes:
      - ./dboceano/import:/data/dboceano/import/
       ...
      - ./dboceano/log:/data/dboceano/log
    links:
      - bdd:bdd

  geonetwork:
    image: geocat/geonetwork:3.2.0-postgres
    ports:
      - "8081:8080"
    volumes:
       ...
```



DOCKER COMPOSE : VOLUMES

```
version: '2'
services:
  bdd:
    build:
      context: ./bdd
    expose:
      - 5432
      - 22
    volumes:
      - ./bdd/dump_folder:/data/dump_folder
      - ./bdd/data:/var/lib/postgresql/data
    environment:
      - UNIX_USER=user
      - UNIX_PASSWORD=userpassword
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=pgpassword
      - POSTGRES_DB=db

  dboceano:
    build:
      context: ./dboceano
    ports:
      - "8080:8080"
    volumes:
      - ./dboceano/import:/data/dboceano/import/
      ...
      - ./dboceano/log:/data/dboceano/log
    links:
      - bdd:bdd

  geonetwork:
    image: geocat/geonetwork:3.2.0-postgres
    ports:
      - "8081:8080"
    volumes:
      ...
```

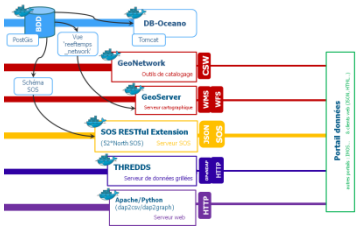
Déclarations des volumes



DOCKER COMPOSE : LINKS

```
version: '2'
services:
  bdd:
    build:
      context: ./bdd
    expose:
      - 5432
      - 22
    volumes:
      - ./bdd/dump_folder:/data/dump_folder
      - ./bdd/data:/var/lib/postgresql/data
    environment:
      - UNIX_USER=user
      - UNIX_PASSWORD=userpassword
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=pgpassword
      - POSTGRES_DB=db

  dboceano:
    build:
      context: ./dboceano
    ports:
      - "8080:8080"
    volumes:
      - ./dboceano/import:/data/dboceano/import/
       ...
      - ./dboceano/log:/data/dboceano/log
    links:
      bdd:bdd

  geonetwork:
    image: geocat/geonetwork:3.2.0-postgres
    ports:
      - "8081:8080"
    volumes:
       ...
```

Lien de dépendance entre conteneurs

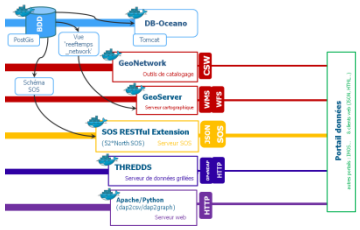


DOCKER COMPOSE : ETC...

Rendre accessibles les ports d'un conteneur

Déclarations des variables d'environnement

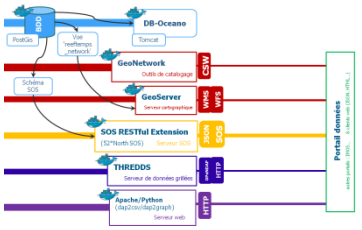
```
version: '2'
services:
  bdd:
    build:
      context: ./bdd
    expose:
      - 5432
      - 22
    volumes:
      - ./bdd/dump_folder:/data/dump_folder
      - ./bdd/data:/var/lib/postgresql/data
    environment:
      - UNIX_USER=user
      - UNIX_PASSWORD=userpassword
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=pgpassword
      - POSTGRES_DB=db

  dboceano:
    build:
      context: ./dboceano
    ports:
      - "8080:8080"
    volumes:
      - ./dboceano/import:/data/dboceano/import/
       ...
      - ./dboceano/log:/data/dboceano/log
    links:
      - bdd:bdd

  geonetwork:
    image: geocat/geonetwork:3.2.0-postgres
    ports:
      - "8081:8080"
    volumes:
       ...
```



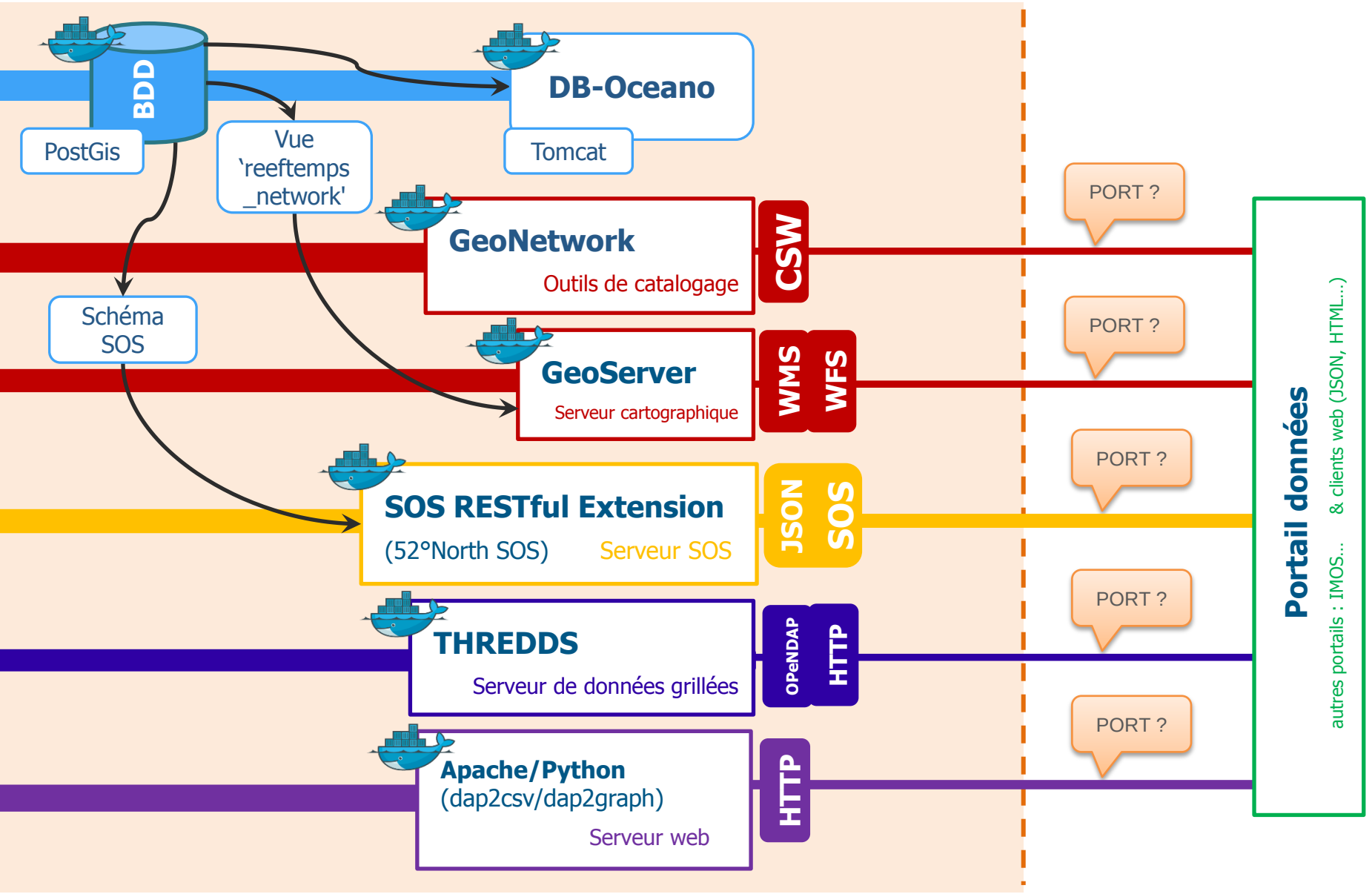
DOCKER COMPOSE : PORTS

```
version: '2'
services:
  bdd:
    build:
      context: ./bdd
    expose:
      - 5432
      - 22
    volumes:
      - ./bdd/dump_folder:/data/dump_folder
      - ./bdd/data:/var/lib/postgresql/data
    environment:
      - UNIX_USER=user
      - UNIX_PASSWORD=userpassword
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=pgpassword
      - POSTGRES_DB=db

  dboceano:
    build:
      context: ./dboceano
    ports:
      - "8080:8080"
    volumes:
      - ./dboceano/import:/data/dboceano/import/
       ...
      - ./dboceano/log:/data/dboceano/log
    links:
      - bdd:bdd

  geonetwork:
    image: geocat/geonetwork:3.2.0-postgres
    ports:
      - "8081:8080"
    volumes:
       ...
```

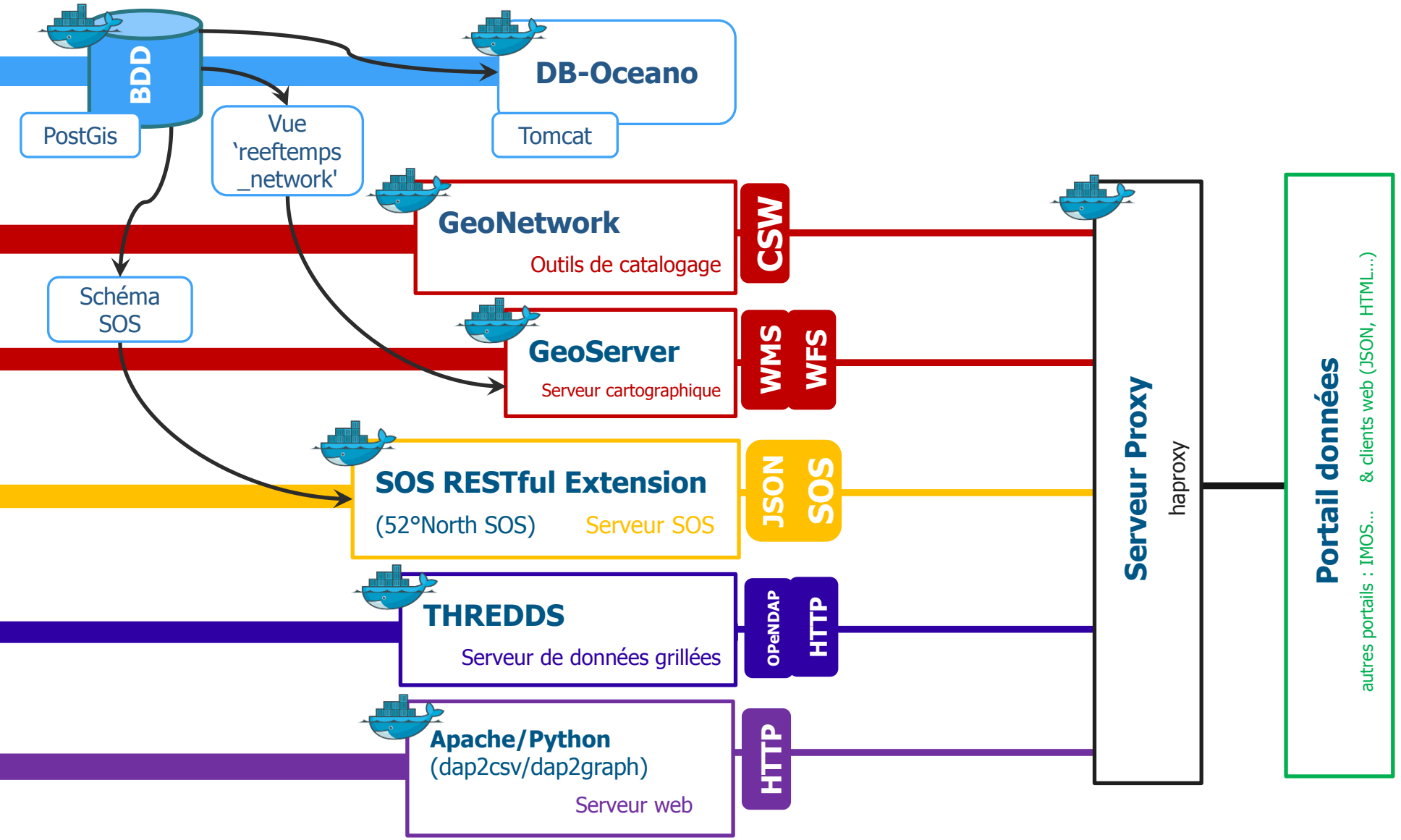
Déclarations des ports des conteneurs mappés par l'hôte



LES PORTS


BDD
PostGis
Vue 'reeftemps _network'
Schéma SOS
DB-Oceano
Tomcat
GeoNetwork
Outils de catalogage
CSW
GeoServer
Serveur cartographique
WMS
WFS
SOS RESTful Extension
(52°North SOS)
Serveur SOS
JSON
SOS
THREDDS
Serveur de données grillées
OPeNDAP
HTTP
Apache/Python
(dap2csv/dap2graph)
Serveur web
HTTP
PORT ?
PORT ?
PORT ?
PORT ?
PORT ?
Portail données
autres portails : IMOS... & clients web (JSON, HTML...)

SERVEUR PROXY



COMPOSE & HAPROXY

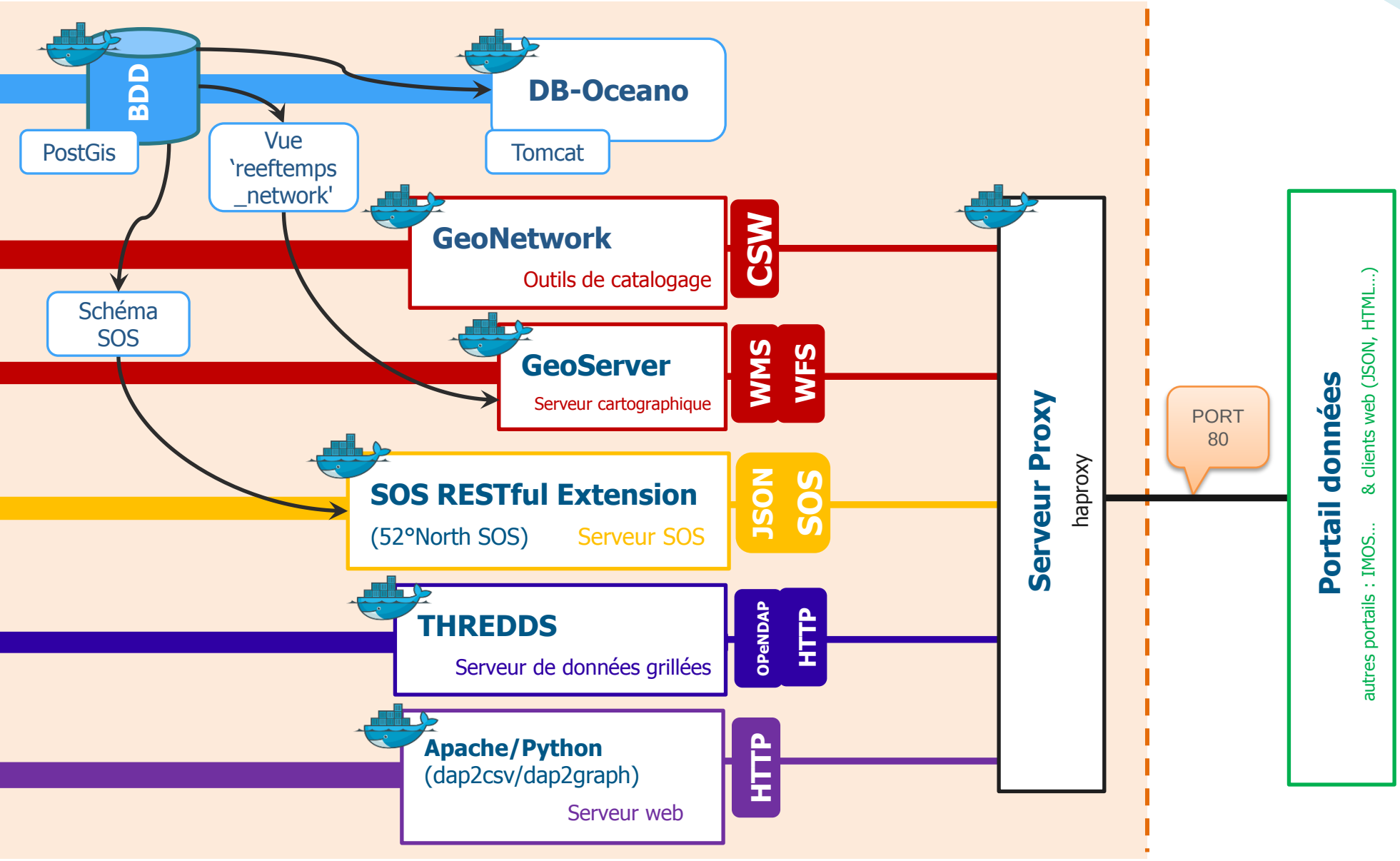
- docker-compse.yml

```
...

haproxy:
  image: docker.io/rafpe/docker-haproxy-rsyslog
  ports:
    - "80:80"
  volumes:
    - ./haproxy/haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg
  links:
    - geonetwork:geonetwork
    - thredds:thredds
    ...
```

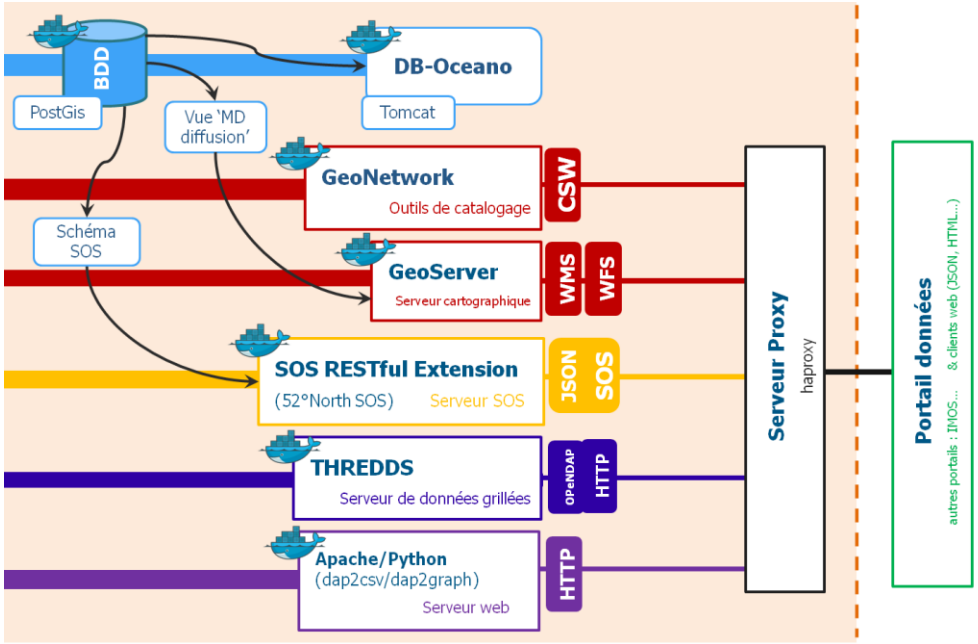
- haproxy.cfg

```
global
  log 127.0.0.1 local2
  log 127.0.0.1 local1 notice

defaults
  log global
  mode http
  ...
  timeout client 50000
  timeout server 50000

frontend http-in
  bind *:80
  acl is_geonetwork path_beg -i /geonetwork
  use_backend geonetwork_srv if is_geonetwork

  acl is_thredds path_beg -i /thredds
  use_backend thredds_srv if is_thredds
  ...

backend geonetwork_srv
  server geonetwork_srv geonetwork:8080

backend thredds_srv
  server thredds_srv thredds:8080
...
```

SERVEUR PROXY



COMPOSE ET VOLUMES



• Points importants

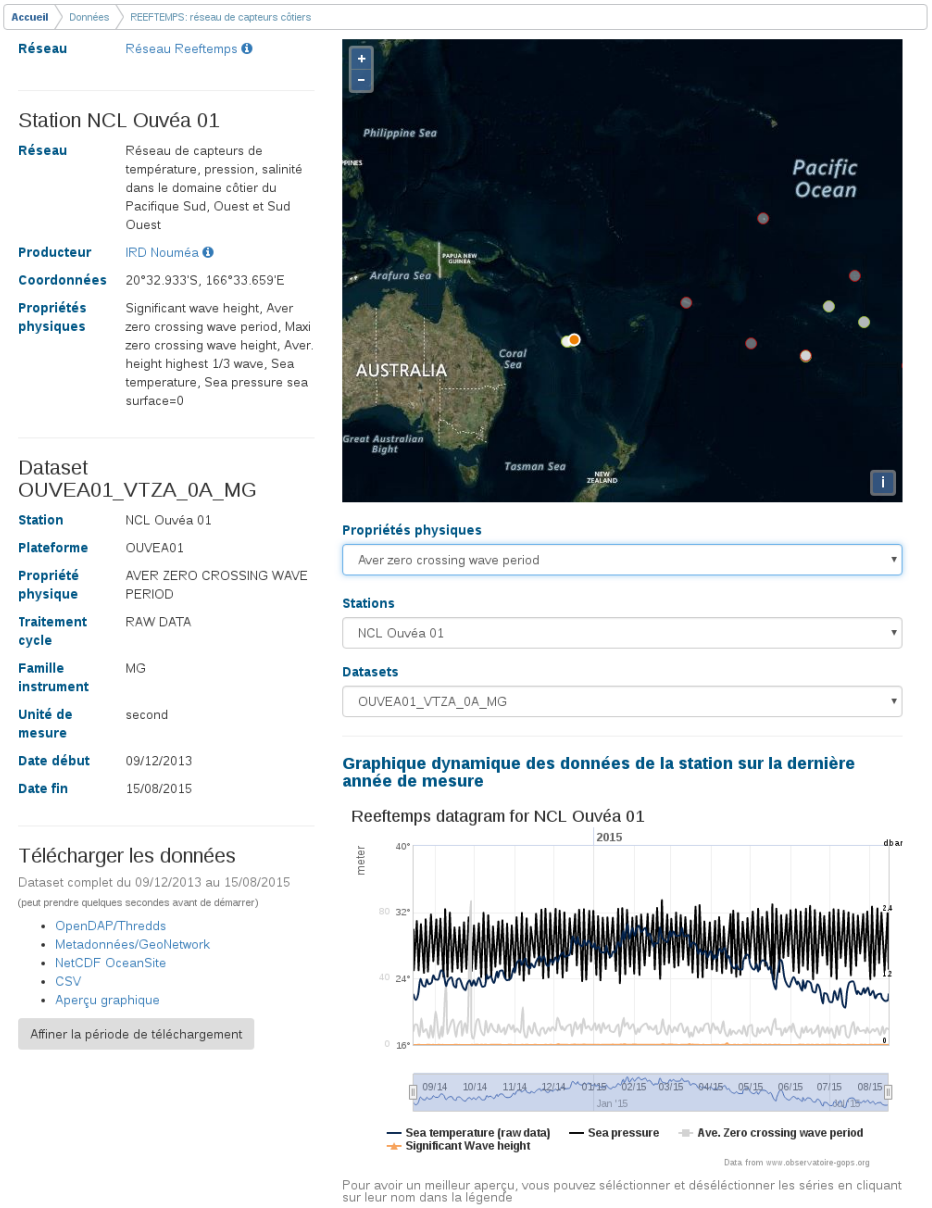
- Fichiers de configuration et logs ont été mappés pour faciliter les interventions et le suivie du système d'information.

- Fichier ressource des bases de données ont aussi été mappés pour assurer la persistance des données.

- Certains volumes sont partagés entre plusieurs conteneurs.

- Le déploiement est décrit dans le fichier docker-compose.yml, le projet est donc simple à migrer.

PAGE DE VISUALISATION



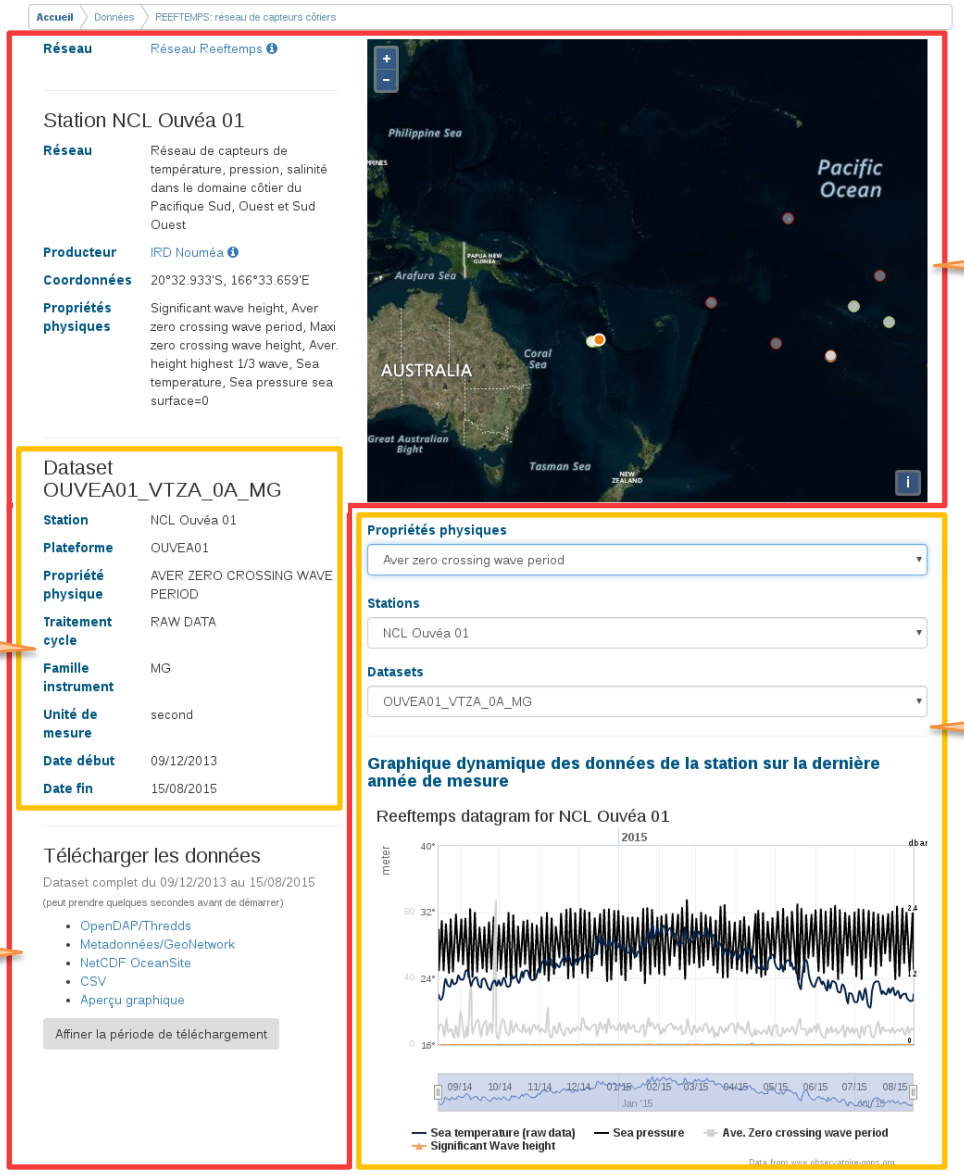
- Visualisation des stations sur carte interactive.

- Métadonnées des stations et des jeux de données.

- Lien de téléchargement et de redirection (OpenDAP, csv, png, GeoNetwork, netcdf).

- Visualisation des séries temporelles (sélection par propriété physique -> station -> jeu de données).

PAGE DE VISUALISATION



• Des atouts

- Sa simplicité, son agilité et ses performances.
- Un écosystème, communauté, partenariat et ressource.
- SOA et architecture microservice.
- Multiple cas d’usage (utilisation de composants existants, utilisable sur le cloud, reproduction des environnements, tests).

• Précautions

- Illusion de la VM.

- Portabilité des images (Linux, Windows, mac OS).

- Sécurité : user root dans le conteneur, attaque entre conteneur, exploit sur le Kernel, firewall.

M E R C I

Le site est consultable : reeftemps.observatoire-gops.org