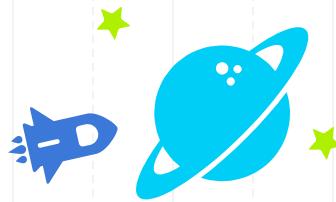


Data & Big Data

Les bases de données en python





Présentation du cours

- **Durée :** 4,5 jours
- **Objectifs :**
 - Comprendre la différence entre les types de bases de données
 - Savoir manipuler une base de données depuis un docker container
 - Interagir en python avec une base de données via SQLAlchemy
 - Maîtriser les bases de ElasticSearch et son SDK en python
- **Projets & notations**
 - QCM
 - TP & Projet final [FACULTATIF] rendu sur GitHub (+2 point → moyenne)





Overview

- Les bases de données : concept et généralités
- L'utilisation de docker : concept, généralités et étude de cas avec docker-compose
- Les bases de données relationnelles : exemple de MySQL
- Python et SQLAlchemy : interaction avec une base de données SQL
- Les bases de données non relationnelles : exemple de Elasticsearch
- Projet de fin de module [FACULTATIF]



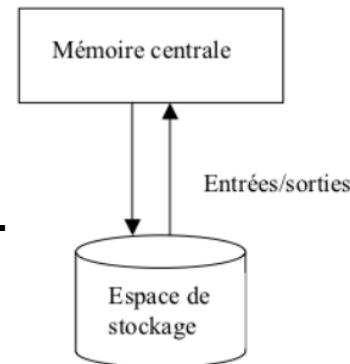


Les bases de données concept & généralités



Le concept de données et de support

- La première vraie question est : **qu'est-ce qu'une donnée ?**
- Une **information quelconque**, ex : « *Voici une personne, elle s'appelle Benjamin.* »
- **Relation entre des informations**, ex : « *Benjamin enseigne les bases de données à H3Hitema, c'est vraiment cool* »
- Des relations de ce genre définissent des structures.
- Définition : **une base de données (BDD) est un ensemble d'informations structurées mémorisées sur un support permanent.**





Data => Réglementation : le RGPD rapidement



Cartographier et tenir des registres



Minimisation des données et respect des durées de conservation



Informer les salariés et candidats et obtenir leur consentement



Garantir la sécurité et la confidentialité des données



Désigner un DPO externe

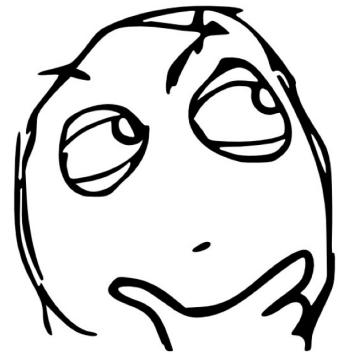
L'entreprise détentrice de vos données





Le concept de système de gestion de base de données SGBD

- La question : pourquoi ne pas utiliser un (ou plusieurs) fichier(s) stockés sur mémoire secondaire ?
 - lourdeur d'accès aux données,
 - manque de sécurité,
 - pas de contrôle d'accès !
- On a donc recours à un logiciel (un SGBD) chargé de **gérer les fichiers** constituant une base de données, de **prendre en charge les fonctionnalités de protection et de sécurité** et de fournir les différents types d'interface nécessaires à l'accès aux données.





SGBD : architecture

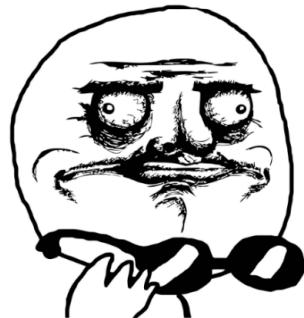
- La plus grande partie des SGBD existants ont une architecture standard qui distingue 3 niveaux :
- **Niveau physique** : gestion sur mémoires secondaires des données, du schémas, des index ; partage des données et gestion de concurrence d'accès ; reprise sur pannes (fiabilité) ; distribution des données et interopérabilité (accès aux réseaux & API si besoin).
- **Niveau logique** : définition de la structure des données, langage de description de données (LDD), consultation et mise à jour des données, langage de requêtes (LR), langage de manipulation de données (LMD), gestion de la confidentialité (sécurité), maintien de l'intégrité.
- **Niveau externe** : vues, environnement de programmation, interface, outils d'aide, outils de saisie, d'impression d'états.





SGBD : utilisation

- L'utilisation d'un SGBD suppose de comprendre les fonctionnalités suivantes :
- **Définition du schéma de données** : description des données contenues dans la base conforme à un modèle de données qui propose des outils de description (structures, contraintes et opérations).
- **Opérations sur les données** : la création, la modification, la destruction et la recherche.
- **Partage des données entre plusieurs utilisateurs** : éviter les blocages tout en empêchant des modifications anarchiques.
- **Optimisation des performances d'une requête en s'appuyant sur l'organisation physique des données** : les fichiers séquentiels, les index et le regroupement des données par hachage.





Pourquoi les BDD ?

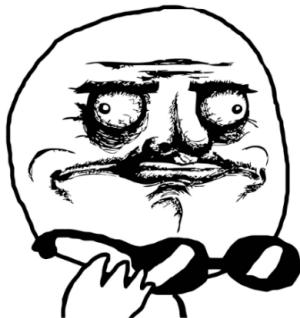
- Il est légitime de se demander pourquoi préférer une base de données plutôt qu'un tableur Excel ou même des listes de documents ?
- Car on peut **poser plus facilement une question** à une base de données, du type : « Combien de personnes ont accepté de participer à l'événement ... » ou « Quel client a acheté ce produit, en quelle quantité ».
- Tout comme les tableurs, les bases de données organisent vos informations de manière précise. Cette organisation va dépendre du type de votre base (relationnelle, sous forme de documents, de graphes...) nous aborderons cela dans la suite de ce cours.
- En résumé, les bases de données sont bien plus efficaces lorsqu'il s'agit de retrouver des informations.





BDD => définition du schéma (ou modèle) de données

- Le schéma de données ou modèle entité-association ou en anglais « **entity-relationship diagram** », abrégé en **ERD**, est un modèle de données ou diagramme pour des **descriptions de haut niveau de modèles conceptuels de données**.
- Il fournit une description graphique pour représenter de tels modèles de données sous la forme de diagrammes contenant des entités et des associations.
- Ils sont souvent utilisés dans les phases amont de conception des systèmes informatiques.





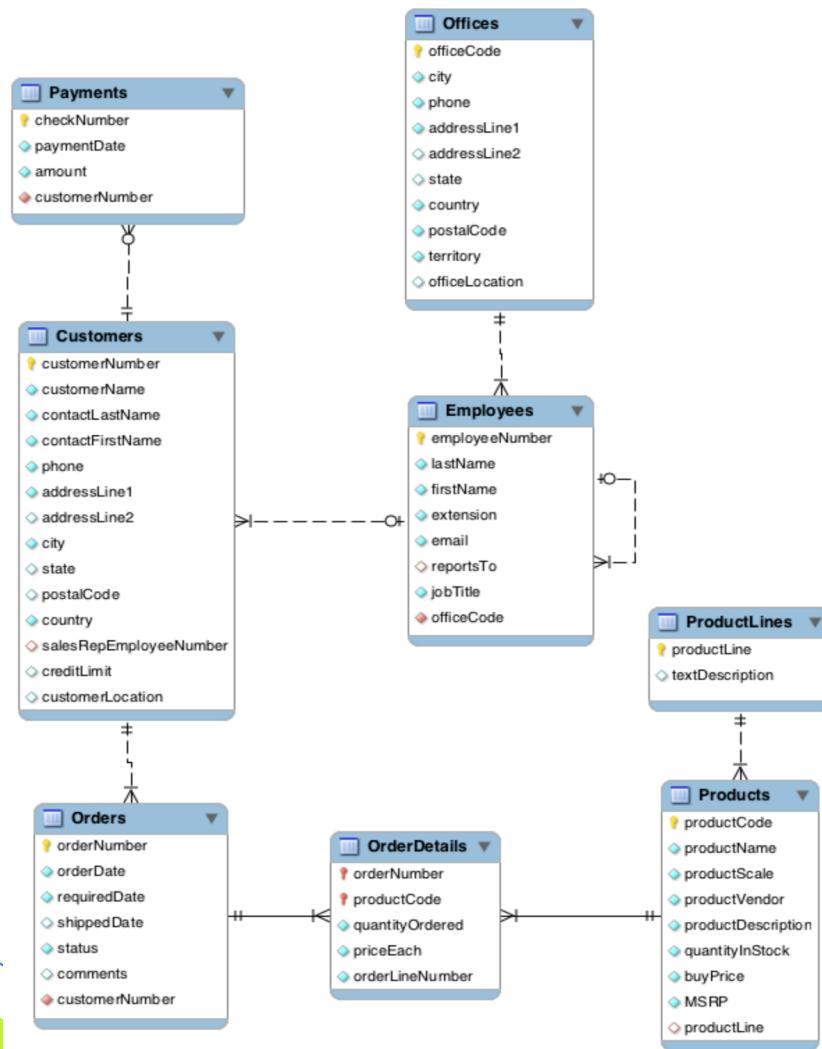
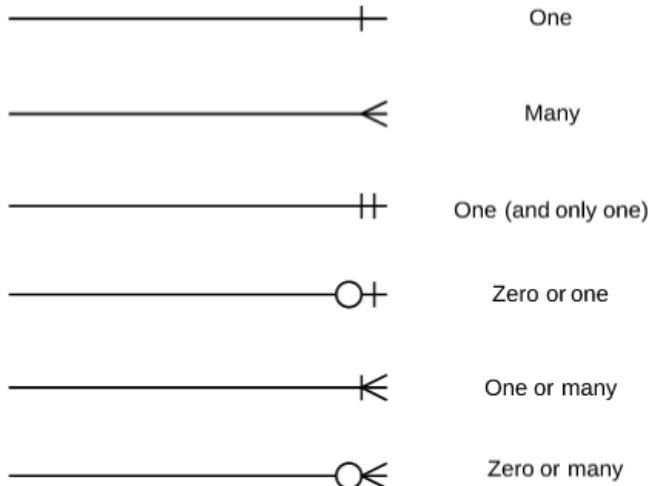
Le modèle entité association (E\A) : définitions

- **Entité** : tout objet du monde réel (**correspond aux noms dans une phrase**)
- **Association** : tout lien entre objets (**correspond aux verbes dans une phrase**)
- **Attribut** : toute propriété d'une entité ou d'une association
- **Cardinalité** : quantification du lien en terme d'instances





Exemple de base de données Retail





L'utilisation de Docker

concept, généralités et étude de cas



Docker overview

- **Les conteneurs:**
 - principes, objectifs et solutions.
- **Docker en pratique :**
 - Installation de Docker, Conteneurs et images, le cycle de vie des conteneurs, lancer son premier conteneur, accéder au conteneur et construire une image Docker : le Dockerfile et ses instructions
- **Prise en main du client Docker :**
 - Introduction a la CLI Docker, interactions avec un conteneur démarré, commandes relatives aux images, réseau et volumes.
- **Bonnes pratiques :**
 - Variables d'environnement et conteneurs: ENV, méta-information et images: LABEL. Paramétrier le build d'une image, modifier le contexte système au cours du build et Auto-guérison (self healing).



Les conteneurs: principes, objectifs et solutions

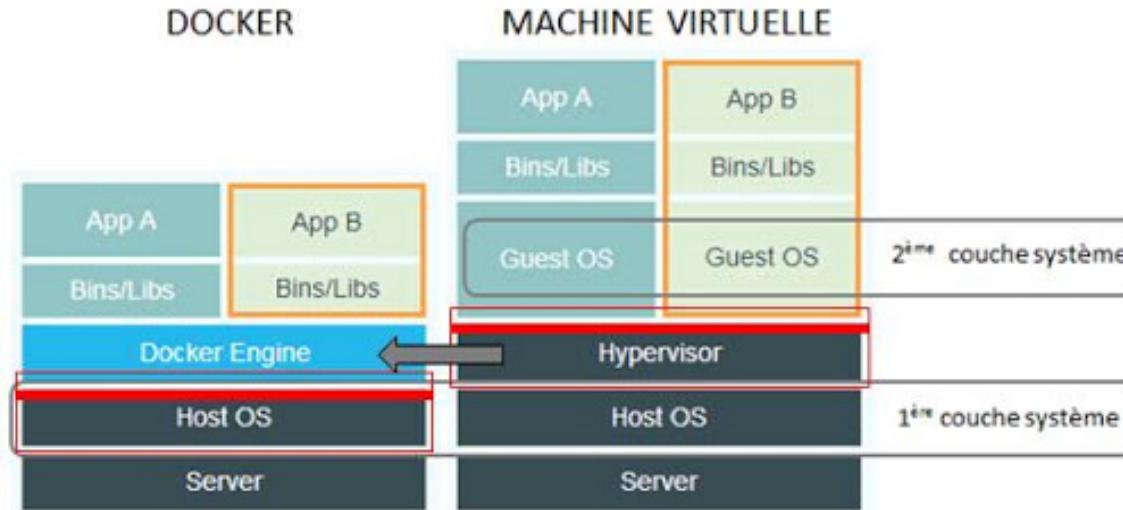
- Les conteneurs logiciels **cherchent à répondre à la même problématique que les conteneurs physiques** aussi appelés conteneurs intermodaux, (les grandes boites de métal standardisées) **qui transportent de nos jours l'essentiel des biens matériels** par camion, train, avion ou bateau.
- Dans une architecture à base de conteneurs:
 - **Le contenu du conteneur**, c'est-à-dire le code et ses dépendances (jusqu'au niveau de l'OS), est de la **responsabilité du développeur**.
 - **La gestion du conteneur et des dépendances** que celui ci va entretenir avec l'infrastructure (stockage, réseau ou calcul) sont de la **responsabilité de l'exploitant**.
- Par ailleurs, **il existent aussi des conteneurs autre que Docker** pour d'autres systèmes, comme FreeBSD ou Solaris. Nous ne les aborderons pas dans le cadre de ce cours.





Comment les conteneurs autorisent la réalisation de nouvelles architectures informatiques

- La première chose à comprendre c'est la différence entre la conteneurisation et la virtualisation.





Comment les conteneurs autorisent la réalisation de nouvelles architectures informatiques

- Une architecture à base de conteneurs offre une **solution de compromis**.
- Le conteneur offre **l'isolation** permettant à un développeur d'embarquer l'ensemble des dépendances dont il a besoin **y compris les dépendances de niveau OS**.
- De plus, un conteneur s'appuie sur le noyau du système hôte. Il est donc **très léger et démarre presque aussi vite que le processus qu'il encapsule**.
- L'idée est donc de **simplifier au maximum le système d'exploitation** en éliminant les services non utilisés par les applicatifs.
- **Le nombre de conteneurs qu'un même hôte peut exécuter est donc nettement plus élevé que son équivalent en machines virtuelles.**





Qu'est-ce qu'un conteneur finalement ?

- Un conteneur est tout simplement **un système de fichiers sur lequel s'exécutent des processus** (type application) de manière:
 - **Contrainte** : grâce à CGroups (Control Groups en référence à la commande chroot qui permet de partitionner les ressources d'un hôte comme le processeur, la mémoire, les accès au réseau ou à d'autres terminaux) qui spécifie les limites en termes de ressources.
 - **Isolée**: grâce notamment au fait que les conteneurs ne se voient pas les uns les autres si ils ne partagent pas un même réseau.





Docker en pratique : Installation

- **Sur Ubuntu**

Aller sur la documentation officielle et installer docker en ligne de commande

- **Sur Mac OS**

Aller sur la documentation officielle et télécharger l'application Docker Desktop

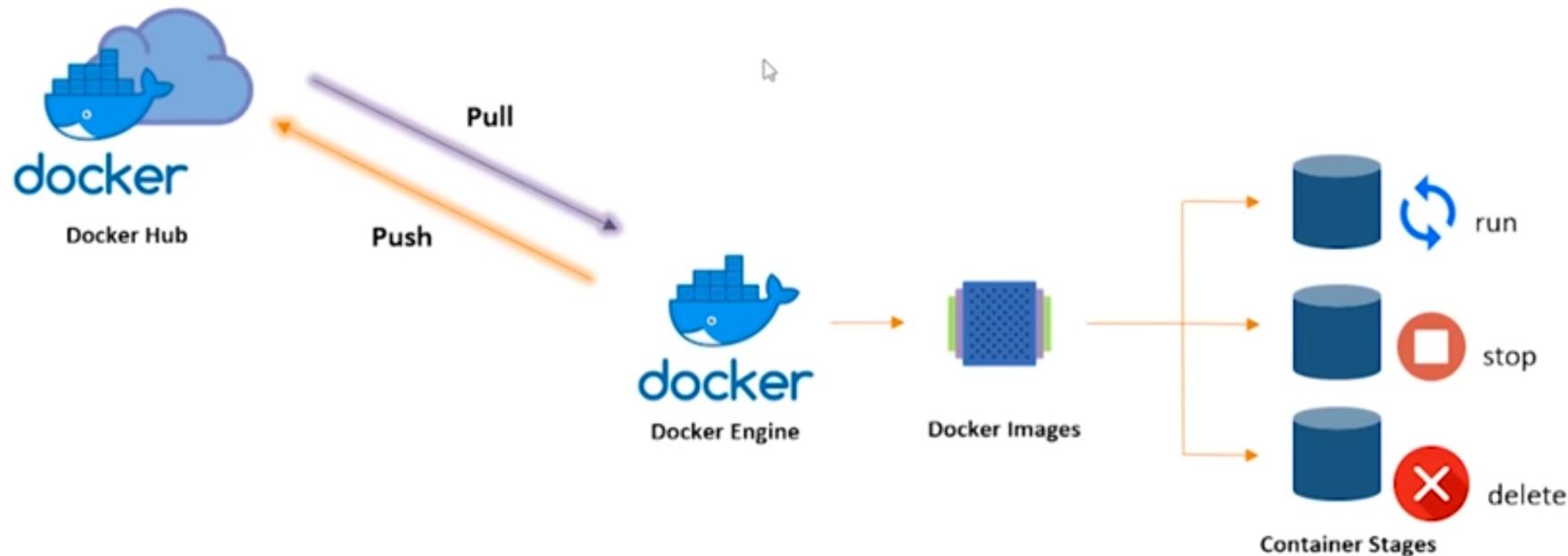
- **Sur Windows :: Installer une distribution linux sur VM**

Vous pouvez utiliser pour cela VirtualBox ou VMWare

- Vérifier pour installation avec la commande : *docker info*



Le cycle de vie des conteneurs



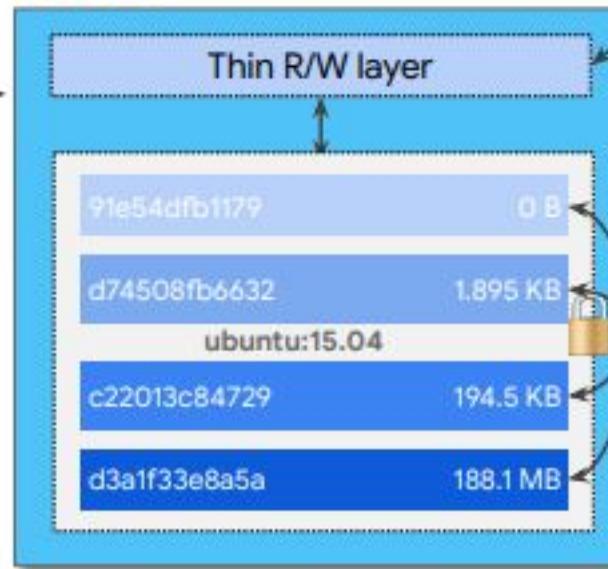


Le concept de image layers

Dockerfile

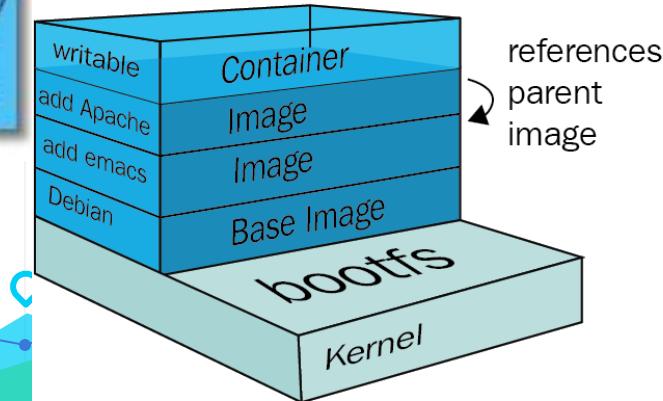
```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

Container



docker commands

```
$> docker build -t py-web-server .
$> docker run -d py-web-server
$> docker images
$> docker ps
$> docker logs <container id>
$> docker stop py-web-server
```





Construire une image Docker : Dockerfile et ses instructions

- Repose sur peu d'instruction mais il est important de bien les comprendre.
Ci-dessous, les principales instructions à connaître :

FROM : Specify the base image

MAINTAINER : Specify the image maintainer

RUN : Run a command

ADD : Add a file or directory

EXPOSE : expose ports to be accessed

ENV : Create an environment variable

CMD : What process to run when launching a container from this image.





Introduction à la CLI Docker : vocabulaire



Docker Engine

- ★ Virtualizes OS, not hardware
- ★ Infrastructure agnostic
- ★ Best for Microservices

Container

- ★ Runs only what you need
- ★ Runs minimal OS, app runtime, file system, etc.
- ★ Runs in its own process

Application

- ★ Application Runtime
- ★ Application Dependencies
- ★ Environment Setting

Docker Engine - Manages and creates Docker images and containers

Container - Isolated process running application, runtime and dependencies

Dockerfile - Contains instructions to assemble the image

Image - Read-only executable for creating containers

Application - Binaries, libraries, and dependencies needed to run application

Docker Hub - Hosts public repositories of Docker images

Local image repository - Stores Docker images locally



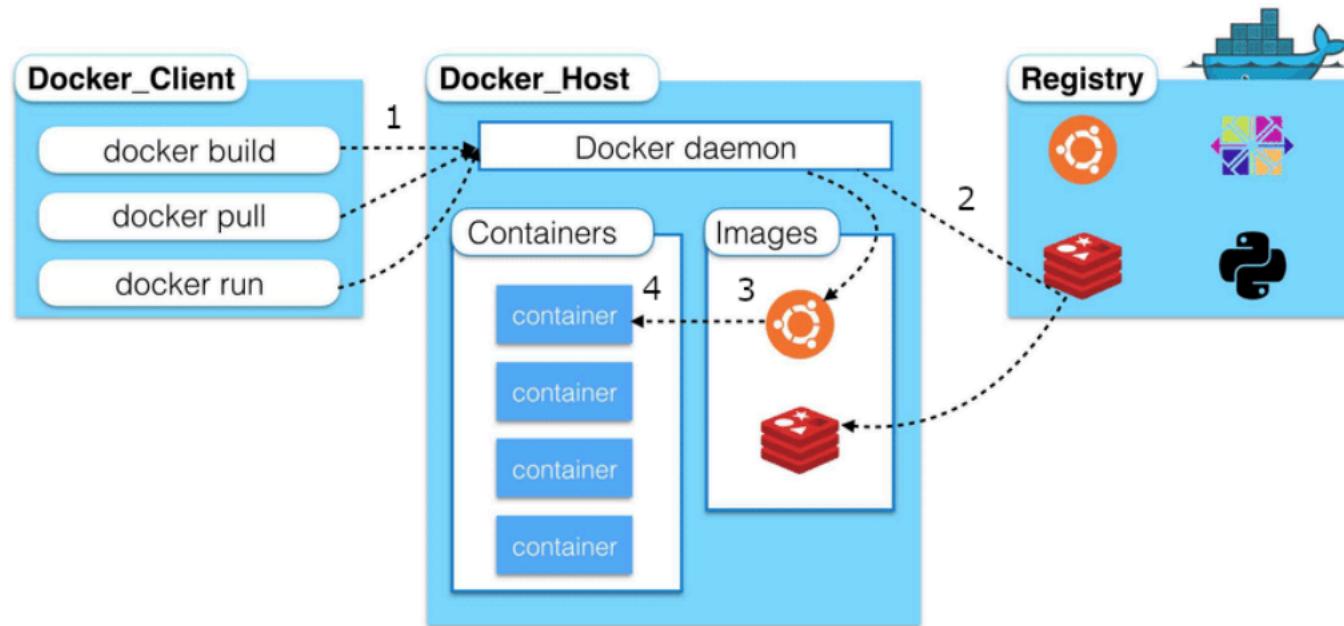
Exemple : Hello world

● Source : [isatorres](#)

Running a Docker container Scenario: Containerized Hello World!				
Steps	What happens?	Image Repository	Container Runtime	Good things to know
Create Dockerfile to assemble image <code>touch Dockerfile</code>	Nothing yet, it's just an empty text file	Empty	Empty	Sample Dockerfile: <code>FROM alpine CMD ["echo", "hello world!"]</code>
Build image <code>docker build dockerfilePath -t imageName</code>	Image is created and added to the local repo	○	Empty	Instructions in the Dockerfile tells Docker where to place files and how to start the application
Create container from image <code>docker run imageName</code> Try it! <code>docker run hello-world</code>	A container process is created from the image	○ ↓ ○	○ ↓ ○	Docker downloads an image named <i>imageName</i> from Docker Hub if not found in the local image repo. A random container name is generated. Provide your own name: <code>docker run --name containerName imageName</code>
 Application is running in an isolated process! The container runs as long as the contained process with PID 1 is running (enter <code>pid 1</code> in a container terminal). Note: Short-running PID 1 processes = short running containers				List all containers (running and exited) <code>docker ps -a</code>
Start a stopped container <code>docker start containerName</code>	The application is started.	○	○ green arrow	Get a terminal to the container <code>docker exec -it containerName sh</code>
Stop the container process <code>docker stop containerName</code>	The application is terminated.	○	○ red square	Show container logs <code>docker logs containerName</code>
Remove stopped container <code>docker rm containerName</code>	Container is removed. The image stays in the local repo.	○	Empty	List local images <code>docker images</code>
Remove local image from local repo <code>docker rmi imageName</code>	Image is removed. Dockerfile persists in the file system.	Empty	Empty	All containers created from an image must be removed before removing an image



Docker pull





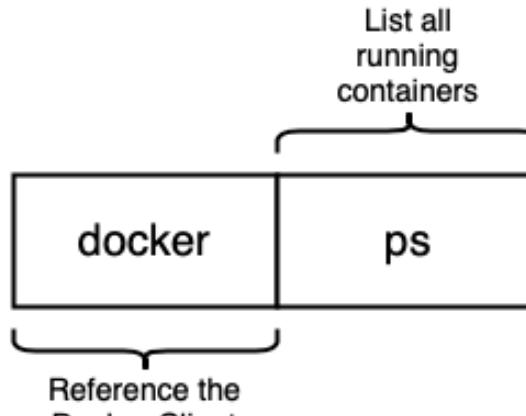
Les principales commandes Docker

Introduction au CLI Docker



Les processus Docker : docker ps

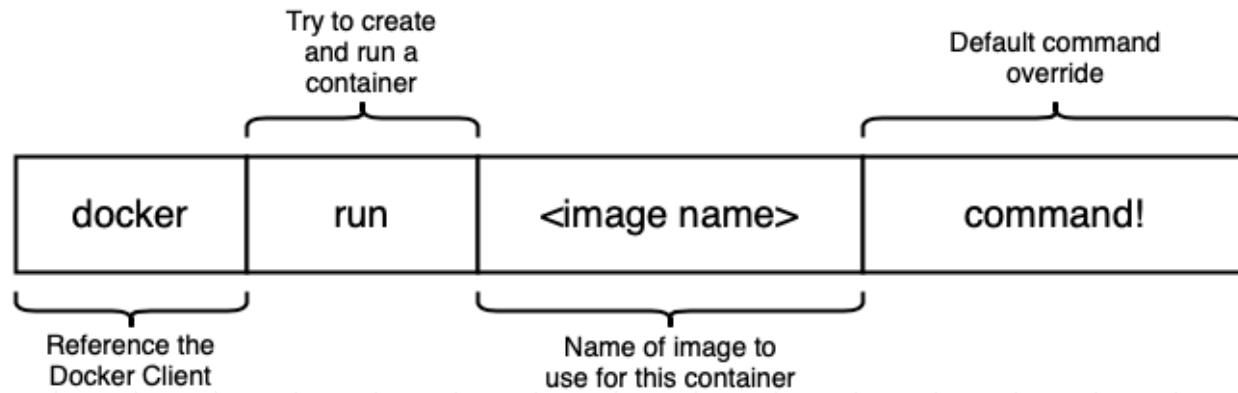
List all running containers





Lancer un container Docker : docker run

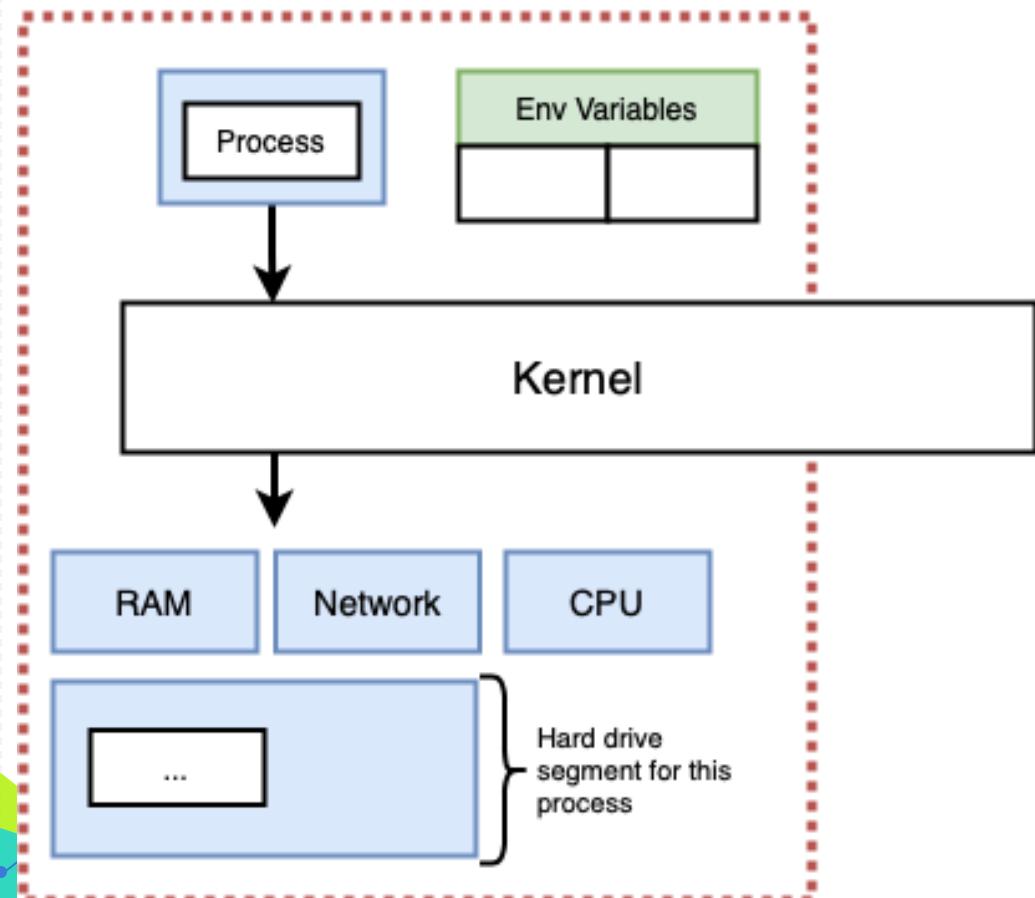
Creating and Running a Container from an Image





Les commandes run : variables d'environnement

Container





Docker run : explication

`docker run`

= `docker create`

+ `docker start`

Try to create
the container

Create a Container

`docker`

`create`

`<image name>`

Reference the
Docker Client

Name of image to
use for this container

Try to create
the container

Start a Container

`docker`

`start`

`<container id>`

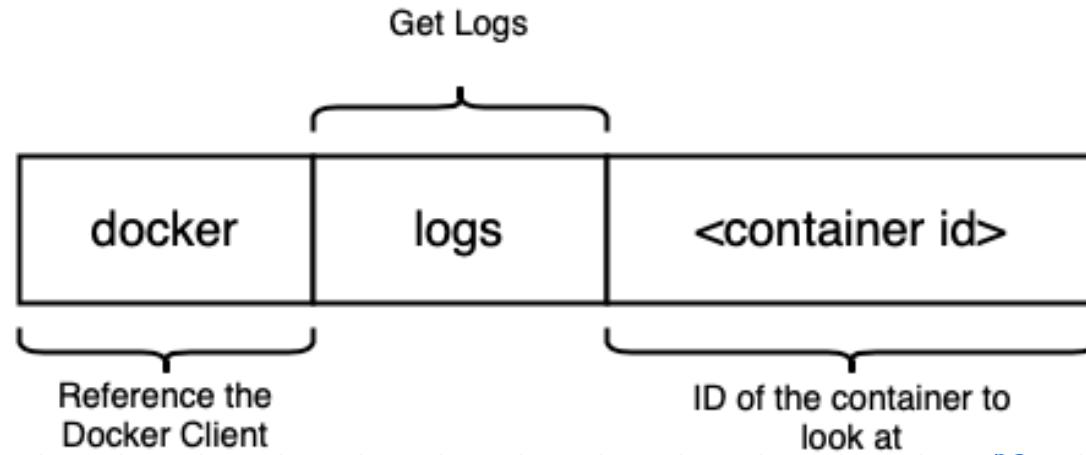
Reference the
Docker Client

ID of the container to
start



Docker logs : regarder ce qu'il se passe à l'intérieur / debug

Get logs from a container





Docker stop/kill

Stop a Container

docker	stop	<container id>
--------	------	----------------

Stop laisse du temps au programme de se fermer correctement (~10s)
Kill va directement fermer le programme sans tenir conte du reste (~1s)

Kill a Container

docker	kill	<container id>
--------	------	----------------





Docker delete container/images

Delete image

docker	rmi	<image id>
--------	-----	------------

Delete container

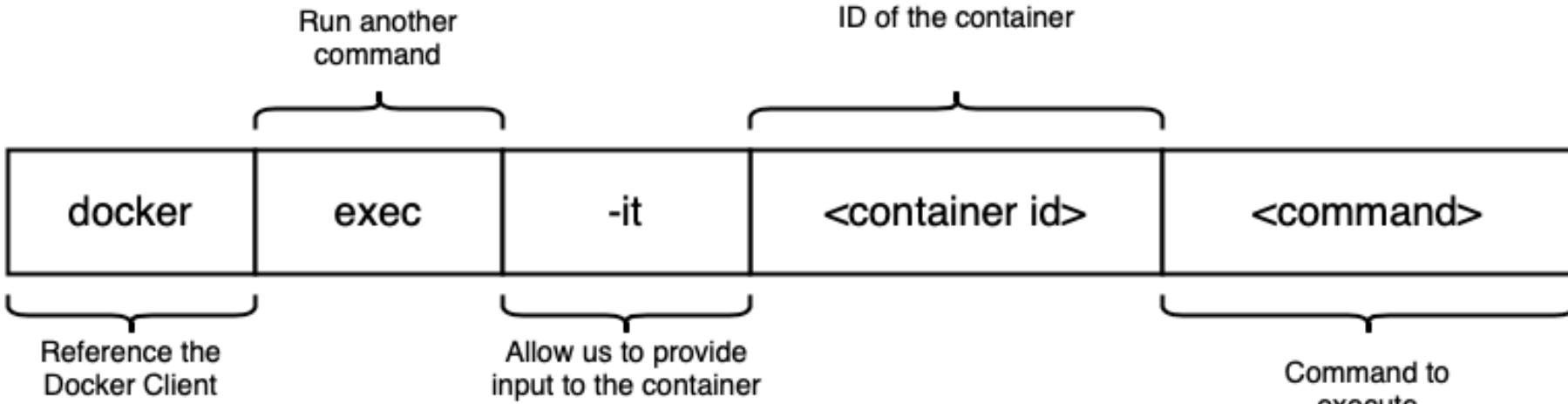
docker	rm	<container id>
--------	----	----------------





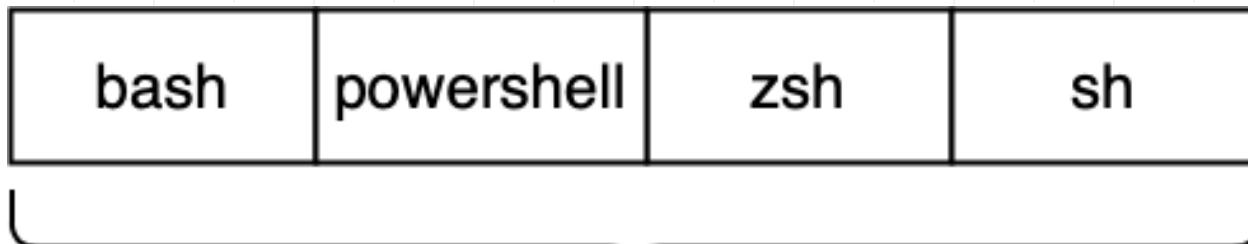
Docker exec : inside the container box

Execute an additional command in a container





Docker exec : general command I shell



Command
Processors

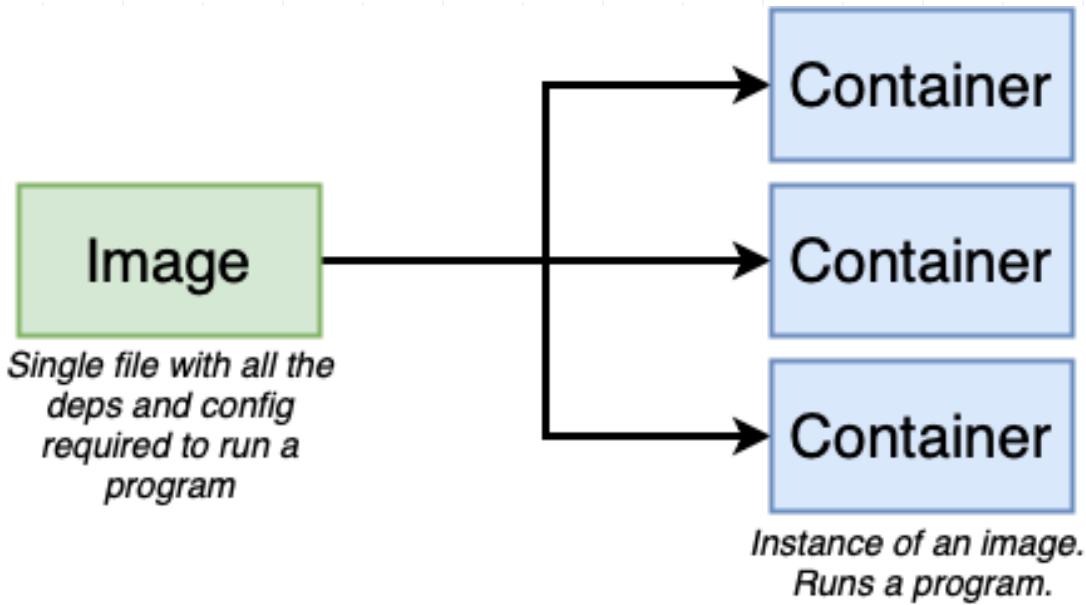




Construire une image docker Instancier un container à partir d'une image personnalisé



Lien entre image & container





Construire une image Docker : Dockerfile et ses instructions

- Repose sur peu d'instruction mais il est important de bien les comprendre.
Ci-dessous, les principales instructions à connaître :

FROM : Specify the base image

MAINTAINER : Specify the image maintainer

RUN : Run a command

ADD : Add a file or directory

EXPOSE : expose ports to be accessed

ENV : Create an environment variable

CMD : What process to run when launching a container from this image.

Flow

Specify a base image

Run some commands to install additional programs

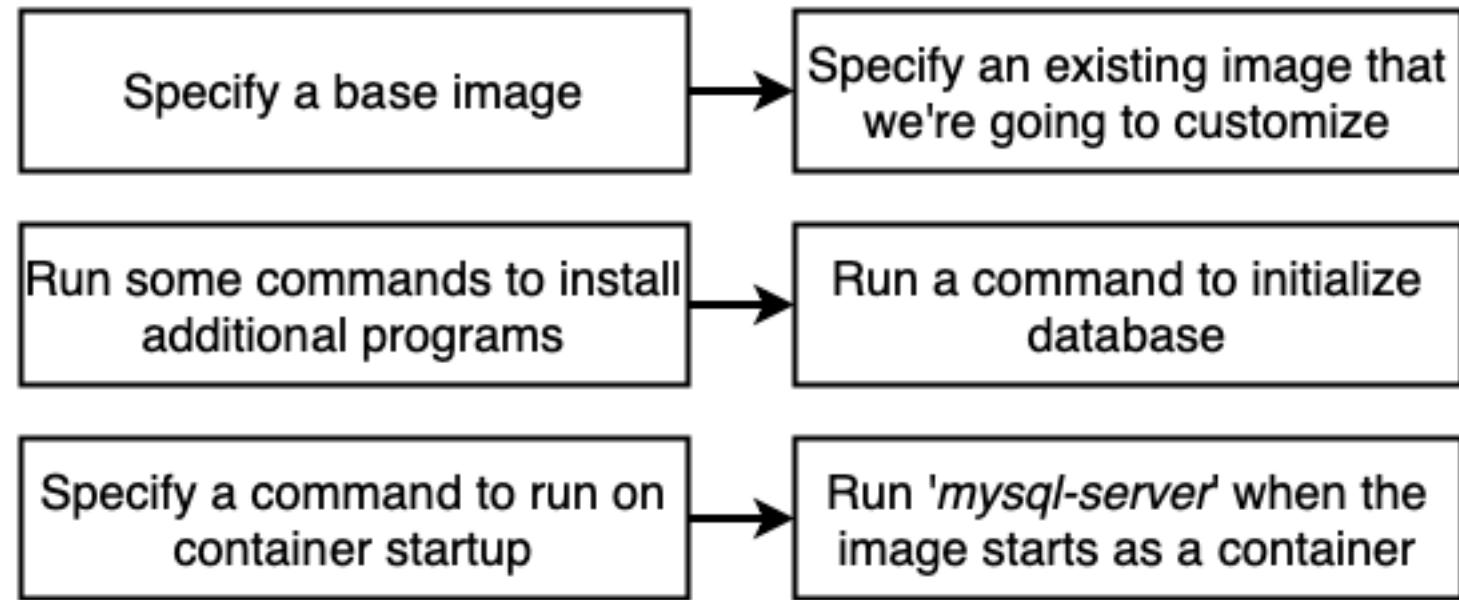
Specify a command to run on container startup





Exemple avec une image MySQL afin de conteneuriser une base existante

Flow





Exemple simple

Conteneuriser une application python

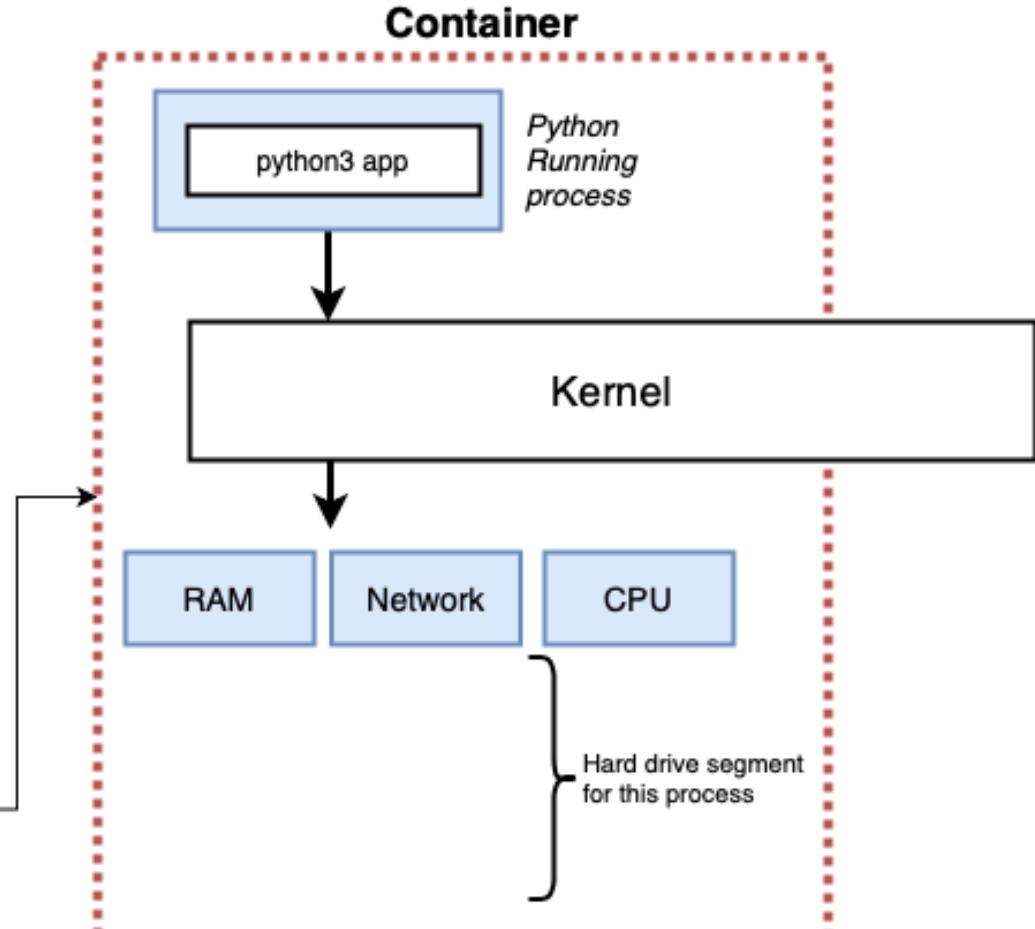
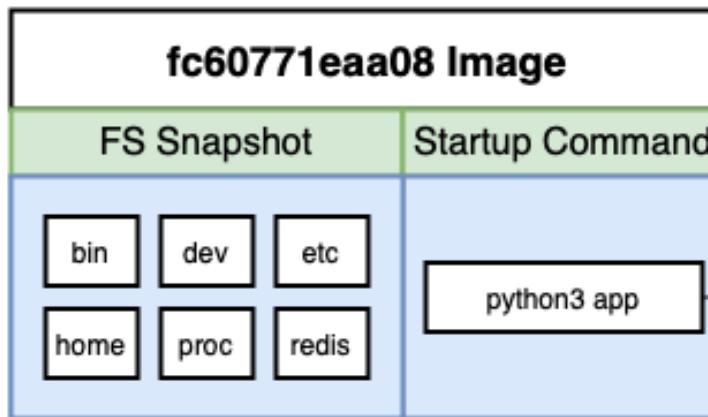
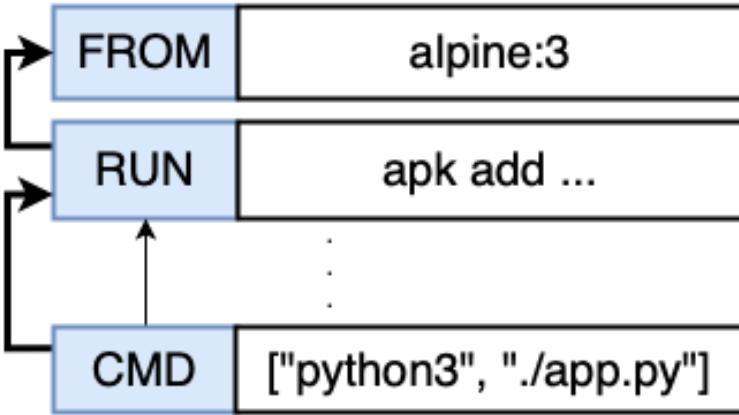


Solution

Dockerfile

Instruction telling Docker Server what to do	Argument to the instruction	
FROM	alpine:3	} := version / tag
WORKDIR	/usr/src/app	} mkdir repo inside the container and go inside
RUN	apk update && apk add python3 && apk add py3-pip && apk add vim	} install some additional programs
COPY	..	} cp content of your local repo inside the container workdir
RUN	pip3 install -r requirements.txt	} install additional python programs
EXPOSE	5000	} allow port to be used by the container
CMD	["python3", "./app.py"]	} command that the container needs to run

Solution I Inside the box





Les volumes docker

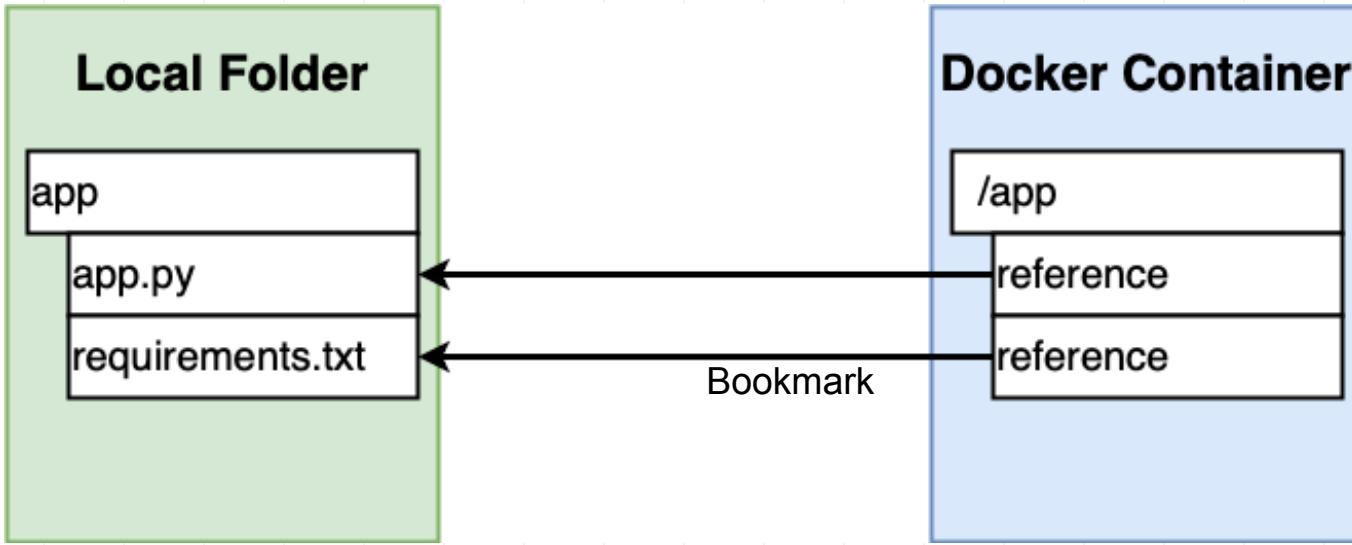


Les volumes docker - généralités

- L'instruction VOLUME permet de créer un point de montage dans l'image.
- Ce dernier se réfèrera à un emplacement, appelé volume, dans l'hôte ou dans un autre conteneur.
- À noter que dans nos exemples les deux dossiers (dans le conteneur et dans l'hôte) sont nommés de manière identique, cependant leurs noms pourraient être différents.
- On peut les utiliser dans un Dockerfile ou bien dans un docker-compose.yml



Volumes - en image



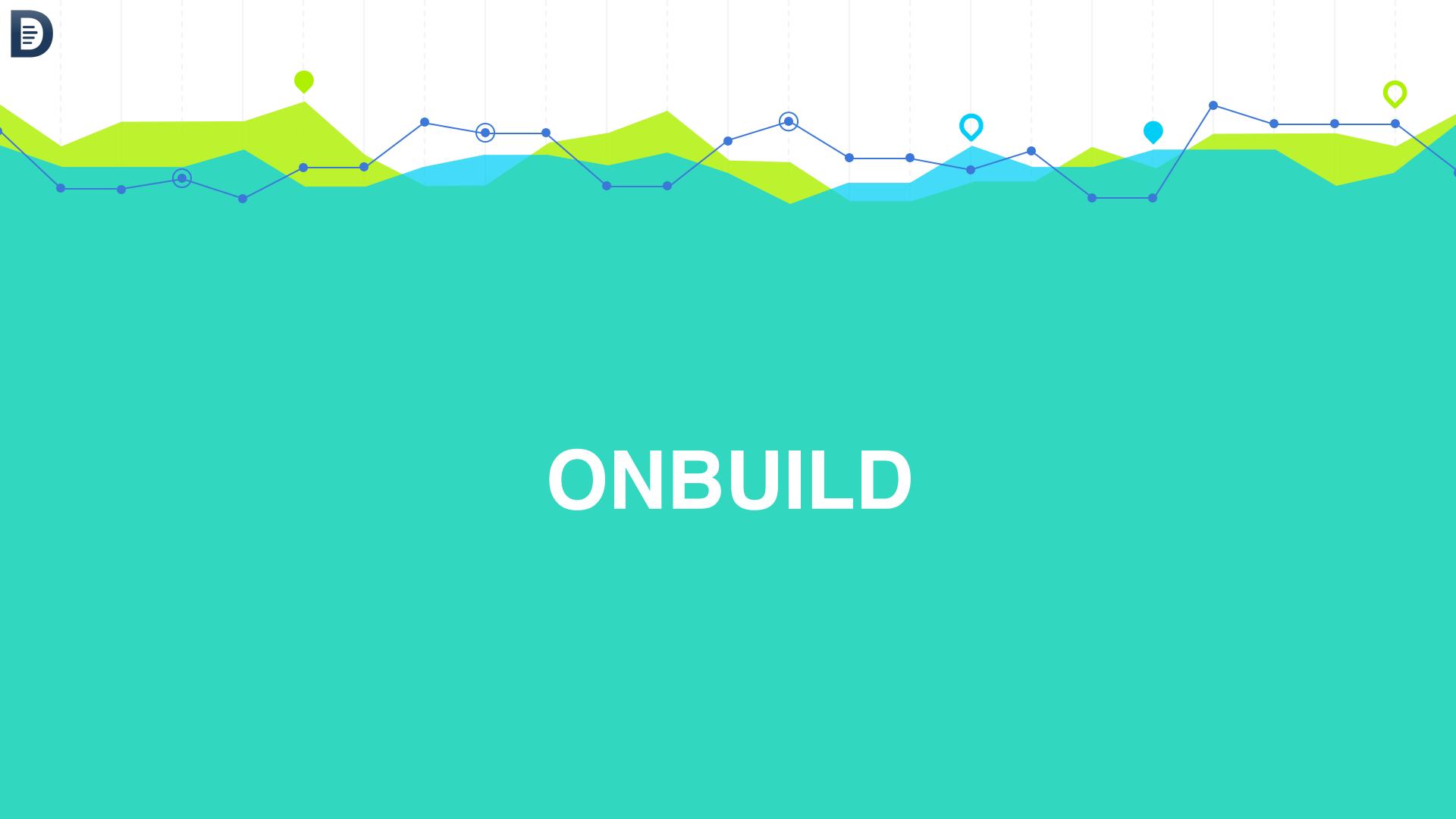


Monter un volume lors du run

```
docker run -p 3000:3000 -v /app/ -v $(pwd):/app <image_id>
```

Put a bookmark on the app folder

Map the pwd into the '/app' folder

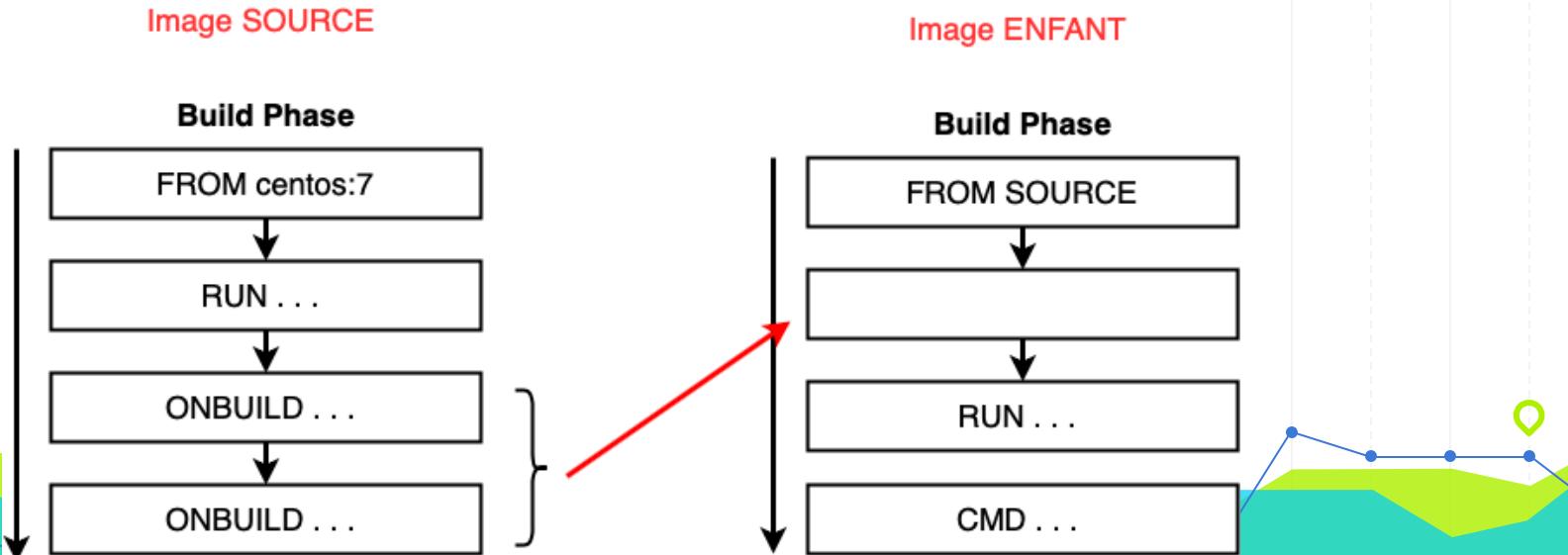


ONBUILD



L'instruction ONBUILD

- ONBUILD permet de définir des instructions qui seront exécutés uniquement lors de la construction d'images enfants.
- L'exécution des instructions ONBUILD n'est effectuée que dans les enfants directs (pas les enfants d'enfants).





Compose

Introduction à l'orchestration de container avec docker-compose



Docker-compose : KASAKO

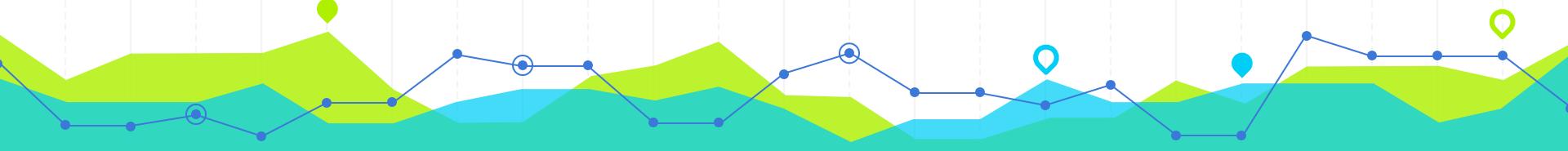
- Compose est un outil Docker permettant de définir et **d'exécuter des applications multi-conteneurs.**
- Il permet **avec un fichier YAML de configurer les services de votre application.**
- Ensuite, **avec une seule commande, vous créez et démarrez tous les services** à partir de votre configuration.
- Il faut donc créer un **fichier docker-compose.yml pour orchestrer plusieurs conteneurs.**





Le principe de services / micro services

- Les microservices sont une **technique de développement logiciel très à la mode** en ce moment
- C'est une variante du style architectural orientée services (SOA), qui structure une application comme un ensemble de **services faiblement couplés**.
- Les microservices, peuvent être **des applications indépendantes qui communiquent** les unes avec les autres en utilisant des API.





YALM QuickStart

- YAML, acronyme de **Yet Another Markup Language** est un format de représentation de données par sérialisation Unicode. Il reprend des concepts d'autres langages comme XML.
- Son objet est de **représenter des informations plus élaborées que le simple CSV en gardant cependant une lisibilité presque comparable, et bien plus grande que XML**.
- Très utilisé pour les **fichiers de configuration, notamment dans le CI/CD** que nous allons voir plus tard.





Exemple d'une fiche de commande (source wikipedia)

- L'idée de YAML est que **presque toute donnée peut être représentée par combinaison de listes, tableaux associatifs et données scalaires.**
- YAML comme le python est **sensible à l'indentation (tabulation)**

```
---  
receipt: Oz-Ware Purchase Invoice  
date: 2012-08-06  
customer:  
  given: Dorothy  
  family: Gale  
  
items:  
  - part_no: A4786  
    descrip: Water Bucket (Filled)  
    price: 1.47  
    quantity: 4  
  
  - part_no: E1628  
    descrip: High Heeled "Ruby" Slippers  
    size: 8  
    price: 100.27  
    quantity: 1  
  
bill-to: &id001  
  street: |  
    123 Tornado Alley  
    Suite 16  
  city: East Centerville  
  state: KS  
  
ship-to: *id001  
  
specialDelivery: >  
  Follow the Yellow Brick  
  Road to the Emerald City.  
  Pay no attention to the  
  man behind the curtain.  
...  
...
```



Exemple simple

YAML Example Code

```
1 # Employee records
2
3 - Employee one:
4   name: Alex
5   job: Developer
6   skills:
7     - python
8     - C/C++
9     - pascal
10
11 - Employee two:
12   name: Bob
13   job: Tester
14   skills:
15     - lisp
16     - fortran
17     - REST APIs
```

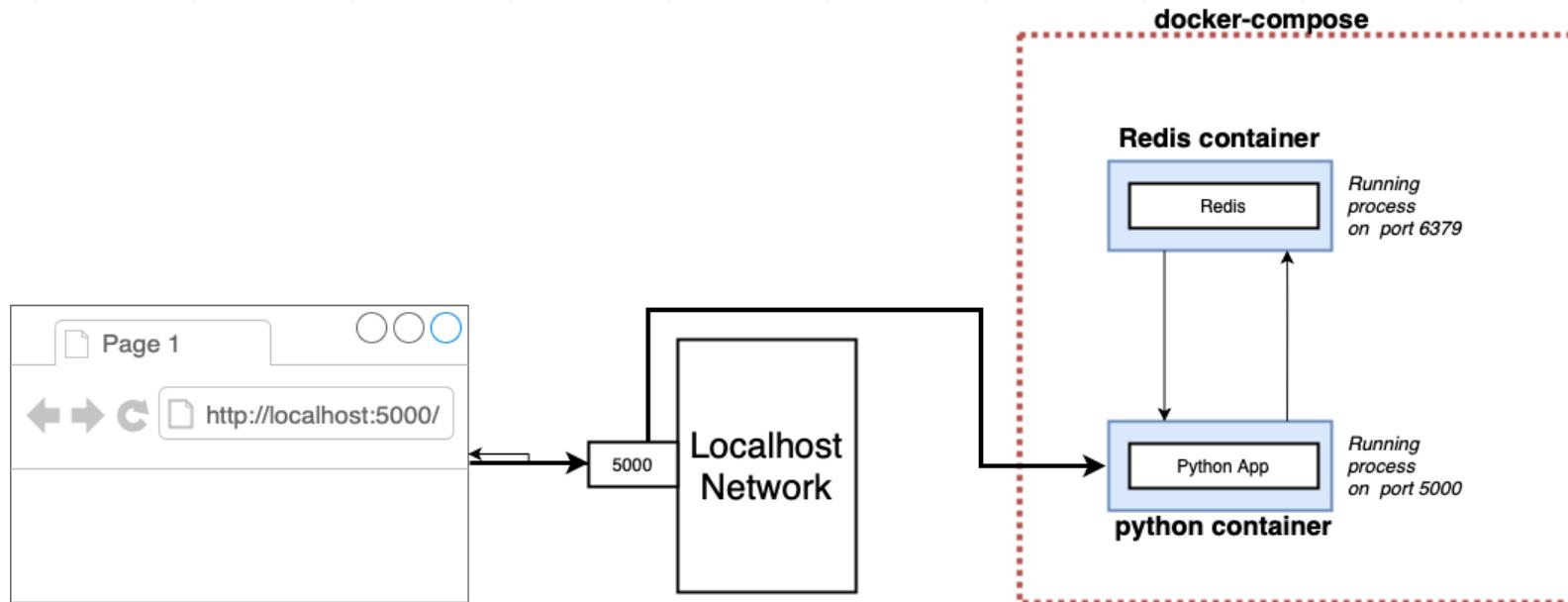
CSEstack.org

Block 1

Block 2



Notre objectif : développer une application multi containers





Architecture

- ◉ app.py : application python
- ◉ Dockerfile de l'application python
- ◉ requirements.txt fichier de dépendances

```
redis
└── app
    ├── Dockerfile
    ├── app.py
    └── requirements.txt
        docker-compose.yml
```



Fichier docker-compose.yml

- Les services sont les différents container à build & run
- Le **service web** : notre application python avec qui utilisera le Dockerfile dans ./app
- Dépend du service redis (sera en écoute)
- Pointe le répertoire local ./app

```
version: '3'
services:
  web:
    build: ./app
    ports:
      - "5000:5000"
    volumes:
      - ./app:/app
    depends_on:
      - redis-container
  redis-container:
    image: redis
```



Les bases de données relationnelles

MySQL & Docker



MySQL généralités

- MySQL est un **système de gestion de bases de données relationnelles (SGBDR)**. Il est sans doute un des SGBDR les plus populaires.
- Le SQL dans “MySQL” signifie “**Structured Query Language**” (langage standard pour les traitements de bases de données.)
- MySQL est **Open Source**. Toutefois, si vous devez intégrer MySQL dans une application commerciale, vous devez vous procurer une **licence auprès de MySQL AB**.
- Ce qui rend MySQL très intéressant est le **nombre d’API dont il dispose**. Vous pouvez en effet l’intégrer dans des applications écrites en : C, C++, Java, PHP, Python, Ruby...
- Ce qui en fait un **standard** dans le monde des bases de données.





Python et SQLAlchemy

Interaction avec les bases de données



Les bonnes pratiques

- Créer un environnement virtuel avec le strict nécessaire
- Faire des exceptions dans son code
- Faire des tests réguliers





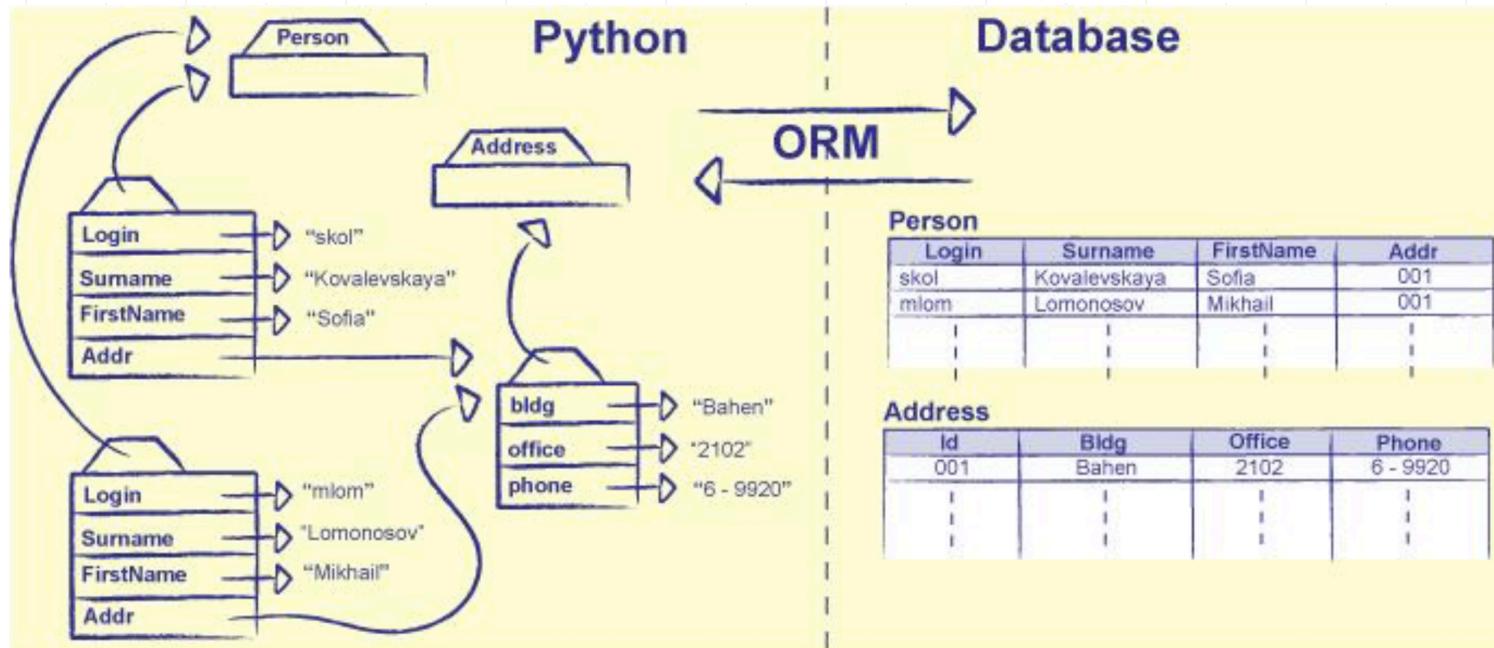
SQLAlchemy

- SQLAlchemy est un **toolkit open source SQL et un mapping objet-relationnel écrit (ORM) en Python** et publié sous licence MIT.
- Un Object-relational mapping (ORM, O/RM, and O/R mapping tool) est **une technique permettant de convertir des données entre des systèmes de « types incompatibles »** (incompatibles de base) à l'aide de langages de programmation orientés objet. Cela crée, « base de données d'objets virtuels » qui peut être utilisée à partir du langage de programmation.





ORM illustration





SQLAlchemy TYPES

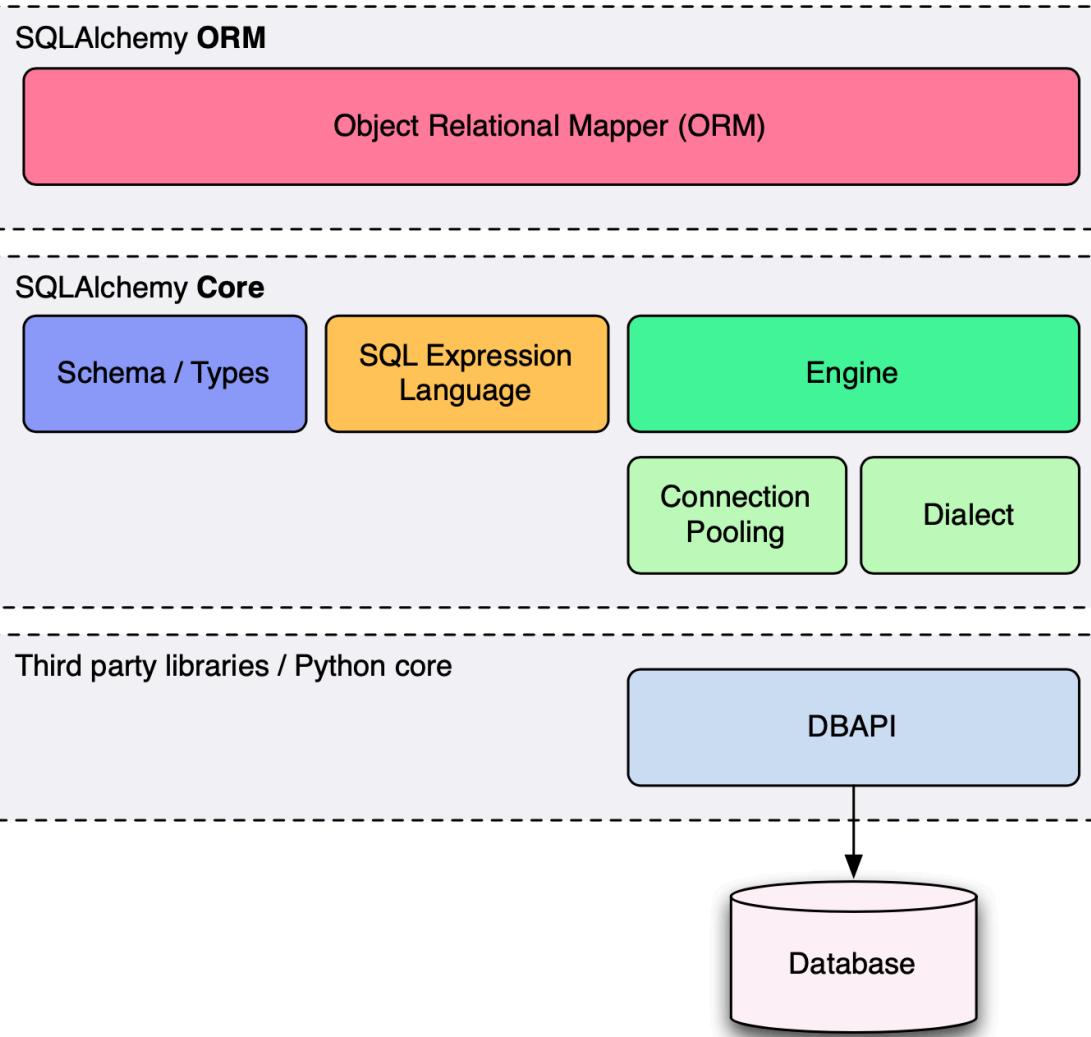
- Integer() - basic integer type, generates INT
- String() - ASCII strings, generates VARCHAR
- Unicode() - Unicode strings - generates VARCHAR, NVARCHAR depending on database
- Boolean() - generates BOOLEAN, INT, TINYINT
- DateTime() - generates DATETIME or TIMESTAMP, returns Python datetime() objects
- Float() - floating point values
- Numeric() - precision numerics using Python Decimal()





SQLAlchemy Architecture

- **Engine** - a registry which provides connectivity to a particular database server.
- **Dialect** - interprets generic SQL and database commands in terms of a specific DBAPI and database backend.
- **Connection Pool** - holds a collection of database connections in memory for fast re-use.
- **SQL Expression Language** - Allows SQL statements to be written using Python expressions
- **Schema / Types** - Uses Python objects to represent tables, columns, and datatypes.





SQLAlchemy Like ONION

Approche 1

Object Relational Mapping

Approche 2

SQL Expressions

Table Metadata, Reflection, DDL

Engine, Connection, Transactions

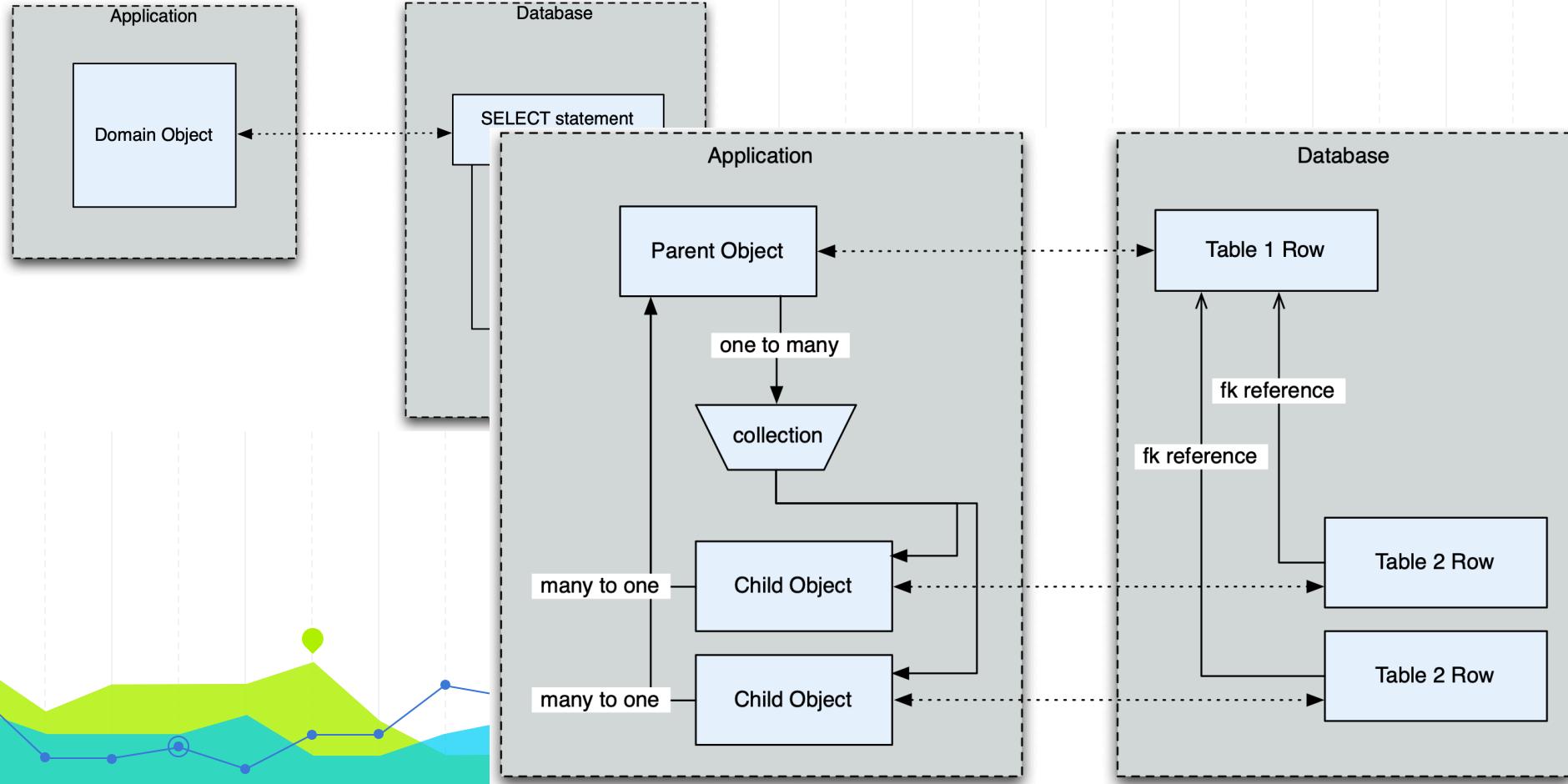




SQLAlchemy : Basic Types

- Integer() - basic integer type, generates INT
- String() - ASCII strings, generates VARCHAR
- Unicode() - Unicode strings - generates VARCHAR, NVARCHAR depending on database
- Boolean() - generates BOOLEAN, INT, TINYINT
- DateTime() - generates DATETIME or TIMESTAMP, returns Python datetime() objects
- Float() - floating point values
- Numeric() - precision numerics using Python Decimal()
- Pour utiliser les types natifs SQL vous pouvez importer **`sqlalchemy.dialects.mysql`**

D SQLAlCHEMY : Que fais l'ORM





Approche 1 : faire des classe(ORMObject)

- Fonctionne bien si on doit créer la base (approche 1).
- N'est pas recommandé quand la base est déjà existante (approche 2) car chronophage.

```
class User(ORMObject):
    tablename = 'user'

    name = String(length=50)
    fullname = String(length=100)

class Address(ORMObject):
    tablename = 'address'

    email_address = String(length=100)
    user = many_to_one("User")
```



Approche 2 : Connection à une base existante via *sqlalchemy.create_engine*

- Quand la connexion est fonctionnelle on peut mapper les tables à l'aide de *sqlalchemy.Metadata* et *sqlalchemy.Table*.
- À noter qu'il faut avant s'assurer de pull les metadata (données de la base SQL).

```
# specify database configurations
config = {
    'host': 'localhost',
    'port': 3306,
    'user': 'newuser',
    'password': 'newpassword',
    'database': 'classicmodels'
}
```

```
db_user = config.get('user')
db_pwd = config.get('password')
db_host = config.get('host')
db_port = config.get('port')
db_name = config.get('database')
```

```
connection_str = f'mysql+pymysql://{{db_user}}:{{db_pwd}}@{{db_host}}:{{db_port}}/{{db_name}}'
# connect to database
engine = create_engine(connection_str)
connection = engine.connect()
print(' - - - MySQL Docker Container Python connection ok - - - \n')
```

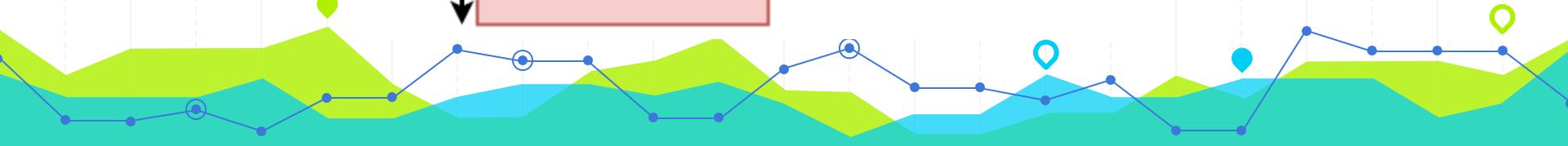
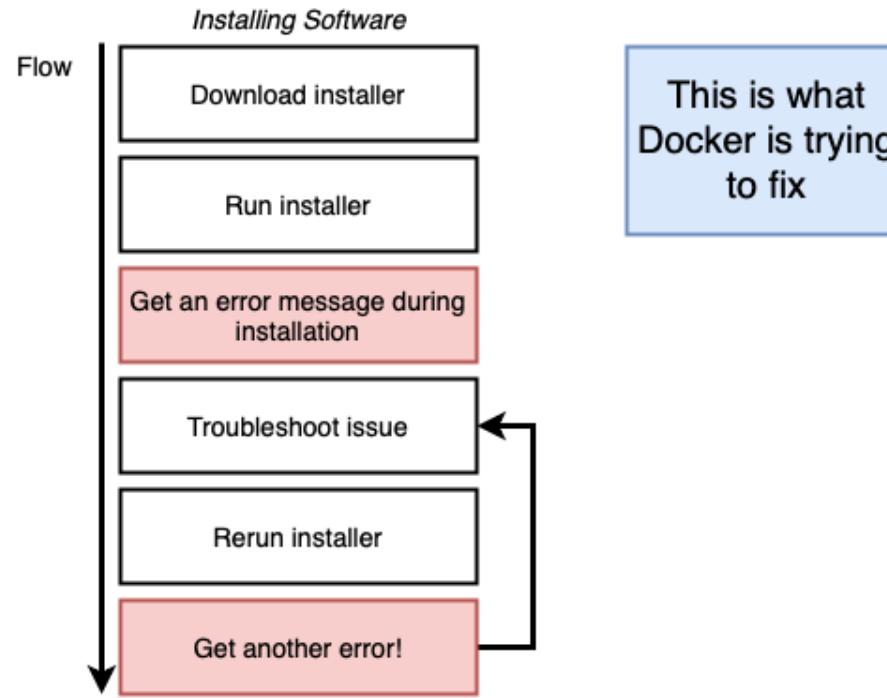


TP MySQL & Docker

Instancier une base de données existante dans un conteneur et interagir avec elle en python



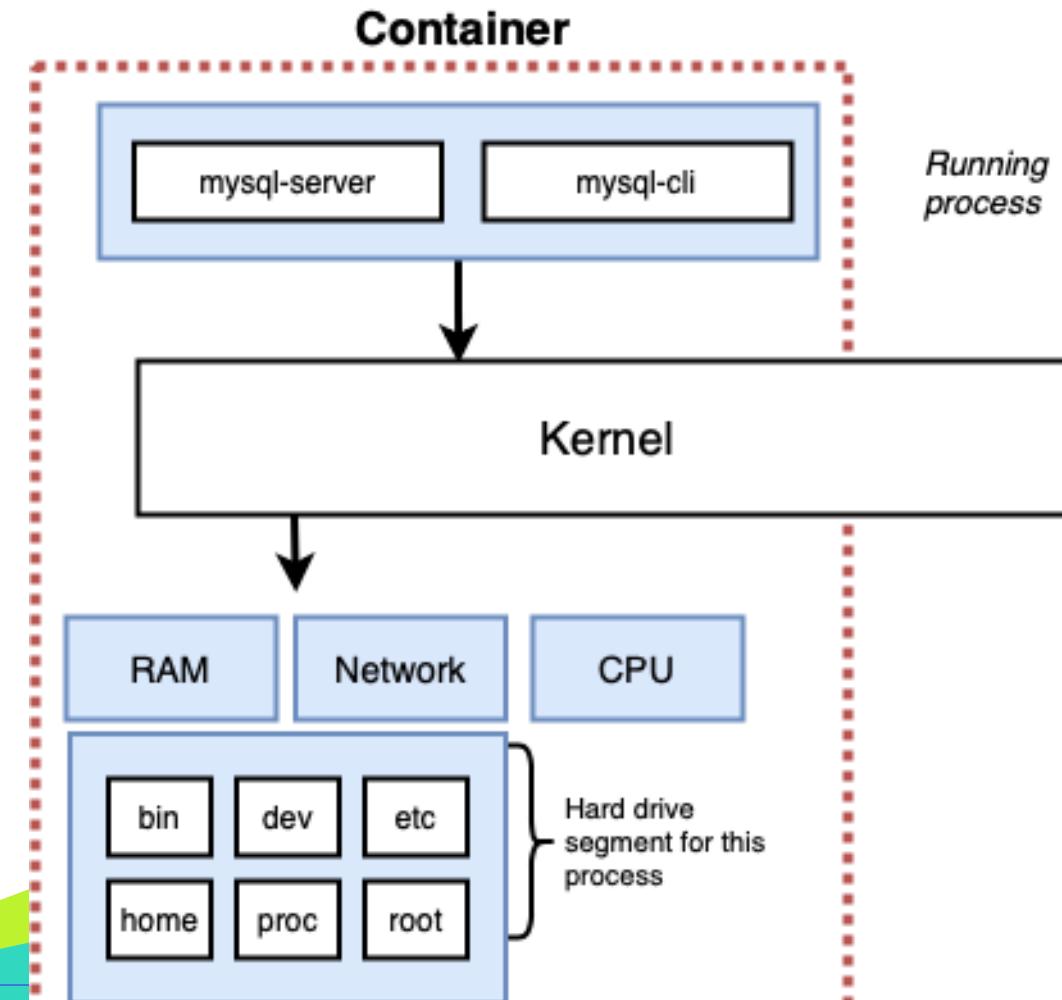
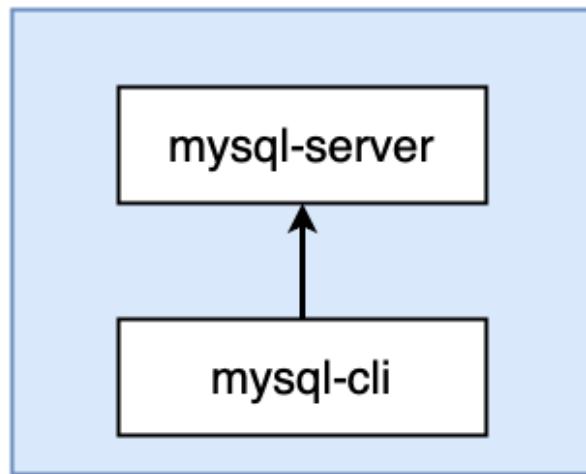
Plus d'installation/environnement complexe à mettre en place





Localhost VS Container

My Computer : localhost





Votre mission

- Set up un SQL container avec la base de donnée bdd.sql ici(<https://drive.google.com/file/d/19E1pMuH6EcdovUDACTJTPVI3a1HJTFSz/view?usp=sharing>)
- Créer un utilisateur pour votre base et afficher ses autorisations
- Écrire un script python afin de se connecter au container MySQL et vérifier les opérations CRUD standard.
- Effectuer les requêtes suivantes (voir page suivante) via le sdk python ou le CLI MySQL
- Vous ferez un push de votre travail dans un repo git dédié avec les fichiers queries.sql (dans le cas où vous passez par le CLI) ou un fichier scripts.py (si vous passez par le SDK)





Votre mission (suite)

1. Préparez une liste de bureaux triés par pays, état, ville.
2. Combien d'employés y a-t-il dans l'entreprise?
3. Quel est le total des paiements reçus?
4. Dressez la liste des lignes de produits contenant des «Voitures».
5. Déclarez le total des paiements pour le 28 octobre 2004.
6. Déclarez les paiements supérieurs à 100 000\$
7. Énumérez les produits de chaque gamme de produits.
8. Combien de produits dans chaque gamme de produits?
9. Quel est le paiement minimum reçu?
10. Répertoriez tous les paiements supérieurs à deux fois le paiement moyen.
11. Quelle est la majoration moyenne en pourcentage du PDSF sur buyPrice?
12. Combien de produits distincts ClassicModels vend-il?
13. Indiquer le nom et la ville des clients qui n'ont pas de représentants commerciaux?
14. Quels sont les noms des cadres avec VP ou Manager dans leur titre? Utilisez la fonction CONCAT pour combiner le prénom et le nom de l'employé dans un seul champ pour le reporting.
15. Quelles commandes ont une valeur supérieure à 5 000 \$?
16. Afficher le représentant du compte pour chaque client ()
17. Afficher le total des paiements pour Atelier graphique.
18. Afficher le total des paiements par date
19. Afficher les produits qui n'ont pas été vendus.
20. Afficher le montant payé par chaque client.
21. Indiquez le nombre de commandes «On Hold» (en attente) pour chaque client.
22. Listez les produits commandés un lundi.
23. Afficher tous les produits achetés par Herkku Gifts.
24. Pour les commandes contenant plus de deux produits, indiquez les produits qui constituent plus de 50% de la valeur de la commande.



Expected outputs

```
-- MySQL Docker Container Python connection ok ---  
--- docker ps info ---  
b'CONTAINER ID        IMAGE               COMMAND             CREATED          STATUS          PORTS     NAMES  
      NAMES\\nffeb1104c6f1      sqlc:01           "docker-entrypoi  
--- MySQL Database `classicmodels` connection ok ---  
--- Data Mapping OK ---  
--- Tables into database ---  
  
- offices  
- payments  
- productlines  
- employees  
- products  
- customers  
- orders  
- orderdetails  
--- Selection of PAYMENTS Table ---  
  
Type : <class 'sqlalchemy.sql.schema.Column'>  
  
Columns : ['customerNumber', 'checkNumber', 'paymentDate', 'amount']  
  
### All Payments with date > 01 June 2005:  
242 payed 12432.3200000000 on 2005-06-03  
353 payed 46656.9400000000 on 2005-06-09  
  
--- Connection Close ---
```



Quelques indices

- Pull MySQL container:
docker pull mysql:latest
- Set up la base de donnée bdd.sql via un Dockerfile :
- Show SQL Databases :
mysql>SHOW DATABASES ;
- Show SQL Users :
mysql> select host, user from mysql.user;
- Créer un user :
mysql> CREATE USER 'newuser'@'%' IDENTIFIED BY 'newpassword';
- Donner les droits à une base aux users :
mysql> GRANT ALL PRIVILEGES ON my_database.* to 'newuser'@'%';
- Show privileges
mysql> SHOW GRANTS;

```
FROM mysql

ENV MYSQL_DATABASE=mysqlsampledatabase \
    MYSQL_ROOT_PASSWORD=root_password

ADD init.sql /docker-entrypoint-initdb.d

EXPOSE 3306
```





Docker Compose

Testing automatisé



Docker-compose.yml < tests service

- Possibilité d'ajouter un service de test dans le fichier docker-compose.yml comme un service tel que :

```
tests:  
  build: ./app  
  volumes:  
    - ./db:/docker-entrypoint-initdb.d/:ro  
    - .:/app  
  command: ["pytest","./app/tests/test.py"]
```



Docker-compose.yml

3 services - exemple

- ◎ 2 services «fonctionnels»
- ◎ 1 service de test

```
version: "3"
services:
  app:
    build: ./app
    links:
      - db
    ports:
      - "5000:5000"
  db:
    image: mysql:5.7
    ports:
      - "32000:3306"
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_USER: newuser
      MYSQL_PASSWORD: newpassword
    volumes:
      - ./db:/docker-entrypoint-initdb.d/:ro
      - .:/app

  tests:
    build: ./app
    volumes:
      - ./db:/docker-entrypoint-initdb.d/:ro
      - .:/app
    command: ["pytest", "./app/tests/test.py"]
```



Docker-compose.yml file

- En entrant la commande `docker-compose up --build` vous pouvez voir le test en question (test unitaire simple situé dans le dossier `test` de l'application)

```
Successfully built 98b6ce1f5e6b
Successfully tagged sql-compose_tests:latest
Recreating sql-compose_tests_1 ...
Recreating sql-compose_tests_1 ... done
Recreating sql-compose_app_1 ... done
Attaching to sql-compose_db_1, sql-compose_tests_1, sql-compose_app_1
tests_1 | ===== test session starts =====
tests_1 | platform linux -- Python 3.6.12, pytest-6.2.2, py-1.10.0, pluggy-0.13.1
tests_1 | rootdir: /app
tests_1 | collected 1 item
tests_1 |
tests_1 | app/tests/test.py . [100%]
tests_1 |
tests_1 | ===== 1 passed in 0.42s =====
sql-compose_tests_1 exited with code 0
db_1 | 2021-02-15 23:40:56+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.33-1debian10 started.
db_1 | 2021-02-15 23:40:56+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db_1 | 2021-02-15 23:40:56+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.33-1debian10 started.
```





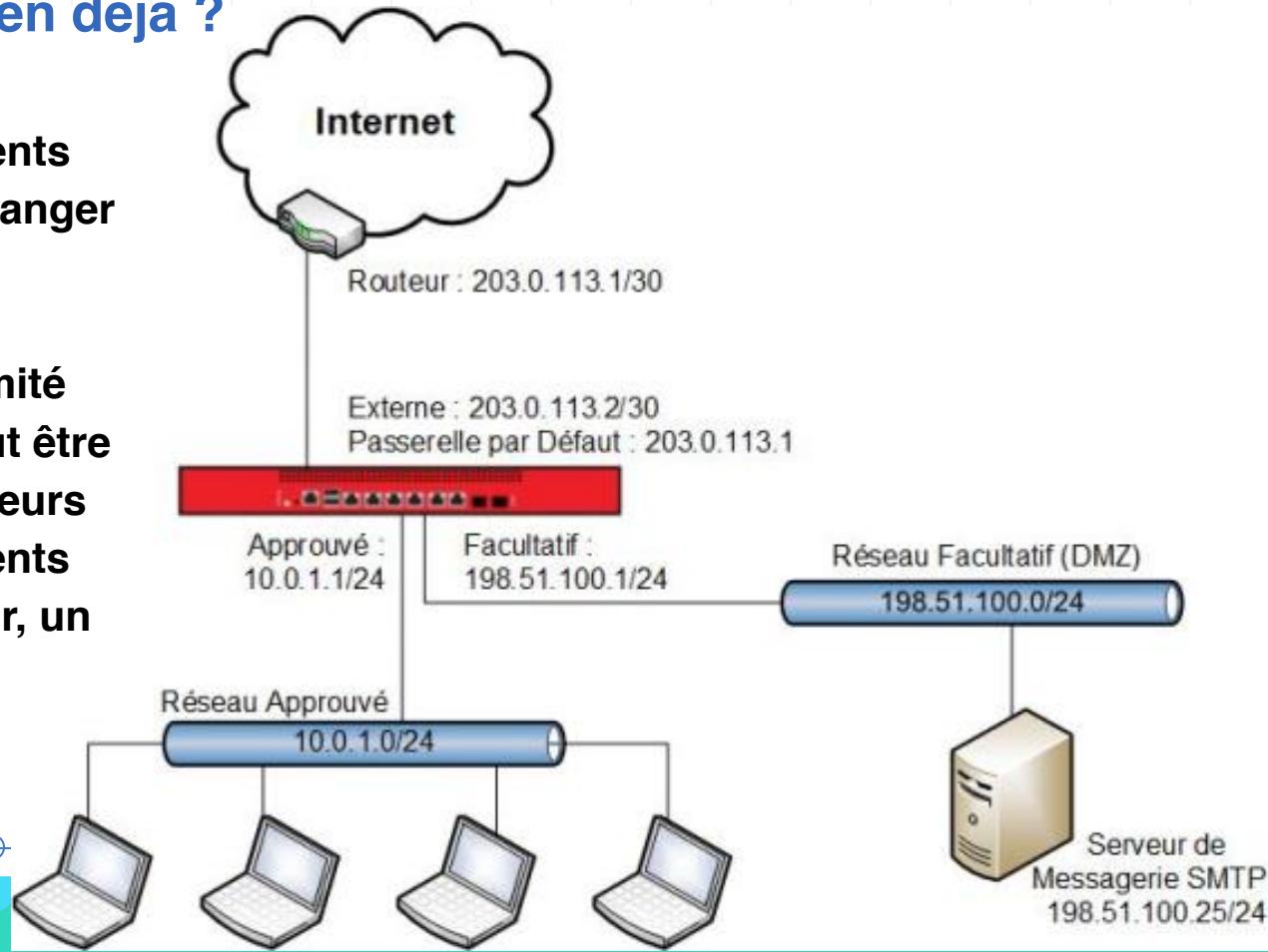
Networks

Introduction aux Networks Docker



C'est quoi un réseaux en déjà ?

- Un ensemble d'équipements reliés entre eux pour échanger des informations.
- On appelle nœud l'extrémité d'une connexion, qui peut être une intersection de plusieurs connexions ou équipements (un ordinateur, un routeur, un concentrateur, un commutateur).





Quelques notions de bases

- L'adressage IP
 - principes
 - classes et typage
- Principe élémentaire de réseau
 - Sous réseaux
 - Notation CIDR
 - masque de sous réseaux
- Architecture et éléments de langage





L'adressage IP

- L'adresse IP **identifie l'emplacement d'un hôte sur le réseau.**
- Une adresse IP doit être **unique et présenter un format normalisé.**
- Chaque adresse IP **comporte deux parties un ID de réseau et un ID d'hôte**
- Il y a **3 types de classes** d'adresse IP





L'adressage IP - Les classes d'adresses

Classe	Adresse IP	ID de réseau	ID d'hôte
A	w.x.y.z	w	x.y.z
B	w.x.y.z	w.x	y.z
C	w.x.y.z	w.x.y	z

	Nombre de réseau	Nombre d'hôtes par réseau	Plage d'ID de réseau
Classe A	126	16 777 214	1-126
Classe B	16 384	65 534	128-191
Classe C	2 097 152	254	192-223



Les masques de sous réseaux

- Chaque hôte d'un réseau TCP/IP nécessite un masque de sous-réseau.
- Un masque de sous-réseau est une adresse 32 bits utilisée pour bloquer ou « masquer » une partie de l'adresse IP afin de distinguer l'ID de réseau à partir de l'ID d'hôte.
- Cela permet à TCP/IP de déterminer si une adresse IP se trouve sur un réseau local ou un réseau distant.



Notation CIDR

- La notation ‘CIDR’ ou Classless Inter-Domain Routing donne le numéro du réseau suivi par une barre oblique (ou slash, « / ») et le nombre de bits à 1 dans la notation binaire du masque de sous-réseau.
- Le masque 255.255.255.224, du problème, équivalent en binaire à 11111111.11111111.11111111.11100000, est représenté par /27 (27 bits à la valeur 1, suivis de 5 bits 0).
- La notation 200.100.10.32/27 désigne donc l’adresse IP 200.100.10.32 avec le masque 255.255.255.224, et signifie que les 27 premiers bits de l’adresse sont dédiés à l’adresse du sous-réseau et le reste à l’adresse de la machine Hôte dans le sous-réseau.





Networks

Bridge

- Principale Network de docker (la machine qui héberge docker:docker0)
- Il permet notamment la communication entre containers
- Ici c'est l'adresse 172.17.0.1 qui est le réseau utilisé par le demon docker
- On remarque aussi le sous réseaux 172.17.0.0/16
- /!\ les IP attribué aux container ne sont pas fixes ni accessible facilement



```
→ network docker network ls
NETWORK ID          NAME      DRIVER      SCOPE
63b222658e0c        bridge    bridge      local
54cf4905dcaa        host      host       local
42ae9d56be42        none     null       local
→ network docker inspect bridge
[
  {
    "Name": "bridge",
    "Id": "63b222658e0ca13d1feb7ab7fafc51d3d7a9b4273809e910ed3e26e9c4eeef8f6",
    "Created": "2021-02-02T15:22:58.110771963Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {
      "com.docker.network.bridge.default_bridge": "true",
      "com.docker.network.bridge.enable_icc": "true",
      "com.docker.network.bridge.enable_ip_masquerade": "true",
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  }
]
```



Docker networks create - Les networks personnalisés

- La commande docker network permet de :
 - Créer un nouveau réseau en personnalisant le Bridge
 - Créer un réseau de type overlay afin de faciliter la communication inter cluster
- Le réseau none = aucune couverture réseaux (rare)
- Le réseau host = uniquement le host (la machine qui héberge docker)

```
[→ network docker network create -d bridge --subnet 172.40.0.0/16 my-network
7c0f602f0904448fbce0c353758074be9c47e96a10a3c6025b18bf87a0f4ae8f
[→ network docker network ls
NETWORK ID          NAME            DRIVER      SCOPE
63b222658e0c        bridge           bridge      local
54cf4905dcaa        host             host       local
7c0f602f0904        my-network       bridge      local
42ae9d56be42        none            null       local
```



Utiliser le Bridge

Exemple redis app

- Via le fichier docker-compose.yml on peut utiliser la configuration du bridge pour créer son propre sous réseaux et attacher des services
- On peut aussi aller beaucoup plus loin dans les configuration (ipv6, proxy ...)

```
version: '3.3'
services:
  web:
    build: ./app
    ports:
      - "5000:5000"
    volumes:
      - ./app:/app
    depends_on:
      - redis-container
  networks:
    - front
    - back

  redis-container:
    image: redis
    networks:
      - back

networks:
  front:
    driver: bridge
    ipam:
      driver: default
      config:
        - subnet: 172.16.238.0/24
  back:
```



Aller plus loin avec les réseaux

- On peut ajouter des machines sur le réseau qui ne sont pas des container avec l'option (`--add-host`) —> /etc/hosts du container
- On peut aussi ajouter des IP de serveurs DNS aux réseaux

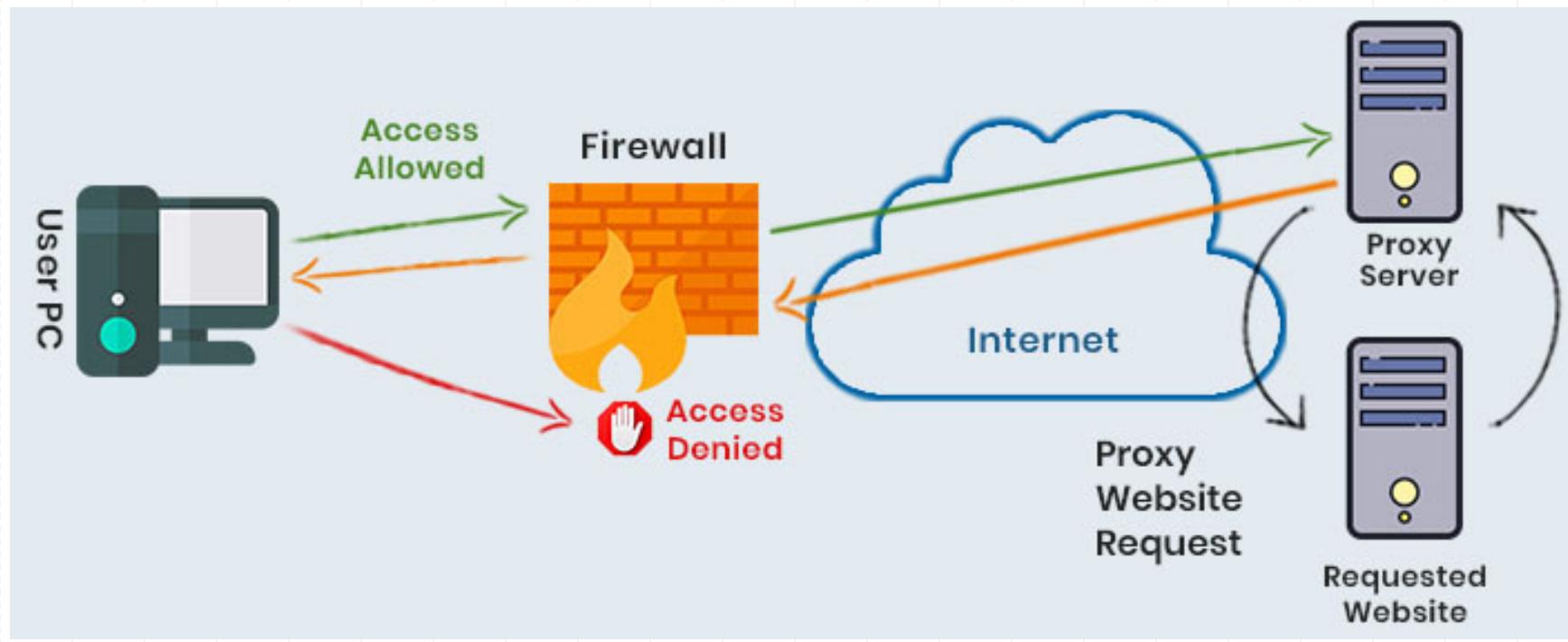


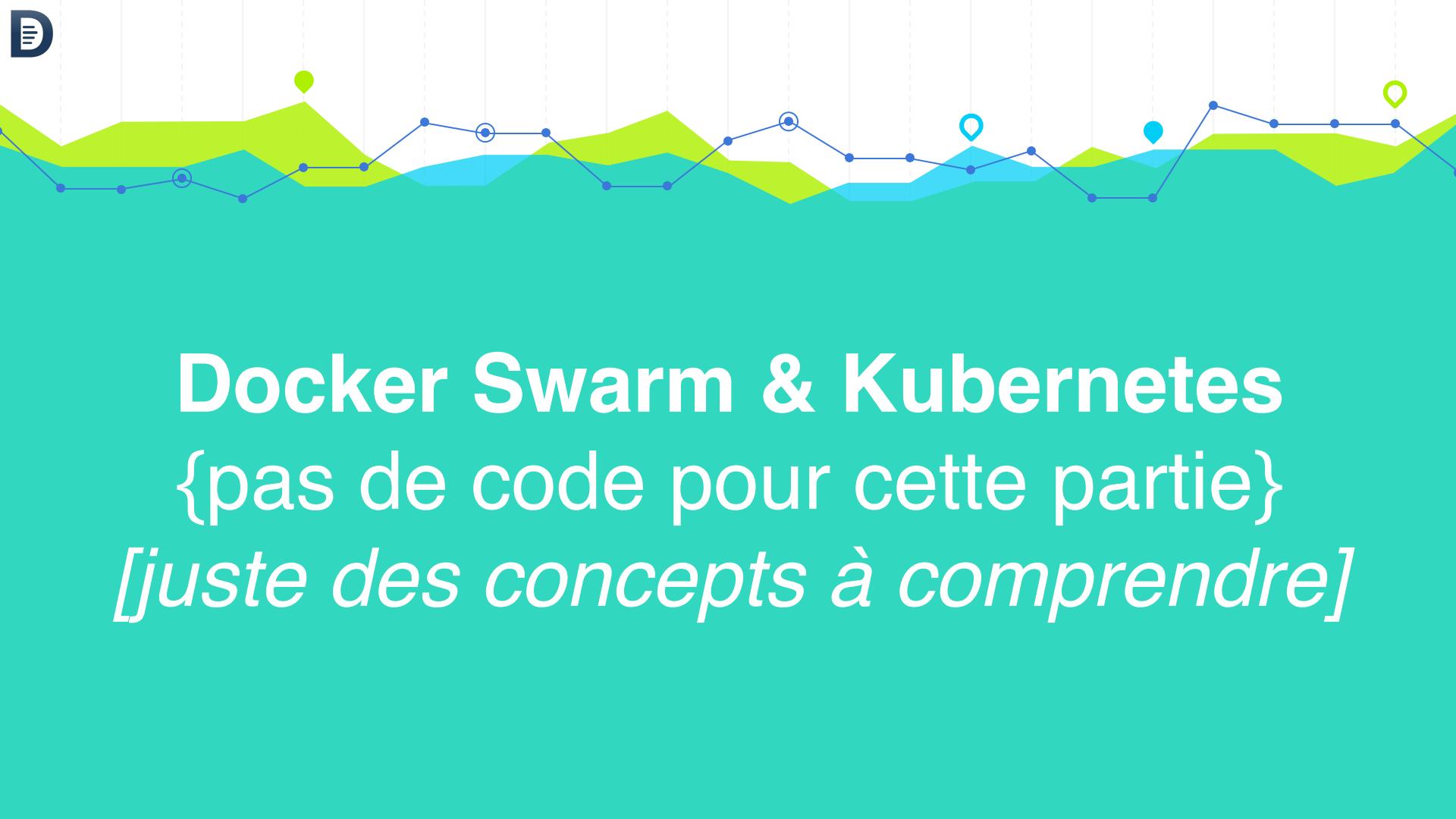
Les concepts de FireWall & Proxy

- Un **Firewall bloque** les connexions tandis qu'un Serveur **Proxy facilite** les connexions
- Un **Serveur Proxy peut également servir de Firewall**
- Les Firewalls existent souvent en tant que interface entre un réseau public et privé, alors que des Serveurs Proxy peuvent également exister avec des réseaux publics des deux côtés.
- Un **Firewall est utilisé pour protéger un réseau interne** contre les attaques tandis qu'un **Serveur Proxy est utilisé pour l'anonymat et pour contourner les restrictions**



FireWall VS Proxy





Docker Swarm & Kubernetes

{pas de code pour cette partie}

[juste des concepts à comprendre]



L'orchestration de containers

- Dans les parties précédentes, nous nous sommes concentrés sur la mise en oeuvre d'architectures mono hôte (notre localhost)
- Le sujet de l'orchestration c'est donc de savoir gérer des containers/services multi hôte.
- On va donc s'intéresser à 2 outils d'orchestration qui sont docker swarm et kubernetes

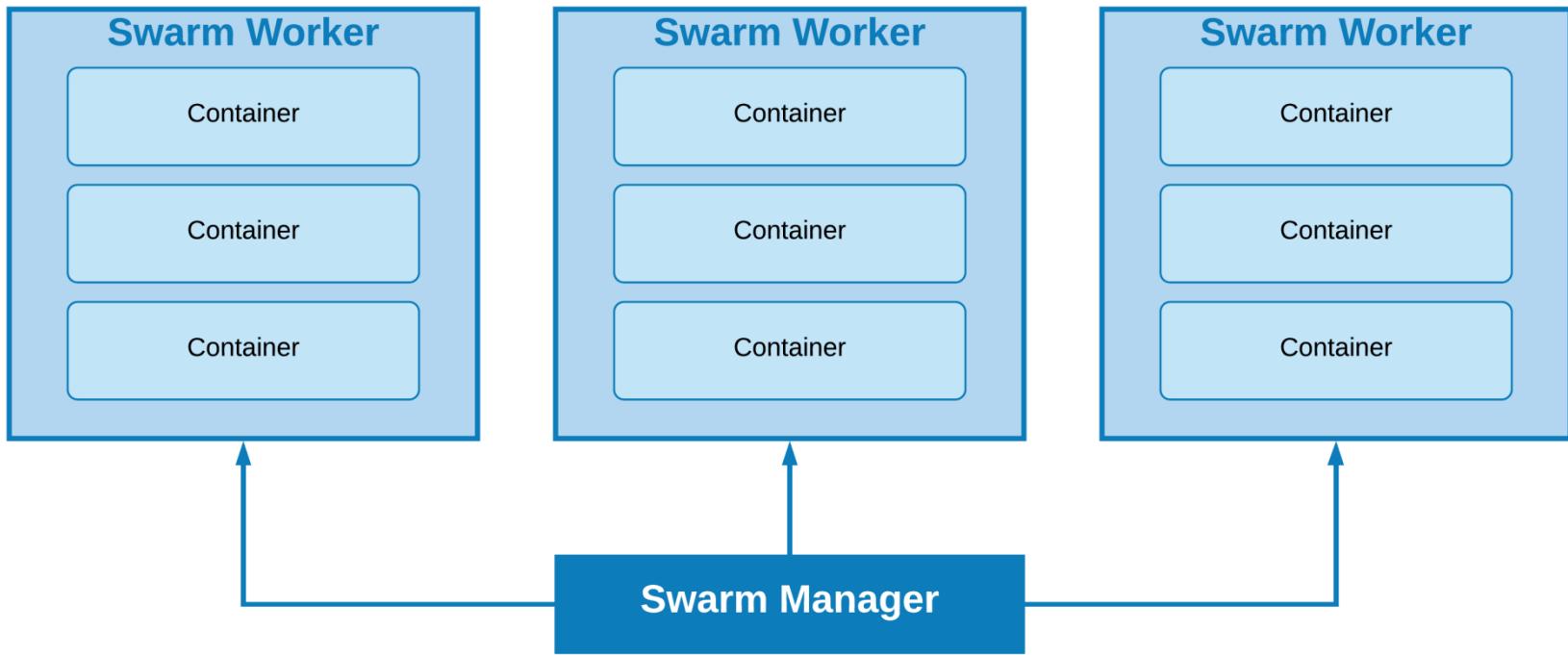


Docker Swarm

- Pour assurer un **haute productivité de votre application => plusieurs host (noeuds)**
- **Swarm permet de gérer centralisent un ensemble d'hôte (disposant d'une instance de docker-engine comme un hôte unique).** Pour cela il s'appuie notamment sur :
 - **Du client Docker** : il sera utilisé aussi bien pour gérer nos conteneurs ainsi que Swarm
 - **Un noeud maître Swarm (Swarm Master)** et si besoin, dans le cas d'une architecture hautement disponible, un ou plusieurs répliques de ce noeud maître qui en tant que contrôleur, gère les différents noeuds du cluster et orchestre le déploiement des conteneurs.
 - **Plusieurs noeuds Swarm (Swarm Node)**, qui sont simplement des hôtes sur lesquels nos conteneurs vont s'exécuter.



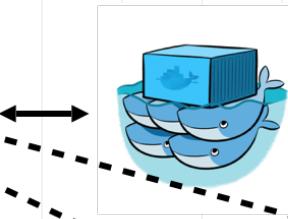
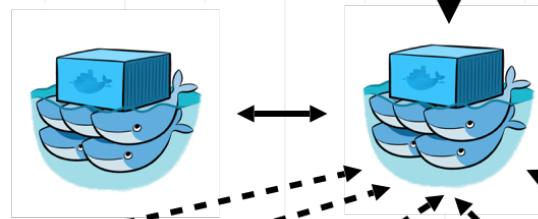
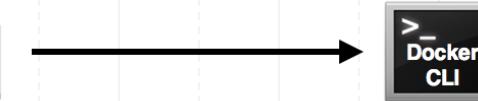
Docker Swarm en image



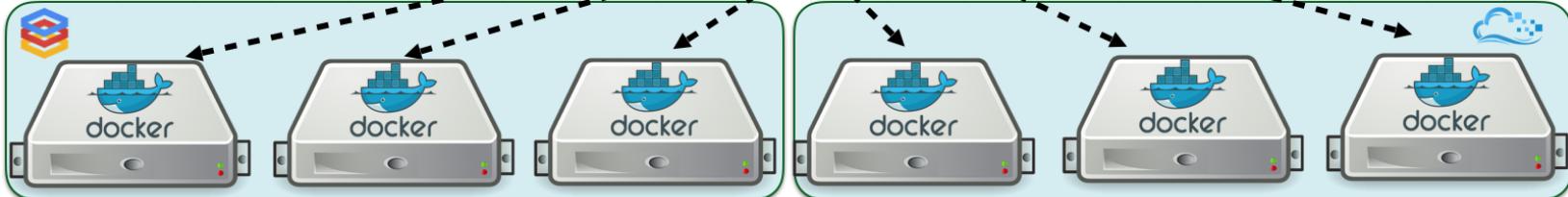


Compose

.yml Description



Swarm



Cluster Managers

CM1

CM2





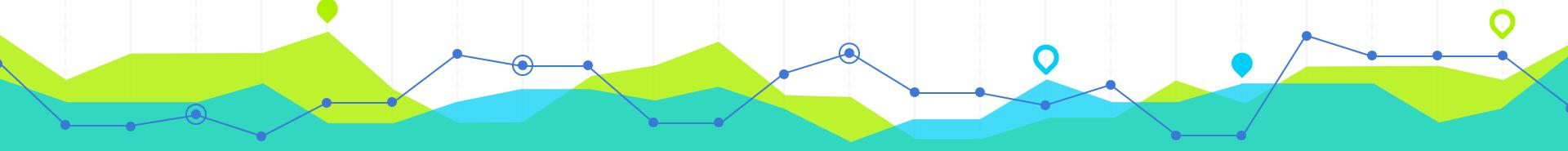
Docker Swarm & Réseaux overlay

- L'avantage d'utiliser docker swarm est qu'il **intègre naturellement la communication entre host via un réseau spécifique : le overlay** (non plus le bridge)
- Nous n'entrerons pas en détail dans ce cours sur la configuration/paramétrage de ce réseau.
- Ce qu'il faut savoir c'est que c'est un **outil fiable et pratique**



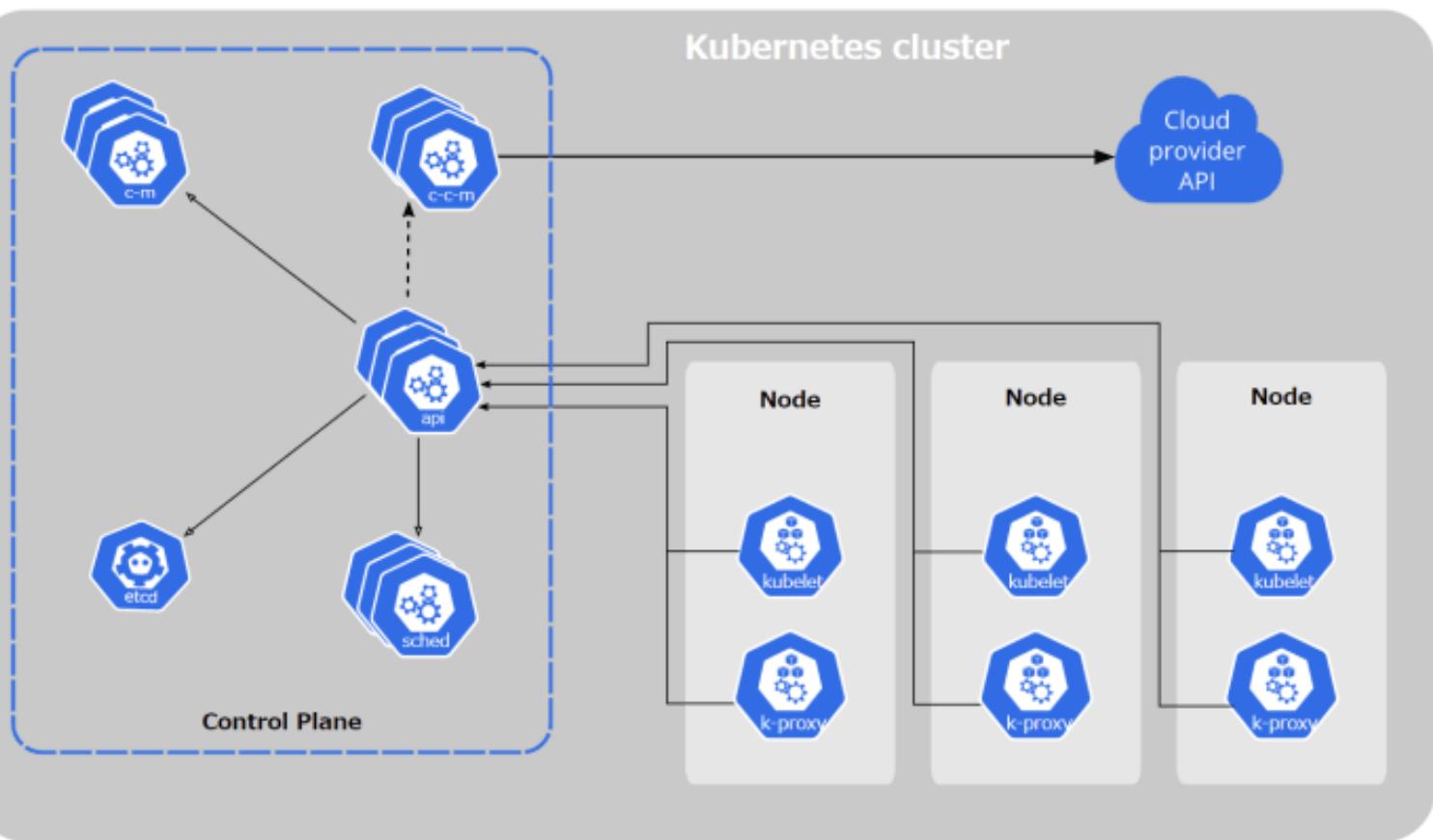
Kubernetes

- Comme Swarm c'est **un orchestrateur** et sa **CLI** est **kubectl**
- Il dispose d'un outils assez pratique : **minikube** qui permet de monter un cluster à **un seul noeud**.
- **Ses principaux composant sont :**
 - **Contrôleur** : est en charge de scruter l'état de santé des services déployés dans le cluster.
 - **Scheduler**: est en charge du démarrage des conteneurs, qui sont groupés en pods
 - **etcd**: est un stockage répliqué de haute disponibilité qui assure la persistance de la configuration du cluster.





Kubernetes Cluster

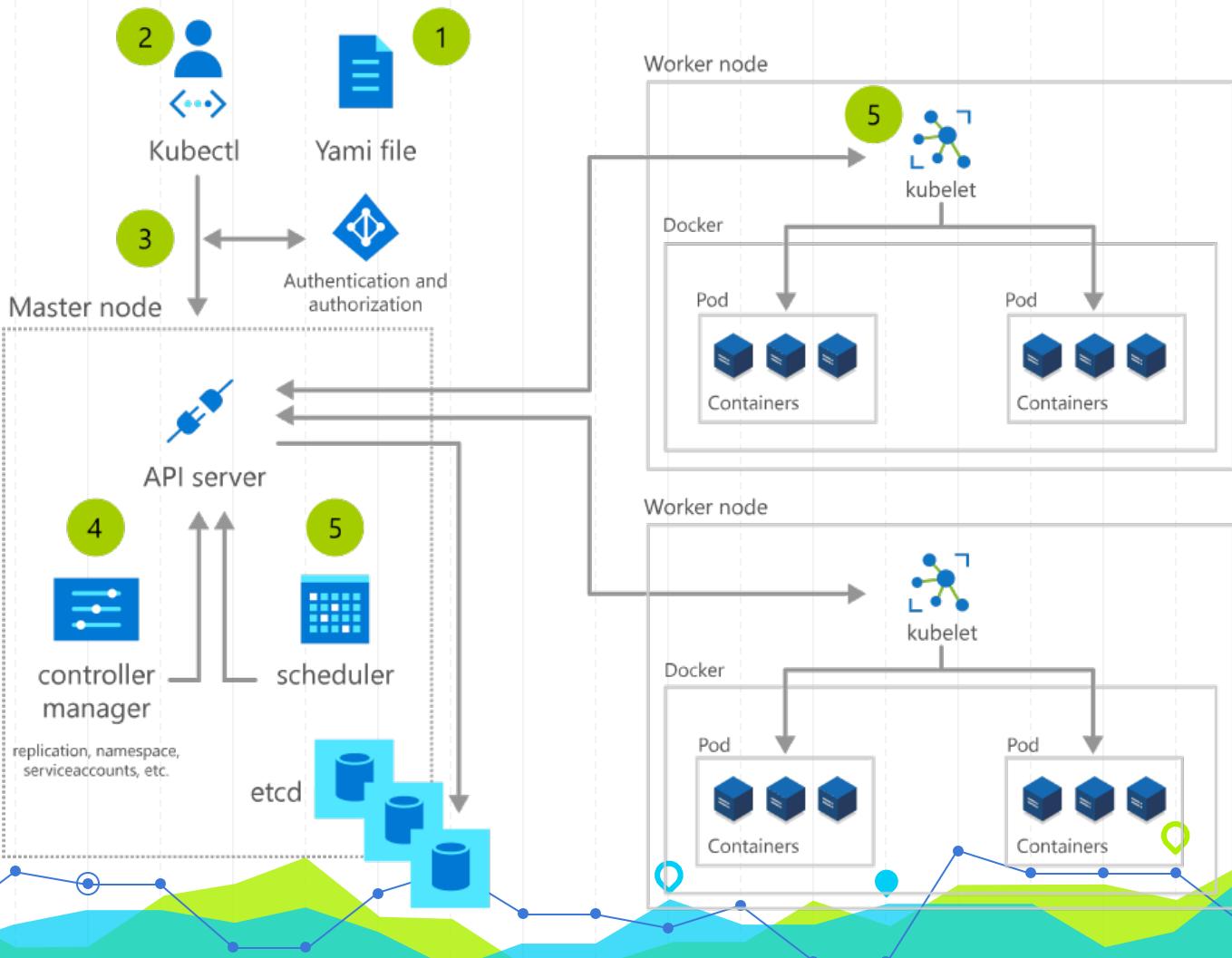


API server	
Cloud controller manager (optional)	
Controller manager	
etcd (persistence store)	
kubelet	
kube-proxy	
Scheduler	
Control plane	
Node	



Kubernetes

Deployment Workflow





Les bases de données non relationnelles

Elasticsearch & Docker



Le NoSQL vs SQL

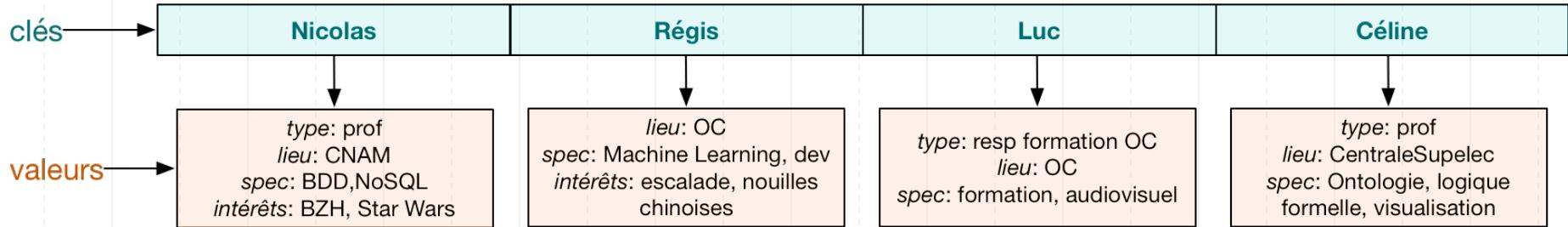
- Cette approche propose de **relâcher certaines contraintes lourdes du relationnel pour favoriser la distribution** (structure des données, langage d'interrogation ou la cohérence).
- Le NoSQL est à la fois une autre **manière d'interroger les données, mais aussi de les stocker**.
- Différentes familles de bases NoSQL existent : **Clé/Valeur, colonnes, documents, graphes**. Chacune de ces familles répond à des besoins très spécifiques





Le NoSQL I approche Key:Value

- Le but de la famille clé-valeur est l'efficacité et la simplicité. Tout repose sur le couple Clé/Valeur. La clé identifie la donnée de manière unique et permet de la gérer. La valeur contient n'importe quel type de données.





Le NoSQL I approche colonne

- Traditionnellement, les données sont représentées en ligne.
- Le stockage orienté colonne **se focalise sur chaque attribut et en les distribuant**. Il est alors possible de **focaliser les requêtes sur une ou plusieurs colonnes, sans avoir à traiter les informations inutiles**.

Stockage orienté lignes

id	type	lieu	spec	intérêts
Nicolas	prof	CNAM	BDD, NoSQL	BZH, Star Wars
Régis		OC	Machine Learning, Dev	escalade, nouilles chinoises
Luc	resp formation OC	OC	formation, audiovisuel	
Céline	prof	CentraleSupelec	Ontologie, logique formelle, visualisation	

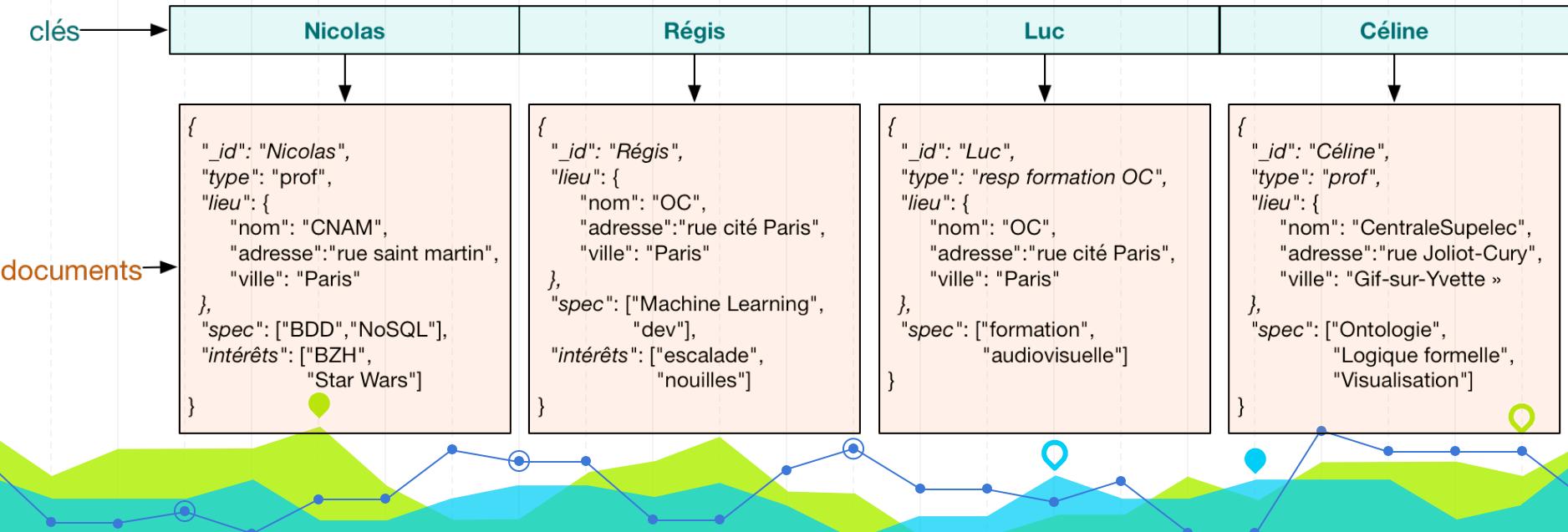
Stockage orienté colonnes

id	type	id	lieu	id	spec	id	intérêts
Nicolas	prof	Céline	Centrale Supelec	Nicolas	BDD	Nicolas	BZH
Céline	prof	Nicolas	CNAM	Nicolas	NoSQL	Nicolas	Star Wars
		Régis	OC	Régis	Machine Learning	Régis	escalade
		Luc	OC	Luc	Dev	Luc	nouilles chinoises
					formation		
					audiovisuel		
					Ontologie		
					logique formelle		
					visualisation		



Le NoSQL I approche documents

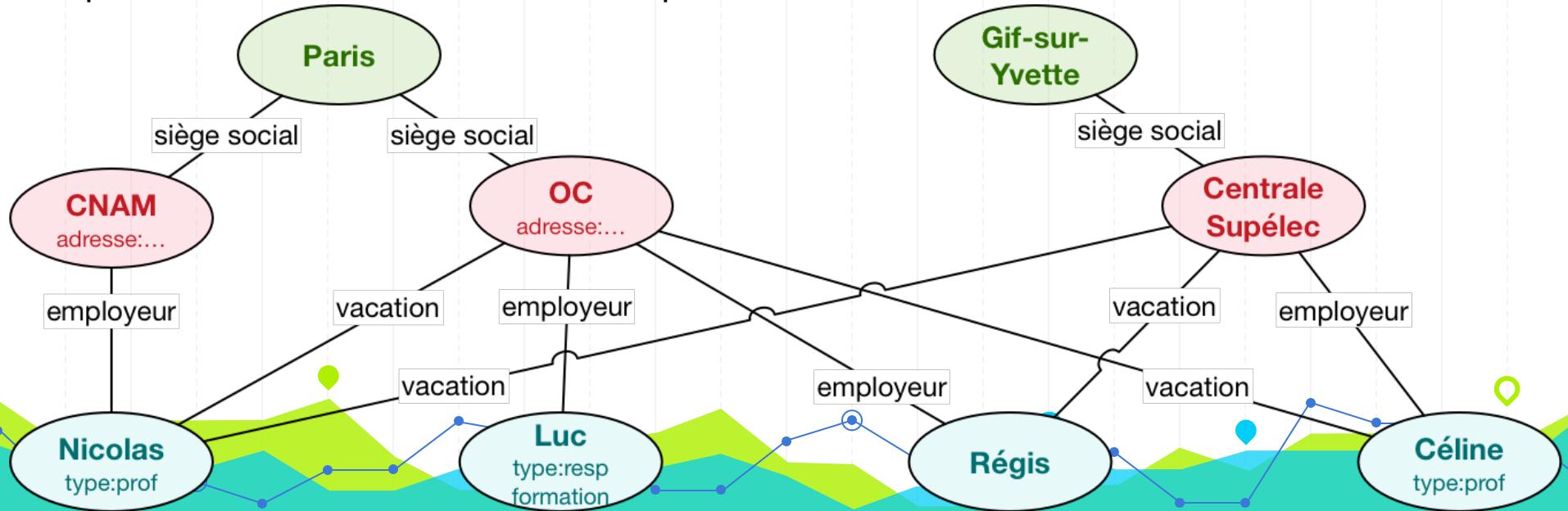
- Les bases orientées documents sont **souvent utilisés pour des requêtes complexes**.
- Le but de ce stockage est de **manipuler des documents contenant des informations avec une structure complexe** (types, listes, imbrications). Il repose sur le **principe du clé/valeur, mais avec une extension sur les champs qui composent ce document**.





Le NoSQL I approche graphe

- Les trois premières familles NoSQL n'adressent pas le problème de corrélations entre les éléments.
- Prenons l'exemple d'un réseau social : dans certains cas, il devient très complexe de calculer la distance entre deux personnes non directement connectées. Et c'est ce type d'approche que résolvent les bases orientées Graphe.





Tous les chemins mènent à la base

- Comme vous devez vous en doutez, il n'existe pas qu'un outils pour traiter ce genre de données.
- Dans ce cours nous allons voir uniquement **Elasticsearch**.





Elasticsearch en 2 mots

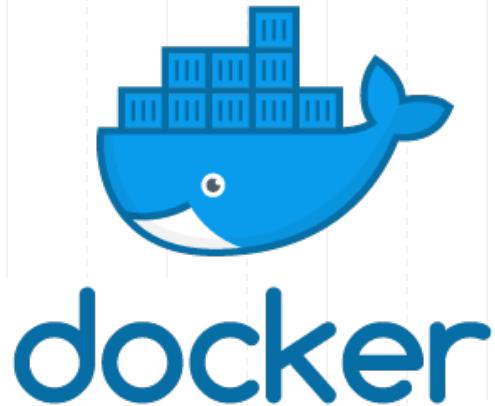
- Elasticsearch est **un outil de recherche distribué en temps réel et un outil d'analyse**.
Il est utilisé notamment pour les besoins suivants :
 - recherche full text
 - recherche structurée
 - analyse (texte, logs, ...)
 - et les trois combinés
 - **Qui l'utilise :**
 - Wikipedia
 - StackOverflow
 - GitHub
 - Goldman Sachs
- 



Notre environnement de travail pour ce cours



elasticsearch



kibana





Inside the box (1/2)

- Elasticsearch a besoin de :
 - Apache Lucene™, un moteur de recherche full-text
 - Java et donc la JVM
- **Quels intérêts :**
 - **un stockage de document temps réel distribué** où tous les champs sont indexés et consultable
 - **un moteur de recherche distribué** avec de l'analyse temps réel
 - **capable de supporter la monté en charge avec une centaine de servers et des peta-octets de données structurées ou non**





Inside the box (2/2)

- Le stockage orienté document, c'est donc un vocabulaire particulier.
 - Quelques comparaisons avec une base de donnée relationnelle

Relational DB	Base de données	Tables	Lignes	Colonnes
Mongo DB	Base de données	Collections	Documents	Champs
Elasticsearch	Index	Types	Documents	Champs

- Communication avec Elasticsearch :
 - Java API sur le port 9300
 - Restful API sur le port 9200
- *Dans ce cours nous traiterons uniquement que de l'API Rest*





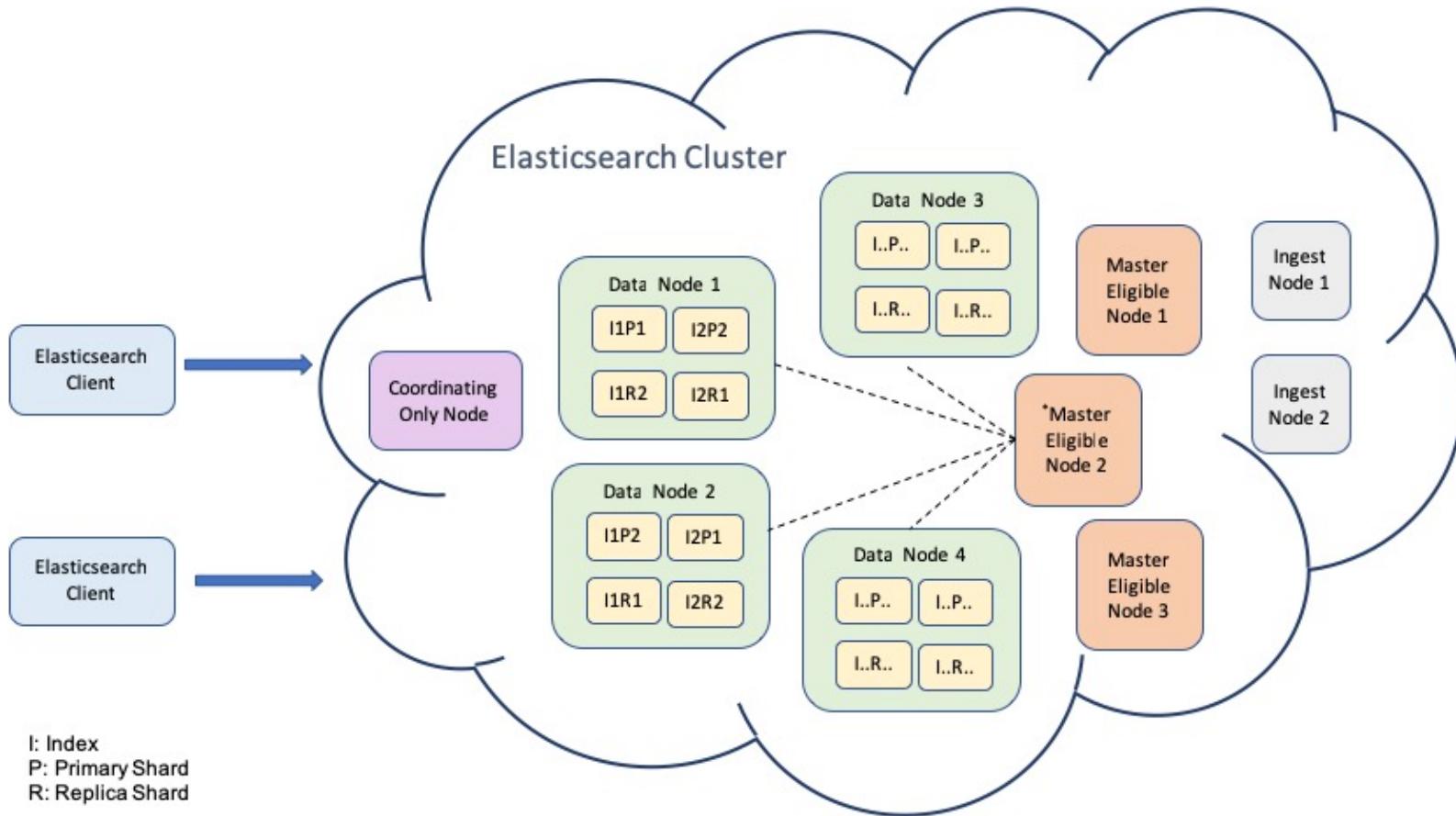
Element d'architecture / overview

- **Cluster : ensemble de serveurs (noeuds ayant elasticsearch / même id de cluster)**
 - communication via port 9300
 - haute dispo : perfomance et redondance
- **Noeuds : serveur ayant un service elastic**
 - différents type : master, data, client et ingest
- **Index : une instance de base de données**
 - un à plusieurs par serveur (et cluster)

- **Shards: découpage logique d'un index (un à plusieurs shards)**
 - répartition des shards sur plusieurs noeuds (répartir un index)
 - joue sur les performances
 - ajout d'un noeud = réaffectation des shards
 - important pour déterminer nb max de noeuds
- **Rélicas : rélicas de shards d'un index**
 - redondance
 - mais aussi performance (interrogeables)
 - communication via port 9300



Element d'architecture | overview





Element d'architecture I indexes

- **Type (table) : regroupement de documents**
 - structure non fixe —> évolutive
 - on peut la définir par un mapping
- **Mapping : définition de les propriétés du type**
 - par exemple : les noms des champs, leur caractéristiques ...
- **Document : élément le plus fin (équivaut à une ligne en sql)**
 - un objet d'un type
 - composé des propriétés du type





Structure de données : le documents

- Les documents sont la **structure principale de stockage** dans elasticsearch. Chaque document contient des champs réservés (**les métadonnées**) du document, comme par exemple :
 - _index** : où réside le document
 - _type** : type que le document représente (bdd)
 - _id** : identificateur unique du document
 - _source** : les données sous forme de dictionnaire

The diagram illustrates the mapping between source documents and their indexed representation in Elasticsearch. On the left, three source documents are defined as dictionaries:

```
doc1 = {  
    "city": "New Delhi",  
    "country": "India"  
}  
doc2 = {  
    "city": "London",  
    "country": "England"  
}  
doc3 = {  
    "city": "Los Angeles",  
    "country": "USA"  
}
```

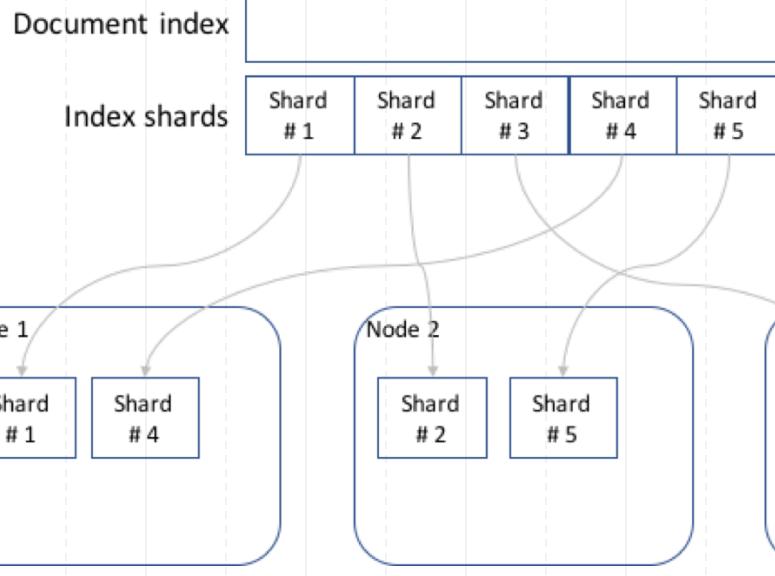
An arrow points from the `_source` field of the first document to its corresponding indexed representation on the right. The indexed representation is a dictionary with additional metadata fields:

```
{  
    "_index": "cities",  
    "_type": "places",  
    "_id": "1",  
    "_version": 9,  
    "_seq_no": 20,  
    "_primary_term": 2,  
    "found": true,  
    "_source": {  
        "city": "New Delhi",  
        "country": "India"  
    }  
}
```



Diviser pour mieux régner

Le concept de SHARDS



Console Search Profiler Grok Debugger

Nb of shard into this nodes

```
1 GET _cluster/health
2
3 GET _cat/nodes?v
4 #bla
5 GET _cat/indices?▶
```

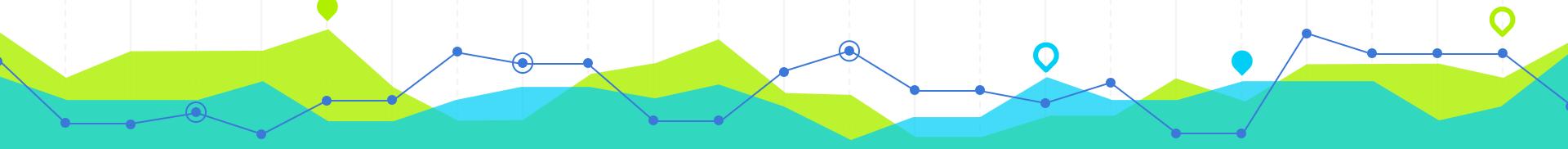
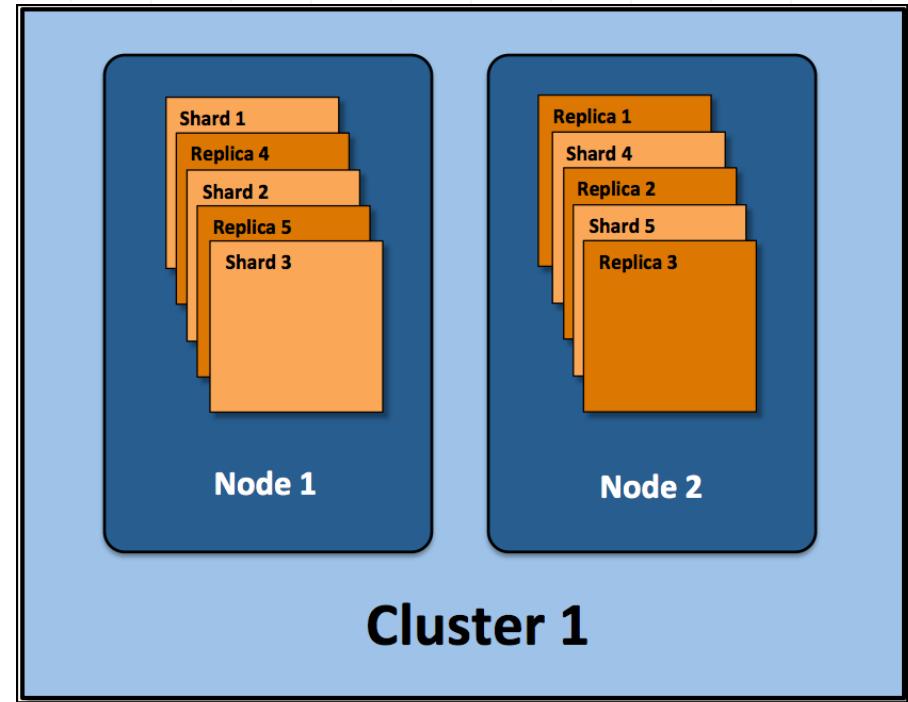
	1	health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
2	2	yellow	open	megacorp	l2v7L_UkTue7LDZ6uceVTg	1	1	1	0	5.7kb	5.7kb
3	3	yellow	open	cities	jgeZDaKXQv2vBhjVovnvoQ	1	1	3	0	4.3kb	4.3kb
4	4	green	open	.kibana_task_manager_1	4oBe51AQSJG0JIxvk1h01Q	1	0	2	0	46kb	46kb
5	5	green	open	.apm-agent-configuration	HJY05P5sQ9uR1NhEnVGWw	1	0	0	0	283b	283b
6	6	green	open	.kibana_1	aYcdXn5WThuYR-ida-pcYw	1	0	3	0	14.7kb	14.7kb
7	7	yellow	open	customer	m7hpWrpcTx-6lFi7dQhI2Q	1	1	2	0	6.7kb	6.7kb
8	8										





Nodes, Shards & Replica

- En plus de subdiviser votre index en plusieurs shards (fragments). Votre node crée des replica (sauvegarde) de chaque shard.
- Chaque replica est stocké sur un noeud différents du shard dont il est originaire.
- Il peut donc y avoir une pane de disque sur le node1 , les données ne seront pas perdu car elle seront sur le node2.





Le vocabulaire HTTP - overview

GET

- Retrieves a resource
- Guaranteed not to cause side-effect (SAFE)
- Cacheable

POST

- Creates a new resource
- Unsafe, effect of this verb isn't defined by HTTP

PUT

- Updates an existing resource
- Used for resource creation when client knows URI
- Can call N times, same thing will always happen (idempotent)

DELETE

- Removes a resource
- Can call N times, same thing will always happen (idempotent)





HTTP & Elasticsearch

- PUT : création ou modification d'un document
- GET : récupération d'un document
- HEAD : test si un document existe
- DELETE : suppression d'un document
- Retourne
 - **un code de retour HTTP (200, 404, etc.)**
 - **une réponse encodé en JSON (sauf pour les requêtes HEAD)**

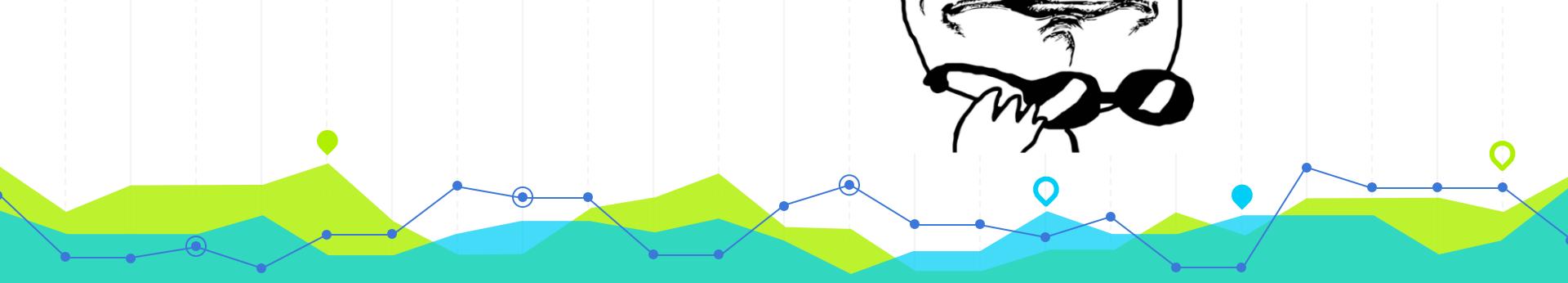
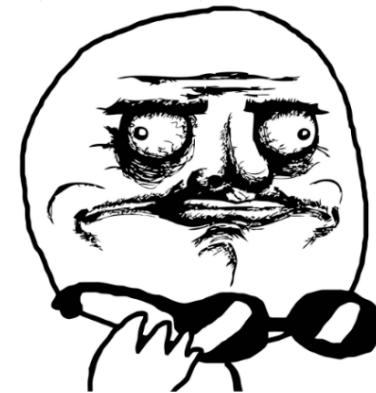




Search SCORE

- Pour **chaque recherche**, elastic attribue un score, on peut voir ça comme **un score de pertinence**
- Ce score est **calculé par rapport à la méthode TF-IDF** (de l'anglais term frequency-inverse document frequency).

$\text{score}(q,d) =$

$$\text{queryNorm}(q) * \text{coord}(q,d) * \text{SUM} (\text{tf}(t \text{ in } d), \text{idf}(t)^2, \text{t.getBoost()}, \text{norm}(t,d)) \text{ (t in q)}$$




Term Frequency-Inverse Document Frequency - TF IDF Algorithm

- C'est une méthode de pondération (statistique) souvent utilisée en recherche d'information et en particulier dans la fouille de textes.
- Cette statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus (l'ensemble de document).
- Il s'agit donc de calculer le poids d'un terme (ou d'un document)
- Ce poids (de mots/ termes) augmente proportionnellement au nombre d'occurrences du mot dans le document.



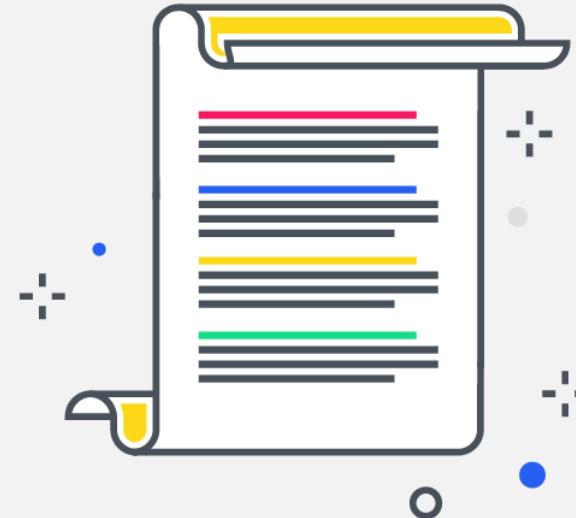


TF IDF Algorithm en image

Frequency of a term across multiple documents



Frequency of a term across a single document





TF-IDF Algorithm : la formule

$$w_{x,y} = tf_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

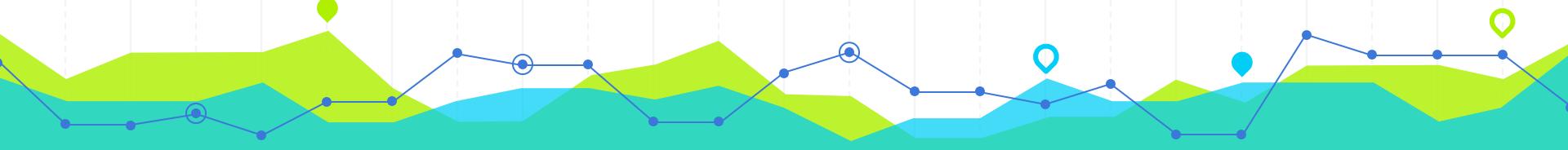




TF IDF Algorithm : plus de formules

Les différentes version du TF-IDF

Name	Value
Unary	1
Inverse Document Frequency	$\log \frac{N}{n_t}$
Inverse Document Frequency Smooth	$\log(1 + \frac{N}{n_t})$
Inverse Document Frequency Max	$\log(1 + \frac{\max\{t' \in d\} n_t}{n_t})$
Probabilistic Inverse Document Frequency	$\log \frac{N - n_t}{n_t}$





En pratique

Installer son environnement



Installation avec docker

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html>

```
[→ elasticsearch docker pull docker.elastic.co/elasticsearch/elasticsearch:7.11.1
7.11.1: Pulling from elasticsearch/elasticsearch
ddf49b9115d7: Pull complete
60135e09cc9c: Pull complete
81ec2635a090: Pull complete
70345223b04b: Pull complete
9cddfc9583aa: Pull complete
29d9cd80cb8a: Pull complete
6dbbe9fd73c5: Pull complete
Digest: sha256:bff764c6dcc1241288698722a83e72d54fbba336fd447c5a78f23c5cf159435b
Status: Downloaded newer image for docker.elastic.co/elasticsearch/elasticsearch:7.11.1
docker.elastic.co/elasticsearch/elasticsearch:7.11.1
[→ elasticsearch docker run -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.11.1
{"type": "server", "timestamp": "2021-02-21T17:36:17,689Z", "level": "INFO", "component": "o.e.n.Node", "cluster.name": "docker-cluster", "node.name": "a06e897599f9", "message": "version[7.11.1], pid[6], build[default/docker/ff17057114c2199c9c1bbecc727003a907c0db7a/2021-02-15T13:44:09.394032Z], OS[Linux/4.19.76-linuxkit/amd64], JVM[AdoptOpenJDK/OpenJDK 64-Bit Server VM/15.0.1/15.0.1+9]" }
{"type": "server", "timestamp": "2021-02-21T17:36:17,697Z", "level": "INFO", "component": "o.e.n.Node", "cluster.name": "docker-cluster", "node.name": "a06e897599f9", "message": "JVM home [/usr/share/elasticsearch/jdk], using bundled JDK [true]" }
{"type": "server", "timestamp": "2021-02-21T17:36:17,699Z", "level": "INFO", "component": "o.e.n.Node", "cluster.name": "docker-cluster"}]
```



Installation LOCAL | Elastic configuration > yaml file

- Le fichier de configuration de elastic permet de configurer un cluster sur un host (sa machine local ou bien une VM). On peut aussi retrouver ce fichier dans le container.
- On peut y préciser notamment :
 - Le nom du cluster
 - Le nom du noeud principal
 - Les chemins des data et logs
 - Des paramètres réseaux
 - L'adresse de communication avec l'host (ici toute les adresses)

```
[→ elastic docker exec -it a06e897599f9 /bin/bash
[root@a06e897599f9 elasticsearch]# ls
LICENSE.txt  README.asciidoc  config  jdk  logs      plugins
NOTICE.txt   bin              data    lib  modules
[root@a06e897599f9 elasticsearch]# cat config/elasticsearch.yml
cluster.name: "docker-cluster"
network.host: 0.0.0.0
```



PING du cluster

- A noter que elastic consomme de la ressource et met donc du temps à se lancer.

```
➜ elasticsearch curl 127.0.0.1:9200/
{
  "name" : "a06e897599f9",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "_9CFPn8eQuSD1UwcscrBgw",
  "version" : {
    "number" : "7.11.1",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "ff17057114c2199c9c1bbecc727003a907c0db7a",
    "build_date" : "2021-02-15T13:44:09.394032Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
```



Health : état de santé du cluster

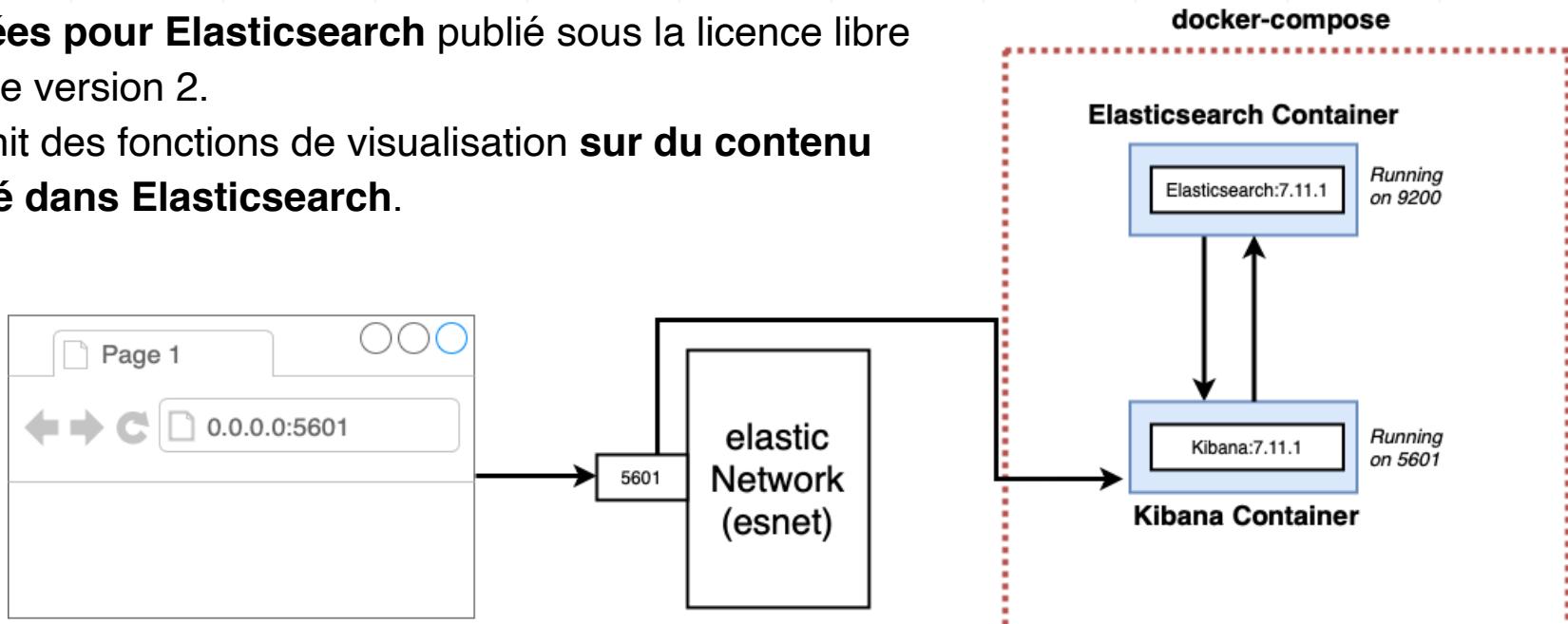
- Avant de commencer il est bon de checker l'état du cluster, si tout est bien ok.
- Si vous n'avez pas jq, vous pouvez aussi faire un `curl http://localhost:9200/_cluster/health?pretty`

```
→ ~ curl 0.0.0.0:9200/_cluster/health | jq
% Total    % Received % Xferd  Average Speed   Time     Time     Time
Current                                            Dload  Upload   Total   Spent   Left
Speed
  0      0      0      0      0      0      0      0  --::--  --::--  --::--
100  390  100  390      0      0  24375      0  --::--  --::--  --::--
- 24375
{
  "cluster_name": "docker-cluster",
  "status": "green",
  "timed_out": false,
  "number_of_nodes": 1,
  "number_of_data_nodes": 1,
  "active_primary_shards": 0,
  "active_shards": 0,
  "relocating_shards": 0,
  "initializing_shards": 0,
  "unassigned_shards": 0,
  "delayed_unassigned_shards": 0,
  "number_of_pending_tasks": 0,
  "number_of_in_flight_fetch": 0,
  "task_max_waiting_in_queue_millis": 0,
  "active_shards_percent_as_number": 100
}
```



Utiliser docker-compose pour lancer Kibana

- Kibana est une **solution de visualisation de données pour Elasticsearch** publié sous la licence libre Apache version 2.
- Il fournit des fonctions de visualisation **sur du contenu indexé dans Elasticsearch**.





Utiliser docker-compose

Bien définir les services

- 2 services :
 - Elasticsearch sur le port 9200
 - Kibana sur le port 5601
- Un network pour faire communiquer les 2 containers

```
version: '2.2'
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.11.1
    container_name: elasticsearch
    restart: always
    environment:
      - xpack.security.enabled=false
      - discovery.type=single-node
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
    cap_add:
      - IPC_LOCK
    volumes:
      - ./elas1:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
      - 9300:9300
    networks:
      - esnet

  kibana:
    container_name: kibana
    image: docker.elastic.co/kibana/kibana:7.11.1
    restart: always
    ports:
      - 5601:5601
    depends_on:
      - elasticsearch
    networks:
      - esnet

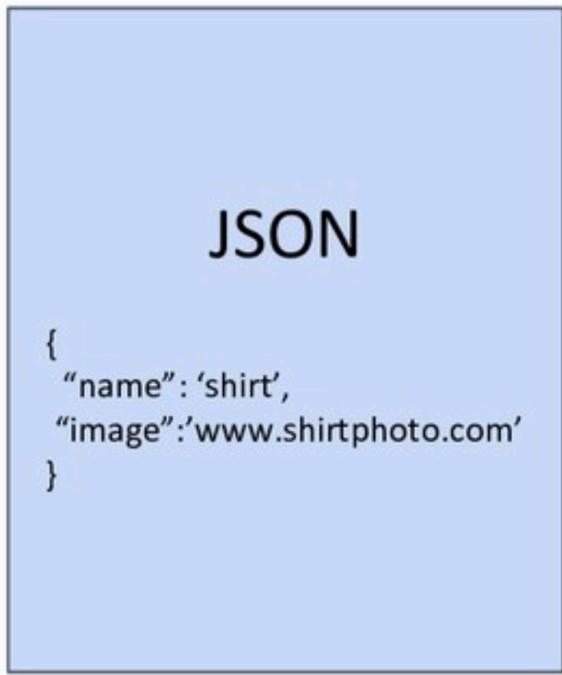
networks:
  esnet:
    driver: bridge
```



En pratique
Premiers endpoints



Communication avec Elasticsearch



Indexing
API



ES



Vérification de l'installation | ingest data > create index hello world

- Afin de vérifier notre installation ajoutons des data à elastic en créant un index hello-wold
- On peut vérifier que l'indice à été correctement crée en lisant tout les indices stocké par elasticsearch
- On peut aussi voir les Indices et les Shards dans le browser Chrome en ajoutant une extension (<https://chrome.google.com/webstore/detail/elasticsearch-head/ffmkiejjmecolpfloofpjologoblkegm/related>)

```
→ curl -XPUT http://localhost:9200/hello-world
{"acknowledged":true,"shards_acknowledged":true,"index":"hello-world"}☒
→ ~ curl http://localhost:9200/_cat/indices?v
health status index          uuid                               pri  rep docs.count docs.deleted store.size pri.store.size
yellow open  cities           aw3sZ8KxRiSxEzeP38W7oA   1    1      3             0        4.6kb     4.6kb
green  open  .apm-custom-link BrvcprG7QFS2VWGR0P7R4A   1    0      0             0        208b     208b
green  open  .kibana_task_manager_1 WpXM7UnSSm-YdMYJ2SivJw  1    0      8             59       255kb     255kb
green  open  .kibana-event-log-7.11.1-000001 pZXs3WPFS8iy3AZJ4qit_g  1    0      1             0        5.6kb     5.6kb
green  open  .apm-agent-configuration uYoR2DBLQuyWRMY3VcidsQ  1    0      0             0        208b     208b
yellow open  hello-world      IsYGKoDJTSy6a7jFYQJsyQ  1    1      0             0        208b     208b
green  open  .kibana_1         p91c4uzaSpWyjQMeT788Kg  1    0     24             38       2.1mb     2.1mb
```



Elasticsearch chrome extension view

ElasticSearch Head | chrome-extension://ffmkiejjmecolpfloofpjologblkegm/elasticsearch-head/index.html

Elasticsearch http://localhost:9200/ Se connecter docker-cluster Santé du cluster: yellow (7 9)

Aperçu Index Navigateur Recherche Structurée [+] Autres requêtes [+] Index Filter Rafraîchir

Kibana stuff

Index

Shards

hello-world size: 208B (208B) docs: 0 (0) Info Actions .cities size: 4.66ki (4.66ki) docs: 3 (3) Info Actions .kibana_task_manager_1 size: 213ki (213ki) docs: 8 (88) Info Actions .kibana_1 size: 2.11Mi (2.11Mi) docs: 24 (62) Info Actions .kibana-event-log-7.11.1-000001 size: 5.63ki (5.63ki) docs: 1 (1) Info Actions .ds-ilm-history-5-2021.02.21-000001 size: unknown docs: unknown Info Actions .apm-custom-link size: 208B (208B) docs: 0 (0) Info Actions .apm-agent-configuration size: 208B (208B) docs: 0 (0) Info Actions .kibana_task_manager .kibana .kibana-event-log-7.11.1

Unassigned 0 0

eda0f4516b88 0 0 0 0 0 0 0 0



Check Elastic in your browser

localhost:5601/app/dev_tools#/console

elastic Search Elastic

Dev Tools

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help

200 - OK 481 ms

```
1 GET _search
2 {
3   "query": {
4     "match_all": {}
5   }
6 }
```

```
1 #! Deprecation: this request accesses system indices: [.apm-agent-configuration, .apm-custom-link, .kibana_1, .kibana_task_manager_1], but in a future major version, direct access to system indices will be prevented by default
2 {
3   "took" : 54,
4   "timed_out" : false,
5   "_shards" : {
6     "total" : 8,
7     "successful" : 8,
8     "skipped" : 0,
9     "failed" : 0
10 },
11   "hits" : {
12     "total" : {
13       "value" : 1045,
14       "relation" : "eq"
15     },
16     "max_score" : 1.0,
17     "hits" : [
18       {
19         "_index" : ".kibana-event-log-7.11.1-000001",
20         "_type" : "_doc",
21         "_id" : "eEysxncB4FJ94ikDbV-r",
22         "_score" : 1.0,
23         "_source" : {
24           "@timestamp" : "2021-02-21T22:18:37.325Z",
25           "event" : {
26             "provider" : "eventLog",
27             "action" : "starting"
28           },
29           "message" : "eventLog starting"
30         }
31       }
32     ]
33   }
34 }
```



Simple PUT (ajouter un index)

Index: Products, Shards: 2, replicas: 2

Console Search Profiler Grok Debugger

```
19
20
21
22
23 #add products index
24 PUT /products
25 {
26   "settings": {
27     "number_of_shards": 2,
28     "number_of_replicas": 2
29   }
30 }
31 #make sure products is into index (master) node
32 GET _cat/indices
33 #check (more details) our product index
34 GET products/
```

▶ ⏴

```
1 { "products" : {
2   "aliases" : { },
3   "mappings" : { },
4   "settings" : {
5     "index" : {
6       "number_of_shards" : "2",
7       "blocks" : {
8         "read_only_allow_delete" : "true"
9       },
10      "provided_name" : "products",
11      "creation_date" : "1572021925755",
12      "number_of_replicas" : "2",
13      "uuid" : "TriEr2_0SiadS7x0GMQ5Xg",
14      "version" : {
15        "created" : "7040099"
16      }
17    }
18  }
19 }
20 }
21 }
```





Les informations basique

Health, nodes, shards, indices

☰ D Dev Tools

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help

```
1 #info about cluster
2 GET _cluster/health
3
4 #info about nodes
5 GET _cat/nodes?v
6
7 #info about shards
8 GET _cat/shards?v
9
10 #more info about nodes and data stored
11 GET _cat/indices?v
12
13
```

	1	health	status	index	docs.count	docs.deleted	store.size	pri.store.size	uuid	pri	rep
	2	yellow	open	cities	3	0	4.6kb		aw3z8KxRiSxEzeP38W7oA	1	1
	3	green	open	.apm-custom-link	0	0	208b		BrvCprG7QFS2VWGR0P7R4A	1	0
	4	green	open	.kibana-event-log	7.11.1-000001	0	5.6kb		pZXs3WPFS8iy3AZJ4qit_g	1	0
	5	green	open	.kibana_task_manager_1	8	260	238kb		WpxM7UnSSm-YdMYJ2SivJw	1	0
	6	green	open	.apm-agent-configuration	0	0	208b		uYoR2DBLQuyWRMY3VcidsQ	1	0
	7	yellow	open	hello-world	0	0	208b		IsYGKoDJTSy6a7jFYQJsyQ	1	1
	8	yellow	open	accounts	1000	0	392.5kb		G1qq5BPuTt21lkbtZHbjuQ	1	1
	9	green	open	.kibana_1	40	4	2.1mb		p91c4uzasPWyjQMeT788Kg	1	0



Delete and create index

Be careful

Dev Tools

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help

200 - OK 1768 ms

```
1 #  
2 #  DELETE index  
3 #  
4 #  
5 #delete index --> 🚧 if index exist 🚧  
6 #DELETE /movies_index  
7 #  
8 #  
9 #  
10 #  PUT index (add bdd)  
11 #  
12 #  
13 #add INDEX "products"  
14 # --> 2 shard into 1 node  
15 PUT /products |   
16 {  
17   "settings": {  
18     "number_of_shards": 2,  
19     "number_of_replicas": 2  
20   }  
21 }  
22
```





Update document with POST

Voir la modification en faisant une requête

History Settings Help

200 - OK 36 ms

```
1
2 #
3 #     UPDATE doc into index
4 #
5
6 #🚧 si le document n'existe pas -> 404 🚧
7 POST cities/_update/Z5D_A24BeSRnGnjbYHc-
8 {
9     "doc" : {
10        | "city" : "Paris mother fucker "
11    }
12 }
13
14
15 POST cities/_update/3
16 {
17     "doc":{
18        | "country" : "America Bitch"
19    }
20 }
21
22 #
23 # verification
24 #
25 GET cities/_search
26 {
27     "query": {
28        | "match_all": {}
29    }
30 }
```

```
1 {
2     "_index" : "cities",
3     "_type" : "_doc",
4     "_id" : "3",
5     "_version" : 2,
6     "result" : "noop",
7     "_shards" : {
8         | "total" : 0,
9         | "successful" : 0,
10        | "failed" : 0
11    },
12     "_seq_no" : 3,
13     "_primary_term" : 1
14 }
```



En pratique

Le endpoint _search et les queries



Matching : requêtes de simple correspondance

- Le service **_search (API Elastic)** permet de faire des **requêtes** sur la base d'un matching
- Ces requêtes utilisent le **format DSL (Domain Specific Language)**, il faut envoyer un document JSON "requête"

```
{  
  "query" : {  
    "match" : { "nom_de_l'attribut" : "la\valeur\_\_a\chercher"}  
  }  
}
```

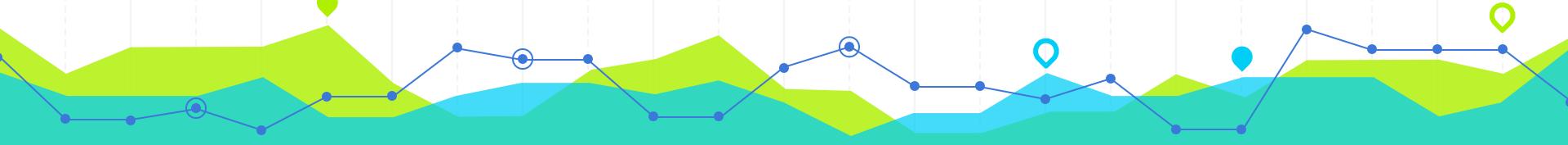




Matching : requêtes par intervalle

- Le service **_search (API Elastic)** permet de faire des requêtes sur la base d'un matching et affiner ces critères de recherche, notamment avec :
- Des ranges (intervalles)
- La notion de MUST que nous venons par la suite.

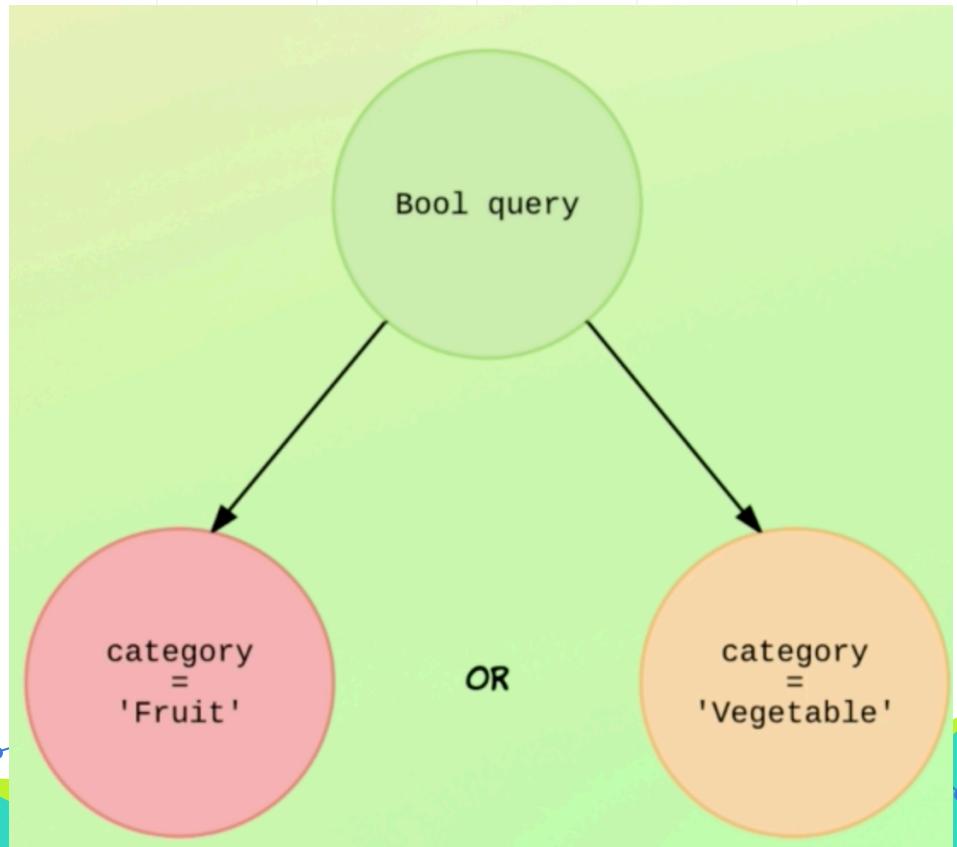
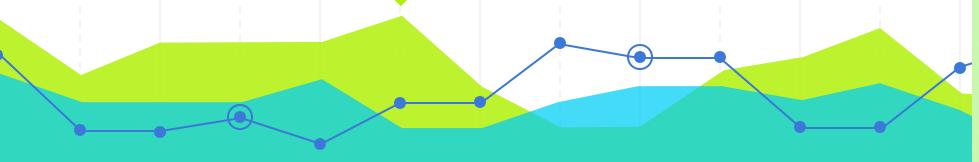
```
{  
  "query" : {  
    "range" : { "nom_de_l'attribut" : "la\_\_borne"}  
  }  
}
```





Les boolean queries

- Contrairement à une query simple elle va prendre un opérateur logique (OR par défaut) dans les critères de recherche.
- Ce type de query va donc impliquer les notions de critère dans la recherche (must / should)**

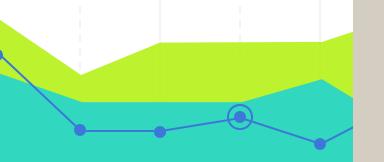




Notions de critères : SHOULD / MUST

- SHOULD : critère optionnel avec calcul de score de pertinence
- MUST : critère obligatoire (sans score).
- A noter qu'il faut utiliser **ces critères en les imbriquant dans un opérateur 'bool'**

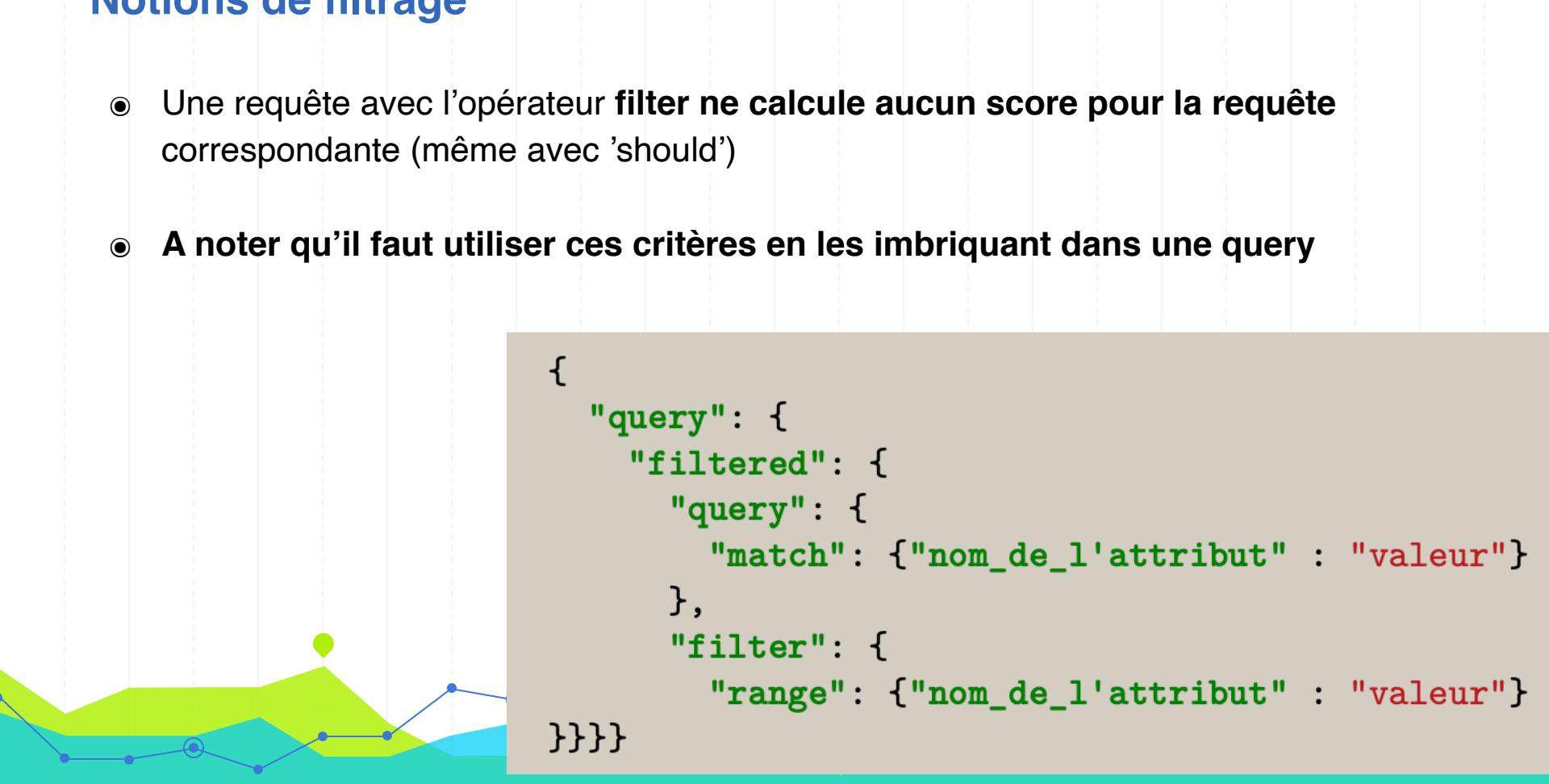
```
{"query":{  
    "bool": {  
        "should": { "match": { "nom_de_l'attribut" : "valeur" }},  
        "must": { "range": { "nom_de_l'attribut" : "valeur"}},  
        "must_not": [  
            {"match":{ "nom_de_l'attribut" : "valeur"}},  
            {"match":{ "nom_de_l'attribut" : "valeur"}},  
            ...  
        ]  
    }}}
```





Notions de filtrage

- Une requête avec l'opérateur **filter** ne calcule aucun score pour la requête correspondante (même avec 'should')
- A noter qu'il faut utiliser ces critères en les imbriquant dans une query



```
{  
    "query": {  
        "filtered": {  
            "query": {  
                "match": {"nom_de_l'attribut" : "valeur"}  
            },  
            "filter": {  
                "range": {"nom_de_l'attribut" : "valeur"}  
            }  
        }  
    }  
}
```



Notions d'agrégats

- L'agrégation c'est l'opération de regrouper des documents pour pouvoir appliquer une (ou plusieurs) statistique(s).
- Pour utiliser aggs, **il faut donner la clé où prendre les valeurs terms : field : « XXX » (identique à un GROUP BY)** et donner la clé où mettre la valeur d'agrégation.
- Par défaut, cela va compter le nombre d'occurrences.

```
{"aggs" : {  
    "nom_de_la_cle_en_sortie" : {  
        "terms" : {"field" : "nom_de_la_cle_a_grouper"}  
    }  
}}
```





Notions d'agrégats

- On peut également **changer terms en "avg", "min", "max", "cardinality" (distinct)**...
Voir même ressortir les mots importants "significant_terms" (grâce à leurs occurrences dans un ensemble filtrés)
- Il est aussi possible de filtrer les documents avant l'agrégat. Il suffit de faire une requête classique "query".

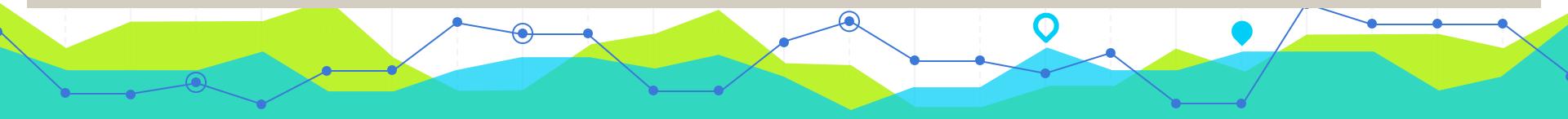
```
{"query" :{  
    "match" : {"cle_a_filtrer" : "valeur"}  
}, "aggs" : {  
    "cle_de_sortie" : {  
        "avg" : {"field" : "cle_a_grouper"}  
    }}}
```



Grouper les agrégats

- Il est possible d'imbriquer les agrégats pour grouper par type.

```
{"aggs" : {  
    "nom_groupe" : {  
        "terms" : {  
            "field" : "cle_du_group_by"  
        },  
        "aggs" : {  
            "nom_sous_groupe" : {  
                "avg" : {"field" : "cle_pour_la_moyenne"}  
            }  
        }  
    }  
}}
```





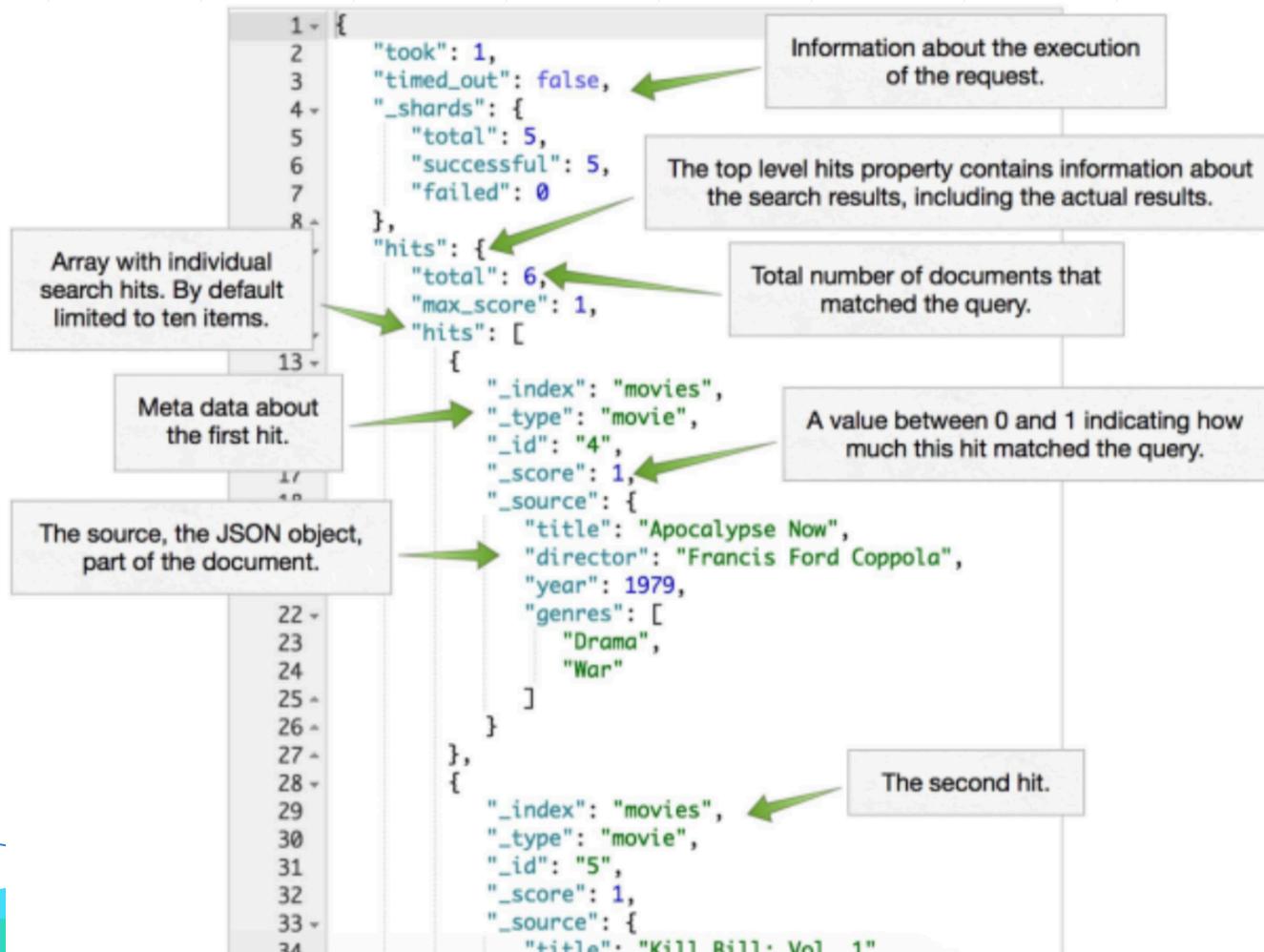
Grouper par plage de valeur

- Il est possible aussi de grouper par plages de valeur plutôt que les valeurs du document.

```
{"aggs" : {  
    "nom_cle_sortie" : {  
        "range" : {  
            "field" : "cle_ou_prendre_la_valeur",  
            "ranges" : [  
                {"to" : "valeur_max"},  
                {"from" : "valeur_min", "to" : "valeur_max"},  
                {"from" : "valeur_min"}  
            ]  
        }  
    }  
}}
```



Format de retour





GET - Endpoint _search

History Settings Help

200 - OK

49 ms

```
1
2 #
3 # FIRST QUERY
4 #
5
6 GET cities/_search
7 {}
8 {
9   "query": {
10     "match_all": {}
11   }
12
13
14
15
```



```
1 {
2   "took" : 2,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 3,
13      "relation" : "eq"
14    },
15    "max_score" : 1.0,
16    "hits" : [
17      {
18        "_index" : "cities",
19        "_type" : "places",
20        "_id" : "1",
21        "_score" : 1.0,
22        "_source" : {
23          "city" : "New Delhi",
24          "country" : "India"
25        }
26      },
27      {
28        "_index" : "cities",
29        "_type" : "places",
30        "_id" : "2",
31        "_score" : 1.0,
32        "_source" : {
33          "city" : "London",
34          "country" : "England"
35        }
36      },
37    ]
38  }
```



GET - Endpoint _search avec l'option query

History Settings Help

200 - OK 5807 ms

```
1 #  
2 #   PUT index (add bdd)  
3 #  
4 #  
5 PUT /movies  
6 {  
7   "settings": {  
8     "number_of_shards": 5,  
9     "number_of_replicas": 2  
10    }  
11  }  
12  
13  
14 #  
15 #   MOVIES DATASET EXPLORATION AFTER BULK # |  
16 #  
17  
18 # ~ SELECT * FROM movies  
19 GET movies/_search  
20 {  
21   "query": {  
22     "match_all": {}  
23   }  
24 }  
25  
26 #  
27 # ~ SELECT title FROM movies WHERE "Star Wars" IS IN movies.title  
28 #  
29 GET movies/_search  
30 {  
31   "query": {  
32     "match": {  
33       "fields.title": "Star Wars"  
34     }  
35   }  
36 }
```

```
1 {  
2   "took" : 5771,  
3   "timed_out" : false,  
4   "_shards" : {  
5     "total" : 5,  
6     "successful" : 5,  
7     "skipped" : 0,  
8     "failed" : 0  
9   },  
10  "hits" : {  
11    "total" : {  
12      "value" : 4849,  
13      "relation" : "eq"  
14    },  
15    "max_score" : 1.0,  
16    "hits" : [  
17      {  
18        "_index" : "movies",  
19        "_type" : "movie",  
20        "_id" : "3",  
21        "_score" : 1.0,  
22        "_source" : {  
23          "fields" : {  
24            "directors" : [  
25              "Denis Villeneuve"  
26            ],  
27            "release_date" : "2013-08-30T00:00:00Z",  
28            "rating" : 8.2,  
29            "genres" : [  
30              "Crime",  
31              "Drama",  
32              "Thriller"  
33            ],  
34            "image_url" : "http://ia.media-imdb.com/images/M  
/MV5BMTg0NTIzMjQ1NV5BML5BanBnXkFtZTcwNDc3MzM5Q@@._V1_SX400_.jpg",  
35            "plot" : "When Keller Dover's daughter and her friend go missing, he takes  
matters into his own hands as the police pursue multiple leads and the
```



Visualisation de l'index avec l'extension chrome

Elasticsearch http://localhost:9200/ Se connecter docker-cluster Santé du cluster: yellow (13 26) Info

Aperçu Index Navigateur Recherche Structurée [+] Autres requêtes [+]

Aperçu du cluster Sort Cluster Sort Indices View Aliases Index Filter Rafraîchir

movies
size: 4.67Mi (4.67Mi)
docs: 4 849 (4 849)
[Info](#) [Actions](#)

hello-world
size: 208B (208B)
docs: 0 (0)
[Info](#) [Actions](#)

cities
size: 10.3ki (10.3ki)
docs: 3 (4)
[Info](#) [Actions](#)

accounts
size: 393ki (393ki)
docs: 1 000 (1 000)
[Info](#) [Actions](#)

.kibana_task_manager_1
size: 308ki (308ki)
docs: 8 (454)
[Info](#) [Actions](#)

.kibana_1
size: 4.27Mi (4.27Mi)
docs: 62 (199)
[Info](#) [Actions](#)

.kibana-event-log-7.11.1-000001
size: 5.63ki (5.63ki)
docs: 1 (1)
[Info](#) [Actions](#)

.ds-ilm-history-5-2021.02.21-000001
size: unknown
docs: unknown
[Info](#) [Actions](#)

.kibana X

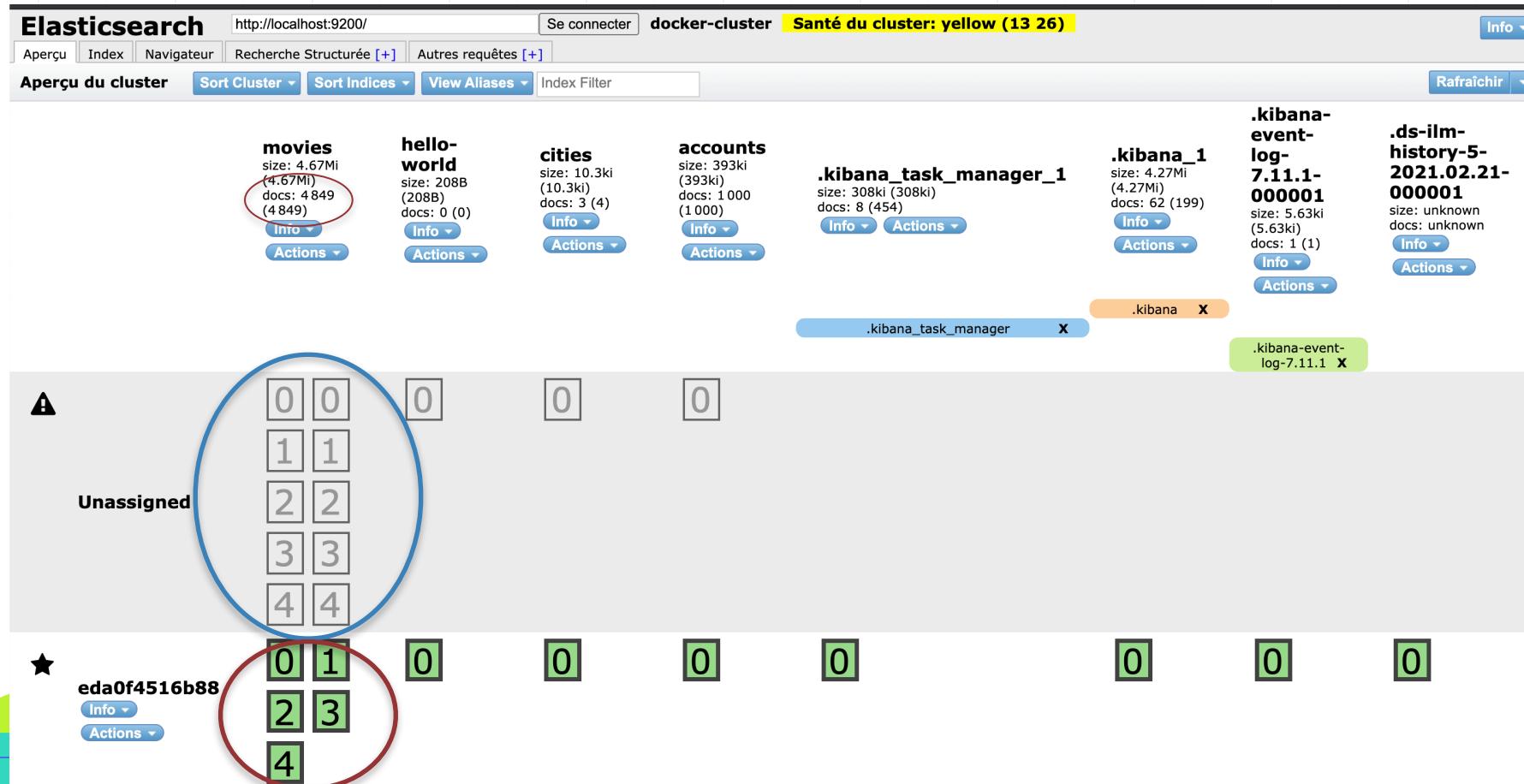
.kibana-event-log-7.11.1 X

Unassigned ⚠

0	0	0	0	0
1	1			
2	2			
3	3			
4	4			

eda0f4516b88 ★

0	1	0	0	0	0	0	0	0
2	3							
4								





Bien connaître son Mapping avant de faire des requêtes complexes

History Settings Help

200 - OK

35 ms

```
1
2 #
3 # GET DOC MAPPING
4 #
5 GET /movies/_mapping
6
```

```
1 {
2   "movies" : {
3     "mappings" : {
4       "properties" : {
5         "fields" : {
6           "properties" : {
7             "actors" : {
8               "type" : "text",
9               "fields" : {
10                 "keyword" : {
11                   "type" : "keyword",
12                   "ignore_above" : 256
13                 }
14               }
15             },
16             "directors" : {
17               "type" : "text",
18               "fields" : {
19                 "keyword" : {
20                   "type" : "keyword",
21                   "ignore_above" : 256
22                 }
23               }
24             },
25             "genres" : {
26               "type" : "text",
27               "fields" : {
28                 "keyword" : {
29                   "type" : "keyword",
30                   "ignore_above" : 256
31                 }
32               }
33             },
34             "image_url" : {
35               "type" : "text",
36               "fields" : {
37                 "keyword" : {
38                   "type" : "keyword"
39                 }
40               }
41             }
42           }
43         }
44       }
45     }
46   }
47 }
```



L'extension chrome : visualiser vos indices

← → ⌂ ★ Elasticsearch Head | chrome-extension://ffmkiejjmecolpfloofpjologblkegm/elasticsearch-head/index.html ⋮

Elasticsearch <http://localhost:9200/> Se connecter docker-cluster Santé du cluster: yellow (13 26) Info

Aperçu Index Navigateur Recherche Structurée [+] Autres requêtes [+]

Rafraîchir

Navigateur

Tous les index ▼

INDEX

- .apm-agent-configuration
- .apm-custom-link
- .ds-ilm-history-5-2021.02.21-000001
- .kibana-event-log-7.11.1-000001
- .kibana_1
- .kibana_task_manager_1
- accounts
- cities
- hello-world
- movies

TYPES

- _doc
- docs
- movie
- places

CHAMPS

- @timestamp ?
- account_number
- action.actionTypeId
- action.name
- action.secrets
- action_task_params.actionId
- action_task_params.apiKey
- address
- age
- agent_name
- alert.actions.actionRef
- alert_actions.actionTypeId

Recherche sur 5 des 5 shards. 4849 résultats. 0.039 secondes

	fields.title	fields.rank	fields.running_time_secs	fields.year	id	type
	Prisoners	3	9180	2013	tt1392214	add
	Thor: The Dark World	5		2013	tt1981115	add
	X-Men: Days of Future Past	9		2014	tt1877832	add
	Now You See Me	11	6900	2013	tt1670345	add
	We're the Millers	13	6600	2013	tt1723121	add
	Star Trek Into Darkness	16	7920	2013	tt1408101	add
	After Earth	17	6000	2013	tt1815862	add
	La vie d'Adèle	19	10740	2013	tt2278871	add
	Runner Runner	35	5460	2013	tt2364841	add
	I Spit on Your Grave 2	36	6360	2013	tt2537176	add
	Battle of the Year	37	6600	2013	tt1532958	add
	The Bling Ring	41	5400	2013	tt2132285	add
	Furious 6	42	7800	2013	tt1905041	add
	Metallica Through the Never	49	5640	2013	tt2172935	add
	Kick-Ass 2	53	6180	2013	tt1650554	add
	Diana	54	6780	2013	tt1758595	add
	2 Guns	62	6540	2013	tt1272878	add
	Wreck-It Ralph	73	6480	2012	tt1772341	add
	Frozen	75		2013	tt2294629	add
	Spring Breakers	76	5640	2012	tt2101441	add
	The Shawshank Redemption	80	8520	1994	tt0111161	add
	Hummingbird	93	6000	2013	tt1893256	add
	Mud	95	7800	2012	tt1935179	add
	Captain Phillips	101	8040	2013	tt1535109	add
	Psycho	105	6540	1960	tt0054215	add
	The Fifth Estate	110	7680	2013	tt1837703	add
	The Purge	125	5100	2013	tt2184339	add
	The Perks of Being a Wallflower	127	6120	2012	tt1659337	add



L'extension chrome : visualiser vos champs rapidement

ElasticSearch Head | chrome-extension://ffmkiejjmecolpfloofpjologblkegm/elasticsearch-head/index.html

Se connecter docker-cluster Santé du cluster: yellow (13 26) Info

Aperçu Index Navigateur Recherche Structurée [+] Autres requêtes [+]

Chercher dans movies (4849 docs) les documents correspondant à must match_all

Recherche Format d'affichage des résultats Tableau Nombre de Résultats: 10 Voir la requête source

Recherche sur 5 des 5 shards. 4849 résultats. 0.026 secondes

_index	_type	_id	_score	fields.release_date	fields.rating	fields.image_url	fields.plot
movies	movie	3	1	2013-08-30T00:00:00Z	8.2	http://ia.media-imdb.com/images/M/MV5BMTg0NTIzMjQ1NV5BMi5BanBnXkFtZTcwNdC3MzM5OQ@@._V1_SX400_.jpg	When Keller Dover's daughter
movies	movie	5	1	2013-10-30T00:00:00Z		http://ia.media-imdb.com/images/M/MV5BMTQyNzAwOTUxOF5BMi5BanBnXkFtZTcwMTE0OTc5OQ@@._V1_SX400_.jpg	Faced with an enemy that e
movies	movie	9	1	2014-05-21T00:00:00Z		http://ia.media-imdb.com/images/M/MV5BMTQ0NzIwNTA1MV5BMi5BanBnXkFtZTgwNjY2OTcwMDE@._V1_SX400_.jpg	The X-Men send Wolverine
movies	movie	11	1	2013-05-21T00:00:00Z	7.3	http://ia.media-imdb.com/images/M/MV5BMTY0NDY3MDMxN15BMi5BanBnXkFtZTcwOTM5NzMzOQ@@._V1_SX400_.jpg	An FBI agent and an Interp
movies	movie	13	1	2013-08-03T00:00:00Z	7.2	http://ia.media-imdb.com/images/M/MV5BMjA5Njc0NDUxNV5BMi5BanBnXkFtZTcwMjYzNzU1OQ@@._V1_SX400_.jpg	A veteran pot dealer create
movies	movie	16	1	2013-05-02T00:00:00Z	7.9	http://ia.media-imdb.com/images/M/MV5BMTk2NzcOTgxNF5BMi5BanBnXkFtZTcwODQ5ODczOQ@@._V1_SX400_.jpg	After the crew of the Enterp
movies	movie	17	1	2013-05-01T00:00:00Z	4.9	http://ia.media-imdb.com/images/M/MV5BMTY3MzQyMjkwMj5BMi5BanBnXkFtZTcwMDk2OTE0OQ@@._V1_SX400_.jpg	A crash landing leaves Kitai
movies	movie	19	1	2013-05-23T00:00:00Z	7.3	http://ia.media-imdb.com/images/M/MV5BMTQ5NTg5ODK4OV5BMi5BanBnXkFtZTgwODc4MTMzMDE@._V1_SX400_.jpg	The story of a young lesbian
movies	movie	35	1	2013-09-25T00:00:00Z	5.4	http://ia.media-imdb.com/images/M/MV5BMTU5OTA0MjI4Mj5BMi5BanBnXkFtZTgwMTgxOTQwMDE@._V1_SX400_.jpg	When a poor college studen
movies	movie	36	1	2013-08-25T00:00:00Z	5.5	http://ia.media-imdb.com/images/M/MV5BMTc1Nzk3NjM0M15BMi5BanBnXkFtZTgwMjM5MzYyMDE@._V1_SX400_.jpg	





L'extension chrome : visualiser vos champs rapidement

Elasticsearch http://localhost:9200/ Se connecter docker-cluster Santé du cluster: yellow (13 26) Info

Aperçu Index Navigateur Recherche Structurée [+] Autres requêtes [+]

Historique

```
_search {"query": {"match_all": {}}}; return root;
_search {"query": {"match": {"fields.title": "Star Wars"}}}; return root;
```

Recherche

```
http://localhost:9200/_search POST
{"query": {"match": {"fields.title": "Star Wars"}}}
```

Recherche sur 13 des 13 shards. 29 résultats. 0.012 secondes

	movie._source.fields.title	movie._source.fields.rank	movie._source.fields.running_time_secs	movie._source.fields
evil Darth Vader.	Star Wars	226	7260	M
	Star Wars: Episode VII	168		M
	Rock Star	2509	6300	M
	Star Wars: Episode I - The Phantom Menace	469	8160	E
of his plan to capture Luke.	Star Wars: Episode V - The Empire Strikes Back	651	7440	M
	Bride Wars	1810	5340	K
	Star Wars: Episode III - Revenge of the Sith	371	8400	H
	Star Wars: Episode II - Attack of the Clones	922	8520	H
	Star Wars: Episode VI - Return of the Jedi	782	8040	M
	Star Trek	128	7620	C

Requête Valider le JSON Pretty

Transformation des résultats ?

```
function(root, prev) {
    return root;

}

Répétition automatique de la requête
Répéter la requête toutes les do not repeat

Options d'affichage ?
 En JSON brut
 En graphe
 En tableau
```



La différence entre les query simple et les query booléennes

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help

200 - OK 84 ms

```
1 GET movies/_search
2 {
3   "query": {
4     "match": {
5       "fields.title": {
6         "query": "Star Wars",
7         "operator": "and"
8       }
9     }
10 }
11 }
```

```
1 {
2   "took" : 36,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 5,
6     "successful" : 5,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 7,
13      "relation" : "eq"
14    },
15    "max_score" : 11.326516,
16    "hits" : [
17      {
18        "_index" : "movies",
19        "_type" : "movie",
20        "_id" : "226",
21        "_score" : 11.326516,
22        "_source" : {
23          "fields" : {
24            "directors" : [
25              "George Lucas"
26            ],
27            "release_date" : "1977-05-25T00:00:00Z",
28            "rating" : 8.7,
29            "genres" : [
30              "Action",
31              "Adventure",
32              "Fantasy",
33              "Sci-Fi"
34            ],
35            "plot" : "A George Lucas film directed by George Lucas, starring Harrison Ford as Han Solo and Mark Hamill as Luke Skywalker. It is the first film in the Star Wars franchise and was released in 1977. The plot follows the Rebel Alliance's efforts to destroy the Death Star, an enormous battle station built by the Galactic Empire to oblinate entire planets. The film features a cast of iconic characters including Han Solo, Luke Skywalker, Leia Organa, Chewbacca, and Obi-Wan Kenobi. The music by John Williams is a hallmark of the franchise, and the special effects by Industrial Light & Magic were groundbreaking for their time."}
36          }
37        }
38      }
39    ]
40  }
41 }
```



La différence entre les query simple et les query emboîtées

History Settings Help

200 - OK

140 ms

1

D

Dev Tools

1

10 of 10

[Search Profiler](#) [Grok Debugger](#)

Painless Lab

BETA

History Settings Help

200 - 01

69 ms

```
1
2 #
3 # MUST operator
4 #
5
6 GET movies/_search
7 {
8     "query": {
9         "bool": {
10            "must": [
11                { "match": { "fields.title": "Star Wars"}},
12                { "match": { "fields.directors": "George Miller" }},
13            ],
14            "must_not": { "match": { "fields.directors": "George Lucas" }},
15            "should": [
16                { "match": { "fields.title": "Star" }},
17                { "match": { "fields.directors": "George Lucas" }}
18            ]
19        }
20    }
21 }
22 }
```

```
1  {
2    "took" : 20,
3    "timed_out" : false,
4    "_shards" : {
5      "total" : 5,
6      "successful" : 5,
7      "skipped" : 0,
8      "failed" : 0
9    },
10   "hits" : {
11     "total" : {
12       "value" : 25,
13       "relation" : "eq"
14     },
15     "max_score" : 14.557282,
16     "hits" : [
17       {
18         "_index" : "movies",
19         "_type" : "movie",
20         "_id" : "2509",
21         "_score" : 14.557282,
22         "_source" : {
23           "fields" : {
24             "directors" : [
25               "Stephen Herek"
26             ],
27             "release_date" : "2001-09-04T00:00:00Z",
28             "rating" : 5.9,
29             "imdb_id" : "tt0134295"
30           }
31         }
32       }
33     ]
34   }
35 }
```



Les notions de critères optionnel

History Settings Help

200 - OK 69 ms

```
1
2  #
3 # MUST operator
4 #
5
6 GET movies/_search
7 {
8   "query": {
9     "bool": {
10    "must": [
11      { "match": { "fields.title": "Star Wars"}}
12    ],
13    "must_not": { "match": { "fields.directors": "George Miller" }},
14    "should": [
15      { "match": { "fields.title": "Star" }},
16      { "match": { "fields.directors": "George Lucas"}}
17    ]
18  }
19 }
20 }
21 }
22 }
```



```
1 {
2   "took" : 20,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 5,
6     "successful" : 5,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 25,
13      "relation" : "eq"
14    },
15    "max_score" : 14.557282,
16    "hits" : [
17      {
18        "_index" : "movies",
19        "_type" : "movie",
20        "_id" : "2509",
21        "_score" : 14.557282,
22        "_source" : {
23          "fields" : {
24            "directors" : [
25              "Stephen Herek"
26            ],
27            "release_date" : "2001-09-04T00:00:00Z",
28            "rating" : 5.9,
29            "genres" : [
30              "Comedy",
31              "Drama",
32              "Music"
33            ],
34            "image_url" : "http://ia.media-imdb.com/images/M/ /MV5BmjE4NTYyNTQ0M15BML5BanBnXkFtZTcwNDYwMTAyMQ@@._V1_SX400_.jpg",
35          }
36        }
37      }
38    ]
39  }
40}
```



Le multi matching

History Settings Help

200 - OK 144 ms

```
1 #  
2 # multiple search  
3 #  
4 #  
5 GET movies/_search  
6 {  
7   "query": {  
8     "multi_match": {  
9       "query": "Star+Wars+George",  
10      "fields": ["fields.title", "fields.directors"]  
11    }  
12  }  
13 }  
14  
15
```

```
1 {  
2   "took" : 101,  
3   "timed_out" : false,  
4   "_shards" : {  
5     "total" : 5,  
6     "successful" : 5,  
7     "skipped" : 0,  
8     "failed" : 0  
9   },  
10  "hits" : {  
11    "total" : {  
12      "value" : 82,  
13      "relation" : "eq"  
14    },  
15    "max_score" : 11.326516,  
16    "hits" : [  
17      {  
18        "_index" : "movies",  
19        "_type" : "movie",  
20        "_id" : "226",  
21        "_score" : 11.326516,  
22        "_source" : {  
23          "fields" : {  
24            "directors" : [  
25              "George Lucas"  
26            ],  
27            "release_date" : "1977-05-25T00:00:00Z",  
28            "rating" : 8.7,  
29            "genres" : [  
30              "Action",  
31              "Adventure",  
32              "Fantasy",  
33              "Sci-Fi"  
34            ],  
35            "image_url" : "http://ia.media-imdb.com/images/M  
/MV5BMTU4NTczODkwM15BML5BanBnXkFtZTcwMzEyMTIyMw@._V1_SX400_.jpg",  
36          }  
37        }  
38      }  
39    }  
40  }  
41 }  
42 }
```



Matching avec un terme particulier

History Settings Help

200 - OK 74 ms

```
1
2 #
3 # match with term in particular field
4 #
5 GET /products/_search
6 {
7   "query": {
8     "term": {
9       "name": "lobster"
10    }
11  }
12 }
13
14 #
15 #other
16 #
17 GET /products/_search
18 {
19   "query": {
20     "term": {
21       "is_active": true
22     }
23  }
24 }
```

```
1 {
2   "took" : 43,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 487,
13      "relation" : "eq"
14    },
15    "max_score" : 0.7194644,
16    "hits" : [
17      {
18        "_index" : "products",
19        "_type" : "_doc",
20        "_id" : "1",
21        "_score" : 0.7194644,
22        "_source" : {
23          "name" : "Wine - Maipo Valle Cabernet",
24          "price" : 152,
25          "in_stock" : 38,
26          "sold" : 47,
27          "tags" : [
28            "Alcohol",
29            "Wine"
30          ],
31          "description" : "Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer tincidunt ante vel ipsum. Praesent blandit lacinia erat. Vestibulum sed magna at nunc commodo placerat. Praesent blandit. Nam nulla. Integer pede justo, lacinia eget, tincidunt eget, tempus vel, pede. Morbi porttitor lorem id ligula.",
32          "is_active" : true,
33          "created" : "2001/05/12"
34        }
35      }
36    ]
37  }
38 }
```



Rechercher par mots clefs

History Settings Help

200 - OK 61 ms

```
1
2  #
3  #search by keyword
4  #
5  GET /products/_search
6  {
7  query": {
8  "terms": {
9  "tags.keyword": ["Soup", "Cake"]
10 }
11 }
12 }
```



```
quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut erat
",
"is_active" : false,
"created" : "2008/05/18"
}
},
{
"_index" : "products",
"_type" : "_doc",
"_id" : "84",
"_score" : 1.0,
"_source" : {
"name" : "Soup - Base Broth Beef",
"price" : 123,
"in_stock" : 30,
"sold" : 325,
"tags" : [
"Soup",
"Meat"
],
"description" : "Etiam pretium iaculis justo. In hac habitasse platea
dictumst. Etiam faucibus cursus urna. Ut tellus. Nulla ut erat id mauris
vulputate elementum.",
"is_active" : false,
"created" : "2002/03/22"
}
},
{
"_index" : "products",
"_type" : "_doc",
"_id" : "92",
"_score" : 1.0,
"_source" : {
"name" : "Soup - Campbells Chili",
"price" : 15,
"in_stock" : 26,
"sold" : 153
}
```



Rechercher par range

History Settings Help

200 - OK 54 ms

```
1
2 #
3 # search by parameters | range
4 #
5
6 GET /products/_search|  
7 {  
8   "query": {  
9     "range": {  
10       "in_stock": {  
11         "gte":1,  
12         "lte": 3  
13       }  
14     }  
15   }  
16 }
```



```
1 {  
2   "took" : 28,  
3   "timed_out" : false,  
4   "_shards" : {  
5     "total" : 1,  
6     "successful" : 1,  
7     "skipped" : 0,  
8     "failed" : 0  
9   },  
10  "hits" : {  
11    "total" : {  
12      "value" : 58,  
13      "relation" : "eq"  
14    },  
15    "max_score" : 1.0,  
16    "hits" : [  
17      {  
18        "_index" : "products",  
19        "_type" : "_doc",  
20        "_id" : "12",  
21        "_score" : 1.0,  
22        "_source" : {  
23          "name" : "Chocolate Eclairs",  
24          "price" : 195,  
25          "in_stock" : 1,  
26          "sold" : 202,  
27          "tags" : [ ],  
28          "description" : "Aliquam non mauris. Morbi non lectus. Aliquam sit amet diam  
in magna bibendum imperdiet. Nullam orci pede, venenatis non, sodales sed,  
tincidunt eu, felis. Fusce posuere felis sed lacus. Morbi sem mauris,  
laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem.  
Sed sagittis. Nam congue, risus semper porta volutpat, quam pede lobortis  
ligula, sit amet eleifend pede libero quis orci. Nullam molestie nibh in  
lectus.",  
29          "is_active" : true,  
30          "created" : "2005/08/02"  
31      }  
32    }  
33  }  
34 }
```



Rechercher par range (datetime)

History Settings Help

200 - OK 46 ms

```
1
2 #
3 # search by parameters | range
4 #
5
6 GET /products/_search
7 {
8   "query": {
9     "range": {
10       "in_stock": {
11         "gte":1,
12         "lte": 3
13       }
14     }
15   }
16 }
17
18
19
20 GET /products/_search
21 {
22   "query": {
23     "range": {
24       "created": {
25         "gte": "2010/01/01",
26         "lte": "2010/12/31"
27       }
28     }
29   }
30 }
```



```
1 {
2   "took" : 15,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 57,
13      "relation" : "eq"
14    },
15    "max_score" : 1.0,
16    "hits" : [
17      {
18        "_index" : "products",
19        "_type" : "_doc",
20        "_id" : "24",
21        "_score" : 1.0,
22        "_source" : {
23          "name" : "Wine - Ej Gallo Sierra Valley",
24          "price" : 21,
25          "in_stock" : 27,
26          "sold" : 425,
27          "tags" : [
28            "Alcohol",
29            "Wine"
30          ],
31          "description" : "Duis aliquam convallis nunc. Proin at turpis a pede posuere nonummy. Integer non velit. Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec pharetra, magna vestibulum aliquet ultrices, erat tortor sollicitudin mi, sit amet lobortis sapien sapien non mi. Integer ac neque. Duis bibendum. Morbi non quam nec dui luctus rutrum. Nulla tellus. In sagittis dui vel nisi."
32        }
33      }
34    ]
35  }
```



Logique booléenne des moteurs de recherche

History Settings Help

200 - OK 44 ms

```
1 #
2 # BOOLEAN LOGIC |
3 #
4
5 #get with bool range
6 GET receipe/_search
7 {
8   "query": {
9     "bool": {
10       "must": [
11         {
12           "match": {
13             "ingredients.name": "parmesan"
14           }
15         },
16         {
17           "range": {
18             "preparation_time_minutes": {
19               "gte": 10,
20               "lte": 20
21             }
22           }
23         }
24       ]
25     }
26   }
27 }
28 }
```



```
1 {
2   "took" : 7,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 3,
13      "relation" : "eq"
14    },
15    "max_score" : 2.3795729,
16    "hits" : [
17      {
18        "_index" : "receipe",
19        "_type" : "_doc",
20        "_id" : "1",
21        "_score" : 2.3795729,
22        "_source" : {
23          "title" : "Fast and Easy Pasta With Blistered Cherry Tomato Sauce",
24          "description" : "Cherry tomatoes are almost always sweeter, riper, and higher in pectin than larger tomatoes at the supermarket. All of these factors mean that cherry tomatoes are fantastic for making a rich, thick, flavorful sauce. Even better: It takes only four ingredients and about 10 minutes, start to finish-less time than it takes to cook the pasta you're gonna serve it with."
25        },
26        "preparation_time_minutes" : 12,
27        "servings" : {
28          "min" : 4,
29          "max" : 6
30        },
31        "steps" : [
32          "Place pasta in a large skillet or saut  pan and cover with water and a big pinch of salt. Bring to a boil over high heat, stirring occasionally."
33        ]
34      }
35    ]
36  }
37 }
```



Logique booléenne des moteurs de recherche

History Settings Help

200 - OK 46 ms

```
1 #get with bool range & filters
2 GET receipe/_search
3 {
4   "query": {
5     "bool": {
6       "must": [
7         {
8           "match": {
9             "ingredients.name": "parmesan"
10            }
11          }
12        ],
13        "must_not": [
14          {
15            "match": {
16              "ingredients.name": "tuna"
17            }
18          ]
19        ],
20        "filter": [
21          {
22            "range": {
23              "preparation_time_minutes": {
24                "lte": 15
25              }
26            }
27          }
28        ]
29      }
30    }
31  }
32 }
```

```
1 {
2   "took" : 7,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 2,
13      "relation" : "eq"
14    },
15    "max_score" : 1.379573,
16    "hits" : [
17      {
18        "_index" : "receipe",
19        "_type" : "_doc",
20        "_id" : "1",
21        "_score" : 1.379573,
22        "_source" : {
23          "title" : "Fast and Easy Pasta With Blistered Cherry Tomato Sauce",
24          "description" : "Cherry tomatoes are almost always sweeter, riper, and higher in pectin than larger tomatoes at the supermarket. All of these factors mean that cherry tomatoes are fantastic for making a rich, thick, flavorful sauce. Even better: It takes only four ingredients and about 10 minutes, start to finish-less time than it takes to cook the pasta you're gonna serve it with."
25        },
26        "preparation_time_minutes" : 12,
27        "servings" : {
28          "min" : 4,
29          "max" : 6
30        },
31        "steps" : [
32          "Place pasta in a large skillet or saut\u00e9 pan and cover with water and a big pinch of salt. Bring to a boil over high heat, stirring occasionally."
33        }
34      }
35    ]
36  }
37 }
```



Sorting result

```
1 | GET /recipe/default/_search
2 - {
3   "_source": false,
4   "query": {
5     "match_all": {}
6   },
7   "sort": [
8     "preparation_time_minutes"
9   ]
10 }
11
12 |
```

```
1 - {
2   "took": 42,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "failed": 0
8   },
9   "hits": {
10    "total": 20,
11    "max_score": null,
12    "hits": [
13      {
14        "_index": "recipe",
15        "_type": "default",
16        "_id": "5",
17        "_score": null,
18        "sort": [
19          8
20        ]
21      },
22      {
23        "_index": "recipe",
24        "_type": "default",
25        "_id": "6",
26        "_score": null,
27        "sort": [
28          10
29        ]
30      },
31    ]
32  }
33}
```



Sorting by multi value fields

```
1 GET /recipe/default/_search
2 {
3     "_source": "ratings",
4     "query": {
5         "match_all": {}
6     },
7     "sort": [
8         {
9             "ratings": {
10                 "order": "desc",
11                 "mode": "avg"
12             }
13         }
14     ]
15 }
```





Debugging query

History Settings Help

```
11 GET receipt/_mapping
12
13 #get with bool range & filters
14 GET receipt/_search
15 {
16   "query": {
17     "bool": {
18       "must": [
19         {
20           "match": {
21             "ingredients.name": {
22               "query": "parmesan",
23               "_name": "parmesan_must"
24             }
25           }
26         ]
27       ],
28       "must_not": [
29         {
30           "match": {
31             "ingredients.name": {
32               "query": "tuna",
33               "_name": "tuna_must_not"
34             }
35           }
36         ],
37       "should": [
38         {
39           "match": {
40             "ingredients.name": {
41               "query": "cheese"
42             }
43           }
44         }
45       ]
46     }
47   }
48 }
```

```
200 - OK 97 ms
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
  "name": "Finely grated Parmesan cheese",
  "quantity": "60g"
},
{
  "name": "Minced flat-leaf parsley leaves",
  "quantity": "Small handful"
},
],
"created": "2017/04/27",
"ratings": [
  1.5,
  2.0,
  4.0,
  3.5,
  3.0,
  5.0,
  1.5
],
"matched_queries": [
  "prep_time_filter",
  "_parsley_shloud",
  "parmesan_must"
],
{
  "_index": "receipt",
  "_type": "_doc",
  "_id": "1",
  "_score": 1.379573,
  "_source": {
    "title": "Fast and Easy Pasta With Blistered Cherry Tomato Sauce"
  }
}
```



Aggregation

History Settings Help

200 - OK 127 ms

```
90  
91 # -----  
92 # AGGREGATION  
93 # -----  
94  
95  
96 GET orders/_search  
97 {  
98   "size": 0,  
99   "aggs": {  
100     "total_sale": {  
101       "sum": {  
102         "field": "total_amount"  
103       }  
104     }  
105   }  
106 }  
107  
108  
109  
110 |
```

```
1 {  
2   "took" : 73,  
3   "timed_out" : false,  
4   "_shards" : {  
5     "total" : 1,  
6     "successful" : 1,  
7     "skipped" : 0,  
8     "failed" : 0  
9   },  
10  "hits" : {  
11    "total" : {  
12      "value" : 1000,  
13      "relation" : "eq"  
14    },  
15    "max_score" : null,  
16    "hits" : [ ]  
17  },  
18  " aggregations" : {  
19    "total_sale" : {  
20      "value" : 109209.60997009277  
21    }  
22  }  
23 }  
24
```





Aggregation

History Settings Help

200 - OK 331 ms

```
96 GET orders/_search
97 {
98   "size": 0,
99   "aggs": {
100     "total_sale": {
101       "sum": {
102         "field": "total_amount"
103       }
104     },
105     "avg_sale": {
106       "avg": {
107         "field": "total_amount"
108       }
109     },
110     "min_sale": {
111       "min": {
112         "field": "total_amount"
113       }
114     },
115     "max_sale": {
116       "max": {
117         "field": "total_amount"
118       }
119     },
120     "boxplot": {
121       "boxplot": {
122         "field": "total_amount"
123       }
124     }
125   }
126 }
```

```
11   "total": {
12     "value": 1000,
13     "relation": "eq"
14   },
15   "max_score": null,
16   "hits": []
17 },
18   " aggregations": {
19     "boxplot": {
20       "min": 10.270000457763672,
21       "max": 281.7699890136719,
22       "q1": 64.65999946594238,
23       "q2": 98.78628550938197,
24       "q3": 148.10366719563802,
25       "lower": 10.270000457763672,
26       "upper": 272.8999938964844
27     },
28     "max_sale": {
29       "value": 281.7699890136719
30     },
31     "avg_sale": {
32       "value": 109.20960997009277
33     },
34     "total_sale": {
35       "value": 109209.60997009277
36     },
37     "min_sale": {
38       "value": 10.270000457763672
39     }
40   }
41 }
```



Aggregation

```
1 GET /recipe/default/_search
2 {
3     "size": 0,
4     "aggs": {
5         "my_filter": {
6             "filters": {
7                 "filters": {
8                     "pasta": {
9                         "match": {
10                            "title": "pasta"
11                        }
12                    },
13                    "spaghetti": {
14                        "match": {
15                            "title": "spaghetti"
16                        }
17                    }
18                }
19            },
20            "ags": {
21                "avg_rating": {
22                    "avg": {
23                        "field": "ratings"
24                    }
25                }
26            }
27        }
28    }
29 }
```

```
1 {
2     "took": 5,
3     "timed_out": false,
4     "_shards": {
5         "total": 5,
6         "successful": 5,
7         "failed": 0
8     },
9     "hits": {
10        "total": 20,
11        "max_score": 0,
12        "hits": []
13    },
14    "aggregations": {
15        "my_filter": {
16            "buckets": {
17                "pasta": {
18                    "doc_count": 8,
19                    "avg_rating": {
20                        "value": 3.3947368421052633
21                    }
22                },
23                "spaghetti": {
24                    "doc_count": 4,
25                    "avg_rating": {
26                        "value": 2.3684210526315788
27                    }
28                }
29            }
30        }
31    }
32 }
```



Aggregation

History Settings Help

200 - OK 83 ms

```
122     "field": "total_amount"
123   }
124 }
125 }
126 }
127
128
129
130 # -----
131 # HISTOGRAM
132 # -----
133
134 GET orders/_search
135 {
136   "size": 0,
137   "aggs": {
138     "amount_distrib": {
139       "histogram": {
140         "field": "total_amount",
141         "interval": 50
142       }
143     }
144   }
145 }
146
147
148
149
150
151
152
```

```
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 1000,
13      "relation" : "eq"
14    },
15    "max_score" : null,
16    "hits" : []
17  },
18  "aggregations" : {
19    "amount_distrib" : {
20      "buckets" : [
21        {
22          "key" : 0.0,
23          "doc_count" : 164
24        },
25        {
26          "key" : 50.0,
27          "doc_count" : 347
28        },
29        {
30          "key" : 100.0,
31          "doc_count" : 249
32        },
33        {
34          "key" : 150.0,
35          "doc_count" : 152
36        }
37      ]
38    }
39  }
```





Data Mining Introduction

Elastic text Analyzer for searching



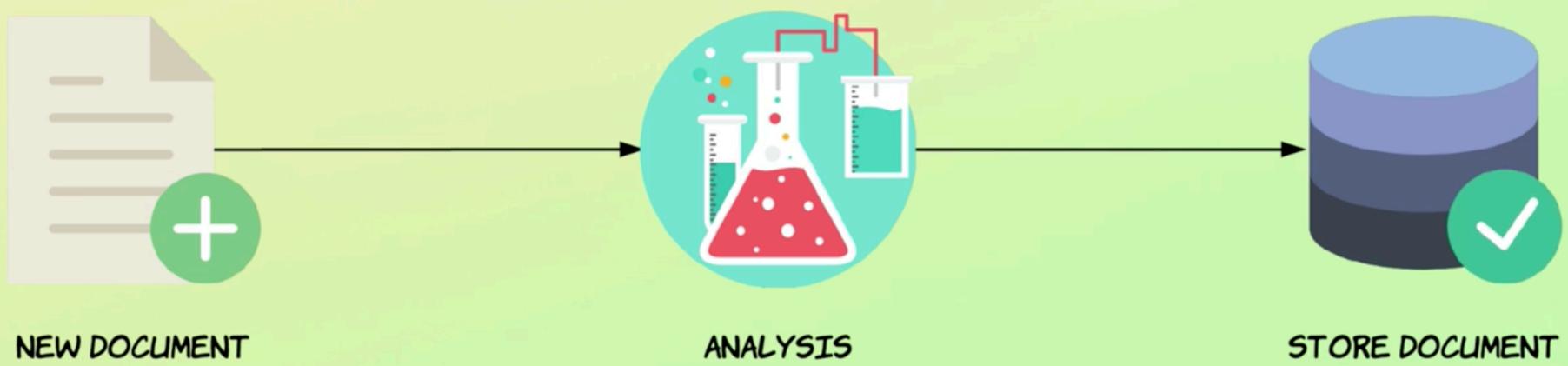
Analyzer KESAKO

- Elasticsearch ce n'est pas qu'une façon de stocker des documents, c'est aussi une façon de les traiter.
- Dans une recherche (recherche dans un index) Elasticsearch qu'on appelle **recherche full-text, est une simple collecte de documents**, qui s'effectue via une comparaison de tokens (unité élémentaire de texte).
- Ces **tokens vivent dans un (reversed index)** index inversé (non accessible directement) et sont **extraits du contenu des documents lors de l'indexation**.
- Plus vos **tokens sont proprement indexés**, et plus facilement un utilisateur trouvera vos documents : c'est le rôle de l'analyser. C'est la **logique derrière les moteurs de recherche**.



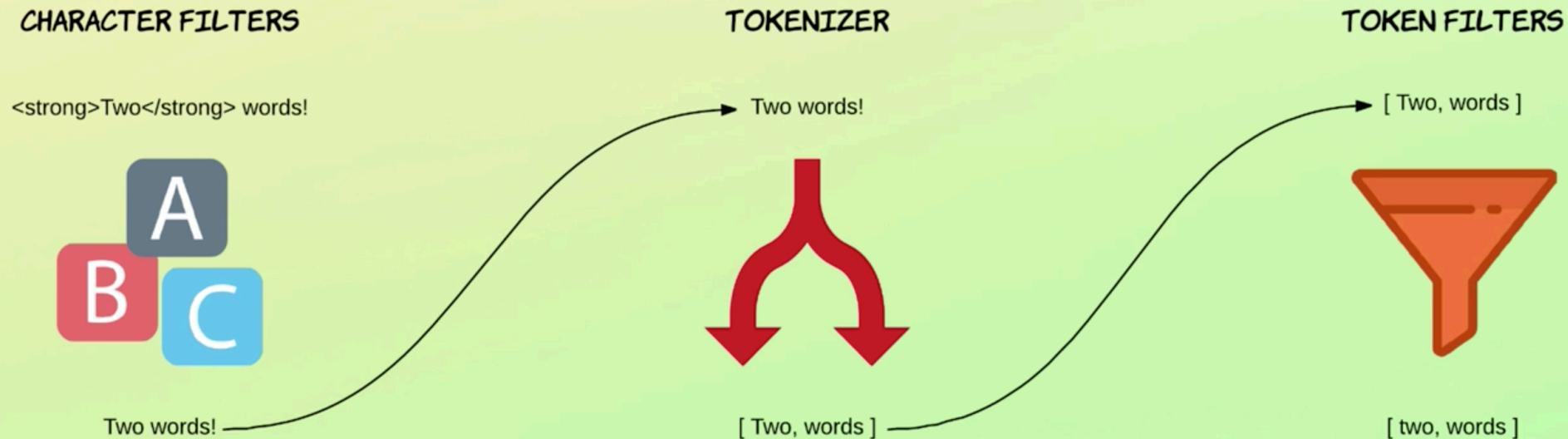


Analyzer processus en image





Comment fonctionne le default analyzer





Analyzer : les notions de bases

Raw

Walking The: Dog

Tokenizer

[‘Walking’, ‘The’, ‘Dog’]

Token Filters

[‘walk’, ‘dog’]





Character filters

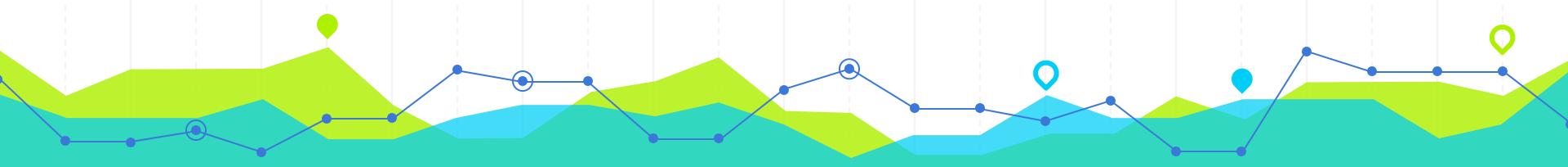
HTML Strip Character Filter (`html_strip`)



Strips out HTML elements like `` and decodes HTML entities like `&`

"*<p>I'm in the mood for drinking semi-dry red wine!</p>*"

"I'm in the mood for drinking semi-dry red wine!"





Character filters

Mapping Character Filter (mapping)



Replaces values based on a map of keys and values.

"I broke my leg_sad_ But luckily I have some red wine around_happy_"



"I broke my leg :-(But luckily I have some red wine around :-)"





Character filters

Pattern Replace (pattern_replace)



Uses a regular expression to match characters and replaces them with the specified replacement. Capture groups may be used.

Pattern: ([a-zA-Z0-9]+) (-?)

Replacement: \$1

"9c559866-d0a5-4dee-84fe-54d878fe1800"

"9c559866d0a54dee84fe54d878fe1800"



Tokenizers types

Break a string into components.

Input: “Walking The: Dog”

Standard: [“Walking”, “The”, “Dog”]

Whitespace: [“Walking”, “The:”, “Dog”]

Edge N-grams: [“W”, “Wa”, “Wal”, “Walk”, ...]





Tokenizers types

UAX URL Email Tokenizer (uax_url_email)



Like the standard tokenizer, but treats URLs and e-mail addresses as single tokens.

*"Contact us at info@codingexplained.com or visit
https://codingexplained.com"*

[Contact, us, at, info@codingexplained.com, or, visit,
https://codingexplained.com]



Tokenizers types

Edge N-Gram Tokenizer (edge_ngram)



Breaks text into words when encountering certain characters and then emits N-grams of each word beginning from the start of the word.

"Red wine"

[Re, Red, wi, win, wine]





Tokenizers types

Path Tokenizer (path_hierarchy)



Splits hierarchical values (e.g. file system paths) and emits a term for each component in the tree.

/path/to/some/directory

[/path, /path/to, /path/to/some, /path/to/some/directory]





Token filters types

Input: [“Walking”, “The”, “Dog”]

Lowercase: [“walking”, “the”, “dog”]

Stop Words: [“walking”, “dog”]

Stemming: [“walk”, “dog”]





Token filters types

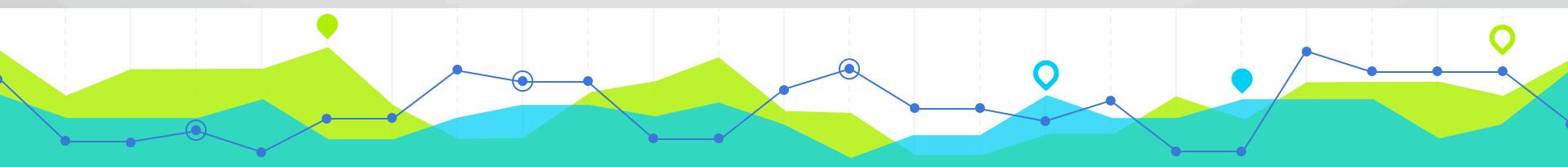
Stop Token Filter (stop)



Removes stop words.

[*I'm, in, the, mood, for, drinking, semi, dry, red, wine*]

[I'm, mood, drinking, semi, dry, red, wine]





Token filters types

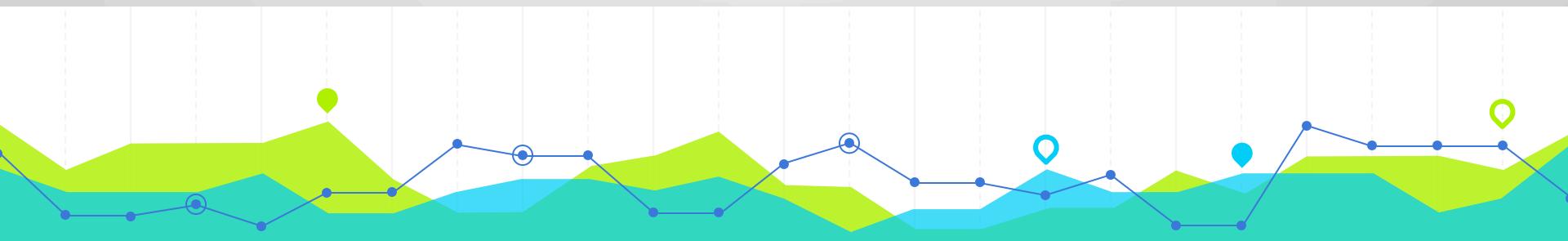
Word Delimiter Token Filter (`word_delimiter`)



Splits words into subwords and performs transformations on subword groups.

[*Wi-Fi, PowerShell, CE1000, Andy's*]

[Wi, Fi, Power, Shell, CE, 1000, Andy]





Token filters types

Snowball Token Filter (snowball)

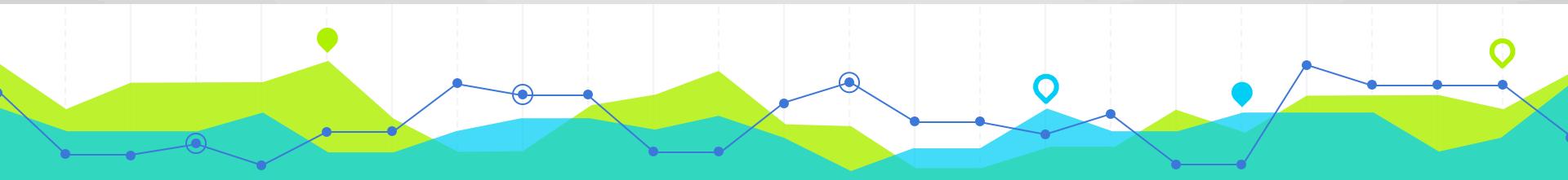


Stems words based on a Snowball algorithm.

[I'm, in, the, mood, for, drinking, semi, dry, red, wine]



[I'm, in, the, mood, for, drink, semi, dry, red, wine]





Token filters types

Synonym Token Filter (synonym)

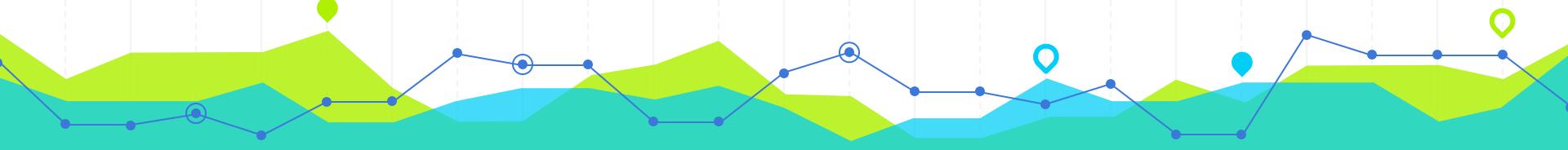


Adds or replaces tokens based on a synonym configuration file.

[I, am, very, happy]



[I, am, very, happy/delighted]





Endpoint _analyze

```
1 POST _analyze
2 {
3     "tokenizer": "standard",
4     "text": "I'm in the mood for drinking semi-dry red wine!"
5 }
6
7 POST _analyze
8 {
9     "filter": [ "lowercase" ],
10    "text": "I'm in the mood for drinking semi-dry red wine!"
11 }
12
13 POST _analyze
14 {
15     "analyzer": "standard",
16     "text": "I'm in the mood for drinking semi-dry red wine!"
17 }
```

```
1 {
2     "tokens": [
3         {
4             "token": "i'm",
5             "start_offset": 0,
6             "end_offset": 3,
7             "type": "<ALPHANUM>",
8             "position": 0
9         },
10        {
11            "token": "in",
12            "start_offset": 4,
13            "end_offset": 6,
14            "type": "<ALPHANUM>",
15            "position": 1
16        },
17        {
18            "token": "the",
19            "start_offset": 7,
20            "end_offset": 10,
21            "type": "<ALPHANUM>",
22            "position": 2
23        },
24        {
25            "token": "mood",
26            "start_offset": 11,
27            "end_offset": 15,
28            "type": "<ALPHANUM>",
29            "position": 3
30        }
31 ]
```



Built in analyzer overview

Simple Analyzer (simple)



Divides text into terms when encountering a character that is not a letter. Also lowercases all terms.

"I'm in the mood for drinking semi-dry red wine!"

[i, m, in, the, mood, for, drinking, semi, dry, red, wine]





Built in analyzer overview

Stop Analyzer (stop)



Like the simple analyzer, but also removes stop words.

"I'm in the mood for drinking semi-dry red wine!"

[i, m, mood, drinking, semi, dry, red, wine]





Built in analyzer overview

Language Analyzers (english, ...)



Language-specific analyzers, e.g. for English or Spanish.

"I'm in the mood for drinking semi-dry red wine!"

[i'm, mood, drink, semi, dry, red, wine]





Build Elastic Analyzer

Customiser un analyzer pour un index



Créer un analyzer pour un index

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/analyzer.html>

```
PUT my-index-000001/_doc/1
{
    "title": "The Quick Brown Fox"
}

PUT my-index-000001/_doc/2
{
    "title": "A Quick Brown Fox"
}

GET my-index-000001/_search
{
    "query": {
        "query_string": {
            "query": "\"the quick brown fox\""⑥
        }
    }
}
```

```
PUT my-index-000001
{
    "settings": {
        "analysis": {
            "analyzer": {
                "my_analyzer": {①
                    "type": "custom",
                    "tokenizer": "standard",
                    "filter": [
                        "lowercase"
                    ]
                },
                "my_stop_analyzer": {②
                    "type": "custom",
                    "tokenizer": "standard",
                    "filter": [
                        "lowercase",
                        "english_stop"
                    ]
                },
                "filter": {
                    "english_stop": {
                        "type": "stop",
                        "stopwords": "_english_"
                    }
                }
            },
            "mappings": {
                "properties": {
                    "title": {
                        "type": "text",
                        "analyzer": "my_analyzer",③
                        "search_analyzer": "my_stop_analyzer",④
                        "search_quote_analyzer": "my_analyzer"⑤
                    }
                }
            }
        }
    }
}
```



Analyzer pour le français from scratch

y Settings Help

```
PUT french
{
  "settings": {
    "analysis": {
      "filter": {
        "french_elision": {
          "type": "elision",
          "articles_case": true,
          "articles": ["l", "m", "t", "qu", "n", "s", "j", "d", "c", "jusqu", "quoiqu", "lorsqu",
                      "puisqu"]
        },
        "french_synonym": {
          "type": "synonym",
          "ignore_case": true,
          "expand": true,
          "synonyms": [
            "salade, laitue",
            "mayo, mayonnaise",
            "grille, toaste"
          ]
        },
        "french_stemmer": {
          "type": "stemmer",
          "language": "light_french"
        }
      },
      "analyzer": {
        "french_heavy": {
          "tokenizer": "icu_tokenizer",
          "filter": [
            "french_elision",
            "icu_folding",
            "french_synonym",
            "french_stemmer"
          ]
        },
        "french_light": {
          "tokenizer": "icu_tokenizer",
          "filter": [
            "french_elision",
            "icu_folding"
          ]
        }
      }
    }
  }
}
```

```
1 #! Deprecation: The ignore_case option on the synonym_graph filter is deprecated. Instead, insert a
lowercase filter in the filter chain before the synonym_graph filter.
2 {
3   "acknowledged" : true,
4   "shards_acknowledged" : true,
5   "index" : "french"
6 }
7
```

200 - OK

1642 ms



Utiliser un custom analyzer

```
GET french/_analyze
{
  "tokenizer": "icu_tokenizer",
  "filter": ["french_synonym", "french_elision", "lowercase"],
  "text" : "je dois bosser pour mon QCM sinon je vais avoir une sale note
  ...
}

GET french/_analyze
{
  "analyzer": "french_light",
  "text" : "je dois bosser pour mon QCM sinon je vais avoir une sale note
  ...
}
```





Utiliser un analyzer dans un _mapping d'index

```
1 PUT /analyzers_test/default/_mapping
2 {
3     "properties": {
4         "description": {
5             "type": "text",
6             "analyzer": "my_analyzer"
7         },
8         "teaser": {
9             "type": "text",
10            "analyzer": "standard"
11        }
12    }
13 }
```





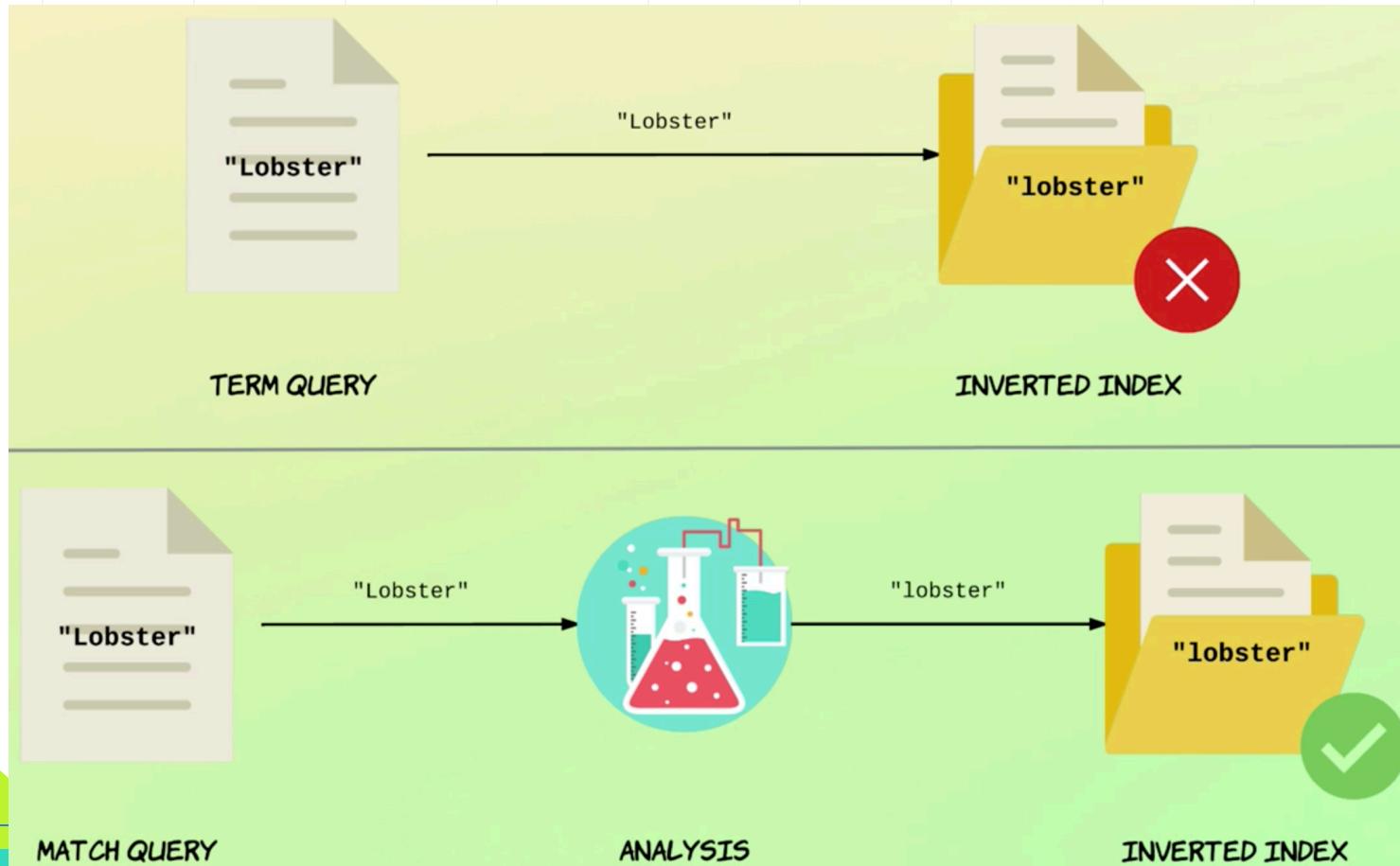
Eviter l'erreur `TOO_MANY_REQUESTS`

- Afin d'éviter une surcharge du cluster on peut le modifier dynamiquement tel que :

```
PUT _cluster/settings
{
  "transient": {
    "cluster.routing.allocation.disk.watermark.low": "1gb",
    "cluster.routing.allocation.disk.watermark.high": "1gb",
    "cluster.routing.allocation.disk.watermark.flood_stage": "1gb",
    "cluster.info.update.interval": "1m"
  }
}
```



La différence entre les term query et les match query : l'analyseur





Les inverted index en image

Term	Document #1	Document #2
best	X	
carbonara		X
delicious		X
pasta	X	X
pesto	X	
recipe	X	X
the	X	
with	X	



Python API *Overview*



Python Elastic Client

GET request to
localhost:9200/products_index/product

```
{  
  "query": {  
    "match": {  
      "name": "necklace"  
    }  
  }  
}
```

```
from elasticsearch import Elasticsearch  
from elasticsearch_dsl import Search  
  
client = Elasticsearch()  
s = Search(  
  using=client,  
  index="products_index",  
  doc_type="products"  
)  
s.query(  
  "match",  
  name="necklace"  
)execute()
```



Python Elastic Client : Query

GET request to
localhost:9200/products_index/products

```
{  
  "query": {  
    "match": {  
      "name": "necklace"  
    }  
  }  
}
```

```
from elasticsearch import Elasticsearch  
from elasticsearch_dsl import Search  
  
client = Elasticsearch()  
s = Search(  
    using=client,  
    index="products_index",  
    doc_type="products"  
)  
s.query(  
    "match",  
    name="necklace"  
)  
.execute()
```



Contents ⚙

- 1 Elasticsearch python API overview
 - 1.1 Avant de commencer
 - 1.1.1 Lancer elasticsearch avec docker
 - 1.1.2 Lancer elasticsearch avec docker-compose
 - 1.1.2.1 🚧Attention à votre configuration [Exercice [optionnel]]
 - 1.1.3 🖥 Exercice [optionnel]
 - 1.2 Ping du container
 - 1.3 Create, delete and verify index
 - 1.4 Insert documents
 - 1.4.1 🖥 Exercice [optionnel]
 - 1.4.2 Retrieve data with id : get
 - 1.4.3 🖥 Exercice [optionnel]
 - 1.4.4 Mapping
 - 1.5 Le endpoint _search et les query
 - 1.5.1 🖥 Exercice [optionnel]
 - 1.5.2 Affiner ces critères de recherche avec _source
 - 1.5.3 Logique booléenne
 - 1.5.4 Les critères : SHOULD / MUST
 - 1.5.5 Filtrer ses query avec filter
 - 1.5.6 Recherche avec un prefix
 - 1.5.7 Rechercher avec des regex
 - 1.5.8 Aggregation
 - 1.5.9 🖥 Exercice [optionnel]

1 Elasticsearch python API overview

Entrée [1]:

```
import warnings
from elasticsearch import Elasticsearch, RequestsHttpConnection
warnings.filterwarnings('ignore')
```

1.1 Avant de commencer

1.1.1 Lancer elasticsearch avec docker

Pour ce faire, on va run un cluster elastic dans un container. Si vous n'avez pas déjà l'image elastic dans votre registry local il faut la pull du hub avec la commande suivante:

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:7.11.1
```

puis on run le container sur le port 9200 tel que:

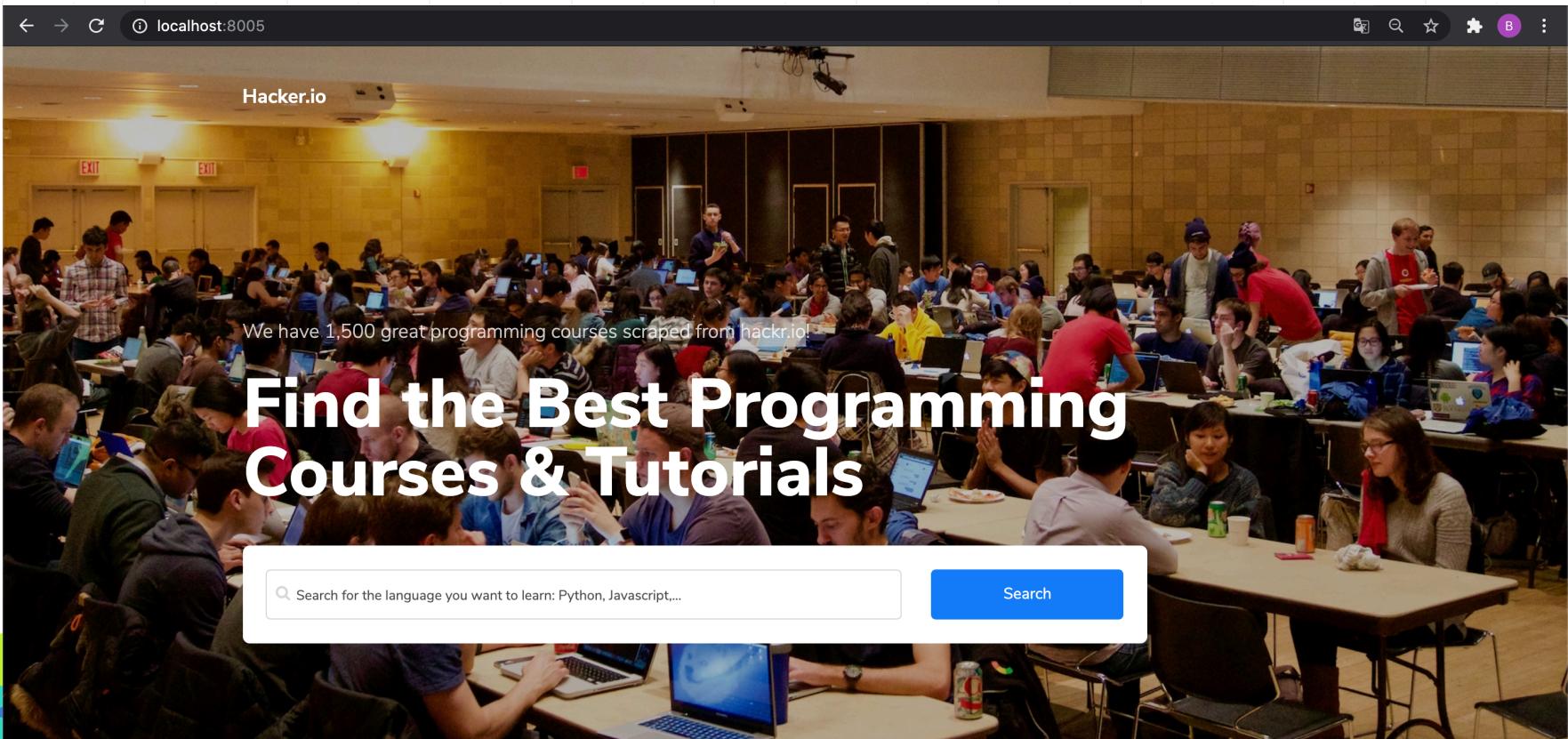
```
docker run -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.11.1
```

1.1.2 Lancer elasticsearch avec docker-compose



Projet : moteur de recherche avec flask et elasticsearch

- <https://github.com/dineshsonachalam/Building-a-search-engine-using-Elasticsearch>





En pratique

Kibana

D Kibana : Discover

(créer un index pattern avant)

Discover

New Save Open Share Inspect

Search

+ Add filter

movies*

Search field names

Filter by type 0

Available fields 17

- _id
- _index
- #_score
- _type
- fields.actors
- fields.directors
- fields.genres
- fields.image_url
- fields.plot
- fields.rank
- fields.rating
- fields.release_date
- # fields.running_time_secs
- fields.title
- # fields.year
- id
- type

2,437 hits Dec 21, 2005 @ 00:00:00.000 - Dec 21, 2018 @ 00:00:00.000

Count fields.release_date per 30 days

Absolute December 2018 Now

00:00 00:30 01:00 01:30 02:00 02:30 03:00 03:30 04:00 04:30 05:00

SU MO TU WE TH FR SA

25 26 27 28 29 30 1

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30 31 1 2 3 4 5

Hide chart

Time _source

> Jan 1, 2017 @ 01:00:00.000 fields.release_date: Jan 1, 2017 @ 01:00:00.000 fields.genres: Action heroes are assembled to form the Justice League, to combat a threat beyond each individual's capabilities. fields.title: Justice League fields.rank: 1,295 fields.year: 2,017 id: tt0974015 type: add _id: 1295 _type: movie _index: movies _score: -

> Dec 1, 2016 @ 01:00:00.000 fields.directors: James Cameron fields.release_date: Dec 1, 2016 @ 01:00:00.000 fields.genres: Action, Adventure, Fantasy, Sci-Fi fields.plot: A sequel to the 2009 global phenomenon. fields.title: Avatar 2 fields.rank: 909 fields.year: 2,016 fields.actors: Zoe Saldana, Sigourney Weaver, Sam Worthington id: tt1630029 type: add _id: 909 _type: movie _index: movies _score: -

> Jan 1, 2016 @ 01:00:00.000 fields.directors: Greg Berlanti fields.release_date: Jan 1, 2016 @ 01:00:00.000 fields.genres: Action, Adventure, Drama, Fantasy, Sci-Fi fields.title: The Flash fields.rank: 2,281 fields.year: 2,016 id: tt0439572 type: add _id: 2281 _type: movie _index: movies _score: -

> Dec 23, 2015 @ 01:00:00.000 fields.directors: Jennifer Yuh fields.release_date: Dec 23, 2015 @ 01:00:00.000 fields.genres: Animation, Action, Adventure, Comedy, Family fields.title: Kung Fu Panda 3 fields.rank: 1,530 fields.actors: Bryan Cranston, Angelina Jolie, Rebel Wilson fields.year: 2,015 id: tt2267968 type: add _id: 1530 _type: movie _index: movies _score: -

End date Dec 21, 2018 @ 00:00:00.000



Créer un index pattern : visualiser les data de elastic dans kibana

localhost:5601/app/management/kibana/indexPatterns/patterns/a6123e20-757c-11eb-acc5-b1bbe01eeb04#/?_a=(tab:indexedFields)

elastic Search Elastic

Stack Management / Index patterns / movies*

Ingest ⓘ

Ingest Node Pipelines

Data ⓘ

Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters

Alerts and Insights ⓘ

Alerts and Actions
Reporting

Kibana ⓘ

Index Patterns

Saved Objects
Tags
Spaces
Advanced Settings

Stack ⓘ

License Management
8.0 Upgrade Assistant

movies*

Time field: 'fields.release_date'

This page lists every field in the **movies*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#)

Fields (26) Scripted fields (0) Field filters (0)

Search All field types

Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		•	•	
_index	string		•	•	
_score	number				
_source	_source				
_type	string		•	•	
fields.actors	string		•		
fields.actors.keyword	string		•	•	
fields.directors	string		•		
fields.duration	time		•	•	
fields.duration.keyword	string		•	•	
fields.popularity	float		•	•	
fields.popularity.keyword	string		•	•	
fields.release_date	date	epoch_millis	•	•	
fields.release_date.keyword	string		•	•	
fields.title	string		•	•	
fields.title.keyword	string		•	•	
fields.vote_average	float		•	•	
fields.vote_average.keyword	string		•	•	
fields.vote_count	integer		•	•	
fields.vote_count.keyword	string		•	•	



Créer une visualisation dans kibana

☰ D Visualize / Create

Inspect Share Save

Search

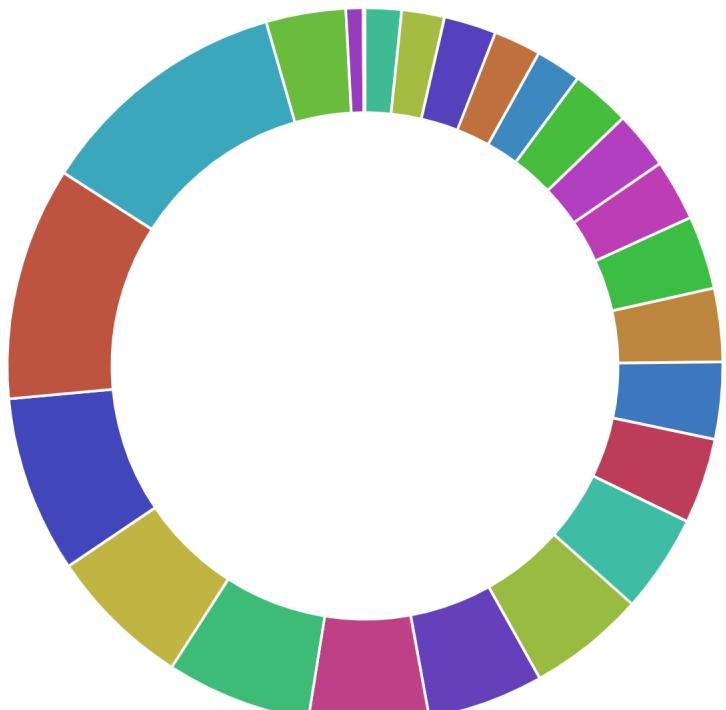
KQL

⌚

Feb 24, 1993 @ 23:30:00.317 → now

⟳ Refresh

⟳ + Add filter



- 1993
- 1994
- 1995
- 1996
- 1997
- 1998
- 1999
- 2000
- 2001
- 2002
- 2003
- 2004
- 2005
- 2006
- 2007
- 2008
- 2009
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015
- 2016
- 2017

movies*

Data Options

Split slices

Aggregation

Date Histogram help

Date Histogram

Field

fields.release_date

Minimum interval

Year

Select an option or create a custom value. Examples: 30s, 20m, 24h, 2d, 1w, 1M

Drop partial buckets

Custom label

> Advanced

+ Add

X Discard

▷ Update



CI/CD

Introduction aux concepts de CI/CD



Comprendre les enjeux de l'intégration continue et du déploiement

- CI/CD est l'acronyme de Continuous Integration et **Continuous Delivery ou Deployment**

Continuous Integration

CI est une pratique de développement où pour chaque modification faite par des développeurs, leur code est mergé, buildé et les tests sont automatiquement exécutés et un statut leur est donné.

Continuous Delivery ou Deployment

CD est une pratique de développement où les changements sont automatiquement buildés, testés et l'application est préparé pour la partie production, ou jusqu'à la mise en production.

- Le Continuous Delivery le produit développé est packagé et mis à disposition
- Le Continuous Deployment va plus loin car c'est l'activité qui met à jour le package en production.



CD en image

CONTINUOUS DELIVERY

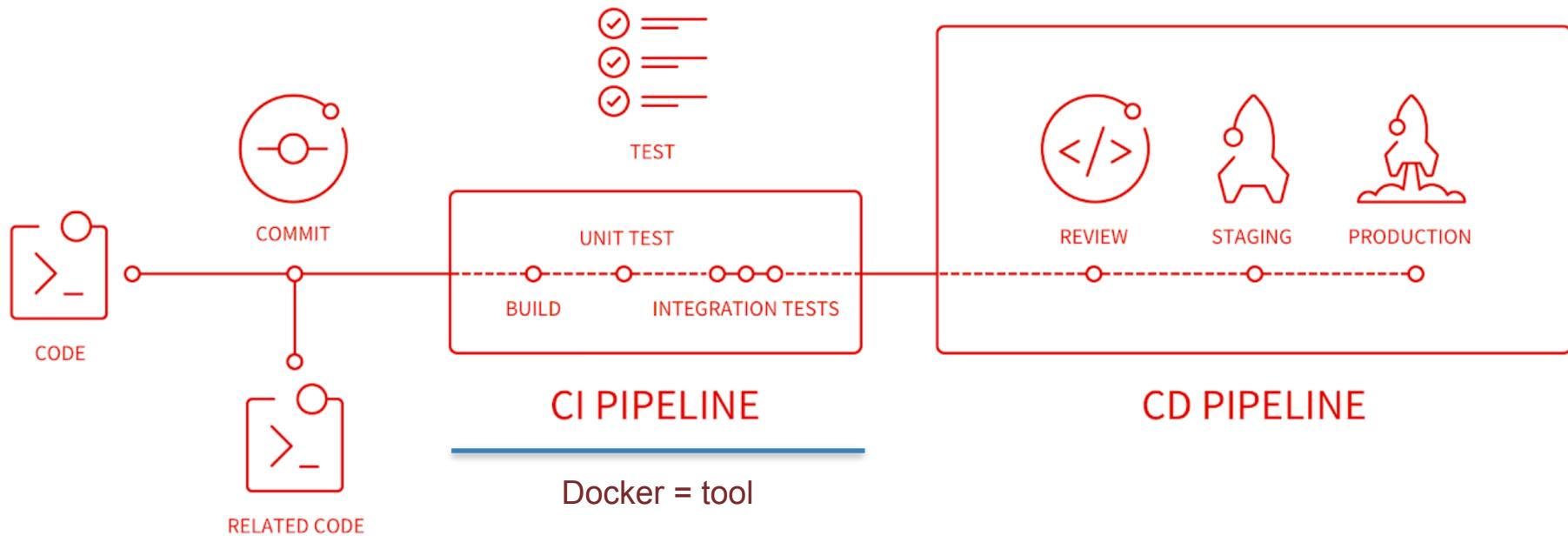


CONTINUOUS DEPLOYMENT





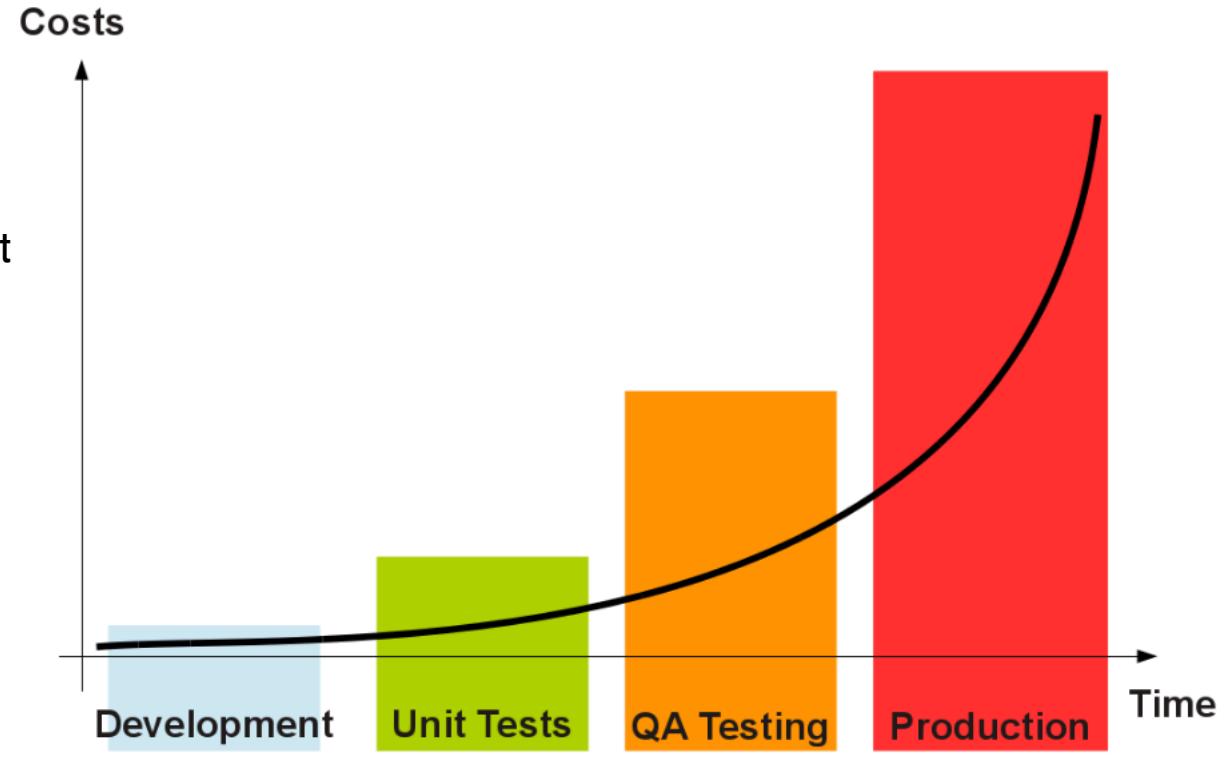
CD en image





Pourquoi ?

- Déetecter les erreurs au plus tôt
- Economiser de l'argent





La méthodologie devops



Cycle en V



Agilité



DevOps

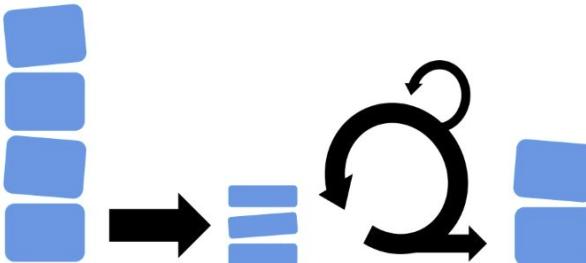


Comment faire du CI/CD

Coté méthodologique

Avant de développer, il faut **planifier les modifications** que l'on souhaite !

CI/CD s'accompagne souvent de la **méthodologie Scrum**. Car les dévs doivent intégrer quotidiennement leurs modifications



Coté outils

Lors du développement, il faut:

1. Un **SCM (Source Control Manager)** comme Git, SVN, RTC, Mercurial, ...
2. Un “**maître d’orchestre**” pour gérer les étapes de la pipeline CI/CD comme Gitlab CI, Jenkins, Travis, Circle CI, ...
3. Un **système de rédaction des tests**
4. Un **paquetage intelligent type docker**





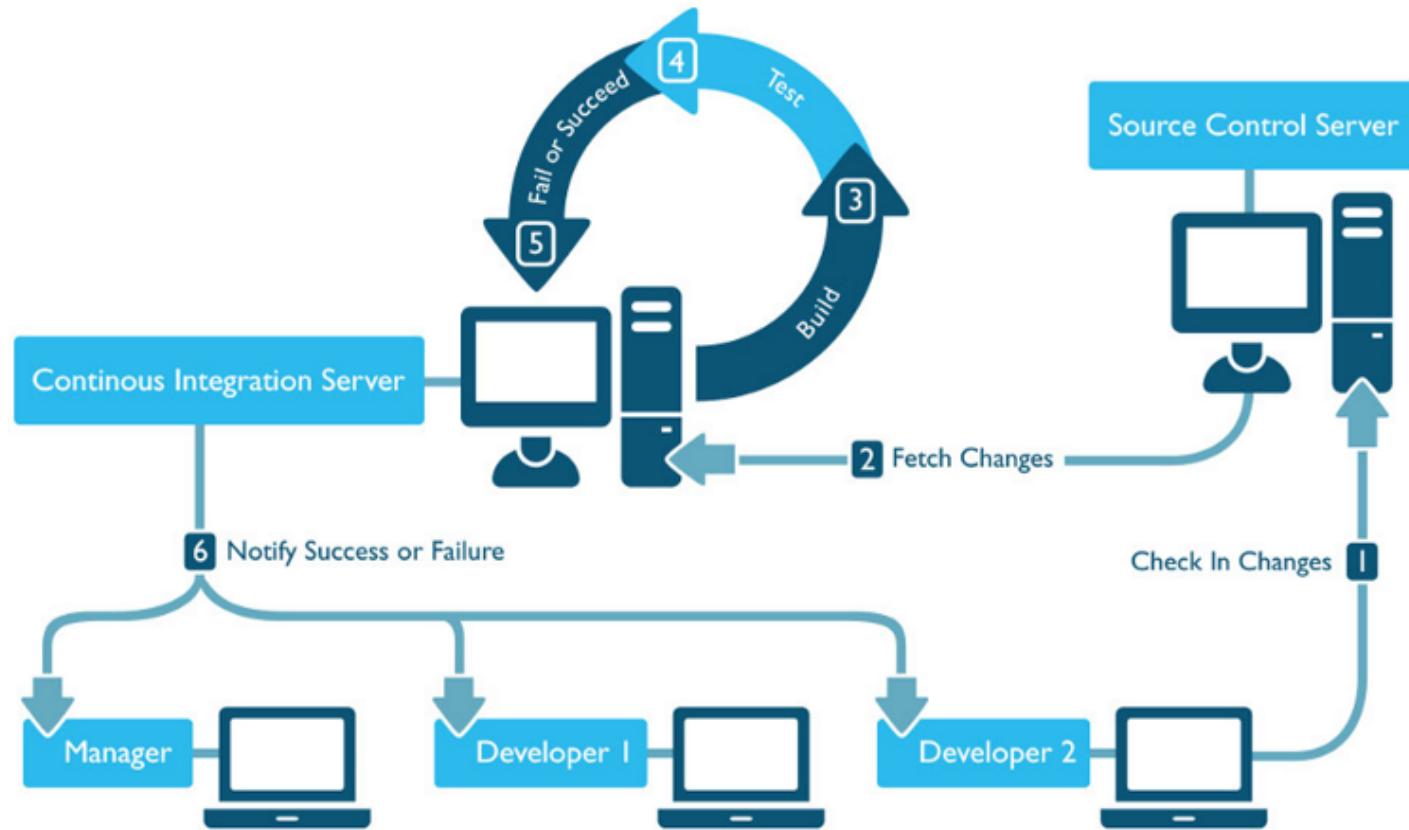
CI/CD les principaux avantages

- toutes les modifications sont immédiatement testées
- les équipes sont rapidement notifiées en cas de code incompatible ou d'erreurs
- les problèmes d'intégration sont détectés et réparés de façon continue, évitant “l'intégration Hell” ainsi que des problèmes de dernière minute
- L'automatisation n'introduit pas d'erreurs humaines, due à la répétabilité ou d'inattention
- une version est toujours disponible pour un test, une démonstration ou une distribution.





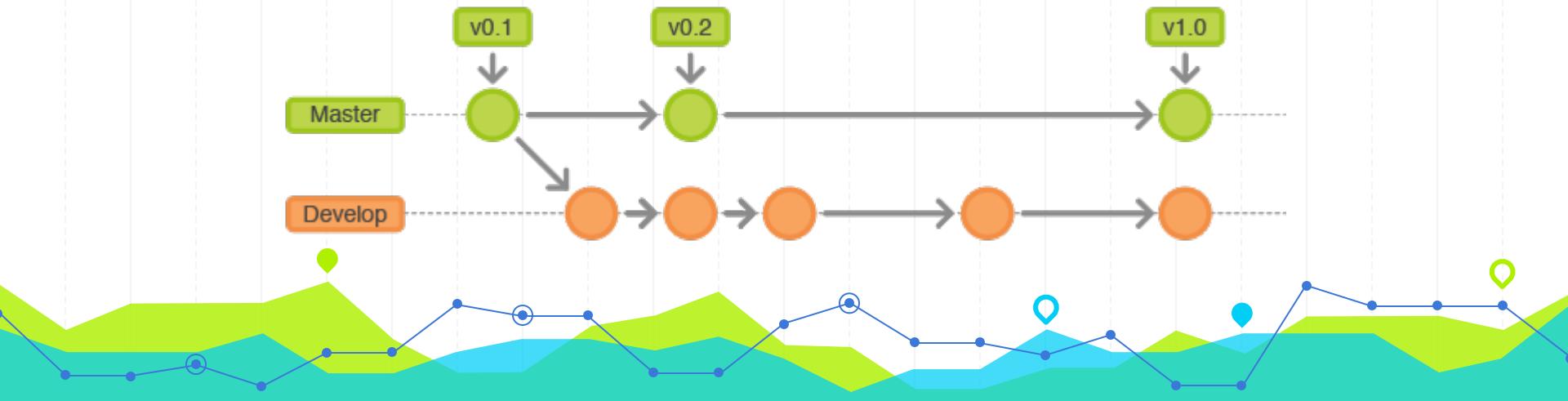
CI Workflow





Le cas général de la CI : déclenché via le SCM

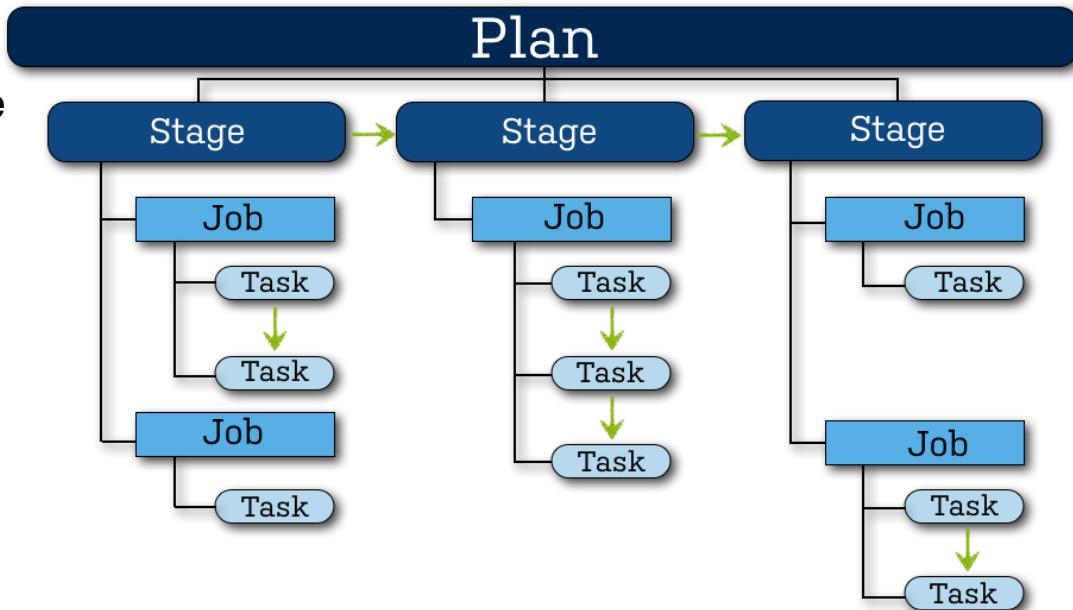
- Un développement est une suite successive de modifications, afin de travailler de manière collaborative le code est stocké sur un serveur qui s'assure de conserver toutes les modifications.
- Dans le cas d'un pipeline, les modifications qui sont envoyées par les développeurs, déclenchent une cascade d'actions à commencer par une notification à l'outil de CI.





Déclenchement de la CI

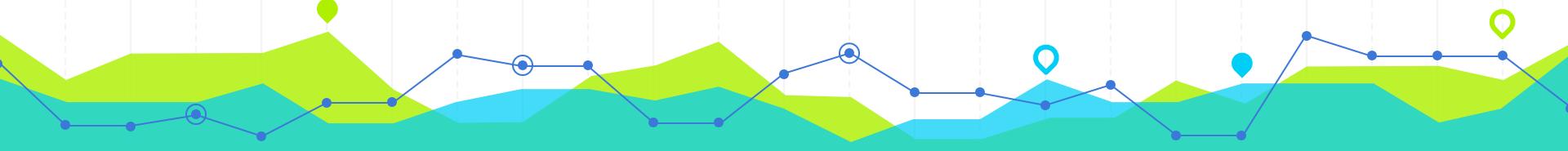
- Lorsque un commit est envoyé par un développeur, l'outil de CI va être notifié (ou périodiquement vérifier la présence ou non d'un commit).
- Lorsqu'une modification est détectée, alors il exécute le “plan”.
- Celui ci exécute les “stages” de manière séquentielle, à l'intérieur desquels, il exécute des “jobs” (de manière parallèle) et des “tasks” (séquentiellement).





Build avec la pipeline

- La pipeline va récupérer (avec un git clone ou équivalent) le contenu du répo. Maintenant que la pipeline possède la dernière version du code il faut build l'application.
- Mais pour build et run l'application faut donc créer l'environnement avec les versions et les dépendances requises, d'où l'importance d'un outil de paquetage type docker.
- Avec cette étape de build, nous pouvons être certains qu'à chaque instant le code présent sur le répo est compilable... ce qui ne veut pas dire qu'il fonctionne comme prévu !





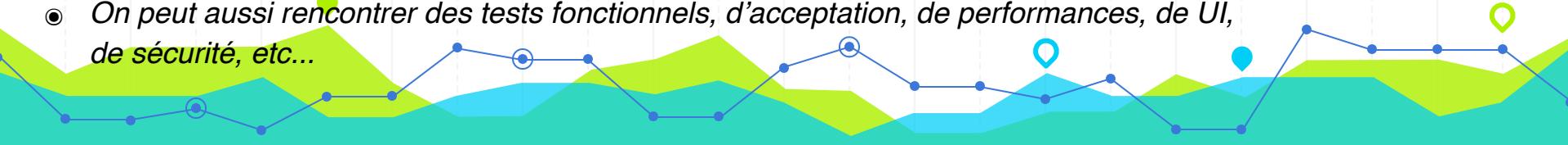
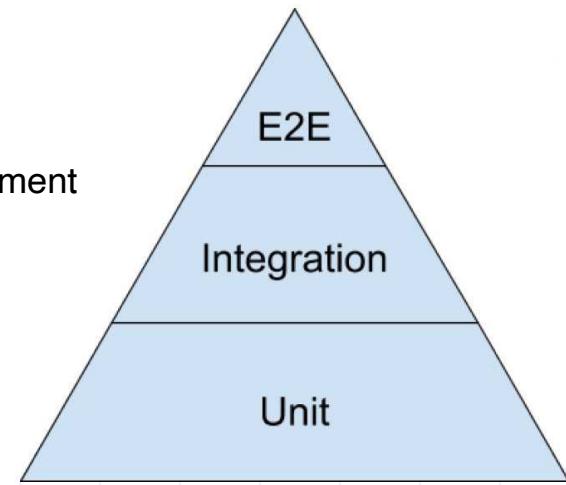
Les tests

C'est quoi ? Quels sont les différents types de tests, et dans quel cadre sont-ils utilisés ?



Les différents tests

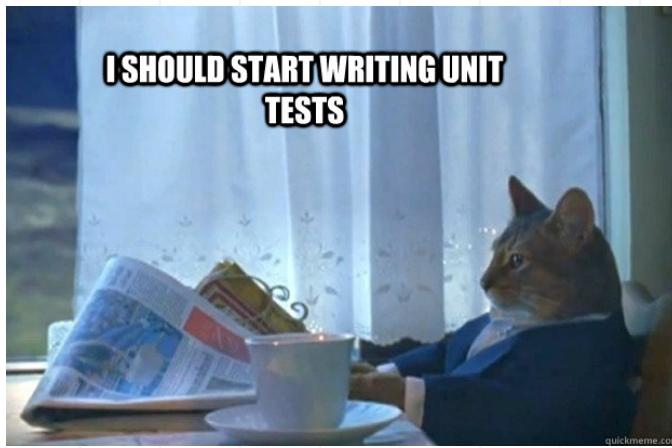
- **Les tests unitaires:** c'est des tests rapides, pas chers, très simples n'ayant pas de dépendance, on peut rapidement :
 - trouver les erreurs car scope très petit
 - sécuriser la maintenance (signalent une régression)
 - aident à la documentation du code (en complément)
- **Les tests d'intégrations:** c'est des tests moyennement rapides et moyennement complexes:
 - permet de sécuriser une partie importante d'une fonctionnalité
 - permet de vérifier l'interaction entre plusieurs composants
- **Les tests End-2-End:** c'est des tests souvent longs, et coûteux
 - permettant de tester une très grosse portion de l'application
- *On peut aussi rencontrer des tests fonctionnels, d'acceptation, de performances, de UI, de sécurité, etc...*





Les tests unitaires (outil : pytest)

- Un test unitaire teste ... une unité de code. Autrement dit, une très petite portion qui peut varier selon le cas de figure (une méthode, une classe)



```
# Fichier de code
def fonction_a_tester(param1, param2):
    return param1 + param2

# Fichier de test
from fichier_de_code import fonction_a_tester

assert fonction_a_tester(1, 1) == 2 # test de l'addition
assert fonction_a_tester(1, -1) == 0 # test avec chiffre négatif
assert fonction_a_tester(4, 2) == 6 # test avec autre chose que des 1
assert fonction_a_tester(4.5, 2) == 6.5 # test avec des floats
```

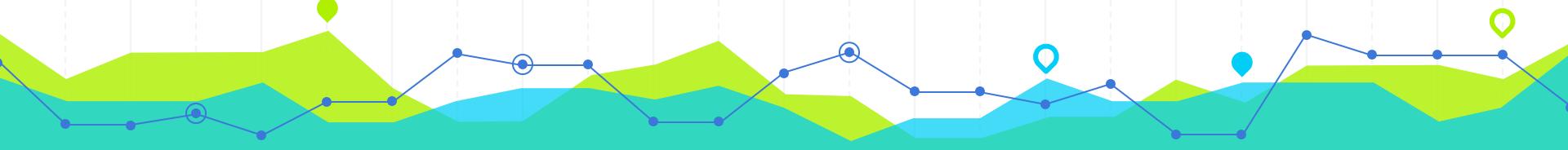




Les tests unitaires

- **Plusieurs constats :**

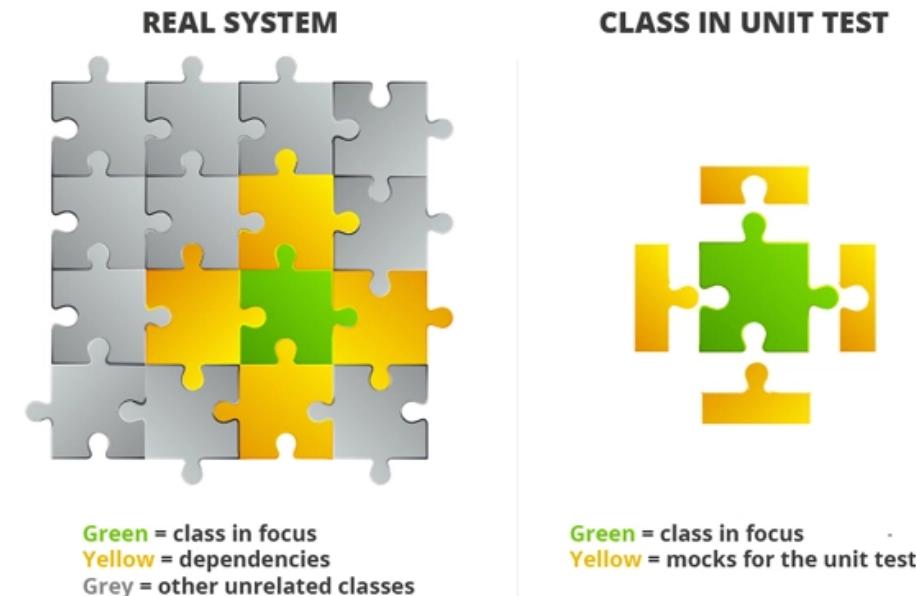
- C'est plutôt simple à écrire, donc plutôt rapide...
- On teste le même code plusieurs fois, avec plusieurs cas de figure, pour s'assurer que le code fonctionne dans tous les cas prévus
- C'est extrêmement rébarbatif et les tests unitaires sont dans 99% des cas, ennuyeux, mais ce sera un filet de sécurité pour la suite !





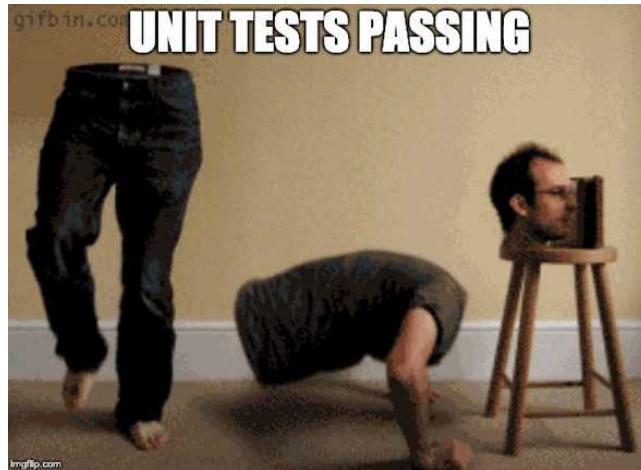
Les tests d'intégration

- Un test d'intégration teste... l'intégration entre deux fonctions, classes.
- Autrement dit, on vérifie la bonne coopération entre plusieurs modules de l'application.
- C'est souvent un test plus large, plus complexe et plus coûteux d'un test unitaire.
- L'objectif est donc de faire interagir plusieurs modules de l'application entre eux afin de voir leur bonne coopération il faut créer des "stub" (bouchons) pour simuler le monde extérieur.





Les bonnes pratiques de l'intégration





Les tests end-2-end

- Un test End-2-End teste... de bout en bout ! L'objectif est de vérifier si une application (web, mobile...) se comporte comme prévu, du début à la fin.
- Ils consistent à vérifier que l'utilisateur final puisse achever les principaux scénarios d'utilisation de l'application. Généralement, dans le cas d'applications webs, on le couple à Selenium.
- Bien qu'ils permettent de valider les scénarios utilisateurs, notons qu'ils ne remplacent pas l'utilisateur final.

The screenshot shows the Google sign-in page with the URL `john.doe@screenster.io` entered into the 'Email or phone' input field. A red box highlights this input field. Below the input fields are links for 'Forgot email?' and 'More options'. A blue 'NEXT' button is at the bottom right.

Action 1: Sign in with a valid email

Action 2: Click Next

The screenshot shows the Google password entry page after a successful sign-in. It displays the greeting 'Hi John' and the email address 'john.doe@screenster.io'. Below is an 'Enter your password' input field, which is also highlighted with a red box. A blue 'NEXT' button is at the bottom right.

Action 3: Input valid password

Action 4: Click Next



Les différents outils de test

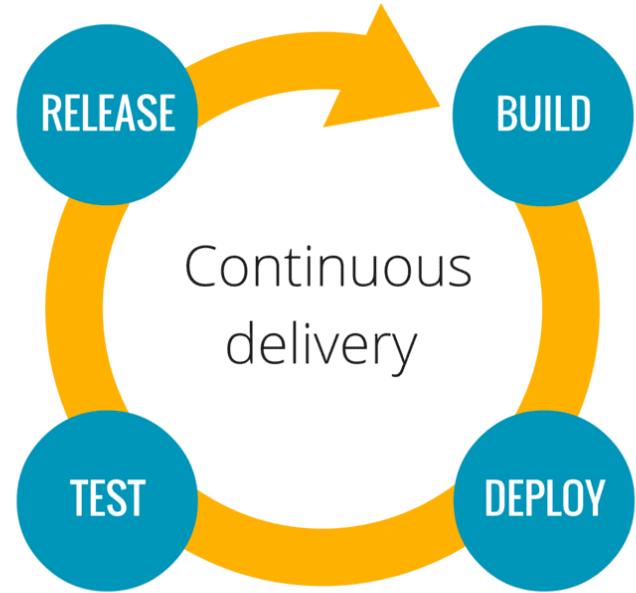
- Il existe une multitude d'outils, de frameworks, qui permettent de réaliser des tests automatiques.
 - JUnit en Java
 - PHPUnit pour Php
 - JSUnit pour JavaScript
 - Pytest pour Python
 - SUnit pour Smalltalk
 - ...
- Évidemment le choix de votre outils dépendra de votre langage de programmation.





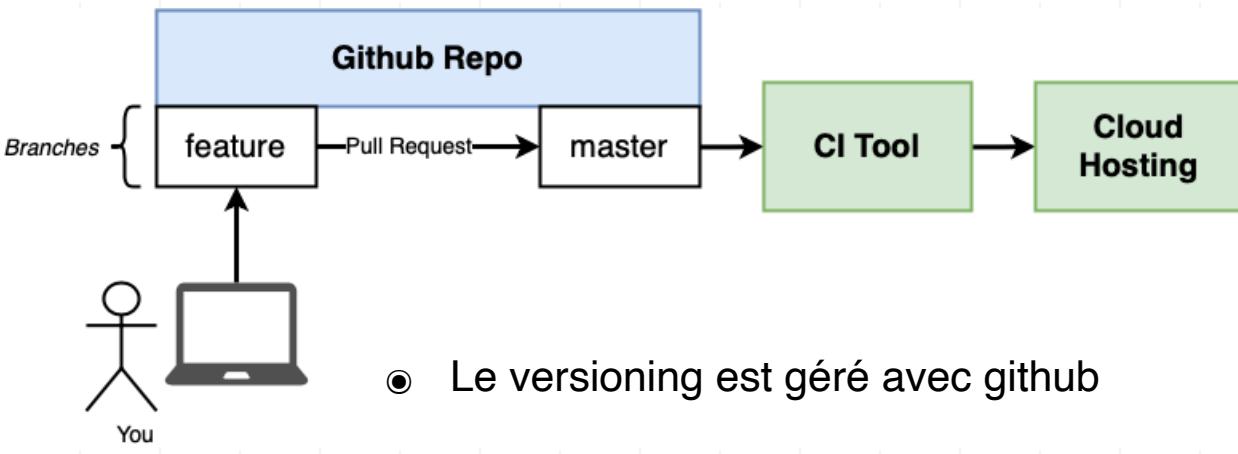
Introduction à la CD avec docker-compose

- En effet docker-compose n'est pas qu'un outil de build il peut aussi effectuer du testing automatique.
- Il facilite ensuite les opérations de deployment et de release dans le cloud (GCP avec GKE par exemple)

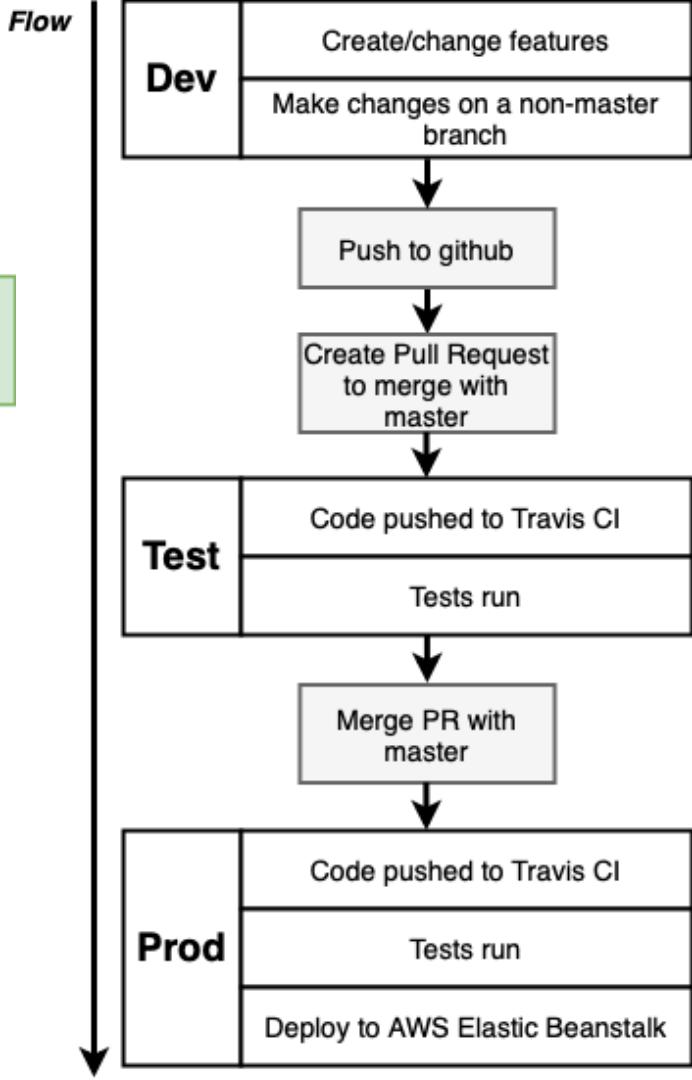
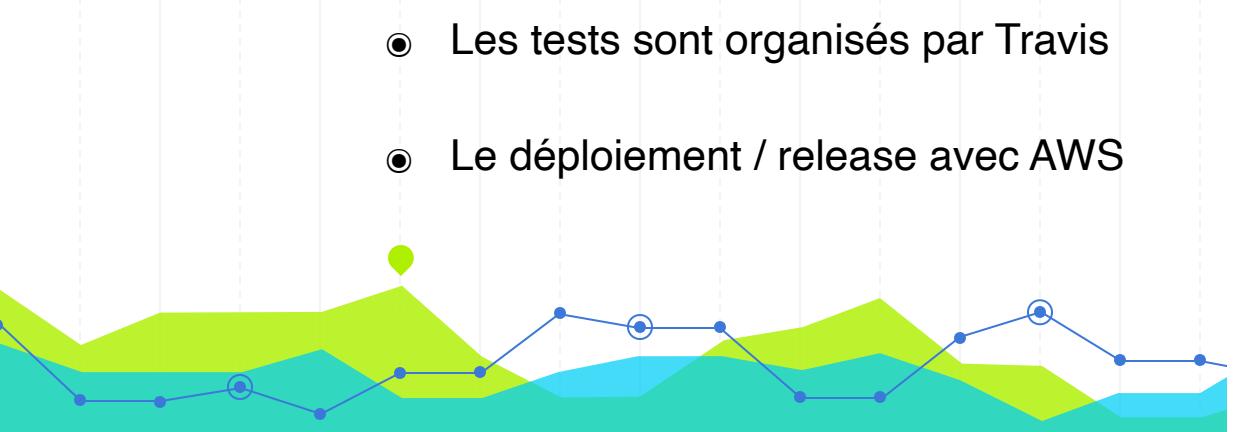




CD - Workflow exemple



- Le versioning est géré avec github
- Les tests sont organisés par Travis
- Le déploiement / release avec AWS





Ressources

- <https://docs.docker.com>
- <https://www.sqlalchemy.org>
- <https://www.elastic.co/guide/index.html>
- <https://elasticsearch-py.readthedocs.io/>
- <http://b3d.bdpedia.fr>





Merci à vous

++