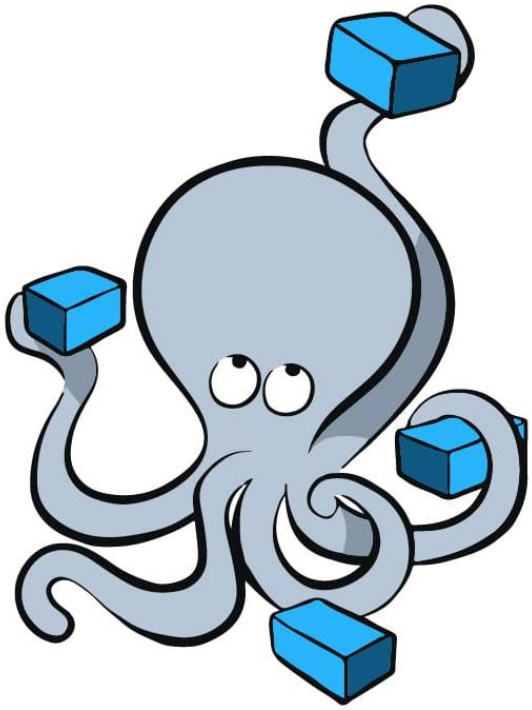
Docker Compose et Volume



docker

Compose et Volume

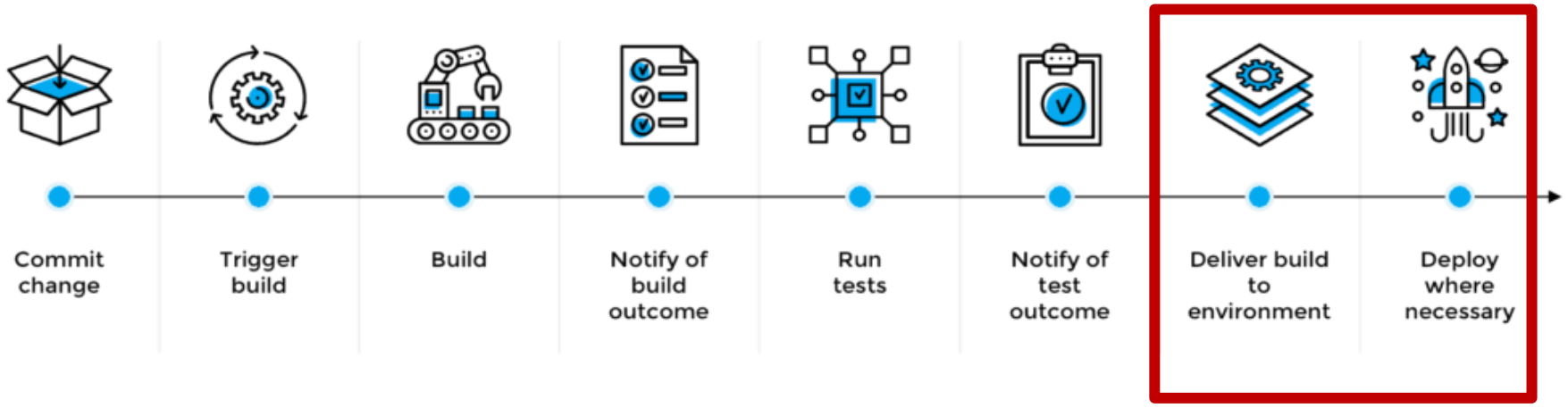
Bureau E204

Plan du cours

- Introduction
- Docker
- Docker Compose
- Docker Volume
- Docker et Jenkins

Introduction

- Notre application Spring Boot codé, compilé et testé (fonctionnellement et qualitativement) doit être intégrée dans une chaine devOps complète (CI/CD).
- La chaine d'intégration (CI) continue a été réalisée grâce à Jenkins via la création d'une pipeline automatisée déclenchée lors d'une détection d'un push dans le référentiel du code.
- Dans ce cours, on va s'intéresser **à la chaine CD (Continuos delivery and deployment)**



Introduction

- Qu'est ce qu'une livraison continue ?
- Qu'est ce qu'on doit livrer ?
- Où dois-je livrer le livrable ?
- Quelle est la différence entre la livraison continue et le déploiement continu ?

Introduction

- L'objectif de la partie CD (déploiement et livraison continu) est de placer notre application dans une machine de production et d'assurer son fonctionnement (Communication avec la base de données, web services fonctionnels, etc..)

- La machine de production peut être:
  - ✓ Une machine physique
  - ✓ Une machine virtuelle
  - ✓ Une image Docker

Introduction

Nous avons vu que nous pouvons isoler chaque application à l'intérieur d'une image où nous pouvons définir son environnement dans un Dockerfile.

Puis, avec un simple "docker build" et "docker run", notre application sera accessible via le port que nous avons exposé:

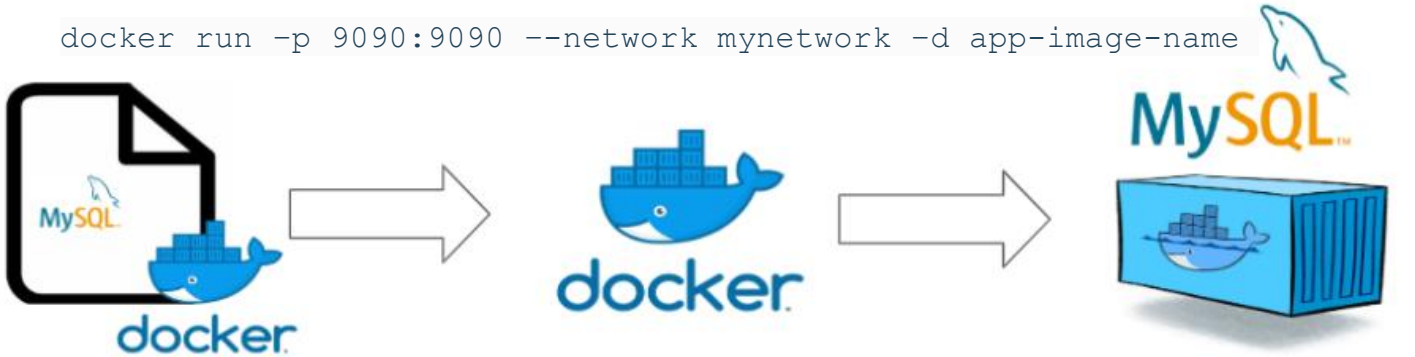
- docker build –t <image_name> .
- docker run –p 8080:8080 <image_name>

Introduction

Chaque application a besoin de connecter à un serveur base de données.

→ Pour que ces deux-là puissent communiquer, il faut les mettre sous le même réseau et lancer la base de données avant le démarrage de l'application.

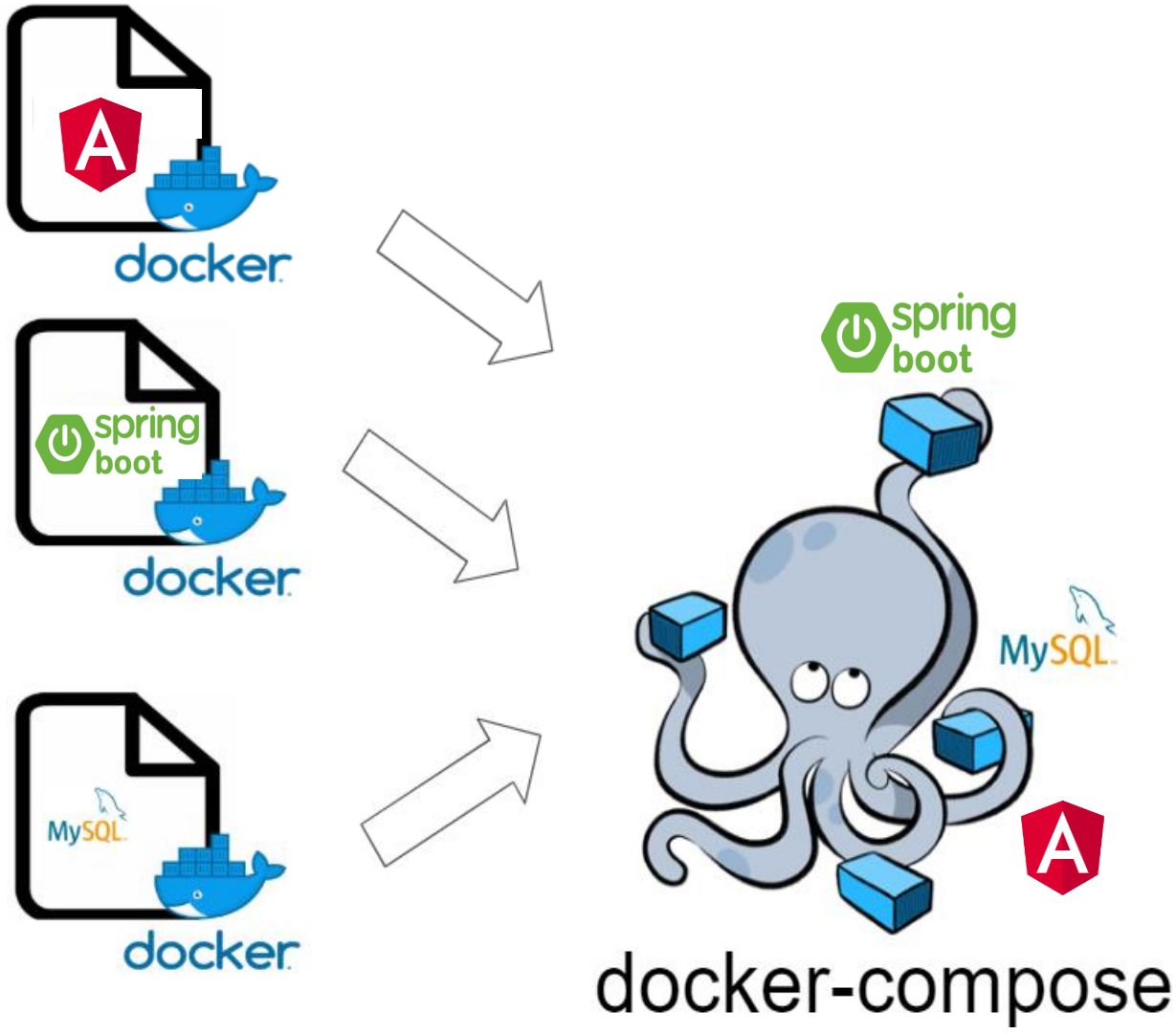

```
docker run -p 9090:9090 --network mynetwork -d app-image-name
```




```
docker run --name mysqldb --network mynetwork -e MYSQL_ROOT_PASSWORD=my-secret-pw -v
/home/mysql/data:/var/lib/mysql -d mysql:8
```

Introduction

→ Et là, il nous faut docker compose.



Docker Compose

- Docker Compose est un outil permettant de définir et d'exécuter des applications Docker multi-conteneurs.
- Dans cette logique, chaque partie de l'application (code, base de données, serveur web, ...) sera hébergée par un conteneur.
- Cet outil repose sur le langage YAML pour décrire l'architecture physique de l'application.
- Le fichier Compose comporte la **version** (OBSOLÈTE), les **services** (REQUIS), les **réseaux**, les **volumes**, les **configurations** et les **secrets**.
- Après la configuration du fichier YAML, il suffit d'exécuter une seule commande pour créer et démarrer tous les services.

Docker Compose

- L'utilisation de Docker Compose se résume à un processus en trois étapes :

1. Définir l'environnement de votre application à l'aide d'un « DockerFile » afin qu'il puisse être reproduit partout.
2. Définir les services qui composent votre application dans « Docker-compose.yml » afin qu'ils puissent être exécutés ensemble dans un environnement isolé.
3. Exécuter la commande « docker compose up » pour lancer votre application entière.

Docker Compose - Exemple

docker-compose.yml

```yaml
# docker-compose.yml
version: '2'
services:
  image_node:
    image: node:12
    working_dir: /app
    ports:
    - '8080:8081'
```

La version du format de fichier Compose (1,2 ou 3)

Définir les images à exécuter simultanément

Le nom de base de l'image (node.js 12.x)

Répertoire de travail des commandes

Exposer le port du conteneur à l'hôte

La définition d'un conteneur

Docker Compose - Les 3 fonctions principales

Les 3 fonctions principales de docker-compose sont :

- Comment lancer un docker-compose ? (se mettre dans le dossier contenant le fichier docker-compose.yml) :

  docker-compose up -d

- Comment vérifier les logs des conteneurs qui ont été lancé ?

  docker-compose logs

- Comment arrêter un docker compose ?

  docker-compose down

Docker Compose – Installation

- Installer le plugin Docker Compose:
  - ✓ sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  - ✓ sudo chmod +x /usr/local/bin/docker-compose
- Vérifier l'installation:
  - ✓ docker-compose --version

```
[root@localhost vagrant]# docker-compose --version
docker-compose version 1.23.2, build 1110ad01
```

Docker Compose - Exécuter des conteneurs ensemble

- Pour utiliser « Docker-compose », nous allons configurer les deux images 'SonarQube' et 'Nexus' afin de les lancer simultanément.
- Créer un dossier nommé « SonarAndNexus » et ensuite aller dans ce répertoire

```
[root@localhost vagrant]# mkdir SonarAndNexus
[root@localhost vagrant]# cd SonarAndNexus/
[root@localhost SonarAndNexus]#
```

- Créez maintenant le fichier YAML en utilisant votre éditeur de texte préféré:

```
[root@localhost SonarAndNexus]# nano docker-compose.yml
```

Docker Compose - Exécuter des conteneurs ensemble

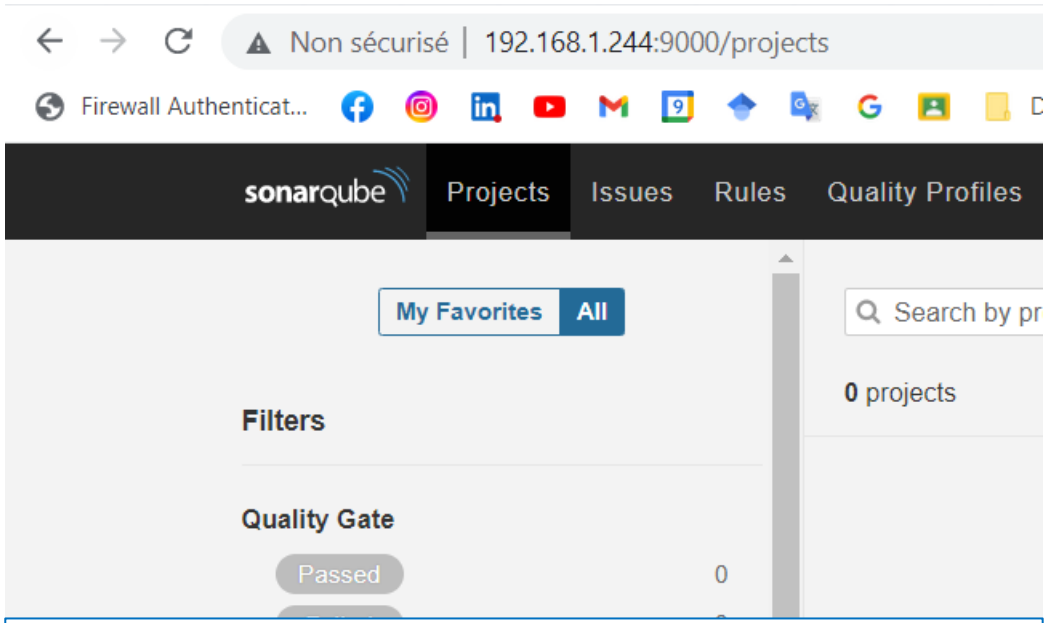
- Rédiger le fichier « docker-compose.yml »:

```
# docker-compose.yml
version: '2'
services:
 sonarqube:
  image: sonarqube:8.9.7-community
  ports:
  - "9000:9000"
  - "9092:9092"
 nexus:
  image: sonatype/nexus3
  ports:
  - "8081:8081"
```

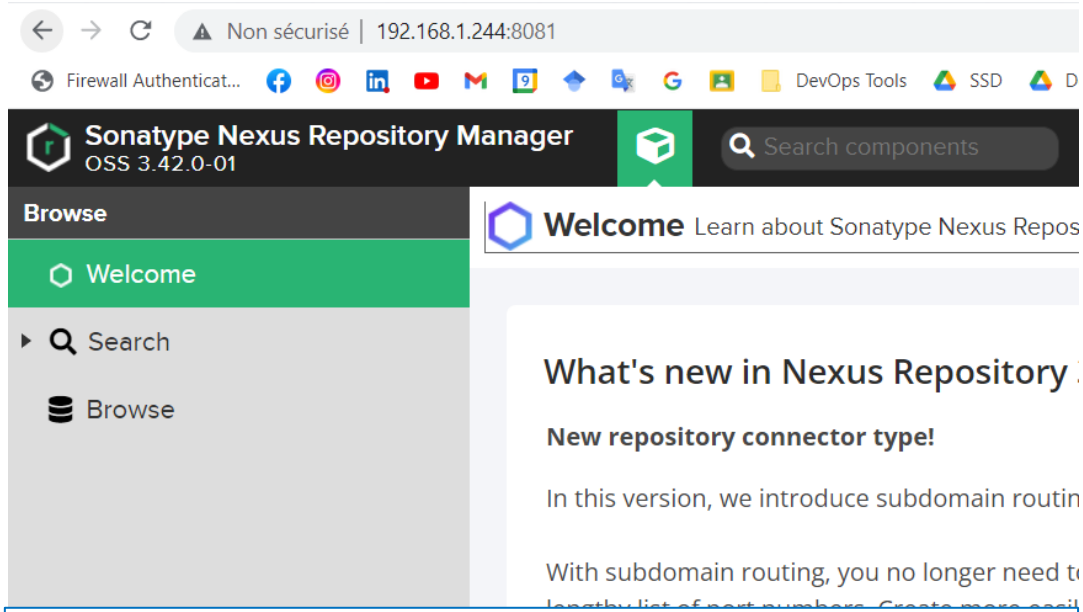
Docker Compose - Exécuter des conteneurs ensemble

- Exécuter la commande suivante pour créer le conteneur

```
[root@localhost SonarAndNexus]# docker-compose up
Creating sonarandnexus_sonarqube_1 ... done
Creating sonarandnexus_nexus_1     ... done
Attaching to sonarandnexus_nexus_1, sonarandnexus_sonarqube_1
sonarqube_1  | 2022.10.09 21:19:48 INFO  app[][o.s.a.AppFileSystem] Cleaning or creating temp directory
sonarqube_1  | 2022.10.09 21:19:48 INFO  app[][o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 1
sonarqube_1  | 2022.10.09 21:19:48 INFO  app[][o.s.a.ProcessLauncherImpl] Launch process[[key='es', ipc
```



SonarQube



Nexus

Docker Compose - Exécuter des conteneurs ensemble

- Comment vérifier les logs des conteneurs qui ont été lancés ?

docker-compose logs

```
[root@localhost SonarAndNexus]# docker-compose logs
Attaching to sonarandnexus_nexus_1, sonarandnexus_sonarqube_1
sonarqube_1  | 2022.10.09 21:19:48 INFO  app[][o.s.a.AppFileSystem] Cleaning or cre
sonarqube_1  | 2022.10.09 21:19:48 INFO  app[][o.s.a.es.EsSettings] Elasticsearch
sonarqube_1  | 2022.10.09 21:19:48 INFO  app[][o.s.a.ProcessLauncherImpl] Launch pr
elasticsearch]: /opt/sonarqube/elasticsearch/bin/elasticsearch
sonarqube_1  | 2022.10.09 21:19:48 INFO  app[][o.s.a.SchedulerImpl] Waiting for Ela
sonarqube_1  | warning: no-jdk distributions that do not bundle a JDK are deprecate
sonarqube_1  | 2022.10.09 21:19:58 INFO  es[][o.e.n.Node] version[7.16.2], pid[40],
:42:46.604893745Z], OS[Linux/3.10.0-1160.76.1.el7.x86_64/amd64], JVM[Eclipse Adopti
sonarqube_1  | 2022.10.09 21:19:58 INFO  es[][o.e.n.Node] JVM home [/opt/java/openj
nexus_1      | 2022-10-09 21:20:09,885+0000 INFO  [FelixStartLevel] *SYSTEM org.son
nexus_1      | 2022-10-09 21:20:11,535+0000 WARN  [CM Event Dispatcher (Fire Config
s not writeable: file:/opt/sonatype/nexus/etc/karaf/jmx.acl.cfg
nexus_1      | 2022-10-09 21:20:12,130+0000 WARN  [CM Event Dispatcher (Fire Config
tall - File is not writeable: file:/opt/sonatype/nexus/etc/karaf/org.apache.karaf.
nexus_1      | 2022-10-09 21:20:12,142+0000 WARN  [CM Event Dispatcher (Fire Config
```

Docker Compose - Exécuter des conteneurs ensemble

- Comment arrêter un docker compose ?

  docker-compose down

```
[root@localhost SonarAndNexus]# docker-compose down
Stopping sonarandnexus_nexus_1     ... done
Stopping sonarandnexus_sonarqube_1 ... done
Removing sonarandnexus_nexus_1     ... done
Removing sonarandnexus_sonarqube_1 ... done
Removing network sonarandnexus_default
```

Docker Compose - Exécuter des conteneurs ensemble

- Une fois que nous arrêtons l'exécution du « Docker-compose » et nous le démarrons une autre fois, nous devons **refaire** la configuration.
- En fait, la configuration est stockée dans le conteneur. Mais, si nous le supprimons, nous supprimons aussi les données de configuration.

Comment palier à ce problème ?

→ Docker Volume.

Docker Volume

- Les volumes sont le mécanisme privilégié pour la persistance des données générées et utilisées par les conteneurs Docker.
- Les volumes permettent de garder en mémoire des données de manière permanente.
- Le volume est une fonctionnalité très intéressante dans Docker. Il rend l'utilisation des conteneurs encore plus attrayante.
- Avec des volumes bien configurés, il est possible de réutiliser certaines données dans un autre conteneur, de les exporter ailleurs ou de les importer.

Docker Volume – Configuration dans Docker-Compose

```
# docker-compose.yml
version: '2'
services:
 sonarqube:
  image: sonarqube:8.9.7-community
  ports:
  - "9000:9000"
  - "9092:9092"
  volumes:
  - 'SonarQube_data:/opt/SonarQube/data'
  - 'SonarQube_extensions:/opt/SonarQube/extensions'
  - 'SonarQube_logs:/opt/SonarQube/logs'
 nexus:
  image: sonatype/nexus3
  ports:
  - "8081:8081"
  volumes:
  - 'nexus-data:/nexus-data'
volumes:
  SonarQube_data:
  SonarQube_extensions:
  SonarQube_logs:
  nexus-data:
```

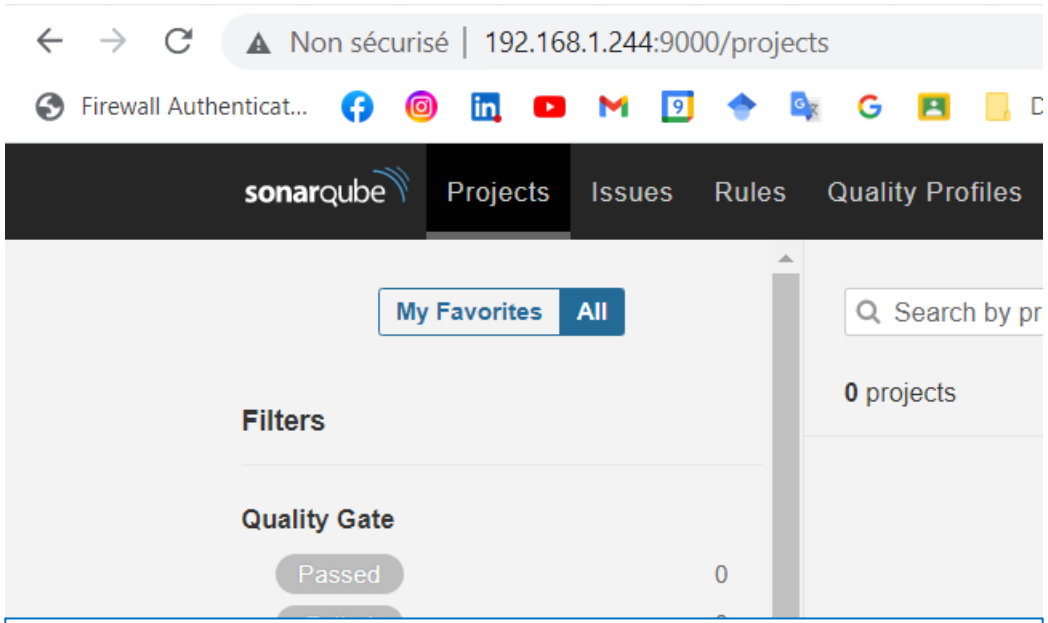
Les espaces de stockage réservés pour 'SonarQube'

L'espace de stockage réservé pour 'Nexus'

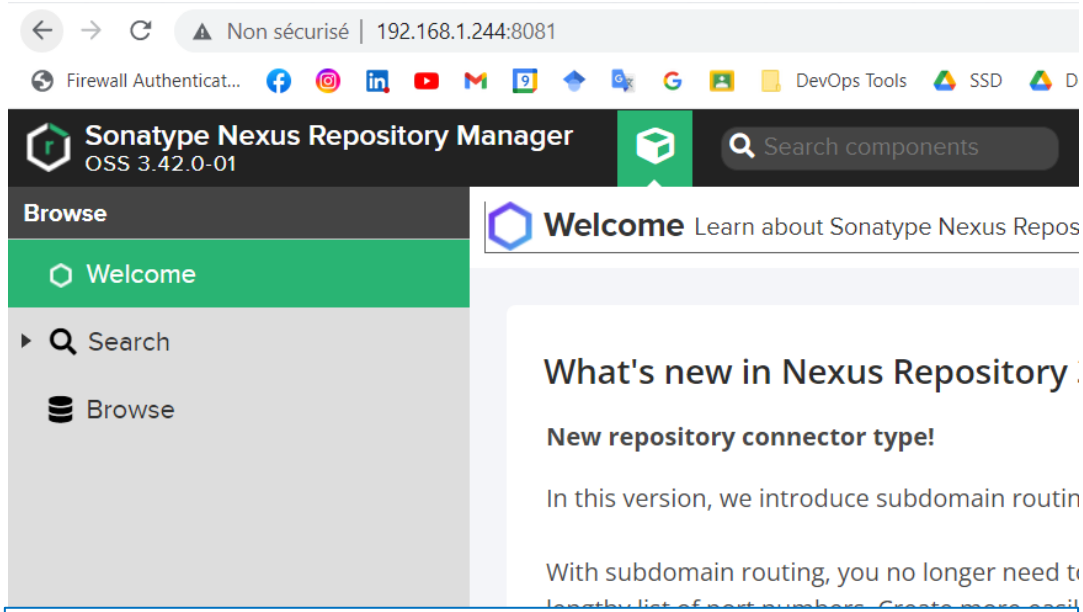
Déclaration des espaces de stockage

Docker Volume – Configuration dans Docker-Compose

```
[root@localhost SonarAndNexus]# docker-compose up
Creating volume "sonarandnexus_SonarQube_data" with default driver
Creating volume "sonarandnexus_SonarQube_extensions" with default driver
Creating volume "sonarandnexus_SonarQube_logs" with default driver
Removing sonarandnexus_sonarqube_1
sonarandnexus_nexus_1 is up-to-date
Recreating a13b6c82d625_sonarandnexus_sonarqube_1 ... done
```



SonarQube



Nexus

Docker et Jenkins

- Pour automatiser la création des images « Docker » dans « Jenkins »
  1. Installer le plugin « Docker Pipeline »:

Docker et Jenkins

1. Implémenter le fichier « DockerFile » pour créer l'image pour déployer la partie « BackEnd ».
2. Ajouter les « stages » nécessaires pour créer, construire et déposer l'image à déployer (Partie Spring) dans « DockerHub »

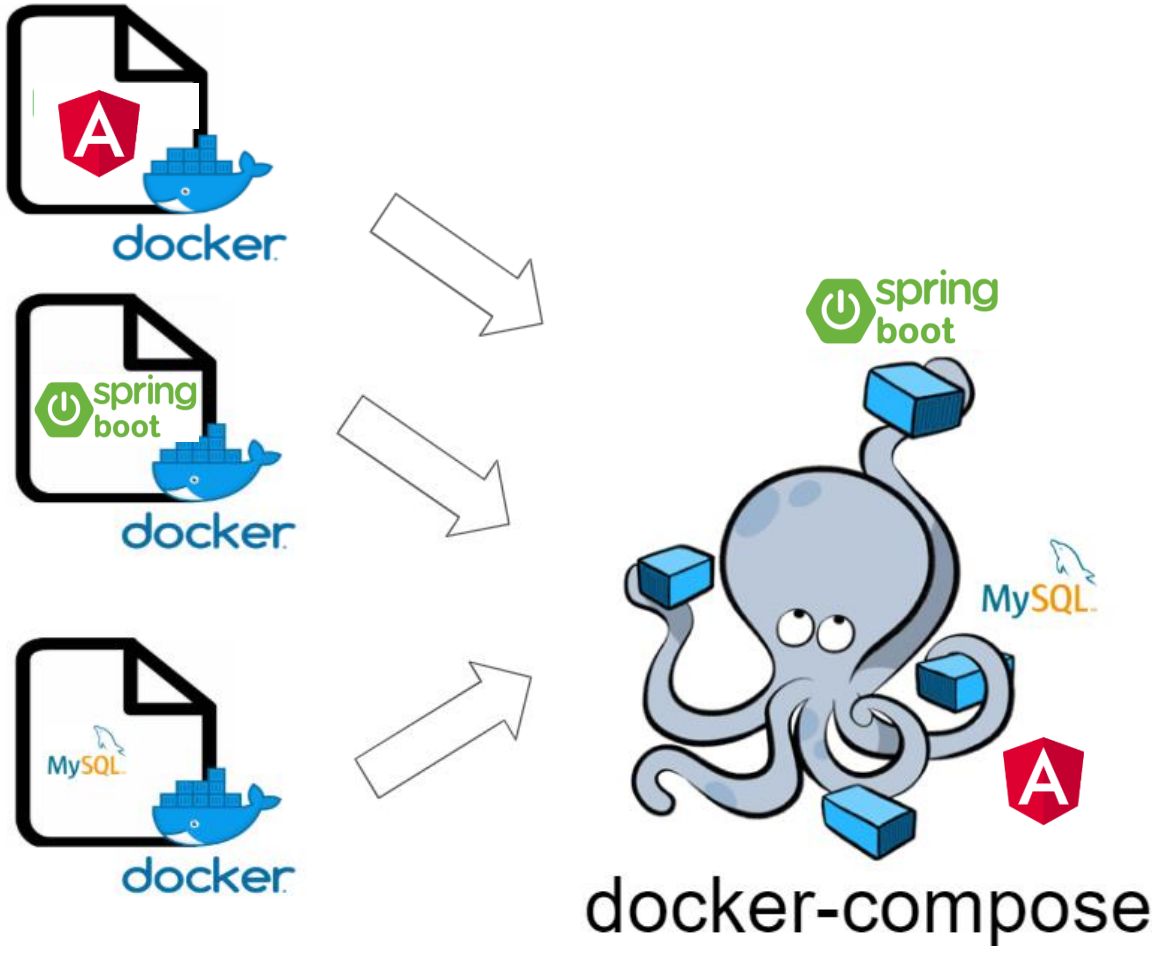
```
stage('Building image') {
  steps{
                « A Compléter …»
  }
}
```

```
stage('Deploy Image') {
  steps{
                « A Compléter …»
  }
}
```

3. Implémenter le fichier « Docker-compose » adéquat pour déployer l'application « TP Achat »

Docker et Jenkins

4. Ajouter le « stage » nécessaire pour lancer le fichier « Docker-compose » automatiquement avec l'orchestrateur Jenkins.



Docker et Jenkins

Stage View

	Declarative: Checkout SCM	Get Code from GitHub	Maven Clean	Maven build	Maven test	Sonar Test	Create package	Publish to nexus	Docker Build	Docker Push	Docker Compose
Average stage times: (Average full run time: ~20min 30s)	1s	1s	2s	5s	7s	18s	7s	6s	11s	12min 3s	7min 9s
#20 oct. 23 22:39 — 2 commits	1s	1s	2s	5s	8s	15s	5s	5s	8s	8min 5s	5min 54s aborted
#19 oct. 23 22:20 — 3 commits	1s	1s	2s	5s	6s	19s	9s	7s	15s	8min 38s	8min 24s aborted
#18 oct. 23 21:58 — No Changes	1s	1s	2s	5s	5s	18s	8s	6s	9s	19min 26s	

Docker et Jenkins

```
+ docker-compose up
 Container projet-mysqldb-1  Created
 Container projet-app-1  Creating
 Container projet-app-1  Created
Attaching to projet-app-1, projet-mysqldb-1
projet-mysqldb-1  | 2023-10-23 21:48:40+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.34-1.el8 started.
projet-mysqldb-1  | 2023-10-23 21:48:41+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
projet-mysqldb-1  | 2023-10-23 21:48:41+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.34-1.el8 started.
projet-mysqldb-1  | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
projet-mysqldb-1  | 2023-10-23T21:51:52.023899Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in
a future release. Please use SET GLOBAL host_cache_size=0 instead.
projet-mysqldb-1  | 2023-10-23T21:51:52.025227Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.34) starting as process 1
projet-mysqldb-1  | 2023-10-23T21:51:52.046870Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
projet-mysqldb-1  | 2023-10-23T21:52:03.283964Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
projet-mysqldb-1  | 2023-10-23T21:52:05.668508Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
projet-mysqldb-1  | 2023-10-23T21:52:05.668589Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections
are now supported for this channel.
projet-mysqldb-1  | 2023-10-23T21:52:05.752552Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in
the path is accessible to all OS users. Consider choosing a different directory.
projet-mysqldb-1  | 2023-10-23T21:52:05.977466Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060,
socket: /var/run/mysqld/mysqlx.sock
projet-mysqldb-1  | 2023-10-23T21:52:05.993684Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.34'  socket:
'/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
projet-app-1      |
projet-app-1      |   .   ____          _            __ _ _
projet-app-1      |  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
projet-app-1      | ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
projet-app-1      |  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
```

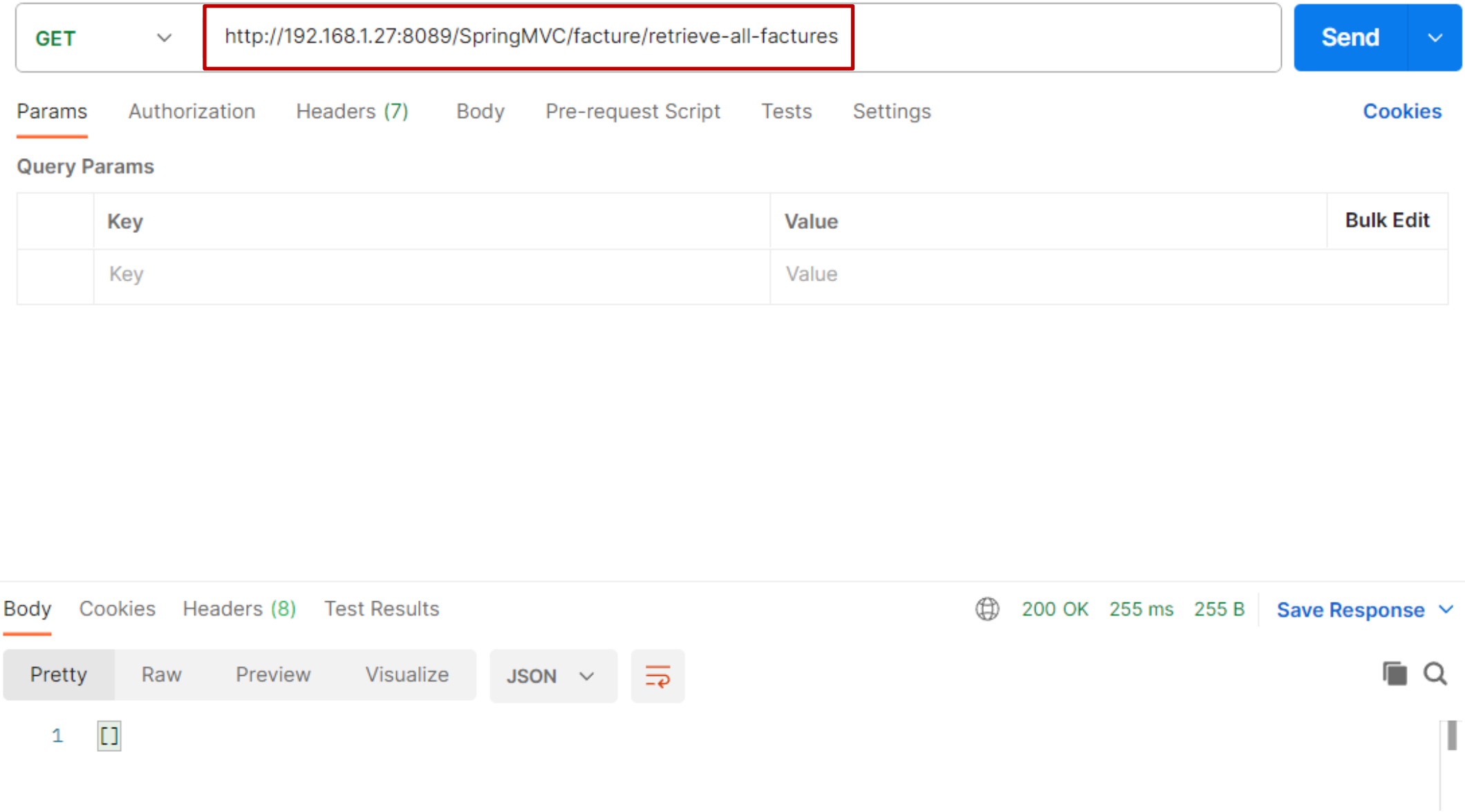
Docker et Jenkins

```
projet-app-1      |   .   ____          _            __ _ _
projet-app-1      |  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
projet-app-1      | ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
projet-app-1      |  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
projet-app-1      |   '  |____| .__|_| |_|_| |_\__, | / / / /
projet-app-1      |  =========|_|==============|___/=/_/_/_/
projet-app-1      |  :: Spring Boot ::                (v2.5.3)
projet-app-1      |
projet-app-1      | 2023-10-23 21:52:06 - INFO  - com.esprit.examen.TpAchatProjectApplication - Starting TpAchatProjectApplication v4.0.0 using Java
11.0.9 on bd5be5deb479 with PID 1 (/app/app.jar started by root in /app)
projet-app-1      | 2023-10-23 21:52:06 - INFO  - com.esprit.examen.TpAchatProjectApplication - No active profile set, falling back to default
profiles: default
projet-app-1      | 2023-10-23 21:52:08 - INFO  - o.s.data.repository.config.RepositoryConfigurationDelegate - Bootstrapping Spring Data JPA
repositories in DEFAULT mode.
projet-app-1      | 2023-10-23 21:52:08 - INFO  - o.s.data.repository.config.RepositoryConfigurationDelegate - Finished Spring Data repository
scanning in 138 ms. Found 10 JPA repository interfaces.
projet-app-1      | 2023-10-23 21:52:09 - INFO  - org.springframework.boot.web.embedded.tomcat.TomcatWebServer - Tomcat initialized with port(s):
8089 (http)
projet-app-1      | 2023-10-23 21:52:09 - INFO  - org.apache.catalina.core.StandardService - Starting service [Tomcat]
projet-app-1      | 2023-10-23 21:52:09 - INFO  - org.apache.catalina.core.StandardEngine - Starting Servlet engine: [Apache Tomcat/9.0.50]
projet-app-1      | 2023-10-23 21:52:09 - INFO  - o.a.c.core.ContainerBase.[Tomcat].[localhost].[/SpringMVC] - Initializing Spring embedded
WebApplicationContext
projet-app-1      | 2023-10-23 21:52:09 - INFO  - o.s.b.web.servlet.context.ServletWebServerApplicationContext - Root WebApplicationContext:
initialization completed in 2121 ms
projet-app-1      | 2023-10-23 21:52:09 - INFO  - org.hibernate.jpa.internal.util.LogHelper - HHH000204: Processing PersistenceUnitInfo [name:
default]
projet-app-1      | 2023-10-23 21:52:09 - INFO  - org.hibernate.Version - HHH000412: Hibernate ORM core version 5.4.32.Final
projet-app-1      | 2023-10-23 21:52:09 - INFO  - org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
```

Docker et Jenkins

```
projet-app-1      | Hibernate: alter table fournisseur_secteur_activites add constraint FKllqt521h3meitfrk3c7klahtk foreign key
(secteur_activites_id_secteur_activite) references secteur_activite (id_secteur_activite)
projet-app-1      | Hibernate: alter table fournisseur_secteur_activites add constraint FK6b9f3m4w6c3vy3xy160xd3t9l foreign key
(fournisseurs_id_fournisseur) references fournisseur (id_fournisseur)
projet-app-1      | Hibernate: alter table operateur_factures add constraint FKrgr8rgievvo95rvhdreg5lqwq foreign key (factures_id_facture) references
facture (id_facture)
projet-app-1      | Hibernate: alter table operateur_factures add constraint FKthvl793sgcnwyihwwpevgt69w foreign key (operateur_id_operateur)
references operateur (id_operateur)
projet-app-1      | Hibernate: alter table produit add constraint FKffpayryoug1422jxg4wlgr3ak foreign key (categorie_produit_id_categorie_produit)
references categorie_produit (id_categorie_produit)
projet-app-1      | Hibernate: alter table produit add constraint FKev4l89l3r0e9ogj935x8nsdfb foreign key (stock_id_stock) references stock
(id_stock)
projet-app-1      | Hibernate: alter table reglement add constraint FK5dxl3urxbrs35sol5hoxmb755 foreign key (facture_id_facture) references facture
(id_facture)
projet-app-1      | 2023-10-23 21:52:14 - INFO  - o.h.e.transaction.jta.platform.internal.JtaPlatformInitiator - HHH000490: Using JtaPlatform
implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
projet-app-1      | 2023-10-23 21:52:14 - INFO  - o.s.orm.jpa.LocalContainerEntityManagerFactoryBean - Initialized JPA EntityManagerFactory for
persistence unit 'default'
projet-app-1      | 2023-10-23 21:52:16 - WARN  - o.s.b.a.orm.jpa.JpaBaseConfiguration$JpaWebConfiguration - spring.jpa.open-in-view is enabled by
default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
projet-app-1      | 2023-10-23 21:52:16 - INFO  - org.springframework.boot.web.embedded.tomcat.TomcatWebServer - Tomcat started on port(s): 8089
(http) with context path '/SpringMVC'
projet-app-1      | 2023-10-23 21:52:17 - INFO  - com.esprit.examen.TpAchatProjectApplication - Started TpAchatProjectApplication in 10.941 seconds
(JVM running for 13.985)
projet-app-1      | 2023-10-23 21:54:11 - INFO  - o.a.c.core.ContainerBase.[Tomcat].[localhost].[/SpringMVC] - Initializing Spring DispatcherServlet
'dispatcherServlet'
projet-app-1      | 2023-10-23 21:54:11 - INFO  - org.springframework.web.servlet.DispatcherServlet - Initializing Servlet 'dispatcherServlet'
projet-app-1      | 2023-10-23 21:54:11 - INFO  - org.springframework.web.servlet.DispatcherServlet - Completed initialization in 1 ms
```

Docker et Jenkins

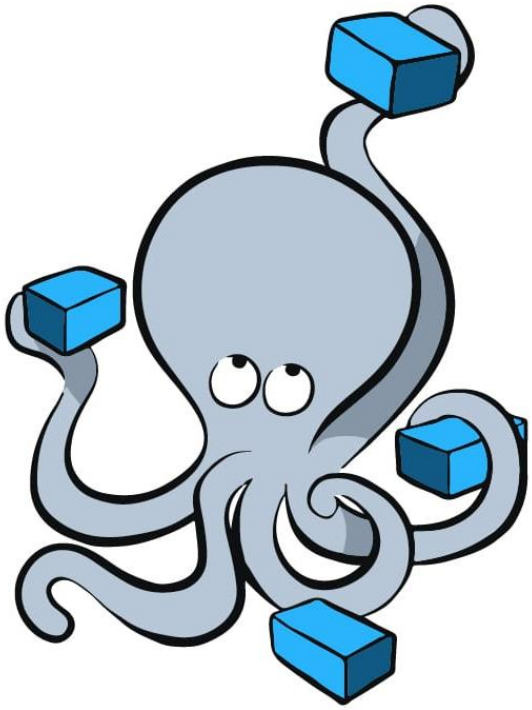


Si vous avez des questions, n'hésitez pas à nous contacter :

Département Informatique
UP ASI

Bureau E204

Docker Compose et Volume



docker

Compose et Volume

Bureau E204