DOCUMENTATION UTILISATEUR

PROJET SQUARE – INGENIEUR INFORMATIQUE II 2019-2020

INFORMATIONS

Nom du projet	**Square**
Type de document	Documentation utilisateur
Date	02/12/2019
Version	1.0
Mots-clés	Java - Quarkus - Docker - Conteneurisation - Déploiement - Automatisation – Persistance - REST – http - Postman
Auteurs	FAU Nicolas – nfau1 MARTI Emilie - emarti

TABLE DES MATIERES

1 – PRESENTATION DU PROJET

Square est une application qui permet d'automatiser le déploiement et la mise en échelle de conteneurs docker. Le programme

Il s'agit d'un programme développé en 6 semaines dans le cadre de la deuxième année d'études d'Ingénieur Informatique à l'ESIPE, Marne la Vallée.

2 – GUIDE D'INSTALLATION

Ce guide d'installation s'applique à Linux.

2.1 INSTALLATION DE L'ENVIRONNEMENT

2.1.1 SCRIPT D'INSTALLATION

Le script square_install.sh va installer les logiciels suivants :

- Maven
- Docker
- Postman

Pour exécuter le script, faire la commande suivante :

```
>> ./square_install.sh
```

Si vous n'êtes pas root, n'oubliez pas de rajouter sudo au début de la commande.

2.1.2 JAVA 11

Installation du package openjdk-11.tar.gz :

```
>> tar -zxvf openjdk-11.tar.gz
```

Variable d'environnement JAVA_HOME

```
>> export JAVA_HOME=path/to/openjdk-11
```

2.1.3 PGADMIN ET POSTGRESQL

Suivre le guide d'installation suivant : https://wiki.postgresql.org/wiki/Apt

Pour procéder, il faut générer le mot de passe de postgres.

```
>> sudo -u postgres psql postgres
>\# \password postgres
```

Il faut écrire *postgres* quand le nouveau mot de passe est demandé.

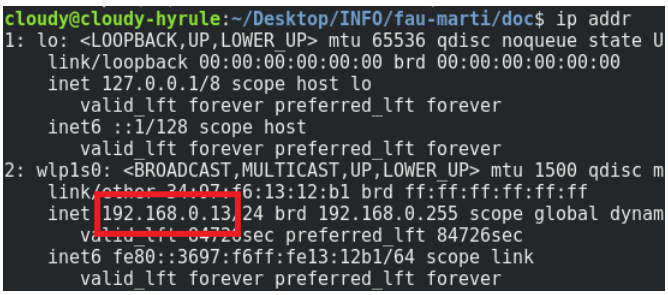
```
>\# exit
```

2.2 AIDE AU PARAMETRAGE

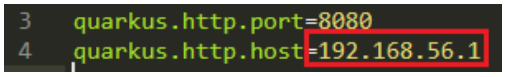
ADRESSE IP HOTE

On récupère l'adresse IP :

>> ip addr



Dans le fichier square/main/src/ressources/application.properties on va changer la ligne suivante :



Et on change l'adresse IP par la nôtre.

2.3 BASE DE DONNEES

L'application Square utilise une base de données pour la persistance des log.

2.3.1 EN LIGNE DE COMMANDE

Pour créer une base de données en ligne de commande, suivez les instructions suivantes :

>> sudo -u postgres psql postgres

># CREATE DATABASE square;

># CRATE DATABASE restcrud;

># CREATE USER nfau_marti WITH PASSWORD 'Square93';

># GRANT ALL PRIVILEGES ON DATABASE "square" to nfau_marti;

># GRANT ALL PRIVILEGES ON DATABASE "restcrud" to nfau_marti;

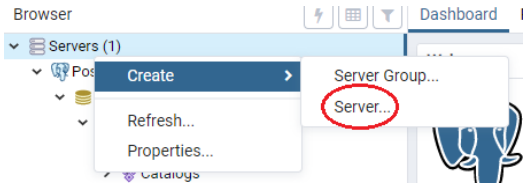
># exit

2.3.2 AVEC PGADMIN4

Pour créer une base de données à l'aide de pgAdmin 4 sur Linux, suivez les instructions suivantes :

CONNEXION A POSTGRESQL

Dans la section « Browser », faire clic droit sur Servers et choisir l'option Create > Server.



Renseignez les informations suivantes (tableau par onglet) :

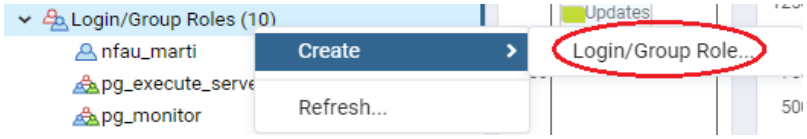
General	
Name	PostgreSQL

Connection	
Host name/address	localhost
Port	5432
Maintenance database	postgres
Username	postgres
Password	postgres

Cochez la case « Save password ? » puis cliquez sur « Save ».

CREATION DE L'UTILISATEUR FAU_MARTI

Dans la section « Browser », faire clic droit sur Servers > PostgreSQL > Login/Group Roles et choisir l'option Create > Login/Group Role.



Renseignez les informations suivantes :

General	
Name	nfau_marti

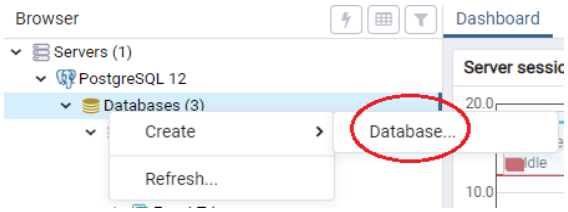
Definition	
Password	Square93

Privileges	
Can login?	Yes
Superuser?	No
Create roles?	No
Create databases?	Yes
Update catalog?	No
Inherit rights from the parent roles?	Yes
Can initiate streaming replication and backups?	No

Connection	
Host name/address	localhost
Port	5432
Maintenance database	postgres
Username	postgres
Password	postgres

CREATION DE LA BASE DE DONNEE

Dans la section « Browser », faire clic droit sur Servers > PostgreSQL > Databases et choisir l'option Create > Database.



Renseignez les informations suivantes :

General	
Database	square
Owner	nfau_marti

CREATION DE LA BASE DE DONNEES DE TEST

Répétez les pas de la section précédente, avec pour nom « restcrud ».

3 – GUIDE DE DEMARRAGE RAPIDE

3.1 MODE DEV

Pour compiler en mode développement, il faut se placer dans le dossier fau-marti/square/ et faire la commande suivante :

```
>> ./mvnw clean compile quarkus:dev
```

Le code source va être compilé et le serveur Square va démarrer. Aller à la section « pas à pas » pour découvrir comment utiliser l'application Square.

3.2 MODE TEST

Pour compiler en mode test, il faut se placer dans le dossier fau-marti/square/ et faire la commande suivante :

```
>> ./mvnw test
```

Les tests, écrits avec JUnit5, vont être automatiquement lancés.

3.3 MODE PACKAGE

Pour exécuter le jar de l'application, il faut se placer dans le dossier fau-marti/ et faire la commande suivante :

```
>> java -jar square.jar
```

Aller à la section « pas à pas » pour découvrir comment utiliser l'application Square.

4 – FICHE DES FONCTIONNALITES

4.1 DOCKER

Square permet de faire les actions suivantes :

- Déployer une instance docker ;
- Lister les instances docker actives ;
- Arrêter une instance docker.

4.3 AUTOSCALE

Square donne la possibilité d'automatiser les étapes citées ci-dessus, en proposant l'autoscale. Il permet à l'utilisateur de décider le nombre d'instances docker qui doivent être vivantes à tout moment de façon automatique et dynamique.

Square permet de faire les actions suivantes :

- Démarrer / Mettre à jour l'autoscale ;
- Lister les actions à faire par l'autoscale ;
- Arrêter le service d'autoscale.

4.4 LOGS

Square sauvegarde périodiquement des informations, sous forme de log, sur les applications qui tournent dans les conteneurs que Square gère. Ces logs peuvent être consultés à tout moment.

Square permet de faire les actions suivantes :

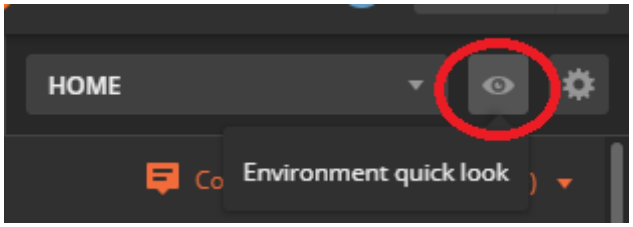
- Lister les logs depuis une date donnée.
- Lister les logs depuis une date et un filtre donné.

5 – PAS A PAS AVEC POSTMAN

Le logiciel choisi pour faire les requêtes vers notre application Square est Postman. Il s'agit d'une interface graphique qui permet de tester nos services de façon simple et intuitive.

5.1 PARAMETRAGES

Afin de simplifier nos requêtes, nous allons définir une variable d'environnement. Pour ce faire, nous cliquons sur l'œil en haut à droite :



Nous allons sur HOME > Edit et rajoutons la variable IP dont la valeur est l'adresse IP de votre machine.

Pour créer une requête avec Postman, il faut choisir le verbe (GET/POST), indiquer le nom de la requête précédé de {{IP}}:8080. Lorsqu'on veut passer un objet JSON en paramètres, on va dans l'onglet Headers et on rajoute :

Key : Content-Type | Value : application/json

Dans l'onglet Body on check la case "raw" et on écrit notre Json.

5.2 LISTE DES REQUETES

5.2.1 DOCKER

Action	Requête	Paramètre JSON
Démarrer une instance	/app/deploy	{ "app" : "name:port" }
Lister les instances	/app/list	N/A
Stopper une instance	/app/stop	{ "id" : <id> }

5.2.2 AUTOSCALE

Action	Requête	Paramètre JSON
Start/Mettre à jour l'auto-scale	/auto-scale/update	{ "name:port" : <number>, * }
Lister les actions de l'auto-scale	/auto-scale/status	N/A
Arrêter le service d'auto-scale	/auto-scale/stop	N/A

* On peut mettre autant d'instructions qu'on le souhaite, séparées par une virgule.

5.2.3 LOGS

Action	Requête	Paramètre JSON
Lister les logs depuis <time> minutes	/auto-scale/update	N/A
Lister les logs filtrés depuis <time> minutes	/auto-scale/status	N/A