Mathieu BOISNARD
Valentin FRIES
Mathieu MILANO

ESGI 2015-2016
3A AL

Raspberry

Cubic Control System

Installation et configuration
du serveur de Production RC2S

TABLE DES MATIERES

INTRODUCTION

SYSTEME D'EXPLOITATION

Distribution utilisée : Debian 8.4
IP fixe : 192.168.1.108

Utilisateur système	Mot de passe
root	root
rc2s	rc2s

LOGICIELS INSTALLES

Nom	Version	Emplacement personnalisé
Oracle Java JDK	8u92	/opt/jdk1.8.0_92
Docker	1.11.1	-
Eclipse Che	4.0.1	/opt/eclipse-che-4.0.1
Gradle	2.13	/opt/gradle-2.13
MySQL Server	?	-
Apache2	?	-
Payara	4.1.1.162	/home/rc2s/payara41
NodeJS	6.2.0	/opt/nodev6.2.0

CONFIGURATION

CONFIGURATION JDK 1.8U92

Commencez par télécharger le dernier JDK depuis le site d'Oracle :
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Dézipper l'archive dans le répertoire /opt :

```
# tar -xvf  jdk1.8.0_92.tar.gz -C /opt
```

Mettre à jour les alternatives aux commandes Java usuelles du système et des utilisateurs :

```
#  sudo update-alternatives --install /usr/bin/java java /opt/java/jdk1.8.0_92/bin/java 100
#  sudo update-alternatives --install /usr/bin/javac javac /opt/java/jdk1.8.0_92/bin/javac 100
#  sudo update-alternatives --install /usr/bin/javaws javaws /opt/java/jdk1.8.0_92/bin/javaws 100
```

Enfin, ajouter la variable d'environnement `JAVA_HOME` dans le fichier `/etc/profile`, et mettre à jour la variable d'environnement `PATH` en y ajoutant la valeur : `JAVA_HOME/bin`.

CONFIGURATION MYSQL

Installer le server MySQL :

```
# apt-get install mysql-server
```

Quand cela vous sera demandé, définissez le mot de passe de l'utilisateur `root` : `root`.

De manière à sécuriser l'installation de la base de données (empêcher les connexions root distantes, supprimer l'utilisateur anonyme…), exécutez la commande suivante et suivez les instructions :

```
# mysql_secure_installation
```

De manière à permettre les connexions distantes au serveur de base de données, éditez le fichier `/etc/mysql/my.cnf` de la manière suivante :

1. Assurez-vous que la ligne `skip-networking` soit commentée
2. Ajoutez la ligne suivante : `bind-address=192.168.1.108`

Enregistrez le fichier et redémarrez le service MySQL :

```
# service mysql restart
```

Connectez-vous à la base de données :

```
# mysql –u root -p
```

Créez un nouvel utilisateur pour les connexions distantes :

```
mysql> CREATE USER 'remote'@'localhost' IDENTIFIED BY 'remote';
mysql> FLUSH PRIVILEGES;
mysql> exit
```

Enfin, ajoutez une règle dans le pare-feu du serveur pour autoriser les connexions au port de MySQL :

Ouvrir complètement le port 3306

```
# /sbin/iptables -A INPUT -i eth0 -p tcp --destination-port 3306 -j ACCEPT
```

OU *N'ouvrir le port 3306 que sur le réseau local du serveur*

```
# /sbin/iptables -A INPUT -i eth0 -s 192.168.1.0/24 -p tcp --destination-port
3306 -j ACCEPT
```

Mémo : Utilisateurs configurés

- **Utilisateur :** root, **Mot de passe :** root, **Port :** 3306, **Accès local uniquement**
- **Utilisateur :** remote, **Mot de passe :** remote, **Port :** 3306, **Accès distant uniquement**

CONFIGURATION DOCKER

De manière à installer la dernière version de Docker, il est nécessaire d'ajouter le repository officiel dans le fichier `/etc/apt/sources.list` :

```
# echo "deb https://apt.dockerproject.org/repo debian-jessie main" >>
/etc/apt/sources.list
```

L'installation de Docker s'effectue ensuite grâce à la commande apt :

```
# apt-get update
# apt-get install docker-engine
```

Il suffit ensuite de démarrer le service Docker et de vérifier que l'installation s'est correctement effectuée :

```
# service docker start
# docker run –rm hello-world
```

CONFIGURATION GRADLE

Télécharger la dernière version de Gradle : https://gradle.org/gradle-download/

Dézipper l'archive dans le répertoire /opt :

```
# unzip gradle-2.13-bin.zip -d /opt
```

Enfin, ajouter la variable d'environnement `GRADLE_HOME` dans le fichier `/etc/profile`, et mettre à jour la variable d'environnement `PATH` en y ajoutant la valeur : `GRADLE_HOME/bin`.

CONFIGURATION ECLIPSE CHE

Télécharger la dernière version d'Eclipse Che : https://eclipse.org/che/download/

Dézipper l'archive dans le répertoire /opt :

```
# unzip eclipse-che-4.0.1.zip -d /opt
```

Eclipse CHE ne pouvant être exécuté que par un utilisateur ordinaire (non root), changez l'utilisateur propriétaire du répertoire :

```
# chown -R rc2s:rc2s /opt/eclipse-che-4.0.1/
```

Créer un script dans `/etc/init.d/` afin de lancer Eclipse Che au démarrage de la machine :

```
# /opt/eclipse-che-4.0.1/bin/che.sh run -r:192.168.1.108
```

Attention : *l'IP précisée lors de la première exécution d'Eclipse Che est mise en cache par le programme, obligeant toutes les futures exécutions à utiliser la même adresse IP. Assurez-vous de ne pas faire d'erreur lors du premier lancement de l'IDE !*

Les différents projets créés par Eclipse Che sont hébergés dans un *Workspace*, qui n'est rien autre qu'un **container Docker**. Il est donc nécessaire de créer un container **rc2s** utilisant une *stack* personnalisée :

RC2S stack

```
FROM codenvy/ubuntu_jdk8

ENV GRADLE_VERSION=2.13
ENV GRADLE_HOME=/home/rc2s/gradle-$GRADLE_VERSION
ENV PATH=$GRADLE_HOME/bin:$PATH

RUN wget -P /home/rc2s/ --quiet https://services.gradle.org/distributions/gradle-
```

```
$GRADLE_VERSION-bin.zip && \
    cd /home/rc2s/ && \
    unzip gradle-$GRADLE_VERSION-bin.zip && \
    rm gradle-$GRADLE_VERSION-bin.zip

EXPOSE 9000
WORKDIR /projects

RUN sed -i '$ d' /home/rc2s/.bashrc
RUN     echo     "export     GRADLE_HOME=$GRADLE_HOME\nexport     PATH=$PATH"     >>
/home/rc2s/.bashrc
```

Une fois le *Workspace* créé, ajoutez une commande personnalisée afin de permettre la construction des projets Gradle :

Commande : build project

```
name="$(echo ${current.project.path} | cut -d'/' -f 3 | sed 's/[^a-zA-Z0-9]//g')"
&& server="$(echo ${current.project.path}/$name-server)" && cd $server && gradle
build && cd ${current.project.path} && gradle build
```

CONFIGURATION PAYARA

1 TÉLÉCHARGEMENT DU SERVEUR

Télécharger la dernière version de Payara Server Full : http://www.payara.fish/downloads

Dézipper l'archive dans un répertoire sans restriction de droit (ici le répertoire de notre utilisateur `rc2s`) :

```
# unzip payara-4.1.1.162.zip -d /home/rc2s
```

2 INSTALLATION DU DOMAINE RC2S

Créez un nouveau domaine pour l'application RC2S :

```
# asadmin create-domain rc2s-payara
```

Utilisateur : rc2s
Mot de passé : rc2s

3 INSTALLATION DU CONNECTOR/J MYSQL

Télécharger la dernière version de jconnector de MySQL : https://dev.mysql.com/downloads/connector/j/

Dézipper l'archive et copier le fichier `mysql-connector-java-{version}-bin.jar` dans le répertoire `/home/rc2s/payara41/glassfish/domain/rc2s-payara/lib` :

```
# unzip mysql-connector-java-{version}.zip
# cp mysql-connector-java-{version}/mysql-connector-java-{version}-bin.jar
/home/rc2s/payara41/glassfish/domain/rc2s-payara/lib/
```

Pensez à redémarrer le serveur d'application – si celui-ci était actif – de manière à ce qu'il prenne connaissance du driver MySQL.

4 MISE A JOUR DES CONFIGURATIONS

Dans le package RC2S qui vous a été fourni, vous trouverez dans le répertoire `05 – Configuration Payara` tous les fichiers de configurations nécessaires au bon fonctionnement de votre serveur d'applications.

Copiez tous les fichiers de ce répertoire vers le répertoire de configuration du domaine `rc2s-payara` (écrasez les fichiers déjà présents):

```
# cp 05\ -\ Configuration\ Payara/* /home/rc2s/payara41/glassfish/domain/rc2s-payara/config/
```

Pensez à redémarrer le serveur d'application – si celui-ci était actif – de manière à ce qu'il prenne connaissance de ces nouvelles configurations.

5 MODIFICATIONS IP ET PORTS

Depuis l'interface d'administration de Payara, se rendre dans **Configurations → server-config → ORB → IIOP Listeners** et modifier l'adresse IP de `orb-listener-1` en `192.168.1.108`.

Puis se rendre dans **Configurations → server-config → Network Config → Network Listeners** et modifier le port de `http-listener-1` en `8180` afin d'éviter un conflit de port avec **Eclipse Che**, qui utilise déjà le port **8080**.

Récapitulatif des ports utilisés :

Ports utilisés par RC2S :

- **80** : HTTP Accès Apache (répertoire pour dépendance Gradle)
- **4848** : HTTPS Accès Administration Payara
- **3700** : IIOP Accès connexion Clients JavaFX
- **8080** : HTTP Accès Eclipse Che
- **3306** : TCP Accès MySQL

Ports utilisés en plus par les applications :

- **8180** : HTTP Payara HTTP Index Page
- **8181** : HTTPS Payara HTTPS Index Page
- **7676** : JMS Payara Java Messaging Service
- **3820** : IIOP_SSL Payara IIOP Secured
- **3920** : IIOP_MUTUALAUTH Payara IIOP Secured
- **8686** : JMX_ADMIN Payara JMX
- **6666** : OSGI_SHELL Payara OSGI
- **9009** : JAVA_DEBUGGER Payara

CREATION D'UN CERTIFICAT POUR SIGNER LES FICHIERS JAR :

```
# keytool -genkey -alias RC2S -keyalg RSA -keystore RC2S.jks -keysize 2048
# keytool -certreq -alias RC2S -keystore RC2S.jks -file RC2S.csr
# keytool -export -alias RC2S -file RC2S.crt -keystore RC2S.jks
// Keytool default password: changeit
# keytool -import -v -trustcacerts -alias RC2S -file RC2S.crt -keystore #
/opt/jdk1.8.0_92/jre/lib/security/cacerts
// Liste les certificats présents dans cacerts
# keytool -list -keystore /opt/jdk1.8.0_92/jre/lib/security/cacerts
// Delete l'alias RC2S
# keytool -delete -alias RC2S
```