Dossier d’installation

Ce document est destiné aux développeurs qui souhaitent mettre en place un environnement de développement.

**Thomas GIRAULT**

Dossier d’installation de l’environnement de développement v1.0

# Table des illustrations

**Aucune entrée de table d'illustration n'a été trouvée.**

Sommaire

[Table des illustrations 1](#_Toc446871432)

[Suivi des modifications 3](#_Toc446871433)

[Caracteristiques techniques 3](#_Toc446871434)

[1. Prérequis 4](#_Toc446871435)

[1.1. Caractéristiques de la machine 4](#_Toc446871436)

[1.2. Configuration réseau 4](#_Toc446871437)

[1.2.1. Configuration de l’interface réseau 4](#_Toc446871438)

[1.2.2. Configuration des paramètres réseau globaux 4](#_Toc446871439)

[1.2.3. Configuration du nom d’hôte de la machine 4](#_Toc446871440)

[1.2.4. Configuration du FQDN de la machine 5](#_Toc446871441)

[1.2.5. Configuration des serveurs DNS 5](#_Toc446871442)

[1.2.6. Redémarrage du service réseau 5](#_Toc446871443)

[1.1. Configuration des 5](#_Toc446871444)

[2. Installation de docker 5](#_Toc446871445)

[3. Jenkins 5](#_Toc446871446)

[3.1. Démarrer Jenkins 5](#_Toc446871447)

[4. Shipyard 6](#_Toc446871448)

[4.1. Démarrer Shipyard 6](#_Toc446871449)

[4.1.1. Datastore 6](#_Toc446871450)

[4.1.2. Discovery 6](#_Toc446871451)

[4.1.3. Proxy 6](#_Toc446871452)

[4.1.4. Swarm Manager 7](#_Toc446871453)

[4.1.5. Swarm Agent 7](#_Toc446871454)

[4.1.6. Controller 7](#_Toc446871455)

[4.2. Arrêter Shipyard 8](#_Toc446871456)

[4.2.1. Arrêter les conteneurs Shipyard 8](#_Toc446871457)

[4.2.2. Supprimer les conteneurs Shipyard 8](#_Toc446871458)

[5. Installer un serveur Git 8](#_Toc446871459)

[6. Annexes 9](#_Toc446871460)

[6.1. Variables d’environnement 9](#_Toc446871461)

# Suivi des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Modifié par | Version |
| 24/01/2016 | Initialisation du document | TGI | v1.0 |
|  |  |  |  |
|  |  |  |  |

# Caracteristiques techniques

Le poste de développement doit répondre au minimum aux caractéristiques suivantes :

|  |  |
| --- | --- |
| Caractéristiques | |
| Système d’exploitation | Linux Centos 7 |
| Type du système | Système d’exploitation 64 bits |
| Processeur | Intel® Core 2 Duo CPU P8700 @ 2.53 GHz 2.53 GHz |
| Mémoire installée (RAM) | 4,00 Go |

# Prérequis

## Caractéristiques de la machine

Le serveur linux est installé avec le système d’exploitation CentOS 7.

|  |  |
| --- | --- |
| Caractéristiques | |
| Système d’exploitation | Linux Centos 7 |
| Type du système | Système d’exploitation 64 bits |
| Processeur | Intel® Core 2 Duo CPU P8700 @ 2.53 GHz 2.53 GHz |
| Mémoire installée (RAM) | 4,00 Go |

## Configuration réseau

Se connecter sur le serveur avec le compte administrateur root.

### Configuration de l’interface réseau

1. Éditer le script de l’interface réseau /etc/sysconfig/network-scripts/ifcfg-enp0s3
2. Renseigner les lignes suivantes :

TYPE=Ethernet

BOOTPROTO=static

NAME=eth0

DEVICE=enp0s3

ONBOOT=yes

HWADDR=AA:BB:CC:DD:EE:FF

IPADDR=

NETWORK=

NETMASK=

BROADCAST=

GATEWAY=

### Configuration des paramètres réseau globaux

1. Éditer le fichier /etc/sysconfig/network
2. Renseigner les lignes suivantes :

NETWORKING=yes

NETWORKING\_IPV6=no

### Configuration du nom d’hôte de la machine

1. Éditer le fichier /etc/hostname
2. Renseigner les lignes suivantes :

HOSTNAME=

Le nom d’hôte suit le format slnx<projet><env><numero>.

### Configuration du FQDN de la machine

1. Éditer le fichier /etc/hosts
2. Renseigner les lignes suivantes :

FQDN : Fully Qualified Domain Name

### Configuration des serveurs DNS

1. Éditer le fichier /etc/resolv.conf
2. Renseigner les lignes suivantes :

nameserver

nameserver

### Redémarrage du service réseau

1. Redémarrer le service réseau

service network restart

# Installation de docker

TODO

# Shipyard

Shipyard est un outil qui permet d’administrer les conteneurs Docker.

## Démarrer Shipyard

### Démarrage automatique

curl -sSL https://shipyard-project.com/deploy | bash -s

### Démarrage manuel

### Datastore

Démarre le conteneur RethinkDB, la base de données utilisée par Shipyard.

docker run \

-ti \

-d \

--restart=always \

--name shipyard-rethinkdb \

rethinkdb

### Discovery

Pour activer l’élection d’un maître Swarm, nous devons utiliser une paire clé/valeur externe stockée dans le conteneur Swarm. Dans l’exemple qui suit, nous utilisons etcd. Cependant vous pouvez utiliser n’importe quelle clé/valeur supportée par Swarm.

docker run \

-ti \

-d \

-p 4001:4001 \

-p 7001:7001 \

--restart=always \

--name shipyard-discovery \

microbox/etcd -name discovery

### Proxy

Démarre le conteneur proxy qui redirige simplement les requêtes TCP vers le socket Unix sur lequel Docker écoute.

docker run \

-ti \

-d \

-p 2375:2375 \

--hostname=$HOSTNAME \

--restart=always \

--name shipyard-proxy \

-v /var/run/docker.sock:/var/run/docker.sock \

-e PORT=2375 \

shipyard/docker-proxy:latest

### Swarm Manager

Démarre un conteneur Swarm configuré pour gérer les contenurs.

docker run \

-ti \

-d \

--restart=always \

--name shipyard-swarm-manager \

swarm:latest \

manage --host tcp://0.0.0.0:3375 etcd://192.168.0.100:4001

### Swarm Agent

Démarre l’agent Swarm qui autorise les nœuds à ordonnancer les conteneurs.

docker run \

-ti \

-d \

--restart=always \

--name shipyard-swarm-agent \

swarm:latest \

join --addr 192.168.0.100:2375 etcd://192.168.0.100:4001

### Controller

Démarre le contrôleur Shipyard.

docker run \

-ti \

-d \

--restart=always \

--name shipyard-controller \

--link shipyard-rethinkdb:rethinkdb \

--link shipyard-swarm-manager:swarm \

-p 8080:8080 \

shipyard/shipyard:latest \

server \

-d tcp://swarm:3375

Dès lors que le contrôleur est démarré, il est possible de se connecter à l’adresse suivante : http://192.168.0.100:8080

* Login : admin
* Mot de passe : shipyard

Lien: http://shipyard-project.com/docs/deploy/manual/

## Arrêter Shipyard

Arrêter les conteneurs Shipyard.

docker stop shipyard-controller

docker stop shipyard-swarm-agent

docker stop shipyard-swarm-manager

docker stop shipyard-proxy

docker stop shipyard-certs

docker stop shipyard-discovery

docker stop shipyard-rethinkdb

Supprimer les processus Shipyard.

docker ps -a | grep 'shipyard' | awk '{print $1}' | xargs --no-run-if-empty docker rm

# Jenkins

## Démarrer Jenkins

Démarrer le conteneur Jenkins.

docker run -d -p 8081:8080 -p 50000:50000 -v /var/jenkins\_home jenkins

**Paramètres:**

-d : permet d’exécuter le conteneur en tâche de fond

-p : permet d’associer les ports réseau du conteneur avec les ports de la machine hôte.

-v : permet d’associer un volume

Dès lors que le contrôleur est démarré, il est possible de se connecter à l’adresse suivante : http://192.168.0.100:8081

Lien: https://hub.docker.com/\_/jenkins/

## Arrêter Jenkins

Arrêter le conteneur Jenkins.

docker stop jenkins

Supprimer le processus jenkins.

docker ps -a | grep 'jenkins' | awk '{print $1}' | xargs --no-run-if-empty docker rm

# Annexes

## Variables d’environnement

|  |  |  |
| --- | --- | --- |
| Variable | Valeur | Description |
| $\_ IPADRESS |  | Adresse IP de la machine |
| $\_HOSTNAME |  | Nom d’hôte de la machine |
| $\_FQDN |  | Fully Qualified Domain Name de la machine |
| $\_DOMAIN | .fr.local | Domaine de la machine |
| $\_NETWORK |  | Adresse du réseau |
| $\_GATEWAY |  | Passerelle vers le réseau |
| $\_NETMASK |  | Masque du réseau |
| $\_BROADCAST |  | Adresse de broadcast |
| $\_DNS1 |  | Adresse DNS 1 |
| $\_DNS2 |  | Adresse DNS 2 |