

canarie



Guide technique

Déploiement des conteneurs Jenkins dans le nuage de l'ATIR

Auteur : Don McCullough
Architecte en infonuagique
dair.admin@canarie.ca

canarie.ca | [@canarie_inc](https://twitter.com/canarie_inc)

Table des matières

Aperçu	3
Exigence	Error! Bookmark not defined.
Configuration	3
Configuration de Jenkins.....	7
Dépendances.....	8
Problèmes	8
Remplacer les certificat auto-signé.....	8
Mise à niveau de Jenkins	9

Aperçu

Jenkins est un important serveur d'automatisation de source ouverte. On y retrouve des centaines de modules d'extension facilitant l'élaboration, le déploiement et l'automatisation de projets en tout genre. Ce guide explique comment lancer et configurer un environnement Jenkins de base avec Docker dans le nuage de l'ATIR.

Prérequis

Vous devriez d'abord lire le [Guide technique de l'ATIR – Utiliser Docker avec l'ATIR](#) pour savoir comment lancer des hôtes Docker dans l'ATIR avec Docker Machine. Le document [Docker Commands for Daily Life](#) est également un aide-mémoire utile sur les commandes les plus courantes de Docker.

Par ailleurs, vous aurez besoin d'une application cliente Git pour cloner le dépôt renfermant Jenkins et Nginx (serveur HTTP et serveur inverse par procuration). Si vous connaissez mal Git, apprenez-en davantage en lisant le [guide d'installation officiel](#). Une autre possibilité consiste à [télécharger ce dépôt](#) sous forme d'archive zip.

Installation

Exécutez les commandes que voici avec la machine sur laquelle Docker Machine et Docker Compose sont installés.

1^{re} étape : Clonez ce dépôt.

```
git clone https://solutions.cloud.canarie.ca:3000/DAIR/dair-jenkins-guide
cd dair-jenkins-guide
```

2^e étape : Assurez-vous que votre configuration autorise un déploiement sur l'instance de l'ATIR que vous avez créée.

```
docker-machine active
```

Remarque : la commande « `docker-machine ls` » pourrait vous aider à établir quelles machines vous avez déjà créées.

Si la commande restitue « No active found » (aucune machine active trouvée), exécutez les deux commandes que voici.

```
$ docker-machine env Hostname
cp://208.75.xx.yy:2376"
$ eval $(docker-machine env Hostname)
```

Où **Hostname** sera remplacé par le nom d'hôte Docker que vous avez choisi à la création de votre machine Docker (veuillez lire la section « Exigences » du guide « Utiliser Docker avec l'ATIR » mentionné précédemment).

3^e étape : Construisez les images Docker. Il y en a deux. La première est l'image Jenkins proprement dite, la seconde, une image Nginx. Cette dernière vous permettra d'accéder à Jenkins par procuration et de clore le protocole SSL.

```
docker-compose build
```

En exécutant la commande « `docker images` », vous devriez voir quelque chose de semblable apparaître à l'écran :

```
$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED
SIZE
dair-jenkins-guide_nginx  latest      b6554930ab8e     3 hours ago
111 MB
dair-jenkins-guide_jenkins  latest      d1dd6dd31ba6     4 hours ago
700 MB
docker.io/nginx          1.15.2      c82521676580     8 days ago
109 MB
docker.io/jenkins/jenkins  lts        80fc01fa46f9     2 weeks ago
700 MB
```

Les deux premières images sont les dernières à avoir été créées. Les deux autres sont les images de départ que vous avez téléchargées dans le cadre de la présente démarche.

4^e étape : Le moment est venu de lancer les conteneurs.

```
docker-compose up -d
```

Pour vous assurer que les conteneurs ont été initialisés et fonctionnent, exécutez la commande « `docker ps` »; voici ce que vous devriez voir apparaître à l'écran.

```

$ docker ps
CONTAINER ID          IMAGE                    COMMAND                  CREATED
STATUS              PORTS                   NAMES
811d7a26cffe        dair-jenkins-guide_nginx  "nginx -g 'daemon ..." 4 hours ago
Up 4 hours          0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp  nginx
4e88fe01ac10        dair-jenkins-guide_jenkins  "/sbin/tini -- /us..." 4 hours ago
Up 4 hours          8080/tcp, 0.0.0.0:5000->5000/tcp, 50000/tcp  jenkins

```

Maintenant que les services conteneurisés fonctionnent, accédez à l'interface Web de Jenkins avec le navigateur à l'adresse https://<floating_ip>. Un avertissement indiquant que le certificat n'est pas valable s'affichera à l'écran. Ignorez-le. Puisque Nginx utilise un certificat différent, auto-signé, ajoutez une exception à Jenkins.

La page ressemblera à ceci :

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

[Continue](#)

Lancez la commande ci-dessous pour obtenir le mot de passe de l'administrateur Jenkins :

```

docker-compose exec jenkins cat
/var/jenkins_home/secrets/initialAdminPassword

```

À présent, personnalisez Jenkins en installant quelques modules d'extension.

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.121.2

Si vous hésitez sur l'option à choisir, installez les modules suggérés. Vous pourrez toujours les retrancher plus tard, une fois la configuration terminée.

Après avoir installé les modules, complétez le formulaire concernant le premier administrateur :

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.121.2 [Continue as admin](#) [Save and Continue](#)

Pour terminer, saisissez l'URL de Jenkins :

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.121.2 Not now

La valeur par défaut est la bonne, car vous avez déjà accédé à Jenkins grâce à l'adresse IP flottante de l'hôte Docker Machine.

Désormais, vous pouvez vous connecter avec le compte d'utilisateur que vous venez de créer et commencer à vous servir de Jenkins!

Configuration de Jenkins

Jenkins peut être configuré d'une multitude de manières et même plus, si on prend en compte les innombrables modules d'extension offerts. Si vous n'êtes pas en terrain familier, nous vous suggérons vivement de lire la [documentation](#).

Jenkins peut être presque entièrement configuré grâce à l'interface Web. Cependant, si vous devez modifier quoi que ce soit manuellement ou voulez simplement faire quelques essais, exécuter la commande qui suit créera une coquille interactive qui vous permettra de le faire dans le conteneur.

```
docker-compose exec jenkins bash
```

Les modifications apportées au répertoire `/var/jenkins_home` demeureront, même après la destruction du conteneur et la machine les reproduira, parce que le volume est géré par Docker, à l'intérieur du conteneur.

Dépendances

Si elles ne figurent pas déjà dans l'image Jenkins, ajoutez des dépendances. Veuillez consulter [jenkins/Dockerfile](#) pour savoir comment procéder, puis rebâissez l'image et recréez le conteneur.

```
docker-compose build
docker-compose up -d --no-deps jenkins
```

Problèmes

Si vous éprouvez des difficultés, il est possible de consulter le journal de chaque conteneur en lançant les commandes que voici :

```
docker-compose logs jenkins
docker-compose logs nginx
```

Remplacer le certificat auto-signé

Lorsque vous avez bâti l'image Nginx, vous avez créé un certificat auto-signé qui a activé le protocole HTTPS. C'est pourquoi vous avez dû ajouter une exception la première fois que vous avez accédé à Jenkins. Si vous avez votre propre domaine et un certificat bien à vous, il est possible de les substituer au certificat auto-signé une fois que Nginx fonctionne, en utilisant les commandes suivantes :

```
docker cp /path/to/cert nginx:/etc/nginx/ssl/server.crt
docker cp /path/to/key nginx:/etc/nginx/ssl/server.key
docker-compose restart nginx
```

Le répertoire `/etc/nginx/ssl` est monté comme un volume de données dans Docker. Le certificat et sa clé demeureront en place, même le conteneur est retranché, à la relance du système. Pour supprimer ce volume, exécutez la commande que voici :


```
docker volume rm dair-jenkins-guide_nginx-ssl
```

Mise à niveau de Jenkins

Le fichier Docker employé pour bâtir l'image de Jenkins utilise la version LTS de Jenkins, dont les fonctionnalités pourraient devenir désuètes. Pour utiliser une version plus récente, modifiez la première ligne de `jenkins/Dockerfile` et remplaçant `lts` par `latest` (pour la version la plus récente) ou par le numéro d'une version précise, 2.135, par exemple. Veuillez noter que cette modification s'effectue à votre propre péril, car la documentation pourrait alors ne pas s'appliquer aux versions subséquentes. Après avoir modifié `jenkins/Dockerfile`, vous devrez rebâtir l'image et le conteneur Jenkins avec les commandes suivantes :

```
docker-compose build
docker-compose up -d --no-deps jenkins
```