

BUILDAH: Lab 1 construction d'un simple serveur http

Objet	Création d'une conteneur http
Niveau requis	débutant, avisé
Débutant, à savoir	
Suivi	:DRAFT:

Cet exemple ajoute un service Web (httpd) à un conteneur et le configure pour son exécution. Dans l'exemple, au lieu de valider l'image dans Buildah (conteneurs-storage stocké localement dans /var/lib/containers), nous expliquons comment valider l'image afin qu'elle puisse être gérée par le service Docker local (docker-daemon qui stocke localement dans /var/lib/docker). Vous aurier simplement pu l'envoyer facilement à Buildah, ce qui vous permettrait de le transmettre ensuite à un service Docker (docker), à un référentiel OSTree local (ostree) ou à un autre stockage (oci) compatible OCI. (Taper man buildah pour plus de détails.)

Créer un conteneur de travail:

```
# buildah from scratch  
working-container
```

Cela crée juste un conteneur vide (aucune image) que l'on peut monter comme suit:

```
# scratchmnt=$(buildah mount working-container)  
# echo $scratchmnt  
/var/lib/containers/storage/devicemapper/mnt/cc92011e9a2b077d03a97c0809f1f3e  
7fef0f29bdc6ab5e86b85430ec77b2bf6/rootfss
```

Installer le serveur

Initialiser une base de données RPM dans l'image de travail et ajouter le paquetage redhat-release (qui inclut d'autres fichiers nécessaires au bon fonctionnement des RPM):

```
# rpm --root $scratchmnt --initdb  
# yum install yum-utils          (if not already installed)  
# yumdownloader --destdir=/tmp redhat-release-server  
# rpm --root $scratchmnt -ihv /tmp/redhat-release-server*.rpm
```

Installer le service httpd dans le répertoire scratch:

```
# yum install -y --installroot=$scratchmnt httpd
```

Ajouter du texte à un fichier index.html dans le conteneur pour pouvoir le tester ultérieurement:

```
# echo "Your httpd container from scratch worked." >
$scratchmnt/var/www/html/index.html
```

Au lieu d'exécuter httpd en tant que service init, définir quelques options de configuration de buildah pour exécuter le démon httpd directement à partir du conteneur:

```
# buildah config --cmd "/usr/sbin/httpd -DFOREGROUND" working-container
# buildah config --port 80/tcp working-container
```

Enregistrer la configuration

```
# buildah commit working-container docker-daemon:myhttpd:latest
```



Par défaut, la commande `buildah commit` ajoute le nom du référentiel docker.io au nom de l'image et la copie dans la zone de stockage du service Docker local (/var/lib/docker). Pour ajouter le référentiel à un registre local tout en désactivant la vérification tls utiliser:

```
buildah commit --tls-verify=false containerID
docker://localhost:5000/imageId
```

On peut utiliser l'ID d'image pour exécuter la nouvelle image en tant que conteneur à l'aide de la commande docker:

```
# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
docker.io/myhttpd   latest             47c0795d7b0e       9 minutes ago
665.6 MB
# docker run -p 8080:80 -d --name httpd-server 47c0795d7b0e
# curl localhost:8080
Your httpd container from scratch worked.
```

Exécuter un conteneur avec Buildah

Pour vérifier que l'image que vous avez construite précédemment fonctionne, vous devez créer un conteneur de travail à partir de l'image, puis utiliser `buildah run` pour exécuter le conteneur de travail.

```
# buildah from myecho
```

```
myecho-working-container
# buildah containers
CONTAINER ID   BUILDER  IMAGE ID      IMAGE NAME      CONTAINER NAME
dc8f21af4a47   *        1456eedf8101  registry.access.redhat.com/rhel7/rhel-
minimal:latest
    rhel-minimal-working-container
6d1ffccb557d   *        ab230ac5aba3  docker.io/library/myecho:latest
    myecho-working-container
# buildah run myecho-working-container
This container works!
```

Les étapes de la construction ont utilisé l'image (myecho) pour créer un conteneur (myecho-working-container). Après cela, les conteneurs Buildah ont montré que le conteneur existe et Buildah a exécuté le conteneur, produisant la sortie: Ce conteneur fonctionne!

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=labs:buildah-lab-1-http-server>

Last update: **2025/02/19 10:59**

