

Docker: Lab DC 1 créer des conteneurs avec les navigateurs Windows

Objet	Création d'un conteneur IE ou Edge
Niveau requis	débutant, avisé
Débutant, à savoir	
Suivi	:DRAFT:

Ce lab contient des instructions de construction et Dockerfile pour créer des images Docker avec des navigateurs Windows uniquement: Internet Explorer et Microsoft Edge

Configuration requise

- Machine Bare Metal ou VM avec virtualisation activée et Linux installé. Cet exemple a été testé sur Ubuntu 18.04.

```
$ uname -a
Bureau Linux 4.15.0-46-generic # 49-Ubuntu SMP Mer 6 Fév 09:33:07 UTC 2019
x86_64 x86_64 x86_64 GNU / Linux
```

Pour vérifier que la virtualisation est prise en charge, vérifier que le fichier /dev/kvm est présent:

```
$ ls -l /dev/kvm
crw-rw---- 1 racine kvm 10, 232 mars 8 19:38 /dev/kvm
```

- Emulateur de machine Qemu installé. Il est important d'utiliser la même version de qemu sur la machine hôte où les images sont construites et dans l'image Docker. Pour vérifier le type de version de qemu:

```
$ qemu-system-x86_64 -version
Emulateur QEMU version 2.11.1 (Debian 1: 2.11 + dfsg-1ubuntu7.10)
Copyright (c) 2003-2017 Fabrice Bellard et les développeurs du projet QEMU
```

* Clé de licence Windows

Procédure de construction

Étapes préparatoires

Créer l'environnement de configuration docker

- créer les dossiers windows-image/image

```
$ mkdir -c windows-image/image
```

- charger le fichier windows-images/image/Dockerfile avec le contenu suivant

```
$ cat windows-images/image/Dockerfile

FROM ubuntu:18.04
RUN apt update && \
    apt -y upgrade && \
    apt -y install xvfb x11-utils x11vnc qemu
COPY hdd.img /
COPY snapshot.img /
COPY entrypoint.sh /
ENTRYPOINT ["/entrypoint.sh"]
```

- charger le fichier windows-images/image/entrypoint.sh avec le contenu suivant

```
$ cat windows-images/image/entrypoint.sh

#!/bin/sh

#!/bin/bash
SCREEN_RESOLUTION=${SCREEN_RESOLUTION:-"1024x768x24"}
DISPLAY_NUM=99
export DISPLAY=":$DISPLAY_NUM"

clean() {
    if [ -n "$XVFB_PID" ]; then
        kill -TERM "$XVFB_PID"
    fi
    if [ -n "$X11VNC_PID" ]; then
        kill -TERM "$X11VNC_PID"
    fi
}

trap clean INT TERM

# Démarrage de l'image docker sans serveur X avec le wrapper Xvfb.
# - la première ligne démarre le wrapper graphique
# - les lignes suivantes démarrent la VM

xvfb-run -l -n $DISPLAY_NUM -s "-ac -screen 0 $SCREEN_RESOLUTION -noreset -listen tcp" \
    qemu-system-x86_64 -enable-kvm \
    -machine q35 -smp sockets=1,cores=1,threads=2 -m 2048 \
```

```
-usb -device usb-kbd -device usb-tablet -rtc base=localtime \  
-net nic,model=virtio -net user,hostfwd=tcp::4444-:4444 \  
-drive file=snapshot.img,media=disk,if=virtio \  
-loadvm windows &
```

```
XVFB_PID=$!
```

```
retcode=1  
until [ $retcode -eq 0 ]; do  
  xdpinfo -display $DISPLAY >/dev/null 2>&1  
  retcode=$?  
  if [ $retcode -ne 0 ]; then  
    echo Waiting xvfb...  
    sleep 1  
  fi  
done
```

```
# x11vnc pour prendre le contrôle du bureau à distance
```

```
x11vnc -display $DISPLAY -passwd selenoid -shared -forever -loop500 -rfbport  
5900 -rfbportv6 5900 -logfile /dev/null &  
X11VNC_PID=$!
```

```
wait
```

- Se déplacer dans le dossier windows-images/:

```
$ cd windows-images
```

Télécharger l'image d'installation de Windows 10

A partir du site Web de téléchargement de logiciels Microsoft.

Télécharger les pilotes virtio

virtio-win-0.1.141.iso. Dans les étapes suivantes, on suppose q'on a maintenant deux fichiers dans le répertoire actuel:

```
$ ls  
virtio-win-0.1.141.iso Win10_18090ct_Français_x32.iso
```

Installation sous Windows

Démarrer la VM

- Créer une image de disque dur sur lequel Windows sera installé:

```
$ qemu-img créer -f qcow2 hdd.img 40G
```

- Lancer la machine virtuelle et commencer l'installation:

```
$ sudo qemu-system-x86_64 -enable-kvm \  
    -machine q35 -smp sockets = 1, cœurs = 1, threads = 2 -m 2048 \  
    -usb -device usb-kbd -device usb-tablet -rtc base = heure locale \  
    -net nic, model = utilisateur virtio -net, hostfwd = tcp :: 4444-:4444 \  
    -drive file = hdd.img, media = disque, si = virtio \  
    -drive file = Win10_18090ct_English_x32.iso, media = cdrom \  
    -drive file = virtio-win-0.1.141.iso, media = cdrom
```

Installer Windows

Windows démarrera à partir de l'image d'installation

- Installer le pilote de stockage virtio
 - Cliquer sur Charger le pilote
 - Pointer sur le répertoire E:\viostor\w10\x86
 - Cliquer sur Suivant pour installer le pilote
 - Choisir la partition d'installation et cliquer sur suivant
 - Attendre la fin de l'installation
- Configurer l'utilisateur et le mot de passe
- Installer le pilote Ethernet virtio
 - Ouvrir le Gestionnaire de périphériques et cliquer sur Mettre à jour le pilote
 - Choisir virtio cdrom et cliquer sur OK
 - Installer le pilote
 - Se connecter au réseau
- Désactiver le pare-feu Windows, ou ajouter une règle de pare-feu pour autoriser l'accès au port 4444. Cela est nécessaire pour accéder au port binaire de WebDriver avec le test Selenium.
- Terminer l'installation de Windows
 - installer les mises à jour,
 - modifier la résolution de l'écran,
 - appliquer les modifications du registre,
 - etc...

Ajouter de fichiers binaires WebDriver

Ces fichiers binaires gèrent les demandes de test Selenium et lancent le navigateur correspondant.

- **Pour Internet Explorer** - télécharger une archive avec le binaire du pilote sur le site officiel de Selenium, décompresser et mettre le binaire dans le répertoire C:\Windows\System32.

- **Pour Microsoft Edge**, le binaire peut être installé avec la commande suivante: `DISM.exe /Online /Add-Capability /CapabilityName:Microsoft.WebDriver~~~00.1.1.0`

Configuration de la VM

Création d'un instantané de la VM

Cet instantané contient l'état de la machine et est nécessaire pour restaurer rapidement la machine virtuelle au lieu d'effectuer un démarrage complet, ce qui est lent. Pour le créer:

- Arrêter la machine virtuelle
- Créer une image contenant l'état de la machine virtuelle: `$ qemu-img créer -b hdd.img -f qcow2 snapshot.img`

Modifier l'instantané VM

- Exécuter la VM en utilisant `snapshot.img` en tant que système de fichiers:

```
$ sudo qemu-system-x86_64 -enable-kvm \  
    -machine q35 -smp sockets = 1, cœurs = 1, threads = 2 -m 2048 \  
    -usb -device usb-kbd -device usb-tablet -rtc base = heure locale \  
    -net nic, model = utilisateur virtio -net, hostfwd = tcp :: 4444-:4444 \  
    -drive file = snapshot.img, media = disque, si = virtio \  
    -moniteur stdio
```



qemu fonctionne avec un moniteur connecté à `stdio`.

- Configurer le navigateur (requis uniquement pour Internet Explorer).
 - Ouvrir Internet Explorer. La première fois que ce navigateur est lancé, il demande la configuration de la sécurité. L'option "Ne pas utiliser les paramètres recommandés" doit être sélectionnée.
 - Modifier les options Internet. Ces options peuvent être ouvertes à l'aide du bouton de configuration situé en haut d'Internet Explorer. Dans l'onglet "Sécurité", le mode de protection des zones "Internet" et "Sites sensibles" doivent être désactivés.
 - Fermer Internet Explorer. Sélectionner l'option "Toujours fermer tous les onglets" lors de la fermeture d'Internet Explorer
 - Ouvrir à nouveau Internet Explorer et vérifier que le mode protégé est désactivé (cela peut être vu dans une boîte de message au bas du navigateur).
- Exécuter la commande binaire du pilote Web.
 - **Pour Microsoft Edge** - ouvrir une invite de commande avec des privilèges d'administrateur et exécuter: `MicrosoftWebDriver.exe --host = 10.0.2.15 --port = 4444 --verbose`
 - **Pour Internet Explorer** - ouvrir l'invite de commande en tant qu'utilisateur sans privilège et exécuter: `C:\Windows\System32\IEDriverServer.exe --host =`

0.0.0.0 --port = 4444 --log-level=DEBUG

- Réduire la fenêtre d'invite de ligne de commande lorsque le pilote est opérationnel.
- Basculer vers le terminal où qemu s'exécute et taper à l'invite de qemu: (qemu) savevm windows
- Arrêter VM: (qemu) quit

Pour démarrer la VM à partir d'un instantané manuellement, utiliser la commande suivante (cf entrypoint.sh) :



```
$ sudo qemu-system-x86_64 -enable-kvm \  
    -machine q35 -smp sockets=1, core=1, threads=2 -m 2048 \  
    -usb -device usb-kbd -device usb-tablet -rtc base = heure  
locale \  
    -net nic, model=user virtio -net, hostfwd=tcp::4444-:4444  
\  
    -drive file=snapshot.img, media=disk, if=virtio \  
    -loadvm windows
```

Construction de l'image Docker

Déplacer les fichiers de système de fichiers et d'état vers le répertoire image de ce référentiel:

```
$ mv hdd.img snapshot.img image  
$ cd image
```

Construire l'image Docker en utilisant le fichier Docker fourni:

```
$ docker build -t windows/edge:18. # Pour Microsoft Edge
```

```
$ docker build -t windows/ie:11. # Pour Internet Explorer
```

Exécuter un conteneur à partir de l'image:

```
$ docker run -it --rm --privileged -p 4444:4444 -p 5900:5900 windows/edge:18  
# pour Microsoft Edge
```

```
$ docker run -it --rm --privileged -p 4444:4444 -p 5900:5900 windows/ie:11 #  
pour Internet Explorer
```

Se connecter à vnc

Pour voir l'écran Windows à l'intérieur du conteneur en cours d'exécution,

```
vnc://localhost:5900
```

en utilisant le sélénoïde comme mot de passe.

From:

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

Permanent link:

<http://www.ouarte.garden/doku.php?id=labs:docker-lab-dc1-edge>

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

