

HERCULES: LAB 1 INSTALLATION DE ZOS1.10 ADCD DANS UNE IMAGE COW

Objet	Installation ADCD 1.10 dans une image COW
Niveau requis	débutant, avisé
Débutant, à savoir	
Suivi	:DONE:

Table of Contents

- [HERCULES: LAB 1 INSTALLATION DE ZOS1.10 ADCD DANS UNE IMAGE COW](#)
 - [Etape 1: Préparation hyperviseur/workstation](#)
 - [Workstation: Installer git rexx x3270 et build-essentials](#)
 - [Hyperviseur: Installer hercules](#)
 - [Tester l'installation de hercules](#)
 - [Etape 2: Obtenir le système de développement d'applications \(ADCD\)](#)
 - [Description des disques fournis](#)
 - [Préparer le système de fichiers.](#)
 - [Etape 3: Création de la VM](#)
 - [Création de l'image QCOW2](#)
 - [Génération de l'image COW](#)
 - [Etape 4: lancement de zos](#)
 - [Montage du système de fichier](#)
 - [Démarrage de la VM](#)
 - [Etape 5: connexion](#)
 - [Etape 6: mise en réseau](#)
 - [Etape 7: tout fermer](#)

Etape 1: Préparation hyperviseur/workstation

Workstation: Installer git rexx x3270 et build-essentials

La workstation est un poste utilisateur depuis lequel on pourra se connecter au système Z

```
$ sudo apt-get install git build-essential libtool automake autoconf  
libregina3-dev screen x3270
```

Hyperviseur: Installer hercules

Installation des RPM prédéfinis:

- Télécharger le fichier RPM

```
$ scp
user@xx.xx.xxx.xx:/data/jacques.nougat/repos/support/centos/noarch/hercules-3.07-1.x86_64.rpm .
```

- Installer le RPM:

```
rpm -Uvh hercules-3.07-1.x86_64.rpm
Préparation... #####
[100%]
 1:hercules #####
[100%]
```

Cela placera les exécutables Hercules dans /usr/bin et les bibliothèques dynamiques dans /usr/lib et /usr/lib/hercules, que l'on peut exécuter n'importe où. Les exemples de fichiers de configuration seront placés dans /etc/hercules et le jeu de cartes IPLable de l'utilitaire autonome ZZSA (zzsacard.bin) sera placé dans /usr/share/hercules.

Tester l'installation de hercules

Dans l'hyperviseur

```
$ hercules -f /etc/hercules/sample.cnf
```

Dans un terminal depuis la workstation

```
c3270 localhost:3270
```

ça devrait ressembler à ça:

```
/=====\
|
| Hercules Version   : 4.00           |
| Host name          : MS              |
| Host OS            : DEBIAN          |
| Host Architecture  : x86             |
| Processors         : MP=2            |
| LPAR Name          : HERCULES        |
| Device number      : 0:001F          |
| Subchannel         : 0004            |
|
| HHH      HHH      The S/370, ESA/390 and z/Architecture   |
| HHH      HHH      Emulator                               |
| HHH      HHH      |                                       |
| HHH      HHH      EEEE RRR   CCC U  U L   EEEE   SSS      |
```

```
| HHHHHHHHHHHHHHHHHHHH E R R C U U L E S |
| HHHHHHHHHHHHHHHHHHHH EEE RRR C U U L EEE SS |
| HHHHHHHHHHHHHHHHHHHH E R R C U U L E S |
| HHH HHH EEEE R R CCC UU LLLL EEEE SSS |
| HHH HHH |
| HHH HHH |
| HHH HHH My PC thinks it's a MAINFRAME |
| |
| Copyright (c) 1999-2011 Roger Bowler, Jan Jaeger, and others |
| |
\=====/
```

pour sortir de la console, utiliser CTRL -] et taper quit

Etape 2: Obtenir le système de développement d'applications (ADCD)

ADCD est créé pour fournir aux éditeurs de logiciels indépendants qualifiés le dernier système z/OS ainsi que les produits middleware associés au dernier niveau de maintenance disponible.

Le téléchargement du DemoPkg nécessite 16 go d'espace. Il est constitué:

- d'un fichier de configuration hercules
- des images cckd des disques

Description des disques fournis

DVD	Disque	Description
Disk 1	zares1.gz	RES Volume 1 - Requis pour l'IPL
	zares2.gz	RES Volume 2 - Requis pour l'IPL
	zasys1.gz	Volume système 1 - Requis pour l'IPL
	zauss1.gz	Volume 1 des services UNIX - Requis pour l'IPL
	zapr1d1.gz	Bibliothèques de produits cibles - Nécessaires pour exécuter les produits sélectionnés
	zapr1d2.gz	Bibliothèques de produits cibles - Nécessaires à l'exécution de produits sélectionnés
Disk 2	zadis1.gz	Distribution Volume 1
	zadis2.gz	Distribution Volume 2
	zadis3.gz	Distribution Volume 3
	zadis4.gz	Distribution Volume 4
	zadis5.gz	Distribution Volume 5
	zadis6.gz	Distribution Volume 5
Disk 3	zapr3d3.gz	Ensembles de données Java HFS pour 31 et 64 bits pour 1.4, 1,6, Jeux de données Java HFS pour 64 bits pour 1.5 Java HFS pour 32 bits 1.5 est sur le volume ZAPRD2.
	zacic1.gz	Target et Dlibs pour CICS 3.2
	zaims1.gz	Target et Dlibs pour IMS 10
	sares1.gz	sares1.gz

DVD	Disque	Description
Disk 4	zadb81.gz	Jeux de données cible et requis pour DB2 v8
	zadb82.gz	Utilitaires DB2 v8 Jeux de données cible et requis
	zadb83.gz	Jeux de données cibles de l'utilitaire Performance Analyzer de DB2 v9
	zadb84.gz	Jeux de données DLIB de l'utilitaire DB2 v9 Performance Analyzer
Disk 5	zawas1.gz	Ensembles de données cibles WAS 6.1, ESB 6.0.1 et WPS 6.0.1
	zawas2.gz	Jeux de données DLIB WAS 6.1, ESB 6.0.1 et WPS 6.0.1
	zawas3.gz	WAS 6.1 Large ZFS dataset

Préparer le système de fichiers.

Une fois le téléchargement terminé appliquer les changements suivants:

- renommer le dossier "IBM\ ZOS\ 1.10/" en "mainframe".

```
$ sudo mv IBM\ ZOS\ 1.10/ mainframe
```

- créer un dossier conf dans le dossier mainframe et y placer le fichier hercules.cnf

```
$ cd mainframe
$ mkdir conf
$ mv hercules.cnf conf
```

- créer un dossier cckd dans le dossier mainframe et y placer les images cckd (la commande ci-dessous permet également de renommer les fichiers
ibm.adcd.zos.1.10.disk.##.of.##.VOLSER.cckd en VOLSER.cckd)

```
$ mkdir cckd
$ for FILE in `ls *.cckd`; do NAME=`echo "$FILE" | cut -d'.' -f10`; mv $FILE "cckd/$NAME.cckd"; done
```

- créer un dossier appelé prtr dans le dossier mainframe. On doit maintenant avoir trois dossiers: conf, dasd et prtr

```
├── cckd
│   ├── sares1.cckd
│   ├── zacic1.cckd
│   ├── zadb81.cckd
│   ├── zadb82.cckd
│   ├── zadb83.cckd
│   ├── zadb84.cckd
│   ├── zadb91.cckd
│   ├── zadb92.cckd
│   ├── zadis1.cckd
│   ├── zadis2.cckd
│   ├── zadis3.cckd
│   ├── zadis4.cckd
│   └── zadis5.cckd
```

```

├── zadis6.cckd
├── zaims1.cckd
├── zaprd1.cckd
├── zaprd2.cckd
├── zaprd3.cckd
├── zaprd4.cckd
├── zares1.cckd
├── zares2.cckd
├── zasys1.cckd
├── zauss1.cckd
├── zawas1.cckd
├── zawas2.cckd
├── zawas3.cckd
├── conf
│   └── hercules.cnf
└── prtr

```

- Créer l'archive tar.gz avec la structure ainsi dévinie

```
$ tar zcvf adcd-zos-110.tar.gz *
```

Etape 3: Création de la VM

Dans l'exemple on utilise le mode «Copy-On-Write» (COW), car:

- aucune modification n'est appliquée à l'image disque d'origine, toutes les modifications sont enregistrées dans un fichier séparé en préservant l'image d'origine.
- plusieurs fichiers COW peuvent pointer sur la même image pour tester plusieurs configurations simultanément sans mettre en péril le système de base.

Création de l'image QCOW2

Virt-make-fs est un outil de ligne de commande permettant de créer une image de système de fichiers à partir d'une archive tar ou de certains fichiers d'un répertoire. Il peut créer des types de systèmes de fichiers courants tels que ext2/3 ou NTFS.

- Télécharger l'archive tar.gz contenant les images disques de la LPAR

```
$ scp user@xx.xx.xxx.xx:/data/jacques.nougat/repos/support/ibm/adcd-
zos-110.tar.gz /var/www/html/hercules/
```

adcd-zos-110.tar.gz	100%	16GB	11.2MB/s	24:05
---------------------	------	------	----------	-------

- Créer l'image QCOW2

```
$ virt-make-fs --format=qcow2 --size=+2G adcd-zos-110.tar.gz adcd-zos-110-
```

e2.qcow2

Génération de l'image COW

Les générations d'images COW sont des images créées simplement par l'utilitaire `qemu-img` :

```
$ qemu-img create -f <Format> -b <nom de l'image de référence> <nom de l'image COW>
```

```
$ qemu-img create -f qcow2 -b adcd-zos-110-e2.qcow2 zos110.qcow2
Formatting 'zos110.qcow2', fmt=qcow2 size=21146817024 backing_file='adcd-zos-110-e2.qcow2' encryption=off cluster_size=65536
```

L'image créée est plus petite que l'image de référence (193K pour une référence d'environ 17G) car pendant la création, il n'y a pas de différence entre l'original et la COW.

```
17G -rw-r--r-- 1 root root 17G 1 avril 16:18 adcd-zos-110-e2.qcow2
16G -rw-r--r-- 1 root root 16G 1 avril 16:05 adcd-zos-110.tar.gz
196K -rw-r--r-- 1 root root 193K 1 avril 16:21 zos110.qcow2
```

Note: On peut générer d'autres images COW à partir de la même référence.

```
$ qemu-img create -f qcow2 -b adcd-zos-110-e2.qcow2 zos110-2.qcow2
Formatting 'zos110-2.qcow2', fmt=qcow2 size=21146817024 backing_file='adcd-zos-110-e2.qcow2' encryption=off cluster_size=65536
```

```
$ ls -lsah
total 33G
4,0K drwxr-xr-x 2 root root 4,0K 1 avril 16:26 .
4,0K drwxr-xr-x 8 root root 4,0K 1 avril 10:38 ..
17G -rw-r--r-- 1 root root 17G 1 avril 16:18 adcd-zos-110-e2.qcow2
16G -rw-r--r-- 1 root root 16G 1 avril 16:05 adcd-zos-110.tar.gz
196K -rw-r--r-- 1 root root 193K 1 avril 16:26 zos110-2.qcow2
196K -rw-r--r-- 1 root root 193K 1 avril 16:21 zos110.qcow2
```



pour assurer la préservation de l'image d'origine, on peut la protéger en écriture `chmod -aw adcd-zos-110-e2.qcow2`.

Etape 4: lancement de zos

Montage du système de fichier

```
$ guestmount -a /var/www/html/hercules/zos110.qcow2 -m /dev/sda /mnt
```

Démarrage de la VM

```
$ /hercules -f /mnt/conf/hercules.conf
```

dans la workstation, et dans un terminal séparé, se connecter avec x3270

```
x3270 localhost:23
```

maintenant de retour à la fenêtre hercules faire l'ipl:

```
ipl a80
```

Retourner sur le terminal x3270 et attendre les BOOT de l'environnement zos.

Répondre au message 00 - R 00,I

```
IEA247I USING IEASYS00 FOR OS/390 02.10.00 HBB7703
IEA598I TIME ZONE = W.00.00.00
IXL011I XES HARDWARE SUPPORT IS NOT INSTALLED. REASON: 02
IXC414I CANNOT JOIN SYSPLEX TESTPLX WHICH IS RUNNING IN MONOPLEX MODE:
  CONFIGURATION REQUIREMENT
IXC404I SYSTEM(S) ACTIVE OR IPLING: DEMOPKG
| IXC420D REPLY I TO INITIALIZE SYSPLEX TESTPLX, OR R TO REINITIALIZE XCF
R 00,I
IEE600I REPLY TO 00 IS;I
IXC413I MULTISYSTEM SYSPLEX CONFIGURATION PREVENTED BY PLEXCFG=MONOPLEX
IXC413I XCFLOCAL SYSPLEX CONFIGURATION PREVENTED BY PLEXCFG=MONOPLEX
IXC418I SYSTEM DEMOPKG IS NOW ACTIVE IN SYSPLEX TESTPLX
```

une fois que tout est terminé ouvrir une troisième fenêtre et se connecter à «localhost» ça devrait ressembler à ça

```
/=====
====\
|                               |
| Mainframe Operating System   | z/OS
V1.10|
|                               |
|                               |
|                               |
|                               |
```

```

|   DDD   U   U ZZZZZ AAA      |
|   D   D U   U   Z  A   A      |
|   D   D U   U   Z  AAAAA      |
|   D   D U   U   Z  A   A      |
|   DDD   UUU  ZZZZZ A   A      |
|                                     |
|                                     |
|Welcome to DUZA Mainframe System!      |
|                                     |
| TS0      - Logon to TS0/ISPF          SA      - System Automation      |
| CICS      - CICS System                TWS      - Workload Scheduler      |
| IMS       - IMS System                  |
|                                     |
| Enter your choice==>                  |
| Enter one of above commands in red      |
|                                     |
| Your IP(           :           ), SNA LU(LCL702 )      01/17/13 03:55:47|
|                                     |
\=====
====/

```

Etape 5: connexion

Au “=>” prompt entrer “tso”. fournir un id d'utilisateur (par défaut “ibmuser”). et un mot de passe (par défaut entrez “sys1”).

Eventuellement, on peut voir ispf, suivi de ***. taper sur enter.

bienvenue à zos!

Voilà, c'est tout ce qu'il y a à vraiment à faire. maintenant on va apprendre à installer gccmvs ou à compiler des programmes c dans omvs



Pour passer en environnement UNIX sur la commande prompt ==> taper “tso omvs”



facultatif mais recommandé

Etape 6: mise en réseau

- A partir de l'écran de menu ispf "3.4"
- Remplacer "ibmuser" avec "duza" et hit enter
- Taper page down et trouver "duza.tcparms"
- sur cette meme ligne trois caracteres a gauche de la lettre "d" in "duza" taper la lettre "e" et appuyer sur enter
- sur la première ligne blanche à proximité de "profil", taper la lettre "e" et appuyer sur page suivante à la ligne 90, qui ressemble à cela :

```
000090 DEVICE LCS1   LCS           E20
000091 LINK ETH1   ETHERNET   0   LCS1
000092
000093 HOME
000094     10.0.1.20     ETH1
000095
000096 GATEWAY
000097     10.0.1.100 = ETH1 1500 HOST
000098
000099 DEFAULTNET 10.0.1.100 ETH1 1500 0
```

- faire en sorte que cela ressemble à ceci:

```
000090 DEVICE CTCA1 CTC e20
000091 LINK CTC1 CTC 1 CTCA1
000092
000093 HOME
000094     192.168.0.210   CTC1
000095
000096 GATEWAY
000097     192.168.0.1    = CTC1 1492 HOST
000098
000099 DEFAULTNET 192.168.0.5 CTC1 1492 0
```



si on n'est pas sur un réseau 192.168.0.*, il faut faire correspondre à la première adresse ip pour être celle définie dans le fichier .conf. la seconde est l'adresse ip du routeur et la troisième est l'adresse ip de la machine host hercules

- quand c'est fait sur la ligne Command ==> taper save et puis end
- revenir a l'autre c3270 et taper stop tcpip, puis start tcpip
- dans linux dans un nouveau terminal, saisir les commandes suivantes:

```
echo "1" > /proc/sys/net/ipv4/conf/all/proxy_arp
echo "1" > /proc/sys/net/ipv4/conf/all/forwarding
```

- alors essayer de se connecter sur l'image principale: telnet 192.168.0.210 1023 (user/pass:

ibmuser/sys1)

Etape 7: tout fermer

- dans le premier c3270 ouvert taper s shutdown et puis taper z eod
- attendre une minute ou deux
- dans la fenêtre hercules taper exit
- pour démarrer à nouveau suivre l'étape 6

From:

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

Permanent link:

<http://www.ouarte.garden/doku.php?id=labs:hercules-lab-1-zos110>

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

