Cloning WebSphere, DB2 and WebSphere MQ on Linux under z/VM

Michael MacIsaac - mikemac@us.ibm.com
Monday February 12th, 3:00 PM
Session 9210
Abstract

Creating WebSphere Application Server, DB2 and WebSphere MQ servers typically involves installing an **operating system**, then installing and configuring the **software**. The **data** associated with each solution also resides on the same server. When it is time to upgrade to a new release, each server must be upgraded individually. As the number of these types of servers increases, upgrading becomes cumbersome. This presentation describes how to install the middleware's executable files on virtual Linux servers under z/VM. Other virtual servers can be cloned and configured to link these servers' disks read-only. In this fashion the cloned servers access the executables read-only while the application data is read-write. When a new version of middleware needs to be rolled out, it can be installed onto another set of disks on the masters for testing purposes. This presentation will give a live demonstration of the middleware cloning process.
Who am I?, who are you?

- Mike MacIsaac, mikemac@us.ibm.com
  - 20 years at IBM in Kingston and Poughkeepsie, NY
  - Programmer early in career
  - OS/390, USS and Linux redbook project leader
  - Marketing technical support of z/VM, Linux, IBM software, ...
  - Wrote much of *z/VM and Linux on IBM System z: The Virtualization Cookbook v2*
    - [http://linuxvm.org/present](http://linuxvm.org/present)
    - Describes all steps in detail
  - This section was written with Carlos Ordonez of IBM Poughkeepsie

- Who are you?
  - System z Linux in production?
  - System z Linux in test/proof of concept?
  - Tried cloning IBM middleware?
Short history of this presentation

- Gave similar presentation at Baltimore SHARE, August 2006
  - Live demo cloning one Linux server with WAS, DB2 and MQSeries
  - Goal is to ease the maintenance of many virtual servers with common middleware
- Disclaimers:
  - I do not work for WAS, DB2 nor MQSeries development
  - I do not have this middleware in production
  - Published book has the text:
    - TODO: Describe upgrading WebSphere V6.0.2 to V6.1
    - It was not simple to migrate to V6.1 - got it to work once
  - This environment is somewhat (b)leading edge
  - MQSeries will not support the environment described!!
    - Because RPMs have not been installed on target system (RPM runs a big post script)
    - "We understand that customers would like additional mechanisms for installation, and we are always looking at things that might be built into future versions of the product,..."
      - marke_taylor at uk.ibm.com
- However:
  - The demo was live, not canned
  - WAS has introduced the concepts of profiles (all r/w data)
  - Hope to add new chapter "Sharing IBM Middleware binaries read-only" to SLES 10 book
  - Demand for virtualization will drive this type of support/usage
First an aside: Thinking about systems

- Computing model in the dark ages - no OS needed
Thinking about systems (cont'd)

- Processing another data set
Thinking about systems (cont'd)

- Running a different program
- It was very easy to keep programs and data separate
Thinking about systems (cont'd)

- Computing model today
  - How do your systems look?

Computing model hierarchy:
- Hardware
- Operating System
- Applications
- Data
Thinking about systems (cont'd)

- What is the ideal system?
# FHS summary

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Root file system (must be able to boot/repair)</td>
</tr>
<tr>
<td>/bin</td>
<td>Essential commands (static)</td>
</tr>
<tr>
<td>/boot</td>
<td>Static files of the boot loader (static)</td>
</tr>
<tr>
<td>/dev</td>
<td>Device files (static, maintained by OS)</td>
</tr>
<tr>
<td>/etc</td>
<td>Host-specific system configuration (static)</td>
</tr>
<tr>
<td>/etc/opt</td>
<td>Add-on application configuration (site specific)</td>
</tr>
<tr>
<td>/home</td>
<td>User's home directories (site-specific, optional)</td>
</tr>
<tr>
<td>/lib</td>
<td>Essential shared libraries and kernel modules</td>
</tr>
<tr>
<td>/media</td>
<td>Mount point for removable media (N/A on System z)</td>
</tr>
<tr>
<td>/mnt</td>
<td>Temporary mount point (usually empty)</td>
</tr>
<tr>
<td>/opt</td>
<td>Add-on application software packages (site-specific)</td>
</tr>
<tr>
<td>/root</td>
<td>Root user home directory (static, optional)</td>
</tr>
<tr>
<td>/sbin</td>
<td>Essential system binaries (static)</td>
</tr>
<tr>
<td>/srv</td>
<td>Data for services provided by this system (site specific)</td>
</tr>
<tr>
<td>/tmp</td>
<td>Temporary files (delete when system is booted?)</td>
</tr>
<tr>
<td>/usr</td>
<td>Sharable read-only data</td>
</tr>
<tr>
<td>/usr/bin</td>
<td>Most user commands</td>
</tr>
<tr>
<td>/usr/include</td>
<td>Header files included by C programs</td>
</tr>
<tr>
<td>/usr/lib</td>
<td>Libraries</td>
</tr>
<tr>
<td>/usr/local</td>
<td>Local hierarchy (empty after main installation)</td>
</tr>
<tr>
<td>/usr/sbin</td>
<td>Non-vital system binaries</td>
</tr>
<tr>
<td>/usr/share</td>
<td>Architecture-independent data</td>
</tr>
<tr>
<td>/var</td>
<td>Variable data</td>
</tr>
<tr>
<td>/var/cache</td>
<td>Application cache data</td>
</tr>
<tr>
<td>/var/lib</td>
<td>Variable state information</td>
</tr>
<tr>
<td>/var/local</td>
<td>Variable data for /usr/local</td>
</tr>
<tr>
<td>/var/lock</td>
<td>Lock files</td>
</tr>
<tr>
<td>/var/log</td>
<td>Log files and directories</td>
</tr>
<tr>
<td>/var/opt</td>
<td>Variable data for /opt</td>
</tr>
<tr>
<td>/var/run</td>
<td>Data relevant to running processes</td>
</tr>
<tr>
<td>/var/spool</td>
<td>Application spool data</td>
</tr>
<tr>
<td>/var/tmp</td>
<td>Temporary files preserved between system reboots</td>
</tr>
</tbody>
</table>

# FHS summary

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Root file system (must be able to boot/repair)</td>
</tr>
<tr>
<td>/bin</td>
<td>Essential commands (static)</td>
</tr>
<tr>
<td>/boot</td>
<td>Static files of the boot loader (static)</td>
</tr>
<tr>
<td>/dev</td>
<td>Device files (static, maintained by OS)</td>
</tr>
<tr>
<td>/etc</td>
<td>Host-specific system configuration (static)</td>
</tr>
<tr>
<td>/etc/opt</td>
<td>Add-on application configuration (site specific)</td>
</tr>
<tr>
<td>/home</td>
<td>User's home directories (site-specific, optional)</td>
</tr>
<tr>
<td>/lib</td>
<td>Essential shared libraries and kernel modules</td>
</tr>
<tr>
<td>/media</td>
<td>Mount point for removeable media (N/A on System z)</td>
</tr>
<tr>
<td>/mnt</td>
<td>Temporary mount point (usually empty)</td>
</tr>
<tr>
<td>/opt</td>
<td>Add-on application software packages (site-specific)</td>
</tr>
<tr>
<td>/root</td>
<td>Root user home directory (static, optional)</td>
</tr>
<tr>
<td>/sbin</td>
<td>Essential system binaries (static)</td>
</tr>
<tr>
<td>/srv</td>
<td>Data for services provided by this system (site specific)</td>
</tr>
<tr>
<td>/tmp</td>
<td>Temporary files (delete when system is booted?)</td>
</tr>
<tr>
<td>/usr</td>
<td>Sharable read-only data</td>
</tr>
<tr>
<td>/usr/bin</td>
<td>Most user commands</td>
</tr>
<tr>
<td>/usr/include</td>
<td>Header files included by C programs</td>
</tr>
<tr>
<td>/usr/lib</td>
<td>Libraries</td>
</tr>
<tr>
<td>/usr/local</td>
<td>Local hierarchy (empty after main installation)</td>
</tr>
<tr>
<td>/usr/sbin</td>
<td>Non-vital system binaries</td>
</tr>
<tr>
<td>/usr/share</td>
<td>Architecture-independent data</td>
</tr>
<tr>
<td>/var</td>
<td>Variable data</td>
</tr>
<tr>
<td>/var/cache</td>
<td>Application cache data</td>
</tr>
<tr>
<td>/var/lib</td>
<td>Variable state information</td>
</tr>
<tr>
<td>/var/local</td>
<td>Variable data for /usr/local</td>
</tr>
<tr>
<td>/var/lock</td>
<td>Lock files</td>
</tr>
<tr>
<td>/var/log</td>
<td>Log files and directories</td>
</tr>
<tr>
<td>/var/opt</td>
<td>Variable data for /opt</td>
</tr>
<tr>
<td>/var/run</td>
<td>Data relevant to running processes</td>
</tr>
<tr>
<td>/var/spool</td>
<td>Application spool data</td>
</tr>
<tr>
<td>/var/tmp</td>
<td>Temporary files preserved between system reboots</td>
</tr>
</tbody>
</table>
OS/Apps/Data view and virtualization - some models

Key:
- Shared /usr/ r/o
- r/w
- Shared r/o

1. Applications
   - Data
   - Operating System

2. Applications
   - Operating System
   - Data

3. Applications
   - Data
   - Operating System

4. Applications
   - Operating System
   - Data
Linux cloning (one method) - block diagram

- Process described in detail for:
  - SLES 10 - Session 9216
    - *The Virtualization Cookbooks* ...
    - Thu 9:30 AM - Room 24
  - RHEL 5 - Session 9217
    - Thu 4:30 PM - Room 24
Cloning middleware - define 3 new user IDs

- **z/VM user directory for middleware user IDs:**

<table>
<thead>
<tr>
<th>USER</th>
<th>LNX4VM</th>
<th>512M</th>
<th>1G</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MQS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **WebSphere master**
  - USER WAS LNX4VM 512M 1G G
  - INCLUDE LNXDFLT
  - MDISK 100 3390 0001 3038 MMA71F MR LNX4VM LNX4VM LNX4VM
  - MDISK 102 3390 3039 0300 MMA71F MR LNX4VM LNX4VM LNX4VM
  - MINIOPT NOMDC
  - MDISK 300 3390 0001 3338 MMA720 MR LNX4VM LNX4VM LNX4VM
  - MDISK 400 3390 0001 3338 MMA721 MR LNX4VM LNX4VM LNX4VM

  **root file system**

  **swap space**

  **WAS production - full**

  **WAS test - full 3390-3**

  **DB2 master**

<table>
<thead>
<tr>
<th>USER</th>
<th>LNX4VM</th>
<th>512M</th>
<th>1G</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MQS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **DB2 production - half**

  **DB2 test - half 3390-3**

  **MQSeries master - Don't define**

<table>
<thead>
<tr>
<th>USER</th>
<th>LNX4VM</th>
<th>512M</th>
<th>1G</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **MQSeries master - Don't define**

<table>
<thead>
<tr>
<th>USER</th>
<th>LNX4VM</th>
<th>512M</th>
<th>1G</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **MQSeries master - Don't define**

<table>
<thead>
<tr>
<th>USER</th>
<th>LNX4VM</th>
<th>512M</th>
<th>1G</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **MQSeries master - Don't define**
## Cloning middleware - user IDs

<table>
<thead>
<tr>
<th>Software</th>
<th>Normal status</th>
<th>z/VM User ID of &quot;master&quot;</th>
<th>Important minidisks (production and test)</th>
<th>Mount point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux master</td>
<td>Shut down</td>
<td>SLES9X</td>
<td>100: root fs</td>
<td>/sles9xmaster</td>
</tr>
<tr>
<td>Linux controller</td>
<td>Running</td>
<td>SLES9X</td>
<td>200: root fs</td>
<td>/</td>
</tr>
<tr>
<td>WebSphere</td>
<td>Logged off</td>
<td>WAS</td>
<td>300 and 400</td>
<td>/opt/IBM/Websphere/</td>
</tr>
<tr>
<td>DB2</td>
<td>Logged off</td>
<td>DB2</td>
<td>301 and 401</td>
<td>/opt/IBM/db2/</td>
</tr>
<tr>
<td>MQSeries</td>
<td>Logged off</td>
<td>MQS</td>
<td>302 and 402</td>
<td>/opt/mqm/</td>
</tr>
</tbody>
</table>
Cloning middleware - concept

- z/VM user directory for middleware clones:

  USER LINUX05 LNX4VM 512M 1G EG
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3038 MMA71C MR LNX4VM LNX4VM LNX4VM
  MDISK 102 3390 3039 0300 MMA71C MR LNX4VM LNX4VM LNX4VM
  MINIOPT NOMDC
  LINK WAS 300 300 RR
  LINK WAS 400 400 RR
  LINK DB2 301 301 RR
  LINK DB2 401 401 RR
  LINK MQS 302 302 RR
  LINK MQS 402 402 RR

  Clone target user ID
  root file system
  swap space
  WAS production/test read-only link
  DB2 production/test read-only link
  MQSeries production/test read-only link - DON'T USE!

  USER LINUX06 LNX4VM 512M 1G G
  ...
  USER LINUX07 LNX4VM 512M 1G G
  ...
Cloning middleware - cloned Linux

- Block diagram
  - MQSeries updated

```
LINUX05 100 - read-write
/
usr/ var/ tmp/ ... dev/ opt/
IBM/
wasprofiles/
WAS 300 WebSphere/ read-only
DB2 301 db2/ read-only
MQS 302 mqm/ read-only
```
Cloning WebSphere manually

- From the controller, clone a vanilla Linux
  
  ```
  # clone.sh linux05
  ...
  ```

- SSH into new server and activate WebSphere production disk (300)
  
  ```
  # chccwdev -e 0.0.0300
  ```

- Modify `/etc/zipl.conf` to access disks read-only and run `zipl`
  
  ```
  [ipl]
  target = /boot/zipl
  image = /boot/image
  ramdisk = /boot/initrd,0x1000000
  parameters = "root=/dev/dasda1 dasd=100-102,300(ro),400(ro) selinux=0 TERM=dumb"
  
  # zipl
  ```

- Modify `/etc/fstab` to mount disks read-only
  
  ```
  /dev/dasda1 / ext3 acl,user_xattr 1 1
  /dev/dasdb1 swap swap pri=42 0 0
  /dev/dasdc1 swap swap pri=42 0 0
  LABEL=was-prod /opt/IBM/WebSphere ext2 ro,acl,user_xattr 0 0
  devpts /dev/pts devpts mode=0620,gid=5 0 0
  proc /proc proc defaults 0 0
  sysfs /sys sysfs noauto 0 0
  ```
Cloning WebSphere manually (cont'd)

- Mount new r/o file system
  ```
  # mount /opt/IBM/WebSphere
  # mount | grep opt
  /dev/dasdd1 on /opt/IBM/WebSphere type ext2 (ro,acl,user_xattr)
  ```

- Reboot to test changes
  ```
  # reboot
  ```

- Copy the wasprofile.properties file to the r/w /opt/IBM/wasprofiles/ directory
  ```
  # cd /opt/IBM/wasprofiles
  # mkdir properties
  # cd properties
  # cp /opt/IBM/WebSphere/AppServer/properties/wasprofile.properties .
  ```

- Add the WebSphere bin/ directory to the PATH
  ```
  # cat /root/.bash_profile
  export PATH=$PATH:/opt/IBM/wasprofiles/AppSrv01/bin
  ```
Cloning WebSphere manually (cont'd)

► Create a script that is a wrapper around the `wasprofile.sh` command

```
# cat /usr/local/sbin/wasprofile
#!/bin/bash
HOST=`hostname -s`/opt/IBM/WebSphere/AppServer/bin/wasprofile.sh 
  -create 
  -profileName AppSrv01 
  -profilePath /opt/IBM/wasprofiles/AppSrv01 
  -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/default 
  -nodeName "$HOST"Node01 
  -cellName "$HOST"Node01Cell 
  -hostName "$HOST".pbm.ihost.com
```

► Run the `wasprofile` script and view the resulting directory

```
# wasprofile
INSTCONFSUCCESS: Success: The profile now exists
# ls /opt/IBM/wasprofiles/AppSrv01
bin/ etc/ installableApps/ installedConnectors/ properties/ wstemp/
config/ firststeps/ installedApps/ logs/ temp/
```

► Start the server

```
# startServer.sh server1
...
ADMU3000I: Server server1 open for e-business; process id is 1340
```
Test WebSphere

- http://129.40.178.133:90
FHS summary

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Root file system (must be able to boot/repair)</td>
</tr>
<tr>
<td>/bin</td>
<td>Essential commands (static)</td>
</tr>
<tr>
<td>/boot</td>
<td>Static files of the boot loader (static)</td>
</tr>
<tr>
<td>/dev</td>
<td>Device files (static, maintained by OS)</td>
</tr>
<tr>
<td>/etc</td>
<td>Host-specific system configuration (static)</td>
</tr>
<tr>
<td>/etc/opt</td>
<td>Add-on application configuration (site specific)</td>
</tr>
<tr>
<td>/home</td>
<td>User's home directories (site-specific, optional)</td>
</tr>
<tr>
<td>/lib</td>
<td>Essential shared libraries and kernel modules</td>
</tr>
<tr>
<td>/media</td>
<td>Mount point for removable media (N/A on System z)</td>
</tr>
<tr>
<td>/mnt</td>
<td>Temporary mount point (usually empty)</td>
</tr>
<tr>
<td>/opt</td>
<td>Add-on application software packages (site-specific)</td>
</tr>
<tr>
<td>/opt/IBM/WebSphere</td>
<td>Websphere application</td>
</tr>
<tr>
<td>/opt/IBM/db2</td>
<td>DB2 application</td>
</tr>
<tr>
<td>/opt/mqm</td>
<td>MQSeries Application</td>
</tr>
<tr>
<td>/opt/IBM/wasprofiles</td>
<td>WebSphere application data</td>
</tr>
<tr>
<td>/root</td>
<td>Root user home directory (static, optional)</td>
</tr>
<tr>
<td>/sbin</td>
<td>Essential system binaries (static)</td>
</tr>
<tr>
<td>/srv</td>
<td>Data for services provided by this system (site specific)</td>
</tr>
<tr>
<td>/tmp</td>
<td>Temporary files (delete when system is booted?)</td>
</tr>
<tr>
<td>/usr</td>
<td>Sharable read-only data</td>
</tr>
<tr>
<td>/var</td>
<td>Variable data</td>
</tr>
<tr>
<td>/var/cache</td>
<td>Application cache data</td>
</tr>
<tr>
<td>/var/lib</td>
<td>Variable state information</td>
</tr>
<tr>
<td>/var/local</td>
<td>Variable data for /usr/local</td>
</tr>
<tr>
<td>/var/lock</td>
<td>Lock files</td>
</tr>
<tr>
<td>/var/log</td>
<td>Log files and directories</td>
</tr>
<tr>
<td>/var/opt</td>
<td>Variable data for /opt</td>
</tr>
<tr>
<td>/var/run</td>
<td>Data relevant to running processes</td>
</tr>
<tr>
<td>/var/spool</td>
<td>Application spool data</td>
</tr>
<tr>
<td>/var/tmp</td>
<td>Temporary files preserved between system reboots</td>
</tr>
</tbody>
</table>
Automating cloning of WebSphere

Create new script in master image's services directory

```bash
# cd /sles9master/etc/init.d/
# cat wasprofile
#!/bin/bash
echo "$0 - Creating a WAS profile ..." | tee /tmp/wasprofile.log
host=`hostname -s`
domain=`hostname -d`/opt/IBM/WebSphere/AppServer/bin/
wasprofile.sh -create \\
    -profileName AppSrv01 \\
    -profilePath /opt/IBM/wasprofiles/AppSrv01 \\
    -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/default \\
    -nodeName ${host}Node01 \\
    -cellName ${host}Node01Cell \\
    -hostName ${host}.${domain}
echo "Return code from wasprofile.sh = $?" | tee -a /tmp/wasprofile.log
symlink="/etc/init.d/rc5.d/S99wasprofile"
if [ -h $symlink ]; then # remove symlink so script runs once
    rm $symlink
    echo "Removed symbolic link $symlink" | tee -a /tmp/wasprofile.log
fi
/opt/IBM/wasprofiles/AppSrv01/bin/startServer.sh server1 | \
    tee -a /tmp/wasprofile.log
```

Create symbolic link

```bash
# cd rc5.d
# ln -s ../wasprofile S99wasprofile
```
Automating cloning of WebSphere (cont'd)

▶ Add a function to the clone.sh script

```bash
#+--------------------------------------------------------------------------+
function clone_was()
#   Arg 1: Mount point of the newly cloned server
# Clone a WebSphere Application Server
#+--------------------------------------------------------------------------+
{
  echo "Cloning WebSphere ..."
  echo "Modifying zipl.conf and running zipl ..."
  cd $1/etc
  cp zipl.conf zipl.conf.orig
  sed -i -e 's:dasd=100-102:dasd=100-102,300(ro),400(ro):g' zipl.conf
  chroot $1 zipl
  echo "Modifying fstab ..."
  cp fstab fstab.orig
  sed -i -e '4a LABEL=was-prod /opt/IBM/WebSphere ext2' fstab
  echo "making symlink to /etc/init.d/wasprofile ..."
  cd $1/etc/init.d/rc5.d
  ln -s ../wasprofile S99wasprofile
  cd
}
```
Clone a WebSphere server with automation

➤ Use the clone.sh script with the --was flag

```bash
# clone.sh --was linux05
...
```

WARNING!!: this will copy 100 and 102 disks to LINUX01 100 and 102
New host name will be: lat135.pbm.ihost.com
New TCP/IP address will be: 129.40.178.135
Other network data is retrieved from LINUX01 PARMFILE on 191 disk
A WebSphere system will be cloned
Are you sure you want to overwrite these disks (y/n): y
...

➤ Jump to the 3270 console after clone is XAUTOLOGed
...

```
/etc/init.d/rc5.d/S99wasprofile - Creating a WAS profile ...
Jun 22 12:33:13 lat133 kernel: eth0: no IPv6 routers present
INSTCONFSUCCESS: Success: The profile now exists.
Return code from wasprofile.sh = 0
Removed symbolic link /etc/init.d/rc5.d/S99wasprofile
ADMU0116I: Tool information is being logged in file
        /opt/IBM/wasprofiles/AppSrv01/logs/server1/startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 2118
...
```

➤ Test WebSphere
DB2

- Same process as with WAS in general - how to create an instance

```bash
echo "Creating a DB2 instance ..." | tee /tmp/db2instance.log
mkdir /tmp/noskel # create a DB2 instance user and group
groupadd db2grp1
useradd -g db2grp1 -m -k /tmp/noskel db2inst1

touch /home/db2inst1/.profile

useradd -g db2fgrp1 -m -k /tmp/noskel db2fenc1
groupadd db2agrp1 # create a DB2 administrative user and group
useradd -g db2agrp1 -m -k /tmp/noskel db2admn1
touch /home/db2admn1/.profile

/opt/IBM/db2/V8.1/instance/db2icrt -a SERVER -s ese -u db2fenc1 -p db2c_db2inst1/db2inst1
echo "Return code from db2icrt = $?" | tee -a /tmp/db2instance.log

/opt/IBM/db2/V8.1/instance/dascrt -u db2admn1
echo "Return code from dascrt = $?" | tee -a /tmp/db2instance.log

cp /opt/IBM/db2/V8.1/das/dasprofile /home/db2admn1/das

su - db2inst1 -c "db2iauto -on db2inst1"
echo "Return code from db2iauto = "$?" | tee -a /tmp/db2instance.log

symlink="/etc/init.d/rc5.d/S98db2instance"
if [ -h $symlink ]; then # remove symlink so script runs once
  rm $symlink
  echo "Removed symbolic link $symlink" | tee -a /tmp/db2instance.log
fi

su - db2inst1 -c "db2start"
echo "Return code from db2start = "$?" | tee -a /tmp/db2instance.log
```
MQSeries

- DO NOT follow the steps in the book
  - They will work to create a single queue, but have not been tested
  - They are not supported by MQSeries development

- Alternate approach
  - Install MQSeries RPMs into the "master (golden) image"
  - Cloned virtual servers will have MQSeries
  - Pros:
    - Supported
    - Diverse levels of MQSeries releases
  - Cons:
    - No shared binaries
    - More work to upgrade
    - All cloned virtual servers will have MQSeries
Personnel roles

- Many possible roles
  - z/VM sysadmin (MAINT, etc)
  - z/VM Network admin (TCPMAINT)
  - Linux sysadmin (LNXMAINT, SLES9X, SLES9)
  - Middleware admins
    - WebSphere admin (WAS)
    - DB2 admin (DB2)
    - MQ Series admin (MQS)
  - Application owners
    - WebSphere (LINUXnn)
    - DB2 (LINUXnn)
    - MQ Series (LINUXnn)
Resources

- Book *z/VM and Linux on IBM System z: The Virtualization Cookbook Version 2* (SLES 9)
  - [http://linuxvm.org/present/misc/virt-cookbook-2.pdf](http://linuxvm.org/present/misc/virt-cookbook-2.pdf)
  - Contains all steps in detail *(Don't follow MQSeries steps!)*
- Files associated with the book
  - [http://linuxvm.org/present/misc/virt-cookbook-2.tgz](http://linuxvm.org/present/misc/virt-cookbook-2.tgz)
- *The* Linux for zSeries and S/390 portal
  - [http://linuxvm.org/](http://linuxvm.org/)
- The linux-390 list server
  - [http://www2.marist.edu/htbin/wlvindex?linux-390](http://www2.marist.edu/htbin/wlvindex?linux-390)
- Linux for zSeries and S/390 developerWorks®
- SUSE LINUX Enterprise Server 9 evaluation
- z/VM publications
- z/VM performance tips
Questions - ???

Q: What is the answer to life?
A: 42

But what is the question of life?

Contact:
Mike MacIsaac
mikemac at us.ibm.com
1-845-433-7061
Timeline of project

Project started: 11/04

2004

1 Redbook published *From LPAR to Virtual Servers in Two Days*, SG24-6695-00: 6/05

2 The Virtualization Cookbook published on linuxvm.org, 2/06

3 Redbook published *The Virtualization Cookbook for SLES9*, SG24-6695-01, 4/06

4 Redbook: *The Virtualization Cookbook for RHEL4*, SG24-7272-00, 9/06

5 The Virtualization Cookbook 2 published on linuxvm.org, 8/06

6 Redbook published *The Virtualization Cookbook(s) for RHEL 5 and SLES 10*, 2/07

7 The Virtualization Cookbook(s) published on linuxvm.org, 8/06

2005

2006

2007

Announcing!