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

Michael MacIsaac - mikemac@us.ibm.com
Tuesday, August 15th, 1:30 PM
Session 1189/2954/9310

Abstract

Creating WebSphere Application Server, DB2 and WebSphere MQ Series 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: call these the master servers. Other virtual servers can be cloned and configured to link the master servers' disks read-only: call these the clones. In this fashion the clones 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. When that updated version of middleware is deemed acceptable, the clones can easily switch to the new level.
Who am I?, who are you?

- Mike MacIsaac, mikemac@us.ibm.com
  - 20 years at IBM in NY
  - z/VM and Linux evangelist
  - Wrote much of *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9*
- Who are you?
  - No Linux on zSeries?
  - Testing Linux/PoC?
  - Linux in production?
  - WebSphere Application Server in production on any platform?
  - DB2 in production?
  - MQ Series in production?

Outline for the presentation

- Overview
- Roles
- Setting up a cloning infrastructure
  - z/VM
  - Linux
- WebSphere cloning
  - Installing on the "master"
  - Setting up a clone manually
  - Automating cloning
- DB2 cloning
  - Installing on the "master"
  - Setting up a clone manually
  - Automating cloning
- MQ Series cloning
  - Installing on the "master"
  - Setting up a clone manually
  - Automating cloning
Overview: assumptions

- Discusses some IBM middleware products
  - WebSphere Application Server V6.0.2 (V6.1?)
  - DB2 V8.2
  - MQ Series V6.0
- Install binaries on a master server
- Link binaries R/O and create application data R/W on clones
  - WebSphere uses profiles
  - DB2 uses instances
  - MQ Series uses queue managers
- Use 2 disks on each of the masters
  - Production - stable version
  - Test - version.next
- Clones can alternate between production and test binaries

<table>
<thead>
<tr>
<th>LPAR 2: z/VM 5.2 on a z990</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAINT: SYSTEM CONFIG (CF1), USER DIRECT(2CC)</td>
</tr>
<tr>
<td>TCPMAINT: SYSTEM DTCPARMS on 198</td>
</tr>
<tr>
<td>TCPIP: &lt;system_ID&gt; TCPIP on TCMAINT 198</td>
</tr>
<tr>
<td>AUTOLOG1: PROFILE EXEC on 191</td>
</tr>
<tr>
<td>DTCVSW1 and DTCVSW2: VSWITCH controllers</td>
</tr>
<tr>
<td>LINXMAINT: files on 192</td>
</tr>
<tr>
<td>SLES9: master image (IPL 100), controller (IPL 200)</td>
</tr>
<tr>
<td>SLES9X: master image (IPL 100), controller (IPL 200)</td>
</tr>
<tr>
<td>LINUX0n: virtual server on 100-102 (IPL 100)</td>
</tr>
<tr>
<td>WAS, DB2, MQS: IPL 100, 30x: production, 40x: test</td>
</tr>
</tbody>
</table>

For details see:
sessions 9216-7
Thurs@3:00 & 4:30
Room 321
Overview: Redbook, whitebook, red, red, white

- Redbook published From LPAR to Virtual Servers in Two Days, SG24-6695-00, 6/05
- First draft: 3/05
- Project started: 11/04

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

- Redbook published The Virtualization Cookbook for SLES9, SG24-6695-01, 4/06
- First draft of redbook: The Virtualization Cookbook for RHEL4, SG24-7272-00, 7/06

Overview: how to get it (let me count the ways :))

- Redbook version (SLES9 SP3, VM 5.2)
  - z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9
  - Associated files:

- "Whitebook" version (about same as redbook)
  - z/VM and Linux on IBM System z: The Virtualization Cookbook
  - Associated files
    - http://linuxvm.org/present/misc/virt-cookbook-1.tgz

- Redbook draft (RHEL4 U3, VM 5.2)
  - z/VM and Linux on IBM System z: The Virtualization Cookbook for Red Hat Enterprise Linux 4
  - Associated files

- Whitebook-2 draft
  - Associated files
    - http://linuxvm.org/present/misc/virt-cookbook-2.tgz
Overview: Cloning process block diagram

Before first IPL:
- WebSphereclone
  - usr/
  - var/
  - opt/
  - ... dev/
  - etc/
  - IBM/
  - init.d/
  - wasprofiles/
  - rc5.d/ WAS 200
  - WebSphere/
  - S99wasprofile

After first IPL:
- WebSphereclone
  - usr/
  - var/
  - opt/
  - ... dev/
  - etc/
  - IBM/
  - init.d/
  - wasprofiles/
  - wasprofile rc5.d/ WAS 200
  - WebSphere/
  - logs/ AppSrv01/ ...

Overview: User IDs, disks and mount points

<table>
<thead>
<tr>
<th>Software</th>
<th>z/VM User ID of &quot;master&quot;</th>
<th>Important minidisks (production and test)</th>
<th>Mount point on clones</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>SLES9 or SLES9X</td>
<td>100-102: master image</td>
<td>/sles9xmaster</td>
</tr>
<tr>
<td>WebSphere</td>
<td>WAS</td>
<td>300 and 400</td>
<td>/opt/IBM/Websphere/</td>
</tr>
<tr>
<td>DB2</td>
<td>DB2</td>
<td>301 and 401</td>
<td>/opt/IBM/db2/</td>
</tr>
<tr>
<td>MQ Series</td>
<td>MQS</td>
<td>302 and 402</td>
<td>/opt/mqm/</td>
</tr>
</tbody>
</table>
Overview: Cloned system block diagram

Roles
- Many possibilities
  - 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)
- z/VM expertise
- Linux expertise
- Middleware expertise
- AD expertise
Setting up: z/VM

- Define 3 user IDs for masters

  USER WAS    WAS    512M 1G G
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3038 VMA78B MR LNX4VM LNX4VM LNX4VM
  MDISK 102 3390 3039 0300 VMA78B MR LNX4VM LNX4VM LNX4VM
  MINIOPT NOMDC
  MDISK 300 3390 0001 3338 VMA78C MR LNX4VM LNX4VM LNX4VM
  MDISK 400 3390 0001 3338 VMA78D MR LNX4VM LNX4VM LNX4VM
  USER DB2    DB2    512M 1G G
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3038 VMA78E MR LNX4VM LNX4VM LNX4VM
  MDISK 102 3390 3039 0300 VMA78E MR LNX4VM LNX4VM LNX4VM
  MINIOPT NOMDC
  MDISK 301 3390 0001 1669 VMA78F MR LNX4VM LNX4VM LNX4VM
  MDISK 401 3390 1670 1669 VMA78F MR LNX4VM LNX4VM LNX4VM
  USER MQS    MQS    512M 1G G
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3038 VMA790 MR LNX4VM LNX4VM LNX4VM
  MDISK 102 3390 3039 0300 VMA790 MR LNX4VM LNX4VM LNX4VM
  MINIOPT NOMDC
  MDISK 302 3390 0001 1669 VMA791 MR LNX4VM LNX4VM LNX4VM
  MDISK 402 3390 1670 1669 VMA791 MR LNX4VM LNX4VM LNX4VM

  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

  USER LINUX05 LNX4VM 512M 1G EG
  INCLUDE LNXDFLT
  OPTION APPLMON
  MDISK 100 3390 0001 3038 VMA788 MR LNX4VM LNX4VM LNX4VM
  MDISK 102 3390 3039 0300 VMA788 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

  USER LINUX06 LNX4VM 512M 1G G
  ...
  USER LINUX07 LNX4VM 512M 1G G
  ...

- Define user IDs for clones

- Bring changes online
  - DISKMAP
  - DIRECTXA
Setting up: Linux master image

- Set new clones to have Java, compat libraries and empty mount points
- Modify the master image with `chroot`:
  - Add RPMs
    ```
    # chroot /sles9master/
    # yast -i IBMJava2-JRE IBMJava2-SDK compat compat-32bit
    ...
    
    # Create mount points
    # mkdir -p /opt/IBM/WebSphere /opt/IBM/db2 /opt/mqm
    # exit
    
    # Could also be done *after* cloning
    - To keep all clones as lean as possible
- Clone a new Linux
  ```
  ```
  # clone.sh WAS
  ```

WebSphere cloning - installing the master

- Install WebSphere Network Deployment V6.0.2 on master image
  ```
  
  SSH into new Linux running on WAS user ID
  ```
  ```
  # Activate minidisk 300, format, partition and make into file system:
  # chccwdev -e 0.0.0300
  # dasdfmt -b 4096 -y -f /dev/dasdd
  # fdasd -a /dev/dasdd
  ```
  ```
  # Make into file system with label "was-prod"
  # mke2fs -j -L was-prod /dev/dasdd1
  ```
  ```
  # Modify `/etc/zipl.conf` - add two new minidisks - run zipl
  ```
  [ipl]
  ```
  ```
  target = /boot/zipl
  image = /boot/image
  ramdisk = /boot/initrd,0x1000000
  parameters = "root=/dev/dasda1 dasd=0-102,300,400 selinux=0 TERM=dumb elevator=cfq"
  ```
  ```
  # zipl
  ```
  ```
  # Modify `/etc/fstab` - mount one minidisk (production)
  ```
  /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 ext3 acl,user_xattr 0 0
  devpts /dev/pts devpts mode=0620,gid=5 0 0
  ```
  ```
  ...
Installing the master WebSphere (cont'd)

- Reboot - new production disk should be mounted over /opt/IBM/WebSphere/
  ```
  # reboot
  # exit
  ... (new SSH session a minute later)
  # mount | grep opt
  /dev/dasdd1 on /opt/IBM/WebSphere type ext3 (rw,acl,user_xattr)
  ```
- Start VNC server for a graphical environment
  ```
  # vncserver
  ```
- Install WebSphere V6 but don't create a profile (see book for details)
- Install RefreshPack 2 to get to V6.0.2
- Modify new wasprofile.properties file with r/w profiles directory
  ```
  # cd /opt/IBM/WebSphere/AppServer/properties
  # vi wasprofile.properties
  ...
  WS_CMT_LOG_HOME=/opt/IBM/wasprofiles/logs
  ...
  WS_PROFILE_REGISTRY=/opt/IBM/wasprofiles/properties
  ...
  ```
- Shutdown and logoff the master WebSphere system!
  - Don't write to file system that is R/O to clones
  - Save resources - especially memory

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 0.0.0.300
  ```
- Modify /etc/zipl.conf to access disks read-only and run zipl
  ```
  ...
  [zipl]
  target = /boot/zipl
  image = /boot/image
  ramdisk = /boot/initrd,0x100000
  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,usr_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,usr_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
  ```bash
  mount /opt/IBM/WebSphere
  mount | grep opt
  /dev/dasd1 on /opt/IBM/WebSphere type ext2 (ro,acl,user_xattr)
  ```
- Reboot to test changes
  ```bash
  reboot
  ```
- Copy the wasprofile.properties file to the r/w /opt/IBM/wasprofiles/ directory
  ```bash
  cd /opt/IBM/wasprofiles
  mkdir properties
  cd properties
  cp /opt/IBM/WebSphere/AppServer/properties/wasprofile.properties .
  ```
- Add the WebSphere bin/ directory to the PATH
  ```bash
  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
  ```bash
  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
  ```bash
  wasprofile
  INSTCONFSUCCESS: Success: The profile now exists
  ```
  ```bash
  ls /opt/IBM/wasprofiles/AppSrv01
  bin/    etc/    installableApps/  installedConnectors/  properties/  wstemp/
  config/  firststeps/  installedApps/  logs/  temp/
  ```
- Start the server
  ```bash
  startServer.sh server1
  ... 
  ADMU3000I: Server server1 open for e-business; process id is 1340
  ```
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 \n  -profilePath /opt/IBM/wasprofiles/AppSrv01 \n  -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/default \n  -nodeName ${host}Node01 \n  -cellName ${host}Node01Cell \n  -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
               ro,acl,user_xattr  0 0' 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
  
  ```bash
  ...
  /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 cloning**

- Set up the DB2 master from the controller
  - Clone a vanilla server to the DB2 master user ID
    ```bash
    # clone.sh db2
    ...
    ```
- SSH into the master
  - Configure minidisks 301 and 401 read/write
    - Enable - `chccwdev`
    - Format - `dasdfmt`
    - Partition - `fdasd`
    - Make a file system - `mke2fs -j -L <label>`
    - Labels are `db2-prod` and `db2-test`
  - Configure `/etc/zipl.conf` and `/etc/fstab` to mount db2-prod (301) disk over `/opt/IBM/db2`
  - Reboot to test changes
    ```bash
    # mount | grep opt
    /dev/dasdd1 on /opt/IBM/db2 type ext3 (ro,acl,user_xattr)
    ```
  - Start the VNC server
  - Install DB2 from a graphical environment (details in the book)
    - Don't create a DB2 instance
- Shut down the DB2 master

**Clone a DB2 server manually**

- Clone a vanilla server to the target clone user ID
  ```bash
  # clone.sh linux06
  ...
  ```
- SSH into new clone
  - Enable disk `db2-prod` read-only
    - Modify `/etc/zipl.conf`, `/etc/zipl.conf`, run zipl, reboot to test
  - Create DB2 users and groups
    - For the instance `db2inst1`
      ```bash
      # groupadd db2grp1
      # mkdir /tmp/noskel
      # useradd -g db2grp1 -m -k /tmp/noskel db2inst1
      # passwd db2inst1
      ```
    - For the DB2 fenced user `db2fenc1`
    - For the DB2 administration user `db2admn1`
  - Create an instance
    ```bash
    # touch /home/db2inst1/.profile
    # /opt/IBM/db2/V8.1/instance/db2icrt -u db2fenc1 db2inst1
    DBI1070I Program db2icrt completed successfully.
    ```
  - Create an administrative user
    ```bash
    # touch /home/db2admn1/.profile
    # /opt/IBM/db2/V8.1/instance/dascrt db2admn1
    ```
Test the DB2 server

- Change user to db2inst1, create a sample database and query it
  
  ```bash
  # su - db2inst1
  db2inst1> db2sampl
 db2inst1> db2
  ...
  db2 => connect to sample
  Database server = DB2/LINUX64 8.2.0
  SQL authorization ID = DB2INST1
  Local database alias = SAMPLE
  ...
  db2 => select * from staff where dept = 20
  ID     NAME      DEPT   JOB   YEARS    SALARY    COMM
  ------ --------- ------ ----- ------ --------- ---------
  10 Sanders   20 Mgr   7  18357.50 -
  20 Pernal    20 Sales  8  18171.25  612.45
  80 James     20 Clerk -  13504.60  128.20
  190 Sneider  20 Clerk  8  14252.75  126.50
  ...
  ```

Automating cloning of DB2

- Create a script in master image’s services directory
  ```bash
  #!/bin/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
  groupadd db2fgrp1 # create a DB2 fenced user and group
  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
  /opt/IBM/db2/V8.1/instance/dascrt -u db2admn1
  echo "Return code from db2icrt = $?" | tee -a /tmp/db2instance.log
  echo "Return code from dascrt = $?" | tee -a /tmp/db2instance.log
  /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
  fi
  su - db2inst1 -c "db2start"
  echo "Return code from db2start = $?" | tee -a /tmp/db2instance.log
  ```
Automating cloning of DB2 (cont'd)

- Add a function to the clone.sh script

```bash
# function clone_db2()
# Arg 1: Mount point of the newly cloned server
# Clone a DB2 Server
#
{
  echo "Cloning DB2 ...
  echo "Modifying zipl.conf and running zipl ..."
  cd $1/etc
  cp zipl.conf zipl.conf.orig
  sed -i -e 's:dasd=100-102:dasd=100-102,301(ro),401(ro):g' zipl.conf
  chroot $1 zipl
  echo "Modifying fstab ...
  cp fstab fstab.orig
  sed -i -e '4a LABEL=db2-prod /opt/IBM/db2 ext2' fstab
  chroot $1/etc/inittab
  echo "Making symlink to /etc/init.d/db2instance ..."
  cd $1/etc/init.d/rc5.d
  ln -s ../db2instance S98db2instance
  cd
}
```

Test DB2 automated cloning

- Clone a vanilla server to the target clone user ID

```
  # clone.sh --db2 linux06
  ...
  WARNING!!: this will copy 100 and 102 disks to LINUX02 100 and 102
  New host name will be: lat136.pbm.ihost.com
  New TCP/IP address will be: 129.40.178.136
  Other network data is retrieved from LINUX02 PARMFILE on 191 disk
  A DB2 system will be cloned
  Are you sure you want to overwrite these disks (y/n): y
  ...
  SSH into new clone
  ...
  View the log file
  ...
  Creating a DB2 instance ...
  Return code from db2icrt = 0
  Return code from dascrt = 0
  Return code from db2autostart = 0
  Removed symbolic link /etc/init.d/rc5.d/S98db2instance
  Return code from db2start = 0
  Test DB2
  Test a reboot
```
MQ Series cloning

- Set up the DB2 master from the controller
  - Clone a vanilla server to the MQ Series master user ID
    # clone.sh mqs
    ...
- SSH into the master
  - Configure minidisks 302 and 402 read/write
    - Enable - `chccwdev`
    - Format - `dasdfmt`
    - Partition - `fdasd`
    - Make a file system - `mke2fs -j -L <label>`
      - Labels are `mqs-prod` and `mqs-test`
  - Configure `/etc/zipl.conf` and `/etc/fstab` to mount mqs-prod (302) disk over `/opt/mqm`
  - Reboot to test changes
    # mount | grep opt
    /dev/dasdd1 on /opt/mqm type ext3 (ro,acl,user_xattr)

- Install MQ Series from RPMs
  # rpm -ivh MQSeriesRuntime-6.0.0-0.s390.rpm MQSeriesSDK-6.0.0-0.s390.rpm
  MQSeriesServer-6.0.0-0.s390.rpm MQSeriesClient-6.0.0-0.s390.rpm
  MQSeriesSamples-6.0.0-0.s390.rpm MQSeriesJava-6.0.0-0.s390.rpm
  ...
- Shut down the MQ Series master

Clone a MQ Series server manually

- Clone a vanilla server to the target clone user ID
  # clone.sh linux07
  ...
- SSH into new clone
  - Enable disk `mqs-prod` read-only
  - Modify `/etc/zipl.conf`, run zipl, reboot to test
  - Create an MQ Series user and group
    # groupadd mqm
    # useradd -g mqm -d /var/mqm mqm

- Create a `.bash_profile` in the mqm home
  # cd /var/mqm
  # vi .bash_profile
  export PATH=/opt/mqm/bin

- Test by creating an MQ Series queue manager
  # su - mqm
  $ crtmqm -q test.queue.manager
Automating cloning of MQ Series (cont'd)

Create a script in master image's services directory

```bash
#!/bin/bash

echo "Creating an MQ Series user ..." | tee /tmp/mqmuser.log
groupadd mqm
echo "Return code from groupadd = $?" | tee -a /tmp/mqmuser.log
useradd -g mqm -d /var/mqm mqm
echo "Return code from useradd = $?" | tee -a /tmp/mqmuser.log
chown -R mqm.mqm /var/mqm
echo "Return code from chown = $?" | tee -a /tmp/mqmuser.log
if [ -h $symlink ]; then # symlink found - remove it so script runs once
  rm $symlink
  echo "Removed symbolic link $symlink" | tee -a /tmp/mqmuser.log
fi
```

Add a function to the clone.sh script

```bash
function clone_mqs()
#
# Clone a MQ Series Server
#
{
  echo "Cloning MQ Series ..."
  echo "Modifying zipl.conf and running zipl ..."
  cd $1/etc
  cp zipl.conf zipl.conf.orig
  sed -i -e 's:dasd=100-102:dasd=100-102,302(ro),402(ro):g' zipl.conf
  chroot $1 zipl
  echo "Modifying fstab ..."
  cp fstab fstab.orig
  sed -i -e 's/LABEL=mqs-prod /opt/mqm ext2 ro,acl,user_xattr 0 0' fstab
  echo "Making symlink to /etc/init.d/mqmuser ..."
  cd $1/etc/init.d/rcS.d
  ln -s ../mqmuser S97mqmuser
  cd
}
```
Test MQ Series automated cloning

- Clone a vanilla server to the target clone user ID
  
  ```bash
  # clone.sh --mqs linux07
  ...
  An MQ Series system will be cloned
  Are you sure you want to overwrite these disks (y/n): y
  ...
  Cloning MQ Series ...
  ...
  ```

- SSH into new clone
  - View the log file
    ```bash
    # cat /tmp/mqmuser.log
    Creating an MQ Series user ...
    Return code from groupadd = 0
    Return code from useradd = 0
    Return code from chown = 0
    Removed symbolic link /etc/init.d/rc5.d/S97mqmuser
    ```
  - Test MQ Series
  - Test a reboot

---

Live Demo!

Remember:
If it's not working, just pretend it is
Resources

- The Linux for zSeries and S/390 portal
  ➤ http://linuxvm.org/
- The linux-390 list server
  ➤ http://www2.marist.edu/htbin/wlindex?linux-390
- Linux for zSeries and S/390 developerWorks®
- SUSE LINUX Enterprise Server 9 evaluation
  ➤ http://www.novell.com/products/linuxenterpriseserver/eval.html
- Red Hat Enterprise Linux 4 evaluation
  ➤ http://www.redhat.com/rhel/details/eval/
- z/VM publications
- z/VM performance tips
  ➤ http://www.vm.ibm.com/perf/tips/

Questions - ???

- Mike's email: mikemac@us.ibm.com
- Associated files: http://linuxvm.org/present/misc/virt-cookbook-2.tgz
- This session: 1189/2954/9310