The Virtualization Cookbook: Day 2 - Linux

Michael MacIsaac - mikemac@us.ibm.com
Jin Xiong - jinxiong@us.ibm.com
Thursday, March 9th, 4:30 PM
Session 9217

IBM Systems Group

Abstract

There is a new redbook with a similar title: *z/VM and Linux on zSeries: From LPAR to Virtual Servers in Two Days*. As the title suggests a goal of the redbook is to allow you to install and configure z/VM, install and configure Linux and be cloning Linux in 2 working days.

On the second day, you install Linux twice onto the same virtual machine. Think of it as a dual boot PC. The first install is called the golden image which will be cloned from. The second install is called the controller which will do the cloning, but also other tasks. Both of these images are configured with the option of using scripts for speed, or manually to better understand the configuration. All of the data on a PC Linux NFS server which is needed to 'bootstrap' the first two installations is copied over the zSeries controller so the PC can be retired. Then cloning is addressed and the supplied clone script is described. After 'day 2,' you should be ready to clone the golden image and configure the clones into Web, LDAP, file and print, development or other servers. Finally backup and restore issues are addressed.

© 2006 IBM Corporation

Who are we?, who are you?

- Mike MacIsaac, mikemac@us.ibm.com
- Jin Xiong, jinxiong@us.ibm.com
- Who are you?
  - Have you tried the steps in this book?
  - Are you thinking about using this book?
  - Have you never heard about this book?
  - ...
  - Any other categories of attendees?
  - How many were not in the first hour?

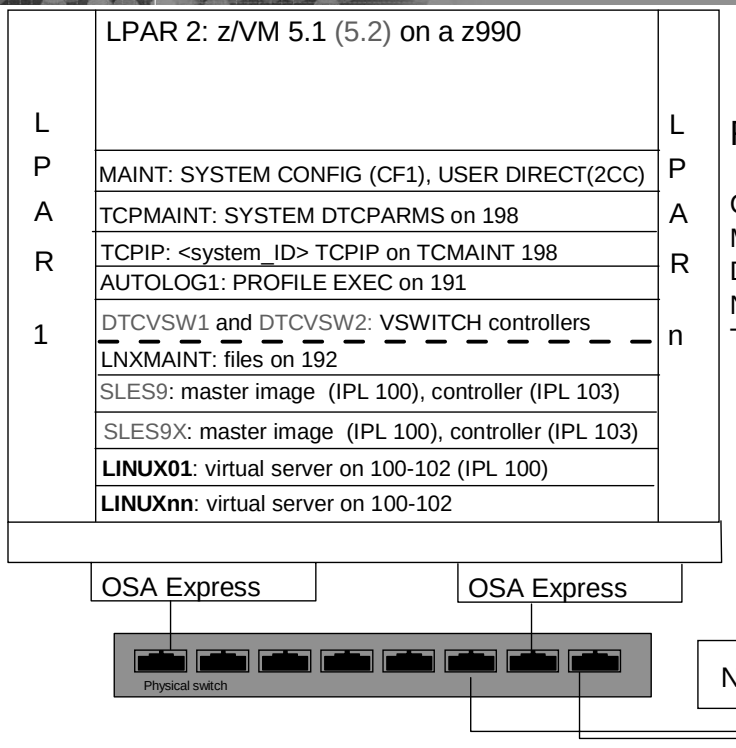
Outline for the two presentations

- Planning
- Configuring a Windows desktop
- Installing and configuring z/VM
- Configuring an NFS server
- **Install and configure Linux - Jin Xiong**
- **Configure NFS on controller - Jin Xiong**
- **Configure Linux for cloning - Jin Xiong**
- **Basic Linux virtual servers - Jin Xiong**
- **A virtual communications server**
- **A virtual communications controller server**
- Monitoring z/VM and Linux
- **Backup and restore**
- Appendix - Relabel z/VM system volumes

Outline

- ~~Planning~~
- ~~Configuring a Windows desktop~~
- ~~Installing and configuring z/VM~~
- ~~Configuring an NFS server~~
- **Install and configure Linux**
- Configure NFS on controller
- Configure Linux for cloning
- Basic Linux virtual servers
- A virtual communications server
- A virtual communications controller server
- ~~Monitoring z/VM and Linux~~
- Backup and restore
- ~~Appendix - Relabel z/VM system volumes~~

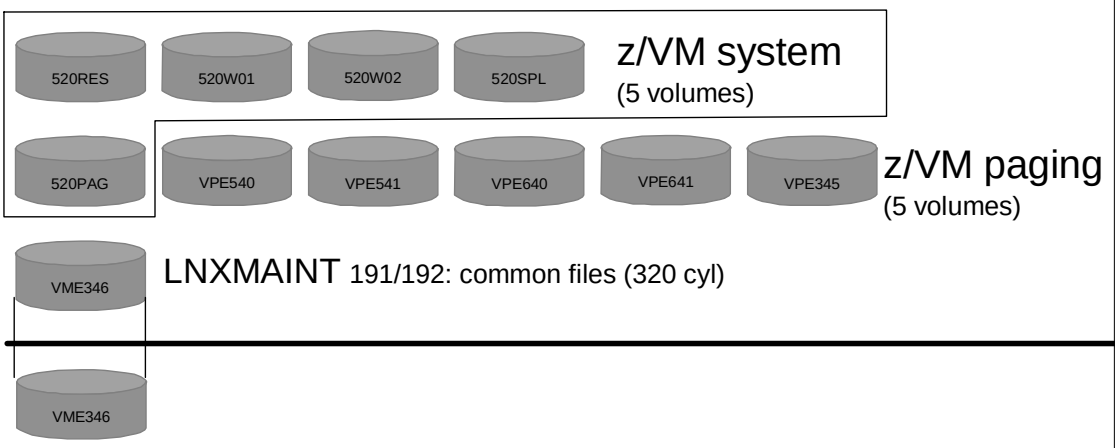
Review: Block Diagram



Resources:

CPU: 2 IFLs, shared
Memory: 3GB/1GB
Disk: 32 3390-3 DASD
Network: 16 OSA-E addresses
TCP/IP 13 TCP/IP addresses

Review - End of 1 day - DASD view



LINUX volumes
(this hour)

Install and configure Linux - define user ID SLES9X

```
PROFILE LNXDFLT
  IPL CMS
  MACHINE ESA 4
  CPU 00 BASE
  CPU 01
  NICDEF 600 TYPE QDIO LAN SYSTEM VSW1
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK LNXMAINT 192 191 RR
  LINK TCPMAINT 592 592 RR
...
USER SLES9X xxxxxxxx 256M 1G BG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 3038 VMA781 MR MDPW MDPW MDPW
 MDISK 102 3390 3039 0300 VMA781 MR MDPW MDPW MDPW
 MINIOPT NOMDC
 MDISK 103 3390 0001 3338 VMA782 MR MDPW MDPW MDPW
 MDISK 104 3390 0321 3338 VMA783 MR MDPW MDPW MDPW
 MDISK 105 3390 0001 3338 VMA784 MR MDPW MDPW MDPW
 MDISK 106 3390 0001 3338 VMA785 MR MDPW MDPW MDPW
 MDISK 107 3390 0001 3338 VMA786 MR MDPW MDPW MDPW
 MDISK 108 3390 0001 3338 VMA787 MR MDPW MDPW MDPW
```

VSWITCH "adapter"

master image

controller

/backup/

/nfs/ logical volume

Install and configure Linux (cont'd)

- Prepare bootstrap files
  - FTP from SLES9 install tree on Linux PC to LNXMAINT 192:
    - Kernel SLES9X KERNEL
    - Initial RAMdisk SLES9X INITRD
  - Install the master image on 100-102
    - Trickier install panels are documented well
  - Configure the master image
    - Apply service if necessary - Y.O.U. is documented
    - Copy associate files from NFS server via "scp -r"
    - Script **config-golden** can be used to automate configuration, or
    - All steps can be done manually
      - Add additonal RPMs
      - Turn off unneeded services
      - Configure rsyncd to allow /etc/ to be copied by controller
      - Configure sitar

Install and configure Linux (cont'd)

    - All steps can be done manually (cont'd)
      - Set the software clock from the controller (**/usr/sbin/ntpd -q**)
      - Turn off the hz_timer
      - Harden the system with Bastille Linux
  - Reboot system
- Install the controller - same as master except
  - Minidisk 100, /dev/dasda, is /sles9master (Don't format!!)
  - Mindisk 104, /dev/dasde, becomes a file system over /backup
  - Mindisks 105-108 become a logical volume mounted over /nfs
- Configure the controller
  - Apply service, if necessary
  - Copy files from NFS server
  - Automate configuration with **config-controller**, or do manually
    - Remove some RPMs
    - Add some RPMs, including cpint by Neale Ferguson

Install and configure Linux (cont'd)

- Manual configuration (cont'd)
  - Install cmsfs package by Rick Troth
  - Turn off unneeded services
  - Turn on the NFS server
  - Turn on the NTP server
    - Get accurate time from two Stratrum-2 servers on the Internet
  - Turn off hz_timer
  - Configure SSH keys for key-based authentication to clones
  - Harden the controller with Bastille Linux
  - Reboot
- You now have a dual-boot Linux ID
  - Controller normally runs
  - Golden image can be brought up (or can be **chroot**'ed into)

Outline

- ~~Planning~~
- ~~Configuring a Windows desktop~~
- ~~Installing and configuring z/VM~~
- ~~Configuring an NFS server~~
- ~~Install and configure Linux~~
- **Configure NFS on controller**
- Configure Linux for cloning
- Basic Linux virtual servers
- A virtual communications server
- A virtual communications controller server
- ~~Monitoring z/VM and Linux~~
- Backup and restore

Configure NFS on controller

- Copy files from PC NFS server"

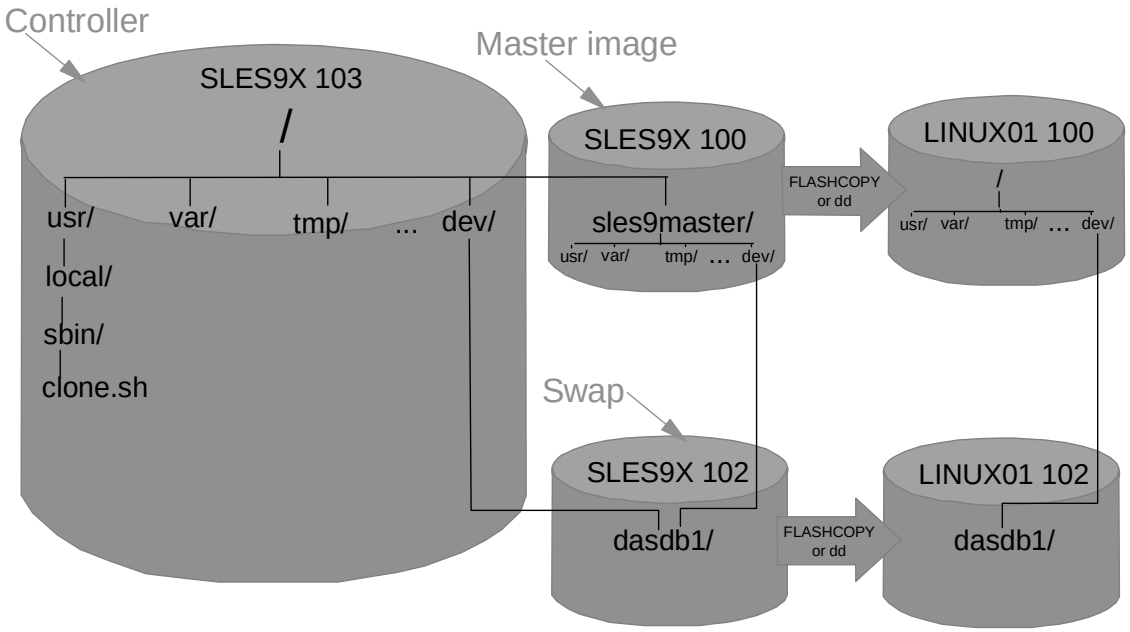
```
# scp -rp pc.nfs.server.ip:/nfs/sles9x/sles9sp3root /nfs
# scp -rp pc.nfs.server.ip:/nfs/virt-cookbook /nfs
```

- Configure NFS server
  - Same steps as on PC server
- Change YaST install tree location
  - Change source on the golden image
    - # chroot /sles9master
    - yast => Software => Change Source of Installation
    - # exit
    - Somewhat of a paradox, but useful on clones
  - Change source on controller
  - Change source in CMS parameter file
- Retire PC NFS server (if desired)

Outline

- ~~Planning~~
- ~~Configuring a Windows desktop~~
- ~~Installing and configuring z/VM~~
- ~~Configuring an NFS server~~
- ~~Install and configure Linux~~
- ~~Configure NFS on controller~~
- **Configure Linux for cloning**
- Basic Linux virtual servers
- A virtual communications server
- A virtual communications controller server
- ~~Monitoring z/VM and Linux~~
- Backup and restore
- ~~Appendix - Relabel z/VM system volumes~~

Cloning - block diagram

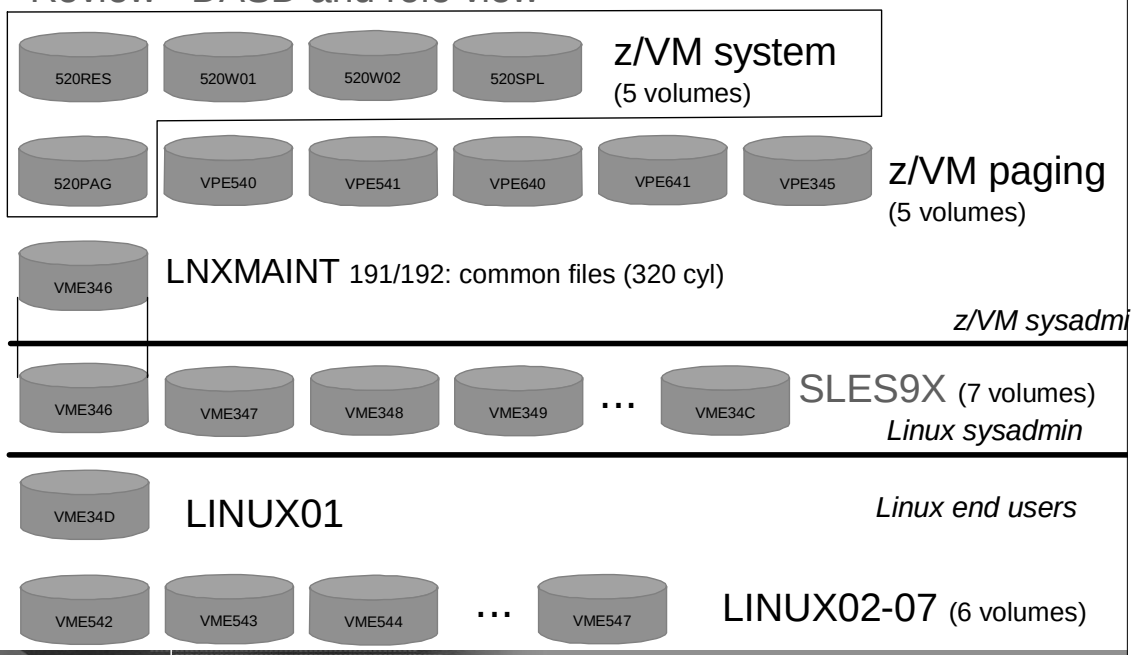


Cloning details

```
# main() # the first three lines are some important global variables
master_mnt_pt="/sles9master"                # set directory of master root file system
cloned_mnt_pt="/mnt/sles9cloned"            # set directory of temporary mount point
backup_dir="/backup/linux"                  # set directory of Linux backups

process_arguments $@                        # process arguments passed by user
check_target_id                             # be sure user ID exists and is logged off
get_parmfile_info                           # get info from source and target parm files
ask_are_you_sure                            # confirm that disks will be overwritten
copy_system                                 # copy 100 and 102 disks to target ID
modify_cloned_image $clone_id 100 1100      # modify newly copied system
cp_cmd XAUTOLOG $clone_id                   # bring new clone to life
make_backup_dir                             # make a backup directory
echo "Successfully cloned $master_mnt_pt to $clone_id"
echo "You should be able to ping $target_IP within one minute"
exit 0
```

Review - DASD and role view



Outline

- ~~Planning~~
- ~~Configuring a Windows desktop~~
- ~~Installing and configuring z/VM~~
- ~~Configuring an NFS server~~
- ~~Install and configure Linux~~
- ~~Configure NFS on controller~~
- ~~Configure Linux for cloning~~
- **Basic Linux virtual servers**
- A virtual communications server
- A virtual communications controller server
- ~~Monitoring z/VM and Linux~~
- Backup and restore
- ~~Appendix - Relabel z/VM system volumes~~

Basic Linux virtual servers

- Basic virtual server types
  - A virtual Web server
  - A virtual LDAP server
  - A virtual file and print server
  - A virtual application development server
- Clone a virtual server, with exception of LDAP server
- Install scripts to install/setup virtual server
- Procedures on how to bring them to life
- Also: Sharing executables via DCSS/XIP2

A virtual Web server

- Clone a server and install Apache2 RPMs
  - use the script or do it manually
- Testing Apache
  - Start Apache: rcapache2 start
  - Point Web browser to server address
- Configure HTTPS in Apache
  - Create a server certificate using openssl
    - self-signed
    - generate request for CA-signed
  - Configure Apache for HTTPS
    - runs on a different port, another Web server
    - create a virtual host
- Populate your Web site
  - Web pages located in /srv/www/htdocs/ by default

A virtual LDAP server

- OpenLDAP, pretty standard with most distros
- Centralized login authentication and user and group ID resolution
- You must have DNS for LDAP to work successfully
- Install Linux image manually
  - During the 2nd half of the YaST install, setup LDAP in the Service Configuration Panel
- After install, retrieve scripts and run config-golden
- Add a new user using YaST and login using new credentials
- Setting another virtual server to use LDAP server for login authentication
  - Add 2 RPMs: pam_ldap and nss_ldap
  - Use YaST to modify LDAP client configuration
  - Modify the OpenLDAP client configuration file
  - Test that it's working by logging in using a user defined on the LDAP server

A virtual file and print server

- Samba
  - Allows Windows clients to map Linux file systems as shared drives
  - Acts as middle-man between Windows clients and a Linux print server
- Clone a virtual server and install necessary RPMs
- Tweak Bastille configuration to allow NFS to run
- Configure the Samba configuration file to indicate which directories to share and how
- Add a Samba user
- Start Samba: rcnmb start, rcsmb start
- Test the changes
  - Map network drive on Windows client to the Samba share
- Print server configuration: *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864.

A virtual application development server

- "Pure" development environment
  - Most distributions come with a basic set of development tools
    - C/C++ development environment
    - Java development environment
    - Scripting languages
  - Web dev platform LAMP; Linux, Apache, MySQL, and Python/Perl
  - Basic C/C++ and Java development techniques
- Integrated development environment (IDE) - 1st edition only
  - The Eclipse application development platform
    - Very popular, is the basis for many other apps
    - Can be used for virtually any type of development, supports many programming languages with plug-ins
  - Installing Eclipse for Java and C/C++ development
    - Eclipse is not officially supported on Linux for zSeries
    - Build from source works on 31-bit platform only

Using DCSS/XIP2

- Reduce memory requirements of Linux servers
  - One copy of shared data loaded in memory per VM
  - DCSS is z/VM Discontiguous Saved Segment
  - XIP2 is Linux file system that allows a DCSS to be mounted
- Creating a DCSS/XIP2 shared file system:
  - Determine the size of the DCSS
  - Determine start and end addresses of the DCSS
  - Prepare Linux ID for DCSS creation (only for first guest)
  - Create a file system image for the DCSS in Linux
  - Create a DCSS from the file system image in VM
  - Change Linux to use DCSS
  - Test DCSS using XIP2
  - Activate XIP2 file system at boot time
  - Modify other Linux IDs to use DCSS - and save memory!
- Section based on work by Carsten Otte and Carlos Ordonez

Outline

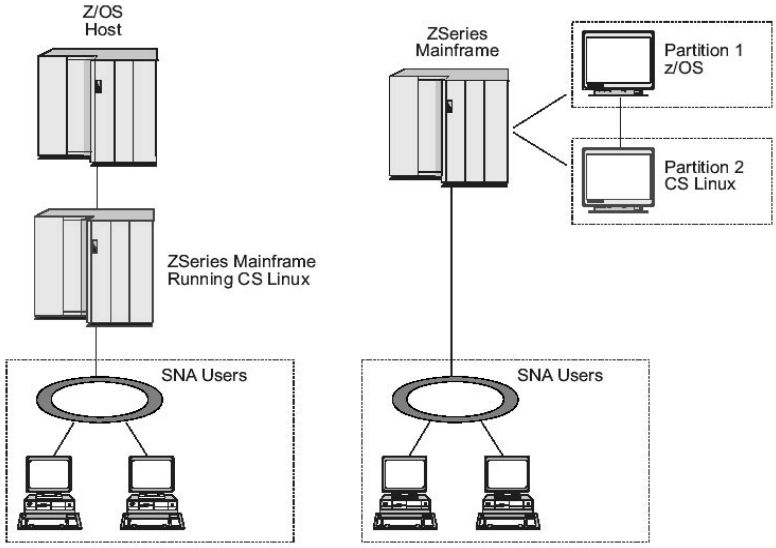
- ~~Planning~~
- ~~Configuring a Windows desktop~~
- ~~Installing and configuring z/VM~~
- ~~Configuring an NFS server~~
- ~~Install and configure Linux~~
- ~~Configure NFS on controller~~
- ~~Configure Linux for cloning~~
- ~~Basic Linux virtual servers~~
- **A virtual communications server**
- A virtual communications controller server
- ~~Monitoring z/VM and Linux~~
- Backup and restore
- ~~Appendix - Relabel z/VM system volumes~~

A virtual communications server

- Install IBM Communications Server for Linux (CS Linux)
  - SLES9 SP1 is required
  - Many kernel-souce type RPMs are required
  - Linux Streams module must be built
  - Then CS Linux can be built
- Configure CS Linux
  - Set some environment variables
  - Start SNA service
- CS Linux example - an EE link
  - Graphical config tool xsnaadmin is briefly described
- Documentation can be found starting at:
  - http://www-306.ibm.com/software/network/commserver/library/

CS Linux Examples

- In the first example, CS Linux is installed on a separate z800 system to offload the main z/OS system. An Enterprise Extender link using IP, or an LLC2 link, is used to connect the two systems.
- In the second example, CS Linux is installed on one or more VMs or LPARs in the main z/OS system. Although CS Linux and z/OS Communications Server are on the same mainframe, they are two separate SNA nodes, and so an Enterprise Extender link using HyperSockets IP or an LLC2 link is still required between them.



Outline

- ~~Planning~~
- ~~Configuring a Windows desktop~~
- ~~Installing and configuring z/VM~~
- ~~Configuring an NFS server~~
- ~~Install and configure Linux~~
- ~~Configure NFS on controller~~
- ~~Configure Linux for cloning~~
- ~~Basic Linux virtual servers~~
- ~~A virtual communications server~~
- **A virtual communications controller server**
- ~~Monitoring z/VM and Linux~~
- Backup and restore
- ~~Appendix - Relabel z/VM system volumes~~

A virtual communications controller server

- Install IBM Communication Controller for Linux (CC Linux)
  - SLES9 SP1 is required
  - Copy product code to virtual server
  - Install kernel source and other RPMs
  - Run CC Linux install program - **setuplinux390.bin**
- Configure OSA card(s)
  - Modify original NCP module
  - Build the CC Linux load module
  - Transfer the load module to Linux
  - Start the CC Linux engine - **cclengine** command
- Documentation can be found starting at:
  - http://www-306.ibm.com/software/network/ccl/
  - http://www.ibm.com/support/docview.wss?rs=2192&uid=swg27005786

Outline

- ~~Planning~~
- ~~Configuring a Windows desktop~~
- ~~Installing and configuring z/VM~~
- ~~Configuring an NFS server~~
- ~~Install and configure Linux~~
- ~~Configure NFS on controller~~
- ~~Configure Linux for cloning~~
- ~~Basic Linux virtual servers~~
- ~~A virtual communications server~~
- ~~A virtual communications controller server~~
- ~~Monitoring z/VM and Linux~~
- **Backup and restore**
- ~~Appendix - Relabel z/VM system volumes~~

Backup and restore

- Overview
  - Incremental backup of z/VM
  - Incremental backup of Linux
  - Disaster recovery
- Incremental backup of z/VM
  - There are 6 important configuration files:
    - SYSTEM CONFIG (on MAINT CF1)
    - USER DIRECT (on MAINT 2CC)
    - <system_ID> TCPIP, SYSTEM DTCPARMS (TCPMAINT 198)
    - TCPIP DATA (on TCPMAINT 592)
    - PROFILE EXEC (on AUTOLOG1 191)
  - If you have copies of these files, you can restore a z/VM system
  - Recall that the /backup file system was created on the controller
  - These files can be backed up nightly - here is one method ...
    - What is the problem with this assumption?

Backup and restore (cont'd)

- On controller: to backup Linuxes a script, **backup_linux**, is written:

```
# tail -7 /etc/cron.daily/backup_linux
backup_dir="/backup/linux"
cd $backup_dir
for i in LINUX*-on-*  # iterate through directories starting with LINUX*-on-*
do
  IP_addr=${i#LINUX*-on-}  # this chops the head off and grabs the IP address
  cd $backup_dir/$i        # change directory
  rsync -r --timeout=30 $IP_addr:/etc .    # use rsync to back up the /etc dir
done
```

- On clones, sitar is run nightly

```
#!/bin/bash
sitar --format=html --outfile=/etc/sitar.html
```

  - By writing output of sitar to /etc/, it is also backed up
- Disaster recovery
  - Follow good s/390 procedures (but for a different backup ...)
  - Consider that three volumes (VME346-8 in book) contain
    - LNXMAINT 192 (Common CMS files)
    - LINUX00 100 (Golden), 103 (Controller), 104 (/backup)
    - Along with the 5 vanilla z/VM volumes, a system could be rebuilt

Live Demo!

Resources

- Redbook *z/VM and Linux on zSeries: From LPAR to Virtual Servers in Two Days*
  - http://www.redbooks.ibm.com/abstracts/sg246695.html
- Files associated with the redbook
  - ftp://www.redbooks.ibm.com/redbooks/SG246695/
- *The* Linux for zSeries and S/390 portal
  - http://linuxvm.org/
- The linux-390 list server
  - http://www2.marist.edu/htbin/wlvindex?linux-390
- Linux for zSeries and S/390 developerWorks®
  - http://awlinux1.alphaworks.ibm.com/developerworks/linux390/index.shtml
- SUSE LINUX Enterprise Server 9 evaluation
  - http://www.novell.com/products/linuxenterpriseserver/eval.html
- z/VM publications
  - http://www.vm.ibm.com/pubs/
- z/VM performance tips
  - http://www.vm.ibm.com/perf/tips/

Questions - ???