

Extreme Cloud Administration Toolkit

xCAT on z/VM and zLinux

Version 0.24

Doc. Version	0.24
Date	08/20/10
Owner	Thang Pham
Author	Thang Pham (thang.pham@us.ibm.com) Wesley Yee (wesley.yee@us.ibm.com) Mike MacIsaac (mikemac@us.ibm.com)
File Name	xCAT2onzVM.odt

Document Abstract

This document provides an overview, an installation, and a quick start guide on basic z/VM and zLinux administration using xCAT.

Table of Contents

Document Abstract.....	2
Terminology.....	4
Support on z/VM and zLinux.....	5
Design Architecture.....	6
Prerequisite.....	7
Installation of xCAT.....	10
Installation of xCAT UI.....	13
SSL Configuration.....	13
Installation of zHCP.....	14
Getting Started.....	16
xCAT Commands.....	18
Installing Linux Using AutoYast or Kickstart.....	21
Cloning Virtual Servers.....	23
Setting Up Ganglia on xCAT.....	24
Ganglia Monitoring on xCAT.....	26
Statelite.....	27
Updating Linux.....	34
Limitations.....	35
Appendix A: Setting Up a Second Network	36
Appendix B: Customizing Autoyast and Kickstart.....	40

Terminology

This section outlines the terminology used within this document.

DirMaint	CMS application that helps manage an installation's VM directory.
Ganglia	<i>"Ganglia consists of two unique daemons (gmond and gmetad), a PHP-based web frontend and a few other small utility programs. Gmond is a multi-threaded daemon which runs on each cluster node you want to monitor. Gmetad is the daemon that monitors the other nodes by periodically polling them, parsing the collected XML, and saving all the numeric, volatile metrics to the round-robin databases."</i> - Ganglia Development Team
Life cycle	A collection of tasks that include: power on/off of a virtual server, and create/edit/delete of a virtual server.
SMAPI	The Systems Management APIs simplify the task of managing many virtual images running under a single z/VM image.
Virtual server	A server composed of virtualized resources. An operating system can be installed on a virtual server.
VMCP	Linux module that allows execution of CP commands.
CP	<i>"The Control Program (CP) is the operating system that underlies all of z/VM. It is responsible for virtualizing your z/Series machine's real hardware, and allowing many virtual machines to simultaneously share the hardware resource."</i> - IBM
xCAT	xCAT (Extreme Cloud Administration Tool) is a toolkit that provides support for the deployment and administration of large cloud environments.
zHCP	zHCP (System z Hardware control point) is a Linux virtual server that interfaces with SMAPI and CP and manages other virtual servers on z/VM.
AutoYaST	<i>"AutoYaST is a system for installing one or more SUSE Linux systems automatically and without user intervention. AutoYaST installations are performed using an autoyast profile with installation and configuration data."</i> -SUSE
Kickstart	<i>"Automated installation for Red Hat. It uses a file containing the answers to all the questions that would normally be asked during a typical Red Hat Linux installation."</i> -Red Hat

Support on z/VM and zLinux

This section provides a list of supported functionalities on xCAT for z/VM and zLinux.

1. Life cycle support for xCAT on z/VM, which includes:
 - a) Power on/off virtual server
 - b) Create virtual server
 - c) Edit virtual server configuration
 - d) Delete virtual server
2. Software and hardware inventory of a virtual server
3. Cloning a virtual server
4. Installation of zLinux on a virtual server using kickstart/autoyast
5. Monitoring of zLinux systems using Ganglia
6. Remote shell
7. Updating Linux
8. NFS read-only root filesystem (Statelite)

Design Architecture

This section provides information on the architecture of xCAT on z/VM and zLinux.

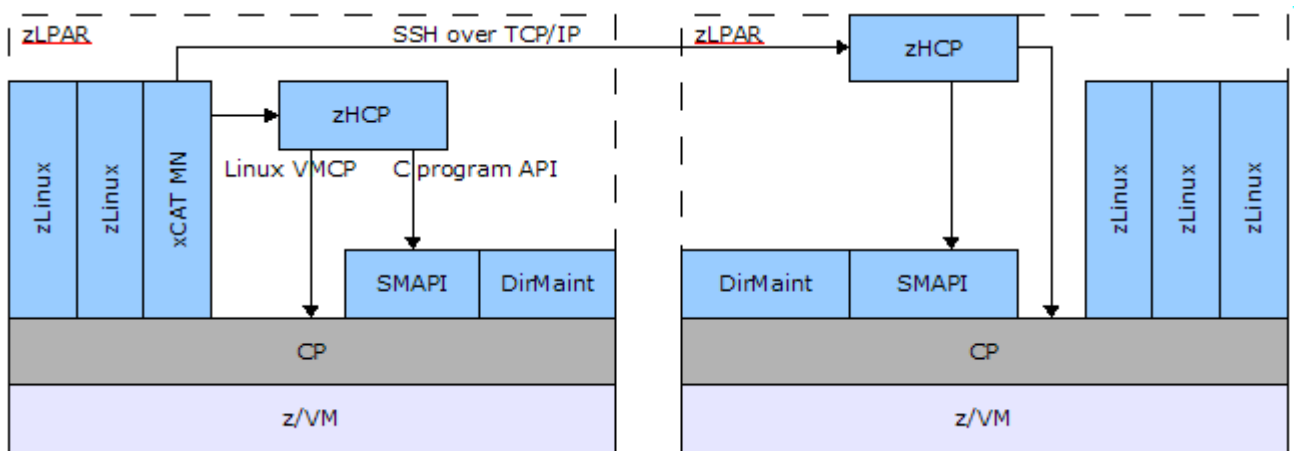


Figure 1. Shows the layout of xCAT on System z.

xCAT is used to manage virtual servers within multiple z/VM partitions. The xCAT management node (MN) can be installed on any zLinux. It communicates with a System z hardware control point (zHCP) over SSH, which is required on each z/VM partition in order to manage it. The zHCP is a zLinux, and its purpose is to interface with the SMAPI and CP layer. It utilizes a C socket interface to communicate with the SMAPI layer and VMCP Linux module to communicate with the CP layer.

xCAT mainly relies on Linux VMCP module to perform its actions. However, more complicated actions, such as creating/deleting/editing virtual servers, require SMAPI.

Prerequisite

This section details what is required before you setup xCAT on z/VM and zLinux.

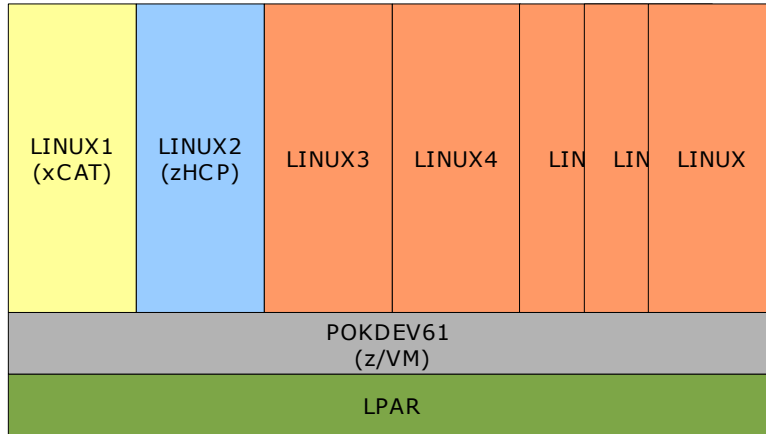


Figure 1. Sample environment

Before you can install xCAT, there are a couple of prerequisites. You need to have two virtual servers (one server for the xCAT MN and the other for the zHCP) running zLinux.

Both the xCAT MN and zHCP are linked to LNXMAINT, which contains files used by both CMS and Linux. LNXMAINT has the following directory entry:

```
USER LNXMAINT DRCT 64M 128M BEG
INCLUDE TCPCMSU
LINK TCPMAINT 0592 0592 RR
MDISK 0191 3390 1 20 EM6340 MR
MDISK 0192 3390 1 279 EM6341 MR
```

It is recommended that you have the following PROFILE EXEC on LNXMAINT 192 disk (which is linked to all virtual servers on the z/VM partition).

```
/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
'SWAPGEN 300 1048576' /* create a 512M VDISK disk swap space */
'SWAPGEN 301 2097152' /* create a 1G VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (dsc = 'DSC') then /* User is disconnected */
  'CP IPL 100'
else /* User is interactive -> prompt */
do
  say 'Do you want to IPL Linux from minidisk 100? y/n'
  parse upper pull answer .
  if (answer = 'Y') then 'CP IPL 100'
end /* else */
```

This statement in the PROFILE EXEC enables each virtual server to IPL 100 upon startup.

For more information on how to setup z/VM, refer to the *z/VM and Linux on IBM System z The Virtualization Cookbook for SLES 10 SP2 (SG24-7493)*.

xCAT Management Node

In our development environment, the xCAT MN has the following directory entry:

```
USER LINUX1 PWD 1G 2G G
INCLUDE LNXDFLT
COMMAND SET VSWITCH VSW2 GRANT LINUX1
MDISK 0100 3390 0001 10016 EMC21A MR
MDISK 0101 3390 0001 10016 EMC21C MR
MDISK 0102 3390 0001 10016 EMC28B MR
```

where the user profile, LNXDFLT, contains:

```
PROFILE LNXDFLT
CPU 00 BASE
CPU 01
IPL CMS
MACHINE ESA 4
CONSOLE 0009 3215 T
NICDEF 0800 TYPE QDIO LAN SYSTEM VSW2
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK MAINT 0402 0402 RR
LINK LNXMAINT 0192 0191 RR
LINK TCPMAINT 0592 0592 RR
```

It is recommended that you use LVM for the install directory (/install), so you are not constrained by disk size. In our development environment, we allocated 4GB to / and the rest into an LVM partition for /install. The xCAT MN is connected to NICDEF 0800 which uses VSW2, a layer 2 VSWITCH. If you plan to have your virtual servers managed by xCAT, each virtual server must be connected to this VSWITCH. xCAT requires a layer 2 network in order to use DHCP. It is recommended that you create a layer 2 VSWITCH, which will allow virtual servers to communicate across LPARs and CECs.

The xCAT MN can run on any Linux distribution, SLES or RHEL. It is recommended that you use SLES 10 SP3 or newer for the xCAT MN because it offers the rrdtool package, which is required for Ganglia. In our development environment, the xCAT MN was setup on SLES 11 SP1 with Server Base, Gnome, and X Windows packages installed.

System z Hardware Control Point

In our development environment, the zHCP has the following directory entry:

```
USER LINUX2 DRCT 512M 1G BCDG
COMMAND SET VSWITCH VSW2 GRANT LINUX2
CPU 00 BASE
CPU 01
IPL CMS
MACHINE ESA 4
```



```
OPTION LNKNOPAS
CONSOLE 0009 3215 T
NICDEF 0800 TYPE QDIO LAN SYSTEM VSW2
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK LNMAINT 0192 0191 RR
LINK TCPMAINT 0592 0592 RR
MDISK 0100 3390 1 10016 EMC278
```

It is recommended that you mount / onto MDISK 0100. You do not need 10016 cylinders allocated to the zHCP, but you do need enough for small Linux operating system. The zHCP is connected to NICDEF 0800 which uses VSW2, a layer 2 VSWITCH. The zHCP can only use one network (in our development environment, VSWITCH VSW2). This network must be specified in the directory entry and not in a profile. The zHCP has B, C, D, and G privileges. It needs class B privilege to use the FLASHCOPY command (if permitted), class C privilege to use the SEND command, and class D privilege to use the PURGE command.

The zHCP can run on any Linux distribution, SLES or RHEL. In our development environment, the zHCP was setup on SLES 10 SP3 with Server Base package installed.

Installation of xCAT

This section details how to install the xCAT management node. For more details, refer to the *xCAT 2 Top Doc* located at <http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/index.html>.

If you have Red Hat Enterprise Linux:

1. Logon as root using a Putty terminal
2. Disable SELinux

```
# echo 0 > /selinux/enforce
```

The command above will switch off enforcement temporarily, until you reboot the system.

To make it permanent, edit /etc/selinux/config and change SELINUX=enforcing to SELINUX=permissive.

3. Add the RHEL repository to yum

- a) Create a repository file

```
# touch /etc/yum.repos.d/rhel-dvd.repo
```

- b) Insert the following into the repository file rhel-dvd.repo

```
[rhel-dvd]
name=RHEL DVD
baseurl=ftp://xxx-ftp-path
enabled=1
gpgcheck=1
```

where xxx-ftp-path is the FTP path to the RHEL DVD. For example:

```
[rhel-dvd]
name=RHEL DVD
baseurl=ftp://10.1.100.2/rhel5.5/s390x/Server
enabled=1
gpgcheck=1
```

- c) Download the RPM-GPG-KEY-redhat-release from the FTP server (e.g. <ftp://10.1.100.2/rhel5.5/s390x/>) onto this node.

- d) Import the key

```
# rpm -import RPM-GPG-KEY-redhat-release
```

4. Make an xcat directory under /root

```
# mkdir /root/xcat
```

5. Download the latest xCAT tarballs, xcat-core-**xxx**.tar.bz2 and xcat-dep-**xxx**.tar.bz2 (where xxx is the release and version number) from <http://xcat.sourceforge.net/#download> onto /root/xcat

6. Extract the contents of each tarball

```
# cd /root/xcat
# tar jxf xcat-core-xxx.tar.bz2
# tar jxf xcat-dep-xxx.tar.bz2
```

7. Create a yum repositories for xCAT

If you have Red Hat Enterprise Linux 5:

```
# /root/xcat/xcat-dep/rh5/s390x/mklocalrepo.sh
```

```
# /root/xcat/xcat-core/mklocalrepo.sh
```

If you have Red Hat Enterprise Linux 6:

```
# /root/xcat/xcat-dep/rh6/s390x/mklocalrepo.sh
```

```
# /root/xcat/xcat-core/mklocalrepo.sh
```

8. Use `yum` to install xCAT

```
# yum clean metadata
# yum install xCAT
```

Ignore the warning messages (if any) about the keys and accept them.

If you have SUSE Linux Enterprise Server:

1. Logon as root using a Putty terminal
2. Install the DHCP server through `yast` (if not already)

```
# yast -i dhcp-server
```
3. Make an xcat directory under `/root`

```
# mkdir /root/xcat
```
4. Download the latest xCAT tarballs, `xcat-core-xxx.tar.bz2` and `xcat-dep-xxx.tar.bz2` (where xxx is the version number) from <http://xcat.sourceforge.net/#download> onto `/root/xcat`
5. Extract the contents of each tarball

```
# cd /root/xcat
# tar jxf xcat-core-xxx.tar.bz2
# tar jxf xcat-dep-xxx.tar.bz2
```

6. Add the xCAT repositories to `zypper`

If you have SUSE Linux Enterprise Server 10:

```
# zypper sa file:///root/xcat/xcat-dep/sles10/s390x xCAT-dep
# zypper sa file:///root/xcat/xcat-core xcat-core
```

If you have SUSE Linux Enterprise Server 11:

```
# zypper ar file:///root/xcat/xcat-dep/sles11/s390x xCAT-dep
# zypper ar file:///root/xcat/xcat-core xcat-core
```

Ignore the warning messages (if any) about the keys and accept them.

7. Use `zypper` to install xCAT

```
# zypper install xCAT
```

Continue with the following steps once you completed installing xCAT:

1. Add the xCAT commands to path

```
# source /etc/profile.d/xcat.sh
```
2. Check if the database is initialize

```
# tabdump site
```

The output should look similar to the following:

```
#key,value,comments,disable
"blademaxp","64",,
"domain","endicott.ibm.com",,
"fsptimeout","0",,
"installdir","/install",,
"ipmimaxp","64",,
"ipmiretries","3",,
"ipmitimeout","2",,
"consoleondemand","no",,
"master","10.1.100.1",,
"maxssh","8",,
"ppcmaxp","64",,
"ppcretry","3",,
```

```
"ppctimeout","0",,  
"rsh","/usr/bin/ssh",,  
"rcp","/usr/bin/scp",,  
"sharedtftp","0",,  
"SNsyncfiledir","/var/xcat/syncfiles",,  
"tftpdirdir","/tftpboot",,  
"xcatdport","3001",,  
"xcatiport","3002",,  
"xcatconfdir","/etc/xcat",,  
"timezone","US/Eastern",,  
"nameservers","10.1.100.1",,
```

3. Setup an FTP server on the xCAT MN to contain Linux distributions

- a) Download the desire Linux ISO into /install
- b) Go into /install directory

```
# cd /install
```

- c) Extract the ISO into the xCAT install tree /install

```
# copycds -n xxx -a s390x /install/yyy.iso
```

where xxx is the distribution name and yyy is the ISO name.

For example, if you have a SUSE Linux Enterprise Server 10 SP3 ISO:

```
# copycds -n sles10.3 -a s390x /install/SLES-10-SP3-DVD-s390x-DVD1.iso
```

or if you have a Red Hat Enterprise Linux 5.4 ISO:

```
# copycds -n rhel5.4 -a s390x /install/RHEL5.4-Server-20090819.0-s390x-DVD.iso
```

- d) Remove the ISO from /install since we do not need the ISO any longer, and it consumes disk space

```
# rm /install/SLES-10-SP3-DVD-s390x-DVD1.iso
```

Installation of xCAT UI

This section details the installation of the xCAT UI.

If you have Red Hat Enterprise Linux:

1. Use yum to install the following packages (accept the dependencies)
`# yum install php php-pear httpd`
2. Allow httpd to make network connections (if SELinux is enabled)
`# /usr/sbin/setsebool httpd_can_network_connect=1`
3. Install the xCAT-UI
`# yum install xCAT-UI`

If you have SUSE Linux Enterprise Server:

1. Use zypper to install the following packages (accept the dependencies)
`# zypper in php5-openssl apache2 apache2-mod_php5`
2. Install the xCAT-UI
`# zypper in xCAT-UI`

SSL Configuration

This section details the configuration of SSL on the xCAT server. SSL stands for Secure Socket Layer, which is a security protocol for communications over networks.

If you have Red Hat Enterprise Linux:

No actions required. SSL should be configured by default.

If you have SUSE Linux Enterprise Server:

You can find the following instructions from http://en.opensuse.org/Apache_Howto_SSL.

1. Apache should be set to start with SSL. Verify with the following command
`# a2enmod ssl`
"ssl" already present
2. Make sure that SSL is active
`# a2enflag SSL`
3. Create self signed keys
`# gensslcert`
4. Copy `/etc/apache2/vhosts.d/vhost-ssl.template` to `/etc/apache2/vhosts.d/vhost-ssl.conf`
`# cp /etc/apache2/vhosts.d/vhost-ssl.template /etc/apache2/vhosts.d/vhost-ssl.conf`
5. For the enabled modules, server flags, generated keys and vhosts to take effect, restart the apache service
`# service apache2 restart`
Syntax OK
Shutting down httpd2 (waiting for all children to terminate) done
Starting httpd2 (prefork)
6. Open a browser to the xCAT UI at `https://xxx/xcat`, where xxx is the host name of the xCAT MN. For example, <https://gpok1.endicott.ibm.com/xcat>. You will get a "Untrusted certificate" warning when you first try to access the URL. This is expected because of the use of a self-signed certificate.

Installation of zHCP

This section details the installation of the zHCP.

Before the zHCP can be installed on a virtual server running Linux, the following must be done (if not already).

1. Install and configure SMAPI and DirMaint for each z/VM partition. Refer to (step 1: Configure DirMaint through step 6: Customize EXTENT CONTROL)
http://publib.boulder.ibm.com/infocenter/director/v6r1x/index.jsp?topic=/director.install_6.1/fqm0_t_installing_z_map_agents.html

2. Grant the zHCP access to DirMaint.

a) Open a 3270 console, logon MAINT, and issue the following commands

```
==> DIRM FOR ALL AUTHFOR LINUX2 CMDL 140A CMDS ADGHOPS
```

```
==> DIRM FOR ALL AUTHFOR LINUX2 CMDL 150A CMDS ADGHOPS
```

b) Change VSMWORK1 AUTHLIST

```
==> SET FILEPOOL VMSYS
```

```
==> QUERY FILEPOOL CONNECT
```

```
==> ACCESS VMSYS:VSMWORK1.DATA A (FORCERW
```

```
==> ACCESS VMSYS:VSMWORK1. B (FORCERW
```

```
==> X VSMWORK1 AUTHLIST B
```

Append the z/VM userID you wish to have DIRMAINT access to the end of VSMWORK1 AUTHLIST

```
LINUX2          ALL          ALL
```

c) Restart SMAPI

```
==> FORCE VSMWORK1
```

```
==> XAUTOLOG VSMWORK1
```

3. Give the virtual server where you will install the zHCP B, C, D, and G privileges. The zHCP needs class B privilege to use the FLASHCOPY command (if available), class C privilege to use the SEND command, and class D privilege to use the PURGE command. In order for the zHCP to have these privileges, you must open a 3270 console, logon to MAINT after the user has been created, and issue:

```
==> DIRM FORUSER LINUX2 CLASS BCDG
```

4. Log off MAINT

```
==> LOGOFF
```

Install the zHCP

1. Logon to the xCAT MN as root using a Putty terminal
2. Go into the directory where you extracted the xcat-dep tarball, e.g. /root/xcat. Send the zHCP RPM (zhcp-1-1.s390x.rpm) located in /root/xcat/xcat-dep/<os>/s390x to the zHCP, where <os> is the operating system installed on the zHCP. For example,

```
# scp /root/xcat/xcat-dep/sles10/s390x/zhcp-1-1.s390x.rpm root@10.1.100.2:
```

3. Exit the Putty session to the xCAT MN
4. Logon to the zHCP Linux as root using a Putty terminal
5. Install gcc and gcc-c++ (if not already)

If you have Red Hat Enterprise Linux:

```
# yum install gcc gcc-c++
```

If you have SUSE Linux Enterprise Server:

```
# yast -i gcc gcc-c++
```

6. Install the RPM

```
# rpm -i /root/zhcp-1-1-s390x.rpm
```

7. Append the zHCP library path to /etc/ld.so.conf
echo /opt/zhcp/lib >> /etc/ld.so.conf
8. Configure dynamic linker run time bindings
ldconfig

Getting Started

This section details how to manage z/VM and zLinux using xCAT.

1. Logon the xCAT MN as root using a Putty terminal
2. Load Linux VMCP module on the xCAT MN (if not already)
3. Set up the `passwd` table. This table will contain the default password for new nodes installed through autoyast/kickstart and other methods.

```
# modprobe vmcp
# chtab key=system passwd.username=root passwd.password=xxx
```

where xxx is the root password.

4. Set up the `hosts` table (it will be used to setup `/etc/hosts`). You need to determine the regular expression that represents the nodes that xCAT will manage.

```
# chtab node=xxx hosts.ip="yyy" hosts.hostnames="zzz"
```

where xxx is the node range, yyy is the regular expression for the IP addresses, and zzz is the regular expression for the hostnames. Each time a new node is added to xCAT, you will need to run `makehosts`. You will need to setup the hosts table for each group you create.

In our development environment, we setup nodes belonging to `group=all` to have hostnames of `gpok1`, `node2`, etc. and IP addresses of `10.1.100.1`, `10.1.100.2`, etc. in `/etc/hosts` with the following:

```
# chtab node=all hosts.ip="gpok(\d+)|10.1.100.(\$1+0)" hosts.hostnames="|
(.*)|(\$1).endicott.ibm.com|"
```

5. Setup the `networks` table. You need to set the DHCP, DNS, and FTP server to the IP address of your xCAT MN.

In our development environment, we setup up the xCAT MN to manage the network `10.1.100.0`, which has a netmask of `255.255.255.0`, a gateway of `10.1.100.1`, and on an ethernet interface `eth1`. Our DHCP, DNS, and FTP servers are at `10.1.100.1`. This is the command that we used:

```
# chtab net=10.1.100.0 networks.mask=255.255.255.0 networks.mgtifname=eth1
networks.gateway=10.1.100.1 networks.dhcpserver=10.1.100.1
networks.tftpserver=10.1.100.1 networks.nameservers=10.1.100.1
```

6. Define the DHCP interfaces in the `site` table to limit which network the DHCP server will listen on. In our development environment, we setup `eth1` as the interface where we have the DHCP server listening on.

```
# chtab key=dhcpinterfaces site.value='all|eth1'
```

7. Edit the nameserver and master in the `site` table to point it to the xCAT MN. In our development environment, we setup our nameserver and master to be `10.1.100.1`.

```
# chtab key=nameservers site.value='10.1.100.1'
# chtab key=master site.value='10.1.100.1'
```

8. Configure the DHCP server
 - a. Add networks into the DHCP configuration

```
# makedhcp -n
```

- b. Restart DHCP

```
# service dhcpd restart
```

- c. Add the relevant networks to DHCP

```
# makedhcp -a
```

9. Configure the DNS server

- a. Setup DNS from entries in /etc/hosts

```
# makedns
```

- b. Edit /etc/resolv.conf to contain the appropriate domain and nameserver (if not already). In our development environment, we setup /etc/resolv.conf to contain:

```
domain endicott.ibm.com
nameserver 10.1.100.1
```

- c. Start DNS

```
# service named start
```

- d. Start DNS on boot

```
# chkconfig --level 345 named on
```

10. Start by adding the zHCP node into the database (Use the DNS hostname of that node when adding). In our development environment, our zHCP has a hostname of gpok2, a userID of LINUX2, and belonged to the group=all. This is the command that we used:

```
# mkdef -t node -o gpok2 userid=LINUX2 hcp=gpok2.endicott.ibm.com mgt=zvm
groups=all
```

11. Add more nodes (if any) that you want to manage into the database. For example, if you have a node with a hostname of gpok3 and userID of LINUX3 on the same z/VM partition (managed by the zHCP on gpok2), you would use the following command:

```
# mkdef -t node -o gpok3 userid=LINUX3 hcp=gpok2.endicott.ibm.com mgt=zvm
groups=all
```

The node IP address should follow the rule you specified in the hosts table (step 4).

12. Update /etc/hosts

```
# makehosts
```

13. Update DNS

```
# makedns
```

14. Setup the SSH keys for the node range that you want to manage

```
# xdsh xxx -K
```

where **xxx** is the node range. For example, if you were to setup the SSH keys for the nodes you added above in steps 10 and 11, you can use:

```
# xdsh all -K
```

The `xdsh` command will prompt you for a root password. It is the root password for the node or group you are trying to push the public SSH key to. It is recommended that you put nodes with the same root password into the same group. More importantly, the `xdsh` command will only work for nodes that are online.

15. Start using supported xCAT commands

xCAT Commands

This section lists the current xCAT commands supported on z/VM and zLinux.

`rpower` – Controls the power for a node or noderange.

The syntax is: `rpower <node> [on|off|stat|reset]`

```
# rpower gpok3 stat
```

`mkvm` – Creates a new virtual server with the same profile/resources as the specified node (cloning). Alternatively, creates a new virtual server based on a directory entry.

The syntax is: `mkvm <new node> /tmp/<directory entry text file>`

```
# mkvm gpok3 /tmp/dirEntry.txt
```

For cloning, the syntax is: `mkvm <target Linux> <source Linux> pool=<disk pool> pw=<multi password>`

```
# mkvm gpok4,gpok5 gpok3 pool=POOL1
```

`rmvm` – Removes a virtual server. The syntax is: `rmvm <node>`.

```
# rmvm gpok3
```

`lsvm` – List a virtual server's configuration. Options supported are:

- List the directory entry.

The syntax is: `lsvm <node>`

```
# lsvm gpok3
```

- List the defined network names available.

The syntax is: `lsvm <node> --getnetworknames`

```
# lsvm gpok3 --getnetworknames
```

- List the configuration for a given network.

The syntax is: `lsvm <node> --getnetwork [networkname]`

```
# lsvm gpok3 --getnetwork GLAN1
```

- List the disk pool names available.

The syntax is: `lsvm <node> --diskpoolnames`

```
# lsvm gpok3 --diskpoolnames
```

- List the configuration for a given disk pool.

The syntax is: `lsvm <node> --diskpool [pool name] [space (free or used)]`

```
# lsvm gpok3 --diskpool POOL1 free
```

`chvm` – Changes the virtual server's configuration. Options supported are:

- Adds a 3390 (ECKD) disk to a virtual server's directory entry.

The syntax is: `chvm <node> --add3390 [disk pool] [device address] [cylinders] [mode] [read password] [write password] [multi password]`

```
# chvm gpok3 --add3390 POOL1 0101 3338 MR
```

- Adds a 3390 (ECKD) disk that is defined in a virtual server's directory entry to that virtual server's active configuration.

The syntax is: `chvm <node> --add3390active [device address] [mode]`

```
# chvm gpok3 --add3390active 0101 MR
```

- Adds a 9336 (FBA) disk to a virtual server's directory entry.

- The syntax is: `chvm <node> --add9336 [disk pool] [virtual device] [block size] [mode] [blocks] [read password] [write password] [multi password]`
- ```
chvm gpok3 --add9336 POOL3 0101 4096 3338 MR
```
- Adds a network adapter to a virtual server's directory entry (case sensitive).  
The syntax is: `chvm <node> --addnic [address] [type] [device count]`  

```
chvm gpok3 --addnic 0600 QDIO 3
```
  - Adds a virtual processor to a virtual server's directory entry.  
The syntax is: `chvm <node> --addprocessor [address]`  

```
chvm gpok3 --addprocessor 01
```
  - Adds a virtual processor to a virtual server's active configuration (case sensitive).  
The syntax is: `chvm <node> --addprocessoractive [address] [type]`  

```
chvm gpok3 --addprocessoractive 01 ZAAP
```
  - Adds a v-disk to a virtual server's directory entry.  
The syntax is: `chvm <node> --addvdisk [userID] [device address] [size]`  

```
chvm gpok3 --addvdisk 0300 2097120
```
  - Connects a given network adapter to a GuestLAN.  
The syntax is: `chvm <node> --connectnic2guestlan [address] [lan] [owner]`  

```
chvm gpok3 --connectnic2guestlan 0600 GLAN1 LN1OWNR
```
  - Connects a given network adapter to a VSwitch.  
The syntax is: `chvm <node> --connectnic2vswitch [address] [vswitch]`  

```
chvm gpok3 --connectnic2vswitch 0600 VSW1
```
  - Copy a disk attached to a given virtual server.  
The syntax is: `chvm <node> --copydisk [target address] [source node] [source address]`  

```
chvm gpok3 --copydisk 0100 gpok2 0101
```
  - Adds a dedicated device to a virtual server's directory entry.  
The syntax is: `chvm <node> --dedicatedevice [virtual device] [real device] [mode]`  

```
chvm gpok3 --dedicatedevice 0101 637F RW
```
  - Deletes the IPL statement from the virtual server's directory entry.  
The syntax is: `chvm <node> --deleteipl`  

```
chvm gpok3 --deleteipl
```
  - Formats a disk attached to a given virtual server (only ECKD disks supported). The disk should not be linked to any other virtual server. This command is best used after `add3390()`.  
The syntax is: `chvm <node> --formatdisk [disk address] [multi password]`  

```
chvm gpok3 --formatdisk 0100 PWD
```
  - Disconnects a given network adapter.  
The syntax is: `chvm <node> --disconnectnic [address]`  

```
chvm gpok3 --disconnectnic 0600
```
  - Grant VSwitch access for given virtual server.  
The syntax is: `chvm <node> --grantvswitch [VSwitch]`  

```
chvm gpok3 --grantvswitch VSW1
```
  - Removes a minidisk from a virtual server's directory entry.  
The syntax is: `chvm <node> --removedisk [virtual device]`  

```
chvm gpok3 --removedisk 0101
```
  - Removes a network adapter from a virtual server's directory entry.  
The syntax is: `chvm <node> --removenic [address]`  

```
chvm gpok3 --removenic 0700
```
  - Removes a processor from an active virtual server's configuration.

The syntax is: `chvm <node> --removeprocessor [address]`

```
chvm gpok3 --removeprocessor 01
```

- Replaces a virtual server's directory entry.

The syntax is: `chvm <node> --replacevs [directory entry]`

```
chvm gpok3 --replacevs /tmp/dirEntry.txt
```

- Sets the IPL statement for a given virtual server.

The syntax is: `chvm <node> --setipl [ipl target] [load parms] [parms]`

```
chvm gpok3 --setipl CMS
```

- Sets the password for a given virtual server.

The syntax is: `chvm <node> --setpassword [password]`

```
chvm gpok3 --setpassword PSSWD
```

`rscan` – Collects the node information from one or more hardware control points.

The syntax is `rscan <hcp>`.

```
rscan gpok2
```

`rinv` – Remote hardware and software inventory. The syntax is: `rinv <node> <all|config>`.

```
rinv gpok3 all
```

## Installing Linux Using AutoYast or Kickstart

This section provides details on the installation of Linux using autoyast or kickstart.

Before you begin, make sure the following is done.

- The FTP server must be setup during the xCAT MN installation, and the FTP root directory (/install) must contain the appropriate Linux distribution.
- If you are managing an IP address range starting at 0 (e.g. 10.1.100.0), be sure that the netmask is set correctly (e.g. 255.255.255.0) on the xCAT MN or else the node you are trying to provision cannot find the repository.

There are two ways to install Linux onto a z/VM virtual server, depending on which Linux distribution you want. One is through autoyast, which is used to install SUSE Linux Enterprise Server (SLES) releases. The other is through kickstart, which is used to install Red Hat Enterprise Linux (RHEL) releases.

In the following example, we will provision a new node (**gpok3**) with a userID (**LINUX3**) that is managed by our zHCP (**gpok2**). You will need to substitute the node name, userID, and zHCP name with appropriate values.

1. Logon the xCAT MN as root using a Putty terminal
2. Create the node definition

```
mkdef -t node -o gpok3 userid=LINUX3 hcp=gpok2.endicott.ibm.com mgt=zvm
groups=all
```
3. Update /etc/hosts

```
makehosts
```
4. Update DNS

```
makedns
```
5. Create the new virtual server using the desired directory entry. For our example, we used the following:

```
USER LINUX3 PWD 512M 1G G
INCLUDE LNXDFLT
COMMAND SET VSWITCH VSW2 GRANT LINUX3
```

To create the virtual server, copy the directory entry above into a text file (dirEntry.txt) and issue the following command (the full file path must be given):

```
mkvm gpok3 /tmp/dirEntry.txt
```

The directory entry text file should not contain any extra new lines (/n). A MAC address will be assigned to the userID upon creation.

6. Copy the default autoyast/kickstart template available in xCAT (if not already). Customize this template to how you see fit. For more information on how to customize the template, see Appendix B.

If you want to install a SUSE Linux Enterprise Server:

```
mkdir -p /install/custom/install/sles
cp /opt/xcat/share/xcat/install/sles/compute.sles10.s390x.tmpl
/install/custom/install/sles
```

There are two templates available for SLES, one for SLES 10 (compute.sles10.s390x.tmpl) and the other for SLES 11 (compute.sles11.s390x.tmpl). It is recommended that you copy both templates into /install/custom/install/sles.

If you want to install a Red Hat Enterprise Linux:

```
mkdir -p /install/custom/install/rh
cp /opt/xcat/share/xcat/install/rh/compute.rhel5.s390x.tmpl
/install/custom/install/rh
```

7. Add disks to the new node (the default autoyast/kickstart template available in xCAT requires 1 3390-MOD9 disks attached at 0100).

```
chvm gpok3 --add3390 POOL1 0100 10016 MR
```

Be sure that each disk in the pool is attached to SYSTEM.

8. Set up the `noderes` and `nodetype` tables. You need to determine the OS and profile (autoyast/kickstart template) for the node. Here, we have `nodetype.os=sles10.3`. You can find available OS and profiles by issuing:

```
tabdump osimage
```

If you want to install a [SUSE Linux Enterprise Server](#):

```
chtab node=gpok3 noderes.netboot=zvm nodetype.os=sles10.3 nodetype.arch=s390x
nodetype.profile=compute
```

If you want to install a [Red Hat Enterprise Linux](#):

```
chtab node=gpok3 noderes.netboot=zvm nodetype.os=rhe15.4 nodetype.arch=s390x
nodetype.profile=compute
```

9. Verify the definition

```
lsdef gpok3
```

It should look similar to this:

```
Object name: gpok3
arch=s390x
groups=all
hcp=gpok2.endicott.ibm.com
hostnames=gpok3.endicott.ibm.com
ip=10.1.100.3
mac=02:00:01:FF:FF:FF
mgt=zvm
netboot=zvm
os=sles10.3
postbootscripts=otherpkgs
postscripts=syslog,remoteshell,syncfiles
profile=compute
userid=LINUX3
```

10. Add the new node to DHCP

```
makedhcp -a
```

11. Power on the node

```
rpower gpok3 on
```

12. Prepare the new node for installation

```
nodeset gpok3 install
```

13. Boot the new node from reader

```
rnetboot gpok3 ipl=00C
```

14. In Gnome or KDE, open the VNC viewer to see the installation progress. It might take a couple of minutes before you can connect.

```
vncviewer gpok3:1
```

The default VNC `password` is 12345678. If you have trouble connecting to the vncviewer, open a 3270 console to the node, try steps 12 and 13 again, and look at the progress on the console.

15. (Only for SLES 10 SP2 or older) Once the `first` phase of installation is complete, restart the virtual server to complete the `final` phase of installation

```
rpower gpok3 reset
```

16. The default password for the node can be found in the `passwd` table. See *Getting Started* section step 2.

## Cloning Virtual Servers

This section shows how to clone a virtual server running Linux.

In the following example, we will clone the virtual server that we created (**gpok3**) in the previous section *Installing Linux Using Autoyast or Kickstart*. The new virtual servers will have node names (**gpok4** and **gpok5**) and userIDs (**LINUX4** and **LINUX5**) respectively, and managed by the same zHCP (**gpok2**). You will need to substitute the node names, userIDs, and zHCP name with appropriate values.

1. Logon the xCAT MN as root using a Putty terminal (if not already)
2. The source node must be online. If it is not online, bring it online.  
`# rpower gpok3 on`
3. Setup the SSH keys for the source node to be cloned (if not already)  
`# xdsh gpok3 -K`
4. Create the table definitions for new nodes (gpok4 and gpok5)  
`# mkdef -t node -o gpok4 userid=LINUX4 hcp=gpok2.endicott.ibm.com mgt=zvm groups=all`  
`# mkdef -t node -o gpok5 userid=LINUX5 hcp=gpok2.endicott.ibm.com mgt=zvm groups=all`
5. Update /etc/hosts  
`# makehosts`
6. Update DNS  
`# makedns`
7. Add the new node to DHCP  
`# makedhcp -a`
8. In order to clone a virtual server running Linux, the partition must be mounted by path. This is done by default for the node (**gpok3**) that we created in the previous section and in general, for nodes provision by xCAT using the default templates.

For SUSE Linux Enterprise Server:

The root directory under /etc/fstab, which contains information on the system partitions and disks, should be similar to this:

```
/dev/disk/by-path/ccw-0.0.0100-part1 / ext3 acl,user_xattr 1 1
```

The parameters under /etc/zipl.conf, which specifies which disks to bring online when the system is IPLed, should be similar to this:

```
parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb"
```

If you happen to edit zipl.conf, you must run zipl after you made the changes so that changes are written to the boot record.

9. Clone virtual server(s) running Linux:  
`# mkvm gpok4, gpok5 gpok3 pool=POOL1`

This will create two virtual servers (gpok4 and gpok5) identical to gpok3. It will use disks in disk pool POOL1.

If FLASHCOPY is not enabled on your z/VM system, then this will take some time to complete depending on the number of nodes you want to clone.

10. Check the boot status of the nodes by pinging them:  
`# pping gpok4, gpok5`

If the node returns a ping, then it is fully booted and you can start using it.

## Setting Up Ganglia on xCAT

This section details how to the set up Ganglia on zLinux.

If you have Red Hat Enterprise Linux:

Not yet supported because Red Hat is missing the rrdtool package.

If you have SUSE Linux Enterprise Server:

1. Logon the xCAT MN as root using a Putty terminal (if not already)
2. Go into the directory where you extracted the xcat-dep tarball, e.g. /root/xcat. Locate the Ganglia RPMs under /root/xcat/xcat-dep/sles10/s390x. You will need to install the following RPMs in the later steps.

```
libconfuse-2.6-1.s390x.rpm
libganglia-3.1.1-1.s390x.rpm
ganglia-gmetad-3.1.1-1.s390x.rpm
ganglia-gmond-3.1.1-1.s390x.rpm
ganglia-web-3.1.1-1.s390x.rpm
```

3. Set up ganglia on the xCAT MN
  - a) Install PHP and apache packages (if not already). Use yast to install the following packages

```
yast -i libapr1 pkgconfig php5-pear php5-gd apache2 apache2-mod_php5
```

- b) Install the Ganglia library RPMs

```
rpm -i libconfuse-2.6-1.s390x.rpm
rpm -i libganglia-3.1.1-1.s390x.rpm
```

- c) Install gmetad (This monitors other nodes by periodically polling them)

```
yast -i rrdtool
rpm -i ganglia-gmetad-3.1.1-1.s390x.rpm
```

- d) Install gmond

```
rpm -i ganglia-gmond-3.1.1-1.s390x.rpm
```

- e) Install the ganglia web RPM

```
rpm -i ganglia-web-3.1.1-1.s390x.rpm
```

- f) Configure the apache server

- i. Specify the NameVirtualHost in /etc/apache2/listen.conf

```
NameVirtualHost 10.1.100.1:80
```

- ii. Copy /etc/apache2/vhosts.d/vhost.template

```
cd /etc/apache2/vhosts.d
cp vhost.template ganglia.conf
```

- iii. Edit /etc/apache2/conf.d/ganglia.conf

```
<VirtualHost 10.1.100.1:80>
 ServerName gpok253.endicott.ibm.com
 DocumentRoot /srv/www/htdocs/ganglia
 <Directory "/srv/www/htdocs/ganglia">
 Order allow,deny
 Allow from all
 </Directory>
</VirtualHost>
```

- g) Restart the apache server

```
/etc/init.d/apache2 restart
```

- h) Restart gmond and gmetad



- ```
# service gmond restart
# service gmetad restart
```
4. Create the directory /install/ganglia on the xCAT MN

```
# mkdir -p /install/ganglia
```
 5. Copy the following packages from /root/xcat/xcat-dep/sles10/s390x into /install/ganglia

```
libganglia-3.1.1-1.s390x.rpm
libconfuse-devel-2.6-1.s390x.rpm
ganglia-gmond-3.1.1-1.s390x.rpm
```

Ganglia Monitoring on xCAT

This section details how to use Ganglia on zLinux.

1. Logon the xCAT MN as root using a Putty terminal (if not already)
2. Transfer ganglia RPMs required to run gmond over to nodes you want to monitor

```
# xdcp <node> /install/ganglia/ganglia-gmond-3.1.1-1.s390x.rpm
# xdcp <node> /install/ganglia/libconfuse-2.6-1.s390x.rpm
# xdcp <node> /install/ganglia/libganglia-3.1.1-1.s390x.rpm
```

The command transfers the files into /root directory on the target nodes.

3. Install the RPMs

```
# xdsh <node> rpm -i libconfuse-2.6-1.s390x.rpm
# xdsh <node> rpm -i libganglia-3.1.1-1.s390x.rpm
# xdsh <node> rpm -i ganglia-gmond-3.1.1-1.s390x.rpm
```

Make sure the target node has *libapr1* package installed.

4. Ensure the nodetype of all nodes you wish to monitor have the type of 'osi'. This can be done by editing the nodetype table.

```
# tabedit nodetype
```

5. Add gangliamon to the monitoring table

```
# monadd gangliamon
```

6. Configure the node

```
# moncfg gangliamon -r
```

This runs the ganglia configuration script on all the nodes.

7. If you want to start gangliamon:

```
# monstart gangliamon -r
```

The command will start the gmond daemon on all the nodes. The `-r` flag is required to ensure the gmond daemon is started on each node. You may also specify a particular node to start:

```
# monstart gangliamon gpok3 -r
```

If you want to stop gangliamon:

```
# monstop gangliamon -r
```

Statelite

This section details how to configure an NFS read-only root filesystem. For more details, refer to the *xCAT Statelite Cookbook* located at <http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/index.html>.

If you have SUSE Linux Enterprise Server:

1. Logon the xCAT MN as root using a Putty terminal (if not already)
2. Edit /etc/exports to export the /install directory. It should contain these two directories:

```
/install      *(rw,no_root_squash,sync,no_subtree_check)
/lite/state   *(rw,no_root_squash,sync,no_subtree_check)
```

3. Restart the NFS server

```
# service nfsserver restart
```

4. Check that the NFS server is running

```
# rpcinfo -p
```

Make sure nfs is listed, e.g.

```
100003      2    tcp    2049  nfs
100003      2    tcp    2049  nfs
100003      3    tcp    2049  nfs
100003      4    tcp    2049  nfs
100003      2    udp    2049  nfs
100003      3    udp    2049  nfs
100003      4    udp    2049  nfs
```

5. Edit the litefile table. This table specifies which files should be kept persistent across reboots. By default, all files are kept under tmpfs, unless a persistent, ro, or bind option is specified. Refer to the litefile table description for more details.

```
# tabedit litefile
```

Copy the following defaults into the litefile table.

```
#image, file, options, comments, disable
"ALL", "/etc/inittab",,,,
"ALL", "/etc/lvm/.cache",,,,
"ALL", "/etc/mtab",,,,
"ALL", "/etc/ntp.conf",,,,
"ALL", "/etc/resolv.conf",,,,
"ALL", "/etc/ssh/", "persistent",,
"ALL", "/etc/sysconfig",,,,
"ALL", "/etc/syslog-ng",,,,
"ALL", "/tmp",,,,
"ALL", "/var/tmp",,,,
"ALL", "/var/run",,,,
"ALL", "/etc/yp.conf",,,,
"ALL", "/var/lib",,,,
"ALL", "/var/empty",,,,
"ALL", "/var/spool",,,,
"ALL", "/var/lock",,,,
"ALL", "/var/log",,,,
"ALL", "/var/cache",,,,
"ALL", "/etc/fstab",,,,
"ALL", "/var/adm",,,,
```

```
"ALL", "/root/.viminfo",,,
"ALL", "/root/.bash_history",,,
"ALL", "/opt/xcat/",,,
"ALL", "/xcatpost/",,,
```

6. Edit the litetree table. This table controls where the files specified in the litefile table come from.

```
# tabedit litetree
```

Copy the following into the litetree table. You will need to determine the Linux distribution you want. In our example, SLES11 SP1 is used.

```
#priority,image,directory,comments,disable
"1.0",,"10.1.100.1:/install/netboot/sles11.1/s390x/compute",,,
```

7. Edit the statalite table. This table controls where the permanent files are kept.

```
# tabedit statelite
```

Copy the following into the statelite table. You will need to determine the statelite node range and the IP address of the xCAT MN. In our example, the node range is **all** and the IP address is **10.1.100.1**.

```
#node,image,statemnt,comments,disable
"all",,"10.1.100.1:/lite/state",,,
```

8. Create the persistent directory

```
# mkdir -p /lite/state
```

9. Ensure policies are set up correctly. When a node boots up, it queries the xCAT database to get the lite-files and the lite-tree. In order for this to work, the command must be set in the policy table to allow nodes to request it. (This should already be done automatically when xCAT was installed)

```
# chtab priority=4.7 policy.commands=litefile policy.rule=allow
# chtab priority=4.8 policy.commands=litetree policy.rule=allow
```

10. Download and copy the packages from the Linux distro media into /install (if not already)

```
# copycds -n xxx -a s390x /install/yyy.iso
```

where xxx is the distribution name and yyy is the ISO name.

For example, if you have a SLES 11 SP1 ISO:

```
# copycds -n sles11.1 -a s390x /install/SLES-11-SP1-DVD-s390x-GMC3-DVD1.iso
```

11. Create a list of packages that should be installed onto the statelite image. You should start with the base packages in the compute template and if desired, add more packages by editing the .pkglist.

```
# mkdir -p /install/custom/netboot/sles
# cp /opt/xcat/share/xcat/netboot/sles/compute.sles11.s390x.pkglist
/install/custom/netboot/sles
```

12. Create the statelite image

```
# genimage -i eth1 -n qeth -o sles11.1 -p compute -m statelite
```

This command creates a SLES11 SP1 image with an eth1 interface, qeth network driver, and uses the compute profile. The interface used must match the xCAT MN interface that DHCP listens on. The genimage command creates an image under /install/netboot/sles11.1/s390x/compute/rootimg. It also creates a ramdisk and kernel that is used to boot the statelite node.

13. Modify the statelite image by creating symbolic links will all the files listed under the litetree table

```
# liteimg -o sles11.1 -a s390x -p compute
```

14. Create the statelite node definition.

For our example, we will create a new node (gpok6) with a userID (LINUX6) that is managed by our zHCP (gpok2). You will need to substitute the node names, userIDs, and zHCP name with appropriate values.

```
# mkdef -t node -o gpok6 userid=LINUX6 hcp=gpok2.endicott.ibm.com mgt=zvm
```

```
groups=all
```

15. Update /etc/hosts

```
# makehosts
```

16. Update DNS

```
# makedns
```

17. Create the new virtual server using the desired directory entry. For our example, we used the following:

```
USER LINUX6 PWD 512M 1G G
COMMAND SET VSWITCH VSW2 GRANT LINUX6
CPU 00 BASE
CPU 01
IPL CMS
MACHINE ESA 4
CONSOLE 0009 3215 T
NICDEF 0800 TYPE QDIO LAN SYSTEM VSW2
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
```

To create the virtual server, copy the directory entry above into a text file (dirEntry.txt) and issue the following command (the full file path must be given):

```
# mkvm gpok6 /tmp/dirEntry.txt
```

The new virtual server should be attached to the same vswitch as the one used by the hardware control point (in our case, **VSW2**) and have the same network adapter address (in our case, **0800**) for the interface given in step 12 (in our case, **eth1**).

18. Clone this node as many times as you want to achieve the number of statelite nodes you desire. Refer to *Cloning Virtual Servers* section above. In order to clone, the source statelite node must be online and have SSH keys setup.

19. Add the new node to DHCP

```
# makedhcp -a
```

20. Set up the `noderes` and `nodetype` tables. The values of `nodetype.os` and `nodetype.profile` were determined in step 11, where the statelite image was created.

```
# chtab node=xxx noderes.netboot=zvm nodetype.os=yyy nodetype.arch=s390x
nodetype.profile=zzz
```

where xxx is the node name, yyy is the operating system, and zzz is the profile name.

In our example, we used the following:

```
# chtab node=gpok6 noderes.netboot=zvm nodetype.os=sles11.1 nodetype.arch=s390x
nodetype.profile=compute
```

21. Prepare the node(s) to boot from the statelite image

```
# nodeset xxx statelite
```

where xxx is the node name.

22. Boot the statelite node(s). During this process, the symbolic links are made to files listed under the litefile table.

```
# rnetboot xxx ipl=00c
```

where xxx is the node name.

Caution: Do not try to boot more than 20 nodes at one time. The xCAT MN will be bogged down as all

the nodes are trying to access the NFS server at once. Try booting 20 or less at a time and waiting till those nodes are pingable before booting the next batch.

23. Check the boot status of the nodes by pinging them:

```
# ping xxx
```

where xxx is the node name. If the node returns a ping, then it is fully booted and you can start using it.

If you have Red Hat Enterprise Linux:

1. Logon the xCAT MN as root using a Putty terminal (if not already)

2. Edit /etc/exports to export the /install directory. It should look similar to this:

```
/install      *(rw,no_root_squash,sync,no_subtree_check)
/lite/state   *(rw,no_root_squash,sync,no_subtree_check)
```

3. Restart the NFS server

```
# service nfs restart
```

4. Edit the litfile table. This table specifies which files should be kept persistent across reboots. By default, all files are kept under tmpfs, unless a persistent, ro, or bind option is specified. Refer to the litfile table description for more details.

```
# tabedit litfile
```

Copy the following defaults into the litfile table.

```
#image, file, options, comments, disable
"ALL", "/etc/adjtime",,,
"ALL", "/etc/fstab",,,
"ALL", "/etc/inittab",,,
"ALL", "/etc/lvm/.cache",,,
"ALL", "/etc/mtab",,,
"ALL", "/etc/ntp.conf",,,
"ALL", "/etc/ntp.conf.predhclient",,,
"ALL", "/etc/resolv.conf",,,
"ALL", "/etc/resolv.conf.predhclient",,,
"ALL", "/etc/ssh/", "persistent",,
"ALL", "/tmp/",,,
"ALL", "/var/account/",,,
"ALL", "/var/arpwatch",,,
"ALL", "/var/cache/alchemy",,,
"ALL", "/var/cache/foomatic",,,
"ALL", "/var/cache/logwatch",,,
"ALL", "/var/cache/man",,,
"ALL", "/var/cache/mod_ssl",,,
"ALL", "/var/cache/mod_proxy",,,
"ALL", "/var/cache/php-pear",,,
"ALL", "/var/cache/systemtap",,,
"ALL", "/var/empty",,,
"ALL", "/var/db/nsd",,,
"ALL", "/var/gdm",,,
"ALL", "/var/lib/dav",,,
"ALL", "/var/lib/dhcp",,,
"ALL", "/var/lib/dhclient",,,
"ALL", "/var/lib/php",,,
```

```

"ALL", "/var/lib/scsi/",,,
"ALL", "/var/lib/ups/",,,
"ALL", "/var/lib/random-seed",,,
"ALL", "/var/lib/iscsi",,,
"ALL", "/var/lib/logrotate.status",,,
"ALL", "/var/lib/ntp/",,,
"ALL", "/var/lib/xen/ntp",,,
"ALL", "/var/lock/",,,
"ALL", "/var/log/",,,
"ALL", "/var/run/",,,
"ALL", "/var/tmp/",,,
"ALL", "/var/tux/",,,
"ALL", "/opt/xcat/",,,
"ALL", "/xcatpost/",,,

```

5. Edit the litetree table. This table controls where the files specified in the litefile table come from.

```
# tabedit litetree
```

Copy the following into the litetree table. You will need to determine the Linux distribution you want. In our example, RHEL 5.4 is used.

```

#priority,image,directory,comments,disable
"1.0",,"10.1.100.1:/install/netboot/rhel5.4/s390x/compute",,

```

6. Edit the statelite table. This table controls where the permanent files are kept.

```
# tabedit statelite
```

Copy the following into the statelite table. You will need to determine the statelite node range and the IP address of the xCAT MN. In our example, the node range is **all** and the IP address is **10.1.100.1**.

```

#node,image,statemnt,comments,disable
"all",,"10.1.100.1:/lite/state",,

```

7. Create the persistent directory

```
# mkdir -p /lite/state
```

8. Ensure policies are set up correctly. When a node boots up, it queries the xCAT database to get the lite-files and the lite-tree. In order for this to work, the command must be set in the policy table to allow nodes to request it. (This should already be done automatically when xCAT was installed)

```

# chtab priority=4.7 policy.commands=litefile policy.rule=allow
# chtab priority=4.8 policy.commands=litetree policy.rule=allow

```

9. Download and copy the packages from the Linux distro media into /install (if not already)

```
# copycds -n xxx -a s390x /install/yyy.iso
```

where xxx is the distribution name and yyy is the ISO name.

For example, if you have a RHEL 5.4 ISO:

```
# copycds -n rhel5.4 -a s390x /install/RHEL5.4-Server-20090819.0-s390x-DVD.iso
```

10. Create a list of packages that should be installed onto the statelite image. You should start with the base packages in the compute template and if desired, add more packages by editing the .pkglist.

```

# mkdir -p /install/custom/netboot/rh
# cp /opt/xcat/share/xcat/netboot/sles/compute.rhe5.s390x.pkglist
/install/custom/netboot/rh

```

11. Create the statelite image

```
# genimage -i eth1 -n qeth -o rhel5.4 -p compute -m statelite
```

This command creates a RHEL 5.4 image with an eth1 interface, qeth network driver, and uses the compute profile. The interface used must match the xCAT MN interface that DHCP listens on. The genimage command creates an image under /install/netboot/rhel5.4/s390x/compute/rootimg. It also

creates a ramdisk and kernel that is used to boot the statelite node.

12. Modify the statelite image by creating symbolic links will all the files listed under the litetree table

```
# liteimg -o rhe15.4 -a s390x -p compute
```

13. Create the statelite node definition.

For our example, we will create a new node (gpok6) with a userID (LINUX6) that is managed by our zHCP (gpok2). You will need to substitute the node names, userIDs, and zHCP name with appropriate values.

```
# mkdef -t node -o gpok6 userid=LINUX6 hcp=gpok2.endicott.ibm.com mgt=zvm
groups=all
```

14. Update /etc/hosts

```
# makehosts
```

15. Update DNS

```
# makedns
```

16. Create the new virtual machine using the desired directory entry. For our example, we used the following:

```
USER LINUX6 PWD 512M 1G G
COMMAND SET VSWITCH VSW2 GRANT LINUX6
CPU 00 BASE
CPU 01
IPL CMS
MACHINE ESA 4
CONSOLE 0009 3215 T
NICDEF 0800 TYPE QDIO LAN SYSTEM VSW2
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
```

To create the virtual server, copy the directory entry above into a text file (dirEntry.txt) and issue the following command (the full file path must be given):

```
# mkvm gpok6 /tmp/dirEntry.txt
```

The new virtual server should be attached to the same vswitch as the one used by the hardware control point (in our case, VSW2) and have the same network adapter address (in our case, 0800) for the interface given in step 12 (in our case, eth1).

17. Clone this node as many times as you want to achieve the number of statelite nodes you desire. Refer to *Cloning Virtual Servers* section above. In order to clone, the source statelite node must be online and have SSH keys setup.

18. Add the new node to DHCP

```
# makedhcp -a
```

19. Set up the noderes and nodetype tables. The values of nodetype.os and nodetype.profile were determined in step 11, where the statelite image was created.

```
# chtab node=xxx noderes.netboot=zvm nodetype.os=yyy nodetype.arch=s390x
nodetype.profile=zzz
```

where xxx is the node name, yyy is the operating system, and zzz is the profile name.

In our example, we used the following:

```
# chtab node=gpok6 noderes.netboot=zvm nodetype.os=rhe15.4 nodetype.arch=s390x
nodetype.profile=compute
```


20. Prepare the node(s) to boot from the statelite image

```
# nodeset xxx statelite
```

where **xxx** is the node name.

21. Boot the statelite node(s). During this process, the symbolic links are made to files listed under the litefile table.

```
# rnetboot xxx ip1=00c
```

where **xxx** is the node name.

Caution: Do not try to boot more than 20 nodes at one time. The xCAT MN will be bogged down as all the nodes are trying to access the NFS server at once. Try booting 20 or less at a time and waiting till those nodes are pingable before booting the next batch.

22. Check the boot status of the node(s) by pinging them:

```
# pping xxx
```

where **xxx** is the node name. If the node returns a ping, then it is fully booted and you can start using it.

Updating Linux

This section details how to update the Linux operating system.

1. Download and xtract the ISO into the xCAT install tree /install (if not already)

```
# copycds -n xxx -a s390x /install/yyy.iso
```

where `xxx` is the distribution name and `yyy` is the ISO name.

For example, if you have a [SUSE Linux Enterprise Server 10 SP3](#) ISO:

```
# copycds -n sles10.3 -a s390x /install/SLES-10-SP3-DVD-s390x-DVD1.iso
```

or if you have a [Red Hat Enterprise Linux 5.4](#) ISO:

```
# copycds -n rhel5.4 -a s390x /install/RHEL5.4-Server-20090819.0-s390x-DVD.iso
```

2. Update the node

```
# updatenode xxx -o yyy
```

where `xxx` is the node name and `yyy` is the operating system version.

For example, if you want to update `gpok5` to RHEL5.4 (assuming `gpok5` has RHEL 5.3):

```
# updatenode gpok5 -o rhel5.4
```

The command requires the node to be online. It will take several minutes to complete the update. You can only update to the next release. For example, you can only update RHEL5.3 to RHEL5.4. You cannot skip releases, e.g. updating RHEL5.3 to RHEL5.5.

Warning: You cannot update SLES10.3 to SLES11. There is a bug in `rug` where you cannot add a repository/service.

Limitations

This section highlights the limitations of xCAT on z/VM and zLinux.

1. xCAT is only supported on z/VM 5.4 or newer, and SLES 10 SP2 or newer.
2. zHCP is only supported on RHEL 5.4 or newer, and SLES 10 SP2 or newer.
3. The default autoyast and kickstart templates available on xCAT was tested on SLES 10.2/10.3/11/11.1 and RHEL 5.3/5.4/5.5.
4. Statelite is only supported on SLES 11 or newer, and RHEL 5.4 or newer.
5. Nodes that the zHCP manages must have the Linux VMCP module.
6. SCSI/FCP disks are not yet supported. Only ECKD disks are supported.
7. The zHCP should be attached to only one network, and it must be a layer 2 network, either a guest LAN or VSwitch. Nodes that are managed by this zHCP must be on the same network.
8. In order for the xCAT MN to manage across multiple LPARs and CECs, you must use a layer 2 VSWITCH. The network hardware must be configured in such a way that these VSWITCHes can communicate across multiple LPARs and CECs.

Appendix A: Setting Up a Second Network

This section details how to setup a second network based on a layer 2 VSWITCH.

If you have [Red Hat Enterprise Linux](#):

SSH to the desire Linux where you want to setup the private network. A network script must be added under `/etc/sysconfig/network-scripts/` to let the system know about the new interface and a qeth group must be created under `/sys/bus/ccwgroup/drivers/qeth/group`.

In the following example, we will configure an ethernet interface (***eth1***) for a layer 2 VSWITCH attached to ***0800***. We will assume there is an existing ethernet interface (***eth0***) attached to ***0600***.

Copy the hardware settings from the existing network `/etc/sysconfig/network-scripts/ifcfg-eth0`.

```
# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-  
scripts/ifcfg-eth1
```

Edit the network settings.

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

It should look similar to the following:

```
# IBM QETH  
DEVICE=eth1  
ARP=no  
BOOTPROTO=static  
BROADCAST=10.1.100.255  
IPADDR=10.1.100.1  
IPV6INIT=yes  
IPV6_AUTOCONF=yes  
MTU=1500  
NETMASK=255.255.255.0  
NETTYPE=qeth  
NETWORK=10.1.100.0  
ONBOOT=yes  
PORTNAME=PORT800  
OPTIONS="layer2=1"  
SUBCHANNELS=0.0.0800,0.0.0801,0.0.0802  
MTU=1500
```

Load the qeth driver

```
# modprobe qeth
```

Create a qeth group device

```
# echo 0.0.0800,0.0.0801,0.0.0802 > /sys/bus/ccwgroup/drivers/qeth/group
```

Declare the qeth group device as Layer 2

```
# echo 1 > /sys/bus/ccwgroup/drivers/qeth/0.0.0800/layer2
```

Bring the device back online (you need to reset the device after each reboot)

```
# echo 1 > /sys/bus/ccwgroup/drivers/qeth/0.0.0800/online
```

Verify the state of the device (1 = online)

```
# cat /sys/bus/ccwgroup/drivers/qeth/0.0.0800/online
```

Check to see what interface name was assigned to the device

```
# cat /sys/bus/ccwgroup/drivers/qeth/0.0.0800/if_name
```

A qeth device requires an alias definition in /etc/modprobe.conf. Edit this file and add an alias for your interface

```
# vi /etc/modprobe.conf
```

```
alias eth0 qeth
alias eth1 qeth
options dasd_mod dasd=0.0.0100,0.0.0103,0.0.0300,0.0.0301
```

Start the new interface

```
# ifup eth1
```

If you have [SUSE Linux Enterprise Server 10](#):

SSH to the desire Linux where you want to setup the private network. Two configuration files must be added under /etc/sysconfig/ to let the system know about the new interface, one for hardware and one for network settings.

In the following example, we will configure an ethernet interface (**eth1**) for a layer 2 VSwitch attached to **0800**. We will assume there is an existing ethernet interface (**eth0**) attached to **0600**.

Copy the hardware settings from the existing network /etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-**0.0.0600**. Both interfaces will use the qdio/qeth drivers, therefore, the configuration files can be identical except for the virtual addresses. The existing file is copied to specify the new NIC. The only difference needed is to change the **060X** values to **080X**.

```
# cd /etc/sysconfig/hardware/
# sed *600 -e 's/060/080/g' > hwcfg-qeth-bus-ccw-0.0.0800
```

Edit the hardware settings.

```
# vi hwcfg-qeth-bus-ccw-0.0.0800
```

It should look similar to the following:

```
STARTMODE="auto"
MODULE="qeth"
MODULE_OPTIONS=""
MODULE_UNLOAD="yes"
SCRIPTUP="hwup-ccw"
SCRIPTUP_ccw="hwup-ccw"
SCRIPTUP_ccwgroup="hwup-qeth"
SCRIPTDOWN="hwdown-ccw"
CCW_CHAN_IDS="0.0.0800 0.0.0801 0.0.0802"
CCW_CHAN_NUM="3"
CCW_CHAN_MODE="OSAPORT"
QETH_LAYER2_SUPPORT="1"
```

Copy the network settings from the existing network /etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.0600.

```
# cd /etc/sysconfig/network
# cp ifcfg-qeth-bus-ccw-0.0.0600 ifcfg-qeth-bus-ccw-0.0.0800
```

Edit the network settings.

```
# vi ifcfg-qeth-bus-ccw-0.0.0800
```

It should look similar to the following:

```

BOOTPROTO="static"
UNIQUE=""
STARTMODE="onboot"
IPADDR="10.1.100.1"
NETMASK="255.255.255.0"
NETWORK="10.1.100.0"
BROADCAST="10.1.100.255"
_nm_name='qeth-bus-ccw-0.0.0800'

```

Reboot the virtual server to have the changes take effect.

```
# reboot
```

If you have [SUSE Linux Enterprise Server 11](#):

SSH to the desire Linux where you want to setup the private network. A configuration file must be added under /etc/sysconfig/network and /etc/udev/rules.d to let the system know about the new interface.

In the following example, we will configure an ethernet interface (**eth1**) for a layer 2 vswitch attached to **0800**. We will assume there is an existing ethernet interface (**eth0**) attached to **0600**.

Copy the hardware settings from the existing network /etc/udev/rules.d/51-qeth-**0.0.0600**.rules. Both interfaces will use the qdio/qeth drivers, therefore, the configuration files can be identical except for the virtual addresses. The existing file is copied to specify the new NIC. The only difference needed is to change the **060X** values to **080X**.

```
# sed /etc/udev/rules.d/51-qeth-0.0.0600.rules -e 's/060/080/g' >
/etc/udev/rules.d/51-qeth-0.0.0800.rules
```

Edit the udev rules. You must also enable layer2 for the device. Take note of ATTR{layer2}="1".

```
# vi /etc/udev/rules.d/51-qeth-0.0.0800.rules
```

It should look similar to the following:

```

# Configure qeth device at 0.0.0800/0.0.0801/0.0.0802
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth", IMPORT{program}="collect
0.0.0800 %k 0.0.0800 0.0.0801 0.0.0802 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.0800", IMPORT{program}="collect
0.0.0800 %k 0.0.0800 0.0.0801 0.0.0802 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.0801", IMPORT{program}="collect
0.0.0800 %k 0.0.0800 0.0.0801 0.0.0802 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.0802", IMPORT{program}="collect
0.0.0800 %k 0.0.0800 0.0.0801 0.0.0802 qeth"
TEST=="[ccwgroup/0.0.0800]", GOTO="qeth-0.0.0800-end"
ACTION=="add", SUBSYSTEM=="ccw", ENV{COLLECT_0.0.0800}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.0800,0.0.0801,0.0.0802"
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth",
ENV{COLLECT_0.0.0800}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.0800,0.0.0801,0.0.0802"
LABEL="qeth-0.0.0800-end"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.0800",
ATTR{portname}="OSAPORT"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.0800", ATTR{portno}="0"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.0800", ATTR{layer2}="1"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.0800", ATTR{online}="1"

```

Copy the network settings from the existing network /etc/sysconfig/network/ifcfg-eth0.

```
# cp /etc/sysconfig/network/ifcfg-eth0 /etc/sysconfig/network/ifcfg-eth1
```

Edit the network settings.

```
# vi /etc/sysconfig/network/ifcfg-eth1
```

It should look similar to the following:

```
BOOTPROTO='static'  
IPADDR='10.1.100.1'  
BROADCAST='10.1.100.255'  
NETMASK='255.255.255.0'  
NETWORK='10.1.100.0'  
STARTMODE='onboot'  
NAME='OSA Express Network card (0.0.0800)'
```

Reboot the virtual server to have the changes take effect.

```
# reboot
```

Appendix B: Customizing Autoyast and Kickstart

This section details how to customize the autoyast and kickstart templates.

Autoyast and kickstart allows you to customize a Linux system based on a template. While you would typically go through various panels to manually customize your Linux system during boot, you no longer have to with autoyast and kickstart. This allows you to configure a vanilla Linux system faster and more effectively.

If you want to customize SUSE Linux Enterprise Server:

1. Base your customization on the default templates in `/opt/xcat/share/xcat/install/sles`. The ones available are `compute.sles10.s390x.tmpl` and `compute.sles11.s390x.tmpl`; you need select the one that is appropriate for the operating system you are planning to customize. These templates are configured to setup the network for you using DHCP.
2. Determine the number of disks (ECKD only) your vanilla system will have and the mount points for each disk.
3. Copy the default template `/opt/xcat/share/xcat/install/sles/xxx.tmpl`, where `xxx` is the template name, into `/install/custom/install/sles/`. For our example, we will use `compute.sles11.s390x.tmpl`:

```
# cp /opt/xcat/share/xcat/install/sles/compute.sles11.s390x.tmpl
/install/custom/install/sles/custom.sles11.s390x.tmpl
```
4. Add this template to the `osimage` table. For our example, we customized the autoyast template for SLES 11.1 and added it to the `osimage` table using:

```
# chtab imagename=sles11.1-s390x-install-custom
osimage.profile=custom.sles11.s390x.tmpl osimage.imagetype=linux
osimage.provmethod=install osimage.osname=Linux osimage.osvers=sles11.1
```
5. Add the extra disk to the custom template using the following format:

```
<!-- Dasd attached at 0101 -->
<listentry>
  <bus>None</bus>
  <bus_hwcfg>none</bus_hwcfg>
  <channel>0.0.0101</channel>
  <format config:type="boolean">>true</format>
  <dev_name>/dev/dasdb</dev_name>
  <dev_names config:type="list">
    <listentry>/dev/dasdb</listentry>
    <listentry>/dev/disk/by-path/ccw-0.0.0101</listentry>
  </dev_names>
  <device>DASD</device>
  <driver>io_subchannel</driver>
  <drivers config:type="list">
    <listentry>
      <active config:type="boolean">>true</active>
      <modprobe config:type="boolean">>true</modprobe>
      <modules config:type="list">
        <module_entry config:type="list">
          <listentry>dasd_eckd_mod</listentry>
          <listentry></listentry>
        </module_entry>
      </modules>
    </listentry>
  </drivers>
```



```

<formatted config:type="boolean">true</formatted>
<partition_info>/dev/dasdb1 (Linux native)</partition_info>
<resource>
  <disk_log_geo config:type="list">
    <listentry>
      <heads config:type="integer">15</heads>
      <sectors config:type="integer">12</sectors>
    </listentry>
  </disk_log_geo>
  <io config:type="list">
    <listentry>
      <active config:type="boolean">true</active>
      <length config:type="integer">1</length>
      <mode>rw</mode>
    </listentry>
  </io>
</resource>
<sysfs_bus_id>0.0.0101</sysfs_bus_id>
</listentry>

```

The variables highlighted in **red** need to be modified based on how you customize your Linux. In the example above, a disk attached at **0101** is added. The device name for this disk is **dasdb**. The device name is determined by how many devices are attached to the Linux system. For example, the first device gets a name of **dasda**, the second device gets a name of **dasdb**, and so on. You need to place the new **dasd** before the swap space and rename the device names for each swap space following the rule above. Once complete, add it to the `<dasd>` section of the template.

6. Add a module entry for the new disk:

```

<module_entry>
  <device>dasd-bus-ccw-0.0.0101</device>
  <module>dasd_eckd_mod</module>
  <options></options>
</module_entry>

```

The variables highlighted in **red** need to be modified based on how you customize your Linux. In the example above, a disk attached at **0101** is added. Once complete, add it to the `<modules>` section of the template.

7. Add the new mount point for the new disk:

```

<!-- /usr partition -->
<drive>
  <device>/dev/dasdb</device>
  <partitions config:type="list">
    <partition>
      <create config:type="boolean">true</create>
      <filesystem config:type="symbol">ext3</filesystem>
      <format config:type="boolean">true</format>
      <mount>/usr</mount>
      <mountby config:type="symbol">path</mountby>
      <partition_id config:type="integer">131</partition_id>
      <partition_nr config:type="integer">1</partition_nr>
      <partition_type>primary</partition_type>
    </partition>
  </partitions>

```

```

    <size>max</size>
  </partition>
</partitions>
<use>all</use>
</drive>

```

The variables highlighted in **red** need to be modified based on how you customize your Linux. In the example above, a disk attached at **0101** is added. The device name given to disk is **dasdb**. This disk will be mounted at **/usr** and will have a **ext3** file system. You need to place the new mount point before the swap space and rename the device names for each swap space following the naming rule. Once complete, add it to the <partitioning> section of the template.

8. Add the software you need to the <software> section of the template. You need to determine the package name and add it to the patterns list, e.g. <pattern>gnome</pattern>.

For more information, refer to http://www.suse.de/~ug/autoyast_doc/index.html.

If you want to customize Red Hat Enterprise Server:

1. Base your customization on the default template (compute.rhel5.s390x.tmpl) in /opt/xcat/share/xcat/install/rh/. This template is configured to setup the network for you using DHCP.
2. Determine the number of disks (ECKD only) your vanilla system will have and the mount points for each disk.
3. Copy the default template /opt/xcat/share/xcat/install/rh/**xxx**.tmpl, where xxx is the template name, into /install/custom/install/rh/. For our example, we will use compute.rhel5.s390x.tmpl:

```

# cp /opt/xcat/share/xcat/install/rh/compute.rhel5.s390x.tmpl
  /install/custom/install/rh/custom.rhel5.s390x.tmpl

```

4. Add this template to the osimage table. For our example, we customized the kickstart template for RHEL 5.4 and added it to the osimage table using:

```

# chtab imagename=rhel5.4-s390x-install-custom
  osimage.profile=custom.rhel5.s390x.tmpl osimage.imagetype=rhel5.4
  osimage.osname=linux

```

5. Add the disk and mount point to the template using the following format:

```

clearpart --initlabel -drives=dasda,dasdb
part / --fstype ext3 --size=100 --grow -ondisk=dasda
part /usr --fstype ext3 --size=100 --grow -ondisk=dasdb

```

The variables highlighted in **red** need to be modified based on how you customize your Linux and appended to the template. In the example above, a disk is added with a device name of **dasdb**. The disk will be mounted at **/usr** and will have a **ext3** file system.

6. Add the software you need to the %packages section.

For more information, refer to <http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/custom-guide/s1-kickstart2-file.html>.

