

Automating Linux cloud installations

Simplify the process for installing Linux on a new Power System or System p LPAR

Skill Level: Intermediate

[W. David Ashley \(dashley@us.ibm.com\)](mailto:dashley@us.ibm.com)

Senior IT Specialist

IBM

[David Boutcher](#)

Former Senior Technical Staff Member

IBM

11 Nov 2008

A simplified process for installing operating systems can reduce the amount of time you spend on administration of your cloud computing environment. This article describes how you can automate the installation of SUSE Linux® on a new IBM Power™ System or System p® LPAR. You can also use the same techniques to install Red Hat Linux or AIX®.

One of the features of cloud computing is the ability to move applications from one processor environment to another. This feature requires that the target operating system for the application exists *before* an application can be moved. Wouldn't it be nice to be able to automate the installation of the new operating system?

Automated installations of Linux are a well-known feature of Intel™ architecture systems. However, this is a tougher problem to solve for System p or IBM Power Systems using a hardware management console (HMC). One of the advantages of the solution described in this article is that it is a pure Linux solution and does not require that you have any specific AIX skills.

Overview of the automation solution

The goal of this solution is to automate the operating system installation by using a set of configurations that can be easily maintained and modified. The solution has the following characteristics :

- The newly created and installed LPAR uses a static IP address. This relates only to the final configuration of the LPAR; you can use the dynamic host configuration protocol (DHCP) during the install process.
- The Automatic Linux Installation and Configuration with YaST2 (AutoYaST) configuration XML file is generic enough that it can be used for a variety of server types including HTTP, MySQL, etc.
- The AutoYaST file does *not* contain system specific information such as an IP address, host name, etc.
- Everything is automated in such a way that you can use a single command at the HMC to install the new LPAR.

The steps to build and use the automated solution are:

1. [Configuring the AutoYaST file](#)
2. [Configuring the DHCP/BOOTP and TFTP servers](#)
3. [Using the HMC `lpar_netboot` command](#)
4. [Automating `lpar_netboot` with the `-g` parameter](#)
5. [Reusing the automated solution](#)

Step 1. Configuring the AutoYaST file

SUSE Linux uses an XML configuration file named AutoYaST.xml to control the installation of operating systems. By default, AutoYaST installs a system with an identical configuration each time you use it. This enables you to have individual AutoYaST files for different configurations. For example, you may have one AutoYaST file for Web servers and another for MySQL servers. However, configuring and maintaining multiple AutoYaST files for servers that differ only in their IP address and host name can be difficult and time consuming.

To create a customized AutoYaST file that acquires an IP address and host name from the DHCP server, make a copy of the AutoYaST file from your SUSE installation CD and modify it as follows:

1. The standard AutoYaST file is set up to use DHCP. Modify the network

sections so that DHCP is used for only the initial install of the operating system. Check to be sure that there is no host name being assigned in any of the network sections. If there is, then completely remove that section.

2. If the XML file contains static IP information, DNS information, or any other network information, then remove those sections from the file.
3. Copy the code from Listing 1 and place it in the AutoYaST file just before the last XML tag. This new section contains a script that converts the current booted DHCP configuration into a static network configuration that will be used the next time the LPAR is booted. This script eliminates the need to have a separate AutoYaST file with a hard-coded IP address for each LPAR. Your only administration point is maintaining the DHCP server.

Listing 1. AutoYaST DHCP conversion script

```
<scripts>
  <init-scripts config:type="list">
    <listentry>
      <filename>network_setup</filename>
      <interpreter>shell</interpreter>
      <source><![CDATA[#!/bin/bash
### Disable ipv6
echo 'install ipv6 /bin/true' >> /etc/modprobe.conf.local

### Variables to Calculate Network Configuration Settings for a static
configuration ###
ACTIVE_INTERFACE=`/sbin/ifconfig | grep eth | awk '{print $1}'`
IP_ADDRESS=`/sbin/ifconfig $ACTIVE_INTERFACE | grep 'inet addr' | awk
'{print $2}'
| sed 's/addr://'`
NETMASK=`/sbin/ifconfig $ACTIVE_INTERFACE | grep 'inet addr' | awk
'{print $4}'
| sed 's/Mask://'`
BROADCAST=`/sbin/ifconfig $ACTIVE_INTERFACE | grep Bcast: | awk '{ print
$3 }'
| sed 's/Bcast://'`
NETWORK=`/sbin/ip route list | grep $IP_ADDRESS | awk '{ print $1 }'
| sed 's/\/[1-9][1-9]//'`
GATEWAY=`/sbin/route | grep default | awk '{print $2}'`
HOSTNAME=`/usr/bin/host $IP_ADDRESS | awk '{print $5}' | sed 's/.$//'`

### Setup HOSTNAME
echo "$HOSTNAME" > /etc/HOSTNAME

### Setup Gateway Address
echo "default $GATEWAY - -" > /etc/sysconfig/network/routes

### Setup /etc/hosts with correct host information
HOST=`/usr/bin/host $IP_ADDRESS | awk '{print $5}' | sed 's/.$//' | cut
-d "." -f 1`
echo "$IP_ADDRESS $HOSTNAME $HOST" >> /etc/hosts

### Network configuration file rewrite for static configuration
INT_CONF_FILE=/etc/sysconfig/network/ifcfg-eth-id-`ifconfig eth0 | grep
```

```
HWaddr
| awk '{ print $5 }' | perl -ne '$var=$_; print lc($var)'\`

echo 'DEVICE=eth0' > $INT_CONF_FILE
echo 'BOOTPROTO=static' >> $INT_CONF_FILE
echo "IPADDR=$IP_ADDRESS" >> $INT_CONF_FILE
echo "NETMASK=$NETMASK" >> $INT_CONF_FILE
echo "BROADCAST=$BROADCAST" >> $INT_CONF_FILE
echo "NETWORK=$NETWORK" >> $INT_CONF_FILE
echo 'STARTMODE=onboot' >> $INT_CONF_FILE
echo 'TYPE=Ethernet' >> $INT_CONF_FILE

### Restart Network
/etc/init.d/network restart
    ]]></source>
</listentry>
</init-scripts>
</scripts>
```

Step 2. Configuring the DHCP/BOOTP and TFTP servers

This solution uses a standard Linux DHCP server that is a part of all major Linux distributions. In this step, you configure the DHCP server to provide network boot information to new LPARs. The initial code is loaded into the LPAR via the Trivial File Transfer Protocol (TFTP) server. To set up the network boot, you need both the DHCP server and TFTP server loaded on a server in the network.

The DHCP server used for this solution is not configured to dynamically provide TCP/IP addresses to systems in the network, but it can coexist on a network with DHCP servers that do perform this traditional role. This type of configuration works even if the DHCP server is on a different network than the system being booted. For example, you could use it to dynamically boot systems in New York from servers in Texas.

For the `lpar_netboot` command to work as described in [Step 3](#) of this solution, you need to configure the DHCP server for each LPAR you install. The advantage of this approach is that all the configuration information for automatically installed LPARs is contained in the `dhcpd.conf` file. Therefore, the AutoYaST file that you modified in the previous step does not need to contain any hard-coded information that defines the network configuration of the LPAR.

Listing 2 contains a sample `dhcpd.conf` file. Use the sample as a model and add additional `host` definitions for each LPAR you want to configure. The zip file in the [Download](#) section of this article includes a script named `addsystem` that you can use to automate the editing of the `dhcpd.conf` file.

Each time you make changes to the `dhcpd.conf` file, you must stop and restart the DHCP server to make the changes take effect.

Listing 2. Sample `dhcpd.conf` file

```
ddns-update-style ad-hoc;
allow bootp;
subnet 0.0.0.0 netmask 0.0.0.0 {
    host rcc-hny-poc-003 {
        option host-name          "myserver.cloud9.ibm.com";
        fixed-address             192.168.0.203;
        hardware ethernet 16:E9:10:94:87:03;
        filename                  "inst64";
    }
    next-server                   192.168.0.201;
    option routers                192.168.0.201;
}
}
```

Following are details about the options in the sample `dhcp.conf` file:

- `subnet` configures this server to respond to requests from other servers regardless of where they are in the network. Because you are not configuring this DHCP server to perform dynamic addressing, it will respond to the boot requests from your LPARs without interfering with normal DHCP operation on the network.
- `host` sections define each LPAR you want to configure. You can have as many `host` sections as you need in the configuration file. The options contained within the `host` section in the sample are the minimum needed to install the LPAR and be able to continue to use the entry for subsequent booting of the LPAR as a DHCP client:
 - `host-name` is the name assigned to the IP address of the client by the DNS. If the client has no DNS entry, this option should specify the name you want to be contained in the local host file of the LPAR.
 - `fixed-address` is the IP address to be assigned to the LPAR. The specified address is used instead of defaulting to the DHCP server's pool of addresses. This has the advantage of reserving this IP address for this LPAR on every boot of the operating system.
 - `hardware ethernet` specifies the MAC address of the LPAR. This is used during the BOOTP phase of the installation to associate the LPAR with this `host` entry.
 - `filename` specifies the Linux kernel to be loaded when servicing a BOOTP request. This file must be accessible via TFTP on the server specified on the `next-server` option. Or, if `next-server` is omitted, the file needs to be accessible via TFTP on the DHCP server.
- `next-server` specifies the address of the server where the Linux kernel defined on the `filename` options is accessible via TFTP. If the TFTP server is located on the same server as the DHCP server, you can omit

this option.

- `option routers` specifies the address of your TFTP server if your routers are configured in such a way that the TFTP server request is not forwarded on the network to the TFTP server. If the TFTP server and the DHCP server are on the same machine, you can omit this option.

To configure the TFTP server, you can use the `atftpd` package available in SLES distributions. The TFTP configuration provides a directory where files can be downloaded using TFTP. The download directory is usually `/tftpboot`. The file you specify on the `filename` parameter of each `host` section in your `dhcpd.conf` file needs to be available in this download directory.

For example, to network-install SUSE, the file is the `inst64` file available from the SUSE installation CD in the `/suseboot` directory. You would need to place that file in the `/tftpboot` directory and ensure that its permissions allow it to be readable by everyone.

Step 3. Using the HMC `lpar_netboot` command

Just as the HMC can be used to activate an existing LPAR, it can also be used to install an operating system on a new LPAR. Although the HMC provides a graphical user interfaces, all HMC functions are also available through the command line. By using the command line, you can further automate this step of the solution.

The HMC command that kicks off a network install is `lpar_netboot`. The [Resources](#) section of this article contains a link to complete details on using the `lpar_netboot` command. This solution uses the form of `lpar_netboot` that network-boots an LPAR.

Listing 3 is an example of an `lpar_netboot` command. Modify the example as necessary, then use it to power on an LPAR and attempt to network-boot it from a server in the network.

The example command assumes the following:

- As described in the previous step of this article, your DHCP/BOOTP server is configured with the MAC address of the new LPAR, and it has an entry for the SUSE install program.
- You plan to always boot the new LPAR as a DHCP client with the static IP address defined for it in the `dhcpd.conf` file.

Listing 3. Sample `lpar_netboot` command

```
lpar_netboot -f -t ent -m 16E910948703 -s auto -d auto -S 192.168.0.201
```

```
-G \  
192.168.0.201 -C 192.168.0.203 lpar1 lpar1 SystemA
```

Note: Listing 3 uses the backslash character to indicate a line continuation; it is not part of the command. When entering the command, do not use the backslash.

Following are details about the parameters used in the example:

- `-m` is the MAC address of the client to be booted. You can obtain the MAC address by using another form of the `lpar_netboot` command.
- `-S` is the IP address of the DHCP/BOOTP server.
- `-G` is the IP address of the gateway needed to reach the DHCP/BOOTP server. If the DHCP/BOOTP server is on the local network, then use the IP address of the DHCP/BOOTP server instead.
- `-C` is the IP address you want to assign to the booted client.
- The three names at the end of the command are:
 - The HMC name for the LPAR (`lpar1`)
 - The LPAR configuration name, which by default is the same name as the name of the LPAR itself (`lpar1`)
 - The name of the server that owns the LPAR at the HMC (`SystemA`)

The zip file in the [Download](#) section of this article includes a sample script named `autoinstall`. You can install this script on your HMC server to automate the installation of a new LPAR using the techniques described in this article. The script automatically obtains an LPAR's MAC address, calls the DHCP `addsystem` script to configure and restart the DHCP server, and calls the `lpar_netboot` command to install the LPAR. Lastly, it invokes a `mkvterm` command that lets you watch the install take place.

Step 4. Automating `lpar_netboot` with the `-g` parameter

Using the previous steps in this article, your new LPARs load the Linux installer. Then if you open a console connection to the LPAR from the HMC, you see the first screen of the installation program.

To further automate the install, you can use the `-g` argument of the `lpar_netboot` command to pass additional parameters to the install program of the newly booted system. Linux distributions, such as SUSE and Red Hat, allow fully automated installation over the network as long as you pass the correct information to the installation program to kick off the install.

The information you pass on the `-g` argument is a string that contains the information needed by the installer to find the SUSE AutoYaST XML file and the SUSE operating system install RPMs. The format of this information is documented by each distribution and is the same for all architectures (i386, Power, etc.).

Listing 4 is an example of a `-g` parameter that kicks off an automated install from a server. Modify the example as necessary, then use it with an `lpar_netboot` command to kick off an automated installation of the operating system.

The example assumes the following:

- You have a Web server running on your installation server. The Web server can be the same server that has the DHCP server and TFTP server, but it does not have to be.
- You have copied the SUSE installation CD image to a directory under the Web server document root directory.
- You have placed a copy of the AutoYaST.xml file that you configured in [Step 1](#) of this article in the same Web server directory as the SUSE installation image.

Listing 4. Sample `lpar_netboot` command `-g` parameter

```
-g "hostip=192.168.0.203 netmask=255.255.255.0 \  
gateway=192.168.0.1 nameserver=192.168.0.1 insmod=ibmveth \  
install=http://192.168.0.201/suse \  
autoyast=http://192.168.0.201/autoyast.xml"
```

Note: Listing 4 uses backslash characters to indicate line continuations; the characters are not part of the command. When entering the command, do not use the backslash characters.

The `insmod` argument in Listing 4 loads a virtual Ethernet adapter. If the LPAR you are booting has physical Ethernet adapters assigned to it, you do not need this argument.

Step 5. Reusing the automated solution

You now have a configuration that lets you easily manage LPARs in a cloud environment. Use the following steps to bring up new LPARs:

1. Create the LPAR at the HMC.
2. Log on to the HMC SSH interface and run `lpar_netboot` to determine

the MAC address of the LPAR.

3. Reconfigure the `dhcpd.conf` file with the new LPAR information.
4. Stop and restart the DHCP server to make the changes to `dhcpd.conf` take effect.
5. Log on to the HMC again and run `lpar_netboot` to install the operating system on the LPAR.

All you have to do now is wait for the operating system installation to finish. After that, your cloud environment has a new LPAR that is ready to accept the processes you want to load on it.

Downloads

Description	Name	Size	Download method
Code for this article	autoinstall.zip	2KB	HTTP

[Information about download methods](#)

Resources

Learn

- Learn more about the [lpar_netboot command](#) in the online reference entry from the IBM Systems Information Center.
- The article "[DHCP \(Dynamic Host Configuration Protocol\)](#)" looks at configuring a DHCP server.
- For more details on cloud computing platforms and applications, see "[Cloud computing with Linux](#)" (developerWorks, September 2008).
- For additional background on cloud computing, see "[Cloud computing with Amazon Web Services, Part 1: Introduction](#)" (developerWorks, July 2008).
- In the [developerWorks Linux zone](#), find more resources for Linux developers (including developers who are [new to Linux](#)), and scan our [most popular articles and tutorials](#).
- See all [Linux tips](#) and [Linux tutorials](#) on developerWorks.
- Stay current with [developerWorks technical events and Webcasts](#).

Get products and technologies

- [Order the SEK for Linux](#), a two-DVD set containing the latest IBM trial software for Linux from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.
- With [IBM trial software](#), available for download directly from developerWorks, build your next development project on Linux.

Discuss

- Get involved in the [developerWorks community](#) through blogs, forums, podcasts, and spaces.

About the authors

W. David Ashley

David Ashley has over 25 years of IT experience working with systems ranging from embedded devices to large IBM mainframe installations. He has been an IT instructor, development team lead, systems architect, systems management consultant, and a database integrator. He currently works for IBM Lab Services as a consultant on System p projects and contributes to the open source Open Object Rexx project in his spare time.

David Boutcher

David Boutcher was a Senior Technical Staff Member in the IBM Linux Technology Center and worked at IBM development laboratories from 1986 until recently on a variety of operating systems. His work has focused primarily on the areas of communications and distributed computing.

Trademarks

IBM, the IBM logo, ibm.com, DB2, developerWorks, Lotus, Rational, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. See the current list of [IBM trademarks](#).

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.