xCAT 2 on AIX

How-To: Install AIX standalone nodes

(using standard NIM rte method)

Version: 1.0

Date: 6/16/2008

1.0 Installing AIX nodes (using standard NIM rte method)

This “How-To” describes how AIX standalone nodes can be deployed using xCAT and AIX/NIM commands.

NIM is an AIX tool that enables a cluster administrator to centrally manage the installation and configuration of AIX and optional software on machines within a networked environment. Setting up NIM involves various tasks including, installing NIM software, configuring NIM and creating some basic NIM installation resources.

The specific tasks that you need to perform depend on which features of NIM that you plan to use. For more information about NIM, see the *IBM AIX Installation Guide and Reference*. (http://www-03.ibm.com/servers/aix/library/index.html)

This “How-To” describes only one basic set of steps that may be used to install an AIX standalone node using the NIM “rte” installation method and is not meant to be a comprehensive guide of all the available NIM options.

1.1 Assumptions

- An AIX system has been installed to use as an xCAT management node.
- All relevant base AIX services are configured and running. This includes (but is not limited to) bootp, tftp, NFS, and hostname resolution.
- The cluster network is set up.
- xCAT and prerequisite software has been installed on the management node.
- LPARs have already been created using the HMC interfaces.
- You have some experience using AIX Network Installation Management (NIM).

Req – ethernet connection - management network etc.
– xCAT network defs – needed for AIX setup
– name resolution - DNS
– gathering required network info
– config nodes (resolv_conf, passwords, hosts, rhosts?)
TBD

1.2 Install and Configure NIM

NIM enables a cluster administrator to centrally manage the installation and configuration of AIX and optional software on machines within a networked environment. Setting up NIM involves various tasks including, installing NIM software, configuring NIM and creating some basic NIM installation resources.

The specific tasks that you need to perform depend on which features of NIM that you plan to use. For more information about NIM, see the *IBM AIX Installation Guide and Reference*. (http://www-03.ibm.com/servers/aix/library/index.html)

To simplify the NIM setup process, you can use the AIX **nim_master_setup** command. This command automatically installs NIM software, configures NIM, creates basic resources and creates a resource group containing the resources that are created. This command must be run on the machine that will be used as the NIM master. See the **man** page for details.

When you run the command it will display a list of the resources that were created. If the default resources that are created by this script are not what you need, you can use individual NIM commands to create additional resources. The resources created are stored in the */export/nim* directory by default. The file system is automatically created with the correct size needed to store the required resources.

In this example we will be doing a NIM “rte” type installation so we will disable the creation of the backup image (*mksysb*) by using the “-B” option. The default location of the NIM master install images is /dev/cd0. If you are using another

device or directory as a source you would have to specify that on the command line. Assuming the images are available from the default device you could run the command as follows.

nim_master_setup -B -a filesystem=/install

This assumes that the required NIM software is available on /dev/cd0. If this is not the case then you must specify a different source directory (device). (ex. -a device=/ mnt/610_lpp")

You could also do the NIM configuration using the AIX SMIT interfaces. (Just type the SMIT fastpath "smit eznim" to get started.)

To make sure that NIM is set up correctly issue the **lsnim** command on the NIM master as follows.

lsnim

You should see output similar to the following:

```
master              machines    master
boot                resources   boot
nim_script          resources   nim_script
master_net          networks    ent
master_net_conf     resources   resolv_conf
bid_ow              resources   bosinst_data
530lpp          resources   lpp_source
530spot         resources   spot
basic_res_grp       groups      res_group
```

To get details for each resource definition use "**lsnim -l <resource name>**". For example, if the name of your SPOT resource is "*530spot_res*" then you could get the details by running:

lsnim -l 530spot

To see the actual contents of a resource use "**nim -o showres <resource name>**". For example, to show the file sets that are installed in your SPOT you could run:

nim -o showres 530spot | pg

1.3 Create additional NIM definitions and resources if needed.

For this example we are assuming that the xCAT management node and the LPARs are all connected to the same network and that the resources created by the **nim_master_setup** command are sufficient.

However, depending on your specific situation, you may need to create additional NIM resources or definitions.

For example, if the management node is not on the same network as the nodes you would have to create additional NIM network and route definitions. NIM network definitions represent the networks used in the NIM environment. When you run **nim_master_setup**, the network associated with the NIM master is automatically defined. You need to define additional networks only if there are nodes that reside on other local area networks or subnets. If the physical network is changed in any way, the NIM network definitions need to be modified. See the NIM documentation for details on creating additional network and route definitions.

Note: The NIM machine definitions will be generated automatically from the xCAT node definitions in a later step.

1.4 Create an osimage definition

Use the xCAT **mknimimage** command to create an xCAT *osimage* definition. xCAT uses the information in this definition to keep track of a unique operating system image and how it will be deployed.

When you run the command you must provide the names of the NIM resources that you wish to use for this image. In this case we will use the resources that were created when we ran the nim_master_setup command.

For example, to create an *osimage* called "53*gold*" you could issue the following command.

mknimimage 53gold spot=530spot lpp_source=530lpp bosinst_data=bid_ow resolv_conf=master_net_conf nim_script=nim_script boot= boot

Or, since all these resources are contained in the – resource group you could issue the following command.

mknimimage 53gold res_group=basic_res_grp

See the **mknimimage man** page for additional details.

To display the xCAT image definition run the xCAT **lsdef** command.

lsdef -t osimage -l -o 61cosi

To get details for the NIM resource definitions use the AIX **lsnim** command. For example, if the name of your SPOT resource is "*61cosi*" then you could get the details by running:

lsnim -l 61cosi

To see the actual contents of a NIM resource use "*nim -o showres <resource name>*". For example, to get a list of the software installed in your SPOT you could run:

nim -o showres 61cosi

1.5 Define xCAT networks

Create a network definition for each network that contains cluster nodes. You will need a name for the network and values for the following attributes.

net The network address.
mask The network mask.
gateway The network gateway.

In our example we will assume that all the cluster node management interfaces and the xCAT management node interface are on the same network. You can use the xCAT **mkdef** command to define the network.

For example:

mkdef -t network -o net1 net=9.114.113.224 mask=255.255.255.224 gateway=9.114.113.254

Note: NIM also requires network definitions. When NIM was configured in an earlier step the default NIM master network definition was created. The NIM definition should match the one you create for xCAT. If multiple cluster subnets are needed then you will need an xCAT and NIM network definition for each one. A future xCAT enhancement will simplify this by automatically creating NIM network definitions based on the xCAT definitions.

1.6 Define the HMC as an xCAT node

The xCAT hardware control support requires that the hardware control point for the nodes also be defined as a cluster node.

The following command will create an xCAT node definition for an HMC with a host name of "*hmc01*". The *groups, nodetype, mgt, username*, and *password* attributes must be set.

mkdef –t node –o hmc01 groups="all" nodetype=hmc mgt=hmc username=hscroot password=abc123

1.7 Discover the LPARs managed by the HMC

This step assumes that the partitions are already created using the standard HMC interfaces.

Use the **rscan** command to gather the LPAR information. This command can be used to display the LPAR information in several formats and can also write the

LPAR information directly to the xCAT database. In this example we will use the "-z" option to create a stanza file that contains the information gathered by **rscan** as well as some default values that could be used for the node definitions.

To write the stanza format output of **rscan** to a file called "*mystanzafile"* run the following command.

rscan –z hmc01 > mystanzafile

This file can then be checked and modified as needed. For example you may need to add a different name for the node definition or add additional attributes and values.

Note: The stanza file will contain stanzas for things other than the LPARs. This information must also be defined in the xCAT database. It is not necessary to modify the non-LPAR stanzas in any way.

The updated stanza file might look something like the following.

```
Server-9117-MMA-SN10F6F3D:
    objtype=node
    nodetype=fsp
    id=5
    model=9118-575
    serial=02013EB
    hcp=hmc01
    pprofile=
    parent=Server-9458-10099201WM_A
    groups=fsp,all
    mgt=hmc

node01:
    objtype=node
    nodetype=lpar,osi
    id=9
    hcp=hmc01
    pprofile=lpar9
    parent=Server-9117-MMA-SN10F6F3D
    groups=all
    mgt=hmc

node02:
    objtype=node
    nodetype=lpar,osi
    id=7
    hcp=hmc01
    pprofile=lpar6
    parent=Server-9117-MMA-SN10F6F3D
    groups=all
    mgt=hmc
```

Note: The **rscan** command supports an option to automatically create node definitions in the xCAT database. To do this the LPAR name gathered by **rscan** is used as the node name and the command sets several default values. If you use the "-w" option make sure the LPAR name you defined will be the name you want used as your node name.

1.8 Define xCAT cluster nodes

The information gathered by the **rscan** command can be used to create xCAT node definitions.

Since we have put all the node information in a stanza file we can now pass the contents of the file to the **mkdef** command to add the definitions to the database.

cat mystanzafile | mkdef –z

You can use the xCAT **lsdef** command to check the definitions (ex. "*lsdef –l node01*"). After the node has been defined you can use the **chdef** command to make any additional updates to the definitions, if needed.

1.9 Define xCAT groups (optional)

There are two basic ways to create xCAT node groups. You can either set the "groups" attribute of the node definition or you can create a group directly.

You can set the "groups" attribute of the node definition when you are defining the node with the **mkdef** command or you can modify the attribute later using the **chdef** command. For example, if you want a set of nodes to be added to the group "aixnodes" you could run **chdef** as follows.

chdef –t node –o node01,node02,node03 groups=aixnodes

Note that this example will overwrite any value that is currently set. The chdef command also has an option for adding to or deleting from an existing attribute value.

The second option would be to create a new group definition directly using the **mkdef** command as follows.

mkdef –t group –o aixnodes members="node01,node02,node03"

These two options will result in exactly the same definitions and attribute values being created.

1.10 Set up customization scripts (optional)

xCAT supports the running of user-provided customization scripts on the nodes when they are deployed. For diskless nodes these scripts are run when the /etc/inittab file is processed during the node boot up.

To have your script run on the nodes:

1. Put a copy of your script in /install/postscripts on the xCAT management node. (Make sure it is executable.)

2. Set the "postscripts" attribute of the node or group definition to include the comma separated list of the scripts that you want to be executed on the nodes. For example, if you want to have your two scripts called "foo" and "bar" run on node "node01" you could use the chdef command as follows.

   chdef -t node -o node01 postscripts=foo,bar

   The order of the scripts in the list determines the order in which they will be run.

XCAT also runs some scripts to do default node configuration. You can see what scripts xCAT will run by looking at the "xcatdefaults" entry in the xCAT "postscripts" database table. (I.e. Run "tabdump postscripts".). You can change the default setting by using the xCAT **chtab** or **tabedit** command.

1.11 Create NIM client & group definitions

You can use the xCAT **xcat2nim** command to automatically create NIM machine and group definitions based on the information contained in the xCAT node and group definitions. By doing this you synchronize the NIM and xCAT names so that you can use the same target names when running either an xCAT or NIM command.

To create NIM machine definitions you could run the following command.

xcat2nim -t node –o aixnodes

To create NIM group definitions you could run the following command.

xcat2nim –t group –o aixnodes

To check the NIM definitions you could use the NIM **lsnim** command or the xCAT **xcat2nim** command. For example, the following command will display the NIM definitions of the nodes: *node01*, *node02*, and *node03* (from data stored in the NIM database).

xcat2nim -t node -l –o node01-node03

1.12 Initialize the AIX/NIM nodes

You can use the xCAT nimnodeset command to initialize the AIX standalone nodes. This command uses information from the xCAT database and default values to run the appropriate NIM commands.

For example, to set up all the nodes in the group "*aixnodes*" to install using the *osimage* named "*61gold*" you could issue the following command.

nimnodeset -i 53gold aixnodes

To verify that you have allocated all the NIM resources that you need you can run the "**lsnim –l**" command. For example, to check node "node01" you could run the following command.

lsnim -l node01

The command will also set the "profile" attribute in the xCAT node definitions to "61gold ".

To verify that NIM has allocated the required resources for a node and that the node is ready for a network boot you can run the "**lsnim –l**" command. For example, to check node "*node01*" you could run the following command.

lsnim -l node01

Note:

In preparation for the network boot, NIM configures **bootp**. Once the **nimnodeset** command completes you can verify that the */etc/bootptab* file has an entry for each node you wish to boot. Also, it is recommended that you stop and restart the **inetd** service to ensure the new **bootp** configuration is loaded:

stopsrc -s inetd

startsrc -s inetd

1.13 Initiate a network boot

Initiate a remote network boot request using the xCAT **rnetboot** command. For example, to initiate a network boot of all nodes in the group "*aixnodes*" you could issue the following command.

rnetboot aixnodes

Note: If you receive timeout errors from the **rnetboot** command, you may need to increase the default 60-second timeout to a larger value by setting ppctimeout in the site table:

chdef -t site -o clustersite ppctimeout=180

1.14 Verify the deployment

- As soon as the **rnetboot** command returns you can open a remote console to monitor the boot progress using the xCAT **rcons** command. This command requires that you have conserver installed and configured. To configure conserver, run:

  makeconservercf

  Kill the conserver daemon if it is running, and restart it:

  conserver &

(You may need to add /opt/freeware/bin and /opt/freeware/sbin to your PATH first).

To start a console:

rcons node01

- You can use the AIX **lsnim** command to see the state of the NIM installation for a particular node, by running the following command on the NIM master:

  lsnim -l <clientname>

- Retry and troubleshooting tips:
  - For p6 lpars, it may be helpful to bring up the HMC web interface in a browser and watch the lpar status and reference codes as the node boots.
  - Verify network connections
  - If the **rnetboot** returns "unsuccessful" for a node, verify that bootp and tftp is configured and running properly.
    - View /etc/bootptab to make sure an entry exists for the node.
    - Verify that the information in /tftpboot/<node>.info is correct.
    - Stop and restart inetd:
      stopsrc -s inetd
      startsrc -s inetd
    - Stop and restart tftp:
      stopsrc -s tftp
      startsrc -s tftp
    -
  - Verify NFS is running properly and mounts can be performed with this NFS server:
    - View /etc/exports for correct mount information.
    - Run the showmount and exportfs commands.
    - Stop and restart the NFS and related daemons:
      stopsrc -g nfs
      startsrc -g nfs
    - Attempt to mount a filesystem from another system on the network.
  - If the **rnetboot** operation is successful, but lsnim shows that the node is stuck at one of the netboot phases, you may need to redo your NIM definitions. Try the "short" approach first:

    nim -F -o reset node01

    nim -o dkls_init node01

    rnetboot -f node01
  - If that doesn't work, you may need to delete the entire client definition from NIM and recreate it:

    nim -F -o reset node01

nim -o deallocate -a root=root -a paging=paging -a dump=dump -a spot=61cosi node01

nim -o remove node01

mkdsklsnode -i 61cosi node01

rnetboot -f node01

2.0 Cleanup

The NIM definitions and resources that are created by xCAT commands are not automatically removed. It is therefore up to the system administrator to do some clean up of unused NIM definitions and resources from time to time. (The NIM lpp_source and SPOT resources are quite large.) There are xCAT commands that can be used to assist in this process.

2.1 Removing NIM resources

Use the xCAT **rmnimimage** command to remove all the NIM resources associated with a given xCAT *osimage* definition. The command will only remove a NIM resource if it is not allocated to a node. You should always clean up the NIM node definitions before attempting to remove the NIM resources. The command will also remove the xCAT *osimage* definition that is specified on the command line.

For example, to remove the "61spot" *osimage* definition along with all the associated NIM resources run the following command.

rmnimimage 53gold

If necessary, you can also remove the NIM definitions directly by using NIM commands. See the AIX/NIM documentation for details.

3.0 Notes

3.1 Terminology

image

The term "image" is used extensively in this document. The precise meaning of an "image" will vary depending on the context in which the term is being used. In general you can think of an image as the basic operating system image as well as other resources etc. that are needed to boot a node. In most cases in this document we will be referring to an image as either an xCAT *osimage* definition or an AIX/NIM diskless image, (COSI).

osimage - This is an xCAT object that can be used to describe an operation system image. This definition can contain various types of information depending on what will be installed on the node and how it will be installed. The image definition is not node specific and can be used to deploy multiple nodes. It contains all the information that

will be needed by the underlying xCAT and NIM support to deploy the node.

Notes for mksysb install:

This procedure was written for a NIM rte install. Here are some notes to help you if you would like to do a mksysb install instead. If you need to install a VIOS or IVM on a pSeries LPAR, you will need to do a mksysb install. Many of these steps can be easily performed using **smitty nim**.

- Obtain a mksysb image or create one following NIM instructions.
- Create a NIM mksysb resource
- Create a NIM spot definition
- Create NIM resource group with the spot, mksysb, resolv.conf, and bosinst_data resources.
- Run bosinst for a mksysb install:

  nim -o bos_inst -a source=mksysb -a boot_client=no -a group=<mksysb_resgrp> -a accept_licenses=yes <node>