

IBM Systems ^{MEDIA}

Using NIM with VIO Servers

AIX expert Jaqui Lynch takes users through the process step by step.



By Jaqui Lynch

09/27/2019

Network Install Manager (NIM) has been around for a long time, and it can be used to install, upgrade, backup and restore VIO servers as well as regular AIX LPARs. It normally takes several hours to completely install a VIO server and customize it, but with NIM, you can install the first VIO server and tailor it without creating all of the virtual devices. You can then take a backup, which becomes your golden image, that's then used to install all of the other VIO servers. This saves a significant amount of time and provides for consistent installations.

NIM Basics

Setting up a NIM environment requires a NIM master and a number of client instances. The NIM master owns and provides the resources necessary for the clients to be serviced. All NIM operations on clients will require one or more of these resources. The master stores information about NIM and its setup in its own NIM database, which needs to be backed up regularly. It stores the resources in ordinary AIX filesystems. These filesystems are made available to clients using NFS.

NIM depends on certain protocols and network services in order to function correctly. These include NFS, bootp or DHCP, and tftp. It also used to require the use of rsh and the other rcmd commands, but in AIX v5.3, it's now possible to use either basic nimsh or Openssl, either of which is more secure than the r-commands. These new services come as part of the bos.sysmgt.nim.client fileset, and the old rcmds are still supported in 5.3 for compatibility reasons.

Planning for a NIM Server

The primary NIM server (known as the NIM master) will need to be installed at the latest level of AIX. This will allow you to maintain LPARs at the same level or lower. It should be a standalone LPAR (no dependencies on VIO servers), and it shouldn't be combined with backup servers such as Tivoli Storage Manager. This allows you to update the NIM server without affecting production backup systems. The NIM LPAR will need CPU (.5 to 1 core), memory, a network adapter and a SAN adapter with disk or dedicated local disk.

My current NIM server has .5 of a core but can grow to 2 cores dynamically, 8GB of memory and dedicated adapters. It has 2x 100GB disks (one for rootvg and one for cloning to during upgrades), plus additional disks for two other volume groups. The first is NIMVG, which I give 300-500GB to, depending on how many systems I have to support and how

many different O/S levels I need to keep around. I have a second volume group called datavg. This has two filesystems: /software and /backups. /software is all of the software that I need to share with clients for installations. /backups is where all of the client LPAR mksysb images are written to. These two filesystems are shared with the client LPARs via NFS.

As of the date this article is being written, the latest VIO software is v3.1.0.21 and this requires the NIM server to be installed at AIX 7.2 tl3 sp3.

NIM Resources

For a VIO server, there are three critical NIM resources apart from the machine (LPAR) itself. These are the SPOT, MKSYSB and bosinst_data.

SPOT

The SPOT (Shared Product Object Tree) is a directory that's created from the VIO mksysb. The SPOT is used in a similar fashion to the boot images. For a VIO server, we create the spot from the mksysb image provided on the installation DVD.

mksysb

The NIM master uses the SPOT and a mksysb image to install a VIO instance. The initial image for a VIO server is provided on the installation DVD.

bosinst_data

This is a file, not a directory, and it contains the necessary information to allow the installation to take place without manual intervention. It's used to define defaults such as default disk drive, type of installation and so on. For a VIO server, we use the bosinst_data file provided on the installation DVD.

Using the NIM Master

Previous articles have discussed installing and setting up the NIM master. Assuming this has been done, you now need to tailor it for the VIO. In order to support VIO 3.1.0.21, the NIM master must be installed at or upgraded to AIX v7.2 tl3 sp3. Once that's done, you can add machine resources (these are the VIO LPARs). To add a machine, you need to

ensure its network name is resolvable (either DNS or add to `/etc/hosts` on the NIM master). We would add a machine resource for each VIO server we want to install or maintain.

You will also need to check that the NIM master is enabled for bootp and tftp in `/etc/inetd.conf`.

Setting up NIM to Work With VIO Servers

The next step is setting up the SPOT, `bosinst_data` and `mksysb` resources needed for the VIO servers.

First, extract the `mksysb` resource from the DVD or ISO. I download the 3.1.0.21 flash image from the IBM-entitled software website and use that. It comes as a single ISO. If you use the set of two ISO images. They come at a lower level (3.1.0.10) and you have to extract 3 `mksysb` images and combine them. It's much easier to use the flash ISO.

Download the flash ISO to your NIM server and then `loopmount` it so you can copy the `mksysb` and other resources from it. As a point of reference, I put all of my software to be used, including ISO images, into `/software`, and I create a mount point of `/cdrom` to be used for `loopmounts`.

The line below mounts the 3.1.0.21 flash image that's stored in `/software/powervm31` onto `/cdrom`. For the flash image, you must use `udfs` instead of `cdrfs` to mount it.

Now we copy the resources we will need to create the resources for the VIO.

NIM Commands to build initial resources for VIO

Next, we set up the `mksysb`, SPOT and `bosinst_data` resources using the images we just copied above. You can do all of these using `smitty nim`, but I prefer to use line commands.

Create VIO `mksysb` resource using `smitty - mksysb_vios31021` from `/nim/images/vios31021-flash-mksysb_image` or the command

below:

Create the SPOT from the mksysb:

Create the bosinst_data:

You can now check the spot resource.

It should come back with a blank line.

And you can then list each of the three resources you just created.

mksysb_vios31021:

```
class      = resources
type       = mksysb
creation_date = Thu Aug 22 14:42:34 2019
Rstate     = ready for use
prev_state = unavailable for use
location   = /nim/images/vios31021-flash-mksysb_image
version    = 7
release    = 2
mod        = 3
oslevel_r  = 7200-03
oslevel_s  = 7200-03-03-1914
alloc_count = 0
```

```
server    = master
extracted_spot = spotvios31021
```

lsnim -l viobosinstdata

viobosinstdata:

```
class    = resources
type     = bosinst_data
Rstate   = ready for use
prev_state = unavailable for use
location = /nim/bosinst_data/viobosinst.data.txt
alloc_count = 0
server   = master
```

lsnim -l spotvios31021

spotvios31021:

```
class    = resources
type     = spot
plat_defined = chrp
arch     = power
Rstate   = ready for use
prev_state = verification is being performed
location = /nim/spot/spotvios31021/usr
version  = 7
release  = 2
```

```
mod      = 3
oslevel_r = 7200-03
oslevel_s = 7200-03-03-1914
alloc_count = 0
server    = master
if_supported = chrp.64 ent
Rstate_result = success
mkysb_source = mkysb_vios31021
```

You're now ready to use these resources to install your VIO server.

Installing the VIO

You need to ensure the network ports for the VIO aren't aggregated yet. NIM expects to install over a regular access (non-aggregated) port to the VIO server. This also applies if you're using the HMC to install the VIO image.

First, you define the VIO profile on the HMC including network ports, vlan, etc. Then make sure the disk is zoned at the switch and mapped at the storage. I usually give my VIO server 3x 100GB SAN disks (one for rootvg and one for cloning for updates. The third one comes in if I choose to use FBO (file backed optical) to load ISO images for client LPARs.

Allocate the SPOT, bosinst and mkysb to the VIO LPAR (assume vio1).

```
nim -o allocate -a bosinst_data=viobosinstdata  vio1
```

```
nim -o allocate -a mkysb=mkysb_vios31021  vio1
```

```
nim -o allocate -a spot=spotvios31021  vio1
```

Now you can set up the VIO to do a NIM mkysb install. I usually use smitty nim to do this as it lets me also choose to tell the VIO to not remain a NIM client, but the command will be something like:

```
nim -o bos_inst -a source=mkysb -a boot_client=no -a accept_licenses=yes vio1
```

Check the exports, etc.

```
showmount -e
```

This should show the SPOT, bosinst_data and mkysyb all exported to the VIO.

```
ls -l /tftpboot
```

This should show files needed for the installs.

```
tail /etc/bootptab
```

This should show a line for the VIO that includes its IP address and subnet mask and gateway information.

At this point, you're ready to do the install:

- Boot VIO1 in SMS mode and follow prompts to install from network. Make sure you know which adapter to install over.
- At prompt, type 1 and hit enter to use as console
- Then: sits on 0c44/0c46 for a while; when prompted, type 1 and press enter for English
- 2 – change show settings
- Selected the correct hdisk—in my case it was hdisk0 (300GB internal disk) 0 to install

Once the VIO server is up, you'll need to set the password, reply a to accept the license and then "*license -accept.*"

After that, it's time to do normal customizations such as fixing page space, setting up SEAs, etc.

A second option for the install is to install (or restore) from a backup you have taken from a current VIO server. This involves using an ios_mkysyb. Using the mkysyb taken from the VIO server, first copy it into /nim/images then define it as an ios_mkysyb resource:

You can now use that image, along with the SPOT and bosinst_data resources to do a mkysyb restore or install.

An Eye on Maintenance

When performing maintenance with updateios on a VIO server, I always take a clone first. That way, I have an easy failback. This is done using `alt_disk_copy`. On my VIO system, I have the following disks:

```
hdisk0  rootvg
```

```
hdisk1  not used
```

```
hdisk2  fbovg
```

In addition, I have all of my disks for `nimvg` and `datavg`

`hdisk1` is the disk I clone to prior to any updates. To create the clone to `hdisk1`:

```
alt_disk_copy -B -d hdisk1
```

The `-B` says don't update the bootlist but you should still check it:

```
bootlist -m normal -o
```

Once the copy is done, you can now perform your updates either directly on the VIO or using NIM. To perform updates using NIM, you will use the `nim -o updateios` command. Options include `install`, `commit`, `reject`, `cleanup` and `remove`. You will need to create `install_bundles` and other resources prior to using NIM `updateios`.

Upgrading to VIO Server v3 from v2

This is an upgrade, not an update. To upgrade to v3, you need to ensure your VIO is installed at 2.2.6.32 or higher (if on v2). The v3 upgrade is done using `viosupgrade` on the VIO or `nim -o viosupgrade`. Be aware that the flags are different on both. It will install to an alternate disk which must be unused. You can check the disks as `padmin` using:

If `hdisk1` (our target) has a volume group such as `altinst_rootvg` on it or doesn't show as free, then you need to either choose another disk that's free or clear `hdisk0`. Either way, `altinst_rootvg` and `old_rootvg` cannot exist prior to the upgrade.

To free the disk, export the volume group assuming it's not critical, then:

The above removes ownership. The disk should now show as free with `lspv -free`.

The NIM version of `viosupgrade` backs up the virtual and logical configuration data, installs the specified image and restores the virtual and logical configuration data of the Virtual I/O Server (VIOS). If you want to save some files, there are specific steps in the `viosupgrade` document on how to do this. There are also specific steps to follow if you're using shared storage pools (SSPs). For a non-SSP environment, the `viosupgrade` command will be something like:

After the upgrade completes, the VIO will reboot from the alternate disk and should restore all of the virtual and logical definitions. You can then do your customizations as this is a complete reinstall.

Backups and Restores

It's critical to back up your VIO servers correctly. I normally NFS export `/backups` from my NIM server to all my LPARs, including VIO servers. I then use cron to take backups. The script I use for my VIO servers is in the references. It takes two different backups using `backupios`. The first is a backup that can be used to restore from the HMC or from an NFS server (creates a tar file) and the second is a standard `ios_mksysb` format. The restore is the same as the install above, but uses the `mksysb` image just created. That image will need to be copied into `/nim/images` and then a NIM resource will need to be created. Having both backups provides you with maximum flexibility for restores.

Summary

The ability to support VIO servers using NIM saves a great deal of time for administrators as well as proving a more consistent method for installations. It also provides a backup and restore strategy for the `mksysb` images that many backup systems don't provide for. With all of the VIO servers backed up to the NIM server, you should be sure to back up the NIM server along with its backup and NIM volume groups so they can be taken offsite or sent to the disaster recovery location. This provides a single point for restores in a disaster recovery or other recovery situation.

References

1. <http://www.circle4.com/movies/>

2. Handout from pTechu May 2017 on NIM 101

<http://www.circle4.com/ptechu/nim101-may2317.pdf> (<http://www.circle4.com/ptechu/nim101-may2317.pdf>)

3. Handout from pTechu October 2019 on NIM 201

<http://www.circle4.com/ptechu/nim201-Sep112019.pdf> (<http://www.circle4.com/ptechu/nim201-Sep112019.pdf>)

4. Supported methods of cloning an operating systems instance

<http://www-01.ibm.com/support/docview.wss?uid=isg3T1012273> (<http://www-01.ibm.com/support/docview.wss?uid=isg3T1012273>)

5. NIM installation using VIO server

<http://www-01.ibm.com/support/docview.wss?uid=isg3T1011386> (<http://www-01.ibm.com/support/docview.wss?uid=isg3T1011386>)

6. NIM install and backup of VIO servers

<https://www.ibm.com/support/pages/nim-installation-and-backup-vio-server-using-traditional-nim-methods> (<https://www.ibm.com/support/pages/nim-installation-and-backup-vio-server-using-traditional-nim-methods>)

<http://www-01.ibm.com/support/docview.wss?uid=isg3T1011386> (<http://www-01.ibm.com/support/docview.wss?uid=isg3T1011386>)

7. Configuring NIM

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.install/nim_basic.htm (https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.install/nim_basic.htm)

8. NIM updateios

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/install/nim_op_updateios.html (https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/install/nim_op_updateios.html)

9. Using NIM version of viosupgrade

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/v_commands/viosupgrade.html (https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/v_commands/viosupgrade.html)

10. alt_disk_copy command

https://www.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.cmds1/alt_disk_copy.htm (https://www.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.cmds1/alt_disk_copy.htm)

11. Troubleshooting NIM LED hangs

<http://www-01.ibm.com/support/docview.wss?uid=isg3T1012561> (<http://www-01.ibm.com/support/docview.wss?uid=isg3T1012561>)

12. Alternate disk install

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_71/install/alternate_disk_installation.html (https://www.ibm.com/support/knowledgecenter/en/ssw_aix_71/install/alternate_disk_installation.html)

13. NIM from A-Z

<http://www.redbooks.ibm.com/redbooks/pdfs/sg247296.pdf> (<http://www.redbooks.ibm.com/redbooks/pdfs/sg247296.pdf>)

14. NIM Basics

<https://aixexpert.wordpress.com/nim/nim-basics/> (<https://aixexpert.wordpress.com/nim/nim-basics/>)

Appendix A

Sample VIOBackup Script to Take a mkysb to the NIM Server

About the author

Jaqui Lynch has over 38 years of experience working with a projects and OSes across vendor platforms, including IBM Z, UNIX systems and more.

Related Content

[Systems management \(/systems-management\) Configuring MQ on iSeries →](#) (<https://ibmsystemsmag.com/Power-Systems/10/2002/mq-configuration-middleware>)

[Application development \(/application-development\) Service Programs and Signatures →](#) (<https://ibmsystemsmag.com/Power-Systems/10/2003/service-programs-signatures>)

[Systems management \(/systems-management\) A Look at File Systems →](#) (<https://ibmsystemsmag.com/Power-Systems/09/2004/file-systems-commands>)

IBM ^{MEDIA} Systems

IBM Systems magazine is a trademark of International Business Machines Corporation. The editorial content of IBM Systems magazine is placed on this website by MSP TechMedia under license from International Business Machines Corporation.

© 2020 Key Enterprises LLC. All rights reserved