# Install and configure a GPFS cluster on AIX

0 people like this|Updated 5/12/14 by [gcorneau](https://www.ibm.com/developerworks/community/profiles/html/profileView.do?userid=0100003X4E)|Tags: None

**Note: This guide is for GPFS V3, not GPFS V4! - May 2014**

### Objectives

* Verify the system environment
* Create a GPFS cluster
* Define NSD's
* Create a GPFS file system

### You will need

|  |  |  |
| --- | --- | --- |
|  | Requirements for this lab (not necessarily GPFS minimum requirements):   * Two AIX 6.1 or 7.1 operating systems (LPARs)   + Very similar to Linux installation. AIX LPP packages replace the Linux RPMs, some of the administrative commands are different. * At least 4 hdisks * GPFS 3.4 Software with latest PTFs |  |

### Step 1: Verify Environment

|  |  |  |
| --- | --- | --- |
|  | 1. Verify nodes properly installed    1. Check that the operating system level is supported    2. On the system run *oslevel*    3. Check the GPFS FAQ:[*http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsclustersfaq.html\_\_*](http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsclustersfaq.html__)    4. Is the installed OS level supported by GPFS? Yes No    5. Is there a specific GPFS patch level required for the installed OS? Yes No    6. If so what patch level is required? \_\_\_\_\_\_\_\_\_\_\_ 2. Verify nodes configured properly on the network(s)    1. Write the name of Node1: \_\_\_\_\_\_\_\_\_\_\_\_    2. Write the name of Node2: \_\_\_\_\_\_\_\_\_\_\_\_    3. From node 1 ping node 2    4. From node 2 ping node 1    5. If the pings fail, resolve the issue before continuing. 3. Verify node-to-node ssh communications (For this lab you will use ssh and scp for secure remote commands/copy)    1. On each node create an ssh-key. To do this use the command ssh-keygen; if you don't specify a blank passphrase, -N, then you need to press enter each time you are promoted to create a key with no passphrase until you are returned to a prompt. The result should look something like this:    2. # ssh-keygen -t rsa -N "" -f $HOME/.ssh/id\_rsa Generating public/private rsa key pair. Created directory '/.ssh'. Your identification has been saved in /.ssh/id\_rsa. Your public key has been saved in /.ssh/id\_rsa.pub. The key fingerprint is: 7d:06:95:45:9d:7b:7a:6c:64:48:70:2d:cb:78:ed:61 root@node1    3. On node1 copy the $HOME/.ssh/id\_rsa.pub file to $HOME/.ssh/authorized\_keys    4. # cp $HOME/.ssh/id\_rsa.pub $HOME/.ssh/authorized\_keys    5. From node1 copy the $HOME/.ssh/id\_rsa.pub file from node2 to /tmp/id\_rsa.pub    6. # scp node2:/.ssh/id\_rsa.pub /tmp/id\_rsa.pub    7. Add the public key from node2 to the authorized\_keys file on node1    8. # cat /tmp/id\_rsa.pub >> $HOME/.ssh/authorized\_keys    9. Copy the authorized key file from node1 to node2    10. # scp $HOME/.ssh/authorized\_keys node2:/.ssh/authorized\_keys    11. To test your ssh configuration ssh as root from node 1 to node1 and node1 to node2 until you are no longer prompted for a password or for addition to the known\_hosts file.    12. node1# ssh node1 date    13. node1# ssh node2 date    14. node2# ssh node1 date    15. node2# ssh node2 date    16. Supress ssh banners by creating a .hushlogin file in the root home directory    17. # touch $HOME/.hushlogin 4. Verify the disks are available to the system 5. For this lab you should have 4 disks available for use *hdiskw-hdiskz*.    1. Use *lspv* to verify the disks exist    2. Ensure you see 4 unused disks besides the existing rootvg disks and/or other volume groups. |  |

### Step 2: Install the GPFS software

|  |  |  |
| --- | --- | --- |
|  |  |  |

#### **On node1**

|  |  |  |
| --- | --- | --- |
|  | 1. Locate the GPFS software in /yourdir/gpfs/base/ 2. # cd /yourdir/gpfs/base/ 3. Run the *inutoc* command to create the table of contents, if not done already 4. # inutoc . 5. Install the base GPFS code using the installp command 6. # installp -aXY -d/yourdir/gpfs/base all 7. Locate the latest GPFS updates in /yourdir/gpfs/fixes/ 8. # cd /yourdir/gpfs/fixes/ 9. Run the *inutoc* command to create the table of contents, if not done already 10. # inutoc . 11. Install the GPFS PTF updates using the installp command 12. # installp -aXY -d/yourdir/gpfs/fixes all 13. Repeat Steps 1-7 on node2. On node1 and node2 confirm GPFS is installed using the *lslpp* command 14. # lslpp -L gpfs.\\* 15. the output should look similar to this 16. Fileset Level State Type Description (Uninstaller) ---------------------------------------------------------------------------- gpfs.base 3.4.0.11 A F GPFS File Manager gpfs.docs.data 3.4.0.4 A F GPFS Server Manpages and Documentation gpfs.gnr 3.4.0.2 A F GPFS Native RAID gpfs.msg.en\_US 3.4.0.11 A F GPFS Server Messages U.S. English 17. Note: Exact versions of GPFS may vary from this example, the important part is that the base, docs and msg filesets are present. 18. Note2: The gpfs.gnr fileset is used by the [Power 775 HPC cluster only](http://www-03.ibm.com/systems/power/hardware/775/). 19. Confirm the GPFS binaries are in your $PATH using the *mmlscluster* command 20. # mmlscluster mmlscluster: This node does not belong to a GPFS cluster. mmlscluster: Command failed. Examine previous error messages to determine cause. 21. Note: The path to the GPFS binaries is: /usr/lpp/mmfs/bin |  |

### Step 3: Create the GPFS cluster

|  |  |  |
| --- | --- | --- |
|  | For this exercise the cluster is initially created with a single node. When creating the cluster make node1 the primary configuration server and give node1 the designations quorum and manager. Use ssh and scp as the remote shell and remote file copy commands.  \*Primary Configuration server (node1): \_\_\_\_\_\_\_\_*\_*\_  \*Verify fully qualified path to ssh and scp: ssh path\_*\_*\_\_\_\_\_\_\_\_  scp path\_\_\_\_\_\_\_\_\_\_\_\_\_   1. Use the *mmcrcluster* command to create the cluster 2. # mmcrcluster -N node1:manager-quorum -p node1 -r /usr/bin/ssh -R /usr/bin/scp Thu Mar 1 09:04:33 CST 2012: mmcrcluster: Processing node node1 mmcrcluster: Command successfully completed mmcrcluster: Warning: Not all nodes have proper GPFS license designations.  Use the mmchlicense command to designate licenses as needed. 3. Run the *mmlscluster* command again to see that the cluster was created 4. # mmlscluster  =============================================================================== | Warning: | | This cluster contains nodes that do not have a proper GPFS license | | designation. This violates the terms of the GPFS licensing agreement. | | Use the mmchlicense command and assign the appropriate GPFS licenses | | to each of the nodes in the cluster. For more information about GPFS | | license designation, see the Concepts, Planning, and Installation Guide. | ===============================================================================  GPFS cluster information ========================   GPFS cluster name: node1.ibm.com  GPFS cluster id: 13882390374179224464  GPFS UID domain: node1.ibm.com  Remote shell command: /usr/bin/ssh  Remote file copy command: /usr/bin/scp  GPFS cluster configuration servers: -----------------------------------   Primary server: node1.ibm.com  Secondary server: (none)  Node Daemon node name IP address Admin node name Designation -----------------------------------------------------------------------------------------------  1 node1.lab.ibm.com 10.0.0.1 node1.ibm.com quorum-manager 6. Set the license mode for the node using the *mmchlicense* command. Use a server license for this node. 7. # mmchlicense server --accept -N node1  The following nodes will be designated as possessing GPFS server licenses:  node1.ibm.com mmchlicense: Command successfully completed |  |

### Step 4: Start GPFS and verify the status of all nodes

|  |  |  |
| --- | --- | --- |
|  | 1. Start GPFS on all the nodes in the GPFS cluster using the *mmstartup* command 2. # mmstartup -a 3. Check the status of the cluster using the *mmgetstate* command 4. # mmgetstate -a  Node number Node name GPFS state ------------------------------------------  1 node1 active |  |

### Step 5: Add the second node to the cluster

|  |  |  |
| --- | --- | --- |
|  | 1. One node 1 use the *mmaddnode* command to add node2 to the cluster 2. # mmaddnode -N node2 3. Confirm the node was added to the cluster using the *mmlscluster* command 4. # mmlscluster 5. Use the *mmchcluster* command to set node2 as the secondary configuration server 6. # mmchcluster -s node2 7. Set the license mode for the node using the *mmchlicense* command. Use a server license for this node. 8. # mmchlicense server --accept -N node2 9. Start node2 using the *mmstartup* command 10. # mmstartup -N node2 11. Use the *mmgetstate* command to verify that both nodes are in the active state 12. # mmgetstate -a |  |

### Step 6: Collect information about the cluster

|  |  |  |
| --- | --- | --- |
|  | Now we will take a moment to check a few things about the cluster. Examine the cluster configuration using the *mmlscluster* command   1. What is the cluster name? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. What is the IP address of *node2*? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3. What date was this version of GPFS "Built"? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4. Hint: look in the GPFS log file: /var/adm/ras/mmfs.log.latest |  |

### Step 7: Create NSDs

|  |  |  |
| --- | --- | --- |
|  | You will use the 4 *hdisks*.   * Each disk will store both data and metadata * The storage pool column blank (not assigning storage pools at this time) * The NSD server field (ServerList) is left blank (both nodes have direct access to the shared LUNs)  1. On node1 create the directory /yourdir/data 2. Create a disk descriptor file */yourdir/data/diskdesc.txt* using the format: 3. #DiskName:ServerList::DiskUsage:FailureGroup:DesiredName:StoragePool hdiskw:::dataAndMetadata::nsd1: hdiskx:::dataAndMetadata::nsd2: hdisky:::dataAndMetadata::nsd3: hdiskz:::dataAndMetadata::nsd4: 4. Note: hdisk numbers will vary per system. 5. Create a backup copy of the disk descriptor file */yourdir/data/diskdesc\_bak.txt* 6. # cp /yourdir/data/diskdesc.txt /yourdir/data/diskdesc\_bak.txt 7. Create the NSD's using the mmcrnsd command 8. # mmcrnsd -F /yourdir/data/diskdesc.txt |  |

### Step 8: Collect information about the NSD's

|  |  |  |
| --- | --- | --- |
|  | Now collect some information about the NSD's you have created.   1. Examine the NSD configuration using the *mmlsnsd* command    1. What *mmlsnsd* flag do you use to see the operating system device (/dev/hdisk?) associated with an NSD? \_\_\_\_\_\_\_ |  |

### Step 9: Create a file system

|  |  |
| --- | --- |
|  | Now that there is a GPFS cluster and some NSDs available you can create a file system. In this section we will create a file system.   * Set the file system blocksize to 64kb * Mount the file system at /gpfs  1. Create the file system using the *mmcrfs* command 2. # mmcrfs /gpfs fs1 -F diskdesc.txt -B 64k 3. Verify the file system was created correctly using the *mmlsfs* command 4. # mmlsfs fs1 5. Is the file system automatically mounted when GPFS starts? \_\_\_\_\_\_\_*\_*\_\_\_\_\_\_\_\_\_ 6. Mount the file system using the \_mmmount\_ command 7. # mmmount all -a 8. Verify the file system is mounted using the *df* command 9. # df -k Filesystem 1024-blocks Free %Used Iused %Iused Mounted on /dev/hd4 65536 6508 91% 3375 64% / /dev/hd2 1769472 465416 74% 35508 24% /usr /dev/hd9var 131072 75660 43% 620 4% /var /dev/hd3 196608 192864 2% 37 1% /tmp /dev/hd1 65536 65144 1% 13 1% /home /proc - - - - - /proc /dev/hd10opt 327680 47572 86% 7766 41% /opt /dev/fs1 398929107 398929000 1% 1 1% /gpfs 10. Use the *mmdf* command to get information on the file system. 11. # mmdf fs1 12. How many inodes are currently used in the file system? \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |