BLOG O' MATTY

SUBSCRIBE VIA RSS SUBSCRIBE VIA E-MAIL

HOME BLOG HOME ARTICLES CODE PRESENTATIONS

Deploying highly available zones with Solaris Cluster 3.2

I discussed my first impressions of Solaris Cluster 3.2 a while back, and have been using it in various capacities ever since. One thing that I really like about Solaris Cluster is its ability to manage resources running in zones, and fail these resources over to other zones running on the same host, or a zone running on a secondary host. Additionally, Solaris Cluster allows you to migrate zones between nodes, which can be quite handy when resources are tied to a zone and can't be managed as a scalable services. Configuring zone failover is a piece of cake, and I will describe how to do it in this blog post.

To get failover working, you will first need to download and install Solaris Cluster 3.2. You can grab the binaries from sun.com, and you can install them using the installer script that comes in the zip file:

```
$ unzip suncluster_3_2u2-ga-solaris-x86.zip


$ cd Solaris_x86


$ ./installer
```

Once you run through the installer, the binaries should be placed in /usr/cluster, and you should be ready to configure the cluster. Prior to doing so, you should add something similar to the following to /etc/profile to make life easier for cluster administrators:

```
PATH=/bin:/usr/bin:/sbin:/usr/sbin:/usr/sfw/bin:/usr/cluster/bin
export PATH

TERM=vt100
export TERM
```

Also, if you selected the disabled remote services option during the Solaris install, you should reconfigure the rpc/bind service to allow connections from other cluster members (Solaris Cluster uses RPC extensively for cluster communications). This can be accomplished with the svccfg utility:

```
$ svccfg
svc:> select rpc/bind
svc:/network/rpc/bind> setprop config/local_only=false
```

Once the properties are adjusted, you can refresh the rpc/bind service to get these properties to go into effect:

```
$ svcadm refresh rpc/bind
```

Now that the environment is set up, you can run scinstall on each node to configure the cluster. I personally like to configure the first node and then add the second node to the cluster (this requires you to run scinstall on node one, then again on node two), but you can configured everything in one scinstall run if you prefer. Once scinstall completes and the nodes reboot, you should be able to run the cluster command to see if the cluster is operational:

```
$ cluster status

=== Cluster Nodes ===

--- Node Status ---

Node Name                                       Status
---------                                       ------
snode1                                          Online
snode2                                          Online

=== Cluster Transport Paths ===

Endpoint1               Endpoint2               Status
---------               ---------               ------
snode1:e1000g2          snode2:e1000g2          Path online
snode1:e1000g1          snode2:e1000g1          Path online

=== Cluster Quorum ===

--- Quorum Votes Summary ---

            Needed   Present   Possible
            ------   -------   --------
```

Recommended Books

Search

Search for:

[Search]

Blogroll

- Adam Leventhal
- Ben Rockwood
- Brendan Gregg
- Chris Haley
- Chris Siebenmann
- Clay McClure
- Derek Crudgington
- Gerardo López-Fernández
- OpenBSD Journal
- Planet CentOS
- Planet FreeBSD
- Planet Solaris
- Planet SysAdmin
- Scott Cromar
- Scott Dickson

Categories

- Apache (29)
- Apple (26)
- Articles, Presentations and Certifications (39)
- Brocade (8)
- cfengine (5)

```
            1        1         1

--- Quorum Votes by Node ---

Node Name       Present       Possible       Status
---------       -------       --------       ------
snode1          1             1              Online
snode2          0             0              Online

=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name     Primary     Secondary     Status
-----------------     -------     ---------     ------

--- Spare, Inactive, and In Transition Nodes ---

Device Group Name   Spare Nodes   Inactive Nodes   In Transistion Nodes
-----------------   -----------   --------------   --------------------

--- Multi-owner Device Group Status ---

Device Group Name           Node Name           Status
-----------------           ---------           ------

=== Cluster Resource Groups ===

Group Name       Node Name       Suspended      State
----------       ---------       ---------      -----

=== Cluster Resources ===

Resource Name       Node Name       State       Status Message
-------------       ---------       -----       --------------

=== Cluster DID Devices ===

Device Instance             Node                Status
---------------             ----                ------
/dev/did/rdsk/d1            snode1              Ok
                            snode2              Ok

/dev/did/rdsk/d3            snode1              Ok

/dev/did/rdsk/d5            snode2              Ok

=== Zone Clusters ===

--- Zone Cluster Status ---

Name    Node Name    Zone HostName    Status    Zone Status
----    ---------    -------------    ------    -----------
```

In the cluster status output above, we can see that we have a 2-node cluster that contains the hosts named snode1 and snode2. If there are no errors in the status output, you can register the HAStoragePlus resource type (this manages disk storage, and allows volumes and pools to failover between node) with the cluster. This can be accomplished with the clresourcetype command:

```
$ clresourcetype register SUNW.HAStoragePlus
```

Next you will need to create a resource group, which will contain all of the zone resources:

```
$ clresourcegroup create hazone-rg
```

Once the resource group is created, you will need to add a HAStoragePlus resource to manage the UFS file system(s) or ZFS pool your zone lives on. In the example below, a ZFS pool named hazone-pool resource is added to manage the ZFS pool named hazonepool:

```
$ clresource create -t SUNW.HAStoragePlus -g hazone-rg -p Zpools=hazonepool -p AffinityOn=True
hazone-zpool
```

After the storage is configured, you will need to update DNS or /etc/hosts with the name and IP address that you plan to assign to the highly available zone (this is the hostname that hosts will use to access services in the highly available zone). For simplicity purposes, I added an entry to /etc/hosts on each node:

```
# Add a hazone-lh entry to DNS or /etc/hosts
192.168.1.23 hazone-lh
```

Next you will need to create a logical hostname resource. This resource type is used to manage interface failover, which allows one or more IP addresses to float between cluster nodes. To create a logical hostname resource, you can use the clreslogicalhostname utility:

```
$ clreslogicalhostname create -g hazone-rg -h hazone-lh hazone-lh
```

Now that the storage and logical hostname resources are configured, you can bring the resource group that

Debian Linux (1)
Djbdns (1)
DNS & BIND (14)
Emulex (3)
Firefox (3)
Food and Beverages (1)
FreeBSD Misc (2)
FreeBSD Networking (1)
FreeBSD Utilities (2)
FreeNAS (1)
Gadgets (1)
Games (1)
Gnome (4)
GNU Privacy Guard (4)
Hardware (11)
Health Foods (2)
Home Security (1)
HTTP (4)
Investing (2)
ISC DHCP (3)
Java (12)
Libvirt (1)
Links (15)
Linux Debugging (10)
Linux Installation (2)
Linux iSCSI (2)
Linux Kernel (18)
Linux Kickstart (2)
Linux KVM (9)
Linux LVM (2)
Linux LXC Containers (1)
Linux Misc (11)
Linux Networking (21)
Linux NFS (1)
Linux Package Management (23)
Linux performance (5)
Linux Recovery (4)
Linux Security (15)
Linux SELinux (2)
Linux Storage (28)
Linux Utilities (92)
Linux Virtualization (2)
Linux X Windows (3)
Microsoft DNS (1)
Monitoring (5)
Music (38)
MySQL (3)
net-snmp (4)
Networking (4)
NFSv3 (5)
OpenBSD Kernel (7)
OpenBSD Misc (1)
OpenBSD Networking (2)
OpenBSD Ports (3)
OpenBSD Security (7)
OpenBSD Utilities (7)
OpenFiler (1)
OpenLDAP (4)
OpenSSH (11)

contains these resources online:

```
$ clresourcegroup online -M hazone-rg
```

If the cluster, clresourcegroup and clresource status commands list everything in an online state, we can create a zone with the zonecfg and zoneadm commands. The zone needs to be installed on each node so the zone gets put into the installed state, and the Sun documentation recommends removing the installed zone on the first node prior to installing it on the second node. This will work, though I think you can play with the index file to simplify this process (this is unsupported though). Once the zones are installed, you should failover the shared storage to each node in the cluster, and boot the zones to be extra certain. If this works correctly, then you are ready to register the SUNW.gds resource type:

```
$ clresourcetype register SUNW.gds
```

The SUNW.gds resource type provides the cluster hooks to bring the zone online, and will optionally start one or more services in a zone. To configure the resource type, you will need to create a configuration file that describes the resources used by the zone, the resource group the resources are part of, and the logical hostname to use with the zone. Here is an example configuration file I used to create my highly available zone:

```
$ cat /etc/zones/sczbt_config
# The resource group that contains the resources the zones depend on
RG=hazone-rg
# The name of the zone resource to create
RS=hazone-zone
# The directory where this configuration file is stored
PARAMETERDIR=/etc/zones
SC_NETWORK=true
# Name of the logical hostname resource
SC_LH=hazone-lh
# Name of the zone you passed to zonecfg -z
Zonename=hazone
Zonebrand=native
Zonebootopt=”"
Milestone=”multi-user-server”
FAILOVER=true
# ZFS pool that contains the zone
HAS_RS=hazone-zpool
```

The Solaris Cluster highly available guide for containers describes each of these parameters, so I won’t go into detail on the individual options. To tell the cluster framework that it will be managing the zone, you can execute the sczbt_register command passing it the configuration file you created as an argument:

```
$ cd /opt/SUNWsczone/sczbt/util

$ ./sczbt_register -f /etc/zones/sczbt_config
```

Once the zone is tied into the cluster framework, you can bring the zone resource group (and the zone) online with the clresourcegroup command:

```
$ clresourcegroup online -n snode2 hazone-rg
```

If the zone came online (which it should if everything was executed above), you should see the following:

```
$ clresourcegroup status

=== Cluster Resource Groups ===

Group Name       Node Name       Suspended      Status
----------       ---------       ---------      ------
hazone-rg        snode1          No             Offline
                 snode2          No             Online
```

```
$ zoneadm list -vc

  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   1 hazone           running    /hazonepool/zones/hazone       native   shared
```

```
$ zlogin hazone zonename
hazone
```

OpenSSL (8)
Oracle (3)
OS X (1)
Perl (11)
pets (1)
Postfix (1)
Procmail (1)
Puppet (1)
Python (5)
QLogic (1)
Rants (26)
Ruby (1)
Samba (2)
Secure Shell (2)
Security (9)
Sendmail (3)
Service providers (8)
Shell (9)
Smart Phones (2)
Smartmontools (7)
Snort (1)
Software (18)
Solaris Cluster (7)
Solaris Debugging (17)
Solaris DTrace (32)
Solaris Fault Management (7)
Solaris Install (3)
Solaris Kernel (7)
Solaris Linker (6)
Solaris Live Upgrade (2)
Solaris Misc (22)
Solaris Networking (27)
Solaris NFS (4)
Solaris Package Management (1)
Solaris Patching (20)
Solaris Performance Tuning (2)
Solaris Recovery (2)
Solaris Resource Management (2)
Solaris Security (13)
Solaris Shell (1)
Solaris SMF (9)
Solaris Storage (40)
Solaris Utilities (66)
Solaris Volume Manager (7)
Solaris Xen (3)
Solaris ZFS (28)
Solaris Zones (18)
Storage (2)
Storage Area Networking (9)
Sun Directory Server (3)
Sun Servers (3)
Sun Web Server (1)
syslog-ng (2)
Truecrypt (1)
Uncategorized (29)
UNIX (3)
UNIX Shell (33)
Veritas Cluster Server (1)
Veritas File System (6)

If you have services that you want to start and stop when you bring your zone online, you can use SMF or the ServiceStartCommand, ServiceStopCommand and ServiceProbeCommand SUNW.gds configuration options. Here are a couple of sample entries that could be added to the configuration file listed above:

```
ServiceStartCommand=”/usr/local/bin/start-customapp”
ServiceStopCommand=”/usr/local/bin/stop-customapp”
ServiceProbeCommand=”/usr/local/bin/probe-customapp”
```

As the names indicate, ServiceStartCommand contains the command to run to start the service, ServiceStopCommand contains the command to run to stop the service, and ServiceProbeCommand contains the command to run to verify the service is up and operational. This is super useful stuff, and it’s awesome that zones will now failover to a secondary node when a server fails, or when a critical error occurs and a zone is unable to run.

Related Posts

- New zone features in Solaris 10 update 9
- Using cfgadm to display LUNs on Solaris hosts

matty on April 10, 2009 | Filed Under Solaris Cluster

9 Comments

Chris on April 28th, 2009

Besides setting up a failover filesystem how would you setup a global file system to be mounted on all nodes in the cluster?
So for example you have node1, node2
each node is running zone’s…
now each zone I would want to mount the same SAN lun… (granted only one zone will be running application at a time so should hopefully avoid clobbering writes)

Thanks

Sebastian on May 5th, 2009

Unfortunately patching a cluster with failover zones installed is not a piece of cake…

Kartik on November 18th, 2010

In the command: clresource create -t SUNW.HAStoragePlus -g hazone-rg -p Zpools=hazonepool -p AffinityOn=True hazone-zpool

is Zpools a predefined/standard property?
Is it on shared storage? Is hazonepool local to cluster node 1?

You mention: “Once the zones are installed, you should failover the shared storage to each node in the cluster, and boot the zones to be extra certain”
How do I failover the shared storage to each node – where have you configured shared storage?

Kartik on November 18th, 2010

Okay, I made it this far…..
you suggest: “The zone needs to be installed on each node so the zone gets put into the installed state, and the Sun documentation recommends removing the installed zone on the first node prior to installing it on the second node. ”
so how do I set the zonepath for the zone, I have one shared zpool that is where the zonepath will be, how do I create two zones on one zonepath?

Please help

Kartik on November 18th, 2010

still trying to figure out how to install zones. Also I could not get the zone to boot because, it said address already in use by global zone…..

Kartik on November 18th, 2010

Okay, I completed all the steps, but on one node the zone is in an installed state and it is in running state on the other node, I tested failover but the zone does not failover in running state – ie if the node it is running on fails the zone fails as well……

Kartik on November 19th, 2010

It works! Thanks for the excellent blog entry. However one thing still does not work – the global zone controls the logical interface so I was not able to plumb the zones by it,

Veritas Netbackup (3)
Veritas Volume Manager (19)
VirtualBox (1)
VMWare ESX Server (11)
Web development (3)
Web Utilities (1)
Windows Server (3)
Wireless (2)
Wordpress (7)

Archives

April 2011 (4)
March 2011 (8)
February 2011 (3)
January 2011 (9)
December 2010 (5)
November 2010 (7)
October 2010 (6)
September 2010 (3)
August 2010 (5)
July 2010 (14)
June 2010 (4)
May 2010 (7)
April 2010 (6)
March 2010 (8)
February 2010 (2)
January 2010 (5)
December 2009 (13)
November 2009 (11)
October 2009 (7)
September 2009 (10)
August 2009 (17)
July 2009 (28)
June 2009 (29)
May 2009 (25)
April 2009 (34)
March 2009 (24)
February 2009 (14)
January 2009 (5)
December 2008 (7)
November 2008 (2)
October 2008 (3)
September 2008 (2)
July 2008 (7)
June 2008 (2)
May 2008 (1)
April 2008 (4)
March 2008 (5)
February 2008 (12)
January 2008 (5)
December 2007 (5)
November 2007 (15)
October 2007 (16)
September 2007 (6)
August 2007 (4)
July 2007 (8)
June 2007 (20)
May 2007 (13)
April 2007 (14)

how do I do that?

Kartik on November 21st, 2010

hazone shows up offline on both nodes, still trying to learn and understand this

Kartik on November 29th, 2010

does not work for me. without cluster hooks, the zones start on either node without issue – my issue is that when I put in the zone resource by using the /opt/SUNWsczone/sczbt/util/sczbt_register script the zone stays on OFFLINE state on both nodes, clresource enable brings it online – after a while. The zone then comes online but does not failover.

Leave a Comment

Username (required) :

Email (required) :

Web Site :

Comment :

Submit

March 2007 (13)
February 2007 (13)
January 2007 (34)
December 2006 (27)
November 2006 (24)
October 2006 (17)
September 2006 (23)
August 2006 (18)
July 2006 (31)
June 2006 (23)
May 2006 (19)
April 2006 (30)
March 2006 (25)
February 2006 (28)
January 2006 (25)
December 2005 (23)
November 2005 (30)
October 2005 (54)
September 2005 (49)
August 2005 (10)
July 2005 (8)
June 2005 (5)
May 2005 (10)
April 2005 (6)
March 2005 (6)
February 2005 (13)
January 2005 (15)
December 2004 (3)
November 2004 (4)
October 2004 (2)

Search

The content is copyrighted to Blog O' Matty and may not be reproduced on other websites.

All information is provided AS IS, and no warranty of ANY kind is provided. Use all information on this site at your own risk!

Copyright Blog O' Matty | Corpvox WordPress Theme