Quickly Configure a HA Cluster with the SUSE Linux Enterprise HA Extension 11 SP2

Course ATT1803
Version 1.0.0
Lecture Manual
August 30, 2012
Proprietary Statement
Copyright © 2012 Novell, Inc. All rights reserved.

Novell, Inc., has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the Novell Legal Patents Web page (http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com
Novell Trademarks
For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials
All third-party trademarks are the property of their respective owners.

Software Piracy
Throughout the world, unauthorized duplication of software is subject to both criminal and civil penalties.

If you know of illegal copying of software, contact your local Software Antipiracy Hotline. For the Hotline number for your area, access Novell’s World Wide Web page (http://www.novell.com) and look for the piracy page under “Programs.” Or, contact Novell’s anti-piracy headquarters in the U.S. at 800-PIRATES (747-2837) or 801-861-7101.

Disclaimer
Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

This Novell Training Manual is published solely to instruct students in the use of Novell networking software. Although third-party application software packages are used in Novell training courses, this is for demonstration purposes only and shall not constitute an endorsement of any of these software applications.

Further, Novell, Inc. does not represent itself as having any particular expertise in these application software packages and any use by students of the same shall be done at the student’s own risk.
## Contents

SECTION 1: Introduction to HA Clustering in SLE11 4

Objective 1: Lab Environment Overview 6

Objective 2: Cluster Terminology 14

Objective 3: Overview of SLE HA 19

Objective 4: SLE HA Architecture 24

Objective 5: HA Stack Configuration Overview 32

Objective 6: Important Files and Directories 34

Objective 7: Configure the HA Stack 36

Objective 8: Introduction to the Cluster Information Base 51

Objective 9: Introduction to Cluster Management Utilities 54

Objective 10: Introduction to OCFS2 65

Objective 11: OCFS2 Architecture 68

Objective 12: Configure OOCFS2 Shared Storage 74
HA Clustering with the SLE HA Extension 11
Section 1 – Install and Configure HA Clustering in SLE11
Objectives

- Lab Environment Overview
- Cluster Terminology
- Overview of SLE HA
- SLE HA Architecture
- HA Daemon Configuration Overview
- Important Files and Directories
- Configure the HA Daemon
- Introduction to the Cluster Information Base
- Introduction to Cluster Management Utilities
- Introduction to OCFS2
- OCFS2 Architecture
- Configure an OCFS Cluster
Lab Environment Overview
Lab Network Configuration Diagram

Node1
- eth0
- eth1
- eth2
- eth3

Node2
- eth0
- eth1
- eth2
- eth3

Node3
- eth0
- eth1
- eth2
- eth3

LAN

SAN1
- eth0
- eth1
- eth2
- eth3

SAN2
- eth0
- eth1
- eth2
- eth3

SAN1
- Storage1 (iSCSI Target)

SAN2
- Storage2 (iSCSI Target)

SAN2
- Storage3 (iSCSI Target)

DRBD1

eth0
eth1
eth2
eth3
Node Network Configuration

Node1

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Addr</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>• eth0</td>
<td>172.17.2.11/24</td>
<td>node1</td>
</tr>
<tr>
<td>• eth1</td>
<td>(no address)</td>
<td></td>
</tr>
<tr>
<td>• eth2</td>
<td>192.168.2.11/24</td>
<td>node1-san1</td>
</tr>
<tr>
<td>• eth3</td>
<td>192.168.102.11/24</td>
<td>node1-san2</td>
</tr>
</tbody>
</table>

Disks
- OS = /dev/sda

iSCSI Initiator Name = iqn.1996-04.de.suse:node1
Node Network Configuration

Node2

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Addr</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>172.17.2.12/24</td>
<td>node2</td>
</tr>
<tr>
<td>eth1</td>
<td>(no address)</td>
<td>node2-san1</td>
</tr>
<tr>
<td>eth2</td>
<td>192.168.2.12/24</td>
<td>node2-san2</td>
</tr>
<tr>
<td>eth3</td>
<td>192.168.102.12/24</td>
<td></td>
</tr>
</tbody>
</table>

Disks
- OS = /dev/sda

iSCSI Initiator Name = iqn.1996-04.de.suse:node2
Node Network Configuration

Node3

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Addr</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>172.17.2.13/24</td>
<td>node3</td>
</tr>
<tr>
<td>eth1</td>
<td>(no address)</td>
<td></td>
</tr>
<tr>
<td>eth2</td>
<td>192.168.2.13/24</td>
<td>node3-san1</td>
</tr>
<tr>
<td>eth3</td>
<td>192.168.102.13/24</td>
<td>node3-san2</td>
</tr>
</tbody>
</table>

Disks
- OS = /dev/sda
- iSCSI Initiator Name = iqn.1996-04.de.suse:node3
Node Network Configuration

Storage1

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Addr</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>• eth0</td>
<td>172.17.2.14/24</td>
<td>storage1</td>
</tr>
<tr>
<td>• eth1</td>
<td>172.18.2.14/24</td>
<td>storage1-drbd1</td>
</tr>
<tr>
<td>• eth2</td>
<td>192.168.2.14/24</td>
<td>storage1-san1</td>
</tr>
<tr>
<td>• eth3</td>
<td>192.168.102.14/24</td>
<td>storage1-san2</td>
</tr>
</tbody>
</table>

Disks
- OS = /dev/sda
- Extra = /dev/sdb - /dev/sdh
Node Network Configuration

Storage2

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Addr</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>172.17.2.15/24</td>
<td>storage2</td>
</tr>
<tr>
<td>eth1</td>
<td>172.18.2.15/24</td>
<td>storage2-drbd1</td>
</tr>
<tr>
<td>eth2</td>
<td>192.168.2.15/24</td>
<td>storage2-san1</td>
</tr>
<tr>
<td>eth3</td>
<td>192.168.102.15/24</td>
<td>storage2-san2</td>
</tr>
</tbody>
</table>

Disks
- OS = /dev/sda
- Extra = /dev/sdb - /dev/sdh
## Node Network Configuration

### Storage3

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Addr</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>172.17.2.18/24</td>
<td>storage3</td>
</tr>
<tr>
<td>eth1</td>
<td>172.18.2.18/24</td>
<td>storage3-drbd1</td>
</tr>
<tr>
<td>eth2</td>
<td>192.168.2.18/24</td>
<td>storage3-san1</td>
</tr>
<tr>
<td>eth3</td>
<td>192.168.102.18/24</td>
<td>storage3-san2</td>
</tr>
</tbody>
</table>

### Disks
- OS = /dev/sda
- Extra = /dev/sdb - /dev/sdh
Cluster Terminology
## Cluster Terminology (1/4)

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Member of a cluster</td>
</tr>
<tr>
<td>heartbeat</td>
<td>Signal sent between nodes to notify each other that each node is still “alive”. (With OpenAIS, this performed by the Totem protocol.)</td>
</tr>
<tr>
<td>resource</td>
<td>Anything managed by the cluster. Application, Volume, etc.</td>
</tr>
</tbody>
</table>
## Cluster Terminology (2/4)

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
</table>
| quorum | -The nodes that are alive, participating, and in agreement as to the state of the cluster  
                -Typical formula for determining quorum: $n/2+1$ |
| epoch  | -The “version” of the cluster  
                -Represented as a number comprised of:  
                    $\text{admin\_epoch} \cdot \text{epoch} \cdot \text{num\_updates}$ (i.e. 0.1.13)  
                -based on the number of times the cluster configuration has changed  
                -Linux HA has 2 epoch values:  
                    $\text{admin\_epoch}$ Manually updated by administrator to force an epoch update  
                    $\text{epoch}$ Automatically updated when the cluster configuration changes  
                    $\text{num\_updates}$ Automatically updated when the CIB has been updated |
## Cluster Terminology (3/4)

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Split Brain** | - A state that occurs when cluster nodes are not in agreement with each other on the state of the cluster membership  
- Must be resolved before the cluster can continue to operate |
| **fencing** | - The process of denying access to resources (disks) by cluster nodes to protect the resource |
| **STONITH** | - Acronym: Shoot The Other Node In The Head  
- The process of “killing” a node that is not in agreement with the cluster quorum  
- Method used to “fence” nodes from resources |
### Cluster Terminology (4/4)

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Active/Active**  | - A cluster where all (both) nodes are running instances of a resource simultaneously and providing service  
                      - Uses cloned resources                                                                                                                   |
| **Active/Passive** | - A cluster where only one instance of a service is running on any node at one time  
                      - Uses primitive resources  
                      - **Hot Active/Passive**=Multiple instances of the resource running on simultaneously but only one is actively providing service  
                      - **Cold Active/Passive**=Only one instance of the resource is running at a time.                                                          |
Overview of SLE HA
The SLE High Availability Extension

- Beginning with SLE11, all clustering packages are no longer included on the base Server installation media.

- All clustering packages are available as an extension to the base server and ship on separate installation media named the SLE High Availability Extension (SLE-HAE 11).

- The SLE-HAE 11 can be downloaded freely but support for packages on it is available separate from the support for the base server install.
Main Components of the SLE-HAE

- **High Availability Suite** (Corosync/OpenAIS+Pacemaker)
- **Cluster Logical Volume Manager** (cLVM)
- **Oracle Cluster File System** (OCFS2)
- **Distributed Replicated Block Device** (DRBD)
- **The Linux IP Virtual Server** (IPVS)
Responsibilities of an HA Cluster Manager

Cluster Communication (Corosync/OpenAIS)
• Communicate (heartbeat) with other cluster nodes
• Propagate cluster configuration/state information

Manage cluster node membership (Corosync/OpenAIS)
• Maintain a knowledge of the cluster nodes that are alive
• Manage the process of nodes joining the cluster
• Manage the removal of nodes from the cluster

Manage cluster resources (Pacemaker)
• Maintain a knowledge of the status of cluster resources
• Manage the location of cluster resources
• Ensure that cluster resources are available
Cluster Stack Comparison

- Heartbeat2 vs Corosync/OpenAIS + Pacemaker (with Storage Components)

In the SLE HA Extension the cluster components are more clearly defined in layers by the components themselves.

- Corosync/OpenAIS handles all of the cluster communication and membership
- Pacemaker handles all of resource management services
- Heartbeat components (LRM) are only used for local resource management
- Resource Agent scripts are used to manage the actual resources

Another feature of the SLE HA Extension is its close integration with the new common Distributed Lock Manager (DLM). All of the cluster aware storage components now use this common DLM for their locking. The DLM uses the totem protocol (managed by Corosync) for its cluster communication.
SLE HA Architecture
Cluster Resource Manager (CRM)

- The main controlling process in Linux HA
- Every action taken in the Resource Allocation Layer is handled by the CRM
- All communication in the Resource Allocation Layer and higher passes through the CRM
- Maintains the Cluster Information Base (CIB)
- One CRM in the cluster is selected as the Designated Coordinator (DC) or the “Master”
- All writes to the CIB are done by the DC

Acronyms:
CRM = Cluster Resource Manager
DC = Designated Controller (master node)
Cluster Information Base (CIB)

- An in memory XML representation of the configuration and status of the cluster
- There is one “Master” CIB in the cluster maintained by the DC. All other nodes contain a replica.
- Changes to the CIB can be made using the cibadmin command (xml), the crm shell or other utilities
- The CRM on the DC replicates changes to the CIB to the CRMs on the other cluster nodes

Acronyms:
CRM = Cluster Resource Manager
DC = Designated Controller (master node)
CIB = Cluster Information Base
Policy Engine (PE)

- The PE process runs on each node but is only active on the DC node
- The PE helps determine which steps must be taken cluster wide by the DC in reaction to a change in the DC
- Generates a transition graph of all resource actions and dependencies to achieve the next cluster state
- The PE sends messages to the relevant CRMs whose Local Resource Managers (LRMs) perform the specified resource manipulations

Acronyms:
CRM = Cluster Resource Manager
DC = Designated Controller (master node)
CIB = Cluster Information Base
PE = Policy Engine
Quickly Configure a HA Cluster with the SUSE Linux Enterprise HA Extension 11 SP2

STONITH Daemon

- Daemon that is used to fence nodes
- Uses “plug-in” agents to perform the fencing actions
- STONITH-NG is the next evolution of the STONITH daemon that supports monitoring, notification and other features

Acronyms:
CRM = Cluster Resource Manager
DC = Designated Controller (master node)
CIB = Cluster Information Base
PE = Policy Engine
STONITH = Shoot The Other Node In The Head (Fencing daemon)
Local Resource Manager (LRM)

- Calls local Resource Agents (RAs) on behalf of the CRM
- It performs start / stop / monitor operations reporting the result to the CRM
- The LRM is authoritative about the state of its local node

Acronyms:
CRM = Cluster Resource Manager
DC = Designated Controller (master node)
CIB = Cluster Information Base
PE = Policy Engine
STONITH = Shoot The Other Node In The Head (Fencing daemon)
LRM = Local Resource Manager
Resource Agents (RA)

- A script that starts / stops / monitors a local resource (service)
- RAs are called by the LRM
- RAs should written to a spec called the Open Clustering Framework (OCF)
- Third parties can drop their own custom RAs into a defined location in the file system to provide cluster integration of their products

Acronyms:
CRM = Cluster Resource Manager
DC = Designated Controller (master node)
CIB = Cluster Information Base
PE = Policy Engine
LRM = Local Resource Manager
STONITH = Shoot The Other Node In The Head (Fencing daemon)
RA = Resource Agent
Corosync/OpenAIS

- Cluster communication daemon
- Uses Totem protocol as the cluster “heartbeat”
- Propagates cluster status and configuration information between the cluster nodes

Acronyms:
CRM = Cluster Resource Manager
DC = Designated Controller (master node)
CIB = Cluster Information Base
PE = Policy Engine
LRM = Local Resource Manager
STONITH = Shoot The Other Node In The Head (Fencing daemon)
RA = Resource Agent
OpenAIS = Cluster membership daemon
HA Stack Configuration Overview
SLE HA Daemon Configuration Overview

Steps to Configure Linux HA

• Install the SLE HA Extension on all cluster nodes
• Configure The cluster manager on one cluster node
• Propagate the cluster configuration to all cluster nodes
• Start cluster manager daemon on all nodes
Important HA Files and Directories
# Important HA Files and Directories

<table>
<thead>
<tr>
<th>File/Dir</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/corosync/</td>
<td>-main configuration directory for the corosync/openais daemon</td>
</tr>
<tr>
<td>/etc/corosync/corosync.conf</td>
<td>-main configuration file for the corosync/openais daemon</td>
</tr>
<tr>
<td>/etc/corosync/authkey</td>
<td>-file defining cluster communication encryption key (this is a binary file)</td>
</tr>
<tr>
<td>/var/lib/heartbeat/</td>
<td>-directory containing transient pacemaker daemon data</td>
</tr>
<tr>
<td>/var/lib/heartbeat/crm/cib.xml</td>
<td>-persistent copy of the CIB stored on disk</td>
</tr>
<tr>
<td>/etc/init.d/openais</td>
<td>-init script used to start/stop the corosync/openais daemon</td>
</tr>
</tbody>
</table>

The `/etc/corosync/corosync.conf` (and `/etc/corosync/authkey` if cluster encryption is enabled) files must be identical on all cluster nodes.

The `/var/lib/heartbeat/crm/cib.xml` file should not be edited by hand. A checksum of the file is made by Pacemaker and if the file does not match the checksum Pacemaker will refuse to load the file and in turn fail to start.

The `/etc/init.d/openais` init script first checks to see if the SBD daemon has been configured (by checking for the presence of the `/etc/sysconfig/sbd` file) and if so attempts to start the SBD daemon first. If the SBD daemon starts successfully (or if SBD is not configured) it then starts the Corosync daemon. The Corosync daemon then starts the Pacemaker process. If the SBD daemon is configured but does not start then the init script errors out ad does not attempt to start the Corosync daemon.
Configure the HA Stack
Configure the Cluster with YaST
Configure the HA Daemon (1/6)

YaST2 Cluster

- YaST -> Other -> Cluster
  - YaST cluster
  - Performs initial configuration of the corosync/openais daemon configuration files
    - corosync.conf
    - authkeys
... Or with the Bootstrap Scripts
Cluster Bootstrap Scripts

The Quickest way to get a cluster configured

- **sleha-init**
  - Configures the cluster on the first cluster node
  - Configures/enables the following:
    > Corosync (*without* an encryption key)
    > Csync2 (with an encryption key)
    > SBD (if shared storage is provided)
    > Hawk

- **sleha-join**
  - Copies the cluster configuration from the first cluster node to the new cluster node

The cluster bootstrap scripts configure Corosync without encryption by default. Enabling encryption (also known as secauth) adds CPU processing overhead to all cluster communication message handling and decreases total throughput of the cluster communication channel.

If encryption of the cluster communication is desired you may do the following after running the **sleha-init** script on the first cluster node before running **sleha-join** on the other cluster nodes:

1) Generate a cluster encryption key: `corosync-keygen`
2) Enable secauth in the `corosync.conf` file: (change `secauth: no` to `secath: yes`)
LAB 1-1: Install the SLE HA Extension

Summary: In this exercise, you add the SUSE Linux Enterprise High Availability Extension as a software installation source and then install the HA Extension components.

Duration: 10 min.
LAB 1-2: Create a LVM Volume Group on the Storage Server

Summary: In this exercise, you create a LVM volume group from multiple disks on the iSCSI target server.

Duration: 10 min.
LAB 1-3: Create a 4MB Logical Volume for an SBD Device

Summary: In this exercise you create a logical volume on the iSCSI target server to be an SBD device.

Duration: 10 min.
LAB 1-4: Create a Logical Volume for OCFS2 on the Storage Server

Summary: In this exercise, you create a logical volume on the iSCSI target server to be an OCFS2 volume.

Duration: 10 min.
LAB 1-5: Configure an iSCSI Target Server from the CLI

Summary: In this exercise you configure an iSCSI Target Server.

Duration: 10 min.
LAB 1-6: Create an iSCSI Target for the SBD Device from the CLI

Summary: In this exercise you create a new iSCSI target on the target server for the SBD device.

Duration: 10 min.
LAB 1-7: Create an iSCSI Target for the OCFS2 Volume from the CLI

Summary: In this exercise, you create a new iSCSI target on the target server for the OCFS2 volume.

Duration: 10 min.
LAB 1-8: Configure an iSCSI Initiator

Summary: In this exercise configure the iSCSI initiator on the cluster nodes and discover and connect to iSCSI target(s) hosted on a SLES server.

Special Instructions
Use the following values in the exercise:

TARGET_SERVER_IP1=192.168.2.14
INITIATOR_NAME=iqn.1996.de.suse:(node name)

Duration: 10 min.
LAB 1-9: Configure the First Cluster Node with sleha-init

Summary: In this exercise you use the sleha-init cluster bootstrap script to configure the first cluster node.

Duration: 10 min.
LAB 1-10: Configure the Other Cluster Nodes with sleah-join

**Summary:** In this exercise you use the sleha-join cluster bootstrap script to configure the other cluster nodes.

**Duration:** 10 min.
Introduction to the Cluster Information Base
The Cluster Information Base (CIB)

```
<clib epoch="10" num_updates="7" admin_epoch="0" validate-with="pacemaker-1.2" crm_feature_set="3.0.1" have-quorum="1" dc-uuid="node1">
  <configuration>
    <crm_config>
      <cluster_property_set id="cib-bootstrap-options">
        <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="1.1.1-536bf0b9d3ba6d412c67b27f89682ae9380b28ff"/>
        <nvpair id="cib-bootstrap-options-cluster-infrastructure" name="cluster-infrastructure" value="openais"/>
        <nvpair id="cib-bootstrap-options-expected-quorum-votes" name="expected-quorum-votes" value="2"/>
      </cluster_property_set>
    </crm_config>
    <nodes>
      <node id="node1" uname="node1" type="normal"/>
      <node id="node2" uname="node2" type="normal"/>
    </nodes>
    <resources/>
    <constraints/>
  </configuration>
  <status>
    <node_state id="node1" uname="node1" ha="active" in_ccm="true" crmd="online" join="member" expected="member" crm-debug-origin="do_state_transition" shutdown="0">
      <transient_attributes id="node1">
        <instance_attributes id="status-node1">
          <nvpair id="status-node1-probe_complete" name="probe_complete" value="true"/>
        </instance_attributes>
      </transient_attributes>
    </node_state>
    <node_state id="node2" uname="node2" ha="active" in_ccm="true" crmd="online" join="member" expected="member" crm-debug-origin="do_state_transition" shutdown="0">
      <transient_attributes id="node2">
        <instance_attributes id="status-node2">
          <nvpair id="status-node2-probe_complete" name="probe_complete" value="true"/>
        </instance_attributes>
      </transient_attributes>
    </node_state>
  </status>
</clib>
```

Example CIB XML from a running 2 node cluster
The Cluster Information Base (CIB)

```
<configuration>
  <crm_config>
    <cluster_property_set id="cib-bootstrap-options">
      <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="1.1.1-536bf0b9d3ba6d412c67b7f89682ae9380b28ff"/>
      <nvpair id="cib-bootstrap-options-cluster-infrastructure" name="cluster-infrastructure" value="openais"/>
      <nvpair id="cib-bootstrap-options-expected-quorum-votes" name="expected-quorum-votes" value="2"/>
    </cluster_property_set>
  </crm_config>
  <nodes>
    <node id="node1" uname="node1" type="normal"/>
    <node id="node2" uname="node2" type="normal"/>
  </nodes>
  <resources/>
  <constraints/>
</configuration>

<status>
  <node_state id="node1" uname="node1" ha="active" in_ccm="true" crmd="online" join="member" expected="member" crm-debug-origin="do_state_transition" shutdown="0">
    <instance_attributes id="status-node1">
      <nvpair id="status-node1-probe_complete" name="probe_complete" value="true"/>
    </instance_attributes>
  </node_state>
  <node_state id="node2" uname="node2" ha="active" in_ccm="true" crmd="online" join="member" expected="member" crm-debug-origin="do_state_transition" shutdown="0">
    <instance_attributes id="status-node2">
      <nvpair id="status-node2-probe_complete" name="probe_complete" value="true"/>
    </instance_attributes>
  </node_state>
</status>
```

The Cluster Information Base consists of two main parts `<configuration>` and `<status>`. The `<cib>` tag also contains important information about the cluster.

**<configuration>**
- Stores the cluster configuration information
- This is stored on disk so that it can be loaded into memory when the heartbeat daemon starts
- This portion of the CIB is persistent

**<status>**
- Stores the current state of the cluster
- This is dynamically generated and therefore never stored on disk
- This portion of the CIB is dynamically generated.

The `<cib>` section contains information about the state of the cluster as known by an individual node.
Introduction to Cluster Management Utilities
The CRM_GUI is a graphical cluster management utility.
HA Web Konsole (Hawk)

The HA Web Konsole (Hawk) is a web based graphical cluster management utility.
**crm_mon Command**

- The `crm_mon` command is a command line utility that displays the current state of the cluster.

The `crm_mon` command is no-interactive and does nothing more than to display the current state of the cluster (nodes and resource). The default refresh time is 15 seconds. This can be changed using the `-i` option.

Example: `crm_mon -i 1` (for a 1 second refresh)
The crm Shell

• The crm shell is a command line utility that can be used to create, view and manage cluster resources and the cluster in general

• The crm shell can be execute as a command line utility with the options appended on a single command line

• The crm shell can be invoked as an interactive shell

```
node1:~/Desktop # crm configure
crm(live)configure# up
crm(live)# resource
crm(live)resource# show
  stonith-sbd  (stonith:external/sbd) Started
  IP_1  (ocf::heartbeat:IPaddr2) Started
Clone Set: base-clone [base-group]
         Started: [ node1 node2 ]
         Stopped: [ base-group:2 ]
Clone Set: c-clusterfs [clusterfs]
         Started: [ node1 node2 ]
         Stopped: [ clusterfs:2 ]
crm(live)resource# 
```

The crm command has a number of sub functions that are used to manage the cluster. Here are some examples of how the crm command can be used:

**Resource Management:**
crm resource is used to manage cluster resources

**Resource Configuration:**
crm configure is used to configure cluster resources

**Node Configuration:**
crm node is used to configure cluster nodes
cibadmin

- Administrative command for directly manipulating the CIB using XML
- Operates on the XML trees of the CIB

```bash
node1:/Desktop # cibadmin -Q
<cib epoch="34" num_updates="2507" admin_epoch="0" validate-with="pacemaker-1.2"
  crm_feature_set="3.0.5" update-origin="node1" update-client="cibadmin" update-user="hacluster"
  cib-last-written="Fri Apr 6 09:54:48 2012" have-quorum="1" dc-uid="node1">
  <configuration>
    <crm_config>
      <cluster_property_set id="cib-bootstrap-options">
        <nvpair id="cib-bootstrap-options-stonith-enabled" name="stonith-enabled" value="true"/>
        <nvpair id="cib-bootstrap-options-no-quorum-policy" name="no-quorum-policy" value="freeze"/>
        <nvpair id="cib-bootstrap-options-placement-strategy" name="placement-strategy" value="balanced"/>
        <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="1.1.6-b988976485d15cb702c9307df55512d323831a5e"/>
        <nvpair id="cib-bootstrap-options-cluster-infrastructure" name="cluster-infrastructure" value="openais"/>
    </cluster_property_set>
  </configuration>
</cib>
```

The cibadmin utility does not have knowledge of the meaning of the updates and queries to the CIB, it simply modifies or displays the xml as specified. It can only deal with valid XML subtrees.
Csnc2

- Allows the synchronization of files across multiple machines simultaneously via a secure channel.

The cibadmin utility does not have knowledge of the meaning of the updates and queries to the CIB, it simply modifies or displays the XML as specified. It can only deal with valid XML subtrees.
Parallel SSH (pssh)

- Allows execution of command on multiple machines simultaneously via ssh

The cibadmin utility does not have knowledge of the meaning of the updates and queries to the CIB, it simply modifies or displays the xml as specified. It can only deal with valid XML subtrees.
LAB 1-11: Connect to the HA Web Konsole (Hawk)

Summary: In this exercise you enable and then browse to and bookmark the HA Web Konsole (Hawk).

Duration: 10 min.
LAB 1-12: Set Cluster Defaults with Hawk

Summary: In this exercise, you explicitly set some cluster defaults using the HA Web Konsole.

Duration: 10 min.
LAB 1-13: Configure Parallel SSH on the Cluster Nodes

Summary: In this exercise you configure Parallel SSH on the cluster nodes.

Duration: 10 min.
Introduction to OCFS2
Description of OCFS2

• Parallel cluster aware file system that allows multiple hosts to mount it simultaneously as a local file system.
• Used in clusters to provide shared storage.
• Consists of 2 main components:
  – Kernel file system driver - ocfs2.ko
  – Cluster manager – o2cb or Pacemaker
Advantages of OCFS2

- Multiple nodes can directly access shared storage (SAN)
- All file system access operations are done in kernel space
  - Avoid user space context switch required by traditional “network file systems” that have user space daemons
OCFS2 Architecture
OCFS2 with Pacemaker (1/4)

- User-space cluster manager
- Required to mount an OCFS2 file system in cluster mode
- Comprised of 4 Main components
  - Node Manager (CRM / CIB / PE)
  - Heartbeat / Network Communication Agent (OpenAIS)
  - Distributed Lock Manager (DLM)
  - Cluster File System Manager (O2CB)
- Runs in user space (Pacemaker, OpenAIS, O2CB) and kernel space (DLM)
OCFS2 with Pacemaker (2/4)

Distributed Lock Manager (DLM)

- Pacemaker contains the new common Distributed Lock manager that is also used by the cluster LVM daemon

- The DLMMaintains locking state of all resources across the cluster
  - Supports cluster aware flock system call
  - POSIX lockf and fcntl locks are now supported (unlike with the built-in OCFS2 DLM)

- Locks are distributed across all nodes to distribute overhead and provide failover

- The new common DLM uses network communication via OpenAIS to communicate the locks to the cluster nodes
OCFS2 with Pacemaker (3/4)

Kernel File System Drivers

- **OCFS2**
  - Kernel file system module responsible for plugging into VFS to provide a POSIX interface to the file system
  - Manages all file system operations with mounted OCFS2 volumes

- **JBD**
  - Journaling Block Device kernel module that provides all file system journaling for the OCFS2 file system kernel module

- **VFS**
  - The Virtual File System layer that provides a uniform POSIX interface to the kernel and applications
OCFS2 with Pacemaker (4/4)

OCFS2 Volume

- The OCFS2 file system is a general purpose file system that is parallel cluster aware
- Can be mounted by a cluster (requires active O2CB stack) or as a local file system
  - File system type (local/cluster) can be changed with the `tunefs.ocfs2` command
OCFS2 Architecture

OCFS2 Cluster Node Relationships
(SLE-HAE)
Configure OCFS2 Shared Storage
Configure an OCFS2 Cluster

Steps

• Install OCFS2 packages (installed with the HA Extension by default)

• Configure a DLM clone cluster resource

• Configure an O2CB clone cluster resource

• Format SAN volume with an OCFS2 file system

• Configure an OCFS2 volume clone cluster resource
LAB 1-14: Configure OCFS2 Shared Storage with a Wizard

**Summary:** In this exercise you configure OCFS2 shared storage using a Hawk wizard.

**Special Instructions**
Use the following values in the exercise:

**Duration:** 10 min.
Quickly Configure a HA Cluster with the SUSE Linux Enterprise HA Extension 11 SP2

Unpublished Work of SUSE. All Rights Reserved.
This work is an unpublished work and contains confidential, proprietary, and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer
This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.