

Linux HA群集安装与基本配置



概述

- ▶ 规划设计
- ▶ 操作系统安装
- ▶ 群集组件安装
- ▶ 群集节点准备
- ▶ 群集的创建

◆ 规划设计

- ▶ 规划设计的考虑点
- ▶ 群集拓扑结构

规划设计的考虑点

▸ 软件

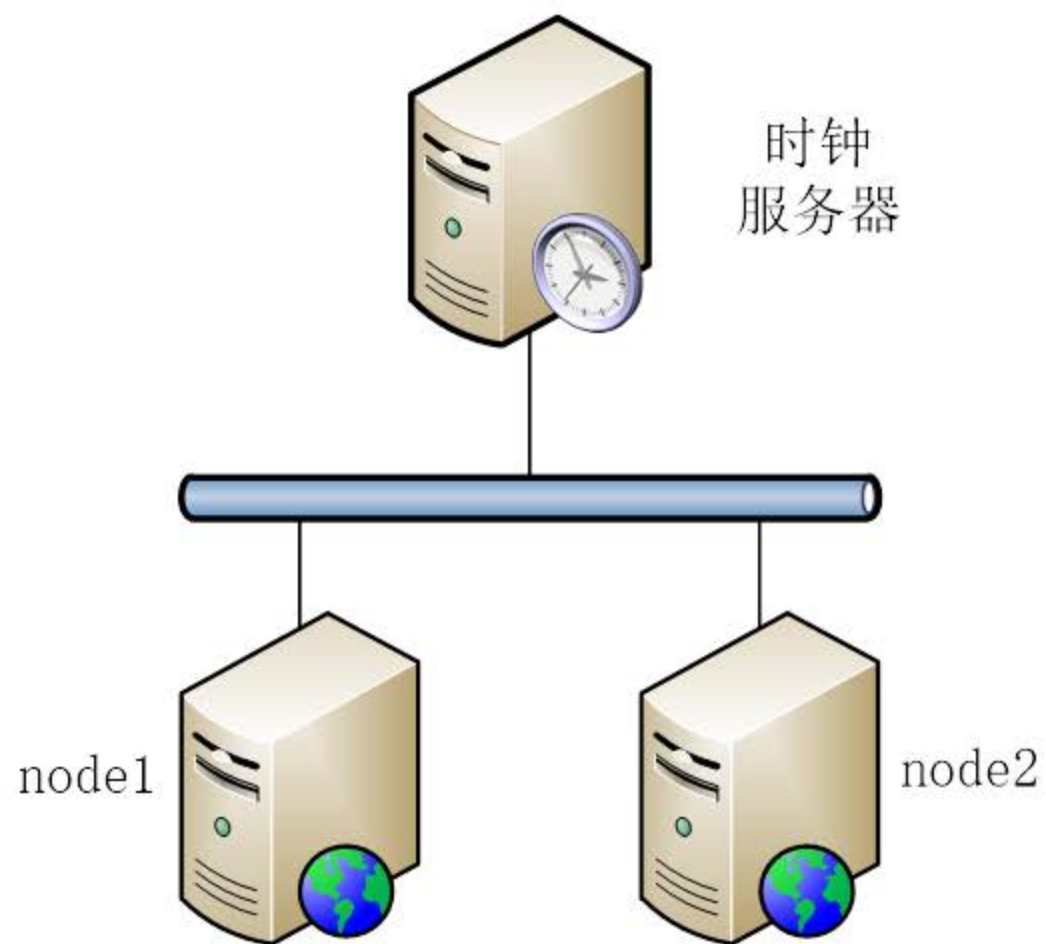
- 群集类型：A/P，A/A....
- 操作系统：Redhat/CentOS，Suse，Ubuntu....
- 群集软件：Pacemaker+Corosync，.....
- 群集配置：群集名称、资源依赖、次序等约束

▸ 硬件

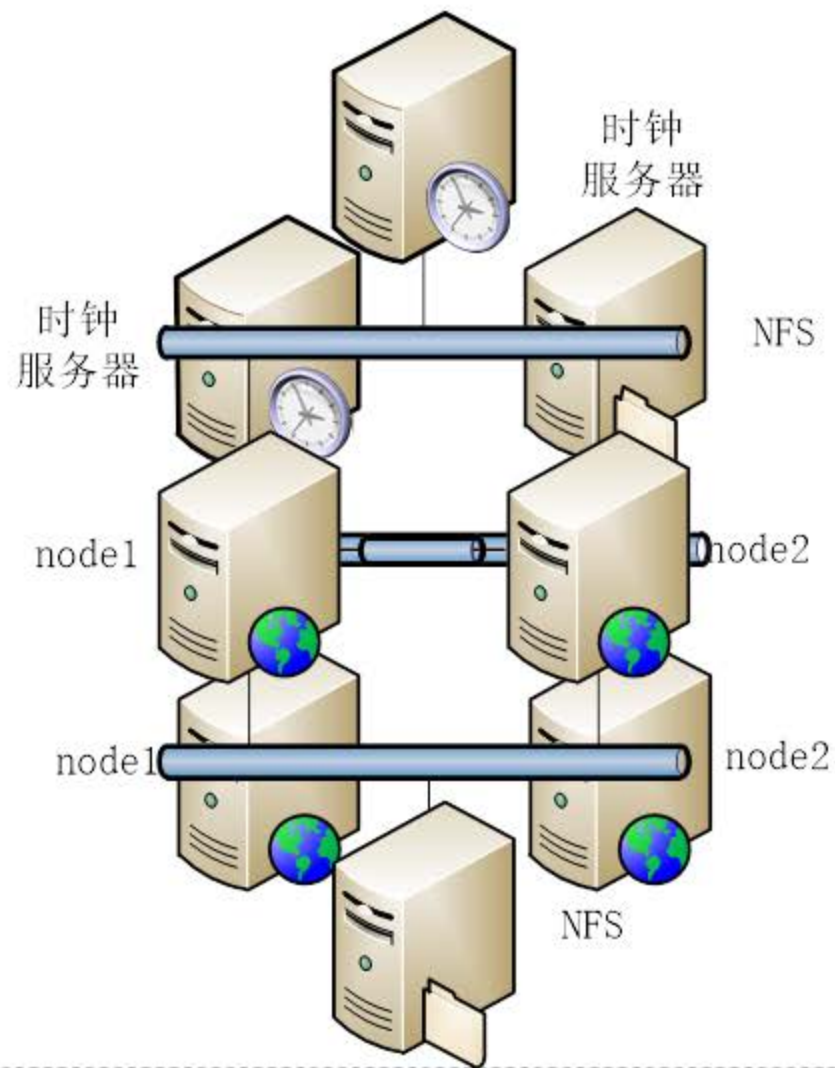
- 服务器：
 - HCL、带外管理、冗余电源.....
- 网络：
 - 将业务、心跳、存储、带外管理网络分开
 - 网卡的捆绑、冗余
- 存储：
 - 控制器、接口、RAID级别....
 - 空间规划



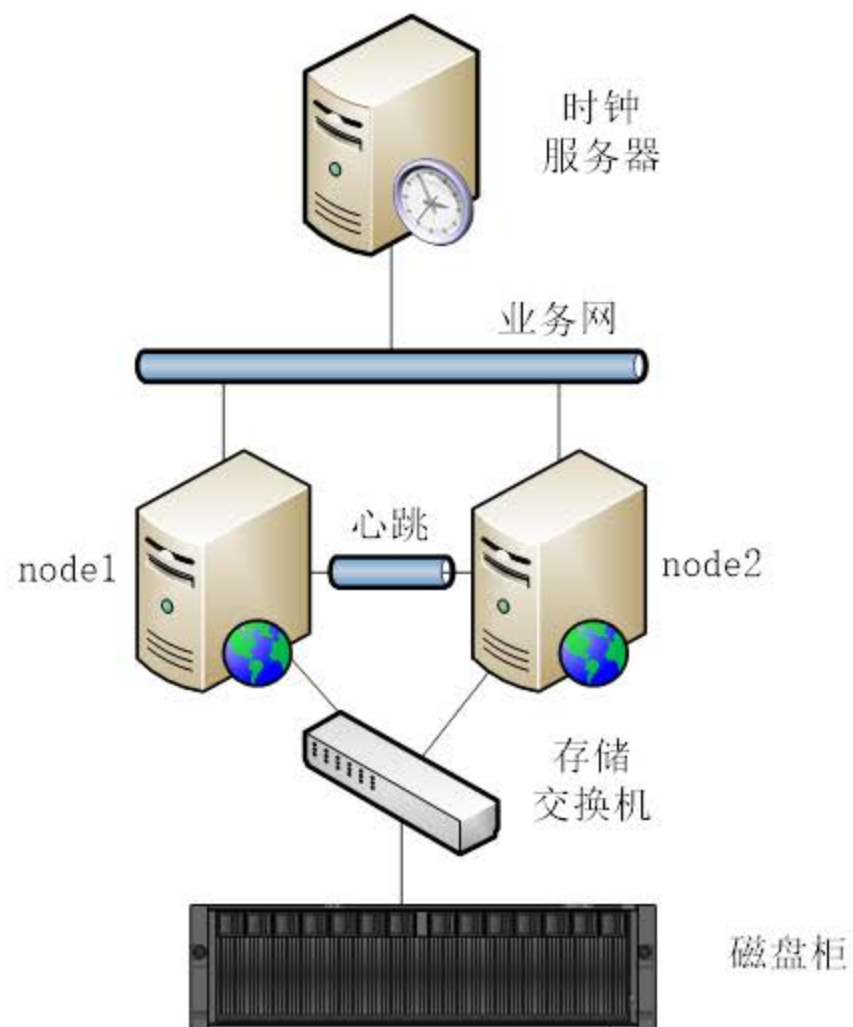
小型环境



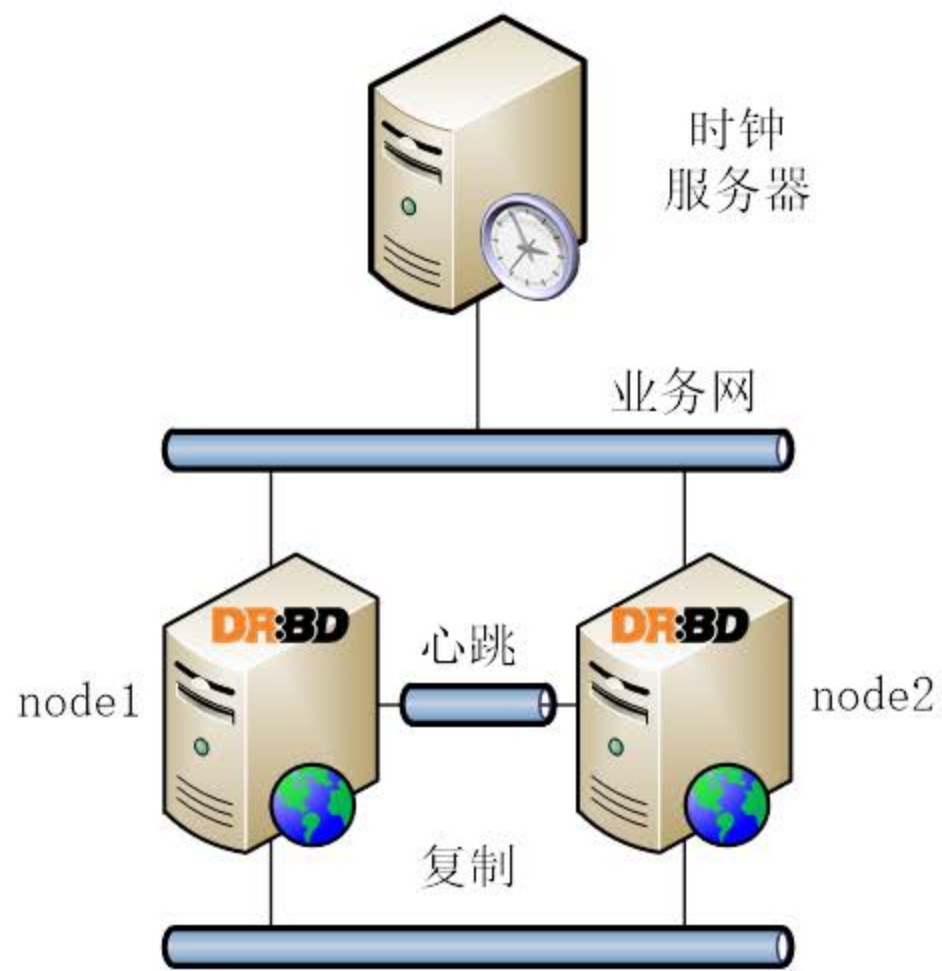
使用NFS共享存储



使用SAN共享存储



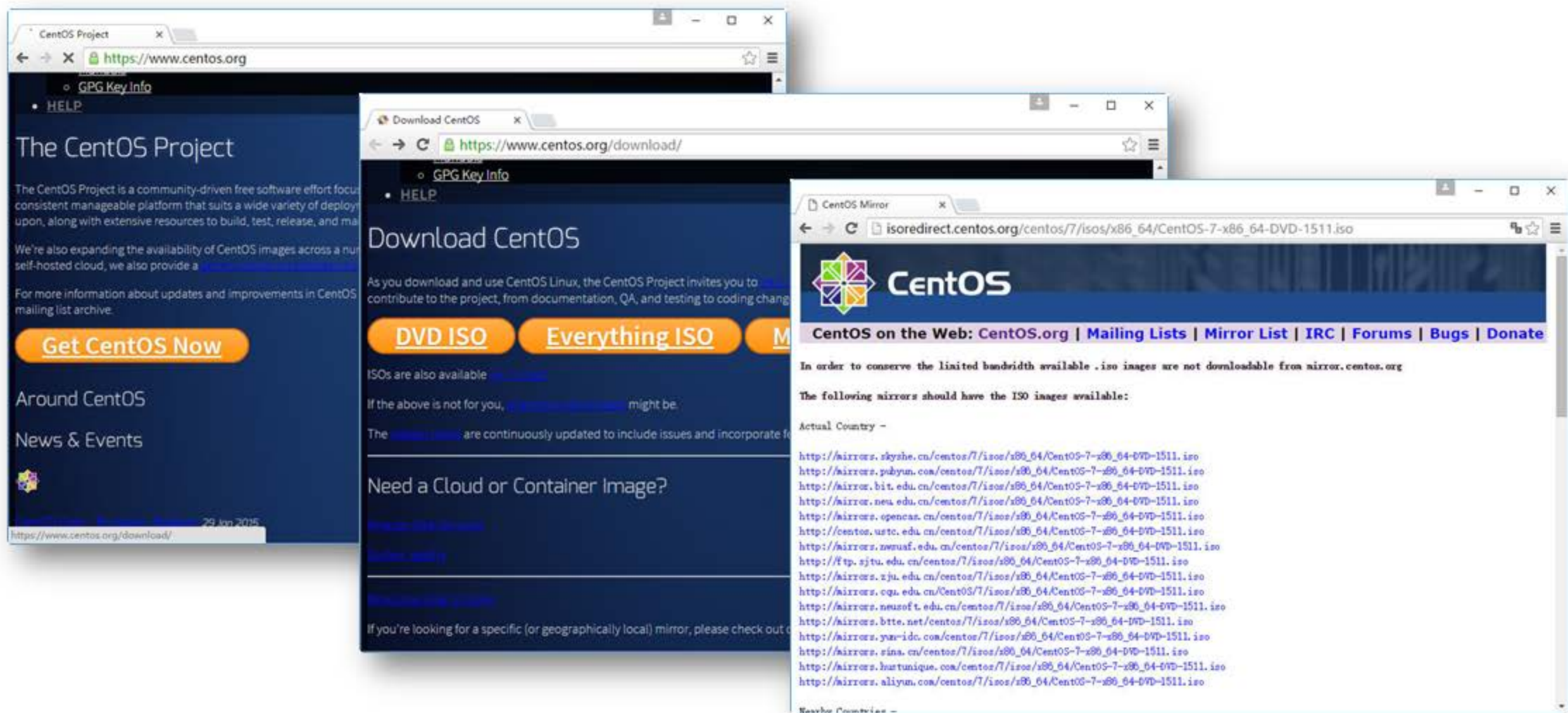
使用DRBD复制



◆ 操作系统安装

- ▶ 操作系统安装
- ▶ 通过kickstart简化安装
- ▶ 操作系统升级

CentOS 7 安装ISO文件下载



CentOS 操作系统安装-GUI



启用网络连接

```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.el7.x86_64 on an x86_64

localhost login: root
Password:
[root@localhost ~]# ifconfig
eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 00:0c:29:cb:09:89 txqueuelen 1000 (Ethernet)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 128 bytes 11136 (10.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 128 bytes 11136 (10.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:8d:c0:c2 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]#
```

```
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno16777736
UUID=51ca47f2-49c8-47af-a26c-f82d4ea6a120
DEVICE=eno16777736
ONBOOT=yes
```


通过kickstart来简单安装

```
192.168.1.11/mini.txt x
192.168.1.11/mini.txt
# 最小配置的VM 用于测试
#
install
cdrom
text
lang en_US.UTF-8
keyboard us
network --onboot yes --device eth0 --bootproto dhcp --noipv6
rootpw 123456
firewall --disabled
authconfig --enableshadow --passalgo=sha512
selinux --disabled
timezone --utc Asia/Shanghai
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb quiet"
reboot

# 下面是测试环境下，用于分区操作，生产环境中不要使用
#clearpart --all --drives=vda
#part /boot --fstype=ext4 --size=100 --ondisk=vda
#part pv.008002 --grow --size=1 --grow --ondisk=vda
#volgroup VolGroup --pesize=4096 pv.008002
#logvol /home --fstype=ext4 --name=lv_home --vgname=VolGroup --grow --size=100
#logvol swap --name=lv_swap --vgname=VolGroup --grow --size=1024 --maxsize=1024
#logvol / --fstype=ext4 --name=lv_root --vgname=VolGroup --grow --size=1024
#

%packages
@base
@core
%end
```



◆ 群集组件安装

- ▶ 配置yum库
- ▶ 安装 Pacemaker 等群集组件

配置yum使用本地CDROM

```
# cd /etc/yum.repos.d/
# ls
CentOS-Base.repo  CentOS-CR.repo  CentOS-Debuginfo.repo  CentOS-fasttrack.repo
CentOS-Media.repo  CentOS-Sources.repo  CentOS-Vault.repo
# mkdir bak
# mv *.repo bak/
# cp bak/CentOS-Media.repo .
# vi /etc/yum.repos.d/CentOS-Media.repo
.....
[c7-media]
name=CentOS-$releasever - Media
baseurl=file:///media/CentOS/
        file:///media/cdrom/
        file:///media/cdrecorder/
gpgcheck=1
enabled=0  ← 将0修改1，以启用本地CDROM
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

# mkdir /media/CentOS
# mount /dev/cdrom /media/CentOS/
```

安装 Pacemaker 配置工具

- ▶ 通过yum来安装软件包，会自动安装其依赖的软件包

```
# yum -y install pacemaker corosync pcs \
psmisc policycoreutils-python fence-agents-all
```

- ▶ 了解一个每个软件包的用途，是一个好的学习习惯

```
# rpm -qi corosync
```

```
.....
```


◆ 群集节点准备

- ▶ 配置主机名及解析
- ▶ 配置SSH Key互信(可选)
- ▶ 配置时钟
- ▶ 配置iptables 防火墙允许集群组件运行
- ▶ 配置pcs守护程序
- ▶ 配置hacluster账户密码
- ▶ 集群配置文件

配置主机名及解析

- ▶ 使用相对短的名称，在所有节点上配置主机名

```
# hostnamectl set-hostname node1
```

- ▶ 配置DNS服务器或hosts文件进行名称解析

```
# vi /etc/hosts
```

添加如下内容

```
192.168.206.91 node1
```

```
192.168.206.92 node2
```

◆ 群集节点准备

✔ 配置主机名及解析

- ▶ 配置SSH Key互信(可选)
- ▶ 配置时钟
- ▶ 配置iptables 防火墙允许集群组件运行
- ▶ 配置pcs守护程序
- ▶ 配置hacluster账户密码
- ▶ 集群配置文件

配置SSH Key互信(可选)

- ▶ 创建ssh key来免去登入要输入密码的麻烦
- ▶ 节点1

```
# ssh-keygen -t rsa -P ''  
# ls ~/.ssh/*  
/root/.ssh/id_rsa  
/root/.ssh/id_rsa.pub  
/root/.ssh/known_hosts  
# ssh-copy-id -i ~/.ssh/id_rsa.pub root@node2
```

- ▶ 节点2

```
# cat ~/.ssh/authorized_keys
```



◆ 群集节点准备

✓ 配置主机名及解析

✓ 配置SSH Key互信(可选)

- ▶ 配置时钟
- ▶ 配置iptables 防火墙允许集群组件运行
- ▶ 配置pcs守护程序
- ▶ 配置hacluster账户密码
- ▶ 集群配置文件

配置时钟

- ▶ 群集中各个节点的时钟要一致
- ▶ 部署企业自有的NTP服务器
 - ▶ Linux NTP服务器
 - ▶ Windows域控制器
- ▶ 修改cron表，通过NTP服务器来校时



```
# /sbin/ntpdate time.windows.com
```

```
# crontab -e
```

添加如下一行，每30分钟同步一次时钟。

```
*/30 * * * * /sbin/ntpdate time.windows.com &> /dev/null
```

检查一下两个节点的时钟是否一致。

```
# ssh node2 'date'; date
```


◆ 群集节点准备

- ✓ 配置主机名及解析
- ✓ 配置SSH Key互信(可选)
- ✓ 配置时钟
 - ▶ 配置iptables 防火墙允许集群组件运行
 - ▶ 配置pcs守护程序
 - ▶ 配置hacluster账户密码
 - ▶ 集群配置文件

配置 iptables 防火墙允许集群组件运行

- ▶ 需要为进入的流量启用以下端口：
 - ▶ TCP：端口 2224、3121、21064
 - ▶ UDP：端口 5405
 - ▶ DLM（如果使用 cLVM/GFS2 的 DLM 锁管理器）：端口 21064
- ▶ 添加防火墙规则

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
# firewall-cmd --reload
```



◆ 群集节点准备

- ✓ 配置主机名及解析
- ✓ 配置SSH Key互信(可选)
- ✓ 配置时钟
- ✓ 配置iptables 防火墙允许集群组件运行
 - ▶ 配置pcs守护程序
 - ▶ 配置hacluster账户密码
 - ▶ 集群配置文件

配置 pcs守护程序

- ▶ pcs软件包包括守护程序、命令行工具、GUI (Web)工具

```
# rpm -qi pcs
```

```
.....
```

```
Description :
```

```
pcs is a corosync and pacemaker configuration tool. It permits users  
to easily view, modify and created pacemaker based clusters.
```



- ▶ pcs daemon , 同步群集中所有节点的corosync的配置
- ▶ 所有节点的pcs daemon必须设置为自动启动, 而且启动

```
# systemctl start pcsd
```

```
# systemctl enable pcsd
```

```
pcsd与pcsd.services一样
```

```
# systemctl status pcsd.service
```

pcs命令行工具概述

```
# pcs --help
```

```
Usage: pcs [-f file] [-h] [commands]...
```

```
Control and configure pacemaker and corosync.
```

▶ 有以下命令

- ▶ cluster 配置集群选项和节点
- ▶ resource 创建和管理集群资源
- ▶ stonith 将 fence 设备配置为与 Pacemaker 一同使用
- ▶ constraint 管理资源限制
- ▶ property 设定 Pacemaker 属性
- ▶ status 查看当前集群和资源状态
- ▶ config 以用户可读格式显示完整集群配置

◆ 群集节点准备

- ✓ 配置主机名及解析
- ✓ 配置SSH Key互信(可选)
- ✓ 配置时钟
- ✓ 配置iptables 防火墙允许集群组件运行
- ✓ 配置pcs守护程序
 - ▶ 配置hacluster账户密码
 - ▶ 集群配置文件

配置hacluster账户密码

- ▶ 安装群集软件包时，会创建一个账户hacluster，它的密码是禁用的。

```
# tail -n 2 /etc/passwd
hacluster:x:189:189:cluster user:/home/hacluster:/sbin/nologin
unbound:x:994:991:Unbound DNS resolver:/etc/unbound:/sbin/nologin
# tail -n 2 /etc/shadow
hacluster:!!:17058::::::
unbound:!!:17058::::::
```

- ▶ 此账户用于群集间通信时的身份验证
- ▶ 必须在每个节点上设置其密码以启用此账户

```
# echo "linuxplus" | passwd --stdin hacluster
Changing password for user hacluster.
passwd: all authentication tokens updated successfully.
```



◆ 群集节点准备

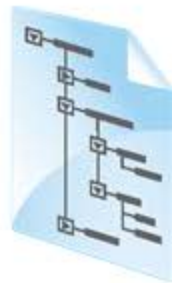
- ✓ 配置主机名及解析
- ✓ 配置SSH Key互信(可选)
- ✓ 配置时钟
- ✓ 配置iptables 防火墙允许集群组件运行
- ✓ 配置pcs守护程序
- ✓ 配置hacluster账户密码
 - ▶ 集群配置文件

集群及 Pacemaker 配置文件

- ▶ 配置文件为 corosync.conf 和 cib.xml。默认不存在

```
# ls /etc/corosync/  
corosync.conf.example      corosync.conf.example.udpu  corosync.xml.example  
uidgid.d/  
# ls /var/lib/pacemaker/cib/cib.xml  
ls: cannot access /var/lib/pacemaker/cib/cib.xml: No such file or directory
```

- ▶ corosync.conf 文件提供 corosync 使用的集群参数
- ▶ cib.xml 是 XML 文件，存储集群配置及所有资源的信息。pcsd守护程序负责在整个集群节点上同步 CIB 的内容
- ▶ 最佳策略
 - ▶ 虽然可以手工创建、修改，但建议通过pcs工具进行管理和维护



◆ 群集节点准备

- ✓ 配置主机名及解析
- ✓ 配置SSH Key互信(可选)
- ✓ 配置时钟
- ✓ 配置iptables 防火墙允许集群组件运行
- ✓ 配置pcs守护程序
- ✓ 配置hacluster账户密码
- ✓ 集群配置文件

◆ 群集创建

按照以下步骤创建运行的集群：

▶ 认证组成集群的节点

```
# pcs cluster auth [node] [...] [-u username] [-p password]
```

▶ 配置和同步集群节点

```
# pcs cluster setup [--start] [--local] --name cluster_name node1 [node2] [...]
```

▶ 在集群节点中启动集群服务

```
# pcs cluster start [--all] [node] [...]
```

认证组成集群的节点

- ▶ 在任意一个节点上验证所有的节点
- ▶ 使用前面设置的hacluster账户
- ▶ 注意iptables规则设置，

```
# pcs cluster auth [node] [...] [-u username] [-p password]
```

```
# pcs cluster auth node1 node2
```

```
Username: hacluster
```

```
Password: ****
```

```
node2: Authorized
```

```
node1: Authorized
```

配置和同步集群节点

- ▶ 创建配置文件/etc/corosync/corosync.conf :
 - ▶ 手工创建：参考example文件
 - ▶ 自动生成：通过pcs setup命令

```
# pcs cluster setup --name cluster1 node1 node2
Shutting down pacemaker/corosync services...
Redirecting to /bin/systemctl stop pacemaker.service
Redirecting to /bin/systemctl stop corosync.service
Killing any remaining services...
Removing all cluster configuration files...
node1: Succeeded
node2: Succeeded
Synchronizing pcsd certificates on nodes node1, node2...
node1: Success
node2: Success

Restarting pcsd on the nodes in order to reload the certificates...
node1: Success
node2: Success

# cat /etc/corosync/corosync.conf
```

考察操作的结果

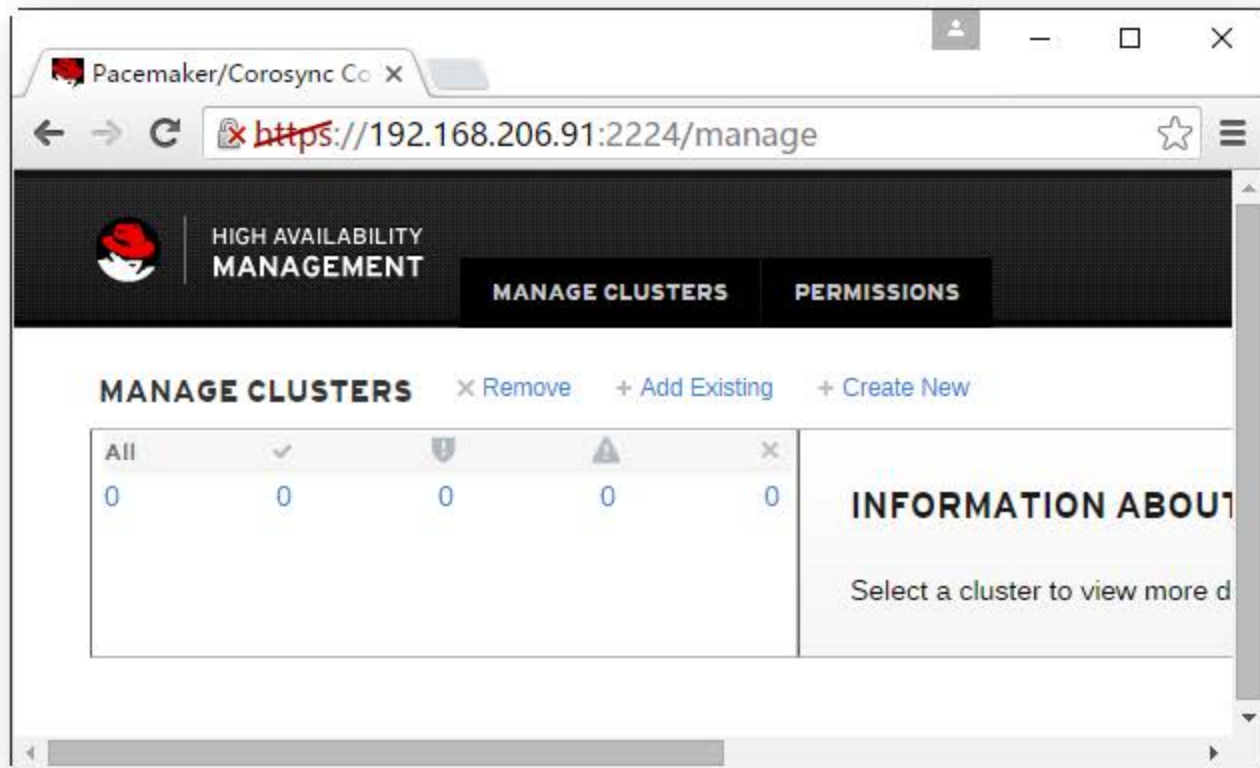
```
# cat /etc/corosync/corosync.conf
totem {
    version: 2
    secauth: off
    cluster_name: cluster1
    transport: udpu
}

nodelist {
    node {
        ring0_addr: node1
        nodeid: 1
    }

    node {
        ring0_addr: node2
        nodeid: 2
    }
}

quorum {
    provider: corosync_votequorum
    two_node: 1
}

logging {
    to_logfile: yes
    logfile: /var/log/cluster/corosync.log
    to_syslog: yes
}
```



启动集群

```
# pcs status
Error: cluster is not currently running on this node
# pcs cluster start --all
# pcs status
Cluster name: cluster1
WARNING: no stonith devices and stonith-enabled is not false
Last updated: Thu Sep 15 19:17:02 2016      Last change: Thu Sep 15 19:17:00 2016 by hacluster via
crmd on node1
Stack: unknown
Current DC: NONE
2 nodes and 0 resources configured

Node node1: UNCLEAN (offline)
Node node2: UNCLEAN (offline)

Full list of resources:

PCSD Status:
  node1: Online
  node2: Online

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```



总结

- ▶ 规划设计
- ▶ 操作系统安装
- ▶ 群集组件安装
- ▶ 群集节点准备
- ▶ 群集的创建