

ZettaStor DBS 产品安装 手册

鹏云网络

2015 年 6 月 17 日

目 录

第一章 产品概述	4
第二章 安装说明	4
第三章 环境准备	6
3.1 创建系统运行用户	6
3.2 防火墙	7
3.3 SSH 服务	8
3.4 修改内核参数	8
3.5 Jdk 安装	9
3.6 安装 Log4perl	10
3.7 安装 hdparm	10
3.8 配置/etc/hosts 文件	10
第四章 修改配置文件	11
4.1 系统部署规划	11
4.2 配置 config.properties 文件	11
4.3 配置 config/integtest_settings.xml 文件:	13
4.4 修改 config/ zookeeper.properties 配置信息	13
4.5 CentOS 系统 postgresql 数据库安装	14
4.6 Ubuntu 系统 postgresql 数据库安装	15
第五章 安装部署	16
5.1 安装系统所需要的基础包	16
5.2 部署 deployment_daemon	16
5.3 部署 zookeeper 服务	16
5.4 部署鹏云存储系统核心组件	16
第六章 系统重新部署	16
第七章 系统维护	20

编 注

编制人	联系电话	时间
扈宇春	13381056202	2015-6-17

第一章 产品概述

ZettaStor DBS 是软件定义的分布式存储。它运用分布式计算技术把大量标准 x86 服务器的存储介质进行聚合,将这些存储资源整合成为既具备传统 SAN / NAS 的企业级功能和特性,又具有高弹性、高扩展性、高可靠性的存储系统。可称做 Server SAN。

ZettaStor DBS 的构架是开放的,存储服务器和存储介质对应用是透明的,可以使用任何型号的标准服务器做为存储节点。服务器可以是专用的,也可以利用应用服务器闲置的存储介质构建低成本的存储系统。

ZettaStor DBS 提供高可用性和弹性扩展能力。33%的存储单元可以从系统中直接移除而不影响应用对存储的使用,或者导致数据的丢失。新增的存储单元会被自动识别、加入到存储系统,不会导致业务中断。

ZettaStor DBS 为应用提供高速的块设备接口,接口支持 iSCSI 和 NBD 协议,应用可以像访问本地硬盘一样访问存储系统提供的资源。OpenStack 和 Hadoop 也可以无缝地管理和使用鹏云块设备。

ZettaStor DBS 系统组件包括 InfoCenter、ControlCenter、DriverContainer、DataNode、DIH、Console、deployment_daemon 七个模块:

deployment_daemon 是其他各模块通信的基础;

DIH 监控管理所有服务的状态;

InfoCenter、ControlCenter、Console 支持热备方式,提供高可用性保证;

DriverContainer 根据系统目前的负载来分配网络驱动;

DataNode 把各个节点上的各类存储介质进行聚合,形成可统一管理的存储池,对外提供直接高性能的块设备接口;

Console 提供 Web 化的管理界面,用户可以通过 Web 界面实现对系统的管理。

第二章 安装说明

ZettaStor DBS 系统运行于 Linux 操作系统之上，在 Ubuntu12.04、CentOS6.6、CentOS7 上完整测试通过，本安装手册操作步骤基于 CentOS6.6、Ubuntu12.04 操作系统。

公司发布的最新的 ZettaStor DBS 系统安装包 pengyun-X.X.X-release。其中 X.X.X 是版本号。本文以 pengyun-1.0.0-internal 安装为例，对应其他版本，需把在对应位置的版本标识做替换。

安装完成后系统目录：

执行脚本	生成的一级目录	二级目录	说明
bin/deploy.pl	/var/deployment_daemon 具体创建的目录在 config.properties 设置 deployment.directory 参数的路径	tars	tar.gz 包的放置目录
		_packages	tar.gz 包的解压目录
		Packages	Copy tar.gz 的解压文件，是具体的工作目录。
bin/daemonClient.pl -c deploy:all	/var/testing	bin	不关注
		logs	不关注
		_packages	tar.gz 包的解压目录
		packages	Copy tar.gz 的解压文件，是具体的工作目录。

Package 下的三级目录	四五级目录	说明
/var/testing / packages /Pengyun-console	bin/shutdown.sh	停止脚本
	bin/startup.sh	启动脚本
	logs/ console.log	日志文件
	Status	状态
	PMPid	守护进程 ID
	SPid	进程 ID
/var/testing / packages /pengyun-controlcenter	bin/start_controlcenter.sh	启动脚本
	logs/ control-center.log	日志文件
	Status	状态
	PMPid	守护进程 ID
	SPid	进程 ID
/var/testing / packages /pengyun-coordinator	bin/start_drivercontainer.sh	启动脚本
	logs/ driver-container.log	日志文件
	Status	状态
	PMPid	守护进程 ID
	SPid	进程 ID
/var/testing / packages /pengyun-datanode	bin/start_datanode.sh	启动脚本
	logs/ datanode.log	日志文件

	Status	状态
	PMPid	守护进程 ID
	SPid	进程 ID
	/var/testing/packages/ pengyun-datanode/var/ storage/rawDisks	手动清除裸盘头字段 (page) 的元数据是, 用 dd 命令时设备的名称
/var/testing / packages /pengyun-infocenter	bin/start_infocenter.sh	启动脚本
	logs/ infomation-center.log	日志文件
	Status	状态
	PMPid	守护进程 ID
	SPid	进程 ID
/var/testing / packages /pengyun-instancehub	bin/stop_dih.sh	停止脚本
	bin/start_dih.sh	启动脚本
	logs/ dih.log	日志文件
	Status	状态
	PMPid	守护进程 ID
	SPid	进程 ID
/var/deployment_daemon / packages /pengyun-deployment_daemon	logs/ deployment-daemon.log	日志文件
	bin/startup.sh	启动脚本

第三章 环境准备

3.1 创建系统运行用户

在每台服务器, 创建新用户

- 1) 创建 “py_ops” 用户。

登陆系统, 在命令行下输入 “#useradd -m -s /bin/bash py_ops” ,

- 2) 设置修改密码

然后输入 “#passwd py_ops” 密码为 “312”。

- 3) 设置用户拥有 root 权限

在命令行下输入 “sudo vi /etc/passwd”, 找到用户 “py_ops” 所在的行, 将 uid 和 gid 改为 0。

备注：由于需要部署和安装多台机器，而部分和安装都是基于脚本来操作，所以建议将所有机器的用户名和密码统一。

3.2 防火墙

1) 开启防火墙，开放端口命令

Console 对最终管理用户提供服务器端 8080 端口；

ISCSI 驱动对外提供的服务 3260--3270；

NBD 驱动对外提供的服务 1234---1334；

deployment_daemon:	10002
DIH:	10000
InfoCenter	8020
ControlCenter	8010
DriverContainer	9000
DataNode	10011
Console	8080
zookeeper	2181, 3888, 2888

编辑防火墙配置文件：vi /etc/sysconfig/iptables

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 10002 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 10000 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8020 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8010 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9000 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 10011 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2181 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3888 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2888 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 1234 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 1235 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 1236 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3260 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3261 -j ACCEPT
```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3262 -j ACCEPT
```

重新启动防火墙：`service iptables restart`

2) 可以选择关闭防火墙

临时关闭命令：`service iptables stop`

永久关闭防火墙：

```
chkconfig iptables off
```

```
echo service iptables stop >> /etc/rc.local
```

检查命令在/etc/rc.local 的“exit 0”标识之前。

两个命令同时运行，运行完成后查看防火墙关闭状态

```
service iptables status
```

3.3 SSH 服务

所有系统提供 ssh 服务。如本机用 ssh 连接其它机器的 root 权限账户时，提示“permission denied”，请修改/etc/ssh/sshd_config 文件，PermitRootLogin without-password 改成 PermitRootLogin yes，重启 ssh 服务即可 `servicessh restart`。（Centos 不需要做任何变更）

3.4 修改内核参数

1) 修改内核参数

`vim /etc/sysctl.conf`，在文件末尾添加如下两行

```
net.ipv4.tcp_max_syn_backlog = 8192
```

```
#调整 SYN 队列的长度，修改可容纳等待连接的网络连接数
```

```
vm.max_map_count= 67107840
```

```
#调整虚拟内存
```

保存后执行 `sysctl -p` 生效

2) 调整 open files 值

the max of open files 调整,使用 `ulimit -a` 查看当前的 open files 值，默认是

1024。修改 open files 值为 65536

vim /etc/security/limits.conf, 在文件末尾添加:

```
* soft nofile 65536
* hard nofile 65536
root soft nofile 65536
root hard nofile 65536
py_ops soft nofile 65536
py_ops hard nofile 65536
```

需要重新启动服务器后生效。

3.5 Jdk 安装

- 1) 以 “py_ops” 登陆到系统;
- 2) 将 tools 中的 jdk-8u5-linux-x64.tar.gz 解压到 “/usr/local” 目录下;

```
tar xvf jdk-8u5-linux-x64.tar.gz
```

```
cp -r jdk1.8.0_05 /usr/local
```

- 3) 设置 jdk 环境变量

在命令行下输入 “vi /etc/profile”, 在文件中添加:

```
export JAVA_HOME=/usr/local/jdk1.8.0_05
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

备注: 在实施的时候注意小版本号对应。

- 4) 设置 root 下 java 可用

由于系统环境变量的继承关系, root 下使用 java 命令会出现“command not found”, 所以需要设置 root 下 java 可用。

在命令行下输入

```
update-alternatives --install "/usr/bin/java" "java" "$JAVA_HOME/jre/bin/java" 1
```

```
update-alternatives --install "/usr/bin/javac" "javac" "$JAVA_HOME/bin/javac" 1
update-alternatives --set java $JAVA_HOME/jre/bin/java
update-alternatives --set javac $JAVA_HOME/bin/javac
```

或者软链接:

```
ln -s /usr/local/jdk1.8.0_05/bin/jar /bin/jar
ln -s /usr/local/jdk1.8.0_05/bin/java /bin/java
ln -s /usr/local/jdk1.8.0_05/bin/javac /bin/javac
ln -s /usr/local/jdk1.8.0_05/bin/javah /bin/javah
ln -s /usr/local/jdk1.8.0_05/bin/javadoc /bin/javadoc
```

备注: 在实施的时候注意小版本型号对应。

3.6 安装 Log4perl

在每台服务器上安装 Log4perl:

```
sudo perl -MCPAN -e shell
install Log::Log4perl
install HTML::TokeParser::Simple
```

3.7 安装 hdparm

系统依赖 hdparm 命令去判断磁盘是 SSD 或 SATA 盘。hdparm 命令有可能在 DataNode 不存在, 需要安装 hdparm 软件包。将 tools 中的找到 hdparm-9.43-4.el6.x86_64.rpm 并执行。

```
rpm -ivh hdparm-9.43-4.el6.x86_64.rpm
```

3.8 配置/etc/hosts 文件

将系统所有服务器和 IP 地址的对应关系添加到所有服务器的/etc/hosts 文件中:

```
vi /etc/hosts
127.0.0.1    localhost
10.0.1.101  server1
10.0.1.102  server2
10.0.1.103  server3
10.0.1.104  server4
```

第四章 修改配置文件

4.1 系统部署规划

将 pengyun-1.0.0-release 安装包解压到一个目录，本例解压安装包到 /home/py_ops/pengyun-deploy 目录下，以下的部署建议 ControlCenter、InfoCenter、Console、Utils 在一台服务器，与 DataNode 节点分开部署，DriverContainer 部署在所有 DataNode 节点上。

应用	部署节点	备注
DIH	10.0.1.201: 10.0.1.204	需部署至所有节点，其中 DIH.center.host.list 部署在最小的 IP
InfoCenter	10.0.1.201	部署至某一节点，建议与 DataNode 分开，部署在不同节点上
ControlCenter	10.0.1.201	部署至某一节点，建议与 DataNode 分开，部署在不同节点上
Console	10.0.1.201	部署至某一节点，建议与 DataNode 分开，部署在不同节点上
Utils	10.0.1.201	部署至某一节点，建议与 DataNode 分开，部署在不同节点上
DataNode	10.0.1.202: 10.0.1.204	部署在每个存储节点上
DriverContainer	10.0.1.202: 10.0.1.204	需部署在某一个或多个节点上，此处部署在所有节点
deployment_daemon	10.0.1.201: 10.0.1.204	需部署至所有节点上

4.2 配置 config.properties 文件

编辑 config/deploy.properties 文件：

设置 remoter.user 为创建的用户“**py_ops**”，设置 remote.password 为密码“**312**”。

该 remoter.user 必须具有 root 权限。

设置 production.version 为 **release**。（和 pengyun-1.0.0-release 对应）

deployment.directory 的设置一般不做改动。

remote.network=**10.0.1.0/24** 设置对应系统部署的子网。

DIH 部署设置：

DIH.dir.name 为 pengyun-instancehub，

DIH.version 为 1.0.0

DIH.deploy.host.list 为 10.0.1.201:10.0.1.204, 表示 DIH 部署到的机器的范围从 201 到 204, 以 “:” 分割开始和结束, 为连续的 IP。

DIH.center.host.list = 10.0.1.201, 建议选择 DIH.deploy.host.list 中最小的 IP。

ControlCenter 部署设置:

ControlCenter.dir.name 设置为 pengyun-controlcenter

ControlCenter.version 设置为 1.0.0

ControlCenter.deploy.host 设置为 10.0.1.201

InfoCenter 设置:

InfoCenter.dir.name 设置为 pengyun-infocenter

InfoCenter.version 为 1.0.0

InfoCenter.deploy.host 设置为 10.0.1.201

DriverContainer 部署设置:

DriverContainer.dir.name 设置为 pengyun-drivecontainer

DriverContainer.version 设置 1.0.0

DriverContainer.deploy.host 设置为 10.0.1.202: 10.0.1.204

#DriverContainer 建议部署在每个 DataNode 节点上。

Console 的部署设置:

Console.dir.name 设置为 pengyun-console

Console.version 设置为 1.0.0

Console.deploy.host 设置为 10.0.1.201

Utils 的部署设置:

Utils.dir.name 设置为 pengyun-utils

Utils.version 设置为 1.0.0

Utils.deploy.host 设置为 10.0.1.201

Utils.remote.timeout=15000

DataNode 部署设置:

DataNode.dir.name 设置为 pengyun-datanode

DataNode.version 设置为 1.0.0

DataNode.deploy.host.lis 为 10.0.1.202:10.0.1.204

Deployment_daemon 设置:

deployment_daemon.dir.name 设置为 pengyun-deployment_daemon

deployment_daemon.version 设置为 1.0.0

```
deployment_daemon.deploy.host.list=10.0.1.201:10.0.1.204
```

4.3 配置 config/integtest_settings.xml 文件:

- 1) 修改 Segment Size 的大小

```
<property name="segment.size.byte" value="1073741824" />
```

安装包默认是 1G，可修改为 4G、8G、16G 等。

- 2) 修改 Center DIH 配置:

```
<property name="center.dih.endpoint" value="10.0.1.212:10000" />
```

- 3) 修改 ControlCenter 数据库连接配置文件

```
<property name="jdbc.url" value="jdbc:postgresql:
//10.0.1.212:5432/controlcenterdb" />    数据库的 IP 地址
```

- 4) 修改 infocenter 数据库连接配置文件

```
<property name="jdbc.url" value="jdbc:postgresql:
//10.0.1.201:5432/infocenterdb" />    数据库的 IP 地址
```

- 5) 修改完成后，为配置生效在部署时，在以下对话框输入 “y”

```
pengyun-deploy bin/daemonClient.pl -c deploy:all
*** Do you want to update configuration for all services?[y/n] y
```

4.4 修改 config/zookeeper.properties 配置信息

在 InfoCenter，ControlCenter 应用服务器上部署 zookeeper 服务。

- 1) 修改 InfoCenter 和 ControlCenter 操作系统的用户、密码

```
remote.user=py_ops
remote.password=312
```

- 2) 在文件末尾列举 InfoCenter 和 ControlCenter 服务器

```
server.1=10.0.1.201
server.2=10.0.1.202
server.3=10.0.1.203
server.4=10.0.1.204
```

4.5 CentOS 系统 postgresql 数据库安装

- 1) 查看 Postgres 数据库是否已被安装

```
rpm -qa | grep postgres
```

- 2) CentOS6.6 已经存在版本较低的 Postgres 数据库，卸载。

```
rpm -e postgresql-devel-8.4.20-1.el6_5.x86_64
```

```
rpm -e postgresql-8.4.20-1.el6_5.x86_64
```

```
rpm -e postgresql-libs-8.4.20-1.el6_5.x86_64
```

- 3) yum 安装（联网并配置 DNS）

```
yum install
```

```
http://yum.postgresql.org/9.3/redhat/rhel-6-x86\_64/pgdg-redhat93-9.3-1.noarc  
h.rpm
```

```
yum install postgresql93-server postgresql93-contrib
```

- 4) 查看成功安装的包

```
# rpm -qa | grep postgres
```

```
postgresql93-contrib-9.3.5-1PGDG.rhel6.x86_64
```

```
postgresql93-libs-9.3.5-1PGDG.rhel6.x86_64
```

```
postgresql93-server-9.3.5-1PGDG.rhel6.x86_64
```

```
postgresql93-9.3.5-1PGDG.rhel6.x86_64
```

- 5) 初始化数据库

```
service postgresql-9.3 initdb
```

- 6) 启动 Postgres 数据库

```
service postgresql-9.3 start 或者/etc/init.d/postgresql-9.3 start
```

- 7) 修改配置文件，并创建用户、数据库并进行授权

```
vi /var/lib/pgsql/9.3/data/postgresql.conf 修改配置文件
```

```
listen_addresses = '*' 删除此行前面的#号，并将 localhost 改成*
```

```
vi /var/lib/pgsql/9.3/data/pg_hba.conf 修改配置文件（Centos）
```

```
host all all 0.0.0.0/0 md5 添加到配置文件尾
```

```
service postgresql-9.3 restart 服务重启
```

```
su - postgres; psql 登录 Postgres 数据库
```

create user py with password '312';	创建用户 py
create database infocenterdb owner py;	创建数据库 infocenterdb
create database controlcenterdb owner py;	创建数据库 controlcenterdb
grant all privileges on database infocenterdb to py;	授权
grant all privileges on database controlcenterdb to py;	授权

4.6 Ubuntu 系统 postgresql 数据库安装

1) 检查是否安装:

```
dpkg --status postgresql
dpkg --status postgresql-client
```

2) 安装步骤:

apt-get install -y postgresql	安装 Postgres 服务器端
apt-get install -y postgresql-client	安装 Postgres 客户端
vi /etc/postgresql/9.1/main/postgresql.conf	修改配置文件
listen_addresses = '*'	删除此行前面的#号, 并将 localhost 改成*
/etc/postgresql/9.1/main/pg_hba.conf	修改配置文件
host all all 0.0.0.0/0 md5	添加到配置文件尾
/etc/init.d/postgresql restart	服务重启
Psql	登录 Postgres 数据库
create user py with password '312';	创建用户 py
create database infocenterdb owner py;	创建数据库 infocenterdb
create database controlcenterdb owner py;	创建数据库 controlcenterdb
grant all privileges on database infocenterdb to py;	授权
grant all privileges on database controlcenterdb to py;	授权

3) 修改配置文件, 并创建用户、数据库并进行授权

vi /etc/postgresql/9.1/main/postgresql.conf	修改配置文件 (Ubuntu)
listen_addresses = '*'	删除此行前面的#号, 并将 localhost 改成*
vi /etc/postgresql/9.1/main/pg_hba.conf	修改配置文件 (Ubuntu)

第五章 安装部署

5.1 安装系统所需要的基础包

在部署包目录下，输入“bin/make-perl-lib.pl”，便开始安装系统所需要的基础包，就是安装 lib_src 目录下的 gz 包。

5.2 部署 deployment_daemon

在部署包目录下，输入“bin/deploy.pl”。

5.3 部署 zookeeper 服务

在部署包目录下，输入“bin/zookeeper.pl”。

5.4 部署鹏云存储系统核心组件

在部署包目录下，输入“bin/daemonClient.pl -c deploy:all”，便开始按照配置部署系统到所有的机器上。

配置文件生效：

```
#bin/daemonClient.pl -c deploy:all
```

```
*** Do you want to update configuration for all services?[y/n] y
```

第六章 设备挂载

6.1 NBD 安装挂载

在卷读取客户端安装 NBD Server，NBD Client。

1) NBD 安装（Ubuntu）

```
sudo apt-get install nbd-client nbd-server  
sudo modprobe nbd  
sudo lsmod | grep nbd
```

2) nbd server 安装（CentOS）

```
yum install kernel-devel kernel-headers
```



```

cd /tmp
  groupadd builder
  useradd builder -g builder
rpm -ivh kernel-2.6.32-220.el6.src.rpm(安装包在 tools 文件夹中)
cd ~/rpmbuild/SOURCES
tar jxf linux-2.6.32-220.el6.tar.bz2 -C /usr/src/kernels/
cd /usr/src/kernels
mv $(uname -r) $(uname -r)-old
mv linux-2.6.32-220.el6 $(uname -r)
cd $(uname -r)
make mrproper
*****
find / -name Module.symvers
cp *-old/Module.symvers .
*****
cp /boot/config-$(uname -r) /.config
make oldconfig
make prepare
make scripts
make CONFIG_BLK_DEV_NBD=m M=drivers/block
cp drivers/block/nbd.ko /lib/modules/$(uname -r)/kernel/drivers/block/
depmod -a
modprobe nbd

```

3) NBD Client 安装 (CentOS)

```

#tar xvf nbd-3.8.tar.xz
#configure
#make
#make install

```

4) 检查 NBD server 是否正确安装

```

# ls /dev/nbd* -hl
brw-r----- 1 root disk 43, 0 Jul 27 06:40 /dev/nbd0
brw-r----- 1 root disk 43, 1 Jul 27 06:40 /dev/nbd1
brw-r----- 1 root disk 43, 2 Jul 27 06:40 /dev/nbd2
brw-r----- 1 root disk 43, 9 Jul 27 06:40 /dev/nbd9

```

5) 修改 I/O 调度算法

```

echo deadline > /sys/block/nbdX/queue/scheduler

```

6) 将/dev/nbdX 设备与主机连接.

```

# nbd-client 10.0.1.209 1234 /dev/nbd1

```

10.0.1.209 1234 为 NBD 服务端的 IP 和端口，即在 ZettaStor 的 Console 界面上挂载 NBD 时显示的 IP 地址和端口，bs 用于设置块大小，默认是 1024，可以是

512、1024、2048、4096, nbd_device 映射到本地的哪个 nbd 设备(如: /dev/nbd0)。

7) 连接成功后,可以查看到 nbd-client 进程。

```
# ps -ef | grep nbd
root 3156 1 0 06:44 pts/0 00:00:00 nbd-client 192.168.1.1 1234 /dev/nbd0
```

8) 格式化块设备, 使用前需要格式化这个块设备

```
# mkfs.ext4 /dev/nbd0
```

9) 挂载这个块设备.

```
# mkdir /mnt/nbd0          #创建挂载目录
# mount /dev/nbd0 /mnt/nbd0 #把块设备挂载在/mnt/nbd0 目录下
```

6.2 ISCSI 挂载

1) 前提: 在部署所有 DriverContainer 的机器上都应该安装 nbd server, nbd client。

2) 启动 iscsitarget

```
#/etc/init.d/iscsi-target start(CentOS)
#/etc/init.d/open-iscsi start (Ubuntu)
```

3) 启动 nbd

```
#modprobe nbd
```

4) 发现 iscsi target

```
sudo iscsiadm -m discovery -t sendtargets -p 10.0.1.212:3260
```

显示信息如下:

```
192.168.35.17:3260,1 iqn.2013-02.node2:1G
```

5) 连接 target

```
sudo iscsiadm -m node --targetname iqn.2013-02.node2:1G -p 192.168.35.17:3261 --login
```

执行上述操作后, 就可在本地主机上多出一个/dev/sdx

如果要删除使用: sudoiscsiadm -m node --targetname iqn.2013-02.node2 -p 192.168.35.17 -u

第七章 系统重新部署

需要将上次安装系统运行的进程停掉, 创建的数据库表删除掉, 安装的文件包删除掉。在这里介绍一个批量处理的工具 vmtools, 通过三个不同的参数完成系统的清除工作。

1) 修改或新增 vmtools 配置文件

修改或新增 toolsvmtools 的配置文件 wipeout/resources/config/deploy.conf 红色字体部分，将所有的 host 的 IP 地址都添加进去。

```
[vm1]
ip.address=10.0.1.201          服务器 IP 地址
host.ip.address=10.0.1.201    服务器 IP 地址
guest.available.disks=sdb

[vm2]
ip.address=10.0.1.202          服务器 IP 地址
host.ip.address=10.0.1.202    服务器 IP 地址
guest.available.disks=sdb
```

2) 执行 vmtools.pl，清除旧系统信息

```
#zookeeper/zookeeper.pl
#./vmtools.pl wipeout all    停掉现在正在运行的进程，并删除已经部署
                             的文件包
#./vmtools.pl cleandb 【主机名】 删除已经创建的 Postgres 数据库的
                             表。
#./vmtools.pl wipeoutdisk all 清除存放在裸盘头部的元数据
```

3) 安装新系统

```
cd pengyun-deploy 安装包目录里
#bin/deploy.pl      重新部署 deployment_daemon
#bin/zookeeper      部署并启动 zookeeper
#bin/daemonClient.pl -c deploy:all 重新部署系统的其他组件。
```

手工清除:

但适用于对特定服务器进行调试。分别登录到所有 Host 进行操作:

- 1) 删除/var/testing和/var/deployment_daemon目录
- 2) 停掉所有的java进程：#kill -9 java

第八章 系统维护

- 1) 通过 Console 的“服务管理”菜单查看、停止、启动各服务组件

服务代表着一个应用，例如：DataNode, ControlCenter, InfoCenter等。

服务名↑	服务ID↑	组ID↑	所在主机↑	所在端口↑	服务状态↑	操作
DIH	1129061924047671322		10.0.1.212	10000	OK	停止
DataNode	1285977971899062022	0	10.0.1.209	10011	OK	停止
DriverContainer	1444203269584569956		10.0.1.212	9000	OK	停止
InfoCenter	1916842860584693656		10.0.1.211	8020	OK	停止
DIH	245722447488956489		10.0.1.210	10000	OK	停止
DataNode	2526946231486659739	2	10.0.1.210	10011	OK	停止
DIH	2765569648483355662		10.0.1.211	10000	OK	停止
ControlCenter	3053060083777236121		10.0.1.212	8010	SUSPEND	停止
DIH	3342362720625457254		10.0.1.209	10000	OK	停止
DataNode	349562949011068194	1	10.0.1.212	10011	OK	停止
InfoCenter	3866307641996347503		10.0.1.212	8020	SUSPEND	停止
ControlCenter	505422520292520824		10.0.1.211	8010	OK	停止
DataNode	855764629826121689	0	10.0.1.211	10011	OK	停止

- 2) 通过命令查看、停止、启动或重新部署各服务组件

针对在 DIH、InfoCenter、ControlCenter、DriverContainer、Console、DataNode 单个应用在部署过程中或部署完成后需要对某个应用进行关掉、重启或重新部署等操作。每台机器上安装了守护进程 deployment_daemon 后，可以对单个应用进行管理和操作。

语法：

```
bin/daemonClient.pl -c/--command command:service -p/--pToDeployHost host
```

例子：

```
bin/daemonClient.pl -c deploy:all
```

```
bin/daemonClient.pl -c activate:DIH -p 10.0.1.201
```

参数含义：

activate: 重新部署该应用，原来的安装包将会被覆盖。

deactivate: 同步 shutdown，在 shutdown 前应用会处理为完成的数据

destroy: 强制 shutdown，不会等待未处理的数据处理完

start: 启动应用

restart: 重启服务，重启前应用会把未处理晚的数据处理完。

wipeout: kill 掉所有的应用，并删除的安装目录下的所有文件，完全擦除。

status: 查看应用的状态