ZettaStor 产品
安装手册

（针对 CENTOS7 操作系统）

鹏云网络

目录

编制人	时间	备注
扈宇春	2015-6-17	初稿
马海清	2016-3-5	修改为针对 CentOS7 系统的安装手册
马海清	2016-8-5	将 py_nbd 更新为 pyd，增加 NTP 的配置
陆娜	2016-10-21	添加 MonitorCenter 部分

第一章 产品概述

ZettaStor 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 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 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 为应用提供高速的块设备接口，接口支持 ISCSI 和 PYD 协议，应用可以如同访问本地硬盘一样访问存储系统提供的资源。ZettaStor 系统可以与 OpenStack、Hadoop、VMware 和 fusion computer 进行无缝对接。

ZettaStor 系统组件包括 InfoCenter、ControlCenter、MonitorCenter、DriverContainer、DataNode、DIH、Console、deployment_daemon 八个模块：

- deployment_daemon 是其它各模块通信的基础；
- DIH 监控管理所有服务的服务状态；
- InfoCenter、ControlCenter、MonitorCenter、Console 支持热备方式（可以部署多个），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 系统运行于 Linux 操作系统之上，在 redhat6、redhat7、Ubuntu12.04、CentOS6.6、CentOS7 上完整测试通过，本安装手册操作步骤基于 CentOS7 操作系统来详细说明一下软件安装部署的过程，对于其它版本的 linux 请根据相应的步骤来执行命令（不同的 linux 版本同一功能的命令可能不一样）。

公司发布的 ZettaStor 系统安装包的文件名格式为 pengyun-X.X.X-rele-ase.tar.gz，其中 X.X.X 是版本号。本文以默认版本 pengyun-1.0.0 - release.tar.gz 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。（注：对外发布版本为 2.0 以上）

ZettaStor 安装完成后系统目录会产生一级目录和二级目录，详情可查询附录一。

第三章 环境准备

3.1 硬件配置

- 基于 X86 构架的标准服务器
- 四核以上 CPU
- 32G 以上内存
- 1GB/10GB/Infiniband 网卡
- SATA/SAS/HBA 磁盘控制器，至少一块磁盘
- 2 块 128GB 以上 SSD
- 最小部署规模为 3 台主机

3.2 操作系统

redhat6、redhat7、Ubuntu12.04、CentOS6.6、CentOS7、SUSE11

3.3 软件清单

1) 部署 ZettaStor 系统使用的软件清单和版本信息，对于所列出的软件包，应使用版本不低于此版本的软件包：

1	kernel-headers	kernel-headers-3.10.0-229.el7.x86_64.rpm
2	kernel-devel	kernel-devel-3.10.0-229.el7.x86_64.rpm
3	make	make-3.82-21.el7.x86_64.rpm
4	gcc	gcc-4.8.3-9.el7.x86_64.rpm
5	gcc-c++	gcc-c4.8.3-9.el7.x86_64.rpm
6	flex	flex-2.5.37-3.el7.x86_64.rpm
7	patch	patch-2.7.1-8.el7.x86_64.rpm
8	glibc	glibc-2.17-78.el7.x86_64.rpm
9	zlib	zlib-1.2.7-13.el7.x86_64.rpm
10	hdparm	hdparm-9.43-5.el7.x86_64.rpm
11	uuid	uuid-1.6.2-26.el7.x86_64.rpm

12	iscsitarget	iscsitarget-1.4.20.3+svn499.tar.gz
13	JDK	Jdk1.8.0_65 或以上
14	数据库	Postgresql9.3 或以上

2）部分软件的安装详解：

- **JDK 的安装**

目前的 ZettaStor 系统使用的 jdk 版本为 1.8，这里我们推荐使用的版本为 1.8.0_65。其安装步骤如下：

Step1: 将 jdk-8u5-linux-x64.tar.gz 解压到/usr/local 目录下，命令如下：

tar -zxvf jdk-8u5-linux-x64.tar.gz -C /usr/local

Step2: 设置 jdk 环境变量

在/etc/profile 文件中添加如下内容

JAVA_HOME=/usr/local/jdk1.8.0_65
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
JRE_HOME=/usr/local/jdk1.8.0_65/jre
PATH=$PATH:$HOME/bin:$JRE_HOME/bin
export JAVA_HOME
export JRE_HOME

Step3: 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：

alternatives --install "/usr/bin/java" "java" "/usr/local/jdk1.8.0_65/jre/bin/java" 1
alternatives --install "/usr/bin/javac" "javac" "/usr/local/jdk1.8.0_65/bin/javac" 1
alternatives --set java /usr/local/jdk1.8.0_65/jre/bin/java
alternatives --set javac /usr/local/jdk1.8.0_65/bin/javac

安装完毕之后使用“alternatives --config java“命令确认使用的 java 为我们安装的 java，而不是系统自带的 java 版本。

（注意： 在安装的时候并不一定需要 1.8.0_65 版本，对于高版本的只需要在操作过程中将对应的小版本号修改一下即可。）

- **数据库的安装**

在 ZettaStor 产品中，使用的数据库为 postgresql。一般情况下来说，我们只需要在部署数据库的节点机器上安装此软件包。

1）执行脚本 install-postgresql.sh；

2）修改配置文件，并创建用户、数据库并进行授权

Step1：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*

Step2：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

Step3：service postgresql-9.3 restart　　服务重启

Step4：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

create database monitorcenterdb owner py;　　创建数据库 monitorcenterdb

grant all privileges on database infocenterdb to py;　　授权

grant all privileges on database controlcenterdb to py;　　授权

grant all privileges on database monitorcenterdb to py;　　授权

3）部署 ZettaStor 系统前必须打开数据库；

查看数据库状态：systemctl status postgresql-9.3.service -l

Failed：关闭状态，active(running)：开启状态；

打开数据库：systemctl start postgresql-9.3.service

3.4 系统优化

- **防火墙的设置：**

在部署 ZettaStor 系统的节点机器上需要开放服务所需要的端口。考虑到部署 ZettaStor 的节点机器一般都在一个单独的网络里面或者与因特网是隔离的，所以简单的方式就是将防火墙关闭。

1）在 CentOS7 系统中，关闭防火墙的命令如下：

systemctl stop firewalld.service

systemctl disable firewalld.service

查看此时防火墙的状态（确认是否已经关闭），命令如下：

systemctl status firewalld.service

2）也可以使用如下方式在开始防火墙的状态下开放端口。

系统所需的模块端口如下:

模块	端口
deployment_daemon	10002
DIH	10000
InfoCenter	8020
ControlCenter	8010
MonitorCenter	11000
DriverContainer	9000
DataNode	10011
Console	8080
Zookeeper	2181, 3888, 2888
ISCSI 驱动	3260--3270
PYD 驱动	1234--1334

编辑防火墙配置文件，开启防火墙，开放端口命令：

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

编辑完后，重新启动防火墙：service iptables restart

■ 内核参数的设置

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

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 查看当前允许的最大打开文件数目，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

（注：此项修改需要重新启动服务器后生效）

■ NTP 的设置

我们选择系统中的某些机器作为时间服务器，所有其它的机器均向时间服务器同步时间。

1) 首先检查节点上是否可以执行 ntpdate 这个命令，如果不能执行的话，需要安装软件包。在联网或者设置安装源的情况下可以使用命令 yum install ntp 来进行安装。

2）安装完毕之后，检查服务的状态 service ntp status 或者/etc/init.d/ntp status。

3）修改配置文件：/etc/ntp.conf

修改允许其他主机通过本 ntp 服务器查询、同步 配置文件中默认只允许本机进行查询，如下

restrict 127.0.0.1 restrict ::1

将上面的内容按照下面的格式进行修改：

restrict [ip] mask [mask] [parameter]

例如 restrict 10.0.1.0 mask 255.255.255.0 nomodify notrap

部分参数的含义如下：

nomodify：不允许客户端修改服务器的时间参数，但是允许客户端透过这部主机进行时间校验；

noquery:不允许客户端进行时间校验（个人不建议使用此参数）；

Notrap: 不提供 trap 时间登录；

notrust: 拒绝没有认证的客户端。

4）修改完毕之后将服务重新启动一下：

service ntp restart 或者 /etc/init.d/ntp restart

5）在其它主机上客户机一端，编辑文件 vi /etc/crontab（修改/etc/crontab 这种方法只有 root 用户能用）

比如每隔 2 小时进行时间同步一次：0 */2 * * * /usr/sbin/ntpdate 10.0.1.207

3.5 ZettaStor 系统配置

◆ 配置 hostname 和 hosts

在日常的使用中，我们习惯对每个机器进行编号，省去记忆 IP 地址的烦恼。编辑/etc/hostname 文件，设置对应的别名即可，通常使用的名字为 server**;

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

◆ 启动脚本的配置

1）将 ZettaStor 文件夹中的 pengyun-sys 放置到/etc/init.d/文件夹下面，并使用命令 chmod a+x /etc/init.d/pengyun-sys 赋予正确的权限。

2) 然后编辑文件/etc/rc.d/rc.local：

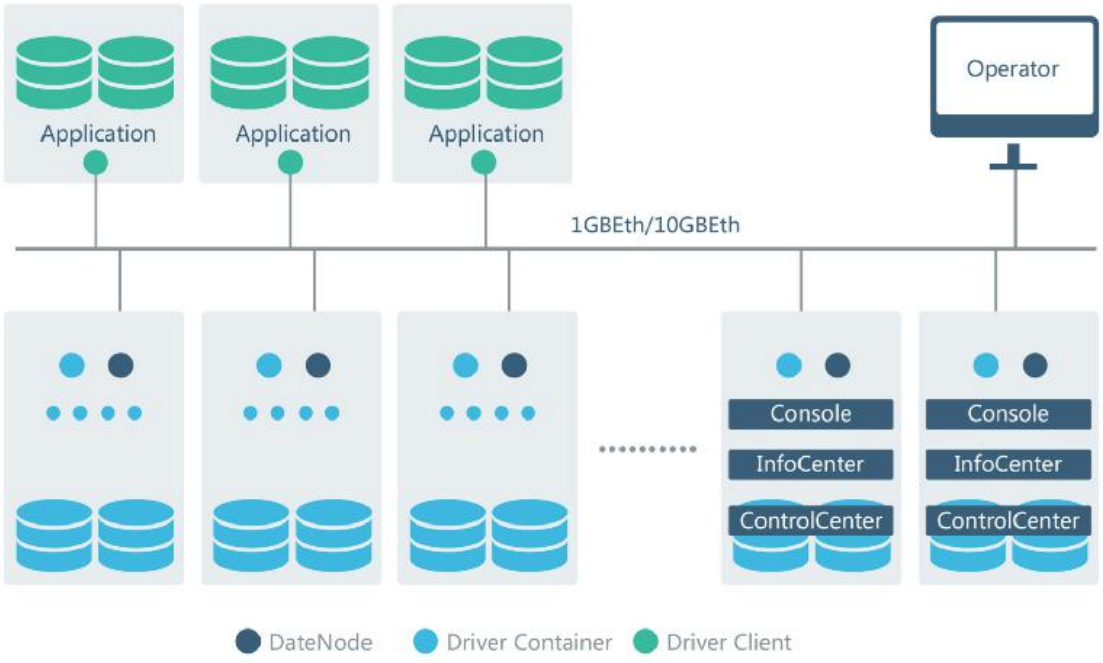
在文件中添加/usr/bin/perl /etc/init.d/pengyun-sys 实现在启动过程中调用。

启动脚本配置完毕之后，需要重启机器确认配置可以生效。

第四章 ZettaStor 配置文件的修改

4.1 系统部署规划

ZettaStor 系统存储节点上需部署 DataNode，DriverContainer 可以以集群方式部署在多个存储节点上；Console、InfoCenter、ControlCenter 和 MonitorCenter 支持热备方式，可以部署在专用服务器上，也可部署在存储服务器上。



❖ 实例

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、MonitorCenter、Console、Utils 在一台服务器，与 DataNode 节点分开部署，DriverContainer 按需求部署一个或多个节点。“:”表示连续的 IP 地址，“,”用来间隔不连续的 IP 地址，如“10.0.1.201,10.0.1.203”。

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	部署至某一台节点，建议与 DataNode 分开，部署在不同节点上
ControlCenter	10.0.1. 201	部署至某一台节点，建议与 DataNode 分开，部署在不同节点上
MonitorCenter	10.0.1.201	部署至某一台节点，建议与 DataNode 分开，部署在不同节点上
Console	10.0.1. 201	部署至某一台节点，建议与 DataNode 分开，部署在不同节点上
Utils	10.0.1. 201	部署至某一台节点，建议与 DataNode 分开，部署在不同节点上
DataNode	10.0.1.202:10.0.1.204	至少部署在三个存储节点上
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 DataNode 节点
deployment_daemon	10.0.1.201:10.0.1.204	需部署至所有节点上

4.2 修改配置文件

按照 ZettaStor 系统部署规划编辑三个配置文件，分别为 config/deploy.properties、config/integtest_settings.xml 和 config/zookeeper.properties。

注：红色字体是要修改的地方。

◆ 编辑 config/deploy.properties 文件：

主要描述各个服务名称，版本，所在的节点信息，服务端口，以及服务部署的时长。

```
### global constants ###
thrift.transport.timeout=10000
thrift.transport.maxsize =10000000
deployment.thread.amount=1
daemon.port =10002
production.version=release      #和 pengyun-1.0.0-release 对应
remote.network=10.0.1.0/24    #设置对应系统部署的子网
remote.user=py_ops              #remoter.user 必须具有 root 权限
remote.password=312             #密码自行修改

# specify the absolute path
deployment.directory=/var/deployment_daemon

### Instance hub has to be deployed to all hosts listed in this configuration file
DIH.dir.name=pengyun-instancehub
DIH.version =1.0.0
DIH.deploy.host.list=10.0.1.201:10.0.1.204  #表示 DIH 部署的范围从 201 到 204，冒号表示
                                            连续，分隔使用逗号
DIH.deploy.port=10000
DIH.deploy.agent.jmx.port=11000
DIH.center.host.list=10.0.1.201     #建议选择 DIH.deploy.host.list 中最小的 IP
DIH.remote.timeout=60000

### infocenter port: 8020 ###
InfoCenter.dir.name=pengyun-infocenter
InfoCenter.version=1.0.0
InfoCenter.deploy.host.list=10.0.1.201
InfoCenter.deploy.port=8020
InfoCenter.deploy.agent.jmx.port=8120
InfoCenter.remote.timeout=60000

### controlcenter port: 8010 ###
ControlCenter.dir.name=pengyun-controlcenter
ControlCenter.version=1.0.0
ControlCenter.deploy.host.list=10.0.1.201
ControlCenter.deploy.port=8010
```

```
ControlCenter.deploy.agent.jmx.port=8110
ControlCenter.remote.timeout=60000

### monitorcenter port: 8030 ###
MonitorCenter.dir.name=pengyun-system_monitor
MonitorCenter.version=1.0.0
MonitorCenter.deploy.host.list=10.0.1.201
MonitorCenter.deploy.port=8030
MonitorCenter.deploy.agent.jmx.port=8130
MonitorCenter.remote.timeout=60000

### coordinator port: 9000 ###
DriverContainer.dir.name=pengyun-drivecontainer
DriverContainer.version =1.0.0
DriverContainer.deploy.host.list=10.0.1.202:10.0.1.204
DriverContainer.deploy.port=9000
DriverContainer.deploy.agent.jmx.port=9100
DriverContainer.remote.timeout=60000

DataNode.dir.name=pengyun-datanode
DataNode.version=1.0.0
DataNode.deploy.host.list=10.0.1.202:10.0.1.204
DataNode.deploy.port=10011
DataNode.deploy.agent.jmx.port=11011
DataNode.remote.timeout=600000
DataNode.initArchives=true

DataNode.testingmode=false
DataNode.teststation.number_raw_disks=2
DataNode.teststation.volume_size_in_mb=32
DataNode.teststation.segment_size_in_mb=16
DataNode.teststation.raw_disk_size_in_mb=36
DataNode.teststation.ram_disk_size_in_mb=36

### ScriptContainer ###
#ScriptContainer.dir.name=pengyun-scriptcontainer
#ScriptContainer.version =1.0.0
#ScriptContainer.deploy.host.list=10.0.1.255
```

```
#ScriptContainer.deploy.port=9090
#ScriptContainer.deploy.agent.jmx.port=9190
#ScriptContainer.remote.timeout=60000

### Utils ###
Utils.dir.name=pengyun-utils
Utils.version=1.0.0
Utils.deploy.host.list=10.0.1.201
Utils.remote.timeout=15000

### Console ###
Console.dir.name=pengyun-console
Console.version=1.0.0
Console.deploy.host.list=10.0.1.201
Console.deploy.port=8080
Console.deploy.agent.jmx.port=8180
Console.remote.timeout=60000

deployment_daemon.dir.name=pengyun-deployment_daemon
deployment_daemon.version =1.0.0
deployment_daemon.deploy.host.list=10.0.1.201:10.0.1.204
deployment_daemon.deploy.specified=false
deployment_daemon.deploy.port=10002
Deployment_daemon.deploy.agent.jmx.port=11002
deployment_daemon.remote.timeout=60000
```

◆ 编辑 config/integtest_settings.xml 文件

必须修改的四个地方：

1) 修改 centerDIH 配置：

```
<property name="center.dih.endpoint"
    value="10.0.1.201:10000" />
```

2) 修改 controlcenter 数据库连接配置文件

```
<property name="jdbc.url" value="jdbc:postgresql:
//10.0.1.201:5432/controlcenterdb" />
```

数据库的 IP 地址

3) 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

4) 修改 monitorcenter 数据库连接配置文件

<property name="jdbc.url" value="jdbc:postgresql:

//10.0.1.201:5432/monitorcenterdb" /> 数据库的 IP 地址

按实际需求修改：

a. 根据实际需求修改 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。

b. 根据实际需要的节点个数修改 pengyun-infocenter 项目中文件 infocenter.properties 中 group.count 属性的值：

<property name="group.count" value="4" /> (最小等于 3)

修改完成后，为使配置生效在部署时，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] ： 输入 y

◆ 编辑 config/zookeeper.properties 文件

在 InfoCenter、ControlCenter、MonitorCenter 和 Console 这些支持热备方式（即可以部署多个）的应用服务器上部署 zookeeper 服务。

1) 修改 InfoCenter、ControlCenter、MonitorCenter 和 Console 操作系统的用户、密码。（注： 该 remoter.user 必须具有 root 权限）

remote.user=py_ops

remote.password=312

2) 在文件末尾列举 InfoCenter、ControlCenter、MonitorCenter 和 Console 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

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 ZettaStor DBS 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。

为使配置文件生效：

#bin/daemonClient.pl -c deploy:all

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

✧ 部署成功提示信息：Successfully to complete operation DEPLOY on service [DIH, InfoCenter, ControlCenter, Console, MonitorCenter, DataNode, DriverContainer](最后一行)。

5.5 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、MonitorCenter、 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/--p ToDeployHost host

✧ 实例：(在部署目录下执行)

bin/daemonClient.pl -c status: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。

5.6 登录验证

打开浏览器，地址一栏输入 http://【Console 的 IP 地址】:8080

初始用户名和密码都为：admin，输入后成功登录 ZettaStor 系统。

附录一

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/ controlcenter.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/drivercontainer.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/infocenter.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_daemo-n.log	日志文件
	bin/startup.sh	启动脚本