|  |
| --- |
| **北京易腾时代科技有限公司** |
| CentOS系统环境部署 |
|  |
|  |
|  |
| **王笙光** |
| **4/10/2014** |
|  |

168

北京易腾时代科技有限公司

CentOS环境部署指南

版本V1.0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本历史 | | | | |
| 版本号 | 修订日期 | 修订人 | 变更点 | 最后一次修改时间 |
| V1.0 | 2014-04-10 | 王笙光 | Init | 2014-04-10 |
|  |  |  |  |  |

目录

[TSingTao系统环境部署 5](#_Toc384890103)

[1. 简介 5](#_Toc384890104)

[2. 操作系统安装 5](#_Toc384890105)

[2.1. 安装流程 6](#_Toc384890106)

[2.1.1. 使用光盘安装 6](#_Toc384890107)

[3. 操作系统优化/安装组件 20](#_Toc384890108)

[3.1. 修改网卡配置 20](#_Toc384890109)

[3.2. 更新源 21](#_Toc384890110)

[3.3. 安装组件 22](#_Toc384890111)

[3.4. 系统优化 22](#_Toc384890112)

[4. 中间件安装 23](#_Toc384890113)

[4.1. 目录结构 25](#_Toc384890114)

[4.2. 安装JDK 25](#_Toc384890115)

[4.3. 安装Tomcat 26](#_Toc384890116)

[~~4.4.~~ ~~安装Terracotta~~不再使用 29](#_Toc384890117)

[4.5. 安装Nginx 30](#_Toc384890118)

[5. 系统部署 36](#_Toc384890119)

[5.1. 端口配置规范 36](#_Toc384890120)

[5.2. 注意事项 37](#_Toc384890121)

[5.3. 组件部署 38](#_Toc384890122)

[5.3.1. REP报表服务器部署 38](#_Toc384890123)

[5.3.2. LMS日志服务器部署 38](#_Toc384890124)

[5.4. 应用部署 41](#_Toc384890125)

[6. 部分命令 42](#_Toc384890126)

[6.1. 查看操作系统版本号 42](#_Toc384890127)

[6.2. 常用命令 42](#_Toc384890128)

[6.3. 查看系统信息 42](#_Toc384890129)

[6.4. 压缩和解压 43](#_Toc384890130)

[6.5. 查看磁盘空间 43](#_Toc384890131)

[6.6. 挂载磁盘 43](#_Toc384890132)

[7. FAQ 44](#_Toc384890133)

CentOS系统环境部署

# 简介

注意！中间件版本参考实际安装的版本，安装过程中注意文件路径。

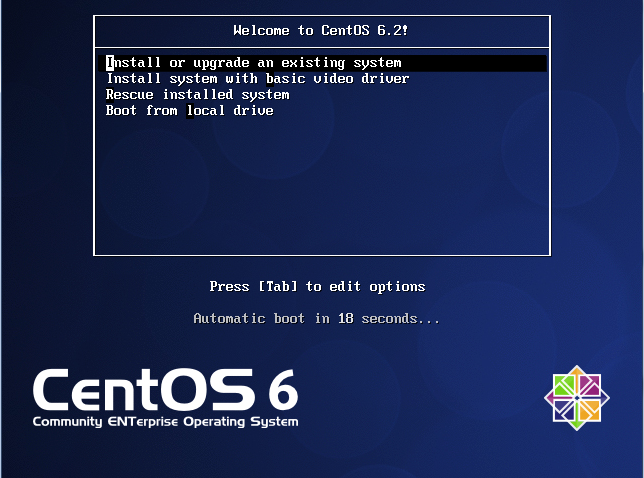
* 操作系统：CentOS 6.2
* JDK：jdk-6u45
* Tomcat：7.0.40
* ~~Terracotta：3.7.5~~
* Nginx：1.4.1
* REP：
* LMS：

# 操作系统安装

本例安装的是32位版本。

## 安装流程

### 使用光盘安装



**选择第一项Install or upgrade an existing system**

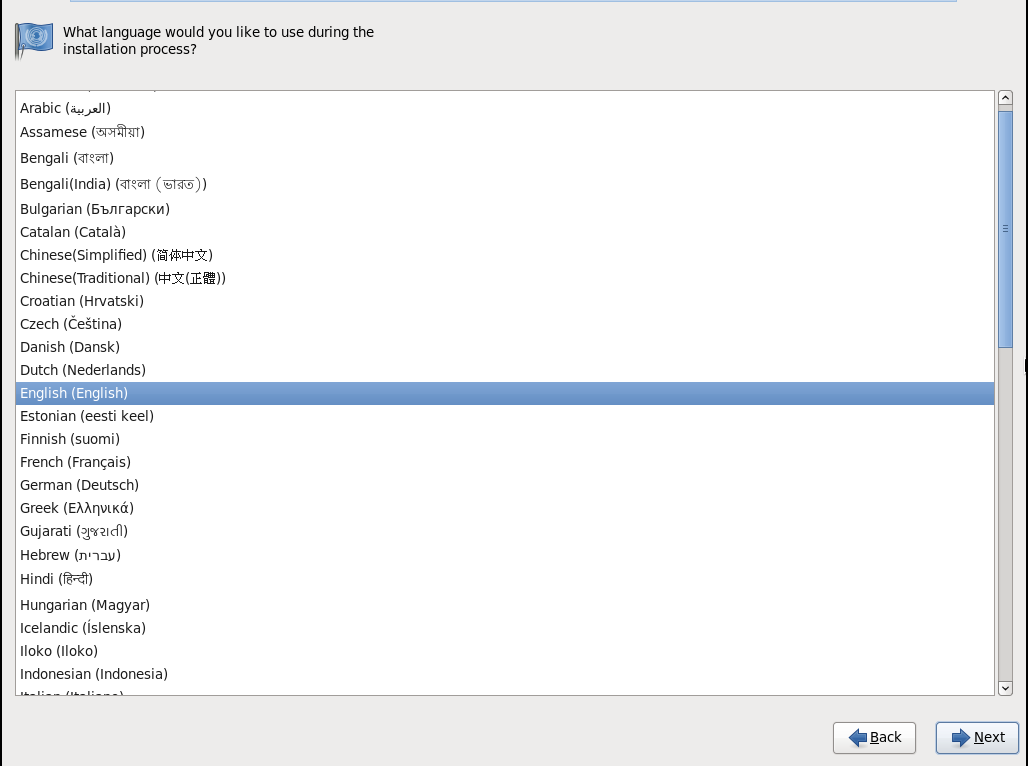
* Install or upgrade an existing system 安装或升级现有的系统
* Install system with basic video driver 安装过程中采用 基本的显卡驱动
* Rescue installed system 进入系统修复模式
* Boot from local drive 退出安装从硬盘启动



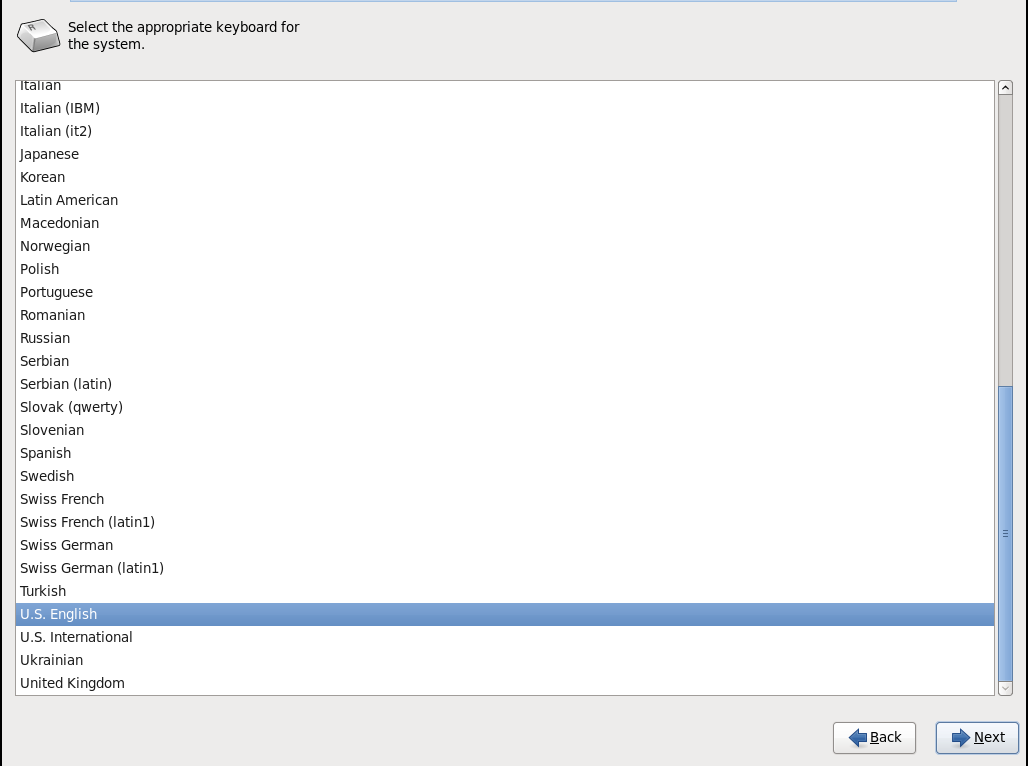
**是否对CD中的文件进行测试，选择"Skip"跳过；**



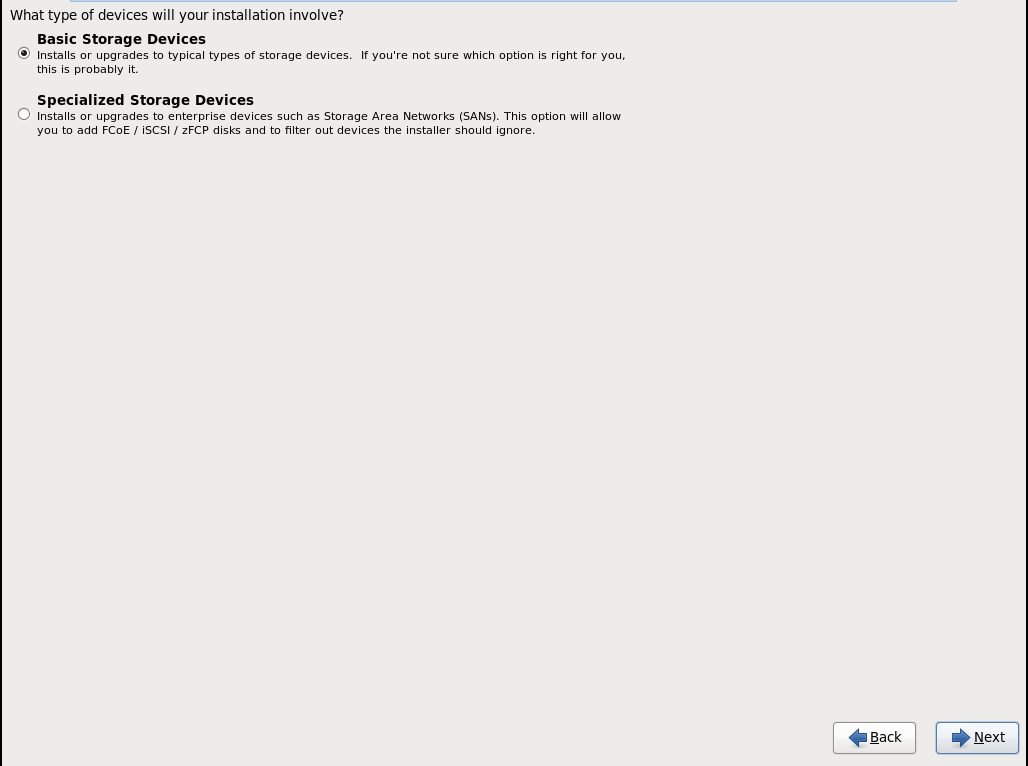
**选择“Next”**

****

**选择语言：English(English)**

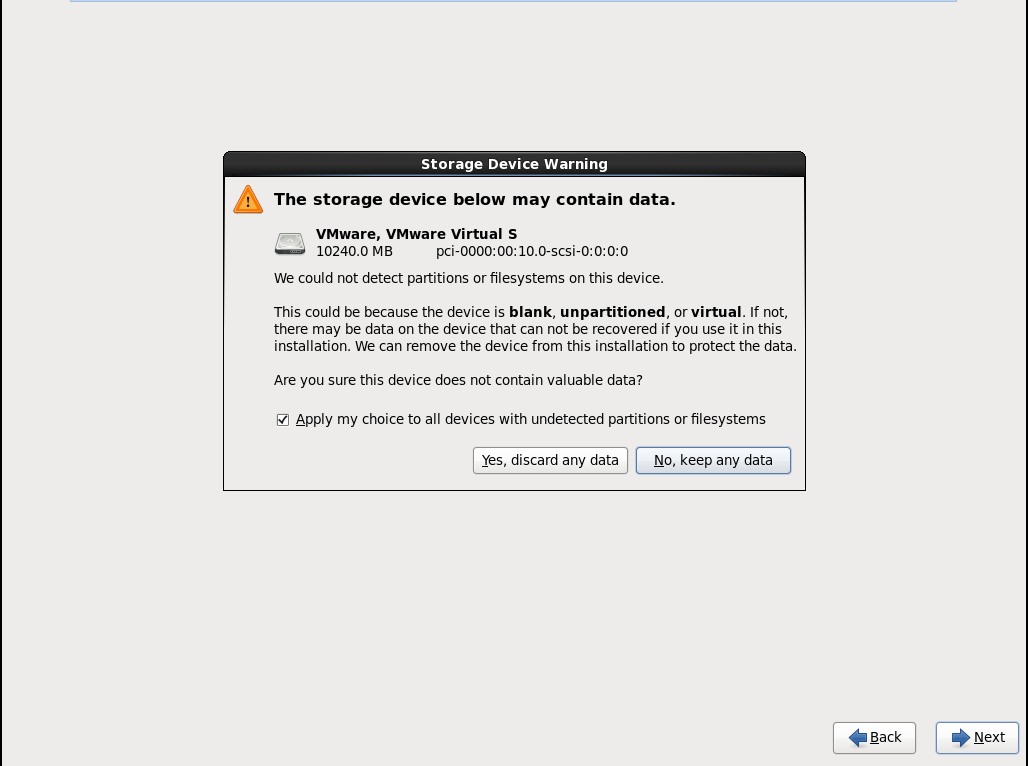
****

**选择键盘：美式键盘U.S. English**

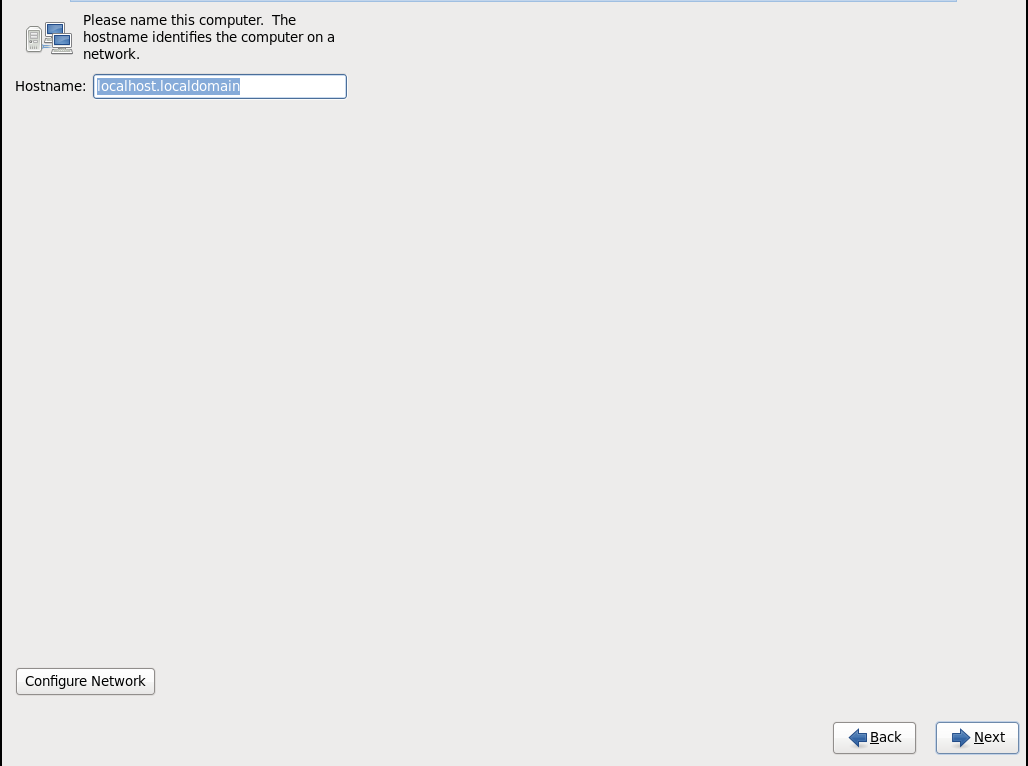
****

**选择第一项Basic Storage Devices，点击“Next”。**

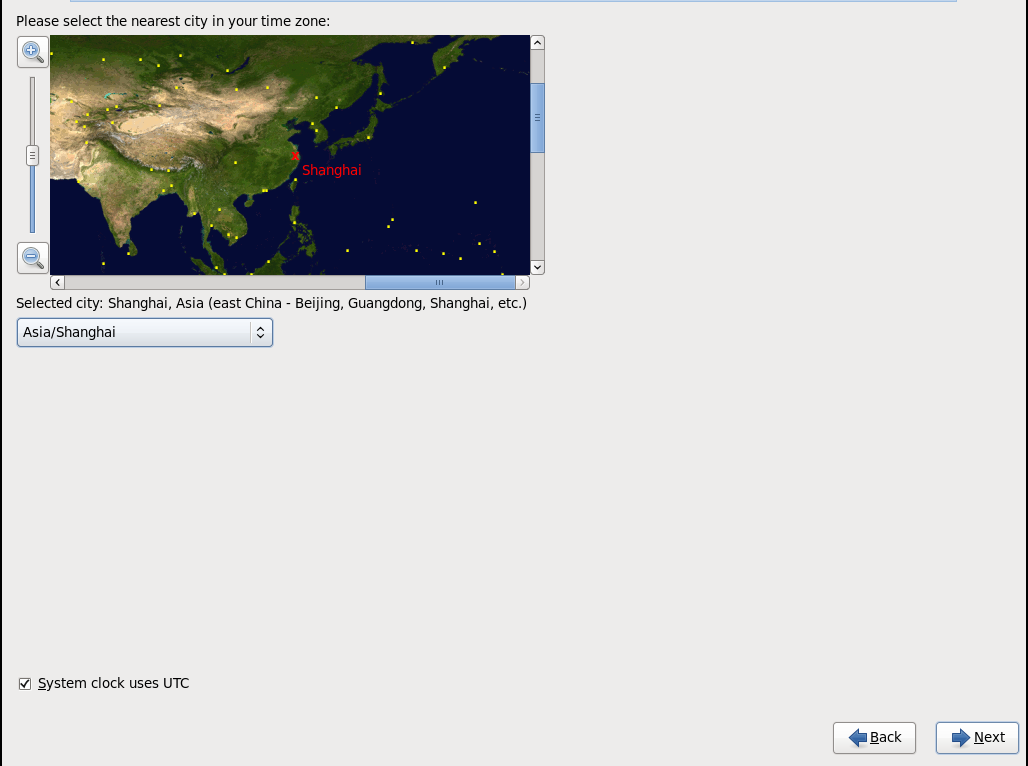
* Basic Storage Devices安装或升级典型的存储设备
* Specialized Storage Devices为企业级网络如存储区域网络(SANs)进行安装或升级



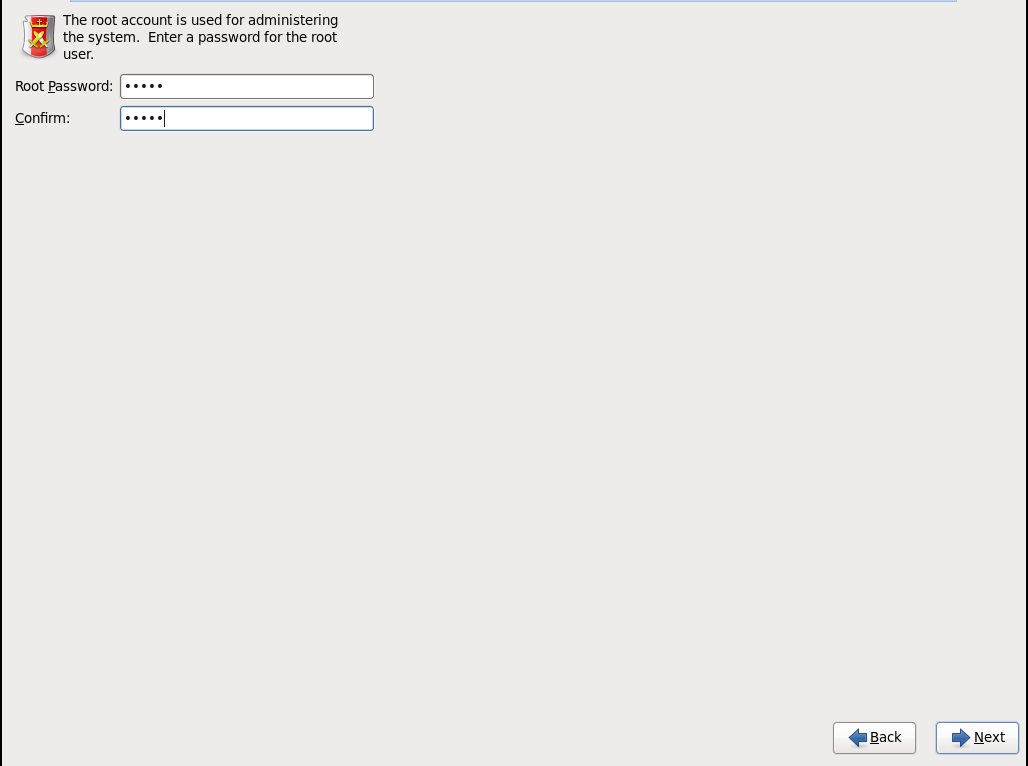
**选择“Yes, discard any data”是，丢弃所有数据**

****

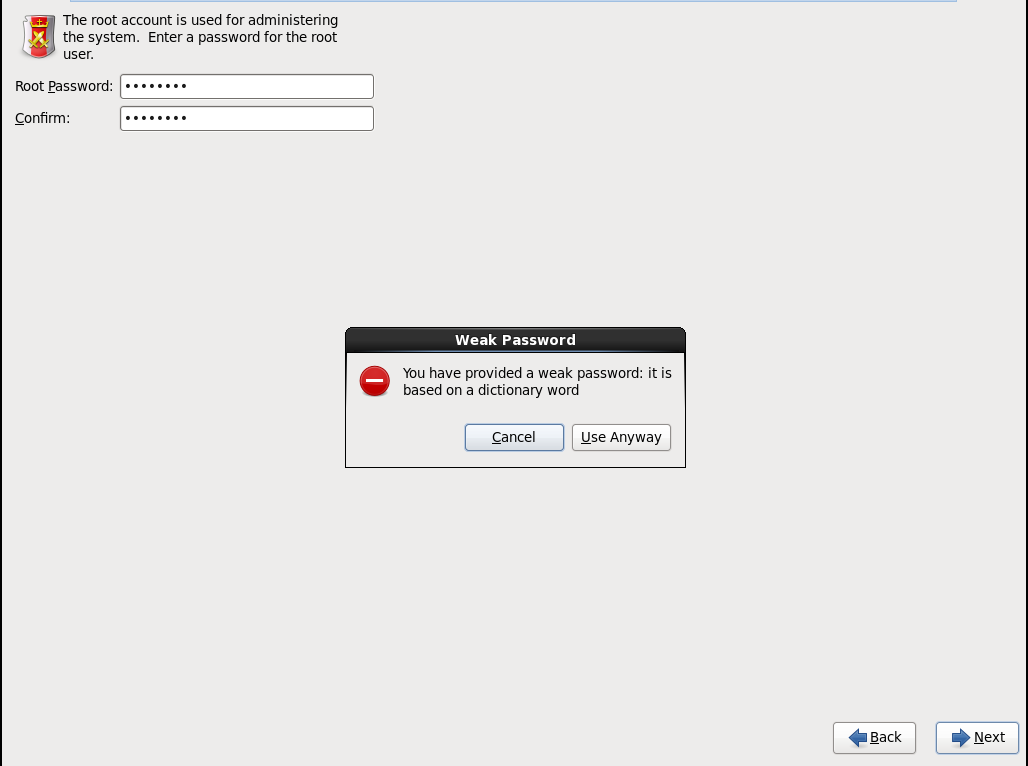
**计算机名，暂时用默认。localhost对应的计算机名，可改为“app55”。**

****

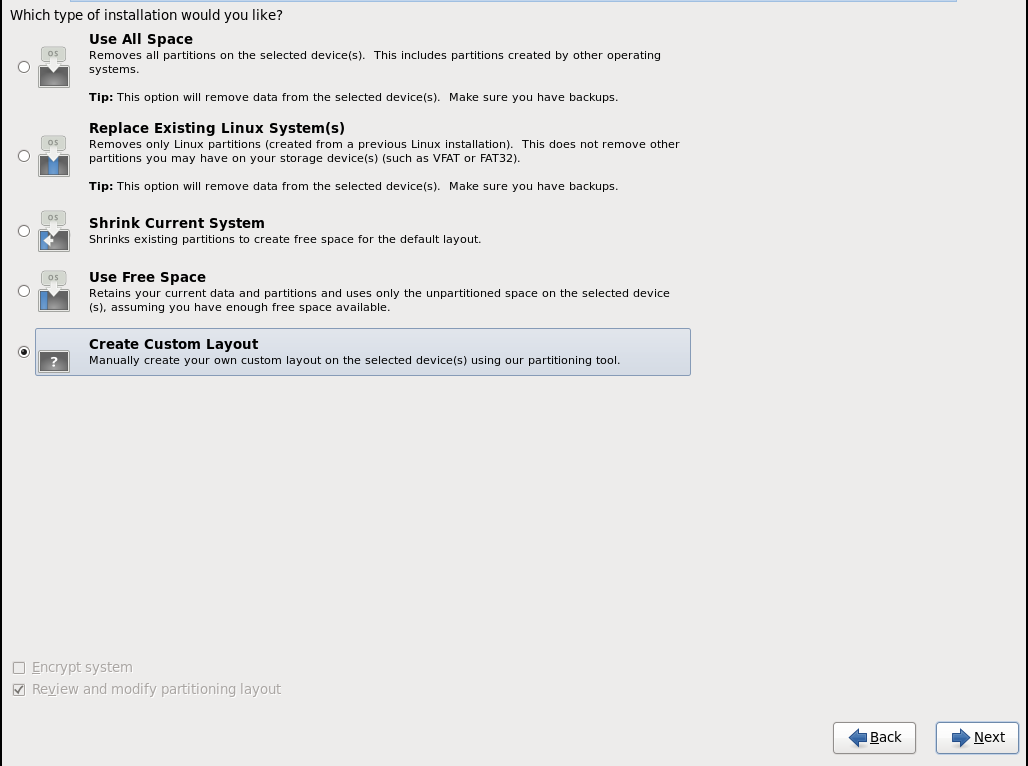
**选择时区：亚洲/上海 Asia/Shanghai**

****

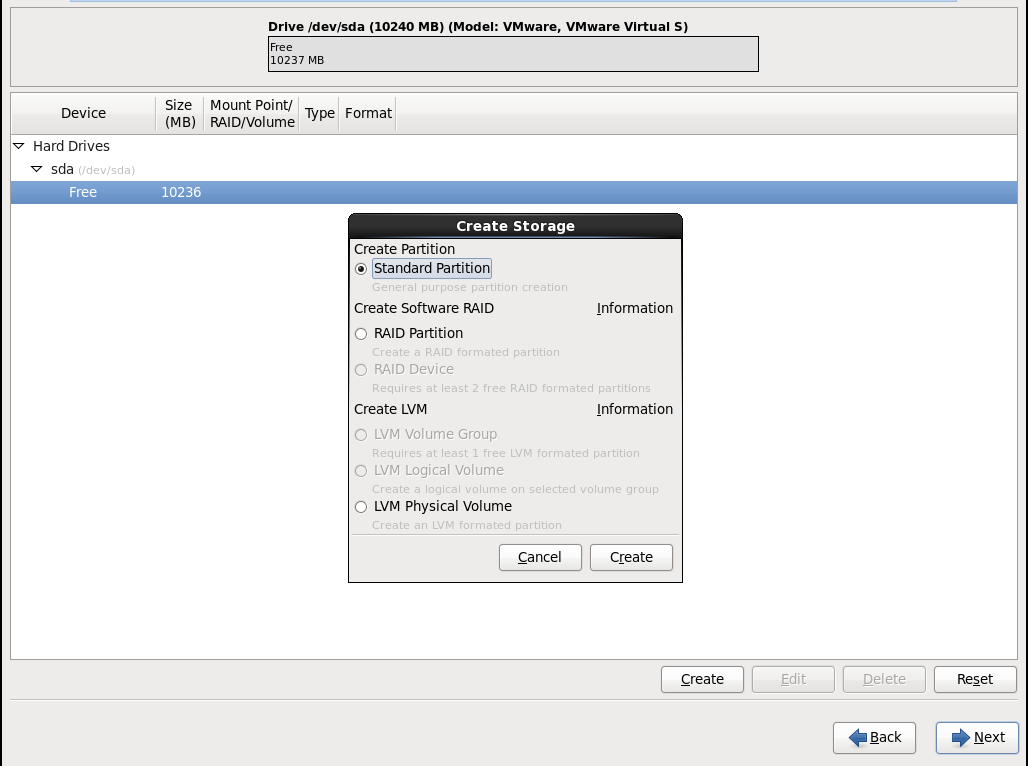
**设置root密码，至少6位。**

****

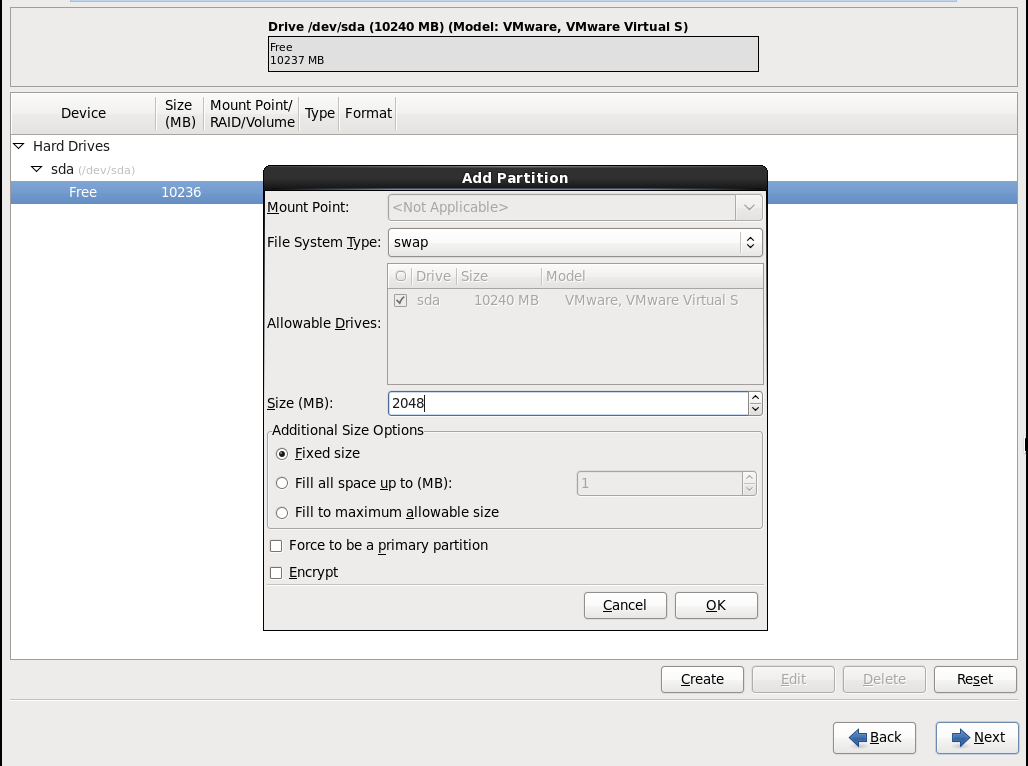
**提示密码过于简单，选择“Use Anyway”仍然使用。**

****

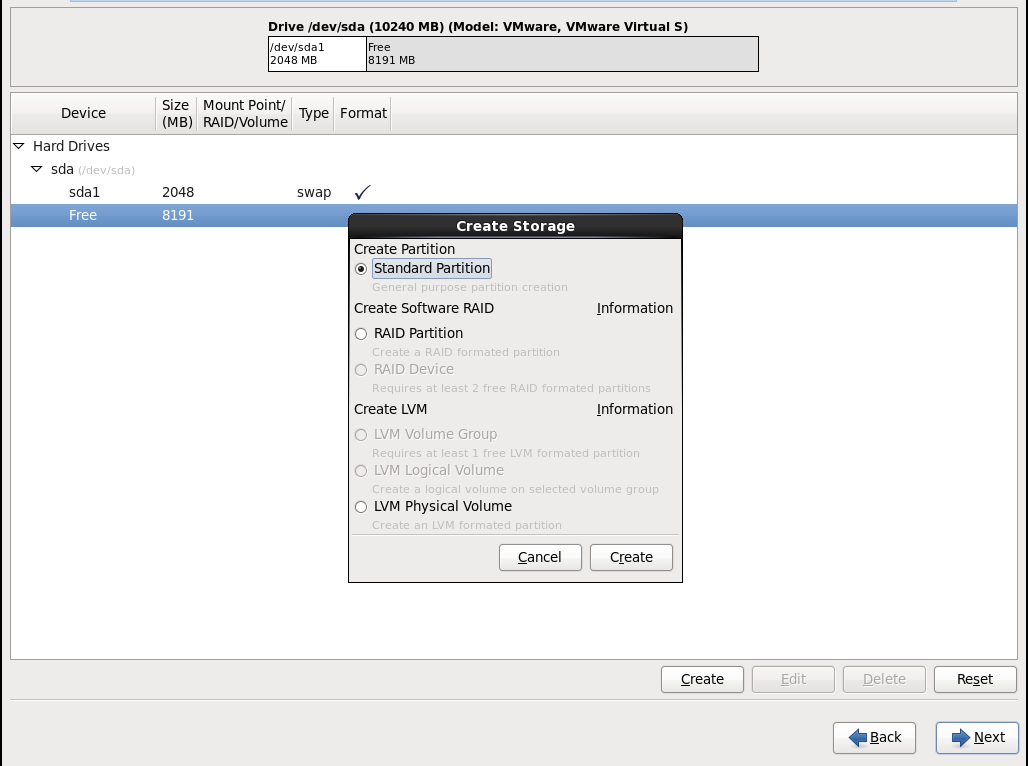
**系统分区：选择最后一项“Create Custom Layout”自定义分区。**

****

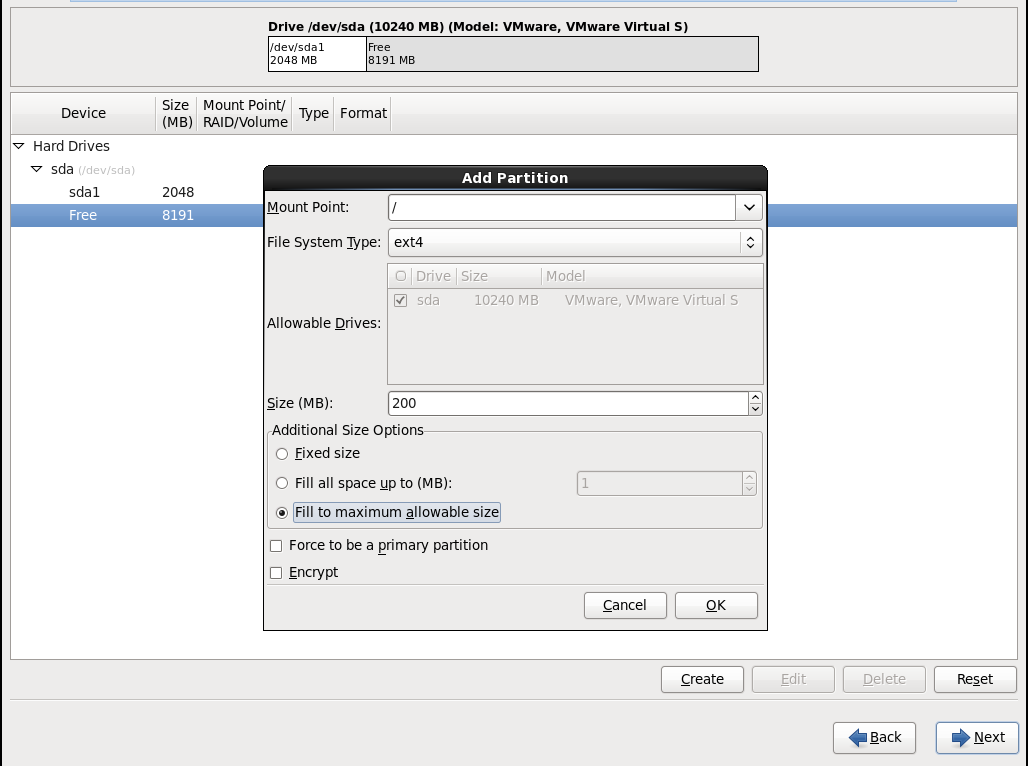
**可以看到当前系统可用磁盘空间为10G，选中这个设备，点击下方的“Create”按钮创建分区。选择Create Partition下的Standard Partition创建标准分区，点击“Create”。**

****

**创建swap区，内存小于2G时，设置为内存的2倍；内存大于或等于2G时，设置为2G；根据实际情况进行配置。**

****

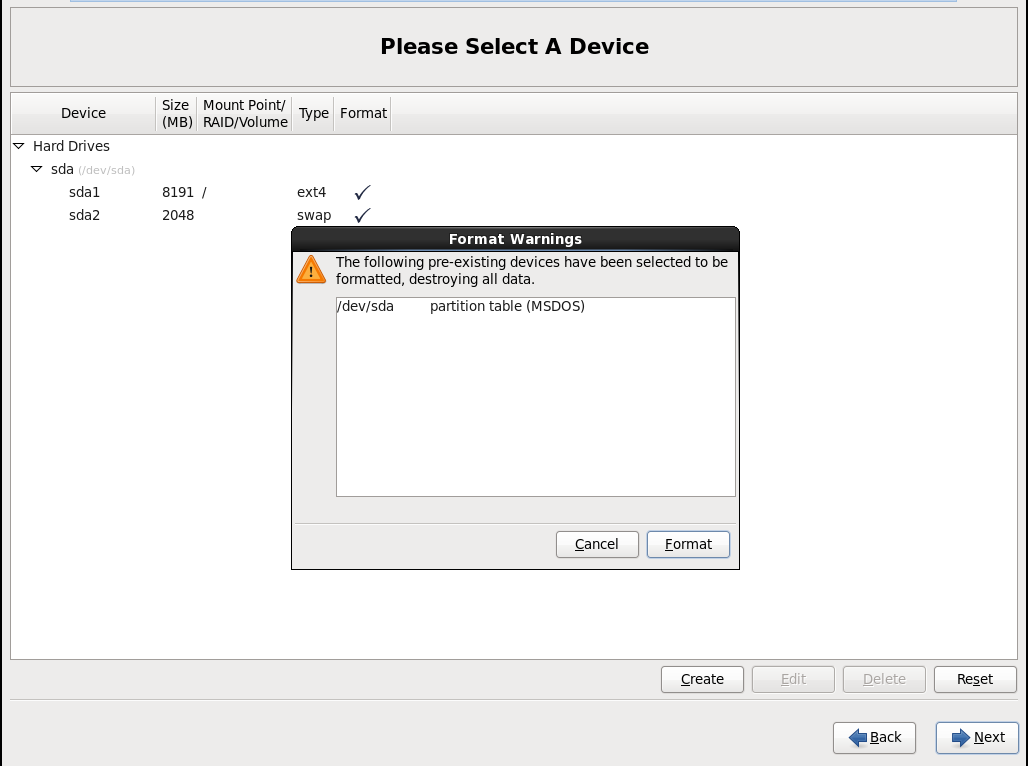
**Swap区划分完成后，点击剩余空间创建磁盘分区，仍然选择创建标准分区。**

****

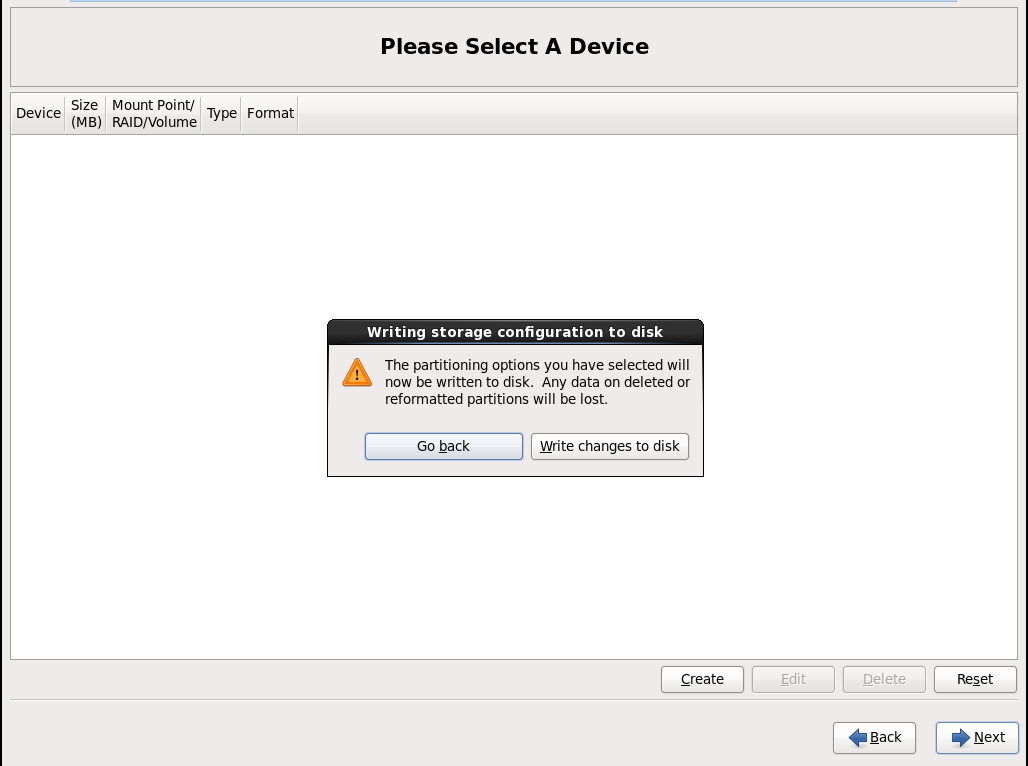
**其余空间全部划分到/下，下面选择“Fill to maximum allowable size”使用全部可用空间。点击“OK”。**

****

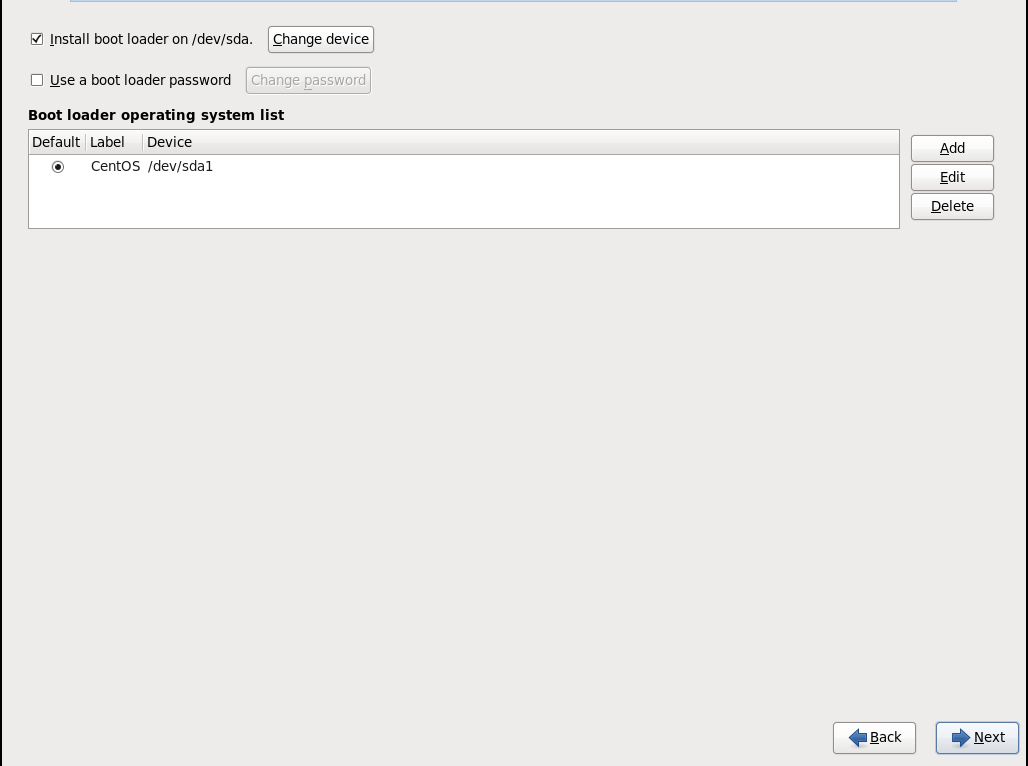
**分区完成，点击“Next”。**

****

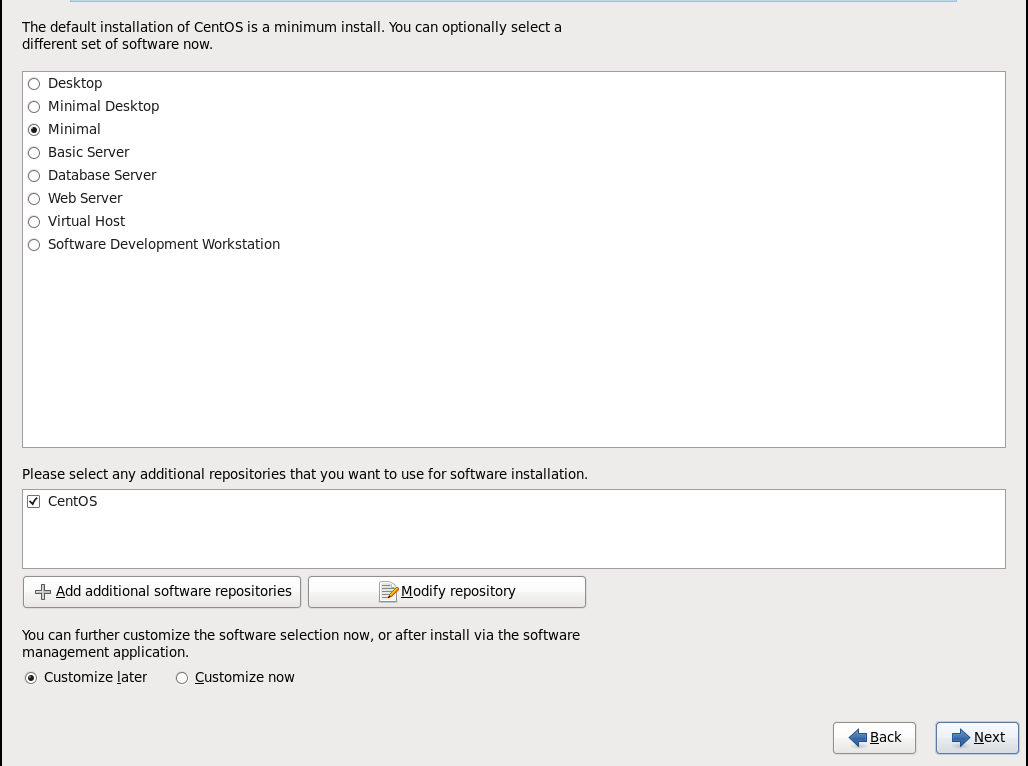
**格式化警告，点击“Format”。**

****

**又一个警告，点击“Write changes to disk”将修改写入磁盘。**

****

**分区完成，直接点击下一步。**

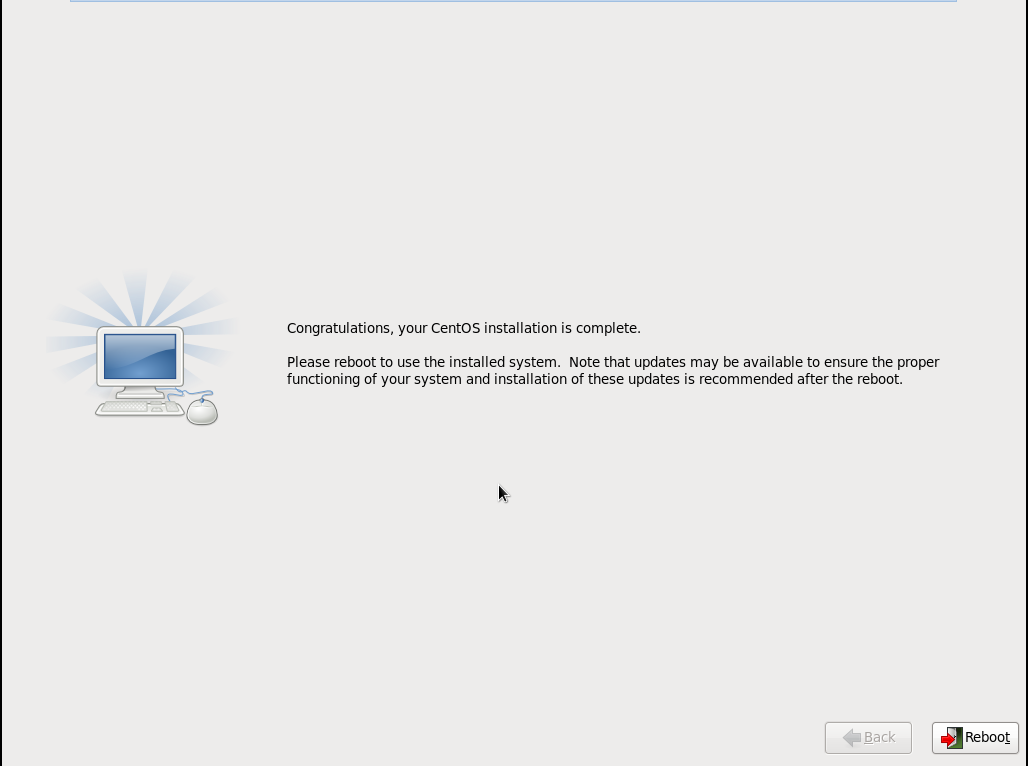
****

**选择“Basic Server”其它默认即可，点击“Next”。**

* Desktop ：基本的桌面系统，包括常用的桌面软件，如文档查看工具。
* Minimal Desktop ：基本的桌面系统，包含的软件更少。
* Minimal ： 基本的系统，不含有任何可选的软件包。
* Basic Server : 基本系统的平台支持，不包含桌面。
* Database Server ： 基本系统平台，加上mysql和PostgreSQL数据库，无桌面。
* Web Server ： 基本系统平台，加上PHP，Web server，还有mysql和PostgreSQL数据库的客户端，无桌面。
* Virtual Host ： 基本系统加虚拟化平台。
* Software Development Workstation ： 包含的软件包较多，基本系统，虚拟化平台，桌面环境，开发工具。



**开始安装。。。**

****

**Congratulations，Reboot重启系统。**

**进入系统后需要输入用户名和密码登陆，默认用户root。**

# 操作系统优化/安装组件

## 修改网卡配置

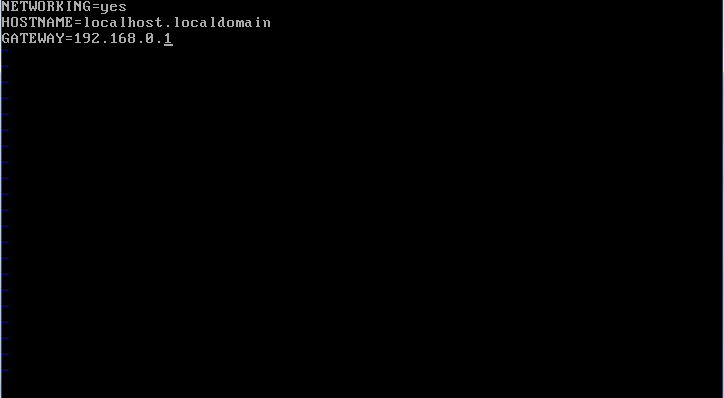
vi /etc/sysconfig/network-scripts/ifcfg-eth0修改ONBOOT=“yes”



1. 重启网络服务 #service network restart。
2. vi /etc/sysconfig/network-scripts/ifcfg-eth0修改IP地址：增加IPADDR="192.168.0.90" NETMASK="255.255.255.0"



1. 修改网关：# vi /etc/sysconfig/network加入GATEWAY=192.168.0.1



1. 修改DNS，#vi /etc/resolv.conf加入nameserver 192.168.0.1

nameserver 8.8.8.8

1. 重启网络服务#service network restart。

## 更新源

可选操作，使用163的源。

* mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup
* wget http://mirrors.163.com/.help/CentOS6-Base-163.repo -O /etc/yum.repos.d/CentOS-Base.repo

这个163的源已经没有了，这步可以不做了

* **不要执行#yum update否则系统会升级到6.4。**

## 安装组件

* yum install openssh-server #安装openssh
* yum install lrzsz #安装securCRT组件
* yum -y install gcc automake autoconf libtool make gcc-c++ patch unzip #安装依赖包
* yum install libtool

## 系统优化

1. echo "ulimit -SHn 102400" >> /etc/rc.local
2. vi /etc/security/limits.conf 加入：

\* soft nofile 65535

\* hard nofile 65535

1. 关闭selinux，如果关闭后ssh连不上则重启服务器

sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config

或vim /etc/sysconfig/selinux

# SELINUX=enforcing

SELINUX=disabled

1. 关闭部分功能

chkconfig bluetooth off

chkconfig cups off

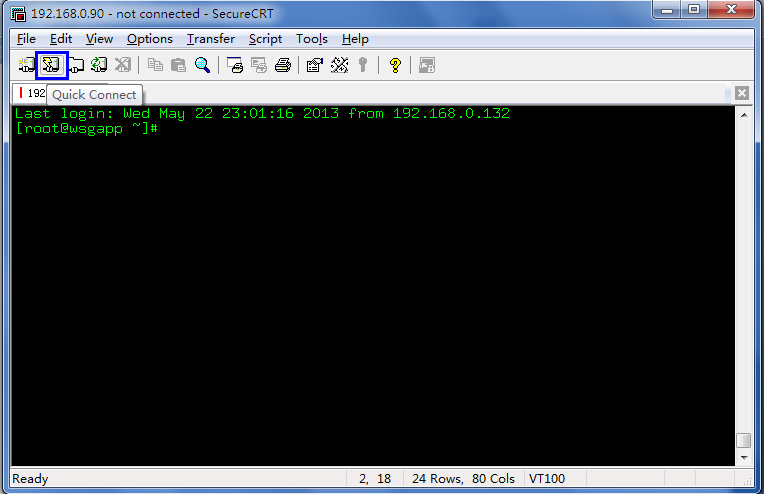
chkconfig ip6tables off

1. 关闭ipv6

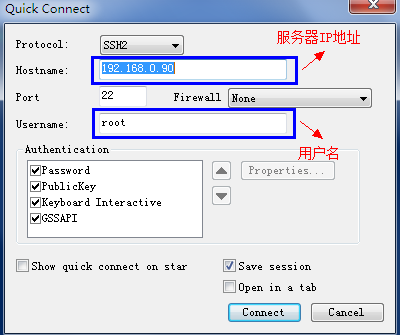
echo "NETWORKING\_IPV6=off" >> /etc/sysconfig/network

# 中间件安装

**注：以下所有安装均使用SecureCRT进行安装；**SecureCRT的配置方式如下：

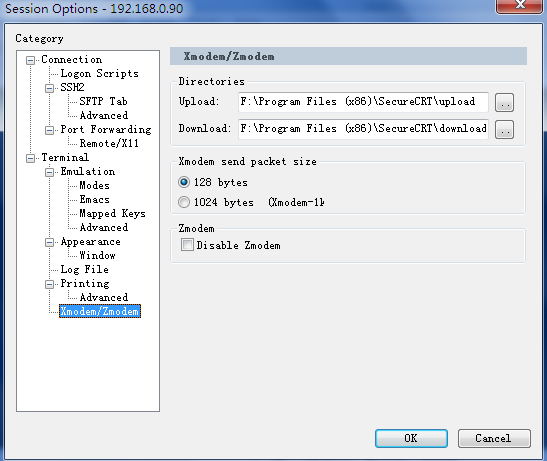


**点击如图蓝框按钮**

****

**输入服务器IP地址和用户名，其它默认。**

可直接使用rz命令进行上传，sz filename 进行下载，上传和下载的默认目录如下图所示，可自行修改：



## 目录结构

1. 分别创建如下目录：mkdir /opt/app /opt/local /opt/tools /opt/logs
2. 目录说明：

* app 应用目录
* local 中间件安装目录
* tools 安装程序/中间件依赖包安装目录
* logs 应用日志目录，包括接口日志、监控日志等

## 安装JDK

使用RPM方式安装

Rpm 安装方式 rpm -ivh jdk-8u73-linux-x64.rpm

默认安装位置 /usr/java

1. 使用rz命令上传文件jdk-6u45-linux-x64.bin到tools目录下：
   1. cd /opt/tools
   2. rz
2. chmod +x jdk-6u45-linux-x64.bin # 修改权限
3. ./jdk-6u45-linux-x64.bin # 执行安装
4. 配置环境变量：
   1. vi /etc/profile
   2. 在文件末尾加入

export JAVA\_HOME=/opt/tools/jdk1.6.0\_45

export CLASSPATH=.:%JAVA\_HOME%/lib/dt.jar:%JAVA\_HOME%/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

* 1. 按“:”输入wq保存退出。

1. 修改默认的java执行
   1. 查看当前是否存在java命令

update-alternatives --config java

update-alternatives --config javac

如果存在可以将原有的删除

update-alternatives --remove java /opt/tools/jdk1.6.0\_xx/bin/java

update-alternatives --remove javac /opt/tools/jdk1.6.0\_xx/bin/javac

或加入新的配置

update-alternatives --install /usr/bin/java java /opt/tools/jdk1.6.0\_45/bin/java 300

update-alternatives --install /usr/bin/javac javac /opt/tools/jdk1.6.0\_45/bin/javac 300

再次执行

update-alternatives --config java

update-alternatives --config javac

通过提示输入序号指定使用哪个JDK版本。

1. 使用java -version命令查看当前JDK版本。

## 安装Tomcat

1. 使用rz命令上传文件apache-tomcat-7.0.40.tar.gz到tools目录。
2. 执行如下命令解压：

tar xzvf apache-tomcat-7.0.40.tar.gz

1. 优化Tomcat参数，修改bin/catalina.sh文件，在开始位置增加

JAVA\_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=128m -XX:PermSize=128m -XX:+UseBiasedLocking"

1. 修改Tomcat配置文件：在conf目录下
   1. 修改server.xml

<?xml version='1.0' encoding='utf-8'?>

<Server port="8005" shutdown="SHUTDOWN">

<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />

<Listener className="org.apache.catalina.core.JasperListener" />

<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />

<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />

<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

<GlobalNamingResources>

<Resource name="UserDatabase" auth="Container"

type="org.apache.catalina.UserDatabase"

description="User database that can be updated and saved"

factory="org.apache.catalina.users.MemoryUserDatabaseFactory"

pathname="conf/tomcat-users.xml" />

</GlobalNamingResources>

<Service name="Catalina">

<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"

maxThreads="1000" minSpareThreads="100"

acceptCount="700"

/>

<Connector executor="tomcatThreadPool"

port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

<Connector executor="tomcatThreadPool"

port="8009" protocol="AJP/1.3" redirectPort="8443" />

<!--增加jvmRoute参数，注意每一个tomcat（同一个负载下的每一个tomcat）都不能相同。-->

<Engine name="Catalina" defaultHost="localhost" jvmRoute="35tomcat1">

<Realm className="org.apache.catalina.realm.LockOutRealm">

<Realm className="org.apache.catalina.realm.UserDatabaseRealm"

resourceName="UserDatabase"/>

</Realm>

<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true">

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"

prefix="localhost\_access\_log." suffix=".txt"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>

</Engine>

</Service>

</Server>

* 1. 修改tomcat-users.xml

<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

<role rolename="manager-gui" />

<user username="tomcat" password="eteng" roles="manager-gui" />

</tomcat-users>

1. 增加应用部署配置文件
   1. 在conf目录下创建目录Catalina/localhost。
   2. 在conf/Catalina/localhost目录下增加ROOT.xml文件。
   3. 修改ROOT.xml内容为如下配置：

<Context path="/" docBase="/opt/app/JXWL/WebContent" debug="0" workDir="work/Catalina/localhost/jxwl">

<Resource name="jxwl" auth="Container"

type="javax.sql.DataSource"

driverClassName="com.mysql.jdbc.Driver"

url="jdbc:mysql://192.168.0.200:3306/ctrm\_develop"

username="root"

password="3dzdyvc="

min="1"

max="5"

jdbcTestStmt="select 1+1"

checkLevelObject="2"

factory="org.objectweb.jotm.datasource.DataSourceFactory"/>

<Transaction factory="org.objectweb.jotm.UserTransactionFactory" jotm.timeout="60"/>

</Context>

关闭防火墙 service iptables stop

或添加对外访问端口 修改/etc/sysconfig/iptables 文件，

添加以下内容：   
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT

1. ~~上传terracotta-toolkit-1.6-runtime-5.5.0.jar、~~terracotta-session-1.3.5.jar到lib目录下；
2. 安装ARP

* 上传文件apr-1.4.6.tar.gz、apr-util-1.5.2.tar.gz到local下。
* 执行解压tar zxvf 文件名
* 安装apr
  + # cd apr-1.4.6
  + # ./configure --prefix=/usr/local/apr
  + # make
  + # make install
* 安装apr-util
* # cd apr-util-1.5.2
* # ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
* # make
* # make install
* 到tomcat/bin目录下安装tomcat-native.tar.gz
  + # cd apache-tomcat-7.0.40
  + # cd bin
  + # tar zxvf tomcat-native.tar.gz
  + # cd tomcat-native-1.1.27-src/jni/native
  + # ./configure --with-apr=/usr/local/apr
  + # make && make install
* 修改tomcat参数
  + # cd apache-tomcat-7.0.40/bin
  + # vi catalina.sh
  + 在语句echo "Using CATALINA\_BASE: $CATALINA\_BASE"的上一行
  + 加入启动参数：

CATALINA\_OPTS="$CATALINA\_OPTS -Djava.library.path=/usr/local/apr/lib"

* 修改环境变量
  + # vi /etc/environment
  + 加入export LD\_LIBARY-PATH=/usr/local/apr:/usr/local/apr/lib

如果参数LD\_LIBARY-PATH已存在，在后面加上/usr/local/apr:/usr/local/apr/lib即可，中间用英文冒号“:”分隔。

1. 将配置好的Tomcat复制到/opt/local目录下，并且重命名：

cp -r apache-tomcat-7.0.40 /opt/local/tomcat-1

-1

1. 根据项目需要复制多个Tomcat。

## ~~安装Terracotta~~不再使用

1. cd /opt/tools
2. rz terracotta-3.7.5.tar.gz
3. tar xzvf terracotta-3.7.5.tar.gz
4. 在根目录增加配置文件：tc-config.xml

<?xml version="1.0" encoding="UTF-8"?>

<con:tc-config xmlns:con="http://www.terracotta.org/config">

<servers>

<server host="serverIP" name="serverName">

<dso-port bind=" serverIP ">9510</dso-port>

<jmx-port bind=" serverIP ">9520</jmx-port>

<data>terracotta/server-data</data>

<logs>terracotta/server-logs</logs>

<statistics>terracotta/cluster-statistics</statistics>

<dso>

<persistence>

<mode>permanent-store</mode>

</persistence>

</dso>

</server>

<ha>

<mode>networked-active-passive</mode>

<networked-active-passive>

<election-time>5</election-time>

</networked-active-passive>

</ha>

</servers>

<clients>

<logs>terracotta/client-logs</logs>

</clients>

</con:tc-config>

1. 在start-tc-server.sh中配置内存大小，确保为512mb。如：${JAVA\_COMMAND} -Xms512m Xmx512m。
2. 启动Terracotta：（使用这种方法时，不能关闭终端）

bin/start-tc-server.sh -f tc-config.xml

1. 后台启动

nohup ./start-tc-server.sh -f ../tc-config.xml &

## 安装Nginx

1. 建立用户

#useradd www

1. 上传Nginx安装包及依赖包到/opt/tools目录下

nginx-1.4.1.tar.gz 、zlib-1.2.3.tar.gz、pcre-8.31.tar.gz

1. 安装zlib，支撑nginx compress。

tar -xvzf zlib-1.2.3.tar.gz  
cd zlib-1.2.3  
./configure  
make  
make install

1. 安装pcre，支撑nginx的rewirte规则

tar xzvf pcre-8.31.tar.gz

 cd pcre-8.31

 ./configure --prefix=/opt/tools/pcre-8.34--enable-utf8 --enable-unicode-properties

make && make install

1. 安装nginx前的准备（nginx安装路径不能为编译路径）

cd /opt/tools

tar zxvf nginx-1.4.1.tar.gz

mkdir /opt/local/nginx

mkdir /opt/local/nginx/var

mkdir /opt/local/nginx/logs

chown -R www.www /opt/local/nginx/var /opt/local/nginx/logs

1. 安装nginx\_upstream\_jvm\_route

cd /opt/tools

unzip nginx-upstream-jvm-route.zip

cd /opt/tools/nginx-1.4.1

patch -p0 < /opt/tools/nginx\_upstream\_jvm\_route/jvm\_route.patch

提示成功

1. 安装Nginx

#cd /opt/tools/nginx-1.4.1

#./configure --user=www --group=www --prefix=/opt/local/nginx --with-http\_stub\_status\_module --with-zlib=/opt/tools/zlib-1.2.3 --with-pcre=/opt/tools/pcre-8.31 --with-http\_stub\_status\_module --add-module=/opt/tools/nginx\_upstream\_jvm\_route

#make && make install

1. 为应用系统做负载均衡的配置：

* 配置nginx.conf：

cd /opt/local/nginx/conf

vi nginx.conf

修改为如下配置：

#body;

worker\_processes 16; #开启进程数

error\_log logs/error.log error;

pid logs/nginx.pid;

events {

use epoll;

worker\_connections 10240;

}

http {

include mime.types;

default\_type application/octet-stream;

include proxy.conf;

include gzip.conf;

large\_client\_header\_buffers 4 32k;

sendfile on;

keepalive\_timeout 65;

upstream business { #应用负载配置

server 192.168.0.35:8080 srun\_id=35tomcat1;

server 192.168.0.35:8081 srun\_id=35tomcat2;

server 192.168.0.36:8080 srun\_id=36tomcat1;

server 192.168.0.36:8081 srun\_id=36tomcat2;

jvm\_route $cookie\_JSESSIONID|sessionid reverse;

}

upstream tomcat-rep { #报表平台集群

server 192.168.1.53:8019;

server 192.168.1.53:8020;

}

upstream tomcat-lms { #日志平台集群

server 192.168.1.53:9020;

server 192.168.1.53:9021;

}

# lms cluster port

server { #监听端口10000，为日志服务器端口

listen 10000;

location / {

proxy\_pass http://tomcat-lms;

}

}

# firewall through port

server { #监听端口80，为应用服务器端口

listen 80;

location / {

proxy\_pass http://tomcats;

}

location /repapp { #请求http://xxxxx/repapp为访问报表服务器

proxy\_pass tomcat-rep;

}

location ^~ /socket.io { #请求为xxx/socket.io为访问Node服务器

proxy\_pass http://127.0.0.1:3200;

}

location = /nginx\_status {

stub\_status on;

}

location ~ ^/(WEB-INF)/ {

deny all;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

* 配置proxy.conf

cd /opt/local/nginx/conf

vi proxy.conf

修改为如下配置：

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

client\_max\_body\_size 10m;

client\_body\_buffer\_size 128k;

proxy\_connect\_timeout 300;

proxy\_send\_timeout 300;

proxy\_read\_timeout 300;

proxy\_buffer\_size 4k;

proxy\_buffers 4 32k;

proxy\_busy\_buffers\_size 64k;

proxy\_temp\_file\_write\_size 64k;

* 配置gzip.conf

cd /opt/local/nginx/conf

vi gzip.conf

修改为如下配置：

gzip on;

gzip\_min\_length 1000;

gzip\_types text/plain text/css application/x-javascript;

* nginx配置文件正确性测试

/opt/local/nginx/sbin/nginx -t

**查看检验结果必须验证通过才行，如果出现error等信息需要分析解决。**

1. 作为web服务器配置时，proxy.conf和gzip.conf与上述配置相同。

* nginx.conf的配置如下：

**注：该配置环境下，所有的静态文件均从web服务器读取，包含js文件，补丁需要同时打在web服务器的应用下，或取消js静态文件读取。**

#body;

worker\_processes 4;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

error\_log logs/error.log error;

pid logs/nginx.pid;

events {

use epoll;

worker\_connections 10240;

}

http {

include mime.types;

default\_type application/octet-stream;

include proxy.conf;

include gzip.conf;

large\_client\_header\_buffers 4 32k;

sendfile on;

#tcp\_nopush on;

keepalive\_timeout 65;

#limit\_zone one $binary\_remote\_addr 10m;

server {

listen 80;

location / {

proxy\_pass http://192.168.1.53;

}

location ~ \.(htm|html|gif|jpg|jpeg|png|ico|rar|css|zip|txt|flv|swf|doc|ppt|xls|pdf)$ {

root /opt/app/isa;

access\_log off;

expires 24h;

}

location ~ ^/(WEB-INF)/ {

deny all;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

1. 启动方式

#./opt/local/nginx/sbin/nginx

1. 关闭方式

#kill nginxPid

# 系统部署

## 端口配置规范

注意：所以服务器的端口定义规则如下：

* **应用服务器：**
  + http端口从8080开始，顺序加1；
  + shutdown端口从8010开始，顺序加1；
  + ajp端口从8020开始，顺序加1；
  + ssl端口从8030开始，顺序加1；
* **报表服务器：**
  + http端口从8180开始，顺序加1；
  + shutdown端口从8110开始，顺序加1；
  + ajp端口从8120开始，顺序加1；
  + ssl端口从8130开始，顺序加1；
* **日志服务器：**
  + http端口从8280开始，顺序加1；
  + shutdown端口从8210开始，顺序加1；
  + ajp端口从8220开始，顺序加1；
  + ssl端口从8230开始，顺序加1；
* **短信服务器：**
  + http端口从8380开始，顺序加1；
  + shutdown端口从8310开始，顺序加1；
  + ajp端口从8320开始，顺序加1；
  + ssl端口从8330开始，顺序加1；
* **计费服务器：**
  + http端口从8480开始，顺序加1；
  + shutdown端口从8410开始，顺序加1；
  + ajp端口从8420开始，顺序加1；
  + ssl端口从8430开始，顺序加1；
* **Node服务器端口：3200**
* **Terracotta服务器：9510、9520**
* **Nginx服务器：**
  + 应用服务器负载：80或根据实际情况定义；
  + 报表服务器负载：监听80端口/repapp请求；
  + 日志服务器负载：监听10000或根据实际情况定义；
  + Node服务器转发：监听80端口/socket.io请求；
* **NginxWeb服务器**
  + 监听80端口将所有请求指向到内网Nginx监听端口；

## 注意事项

由于部署环境为Linux系统，因此系统中所有的路径须遵循Linux的目录结构，系统中所有的log日志、监控日志、其它自定义日志和文件暂存位置均放在“/opt/logs/项目名\_日志类型/”目录下，需要修改common\_config.xml中<tracepath>目录，interface.xml中<defaultlogpath>、<defaultUploadPath>目录、ftp接口上传和下载目录、不同接口自定义日志目录的路径和其它所使用到的路径。

## 组件部署

### REP报表服务器部署

1. 上传REP.zip至服务器/opt/app目录下；
2. 解压缩：unzip REP.zip；
3. 拷贝Tomcat至/opt/local目录下并重命名为tomcat-rep-0根据需要配置多个Tomcat时，末位加1；
4. 修改tomcat-rep-x/conf/Catalina/localhost/ROOT.xml文件，修改项目路径和数据库连接，修改连接池大小，确定连接池名称与项目中common\_config.xml指定的名称相同。
5. 修改ROOT.xml文件名为repapp.xml。

mv ROOT.xml repapp.xml

1. 由于报表服务器不需要使用session，所以不需要使用Terracotta，去掉context.xml中的相关配置。
2. 修改Tomcat访问端口、内存、监控端口；

### LMS日志服务器部署

1. 上传LMS.zip至服务器/opt/app目录下；
2. 解压缩：unzip LMS.zip；
3. 拷贝Tomcat至/opt/local目录下并重命名为tomcat-lms-0根据需要配置多个Tomcat时，末位加1；
4. 修改tomcat-lms-x/conf/Catalina/localhost/ROOT.xml文件，修改项目路径和数据库连接，修改连接池大小，确定连接池名称与项目中common\_config.xml指定的名称相同。
5. 日志服务器不需要使用session，所以不需要使用Terracotta，去掉context.xml中的相关配置。
6. 日志系统需要使用Terracotta提供的Ehcache，修改ehcache.xml文件内容如下：

<ehcache name="ET\_EHCACHE\_TEST" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://ehcache.org/ehcache.xsd" updateCheck="false">

<diskStore path="/opt/ehcache"/>

<cache name="testwrite"

eternal="true"

maxElementsInMemory="1000"

maxElementsOnDisk="100"

memoryStoreEvictionPolicy="FIFO"

overflowToDisk="false">

<cacheWriter writeMode="write-behind"

maxWriteDelay="5"

minWriteDelay="1"

rateLimitPerSecond="5"

writeCoalescing="true"

writeBatching="true"

writeBatchSize="10"

retryAttempts="2"

retryAttemptDelaySeconds="2">

<cacheWriterFactory class="et.common.ehcache.write.ETCacheWriterFactory" />

</cacheWriter>

</cache>

<cache name="sysinit"

maxElementsInMemory="100"

eternal="false"

timeToIdleSeconds="10"

timeToLiveSeconds="10"

memoryStoreEvictionPolicy="LFU">

<cacheEventListenerFactory class="et.common.ehcache.listener.ETCacheEventListenserFactory" listenFor="all"/>

<searchable>

<searchAttribute name="sysinit\_id" expression = "value.getSysinit\_id()"/>

<searchAttribute name="initvalue" expression = "value.getInitvalue()"/>

</searchable>

</cache>

<cache name="billcoderule"

maxElementsInMemory="100"

maxElementsOnDisk="0"

eternal="false"

timeToIdleSeconds="120"

timeToLiveSeconds="0"

memoryStoreEvictionPolicy="LFU"

diskExpiryThreadIntervalSeconds="1"

transactionalMode="local">

</cache>

<cache name="billcodemax"

maxElementsInMemory="100"

maxElementsOnDisk="100"

eternal="false"

timeToIdleSeconds="120"

timeToLiveSeconds="0"

memoryStoreEvictionPolicy="LFU">

</cache>

<cache name="defrefmodel"

maxElementsInMemory="100"

eternal="false"

timeToIdleSeconds="10"

timeToLiveSeconds="10"

memoryStoreEvictionPolicy="LFU">

<cacheEventListenerFactory class="et.common.ehcache.listener.ETCacheEventListenserFactory" listenFor="all"/>

</cache>

<terracottaConfig url="**serverIP**:9510"/>

</ehcache>

1. 修改interface.xml启用相应的接口；
2. 修改Tomcat访问端口、内存、监控端口；

## 应用部署

1. 拷贝WebContent副本，删除其中的CVS（SVN）文件和文件夹，修改文件夹名称为项目名称（如：TSingTao）；
2. 修改相应的配置文件和路径；
   1. common\_config.xml；
   2. interfase.xml;
   3. simplelog.properties;
3. 打包成zip格式；
4. 上传至服务器/opt/app目录下；
5. 使用unzip解压缩。
6. 拷贝Tomcat至/opt/local目录下并重命名为tomcat-tst-0根据需要配置多个Tomcat时，末位加1；
7. 修改tomcat-tst-x/conf/Catalina/localhost/ROOT.xml文件，修改项目路径和数据库连接，修改连接池大小，确定连接池名称与项目中common\_config.xml指定的名称相同。
8. 业务系统服务器需要使用Terracotta提供的Session共享功能，在context.xml中配置。
9. 修改Tomcat访问端口、内存、监控端口；

# 部分命令

## 查看操作系统版本号

* # cat /etc/redhat-release
* # rpm -q centos-release

## 常用命令

ps -aux #查看进程

ps -ef|grep java #查看java进程（用于检查tomcat是否在运行）

kill pid #杀死进程

sudo su #切换到root用户

chmod #修改权限

rz #上传

sz #下载

cp #复制

df #查看磁盘空间

mv #移动或重命名

clear #清空控制台

mkdir #新建文件夹

rm #删除

vi filename #文本编辑，Esc、shift+:、q!不保存退出、wq保存退出

>error.log #清空log日志

## 查看系统信息

1. 查看系统版本： lsb\_release -a
2. 查看CPU完整信息：cat /proc/cpuinfo
3. 查看CPU型号：cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c
4. 查看CPU数量：cat /proc/cpuinfo | grep physical | uniq -c
5. 查看CPU运行模式（32/64）：getconf LONG\_BIT
6. 查看内存信息：cat /proc/meminfo

## 压缩和解压

.tar

解包：tar xvf FileName.tar

打包：tar cvf FileName.tar DirName

（注：tar是打包，不是压缩！）

---------------------------------------------

.gz

解压1：gunzip FileName.gz

解压2：gzip -d FileName.gz

压缩：gzip FileName

.tar.gz 和 .tgz

解压：tar zxvf FileName.tar.gz

压缩：tar zcvf FileName.tar.gz DirName

## 查看磁盘空间

df -hl 查看磁盘剩余空间

df -h 查看每个根路径的分区大小

du -sh [目录名] 返回该目录的大小

du -sm [文件夹] 返回该文件夹总M数

## 挂载磁盘

1. fdisk -l :打印当前的磁盘分区表
2. fdisk /dev/sdb ：未进行挂载和分区的磁盘
3. 键入 m 查看帮助；键入n添加新的分区
4. 键入p创建主分区；
5. 创建盘符，直接敲回车或者输入提示默认值
6. 起始和结束位置，如果只分成一个区直接敲回车即可
7. 键入w保存退出
8. mkfs -t ext4 /dev/sdb1 ：格式化指定分区
9. mount /dev/sdb1 /opt/app 挂载分区，挂到/opt/app目录下
10. vi /etc/fstab
11. 在最下面添加：/dev/sdb1       /opt/app      ext4    defaults,        0       0

参数说明：按参数位置

1. 指代文件系统的设备名。最初，该字段只包含待挂载分区的设备名（如/dev/sda1）。现在，除设备名外，还可以包含LABEL或UUID
2. 文件系统挂载点。文件系统包含挂载点下整个目录树结构里的所有数据，除非其中某个目录又挂载了另一个文件系统
3. 文件系统类型。下面是多数常见文件系统类型（ext3,tmpfs,devpts,sysfs,proc,swap,vfat）
4. mount命令选项。mount选项包括noauto（启动时不挂载该文件系统）和ro（只读方式挂载文件系统）等。在该字段里添加用户或属主选项，即可允许该用户挂载文件系统。多个选项之间必须用逗号隔开。其他选项的相关信息可参看mount命令手册页（-o选项处）
5. 转储文件系统？该字段只在用dump备份时才有意义。数字1表示该文件系统需要转储，0表示不需要转储
6. 文件系统检查？该字段里的数字表示文件系统是否需要用fsck检查。0表示不必检查该文件系统，数字1示意该文件系统需要先行检查（用于根文件系统）。数字2则表示完成根文件系统检查后，再检查该文件系统

# FAQ

修改 **/boot/grub/grub.conf**  追加 **acpi=off 禁用acpi**