

目錄

关于本文	1.1
1. 初始设置	1.2
1.1. 添加用户	1.2.1
1.2. 防火墙和SELinux	1.2.2
1.3. 配置网络	1.2.3
1.4. 配置服务	1.2.4
1.5. 系统更新	1.2.5
1.6. 软件仓库	1.2.6
1.7. 配置vim	1.2.7
1.8. 配置NTP	1.2.8
1.9. 配置SSH	1.2.9
2. 虚拟化	1.3
2.1. KVM	1.3.1
2.2. oVirt	1.3.2
2.3. Xen	1.3.3
2.4. Docker	1.3.4
2.5. Kubernetes	1.3.5
3. 桌面环境	1.4
3.1. GNOME桌面	1.4.1
3.2. KDE桌面	1.4.2
3.3. Xrdp服务器	1.4.3
3.4. VNC服务器	1.4.4
3.5. RDP连接到Windows	1.4.5
4. 存储服务器	1.5
4.1. NFS服务器	1.5.1
4.2. iSCSI	1.5.2
4.3. Ceph	1.5.3

4.4. GlusterFS	1.5.4
5. Web服务器	1.6
5.1. Apache httpd	1.6.1
5.2. Nginx	1.6.2
5.3. 创建SSL证书	1.6.3
6. 数据库	1.7
6.1. MariaDB	1.7.1
6.2. PostgreSQL	1.7.2
6.3. Oracle Database	1.7.3
6.4. Memcached	1.7.4
6.5. Redis	1.7.5
6.6. MS SQL Server	1.7.6
7. 目录服务	1.8
7.1. FreeIPA	1.8.1
7.2. OpenLDAP	1.8.2
7.3. NIS	1.8.3
8. 文件服务器	1.9
8.1. FTP	1.9.1
8.2. Samba	1.9.2
8.3. ownCloud	1.9.3
9. 邮件服务器	1.10
9.1. 安装Postfix	1.10.1
9.2. 安装Dovecot	1.10.2
9.3. 配置SSL	1.10.3
9.4. 邮件日志报告	1.10.4
9.5. WebMail	1.10.5
9.6. iRedMail	1.10.6
10. 网络服务	1.11
10.1. DNS DHCP服务	1.11.1
10.2. 代理服务器	1.11.2

10.3. 网络性能测试	1.11.3
10.4. PXE	1.11.4
10.5. OpenVPN	1.11.5
10.6. PPTP	1.11.6
10.7. L2TP IPSec	1.11.7
11. 负载均衡	1.12
11.1. HAProxy	1.12.1
11.2. Pen	1.12.2
11.3. Pound	1.12.3
11.4. LVS	1.12.4
12. 系统监控	1.13
12.1. OSQuery	1.13.1
12.2. Munin	1.13.2
12.3. SysStat	1.13.3
12.4. Zabbix	1.13.4
12.5. MRTG	1.13.5
12.6. Cacti	1.13.6
12.7. Nagios	1.13.7
12.8. Monitorix	1.13.8
12.9. psacct	1.13.9
13. 语言开发环境	1.14
13.1. Ruby	1.14.1
13.2. JavaScript	1.14.2
13.3. PHP	1.14.3
13.4. Python	1.14.4
13.5. Java	1.14.5
14. 云计算	1.15
14.1. OpenStack	1.15.1
15. 认证服务器	1.16
15.1. FreeRADIUS	1.16.1

15.2. privacyIDEA	1.16.2
附0. 一些系统配置	1.17
附0.1. 本地化设置	1.17.1
附0.2. 密码相关设置	1.17.2
附0.3. 磁盘相关设置	1.17.3
附0.4. 显示硬件信息	1.17.4
附0.5. 分布式文件系统	1.17.5
附0.6. 更改运行级别	1.17.6
附1. 一些可能有用的	1.18
附1.1. 系统安全	1.18.1
附1.2. 加入Windows活动目录	1.18.2
附1.3. 访问控制	1.18.3
附1.4. 文件同步	1.18.4
附1.5. PowerShell	1.18.5
附1.6. 项目管理与版本控制	1.18.6
附1.7. 系统管理工具	1.18.7
附1.8. 配置管理工具	1.18.8
附1.9. 防火墙	1.18.9
附1.10. 高可用性集群	1.18.10
附1.11. 消息服务器	1.18.11
附1.12. 备份管理工具	1.18.12

CentOS7服务器的一些配置

本文都是比较基础的东西，加上个人使用习惯等原因，并不一定适合所有人，可以在这基础上适当调整，这里算是做个记录。

内容主要参考了[Server World](#)（有些测试不到的就直接翻译下搬过来），再加上平时收集的资料。

本文档使用GitBook发布:<https://www.gitbook.com/book/izombielandgit/centos7-server-configuration/details>

本文档GitHub:<https://github.com/izombielandgit/CentOS7-Server-Configuration>

1. 初始化设置

系统的下载和安装就不单独写了，假定以最小化安装系统。本章主要记录一下新装系统后的一些初始化的配置以让系统更适合自己的使用。

- 1.1. 添加用户
- 1.2. 防火墙和SELinux
 - 1.2.1. 防火墙
 - 1.2.2. SELinux
- 1.3. 配置网络
- 1.4. 配置服务
- 1.5. 系统更新
- 1.6. 软件仓库
- 1.7. 配置vim
- 1.8. 配置NTP
 - 1.8.1. 配置NTP服务器
 - 1.8.1.1. NTPd
 - 1.8.1.2. Chrony
 - 1.8.2. 配置NTP客户端
 - 1.8.2.1. CentOS
 - 1.8.2.2. Windows
- 1.9. 配置SSH
 - 1.9.1. 配置SSH服务器
 - 1.9.1.1. 密码验证登录
 - 1.9.1.2. 密钥验证登录
 - 1.9.2. 配置SSH客户端
 - 1.9.2.1. CentOS客户端
 - 1.9.2.2. Windows客户端
 - 1.9.3. SSH文件传输
 - 1.9.3.1. CentOS客户端
 - 1.9.3.2. Windows客户端
 - 1.9.3.3 仅SFTP + Chroot
 - 1.9.4. SSH端口转发
 - 1.9.5. X11转发
 - 1.9.5.1. CentOS客户端

1. 初始化设置

- 1.9.5.2. Windows客户端
 - 1.9.6. SSHPass
 - 1.9.7. SSH-Agent
 - 1.9.8. Parallel SSH
 - 1.9.9. 微信提醒

1.1. 添加用户

命令执行中输入错误，直接退格不能删除（会出现新的字符），按Ctrl+退格可以删除

1.1. 添加用户

以用户"cent"为例

添加用户：

```
useradd cent # 添加用户cent
```

```
passwd cent # 修改cent密码
```

```
Changing password for user cent.  
New UNIX password:          # 设置密码  
Retype new UNIX password:  # 确认密码  
passwd: all authentication tokens updated successfully.
```

使用新用户登录：

以“cent”用户登录后运行

```
su - # 切换到root
```

```
Password: # 输入root密码。
```

设置只有某个用户能够切换到root：

```
usermod -G wheel cent
```

编辑 /etc/pam.d/su 文件，取消 auth required pam_wheel.so use_uid 一行的注释

编辑 /etc/aliases 文件，在 # Person who should get root's mail 下面添加一行 root:cent ，保存后运行：

```
newaliases
```

以重新加载生效。

1.1. 添加用戶

1.2. 防火墙和SELinux

1.2.1. 防火墙

CentOS7默认使用firewalld防火墙，运行以下命令以查看状态：

```
systemctl status firewalld
```

firewalld的一些简单配置可以参考[这里](#)。

若使用iptables，按下面步骤操作：

```
systemctl stop firewalld # 停止firewalld
```

```
systemctl disable firewalld # 取消开机启动
```

```
yum -y install iptables-services # 安装iptables
```

编辑 /etc/sysconfig/iptables 文件配置规则

```
systemctl restart iptables # 编辑规则后重启防火墙
```

```
systemctl enable iptables # 设置开机启动
```

1.2.2. SELinux

SELinux一些简单配置可以参考[这里](#)。

运行以下命令查看状态：

```
getenforce
```

Enforcing # 表示已启用

对SELinux不熟悉的一般推荐禁用：

编辑 /etc/selinux/config 文件，将 SELINUX= 一行改为 SELINUX=disabled ，重启系统后生效。

1.3. 配置网络

网络的设置推荐在安装系统时在图形界面设置好

网络接口以“eth0”为例，更改为实际环境的接口名称

设置**hostname**：

```
hostnamectl set-hostname dlp.srv.world # “dlp.srv.world”更改为自己的主机名
```

设置静态**IP**：

```
nmcli d # 列出设备
```

DEVICE	TYPE	STATE	CONNECTION
eth0	ethernet	connected	eth0
lo	loopback	unmanaged	--

```
nmcli c modify eth0 ipv4.addresses 10.0.0.30/24 # 设置eth0网卡的IPv4地址，根据自己网络情况设置
```

```
nmcli c modify eth0 ipv4.gateway 10.0.0.1 # 设置默认网关，根据自己网络情况设置
```

```
nmcli c modify eth0 ipv4.dns 10.0.0.1 # 设置DNS，根据自己网络情况设置
```

```
nmcli c modify eth0 ipv4.method manual # 手动设置“manual”,DHCP设置“auto”
```

```
nmcli c down eth0; nmcli c up eth0 # 重启接口以加载设置
```

```
nmcli d show eth0 # 显示设置
```

```
GENERAL.DEVICE:                  eth0
GENERAL.TYPE:                    ethernet
GENERAL.HWADDR:                 00:0C:29:CD:9C:2D
GENERAL.MTU:                     1500
GENERAL.STATE:                  100 (connected)
GENERAL.CONNECTION:              eth0
GENERAL.CON-PATH:                /org/freedesktop/NetworkManager/
ActiveConnection/0
WIRED-PROPERTIES.CARRIER:       on
IP4.ADDRESS[1]:                 ip = 10.0.0.30/24, gw = 10.0.0.1
IP4.DNS[1]:                      10.0.0.1
IP6.ADDRESS[1]:                 ip = fe80::20c:29ff:fedc:9c2d/64
, gw = ::
```

```
ip addr show # 显示状态
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:cd:9c:2d brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.30/24 brd 10.0.0.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fedc:9c2d/64 scope link
        valid_lft forever preferred_lft forever
```

在 `/proc/sys/net/ipv4/conf/` 查看接口名称，编辑 `/etc/sysconfig/network-scripts/ifcfg-接口名称`，也可以进行配置。

添加静态路由（非必须）：

编辑（新建）`/etc/sysconfig/network-scripts/route-接口名称` 文件，输入 `IP(网段)/掩码 via 网关 dev 接口名称`

如果习惯网络接口名称像**ethX**类的格式：

1.3. 配置网络

编辑 `/etc/default/grub` 文件，在 `GRUB_CMDLINE_LINUX=` 一行加入 `net.ifnames=0`，如 `GRUB_CMDLINE_LINUX="net.ifnames=0 rd.lvm.lv=..."`，再运行 `grub2-mkconfig -o /boot/grub2/grub.cfg` 应用修改然后重启系统生效

如果不使用**IPv6**按以下操作禁用：

禁用**IPv6**有可能影响一些服务的运行（如[FreeRADIUS](#)）。

编辑 `/etc/default/grub` 文件，在 `GRUB_CMDLINE_LINUX=` 一行加入 `ipv6.disable=1`，如 `GRUB_CMDLINE_LINUX="ipv6.disable=1 rd.lvm.lv=..."`，再运行 `grub2-mkconfig -o /boot/grub2/grub.cfg` 应用修改然后重启系统生效

1.4. 配置服务

```
systemctl -t service # 列出正在运行的服务
```

UNIT	LOAD	ACTIVE	SUB	DE
auditd.service	loaded	active	running	Se
avahi-daemon.service	loaded	active	running	Av
crond.service	loaded	active	running	Co
dbus.service	loaded	active	running	D-
getty@tty1.service	loaded	active	running	Ge
tty on tty1				
...				
...				
...				
systemd-udevd.service	loaded	active	running	ud
ev Kernel Device Manager				
systemd-update-utmp.service	loaded	active	exited	Up
date UTMP about System Reboot/Shutdown				
systemd-user-sessions.service	loaded	active	exited	Pe
rmit User Sessions				
systemd-vconsole-setup.service	loaded	active	exited	Se
tup Virtual Console				
tuned.service	loaded	active	running	Dy
namic System Tuning Daemon				

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type.

39 loaded units listed. Pass --all to see loaded but inactive units, too.

To show all installed unit files use 'systemctl list-unit-files'

```
systemctl list-unit-files -t service #列出所有服务
```

```
UNIT FILE                                STATE
auditd.service                            enabled
autovt@.service                           disabled
avahi-daemon.service                     enabled
blk-availability.service                 disabled
brandbot.service                          static
...
...
...
systemd-user-sessions.service           static
systemd-vconsole-setup.service          static
teamd@.service                           static
tuned.service                            enabled
wpa_supplicant.service                  disabled

125 unit files listed.
```

```
systemctl stop postfix # 停止stop (如postfix)
```

```
systemctl disable postfix # 禁止开机启动disable
```

```
rm '/etc/systemd/system/multi-user.target.wants/postfix.service'
```

其他：运行start、重启restart、开启开机启动enable、运行状态status等

关于[systemd](#)一些参考

举一个自己添加[systemd](#)服务的例子，开关机微信通知：

新建 /etc/systemd/system/wx-notify.service :

```
[Unit]
Description=Notification when the server is turned on and off
After=network.target

[Service]
Type=idle
# 如果只需要关机执行脚本，ExecStart=可以设置为/bin/true
ExecStart=/opt/wx-notify.sh '开机'
ExecStop=/opt/wx-notify.sh '关机'
RemainAfterExit=yes

[Install]
WantedBy=default.target
```

新建 /opt/wx-notify.sh :

```
#!/bin/bash

CorpID='      ' # 填入企业ID
Secret='      ' # 填入应用Secret
GURL="https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=$CorpID&corpsecret=$Secret"

# get acccess_token
GToken=`/usr/bin/curl -s -G $GURL`
Token=`echo $GToken |awk -F '"' '{print $10}'` 
PURL="https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token=$Token"

wxAppID=xxxxxx # 填入应用AgentId
wxUserID=1 # 企业微信中部门成员ID(企业微信成员信息中称为帐号)
Time=`date +'%Y-%m-%d %H:%M:%S'`
wxMsg='服务器状态提醒：\n主机名：``hostname``\n状态类型：'$1'\n状态时间：'`${Time}`
Body='{"touser":"'${wxUserID}'", "msgtype":"text", "agentid":"'${wxAppID}'", "text": {"content":"'${wxMsg}'"}, "safe":"0" }'

/usr/bin/curl --data-ascii "$Body" $PURL
```

```
chmod 700 /opt/wx-notify.sh  
systemctl start wx-notify  
systemctl enable wx-notify
```

有一些SysV服务（如netconsole），受chkconfig控制：

```
chkconfig --list #列出服务
```

```
Note: This output shows SysV services only and does not include native  
systemd services. SysV configuration data might be overridden by native  
systemd configuration.
```

If you want to list systemd services use 'systemctl list-unit-files'.

To see services enabled on particular target use
'systemctl list-dependencies [target]'.

iprdump	0:off	1:off	2:on	3:on	4:on	5:on
	6:off					
iprinit	0:off	1:off	2:on	3:on	4:on	5:on
	6:off					
iprupdate	0:off	1:off	2:on	3:on	4:on	5:on
	6:off					
netconsole	0:off	1:off	2:off	3:off	4:off	5:off
	6:off					
network	0:off	1:off	2:on	3:on	4:on	5:on
	6:off					

```
chkconfig netconsole off #禁止开机启动off (开启开机启动on)
```

结束进程：

[参考这里](#)

```
ps -ef # 加" | grep 搜索的名称"更准确的定位  
kill 查找到的PID
```

注：标准的kill命令通常都能达到目的。终止有问题的进程，并把进程的资源释放给系统。然而，如果进程启动了子进程，只杀死父进程，子进程仍在运行，因此仍消耗资源。为了防止这些所谓的“僵尸进程”，应确保在杀死父进程之前，先杀死其所有的子进程。

```
kill -l PID
```

-l 选项告诉kill命令用好像启动进程的用户已注销的方式结束进程。当使用该选项时，kill命令也试图杀死所留下的子进程。这个命令也不是总能成功，可能仍然需要先手工杀死子进程，然后再杀死父进程。

给父进程发送一个 TERM 信号，试图杀死它和它的子进程：

```
kill -TERM PPID
```

killall 命令杀死同一进程组内的所有进程。其允许指定要终止的进程的名称，而非PID。

```
killall httpd
```

停止和重启进程：

```
kill -HUP PID
```

立即终止进程：

```
kill -9 PID
```

该命令迫使进程在运行时突然终止，进程在结束后不能自我清理。危害是导致系统资源无法正常释放，一般不推荐使用，除非其他办法都无效。当使用此命令时，一定要通过 ps -ef 确认没有剩下任何僵尸进程。只能通过终止父进程来消除僵尸进程。如果系统中有僵尸进程，并且其父进程是init（杀死init进程意味着关闭系统），而且僵尸进程占用了大量的系统资源，那么就需要在某个时候重启机器以清除进程表。

1.5. 系统更新

联网后更新系统，运行：

```
yum -y update
```

生产环境正常运行后建议不升级内核。编辑 `/etc/yum.conf` 文件，在 `[main]` 的最后添加以下内容：

```
exclude=kernel*
exclude=centos-release*
```

或是在运行时加上参数 `--exclude=kernel*`：

```
yum --exclude=kernel* update
```

yum命令中断后，再运行yum时，出现：

```
Existing lock /var/run/yum.pid: another copy is running as pid 3046.
Another app is currently holding the yum lock; waiting for it to exit...
Another app is currently holding the yum lock; waiting for it to exit...
```

运行以下命令：

```
rm -f /var/run/yum.pid
```

运行yum时遇到问题也可尝试：

```
yum clean all
```

1.6. 软件仓库

配置国内源（可选）

```
wget http://mirrors.163.com/.help/CentOS7-Base-163.repo
cp CentOS7-Base-163.repo /etc/yum.repos.d/
cd /etc/yum.repos.d
mv CentOS-Base.repo CentOS-Base.repo.bak
mv CentOS7-Base-163.repo CentOS-Base.repo
yum clean all
yum makecache
yum update
```

安装优先级插件：

```
yum -y install yum-plugin-priorities
```

将官方仓库优先级设置为1（最高）：

```
sed -i -e "s/\]$/\]\npriority=1/g" /etc/yum.repos.d/CentOS-
Base.repo
```

添加**EPEL**仓库：

```
yum -y install epel-release
```

```
sed -i -e "s/\]$/\]\npriority=5/g" /etc/yum.repos.d/epel.repo # 设
置优先级为5
```

以下为另一种设置方式：

```
sed -i -e "s(enabled=1)enabled=0/g" /etc/yum.repos.d/epel.repo # 将
enabled值设置为0
```

[enabled=0] 时使用**epel**库，需添加 --enablerepo=epel

```
yum --enablerepo=epel install [Package]
```

添加**SCLo**仓库：

```
yum -y install centos-release-scl-rh centos-release-scl
```

```
sed -i -e "s/\]$/\]\npriority=10/g" /etc/yum.repos.d/CentOS-SCLo-scl.repo # 设置优先级为10
```

```
sed -i -e "s/\]$/\]\npriority=10/g" /etc/yum.repos.d/CentOS-SCLo-scl-rh.repo
```

以下为另一种设置方式：

```
sed -i -e "s/enabled=1/enabled=0/g" /etc/yum.repos.d/CentOS-SCLo-scl.repo
```

```
sed -i -e "s/enabled=1/enabled=0/g" /etc/yum.repos.d/CentOS-SCLo-scl-rh.repo
```

```
yum --enablerepo=centos-sclo-rh install [Package]
```

```
yum --enablerepo=centos-sclo-sclo install [Package]
```

添加**Remi**仓库：

```
yum -y install http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
```

```
sed -i -e "s/\]$/\]\npriority=10/g" /etc/yum.repos.d/remi-safe.repo # 设置优先级为10
```

以下为另一种设置方式：

```
sed -i -e "s/enabled=1/enabled=0/g" /etc/yum.repos.d/remi-safe.repo
```

```
yum --enablerepo=remi-safe install [Package]
```

安装**FFMPEG**：

先安装**epel**，然后运行：

```
rpm --import http://li.nux.ro/download/nux/RPM-GPG-KEY-nux.ro
rpm -Uvh http://li.nux.ro/download/nux/dextop/el7/x86_64/nux-dextop-release-0-5.el7.nux.noarch.rpm
sed -i -e "s/\]$/\]\npriority=10/g" /etc/yum.repos.d/nux-dextop.repo
yum -y install ffmpeg ffmpeg-devel
```

批量安装开发环境组件（包括了 `gcc` 、 `make` 等软件）

```
yum -y groupinstall "Development tools"
```

配置中其它一些参数 `/etc/yum.conf` :

```
[main]
cachedir=/var/cache/yum/$basearch/$releasever # yum下载的RPM包的缓存目录
keepcache=0 # 是否保存缓存，1保存，0不保存
debuglevel=2 # 调试级别(0-10)，默认为2
logfile=/var/log/yum.log # yum日志文件路径
exactarch=1 # 在更新的时候，是否允许更新不同版本的RPM包，比如是否在i386上更新i686的RPM包
obsoletes=1 # update的参数，具体请参阅yum(8)，简单的说就是相当于upgrade，允许更新陈旧的RPM包
gpgcheck=1 # 是否检查GPG(GNU Private Guard)，一种密钥方式签名
plugins=1 # 是否允许使用插件，用yum-fastestmirror等插件需要开启。
installonly_limit=5 # 允许保留多少个内核包。
exclude=selinux* # 屏蔽不想更新的RPM包，可用通配符，多个RPM包之间使用空格分离
metadata_expire=90m # 设定保存时长
```

安装tree
摘自[这里](#)：

```
yum -y install tree
```

使用：

```
tree / #列出系统所有目录及子目录，如果文件及目录过多可能会很慢
```

```
tree -L 2/ #指定要查看的目录深度，这里指查看两层目录
```

```
tree --help #查看详细帮助信息
```

```
usage: tree [-acdfghilnpqrstuvxACDFQNSUX] [-H baseHref] [-T title] [-L level [-R]] [-P pattern] [-I pattern] [-o filename] [--version] [--help] [--inodes] [--device] [--noreport] [--nolinks] [--dirsfirst] [--charset charset] [--filelimit[=]#] [--si] [--timefmt[=]<fmt>] [<directory list>]
----- Listing options -----
```

```

-a          All files are listed. 显示所有文件和目录
-d          List directories only. 仅显示目录
-l          Follow symbolic links like directories. 显示链接
原内容

-f          Print the full path prefix for each file. 每个文
件都显示完整的路径名称

-x          Stay on current filesystem only. 仅限当前的文件系
统

-L level    Descend only level directories deep.

-R          Rerun tree when max dir level reached.

-P pattern  List only those files that match the pattern giv
en.

-I pattern  Do not list files that match the given pattern.

--noreport  Turn off file/directory count at end of tree lis
ting.

--charset X Use charset X for terminal/HTML and indentation
line output.

--filelimit # Do not descend dirs with more than # files in th
em.

--timefmt <f> Print and format time according to the format <f>

.

-o filename Output to file instead of stdout.

--du        Print directory sizes.

--prune     Prune empty directories from the output.

----- File options -----

-q          Print non-printable characters as '?'.

-N          Print non-printable characters as is.

-Q          Quote filenames with double quotes.

-p          Print the protections for each file.

-u          Displays file owner or UID number. 显示用户

-g          Displays file group owner or GID number. 显示组

-s          Print the size in bytes of each file.

-h          Print the size in a more human readable way.

--si        Like -h, but use in SI units (powers of 1000).

-D          Print the date of last modification or (-c) stat
us change.

-F          Appends '/', '=', '*', '@', '|' or '>' as per ls
-F.

--inodes    Print inode number of each file.

--device   Print device ID number to which each file belong
s.

----- Sorting options -----

```

```
-v           Sort files alphanumerically by version.  
-r           Sort files in reverse alphanumeric order.  
-t           Sort files by last modification time.  
-c           Sort files by last status change time.  
-U           Leave files unsorted.  
--dirsfirst  List directories before files (-U disables).  
----- Graphics options -----  
-i           Don't print indentation lines.  
-A           Print ANSI lines graphic indentation lines.  
-S           Print with ASCII graphics indentation lines.  
-n           Turn colorization off always (-C overrides).  
-C           Turn colorization on always.  
----- XML/HTML options -----  
-X           Prints out an XML representation of the tree.  
-H baseHref  Prints out HTML format with baseHref as top directory.  
-T string    Replace the default HTML title and H1 header with string.  
--nolinks    Turn off hyperlinks in HTML output.  
---- Miscellaneous options ----  
--version    Print version and exit.  
--help       Print usage and this help message and exit.
```

其它一些工具：

```
yum -y install net-tools wget unzip
```

1.7. 配置vim

安装vim：

```
yum -y install vim-enhanced
```

编辑 `/etc/profile` 文件，设置命令别名（所有用户），如果在某用户下设置，编辑 `~/.bashrc` 文件，添加以下内容：

```
alias vi='vim'
```

```
source /etc/profile # 重载生效
```

配置：

编辑 `~/.vimrc` 文件（只对某用户生效），如果对所有用户生效，在 `/etc/vimrc` 写入相同内容（建议不用）：

```
" 设置tab显示为4个空格的距离
set tabstop=4
" 设置行高亮
set cursorline
" :h highlight查看详细信息
" 行高亮美化(:h highlight查看详细信息)
" CursorLine和CursorColumn表示当前所在的行/列
" cterm表示为原生vim设置样式，设置为NONE表示可以自定义设置
" ctermbg设置终端vim的背景色
" ctermfg设置终端vim的前景色
" guibg和guifg分别是设置gvim的背景色和前景色，使用终端不需要设置
highlight CursorLine cterm NONE ctermbg=black ctermfg=green gu
ibg=NONE guifg=NONE
" 设置列高亮
set cursorcolumn
highlight CursorColumn cterm NONE ctermbg=black ctermfg=green gu
ibg=NONE guifg=NONE
" use extended function of vim (no compatible with vi)
set nocompatible
" specify encoding
set encoding=utf-8
" specify file encoding
set fileencodings=ucs-bom,utf-8,cp936,gb18030,big5,euc-jp,euc-kr
```

1.7. 配置vim

```
,latin1
" specify file formats
set fileformats=unix,dos
" take backup
" if not, specify [ set nobackup ]
set backup
" specify backup directory//指定的存储目录先建好
set backupdir=~/backup
" take 50 search histories
set history=50
" ignore Case
set ignorecase
" distinct Capital if you mix it in search words
set smartcase
" highlights matched words
" if not, specify [ set nohlsearch ]
set hlsearch
" use incremental search
" if not, specify [ set noincsearch ]
set incsearch
" show line number
" if not, specify [ set nonumber ]
set number
" Visualize break ( $ ) or tab ( ^I )//break和tab显示为$和^I(个人习惯不使用)
" set list
" highlights parentheses
set showmatch
" show color display
" if not, specify [ syntax off ]
syntax on
" change colors for comments if it's set [ syntax on ]
highlight Comment ctermfg=LightCyan
" wrap lines
" if not, specify [ set nowrap ]
set wrap
" paste不会改变复制文本的格式,也可以在编辑时在set前加:来临时启用set paste
或set nopaste
set paste
```

各参数的含义可以网上查资料，根据自己需要设置。

1.7. 配置vim

1.8. 配置NTP

1.8.1. 配置NTP服务器

1.8.1.1. NTPd

安装NTPd：

```
yum -y install ntp
```

编辑 /etc/ntp.conf 文件

添加允许接收请求的网络范围：

```
restrict 10.0.0.0 mask 255.255.255.0 nomodify notrap
```

更改同步服务器：

```
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst  
server 0.cn.pool.ntp.org iburst  
server 1.cn.pool.ntp.org iburst  
server 2.cn.pool.ntp.org iburst  
server 3.cn.pool.ntp.org iburst
```

以上服务器地址可以在<http://www.pool.ntp.org/zh/>查找。

启动并设置开机启动：

```
systemctl start ntpd  
systemctl enable ntpd
```

firewalld防火墙设置（NTP使用端口123/UDP）：

```
firewall-cmd --add-service=ntp --permanent  
firewall-cmd --reload
```

验证运行：

```
ntpq -p
```

1.8.1.2. Chrony

Chrony与NTPd选择一个安装。

安装：

```
yum -y install chrony
```

编辑 /etc/chrony.conf 文件

更改同步服务器：

```
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst  
server 0.cn.pool.ntp.org iburst  
server 1.cn.pool.ntp.org iburst  
server 2.cn.pool.ntp.org iburst  
server 3.cn.pool.ntp.org iburst
```

添加允许接收请求的网络范围：

```
allow 10.0.0.0/24
```

启动并设置开机启动：

```
systemctl start chronyd  
systemctl enable chronyd
```

防火墙设置同NTPd。

验证运行：

```
chronyc sources
```

1.8.2. 配置NTP客户端

1.8.2.1. CentOS

CentOS的NTP客户端设置大致与服务器相同，参考[NTPd](#)或[Chrony](#)的设置。与服务器设置不同之处在于，客户端不需要从其他计算机接收时间同步请求，因此它不需要设置访问权限。

使用命令来立即同步时间，按以下操作：

```
yum -y install ntpdate  
ntpdate cn.pool.ntp.org # 系统时间与网络同步  
hwclock --systohc # 将时间写入硬件  
systemctl start ntpdate  
systemctl enable ntpdate
```

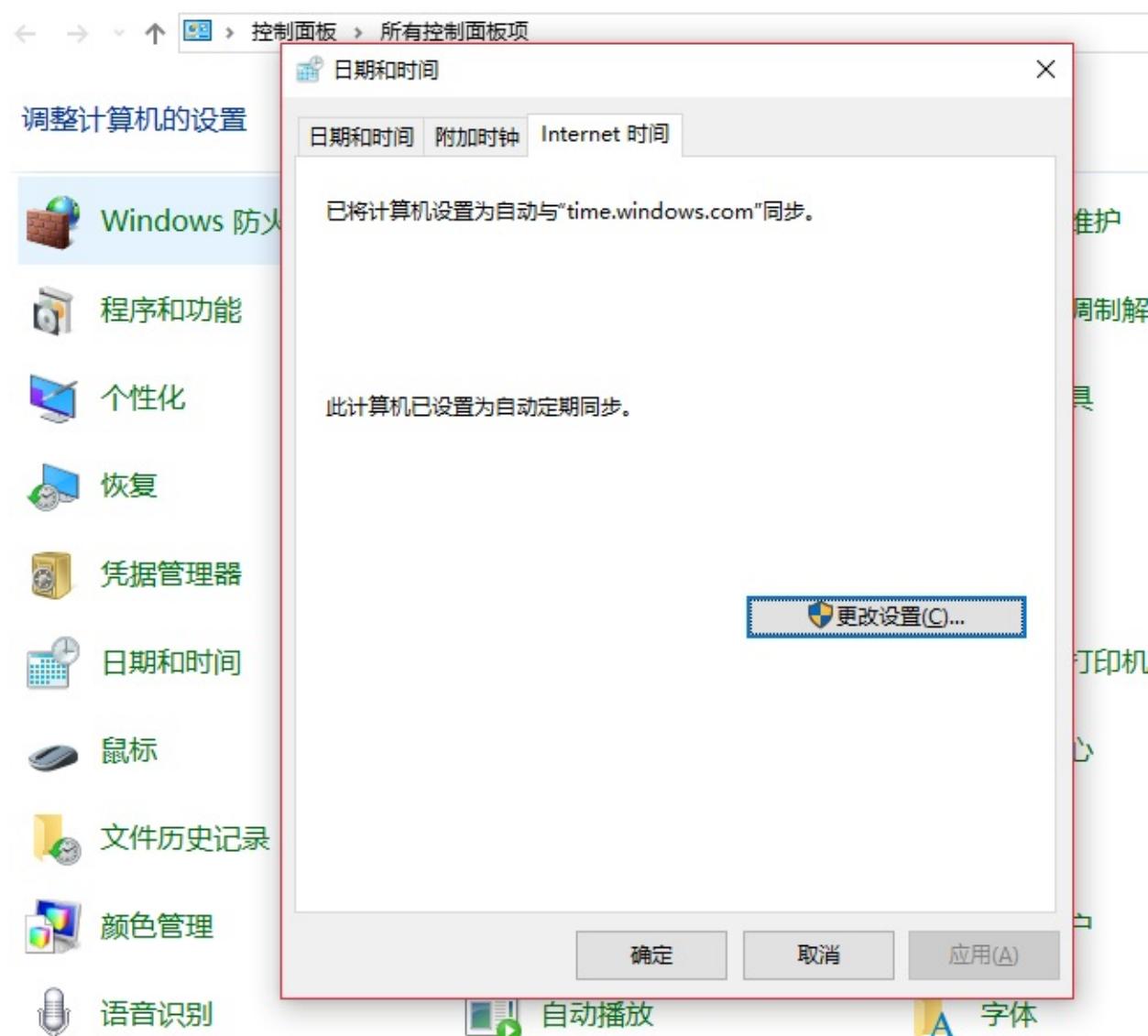
以上为[Server World](#)内容，下面是收集的一些关于时间的命令：

```
timedatectl # 查看系统时间  
timedatectl list-timezones # 列出可用时区  
timedatectl set-timezone Asia/Shanghai # 设置时区（上海）  
timedatectl set-ntp yes # 同步系统时间
```

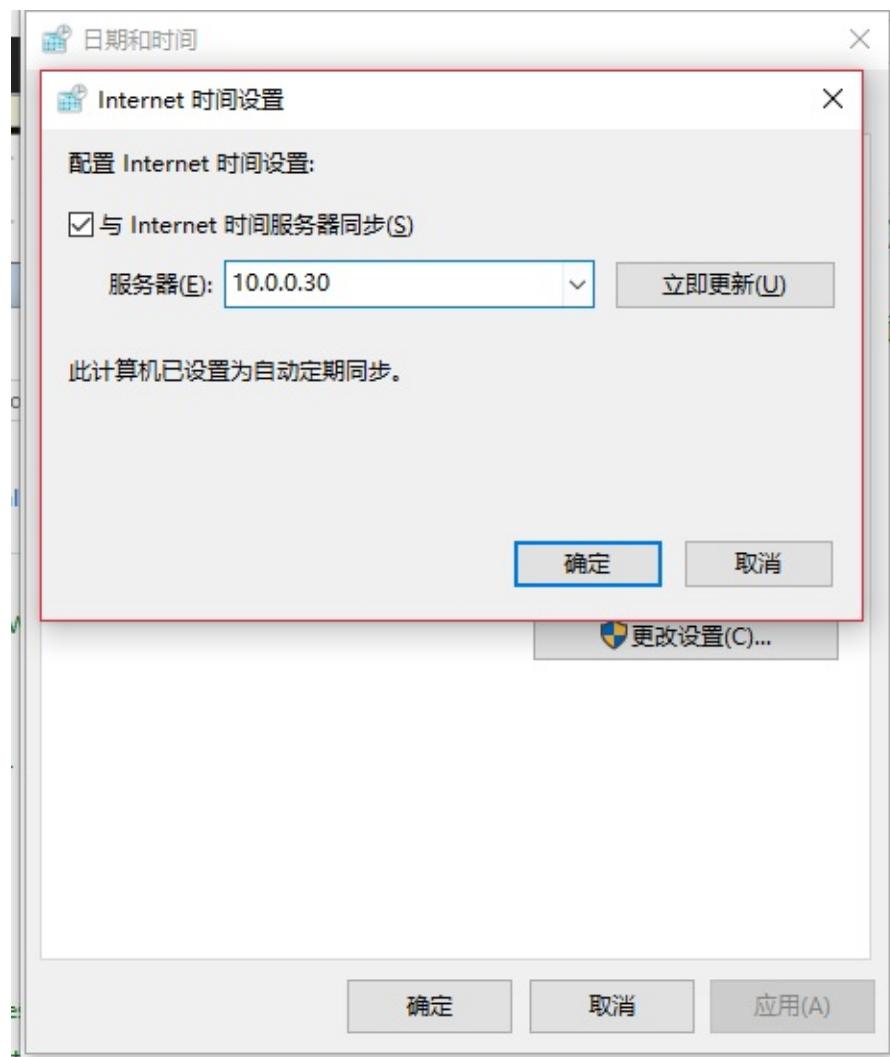
1.8.2.2. Windows

打开Windows“控制面板”->“日期和时间”->“Internet时间”->“更改设置”->“立即更新”：

1.8. 配置NTP



1.8. 配置NTP



配置Windows的NTP客户端服务，到[这里](#)查看详细步骤

1.9. 配置SSH

1.9.1. 配置SSH服务器

1.9.1.1. 密码验证登录

CentOS默认安装[OpenSSH](#)，编辑 `/etc/ssh/sshd_config` 文件，做一些安全设置（根据需要修改）：

禁止root远程登录，将 `PermitRootLogin` 一行取消注释，并修改为：

```
PermitRootLogin no
```

禁止空密码，使用密码验证登录：

```
PermitEmptyPasswords no
PasswordAuthentication yes
```

```
systemctl restart sshd # 重启服务以生效
```

`firewall`防火墙设置（SSH默认端口22/TCP）：

```
firewall-cmd --add-service=ssh --permanent
firewall-cmd --reload
```

修改端口：

编辑 `/etc/ssh/sshd_config` 文件，`Port 22` 取消注释，并添加一行想要使用的端口（不要和服务器其他常用端口冲突）如10000，则添加 `Port 10000`，将对应的防火墙规则先添加好。运行 `systemctl restart sshd` 重启SSH后测试是否可以通过添加的端口连接。可以连接后将 `Port 22` 重新注释并重启SSH。

1.9.1.2. 密钥验证登录

为客户端创建一个私钥，并为服务器创建一个公钥。

为每个用户创建密钥对，因此使用普通用户登录并按如下操作：

1.9. 配置SSH

```
ssh-keygen -t rsa # 创建密钥对
```

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/cent/.ssh/id_rsa): #  
回车  
Created directory '/home/cent/.ssh'.  
Enter passphrase (empty for no passphrase): # 设置密码短语 (没有密  
码短语则直接回车)  
Enter same passphrase again: # 确认密码短语  
Your identification has been saved in /home/cent/.ssh/id_rsa.  
Your public key has been saved in /home/cent/.ssh/id_rsa.pub.  
The key fingerprint is:  
38:f1:b4:6d:d3:0e:59:c8:fa:1d:1d:48:86:f0:fe:74 cent@dlp.srv.wor  
ld  
The key's randomart image is:
```

```
mv ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

```
chmod 600 ~/.ssh/authorized_keys # 更改文件权限
```

将在服务器上创建的密钥传输到客户端，然后可以使用密钥身份验证登录：

```
mkdir ~/.ssh
```

```
chmod 700 ~/.ssh
```

```
scp cent@10.0.0.30:/home/cent/.ssh/id_rsa ~/.ssh/ # 将密钥复制到本地  
ssh 目录
```

```
cent@10.0.0.30's password:  
id_rsa
```

```
ssh -i ~/.ssh/id_rsa cent@10.0.0.30 # 使用密钥验证登录
```

```
Enter passphrase for key '/home/cent/.ssh/id_rsa': # 之前设置的密  
码短语  
Last login: Wed Jul 30 21:37:19 2014 from www.srv.world
```

将 `/etc/ssh/sshd_config` 文件中的 `PasswordAuthentication` 设置为 `no`，禁止密码验证登录，会更安全。

Windows客户端比较简单，在此不作介绍了。

1.9.2. 配置SSH客户端

1.9.2.1. CentOS客户端

```
yum -y install openssh-clients
```

使用普通用户连接到SSH服务器：

```
ssh cent@dlp.srv.world # ssh [用户名@域名或IP]
```

```
The authenticity of host 'dlp.srv.world (<no hostip for proxy command>)' can't be established.  
ECDSA key fingerprint is xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:  
60:90:d8.  
Are you sure you want to continue connecting (yes/no)? yes # 输入yes确认  
Warning: Permanently added 'dlp.srv.world' (ECDSA) to the list o  
f known hosts.  
cent@dlp.srv.world's password: # 用户cent的密码
```

可以使用SSH在远程主机上执行命令，如下所示：

```
ssh cent@dlp.srv.world "cat /etc/passwd" # 在远程主机上执行 cat  
/etc/passwd
```

```
cent@dlp.srv.world's password:  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
...  
...  
postfix:x:89:89::/var/spool/postfix:/sbin/nologin  
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nolog  
in
```

1.9.2.2. Windows客户端

Windows下SSH客户端很多，推荐[Xshell](#)，使用方法就不多说了。

1.9.3. SSH文件传输

1.9.3.1. CentOS客户端

使用SCP (Secure Copy)

命令格式： `scp [选项] 源文件 目标`

```
scp ./test.txt cent@www.srv.world:~/ # 将本地的 test.txt 复制到远程
服务器 www.srv.world
```

```
cent@www.srv.world's password: # 输入远程主机cent用户的密码
100%    10    0.0KB/s   00:00
```

```
scp cent@www.srv.world:/home/cent/test.txt ./test.txt # 将远程服务
器 www.srv.world 上的 /home/cent/test.txt 复制到本地
```

```
cent@prox.srv.world's password:# 输入远程主机cent用户的密码
test.txt    100%    10    0.0KB/s   00:00
```

使用SFTP (SSH File Transfer Protocol)

SFTP服务器默认启用，如果没有启用，编辑 `/etc/ssh/sshd_config` 文件，添加一行内容：`Subsystem sftp /usr/libexec/openssh/sftp-server`，然后重启SSH服务。

命令格式： `sftp [选项] [用户名@远程主机域名或IP]`

```
sftp cent@www.srv.world
```

```
cent@www.srv.world's password: # 输入远程主机cent用户的密码
Connected to prox.srv.world.
sftp>
```

```
sftp> pwd # 显示远程服务器上的当前目录
Remote working directory: /home/cent
```

1.9. 配置SSH

```
sftp> !pwd # 显示本地服务器上的当前目录  
/home/redhat
```

```
sftp> ls -l # 在FTP服务器上显示当前目录中的文件  
drwxrwxr-x 2 cent cent 6 Jul 29 21:33 public_html  
-rw-rw-r-- 1 cent cent 10 Jul 28 22:53 test.txt
```

```
sftp> !ls -l # 在本地服务器上的当前目录中显示文件  
total 4  
-rw-rw-r-- 1 redhat redhat 10 Jul 29 21:31 test.txt
```

```
sftp> cd public_html # 更改目录  
sftp> pwd  
Remote working directory: /home/cent/public_html
```

```
sftp> put test.txt redhat.txt # 将文件上传到远程服务器  
Uploading test.txt to /home/cent/redhat.txt  
test.txt 100% 10 0.0KB/s 00:00  
sftp> ls -l  
drwxrwxr-x 2 cent cent 6 Jul 29 21:33 public_html  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:39 redhat.txt  
-rw-rw-r-- 1 cent cent 10 Jul 28 22:53 test.txt
```

```
sftp> put *.txt # 将多个文件上传到远程服务器  
Uploading test.txt to /home/cent/test.txt  
test.txt 100% 10 0.0KB/s 00:00  
Uploading test2.txt to /home/cent/test2.txt  
test2.txt 100% 0 0.0KB/s 00:00  
sftp> ls -l  
drwxrwxr-x 2 cent cent 6 Jul 29 21:33 public_html  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:39 redhat.txt  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:45 test.txt  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:46 test2.txt
```

1.9. 配置SSH

```
sftp> get test.txt # 从远程服务器下载文件  
Fetching /home/cent/test.txt to test.txt  
/home/cent/test.txt 100% 10 0.0KB/s 00:00
```

```
sftp> get *.txt # 从远程服务器下载多个文件  
Fetching /home/cent/redhat.txt to redhat.txt  
/home/cent/redhat.txt 100% 10 0.0KB/s 00:00  
Fetching /home/cent/test.txt to test.txt  
/home/cent/test.txt 100% 10 0.0KB/s 00:00  
Fetching /home/cent/test2.txt to test2.txt  
/home/cent/test2.txt 100% 10 0.0KB/s 00:00
```

```
sftp> mkdir testdir # 在远程服务器上创建目录  
sftp> ls -l  
drwxrwxr-x 2 cent cent 6 Jul 29 21:33 public_html  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:39 redhat.txt  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:45 test.txt  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:46 test2.txt  
drwxrwxr-x 2 cent cent 6 Jul 29 21:53 testdir
```

```
sftp> rmdir testdir # 删除远程服务器上的目录  
rmdir ok, `testdir' removed  
sftp> ls -l  
drwxrwxr-x 2 cent cent 6 Jul 29 21:33 public_html  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:39 redhat.txt  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:45 test.txt  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:46 test2.txt
```

```
sftp> rm test2.txt # 删除远程服务器上的文件  
Removing /home/cent/test2.txt  
sftp> ls -l  
drwxrwxr-x 2 cent cent 6 Jul 29 21:33 public_html  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:39 redhat.txt  
-rw-rw-r-- 1 cent cent 10 Jul 29 21:45 test.txt
```

```
sftp> !cat /etc/passwd # 使用“!命令”来执行命令  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
...  
...  
redhat:x:1001:1001::/home/redhat:/bin/bash
```

```
sftp> quit # 退出  
221 Goodbye.
```

1.9.3.2. Windows客户端

推荐[Xftp](#)，使用方法也不多说了。

1.9.3.3 仅SFTP + Chroot

应用此设置的某些用户只能使用SFTP访问和访问允许的目录。

例如，将/home设置为Chroot目录：

```
groupadd sftp_users # 为SFTP创建一个组
```

```
usermod -G sftp_users cent # 仅用于SFTP的用户“cent”
```

编辑 `/etc/ssh/sshd_config` 文件：

```
# 注释下面一行，并添加上第二行  
#Subsystem sftp /usr/libexec/openssh/sftp-server  
Subsystem sftp internal-sftp  
  
# 添加以下内容  
Match Group sftp_users  
    X11Forwarding no  
    AllowTcpForwarding no  
    ChrootDirectory /home  
    ForceCommand internal-sftp
```

```
systemctl restart sshd # 重启服务以生效
```

尝试使用用户访问并确保设置：

```
ssh cent@10.0.0.30 # 通过SSH客户端访问
```

```
cent@10.0.0.30's password: # 输入密码  
This service allows sftp connections only.  
Connection to 10.0.0.30 closed. # 拒绝访问
```

```
sftp cent@10.0.0.30 # 通过SFTP客户端访问
```

```
Connecting to 10.0.0.30...  
cent@10.0.0.30's password:  
sftp> ls -l  
drwx----- 3 1000 1000 4096 Jul 9 12:06 cent  
drwx----- 2 1001 1001 59 Jul 8 22:06 hirokun  
sftp> pwd  
Remote working directory: /  
sftp> exit
```

1.9.4. SSH端口转发

SSH端口转发可以将一个端口转发到另一个端口。

例如，配置转发设置，将本地的8081转发到本地的5901（VNC）。此示例显示简单的设置，但它可以将大多数端口转发到本地或其他服务器上的大多数端口。

启动[VNC](#)，然后按以下操作：

```
ssh -L 0.0.0.0:8081:localhost:5901 cent@localhost # 将到本地8081的  
连接转发到5901
```

```
cent@localhost's password: # 用户cent的密码（登录到本地）  
Last login: Thu Jul 10 01:35:15 2014
```

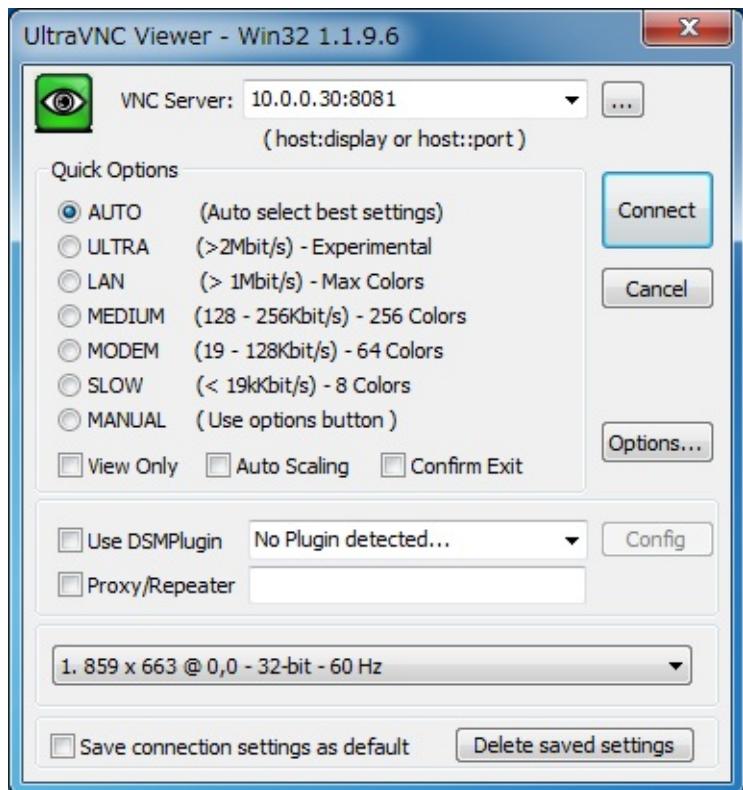
```
netstat -lnp | grep 8081 # netstat -lnp | grep 8081
```

1.9. 配置SSH

```
(Not all processes could be identified, non-owned process info  
will not be shown, you would have to be root to see it all.)  
tcp 0 0 0.0.0.0:8081 0.0.0.0:* LISTEN 3238/ssh
```

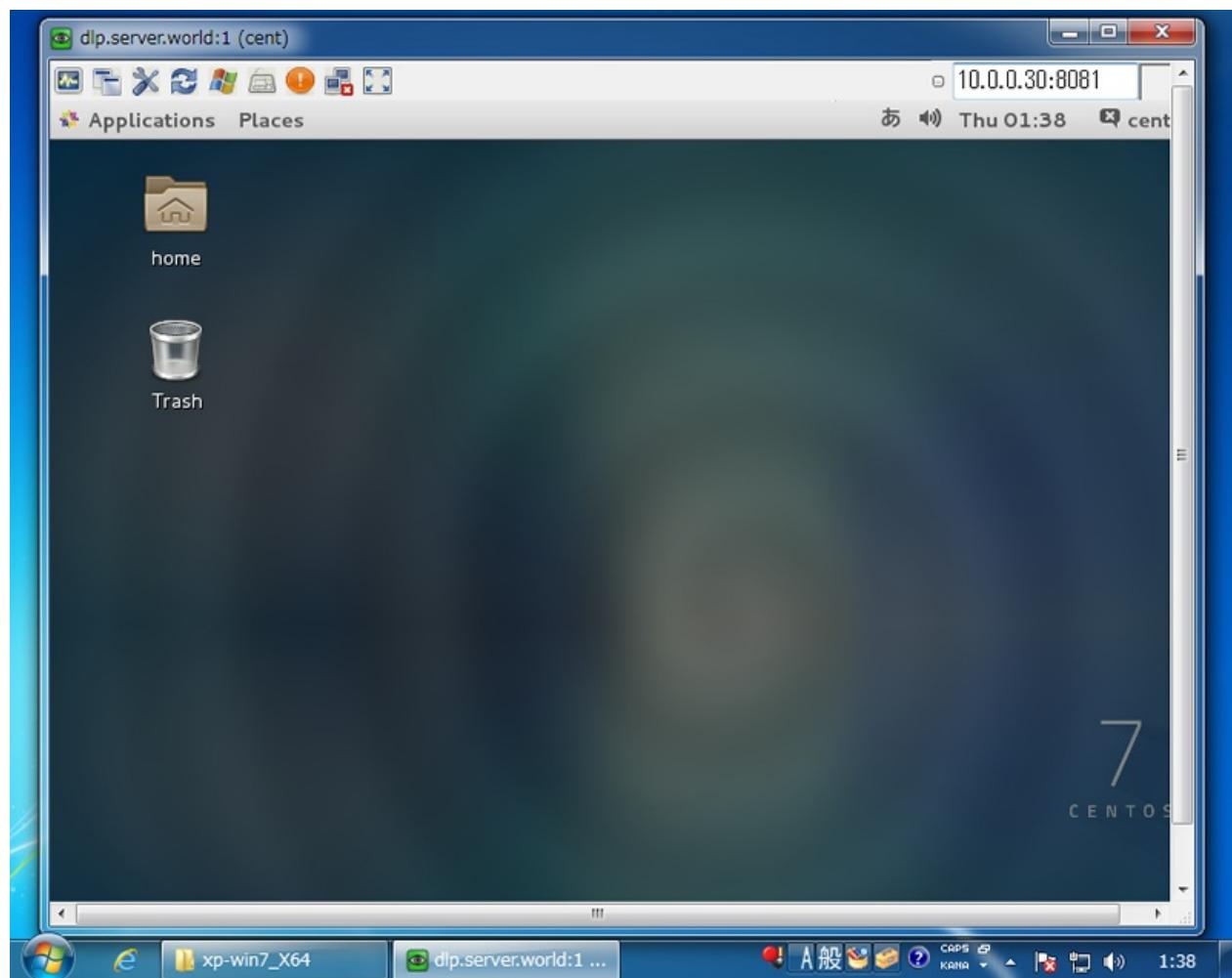
保持这个会话，进行下一步。可以使用 `-f` 选项在后台启动进程作为一个守护进程，但使用完成后需要手动结束进程。

连接到本地配置的端口：



连接成功：

1.9. 配置SSH



使用客户端工具进行端口转发

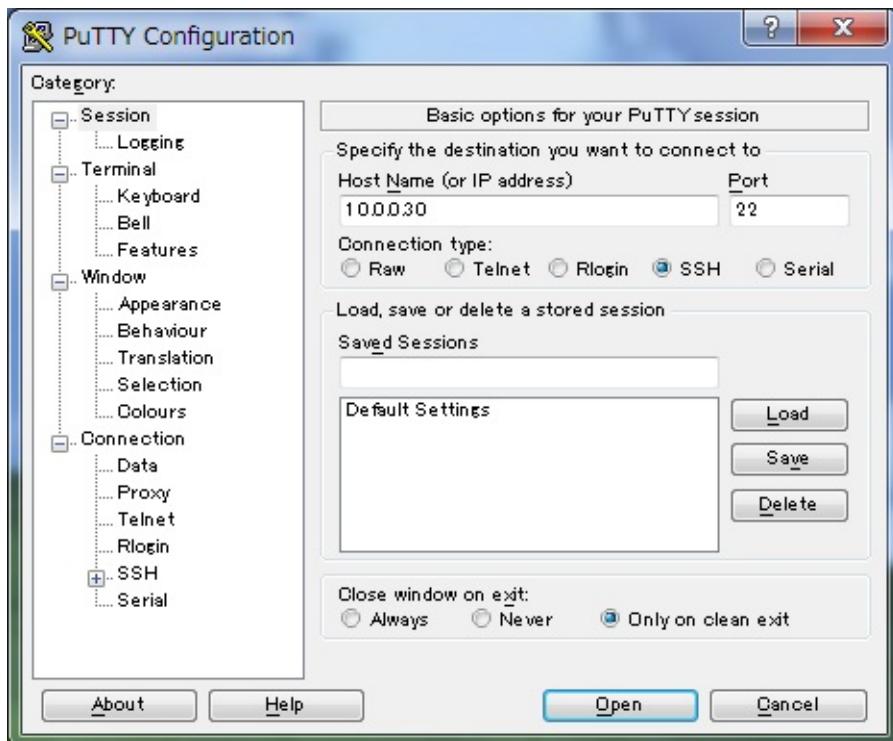
可以使用具有端口转发功能的工具，而无需配置服务器。

例如，使用Putty的端口转发连接到VNC服务器。

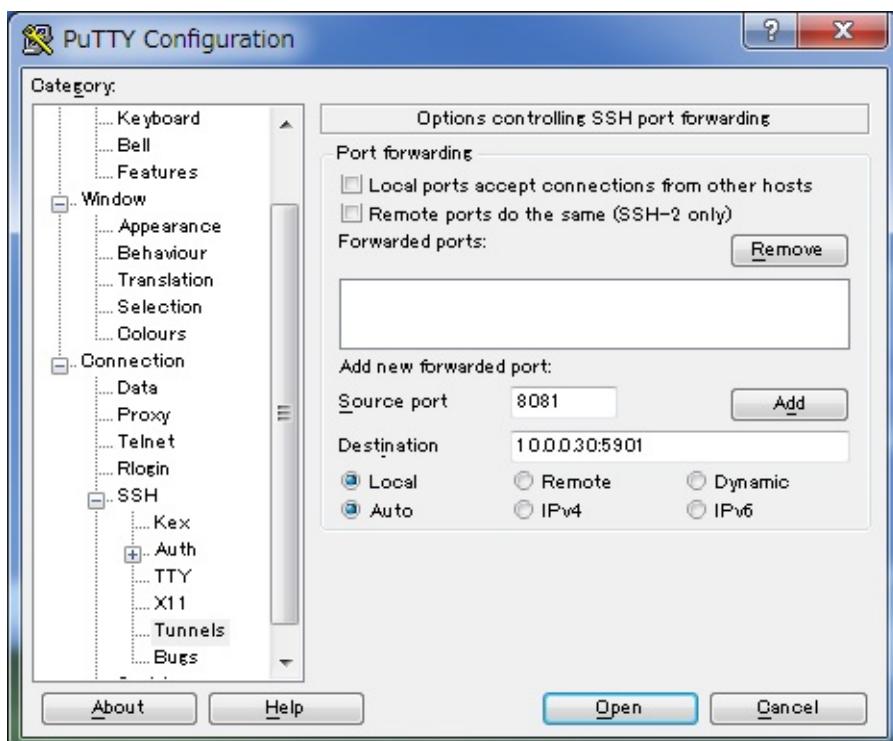
[启动VNC](#)。

在客户端电脑上启动Putty并指定目标服务器：

1.9. 配置SSH

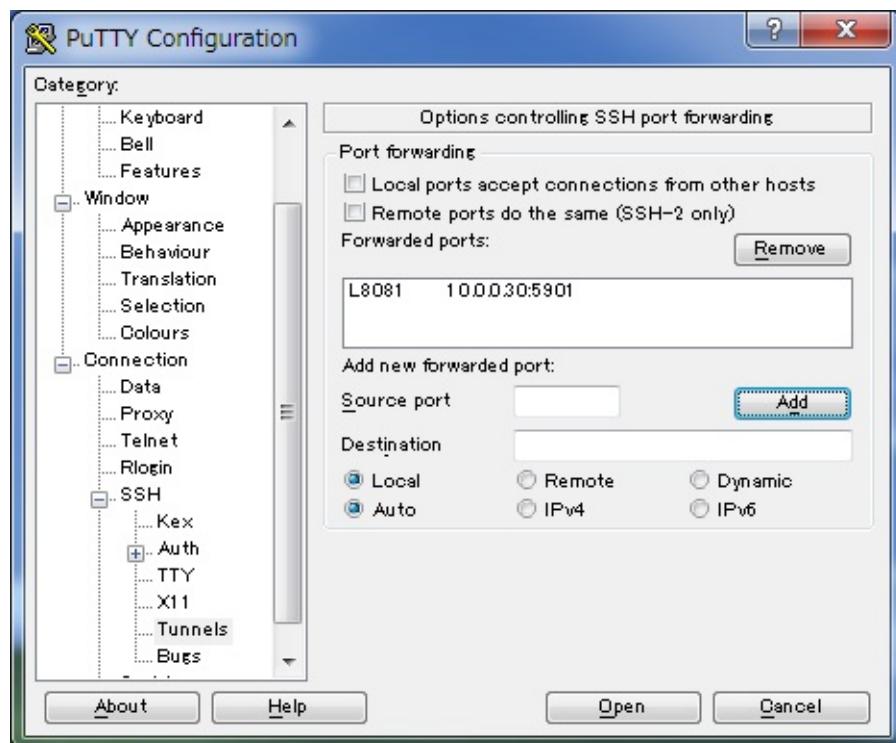


在左侧菜单中选择“Connection”->“SSH”->“Tunnels”，在“Source port”字段上输入本地PC上任何可用的端口，在“目标服务器”字段输入“目标服务器:端口”。然后，点击“Add”按钮：



确保设置已添加，然后单击“Open”按钮进行连接：

1.9. 配置SSH

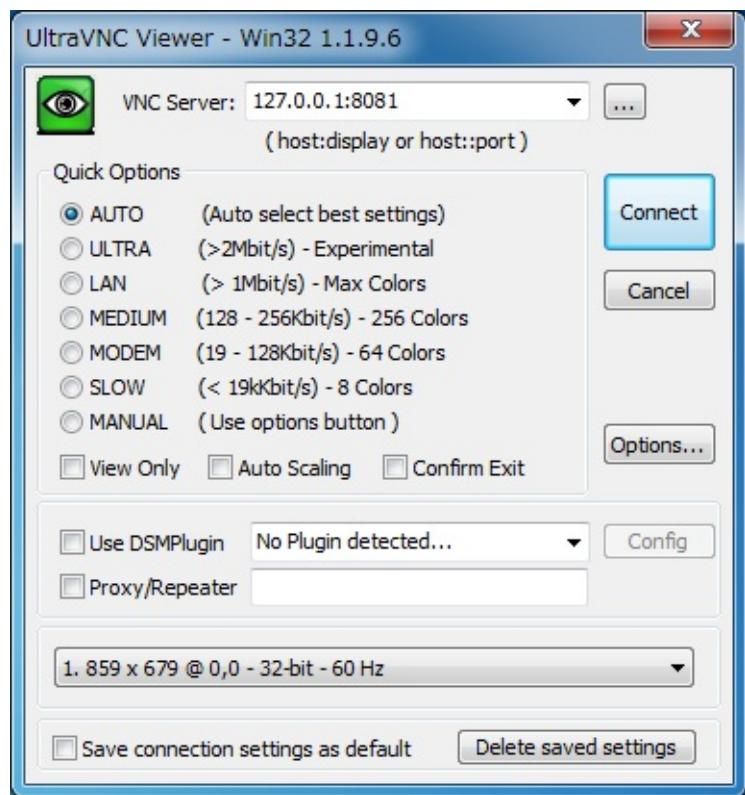


登录服务器并保持会话，然后进入下一步：

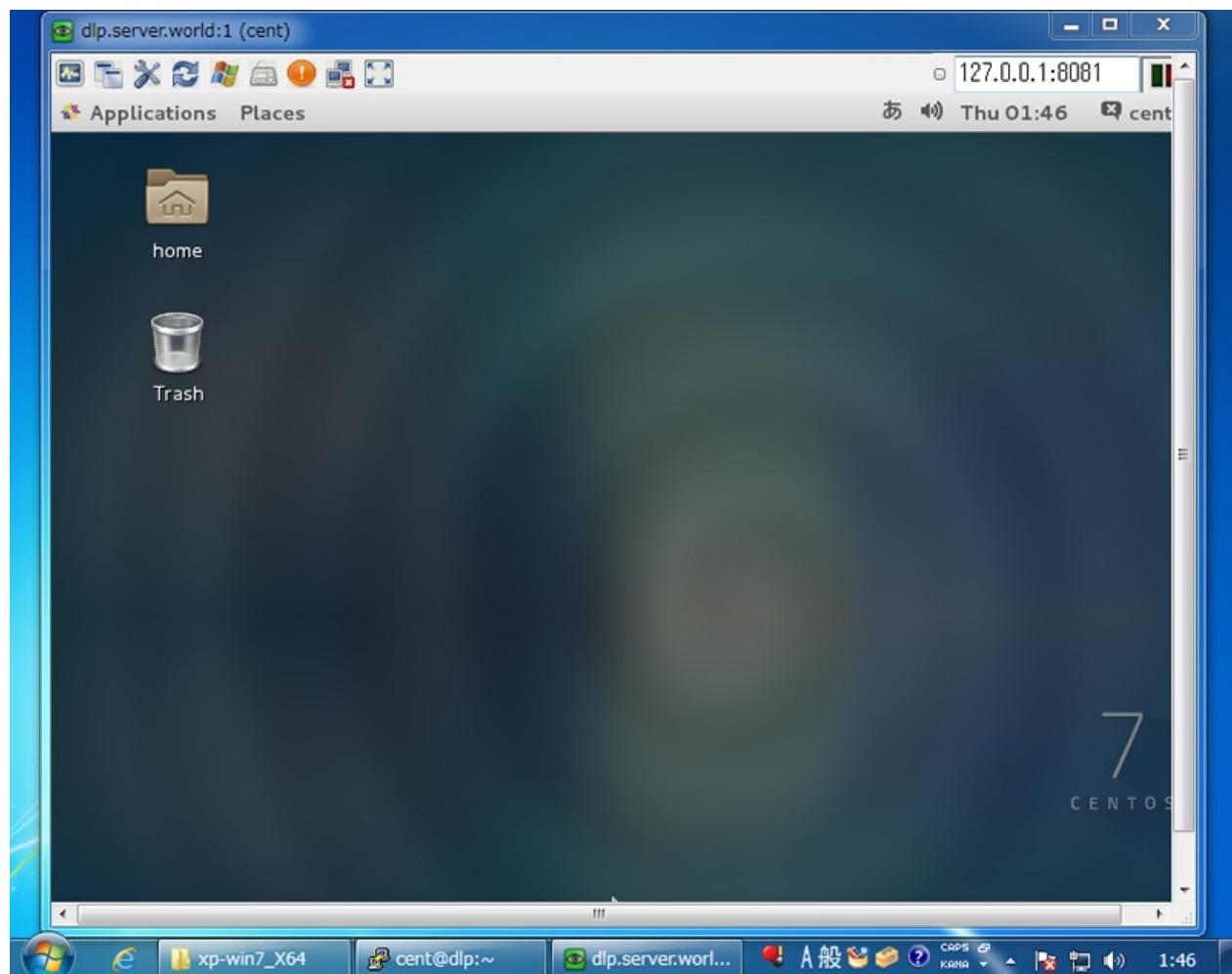
The screenshot shows a terminal window with the title bar 'cent@dlp:~'. The window displays the following text:
login as: cent
cent@10.0.0.30's password:
Last login: Thu Jul 10 01:37:49 2014 from localhost
[cent@dlp ~]\$

在客户端电脑上启动VNC客户端，并连接到“[localhost]:[之前设置为source port的端口]”：

1.9. 配置SSH



连接成功：



1.9.5. X11转发

可以使用SSH X11转发在本地客户端显示和使用远程服务器上的GUI应用程序。

在SSH服务器上启用X11转发功能：

编辑 `/etc/ssh/sshd_config` 文件：

```
# 取消注释  
X11Forwarding yes  
X11DisplayOffset 10
```

```
systemctl restart sshd
```

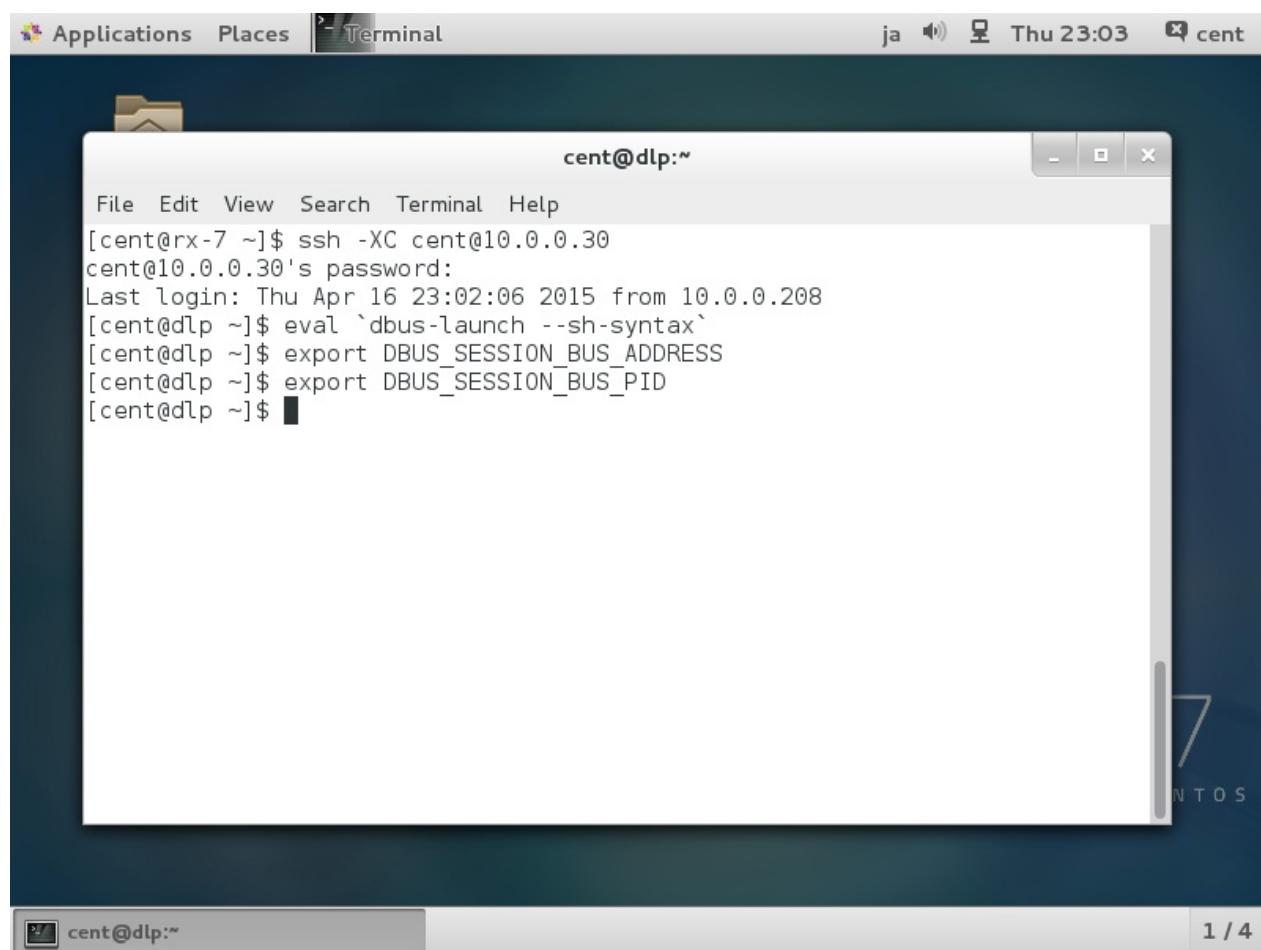
1.9.5.1. CentOS客户端

这是在[安装了桌面环境的CentOS客户端](#)上使用GUI应用程序的示例。

启动终端（Terminal）并使用 `ssh -XC xxx` 连接到启用了X11转发的SSH服务器，连接后，输入如下命令：

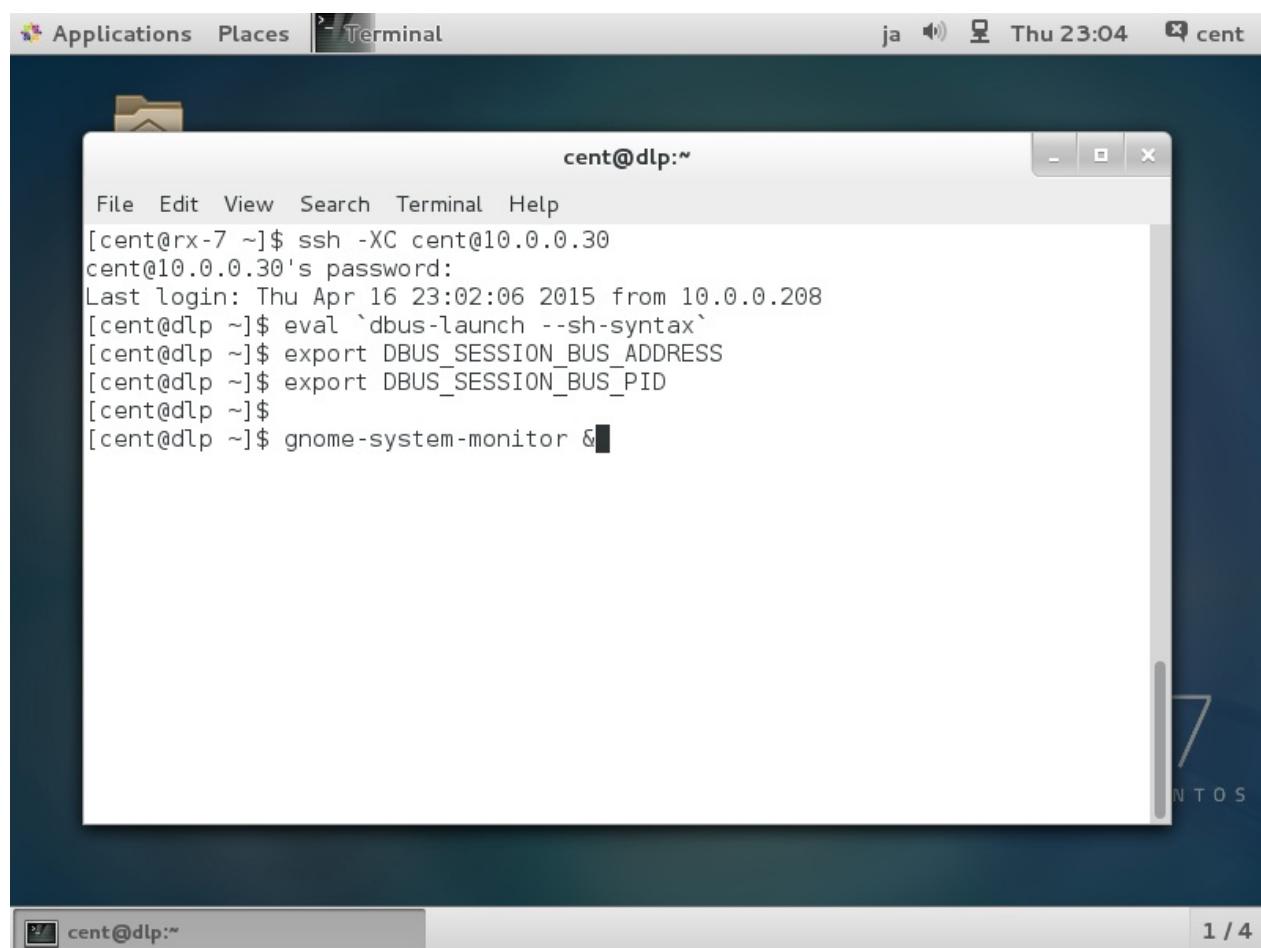
```
eval `dbus-launch --sh-syntax`  
export DBUS_SESSION_BUS_ADDRESS  
export DBUS_SESSION_BUS_PID
```

1.9. 配置SSH

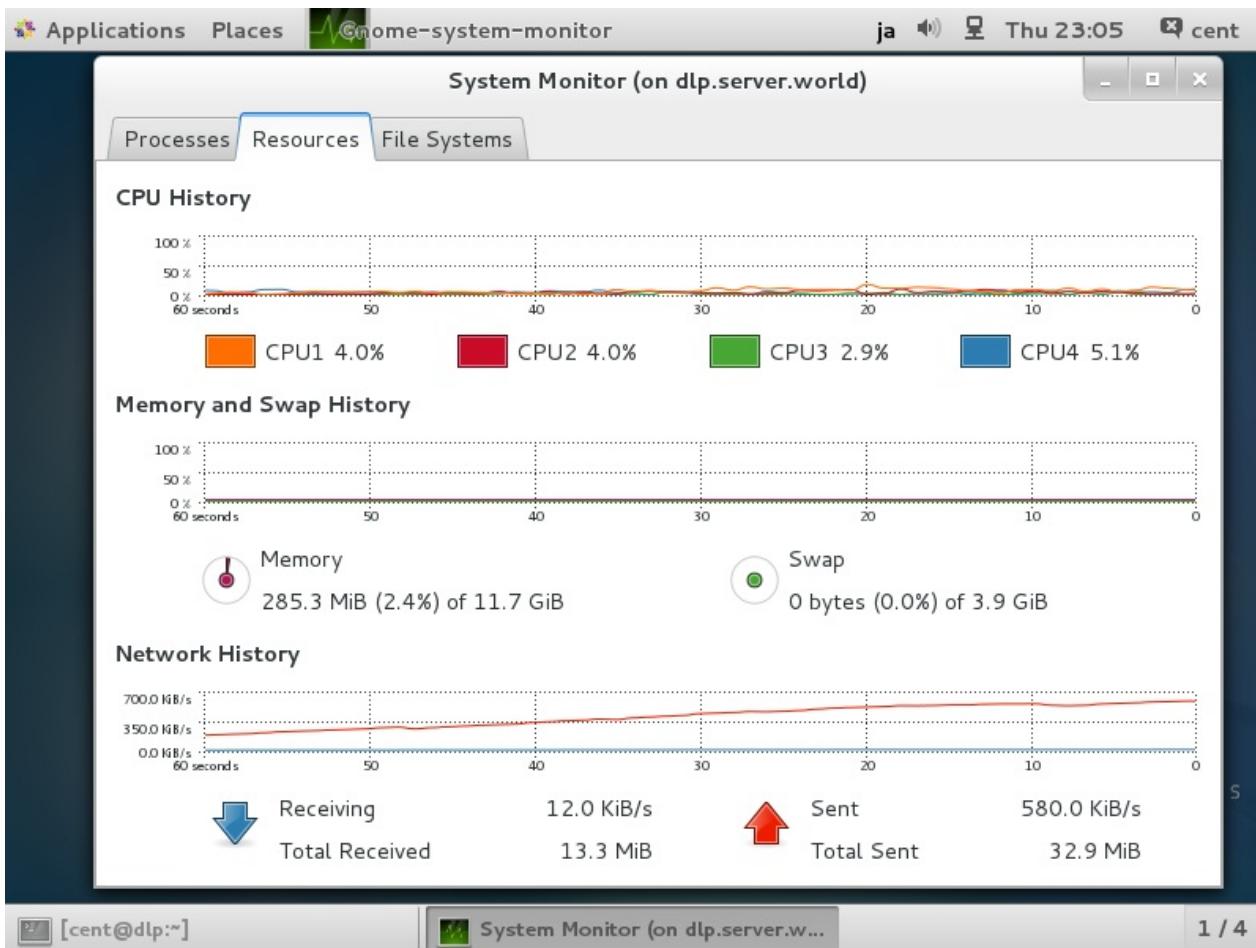


接下来，执行想要的GUI应用程序。例如，启动系统监视器（System Monitor）：

1.9. 配置SSH



显示在远程服务器上的系统监视器：



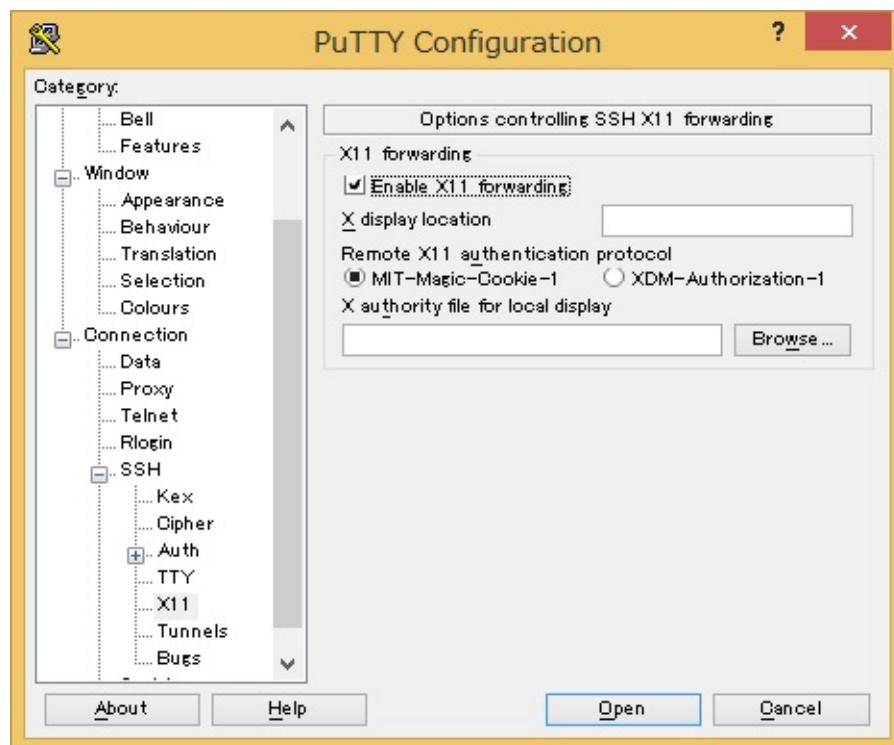
1.9.5.2. Windows客户端

演示使用Windows 8.1和Putty为例。

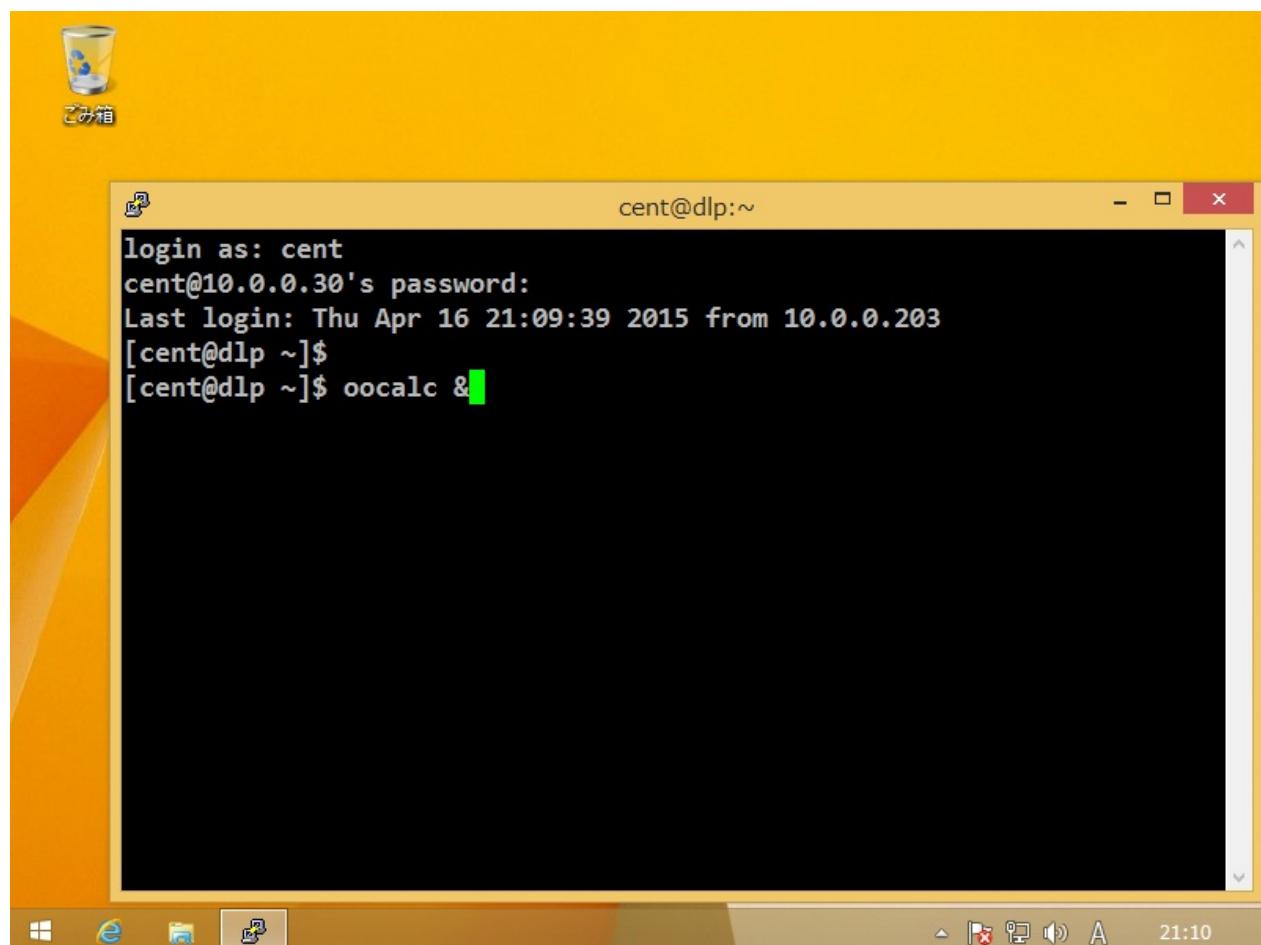
首先[下载并安装Xming](#)（可默认安装全部）。

安装Xming后，启动Putty，选择左侧菜单上的“X11”，然后在右侧窗格中选中“Enable X11 forwarding”复选框：

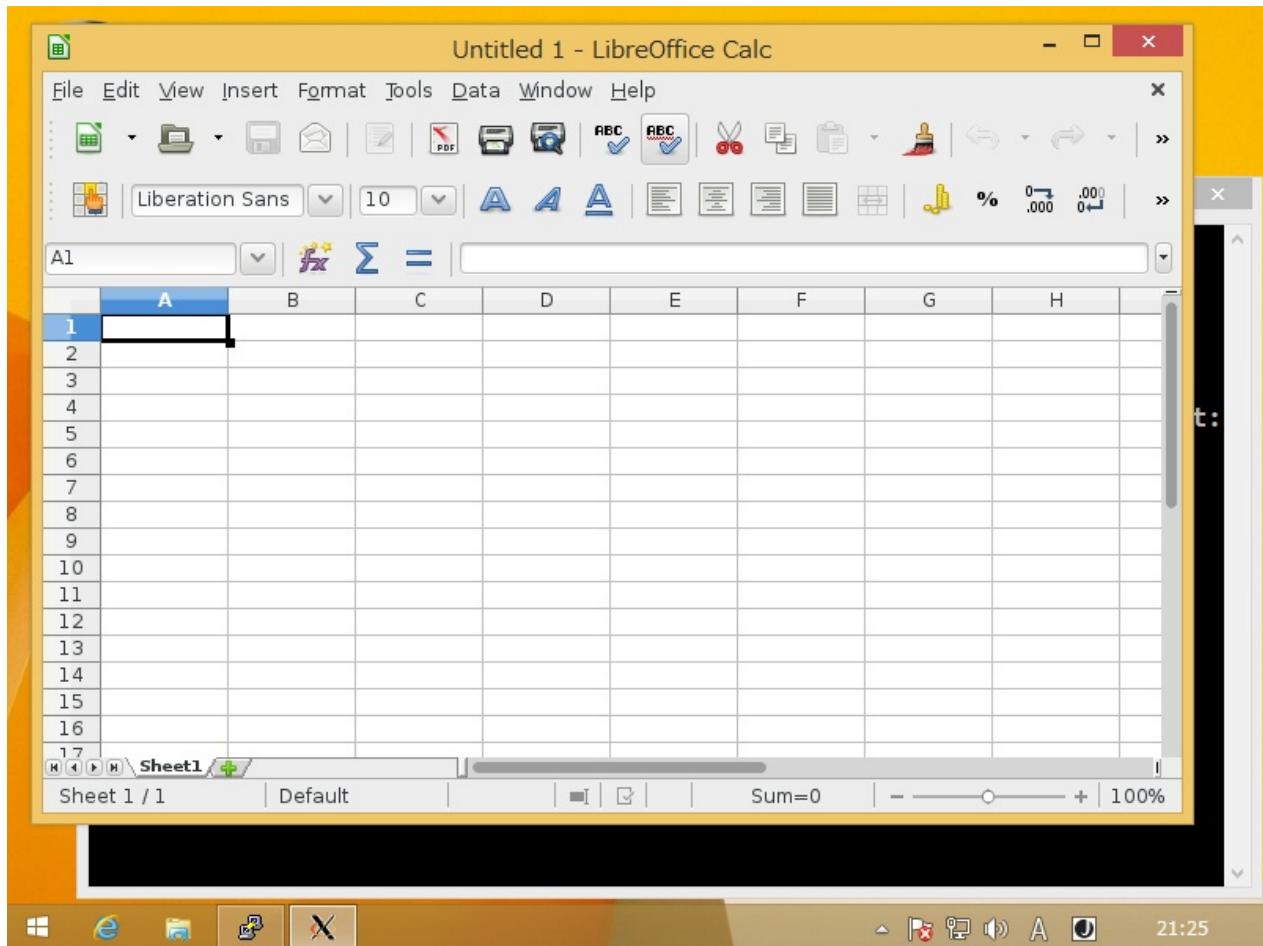
1.9. 配置SSH



登录到远程服务器后，执行想要的GUI应用程序。例如，启动Libre Office Calc：



显示在远程服务器上的Libre Office Calc：



1.9.6. SSHPass

使用SSHPass自动输入口令验证密码。这很方便，但有安全隐患（密码泄漏），如果使用，请特别小心。

安装SSHPass：

```
yum --enablerepo=epel -y install sshpass # 从EPEL安装
```

使用SSHPass：

-p password (从参数) :

```
sshpass -p password ssh 10.0.0.51 hostname #
```

node01.srv.world

-f file (从文件) :

```
echo 'password' > sshpass.txt
```

1.9. 配置SSH

```
chmod 600 sshpass.txt
```

```
sshpass -f sshpass.txt ssh 10.0.0.51 hostname
```

```
node01.srv.world
```

-e (从环境变量) :

```
export SSHPASS=password
```

```
sshpass -e ssh 10.0.0.51 hostname
```

```
node01.srv.world
```

1.9.7. SSH-Agent

使用SSH-Agent在密钥对认证上自动输入密码。

必须首先[设置密钥对](#)。

使用SSH-Agent :

启动SSH-Agent :

```
eval `ssh-agent`
```

```
Agent pid 2168
```

```
ssh-add # 添加身份
```

```
Enter passphrase for /home/cent/.ssh/id_rsa:  
Identity added: /home/cent/.ssh/id_rsa (/home/cent/.ssh/id_rsa)
```

```
ssh-add -l # 确认
```

```
2048 xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:4c:b2 /home/cent/  
.ssh/id_rsa (RSA)
```

1.9. 配置SSH

```
ssh node01.srv.world hostname # 尝试不输入密码连接SSH
```

```
node01.srv.world
```

退出SSH-Agent：

```
eval `ssh-agent -k`
```

```
Agent pid 2168 killed
```

1.9.8. Parallel SSH

安装Parallel SSH以连接到多个主机。

安装PSSH：

```
yum --enablerepo=epel -y install pssh # 从EPEL安装
```

使用PSSH：

这里演示没有密码短语的密钥对认证，如果在密钥对中设置密码，首先启动[SSH-Agent](#)自动输入密码。

```
pssh -H "10.0.0.51 10.0.0.52" -i "hostname" # 连接到主机并执行 hostname 命令
```

```
[1] 17:28:02 [SUCCESS] 10.0.0.51  
node01.srv.world  
[2] 17:28:02 [SUCCESS] 10.0.0.52  
node02.srv.world
```

可以从文件中读取主机列表：

编辑 `pssh_hosts.txt` 文件：

```
# 如下在每行写入主机  
cent@10.0.0.51  
cent@10.0.0.52
```

```
pssh -h pssh_hosts.txt -i "uptime"
```

```
[1] 19:37:59 [SUCCESS] cent@10.0.0.52  
19:37:59 up 1:35, 0 users, load average: 0.00, 0.00, 0.00  
[2] 19:37:59 [SUCCESS] cent@10.0.0.51  
19:37:59 up 1:35, 0 users, load average: 0.00, 0.00, 0.00
```

也可以使用密码认证连接，但需要所有主机密码都是一样的：

```
pssh -h pssh_hosts.txt -A -O PreferredAuthentications=password -i  
"uname -r"
```

```
Warning: do not enter your password if anyone else has superuser  
privileges or access to your account.  
Password: # 输入密码  
[1] 12:54:06 [SUCCESS] cent@10.0.0.51  
2.6.32-504.12.2.el6.x86_64  
[2] 12:54:06 [SUCCESS] cent@10.0.0.52  
2.6.32-504.12.2.el6.x86_64
```

注：PSSH软件包括 `pscp.pssh` , `prsync` , `pslurp` , `pnuke` 等命令，与 `pssh` 用法相同。

1.9.9. 微信提醒

（企业微信官方接口文档：<https://qydev.weixin.qq.com/wiki/index.php>；脚本参考<https://blog.csdn.net/bwlab/article/details/50725335>和<https://my.oschina.net/u/3658138/blog/1586428>）（很多内容不懂，所以只测试了能正常发送）。

先注册企业微信，创建一个应用，在应用的可见范围设置权限。

远程登陆SSH时，微信提醒

1.9. 配置SSH

新建文件 /etc/ssh/sshrc :

```
#!/bin/bash

CorpID='      ' # 填入企业ID
Secret='      ' # 填入应用Secret
GURL="https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=$CorpID&corpsecret=$Secret"

# get acccess_token
GToken=`/usr/bin/curl -s -G $GURL`
Token=`echo $GToken |awk -F '"' '{print $10}'` 
PURL="https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token=$Token"

wxAppID=xxxxxx # 填入应用AgentId
wxUserID=1 # 企业微信中部门成员ID(企业微信成员信息中称为帐号)
Last=`last` 
IP=`echo $Last |awk -F ' ' '{print $3}'` 
Time=`date +%Y-%m-%d %H:%M:%S` 
wxMsg='服务器登陆提醒：\n主机名：'`hostname`'\n登录用户：'`whoami`'\n
登录IP：'$IP`\n登录时间：'$Time` 
Body='{ "touser":"'${wxUserID}'", "msgtype":"text", "agentid":"'${wxAppID}'", "text":{ "content":"'${wxMsg}'"}, "safe":"0" }'

/usr/bin/curl --data-ascii "$Body" $PURL >/dev/null 2>&1
```

可以不用 chmod +x , 登录ssh时会先执行此脚本内容。

2. 虚拟化

CentOS7可以使

用KVM，oVirt，Xen，Docker，Kubernetes，VirtualBox，VMware等虚拟化系统。

- 2.1. KVM

- 2.1.1. 安装KVM
- 2.1.2. 创建一个虚拟机
- 2.1.3. 图形界面管理
- 2.1.4. 基本操作
- 2.1.5. 虚拟管理工具
- 2.1.6. 实时迁移
- 2.1.7. SPICE服务器
- 2.1.8. SPICE客户端
 - 2.1.8.1. CentOS客户端
 - 2.1.8.2. Windows客户端
- 2.1.9. KVM嵌套
- 2.1.10. 快照管理

- 2.2. oVirt

- 2.2.1. 配置控制服务器
- 2.2.2. 配置节点
- 2.2.3. 添加管理的目标节点
- 2.2.4. 添加存储
- 2.2.5. 创建虚拟机

- 2.3. Xen

- 2.3.1. 安装Xen
- 2.3.2. 创建虚拟机

- 2.4. Docker

- 2.4.1. 安装Docker
- 2.4.2. 添加镜像
- 2.4.3. 访问容器
- 2.4.4. 使用Dockerfile
- 2.4.5. 使用Docker-Registry
- 2.4.6. 持久化存储

- 2.5. Kubernetes

- 2.5.1. 配置管理节点
- 2.5.2. 配置容器节点
- 2.5.3. 创建Pod
- 2.5.4. 持久化存储

2.1. KVM

KVM（Kernel-based Virtual Machine）是Linux下x86硬件平台上的全功能虚拟化解决方案，包含一个可加载的内核模块 kvm.ko 提供和虚拟化核心架构和处理器规范模块。要求计算机CPU具有Intel VT或AMD-V功能。

```
egrep '^flags.*(vmx|svm)' /proc/cpuinfo # 运行以验证是否支持（运行后有显示则支持）。
```

```
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx m
mxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow constant_tsc rep
_good nonstop_tsc extd_apicid pni monitor cx16 popcntlahf_lm cm
p_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefe
tch osvw ibs skinit wdt npt lbrv svm_lock nrip_save
```

2.1.1. 安装KVM

```
yum -y install qemu-kvm libvirt virt-install bridge-utils
```

```
yum -y install libguestfs-tools # 安装一些管理工具包
```

```
lsmod | grep kvm # 确认模块已加载
```

```
kvm_intel      138567  0
kvm            441119  1 kvm_intel
```

启动并设置开机启动：

```
systemctl start libvirtd
systemctl enable libvirtd
```

为**KVM**虚拟机配置桥接网络：

这里以“eth0”为例，实际操作中替换为你自己环境的接口名称（IP和网关等也是）。

2.1. KVM

```
nmcli c add type bridge autoconnect yes con-name br0 ifname br0 #  
添加桥接“br0”
```

```
nmcli c modify br0 ipv4.addresses 10.0.0.30/24 ipv4.method manual  
# 给br0设置IP
```

```
nmcli c modify br0 ipv4.gateway 10.0.0.1 # 给br0设置网关
```

```
nmcli c modify br0 ipv4.dns 10.0.0.1 # 给br0设置DNS
```

```
nmcli c delete eth0 # 删除eth0的当前设置（如果是远程操作会断开，最好是  
本机操作或双网卡）
```

```
nmcli c add type bridge-slave autoconnect yes con-name eth0 ifname  
eth0 master br0 # 添加eth0接口作为br0的成员
```

设置完成后重启电脑。

```
ip addr # 查看修改后信息
```

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    group default
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP>
    mtu 1500 qdisc pfifo_fast master br0 state UP group default
    qlen 1000
        link/ether 00:0c:29:9f:9b:d3 brd ff:ff:ff:ff:ff:ff
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc no queue state DOWN group default
        link/ether 22:f8:64:25:97:44 brd ff:ff:ff:ff:ff:ff
        inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
            valid_lft forever preferred_lft forever
4: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
        link/ether 00:0c:29:9f:9b:d3 brd ff:ff:ff:ff:ff:ff
        inet 10.0.0.30/24 brd 10.0.0.255 scope global br0
            valid_lft forever preferred_lft forever
        inet6 fe80::20c:29ff:fe9f:9bd3/64 scope link
            valid_lft forever preferred_lft forever

```

2.1.2. 创建一个虚拟机

以安装CentOS7为例

通过网络以文本模式安装客户机，可以在控制台或Putty等远程连接上使用。此外，虚拟机的映像默认放置在 /var/lib/libvirt/images 作为存储池，但本示例演示创建和使用新的存储池。

```

virt-install \
--name centos7 \
--ram 4096 \
--disk path=/var/kvm/images/centos7.img,size=30 \
--vcpus 2 \
--os-type linux \

```

```
--os-variant centos7.0 \
--network bridge=br0,model=virtio \
--graphics none \
--console pty,target_type=serial \
--location 'http://mirrors.163.com/centos/7/os/x86_64/' \
--extra-args 'console=ttyS0,115200n8 serial'

#Centos6的资料找的其他示例
#raw格式磁盘
virt-install \
--name=centos7-1 \
--ram 4096 \
--disk path=/var/kvm/images/centos7-1.img,size=30,bus=virtio \
--vcpus 2 \
--os-type linux \
--os-variant centos7.0 \
--network bridge=br0,model=virtio \
--accelerate \
--cdrom /data/iso/centos7.iso \
--vnc \
--vncport=5910 \
--vnclisten=0.0.0.0 \
--noautoconsole

#qcow2格式(空间动态增长)
qemu-img create -f qcow2 centos7-2.qcow2 30G
virt-install \
--virt-type kvm # 如果提示不支持kvm，检查BIOS是否开启虚拟化
--name=centos7-2 \
--ram 4096 \
--disk path=/var/kvm/images/centos7-2.qcow2,format=qcow2,size=30 \
,bus=virtio \
--vcpus 2 \
--os-type linux \
--os-variant centos7.0 \
--network bridge=br0,model=virtio \
--accelerate \
--cdrom /data/iso/centos7.iso \
--vnc \
--vncport=5910 \
--vnclisten=0.0.0.0 \
--noautoconsole
```

2.1. KVM

上面选项的意思如下（更多的选项可使用 `man virt-install` 命令查看）。

`--name` 指定虚拟机的名称

`--ram` 指定虚拟机的内存大小

`--disk path=xxx ,size=xxx` “`path`=”指定虚拟机磁盘的位置，“`size`=”指定虚拟机的磁盘空间

`--vcpus` 指定虚拟CPU

`--os-type` 指定客户机的类型

`--os-variant` 指定客户机的种类（可以使用命令 `osinfo-query os` 查看列表）

`--network` 指定虚拟机的网络类型

`--graphics` 指定图像的种类。如果设置为“`none`”，则意味着没有图像。

`--console` 指定控制台类型

`--location` 指定安装源位置

`--extra-args` 指定在内核中设置的参数

创建模板：

在文本模式下安装，与普通的安装过程相同。安装完成后，首先重新启动，然后登录。

使用快捷键“`Ctrl + J`”从客户机转到主机

使用命令 `virsh console` 虚拟机名称 从主机转到客户机，如：

`virsh console centos7`

因为从网络安装客户机后，它是最小化安装，因此可以将其保存为模板以便以后创建新虚拟机。

```
virt-clone --original centos7 --name template --file
/var/kvm/images/template.img # 创建一个模板
```

2.1. KVM

```
Allocating 'template.img' | 20 GB 01:44  
Clone 'template' created successfully.
```

```
ll /var/kvm/images/template.img # 磁盘映像
```

```
-rwxr-xr-x 1 root root 32212254720 Jul 11 23:34 /var/kvm/images/  
template.img
```

```
ll /etc/libvirt/qemu/template.xml # xml文件
```

```
-rw----- 1 root root 1843 Jul 11 23:32 /etc/libvirt/qemu/templa  
te.xml
```

在使用之前，对客户机设置基本初始配置。

定义新的存储池：

```
mkdir /etc/libvirt/storage
```

编辑 /etc/libvirt/storage/disk01.xml 文件：

2.1. KVM

```
# 新建
<pool type='dir'>
    # 设置一个名称
    <name>disk01</name>
    <capacity>0</capacity>
    <allocation>0</allocation>
    <available>0</available>
    <source>
    </source>
    <target>
        # 指定池目录
        <path>/var/kvm/images</path>
        <permissions>
            <mode>0700</mode>
            <owner>-1</owner>
            <group>-1</group>
        </permissions>
    </target>
</pool>
```

```
virsh pool-define /etc/libvirt/storage/disk01.xml # 定义池
```

```
Pool disk01 defined from /etc/libvirt/storage/disk01.xml
```

```
virsh pool-start disk01 # 启动池
```

```
Pool disk01 started
```

```
virsh pool-autostart disk01 # 设置自动启动
```

```
Pool disk01 marked as autostarted
```

```
virsh pool-list # 显示池列表
```

Name	State	Autostart

disk01	active	yes

```
virsh pool-info disk01 # 显示详细信息
```

```
Name:          disk01
UUID:          2de62477-7132-4512-b5d8-003e28da105c
State:         running
Persistent:    yes
Autostart:     yes
Capacity:      197.17 GiB
Allocation:    2.90 GiB
Available:     194.27 GiB
```

调整磁盘大小

使用命令 `qemu-img resize 需调整的镜像名 +30G` , 好像只能增加

让虚拟机显示主机的**CPU**信息

运行 `virsh edit 虚拟机名称` , 在内容中找到“cpu mode”改为“cpu mode='host-passthrough” (修改后如“”)

2.1.3. 图形界面管理

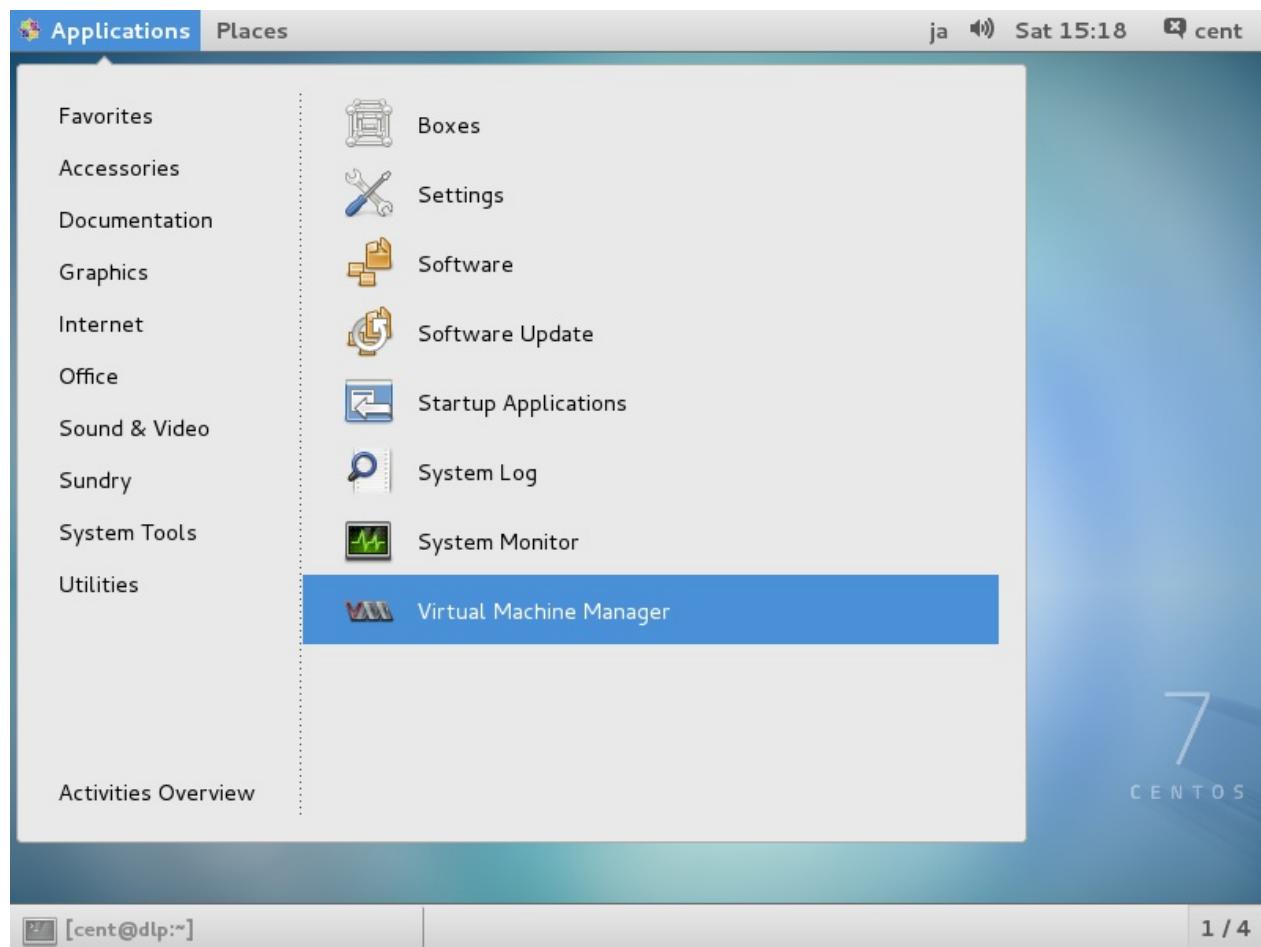
如果安装了[桌面环境](#)，则可以在图形界面上创建虚拟机。此示例显示在图形界面上安装Windows Server 2012 R2。

先安装管理工具：

```
yum -y install virt-manager
```

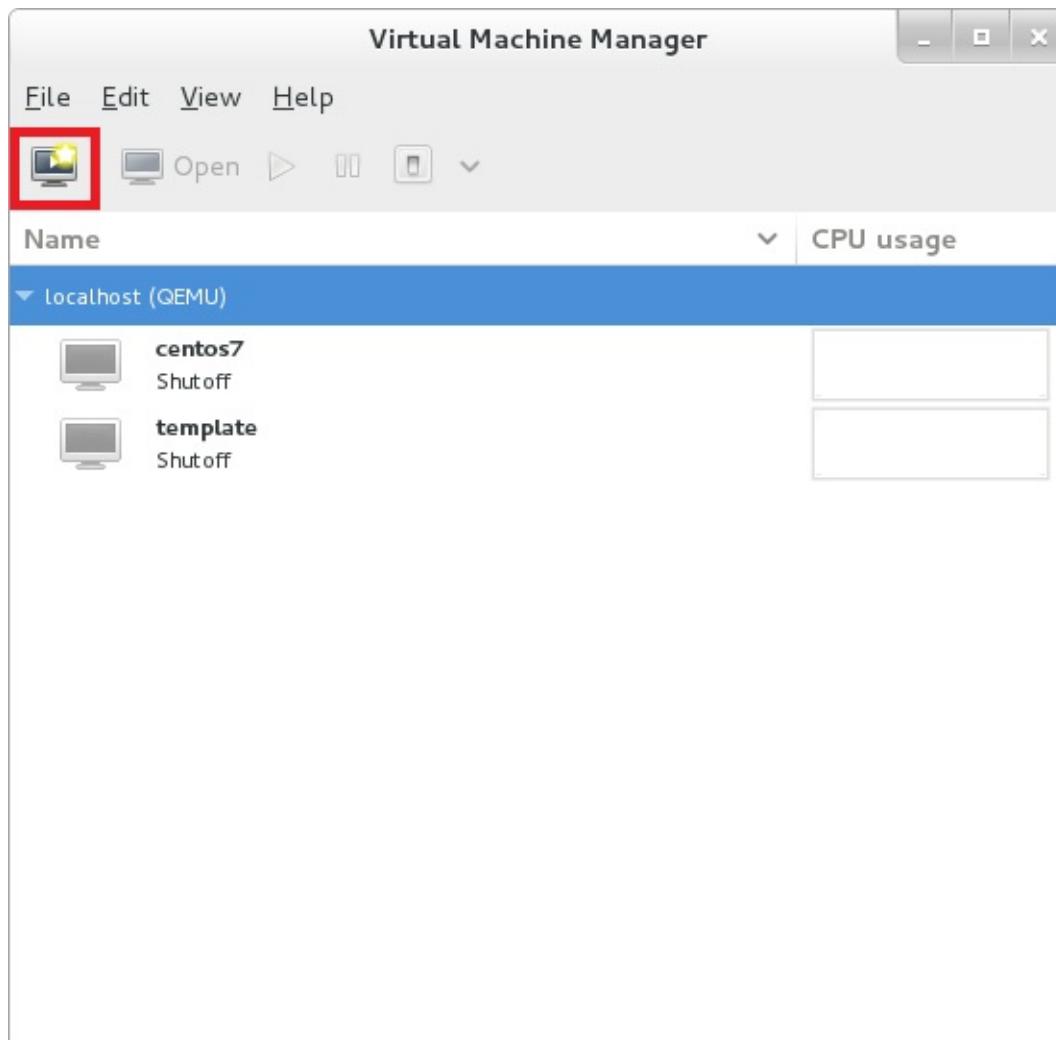
启动桌面并运行“Virtual Machine Manager”：

2.1. KVM



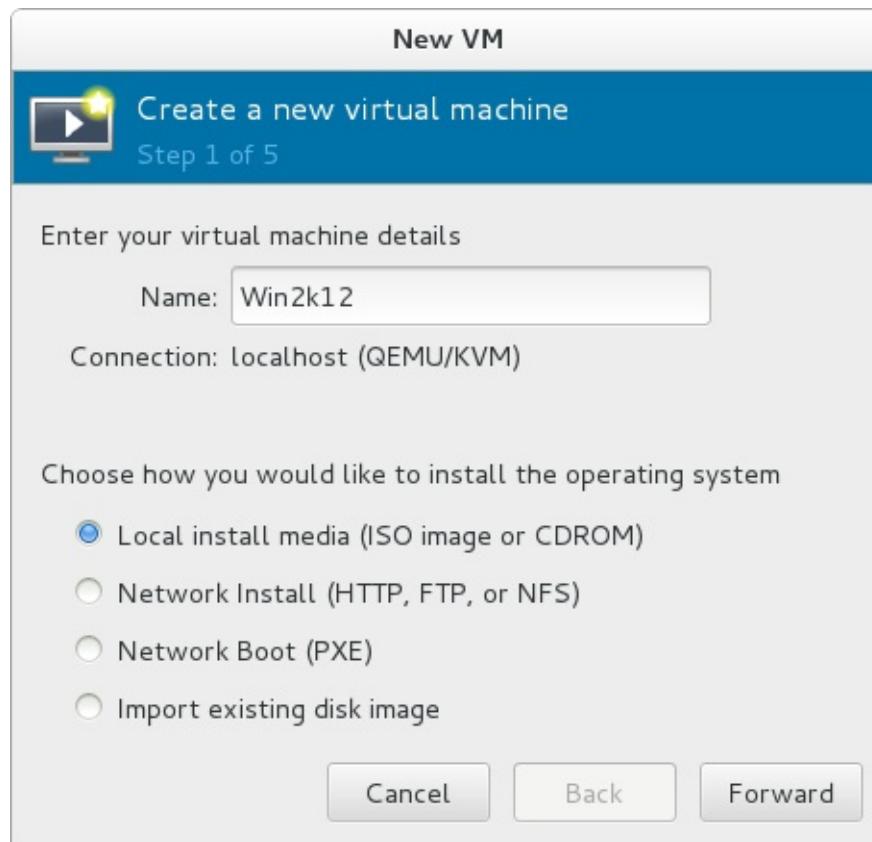
单击“New”按钮（左上角的电脑图标），然后打开向导以创建新的虚拟机：

2.1. KVM

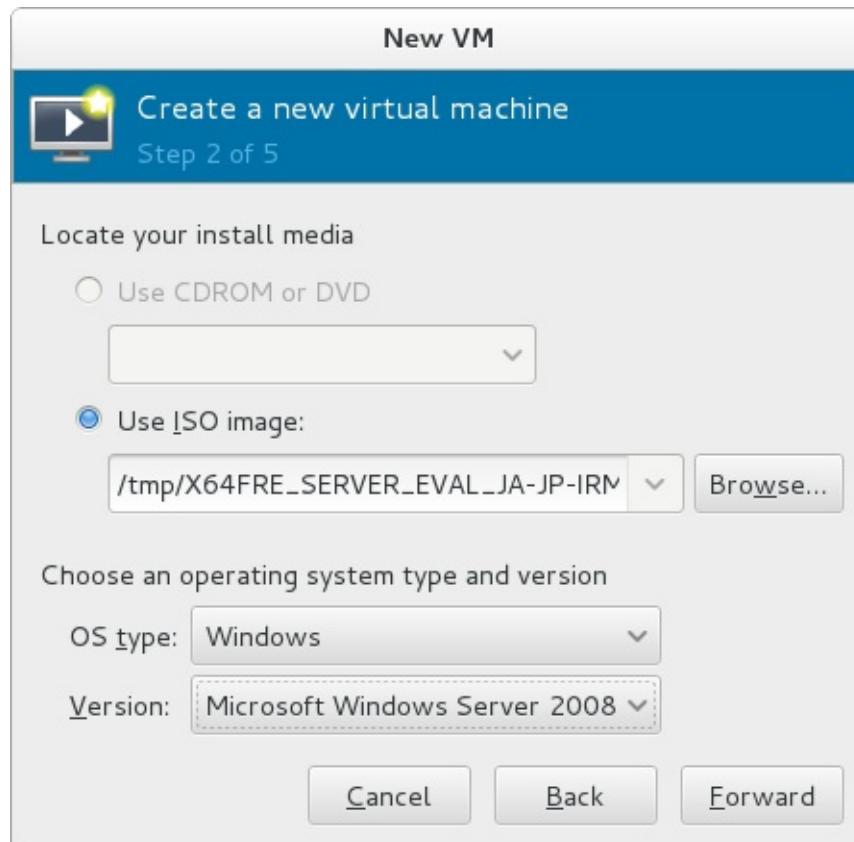


指定虚拟机和安装源的名称。此示例选择本地媒体：

2.1. KVM

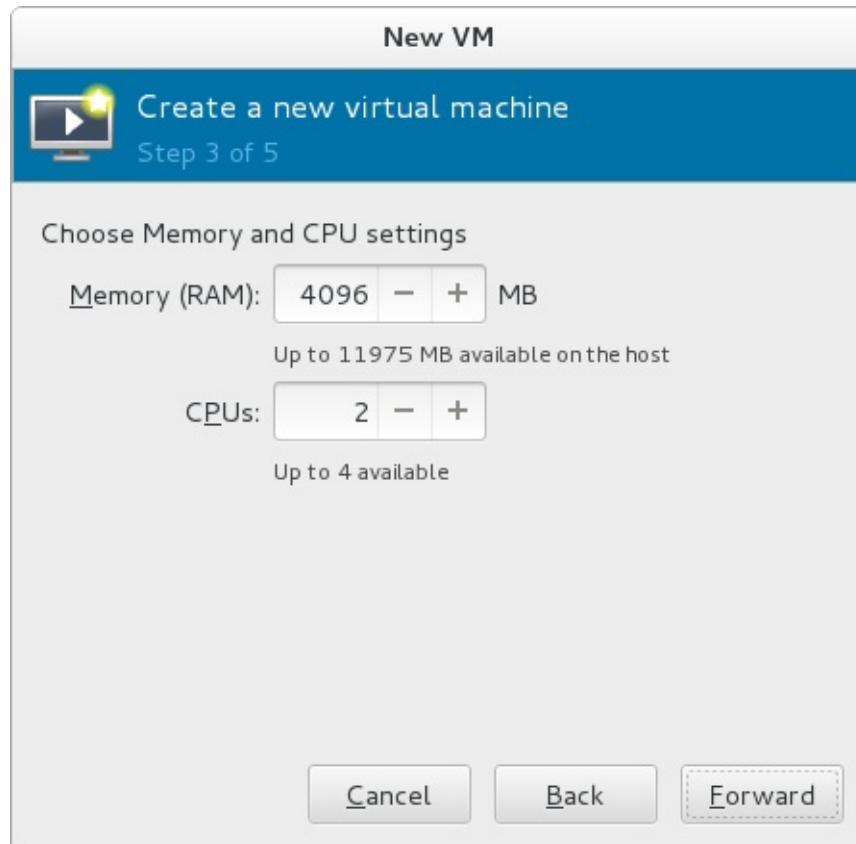


选择安装介质或ISO映像，并指定操作系统类型和版本。Windows Server 2012没有被列出，可以选择Windows 2008来安装：

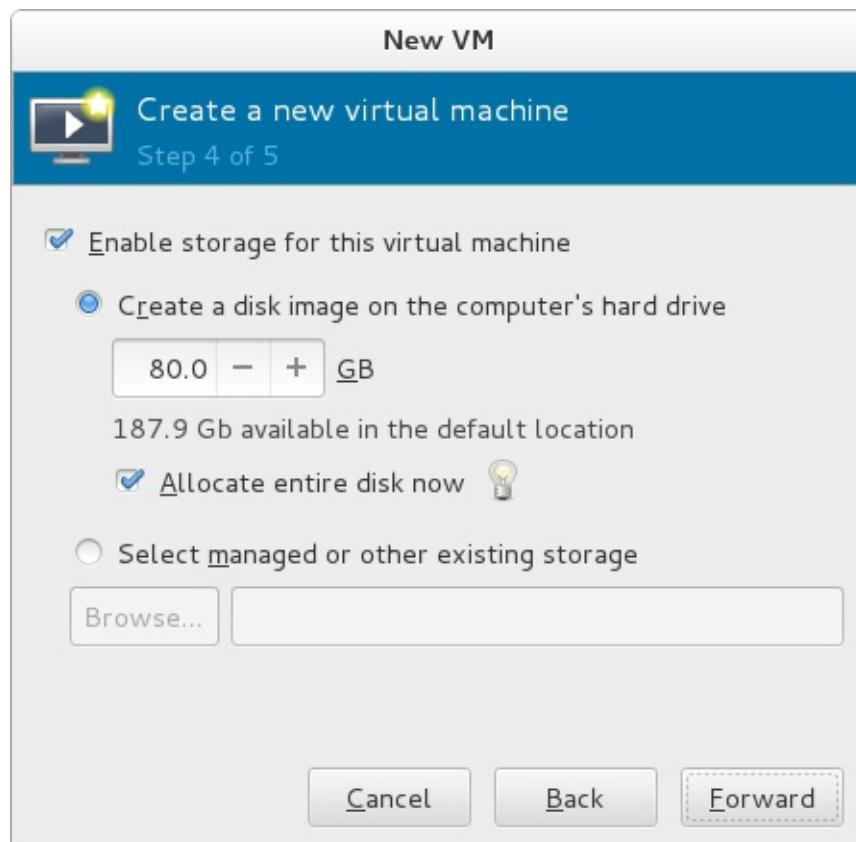


指定内存大小和虚拟CPU数量：

2.1. KVM

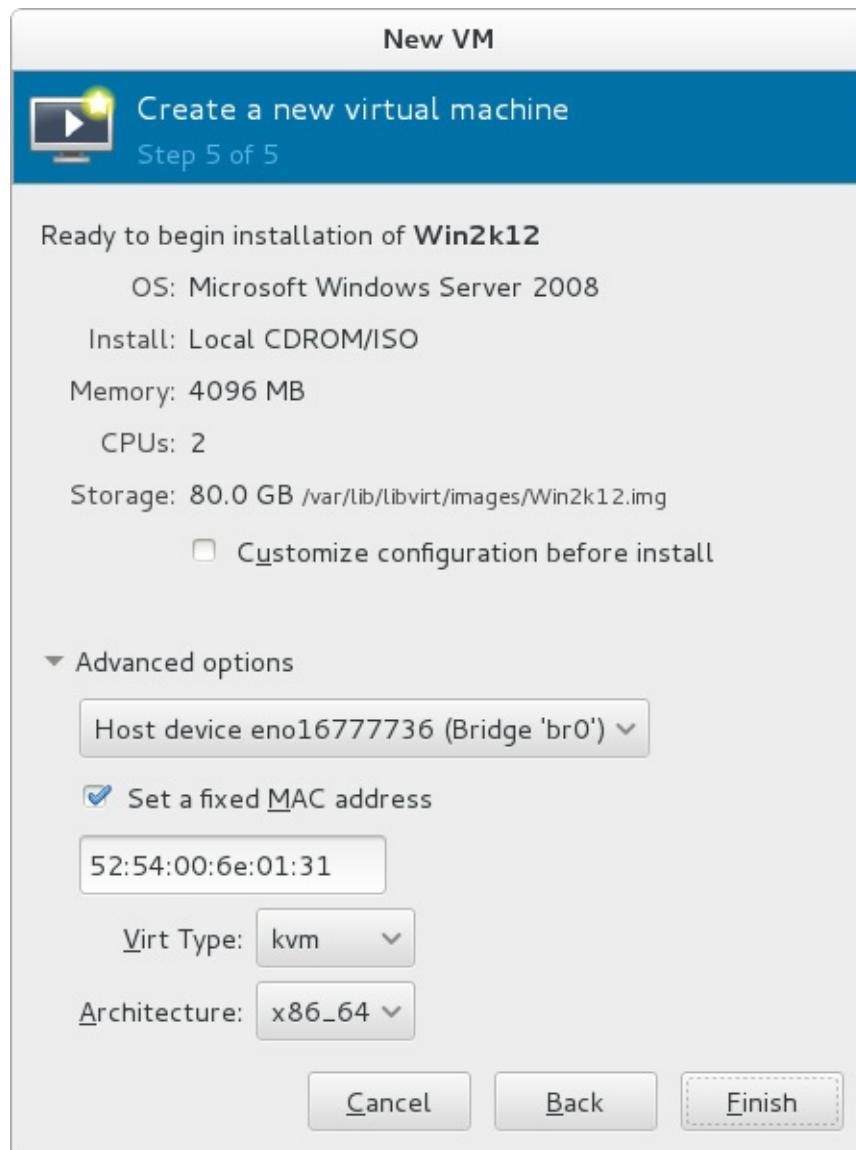


指定磁盘大小：



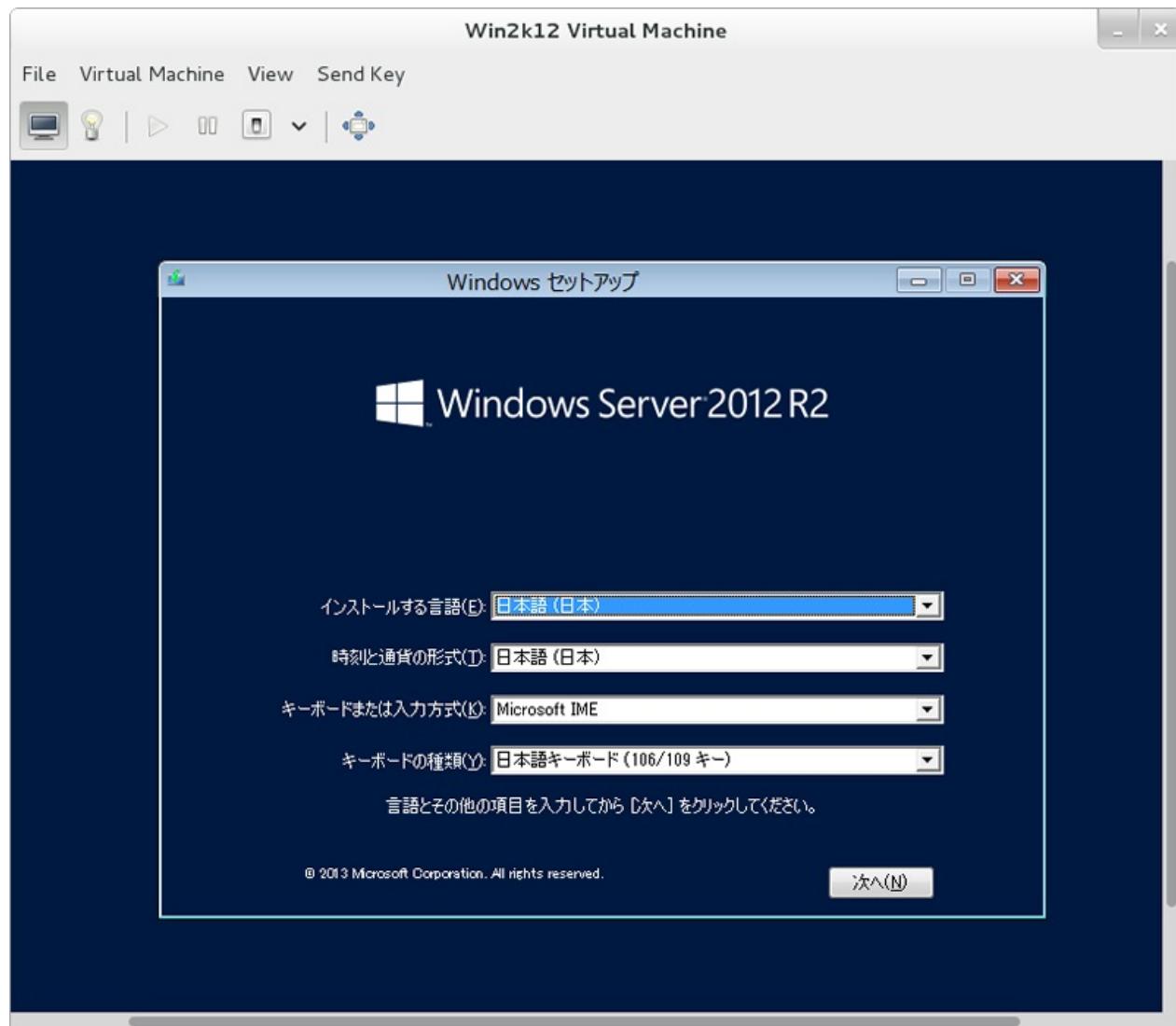
打开“Advanced options”，确保网络等的设置正确：

2.1. KVM



Windows Server 2012 R2安装开始：

2.1. KVM



另，Windows Virtio Drivers下载地址：

旧版本：<https://fedorapeople.org/groups/virt/virtio-win/deprecated-isos/latest/>

最新版本：https://fedoraproject.org/wiki/Windows_Virtio_Drivers

2.1.4. 基本操作

virsh命令的基本操作

启动虚拟机：

```
virsh start centos7 # 启动虚拟机“centos7”
```

```
Domain centos7 started
```

```
virsh start centos7 --console # 启动并连接到“centos7”的控制台
```

2.1. KVM

```
Domain centos7 started  
Connected to domain centos7
```

停止虚拟机：

```
virsh shutdown centos7 # 停止虚拟机“centos7”
```

```
Domain centos7 is being shutdown
```

```
virsh destroy centos7 # 强制停止虚拟机“centos7”
```

```
Domain centos7 destroyed
```

虚拟机设置自动启动：

```
virsh autostart centos7 # 启用“centos7”的自动启动
```

```
Domain centos7 marked as autostarted
```

```
virsh autostart --disable centos7 # 禁用“centos7”的自动启动
```

```
Domain centos7 unmarked as autostarted
```

列出所有虚拟机：

```
virsh list # 列出所有活动的虚拟机
```

Id	Name	State
2	centos7	running

```
virsh list --all # 列出所有虚拟机（包括非活动）
```

2.1. KVM

Id	Name	State

-	centos7	shut off
-	template	shut off
-	Win2k12	shut off

切换控制台：

使用快捷键“Ctrl +]”从客户机转到主机

使用命令 `virsh console` 虚拟机名称 从主机转到客户机，如：

```
virsh console centos7
```

```
Connected to domain centos7
Escape character is ^] # 回车

CentOS Linux 7 (Core)
Kernel 3.10.0-123.el7.x86_64 on an x86_64

localhost login: # 已切换到客户机
Password: # 输入客户机密码
```

其他选项：

老版本的资料，可能会有部分失效

```
virsh suspend centos7 # 暂停虚拟机
virsh resume centos7 # 恢复虚拟机
```

autostart	#自动加载指定的一个虚拟机
connect	#重新连接到hypervisor
console	#连接到客户会话
create	#从一个SML文件创建一个虚拟机
start	#开始一个非活跃的虚拟机
destroy	#删除一个虚拟机
define	#从一个XML文件定义一个虚拟机
domid	#把一个虚拟机名或UUID转换为ID
domuuid	#把一个郁闷或ID转换为UUID
dominfo	#查看虚拟机信息
domstate	#查看虚拟机状态
domblkstat	#获取虚拟机设备快状态
domifstat	#获取虚拟机网络接口状态
dumpxml	#XML中的虚拟机信息
edit	#编辑某个虚拟机的XML文件
list	#列出虚拟机
migrate	#将虚拟机迁移到另一台主机
quit	#退出非交互式终端
reboot	#重新启动一个虚拟机
resume	#重新恢复一个虚拟机
save	#把一个虚拟机的状态保存到一个文件
dump	#把一个虚拟机的内核dump到一个文件中以方便分析
shutdown	#关闭一个虚拟机
setmem	#改变内存的分配
setmaxmem	#改变最大内存限制值
suspend	#挂起一个虚拟机
vcpuinfo	#虚拟机的cpu信息
version	#显示virsh版本

```
virsh --help # 通过此命令查看更多
```

virsh [options]... [<command_string>]
virsh [options]... <command> [args...]
options:
-c --connect=URI hypervisor connection URI

```

-r | --readonly          connect readonly
-d | --debug=NUM        debug level [0-4]
-h | --help              this help
-q | --quiet             quiet mode
-t | --timing            print timing information
-l | --log=FILE          output logging to file
-v                      short version
-V                      long version
--version[=TYPE]         version, TYPE is short or long (defa
ult short)
-e | --escape <char>     set escape sequence for console

```

commands (non interactive mode):

Domain Management (help keyword 'domain')	
attach-device	从一个XML文件附加装置
attach-disk	附加磁盘设备
attach-interface	获得网络界面
autostart	自动开始一个域
blkdeviotune	Set or query a block device I /O tuning parameters.
blkiotune	Get or set blkio parameters
blockcommit	Start a block commit operatio n.
blockcopy	Start a block copy operation.
blockjob	Manage active block operation
blockpull	Populate a disk from its back ing image.
blockresize	Resize block device of domain
change-media drive	Change media of CD or floppy drive
console	连接到客户会话
cpu-baseline	compute baseline CPU
cpu-compare	compare host CPU with a CPU d escribed by an XML file
cpu-stats	show domain cpu statistics
create	从一个 XML 文件创建一个域
define	从一个 XML 文件定义 (但不开始) 一 个域
desc	show or set domain's descript

ion or title	
destroy	destroy (stop) a domain
detach-device	从一个 XML 文件分离设备
detach-disk	分离磁盘设备
detach-interface	分离网络界面
domdisplay	domain display connection URI
domhostname	print the domain's hostname
domid	把一个域名或 UUID 转换为域 id
domif-setlink	set link state of a virtual i
nterface	
domiftune	get/set parameters of a virtu
al interface	
domjobabort	abort active domain job
domjobinfo	domain job information
domname	将域 id 或 UUID 转换为域名
dompmsuspend	suspend a domain gracefully u
sing power management functions	
dompmwakeup	wakeup a domain from pmsuspen
ded state	
domuuid	把一个域名或 id 转换为域 UUID
domxml-from-native	Convert native config to doma
in XML	
domxml-to-native	Convert domain XML to native
config	
dump	把一个域的内核 dump 到一个文件中以
方便分析	
dumpxml	XML 中的域信息
edit	编辑某个域的 XML 配置
inject-nmi	Inject NMI to the guest
send-key	Send keycodes to the guest
managedsave	managed save of a domain stat
e	
managedsave-remove	Remove managed save of a doma
in	
maxvcpus	connection vcpu maximum
memtune	Get or set memory parameters
migrate	将域迁移到另一个主机中
migrate-setmaxdowntime	set maximum tolerable downtim
e	
migrate-setspeed	Set the maximum migration ban
dwidth	
migrate-getspeed	Get the maximum migration ban

```

dwidth
    numatune          Get or set numa parameters
    reboot            重新启动一个域
    reset             reset a domain
    restore           从一个存在一个文件中的状态恢复一个
域
    resume            重新恢复一个域
    save              把一个域的状态保存到一个文件
    save-image-define redefine the XML for a domain
's saved state file
    save-image-dumpxml saved state domain informatio
n in XML
    save-image-edit  edit XML for a domain's saved
state file
    schedinfo         显示/设置日程安排变量
    screenshot        take a screenshot of a curren
t domain console and store it into a file
    setmaxmem         改变最大内存限制值
    setmem            改变内存的分配
    setvcpus          改变虚拟 CPU 的号
    shutdown          关闭一个域
    start             开始一个（以前定义的）非活跃的域
    suspend           挂起一个域
    ttyconsole        tty 控制台
    undefine          undefine a domain
    update-device     update device from an XML fil
e
    vcpucount         domain vcpu counts
    vcpuinfo          detailed domain vcpu informati
on
    vcpupin           control or query domain vcpu
affinity
    emulatorpin      control or query domain emula
tor affinity
    vncdisplay        vnc 显示

Domain Monitoring (help keyword 'monitor')
    domblkerror       Show errors on block devices
    domblkinfo        domain block device size info
rmation
    domblklist        list all domain blocks
    domblkstat        获得域设备块状态

```

domcontrol	domain control interface stat
e	
domif-getlink	get link state of a virtual i
nterface	
domiflist	list all domain virtual inter
faces	
domifstat	获得域网络接口状态
dominfo	域信息
dommemstat	get memory statistics for a d
omain	
domstate	域状态
list	列出域
 Host and Hypervisor (help keyword 'host')	
capabilities	性能
freecell	NUMA可用内存
hostname	打印管理程序主机名
node-memory-tune	Get or set node memory parame
ters	
nodecpustats	Prints cpu stats of the node.
nodeinfo	节点信息
nodememstats	Prints memory stats of the no
de.	
nodesuspend	suspend the host node for a g
iven time duration	
qemu-attach	QEMU Attach
qemu-monitor-command	QEMU Monitor Command
qemu-agent-command	QEMU Guest Agent Command
sysinfo	print the hypervisor sysinfo
uri	打印管理程序典型的URI
version	显示版本
 Interface (help keyword 'interface')	
iface-begin	create a snapshot of current
interfaces settings, which can be later committed (iface-commit)	
or restored (iface-rollback)	
iface-bridge	create a bridge device and at
tach an existing network device to it	
iface-commit	commit changes made since ifa
ce-begin and free restore point	
iface-define	define (but don't start) a ph
ysical host interface from an XML file	

iface-destroy	destroy a physical host interface
face (disable it / "if-down")	
iface-dumpxml	interface information in XML
iface-edit	edit XML configuration for a
physical host interface	
iface-list	list physical host interfaces
iface-mac	convert an interface name to
interface MAC address	
iface-name	convert an interface MAC address to interface name
iface-rollback	rollback to previous saved configuration created via iface-begin
iface-start	start a physical host interface
ce (enable it / "if-up")	
iface-unbridge	undefine a bridge device after detaching its slave device
iface-undefine	undefine a physical host interface (remove it from configuration)
 Network Filter (help keyword 'filter')	
nwfilter-define	define or update a network filter from an XML file
nwfilter-dumpxml	network filter information in XML
nwfilter-edit	edit XML configuration for a network filter
nwfilter-list	list network filters
nwfilter-undefine	undefine a network filter
 Networking (help keyword 'network')	
net-autostart	自动开始网络
net-create	从一个 XML 文件创建一个网络
net-define	从一个 XML 文件定义(但不开始)一个网络
net-destroy	destroy (stop) a network
net-dumpxml	XML 中的网络信息
net-edit	为网络编辑 XML 配置
net-info	network information
net-list	列出网络
net-name	把一个网络UUID 转换为网络名
net-start	开始一个(以前定义的)不活跃的网络
net-undefine	取消定义一个非活跃的网络

net-update	update parts of an existing network's configuration
net-uuid	把一个网络名转换为网络UUID
Node Device (help keyword 'nodedev')	
nodedev-create	create a device defined by an XML file on the node
nodedev-destroy	destroy (stop) a device on the node
nodedev-detach	detach node device from its device driver
nodedev-dumpxml	XML 中的节点设备详情
nodedev-list	这台主机中中的枚举设备
nodedev-reattach	reattach node device to its device driver
nodedev-reset	重置节点设备
Secret (help keyword 'secret')	
secret-define	define or modify a secret from an XML file
secret-dumpxml	secret attributes in XML
secret-get-value	Output a secret value
secret-list	list secrets
secret-set-value	set a secret value
secret-undefine	undefine a secret
Snapshot (help keyword 'snapshot')	
snapshot-create	Create a snapshot from XML
snapshot-create-as	Create a snapshot from a set of args
snapshot-current	Get or set the current snapshot
snapshot-delete	Delete a domain snapshot
snapshot-dumpxml	Dump XML for a domain snapshot
snapshot-edit	edit XML for a snapshot
snapshot-info	snapshot information
snapshot-list	List snapshots for a domain
snapshot-parent	Get the name of the parent of a snapshot
snapshot-revert	Revert a domain to a snapshot

Storage Pool (help keyword 'pool')	
find-storage-pool-sources-as	找到潜在存储池源
find-storage-pool-sources	发现潜在存储池源
pool-autostart	自动启动某个池
pool-build	建立池
pool-create-as	从一组变量中创建一个池
pool-create	从一个 XML 文件中创建一个池
pool-define-as	在一组变量中定义池
pool-define	在一个 XML 文件中定义 (但不启动)
一个池	
pool-delete	删除池
pool-destroy	destroy (stop) a pool
pool-dumpxml	XML 中的池信息
pool-edit	为存储池编辑 XML 配置
pool-info	存储池信息
pool-list	列出池
pool-name	将池 UUID 转换为池名称
pool-refresh	刷新池
pool-start	启动一个 (以前定义的) 非活跃的池
pool-undefine	取消定义一个不活跃的池
pool-uuid	把一个池名称转换为池 UUID
Storage Volume (help keyword 'volume')	
vol-clone	clone a volume.
vol-create-as	从一组变量中创建卷
vol-create	从一个 XML 文件创建一个卷
vol-create-from olume as input	create a vol, using another v
vol-delete	删除卷
vol-download file	download volume contents to a
vol-dumpxml	XML 中的卷信息
vol-info	存储卷信息
vol-key given volume name or path	returns the volume key for a
vol-list	列出卷
vol-name given volume key or path	returns the volume name for a
vol-path given volume name or key	returns the volume path for a
vol-pool a given volume key or path	returns the storage pool for

```

vol-resize           resize a vol
vol-upload          upload file contents to a vol
volume
vol-wipe            wipe a vol

Virsh itself (help keyword 'virsh')
cd                  change the current directory
connect            连接（重新连接）到 hypervisor
echo                echo arguments
exit                退出这个非交互式终端
help               打印帮助
pwd                print the current directory
quit               退出这个非交互式终端

(specify help <group> for details about the commands in the group)

(使用 --help <command> 来获得这个命令的详细信息)

```

2.1.5. 虚拟管理工具

为虚拟管理安装有用的工具。

```

yum -y install libguestfs-tools virt-top

virt-ls -l -d centos7 /root # “ls”虚拟机中的目录

```

```

total 36
dr-xr-x---. 2 root root 4096 Jan  8 22:38 .
drwxr-xr-x. 17 root root 4096 Jan  8 22:36 ..
-rw-----. 1 root root    61 Jan  8 22:38 .bash_history
-rw-r--r--. 1 root root   18 Dec 29  2013 .bash_logout
-rw-r--r--. 1 root root  176 Dec 29  2013 .bash_profile
-rw-r--r--. 1 root root  176 Dec 29  2013 .bashrc
...

```

```

virt-cat -d centos7 /etc/passwd # “cat”虚拟机中的文件

```

2.1. KVM

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
...
```

```
virt-edit -d centos7 /etc/fstab # 编辑虚拟机中的文件
```

```
#
# /etc/fstab
# Created by anaconda on Thu Jan  8 13:20:43 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for
# more info
#
/dev/mapper/centos-root /           xfs      defaults
                          1 1
UUID=537b215f-30a1-4e82-b05d-f480aa8e1034 /boot xfs      defaults
                          1 2
/dev/mapper/centos-swap swap       swap      defaults
                          0 0
```

```
virt-df -d centos7 # 显示虚拟机中的磁盘使用情况
```

Filesystem	1K-blocks	Used	Available
Use%			
centos7:/dev/sda1	508588	72348	436240
15%			
centos7:/dev/centos/root	8910848	779252	8131596
9%			

```
guestmount -d centos7 -i /media # 装载虚拟机的磁盘，ll /media 查看
```

```
total 32
lrwxrwxrwx.  1 root root    7 Jan  8 22:22 bin -> usr/bin
dr-xr-xr-x.  4 root root 4096 Jan  8 22:37 boot
drwxr-xr-x.  2 root root    6 Jan  8 22:20 dev
drwxr-xr-x. 74 root root 8192 Jan  8 22:36 etc
...
```

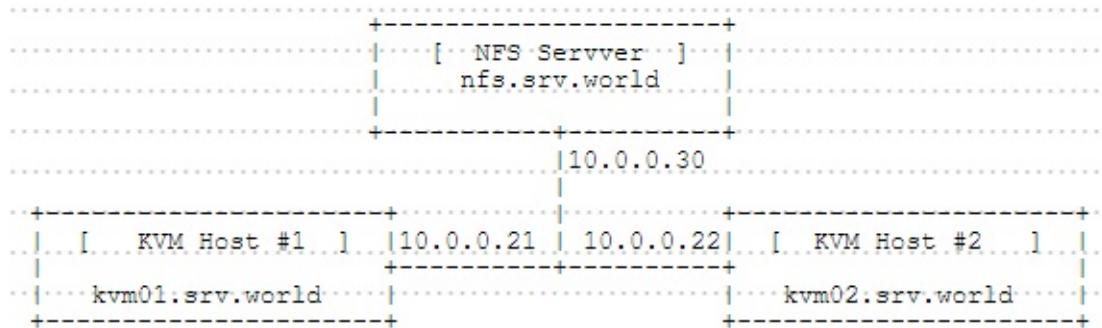
`virt-top # 显示虚拟机的状态`

```
virt-top 22:32:14 - x86_64 4/4CPU 2801MHz 11968MB
2 domains, 1 active, 1 running, 0 sleeping, 0 paused, 1 inactive
D:0 O:0 X:0
CPU: 0.2% Mem: 500 MB (500 MB by guests)

      ID S RDRQ WRRQ RXBY TXBY %CPU %MEM      TIME     NAME
      6 R   0   0           0.2   4.0   0:09.14 guestfs-07nss1p3
kxvyl1r5
-
                                         (centos7)
```

2.1.6. 实时迁移

需要2个KVM主机服务器和一个存储服务器（先设置好DNS或hosts以正常解析域名或IP地址），如下所示：



配置存储虚拟机映像的存储服务器。存储服务器可以使
用NFS，iSCSI，GlusterFS。本示例使用NFS存储服务器。

配置两台KVM主机服务器，并将存储服务器提供的目录（本示例
为 /var/kvm/images）挂载到KVM服务器上的相同挂载点上。

在一台KVM主机服务器（本示例10.0.0.21）上创建并启动虚拟机。

2.1. KVM

在虚拟机运行的服务器上执行实时迁移。完成后，虚拟机将迁移到另一台KVM主机上，如下所示：

```
virsh list # 在10.0.0.21列出运行中的虚拟机（本示例为“centos7”）
```

Id	Name	State

3	centos7	running

```
virsh migrate --live centos7 qemu+ssh://10.0.0.22/system # 执行后提示输入10.0.0.22的root密码
```

```
virsh list # 再次列出，“centos7”不再显示（已迁移）
```

Id	Name	State

```
virsh list # 在10.0.0.22列出运行中的虚拟机
```

Id	Name	State

1	centos7	running

```
virsh migrate --live centos7 qemu+ssh://10.0.0.21/system # 迁移回到10.0.0.21
```

```
virsh list
```

Id	Name	State

备份还原

虚拟机配置文件是扩展名为 `xml` 的文件，默认存储在 `/etc/libvirt/qemu` 下。

虚拟机关闭后，运行 `virsh dumpxml centos7 > /指定路径/centos7.xml`

将`xml`和磁盘镜像文件拷到新宿主主机上，在新主机运行：

```
virsh define /指定路径/centos7.xml
```

注意：如果磁盘镜像存储的路径与原来不同，需要先编辑 centos7.xml 。

2.1.7. SPICE服务器

安装桌面虚拟化“SPICE (Simple Protocol for Independent Computing Environment)”。可以从远程客户端计算机连接到虚拟机。

```
yum -y install spice-server spice-protocol #通常会作为KVM的依赖已  
安装
```

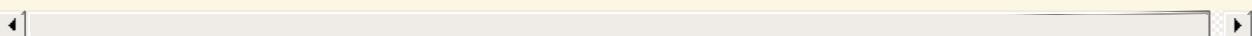
编辑现有虚拟机的xml文件并使用SPICE启动虚拟机，如下所示。如果是没有图形的虚拟机，可以更改为如下设置，但如果创建了带有图形的虚拟机，删除xml文件中的 `<graphics>***` 和 `<video>***` 部分，因为qxl用于图形。

```
virsh edit centos7 # 编辑“centos7”的配置
```

2.1. KVM

```
<domain type='kvm'>
  <name>centos7</name>
  <uuid>b38a50ca-a1ae-4d37-ba10-caf1e05b43ce</uuid>
  <memory unit='KiB'>4194304</memory>
  <currentMemory unit='KiB'>4194304</currentMemory>
  <vcpu placement='static'>2</vcpu>
  .
  .
  .

  # 加入以下内容
  # 在"passwd=***"部分设置密码
  # 给"sound"部分指定唯一的编号"slot='0x06'"
  # "video"部分中的"slot='0x02'"是固定的编号
  <graphics type='spice' port='5900' autoport='no' listen='0.0.0.0' passwd='password'>
    <listen type='address' address='0.0.0.0' />
  </graphics>
  <sound model='ac97'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />
  </sound>
  <video>
    <model type='qxl' ram='65536' vram='32768' heads='1' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0' />
  </video>
  # 以上为加入的内容
  <memballoon model='virtio'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0' />
  </memballoon>
  </devices>
</domain>
```



```
virsh start centos7 # 启动虚拟机
```

```
Domain centos7 started
```

2.1. KVM

如果想在初始创建虚拟机时启用SPICE，请如下指定。然后，可以使用SPICE安装需要图形界面的系统（如Windows），而不用在KVM主机上安装桌面环境。

```
virt-install \
--name Win2k12R2 \
--ram 6144 \
--disk path=/var/kvm/images/Win2k12R2.img,size=100 \
--vcpus=4 \
--os-type windows \
--os-variant=win2k12r2 \
--network bridge=br0 \
--graphics spice,listen=0.0.0.0,password=password,keymap=en \
--video qxl \
--cdrom /tmp/en_windows_server_2012_r2_vl_with_update_x64_dvd_40
65221.iso
```

firewalld防火墙设置（端口5900/TCP）：

```
firewall-cmd --add-port=5900/tcp --permanent
firewall-cmd --reload
```

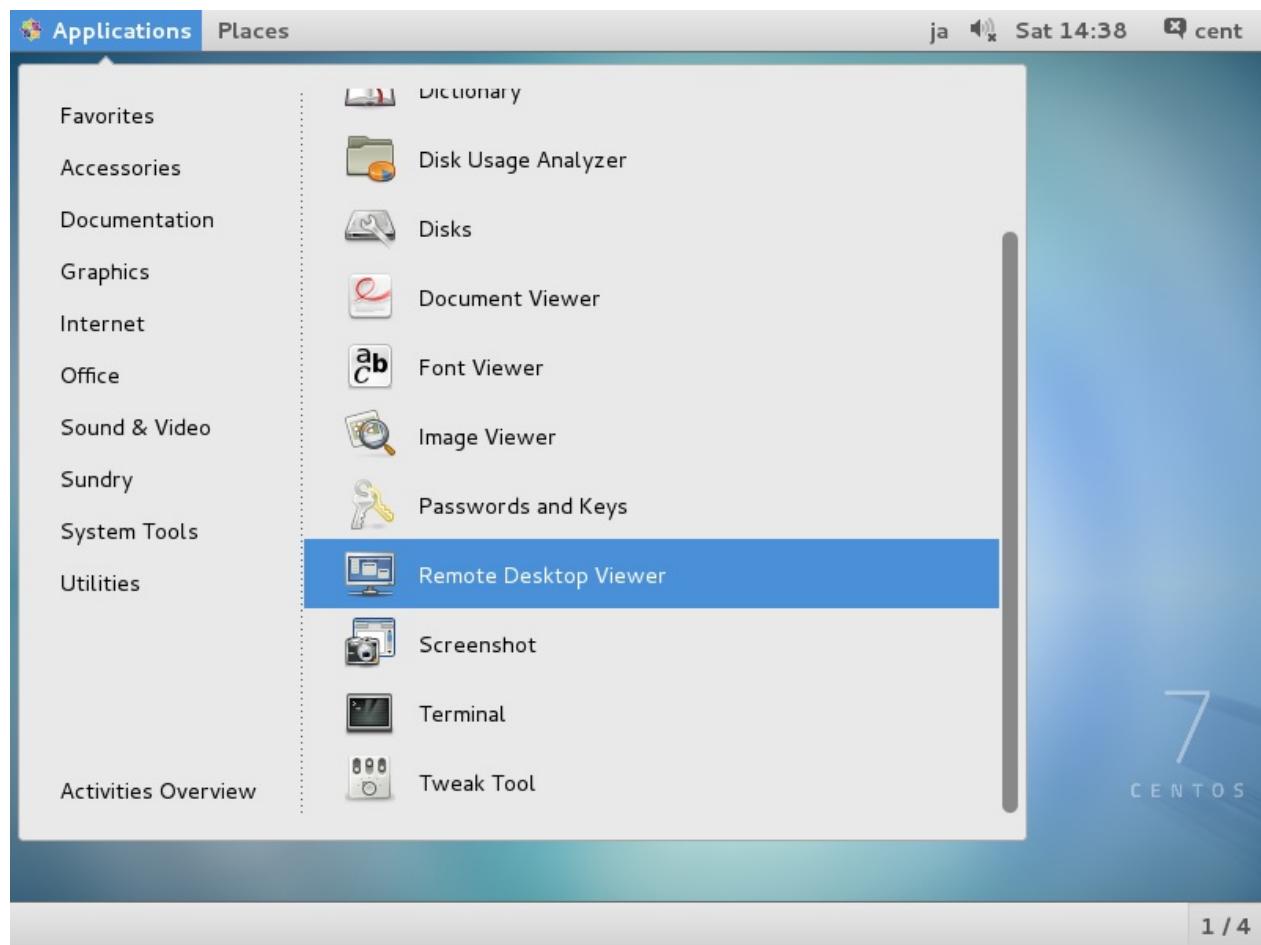
配置完成。参阅[从SPICE客户端连接到SPICE服务器](#)。

2.1.8. SPICE客户端

2.1.8.1. CentOS客户端

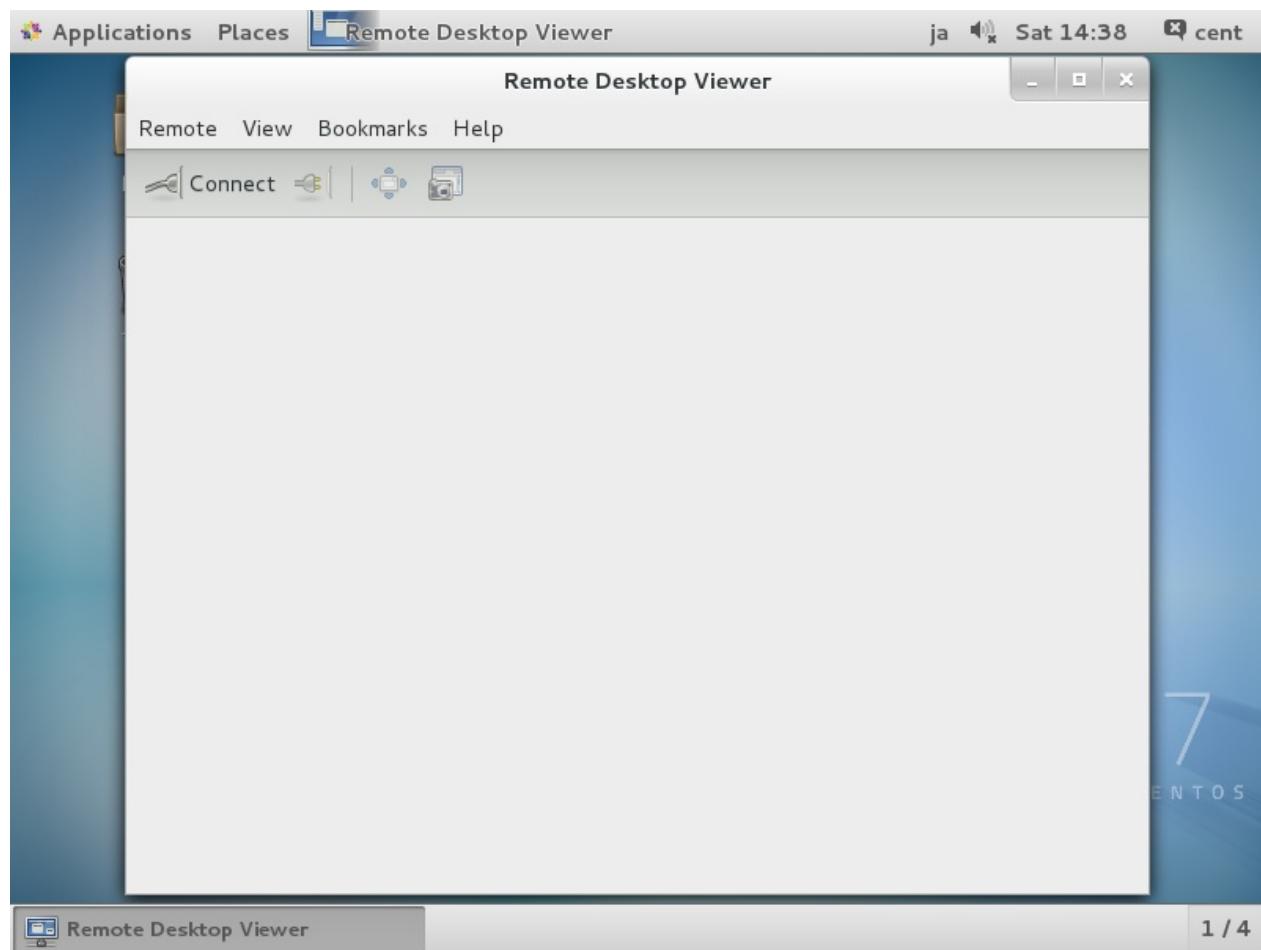
启动“Remote Desktop Viewer”：

2.1. KVM



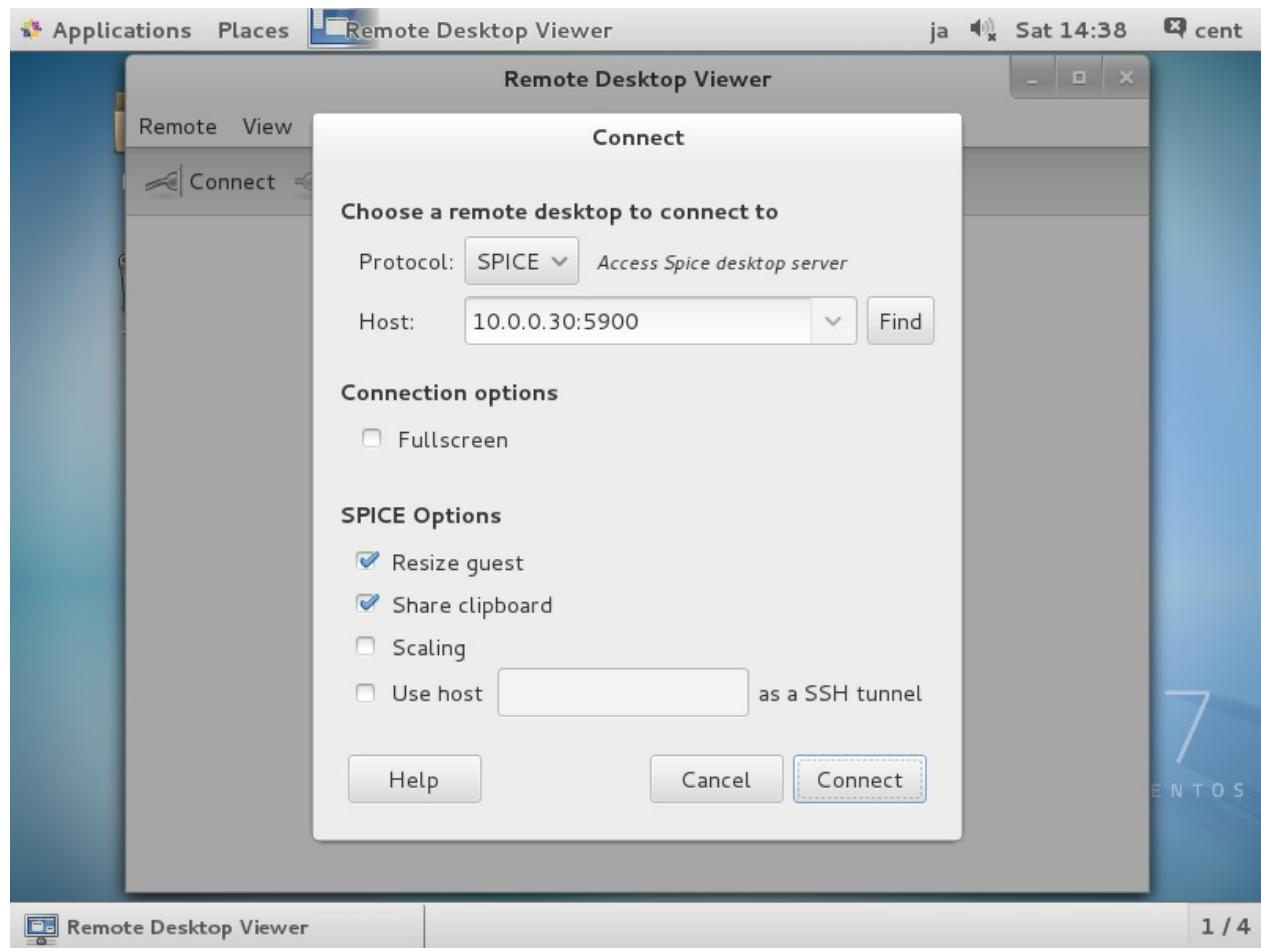
点击“Connect”按钮：

2.1. KVM



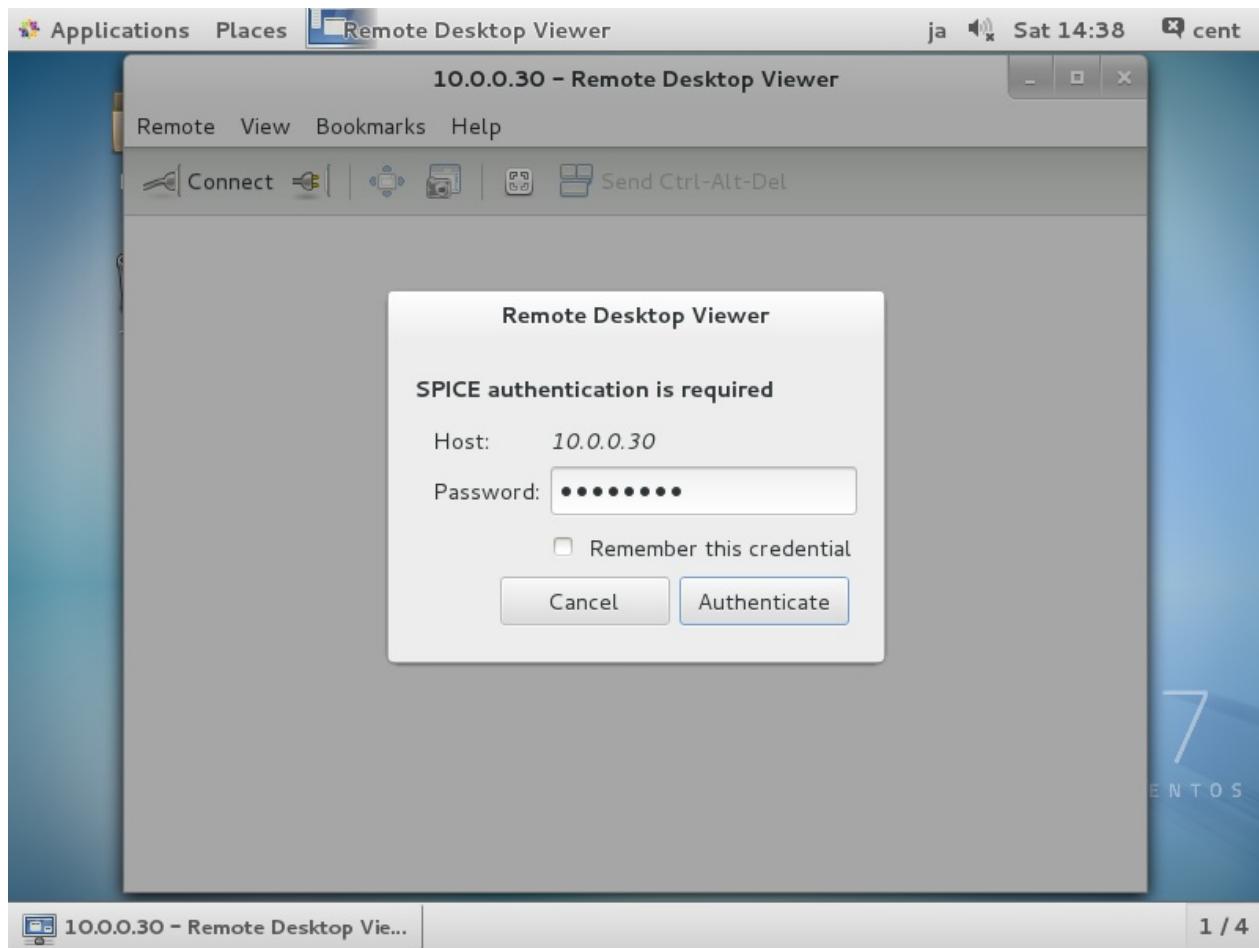
在“Protocol”字段选择“SPICE”，并在“Host”字段输入[服务器名称或IP地址:设置端口]，然后单击“Connect”按钮：

2.1. KVM

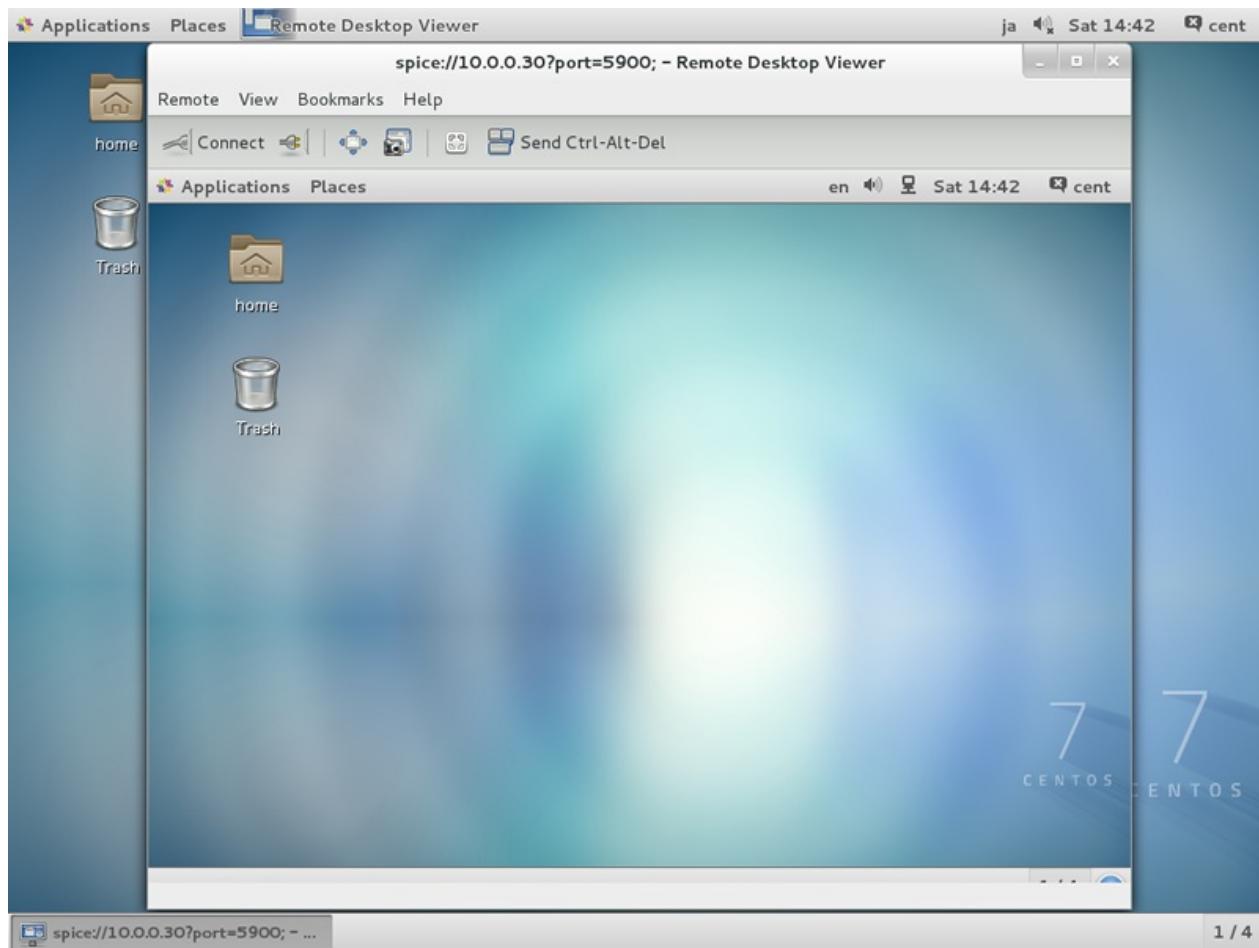


输入设置的密码，点击“Authenticate”按钮：

2.1. KVM



成功认证之后：

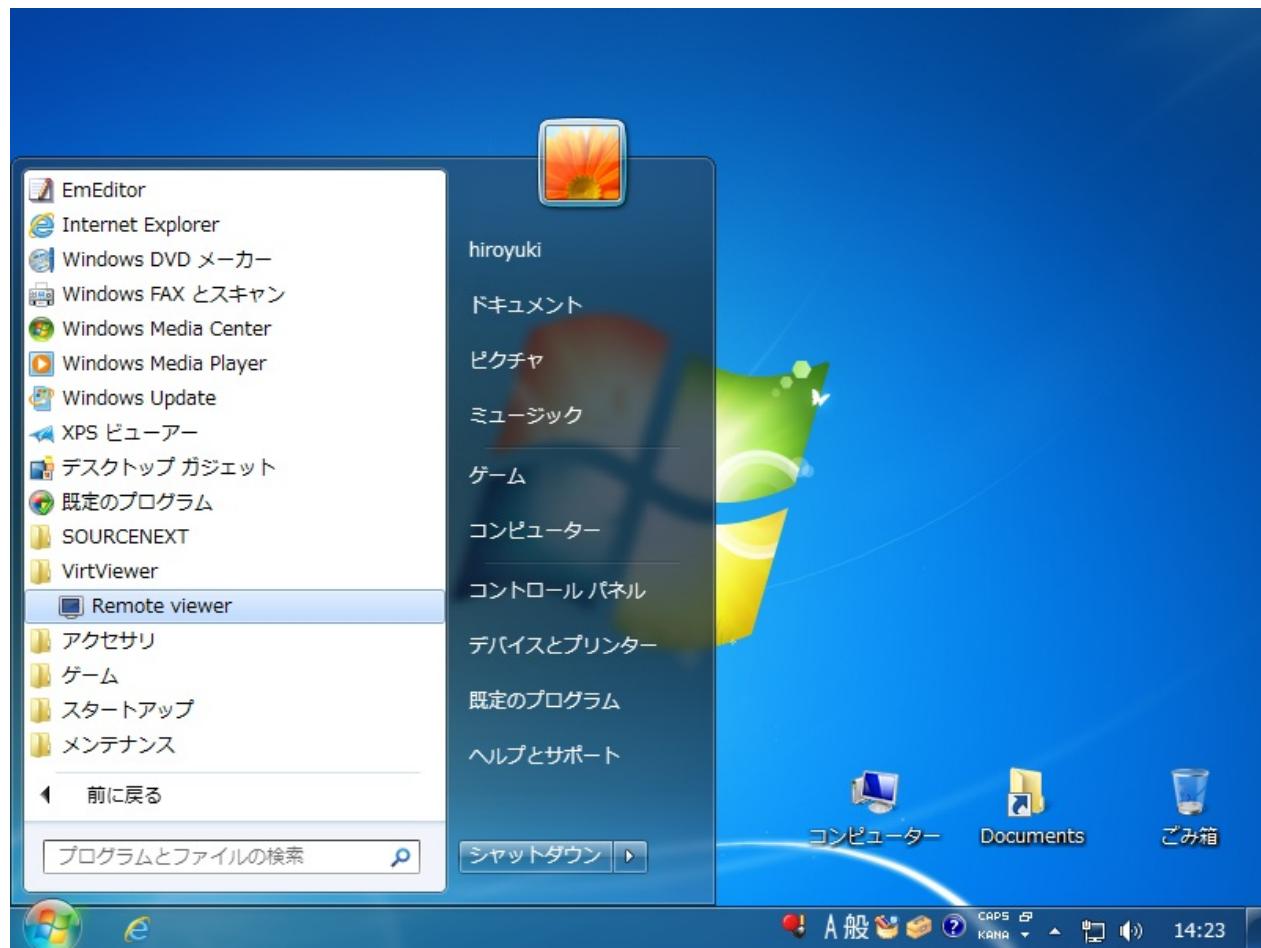


2.1.8.2. Windows客户端

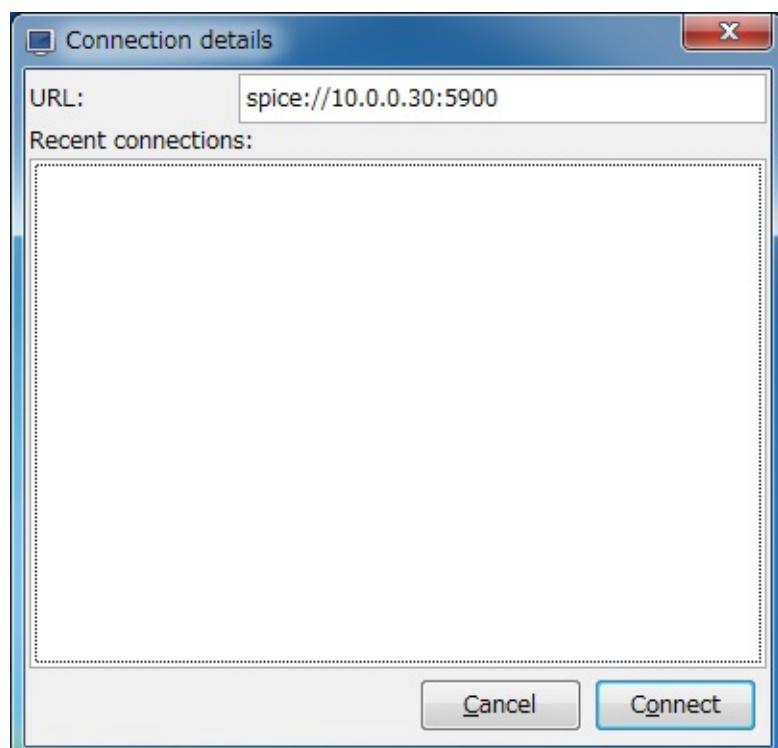
到[这里](#)下载Virt-Viewer安装程序。

安装并启动“Virt-Viewer”：

2.1. KVM

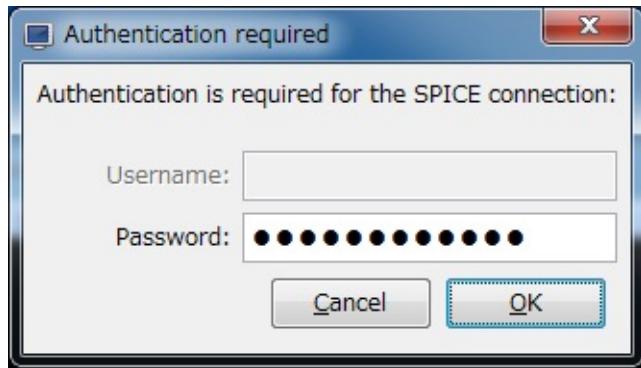


在URL字段中输入“spice://服务器名称或IP地址:端口”，然后单击“Connect”按钮：

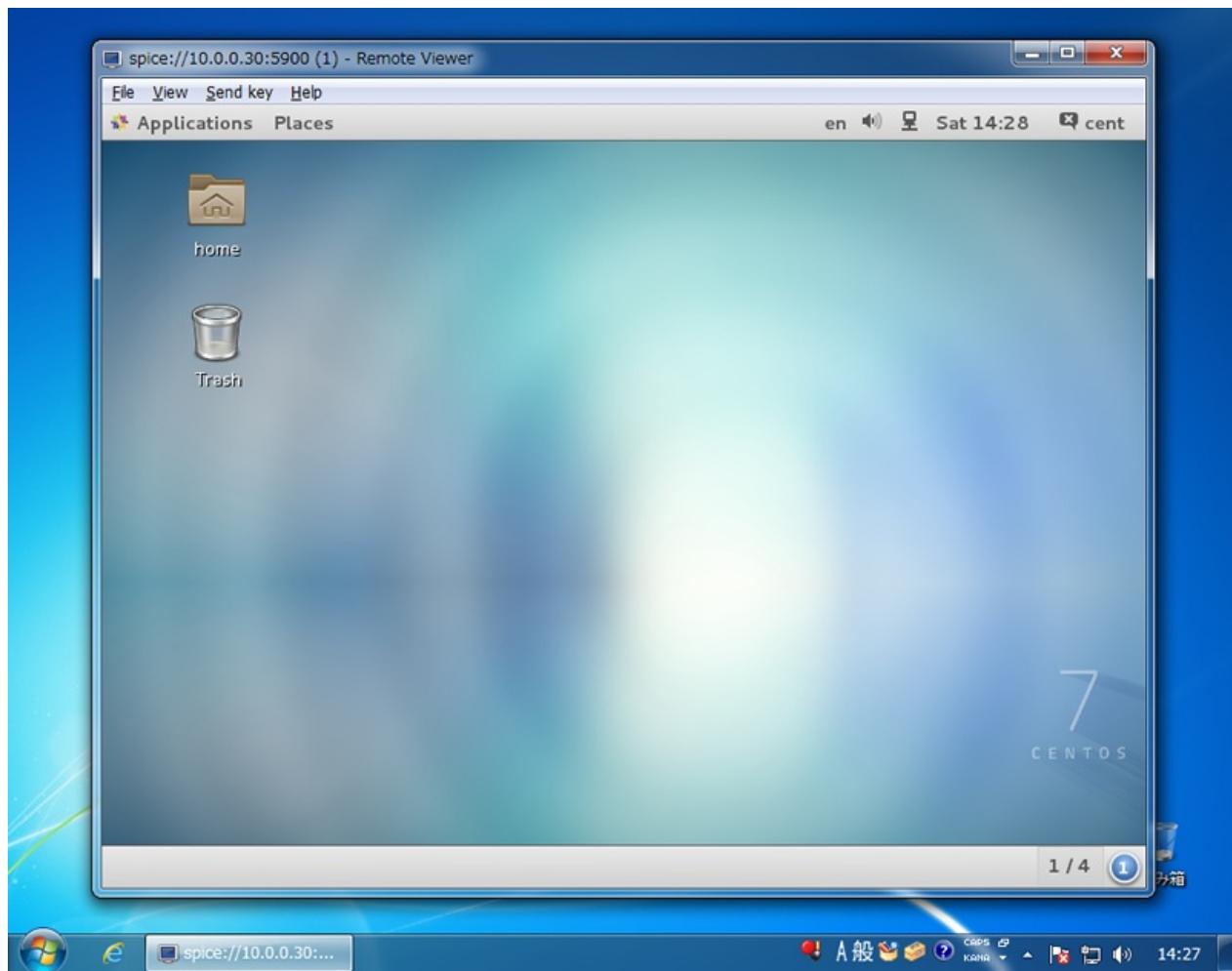


输入设置的密码，然后单击“OK”按钮：

2.1. KVM



成功认证之后：



2.1.9. KVM嵌套

配置KVM嵌套。可以在KVM主机上安装KVM并创建虚拟机作为KVM嵌套。

启用KVM嵌套的设置：

编辑 `/etc/modprobe.d/kvm-nested.conf` 文件，加入以下内容：

```
options kvm_intel nested=1
```

```
modprobe -r kvm_intel # 卸载
```

```
modprobe kvm_intel # 重载
```

```
cat /sys/module/kvm_intel/parameters/nested # 显示 Y 表示已启用
```

可以配置KVM嵌套，并且可以在嵌套的虚拟机上创建虚拟机。编辑要嵌套的虚拟机的配置，如下所示：

```
virsh edit centos7 # 编辑虚拟机“centos7”
```

```
<cpu mode='host-passthrough'> # 更改“cpu mode”部分
```

2.1.10. 快照管理

使用快照功能需要镜像文件是qcow2格式。

```
# 查看镜像文件格式，先停用虚拟机  
qemu-img info 镜像文件名称  
# 转换格式（未测试过，建议先备份，最好是新建虚拟机时就使用qcow2格式）  
qemu-img convert -f raw -O qcow2 镜像文件名称
```

快照命令：

Snapshot (help keyword 'snapshot')	
snapshot-create	使用 XML 生成快照
snapshot-create-as	使用一组参数生成快照
snapshot-current	获取或者设定当前快照
snapshot-delete	删除域快照
snapshot-dumpxml	为域快照转储 XML
snapshot-edit	编辑快照 XML
snapshot-info	快照信息
snapshot-list	为域列出快照
snapshot-parent	获取快照的上级快照名称
snapshot-revert	将域转换为快照

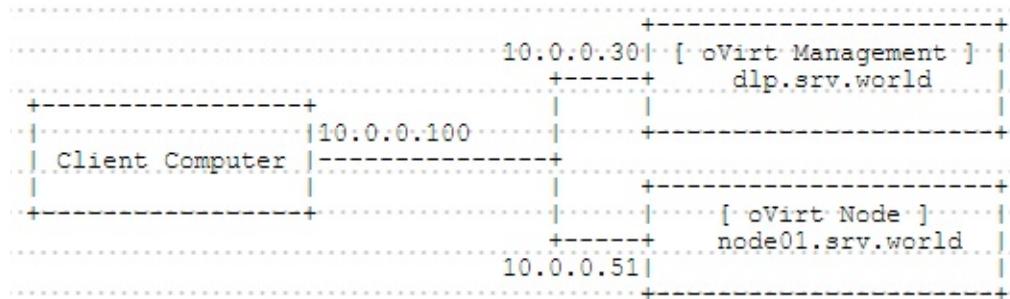
```
# 创建快照  
virsh snapshot-create-as 虚拟机名 快照名  
# 恢复快照  
virsh snapshot-revert 虚拟机名 快照名  
# 删除快照  
virsh snapshot-delete 虚拟机名 快照名
```

快照目录 /var/lib/libvirt/qemu/snapshot/虚拟机名

2.2. oVirt

oVirt是基于KVM项目的开源软件，该虚拟机软件支持主流的x86硬件，并允许用户在其上运行Linux及Windows操作系统。提供基于Web的虚拟机管理控制平台，无论是一台主机上的几个虚拟机，还是管理数百台主机上的成千个虚拟机，它皆能胜任。

本例基于以下环境：



2.2.1. 配置控制服务器

配置oVirt控制服务器：

```
yum -y install http://resources.ovirt.org/pub/yum-repo/ovirt-release35.rpm # 可以先查看版本，确认需要下载的链接
```

```
yum -y install ovirt-engine
```

```
touch /etc/exports
```

```
systemctl start rpcbind nfs-server
```

```
engine-setup
```

```
[ INFO ] Stage: Initializing
[ INFO ] Stage: Environment setup
          Configuration files: ['/etc/ovirt-engine-setup.conf.d/10-packaging-jboss.conf',
          '/etc/ovirt-engine-setup.conf.d/10-packaging.conf']
          Log file: /var/log/ovirt-engine/setup/ovirt-engine-setup-20150710215442-svdtg0.log
          Version: otopi-1.3.2 (otopi-1.3.2-1.el7.centos)
```

```

[ INFO ] Stage: Environment packages setup
[ INFO ] Stage: Programs detection
[ INFO ] Stage: Environment setup
[ INFO ] Stage: Environment customization

      === PRODUCT OPTIONS ===

      # 回车
Configure Engine on this host (Yes, No) [Yes]:
      # 回车
Configure WebSocket Proxy on this host (Yes, No) [Yes]

:

      === PACKAGES ===

[ INFO ] Checking for product updates...
[ INFO ] No product updates found

      === ALL IN ONE CONFIGURATION ===

      === NETWORK CONFIGURATION ===

Setup can automatically configure the firewall on this
system.
Note: automatic configuration of the firewall may over
write current settings.
      # 回车
Do you want Setup to configure the firewall? (Yes, No)
[Yes]:
The following firewall managers were detected on this
system: firewalld
      # 输入“firewalld”
Firewall manager to configure (firewalld): firewalld

[ INFO ] firewalld will be configured as firewall man
ager.
      # 指定此主机的FQDN（通常如下自动指定）
Host fully qualified DNS name of this server [dlp.srv.
world]:
      # 回车

      === DATABASE CONFIGURATION ===

```

```

# 选择本地数据库或远程数据库（本处选择本地）
Where is the Engine database located? (Local, Remote)

[Local]:
    Setup can configure the local postgresql server automatically for the engine to run.
    This may conflict with existing applications.

    # 选择自动或手动设置数据库（本处选择自动）
    Would you like Setup to automatically configure postgresql and create Engine database,
        or prefer to perform that manually? (Automatic, Manual)

) [Automatic]:


--- OVIRT ENGINE CONFIGURATION ---


# 设置oVirt管理员密码
Engine admin password:
Confirm engine admin password:
    # 选择应用程序模式（选择“Both”）

Application mode (Virt, Gluster, Both) [Both]:


--- PKI CONFIGURATION ---


# 指定证书的组织名称
Organization name for certificate [srv.world]:


--- APACHE CONFIGURATION ---


Setup can configure the default page of the web server to present the application home page.
This may conflict with existing applications.

    # 选择“Yes”或“No”（本处选择“Yes”）

Do you wish to set the application as the default page of the web server? (Yes, No) [Yes]:
    Setup can configure apache to use SSL using a certificate issued from the internal CA.

    # 选择自动或手动设置证书

Do you wish Setup to configure that, or prefer to perform
that manually? (Automatic, Manual) [Automatic]:

```

```

--== SYSTEM CONFIGURATION ==-

    # 选择“Yes”或“No”（本处选择“Yes”）
Configure an NFS share on this server to be used as an
ISO Domain? (Yes, No) [Yes]:
    # 指定本地ISO域的路径（本处保持默认）
Local ISO domain path [/var/lib/exports/iso]:
    # 指定本地ISO域的ACL（本处保持默认）
Local ISO domain ACL - note that the default will rest
rict access to dlp.srv.world only,
for security reasons [dlp.srv.world(rw)]:
    # 指定本地ISO域的名称（本处保持默认）
Local ISO domain name [ISO_DOMAIN]:


--== MISC CONFIGURATION ==-


--== END OF CONFIGURATION ==-


[ INFO ] Stage: Setup validation
[WARNING] Less than 16384MB of memory is available

--== CONFIGURATION PREVIEW ==-

Application mode : both
Firewall manager : firewalld
Update Firewall : True
Host FQDN       : dlp.srv.worl
d
Engine database name      : engine
Engine database secured connection : False
Engine database host       : localhost
Engine database user name  : engine
Engine database host name validation : False
Engine database port        : 5432
Engine installation        : True
NFS setup                 : True
PKI organization          : srv.world
NFS mount point           : /var/lib/exp
orts/iso
NFS export ACL            : dlp.srv.worl
d(rw)

```

2.2. oVirt

```
Configure local Engine database      : True
Set application as default page    : True
Configure Apache SSL                 : True
Configure WebSocket Proxy           : True
Engine Host FQDN                   : dlp.srv.worl
d

# 确认无误后回车
Please confirm installation settings (OK, Cancel) [OK]
:
[ INFO  ] Stage: Transaction setup
[ INFO  ] Stopping engine service
[ INFO  ] Stopping ovirt-fence-kdump-listener service
[ INFO  ] Stopping websocket-proxy service
[ INFO  ] Stage: Misc configuration
[ INFO  ] Stage: Package installation

.....
.....
[ INFO  ] Starting engine service
[ INFO  ] Restarting httpd
[ INFO  ] Stage: Clean up
          Log file is located at /var/log/ovirt-engine/setup/ovi
rt-engine-setup-20150710215442-svdtg0.log
[ INFO  ] Generating answer file '/var/lib/ovirt-engine/setup/an
swers/20150710215801-setup.conf'
[ INFO  ] Stage: Pre-termination
[ INFO  ] Stage: Termination
[ INFO  ] Execution of setup completed successfully
```

编辑 /etc/sysconfig/nfs 文件：

```
# 添加到最后
NFS4_SUPPORT="no"
```

```
mkdir /var/lib/exports/data
```

```
chown vdsm:kvm /var/lib/exports/data
```

编辑 /etc/exports.d/ovirt-engine-iso-domain.exports 文件：

```
# 添加数据的共享设置（如果需要，更改ACL设置）
/var/lib/exports/iso    10.0.0.0/24(rw)
/var/lib/exports/data   10.0.0.0/24(rw)
```

```
systemctl restart rpc-statd nfs-server
```

2.2.2. 配置节点

在oVirt节点安装KVM（不需要配置桥接网络）。

在oVirt节点安装一些必需的软件包（对于其他一些必需的设置，桥接网络等，将由oVirt控制服务器自动配置）：

```
yum -y install http://resources.ovirt.org/pub/yum-repo/ovirt-
release35.rpm
```

```
yum -y install vdsm
```

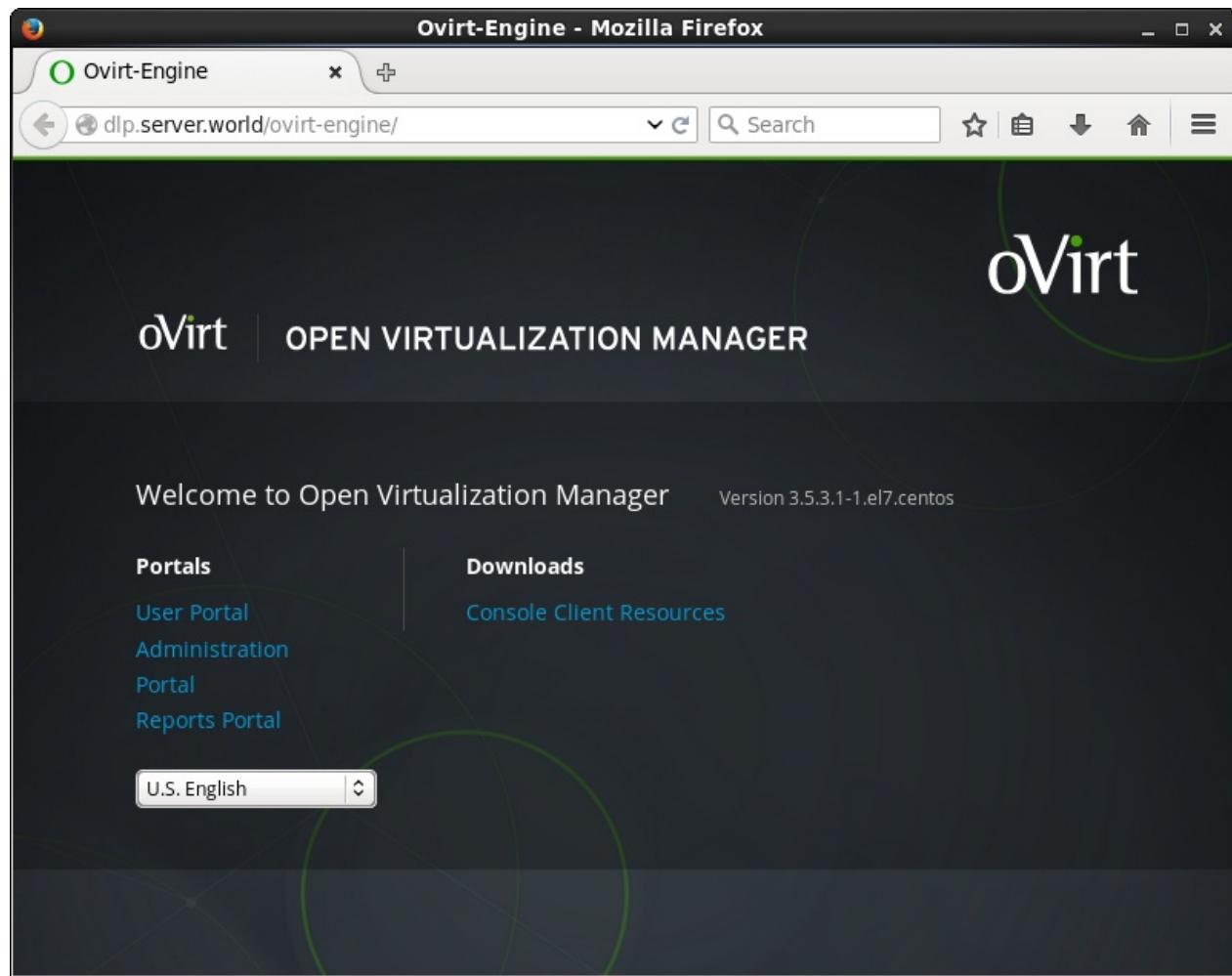
2.2.3. 添加管理的目标节点

在[安装了桌面环境的客户端上](#)安装Web浏览器和Spice扩展（如果不[在oVirt管理门户上](#)连接到虚拟机的控制台，可以不用安装Spice扩展。也可以在Windows计算机上使用Firefox或Chrome操作）：

```
yum -y install firefox spice-xpi
```

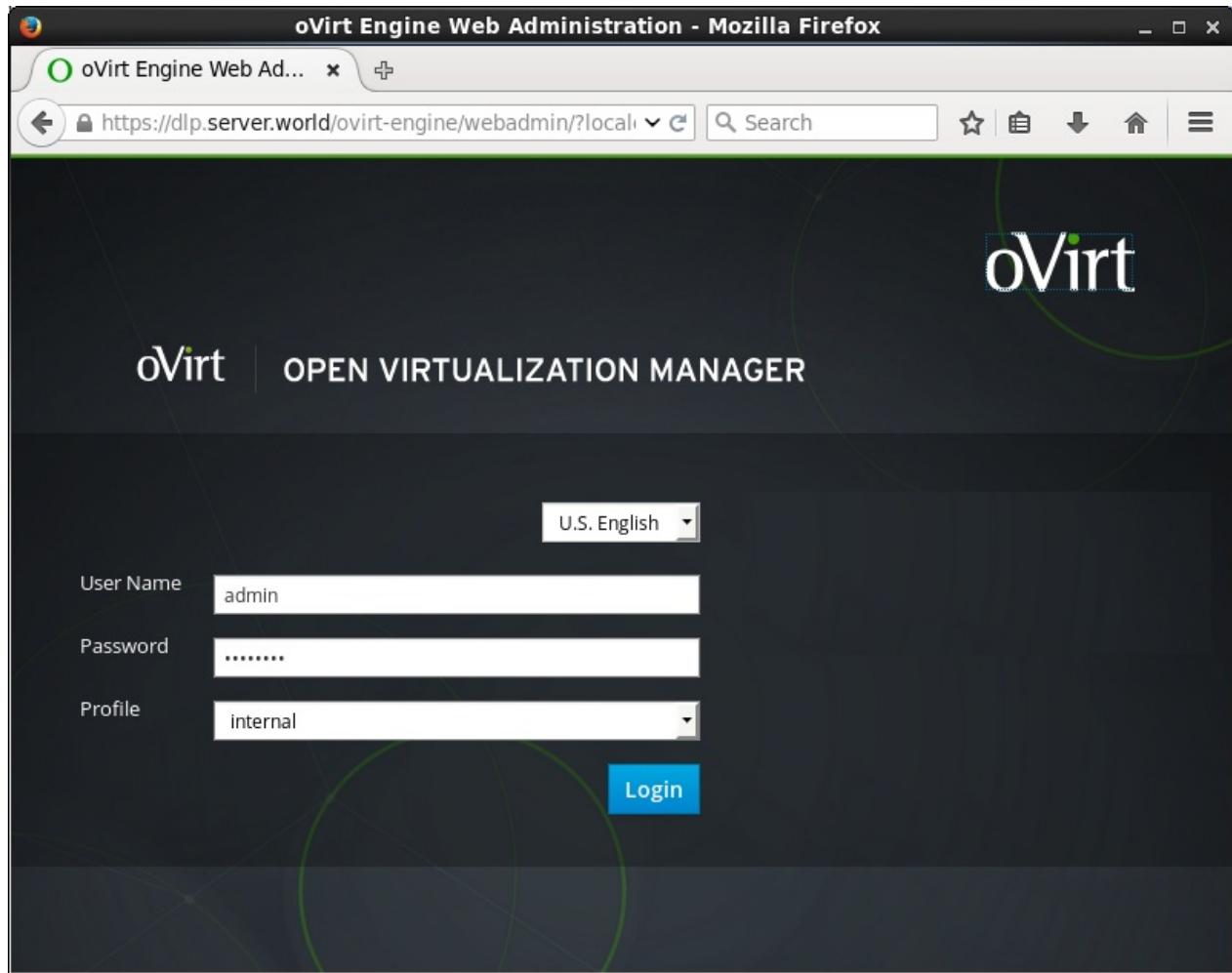
使用Web浏览器访问 [http://\(oVirt控制服务器的主机名或IP地址\)/](http://(oVirt控制服务器的主机名或IP地址)/)，然后点击“Administration Portal”：

2.2. oVirt



使用在安装oVirt控制服务器期间设置的管理员用户密码登录：

2.2. oVirt



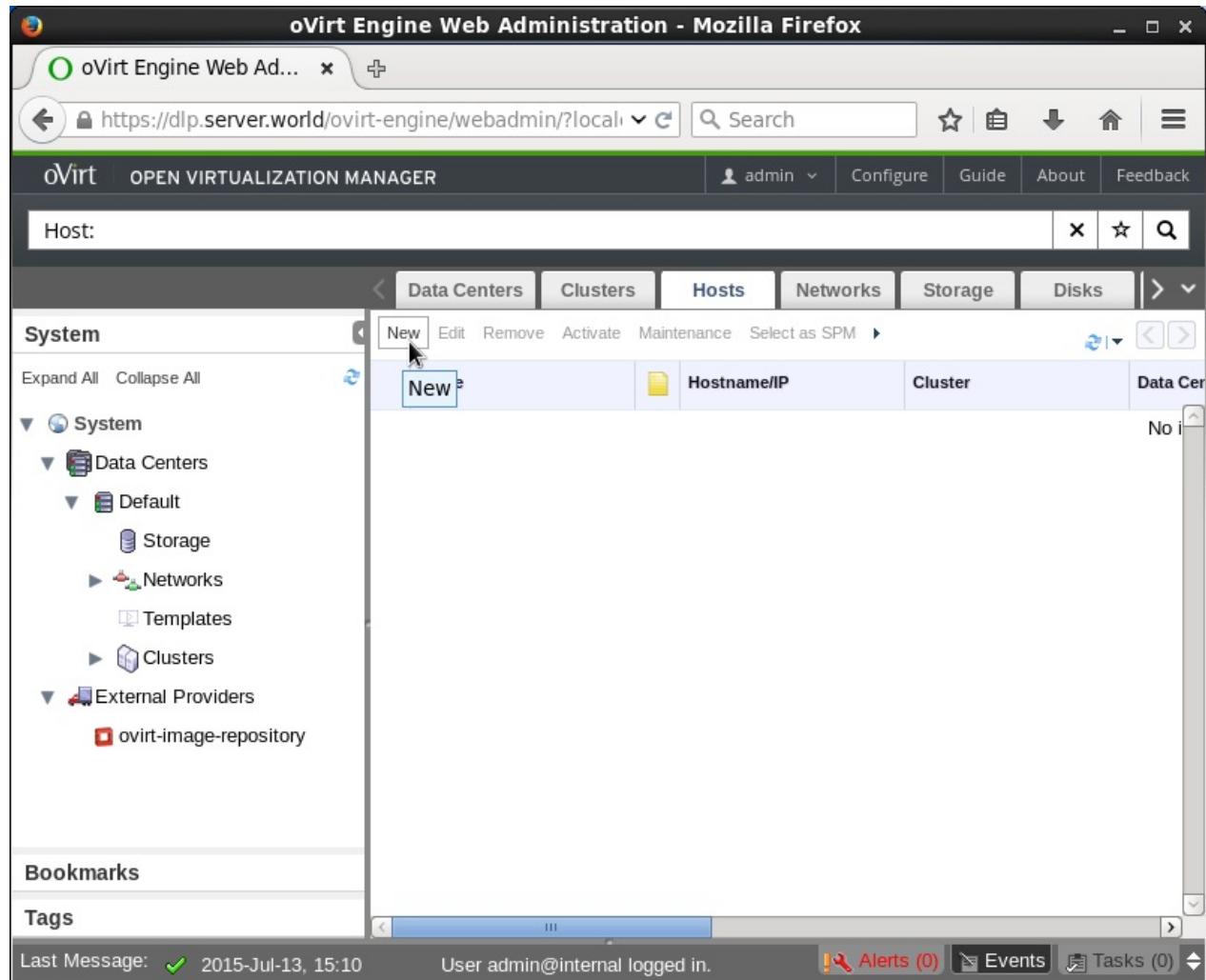
登录成功，下面为oVirt管理门户的主页：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface. The browser title bar reads "oVirt Engine Web Administration - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world/ovirt-engine/webadmin/?local". The main header bar includes the "oVirt OPEN VIRTUALIZATION MANAGER" logo, a user dropdown set to "admin", and links for "Configure", "Guide", "About", and "Feedback". A search bar with placeholder "Vms:" and a "Data Centers" button are also present. The left sidebar, titled "System", contains sections for "Data Centers" (with "Default" selected), "Networks", "Templates", "Clusters", and "External Providers" (with "ovirt-image-repository" listed). The right panel displays a table with columns: Name, Host, IP Address, and FQDN. The table currently has no data rows.

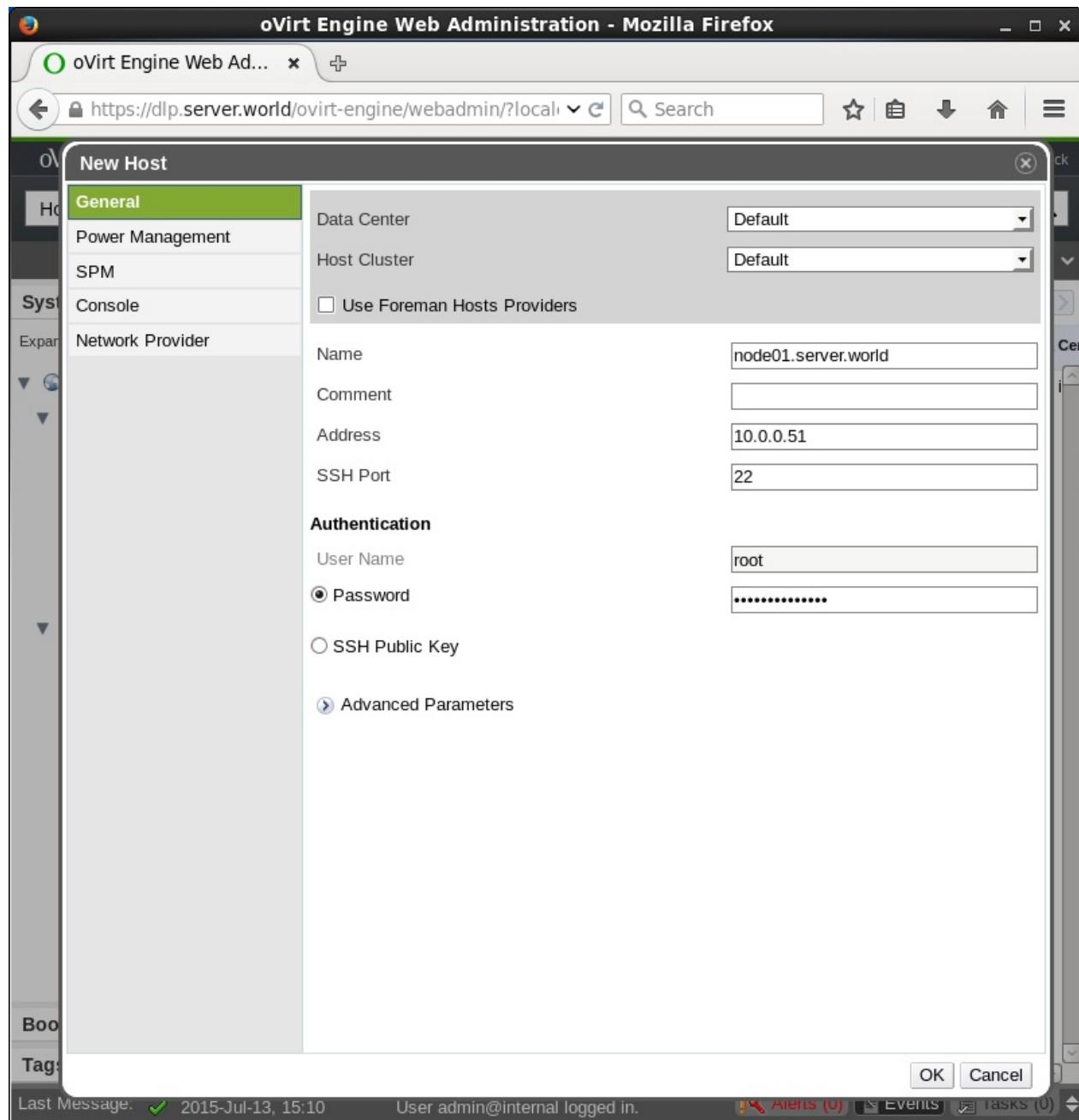
添加目标节点以进行管理。转到“Hosts”标签，然后点击“New”：

2.2. oVirt



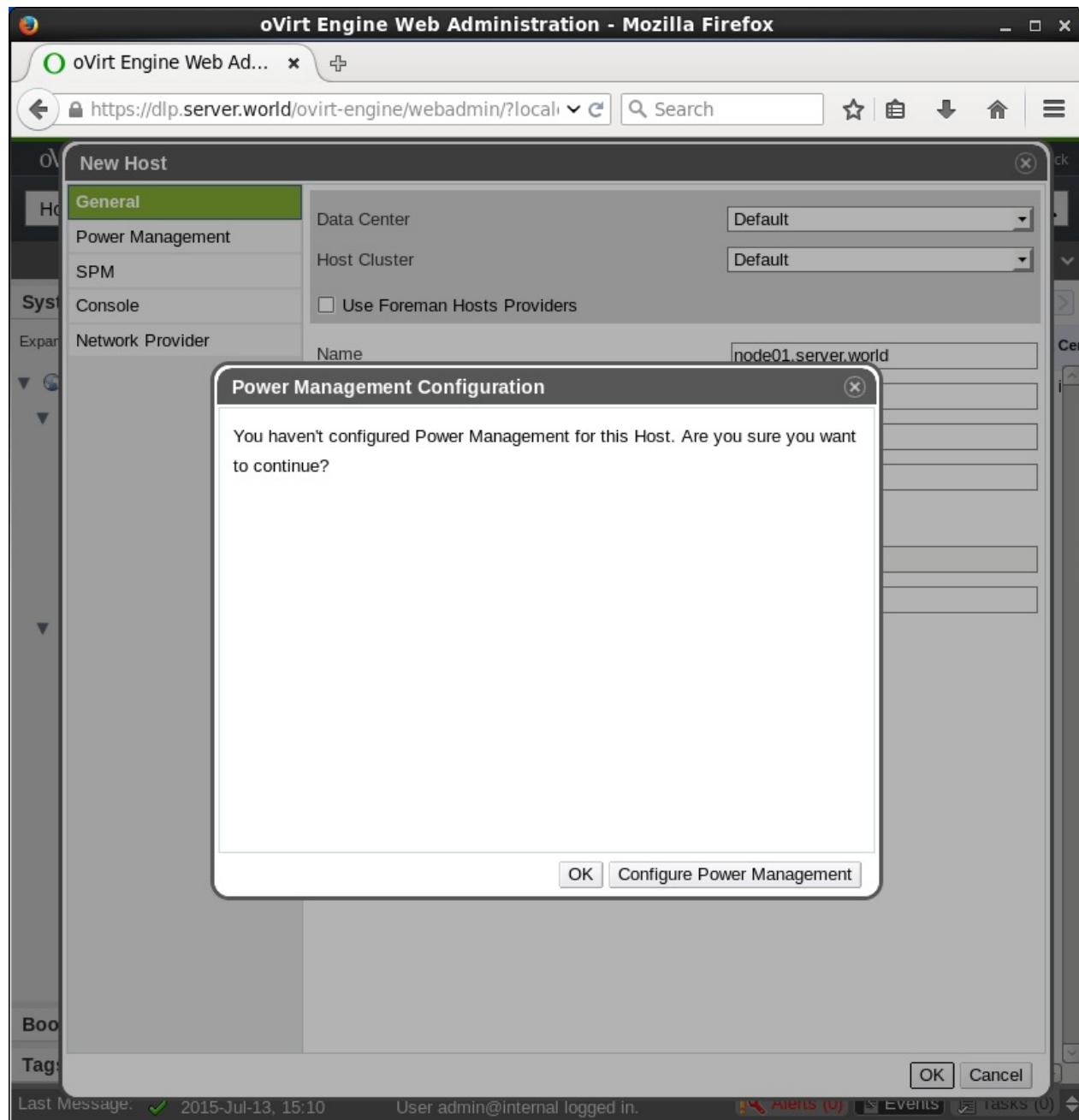
输入要添加的节点的信息，然后点击“OK”：

2.2. oVirt



配置“Power Management”。此功能启用由HP或Dell制造的特定硬件。因此，如果您的机器有对应则配置，如果没有则不配置，点击“OK”继续：

2.2. oVirt



添加节点后，它的条目将如下所示显示在“Hosts”标签上（从oVirt控制服务器到节点的自动安装和配置是在后台运行，需要等待几分钟，直到完成）：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The title bar reads "oVirt Engine Web Administration - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world/ovirt-engine/webadmin/?local". The top navigation bar includes links for "Hosts", "Data Centers", "Clusters", "Networks", "Storage", and "Disks". A search bar and user authentication dropdown are also present.

The left sidebar menu is expanded, showing the "System" section with "Data Centers" and "External Providers". Under "Data Centers", "Default" is selected, showing sub-options for "Storage", "Networks", "Templates", and "Clusters". Under "External Providers", "ovirt-image-repository" is listed.

The main content area displays a table for hosts. A single host, "node01.server.world", is listed with the IP "10.0.0.51" and assigned to the "Default" cluster. Below the table, a detailed view of the host's configuration is shown in tabs: General, Virtual Machines, Network Interfaces, Host Hooks, and Permission. The "General" tab displays various system parameters like OS Version, Kernel Version, KVM Version, LIBVIRT Version, VDSM Version, SPICE Version, GlusterFS Version, SPM Priority, Active VMs, Logical CPU Cores, Online Logical CPU Cores, Boot Time, Hosted Engine HA, iSCSI Initiator Name, and Kdump Status.

At the bottom of the interface, there is a "Bookmarks" section, a "Tags" section, and a message bar indicating the last message was received on 2015-Jul-13, 15:14. Action items at the bottom include "Power Management is not configured for this Host. Enable Power Management", "Installing Host node01.server.world. Stag...", "Alerts (1)", "Events", and "Tasks (1)".

几分钟后，显示系统信息，节点激活如下。可以添加节点：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The title bar reads "oVirt Engine Web Administration - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world/ovirt-engine/webadmin/?local=true". The top navigation bar includes links for "Hosts", "Data Centers", "Clusters", "Networks", "Storage", and "Disks". The main left sidebar under "System" shows a tree structure with "System", "Data Centers" (selected), "External Providers", and "Tags". The "Data Centers" section has a single entry "Default". The main content area displays details for the host "node01.server.world" (IP 10.0.0.51). The host status table includes:

Name	Hostname/IP	Cluster	Data Center
node01.server.world	10.0.0.51	Default	Default

Below the table, a summary table provides system information:

OS Version:	RHEL - 7 - 1.15C	SPM Priority:	Medium
Kernel Version:	3.10.0 - 229.4.2.	Active VMs:	0
KVM Version:	2.1.2 - 23.el7_1	Logical CPU Cores:	4
LIBVIRT Version:	libvirt-1.2.8-16.el7_1	Online Logical CPU Cores:	0,1,2,3
VDSM Version:	vdsm-4.16.20-0	Boot Time:	2015-Jul-13, 14:37
SPICE Version:	0.12.4 - 9.el7	Hosted Engine HA:	[N/A]
GlusterFS Version:	[N/A]	iSCSI Initiator Name:	iqn.1994-05.com.redhat:node01
		Kdump Status:	Disabled

At the bottom, there are sections for "Bookmarks" and "Tags", and a message "Last Message: 2015-Jul-13, 15:15". The footer includes links for "Alerts (1)", "Events", and "Tasks (0)".

2.2.4. 添加存储

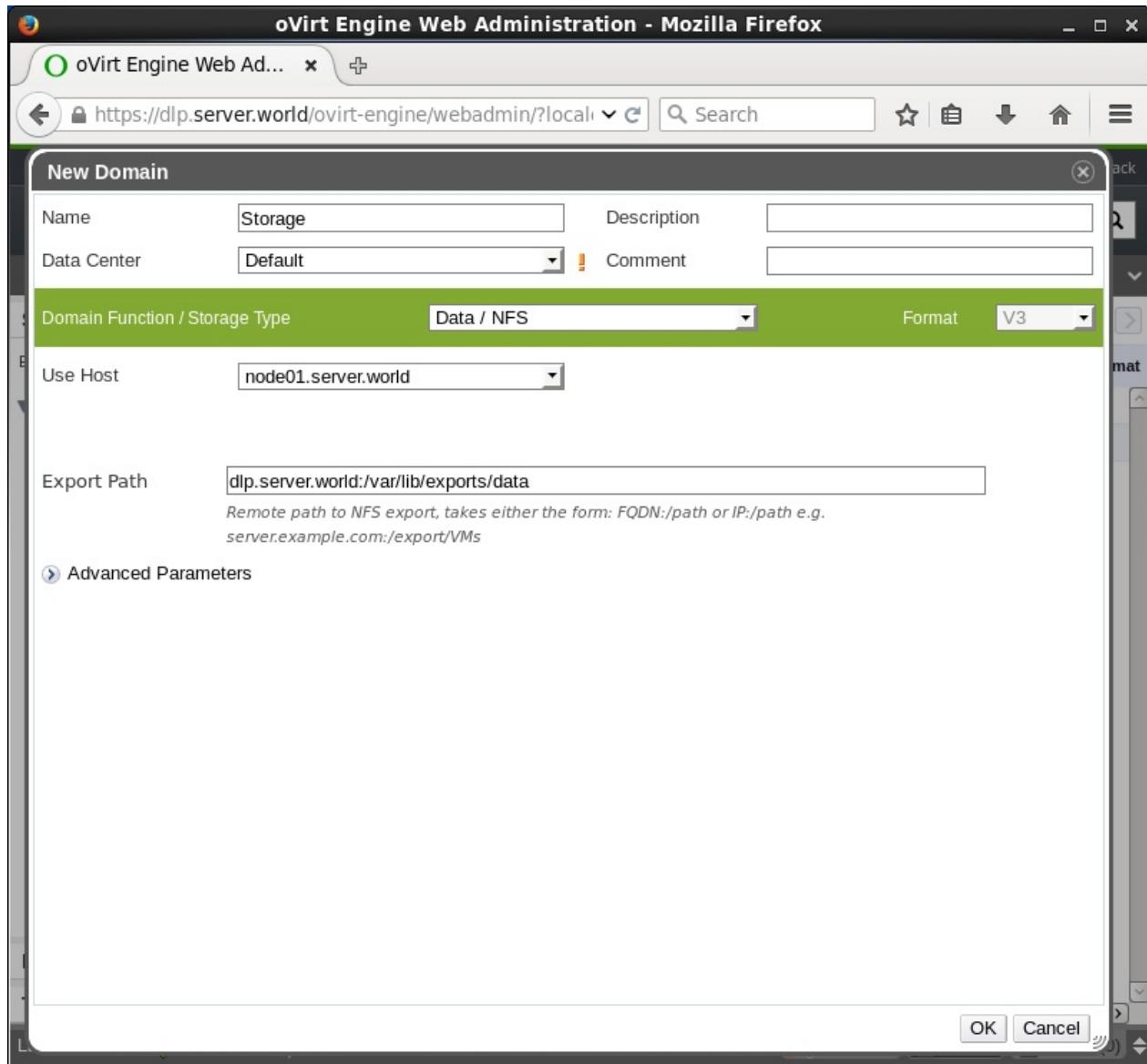
登录到oVirt管理门户并转到“Storage”标签，然后点击“New Domain”：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The title bar reads "oVirt Engine Web Administration - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world/ovirt-engine/webadmin/?local". The top navigation bar includes links for "Data Centers", "Clusters", "Hosts", "Networks", "Storage" (which is highlighted with a red box), and "Disks". Below the navigation bar is a search bar and a user dropdown. The main content area has a sidebar titled "System" with sections for "Data Centers", "External Providers", "Bookmarks", and "Tags". The main panel displays a table for "Domain" management. A new row is being added, indicated by a blue border around the "New Domain" button. The table columns are "Name", "Domain Type", "Storage Type", and "Format". Two existing entries are shown: "ISO_DOMAIN" (ISO, NFS, V1) and "ovirt-image-repository" (Image, OpenStack Glance, V1). At the bottom of the interface, there are status messages, an "Alerts (1)" icon, and links for "Events" and "Tasks (0)".

在“Name”字段输入任意名称。在“Export Path”字段输入在oVirt控制服务器设置中添加的存储路径：

2.2. oVirt



添加后，如下列出：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The title bar reads "oVirt Engine Web Administration - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world/ovirt-engine/webadmin/?local=1&view=storage&domain_id=1". The top navigation bar includes links for "Data Centers", "Clusters", "Hosts", "Networks", "Storage", and "Disks". The "Storage" tab is selected. The left sidebar under "System" shows a tree structure with "System", "Data Centers" (selected), "External Providers", and "Tags". Under "Data Centers", there are "Default", "Storage", "Networks", "Templates", and "Clusters". The main content area displays a table of storage domains:

	Domain Name	Domain Type	Storage Type	Format
ISO_DOMAIN	ISO	NFS	V1	
ovirt-image-repository	Image	OpenStack Glance	V1	
Storage	Data (Master)	NFS	V3	

Below the table, detailed information for the "Storage" domain is shown:

General	Data Center	Virtual Machines	Templates	Disks	Disk S
Size: 26 GB					
Available: 24 GB					
Used: 2 GB					
Allocated: < 1 GB					
Over Allocation Ratio: 0%					

Path: dlp.server.world:/var/lib/exports/data

At the bottom, a message says "Last Message: 2015-Jul-13, 15:33 Storage Domain Storage was attached t...". There are also "Alerts (1)", "Events", and "Tasks (0)" buttons.

在顶部窗格中选择“ISO_DOMAIN”，并转到“Data Center”标签，然后点击“Attach”按钮，如下所示：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The main title bar reads "oVirt Engine Web Administration - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world/ovirt-engine/webadmin/?local=1&view=storage&domain_id=1". The top navigation bar includes links for "Data Centers", "Clusters", "Hosts", "Networks", "Storage", and "Disks". The "Storage" tab is selected. The left sidebar under "System" shows "Data Centers" expanded, with "Default" selected. The main content area displays a table of storage domains:

Domain Name	Domain Type	Storage Type	Format
ISO_DOMAIN	ISO	NFS	V1
ovirt-image-repository	Image	OpenStack Glance	V1
Storage	Data (Master)	NFS	V3

A red box highlights the "ISO_DOMAIN" row. Below the table, there is a "Data Center" tab and an "Attach" button, both highlighted with red boxes. A tooltip "Attach" is shown over the "Attach" button. The status bar at the bottom indicates "Storage Domain Storage was attached t...".

选中“default”复选框，然后点击“OK”：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The main menu bar includes 'File', 'Edit', 'View', 'Search', 'Configure', 'Guide', 'About', and 'Feedback'. The top navigation bar shows the URL 'https://dlp.server.world/ovirt-engine/webadmin/?local' and a search bar. The main content area is titled 'Storage:' and contains tabs for 'Data Centers', 'Clusters', 'Hosts', 'Networks', 'Storage', and 'Disks'. On the left, a sidebar under 'System' shows 'Data Centers' (Default, Storage, Networks, Templates, Clusters) and 'External Providers' (ovirt-image-repository). A central modal dialog is titled 'Attach to Data Center' and lists a single item: 'Default' (Storage Type: Shared). To the right of the modal is a table showing storage domains: 'NFS V1', 'OpenStack Glance V1', and 'NFS V3'. At the bottom of the screen, a message bar indicates 'Storage Domain Storage was attached t...' and shows 'Alerts (1)', 'Events', and 'Tasks (0)'.

确认“ISO_DOMAIN”为“Active”，所有完成：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The left sidebar displays the navigation tree under 'System' with sections for Data Centers, Clusters, Hosts, Networks, Storage, Disks, External Providers, Bookmarks, and Tags. The main content area is titled 'Storage:' and shows a table of storage domains. The table has columns for Domain Name, Domain Type, Storage Type, and Format. It lists three domains: ISO_DOMAIN (ISO, NFS, V1), ovirt-image-repository (Image, OpenStack Glance, V1), and Storage (Data (Master), NFS, V3). Below the table, there are tabs for General, Data Center, Images, Permissions, and Events. The General tab shows options like Attach, Detach, Activate, and Maintenance, with 'Default' selected. The bottom status bar shows a message about ISO_DOMAIN being attached and an alert for 1 event.

Domain Name	Domain Type	Storage Type	Format
ISO_DOMAIN	ISO	NFS	V1
ovirt-image-repository	Image	OpenStack Glance	V1
Storage	Data (Master)	NFS	V3

2.2.5. 创建虚拟机

在oVirt管理门户上创建虚拟机。

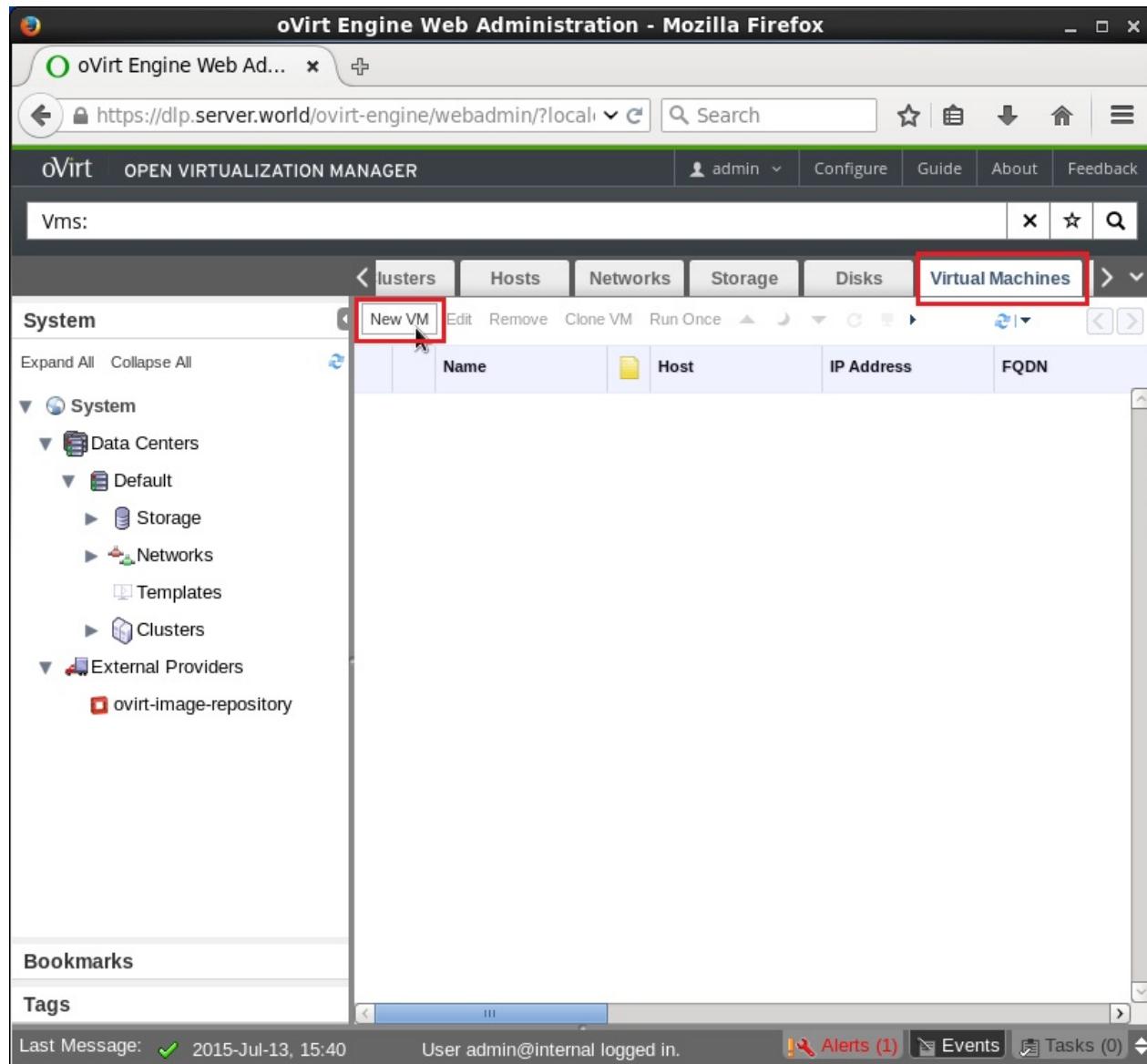
在oVirt控制服务器上下载要为客户机安装的ISO文件。接下来，按如下方式上传：

```
engine-iso-uploader -i ISO_DOMAIN upload /tmp/CentOS-7-x86_64-DVD-1503-01.iso # 命令格式： engine-iso-uploader -i [ISO Domain] upload [ISO File]
```

2.2. oVirt

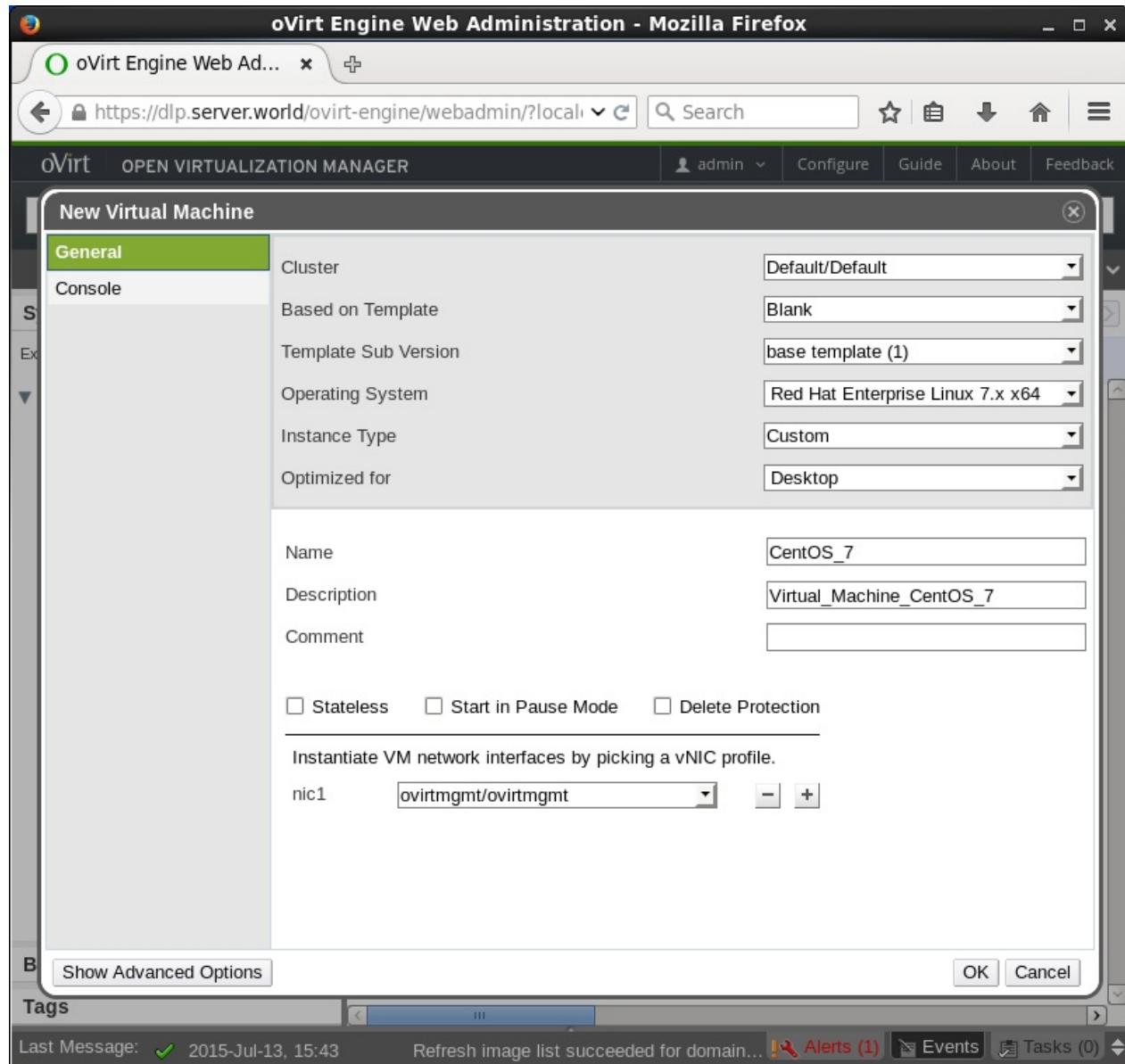
```
Please provide the REST API password for the admin@internal oVirt Engine user (CTRL+D to abort): # oVirt管理员密码  
Uploading, please wait...  
INFO: Start uploading /tmp/CentOS-7-x86_64-DVD-1503-01.iso  
INFO: /tmp/CentOS-7-x86_64-DVD-1503-01.iso uploaded successfully
```

访问oVirt管理门户并转到“Virtual Machine”标签，然后点击“New VM”：



输入要创建的虚拟机的值，然后点击“OK”：

2.2. oVirt



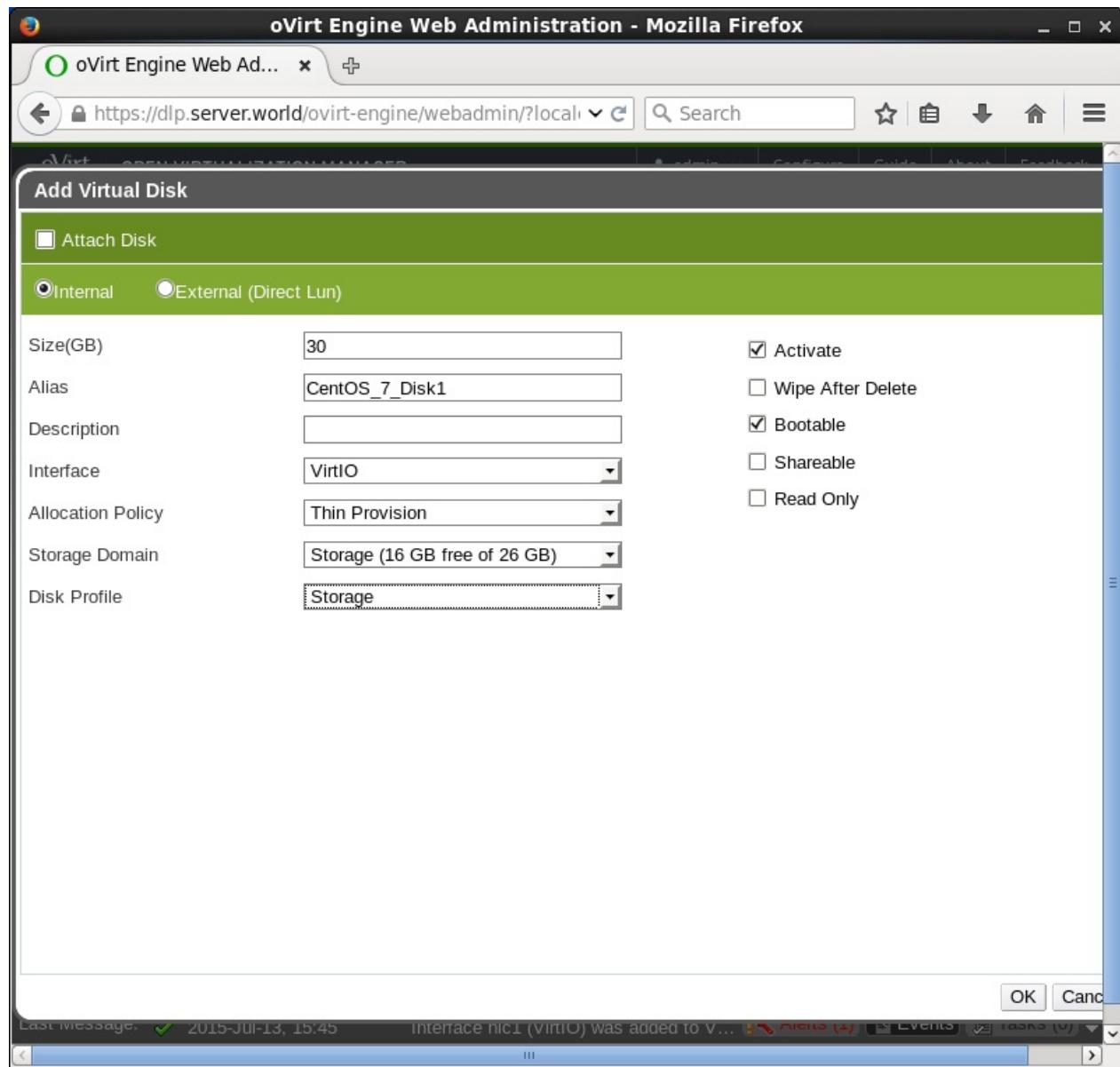
点击“Configure Virtual Disks”：

2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The main window displays the 'System' section of the navigation tree, which includes 'Clusters', 'Hosts', 'Networks', 'Storage', 'Disks', and 'Virtual Machines'. A modal dialog box titled 'New Virtual Machine - Guide Me' is open, indicating that a 'Virtual Machine created.' and prompting the user to 'Configure Virtual Disks'. The configuration panel shows basic settings: Priority: Low, Number of Monitors: 1, and USB Policy: Disabled. The background shows a list of external providers, including 'ovirt-image-repository'. At the bottom, a message bar indicates a recent event: 'Interface nic1 (VirtIO) was added to VM ...'.

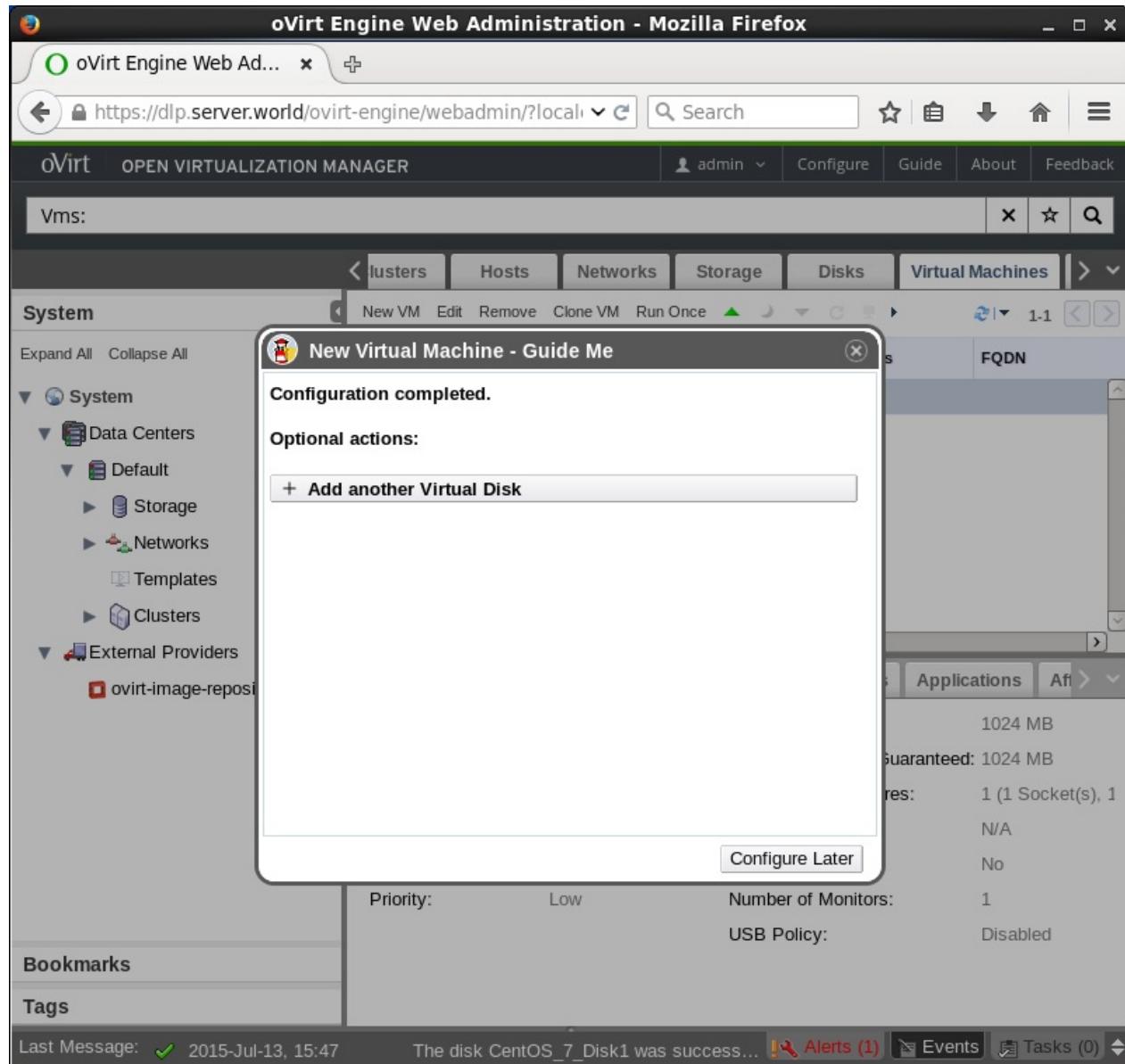
输入虚拟磁盘的值，然后点击“OK”：

2.2. oVirt



点击“Configure Later”（如果想添加更多虚拟磁盘，点击“Add another Virtual Disk”）：

2.2. oVirt



点击“Run Once”以启动虚拟机：

2.2. oVirt

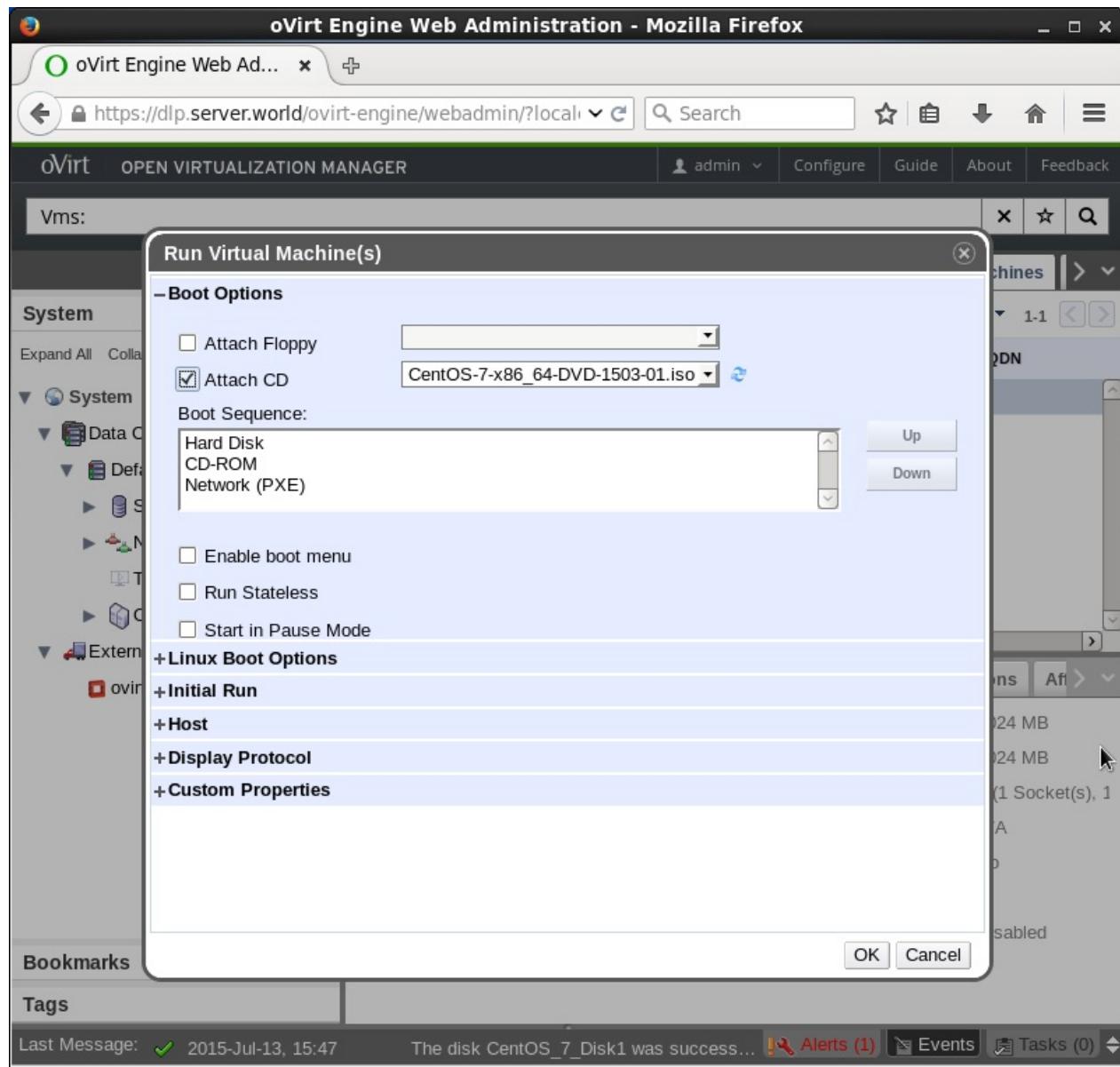
The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The title bar reads "oVirt Engine Web Administration - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world/ovirt-engine/webadmin/?local". The top navigation bar includes links for "Clusters", "Hosts", "Networks", "Storage", "Disks", "Virtual Machines", and user authentication ("admin"). Below the navigation is a search bar and a toolbar with icons for "Configure", "Guide", "About", and "Feedback". A sidebar on the left titled "System" lists "Data Centers" (Default, Storage, Networks, Templates, Clusters) and "External Providers" (ovirt-image-repository). The main content area is titled "Virtual Machines" and shows a table with columns: Name, Host, Run Once, IP Address, and FQDN. A row for "CentOS_7" is selected, highlighted with a blue border. The "Run Once" button is highlighted with a mouse cursor. The details panel below the table shows the following configuration for the CentOS_7 VM:

Name:	CentOS_7	Defined Memory:	1024 MB
Description:	Virtual_Machine_	Physical Memory Guaranteed:	1024 MB
Template:	Blank	Number of CPU Cores:	1 (1 Socket(s), 1
Operating System:	Red Hat Enterpri	Guest CPU Count:	N/A
Default Display Type:	SPICE	Highly Available:	No
Priority:	Low	Number of Monitors:	1
		USB Policy:	Disabled

At the bottom of the interface, there are "Bookmarks" and "Tags" sections, and a status bar showing "Last Message: 2015-Jul-13, 15:47" and "The disk CentOS_7_Disk1 was success...".

选中“Attach CD”复选框以安装客户机虚拟机进行初始引导，然后单击“OK”：

2.2. oVirt



点击“Console”按钮：

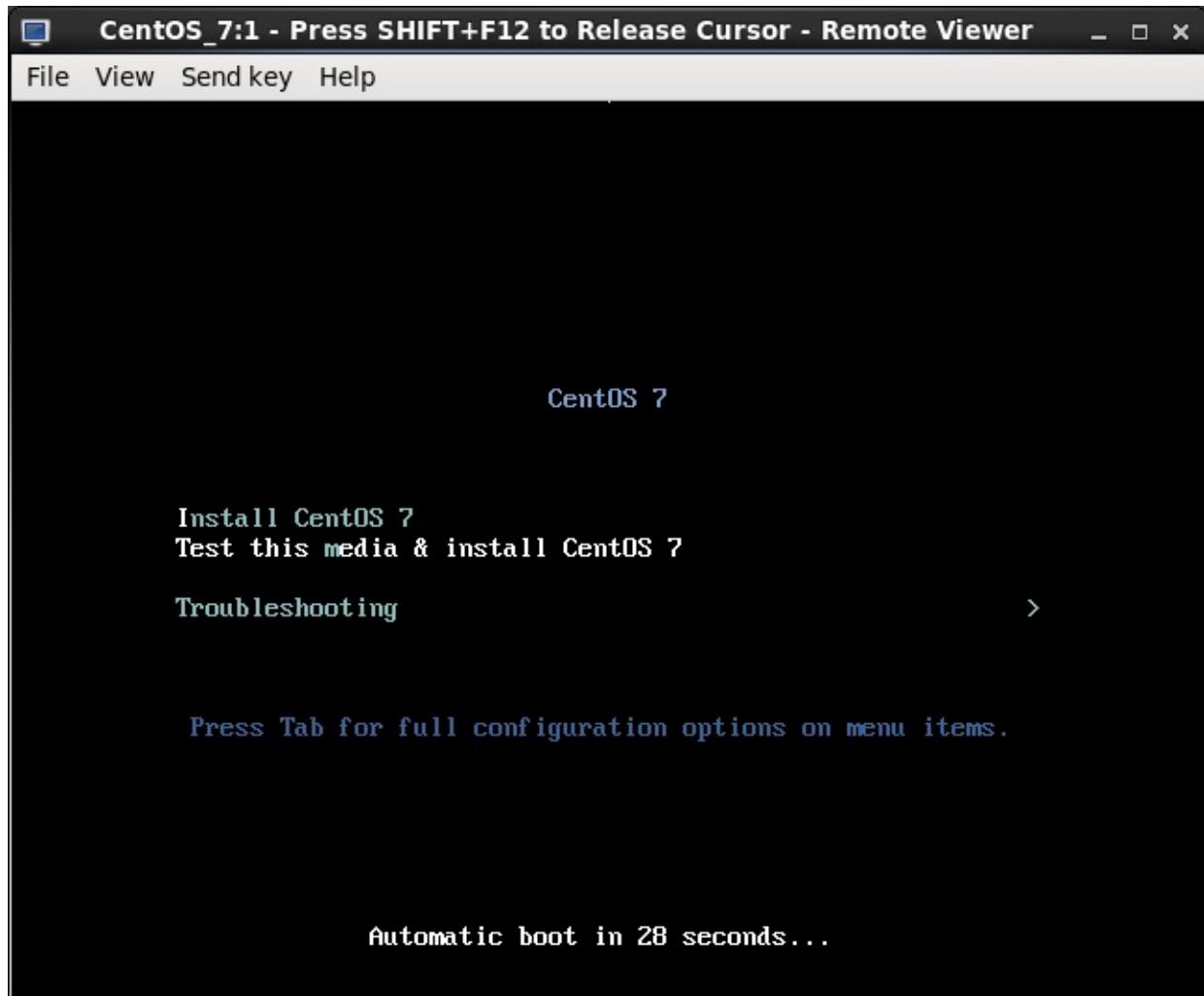
2.2. oVirt

The screenshot shows the oVirt Engine Web Administration interface in Mozilla Firefox. The main navigation bar includes 'Clusters', 'Hosts', 'Networks', 'Storage', 'Disks', and 'Virtual Machines'. The 'Virtual Machines' tab is active. On the left, a sidebar shows the 'System' section with 'Data Centers' expanded, showing 'Default' (with 'Storage', 'Networks', 'Templates', and 'Clusters'), and 'External Providers' (with 'ovirt-image-repository'). The main content area displays a table of virtual machines. One row for 'CentOS_7' is selected, with its details shown in a large panel below. The 'Console' button in this panel is highlighted with a red box. The details panel contains the following configuration:

Name:	Defined Memory:
CentOS_7	1024 MB
Description:	Physical Memory Guaranteed:
Virtual_Machine_	1024 MB
Template:	Number of CPU Cores:
Blank	1 (1 Socket(s), 1
Operating System:	Guest CPU Count:
Red Hat Enterpri	N/A
Default Display Type:	Highly Available:
SPICE	No
Priority:	Number of Monitors:
Low	1
	USB Policy:
	Disabled

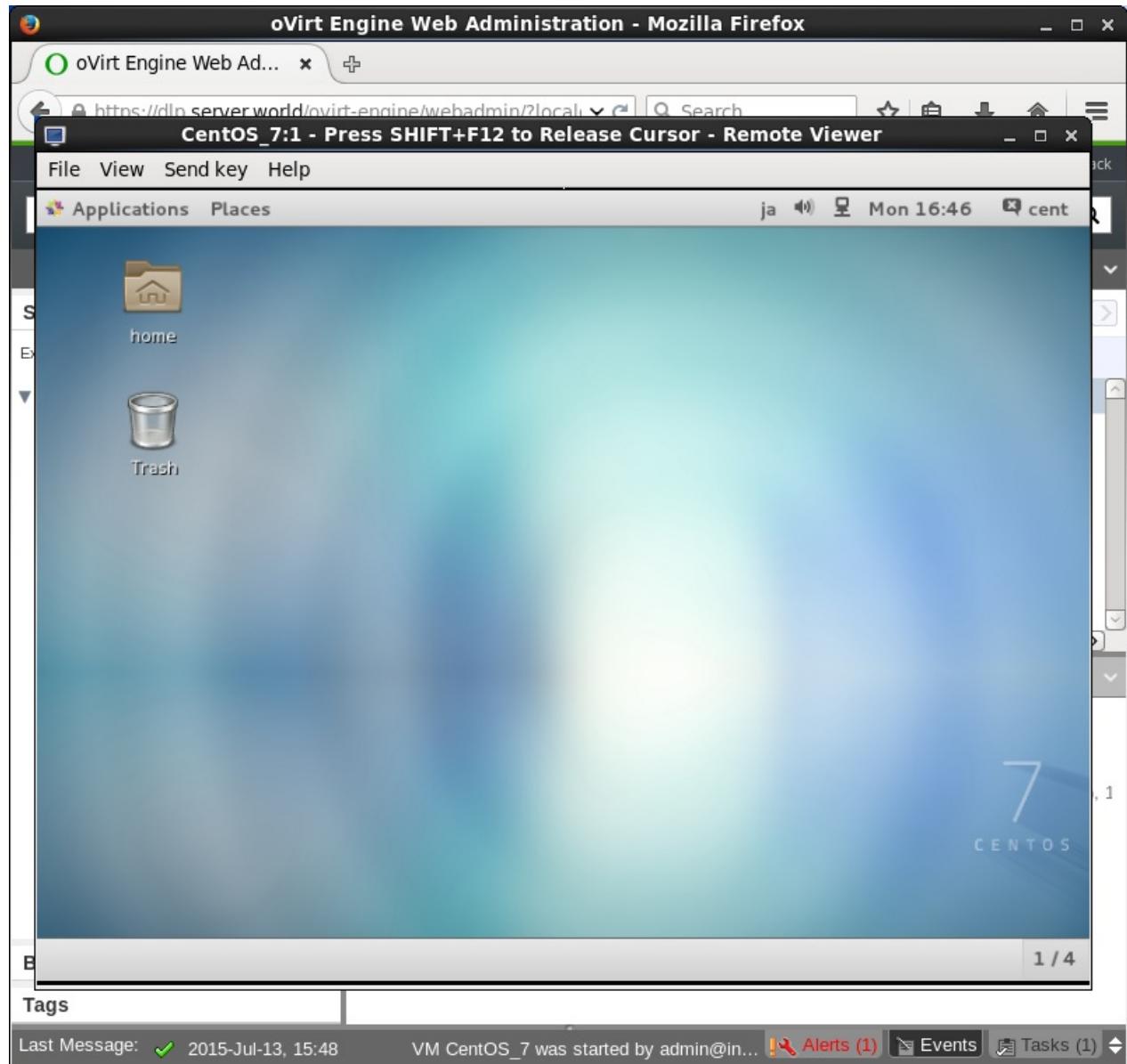
At the bottom, a message indicates 'VM CentOS_7 was started by admin@in...' and shows links for 'Alerts (1)', 'Events', and 'Tasks (1)'.

显示虚拟机屏幕，继续安装：



安装完成，虚拟机运行：

2.2. oVirt



2.3. Xen

Xen是一个开放源代码虚拟机监视器，由剑桥大学开发。

2.3.1. 安装Xen

启用CentOS Xen存储库并安装一些软件包：

```
yum -y install centos-release-xen  
sed -i -e "s(enabled=1|enabled=0)/g" /etc/yum.repos.d/CentOS-Xen.repo  
yum --enablerepo=centos-virt-xen -y update kernel  
yum --enablerepo=centos-virt-xen -y install xen
```

编辑 /etc/default/grub 文件：

```
# 更改Domain0的内存量（在系统上指定适当的值，4096M替换为自己要指定的值）  
GRUB_CMDLINE_XEN_DEFAULT="dom0_mem=4096M,max:4096M cpuinfo com1=115200,8n1 ....
```

```
/bin/grub-bootxen.sh  
reboot  
xl info # 显示信息
```

```

host                  : dlp.srv.world
release              : 3.18.21-17.el7.x86_64
version              : #1 SMP Fri Dec 18 18:04:14 UTC 2015
machine              : x86_64
nr_cpus              : 6
max_cpu_id          : 5
nr_nodes             : 1
cores_per_socket     : 1
threads_per_core    : 1
cpu_mhz              : 2594
hw_caps              : xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
virt_caps            : hvm
total_memory         : 12287
free_memory          : 8054
sharing_freed_memory: 0
sharing_used_memory : 0
outstanding_claims  : 0
free_cpus            : 0
xen_major            : 4
xen_minor            : 6
xen_extra            : .0-9.el7
xen_version          : 4.6.0-9.el7
xen_caps             : xen-3.0-x86_64 xen-3.0-x86_32p hvm-3.0-
x86_32 hvm-3.0-x86_32p hvm-3.0-x86_64
xen_scheduler        : credit
xen_pagesize         : 4096
platform_params      : virt_start=0xfffff80000000000
xen_changeset       : Thu Jan 14 15:35:35 2016 +0000 git:6e85
97a-dirty
xen_commandline      : placeholder dom0_mem=4096M,max:4096M cp
uinfo com1=115200,8n1
                               console=com1,tty loglvl=all guest_loglv
l=all
cc_compiler          : gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5
-4)
cc_compile_by        : mockbuild
cc_compile_domain    : centos.org
cc_compile_date      : Wed Jan 20 12:25:53 UTC 2016
xend_config_format  : 4

```

配置桥接网络：

这里以“eno16777736”为例，实际操作中替换为你自己环境的接口名称（IP和网关等也是）。

```
nmcli c add type bridge autoconnect yes con-name br0 iface br0 #  
添加桥接“br0”
```

```
nmcli c modify br0 ipv4.addresses 10.0.0.30/24 ipv4.method manual  
# 给br0设置IP
```

```
nmcli c modify br0 ipv4.gateway 10.0.0.1 # 给br0设置网关
```

```
nmcli c modify br0 ipv4.dns 10.0.0.1 # 给br0设置DNS
```

```
nmcli c delete eno16777736 # 删除eno16777736的当前设置（如果是远程操作会断开，最好是本机操作或双网卡）
```

```
nmcli c add type bridge-slave autoconnect yes con-name eno16777736  
iface eno16777736 master br0 # 添加eno16777736接口作为br0的成员
```

```
systemctl stop NetworkManager; systemctl start NetworkManager # 停  
止并启动NetworkManager
```

```
ip addr
```

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
  group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP>
  mtu 1500 qdisc pfifo_fast master br0 state UP group default
  qlen 1000
    link/ether 00:0c:29:9f:9b:d3 brd ff:ff:ff:ff:ff:ff
3: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
  state UP group default
    link/ether 00:0c:29:9f:9b:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.30/24 brd 10.0.0.255 scope global br0
      valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe9f:9bd3/64 scope link
      valid_lft forever preferred_lft forever

```

2.3.2. 创建虚拟机

安装客户机并创建虚拟机。本例演示安装CentOS7。

本例使用Libvirt，因此首先安装：

```
yum --enablerepo=centos-virt-xen -y install libvirt libvirt-daemon-xen virt-install # 启用CentOS Xen存储库
```

使用“Para-Virtualization”创建虚拟机：

```
mkdir -p /var/xen/images # 为镜像创建目录
```

```

virt-install \
--connect xen:/// \
--paravirt \
--name centos7 \
--ram 4096 \
--disk path=/var/xen/images/centos7.img,size=20 \
--vcpus 2 \
--os-type linux \
--os-variant rhel7 \
--network bridge=br0 \
--graphics none \
--location 'http://ftp.iij.ad.jp/pub/linux/centos/7/os/x86_64/' \
\
--extra-args 'text console=com1 utf8 console=hvc0'

```

Starting install... # 安装开始

安装完成后，客户机的登录提示如下：

```

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

localhost login:

```

可以像KVM的示例一样使用 `virsh` 命令在客户机和主机之间切换，也可以使用 `Xen` 管理工具，如下所示：

```

[root@localhost ~]# # "Ctrl + ]"从客户机转到主机
[root@d1p ~]# # 主机的控制台

```

`xl list # 显示活动域`

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	4090	6	r-----	226.3
centos7	2	4096	2	-b----	17.0

```
[root@dlp ~]# xl console centos7 # 转到客户机  
[root@localhost ~]# # 客户机的控制台
```

2.4. Docker

Docker是操作系统级虚拟化工具，它可以在容器中自动部署应用程序。

2.4.1. 安装Docker

```
yum -y install docker
```

```
systemctl start docker  
systemctl enable docker
```

下载官方镜像并创建一个容器，并在容器中输出“Welcome to the Docker World”：

```
docker pull centos # 下载centos镜像
```

```
Trying to pull repository docker.io/library/centos ...  
latest: Pulling from library/centos  
47d44cb6f252: Extracting      32 B/32 B  
...  
...
```

```
docker run centos /bin/echo "Welcome to the Docker World" # 在容器  
内部运行 echo
```

```
Welcome to the Docker World
```

使用 `i` 和 `t` 选项连接到容器的交互会话，如下所示（如果从容器会话退出，则容器的进程结束）：

```
[root@dlp ~]# docker run -i -t centos /bin/bash
[root@06c8cbea8dc3 /]# # 容器的控制台

bash-4.3# uname -a
Linux 06c8cbea8dc3 3.10.0-123.13.2.el7.x86_64 #1 SMP Thu Dec 18
14:09:13 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
[root@06c8cbea8dc3 /]# exit
exit
[root@dlp ~]# # 回到主机的控制台
```

如果要从容器会话中退出并保持容器的进程，按“Ctrl + p”和“Ctrl + q”：

```
[root@dlp ~]# docker run -i -t centos /bin/bash
[root@64241ed538ed /]# # 按“Ctrl + p”和“Ctrl + q”回到主机的控制台
[root@dlp ~]#
```

```
docker ps # 显示docker进程
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
64241ed538ed	centos:7	"/bin/bash"	35 seconds ago	Up 34 seconds
	clever_bartik			

```
[root@dlp ~]# docker attach 64241ed538ed # 连接到容器的会话
[root@64241ed538ed /]# # 已连接
```

```
docker kill 64241ed538ed # 从主机的控制台关闭容器的进程
```

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			

2.4.2. 添加镜像

为容器添加镜像。

2.4. Docker

例如，使用安装httpd来更新官方镜像，并将其添加为容器的新镜像。每次执行 docker run 命令时都会生成容器，因此要添加最新执行的容器，如下所示：

```
docker images # 显示镜像
```

REPOSITORY	TAG	IMAGE ID	CREA
	VIRTUAL SIZE		
centos	7	8efe422e6104	4 da
ys ago	224 MB		
centos	centos7	8efe422e6104	4 da
ys ago	224 MB		
centos	latest	8efe422e6104	4 da
ys ago	224 MB		

```
docker run centos /bin/bash -c "yum -y update; yum -y install httpd" # 启动一个容器并安装httpd
```

```
docker ps -a | head -2
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS		PORTS NAMES	
a0294a053f8c	centos:7	"/bin/bash -c 'yum -	37 seconds ago
Exited (0) 19 seconds ago		suspicious_morse	

```
docker commit a0294a053f8c my_image/centos_httpd # 添加镜像
```

```
d0938f54bfd62c2a108249c1f969aaeb80be51fbbaee15b594004d4875327609
```

```
docker images # 显示镜像
```

REPOSITORY	TAG	IMAGE ID
CREATED	VIRTUAL SIZE	
my_image/centos_httpd	latest	d0938f54bfd6
17 seconds ago	338.3 MB	
centos	7	8efe422e6104
4 days ago	224 MB	
centos	centos7	8efe422e6104
4 days ago	224 MB	
centos	latest	8efe422e6104
4 days ago	224 MB	

```
docker run my_image/centos_httpd /usr/bin/which httpd # 从新镜像生成容器并执行 which 命令以确认httpd存在
```

```
/usr/sbin/httpd
```

2.4.3. 访问容器

如果想要访问作为守护进程在容器中运行的HTTP或SSH服务，按如下设置：

使用上一节安装httpd的容器为例：

```
docker run -it -p 8081:80 my_image/centos_httpd /bin/bash # 启动容器并连接到shell会话，将主机的端口和容器的端口映射，格式为 -p xxx:xxx
```

```
[root@821bc61cb2e6 /]# /usr/sbin/httpd &
[root@821bc61cb2e6 /]# echo "httpd on Docker Container" > /var/www/html/index.html
[root@821bc61cb2e6 /]# # 按“Ctrl + p”和“Ctrl + q”回到主机的控制台
```

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS		
821bc61cb2e6	my_image/centos_httpd:latest	"/bin/bash"	54 seconds ago
Up 53 seconds	0.0.0.0:8081->80/tcp		

从与容器在同一局域网的客户端通过HTTP访问URL，并确认可以正常访问：



2.4.4. 使用 Dockerfile

使用Dockerfile并自动创建Docker镜像，对于配置管理也很有用。

Dockerfile的格式为： [指令 参数] 。

指令参考以下描述：

2.4. Docker

指令	描述
FROM	为后续指令设置基础镜像
MAINTAINER	设置生成的镜像的作者字段
RUN	将在创建Docker镜像时执行任何命令
CMD	当Docker容器将被执行时执行任何命令
ENTRYPOINT	当Docker容器将被执行时执行任何命令
LABEL	向镜像添加元数据
EXPOSE	通知Docker容器在运行时监听指定的网络端口
ENV	设置环境变量
ADD	复制新文件，目录或远程文件URL
COPY	复制新文件或目录（和“ADD”不同是不能指定远程URL，也不会自动解压文件）
VOLUME	创建具有指定名称的挂载点，并将其标记为从本机主机或其他容器保留外部挂载的卷
USER	设置用户名或UID
WORKDIR	设置工作目录

例如，创建一个Dockerfile以安装httpd并添加index.html，并使用80端口启动httpd：

编辑 Dockerfile 文件：

```
FROM centos
MAINTAINER serverworld <admin@srv.world>
RUN yum -y install httpd
RUN echo "Hello DockerFile" > /var/www/html/index.html
EXPOSE 80
CMD ["-D", "FOREGROUND"]
ENTRYPOINT ["/usr/sbin/httpd"]
```

```
docker build -t web_server:latest . #构建镜像： docker build -t
[image name]:[tag] .
```

```
Sending build context to Docker daemon 10.24 kB
Step 0 : FROM centos
--> 7322fbe74aa5
Step 1 : MAINTAINER serverworld <admin@srv.world>
--> Running in fa5364b3d41f
--> 57d8fd36b7f7
.....
.....
Removing intermediate container 3efa8e1dcae9
Successfully built 7c39aaa338b4
```

docker images

REPOSITORY	TAG	IMAGE ID	CREATED
VIRTUAL SIZE			
web_server	latest	7c39aaa338b4	24 seconds ago
go	283.9 MB		
docker.io/centos	latest	ce20c473cd8a	8 weeks ago
	172.3 MB		

```
docker run -d -p 80:80 web_server #在后台运行容器
```

docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	POR	TS	NAMES
eda2b1482272	web_server	"/usr/sbin/httpd -D F"	35 seconds ago
Up 34 seconds	0.0.0.0:80->80/tcp		mad_bhabha

```
curl http://localhost/
```

Hello DockerFile

2.4.5. 使用Docker-Registry

在要配置为注册服务器的主机上安装Docker-Registry：

```
yum -y install docker-registry
```

2.4. Docker

编辑 `/etc/docker-registry.yml` 文件：

```
# 添加
search_backend: _env:SEARCH_BACKEND:sqlalchemy

# 指定搜索的数据库文件（如果需要可更改）
sqlalchemy_index_database: _env:SQLALCHEMY_INDEX_DATABASE:sqlite
:///tmp/docker-registry.db

# 存储镜像的目录（如果需要可更改）
storage_path: _env:STORAGE_PATH:/var/lib/docker-registry
```

```
mkdir /var/lib/docker-registry # 创建存储镜像的目录
```

```
systemctl start docker-registry
systemctl enable docker-registry
```

```
curl localhost:5000 # 确认能够访问
```

```
"\"docker-registry server\""
```

当从Docker节点使用注册服务器时，Docker服务默认使用HTTPS访问，如果想使用HTTP访问，需要如下更改每个Docker节点上的设置：编辑 `/etc/sysconfig/docker` 文件：

```
# 取消注释并指定Docker-Registry服务器
INSECURE_REGISTRY='--insecure-registry dlp.srv.world:5000'
```

```
systemctl restart docker
```

上面的设置完成后，可以使用注册服务器。以下是推送镜像到注册服务器的情况：

添加标签并推送：

```
docker tag web_server dlp.srv.world:5000/httpd
```

```
docker push dlp.srv.world:5000/httpd
```

```
docker images
```

2.4. Docker

REPOSITORY	VIRTUAL SIZE	TAG	IMAGE ID	CREATED
web_server	282.8 MB	latest	4d62ac763587	About a minute ago
dlp.srv.world:5000/httpd	282.8 MB	latest	4d62ac763587	About a minute ago
docker.io/centos	194.7 MB	latest	14dab3d40372	36 hours ago

以下是从注册服务器镜像的情况：

```
docker images
```

REPOSITORY	TAG	IMAGE ID	CREA
TED	VIRTUAL SIZE		

```
docker search dlp.srv.world:5000/httpd # 使用单词“httpd”搜索注册服务器
```

INDEX	NAME	DESCRIPTION
STARS	OFFICIAL	AUTOMATED
srv.world	dlp.srv.world:5000/library/httpd	0

```
docker pull dlp.srv.world:5000/httpd
```

```
docker images
```

REPOSITORY	VIRTUAL SIZE	TAG	IMAGE ID	CREATED
dlp.srv.world:5000/httpd	282.8 MB	latest	4d62ac763587	14 minute s ago

下面是使用**HTTPS**访问注册服务器的设置，本例演示配置为使用Apache httpd：

先在注册服务器[创建证书](#)

在每个Docker节点上的 `/etc/docker/certs.d` 下创建一个名称为在创建证书时为“Common Name”指定的名称目录，然后将注册服务器上的“`xxx.crt`”传到这个目录。另外，如果创建的自签名证书，还要传“`ca-bundle.crt`”文件。

2.4. Docker

```
ll /etc/docker/certs.d/dlp.srv.world
```

```
total 268
-r--r--r-- 1 root root 266702 Dec 18 11:09 ca-bundle.crt
-rw-r--r-- 1 root root 1334 Dec 18 11:09 server.crt
```

编辑 `/etc/sysconfig/docker` 文件：

```
# 注释下面一行
#INSECURE_REGISTRY='--insecure-registry dlp.srv.world:5000'
```

```
systemctl restart docker
```

在Docker注册服务器上[安装Apache httpd](#)并[配置使用SSL](#)。

在Docker注册服务器上如下配置httpd：

编辑 `/etc/httpd/conf.d/docker-registry.conf` 文件：

```
ProxyRequests off
ProxyPreserveHost on
ProxyPass / http://127.0.0.1:5000/
ProxyPassReverse / http://127.0.0.1:5000/
<Location />
    AuthType Basic
    AuthName "Basic Authentication"
    AuthUserFile /etc/httpd/conf/.htpasswd
    require valid-user
</Location>
```

```
htpasswd -c /etc/httpd/conf/.htpasswd cent
```

```
New password:
Re-type new password:
Adding password for user cent
```

```
systemctl restart httpd
```

完成后，使用HTTPS从任一Docker节点访问注册服务器：

```
docker login dlp.srv.world
```

```
Username: cent # 用htpasswd添加的用户登录
Password:
Email:
WARNING: login credentials saved in /root/.docker/config.json
Login Succeeded
```

```
docker tag web_server dlp.srv.world/webserver
```

```
docker push dlp.srv.world/webserver
```

```
docker search dlp.srv.world/web
```

INDEX	NAME	DESCRIPTION
STARS	OFFICIAL AUTOMATED	
srv.world	dlp.srv.world/library/webserver	0

2.4.6. 持久化存储

当容器被移除时，其中的数据也会丢失，因此如果需要，可在容器中使用外部文件系统作为持久化存储。

例如，使用busybox镜像创建仅用于保存数据的存储服务器的容器：

编辑 Dockerfile 文件：

```
FROM busybox
MAINTAINER ServerWorld <admin@srv.world>

VOLUME /storage
CMD /bin/sh
```

```
docker build -t storage . # 构建镜像
```

```
docker images
```

2.4. Docker

REPOSITORY	TAG	IMAGE ID	CREA
storage	latest	65c5cce81114	20 s
econds ago	1.113 MB		
docker.io/centos	latest	14dab3d40372	6 da
ys ago	194.7 MB		
docker.io/busybox	latest	fc0db02f3072	13 d
ays ago	1.113 MB		

```
docker run -i -t --name storage_server storage # 使用任意名称生成容器
```

```
/ # exit
```

要从其他容器使用上面的容器作为存储服务器，添加一个选项 `--volumes-from` :

```
[root@dlp ~]# docker run -i -t --name centos_server --volumes-from storage_server centos /bin/bash

[root@b9b7a2d35b51 /]# df -hT
Filesystem                      Type   Size  Used Avail
Use% Mounted on
/dev/mapper/docker-253:0-67164897-.... ext4    99G  266M  94G
  1% /
tmpfs                          tmpfs   2.0G    0  2.0G
  0% /dev
shm                           tmpfs   64M    0   64M
  0% /dev/shm
tmpfs                          tmpfs   2.0G    0  2.0G
  0% /sys/fs/cgroup
/dev/mapper/centos-root        xfs     27G  3.2G  24G
  13% /storage
tmpfs                          tmpfs   2.0G    0  2.0G
  0% /run/secrets

[root@b9b7a2d35b51 /]# echo "persistent storage" >> /storage/testfile.txt
[root@b9b7a2d35b51 /]# ll /storage
total 4
-rw-r--r-- 1 root root 19 Dec 22 02:15 testfile.txt
```

确认数据已保存到存储服务器，如下所示：

```
docker start storage_server
```

```
docker attach storage_server
```

```
/ # cat /storage/testfile.txt
persistent storage
```

在外部文件系统中保存数据的其他方法：可以将Docker主机上的目录挂载到容器中：

```
mkdir -p /var/docker/disk01 # 创建一个目录
```

```
echo "persistent storage" >> /var/docker/disk01/testfile.txt
```

2.4. Docker

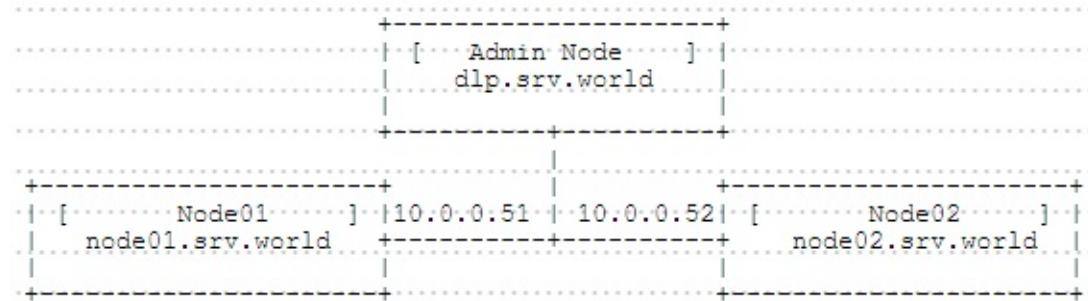
运行一个容器，将上面的目录挂载到 /mnt :

```
[root@www ~]# docker run -i -t -v /var/docker/disk01:/mnt centos  
/bin/bash  
  
[root@bc9a4d5578a6 /]# df -hT  
Filesystem  
Type  
Size  
Used  
Avail  
Use%  
Mounted on  
/dev/mapper/docker-253:0-67164897-..... ext4  
99G  
266M  
94G  
1% /  
tmpfs  
tmpfs  
2.0G  
0  
2.0G  
0% /dev  
shm  
tmpfs  
64M  
0  
64M  
0% /dev/shm  
tmpfs  
tmpfs  
2.0G  
0  
2.0G  
0% /sys/fs/cgroup  
/dev/mapper/centos-root  
xfs  
27G  
3.2G  
24G  
13% /mnt  
tmpfs  
tmpfs  
2.0G  
0  
2.0G  
0% /run/secrets  
  
[root@bc9a4d5578a6 /]# cat /mnt/testfile.txt  
persistent storage
```

2.5. Kubernetes

[Kubernetes](#)是来自Google云平台的开源容器集群管理系统。基于[Docker](#)构建一个容器的调度服务。该系统可以自动在一个容器集群中选择一个工作容器供使用。

本例使用如下环境（一个管理节点和两个容器节点配置Kubernetes集群）：



2.5.1. 配置管理节点

安装所需的软件包：

```
yum -y install kubernetes etcd flannel
```

配置Kubernetes：

```
openssl genrsa -out /etc/kubernetes/service.key 2048 # 生成RSA密钥
```

编辑 `/etc/kubernetes/controller-manager` 文件：

```
# 添加
KUBE_CONTROLLER_MANAGER_ARGS="--service_account_private_key_file
=/etc/kubernetes/service.key"
```

编辑 `/etc/kubernetes/apiserver` 文件：

2.5. Kubernetes

```
# 更改
KUBE_API_ADDRESS="--address=0.0.0.0"

# 更改为管理节点的主机名或IP地址
KUBE_ETCD_SERVERS="--etcd_servers=http://dlp.srv.world:2379"

# Kubernetes服务的IP范围（如需要可更改）
KUBE_SERVICE_ADDRESSES="--service-cluster-ip-range=10.254.0.0/16"
"

# 添加
KUBE_API_ARGS="--service_account_key_file=/etc/kubernetes/service.key"
```

编辑 `/etc/etcd/etcd.conf` 文件：

```
# 取消注释
ETCD_LISTEN_PEER_URLS="http://localhost:2380"

# 添加etcd主机的主机名或IP地址
ETCD_LISTEN_CLIENT_URLS="http://dlp.srv.world:2379,http://localhost:2379"
```

编辑 `/etc/kubernetes/config` 文件：

```
# 更改为管理节点的主机名或IP地址
KUBE_MASTER="--master=http://dlp.srv.world:8080"
```

```
systemctl start etcd kube-apiserver kube-controller-manager kube-scheduler
systemctl enable etcd kube-apiserver kube-controller-manager kube-scheduler
```

配置 Flannel 网络：

编辑 `flannel-config.json` 文件：

```
# 指定想要在容器节点内使用的网络范围
{
    "Network": "172.16.0.0/16",
    "SubnetLen": 24,
    "Backend": {
        "Type": "vxlan",
        "VNI": 1
    }
}
```

编辑 `/etc/sysconfig/flanneld` 文件：

```
# # 更改为Flannel主机的主机名或IP地址
FLANNEL_ETCD="http://dlp.srv.world:2379"

# 确认参数
FLANNEL_ETCD_KEY="/atomic.io/network"
```

```
etcdctl set atomic.io/network/config < flannel-config.json
```

```
systemctl start flanneld
systemctl enable flanneld
```

确认设置。如果显示以下结果，表示正常：

```
kubectl cluster-info
```

```
Kubernetes master is running at http://localhost:8080
```

2.5.2. 配置容器节点

在所有节点安装并运行Docker服务。

在所有节点上安装Kubernetes和Flannel：

```
yum -y install kubernetes flannel
```

在所有节点上配置Kubernetes，如下所示：

2.5. Kubernetes

编辑 `/etc/kubernetes/config` 文件：

```
# 更改为管理节点的主机名或IP地址  
KUBE_MASTER="--master=http://dlp.srv.world:8080"
```

编辑 `/etc/kubernetes/kubelet` 文件：

```
# 更改  
KUBELET_ADDRESS="--address=0.0.0.0"  
  
# 更改为本机主机名  
KUBELET_HOSTNAME="--hostname_override=node01"  
  
# 更改为管理节点的主机名或IP地址  
KUBELET_API_SERVER="--api_servers=http://dlp.srv.world:8080"
```

编辑 `/etc/sysconfig/flanneld` 文件：

```
# 更改为管理节点的主机名或IP地址  
FLANNEL_ETCD="http://dlp.srv.world:2379"
```

```
nmcli c down docker0 # 停止“docker0”接口
```

```
Connection 'docker0' successfully deactivated  
(D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/1)
```

```
systemctl start flanneld kube-proxy kubelet  
systemctl enable flanneld kube-proxy kubelet  
systemctl restart docker
```

确认设置。如果每个节点的状态如下，表示正常（在管理节点上操作）：

```
kubectl get nodes
```

NAME	LABELS	STATUS
node01	kubernetes.io/hostname=node01	Ready
node02	kubernetes.io/hostname=node02	Ready

2.5.3. 创建Pod

Kubernetes集群中的容器作为Pod进行管理，可以运行容器来创建Pod。

例如，使用安装了httpd的单个容器创建一个Pod：

在节点上[创建一个安装了httpd的容器的镜像](#)

本来镜像名称“web_server”用于配置。

将上面的容器镜像复制到所有其他节点，如下所示（在**node01**中操作）：

```
docker save web_server > web_server.tar # 将容器镜像输出到文件
```

```
scp web_server.tar node02:/root/web_server.tar # 将镜像复制到其他节点
```

加载刚复制的容器镜像（在**node02**中操作）：

```
docker load < web_server.tar
```

```
docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
web_server	latest	084ef53f3d83	11 minutes ago
		283.9 MB	
docker.io/centos	latest	ce20c473cd8a	8 weeks ago
		172.3 MB	

在管理节点上创建一个Pod：

编辑 `pod-webserver.yaml` 文件：

```
apiVersion: v1
kind: Pod
metadata:
  # Pod的名称
  name: httpd
  labels:
    # Pod的标签
    app: web_server
spec:
  containers:
    # 容器的名称
    - name: httpd
      # 容器镜像
      image: web_server
      ports:
        # 容器端口
        - containerPort: 80
```

```
kubectl create -f pod-webserver.yaml # 创建Pod
```

pods/httpd

```
kubectl get pods -o wide # 显示Pod列表
```

NAME	READY	STATUS	RESTARTS	AGE	NODE
httpd	0/1	Running	0	8s	node01

```
kubectl get pod httpd -o yaml | grep "podIP" # 显示在Pod上分配的IP地址
```

podIP: 172.16.35.10

```
curl http://172.16.35.10/ # 访问Pod
```

Hello DockerFile

如果要删除一个Pod，按照以下步骤操作：

```
kubectl delete pod httpd
```

```
pod/httpd
```

2.5.4. 持久化存储

如果要使用持久化数据，则需要使用外部存储。例如，通过挂载外部存储器创建一个Pod，该存储器与用于在节点Pod上保存数据的目录进行映射。

本例使用上一节的环境和容器镜像。

在管理节点上创建一个Pod：

编辑 `pod-webserver.yaml` 文件：

```
apiVersion: v1
kind: Pod
metadata:
  name: httpd
spec:
  containers:
    - name: httpd
      image: web_server
      ports:
        - containerPort: 80
      volumeMounts:
        # 要使用的卷（这是在“volumes”部分中定义的卷）
        - name: httpd-storage
          # 容器内的挂载点
          mountPath: /var/www/html
  volumes:
    # 任意名称
    - name: httpd-storage
      hostPath:
        # 主机节点上保存数据的目录
        path: /var/docker/disk01
```

```
kubectl create -f pod-webserver.yaml
```

```
pods/httpd
```

```
kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	NODE
httpd	0/1	Running	0	8s	node01

```
kubectl get pod httpd -o yaml | grep "podIP"
```

```
podIP: 172.16.89.7
```

转到Pod正在运行的节点，并确认正常工作（在**node01**中操作）：

```
echo "Persistent Storage" > /var/docker/disk01/index.html
```

```
curl http://172.16.89.7/
```

```
Persistent Storage
```

3. 桌面环境

桌面环境对服务器不是必需的，但有时安装或使用应用程序需要桌面环境。

- 3.1. GNOME桌面
- 3.2. KDE桌面
- 3.3. Xrdp服务器
- 3.4. VNC服务器
 - 3.4.1. VNC服务器
 - 3.4.2. VNC客户端
 - 3.4.2.1. Windows客户端
 - 3.4.2.2. noVNC
 - 3.4.2.3. Guacamole
- 3.5. RDP连接到Windows

3.1. GNOME桌面

```
yum -y groups install "GNOME Desktop" # 安装GNOME桌面环境
```

```
startx # 安装后运行
```

GNOME桌面环境启动。第一次引导，初始设置运行如下。

选择系统语言：

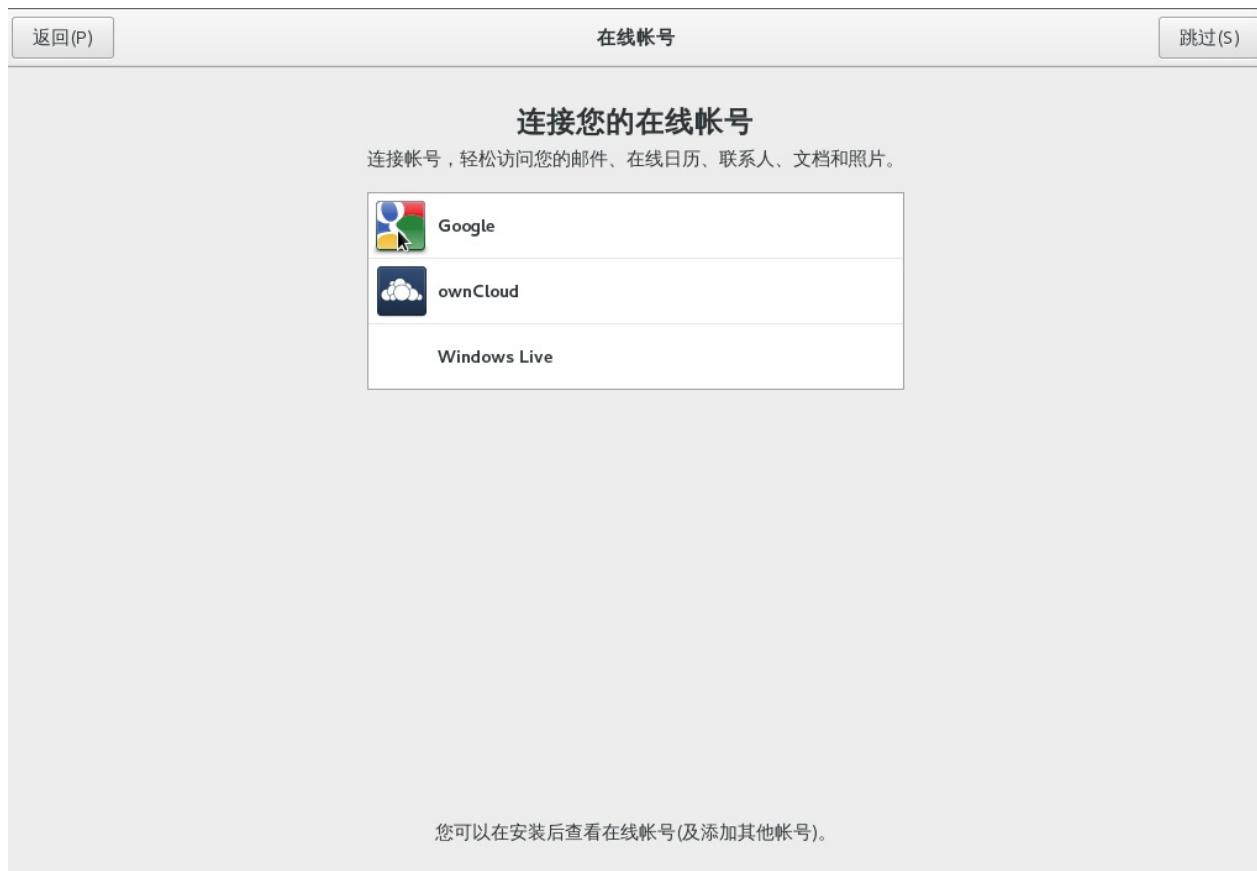


选择键盘类型：

3.1. GNOME桌面

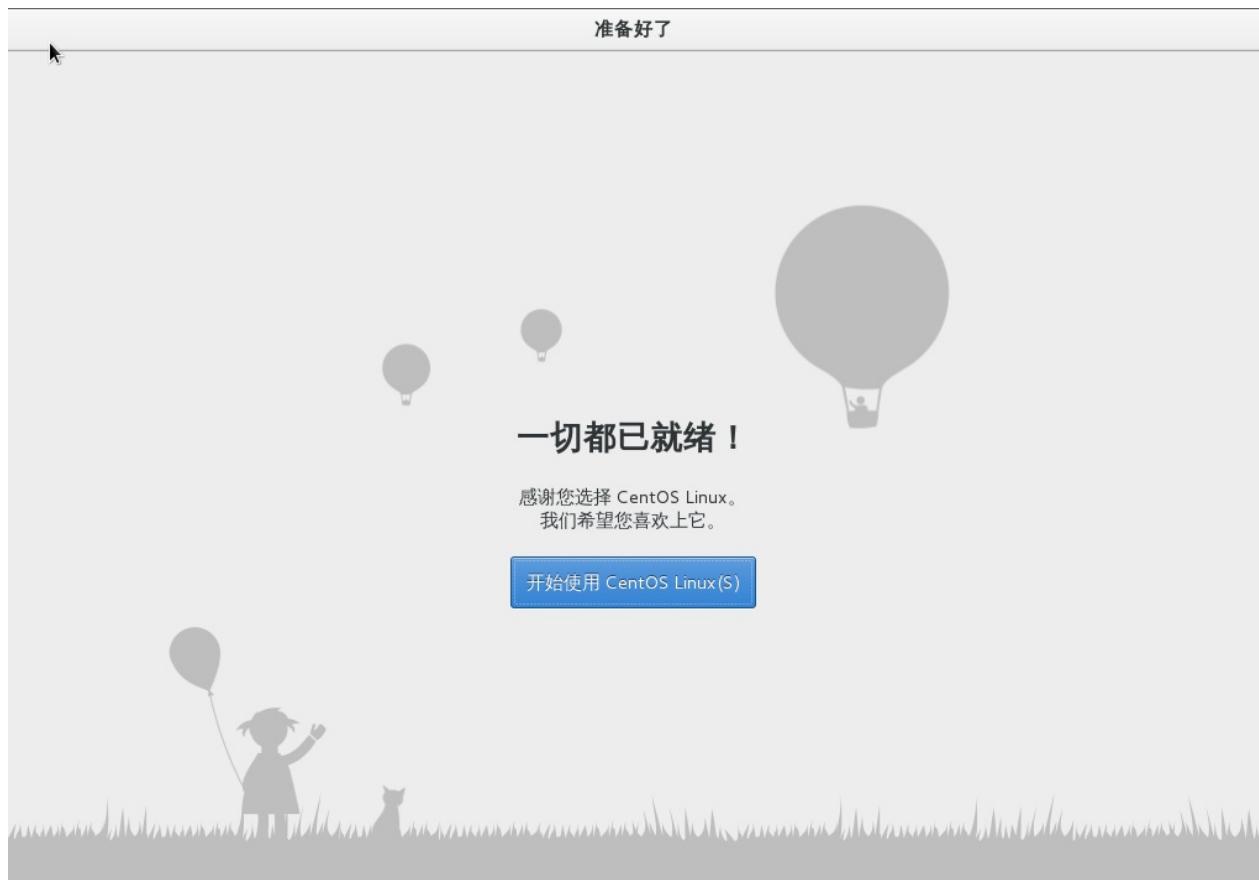


是否需要设置在线帐号：

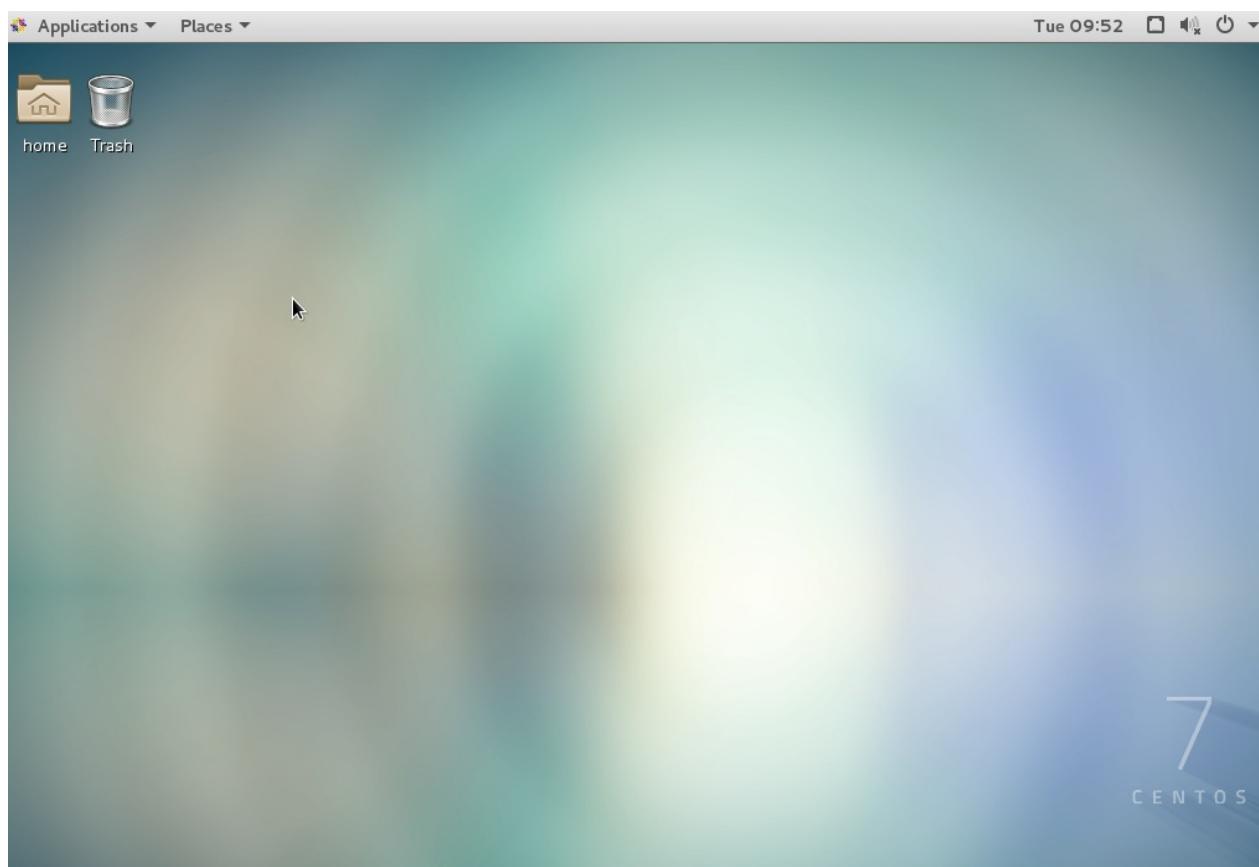


配置完成后，单击“开始使用CentOS Linux”：

3.1. GNOME桌面



GNOME桌面环境启动后：



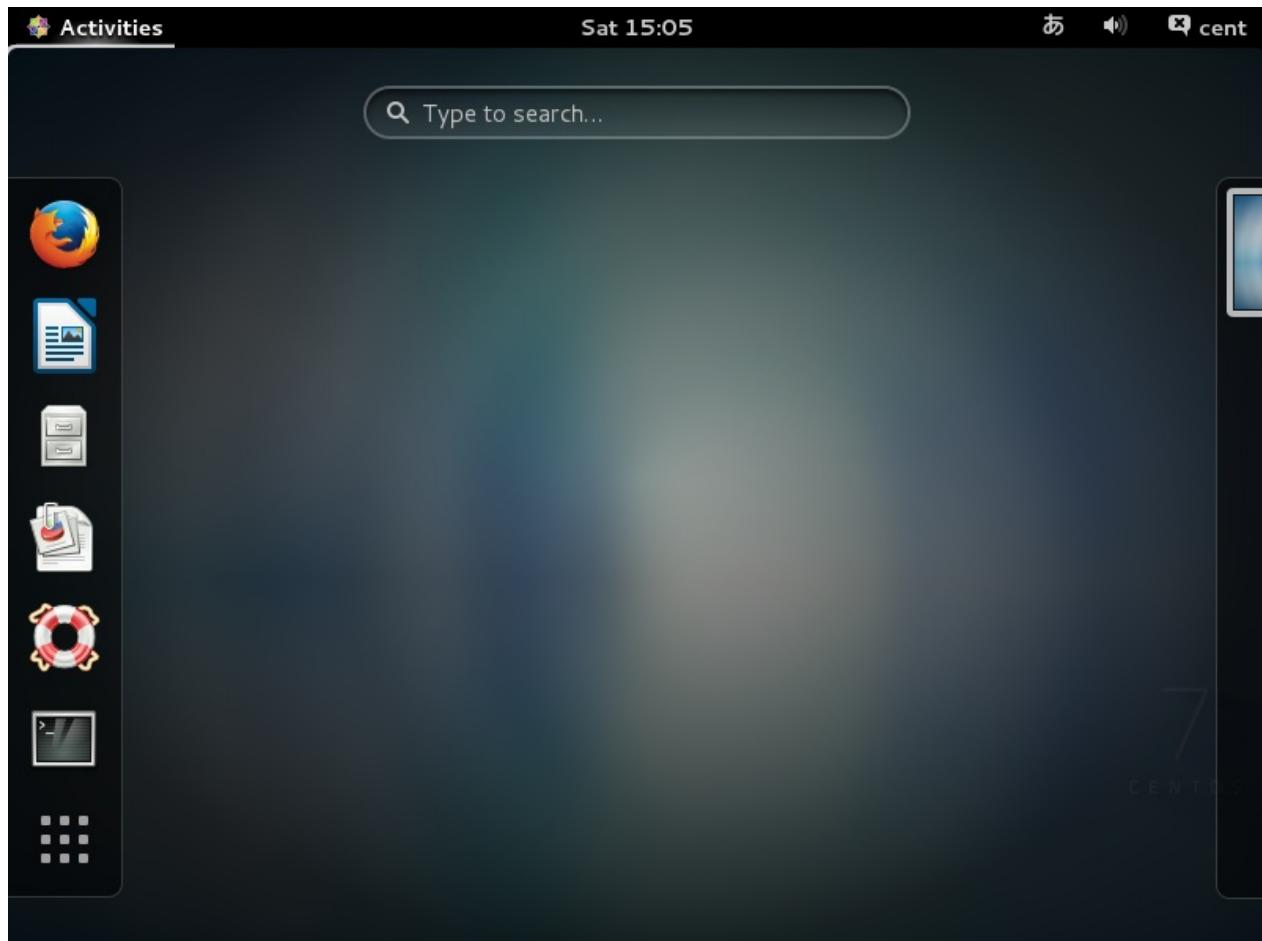
3.1. GNOME桌面

CentOS7的GNOME桌面默认以经典模式启动，如果要使用**GNOME Shell**，按以下设置：

```
echo "exec gnome-session" >> ~/.xinitrc
```

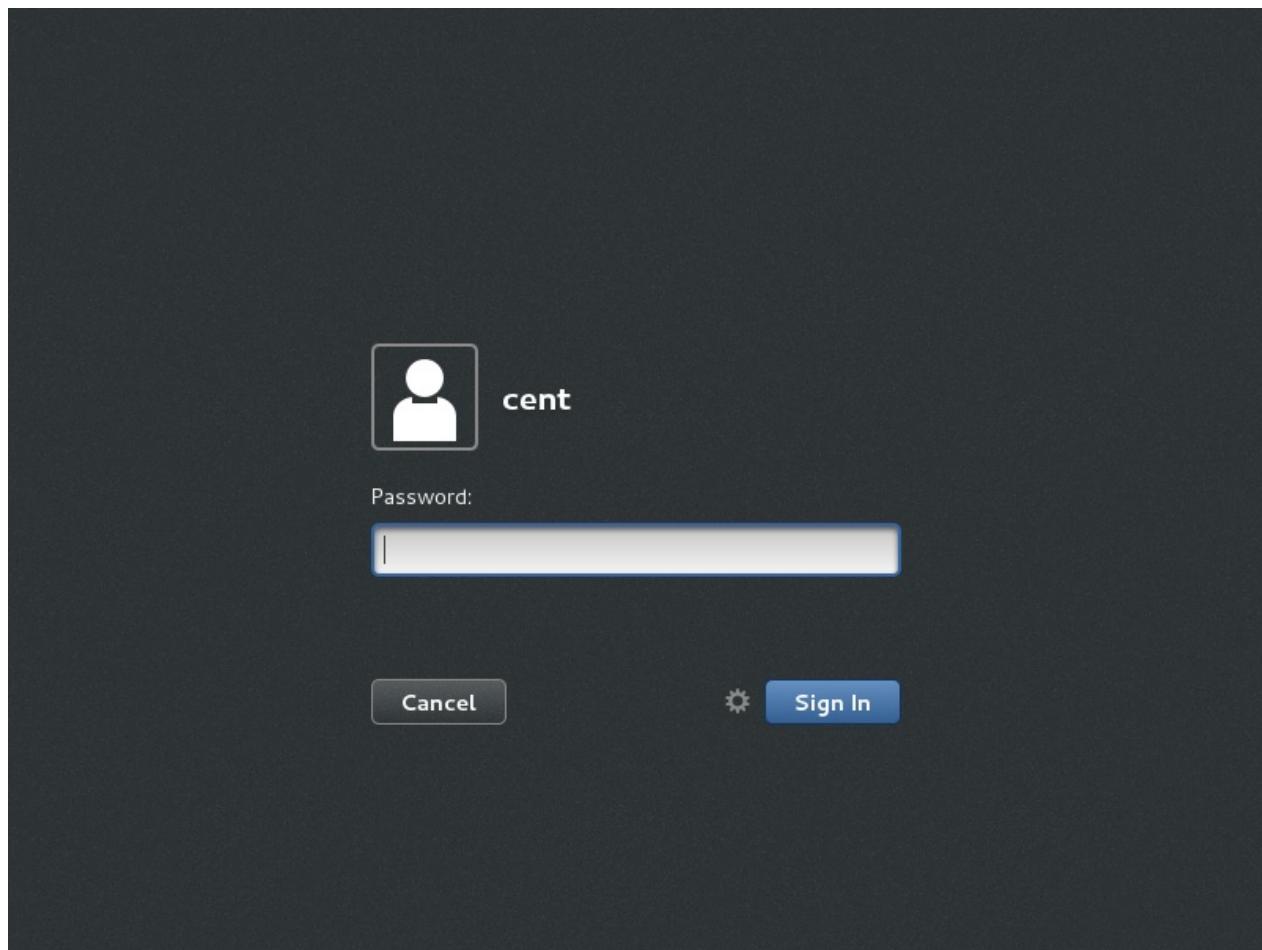
```
startx
```

GNOME Shell启动：



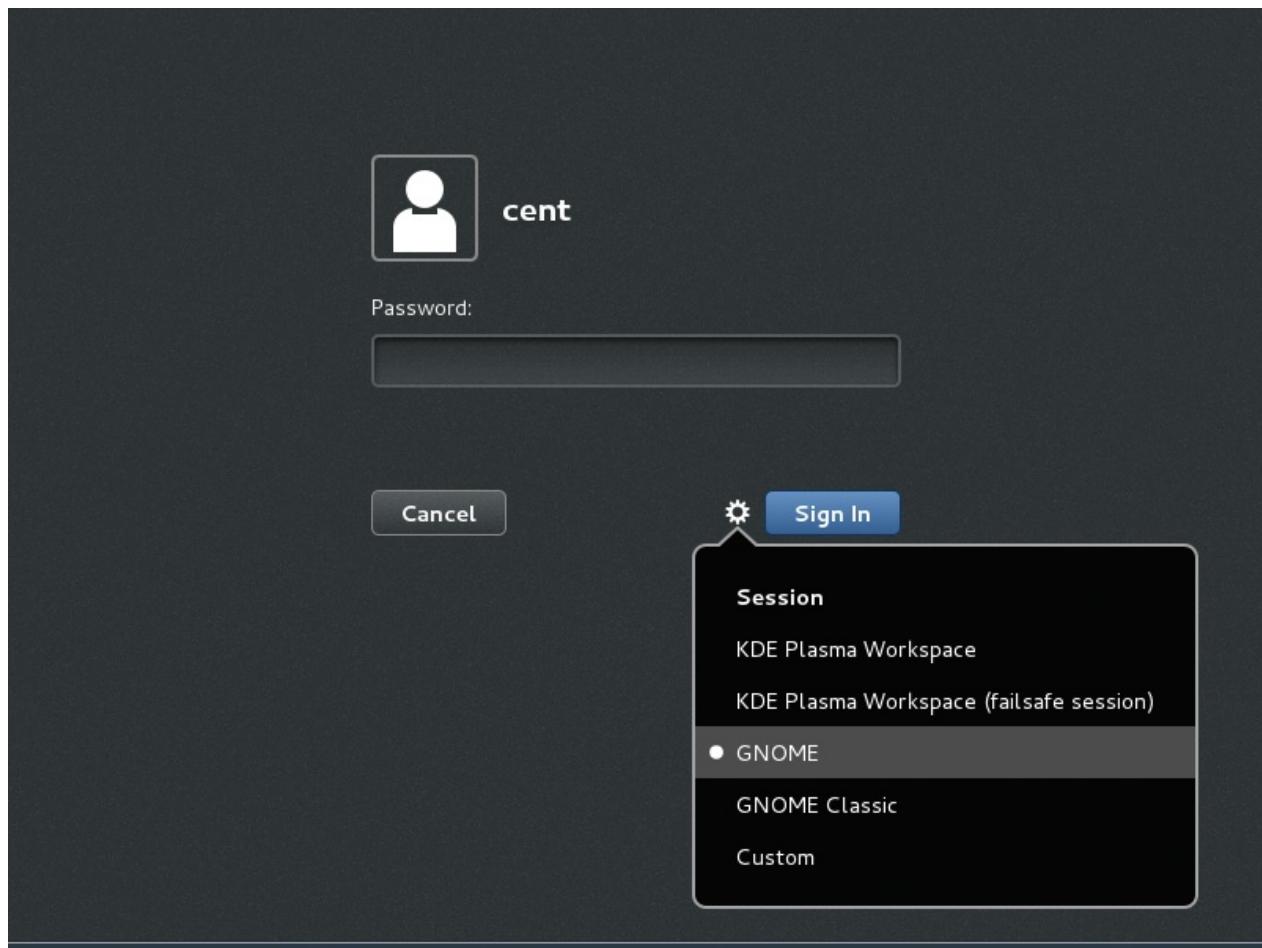
如果设置了系统图形登录，也可以切换到GNOME Shell。单击位于“Sign In”按钮旁边的按钮：

3.1. GNOME桌面



在列表中选择“GNOME”（默认为GNOME Classic）：

3.1. GNOME桌面



登录GNOME Shell：

3.1. GNOME 桌面



一些补充：

切换默认以命令行还是桌面启动：

查看 `/etc/inittab`

```
# inittab is no longer used when using systemd.  
#  
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.  
#  
# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target  
#  
# systemd uses 'targets' instead of runlevels. By default, there  
# are two main targets:  
#  
# multi-user.target: analogous to runlevel 3  
# graphical.target: analogous to runlevel 5  
#  
# To view current default target, run:  
# systemctl get-default  
#  
# To set a default target, run:  
# systemctl set-default TARGET.target  
#
```

有两种启动方式：

```
# multi-user.target: analogous to runlevel 3 # 命令行模式  
# graphical.target: analogous to runlevel 5 # 图形模式
```

运行 `systemctl get-default` 查看当前默认：

```
multi-user.target
```

运行 `systemctl set-default graphical.target` 切换至图形模式启动。

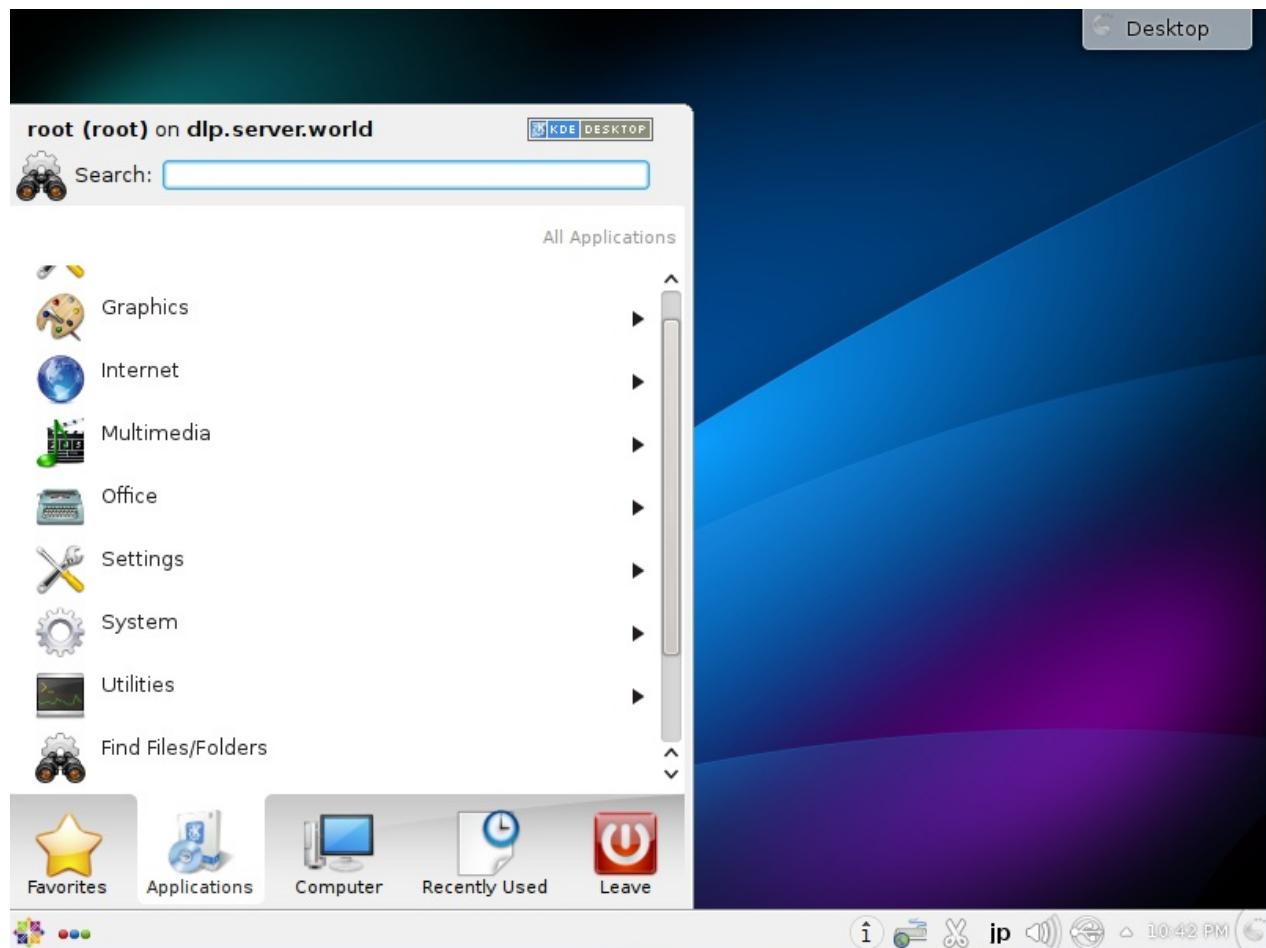
重启检查是否生效。

运行 `systemctl set-default multi-user.target` 切换回命令行模式启动。

3.2. KDE桌面

```
yum -y groups install "KDE Plasma Workspaces" # 安装KDE桌面环境  
echo "exec startkde" >> ~/.xinitrc # 安装完成后运行  
startx # 启动桌面
```

KDE桌面环境启动：



3.3. Xrdp服务器

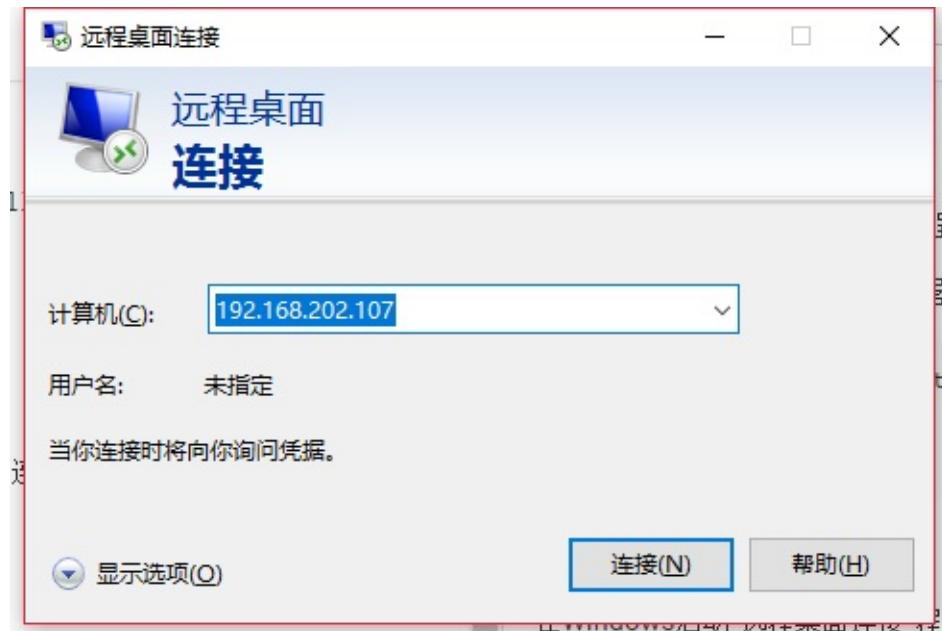
安装Xrdp服务器以从Windows远程桌面功能连接到CentOS。

(从EPEL) 安装并启动Xrdp服务器：

```
yum --enablerepo=epel -y install xrdp  
systemctl start xrdp  
systemctl enable xrdp
```

打开防火墙端口3389/TCP。

在Windows启动“远程桌面连接”程序，“计算机”后面输入CentOS的IP地址：

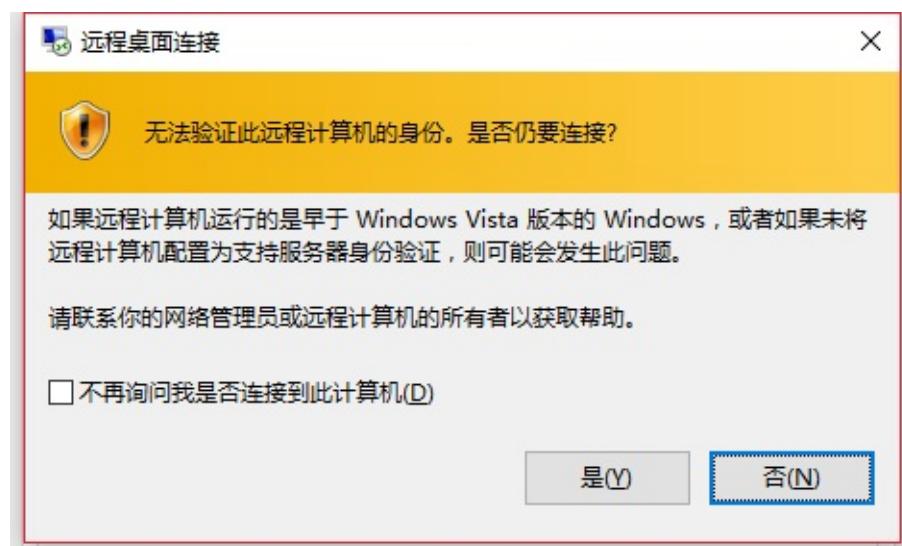


点击左下角“显示选项”，上面标签栏从“常规”改到“显示”并将下面“颜色”改为“真彩色(24位)”（默认为32位，但Xrdp目前最高只支持24位，可以选择比24位低的）：

3.3. Xrdp服务器

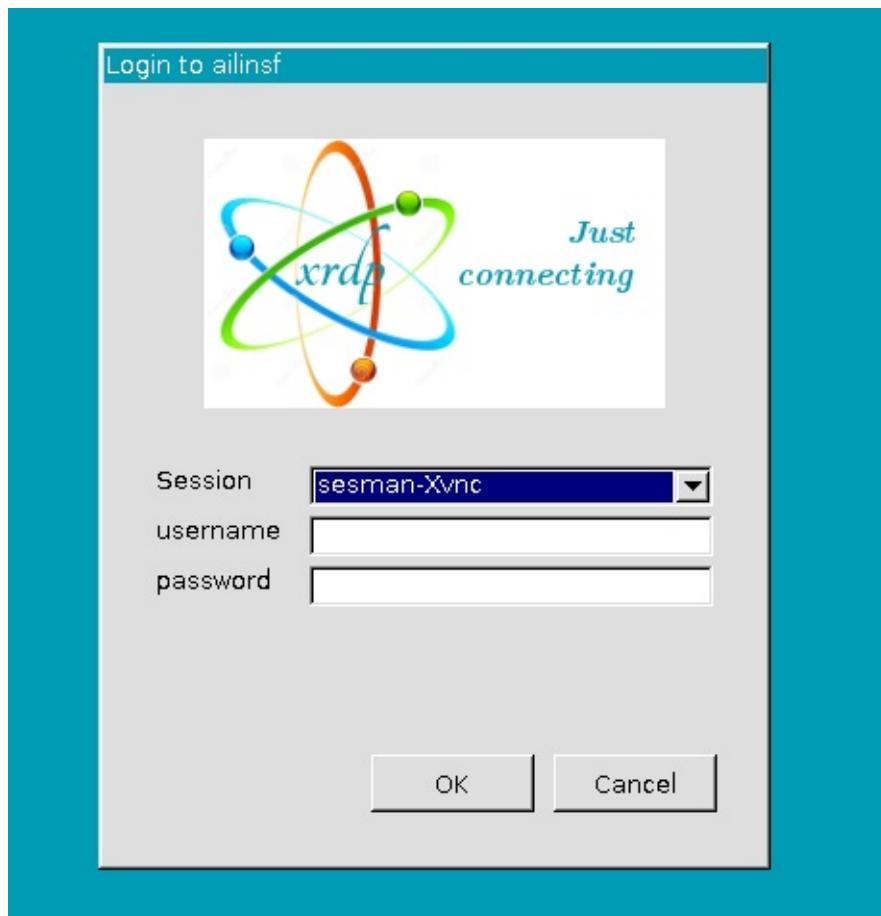


选择好“颜色”后，点击“连接”，出现确认信息，可以选择“不再询问”然后点“是”：



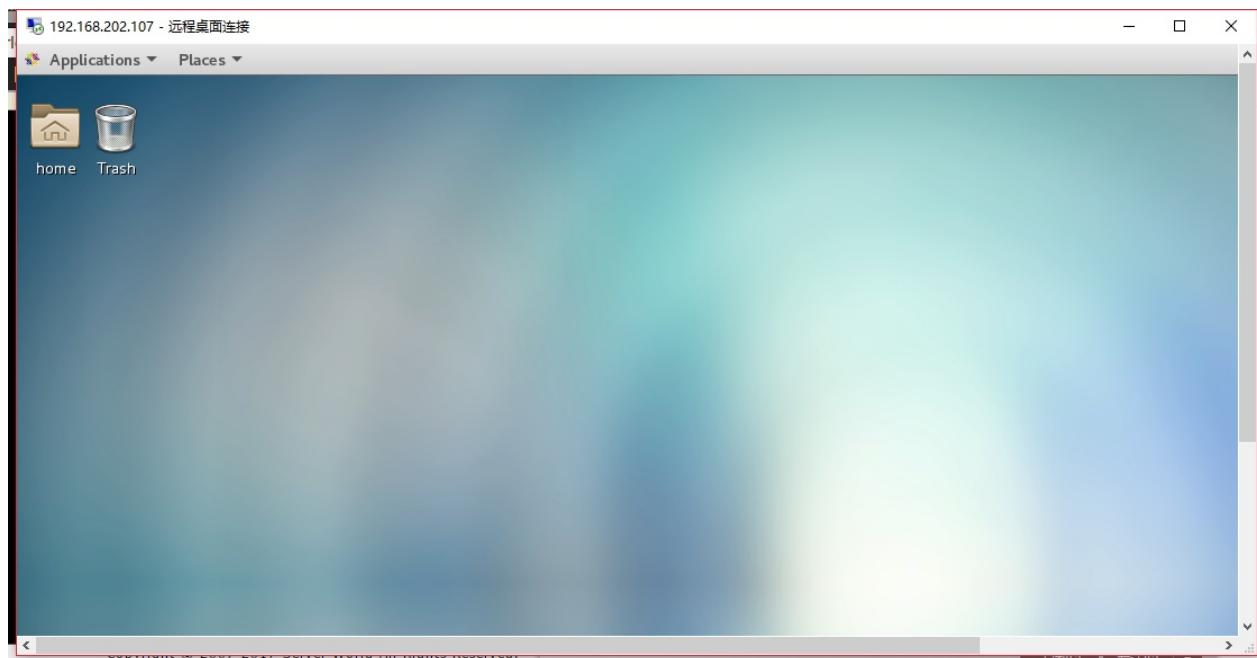
连接成功后出现登录界面，输入CentOS系统的用户名密码：

3.3. Xrdp服务器



如果首次登录失败（有时在首次登录时发生），重试再次登录。

进入桌面后：



3.4. VNC服务器

3.4.1. VNC服务器

安装VNC服务器以从远程客户端连接GUI。此示例基于[GNOME桌面环境](#)。

```
yum -y install tigervnc-server # 安装TigerVNC服务器
```

```
su - cent # 切换到要配置VNC的用户
```

```
vncpasswd # 设置VNC密码
```

```
Password: # 输入密码
```

```
Verify: # 确认密码
```

```
vncserver :1 -geometry 800x600 -depth 24 # 运行时显示编号'1'，屏幕分辨率'800x600'，颜色深度'24'
```

打开防火墙端口5901/TCP（对应上面的数字“1”）。

更多配置：

拷贝一份配置文件示例：

```
cp /lib/systemd/system/vncserver@.service  
/etc/systemd/system/vncserver@:1.service
```

编辑 `/etc/systemd/system/vncserver@:1.service` 文件，找到对应的内容进行修改：

```
User=<USER>
```

```
PIDFile=/home/<USER>/.vnc/%H%i.pid
```

改为

```
#新版本这里有变化，总之就是把<USER>替换为对应的用户名  
User=cent # cent为使用的用户  
  
PIDFile=/home/cent/.vnc/%H%i.pid  
  
# 如果使用root则修改对应  
User=root  
  
PIDFile=/root/.vnc/%H%i.pid
```

重启systemd：

```
systemctl daemon-reload
```

启动并添加开机启动：

```
systemctl start vncserver@:1.service  
systemctl enable vncserver@:1.service
```

注：非root用户时，如果用上面的方法启动报错，可以尝试（在对应用户非root权限下）运行 `vncserver :1 -geometry 800x600 -depth 24`（对应的关闭命令 `vncserver -kill :1`）。在可以启动服务后再使用 `systemctl` 尝试。修改密码时，先将用户目录下 `passwd` 文件删除后，使用 `vncserver :1` 来重新生成。

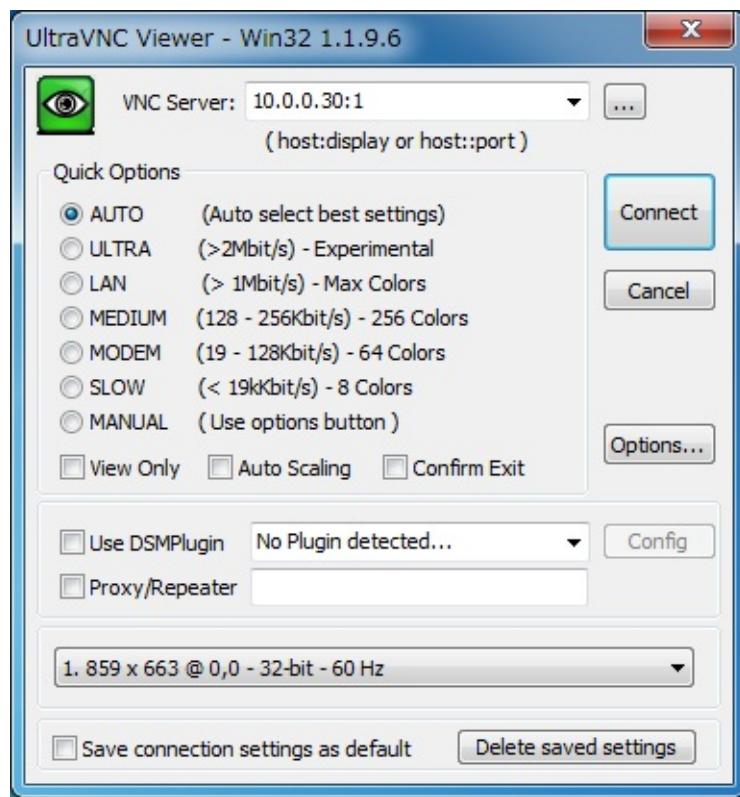
3.4.2. VNC客户端

3.4.2.1. Windows客户端

在客户机上安装VNC viewer，如[UltraVNC](#)。

安装完成后，运行“UltraVNC Viewer”，输入 IP地址:显示编号（10.0.0.30:1）：

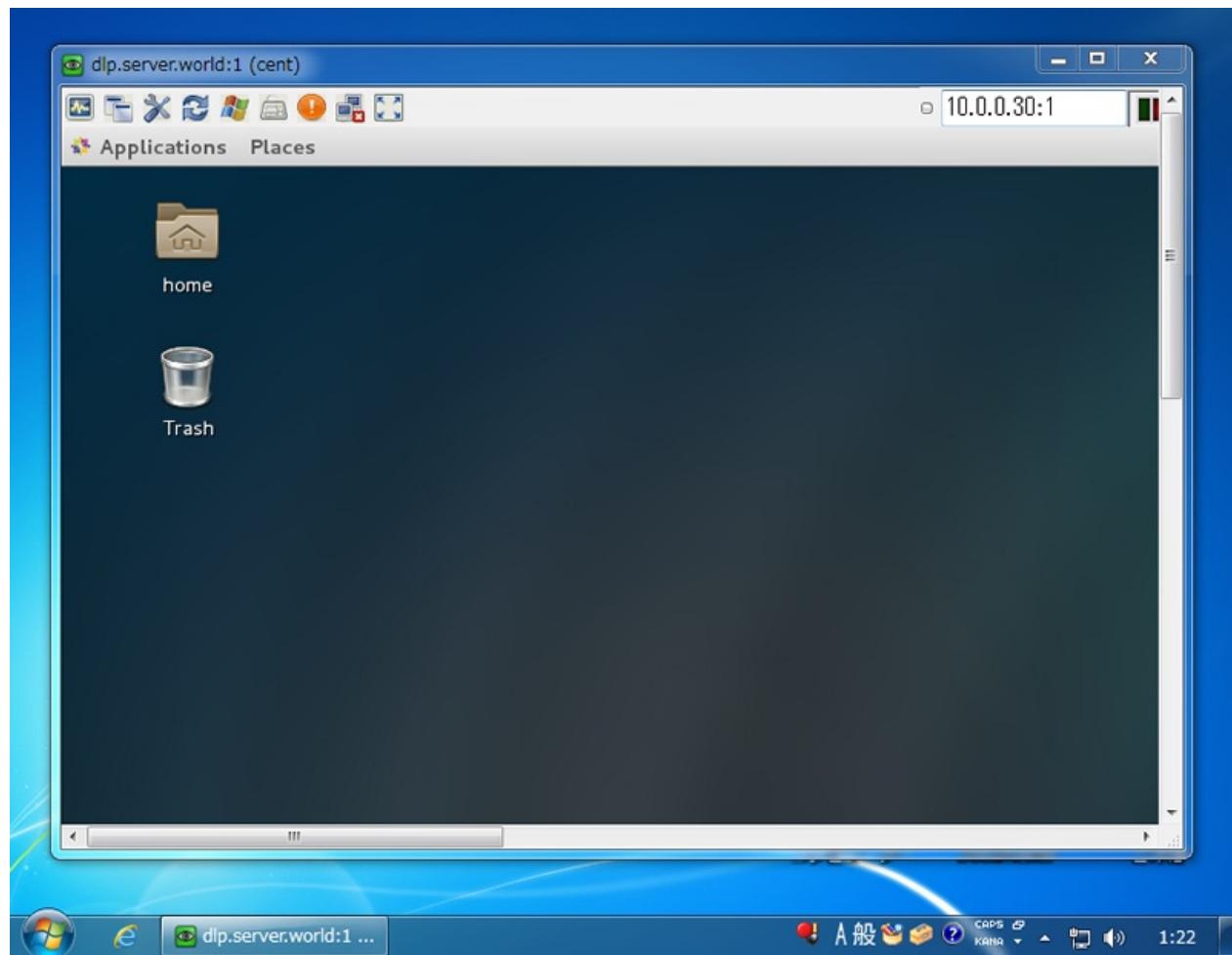
3.4. VNC服务器



输入在服务端设置的密码：



连接成功：



3.4.2.2. noVNC

noVNC是通过Web浏览器连接到VNC服务器的VNC客户端。

```
yum --enablerepo=epel -y install novnc python-websockify numpy # 从EPEL安装
```

```
cd /etc/pki/tls/certs
```

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout /etc/pki/tls/certs/novnc.pem -out /etc/pki/tls/certs/novnc.pem -days 365
```

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/pki/tls/certs/novnc.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN # 国家
State or Province Name (full name) [Some-State]:SC # 省
Locality Name (eg, city) []:CD # 城市
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Server World # 公司
Organizational Unit Name (eg, section) []:IT Solution # 部门
Common Name (eg, YOUR name) []:dlp.srv.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
```

运行VNC服务器（见[VNC服务器章节](#)）。

在运行VNC服务器并在端口“6080”上代理 `localhost:5901` 的用户启动 Websockify：

```
websockify -D --web=/usr/share/novnc/ --
cert=/etc/pki/tls/certs/novnc.pem 6080 localhost:5901
```

```
WebSocket server settings:
- Listen on :6080
- Flash security policy server
- Web server. Web root: /usr/share/novnc
- SSL/TLS support
- Backgrounding (daemon)
```

在防火墙打开端口6080/TCP。

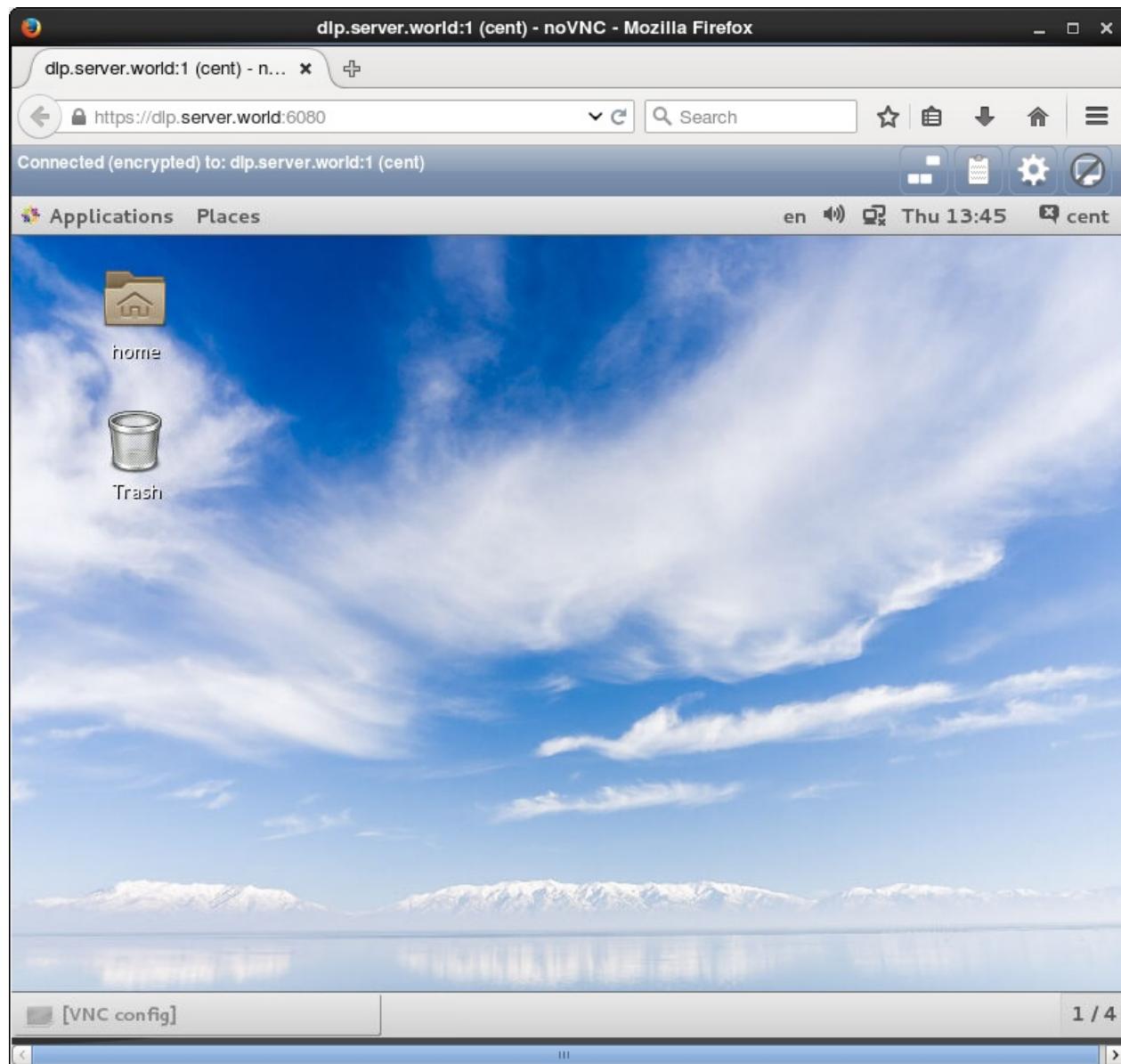
3.4. VNC服务器

在客户端浏览器上访问 `http(s)://服务器IP地址或域名:6080/`，使用VNC密码登录：



登录后可以在浏览器中操作CentOS：

3.4. VNC服务器



上面的方式每个端口（6080/tcp）只能访问一个vnc服务端，如果有多个服务端需要访问，这样的方法不是很方便，需如下操作。

创建 `vnc_tokens` 文件，如 `/root/vnc_tokens`，内容如下：

```
# 以"token: host:port"的格式输入实际需要连接的vnc服务端
host1: 192.168.0.100:5901
host2: 192.168.0.110:5901
```

运行：

```
websockify -D --web=/usr/share/novnc/ --cert=/etc/pki/tls/certs/
novnc.pem --ssl-only --target-config /root/vnc_tokens 6080
```

3.4. VNC服务器

--ssl-only 指仅允许通过 https:// 方式访问

--target-config 指定配置文件路径（可以是文件，也可以是包含配置文件的目录）

如果使用单独的证书文件和密钥文件，则 --cert= 指定证书文件， --key= 指定密钥文件

更多内容可以运行 websockify --help 查看。

运行后在浏览器打开地址： https://主机名:6080/vnc_auto.html?

path=websockify/?token=host1 连接host1（连接host2，则修改为 token=host2 即可）

Apache配置反向代理示例：

代理 https://主机名:6080/：

```

<VirtualHost 80>
ServerAdmin webmaster@localhost
ServerName vnc.x.com
RewriteEngine On
RewriteCond %{HTTPS} !=On
RewriteRule (.*)
https:// %{SERVER_NAME} %{REQUEST_URI} [R=301,L]
</VirtualHost>

<VirtualHost 443>
ServerName vnc.x.com
ServerAdmin webmaster@localhost
ErrorLog logs/vnc-ssl_error_log
TransferLog logs/vnc-ssl_access_log
LogLevel warn
SSLEngine On

# 关闭证书检查保护(如果不启用，连接会报错)
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
SSLProxyCheckPeerExpire off

SSLProtocol -All +TLSv1 +TLSv1.1 +TLSv1.2
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
SSLProxyEngine on
ProxyRequests Off

RewriteEngine On
RewriteCond %{HTTP:Upgrade} =websocket
RewriteRule /(.*)
wss://localhost:6080/$1 [P,L]
RewriteCond %{HTTP:Upgrade} !=websocket
RewriteRule /(.*)
https://localhost:6080/$1 [P,L]

<Proxy *>
Require all granted
</Proxy>
ProxyPass / https://localhost:6080/
ProxyPassReverse / https://localhost:6080/
</VirtualHost>

```

配置完成后使用 `http://vnc.x.com` 或 `https://vnc.x.com` 访问

可以将 `/usr/share/novnc/index.html` 备份，然后替换自己的网页，在网页上将不同主机对应的链接放上去方便进入，在此不再赘述。

3.4.2.3. Guacamole

[Guacamole](#)是一个基于HTML 5和JavaScript的VNC查看器（支持VNC，RDP，SSH等），服务端基于Java的VNC-to-XML代理开发。要求浏览器支持HTML 5。

安装教程<http://guacamole.apache.org/doc/gug/installing-guacamole.html>。

参照[官方文档](#)安装配置后，可以正常使用，不过感觉内容较多配置比较复杂（安装算比较简单），如果想使用轻量的还是noVNC吧。

简单记录下安装过程，更多内容和介绍在文档中查看。

3.4.2.3.1. 安装Guacamole

构建guacamole-server

必须的依赖：

```
yum -y install cairo-devel libjpeg-turbo-devel libjpeg-devel  
libpng-devel uuid-devel
```

可选的依赖（具体对应关系查看[官方文档](#)，不安装则无法使用对应的功能）：

```
yum -y install ffmpeg-devel freerdp-devel pango-devel libssh2-devel  
libtelnet-devel libvncserver-devel pulseaudio-libs-devel openssl-devel  
libvorbis-devel libwebp-devel
```

注意：上面的安装可能有找不到的软件包，在[这里](#)添加软件仓库或查看安装方法。

安装编译环境：

```
yum -y groupinstall "Development tools"
```

下载 `guacamole-server`，下载前可以[查看最新版本](#)。

```
wget  
http://mirrors.shu.edu.cn/apache/guacamole/0.9.14/source/guacamole-server-0.9.14.tar.gz
```

解压安装：

```
tar -xzf guacamole-server-0.9.14.tar.gz  
cd guacamole-server-0.9.14  
.configure --with-init-dir=/etc/init.d  
make  
make install  
ldconfig
```

部署Guacamole

先参考[这里](#)安装JDK和Tomcat，假设Tomcat路径为 /usr/tomcat9。

```
cd ~  
wget http://mirrors.shu.edu.cn/apache/guacamole/0.9.14/binary/guacamole-0.9.14.war  
cp guacamole-0.9.14.war /usr/tomcat9/webapps/guacamole.war  
systemctl restart tomcat9  
/etc/init.d/guacd start # 也可以用service命令来控制(如service guacd start)  
chkconfig --add guacd # 设置guacd服务开机启动
```

启动完成后，在浏览器打开 <http://主机名:8080/guacamole>。

3.4.2.3.2. 配置反向代理

修改Tomcat配置，编辑 /usr/tomcat9/conf/server.xml 文件：

```
<Connector port="8080" protocol="HTTP/1.1"  
connectionTimeout="20000"  
URIEncoding="UTF-8" # 增加此行  
redirectPort="8443" />
```

重启Tomcat。

反向代理设置可以参考Apache的[反向代理设置](#)和[WebSocket代理](#)。

下面给一个示例：

```
<VirtualHost 80>
ServerAdmin webmaster@localhost
ServerName guacamole.x.com
RewriteEngine On
RewriteCond %{HTTPS} !=On
RewriteRule (.*)
https:// %{SERVER_NAME} %{REQUEST_URI} [R=301,L]
</VirtualHost>

<VirtualHost 443>
ServerName guacamole.x.com
ServerAdmin webmaster@localhost
ErrorLog logs/guacamole-ssl_error_log
TransferLog logs/guacamole-ssl_access_log
LogLevel warn
SSLEngine On
SSLProtocol -All +TLSv1 +TLSv1.1 +TLSv1.2
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
SSLProxyEngine on
ProxyRequests Off
<Proxy *>
Require all granted
</Proxy>
ProxyPass / http://localhost:8080/guacamole/ flushpackets=on
ProxyPassReverse / http://localhost:8080/guacamole/
ProxyPass /websocket-tunnel ws://localhost:8080/guacamole/websoc
ket-tunnel
ProxyPassReverse /websocket-tunnel ws://localhost:8080/guacamole
/websocket-tunnel
</VirtualHost>
```

完成后通过 `http://guacamole.x.com/` 访问。

官方的介绍如下：

```
<Location /guacamole/>
    Order allow,deny
    Allow from all
    ProxyPass http://HOSTNAME:8080/guacamole/ flushpackets=on # "flushpackets=on"不要忘记添加
    ProxyPassReverse http://HOSTNAME:8080/guacamole/
</Location>

<Location /guacamole/websocket-tunnel> # "/guacamole/websocket-tunnel"的Location需要在"/guacamole/"后面
    Order allow,deny
    Allow from all
    ProxyPass ws://HOSTNAME:8080/guacamole/websocket-tunnel
    ProxyPassReverse ws://HOSTNAME:8080/guacamole/websocket-tunnel
</Location>
```

通过类似 `http://guacamole.x.com/guacamole/` 的链接来访问。

如果需要更改路径：

```
<Location /new-path/>
    Order allow,deny
    Allow from all
    ProxyPass http://HOSTNAME:8080/guacamole/ flushpackets=on
    ProxyPassReverse http://HOSTNAME:8080/guacamole/
    ProxyPassReverseCookiePath /guacamole/ /new-path/
</Location>

<Location /new-path/websocket-tunnel>
    Order allow,deny
    Allow from all
    ProxyPass ws://HOSTNAME:8080/guacamole/websocket-tunnel
    ProxyPassReverse ws://HOSTNAME:8080/guacamole/websocket-tunnel
</Location>
```

通过类似 `http://guacamole.x.com/new-path/` 的链接来访问。

3.4.2.3.3. 配置Guacamole

GUACAMOLE_HOME

GUACAMOLE_HOME 是Guacamole配置目录（默认为 /etc/guacamole ）的名称，由以下可选文件组成：

- `guacamole.properties``：Guacamole的主要配置文件。这个文件中的属性决定了Guacamole如何连接到guacd，并且可以配置安装扩展认证。
- `logback.xml`：使用称为Logback的日志记录系统来记录所有消息。默认Guacamole只会记录控制台日志，可以通过提供自己的Logback配置文件来改变。
- `extensions/`：所有Guacamole扩展的安装位置。Guacamole将在启动时自动加载该目录下的所有 `.jar` 文件。
- `lib/`：Guacamole扩展所需的库的搜索目录。Guacamole使这个目录内的 `.jar` 文件对所有扩展可用。如果扩展需要额外的库，如数据库驱动程序，应当放在这里。

如果不能或不想使用 `/etc/guacamole` 作为 GUACAMOLE_HOME ，位置可通过以下任一方法来覆盖：

- 在运行servlet容器的用户的主目录中创建一个名为 `.guacamole`，如果存在这个目录，将自动用作 GUACAMOLE_HOME
- 使用环境变量指定 GUACAMOLE_HOME 备用目录的完整路径，务必查阅servlet容器的文档，以确定如何正确设置环境变量
- 使用系统属性 `guacamole.home` 指定备用目录的完整路径

guacamole.properties :

3.5. RDP连接到Windows

安装[FreeRDP](#)来使用RDP（Remote Desktop Protocol 远程桌面协议）连接到Windows计算机。

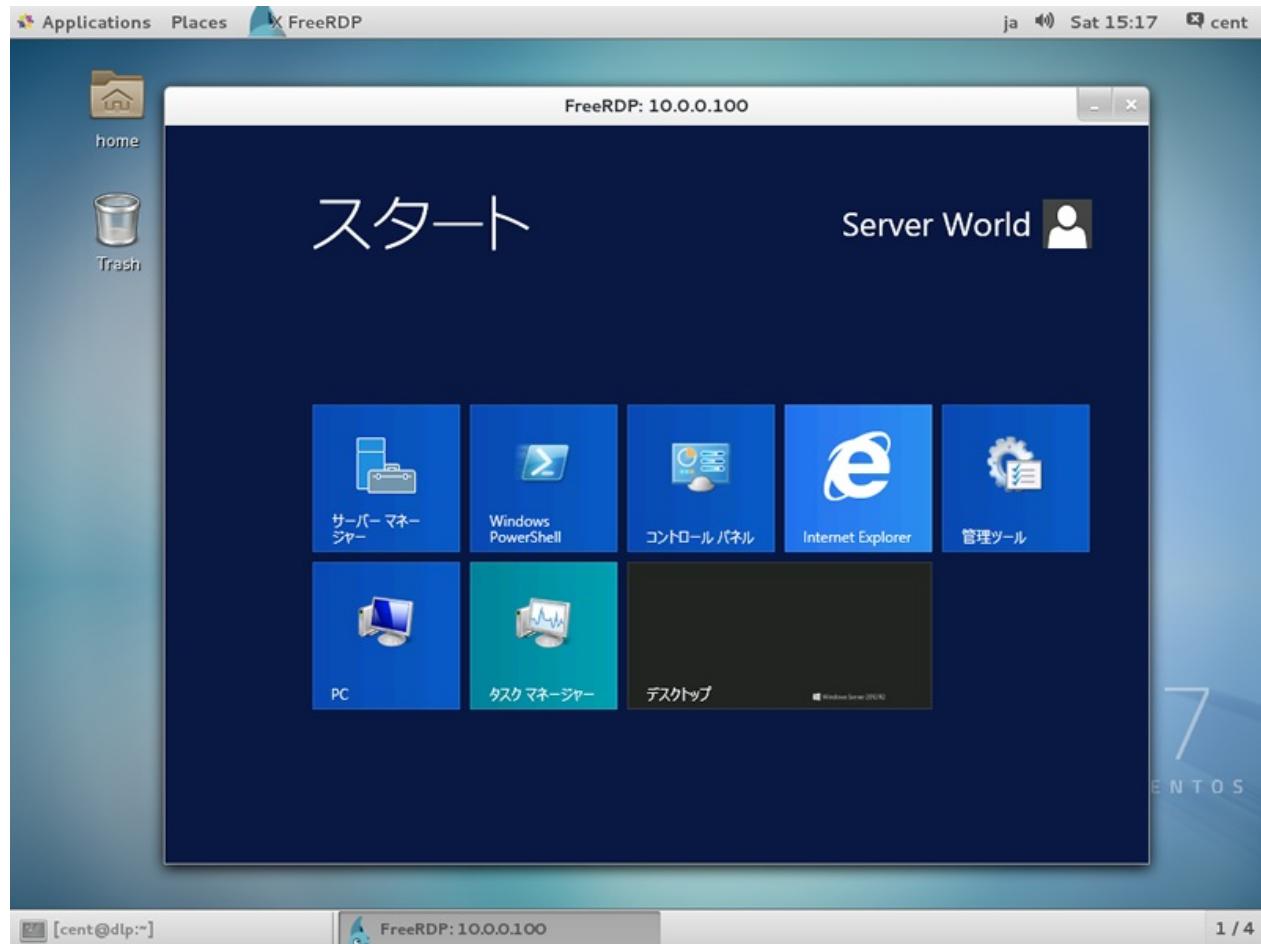
```
yum -y install freerdp
```

在桌面环境的终端下运行：

```
xfreerdp -g 800x600 -u Windows用户名 10.0.0.100 # 连接到Windows，如果RDP端口非默认，则使用 IP:端口 格式：
```

```
connected to 10.0.0.100:3389  
Password: # Windows用户对应的密码
```

连接成功：



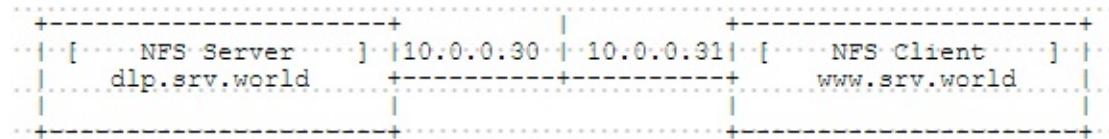
4. 存储服务器

- 4.1. NFS服务器
 - 4.1.1. 配置NFS服务器
 - 4.1.2. 配置NFS客户端
 - 4.1.2.1. CentOS客户端
 - 4.1.2.2. Windows客户端
- 4.2. iSCSI
 - 4.2.1. 配置iSCSI目标
 - 4.2.2. 配置iSCSI启动器
 - 4.2.2.1. CentOS
 - 4.2.2.2. Windows
- 4.3. Ceph
 - 4.3.1. 配置Ceph集群
 - 4.3.2. 客户端
 - 4.3.2.1. 用作块设备
 - 4.3.2.2. 用作文件系统
- 4.4. GlusterFS
 - 4.4.1. 安装GlusterFS
 - 4.4.2. 配置存储集群
 - 4.4.2.1. 分布式配置
 - 4.4.2.2. 复制配置
 - 4.4.2.3. 条带卷配置
 - 4.4.2.4. 分布式 + 复制
 - 4.4.2.5. 条带 + 复制
 - 4.4.3. 客户端设置

4.1. NFS服务器

配置NFS服务器以共享目录到网络。

此示例基于以下环境：



4.1.1. 配置NFS服务器

```
yum -y install nfs-utils
```

编辑 `/etc/idmapd.conf` 文件，将 `Domain =` 一行取消注释并修改为自己的域名 `Domain = srv.world`。

编辑 `/etc/exports` 文件，写入NFS输出设置 `/home
10.0.0.0/24(rw,no_root_squash)`。

```
systemctl start rpcbind nfs-server
systemctl enable rpcbind nfs-server
```

`firewall`防火墙设置：

```
firewall-cmd --add-service=nfs --permanent
firewall-cmd --reload
```

输出设置的基本选项（这里基本只是翻译过来的，具体用法可网上查下资料，有空了可能会再整理下）：

选项	描述
<code>rw</code>	在NFS卷上同时允许读取和写入请求。
<code>ro</code>	在NFS卷上只允许读取请求。
<code>sync</code>	只有在更改已提交到稳定存储后才会对请求进行回复。 (默认)

<code>async</code>	此选项允许NFS服务器违反NFS协议并在该请求所做的任何更改已提交到稳定存储器之前对请求进行回复。
<code>secure</code>	此选项要求请求源自小于 <code>IPPORT_RESERVED</code> (1024) 的Internet端口。 (默认)
<code>insecure</code>	此选项接受所有端口。
<code>wdelay</code>	如果其怀疑另一个相关的写请求可能正在进行或可能很快到达，则延迟向磁盘稍微提交写入请求。 (默认)
<code>no_wdelay</code>	如果同时设置了 <code>async</code> ，此选项不起作用。如果NFS服务器怀疑另一个相关的写请求可能正在进行或可能很快到达，则NFS服务器通常会将写请求提交到磁盘。这允许多个写请求提交到磁盘，其中一个操作可以提高性能。如果NFS服务器主要收到小的无关的请求，这种行为实际上可能会降低性能，因此 <code>no_wdelay</code> 可用来关闭它。
<code>subtree_check</code>	此选项启用子树检查。 (默认)
<code>no_subtree_check</code>	此选项禁用子树检查，这稍微有安全影响，但可以提高某些情况下的可靠性。
<code>root_squash</code>	将请求从uid/gid 0映射到匿名uid/gid。注意，这不适用于可能同样敏感的任何其他uid或gid，如bin用户或staff组。在登入NFS主机使用分享之目录的使用者如果是root时，那么这个使用者的权限将被压缩成为匿名使用者，通常他的UID与GID都会变成nobody那个系统账号的身份。
<code>no_root_squash</code>	关闭root squashing。此选项主要适用于无磁盘客户端。登入NFS主机使用分享目录的使用者，如果是root的话，那么对于这个分享的目录来说，他就具有root的权限！这个项目“极不安全”，不建议使用。
<code>all_squash</code>	Map all uids and gids to the anonymous user. Useful for NFS exported public FTP directories, news spool directories, etc.
<code>no_all_squash</code>	Turn off all squashing. (Default)
<code>anonuid=UID</code>	These options explicitly set the uid and gid of the anonymous account. This option is primarily useful for PC/NFS clients, where you might want all requests appear to be from one user. As an example, consider the export entry for /home/joe in the example section below, which maps all requests to uid 150.
<code>anongid=GID</code>	Read above (<code>anonuid=UID</code>)

4.1.2. 配置NFS客户端

4.1.2.1. CentOS客户端

```
yum -y install nfs-utils
```

编辑 `/etc/idmapd.conf` 文件，

编辑 `/etc/idmapd.conf` 文件，将 `Domain =` 一行取消注释并修改为自己的域名 `Domain = srv.world`。

```
systemctl start rpcbind
systemctl enable rpcbind
```

```
mount -t nfs dlp.srv.world:/home /home # 挂载NFS
```

```
df -hT # 查看
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/mapper/centos-root	xfs	46G	1.4G	45G	4%	/
devtmpfs	devtmpfs	1.9G	0	1.9G	0%	/dev
tmpfs	tmpfs	1.9G	0	1.9G	0%	/dev/shm
tmpfs	tmpfs	1.9G	8.3M	1.9G	1%	/run
tmpfs	tmpfs	1.9G	0	1.9G	0%	/sys/fs/cgroup
/dev/vda1	xfs	497M	219M	278M	45%	/boot
dlp.srv.world:/home	nfs4	46G	1.4G	45G	4%	/home

在`fstab`上配置NFS挂载以在系统引导时挂载：编辑 `/etc/fstab` 文件，在下面加入一行 `dlp.srv.world:/home /home nfs defaults 0 0`。

配置auto-mounting。例如，设置NFS目录在`/mntdir`上：

```
yum -y install autofs
```

编辑 `/etc/auto.master` 文件，在最后加入一行 `/ - /etc/auto.mount`。

编辑 `/etc/auto.mount` 文件，新建 `/mntdir -fstype=nfs,rw dlp.srv.world:/home`（格式为：`[挂载点] [选项] [位置]`）。

```
mkdir /mntdir # 创建目录
```

```
systemctl start autofs  
systemctl enable autofs
```

```
cd /mntdir # 到挂载点，以确保其正常挂载
```

```
ll
```

```
total 0  
drwx----- 2 cent cent 59 Jul 9 2014 cent
```

```
cat /proc/mounts | grep mntdir
```

```
/etc/auto.mount /mntdir autofs rw,relatime,fd=18,pgrp=2093,timeo  
ut=300,minproto=5,maxproto=5,direct 0 0  
dlp.srv.world:/home /mntdir nfs4 rw,relatime,vers=4.0,rsize=5242  
88,wsize=524288,namlen=255,hard,proto=tcp,  
port=0,timeo=600,retrans=2,sec=sys,clientaddr=10.0.0.31,local_lo  
ck=none,addr=10.0.0.30 0 0
```

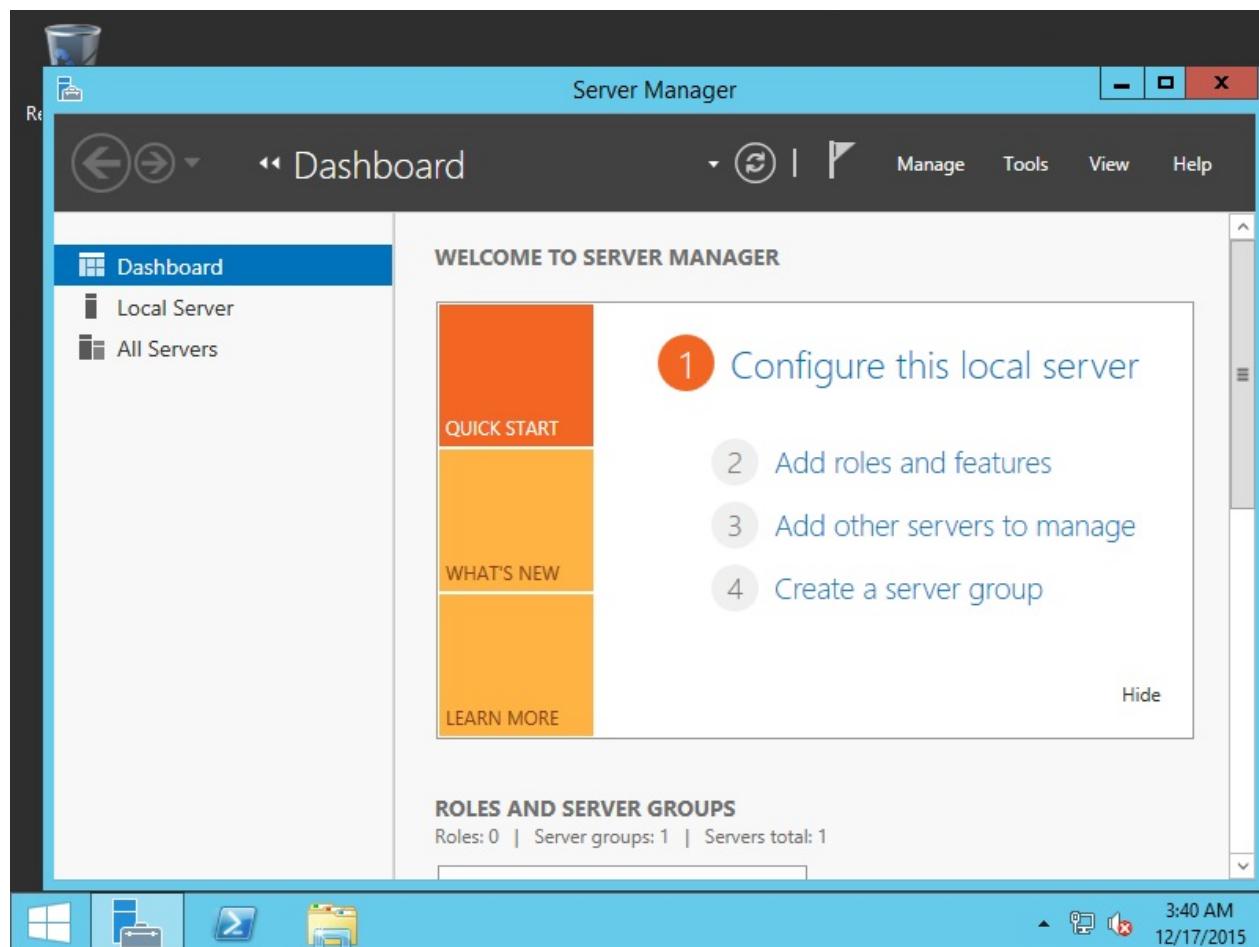
4.1.2.2. Windows客户端

4.1.2.2.1. Windows Server系统

以Windows Server 2012 R2为例。

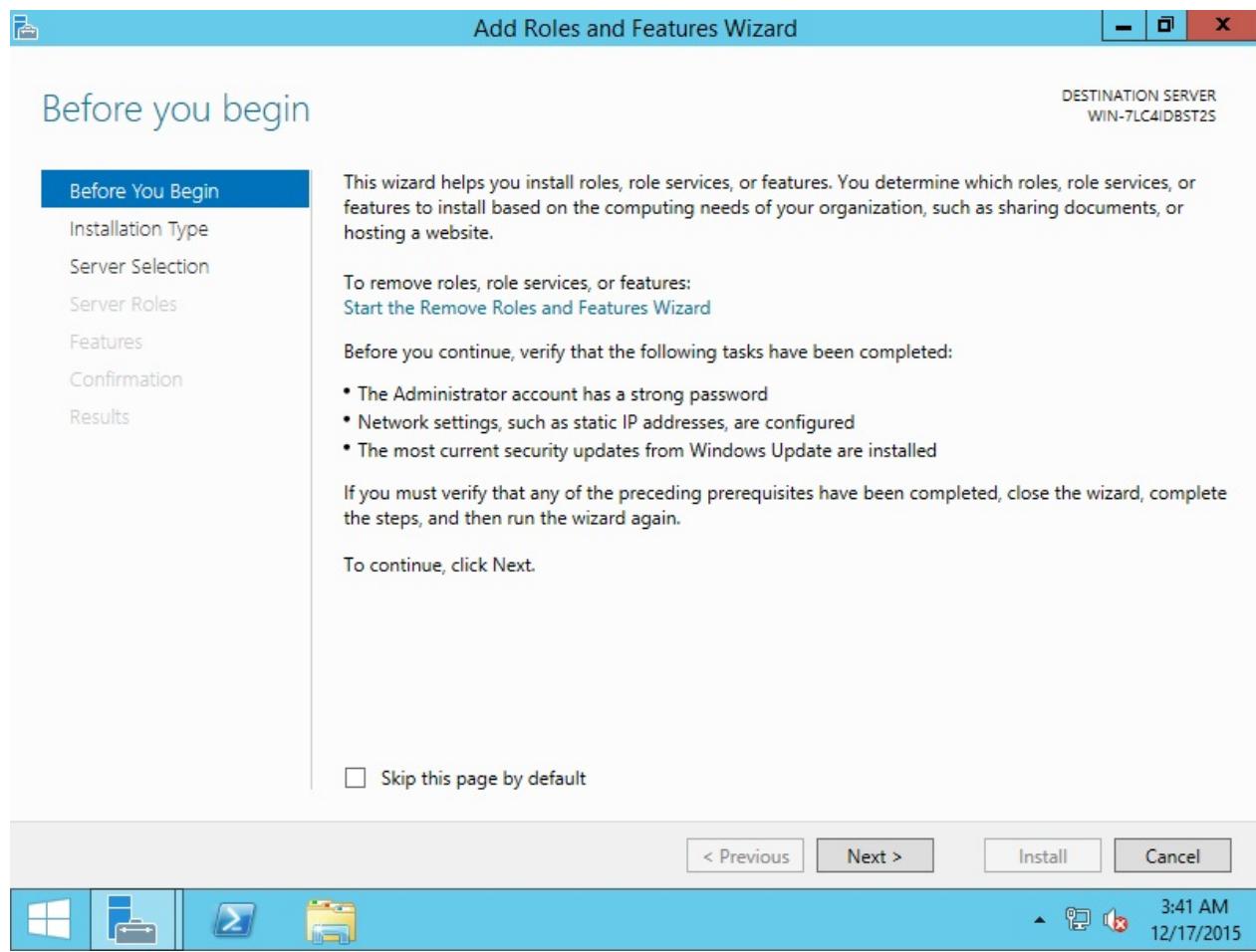
打开“Server Manager”（服务器管理器），单击“Add Roles and Features”（添加角色和功能）：

4.1. NFS服务器



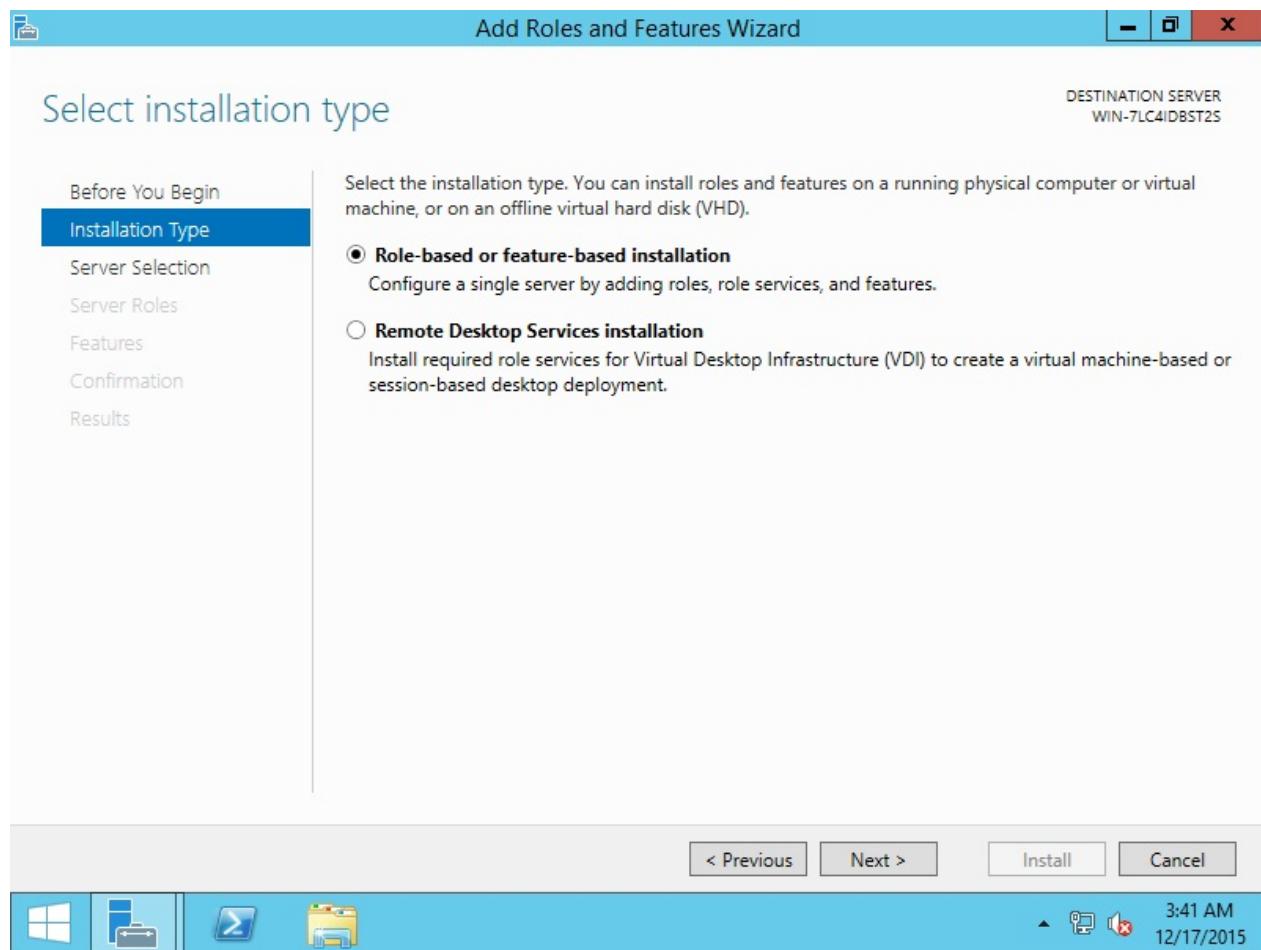
单击“Next”（下一步）继续：

4.1. NFS服务器



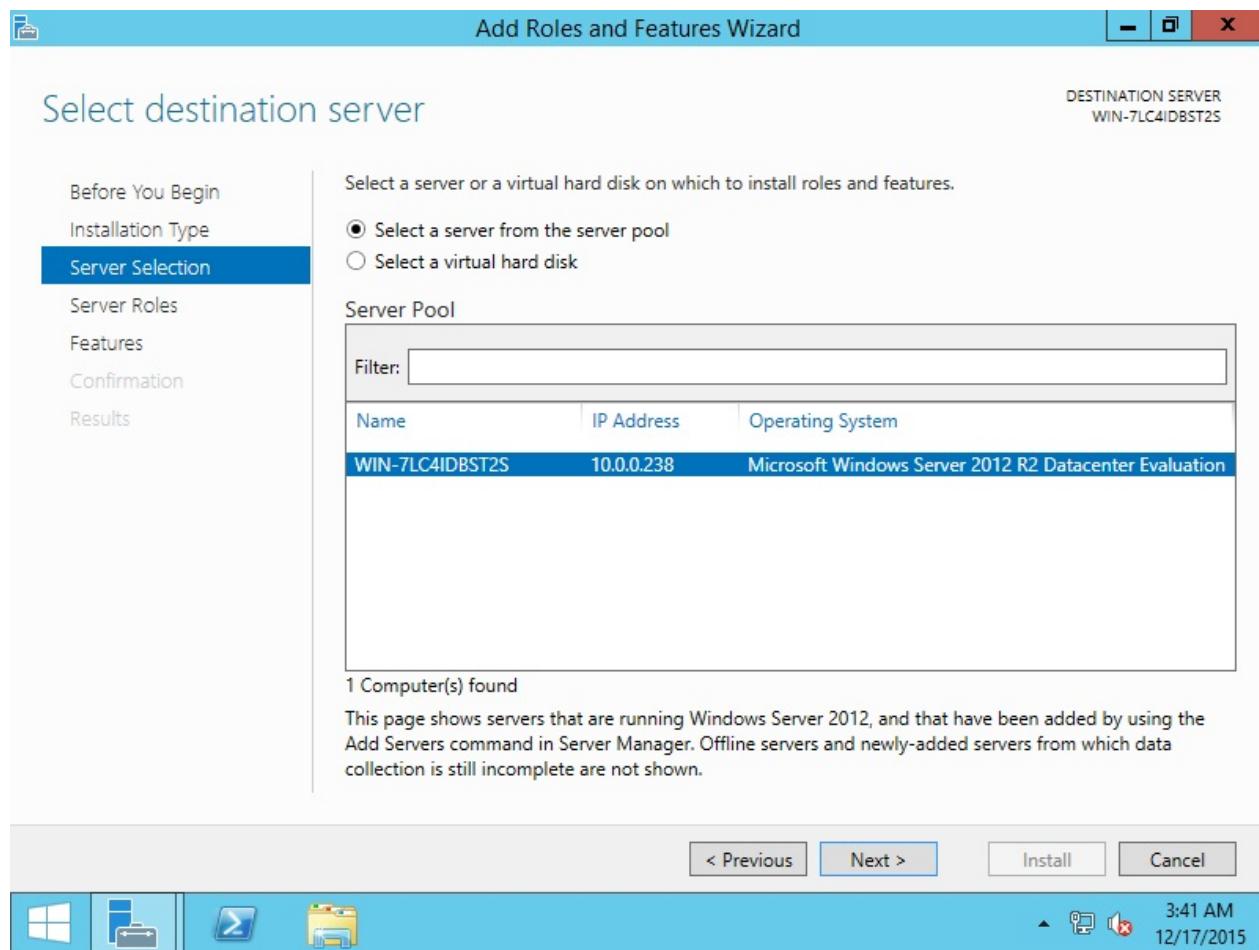
选中“Roles-based or Feature-based installation”（基于角色或基于功能的安装），然后单击“Next”（下一步）继续：

4.1. NFS服务器



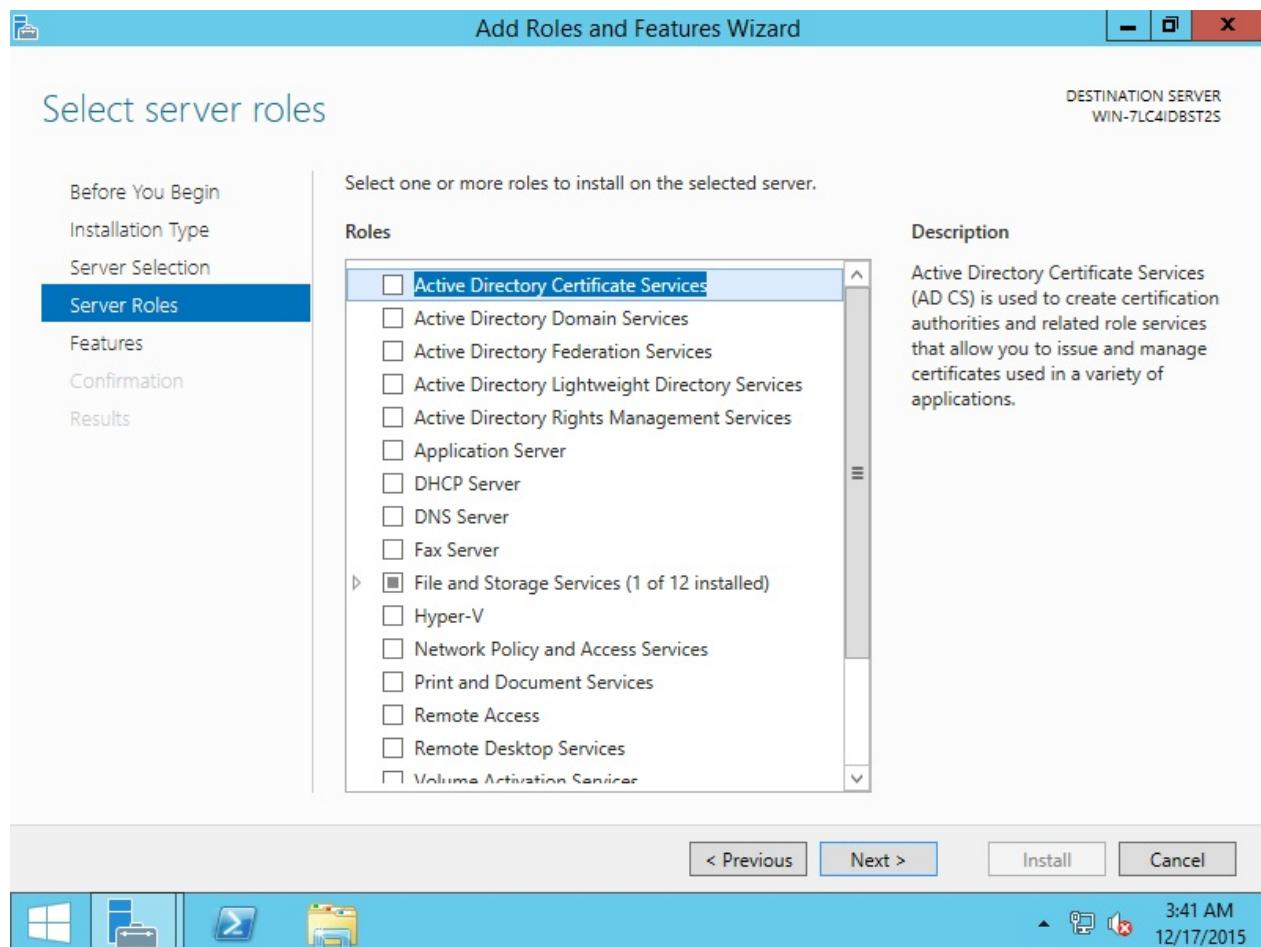
选择您要添加角色或功能的服务器：

4.1. NFS服务器



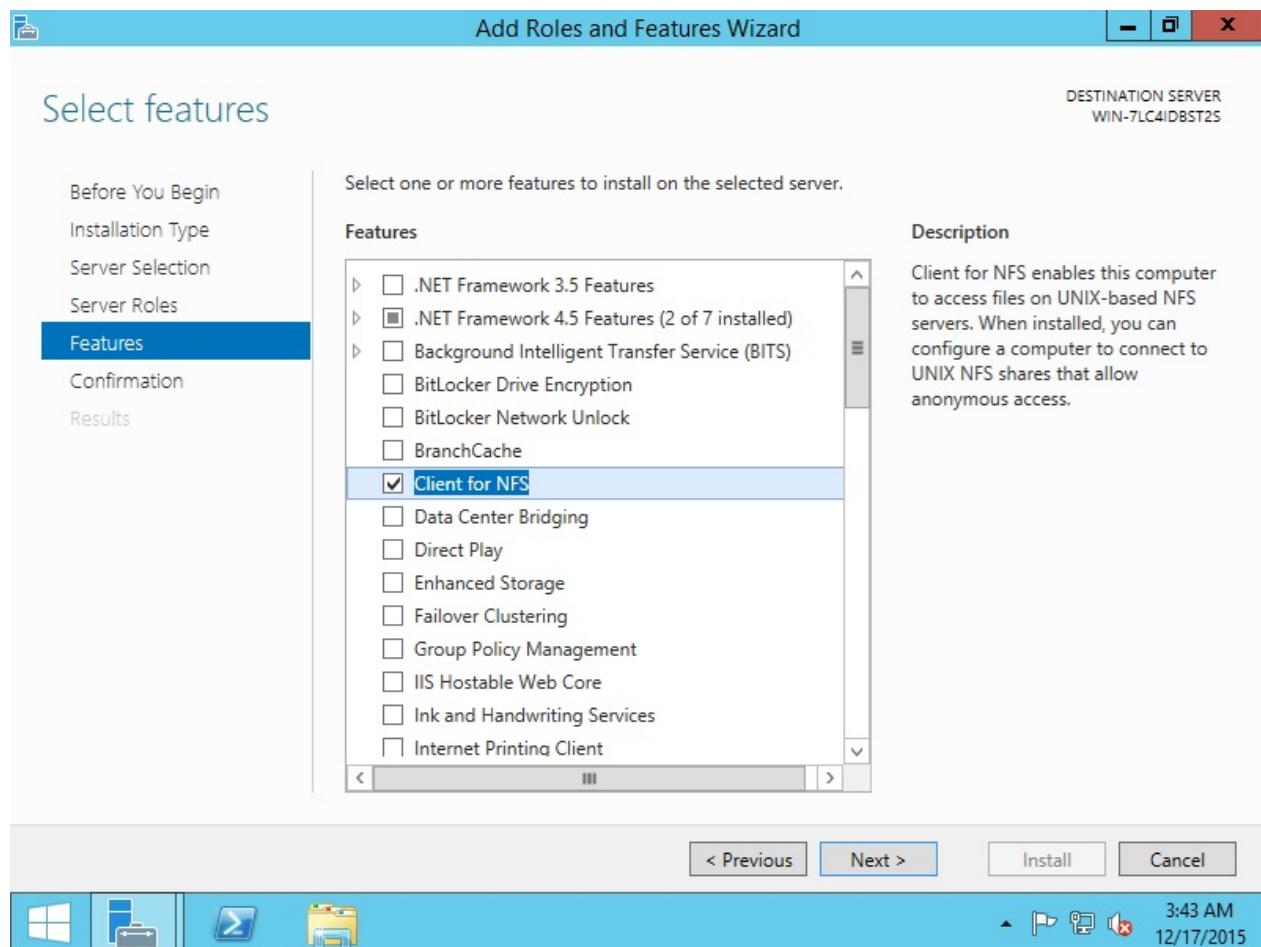
不选中任何内容，单击“Next”（下一步）：

4.1. NFS服务器



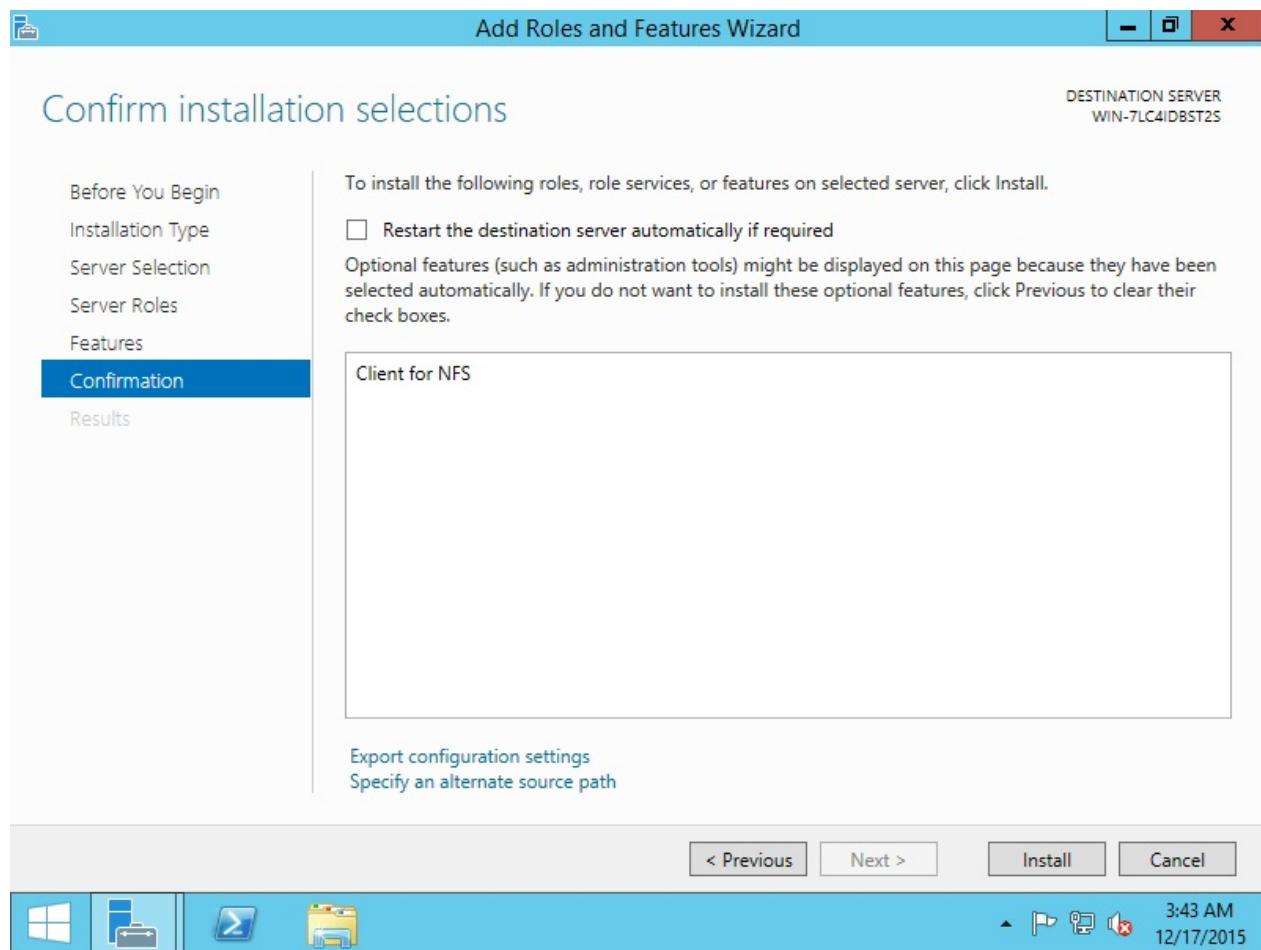
选中“Client for NFS”（NFS客户端）框继续：

4.1. NFS服务器



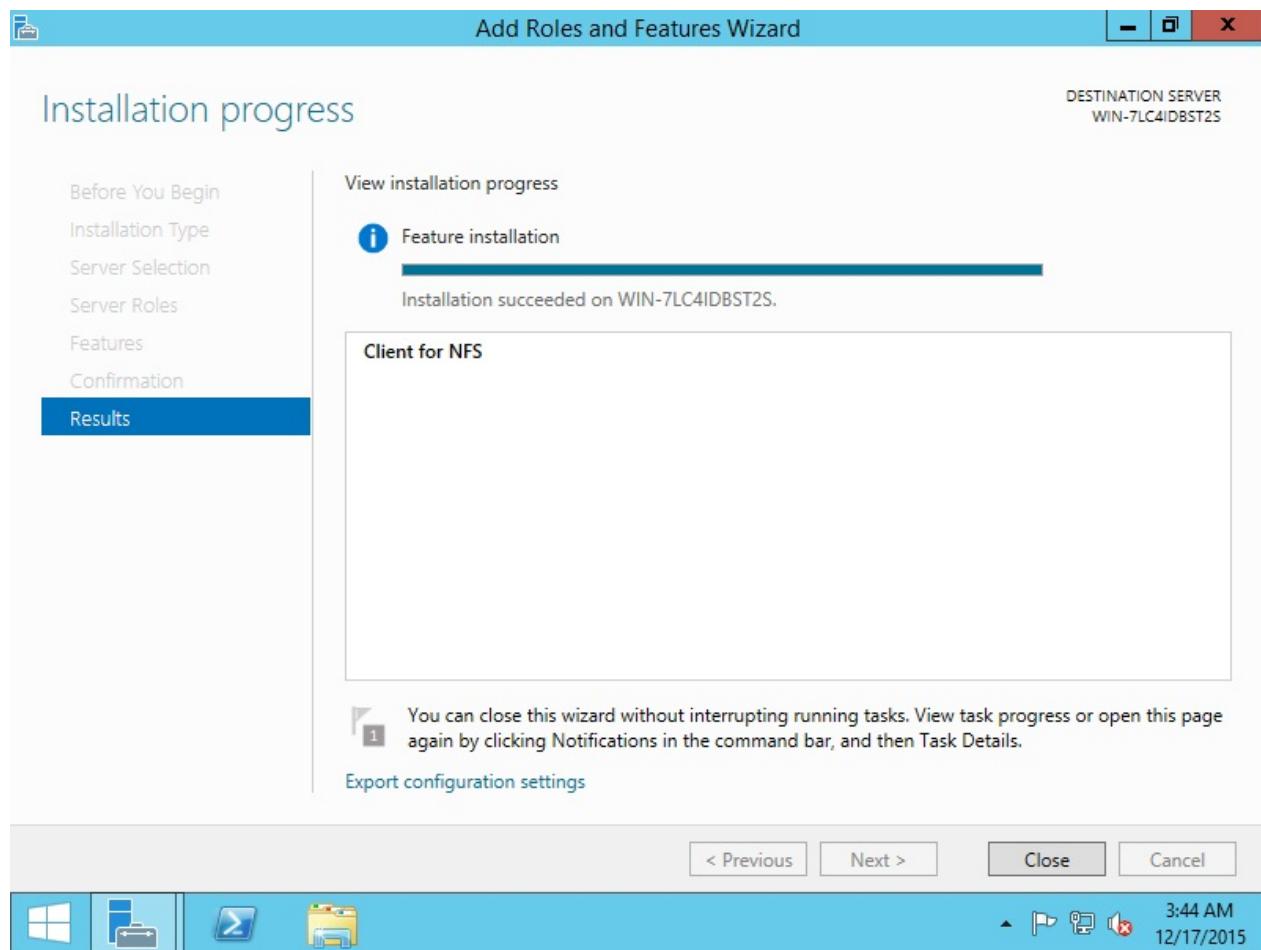
单击“Install”（安装）按钮开始安装：

4.1. NFS服务器



安装完成后，单击“Close”（关闭）按钮：

4.1. NFS服务器

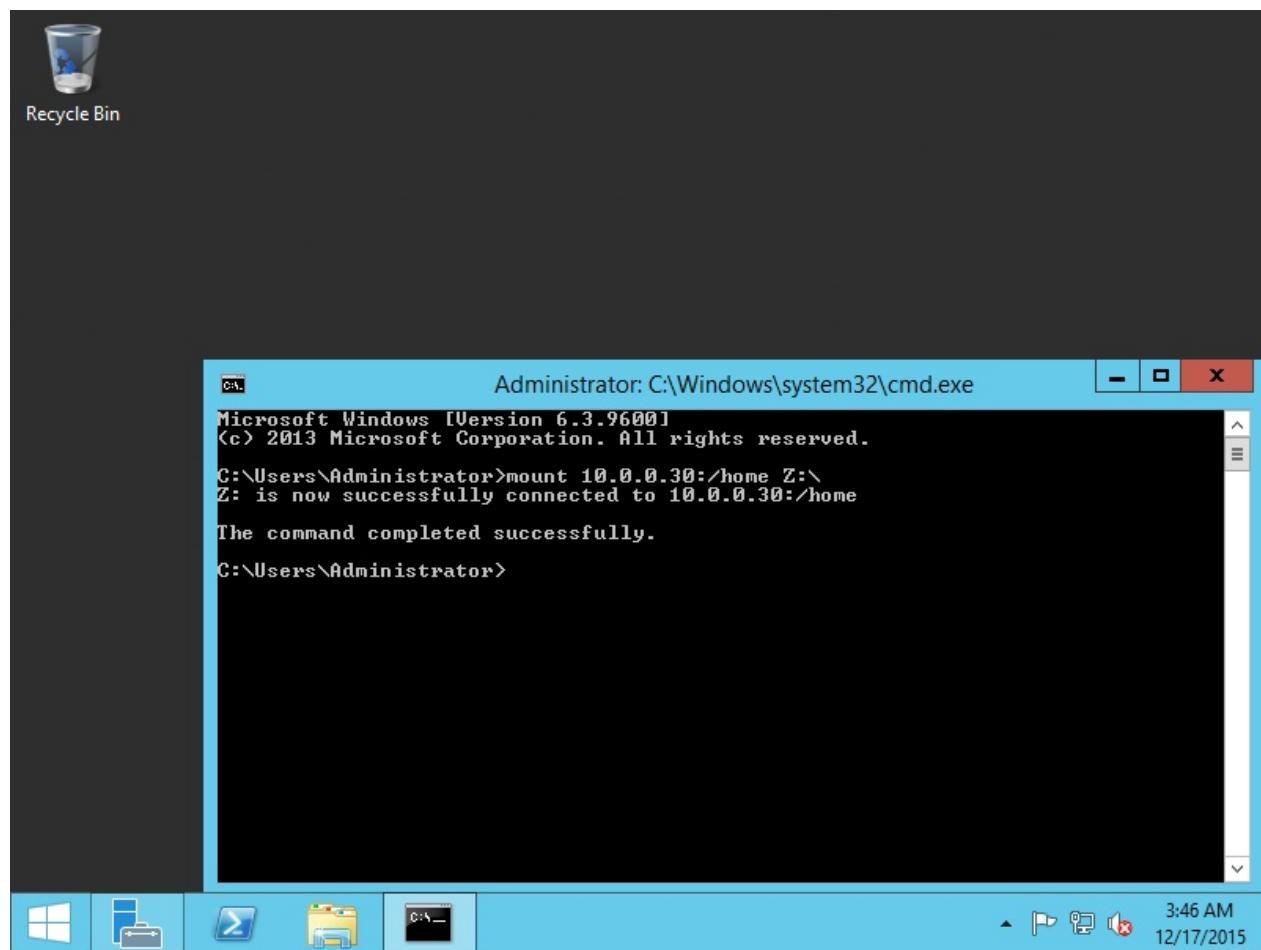


现在可以挂载NFS共享，使用管理员权限运行cmd.exe：

```
mount [NFS服务器主机名或IP地址]:/[共享名称] [本地驱动器]:\
```

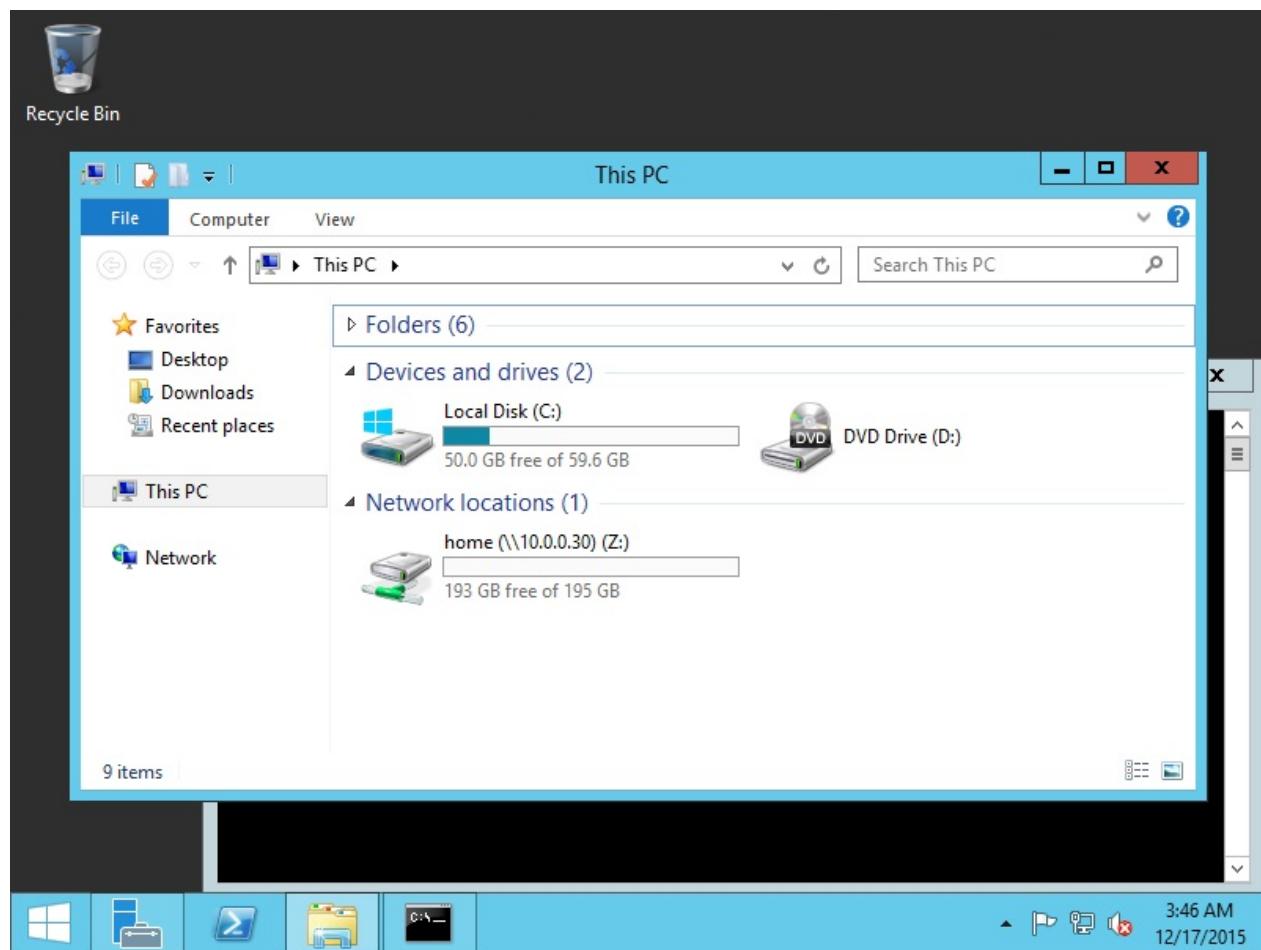
如果连接正常，成功消息如下所示：

4.1. NFS服务器



打开资源管理器，然后挂载NFS共享显示如下：

4.1. NFS服务器

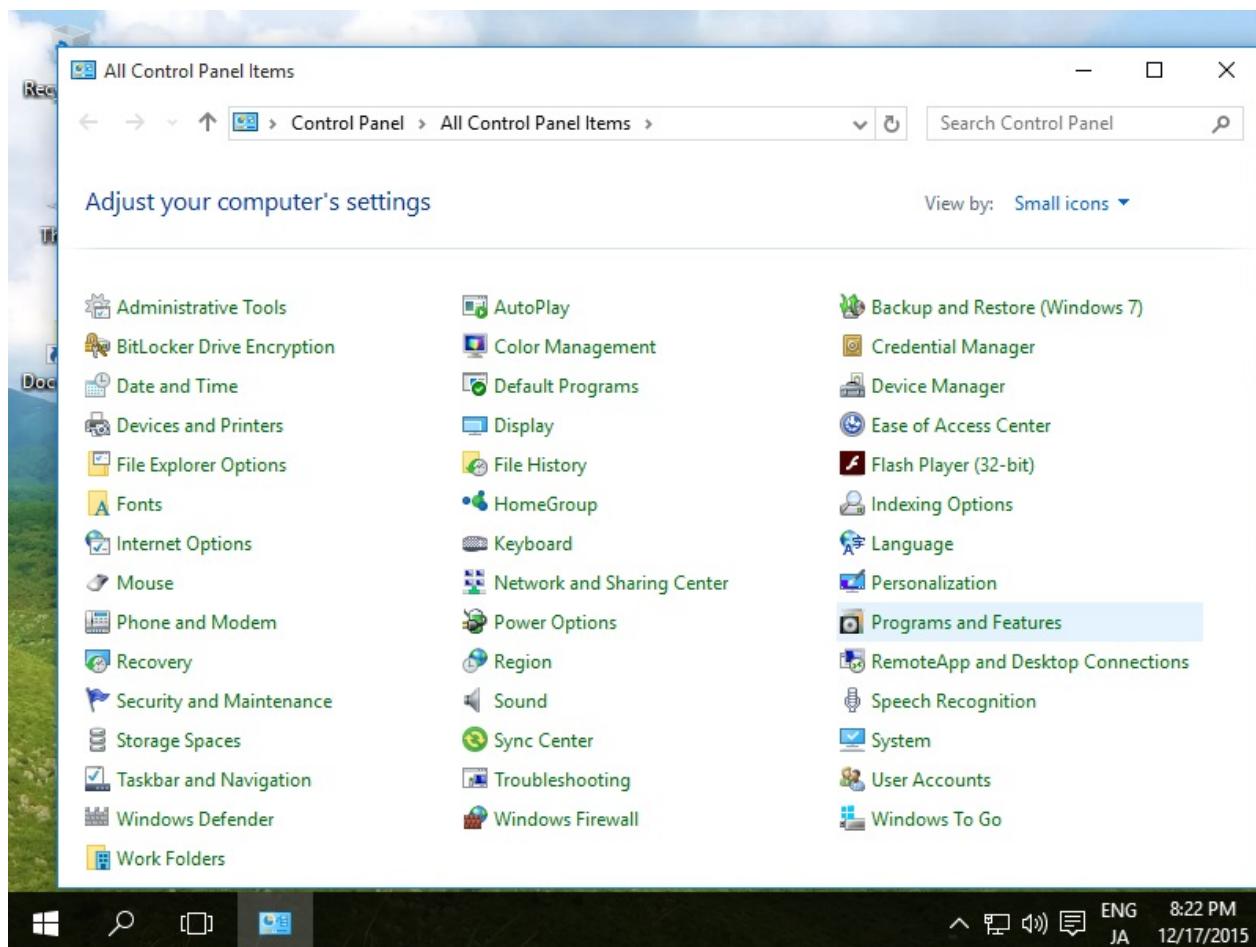


4.1.2.2. Windows桌面系统

以Windows 10为例。

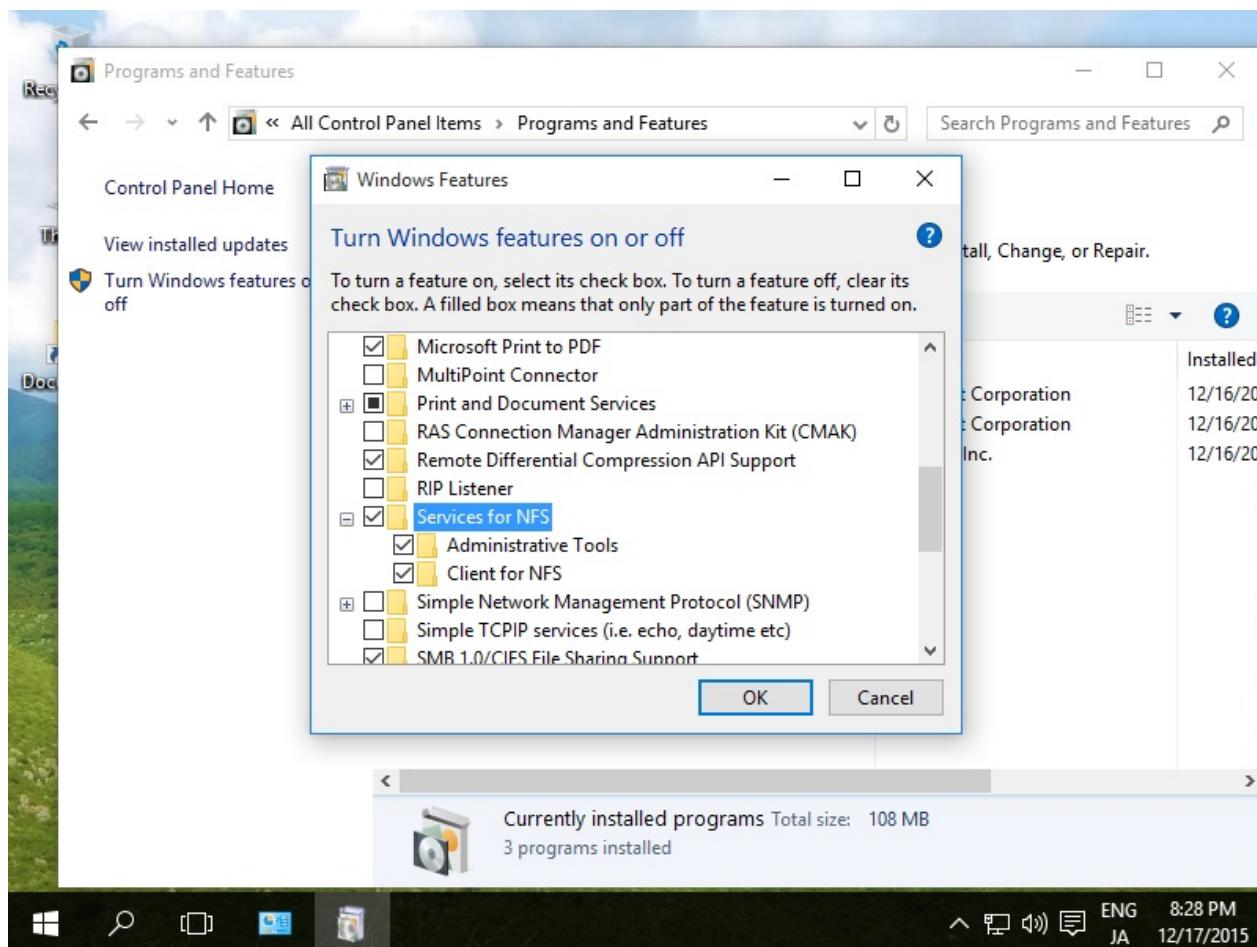
打开“Control Panel”（控制面板）->“Programs and Features”（程序和功能）：

4.1. NFS服务器



单击左侧的“Turn Windows features on or off”（启用或关闭Windows功能），并选择“Services for NFS”（NFS服务）的复选框，如下所示，然后单击“OK”（确定）按钮：

4.1. NFS服务器

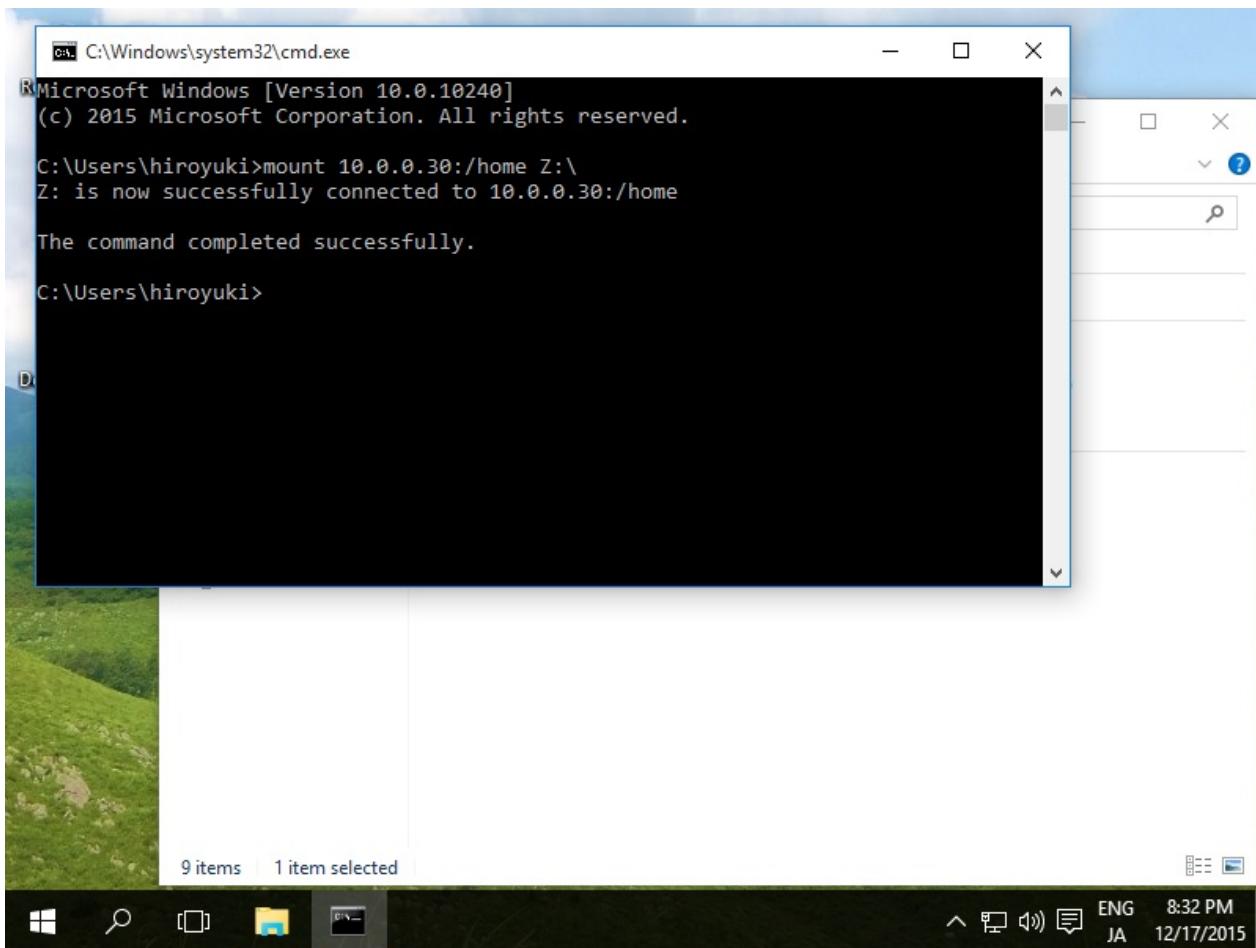


现在可以挂载NFS共享，使用管理员权限运行cmd.exe：

```
mount [NFS服务器主机名或IP地址]:/[共享名称] [本地驱动器]:\
```

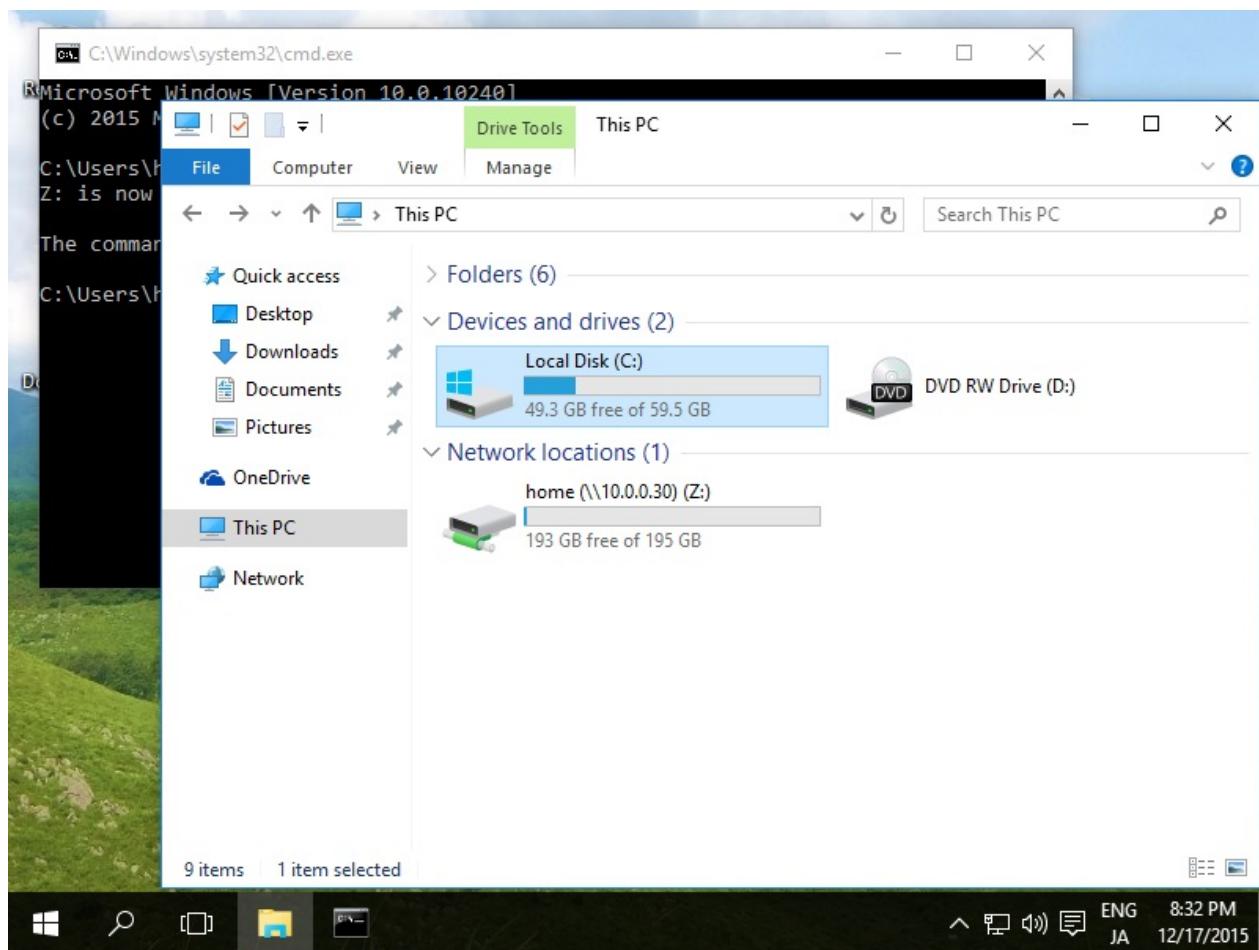
如果连接正常，成功消息如下所示：

4.1. NFS服务器



打开资源管理器，然后挂载NFS共享显示如下：

4.1. NFS服务器

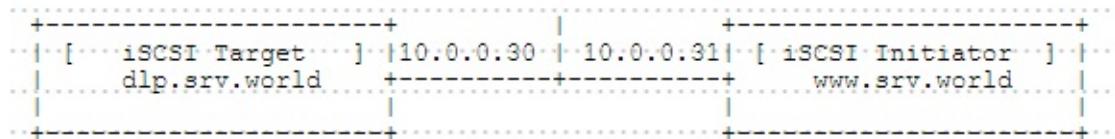


4.2. iSCSI

4.2.1. 配置 iSCSI 目标

网络上的存储称为 iSCSI 目标（iSCSI Target），连接到 iSCSI 目标的客户端称为 iSCSI 启动器（iSCSI Initiator）。

本例基于以下环境：



首先安装管理工具：

```
yum -y install targetcli
```

配置 iSCSI 目标，例如，在 `/iscsi_disks` 目录下创建磁盘映像并将其设置为 SCSI 设备：

```
mkdir /iscsi_disks # 创建目录
```

```
targetcli # 进入管理控制台
```

```

targetcli shell version 2.1.fb34
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> cd backstores/fileio

# 在 /iscsi_disks/disk01.img 上创建名为 "disk01" 的 10G 磁盘映像
/backstores/fileio> create disk01 /iscsi_disks/disk01.img 10G
Created fileio disk01 with size 10737418240

/backstores/fileio> cd /iscsi

# 创建目标
/iscsi> create iqn.2014-07.world.srv:storage.target00
Created target iqn.2014-07.world.srv:storage.target00.
Created TPG 1.

```

4.2. iSCSI

```
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260
.

/iscsi> cd iqn.2014-07.world.srv:storage.target00/tpg1/luns

# 设置LUN
/iscsi/iqn.20...t00/tpg1/luns> create /backstores/fileio/disk01
Created LUN 0.

/iscsi/iqn.20...t00/tpg1/luns> cd ../acls

# 设置ACL（它是允许连接的启动器的IQN）
/iscsi/iqn.20...t00/tpg1/acls> create iqn.2014-07.world.srv:www.
srv.world
Created Node ACL for iqn.2014-07.world.srv:www.srv.world
Created mapped LUN 0.

/iscsi/iqn.20...t00/tpg1/acls> cd iqn.2014-07.world.srv:www.srv.
world

# 设置用于验证的UserID
/iscsi/iqn.20....srv.world> set auth userid=username
Parameter userid is now 'username'.

/iscsi/iqn.20....srv.world> set auth password=password
Parameter password is now 'password'.

/iscsi/iqn.20....srv.world> exit
Global pref auto_save_on_exit=true
Last 10 configs saved in /etc/target/backup.
Configuration saved to /etc/target/saveconfig.json
```

配置完成后，目标进入监听如下：

```
ss -napt | grep 3260
```

```
LISTEN      0        256          *:3260          * : *
```

```
systemctl enable target
```

4.2. iSCSI

firewalld防火墙规则：

```
firewall-cmd --add-service=iscsi-target --permanent  
firewall-cmd --reload
```

使用**scsi-target-utils**配置*iSCSI*目标：

```
yum --enablerepo=epel -y install scsi-target-utils # 从EPEL安装
```

配置*iSCSI*目标，例如，在`/iscsi_disks`目录下创建磁盘映像并将其设置为共享磁盘：

```
mkdir /iscsi_disks
```

```
dd if=/dev/zero of=/iscsi_disks/disk01.img count=0 bs=1 seek=10G
```

编辑`/etc/tgt/targets.conf`文件：

```
# 添加以下内容到最后  
# 如果您设置了一些设备，添加<target>-</target>，并按照相同的方式设置  
# 命名规则：[ iqn.year-month.domain:any name ]  
<target target00>  
    # 提供的设备作为iSCSI目标  
    backing-store /iscsi_disks/disk01.img  
    # 允许连接的iSCSI启动器IP地址  
    initiator-address 10.0.0.31  
    # 验证信息（设置任意“用户名”，“密码”）  
    incominguser username password  
</target>
```

如果启用了SELinux，更改SELinux Context：

```
chcon -R -t tgtd_var_lib_t /iscsi_disks
```

```
semanage fcontext -a -t tgtd_var_lib_t /iscsi_disks
```

firewalld防火墙规则：

```
firewall-cmd --add-service=iscsi-target --permanent  
firewall-cmd --reload
```

4.2. iSCSI

启动tgtd并验证状态：

```
systemctl start tgtd  
systemctl enable tgtd
```

```
tgtadm --mode target --op show
```

```
Target 1: iqn.2015-12.world.srv:target00
System information:
  Driver: iscsi
  State: ready
I_T nexus information:
LUN information:
  LUN: 0
    Type: controller
    SCSI ID: IET      00010000
    SCSI SN: beaf10
    Size: 0 MB, Block size: 1
    Online: Yes
    Removable media: No
    Prevent removal: No
    Readonly: No
    SWP: No
    Thin-provisioning: No
    Backing store type: null
    Backing store path: None
    Backing store flags:
  LUN: 1
    Type: disk
    SCSI ID: IET      00010001
    SCSI SN: beaf11
    Size: 10737 MB, Block size: 512
    Online: Yes
    Removable media: No
    Prevent removal: No
    Readonly: No
    SWP: No
    Thin-provisioning: No
    Backing store type: rdwr
    Backing store path: /iscsi_disks/disk01.img
    Backing store flags:
Account information:
  username
ACL information:
  10.0.0.31
```

4.2.2. 配置iSCSI启动器

基于上面示例相同环境。

4.2.2.1. CentOS

```
yum -y install iscsi-initiator-utils
```

编辑 `/etc/iscsi/initiatorname.iscsi` 文件：

```
# 更改为您在iSCSI目标服务器上设置的相同IQN
InitiatorName=iqn.2014-07.world.srv:www.srv.world
```

编辑 `/etc/iscsi/iscsid.conf` 文件：

```
# 取消注释
node.session.auth.authmethod = CHAP

# 取消注释并指定在iSCSI目标服务器上设置的用户名和密码
node.session.auth.username = username
node.session.auth.password = password
```

```
iscsiadm -m discovery -t sendtargets -p 10.0.0.30 #发现目标
```

```
[ 635.510656] iscsi: registered transport (tcp)
10.0.0.30:3260,1 iqn.2014-07.world.srv:storage.target00
```

```
iscsiadm -m node -o show #发现后确认状态
```

```
# BEGIN RECORD 6.2.0.873-21
node.name = iqn.2014-07.world.srv:storage.target00
node.tpgt = 1
node.startup = automatic
node.leading_login = No
...
...
...
node.conn[0].iscsi.IFMarker = No
node.conn[0].iscsi.OFMarker = No
# END RECORD
```

4.2. iSCSI

```
iscsiadm -m node --login # 登录到目标
```

```
Logging in to [iface: default, target: iqn.2014-07.world.srv:storage.target00, portal: 10.0.0.30,3260] (multiple)
[ 708.383308] scsi2 : iSCSI Initiator over TCP/IP
[ 709.393277] scsi 2:0:0:0: Direct-Access      LIO-ORG disk01
               4.0 PQ: 0 ANSI: 5
[ 709.395709] scsi 2:0:0:0: alua: supports implicit and explicit TPGS
[ 709.398155] scsi 2:0:0:0: alua: port group 00 rel port 01
[ 709.399762] scsi 2:0:0:0: alua: port group 00 state A non-preferred supports T0LUSNA
[ 709.401763] scsi 2:0:0:0: alua: Attached
[ 709.402910] scsi 2:0:0:0: Attached scsi generic sg0 type 0
Login to [iface: default, target: iqn.2014-07.world.srv:storage.target00, portal: 10.0.0.30,3260] successful.
```

```
iscsiadm -m session -o show # 确认已建立的会话
```

```
tcp: [1] 10.0.0.30:3260,1 iqn.2014-07.world.srv:storage.target00
(non-flash)
```

确认分区：

major	minor	#blocks	name
252	0	52428800	sda
252	1	512000	sda1
252	2	51915776	sda2
253	0	4079616	dm-0
253	1	47833088	dm-1
8	0	20971520	sdb

从目标服务器提供的新设备添加为“sdb”

设置iSCSI设备后，在启动器上配置如下来使：

```
parted --script /dev/sdb "mklabel msdos" # 创建标签
```

4.2. iSCSI

```
parted --script /dev/sdb "mkpart primary 0% 100%" # 创建分区
```

```
mkfs.xfs -i size=1024 -s size=4096 /dev/sdb1 # 格式化为XFS
```

```
meta-data=/dev/sdb1      isize=1024    agcount=16, agsize=32761  
6 blks  
          =      sectsz=4096  attr=2, projid32bit=1  
          =      crc=0  
data      =      bsize=4096   blocks=5241856, imaxpct=  
25  
          =      sunit=0      swidth=0 blks  
naming    =version 2      bsize=4096   ascii-ci=0 ftype=0  
log       =internal log    bsize=4096   blocks=2560, version=2  
          =      sectsz=4096  sunit=1 blks, lazy-count  
=1  
realtime  =none           extsz=4096   blocks=0, rtextents=0
```

```
mount /dev/sdb1 /mnt # 挂载
```

```
[ 6894.010661] XFS (sdb1): Mounting Filesystem  
[ 6894.031358] XFS (sdb1): Ending clean mount
```

```
df -hT
```

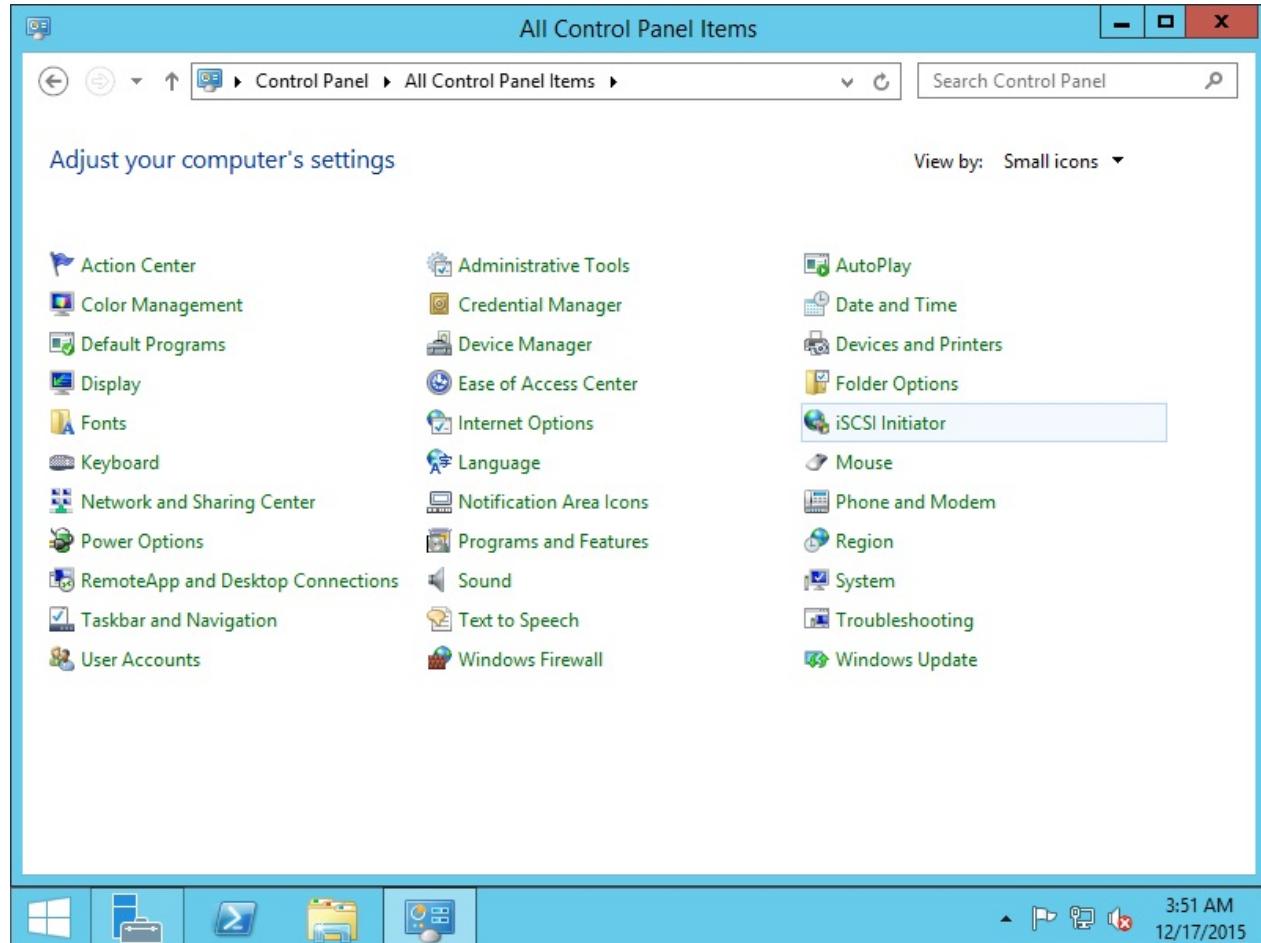
Filesystem on	Type	Size	Used	Avail	Use%	Mounted
/dev/mapper/centos-root	xfs	46G	1023M	45G	3%	/
devtmpfs	devtmpfs	1.9G	0	1.9G	0%	/dev
tmpfs	tmpfs	1.9G	0	1.9G	0%	/dev/shm
tmpfs	tmpfs	1.9G	8.3M	1.9G	1%	/run
tmpfs	tmpfs	1.9G	0	1.9G	0%	/sys/fs/ cgroup
/dev/sda1	xfs	497M	120M	378M	25%	/boot
/dev/sdb1	xfs	20G	33M	20G	1%	/mnt

4.2.2.2. Windows

以Windows Server 2012 R2为例（同样适用于Windows 7/8/10）。

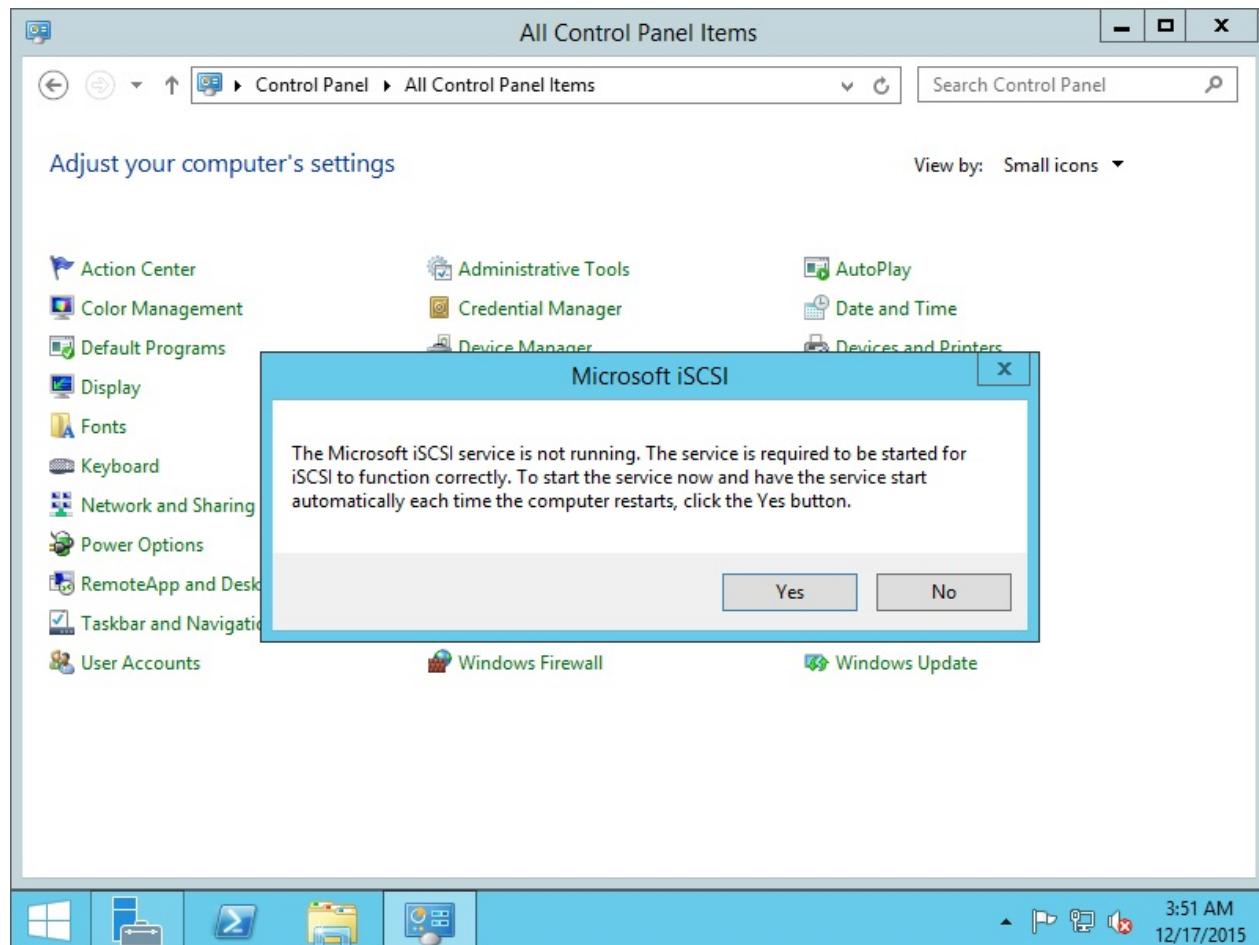
4.2. iSCSI

打开[Control Panel] - [iSCSI Initiator] :



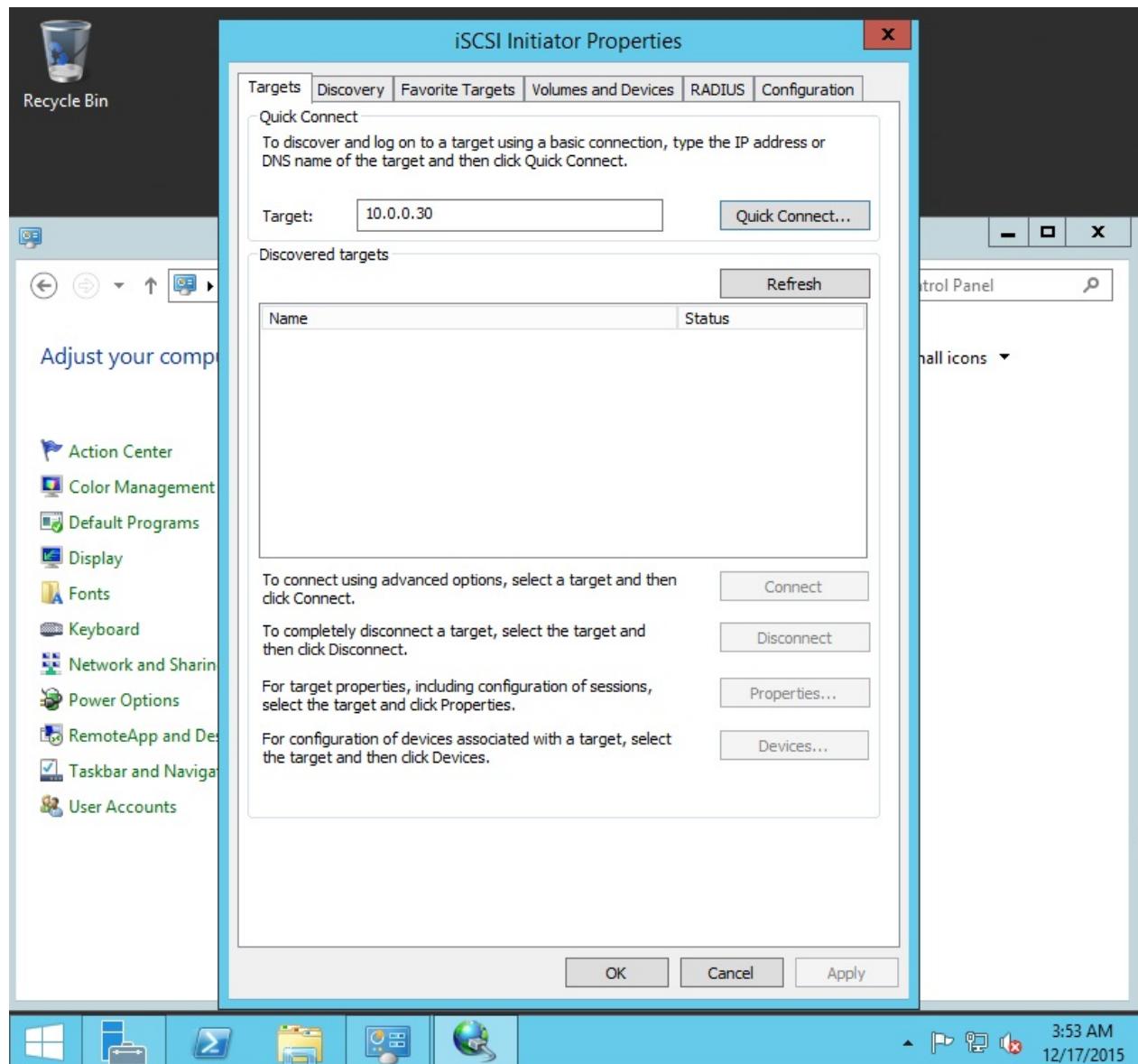
点击[Yes]继续 :

4.2. iSCSI



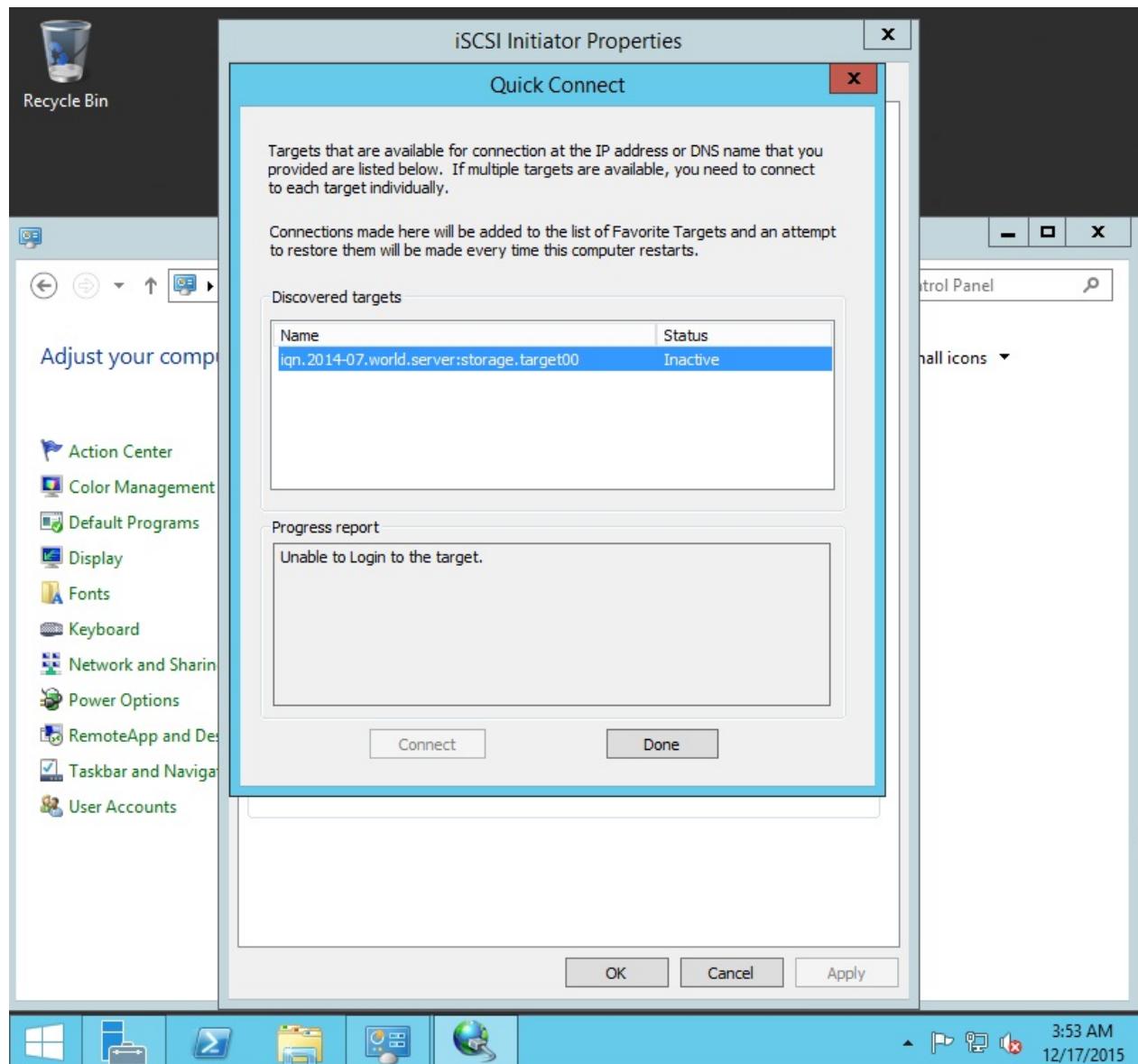
在[Target]部分中输入iSCSI目标服务器的主机名或IP地址，然后单击[Quick Connect]：

4.2. iSCSI



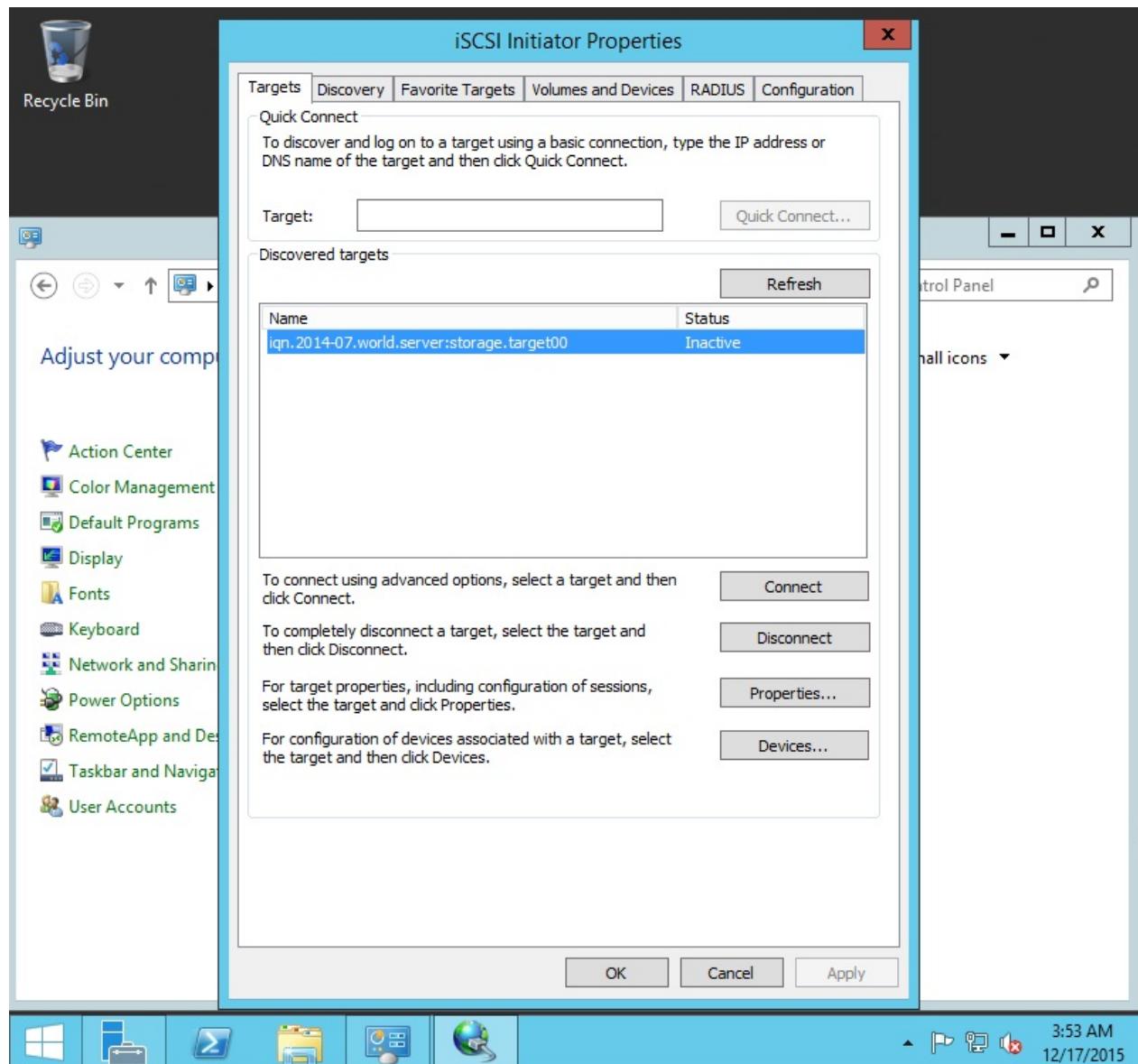
检测到iSCSI目标服务器，然后单击[Done]：

4.2. iSCSI



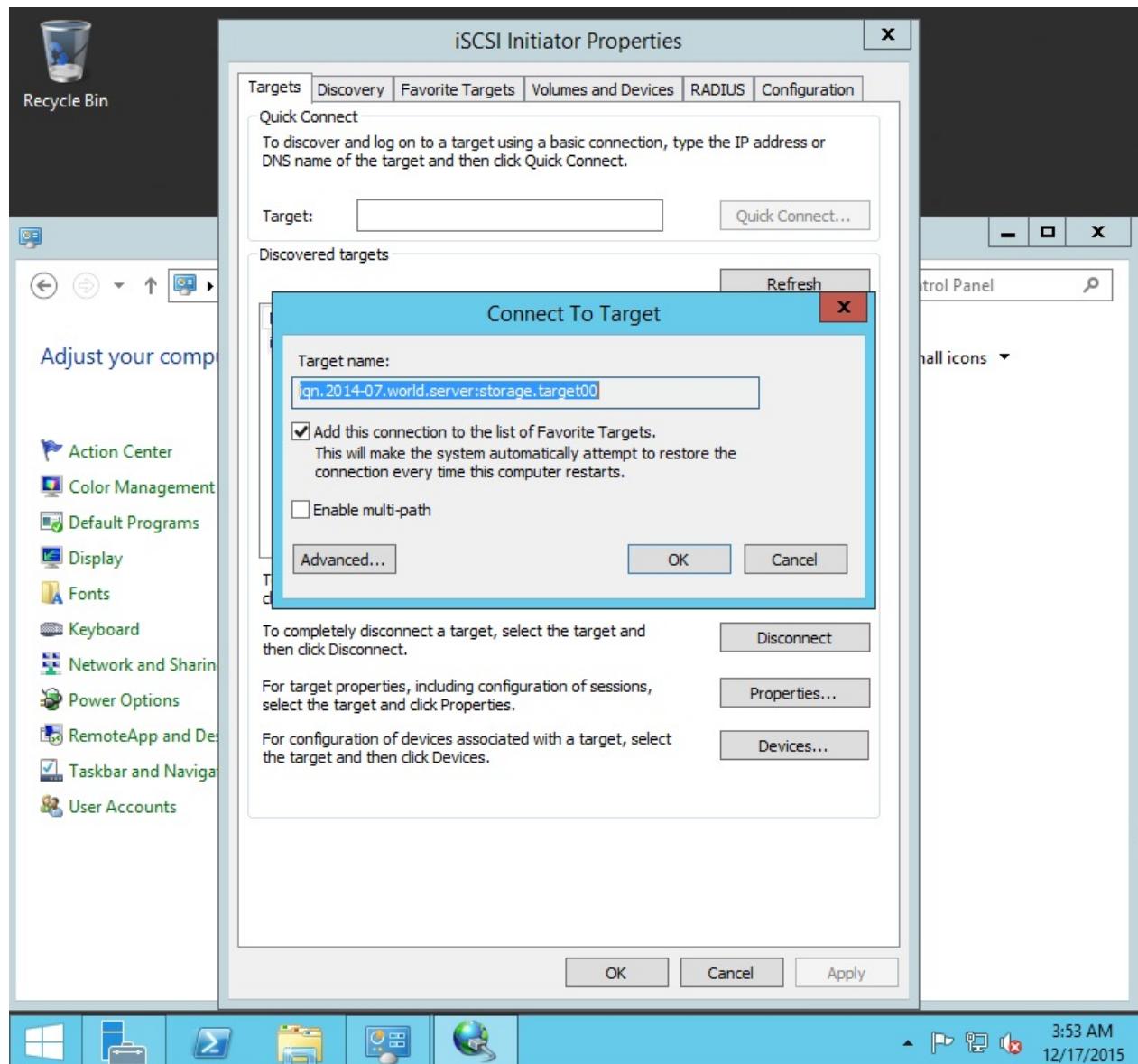
单击[Connect]：

4.2. iSCSI



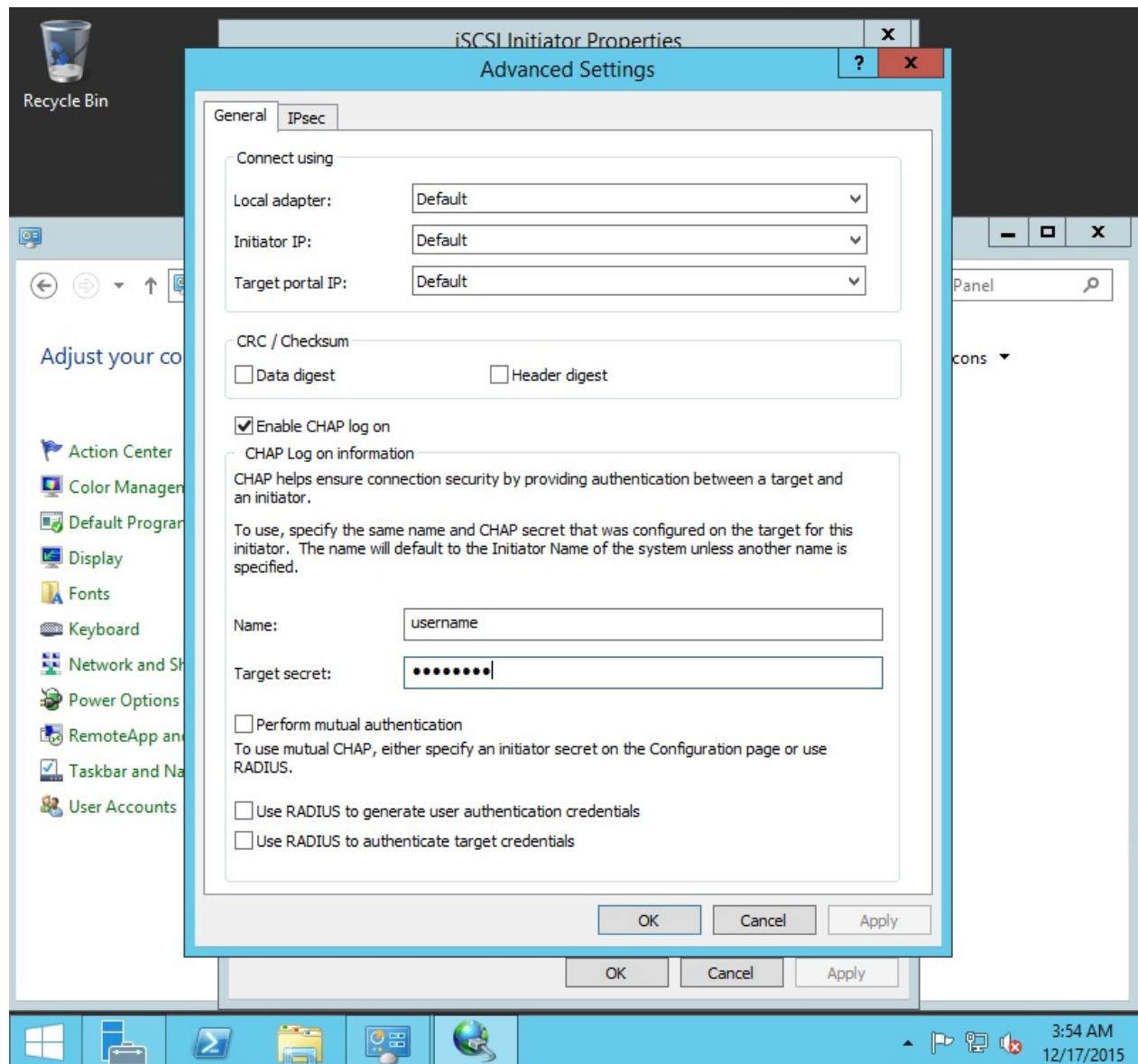
单击[Advanced...]：

4.2. iSCSI



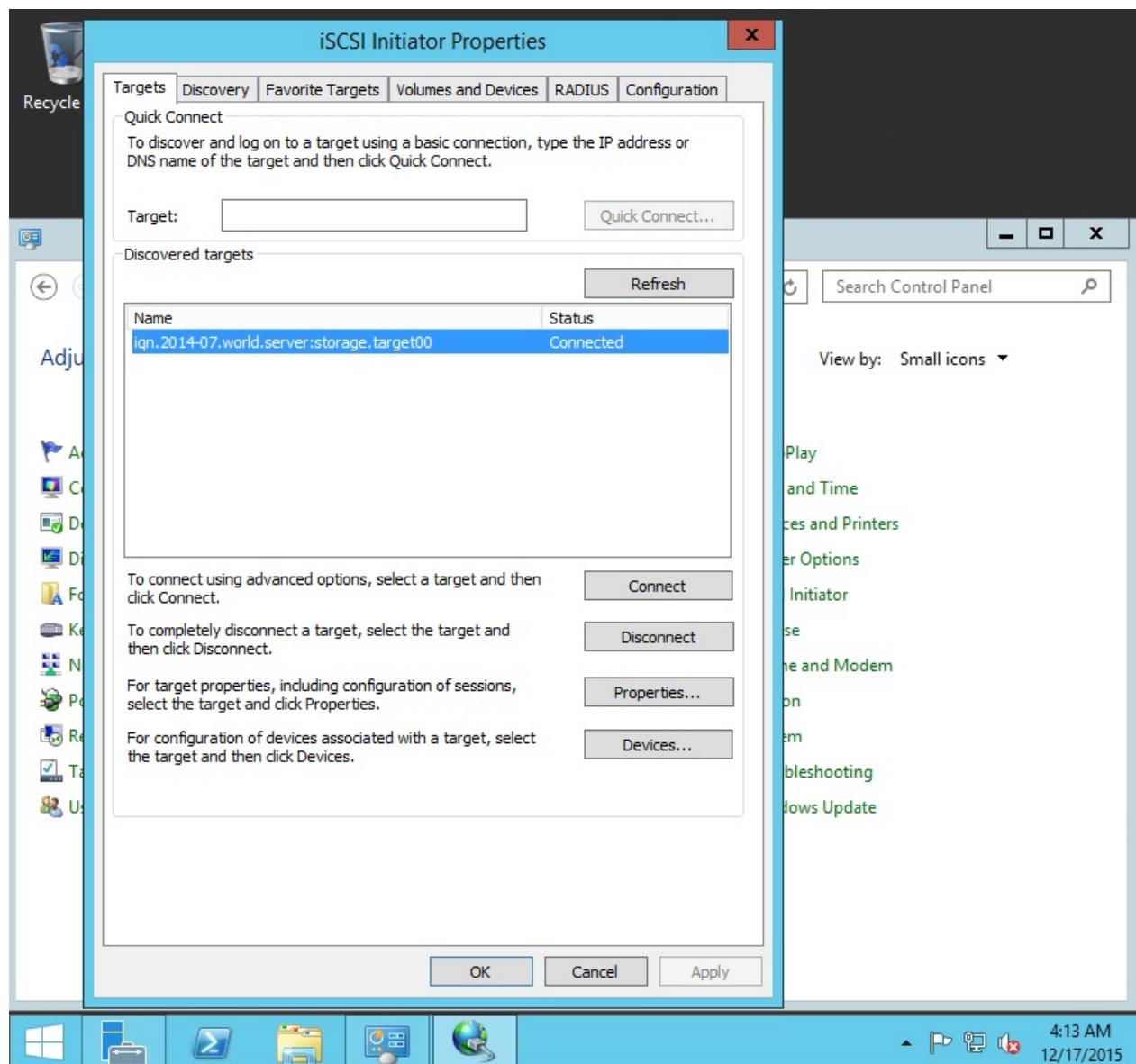
选中[Enable CHAP log on]并在[Name]和[Target Secret]输入在iSCSI目标配置中设置的用户名和密码，单击[OK]，返回到上一步界面，再单击[OK]继续：

4.2. iSCSI



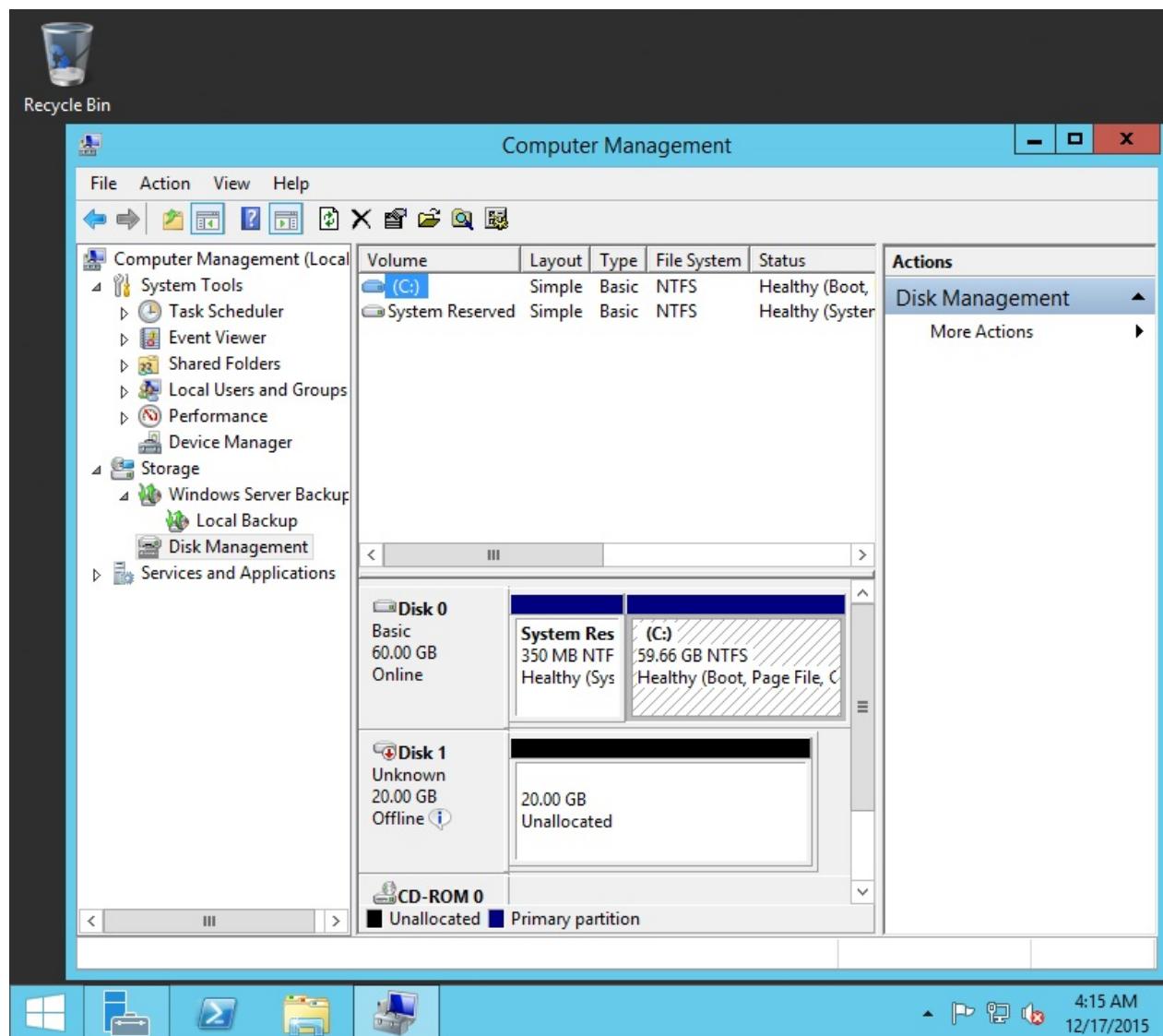
验证成功后，状态将变为[Connected]，如下所示。可以使用iSCSI存储：

4.2. iSCSI



打开[Computer Management]，然后附加iSCSI存储：

4.2. iSCSI

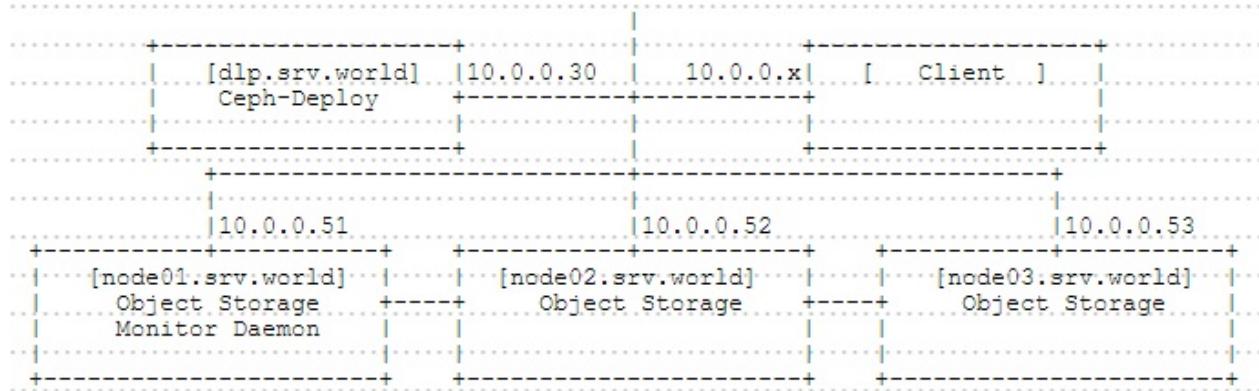


4.3. Ceph

4.3.1. 配置Ceph集群

安装分布式文件系统“[Ceph](#)”以配置存储集群。

本例配置具有1个管理节点和3个存储节点的集群，如下所示：



在所有节点上为Ceph管理员添加一个用户，本例添加“cent”用户。

使用sudo设置授予root权限给刚刚添加的Ceph管理员用户：

```
echo -e 'Defaults:cent !requiretty\ncent ALL = (root) NOPASSWD:ALL'
| tee /etc/sudoers.d/ceph
```

```
chmod 440 /etc/sudoers.d/ceph
```

安装所需的软件包：

```
yum -y install centos-release-ceph-hammer epel-release yum-plugin-priorities

sed -i -e "s/enabled=1/enabled=1\npriority=1/g"
/etc/yum.repos.d/CentOS-Ceph-Hammer.repo
```

firewalld防火墙规则，允许SSH服务：

```
firewall-cmd --add-service=ssh --permanent
firewall-cmd --reload
```

在所有节点上设置以上所有。

4.3. Ceph

在监视节点（监视守护程序）上，firewalld防火墙规则，允许所需的端口（6789/TCP）：

```
firewall-cmd --add-port=6789/tcp --permanent  
firewall-cmd --reload
```

在存储节点（对象存储）上，firewalld防火墙规则，允许所需的端口（6800-7100/TCP）：

```
firewall-cmd --add-port=6800-7100/tcp --permanent  
firewall-cmd --reload
```

以Ceph管理员用户身份登录并配置Ceph，从Ceph管理员节点（本例为“dlp.srv.world”）为所有存储节点设置SSH密钥对：

```
ssh-keygen
```

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/cent/.ssh/id_rsa):  
Created directory '/home/cent/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/cent/.ssh/id_rsa.  
Your public key has been saved in /home/cent/.ssh/id_rsa.pub.  
The key fingerprint is:  
54:c3:12:0e:d3:65:11:49:11:73:35:1b:e3:e8:63:5a cent@dlp.srv.world  
The key's randomart image is:
```

编辑 ~/.ssh/config 文件：

4.3. Ceph

```
Host dlp
    Hostname dlp.srv.world
    User cent
Host node01
    Hostname node01.srv.world
    User cent
Host node02
    Hostname node02.srv.world
    User cent
Host node03
    Hostname node03.srv.world
    User cent
```

```
chmod 600 ~/.ssh/config
```

```
ssh-copy-id node01 # 传输密钥文件
```

```
cent@node01.srv.world's password:
```

```
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh 'node01'"
and check to make sure that only the key(s) you wanted were added.
```

```
ssh-copy-id node02
```

```
ssh-copy-id node03
```

从管理节点将Ceph安装到所有节点：

```
sudo yum -y install ceph-deploy
```

```
mkdir ceph
cd ceph
```

```
ceph-deploy new node01
```

编辑 ./ceph.conf 文件：

4.3. Ceph

```
# 添加到最后  
osd pool default size = 2
```

在每个节点上安装Ceph：

```
ceph-deploy install dlp node01 node02 node03
```

监视和密钥的设置：

```
ceph-deploy mon create-initial
```

从管理节点配置Ceph集群（在此之前，在Node01上创建目录/storage01，在Node02上创建/storage02，在node03上创建/storage03）：

```
ceph-deploy osd prepare node01:/storage01 node02:/storage02  
node03:/storage03 # 准备对象存储守护程序
```

```
ceph-deploy osd activate node01:/storage01 node02:/storage02  
node03:/storage03 # 激活对象存储守护程序
```

传输配置文件：

```
ceph-deploy admin dlp node01 node02 node03
```

```
sudo chmod 644 /etc/ceph/ceph.client.admin.keyring
```

显示状态（如果没有问题，显示如下）：

```
ceph health
```

```
HEALTH_OK
```

BTW，如果想清理设置并重新配置，参照下面的步骤：

```
ceph-deploy purge dlp node01 node02 node03 # 删除包
```

删除设置：

```
ceph-deploy purgedata dlp node01 node02 node03
```

```
ceph-deploy forgetkeys
```

4.3.2. 客户端

基于上面示例相同环境，将客户端配置为使用Ceph存储

4.3.2.1. 用作块设备

创建块设备并将其挂载到客户端上。

首先，参照上一节为客户端上的用户配置Sudo和SSH密钥对，然后从Ceph管理节点安装Ceph，如下所示：

```
ceph-deploy install client
```

```
ceph-deploy admin client
```

创建块设备并将其挂载到客户端上：

```
sudo chmod 644 /etc/ceph/ceph.client.admin.keyring
```

```
rbd create disk01 --size 10240 # 创建一个10G的磁盘
```

```
rbd ls -l # 显示列表
```

NAME	SIZE	PARENT	FMT	PROT	LOCK
disk01	10240M				2

```
sudo rbd map disk01 # 将映像映射到设备
```

```
/dev/rbd0
```

```
rbd showmapped # 显示映射
```

id	pool	image	snap	device
0	rbd	disk01	-	/dev/rbd0

```
sudo mkfs.xfs /dev/rbd0 # 格式化为XFS
```

```
sudo mount /dev/rbd0 /mnt # 挂载设备
```

```
df -hT
```

Filesystem on	Type	Size	Used	Avail	Use%	Mounted
/dev/mapper/centos-root	xfs	27G	1.3G	26G	5%	/
devtmpfs	devtmpfs	2.0G	0	2.0G	0%	/dev
tmpfs	tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	tmpfs	2.0G	8.4M	2.0G	1%	/run
tmpfs	tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
/dev/vda1	xfs	497M	151M	347M	31%	/boot
/dev/rbd0	xfs	10G	33M	10G	1%	/mnt

4.3.2.2. 用作文件系统

在客户端上用作文件系统挂载。

在要设置MDS的节点上创建MDS（MetaData Server），本例设置为node01：

```
ceph-deploy mds create node01
```

在MDS节点上创建至少2个RADOS池并激活MetaData Server，对于在创建命令结束时指定的pg_num，参考[官方文档](#)并确定适当的值：

```
sudo chmod 644 /etc/ceph/ceph.client.admin.keyring
```

创建池：

```
ceph osd pool create cephfs_data 128
```

```
pool 'cephfs_data' created
```

```
ceph osd pool create cephfs_metadata 128
```

```
pool 'cephfs_metadata' created
```

启用池：

```
ceph fs new cephfs cephfs_metadata cephfs_data
```

```
new fs with metadata pool 2 and data pool 1
```

4.3. Ceph

显示列表：

```
ceph fs ls
```

```
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data ]
```

```
ceph mds stat
```

```
e5: 1/1/1 up {0=node01=up:active}
```

在客户端上挂载CephFS：

```
yum -y install ceph-fuse
```

获取管理员密钥

```
ssh cent@node01.srv.world "sudo ceph auth tool -p /etc/ceph/ceph.client.admin.keyring" > admin.key
```

```
cent@node01.srv.world's password:
```

```
chmod 600 admin.key
```

```
mount -t ceph node01.srv.world:6789:/ /mnt -o name=admin,secretfile=admin.key
```

```
df -hT
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/mapper/centos-root	xfs	27G	1.3G	26G	5%	/
devtmpfs	devtmpfs	2.0G	0	2.0G	0%	/dev
tmpfs	tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	tmpfs	2.0G	8.3M	2.0G	1%	/run
tmpfs	tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
/dev/vda1	xfs	497M	151M	347M	31%	/boot
10.0.0.51:6789:/	ceph	80G	19G	61G	24%	/mnt

4.4. GlusterFS

安装GlusterFS以配置存储集群。

4.4.1. 安装GlusterFS

在集群中的所有节点上安装GlusterFS服务器：

```
yum -y install centos-release-gluster38  
sed -i -e "s(enabled=1)enabled=0/g" /etc/yum.repos.d/CentOS-  
Gluster-3.8.repo  
yum --enablerepo=centos-gluster38,epel -y install glusterfs-server  
  
systemctl start glusterd  
systemctl enable glusterd
```

firewalld防火墙规则，允许所有节点上的GlusterFS服务：

```
firewall-cmd --add-service=glusterfs --permanent  
firewall-cmd --reload
```

可以从使用GlusterFS Native Client的客户端挂载GlusterFS卷。

GlusterFS支持NFS（v3），因此如果从客户端用NFS挂载GlusterFS卷，配置如下：

```
yum -y install rpcbind  
  
systemctl start rpcbind  
systemctl enable rpcbind  
systemctl restart glusterd
```

firewalld防火墙规则：

```
firewall-cmd --add-service={nfs, rpc-bind} --permanent  
firewall-cmd --reload
```

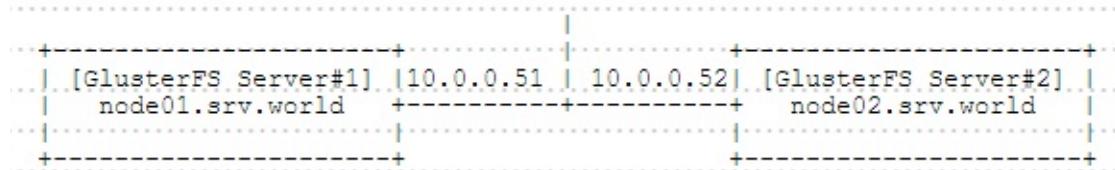
GlusterFS的安装和基本设置已完成，下一节是有关集群的设置。

4.4.2. 配置存储集群

建议对不同于 / 的分区使用GlusterFS卷，本例环境设置为sdb1挂载在所有节点上为GlusterFS配置的 /glusterfs 目录。

4.4.2.1. 分布式配置

创建具有多个服务器的分布式卷，本例演示使用两个服务器的配置，也可以使用三个以上的服务器。



先在所有节点安装GlusterFS。

在所有节点创建GlusterFS卷的目录：

```
mkdir /glusterfs/distributed
```

在节点（任一节点）上按如下方式配置集群：

探测节点：

```
gluster peer probe node02
```

peer probe: success.

显示状态：

gluster peer status

4.4. GlusterFS

```
Number of Peers: 1

Hostname: node02
Uuid: 2ca22769-28a1-4204-9957-886579db2231
State: Peer in Cluster (Connected)
```

创建卷：

```
gluster volume create vol_distributed transport tcp \
node01:/glusterfs/distributed \
node02:/glusterfs/distributed
```

```
volume create: vol_distributed: success: please start the volume
to access data
```

启动卷：

```
gluster volume start vol_distributed
```

```
volume start: vol_distributed: success
```

显示卷信息：

```
gluster volume info
```

```
Volume Name: vol_distributed
Type: Distribute
Volume ID: 5dc9f392-2bfe-4100-b8a5-1f9a817cf54a
Status: Started
Number of Bricks: 2
Transport-type: tcp
Bricks:
Brick1: node01:/glusterfs/distributed
Brick2: node02:/glusterfs/distributed
Options Reconfigured:
transport.address-family: inet
performance.readdir-ahead: on
nfs.disable: on
```

如果要从客户端使用NFS挂载，首先启动所需的服务，并配置如下：

```
gluster volume set vol_distributed nfs.disable off
```

```
volume set: success
```

4.4.2.2. 复制配置

创建具有多个服务器的复制卷，本例演示使用两个服务器的配置，也可以使用三个以上的服务器。



先在所有节点安装GlusterFS。

在所有节点创建GlusterFS卷的目录：

```
mkdir /glusterfs/replica
```

在节点（任一节点）上按如下方式配置集群：

探测节点：

```
gluster peer probe node02
```

4.4. GlusterFS

```
peer probe: success.
```

显示状态：

```
gluster peer status
```

```
Number of Peers: 1

Hostname: node02
Uuid: 2ca22769-28a1-4204-9957-886579db2231
State: Peer in Cluster (Connected)
```

创建卷：

```
gluster volume create vol_replica replica 2 transport tcp \
node01:/glusterfs/replica \
node02:/glusterfs/replica
```

```
volume create: vol_replica: success: please start the volume to
access data
```

启动卷：

```
gluster volume start vol_replica
```

```
volume start: vol_replica: success
```

显示卷信息：

```
gluster volume info
```

```
Volume Name: vol_replica
Type: Replicate
Volume ID: 0d5d5ef7-bdfa-416c-8046-205c4d9766e6
Status: Started
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: node01:/glusterfs/replica
Brick2: node02:/glusterfs/replica
Options Reconfigured:
transport.address-family: inet
performance.readdir-ahead: on
nfs.disable: on
```

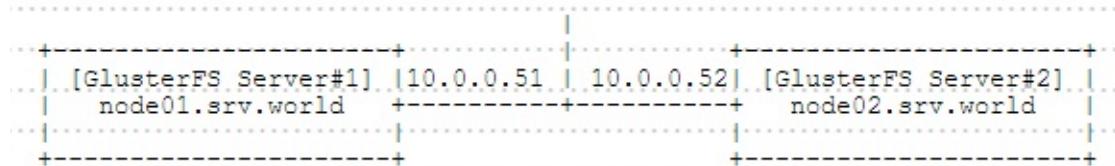
如果要从客户端使用NFS挂载，首先启动所需的服务，并配置如下：

```
gluster volume set vol_replica nfs.disable off
```

```
volume set: success
```

4.4.2.3. 条带卷配置

创建具有多个服务器的条带卷，本例演示使用两个服务器的配置，也可以使用三个以上的服务器。



先在所有节点安装GlusterFS。

在所有节点创建GlusterFS卷的目录：

```
mkdir /glusterfs/stripped
```

在节点（任一节点）上按如下方式配置集群：

探测节点：

```
gluster peer probe node02
```

4.4. GlusterFS

```
peer probe: success.
```

显示状态：

```
gluster peer status
```

```
Number of Peers: 1

Hostname: node02
Uuid: 2ca22769-28a1-4204-9957-886579db2231
State: Peer in Cluster (Connected)
```

创建卷：

```
gluster volume create vol_stripped stripe 2 transport tcp \
node01:/glusterfs/stripped \
node02:/glusterfs/stripped
```

```
volume create: vol_stripped: success: please start the volume to
access data
```

启动卷：

```
gluster volume start vol_stripped
```

```
volume start: vol_stripped: success
```

显示卷信息：

```
gluster volume info
```

```
Volume Name: vol_striped
Type: Stripe
Volume ID: b6f6b090-3856-418c-aed3-bc430db91dc6
Status: Started
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: node01:/glusterfs/stripped
Brick2: node02:/glusterfs/stripped
Options Reconfigured:
transport.address-family: inet
performance.readdir-ahead: on
nfs.disable: on
```

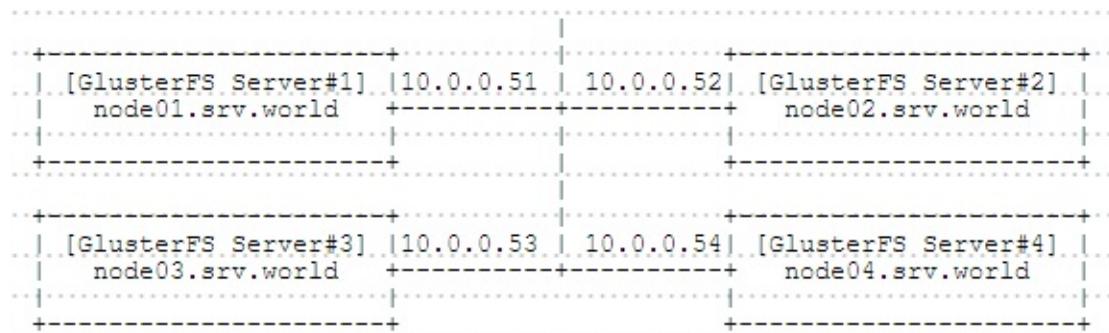
如果要从客户端使用NFS挂载，首先启动所需的服务，并配置如下：

```
gluster volume set vol_striped nfs.disable off
```

```
volume set: success
```

4.4.2.4. 分布式 + 复制

本例创建具有四个服务器的分布式 + 复制卷



先在所有节点安装GlusterFS。

在所有节点创建GlusterFS卷的目录：

```
mkdir /glusterfs/dist-replica
```

在节点（任一节点）上按如下方式配置集群：

探测节点：

4.4. GlusterFS

```
gluster peer probe node02
```

```
peer probe: success.
```

```
gluster peer probe node03
```

```
peer probe: success.
```

```
gluster peer probe node04
```

```
peer probe: success.
```

显示状态：

```
gluster peer status
```

```
Number of Peers: 3
```

```
Hostname: node02
```

```
Uuid: 2ca22769-28a1-4204-9957-886579db2231
```

```
State: Peer in Cluster (Connected)
```

```
Hostname: node03
```

```
Uuid: 79cff591-1e98-4617-953c-0d3e334cf96a
```

```
State: Peer in Cluster (Connected)
```

```
Hostname: node04
```

```
Uuid: 779ab1b3-fda9-46da-af95-ba56477bf638
```

```
State: Peer in Cluster (Connected)
```

创建卷：

```
gluster volume create vol_dist-replica replica 2 transport tcp \
node01:/glusterfs/dist-replica \
node02:/glusterfs/dist-replica \
node03:/glusterfs/dist-replica \
node04:/glusterfs/dist-replica
```

```
volume create: vol_dist-replica: success: please start the volume to access data
```

启动卷：

```
gluster volume start vol_dist-replica
```

```
volume start: vol_dist-replica: success
```

显示卷信息：

```
gluster volume info
```

```
Volume Name: vol_dist-replica
Type: Distributed-Replicate
Volume ID: 784d2953-6599-4102-afc2-9069932894cc
Status: Started
Number of Bricks: 2 x 2 = 4
Transport-type: tcp
Bricks:
Brick1: node01:/glusterfs/dist-replica
Brick2: node02:/glusterfs/dist-replica
Brick3: node03:/glusterfs/dist-replica
Brick4: node04:/glusterfs/dist-replica
Options Reconfigured:
transport.address-family: inet
performance.readdir-ahead: on
nfs.disable: on
```

如果要从客户端使用NFS挂载，首先启动所需的服务，并配置如下：

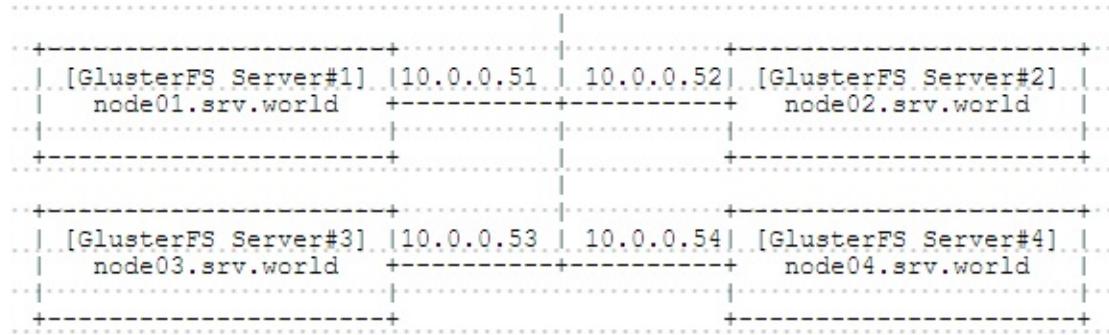
```
gluster volume set vol_dist-replica nfs.disable off
```

```
volume set: success
```

4.4.2.5. 条带 + 复制

本例创建具有四个服务器的条带 + 复制卷

4.4. GlusterFS



先在所有节点安装GlusterFS。

在所有节点创建GlusterFS卷的目录：

```
mkdir /glusterfs/strip-replica
```

在节点（任一节点）上按如下方式配置集群：

探测节点：

```
gluster peer probe node02
```

```
peer probe: success.
```

```
gluster peer probe node03
```

```
peer probe: success.
```

```
gluster peer probe node04
```

```
peer probe: success.
```

显示状态：

```
gluster peer status
```

```
Number of Peers: 3
```

```
Hostname: node02
Uuid: 2ca22769-28a1-4204-9957-886579db2231
State: Peer in Cluster (Connected)
```

```
Hostname: node03
Uuid: 79cff591-1e98-4617-953c-0d3e334cf96a
State: Peer in Cluster (Connected)
```

```
Hostname: node04
Uuid: 779ab1b3-fda9-46da-af95-ba56477bf638
State: Peer in Cluster (Connected)
```

创建卷：

```
gluster volume create vol_strip-replica stripe 2 replica 2 trans
port tcp \
node01:/glusterfs/strip-replica \
node02:/glusterfs/strip-replica \
node03:/glusterfs/strip-replica \
node04:/glusterfs/strip-replica
```

```
volume create: vol_strip-replica: success: please start the volu
me to access data
```

启动卷：

```
gluster volume start vol_strip-replica
```

```
volume start: vol_strip-replica: success
```

显示卷信息：

```
gluster volume info
```

```
Volume Name: vol_strip-replica
Type: Striped-Replicate
Volume ID: ec36b0d3-8467-47f6-aa83-1020555f58b6
Status: Started
Number of Bricks: 1 x 2 x 2 = 4
Transport-type: tcp
Bricks:
Brick1: node01:/glusterfs/stripe-replica
Brick2: node02:/glusterfs/stripe-replica
Brick3: node03:/glusterfs/stripe-replica
Brick4: node04:/glusterfs/stripe-replica
Options Reconfigured:
transport.address-family: inet
performance.readdir-ahead: on
nfs.disable: on
```

如果要从客户端使用NFS挂载，首先启动所需的服务，并配置如下：

```
gluster volume set vol_strip-replica nfs.disable off
```

```
volume set: success
```

4.4.3. 客户端设置

这是GlusterFS客户端安装GlusterFS卷的设置。

使用**GlusterFS Native Client**安装，配置如下：

```
yum -y install centos-release-gluster38
```

```
yum -y install glusterfs glusterfs-fuse
```

将卷vol_distributed挂载到 /mnt：

```
mount -t glusterfs node01.srv.world:/vol_distributed /mnt
```

```
df -hT
```

Filesystem	Type	Size	Used	Ava
il Use% Mounted on				
/dev/mapper/centos-root	xfs	27G	1.5G	2
6G 6% /				
devtmpfs	devtmpfs	2.0G	0	2.
0G 0% /dev				
tmpfs	tmpfs	2.0G	0	2.
0G 0% /dev/shm				
tmpfs	tmpfs	2.0G	8.4M	2.
0G 1% /run				
tmpfs	tmpfs	2.0G	0	2.
0G 0% /sys/fs/cgroup				
/dev/vda1	xfs	497M	208M	29
0M 42% /boot				
tmpfs	tmpfs	396M	0	39
6M 0% /run/user/0				
node01.srv.world:/vol_distributed	fuse.glusterfs	53G	3.0G	5
1G 6% /mnt				

也支持**NFS (v3)**，因此可以使用NFS进行装载（需要先在GlusterFS服务器上配置）：

```
yum -y install nfs-utils
```

```
systemctl start rpcbind rpc-statd
systemctl enable rpcbind rpc-statd
```

```
mount -t nfs -o mountvers=3 node01.srv.world:/vol_distributed /mnt
df -hT
```

4.4. GlusterFS

Filesystem	Type	Size	Used	Avail	Use
% Mounted on					
/dev/mapper/centos-root	xfs	27G	1.5G	26G	6%
% /					
devtmpfs	devtmpfs	2.0G	0	2.0G	0
% /dev					
tmpfs	tmpfs	2.0G	0	2.0G	0
% /dev/shm					
tmpfs	tmpfs	2.0G	8.4M	2.0G	1%
% /run					
tmpfs	tmpfs	2.0G	0	2.0G	0
% /sys/fs/cgroup					
/dev/vda1	xfs	497M	208M	290M	42%
% /boot					
tmpfs	tmpfs	396M	0	396M	0
% /run/user/0					
node01.srv.world:/vol_distributed	nfs	53G	3.0G	51G	6%
% /mnt					

5. Web服务器

- 5.1. Apache httpd
 - 5.1.1. 安装httpd
 - 5.1.2. 使用Perl脚本
 - 5.1.3. 使用PHP脚本
 - 5.1.4. 使用Ruby脚本
 - 5.1.5. 使用Python脚本
 - 5.1.6. 启用Userdir
 - 5.1.7. 虚拟主机
 - 5.1.8. 配置SSL
 - 5.1.9. 启用基本身份验证
 - 5.1.10. 基本身份验证 + PAM
 - 5.1.11. 基本身份验证 + LDAP
 - 5.1.12. 启用Kerberos身份验证
 - 5.1.13. 使用WebDAV
 - 5.1.14. Perl + mod_perl
 - 5.1.15. PHP + PHP-FPM
 - 5.1.16. Python + mod_wsgi
 - 5.1.17. 配置mod_proxy
 - 5.1.17.1. 正向代理
 - 5.1.17.2. 反向代理
 - 5.1.18. 配置mod_proxy_wstunnel
 - 5.1.19. 配置mod_ratelimit
 - 5.1.20. 配置mod_limitipconn
 - 5.1.21. 配置mod_evasive
 - 5.1.22. 配置mod_security
- 5.2. Nginx
 - 5.2.1. 安装Nginx
 - 5.2.2. 虚拟主机
 - 5.2.3. 启用Userdir
 - 5.2.4. 配置SSL
 - 5.2.5. 启用基本身份验证
 - 5.2.6. 反向代理

- [5.2.7. PHP-FPM](#)
- [5.3. 创建SSL证书](#)
 - [5.3.1. 创建自签名SSL证书](#)
 - [5.3.2. 使用Let's Encrypt免费证书](#)

5.1. Apache httpd

Apache HTTP Server（简称Apache）是Apache软件基金会的一个开放源码的网页服务器

5.1.1. 安装httpd

```
yum -y install httpd # 安装httpd
```

```
rm -f /etc/httpd/conf.d/welcome.conf # 删除欢迎界面
```

根据自己的环境配置httpd，编辑 `/etc/httpd/conf/httpd.conf` 文件：

```
# 改为管理员邮件地址
ServerAdmin root@srv.world

# 更改为你服务器的名称
ServerName www.srv.world:80

# 不读取“.htaccess”文件
AllowOverride None

# 将Options Indexes FollowSymLinks改为
Options FollowSymLinks

# 添加目录索引
DirectoryIndex index.html index.cgi index.php

# 添加服务器响应头到结尾
ServerTokens Prod

# 打开keepalive
KeepAlive On
```

相关资料（其他配置或更进一步的内容可自己查阅资料）：

AllowOverride :

是否读取“.htaccess”文件。“All”为是，“None”为否

5.1. Apache httpd

语法为： AllowOverride All|None|directive-type [directive-type] ...

从安全性考虑，根目录的AllowOverride属性一般配置成“None”

FollowSymLinks 选项：

Options FollowSymLinks 禁止显示Apache目录列表

Options Indexes FollowSymLinks 显示Apache目录列表

ServerTokens的一些可能的赋值：

ServerTokens Prod 显示“Server: Apache”

ServerTokens Major 显示“Server: Apache/2”

ServerTokens Minor 显示“Server: Apache/2.2”

ServerTokens Min 显示“Server: Apache/2.2.17”

ServerTokens OS 显示“Server: Apache/2.2.17 (Unix)”

ServerTokens Full 显示“Server: Apache/2.2.17 (Unix) PHP/5.3.5”

KeepAlive的总结：

在内存非常充足的服务器上，不管是否关闭**KeepAlive**功能，服务器性能不会有明显变化；如果服务器内存较少，或者服务器有非常大量的文件系统访问时，或者主要处理动态网页服务，关闭**KeepAlive**后可以节省很多内存，而节省出来的内存用于文件系统Cache，可以提高文件系统访问的性能，并且系统会更加稳定。

```
systemctl start httpd  
systemctl enable httpd
```

客户端访问控制：

5.1. Apache httpd

```
# 禁止访问  
# httpd 2.2上的配置  
Order deny,allow  
Deny from all  
# httpd 2.4上的配置  
Require all denied  
  
# 允许访问  
# httpd 2.2上的配置  
Order allow,deny  
Allow from all  
# httpd 2.4上的配置  
Require all granted
```

httpd 2.4上的示例配置：

5.1. Apache httpd

```
# 仅允许IP为192.168.1.1和192.168.1.2的主机访问
# <RequireAll> ... </RequireAll>写在<Directory "> ... </Directory>
>内
<RequireAll>
    Require all granted
    Require ip 192.168.1.1 192.168.1.2
</RequireAll>

# 仅允许192.168.0.0/24网段的主机访问
<RequireAll>
    Require all granted
    Require ip 192.168.0.0/24
</RequireAll>

# 禁止192.168.1.3的主机访问（其他的都允许访问）
<RequireAll>
    Require all granted
    Require not ip 192.168.1.3
</RequireAll>

# 允许或拒绝所有访问（可以不用加容器<RequireAll> ... </RequireAll>，直接写在<Directory "> ... </Directory>内）
Require all granted # 允许所有访问
Require all denied # 拒绝所有访问
```

firewalld防火墙设置（HTTP默认端口80/TCP）：

```
firewall-cmd --add-service=http --permanent
firewall-cmd --reload
```

使用非80端口（如81）：

编辑 `/etc/httpd/conf/httpd.conf` 文件：

```
# 添加（或修改）一行
Listen 81
```

```
systemctl restart httpd
```

5.1. Apache httpd

添加对应端口的防火墙规则。

编辑 `/var/www/html/index.html`，创建一个HTML测试页：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Test Page
</div>
</body>
</html>
```

使用Web浏览器从客户端电脑访问，如果显示以下页面，则运行正常：



5.1.2. 使用Perl脚本

启用CGI执行并使用Perl脚本。

```
yum -y install perl perl-CGI # 安装Perl
```

默认情况下，在 `/var/www/cgi-bin` 目录下允许CGI。目录下可以使用Perl脚本。其下的所有文件都被处理为CGI。

```
grep -n "^ *ScriptAlias" /etc/httpd/conf/httpd.conf # 显示CGI的设置
```

```
247: ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

如果要在其他目录中允许CGI（例如，在 `/var/www/html/cgi-enabled` 允许），请如下配置：

5.1. Apache httpd

编辑 `/etc/httpd/conf.d/cgi-enabled.conf` 文件：

```
# 创建内容，将.cgi和.pl作为CGI脚本
<Directory "/var/www/html/cgi-enabled">
    Options +ExecCGI
    AddHandler cgi-script .cgi .pl
</Directory>
```

```
systemctl restart httpd # 重启httpd
```

如果启用了SELinux，并允许CGI在不是默认的目录下，如上面新加的目录，更改规则如下：

```
chcon -R -t httpd_sys_script_exec_t /var/www/html/cgi-enabled
semanage fcontext -a -t httpd_sys_script_exec_t /var/www/html/cgi-enabled
```

编辑 `/var/www/html/cgi-enabled/index.cgi` 文件，创建一个CGI测试页：

```
#!/usr/bin/perl

print "Content-type: text/html\n\n";
print "<html>\n<body>\n";
print "<div style=\"width: 100%; font-size: 40px; font-weight: bold; text-align: center;\">\n";
print "CGI Test Page";
print "\n</div>\n";
print "</body>\n</html>\n";
```

```
chmod 705 /var/www/html/cgi-enabled/index.cgi # 更改权限
```

使用Web浏览器从客户端电脑访问，如果显示以下页面，则运行正常：



5.1.3. 使用PHP脚本

```
yum -y install php php-mbstring php-pear # 安装PHP
```

编辑 `/etc/php.ini` 文件，将 `date.timezone =` 一行取消注释并设置为自己的时区：`date.timezone = "Asia/Shanghai"`

```
systemctl restart httpd
```

编辑 `/var/www/html/index.php` 文件，创建一个PHP测试页面：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
<?php
    print Date("Y/m/d");
?>
</div>
</body>
</html>
```

使用Web浏览器从客户端电脑访问，如果显示以下页面，则运行正常：



5.1.4. 使用Ruby脚本

配置httpd以使用Ruby脚本作为CGI。

```
yum -y install ruby # 安装Ruby
```

默认情况下，在`/var/www/cgi-bin`目录下允许CGI。目录下可以使用Perl脚本。其下的所有文件都被处理为CGI。

```
grep -n "^ *ScriptAlias" /etc/httpd/conf/httpd.conf # 显示CGI的设置
```

```
247: ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

如果要在其他目录中允许CGI（例如，在`/var/www/html/cgi-enabled`允许），请如下配置：

编辑`/etc/httpd/conf.d/cgi-enabled.conf`文件：

```
# 创建内容，将.rb作为CGI脚本
<Directory "/var/www/html/cgi-enabled">
    Options +ExecCGI
    AddHandler cgi-script .rb
</Directory>
```

```
systemctl restart httpd # 重启httpd
```

如果启用了SELinux，并允许CGI在不是默认的目录下，如上面新加的目录，更改规则如下：

5.1. Apache httpd

```
chcon -R -t httpd_sys_script_exec_t /var/www/html/cgi-enabled  
semanage fcontext -a -t httpd_sys_script_exec_t /var/www/html/cgi-enabled
```

编辑 `/var/www/html/cgi-enabled/index.rb` 文件，创建一个CGI测试页面：

```
#!/usr/bin/ruby  
  
print "Content-type: text/html\n\n"  
print "<html>\n<body>\n"  
print "<div style=\"width: 100%; font-size: 40px; font-weight: bold; text-align: center;\">\n"  
print "Ruby Script Test Page"  
print "\n</div>\n"  
print "</body>\n</html>\n"
```

```
chmod 705 /var/www/html/cgi-enabled/index.rb
```

使用Web浏览器从客户端电脑访问，如果显示以下页面，则运行正常：



5.1.5. 使用Python脚本

启用CGI执行并使用Python脚本。

```
yum -y install python # 安装Python
```

默认情况下，在 `/var/www/cgi-bin` 目录下允许CGI。目录下可以使用Perl脚本。其下的所有文件都被处理为CGI。

```
grep -n "^ *ScriptAlias" /etc/httpd/conf/httpd.conf # 显示CGI的设置
```

5.1. Apache httpd

```
247: ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

如果要在其他目录中允许CGI（例如，在`/var/www/html/cgi-enabled`允许），请如下配置：

编辑`/etc/httpd/conf.d/cgi-enabled.conf`文件：

```
# 创建内容，将.py作为CGI脚本
<Directory "/var/www/html/cgi-enabled">
    Options +ExecCGI
    AddHandler cgi-script .py
</Directory>
```

```
systemctl restart httpd # 重启httpd
```

如果启用了SELinux，并允许CGI在不是默认的目录下，如上面新加的目录，更改规则如下：

```
chcon -R -t httpd_sys_script_exec_t /var/www/html/cgi-enabled
semanage fcontext -a -t httpd_sys_script_exec_t /var/www/html/cgi-enabled
```

编辑`/var/www/html/cgi-enabled/index.py`文件，创建一个CGI测试页面：

```
#!/usr/bin/env python

print "Content-type: text/html\n\n"
print "<html>\n<body>"
print "<div style=\"width: 100%; font-size: 40px; font-weight: bold; text-align: center;\">"
print "Python Script Test Page"
print "</div>\n</body>\n</html>"
```

```
chmod 705 /var/www/html/cgi-enabled/index.py
```

使用Web浏览器从客户端电脑访问，如果显示以下页面，则运行正常：

5.1. Apache httpd



5.1.6. 启用Userdir

启用Userdir，用户可以使用此设置创建网站。

配置httpd，编辑 `/etc/httpd/conf.d/userdir.conf`：

```
# 注释掉下面一行
#UserDir disabled

# 取消下面一行注释
UserDir public_html

<Directory "/home/*public_html">
    # 修改下面两行
    AllowOverride All
    Options None
    Require method GET POST OPTIONS
</Directory>
```

```
systemctl restart httpd
```

创建一个用户的测试页：

```
mkdir public_html
chmod 711 /home/cent
chmod 755 /home/cent/public_html
```

编辑 `./public_html/index.html` 文件：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
UserDir Test Page
</div>
</body>
</html>
```

使用Web浏览器从客户端电脑访问，如果显示以下页面，则运行正常：



5.1.7. 虚拟主机

配置虚拟主机以使用多个域名，以下示例在域名为 `srv.world`，虚拟域名
为 `virtual.host`（根目录 `/home/cent/public_html`）的环境中设置。

编辑 `/etc/httpd/conf.d/vhost.conf` 文件：

5.1. Apache httpd

```
# 原始域名
<VirtualHost 80>
    DocumentRoot /var/www/html
    ServerName www.srv.world
</VirtualHost>

# 虚拟域名
<VirtualHost 80>
    DocumentRoot /home/cent/public_html
    ServerName www.virtual.host
    ServerAdmin webmaster@virtual.host
    ErrorLog logs/virtual.host-error_log # log目录/var/log/httpd
    CustomLog logs/virtual.host-access_log combined
    LogLevel warn
</VirtualHost>
```

个人建议的配置：

创建 /etc/httpd/vhosts 目录，将虚拟主机配置文件 *.conf 放在该目录

```
mkdir /var/www/tmp # 创建一个临时目录，下面不放任何内容
```

编辑 /etc/httpd/conf/httpd.conf 文件：

```
# 在最后一行“IncludeOptional conf.d/*.conf”下面添加以下内容
IncludeOptional vhosts/*.conf

<VirtualHost 80>
    DocumentRoot /var/www/tmp
    ServerName localhost
    ServerAlias *
    ServerSignature off
<Directory /var/www/tmp>
    AllowOverride None
    Options FollowSymLinks
    Require all denied
</Directory>
</VirtualHost>
```

```
systemctl restart httpd
```

5.1. Apache httpd

如果未设置其他虚拟主机，通过IP或域名访问Web服务器默认访问不到网页，然后再根据自己的需要设置虚拟主机。

编辑 `/etc/httpd/vhosts/vhost.conf` 文件：

```
<VirtualHost 80>
    DocumentRoot /home/cent/public_html
    ServerName www.virtual.host
    ServerAlias www2.virtual.host # 如果有多个域名指向同一路径，则每个
                                    # 域名写一行ServerAlias
    ServerAdmin webmaster@virtual.host
    ErrorLog logs/virtual.host-error_log
    CustomLog logs/virtual.host-access_log combined
    LogLevel warn
<Directory /home/cent/public_html>
    AllowOverride All
    Options FollowSymLinks
    Require all granted
</Directory>
</VirtualHost>
```

```
systemctl restart httpd
```

编辑 `~/public_html/virtual.php` 文件，创建测试页：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Virtual Host Test Page
</div>
</body>
</html>
```

使用Web浏览器从客户端电脑访问，如果显示以下页面，则运行正常：

5.1. Apache httpd



另CentOS6的虚拟主机配置，编辑 `/etc/httpd/conf/httpd.conf` 文件：

```
NameVirtualHost *:80 # 取消该行注释

# 在下面加入以下内容

<VirtualHost 80>
DocumentRoot /var/www/pma
ServerName www.virtual.host
ServerAlias www2.virtual.host
ServerAdmin webmaster@virtual.host
ErrorLog logs/virtual.host-error_log
CustomLog logs/virtual.host-access_log combined
</VirtualHost>

<VirtualHost 80>
DocumentRoot /var/www/tmp # 指向一个不存在或是空的文件夹
ServerName *
ServerAlias *
</VirtualHost>
```

5.1.8. 配置SSL

配置SSL以使用安全加密连接。

首先创建证书。

```
yum -y install mod_ssl
```

根据自己的环境配置SSL，编辑 `/etc/httpd/conf.d/ssl.conf` 文件：

5.1. Apache httpd

```
# 取消下面一行注释  
DocumentRoot "/var/www/html"  
  
# 取消注释并指定服务器名称  
ServerName www.srv.world:443  
  
# 更改下面一行  
SSLProtocol -All +TLSv1.2  
  
# 更改为在上一步创建的文件  
SSLCertificateFile /etc/pki/tls/certs/server.crt  
  
# 更改为在上一步创建的文件  
SSLCertificateKeyFile /etc/pki/tls/certs/server.key
```

```
systemctl restart httpd
```

firewalld防火墙设置（HTTPS默认端口443/TCP）：

```
firewall-cmd --add-service=https --permanent  
firewall-cmd --reload
```

使用非443端口（如1443）：

编辑 `/etc/httpd/conf.d/ssl.conf` 文件：

```
# 添加（或修改）一行  
Listen 1443 https
```

```
systemctl restart httpd
```

添加对应端口的防火墙规则。

配置虚拟主机：

编辑 `/etc/httpd/conf/httpd.conf` 文件：

```
# 在最后添加以下内容
<VirtualHost 443>
    DocumentRoot /var/www/tmp
    ServerName localhost:443
    ServerAlias *
    ServerSignature off
    SSLEngine On
    SSLProtocol -All +TLSv1.2
    SSLCertificateFile /etc/pki/tls/certs/server.crt
    SSLCertificateKeyFile /etc/pki/tls/certs/server.key
<Directory /var/www/tmp>
    AllowOverride None
    Options FollowSymLinks
    Require all denied
</Directory>
</VirtualHost>
```

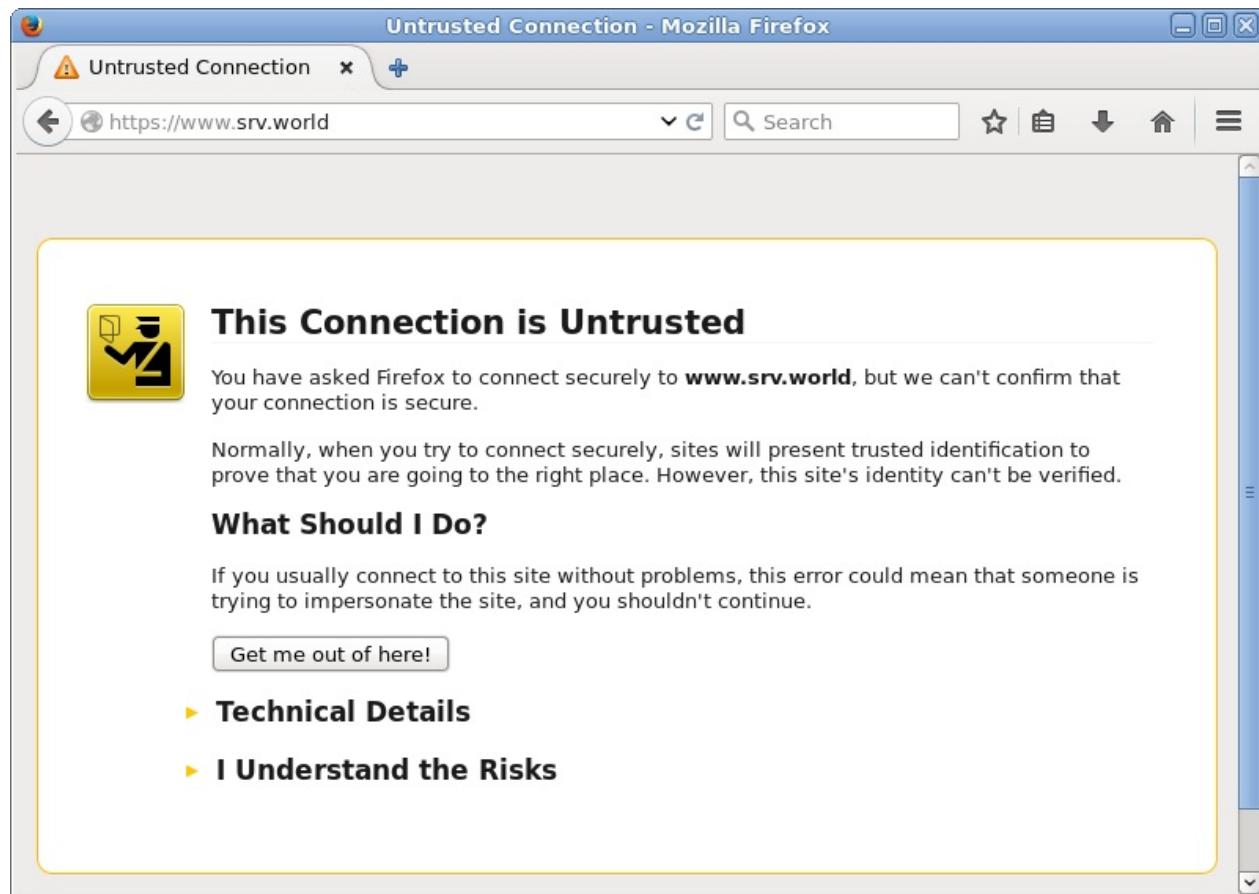
编辑 `/etc/httpd/conf.d/ssl.conf` 文件，将“VirtualHost”内容注释掉（可以先做个备份）。

编辑 `/etc/httpd/vhosts/vhost.conf` 文件：

```
<VirtualHost    80> # 将80端口的HTTP访问转到443端口的HTTPS
  ServerName www.virtual.host
  ServerAlias www2.virtual.host
  ServerAdmin webmaster@localhost
  ServerSignature Off
  RewriteEngine On
  RewriteCond %{HTTPS} !=on
  RewriteRule (.*) https:// %{HTTP_HOST}%{REQUEST_URI} [R=301,L]
  # RewriteRule ^ https:// %{SERVER_NAME}%{REQUEST_URI} [END, QSA
  ,R=permanent]
  # 如果是转到非443的HTTPS（如1443），%{HTTP_HOST}改为%{SERVER_NAME}
  }:1443
</VirtualHost>
<VirtualHost    443>
  DocumentRoot /home/cent/public_html
  ServerName www.virtual.host:443
  ServerAlias www2.virtual.host
  ServerAdmin webmaster@localhost
  ErrorLog logs/virtual.host-ssl_error_log
  TransferLog logs/virtual.host-ssl_access_log
  LogLevel warn
  SSLEngine On
  SSLProtocol -All +TLSv1.2
  SSLCertificateFile /etc/pki/tls/certs/server.crt
  SSLCertificateKeyFile /etc/pki/tls/certs/server.key
<Directory /home/cent/public_html>
  AllowOverride All
  Options FollowSymLinks
  Require all granted
</Directory>
</VirtualHost>
```

使用Web浏览器通过HTTPS从客户端计算机访问测试页。下面的示例是Fiorefix，显示以下屏幕，因为证书是自己创建的，但它没有问题，继续下一步：

5.1. Apache httpd



访问成功：



5.1.9. 启用基本身份验证

启用基本身份验证以限制特定网页上的访问。

以在目录 /var/www/html/auth-basic 下设置基本身份验证设置为例：

编辑 /etc/httpd/conf.d/auth_basic.conf 文件：

5.1. Apache httpd

```
# 创建
<Directory /var/www/html/auth-basic>
    AuthType Basic
    AuthName "Basic Authentication"
    AuthUserFile /etc/httpd/conf/.htpasswd
    require valid-user
</Directory>
```

```
htpasswd -c /etc/httpd/conf/.htpasswd cent # 添加用户：使用 -c 创建
一个新文件（添加 -c 选项仅用于初始注册）
```

```
New password: # 设置密码
Re-type new password: # 确认密码
Adding password for user cent
```

```
systemctl restart httpd
```

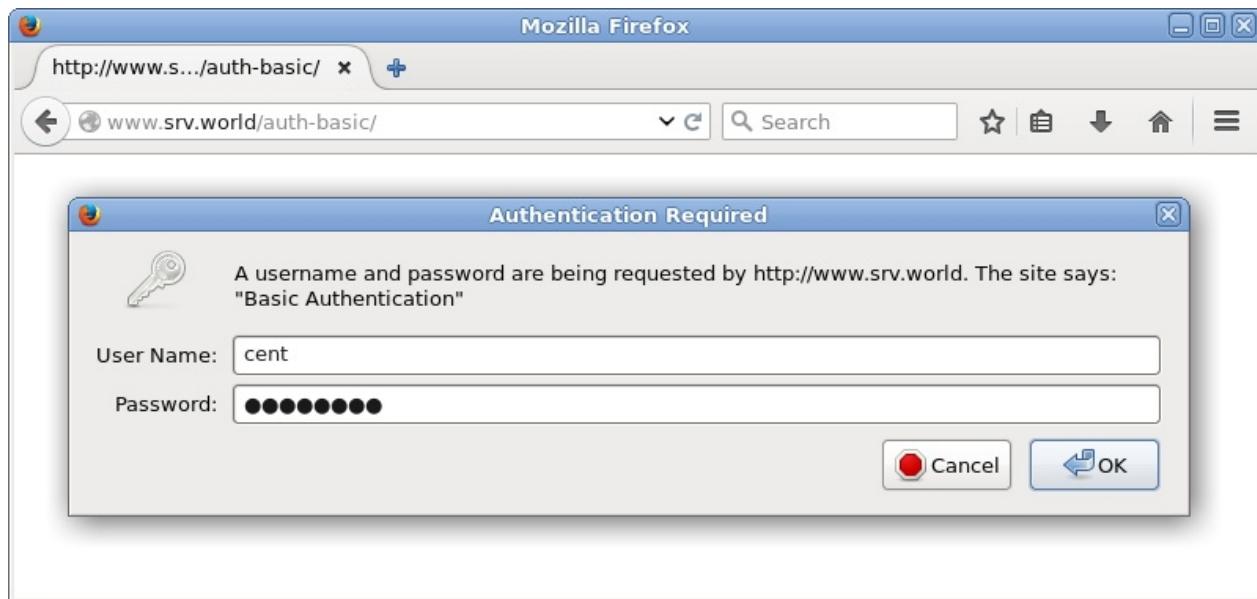
```
mkdir /var/www/html/auth-basic
```

编辑 /var/www/html/auth-basic/index.html 文件，创建测试页：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Test Page for Basic Auth
</div>
</body>
</html>
```

使用Web浏览器从客户端电脑访问，如下所示需要认证，使用上面添加的用户密码验证：

5.1. Apache httpd



访问成功：



5.1.10. 基本身份验证 + PAM

限制特定网页上的访问，并使用系统用户通过SSL连接进行身份验证。

先配置SSL

以在 /var/www/html/auth-pam 目录下设置基本认证为例：

```
yum --enablerepo=epel -y install mod_authnz_external pwauth # 从  
EPEL安装
```

编辑 /etc/httpd/conf.d/authnz_external.conf 文件，创建测试页：

5.1. Apache httpd

```
# 将以下内容添加到最后
<Directory /var/www/html/auth-pam>
    SSLRequireSSL
    AuthType Basic
    AuthName "PAM Authentication"
    AuthBasicProvider external
    AuthExternal pauth
    require valid-user
</Directory>
```

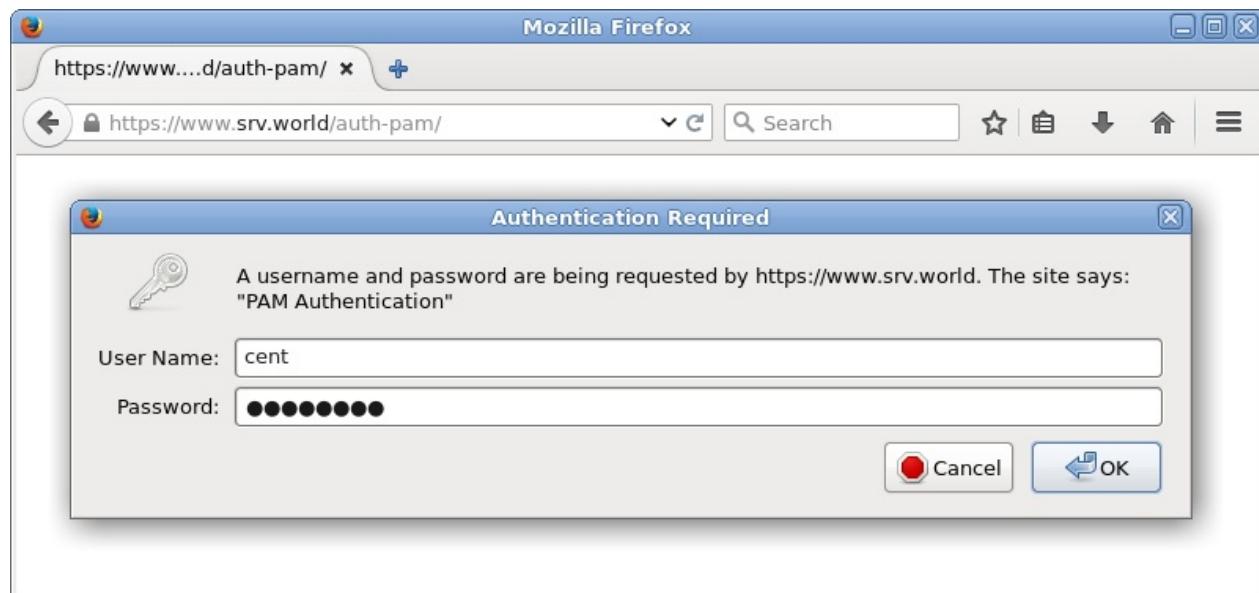
```
mkdir /var/www/html/auth-pam
```

编辑 /var/www/html/auth-pam/index.html 文件：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Test Page for PAM Auth
</div>
</body>
</html>
```

```
systemctl restart httpd
```

在客户端上使用Web浏览器访问测试页面，使用系统上的用户进行身份验证：



访问成功：



5.1.11. 基本身份验证 + LDAP

限制特定网页上的访问，并使用LDAP用户通过SSL连接进行身份验证。

[配置LDAP服务器](#)

[配置SSL](#)

以在 `/var/www/html/auth-ldap` 目录下设置基本认证为例：

```
yum -y install mod_ldap
```

编辑 `/etc/httpd/conf.d/auth_ldap.conf` 文件：

```
# 将以下内容添加到最后
<Directory /var/www/html/auth-ldap>
    SSLRequireSSL
    AuthName "LDAP Authentication"
    AuthType Basic
    AuthBasicProvider ldap
    AuthLDAPURL ldap://dlp.srv.world/dc=srv,dc=world?uid?sub?(objectClass=*)
    Require ldap-filter objectClass=posixAccount
</Directory>
```

```
mkdir /var/www/html/auth-ldap
```

5.1. Apache httpd

编辑 `/var/www/html/auth-ldap/index.html` 文件，创建测试页：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Test Page for LDAP Auth
</div>
</body>
</html>
```

在客户端上使用Web浏览器访问测试页面，使用LDAP上的用户进行身份验证：



访问成功：



5.1.12. 启用Kerberos身份验证

5.1. Apache httpd

启用Kerberos身份验证以限制对特定网页的访问。用户可以通过Windows Active Directory进行身份验证。因此，必须在LAN中运行Windows Active Directory。

本例基于下面的环境：

Domain Server	:	Windows Server 2012 R2
Domain Name	:	FD3S01
Realm	:	SRV.WORLD
Hostname	:	fd3s.srv.world

配置SSL

以在 `/var/www/html/auth-kerberos` 目录下设置Kerberos身份验证为例：

```
yum -y install mod_auth_kerb
```

编辑 `/etc/krb5.conf` 文件：

```
# 取消注释并更改为Realm名称
default_realm = SRV.WORLD

# 在[realms]下添加以下内容
[realms]
    SRV.WORLD = {
        kdc = fd3s.srv.world
        admin_server = fd3s.srv.world
    }

# 在[domain_realm]下添加以下内容
[domain_realm]
    .srv.world = SRV.WORLD
    srv.world = SRV.WORLD
```

```
echo "HTTP/fd3s.srv.world@SRV.WORLD" >
/etc/httpd/conf.d/krb5.keytab # 创建keytab，格式 HTTP/[AD's hostname
or IP address]@[Realm name]
```

编辑 `/etc/httpd/conf.d/auth_kerberos.conf` 文件：

5.1. Apache httpd

```
<Directory /var/www/html/auth-kerberos>
    SSLRequireSSL
    AuthType Kerberos
    AuthName "Kerberos Authentication"
    KrbAuthRealms SRV.WORLD
    Krb5Keytab /etc/httpd/conf.d/krb5.keytab
    KrbMethodNegotiate Off
    KrbSaveCredentials Off
    KrbVerifyKDC Off
    Require valid-user
</Directory>
```

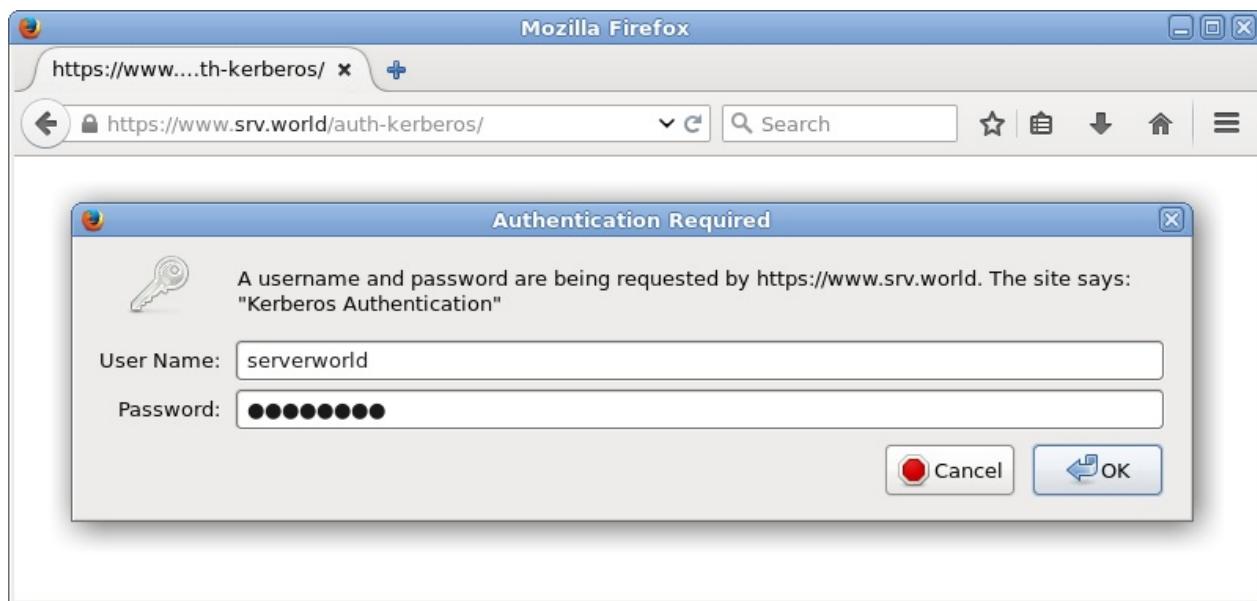
```
mkdir /var/www/html/auth-kerberos
```

编辑 /var/www/html/auth-kerberos/index.html 文件，创建测试页：

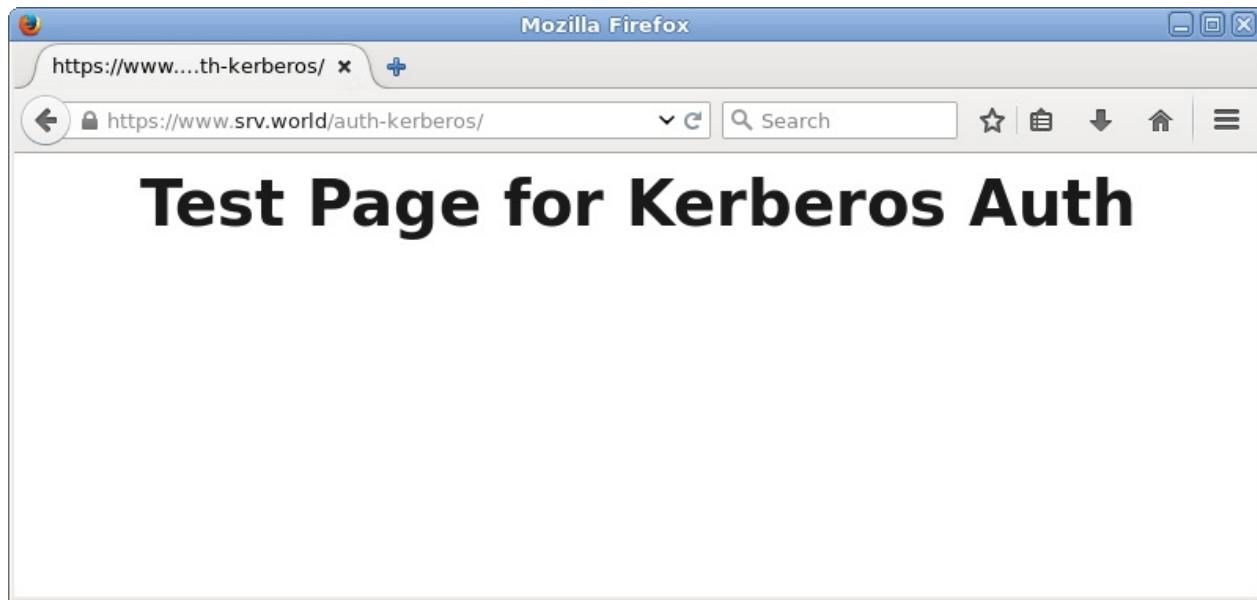
```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Test Page for Kerberos Auth
</div>
</body>
</html>
```

在客户端上使用Web浏览器访问测试页面，使用Active Directory上的用户进行身份验证：

5.1. Apache httpd



访问成功：



5.1.13. 使用WebDAV

这是使用SSL连接配置WebDAV设置的示例。

配置SSL

例如，创建一个目录 webdav ，使它仅可以通过SSL连接到WebDAV目录：

```
mkdir /home/webdav  
chown apache. /home/webdav  
chmod 770 /home/webdav
```

5.1. Apache httpd

编辑 `/etc/httpd/conf.d/webdav.conf` 文件：

```
DavLockDB "/tmp/DavLock"
Alias /webdav /home/webdav
<Location /webdav>
    DAV On
    SSLRequireSSL
    Options None
    AuthType Basic
    AuthName WebDAV
    AuthUserFile /etc/httpd/conf/.htpasswd
    <RequireAny>
        Require method GET POST OPTIONS
        Require valid-user
    </RequireAny>
</Location>
```

`htpasswd -c /etc/httpd/conf/.htpasswd cent` # 添加用户：使用 `-c` 创建一个新文件（添加 `-c` 选项仅用于初始注册）

```
New password: # 设置密码
Re-type new password: # 确认密码
Adding password for user cent
```

`systemctl restart httpd`

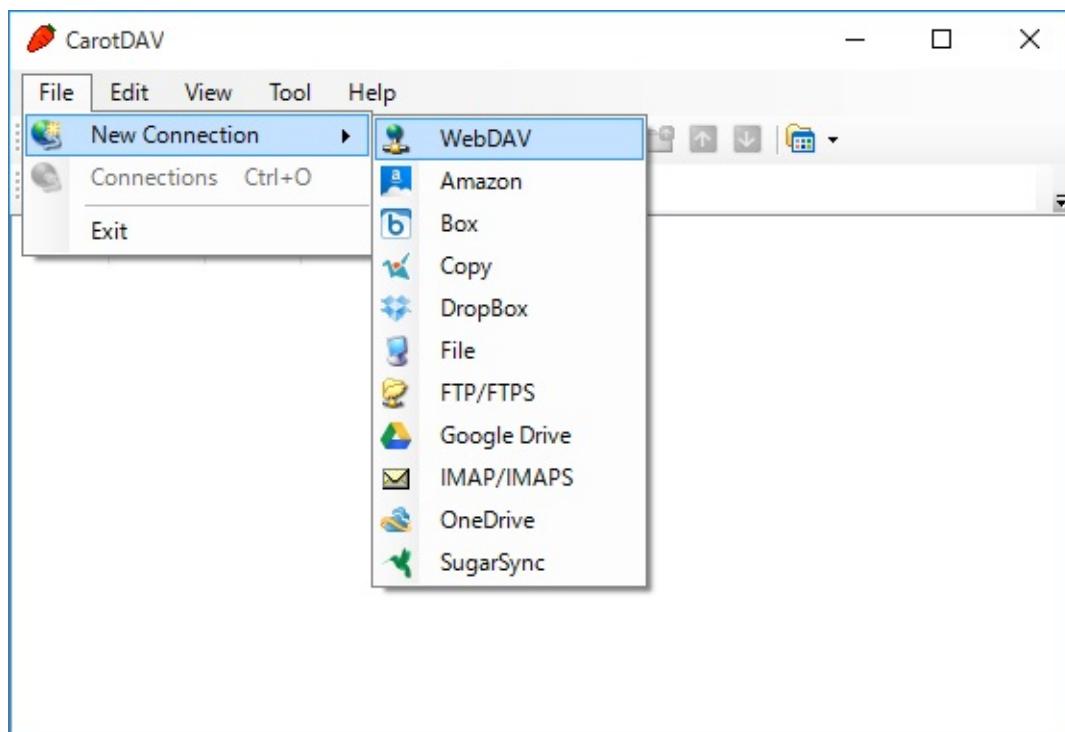
如果启用了SELinux，更改规则如下：

```
chcon -R -t httpd_sys_rw_content_t /home/webdav
semanage fcontext -a -t httpd_sys_rw_content_t /home/webdav
```

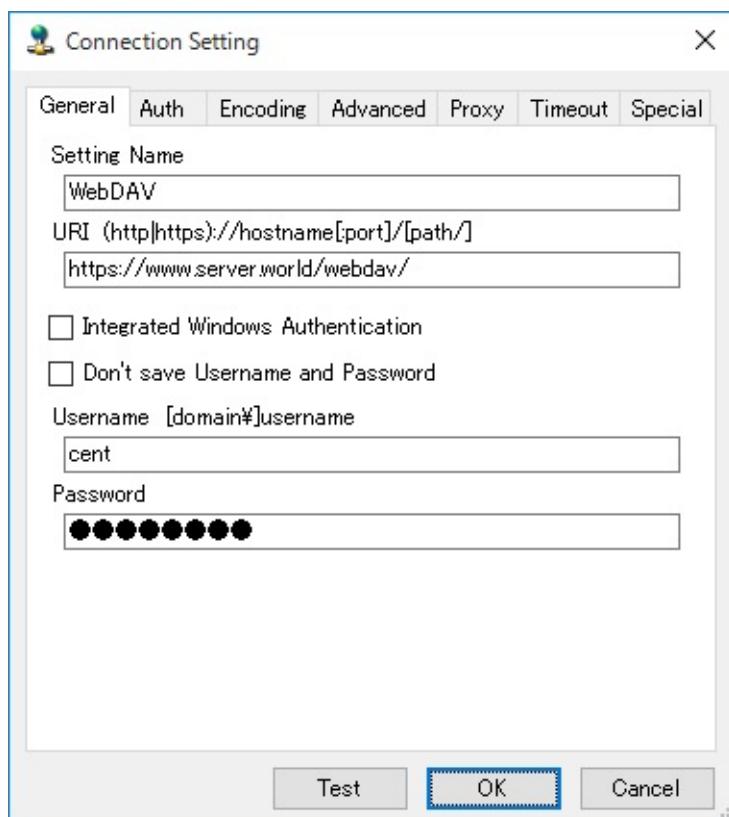
这里是PC上的WebDAV客户端设置（Windows 10）。

下载“[CarotDAV](#)”，这是一个免费的WebDAV客户端。安装并启动CarotDAV，显示以下屏幕，单击“File”按钮并选择“WebDAV”：

5.1. Apache httpd

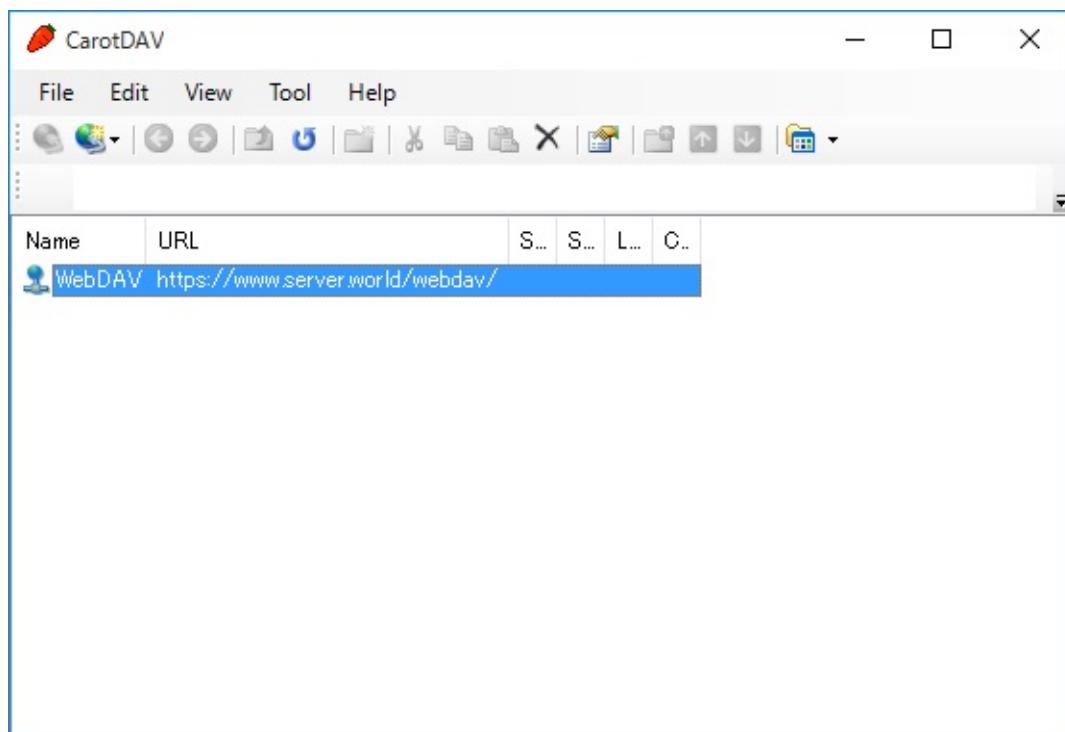


在“Setting Name”字段中输入任意名称，在“URI”字段中输入 服务器名称/ webdav 目录，并输入用户名和密码，如下所示：

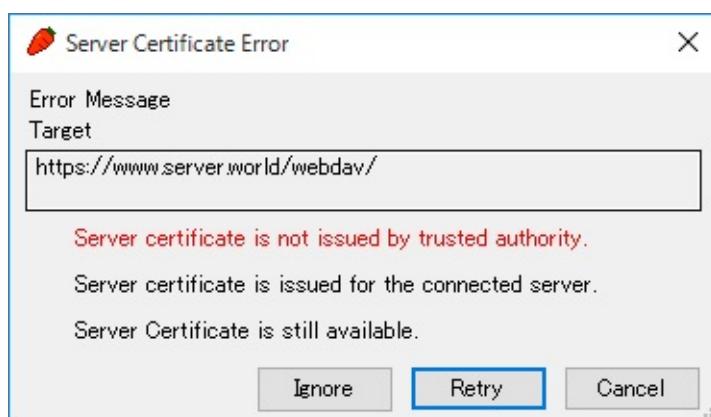


配置添加如下，点击它连接到服务器：

5.1. Apache httpd

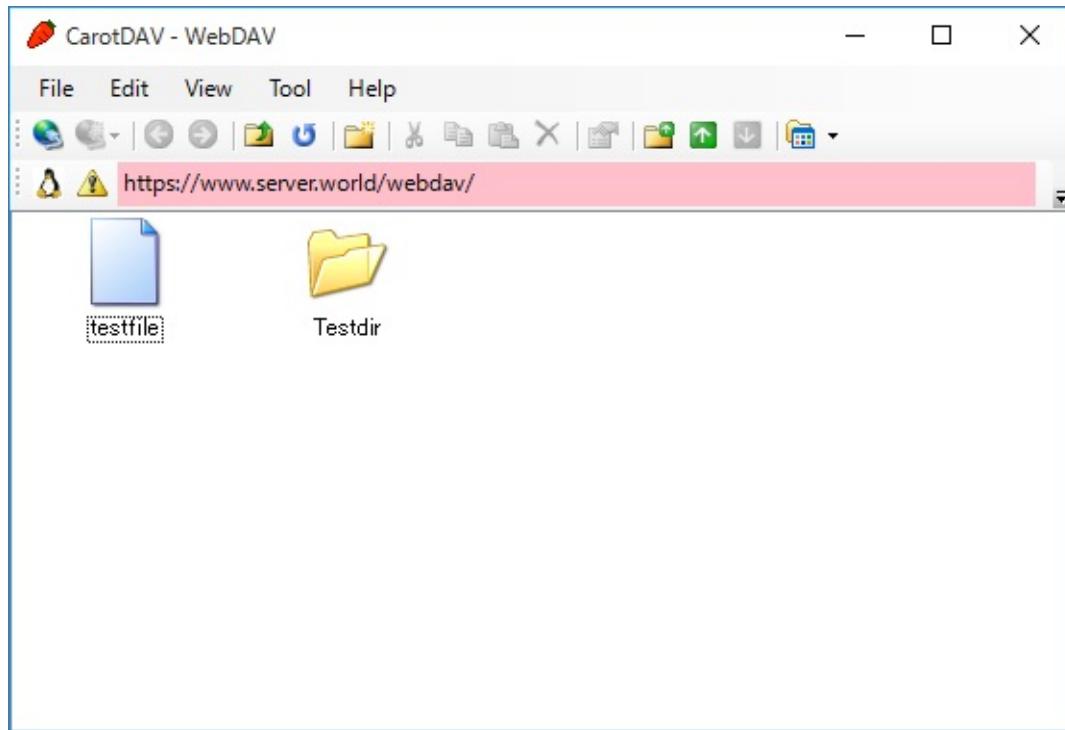


警告如下所示，因为电脑上没有安装SSL证书，这没有问题，点击“Ignore”，然后下一步：



访问成功：

5.1. Apache httpd



5.1.14. Perl + mod_perl

安装mod_perl使Perl脚本速度更快。

```
yum --enablerepo=epel -y install mod_perl # 从EPEL安装
```

配置PerlRun模式，总是将Perl解释器放在内存上：

编辑 /etc/httpd/conf.d/perl.conf 文件：

5.1. Apache httpd

```
# 取消注释（检查代码并向日志输出警告）
PerlSwitches -w

# 取消注释
PerlSwitches -T

# 取消注释如下
Alias /perl /var/www/perl
<Directory /var/www/perl> # mod_perl环境的目录
    SetHandler perl-script # 在此目录下将文件作为perl脚本处理
#    AddHandler perl-script .cgi # 如果不想将所有文件作为CGI处理，则
设置特定扩展
#    PerlResponseHandler ModPerl::Registry
    PerlResponseHandler ModPerl::PerlRun # 指定PerlRun模式
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Directory>

# 取消注释和添加以下内容
<Location /perl-status>
    SetHandler perl-script
    PerlResponseHandler Apache2::Status
    Require ip 127.0.0.1 10.0.0.0/24 # 添加访问权限
#    Order deny,allow
#    Deny from all
#    Allow from .example.com
</Location>
```

```
systemctl restart httpd
```

创建测试脚本以确保设置没有问题：

```
mkdir /var/www/perl
```

编辑 /var/www/perl/test-mod_perl.cgi 文件：

```
#!/usr/bin/perl

use strict;
use warnings;

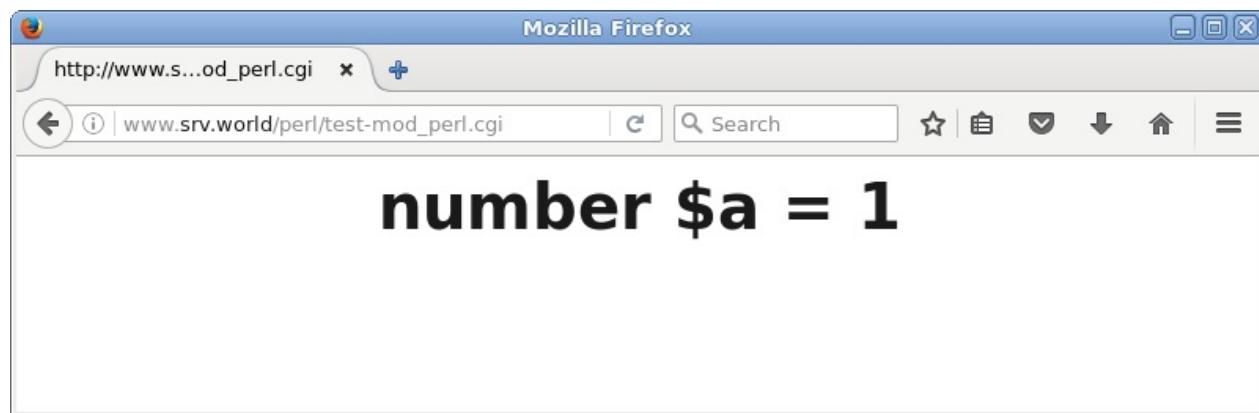
print "Content-type: text/html\n\n";
print "<html>\n<body>\n";
print "<div style=\"width:100%; font-size:40px; font-weight:bold
; text-align:center;\">";
my $a = 0;
&number();

print "</div>\n</body>\n</html>";

sub number {
    $a++;
    print "number \$a = $a";
}
```

```
chmod 705 /var/www/perl/test-mod_perl.cgi
```

如果显示如下所示的结果，表示正常：



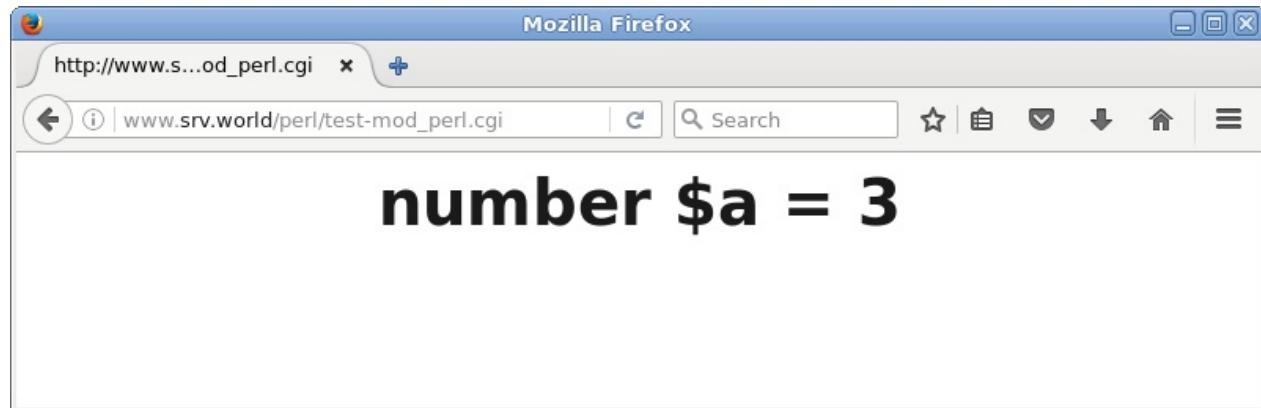
配置在内存上具有代码缓存的Registry模式：

编辑 /etc/httpd/conf.d/perl.conf 文件：

```
Alias /perl /var/www/perl
<Directory /var/www/perl>
    SetHandler perl-script
    PerlResponseHandler ModPerl::Registry # 取消注释
#    PerlResponseHandler ModPerl::PerlRun # 注释
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Directory>
```

```
systemctl restart httpd
```

访问上面示例的测试脚本，然后变量通过重新加载而增加，因为变量被高速缓存在内存上。所以有必要编辑 Registry 模式的代码：



编辑 `/var/www/perl/test-mod_perl.cgi` 文件：

5.1. Apache httpd

```
#!/usr/bin/perl

use strict;
use warnings;

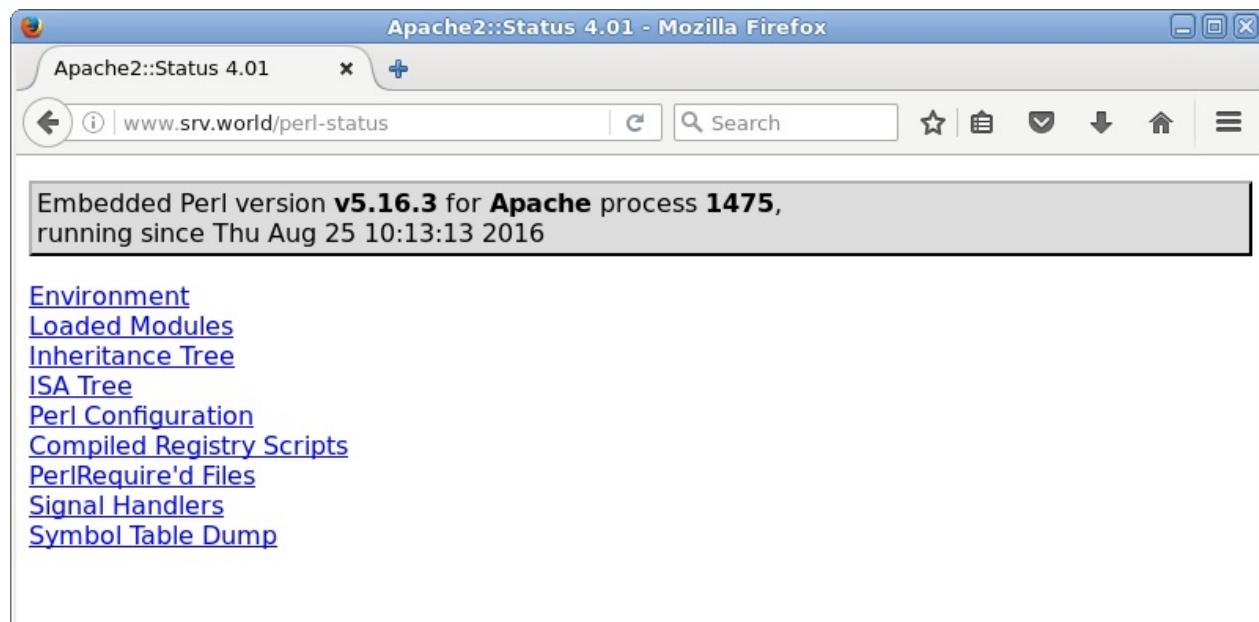
print "Content-type: text/html\n\n";
print "<html>\n<body>\n";
print "<div style=\"width:100%; font-size:40px; font-weight:bold
; text-align:center;\">";

my $a = 0;
&number($a);
print "</div>\n</body>\n</html>";

sub number {
    my($a) = @_;
    $a++;
    print "number \$a = $a";
}

}
```

可以访问 `http://(hostname or IP address)/perl-status` 查看mod_perl的状态



5.1. Apache httpd



5.1.15. PHP + PHP-FPM

先安装PHP

```
yum -y install php-fpm # 安装PHP-FPM
```

配置Apache httpd：

编辑 /etc/httpd/conf.d/php.conf 文件：

```
# 作如下更改
<FilesMatch \.php$>
#     SetHandler application/x-httpd-php
     SetHandler "proxy:fcgi://127.0.0.1:9000"
</FilesMatch>
```

```
systemctl start php-fpm
systemctl enable php-fpm
systemctl restart httpd
```

```
echo '<?php phpinfo(); ?>' > /var/www/html/info.php # 创建phpinfo
```

访问它，如果显示“FPM/FastCGI”则表示安装成功：

5.1. Apache httpd

System	Linux www.srv.world 3.10.0-327.22.2.el7.x86_64 #1 SMP Thu Jun 23 17:05:11 UTC 2016 x86_64
Build Date	Aug 11 2016 21:26:33
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/mbstring.ini, /etc/php.d/phar.ini, /etc/php.d posix.ini, /etc/php.d/sysvmsg.ini, /etc/php.d/sysvsem.ini, /etc/php.d/sysvshm.ini, /etc/php.d/wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zip.ini

5.1.16. Python + mod_wsgi

安装mod_wsgi（WSGI：Web Server Gateway Interface/Web服务器网关接口），使Python脚本更快。

```
yum -y install mod_wsgi # 安装mod_wsgi
```

编辑 `/etc/httpd/conf.d/wsgi.conf`，配置mod_wsgi：

示例为让可以访问 `/test_wsgi` 的后端是 `/var/www/html/test_wsgi.py`：

```
WSGIScriptAlias /test_wsgi /var/www/html/test_wsgi.py
```

```
systemctl restart httpd
```

创建在上面设置的测试脚本：

```
编辑 /var/www/html/test_wsgi.py 文件：
```

5.1. Apache httpd

```
def application(environ,start_response):
    status = '200 OK'
    html = '<html>\n' \
           '<body>\n' \
           '<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">\n' \
               'mod_wsgi Test Page\n' \
           '</div>\n' \
           '</body>\n' \
           '</html>\n'
    response_header = [('Content-type','text/html')]
    start_response(status,response_header)
    return [html]
```



如果使用Django

编辑 `/etc/httpd/conf.d/django.conf` 文件，配置：

示例为在“cent”用户的 `/home/cent/venv/testproject` 下配置“testapp”：

```
WSGIProcessGroup testapp
WSGIApplicationGroup main
WSGIFileWrapper mod_wsgi.WSGIFileWrapper

WSGIHandler testapp
WSGIPathInfoRoot /home/cent/venv/testproject/testproject/
WSGIPathRoot /home/cent/venv/testproject/testproject/
WSGIWrapper mod_wsgi.WSGIWrapper

WSGIDaemonProcess testapp python-path=/home/cent/venv/testproject:/home/cent/venv/lib/python2.7/site-packages
WSGIProcessGroup testapp
WSGIScriptAlias /django /home/cent/venv/testproject/testproject/wsgi.py

<Directory /home/cent/venv/testproject>
    Require all granted
</Directory>
```

```
systemctl restart httpd
```



5.1.17. 配置mod_proxy

5.1.17.1. 正向代理

启用mod_proxy模块以配置正向代理设置。

mod_proxy包含在httpd包中，默认启用，因此可以快速配置：

```
grep "mod_proxy" /etc/httpd/conf.modules.d/00-proxy.conf # 模块默认  
启用
```

```
LoadModule proxy_module modules/mod_proxy.so  
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so  
.....  
LoadModule proxy_http_module modules/mod_proxy_http.so  
LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
```

编辑 /etc/httpd/conf.d/f_proxy.conf 文件：

```
<IfModule mod_proxy.c>  
    # 正向代理功能On  
    ProxyRequests On  
    <Proxy *>  
        # 访问权限  
        Require ip 127.0.0.1 10.0.0.0/24  
    </Proxy>  
</IfModule>
```

编辑 /etc/httpd/conf/httpd.conf 文件：

5.1. Apache httpd

```
# 更改监听端口  
Listen 8080
```

```
systemctl restart httpd
```

firewalld防火墙设置，添加上面设置的端口（8080/TCP）：

```
firewall-cmd --add-port=8080/tcp --permanent  
firewall-cmd --reload
```

如果启用了SELinux，更改布尔值：

```
setsebool -P httpd_can_network_relay on
```

在客户端上配置代理客户端设置，并确保可以正常访问任何网站。

5.1.17.2. 反向代理

启用mod_proxy模块以配置反向代理设置。本例基于以下环境：

```
(1) www.srv.world      [10.0.0.31]      - Web Server#1  
(2) node01.srv.world   [10.0.0.51]      - Web Server#2
```

本例配置将(1)Web服务器的请求转发到(2)Web服务器。

mod_proxy包含在httpd包中，默认启用，因此可以快速配置：

```
grep "mod_proxy" /etc/httpd/conf.modules.d/00-proxy.conf # 模块默认  
启用
```

```
LoadModule proxy_module modules/mod_proxy.so  
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so  
....  
LoadModule proxy_http_module modules/mod_proxy_http.so  
LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
```

编辑 /etc/httpd/conf.d/r_proxy.conf 文件：

5.1. Apache httpd

```
<IfModule mod_proxy.c>
    ProxyRequests Off
    <Proxy *>
        Require all granted
    </Proxy>
    # 后端服务器和转发路径
    ProxyPass / http://node01.srv.world/
    ProxyPassReverse / http://node01.srv.world/
</IfModule>
```

```
systemctl restart httpd
```

访问前端服务器以确保后端服务器响应如下所示：



可以配置负载均衡设置：

- | | | |
|----------------------|-------------|----------------|
| (1) www.srv.world | [10.0.0.31] | - Web Server#1 |
| (2) node01.srv.world | [10.0.0.51] | - Web Server#2 |
| (3) node02.srv.world | [10.0.0.52] | - Web Server#3 |

本例配置将(1)Web服务器的http请求转发到(2)Web服务器和(3)Web服务器：

编辑 /etc/httpd/conf.d/r_proxy.conf 文件：

```
<IfModule mod_proxy.c>
    ProxyRequests Off
    <Proxy *>
        Require all granted
    </Proxy>
    # 指定使用“lbmethod”进行负载均衡的方式。也可以设置“bytraffic”。
    ProxyPass / balancer://cluster lbmethod=byrequests
    <proxy balancer://cluster>
        BalancerMember http://node01.srv.world/ loadfactor=1
        BalancerMember http://node02.srv.world/ loadfactor=1
    </proxy>
</IfModule>
```

```
systemctl restart httpd
```

访问前端服务器以确保后端服务器响应如下所示：



使用反向代理申请Let's Encrypt证书的示例：

```
<VirtualHost 80>
ServerAdmin webmaster@localhost
ServerName localhost
ServerAlias v.x.com
RewriteEngine On
RewriteCond %{HTTPS} !=On
RewriteRule (.*)
https:// %{SERVER_NAME}%{REQUEST_URI} [R=301,L]
</VirtualHost>

<VirtualHost 443>
DocumentRoot /var/www/html
ServerName localhost
ServerAlias v.x.com
ServerAdmin webmaster@localhost
ErrorLog logs/v-ssl_error_log
TransferLog logs/v-ssl_access_log
LogLevel warn
SSLEngine On
SSLProtocol -All +TLSv1.2
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
SSLProxyEngine on
ProxyRequests Off
<Directory /var/www/html>
AllowOverride All
Options FollowSymLinks
Require all granted
</Directory>
<Proxy *>
Require all granted
</Proxy>
ProxyPass /.well-known ! # 不转发/.well-known的请求
ProxyPass / http://node01.srv.world/ # 转发到的服务器
</VirtualHost>
```

5.1.18. 配置mod_proxy_wstunnel

启用mod_proxy_wstunnel模块以设置WebSocket代理。

5.1. Apache httpd

例如，对于在 `localhost:1337` 上侦听的应用程序（[示例应用程序来自这里](#)），配置httpd在`/chat`设置代理：

编辑 `/etc/httpd/conf.modules.d/00-proxy.conf` 文件：

```
# 添加到最后
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

编辑 `/etc/httpd/conf.d/wstunnel.conf` 文件：

```
ProxyRequests Off
<Proxy *>
    Require all granted
</Proxy>

ProxyPass /socket.io/ http://127.0.0.1:1337/socket.io/
ProxyPassReverse /socket.io/ http://127.0.0.1:1337/socket.io/

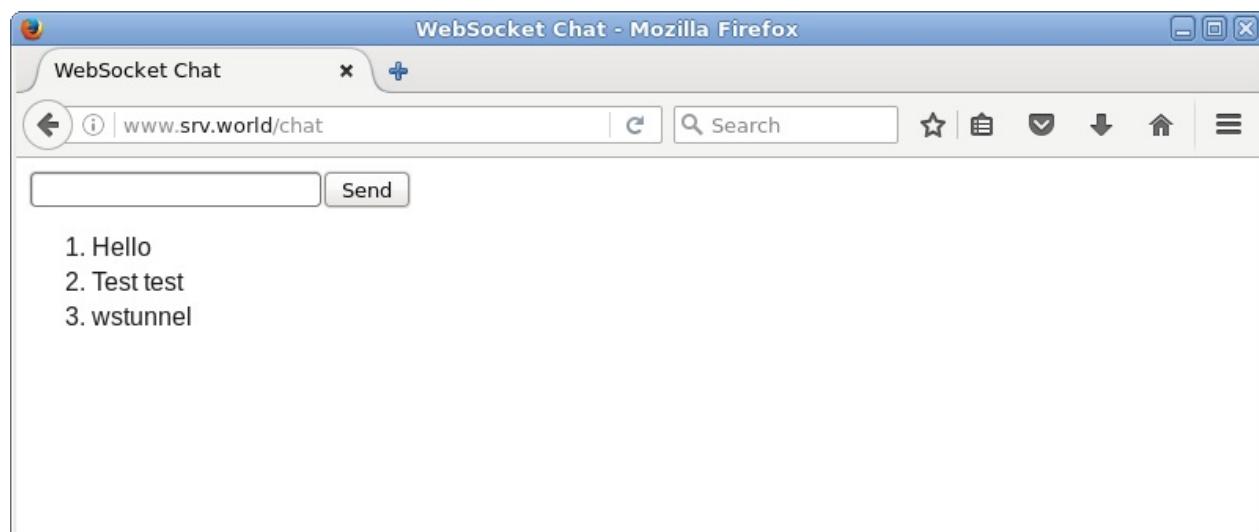
ProxyPass /chat http://127.0.0.1:1337/
ProxyPassReverse /chat http://127.0.0.1:1337/
```

```
systemctl restart httpd
```

如果启用了SELinux，更改规则如下：

```
semanage port -a -t http_port_t -p tcp 1337
```

访问示例应用程序以确保它在代理环境中正常工作：



下面是其它一些反向代理加WebSocket代理的示例：

[noVNC](#)

[Guacamole](#)

5.1.19. 配置mod_ratelimit

启用mod_ratelimit模块以限制客户端的带宽。

mod_ratelimit包含在httpd包中，因此可以快速配置。

编辑 `/etc/httpd/conf.modules.d/00-base.conf` 文件：

```
# 取消注释
LoadModule ratelimit_module modules/mod_ratelimit.so
```

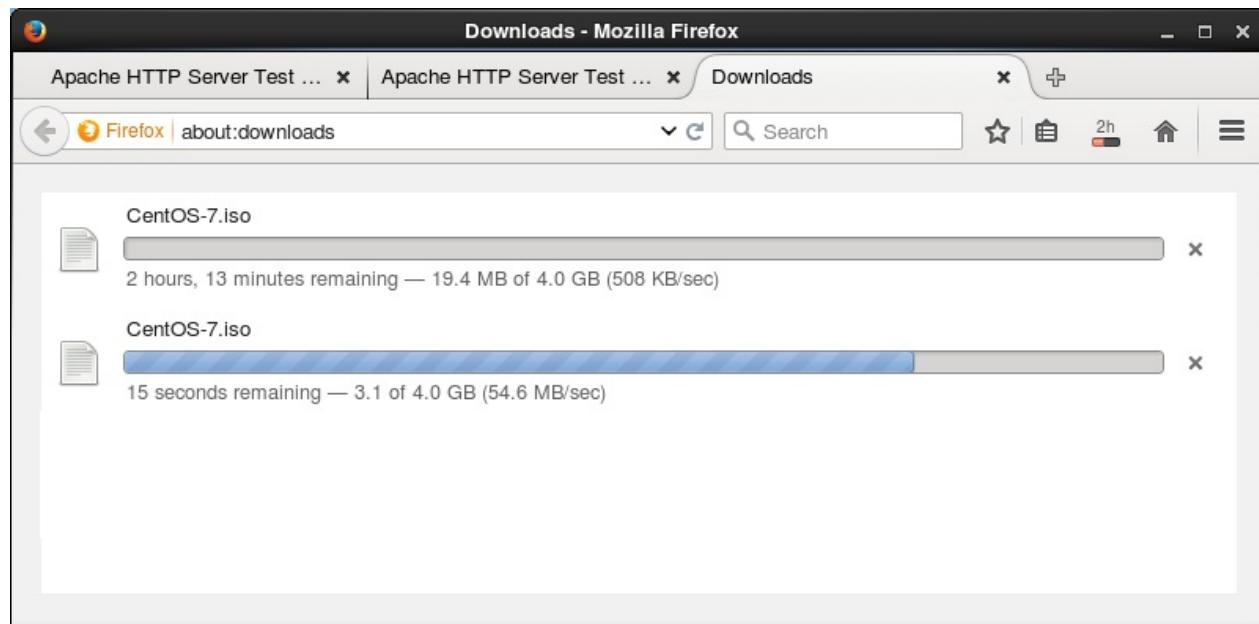
编辑 `/etc/httpd/conf.d/ratelimit.conf` 文件：

```
# 例如，在/download目录下限制带宽为500KB/秒
<IfModule mod_ratelimit.c>
    <Location /download>
        SetOutputFilter RATE_LIMIT
        SetEnv rate-limit 500
    </Location>
</IfModule>
```

```
systemctl restart httpd
```

访问位置以确保设置有效（上面是从限制的目录下载，下面是没有限制的）：

5.1. Apache httpd



5.1.20. 配置mod_limitipconn

使用mod_limitipconn限制每个IP地址的并发连接。

```
yum --enablerepo=epel -y install mod_limitipconn # 从EPEL安装
```

编辑 /etc/httpd/conf.d/limitipconn.conf 文件：

```
# 默认设置没有限制
MaxConnPerIP 0

# /limit目录设置
<Location /limit>
    # 限制并发连接3
    MaxConnPerIP 3
    # 如果MIME类型为“text/*”，则不适用上面规则
    NoIPLimit text/*
</Location>

# /limit2目录设置
<Location /limit2>
    # 限制并发连接2
    MaxConnPerIP 2
    # 如果MIME类型为“application/x-tar”，则不适用上面规则
    OnlyIPLimit application/x-tar
</Location>
```

5.1. Apache httpd

```
systemctl restart httpd
```

使用httpd-tools包中包含的命令“ab”验证是否正常工作，如下所示：

```
ab -n 10 -c 10 http://localhost/limit/index.html
```

```
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)....done

Server Software:           Apache/2.4.6
Server Hostname:          localhost
Server Port:              80

Document Path:            /limit/index.html
Document Length:          130 bytes

Concurrency Level:         10
Time taken for tests:    0.004 seconds
Complete requests:        10
Failed requests:          0
Write errors:              0
Total transferred:        3910 bytes
HTML transferred:         1300 bytes
Requests per second:      2223.21 [#/sec] (mean)
Time per request:         4.498 [ms] (mean)
Time per request:         0.450 [ms] (mean, across all concurrent
                         requests)
Transfer rate:            848.90 [Kbytes/sec] received
.
.
```

```
ab -n 10 -c 10 http://localhost/limit/test.gif
```

5.1. Apache httpd

```
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)....done

Server Software:        Apache/2.4.6
Server Hostname:        localhost
Server Port:            80

Document Path:          /limit/test.gif
Document Length:        228 bytes

Concurrency Level:      10
Time taken for tests:  0.005 seconds
Complete requests:     10
Failed requests:        7
    (Connect: 0, Receive: 0, Length: 7, Exceptions: 0)
Write errors:           0
Non-2xx responses:     7
Total transferred:     4838 bytes
HTML transferred:       2777 bytes
Requests per second:   2182.45 [#/sec] (mean)
Time per request:      4.582 [ms] (mean)
Time per request:      0.458 [ms] (mean, across all concurrent
requests)
Transfer rate:         1031.12 [Kbytes/sec] received
.....
.....
```

```
ab -n 10 -c 10 http://localhost/limit2/test.tar
```

```
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)....done

Server Software:        Apache/2.4.6
Server Hostname:       localhost
Server Port:          80

Document Path:         /limit2/test.tar
Document Length:      10240 bytes

Concurrency Level:     10
Time taken for tests: 0.006 seconds
Complete requests:    10
Failed requests:       8
          (Connect: 0, Receive: 0, Length: 8, Exceptions: 0)
Write errors:          0
Non-2xx responses:    8
Total transferred:    24900 bytes
HTML transferred:     22872 bytes
Requests per second:  1785.40 [#/sec] (mean)
Time per request:     5.601 [ms] (mean)
Time per request:     0.560 [ms] (mean, across all concurrent
                      requests)
Transfer rate:        4341.44 [Kbytes/sec] received
.
.
```

5.1.21. 配置mod_evasive

启用mod_evasive模块来防御DoS攻击等。

```
yum --enablerepo=epel -y install mod_evasive # 从EPEL安装
```

编辑 /etc/httpd/conf.d/mod_evasive.conf 文件：

5.1. Apache httpd

```
# 每页间隔相同页面的请求数量的阈值  
DOSPageCount      5  
  
# 每个站点间隔相同监听器上的同一客户端对任何对象的请求数量的阈值  
DOSSiteCount      50  
  
# 页计数阈值的时间间隔  
DOSPageInterval    1  
  
# 站点计数阈值的时间间隔  
DOSSiteInterval    1  
  
# 如果将客户端添加到阻止列表中，则客户端将被阻止的时间量（以秒为单位）  
DOSBlockingPeriod  300  
  
# 如果IP地址被列入黑名单，通知邮件地址  
DOSEmailNotify     root@localhost  
  
# 指定日志目录  
DOSLogDir          "/var/log/mod_evasive"
```

```
mkdir /var/log/mod_evasive  
chown apache. /var/log/mod_evasive  
systemctl restart httpd
```

使用RPM软件包中包含的测试工具进行测试：

```
perl /usr/share/doc/mod_evasive-*/test.pl
```

5.1. Apache httpd

```
HTTP/1.1 200 OK
.....
.....
HTTP/1.1 403 Forbidden # 如果被阻止，转到“403 Forbidden”
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
.....
.....
HTTP/1.1 403 Forbidden
```

```
ll /var/log/mod_evasive
```

```
total 4
-rw-r--r-- 1 apache apache 5 Aug  5 15:42 dos-127.0.0.1
```

如果设置了通知，则发送如下：

```
mail
```

```
Heirloom Mail version 12.5 7/5/10. Type ? for help.  
"/var/spool/mail/root": 1 message 1 new  
>N 1 Apache Wed Aug 3 19:42 20/673  
& 1  
Message 1:  
From apache@www.srv.world Wed Aug 3 19:42:55 2015  
Return-Path: <apache@www.srv.world>  
X-Original-To: root@localhost  
Delivered-To: root@localhost.srv.world  
Date: Wed, 05 Aug 2015 15:42:54 +0900  
To: root@localhost.srv.world  
User-Agent: Heirloom mailx 12.5 7/5/10  
Content-Type: text/plain; charset=us-ascii  
From: apache@www.srv.world (Apache)  
Status: R  
  
To: root@localhost  
Subject: HTTP BLACKLIST 127.0.0.1  
  
mod_evasive HTTP Blacklisted 127.0.0.1
```

5.1.22. 配置mod_security

使用mod_security模块配置Web应用程序防火墙（WAF）。

```
yum -y install mod_security
```

安装后，配置文件放在下面的目录中，并且设置被启用。其中有一些设置，还可以添加自己的规则：

```
cat /etc/httpd/conf.d/mod_security.conf
```

```
<IfModule mod_security2.c>
    # ModSecurity Core Rules Set configuration
    IncludeOptional modsecurity.d/*.conf
    IncludeOptional modsecurity.d/activated_rules/*.conf

    # Default recommended configuration
    SecRuleEngine On
    SecRequestBodyAccess On
    SecRule REQUEST_HEADERS:Content-Type "text/xml" \
        "id:'200000',phase:1,t:none,t:lowercase,pass,nolog,ctl:
    requestBodyProcessor=XML"

    .....
    .....
# 如果不希望在匹配规则时阻止请求，指定对参数“SecRuleEngine DetectionOnl
y”的更改
```

可以如下编写规则：

```
SecRule VARIABLES OPERATOR [ACTIONS]
```

每个参数有许多种值，请参考[官方文档](#)。

举例设置一些规则并验证它正常工作：

```
编辑 /etc/httpd/modsecurity.d/activated_rules/rules-01.conf 文件：
```

5.1. Apache httpd

```
# 匹配规则时的默认操作
SecDefaultAction "phase:2,deny,log,status:406"

# “etc/passwd”包含在请求URI中
SecRule REQUEST_URI "etc/passwd" "id:'500001'"

# “..”包含在请求URI中
SecRule REQUEST_URI "\.\./" "id:'500002'"

# “<SCRIPT”包含在参数中
SecRule ARGS "<[Ss][Cc][Rr][Ii][Pp][Tt]" "id:'500003'"

# “SELECT FROM”包含在参数中
SecRule ARGS "[Ss][Ee][Ll][Ee][Cc][Tt][[:space:]]+[Ff][Rr][Oo][Mm]" "id:'500004'"
```

```
systemctl restart httpd
```

访问包含您设置的字词的URI，并验证其是否正常工作：



mod_security的日志放在如下所示的目录中：

```
cat /var/log/httpd/modsec_audit.log
```

```
--75d36531-A--  
[28/Oct/2015:13:52:52 +0900] VjBUpAKZ9yAFgyhKj8zyyAAAAAE 10.0.0.  
108 53545 10.0.0.31 80  
--75d36531-B--  
GET /?.../etc/passwd HTTP/1.1  
Host: www.srv.world  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100  
101 Firefox/38.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/  
*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
  
--75d36531-F--  
HTTP/1.1 406 Not Acceptable  
Content-Length: 251  
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/html; charset=iso-8859-1  
  
--75d36531-E--  
  
--75d36531-H--  
Message: Access denied with code 406 (phase 2). Pattern match "e  
tc/passwd"  
at REQUEST_URI. [file "/etc/httpd/modsecurity.d/activated_rules/  
rules-01.conf"] [line "3"] [id "500001"]  
Action: Intercepted (phase 2)  
Stopwatch: 1446007972909468 1333 (- - -)  
Stopwatch2: 1446007972909468 1333; combined=418, p1=395, p2=17,  
p3=0, p4=0, p5=6, sr=116, sw=0, l=0, gc=0  
Response-Body-Transformed: Dechunked  
Producer: ModSecurity for Apache/2.7.3 (http://www.modsecurity.org/); OWASP CRS/2.2.6.  
Server: Apache/2.4.6 (CentOS)  
Engine-Mode: "ENABLED"  
  
--75d36531-Z--
```

5.1. Apache httpd

普通规则由官方存储库提供，很容易应用。但可能需要自定义它们让自己的网站不阻止必要的请求：

```
yum -y install mod_security_crs
```

规则放置如下，它们被链接到目

录 /etc/httpd/modsecurity.d/activated_rules :

```
ll /usr/lib/modsecurity.d/base_rules
```

```
total 332
-rw-r--r-- 1 root root 1980 Jun 10 2014 modsecurity_35_bad_robots.data
-rw-r--r-- 1 root root 386 Jun 10 2014 modsecurity_35_scanner.s.data
-rw-r--r-- 1 root root 3928 Jun 10 2014 modsecurity_40_generic_attacks.data
-rw-r--r-- 1 root root 2610 Jun 10 2014 modsecurity_41_sql_injection_attacks.data
-rw-r--r-- 1 root root 2224 Jun 10 2014 modsecurity_50_outbound.data
-rw-r--r-- 1 root root 56714 Jun 10 2014 modsecurity_50_outbound_malware.data
-rw-r--r-- 1 root root 22861 Jun 10 2014 modsecurity_crs_20_protocol_violations.conf
-rw-r--r-- 1 root root 6915 Jun 10 2014 modsecurity_crs_21_protocol_anomalies.conf
-rw-r--r-- 1 root root 3792 Jun 10 2014 modsecurity_crs_23_request_limits.conf
-rw-r--r-- 1 root root 6933 Jun 10 2014 modsecurity_crs_30_http_policy.conf
-rw-r--r-- 1 root root 5394 Jun 10 2014 modsecurity_crs_35_bad_robots.conf
-rw-r--r-- 1 root root 19157 Jun 10 2014 modsecurity_crs_40_generic_attacks.conf
-rw-r--r-- 1 root root 43961 Jun 10 2014 modsecurity_crs_41_sql_injection_attacks.conf
-rw-r--r-- 1 root root 87470 Jun 10 2014 modsecurity_crs_41_xss_attacks.conf
-rw-r--r-- 1 root root 1795 Jun 10 2014 modsecurity_crs_42_tight_security.conf
-rw-r--r-- 1 root root 3660 Jun 10 2014 modsecurity_crs_45_tro
```

5.1. Apache httpd

```
jans.conf
-rw-r--r-- 1 root root 2253 Jun 10 2014 modsecurity_crs_47_com
mon_exceptions.conf
-rw-r--r-- 1 root root 2787 Jun 10 2014 modsecurity_crs_48_loc
al_exceptions.conf.example
-rw-r--r-- 1 root root 1835 Jun 10 2014 modsecurity_crs_49_inb
ound_blocking.conf
-rw-r--r-- 1 root root 22314 Jun 10 2014 modsecurity_crs_50_out
bound.conf
-rw-r--r-- 1 root root 1448 Jun 10 2014 modsecurity_crs_59_out
bound_blocking.conf
-rw-r--r-- 1 root root 2674 Jun 10 2014 modsecurity_crs_60_cor
relation.conf
```

5.2. Nginx

5.2.1. 安装Nginx

```
yum --enablerepo=epel -y install nginx # 从EPEL安装Nginx
```

配置基本设置，编辑 `/etc/nginx/nginx.conf` 文件：

```
# 更改主机名  
server_name www.srv.world;
```

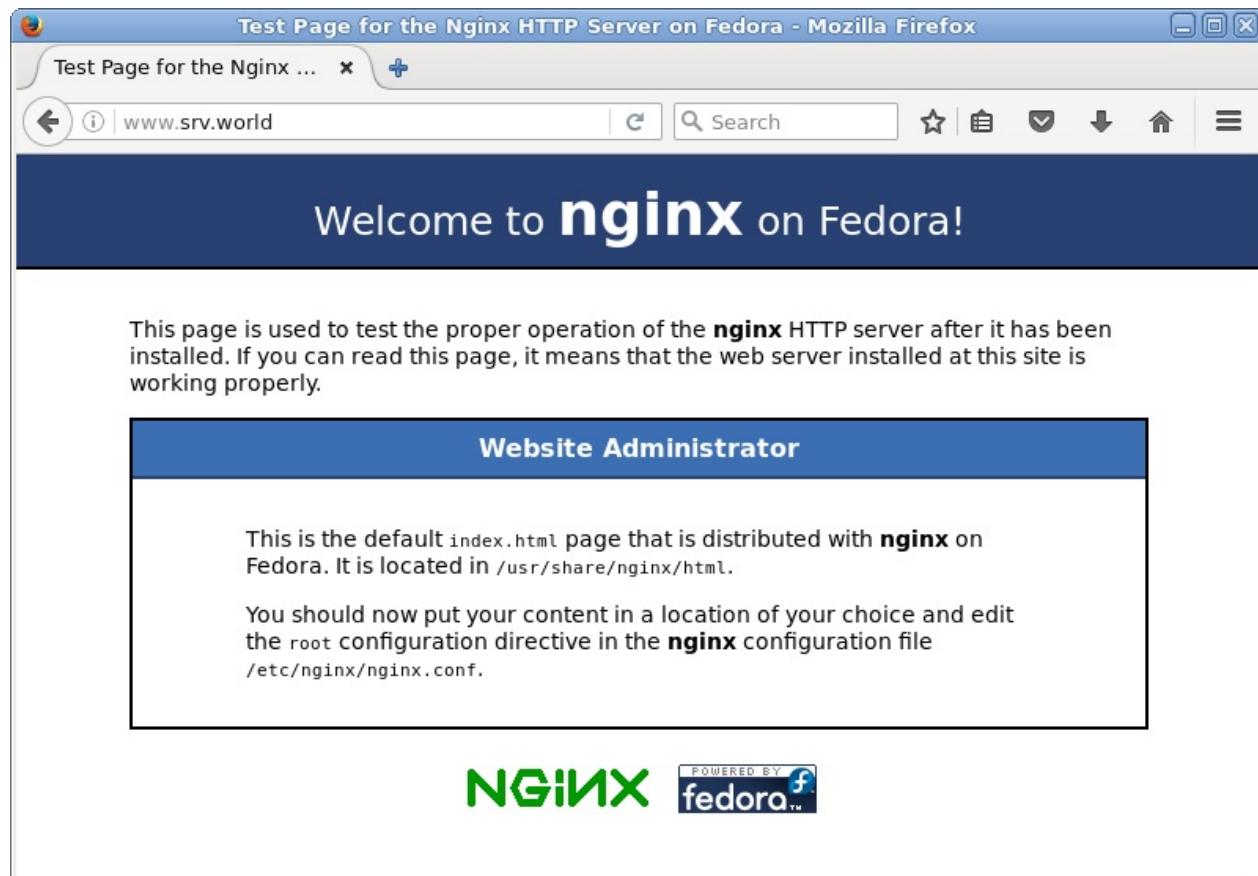
```
systemctl start nginx  
systemctl enable nginx
```

firewalld防火墙设置（HTTP默认端口80/TCP）：

```
firewall-cmd --add-service=http --permanent  
firewall-cmd --reload
```

使用Web浏览器从客户端电脑访问Nginx的默认页面，如果显示以下页面，则运行正常：

5.2. Nginx



5.2.2. 虚拟主机

本例配置其他域名为“virtual.host”：

编辑 `/etc/nginx/conf.d/virtual.host.conf` 文件：

```
server {
    listen      80;
    server_name www.virtual.host;

    location / {
        root   /usr/share/nginx/virtual.host;
        index index.html index.htm;
    }
}
```

编辑 `/usr/share/nginx/virtual.host/index.html` 文件，创建测试页以确保其正常工作：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Nginx Virtual Host Test Page
</div>
</body>
</html>
```



5.2.3. 启用Userdir

启用普通用户的Userdir在主目录中打开其网站。

编辑 `/etc/nginx/nginx.conf` 文件：

```
# 在“server”部分添加以下内容
location ~ ^/~(.+?)(/.*)?$ {
    alias /home/$1/public_html$2;
    index index.html index.htm;
    autoindex on;
}
```

```
systemctl restart nginx
```

使用普通用户创建测试页以确保其正常工作：

```
chmod 711 /home/cent
mkdir ~/public_html
chmod 755 ~/public_html
```

5.2. Nginx

编辑 `~/public_html/index.html` 文件：

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">
Nginx UserDir Test Page
</div>
</body>
</html>
```



5.2.4. 配置SSL

配置SSL以使用安全加密连接。

[首先创建证书](#)。

编辑 `/etc/nginx/nginx.conf` 文件：

5.2. Nginx

```
# 在“server”部分添加以下内容
server {
    listen      80 default_server;
    listen      [::]:80 default_server;
    listen      443 ssl;
    server_name www.srv.world;
    root        /usr/share/nginx/html;

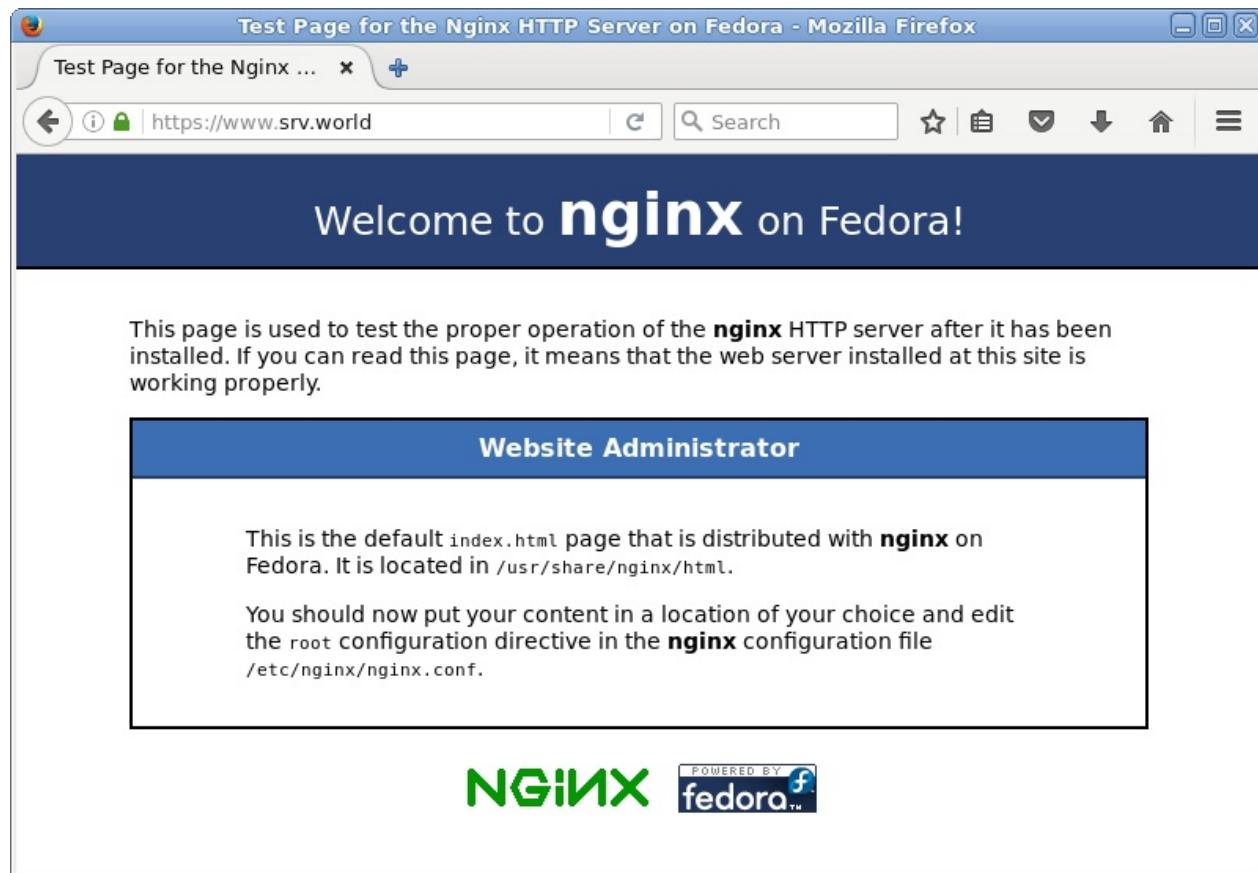
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    ssl_ciphers ECDHE+RSAGCM:ECDH+AESGCM:DH+AESGCM:ECDH+AES2
56:DH+AES256:ECDH+AES128:DH+AES:!aNULL!eNULL:!EXPORT:!DES:!3DES:
!MD5:!DSS;
    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;
```

firewalld防火墙设置（HTTPS默认端口443/TCP）：

```
firewall-cmd --add-service=https --permanent
firewall-cmd --reload
```

使用HTTPS访问默认页面，以确保其正常工作：

5.2. Nginx



5.2.5. 启用基本身份验证

启用基本身份验证以限制特定网页上的访问。

以在目录 /var/www/html/auth-basic 下设置基本身份验证设置为例：

```
yum -y install httpd-tools
```

编辑 /etc/nginx/nginx.conf 文件：

```
# 在“server”部分添加以下内容
location /auth-basic {
    auth_basic           "Basic Auth";
    auth_basic_user_file "/etc/nginx/.htpasswd";
}
```

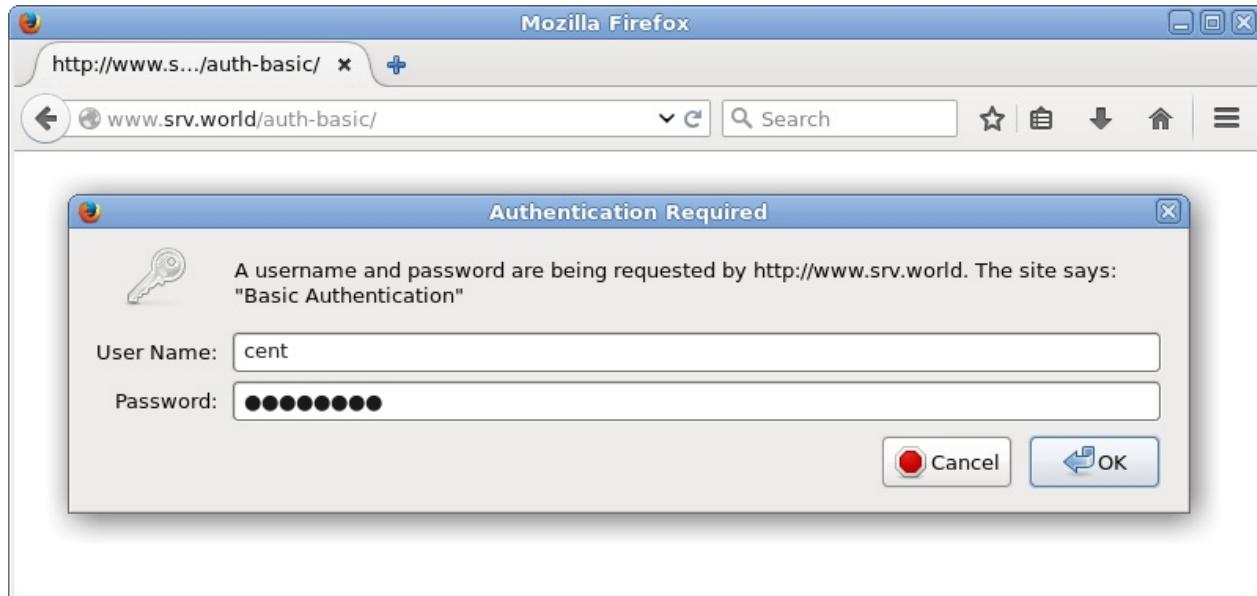
```
htpasswd -c /etc/nginx/.htpasswd cent
```

5.2. Nginx

```
New password: # 设置密码  
Re-type new password: # 确认密码  
Adding password for user cent
```

```
systemctl restart nginx
```

使用Web浏览器从客户端电脑访问，如下所示需要认证，使用上面添加的用户密码验证：



访问成功：



5.2.6. 反向代理

反向代理

例如，将端口80上配置到Nginx的HTTP连接转发到后端Apache httpd服务器：

5.2. Nginx

编辑 /etc/nginx/nginx.conf 文件：

```
# 在“server”部分作以下更改
server {
    listen      80 default_server;
    listen      [::]:80 default_server;
    server_name www.srv.world;

    proxy_redirect          off;
    proxy_set_header        X-Real-IP $remote_addr;
    proxy_set_header        X-Forwarded-For $proxy_add_x_for-
                           rwarded_for;
    proxy_set_header        Host $http_host;

    location / {
        proxy_pass http://node01.srv.world/;
    }
}
```

```
systemctl restart nginx
```

将后端的httpd设置更改为logging X-Forwarded-For header：

编辑 /etc/httpd/conf/httpd.conf 文件：

```
# 更改如下
LogFormat "\"%{X-Forwarded-For}i\" %l %u %t \"%r\" %>s %b \"%{Re-
ferer}i\" \"%{User-Agent}i\"" combined
```

```
systemctl restart httpd
```

从客户端HTTP访问Nginx服务器，确保所有工作正常，访问如下：



转发 WebSocket

例如，配置Nginx为在后端服务器的端口1337上工作的应用程序/chat设置代理开启（[示例应用程序来自这里](#)）。

编辑 `/etc/nginx/nginx.conf`：

```
# 在“server”部分作以下更改
server {
    listen      80 default_server;
    listen      [::]:80 default_server;
    server_name www.srv.world;

    location /socket.io/ {
        proxy_pass http://node01.srv.world:1337/socket.io/;
        proxy_http_version 1.1;
        proxy_set_header   Upgrade $http_upgrade;
        proxy_set_header   Connection "upgrade";
    }

    location /chat {
        proxy_pass          http://node01.srv.world:1337/;
        proxy_http_version 1.1;
        proxy_set_header   Upgrade $http_upgrade;
        proxy_set_header   Connection "upgrade";
    }

    location / {
        proxy_pass http://node01.srv.world/;
    }
}
```

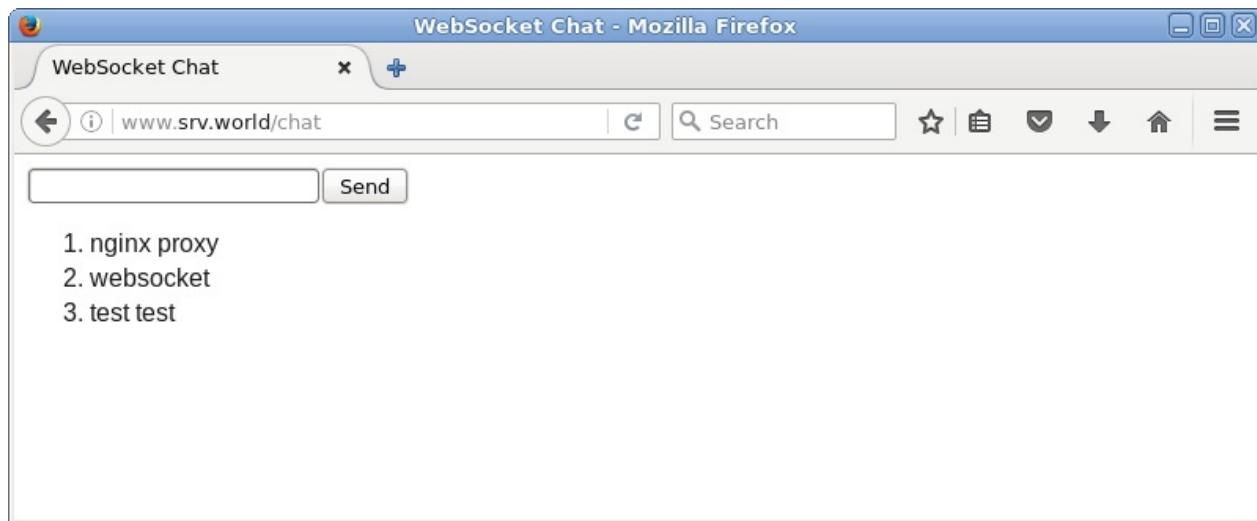
```
systemctl restart nginx
```

如果启用了SELinux，更改规则如下：

```
semanage port -a -t http_port_t -p tcp 1337
```

访问示例应用程序以确保它在代理环境中正常工作：

5.2. Nginx



负载均衡

配置Nginx的反向代理设置使用负载均衡功能：

```
(1) www.srv.world      [10.0.0.31]    - Nginx Server
(2) node01.srv.world   [10.0.0.51]    - Backend Web Server#1
(3) node02.srv.world   [10.0.0.52]    - Backend Web Server#2
(4) node03.srv.world   [10.0.0.53]    - Backend Web Server#3
```

例如，将Nginx配置为带有后端httpd服务器的负载均衡的代理服务器：

编辑 `/etc/nginx/nginx.conf` 文件：

5.2. Nginx

```
# 添加到“http”部分
# “backup”表示此服务器仅在其他服务器关闭时启用
# “weight = *”表示均衡权重
http {
    upstream backends {
        server node01.srv.world:80 weight=3;
        server node02.srv.world:80;
        server node03.srv.world:80 backup;
    }

    # 在“server”部分作以下更改
    server {
        listen      80 default_server;
        listen      [::]:80 default_server;
        server_name www.srv.world;

        proxy_redirect          off;
        proxy_set_header         X-Real-IP $remote_addr;
        proxy_set_header         X-Forwarded-For $proxy_add_x_for
        forwarded_for;
        proxy_set_header         Host $http_host;

        location / {
            proxy_pass http://backends;
        }
    }
}
```

```
systemctl restart nginx
```

从客户端访问Nginx服务器，以确保正常工作：





5.2.7. PHP-FPM

```
yum --enablerepo=epel -y install php php-mbstring php-pear php-fpm  
# 从EPEL安装PHP和PHP-FPM
```

配置PHP-FPM和Nginx，编辑 `/etc/php-fpm.d/www.conf` 文件：

```
# 更改  
user = nginx  
  
# 更改  
group = nginx
```

```
systemctl start php-fpm  
systemctl enable php-fpm
```

编辑 `/etc/nginx/nginx.conf` 文件：

```
# 在“server”部分添加以下内容  
location ~ \.php$ {  
    fastcgi_pass 127.0.0.1:9000;  
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
    fastcgi_param PATH_INFO $fastcgi_path_info;  
    include fastcgi_params;  
}
```

```
systemctl restart nginx
```

5.2. Nginx

```
echo "<?php phpinfo() ?>" > /usr/share/nginx/html/info.php # 创建测试页以确保PHP脚本正常工作
```

The screenshot shows a Mozilla Firefox browser window titled "phpinfo() - Mozilla Firefox". The address bar displays "phpinfo() | www.srv.world/info.php". The main content area shows the PHP information page. At the top, it says "PHP Version 5.4.16" and features a large "php" logo. Below this is a table with the following data:

System	Linux www.srv.world 3.10.0-327.22.2.el7.x86_64 #1 SMP Thu Jun 23 17:05:11 UTC 2016 x86_64
Build Date	Aug 11 2016 21:26:33
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/mbstring.ini, /etc/php.d/phar.ini, /etc/php.d posix.ini, /etc/php.d/sysvmsg.ini, /etc/php.d/sysvsem.ini, /etc/php.d/sysvshm.ini, /etc/php.d/wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zip.ini

5.3. 创建SSL证书

5.3.1. 创建自签名SSL证书

如果您使用您的服务器作为商业用途，最好购买和使用来自Verisign等的正式证书。

```
cd /etc/pki/tls/certs
```

```
make server.key # 创建密钥
```

```
umask 77 ; \
/usr/bin/openssl genrsa -aes128 2048 > server.key
Generating RSA private key, 2048 bit long modulus
...
...
e is 65537 (0x10001)
Enter pass phrase: # 设置密码短语
Verifying - Enter pass phrase: # 确认
```

```
openssl rsa -in server.key -out server.key # 从私钥中删除密码
```

```
Enter pass phrase for server.key: # 输入密码短语
writing RSA key
```

```
make server.csr # 创建SSL证书请求文件
```

```
umask 77 ; \
/usr/bin/openssl req -utf8 -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN # 国家
State or Province Name (full name) []:SC # 省
Locality Name (eg, city) [Default City]:CD # 城市
Organization Name (eg, company) [Default Company Ltd]:GTS # 公司
Organizational Unit Name (eg, section) []:Server World # 部门
Common Name (eg, your name or your server's hostname) []:www.srv
.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: # 回车
An optional company name []: # 回车
```

```
openssl x509 -in server.csr -out server.crt -req -signkey
server.key -days 3650 # 创建证书
```

```
Signature ok
subject=/C=JP/ST=Hiroshima/L=Hiroshima/O=GTS/OU=Server World/CN=
www.srv.world/emailAddress=xxx@srv.world
Getting Private key
```

5.3.2. 使用Let's Encrypt免费证书

Let's Encrypt是一个免费、开放，自动化的证书颁发机构，由ISRG（Internet Security Research Group）运作。

首先按[上一节内容](#)生成自签名证书。

5.3. 创建SSL证书

然后根据使用的[Apache](#)或是[Nginx](#)配置好SSL。

配置好需要生成网站的虚拟主机（最好先单独的配置文件，在本操作前备份一个）

确保域名解析正常，网站可以通过<https>正常访问（会提示证书不安全等问题，不用管它）（经测试，80端口必须能够被访问到，<https>的端口可以不是443）。

[Let's Encrypt](#)推荐使用Certbot ACME客户端生成证书。

这里以Apache httpd为例：

进入[Certbot](#)，在“*I'm using*”后面选择使用的Web服务器“Apache”，在“*on*”后面选择操作系统“CentOS/RHEL 7”。

选择完成后，会转到对应的操作说明页面，按页面提示来运行命令即可：

```
yum --enablerepo=epel -y install python-certbot-apache # 从EPEL安装
```

```
certbot --apache # 自动获得并安装证书（推荐使用手动更改Apache配置）
```

```
certbot --apache certonly # 仅获得证书
```

如果不留邮箱可以加入以下选项：

```
certbot --apache certonly --register-unsafely-without-email
```

生成的证书在目录 /etc/letsencrypt/live/对应域名 下，编辑虚拟主机配置文件，将证书路径部分修改为如下格式：

```
SSLCertificateFile /etc/letsencrypt/live/www.srv.world/cert.pem  
SSLCertificateKeyFile /etc/letsencrypt/live/www.srv.world/privkey.pem  
SSLCertificateChainFile /etc/letsencrypt/live/www.srv.world/chain.pem
```

获得的证书有效期为**90**天，通过运行以下命令来测试证书的自动更新：

```
certbot renew --dry-run
```

如果正常，可以通过添加cron或systemd作业来自动更新（建议每天运行两次，除非证书需要续订或撤销，否则不会执行任何操作），该作业运行以下操作：

```
certbot renew --quiet
```

有关续订的更多详细信息和选项，可参阅[完整文档](#)。

操作比较简单，就不多说了，提一下可能需要注意的地方：

1. 安装Certbot需要[配置EPEL](#)。
2. 获取证书命令运行时，根据配置文件读取需要生成证书的域名，直接输入数字选择，如果不在其中，直接输入域名后回车，根据提示完成。
3. 生成的证书用在对应的域名如 `www.srv.world` 的证书如果用在 `www.srv1.world`，访问 `www.srv1.world` 时浏览器可能会提示存在安全问题。

更新

最近按照上面的方法报错，好像是版本过旧的问题，更换为<https://certbot.eff.org/docs/install.html#certbot-auto>的方式操作：

```
wget https://dl.eff.org/certbot-auto
chmod a+x ./certbot-auto
./certbot-auto --help # 查看帮助信息
./certbot-auto # 运行脚本，会自动安装一些依赖的组件
./certbot-auto --apache certonly --register-unsafely-without-email # 与之前生成证书的命令相似
./certbot-auto renew
```

6. 数据库

- 6.1. MariaDB
 - 6.1.1. MariaDB 5.5
 - 6.1.1.1. 安装MariaDB
 - 6.1.1.2. 安装phpMyAdmin
 - 6.1.1.3. MariaDB复制
 - 6.1.2. MariaDB 10.1
 - 6.1.2.1. 安装MariaDB
 - 6.1.2.2. MariaDB Galera集群
- 6.2. PostgreSQL
 - 6.2.1. 安装PostgreSQL 9.2
 - 6.2.2. 安装phpPgAdmin
 - 6.2.3. PostgreSQL复制
- 6.3. Oracle Database
 - 6.3.1. 前提条件
 - 6.3.2. 安装Oracle Database 12c
 - 6.3.3. 添加网络侦听器
 - 6.3.4. 创建数据库
 - 6.3.5. 使用企业管理器
 - 6.3.6. 创建Systemd文件
- 6.4. Memcached
 - 6.4.1. 安装Memcached
 - 6.4.2. 基本用法
 - 6.4.3. 在Python上使用
 - 6.4.4. 在PHP上使用
 - 6.4.5. 在Node.js上使用
- 6.5. Redis
 - 6.5.1. 安装Redis
 - 6.5.2. 基本用法
 - 6.5.3. 在Python上使用
 - 6.5.4. 在PHP上使用
 - 6.5.5. 在Node.js上使用
 - 6.5.6. Redis复制

6. 数据库

- 6.5.7. Redis Sentinel
- 6.5.8. Redis Benchmark
- 6.6. MS SQL Server
 - 6.6.1. 安装SQL Server
 - 6.6.2. 安装SQL Server命令行工具
 - 6.6.3. 本地连接
 - 6.6.4. 升级
 - 6.6.5. 卸载

6.1. MariaDB

MariaDB数据库管理系统是MySQL的一个分支，主要由开源社区在维护，采用GPL授权许可。MariaDB的目的是完全兼容MySQL，包括API和命令行。

官网的[yum安装方法](#)

另：[MySQL](#)与MariaDB类似，官网[yum源](#)

6.1.1. MariaDB 5.5

6.1.1.1. 安装MariaDB

安装MariaDB 5.5（CentOS7默认版本）以配置数据库服务器。

```
yum -y install mariadb-server
```

编辑 `/etc/my.cnf` 文件：

```
# 在[mysqld]下面添加以下内容  
character-set-server=utf8
```

```
systemctl start mariadb  
systemctl enable mariadb
```

MariaDB的初始设置：

```
mysql_secure_installation
```

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL  
MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULL  
Y!
```

```
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, a  
nd
```

6.1. MariaDB

```
you haven't set the root password yet, the password will be blank,  
so you should just press enter here.
```

```
Enter current password for root (enter for none):  
OK, successfully used password, moving on...
```

```
Setting the root password ensures that nobody can log into the MariaDB  
root user without the proper authorisation.
```

```
# 设置root密码  
Set root password? [Y/n] y # 输入y确认  
New password: # 设置密码  
Re-enter new password: # 确认密码  
Password updated successfully!  
Reloading privilege tables..  
... Success!
```

```
By default, a MariaDB installation has an anonymous user, allowing anyone  
to log into MariaDB without having to have a user account created for  
them. This is intended only for testing, and to make the installation  
go a bit smoother. You should remove them before moving into a  
production environment.
```

```
# 移除anonymous（匿名）用户  
Remove anonymous users? [Y/n] y # 输入y确认  
... Success!
```

```
Normally, root should only be allowed to connect from 'localhost'. This  
ensures that someone cannot guess at the root password from the network.
```

```
# 禁止远程root登录  
Disallow root login remotely? [Y/n] y # 输入y确认  
... Success!
```

```
By default, MariaDB comes with a database named 'test' that anyone can
```

6.1. MariaDB

```
access. This is also intended only for testing, and should be removed  
before moving into a production environment.
```

```
# 移除测试数据库  
Remove test database and access to it? [Y/n] y # 输入y确认  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success!
```

```
Reloading the privilege tables will ensure that all changes made  
so far  
will take effect immediately.
```

```
# 重新加载权限表  
Reload privilege tables now? [Y/n] y # 输入y确认  
... Success!
```

```
Cleaning up...
```

```
All done! If you've completed all of the above steps, your MariaDB  
installation should now be secure.
```

```
Thanks for using MariaDB!
```

使用root连接MariaDB：

```
mysql -u root -p
```

6.1. MariaDB

```
Enter password: # 输入上面设置的MariaDB的root密码
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 5.5.37-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, Monty Program Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> select user,host,password from mysql.user; # 显示用户列表
+-----+-----+-----+
| user | host      | password          |
+-----+-----+-----+
| root | localhost | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx |
| root | 127.0.0.1 | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx |
| root | ::1        | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx |
+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [(none)]> show databases; # 显示数据库列表
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

MariaDB [(none)]> exit # 退出连接
Bye
```

如果MariaDB用于远程主机，firewalld防火墙设置（MariaDB使用端口3306/TCP）：

```
firewall-cmd --add-service=mysql --permanent
firewall-cmd --reload
```

6.1.1.2. 安装phpMyAdmin

安装phpMyAdmin以便在Web浏览器上从客户端运行MariaDB。

[安装Apache httpd](#)

[安装PHP](#)

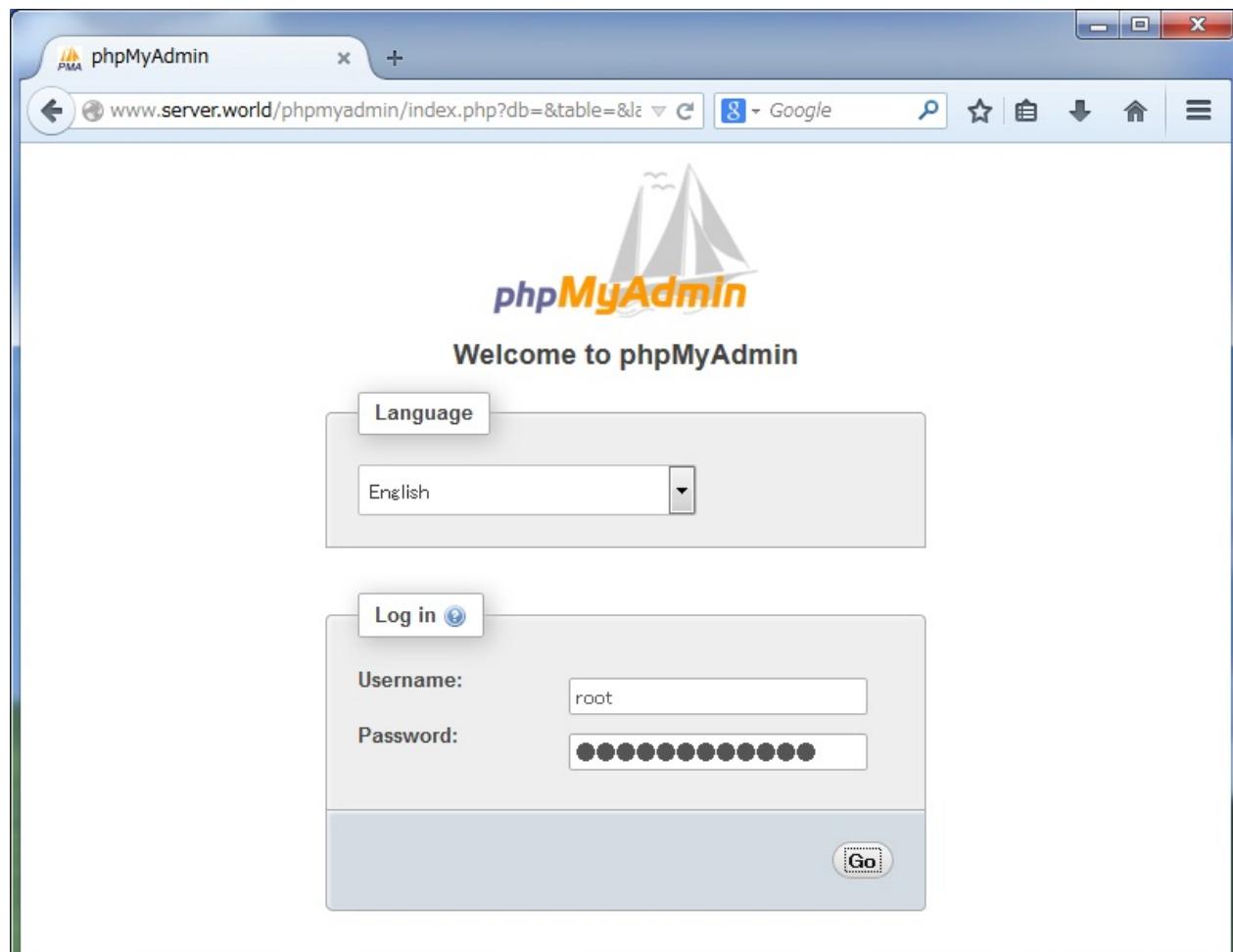
```
yum --enablerepo=epel -y install phpMyAdmin php-mysql php-mcrypt #  
从EPEL安装phpMyAdmin
```

编辑 `/etc/httpd/conf.d/phpMyAdmin.conf` 文件：

```
# 在几个Require ip后添加允许访问的IP地址（10.0.0.0/24根据实际情况替换）  
Require ip 127.0.0.1 10.0.0.0/24
```

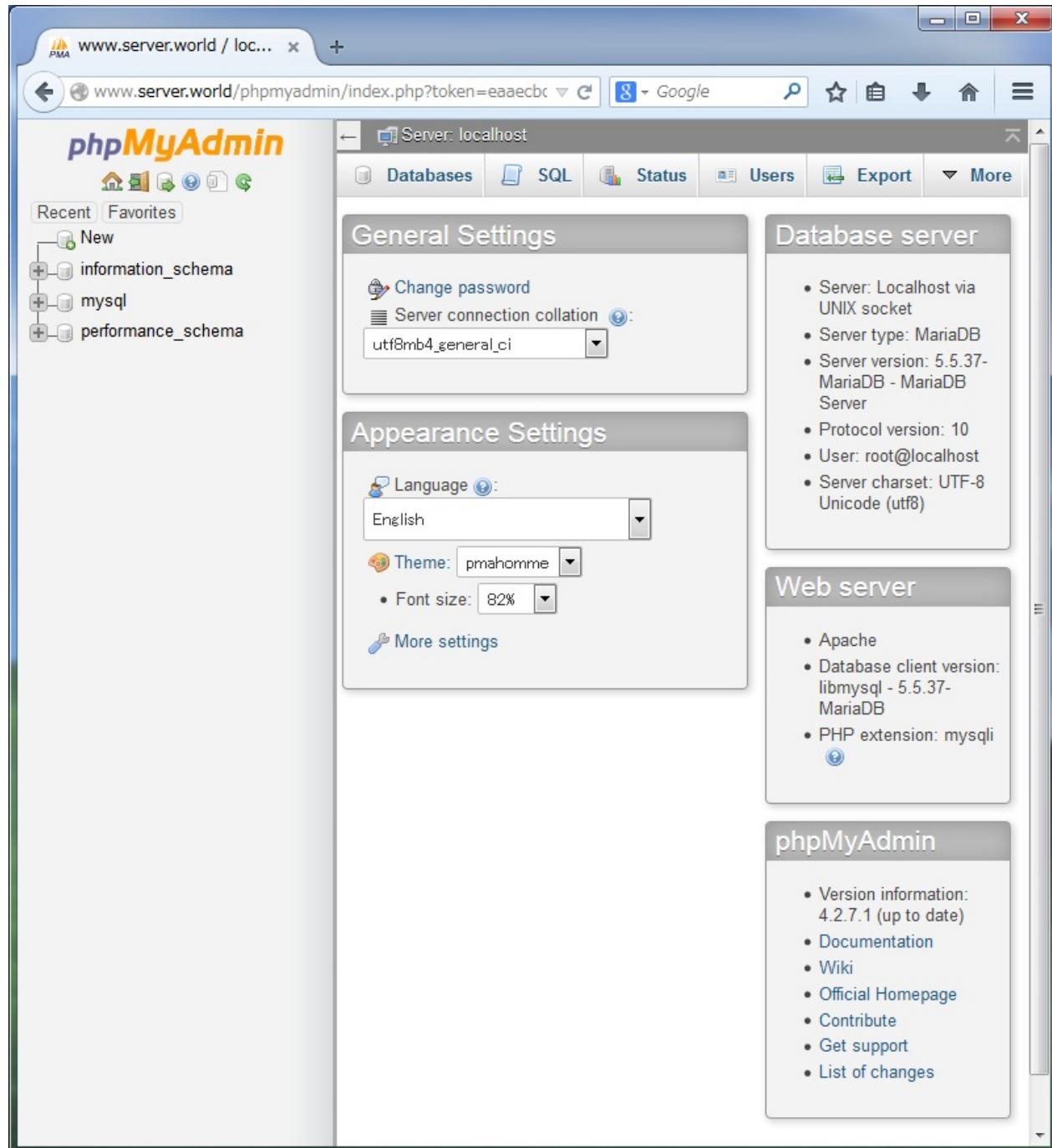
```
systemctl restart httpd
```

使用Web浏览器从客户端访问 `http://(主机名或IP地址)/phpmyadmin/`，然后使用MariaDB的用户在以下界面登录。此示例使用root用户：



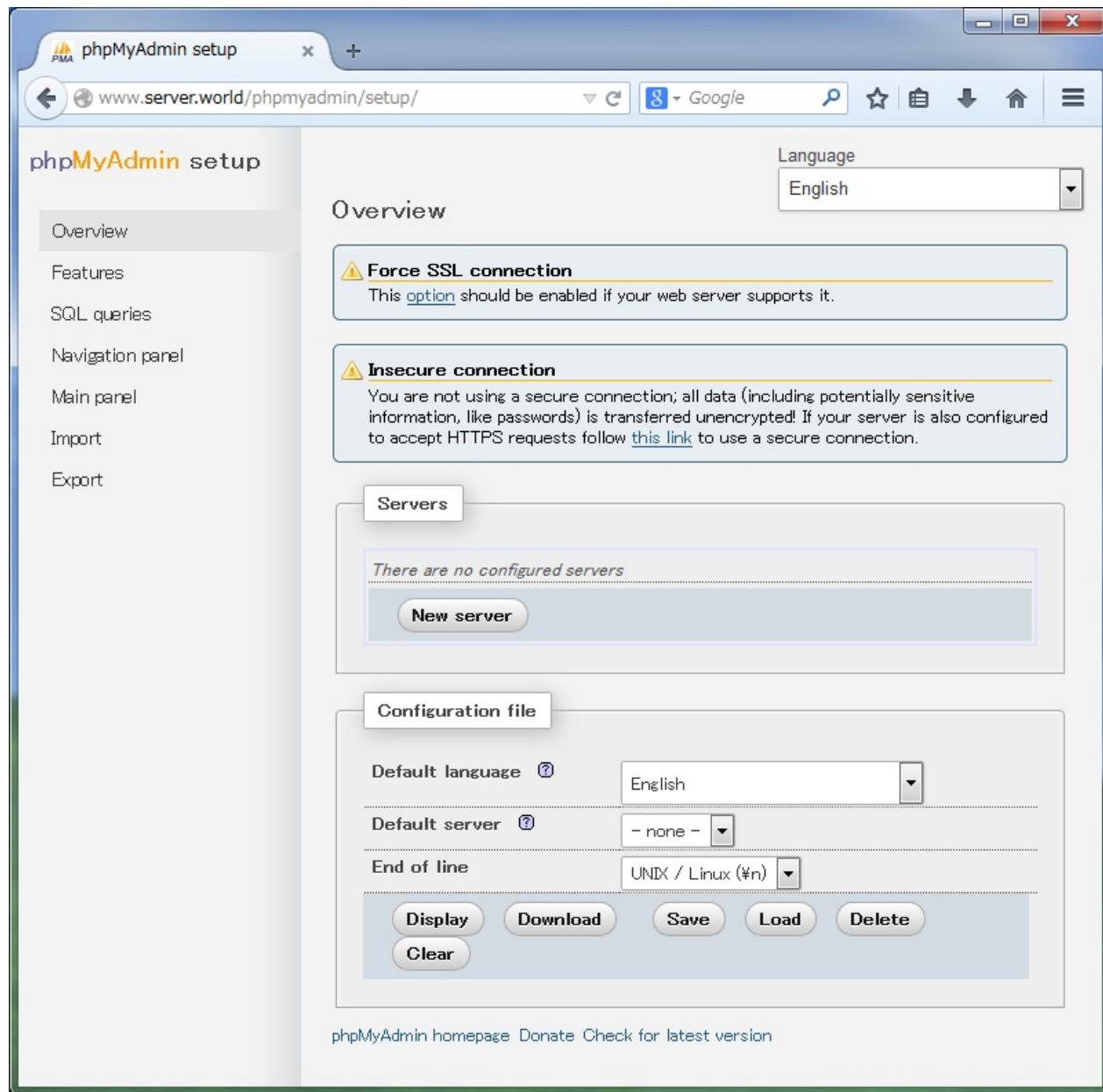
6.1. MariaDB

登录成功，可以在这里操作MariaDB：



可以在 [http://\(主机名或IP地址\)/phpmyadmin/setup](http://(主机名或IP地址)/phpmyadmin/setup) 更改MariaDB的设置：

6.1. MariaDB



6.1.1.3. MariaDB 复制

此配置是通用的主从设置。

在MariaDB主服务器上，编辑 `/etc/my.cnf` 文件，更改设置：

```
[mysqld]
# 在[mysqld]部分添加以下内容：获取二进制日志
log-bin=mysql-bin
# 定义唯一的服务器ID
server-id=101
```

```
systemctl restart mariadb
```

6.1. MariaDB

创建用于复制的用户：

```
mysql -u root -p
```

```
Enter password: # 输入root密码
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.41-MariaDB-log MariaDB Server

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

# 创建用户（在“password”部分设置任意密码）
MariaDB [(none)]> grant replication slave on *.* to replica@'%'
identified by 'password';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
```

在从服务器上安装和启动MariaDB服务器：

在主服务器配置防火墙规则（如果从服务器也有读取访问，使用同样的方法设置firewalld防火墙）：

```
firewall-cmd --add-service=mysql --permanent
firewall-cmd --reload
```

编辑 /etc/my.cnf 文件，更改从服务器上的设置：

```
[mysqld]
# 在[mysqld]部分添加以下内容：获取二进制日志
log-bin=mysql-bin
# 定义唯一的服务器ID（与主服务器ID不同）
server-id=102
# 只读
read_only=1
# 定义自己的主机名
report-host=node01.server.world
```

```
systemctl restart mariadb
```

在主服务器上获取转储数据：

```
mysql -u root -p
```

6.1. MariaDB

```
Enter password:  
Your MariaDB connection id is 3  
Server version: 5.5.41-MariaDB-log MariaDB Server  
  
Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
# 锁定所有表  
MariaDB [(none)]> flush tables with read lock;  
Query OK, 0 rows affected (0.00 sec)  
  
# 显示状态 (记住File, Position的值)  
MariaDB [(none)]> show master status;  
+-----+-----+-----+-----  
+  
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB |  
|  
+-----+-----+-----+-----  
+  
| mysql-bin.000001 | 465 | |  
|  
+-----+-----+-----+-----  
+  
1 row in set (0.00 sec)
```

保持上面的窗口，打开另一个窗口执行转储：

```
mysqldump -u root -p --all-databases --lock-all-tables --events >  
mysql_dump.sql
```

```
Enter password:
```

回到之前的窗口：

6.1. MariaDB

```
# 解锁  
MariaDB [(none)]> unlock tables;  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> exit  
Bye
```

将转储传输到从服务器：

```
scp mysql_dump.sql node01.server.world:/tmp/
```

```
root@node01.server.world's password:  
mysql_dump.sql 100% 515KB 514.7KB/s 00:00
```

确保设置正常工作在主服务器上创建数据库。在从服务器上配置复制设置：

```
mysql -u root -p < /tmp/mysql_dump.sql # 从服务器导入主服务器转储
```

```
Enter password:
```

```
mysql -u root -p
```

```
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 3  
Server version: 5.5.41-MariaDB-log MariaDB Server  
  
Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> change master to  
    -> master_host='10.0.0.31', # 主服务器的IP  
    -> master_user='replica', # 复制ID（上面主服务器创建的用户复制的用户）  
    -> master_password='password', # 复制ID的密码  
    -> master_log_file='mysql-bin.000001', # File值在主机上确认
```

```

-> master_log_pos=465; # Position值在主机上确认
Query OK, 0 rows affected (0.58 sec)

# 开始复制
MariaDB [(none)]> start slave;
Query OK, 0 rows affected (0.00 sec)

# 显示状态
MariaDB [(none)]> show slave status\G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 10.0.0.31
Master_User: replica
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000001
Read_Master_Log_Pos: 536
Relay_Log_File: mariadb-relay-bin.000002
Relay_Log_Pos: 600
Relay_Master_Log_File: mysql-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 536
Relay_Log_Space: 896
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:

```

```
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
    Last_IO_Errorno: 0
    Last_SQL_Error:
    Last_SQL_Errorno: 0
    Last_SQL_Error:
Replicate_Ignore_Server_Ids:
    Master_Server_Id: 101
1 row in set (0.00 sec)
```

6.1.2. MariaDB 10.1

6.1.2.1. 安装MariaDB

CentOS7官方仓库中的MariaDB版本是5.5，可以使用RPM包安装10.1。

可以从[CentOS SCLo软件集合](#)安装（即使已经安装了5.5，也可以安装，因为10.1位于另一路径）：

```
yum --enablerepo=centos-scllo-rh -y install rh-mariadb101-mariadb-
server # 从SCLo安装
```

此方法将MariaDB 10.1安装在 /opt 目录下，要使用它，需加载如下的环境变量：

```
scl enable rh-mariadb101 bash
```

```
mysql -V
```

```
mysql Ver 15.1 Distrib 10.1.14-MariaDB, for Linux (x86_64) using
EditLine wrapper
```

```
which mysql
```

```
/opt/rh/rh-mariadb101/root/usr/bin/mysql
```

如果希望在登录时自动启用MariaDB 10.1，需如下配置：

编辑 /etc/profile.d/rh-mariadb101.sh 文件：

```
#!/bin/bash

source /opt/rh/rh-mariadb101/enable
export X_SCLS=`scl enable rh-mariadb101 'echo $X_SCLS'`"
```

启用MariaDB 10.1并配置初始设置：

编辑 `/etc/opt/rh/rh-mariadb101/my.cnf.d/mariadb-server.cnf` 文件：

```
# 在[mysqld]下面添加以下内容
character-set-server=utf8
```

```
systemctl start rh-mariadb101-mariadb
systemctl enable rh-mariadb101-mariadb
```

以下初始设置，连接数据库等操作与MariaDB 5.5一样

```
mysql_secure_installation

mysql -u root -p
```

如果MariaDB用于远程主机，firewalld防火墙设置（MariaDB使用端口3306/TCP）：

```
firewall-cmd --add-service=mysql --permanent
firewall-cmd --reload
```

安装phpMyAdmin和MariaDB复制基本与MariaDB 5.5一样。

6.1.2.2. MariaDB Galera集群

集群中的所有节点在此配置中成为主服务器。

在所有节点上安装MariaDB Galera软件包，如下所示：

```
yum --enablerepo=centos-sclrh -y install rh-mariadb101-mariadb-
server-galera #从SCLo安装
```

firewalld防火墙设置（MariaDB使用端口3306/TCP）：

6.1. MariaDB

```
firewall-cmd --add-service=mysql --permanent  
firewall-cmd --add-port={3306/tcp,4567/tcp,4568/tcp,4444/tcp} --permanent  
firewall-cmd --reload
```

配置第一个节点，如下所示：

```
mv /etc/opt/rh/rh-mariadb101/my.cnf.d/galera.cnf /etc/opt/rh/rh-mariadb101/my.cnf.d/galera.cnf.org
```

编辑 /etc/opt/rh/rh-mariadb101/my.cnf.d/mariadb-server.cnf 文件：

```
# 取消注释并作如下更改  
[galera]  
# Mandatory settings  
wsrep_on=ON  
wsrep_provider=/opt/rh/rh-mariadb101/root/usr/lib64/galera/libgalera_smm.so  
wsrep_cluster_address=gcomm://  
  
# 取消注释如下  
binlog_format=row  
default_storage_engine=InnoDB  
innodb_autoinc_lock_mode=2  
bind-address=0.0.0.0  
  
# 添加：集群名称  
wsrep_cluster_name="MariaDB_Cluster"  
  
# 本机IP地址  
wsrep_node_address="10.0.0.31"  
  
# 复制提供程序  
wsrep_sst_method=rsync
```

启动Galera集群：

```
/opt/rh/rh-mariadb101/root/usr/bin/galera_new_cluster
```

运行基本初始设置：

6.1. MariaDB

```
mysql_secure_installation
```

配置除上一节点以外的其他节点，如下所示：

```
mv /etc/opt/rh/rh-mariadb101/my.cnf.d/galera.cnf /etc/opt/rh/rh-mariadb101/my.cnf.d/galera.cnf.org
```

编辑 `/etc/opt/rh/rh-mariadb101/my.cnf.d/mariadb-server.cnf` 文件：

```
# 取消注释并作如下更改
[galera]
# Mandatory settings
wsrep_on=ON
wsrep_provider=/opt/rh/rh-mariadb101/root/usr/lib64/galera/libgalera_smm.so

# 指定集群中的所有节点
wsrep_cluster_address="gcomm://10.0.0.31,10.0.0.51"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# 添加：集群名称
wsrep_cluster_name="MariaDB_Cluster"

# 本机IP地址
wsrep_node_address="10.0.0.51"

# 复制提供程序
wsrep_sst_method=rsync
```

```
systemctl start rh-mariadb101-mariadb
```

群集设置完成，确保状态如下（“wsrep_local_state_comment”应该为“Synced”）：

```
mysql -u root -p
```

```
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.1.14-MariaDB MariaDB Server
```

```
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> show status like 'wsrep_%';
+-----+-----+
| Variable_name | Value
+-----+-----+
| wsrep_apply_oooe | 0.000000
| wsrep_apply_oool | 0.000000
| wsrep_apply_window | 0.000000
| wsrep_causal_reads | 0
| wsrep_cert_deps_distance | 0.000000
| wsrep_cert_index_size | 0
| wsrep_cert_interval | 0.000000
| wsrep_cluster_conf_id | 2
| wsrep_cluster_size | 2
| wsrep_cluster_state_uuid | d3305600-5ed1-11e6-a6aa-736abc91d198
| wsrep_cluster_status | Primary
| wsrep_commit_oooe | 0.000000
| wsrep_commit_oool | 0.000000
| wsrep_commit_window | 0.000000
```

wsrep_connected	ON
wsrep_evs_delayed	
wsrep_evs_evict_list	
wsrep_evs_repl_latency	0.000462792/0.000761504/0.00102
34/0.000230342/3	
wsrep_evs_state	OPERATIONAL
wsrep_flow_control_paused	0.000000
wsrep_flow_control_paused_ns	0
wsrep_flow_control_recv	0
wsrep_flow_control_sent	0
wsrep_gcomm_uuid	1eac02a8-5ed2-11e6-baa3-4e98583
04c9a	
wsrep_incoming_addresses	10.0.0.31:3306,10.0.0.51:3306
wsrep_last_committed	0
wsrep_local_bf_aborts	0
wsrep_local_cached_downto	18446744073709551615
wsrep_local_cert_failures	0
wsrep_local_commits	0
wsrep_local_index	1
wsrep_local_recv_queue	0
wsrep_local_recv_queue_avg	0.000000
wsrep_local_recv_queue_max	1
wsrep_local_recv_queue_min	0

6.1. MariaDB

```
| wsrep_local_replays          | 0
  |
| wsrep_local_send_queue       | 0
  |
| wsrep_local_send_queue_avg   | 0.500000
  |
| wsrep_local_send_queue_max   | 2
  |
| wsrep_local_send_queue_min   | 0
  |
| wsrep_local_state            | 4
  |
| wsrep_local_state_comment    | Synced
  |
| wsrep_local_state_uuid       | d3305600-5ed1-11e6-a6aa-736abc9
1d198
  |
| wsrep_protocol_version       | 7
  |
| wsrep_provider_name          | Galera
  |
| wsrep_provider_vendor         | Codership Oy <info@codership.com
>
  |
| wsrep_provider_version        | 3.12(r9921e73)
  |
| wsrep_ready                  | ON
  |
| wsrep_received                | 2
  |
| wsrep_received_bytes          | 194
  |
| wsrep_repl_data_bytes         | 0
  |
| wsrep_repl_keys               | 0
  |
| wsrep_repl_keys_bytes         | 0
  |
| wsrep_repl_other_bytes        | 0
  |
| wsrep_replicated              | 0
  |
| wsrep_replicated_bytes        | 0
```

6.1. MariaDB

```
| wsrep_thread_count | 2  
|  
+-----+-----+  
|-----+  
57 rows in set (0.01 sec)
```



6.2. PostgreSQL

PostgreSQL是一个自由的对象-关系数据库服务器(ORDBMS数据库管理系统)

官网的[yum安装方法](#)

6.2.1. 安装PostgreSQL 9.2

```
yum -y install postgresql-server
```

```
postgresql-setup initdb
```

```
Initializing database ... OK
```

编辑 `/var/lib/pgsql/data/postgresql.conf` 文件：

```
# 如果允许从远程主机访问，取消注释并更改  
listen_addresses = '*'
```

```
# 如果更改日志格式，取消注释并更改，下面的示例是[日期 用户 数据库 ***]格式  
log_line_prefix = '%t %u %d '
```

```
systemctl start postgresql  
systemctl enable postgresql
```

如果PostgreSQL用于远程主机，firewalld防火墙设置（PostgreSQL使用端口5432/TCP）：

```
firewall-cmd --add-service=postgresql --permanent  
firewall-cmd --reload
```

设置PostgreSQL管理员用户的密码，添加一个用户并添加一个测试数据库：

```
su - postgres
```

6.2. PostgreSQL

```
-bash-4.2$ psql -c "alter user postgres with password 'password'  
" # 设置密码  
ALTER ROLE  
  
-bash-4.2$ createuser cent # 添加数据库用户 ("cent"为例)  
  
-bash-4.2$ createdb testdb -O cent # 创建一个测试数据库 (所有者是上面  
的用户 "cent")
```

以刚刚添加的用户身份登录系统并操作数据库作为测试操作：

```
psql -l # 显示数据库
```

```
          List of databases  
 Name     | Owner      | Encoding | Collate   | Ctype    |  
 Access privileges  
-----+-----+-----+-----+-----+-----+  
-----  
 postgres | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =  
 template0 | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =  
 c/postgres +  
             |           |           |           |           | p  
 ostgres=CTc/postgres  
 template1 | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =  
 c/postgres +  
             |           |           |           |           | p  
 ostgres=CTc/postgres  
 testdb   | cent     | UTF8      | en_US.UTF-8 | en_US.UTF-8 |  
(4 rows)
```

```
psql testdb # 连接到测试数据库
```

```
psql (9.2.13)
Type "help" for help.

# 设置密码
testdb=# alter user cent with password 'password';
ALTER ROLE

# 创建test表
testdb=# create table test ( no int, name text );
CREATE TABLE

# 插入测试数据
testdb=# insert into test (no, name) values (1, 'cent');
INSERT 0 1

# 显示表
testdb=# select * from test;
 no | name
----+-----
 1 | cent
(1 row)

# 删除test表
testdb=# drop table test;
DROP TABLE

# 退出
testdb=# \q
```

```
dropdb testdb # 删除测试数据库
```

6.2.2. 安装phpPgAdmin

[安装Apache httpd](#)

[安装PHP](#)

```
yum --enablerepo=epel -y install phpPgAdmin php-pgsql # 从EPEL安装
```

编辑 /etc/phpPgAdmin/config.inc.php 文件：

6.2. PostgreSQL

```
# 添加
$conf['servers'][0]['host'] = 'localhost';

# 如果允许使用特权用户登录（如postgres，root）更改为false
$conf['extra_login_security'] = false;

# 更改
$conf['owned_only'] = true;
```

编辑 `/var/lib/pgsql/data/pg_hba.conf` 文件：

```
# 更改如下并添加访问权限
host    all        all      127.0.0.1/32      md5
host    all        all      10.0.0.0/24      md5
host    all        all      ::1/128        md5
```

编辑 `/etc/httpd/conf.d/phpPgAdmin.conf` 文件：

```
# 添加访问权限
Require local
Require ip 10.0.0.0/24
```

```
systemctl restart postgresql httpd
```

如果启用了SELinux：

```
setsebool -P httpd_can_network_connect_db on
```

访问 `http://(主机名或IP地址)/phpPgAdmin/`，单击左侧菜单上的“PostgreSQL”：

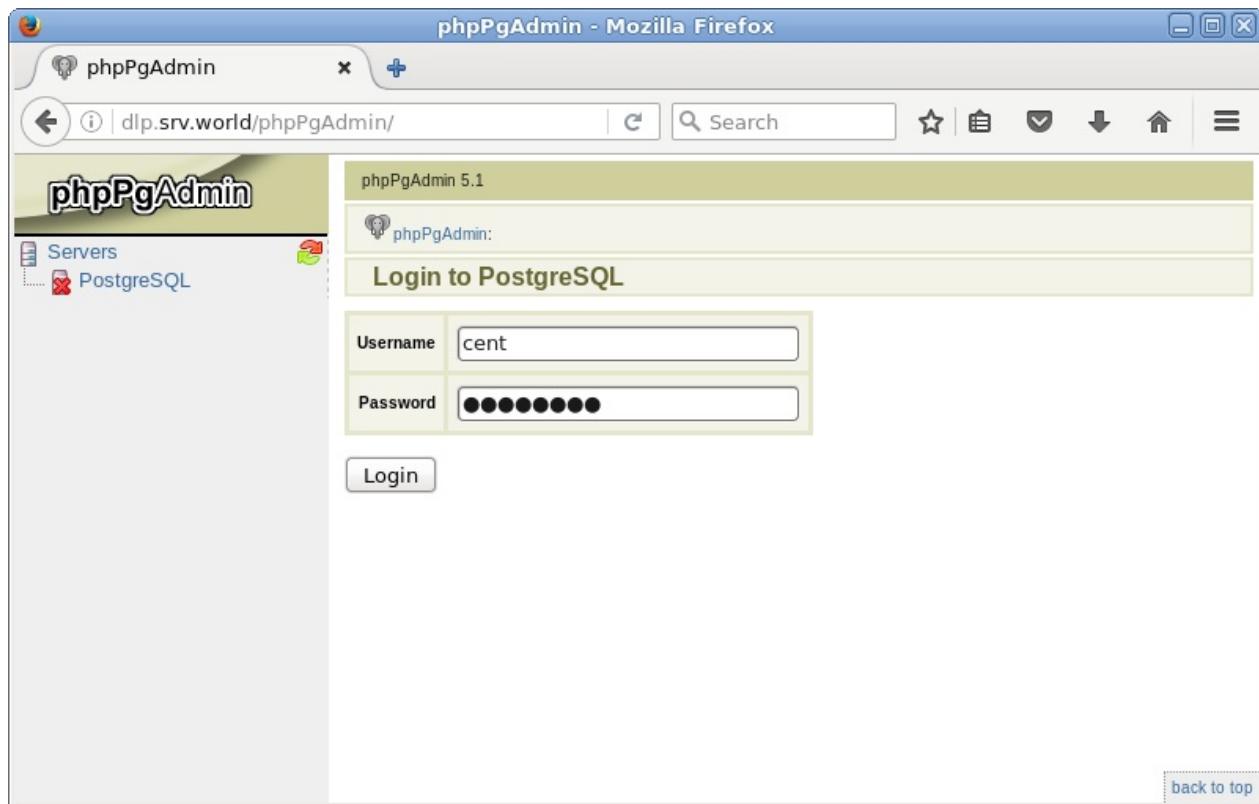
6.2. PostgreSQL

The screenshot shows the phpPgAdmin 5.1 homepage. At the top, there's a toolbar with icons for back, forward, search, and other browser functions. Below the toolbar, the title bar says "phpPgAdmin - Mozilla Firefox". The main content area has a green header bar with the text "phpPgAdmin 5.1" and "Introduction". On the left, there's a sidebar with "Servers" and "PostgreSQL" listed. The main content area displays the title "phpPgAdmin 5.1 (PHP 5.4.16)" and two dropdown menus: "Language" set to "English" and "Theme" set to "Default". Below these, a message says "Welcome to phpPgAdmin." followed by a bulleted list: "• [phpPgAdmin Homepage](#)", "• [PostgreSQL Homepage](#)", "• [Report a Bug](#)", and "• [View online FAQ](#)". In the bottom right corner of the content area, there's a link "back to top".

验证PostgreSQL中的用户和密码：

The screenshot shows the "Login to PostgreSQL" page. The title bar says "phpPgAdmin - Mozilla Firefox". The main content area has a green header bar with the text "phpPgAdmin 5.1" and "Login to PostgreSQL". On the left, there's a sidebar with "Servers" and "PostgreSQL" listed. The main content area contains two input fields: "Username" with the value "cent" and "Password" with several black dots representing the password. Below the password field is a "Login" button. In the bottom right corner of the content area, there's a link "back to top".

登录成功，可以在这里操作PostgreSQL：



6.2.3. PostgreSQL 复制

此配置是主从设置。

配置主服务器：

```
yum -y install postgresql-server  
postgresql-setup initdb
```

Initializing database ... OK

编辑 `/var/lib/pgsql/data/postgresql.conf` 文件：

6.2. PostgreSQL

```
# 取消注释并更改
listen_addresses = '*'

# 取消注释并更改
wal_level = hot_standby

# 取消注释并更改
# on => sync
# remote_write => memory sync
# local => slave is asynchronous
# off => asynchronous
synchronous_commit = local

# 取消注释并更改（启用archive_mode）
archive_mode = on

# 取消注释并更改（获取档案命令）
archive_command = 'cp %p /var/lib/pgsql/archive/%f'

# 取消注释并更改（从服务器 + 1）
max_wal_senders = 2

# 取消注释并更改
wal_keep_segments = 10

# 取消注释并更改（任意名称）
synchronous_standby_names = 'slave01'
```

编辑 `/var/lib/pgsql/data/pg_hba.conf` 文件：

```
# 添加到最后
# host replication [复制用户] [允许的IP地址] password
host    replication    replica        127.0.0.1/32
      md5
host    replication    replica        10.0.0.30/32
      md5
host    replication    replica        10.0.0.51/32
      md5
```

6.2. PostgreSQL

```
systemctl start postgresql  
systemctl enable postgresql
```

创建用于复制的用户：`su - postgres`

```
-bash-4.2$ createuser --replication -P replica  
Enter password for new role:  
Enter it again:
```

firewalld防火墙规则：

```
firewall-cmd --add-service=postgresql --permanent  
firewall-cmd --reload
```

配置从服务器：

```
yum -y install postgresql-server  
su - postgres
```

6.2. PostgreSQL

```
# 从主服务器获取备份
-bash-4.2$ pg_basebackup -h 10.0.0.30 -U replica -D /var/lib/pgs
ql/data -P --xlog
Password: # 用户“replica”的密码

-bash-4.2$ vi /var/lib/pgsql/data/postgresql.conf

# 取消注释并更改
hot_standby = on

-bash-4.2$ cp /usr/share/pgsql/recovery.conf.sample /var/lib/pgs
ql/data/recovery.conf

-bash-4.2$ vi /var/lib/pgsql/data/recovery.conf

# 取消注释并更改（获取档案命令）
restore_command = 'scp 10.0.0.30:/var/lib/pgsql/archive/%f %p'

# 取消注释并更改
standby_mode = on

# 取消注释并更改（连接信息到主服务器）
primary_conninfo = 'host=10.0.0.30 port=5432 user=replica passwo
rd=password application_name=slave01'

-bash-4.2$ exit
logout
```

```
systemctl start postgresql
systemctl enable postgresql
```

确保设置正常：

```
su - postgres
```

6.2. PostgreSQL

```
-bash-4.2$ psql -c "select application_name, state, sync_priority, sync_state from pg_stat_replication;"  
  
application_name | state | sync_priority | sync_state  
-----+-----+-----+-----  
slave01 | streaming | 1 | sync  
(1 row)
```

6.3. Oracle Database

[Oracle Database](#)，又名Oracle RDBMS，或简称Oracle。是甲骨文公司的一款关系数据库管理系统。

6.3.1. 前提条件

安装Oracle Database 12c之前，在此处更改一些设置以安装Oracle数据库的要求。

安装桌面环境

安装所需的软件包：

```
yum -y install binutils compat-libcap1 gcc gcc-c++ glibc glibc.i686  
glibc-devel glibc.i686 ksh libaio libaio.i686 libaio-devel libaio-  
devel.i686 libgcc libgcc.i686 libstdc++ libstdc++17.i686 libstdc++-  
devel libstdc++-devel.i686 compat-libstdc++-33 compat-  
libstdc++-33.i686 libXi libXi.i686 libXtst libXtst.i686 make  
sysstat
```

编辑内核参数：

```
MEMTOTAL=$(free -b | sed -n '2p' | awk '{print $2}')
```

```
SHMMAX=$(expr $MEMTOTAL / 2)
```

```
SHMMNI=4096
```

```
PAGESIZE=$(getconf PAGE_SIZE)
```

```
cat >> /etc/sysctl.conf << EOF
```

6.3. Oracle Database

```
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.shmmmax = $SHMMAX
kernel.shmall = `expr \$SHMMAX / \$PAGESIZE \` \* `\$SHMMNI / 16 \```
kernel.shmmni = $SHMMNI
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
EOF
```

```
sysctl -p
```

```
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.shmmmax = 6274715648
kernel.shmall = 392169728
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
```

为Oracle数据库服务创建用户和组：

```
i=54321; for group in oinstall dba backupdba oper dgdba kmdba; do
groupadd -g $i $group; i=`expr $i + 1`done
```

```
useradd -u 1200 -g oinstall -G dba,oper,backupdba,dgdba,kmdba -d
/home/oracle oracle
```

```
passwd oracle
```

6.3. Oracle Database

```
Changing password for user oracle.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.
```

```
mkdir -p /u01/app/oracle
```

```
chown -R oracle:oinstall /u01/app
```

```
chmod -R 775 /u01
```

编辑 `/etc/pam.d/login` 文件：

```
# 在第14行附近添加  
session      required      pam_sselinux.so open  
session      required      pam_namespace.so  
session      required      pam_limits.so  
session      optional      pam_keyinit.so force revoke  
session      include       system-auth  
-session     optional      pam_ck_connector.so
```

编辑 `/etc/security/limits.conf` 文件：

```
# 添加到最后  
oracle  soft  nproc  2047  
oracle  hard   nproc  16384  
oracle  soft  nofile 1024  
oracle  hard  nofile 65536  
oracle  soft  stack  10240  
oracle  hard  stack  32768
```

使用用户“`oracle`”登录系统并设置环境变量：

编辑 `~/.bash_profile` 文件：

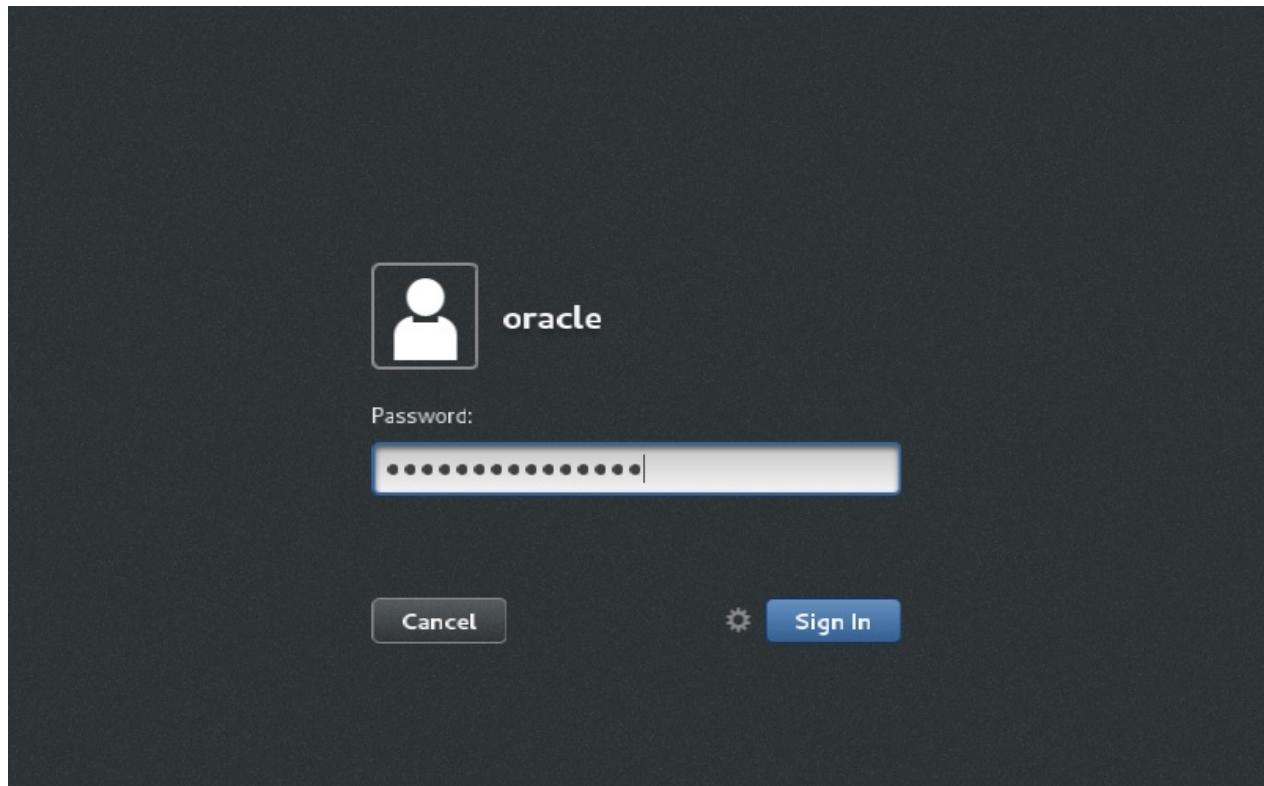
```
# 添加到最后  
umask 022  
export ORACLE_BASE=/u01/app/oracle
```

创建一个临时目录进行安装：

```
mkdir tmp
```

6.3.2. 安装Oracle Database 12c

使用上一步创建的用户“oracle”登录桌面环境：



下载Oracle Database 12c for Linux并上传到服务器tmp目录，并运行安装程序：

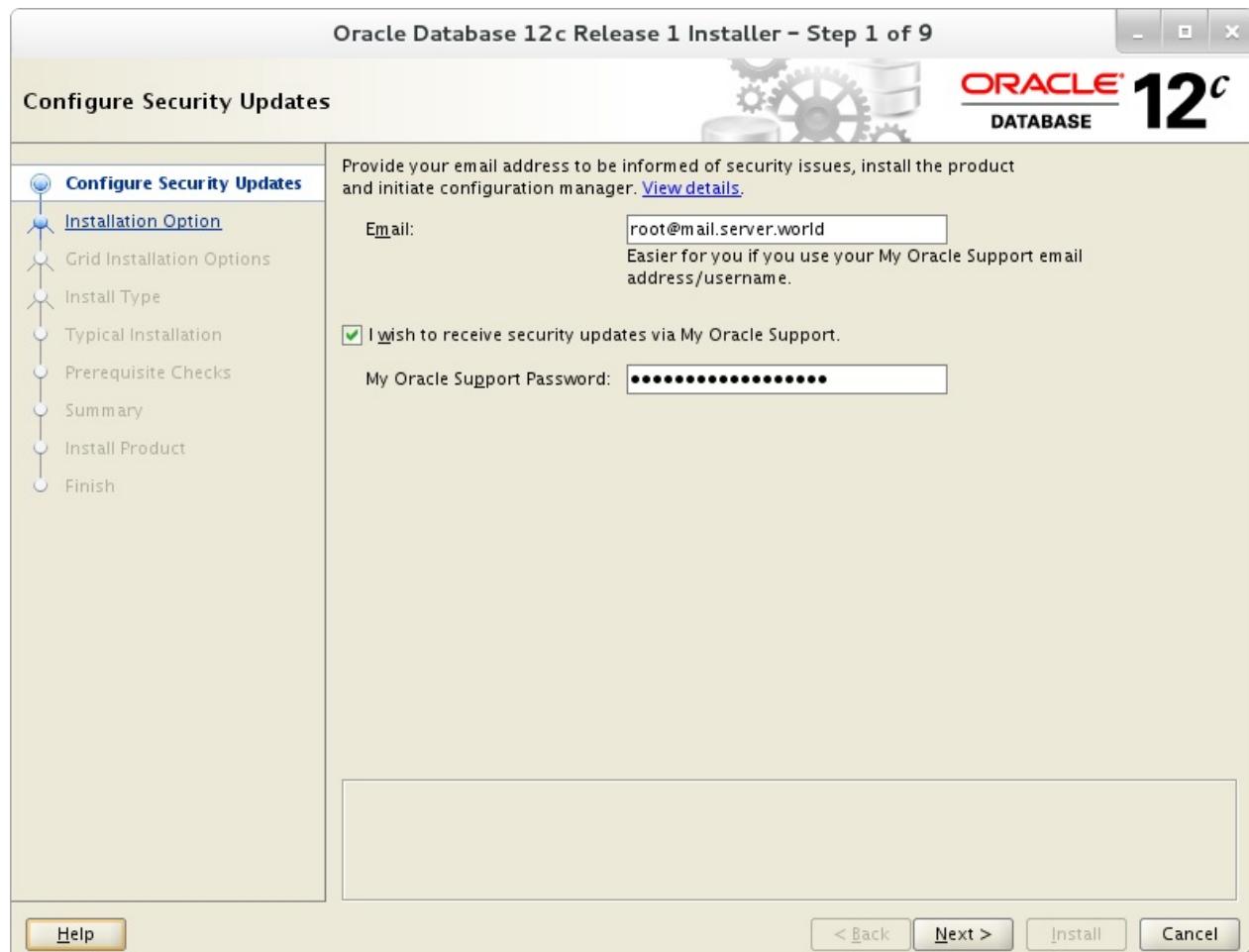
```
cd tmp  
unzip linuxamd64_12102_database_1of2.zip  
unzip linuxamd64_12102_database_2of2.zip
```

```
./database/runInstaller
```

Oracle安装程序启动如下：

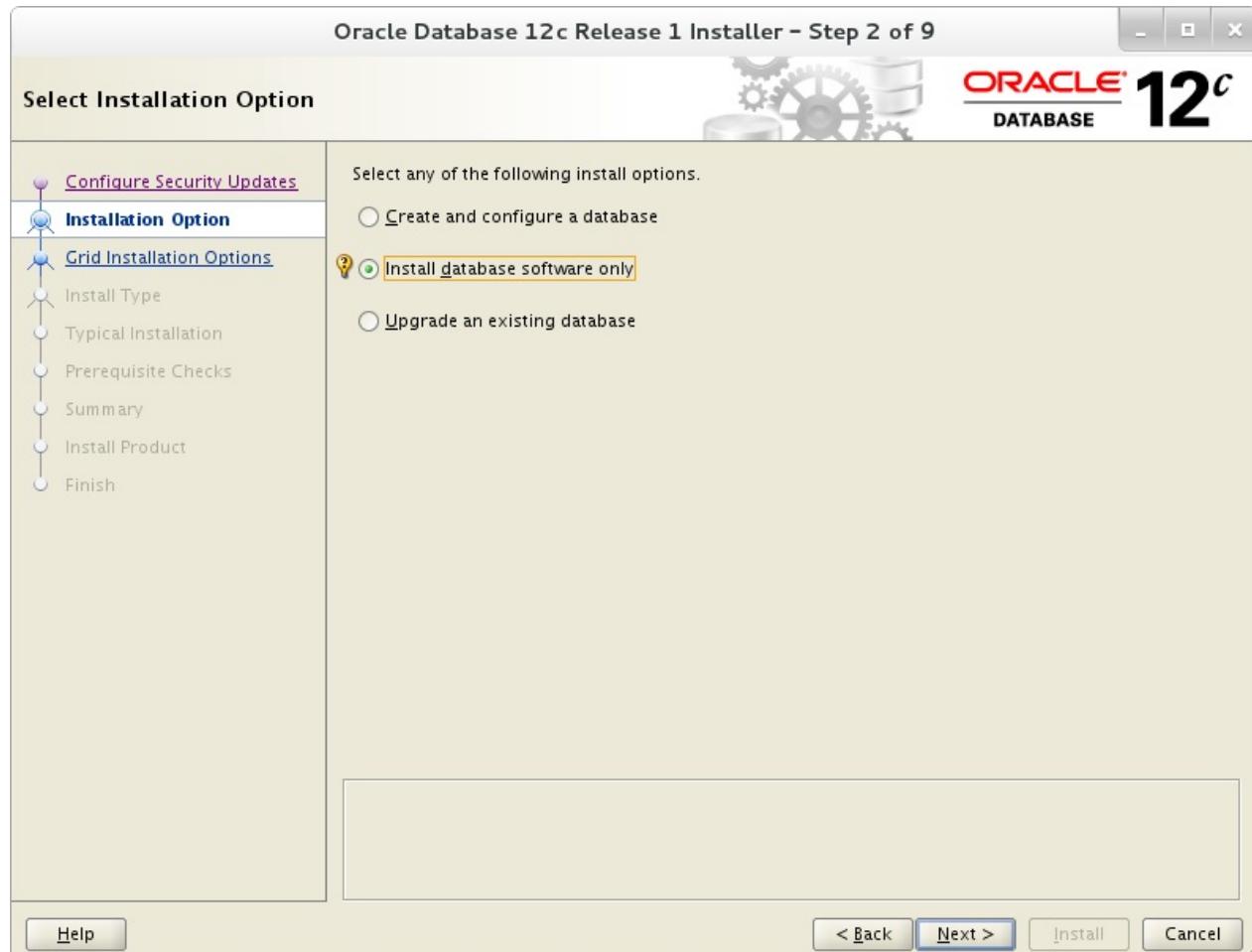
首先，设置电子邮件地址和密码，是否从Oracle接收信息，如安全问题等：

6.3. Oracle Database



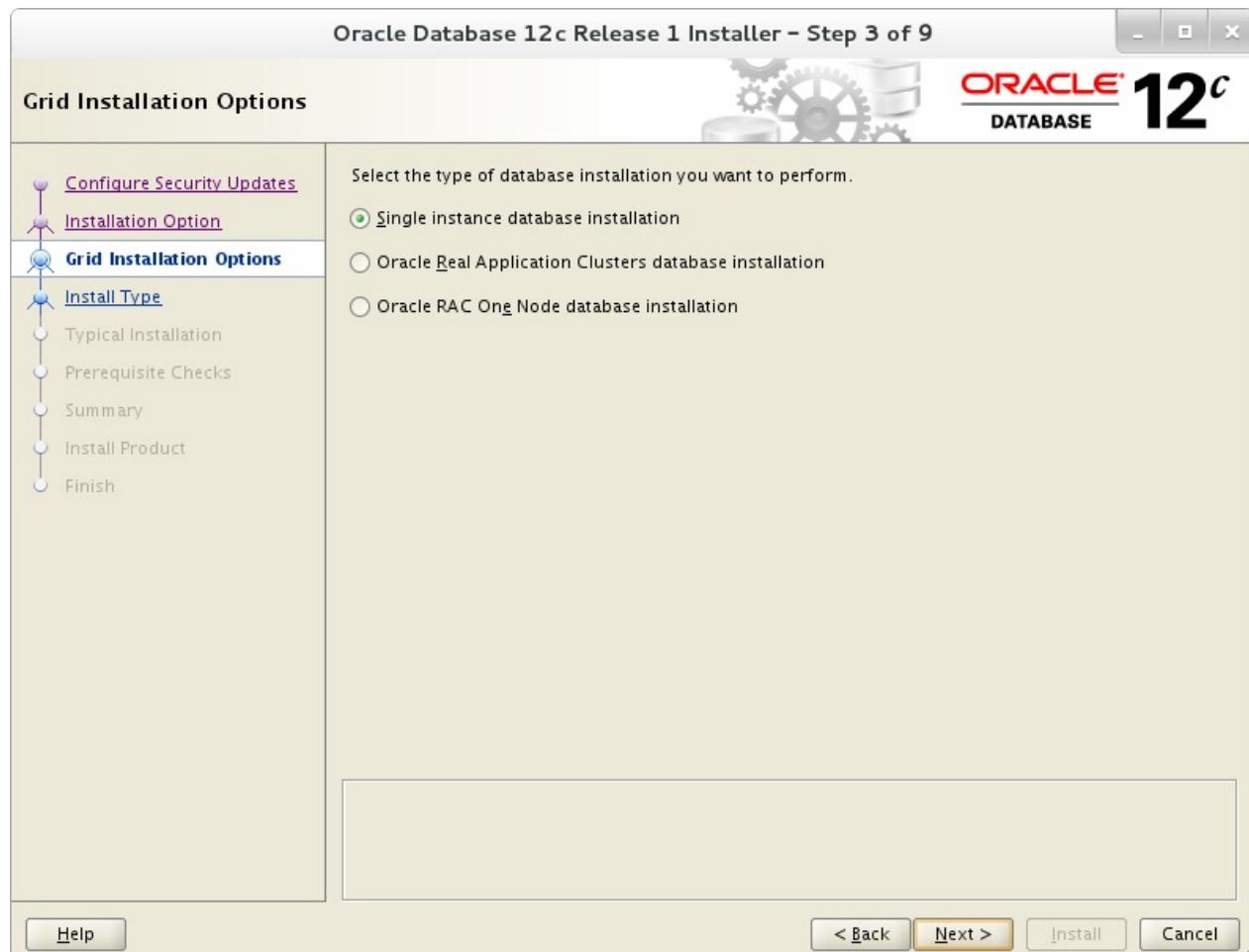
在此示例中，选择“Install database software only”：

6.3. Oracle Database



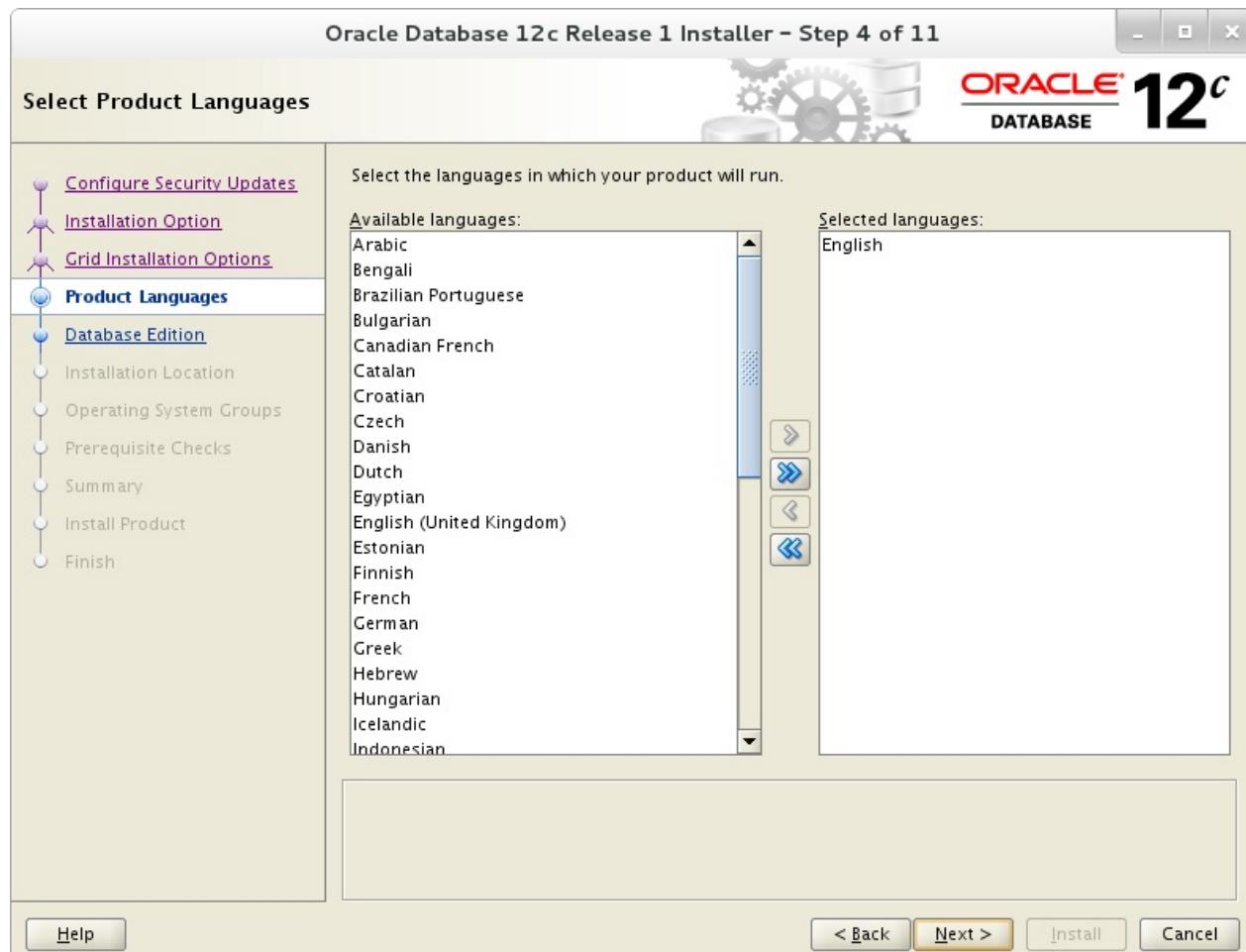
在此示例中，选择“Single Instance Database installation”：

6.3. Oracle Database



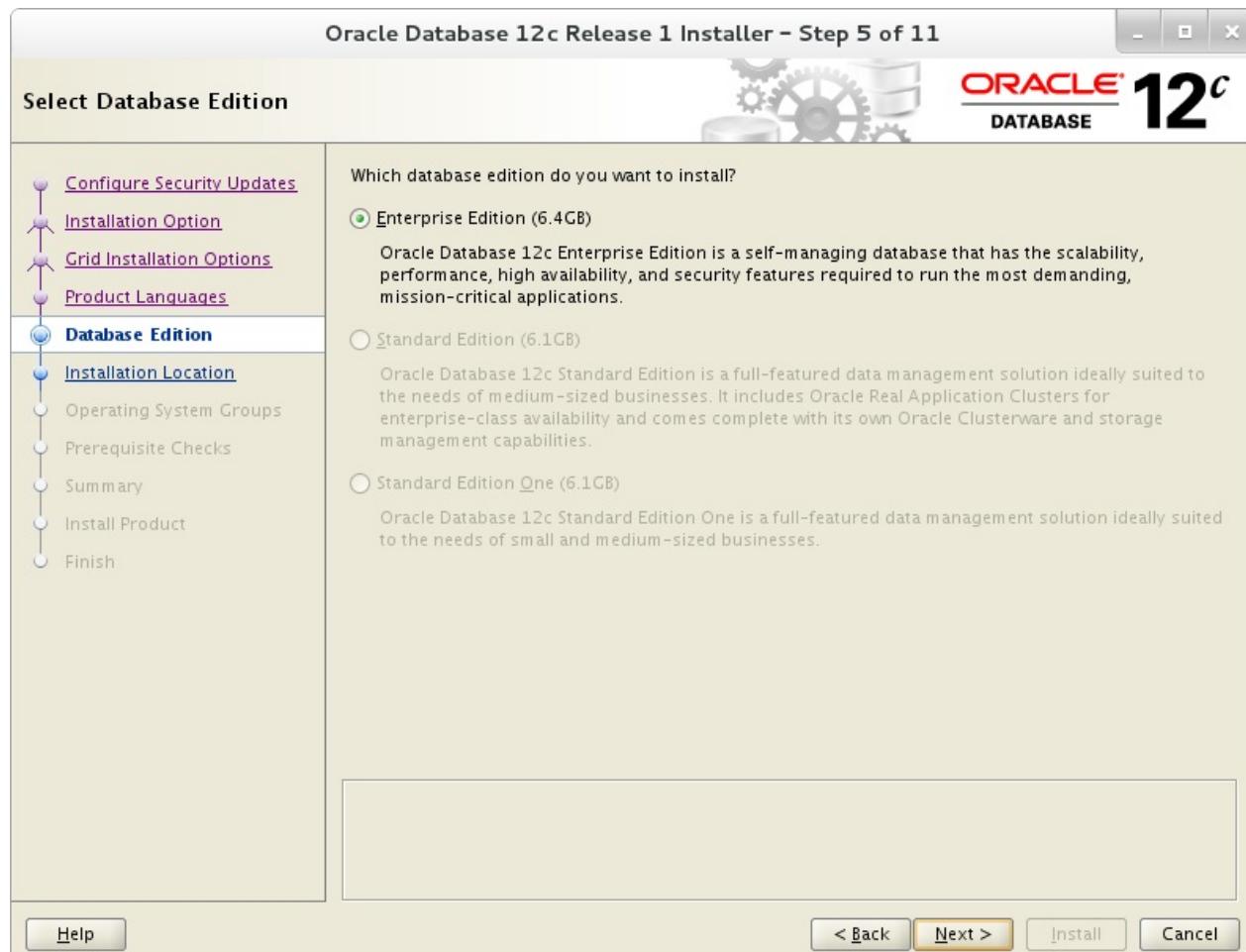
选择语言：

6.3. Oracle Database



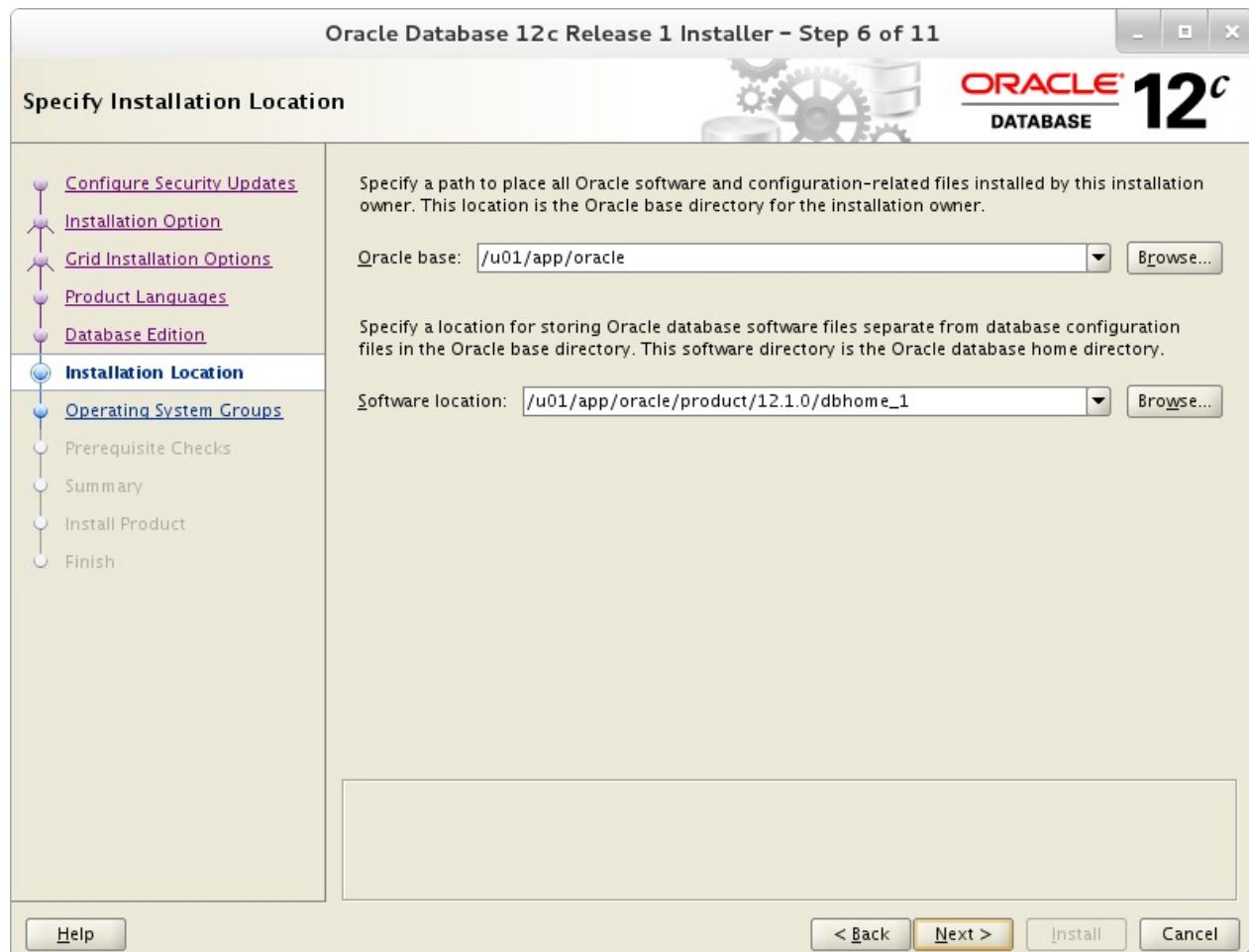
选择要安装的版本：

6.3. Oracle Database



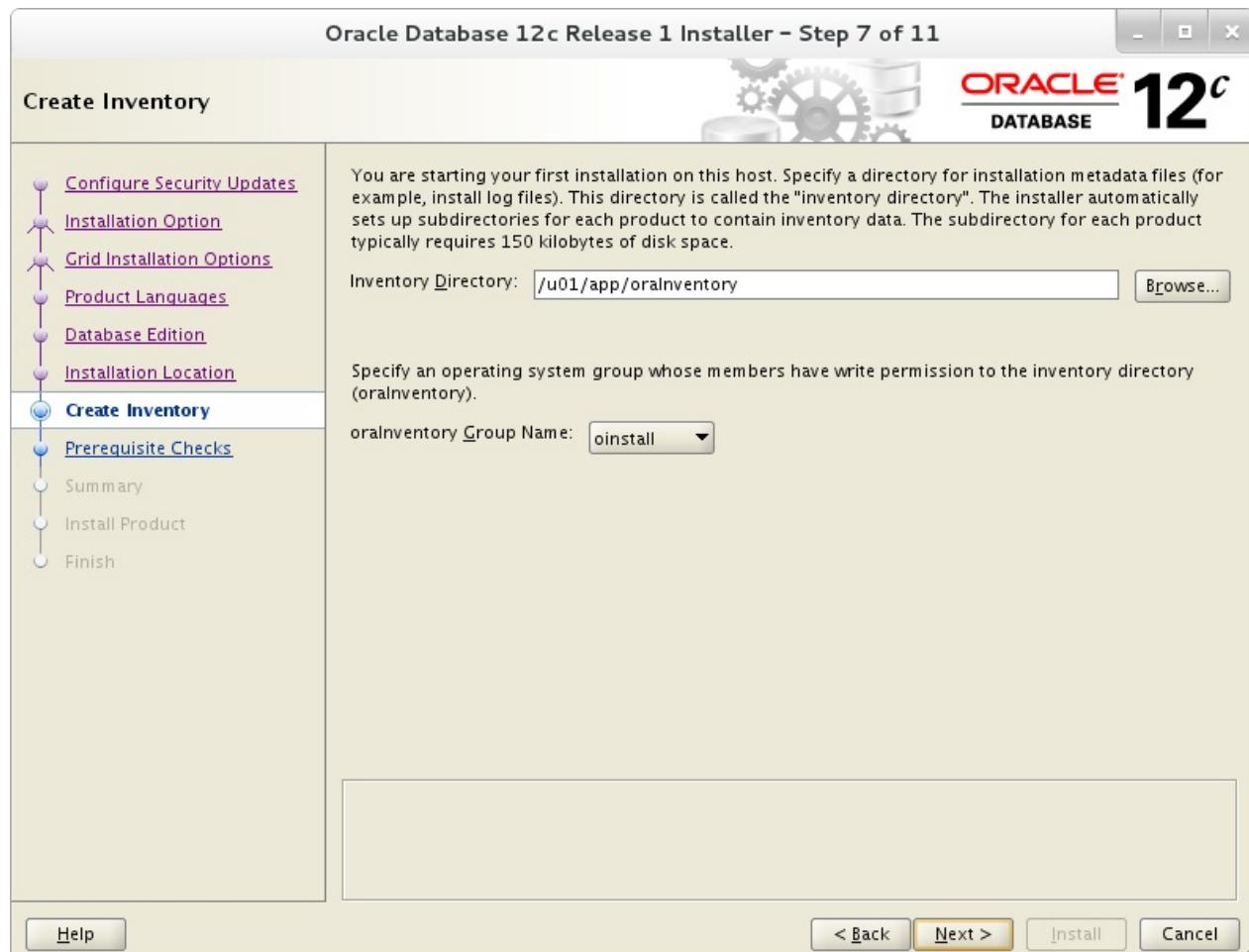
指定Oracle的基本目录和文件，在此示例中，保留默认值并继续下一步：

6.3. Oracle Database



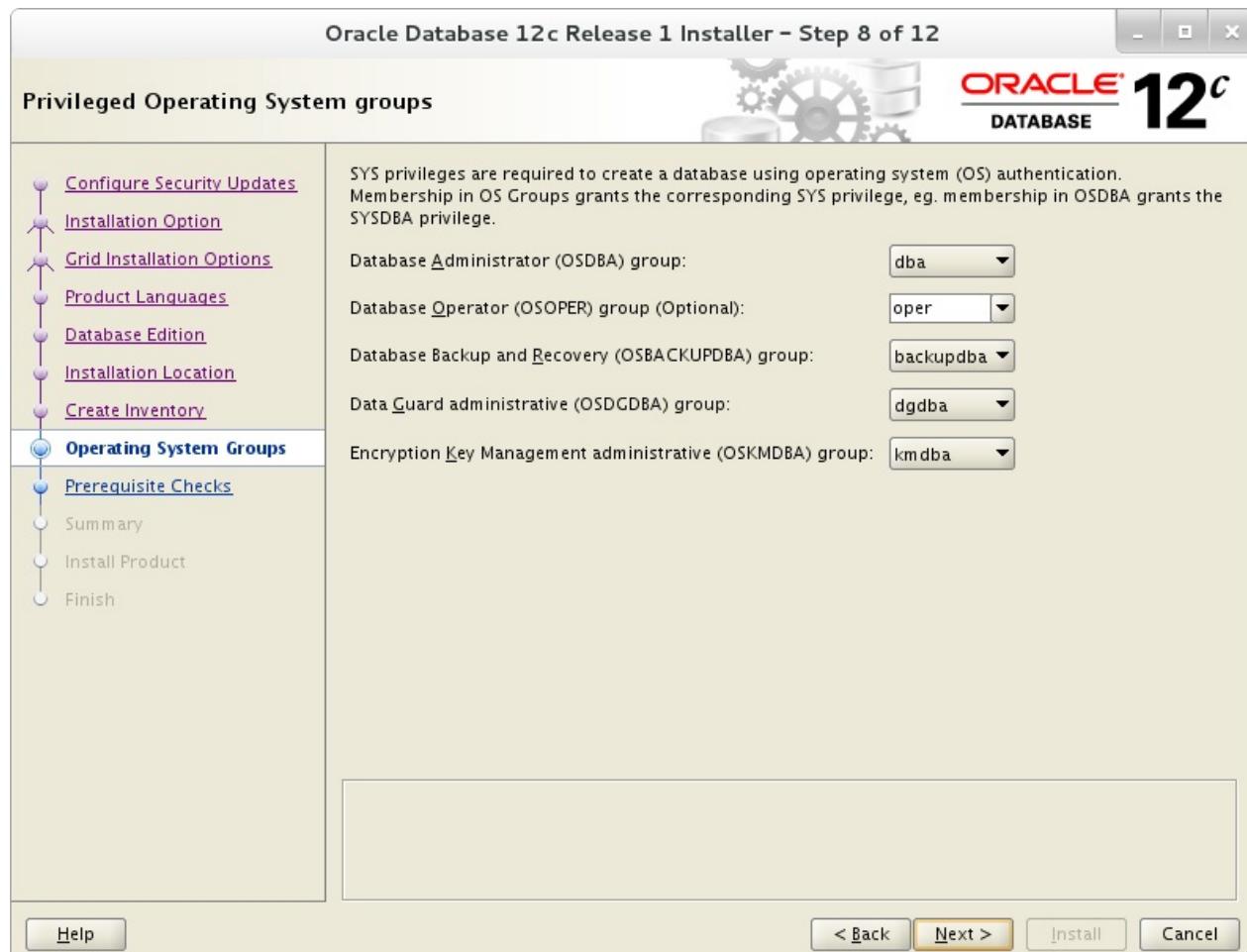
指定安装的目录，在此示例中，保留默认值并继续下一步：

6.3. Oracle Database



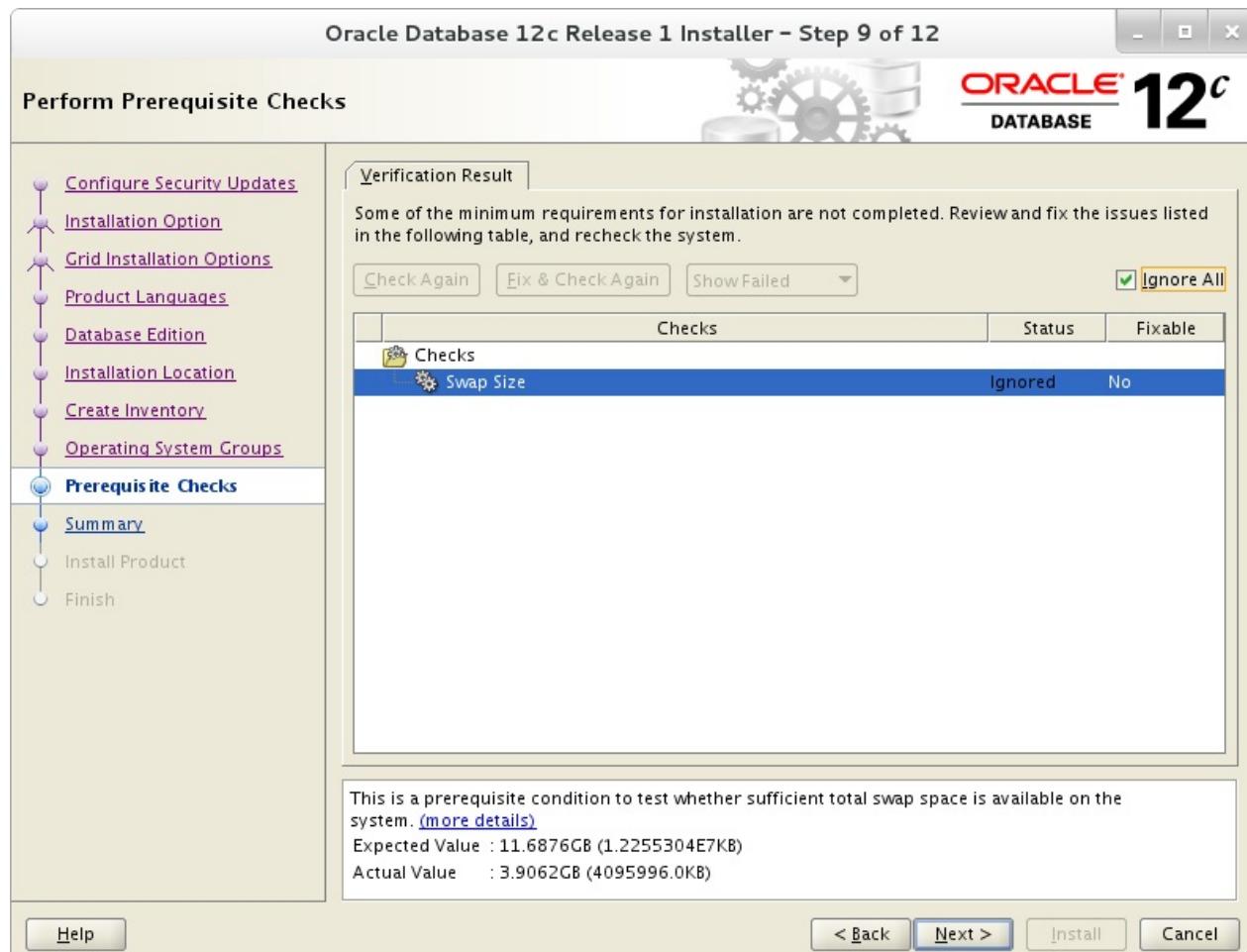
指定特权组，在此示例中，保留默认值并继续下一步：

6.3. Oracle Database



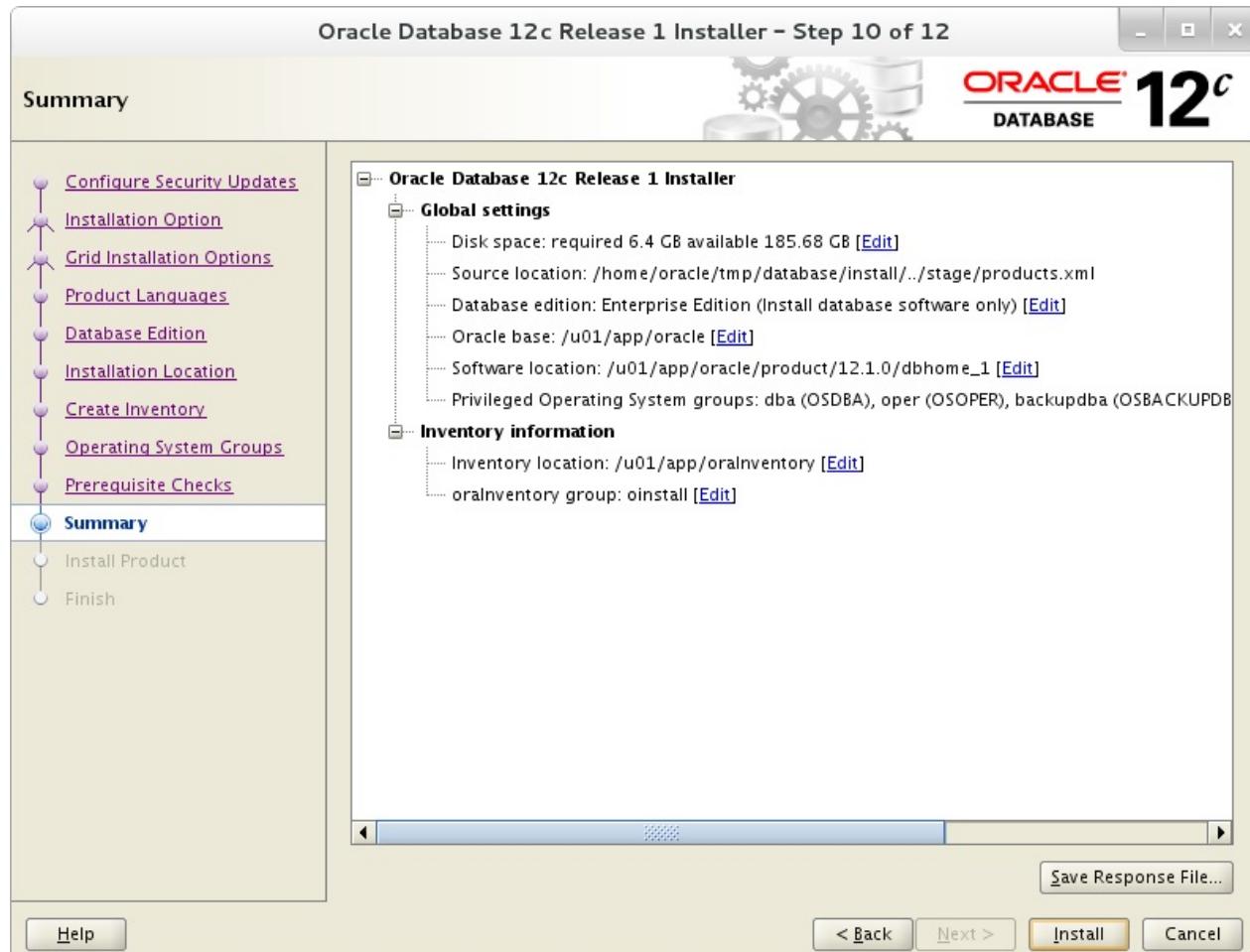
检查前提条件，如果某些设置未配置为推荐设置，则通知如下显示，再次确认：

6.3. Oracle Database



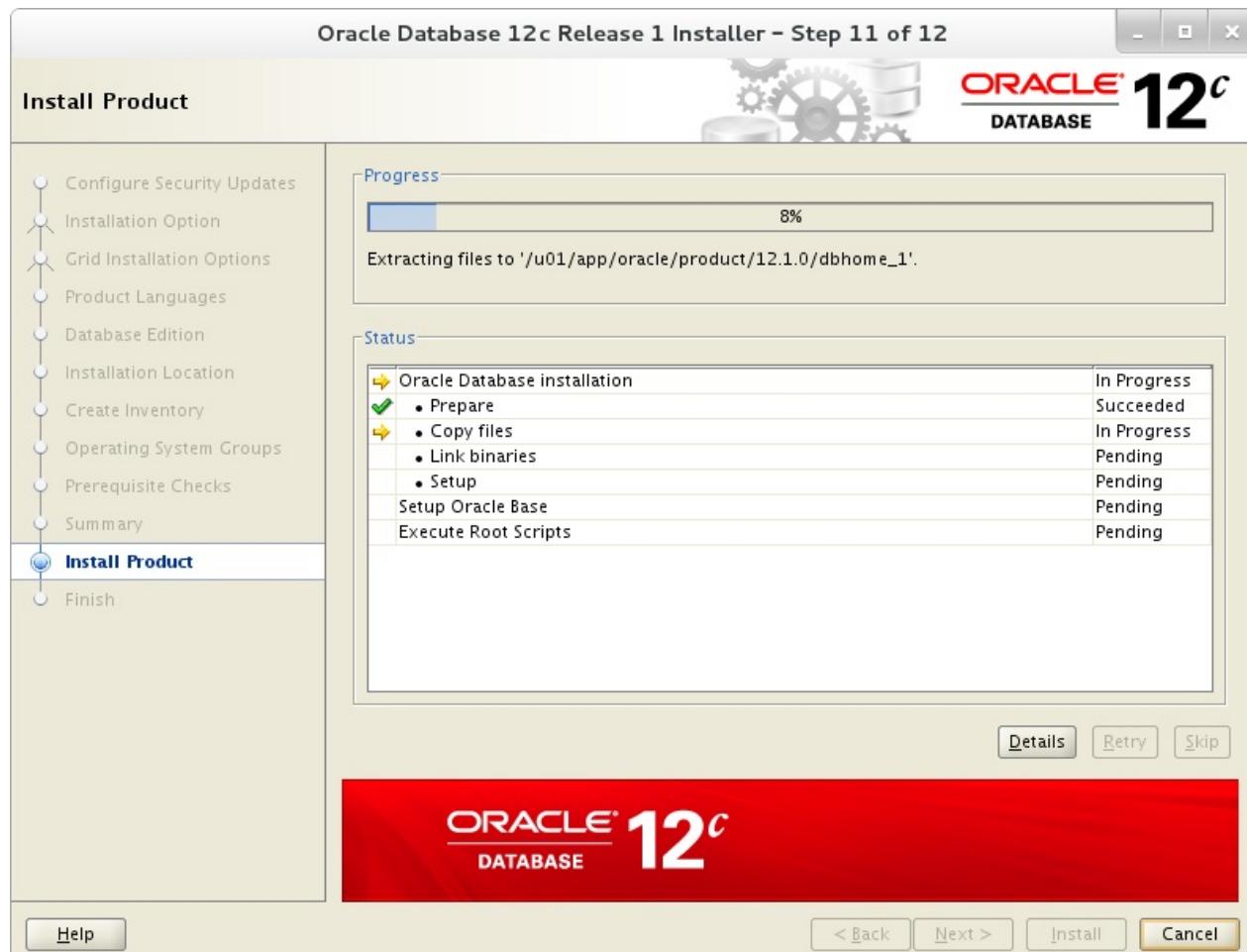
将显示配置的摘要，如果全部确定，单击“Install”：

6.3. Oracle Database



安装开始：

6.3. Oracle Database



显示以下界面，然后打开终端，使用root用户执行以下命令：



/u01/app/oralInventory/orainstRoot.sh

```
Changing permissions of /u01/app/oraInventory.  
Adding read,write permissions for group.  
Removing read,write,execute permissions for world.  
  
Changing groupname of /u01/app/oraInventory to oinstall.  
The execution of the script is complete.
```

```
/u01/app/oracle/product/12.1.0/dbhome_1/root.sh
```

```
Performing root user operation.
```

```
The following environment variables are set as:
```

```
ORACLE_OWNER= oracle  
ORACLE_HOME= /u01/app/oracle/product/12.1.0/dbhome_1
```

```
Enter the full pathname of the local bin directory: [/usr/local/  
bin]: # 回车
```

```
Copying dbhome to /usr/local/bin ...
```

```
Copying oraenv to /usr/local/bin ...
```

```
Copying coraenv to /usr/local/bin ...
```

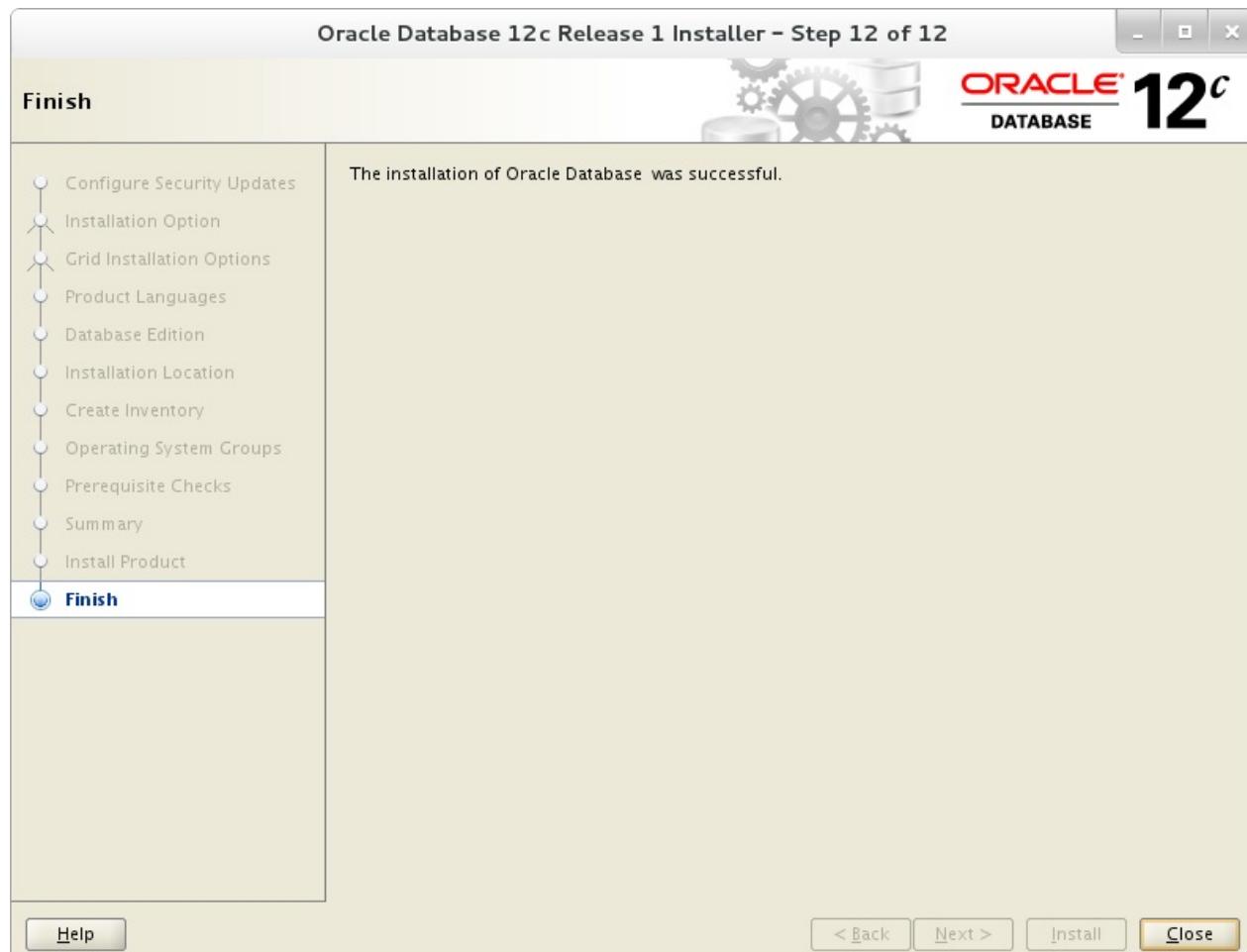
```
Creating /etc/oratab file...
```

```
Entries will be added to the /etc/oratab file as needed by  
Database Configuration Assistant when a database is created  
Finished running generic part of root script.
```

```
Now product-specific root actions will be performed.
```

安装完成，点击“Close”按钮：

6.3. Oracle Database



为Oracle用户设置环境变量：

编辑 `~/.bash_profile` 文件：

```
# 添加到最后
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export PATH=$PATH:$ORACLE_HOME/bin
```

```
source ~/.bash_profile
```

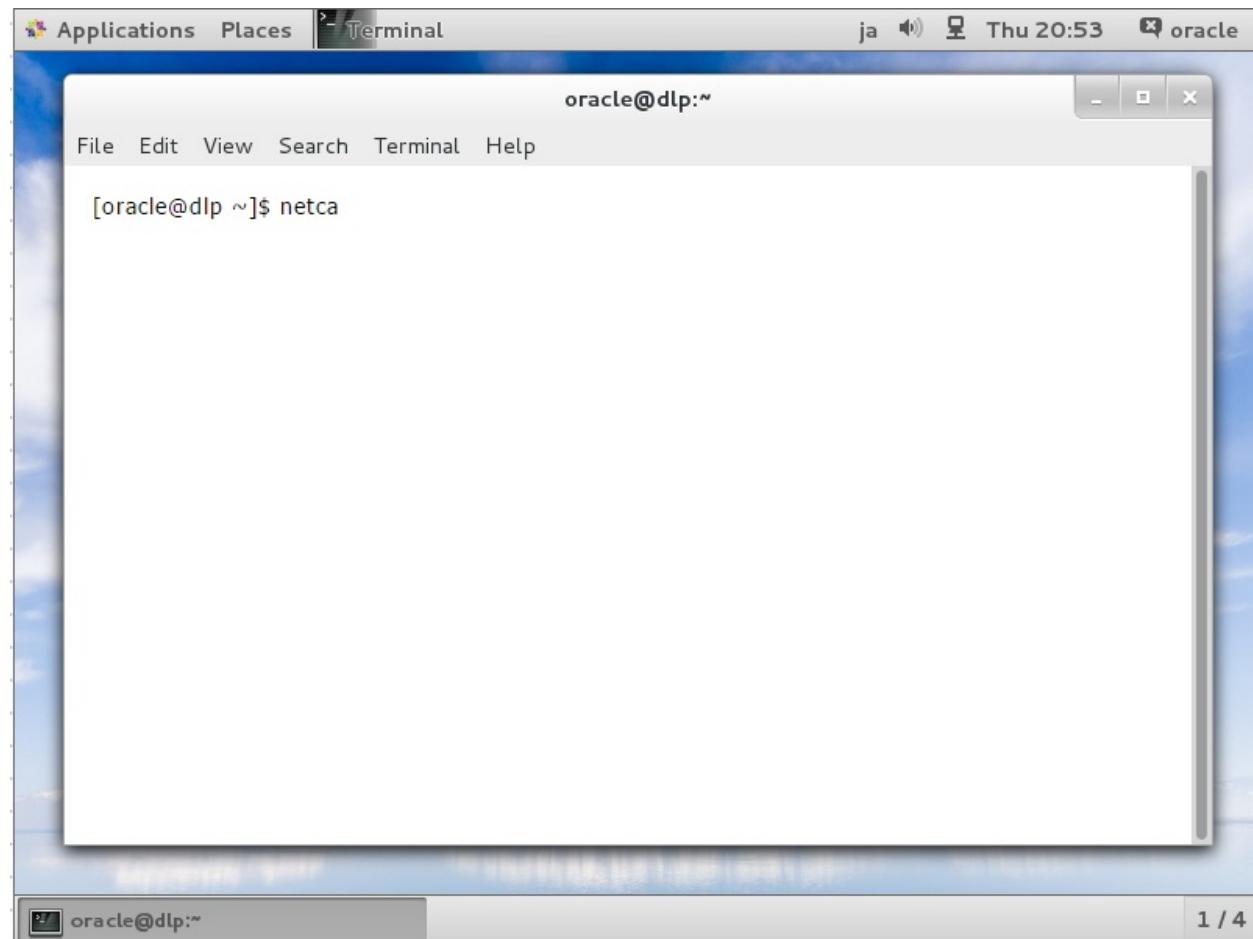
```
rm -rf tmp
```

6.3.3. 添加网络监听器

创建Oracle网络监听器，它是Oracle上的网络服务。

使用oracle管理员用户登录并输入命令 `netca` ，如下所示：

6.3. Oracle Database



A screenshot of a terminal window titled "Terminal". The window shows the command `[oracle@dlp ~]$ netca` being run. The terminal is located on a desktop with a blue sky and clouds background. The window title bar also displays "oracle@dlp:~".

选中“Listener Configuration”框，然后转到下一步：



下一步：

6.3. Oracle Database



设置监听器的名称：

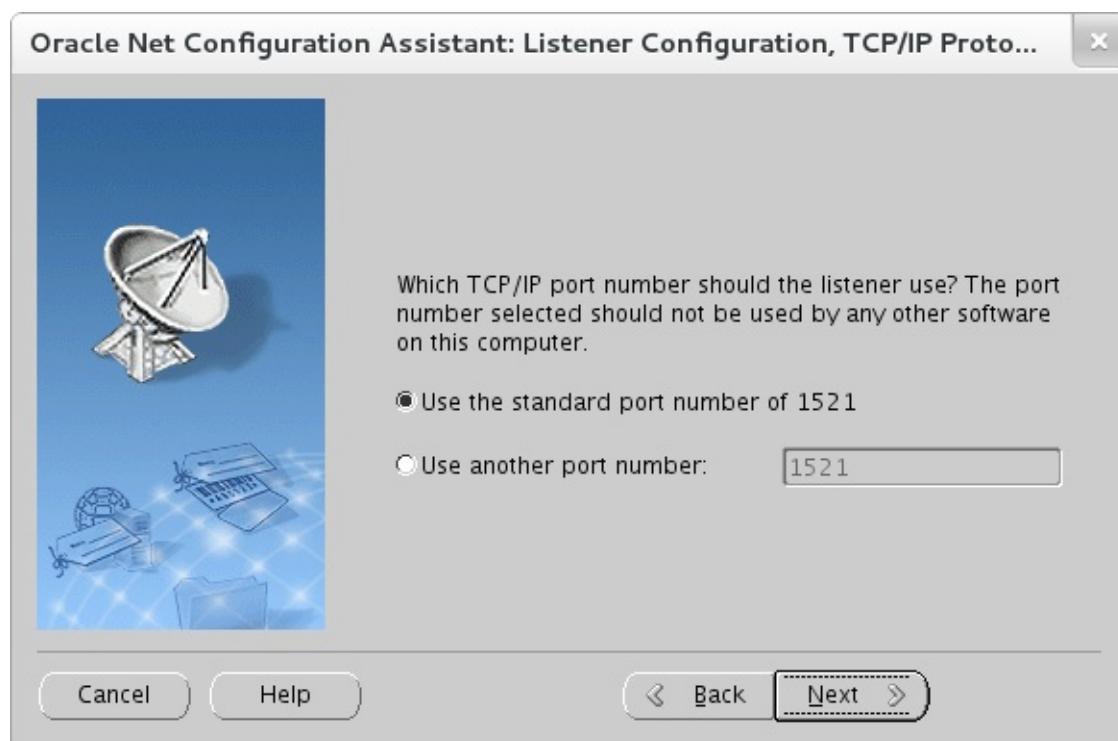


本例保持默认“TCP”：

6.3. Oracle Database



设置端口，本例保持默认：



如果要创建更多监听器，选择“Yes”。本例选择“No”：

6.3. Oracle Database



配置完成：



创建侦听器后，tnslsnr侦听您配置的端口，如下所示：

```
ss -napt
```

6.3. Oracle Database

State	Recv-Q	Send-Q	Local Address:Port	Peer
Address:Port				
LISTEN	0	128	*:22	
	*	*		
LISTEN	0	128	127.0.0.1:631	
	*	*		
LISTEN	0	100	127.0.0.1:25	
	*	*		
ESTAB	0	52	10.0.0.30:22	1
0.0.0.5:50113				
LISTEN	0	128	:::1521	
	:::*	users:(("tnslsnr",3988,9))		
LISTEN	0	128	:::22	
	:::*			
LISTEN	0	128	::1:631	
	:::*			
LISTEN	0	100	::1:25	
	:::*			

```
tnsping localhost
```

```
TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 0  
4-JUL-2015 01:03:07
```

```
Copyright (c) 1997, 2014, Oracle. All rights reserved.
```

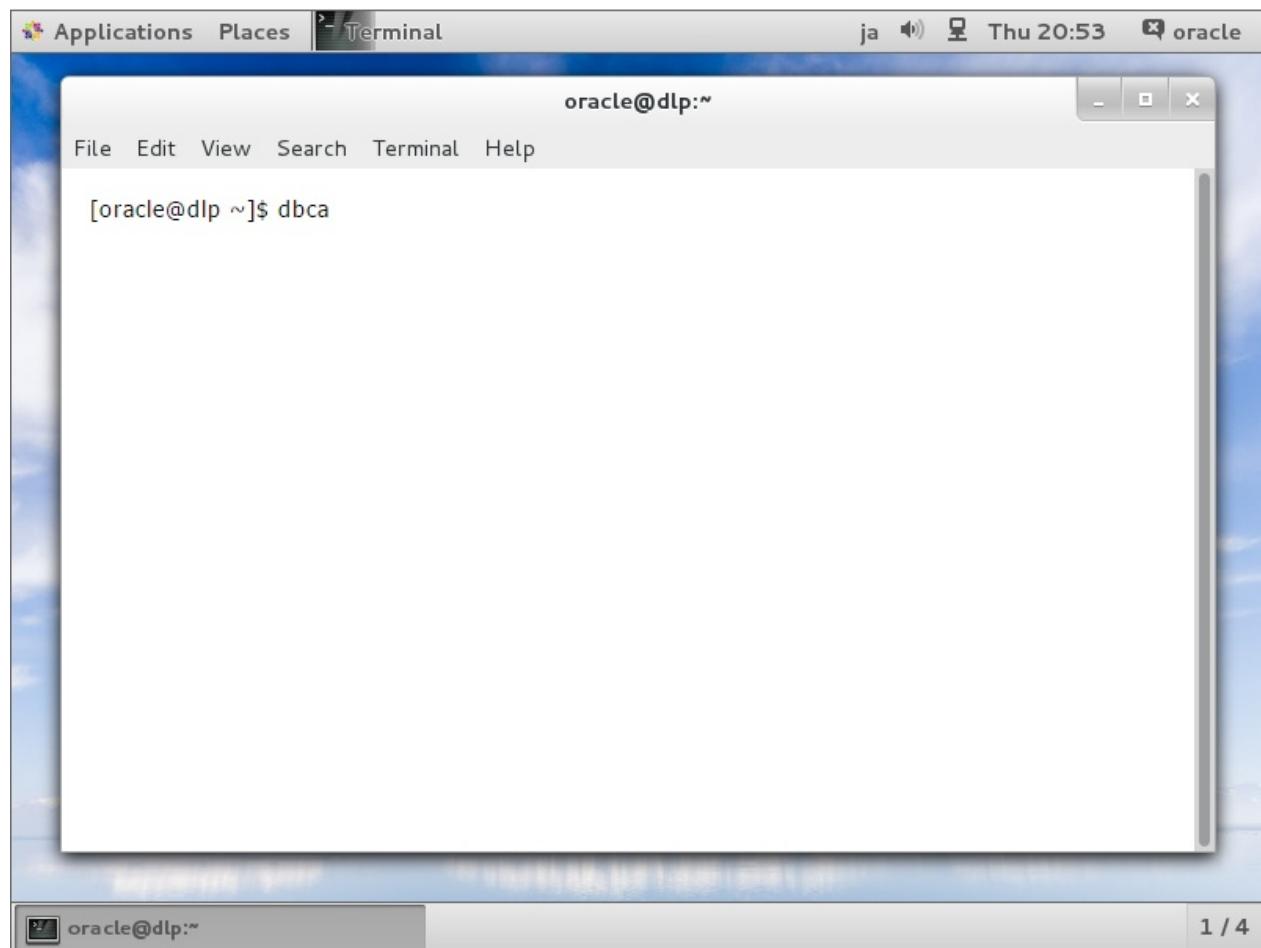
```
Used parameter files:
```

```
Used HOSTNAME adapter to resolve the alias  
Attempting to contact (DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=)  
(ADDRESS=(PROTOCOL=TCP)(HOST=127.0.0.1)  
(PORT=1521)))  
OK (0 msec)
```

6.3.4. 创建数据库

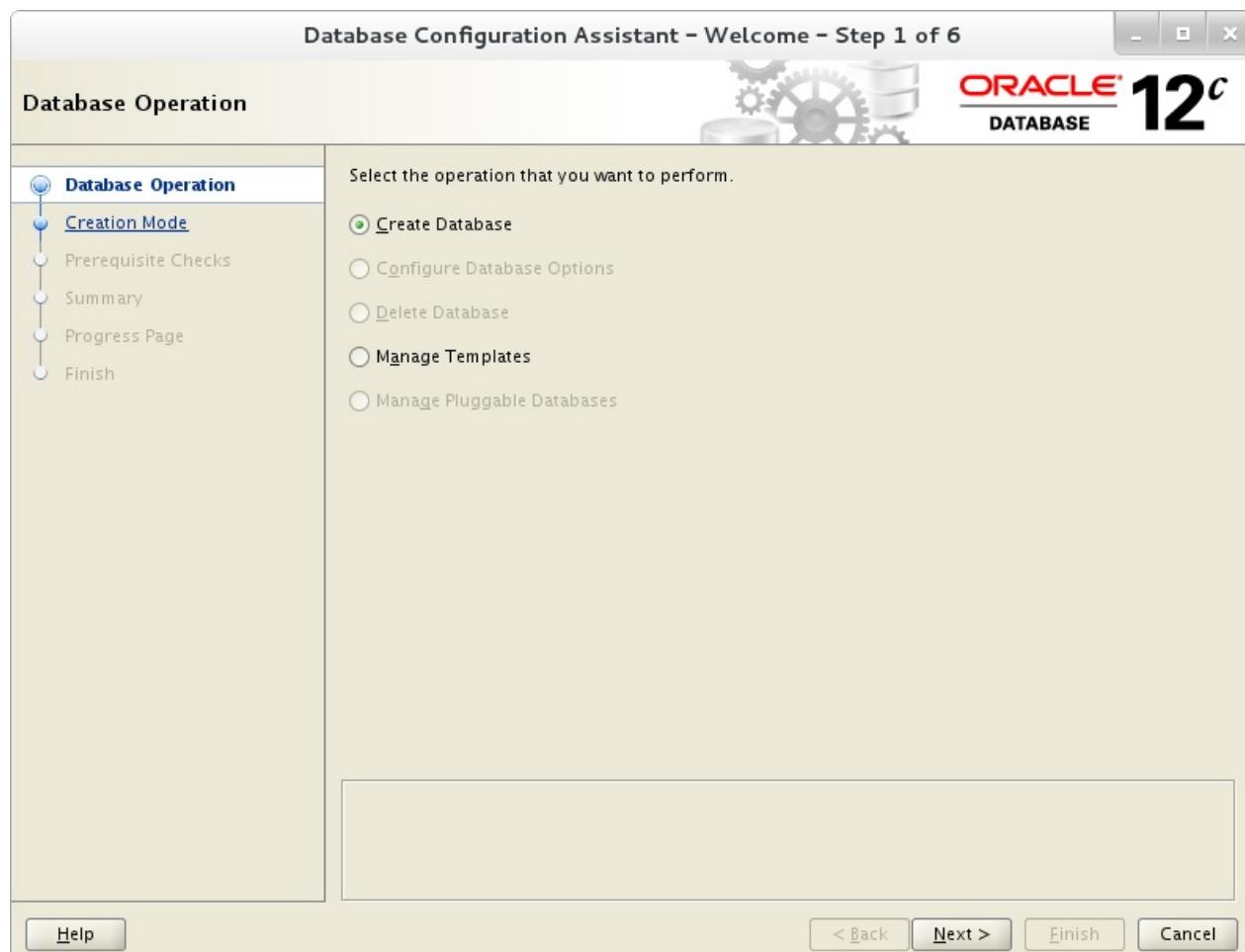
使用oracle管理员用户登录并输入命令 `dbca`，如下所示：

6.3. Oracle Database



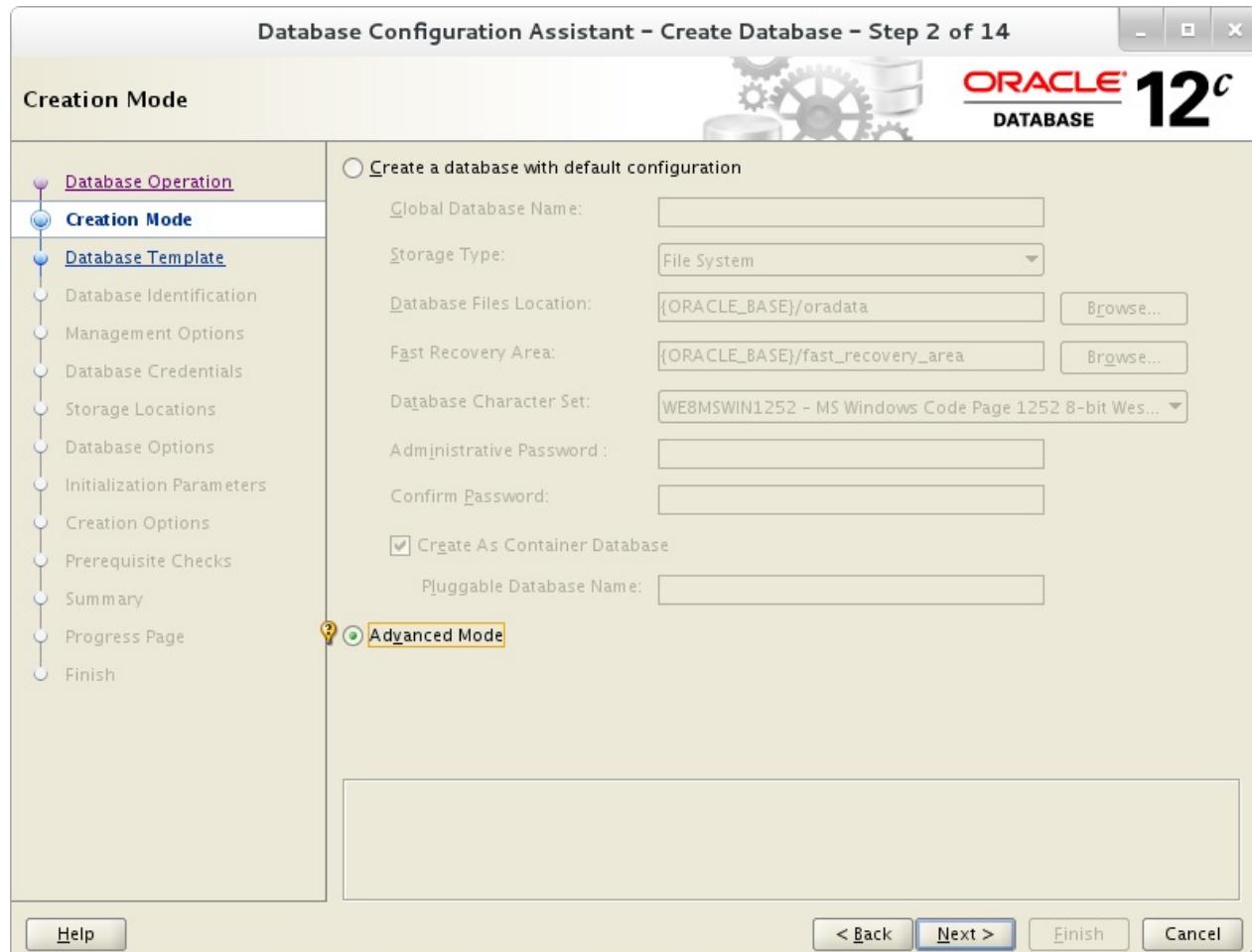
选择“Create Database”，然后转到下一步：

6.3. Oracle Database



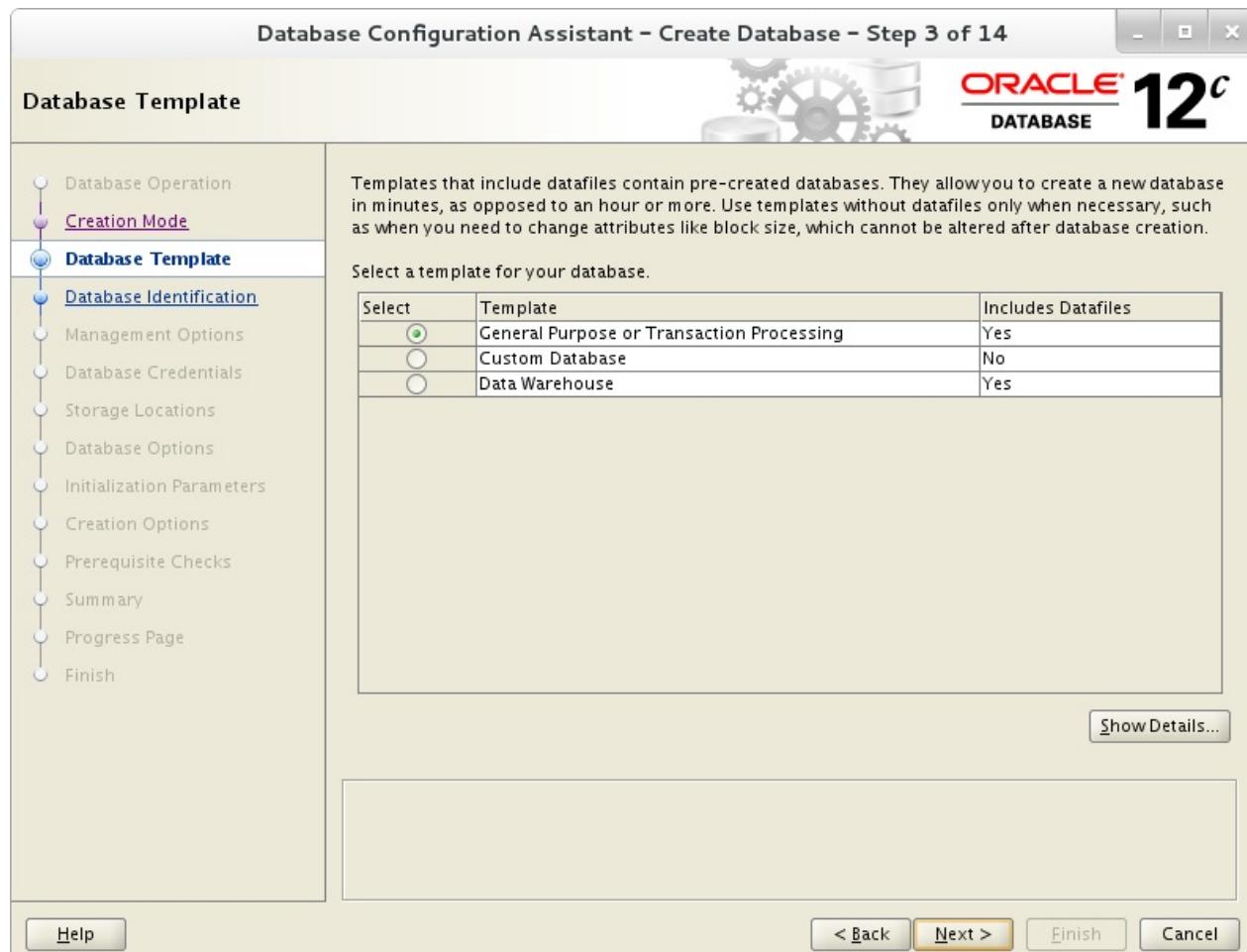
本例选择“Advanced Mode”，然后转到下一步：

6.3. Oracle Database



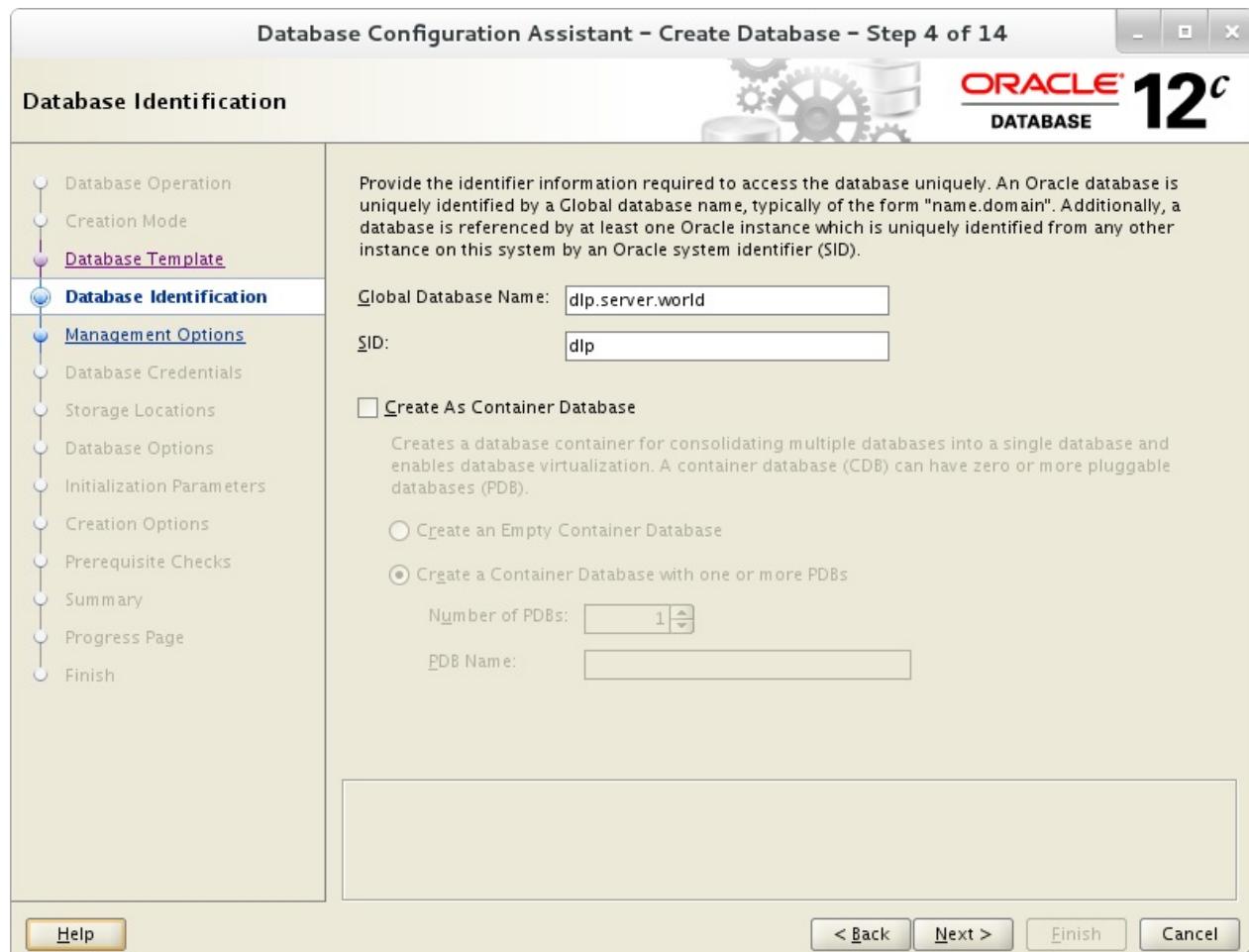
本例选择“General Purpose or Transaction Processing”，然后转到下一步：

6.3. Oracle Database



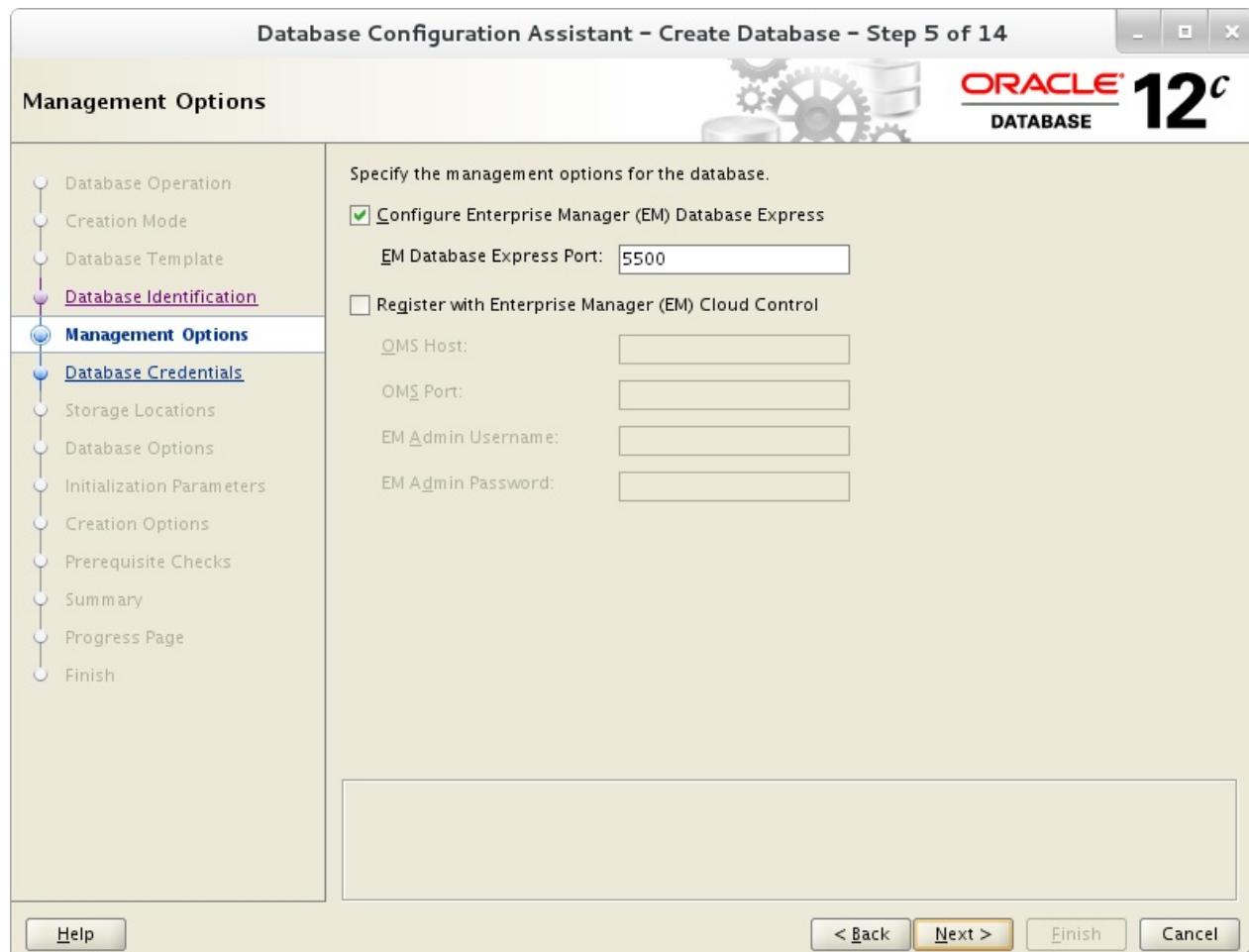
设置全局数据库名称和SID：

6.3. Oracle Database



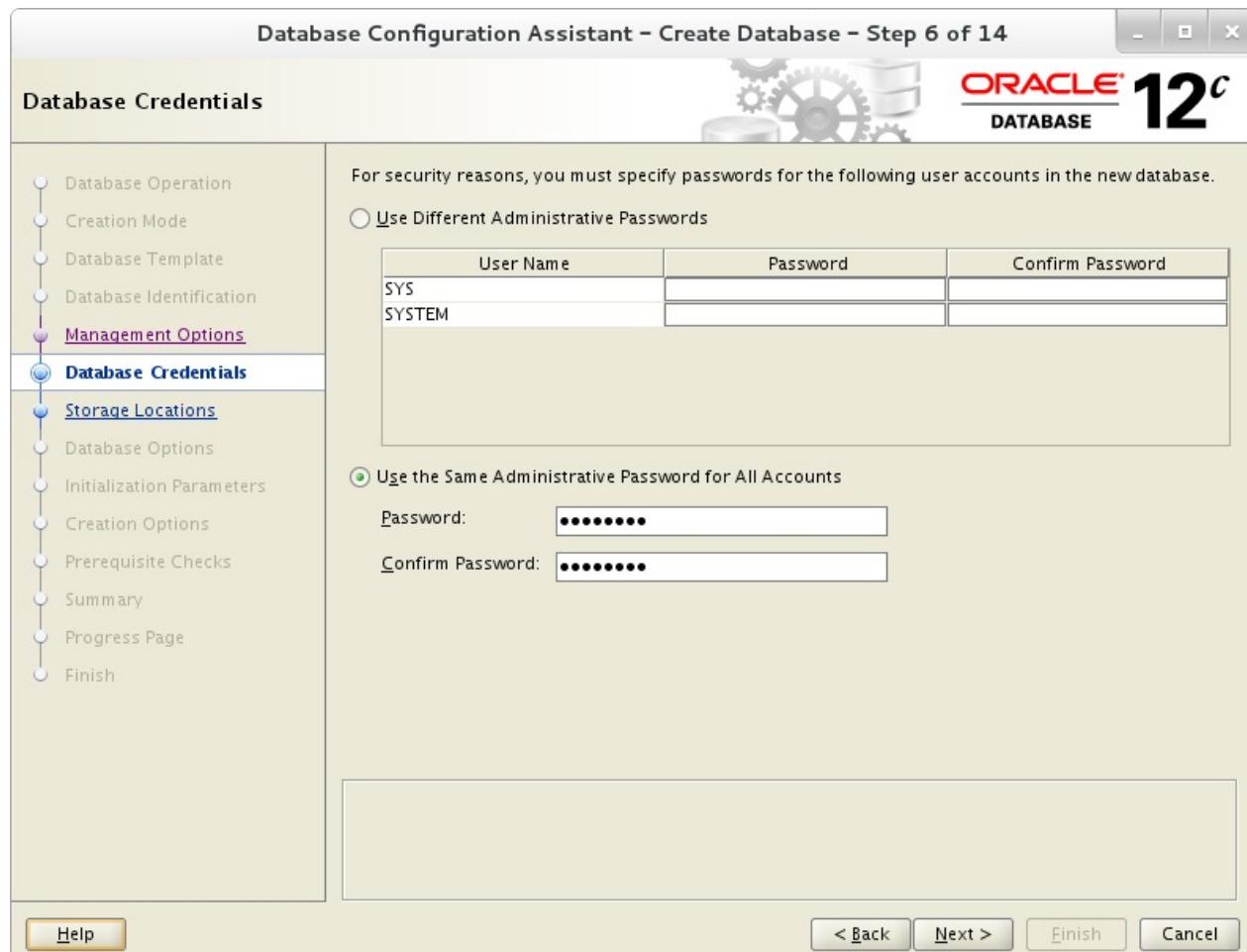
本例下一步保持默认：

6.3. Oracle Database



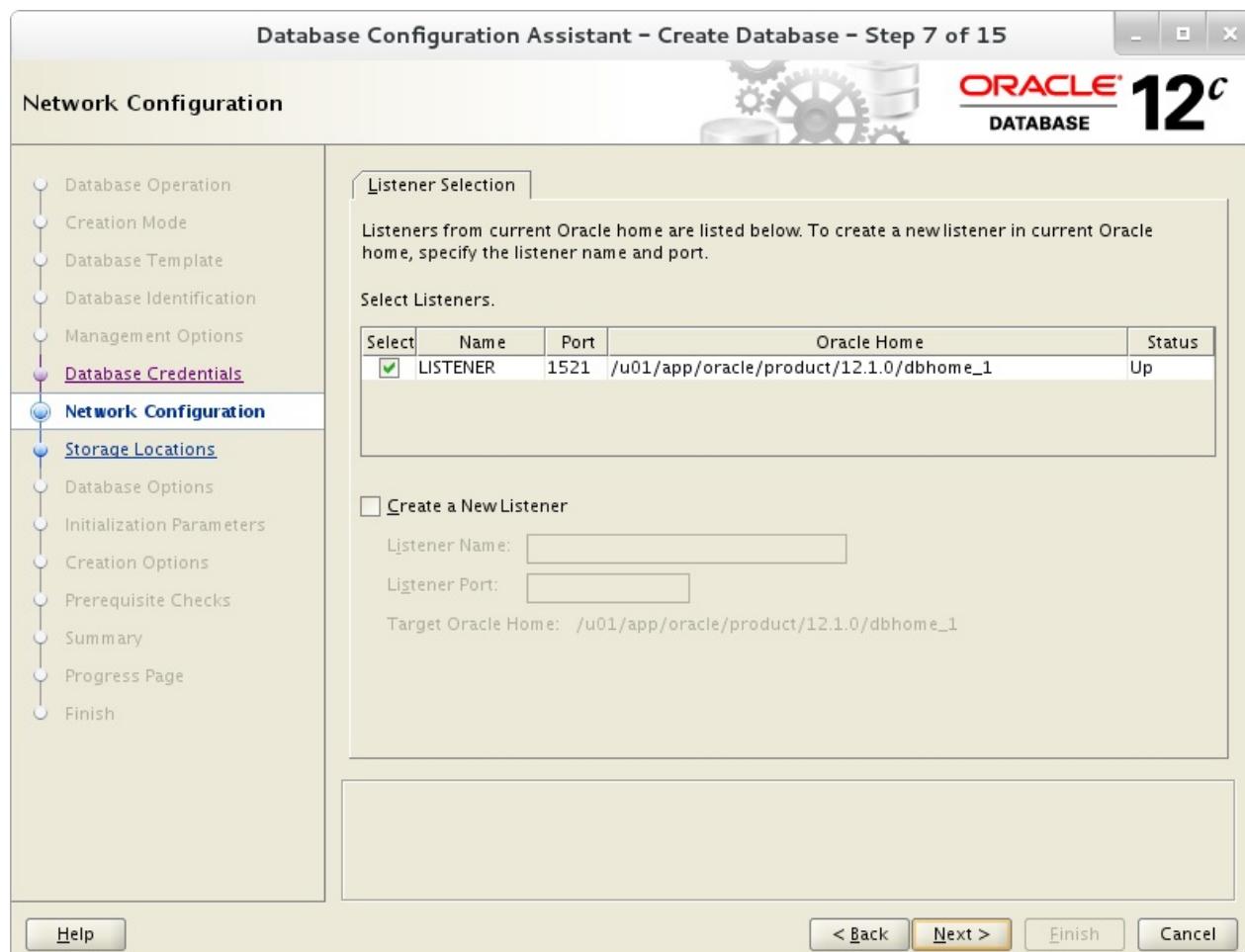
为用户设置密码以确保安全：

6.3. Oracle Database



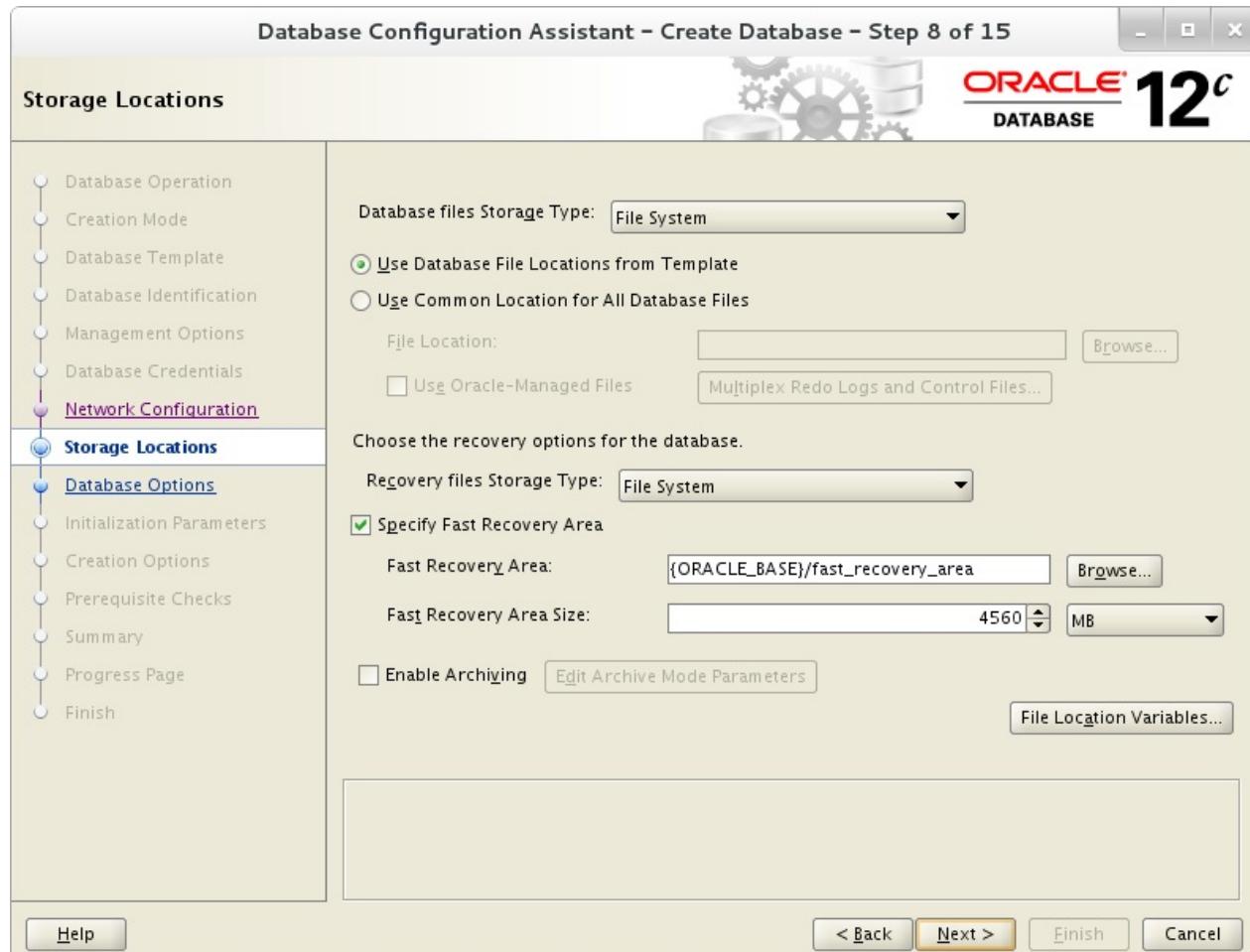
配置侦听器。本例保持默认值，然后转到下一步：

6.3. Oracle Database



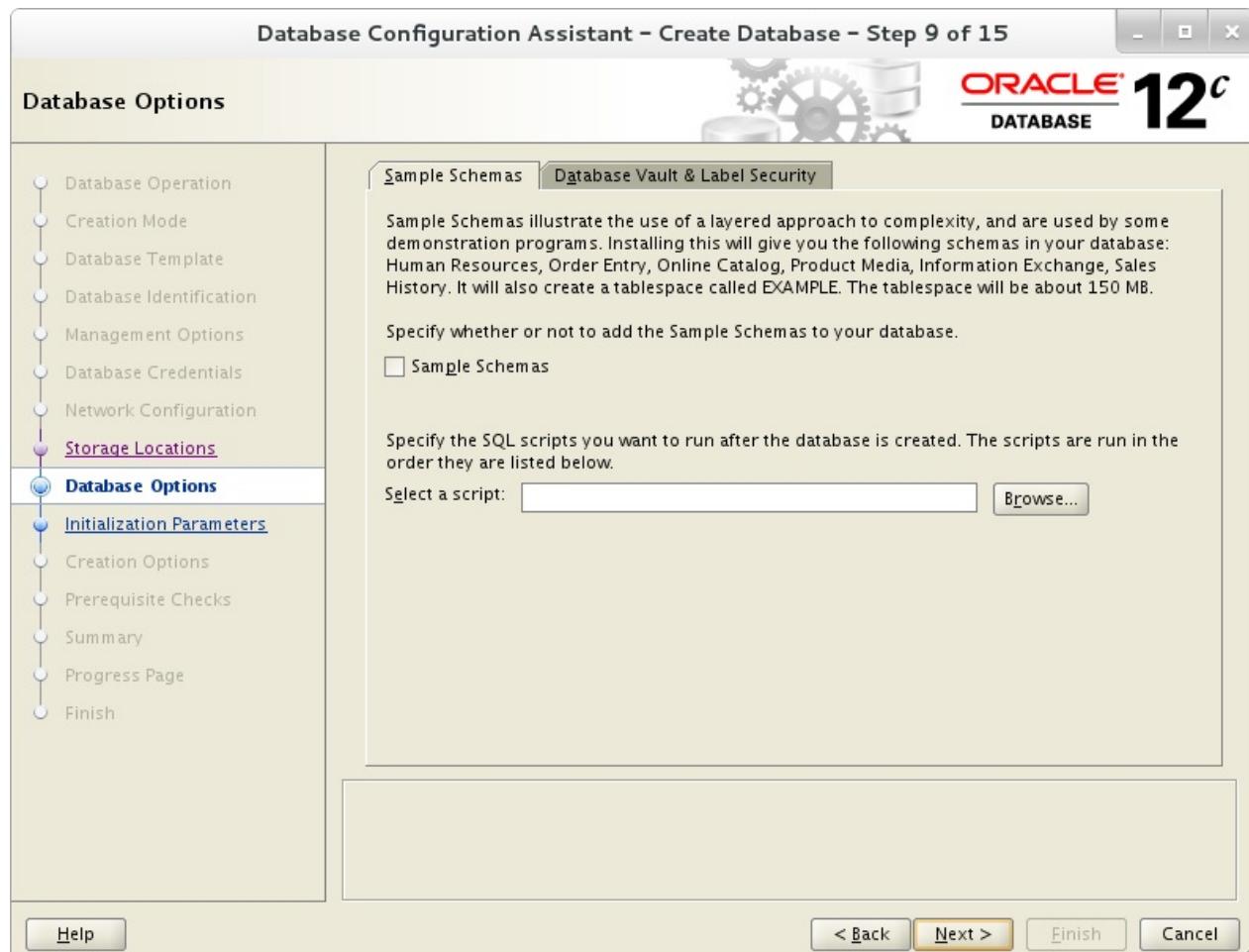
配置存储设置。本例保持默认值，然后转到下一步：

6.3. Oracle Database



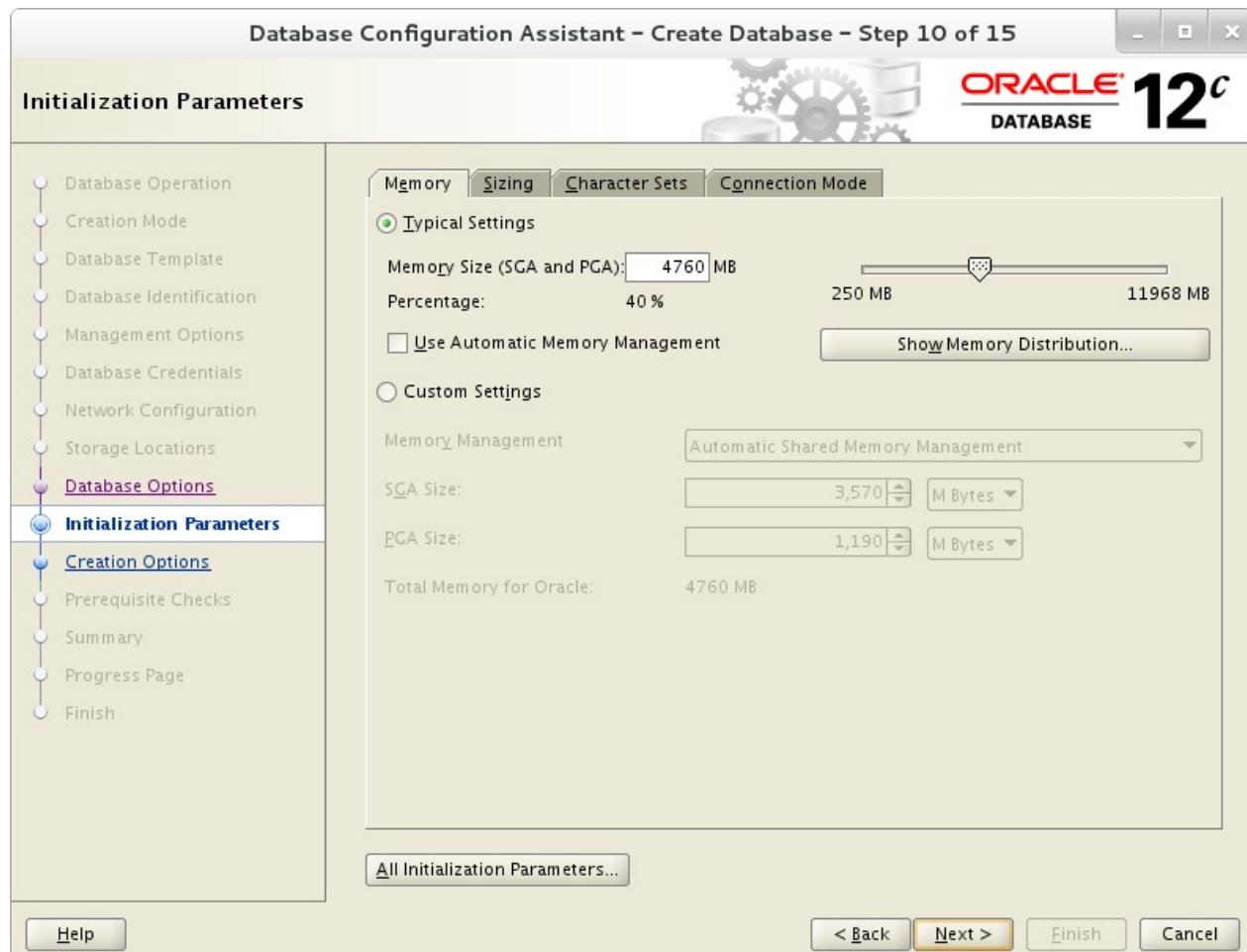
配置示例模式和脚本。如果要添加它们，请进行设置：

6.3. Oracle Database



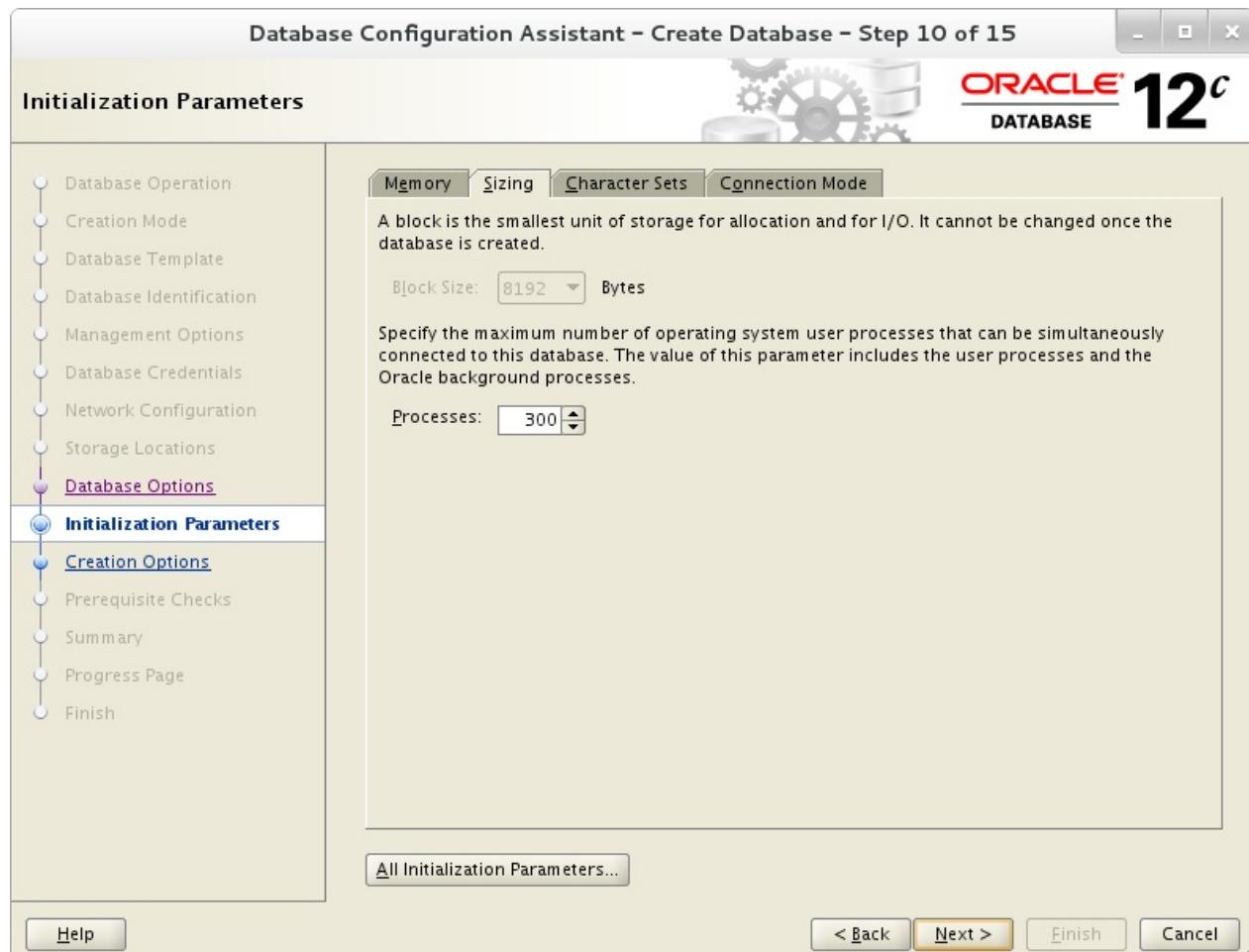
配置内存设置。设置后，转到下一个选项卡：

6.3. Oracle Database



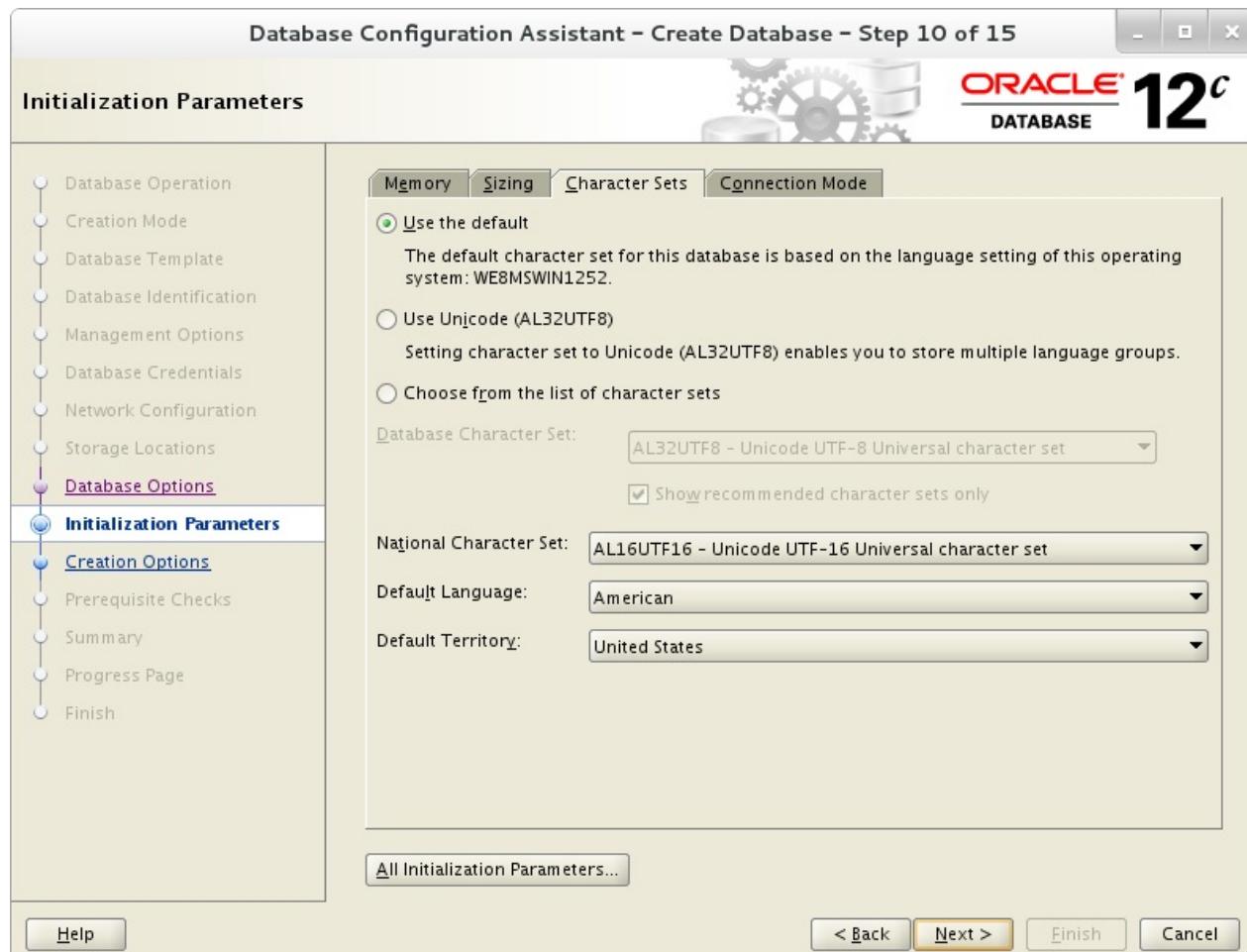
指定最大进程：

6.3. Oracle Database



设置字符设置：

6.3. Oracle Database



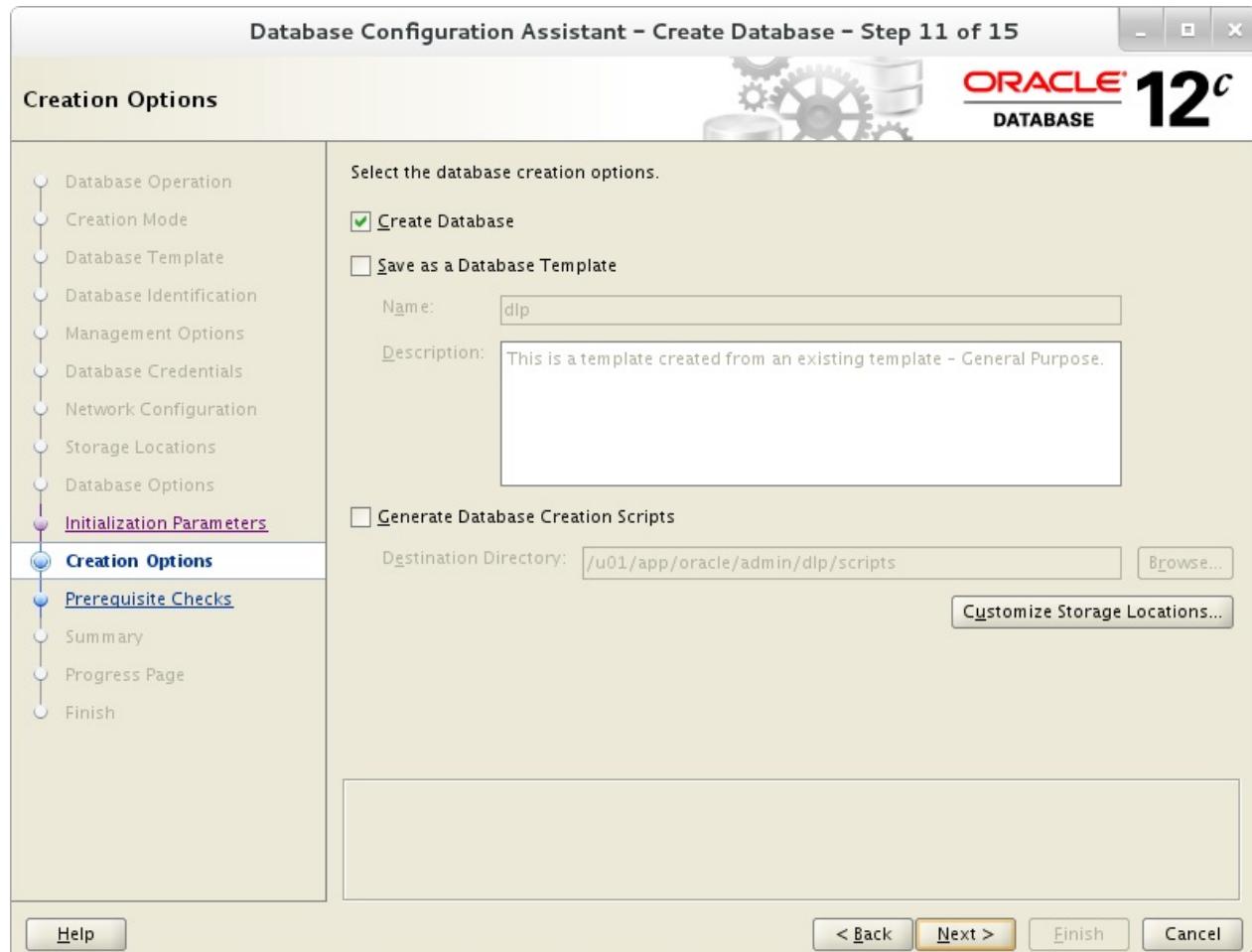
选择连接模式。如果服务器没有很多客户端，选择专用服务器模式；如果服务器有多个客户端，选择共享服务器模式：

6.3. Oracle Database



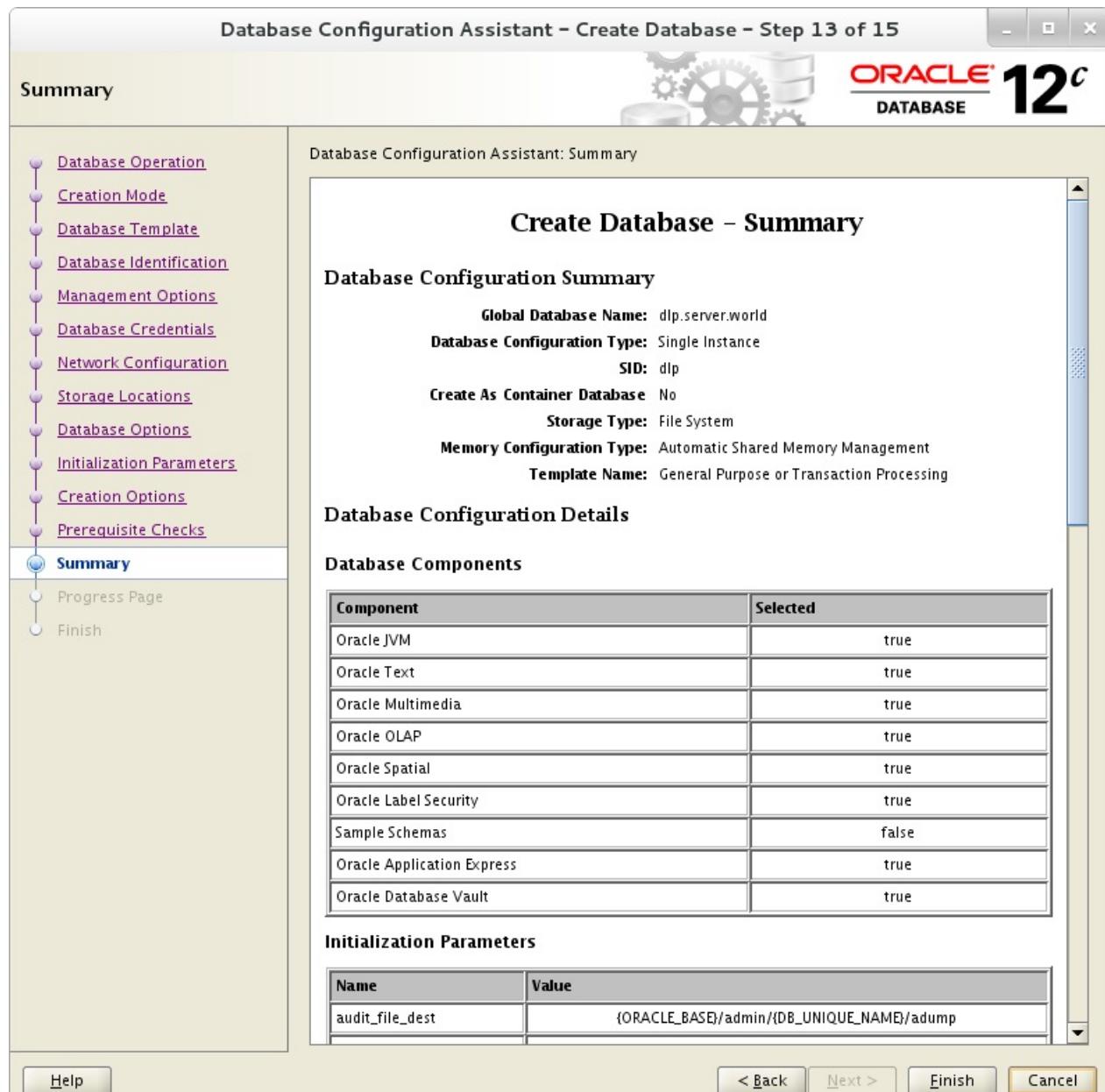
下一步：

6.3. Oracle Database



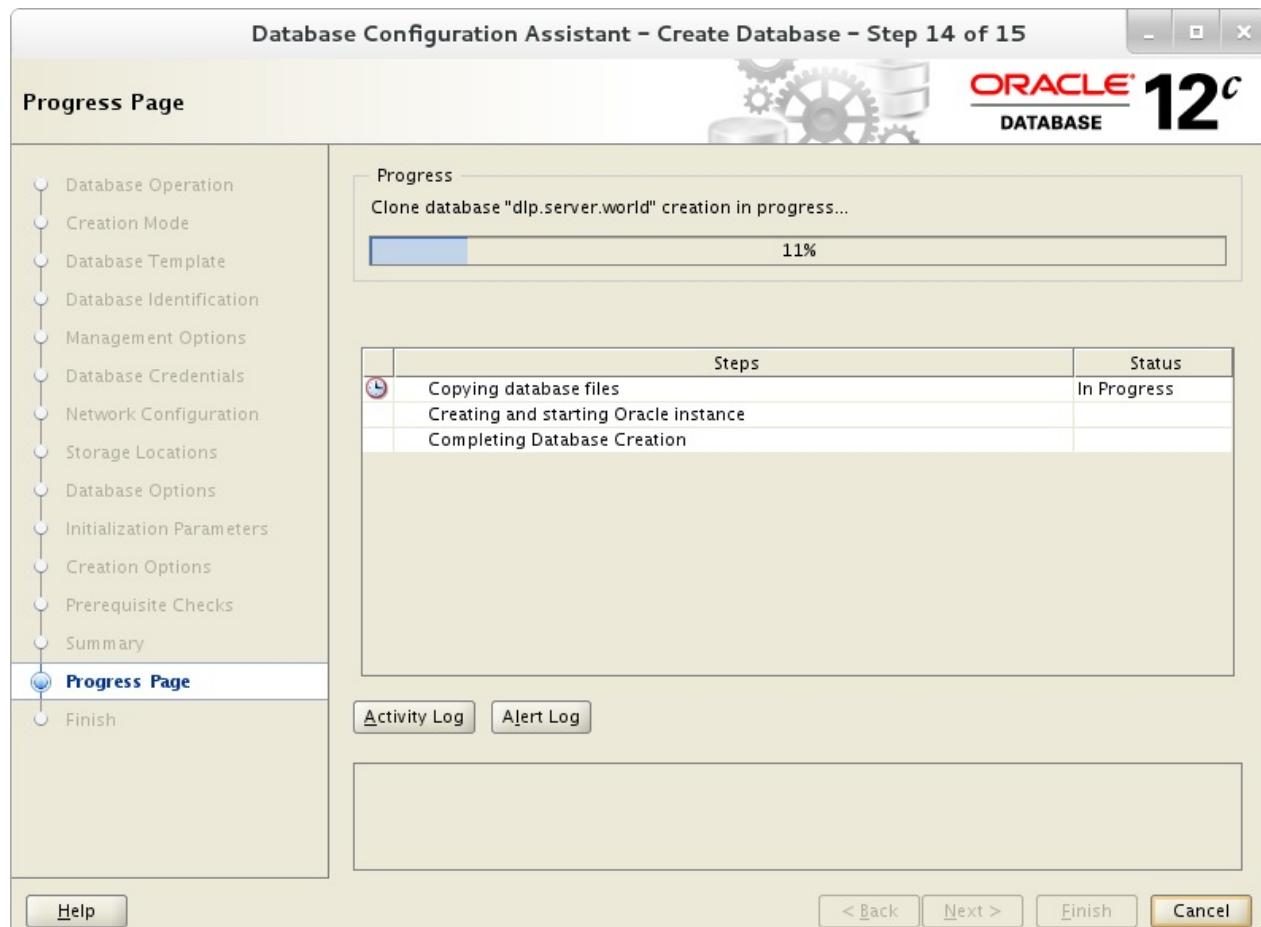
配置完成。点击“Finish”按钮完成：

6.3. Oracle Database



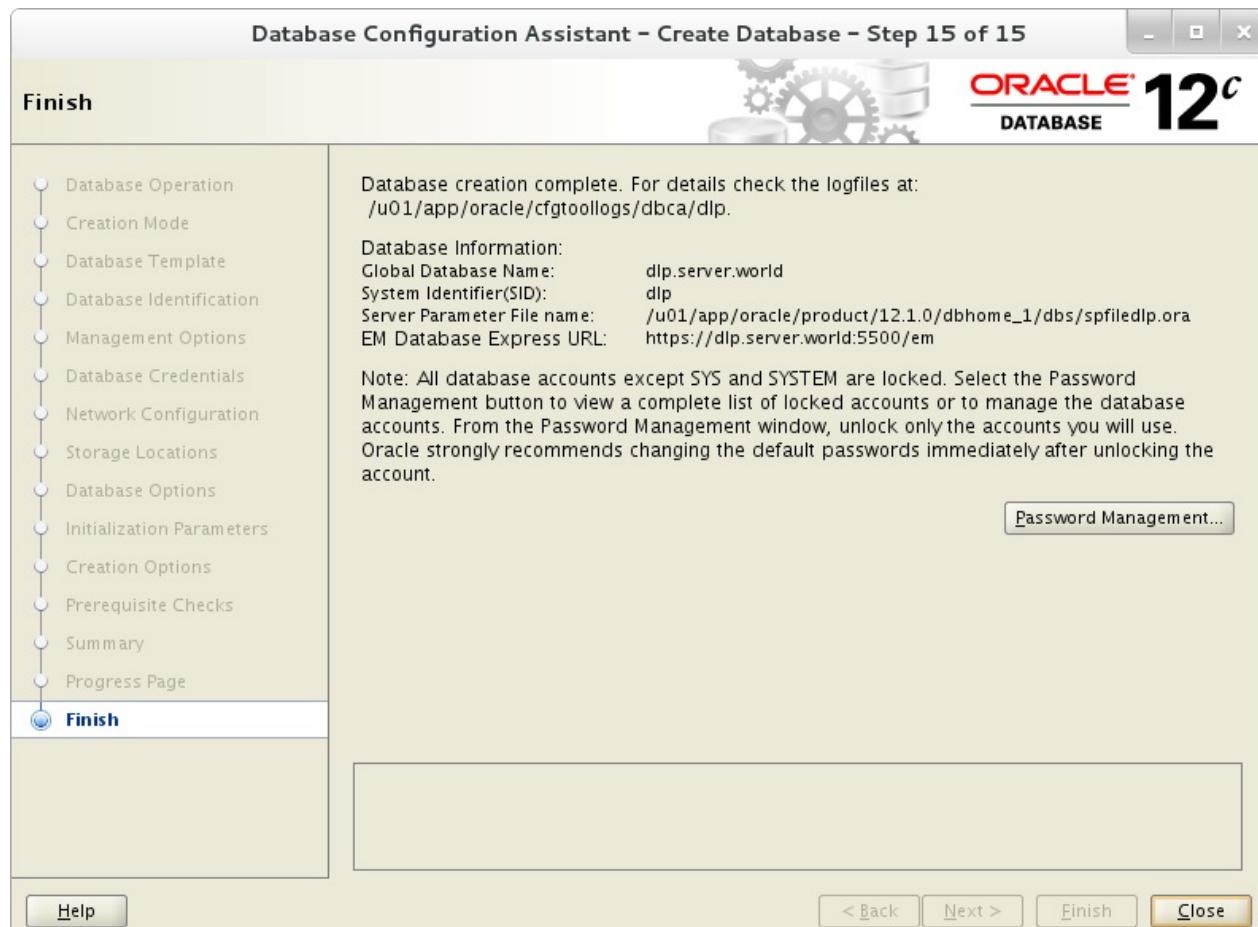
数据库创建开始：

6.3. Oracle Database



完成创建数据库后，单击“Close”完成：

6.3. Oracle Database



将数据库SID添加到环境变量：

编辑 /etc/oratab 文件：

```
# 如下更改  
dlp:/u01/app/oracle/product/12.1.0/dbhome_1:Y
```

编辑 ~/.bash_profile 文件：

```
# 添加到最后  
export ORACLE_SID=dlp
```

6.3.5. 使用企业管理器

访问企业管理器，可以在Web GUI上管理数据库。

创建数据库后，数据库服务运行时，可以访问企业管理器。访问数据库创建完成时显示的URL，然后显示登录表单，可以使用数据库用户登录。

6.3. Oracle Database



登录成功，这里可以管理数据库：

6.3. Oracle Database



6.3.6. 创建Systemd文件

为Oracle数据库服务创建Systemd文件。

以root用户身份登录并创建Systemd文件：

编辑 /etc/sysconfig/dlp.oracledb 文件：

6.3. Oracle Database

```
# 定义环境变量  
ORACLE_BASE=/u01/app/oracle  
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1  
ORACLE_SID=dlp
```

配置侦听器服务：

编辑 `/usr/lib/systemd/system/dlp@lsnrctl.service` 文件：

```
# 这里为示例，可自行修改  
[Unit]  
Description=oracle net listener  
After=network.target  
  
[Service]  
Type=forking  
EnvironmentFile=/etc/sysconfig/dlp.oracledb  
ExecStart=/u01/app/oracle/product/12.1.0/dbhome_1/bin/lsnrctl start  
ExecStop=/u01/app/oracle/product/12.1.0/dbhome_1/bin/lsnrctl stop  
User=oracle  
  
[Install]  
WantedBy=multi-user.target
```

配置数据库服务：

编辑 `/usr/lib/systemd/system/dlp@oracledb.service` 文件：

```
# 这里为示例，可自行修改
[Unit]
Description=oracle net listener
After=network.target lsnrctl.service

[Service]
Type=forking
EnvironmentFile=/etc/sysconfig/dlp.oracledb
ExecStart=/u01/app/oracle/product/12.1.0/dbhome_1/bin/dbstart /u01/app/oracle/product/12.1.0/dbhome_1
ExecStop=/u01/app/oracle/product/12.1.0/dbhome_1/bin/dbshut /u01/app/oracle/product/12.1.0/dbhome_1
User=oracle

[Install]
WantedBy=multi-user.target
```

```
systemctl daemon-reload
systemctl enable dlp@lsnrctl dlp@oracledb
```

6.4. Memcached

Memcached是一个高性能的分布式内存对象缓存系统，用于动态Web应用以减轻数据库负载。

6.4.1. 安装Memcached

```
yum -y install memcached
```

可以在/etc/sysconfig/memcached更改Memcached的设置。可以使用 man memcached 查看其他选项：

编辑 /etc/sysconfig/memcached 文件：

```
# 监听端口
PORT="11211"

# 进程所有者
USER="memcached"

# 最大连接数
MAXCONN="1024"

# 最大高速缓存大小 (MB)
CACHESIZE="64"

# 可以在这里指定选项
# 例如，默认侦听所有，要更改为只有本地，添加如下选项
OPTIONS="-l 127.0.0.1"
```

```
systemctl start memcached
systemctl enable memcached
```

如果Memcached用于远程主机，firewalld防火墙设置（Memcached使用端口11211/TCP）：

```
firewall-cmd --add-port=11211/tcp --permanent  
firewall-cmd --reload
```

6.4.2. 基本用法

这是Memcached与Telnet客户端连接时的基本用法。

```
yum -y install telnet # 安装Telnet客户端
```

连接到本地Memcached：

```
telnet localhost 11211
```

```
Trying ::1...  
Connected to localhost.  
Escape character is '^]'.
```

```
# 显示Memcached的状态  
stats  
STAT pid 10856  
STAT uptime 12222  
STAT time 1468912383  
.....  
.....  
STAT evictions 0  
STAT reclaimed 0  
END
```

```
# 保存数据（在内存上）  
# set [Key] [Flag] [Validity Term(sec)] [Data Size(byte)]  
# Flag : 0=压缩关闭, 1=压缩  
# Validity Term=0 表示不确定  
# after inputting command above, input a Value of the Key  
# 输入上述命令后，输入Key的值  
set test_key 0 0 10  
test_value  
STORED  
  
# 提取Key的值  
get test_key
```

6.4. Memcached

```
VALUE test_key 0 10
test_value
END

# 替换Key的值
replace test_key 0 0 11
test_value2
STORED
get test_key
VALUE test_key 0 11
test_value2
END

# 追加Key的值
append test_key 0 0 5
,test
STORED
get test_key
VALUE test_key 0 16
test_value2,test
END

# 前缀Key的值
prepend test_key 0 0 6
test1,
STORED
get test_key
VALUE test_key 0 22
test1,test_value2,test
END

# 删除Key
delete test_key
DELETED

# 递增Key的值
set mycounter 0 0 1
1
STORED
incr mycounter 1
2
get mycounter
```

6.4. Memcached

```
VALUE mycounter 0 1
2
END
```

```
# 递减Key的值
decr mycounter 1
1
get mycounter
VALUE mycounter 0 1
1
END
```

```
# 删除内存上的所有缓存数据
flush_all
OK
```

对于CAS（Check And Set检查和设置）操作，使用 `cas` 命令行如下：

```
# 提取CAS ID的值
# 在下面的示例中，CAS ID = 15
gets test_key
VALUE test_key 0 10 15
test_value
END

# 使用cas命令更新数据
# cas [Key] [Flag] [validity term(sec)] [data size(byte)] [CAS ID]
cas test_key 0 0 11 15
test2_value
STORED
gets test_key
VALUE test_key 0 11 16
test2_value
END
```

6.4.3. 在Python上使用

```
yum -y install python-memcached # 安装Python Memcached客户端库
```

6.4. Memcached

在Python上的基本用法：

编辑 use_memcache.py 文件：

```
#!/usr/bin/env python

import memcache

client = memcache.Client(["127.0.0.1:11211"], cache_cas=True)

# 设置并获取Key
client.set("key01", "value01")
print "key01.value :", client.get("key01")

# 追加并获取Key
client.append("key01", ",value02")
print "key01.value :", client.get("key01")

client.set("key02", 1)

# 递增
client.incr("key02", 100)
print "key02.value :", client.get("key02")

# 递减
client.decr("key02", 51)
print "key02.value :", client.get("key02")

# CAS
client.set("key03", "value03")
print "key03.value :", client.gets("key03")
print "key03.casid :", client.cas_ids["key03"]
client.cas("key03", "value04")
print "key03.value :", client.gets("key03")
```

运行：

```
python use_memcache.py
```

```
key01.value : value01
key01.value : value01,value02
key02.value : 101
key02.value : 50
key03.value : value03
key03.casid : 297
key03.value : value04
```

6.4.4. 在PHP上使用

```
yum --enablerepo=epel -y install php-pecl-memcached # 从EPEL安装
PHP Memcached客户端模块
```

在PHP上的基本用法：

编辑 `use_memcache.php` 文件：

6.4. Memcached

```
<?php
$memcache = new Memcached();
$memcache->addServer('localhost', 11211);
$memcache->setOption(Memcached::OPT_COMPRESSION, false);

// 设置并获取Key
$memcache->set('key01', 'value01');
print 'key01.value : ' . $memcache->get('key01') . "\n";

// 追加并获取Key
$memcache->append('key01', ',value02');
print 'key01.value : ' . $memcache->get('key01') . "\n";

$memcache->set('key02', 1);
print 'key02.value : ' . $memcache->get('key02') . "\n";

// 递增
$memcache->increment('key02', 100);
print 'key02.value : ' . $memcache->get('key02') . "\n";

// 递减
$memcache->decrement('key02', 51);
print 'key02.value : ' . $memcache->get('key02') . "\n";

$memcache->set('key03', 'value03');
print 'key03.value : ' . $memcache->get('key03') . "\n";

// CAS (在下面的示例中，key03的值不会更新为value05)
$memcache->get('key03', null, $cas);
$memcache->replace('key03', 'value04');
if ($memcache->getResultCode() == Memcached::RES_NOTFOUND) {
    $memcache->add('key03', 'value03');
} else {
    $memcache->cas($cas, 'key03', 'value05');
}
print 'key03.value : ' . $memcache->get('key03') . "\n";
?>
```

运行：

```
php use_memcache.php
```

```
key01.value : value01
key01.value : value01,value02
key02.value : 1
key02.value : 101
key02.value : 50
key03.value : value03
key03.value : value04
```

6.4.5. 在Node.js上使用

```
npm install memcache # 安裝Memcached客戶端模塊
```

```
memcache@0.3.0 node_modules/memcache
```

在Node.js上的基本用法：

编辑 use_memcache.js 文件：

```
var memcache = require('memcache');
var client = new memcache.Client();

client.connect();

// 设置并获取Key
client.set('key01', 'value01');
client.get('key01', function (err, val) {
    console.log("key01.value :", val);
});

// 追加并获取Key
client.append('key01', ',value02');
client.get('key01', function (err, val) {
    console.log("key01.value :", val);
});

client.set('key02', 1);
client.get('key02', function (err, val) {
    console.log("key02.value :", val);
});

// 递增
client.increment('key02', 100);
client.get('key02', function (err, val) {
    console.log("key02.value :", val);
});

// 递减
client.decrement('key02', 51);
client.get('key02', function (err, val) {
    console.log("key02.value :", val);
});

// 删除Key
client.delete('key03');
client.get('key03', function (err, val) {
    console.log("key03.value :", val);
});

client.close();
```

6.4. Memcached

运行：

```
node use_memcache.js
```

```
key01.value : value01
key01.value : value01,value02
key02.value : 1
key02.value : 101
key02.value : 50
key03.value : null
```

6.5. Redis

Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库（或叫做NoSQL），并提供多种语言的API。

6.5.1. 安装Redis

```
yum --enablerepo=epel -y install redis # 从EPEL安装
```

配置Redis的基本设置：

编辑 /etc/redis.conf 文件：

```

# 监听端口
port 6379

# 监听接口。默认为仅localhost，如果要从其他主机连接，更改为对应的IP地址或0
# .0.0.0
bind 127.0.0.1

# 数据库数量。数据库ID从0到（设置的值减1）分配
databases 16

# 保存磁盘上的数据库。下面默认设置的意思如下：
# 在900秒后至少有1个key变更
# 在300秒后至少有10个key变更
# 在60秒后至少有10000个key变更
# 如果您要禁用此功能，注释掉所有“save ***”行或指定[save ""]
save 900 1
save 300 10
save 60 10000

# 授权密码
requirepass password

# 替代持久模式（“yes”表示启用）
# 如果启用，Redis失去高性能，但获得更多的安全性
appendonly no

# 如果在磁盘上写入数据时启用“appendonly yes”，只能让操作系统刷新数据
# 原文是“no means do not fsync by Redis (just let the OS flush the
# data)”
#appendfsync always
appendfsync everysec
#appendfsync no

```

```

systemctl start redis
systemctl enable redis

```

firewalld防火墙规则（Redis使用端口6379/TCP）：

```
firewall-cmd --add-port=6379/tcp --permanent  
firewall-cmd --reload
```

6.5.2. 基本用法

这是“redis-cli”客户端程序的基本用法。

以下示例是基本示例，您可以在[官方网站上查看更多命令](#)。

如下连接到**Redis**服务器：

连接到本地服务器，密码是在redis.conf上设置的密码：

```
redis-cli -a password
```

```
# 退出连接  
127.0.0.1:6379> quit
```

可以在连接到服务器之后进行身份验证：

```
redis-cli
```

```
127.0.0.1:6379> auth password  
OK
```

连接到明确指定Database-ID的数据库，如果未指定Database-ID，连接到Database-ID “0”：

```
redis-cli -a password -n 1
```

```
127.0.0.1:6379[1]>  
  
# 更改到Database-ID "2"  
127.0.0.1:6379[1]> select 2  
OK  
127.0.0.1:6379[2]>
```

连接到另一台主机上的Redis，指定“-h [主机名]”

6.5. Redis

```
redis-cli -h node01.srv.world -a password
```

```
10.0.0.51:6379>
```

可以使用redis-cli获得非交互式的结果，例如，获取Key的值：

```
redis-cli -a password get key01
```

```
"value02"
```

控制**Redis**服务器本身的基本用法：

```
redis-cli -a password
```

```
127.0.0.1:6379> info
# Server
redis_version:2.8.19
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:c0359e7aa3798aa2
redis_mode:standalone
os:Linux 3.10.0-327.22.2.el7.x86_64 x86_64
arch_bits:64
.....
.....
# 显示连接的客户端
127.0.0.1:6379> client list
id=3 addr=127.0.0.1:44474 fd=5 name= age=447 idle=0 flags=N db=0
sub=0 psub=0
    multi=-1 qbuf=0 qbuf-free=32768 obl=0 oll=0 omem=0 events=r
cmd=client
id=4 addr=10.0.0.31:43668 fd=6 name= age=10 idle=10 flags=N db=0
sub=0 psub=0
    multi=-1 qbuf=0 qbuf-free=0 obl=0 oll=0 omem=0 events=r cmd=
auth
# 结束客户端的连接
127.0.0.1:6379> client kill 10.0.0.31:43668
OK
```

```
# 转储下面的命令后的所有请求
127.0.0.1:6379> monitor
OK
1469078099.850114 [0 10.0.0.31:43666] "get" "key01"
1469078112.319154 [0 10.0.0.31:43666] "set" "key02" "value02"
.....
.....

# 将数据保存在前台的磁盘上
127.0.0.1:6379> save
OK

# 将数据保存在后台的磁盘上
127.0.0.1:6379> bgsave
Background saving started

# 获取上次保存到磁盘的UNIX时间戳
127.0.0.1:6379> lastsave
(integer) 1469078407

# 将数据保存在磁盘上并关闭Redis
127.0.0.1:6379> shutdown
not connected> quit
```

```
ps aux | grep [r]edis
```

Keys的基本用法：

```
redis-cli -a password
```

```
# 设置Key的值
127.0.0.1:6379> set key01 value01
OK

# 获取Key的值
127.0.0.1:6379> get key01
"value01"

# 删除Key
127.0.0.1:6379> del key01
(integer) 1
```

```
# 确定Key是否存在 (1表示true)
127.0.0.1:6379> exists key01
(integer) 1

# 只有当Key不存在时才设置Key的值，整数0表示未设置值，因为Key已存在
127.0.0.1:6379> setnx key01 value02
(integer) 0

# 设置有有效期的Key的值 (60表示值将在60秒后过期)
127.0.0.1:6379> setex key01 60 value01
OK

# 给已有Key设置过期日期
127.0.0.1:6379> expire key02 30
(integer) 1

# 将值添加到Key
127.0.0.1:6379> append key01 value02
(integer) 15

# 获取Key值的子字符串：[Key] [Start] [End]
127.0.0.1:6379> substr key01 0 3
"valu"

127.0.0.1:6379> set key02 1
OK

# 递增Key的整数值
127.0.0.1:6379> incr key02
(integer) 2

# 递增指定值的Key值的整数值
127.0.0.1:6379> incrby key02 100
(integer) 102

# 递减Key的整数值
127.0.0.1:6379> decr key02
(integer) 101

# 递减指定值的Key值的整数值
127.0.0.1:6379> decrby key02 51
```

```
(integer) 50

# 设置某些Key的值
127.0.0.1:6379> mset key01 value01 key02 value02 key03 value03
OK

# 获取某些Key的值
127.0.0.1:6379> mget key01 key02 key03
1) "value01"
2) "value02"
3) "value03"

# 重命名已有Key
127.0.0.1:6379> rename key01 key02
OK
127.0.0.1:6379> mget key01 key02
1) (nil)
2) "value01"

# 重命名已有Key，但如果重命名的Key已存在，则命令不运行
127.0.0.1:6379> renamenx key02 key03
(integer) 0
127.0.0.1:6379> mget key02 key03
1) "value01"
2) "value03"

# 获取当前数据库上的Key数
127.0.0.1:6379> dbsize
(integer) 4

# 将Key移动到另一个数据库
127.0.0.1:6379> move key03 1
(integer) 1
127.0.0.1:6379> select 1
OK
127.0.0.1:6379[1]> get key03
"value03"

# 删除当前数据库上的所有Key
127.0.0.1:6379> flushdb
OK
```

6.5. Redis

```
# 删除所有数据库上的所有Key  
127.0.0.1:6379> flushall  
OK  
127.0.0.1:6379> quit
```

进程从stdout读取数据：

```
echo 'test_words' | redis-cli -a password -x set key01
```

```
OK
```

```
redis-cli -a password get key01
```

```
"test_words\n"
```

在Redis上，可以使用 `watch` 命令来使用CAS操作。如果另一个进程在multi-exec之间更改了Key的值，则更改不应用于Key：

```
# watch一个Key  
127.0.0.1:6379> watch key01  
OK  
  
127.0.0.1:6379> get key01  
"value01"  
127.0.0.1:6379> multi  
OK  
127.0.0.1:6379> set key01 value02  
QUEUED  
  
127.0.0.1:6379> exec  
1) OK
```

这是列表的基本用法：

```
redis-cli -a password
```

```
# 将值添加到列表（可以使用空格设置多个值）  
127.0.0.1:6379> lpush list01 value01  
(integer) 1
```

```
# 将值附加到列表（可以使用空格设置多个值）
127.0.0.1:6379> rpush list01 value02
(integer) 2

# 获取列表的长度
127.0.0.1:6379> llen list01
(integer) 2

# 获取列表的特定元素
127.0.0.1:6379> lindex list01 0
"value01"

# 获取指定范围的元素
127.0.0.1:6379> lrange list01 0 1
1) "value01"
2) "value02"

# 将特定元素更改为指定值
127.0.0.1:6379> lset list01 1 value03
OK
127.0.0.1:6379> lindex list01 1
"value03"

# 获取头元素并删除它
127.0.0.1:6379> lpop list01
"value01"

# 获取最后一个元素并删除它
127.0.0.1:6379> rpop list01
"value03"

# 缩减列表的指定范围
127.0.0.1:6379> ltrim list01 1 3
OK

127.0.0.1:6379> lrange list02 0 7
1) "value01"
2) "test"
3) "value02"
4) "value03"
5) "value04"
```

```
6) "test"
7) "value05"
8) "test"

# 删除列表的指定数量的元素
127.0.0.1:6379> lrem list02 2 test
(integer) 2
127.0.0.1:6379> lrange list02 0 7
1) "value01"
2) "value02"
3) "value03"
4) "value04"
5) "value05"
6) "test"
```

哈希表的基本用法：

```
redis-cli -a password
```

```
# 设置哈希字段的值
127.0.0.1:6379> hset hash01 field01 value01
(integer) 1

# 获取哈希字段的值
127.0.0.1:6379> hget hash01 field01
"value01"

# 设置一些哈希字段的值
127.0.0.1:6379> hmset hash01 field02 value02 field03 value03
OK

# 获取一些哈希字段的值
127.0.0.1:6379> hmget hash01 field01 field02 field03
1) "value01"
2) "value02"
3) "value03"

# 获取哈希表的所有字段
127.0.0.1:6379> hkeys hash01
1) "field01"
2) "field02"
```

```

3) "field03"

# 获取哈希表的所有字段值
127.0.0.1:6379> hvals hash01
1) "value01"
2) "value02"
3) "value03"

# 获取哈希表的所有值和字段
127.0.0.1:6379> hgetall hash01
1) "field01"
2) "value01"
3) "field02"
4) "value02"
5) "field03"
6) "value03"

# 递增哈希字段值的整数值
127.0.0.1:6379> hincrby hash01 field04 100
(integer) 101

# 确定是否存在哈希字段
127.0.0.1:6379> hexists hash01 field01
(integer) 1

# 获取哈希字段数
127.0.0.1:6379> hlen hash01
(integer) 4

# 删除哈希的特定字段
127.0.0.1:6379> hdel hash01 field04
(integer) 1

```

集合的基本用法：

```
redis-cli -a password
```

```

# 添加成员到一个集合（可以使用空格添加多个成员）
127.0.0.1:6379> sadd set01 member01
(integer) 1

```

```
# 获取集合的成员数
127.0.0.1:6379> scard set01
(integer) 1

# 删除集合的指定成员
127.0.0.1:6379> srem set01 member03
(integer) 1

# 确定指定的成员是否存在
127.0.0.1:6379> sismember set01 member01
(integer) 1

# 获取集合的所有成员
127.0.0.1:6379> smembers set01
1) "member03"
2) "member02"
3) "member01"
127.0.0.1:6379> smembers set02
1) "member02"
2) "member05"
3) "member04"
127.0.0.1:6379> smembers set03
1) "member06"
2) "member02"
3) "member01"

# 获得从所有给定集合的交集得到的集合的成员
127.0.0.1:6379> sinter set01 set02 set03
1) "member02"

# 等于上面的SINTER，但不显示结果，而是存储到目标集合（第一个参数）
127.0.0.1:6379> sinterstore set04 set01 set02 set03
(integer) 1
127.0.0.1:6379> smembers set04
1) "member02"

# 获得从第一集合和所有连续集合之间的差集得到的集合的成员
127.0.0.1:6379> sdiff set01 set02 set03
1) "member03"

# 等于上面的SDIFF，但不显示结果，而是存储到目标集合（第一个参数）
127.0.0.1:6379> sdifftore set05 set01 set02 set03
```

```
(integer) 1

127.0.0.1:6379> smembers set05
1) "member03"

# 获得从所有给定集合的并集得到的集合的成员
127.0.0.1:6379> sunion set01 set02 set03
(integer) 1

# 等于上面的SOUNION，但不显示结果，而是存储到目标集合（第一个参数）
127.0.0.1:6379> sunionstore set06 set01 set02 set03
(integer) 6
127.0.0.1:6379> smembers set06
1) "member06"
2) "member03"
3) "member04"
4) "member02"
5) "member01"
6) "member05"

# 将成员从集合（第一个参数）移动到集合（第二个参数）
127.0.0.1:6379> smove set01 set02 member03
(integer) 1
```

6.5.3. 在Python上使用

这是在Python上使用Redis的示例。

```
yum --enablerepo=epel -y install python-redis # 从EPEL安装Python
Redis客户端库
```

在Python上的基本用法：

编辑 use_redis.py 文件：

```
#!/usr/bin/env python

import redis

client = redis.StrictRedis(host='127.0.0.1', port=6379, db=0, password='password')

# 设置并获取Key
client.set("key01", "value01")
print "key01.value :", client.get("key01")

# 追加并获取Key
client.append("key01", ",value02")
print "key01.value :", client.get("key01")

client.set("key02", 1)

# 递增
client.incr("key02", 100)
print "key02.value :", client.get("key02")

# 递减
client.decr("key02", 51)
print "key02.value :", client.get("key02")

# 列表
client.lpush("list01", "value01", "value02", "value03")
print "list01.value :", client.lrange("list01", "0", "2")

# 哈希
client.hmset("hash01", {"key01": "value01", "key02": "value02", "key03": "value03"})
print "hash01.value :", client.hmget("hash01", ["key01", "key02", "key03"])

# 集合
client.sadd("set01", "member01", "member02", "member03")
print "set01.value :", client.smembers("set01")
```

运行：

```
python use_redis.py
```

```
key01.value : value01
key01.value : value01,value02
key02.value : 101
key02.value : 50
list01.value : ['value03', 'value02', 'value01']
hash01.value : ['value01', 'value02', 'value03']
set01.value : set(['member02', 'member03', 'member01'])
```

6.5.4. 在PHP上使用

这是在PHP上使用Redis的示例。

```
yum --enablerepo=epel -y install php-pecl-redis #从EPEL安装PHP
Redis客户端模块
```

在PHP上的基本用法：

编辑 use_redis.php 文件：

```
<?php
$redis = new Redis();
$redis->connect("127.0.0.1", 6379);
$redis->auth("password");

// 设置并获取Key
$redis->set('key01', 'value01');
print 'key01.value : ' . $redis->get('key01') . "\n";

// 追加并获取Key
$redis->append('key01', ',value02');
print 'key01.value : ' . $redis->get('key01') . "\n";

$redis->set('key02', 1);
print 'key02.value : ' . $redis->get('key02') . "\n";

// 递增
$redis->incr('key02', 100);
print 'key02.value : ' . $redis->get('key02') . "\n";
```

```
// 递减
$redis->decr('key02', 51);
print 'key02.value : ' . $redis->get('key02') . "\n";

// 列表
$redis->lPush('list01', 'value01');
$redis->rPush('list01', 'value02');
print 'list01.value : ';
print_r ($redis->lRange('list01', 0, -1));

// 哈希
$redis->hSet('hash01', 'key01', 'value01');
$redis->hSet('hash01', 'key02', 'value02');
print 'hash01.value : ';
print_r ($redis->hGetAll('hash01'));

// 集合
$redis->sAdd('set01', 'member01');
$redis->sAdd('set01', 'member02');
print 'set01.value : ';
print_r ($redis->sMembers('set01'));
?>
```

运行：

```
php use_redis.php
```

```

key01.value : value01
key01.value : value01,value02
key02.value : 1
key02.value : 101
key02.value : 50
list01.value : Array
(
    [0] => value01
    [1] => value02
)
hash01.value : Array
(
    [key01] => value01
    [key02] => value02
)
set01.value : Array
(
    [0] => member01
    [1] => member02
)

```

6.5.5. 在Node.js上使用

这是在Node.js上使用Redis的示例。

```
npm install redis # 安装Redis客户端模块
```

在Node.js上的基本用法：

编辑 use_redis.js 文件：

```

var redis = require('redis');
var client = new redis.createClient();

client.auth('password');

// 设置并获取Key
client.set('key01', 'value01');
client.get('key01', function (err, val) {
    console.log("key01.value :", val);
}

```

```
});  
  
// 追加并获取Key  
client.append('key01', 'value02');  
client.get('key01', function (err, val) {  
    console.log("key01.value :", val);  
});  
  
client.set('key02', 1);  
client.get('key02', function (err, val) {  
    console.log("key02.value :", val);  
});  
  
// 递增  
client.incrby('key02', 100);  
client.get('key02', function (err, val) {  
    console.log("key02.value :", val);  
});  
  
// 递减  
client.decrby('key02', 51);  
client.get('key02', function (err, val) {  
    console.log("key02.value :", val);  
});  
  
// 列表  
client.rpush('list01', 'value01');  
client.rpush('list01', 'value02');  
client.lrange('list01', 0, -1, function (err, val) {  
    console.log("list01.value :", val);  
});  
  
// 哈希  
client.hset("hash01", "key01", "value01");  
client.hset("hash01", "key02", "value02");  
client.hgetall('hash01', function (err, val) {  
    console.log("hash01.value :", val);  
});  
  
// 集合  
client.sadd("set01", "member01");  
client.sadd("set01", "member02");
```

```
client.smembers('set01', function (err, val) {  
    console.log("set01.value :", val);  
});
```

运行：

```
node use_redis.js
```

```
key01.value : value01  
key01.value : value01,value02  
key02.value : 1  
key02.value : 101  
key02.value : 50  
list01.value : [ 'value01', 'value02' ]  
hash01.value : { key01: 'value01', key02: 'value02' }  
set01.value : [ 'member01', 'member02' ]
```

6.5.6. Redis 复制

此配置是通用的[主从设置](#)。

更改主服务器上的设置：

```
编辑 /etc/redis.conf 文件：
```

```
# 更改自己的IP或0.0.0.0  
bind 0.0.0.0  
  
# 如果需要，可添加以下内容  
# min-slaves-to-write：多少数量的从服务器在线，主服务器接受写请求  
# min-slaves-max-lag：从服务器在多少时间（秒）内返回应答，则判定为在线  
min-slaves-to-write 2  
min-slaves-max-lag 10
```

```
systemctl restart redis
```

更改从服务器上的设置：

```
编辑 /etc/redis.conf 文件：
```

```
# 更改自己的IP或0.0.0.0
bind 0.0.0.0

# 添加主服务器IP和端口
slaveof 10.0.0.30 6379

# 添加在主服务器上设置的连接密码
masterauth password

# 验证参数（将从服务器设置为只读）
slave-read-only yes
```

```
systemctl restart redis
```

在主、从服务器上添加firewalld防火墙规则：

```
firewall-cmd --add-port=6379/tcp --permanent
firewall-cmd --reload
```

如果SELinux在从服务器上启用，如下所示添加复制规则：

编辑 redis_repl.te 文件：

```
module redis_repl 1.0;

require {
    type redis_port_t;
    type redis_t;
    class tcp_socket name_connect;
}

===== redis_t =====
allow redis_t redis_port_t:tcp_socket name_connect;
```

```
checkmodule -m -M -o redis_repl.mod redis_repl.te
```

```
checkmodule: loading policy configuration from redis_repl.te
checkmodule: policy configuration loaded
checkmodule: writing binary representation (version 17) to redis
_repl.mod
```

```
semodule_package --outfile redis_repl.pp --module redis_repl.mod
semodule -i redis_repl.pp
```

验证从服务器的状态，如果显示“master_link_status:up”，表示正常：

```
redis-cli info Replication
```

```
# Replication
role:slave
master_host:10.0.0.30
master_port:6379
master_link_status:up
master_last_io_seconds_ago:1
master_sync_in_progress:0
slave_repl_offset:384
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
```

尝试正常获取Key：

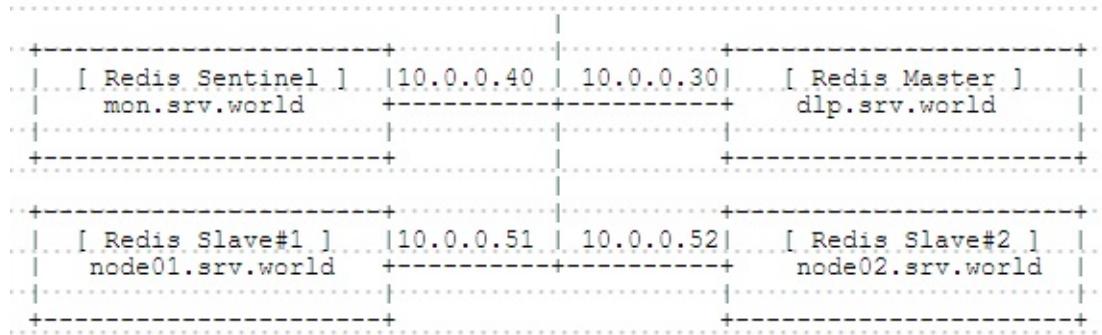
```
redis-cli get key_on_master
```

```
"value_on_master"
```

6.5.7. Redis Sentinel

配置[Redis Sentinel](#)为Redis服务器提供高可用性。

本例基于如下环境：



先参照[上一节](#)内容，在所有Redis主节点和从节点上配置复制设置。

需要注意的有关复制设置的要点：它需要在所有节点上设置相同的认证密码。

此外，如果主节点上启用了SELinux，则需要在主节点上添加与从节点相同的规则，因为主节点在其将关闭时将成为从节点。

对于使用Sentinel的Redis HA，如果在主、从节点上启用了SELinux，则需要添加更多的规则（在所有主、从节点上添加），如下所示：

编辑 `redis_ha.te` 文件：

```

module redis_ha 1.0;

require {
    type etc_t;
    type redis_t;
    class file write;
}

===== redis_t =====
allow redis_t etc_t:file write;

```

```
checkmodule -m -M -o redis_ha.mod redis_ha.te
```

```

checkmodule: loading policy configuration from redis_ha.te
checkmodule: policy configuration loaded
checkmodule: writing binary representation (version 17) to redis
_ha.mod

```

```
semodule_package --outfile redis_ha.pp --module redis_ha.mod
```

6.5. Redis

```
semodule -i redis_ha.pp
```

配置Sentinel服务器：

```
yum --enablerepo=epel -y install redis
```

编辑 `/etc/redis-sentinel.conf` 文件：

```
# 更改[sentinel monitor (任意名称) (主服务器IP) (主服务器端口) (Quorum)]
# Quorum：当指定数量的Sentinel服务器监视到主服务器运行失败时运行故障转移
sentinel monitor mymaster 10.0.0.30 6379 1

# 主认证密码
sentinel auth-pass mymaster password

# Sentinel服务器判断主服务已关闭的条件（默认为30秒）the term Sentinel server looks Master is down (30 sec by default below)
sentinel down-after-milliseconds mymaster 30000

# 运行故障转移时要更改的从服务器数
sentinel parallel-syncs mymaster 1
```

```
systemctl start redis-sentinel
systemctl enable redis-sentinel
```

如果在Sentinel服务器上启用了SELinux，添加如下规则：

```
semanage port -a -t redis_port_t -p tcp 26379
```

编辑 `redis_sentinel.te` 文件：

```
module redis_sentinel 1.0;

require {
    type redis_port_t;
    type etc_t;
    type redis_t;
    class tcp_socket name_connect;
    class file write;
}

===== redis_t =====
allow redis_t redis_port_t:tcp_socket name_connect;
allow redis_t etc_t:file write;
```

```
checkmodule -m -M -o redis_sentinel.mod redis_sentinel.te
```

```
checkmodule: loading policy configuration from redis_sentinel.te
checkmodule: policy configuration loaded
checkmodule: writing binary representation (version 17) to redis
_sentinel.mod
```

```
semodule_package --outfile redis_sentinel.pp --module
redis_sentinel.mod
```

```
semodule -i redis_sentinel.pp
```

如下所示，验证**Sentinel**服务器上的状态，此外，在主节点上停止**Redis**，并确保主、从故障转移正常：

```
redis-cli -p 26379
```

```
# 显示“mymaster”的主节点
127.0.0.1:26379> sentinel get-master-addr-by-name mymaster
1) "10.0.0.30"
2) "6379"

# 显示“mymaster”的主节点的详细信息
127.0.0.1:26379> sentinel master mymaster
1) "name"
2) "mymaster"
3) "ip"
4) "10.0.0.30"
5) "port"
6) "6379"
.....
.....

# 显示“mymaster”的从节点
127.0.0.1:26379> sentinel slaves mymaster
1) 1) "name"
2) "10.0.0.52:6379"
3) "ip"
4) "10.0.0.52"
5) "port"
6) "6379"
.....
.....

2) 1) "name"
2) "10.0.0.51:6379"
3) "ip"
4) "10.0.0.51"
5) "port"
6) "6379"
.....
.....
```

6.5.8. Redis Benchmark

可以使用Redis包中包含的工具运行[基准测试](#)。

使用redis-benchmark工具（有一些选项如指定请求数等等，参阅 `redis-benchmark --help`），如下所示：

```
redis-benchmark -h 10.0.0.30 -p 6379
```

```
===== PING_INLINE =====
100000 requests completed in 1.26 seconds
50 parallel clients
3 bytes payload
keep alive: 1

98.30% <= 1 milliseconds
99.90% <= 2 milliseconds
99.94% <= 3 milliseconds
99.95% <= 4 milliseconds
100.00% <= 4 milliseconds
79491.26 requests per second

===== PING_BULK =====
100000 requests completed in 1.28 seconds
50 parallel clients
3 bytes payload
keep alive: 1

99.99% <= 1 milliseconds
100.00% <= 1 milliseconds
77942.32 requests per second

===== SET =====
100000 requests completed in 1.29 seconds
50 parallel clients
3 bytes payload
keep alive: 1

100.00% <= 0 milliseconds
77579.52 requests per second

===== GET =====
100000 requests completed in 1.28 seconds
50 parallel clients
3 bytes payload
keep alive: 1
```

```
100.00% <= 0 milliseconds
78186.08 requests per second

===== INCR =====
100000 requests completed in 1.29 seconds
50 parallel clients
3 bytes payload
keep alive: 1

100.00% <= 1 milliseconds
77519.38 requests per second

===== LPUSH =====
100000 requests completed in 1.27 seconds
50 parallel clients
3 bytes payload
keep alive: 1

100.00% <= 0 milliseconds
78678.20 requests per second

===== LPOP =====
100000 requests completed in 1.27 seconds
50 parallel clients
3 bytes payload
keep alive: 1

100.00% <= 0 milliseconds
78492.93 requests per second

===== SADD =====
100000 requests completed in 1.28 seconds
50 parallel clients
3 bytes payload
keep alive: 1

100.00% <= 0 milliseconds
78064.01 requests per second

===== SPOP =====
100000 requests completed in 1.28 seconds
```

```
50 parallel clients
3 bytes payload
keep alive: 1

100.00% <= 0 milliseconds
78003.12 requests per second

===== LPUSH (needed to benchmark LRANGE) =====
100000 requests completed in 1.32 seconds
50 parallel clients
3 bytes payload
keep alive: 1

99.67% <= 1 milliseconds
99.97% <= 2 milliseconds
100.00% <= 2 milliseconds
75987.84 requests per second

===== LRANGE_100 (first 100 elements) =====
100000 requests completed in 1.32 seconds
50 parallel clients
3 bytes payload
keep alive: 1

100.00% <= 1 milliseconds
75930.14 requests per second

===== LRANGE_300 (first 300 elements) =====
100000 requests completed in 1.32 seconds
50 parallel clients
3 bytes payload
keep alive: 1

99.96% <= 1 milliseconds
100.00% <= 1 milliseconds
75987.84 requests per second

===== LRANGE_500 (first 450 elements) =====
100000 requests completed in 1.26 seconds
50 parallel clients
3 bytes payload
keep alive: 1
```

```
99.97% <= 1 milliseconds
100.00% <= 1 milliseconds
79113.92 requests per second

===== LRANGE_600 (first 600 elements) =====
100000 requests completed in 1.22 seconds
50 parallel clients
3 bytes payload
keep alive: 1

99.97% <= 2 milliseconds
100.00% <= 2 milliseconds
81900.09 requests per second

===== MSET (10 keys) =====
100000 requests completed in 1.23 seconds
50 parallel clients
3 bytes payload
keep alive: 1

99.80% <= 1 milliseconds
99.95% <= 2 milliseconds
100.00% <= 2 milliseconds
81433.22 requests per second
```

6.6. MS SQL Server

点击进入[官方教程](#)

提示：

本教程需要用户输入和internet连接。如果你希望了解无人参与或脱机安装过程，请参阅[在Linux上的SQL Server安装指南](#)。

6.6.1. 安装SQL Server

先决条件

必须是Red Hat Enterprise Linux (RHEL) 7.3 +，且至少2GB的内存。其它系统要求，请参阅[在Linux上SQL Server的系统需求](#)。

安装SQL Server

若要在RHEL上配置SQL Server，在终端运行以下命令安装 `mssql-server` 包：

重要

如果已经安装了CTP或RC版本的SQL Server 2017，必须在注册GA存储库之前先删除旧的存储库。有关详细信息，请参阅[从预览存储库更改到GA存储库](#)。

1、下载Microsoft SQL Server Red Hat存储库配置文件：

```
sudo curl -o /etc/yum.repos.d/mssql-server.repo https://packages.microsoft.com/config/rhel/7/mssql-server-2017.repo
```

备注

这是累积更新(CU)存储库。有关存储库选项和它们之间的差异的详细信息，请参阅[更改源存储库](#)。

2、运行以下命令，安装SQL Server：

```
sudo yum install -y mssql-server
```

3、软件包安装完成后，运行 `mssql-conf` 设置命令并按照操作提示设置SA密码，并选择你的版本。

```
sudo /opt/mssql/bin/mssql-conf setup
```

提示

如果你尝试本教程中的SQL Server 2017，以下版本可自由授予许可：评估（Evaluation）、开发人员（Developer）和快速（Express）。

确保为SA帐户指定强密码（最少8个字符，包括大写和小写字母、十进制数字和/或非字母数字符号）。

4、配置完成后，验证服务是否运行：

```
systemctl status mssql-server
```

5、若要允许远程连接，在RHEL上打开防火墙的SQL Server端口（默认为1433/TCP）。如果使用的是firewalld防火墙，运行以下命令：

```
sudo firewall-cmd --zone=public --add-port=1433/tcp --permanent  
sudo firewall-cmd --reload
```

此时，SQL Server在RHEL计算机上运行并且已准备好使用。

6.6.2. 安装SQL Server命令行工具

要创建数据库，需要连接一个可以在SQL Server上运行Transact-SQL语句的工具。以下步骤安装SQL Server命令行工具[sqlcmd](#)和[bcp](#)。

1、下载Microsoft SQL Server Red Hat存储库配置文件：

```
sudo curl -o /etc/yum.repos.d/msprod.repo https://packages.microsoft.com/config/rhel/7/prod.repo
```

2、如果安装了以前版本的mssql-tools，请删除所有较旧的unixODBC软件包：

```
sudo yum remove unixODBC-utf16 unixODBC-utf16-devel
```

3、运行以下命令以使用unixODBC开发包安装mssql-tools：

```
sudo yum install -y mssql-tools unixODBC-devel
```

4、为方便起见，添加 /opt/mssql-tools/bin/ 到PATH环境变量。这样可以在不指定完整路径的情况下运行这些工具。运行以下命令来修改登录会话和交互/非登录会话的PATH：

```
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile  
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc  
source ~/.bashrc
```

提示

Sqlcmd只是连接到SQL Server运行查询和执行管理和开发任务的一个工具。
其他工具包括：

- [SQL Server Operations Studio \(Preview\)](#)
- [SQL Server Management Studio](#)
- [Visual Studio Code.](#)
- [mssql-cli \(Preview\)](#)

CentOS 6安装

这里的评论中找到的：

```

curl https://packages.microsoft.com/config/rhel/6/prod.repo > /etc/yum.repos.d/mssql-release.repo
sudo yum -y remove unixODBC-utf16 unixODBC-utf16-devel
sudo ACCEPT_EULA=Y yum -y install msodbcsql-13.1.4.0-1 mssql-tools-14.0.3.0-1 unixODBC-devel --disableplugin=priorities --nogpgcheck
echo -e '\nexport PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo -e '\nexport PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc

```

6.6.3. 本地连接

以下步骤使用sqlcmd在本地连接到新的SQL Server实例。

1、使用SQL Server服务器主机（-S），用户名（-U）和密码（-P）的参数运行sqlcmd。在本例中，在本地连接，所以服务器名称是localhost，用户名是SA，密码是在安装期间为SA帐户提供的密码：

```
sqlcmd -S localhost -U SA -P '<YourPassword>'
```

提示

可以在命令行中省略密码以提示输入密码

如果以后决定远程连接，为-S参数指定主机名或IP地址，并确保防火墙上的端口1433/tcp已打开。

2、如果成功，应该出现一个sqlcmd命令提示符：

```
1>
```

3、如果连接失败，首先尝试从错误消息中诊断问题。然后查看[连接故障排除建议](#)。

4、结束sqlcmd会话，运行：

QUIT

6.6.4. 升级

官方教程

```
sudo yum update mssql-server
```

这些命令下载最新的软件包并替换位于 `/opt/mssql/` 下的二进制文件。用户生成的数据库和系统数据库不受此操作的影响。

6.6.5. 卸载

官方教程

```
sudo yum remove mssql-server
```

删除软件包不会删除生成的数据库文件。如果要删除数据库文件，请使用以下命令：

```
sudo rm -rf /var/opt/mssql/
```

7. 目录服务

- 7.1. FreeIPA
 - 7.1.1. 配置IPA服务器
 - 7.1.2. 添加用户帐户
 - 7.1.3. 配置IPA客户端
 - 7.1.4. 用户管理基本操作
 - 7.1.5. Web管理控制台
 - 7.1.6. FreeIPA复制
- 7.2. OpenLDAP
 - 7.2.1. 配置LDAP服务器
 - 7.2.2. 添加用户帐户
 - 7.2.3. 配置LDAP客户端
 - 7.2.4. 配置TLS
 - 7.2.5. OpenLDAP复制
 - 7.2.6. OpenLDAP多主复制
 - 7.2.7. 安装phpLDAPadmin
- 7.3. NIS
 - 7.3.1. 配置NIS服务器
 - 7.3.2. 配置NIS客户端
 - 7.3.3. 配置NIS从服务器

7.1. FreeIPA

7.1.1. 配置IPA服务器

配置IPA服务器以在本地网络中共享用户的帐户。

安装FreeIPA：

```
yum -y install ipa-server ipa-server-dns bind bind-dyndb-ldap
```

设置FreeIPA服务器：

编辑 /etc/hosts 文件：

```
# 添加本机IP
10.0.0.30 dlp.srv.world dlp
```

```
ipa-server-install --setup-dns
```

```
The log file for this installation can be found in /var/log/ipa
server-install.log
=====
=====
```

```
This program will set up the IPA Server.
```

This includes:

- * Configure a stand-alone CA (dogtag) for certificate management
- * Configure the Network Time Daemon (ntpd)
- * Create and configure an instance of Directory Server
- * Create and configure a Kerberos Key Distribution Center (KDC)
- * Configure Apache (httpd)

To accept the default shown in brackets, press the Enter key.

```
# 设置DNS (现有BIND设置被覆盖)
```

```
Existing BIND configuration detected, overwrite? [no]: yes
Enter the fully qualified domain name of the computer
```

7.1. FreeIPA

```
on which you're setting up server software. Using the form  
<hostname>.<domainname>
```

```
Example: master.example.com.
```

```
# 确认主机名并回车
```

```
Server host name [dlp.srv.world]:
```

```
The domain name has been determined based on the host name.
```

```
# 确认域名并回车
```

```
Please confirm the domain name [srv.world]:
```

```
The kerberos protocol requires a Realm name to be defined.
```

```
This is typically the domain name converted to uppercase.
```

```
# 确认领域名称并回车
```

```
Please provide a realm name [SRV.WORLD]:
```

```
Certain directory server operations require an administrative user.
```

```
This user is referred to as the Directory Manager and has full access
```

```
to the Directory for system management tasks and will be added to the
```

```
instance of directory server created for IPA.
```

```
The password must be at least 8 characters long.
```

```
# 目录管理器的密码
```

```
Directory Manager password:
```

```
Password (confirm):
```

```
The IPA server requires an administrative user, named 'admin'.
```

```
This user is a regular system account used for IPA server administration.
```

```
# IPA管理员密码
```

```
IPA admin password:
```

```
Password (confirm):
```

```
# 设置DNS转发器，回答yes或no
```

```
Do you want to configure DNS forwarders? [yes]:
```

```
Enter the IP address of DNS forwarder to use, or press Enter to finish.
```

7.1. FreeIPA

```
# 如果设置DNS转发器，指定DNS转发器的IP
Enter IP address for a DNS forwarder: 10.0.0.10
DNS forwarder 10.0.0.10 added

# 如果DNS转发器正常，则不用输入
Enter IP address for a DNS forwarder:

# 设置反向区域，回答yes或no
Do you want to configure the reverse zone? [yes]:
# 反向区域名称（如果设置了反向区域）
Please specify the reverse zone name [0.0.10.in-addr.arpa.]:


The IPA Master Server will be configured with:
Hostname:      dlp.srv.world
IP address:    10.0.0.30
Domain name:   srv.world
Realm name:    SRV.WORLD


BIND DNS server will be configured to serve IPA domain with:
Forwarders:    10.0.0.10
Reverse zone:  0.0.10.in-addr.arpa.


# 确认设置并输入“yes”继续
Continue to configure the system with these values? [no]: yes

The following operations may take some minutes to complete.
Please wait until the prompt is returned.

Configuring NTP daemon (ntpd)

...
...
...
=====
=====

Setup complete

Next steps:
1. You must make sure these network ports are open:
   TCP Ports:
   * 80, 443: HTTP/HTTPS
   * 389, 636: LDAP/LDAPS
   * 88, 464: kerberos
   UDP Ports:
   * 88, 464: kerberos
```

7.1. FreeIPA

```
* 123: ntp

2. You can now obtain a kerberos ticket using the command: 'kinit admin'
   This ticket will allow you to use the IPA tools (e.g., ipa user-add)
   and the web user interface.
```

Be sure to back up the CA certificate stored in /root/cacert.p12
This file is required to create replicas. The password for this
file is the Directory Manager password

获取Kerberos票证并更改默认shell：

```
kinit admin
```

```
Password for admin@SRV.WORLD: # IPA admin密码
```

```
klist # 确认
```

```
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@SRV.WORLD

Valid starting     Expires            Service principal
03/21/2015 14:25:53  03/24/2015 14:25:50  krbtgt/SRV.WORLD@SRV.WORLD
```

```
ipa config-mod --defaultshell=/bin/bash
```

```
Maximum username length: 32
Home directory base: /home
Default shell: /bin/bash
Default users group: ipausers
Default e-mail domain: srv.world
Search time limit: 2
Search size limit: 100
User search fields: uid,givenname,sn,telephonenumber,ou,title
Group search fields: cn,description
Enable migration mode: FALSE
Certificate Subject base: O=SRV.WORLD
Password Expiration Notification (days): 4
Password plugin features: AllowNTHash
SELinux user map order: guest_u:s0$guest_u:s0$user_u:s0$staff_u
:s0-s0:c0.c1023$unconfined_u:s0-s0:c0.c1023
Default SELinux user: unconfined_u:s0-s0:c0.c1023
Default PAC types: MS-PAC
```

firewalld防火墙规则：

```
firewall-cmd --add-service={ssh,dns,freeipa-ldap,freeipa-ldaps}
--permanent
firewall-cmd --reload
```

7.1.2. 添加用户帐户

在FreeIPA服务器上添加用户帐户。

添加用户（在此设置的密码需要在初始登录时更改）：

```
ipa user-add cent --first=CentOS --last=Linux --password
```

7.1. FreeIPA

```
Password: # 设置密码
Enter Password again to verify:
-----
Added user "cent"
-----
User login: cent
First name: CentOS
Last name: Linux
Full name: CentOS Linux
Display name: CentOS Linux
Initials: CL
Home directory: /home/cent
GECOS field: CentOS Linux
Login shell: /bin/bash
Kerberos principal: cent@SRV.WORLD
Email address: cent@srv.world
UID: 1219600001
GID: 1219600001
Password: True
Kerberos keys available: True
```

```
ipa user-find cent # 确认
```

```
-----
1 user matched
-----
User login: cent
First name: CentOS
Last name: Linux
Home directory: /home/cent
Login shell: /bin/bash
Email address: cent@srv.world
UID: 1219600001
GID: 1219600001
Account disabled: False
Password: True
Kerberos keys available: True
-----
Number of entries returned 1
-----
```

7.1. FreeIPA

将现有本地用户添加到IPA目录（本例的用户名设置相同的密码，但是在初始登录时需要更改）：

编辑 ipauser.sh 文件：

```
# 提取UID为1000-9999的本地用户，下面为示例
#!/bin/bash

for line in `grep "x:[1-9][0-9][0-9][0-9]:" /etc/passwd`
do
    USER=`echo $line | cut -d: -f1`
    FIRST=`echo $line | cut -d: -f5 | awk {'print $1'}``LAST=`echo $line | cut -d: -f5 | awk {'print $2'}``[ ! "$FIRST" ] && FIRST=$USER
    [ ! "$LAST" ] && LAST=$USER

    echo $USER | ipa user-add $USER --first=$FIRST --last=$LAST -
    -password
done
```

```
sh ipauser.sh
```

```
-----  
Added user "redhat"  
-----  
User login: redhat  
First name: redhat  
Last name: redhat  
Full name: redhat redhat  
Display name: redhat redhat  
Initials: rr  
Home directory: /home/redhat  
GECOS field: redhat redhat  
Login shell: /bin/bash  
Kerberos principal: redhat@SRV.WORLD  
Email address: redhat@srv.world  
UID: 1219600003  
GID: 1219600003  
Password: True  
Kerberos keys available: True  
-----  
Added user "ubuntu"  
-----  
User login: ubuntu  
First name: ubuntu  
Last name: ubuntu  
Full name: ubuntu ubuntu  
Display name: ubuntu ubuntu  
Initials: uu  
Home directory: /home/ubuntu  
GECOS field: ubuntu ubuntu  
Login shell: /bin/bash  
Kerberos principal: ubuntu@SRV.WORLD  
Email address: ubuntu@srv.world  
UID: 1219600004  
GID: 1219600004  
Password: True  
Kerberos keys available: True
```

7.1.3. 配置IPA客户端

配置FreeIPA客户端连接到FreeIPA服务器。

7.1. FreeIPA

首先在FreeIPA服务器上为FreeIPA客户端添加DNS条目。

```
ipa dnsrecord-add srv.world client01 --a-rec 10.0.0.51 # 格式为 : ipa  
dnsrecord-add [domain name] [record name] [record type] [record]
```

```
Record name: client01  
A record: 10.0.0.51
```

在FreeIPA客户端主机上安装客户端工具并更改DNS设置：

```
yum -y install ipa-client  
  
nmcli c modify eno16777736 ipv4.dns 10.0.0.30  
  
nmcli c down eno16777736; nmcli c up eno16777736
```

在设置为FreeIPA客户端之前与FreeIPA服务器同步时间。

```
ipa-client-install
```

7.1. FreeIPA

```
Discovery was successful!
Hostname: client01.srv.world
Realm: SRV.WORLD
DNS Domain: srv.world
IPA Server: dlp.srv.world
BaseDN: dc=srv,dc=world
```

```
Continue to configure the system with these values? [no]: yes #  
确认设置并点击“yes”继续：
```

```
User authorized to enroll computers: admin # 输入“admin”
```

```
Synchronizing time with KDC...
```

```
Unable to sync time with IPA NTP server, assuming the time is in  
sync. Please check that 123 UDP port is opened.
```

```
Password for admin@SRV.WORLD:
```

```
Successfully retrieved CA cert
```

```
Subject: CN=Certificate Authority,O=SRV.WORLD
Issuer: CN=Certificate Authority,O=SRV.WORLD
Valid From: Fri Mar 20 01:42:15 2015 UTC
Valid Until: Tue Mar 20 01:42:15 2035 UTC
```

```
Enrolled in IPA realm SRV.WORLD
```

```
.....
```

```
.....
```

```
Configured /etc/ssh/ssh_config
```

```
Configured /etc/ssh/sshd_config
```

```
Client configuration complete.
```

如果需要，配置mkhomedir（用户的homedirs在初始登录时创建）

```
authconfig --enablemkhomedir --update
```

```
getsebool: SELinux is disabled
```

```
exit
```

```
logout

CentOS Linux 7 (Core)
Kernel 3.10.0-123.20.1.el7.x86_64 on an x86_64

client01 login: redhat # IPA用户
Password: # 密码
Password expired. Change your password now. # 需要在初始登录时更改
密码
Current Password: # 当前密码
New password: # 新密码
Retype new password: # 确认新密码
Creating home directory for redhat.
[redhat@client01 ~]$ # 登录成功
```

7.1.4. 用户管理基本操作

添加一个FreeIPA用户帐户：

```
ipa user-add cent --first=CentOS --last=Linux --password
```

7.1. FreeIPA

```
Password:  
Enter Password again to verify:  
-----  
Added user "cent"  
-----  
User login: cent  
First name: CentOS  
Last name: Linux  
Full name: CentOS Linux  
Display name: CentOS Linux  
Initials: CL  
Home directory: /home/cent  
GECOS field: CentOS Linux  
Login shell: /bin/bash  
Kerberos principal: cent@SRV.WORLD  
Email address: cent@srv.world  
UID: 1781800001  
GID: 1781800001  
Password: True  
Kerberos keys available: True
```

锁定或解锁FreeIPA用户：

```
ipa user-disable cent
```

```
-----  
Disabled user account "cent"  
-----
```

```
ipa user-enable cent
```

```
-----  
Enabled user account "cent"  
-----
```

搜索FreeIPA用户：

```
ipa user-find cent
```

7.1. FreeIPA

```
-----  
1 user matched  
-----  
User login: cent  
First name: CentOS  
Last name: Linux  
Home directory: /home/cent  
Login shell: /bin/bash  
Email address: cent@srv.world  
UID: 1781800001  
GID: 1781800001  
Account disabled: False  
Password: True  
Kerberos keys available: True  
-----  
Number of entries returned 1  
-----
```

```
ipa user-show --raw cent
```

```
uid: cent  
givenname: CentOS  
sn: Linux  
homedirectory: /home/cent  
loginshell: /bin/bash  
mail: cent@srv.world  
uidnumber: 1781800001  
gidnumber: 1781800001  
nsaccountlock: False  
has_password: True  
has_keytab: True
```

删除FreeIPA用户：

```
ipa user-del cent
```

7.1. FreeIPA

```
-----  
Added group "development"  
-----  
Group name: development  
Description: Development Group  
GID: 1781800006
```

在FreeIPA组中添加成员：

```
ipa group-add-member --users=redhat,ubuntu development
```

```
-----  
Group name: development  
Description: Development Group  
GID: 1781800006  
Member users: redhat, ubuntu  
-----  
Number of members added 2  
-----
```

在FreeIPA组中添加组：

```
ipa group-add-member --groups=development hiroshima
```

```
-----  
Group name: hiroshima  
Description: State Group  
GID: 1781800007  
Member groups: development  
-----  
Number of members added 1  
-----
```

搜索FreeIPA组：

```
ipa group-find development
```

```
-----  
1 group matched  
-----  
Group name: development  
Description: Development Group  
GID: 1781800006  
Member users: redhat, ubuntu  
Member of groups: hiroshima  
-----  
Number of entries returned 1  
-----
```

删除一个FreeIPA组：

```
ipa group-del hiroshima
```

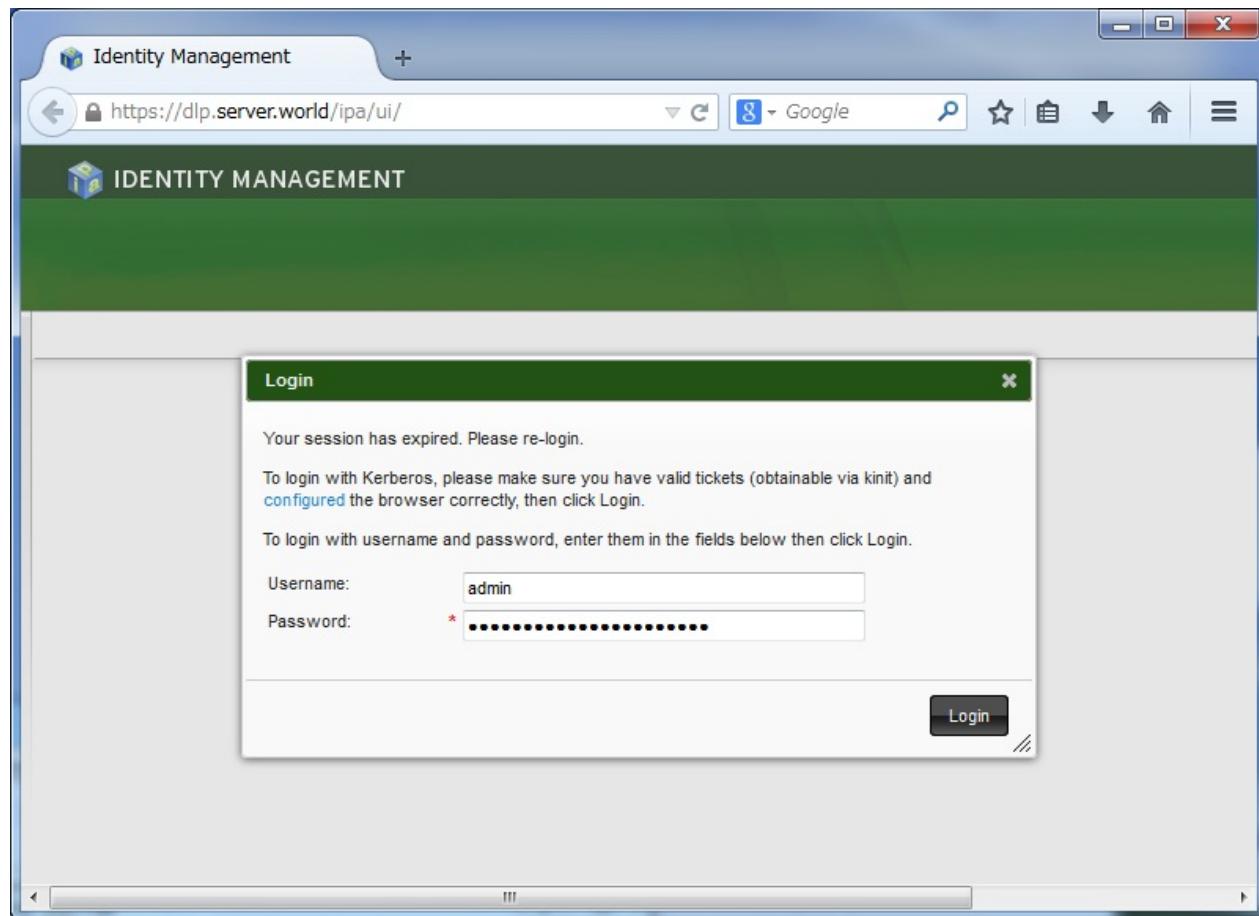
```
-----  
Deleted group "hiroshima"  
-----
```

7.1.5. Web管理控制台

可以在Web管理控制台上操作FreeIPA服务器。

在FreeIPA服务器的同一网络上的任何客户端上启动Web浏览器并访问 [`https://\(FreeIPA服务器主机名或IP\)`](https://(FreeIPA服务器主机名或IP))，使用FreeIPA用户登录到IPA服务器（本例使用管理员用户）：

7.1. FreeIPA



正常登录。可以轻松地在图形界面上操作FreeIPA服务器：

A screenshot of the FreeIPA Identity Management interface. The top navigation bar shows "Logged". Below it, there are tabs for "Identity", "Policy", and "IPA Server", with "Identity" being the active tab. Under "Identity", there are sub-tabs: "Users" (which is selected), "User Groups", "Hosts", "Host Groups", "Netgroups", "Services", "証明書" (Certificates), and "Realm Domains". The main content area is titled "USERS". At the top of the users list are buttons for "更新" (Update), "Delete", "+ Add", "Disable", and "Enable". Below these buttons is a table with columns: "User login", "First name", "Last name", "Status", "UID", "Email address", and "Telephone Number". The table lists six users: admin, cent, debian, fedora, redhat, and ubuntu. Each user row includes a checkbox in the first column. The "Status" column shows "Enabled" with a checked checkbox for all users. The "Email address" and "Telephone Number" columns show email addresses like "cent@server.world" and "debian@server.world" respectively. The "Telephone Number" column shows "Redhat" for redhat and "Ubuntu" for ubuntu. At the bottom of the user list, it says "Showing 1 to 6 of 6 entries."

7.1.6. FreeIPA复制

在副本主机上安装FreeIPA服务器的工具，并更改DNS设置：

```
yum -y install ipa-server ipa-server-dns bind bind-dyndb-ldap
```

将DNS更改为FreeIPA服务器：

```
nmcli c modify eno16777736 ipv4.dns 10.0.0.30
```

```
nmcli c down eno16777736; nmcli c up eno16777736
```

在FreeIPA服务器上为副本主机添加DNS条目：

```
ipa dnsrecord-add srv.world repl01 --a-rec 10.0.0.61
```

```
Record name: repl01
A record: 10.0.0.61
```

```
ipa-replica-prepare repl01.srv.world --ip-address 10.0.0.61
```

```
Directory Manager (existing master) password: # 目录管理器密码

Preparing replica for repl01.srv.world from dlp.srv.world
Creating SSL certificate for the Directory Server
Creating SSL certificate for the dogtag Directory Server
Creating SSL certificate for the Web Server
Exporting RA certificate
Copying additional files
Finalizing configuration
Packaging replica information into /var/lib/ipa/replica-info-rep
l01.srv.world.gpg
Adding DNS records for repl01.srv.world
Using reverse zone 0.0.10.in-addr.arpa.
```

```
scp /var/lib/ipa/replica-info-repl01.srv.world.gpg
root@repl01.srv.world:/var/lib/ipa/ # 将生成的密钥传输到副本主机
```

```
root@repl01.srv.world's password:
replica-info-repl01.srv.world.gpg 100% 35KB 34.6KB/s 00:00
```

7.1. FreeIPA

在FreeIPA服务器上，添加firewalld防火墙规则，允许FreeIPA复制服务：

```
firewall-cmd --add-service=freeipa-replication --permanent  
firewall-cmd --reload
```

在FreeIPA复制主机上，添加firewalld防火墙规则，允许FreeIPA服务：

```
firewall-cmd --add-service={ssh,dns,freeipa-ldap,freeipa-ldaps,f  
reeipa-replication} --permanent  
firewall-cmd --reload
```

下例为DNS设置 `--no-forwarders`，但如果设置它，需指定类似于 `--forwarder=x.x.x.x`：

7.1. FreeIPA

```
Directory Manager (existing master) password: # 目录管理器密码

Run connection check to master
Check connection from replica to remote master 'dlp.srv.world':
    Directory Service: Unsecure port (389): OK
    Directory Service: Secure port (636): OK
    Kerberos KDC: TCP (88): OK
    Kerberos Kpasswd: TCP (464): OK
    HTTP Server: Unsecure port (80): OK
    HTTP Server: Secure port (443): OK
    PKI-CA: Directory Service port (7389): OK

The following list of ports use UDP protocol and would need to be
checked manually:
    Kerberos KDC: UDP (88): SKIPPED
    Kerberos Kpasswd: UDP (464): SKIPPED

Connection from replica to master is OK.
Start listening on required ports for remote master check
Get credentials to log in to remote master
admin@SRV.WORLD password: # admin密码

Execute check on remote master
.....
.....
Global DNS configuration in LDAP server is empty
You can use 'dnsconfig-mod' command to set global DNS options that
would override settings in local named.conf files

Restarting the web server
```

在副本主机上获取Kerberos票证，并确保可以获取FreeIPA目录上的数据。如果可能，可以设置复制设置：

对于FreeIPA客户端，没有必要设置其他设置，即使一个服务器关闭，客户端仍可继续进行身份验证。

```
kinit admin
```

7.1. FreeIPA

```
Password for admin@SRV.WORLD: # admin密码
```

```
klist
```

```
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@SRV.WORLD

Valid starting     Expires            Service principal
03/21/2015 15:13:38 03/24/2015 15:13:35  krbtgt/SRV.WORLD@SRV.WORLD
```

```
ipa user-find
```

```
-----
4 users matched
-----
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
UID: 1219600000
GID: 1219600000
Account disabled: False
Password: True
Kerberos keys available: True

User login: cent
First name: CentOS
Last name: Linux
Home directory: /home/cent
Login shell: /bin/bash
Email address: cent@srv.world
UID: 1219600001
GID: 1219600001
Account disabled: False
Password: True
Kerberos keys available: True
.....
....
```


7.2. OpenLDAP

7.2.1. 配置LDAP服务器

配置LDAP服务器为本地网络共享用户帐户。

安装OpenLDAP服务器：

```
yum -y install openldap-servers openldap-clients
```

```
cp /usr/share/openldap-servers/DB_CONFIG.example  
/var/lib/ldap/DB_CONFIG
```

```
chown ldap. /var/lib/ldap/DB_CONFIG
```

```
systemctl start slapd  
systemctl enable slapd
```

设置OpenLDAP管理员密码：

```
slappasswd #生成加密密码
```

```
New password:  
Re-enter new password:  
{SSHA}xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

编辑 chrootpw.ldif 文件：

```
# 指定“olcRootPW”为上面生成的密码  
dn: olcDatabase={0}config,cn=config  
changetype: modify  
add: olcRootPW  
olcRootPW: {SSHA}xxxxxxxxxxxxxxxxxxxxxx
```

```
ldapadd -Y EXTERNAL -H ldap:// -f chrootpw.ldif
```

7.2. OpenLDAP

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={0}config,cn=config"
```

导入基本schema：

```
ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/schema/cosine.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=cosine,cn=schema,cn=config"
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=nis,cn=schema,cn=config"
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/schema/inetorgperson.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn/inetorgperson,cn=schema,cn=config"
```

在LDAP DB上设置域名：

```
slappasswd # 生成目录管理器密码
```

```
New password:  
Re-enter new password:  
{SSHA}xxxxxxxxxxxxxxxxxxxxxx
```

编辑 chdomain.ldif 文件：

```
# 将“dc=***,dc=***”部分替换为自己的域名  
# 指定“olcRootPW”为上面生成的密码  
dn: olcDatabase={1}monitor,cn=config  
changetype: modify  
replace: olcAccess  
olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth"  
        read by dn.base="cn=Manager,dc=srv,dc=world" read by * none  
  
dn: olcDatabase={2}hdb,cn=config  
changetype: modify  
replace: olcSuffix  
olcSuffix: dc=srv,dc=world  
  
dn: olcDatabase={2}hdb,cn=config  
changetype: modify  
replace: olcRootDN  
olcRootDN: cn=Manager,dc=srv,dc=world  
  
dn: olcDatabase={2}hdb,cn=config  
changetype: modify  
add: olcRootPW  
olcRootPW: {SSHA}xxxxxxxxxxxxxxxxxxxxxx  
  
dn: olcDatabase={2}hdb,cn=config  
changetype: modify  
add: olcAccess  
olcAccess: {0}to attrs=userPassword,shadowLastChange by  
        dn="cn=Manager,dc=srv,dc=world" write by anonymous auth by self  
        write by * none  
olcAccess: {1}to dn.base="" by * read  
olcAccess: {2}to * by dn="cn=Manager,dc=srv,dc=world" write by *  
        read
```

7.2. OpenLDAP

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f chdomain.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={1}monitor,cn=config"

modifying entry "olcDatabase={2}hdb,cn=config"

modifying entry "olcDatabase={2}hdb,cn=config"

modifying entry "olcDatabase={2}hdb,cn=config"
```

编辑 basedomain.ldif 文件：

```
# 将“dc=***,dc=***”部分替换为自己的域名
dn: dc=srv,dc=world
objectClass: top
objectClass: dcObject
objectclass: organization
o: Server World
dc: Srv

dn: cn=Manager,dc=srv,dc=world
objectClass: organizationalRole
cn: Manager
description: Directory Manager

dn: ou=People,dc=srv,dc=world
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=srv,dc=world
objectClass: organizationalUnit
ou: Group
```

```
ldapadd -x -D cn=Manager,dc=srv,dc=world -W -f basedomain.ldif
```

```
Enter LDAP Password: # 目录管理器密码  
adding new entry "dc=srv,dc=world"  
  
adding new entry "cn=Manager,dc=srv,dc=world"  
  
adding new entry "ou=People,dc=srv,dc=world"  
  
adding new entry "ou=Group,dc=srv,dc=world"
```

firewalld 防火墙规则：

```
firewall-cmd --add-service=ldap --permanent  
firewall-cmd --reload
```

7.2.2. 添加用户帐户

添加用户：

```
slappasswd # 生成加密密码
```

```
New password:  
Re-enter new password:  
{SSHA}xxxxxxxxxxxxxxxxxxxx
```

编辑 `ldapuser.ldif` 文件：

7.2. OpenLDAP

```
# 将“dc=***,dc=***”部分替换为自己的域名
dn: uid=cent,ou=People,dc=srv,dc=world
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: Cent
sn: Linux
userPassword: {SSHA}xxxxxxxxxxxxxxxxxxxx
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/cent

dn: cn=cent,ou=Group,dc=srv,dc=world
objectClass: posixGroup
cn: Cent
gidNumber: 1000
memberUid: cent
```

```
ldapadd -x -D cn=Manager,dc=srv,dc=world -W -f ldapuser.ldif
```

```
Enter LDAP Password:
adding new entry "uid=cent,ou=People,dc=srv,dc=world"

adding new entry "cn=cent,ou=Group,dc=srv,dc=world"
```

将本地passwd/group中的用户和组添加到LDAP目录：

编辑 ldapuser.sh 文件：

```
# 提取UID为1000-9999的本地用户和组
# 这里是一个示例，将“SUFFIX=***”替换为自己的域名
#!/bin/bash

SUFFIX='dc=srv,dc=world'
LDIF='ldapuser.ldif'

echo -n > $LDIF
GROUP_IDS=()
grep "x:[1-9][0-9][0-9][0-9]:" /etc/passwd | (while read TARGET_
```

```

USER
do
    USER_ID=$(echo "$TARGET_USER" | cut -d':' -f1)

    USER_NAME=$(echo "$TARGET_USER" | cut -d':' -f5 | cut -d' '
-f1,2)"
    [ ! "$USER_NAME" ] && USER_NAME="$USER_ID"

    LDAP_SN=$(echo "$USER_NAME" | cut -d' ' -f2)"
    [ ! "$LDAP_SN" ] && LDAP_SN="$USER_NAME"

    LASTCHANGE_FLAG=$(grep "${USER_ID}:" /etc/shadow | cut -d':'
-f3)"
    [ ! "$LASTCHANGE_FLAG" ] && LASTCHANGE_FLAG="0"

    SHADOW_FLAG=$(grep "${USER_ID}:" /etc/shadow | cut -d':'
-f9)"
    [ ! "$SHADOW_FLAG" ] && SHADOW_FLAG="0"

    GROUP_ID=$(echo "$TARGET_USER" | cut -d':' -f4)"
    [ ! "$(echo "${GROUP_IDS[@]}" | grep "$GROUP_ID")" ] && GROU
P_IDS=("${GROUP_IDS[@]}" "$GROUP_ID")

    echo "dn: uid=$USER_ID,ou=People,$SUFFIX" >> $LDIF
    echo "objectClass: inetOrgPerson" >> $LDIF
    echo "objectClass: posixAccount" >> $LDIF
    echo "objectClass: shadowAccount" >> $LDIF
    echo "sn: $LDAP_SN" >> $LDIF
    echo "givenName: $(echo "$USER_NAME" | awk '{print $1}')" >>
$LDIF
    echo "cn: $USER_NAME" >> $LDIF
    echo "displayName: $USER_NAME" >> $LDIF
    echo "uidNumber: $(echo "$TARGET_USER" | cut -d':' -f3)" >>
$LDIF
    echo "gidNumber: $(echo "$TARGET_USER" | cut -d':' -f4)" >>
$LDIF
    echo "userPassword: {crypt}$(grep "${USER_ID}:" /etc/shadow
| cut -d':' -f2)" >> $LDIF
    echo "gecos: $USER_NAME" >> $LDIF
    echo "loginShell: $(echo "$TARGET_USER" | cut -d':' -f7)" >>
$LDIF
    echo "homeDirectory: $(echo "$TARGET_USER" | cut -d':' -f6)"
```

7.2. OpenLDAP

```
>> $LDIF
    echo "shadowExpire: $(passwd -S "$USER_ID" | awk '{print $7}')
')" >> $LDIF
    echo "shadowFlag: $SHADOW_FLAG" >> $LDIF
    echo "shadowWarning: $(passwd -S "$USER_ID" | awk '{print $6
}')" >> $LDIF
    echo "shadowMin: $(passwd -S "$USER_ID" | awk '{print $4}')"
>> $LDIF
    echo "shadowMax: $(passwd -S "$USER_ID" | awk '{print $5}')"
>> $LDIF
    echo "shadowLastChange: $LASTCHANGE_FLAG" >> $LDIF
    echo >> $LDIF
done

for TARGET_GROUP_ID in "${GROUP_IDS[@]}"
do
    LDAP_CN=$(grep ":${TARGET_GROUP_ID}:" /etc/group | cut -d':'
' -f1)"

    echo "dn: cn=$LDAP_CN,ou=Group,$SUFFIX" >> $LDIF
    echo "objectClass: posixGroup" >> $LDIF
    echo "cn: $LDAP_CN" >> $LDIF
    echo "gidNumber: ${TARGET_GROUP_ID}" >> $LDIF

    for MEMBER_UID in $(grep ":${TARGET_GROUP_ID}:" /etc/passwd
| cut -d':' -f1,3)
    do
        UID_NUM=$(echo "$MEMBER_UID" | cut -d':' -f2)
        [ $UID_NUM -ge 1000 -a $UID_NUM -le 9999 ] && echo "mem
berUid: $(echo "$MEMBER_UID" | cut -d':' -f1)" >> $LDIF
        done
    echo >> $LDIF
done
)
```

```
sh ldapuser.sh
```

```
ldapadd -x -D cn=Manager,dc=srv,dc=world -W -f ldapuser.ldif
```

Enter LDAP Password:

```
adding new entry "uid=cent,ou=People,dc=srv,dc=world"  
adding new entry "uid=redhat,ou=People,dc=srv,dc=world"  
adding new entry "uid=ubuntu,ou=People,dc=srv,dc=world"  
adding new entry "uid=debian,ou=People,dc=srv,dc=world"  
adding new entry "cn=cent,ou=Group,dc=srv,dc=world"  
adding new entry "cn=redhat,ou=Group,dc=srv,dc=world"  
adding new entry "cn=ubuntu,ou=Group,dc=srv,dc=world"  
adding new entry "cn=debian,ou=Group,dc=srv,dc=world"
```

如果要删除LDAP用户或组，执行以下操作：

```
ldapdelete -x -W -D 'cn=Manager,dc=srv,dc=world'  
"uid=cent,ou=People,dc=srv,dc=world"
```

Enter LDAP Password:

```
ldapdelete -x -W -D 'cn=Manager,dc=srv,dc=world'  
"cn=cent,ou=Group,dc=srv,dc=world"
```

Enter LDAP Password:

7.2.3. 配置LDAP客户端

安装OpenLDAP客户端：

```
yum -y install openldap-clients nss-pam-ldapd
```

配置（`ldapserver=(LDAP服务器主机名或IP)`，`ldapbasedn="dc=(自己的域名)"`）：

```
authconfig --enableldap \  
--enableldapauth \  
--ldapserver=dlp.srv.world \  
--ldapbasedn="dc=srv,dc=world" \  
--enablemkhomedir \  
--update
```

7.2. OpenLDAP

```
exit
```

```
logout
```

```
CentOS Linux 7 (Core)
Kernel 3.10.0-123.20.1.el7.x86_64 on an x86_64
```

```
www login: redhat # LDAP用户
Password: # 密码
Creating directory '/home/redhat'.
```

```
[redhat@www ~]$ # 正常登录
```

```
[redhat@www ~]$ passwd # 尝试更改LDAP密码
Changing password for user redhat.
Enter login(LDAP) password: # 当前密码
New password: # 新密码
Retype new password: # 确认新密码
LDAP password information changed for redhat
passwd: all authentication tokens updated successfully.
```

如果启用了SELinux，需要添加一个规则以允许通过mkhomedir自动创建主目录：

编辑 `mkhomedir.te` 文件：

```
module mkhomedir 1.0;

require {
    type unconfined_t;
    type oddjob_mkhomedir_exec_t;
    class file entrypoint;
}

===== unconfined_t =====
allow unconfined_t oddjob_mkhomedir_exec_t:file entrypoint;
```

```
checkmodule -m -M -o mkhomedir.mod mkhomedir.te
```

```
checkmodule: loading policy configuration from mkhomedir.te
checkmodule: policy configuration loaded
checkmodule: writing binary representation (version 17) to mkhom
edir.mod
```

```
semodule_package --outfile mkhomedir.pp --module mkhomedir.mod
semodule -i mkhomedir.pp
```

7.2.4. 配置TLS

配置LDAP通过TLS以使连接安全。

首先[创建SSL证书](#)。

配置LDAP服务器：

```
cp /etc/pki/tls/certs/server.key \
/etc/pki/tls/certs/server.crt \
/etc/pki/tls/certs/ca-bundle.crt \
/etc/openldap/certs/
```

```
chown ldap. /etc/openldap/certs/server.key \
/etc/openldap/certs/server.crt \
/etc/openldap/certs/ca-bundle.crt
```

编辑 `mod_ssl.ldif` 文件：

```
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/openldap/certs/ca-bundle.crt
-
replace: olcTLS CertificateFile
olcTLS CertificateFile: /etc/openldap/certs/server.crt
-
replace: olcTLS CertificateKeyFile
olcTLS CertificateKeyFile: /etc/openldap/certs/server.key
```

7.2. OpenLDAP

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f mod_ssl.ldif
```

```
SASL/EXTERNAL authentication started  
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth  
SASL SSF: 0  
modifying entry "cn=config"
```

编辑 /etc/sysconfig/slapd 文件：

```
# 在“SLAPD_URLS=”部分添加“ldaps://”  
SLAPD_URLS="ldapi:/// ldap:/// ldaps:///"
```

```
systemctl restart slapd
```

为TLS连接配置LDAP客户端：

```
echo "TLS_REQCERT allow" >> /etc/openldap/ldap.conf  
echo "tls_reqcert allow" >> /etc/nslcd.conf  
authconfig --enableldapts --update
```

```
getsebool: SELinux is disabled
```

```
exit
```

```
logout  
  
CentOS Linux 7 (Core)  
Kernel 3.10.0-123.20.1.el7.x86_64 on an x86_64  
  
www login: redhat  
Password:  
Last login: Tue Aug 19 19:55:52 on ttys0  
[redhat@www ~]$ # 登录正常
```

7.2.5. OpenLDAP复制

7.2. OpenLDAP

如果OpenLDAP主服务器关闭，配置OpenLDAP复制以继续目录服务。

OpenLDAP主服务器称为“**Provider**”，OpenLDAP从服务器在OpenLDAP上称为“**Consumer**”。

先按照[第一节内容](#)在Provider和Consumer上配置好基本的LDAP服务。

配置LDAP Provider，添加syncprov模块：

编辑 mod_syncprov.ldif 文件：

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib64/openldap
olcModuleLoad: syncprov.la
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f mod_syncprov.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=module,cn=config"
```

编辑 syncprov.ldif 文件：

```
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpSessionLog: 100
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f syncprov.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "olcOverlay=syncprov,olcDatabase={2}hdb,cn=config"
```

配置LDAP Consumer：

编辑 `syncrepl.ldif` 文件：

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
add: olcSyncRepl
olcSyncRepl: rid=001
# LDAP服务器URI
provider=ldap://10.0.0.30:389/
bindmethod=simple
# 自己的域名
binddn="cn=Manager,dc=srv,dc=world"
# 目录管理器密码
credentials=password
searchbase="dc=srv,dc=world"
# 包括子树
scope=sub
schemachecking=on
type=refreshAndPersist
# [retry interval重试间隔] [retry times重试次数] [interval of re-
retry重试间隔] [re-retry times重试次数]
retry="30 5 300 3"
# 复制间隔
interval=00:00:05:00
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f syncrepl.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={2}hdb,cn=config"
```

确认设置以搜索数据：

```
ldapsearch -x -b 'ou=People,dc=srv,dc=world'
```

```
# People, srv.world
dn: ou=People,dc=srv,dc=world
objectClass: organizationalUnit
ou: People
...
...
```

配置LDAP客户端以绑定LDAP Consumer：

```
authconfig --ldapserver=dlp.srv.world,slave.srv.world --update
```

7.2.6. OpenLDAP 多主复制

Provider/Consumer的配置，不能在Consumer服务器上添加数据，但如果配置多主设置，则可以在任何主服务器上添加。

先按照[第一节内容](#)在所有服务器上配置好基本的LDAP服务。

在所有服务器如下配置，添加syncprov模块：

编辑 mod_syncprov.ldif 文件：

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib64/openldap
olcModuleLoad: syncprov.la
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f mod_syncprov.ldif
```

7.2. OpenLDAP

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=module,cn=config"
```

编辑 `syncprov.ldif` 文件：

```
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpSessionLog: 100
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f syncprov.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "olcOverlay=syncprov,olcDatabase={2}hdb,cn=config"
```

在所有服务器如下配置，但只有参数 `olcServerID` 和 `provider=***`，在每个服务器上设置不同的值：

编辑 `master01.ldif` 文件：

```

dn: cn=config
changetype: modify
replace: olcServerID
# 在每个服务器上指定唯一的ID号
olcServerID: 0

dn: olcDatabase={2}hdb,cn=config
changetype: modify
add: olcSyncRepl
olcSyncRepl: rid=001
# 指定其他LDAP服务器的URI
provider=ldap://10.0.0.50:389/
bindmethod=simple

# 自己的域名
binddn="cn=Manager,dc=srv,dc=world"
# 目录管理器密码
credentials=password
searchbase="dc=srv,dc=world"
# 包括子树
scope=sub
schemachecking=on
type=refreshAndPersist
# [retry interval重试间隔] [retry times重试次数] [interval of re-
retry重试间隔] [re-retry times重试次数]
retry="30 5 300 3"
# 复制间隔
interval=00:00:05:00
-
add: olcMirrorMode
olcMirrorMode: TRUE

dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov

```

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f master01.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"

modifying entry "olcDatabase={2}hdb,cn=config"

adding new entry "olcOverlay=syncprov,olcDatabase={2}hdb,cn=config"
```

配置LDAP客户端绑定所有LDAP服务器：

```
authconfig --ldapserver=slapd01.srv.world,slapd02.srv.world --
update
```

7.2.7. 安装phpLDAPadmin

安装phpLDAPadmin以通过Web浏览器操作LDAP服务器。

先[安装Apache httpd](#)并[安装PHP](#)。

安装phpLDAPadmin：

```
yum --enablerepo=epel -y install phpLDAPadmin # 从EPEL安装
```

编辑 `/etc/phpLDAPadmin/config.php` 文件：

```
# 将下面一行取消注释
$servers->setValue('login','attr','dn');
# 将下面一行注释
// $servers->setValue('login','attr','uid');
```

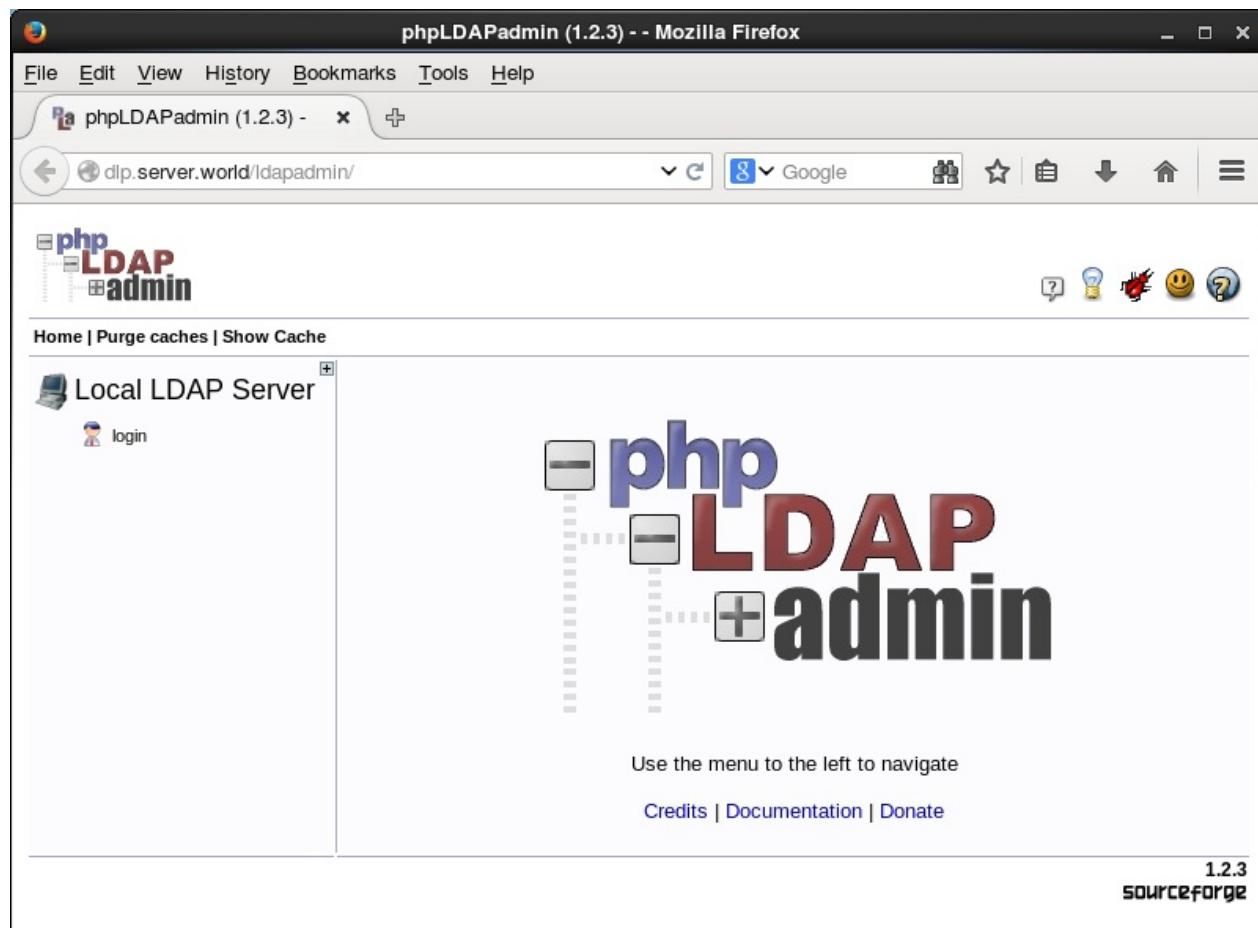
编辑 `/etc/httpd/conf.d/phpLDAPadmin.conf` 文件：

7.2. OpenLDAP

```
Alias /phpLDAPadmin /usr/share/phpLDAPadmin/htdocs
Alias /ldapadmin /usr/share/phpLDAPadmin/htdocs
<Directory /usr/share/phpLDAPadmin/htdocs>
<IfModule mod_authz_core.c>
# Apache 2.4
# 添加访问权限
Require local
Require ip 10.0.0.0/24
```

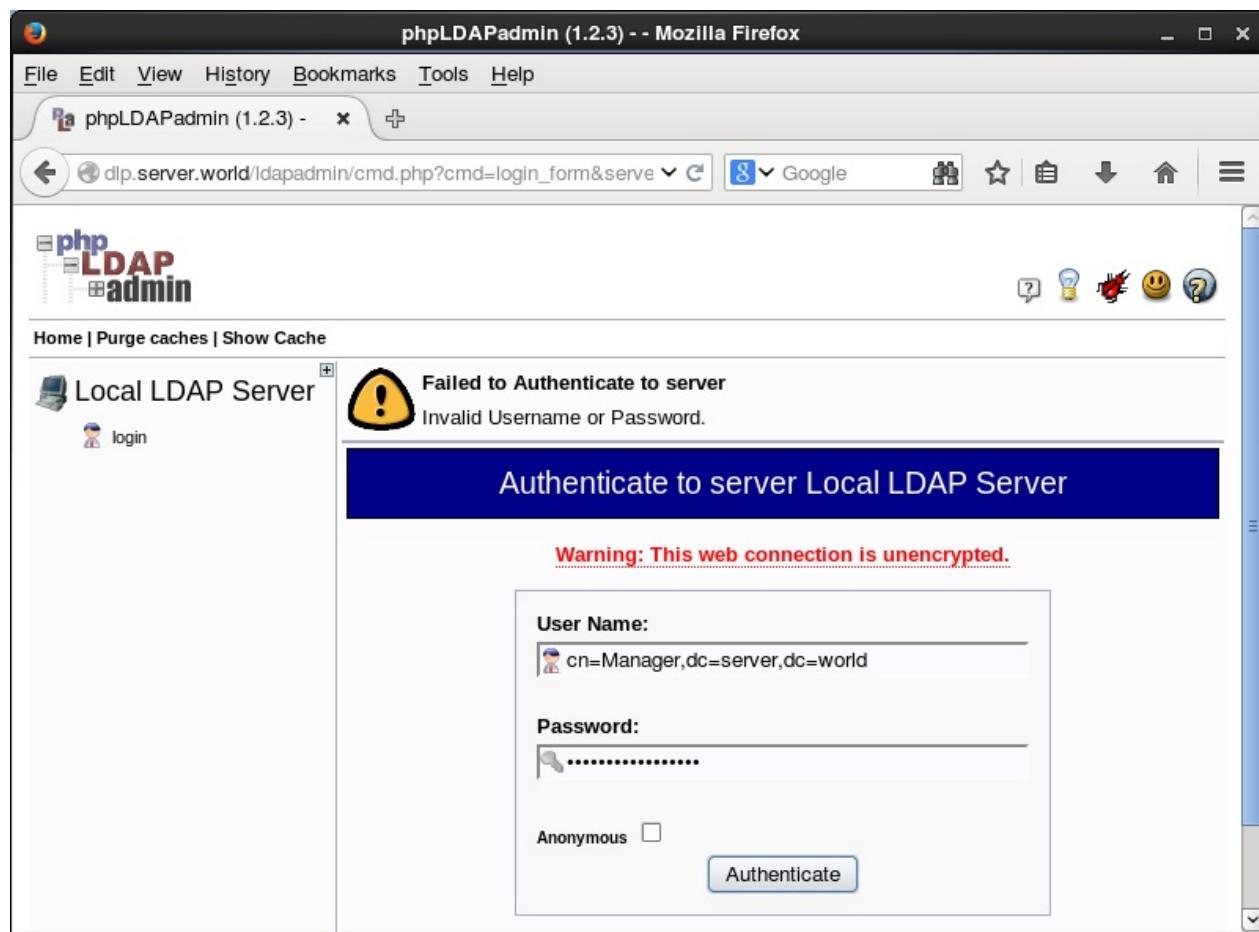
```
systemctl restart httpd
```

从http服务器允许的网络中的客户端访问 `http://(服务器主机名或IP)/ldapadmin/`，然后单击“login”：



使用目录管理器帐户进行身份验证。如下指定uname name（也可以用普通用户登录）：

7.2. OpenLDAP



登录成功，在这里可以管理LDAP服务器：

7.2. OpenLDAP

The screenshot shows the Mozilla Firefox browser displaying the phpLDAPAdmin interface at dlp.server.world/ldapadmin/cmd.php?server_id=1&redirect=t. The title bar reads "phpLDAPAdmin (1.2.3) -- Mozilla Firefox".

The left sidebar, titled "Local LDAP Server", shows a tree structure of LDAP entries under "dc=server, dc=world". The tree includes nodes for "cn=Manager", "ou=Group (3)", "ou=People (3)", and three user entries ("uid=maipo", "uid=redhat", "uid=ubuntu"). There are also two "Create new entry here" buttons.

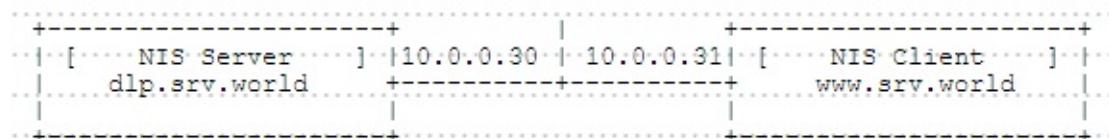
The right panel displays a large "phpLDAPadmin" logo. At the top, a message says "Authenticate to server" followed by "Successfully logged into server." Below the logo, instructions say "Use the menu to the left to navigate". At the bottom, links for "Credits | Documentation | Donate" are visible, along with the version information "1.2.3 sourceforge".

7.3. NIS

7.3.1. 配置NIS服务器

配置NIS服务器为本地网络共享用户帐户。

本例基于以下环境：



```
yum -y install ypserv rpcbind # 安装
```

设置NIS域名：

```
ypdomainname srv.world
```

```
echo "NISDOMAIN=srv.world" >> /etc/sysconfig/network
```

编辑 /var/yp/securenets 文件：

```
# 添加允许访问NIS服务器的IP地址
255.0.0.0      127.0.0.0
255.255.255.0  10.0.0.0
```

编辑 /etc/hosts 文件：

```
# 为NIS数据库添加服务器和客户端的IP地址
10.0.0.30      dlp.srv.world dlp
10.0.0.31      www.srv.world www
```

```
systemctl start rpcbind ypserv ypxfrd yppasswdd
systemctl enable rpcbind ypserv ypxfrd yppasswdd
```

更新NIS数据库：

```
/usr/lib64/yp/ypinit -m
```

At this point, we have to construct a list of the hosts which will run NIS servers. dlp is in the list of NIS server hosts. Please continue to add the names for the other hosts, one per line. When you are done with the list, type a <control D>.

```
next host to add: dlp.srv.world
next host to add: # 按Ctrl + D
The current list of NIS servers looks like this:
dlp.srv.world
Is this correct? [y/n: y] y # 输入y并回车
We need a few minutes to build the databases...
Building /var/yp/srv.world/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory `/var/yp/srv.world'
Updating passwd.byname...
Updating passwd.byuid...
Updating shadow.byname...
Updating group.byname...
Updating group.bygid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
Updating protocols.bynumber...
Updating protocols.byname...
Updating mail_aliases...
gmake[1]: Leaving directory `/var/yp/srv.world'
dlp.srv.world has been set up as a NIS master server.
Now you can run ypinit -s dlp on all slave server.
```

如果在本地服务器中添加了用户，也将它们应用于NIS数据库：

```
cd /var/yp
```

```
make
```

firewalld防火墙规则，需要允许NIS服务或端口。某些服务在重新启动时侦听不同的端口，因此固定端口并让firewalld允许它们：

编辑 `/etc/sysconfig/network` 文件：

```
# 添加到结尾  
YPSERV_ARGS="-p 944"  
YPXFRD_ARGS="-p 945"
```

编辑 `/etc/sysconfig/yppasswdd` 文件：

```
# 添加如下  
YPPASSWDD_ARGS="--port 946"
```

```
systemctl restart rpcbind ypserv ypxfrd yppasswdd
```

```
firewall-cmd --add-service=rpc-bind --permanent  
firewall-cmd --add-port=944/tcp --permanent  
firewall-cmd --add-port=944/udp --permanent  
firewall-cmd --add-port=945/tcp --permanent  
firewall-cmd --add-port=945/udp --permanent  
firewall-cmd --add-port=946/udp --permanent  
firewall-cmd --reload
```

7.3.2. 配置NIS客户端

```
yum -y install ypbind rpcbind # 在客户端安装
```

设置NIS域：

```
ypdomainname srv.world
```

```
echo "NISDOMAIN=srv.world" >> /etc/sysconfig/network
```

编辑 `/etc/hosts` 文件：

```
# 添加NIS服务器和客户端的IP地址  
10.0.0.30 dlp.srv.world dlp  
10.0.0.31 www.srv.world www
```

```
authconfig \
--enablenis \
--nisdomain=srv.world \
--nisserver=dlp.srv.world \
--enablemkhomedir \
--update
```

```
systemctl start rpcbind ypbind
systemctl enable rpcbind ypbind
```

```
exit
```

```
www login: redhat # NIS用户
Password: # NIS密码
Creating directory '/home/redhat'.
[redhat@www ~]$ # 登录成功

# 验证
[redhat@www ~]$ ypwhich
dlp.srv.world

# 尝试更改NIS密码
[redhat@www ~]$ yppasswd
Changing NIS account information for redhat on dlp.srv.world.
Please enter old password: # 当前密码
Changing NIS password for redhat on dlp.srv.world.
Please enter new password: # 新密码
Please retype new password: #确认新密码
The NIS password has been changed on dlp.srv.world.
```

如果启用了SELinux，需要添加一个规则以允许通过mkhomedir自动创建主目录：

编辑 `mkhomedir.te` 文件：

```

module mkhomedir 1.0;

require {
    type unconfined_t;
    type oddjob_mkhomedir_exec_t;
    class file entrypoint;
}

===== unconfined_t =====
allow unconfined_t oddjob_mkhomedir_exec_t:file entrypoint;

```

```
checkmodule -m -M -o mkhomedir.mod mkhomedir.te
```

```

checkmodule: loading policy configuration from mkhomedir.te
checkmodule: policy configuration loaded
checkmodule: writing binary representation (version 17) to mkhomedir.mod

```

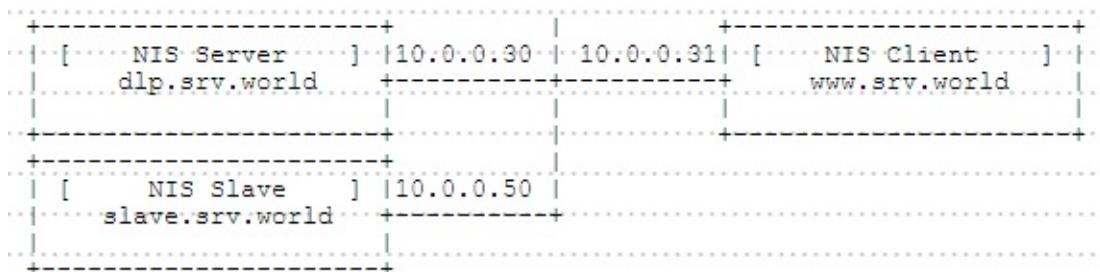
```

semodule_package --outfile mkhomedir.pp --module mkhomedir.mod
semodule -i mkhomedir.pp

```

7.3.3. 配置NIS从服务器

配置NIS从服务器以在NIS主服务器关闭时继续NIS服务。



先参照上一节内容，将NIS从服务器配置为一个NIS客户端。

在NIS从服务器上安装配置Ypserv（防火墙规则按照第一节NIS服务器的firewalld规则配置）：

```
yum -y install ypserv rpcbind
```

7.3. NIS

设置NIS域：

```
ypdomainname srv.world
```

```
echo "NISDOMAIN=srv.world" >> /etc/sysconfig/network
```

编辑 `/var/yp/securenets` 文件：

```
# 指定允许访问的网络  
255.0.0.0      127.0.0.0  
255.255.255.0  10.0.0.0
```

编辑 `/etc/hosts` 文件：

```
# 添加NIS服务器和客户端的IP地址  
10.0.0.30    dlp.srv.world dlp  
10.0.0.50    slave.srv.world slave
```

```
systemctl start rpcbind ypserv ypxfrd yppasswdd  
systemctl enable rpcbind ypserv ypxfrd yppasswdd
```

与NIS主服务器同步：

```
/usr/lib64/yp/ypinit -s dlp.srv.world
```

```
We will need a few minutes to copy the data from dlp.srv.world.  
Transferring group.bygid...  
Trying ypxfrd ... success  
...  
...  
At this point, make sure that /etc/passwd and /etc/group have  
been edited so that when the NIS is activated, the data bases yo  
u  
have just created will be used, instead of the /etc ASCII files.
```

NIS主服务器也需要是NIS客户端。参照上一节内容将NIS主服务器配置为NIS客户端：

编辑 `/var/yp/Makefile` 文件：

```
# 更改  
NOPUSH=false
```

更新NIS数据库：

```
/usr/lib64/yp/ypinit -m
```

```
At this point, we have to construct a list of the hosts which will run NIS  
servers. dlp.srv.world is in the list of NIS server hosts. Please continue to add  
the names for the other hosts, one per line. When you are done  
with the  
list, type a <control D>.  
next host to add: dlp.srv.world  
# 指定NIS从服务器  
next host to add: slave.srv.world  
next host to add: # 按Ctrl + D  
The current list of NIS servers looks like this:
```

```
dlp.srv.world  
slave.srv.world
```

```
Is this correct? [y/n: y] y # 输入y并回车  
We need a few minutes to build the databases...  
Building /var/yp/srv.world/ypservers...  
...
```

```
Now you can run ypinit -s dlp.srv.world on all slave server.
```

在NIS客户端上配置绑定NIS从服务器：

编辑 `/etc/yp.conf` 文件：

```
# 添加从服务器设置到最后  
domain srv.world server dlp.srv.world  
domain srv.world server slave.srv.world
```

```
systemctl restart ypbond
```


8. 文件服务器

- 8.1. FTP
 - 8.1.1. Vsftpd
 - 8.1.1.1. 安裝Vsftpd
 - 8.1.1.2. 安裝Vsftpd使用虛擬用戶
 - 8.1.1.3. Vsftpd + TLS
 - 8.1.2. ProFTPD
 - 8.1.2.1. 安裝ProFTPD
 - 8.1.2.2. ProFTPD + TLS
 - 8.1.3. Pure-FTPd
 - 8.1.3.1. 安裝Pure-FTPd
 - 8.1.3.2. Pure-FTPd + TLS
 - 8.1.3.3. Pure-FTPd + Clamav
 - 8.1.4. FTP客戶端
 - 8.1.4.1. CentOS客戶端
 - 8.1.4.2. Windows客戶端
- 8.2. Samba
 - 8.2.1. 完全訪問共享文件夾
 - 8.2.2. 受限訪問的共享文件夾
 - 8.2.3. Samba Winbind
 - 8.2.4. Samba AD DC
- 8.3. ownCloud

8.1. FTP

8.1.1. Vsftpd

8.1.1.1. 安装Vsftpd

安装[Vsftpd](#)以配置FTP服务器：

```
yum -y install vsftpd
```

编辑 `/etc/vsftpd/vsftpd.conf` 文件（具体配置及意义可以自行网上查找资料）：

8.1. FTP

```
# 禁止匿名登录
anonymous_enable=NO

# 取消注释（允许ascii模式）
ascii_upload_enable=YES
ascii_download_enable=YES

# 取消注释（启用chroot）
chroot_local_user=YES
chroot_list_enable=YES

# 取消注释（指定chroot列表）
chroot_list_file=/etc/vsftpd/chroot_list

# 取消注释
ls_recurse_enable=YES

# 更改（如果使用IPv4）
listen=YES

# 更改（如果不使用IPv6则关闭）
listen_ipv6=NO

# 添加以下内容到最后
# 指定根目录（如果不指定，用户的主目录成为FTP主目录）
local_root=public_html

# 使用本地时间
use_localtime=YES

# 关闭seccomp过滤器（如果无法登录，添加此行）
seccomp_sandbox=NO
```

编辑 `/etc/vsftpd/chroot_list` 文件：

```
# 添加允许移动到其主目录的用户
cent
```

```
systemctl start vsftpd  
systemctl enable vsftpd
```

firewalld防火墙规则：

```
firewall-cmd --add-service=ftp --permanent  
firewall-cmd --reload
```

如果启用了SELinux，更改布尔设置：

```
setsebool -P ftpd_full_access on
```

8.1.1.2. 安装Vsftpd使用虚拟用户

这里介绍完整的安装（使用非标准的FTP端口），内容可能有部分与上一节重复。

```
yum -y install vsftpd  
yum -y install psmisc net-tools systemd-devel libdb-devel perl-DBI  
# 安装vsftpd虚拟用户配置依赖包
```

```
systemctl start vsftpd  
systemctl enable vsftpd
```

配置vsftpd：

```
cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.bak # 备份默认配置文件  
  
sed -i "s/anonymous_enable=YES/anonymous_enable=NO/g"  
'/etc/vsftpd/vsftpd.conf'  
  
sed -i "s/#anon_upload_enable=YES/anon_upload_enable=NO/g"  
'/etc/vsftpd/vsftpd.conf'  
  
sed -i "s/#anon_mkdir_write_enable=YES/anon_mkdir_write_enable=YES/g"  
'/etc/vsftpd/vsftpd.conf'  
  
sed -i "s/#chown_uploads=YES/chown_uploads=NO/g"  
'/etc/vsftpd/vsftpd.conf'
```

8.1. FTP

```
sed -i "s/#async_abor_enable=YES/async_abor_enable=YES/g"  
'/etc/vsftpd/vsftpd.conf'  
  
sed -i "s/#ascii_upload_enable=YES/ascii_upload_enable=YES/g"  
'/etc/vsftpd/vsftpd.conf'  
  
sed -i "s/#ascii_download_enable=YES/ascii_download_enable=YES/g"  
'/etc/vsftpd/vsftpd.conf'  
  
sed -i "s/#ftpd_banner=Welcome to blah FTP  
service./ftpd_banner=Welcome to FTP service./g"  
'/etc/vsftpd/vsftpd.conf'  
  
echo -e  
"use_localtime=YES\nlisten_port=2021\nchroot_local_user=YES\nidle_se  
ssion_timeout=300 # 2021为指定访问端口,注意防火墙打开对应端口  
  
\ndata_connection_timeout=1\nguest_enable=YES\nguest_username=www  
# www为虚拟用户的宿主用户,可自行指定  
  
\nuser_config_dir=/etc/vsftpd/vconf\nvirtual_use_local_privs=YES #  
vconf为虚拟用户配置文件保存路径,可自行设定  
  
\npasv_min_port=30050\npasv_max_port=30090 # 30050/30090为pasv使用  
端口,注意防火墙打开对应端口  
  
\naccept_timeout=5\nconnect_timeout=1  
  
\nallow_writeable_chroot=YES  
  
\nmax_clients=3\nmax_per_ip=1\nanon_max_rate=800000" >>  
/etc/vsftpd/vsftpd.conf # 3为最大客户端连接数;1为同一IP最大连接  
数;800000为最大带宽使用,单位为byte/s
```

建立虚拟用户文件：

编辑 /etc/vsftpd/virtusers.txt 文件（virtusers.txt文件名可随意设置，文本格式即可）：

8.1. FTP

```
# 第一行账号，第二行密码，注意：不能使用root做用户名（系统保留）其他系统默认  
用户也不要使用  
ftpuser1  
123456  
ftpuser2  
123456
```

生成虚拟用户数据文件：

```
db_load -T -t hash -f /etc/vsftpd/virtusers.txt  
/etc/vsftpd/virtusers.db # virtusers.txt与上面文件名一致；virtusers.db文件  
名可随意设置，扩展名为db
```

```
chmod 600 /etc/vsftpd/virtusers.db
```

设定PAM验证文件，并指定对虚拟用户数据库文件进行读取：

```
cp /etc/pam.d/vsftpd /etc/pam.d/vsftpd.bak # 备份默认验证文件
```

编辑 /etc/pam.d/vsftpd 文件：

```
# 在最上面添加以下内容  
# db=/etc/vsftpd/virtusers中virtusers为上面.db文件的文件名  
auth sufficient /lib64/security/pam_userdb.so db=/etc/vsftpd/vir  
tusers  
account sufficient /lib64/security/pam_userdb.so db=/etc/vsftpd/  
virtusers
```

新建宿主用户（如果用www等已有用户，则不需要新建）：

```
useradd vsftpd -d /home/wwwroot -s /sbin/nologin # vsftpd可自行设定  
(不与系统用户冲突)，/home/wwwroot 为用户主目录，也可自行设定
```

```
chown -R vsftpd:vsftpd /home/wwwroot
```

若宿主用户为www，主目录为 /home/wwwroot ，只用运行以下命令：

```
chown -R www:www /home/wwwroot
```

建立虚拟用户个人配置文件：

```
mkdir /etc/vsftpd/vconf # 前面系统配置文件中设定的路径
```

8.1. FTP

```
mkdir -p /home/wwwroot/ftpuser1 # ftpuser1为用户ftpuser1准备的主目录
```

编辑 /etc/vsftpd/vconf/ftpuser1 文件（前面设置的虚拟用户名）：

```
local_root=home/wwwroot/ftpuser1
write_enable=YES
anon_world_readable_only=NO
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_other_write_enable=YES
```

```
systemctl restart vsftpd.service
```

虚拟用户一些配置参数：

当 virtual_use_local_privs=YES 时，虚拟用户和本地用户有相同的权限；

当 virtual_use_local_privs=NO 时，虚拟用户和匿名用户有相同的权限，默认是NO。

当 virtual_use_local_privs=YES ， write_enable=YES 时，虚拟用户具有写权限（上传、下载、删除、重命名）。

当 virtual_use_local_privs=NO ， write_enable=YES ， anon_world_readable_only=YES ， anon_upload_enable=YES 时，虚拟用户不能浏览目录，只能上传文件，无其他权限。

当 virtual_use_local_privs=NO ， write_enable=YES ， anon_world_readable_only=NO ， anon_upload_enable=NO 时，虚拟用户只能下载文件，无其他权限。

当 virtual_use_local_privs=NO ， write_enable=YES ， anon_world_readable_only=NO ， anon_upload_enable=YES 时，虚拟用户只能上传和下载文件，无其他权限。

当 virtual_use_local_privs=NO ， write_enable=YES ， anon_world_readable_only=NO ， anon_mkdir_write_enable=YES 时，虚拟用户只能下载文件和创建文件夹，无其他权限。

当 virtual_use_local_privs=NO ， write_enable=YES ， anon_world_readable_only=NO ， anon_other_write_enable=YES 时，虚拟用户只能下载、删除和重命名文件，无其他权限。

8.1.1.3. Vsftpd + TLS

为Vsftpd启用SSL/TLS以使用安全的FTP连接。

创建自签名证书：

```
cd /etc/pki/tls/certs
```

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout vsftpd.pem -out
vsftpd.pem -days 365
```

```
Generating a 2048 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/pki/tls/certs/vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN # 国家
State or Province Name (full name) [Some-State]:SC # 省
Locality Name (eg, city) []:CD # 城市
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GTS
# 公司
Organizational Unit Name (eg, section) []:Server World # 部门
Common Name (eg, YOUR name) []:www.srv.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
```

```
chmod 400 vsftpd.pem
```

配置Vsftpd：

编辑 `/etc/vsftpd/vsftpd.conf` 文件：

```
# 添加到最后  
# 固定PASV端口  
pasv_enable=YES  
pasv_min_port=21000  
pasv_max_port=21010  
  
# 启用TLS  
rsa_cert_file=/etc/pki/tls/certs/vsftpd.pem  
ssl_enable=YES  
ssl_ciphers=HIGH  
ssl_tlsv1=YES  
ssl_sslv2=NO  
ssl_sslv3=NO  
force_local_data_ssl=YES  
force_local_logins_ssl=YES
```

```
systemctl restart vsftpd
```

firewalld防火墙规则，允许固定的PASV端口：

```
firewall-cmd --add-port=21000-21010/tcp --permanent  
firewall-cmd --reload
```

8.1.2. ProFTPD

8.1.2.1. 安装ProFTPD

安装[ProFTPD](#)以配置FTP服务器。

```
yum --enablerepo=epel -y install proftpd # 从EPEL安装
```

编辑 `/etc/proftpd.conf` 文件：

8.1. FTP

```
# 更改自己的主机名
ServerName      "www.srv.world"

# 更改自己的电子邮件地址
ServerAdmin     root@srv.world

# 添加以下内容，获取访问日志和验证日志
ExtendedLog    /var/log/proftpd/access.log WRITE,READ default
ExtendedLog    /var/log/proftpd/auth.log AUTH auth
```

编辑 `/etc/ftpusers` 文件：

```
# 添加禁止访问FTP的用户
test
```

```
systemctl start proftpd
systemctl enable proftpd
```

`firewalld`防火墙规则：

```
firewall-cmd --add-service=ftp --permanent
firewall-cmd --reload
```

如果启用了SELinux，更改布尔设置：

```
setsebool -P ftpd_full_access on
```

8.1.2.2. ProFTPD + TLS

为ProFTPD配置使用SSL/TLS。

创建自签名证书：

```
cd /etc/pki/tls/certs
openssl req -x509 -nodes -newkey rsa:2048 -keyout proftpd.pem -out
proftpd.pem -days 365
```

```
Generating a 2048 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/pki/tls/certs/proftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN # 国家
State or Province Name (full name) [Some-State]:SC # 省
Locality Name (eg, city) []:CD # 城市
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GTS
# 公司
Organizational Unit Name (eg, section) []:Server World # 部门
Common Name (eg, YOUR name) []:www.srv.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
```

```
chmod 600 proftpd.pem
```

配置ProFTPD：

编辑 /etc/proftpd.conf 文件：

```
# 添加到最后
# 固定PASV端口
PassivePorts 21000 21010

# 启用TLS
TLSEngine on
TLSRequired on
TLSProtocol TLSv1.2
TLSLog /var/log/proftpd/tls.log
TLSRSACertificateFile /etc/pki/tls/certs/proftpd.pem
TLSRSACertificateKeyFile /etc/pki/tls/certs/proftpd.pem
```

8.1. FTP

```
systemctl restart proftpd
```

firewalld防火墙规则，允许固定的PASV端口：

```
firewall-cmd --add-port=21000-21010/tcp --permanent  
firewall-cmd --reload
```

8.1.3. Pure-FTPd

8.1.3.1. 安装Pure-FTPD

安装[Pure-FTPD](#)以配置FTP服务器。

```
yum --enablerepo=epel -y install pure-ftpd #从EPEL安装
```

编辑 `/etc/pure-ftpd/pure-ftpd.conf` 文件：

```
# # 禁止匿名登录  
NoAnonymous yes  
  
# 取消注释（如果仅使用IPv4）  
IPV4only yes  
  
# 取消注释（如果仅使用IPv6）  
IPV6only yes
```

```
systemctl start pure-ftpd  
systemctl enable pure-ftpd
```

firewalld防火墙规则：

```
firewall-cmd --add-service=ftp --permanent  
firewall-cmd --reload
```

如果启用了SELinux，更改布尔设置：

```
setsebool -P ftpd_full_access on
```

8.1.3.2. Pure-FTPd + TLS

为Pure-FTPD配置使用SSL/TLS。

创建自签名证书：

```
cd /etc/pki/tls/certs
```

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout pure-ftpd.pem -out pure-ftpd.pem -days 365
```

```
Generating a 2048 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/pki/tls/certs/pure-ftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN # 国家
State or Province Name (full name) [Some-State]:SC # 省
Locality Name (eg, city) []:CD # 城市
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GTS
# 公司
Organizational Unit Name (eg, section) []:Server World # 部门
Common Name (eg, YOUR name) []:www.srv.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
```

```
chmod 600 pure-ftpd.pem
```

配置Pure-FTPD：

编辑 /etc/pure-ftpd/pure-ftpd.conf 文件：

```
# 取消注释并固定PASV端口  
PassivePorts 21000 21010  
  
# 取消注释  
TLS 1
```

```
systemctl restart pure-ftpd
```

firewalld防火墙规则，允许固定的PASV端口：

```
firewall-cmd --add-port=21000-21010/tcp --permanent  
firewall-cmd --reload
```

8.1.3.3. Pure-FTPd + Clamav

配置Pure-FTPD + Clamav通过FTP连接及时扫描病毒。

先[安装Clamav](#)。

配置Pure-FTPD：

```
编辑 /etc/pure-ftpd/pure-ftpd.conf 文件：
```

```
# 取消注释  
CallUploadScript yes
```

```
编辑 /usr/local/bin/pure-ftpd_clamscan.sh 文件：
```

```
#!/bin/bash  
  
/usr/bin/clamscan --remove --quiet --no-summary "$1"
```

```
chmod 755 /usr/local/bin/pure-ftpd_clamscan.sh
```

```
编辑 /etc/systemd/system/clamav.pure-ftpd.service 文件：
```

```
[Unit]
Description=Clamav Scanning Service for Pure-FTPD
Before=pure-ftpd.service

[Service]
Type=simple
RemainAfterExit=yes
ExecStart=/usr/sbin/pure-uploadscript -B -r /usr/local/bin/pure-
ftpd_clamscan.sh

[Install]
WantedBy=multi-user.target
```

```
systemctl --system daemon-reload
systemctl restart clamav.pure-ftpd pure-ftpd
systemctl enable clamav.pure-ftpd
```

通过FTP上传[测试病毒](#)到服务器，确保设置正常工作。病毒将上传，但会被立即删除。

8.1.4. FTP客户端

8.1.4.1. CentOS客户端

```
yum -y install lftp
```

默认情况下禁止使用root帐户连接，因此可以使用普通用户访问FTP服务器。

```
lftp -u cent www.srv.world # 格式为： lftp [选项] [主机名或IP]
```

```
Password: # 用户密码
lftp cent@www.srv.world:~>

# 显示FTP服务器上的当前目录
lftp cent@www.srv.world:~> pwd
ftp://cent@www.srv.world

# 显示本地服务器上的当前目录
lftp cent@www.srv.world:~> !pwd
```

8.1. FTP

```
/home/redhat

# 显示FTP服务器上当前目录中的文件
lftp cent@www.srv.world:~> ls
drwxr-xr-x    2 1000      1000          23 Jul 19 01:33 public_h
tml
-rw-r--r--    1 1000      1000          399 Jul 20 16:32 test.py

# 显示本地服务器上当前目录中的文件
lftp cent@www.srv.world:~> !ls -l
total 12
-rw-rw-r--  1 redhat redhat 10 Jul 20 14:30 redhat.txt
-rw-rw-r--  1 redhat redhat 10 Jul 20 14:59 test2.txt
-rw-rw-r--  1 redhat redhat 10 Jul 20 14:59 test.txt

# 更改目录
lftp cent@www.srv.world:~> cd public_html
lftp cent@www.srv.world:~/public_html> pwd
ftp://cent@www.srv.world%2Fhome/cent/public_html

# 将文件上传到FTP服务器
# "-a"表示ascii模式（默认为二进制模式）
lftp cent@www.srv.world:~> put -a redhat.txt
22 bytes transferred
Total 2 files transferred
lftp cent@www.srv.world:~> ls
drwxr-xr-x    2 1000      1000          23 Jul 19 01:33 public_h
tml
-rw-r--r--    1 1000      1000          10 Jul 20 17:01 redhat.t
xt
-rw-r--r--    1 1000      1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000      1000          10 Jul 20 17:01 test.txt

# 上传一些文件到FTP服务器
lftp cent@www.srv.world:~> mput -a test.txt test2.txt
22 bytes transferred
Total 2 files transferred
lftp cent@www.srv.world:~> ls
drwxr-xr-x    2 1000      1000          23 Jul 19 01:33 public_h
tml
-rw-r--r--    1 1000      1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000      1000          10 Jul 20 17:06 test.txt
```

8.1. FTP

```
-rw-r--r--    1 1000      1000          10 Jul 20 17:06 test2.txt

# 从FTP服务器下载文件
# "-a"表示ascii模式(默认为二进制模式)
lftp cent@www.srv.world:~> get -a test.py
416 bytes transferred

# 从FTP服务器下载一些文件
lftp cent@www.srv.world:~> mget -a test.txt test2.txt
20 bytes transferred
Total 2 files transferred

# 在FTP服务器上的当前目录中创建一个目录
lftp cent@www.srv.world:~> mkdir testdir
mkdir ok, `testdir' created
lftp cent@www.srv.world:~> ls
drwxr-xr-x    2 1000      1000          23 Jul 19 01:33 public_html
-rw-r--r--    1 1000      1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000      1000          10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000      1000          10 Jul 20 17:06 test2.txt
drwxr-xr-x    2 1000      1000          6 Jul 20 17:16 testdir
226 Directory send OK.

# 删除FTP服务器上当前目录中的目录
lftp cent@www.srv.world:~> rmdir testdir
rmdir ok, `testdir' removed
lftp cent@www.srv.world:~> ls
drwxr-xr-x    2 1000      1000          23 Jul 19 01:33 public_html
-rw-r--r--    1 1000      1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000      1000          10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000      1000          10 Jul 20 17:06 test2.txt

# 删除FTP服务器上当前目录中的文件
lftp cent@www.srv.world:~> rm test2.txt
rm ok, `test2.txt' removed
lftp cent@www.srv.world:~> ls
drwxr-xr-x    2 1000      1000          23 Jul 19 01:33 public_html
```

8.1. FTP

```
tml  
-rw-r--r--    1 1000      1000          399 Jul 20 16:32 test.py  
-rw-r--r--    1 1000      1000          10 Jul 20 17:06 test.txt  
  
# 删除FTP服务器上当前目录中的一些文件  
lftp cent@www.srv.world:~> rm redhat.txt test.txt  
rm ok, 2 files removed  
lftp cent@www.srv.world:~> ls  
drwxr-xr-x    2 1000      1000          23 Jul 19 01:33 public_h  
tml  
  
# 使用"![command]"执行命令  
lftp cent@www.srv.world:~> !cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
...  
...  
redhat:x:1001:1001::/home/redhat:/bin/bash  
  
# 退出  
lftp cent@www.srv.world:~> quit  
221 Goodbye.
```

使用TLS：

编辑 `~/.lftp` 文件：

```
set ftp:ssl-auth TLS  
set ftp:ssl-force true  
set ftp:ssl-protect-list yes  
set ftp:ssl-protect-data yes  
set ftp:ssl-protect-fxp yes  
set ssl:verify-certificate no
```

```
lftp -u cent www.srv.world
```

Password:

```
lftp cent@www.srv.world:~>
```

8.1.4.2. Windows客户端

可下载使用[FileZilla](#)。

使用TLS：

会弹出提示接受证书，确定即可。

8.2. Samba

安装[Samba](#)以配置文件服务器。

8.2.1. 完全访问共享文件夹

本例演示创建一个完全访问的共享文件夹，任何人都可以读取和写入，并且不需要身份验证。

```
yum -y install samba samba-client
```

```
mkdir /home/share
```

```
chmod 777 /home/share
```

编辑 `/etc/samba/smb.conf` 文件：

8.2. Samba

```
# 添加如下内容
unix charset = UTF-8
dos charset = CP932

# 如下更改 (Windows默认)
workgroup = WORKGROUP

# 取消注释并更改为允许的IP地址
hosts allow = 127. 10.0.0.

# 添加 (无验证)
security = user
passdb backend = tdbSAM
map to guest = Bad User

# 将以下内容添加到最后
[Share] # 任意名称
    path = /home/share # 共享目录
    writable = yes # 可写
    guest ok = yes # 允许guest
    guest only = yes # 仅guest
    create mode = 0777 # 完全访问的文件
    directory mode = 0777 # 完全访问的目录
```

```
systemctl start smb nmb
systemctl enable smb nmb
```

firewalld防火墙规则：

```
firewall-cmd --add-service=samba --permanent
firewall-cmd --reload
```

如果启用了SELinux，更改SELinux上下文：

```
setsebool -P samba_enable_home_dirs on
restorecon -R /home/share
```

8.2.2. 受限访问的共享文件夹

本例演示创建需要用户身份验证的共享文件夹。

```
yum -y install samba samba-client
```

```
groupadd security
```

```
mkdir /home/security
```

```
chgrp security /home/security
```

```
chmod 770 /home/security
```

编辑 `/etc/samba/smb.conf` 文件：

```
# 添加如下内容
unix charset = UTF-8

# 如下更改 (Windows默认)
workgroup = WORKGROUP

# 取消注释并更改为允许的IP地址
hosts allow = 127. 10.0.0.

# 将以下内容添加到最后
[Security] # 任意名称
    path = /home/security
    writable = yes
    create mode = 0770
    directory mode = 0770
    guest ok = no # 不允许guest
    valid users = @security # 仅允许security组
```

```
systemctl start smb nmb
systemctl enable smb nmb
```

在Samba中添加用户：

```
smbpasswd -a cent
```

```
New SMB password: # 设置密码  
Retype new SMB password: # 确认密码  
Added user cent.
```

```
usermod -G security cent
```

firewalld防火墙规则：

```
firewall-cmd --add-service=samba --permanent  
firewall-cmd --reload
```

如果启用了SELinux：

```
setsebool -P samba_enable_home_dirs on  
restorecon -R /home/security
```

8.2.3. Samba Winbind

使用Samba Winbind加入Windows Active Directory域。

本教程需要LAN中的Windows Active Directory域服务。

本例演示在以下环境中进行配置：

Domain Server	:	Windows Server 2012 R2
Domain Name	:	FD3S01
Realm	:	SRV.WORLD
Hostname	:	fd3s.srv.world

```
yum -y install samba-winbind samba-winbind-clients pam_krb5
```

配置Winbind：

将DNS更改为Active Directory主机：

```
nmcli c modify ens3 ipv4.dns 10.0.0.100
```

```
nmcli c down ens3; nmcli c up ens3
```

```
authconfig \
--enablekrb5 \
--krb5kdc=fd3s.srv.world \
--krb5adminserver=fd3s.srv.world \
--krb5realm=SRV.WORLD \
--enablewinbind \
--enablewinbindauth \
--smbsecurity=ads \
--smbrealm=SRV.WORLD \
--smbservers=fd3s.srv.world \
--smbworkgroup=FD3S01 \
--winbindtemplatehomedir=/home/%U \
--winbindtemplatehell=/bin/bash \
--enablemkhomedir \
--enablewinbindusedefaultdomain \
--update
```

如果提示下面的错误，没有关系：

```
Job for winbind.service failed. See 'systemctl status winbind.service' and 'journalctl -xn' for details.
```

加入Windows Active Directory域：

```
net ads join -U Administrator # 格式：net ads join -U [AD的管理员用户]
```

```
Enter Serverworld's password:
Using short domain name -- FD3S01
Joined 'SMB' to dns domain 'srv.world'
```

```
systemctl start winbind
systemctl enable winbind
```

显示域信息：

```
net ads info
```

```
LDAP server: 10.0.0.100
LDAP server name: fd3s.srv.world
Realm: SRV.WORLD
Bind Path: dc=SRV,dc=WORLD
LDAP port: 389
Server time: Sat, 09 Jul 2016 01:03:54 JST
KDC server: 10.0.0.100
Server time offset: -4
```

显示AD用户信息：

```
wbinfo -u
```

```
administrator
guest
serverworld
krbtgt
```

尝试切换到AD用户：

```
su - serverworld
```

```
Creating directory '/home/serverworld'.
[serverworld@smb ~]$
```

8.2.4. Samba AD DC

配置Samba Active Directory域控制器（Domain Controller）。

有兴趣的话参考[这里](#)

8.3. ownCloud

ownCloud是一个基于PHP的自建网盘，主要功能包括文件管理（内建文件分享）、音乐、日历、联系人等。有电脑和手机的客户端。与之类似的还有Nextcloud。

安装MariaDB数据库服务器，Apache httpd并配置好PHP和使用SSL。

```
yum --enablerepo=epel -y install php-pear-MDB2-Driver-mysqli php-pear-Net-Curl #从EPEL安装一些需要的软件包
```

```
wget  
http://download.owncloud.org/download/repositories/stable/CentOS_7/c  
e:stable.repo -P /etc/yum.repos.d  
  
yum -y install owncloud  
  
systemctl restart httpd
```

在MariaDB中为ownCloud添加用户和数据库：

```
mysql -u root -p
```

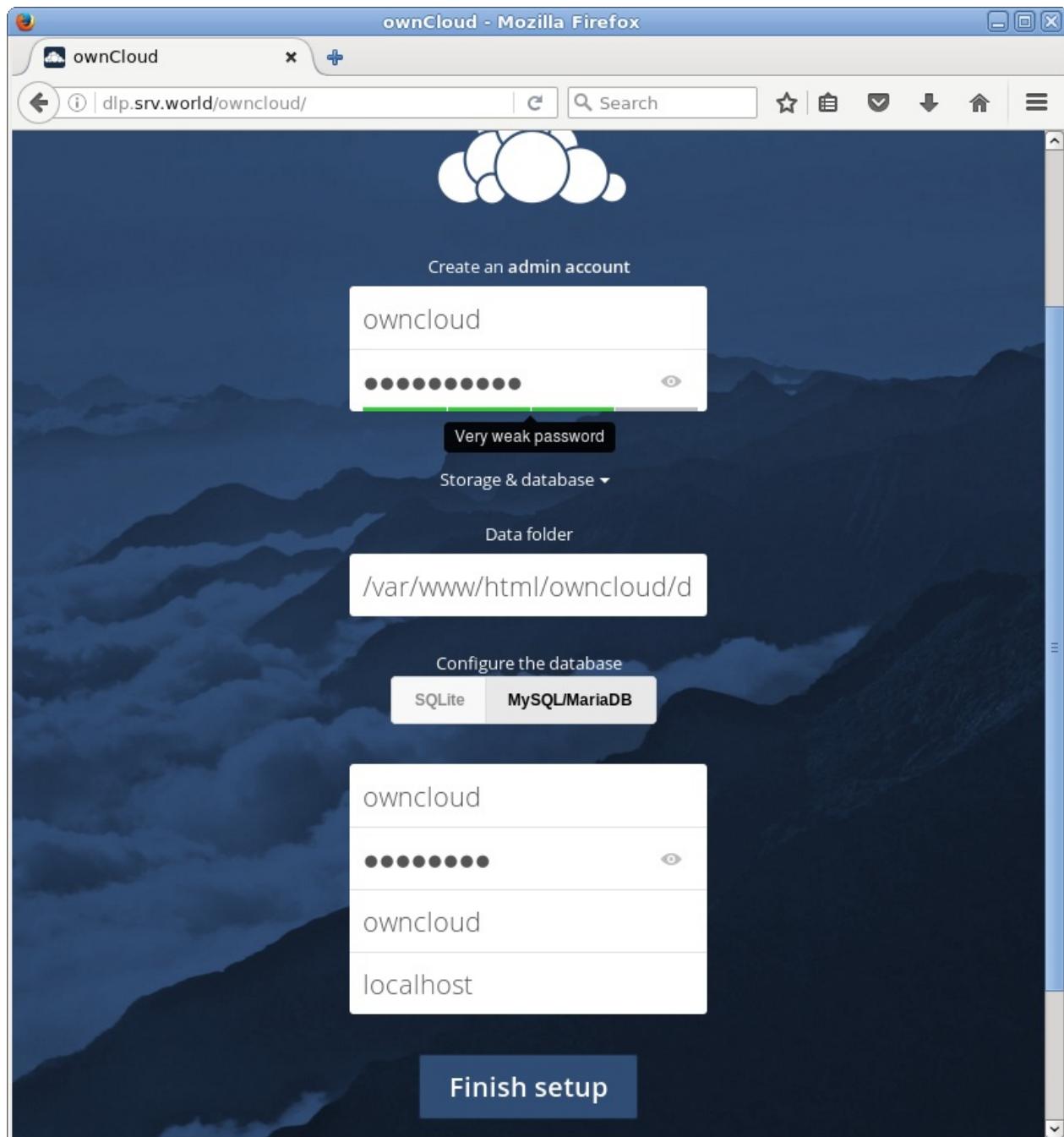
```
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 10  
Server version: 5.5.47-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> create database owncloud;  
Query OK, 1 row affected (0.00 sec)  
MariaDB [(none)]> grant all privileges on owncloud.* to owncloud  
@'localhost' identified by 'password';  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> exit  
Bye
```

如果启用了SELinux，如下更改规则：

```
semanage fcontext -a -t httpd_sys_rw_content_t /var/www/html/own  
cloud/apps  
semanage fcontext -a -t httpd_sys_rw_content_t /var/www/html/own  
cloud/config  
restorecon /var/www/html/owncloud/apps  
restorecon /var/www/html/owncloud/apps
```

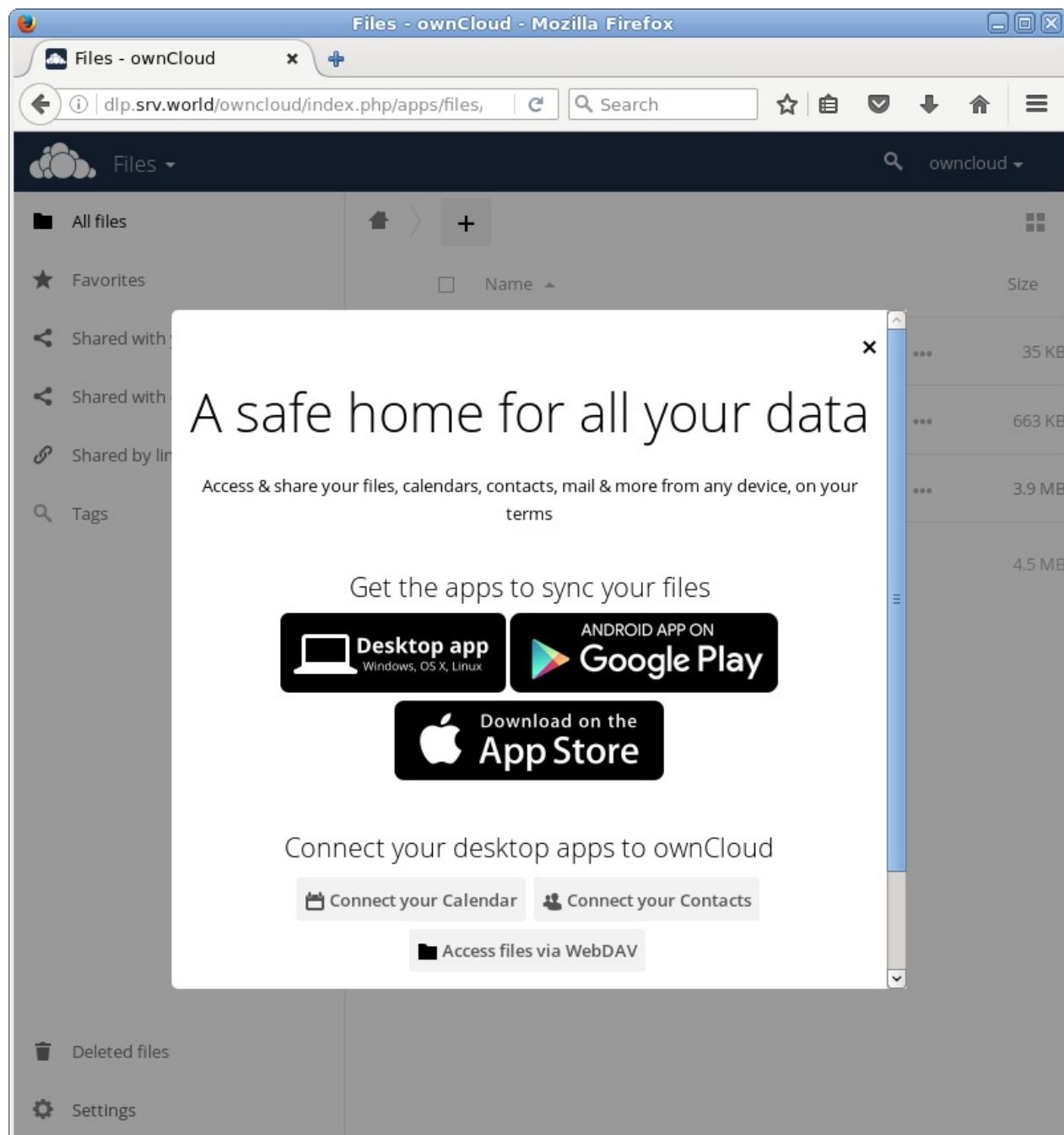
客户端计算机的Web浏览器访问 `http://(服务器的主机名或IP地址)/owncloud/`。
显示以下界面，为ownCloud添加管理员帐号，并在数据库部分中点击“MySQL/MariaDB”，输入MariaDB的用户名密码及数据库名称。所有完成后，点击“Finish Setup”继续：

8.3. ownCloud



如果数据库连接正确，显示如下欢迎界面：

8.3. ownCloud



下面是ownCloud主页。可以使用ownCloud作为云存储：

8.3. ownCloud

The screenshot shows the ownCloud web interface within a Mozilla Firefox browser window. The title bar reads "Files - ownCloud - Mozilla Firefox". The address bar shows the URL "dlp.srv.world/owncloud/index.php/apps/files/". The main interface has a dark header with the "Files" logo and a search bar containing "owncloud". On the left, there's a sidebar with links: "All files", "Favorites", "Shared with you", "Shared with others", "Shared by link", "Tags", "Deleted files", and "Settings". The main area displays a file list with three items:

	Name	Size
	Documents	35
	Photos	663
	ownCloud Manual.pdf	3.9 M

Below the list, it says "2 folders and 1 file" with a total size of "4.5 M".

9. 邮件服务器

- 9.1. 安装Postfix
 - 9.1.1. 安装Postfix
 - 9.1.2. 虚拟域
 - 9.1.3. Postfix + Clamav + Amavisd
- 9.2. 安装Dovecot
- 9.3. 配置SSL
- 9.4. 邮件日志报告
 - 9.4.1. pflogsumm
 - 9.4.2. AWstats
- 9.5. WebMail
 - 9.5.1. SquirrelMail
 - 9.5.2. RainLoop
 - 9.5.3. RoundCube
- 9.6. iRedMail

9.1. 安裝Postfix

9.1.1. 安裝Postfix

安装[Postfix](#)以配置SMTP服务器。

CentOS系统最小安装也会安装Postfix，如果没有，则运行以下命令：

```
yum -y install postfix
```

本例演示配置[SMTP-Auth](#)以使用Dovecot的SASL功能

```
编辑 /etc/postfix/main.cf 文件:
```

```
# 取消注释并指定主机名  
myhostname = mail.srv.world  
  
# 取消注释并指定域名  
mydomain = srv.world  
  
# 取消注释  
myorigin = $mydomain  
  
# 更改  
inet_interfaces = all  
  
# 添加  
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain  
  
# 取消注释并指定您的本地网络  
mynetworks = 127.0.0.0/8, 10.0.0.0/24  
  
# 取消注释（使用Maildir）  
home_mailbox = Maildir/  
  
# 添加  
smtpd_banner = $myhostname ESMTP  
  
# 添加以下内容到最后  
# 将电子邮件大小限制为10M  
message_size_limit = 10485760  
# 将邮箱限制为1G  
mailbox_size_limit = 1073741824  
  
# SMTP-Auth  
smtpd_sasl_type = dovecot  
smtpd_sasl_path = private/auth  
smtpd_sasl_auth_enable = yes  
smtpd_sasl_security_options = noanonymous  
smtpd_sasl_local_domain = $myhostname  
smtpd_recipient_restrictions = permit_mynetworks,permit_auth_destination,permit_sasl_authenticated,reject
```

```
systemctl restart postfix  
systemctl enable postfix
```

firewalld防火墙规则，允许SMTP服务（25/TCP）：

```
firewall-cmd --add-service=smtp --permanent  
firewall-cmd --reload
```

9.1.2. 虚拟域

配置Postfix以使用虚拟域发送另一个不同于原始域的域名的电子邮件。

例如：

当前域名为： `srv.world`

新域名为： `virtual.host`

用户“cent”有一个电子邮件地址：`cent@mail.srv.world`

用户“redhat”有一个电子邮件地址：`cent@mail.virtual.host`

“redhat”用户使用的邮件地址“@”前面的名称和“cent”用户一样：

设置虚拟域：

编辑 `/etc/postfix/main.cf` 文件：

```
# 添加到文件的最后  
virtual_alias_domains = virtual.host  
virtual_alias_maps = hash:/etc/postfix/virtual
```

编辑 `/etc/postfix/virtual` 文件：

```
# 添加到文件的开头  
cent@mail.virtual.host redhat
```

```
postmap /etc/postfix/virtual
```

```
systemctl reload postfix
```

9.1.3. Postfix + Clamav + Amavisd

使用Postfix + Clamav配置病毒扫描。

[安装Clamav](#)。

安装Amavisd和Clamav Server，并首先启动Clamav Server：

```
yum --enablerepo=epel -y install amavisd-new clamav-server clamav-server-systemd # 从EPEL安装
```

```
cp /usr/share/doc/clamav-server*/clamd.sysconfig  
/etc/sysconfig/clamd.amavisd
```

编辑 `/etc/sysconfig/clamd.amavisd` 文件：

```
# 取消注释并更改  
CLAMD_CONFIGFILE=/etc/clamd.d/amavisd.conf  
CLAMD_SOCKET=/var/run/clamd.amavisd/clamd.sock
```

编辑 `/etc/tmpfiles.d/clamd.amavisd.conf` 文件：

```
d /var/run/clamd.amavisd 0755 amavis amavis -
```

编辑 `/usr/lib/systemd/system/clamd@.service` 文件：

```
# 添加到最后  
[Install]  
WantedBy=multi-user.target
```

```
systemctl start clamd@amavisd  
systemctl enable clamd@amavisd
```

配置Amavisd：

编辑 `/etc/amavisd/amavisd.conf` 文件：

9.1. 安装Postfix

```
# 更改自己的域名  
$mydomain = 'srv.world';  
  
# 更改自己的主机名  
$myhostname = 'mail.srv.world';  
  
# 取消注释  
$notify_method = 'smtp:[127.0.0.1]:10025';  
$forward_method = 'smtp:[127.0.0.1]:10025';
```

```
systemctl start amavisd spamassassin  
systemctl enable amavisd spamassassin
```

配置Postfix：

编辑 `/etc/postfix/main.cf` 文件:

```
# 添加到最后  
content_filter=smtp-amavis:[127.0.0.1]:10024
```

编辑 `/etc/postfix/master.cf` 文件:

```
# 添加到最后
smtp-amavis unix - - n - 2 smtp
  -o smtp_data_done_timeout=1200
  -o smtp_send_xforward_command=yes
  -o disable_dns_lookups=yes
127.0.0.1:10025 inet n - n - - smtpd
  -o content_filter=
  -o local_recipient_maps=
  -o relay_recipient_maps=
  -o smtpd_restriction_classes=
  -o smtpd_client_restrictions=
  -o smtpd_helo_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o mynetworks=127.0.0.0/8
  -o strict_rfc821_envelopes=yes
  -o smtpd_error_sleep_time=0
  -o smtpd_soft_error_limit=1001
  -o smtpd_hard_error_limit=1000
```

```
systemctl restart postfix
```

以下几行在此配置后添加到电子邮件的标头部分，并且已知病毒的电子邮件将不会发送到客户端。

9.1. 安裝Postfix



9.2. 安裝Dovecot

安装Dovecot以配置POP/IMAP服务器。

```
yum -y install dovecot
```

本例演示配置为向Postfix提供SASL功能：

编辑 /etc/dovecot/dovecot.conf 文件：

```
# 取消注释  
protocols = imap pop3 lmtp  
  
# 取消注释并更改（如果不使用IPv6）  
listen = *
```

编辑 /etc/dovecot/conf.d/10-auth.conf 文件：

```
# 取消注释并更改（允许纯文本身份验证）  
disable_plaintext_auth = no  
  
# 添加  
auth_mechanisms = plain login
```

编辑 /etc/dovecot/conf.d/10-mail.conf 文件：

```
# 取消注释并添加  
mail_location = mailldir:~/Maildir
```

编辑 /etc/dovecot/conf.d/10-master.conf 文件：

9.2. 安裝Dovecot

```
# 取消注释并添加以下内容
# Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth {
    mode = 0666
    user = postfix
    group = postfix
}
```

编辑 `/etc/dovecot/conf.d/10-ssl.conf` 文件：

```
# 更改 (不需要SSL)
ssl = no
```

```
systemctl start dovecot
systemctl enable dovecot
```

`firewalld`防火墙规则，允许POP（110/TCP）和IMAP（143/TCP）服务：

```
firewall-cmd --add-port={110/tcp,143/tcp} --permanent
firewall-cmd --reload
```

安装完成后可以在客户端测试登录。

9.3. 配置SSL

[先创建证书](#)

配置Postfix和Dovecot使用SSL：

编辑 `/etc/postfix/main.cf` 文件：

```
# 添加到最后
smtpd_use_tls = yes
smtpd_tls_cert_file = /etc/pki/tls/certs/server.crt
smtpd_tls_key_file = /etc/pki/tls/certs/server.key
smtpd_tls_session_cache_database = btree:/etc/postfix/smtpd_scache
```

编辑 `/etc/postfix/master.cf` 文件：

```
# 取消注释
smtps      inet  n       -       n       -       -          smtpd
 -o syslog_name=postfix/smtps
 -o smtpd_tls_wrappermode=yes
```

编辑 `/etc/dovecot/conf.d/10-ssl.conf` 文件：

```
# 更改
ssl = yes

# 指定证书
ssl_cert = </etc/pki/tls/certs/server.crt
ssl_key = </etc/pki/tls/certs/server.key
```

`systemctl restart postfix dovecot`

`firewalld`防火墙规则，允许SMTPS（465/TCP），POP3S（995/TCP），IMAPS（993/TCP）：

9.3. 配置SSL

```
firewall-cmd --add-service={pop3s,imaps} --permanent  
firewall-cmd --add-port=465/tcp --permanent  
firewall-cmd --reload
```

客户端的设置。

9.4. 邮件日志报告

9.4.1. pflogsumm

安装Postfix日志报告工具[pflogsumm](#)。

安装postfix-perl-scripts软件包：

```
yum -y install postfix-perl-scripts
```

生成昨天的日志摘要：

```
perl /usr/sbin/pflogsumm -d yesterday /var/log/maillog
```

```
Postfix log summaries for Jul 14
```

```
Grand Totals
```

```
-----
```

```
messages
```

```
2 received
5 delivered
0 forwarded
0 deferred
0 bounced
0 rejected (0%)
0 reject warnings
0 held
0 discarded (0%)
```

```
2879 bytes received
6572 bytes delivered
1 senders
1 sending hosts/domains
2 recipients
2 recipient hosts/domains
```

```
Per-Hour Traffic Summary
```

```
-----
```

9.4. 邮件日志报告

time rejected	received	delivered	deferred	bounced
<hr/>				
0000-0100 0	0	0	0	0
0100-0200 0	0	0	0	0
0200-0300 0	0	0	0	0
0300-0400 0	0	0	0	0
0400-0500 0	0	0	0	0
0500-0600 0	0	0	0	0
0600-0700 0	0	0	0	0
0700-0800 0	0	0	0	0
0800-0900 0	0	0	0	0
0900-1000 0	0	0	0	0
1000-1100 0	2	5	0	0
1100-1200 0	0	0	0	0
1200-1300 0	0	0	0	0
1300-1400 0	0	0	0	0
1400-1500 0	0	0	0	0
1500-1600 0	0	0	0	0
1600-1700 0	0	0	0	0
1700-1800 0	0	0	0	0
1800-1900 0	0	0	0	0

9.4. 邮件日志报告

1900-2000	0	0	0	0
0				
2000-2100	0	0	0	0
0				
2100-2200	0	0	0	0
0				
2200-2300	0	0	0	0
0				
2300-2400	0	0	0	0
0				

Host/Domain Summary: Message Delivery

sent	cnt	bytes	defers	avg dly	max dly	host/domain
3	4119	0	0.4 s	0.8 s	srv.world	
2	2453	0	0.1 s	0.1 s	mail.srv.world	

Host/Domain Summary: Messages Received

msg	cnt	bytes	host/domain
2	2879	mail.srv.world	

Senders by message count

2	cent@mail.srv.world
---	---------------------

Recipients by message count

3	redhat@srv.world
2	cent@mail.srv.world

Senders by message size

2879	cent@mail.srv.world
------	---------------------

Recipients by message size

4119	redhat@srv.world
2453	cent@mail.srv.world

9.4. 邮件日志报告

```
message deferral detail: none

message bounce detail (by relay): none

message reject detail: none

message reject warning detail: none

message hold detail: none

message discard detail: none

smtp delivery failures: none

Warnings
-----
  tlsmgr (total: 6)
    3  redirecting the request to postfix-owned data_directory /var/li...
    3  request to update table btree:/etc/postfix/smtpd_scache in non-...

Fatal Errors: none

Panics: none

Master daemon messages
-----
  4  daemon started -- version 2.10.1, configuration /etc/postfix
  3  terminating on signal 15
  1  reload -- version 2.10.1, configuration /etc/postfix
```

```
crontab -e
```

```
# 每天1:00AM发送邮件日志摘要到root
00 01 * * * perl /usr/sbin/pflogsumm -e -d yesterday /var/log/maillog | mail -s 'Logwatch for Postfix' root
```

9.4.2. AWstats

安装邮件日志报告工具[AWstats](#)。

先[安装Apache httpd](#)

安装AWstats：

```
yum --enablerepo=epel -y install awstats #从EPEL安装
```

编辑 `/etc/awstats/awstats.mail.srv.world.conf` 文件（自动生成的“awstats.(主机名).conf”文件）：

```
# 更改
LogFile="/usr/share/awstats/tools/maillogconvert.pl standard < /var/log/maillog |"

# 更改
LogType=M

# 注释并添加以下内容
#LogFormat=1
LogFormat="%time2 %email %email_r %host %host_r %method %url %code %bytesd"

# 如下更改
LevelForBrowsersDetection=0
LevelForOSDetection=0
LevelForRefererAnalyze=0
LevelForRobotsDetection=0
LevelForSearchEnginesDetection=0
LevelForKeywordsDetection=0
LevelForFileTypesDetection=0
LevelForWormsDetection=0

# 如下更改
ShowMonthStats=UHB
ShowDaysOfMonthStats=HB
ShowDaysOfWeekStats=HB
ShowHoursStats=HB
ShowDomainsStats=0
ShowHostsStats=HBL
ShowRobotsStats=0
ShowEMailSenders=HBML
```

9.4. 邮件日志报告

```
ShowEMailReceivers=HBML  
ShowSessionsStats=0  
ShowPagesStats=0  
ShowFileTypesStats=0  
ShowOSStats=0  
ShowBrowsersStats=0  
ShowOriginStats=0  
ShowKeyphrasesStats=0  
ShowKeywordsStats=0  
ShowMiscStats=0  
ShowHTTPErrorsStats=0  
ShowSMTPErrorsStats=1
```

编辑 `/etc/httpd/conf.d/awstats.conf` 文件：

```
# 添加允许访问的IP范围  
Require ip 10.0.0.0/24
```

```
systemctl restart httpd  
  
/usr/share/awstats/wwwroot/cgi-bin/awstats.pl -update -  
config=mail.srv.world -configdir=/etc/awstats # 手动更新报告（每小时  
Cron自动更新）
```

```
Create/Update database for config "/etc/awstats/awstats.mail.srv
.world.conf" by AWStats version 7.4 (build 20150714)
From data in log file "/usr/share/awstats/tools/maillogconvert.p
l standard < /var/log/maillog |"...
Phase 1 : First bypass old records, searching new record...
Searching new records from beginning of log file...
Phase 2 : Now process new records (Flush history on disk after 2
0000 hosts)...
Jumped lines in file: 0
Parsed lines in file: 5
  Found 1 dropped records,
  Found 0 comments,
  Found 0 blank records,
  Found 0 corrupted records,
  Found 0 old records,
  Found 4 new qualified records.
```

通过客户端上的Web浏览器访问 `http://(主机名或IP)/awstats/awstats.pl`。显示以下页面，可以查看邮件日志摘要：

9.4. 邮件日志报告

Statistics for mail.srv.world (2016-07) - main - Mozilla Firefox

Statistics for mail.srv.w... +

mail.srv.world/awstats/awstats.pl

Last Update: 14 Jul 2016 - 11:17

Reported period: Jul 2016 OK

Summary

When:
Monthly history
Days of month
Days of week
Hours

Who:
Hosts
Full list
Unresolved IP Address

Sender EMail
Full list
Last

Receiver EMail
Full list
Last

Navigation:
Others:
SMTP Error codes

Summary

Reported period	Month Jul 2016
First	14 Jul 2016 - 10:06
Last	14 Jul 2016 - 10:48
Unique visitors	3
Number of visits	4
Pages	4
Mails	4
Size	5.21 KB (1.3 KB/Mails)
Mails successfully sent	4
Mails failed/refused	0

Monthly history

Month	Unique visitors	Mails	Size
Jan 2016	0	0	0
Feb 2016	0	0	0
Mar 2016	0	0	0
Apr 2016	0	0	0
May 2016	0	0	0
Jun 2016	0	0	0
Jul 2016	3	4	5.21 KB
Aug 2016	0	0	0
Sep 2016	0	0	0
Oct 2016	0	0	0
Nov 2016	0	0	0
Dec 2016	0	0	0
Total	3	4	5.21 KB

9.5. WebMail

安装好[SMTP服务器](#)，[IMAP服务器](#)，[Apache httpd](#)并配置好[PHP](#)和[使用SSL](#)。

所有示例基于以下环境：`www.srv.world` 为WebMail安装的服务器，`mail.srv.world` 为SMTP/IMAP服务器。

9.5.1. SquirrelMail

[SquirrelMail](#)是一个用PHP开发的Web邮件系统。它具备一个客户端邮件程序所应拥有的一切功能，包括支持增强型的MIME、地址簿、文件夹操作等等功能。

```
yum --enablerepo=epel -y install squirrelmail 从EPEL安装

curl -O http://www.squirrelmail.org/plugins/compatibility-2.0.16-
1.0.tar.gz

curl -O http://www.squirrelmail.org/plugins/empty_trash-2.0-
1.2.2.tar.gz

curl -O http://www.squirrelmail.org/plugins/secure_login-1.4-
1.2.8.tar.gz

tar zxvf compatibility-2.0.16-1.0.tar.gz -C
/usr/share/squirrelmail/plugins

tar zxvf empty_trash-2.0-1.2.2.tar.gz -C
/usr/share/squirrelmail/plugins

tar zxvf secure_login-1.4-1.2.8.tar.gz -C
/usr/share/squirrelmail/plugins

rm -f /*.tar.gz

/usr/share/squirrelmail/config/conf.pl # 运行配置脚本
```

SquirrelMail Configuration : Read: config.php (1.4.0)

Main Menu --

- 1. Organization Preferences
- 2. Server Settings

```
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color off
S Save data
Q Quit
```

Command >> 1 # 选择

SquirrelMail Configuration : Read: config.php (1.4.0)

Organization Preferences

```
1. Organization Name      : SquirrelMail
2. Organization Logo       : ../images/sm_logo.png
3. Org. Logo Width/Height : (308/111)
4. Organization Title     : SquirrelMail $version
5. Signout Page           :
6. Top Frame               : _top
7. Provider link          : http://squirrelmail.org/
8. Provider name          : SquirrelMail
```

```
R Return to Main Menu
C Turn color off
S Save data
Q Quit
```

Command >> 5 # 选择

When users click the Sign Out button they will be logged out and then sent to signout_page. If signout_page is left empty, (hit space and then return) they will be taken, as normal, to the default and rather sparse SquirrelMail signout page.

```
[]: /webmail # 更改注销页面
```

SquirrelMail Configuration : Read: config.php (1.4.0)

Organization Preferences

```
1. Organization Name      : SquirrelMail
2. Organization Logo      : ../images/sm_logo.png
3. Org. Logo Width/Height : (308/111)
4. Organization Title     : SquirrelMail $version
5. Signout Page           : /webmail
6. Top Frame               : _top
7. Provider link           : http://squirrelmail.org/
8. Provider name           : SquirrelMail
```

R Return to Main Menu
C Turn color off
S Save data
Q Quit

Command >> r # 返回菜单

SquirrelMail Configuration : Read: config.php (1.4.0)

Main Menu --

```
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages
```

D. Set pre-defined settings for specific IMAP servers

C Turn color off
S Save data
Q Quit

Command >> 2 # 选择

SquirrelMail Configuration : Read: config.php (1.4.0)

Server Settings

General

- ```

1. Domain : localhost
2. Invert Time : false
3. Sendmail or SMTP : Sendmail

A. Update IMAP Settings : localhost:143 (uw)
B. Change Sendmail Config : /usr/sbin/sendmail
```

R Return to Main Menu

C Turn color off

S Save data

Q Quit

Command >> 1 # 选择

The domain name is the suffix at the end of all email addresses.

If

for example, your email address is jdoe@example.com, then your domain

would be example.com.

```
[localhost]: srv.world # 更改自己的域名
SquirrelMail Configuration : Read: config.php (1.4.0)
```

### Server Settings

#### General

- ```
-----  
1. Domain : srv.world  
2. Invert Time : false  
3. Sendmail or SMTP : Sendmail  
  
A. Update IMAP Settings : localhost:143 (uw)  
B. Change Sendmail Config : /usr/sbin/sendmail
```

R Return to Main Menu

C Turn color off

S Save data

Q Quit

Command >> 3 # 选择

You now need to choose the method that you will use for sending messages in SquirrelMail. You can either connect to an SMTP server or use sendmail directly.

1. Sendmail
2. SMTP

```
Your choice [1/2] [1]: 2 # 更改为SMTP  
SquirrelMail Configuration : Read: config.php (1.4.0)
```

Server Settings

General

-
1. Domain : srv.world
 2. Invert Time : false
 3. Sendmail or SMTP : SMTP
-
- A. Update IMAP Settings : localhost:143 (uw)
 - B. Update SMTP Settings : localhost:25

R Return to Main Menu

C Turn color off

S Save data

Q Quit

Command >> A # 选择

```
SquirrelMail Configuration : Read: config.php (1.4.0)
```

Server Settings

General

-
1. Domain : srv.world
 2. Invert Time : false
 3. Sendmail or SMTP : SMTP

IMAP Settings

-
4. IMAP Server : localhost

```
5. IMAP Port : 143
6. Authentication type : login
7. Secure IMAP (TLS) : false
8. Server software : uw
9. Delimiter : /
B. Update SMTP Settings : localhost:25
H. Hide IMAP Server Settings
```

```
R Return to Main Menu
C Turn color off
S Save data
Q Quit
```

Command >> 4 # 选择

This is the hostname where your IMAP server can be contacted.

[localhost]: mail.srv.world # 指定IMAP服务器

SquirrelMail Configuration : Read: config.php (1.4.0)

----- Server Settings

General

```
1. Domain : srv.world
2. Invert Time : false
3. Sendmail or SMTP : SMTP
```

IMAP Settings

```
4. IMAP Server : mail.srv.world
5. IMAP Port : 143
6. Authentication type : login
7. Secure IMAP (TLS) : false
8. Server software : uw
9. Delimiter : /
```

```
B. Update SMTP Settings : localhost:25
H. Hide IMAP Server Settings
```

```
R Return to Main Menu
C Turn color off
```

```
S Save data
Q Quit
```

Command >> 8 # 选择

Each IMAP server has its own quirks. As much as we tried to stick to standards, it doesn't help much if the IMAP server doesn't follow the same principles. We have made some work-arounds for some of these servers. If you would like to use them, please select your IMAP server. If you do not wish to use these work-arounds, you can set this to "other", and none will be used.

bincimap	= Binc IMAP server
courier	= Courier IMAP server
cyrus	= Cyrus IMAP server
dovecot	= Dovecot Secure IMAP server
exchange	= Microsoft Exchange IMAP server
hmailserver	= hMailServer
macosx	= Mac OS X Mailserver
mercury32	= Mercury/32
uw	= University of Washington's IMAP server
gmail	= IMAP access to Google mail (Gmail) accounts
other	= Not one of the above servers

[uw]: dovecot # 更改为Dovecot
SquirrelMail Configuration : Read: config.php (1.4.0)

----- Server Settings

General

1. Domain : srv.world
2. Invert Time : false
3. Sendmail or SMTP : SMTP

IMAP Settings

4. IMAP Server : mail.srv.world
5. IMAP Port : 143

```
6. Authentication type      : login
7. Secure IMAP (TLS)       : false
8. Server software         : dovecot
9. Delimiter                : /
B. Update SMTP Settings    : localhost:25
H. Hide IMAP Server Settings
```

R Return to Main Menu
C Turn color off
S Save data
Q Quit

Command >> 9 # 选择

This is the delimiter that your IMAP server uses to distinguish between folders. For example, Cyrus uses '.' as the delimiter and a complete folder would look like 'INBOX.Friends.Bob', while UW uses '/' and would look like 'INBOX/Friends/Bob'. Normally this should be left at 'detect' but if you are sure you know what delimiter your server uses, you can specify it here.

To have it autodetect the delimiter, set it to 'detect'.

```
[/]: detect # 输入“detect”
```

```
SquirrelMail Configuration : Read: config.php (1.4.0)
```

----- Server Settings

General

```
-----  
1. Domain                  : srv.world
2. Invert Time              : false
3. Sendmail or SMTP         : SMTP
```

IMAP Settings

9.5. WebMail

```
4. IMAP Server          : mail.srv.world
5. IMAP Port            : 143
6. Authentication type  : login
7. Secure IMAP (TLS)    : false
8. Server software      : dovecot
9. Delimiter            : detect

B. Update SMTP Settings : localhost:25
H. Hide IMAP Server Settings

R  Return to Main Menu
C  Turn color off
S  Save data
Q  Quit

Command >> B  # 选择
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Server Settings

General
-----
1. Domain              : srv.world
2. Invert Time          : false
3. Sendmail or SMTP     : SMTP

SMTP Settings
-----
4. SMTP Server          : localhost
5. SMTP Port             : 25
6. POP before SMTP       : false
7. SMTP Authentication   : none
8. Secure SMTP (TLS)     : false
9. Header encryption key :

A. Update IMAP Settings : mail.srv.world:143 (dovecot)
H. Hide SMTP Settings

R  Return to Main Menu
C  Turn color off
S  Save data
Q  Quit
```

```
Command >> 4 # 选择

This is the hostname of your SMTP server.
[localhost]: mail.srv.world # 指定SMTP服务器
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
```

Server Settings

General

- ```
1. Domain : srv.world
2. Invert Time : false
3. Sendmail or SMTP : SMTP
```

#### SMTP Settings

- ```
4. SMTP Server : mail.srv.world
5. SMTP Port : 25
6. POP before SMTP : false
7. SMTP Authentication : none
8. Secure SMTP (TLS) : false
9. Header encryption key :
```
-
- ```
A. Update IMAP Settings : mail.srv.world:143 (dovecot)
H. Hide SMTP Settings
```

R Return to Main Menu  
C Turn color off  
S Save data  
Q Quit

Command >> 7 # 选择

If you have already set the hostname and port number, I can try to automatically detect the mechanisms your SMTP server supports. Auto-detection is \*optional\* - you can safely say "n" here.

Try to detect auth mechanisms? [y/N]: y # "yes" (自动)  
Trying to detect supported methods (SMTP)...  
Testing none: SUPPORTED

```
Testing login: SUPPORTED
Testing plain: SUPPORTED
Testing CRAM-MD5: NOT SUPPORTED
Testing DIGEST-MD5: NOT SUPPORTED
```

What authentication mechanism do you want to use for SMTP connections?

```
none - Your SMTP server does not require authorization.
login - Plaintext. If you can do better, you probably should.
plain - Plaintext. If you can do better, you probably should.
cram-md5 - Slightly better than plaintext.
digest-md5 - Privacy protection - better than cram-md5.
```

\*\*\* YOUR SMTP SERVER MUST SUPPORT THE MECHANISM YOU CHOOSE HERE  
\*\*\*

If you don't understand or are unsure, you probably want "none"

none, login, plain, cram-md5, or digest-md5 [none]: login # 本例选择“login”

SMTP authentication uses IMAP username and password by default.

Would you like to use other login and password for all SquirrelMail

SMTP connections? [y/N]: n # “no” (使用IMAP认证)

SquirrelMail Configuration : Read: config.php (1.4.0)

### ----- Server Settings

#### General

```

1. Domain : srv.world
2. Invert Time : false
3. Sendmail or SMTP : SMTP
```

#### SMTP Settings

```

4. SMTP Server : mail.srv.world
5. SMTP Port : 25
6. POP before SMTP : false
7. SMTP Authentication: login (with IMAP username and password)
```

## 9.5. WebMail

```
8. Secure SMTP (TLS) : false
9. Header encryption key :

A. Update IMAP Settings : mail.srv.world:143 (dovecot)
H. Hide SMTP Settings

R Return to Main Menu
C Turn color off
S Save data
Q Quit
```

Command >> r # 返回菜单

SquirrelMail Configuration : Read: config.php (1.4.0)

-----  
Main Menu --

1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color off

S Save data

Q Quit

Command >> 10 # 选择

SquirrelMail Configuration : Read: config.php (1.4.0)

-----  
Language preferences

1. Default Language : en\_US
2. Default Charset : iso-8859-1
3. Enable lossy encoding : false

R Return to Main Menu

C Turn color off

```
S Save data
Q Quit
```

Command >> 1 # 选择

SquirrelMail attempts to set the language in many ways. If it can not figure it out in another way, it will default to this language. Please use the code for the desired language.

[en\_US]: zh\_CN # 指定自己的语言（没有实际操作，中文可以自己看着选）

SquirrelMail Configuration : Read: config.php (1.4.0)

-----  
Language preferences

1. Default Language : zh\_CN
2. Default Charset : iso-8859-1
3. Enable lossy encoding : false

R Return to Main Menu

C Turn color off

S Save data

Q Quit

Command >> 2 # 选择

This option controls what character set is used when sending mail and when sending HTML to the browser.

This option is used only when default language is 'en\_US'.

[iso-8859-1]: iso-8859-1 # 指定自己的语言默认字符集（没有实际操作，中文可以自己看着选）

SquirrelMail Configuration : Read: config.php (1.4.0)

-----  
Language preferences

1. Default Language : zh\_CN
2. Default Charset : iso-8859-1
3. Enable lossy encoding : false

R Return to Main Menu

C Turn color off

S Save data

Q Quit

```
Command >> r # 返回菜单
SquirrelMail Configuration : Read: config.php (1.4.0)

Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color off
S Save data
Q Quit
```

```
Command >> 4 # 选择
SquirrelMail Configuration : Read: config.php (1.4.0)

General Options
1. Data Directory : /var/lib/squirrelmail/prefs/
2. Attachment Directory : /var/spool/squirrelmail/attac
h/
3. Directory Hash Level : 0
4. Default Left Size : 150
5. Usernames in Lowercase : false
6. Allow use of priority : true
7. Hide SM attributions : false
8. Allow use of receipts : true
9. Allow editing of identity : true
 Allow editing of name : true
 Remove username from header : false
10. Allow server thread sort : true
11. Allow server-side sorting : true
12. Allow server charset search : true
13. Enable UID support : true
```

```

14. PHP session name : SQMSESSID
15. Location base :
16. Only secure cookies if poss. : true
17. Disable secure forms : false
18. Page referal requirement :

```

R Return to Main Menu  
C Turn color off  
S Save data  
Q Quit

```

Command >> 7 # 选择“Hide SM attributions”
Hide SM attributions (y/n) [n]: y # “Yes”
SquirrelMail Configuration : Read: config.php (1.4.0)

```

#### General Options

```

1. Data Directory : /var/lib/squirrelmail/prefs/
2. Attachment Directory : /var/spool/squirrelmail/attac
h/
3. Directory Hash Level : 0
4. Default Left Size : 150
5. Usernames in Lowercase : false
6. Allow use of priority : true
7. Hide SM attributions : true
8. Allow use of receipts : true
9. Allow editing of identity : true
 Allow editing of name : true
 Remove username from header : false
10. Allow server thread sort : true
11. Allow server-side sorting : true
12. Allow server charset search : true
13. Enable UID support : true
14. PHP session name : SQMSESSID
15. Location base :
16. Only secure cookies if poss. : true
17. Disable secure forms : false
18. Page referal requirement :

```

R Return to Main Menu  
C Turn color off  
S Save data  
Q Quit

```
Command >> r # 返回菜单
SquirrelMail Configuration : Read: config.php (1.4.0)

Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color off
S Save data
Q Quit
```

```
Command >> 8 # 选择
SquirrelMail Configuration : Read: config.php (1.4.0)

Plugins
Installed Plugins
1. delete_move_next
2. squirrelspell
3. newmail

Available Plugins:
4. administrator
5. bug_report
6. calendar
7. compatibility
8. empty_trash
9. filters
10. fortune
11. info
12. listcommands
13. mail_fetch
```

```
14. message_details
15. secure_login
16. sent_subfolders
17. spamcop
18. test
19. translate
20. undelete
```

R Return to Main Menu  
C Turn color off  
S Save data  
Q Quit

Command >> 7 # 添加“compatibility”

SquirrelMail Configuration : Read: config.php (1.4.0)

---

### Plugins

#### Installed Plugins

```
1. delete_move_next
2. squirrelspell
3. newmail
4. compatibility
```

#### Available Plugins:

```
5. administrator
6. bug_report
7. calendar
8. empty_trash
9. filters
10. fortune
11. info
12. listcommands
13. mail_fetch
14. message_details
15. secure_login
16. sent_subfolders
17. spamcop
18. test
19. translate
20. undelete
```

R Return to Main Menu

```
C Turn color off
S Save data
Q Quit
```

```
Command >> 8 # 添加“empty_trash”
SquirrelMail Configuration : Read: config.php (1.4.0)
```

---

### Plugins

```
Installed Plugins
1. delete_move_next
2. squirrelspell
3. newmail
4. compatibility
5. empty_trash
```

### Available Plugins:

```
6. administrator
7. bug_report
8. calendar
9. filters
10. fortune
11. info
12. listcommands
13. mail_fetch
14. message_details
15. secure_login
16. sent_subfolders
17. spamcop
18. test
19. translate
20. undelete
```

```
R Return to Main Menu
C Turn color off
S Save data
Q Quit
```

```
Command >> 15 # 添加“secure_login”
SquirrelMail Configuration : Read: config.php (1.4.0)
```

---

### Plugins

```
Installed Plugins
```

```
1. delete_move_next
2. squirrelspell
3. newmail
4. compatibility
5. empty_trash
6. secure_login
```

Available Plugins:

```
7. administrator
8. bug_report
9. calendar
10. filters
11. fortune
12. info
13. listcommands
14. mail_fetch
15. message_details
16. sent_subfolders
17. spamcop
18. test
19. translate
20. undelete
```

R Return to Main Menu

C Turn color off

S Save data

Q Quit

Command >> q # 退出

You have not saved your data.

Save? [Y/n]: y # 保存

Data saved in config.php

Exiting conf.pl.

You might want to test your configuration by browsing to

<http://your-squirrelmail-location/src/configtest.php>

Happy SquirrelMailing!

```
cp /usr/share/squirrelmail/plugins/secure_login/config.sample.php
/usr/share/squirrelmail/plugins/secure_login/config.php
```

## 9.5. WebMail

编辑 `/usr/share/squirrelmail/plugins/secure_login/config.php` 文件：

```
更改 (登录后继续SSL连接)
$change_back_to_http_after_login = 0;
```

```
systemctl restart httpd
```

如果启用了SELinux，如下更改规则：

```
setsebool -P httpd_can_network_connect on
```

访问 `https://(服务器的主机名或IP地址)/webmail/`，登录表单如下所示。验证用户名和密码进行登录：



登录成功，尝试在这里发送或接收消息：



### 9.5.2. RainLoop

Rainloop 是非常现代化的WebMail平台，如果你在使用Gmail或者其他商业邮件客户端，RainLoop的界面非常适合你。

```
curl -O http://repository.rainloop.net/v2/webmail/rainloop-latest.zip

mkdir /var/www/html/rainloop

unzip rainloop-latest.zip -d /var/www/html/rainloop

find /var/www/html/rainloop -type d -exec chmod 755 {} \;

find /var/www/html/rainloop -type f -exec chmod 644 {} \;

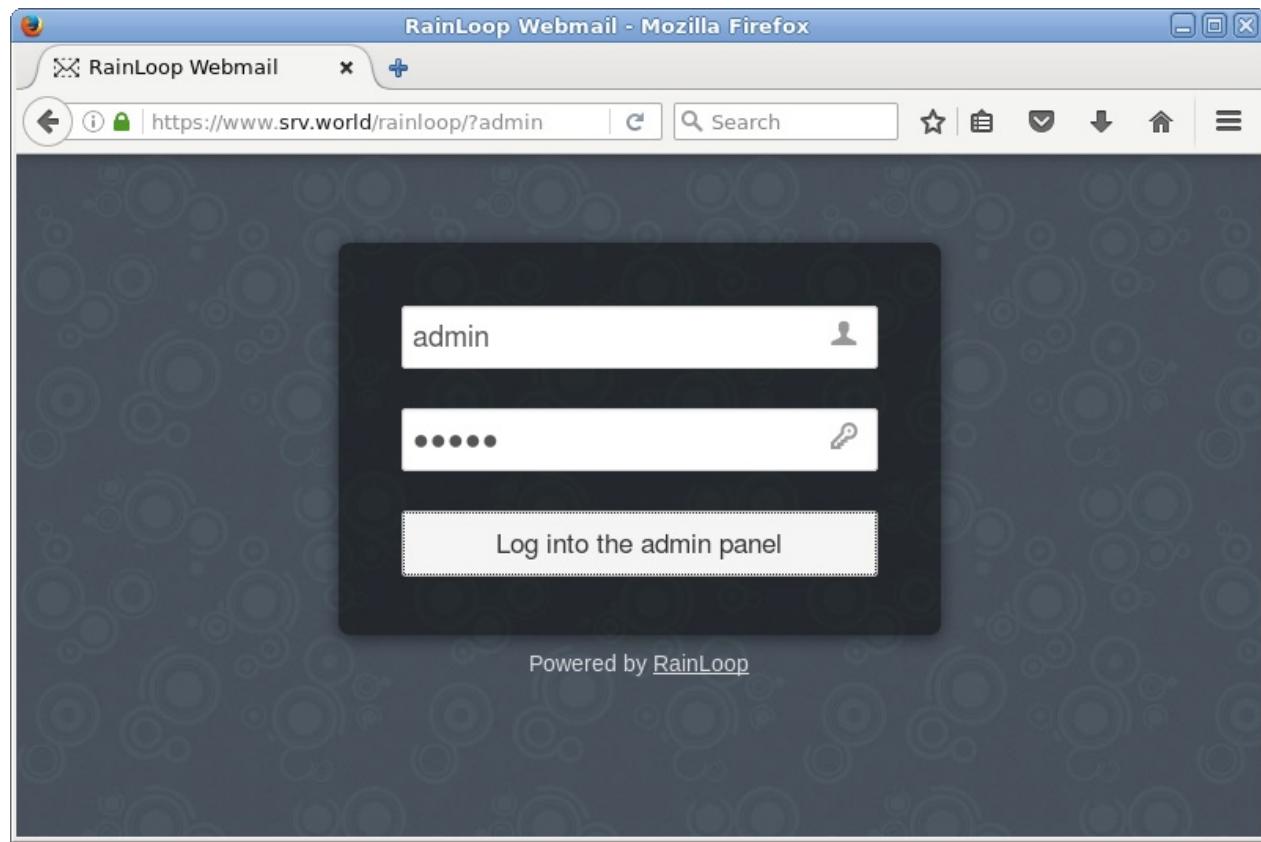
chown -R apache. /var/www/html/rainloop
```

如果启用了SELinux，如下更改规则：

```
chcon -R -t httpd_sys_rw_content_t /var/www/html/rainloop/data
semanage fcontext -a -t httpd_sys_rw_content_t /var/www/html/rainloop/data
```

## 9.5. WebMail

使用客户端的Web浏览器访问 `https://(服务器的主机名或IP地址)/rainloop/?admin`，然后使用用户“admin”和密码“12345”进行初始登录：



登录后，立即更改管理员初始密码。点击“change”：

The screenshot shows the RainLoop Admin Panel interface in Mozilla Firefox. The URL in the address bar is <https://www.srv.world/rainloop/?admin>. A prominent red warning box at the top right of the main content area says "Warning!" in bold red text. The message reads: "You are using the default admin password. For security reasons please [change](#) password to something else now." The word "change" is highlighted with a red box. On the left side, there is a sidebar with a dark background containing the following menu items: General, Login, Branding, Contacts, Domains, Security, Integrations, Plugins, Packages, Licensing, and About. The "General" item is currently selected and highlighted in white. The main content area has a light gray background. It contains a section titled "Interface" with settings for Language (English), Language (admin) (English), and Theme (Default). Below these are three checkboxes: "Allow language selection on settings screen" (checked), "Allow theme selection on settings screen" (checked), and "Allow background selection on settings screen" (unchecked).

输入旧密码和新密码，然后点击“Update Password”。更改后，注销一次并再次登录以确认：

The screenshot shows the RainLoop Admin Panel interface in Mozilla Firefox. The left sidebar contains navigation links: General, Login, Branding, Contacts, Domains, **Security**, Integrations, Plugins, Packages, Licensing, and About. The main content area has a header "RainLoop — Admin Panel (srv.world)".

**Security** section:

- Allow 2-Step Verification    Enforce 2-Step Verification
- Use local proxy for external images
- Allow OpenPGP

[Show PHP information](#)

**Admin Panel Access Credentials** section:

Form fields:

|                  |       |
|------------------|-------|
| Current password | ***** |
| New login        | admin |
| New password     | ***** |
| Repeat           | ***** |

更改显示语言。点击“English”：

The screenshot shows the RainLoop Admin Panel running in Mozilla Firefox. The left sidebar contains navigation links: General, Login, Branding, Contacts, Domains, Security, Integrations, Plugins, Packages, Licensing, and About. The main content area is titled "Interface". It includes settings for Language (English), Language (admin) (English), and Theme (Default). There are several configuration options with checkboxes: "Allow language selection on settings screen" (checked), "Allow theme selection on settings screen" (checked), "Allow background selection on settings screen" (unchecked), "Show thumbnails (attachments)" (checked), and "Allow Gravatar" (checked). Below this is a "Main" section with an "Attachment size limit" set to 25 MB. A note at the bottom states "PHP: upload\_max\_filesize = 2M; post\_max\_size = 8M" with an information icon.

在列表中选择自己的语言并关闭它：

The screenshot shows the RainLoop Admin Panel running in Mozilla Firefox. The title bar reads "RainLoop Webmail - Mozilla Firefox". The main content area is titled "RainLoop — Admin Panel (srv.world)". A modal dialog box is open, titled "Choose your language", listing various languages with their flags and names. The "English" option is selected, indicated by a green checkmark next to its name. Other languages listed include Arabic, Bulgarian, Czech, German, Greek, English (UK), French, Hungarian, Estonian, Icelandic, Japanese, Korean, Norwegian, Portuguese, Russian, Swedish, and Chinese (Simplified). Below the language list, there is a setting for "Attachment size limit" set to "25 MB". A message box at the bottom states "PHP: upload\_max\_filesize = 2M; post\_max\_size = 8M" with an information icon.

虽然管理面板没有更改，但用户界面已更改为自己的语言。接下来，选择左侧菜单中的“Domains”，然后单击“Add Domain”：

## 9.5. WebMail

The screenshot shows the RainLoop Admin Panel interface in Mozilla Firefox. The left sidebar has a dark theme with white text, listing various configuration sections: General, Login, Branding, Contacts, Domains (which is selected and highlighted in grey), Security, Integrations, Plugins, Packages, Licensing, and About. The main content area has a light background. It displays the 'Domains' section title and two buttons: '+ Add Domain' (highlighted with a red box) and '+ Add Alias'. Below these buttons is a descriptive text: 'List of domains webmail is allowed to access. Click on the name to configure the domain.' A table lists four domains: gmail.com, outlook.com, qq.com, and yahoo.com. Each domain entry includes a trash can icon and a checked checkbox.

|             |                                     |
|-------------|-------------------------------------|
| gmail.com   | <input checked="" type="checkbox"/> |
| outlook.com | <input type="checkbox"/>            |
| qq.com      | <input type="checkbox"/>            |
| yahoo.com   | <input type="checkbox"/>            |

输入邮件服务器的信息，如下所示，然后点击右下角的“Add”：

The screenshot shows the RainLoop Admin Panel interface in Mozilla Firefox. The title bar reads "RainLoop Webmail - Mozilla Firefox". The address bar shows the URL "https://www.srv.world/rainloop/?admin#/". The main content area is titled "RainLoop — Admin Panel (srv.world)" and displays the "Add Domain 'srv.world'" configuration page.

**Name (wildcard supported)**: srv.world

**IMAP**

|                |      |
|----------------|------|
| Server         | Port |
| mail.srv.world | 993  |

Secure: SSL/TLS

Use short login (user@domain.com → user)

**Sieve configuration (beta)**

**SMTP**

|                |      |
|----------------|------|
| Server         | Port |
| mail.srv.world | 465  |

Secure: SSL/TLS

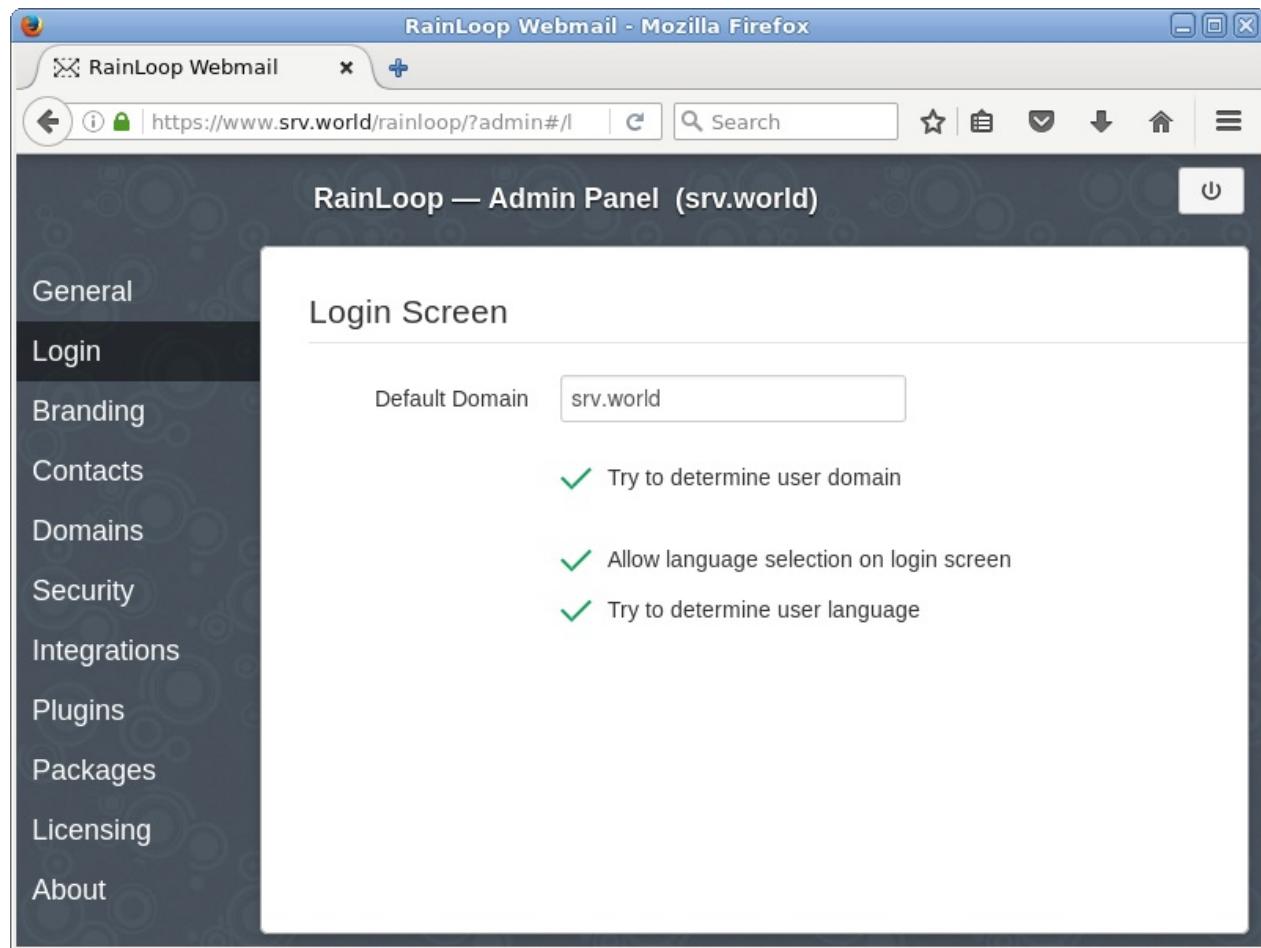
Use short login (user@domain.com → user)

Use authentication

Use php mail() function (beta)

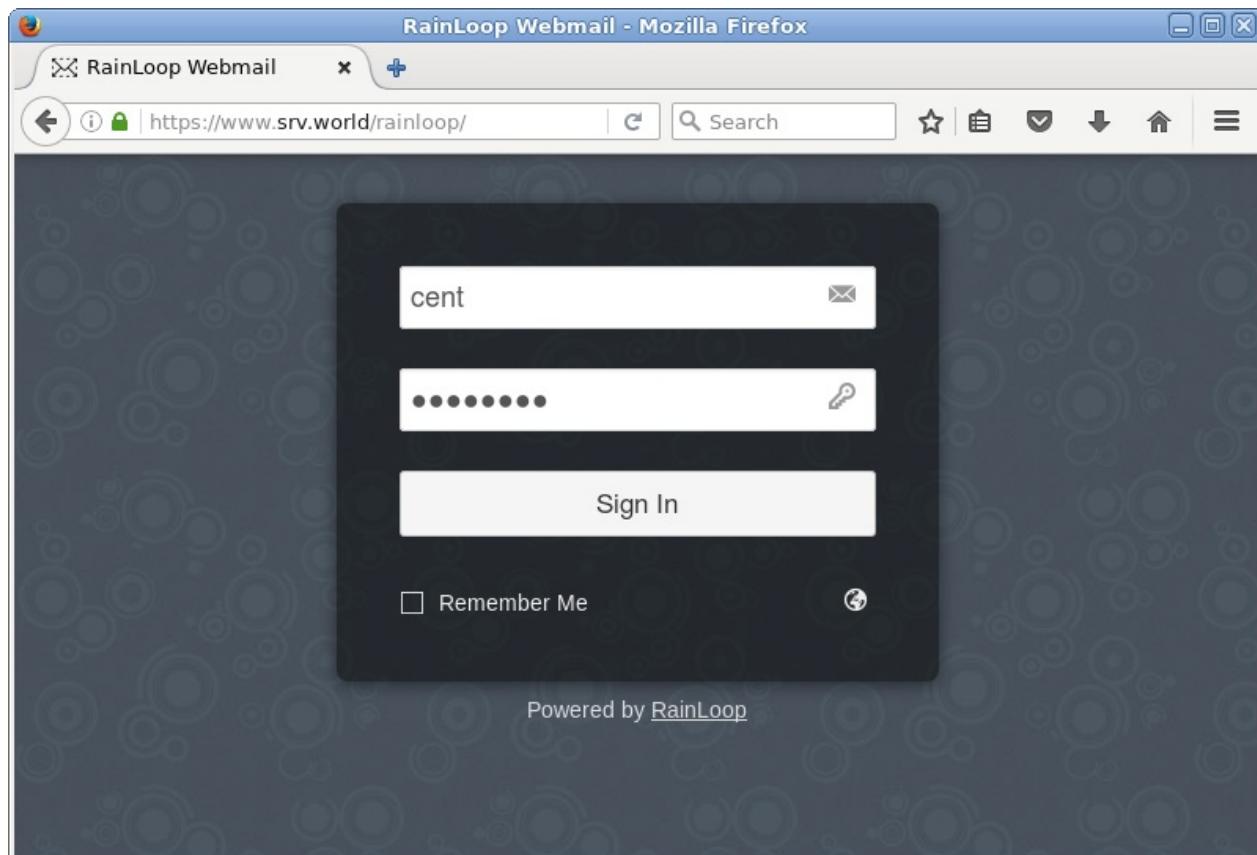
Buttons at the bottom: Test, White List, Close, Add

选择左侧菜单中的“Login”，并在“Default Domain”字段中输入名称，该字段是在上面的“Name”字段中设置的名称，并选中“Try to determine user domain”复选框。最小设置完成，从管理面板注销：

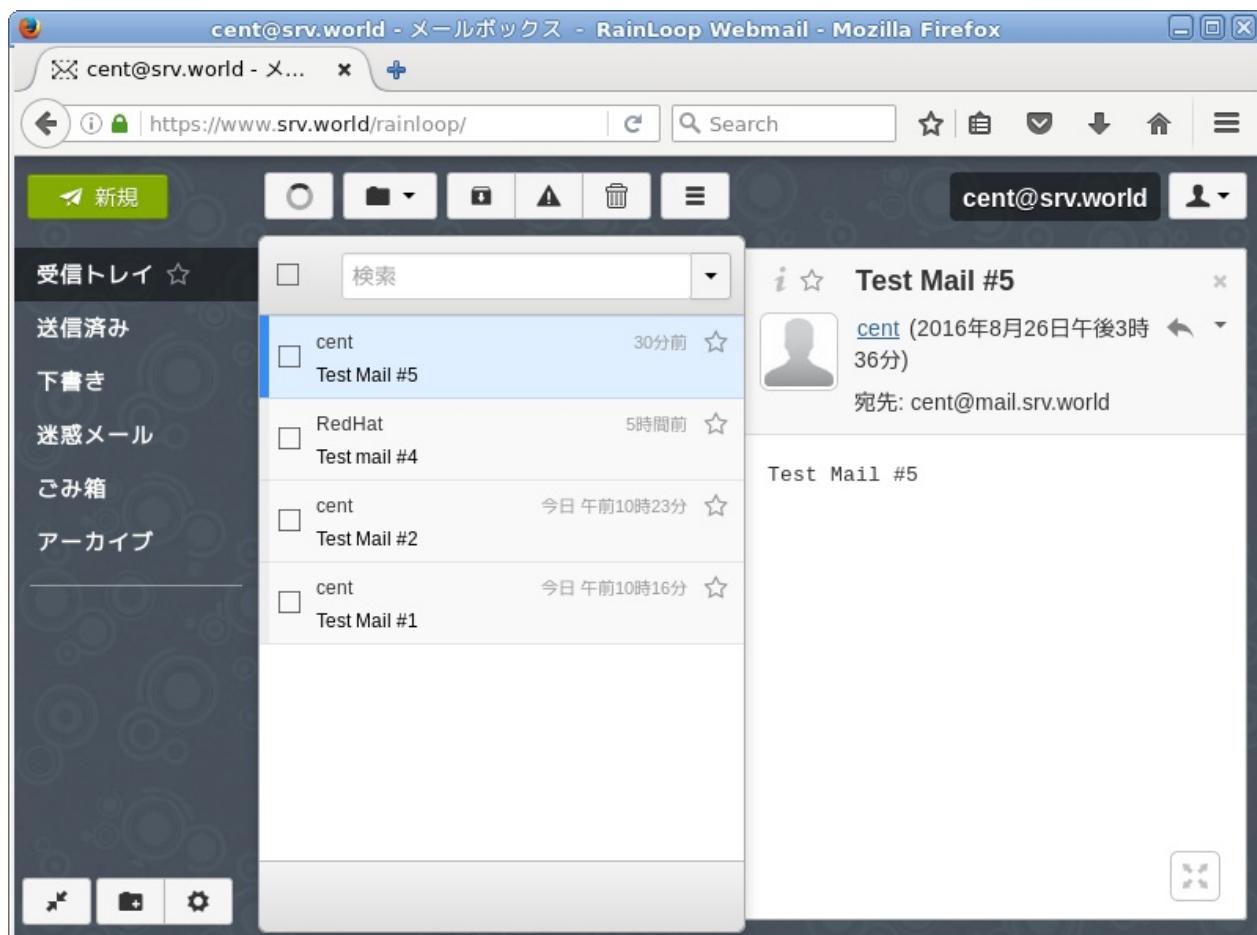


使用 RainLoop：访问 `https://(服务器的主机名或IP地址)/rainloop/`，并用邮件服务器中存在的用户登录：

## 9.5. WebMail



登录成功，下面是RainLoop的用户界面：



### 9.5.3. RoundCube

RoundCube功能包括MIME支持，地址薄，文件夹操作，信息搜索和拼写检查等。

还需安装[MariaDB数据库服务器](#)。

为RoundCube创建数据库：

```
mysql -u root -p
```

```
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 5.5.37-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, Monty Program Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

创建“roundcube”数据库（在“password”部分设置自己的密码）
MariaDB [(none)]> create database roundcube;
Query OK, 1 row affected (0.00 sec)
MariaDB [(none)]> grant all privileges on roundcube.* to roundcube@'localhost' identified by 'password';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> exit
Bye
```

```
yum --enablerepo=epel -y install roundcubemail # 从EPEL安装
```

```
cd /usr/share/roundcubemail/SQL
```

```
mysql -u roundcube -D roundcube -p < mysql.initial.sql
```

```
Enter password: # roundcube数据库密码
```

```
cd
```

```
cp -p /etc/roundcubemail/defaults.inc.php
/etc/roundcubemail/config.inc.php
```

编辑 `/etc/roundcubemail/config.inc.php` 文件：

```
如下更改（将密码替换为“password”）
$config['db_dsnw'] = 'mysql://roundcube:password@localhost/round
cube';

指定IMAP服务器（SSL）
$config['default_host'] = 'ssl://mail.srv.world';

指定IMAP端口（SSL）
$config['default_port'] = 993;

指定SMTP服务器（SSL）
$config['smtp_server'] = 'ssl://mail.srv.world';

指定SMTP端口（SSL）
$config['smtp_port'] = 465;

更改（使用相同的用户进行SMTP验证和IMAP验证）
$config['smtp_user'] = '%u';

更改（对SMTP验证和IMAP认证使用相同的密码）
$config['smtp_pass'] = '%p';

更改（SMTP认证类型）
$config['smtp_auth_type'] = 'LOGIN';

指定SMTP HELO主机
$config['smtp_helo_host'] = 'mail.srv.world';

指定自己的域名
$config['mail_domain'] = 'srv.world';

更改标题
$config['product_name'] = 'Server World Webmail';

更改UserAgent
$config['useragent'] = 'Server World Webmail';
```

```
更改自己的语言（没有实际操作，中文可以自己看着选）
$config['language'] = ja_JP;

将默认字符集更改自己的语言（没有实际操作，中文可以自己看着选）
$config['default_charset'] = 'iso-8859-1';
```

编辑 `/etc/httpd/conf.d/roundcubemail.conf` 文件：

```
允许访问的IP地址
Require ip 10.0.0.0/24
```

```
systemctl restart httpd
```

如果启用了SELinux，如下更改规则：

```
setsebool -P httpd_can_network_connect on
```

访问 `https://(服务器的主机名或IP地址)/roundcubemail/`，显示以下界面，验证用户和密码进行登录：



登录成功：

## 9.5. WebMail

SquirrelMail 1.4.22-15.el7 - Mozilla Firefox

SquirrelMail 1.4.22-15.el7 +

https://www.srv.world/webmail/src/webm | Search | ログアウト

現在のフォルダ: 受信箱 メッセージ作成 アドレス帳 フォルダ オプション 検索 ヘルプ

前リフレッシュ: 金, 6:08 am (メールをチェックする)

受信箱 Drafts Sent Trash

全反転 1 - 3 件目を表示中 (3 件中)

チェックしたもの: 移動 転送 チェックしたものの状態変更: 既読 未読 削除

スレッド表示

| 差出人                             | 日付      | 件名                           |
|---------------------------------|---------|------------------------------|
| <input type="checkbox"/> RedHat | 2:12 am | <a href="#">Test mail #4</a> |
| <input type="checkbox"/> cent   | 1:23 am | <a href="#">Test Mail #2</a> |
| <input type="checkbox"/> cent   | 1:16 am | <a href="#">Test Mail #1</a> |

全反転 1 - 3 件目を表示中 (3 件中)

The screenshot shows the SquirrelMail 1.4.22-15.el7 webmail interface running in Mozilla Firefox. The left sidebar lists folders: '受信箱' (Inbox), 'Drafts', 'Sent', and 'Trash'. The main area displays the '受信箱' (Inbox) with three messages listed. The first message is from 'RedHat' at 2:12 am with subject 'Test mail #4'. The second and third messages are from 'cent' at 1:23 am and 1:16 am respectively, with subjects 'Test Mail #2' and 'Test Mail #1'. There are buttons for '全反転' (Mark All as Read) and 'スレッド表示' (Threaded View). Navigation buttons for the inbox are visible at the bottom.

## 9.6. iRedMail

iRedMail是在操作系统安装好后使用的一套shell脚本，用于快速部署一套功能完善的邮件服务器解决方案。

CentOS的安装教程可以直接参考[官方文档](#)

# 10. 网络服务

- 10.1. DNS DHCP服务
  - 10.1.1. BIND
    - 10.1.1.1. 安装BIND
    - 10.1.1.2. 设置Zone
    - 10.1.1.3. 启动BIND
    - 10.1.1.4. 启用Chroot
    - 10.1.1.5. 设置CNAME
    - 10.1.1.6. 从DNS服务器
  - 10.1.2. DHCP服务器
    - 10.1.2.1. 配置DHCP服务器
    - 10.1.2.2. 配置DHCP客户端
  - 10.1.3. Dnsmasq
    - 10.1.3.1. 安装Dnsmasq
    - 10.1.3.2. 配置DHCP服务器
- 10.2. 代理服务器
  - 10.2.1. 安装Squid
  - 10.2.2. 配置客户端
  - 10.2.3. 基本认证
  - 10.2.4. 反向代理设置
  - 10.2.5. Squid + SquidClamav
  - 10.2.6. Squid + SquidGuard
- 10.3. 网络性能测试
  - 10.3.1. Iperf
- 10.4. PXE
  - 10.4.1. 配置PXE服务器
  - 10.4.2. 网络安装
  - 10.4.3. Kickstart安装
  - 10.4.4. 无盘客户机
- 10.5. OpenVPN
  - 10.5.1. 客户端使用证书认证连接
    - 10.5.1.1. 服务端
    - 10.5.1.2. 客户端

- 10.5.2. 客户端使用用户名密码连接
  - 10.5.2.1. 服务端
  - 10.5.2.2. 客户端
- 10.5.3. 执行脚本示例
- 10.5.4. Ubuntu 16.04安装OpenVPN
  - 10.5.4.1. 系统准备
  - 10.5.4.2. 安装服务端
  - 10.5.4.3. 客户端
- 10.6. PPTP
- 10.7. L2TP IPSec
  - 10.7.1. 服务端
  - 10.7.2. 客户端
  - 10.7.3. 调试
  - 10.7.4. 其他

## 10.1. DNS DHCP服务

### 10.1.1. BIND

#### 10.1.1.1. 安装BIND

安装[BIND](#)以配置解析域名或IP地址的DNS服务器。

```
yum -y install bind bind-utils
```

配置BIND。

本例演示使用公网IP地址 172.16.0.80/29，私有IP地址 10.0.0.0/24，域名 `srv.world` 设置。配置自己的服务器时，使用自己的IP地址和域名。（实际上，172.16.0.80/29 是用于私有IP地址）：

编辑 `/etc/named.conf` 文件：

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND na
med(8) DNS
// server as a caching only nameserver (as a localhost DNS resol
ver only).
//
// See /usr/share/doc/bind*/sample/ for example named configurat
ion files.

options {
 # 更改 (全部侦探)
 listen-on port 53 { any; };
 # 如果不使用IPv6则更改
 listen-on-v6 { none; };
 directory "/var/named";
 dump-file "/var/named/data/cache_dump.db";
 statistics-file "/var/named/data/named_stats.txt";
 memstatistics-file "/var/named/data/named_mem_stats.txt"
};
```

```

查询范围（设置内部服务器等）
allow-query { localhost; 10.0.0.0/24; };
传输范围（如果您有辅助DNS，则设置）
allow-transfer { localhost; 10.0.0.0/24; };

recursion yes;

dnssec-enable yes;
dnssec-validation yes;
dnssec-lookaside auto;

/* Path to ISC DLV key */
bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

pid-file "/run/named/named.pid";
session-keyfile "/run/named/session.key";
};

logging {
 channel default_debug {
 file "data/named.run";
 severity dynamic;
 };
};

从这里更改所有内容
view "internal" {
 match-clients {
 localhost;
 10.0.0.0/24;
 };
 zone "." IN {
 type hint;
 file "named.ca";
 };
 zone "srv.world" IN {
 type master;
 file "srv.world.lan";
 allow-update { none; };
 };
 zone "0.0.10.in-addr.arpa" IN {

```

```

 type master;
 file "0.0.10.db";
 allow-update { none; };

 };

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
};

view "external" {
 match-clients { any; };
 allow-query { any; };
 recursion no;
 zone "srv.world" IN {
 type master;
 file "srv.world.wan";
 allow-update { none; };
 };
 zone "80.0.16.172.in-addr.arpa" IN {
 type master;
 file "80.0.16.172.db";
 allow-update { none; };
 };
};

allow-query -> 允许的查询范围
allow-transfer -> 允许传输zone信息的范围
recursion -> 是否允许递归搜索
view "internal" { *** }; -> 写为内部定义
view "external" { *** }; -> 写为外部定义

对于如何写反向解析，如下写反向网络地址
10.0.0.0/24
network address -> 10.0.0.0
range of network -> 10.0.0.0 - 10.0.0.255
how to write -> 0.0.10.in-addr.arpa

172.16.0.80/29
network address -> 172.16.0.80
range of network -> 172.16.0.80 - 172.16.0.87
how to write -> 80.0.16.172.in-addr.arpa

```

## 10.1.1.2. 设置Zone

### 10.1.1.2.1. 配置名称解析

创建服务器从域名解析IP地址的zone文件。

对于内部zone，本例使用内部地址 10.0.0.0/24，域名 srv.world（在自己的环境替换它们）：

编辑 /var/named/srv.world.lan 文件：

```
$TTL 86400
@ IN SOA dlp.srv.world. root.srv.world. (
 2014071001 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
定义名称服务器
IN NS dlp.srv.world.
名称服务器的内部IP地址
IN A 10.0.0.30
定义邮件交换器
IN MX 10 dlp.srv.world.

定义IP地址和主机名
dlp IN A 10.0.0.30
```

对于外部zone，本例使用外部地址 172.16.0.80/29，域名 srv.world（在自己的环境替换它们）：

编辑 /var/named/srv.world.wan 文件：

```
$TTL 86400
@ IN SOA dlp.srv.world. root.srv.world. (
 2014071001 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
定义名称服务器
IN NS dlp.srv.world.
名称服务器的外部IP地址
IN A 172.16.0.82
定义邮件交换器
IN MX 10 dlp.srv.world.

定义IP地址和主机名
dlp IN A 172.16.0.82
```

### 10.1.1.2.2. 配置地址解析

创建服务器根据IP地址解析域名的zone文件

对于内部zone，本例使用内部地址 10.0.0.0/24，域名 srv.world（在自己的环境替换它们）：

编辑 /var/named/0.0.10.db 文件：

```
$TTL 86400
@ IN SOA dlp.srv.world. root.srv.world. (
 2014071001 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
定义名称服务器
IN NS dlp.srv.world.

定义此域的范围
IN PTR srv.world.
IN A 255.255.255.0

定义IP地址的主机名
30 IN PTR dlp.srv.world.
```

对于外部zone，本例使用外部地址 172.16.0.80/29，域名 srv.world（在自己的环境替换它们）：

编辑 /var/named/80.0.16.172.db 文件：

```
$TTL 86400
@ IN SOA dlp.srv.world. root.srv.world. (
 2014071001 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
定义名称服务器
IN NS dlp.srv.world.

定义此域的范围
IN PTR srv.world.
IN A 255.255.255.248

定义IP地址的主机名
82 IN PTR dlp.srv.world.
```

### 10.1.1.3. 启动BIND

```
systemctl start named
systemctl enable named
```

firewalld防火墙规则，允许DNS服务（53/TCP,UDP）：

```
firewall-cmd --add-service=dns --permanent
firewall-cmd --reload
```

将DNS服务器更改为自己的（设备名称替换为自己的环境）：

```
nmcli c modify eno16777736 ipv4.dns 10.0.0.30
nmcli c down eno16777736; nmcli c up eno16777736
```

验证名称或地址是否正常解析：

```
dig dlp.srv.world.
```

```
; <>> DiG 9.9.4-RedHat-9.9.4-14.el7 <>> dlp.srv.world.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41735
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;dlp.srv.world. IN A

;; ANSWER SECTION:
dlp.srv.world. 86400 IN A 10.0.0.30

;; AUTHORITY SECTION:
srv.world. 86400 IN NS dlp.srv.world.

;; Query time: 1 msec
;; SERVER: 10.0.0.30#53(10.0.0.30)
;; WHEN: Thu Jul 10 14:43:35 JST 2014
;; MSG SIZE rcvd: 75
```

```
dig -x 10.0.0.30
```

```

; <>> DiG 9.9.4-RedHat-9.9.4-14.el7 <>> -x 10.0.0.30
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14268
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;30.0.0.10.in-addr.arpa. IN PTR

;; ANSWER SECTION:
30.0.0.10.in-addr.arpa. 86400 IN PTR dlp.srv.world.

;; AUTHORITY SECTION:
0.0.10.in-addr.arpa. 86400 IN NS dlp.srv.world.

;; ADDITIONAL SECTION:
dlp.srv.world. 86400 IN A 10.0.0.30

;; Query time: 1 msec
;; SERVER: 10.0.0.30#53(10.0.0.30)
;; WHEN: Thu Jul 10 14:44:39 JST 2014
;; MSG SIZE rcvd: 111

```

#### 10.1.1.4. 启用Chroot

如果有需要，配置Chroot环境。

只需安装“bind-chroot”软件包即可。

如果在chroot环境中编辑 `named.conf` 或其他zone文件，请编辑`/var/named/chroot/`下的配置文件。

```
yum -y install bind-chroot
```

```
/usr/libexec/setup-named-chroot.sh /var/named/chroot on
```

```
systemctl stop named
systemctl disable named
systemctl start named-chroot
systemctl enable named-chroot
```

```
ll /var/named/chroot/etc
```

```
total 24
-rw-r--r-- 1 root root 331 Jul 10 14:46 localtime
drwxr-x--- 2 root named 6 Jun 10 17:13 named
-rw-r----- 1 root named 2211 Jul 10 14:13 named.conf
-rw-r--r-- 1 root named 2389 Jun 10 17:13 named.iscdlv.key
-rw-r----- 1 root named 931 Jun 21 2007 named.rfc1912.zones
-rw-r--r-- 1 root named 487 Jul 19 2010 named.root.key
drwxr-x--- 3 root named 24 Jul 10 14:46 pki
-rw-r----- 1 root named 77 Jul 10 14:39 rndc.key
```

```
ll /var/named/chroot/var/named
```

```
total 28
-rw-r--r-- 1 root root 358 Jul 10 14:31 0.0.10.db
drwxr-x--- 7 root named 56 Jul 10 14:46 chroot
drwxrwx--- 2 named named 22 Jul 10 14:39 data
drwxrwx--- 2 named named 4096 Jul 10 14:42 dynamic
-rw-r----- 1 root named 2076 Jan 28 2013 named.ca
-rw-r----- 1 root named 152 Dec 15 2009 named.empty
-rw-r----- 1 root named 152 Jun 21 2007 named.localhost
-rw-r----- 1 root named 168 Dec 15 2009 named.loopback
-rw-r--r-- 1 root root 350 Jul 10 14:30 srv.world.lan
drwxrwx--- 2 named named 6 Jun 10 17:13 slaves
```

### 10.1.1.5. 设置CNAME

如果要为主机设置其它名称，在zone文件中定义CNAME记录：

编辑 /var/named/srv.world.lan 文件：

```
$TTL 86400
@ IN SOA dlp.srv.world. root.srv.world. (
 # 更新序列
 2014071002 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
IN NS dlp.srv.world.
IN A 10.0.0.30
IN MX 10 dlp.srv.world.

dlp IN A 10.0.0.30
格式：别名 IN CNAME 服务器名称
ftp IN CNAME dlp.srv.world.
```

```
systemctl restart named
```

```
dig ftp.srv.world.
```

```
; <>> DiG 9.9.4-RedHat-9.9.4-14.el7 <>> ftp.srv.world.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64374
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ftp.srv.world. IN A

;; ANSWER SECTION:
ftp.srv.world. 86400 IN CNAME dlp.srv.world.
dlp.srv.world. 86400 IN A 10.0.0.30

;; AUTHORITY SECTION:
srv.world. 86400 IN NS dlp.srv.world.

;; Query time: 1 msec
;; SERVER: 10.0.0.30#53(10.0.0.30)
;; WHEN: Thu Jul 10 14:54:56 JST 2014
;; MSG SIZE rcvd: 93
```

### 10.1.1.6. 从DNS服务器

本例演示环境：主DNS为 `dlp.srv.world`，从DNS为 `ns.example.host`。

配置主DNS服务器：

编辑 `/etc/named.conf` 文件：

```
添加从DNS服务器的IP地址
allow-transfer { localhost; 172.16.0.85; };
```

编辑 `/var/named/srv.world.wan` 文件：

```
$TTL 86400
@ IN SOA dlp.srv.world. root.srv.world. (
 # 更新序列
 2014071003 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
IN NS dlp.srv.world.
添加从服务器
IN NS ns.example.host.
IN A 172.16.0.82
IN MX 10 dlp.srv.world.

dlp IN A 172.16.0.82
```

```
systemctl restart named
```

配置从DNS服务器：

编辑 /etc/named.conf 文件：

```
添加如下内容
zone "srv.world" IN {
 type slave;
 masters { 172.16.0.82; };
 file "slaves/srv.world.wan";
 notify no;
};
```

```
systemctl restart named
```

```
ls /var/named/slaves
```

```
srv.world.wan # 来自主服务器的zone文件已传输
```

## 10.1.2. DHCP服务器

### 10.1.2.1. 配置DHCP服务器

配置DHCP（Dynamic Host Configuration Protocol动态主机配置协议）服务器。

```
yum -y install dhcp
```

编辑 /etc/dhcp/dhcpd.conf 文件：

```
指定域名
option domain-name "srv.world";

指定名称服务器的主机名或IP地址
option domain-name-servers dlp.srv.world;

默认租约时间
default-lease-time 600;

最大租约时间
max-lease-time 7200;

此DHCP服务器被声明为有效
authoritative;

指定网络地址和子网掩码
subnet 10.0.0.0 netmask 255.255.255.0 {
 # 指定租约IP地址的范围
 range dynamic-bootp 10.0.0.200 10.0.0.254;
 # 指定广播地址
 option broadcast-address 10.0.0.255;
 # 指定默认网关
 option routers 10.0.0.1;
}
```

```
systemctl start dhcpcd
systemctl enable dhcpcd
```

firewalld防火墙规则，允许DHCP服务（67/UDP）：

```
firewall-cmd --add-service=dhcp --permanent
firewall-cmd --reload
```

### 10.1.2.2. 配置DHCP客户端

CentOS客户端，配置如下（将 `ifcfg-***` 部分替换为自己的接口名称）：

```
nmcli c modify ens3 ipv4.method auto
nmcli c down ens3; nmcli c up ens3
```

Windows的配置就不多说了。

### 10.1.3. Dnsmasq

#### 10.1.3.1. 安装Dnsmasq

安装[Dnsmasq](#)：这是轻量级DNS转发器和DHCP服务器软件。

```
yum -y install dnsmasq
```

编辑 `/etc/dnsmasq.conf` 文件：

```
取消注释（从不转发简单名称）
domain-needed

取消注释（从不转发非路由地址空间中的地址）
bogus-priv

取消注释（每个服务器严格按照resolv.conf的顺序查询）
strict-order

添加（到特定的DNS服务器查询指定的域名）
server=/server.education/10.0.0.10

取消注释（自动添加域名）
expand-hosts

添加（定义域名）
domain=srv.world
```

```
systemctl start dnsmasq
systemctl enable dnsmasq
```

对于DNS记录，将它们添加到 `/etc/hosts` 中，然后Dnsmasq将应答客户端的查询：

编辑 `/etc/hosts` 文件：

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost
4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost
6.localdomain6
add records
10.0.0.30 dlp.srv.world dlp
```

```
systemctl restart dnsmasq
```

firewalld防火墙规则，允许DNS服务：

```
firewall-cmd --add-service=dns --permanent
firewall-cmd --reload
```

从内部网络中的客户端验证解析名称或IP地址：

```
yum -y install bind-utils
```

将DNS设置更改为Dnsmasq服务器（将“ens3”替换为自己的环境）：

```
nmcli c modify ens3 ipv4.dns 10.0.0.30
nmcli c down ens3; nmcli c up ens3
```

```
dig dlp.srv.world.
```

```
; <>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.3 <>> dlp.srv.world
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11613
;; flags: qr aa rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;dlp.srv.world. IN A

;; ANSWER SECTION:
dlp.srv.world. 0 IN A 10.0.0.30

;; Query time: 2 msec
;; SERVER: 10.0.0.30#53(10.0.0.30)
;; WHEN: Wed Aug 31 17:22:51 JST 2016
;; MSG SIZE rcvd: 47
```

```
dig -x 10.0.0.30
```

```

; <>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.3 <>> -x 10.0.0.30
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61937
;; flags: qr aa rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;30.0.0.10.in-addr.arpa. IN PTR

;; ANSWER SECTION:
30.0.0.10.in-addr.arpa. 0 IN PTR dlp.srv.world.

;; Query time: 3 msec
;; SERVER: 10.0.0.30#53(10.0.0.30)
;; WHEN: Wed Aug 31 17:24:00 JST 2016
;; MSG SIZE rcvd: 67

```

### 10.1.3.2. 配置DHCP服务器

在Dnsmasq中启用并配置集成的DHCP功能。

编辑 `/etc/dnsmasq.conf` 文件：

```

添加 (IP地址租约和租约期) line 146: add (range of IP address to lease and term of lease)
dhcp-range=10.0.0.200,10.0.0.250,12h

添加 (定义默认网关)
dhcp-option=option:router,10.0.0.1

添加 (定义NTP, DNS, 服务器和子网掩码)
dhcp-option=option:ntp-server,10.0.0.10
dhcp-option=option:dns-server,10.0.0.10
dhcp-option=option:netmask,255.255.255.0

```

```
systemctl restart dnsmasq
```

`firewalld`防火墙规则，启用DHCP服务（67/UDP）：

```
firewall-cmd --add-service=dhcp --permanent
firewall-cmd --reload
```

设置完成，在客户端电脑[配置DHCP客户端](#)并验证是否正常工作。

## 10.2. 代理服务器

### 10.2.1. 安装Squid

安装[Squid](#)以配置代理服务器。

```
yum -y install squid
```

一般的转发代理设置：

编辑 `/etc/squid/squid.conf` 文件：

```
acl CONNECT method CONNECT
添加（定义新的ACL）
acl lan src 10.0.0.0/24

http_access allow localhost
添加（允许上面定义的ACL）
http_access allow lan

添加以下内容到最后
request_header_access Referer deny all
request_header_access X-Forwarded-For deny all
request_header_access Via deny all
request_header_access Cache-Control deny all

不显示IP地址
forwarded_for off
```

```
systemctl start squid
systemctl enable squid
```

`firewalld`防火墙规则，允许代理服务：

```
firewall-cmd --add-service=squid --permanent
firewall-cmd --reload
```

## 10.2.2. 配置客户端

配置代理客户端以连接到代理服务器。

CentOS客户端如下设置：

编辑 `/etc/profile` 文件：

```
添加以下内容到最后（将代理设置设为环境变量）
MY_PROXY_URL="http://prox.srv.world:3128/"

HTTP_PROXY=$MY_PROXY_URL
HTTPS_PROXY=$MY_PROXY_URL
FTP_PROXY=$MY_PROXY_URL
http_proxy=$MY_PROXY_URL
https_proxy=$MY_PROXY_URL
ftp_proxy=$MY_PROXY_URL

export HTTP_PROXY HTTPS_PROXY FTP_PROXY http_proxy https_proxy f
tp_proxy
```

```
source /etc/profile
```

设置完成。

也可以为每个应用程序设置代理设置，如下所示：

配置yum使用代理，编辑 `/etc/yum.conf` 文件：

```
添加到最后
proxy=http://prox.srv.world:3128/
```

配置wget使用代理，编辑 `/etc/wgetrc` 文件：

```
添加到最后
http_proxy = http://prox.srv.world:3128/
https_proxy = http://prox.srv.world:3128/
ftp_proxy = http://prox.srv.world:3128/
```

Windows客户端在IE选项中设置或具体的程序中设置。

### 10.2.3. 基本认证

设置基本身份验证并限制Squid用户需要身份验证。

```
yum -y install httpd-tools # 安装一个包含htpasswd的软件包
```

配置Squid以设置基本认证：

编辑 /etc/squid/squid.conf 文件：

```
acl CONNECT method CONNECT
为基本验证添加以下内容
auth_param basic program /usr/lib64/squid/basic_ncsa_auth /etc/squid/.htpasswd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 5 hours
acl password proxy_auth REQUIRED
http_access allow password
```

```
htpasswd -c /etc/squid/.htpasswd cent # 添加用户：使用-c创建一个新文件（添加-c选项仅用于初始注册）
```

```
New password: # 设置密码
Re-type new password: # 确认密码
Adding password for user cent
```

```
systemctl restart squid
```

配置CentOS代理客户端使用基本身份验证：

编辑 /etc/profile 文件：

```
添加以下内容到最后
格式：用户名:密码@代理服务器:端口
MY_PROXY_URL="http://cent:password@prox.srv.world:3128/"

HTTP_PROXY=$MY_PROXY_URL
HTTPS_PROXY=$MY_PROXY_URL
FTP_PROXY=$MY_PROXY_URL
http_proxy=$MY_PROXY_URL
https_proxy=$MY_PROXY_URL
ftp_proxy=$MY_PROXY_URL

export HTTP_PROXY HTTPS_PROXY FTP_PROXY http_proxy https_proxy ftp_proxy
```

```
source /etc/profile
```

设置完成。

也可以为每个应用程序设置代理设置，如下所示：

配置yum使用代理，编辑 `/etc/yum.conf` 文件：

```
添加到最后
proxy=http://prox.srv.world:3128/
proxy_username=cent
proxy_password=password
```

配置wget使用代理，编辑 `/etc/wgetrc` 文件：

```
添加到最后
http_proxy = http://prox.srv.world:3128/
https_proxy = http://prox.srv.world:3128/
ftp_proxy = http://prox.srv.world:3128/
proxy_user = cent
proxy_passwd = password
```

Windows客户端不需要特别设置，访问Web时，代理服务器弹出需要身份验证，然后输入用户名和密码。

## 10.2.4. 反向代理设置

将Squid配置为反向代理服务器。

编辑 `/etc/squid/squid.conf` 文件：

```
添加（允许所有http访问）
http_access allow all
And finally deny all other access to this proxy
http_access deny all

指定后端Web服务器
http_port 80 accel defaultsite=www.srv.world

取消注释
数值表示 -> [disk cache size磁盘缓存大小] [number of directories
on top level顶层目录数] [number of directories on 2nd level二级目录
数]
cache_dir ufs /var/spool/squid 100 16 256

添加到最后
cache_peer www.srv.world parent 80 0 no-query originserver

内存缓存大小
cache_mem 256 MB

定义主机名
visible_hostname prox.srv.world
```

```
systemctl start squid
systemctl enable squid
```

如果需要监听Squid上的HTTP访问，更改LAN中的DNS或路由器的设置，然后尝试从具有Web浏览器的客户端电脑访问Squid反向代理服务器，如下所示：



### 10.2.5. Squid + SquidClamav

安装SquidClamav并配置代理服务器扫描下载的文件以防病毒。

安装Clamav。

安装Clamav Scanner：

```
yum --enablerepo=epel -y install clamav-scanner clamav-scanner-
systemd # 从EPEL安装
```

编辑 /etc/clamd.d/scan.conf 文件：

```
注释
#Example

取消注释
LogFile /var/log/clamd.scan

取消注释
PidFile /var/run/clamd.scan/clamd.pid

取消注释
TemporaryDirectory /var/tmp

取消注释
LocalSocket /var/run/clamd.scan/clamd.sock

取消注释
TCPSocket 3310
```

## 10.2. 代理服务器

```
touch /var/log/clamd.scan
chown clamscan. /var/log/clamd.scan
```

```
systemctl start clamd@scan
systemctl enable clamd@scan
```

如果启用了SELinux，如下配置启动clamd：

```
restorecon -v /var/log/clamd.scan
```

安装c-icap：

```
yum -y install gcc make

curl -L -O http://downloads.sourceforge.net/project/c-icap/c-
icap/0.4.x/c_icap-0.4.2.tar.gz

tar zxvf c_icap-0.4.2.tar.gz

cd c_icap-0.4.2

.configure

make

make install

cd

cp /usr/local/etc/c-icap.conf /etc
```

编辑 /etc/c-icap.conf 文件：

```
更改管理地址
ServerAdmin root@srv.world

更改主机名
ServerName prox.srv.world

添加
Service squidclamav squidclamav.so
```

编辑 /etc/tmpfiles.d/c-icap.conf 文件：

```
d /var/run/c-icap 0755 root root -
```

编辑 `/usr/lib/systemd/system/c-icap.service` 文件：

```
[Unit]
Description=c-icap service
After=network.target

[Service]
Type=forking
PIDFile=/var/run/c-icap/c-icap.pid
ExecStart=/usr/local/bin/c-icap -f /etc/c-icap.conf
KillMode=process

[Install]
WantedBy=multi-user.target
```

安装SquidClamav ([下载最新版本](#))：

```
curl -L -O
http://downloads.sourceforge.net/project/squidclamav/squidclamav/6.1
4/squidclamav-6.14.tar.gz

tar zxvf squidclamav-6.14.tar.gz

cd squidclamav-6.14

./configure --with-c-icap

make

make install

cd

ln -s /usr/local/etc/squidclamav.conf /etc/squidclamav.conf
```

编辑 `/etc/squidclamav.conf` 文件：

```
更改（先创建重定向的目标网址）
redirect http://www.srv.world/error.html

更改（与clamd相同）
clamd_local /var/run/clamd.scan/clamd.sock
```

配置Squid：

编辑 `/etc/squid/squid.conf` 文件：

```
添加以下内容到最后
icap_enable on
icap_send_client_ip on
icap_send_client_username on
icap_client_username_header X-Authenticated-User
icap_service service_req reqmod_precache bypass=1 icap://127.0.0
.1:1344/squidclamav
adaptation_access service_req allow all
icap_service service_resp respmod_precache bypass=1 icap://127.0
.0.1:1344/squidclamav
adaptation_access service_resp allow all
```

```
systemctl start c-icap
systemctl enable c-icap
systemctl restart squid
```

配置完成。尝试从具有Web浏览器的客户端电脑访问[以下网站](http://www.srv.world/error.html?url=http://www.eicar.org/dc)，然后，单击测试病毒 `eicar.com`，以确保重定向到配置的网站。



## 10.2.6. Squid + SquidGuard

配置Squid + SquidGuard以设置内容过滤。

安装SquidGuard：

```
yum --enablerepo=epel -y install squidGuard # 从EPEL安装
```

```
mv /etc/squid/squidGuard.conf /etc/squid/squidGuard.conf.org
```

编辑 `/etc/squid/squidGuard.conf` 文件：

```
dbhome /var/lib/squidGuard/db
logdir /var/log/squidGuard

定义“deny”类别
dest deny {
 # 在“deny”类别中定义禁止域列表
 domainlist deny/domains
 # 在“deny”类别中定义禁止的网址列表
 urllist deny/urls
}

acl {
 default {
 # 允许除“deny”类别的所有
 pass !deny all
 # 如果匹配“deny”，重定向的网址
 redirect http://www.srv.world/error.html
 }
}
```

```
mkdir -p /var/lib/squidGuard/db/deny
```

编辑 `/var/lib/squidGuard/db/deny/domains` 文件：

```
写入要禁止访问的网域
yahoo.co.jp
example.com
```

编辑 `/var/lib/squidGuard/db/deny/urls` 文件：

```
写入要禁止访问的网址
www.yahoo.co.jp/deny/
www.example.com/
```

```
squidGuard -C all
chown -R squid. /var/lib/squidGuard/db/deny
```

编辑 `/etc/squid/squid.conf` 文件：

```
添加以下内容到最后
url_rewrite_program /usr/bin/squidGuard -c /etc/squid/squidGuard
.conf
```

```
systemctl restart squid
```

如果启用了SELinux，更改以下规则：

```
chcon -R -t squid_cache_t /var/lib/squidGuard
semanage fcontext -a -t squid_cache_t /var/lib/squidGuard
```

尝试访问上面设置为禁止的网址：



## 10.3. 网络性能测试

### 10.3.1. Iperf

Iperf是一个网络性能测试工具：可以测试TCP和UDP带宽质量，测量最大TCP带宽，具有多种参数和UDP特性，报告带宽，延迟抖动和数据包丢失等。

这需要两台机器，客户端（Sender）和服务器（Receiver）。

安装iperf3（在客户端和服务器上安装）：

```
yum --enablerepo=epel -y install iperf3 # 从EPEL安装
```

在服务器上执行命令，如下所示：

```
iperf3 -s
```

```

Server listening on 5201

```

检查网络带宽，在客户端执行以下命令：

```
iperf3 -c 10.0.0.30
```

```

Connecting to host 10.0.0.30, port 5201
[4] local 10.0.0.206 port 53096 connected to 10.0.0.30 port 52
01
[ID] Interval Transfer Bandwidth Retr Cwnd
[4] 0.00-1.00 sec 434 MBytes 3.64 Gbits/sec 58 407
 KBytes
[4] 1.00-2.00 sec 456 MBytes 3.83 Gbits/sec 0 452
 KBytes
[4] 2.00-3.00 sec 468 MBytes 3.93 Gbits/sec 0 481
 KBytes
[4] 3.00-4.00 sec 444 MBytes 3.73 Gbits/sec 52 370
 KBytes
[4] 4.00-5.00 sec 452 MBytes 3.79 Gbits/sec 0 395
 KBytes
[4] 5.00-6.00 sec 461 MBytes 3.87 Gbits/sec 0 406
 KBytes
[4] 6.00-7.00 sec 436 MBytes 3.66 Gbits/sec 50 313
 KBytes
[4] 7.00-8.00 sec 371 MBytes 3.11 Gbits/sec 52 245
 KBytes
[4] 8.00-9.00 sec 398 MBytes 3.34 Gbits/sec 0 260
 KBytes
[4] 9.00-10.00 sec 422 MBytes 3.54 Gbits/sec 0 269
 KBytes
- - - - -
[ID] Interval Transfer Bandwidth Retr
[4] 0.00-10.00 sec 4.24 GBytes 3.64 Gbits/sec 212
 sender
[4] 0.00-10.00 sec 4.24 GBytes 3.64 Gbits/sec
 receiver

iperf Done.

```

上例显示：传输了4.24 GB的数据，带宽为3.64千比特/秒。

还有很多选项可以尝试使用：

```
iperf3 --help
```

```

Usage: iperf [-s|-c host] [options]
 iperf [-h|--help] [-v|--version]

```

```

Server or Client:
 -p, --port # server port to listen on/connect to
 -f, --format [kmgKMG] format to report: Kbits, Mbits, KBytes,
 MBytes
 -i, --interval # seconds between periodic bandwidth reports
 -F, --file name xmit/recv the specified file
 -A, --affinity n/n,m set CPU affinity
 -B, --bind <host> bind to a specific interface
 -V, --verbose more detailed output
 -J, --json output in JSON format
 -d, --debug emit debugging output
 -v, --version show version information and quit
 -h, --help show this message and quit

Server specific:
 -s, --server run in server mode
 -D, --daemon run the server as a daemon
 -1, --one-off handle one client connection then exit

Client specific:
 -c, --client <host> run in client mode, connecting to <host>
 -u, --udp use UDP rather than TCP
 -b, --bandwidth #[KMG][/#] target bandwidth in bits/sec (0 for
 unlimited) (default 1 Mbit/sec for UDP, unlimited
 for TCP)
 (optional slash and packet count for
 burst mode)
 -t, --time # time in seconds to transmit for (default 10 secs)
 -n, --bytes #[KMG] number of bytes to transmit (instead
 of -t)
 -k, --blockcount #[KMG] number of blocks (packets) to transmit
 (instead of -t or -n)
 -l, --len #[KMG] length of buffer to read or write
 (default 128 KB for TCP, 8 KB for UDP)
 -P, --parallel # number of parallel client streams to run
 -R, --reverse run in reverse mode (server sends, c

```

```
lient receives)
 -w, --window #[KMG] set window size / socket buffer size
 -C, --linux-congestion <algo> set TCP congestion control algo
 rithm (Linux only)
 -M, --set-mss # set TCP maximum segment size (MTU -
 40 bytes)
 -N, --nodelay set TCP no delay, disabling Nagle's
 Algorithm
 -4, --version4 only use IPv4
 -6, --version6 only use IPv6
 -S, --tos N set the IP 'type of service'
 -L, --flowlabel N set the IPv6 flow label (only suppor
 ted on Linux)
 -Z, --zerocopy use a 'zero copy' method of sending
 data
 -O, --omit N omit the first n seconds
 -T, --title str prefix every output line with this s
 tring
 --get-server-output get results from server

[KMG] indicates options that support a K/M/G suffix for kilo-, m
ega-, or giga-
```

iperf3 homepage at: <http://software.es.net/iperf/>  
Report bugs to: <https://github.com/esnet/iperf>

## 10.4. PXE

PXE(Preboot eXecution Environment预启动执行环境)工作于Client/Server的网络模式，支持工作站通过网络从远端服务器下载镜像，并由此支持通过网络启动操作系统。计算机需要具有支持PXE的网卡。

### 10.4.1. 配置PXE服务器

安装所需的软件包：

```
yum -y install syslinux xinetd tftp-server
mkdir /var/lib/tftpboot/pxelinux.cfg
cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
```

启动TFTP：

```
/etc/xinetd.d/tftp
```

```
更改
disable = no
```

```
systemctl start xinetd
systemctl enable xinetd
```

先[参照这里设置好DHCP基本设置](#)，并添加以下设置：

编辑 `/etc/dhcp/dhcpd.conf` 文件：

```
option domain-name-servers 10.0.0.30;
为“next-server”指定PXE服务器的主机名或IP地址
filename "pxelinux.0";
next-server 10.0.0.30;
```

```
systemctl restart dhcpcd
```

## 10.4.2. 网络安装

通过网络从PXE服务器将操作系统安装到客户端计算机。如果客户端计算机没有CD/DVD驱动器，这很有用。

首先将ISO镜像下载到PXE服务器。下例演示位于 `/home/iso` 的CentOS7镜像。

```
mkdir -p /var/pxe/centos7
mkdir /var/lib/tftpboot/centos7
mount -t iso9660 -o loop /home/iso/CentOS-7-x86_64-DVD-1503-01.iso
/var/pxe/centos7
cp /var/pxe/centos7/images/pxeboot/vmlinuz
/var/lib/tftpboot/centos7/
cp /var/pxe/centos7/images/pxeboot/initrd.img
/var/lib/tftpboot/centos7/
cp /usr/share/syslinux/menu.c32 /var/lib/tftpboot/
```

编辑 `/var/lib/tftpboot/pxelinux.cfg/default` 文件：

```
timeout 100
default menu.c32

menu title ##### PXE Boot Menu #####
label 1
 menu label ^1) Install CentOS 7
 kernel centos7/vmlinuz
 append initrd=centos7/initrd.img method=http://10.0.0.30/cent
os7 devfs=nomount

label 2
 menu label ^2) Boot from local drive
 localboot
```

先[参照这里安装好HTTP服务器](#)，并添加以下设置：

编辑 `/etc/httpd/conf.d/pxeboot.conf` 文件：

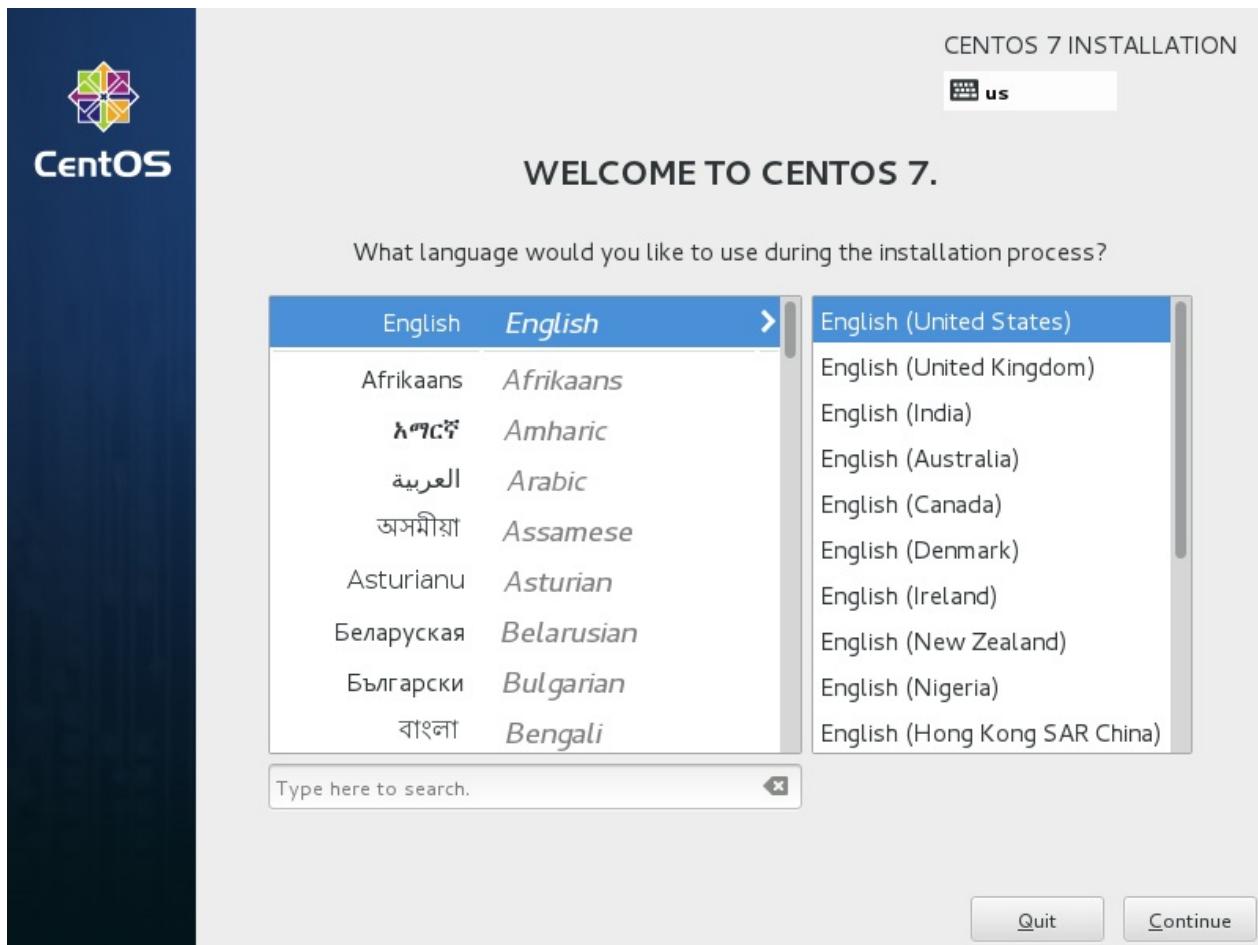
```
Alias /centos7 /var/pxe/centos7
<Directory /var/pxe/centos7>
 Options Indexes FollowSymLinks
 # 允许访问的IP地址
 Require ip 127.0.0.1 10.0.0.0/24
</Directory>
```

```
systemctl restart httpd
```

在客户端计算机的BIOS设置上启用网络引导并启动，然后显示安装菜单，按Enter键继续安装：



安装程序启动，这之后的安装与普通安装相同：



### 10.4.3. Kickstart安装

可以使用Kickstart自动执行安装工作。

先按上节内容配置好基本的网络安装（Network-Install）设置。

配置Kickstart：

生成加密的root密码（记住它）：

```
python -c 'import crypt,getpass; \
print(crypt.crypt(getpass.getpass(), \
crypt.mksalt(crypt.METHOD_SHA512)))'
```

Password:

\$6\$EC1T.oKN5f3seb20\$y1w1MQ7Ih4240wOn.....

```
mkdir /var/www/html/ks
```

## 10.4. PXE

---

编辑 `/var/www/html/ks/centos7-ks.cfg` 文件：

```

install
自动进行每一步骤
autostep
安装后重启
reboot
加密算法
auth --enableshadow --passalgo=sha512
安装源
url --url=http://10.0.0.30/centos7/
安装磁盘
ignoredisk --only-use=sda
键盘布局
keyboard --vckeymap=cn --xlayouts='cn','us'
系统语言设置
lang zh_CN.UTF-8
网络设置
network --bootproto=dhcp --ipv6=auto --activate --hostname=local
host
上面生成的root密码
rootpw --iscrypted 6EC1T.oKN5f3seb20$y1WlMQ7Ih4240wOn.....
时区
timezone Asia/Shanghai --isUtc --nontp
引导程序（bootloader）设置
bootloader --location=mbr --boot-drive=sda
初始化所有分区表
zerombr
clearpart --all --initlabel
分区
part /boot --fstype="xfs" --ondisk=sda --size=500
part pv.10 --fstype="lvmpv" --ondisk=sda --size=51200
volgroup VolGroup --pesize=4096 pv.10
logvol / --fstype="xfs" --size=20480 --name=root --vgname=VolGro
up
logvol swap --fstype="swap" --size=4096 --name=swap --vgname=Vol
Group

%packages
@core

%end

```

## 10.4. PXE

```
chmod 644 /var/www/html/ks/centos7-ks.cfg
```

编辑 `/var/lib/tftpboot/pxelinux.cfg/default` 文件：

```
timeout 100
default menu.c32

menu title ##### PXE Boot Menu #####
label 1
 menu label ^1) Install CentOS 7
 kernel centos7/vmlinuz
 # 更改：指定Kickstart文件
 append initrd=centos7/initrd.img ks=http://10.0.0.30/ks/cento
s7-ks.cfg

label 2
 menu label ^2) Boot from local drive
 localboot
```

在客户端计算机的BIOS设置上启用网络引导并启动，然后显示安装菜单，10秒钟后，安装过程将自动开始，完成并重新启动：



### 10.4.4. 无盘客户机

从PXE服务器启动没有本地硬盘的客户端计算机。

## 10.4. PXE

先按第二节内容配置好基本的网络安装设置。

安装所需的软件包：

```
yum -y install dracut-network nfs-utils
```

在PXE服务器上为无盘客户机构建系统：

```
mkdir -p /var/lib/tftpboot/centos7/root
```

```
yum groups -y install "Server with GUI" --releasever=7 --
installroot=/var/lib/tftpboot/centos7/root/
```

生成加密的root密码（记住它）：

```
python -c 'import crypt,getpass; \
print(crypt.crypt(getpass.getpass(), \
crypt.mksalt(crypt.METHOD_SHA512)))'
```

Password:

```
6EC1T.oKN5f3seb20$y1WlMQ7Ih4240wOn.....
```

编辑 `/var/lib/tftpboot/centos7/root/etc/shadow` 文件：

```
设置上面生成的root密码
root:6EC1T.oKN5f3seb20$y1WlMQ7Ih4240wOn.....:16372:0:99999:7::
:
```

编辑 `/var/lib/tftpboot/centos7/root/etc/fstab` 文件：

```
none /tmp tmpfs defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

```
wget -P /var/lib/tftpboot/centos7/ \
http://mirror.centos.org/centos/7/os/x86_64/images/pxeboot/vmlinuz \
http://mirror.centos.org/centos/7/os/x86_64/images/pxeboot/initrd.img
```

编辑 `/var/lib/tftpboot/pxelinux.cfg/default` 文件：

```
default centos7

label centos7
 kernel centos7/vmlinuz
 append initrd=centos7/initrd.img root=nfs:10.0.0.30:/var/lib/tftpboot/centos7/root rw selinux=0
```

配置NFS服务器与客户端共享系统文件

编辑 `/etc/exports` 文件：

```
/var/lib/tftpboot/centos7/root 10.0.0.0/24(rw,no_root_squash)
```

```
systemctl start rpcbind nfs-server
systemctl enable rpcbind nfs-server
```

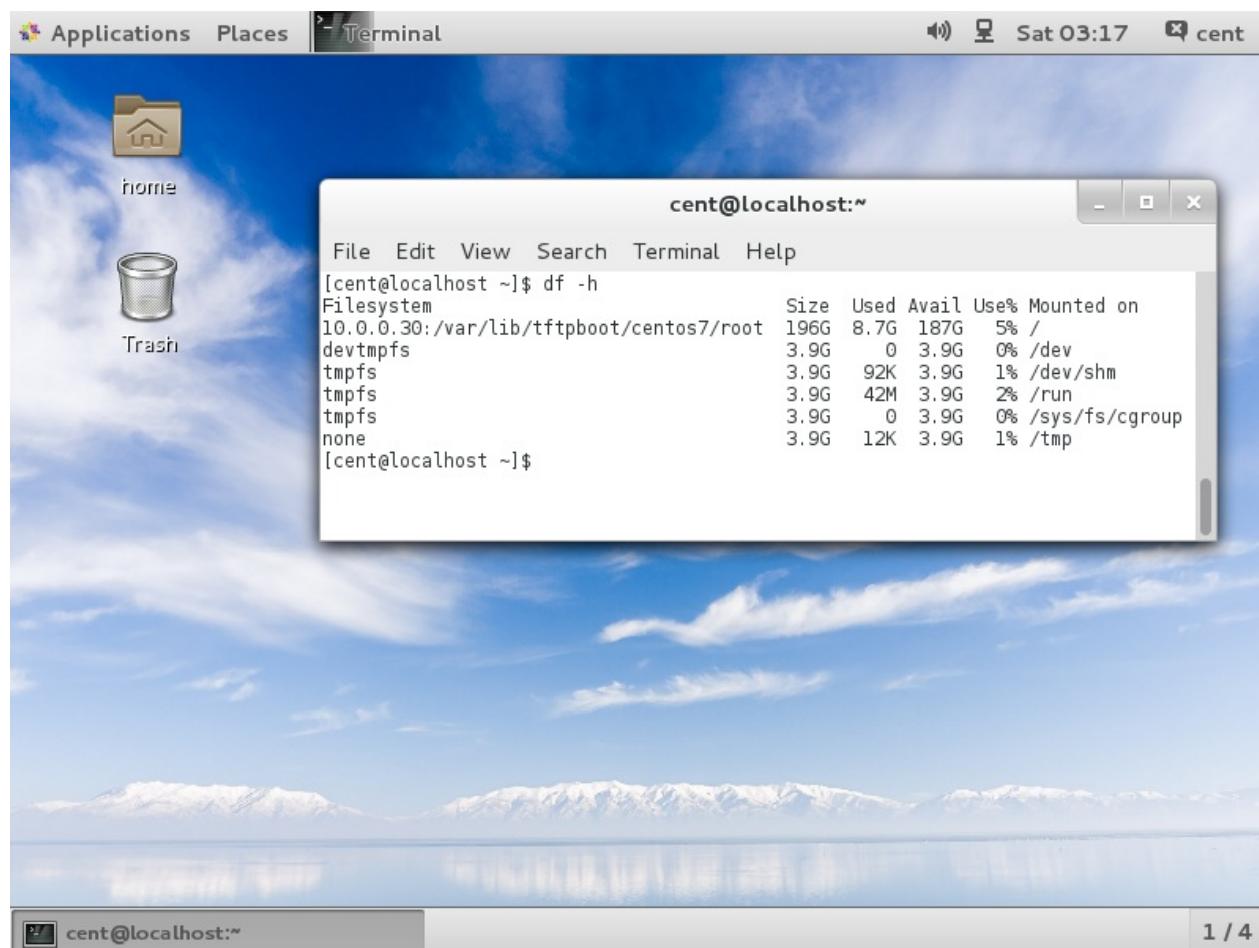
配置完成，启动一个BIOS设置启用了网络引导的无盘计算机，系统将启动如下。

## 10.4. PXE

```
Network boot from Intel E1000
Copyright (C) 2003-2014 VMware, Inc.
Copyright (C) 1997-2000 Intel Corporation

CLIENT MAC ADDR: 00 0C 29 96 72 76 GUID: 564D7D04-8BE2-FF29-9793-A8297C967276
CLIENT IP: 10.0.0.200 MASK: 255.255.255.0 DHCP IP: 10.0.0.30
GATEWAY IP: 10.0.0.1

PXELINUX 4.05 0x54f93f16 Copyright (C) 1994-2011 H. Peter Anvin et al
!PXE entry point found (we hope) at 9E0E:0106 via plan A
UNDI code segment at 9E0E len 0BCE
UNDI data segment at 9878 len 5960
Getting cached packet 01 02 03
My IP address seems to be 0A0000C8 10.0.0.200
ip=10.0.0.200:10.0.0.30:10.0.0.1:255.255.255.0
BOOTIF=01-00-0c-29-96-72-76
SYSUUID=564d7d04-8be2-ff29-9793-a8297c967276
TFTP prefix:
Trying to load: pxelinux.cfg/default ok
Loading centos7/vmlinuz.....
Loading centos7/initrd.img.....
```

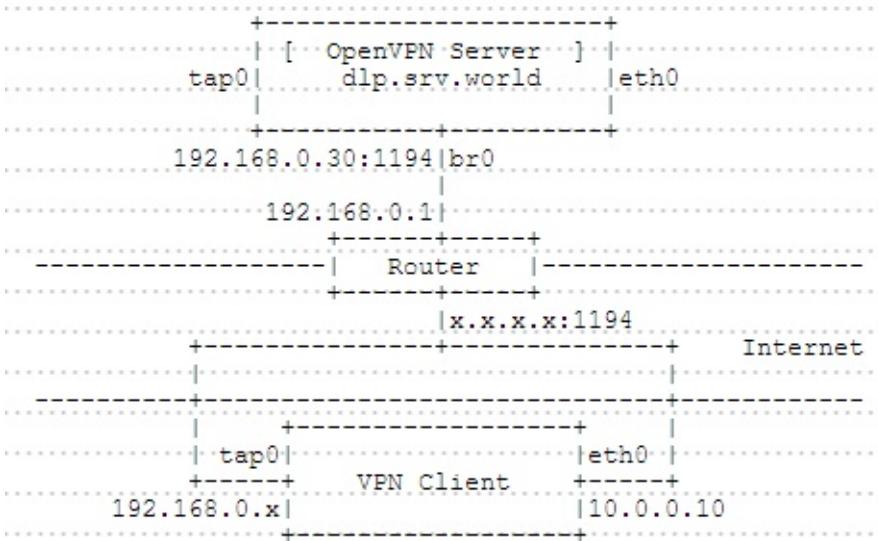


## 10.5. OpenVPN

[OpenVPN](#) (OpenVPN官网所有内容需科学上网才能访问) 是一个功能齐全的SSL VPN，它使用行业标准的SSL/TLS协议实现了OSI模型第2层（数据链路层）或第3层（网络层）的安全网络扩展。OpenVPN支持基于证书、智能卡以及用户名/密码等多种形式的灵活的客户端认证方法，并可以通过应用于VPN虚拟接口的防火墙规则为指定用户或用户组设置访问控制策略。

### 10.5.1. 客户端使用证书认证连接

本例基于以下环境：



使用桥接模式配置OpenVPN，OpenVPN服务器的“br0”和“tap0”由服务自动生成，客户端上“tap0”的IP地址由OpenVPN服务器分配。连接VPN后，客户端可以访问（与服务器）同一本地网络上的任何计算机。

配置之前，必须在网关路由器上配置IP伪装。对于本例，与 `x.x.x.x:1194` 的连接将转发到 `192.168.0.30:1194`。

#### 10.5.1.1. 服务端

安装OpenVPN：

```
yum --enablerepo=epel -y install openvpn easy-rsa net-tools bridge-utils
```

## 10.5. OpenVPN

创建CA证书：

```
cd /usr/share/easy-rsa/2.0
```

编辑 vars 文件：

```
根据自己需要更改
export KEY_COUNTRY="CN"
export KEY_PROVINCE="SC"
export KEY_CITY="CD"
export KEY_ORG="GTS"
export KEY_EMAIL="root@dlp.srv.world"
export KEY_OU="Server_World"
```

```
source ./vars
```

NOTE: If you run ./clean-all, I will be doing a rm -rf on /usr/share/easy-rsa/2.0/keys

```
./clean-all
```

```
./build-ca
```

```
Generating a 2048 bit RSA private key
.....+++
...+++
writing new private key to 'ca.key'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [CN]: # 回车
State or Province Name (full name) [SC]: # 回车
Locality Name (eg, city) [CD]: # 回车
Organization Name (eg, company) [GTS]: # 回车
Organizational Unit Name (eg, section) [Server_World]: # 回车
Common Name (eg, your name or your server's hostname) [GTS CA]:
回车
Name [EasyRSA]:Server-CA # 设置任意名称
Email Address [root@dlp.srv.world]: # 回车
```

创建服务器证书：

```
cd /usr/share/easy-rsa/2.0
./build-key-server server
```

```
Generating a 2048 bit RSA private key
.....+++
...+++
writing new private key to 'server.key'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [CN]: # 回车
State or Province Name (full name) [SC]: # 回车
Locality Name (eg, city) [CD]: # 回车
Organization Name (eg, company) [GTS]: # 回车
Organizational Unit Name (eg, section) [Server_World]: # 回车
Common Name (eg, your name or your server's hostname) [server]:
回车
Name [EasyRSA]:Server-CRT # 设置任意名称
Email Address [root@dlp.srv.world]: # 回车

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/share/easy-rsa/2.0/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'CN'
stateOrProvinceName :PRINTABLE:'SC'
localityName :PRINTABLE:'CD'
organizationName :PRINTABLE:'GTS'
organizationalUnitName:T61STRING:'Server_World'
commonName :PRINTABLE:'server'
name :PRINTABLE:'Server-CRT'
emailAddress :IA5STRING:'root@dlp.srv.world'
Certificate is to be certified until Jun 23 05:59:34 2025 GMT (3
650 days)
Sign the certificate? [y/n]: y # 确认设置并输入y继续执行
proceed with yes
1 out of 1 certificate requests certified, commit? [y/n] y # 输入y继续执行
Write out database with 1 new entries
Data Base Updated
```

生成Diffie Hellman (DH) 参数：

## 10.5. OpenVPN

```
cd /usr/share/easy-rsa/2.0
```

```
./build-dh
```

```
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
```

创建客户端证书：

```
cd /usr/share/easy-rsa/2.0
```

```
./build-key client01
```

```
Generating a 2048 bit RSA private key
.....+++
.....+ ++
writing new private key to 'client01.key'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [CN]: # 回车
State or Province Name (full name) [SC]: # 回车
Locality Name (eg, city) [CD]: # 回车
Organization Name (eg, company) [GTS]: # 回车
Organizational Unit Name (eg, section) [Server_World]: # 回车
Common Name (eg, your name or your server's hostname) [client01]
: # 回车
Name [EasyRSA]:client01 # 设置任意名称
Email Address [root@dlp.srv.world]: # 回车

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/share/easy-rsa/2.0/openssl-1.0.0.c
```

```
nf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'CN'
stateOrProvinceName :PRINTABLE:'SC'
localityName :PRINTABLE:'CD'
organizationName :PRINTABLE:'GTS'
organizationalUnitName:T61STRING:'Server_World'
commonName :PRINTABLE:'client01'
name :PRINTABLE:'client01'
emailAddress :IA5STRING:'root@dlp.srv.world'
Certificate is to be certified until Jun 23 06:01:37 2025 GMT (3
650 days)
Sign the certificate? [y/n]: y # 确认设置并输入y继续执行
proceed with yes
1 out of 1 certificate requests certified, commit? [y/n] y # 输入y继续执行
Write out database with 1 new entries
Data Base Updated
```

配置并启动OpenVPN服务器：

```
cp -pR /usr/share/easy-rsa/2.0/keys /etc/openvpn/keys
cp /usr/share/doc/openvpn-*/*sample/sample-config-files/server.conf
/etc/openvpn/
```

编辑 /etc/openvpn/server.conf 文件：

```

根据需要修改（侦听端口）
port 1194

取消注释“tcp”并注释掉“udp”
proto tcp
;proto udp

更改为tap（使用桥接模式）
dev tap0
;dev tun

更改证书路径
ca keys/ca.crt
cert keys/server.crt
key keys/server.key
dh keys/dh2048.pem

注释
;server 10.8.0.0 255.255.255.0

取消注释并更改：[VPN服务器IP地址] [子网掩码] [客户端的IP范围]
server-bridge 192.168.0.30 255.255.255.0 192.168.0.150 192.168.0
.199

keepalive设置
keepalive 10 120

启用压缩（好像在2.4中已弃用，并在更高版本中被删除）
;comp-lzo

启用persist选项
persist-key
persist-tun

取消注释并指定日志
log /var/log/openvpn.log
log-append /var/log/openvpn.log

指定日志级别（0 - 9，9表示调试级别）
verb 3

```

## 10.5. OpenVPN

```
cp /usr/share/doc/openvpn-*/sample/sample-scripts/bridge-start
/etc/openvpn/openvpn-startup
```

```
cp /usr/share/doc/openvpn-*/sample/sample-scripts/bridge-stop
/etc/openvpn/openvpn-shutdown
```

```
chmod 755 /etc/openvpn/openvpn-startup /etc/openvpn/openvpn-
shutdown
```

编辑 /etc/openvpn/openvpn-startup 文件：

```
更改
eth="eth0" # 根据需要更改
eth_ip="192.168.0.30" # 网桥接口IP
eth_netmask="255.255.255.0" # 子网掩码
eth_broadcast="192.168.0.255" # 广播地址

添加以内容到最后 (定义网关)
eth_gw="192.168.0.1"
route add default gw $eth_gw
```

```
cp /usr/lib/systemd/system/openvpn@.service
/usr/lib/systemd/system/openvpn-bridge.service
```

编辑 /usr/lib/systemd/system/openvpn-bridge.service 文件：

```
在 [Service] 部分如下更改
[Service]
PrivateTmp=true
Type=forking
PIDFile=/var/run/openvpn/openvpn.pid
ExecStartPre=/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
ExecStartPre=/etc/openvpn/openvpn-startup
ExecStart=/usr/sbin/openvpn --daemon --writepid /var/run/openvpn
/openvpn.pid --cd /etc/openvpn/ --config server.conf
ExecStopPost=/etc/openvpn/openvpn-shutdown
ExecStopPost=/bin/echo 0 > /proc/sys/net/ipv4/ip_forward
```

```
systemctl start openvpn-bridge
```

```
[1367.964300] device tap0 entered promiscuous mode
[1367.967487] IPv6: ADDRCONF(NETDEV_UP): tap0: link is not ready
[1367.971388] br0: port 1(eth0) entered forwarding state
[1367.972534] br0: port 1(eth0) entered forwarding state
[1368.006320] IPv6: ADDRCONF(NETDEV_CHANGE): tap0: link becomes ready
[1368.007546] br0: port 2(tap0) entered forwarding state
[1368.008452] br0: port 2(tap0) entered forwarding state
```

```
systemctl enable openvpn-bridge
```

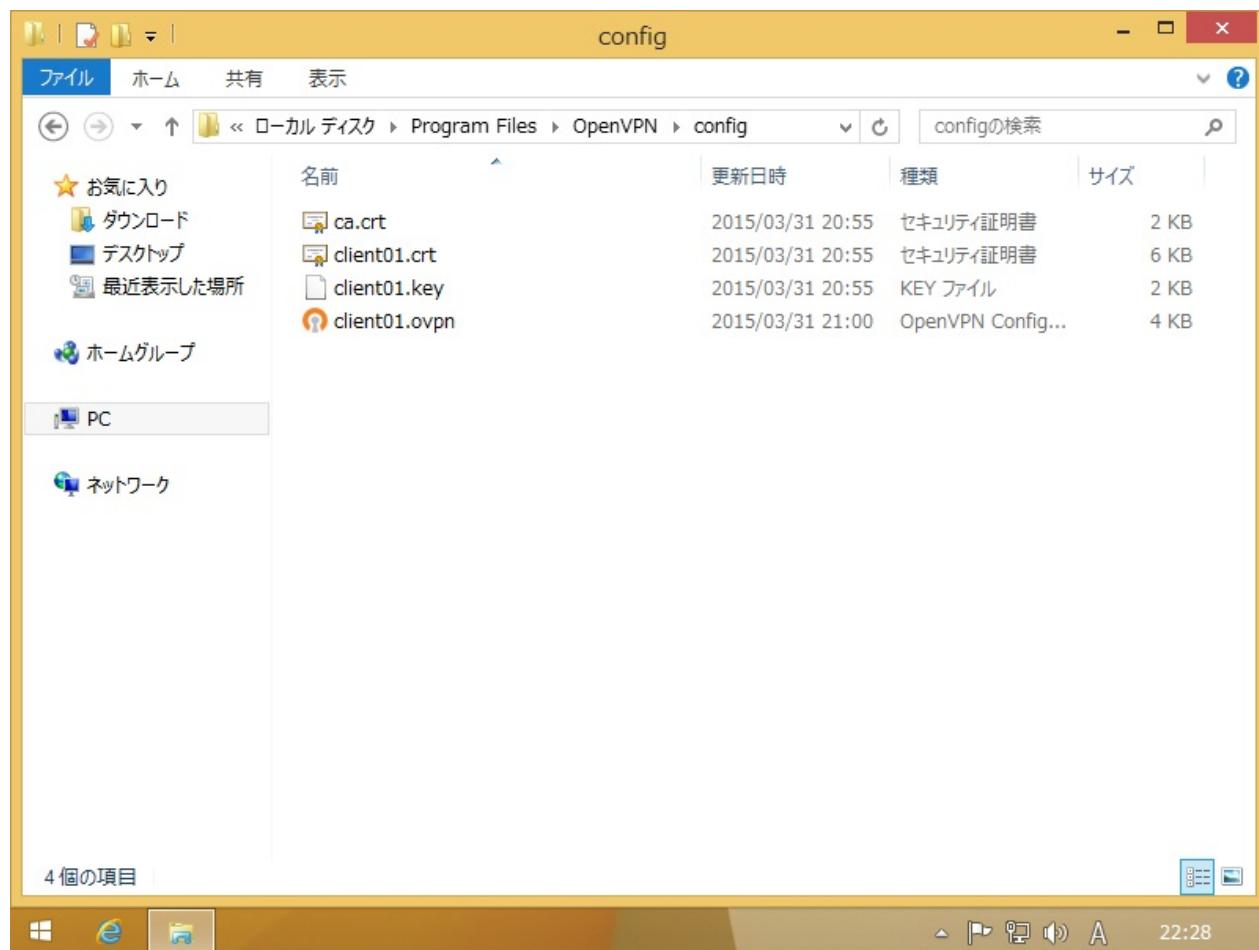
将 /etc/openvpn/keys 下的文件“ca.crt”，“client01.crt”，“client01.key”传输到客户端计算机以连接到OpenVPN服务器。

### 10.5.1.2. 客户端

Windows客户端为例。在[这里](#)下载客户端。安装在默认路径。

安装完成后，将 C:\Program Files\OpenVPN\sample-config 下的“client.ovpn”复制到 C:\Program Files\OpenVPN\config，并重命名为在服务端创建客户端证书时的名称（这里为“client01”）。此外，将在服务器上创建的文件“ca.crt”，“client01.crt”，“client01.key”也复制到 C:\Program Files\OpenVPN\config，如下所示：

## 10.5. OpenVPN



编辑 client01.ovpn 文件：

```
使用默认
client

在服务器配置中指定的设备名称
dev tap0
;dev tun

在服务器配置中指定的协议
proto tcp
;proto udp

OpenVPN服务器的公网IP和端口（替换为自己环境）
remote 172.16.2.1 1194

retry resolving
resolv-retry infinite

no bind for local port
nobind

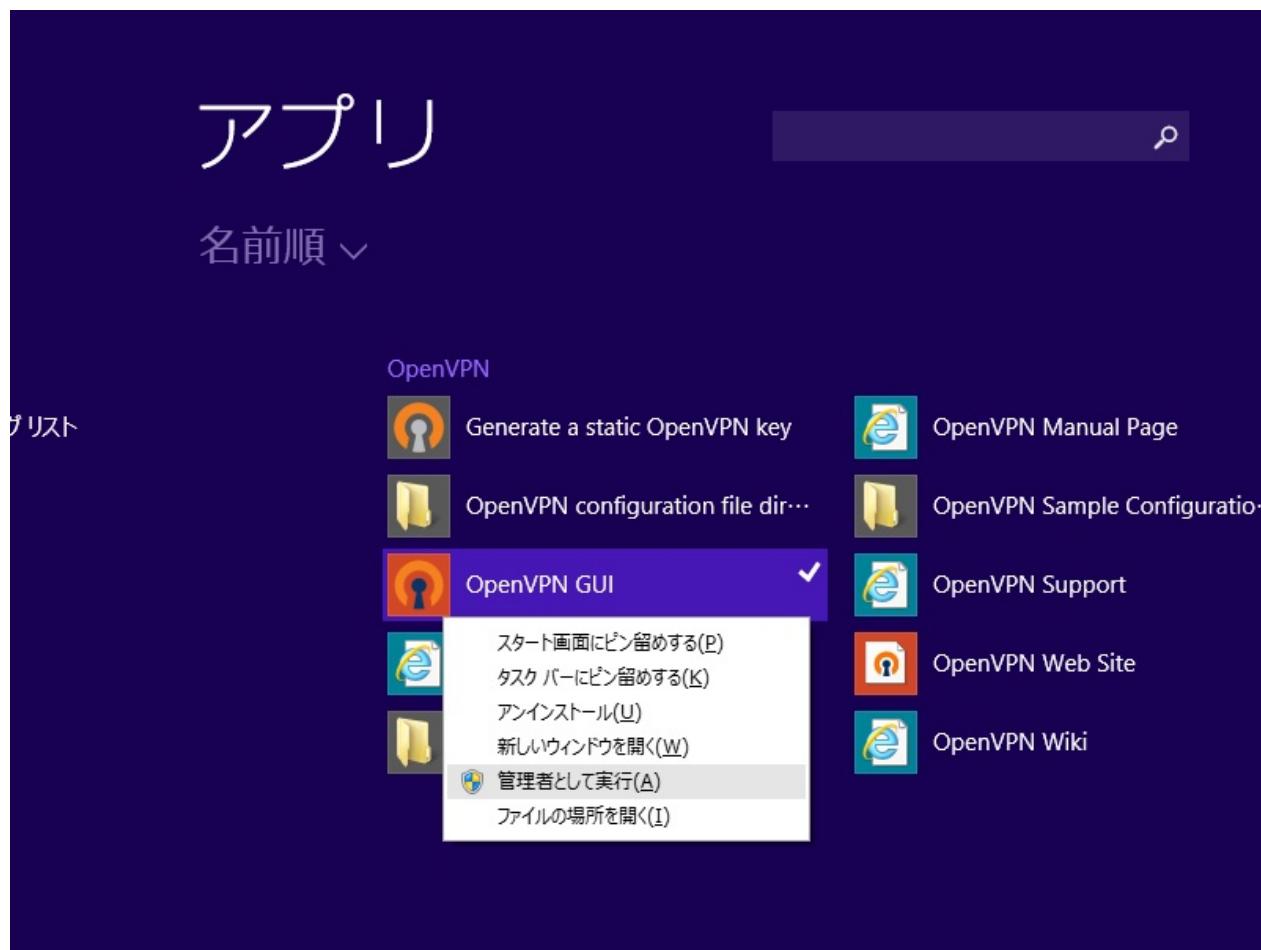
启用persist选项
persist-key
persist-tun

证书路径
ca ca.crt
cert client01.crt
key client01.key

启用压缩
;comp-lzo

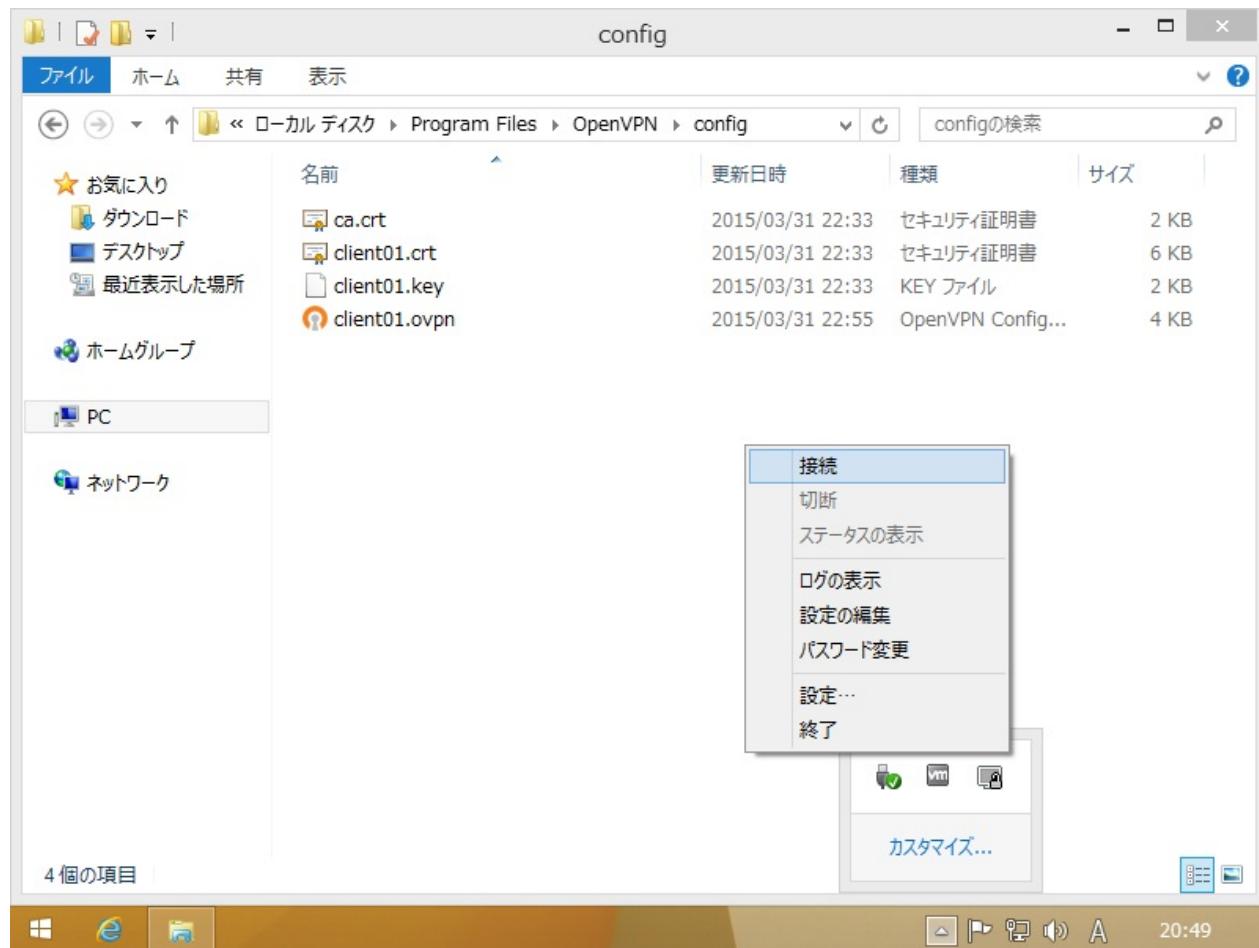
日志级别
verb 3
```

打开“开始”菜单，右键点击“OpenVPN GUI”，点击“以管理员身份运行”：



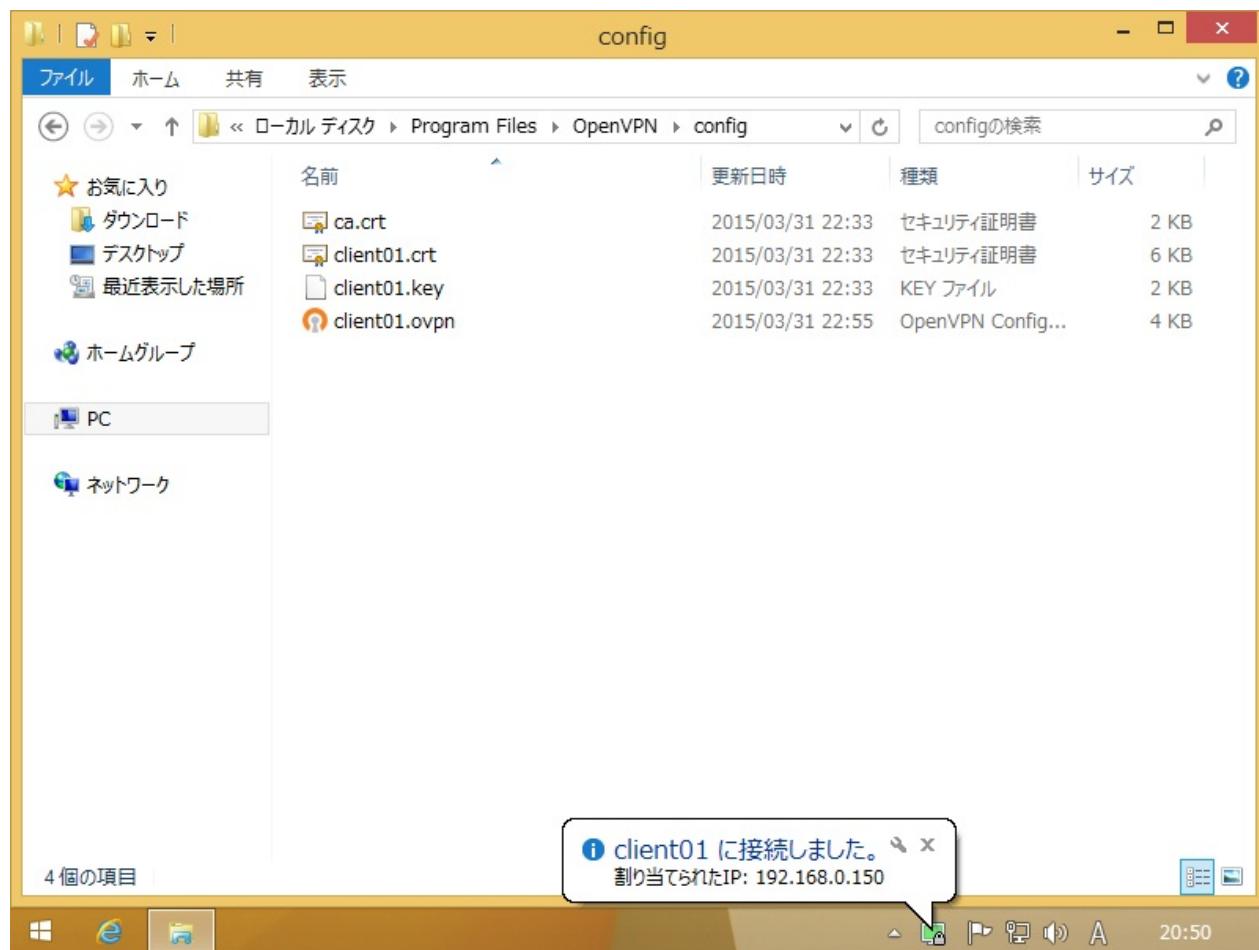
点击任务栏中的OpenVPN图标，右键点击并选择“Connect”：

## 10.5. OpenVPN



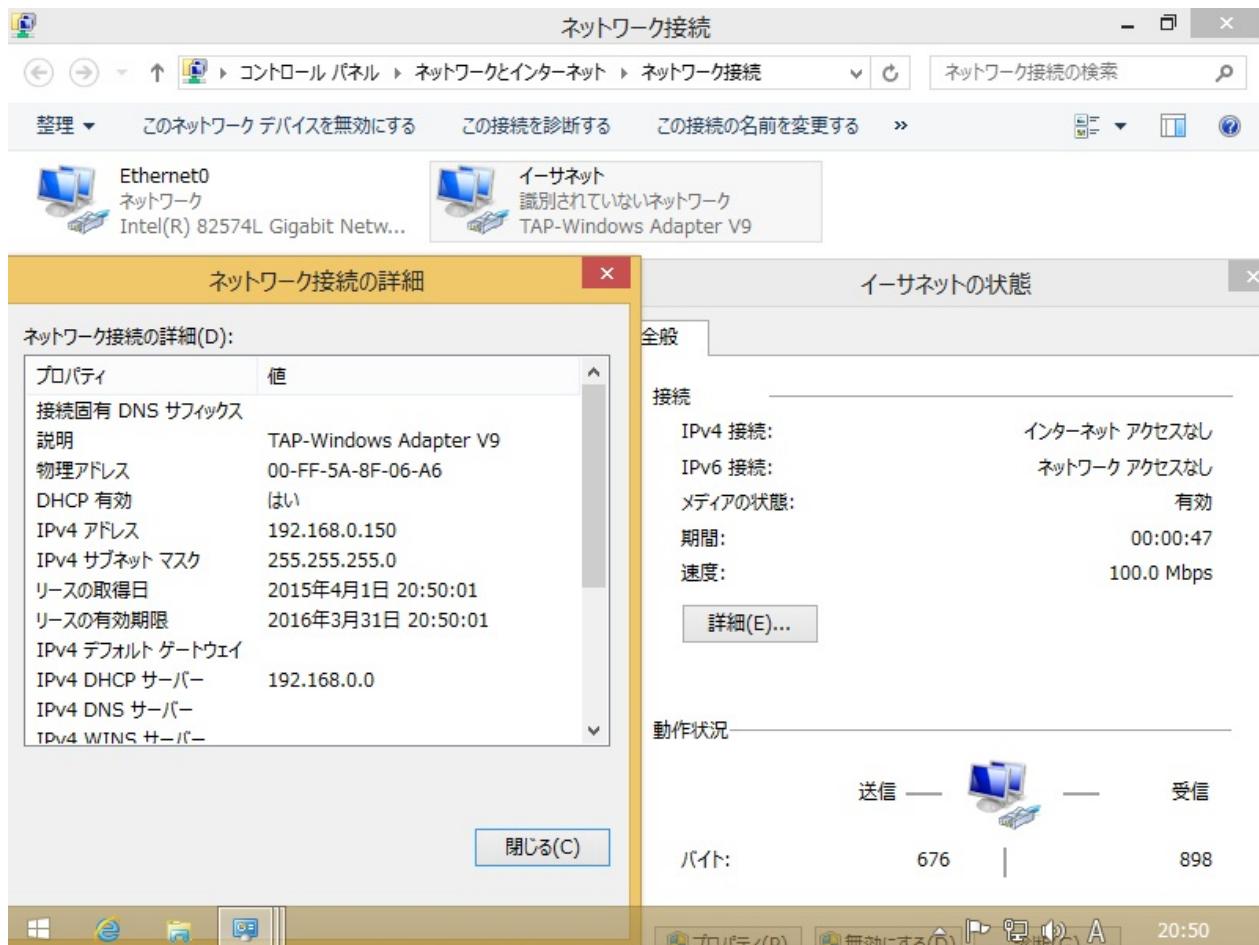
连接后，图标的颜色变为绿色：

## 10.5. OpenVPN

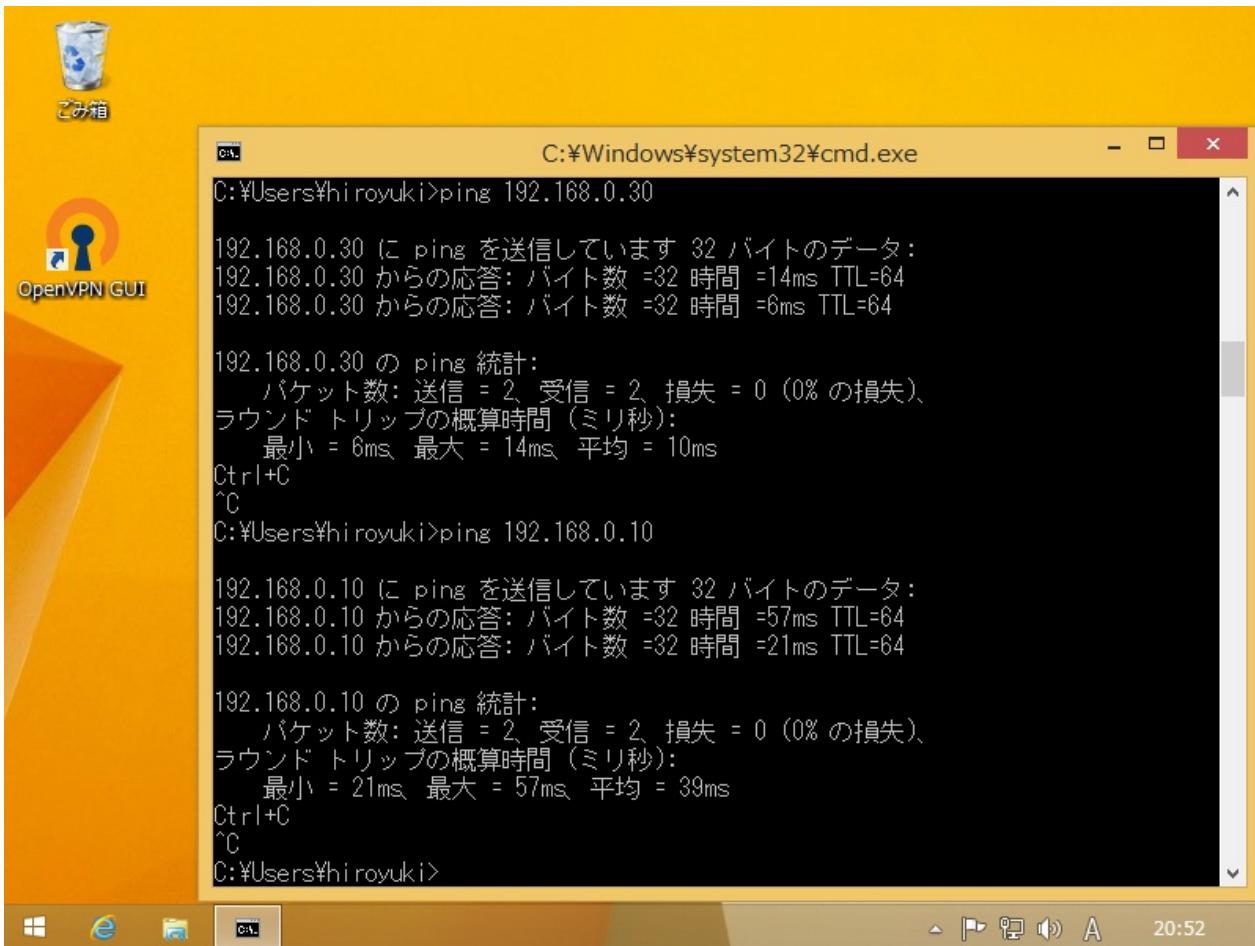


连接后，添加TAP适配器，如下所示：

## 10.5. OpenVPN



由于桥模式配置，客户端不仅可以连接OpenVPN服务器，还可以连接（与服务器）同一网络上的其他计算机，如下所示：



### 10.5.2. 客户端使用用户名密码连接

上一节为server-world上的内容，本节为笔者自己整理的。

更新：之前以为不能使用pam-mysql，在用RADIUS时感觉不太方便就就继续尝试，最终还是可以使用了

#### 10.5.2.1. 服务端

使用**RADIUS**进行认证

首先安装好认证服务器（可以使用[FreeRADIUS](#)或是[privacyIDEA + FreeRADIUS](#)），[privacyIDEA](#)实现可以多因素认证，但本质也是通过RADIUS认证（有三种基本的方法来整合OpenVPN与privacyIDEA，可以自行研究）

安装**OpenVPN**：

```
yum --enablerepo=epel -y install openvpn openssl-devel lzo-devel
pam-devel
```

禁用 SELinux。

开启 IP 转发：

编辑 `/etc/sysctl.conf` 文件：

```
net.ipv4.ip_forward = 1
```

```
sysctl -p
```

```
cat /proc/sys/net/ipv4/ip_forward
```

```
检查是否开启成功, 1为正确, 若为0需检查错误
1
```

生成证书文件（可以参考[第一节生成服务器端证书](#)）：

yum 安装 `easy-rsa`，如下操作：

```
yum -y install easy-rsa
```

```
cd /usr/share/easy-rsa/2.0
```

编辑 `vars` 文件：

```
密钥长度为2048，如果是1024可改为2048
export KEY_SIZE=2048

CA证书有效时间3650天，根据需要修改
export CA_EXPIRE=3650

密钥有效时间3650天，根据需要修改
export KEY_EXPIRE=3650

export KEY_COUNTRY="CN" # 国家
export KEY_PROVINCE="SC" # 省份
export KEY_CITY="CD" # 城市
export KEY_ORG="x" # 组织机构
export KEY_EMAIL="x@x.com" # 邮箱
export KEY_OU="x" # 单位或部门

export KEY_NAME="OpenVPNServer" # openvpn服务器的名称
```

```
source ./vars # 初始化

./clean-all # 清理keys

./build-ca # 生成 ca.crt 和 ca.key

./build-key-server server # 生
成 server.crt , server.csr 和 server.key

./build-dh # 生成 dh2048.pem

openvpn --genkey --secret ta.key # 生成 ta.key
```

keys文件详解可[参考这里](#)。

将生成的 ca.crt , server.crt , server.key , dh2048.pem , ta.key 放到 /etc/openvpn/keys 目录下。

[GitHub上的easy-rsa](#)最新版本为easy-rsa3，生成证书命令有一些变化，如下操作（有问题可官网查看或运行 `./easyrsa help`）：

```
wget https://github.com/OpenVPN/easy-rsa/archive/master.zip
unzip master.zip
cd easy-rsa-master/easyrsa3
cp vars.example vars
```

编辑 vars 文件：

```
取消注释并修改对应内容

set_var EASYRSA_REQ_COUNTRY "US" # 国家
set_var EASYRSA_REQ_PROVINCE "California" # 省份
set_var EASYRSA_REQ_CITY "San Francisco" # 城市
set_var EASYRSA_REQ_ORG "Copyleft Certificate Co" # 组织
机构
set_var EASYRSA_REQ_EMAIL "me@example.net" # 邮箱
set_var EASYRSA_REQ_OU "My Organizational Unit" # 单位或
部门

set_var EASYRSA_KEY_SIZE 2048 # 密钥长度2048

set_var EASYRSA_CA_EXPIRE 3650 # CA有效期3650天

set_var EASYRSA_CERT_EXPIRE 3650 # CERT有效期3650天

客户端使用--ns-cert-type，取消下行注释并改值改为yes，（一般不推荐使用，
而推荐使用--remote-cert-tls功能）
#set_var EASYRSA_NS_SUPPORT "no"

其他内容可以根据自己需要修改
```

保存后继续运行，生成服务器端证书：

```
./easyrsa init-pki # 初始化，会清空已有信息，并在当前目录创建PKI目录，
用于存储一些中间变量及最终生成的证书
./easyrsa build-ca # 创建根证书，会提示设置密码，用于ca对之后生成的ser
ver和client证书签名时使用，然后提示设置Common Name
./easyrsa gen-req server nopass # 创建server证书和private key，no
pass表示不加密private key，然后提示设置Common Name（使用与上一步不同的）
./easyrsa sign-req server server # 给server证书签名，确认信息后输入
yes，然后输入build-ca时设置的密码
./easyrsa gen-dh # 创建Diffie-Hellman
```

OpenVPN服务端需要的文件如下：

easyrsa3/pki/ca.crt

easyrsa3/pki/private/server.key

easyrsa3/pki/issued/server.crt

easyrsa3/pki/dh.pem

openvpn --genkey --secret ta.key # 生成 ta.key 的命令相同

生成客户端证书（如果客户端不使用证书认证，这一步就不需要了），在与上面生  
成服务端证书的easy-rsa不同的文件夹重新解压一次（网上查的资料说是在新目录  
重新生成，不知道可否直接在刚才的目录使用，未测试，如果要测试注意不要再次  
运行 ./easyrsa init-pki），进入新的 easy-rsa-master/easyrsa3 目录后  
同样设置一下 vars 文件，然后开始生成证书：

```
./easyrsa init-pki
./easyrsa gen-req client1 nopass # 创建client1证书和private key，
nopass表示不加密private key，然后提示设置Country Name（设置与上面不同的
）
```

切换到前面生成CA的目录，运行：

```
./easyrsa import-req [上一步生成客户端证书的路径]/easyrsa3/pki/reqs/
client1.req client1 # 导入req
./easyrsa sign-req client client1 # # 给client1证书签名，确认信息后
输入yes，然后输入build-ca时设置的密码
```

文件位置如下：

```
easyrsa3/pki/issued/client.crt
```

```
easyrsa3/pki/private/client.key
```

配置**OpenVPN**：

```
cp /usr/share/doc/openvpn-*/*sample/sample-config-files/server.conf
/etc/openvpn/
```

编辑 `/etc/openvpn/server.conf` 文件：

```
定义侦听IP，不指定则侦听所有IP
;local a.b.c.d

定义协议类型，tcp或udp
proto tcp
;proto udp

定义侦听端口
port 1194

定义使用模式，tap或tun
tap是桥接模式，通过软件在系统中模式出一个tap设备，该设备是一个二层设备，
同时支持链路层协议
tun是路由模式，通过软件在系统中模拟出一个tun路由，tun是ip层的点对点协议
;dev tap
dev tun

ca定义openvpn使用的CA证书文件，该文件通过build-ca命令生成，CA证书主要用于
验证客户端证书的合法性
cert定义openvpn服务器端使用的证书
key定义openvpn服务器使用的密钥文件，该文件必须严格控制其安全性
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server.key

定义Diffie Hellman文件
dh /etc/openvpn/keys/dh2048.pem

网络拓扑类型，不修改
;topology subnet
```

```
定义使用tun路由模式时，给客户端分配的IP地址段（一个客户端占用4个IP）
可以更改子网掩码来更改最大IP数量（自行参考网上子网计算方法），同一服务器同时运行多个配置文件时，IP地址池不能有重复
server 10.8.0.0 255.255.255.0

设置后，同一客户端每次分配到同一IP地址，设置规则每行一个用户名：用户名,ip
。注意需要有效的ip地址
ifconfig-pool-persist ipp.txt

定义使用tap桥接模式时，给分配的IP地址段
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

向客户端推送路由信息，比如使客户端能访问（服务器本身能访问的）192.168.10
.0网段，则加入此配置（并取消注释）
;push "route 192.168.10.0 255.255.255.0"

给具体客户端指定IP等信息，需要在openvpn目录下创建ccd目录，目录下用户名同
名的文件，进去写入对该用户生效的规则，如push route等，详见英文注释及官方文
档
client-config-dir ccd

动态修改防火墙来响应不同用户的访问，详见官方文档
;learn-address ./script

指定默认网关，客户端所有流量通过VPN
;push "redirect-gateway def1 bypass-dhcp"

向客户端推送DNS或WINS信息
;push "dhcp-option DNS 114.114.114.114"

开启后客户端可互相访问
;client-to-client

开启后同一证书可在多个客户端同时登陆，建议关闭
;duplicate-cn

每10秒ping一次，如果120秒ping不通则认为对方掉线，如果客户端容易掉线，可
将数值调小
keepalive 10 120

启用ta.key用于ssl认证，服务端配0，客户端配1
```

```

tls-auth /etc/openvpn/keys/ta.key 0

选择加密算法，需在客户端做相应配置，详见官方文档(如使用cipher AES-256-C
BC，客户端也需相应设置)
;cipher BF-CBC # Blowfish (default)
cipher AES-256-CBC # AES
;cipher DES-EDE3-CBC # Triple-DES

启用数据压缩，客户端也需要相应启用
;comp-lzo

定义最大客户端并发数，默认不限制
;max-clients 100

定义openvpn运行时使用的用户及用户组
user nobody
group nobody

通过keepalive检测超时后，重新启动VPN，不重新读取keys，保留第一次使用的k
eys
persist-key
通过keepalive检测超时后，重新启动VPN，一直保持tun或者tap设备是linkup的
，否则网络连接会先linkdown再linkup
persist-tun

临时状态文件，记录当前连接状态，每分钟刷新
status openvpn-status.log

记录日志，启用log或log-append，log每次重启openvpn后删除原有log信息，l
og-append为追加log信息
;log openvpn.log
log-append openvpn.log

日志级别，0只记录致命错误，4一般使用，5或6debug模式，9详细日志
verb 4

重复的日志最多记录数量
;mute 20

通知客户端，当服务器重新启动时，它可以自动重新连接。
这个好像是新的参数，默认打开，但是只能是UDP协议使用，TCP使用的话不能启动
服务，注释该行即可

```

```
;explicit-exit-notify 1

(如果不添加该参数) 默认值3600,也就是一个小时进行一次TSL重新协商。这个参数在服务端和客户端设置都有效。如果两边都设置了,就按照时间短的设定优先。当两边同时设置成0,表示禁用TSL重协商。使用OTP认证需要禁用
reneg-sec 0

启用管理接口,在本机使用命令“telnet localhost 7505”
management localhost 7505
```

创建ccd目录：

```
mkdir /etc/openvpn/ccd
```

启动OpenVPN（@server 对应配置文件 server.conf 的文件名）：

```
systemctl enable openvpn@server
systemctl start openvpn@server
```

防火墙设置：

参考后面pam-mysql中的内容

配置Radiusplugin：

```
wget
http://www.nongnu.org/radiusplugin/radiusplugin_v2.1a_beta1.tar.gz
在这里查看版本(好像很久没更新了)

tar -zxvf radiusplugin_v2.1a_beta1.tar.gz

cd radiusplugin_v2.1a_beta1

yum -y groupinstall 'Development Tools'

yum -y install libgcrypt-devel

make

mkdir /etc/openvpn/plugin

cp radiusplugin.so /etc/openvpn/plugin
```

## 10.5. OpenVPN

```
cp radiusplugin.cnf /etc/openvpn/plugin
```

编辑 /etc/openvpn/plugin/radiusplugin.cnf 文件：

```
修改对应内容
OpenVPNConfig=/etc/openvpn/server.conf # server.conf改为对应的配置文件
如果多个服务端配置文件，则每个新增一行OpenVPNConfig=
server
{
 # The UDP port for radius accounting.
 acctport=1813
 # The UDP port for radius authentication.
 authport=1812
 # The name or ip address of the radius server.
 name=127.0.0.1 # radius服务器地址，这里是本机
 # How many times should the plugin send the if there is
 no response?
 retry=1
 # How long should the plugin wait for a response?
 wait=1
 # The shared secret.
 sharedsecret=testing123 # 这里改成radius的shared secret,radius默认是testing123
}
```

编辑 /etc/openvpn/server.conf 文件：

```
在最后添加以下内容
client-cert-not-required
username-as-common-name
plugin /etc/openvpn/plugin/radiusplugin.so /etc/openvpn/plugin/radiusplugin.cnf
```

重启服务：

```
systemctl restart radiusd
systemctl restart openvpn@server
```

以下为使用**pam-sql**认证登录服务端安装步骤

安装配置**epel**仓库、**Remi**仓库：

```
yum -y install yum-plugin-priorities
yum -y install epel-release
sed -i -e "s/\$/\npriority=5/g" /etc/yum.repos.d/epel.repo
yum -y install http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
sed -i -e "s/\$/\npriority=10/g" /etc/yum.repos.d/remi-safe.repo
yum update
```

禁用**SELINUX**

```
setenforce 0
```

编辑 `/etc/selinux/config`：

```
修改
SELINUX=disabled
```

开启IP转发，编辑 `/etc/sysctl.conf`：

```
在最后新增
net.ipv4.ip_forward = 1
```

```
sysctl -p
cat /proc/sys/net/ipv4/ip_forward
检查是否开启成功, 1为正确, 若为0需检查错误
1
```

设置防火墙（推荐使用**iptables**, 详见后面规则）：

```
防火墙使用iptables：在/etc/sysconfig/iptables中根据需要加入
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT
-A INPUT -p tcp --dport 1194 -j ACCEPT
-A INPUT -p udp --dport 1194 -j ACCEPT
设置完重启iptables

防火墙使用firewalld，运行：
firewall-cmd --add-port={80/tcp,443/tcp,1194/tcp,1194/udp} --per
manent
firewall-cmd --reload
```

安装数据库：

```
yum -y install mariadb-server
```

编辑 /etc/my.cnf :

```
在[mysqld]下面添加以下内容
character-set-server=utf8
```

```
systemctl start mariadb
systemctl enable mariadb
mysql_secure_installation
```

安装Apache：

```
yum -y install httpd mod_ssl
rm -f /etc/httpd/conf.d/welcome.conf
mkdir /etc/httpd/vhosts
mkdir /var/www/tmp
```

编辑 /etc/httpd/conf/httpd.conf :

```

更改为服务器名称
ServerName www.srv.world:80
将Options Indexes FollowSymLinks改为
Options FollowSymLinks
添加目录索引
DirectoryIndex index.html index.php index.htm
<Directory "/var/www/html">下Require all granted修改
Require all denied
在最后新增
IncludeOptional /etc/httpd/vhosts/*.conf
<VirtualHost 80>
 DocumentRoot /var/www/tmp
 ServerName localhost
 ServerAlias *
 ServerSignature off
<Directory /var/www/tmp>
 AllowOverride None
 Options FollowSymLinks
 Require all denied
</Directory>
</VirtualHost>
<VirtualHost 443>
 DocumentRoot /var/www/tmp
 ServerName localhost:443
 ServerAlias *
 ServerSignature off
 SSLEngine On
 SSLProtocol -All +TLSv1.2
 SSLCertificateFile /etc/pki/tls/certs/localhost.crt
 SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
<Directory /var/www/tmp>
 AllowOverride None
 Options FollowSymLinks
 Require all denied
</Directory>
</VirtualHost>

```

```

systemctl start httpd
systemctl enable httpd

```

安装配置php7.3：

```
yum --enablerepo=remi-safe -y install php73 php73-php php73-php-pear php73-php-mbstring php73-php-mysql php73-php-mcrypt php73-php-bcmath php73-php-gd php73-php-tidy
mv /etc/httpd/conf.modules.d/10-php.conf /etc/httpd/conf.modules.d/10-php.conf.org # 如果存在旧版本
php73 -v
scl enable php73 bash
php -v
```

配置自启动，编辑 `/etc/profile.d/php73.sh`：

```
新建以下内容
#!/bin/bash
source /opt/remi/php73/enable
export X_SCLS=`scl enable php73 'echo $X_SCLS'`"
```

```
systemctl restart httpd
```

配置phpMyAdmin：

```
wget https://files.phpmyadmin.net/phpMyAdmin/4.8.5/phpMyAdmin-4.8.5-all-languages.tar.gz
tar zxvf phpMyAdmin-4.8.5-all-languages.tar.gz
mv phpMyAdmin-4.8.5-all-languages /var/www/pma
cd /var/www/pma
mkdir /var/www/pma/tmp # 这两步是解决“变量 $cfg['TempDir']（./tmp/）无法访问”的提示，不新建也不影响
chmod 777 /var/www/pma/tmp
cp config.sample.inc.php config.inc.php
```

编辑 `config.inc.php`：

```
#在$cfg['blowfish_secret']后面加上短语，32位以上（否则网页端会有提示，不知有何影响），如
$cfg['blowfish_secret'] = 'abcdefghijklmnopqrstuvwxyz123456';
```

## 10.5. OpenVPN

新增网页配置文件 /etc/httpd/vhosts/pma.conf :

```
<VirtualHost 80>
 ServerName localhost
 # ServerAlias x.x.x 指定域名访问
 ServerAdmin webmaster@localhost
 ServerSignature Off
 RewriteEngine On
 RewriteCond %{HTTPS} !=On
 RewriteRule (.*) https:// %{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
<VirtualHost 443>
 DocumentRoot /var/www/pma
 ServerName localhost:443
 # ServerAlias x.x.x 指定域名访问
 ServerAdmin webmaster@localhost
 ErrorLog logs/pma-ssl_error_log
 TransferLog logs/pma-ssl_access_log
 LogLevel warn
 SSLEngine On
 SSLProtocol -All +TLSv1.2
 SSLCertificateFile /etc/pki/tls/certs/localhost.crt
 SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
<Directory /var/www/pma>
 AllowOverride All
 Options FollowSymLinks
<RequireAll>
 Require all granted
 # Require ip 这里可以加上允许访问的IP以限制访问
</RequireAll>
</Directory>
</VirtualHost>
```

重启服务：

```
systemctl restart httpd
```

创建数据库及表：

```
用户 : openvpn
密码 : password
数据库名 : openvpn
表名 : users
字段 : id(int) user_id(varchar) enable(varchar) user_pass(varchar)
 user_name(varchar)

CREATE TABLE `users` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `user_id` varchar(20) NOT NULL,
 `enable` varchar(1) NOT NULL DEFAULT '1',
 `expire_date` date NOT NULL DEFAULT '2099-01-01',
 `user_pass` varchar(50) NOT NULL,
 `user_name` varchar(50) NOT NULL,
 PRIMARY KEY (`id`),
 UNIQUE KEY `user_id` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

users(记录用户账户信息)
user_id -> 登录使用的用户名
enable -> 账号是否启用(如果只使用数字判断可以使用int整数型)
expire_date -> 账号有效期
user_pass -> 登录密码
user_name -> 用户姓名(用于备注使用者)
```

安装OpenVPN：

```
yum --enablerepo=epel -y install openvpn openssl-devel lzo-devel
```

生成证书省略

创建相关文件夹：

```
mkdir -p /etc/openvpn/config/ccd # 存放客户端配置文件
mkdir /etc/openvpn/keys # 存放证书等文件
mkdir /etc/openvpn/logs # 存放日志
mkdir /etc/openvpn/plugin # 存放认证插件
mkdir /etc/openvpn/scripts # 存放执行脚本
```

## 10.5. OpenVPN

```
cp /usr/share/doc/openvpn-*/sample/sample-config-files/server.conf
/etc/openvpn/
```

编辑 `/etc/openvpn/server.conf` :

```
proto tcp
port 1194
dev tun
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server.key
dh /etc/openvpn/keys/dh.pem
server 10.8.0.0 255.255.2255.0
ifconfig-pool-persist /etc/openvpn/config/ipp.txt
;push "route 192.168.10.0 255.255.255.0"
client-config-dir /etc/openvpn/config/ccd
;client-to-client
keepalive 10 120
tls-auth /etc/openvpn/keys/ta.key 0
cipher AES-256-CBC
user nobody
group nobody
persist-key
persist-tun
status /etc/openvpn/logs/openvpn-status.log
log-append /etc/openvpn/logs/openvpn.log
verb 4
;mute 20
reneg-sec 0
management localhost 7505
```

```
systemctl start openvpn@server
systemctl enable openvpn@server
```

配置防火墙规则，这里使用 `iptables`，编辑 `/etc/sysconfig/iptables` :

```
iptables从上往下执行，如果一个条件匹配到多个规则，按最上面的生效，想实现
控制效果要注意顺序
假定以下环境：vpn虚拟网段为10.8.0.0/24，服务器局域网网段192.168.10.0
/24，vpn服务器IP为192.168.10.20
```

```

新增*nat内容，具体原理不太懂，只讲一下实现的效果
eth0可以用eth+(表示eth0, eth1...eth[n])代替，tun0同理
*nat
客户端10.8.0.6可以在192.168.10.2(vpn服务器同网段的其他服务器)未添加路由表的情况下访问到192.168.10.2
如果想通过其他主机主动添加路由表的方式来控制可以像这样添加(win):route add -p 10.8.0.0 mask 255.255.255.0 192.168.10.20
注：还需后面*filter中FORWARD控制能否访问
-A POSTROUTING -o eth0 -s 10.8.0.6 -d 192.168.10.2 -j MASQUERADE
下面扩大适用范围，根据自己需要来选择
-A POSTROUTING -o eth0 -s 10.8.0.6 -j MASQUERADE
-A POSTROUTING -o eth0 -d 192.168.10.0/24 -j MASQUERADE
-A POSTROUTING -o eth+ -j MASQUERADE
-A POSTROUTING -j MASQUERADE
COMMIT

*filter
表示INPUT, FORWARD, OUTPUT默认策略是ACCEPT
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

服务器可以ping出
-A INPUT -p icmp --icmp-type 0 -j ACCEPT
禁止ping入
-A INPUT -p icmp -j DROP
系统默认规则
意思是接受所有本机发出的请求收到的回应
RELATED：该数据包与本机发出的数据包有关
ESTABLISHED：已建立的链接状态
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
下面表示禁止10.8.0.6以外其他客户端访问服务器本机，不影响连接服务器同网段的其他主机(根据需要选择设置)
-A INPUT -i tun0 -s 10.8.0.6 -j ACCEPT
-A INPUT -i tun0 -s 10.8.0.0/24 -j DROP
开放的端口
-A INPUT -p tcp --dport 22 -j ACCEPT
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT
-A INPUT -p tcp --dport 1194 -j ACCEPT
-A INPUT -p udp --dport 1194 -j ACCEPT

```

```

系统默认规则
意思是在INPUT拒绝所有其他不符合上述任何一条规则的数据包，并且发送一条host prohibited的消息给被拒绝的主机
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A INPUT -j DROP

服务器局域网网段的其他主机可以ping通客户端，客户端不能ping通服务端局域网
(根据需求设置)
-A FORWARD -i tun+ -o eth+ -p icmp --icmp-type 0 -j ACCEPT
-A FORWARD -i eth+ -o tun+ -p icmp --icmp-type 0 -j DROP
服务器局域网网段的其他主机可以访问到客户端，接受客户端返回的请求回应
-A FORWARD -i eth+ -o tun+ -j ACCEPT
-A FORWARD -i tun+ -o eth+ -m state --state RELATED,ESTABLISHED
-j ACCEPT
设置对应客户端IP能够访问的地址范围
允许10.8.0.6访问192.168.10.0/24
-A FORWARD -i tun0 -o eth0 -s 10.8.0.6 -d 192.168.10.0/24 -j ACCEPT
允许10.8.0.0/24访问192.168.10.2-192.168.0.5并且禁止访问此段IP的3389/tcp端口
-A FORWARD -i tun0 -o eth0 -s 10.8.0.0/24 -m iprange --dst-range
192.168.10.2-192.168.10.5 -p tcp --dport 3389 -j DROP
-A FORWARD -i tun0 -o eth0 -s 10.8.0.0/24 -m iprange --dst-range
192.168.10.2-192.168.10.5 -j ACCEPT
意思是在FORWARD拒绝所有其他不符合上述任何一条规则的数据包，并且发送一条host prohibited的消息给被拒绝的主机
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

防火墙使用firewalld时，使用--direct如下添加
firewall-cmd --direct --add-rule ipv4 nat POSTROUTING 0 -j MASQUERADE
firewall-cmd --direct --add-rule ipv4 filter INPUT 0 -i tun+ -j ACCEPT
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -o tun+ -j ACCEPT
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i tun+ -s 10.8.0.0/24 -j ACCEPT
这样使用有个问题，加"--permanent"虽然不报错，但是重启后不会生效
并且在测试时，不重启服务器，只reload或是restart时，变更的规则偶尔不能生效

```

安装配置pam-mysql：

```
yum -y install pam-devel mariadb-devel bison-devel ncurses-devel
cmake libxml* zlib*
yum -y groupinstall "Development Tools" "Server Platform Development"
wget https://nchc.dl.sourceforge.net/project/pam-mysql/pam-mysql/
/0.7RC1/pam_mysql-0.7RC1.tar.gz
tar zxvf pam_mysql-0.7RC1.tar.gz
cd pam_mysql-0.7RC1
```

让编译安装的pam-mysql支持MD5密码（参考[这里](#)

<https://bugs.launchpad.net/ubuntu/+source/pam-mysql/+bug/1006005>，Lele Long 的回复）：

```
yum -y install libpam* libssl*
ln -s /usr/include/openssl/md5.h /usr/include/md5.h
```

编辑 `Makefile.in` 文件：

```
第109行修改为
DEFS = @DEFS@ -I. -I$(srcdir) -I. -DHAVE_OPENSSL
```

使用中遇到认证错误（开始正常，一段时间后无法认证通过，需要重启openvpn服务），日志如下：

```
...
Can't initialize threads: error 11
AUTH-PAM: BACKGROUND: user 'testuser' failed to authenticate: Me
mory buffer error
...
```

查了下资料，说是pam-mysql内存泄漏导致（参考[这里](#)

<http://sourceforge.net/p/pam-mysql/bugs/27/#c2aa>，Dmitry Mikhailov的回复和 Johannes Pahl的回复，指向<https://dev.mysql.com/doc/refman/5.7/en/mysql-library-init.html>）。

## 10.5. OpenVPN

修改 `pam_mysql-0.7RC1/pam_mysql.c` :

```
在第2340行: mysql_close(ctx->mysql_hdl); 下面加入以下内容
mysql_library_end();

后面为: xfree(ctx->mysql_hdl);
```

修改后能正常认证，~~是否还会出现该问题待测试~~使用了一段时间未出现问题。

```
./configure --with-openssl=/usr --with-mysql=/usr --with-pam=/usr
--with-pam-mods-dir=/lib64/security
检查是否有以下内容
checking if md5.h is derived from Cyrus SASL Version 1... no
checking md5.h usability... yes
checking md5.h presence... yes
checking for md5.h... yes
checking if md5.h is Solaris's... yes
checking for md5_calc in -lmd5... no
checking for crypt in -lcrypt... yes
checking for crypt... yes
```

```
make && make install
ll /lib64/security/pam_mysql.* # 查看是否已经安装完成
```

新建pam认证文件 `/etc/pam.d/openvpn` :

```
auth sufficient /lib64/security/pam_mysql.so user=openvpn passwd
=password host=localhost db=openvpn table=users usercolumn=user_id
passwdcolumn=user_pass where=enable=1 sqllog=0 crypt=2
account required /lib64/security/pam_mysql.so user=openvpn passwd
=password host=localhost db=openvpn table=users usercolumn=user_id
passwdcolumn=user_pass where=enable=1 sqllog=0 crypt=2
#crypt(0) -- Used to decide to use MySQL's PASSWORD() function or
crypt()
#0 = No encryption. Passwords in database in plaintext. NOT recom
mended!
#1 = Use crypt
#2 = Use MySQL PASSWORD() function
```

```
#3 = MD5
#4 = SHA1

对pam.d有个很简单的了解后，尝试并测试了实现多个条件判断的认证，下面列出作为参考
在users表中新建一个类型为date的expire_date字段，非空，默认值为“2099-01-01”，规则如下：

1、同时满足enable=1且expire_date大于或等于当前日期才能认证通过
account required中对应的条件必须满足
auth sufficient只有一条时，对应条件也必须满足，最好明确写为auth required
account required /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id passwdcolumn=user_pass where=enable=1 sqllog=0 crypt=2
auth sufficient /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id passwdcolumn=user_pass where=expire_date>=CURDATE() sqllog=0 crypt=2
也可以将条件分别列出，如下：
account required /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id
auth required /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id passwdcolumn=user_pass where=enable=1 sqllog=0 crypt=2
auth required /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id passwdcolumn=user_pass where=expire_date>=CURDATE() sqllog=0 crypt=2

2、满足其中一个条件即可
account required /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id
auth sufficient /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id passwdcolumn=user_pass where=enable=1 sqllog=0 crypt=2
auth sufficient /lib64/security/pam_mysql.so user=openvpn passwd=password host=localhost db=openvpn table=users usercolumn=user_id passwdcolumn=user_pass where=expire_date>=CURDATE() sqllog=0 crypt=2
```

## 10.5. OpenVPN

```
只是一个参考，可根据实际需求修改
```

```
chmod 600 /etc/pam.d/openvpn
```

创建openvpn-auth-pam.so（使用openvpn 2.0.9版本，历史版本查看<http://build.openvpn.net/downloads/releases/>）：

```
wget http://build.openvpn.net/downloads/releases/openvpn-2.0.9.tar.gz
tar zxvf openvpn-2.0.9.tar.gz
cd openvpn-2.0.9/plugin/auth-pam
mv openvpn-auth-pam.so /etc/openvpn/plugin/
```

编辑 /etc/openvpn/server.conf：

```
在最后加上以下内容
client-cert-not-required
username-as-common-name
plugin /etc/openvpn/plugin/openvpn-auth-pam.so openvpn
```

```
systemctl restart openvpn@server
```

数据库新建一个用户：

```
INSERT INTO `users` (`id`, `user_id`, `enable`, `user_pass`, `user_name`) VALUES (NULL, 'test', '1', PASSWORD('test'), '测试');
```

在客户端尝试登陆。

顺便放一个执行脚本的示例（后面有专门章节说明）

在上面创建的openvpn数据库中新建表：

```
CREATE TABLE `status` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `user_id` varchar(20) NOT NULL,
 `user_online` tinyint(4) NOT NULL DEFAULT '0',
 `login_times` int(11) NOT NULL DEFAULT '0',
 `last_server` varchar(30) NOT NULL DEFAULT '0',
```

```

`last_login` datetime NOT NULL DEFAULT '1000-01-01 00:00:00',
`last_logout` datetime NOT NULL DEFAULT '1000-01-01 00:00:00',
`last_remote_ip` varchar(50) NOT NULL DEFAULT '0',
`last_login_from` varchar(50) NOT NULL DEFAULT '0',
`last_total` bigint(20) NOT NULL DEFAULT '0',
`total_received` bigint(20) NOT NULL DEFAULT '0',
`total_sent` bigint(20) NOT NULL DEFAULT '0',
`grand_total` int(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`id`),
UNIQUE KEY `user_id` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

status表(记录用户状态)
user_id -> 登录用户名(不允许重复, 每个用户一条记录)
user_online -> 在线状态
login_times -> 登录次数
last_server -> 最近连接的服务器(配置文件名)
last_login -> 最后一次登录时间
last_logout -> 最后一次退出时间
last_remote_ip -> 最后一次从服务端分配到的IP
last_login_from -> 最后一次连接时客户端的IP
last_total -> 最后一次总流量(传入的数据单位默认为byte)
total_received -> 总接收流量(传入的数据单位默认为byte)
total_sent -> 总发送流量(传入的数据单位默认为byte)
grand_total -> 历史总流量(传入的数据单位默认为byte)

CREATE TABLE `logs` (
`log_id` int(11) NOT NULL AUTO_INCREMENT,
`user_id` int(11) NOT NULL,
`log_conn_server` varchar(30) NOT NULL DEFAULT '0',
`log_client_version` varchar(20) NOT NULL DEFAULT '0',
`log_trusted_ip` varchar(50) NOT NULL DEFAULT '0',
`log_trusted_port` varchar(6) NOT NULL DEFAULT '0',
`log_remote_ip` varchar(50) NOT NULL DEFAULT '0',
`log_remote_port` varchar(6) NOT NULL DEFAULT '0',
`log_protocol` varchar(10) NOT NULL DEFAULT '0',
`log_start_time` datetime NOT NULL DEFAULT '1000-01-01 00:00:00',
`log_end_time` datetime NOT NULL DEFAULT '1000-01-01 00:00:00',
`log_duration` int(11) NOT NULL DEFAULT '0',
`log_received` bigint(20) NOT NULL DEFAULT '0',

```

```

`log_send` bigint(20) NOT NULL DEFAULT '0',
`log_total` bigint(20) NOT NULL DEFAULT '0',
`log_note` varchar(50) NOT NULL DEFAULT '未通过脚本执行',
PRIMARY KEY (`log_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

logs表(记录登录日志)
user_id -> 登录用户名(使用status表中对应用户名的id做为值, 如果想要直观
点, 可设置为varchar型, 对示例脚本做相应的调整)
log_conn_server -> 当前连接的服务器(配置文件名)
log_client_version -> 当前客户端平台和版本号
log_trusted_ip -> 客户端IP
log_trusted_port -> 客户端端口
log_remote_ip -> 从服务端分配到的IP
log_remote_port -> 服务器端口
log_protocol -> 服务器协议
log_start_time -> 连接开始时间
log_end_time -> 连接结束时间
log_duration -> 本次连接持续时间(示例脚本默认单位为s)
log_received -> 本次接收流量(传入的数据单位默认为byte)
log_send -> 本次发送流量(传入的数据单位默认为byte)
log_total -> 本次总流量(传入的数据单位默认为byte)
log_note -> 备注(示例脚本用来记录最后一次更新当前日志的脚本类型)

```

补充一个微信提醒的示例（企业微信官方接口文档：<https://qydev.weixin.qq.com/wiki/index.php>；脚本参考<https://blog.csdn.net/bwlab/article/details/50725335>和<https://my.oschina.net/u/3658138/blog/1586428>）（很多内容不懂，所以只测试了能正常发送）。

先注册企业微信，创建一个应用，在应用的可见范围设置权限。

创建用户登录微信提醒脚本 `/etc/openvpn/scripts/connect-wx.sh`：

```

#!/bin/bash
用户登录微信提醒脚本

CorpID=' ' # 填入企业ID
Secret=' ' # 填入应用Secret
GURL="https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=$CorpI
D&corpsecret=$Secret"

get acccess_token
Gtoken=`/usr/bin/curl -s -G $GURL`

a=`echo $Gtoken |awk -F ':' '{print $3}'`

token=`echo $a |awk -F ',', '{print $1}'`

PURL="https://qyapi.weixin.qq.com/cgi-bin/message/send?access_to
ken=$token"

function body() {
local int AppID=xxxxx # 填入应用AgentId
local UserID=1 # 企业微信中部门成员ID(企业微信成员信息中称为帐号)
local PartyID=1 # 部门ID，定义了范围，组内成员都可接收到消息(部门ID与用
户ID可选其一使用，若不使用PartyID下面对应的一行printf也删除)
local Time=`date +'%Y-%m-%d %H:%M:%S'`
local Msg='VPN用户登录提醒：\n用户名：'$common_name'\n登录IP：'$tr
usted_ip'\n登录服务器：'$config'\n登录时间：'$Time'
printf '{\n'
printf '\t"user": "'"$UserID"',\n"
printf '\t"toparty": "'"$PartyID"',\n"
printf '\t"msgtype": "text",\n'
printf '\t"agentid": "'"$AppID"',\n"
printf '\t"text": {\n'
printf '\t\t"content": "'"$Msg"'"\n"
printf '\t},\n'
printf '\t"safe": "0"\n'
printf '}\n'
}
/usr/bin/curl --data-ascii "$(body)" $PURL

```

创建服务器状态提醒脚本 /etc/openvpn/scripts/status-wx.sh :

```

#!/bin/bash
服务器状态提醒脚本

CorpID=' '
Secret=' '
GURL="https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=$CorpI
D&corpsecret=$Secret"

get acccess_token
Gtoken=`/usr/bin/curl -s -G $GURL`

a=`echo $Gtoken |awk -F ':' '{print $3}'`

token=`echo $a |awk -F ',', '{print $1}'`

PURL="https://qyapi.weixin.qq.com/cgi-bin/message/send?access_to
ken=$token"

function body() {
local int AppID=xxxxx
local UserID=1
local PartyID=1
local Time=`date +'%Y-%m-%d %H:%M:%S'`
local Msg='VPN服务器状态提醒：\n服务器：'$config'\n状态类型：'$script_type'\n状态时间：'$Time'
printf '{\n'
printf '\t"user": "'$UserID'",\n"
printf '\t"party": "'$PartyID'",\n"
printf '\t"msgtype": "text",\n'
printf '\t"agentid": "'$AppID'",\n"
printf '\t"text": {\n'
printf '\t\t"content": "'$Msg'"\n"
printf '\t},\n'
printf '\t"safe": "0"\n'
printf '}\n'
}
/usr/bin/curl --data-ascii "$(body)" $PURL

```

经过简单的测试，对脚本进行了一些优化，上面两条留着作为参考，新建发送微信提醒脚本 /etc/openvpn/scripts/weixin.sh :

```
#!/bin/bash
用户登录微信提醒脚本

CorpID=' ' # 填入企业ID
Secret=' ' # 填入应用Secret
GURL="https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=$CorpI
D&corpsecret=$Secret"

get acccess_token
GToken=`/usr/bin/curl -s -G $GURL`
Token=`echo $GToken |awk -F '"' '{print $10}'`
PURL="https://qyapi.weixin.qq.com/cgi-bin/message/send?access_to
ken=$Token"

wxAppID=xxxxx # 填入应用AgentId
wxUserID=1 # 企业微信中部门成员ID(企业微信成员信息中称为帐号)
Body='{ "touser":"'${wxUserID}'", "msgtype":"text", "agentid":"'${w
xAppID}', "text":{ "content":"'${wxMsg}'"}, "safe":"0" }'

/usr/bin/curl --data-ascii "$Body" $PURL
```

创建基础设置脚本 /etc/openvpn/scripts/config.sh :

```

#!/bin/bash
基础设置脚本

数据库服务器主机或IP地址
HOST='localhost'
数据库端口， 默认3306
PORT='3306'
数据库用户名
USER='openvpn'
数据库密码
PASS='在此输入数据库对应用户的密码'
数据库名称
DB='openvpn'
执行mysql语句， 使用方式： ${RUN_MY}"在引号内写入需执行的mysql语句"
RUN_MY='mysql -h"${HOST}" -P"${PORT}" -u"${USER}" -p"${PASS}" "${DB}" -e '
表logs的user_id取自表status的id字段
USER_ID="(SELECT id FROM status WHERE user_id='\"${common_name}\"')"
)

时间变量用于后面微信消息
Time=`date +'%Y-%m-%d %H:%M:%S'`

设置服务器在线状态的值， server[n].conf是对应配置文件的名称
在线状态为1， 表示该用户在server1服务器在线； 在线状态为2， 表示该用户在server2服务器在线
在线状态为3， 表示该用户在server1和server2服务器在线(详见后面执行语句控制运算)
如果有更多服务器， 可以继续添加， 避免使用前面的值的和， 可以使用4， 8， 16
...等
如果不想在状态区分是连接的哪一个服务器， 可以不用下面的的规则， 后面执行语句也相应修改
if [${config} = 'server1.conf']; then ON=1
elif [${config} = 'server2.conf']; then ON=2
else ON=0
fi

```

创建客户端连接时执行的脚本 `/etc/openvpn/scripts/connect.sh` :

```

#!/bin/bash
客户端连接时执行脚本

. /etc/openvpn/scripts/config.sh

如果连接的用户名在status表中不存在，则新插入一条，已有则不做任何操作
${RUN_MY}"INSERT INTO status(user_id) SELECT '${common_name}' FROM dual WHERE NOT EXISTS (SELECT user_id FROM status WHERE user_id='${common_name}')"

更新status表在线状态等对应信息
${RUN_MY}"UPDATE status SET user_online=user_online+${ON},login_times=login_times+1,last_server='${config}',last_login=NOW(),last_remote_ip='${ifconfig_pool_remote_ip}',last_login_from='${trusted_ip}' WHERE user_id='${common_name}'"

插入logs表当前登录信息
${RUN_MY}"INSERT INTO logs(user_id,log_conn_server,log_client_version,log_trusted_ip,log_trusted_port,log_remote_ip,log_remote_port,log_protocol,log_start_time,log_note) SELECT id,'${config}','${IV_PLAT}|${IV_VER}', '${trusted_ip}', '${trusted_port}', '${ifconfig_pool_remote_ip}', '${remote_port_1}', '${proto_1}', NOW(), '${script_type}' FROM status WHERE user_id='${common_name}'"

如果需要微信提醒加入下面内容，不需要的话，直接exit 0结束即可
这里指定仅管理员账号登录时提醒，可根据需求设置自己的规则
在表users中加入admin字段，值为0表示非管理员，值为1表示管理员
admin=`${RUN_MY}"SELECT admin FROM users WHERE user_id='${common_name}'" -N`

wxMsg='VPN用户登录提醒：\n用户名：'${common_name}'\n登录IP：'${trusted_ip}'\n登录服务器：'${config}'\n登录时间：'${Time}'

if [${admin} = 1]; then . /etc/openvpn/scripts/weixin.sh && exit 0

else exit 0

fi

exit 0

```

创建客户端断开连接时执行的脚本 `/etc/openvpn/scripts/disconnect.sh`：

```
#!/bin/bash
客户端断开连接执行脚本

. /etc/openvpn/scripts/config.sh

更新logs表当前用户连接至当前服务器的最新的日志
${RUN_MY}"UPDATE logs SET log_end_time=NOW(),log_duration=TIMESTAMPDIFF(SECOND,log_start_time,NOW()),log_received=${bytes_received},log_send=${bytes_sent},log_total=${bytes_received}+${bytes_sent},log_note='${script_type}' WHERE log_id IN (SELECT * FROM (SELECT MAX(log_id) FROM logs WHERE user_id=${USER_ID} AND log_conn_server='${config}') a)"

更新status表在线状态等对应信息
${RUN_MY}"UPDATE status SET user_online=user_online-${ON},last_logout=NOW(),last_total=${bytes_received}+${bytes_sent},total_received=total_received+${bytes_received},total_sent=total_sent+${bytes_sent},grand_total=grand_total+${bytes_received}+${bytes_sent} WHERE user_id='\$common_name'""

exit 0
```

创建服务器启动时执行的脚本 `/etc/openvpn/scripts/up.sh` :

```

#!/bin/bash
服务器启动执行脚本
这里主要用来处理若服务器未正常执行完断开连接脚本时的状态更新问题
客户端故障导致的掉线，如断网等，服务器在一段时间后会判定断开并执行对应的
断开脚本
因此一般是服务器故障(突然关机或其他原因等)造成的
可以单独写一个脚本手动对各信息进行清理

. /etc/openvpn/scripts/config.sh

更新status表在线状态等对应信息
${RUN_MY}"UPDATE status SET user_online=user_online-$ON, last_1
ogout=NOW() WHERE id IN (SELECT user_id FROM logs WHERE log_conn
_server='${config}' AND log_end_time<log_start_time AND log_note
='client-connect')"

处理logs表中未执行断开连接脚本的日志
${RUN_MY}"UPDATE logs SET log_end_time=NOW(), log_note='$script_t
ype' WHERE log_conn_server='${config}' AND log_end_time<log_star
t_time AND log_note='client-connect'"'

如果需要微信提醒加入下面内容，不需要的话，直接exit 0结束即可
wxMsg='VPN服务器状态提醒：\n服务器：'${config}'\n状态类型：'$script_t
ype'\n状态时间：'${Time}
. /etc/openvpn/scripts/weixin.sh

exit 0

```

创建服务器关闭时执行的脚本 `/etc/openvpn/scripts/down.sh` :

```
#!/bin/bash
服务器关闭执行脚本

. /etc/openvpn/scripts/config.sh

如果需要微信提醒加入下面内容，不需要的话，不用新建此脚本
wxMsg='VPN服务器状态提醒：\n服务器：'${config}'\n状态类型：'${script_type}'\n状态时间：'${Time}'
. /etc/openvpn/scripts/weixin.sh

exit 0
```

完成后运行：

```
chmod 600 /etc/openvpn/scripts/*.sh
chmod +x /etc/openvpn/scripts/*.sh
chown -R nobody:nobody /etc/openvpn/scripts/
```

在服务器配置文件最后加入：

```
script-security 2
up /etc/openvpn/scripts/up.sh
client-connect /etc/openvpn/scripts/connect.sh
client-disconnect /etc/openvpn/scripts/disconnect.sh
down /etc/openvpn/scripts/down.sh # 如果不用微信提醒则不加入此行
```

重启服务，测试各脚本是否正常。

注：微信提醒脚本执行会需要几秒时间，如果需要提醒，客户端连接时会有几行 `SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)` 的提示，待执行完成后才连接成功

编辑 `/etc/systemd/system/multi-user.target.wants/openvpn@server1.service` (如果有多个配置文件，修改一个，其他会自动同步)

```
在After=network.target后面加上mariadb.service
在重启服务器时候，如果数据库先结束，则无法执行disconnect脚本，同样，若
数据库后启动，up脚本无法执行
测试数次，是可以正常更新状态的。不知道会不会还有其他问题
After=network.target mariadb.service
```

最后，openvpn目录下的权限如下：

```
/etc/openvpn/
└── [drwxr-x--- root openvpn] client
 └── [drwxr-xr-x root root] config
 ├── [drwxr-xr-x root root] ccd-server1
 │ └── [-rw-r--r-- root root] user1
 ├── [drwxr-xr-x root root] ccd-server2
 ├── [-rw----- root root] ipp-server1.txt
 └── [-rw----- root root] ipp-server2.txt
 └── [drwxr-xr-x root root] keys
 ├── [-rw----- root root] ca.crt
 ├── [-rw----- root root] dh.pem
 ├── [-rw----- root root] server.crt
 ├── [-rw----- root root] server.key
 └── [-rw----- root root] ta.key
 └── [drwxr-xr-x root root] logs
 ├── [-rw----- root root] openvpn-server1.log
 └── [-rw----- root root] openvpn-server1-status.1
og
 └── [-rw----- root root] openvpn-server2.log
 └── [-rw----- root root] openvpn-server2-status.1
og
 └── [drwxr-xr-x root root] plugin
 └── [-rwxr-xr-x root root] openvpn-auth-pam.so
 └── [drwxr-xr-x nobody nobody] scripts
 ├── [-rwx--x--x nobody nobody] config.sh
 ├── [-rwx--x--x nobody nobody] connect.sh
 ├── [-rwx--x--x nobody nobody] disconnect.sh
 ├── [-rwx--x--x nobody nobody] down.sh
 ├── [-rwx--x--x nobody nobody] up.sh
 └── [-rwx--x--x nobody nobody] weixin.sh
 └── [drwxr-x--- root openvpn] server
 ├── [-rw-r--r-- root root] server1.conf
 └── [-rw-r--r-- root root] server2.conf
```

### 10.5.2.2. 客户端

客户端配置文件存放在安装目录下 config 目录中，文件名保存为“客户端名称.ovpn”。

右键点击快捷方式选择属性，在“目标”的最后加上 `--connect 客户端名称.ovpn`，则运行快捷方式时直接以该配置文件进行连接。

证书文件单独存放（“ca.crt”和“ta.key”放在配置文件同一目录）：

```
client
dev tun
proto tcp
remote 服务器地址 1194
nobind
persist-key
persist-tun
cipher AES-256-CBC
remote-cert-tls server
resolv-retry infinite # 尝试解析每个`remote`的DNS名称的时间，如果指定了多个服务器地址，可以指定一个数值来指定尝试解析的时间（单位为“秒”，如果在指定时间内未成功转到下一个服务器）
reneg-sec 0
auth-nocache

ca ca.crt
tls-auth ta.key 1
;comp-lzo
verb 3
auth-user-pass # 在安装目录下“config”文件中新建文档，第一行用户名，第二行密码，在此参数后面加上该文档的文件名，则可以不用输用户名密码直接连接，如“auth-user-pass pass.txt”，注：Windows新版本GUI已有记住密码功能
如果不在服务端“push route”，可以在客户端添加route信息，比如使客户端能访问（服务器本身能访问的）192.168.10.0网段，则加入此配置（并取消注释）
;route 192.168.10.0 255.255.255.0
```

证书文本直接写入配置文件：

```

client
dev tun
proto tcp
remote 服务器地址 1194
nobind
persist-key
persist-tun
cipher AES-256-CBC
remote-cert-tls server
resolv-retry infinite # 尝试解析每个`remote`的DNS名称的时间，如果指定了多个服务器地址，可以指定一个数值来指定尝试解析的时间（单位为“秒”，如果在指定时间内未成功转到下一个服务器）
reneg-sec 0
auth-nocache

;comp-lzo
verb 3
auth-user-pass

<ca>
-----BEGIN CERTIFICATE-----
*** # ca.crt文本内容粘贴到这里
-----END CERTIFICATE-----
</ca>

key-direction 1
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
*** # ta.key文本内容粘贴到这里
-----END OpenVPN Static key V1-----
</tls-auth>

```

### 10.5.3. 执行脚本示例

关于脚本和环境变量的说明，可以在 `man openvpn` 中查找 SCRIPTING AND ENVIRONMENTAL VARIABLES 下的 Environmental Variables 。

脚本执行顺序

| 指令                    | 说明                                                                                                                                          |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| up                    | Executed after TCP/UDP socket bind and TUN/TAP open.                                                                                        |
| tls-verify            | Executed when we have a still untrusted remote peer.                                                                                        |
| ipchange              | Executed after connection authentication, or remote IP address change.                                                                      |
| client-connect        | 在客户端身份验证后以--mode server模式执行                                                                                                                 |
| route-up              | Executed after connection authentication, either immediately after, or some number of seconds after as defined by the --route-delay option. |
| client-disconnect     | 在客户端实例关闭时以 --mode server 模式执行                                                                                                               |
| down                  | Executed after TCP/UDP and TUN/TAP close.                                                                                                   |
| learn-address         | Executed in --mode server mode whenever an IPv4 address/route or MAC address is added to OpenVPN's internal routing table.                  |
| auth-user-pass-verify | Executed in --mode server mode on new client connections, when the client is still untrusted.                                               |

### 环境变量

一旦设置，一个变量将被无限期地保存下来，直到它被赋一个新的值或者重新启动

| 变量             | 说明                                                                                                                                                     |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| bytes_received | Total number of bytes received from client during VPN session. Set prior to execution of the --client-disconnect script.                               |
| bytes_sent     | Total number of bytes sent to client during VPN session. Set prior to execution of the --client-disconnect script.                                     |
| common_name    | The X509 common name of an authenticated client. Set prior to execution of --client-connect, --client-disconnect, and --auth-user-pass-verify scripts. |
| config         | Name of first --config file. Set on program initiation and reset on SIGHUP.                                                                            |
|                | Set to "1" if the --daemon directive is specified, or                                                                                                  |

|                        |                                                                                                                                                                                                                                                                                                                                                       |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| daemon                 | "0" otherwise. Set on program initiation and reset on SIGHUP.                                                                                                                                                                                                                                                                                         |
| daemon_log_redirect    | Set to "1" if the --log or --log-append directives are specified, or "0" otherwise. Set on program initiation and reset on SIGHUP.                                                                                                                                                                                                                    |
| dev                    | The actual name of the TUN/TAP device, including a unit number if it exists. Set prior to --up or --down script execution.                                                                                                                                                                                                                            |
| foreignoption{n}       | An option pushed via --push to a client which does not natively support it, such as --dhcp-option on a non-Windows system, will be recorded to this environmental variable sequence prior to --up script execution.                                                                                                                                   |
| ifconfig_broadcast     | The broadcast address for the virtual ethernet segment which is derived from the --ifconfig option when --dev tap is used. Set prior to OpenVPN calling the ifconfig or netsh (windows version of ifconfig) commands which normally occurs prior to --up script execution.                                                                            |
| ifconfig_local         | The local VPN endpoint IP address specified in the --ifconfig option (first parameter). Set prior to OpenVPN calling the ifconfig or netsh (windows version of ifconfig) commands which normally occurs prior to --up script execution.                                                                                                               |
| ifconfig_remote        | The remote VPN endpoint IP address specified in the --ifconfig option (second parameter) when --dev tun is used. Set prior to OpenVPN calling the ifconfig or netsh (windows version of ifconfig) commands which normally occurs prior to --up script execution.                                                                                      |
| ifconfig_netmask       | The subnet mask of the virtual ethernet segment that is specified as the second parameter to --ifconfig when --dev tap is being used. Set prior to OpenVPN calling the ifconfig or netsh (windows version of ifconfig) commands which normally occurs prior to --up script execution.                                                                 |
| ifconfig_pool_local_ip | The local virtual IP address for the TUN/TAP tunnel taken from an --ifconfig-push directive if specified, or otherwise from the ifconfig pool (controlled by the --ifconfig-pool config file directive). Only set for --dev tun tunnels. This option is set on the server prior to execution of the --client-connect and --client-disconnect scripts. |

|                         |                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ifconfig_pool_netmask   | The virtual IP netmask for the TUN/TAP tunnel taken from an --ifconfig-push directive if specified, or otherwise from the ifconfig pool (controlled by the --ifconfig-pool config file directive). Only set for --dev tap tunnels. This option is set on the server prior to execution of the --client-connect and --client-disconnect scripts. |
| ifconfig_pool_remote_ip | The remote virtual IP address for the TUN/TAP tunnel taken from an --ifconfig-push directive if specified, or otherwise from the ifconfig pool (controlled by the --ifconfig-pool config file directive). This option is set on the server prior to execution of the --client-connect and --client-disconnect scripts.                          |
| link_mtu                | The maximum packet size (not including the IP header) of tunnel data in UDP tunnel transport mode. Set prior to --up or --down script execution.                                                                                                                                                                                                |
| local                   | The --local parameter. Set on program initiation and reset on SIGHUP.                                                                                                                                                                                                                                                                           |
| local_port              | The local port number, specified by --port or --lport. Set on program initiation and reset on SIGHUP.                                                                                                                                                                                                                                           |
| password                | The password provided by a connecting client. Set prior to --auth-user-pass-verify script execution only when the via-env modifier is specified, and deleted from the environment after the script returns.                                                                                                                                     |
| proto                   | The --proto parameter. Set on program initiation and reset on SIGHUP.                                                                                                                                                                                                                                                                           |
| remote_{n}              | The --remote parameter. Set on program initiation and reset on SIGHUP.                                                                                                                                                                                                                                                                          |
| remoteport{n}           | The remote port number, specified by --port or --rport. Set on program initiation and reset on SIGHUP.                                                                                                                                                                                                                                          |
| route_net_gateway       | The pre-existing default IP gateway in the system routing table. Set prior to --up script execution.                                                                                                                                                                                                                                            |
| route_vpn_gateway       | The default gateway used by --route options, as specified in either the --route-gateway option or the second parameter to --ifconfig when --dev tun is specified. Set prior to --up script execution.                                                                                                                                           |
|                         | A set of variables which define each route to be added, and are set prior to --up script execution.                                                                                                                                                                                                                                             |

|                |                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| route{parm}{n} | parm will be one of "network", "netmask", "gateway", or "metric". n is the OpenVPN route number, starting from 1. If the network or gateway are resolvable DNS names, their IP address translations will be recorded rather than their names as denoted on the command line or configuration file.                                                                                |
| script_context | Set to "init" or "restart" prior to up/down script execution. For more information, see documentation for --up.                                                                                                                                                                                                                                                                   |
| script_type    | One of up, down, ipchange, route-up, tls-verify, auth-user-pass-verify, client-connect, client-disconnect, or learn-address. Set prior to execution of any script.                                                                                                                                                                                                                |
| signal         | The reason for exit or restart. Can be one of sigusr1, sighup, sigterm, sigint, inactive (controlled by --inactive option), ping-exit (controlled by --ping-exit option), ping-restart (controlled by --ping-restart option), connection-reset (triggered on TCP connection reset), error, or unknown (unknown signal). This variable is set just prior to down script execution. |
| tlssid{n}      | A series of certificate fields from the remote peer, where n is the verification level. Only set for TLS connections. Set prior to execution of --tls-verify script.                                                                                                                                                                                                              |
| tlsserial{n}   | The serial number of the certificate from the remote peer, where n is the verification level. Only set for TLS connections. Set prior to execution of --tls-verify script.                                                                                                                                                                                                        |
| tun_mtu        | The MTU of the TUN/TAP device. Set prior to --up or --down script execution.                                                                                                                                                                                                                                                                                                      |
| trusted_ip     | Actual IP address of connecting client or peer which has been authenticated. Set prior to execution of --ipchange, --client-connect, and --client-disconnect scripts.                                                                                                                                                                                                             |
| trusted_port   | Actual port number of connecting client or peer which has been authenticated. Set prior to execution of --ipchange, --client-connect, and --client-disconnect scripts.                                                                                                                                                                                                            |
|                | Actual IP address of connecting client or peer which has not been authenticated yet. Sometimes                                                                                                                                                                                                                                                                                    |

|                |                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| untrusted_ip   | used to nmap the connecting host in a --tls-verify script to ensure it is firewalled properly. Set prior to execution of --tls-verify and --auth-user-pass-verify scripts. |
| untrusted_port | Actual port number of connecting client or peer which has not been authenticated yet. Set prior to execution of --tls-verify and --auth-user-pass-verify scripts.          |
| username       | The username provided by a connecting client. Set prior to --auth-user-pass-verify script execution only when the via-env modifier is specified.                           |

开启了管理模式，可以 telnet 到控制台查看比如客户端连接时会有哪些参数出现。

客户端连接，断开时运行脚本（很久以前写的，对数据库不是很熟，可根据自己需要修改）：

创建数据库（先[安装数据库服务器](#)）：

```
mysql -u root -p
```

```
>CREATE DATABASE openvpn;
> set session sql_mode='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
> CREATE TABLE `openvpn`.`log` (
 `log_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
 `user_id` varchar(32) NOT NULL,
 `log_trusted_ip` varchar(32) NOT NULL DEFAULT '0',
 `log_trusted_port` varchar(10) NOT NULL DEFAULT '0',
 `log_remote_ip` varchar(32) NOT NULL DEFAULT '0',
 `log_remote_port` varchar(10) NOT NULL DEFAULT '0',
 `log_protocol` varchar(10) NOT NULL DEFAULT '0',
 `log_start_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
 `log_end_time` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
 `log_duration` int(10) UNSIGNED NOT NULL DEFAULT '0',
 `log_received` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
 `log_send` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
```

```

`log_total` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
PRIMARY KEY (`log_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

`log_id` log编号（自动增长）
`user_id` 本次登录用户账号
`log_trusted_ip` 本次客户端外网IP，默认为0
`log_trusted_port` 本次客户端端口，默认为0
`log_remote_ip` 本次客户端获取到的虚拟IP，默认为0
`log_remote_port` 本次连接服务端端口号，默认为0
`log_protocol` 本次连接服务端端口协议类型（upd/tcp），默认为0
`log_start_time` 本次登录时间，默认为当前时间
`log_end_time` 本次断开时间，默认为0000-00-00 00:00:00
`log_duration` 本次连接时长（单位：sec），默认为0
`log_received` 服务端本次接收数据（单位：byte），默认为0
`log_send` 服务端本次发送数据（单位：byte），默认为0
`log_total` 所有本次数据流量（单位：byte），默认为0

>set session sql_mode='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';

>CREATE TABLE `openvpn`.`traffic` (
 `user_id` varchar(32) NOT NULL,
 `user_online` tinyint(1) UNSIGNED NOT NULL DEFAULT '0',
 `login_times` int(10) UNSIGNED NOT NULL DEFAULT '0',
 `last_login` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
 `last_logout` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
 `last_login_from` varchar(32) NOT NULL DEFAULT '0',
 `last_duration` int(10) UNSIGNED NOT NULL DEFAULT '0',
 `last_total` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
 `total_duration` int(10) UNSIGNED NOT NULL DEFAULT '0',
 `total_received` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
 `total_sent` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
 `grand_total` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
 PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

`user_id` 用户账号
`user_online` 是否在线（0：否，1：是），默认为0
`login_times` 用户登录次数，默认为0

```

```
`last_login` 最后一次登录时间，默认为当前时间
`last_logout` 最后一次断开时间，默认为0000-00-00 00:00:00
`last_login_from` 最后一次连接用户客户端外网IP，默认为0
`last_duration` 最后一次连接时长（单位：sec），默认为0
`last_total` 最后一次连接数据流量（单位：sec），默认为0
`total_duration` 用户登录总时长(单位:sec)，默认为0
`total_received` 服务端所有接收数据（单位:byte），默认为0
`total_sent` 服务端所有发送数据（单位:byte），默认为0
`grand_total` 所有数据流量（单位:byte），默认为0
```

创建表时若不允许 timestamp 字段默认值为 0，则运行：

```
>show variables like 'sql_mode';

结果大概如下：
Variable_name Value
sql_mode ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

需要将其中NO_ZERO_IN_DATE,NO_ZERO_DATE,去掉
>set session sql_mode='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';

再次查看
>show variables like 'sql_mode';
```

其它一些命令：

```
OPTIMIZE TABLE 表名称 # 删除纪录会闲置一些空间，使用此命令收回这些空间
TRUNCATE TABLE 表名称 # 清空表数据(慎用)，一般表log可能用上(log_id会重新开始计数)
```

因数据库版本和设置不同，命令报错的话可以根据需要，自己创建数据库和表。

脚本文件：

编辑 /etc/openvpn/scripts/config.sh 文件，创建数据库服务器配置信息存储文件：

```
#!/bin/bash
Database Server (数据库服务器主机或IP地址)
HOST='localhost'
Default port = 3306 (端口，默认3306)
PORT='3306'
Username (数据库用户)
USER='root'
Password (数据库密码)
PASS='****'
database name (数据库名称)
DB='openvpn'
```

编辑 `/etc/openvpn/scripts/connect.sh` 文件，创建客户端连接时运行脚本：

```
#!/bin/bash
. /etc/openvpn/scripts/config.sh
insert user_id to table traffic if not exists (如果登陆user_id
不在表traffic中则新建一条（默认）记录)
mysql -h$HOST -P$PORT -u$USER -p$PASS $DB -e "INSERT INTO traffic(user_id) SELECT '$common_name' FROM dual WHERE NOT EXISTS (SELECT user_id from traffic WHERE user_id ='$common_name')"
set status online to user connected && record the last login
time and login from IP (将用户登陆状态设置为1（在线），记录最后一次登陆
时间和外网IP)
mysql -h$HOST -P$PORT -u$USER -p$PASS $DB -e "UPDATE traffic SET
user_online=1,login_times=login_times+1,last_login=now(),last_login_from='$trusted_ip' WHERE user_id='$common_name'"
insert data connection to table log (插入登陆数据到表log)
mysql -h$HOST -P$PORT -u$USER -p$PASS $DB -e "INSERT INTO log (log_id,user_id,log_trusted_ip,log_trusted_port,log_remote_ip,log_
remote_port,log_protocol,log_start_time) VALUES(NULL,'$common_na
me','$trusted_ip','$trusted_port','$ifconfig_pool_remote_ip','$r
emote_port_1','$proto_1',now())"

exit 0
```

编辑 `/etc/openvpn/scripts/disconnect.sh` 文件，创建客户端断开连接时运行脚本（通过 UDP 协议连接，服务器记录客户端退出的时间会有延迟，可以在客户端加入 `explicit-exit-notify 1` 解决）：

```
#!/bin/bash
. /etc/openvpn/scripts/config.sh
add disconnected information to table log (更新退出数据 (时间，流量等) 到表log，异常退出可能会无法执行此语句)
mysql -h$HOST -P$PORT -u$USER -p$PASS $DB -e "UPDATE log SET log_end_time=now(),log_duration=TIMESTAMPDIFF(SECOND,log_start_time,now()),log_received=$bytes_received,log_send=$bytes_sent,log_total=$bytes_received+$bytes_sent WHERE log_id IN (SELECT * FROM (SELECT MAX(log_id) FROM log WHERE user_id='$common_name') a)"
set status offline to user disconnected && add disconnected information to table traffic (更新用户登陆状态设置为0 (离线)，更新历史数据信息 (时间，流量等) 到表traffic)
mysql -h$HOST -P$PORT -u$USER -p$PASS $DB -e "UPDATE traffic SET user_online=0,last_logout=now(),last_duration=TIMESTAMPDIFF(SECOND,last_login,now()),last_total=$bytes_received+$bytes_sent,total_duration=total_duration+TIMESTAMPDIFF(SECOND,last_login,now()),total_received=total_received+$bytes_received,total_sent=total_sent+$bytes_sent,grand_total=grand_total+$bytes_received+$bytes_sent WHERE user_id='$common_name'"
exit 0
```

运行：

```
chmod +x /etc/openvpn/scripts/*.sh
```

再补充一个使用**SQL SERVER**的示例

假设如下环境：

- 数据库服务器IP为 192.168.0.100 (与openvpn服务器不在同一主机)，数据库用户名 openvpn\_user ，密码 openvpn\_pass ，数据库名 openvpn (测试时数据库使用版本为2008R2，注意防火墙设置及数据库远程操作权限)
- 有两台openvpn服务器，分别记录登录情况

安装[sqlcmd](#)：

```
sudo curl -o /etc/yum.repos.d/msprod.repo https://packages.microsoft.com/config/rhel/7/prod.repo
sudo yum remove unixODBC-utf16 unixODBC-utf16-devel # 如果安装了以前版本的mssql-tools，删除所有较旧的unixODBC软件包
sudo yum install -y mssql-tools unixODBC-devel
```

添加/opt/mssql-tools/bin/到PATH环境变量：

```
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
```

在其中创建好如下两个表（不允许为NULL值，字段都指定默认值0，可以根据自己使用情况修改）：

表 log (每次登录信息)

| 列名                 | 数据类型        |
|--------------------|-------------|
| log_id             | int         |
| user_id            | varchar(50) |
| log_conn_server    | varchar(30) |
| log_client_version | varchar(20) |
| log_trusted_ip     | varchar(50) |
| log_trusted_port   | int         |
| log_remote_ip      | varchar(50) |
| log_remote_port    | int         |
| log_protocol       | varchar(10) |
| log_start_time     | datetime    |
| log_end_time       | datetime    |
| log_duration       | int         |
| log_received       | bigint      |
| log_send           | bigint      |
| log_total          | bigint      |
| log_note           | varchar(50) |

表 status (登录信息概况)

| 列名              | 数据类型        |
|-----------------|-------------|
| id              | int         |
| user_id         | varchar(50) |
| user_online     | tinyint     |
| login_times     | int         |
| last_server     | varchar(30) |
| last_login      | datetime    |
| last_logout     | datetime    |
| last_login_from | varchar(50) |
| last_total      | bigint      |
| total_received  | bigint      |
| total_sent      | bigint      |
| grand_total     | bigint      |

表 `status` 中除了 `user_online`，其他字段数据都能从表 `log` 中查询出来，可按实际情况来增加或删减。

脚本文件：

编辑 `/etc/openvpn/scripts/config.sh` 文件，创建数据库服务器配置信息存储文件：

```
#!/bin/bash
数据库服务器IP
HOST=192.168.0.100
数据库用户名
USER=openvpn_user
数据库密码
PASS='openvpn_pass'
数据库名称
DB=openvpn
设置mssql-tool环境变量
PATH="$PATH:/opt/mssql-tools/bin"
服务器编号
SERV_N=serv1-$config # $config为服务端配置文件名
```

编辑 `/etc/openvpn/scripts/connect.sh` 文件，创建客户端连接时运行脚本：

```
#!/bin/bash
. /etc/openvpn/scripts/config.sh
在表status中插入/更新用户登陆状态为1（在线），并记录登陆时间和外网IP等
sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "if exists(select 1
from status where user_id='$common_name') update status set user
_online=1,login_times=login_times+1,last_server='$SERV_N',last_l
ogin=getdate(),last_login_from='$trusted_ip' where user_id='$com
mon_name' else insert into status(user_id,user_online,login_time
s,last_server,last_login,last_login_from) values('$common_name',
1,1,'$SERV_N',getdate(),'$trusted_ip')"
在表log中插入登陆数据
sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "insert into log(use
r_id,log_conn_server,log_client_version,log_trusted_ip,log_trust
ed_port,log_remote_ip,log_remote_port,log_protocol,log_start_tim
e,log_note) values('$common_name','$SERV_N','${IV_PLAT}|${IV_VER
}','$trusted_ip','$trusted_port','$ifconfig_pool_remote_ip','$re
mote_port_1','$proto_1',getdate(),'$script_type')"
exit 0
```

如果每个服务器单独记录在线状态，可以将 `if exists(select 1 from status where user_id='$common_name')` 改为 `if exists(select 1 from status where user_id='$common_name' and last_server='$SERV_N')`，然后其他语句做对应的修改。

还可以使用如下这种写法：

```
var=`sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "SET NOCOUNT ON
select user_online from status where user_id='$common_name' SET
NOCOUNT OFF" -W -h -1`
```

将查询的结果赋值给变量，再用来判断等使用。

编辑 `/etc/openvpn/scripts/disconnect.sh` 文件，创建客户端断开连接时运行脚本：

```

#!/bin/bash
. /etc/openvpn/scripts/config.sh
在表log中更新退出数据（时间，流量等）
sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "update log set log_
end_time=getdate(),log_duration=datediff(second,log_start_time,g
etdate()),log_received=$bytes_received,log_send=$bytes_sent,log_
total=$bytes_received+$bytes_sent,log_note='$script_type' where
log_id in (select max(log_id) from log where user_id='$common_na
me' and log_conn_server='$$SERV_N')"
在表status中更新用户更新数据信息（时间，流量等）
sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "update status set l
ast_logout=getdate(),last_total=$bytes_received+$bytes_sent,tota
l_received=total_received+$bytes_received,total_sent=total_sent+
$bytes_sent,grand_total=grand_total+$bytes_received+$bytes_sent
where user_id='$common_name'"
在表status中更新（没在其他服务器上登录的）用户登陆状态为0（离线）
sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "if not exists(selec
t user_id from log where user_id='$common_name' and log_conn_ser
ver!='$SERV_N' and log_note='client-connect') update status set
user_online=0 where user_id='$common_name'""

exit 0

```

编辑 `/etc/openvpn/scripts/up.sh` 文件，创建客户端连接时运行脚本：

```

#!/bin/bash
. /etc/openvpn/scripts/config.sh
在表status中更新（最后一次登录服务器为$$SERV_N的）用户登陆状态为0（离线）
; !='$SERV_N'以便区分同一用户登录不同服务器的情况
sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "update status set u
ser_online=0,last_logout=getdate() where last_server='$$SERV_N' a
nd user_id not in (select user_id from log where log_conn_ser
ver!='$SERV_N' and log_note='client-connect')"
在表log中更新（登录服务器为$$SERV_N且断开时间小于连接时间的）日志对应内容
sqlcmd -S $HOST -U $USER -P $PASS -d $DB -Q "update log set log_
end_time=getdate(),log_note='$script_type' where log_end_time<lo
g_start_time and log_conn_server='$$SERV_N'""

exit 0

```

多条数据库语句可以使用分号隔开写在一行，保存后同样运行：

```
chmod +x /etc/openvpn/scripts/*.sh
```

其他一些脚本

这里有一个连接状态改变，自动发送邮件的脚本（参考用法，没有测试）：

```
#!/bin/bash

if ["$script_type" == "client-connect"]; then
 subj='Client Connected'
elif ["$script_type" == "client-disconnect"]; then
 subj='Client Disconnected'
else
 subj=$script_type
fi

(curl -s --user 'api:key-redacted' \
 https://api.mailgun.net/v2/redacted.example.com/messages \
 -F from='Open VPN <redacted@redacted.example.com>' \
 -F to='redacted@example.com' \
 -F subject="$subj: $common_name" \
 -F text=\<- >/dev/null 2>&1

echo "Sent email ($?): $script_type: $common_name"

exit 0
```

选择一种类型的脚本后（多个脚本可以用空格分隔），编辑 `/etc/openvpn/server.conf` 文件：

```
在最后添加以下内容
script-security 2 # 2.3版本以后不使用“script-security 3 system”这种写法
up /etc/openvpn/scripts/up.sh
client-connect /etc/openvpn/scripts/connect.sh
client-disconnect /etc/openvpn/scripts/disconnect.sh
```

重启服务：

```
systemctl restart openvpn@server
```

一些需要注意的地方

- 脚本需要以 `exit 0` 结束，以确认脚本以 0 值退出（同理，可以通过返回不同的退出值来控制是否允许客户端登录）
- 使用 `user nobody` 和 `group nobody` 指令时：
  - 若脚本内容有写入文件，最好使用绝对路径，且需要授予 `nobody` 对应的文件或目录权限，如 `chown -R nobody:nobody /etc/openvpn/`
  - 若脚本内容有执行命令，如果没有执行，可以检查命令是否需要以绝对路径写出，或者提前加入到环境变量，如上面的 `config.sh` 中加入 `PATH="$PATH:/opt/mssql-tools/bin"`（尝试了加入到 `/etc/profile` 也不能省略）
  - 总的来说就是使用本指令启动 `openvpn` 后，都使用 `nobody` 身份来运行，因此执行额外命令需要相对应的权限

### 10.5.4. Ubuntu 16.04 安装 OpenVPN

暂时把Ubuntu的放这里（FreeRADIUS认证）。

#### 10.5.4.1. 系统准备

安装新系统，安装过程就不多说了，安装软件的时候把基础软件包和SSH服务装上就可以了。

安装完成后设置：

以下设置都先 `sudo -s`，输入安装时添加的用户的密码进入。

网络设置，编辑 `/etc/network/interfaces`（可以先用 `ls /proc/sys/net/ipv4/conf/` 查看网卡名称）：

```
auto ens3
iface ens3 inet dhcp #注释掉该行
iface ens3 inet static #“ens3”是网卡名称，根据实际情况输入
address 192.168.1.100 #设置IP地址
network 192.168.1.0 #设置网段
netmask 255.255.255.0 #设置子网掩码
broadcast 192.168.1.255 #设置广播地址
gateway 192.168.1.1 #设置网关
dns-nameservers 114.114.114.114 #设置DNS服务器
```

#有其他网卡可以按照上面的方式添加，如：

```
auto ens4
iface ens4 inet static
address 192.168.2.100
network 192.168.2.0
netmask 255.255.255.0
broadcast 192.168.2.255
```

#添加路由表（在终端运行的命令前加上“up”），如：

```
up route add -net 192.168.3.0/24 gw 192.168.2.1
```

设置完成后重启系统。

如果不使用IPv6：

```
echo "net.ipv6.conf.all.disable_ipv6 = 1" >> /etc/sysctl.conf
sysctl -p
ip addr
```

设置时间：

```
timedatectl #查看系统时间是否正确
timedatectl list-timezones #列出可用时区
timedatectl set-timezone Asia/Shanghai
timedatectl set-ntp yes

apt-get install ntpdate # 安装ntpdate工具
ntpdate cn.pool.ntp.org # 系统时间与网络同步
hwclock --systohc # 将时间写入硬件
```

运行更新：

```
apt-get update
apt-get -y upgrade
```

如果 `apt-get -y upgrade` 提示有软件不能更新，可能需要先运行一下 `apt-get -u dist-upgrade`。

安装防火墙，根据习惯安装，这里安装firewalld：

```
apt-get -y install firewalld
systemctl start firewalld #这两步好像不需要，应该是安装好了自动就设置
好了吧
systemctl enable firewalld
```

安装配置**vim**：

```
apt-get -y install vim
```

编辑 `~/.vimrc`（以下是CentOS上的，Ubuntu应该也可以用）：

```
set tabstop=4
set cursorline
highlight CursorLine cterm NONE ctermfg=black ctermbg=green gu
ibg=NONE guifg=NONE
set cursorcolumn
highlight CursorColumn cterm NONE ctermfg=black ctermbg=green gu
ibg=NONE guifg=NONE
" use extended function of vim (no compatible with vi)
set nocompatible
" specify encoding
set encoding=utf-8
" specify file encoding
set fileencodings=ucs-bom,utf-8,cp936,gb18030,big5,euc-jp,euc-kr
,latin1
" specify file formats
set fileformats=unix,dos
" take backup
" if not, specify [set nobackup]
```

```
set backup
" specify backup directory//指定的存储目录先建好
set backupdir=~/backup
" take 50 search histories
set history=50
" ignore Case
set ignorecase
" distinct Capital if you mix it in search words
set smartcase
" highlights matched words
" if not, specify [set nohlsearch]
set hlsearch
" use incremental search
" if not, specify [set noincsearch]
set incsearch
" show line number
" if not, specify [set nonumber]
set number
" Visualize break ($) or tab (^I)//break和tab显示为$和^I(个人习惯不使用)
" set list
" highlights parentheses
set showmatch
" show color display
" if not, specify [syntax off]
syntax on
" change colors for comments if it's set [syntax on]
highlight Comment ctermfg=LightCyan
" wrap lines
" if not, specify [set nowrap]
set wrap
" 各设置的含义可以网上查资料,根据自己需要设置
" set paste/set nopaste(paste不会改变复制文本的格式,可以在编辑时在set前加:来临时启用)
set paste
```

### 10.5.4.2. 安装服务端

```
sudo apt-get update
sudo apt-get install openvpn
```

### 安装编译环境软件

```
sudo apt-get install build-essential
sudo apt-get install libgcrypt20-dev
```

### 下载编译

```
wget http://www.nongnu.org/radiusplugin/radiusplugin_v2.1a_beta1
.tar.gz
tar -zxvf radiusplugin_v2.1a_beta1.tar.gz
cd radiusplugin_v2.1a_beta1
make
mkdir /etc/openvpn/plugin
cp radiusplugin.so /etc/openvpn/plugin
cp radiusplugin.cnf /etc/openvpn/plugin
```

### 编辑 /etc/openvpn/plugin/radiusplugin.cnf 文件：

```
修改对应内容
OpenVPNConfig=/etc/openvpn/server.conf # server.conf改为对应的配
置文件
server
{
 # The UDP port for radius accounting.
 acctport=1813
 # The UDP port for radius authentication.
 authport=1812
 # The name or ip address of the radius server.
 name=127.0.0.1 # radius服务器地址,这里是本机
 # How many times should the plugin send the if there is
 no response?
 retry=1
 # How long should the plugin wait for a response?
 wait=1
 # The shared secret.
 sharedsecret=testing123 # 这里改成radius的shared secret,r
adius默认是testing123
}
```

开启转发：

```
cat /proc/sys/net/ipv4/ip_forward
echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
sysctl -p
cat /proc/sys/net/ipv4/ip_forward
```

防火墙规则，参考CentOS7的规则，ufw防火墙有空了试一下<http://manpages.ubuntu.com/manpages/bionic/en/man8/ufw.8.html>。（注：应该就是在 /etc/ufw/ 目录下，修改对应的 before.rules 、 after.rules ，参考CentOS中iptables规则即可）

编辑 /etc/openvpn/server.conf 文件，进行服务器配置（先创建 /etc/openvpn/ccd 目录）：

```
定义侦听IP，不指定则侦听所有IP
;local a.b.c.d

定义协议类型，tcp或udp
proto tcp
;proto udp

定义侦听端口
port 1194

定义使用模式，tap或tun
tap是桥接模式，通过软件在系统中模式出一个tap设备，该设备是一个二层设备，
同时支持链路层协议
tun是路由模式，通过软件在系统中模拟出一个tun路由，tun是ip层的点对点协议
;dev tap
dev tun

ca定义openvpn使用的CA证书文件，该文件通过build-ca命令生成，CA证书主要用于验证客户端证书的合法性
cert定义openvpn服务器端使用的证书
key定义openvpn服务器使用的密钥文件，该文件必须严格控制其安全性
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server.key
```

```

定义Diffie Hellman文件
dh /etc/openvpn/keys/dh2048.pem

网络拓扑类型，不修改
;topology subnet

定义使用tun路由模式时，给客户端分配的IP地址段（一个客户端占用4个IP）
可以更改子网掩码来更改最大IP数量（自行参考网上子网计算方法），同一服务器同时运行多个配置文件时，IP地址池不能有重复
server 10.8.0.0 255.255.255.0

设置后，同一客户端每次分配到同一IP地址
ifconfig-pool-persist ipp.txt

定义使用tap桥接模式时，给分配的IP地址段
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

向客户端推送路由信息，比如使客户端能访问（服务器本身能访问的）192.168.10.0网段，则加入此配置（并取消注释）
;push "route 192.168.10.0 255.255.255.0"

给具体客户端指定IP等信息，需要在openvpn目录下创建ccd目录，目录下用户名同名的文件，进去写入对该用户生效的规则，如push route等，详见英文注释及官方文档
client-config-dir ccd

动态修改防火墙来响应不同用户的访问，详见官方文档
;learn-address ./script

指定默认网关，客户端所有流量通过VPN
;push "redirect-gateway def1 bypass-dhcp"

向客户端推送DNS或WINS信息
;push "dhcp-option DNS 114.114.114.114"

开启后客户端可互相访问
;client-to-client

开启后同一证书可在多个客户端同时登陆，建议关闭
;duplicate-cn

每10秒ping一次，如果120秒ping不通则认为对方掉线，如果客户端容易掉线，可

```

```
将数值调小
keepalive 10 120

启用ta.key用于ssl认证，服务端配0，客户端配1
tls-auth /etc/openvpn/keys/ta.key 0

选择加密算法，需在客户端做相应配置，详见官方文档(如使用cipher AES-256-C
BC，客户端也需相应设置)
;cipher BF-CBC # Blowfish (default)
cipher AES-256-CBC # AES
;cipher DES-EDE3-CBC # Triple-DES

启用数据压缩，客户端也需要相应启用
;comp-lzo

定义最大客户端并发数，默认不限制
;max-clients 100

定义openvpn运行时使用的用户及用户组
user nobody
group nogroup

通过keepalive检测超时后，重新启动VPN，不重新读取keys，保留第一次使用的k
eys
persist-key
通过keepalive检测超时后，重新启动VPN，一直保持tun或者tap设备是linkup的
，否则网络连接会先linkdown再linkup
persist-tun

临时状态文件，记录当前连接状态，每分钟刷新
status openvpn-status.log

记录日志，启用log或log-append，log每次重启openvpn后删除原有log信息，l
og-append为追加log信息
;log openvpn.log
log-append openvpn.log

日志级别，0只记录致命错误，4一般使用，5或6debug模式，9详细日志
verb 4

重复的日志最多记录数量
;mute 20
```

```
默认值3600，也就是一个小时进行一次TSL重新协商。这个参数在服务端和客户端设置都有效。如果两边都设置了，就按照时间短的设定优先。当两边同时设置成0，表示禁用TSL重协商。使用OTP认证需要禁用
reneg-sec 0

启用管理接口，在本机使用命令“telnet localhost 7505”
management localhost 7505

启用radius认证
client-cert-not-required
username-as-common-name
plugin /etc/openvpn/plugin/radiusplugin.so /etc/openvpn/plugin/radiusplugin.cnf
```

启动服务：

```
systemctl restart openvpn@server
```

### 10.5.4.3. 客户端

客户端配置文件存放在安装目录下 config 目录中，文件名保存为“客户端名称.ovpn”。

右键点击快捷方式选择属性，在“目标”的最后加上 --connect 客户端名称.ovpn ，则运行快捷方式时直接以该配置文件进行连接。

证书文件单独存放（“ca.crt”和“ta.key”放在配置文件同一目录）：

```
client
dev tun
proto tcp
remote 服务器地址 1194
nobind
persist-key
persist-tun
cipher AES-256-CBC
remote-cert-tls server
resolv-retry infinite # 尝试解析每个`remote`的DNS名称的时间，如果指定了多个服务器地址，可以指定一个数值来指定尝试解析的时间（单位为“秒”，如果在指定时间内未成功转到下一个服务器）
reneg-sec 0
auth-nocache

ca ca.crt
tls-auth ta.key 1
;comp-lzo
verb 3
auth-user-pass # 在安装目录下“config”文件中新建文档，第一行用户名，第二行密码，在此参数后面加上该文档的文件名，则可以不用输用户名密码直接连接，如“auth-user-pass pass.txt”，注：Windows新版本GUI已有记住密码功能
如果不在服务端“push route”，可以在客户端添加route信息，比如使客户端能访问（服务器本身能访问的）192.168.10.0网段，则加入此配置（并取消注释）
;route 192.168.10.0 255.255.255.0
```

证书文本直接写入配置文件：

```
client
dev tun
proto tcp
remote 服务器地址 1194
nobind
persist-key
persist-tun
cipher AES-256-CBC
remote-cert-tls server
resolv-retry infinite # 尝试解析每个`remote`的DNS名称的时间，如果指定了多个服务器地址，可以指定一个数值来指定尝试解析的时间（单位为“秒”，如果在指定时间内未成功转到下一个服务器）
reneg-sec 0
auth-nocache

;comp-lzo
verb 3
auth-user-pass

<ca>
-----BEGIN CERTIFICATE-----
*** # ca.crt文本内容粘贴到这里
-----END CERTIFICATE-----
</ca>

key-direction 1
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
*** # ta.key文本内容粘贴到这里
-----END OpenVPN Static key V1-----
</tls-auth>
```

## 10.6. PPTP

首先安装好认证服务器（可以使用FreeRADIUS或是privacyIDEA + FreeRADIUS）

以下内容来自[这里](#)。

安装PPTP服务器：

```
yum -y install pptpd
```

配置pptpd：

编辑 /etc/pptpd.conf 文件：

```
修改通过PPTP服务器获取的IP端
localip 192.168.100.1
remoteip 192.168.100.100-125
```

配置DNS服务器：

编辑 /etc/ppp/options.pptpd 文件：

```
根据需要修改
ms-dns 114.114.114.114
ms-dns 119.29.29.29
```

开启IP转发：

编辑 /etc/sysctl.conf 文件：

```
net.ipv4.ip_forward = 1
```

```
sysctl -p
```

```
cat /proc/sys/net/ipv4/ip_forward
```

```
检查是否开启成功，1为正确，若为0需检查错误
```

```
1
```

防火墙设置：

用iptables替换firewalld。

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A INPUT -i eth0 -p tcp --dport 1723 -j ACCEPT
service iptables save
systemctl start pptpd
```

安装FreeRADIUS客户端：

```
yum -y install radiusclient-ng.x86_64
```

PPTP服务器和FreeRADIUS集成：

编辑 /etc/ppp/options.pptpd 文件：

```
在最后添加以下内容
plugin radius.so
plugin radattr.so
radius-config-file /etc/radiusclient-ng/radiusclient.conf
```

复制“microsoft dictionary”：

```
cp /usr/share/freeradius/dictionary.microsoft
/usr/share/radiusclient-ng/
```

```
cp /usr/share/freeradius/dictionary.merit /usr/share/radiusclient-
ng/
```

编辑 /usr/share/radiusclient-ng/dictionary 文件：

```
在最后添加以下内容
INCLUDE /usr/share/radiusclient-ng/dictionary.microsoft
```

下载dictionary。

更新“microsoft dictionary”：

编辑 /usr/share/radiusclient-ng/dictionary.microsoft 文件，清空内容并  
复制下载的内容。

编辑 `/etc/radiusclient-ng/servers` 文件：

```
插入以下内容
localhost testing123
```

检查RADIUS配置并启动服务：

```
systemctl stop radiusd
```

```
radiusd -X
```

```
systemctl start radiusd
```

重启PPTP服务：

```
systemctl restart pptpd
```

可以创建用户名密码，在客户端进行连接。

在服务器运行 `/var/log/messages` 检查日志。

## 10.7. L2TP IPSec

教程来自[这里](#)，有时间测试了再整理一下。

### 10.7.1. 服务端

安装[StrongSwan](#)：

StrongSwan的发行版已包含在EPEL源中，但是CentOS源的包比较旧，所以在[官网](#)下载安装包，也可以使用[源码编译](#)。

```
wget http://dl.fedoraproject.org/pub/epel/7/x86_64/s/strongswan-5.4.0-2.el7.x86_64.rpm
```

```
rpm -ihv strongswan-5.4.0-2.el7.x86_64.rpm
```

安装依赖：

```
yum -y install gmp-devel
```

编译安装：

```
wget http://download.strongswan.org/strongswan.tar.gz
```

```
tar xzf strongswan.tar.gz
```

```
cd strongswan-*
```

```
./configure --sysconfdir=/etc --enable-openssl --enable-nat-transport --disable-mysql --disable-ldap --disable-static --enable-shared --enable-md4 --enable-eap-mschapv2 --enable-eap-aka --enable-eap-aka-3gpp2 --enable-eap-gtc --enable-eap-identity --enable-eap-md5 --enable-eap-peap --enable-eap-radius --enable-eap-sim --enable-eap-sim-file --enable-eap-simaka-pseudonym --enable-eap-simaka-reauth --enable-eap-simaka-sql --enable-eap-tls --enable-eap-tnc --enable-eap-ttls
```

```
make && make install
```

注：官方教程编译`configure`时没有带这么多参数，实测如果不带参数，安装成功后，在生成证书的时候，`ipsec pki --gen` 命令不会成功，也不会提示任何错误。

配置证书：

每一个完整的SSL证书都有一个公钥和一个私钥。公钥是在网络上传输的，而私钥是藏好用来和接收到的公钥配对的（因此私钥里也有整个公钥，用来配对）。

生成CA证书的私钥，并使用私钥，签名CA证书：

```
ipsec pki --gen --outform pem > ca.key.pem
```

```
ipsec pki --self --in ca.key.pem --dn "C=CN, O=VPN, CN=StrongSwan CA" --ca --lifetime 3650 --outform pem > ca.cert.pem
```

注：“C”表示国家名，“ST”为州/省名，“L”为地区名，“STREET”为（全大写）街道名；“O”表示组织名；“CN”为通用名。

生成服务器证书所需的私钥，并用CA证书签发服务器证书：

```
ipsec pki --gen --outform pem > server.key.pem
```

```
ipsec pki --pub --in server.key.pem | ipsec pki --issue --lifetime 1200 --cacert ca.cert.pem \
--cakey ca.key.pem --dn "C=CN, O=VPN, CN=vpn.linsir.org" \
--san="1.2.3.4" --san="vpn.linsir.org" --flag serverAuth --flag
ikeIntermediate \
--outform pem > server.cert.pem
```

注：第二句是从刚生成的私钥里把公钥提取出来，然后用公钥去参与后面的服务器证书签发。

- iOS客户端要求“CN”也就是通用名必须是你的服务器的URL或IP地址
- Windows 7不但要求了上面的，还要求必须显式说明这个服务器证书的用途（用于与服务器进行认证）：`--flag serverAuth`
- Mac OS X要求了“IP 安全网络密钥互换居间（IP Security IKE Intermediate）”这种增强型密钥用法（EKA）：`--flag ikeIntermediate`
- Android和iOS都要求服务器别名（serverAltName）就是服务器的URL或IP地址：`--san`

所以这里“C”，“O”的值要跟第一步的一致，“CN”值及`--san`值是服务器公网地址或URL（可以设置多个`--san`值）。否则会出现错误“13801:IKE身份验证凭证不可接受”。

安装证书：

把证书复制到strongswan目录下：

```
cp -r ca.cert.pem /etc/strongswan/ipsec.d/cacerts/
```

```
cp -r server.cert.pem /etc/strongswan/ipsec.d/certs/
```

```
cp -r server.pem /etc/strongswan/ipsec.d/private/
```

设备/操作系统使用的ike版本：

- Linux：命令行客户端就是strongswan本身，因此完美兼容，支持 ikev1/ikev2 和所有加密方法的连接
- Android：只支持ikev1（没有最新andriod手机，可能已经支持ikev2。笔者注：较新一点的系统应该已经支持）
- iOS/Mac OS X：IPsec客户端为自己修改的racoon，只支持ikev1，最新的IOS 9和Mac OS X支持ikev2
- Windows：在Windows 7以后支持ikev2，Windows XP需要用l2tp方式

配置StrongSwan：

编辑 /etc/strongswan/ipsec.conf 文件：

```
ipsec.conf - strongSwan IPsec configuration file
basic configuration
config setup
 uniqueids=never # 允许多个客户端使用同一个证书，多设备同时在线

所有项目共用的配置项
conn %default
 keyexchange=ike # ikev1或ikev2都用这个
 left=%any # 服务器端标识，%any表示任意
 leftsubnet=0.0.0.0/0 # 服务器端虚拟IP，0.0.0.0/0表示通配
 right=%any # 客户端标识，%any表示任意

conn IKE-BASE
 ikelifetime=60m
 keylife=20m
 rekeymargin=3m
 keyingtries=1
 leftcert=server.cert.pem # 服务器端证书
 rightsourceip=10.0.0.0/24 # 分配给客户端的虚拟IP段
```

```
for IOS9 and Win 7 or later
conn ike2-eap
 also=IKE-BASE
 keyexchange=ikev2
 ike=aes256-sha1-modp1024,aes128-sha1-modp1024,3des-sha1-modp1024!
 esp=aes256-sha256,aes256-sha1,3des-sha1!
 leftsendcert=always
 leftid=vpn.linsir.org
 leftauth=pubkey
 leftfirewall=yes
 rightauth=eap-mschapv2
 rightsendcert=never
 eap_identity=%any
 rekey=no
 dpdaction=clear
 fragmentation=yes
 auto=add

for IOS, use PSK key
conn IPSec-IKEv1-PSK
 also=IKE-BASE
 keyexchange=ikev1
 fragmentation=yes
 leftauth=psk
 rightauth=psk
 rightauth2=xauth
 auto=add

for android
conn IPSec-xauth
 also=IKE-BASE
 leftauth=psk
 leftfirewall=yes
 right=%any
 rightauth=psk
 rightauth2=xauth
 auto=add

for win xp l2tp, use psk
conn L2TP-PSK
```

```
keyexchange=ikev1
authby=secret
leftprotoport=17/1701 # l2tp端口
leftfirewall=no
rightprotoport=17/%any
type=transport
auto=add
```

说明：

ike : Win7 is aes256 , sha-1 , modp1024 ; iOS is aes256 , sha-256 ,  
modp1024 ; OS X is 3DES , sha-1 , modp1024

esp: Win 7 is aes256-sha1 , iOS is aes256-sha256 , OS X is 3des-sha1

iOS 支持的IKE为aes256-sha256-modp1024 , OS X为3des-sha1-modp1024 ,  
Win7为aes256-sha1-modp1024

注意ESP的顺序与IKE的一致。“leftid”后跟着就是服务器证书的“CN”（Common Name）也是iOS9设置时的远程ID（Remote ID）

具体配置说明可以参考[这里](#)或[官方文档](#)。

编辑 /etc/strongswan/strongswan.conf 文件：

```

strongswan.conf - strongSwan configuration file
#
Refer to the strongswan.conf(5) manpage for details
#
Configuration changes should be made in the included files

charon {
 load_modular = yes
 duplcheck.enable = no # 是为了能同时连接多个设备，所以把冗余检查
关闭
 compress = yes
 plugins {
 include strongswan.d/charon/*.conf
 }
 dns1 = 223.5.5.5
 dns2 = 8.8.8.8
 # for Windows WINS Server
 nbns1 = 223.5.5.5
 nbns2 = 8.8.8.8
}

include strongswan.d/*.conf

```

密码认证文件：

```

/etc/ipsec.secrets - strongSwan IPsec secrets file

: RSA server.pem
: PSK "password"
: XAUTH "password"
vpn %any : EAP "password"
wp设备名称\user : EAP "password"
仅对windowsphone8.1设备，设备名称在`设置-关于-手机信息`中查看

```

启动Strongswan：

```
ipsec start
```

或

```
systemctl start strongswan
```

IKEv1，v2搭建好了，下面配置L2TP/IPSec。

安装xl2tpd：

```
yum -y install ppp xl2tpd
```

如果提示找不到安装包，可以手动下载安装。

```
wget http://dl.fedoraproject.org/pub/epel/7/x86_64/x/xl2tpd-1.3.6-8.el7.x86_64.rpm
```

```
rpm -ihv xl2tpd-1.3.6-8.el7.x86_64.rpm
```

编辑 /etc/strongswan/ipsec.conf 文件：

```
在最后添加以下内容
conn L2TP-PSK
 keyexchange=ikev1
 authby=secret
 leftprotoport=17/1701 # l2tp端口
 leftfirewall=no
 rightprotoport=17/%any
 type=transport
 auto=add
```

编辑 /etc/xl2tpd/xl2tpd.conf 文件：

```
[global]
ipsec saref = no
#listen-addr = 1.2.3.4
port =1701

[lns default]
ip range = 10.0.1.2-10.0.1.254 # ip range不要跟上面的strongswan冲突
local ip = 10.0.1.1
require chap = yes
refuse pap = yes
require authentication = yes
name = vpn
ppp debug = yes
pppoptfile = /etc/ppp/options.xl2tpd
length bit = yes
```

编辑 `/etc/ppp/options.xl2tpd` 文件：

```
require-mschap-v2
ms-dns 223.5.5.5
ms-dns 223.6.6.6
asyncmap 0
auth
crtscts
lock
hide-password
modem
debug
name l2tpd
proxyarp
lcp-echo-interval 30
lcp-echo-failure 4
mtu 1400
noccp
connect-delay 5000
debug
logfile /var/log/xl2tpd.log
```

设置用户名，密码：

## 10.7. L2TP IPSec

编辑 `/etc/ppp/chap-secrets` 文件：

```
client server secret IP addresses
vpn * admin *
```

启动xl2tpd：

编辑 `/usr/lib/systemd/system/xl2tpd.service` 文件：

```
[Unit]
Description=Level 2 Tunnel Protocol Daemon (L2TP)
After=syslog.target network.target
After=ipsec.service
Some ISPs in Russia use l2tp without IPsec, so don't insist anymore
#Wants=ipsec.service

[Service]
Type=simple
PIDFile=/run/xl2tpd/xl2tpd.pid
ExecStart=/usr/bin/xl2tpd -D
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

```
systemctl start xl2tpd
```

配置转发及防火墙：

编辑 `/etc/sysctl.conf` 文件：

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
```

```
sysctl -p
```

ipatables防火墙规则：

```
iptables -A INPUT -p esp -j ACCEPT
iptables -A INPUT -p udp --dport 500 -j ACCEPT
iptables -A INPUT -p udp --dport 4500 -j ACCEPT
L2tp
iptables -A INPUT -p udp -m udp --dport 1701 -j ACCEPT
转发规则
iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o enp7s0f0 -j MASQUERADE
```

注意CentOS网卡名称（这里是enp7s0f0），根据实际情况来，另外有多网卡的，需要自己根据实际情况来设置iptables转发规则。

开机启动：

```
systemctl enable xl2tpd
systemctl enable strongswan
```

### 10.7.2. 客户端

#### Windows 7

导入证书：

新建一个内容如下的bat批处理文件，然后把ca.cert.pem放在同一目录下，然后右键管理员运行：

```
@echo off
@setlocal enableextensions
@set current_dir="%~dp0"
@cd /d "%current_dir%"
@echo %current_dir%
@certutil -addstore root ca.cert.pem
if %ERRORLEVEL% EQU 0 @echo not ok
pause
```

然后新建vpn即可。

注意：win8 win10 Ikev2有bug tcp/ip协议不能设置属性，关闭远程网关，我的连接上之后，需要自己手动添加路由表。

[Win10系统VPN连接IPV4属性无法打开,需要关闭远程网关解决方法](#)。

经测试，win8+使用证书登录的穿透性很差，而使用ca证书+EAP账号密码认证，连接速度很快，而且稳定。

### iOS/Mac

把CA证书发邮件给自己。在iOS上收邮件，导入两者注意是两个证书，一定要导入CA，或者后者不能使用，然后新建IPSec VPN。

可以使用四种方式建立VPN：

IPSec+EAP：

服务器是 IP 或都是 URL

账户和密码填 `ipsec.secrets` 里 EAP 前后的那两个

密钥输入 `ipsec.secrets` 里设置的 PSK 密码。

IPSec+证书：

服务器是 IP 或都是 URL

账户和密码填 `ipsec.secrets` 里 EAP 前后的那两个（XAUTH的那个密码也行）

勾选使用证书并选择之

L2TP：

服务器是 IP 或都是 URL

账户和密码填 `etc/ppp/chap-secrets` 里的

密钥输入 `ipsec.secrets` 里设置的 PSK 密码。

IKEV2 (IOS9) 首先是导入服务器 `ca.cert.pem` 证书，在设置—通用—描述文件中可以查看

类型 IKEv2：

服务器是 IP 或都是 URL

远程ID是 IP 或都是 URL

账户和密码填 `ipsec.secrets` 里 EAP 前后的那两个

### Android

IPSec Xauth PSK

IPSec 预共享密钥：写 `ipsec.secrets` 里 PSK 后面的那个密码。

## 10.7.3. 调试

服务器端的日志就足够检测出绝大多数问题的来源：

```
tail -f /var/log/strongswan-charon.log
```

或：

```
tail -f /var/log/messages
```

或：

```
journalctl -f
```

## 10.7.4. 其他

[Strongswan IKEV2免导入证书配置及调试笔记](#)

[CentOS安装配置L2TP并结合freeradius验证](#)

# 11. 负载均衡

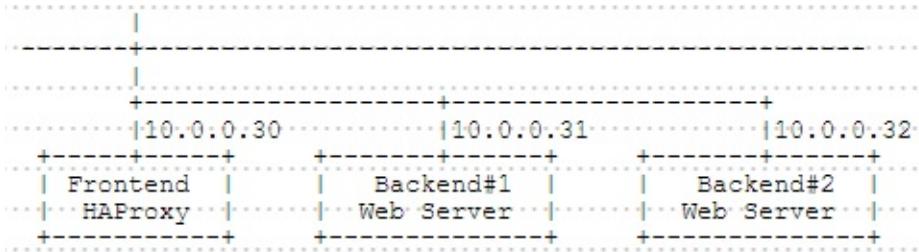
- 11.1. HAProxy
  - 11.1.1. 安装HAProxy
  - 11.1.2. SSL设置
  - 11.1.3. 查看统计信息
    - 11.1.3.1. 网页查看
    - 11.1.3.2. 命令查看
  - 11.1.4. Layer4模式
- 11.2. Pen
  - 11.2.1. 安装Pen
  - 11.2.2. SSL设置
  - 11.2.3. 查看统计信息
  - 11.2.4. MariaDB 负载均衡
- 11.3. Pound
  - 11.3.1. 安装Pound
  - 11.3.2. SSL设置
  - 11.3.3. URL重定向
- 11.4. LVS
  - 11.4.1. 安装LVS
  - 11.4.2. LVS + Keepalived

## 11.1. HAProxy

### 11.1.1. 安装HAProxy

安装[HAProxy](#)以配置负载均衡服务器。

本例基于以下环境：



与HAProxy服务器的HTTP连接转发到后端Web服务器。

```
yum -y install haproxy
```

配置HAProxy：

```
mv /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.org
```

编辑 /etc/haproxy/haproxy.cfg 文件：

```

global
 # 用于日志部分
 log 127.0.0.1 local2 info
 chroot /var/lib/haproxy
 pidfile /var/run/haproxy.pid
 # 最大每进程连接数
 maxconn 256
 # 进程用户和组
 user haproxy
 group haproxy
 # 后台守护进程
 daemon

defaults
 # 运行模式
 mode http

```

## 11.1. HAProxy

```
使用全局设置
log global
获取HTTP请求日志
option httplog
如果后端没有回复，则超时
timeout connect 10s
客户端超时
timeout client 30s
服务器端超时
timeout server 30s

定义前端（为“http-in”部分设置任意名称）
frontend http-in
 # 倾听80端口
 bind *:80
 # 设置默认后端
 default_backend backend_servers
 # 发送X-Forwarded-For头
 option forwardfor

定义后端
backend backend_servers
 # 轮询均衡
 balance roundrobin
 # 定义后端服务器
 server www01 10.0.0.31:80 check
 server www02 10.0.0.32:80 check
```

```
systemctl start haproxy
systemctl enable haproxy
```

配置Rsyslog以获取HAProxy的日志：

编辑 `/etc/rsyslog.conf` 文件：

## 11.1. HAProxy

```
取消注释并添加
$ModLoad imudp
$UDPServerRun 514
$AllowedSender UDP, 127.0.0.1

如下更改
*.info;mail.none;authpriv.none;cron.none,local2.none /var/log/
messages
local2.* /var/lo
g/haproxy.log
```

```
systemctl restart rsyslog
```

将后端的httpd设置更改为记录X-Forwarded-For头：

编辑 /etc/httpd/conf/httpd.conf 文件：

```
如下更改
LogFormat "\"%{X-Forwarded-For}i\" %l %u %t \"%r\" %>s %b \"%{Re
ferrer}i\" \"%{User-Agent}i\"" combined
```

```
systemctl restart httpd
```

如下所示从客户端使用HTTP访问前端服务器以确保所有工作正常：





### 11.1.2. SSL设置

配置HAProxy使用SSL。HAproxy和客户端之间的连接使用SSL进行加密。  
(HAproxy与后端正常连接)。

本例基于上一节环境配置。

创建SSL证书：

```
cd /etc/pki/tls/certs

openssl req -x509 -nodes -newkey rsa:2048 -keyout
/etc/pki/tls/certs/haproxy.pem -out /etc/pki/tls/certs/haproxy.pem -
days 365
```

## 11.1. HAProxy

```
Generating a 2048 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/pki/tls/certs/haproxy.pem'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:CN # 国家
State or Province Name (full name) []:SC # 省
Locality Name (eg, city) [Default City]:CD # 城市
Organization Name (eg, company) [Default Company Ltd]:GTS # 公司
Organizational Unit Name (eg, section) []:Server World # 部门
Common Name (eg, your name or your server's hostname) []:www.srv
.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
```

```
chmod 600 haproxy.pem
```

编辑 /etc/haproxy/haproxy.cfg 文件：

```
添加在“global”部分
global
 # 每个进程的SSL连接数
 maxsslconn 256
 # Diffie-Hellman密钥设置为2048位
 tune.ssl.default-dh-param 2048

添加在“frontend”部分
frontend http-in
 bind *:80
 # 指定端口和证书
 bind *:443 ssl crt /etc/pki/tls/certs/haproxy.pem
```

## 11.1. HAProxy

```
systemctl restart haproxy
```

如下所示从客户端使用HTTPS访问前端服务器以确保所有工作正常：



### 11.1.3. 查看统计信息

#### 11.1.3.1. 网页查看

配置HAProxy以在Web上查看HAProxy的统计信息。

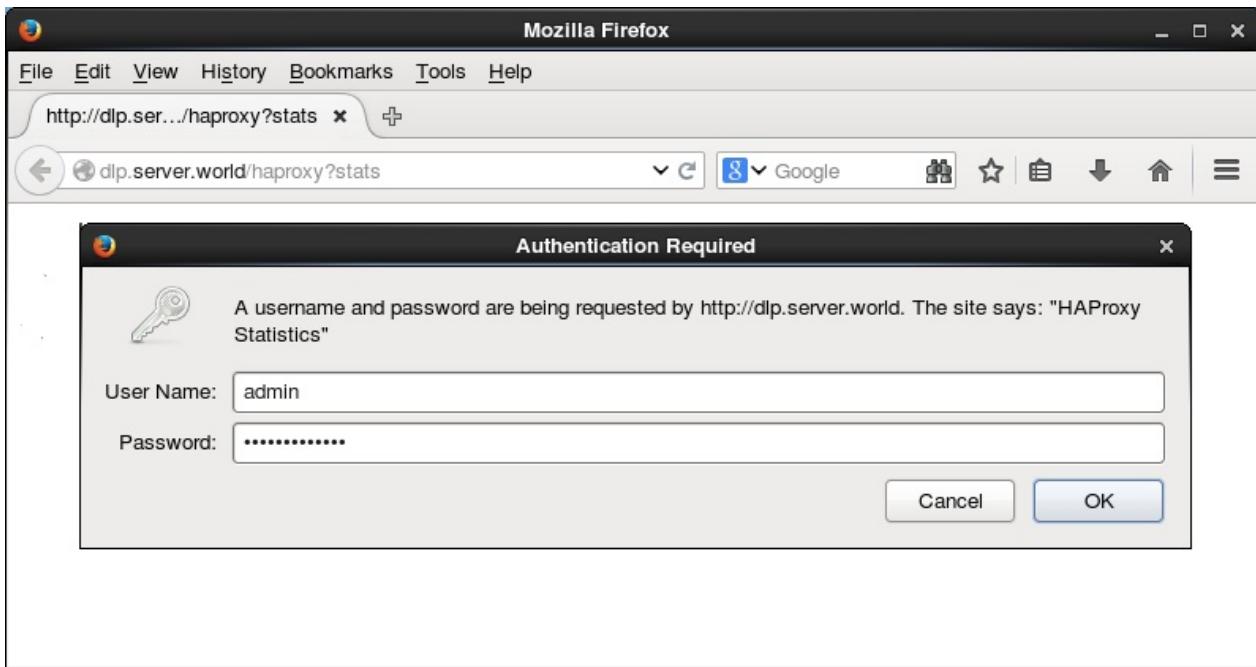
编辑 /etc/haproxy/haproxy.cfg 文件：

```
添加在“frontend”部分
frontend http-in
 bind *:80
 # 启用统计报告
 stats enable
 # 统计网站的验证信息
 stats auth admin:adminpassword
 # 隐藏HAProxy的版本
 stats hide-version
 # 显示HAProxy主机名
 stats show-node
 # 刷新时间
 stats refresh 60s
 # 统计报告URI
 stats uri /haproxy?stats
```

## 11.1. HAProxy

```
systemctl restart haproxy
```

如下所示，客户端使用HTTP/HTTPS访问前端服务器，必须身份验证，输入在配置中设置的验证信息：



登录成功，可以在这里查看HAProxy统计信息：

**Statistics Report for HAProxy on dlp.server.world - Mozilla Firefox**

**HAProxy**

**Statistics Report for pid 1923 on dlp.server.world**

**> General process information**

| http-in  |       |              |     |     |          |      |     |       |       |       |       | Server |     |     |        |     |      |          |      |       |        |         |      |     |     |     |     |        |         |
|----------|-------|--------------|-----|-----|----------|------|-----|-------|-------|-------|-------|--------|-----|-----|--------|-----|------|----------|------|-------|--------|---------|------|-----|-----|-----|-----|--------|---------|
|          | Queue | Session rate |     |     | Sessions |      |     |       | Bytes |       |       | Denied |     |     | Errors |     |      | Warnings |      |       |        |         |      |     |     |     |     |        |         |
| Cur      | Max   | Limit        | Cur | Max | Limit    | Cur  | Max | Limit | Total | LbTot | Last  | In     | Out | Req | Resp   | Req | Conn | Resp     | Retr | Redis | Status | LastChk | Wght | Act | Bck | Chk | Dwn | Dwntme | Thrtile |
| Frontend | 1     | 2            | -   | 1   | 2        | 2000 | 8   |       |       | 1574  | 16445 | 0      | 0   | 4   |        |     |      |          |      |       | OPEN   |         |      |     |     |     |     |        |         |

| backend_servers |       |              |     |     |          |     |     |       |       |       |       | Server |     |     |        |     |      |          |      |       |         |             |             |     |     |     |     |        |         |   |
|-----------------|-------|--------------|-----|-----|----------|-----|-----|-------|-------|-------|-------|--------|-----|-----|--------|-----|------|----------|------|-------|---------|-------------|-------------|-----|-----|-----|-----|--------|---------|---|
|                 | Queue | Session rate |     |     | Sessions |     |     |       | Bytes |       |       | Denied |     |     | Errors |     |      | Warnings |      |       |         |             |             |     |     |     |     |        |         |   |
| Cur             | Max   | Limit        | Cur | Max | Limit    | Cur | Max | Limit | Total | LbTot | Last  | In     | Out | Req | Resp   | Req | Conn | Resp     | Retr | Redis | Status  | LastChk     | Wght        | Act | Bck | Chk | Dwn | Dwntme | Thrtile |   |
| www01           | 0     | 0            | -   | 0   | 1        | 0   | 1   | -     | 1     | 1m50s | 286   | 381    | 0   | 0   | 0      | 0   | 0    | 0        | 0    | 0     | 2m3s UP | L4OK in 0ms | 1           | Y   | -   | 0   | 0   | 0s     | -       |   |
| www02           | 0     | 0            | -   | 0   | 1        | 0   | 1   | -     | 1     | 1m50s | 297   | 369    | 0   | 0   | 0      | 0   | 0    | 0        | 0    | 0     | 0       | 2m3s UP     | L4OK in 0ms | 1   | Y   | -   | 0   | 0      | 0s      | - |
| Backend         | 0     | 0            | -   | 0   | 2        | 0   | 1   | 200   | 2     | 2     | 1m50s | 583    | 750 | 0   | 0      | 0   | 0    | 0        | 0    | 0     | 0       | 2m3s UP     |             | 2   | 2   | 0   |     | 0      | 0s      |   |

### 11.1.3.2. 命令查看

## 11.1. HAProxy

配置HAProxy以使用命令查看HAProxy的统计信息。

```
yum -y install socat
```

编辑 /etc/haproxy/haproxy.cfg 文件：

```
添加在“global”部分
global
 # 绑定UNIX sockets
 stats socket /var/lib/haproxy/stats
```

```
systemctl restart haproxy
```

如下查看统计信息：

显示当前统计：

```
echo "show info" | socat /var/lib/haproxy/stats stdio
```

```
Name: HAProxy
Version: 1.5.2
Release_date: 2014/07/12
Nbproc: 1
Process_num: 1
Pid: 1953
...
...
Idle_pct: 100
node: dlp.srv.world
description:
```

以CSV格式显示统计：

```
echo "show stat" | socat /var/lib/haproxy/stats stdio
```

## 11.1. HAProxy

```
pxname,svname,qcur,qmax,scur,smax,slim,stot,bin,bout,dreq,dres
p,ereq,econ,eresp,.....
http-in,FRONTEND,,,0,1,2000,1,0,187,0,0,1,,,,,OPEN,,,,,,,,,1,2,0
,,,0,0,0,1,,,0,0,.....
backend_servers,www01,0,0,0,0,,0,0,0,,0,,0,0,0,0,UP,1,1,0,0,0,67
,0,,1,3,1,,0,,2,0,,.....
backend_servers,www02,0,0,0,0,,0,0,0,,0,,0,0,0,0,UP,1,1,0,0,0,67
,0,,1,3,2,,0,,2,0,,.....
backend_servers,BACKEND,0,0,0,0,200,0,0,0,0,,0,0,0,0,UP,2,2,0,
,0,67,0,,1,3,0,,0,,.....
```

显示当前会话：

```
echo "show sess" | socat /var/lib/haproxy/stats stdio
```

```
0x7fbc09349150: proto=tcpv4 src=10.0.0.18:55987 fe=http-in be=<N
ONE> srv=<none> ts=08
age=4s calls=8 rq[f=400000h,i=0,an=1ch,rx=26s,wx=,ax=] rp[f=0482
00h,i=0,an=00h,rx=,wx=,ax=]
s0=[7,8h,fd=1,ex=] s1=[0,0h,fd=2,ex=] exp=26s
0x7fbc09351d80: proto=unix_stream src=unix:1 fe=GLOBAL be=<NONE>
srv=<none> ts=0b age=0s
calls=1 rq[f=c08200h,i=0,an=00h,rx=10s,wx=,ax=] rp[f=008002h,i=0
,an=00h,rx=,wx=,ax=] s0=[7,8h,fd=3,ex=]
s1=[7,8h,fd=-1,ex=] exp=
```

其他操作：

进入交互模式：

```
socat readline /var/lib/haproxy/stats
```

```
prompt
显示帮助
> help
Unknown command. Please enter one of the following commands only
:
 clear counters : clear max statistics counters (add 'all' for
all counters)
```

## 11.1. HAProxy

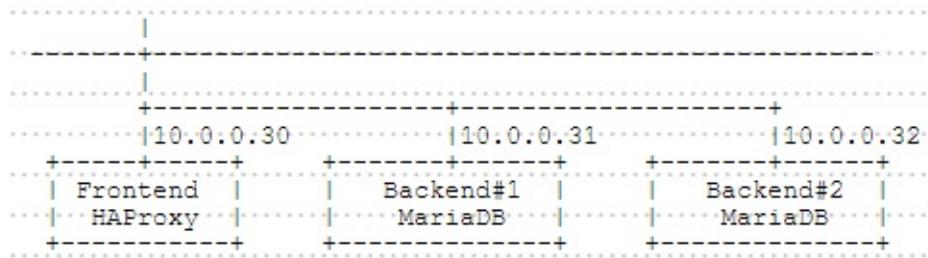
```
clear table : remove an entry from a table
help : this message
prompt : toggle interactive mode with prompt
quit : disconnect
show info : report information about the running process
show pools : report information about the memory pools usage
show stat : report counters for each proxy and server
show errors : report last request and response errors for each proxy
show sess [id] : report the list of current sessions or dump this session
show table [id]: report table usage stats or dump this table's contents
get weight : report a server's current weight
set weight : change a server's weight
set server : change a server's state or weight
set table [id] : update or create a table entry's data
set timeout : change a timeout setting
set maxconn : change a maxconn setting
set rate-limit : change a rate limiting value
disable : put a server or frontend in maintenance mode
enable : re-enable a server or frontend which is in maintenance mode
shutdown : kill a session or a frontend (eg:to release listening ports)
show acl [id] : report available acls or dump an acl's contents
get acl : reports the patterns matching a sample for an ACL
add acl : add acl entry
del acl : delete acl entry
clear acl <id> : clear the content of this acl
show map [id] : report available maps or dump a map's contents
get map : reports the keys and values matching a sample for a map
set map : modify map entry
add map : add map entry
del map : delete map entry
clear map <id> : clear the content of this map
set ssl <stmt> : set statement for ssl
```

```
退出交互模式
> quit
```

### 11.1.4. Layer4模式

在Layer4模式配置HAProxy。

本例基于以下环境：



编辑 /etc/haproxy/haproxy.cfg 文件：

## 11.1. HAProxy

```
global
 log 127.0.0.1 local2 info
 chroot /var/lib/haproxy
 pidfile /var/run/haproxy.pid
 maxconn 256
 maxsslconn 256
 user haproxy
 group haproxy
 daemon

defaults
 # 为Layer4设置“mode tcp”
 mode tcp
 log global
 timeout connect 10s
 timeout client 30s
 timeout server 30s

定义前端和后端服务器
frontend mysql-in
 bind *:3306
 default_backend backend_servers

backend backend_servers
 balance roundrobin
 server db01 10.0.0.31:3306 check
 server db02 10.0.0.32:3306 check
```

```
systemctl restart haproxy
```

如下所示，从客户端使用SQL访问前端服务器以确保所有工作正常：

```
mysql -u keystone -p -h 10.0.0.30 keystone -e "select * from
table01;"
```

## 11.1. HAProxy

```
Enter password:
+-----+
| id | name |
+-----+
| 1 | db01.srv.world |
+-----+
```

```
mysql -u keystone -p -h 10.0.0.30 keystone -e "select * from
table01;"
```

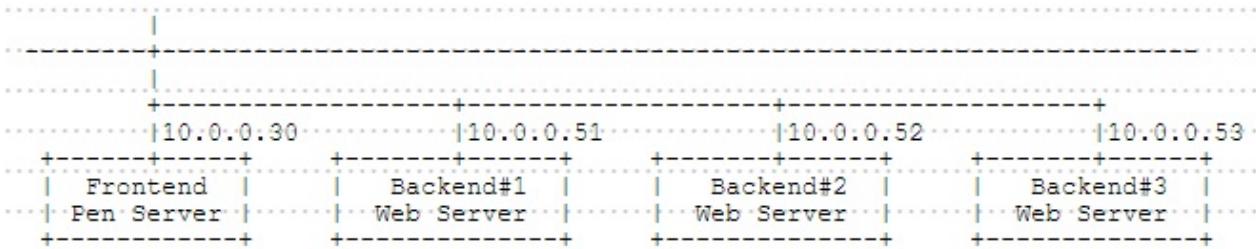
```
Enter password:
+-----+
| id | name |
+-----+
| 1 | db02.srv.world |
+-----+
```

## 11.2. Pen

### 11.2.1. 安装Pen

Pen是轻量级的简单负载均衡软件。Pen基于TCP协议，所以它可以不仅可以用在HTTP，还可以用在SMTP，FTP，LDAP等。

本例基于以下环境：



配置Pen以向Backend#1(后端#1)，Backend#2(后端#2)，Backend#3(后端#3)Web服务器加载均衡。

```
yum --enablerepo=epel -y install pen # 从EPEL安装
```

编辑 /etc/pen.conf 文件：

```

日志文件
LOGFILE=/var/log/pen.log

统计报告文件
WEBFILE=/var/www/pen/webstats.html

最大连接数
MAX_CONNECTIONS=256

发送X-Forwarded-For头
XFORWARDEDFOR=true

轮询模式
ROUNDROBIN=true

倾听端口
PORT=80

后端数量
BACKEND=3

定义后端服务器
SERVER1=10.0.0.51:80
SERVER2=10.0.0.52:80
SERVER2=10.0.0.53:80

```

创建init脚本：

编辑 `/etc/rc.d/init.d/pend` 文件：

```

#!/bin/bash

pend: Start/Stop Pend
chkconfig: - 90 10
description: Pen is a light weight simple load balancer.
pidfile: /var/run/pen.pid

. /etc/rc.d/init.d/functions
. /etc/pen.conf

```

```

LOCKFILE="/var/lock/subsys/pen"
PID=/var/run/pen.pid
PROG=/usr/bin/pen
PROGNAME=Pen

RETVAL=0
start() {
 SERVER=`grep "^SERVER" /etc/pen.conf | cut -d= -f2`

 [$XFORWARDEDFOR = "true"] && SERVER="-H $SERVER"

 [$ROUNDROBIN = "true"] && SERVER="-r $SERVER"

 [$SSLCERTS] && SERVER="-E $SSLCERTS $SERVER"

 echo -n $"Starting $PROGNAME: "

 daemon $PROG $PORT -w $WEBFILE -x $MAX_CONNECTIONS -p $PID -

 l $LOGFILE -S $BACKEND $SERVER
 RETVAL=$?
 echo
 [$RETVAL -eq 0] && touch $LOCKFILE
 return $RETVAL
}
stop() {
 echo -n $"Stopping $PROGNAME: "
 killproc $PROG
 RETVAL=$?
 echo
 [$RETVAL -eq 0] && rm -f $PID $LOCKFILE
 return $RETVAL
}
case "$1" in
 start)
 start
 ;;
 stop)
 stop
 ;;
 status)
 status -p "$PID" -l $PROG $PROGNAME
 ;;
 restart)
 stop
 start
 ;;

```

```
*)
 echo $"Usage: $0 {start|stop|status|restart}"
 exit 1
esac
exit $?
```

```
chmod 755 /etc/rc.d/init.d/pend
```

创建Systemd设置文件：

编辑 `/usr/lib/systemd/system/pen.service` 文件：

```
[Unit]
Description=Pend service
After=network.target

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/etc/rc.d/init.d/pend start
ExecStop=/etc/rc.d/init.d/pend stop

[Install]
WantedBy=multi-user.target
```

```
systemctl start pen
systemctl enable pen
```

在后端服务器上配置httpd，记录X-Forwarded-For的日志：

编辑 `/etc/httpd/conf/httpd.conf` 文件：

```
更改
LogFormat "\"%{X-Forwarded-For}i\" \"%l %u %t \"%r\" %>s %b \"%{Re
ferrer}i\" \"%{User-Agent}i\"" combined
```

```
systemctl restart httpd
```

如下所示从客户端使用HTTP访问前端服务器以确保所有工作正常：



### 11.2.2. SSL设置

配置Pen使用SSL。Pen和客户端之间的连接使用SSL进行加密。（Pen与后端正常连接）。

本例基于上一节环境配置。

创建SSL证书：

```
cd /etc/pki/tls/certs

openssl req -x509 -nodes -newkey rsa:2048 -keyout
/etc/pki/tls/certs/pen.pem -out /etc/pki/tls/certs/pen.pem -days
365
```

```
Generating a 2048 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/pki/tls/certs/pen.pem'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:CN # 国家
State or Province Name (full name) []:SC # 省
Locality Name (eg, city) [Default City]:CD # 城市
Organization Name (eg, company) [Default Company Ltd]:GTS # 公司
Organizational Unit Name (eg, section) []:Server World # 部门
Common Name (eg, your name or your server's hostname) []:www.srv
.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
```

```
chmod 600 pen.pem
```

编辑 /etc/pen.conf 文件：

```
更改端口并指定证书
PORT=443
SSLCERTS=/etc/pki/tls/certs/pen.pem
```

```
systemctl restart pen
```

如下所示从客户端使用HTTPS访问前端服务器以确保所有工作正常：



### 11.2.3. 查看统计信息

先[安装httpd](#)

另外，由于Pen使用了80端口，因此将httpd的端口从80更改为另一个端口（本例使用8081）。

配置Pen：

```
cp /usr/share/doc/pen-*/*penstats /var/www/pen
```

编辑 `/var/www/pen/penstats` 文件：

```
更改
PIDFILE=/var/run/pen.pid
WEBFILE=/var/www/pen/webstats.html
```

编辑 `/etc/httpd/conf.d/pen.conf` 文件：

```
如下更改
Alias /pen/ /var/www/pen/
<Directory /var/www/pen/>
 DirectoryIndex webstats.html
 #Options ExecCGI
 <IfModule mod_authz_core.c>
 # Apache 2.4
 # 添加访问权限
 Require local
 Require ip 10.0.0.0/24
 </IfModule>
```

## 11.2. Pen

```
systemctl restart httpd
chmod 755 /var/www/pen/penstats
```

生成统计：

```
/var/www/pen/penstats > /dev/null
```

添加到Cron：

```
echo '*/5 * * * * /var/www/pen/penstats > /dev/null' >
/etc/cron.d/pend
```

访问 `http://(Pen服务器的主机名或IP地址):(httpd侦听端口)/pen/`，然后可以查看Pen的统计数据如下：

The screenshot shows the Mozilla Firefox browser window displaying the 'Pen status page'. The address bar shows the URL `http://dlp.server.world/pen/`. The main content area is titled 'Pen status page' and shows the following data:

**Time** 2015-02-02 14:42:47, 3 servers, 1 current

| server | address   | status | port | connections | max | soft | max  | hard | sent | received | weight | prio |
|--------|-----------|--------|------|-------------|-----|------|------|------|------|----------|--------|------|
| 0      | 10.0.0.51 | 0      | 80   | 0           | 0   | 0    | 314  | 407  | 0    | 0        | 0      | 0    |
| 1      | 10.0.0.52 | 0      | 80   | 1           | 0   | 0    | 3731 | 2367 | 0    | 0        | 0      | 0    |
| 2      | 10.0.0.53 | 0      | 80   | 0           | 0   | 0    | 850  | 615  | 0    | 0        | 0      | 0    |

**Active clients**

Max number of clients: 2048

| client | address   | age(secs) | last server | connects | sent | received |
|--------|-----------|-----------|-------------|----------|------|----------|
| 0      | 10.0.0.1  | 5         | 0           | 1        | 0    | 0        |
| 1      | 10.0.0.18 | 4         | 1           | 1        | 1149 | 2125     |

**Active connections**

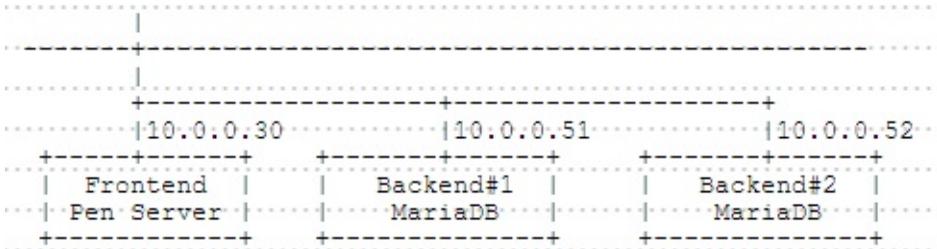
Number of connections: 256 max, 1 used, 1 last

| connection | downfd | upfd | pending | data down | pending | data up | client | server |
|------------|--------|------|---------|-----------|---------|---------|--------|--------|
| 1          | 8      | 9    | 0       | 0         | 1       | 1       |        |        |

### 11.2.4. MariaDB 负载均衡

将Pen配置为加载均衡到两个MariaDB后端。

本例基于以下环境：



先按照[第一节内容](#)，配置好基本设置。

配置Pen：

编辑 `/etc/pen.conf` 文件：

```
LOGFILE=/var/log/pen.log
WEBFILE=/var/www/pen/webstats.html
MAX_CONNECTIONS=256
ROUNDROBIN=true

倾听端口
PORT=3306

后端数量
BACKEND=2

定义后端服务器
SERVER1=10.0.0.51:3306
SERVER2=10.0.0.52:3306
```

`systemctl restart pen`

如下所示，从客户端使用SQL访问前端服务器以确保所有工作正常：

```
mysql -u keystone -p -h 10.0.0.30 keystone -e "select * from table01;"
```

## 11.2. Pen

```
Enter password:
```

| id | name           |
|----|----------------|
| 1  | db01.srv.world |

```
mysql -u keystone -p -h 10.0.0.30 keystone -e "select * from table01;"
```

```
Enter password:
```

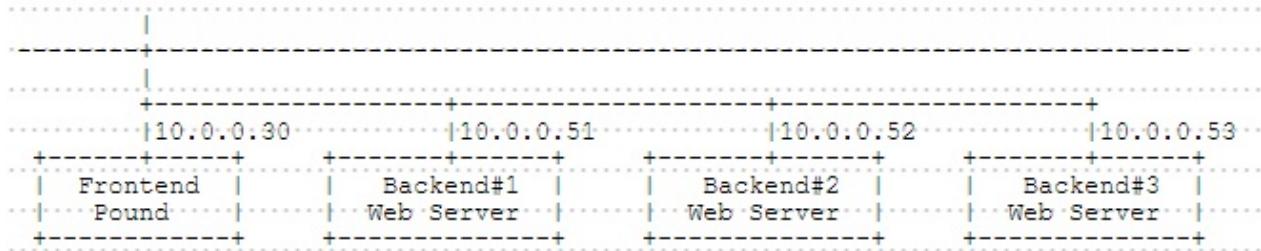
| id | name           |
|----|----------------|
| 1  | db02.srv.world |

## 11.3. Pound

### 11.3.1. 安装Pound

Pound是HTTP/HTTPS负载均衡软件。

本例基于以下环境：



配置Pound以向Backend#1(后端#1)，Backend#2(后端#2)，Backend#3(后端#3)Web服务器加载均衡。

```
yum --enablerepo=epel -y install Pound # 从EPEL安装
```

```
mv /etc/pound.cfg /etc/pound.cfg.org
```

编辑 /etc/pound.cfg 文件：

```

User "pound"
Group "pound"
日志级别（最大：5）
LogLevel 3
指定LogFacility
LogFacility local1
心跳间隔 - 秒
Alive 30

定义前端
ListenHTTP
 Address 0.0.0.0
 Port 80
End

定义后端
Service
 BackEnd
 # 后端服务器的IP地址
 Address 10.0.0.51
 # 后端服务器的端口
 Port 80
 # 设置优先级（值为1-9，最大9）
 Priority 5
 End

 BackEnd
 Address 10.0.0.52
 Port 80
 Priority 5
 End

 BackEnd
 Address 10.0.0.53
 Port 80
 Priority 5
 End
End

```

```

sed -i -e "s/^PIDFile/#PIDFile/"
/usr/lib/systemd/system/pound.service

```

```
systemctl start pound
systemctl enable pound
```

配置Rsyslog以获取Pound的日志：

编辑 `/etc/rsyslog.conf` 文件：

```
如下更改
.info;mail.none;authpriv.none;cron.none,local1.none /var/log/
messages
local1.* /var/lo
g/pound.log
```

```
systemctl restart rsyslog
```

将后端的httpd设置更改为记录X-Forwarded-For的日志：

编辑 `/etc/httpd/conf/httpd.conf` 文件：

```
更改
LogFormat "\"%{X-Forwarded-For}i\" %l %u %t \"%r\" %>s %b \"%{Re
ferer}i\" \"%{User-Agent}i\"" combined
```

```
systemctl restart httpd
```

如下所示从客户端使用HTTP访问前端服务器以确保所有工作正常：





### 11.3.2. SSL设置

配置Pound使用SSL。Pound和客户端之间的连接使用SSL进行加密。（Pound与后端正常连接）。

本例基于上一节环境配置。

创建SSL证书：

```
cd /etc/pki/tls/certs

openssl req -x509 -nodes -newkey rsa:2048 -keyout
/etc/pki/tls/certs/pound.pem -out /etc/pki/tls/certs/pound.pem
```

```
Generating a 2048 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/pki/tls/certs/pound.pem'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:CN # 国家
State or Province Name (full name) []:SC # 省
Locality Name (eg, city) [Default City]:CD # 城市
Organization Name (eg, company) [Default Company Ltd]:GTS # 公司
Organizational Unit Name (eg, section) []:Server World # 部门
Common Name (eg, your name or your server's hostname) []:www.srv
.world # 服务器域名全称
Email Address []:xxx@srv.world # 管理员邮箱
```

```
chmod 600 pound.pem
```

编辑 /etc/pound.cfg 文件：

```
如下添加
ListenHTTP
 Address 0.0.0.0
 Port 80
End
ListenHTTPS
 Address 0.0.0.0
 Port 443
 Cert "/etc/pki/tls/certs/pound.pem"
End
```

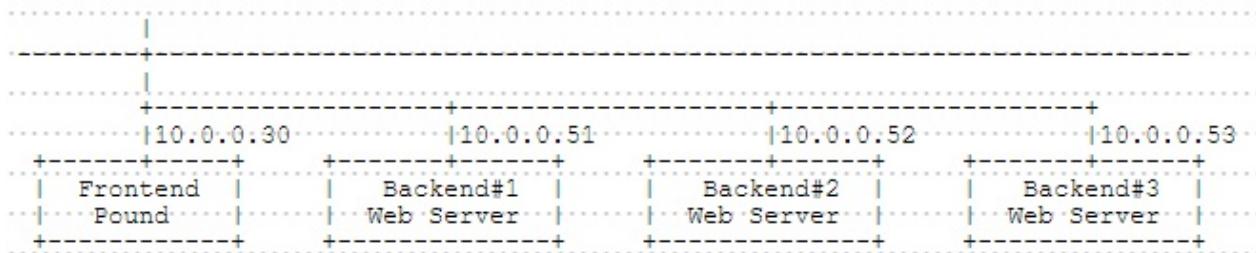
如下所示从客户端使用HTTPS访问前端服务器以确保所有工作正常：



### 11.3.3. URL重定向

来自URL匹配的重定向设置。

本例基于以下环境：



如下配置Pound：

到 `dlp.srv.world` 的HTTP连接转发到Backend#1(后端#1)，

到 `dlp.virtual.host` 的HTTP连接转发到Backend#2(后端#2)，

到除开上面的HTTP连接转发到Backend#3(后端#3)。

配置Pound：

```
mv /etc/pound.cfg /etc/pound.cfg.org
```

编辑 `/etc/pound.cfg` 文件：

```
User "pound"
Group "pound"
LogLevel 3
LogFacility local1
Alive 30

ListenHTTP
 Address 0.0.0.0
 Port 80
End

Service
 # 为dlp.srv.world定义
 HeadRequire "Host: .*dlp.srv.world"
 BackEnd
 Address 10.0.0.51
 Port 80
 Priority 5
 End
End

Service
 # 为dlp.virtual.host
 HeadRequire "Host: .*dlp.virtual.host"
 BackEnd
 Address 10.0.0.52
 Port 80
 Priority 5
 End
End

Service
 # 为其他定义
 HeadRequire "Host: .*"
 BackEnd
 Address 10.0.0.53
 Port 80
 Priority 5
 End
End
```

### 11.3. Pound

```
systemctl restart pound
```

如下所示从客户端使用HTTP访问前端服务器以确保所有工作正常：

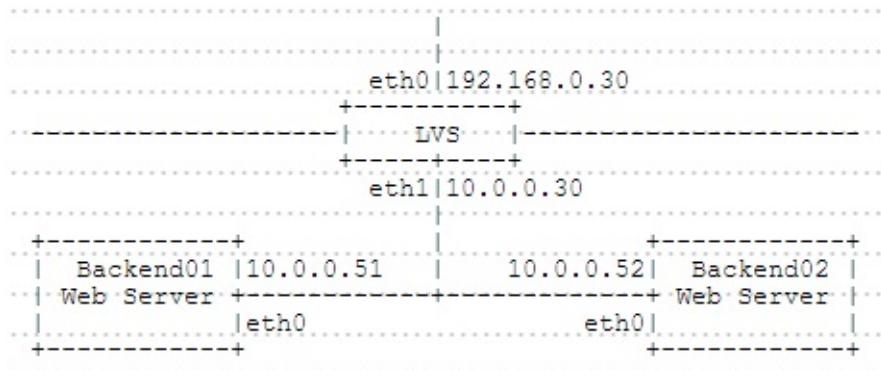


## 11.4. LVS

### 11.4.1. 安装LVS

配置[LVS](#)（Linux Virtual Server虚拟服务器）构建负载均衡器。

本例基于以下环境：



到LVS服务器eth0的HTTP数据包将使用NAT转发到Backend01和Backend02服务器。

首先在两个后端Web服务器上将默认网关更改为LVS的内部IP地址。（本例为 10.0.0.30）

安装ipvsadm：

```
yum -y install ipvsadm
```

启用IP转发：

```
echo 'net.ipv4.ip_forward = 1' >> /etc/sysctl.conf
```

```
sysctl -p
```

```
touch /etc/sysconfig/ipvsadm
```

```
systemctl start ipvsadm
systemctl enable ipvsadm
```

配置负载均衡：

清除表：

```
ipvsadm -C
```

添加虚拟服务，格式为： ipvsadm -A -t (服务器IP:端口) -s (分配方式) (分配方式见下表) ipvsadm -A -t 192.168.0.30:80 -s wlc

添加后端实体服务器，格式为： ipvsadm -a -t (服务器IP:端口) -r (实体服务器IP:端口) -m (“m”表示伪装(NAT))：

```
ipvsadm -a -t 192.168.0.30:80 -r 10.0.0.51:80 -m
```

```
ipvsadm -a -t 192.168.0.30:80 -r 10.0.0.52:80 -m
```

确认表

```
ipvsadm -l
```

| IP Virtual Server version 1.2.1 (size=4096) |                    |           |       |         |        |            |      |
|---------------------------------------------|--------------------|-----------|-------|---------|--------|------------|------|
| Prot                                        | LocalAddress:Port  | Scheduler | Flags | Forward | Weight | ActiveConn | InAc |
| tConn                                       |                    |           |       |         |        |            |      |
| TCP                                         | dlp.srv.world:http | wlc       |       |         |        |            |      |
| ->                                          | 10.0.0.51:http     |           |       | Masq    | 1      | 0          | 0    |
| ->                                          | 10.0.0.52:http     |           |       | Masq    | 1      | 0          | 0    |

配置完成，访问服务IP地址确保它正常工作：





下面是一些分配方式：

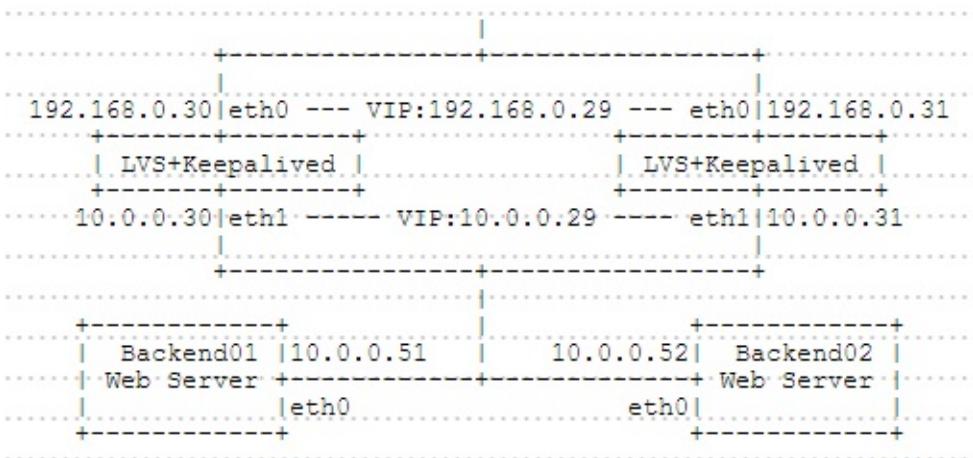
| 方式    | 描述                                                                                                                                                                                         |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rr    | Round Robin轮询。在可用的实体服务器之间平均分配作业。                                                                                                                                                           |
| wrr   | Weighted Round Robin加权轮询。将作业分配给实体服务器的权重。有较高权重的服务器首先接收新作业，且获得比低权重服务器更多的作业。具有相等权重的服务器获得的新作业平均分配。                                                                                             |
| lc    | Least-Connection最小连接。为有较少活动作业的实体服务器分配更多作业。                                                                                                                                                 |
| wlc   | Weighted Least-Connection加权最小连接。为有较少作业并相对于实体服务器权重 ( $C_i/W_i$ ) 的服务器分配更多作业。这是默认值。                                                                                                          |
| lblc  | Locality-Based Least-Connection基于位置的最小连接。如果服务器未过载并可用，则将同一IP地址的作业分配给同一服务器，否则将作业分配给作业较少的服务器，保留它以备将来分配。                                                                                       |
| lblcr | Locality-Based Least-Connection with Replication复制基于位置的最小连接。将同一IP地址的作业分配给为该IP地址设置的服务器中的最小连接节点。如果服务器集中的所有节点都已过载，则将在集群中选择有较少作业的节点，并将其添加到目标的服务器集中。如果服务器集尚未被修改指定的时间，则最多加载的节点将从服务器集中删除，以避免高度复制。 |
| dh    | Destination Hashing目标哈希。通过按照目标的IP地址查找静态分配的哈希表来将作业分配给服务器。                                                                                                                                   |
| sh    | Source Hashing源哈希。通过按照源的IP地址查找静态分配的哈希表来将作业分配给服务器。                                                                                                                                          |
| sed   | Shortest Expected Delay最短预期延迟。将最短预期延迟的传入作业分配给服务器。如果发送到第 <i>i</i> 个服务器，作业将经历的预期延迟是 $(C_i + 1) / U_i$ ，其中 $C_i$ 是第 <i>i</i> 个服务器上的作业数， $U_i$ 是第 <i>i</i> 个服务器的固定服务速率（权重）。                    |
| nq    | Never Queue从不队列。将传入的作业分配给空闲服务器（如果存在），而不是等待更快的服务器，如果所有服务器都忙，则采用最短预期延迟策略来分配作业。                                                                                                               |

## 11.4.2. LVS + Keepalived

LVS + Keepalived服务器本身的冗余配置

本例基于以下环境：

## 11.4. LVS



到LVS服务器eth0的HTTP数据包将使用NAT转发到Backend01和Backend02服务器。

首先在两个后端Web服务器上将默认网关更改为LVS的内部IP地址。（本例为 10.0.0.29）

安装ipvsadm和keepalived：

```
yum -y install ipvsadm keepalived
```

启用IP转发：

```
echo 'net.ipv4.ip_forward = 1' >> /etc/sysctl.conf
```

```
sysctl -p
```

```
touch /etc/sysconfig/ipvsadm
```

```
systemctl start ipvsadm
systemctl enable ipvsadm
```

配置Keepalived：

可以在两个后端服务器上配置相同的设置（除了有一个设置：“priority”优先级部分，在两个后端服务器上进行更改）：

```
mv /etc/keepalived/keepalived.conf
/etc/keepalived/keepalived.conf.org
```

编辑 /etc/keepalived/keepalived.conf 文件：

```
global_defs {
```

```

notification_email {
 root@dlp.srv.world
}
notification_email_from root@dlp.srv.world
smtp_server 127.0.0.1
smtp_connect_timeout 30
router_id LVS_Server
}

vrrp_instance VI_1 {
 state BACKUP
 # 监控接口
 interface eth0
 # 虚拟路由器ID
 virtual_router_id 51
 # 设置优先级（在每个服务器上更改此值）
 # （数字越大意味着优先级越高）
 priority 100
 nopreempt
 # VRRP发送间隔
 advert_int 1
 # Keepalived服务器之间的身份验证信息
 authentication {
 auth_type PASS
 auth_pass password
 }

 virtual_ipaddress {
 # 虚拟IP地址
 192.168.0.29 dev eth0
 10.0.0.29/24 dev eth1
 }
}

virtual_server 192.168.0.29 80 {
 # 监控间隔
 delay_loop 3
 # 分配方式
 lvs_sched rr
 # 路由方式
 lvs_method NAT
 protocol TCP

 # 后端服务器#1
}

```

## 11.4. LVS

```
real_server 10.0.0.51 80 {
 weight 1
 HTTP_GET {
 url {
 # 监控路径
 path /
 # 状态码为正常状态
 status_code 200
 }
 # 超时 (秒)
 connect_timeout 3
 }
}
后端服务器#2
real_server 10.0.0.52 80 {
 weight 1
 HTTP_GET {
 url {
 path /
 status_code 200
 }
 connect_timeout 3
 }
}
}
```

```
systemctl start keepalived
systemctl enable keepalived
```

配置完成，访问服务IP地址确保其正常工作。

# 12. 系统监控

- 12.1. OSQuery
  - 12.1.1. 安装OSQuery
  - 12.1.2. 计划监控
- 12.2. Munin
  - 12.2.1. 安装Munin
  - 12.2.2. 电子邮件通知
  - 12.2.3. 设置阈值
  - 12.2.4. 添加目标主机
  - 12.2.5. 添加目标项目
- 12.3. SysStat
  - 12.3.1. 安装SysStat
  - 12.3.2. 使用方法
- 12.4. Zabbix
  - 12.4.1. 安装Zabbix 3.2
  - 12.4.2. 初始设置
  - 12.4.3. 设置监控本机
  - 12.4.4. 设置电子邮件通知
  - 12.4.5. 添加目标主机
    - 12.4.5.1. CentOS7服务器
    - 12.4.5.2. Windows服务器
  - 12.4.6. 添加目标项目
- 12.5. MRTG
  - 12.5.1. 安装MRTG
  - 12.5.2. 获取CPU负载平均值
  - 12.5.3. 获取内存使用率
  - 12.5.4. 获取磁盘使用率
  - 12.5.5. 获取httpd进程
- 12.6. Cacti
  - 12.6.1. 安装Cacti
  - 12.6.2. Cacti初始设置
  - 12.6.3. 基本监控设置
  - 12.6.4. 电子邮件通知设置

- 12.6.5. 启用阈值插件
- 12.6.6. 设置阈值
- 12.6.7. 添加监控目标主机
- 12.7. Nagios
  - 12.7.1. 安装Nagios
  - 12.7.2. 电子邮件通知设置
  - 12.7.3. 设置阈值
  - 12.7.4. 添加监控目标项
  - 12.7.5. 添加监控目标主机
- 12.8. Monitorix
- 12.9. psacct

## 12.1. OSQuery

[OSQuery](#)是SQL驱动的分析和监控操作系统的工具，是操作系统分析框架，支持OS X和Linux系统。OSQuery能帮助监控和分析低水平的操作系统，提供更直观的性能监控。

### 12.1.1. 安装OSQuery

```
yum -y install https://osquery-
packages.s3.amazonaws.com/centos7/noarch/osquery-s3-centos7-repo-1-
0.0.noarch.rpm
```

```
yum -y install osquery
```

下面是OSQuery的一些基本操作示例（参阅[官方网站](#)查看所有表的详细信息）：

运行osquery shell：

```
osqueryi
```

```
osquery - being built, with love, at Facebook
~~~~~
Using a virtual database. Need help, type '.help'
osquery>

# 显示操作系统版本的所有表的列
osquery> select * from os_version;
+-----+-----+-----+-----+
| name | major | minor | patch | build |
+-----+-----+-----+-----+
| CentOS Linux | 7 | 2 | 1511 | |
+-----+-----+-----+-----+


# 显示系统信息的一些表的列
osquery> select hostname, cpu_brand, hardware_vendor, hardware_m
odel from system_info;
+-----+-----+-----+
| hostname | cpu_brand | ha
rdware_vendor | hardware_model |
+-----+-----+-----+
```

## 12.1. OSQuery

```
+-----+-----+
-----+-----+
| dlp.srv.world | Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz | Re
d Hat          | KVM           |
-----+-----+
-----+-----+-----+-----+
# 显示一些表的列，并指定大于1000的UID的用户信息
osquery> select uid, gid, username, shell from users where uid >
= 1000;
+-----+-----+-----+-----+
| uid   | gid   | username | shell      |
+-----+-----+-----+-----+
| 1000  | 1000  | cent     | /bin/bash  |
| 1001  | 1001  | redhat   | /bin/bash  |
| 1002  | 1002  | ubuntu   | /bin/bash  |
+-----+-----+-----+-----+
# 显示CPU Time的所有表列
osquery> select * from cpu_time;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
| core | user | nice | system | idle    | iowait | irq  | softirq
| steal | guest | guest_nice |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
| 0    | 870   | 0     | 597    | 298134 | 4      | 0    | 11
| 8    | 0     | 0     | 0       |         |         |         |
| 1    | 3717  | 0     | 1164   | 294858 | 10     | 0    | 3
| 1    | 0     | 0     | 0       |         |         |         |
| 2    | 1189  | 0     | 873    | 297573 | 13     | 0    | 0
| 33   | 0     | 0     | 0       |         |         |         |
| 3    | 1150  | 0     | 1233   | 297503 | 6      | 0    | 0
| 2    | 0     | 0     | 0       |         |         |         |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
# 按Ctrl+D退出shell
osquery>
```

## 12.1.2. 计划监控

配置计划并启动守护进程定期输出日志。

在OSQuery配置文件中如下设置：

编辑 `/etc/osquery/osquery.conf` 文件：

```
{  
    "options": {  
        // 选择osquery配置插件 (filesystem是默认)  
        "config_plugin": "filesystem",  
  
        // 选择osquery日志插件 (filesystem是默认)  
        "logger_plugin": "filesystem",  
  
        // 日志目录的路径  
        "logger_path": "/var/log/osquery",  
  
        // 守护进程的PID文件  
        "pidfile": "/var/osquery/osquery.pidfile",  
  
        // 并发查询的线程数  
        "worker_threads": "2",  
  
        // 启用计划分析  
        // 如果在"schedule"部分中添加查询"select * from osquery_schedule"，可以记录性能  
        "enable_monitor": "true"  
    },  
  
    "schedule": {  
        // 示例：每300秒获取CPU Time  
        "cpu_time": {  
            "query": "SELECT * FROM cpu_time;",  
            "interval": 300  
        },  
        // 示例：每小时获取resolv.conf的设置  
        "dns_resolvers": {  
            "query": "SELECT * FROM dns_resolvers;",  
            "interval": 3600  
        }  
    }  
}
```

```
},  
  
"packs": {  
    // 可以包含其他配置文件  
    "hardware-monitoring": "/usr/share/osquery/packs/hardware-m  
onitoring.conf"  
}  
}
```

```
systemctl start osqueryd  
systemctl enable osqueryd
```

查询日志在文件中输出如下（但是，只有当前结果和先前结果之间存在一些差异时，结果才会记录到文件中）：

```
cat /var/log/osquery/osqueryd.results.log
```





## 12.2. Munin

Munin 可监控核心系统资源，包括内存、磁盘、CPU 占用、服务器应用如 MySQL、Apache 和 Squid 等。

### 12.2.1. 安装 Munin

先 [安装 Apache httpd](#)。

安装 Munin 服务器，并安装 Munin 代理以监控 Munin 服务器自身：

```
yum --enablerepo=epel -y install munin munin-node # 从 EPEL 安装
```

配置 Munin：

编辑 `/etc/munin/munin.conf` 文件：

```
# 添加访问权限  
Order Deny,Allow  
Deny from all  
Allow from 127.0.0.1 10.0.0.0/24
```

```
systemctl restart httpd
```

```
htpasswd -c /etc/munin/munin-htpasswd cent # 添加用户：使用 -c 创建一个新文件
```

```
New password: # 设置密码  
Re-type new password: # 确认密码  
Adding password for user cent
```

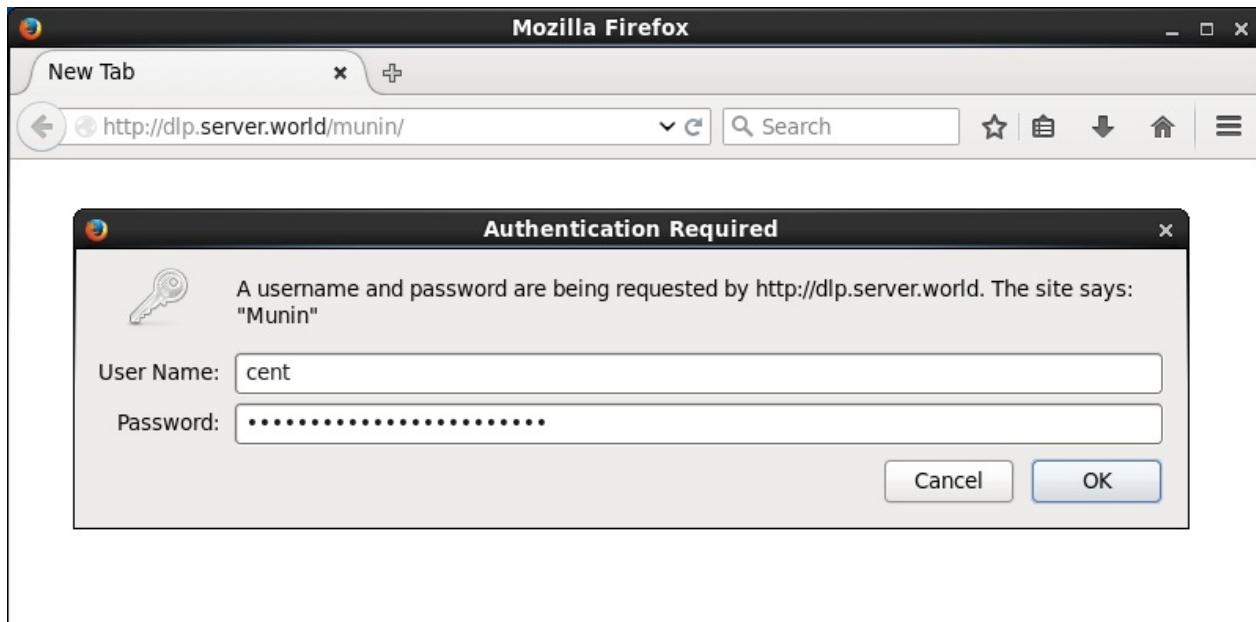
将 Munin 节点配置为监控目标：

编辑 `/etc/munin/munin-node.conf` 文件：

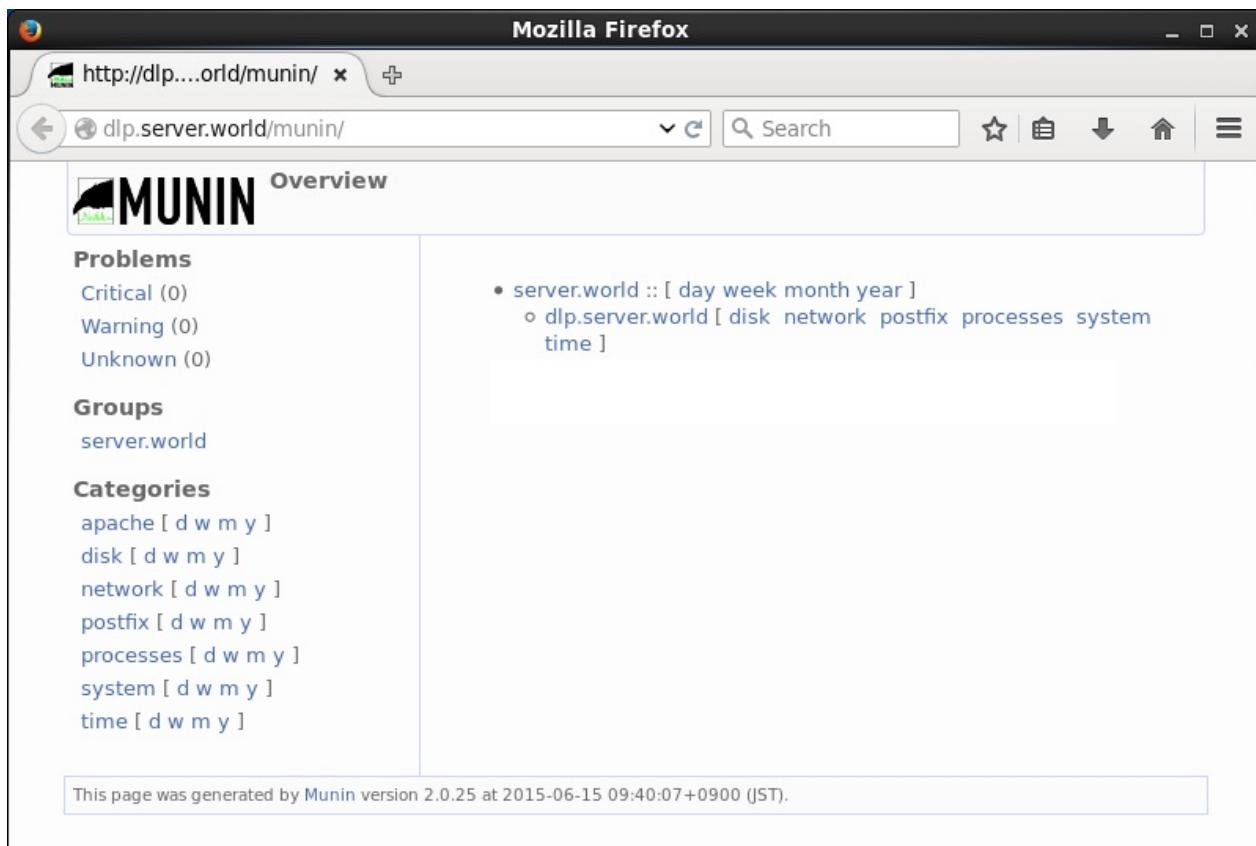
```
# 更改自己的主机名  
host_name dlp.srv.world
```

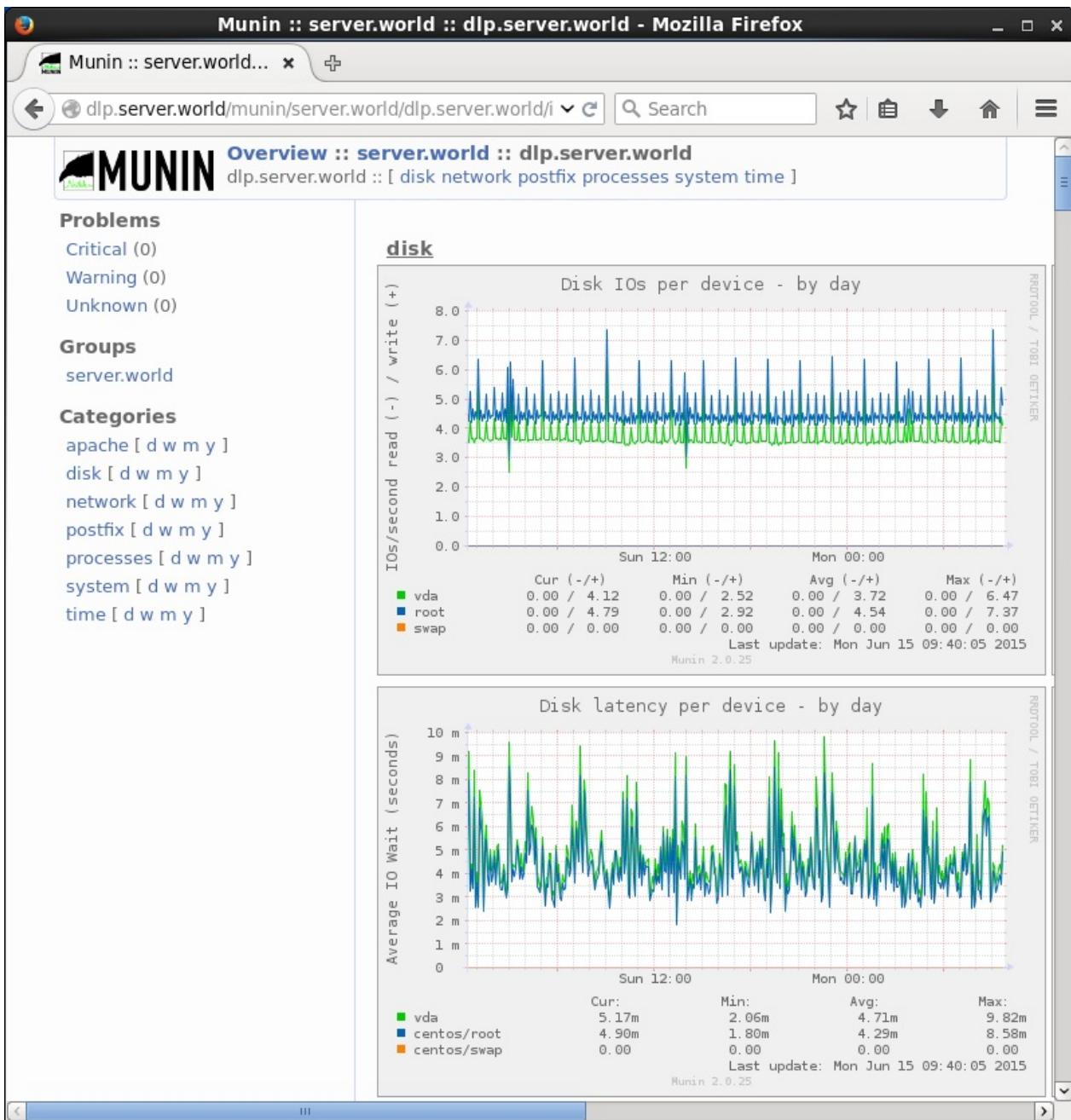
```
systemctl start munin-node  
systemctl enable munin-node
```

从配置中允许的网络中的客户端访问 `http://(Munin服务器的主机名或IP地址)/munin/`，这里需要验证，输入在上面设置的用户名和密码，然后继续：



验证成功后，将显示主页。可以在图表上点击主机名：





### 12.2.2. 电子邮件通知

配置通知设置以发送通知电子邮件。

配置为将通知发送到root帐户：

编辑 `/etc/munin/munin.conf` 文件：

```
# 添加
contact.email.command mail -s "Munin-notification for ${var:group}::${var:host}" root
```

```
su - munin --shell=/bin/bash -c "/usr/share/munin/munin-limits --  
contact email --force" # 尝试强制发送通知
```

电子邮件发送如下：

```
From munin@dlp.srv.world Fri Feb 18 20:02:27 2015

srv.world :: dlp.srv.world :: Disk usage in percent
    OKs: /boot is 26.23, / is 8.94, /dev/shm is 0.00.

srv.world :: dlp.srv.world :: Inode usage in percent
    OKs: /boot is 0.05, / is 3.90, /dev/shm is 0.00.

srv.world :: dlp.srv.world :: File table usage
    OKs: open files is 864.00.

srv.world :: dlp.srv.world :: Disk latency per device :: Average
    latency f
or /dev/vda
    OKs: Read IO Wait time is 0.01, Write IO Wait time is 0.
14.

srv.world :: dlp.srv.world :: Disk latency per device :: Average
    latency f
or /dev/VolGroup/lv_swap
    OKs: Read IO Wait time is 0.00, Write IO Wait time is 0.
00.

srv.world :: dlp.srv.world :: Disk latency per device :: Average
    latency f
or /dev/VolGroup/lv_root
    OKs: Read IO Wait time is 0.01, Write IO Wait time is 0.
12.

srv.world :: dlp.srv.world :: eth0 errors
    OKs: errors is 0.00, errors is 0.00.
```

如果要将通知以文件的形式记录为日志，按以下步骤操作：

编辑 `/etc/munin/munin.conf` 文件：

```
# 添加
contact.log.command tee -a /var/log/munin/alert.log
```

## 12.2.3. 设置阈值

监控目标项目被默认设置为插件，并在某些插件中定义阈值，如果要向插件添加更多阈值，按如下所示进行配置。

可以如下确认插件：

```
ls /etc/munin/plugins # 当前启用的插件位于下面的目录
```

|                    |                |                     |         |
|--------------------|----------------|---------------------|---------|
| cpu                | fw_packets     | ntp_kernel_pll_freq | process |
| es                 |                |                     |         |
| df                 | if_err_eth0    | ntp_kernel_pll_off  | proc_pr |
| i                  |                |                     |         |
| df_inode           | if_eth0        | ntp_offset          | swap    |
| diskstats          | interrupts     | ntp_states          | threads |
| entropy            | irqstats       | open_files          | uptime  |
| forks              | load           | open_inodes         | users   |
| fw_conntrack       | memory         | postfix_mailqueue   | vmstat  |
| fw_forwarded_local | ntp_kernel_err | postfix_mailvolume  |         |

```
ls /usr/share/munin/plugins # 安装的插件位于下面的目录
```

|                  |             |
|------------------|-------------|
| acpi             | nvidia_     |
| amavis           | open_files  |
| apache_acceses   | open_inodes |
| apache_processes | openvpn     |
| apache_volume    | perdition   |
| ....             |             |
| ....             |             |

可以显示每个插件的当前值：

```
munin-run cpu # 显示cpu插件的值
```

```
user.value 4262
nice.value 680
system.value 1933
idle.value 1069496
iowait.value 1891
irq.value 0
softirq.value 52
steal.value 208
guest.value 0
```

```
munin-run df # 显示df插件的值
```

```
_dev_mapper_centos_root.value 4.34168354184342
_dev_vda1.value 25.3611960958576
```

在 `munin.conf` 中设置阈值的格式： [插件名称].[字段名称].[warning(警告) | critical(紧急)] 最小值:最大值

“字段名称”只是执行 `munin-run` 命令时显示的字段名称。“最小值”或“最大值”可以省略。

编辑 `/etc/munin/munin.conf` 文件：

```
# 在目标节点上设置阈值
[dlp.srv.world]
address 127.0.0.1
use_node_name yes
# 在cpu插件中为user字段设置80%为warning和90%为critical
cpu.user.warning :80
cpu.user.critical :90
# 在df插件中设置root分区字段的80%为warning和90%为critical
df._dev_mapper_VolGroup_lv_root.warning :80
df._dev_mapper_VolGroup_lv_root.critical :90
```

如果配置了通知设置，并且值超过阈值，则电子邮件将按如下方式发送：

```
From munin@dlp.srv.world Fri Jun 12 14:35:07 2015
Return-Path: <munin@dlp.srv.world>
X-Original-To: root
Delivered-To: root@dlp.srv.world
Date: Fri, 12 Jun 2015 14:35:07 +0900
To: root@dlp.srv.world
Subject: Munin-notification for srv.world::dlp.srv.world
User-Agent: Heirloom mailx 12.5 7/5/10
Content-Type: text/plain; charset=us-ascii
From: munin@dlp.srv.world (Munin user)
Status: R

srv.world :: dlp.srv.world :: CPU usage
CRITICALs: user is 100.38 (outside range [:80]).
```

### 12.2.4. 添加目标主机

可以监控其他主机。例如，将名为“node01”的主机配置为监控目标。

在新的监控目标上安装“munin-node”：

```
yum --enablerepo=epel -y install munin-node # 从EPEL安装
```

编辑 `/etc/munin/munin-node.conf` 文件：

```
# 更改自己的主机名
host_name node01.srv.world
# 添加连接监控的权限（指定Munin服务器）
allow ^127\.0\.0\.1$
allow ^::1$
allow ^10\.0\.0\.30$
```

```
systemctl start munin-node
systemctl enable munin-node
```

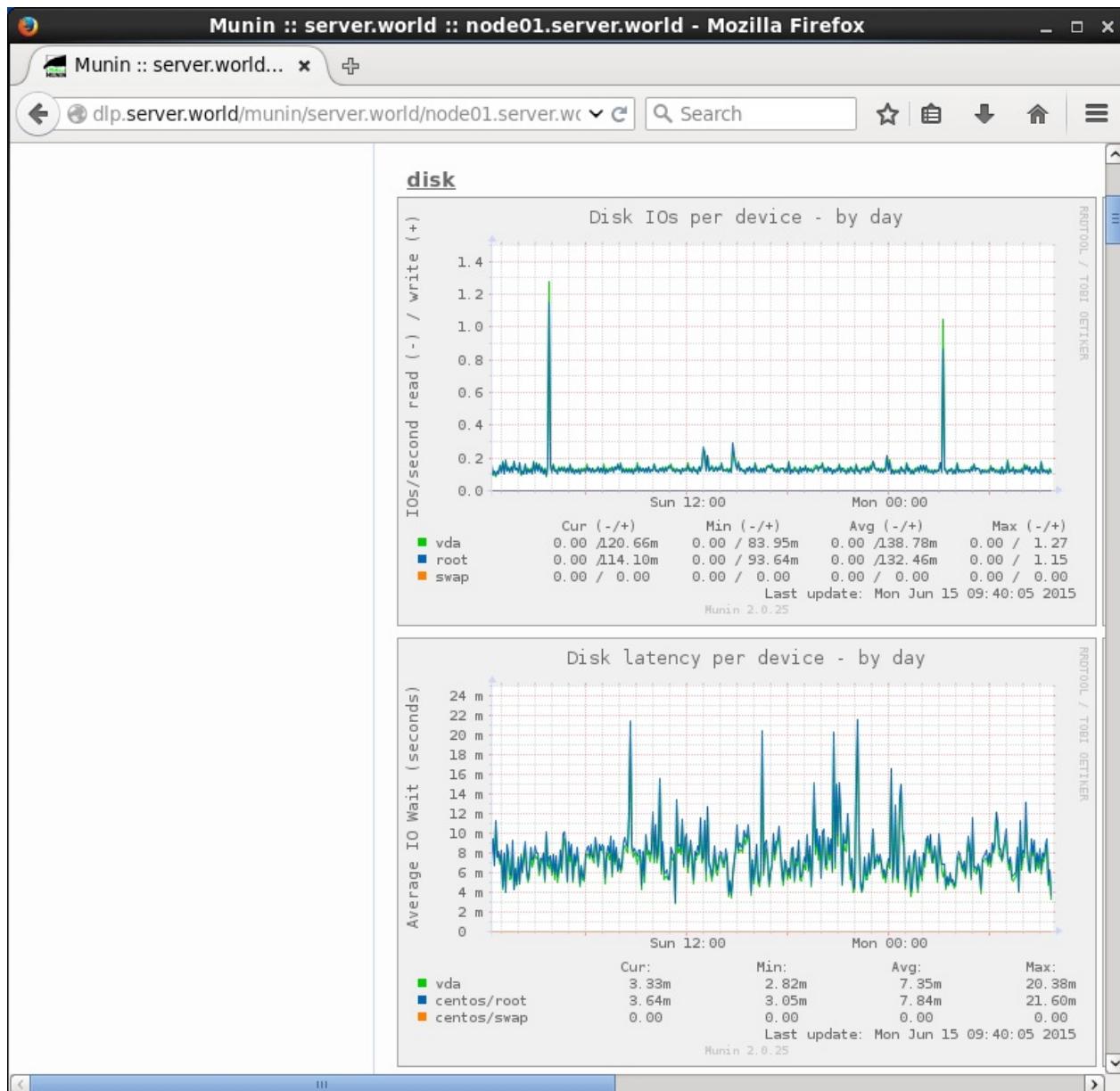
配置**Munin**服务器设置新的目标主机：

编辑 `/etc/munin/munin.conf` 文件：

```
# 添加以下内容到最后  
[node01.srv.world]  
address 10.0.0.51  
use_node_name yes
```

登录到Munin管理网站，几分钟后确认显示新的目标主机：

The screenshot shows a Mozilla Firefox browser window with the title "Mozilla Firefox". The address bar displays "http://dlp....orld/munin/" and the URL "dlp.server.world/munin/". The main content area is titled "Overview" and features the "MUNIN" logo. On the left side, there are three sections: "Problems" (Critical: 0, Warning: 0, Unknown: 0), "Groups" (server.world), and "Categories" (apache [ d w m y ], disk [ d w m y ], network [ d w m y ], postfix [ d w m y ], processes [ d w m y ], system [ d w m y ], time [ d w m y ]). To the right, a list of monitored hosts is shown under the heading "server.world :: [ day week month year ]": "dlp.server.world [ disk network postfix processes system time ]" and "node01.server.world [ apache disk network postfix processes system time ]". At the bottom of the page, a footer note states: "This page was generated by Munin version 2.0.25 at 2015-06-15 09:40:07+0900 (JST)."



### 12.2.5. 添加目标项目

启用插件以添加新的监控目标项目。

插件位于[这里](#)，也可以添加[第三方](#)或原创的插件。

本例添加一个目标项目以启用Apache访问插件。

在目标主机上配置您要启用的新插件：

```
ln -s /usr/share/munin/plugins/apache_accesses  
/etc/munin/plugins/apache_accesses
```

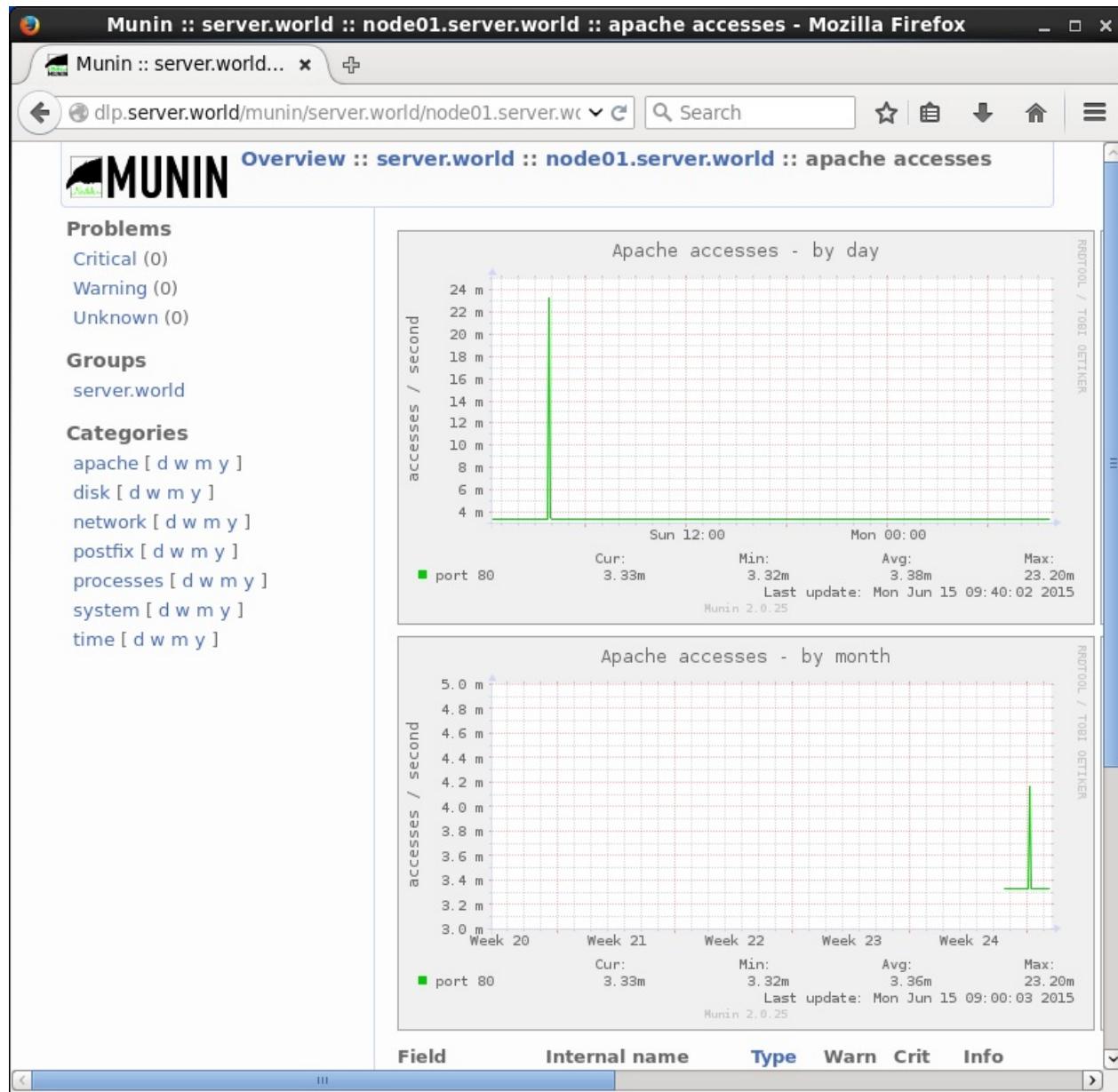
编辑 `/etc/httpd/conf.d/server_status.conf` 文件：

```
ExtendedStatus On
```

```
<Location /server-status>
    SetHandler server-status
    Require local
</Location>
```

```
systemctl restart httpd
systemctl restart munin-node
```

登录到Munin管理网站，几分钟后确认显示新的目标项目：





## 12.3. SysStat

SysStat 包含监控系统性能及效率的一组工具，这些工具对于我们收集系统性能数据，比如CPU使用率、硬盘和网络吞吐数据等。

### 12.3.1. 安装 SysStat

```
yum -y install sysstat
```

```
systemctl start sysstat  
systemctl enable sysstat
```

日志记录由 cron 的设置执行，如下所示：

- 使用 `/usr/lib64/sa/sa1` 命令，每10分钟将日志存储在 `/var/log/sa/sa**`
- 使用 `/usr/lib64/sa/sa2` 命令，将一天的统计信息在每天23:53生成到 `/var/log/sa/sar**`

```
cat /etc/cron.d/sysstat
```

```
# Run system activity accounting tool every 10 minutes  
*/10 * * * * root /usr/lib64/sa/sa1 1 1  
# 0 * * * * root /usr/lib64/sa/sa1 600 6 &  
# Generate a daily summary of process accounting at 23:53  
53 23 * * * root /usr/lib64/sa/sa2 -A
```

如果要更改 SysStat 的某些设置，编辑配置文件 `/etc/sysconfig/sysstat`，如下所示：

```
[root@dlp ~]# vi /etc/sysconfig/sysstat
# sysstat-10.1.5 configuration file.

# How long to keep log files (in days).
# If value is greater than 28, then log files are kept in
# multiple directories, one for each month.
HISTORY=28

# Compress (using gzip or bzip2) sa and sar files older than (in
# days):
COMPRESSAFTER=31

# Parameters for the system activity data collector (see sadc ma
nual page)
# which are used for the generation of log files.
# *注
SADC_OPTIONS="-S DISK"

# *注：有效选项
INT      -> 系统中断
DISK     -> 块设备
SNMP     -> SNMP统计
IPV6     -> IPV6统计
POWER    -> 电源管理统计
ALL      -> 以上所有
XDISK   -> DISK + 分区统计
XALL    -> 以上所有 (ALL + XDISK)
```

## 12.3.2. 使用方法

输入 `sar` 命令并指定如下所示的选项，可以显示日志（更多细节参考 `man sar`）：

| 选项       | 描述         |
|----------|------------|
| -u       | CPU利用率统计   |
| -r       | 内存利用率统计    |
| -b       | I/O和传输速率统计 |
| -B       | 分页统计       |
| -d       | 每个块设备的活动   |
| -n [关键词] | 网络统计       |
| -q       | 队列长度和负载平均值 |
| -A       | 全部显示       |

显示当天的统计报告：

```
sar -u # CPU
```

|             |       |       |       |         |         |   |
|-------------|-------|-------|-------|---------|---------|---|
| 01:10:01 AM | CPU   | %user | %nice | %system | %iowait | % |
| steal       | %idle |       |       |         |         |   |
| 10:20:01 AM | all   | 0.90  | 0.00  | 0.23    | 1.19    |   |
| 0.03        | 97.65 |       |       |         |         |   |
| ...         |       |       |       |         |         |   |
| ...         |       |       |       |         |         |   |
| 01:30:01 PM | all   | 0.04  | 0.00  | 0.06    | 0.14    |   |
| 0.04        | 99.72 |       |       |         |         |   |
| Average:    | all   | 3.60  | 0.00  | 0.05    | 0.21    |   |
| 0.01        | 96.13 |       |       |         |         |   |

```
sar -r # 内存
```

|             |           |           |          |           |          |   |
|-------------|-----------|-----------|----------|-----------|----------|---|
| 01:10:01 AM | kbmemfree | kbmemused | %memused | kbbuffers | kbcached | k |
| bcommit     | %commit   |           |          |           |          |   |
| 10:20:01 AM | 3681144   | 241544    | 6.16     | 10744     | 138392   |   |
| 83984       | 1.04      |           |          |           |          |   |
| ...         |           |           |          |           |          |   |
| ...         |           |           |          |           |          |   |
| 01:40:01 PM | 3663328   | 259360    | 6.61     | 14752     | 145988   |   |
| 87996       | 1.09      |           |          |           |          |   |
| Average:    | 3666930   | 255758    | 6.52     | 13204     | 144710   |   |
| 88273       | 1.10      |           |          |           |          |   |

## 12.3. SysStat

```
sar -b # I/O
```

| 01:10:01 AM | tps   | rtps | wtps  | bread/s | bwrtn/s |
|-------------|-------|------|-------|---------|---------|
| 10:20:01 AM | 45.21 | 9.35 | 35.87 | 257.22  | 409.24  |
| ...         |       |      |       |         |         |
| ...         |       |      |       |         |         |
| 01:40:01 PM | 0.52  | 0.00 | 0.52  | 0.00    | 5.43    |
| Average:    | 3.11  | 0.49 | 2.62  | 14.55   | 29.32   |

```
sar -n DEV # 网络
```

| 01:10:01 AM | IFACE   | rxpck/s  | txpck/s | rxkB/s | txkB/s |
|-------------|---------|----------|---------|--------|--------|
| rxcmp/s     | txcmp/s | rxmcst/s |         |        |        |
| 10:20:01 AM | lo      | 1.04     | 1.04    | 0.09   | 0.09   |
| 0.00        | 0.00    | 0.00     |         |        |        |
| 10:20:01 AM | eth0    | 7.48     | 4.48    | 10.06  | 0.30   |
| 0.00        | 0.00    | 0.00     |         |        |        |
| ...         |         |          |         |        |        |
| ...         |         |          |         |        |        |
| 01:40:01 PM | lo      | 0.01     | 0.01    | 0.00   | 0.00   |
| 0.00        | 0.00    | 0.00     |         |        |        |
| 01:40:01 PM | eth0    | 0.05     | 0.04    | 0.00   | 0.00   |
| 0.00        | 0.00    | 0.00     |         |        |        |
| Average:    | lo      | 0.08     | 0.08    | 0.01   | 0.01   |
| 0.00        | 0.00    | 0.00     |         |        |        |
| Average:    | eth0    | 0.54     | 0.38    | 0.51   | 0.03   |
| 0.00        | 0.00    | 0.00     |         |        |        |

```
sar -q # 负载平均值
```

| 01:10:01 AM | runq-sz | plist-sz | ldavg-1 | ldavg-5 | ldavg-15 |
|-------------|---------|----------|---------|---------|----------|
| 10:20:01 AM | 1       | 99       | 0.04    | 0.07    | 0.04     |
| ...         |         |          |         |         |          |
| ...         |         |          |         |         |          |
| 01:30:01 PM | 1       | 104      | 0.00    | 0.00    | 0.00     |
| 01:40:01 PM | 1       | 101      | 0.00    | 0.00    | 0.00     |
| Average:    | 1       | 102      | 0.11    | 0.08    | 0.06     |

## 12.3. SysStat

日志文件存储在 `/var/log/sa directory`，目录下通过指定日志文件显示过去的统计：

```
sar -A -f /var/log/sa/sa05 # 通过日志文件显示所有统计
```

| 01:10:01 AM | CPU  | %usr  | %nice  | %sys  | %iowait | % |
|-------------|------|-------|--------|-------|---------|---|
| steal       | %irq | %soft | %guest | %idle |         |   |
| 10:20:01 AM | all  | 0.90  | 0.00   | 0.19  | 1.19    |   |
| 0.03        | 0.04 | 0.00  | 0.00   | 97.65 |         |   |
| 10:20:01 AM | 0    | 0.96  | 0.00   | 0.19  | 1.27    |   |
| 0.03        | 0.04 | 0.00  | 0.00   | 97.51 |         |   |
| ...         |      |       |        |       |         |   |
| ...         |      |       |        |       |         |   |
| 01:50:01 PM | 125  | 3     | 4      | 0     | 0       |   |
| 0           |      |       |        |       |         |   |
| 02:00:01 PM | 125  | 3     | 4      | 0     | 0       |   |
| 0           |      |       |        |       |         |   |
| Average:    | 122  | 3     | 4      | 0     | 0       |   |
| 0           |      |       |        |       |         |   |

```
sar -q -s 11:00:00 -e 12:00:00 -f /var/log/sa/sa05 # 通过日志文件显示指定时间的负载平均值
```

| 11:00:01 AM | runq-sz | plist-sz | ldavg-1 | ldavg-5 | ldavg-15 |
|-------------|---------|----------|---------|---------|----------|
| 11:10:01 AM | 1       | 103      | 0.01    | 0.03    | 0.14     |
| 11:20:01 AM | 1       | 102      | 0.01    | 0.01    | 0.06     |
| 11:30:01 AM | 1       | 102      | 0.00    | 0.00    | 0.01     |
| 11:40:01 AM | 1       | 101      | 0.00    | 0.00    | 0.00     |
| 11:50:01 AM | 1       | 101      | 0.01    | 0.01    | 0.00     |
| Average:    | 1       | 102      | 0.01    | 0.01    | 0.04     |

显示当前统计：

```
sar -u 1 3 # 每秒3次显示CPU利用率
```

## 12.3. SysStat

| 01:51:34 AM | CPU   | %user  | %nice | %system | %iowait | % |
|-------------|-------|--------|-------|---------|---------|---|
| steal       | %idle |        |       |         |         |   |
| 01:51:35 AM | all   | 0.00   | 0.00  | 0.50    | 0.00    |   |
| 0.00        |       | 99.50  |       |         |         |   |
| 01:51:36 AM | all   | 0.00   | 0.00  | 0.00    | 0.00    |   |
| 0.50        |       | 99.50  |       |         |         |   |
| 01:51:37 AM | all   | 0.00   | 0.00  | 0.00    | 0.00    |   |
| 0.00        |       | 100.00 |       |         |         |   |
| Average:    | all   | 0.00   | 0.00  | 0.17    | 0.00    |   |
| 0.17        |       | 99.67  |       |         |         |   |

```
sar -b -n DEV 2 5 # 每2秒5次显示I/O和网络
```

## 12.3. SysStat

| 01:54:11 AM | tps     | rtps     | wtps    | bread/s | bwrtn/s |
|-------------|---------|----------|---------|---------|---------|
| 01:54:13 AM | 0.00    | 0.00     | 0.00    | 0.00    | 0.00    |
| 01:54:11 AM | IFACE   | rxpck/s  | txpck/s | rxkB/s  | txkB/s  |
| rxcmp/s     | txcmp/s | rxmcst/s |         |         |         |
| 01:54:13 AM | lo      | 0.00     | 0.00    | 0.00    | 0.00    |
| 0.00        | 0.00    | 0.00     |         |         |         |
| 01:54:13 AM | eth0    | 0.00     | 0.00    | 0.00    | 0.00    |
| 0.00        | 0.00    | 0.00     |         |         |         |
| ...         |         |          |         |         |         |
| ...         |         |          |         |         |         |
| 01:54:19 AM | tps     | rtps     | wtps    | bread/s | bwrtn/s |
| 01:54:21 AM | 0.00    | 0.00     | 0.00    | 0.00    | 0.00    |
| 01:54:19 AM | IFACE   | rxpck/s  | txpck/s | rxkB/s  | txkB/s  |
| rxcmp/s     | txcmp/s | rxmcst/s |         |         |         |
| 01:54:21 AM | lo      | 0.00     | 0.00    | 0.00    | 0.00    |
| 0.00        | 0.00    | 0.00     |         |         |         |
| 01:54:21 AM | eth0    | 0.00     | 0.00    | 0.00    | 0.00    |
| 0.00        | 0.00    | 0.00     |         |         |         |
| Average:    | tps     | rtps     | wtps    | bread/s | bwrtn/s |
| Average:    | 0.70    | 0.00     | 0.70    | 0.00    | 4.80    |
| Average:    | IFACE   | rxpck/s  | txpck/s | rxkB/s  | txkB/s  |
| rxcmp/s     | txcmp/s | rxmcst/s |         |         |         |
| Average:    | lo      | 0.00     | 0.00    | 0.00    | 0.00    |
| 0.00        | 0.00    | 0.00     |         |         |         |
| Average:    | eth0    | 0.20     | 0.00    | 0.01    | 0.00    |
| 0.00        | 0.00    | 0.00     |         |         |         |

SysStat软件包也包括 sar 命令外的其他命令，因此可以显示当前统计数据如下：

```
iostat -mx -d 2 # 每2秒以兆字节显示I/O
```

## 12.3. SysStat

| Device: | rrqm/s    | wrqm/s   | r/s   | w/s   | rMB/s | wMB/s |
|---------|-----------|----------|-------|-------|-------|-------|
| s       | avgqrq-sz | avgqu-sz | await | svctm | %util |       |
| vda     | 0.11      | 1.05     | 0.43  | 0.63  | 0.01  | 0.0   |
| 1       | 28.44     | 0.05     | 51.01 | 9.29  | 0.99  |       |
| dm-0    |           | 0.00     | 0.00  | 0.48  | 1.65  | 0.01  |
| 1       | 14.05     | 0.20     | 95.25 | 4.64  | 0.99  |       |
| dm-1    |           | 0.00     | 0.00  | 0.02  | 0.00  | 0.00  |
| 0       | 8.00      | 0.00     | 1.14  | 0.52  | 0.00  |       |
| ...     |           |          |       |       |       |       |
| ...     |           |          |       |       |       |       |

```
mpstat -P ALL 2 3 # 每2秒3次显示所有CPU利用率
```

| 02:28:57 PM | CPU    | %usr   | %nice  | %sys | %iowait | %irq | %soft |
|-------------|--------|--------|--------|------|---------|------|-------|
|             | %steal | %guest | %idle  |      |         |      |       |
| 02:28:59 PM | all    | 0.00   | 0.00   | 0.00 | 0.00    | 0.00 | 0.00  |
|             | 0.00   | 0.00   | 100.00 |      |         |      |       |
| 02:28:59 PM | 0      | 0.00   | 0.00   | 0.00 | 0.00    | 0.50 | 0.00  |
|             | 0.00   | 0.00   | 99.50  |      |         |      |       |
| 02:28:59 PM | 1      | 0.00   | 0.00   | 0.00 | 0.00    | 0.00 | 0.00  |
|             | 0.00   | 0.00   | 100.00 |      |         |      |       |
| ...         |        |        |        |      |         |      |       |
| ...         |        |        |        |      |         |      |       |

```
pidstat -r -p 1202 1 3 # 每秒3次显示一个进程的存储器使用
```

| 02:34:07 PM | PID  | minflt/s | majflt/s | VSZ    | RSS  | %MEM |
|-------------|------|----------|----------|--------|------|------|
| Command     |      |          |          |        |      |      |
| 02:34:08 PM | 1202 | 0.00     | 0.00     | 175360 | 2456 | 0.06 |
| httpd       |      |          |          |        |      |      |
| 02:34:09 PM | 1202 | 0.00     | 0.00     | 175360 | 2456 | 0.06 |
| httpd       |      |          |          |        |      |      |
| 02:34:10 PM | 1202 | 0.00     | 0.00     | 175360 | 2456 | 0.06 |
| httpd       |      |          |          |        |      |      |
| Average:    | 1202 | 0.00     | 0.00     | 175360 | 2456 | 0.06 |
| httpd       |      |          |          |        |      |      |
| ...         |      |          |          |        |      |      |
| ...         |      |          |          |        |      |      |

## 12.3. SysStat

```
cifsiostat -m 1 3 # 每秒3次以兆字节显示CIFS统计
```

| Filesystem:      | ps/s | fo/s | fc/s | rB/s | wB/s  | rops/s | wo |
|------------------|------|------|------|------|-------|--------|----|
| \\10.0.0.100\tmp | 0.00 | 0.00 | 0.00 | 0.00 | 20.93 | 0.00   |    |
|                  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00  | 0.00   |    |
| ...              |      |      |      |      |       |        |    |
| ...              |      |      |      |      |       |        |    |

## 12.4. Zabbix

Zabbix是一个基于WEB界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案。

### 12.4.1. 安装Zabbix 3.2

先安装Apache httpd，PHP和MariaDB。

```
yum -y install php-mysql php-gd php-xml php-bcmath # 安装一些其他必需的软件包
```

```
yum -y install  
http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-3.2-  
1.el7.noarch.rpm # 安装Zabbix存储库
```

```
yum -y install zabbix-get zabbix-server-mysql zabbix-web-mysql  
zabbix-agent # 安装Zabbix服务器
```

为Zabbix创建数据库：

```
mysql -u root -p
```

```
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 10  
Server version: 5.5.50-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> create database zabbix;  
Query OK, 1 row affected (0.00 sec)  
  
MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@'localhost' identified by 'password';  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@'%'  
    identified by 'password';  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> exit  
Bye
```

```
cd /usr/share/doc/zabbix-server-mysql-*/  
gunzip create.sql.gz  
mysql -u root -p zabbix < create.sql
```

```
Enter password:
```

配置并启动Zabbix服务器：

编辑 `/etc/zabbix/zabbix_server.conf` 文件：

```
# 添加  
DBHost=localhost  
  
# 添加Zabbix数据库密码  
DBPassword=password
```

```
systemctl start zabbix-server  
systemctl enable zabbix-server
```

配置并启动Zabbix代理以监控Zabbix服务器本身：

编辑 `/etc/zabbix/zabbix_agentd.conf` 文件：

```
# 指定Zabbix服务器  
Server=127.0.0.1  
  
# 指定Zabbix服务器  
ServerActive=127.0.0.1  
  
# 更改自己的主机名  
Hostname=d1p.srv.world
```

```
systemctl start zabbix-agent  
systemctl enable zabbix-agent
```

编辑 `/etc/httpd/conf.d/zabbix.conf` 文件，更改httpd设置如下：

```
# 为Zabbix Web前端添加访问权限  
#Require all granted  
Require ip 127.0.0.1 10.0.0.0/24  
  
# 取消注释并更改为自己的时区  
php_value date.timezone Asia/Shanghai
```

```
systemctl restart httpd
```

如果启用了SELinux，更改布尔设置：

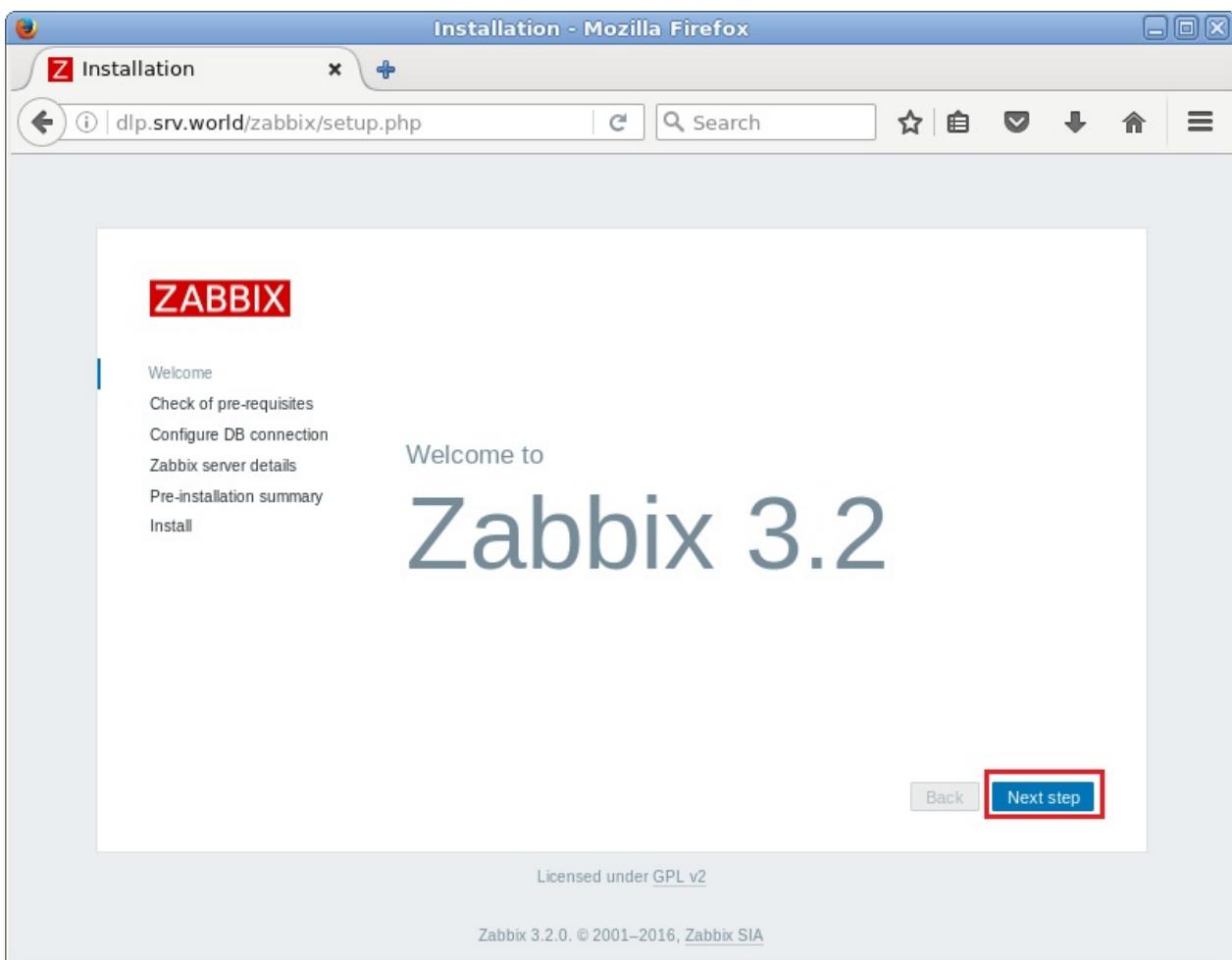
```
setsebool -P httpd_can_connect_zabbix on
```

firewalld防火墙规则：

```
firewall-cmd --add-service={http,https} --permanent  
firewall-cmd --add-port={10051/tcp,10050/tcp} --permanent  
firewall-cmd --reload
```

### 12.4.2. 初始设置

从Zabbix服务器允许的网络中的客户端访问 `http://(Zabbix服务器的主机名或IP地址)/zabbix/`，显示Zabbix起始页，点击“Next step”继续：

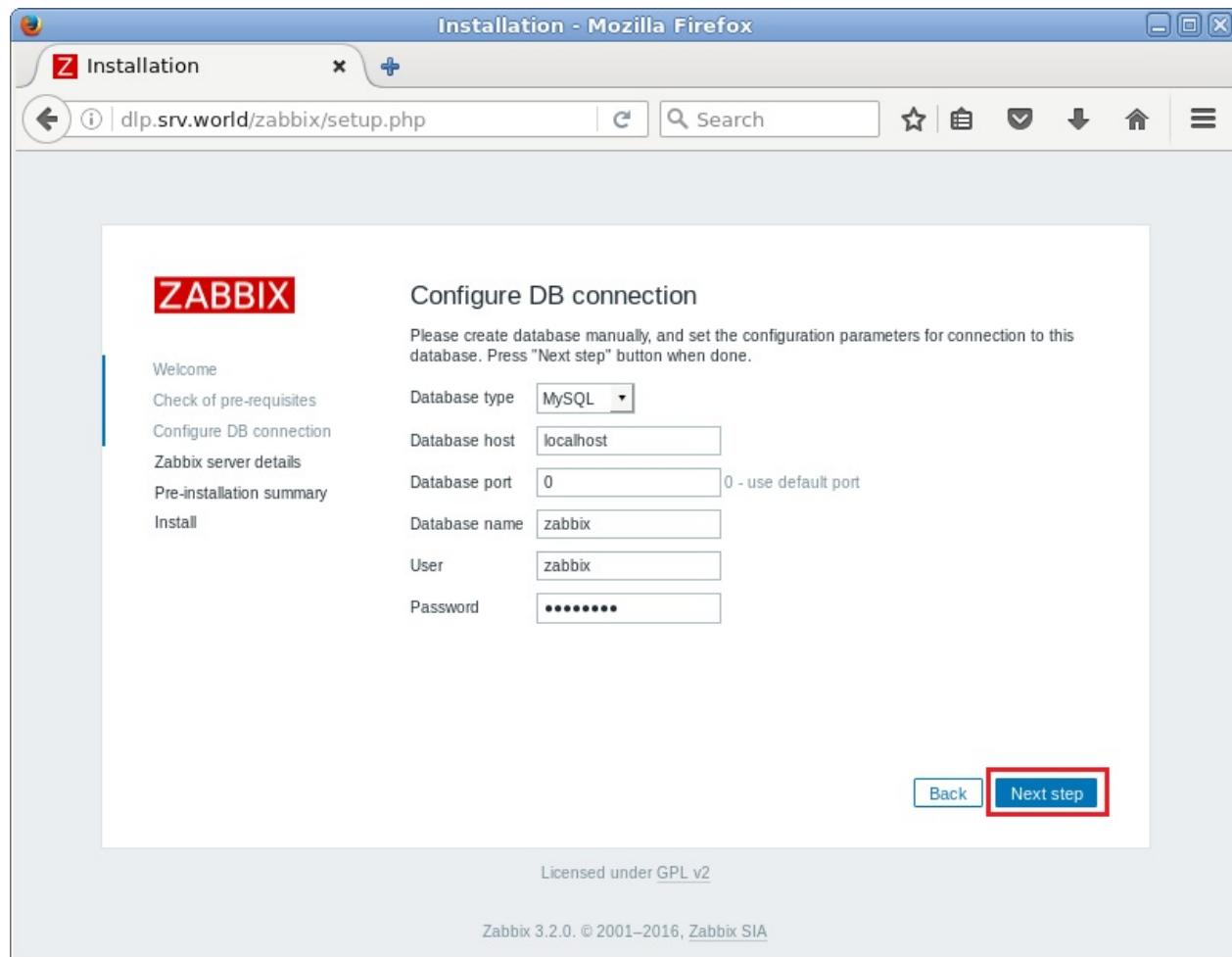


确认所有项目都是“OK”，然后点击“Next step”继续：

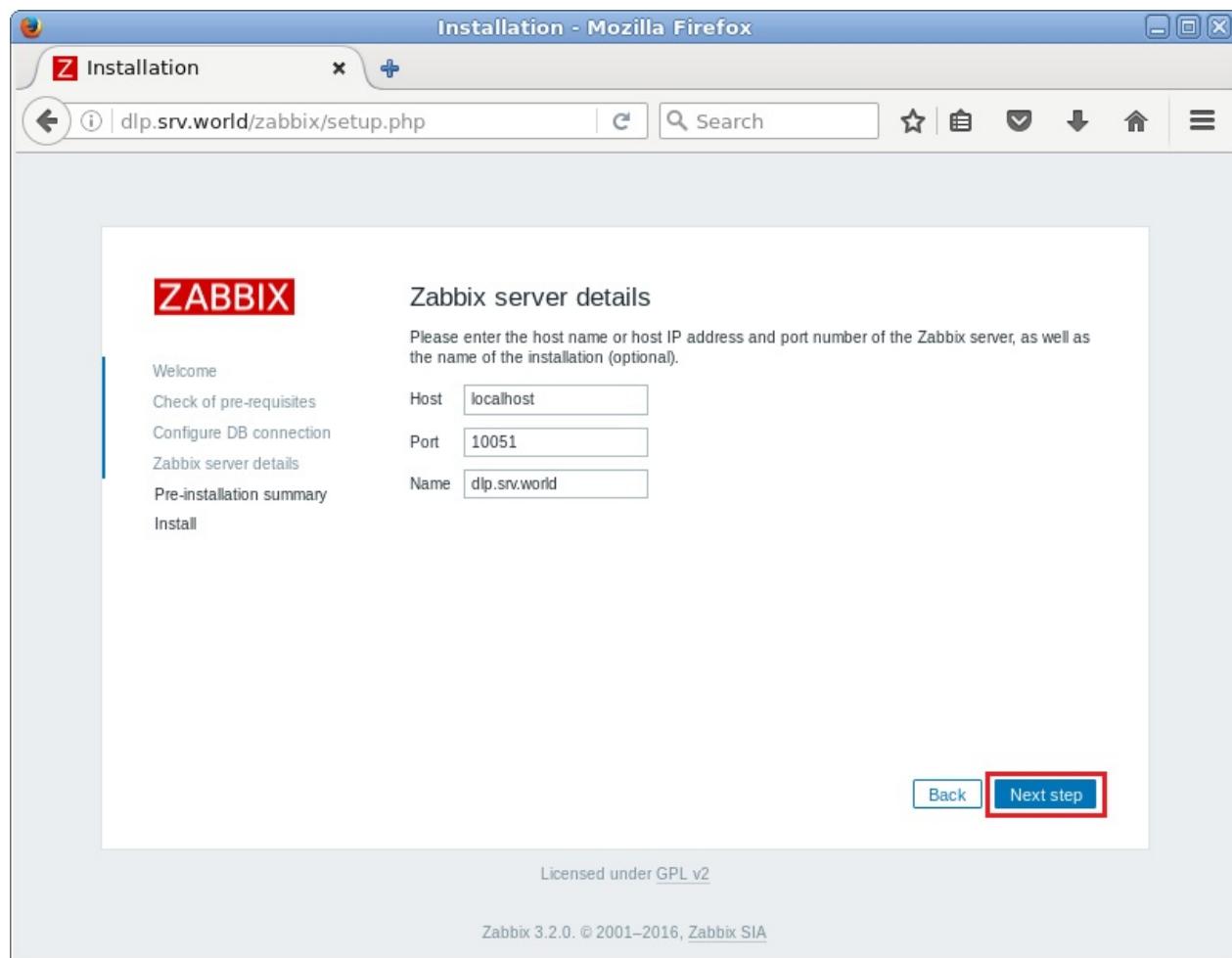
The screenshot shows the Zabbix installation process in Mozilla Firefox. The title bar says 'Installation - Mozilla Firefox'. The address bar shows the URL 'dlp.srv.world/zabbix/setup.php'. The main content area is titled 'Check of pre-requisites'. On the left, there's a sidebar with links: 'Welcome', 'Check of pre-requisites' (which is highlighted in blue), 'Configure DB connection', 'Zabbix server details', 'Pre-installation summary', and 'Install'. The main table lists PHP configuration settings with columns for 'Current value', 'Required', and a status column. Most items are marked as 'OK'. The 'PHP databases support' row shows 'MySQL SQLite3' under 'Current value' and 'OK' under 'Required'. The 'PHP mbstring' row shows 'on' under both 'Current value' and 'Required'. At the bottom right of the table are 'Back' and 'Next step' buttons, with 'Next step' being highlighted with a red box. Below the table, the text 'Licensed under GPL v2' and 'Zabbix 3.2.0. © 2001–2016, Zabbix SIA' are visible.

|                                  | Current value    | Required |
|----------------------------------|------------------|----------|
| PHP version                      | 5.4.16           | 5.4.0 OK |
| PHP option "memory_limit"        | 128M             | 128M OK  |
| PHP option "post_max_size"       | 16M              | 16M OK   |
| PHP option "upload_max_filesize" | 2M               | 2M OK    |
| PHP option "max_execution_time"  | 300              | 300 OK   |
| PHP option "max_input_time"      | 300              | 300 OK   |
| PHP option "date.timezone"       | Asia/Tokyo       | OK       |
| PHP databases support            | MySQL<br>SQLite3 | OK       |
| PHP bcmath                       | on               | OK       |
| PHP mbstring                     | on               | OK       |

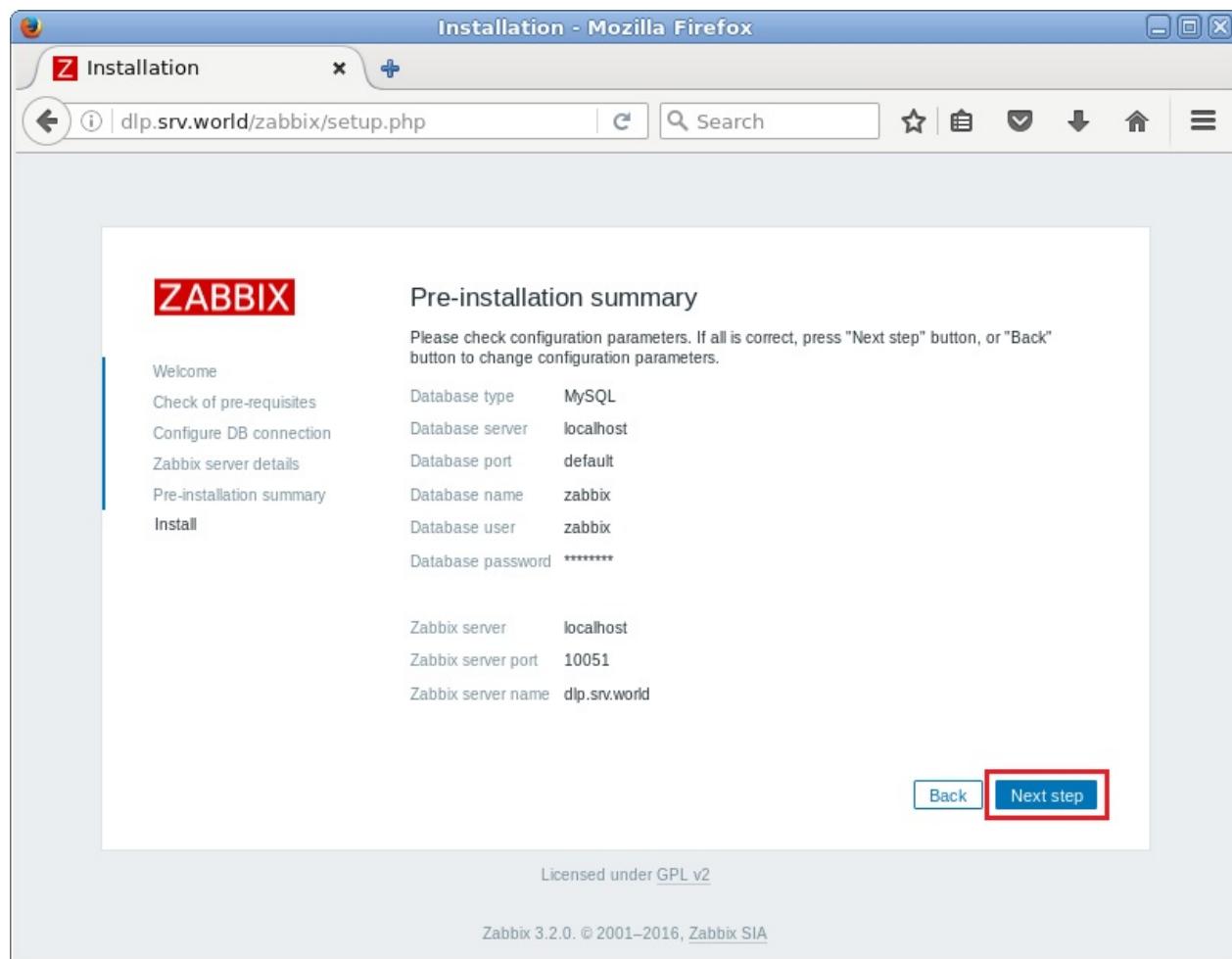
数据库设置部分。将“User”字段默认的“root”更改为“zabbix”，并输入密码：



连接到Zabbix服务器的设置。如果是本地，保持默认。更改“Name”字段为名称：

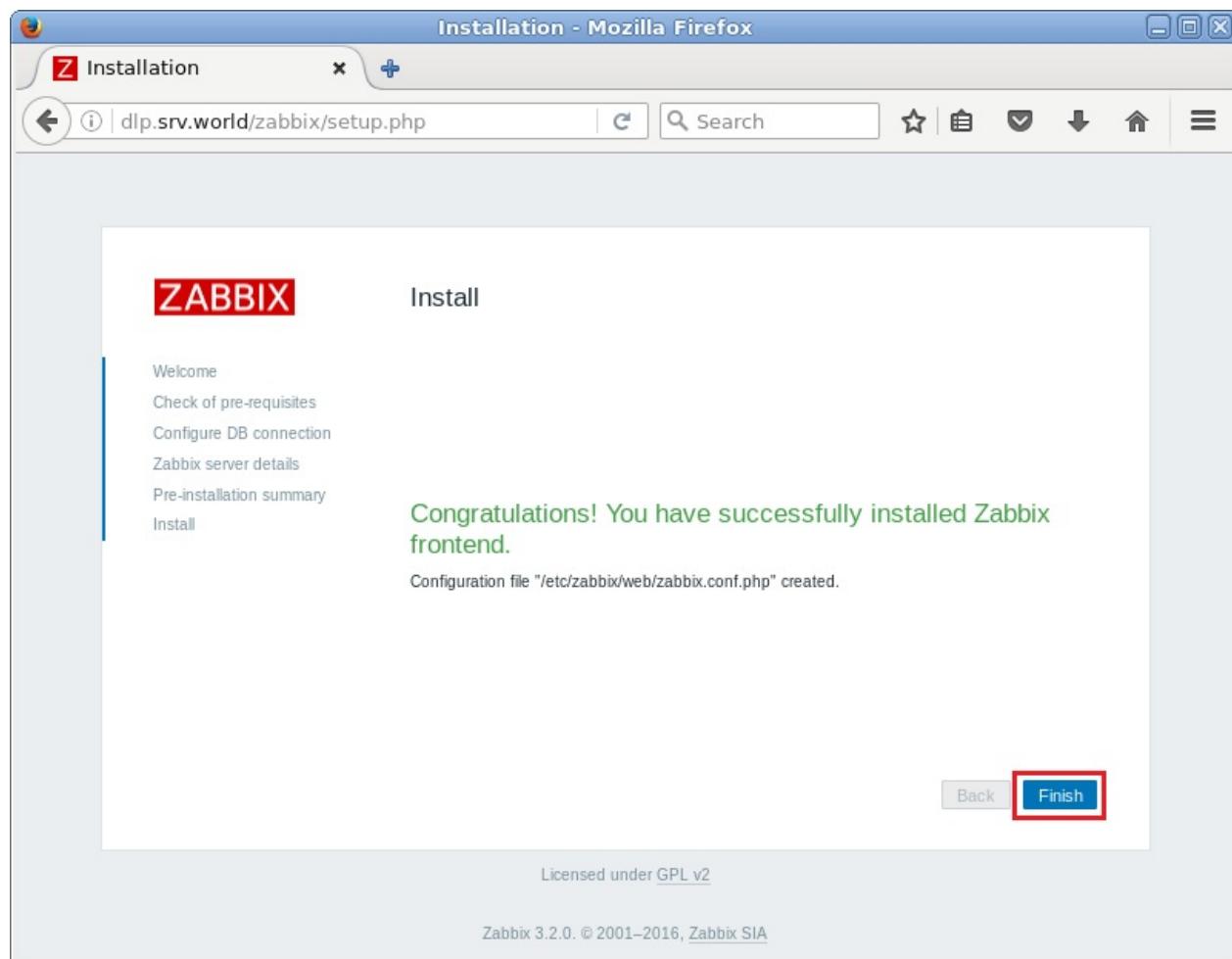


确认前面的设置，如果一切正常，继续下一步：



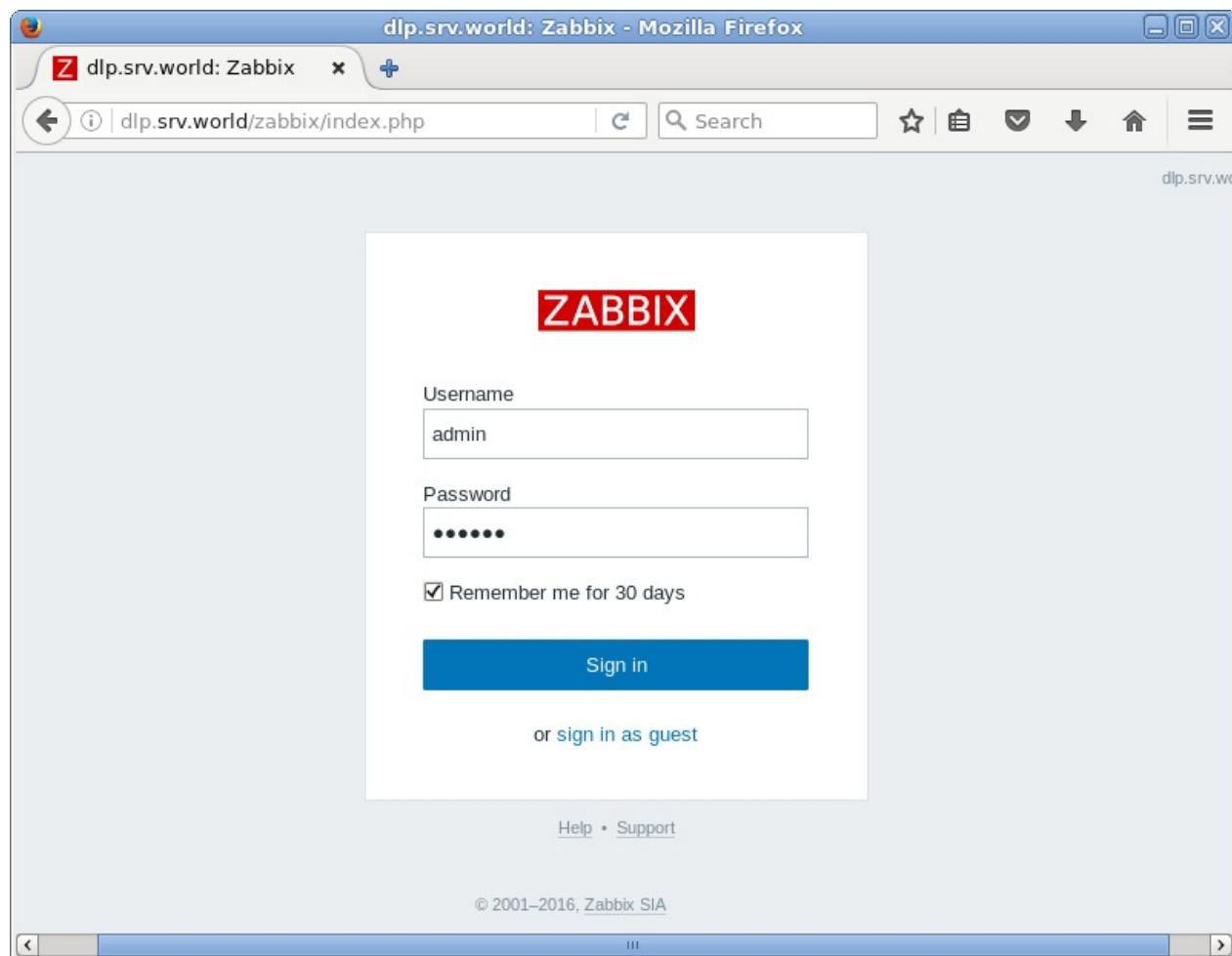
单击“Finish”完成初始设置：

## 12.4. Zabbix



登录页面。可以使用用户名“admin”，初始密码“zabbix”登录：

## 12.4. Zabbix



登录成功。下面是Zabbix管理站点的主页：

## 12.4. Zabbix

The screenshot shows the Zabbix dashboard interface. At the top, there's a header bar with the Zabbix logo and navigation links: Monitoring, Inventory, Reports, Configuration, Administration, and a search bar. Below the header is a secondary navigation bar with links: Dashboard, Problems, Overview, Web, Latest data, Triggers, Graphs, Screens, Maps, Discovery, and IT services. The main content area is titled "Dashboard". It features several sections: "Favourite graphs" (No graphs added), "Favourite screens" (No screens added), "Favourite maps" (No maps added), "Status of Zabbix" (listing parameters like "Zabbix server is running" with value "Yes" and details "localhost:10051"), "System status" (Host group, Disaster, High, Average, Warning, Information, Not classified, showing "No data found." and updated at 11:26:20), "Host status" (Host group, Without problems, With problems, Total, showing "No data found." and updated at 11:26:20), and "Last 20 issues" (Host, Issue, Last change, Age, Info, Ack, Actions, showing "No data found." and updated at 11:26:20). The bottom of the window shows standard browser controls.

首先更改管理员密码。并设置管理员电子邮件地址。

使用管理员帐户“admin”登录Zabbix管理界面，点击右上角人形图标：

## 12.4. Zabbix

The screenshot shows the Zabbix Dashboard interface. On the left, there are three expandable sections: 'graphs' (No graphs added), 'screens' (No screens added), and 'maps' (No maps added). The main content area displays the 'Status of Zabbix' table:

| Parameter                                          | Value | Details         |
|----------------------------------------------------|-------|-----------------|
| Zabbix server is running                           | Yes   | localhost:10051 |
| Number of hosts (enabled/disabled/templates)       | 39    | 0 / 1 / 38      |
| Number of items (enabled/disabled/not supported)   | 0     | 0 / 0 / 0       |
| Number of triggers (enabled/disabled [problem/ok]) | 0     | 0 / 0 [0 / 0]   |
| Number of users (online)                           | 2     | 2               |
| Required server performance, new values per second | 0     |                 |

Below the table, it says "Updated: 11:32:20". At the bottom of the dashboard, there is a navigation bar with categories: Host group, Disaster, High, Average, Warning, Information, and Not classified.

点击“Change Password”：

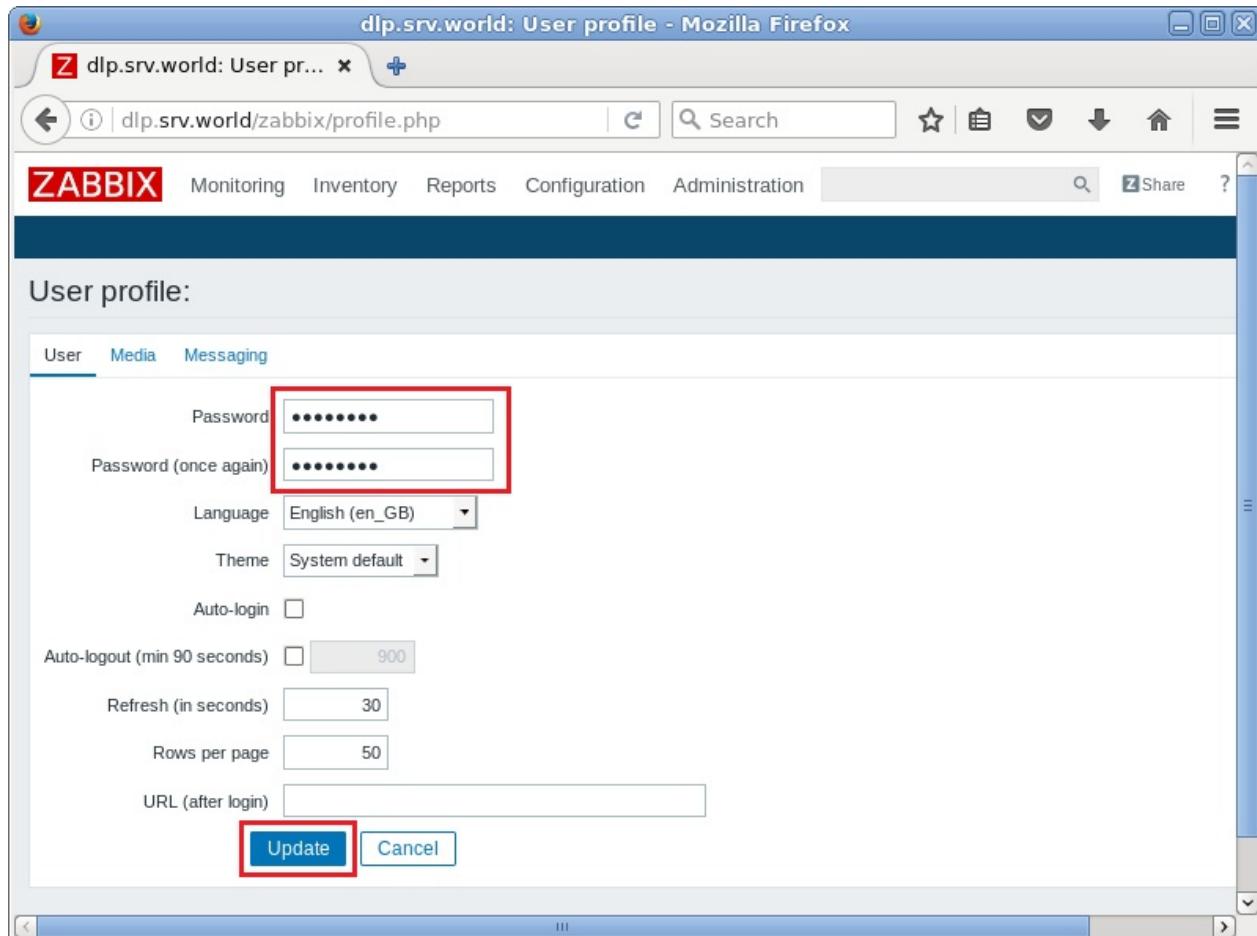
The screenshot shows the 'User profile: Zabbix Administrator' page. At the top, there are tabs: User (which is selected), Media, and Messaging. Below the tabs, there are several configuration fields:

- Password: A text input field with the placeholder "Change password", which is highlighted with a red box.
- Language: A dropdown menu set to "English (en\_GB)".
- Theme: A dropdown menu set to "System default".
- Auto-login: A checkbox that is unchecked.
- Auto-logout (min 90 seconds): A dropdown menu set to "900".
- Refresh (in seconds): A dropdown menu set to "30".
- Rows per page: A dropdown menu set to "50".
- URL (after login): An empty text input field.

At the bottom of the form are two buttons: "Update" and "Cancel".

## 12.4. Zabbix

输入您要更改的密码，然后点击“Update”更改密码。顺便，本例是英语显示，但如果想要更改显示语言，可以在“Language”字段中选择：



再次进入“Profile”页面，移动到“Media”标签，然后点击“Add”：

## 12.4. Zabbix

The screenshot shows the Zabbix User profile interface. At the top, there's a navigation bar with links for Monitoring, Inventory, Reports, Configuration, Administration, and a search bar. Below the navigation is a dark blue header bar with the ZABBIX logo. The main content area is titled "User profile: Zabbix Administrator". Underneath, there are two tabs: "User" and "Media", with "Media" being the active tab and highlighted with a red box. Below the tabs is a table header row with columns: Media, Type, Send to, When active, Use if severity, Status, and Action. A red box highlights the "Add" button under the Type column. At the bottom of the table are "Update" and "Cancel" buttons, with "Update" also highlighted by a red box.

输入管理员电子邮件地址，然后点击“Add”：

The screenshot shows the Zabbix Media configuration dialog box. It has a title bar "Media" and a URL "dlp.srv.world/zabbix/popup\_media.php?dstfrm=userForm". The form contains fields for "Type" (set to "Email"), "Send to" (containing "root@srv.world", which is highlighted with a red box), "When active" (set to "1-7,00:00-24:00"), and a "Use if severity" section with several checkboxes checked: Not classified, Information, Warning, Average, High, and Disaster. There's also an "Enabled" checkbox which is checked. At the bottom are "Add" and "Cancel" buttons, with "Add" highlighted by a red box.

点击“Update”完成：

The screenshot shows the Zabbix User profile configuration page. At the top, there is a navigation bar with links for Monitoring, Inventory, Reports, Configuration, Administration, Share, and Help. Below the navigation bar, the title "User profile:" is displayed. Underneath it, there are three tabs: User, Media, and Messaging. The "Media" tab is selected and highlighted in blue. A table displays a single media configuration entry:

| Media | Type           | Send to         | When active | Use if severity | Status                                      | Action |
|-------|----------------|-----------------|-------------|-----------------|---------------------------------------------|--------|
| Email | root@srv.world | 1-7,00:00-24:00 | NIWAHD      | Enabled         | <a href="#">Edit</a> <a href="#">Remove</a> |        |

Below the table are two buttons: "Update" and "Cancel". The "Update" button is highlighted with a red rectangle. At the bottom of the page, the Zabbix footer is visible, stating "Zabbix 3.2.0. © 2001–2016, Zabbix SIA".

### 12.4.3. 设置监控本机

使用管理员帐户“admin”登录Zabbix管理界面，点击“Configuration”->“Hosts”。安装了Zabbix代理的本地主机显示如下，选中复选框，然后点击“Enable”：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface in Mozilla Firefox. The URL is `dlp.srv.world/zabbix/hosts.php?ddreset=1`. The navigation bar includes links for ZABBIX, Monitoring, Inventory, Reports, Configuration (which is highlighted with a red box), Administration, and others like Share and Help. Below the navigation is a secondary menu with Host groups, Templates, Hosts (also highlighted with a red box), Maintenance, Actions, Event correlation, Discovery, and IT services. The main content area is titled 'Hosts' and features a search bar and filter options for Name, DNS, IP, and Port. A table lists a single host entry: 'Zabbix server' (selected with a checkbox). The host details include Applications (11), Items (64), Triggers (43), Graphs (10), Discovery (2), Web (127.0.0.1:10050), and various status indicators like Template, App, Zabbix, Server, etc. At the bottom of the table are buttons for Enable, Disable, Export, Mass update, and Delete. The 'Enable' button is also highlighted with a red box.

“Status”将转为“enabled”，并开始监控服务器：

## 12.4. Zabbix

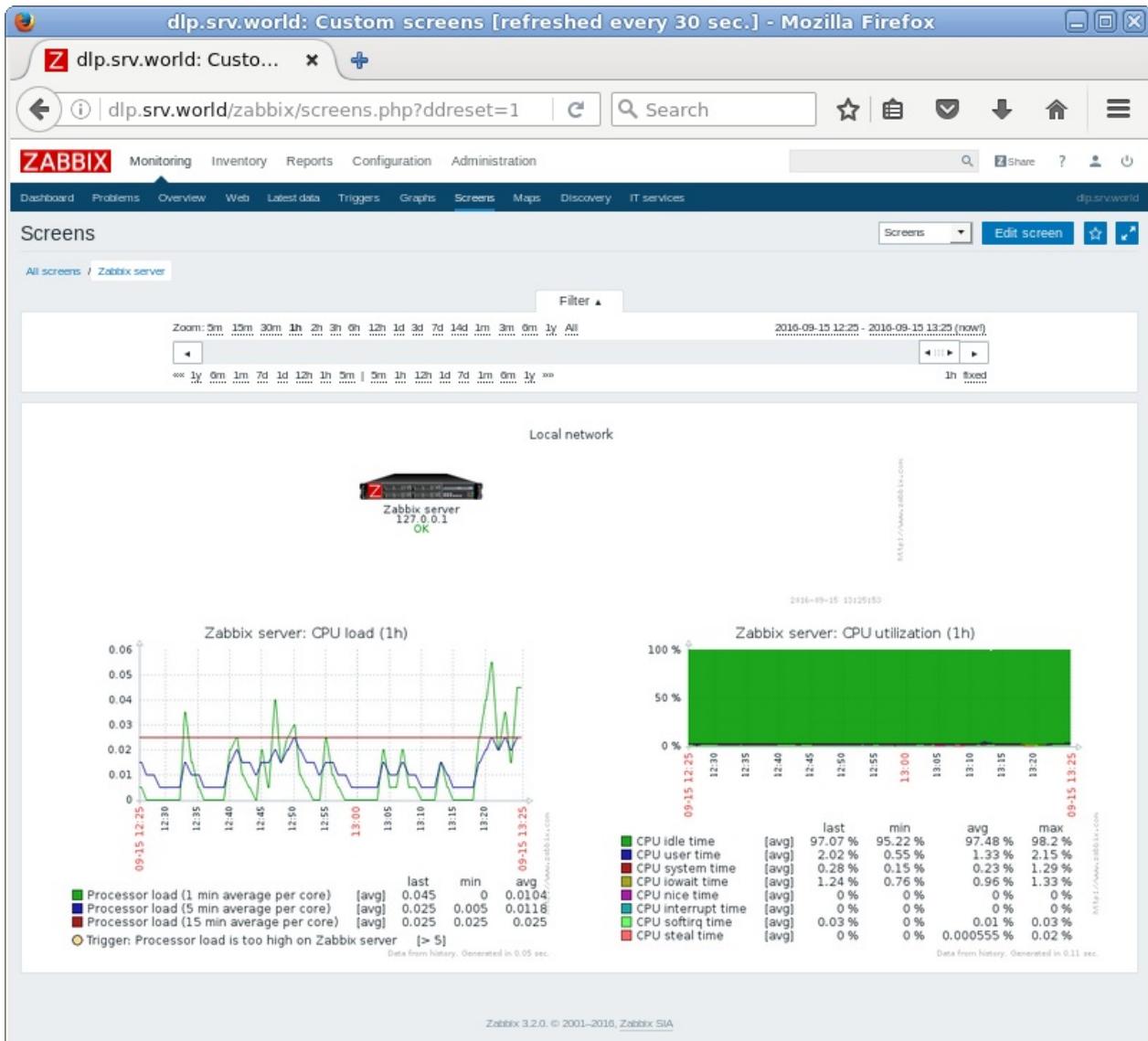
The screenshot shows the Zabbix web interface in Mozilla Firefox. The URL is `dip.srv.world/zabbix/hosts.php`. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, and others. The main menu has tabs for Host groups, Templates, Hosts (which is selected), Maintenance, Actions, Event correlation, Discovery, and IT services. A green header bar indicates 'Host enabled'. Below this, a search bar and a 'Create host' button are visible. A filter section allows searching by Name, DNS, IP, and Port, with 'Apply' and 'Reset' buttons. The main table lists hosts, including 'Zabbix server' which is detailed below:

| Name          | Applications    | Items    | Triggers    | Graphs    | Discovery   | Web                 | Interface                                                                 | Templates | Status                  | Availability | Actions |
|---------------|-----------------|----------|-------------|-----------|-------------|---------------------|---------------------------------------------------------------------------|-----------|-------------------------|--------------|---------|
| Zabbix server | Applications 11 | Items 64 | Triggers 43 | Graphs 10 | Discovery 2 | Web 127.0.0.1:10050 | Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent) | Enabled   | ZBX   SNMP   JMX   IPMI | N            |         |

At the bottom, there are buttons for 0 selected, Enable, Disable, Export, Mass update, and Delete.

几分钟后，收集监控数据如下。以下页面为“Monitoring”->“Screens”：

## 12.4. Zabbix



### 12.4.4. 设置电子邮件通知

设置SMTP服务器以发送邮件通知。

使用管理员帐户“admin”登录Zabbix管理界面，转到“Administration”->“Media Types”，然后点击“Email”：

## 12.4. Zabbix

The screenshot shows the Zabbix Administration - Media types page. The 'Media types' tab is selected. There are three entries listed:

| Name   | Type   | Status  | Used in actions | Details                                                                                     |
|--------|--------|---------|-----------------|---------------------------------------------------------------------------------------------|
| Email  | Email  | Enabled |                 | SMTP server: "mail.company.com", SMTP helo: "company.com", SMTP email: "zabbix@company.com" |
| Jabber | Jabber | Enabled |                 | Jabber identifier: "jabber@company.com"                                                     |
| SMS    | SMS    | Enabled |                 | GSM modem: "/dev/ttyS0"                                                                     |

At the bottom, there are buttons for 'Enable', 'Disable', and 'Delete'. A note at the bottom right says 'Displaying 3 of 3'.

如下设置要使用的SMTP服务器信息，然后点击“Update”按钮：

The screenshot shows the Zabbix Media type configuration dialog for the 'Email' type. The 'Name' field is set to 'Email'. Other fields include:

|                     |                       |
|---------------------|-----------------------|
| Type                | Email                 |
| SMTP server         | localhost             |
| SMTP server port    | 25                    |
| SMTP helo           | localhost             |
| SMTP email          | root@localhost        |
| Connection security | None STARTTLS SSL/TLS |
| Authentication      | None Normal password  |

The 'Enabled' checkbox is checked. At the bottom, the 'Update' button is highlighted with a red box.

## 12.4. Zabbix

在“Details”字段中确认正常更改SMTP服务器：

The screenshot shows the Zabbix Administration interface for managing media types. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, and other management options like General, Proxies, Authentication, User groups, Users, Media types, Scripts, and Queue. A success message 'Media type updated' is displayed above the main table. The 'Media types' table lists three entries: Email (Enabled), Jabber (Enabled), and SMS (Enabled). Each entry provides a detailed description of the configuration (e.g., SMTP server: "localhost", Jabber identifier: "jabber@company.com"). Below the table are buttons for selecting, enabling, disabling, or deleting items.

| Name   | Type   | Status  | Used in actions | Details                                                                        |
|--------|--------|---------|-----------------|--------------------------------------------------------------------------------|
| Email  | Email  | Enabled |                 | SMTP server: "localhost", SMTP helo: "localhost", SMTP email: "root@localhost" |
| Jabber | Jabber | Enabled |                 | Jabber identifier: "jabber@company.com"                                        |
| SMS    | SMS    | Enabled |                 | GSM modem: "/dev/ttyS0"                                                        |

许多项目使用默认模板配置，但默认情况下未设置为发送通知，因此如下设置：

先[设置好管理员邮件地址](#)。

使用管理员帐户“admin”登录Zabbix管理界面，转到“Configuration”->“Actions”，默认情况下，发送通知的默认状态如下所示，点击“Disabled”切换为“Enabled”：

## 12.4. Zabbix

The screenshot shows the Zabbix 'Configuration of actions' page. The 'Actions' tab is selected. A red box highlights the 'Actions' tab in the navigation bar. Another red box highlights the 'Disabled' status of the single listed action.

| Name                                                              | Conditions | Operations                                                       | Status   |
|-------------------------------------------------------------------|------------|------------------------------------------------------------------|----------|
| <input type="checkbox"/> Report problems to Zabbix administrators |            | Send message to user groups: Zabbix administrators via all media | Disabled |

Displaying 1 of 1 found

已启用通知。默认收件人是Zabbix管理员组。

The screenshot shows the Zabbix 'Configuration of actions' page. The 'Actions' tab is selected. A green banner at the top indicates 'Action enabled'. A red box highlights the 'Enabled' status of the single listed action.

| Name                                                              | Conditions | Operations                                                       | Status  |
|-------------------------------------------------------------------|------------|------------------------------------------------------------------|---------|
| <input type="checkbox"/> Report problems to Zabbix administrators |            | Send message to user groups: Zabbix administrators via all media | Enabled |

Displaying 1 of 1 found

如果值超过设置的阈值，则发送通知，如下所示：

Date: Fri, 11 Mar 2016 19:02:46 +0900  
Subject: PROBLEM: Zabbix agent on Zabbix server is unreachable for 5 minutes  
Content-Type: text/plain; charset="UTF-8"  
Status: R

Trigger: Zabbix agent on Zabbix server is unreachable for 5 minutes

Trigger status: PROBLEM

Trigger severity: Average

Trigger URL:

Item values:

1. Agent ping (Zabbix server:agent.ping): Up (1)
2. \*UNKNOWN\* (\*UNKNOWN\*: \*UNKNOWN\*): \*UNKNOWN\*
3. \*UNKNOWN\* (\*UNKNOWN\*: \*UNKNOWN\*): \*UNKNOWN\*

Original event ID: 81

点击Action的名称，可以查看详细信息：

The screenshot shows the Zabbix web interface in a Mozilla Firefox browser. The URL is `dip.srv.world/zabbix/actionconf.php?form=u`. The page title is "dip.srv.world: Configuration of actions - Mozilla Firefox". The navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, Host groups, Templates, Hosts, Maintenance, Actions (which is the active tab), Event correlation, Discovery, and IT services. The main content area is titled "Actions". It displays a form for editing an action named "Report problems to Zabbix administrators". The form fields include "Name" (set to "Report problems to Zabbix administrators"), "Conditions" (empty), "New condition" (set to "Trigger name like %"), "Enabled" (checked), and buttons for "Update", "Clone", "Delete", and "Cancel". At the bottom of the page, a footer states "Zabbix 3.2.0. © 2001–2016, Zabbix SIA".

可以在“Operations”标签上修改通知邮件内容：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface under the 'Actions' configuration. The 'Operations' tab is selected. In the 'Default subject' field, the placeholder '[TRIGGER.STATUS]: {TRIGGER.NAME}' is present. The 'Default message' section contains trigger details: Trigger: {TRIGGER.NAME}, Trigger status: {TRIGGER.STATUS}, Trigger severity: {TRIGGER.SEVERITY}, Trigger URL: {TRIGGER.URL}. Below it, 'Item values:' lists '1. {ITEM.NAME1} ({HOST.NAME1}:'. Under 'Operations', there is a table with one step: Step 1: 'Send message to user groups: Zabbix administrators via all media' (Details: Immediately, Duration: Default). Buttons at the bottom include 'Update', 'Clone', 'Delete', and 'Cancel'. The footer indicates Zabbix 3.2.0, © 2001–2016, Zabbix SIA.

可以在“Recovery Operations”标签上恢复默认邮件内容：

The screenshot shows the Zabbix web interface under the 'Actions' configuration. The 'Recovery operations' tab is selected. The 'Default subject' field contains '[TRIGGER.STATUS]: {TRIGGER.NAME}'. The 'Default message' section is identical to the 'Operations' tab, listing trigger details. Below it, 'Item values:' lists '1. {ITEM.NAME1} ({HOST.NAME1}:'. Under 'Operations', there is a table with one step: Step 1: 'Notify all who received any messages regarding the problem before' (Details: New). Buttons at the bottom include 'Update', 'Clone', 'Delete', and 'Cancel'. The footer indicates Zabbix 3.2.0, © 2001–2016, Zabbix SIA.

## 12.4.5. 添加目标主机

### 12.4.5.1. CentOS7服务器

在要监控的目标服务器上安装Zabbix代理：

```
yum -y install  
http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-3.2-  
1.el7.noarch.rpm
```

```
yum -y install zabbix-agent
```

编辑 `/etc/zabbix/zabbix_agentd.conf` 文件：

```
# 指定Zabbix服务器  
Server=10.0.0.30  
  
# 指定Zabbix服务器  
ServerActive=10.0.0.30  
  
# 更改自己的主机名  
Hostname=node01.srv.world
```

```
systemctl start zabbix-agent  
systemctl enable zabbix-agent
```

firewalld防火墙规则：

```
firewall-cmd --add-port=10050/tcp --permanent  
firewall-cmd --reload
```

使用管理员帐户“admin”登录Zabbix管理界面，转到“Configuration”->“Hosts”，点击“Create Host”：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface in Mozilla Firefox. The URL is `dlp.srv.world/zabbix/hosts.php?ddreset=1`. The top navigation bar includes links for ZABBIX, Monitoring, Inventory, Reports, Configuration (which is highlighted with a red box), Administration, and other tabs like Host groups, Templates, and Maintenance. Below the navigation is a sub-menu with Hosts (highlighted with a red box), Maintenance, Actions, Event correlation, Discovery, and IT services. The main content area is titled 'Hosts' with a 'Create host' button highlighted with a red box. There are search and filter fields for Name, DNS, IP, and Port. Below these are 'Apply' and 'Reset' buttons. A table lists existing hosts, with one entry for 'Zabbix server' shown in detail. The table columns include Name, Applications, Items, Triggers, Graphs, Discovery, Web, Interface, Templates, Status, Availability, and Agent. The 'Zabbix server' row shows 11 applications, 76 items, 47 triggers, 13 graphs, 2 discoveries, and 2 webs. Its IP is 127.0.0.1:10050, it's associated with a Zabbix template, and its status is Enabled. It also has ZBX, SNMP, JMX, and IPMI monitoring enabled. The 'Agent' status is listed as NON. At the bottom of the table are buttons for 0 selected, Enable, Disable, Export, Mass update, and Delete. The footer of the page indicates it's Zabbix 3.2.0, © 2001–2016, Zabbix SIA.

在“Hostname”字段输入主机名，在“Visible name”字段输入任意名称，  
在“Groups”字段选择一个组或新建一个组，在“Agent interfaces”字段输入IP地址和  
DNS名称，其他字段是可选的。全部完成后，转到“Templates”标签：

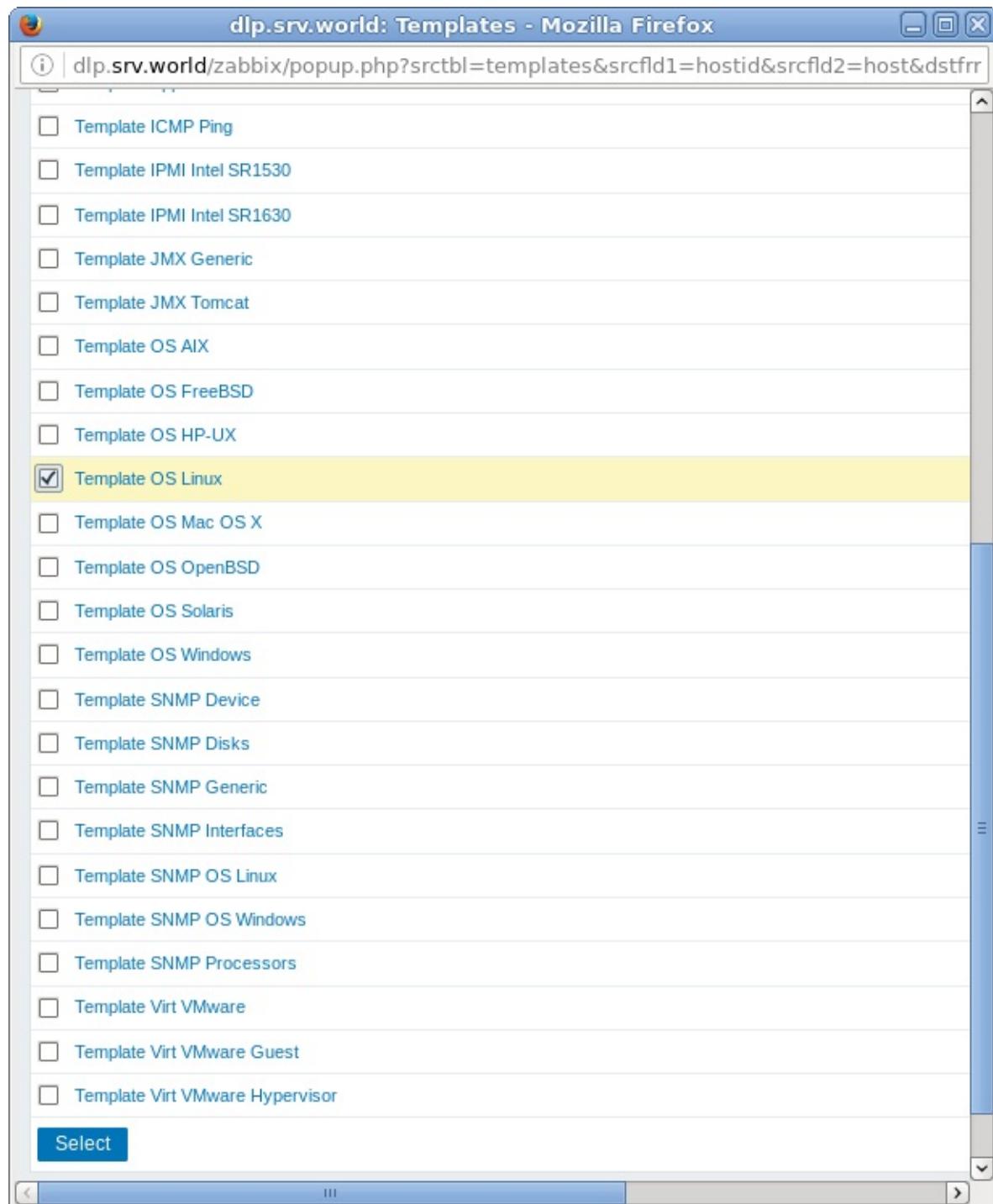
## 12.4. Zabbix

The screenshot shows the Zabbix web interface for configuring hosts. The URL is `dip.srv.world/zabbix/hosts.php?groupid=0&l`. The 'Hosts' tab is selected. The 'Templates' tab is highlighted with a red box. The 'Host name' field contains 'node01.srv.world'. The 'Visible name' field also contains 'node01.srv.world'. In the 'Groups' section, 'Linux servers' is selected in the 'In groups' list. In the 'Other groups' list, 'Discovered hosts', 'Hypervisors', 'Templates', 'Virtual machines', and 'Zabbix servers' are listed. A 'New group' input field is empty. Under 'Agent interfaces', there is one entry: IP address 10.0.0.51, DNS name empty, Connect to IP, Port 10050, and Default selected. An 'Add' button is available. Under 'SNMP interfaces', there is an 'Add' button. At the bottom, there are 'Add' and 'Cancel' buttons.

点击“Select”：

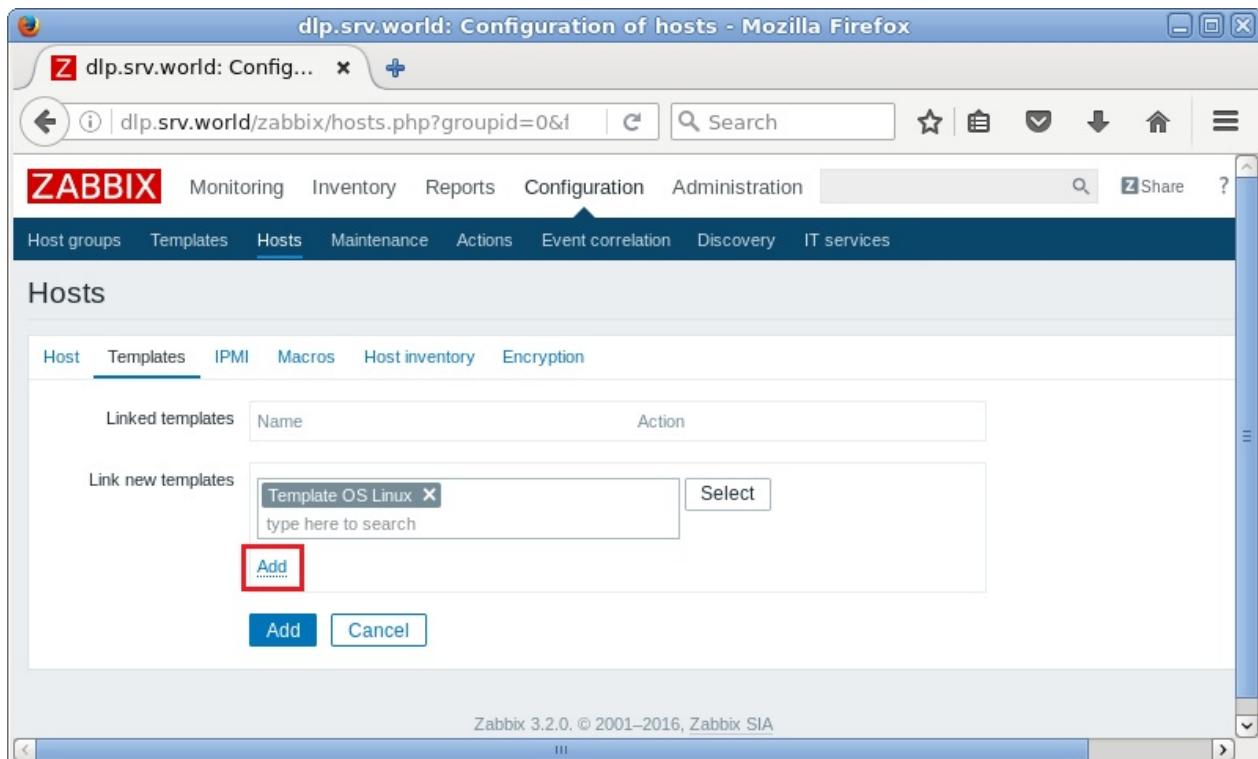
The screenshot shows the Zabbix web interface for linking templates to a host. The URL is `dip.srv.world/zabbix/hosts.php?groupid=0&l`. The 'Hosts' tab is selected. The 'Templates' tab is highlighted with a red box. The 'Linked templates' section shows a table with columns 'Name' and 'Action'. The 'Link new templates' section has a search input 'type here to search' and a 'Select' button highlighted with a red box. Below it are 'Add' and 'Cancel' buttons. At the bottom, it says 'Zabbix 3.2.0. © 2001–2016, Zabbix SIA'.

选择“Template OS Linux”并点击“Select”：



点击“Add”链接：

## 12.4. Zabbix



ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Event correlation Discovery IT services

Hosts

Host Templates IPMI Macros Host inventory Encryption

Linked templates Name Action

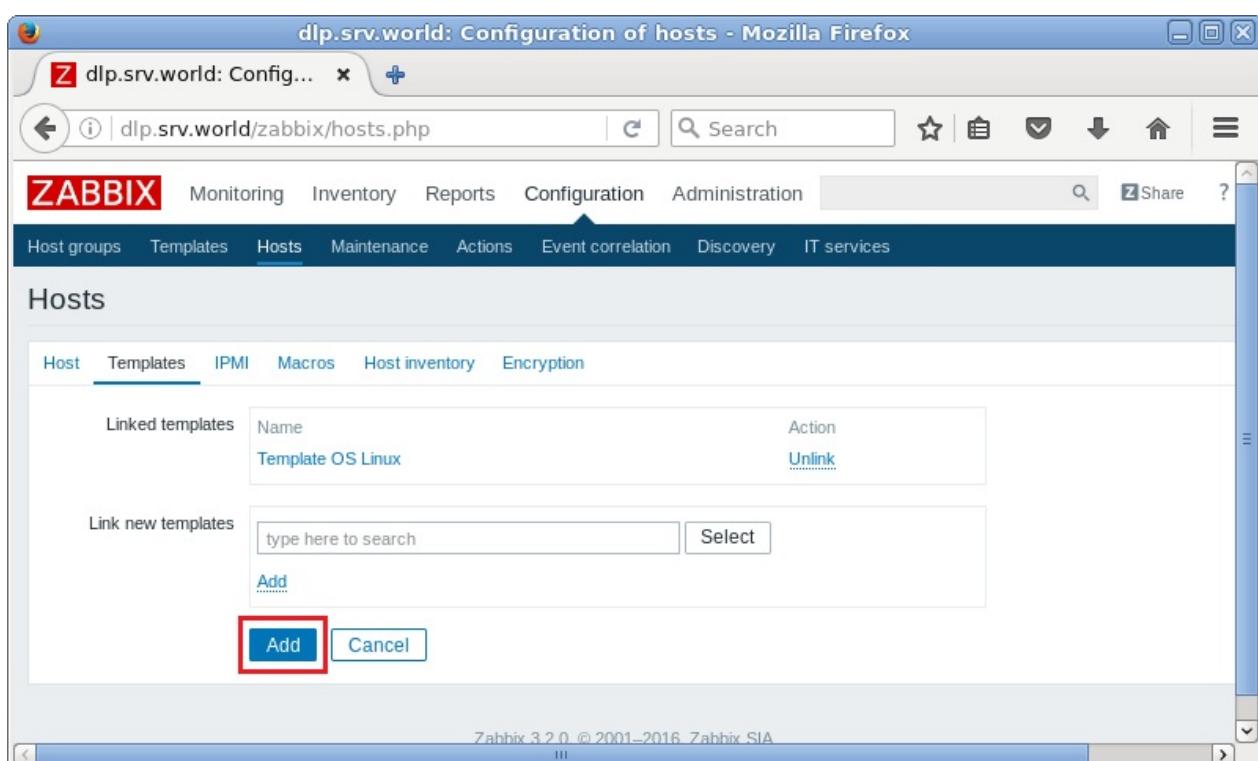
Link new templates Template OS Linux Select

type here to search

Add Cancel

Zabbix 3.2.0. © 2001–2016, Zabbix SIA

确认添加的模板，然后点击“Add”按钮：



ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Event correlation Discovery IT services

Hosts

Host Templates IPMI Macros Host inventory Encryption

Linked templates Name Action

Template OS Linux Unlink

Link new templates type here to search Select

Add Cancel

Zabbix 3.2.0. © 2001–2016, Zabbix SIA

新的监控目标已经添加：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface for managing hosts. The URL is `dip.srv.world/zabbix/hosts.php`. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, and a search bar. Below the navigation is a secondary menu with Host groups, Templates, Hosts (which is selected), Maintenance, Actions, Event correlation, Discovery, and IT services.

The main content area has a green header bar stating "Host added". It displays a table of hosts with columns for Name, Applications, Items, Triggers, Graphs, Discovery, Web, Interface, Templates, Status, and Availability. Two hosts are listed:

| Name             | Applications    | Items    | Triggers    | Graphs    | Discovery   | Web                     | Interface | Templates                                                                                            | Status  | Availability            |
|------------------|-----------------|----------|-------------|-----------|-------------|-------------------------|-----------|------------------------------------------------------------------------------------------------------|---------|-------------------------|
| node01.srv.world | Applications 10 | Items 32 | Triggers 15 | Graphs 5  | Discovery 2 | Web 10.0.0.51:<br>10050 |           | Template<br>OS Linux<br>(Template<br>App<br>Zabbix<br>Agent)                                         | Enabled | ZBX   SNMP   JMX   IPMI |
| Zabbix server    | Applications 11 | Items 76 | Triggers 47 | Graphs 13 | Discovery 2 | Web 127.0.0.1:<br>10050 |           | Template<br>App<br>Zabbix<br>Server,<br>Template<br>OS Linux<br>(Template<br>App<br>Zabbix<br>Agent) | Enabled | ZBX   SNMP   JMX   IPMI |

At the bottom of the table, there are buttons for 0 selected, Enable, Disable, Export, Mass update, and Delete. A status message "Displaying 2 of 2" is shown at the bottom right.

几分钟后，收集监测数据如下：

## 12.4. Zabbix

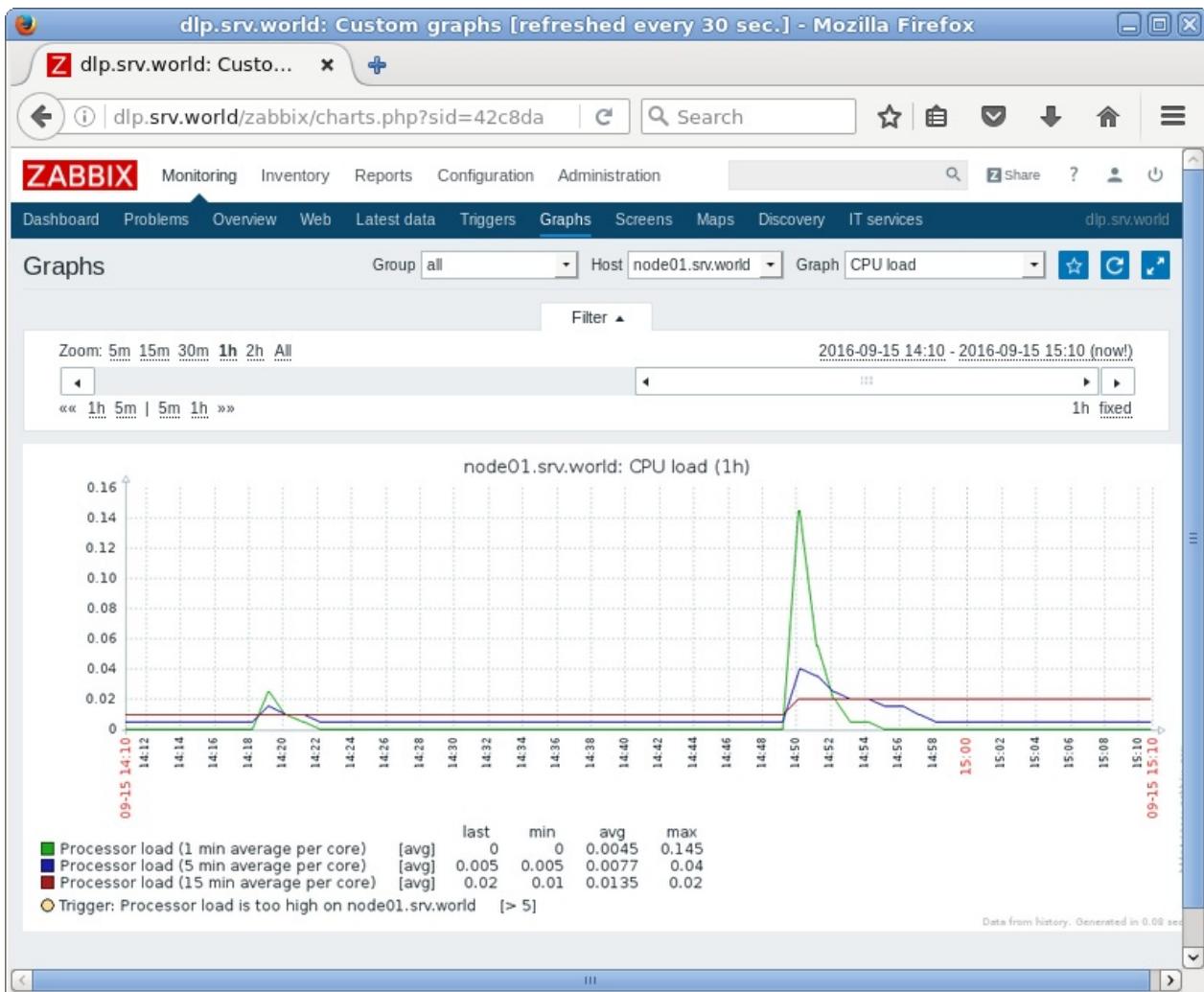
The screenshot shows the Zabbix web interface with the URL `dlp.srv.world/zabbix/latest.php?ddreset=1`. The main title is "Latest data [refreshed every 30 sec.]". The navigation menu includes Monitoring, Inventory, Reports, Configuration, Administration, Dashboard, Problems, Overview, Web, Latest data (selected), Triggers, Graphs, Screens, Maps, Discovery, IT services, and a user icon.

The "Latest data" section has a "Filter" button and search fields for Host groups (Linux servers), Hosts, and Application. It also includes checkboxes for "Show items without data" and "Show details".

The table below lists monitoring items for the host "node01.srv.world".

| Host                                     | Name                                    | Last check         | Last value | Change                | Action                |
|------------------------------------------|-----------------------------------------|--------------------|------------|-----------------------|-----------------------|
| node01.srv.world                         | CPU (13 Items)                          |                    |            |                       |                       |
|                                          | Context switches per second             | 2016-09-15 15:0... | 31 sps     | -2 sps                | <a href="#">Graph</a> |
|                                          | CPU idle time                           | 2016-09-15 15:0... | 99.94 %    | -0.01 %               | <a href="#">Graph</a> |
|                                          | CPU interrupt time                      | 2016-09-15 15:0... | 0 %        |                       | <a href="#">Graph</a> |
|                                          | CPU iowait time                         | 2016-09-15 15:0... | 0 %        |                       | <a href="#">Graph</a> |
|                                          | CPU nice time                           | 2016-09-15 15:0... | 0 %        |                       | <a href="#">Graph</a> |
|                                          | CPU softirq time                        | 2016-09-15 15:0... | 0 %        |                       | <a href="#">Graph</a> |
|                                          | CPU steal time                          | 2016-09-15 15:0... | 0 %        |                       | <a href="#">Graph</a> |
|                                          | CPU system time                         | 2016-09-15 15:0... | 0.04 %     | +0.01 %               | <a href="#">Graph</a> |
|                                          | CPU user time                           | 2016-09-15 15:0... | 0.02 %     |                       | <a href="#">Graph</a> |
|                                          | Interrupts per second                   | 2016-09-15 15:0... | 18 ips     | -2 ips                | <a href="#">Graph</a> |
|                                          | Processor load (1 min average per core) | 2016-09-15 15:0... | 0          |                       | <a href="#">Graph</a> |
|                                          | Processor load (5 min average per core) | 2016-09-15 15:0... | 0.005      |                       | <a href="#">Graph</a> |
| Processor load (15 min average per core) | 2016-09-15 15:0...                      | 0.02               |            | <a href="#">Graph</a> |                       |
| node01.srv.world                         | Filesystems (10 Items)                  |                    |            |                       |                       |
|                                          | Free disk space on /                    | 2016-09-15 15:0... | 25.04 GB   |                       | <a href="#">Graph</a> |
|                                          | Free disk space on / (percentage)       | 2016-09-15 15:0... | 94.66 %    |                       | <a href="#">Graph</a> |
|                                          | Free disk space on /boot                | 2016-09-15 15:0... | 288.77 MB  |                       | <a href="#">Graph</a> |
|                                          | Free disk space on /boot (percentage)   | 2016-09-15 15:0... | 58.14 %    |                       | <a href="#">Graph</a> |

## 12.4. Zabbix



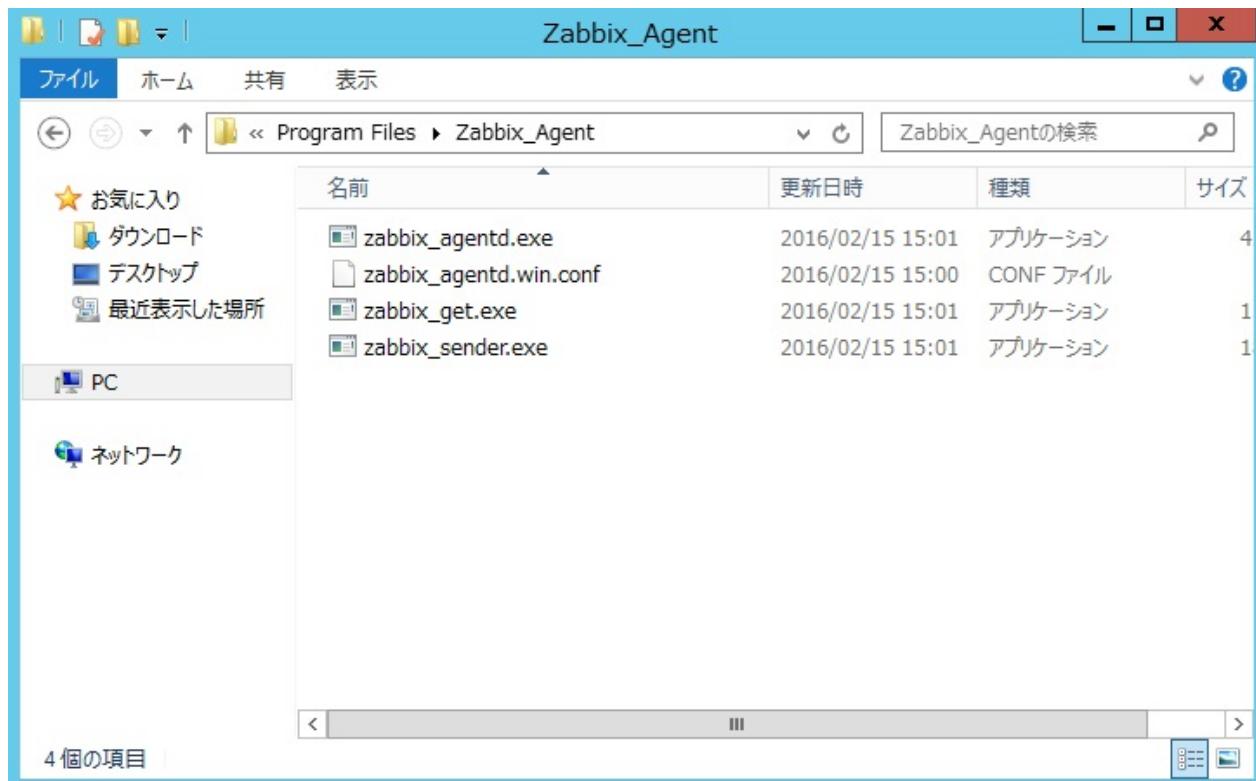
### 12.4.5.2. Windows服务器

以Windows Server 2012 R2为例。

下载Windows目标主机的Zabbix代理（在“Zabbix Sources”部分下载最新版）。

下载后解压文件，将“conf”下的“zabbix\_agentd.win.conf”和“bin”文件夹下的3个exe文件（分32位和64位，按自己实际情况选择）复制到一个文件夹中，如下：

## 12.4. Zabbix



使用文本编辑器打开配置文件 `zabbix_agentd.win.conf` 并更改参数：

```
# 指定日志文件的位置
LogFile=C:\Program Files\Zabbix_Agent\zabbix_agentd.log

# 指定Zabbix服务器
Server=10.0.0.30

# 指定Zabbix服务器
ServerActive=10.0.0.30

# 更改自己的主机名
Hostname=fd3s.srv.world
```

使用管理员权限启动命令提示符，然后输入命令行：

```
cd C:\Program Files\Zabbix_Agent
zabbix_agentd.exe --config "C:\Program
Files\Zabbix_Agent\zabbix_agentd.win.conf" --install
```

## 12.4. Zabbix



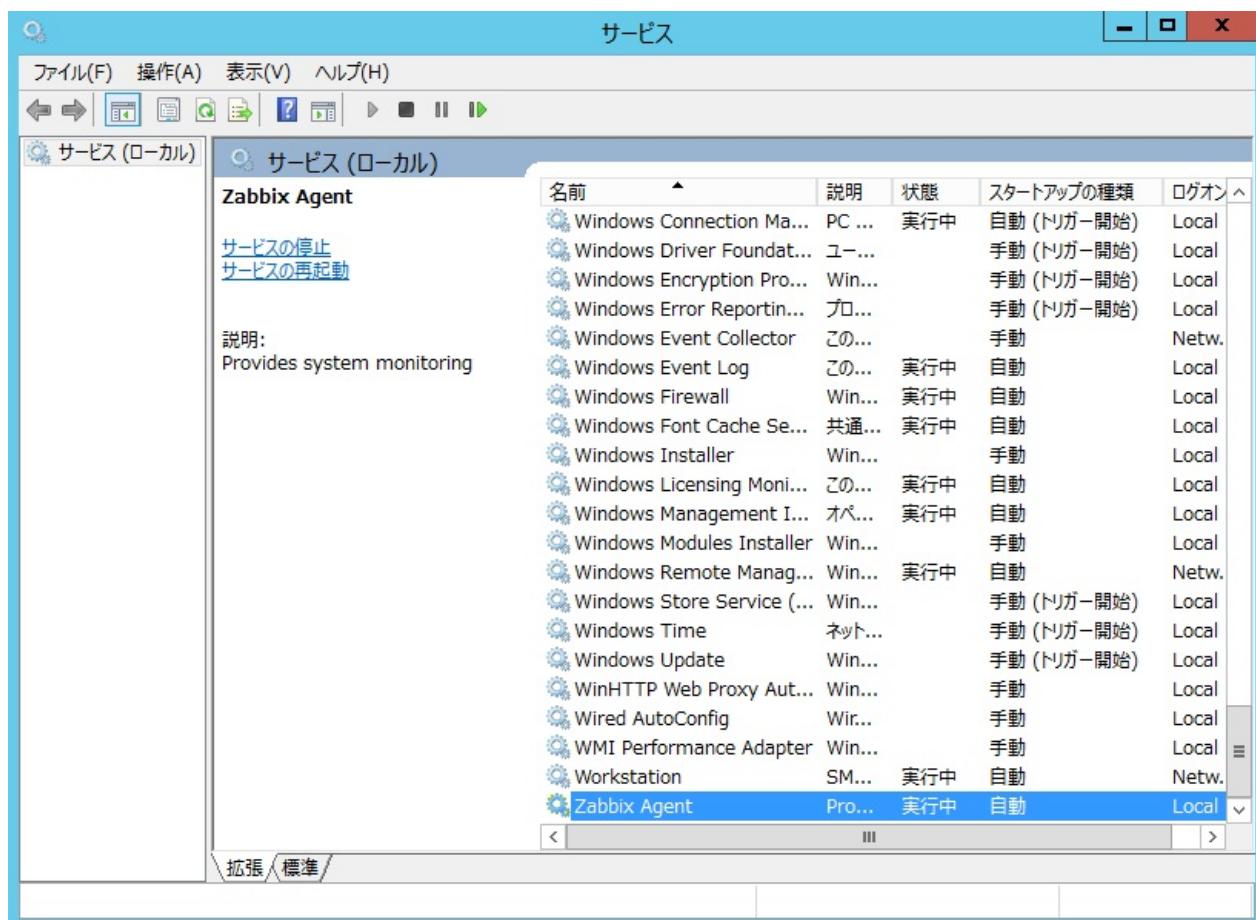
```
管理者: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\$Users\$Administrator>cd C:\Program Files\Zabbix_Agent

C:\$Program Files\Zabbix_Agent>zabbix_agentd.exe --config "C:\Program Files\Zabbix_Agent\zabbix_agentd.win.conf" --install
zabbix_agentd.exe [2716]: service [Zabbix Agent] installed successfully
zabbix_agentd.exe [2716]: event source [Zabbix Agent] installed successfully

C:\$Program Files\Zabbix_Agent>
```

安装后，打开服务管理，然后“Zabbix Agent”添加如下。单击“Start”运行：



使用管理员帐户“admin”登录Zabbix管理界面，转到“Configuration”->“Hosts”，点击“Create Host”：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface for managing hosts. The URL is `d1p.srv.world/zabbix/hosts.php?ddreset=1`. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration (which is highlighted with a red box), Administration, and other system management options. Below the navigation is a secondary menu with Host groups, Templates, Hosts (also highlighted with a red box), Maintenance, Actions, Event correlation, Discovery, and IT services. The main content area is titled 'Hosts' and features a search bar with a 'Group' dropdown set to 'all'. A prominent red box highlights the 'Create host' button. Below the search bar are four input fields: Name, DNS, IP, and Port, each with a corresponding text input box. Underneath these are 'Apply' and 'Reset' buttons. The main table lists two hosts: 'node01.srv.world' and 'Zabbix server'. Each host entry provides a summary of its status (e.g., Applications, Items, Triggers, Graphs) and network information (IP, port). To the right of each host is a detailed view of its configuration, including templates like 'OS Linux (Template)', application types like 'App', and monitoring protocols like 'ZBX', 'SNMP', 'JMX', and 'IPMI'. At the bottom of the table are buttons for 'Enable', 'Disable', 'Export', 'Mass update', and 'Delete'. The status bar at the bottom indicates 'Displaying 2 of 2'.

在“Hostname”字段输入主机名，在“Visible name”字段输入任意名称，  
在“Groups”字段选择一个组或新建一个组，在“Agent interfaces”字段输入IP地址和  
DNS名称，其他字段是可选的。全部完成后，转到“Templates”标签：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface for managing hosts. The URL is `dlp.srv.world/zabbix/hosts.php?groupid=0&l`. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, and a search bar. Below the navigation is a secondary menu with Host groups, Templates, Hosts (which is selected), Maintenance, Actions, Event correlation, Discovery, and IT services. The main content area is titled 'Hosts'. Under the 'Hosts' tab, there are tabs for Host, Templates (which is highlighted with a red box), IPMI, Macros, Host inventory, and Encryption. The 'Host name' field contains 'fd3s.srv.world'. The 'Visible name' field also contains 'fd3s.srv.world'. In the 'Groups' section, there is a list of 'In groups' and 'Other groups'. A 'New group' input field contains 'Windows Servers'. The 'Agent interfaces' section shows an IP address of '10.0.0.100' and a port of '10050'. There is a 'Select' button next to the port field. At the bottom of the interface are 'Add' and 'Remove' buttons.

点击“Select”：

This screenshot shows the same Zabbix interface as the previous one, but with a focus on the 'Templates' tab. The 'Templates' tab is selected and highlighted with a red box. Below it, there is a 'Linked templates' section with a 'Name' input field and an 'Action' column. Underneath is a 'Link new templates' section with a search input field containing 'type here to search' and a 'Select' button highlighted with a red box. At the bottom of this section are 'Add' and 'Cancel' buttons. The footer of the page displays the text 'Zabbix 3.2.0. © 2001–2016, Zabbix SIA'.

选择“Template OS Windows”并点击“Select”：

## 12.4. Zabbix

The screenshot shows a Mozilla Firefox browser window with the title "dlp.srv.world: Templates - Mozilla Firefox". The address bar contains the URL "dlp.srv.world/zabbix/popup.php?srctbl=templates&srcfld1=hostid&srcfld2=host&ds". The main content area displays a list of Zabbix templates, each preceded by a checkbox. The "Template OS Windows" checkbox is checked and highlighted with a yellow background. Other templates listed include: Template IPMI Intel SR1530, Template IPMI Intel SR1630, Template JMX Generic, Template JMX Tomcat, Template OS AIX, Template OS FreeBSD, Template OS HP-UX, Template OS Linux, Template OS Mac OS X, Template OS OpenBSD, Template OS Solaris, and several SNMP and Virt-related templates. At the bottom left is a blue "Select" button, and at the bottom right are navigation arrows.

- Template IPMI Intel SR1530
- Template IPMI Intel SR1630
- Template JMX Generic
- Template JMX Tomcat
- Template OS AIX
- Template OS FreeBSD
- Template OS HP-UX
- Template OS Linux
- Template OS Mac OS X
- Template OS OpenBSD
- Template OS Solaris
- Template OS Windows
- Template SNMP Device
- Template SNMP Disks
- Template SNMP Generic
- Template SNMP Interfaces
- Template SNMP OS Linux
- Template SNMP OS Windows
- Template SNMP Processors
- Template Virt VMware
- Template Virt VMware Guest
- Template Virt VMware Hypervisor

**Select**

点击“Add”链接：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface with the URL `dlp.srv.world/zabbix/hosts.php?groupid=0&l`. The main navigation bar includes Monitoring, Inventory, Reports, Configuration, Administration, Host groups, Templates, Hosts (which is selected), Maintenance, Actions, Event correlation, Discovery, and IT services. Below this, the 'Hosts' section is displayed with tabs for Host, Templates, IPMI, Macros, Host inventory, and Encryption. Under the 'Templates' tab, there's a 'Linked templates' section showing 'Template OS Windows' with an 'Action' column containing a link labeled 'Unlink'. Below this is a 'Link new templates' section with a search input field containing 'Template OS Windows' and a 'Select' button. At the bottom are 'Add' and 'Cancel' buttons. The status bar at the bottom of the browser window indicates 'Zabbix 3.2.0. © 2001–2016, Zabbix SIA'.

确认添加的模板，然后点击“Add”按钮：

This screenshot is similar to the previous one but shows the result of the addition. The 'Template OS Windows' entry now has an 'Action' column with a link labeled 'Unlink'. The 'Add' button at the bottom of the 'Link new templates' section is highlighted with a red box, indicating it was just clicked.

新的监控目标已经添加：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface in Mozilla Firefox. The URL is `dip.srv.world/zabbix/hosts.php`. The main navigation menu includes Monitoring, Inventory, Reports, Configuration, Administration, Host groups, Templates, Hosts (selected), Maintenance, Actions, Event correlation, Discovery, and IT services. The sub-navigation for Hosts includes Details, Host added, and a search bar. Below this is a search/filter section with fields for Name, DNS, IP, and Port, and buttons for Apply and Reset. The main table lists three hosts: `fd3s.srv.world`, `node01.srv.world`, and `Zabbix server`. Each host entry provides a summary of its status (e.g., Applications, Items, Triggers, Graphs, Discovery, Web, Interface, Templates, Status, Availability) and a detailed breakdown of its components (e.g., OS, App, Zabbix Agent). At the bottom of the table are buttons for 0 selected, Enable, Disable, Export, Mass update, and Delete.

| Name                          | Applications    | Items    | Triggers    | Graphs    | Discovery   | Web                      | Interface | Templates                                                                  | Status  | Availability              |
|-------------------------------|-----------------|----------|-------------|-----------|-------------|--------------------------|-----------|----------------------------------------------------------------------------|---------|---------------------------|
| <code>fd3s.srv.world</code>   | Applications 10 | Items 18 | Triggers 9  | Graphs 2  | Discovery 3 | Web 10.0.0.100:<br>10050 |           | Template<br>OS<br>Windows<br>(Template)                                    | Enabled | [ZBX] [SNMP] [JMX] [IPMI] |
| <code>node01.srv.world</code> | Applications 10 | Items 44 | Triggers 19 | Graphs 8  | Discovery 2 | Web 10.0.0.51:<br>10050  |           | Template<br>OS Linux<br>(Template)                                         | Enabled | [ZBX] [SNMP] [JMX] [IPMI] |
| <code>Zabbix server</code>    | Applications 11 | Items 76 | Triggers 47 | Graphs 13 | Discovery 2 | Web 127.0.0.1:<br>10050  |           | Template<br>App<br>Zabbix<br>Server,<br>Template<br>OS Linux<br>(Template) | Enabled | [ZBX] [SNMP] [JMX] [IPMI] |

几分钟后，收集监测数据如下：

## 12.4. Zabbix

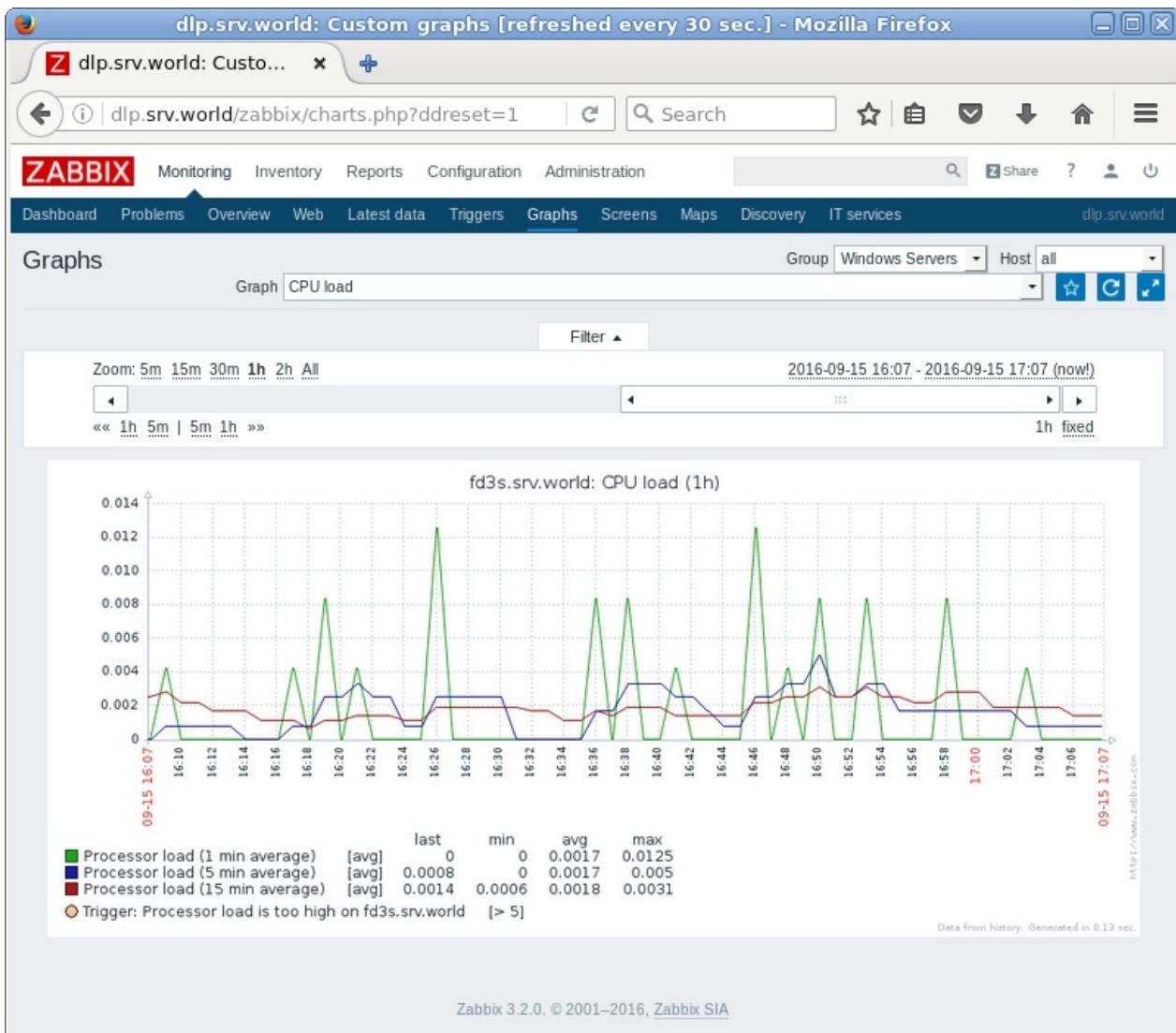
The screenshot shows the Zabbix web interface for monitoring latest data. The URL is `dlp.srv.world/zabbix/latest.php?ddreset=1`. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, and a user menu. Below the navigation is a secondary menu with links for Dashboard, Problems, Overview, Web, Latest data (which is selected), Triggers, Graphs, Screens, Maps, Discovery, IT services, and a dropdown for dlp.srv.world.

The main content area is titled "Latest data". It features a search bar and filter options for Host groups (selected: Windows Servers), Hosts, and Applications. There are checkboxes for "Show items without data" and "Show details". Below these are "Apply" and "Reset" buttons.

The data table lists monitoring items grouped by host and host group. The columns are Host, Name, Last check, Last value, and Change. Each item has a checkbox, a link to its details, and a "Graph" link. The table includes the following data:

| Host           | Name                               | Last check         | Last value         | Change                  |                       |
|----------------|------------------------------------|--------------------|--------------------|-------------------------|-----------------------|
| fd3s.srv.world | <b>CPU (3 Items)</b>               |                    |                    |                         |                       |
| fd3s.srv.world | Processor load (1 min average)     | 2016-09-15 15:2... | 0                  | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | Processor load (5 min average)     | 2016-09-15 15:2... | 0.0008             | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | Processor load (15 min average)    | 2016-09-15 15:2... | 0.0003             | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | <b>Filesystems (8 Items)</b>       |                    |                    |                         |                       |
| fd3s.srv.world | Average disk read queue length     | 2016-09-15 15:2... | 0                  | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | Average disk write queue length    | 2016-09-15 15:2... | 0                  | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | File read bytes per second         | 2016-09-15 15:2... | 0 Bps              | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | File write bytes per second        | 2016-09-15 15:2... | 0 Bps              | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | Free disk space on C:              | 2016-09-15 15:2... | 90.58 GB           | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | Free disk space on C: (percentage) | 2016-09-15 15:2... | 90.89 %            | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | Total disk space on C:             | 2016-09-15 15:0... | 99.66 GB           | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | Used disk space on C:              | 2016-09-15 15:2... | 9.07 GB            | <a href="#">Graph</a>   |                       |
| fd3s.srv.world | <b>General (2 Items)</b>           |                    |                    |                         |                       |
| fd3s.srv.world | System information                 | 2016-09-15 15:0... | Windows FD3S 6.... | <a href="#">History</a> |                       |
| fd3s.srv.world | System uptime                      | 2016-09-15 15:2... | 00:31:29           | +00:00:59               | <a href="#">Graph</a> |
| fd3s.srv.world | <b>Memory (4 Items)</b>            |                    |                    |                         |                       |

## 12.4. Zabbix



### 12.4.6. 添加目标项目

默认情况下，为众所周知的服务提供模板，因此可以轻松地监控它们。

例如，添加HTTP服务为监控目标项目。

使用管理员帐户“admin”登录Zabbix管理界面，转到“Configuration”->“Hosts”，点击需要添加项目的主机名：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface in Mozilla Firefox. The URL is `dlp.srv.world/zabbix/hosts.php`. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration (which is highlighted with a red box), Administration, and other tabs like Host groups, Templates, and Hosts. Below the navigation is a search bar and a toolbar with icons for Share and Help.

The main content area is titled "Hosts" and displays a list of hosts. A green banner at the top says "Host added". There are filters for Name, DNS, IP, and Port, along with "Apply" and "Reset" buttons. The table lists three hosts:

| Name             | Applications    | Items    | Triggers    | Graphs    | Discovery   | Web                      | Interface | Templates                                                                  | Status  | Availability            |
|------------------|-----------------|----------|-------------|-----------|-------------|--------------------------|-----------|----------------------------------------------------------------------------|---------|-------------------------|
| fd3s.srv.world   | Applications 10 | Items 18 | Triggers 9  | Graphs 2  | Discovery 3 | Web 10.0.0.100:<br>10050 |           | Template<br>OS<br>Windows<br>(Template)                                    | Enabled | ZBX   SNMP   JMX   IPMI |
| node01.srv.world | Applications 10 | Items 44 | Triggers 19 | Graphs 8  | Discovery 2 | Web 10.0.0.51:<br>10050  |           | Template<br>OS Linux<br>(Template)                                         | Enabled | ZBX   SNMP   JMX   IPMI |
| Zabbix server    | Applications 11 | Items 76 | Triggers 47 | Graphs 13 | Discovery 2 | Web 127.0.0.1:<br>10050  |           | Template<br>App<br>Zabbix<br>Server,<br>Template<br>OS Linux<br>(Template) | Enabled | ZBX   SNMP   JMX   IPMI |

At the bottom, there are buttons for 0 selected, Enable, Disable, Export, Mass update, and Delete. A status bar at the bottom right says "Displaying 1-3 of 3".

转到“Templates”并点击“Select”：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface at [dlp.srv.world/zabbix/hosts.php?form=update](http://dlp.srv.world/zabbix/hosts.php?form=update). The navigation bar includes Monitoring, Inventory, Reports, Configuration, Administration, Host groups, Templates, Hosts (selected), Maintenance, Actions, Event correlation, Discovery, IT services, and dlp.srv.world. The main content area is titled 'Hosts' and shows 'All hosts / Zabbix server' with various metrics: Enabled (ZBX), SNMP, JMX, IPMI, Applications 11, Items 76, Triggers 47, Graphs 13, Discovery rules 2, and Web scenarios. Below this, there are tabs for Host, Templates (highlighted with a red box), IPMI, Macros, Host inventory, and Encryption. The 'Templates' section contains a table of linked templates:

| Name                       | Action                                                  |
|----------------------------|---------------------------------------------------------|
| Template App Zabbix Server | <a href="#">Unlink</a> <a href="#">Unlink and clear</a> |
| Template OS Linux          | <a href="#">Unlink</a> <a href="#">Unlink and clear</a> |

Below the table is a search input field 'type here to search' and a 'Select' button (also highlighted with a red box). At the bottom are buttons for Update, Clone, Full clone, Delete, and Cancel.

Zabbix 3.2.0. © 2001–2016, [Zabbix SIA](#)

选择“Template App HTTP Service”：

## 12.4. Zabbix

The screenshot shows a Mozilla Firefox browser window with the title "dlp.srv.world: Templates - Mozilla Firefox". The address bar contains the URL "dlp.srv.world/zabbix/popup.php?srctbl=templates&srcfld1=hostid&srcfld2=host&d". The main content area is titled "Templates" and lists various Zabbix template options. The "Template App HTTP Service" and "Template App Zabbix Server" checkboxes are checked, while all other options are unchecked. The "Template App Zabbix Server" option is highlighted with a yellow background.

| Template Option             | Status    |
|-----------------------------|-----------|
| Name                        | unchecked |
| Template App FTP Service    | unchecked |
| Template App HTTP Service   | checked   |
| Template App HTTPS Service  | unchecked |
| Template App IMAP Service   | unchecked |
| Template App LDAP Service   | unchecked |
| Template App MySQL          | unchecked |
| Template App NNTP Service   | unchecked |
| Template App NTP Service    | unchecked |
| Template App POP Service    | unchecked |
| Template App SMTP Service   | unchecked |
| Template App SSH Service    | unchecked |
| Template App Telnet Service | unchecked |
| Template App Zabbix Agent   | unchecked |
| Template App Zabbix Proxy   | unchecked |
| Template App Zabbix Server  | checked   |
| Template ICMP Ping          | unchecked |
| Template IPMI Intel SR1530  | unchecked |
| Template IPMI Intel SR1630  | unchecked |
| Template JMX Generic        | unchecked |
| Template JMX Tomcat         | unchecked |
| Template OS AIX             | unchecked |
| Template OS FreeBSD         | unchecked |

点击“Add”链接：

## 12.4. Zabbix

The screenshot shows the Zabbix 3.2.0 interface for managing hosts. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, and a search bar. Below the navigation is a secondary menu with Host groups, Templates, Hosts (which is selected), Maintenance, Actions, Event correlation, Discovery, and IT services. The main content area is titled 'Hosts' and shows a list of hosts under 'All hosts / Zabbix server'. A sub-menu for 'Enabled' hosts is open, showing metrics like Applications (11), Items (76), Triggers (47), Graphs (13), Discovery rules (2), and Web scenarios. The 'Hosts' tab is active, and the 'Templates' sub-tab is selected. A table lists 'Linked templates' with columns for Name and Action (Unlink, Unlink and clear). Below this is a 'Link new templates' section with a search input field containing 'Template App HTTP Service' and a 'Select' button. A red box highlights the 'Add' button. At the bottom are buttons for Update, Clone, Full clone, Delete, and Cancel.

点击“Update”按钮：

This screenshot is identical to the previous one, showing the Zabbix 3.2.0 hosts configuration page. The 'Update' button at the bottom of the 'Link new templates' section is now highlighted with a red box, indicating the next step in the process.

添加了HTTP服务的模板：

## 12.4. Zabbix

The screenshot shows the Zabbix web interface for managing hosts. The URL is `dip.srv.world/zabbix/hosts.php`. The top navigation bar includes links for Monitoring, Inventory, Reports, Configuration, Administration, and a search bar. Below the navigation is a secondary menu with Host groups, Templates, Hosts (which is selected), Maintenance, Actions, Event correlation, Discovery, and IT services. The main content area is titled "Hosts" and displays a list of three hosts: "fd3s.srv.world", "node01.srv.world", and "Zabbix server". Each host entry provides a summary of its status (e.g., Applications, Items, Triggers, Graphs, Discovery, Web) and its configuration details, including IP (e.g., 10.0.0.100:10050, 10.0.0.51:10050, 127.0.0.1:10050), Status (Enabled), and available monitoring protocols (ZBX, SNMP, JMX, IPMI). A message at the top right says "Host updated". A "Details" button is visible above the host list.

HTTP服务模板有一个项目，检查状态是否活动：

## 12.4. Zabbix

| Item                             | Last Value          | Time                   | Graph      |       |
|----------------------------------|---------------------|------------------------|------------|-------|
| Host boot time                   | 2016-09-15 17:08... | 2016-09-15 10:16...    | Graph      |       |
| Host local time                  | 2016-09-15 17:10... | 2016-09-15 17:10...    | +00:01:00  | Graph |
| Host name                        | 2016-09-15 16:28... | dip.srv.world          | History    |       |
| System information               | 2016-09-15 16:28... | Linux dip.srv.world... | History    |       |
| System uptime                    | 2016-09-15 17:08... | 06:52:14               | +00:09:59  | Graph |
| HTTP service (1 Item)            |                     |                        |            |       |
| HTTP service is running          | Up (1)              | 2016-09-15 17:09...    | Graph      |       |
| Memory (5 Items)                 |                     |                        |            |       |
| Available memory                 | 3.27 GB             | 2016-09-15 17:10...    | +1.78 MB   | Graph |
| Free swap space                  | 3 GB                | 2016-09-15 17:10...    | Graph      |       |
| Free swap space in %             | 100 %               | 2016-09-15 17:10...    | Graph      |       |
| Total memory                     | 3.86 GB             | 2016-09-15 16:28...    | Graph      |       |
| Total swap space                 | 3 GB                | 2016-09-15 16:28...    | Graph      |       |
| Network interfaces (2 Items)     |                     |                        |            |       |
| Incoming network traffic on eth0 | 8.82 Kbps           | 2016-09-15 17:09...    | +1.95 Kbps | Graph |

默认情况下设置触发器，因此如果发生故障，则发送通知如下：

Date: Tue, 15 Mar 2016 20:17:14 +0900  
Subject: PROBLEM: HTTP service is down on Zabbix server  
Content-Type: text/plain; charset="UTF-8"  
Status: R0

Trigger: HTTP service is down on Zabbix server  
Trigger status: PROBLEM  
Trigger severity: Average  
Trigger URL:

Item values:

1. HTTP service is running (Zabbix server:net.tcp.service[http])  
: Down (0)
2. \*UNKNOWN\* (\*UNKNOWN\*: \*UNKNOWN\*): \*UNKNOWN\*
3. \*UNKNOWN\* (\*UNKNOWN\*: \*UNKNOWN\*): \*UNKNOWN\*

Original event ID: 529



## 12.5. MRTG

MRTG（Multi Router Traffic Grapher）是一套可用来绘出网络流量图的软件。

### 12.5.1. 安装MRTG

先[安装Apache httpd](#)。

```
yum -y install net-snmp net-snmp-utils mrtg # 安装MRTG，SNMP
```

配置SNMP（Simple Network Management Protocol简单网络管理协议）：

编辑 /etc/snmp/snmpd.conf 文件：

```
# 注释
#com2sec notConfigUser      default          public

# 取消注释并更改
# “mynetwork”部分更改为自己的本地网络
# 除了“public”或“private”，更改团体名
com2sec  local      localhost      Serverworld
com2sec  mynetwork  10.0.0.0/24  Serverworld

# 取消注释并更改
group MyRWGroup v2c      local
group MyROGroup v2c      mynetwork

# 取消注释
view all      included   .1          80

# 取消注释并更改
access MyROGroup "" v2c      noauth      exact      all      none      none
access MyRWGroup  "" v2c      noauth      exact      all      all      all
```

```
systemctl start snmpd
systemctl enable snmpd
```

验证（将“Serverworld”替换为自己的团体名）：

```
snmpwalk -v2c -c Serverworld localhost system
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Linux dlp.srv.world 3.10.0-229.
4.2.el7.x86_64 #1 SMP Wed May.....
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (91954) 0:15:19
.54
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost> (config
ure /etc/snmp
.....
.....
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.10 = Timeticks: (4) 0:00:00.04
```

配置MRTG：

```
cfgmaker --snmp-options=:::::2 --ifref=descr --ifdesc=descr
Serverworld@10.0.0.30 > /etc/mrtg/mrtg.cfg
```

编辑 /etc/mrtg/mrtg.cfg 文件：

```
# 添加
WorkDir: /var/www/mrtg

# 取消注释
Options[_]: growright, bits

# 取消下行开始的所有行并更改“MaxBytes”的值
Target[10.0.0.30_eth0]: \eth0:Serverworld@10.0.0.30:::::2
noHC[10.0.0.30_eth0]: yes
SetEnv[10.0.0.30_eth0]: MRTG_INT_IP="10.0.0.30" MRTG_INT_DESCR="
eth0"
MaxBytes[10.0.0.30_eth0]: 125000000
Title[10.0.0.30_eth0]: eth0 -- dlp.srv.world
PageTop[10.0.0.30_eth0]: <h1>eth0 -- dlp.srv.world</h1>
.....
.....
```

```
for (( i=1 ; i <= 3 ; i++ )); do env LANG=C mrtg  
/etc/mrtg/mrtg.cfg; done # 执行MRTG三次（显示警告，直到三次）
```

```
2015-06-16 19:54:12, Rateup WARNING: /usr/bin/rateup could not r  
ead the primary log file for 10.0.0.30_eth0  
2015-06-16 19:54:12, Rateup WARNING: /usr/bin/rateup The backup  
log file for 10.0.0.30_eth0 was invalid as well  
2015-06-16 19:54:12, Rateup WARNING: /usr/bin/rateup Can't renam  
e 10.0.0.30_eth0.log to 10.0.0.30_eth0.old updating log file
```

```
indexmaker --columns=1 /etc/mrtg/mrtg.cfg >  
/var/www/mrtg/index.html # 生成索引文件
```

编辑 `/etc/cron.d/mrtg` 文件，添加进Cron：

```
*/5 * * * * root LANG=C LC_ALL=C /usr/bin/mrtg /etc/mrtg/mrtg.cf  
g --lock-file /var/lock/mrtg/mrtg_1 --confcache-file /var/lib/mr  
tg/mrtg.ok
```

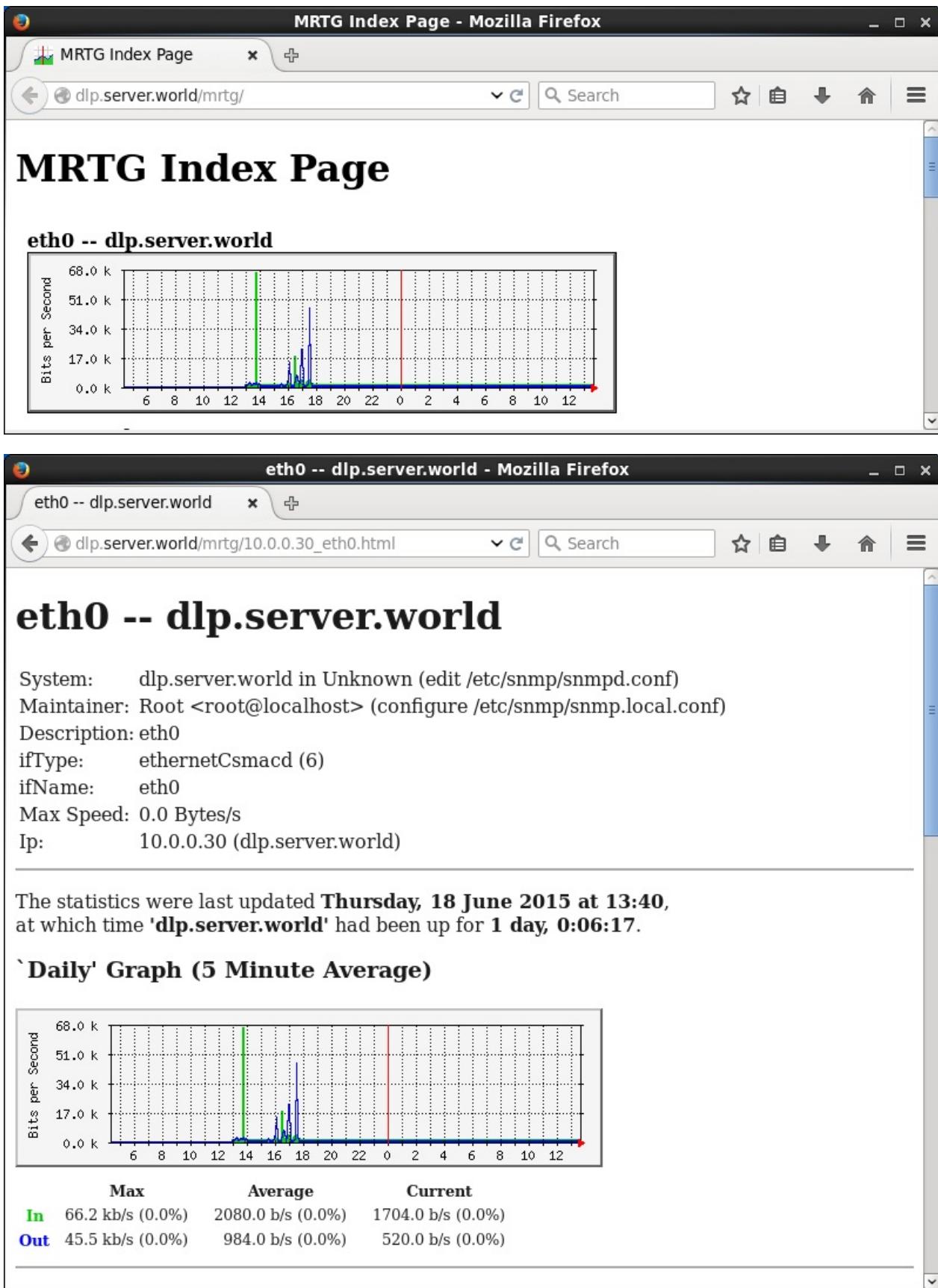
配置httpd从其他主机访问MRTG站点：

编辑 `/etc/httpd/conf.d/mrtg.conf` 文件：

```
# 取消注释并添加访问权限  
Require ip 10.0.0.0/24  
# 添加“DirectoryIndex”  
DirectoryIndex index.html
```

```
systemctl start httpd
```

从客户端浏览器访问 `http://(MRTG主机名或IP地址)/mrtg/`，可以查看MRTG网站：



### 12.5.2. 获取CPU负载平均值

配置MRTG以显示CPU负载平均速率。

编辑 `/etc/mrtg/mrtg.cfg` 文件：

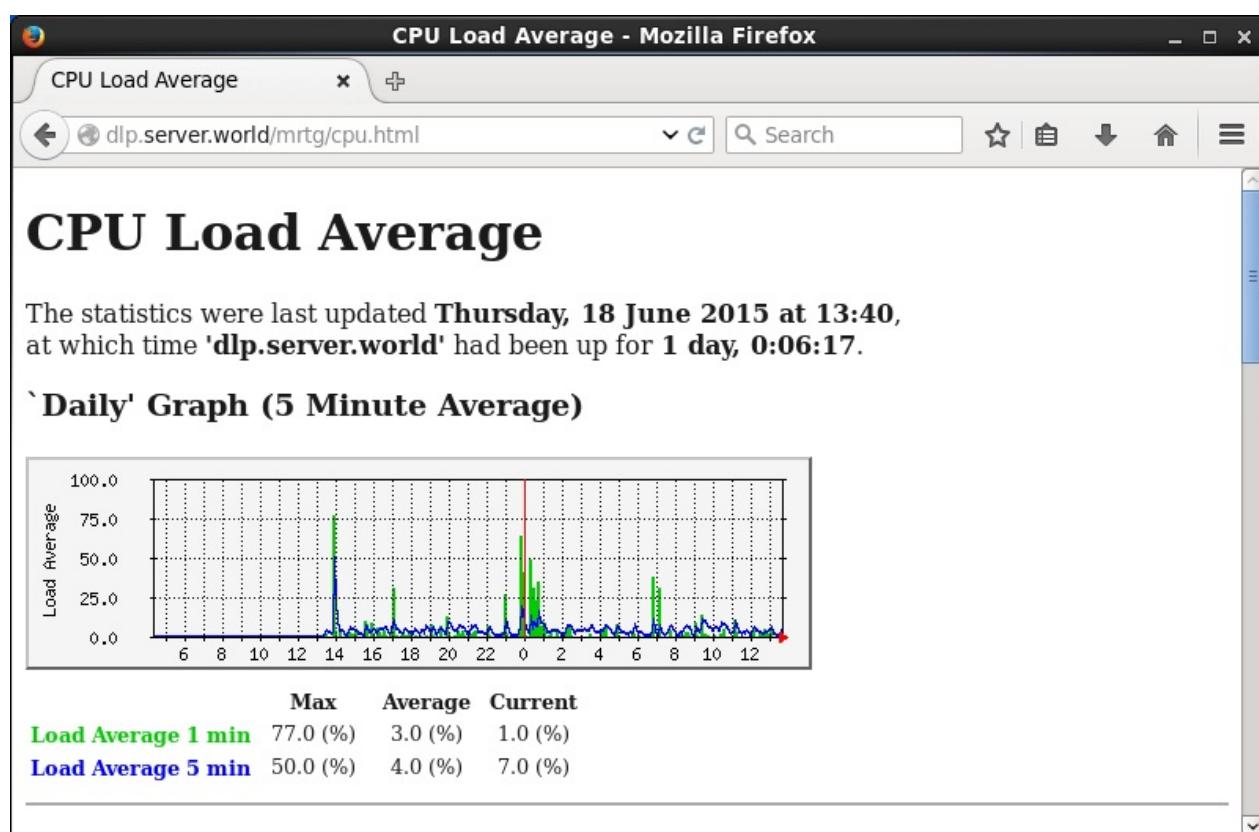
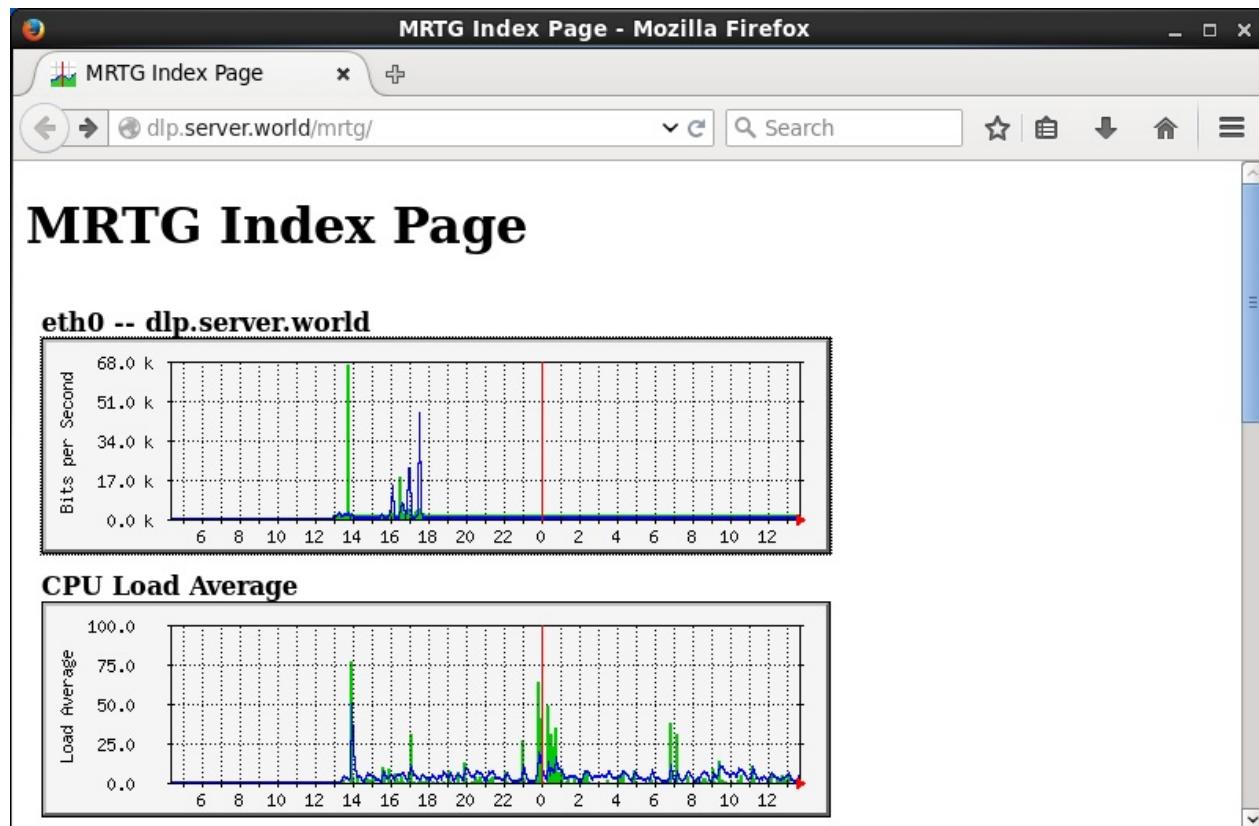
```
# 添加以下内容到最后（将“Serverworld”替换为自己的团体名）
Target[CPU]: .1.3.6.1.4.1.2021.10.1.5.1&.1.3.6.1.4.1.2021.10.1.5
.2:Serverworld@127.0.0.1:::::2
MaxBytes[CPU]: 100
Unscaled[CPU]: dwmy
Options[CPU]: gauge, growright, nopercent
YLegend[CPU]: Load Average
ShortLegend[CPU]: (%) 
LegendI[CPU]: Load Average 1 min
Legend0[CPU]: Load Average 5 min
Legend1[CPU]: Load Average 1 min
Legend2[CPU]: Load Average 5 min
Title[CPU]: CPU Load Average
PageTop[CPU]: <h1>CPU Load Average</h1>
```

```
for (( i=1 ; i <= 3 ; i++ )); do env LANG=C mrtg
/etc/mrtg/mrtg.cfg; done # 执行MRTG三次（显示警告，直到三次）
```

```
2015-06-16 19:20:01, Rateup WARNING: /usr/bin/rateup could not r
ead the primary log file for cpu
2015-06-16 19:20:01, Rateup WARNING: /usr/bin/rateup The backup
log file for cpu was invalid as well
2015-06-16 19:20:01, Rateup WARNING: /usr/bin/rateup Can't renam
e cpu.log to cpu.old updating log file
```

```
indexmaker --columns=1 /etc/mrtg/mrtg.cfg >
/var/www/mrtg/index.html # 生成索引文件
```

从客户端浏览器访问 `http://(MRTG主机名或IP地址)/mrtg/`，可以查看CPU负载平均速率：



### 12.5.3. 获取内存使用率

配置MRTG显示内存使用率。

```
free # 确认总内存
```

|       | total<br>che available | used   | free    | shared | buff/ca |
|-------|------------------------|--------|---------|--------|---------|
| Mem:  | 4047620                | 263968 | 3336184 | 17476  | 447     |
| 468   | 3560284                |        |         |        |         |
| Swap: | 3145724                | 0      | 3145724 |        |         |

编辑 /etc/mrtg/mrtg.cfg 文件:

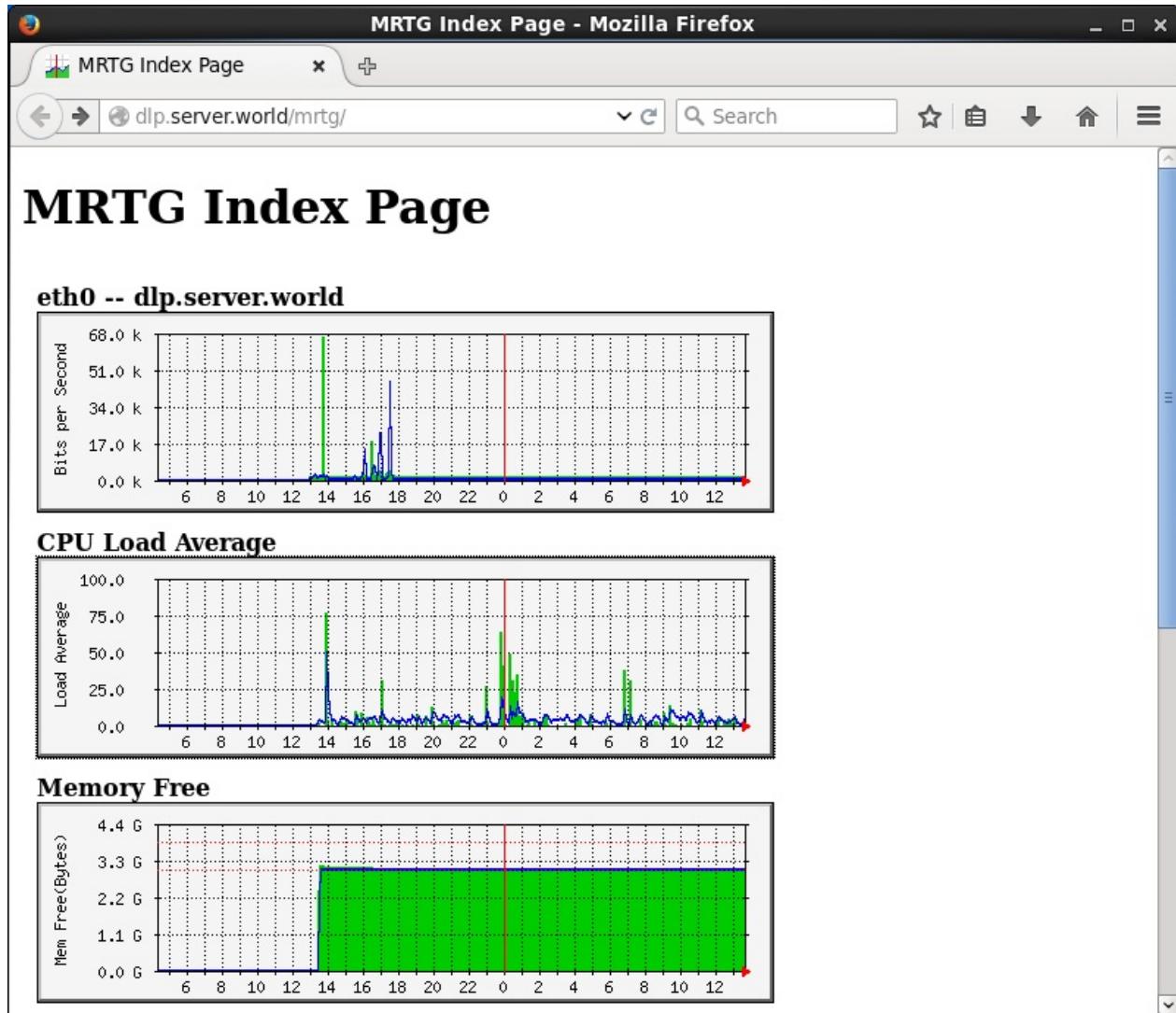
```
# 添加以下内容到最后（将“Serverworld”替换为自己的团体名）
Target[mem]: .1.3.6.1.4.1.2021.4.6.0&.1.3.6.1.4.1.2021.4.4.0:Serverworld@127.0.0.1:::::2
# 总内存
MaxBytes1[Mem]: 4047620
# 总swap
MaxBytes2[Mem]: 3145724
Unscaled[Mem]: dwmly
Options[Mem]: gauge, growright
YLegend[Mem]: Mem Free(Bytes)
ShortLegend[Mem]: Bytes
kilo[Mem]: 1024
kMG[Mem]: k,M,G,T,P
LegendI[Mem]: Real
LegendO[Mem]: Swap
Legend1[Mem]: Memory Free [MBytes]
Legend2[Mem]: Swap Free [MBytes]
Title[Mem]: Memory Free
PageTop[Mem]: <H1>Memory Free</H1>
```

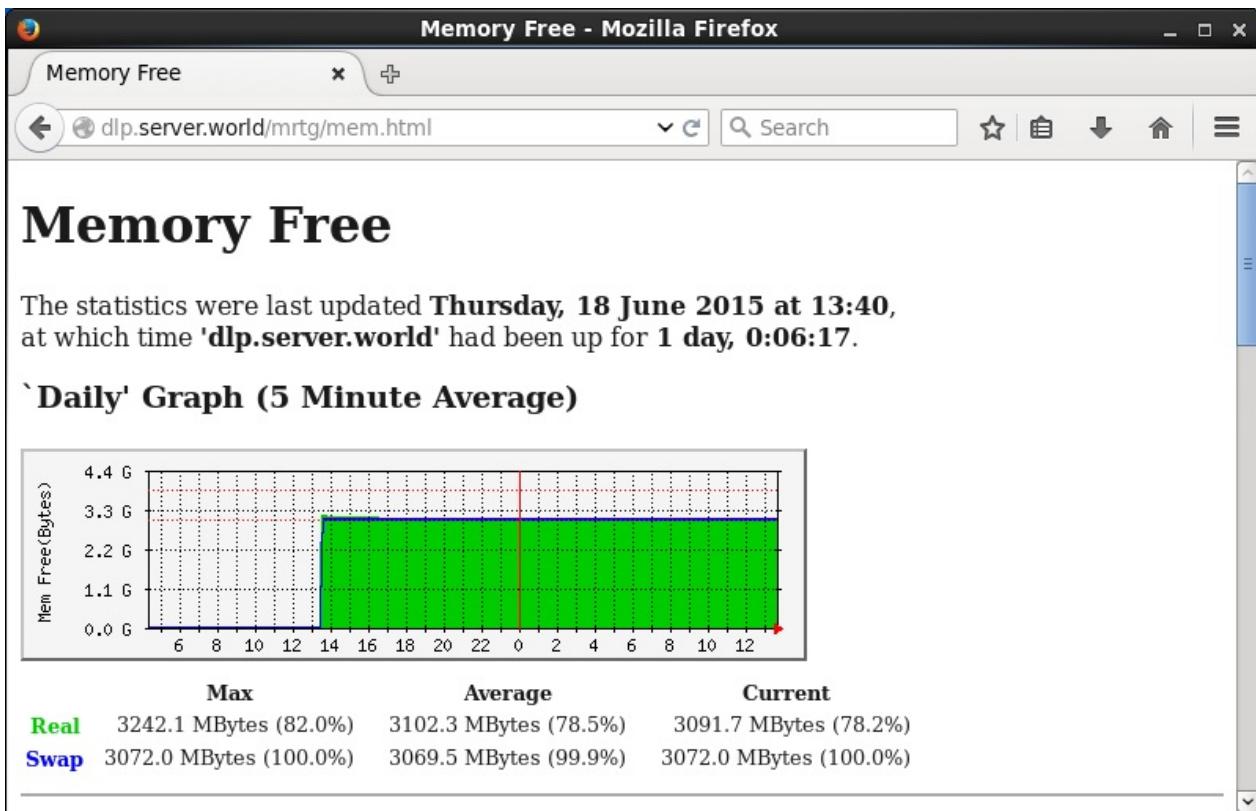
```
for (( i=1 ; i <= 3 ; i++ )); do env LANG=C mrtg
/etc/mrtg/mrtg.cfg; done # 执行MRTG三次（显示警告，直到三次）
```

```
2015-06-16 19:26:12, Rateup WARNING: /usr/bin/rateup could not read the primary log file for mem
2015-06-16 19:26:12, Rateup WARNING: /usr/bin/rateup The backup log file for mem was invalid as well
2015-06-16 19:26:12, Rateup WARNING: /usr/bin/rateup Can't rename mem.log to mem.old updating log file
```

```
indexmaker --columns=1 /etc/mrtg/mrtg.cfg >
/var/www/mrtg/index.html #生成索引文件
```

从客户端浏览器访问 [http://\(MRTG主机名或IP地址\)/mrtg/](http://(MRTG主机名或IP地址)/mrtg/)，可以查看内存使用率：





## 12.5.4. 获取磁盘使用率

配置MRTG显示磁盘使用率（主要显示可用磁盘区）。

编辑 `/etc/snmp/snmpd.conf` 文件：

```
# 取消注释
disk / 10000
```

```
systemctl restart snmpd
```

```
snmpwalk -v2c -c Serverworld localhost .1.3.6.1.4.1.2021.9.1.6 # 确认总磁盘量（将“Serverworld”替换为自己的团体名）
```

```
UCD-SNMP-MIB::dskTotal.1 = INTEGER: 27740944
```

编辑 `/etc/mrtg/mrtg.cfg` 文件：

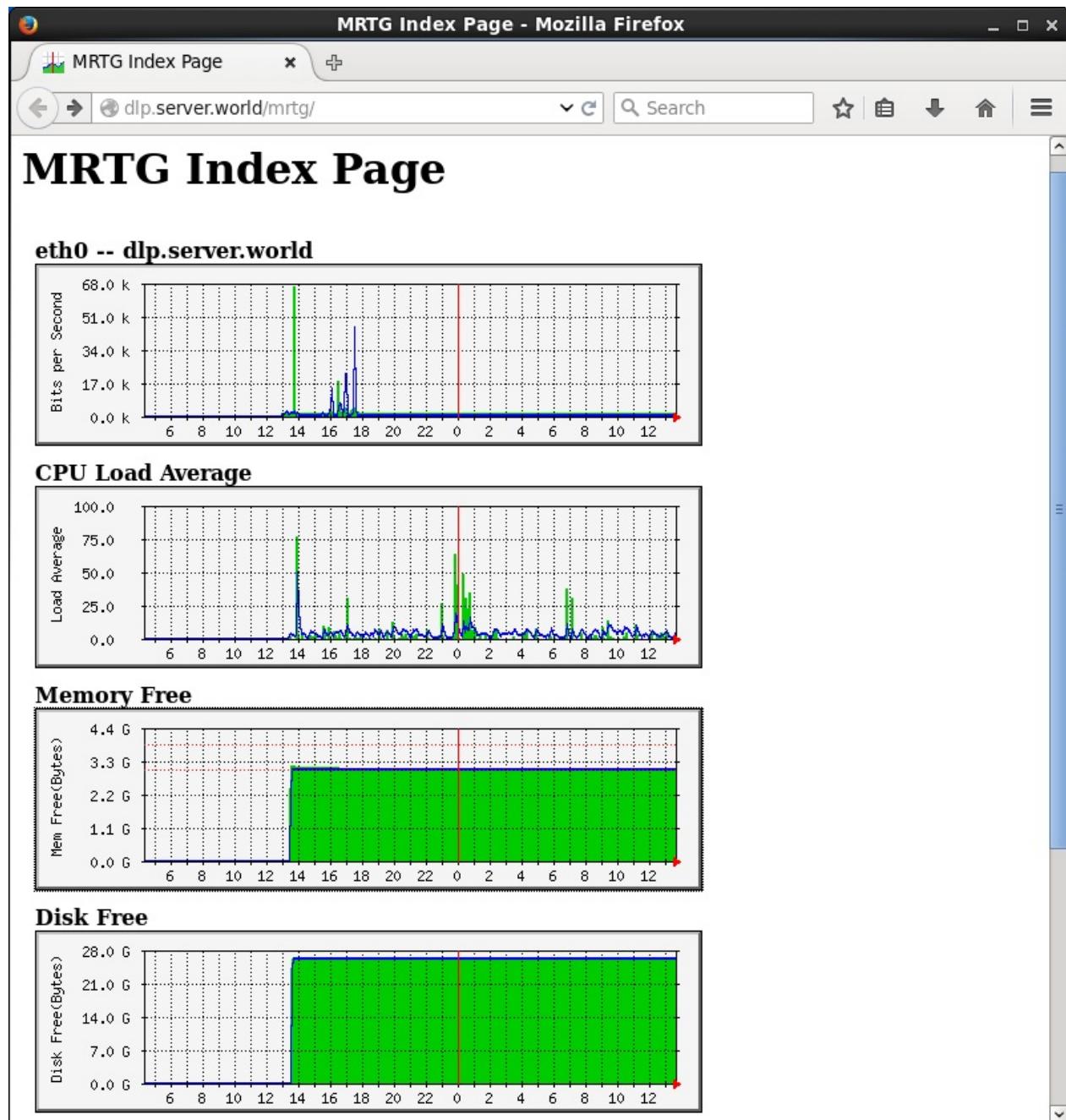
```
# 添加以下内容到最后（将“Serverworld”替换为自己的团体名）
# 在“MaxBytes”部分指定上面的结果
Target[Disk]: .1.3.6.1.4.1.2021.9.1.7.1&.1.3.6.1.4.1.2021.9.1.7.
1:Serverworld@127.0.0.1:::::2
MaxBytes[Disk]: 27740944
kMG[Disk]: k,M,G,T,P
Unscaled[Disk]: dwmy
Options[Disk]: gauge, absolute, growright, nopercent
YLegend[Disk]: Disk Free(Bytes)
ShortLegend[Disk]: Bytes
LegendI[Disk]: / Disk Free [Bytes]
LegendO[Disk]:
Legend1[Disk]: / Disk Free [Bytes]
Legend2[Disk]:
Title[Disk]: Disk Free
PageTop[Disk]: <H1>Disk Free</H1>
```

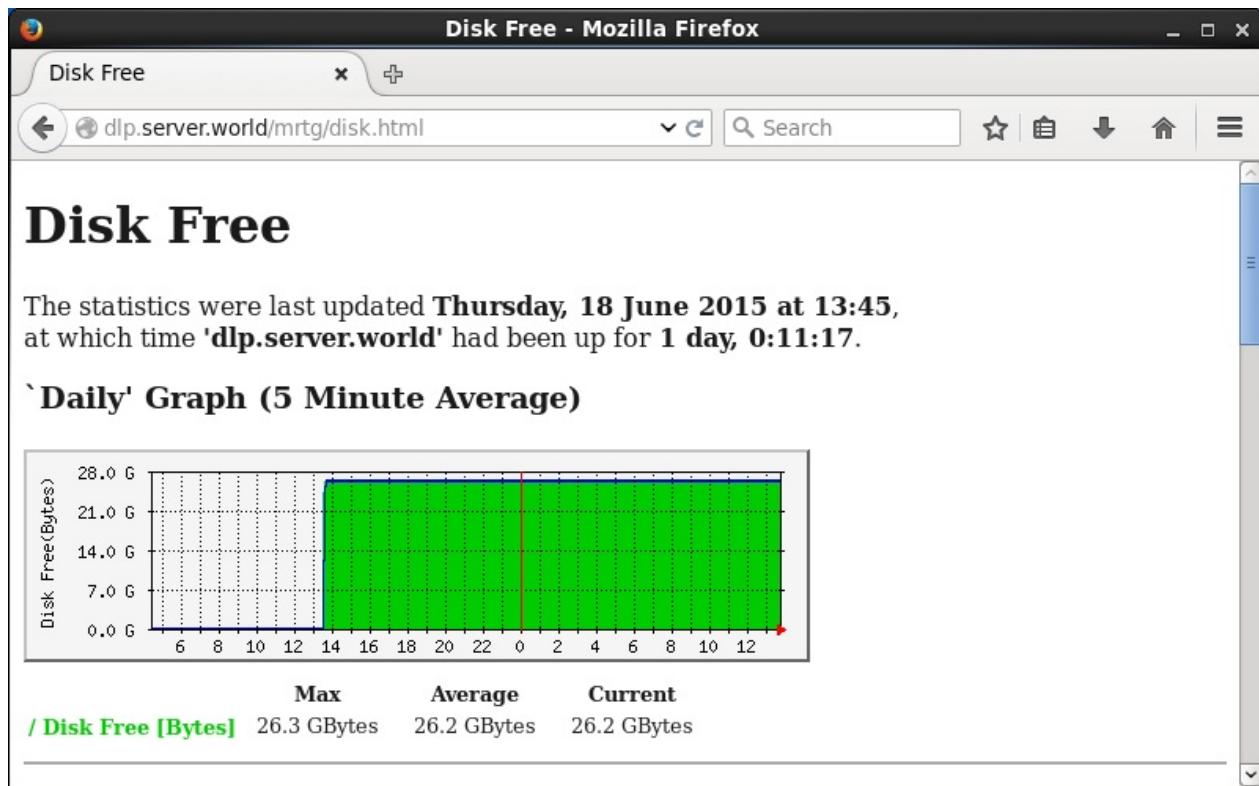
```
for (( i=1 ; i <= 3 ; i++ )); do env LANG=C mrtg
/etc/mrtg/mrtg.cfg; done # 执行MRTG三次（显示警告，直到三次）
```

```
2015-06-16 19:30:23, Rateup WARNING: /usr/bin/rateup could not r
ead the primary log file for disk
2015-06-16 19:30:23, Rateup WARNING: /usr/bin/rateup The backup
log file for disk was invalid as well
2015-06-16 19:30:23, Rateup WARNING: /usr/bin/rateup Can't renam
e disk.log to disk.old updating log file
```

```
indexmaker --columns=1 /etc/mrtg/mrtg.cfg >
/var/www/mrtg/index.html # 生成索引文件
```

从客户端浏览器访问 [http://\(MRTG主机名或IP地址\)/mrtg/](http://(MRTG主机名或IP地址)/mrtg/)，可以查看可用磁盘量：





### 12.5.5. 获取httpd进程

配置MRTG以显示httpd进程数。

编辑 `/etc/snmp/snmpd.conf` 文件：

```
# 添加  
proc httpd
```

```
systemctl restart snmpd
```

编辑 `/etc/mrtg/mrtg.cfg` 文件：

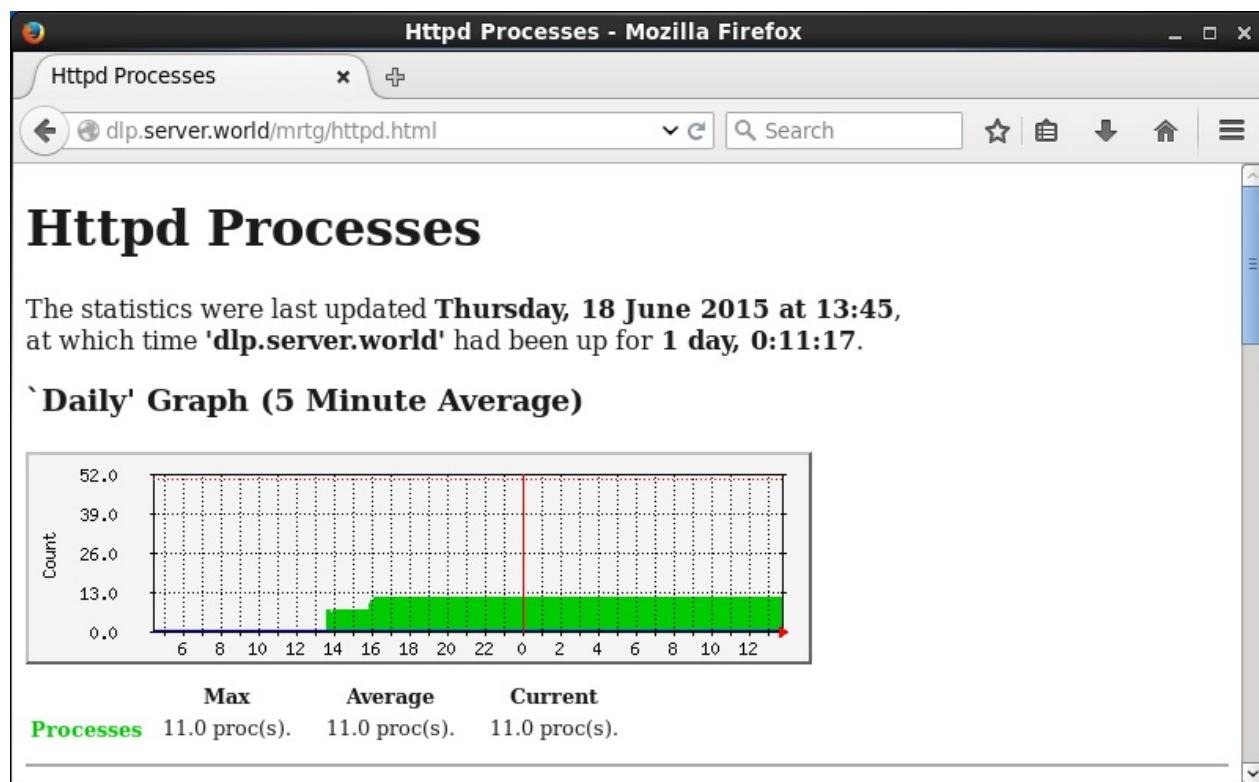
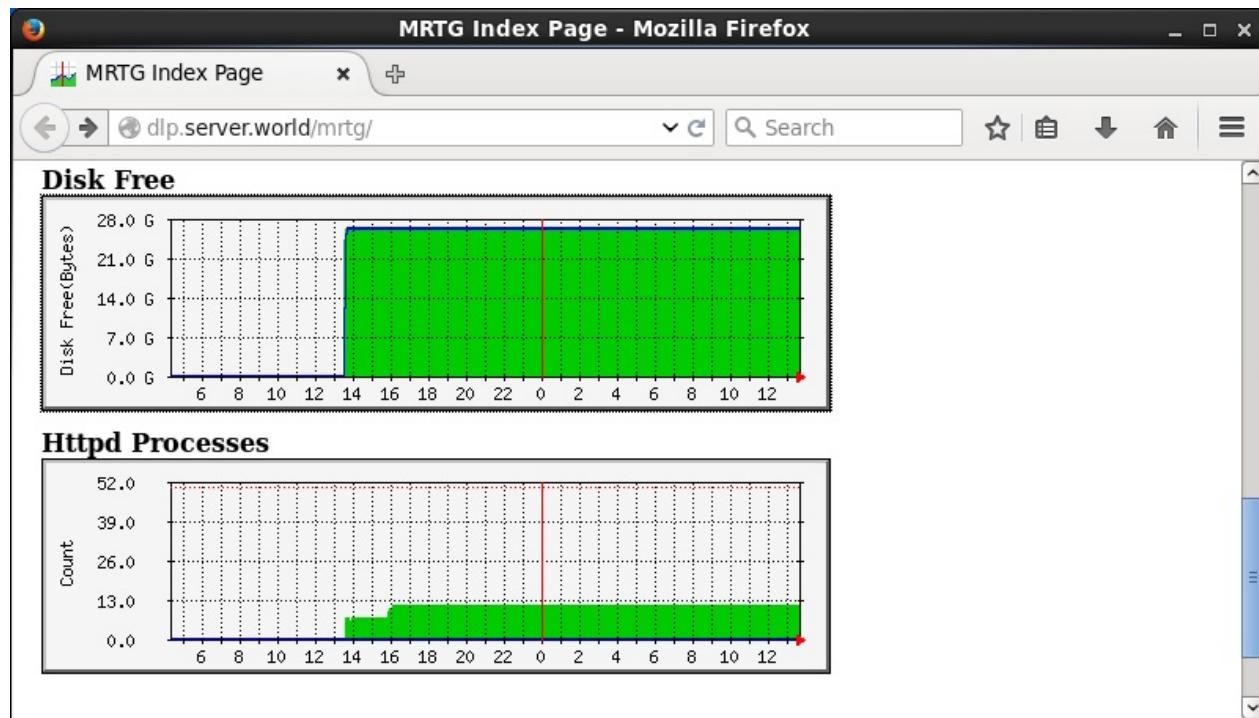
```
# 添加以下内容到最后（将“Serverworld”替换为自己的团体名）
Target[httpd]: .1.3.6.1.4.1.2021.2.1.5.1&.1.3.6.1.4.1.2021.2.1.4
.1:Serverworld@127.0.0.1:::::2
MaxBytes[httpd]: 50
Unscaled[httpd]: dwmy
Options[httpd]: gauge, growright, nopercent
YLegend[httpd]: Count
ShortLegend[httpd]: proc(s).
Title[httpd]: Httpd Processes
LegendI[httpd]: Processes
LegendO[httpd]:
Legend1[httpd]: Httpd Processes
Legend2[httpd]:
PageTop[httpd]: <h1>Httpd Processes</h1>
```

```
for (( i=1 ; i <= 3 ; i++ )); do env LANG=C mrtg
/etc/mrtg/mrtg.cfg; done # 执行MRTG三次（显示警告，直到三次）
```

```
2015-06-16 19:34:41, Rateup WARNING: /usr/bin/rateup could not r
ead the primary log file for httpd
2015-06-16 19:34:41, Rateup WARNING: /usr/bin/rateup The backup
log file for httpd was invalid as well
2015-06-16 19:34:41, Rateup WARNING: /usr/bin/rateup Can't renam
e httpd.log to httpd.old updating log file
```

```
indexmaker --columns=1 /etc/mrtg/mrtg.cfg >
/var/www/mrtg/index.html # 生成索引文件
```

从客户端浏览器访问 `http://(MRTG主机名或IP地址)/mrtg/`，可以查看httpd进程数：



## 12.6. Cacti

Cacti是一套基于PHP，MySQL，SNMP及RRDTool开发的网络流量监测图形分析工具。

### 12.6.1. 安装Cacti

先安装Apache httpd，PHP和MariaDB。

```
yum --enablerepo=epel -y install cacti net-snmp net-snmp-utils php-mysql php-snmp rrdtool
```

从EPEL安装Cacti，SNMP

配置SNMP（Simple Network Management Protocol简单网络管理协议）：

编辑 /etc/snmp/snmpd.conf 文件：

```
# 注释
#com2sec notConfigUser      default          public

# 取消注释并更改
# “mynetwork”部分更改为自己的本地网络
# 除了“public”或“private”，更改团体名
com2sec local      localhost      Serverworld
com2sec mynetwork  10.0.0.0/24  Serverworld

# 取消注释并更改
group MyRWGroup v2c      local
group MyROGroup v2c      mynetwork

# 取消注释
view all      included   .1          80

# 取消注释并更改
access MyROGroup "" v2c      noauth      exact      all      none      none
access MyRWGroup  "" v2c      noauth      exact      all      all      all
```

```
systemctl start snmpd  
systemctl enable snmpd
```

验证（将“Serverworld”替换为自己的团体名）：

```
snmpwalk -v2c -c Serverworld localhost system
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Linux dlp.srv.world 3.10.0-229.  
4.2.el7.x86_64 #1 SMP Wed May.....  
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.  
10  
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (91954) 0:15:19  
.54  
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost> (config  
ure /etc/snmp  
.....  
.....  
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (4) 0:00:00.04  
SNMPv2-MIB::sysORUpTime.10 = Timeticks: (4) 0:00:00.04
```

为Cacti创建数据库并导入表。

```
mysql -u root -p
```

```
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 3342  
Server version: 5.5.41-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
# 创建“cacti”数据库（在“password”部分设置任意密码）  
MariaDB [(none)]> create database cacti;  
Query OK, 1 row affected (0.00 sec)  
MariaDB [(none)]> grant all privileges on cacti.* to cacti@'localhost' identified by 'password';  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> exit  
Bye
```

```
mysql -u cacti -p cacti < /usr/share/doc/cacti-*/cacti.sql
```

```
Enter password: # 上面设置的cacti用户的密码
```

配置Cacti：

编辑 `/etc/cron.d/cacti` 文件：

```
# 取消注释  
*/5 * * * * cacti /usr/bin/php /usr/share/cacti/poller.php > /dev/null 2>&1
```

编辑 `/usr/share/cacti/include/config.php` 文件：

```
# 更改到MariaDB的连接信息
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "password";
$database_port = "3306";
$database_ssl = false;
```

编辑 `/etc/httpd/conf.d/cacti.conf` 文件：

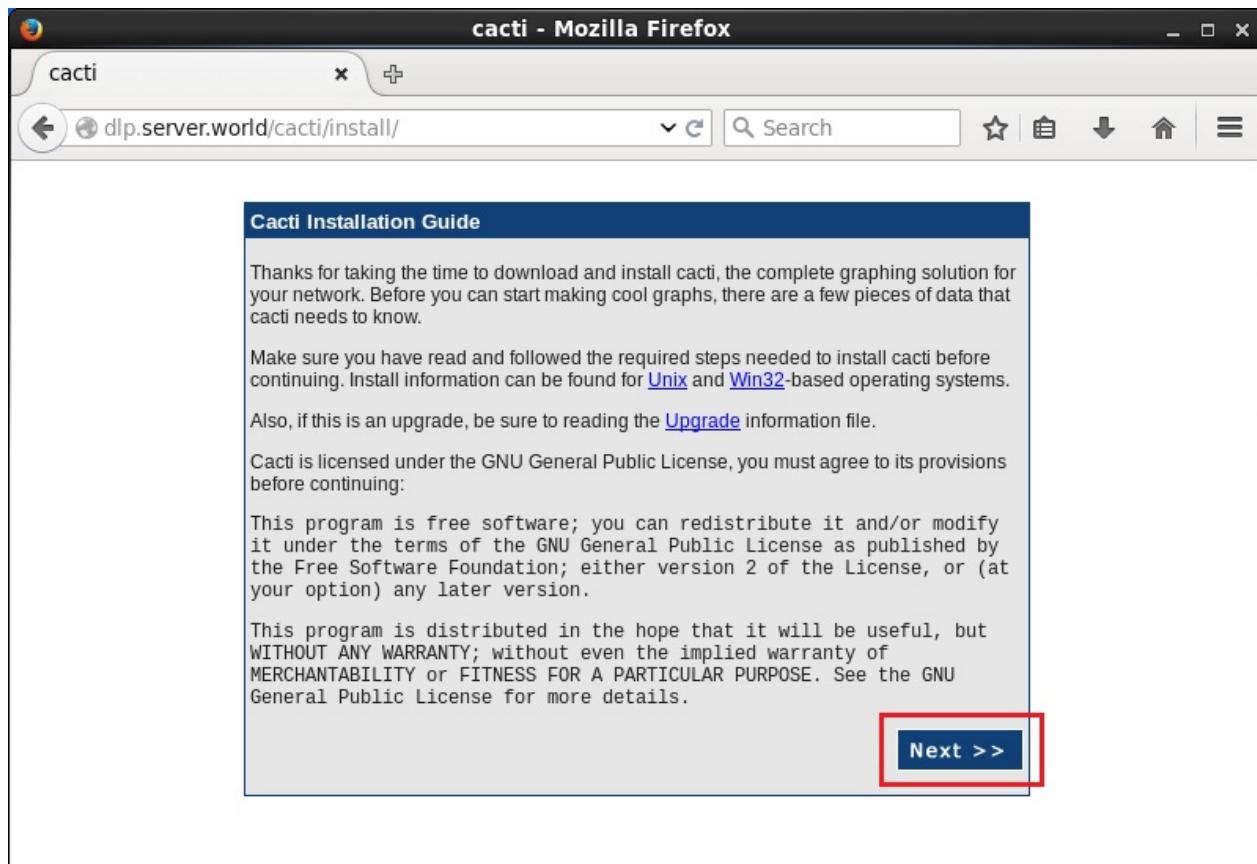
```
# 如果需要，添加访问权限
Require host localhost
Require ip 10.0.0.0/24
```

```
systemctl restart httpd
```

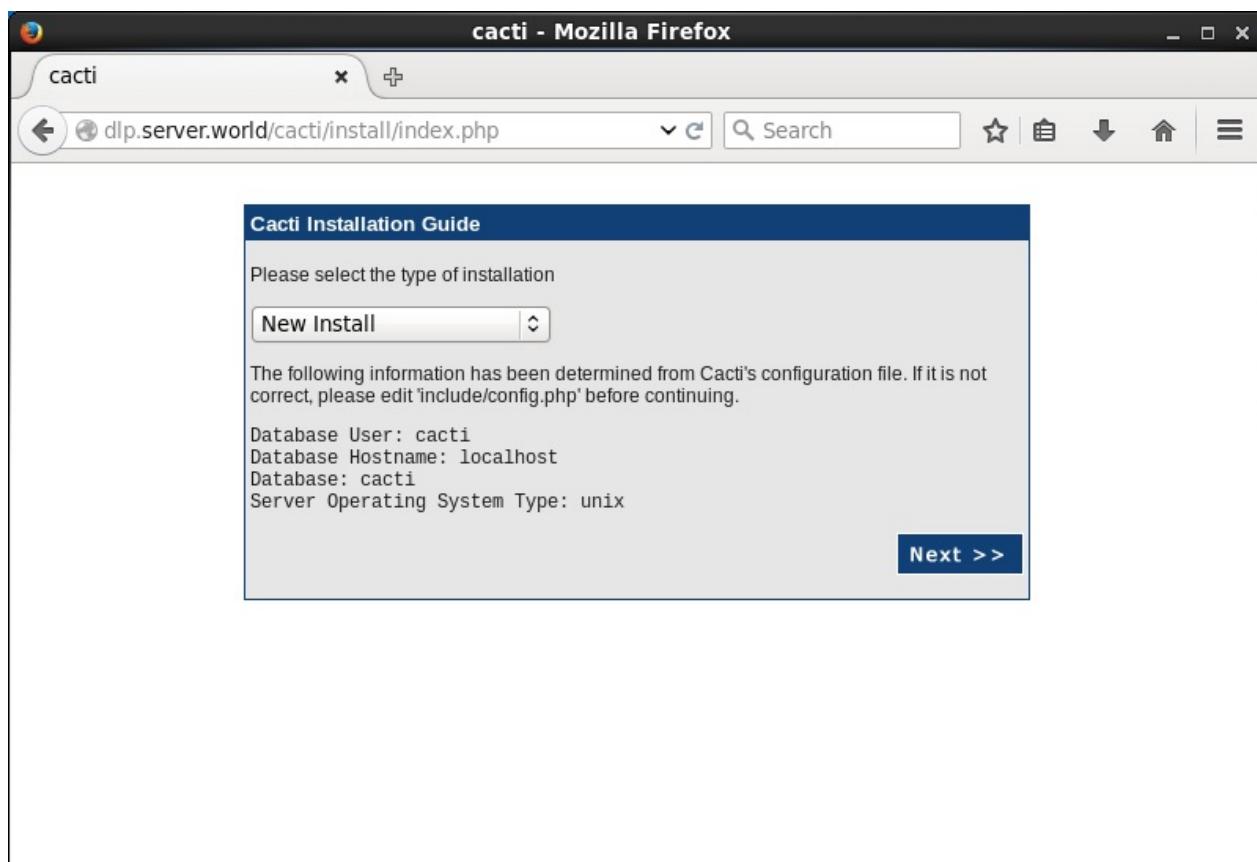
### 12.6.2. Cacti初始设置

从Cacti服务器允许的网络中的客户端访问 `http://(Cacti服务器的主机名或IP地址)/cacti/`，然后进行初始设置运行，点击“Next”继续：

## 12.6. Cacti

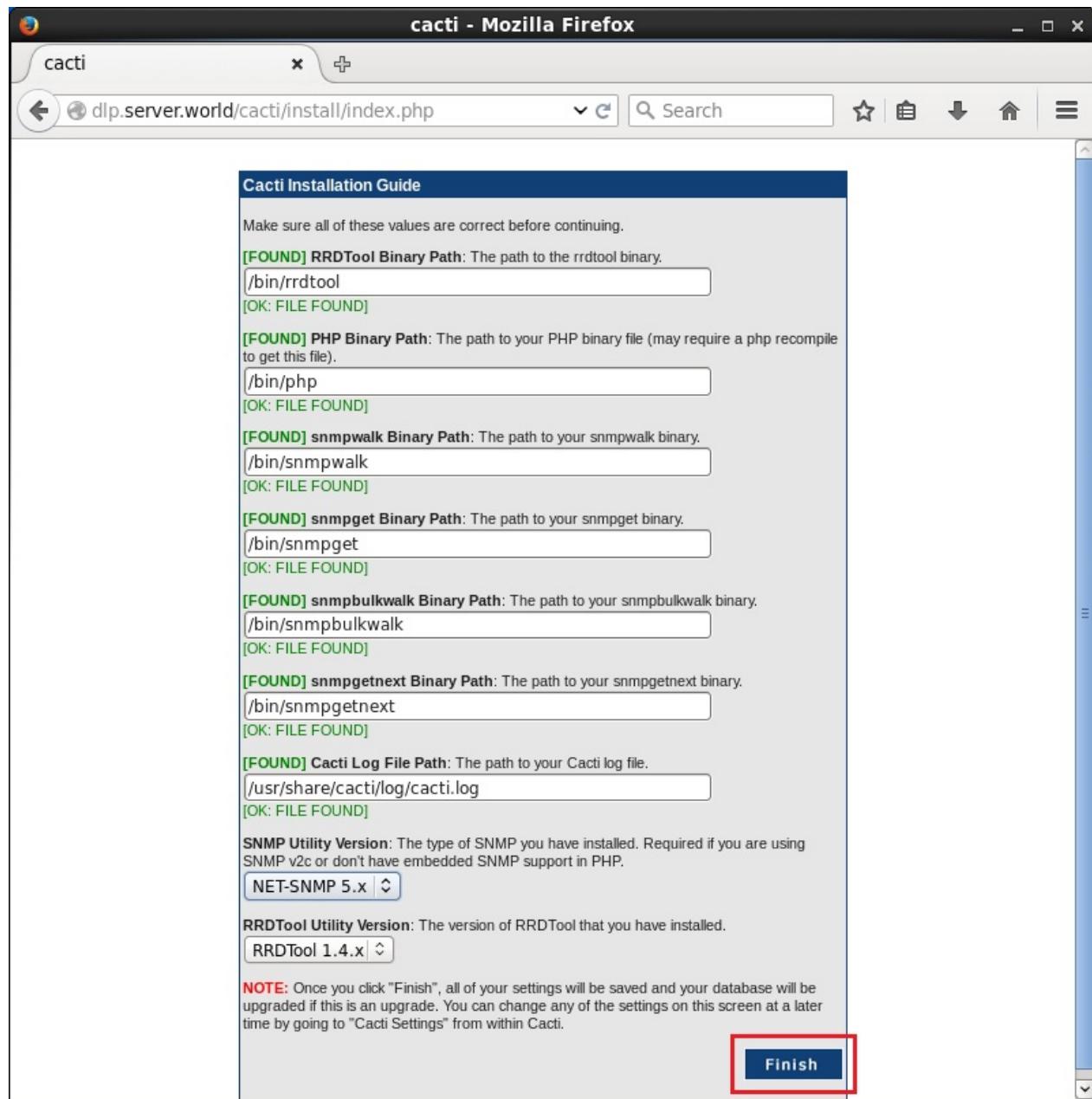


使用默认值继续下一步：



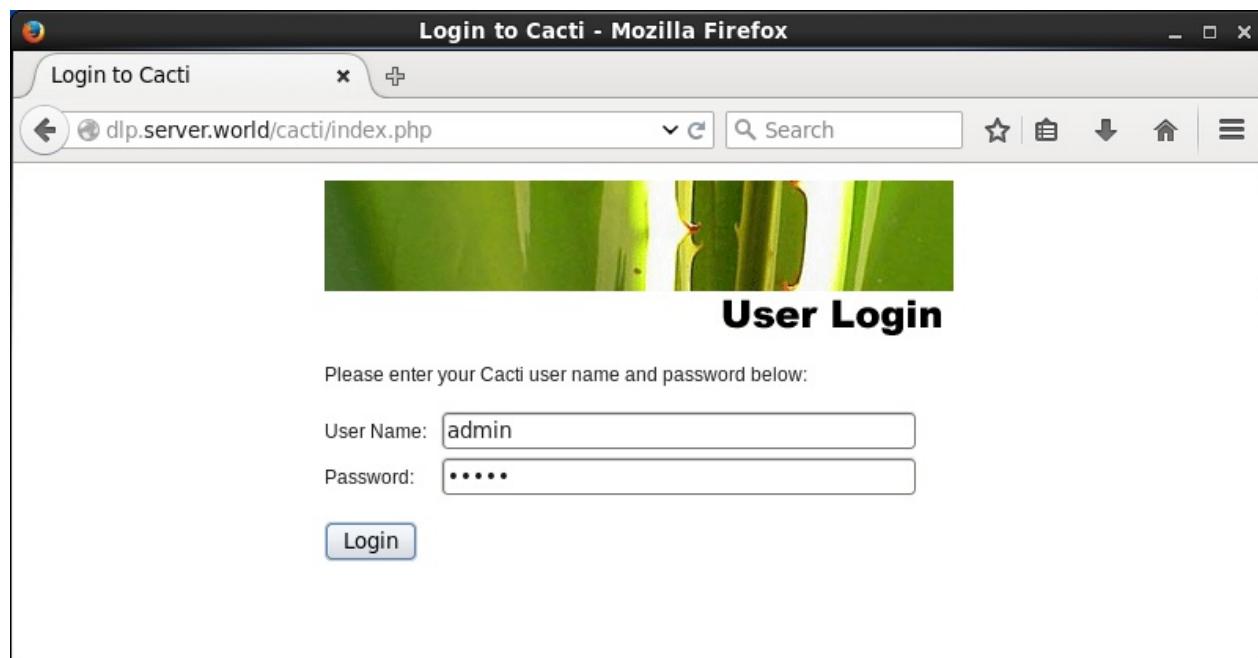
保持默认值，点击“Finish”：

## 12.6. Cacti

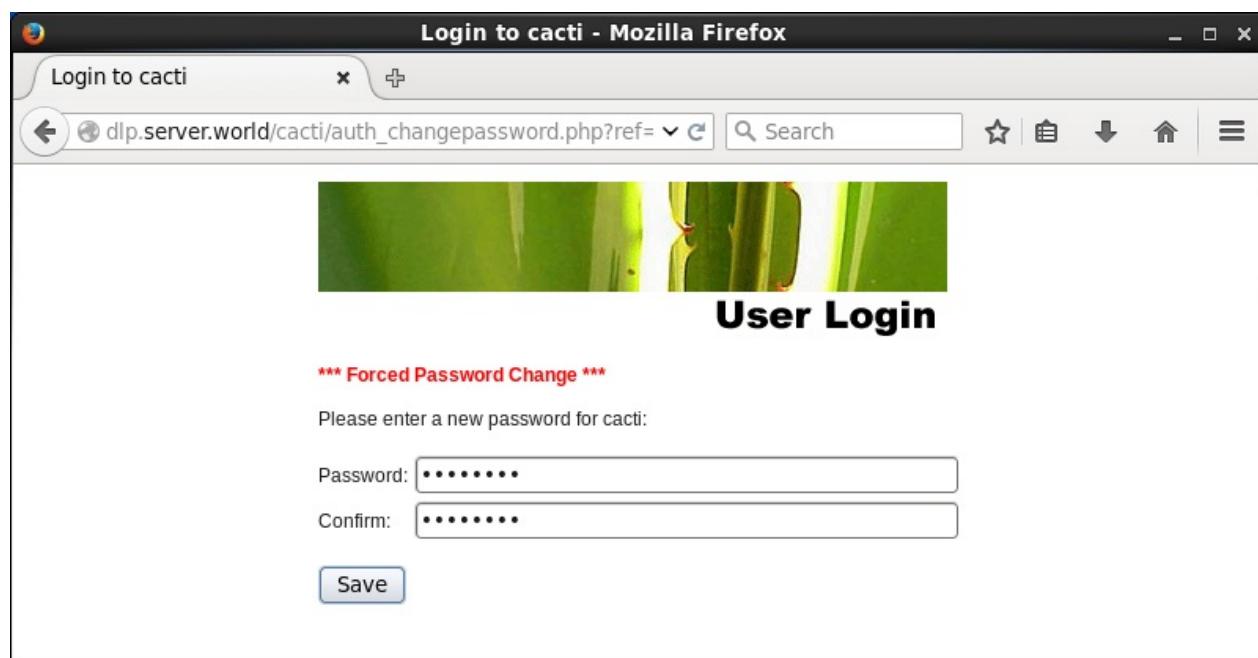


显示登录画面。使用管理员用户“admin”登录，初始密码为“admin”：

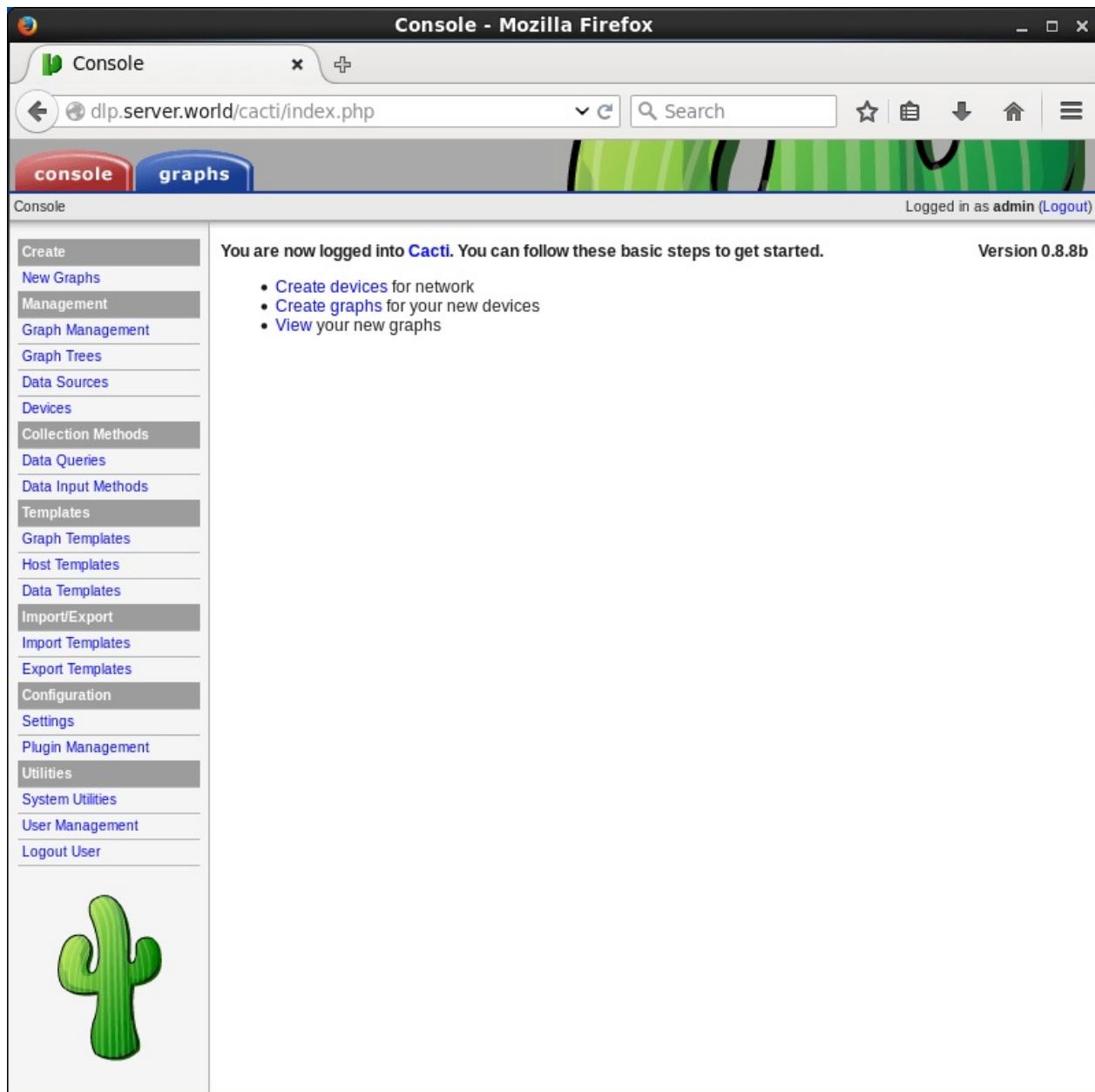
## 12.6. Cacti



登录后，需要更改管理员密码，设置任意密码：



更改密码后，Cacti的主页显示如下：



### 12.6.3. 基本监控设置

登录到Cacti管理网站，然后点击“Create devices”：

## 12.6. Cacti

The screenshot shows the Cacti web interface running in Mozilla Firefox. The title bar reads "Console - Mozilla Firefox". The address bar shows the URL "dlp.server.world/cacti/index.php". The top navigation bar has two tabs: "console" (selected) and "graphs". On the right, it says "Logged in as admin (Logout)". The main content area displays a welcome message: "You are now logged into Cacti. You can follow these basic steps to get started." followed by a bulleted list: "• [Create devices](#) for network", "• [Create graphs](#) for your new devices", and "• [View your new graphs](#)". To the right of this message, it says "Version 0.8.8b". On the left, there is a vertical sidebar with a menu:

- Create
- New Graphs
- Management
- Graph Management
- Graph Trees
- Data Sources
- Devices
- Collection Methods
  - Data Queries
  - Data Input Methods
- Templates
  - Graph Templates
  - Host Templates
  - Data Templates
- Import/Export
  - Import Templates
  - Export Templates
- Configuration
- Settings
- Plugin Management
- Utilities
  - System Utilities
  - User Management
- Logout User

A large green cactus icon is positioned at the bottom of the sidebar.

点击“localhost”：

## 12.6. Cacti

The screenshot shows the Cacti web interface in Mozilla Firefox. The title bar reads "Console -> Devices - Mozilla Firefox". The address bar shows the URL "dlp.server.world/cacti/host.php". The main navigation tabs are "console" (selected) and "graphs". On the left, a sidebar menu lists various management options under "Management": Create, New Graphs, Management, Graph Management, Graph Trees, Data Sources, Devices (highlighted), Collection Methods, Data Queries, Data Input Methods, Templates, Graph Templates, Host Templates, Data Templates, Import/Export (highlighted), Import Templates, Export Templates, Configuration, Settings, Plugin Management, Utilities, System Utilities, User Management, and Logout User. Below the sidebar is a large green cactus icon. The main content area is titled "Devices" and displays a table with one row. The table columns are Description\*\*, ID, Graphs, Data Sources, Status, In State, Hostname, and Current. The single row shows "localhost" in the Description column, with ID 1, 4 graphs, 5 data sources, Up status, and 127.0.0.1 as the Hostname. A "Choose an action:" dropdown is visible on the right. Navigation links "<< Previous" and "Showing Rows 1 to 1 of 1 [1]" are at the top and bottom of the table respectively.

指定SNMP版本“2”，并将团体名更改为在 `snmpd.conf` 中设置的团体名：

## 12.6. Cacti

The screenshot shows the Cacti web interface for editing a device. The title bar reads "Console -> Devices -> (Edit) - Mozilla Firefox". The URL in the address bar is "dlp.server.world/cacti/host.php?action=edit&id=1". The top navigation bar has tabs for "console" and "graphs", with "console" being active. A sidebar on the left lists various management options under "Devices". The main content area is titled "localhost (127.0.0.1)". It displays a "Ping Results" section showing "UDP Ping Success (0.12 ms)". Below this is a "Devices [edit: localhost]" section with several configuration tabs: "General Host Options", "Number of Collection Threads", "Disable Host", "Availability/Reachability Options", "Downed Device Detection", "Ping Timeout Value", "Ping Retry Count", "SNMP Options", and "Additional Options". The "SNMP Options" tab is currently selected. In this tab, the "SNMP Version" dropdown is set to "Version 2" and the "SNMP Community" input field is highlighted with a red box and contains the value "Serverworld". Other fields in this tab include "SNMP Port" (161), "SNMP Timeout" (500), and "Maximum OID's Per Get Request" (10). The right side of the interface shows a green cactus icon.

向下滚动并点击“Save”：

## 12.6. Cacti

The screenshot shows the Cacti web interface for editing a device. The left sidebar has a 'Configuration' section with links for Settings, Plugin Management, Utilities (selected), System Utilities, User Management, and Logout User. A green cactus icon is displayed. The main content area is titled 'Console -> Devices -> (Edit) - Mozilla Firefox'. It shows configuration sections for Downed Device Detection, SNMP Uptime (400), Ping Timeout Value (400), Ping Retry Count (1), and various SNMP options like Version 2, Community (Serverworld), Port (161), Timeout (500), and Maximum OID's Per Get Request (10). There is a 'Notes' section for entering notes about the host. Below these are sections for 'Associated Graph Templates' (listing four templates: Linux - Memory Usage, Unix - Load Average, Unix - Logged in Users, Unix - Processes, all marked as 'Is Being Graphed') and 'Associated Data Queries' (listing one query: Unix - Get Mounted Partitions, marked as Success [4 Items, 2 Rows]). At the bottom right are 'Return' and 'Save' buttons, with 'Save' being highlighted by a red box.

默认情况下会创建负载平均值，内存使用率，登录用户，CPU使用率的图表。对于磁盘使用率，选中要监控的设备上的复选框。全部完成后，点击“OK”：

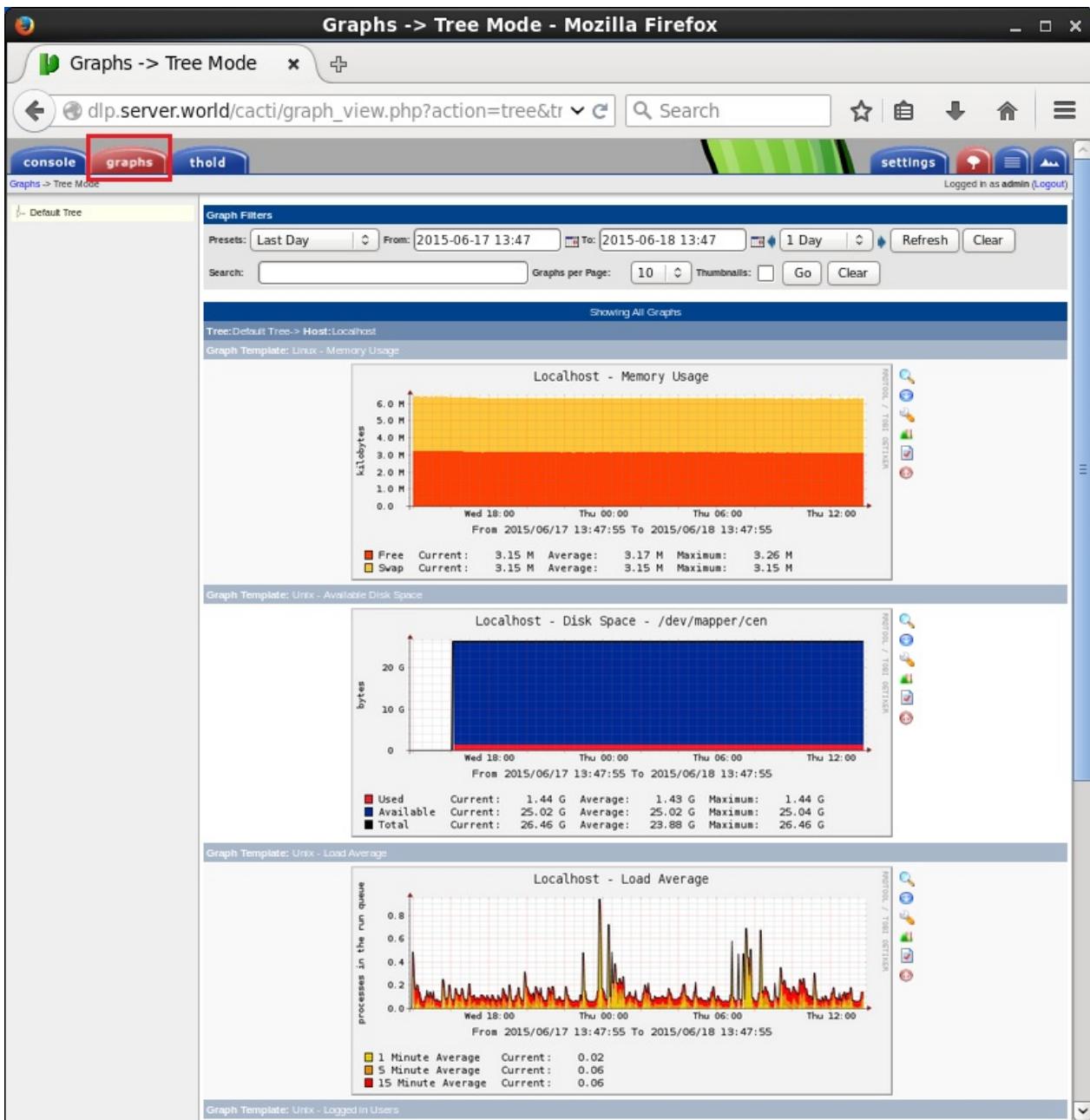
## 12.6. Cacti

The screenshot shows the 'Create New Graphs' page of the Cacti web interface. The URL is `dlp.server.world/cacti/graphs_new.php`. The left sidebar has a 'Create' section with 'New Graphs' highlighted (boxed in red). The main area shows 'localhost (127.0.0.1)' selected as the host (boxed in red). A 'Graph Templates' section lists several pre-defined templates. Below it is a 'Data Query [Unix - Get Mounted Partitions]' table:

| Device Name             | Mount Point |
|-------------------------|-------------|
| /dev/mapper/centos-root | /           |
| /dev/vda1               | /boot       |

At the bottom right of the table are 'Cancel' and 'Create' buttons, with 'Create' boxed in red.

几分钟后，转到“graphs”标签，如下所示查看系统状态：



### 12.6.4. 电子邮件通知设置

下载插件前到[官网插件页面](#)确认为最新版本。

```
wget -P /usr/share/cacti/plugins
http://docs.cacti.net/_media/plugin:settings-v0.71-1.tgz

tar zxvf /usr/share/cacti/plugins/plugin:settings-v0.71-1.tgz -C
/usr/share/cacti/plugins
```

登录到Cacti管理网站，点击左侧菜单上的“Plugin Management”，然后点击右侧窗格中的“Settings”字段上的图标：

## 12.6. Cacti

The screenshot shows the 'Plugin Management' page of the Cacti web interface. The URL is `dlp.server.world/cacti/plugins.php`. The left sidebar has a red box around the 'Plugin Management' item. The main content area displays a table with one row:

| Actions | Name     | Version | Load Order | Description**          | Type   | Status        | Author       |
|---------|----------|---------|------------|------------------------|--------|---------------|--------------|
|         | Settings | 0.71    |            | Global Plugin Settings | System | Not Installed | Jimmy Conner |

Below the table, there are two notes:  
NOTE: Please sort by 'Load Order' to change plugin load ordering.  
NOTE: SYSTEM plugins can not be ordered.

点击光标按钮以启用插件：

## 12.6. Cacti

The screenshot shows the 'Plugin Management' page of the Cacti web interface. The URL in the address bar is `dlp.server.world/cacti/plugins.php`. The left sidebar has a 'Plugin Management' section selected. The main content area displays a table titled 'Plugin Management (Cacti Version: 0.8.8b, Plugin Architecture Version: 3.1)'. The table has columns: Actions, Name, Version, Load Order, Description\*\*, Type, Status, and Author. One row is listed: 'Settings' (Version 0.71, Global Plugin Settings, System, Installed, Jimmy Conner). A red box highlights the 'Actions' column header. Below the table, notes say 'NOTE: Please sort by 'Load Order' to change plugin load ordering.' and 'NOTE: SYSTEM plugins can not be ordered.'

| Actions | Name     | Version | Load Order | Description**          | Type   | Status    | Author       |
|---------|----------|---------|------------|------------------------|--------|-----------|--------------|
|         | Settings | 0.71    |            | Global Plugin Settings | System | Installed | Jimmy Conner |

启用后，状态变为“Active”：

## 12.6. Cacti

The screenshot shows the 'Console -> Plugin Management' page in Mozilla Firefox. The URL is `dlp.server.world/cacti/plugins.php`. The left sidebar has a 'Plugin Management' section selected. The main content area displays a table of plugins:

| Actions | Name     | Version | Load Order | Description**          | Type   | Status | Author       |
|---------|----------|---------|------------|------------------------|--------|--------|--------------|
|         | Settings | 0.71    |            | Global Plugin Settings | System | Active | Jimmy Conner |

Notes at the bottom: 'NOTE: Please sort by 'Load Order' to change plugin load ordering.' and 'NOTE: SYSTEM plugins can not be ordered.'

点击左侧菜单上的“Settings”，转到右侧窗格中的“Mail/DNS”标签，并输入以下项目，然后点击“Save”：

Test Mail -> 测试邮件的收件人地址

Mail Services -> 用来发送电子邮件的服务

From Email Address -> 发件人邮件地址

From Name -> 发件人名称

## 12.6. Cacti

The screenshot shows the Cacti Settings interface in Mozilla Firefox. The left sidebar has a 'Settings' link highlighted with a red box. The main content area shows the 'Mail / DNS' tab selected. A red box highlights the 'Send a Test Email' button. Another red box highlights the 'Save' button at the bottom right.

成功发送电子邮件后，显示“Success”：



### 12.6.5. 启用阈值插件

启用阈值插件以设置阈值。

下载插件前到[官网插件页面](#)确认为最新版本。

```
wget -P /usr/share/cacti/plugins
http://docs.cacti.net/_media/plugin:thold-v0.5.0.tgz

tar zxvf /usr/share/cacti/plugins/plugin:thold-v0.5.0.tgz -C
/usr/share/cacti/plugins
```

登录到Cacti管理网站，点击左侧菜单上的“Plugin Management”，然后点击右侧窗格中的“Thold”字段上的图标：

## 12.6. Cacti

The screenshot shows the Cacti Plugin Management interface within a Mozilla Firefox browser window. The title bar reads "Console > Plugin Management - Mozilla Firefox". The address bar shows the URL "dlp.server.world/cacti/plugins.php". The left sidebar has a red border around the "Plugin Management" item, which is currently selected. Other items in the sidebar include "Create", "New Graphs", "Management", "Graph Management", "Graph Trees", "Data Sources", "Devices", "Collection Methods", "Data Queries", "Data Input Methods", "Templates", "Graph Templates", "Host Templates", "Data Templates", "Import/Export", "Import Templates", "Export Templates", "Configuration", "Settings", "Plugin Management" (selected), "Utilities", "System Utilities", "User Management", and "Logout User". The main content area displays a table titled "Plugin Management (Cacti Version: 0.8.8b, Plugin Architecture Version: 3.1)". The table has columns: Actions, Name, Version, Load Order, Description\*\*, Type, Status, and Author. It lists two rows: "Settings" (Version 0.71, Global Plugin Settings, System, Active, Jimmy Conner) and "Thold" (Version 0.5, Thresholds, General, Not Installed, Jimmy Conner). Below the table, there are two notes: "NOTE: Please sort by 'Load Order' to change plugin load ordering." and "NOTE: SYSTEM plugins can not be ordered.".

| Showing All 2 Rows |          |         |            |                        |         |               |              |
|--------------------|----------|---------|------------|------------------------|---------|---------------|--------------|
| Actions            | Name     | Version | Load Order | Description**          | Type    | Status        | Author       |
|                    | Settings | 0.71    |            | Global Plugin Settings | System  | Active        | Jimmy Conner |
|                    | Thold    | 0.5     |            | Thresholds             | General | Not Installed | Jimmy Conner |

点击光标按钮以启用插件：

## 12.6. Cacti

The screenshot shows the 'Plugin Management' page of the Cacti web interface. The browser title bar reads 'Console -> Plugin Management - Mozilla Firefox'. The address bar shows the URL 'dlp.server.world/cacti/plugins.php'. The main content area is titled 'Plugin Management (Cacti Version: 0.8.8b, Plugin Architecture Version: 3.1)'. A sidebar on the left contains links for 'Create', 'New Graphs', 'Management', 'Graph Management', 'Graph Trees', 'Data Sources', 'Devices', 'Collection Methods', 'Data Queries', 'Data Input Methods', 'Templates', 'Graph Templates', 'Host Templates', 'Data Templates', 'Import/Export', 'Import Templates', 'Export Templates', 'Configuration', 'Settings', 'Plugin Management' (which is highlighted in blue), 'Utilities', 'System Utilities', 'User Management', and 'Logout User'. The 'Plugin Management' section displays a table with two rows:

| Actions | Name     | Version | Load Order | Description**          | Type    | Status    | Author       |
|---------|----------|---------|------------|------------------------|---------|-----------|--------------|
|         | Settings | 0.71    |            | Global Plugin Settings | System  | Active    | Jimmy Conner |
|         | Thold    | 0.5     |            | Thresholds             | General | Installed | Jimmy Conner |

Below the table, there are two notes: 'NOTE: Please sort by 'Load Order' to change plugin load ordering.' and 'NOTE: SYSTEM plugins can not be ordered.'

启用后，状态变为“Active”：

## 12.6. Cacti

The screenshot shows the 'Plugin Management' page of the Cacti web interface. The browser title bar reads 'Console -> Plugin Management - Mozilla Firefox'. The address bar shows the URL 'dlp.server.world/cacti/plugins.php'. The main content area displays a table of installed plugins:

| Actions                           | Name     | Version | Load Order | Description**          | Type    | Status        | Author       |
|-----------------------------------|----------|---------|------------|------------------------|---------|---------------|--------------|
| <a href="#"></a> <a href="#"></a> | Settings | 0.71    |            | Global Plugin Settings | System  | Active        | Jimmy Conner |
| <a href="#"></a> <a href="#"></a> | Thold    | 0.5     |            | Thresholds             | General | <b>Active</b> | Jimmy Conner |

Below the table, there are two notes: 'NOTE: Please sort by 'Load Order' to change plugin load ordering.' and 'NOTE: SYSTEM plugins can not be ordered.'

The left sidebar contains a navigation menu with the following items:

- Create
- New Graphs
- Management
- Graph Management
  - Graph Trees
  - Data Sources
  - Devices
  - Notification Lists
  - Thresholds
- Collection Methods
  - Data Queries
  - Data Input Methods
- Templates
  - Graph Templates
  - Host Templates
  - Data Templates
  - Threshold Templates
- Import/Export
  - Import Templates
  - Export Templates
- Configuration
  - Settings
- Plugin Management**
- Utilities
  - Custom Utilities

点击左侧菜单上的“Notification Lists”，然后点击右侧窗格中的“Add”：

## 12.6. Cacti

The screenshot shows the Cacti web interface in Mozilla Firefox. The title bar reads "Console -> Notification Lists - Mozilla Firefox". The left sidebar menu is visible, with "Notification Lists" highlighted and a red box drawn around it. The main content area displays a table titled "Notification Lists" with one row: "No Notification Lists". A red box highlights the "Add" button in the top right corner of the table header. The status bar at the bottom right says "Logged in as admin (Logout)".

在“Name”字段上输入任意名称，在“Description”字段中输入任意描述，在“Email Address”字段中输入目标电子邮件地址，然后点击“Create”：

The screenshot shows the Cacti web interface in Mozilla Firefox, displaying the "General Settings [new]" dialog for creating a notification list. The dialog has three main sections: "Name" (input field: "Cacti Admin"), "Description" (input field: "Cacti Administrator"), and "Email Addresses" (input field: "root@localhost"). At the bottom right of the dialog are "Cancel" and "Create" buttons. The status bar at the bottom right says "Logged in as admin (Logout)".

## 12.6. Cacti

点击左侧菜单上的“Settings”，并转到右侧窗格中的“Thresholds”标签：

Console -> Cacti Settings - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Console -> Cacti Settings

dlp.server.world/cacti/settings.php?tab=alerts

console graphs thold

Logged in as admin (Logout)

Create New Graphs Management

Graph Management

Graph Trees Data Sources Devices

Notification Lists Thresholds

Collection Methods Data Queries Data Input Methods

Templates

Graph Templates Host Templates Data Templates Threshold Templates

Import/Export Import Templates Export Templates Configuration

**Settings**

Plugin Management Utilities System Utilities User Management Logout User

**General** **Paths** **Poller** **Graph Export** **Visual** **Authentication** **Mail / DNS** **Thresholds**

**Cacti Settings (Thresholds)**

**General**

**Disable All Thresholds**  
Checking this box will disable Alerting on all Thresholds. This can be used when it is necessary to perform maintenance on your network.  
 Disable All Thresholds

**Disable Legacy Notifications**  
Checking this box will disable Legacy Alerting on all Thresholds. Legacy Alerting is defined as any Specific Email Alerts not associated with a Notification List.  
 Disable Legacy Notifications

**Default Status**  
Default Threshold Filter Status  
Any

**Base URL**  
Cacti base URL  
http://dlp.server.world/cacti/

**Thresholds Per Page**  
Number of thresholds to display per page  
30

**Log Threshold Breaches**  
Enable logging of all Threshold failures to the Cacti Log  
 Log Threshold Breaches

**Log Threshold Changes**  
Enable logging of all Threshold changes to the Cacti Log  
 Log Threshold Changes

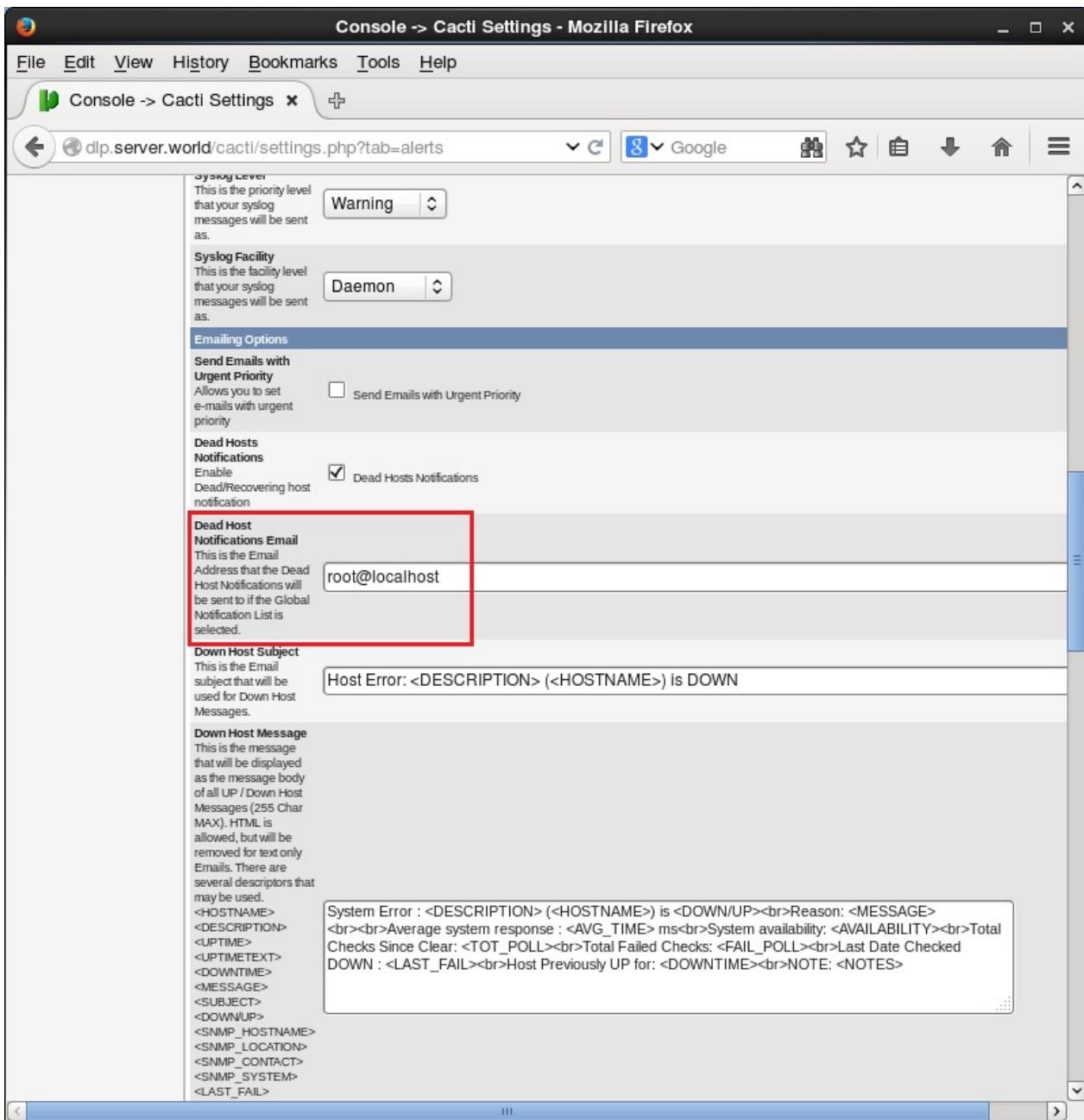
**Debug Log**  
Enable logging of debug messages with Thold  
 Debug Log

**Store Log for x days**  
Keep the database threshold logs for this number of days.  
31

**Default Alerting Options**

向下滚动并在“Dead Host Notification Email”字段上输入通知电子邮件地址，然后点击“Save”：

## 12.6. Cacti



### 12.6.6. 设置阈值

设置阈值前先按照前两节内容设置好电子邮件通知和阈值插件。

本例演示将阈值设置为“Disk free status 无磁盘状态”。

登录到Cacti管理网站，点击左侧菜单上的“Thresholds”，然后点击右侧窗格中的“Add”：

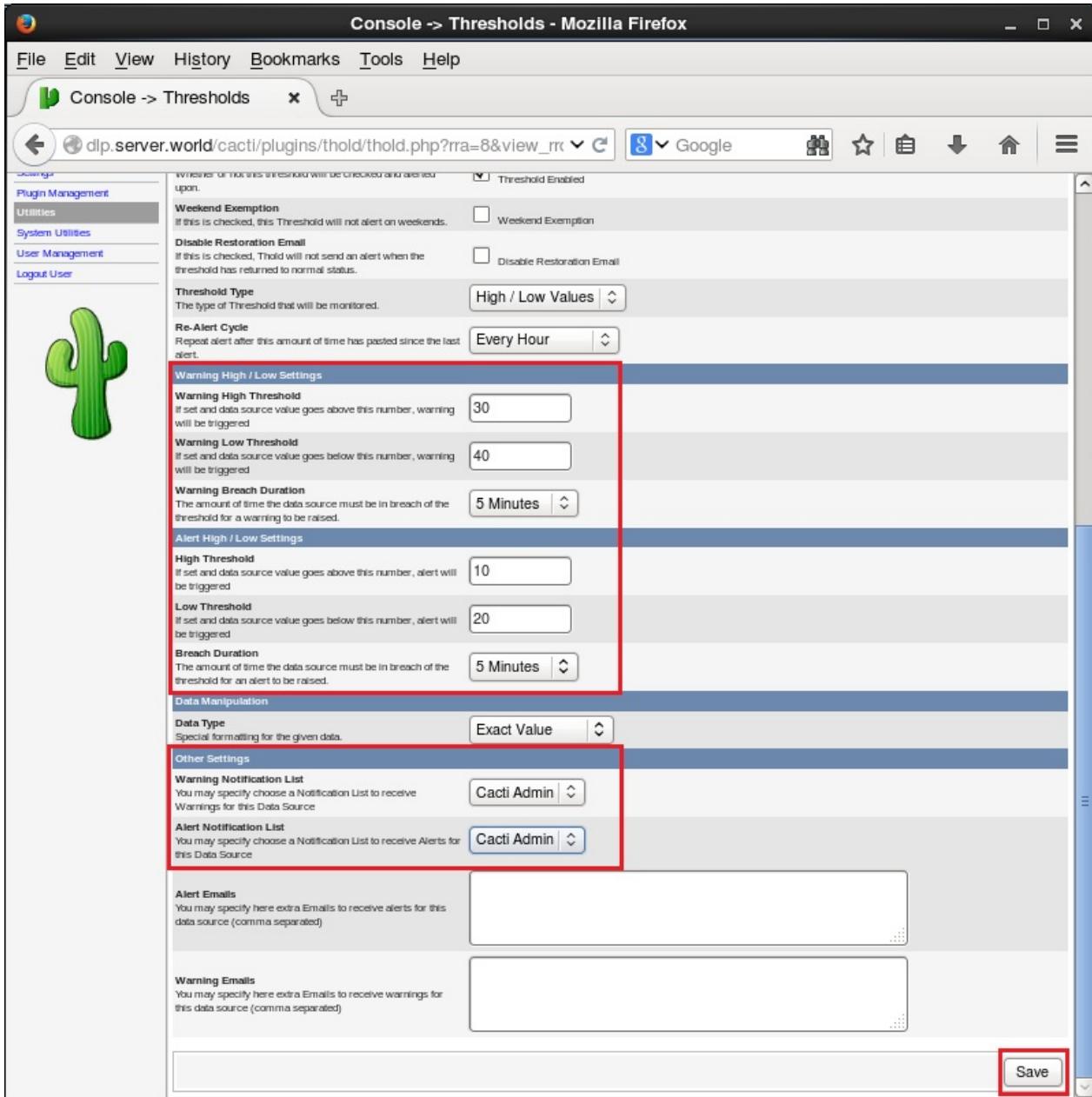
## 12.6. Cacti

The screenshot shows the Cacti web interface for threshold management. The left sidebar has 'Thresholds' selected. The main area is titled 'Threshold Management' with a search bar for Host, Template, and State. It displays two tables: 'No Rows Found' for thresholds and another 'No Rows Found' for alarms. A red box highlights the 'Add' button in the top right corner of the main panel.

选择“Host”，“Graph”，“Data Source”，最后点击“Create”：

The screenshot shows the 'Create Threshold' wizard. The left sidebar has 'Thresholds' selected. The main area is titled 'Threshold Creation Wizard' with fields for Host (localhost), Graph (localhost - Disk Space - /dev/mapper/Vol), and Data Source (hdd\_free). A red box highlights the 'Create' button. Below it is a graph titled 'localhost - Disk Space - /dev/mapper/Vol' showing bytes over time. The graph indicates current usage of 1.57 G, available space of 12.85 G, and total space of 14.42 G. A red box highlights the 'Used' data series in the legend.

输入Warning警告和Alert警戒情况下的阈值。此外，在“Notification List”字段中选择通知目的地。全部完成后，点击“Save”：



可以为项目设置阈值：

## 12.6. Cacti

The screenshot shows the Cacti 'Thresholds' configuration page. On the left, a sidebar lists various management options like 'Create', 'Management', 'Graph Management', and 'Utilities'. The 'Utilities' section is currently selected. In the main area, a message 'Record Updated' is displayed above a graph titled 'localhost - Disk Space - /dev/mapper/Vol'. The graph shows disk usage over time, with a red bar indicating used space and a blue bar indicating free space. Below the graph, two data source items are listed: '1: hdd\_free' and '2: hdd\_used'. A 'Data Source Item [hdd\_free] - Current value: [13473928.093]' section contains settings for propagation, name, and threshold enablement. A 'Threshold Name' field is set to 'localhost - Free Space - /dev/mapper/Vol [hdd\_free]'. The 'Threshold Enabled' checkbox is checked. At the bottom, there's a 'Weekend Exemption' section with a checkbox.

如果系统值超过阈值，则发送通知如下：

```
From cacti@dlp.srv.world Fri Mar 5 20:30:03 2015
Content-Type: text/plain; charset="UTF-8"

An Alert has been issued that requires your attention.

Host: Localhost (127.0.0.1)
URL: http://dlp.srv.world/cacti//graph.php?local_graph_id=5&rra_id=1
Message: ALERT: Localhost - Free Space - /dev/mapper/Vol [hdd_free] is still above threshold of 10 with 13472524
```

## 12.6.7. 添加监控目标主机

在网络上添加监控目标主机。

在要监控的目标主机上安装SNMP：

```
yum -y install net-snmp net-snmp-utils
```

编辑 /etc/snmp/snmpd.conf 文件：

```
# 注释
#com2sec notConfigUser      default          public

# 取消注释并更改
# “mynetwork”部分更改为自己的本地网络
# 除了“public”或“private”，更改团体名
com2sec local      localhost    Serverworld
com2sec mynetwork  10.0.0.0/24 Serverworld

# 取消注释并更改
group MyRWGroup v2c      local
group MyROGroup v2c      mynetwork

# 取消注释
view all      included .1          80

# 取消注释并更改
access MyROGroup "" v2c      noauth      exact      all      none      none
access MyRWGroup "" v2c      noauth      exact      all      all      all

# 取消注释
disk / 10000
```

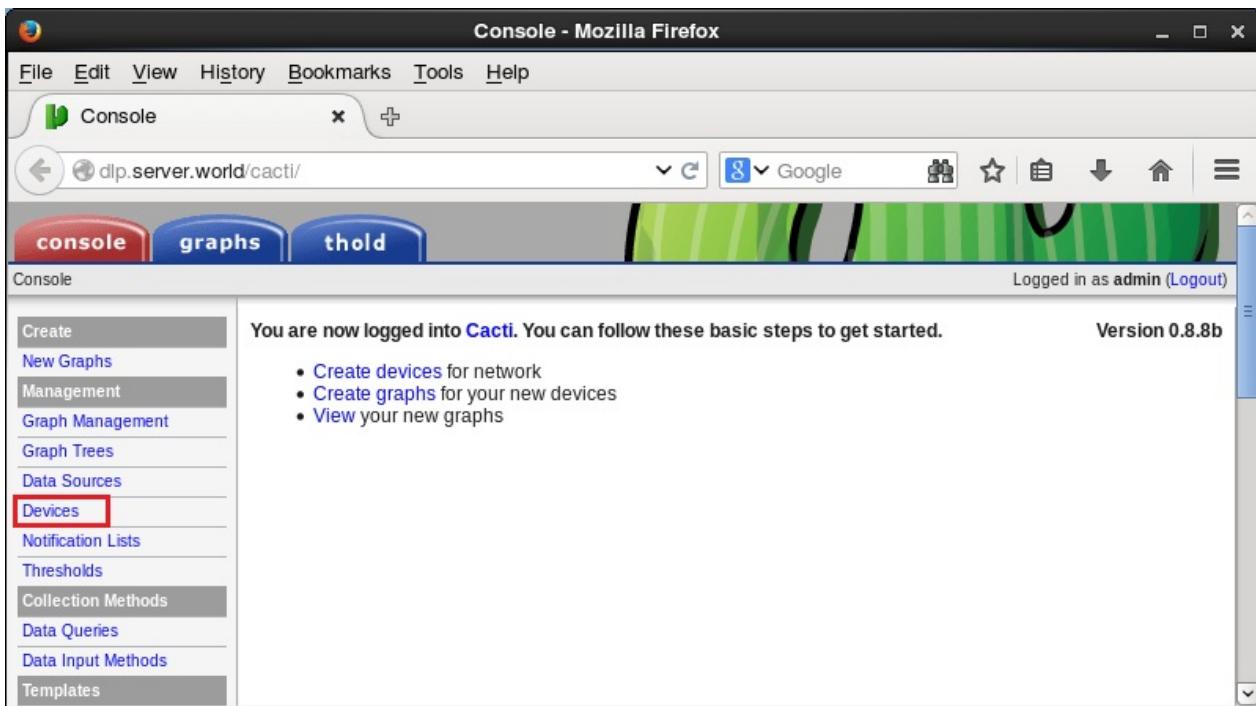
```
systemctl start snmpd
systemctl enable snmpd
```

验证（将“Serverworld”替换为自己的团体名）：

```
snmpwalk -v2c -c Serverworld localhost system
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Linux node01.srv.world 3.10.0-2
29.4.2.el7.x86_64 #1 SMP Wed May 13.....
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (3367) 0:00:33.67
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost> (config
ure /etc/snmp/snmp.local.conf)
...
...
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.10 = Timeticks: (4) 0:00:00.04
```

登录到Cacti管理网站，然后点击左侧菜单上的“Devices”：



点击右窗格中的“Add”：

## 12.6. Cacti

The screenshot shows the Cacti web interface in Mozilla Firefox. The title bar reads "Console -> Devices - Mozilla Firefox". The left sidebar has several tabs: "Console", "graphs", "threshold", "Devices" (which is selected), "Management", "Graph Management", "Graph Trees", "Data Sources", "Location Lists", "Thresholds", "Selection Methods", "Queries", "Input Methods", "Plates", "Host Templates", "Template", and "Threshold Template". The main content area is titled "Devices" and shows a table with one row. The table columns are: Description\*\*, ID, Graphs, Data Sources, Status, In State, Hostname, Current (ms), Average (ms), and Availability. The single row contains: Localhost, 1, 5, 6, Up, -, 127.0.0.1, 0.00, 0.74, and 100. There are "Previous" and "Next" links above and below the table. A "Delete" button is visible at the bottom right of the table area. The top right of the window shows "Logged in as admin (Logout)".

输入以下项目，然后点击“Create”：

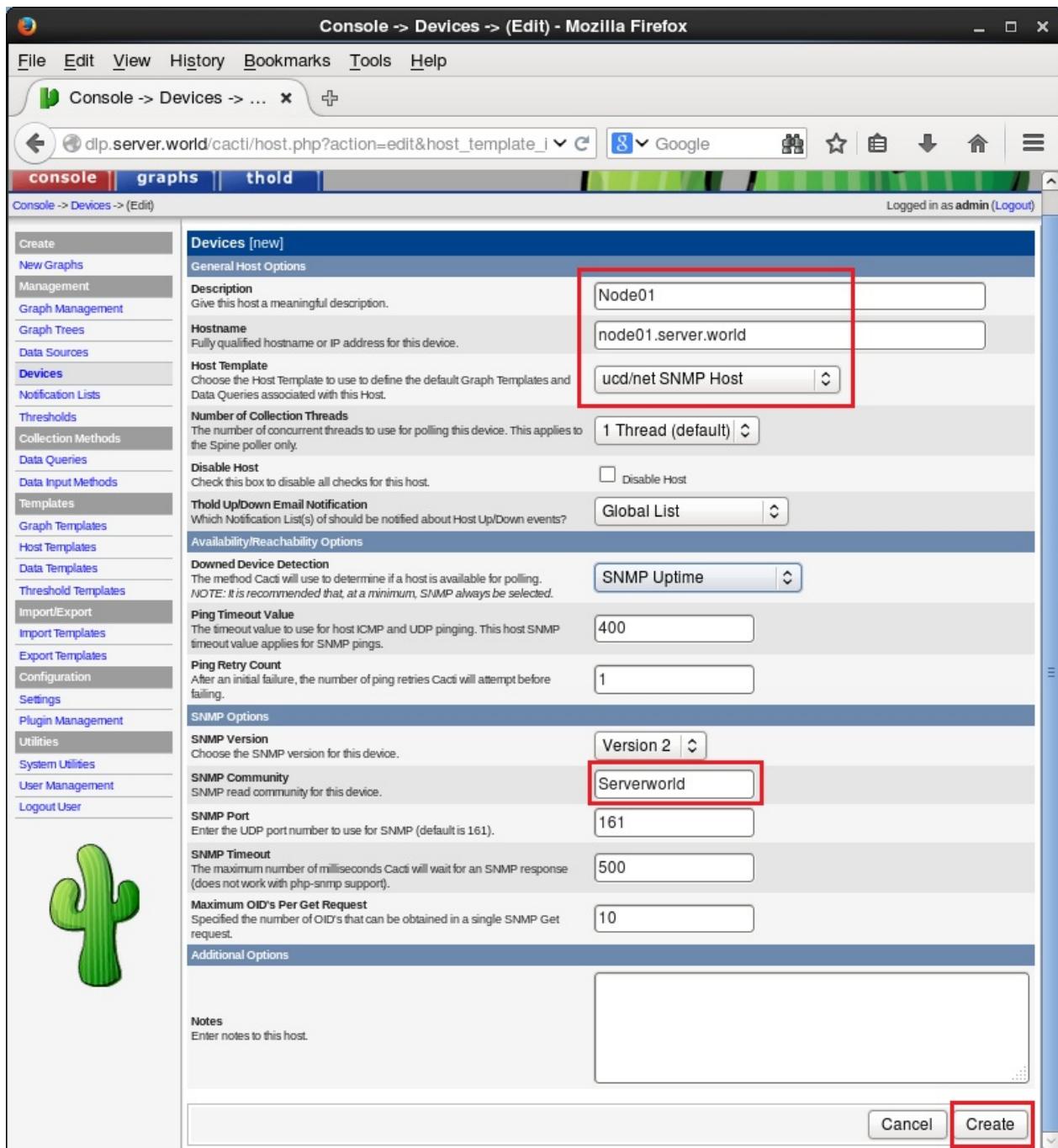
Description -> 简单描述

Hostname -> 目标的主机名或IP地址

Host Template -> ucd/net SNMP Host

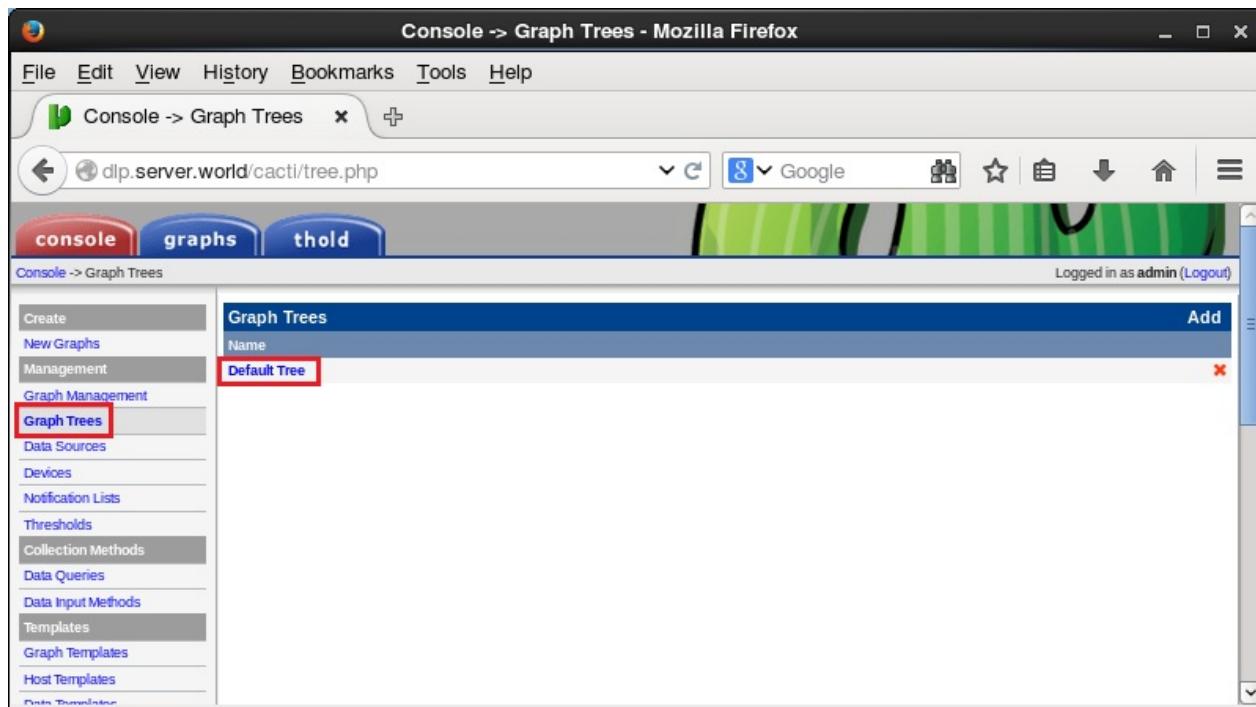
SNMP Community -> 在上面设置的团体名

## 12.6. Cacti

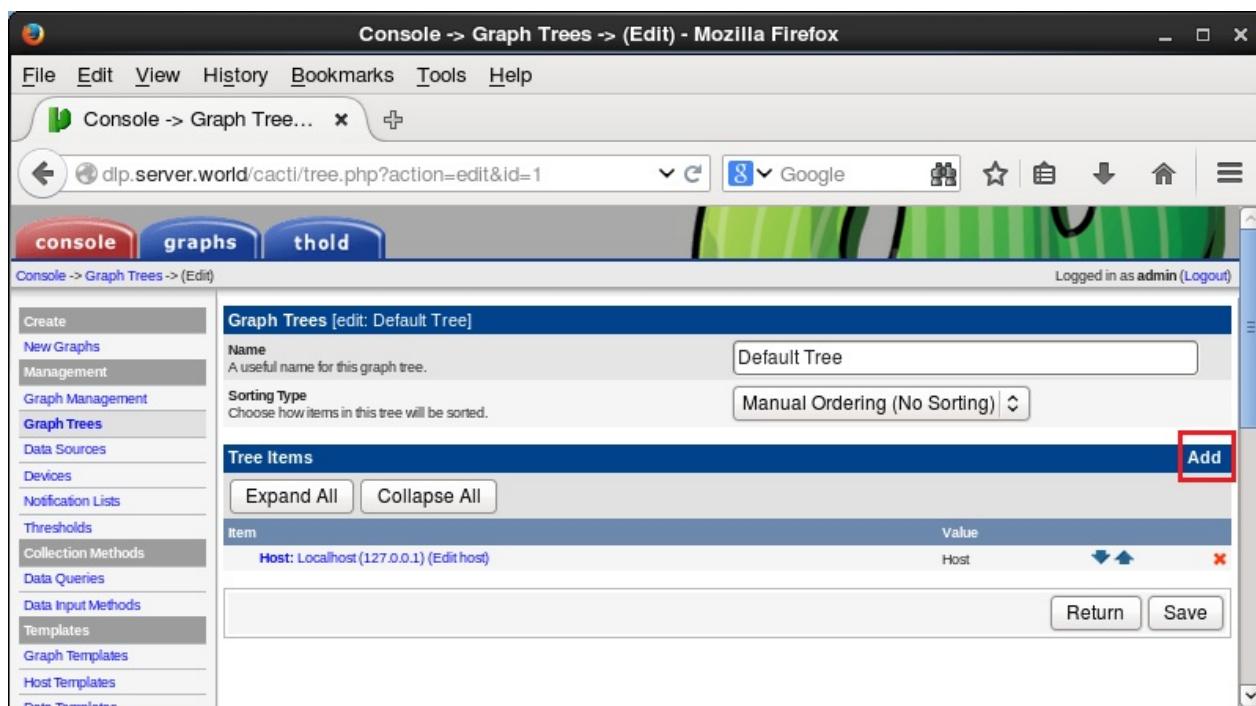


保存设置后，点击左侧菜单上的“Graph Trees”，然后点击右侧窗格中的“Default Tree”：

## 12.6. Cacti

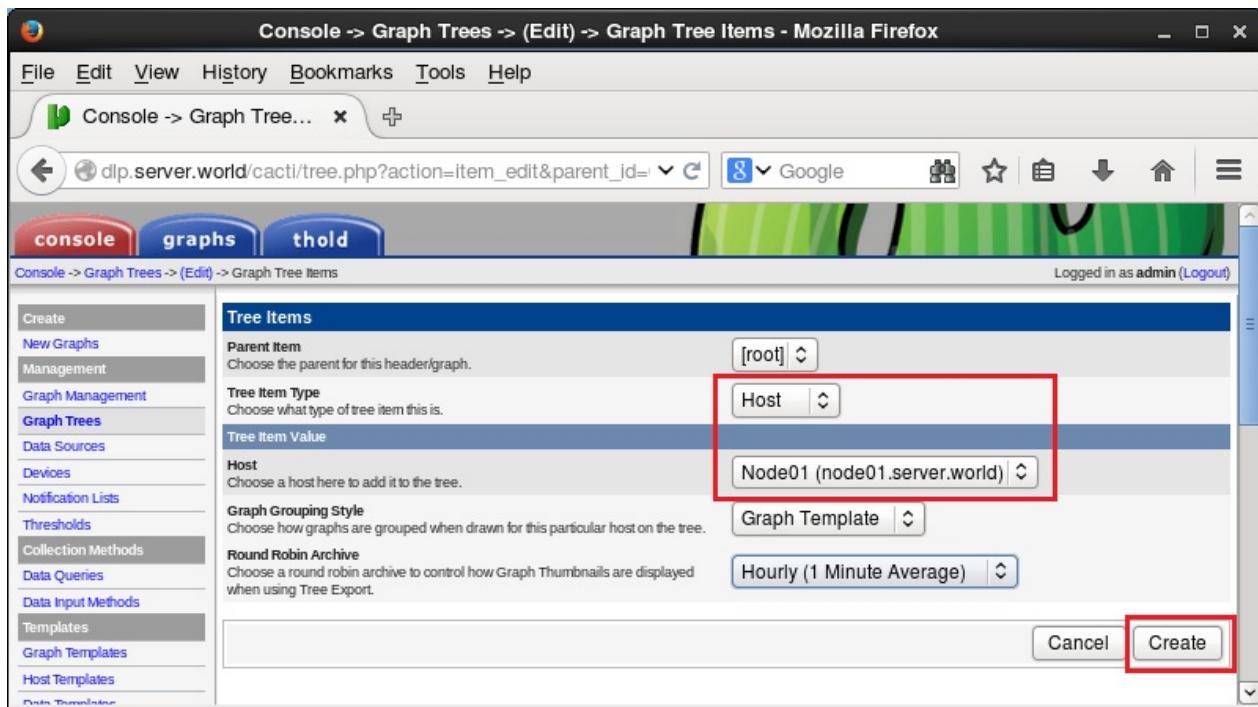


点击右侧窗格中的“Add”：

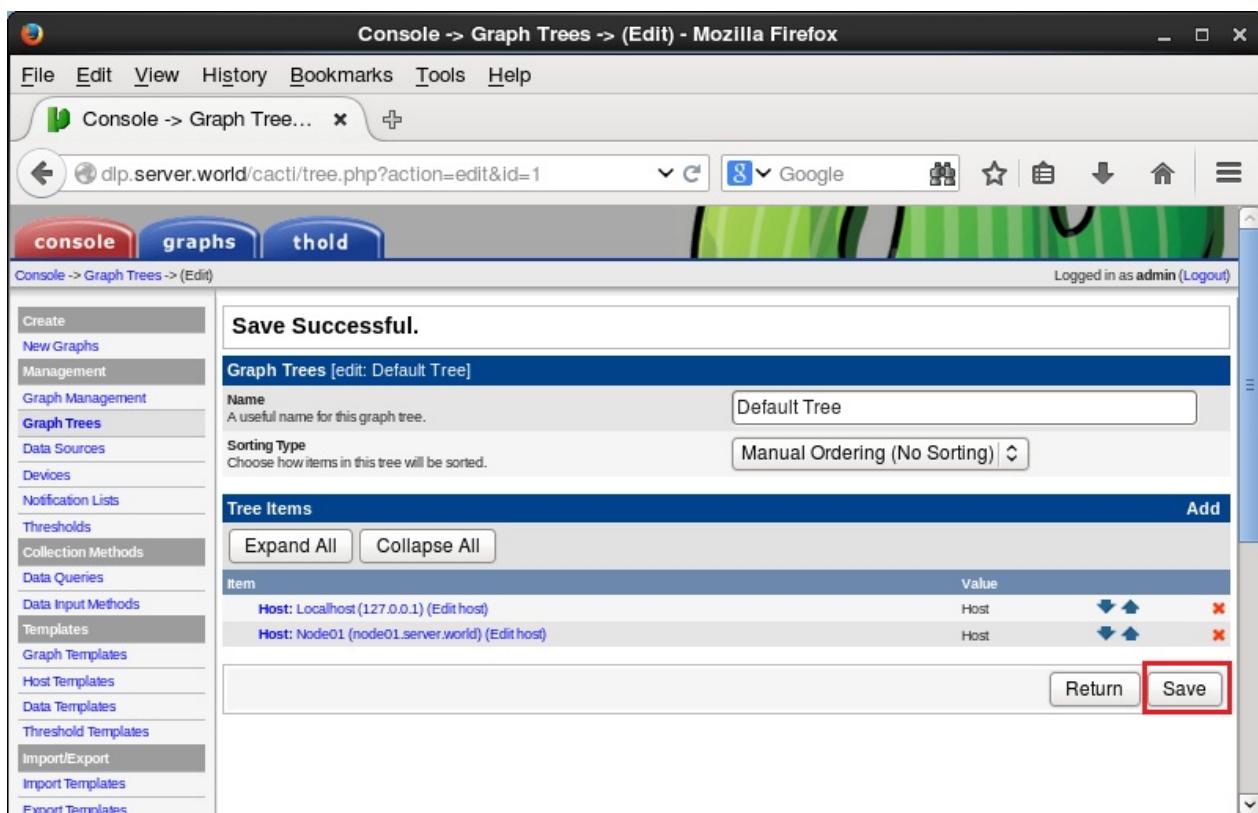


在“Tree Item Type”字段中选择“Host”，在“Host”字段中选择目标主机，然后点击“Create”：

## 12.6. Cacti



点击“Save”：



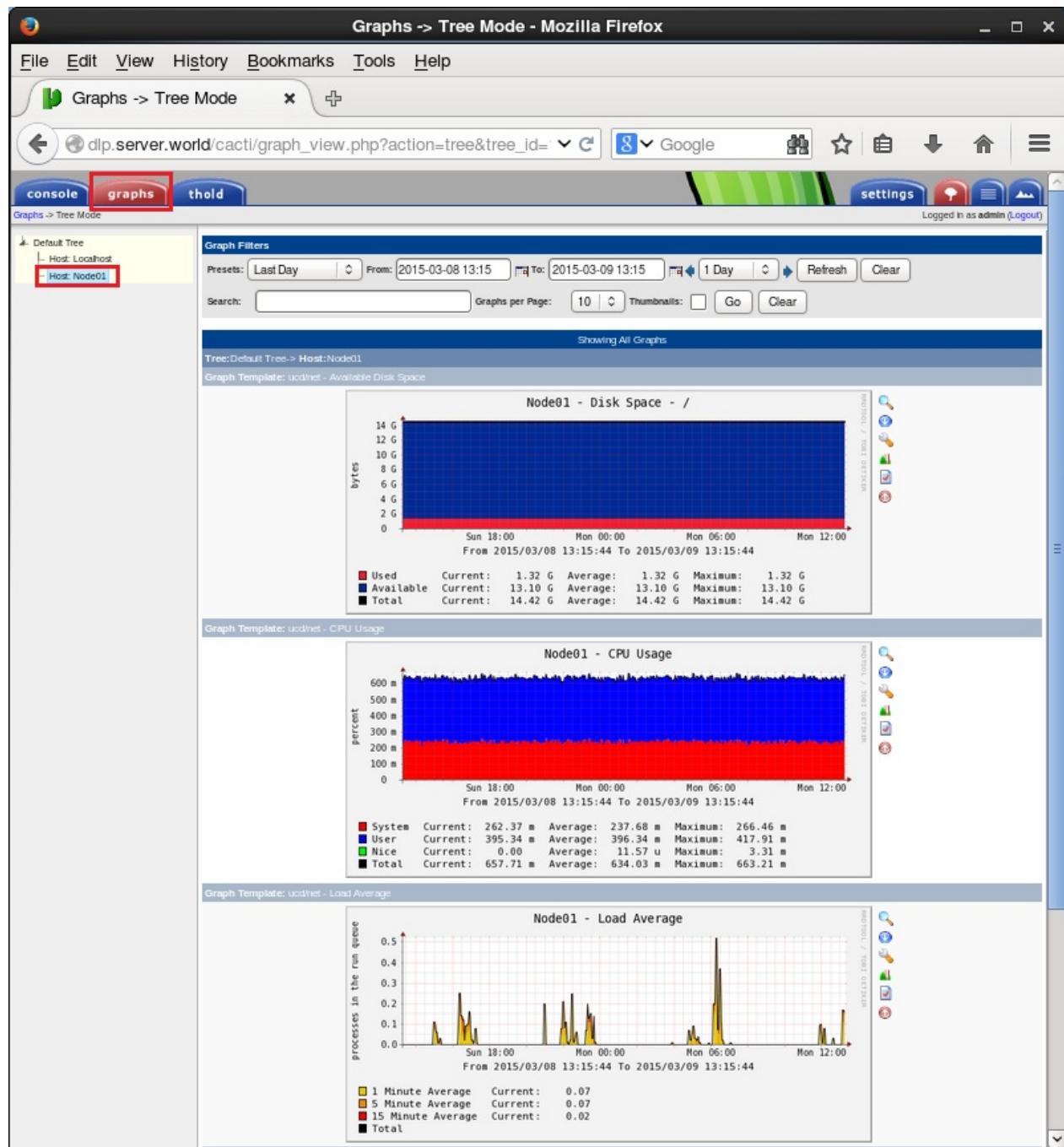
点击左侧菜单上的“New Graphs”，在右窗格中选择要添加的图表，然后点击“Create”：

## 12.6. Cacti

The screenshot shows the 'Create New Graphs' page in Cacti. The left sidebar has a 'New Graphs' section selected. The main area displays 'Node01 (node01.server.world) ucd/net SNMP Host'. It includes fields for 'Host' (Node01) and 'Graph Types' (All). A red box highlights the 'Graph Types' section, which contains three checked checkboxes: 'Edit this Host', 'Create New Host', and 'Auto-create thresholds'. Below this is a 'Graph Templates' section with three entries: 'ucdnet - CPU Usage', 'ucdnet - Load Average', and 'ucdnet - Memory Usage', all with checked checkboxes. A 'Create' dropdown menu is shown. The 'Data Query [SNMP - Interface Statistics]' section lists three network interfaces: Index 1 (Up, lo, lo, softwareLoopback/24, 10000000, 10, 127.0.0.1), Index 2 (Down, eth0, eth0, ethernetCsmacd/0, 0, 0, 00:10:36:8E:ED:8B), and Index 3 (Up, eth1, eth1, ethernetCsmacd/0, 0, 0, 00:10:36:15:52:FB, 10.0.0.11). The 'Data Query [ucd/net - Get Monitored Partitions]' section lists one partition: Index 1 (Mount Point /, Device Name /dev/mapper/VolGroup-lv\_root). At the bottom right are 'Cancel' and 'Create' buttons.

几分钟后，转到“graphs”标签以查看系统状态，在左侧菜单中选择新的目标主机，如下所示：

## 12.6. Cacti



## 12.7. Nagios

Nagios是一个监控系统运行状态和网络信息的监控系统，能监控所指定的本地或远程主机以及服务，同时提供异常通知功能等。

### 12.7.1. 安装Nagios

先安装Apache httpd和PHP。

```
yum --enablerepo=epel -y install nagios nagios-plugins-{ping,disk,users,procs,load,swap,ssh,http} # 从EPEL安装Nagios和基本插件以监视服务器本身
```

配置Nagios：

编辑 /etc/httpd/conf.d/nagios.conf 文件：

```
# 更改设置以设置访问权限  
#Require all granted  
#Require local  
Require ip 127.0.0.1 10.0.0.0/24
```

添加Nagios管理用户：

```
htpasswd /etc/nagios/passwd nagiosadmin
```

```
New password: # 设置密码  
Re-type new password: # 确认密码  
Adding password for user nagiosadmin
```

```
systemctl start nagios  
systemctl enable nagios  
systemctl restart httpd
```

firewalld防火墙规则：

```
firewall-cmd --add-service={http,https} --permanent  
firewall-cmd --reload
```

从Nagios服务器允许的网络中的客户端访问 `http://(Nagios服务器的主机名或IP地址)/nagios/`，并使用Nagios管理员用户“`nagiosadmin`”进行身份验证以登录：



成功验证后，将显示Nagios管理网站：

## 12.7. Nagios

The screenshot shows the Nagios Core web interface running in Mozilla Firefox. The title bar reads "Nagios Core - Mozilla Firefox". The address bar shows the URL "dlp.srv.world/nagios/". The main content area features the Nagios logo and a message indicating that the daemon is running with PID 3347. It displays the version information "Nagios® Core™ Version 4.0.8" and the date "August 12, 2014", with a link to "Check for updates". A blue box at the top right announces a new version of Nagios Core available, with a link to download Nagios 4.2.1. Below this, there are four promotional banners for related products: "Nagios XI" (Easy Configuration Advanced Reporting), "Nagios Log Server" (Monitor and analyze logs from anywhere), "Nagios Network Analyzer" (Real-time netflow and bandwidth analysis), and "Nagios Comments" (Downtime). On the left side, a sidebar menu includes sections for General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems), Reports (Availability, Trends, Alerts, History, Summary, Histogram), Notifications (Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration). A "Get Started" box lists several steps to begin monitoring, and a "Quick Links" box provides links to various Nagios resources.

可以点击“Tactical Overview”查看系统状态等：

## 12.7. Nagios

Nagios Core - Mozilla Firefox

N Nagios Core

dlp.srv.world/nagios/

Search

**Nagios®**

**Tactical Monitoring Overview**

Last Updated: Fri Sep 16 14:32:01 JST 2016  
Updated every 90 seconds  
Nagios® Core™ 4.0.8 - [www.nagios.org](http://www.nagios.org)  
Logged in as nagiosadmin

**General**

Home Documentation

**Current Status**

**Tactical Overview**

Map Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search:

**Reports**

Availability Trends Alerts History Summary Histogram Notifications Event Log

**System**

Comments Downtime Process Info Performance Info Scheduling Queue Configuration

**Monitoring Performance**

|                                  |                         |
|----------------------------------|-------------------------|
| Service Check Execution Time:    | 0.00 / 4.01 / 0.509 sec |
| Service Check Latency:           | 0.00 / 0.00 / 0.000 sec |
| Host Check Execution Time:       | 4.00 / 4.00 / 4.004 sec |
| Host Check Latency:              | 0.00 / 0.00 / 0.000 sec |
| # Active Host / Service Checks:  | 1 / 8                   |
| # Passive Host / Service Checks: | 0 / 0                   |

**Network Outages**

0 Outages

**Network Health**

Host Health:  Service Health: 

**Hosts**

0 Down 0 Unreachable 1 Up 0 Pending

**Services**

0 Critical 1 Warning 0 Unknown 7 Ok 0 Pending

1 Unhandled Problems

**Monitoring Features**

| Flap Detection                                                                                           | Notifications                                                                                           | Event Handlers                                                                                           | Active Checks                                                                                              | Passive Checks                                                                                             |
|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
|  All Services Enabled |  2 Services Disabled |  All Services Enabled |  All Services Enabled |  All Services Enabled |
| No Services Flapping                                                                                     | All Hosts Enabled                                                                                       | All Hosts Enabled                                                                                        | All Hosts Enabled                                                                                          | All Hosts Enabled                                                                                          |
| All Hosts Enabled                                                                                        |                                                                                                         |                                                                                                          |                                                                                                            |                                                                                                            |
| No Hosts Flapping                                                                                        |                                                                                                         |                                                                                                          |                                                                                                            |                                                                                                            |

## 12.7. Nagios

**Current Network Status**  
Last Updated: Fri Sep 16 15:14:18 JST 2016  
Updated every 90 seconds  
Nagios® Core™ 4.0.8 - [www.nagios.org](http://www.nagios.org)  
Logged in as nagiosadmin

**Host Status Totals**  
Up: 1, Down: 0, Unreachable: 0, Pending: 0  
All Problems: 0, All Types: 1

**Service Status Totals**  
Ok: 7, Warning: 1, Unknown: 0, Critical: 0, Pending: 0  
All Problems: 1, All Types: 8

**Service Status Details For All Hosts**

| Host      | Service         | Status  | Last Check          | Duration      | Attempt | Status Information                                                              |
|-----------|-----------------|---------|---------------------|---------------|---------|---------------------------------------------------------------------------------|
| localhost | Current Load    | OK      | 09-16-2016 15:14:09 | 0d 0h 55m 9s  | 1/4     | OK - load average: 0.02, 0.04, 0.05                                             |
|           | Current Users   | OK      | 09-16-2016 15:09:47 | 0d 0h 54m 31s | 1/4     | USERS OK - 1 users currently logged in                                          |
|           | HTTP            | WARNING | 09-16-2016 15:13:24 | 0d 0h 53m 54s | 4/4     | HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.001 second response time |
|           | PING            | OK      | 09-16-2016 15:11:02 | 0d 0h 53m 16s | 1/4     | PING OK - Packet loss = 0%, RTA = 0.08 ms                                       |
|           | Root Partition  | OK      | 09-16-2016 15:11:39 | 0d 0h 52m 39s | 1/4     | DISK OK - free space: / 25308 MB (93% inode=99%):                               |
|           | SSH             | OK      | 09-16-2016 15:12:17 | 0d 0h 52m 1s  | 1/4     | SSH OK - OpenSSH_6.6.1 (protocol 2.0)                                           |
|           | Swap Usage      | OK      | 09-16-2016 15:12:54 | 0d 0h 51m 24s | 1/4     | SWAP OK - 100% free (3071 MB out of 3071 MB)                                    |
|           | Total Processes | OK      | 09-16-2016 15:13:32 | 0d 0h 50m 46s | 1/4     | PROCS OK: 143 processes with STATE = RSZDT                                      |

Results 1 - 8 of 8 Matching Services

### 12.7.2. 电子邮件通知设置

某些项目默认启用，如果需要更改，配置如下：

```
yum -y install mailx
```

编辑 `/etc/nagios/objects/contacts.cfg` 文件：

```
# 设置收件人电子邮件地址
email root@localhost
```

```
systemctl restart nagios
```

## 12.7. Nagios

可以更改Nagios管理网站上的通知设置。

登录并点击“Services”，显示服务列表。在服务名称旁显示的图标就是禁用通知的图标。（HTTP和SSH如下所示）要在服务上启用通知，先点击服务名称：

The screenshot shows the Nagios Core web interface in Mozilla Firefox. The left sidebar has a 'Services' link highlighted with a red box. The main content area displays a table of service status details for the host 'localhost'. The table includes columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. Two services, 'HTTP' and 'SSH', are shown with a red box around their names, indicating they have notifications disabled. The 'HTTP' service is in a yellow 'WARNING' state, while 'SSH' is in an 'OK' state.

| Host      | Service         | Status  | Last Check          | Duration      | Attempt | Status Information                                                              |
|-----------|-----------------|---------|---------------------|---------------|---------|---------------------------------------------------------------------------------|
| localhost | Current Load    | OK      | 09-16-2016 16:04:09 | 0d 1h 46m 19s | 1/4     | OK - load average: 0.04, 0.04, 0.05                                             |
| localhost | Current Users   | OK      | 09-16-2016 16:04:47 | 0d 1h 45m 41s | 1/4     | USERS OK - 1 users currently logged in                                          |
| localhost | HTTP            | WARNING | 09-16-2016 16:03:24 | 0d 1h 45m 4s  | 4/4     | HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.002 second response time |
| localhost | PING            | OK      | 09-16-2016 16:01:02 | 0d 1h 44m 26s | 1/4     | PING OK - Packet loss = 0%, RTA = 0.06 ms                                       |
| localhost | Root Partition  | OK      | 09-16-2016 16:01:39 | 0d 1h 43m 49s | 1/4     | DISK OK - free space: / 25308 MB (93% inode=99%):                               |
| localhost | SSH             | OK      | 09-16-2016 16:02:17 | 0d 1h 43m 11s | 1/4     | SSH OK - OpenSSH_6.6.1 (protocol 2.0)                                           |
| localhost | Swap Usage      | OK      | 09-16-2016 16:02:54 | 0d 1h 42m 34s | 1/4     | SWAP OK - 100% free (3071 MB out of 3071 MB)                                    |
| localhost | Total Processes | OK      | 09-16-2016 16:03:32 | 0d 1h 41m 56s | 1/4     | PROCS OK: 144 processes with STATE = R/SZDT                                     |

点击“Enable notifications for this service”：

## 12.7. Nagios

The screenshot shows the Nagios Core interface in Mozilla Firefox. The URL is <http://dlp.srv.world/nagios/>. The left sidebar has sections for General, Current Status, and Reports. The Current Status section is expanded, showing links like Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, and Reports. The Reports section shows Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log, and a Quick Search bar.

**Service Information**  
Last Updated: Fri Sep 16 16:07:13 JST 2016  
Updated every 90 seconds  
Nagios® Core™ 4.0.8 - [www.nagios.org](http://www.nagios.org)  
Logged in as **nagiosadmin**

**Service**  
**SSH**  
On Host  
**localhost**  
([localhost](#))

Member of  
**No servicegroups.**

127.0.0.1

**Service State Information**

|                                  |                                        |
|----------------------------------|----------------------------------------|
| <b>Current Status:</b>           | <b>OK</b> (for 0d 1h 44m 56s)          |
| <b>Status Information:</b>       | SSH OK - OpenSSH_6.6.1 (protocol 2.0)  |
| <b>Performance Data:</b>         | time=0.019980s;;0.000000;10000000      |
| <b>Current Attempt:</b>          | 1/4 (HARD state)                       |
| <b>Last Check Time:</b>          | 09-16-2016 16:02:17                    |
| <b>Check Type:</b>               | ACTIVE                                 |
| <b>Check Latency / Duration:</b> | 0.000 / 0.023 seconds                  |
| <b>Next Scheduled Check:</b>     | 09-16-2016 16:07:17                    |
| <b>Last State Change:</b>        | 09-16-2016 14:22:17                    |
| <b>Last Notification:</b>        | N/A (notification 0)                   |
| <b>Is This Service Flapping?</b> | <b>NO</b> (0.00% state change)         |
| <b>In Scheduled Downtime?</b>    | <b>NO</b>                              |
| <b>Last Update:</b>              | 09-16-2016 16:07:08 ( 0d 0h 0m 5s ago) |

**Service Commands**

- Disable active checks of this service
- Re-schedule the next check of this service
- Submit passive check result for this service
- Stop accepting passive checks for this service
- Stop obsessing over this service
- Enable notifications for this service **(highlighted)**
- Send custom service notification
- Schedule downtime for this service
- Disable event handler for this service
- Disable flap detection for this service

点击“Commit”：

The screenshot shows the Nagios Core interface in Mozilla Firefox. The URL is <http://dlp.srv.world/nagios/>. The left sidebar is identical to the previous screenshot.

**External Command Interface**  
Last Updated: Fri Sep 16 16:07:42 JST 2016  
Nagios® Core™ 4.0.8 - [www.nagios.org](http://www.nagios.org)  
Logged in as **nagiosadmin**

**You are requesting to enable notifications for a service**

**Command Options**

|                   |                  |
|-------------------|------------------|
| <b>Host Name:</b> | <b>localhost</b> |
| <b>Service:</b>   | <b>SSH</b>       |

**Commit** **Reset**

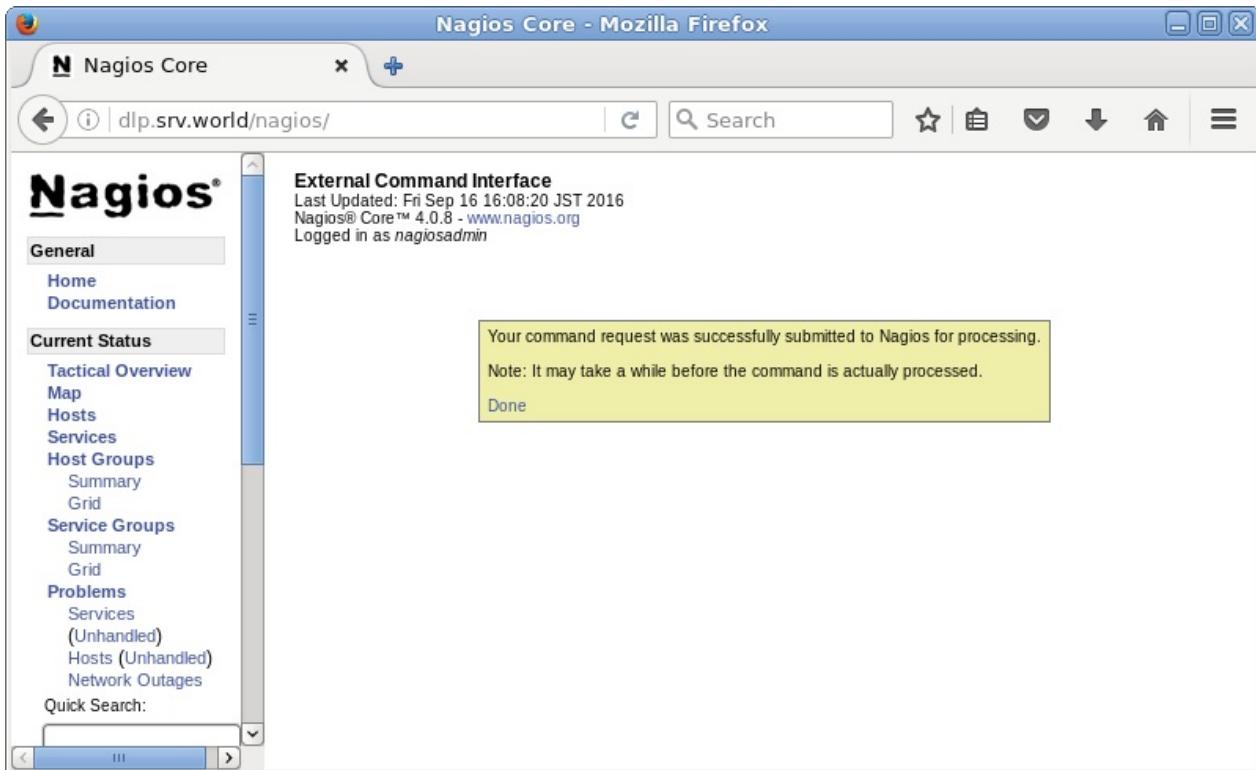
**Command Description**

This command is used to enable notifications for the specified service. Notifications will only be sent out for the service state types you defined in your service definition.

Please enter all required information before committing the command.  
Required fields are marked in red.  
Failure to supply all required values will result in an error.

## 12.7. Nagios

可以设置通知：



如果启用通知并且服务存在一些问题，则会将如下通知发送给设置的收件人：

```
Subject: ** PROBLEM Service Alert: localhost/SSH is CRITICAL **
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
From: nagios@dlp.srv.world
```

\*\*\*\*\* Nagios \*\*\*\*\*

Notification Type: PROBLEM

Service: SSH  
Host: localhost  
Address: 127.0.0.1  
State: CRITICAL

Date/Time: Wed Feb 25 03:29:48 JST 2015

Additional Info:

Connection refused

### 12.7.3. 设置阈值

某些项目默认设置，如果需要更改，配置如下：

默认情况下，在配置文件中设置阈值。例如，监控根分区的磁盘使用情况的项目设置如下：

编辑 `/etc/nagios/objects/localhost.cfg` 文件：

```
# Define a service to check the disk space of the root partition
# on the local machine. Warning if > 20% free, critical if
# > 10% free space on partition.

# 阈值设置为：>20%可用磁盘为警告，>10%可用磁盘为严重
# 可以自己更改这些值
define service{
    use                      local-service
    host_name                localhost
    service_description       Root Partition
    check_command             check_local_disk!20%!10%!
}
```

`systemctl restart nagios`

若需要添加新插件并为其配置，如下设置（以添加`check_ntp_time`插件为例）：

`/usr/lib64/nagios/plugins/check_ntp_time -h` # 显示插件的选项

```
...
...
-w, --warning=THRESHOLD
    Offset to result in warning status (seconds)
-c, --critical=THRESHOLD
...
...
```

编辑 `/etc/nagios/objects/commands.cfg` 文件，添加带有阈值选项的插件的命令的定义：

```
# 添加以下内容到最后
define command{
    command_name      check_ntp_time
    command_line      $USER1$/check_ntp_time -H $ARG1$ -w $ARG
2$ -c $ARG3$
}
```

编辑 `/etc/nagios/objects/localhost.cfg` 文件，添加带有阈值的服务的定义：

```
# 添加以下内容到最后（如果有1秒的时间差为警告，如果2秒为严重）
define service{
    use                      local-service
    host_name                localhost
    service_description       NTP_TIME
    check_command             check_ntp_time!ntp1.jst.
    mfeed.ad.jp!1!2
    notifications_enabled     1
}
```

```
systemctl restart nagios
```

如果系统值超过阈值，则发送通知：

```
Subject: ** PROBLEM Service Alert: localhost/Root Partition is CRITICAL **
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
From: nagios@dlp.srv.world
Status: R

***** Nagios *****

Notification Type: PROBLEM

Service: Root Partition
Host: localhost
Address: 127.0.0.1
State: CRITICAL

Date/Time: Wed Feb 25 07:49:12 JST 2015

Additional Info:

DISK CRITICAL - free space: / 13232 MB (9% inode=15%):
```

### 12.7.4. 添加监控目标项

有很多RPM软件包可以添加监控目标项目，或者也可以自己创建插件。还有许多由[社区提供的插件](#)。

许多插件以RPM软件包方式提供，如下：

```
yum --enablerepo=epel search nagios-plugins- # 使用EPEL搜索
```

```
nagios-plugins-all.x86_64 : Nagios Plugins - All plugins
nagios-plugins-apt.x86_64 : Nagios Plugin - check_apt
nagios-plugins-bdii.x86_64 : Nagios Plugin - check_bdii_entries
nagios-plugins-bonding.x86_64 : Nagios plugin to monitor Linux bonding interfaces
nagios-plugins-breeze.x86_64 : Nagios Plugin - check_breeze
...
...
nagios-plugins-ups.x86_64 : Nagios Plugin - check_ups
nagios-plugins-users.x86_64 : Nagios Plugin - check_users
nagios-plugins-wave.x86_64 : Nagios Plugin - check_wave
```

例如，添加check\_ntp插件以监控系统和NTP服务器之间的时间差：

```
yum --enablerepo=epel -y install nagios-plugins-ntp
```

编辑 /etc/nagios/objects/commands.cfg 文件：

```
# 添加以下内容到最后
define command{
    command_name      check_ntp_time
    command_line      $USER1$/check_ntp_time -H $ARG1$ -w $ARG
2$ -c $ARG3$
}
```

编辑 /etc/nagios/objects/localhost.cfg 文件：

```
# 添加以下内容到最后（如果有1秒的时间差为警告，如果2秒为严重）
define service{
    use                      local-service
    host_name                localhost
    service_description       NTP_TIME
    check_command             check_ntp_time!ntp1.jst.
    mfeed.ad.jp!1!2
    notifications_enabled     1
}
```

```
systemctl restart nagios
```

可以在管理网站上查看新插件的状态：

**Current Network Status**

Last Updated: Fri Sep 16 16:21:34 JST 2016  
Updated every 90 seconds  
Nagios® Core™ 4.0.8 - [www.nagios.org](http://www.nagios.org)  
Logged in as nagiosadmin

**Host Status Totals**

|    |      |             |         |
|----|------|-------------|---------|
| Up | Down | Unreachable | Pending |
| 1  | 0    | 0           | 0       |

All Problems All Types

|   |   |
|---|---|
| 0 | 1 |
|---|---|

**Service Status Totals**

|    |         |         |          |         |
|----|---------|---------|----------|---------|
| Ok | Warning | Unknown | Critical | Pending |
| 8  | 1       | 0       | 0        | 0       |

All Problems All Types

|   |   |
|---|---|
| 1 | 9 |
|---|---|

**Service Status Details For All Hosts**

Limit Results: 100

| Host      | Service         | Status  | Last Check          | Duration      | Attempt | Status Information                                                              |
|-----------|-----------------|---------|---------------------|---------------|---------|---------------------------------------------------------------------------------|
| localhost | Current Load    | OK      | 09-16-2016 16:19:09 | 0d 2h 2m 25s  | 1/4     | OK - load average: 0.01, 0.04, 0.05                                             |
|           | Current Users   | OK      | 09-16-2016 16:19:46 | 0d 2h 1m 47s  | 1/4     | USERS OK - 1 users currently logged in                                          |
|           | HTTP            | WARNING | 09-16-2016 16:18:22 | 0d 2h 1m 10s  | 4/4     | HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.001 second response time |
|           | NTP_TIME        | OK      | 09-16-2016 16:20:17 | 0d 0h 1m 17s  | 1/4     | NTP OK: Offset 0.0003632307053 secs                                             |
|           | PING            | OK      | 09-16-2016 16:21:00 | 0d 2h 0m 32s  | 1/4     | PING OK - Packet loss = 0%, RTA = 0.06 ms                                       |
|           | Root Partition  | OK      | 09-16-2016 16:16:38 | 0d 1h 59m 55s | 1/4     | DISK OK - free space: / 25308 MB (93% inode=99%);                               |
|           | SSH             | OK      | 09-16-2016 16:20:16 | 0d 0h 1m 18s  | 1/4     | SSH OK - OpenSSH_6.6.1 (protocol 2.0)                                           |
|           | Swap Usage      | OK      | 09-16-2016 16:17:53 | 0d 1h 58m 40s | 1/4     | SWAP OK - 100% free (3071 MB out of 3071 MB)                                    |
|           | Total Processes | OK      | 09-16-2016 16:18:30 | 0d 1h 58m 2s  | 1/4     | PROCS OK: 146 processes with STATE = RSZDT                                      |

### 12.7.5. 添加监控目标主机

可以监控网络上的其他服务器。

例如，使用简单的 ping 命令添加用于监控目标服务器：

编辑 /etc/nagios/nagios.cfg 文件：

```
# 取消注释
cfg_dir=/etc/nagios/servers
```

```
mkdir /etc/nagios/servers
```

```
chgrp nagios /etc/nagios/servers
```

```
chmod 750 /etc/nagios/servers
```

## 12.7. Nagios

编辑 /etc/nagios/servers/node01.cfg 文件：

```
define host{
    use          linux-server
    host_name    node01
    alias        node01
    address      10.0.0.51
}

define service{
    use          generic-service
    host_name    node01
    service_description PING
    check_command check_ping!100.0,20%!500.0,60%
}
```

```
systemctl restart nagios
```

可以在管理网站上查看新服务器的状态：

The screenshot shows the Nagios Core web interface running in Mozilla Firefox. The main page displays the 'Current Network Status' with last update information and a summary of host and service status totals. Below this, the 'Host Status Details For All Host Groups' section shows two hosts: 'localhost' and 'node01', both marked as 'UP'. The interface includes a sidebar with navigation links for General, Current Status, Host Groups, Service Groups, Problems, and Quick Search.

还可以监控其他服务器上的服务。

在要监控服务的目标主机上安装nrpe：

```
yum --enablerepo=epel -y install nrpe nagios-plugins-{ping,disk,users,procs,load,swap,ssh}
```

## 12.7. Nagios

编辑 `/etc/nagios/nrpe.cfg` 文件：

```
# 添加访问权限（指定Nagios服务器）
allowed_hosts=127.0.0.1,10.0.0.30

# 允许命令的参数
dont_blame_nrpe=1

# 注释下面内容
#command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5
# -c 10
#command[check_load]=/usr/lib64/nagios/plugins/check_load -w 15,
# 10,5 -c 30,25,20
#command[check_hda1]=/usr/lib64/nagios/plugins/check_disk -w 20%
# -c 10% -p /dev/hda1
#command[check_zombie_procs]=/usr/lib64/nagios/plugins/check_procs -w 5 -c 10 -sZ
#command[check_total_procs]=/usr/lib64/nagios/plugins/check_procs -w 150 -c 200

# 取消注释下面内容
command[check_users]=/usr/lib64/nagios/plugins/check_users -w $ARG1$ -c $ARG2$
command[check_load]=/usr/lib64/nagios/plugins/check_load -w $ARG1$ -c $ARG2$
command[check_disk]=/usr/lib64/nagios/plugins/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
command[check_procs]=/usr/lib64/nagios/plugins/check_procs -w $ARG1$ -c $ARG2$ -s $ARG3$
```

```
systemctl start nrpe
systemctl enable nrpe
```

firewalld防火墙规则：

```
firewall-cmd --add-port=5666/tcp --permanent
firewall-cmd --reload
```

配置Nagios服务器：

## 12.7. Nagios

```
yum --enablerepo=epel -y install nagios-plugins-nrpe
```

编辑 /etc/nagios/nagios.cfg 文件：

```
# 取消注释  
cfg_dir=/etc/nagios/servers
```

```
mkdir /etc/nagios/servers
```

```
chgrp nagios /etc/nagios/servers
```

```
chmod 750 /etc/nagios/servers
```

编辑 /etc/nagios/objects/commands.cfg 文件：

```
# 添加以下内容到最后  
define command{  
    command_name      check_nrpe  
    command_line      $USER1$/check_nrpe -H $HOSTADDRESS$ -c $  
    ARG1$  
}
```

编辑 /etc/nagios/servers/node01.cfg 文件：

```
# 定义目标主机  
define host{  
    use          linux-server  
    host_name    node01  
    alias        node01  
    address     10.0.0.51  
}  
define service{  
    use          generic-service  
    host_name    node01  
    service_description PING  
    check_command check_ping!100.0,20%!500.0,60%  
}  
# 可用磁盘  
define service{  
    use          generic-service  
    host_name    node01
```

## 12.7. Nagios

```
service_description Root Partition
check_command      check_nrpe!check_disk\!20%\!10%\!/
}
# 当前用户
define service{
    use          generic-service
    host_name   node01
    service_description Current Users
    check_command  check_nrpe!check_users\!20\!50
}
# 总进程
define service{
    use          generic-service
    host_name   node01
    service_description Total Processes
    check_command  check_nrpe!check_procs\!250\!400\!RSZ
DT
}
# 当前负载
define service{
    use          generic-service
    host_name   node01
    service_description Current Load
    check_command  check_nrpe!check_load\!5.0,4.0,3.0\!1
0.0,6.0,4.0
}
```

```
systemctl restart nagios
```

可以在管理网站上查看新服务器的状态：

## 12.7. Nagios

The screenshot shows the Nagios Core interface in Mozilla Firefox. The left sidebar has sections for General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration). The main content area is titled "Service Status Details For All Hosts". It shows a table with columns: Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The table lists services for two hosts: localhost and node01. Most services are in an OK state, except for one WARNING state for the HTTP service on localhost.

| Host            | Service         | Status              | Last Check          | Duration      | Attempt                                    | Status Information                                                              |
|-----------------|-----------------|---------------------|---------------------|---------------|--------------------------------------------|---------------------------------------------------------------------------------|
| localhost       | Current Load    | OK                  | 09-16-2016 16:44:09 | 0d 2h 26m 36s | 1/4                                        | OK - load average: 0.03, 0.02, 0.05                                             |
|                 | Current Users   | OK                  | 09-16-2016 16:44:46 | 0d 2h 25m 58s | 1/4                                        | USERS OK - 1 users currently logged in                                          |
|                 | HTTP            | WARNING             | 09-16-2016 16:43:22 | 0d 2h 25m 21s | 4/4                                        | HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.001 second response time |
|                 | NTP_TIME        | OK                  | 09-16-2016 16:45:17 | 0d 0h 25m 28s | 1/4                                        | NTP OK: Offset 0.0005682706833 secs                                             |
|                 | PING            | OK                  | 09-16-2016 16:41:00 | 0d 2h 24m 43s | 1/4                                        | PING OK - Packet loss = 0%, RTA = 0.05 ms                                       |
|                 | Root Partition  | OK                  | 09-16-2016 16:41:38 | 0d 2h 24m 6s  | 1/4                                        | DISK OK - free space: / 25307 MB (93% inode=99%);                               |
|                 | SSH             | OK                  | 09-16-2016 16:45:16 | 0d 0h 25m 29s | 1/4                                        | SSH OK - OpenSSH_6.6.1 (protocol 2.0)                                           |
|                 | Swap Usage      | OK                  | 09-16-2016 16:42:53 | 0d 2h 22m 51s | 1/4                                        | SWAP OK - 100% free (3071 MB out of 3071 MB)                                    |
| node01          | Total Processes | OK                  | 09-16-2016 16:43:30 | 0d 2h 22m 13s | 1/4                                        | PROCS OK: 145 processes with STATE = RSZDT                                      |
|                 | Current Load    | OK                  | 09-16-2016 16:40:32 | 0d 0h 5m 13s  | 1/3                                        | OK - load average: 0.00, 0.01, 0.05                                             |
|                 | Current Users   | OK                  | 09-16-2016 16:41:30 | 0d 0h 4m 15s  | 1/3                                        | USERS OK - 1 users currently logged in                                          |
|                 | PING            | OK                  | 09-16-2016 16:38:44 | 0d 0h 17m 1s  | 1/3                                        | PING OK - Packet loss = 0%, RTA = 0.32 ms                                       |
|                 | Root Partition  | OK                  | 09-16-2016 16:42:28 | 0d 0h 3m 17s  | 1/3                                        | DISK OK - free space: / 25612 MB (94% inode=99%);                               |
| Total Processes | OK              | 09-16-2016 16:43:26 | 0d 0h 2m 19s        | 1/3           | PROCS OK: 100 processes with STATE = RSZDT |                                                                                 |

Results 1 - 14 of 14 Matching Services

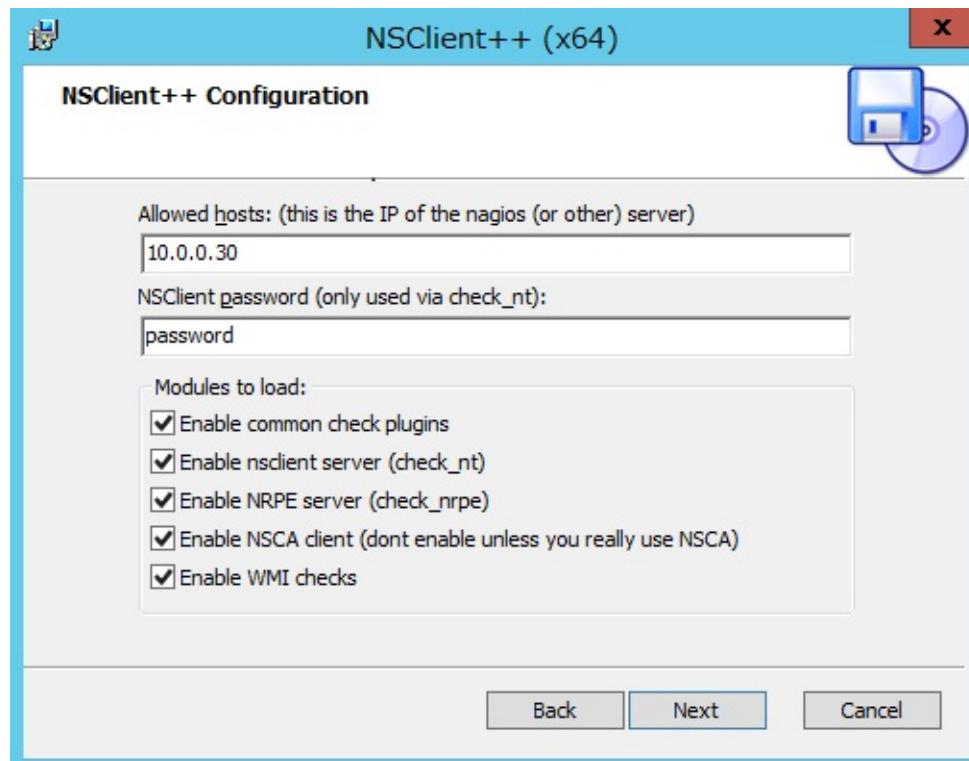
还可以监控网络上的**Windows**服务器。

以添加Windows Server 2012 R2为监控目标为例。

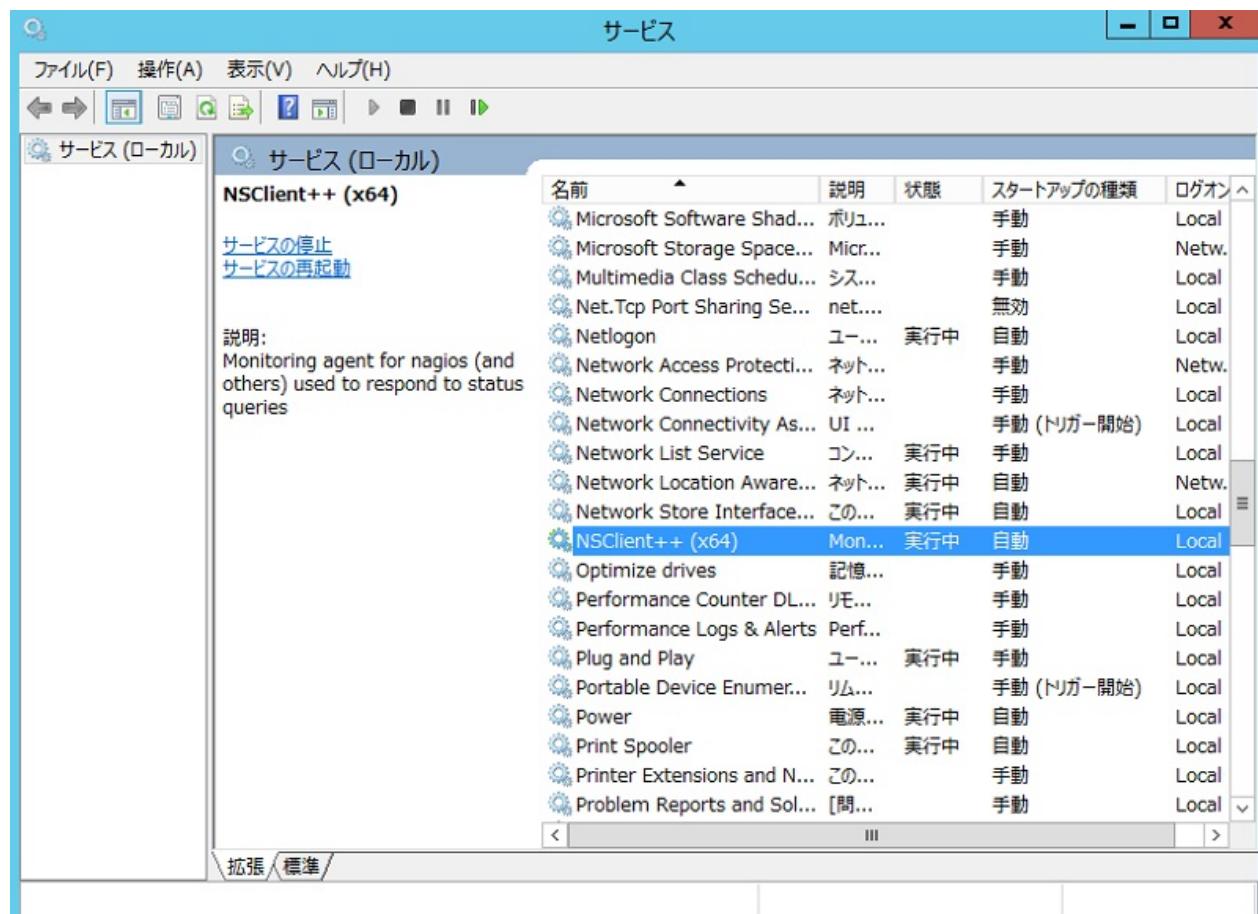
下载“[NSClient++](#)”并将其上传到目标**Windows**服务器上。

安装“[NSClient++](#)”，在安装期间输入Nagios服务器的主机名或IP地址并设置任意密码（该密码用于从Nagios服务器到**Windows**服务器的连接以进行监控），如下所示：

## 12.7. Nagios



安装后，NSClient++已启动（安装程序会设置好Windows防火墙，不需要手动设置）：



配置Nagios服务器：

## 12.7. Nagios

```
yum --enablerepo=epel -y install nagios-plugins-nt
```

编辑 `/etc/nagios/objects/commands.cfg` 文件：

```
# 添加在Windows上设置的密码
command_line $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -v $ARG1
$ $ARG2$ -s password
```

编辑 `/etc/nagios/servers/windows.cfg` 文件：

```
# 定义目标主机
define host{
    use                  windows-server
    host_name           fd3s
    alias               fd3s
    address             10.0.0.100
}

define hostgroup{
    hostgroup_name      windows-servers
    alias               Windows Servers
}

# ping
define service{
    use                  generic-service
    host_name            fd3s
    service_description  PING
    check_command        check_ping!100.0,20%!500.0,60%
}

# NSClient++版本
define service{
    use                  generic-service
    host_name            fd3s
    service_description  NSClient++ Version
    check_command        check_nt!CLIENTVERSION
}

# 正常运行时间
define service{
    use                  generic-service
    host_name            fd3s
    service_description  Uptime
    check_command        check_nt!UPTIME
}
```

```

        }

# CPU负载
define service{
    use          generic-service
    host_name    fd3s
    service_description  CPU Load
    check_command   check_nt!CPULOAD! -l 5,80,90
}

# 内存使用率
define service{
    use          generic-service
    host_name    fd3s
    service_description  Memory Usage
    check_command   check_nt!MEMUSE! -w 80 -c 90
}

# 磁盘空间
define service{
    use          generic-service
    host_name    fd3s
    service_description  C:\ Drive Space
    check_command   check_nt!USEDISKSPACE! -l c -w 8
    0 -c 90
}

# 资源管理器
define service{
    use          generic-service
    host_name    fd3s
    service_description  Explorer
    check_command   check_nt!PROCSTATE! -d SHOWALL -l
    Explorer.exe
}

# IIS
define service{
    use          generic-service
    host_name    winserver
    service_description  W3SVC
    check_command   check_nt!SERVICESTATE! -d SHOWALL
    -l W3SVC
}

```

```
systemctl restart nagios
```

## 12.7. Nagios

可以在管理网站上查看新服务器的状态：

The screenshot shows the Nagios Core web interface running in Mozilla Firefox. The main page displays the following information:

- Current Network Status:** Last Updated: Fri Sep 16 17:27:29 JST 2016, Updated every 90 seconds, Nagios® Core™ 4.0.8 - www.nagios.org, Logged in as nagiosadmin.
- Host Status Totals:** Up: 3, Down: 0, Unreachable: 0, Pending: 0. All Problems: 0, All Types: 3.
- Service Status Totals:** Ok: 20, Warning: 1, Unknown: 0, Critical: 0, Pending: 0. All Problems: 1, All Types: 21.
- Service Status Details For All Hosts:** A table listing services for two hosts: fd3s and localhost. The table columns are Host, Service, Status, Last Check, Duration, Attempt, and Status Information.

| Host      | Service            | Status  | Last Check          | Duration      | Attempt | Status Information                                                                |
|-----------|--------------------|---------|---------------------|---------------|---------|-----------------------------------------------------------------------------------|
| fd3s      | C:\ Drive Space    | OK      | 09-16-2016 17:24:52 | 0d 0h 2m 37s  | 1/3     | c: - total: 99.66 Gb - used: 9.20 Gb (9%) - free 90.46 Gb (91%)                   |
|           | CPU Load           | OK      | 09-16-2016 17:25:59 | 0d 0h 11m 30s | 1/3     | CPU Load 0% (5 min average)                                                       |
|           | Explorer           | OK      | 09-16-2016 17:27:07 | 0d 0h 10m 22s | 1/3     | explorer.exe: Running                                                             |
|           | Memory Usage       | OK      | 09-16-2016 17:26:14 | 0d 0h 11m 15s | 1/3     | Memory usage: total:10111.69 MB - used: 1078.03 MB (11%) - free: 9033.66 MB (89%) |
|           | NSClient++ Version | OK      | 09-16-2016 17:25:21 | 0d 0h 12m 8s  | 1/3     | NSClient++ 0.4.1,90 2013-02-04                                                    |
| localhost | PING               | OK      | 09-16-2016 17:24:04 | 0d 0h 13m 25s | 1/3     | PING OK - Packet loss = 0%, RTA = 0.63 ms                                         |
|           | Uptime             | OK      | 09-16-2016 17:26:29 | 0d 0h 11m 0s  | 1/3     | System Uptime - 0 day(s) 0 hour(s) 39 minute(s)                                   |
|           | Current Load       | OK      | 09-16-2016 17:24:09 | 0d 3h 8m 20s  | 1/4     | OK - load average: 0.00, 0.02, 0.05                                               |
|           | Current Users      | OK      | 09-16-2016 17:24:46 | 0d 3h 7m 42s  | 1/4     | USERS OK - 1 users currently logged in                                            |
|           | HTTP               | WARNING | 09-16-2016 17:23:22 | 0d 3h 7m 5s   | 4/4     | HTTP WARNING: HTTP/1.1 403 Forbidden - 5179 bytes in 0.001 second response time   |
|           | NTP_TIME           | OK      | 09-16-2016 17:25:17 | 0d 1h 7m 12s  | 1/4     | NTP OK: Offset 0.0005807876587 secs                                               |
|           | PING               | OK      | 09-16-2016 17:26:00 | 0d 3h 6m 27s  | 1/4     | PING OK - Packet loss = 0%, RTA = 0.05 ms                                         |
|           | Root Partition     | OK      | 09-16-2016 17:23:13 | 0d 3h 5m 50s  | 1/4     | DISK OK - free space: / 25299 MB (93% inode=99%): SSH OK -                        |

## 12.8. Monitorix

[Monitorix](#)是一个轻量级的系统监控工具。

安装Monitorix：

```
yum --enablerepo=epel -y install monitorix # 从EPEL安装
```

配置Monitorix：

编辑 `/etc/monitorix/monitorix.conf` 文件：

```
# 更改为任意标题
title = Monitorix

# 更改自己的主机名
hostname = dlp.srv.world

# 管理站点的背景颜色
theme_color = white

# 将网络单位更改为bps位每秒（默认值为Bytes per/sec字节每秒）line 12: change network units to bps (default is Bytes per/sec)
netstats_in_bps = y

# 如下更改
<httpd_builtin>
    enabled = y
    host =
    port = 8080
    user = nobody
    group = nobody
    log_file = /var/log/monitorix-httpd
    # 设置管理站点的权限
    hosts_deny = all
    hosts_allow = 10.0.0.0/24
```

```
systemctl start monitorix
systemctl enable monitorix
```

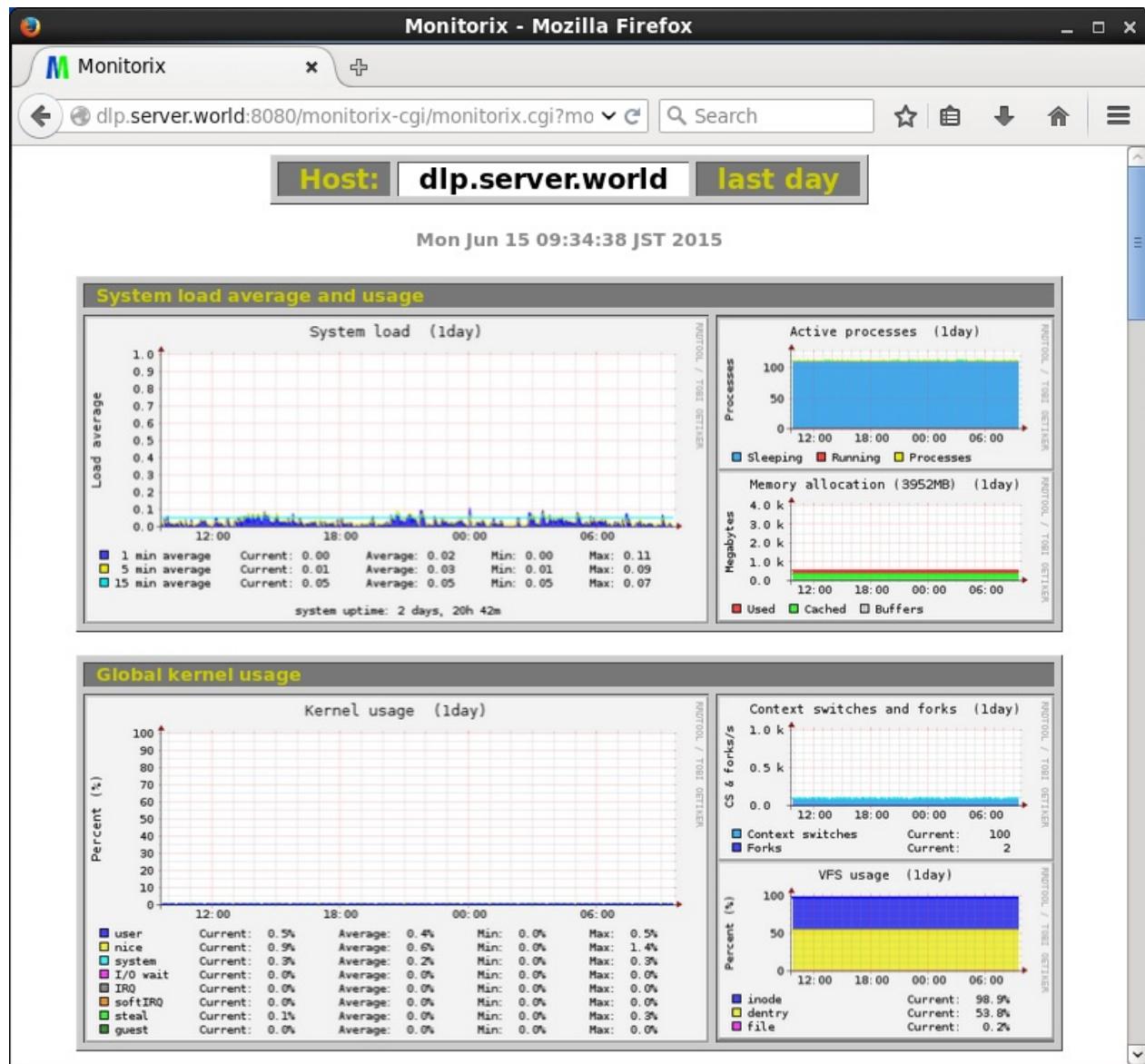
## 12.8. Monitorix

从配置中允许网络中的客户端访问 `http://(Monitorix服务器主机名或IP地址):8080/monitorix/`。Monitorix管理站点显示如下（单击“OK”以查看图表）：



显示图表：

## 12.8. Monitorix



## 12.8. Monitorix



## 12.9. psacct

安装psacct以监控用户活动。

命令历史保存在用户自己的历史文件中，他们可以自己编辑或删除，但是psacct保存由root拥有的所有用户的历史文件。

```
yum -y install psacct # 安装
```

```
systemctl start psacct
systemctl enable psacct
```

如下所示通过 `lastcomm` 命令输出命令历史：

| su              | S | root | ttyS0 | 0.02 | secs | Fri Sep 30 |
|-----------------|---|------|-------|------|------|------------|
| 19:18           |   |      |       |      |      |            |
| bash            | S | cent | ttyS0 | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| cat             |   | cent | ttyS0 | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| ls              |   | cent | ttyS0 | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| bash            | F | cent | ttyS0 | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| ....            |   |      |       |      |      |            |
| ....            |   |      |       |      |      |            |
| systemctl       | S | root | ttyS0 | 0.01 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| pkttymagent     | X | root | ttyS0 | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| systemd-tty-ask |   | root | ttyS0 | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| systemd-cgroups | S | root | —     | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |
| accton          | S | root | —     | 0.00 | secs | Fri Sep 30 |
| 19:18           |   |      |       |      |      |            |

如果输出指定用户的历史记录，使用 `--user` 选项运行：

## 12.9. psacct

---

```
lastcomm --user cent
```

## 12.9. psacct

|             |   |      |       |                      |
|-------------|---|------|-------|----------------------|
| bash        | S | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| cat         |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| ls          |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| bash        | F | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| consoletype |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| bash        | F | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| dircolors   |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| bash        | F | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| tput        |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| tty         |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| grepconf.sh |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| grep        |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| bash        | F | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| id          |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| bash        | F | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| id          |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| bash        | F | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| hostname    |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| bash        | F | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |
| id          |   | cent | ttyS0 | 0.00 secs Fri Sep 30 |
| 19:18       |   |      |       |                      |

如果要输出指定命令的历史记录，使用 `--command` 选项运行：

```
lastcomm --command su
```

|       |   |      |       |           |            |
|-------|---|------|-------|-----------|------------|
| su    | S | cent | ttyS0 | 0.01 secs | Fri Sep 30 |
| 19:23 |   |      |       |           |            |
| su    | S | cent | ttyS0 | 0.01 secs | Fri Sep 30 |
| 19:23 |   |      |       |           |            |
| su    | S | root | ttyS0 | 0.02 secs | Fri Sep 30 |
| 19:18 |   |      |       |           |            |

# 13. 语言开发环境

- 13.1. Ruby
  - 13.1.1. 安装Ruby 2.2
  - 13.1.2. 安装Ruby 2.3
  - 13.1.3. 安装Rails 4
  - 13.1.4. 安装Rails 5
- 13.2. JavaScript
  - 13.2.1. 安装Node.js
  - 13.2.2. 安装Node.js 4
- 13.3. PHP
  - 13.3.1. 安装PHP 5.6
  - 13.3.2. 安装PHP 7.0
  - 13.3.3. 安装PHP 7.1
- 13.4. Python
  - 13.4.1. 安装Python 3.3
  - 13.4.2. 安装Python 3.4
  - 13.4.3. 安装Python 3.5
  - 13.4.4. 安装Django
- 13.5. Java
  - 13.5.1. 安装JDK 8
  - 13.5.2. 安装OpenJDK 8
  - 13.5.3. 安装Tomcat 8

## 13.1. Ruby

Ruby是一种跨平台、面向对象的动态类型编程语言。Ruby体现了表达的一致性和简单性，它不仅是一门编程语言，更是表达想法的一种简练方式。

### 13.1.1. 安装 Ruby 2.2

在CentOS7官方库中的Ruby版本是2.0，如果需要可使用RPM软件包安装2.2。

即使已经安装了2.0，也可以安装，因为2.2位于另一个路径上：

```
yum --enablerepo=centos-scl0-rh -y install rh-ruby22 # 从SCLo安装
```

以上方式安装在 /opt 目录下，要使用它，如下加载环境变量：

```
scl enable rh-ruby22 bash
```

```
ruby -v
```

```
ruby 2.2.2p95 (2015-04-13 revision 50295) [x86_64-linux]
```

```
which ruby
```

```
/opt/rh/rh-ruby22/root/usr/bin/ruby
```

如果想在登录时自动启用Ruby 2.2，编辑 /etc/profile.d/rh-ruby22.sh 文件：

```
#!/bin/bash

source /opt/rh/rh-ruby22/enable
export X_SCLS=`scl enable rh-ruby22 'echo $X_SCLS'``
export PATH=$PATH:/opt/rh/rh-ruby22/root/usr/local/bin
```

### 13.1.2. 安装 Ruby 2.3

```
yum --enablerepo=centos-scl0-rh -y install rh-ruby23 # 从SCLo安装
```

以上方式安装在 /opt 目录下，要使用它，如下加载环境变量：

```
scl enable rh-ruby23 bash
```

```
ruby -v
```

```
ruby 2.3.0p0 (2015-12-25 revision 53290) [x86_64-linux]
```

```
which ruby
```

```
/opt/rh/rh-ruby23/root/usr/bin/ruby
```

如果想在登录时自动启用Ruby 2.3，编辑 /etc/profile.d/rh-ruby23.sh 文件：

```
#!/bin/bash

source /opt/rh/rh-ruby23/enable
export X_SCLS=`scl enable rh-ruby23 'echo $X_SCLS'`"
```

### 13.1.3. 安装 Rails 4

先安装Ruby 2.2。

安装其他所需的软件包：

```
yum --enablerepo=epel,centos-scl-rh -y install rh-ruby22-ruby-devel nodejs libuv gcc make libxml2 libxml2-devel mariadb-devel zlib-devel libxslt-devel #从EPEL，SCLo安装
```

安装Rails 4：

```
gem install bundler
```

```
gem install nokogiri -- --use-system-libraries
```

```
gem install rails --version="~>4.0" --no-ri --no-rdoc
```

```
rails -v
```

### Rails 4.2.6

创建示例应用程序并确保其正常工作：

先[安装MariaDB数据库服务器](#)。

```
gem install mysql2 --no-ri --no-rdoc --with-mysql-
config=/usr/bin/mysql_config
```

```
rails new SampleApp -d mysql
```

```
cd SampleApp
```

编辑 config/database.yml 文件：

```
default: &default
  adapter: mysql2
  encoding: utf8
  pool: 5
  username: root
  password: password # MariaDB密码
  socket: /var/lib/mysql/mysql.sock
```

创建测试应用程序：

```
rake db:create
```

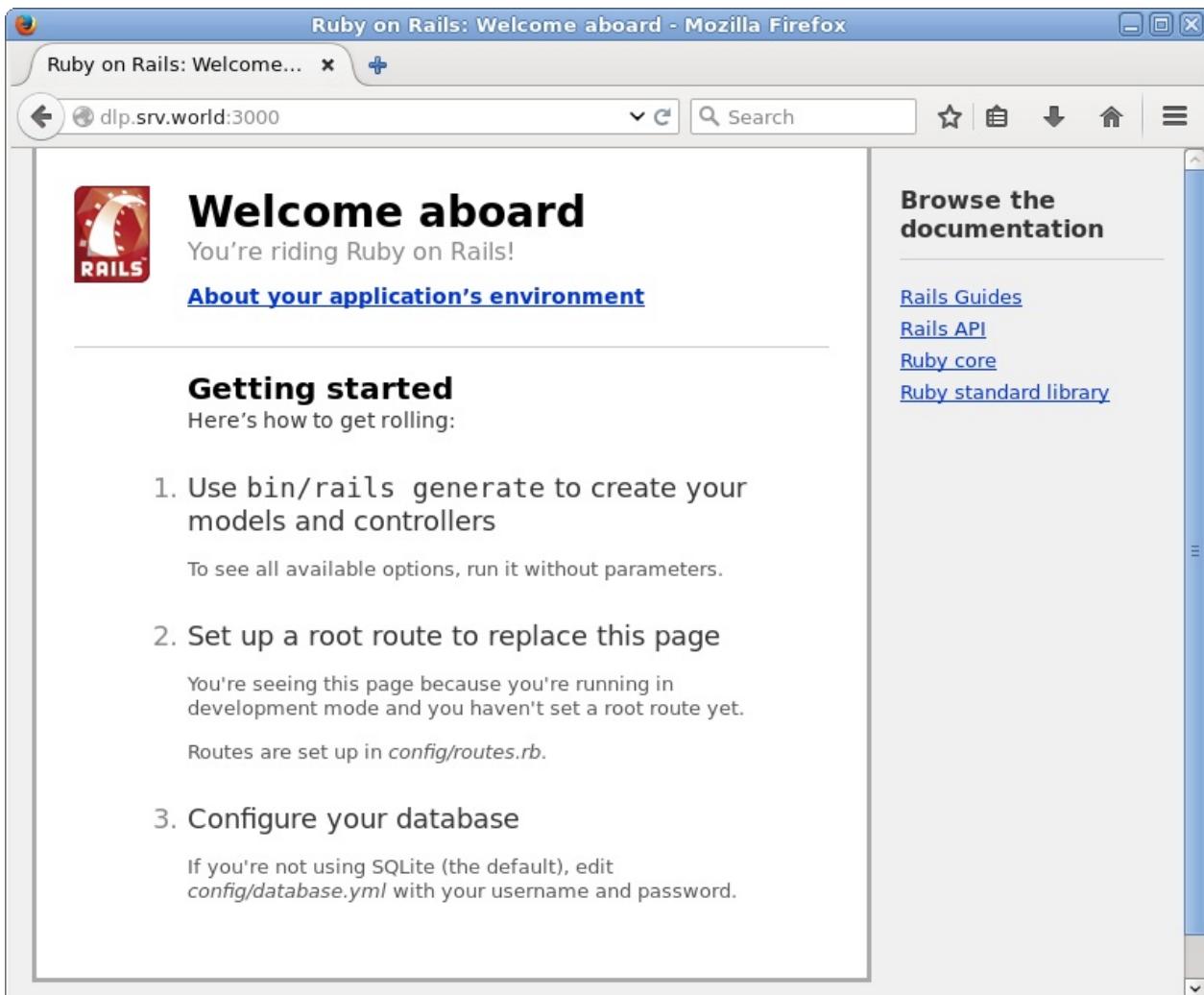
```
rails generate scaffold testapp name:string title:string body:text
```

```
rake db:migrate
```

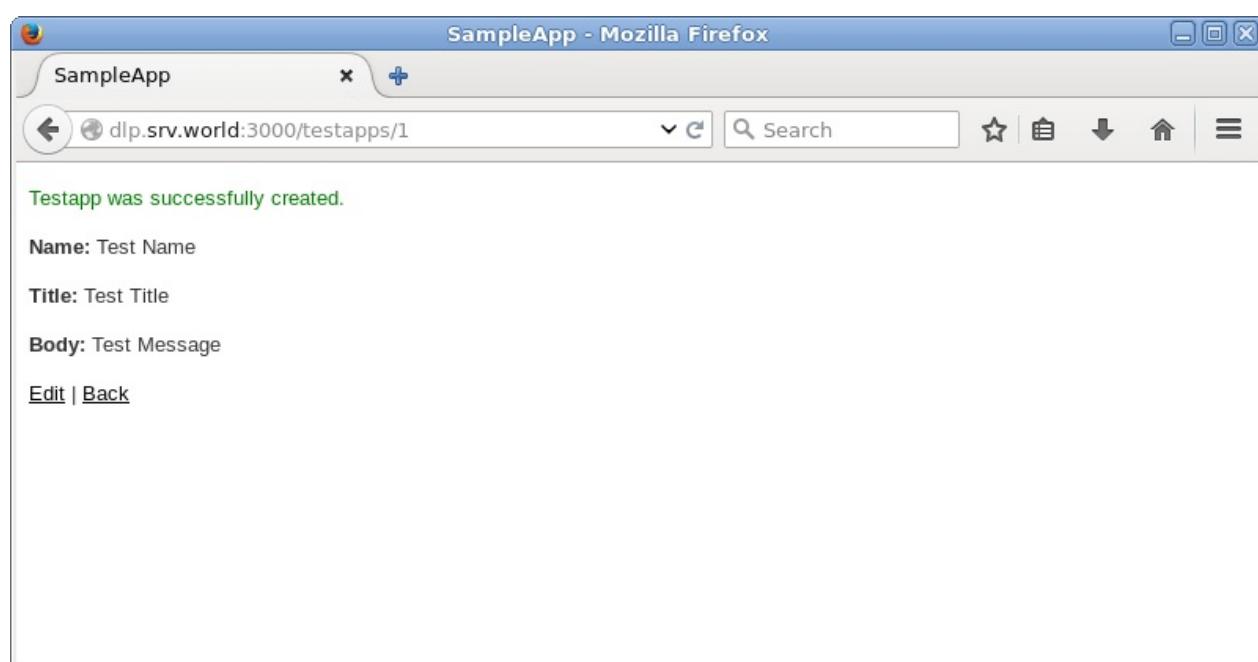
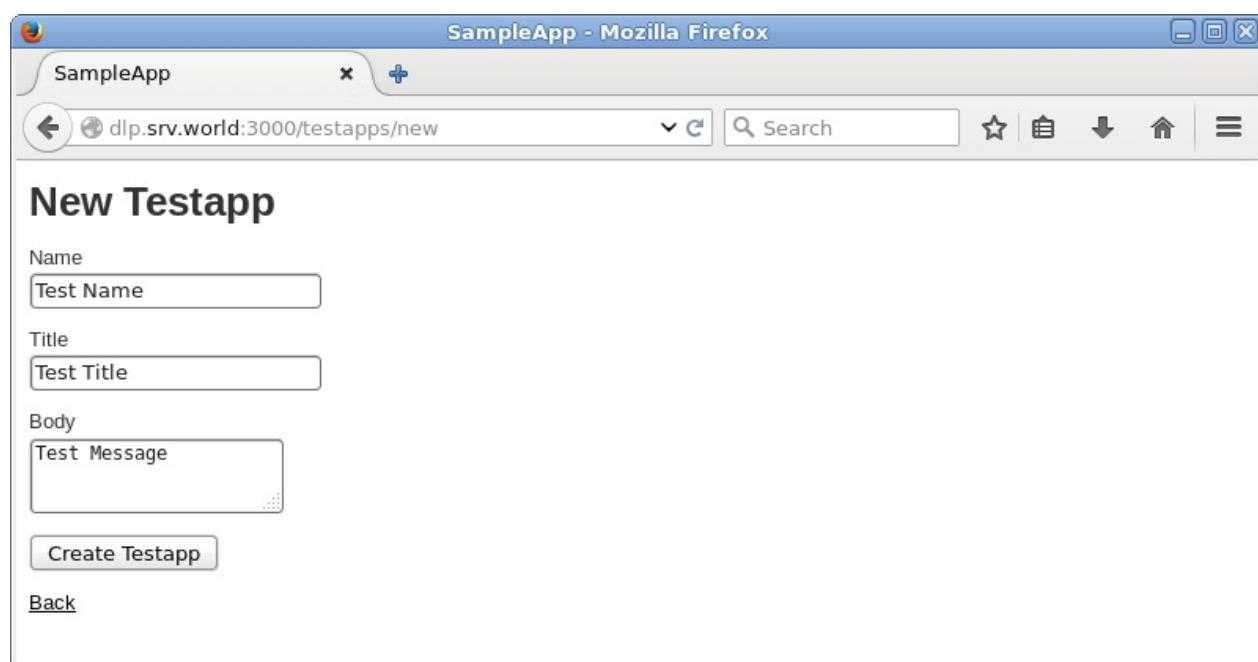
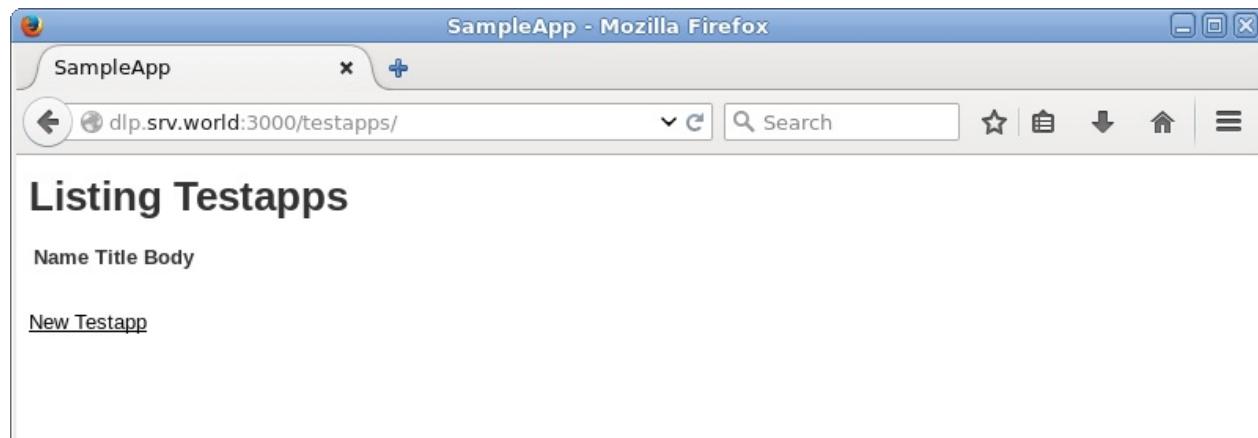
```
rails server --binding=0.0.0.0
```

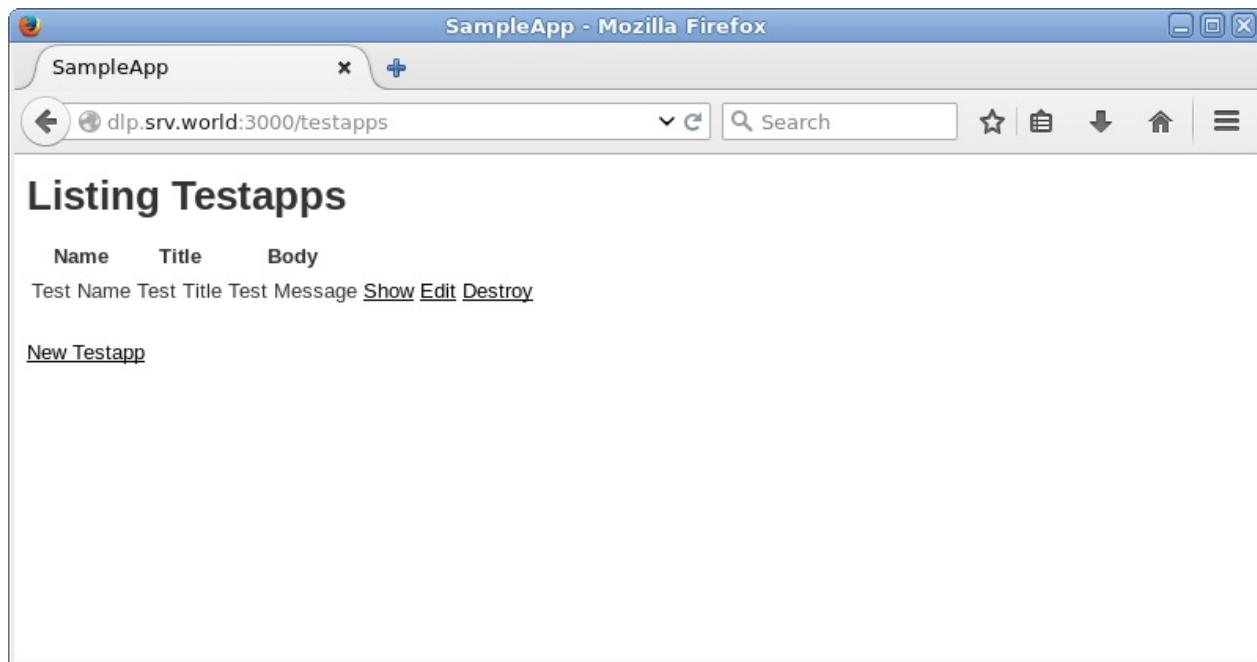
```
=> Booting WEBrick
=> Rails 4.2.6 application starting in development on http://0.0
.0.0:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2016-07-03 19:07:29] INFO  WEBrick 1.3.1
[2016-07-03 19:07:29] INFO  ruby 2.2.2 (2015-04-13) [x86_64-linu
x]
[2016-07-03 19:07:29] INFO  WEBrick::HTTPServer#start: pid=3225
port=3000
```

从客户端计算机访问 `http://(服务器的主机名或IP地址):3000/`。如果以下网站正常显示，表示正常：



访问 `http://(服务器的主机名或IP地址):3000/testapps/`，然后可以使用示例应用程序，如下所示：





### 13.1.4. 安装 Rails 5

先安装Ruby 2.3。

安装其他所需的软件包：

```
yum --enablerepo=epel,centos-scl-o-rh -y install rh-ruby23-ruby-
devel nodejs gcc make libxml2 libxml2-devel mariadb-devel zlib-devel
libxslt-devel #从EPEL，SCLo安装
```

安装Rails 5：

```
gem install bundler

gem install nokogiri -- --use-system-libraries

gem install rails --no-ri --no-rdoc

rails -v
```

Rails 5.0.0

创建示例应用程序并确保其正常工作：

先[安装MariaDB数据库服务器](#)。

## 13.1. Ruby

```
gem install mysql2 --no-ri --no-rdoc --with-mysql-
config=/usr/bin/mysql_config

rails new SampleApp -d mysql

cd SampleApp
```

编辑 config/database.yml 文件：

```
default: &default
  adapter: mysql2
  encoding: utf8
  pool: 5
  username: root
  password: password # MariaDB密码
  socket: /var/lib/mysql/mysql.sock
```

创建测试应用程序：

```
rails db:create
```

```
Created database 'SampleApp_development'
Created database 'SampleApp_test'
```

```
rails generate scaffold testapp name:string title:string body:text
```

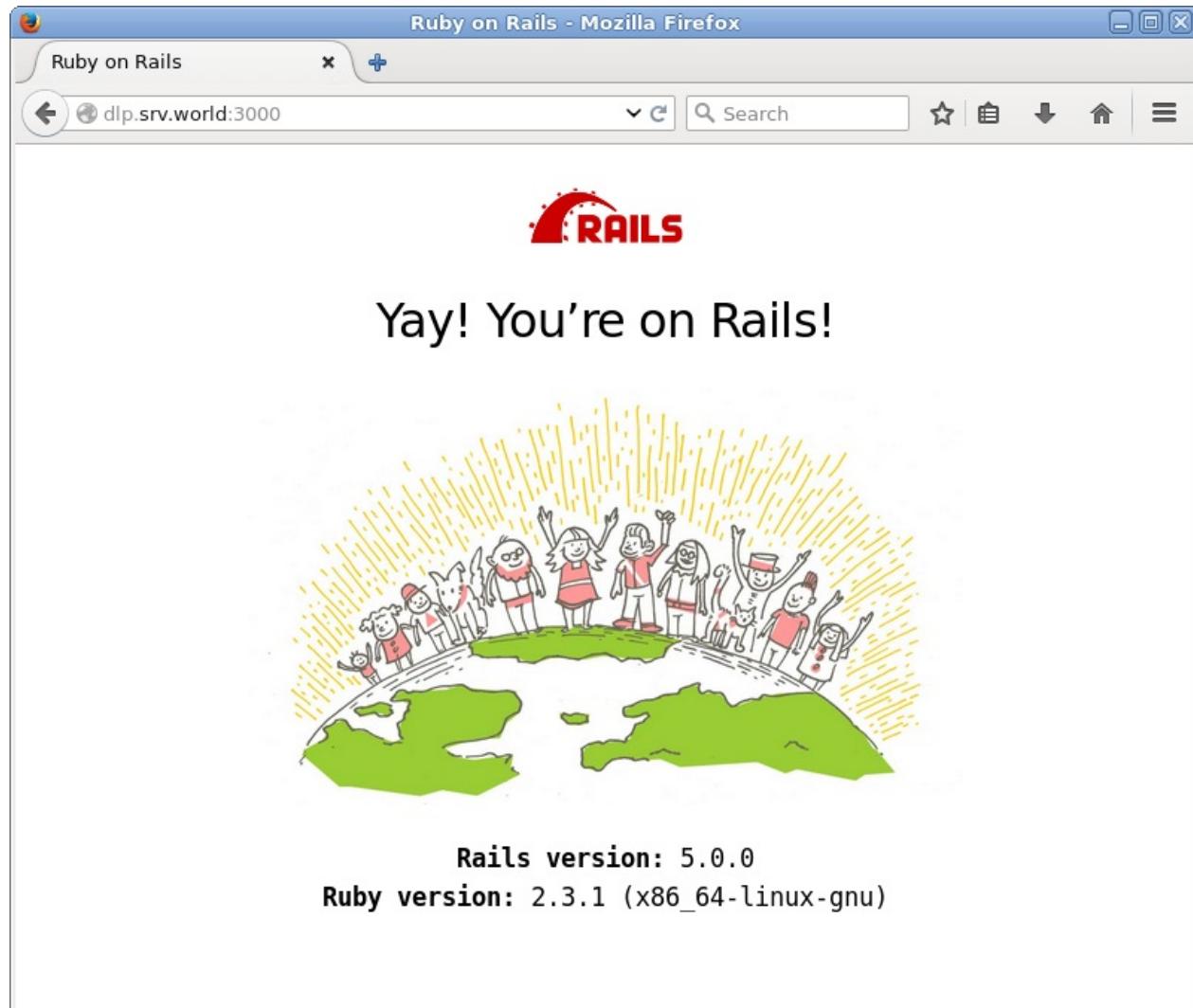
```
rails db:migrate
```

```
rails server --binding=0.0.0.0
```

```
=> Booting Puma
=> Rails 5.0.0 application starting in development on http://0.0
.0.0:3000
=> Run `rails server -h` for more startup options
Puma starting in single mode...
* Version 3.4.0 (ruby 2.3.0-p0), codename: Owl Bowl Brawl
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://0.0.0.0:3000
Use Ctrl-C to stop
```

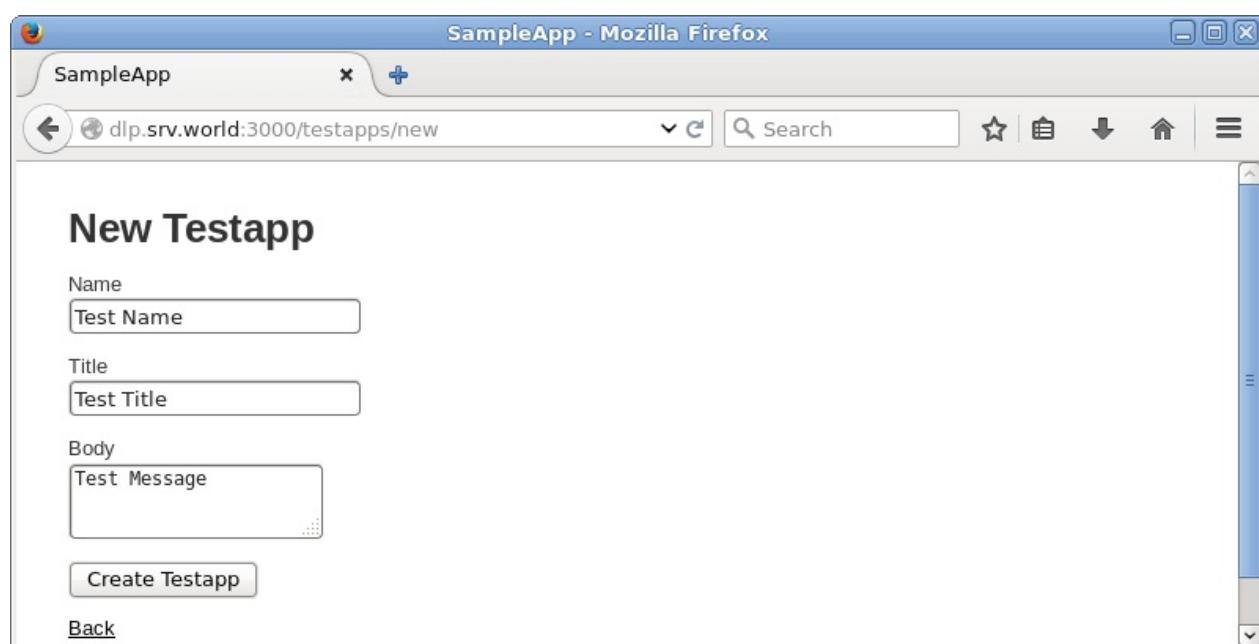
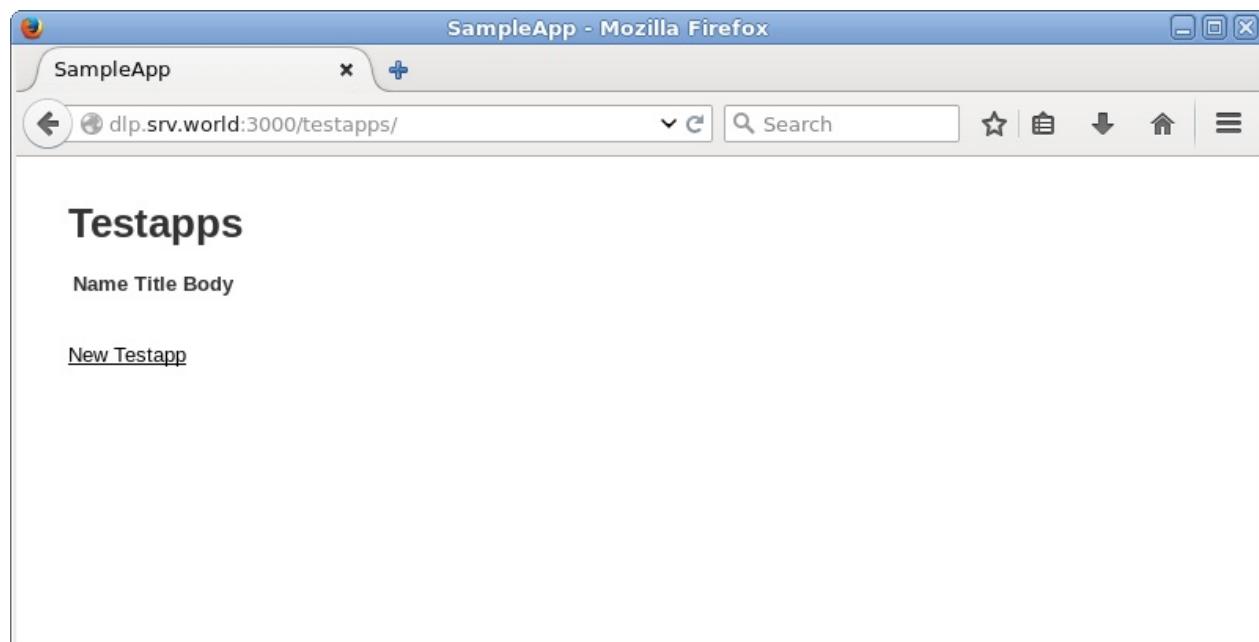
## 13.1. Ruby

从客户端计算机访问 `http://(服务器的主机名或IP地址):3000/` 。如果以下网站正常显示，表示正常：

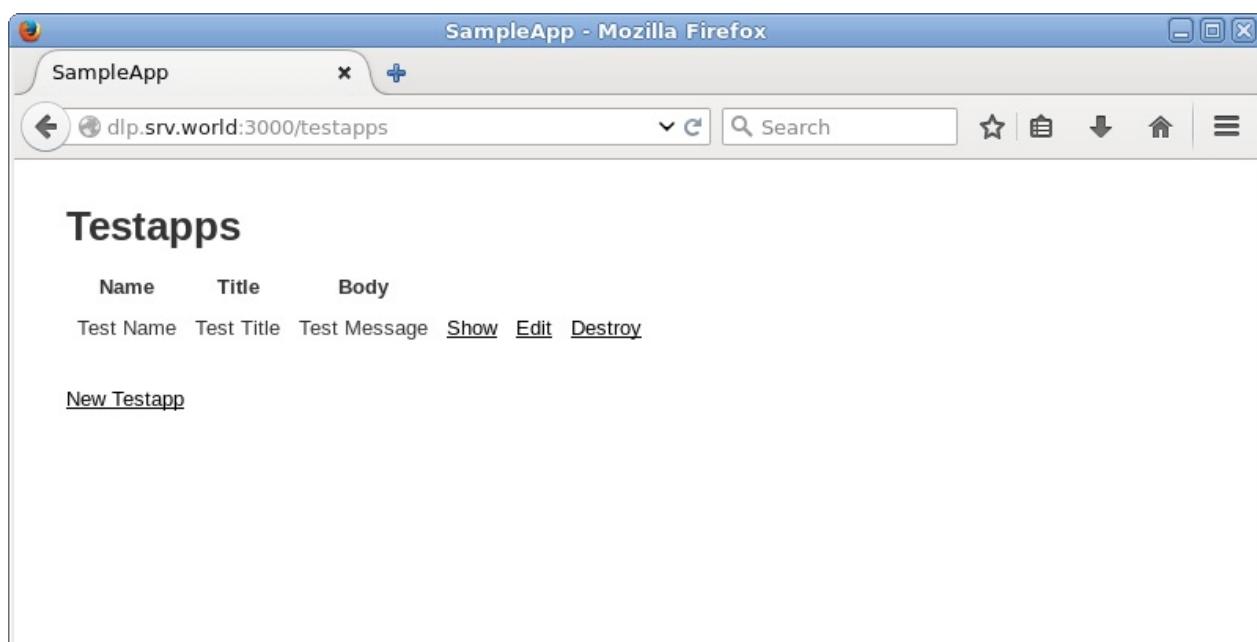
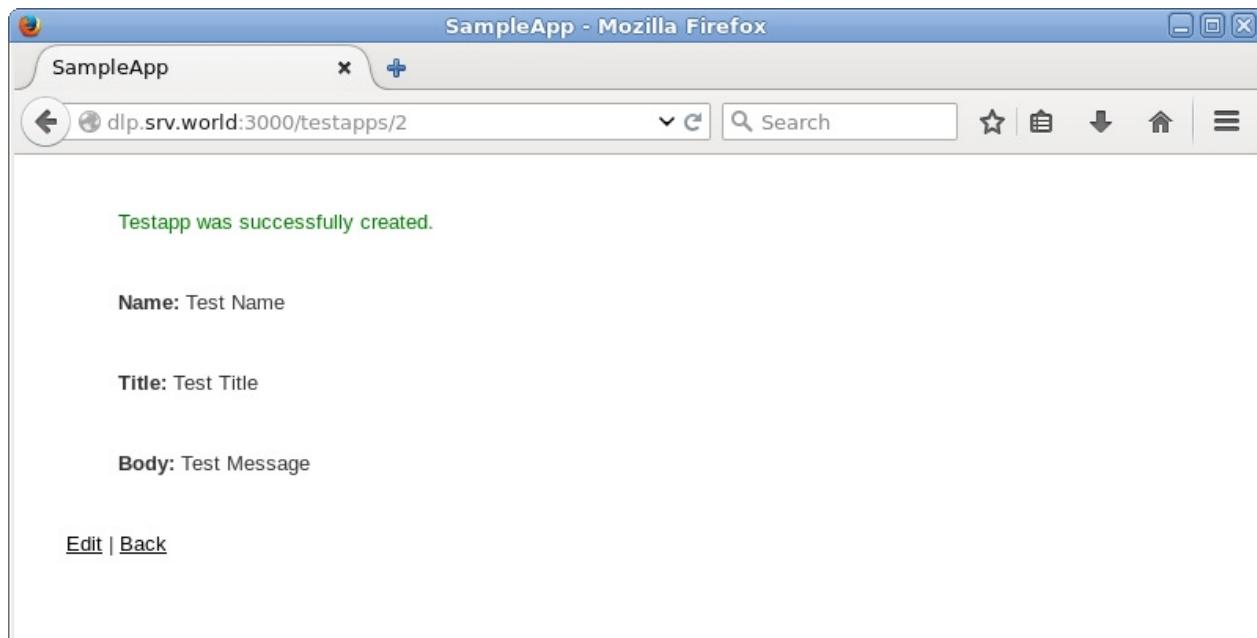


访问 `http://(服务器的主机名或IP地址):3000/testapps/` ，然后可以使用示例应用程序，如下所示：

## 13.1. Ruby



## 13.1. Ruby



## 13.2. JavaScript

安装服务器端JavaScript环境“[Node.js](#)”。

### 13.2.1. 安装Node.js

安装Node.js和[npm](#)（包管理工具）：

```
yum --enablerepo=epel -y install nodejs npm # 从EPEL安装
```

编辑 `helloworld.js` 文件，创建测试工具（可以使用普通用户）：

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('listening on http://127.0.0.1:1337/');
```

```
node helloworld.js & # 运行服务器
```

```
curl http://127.0.0.1:1337/ # 验证（如果回显下面内容表示正常）
```

```
Hello World
```

安装Socket.IO并使用WebSocket创建测试应用程序。

```
npm install socket.io express
```

编辑 `chat.js` 文件：

```
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);

app.get('/', function(req, res){
  res.sendFile(__dirname + '/index.html');
});

io.on('connection', function(socket){
  socket.on('chat message', function(msg){
    io.emit('chat message', msg);
  });
});

http.listen(1337, function(){
  console.log('listening on *:1337');
});
```

编辑 `index.html` 文件：

```

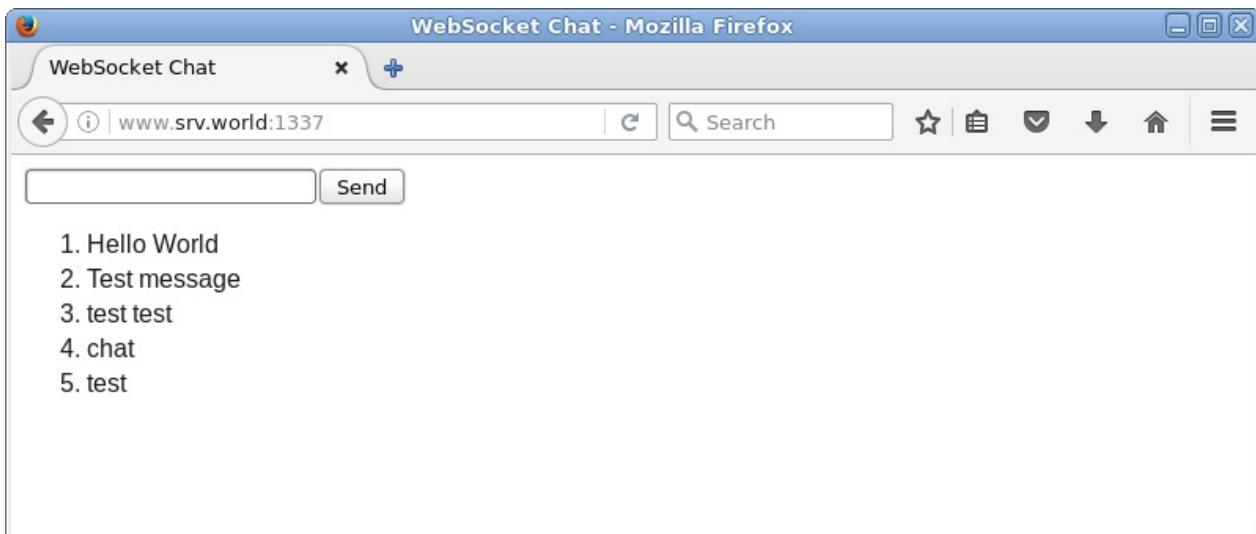
<!DOCTYPE html>
<html>
<head>
<title>WebSocket Chat</title>
</head>
<body>
<form action="">
<input id="sendmsg" autocomplete="off" /><button>Send</button>
</form>
<ul id="messages" style="list-style-type: decimal; font-size: 16px; font-family: Arial;"></ul>
<script src="/socket.io/socket.io.js"></script>
<script src="http://code.jquery.com/jquery.min.js"></script>
<script>
  var socket = io();
  $('form').submit(function(){
    socket.emit('chat message', $('#sendmsg').val());
    $('#sendmsg').val('');
    return false;
  });
  socket.on('chat message', function(msg){
    $('#messages').append($('- ' +
      text(msg));
  });
</script>
</body>
</html>

```

编辑 node chat.js 文件：

```
listening on *:1337
```

从客户端计算机访问 `http://(服务器的主机名或IP地址):1337/`，以确认示例应用程序正常工作：



## 13.2.2. 安装Node.js 4

即使已经安装了0.1.x（可能从EPEL安装），也可以安装，因为4.x位于另一个路径上：

```
yum --enablerepo=centos-scl-o-rh -y install rh-nodejs4 # 从SCLo安装
```

以上方式安装在 /opt 目录下，要使用它，如下加载环境变量：

```
scl enable rh-nodejs4 bash
```

```
node -v
```

v4.4.2

```
which node
```

/opt/rh/rh-nodejs4/root/usr/bin/node

如果想在登录时自动启用Node.js 4，编辑 /etc/profile.d/rh-nodejs4.sh 文件：

```
#!/bin/bash

source /opt/rh/rh-nodejs4/enable
export X_SCLS="`scl enable rh-nodejs4 'echo $X_SCLS'`"
```



## 13.3. PHP

**PHP (Hypertext Preprocessor)** 是一种脚本语言，主要是用途在于处理动态网页，也包含了命令列执行接口（command line interface），或者产生图形使用者接口（GUI）程式。

### 13.3.1. 安装PHP 5.6

在CentOS7官方库中的PHP版本是5.4，如果需要可使用RPM软件包安装5.6。

即使已经安装了5.4，也可以安装，因为5.6位于另一个路径上：

```
yum --enablerepo=centos-scl0-rh -y install rh-php56 # 从SCLo安装
```

以上方式安装在 /opt 目录下，要使用它，如下加载环境变量：

```
scl enable rh-php56 bash
```

```
php -v
```

```
PHP 5.6.5 (cli) (built: Mar 23 2016 19:17:38)
Copyright (c) 1997-2014 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2014 Zend Technologies
```

```
which php
```

```
/opt/rh/rh-php56/root/usr/bin/php
```

如果想在登录时自动启用PHP 5.6，编辑 /etc/profile.d/rh-php56.sh 文件：

```
#!/bin/bash

source /opt/rh/rh-php56/enable
export X_SCLS="`scl enable rh-php56 'echo $X_SCLS'`"
```

要在Apache httpd上使用5.6，如下配置PHP-FPM：

### 13.3. PHP

```
yum --enablerepo=centos-scl-o-rh -y install rh-php56-php-fpm # 从  
SCLo安装
```

编辑 `/etc/httpd/conf.d/php.conf` 文件：

```
# 如下更改  
<FilesMatch \.php$>  
#     SetHandler application/x-httdp-php  
    SetHandler "proxy:fcgi://127.0.0.1:9000"  
</FilesMatch>
```

```
systemctl start rh-php56-php-fpm  
systemctl enable rh-php56-php-fpm  
systemctl restart httpd
```

如果你想5.6嵌入Apache httpd，配置如下：

```
yum --enablerepo=centos-scl-o-rh -y install rh-php56-php # 从SCLo安  
装
```

编辑 `/etc/httpd/conf.modules.d/10-php.conf` 文件：

```
# 如下更改  
<IfModule prefork.c>  
    LoadModule php5_module /opt/rh/httpd24/root/usr/lib64/httpd/mo  
dules/librh-php56-php5.so  
</IfModule>
```

```
systemctl restart httpd
```

```
echo '<?php phpinfo(); ?>' >  
/opt/rh/httpd24/root/var/www/html/info.php # 创建phpinfo以验证  
  
curl http://localhost/info.php | grep 'PHP Version' | tail -1 | sed  
-e 's/<[^>]*>//g'
```

```
% Total    % Received % Xferd  Average Speed   Time     Time
Time      Current                                         Dload  Upload   Total   Spent
Left     Speed
100 68819      0 68819      0       0  5529k      0 --::--:-- --::--:--
--::--:-- 6109k
PHP Version 5.6.5
```

### 13.3.2. 安装PHP 7.0

在CentOS7官方库中的PHP版本是5.4，如果需要可使用RPM软件包安装7.0。

即使已经安装了5.4，也可以安装，因为7.0位于另一个路径上：

```
yum --enablerepo=remi-safe -y install php70 # 从Remi安装
```

以上方式安装在 /opt 目录下且 /bin/php70 链接被创建，要使用它，如下加载环境变量：

```
php70 -v
```

```
PHP 7.0.8 (cli) (built: Jun 22 2016 10:57:20) ( NTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
```

```
which php70
```

```
/bin/php70
```

```
ll /bin/php70
```

```
lrwxrwxrwx 1 root root 32 Jul 6 09:58 /bin/php70 -> /opt/remi/ph
p70/root/usr/bin/php
```

```
scl enable php70 bash # 使用SCL工具加载环境变量
```

```
php -v
```

```
PHP 7.0.8 (cli) (built: Jun 22 2016 10:57:20) ( NTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
```

如果想在登录时自动启用PHP 7.0，编辑 `/etc/profile.d/php70.sh` 文件：

```
#!/bin/bash

source /opt/remi/php70/enable
export X_SCLS=`scl enable php70 'echo $X_SCLS'`"
```

要在Apache httpd上使用7.0，如下配置PHP-FPM：

```
yum --enablerepo=remi-safe -y install php70-php-fpm # 从Remi安装

编辑 /etc/httpd/conf.d/php.conf 文件：
```

```
# 如下更改
<FilesMatch \.php$>
#     SetHandler application/x-httpd-php
     SetHandler "proxy:fcgi://127.0.0.1:9000"
</FilesMatch>
```

```
systemctl start php70-php-fpm
systemctl enable php70-php-fpm
systemctl restart httpd
```

如果你想7.0嵌入Apache httpd，配置如下：

```
yum --enablerepo=remi-safe -y install php70-php # 从Remi安装

mv /etc/httpd/conf.modules.d/10-php.conf
/etc/httpd/conf.modules.d/10-php.conf.org # 重命名并禁用旧版本（如果存在）

systemctl restart httpd

echo '<?php phpinfo(); ?>' > /var/www/html/info.php # 创建phpinfo以验证
```

```
curl http://localhost/info.php | grep 'PHP Version' | tail -1 | sed -e 's/<[^>]*>//g'
```

| % Total           | % Received | % Xferd | Average Speed | Time  | Time   |       |       |                       |
|-------------------|------------|---------|---------------|-------|--------|-------|-------|-----------------------|
| Time              | Current    |         |               | Dload | Upload | Total | Spent |                       |
| Left              | Speed      |         |               |       |        |       |       |                       |
| 100               | 68819      | 0       | 68819         | 0     | 0      | 5529k | 0     | --::--::-- --::--::-- |
|                   |            |         |               |       |        |       |       | --::--::-- 6109k      |
| PHP Version 7.0.8 |            |         |               |       |        |       |       |                       |

### 13.3.3. 安装PHP 7.1

在CentOS7官方库中的PHP版本是5.4，如果需要可使用RPM软件包安装7.1。

即使已经安装了5.4，也可以安装，因为7.1位于另一个路径上：

```
yum --enablerepo=remi-safe -y install php71 # 从Remi安装
```

以上方式安装在 /opt 目录下且 /bin/php71 链接被创建，要使用它，如下加载环境变量：

```
php71 -v
```

```
PHP 7.1.0alpha2 (cli) (built: Jun 22 2016 18:26:46) ( NTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.1.0-dev, Copyright (c) 1998-2016 Zend Technologies
```

```
which php71
```

```
/bin/php71
```

```
ll /bin/php71
```

```
lrwxrwxrwx 1 root root 32 Jul 6 11:10 /bin/php71 -> /opt/remi/php71/root/usr/bin/php
```

### 13.3. PHP

```
scl enable php71 bash # 使用SCL工具加载环境变量
```

```
php -v
```

```
PHP 7.1.0alpha2 (cli) (built: Jun 22 2016 18:26:46) ( NTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.1.0-dev, Copyright (c) 1998-2016 Zend Technologies
```

如果想在登录时自动启用PHP 7.0，编辑 `/etc/profile.d/php71.sh` 文件：

```
#!/bin/bash

source /opt/remi/php71/enable
export X_SCLS=`scl enable php71 'echo $X_SCLS'`"
```

要在Apache httpd上使用7.1，如下配置PHP-FPM：

```
yum --enablerepo=remi-safe -y install php71-php-fpm # 从Remi安装
编辑 /etc/httpd/conf.d/php.conf 文件：
```

```
# 如下更改
<FilesMatch \.php$>
#     SetHandler application/x-httpd-php
     SetHandler "proxy:fcgi://127.0.0.1:9000"
</FilesMatch>
```

```
systemctl start php71-php-fpm
systemctl enable php71-php-fpm
systemctl restart httpd
```

如果你想7.0嵌入Apache httpd，配置如下：

```
yum --enablerepo=remi-safe -y install php71-php # 从Remi安装
mv /etc/httpd/conf.modules.d/10-php.conf
/etc/httpd/conf.modules.d/10-php.conf.org # 重命名并禁用旧版本（如果存在）
```

### 13.3. PHP

```
systemctl restart httpd
```

```
echo '<?php phpinfo(); ?>' > /var/www/html/info.php # 创建phpinfo以验证
```

```
curl http://localhost/info.php | grep 'PHP Version' | tail -1 | sed -e 's/<[^>]*>//g'
```

```
% Total    % Received % Xferd  Average Speed   Time     Time  
Time   Current                                         Dload  Upload   Total   Spent  
Left   Speed  
100 68819      0 68819      0       0  5529k      0  --::--:--  --::--:--  
--::--:-- 6109k  
PHP Version 7.1.0
```

## 13.4. Python

Python是一种面向对象的解释性的计算机程序设计语言，也是一种功能强大而完善的通用型语言，已经具有十多年的发展历史，成熟且稳定。

### 13.4.1. 安装 Python 3.3

在CentOS7官方库中的Python版本是2.7，如果需要可使用RPM软件包安装3.3。

即使已经安装了2.7，也可以安装，因为3.3位于另一个路径上：

```
yum --enablerepo=centos-scl-o-rh -y install python33 # 从SCLo安装
```

以上方式安装在 /opt 目录下，要使用它，如下加载环境变量：

```
scl enable python33 bash
```

```
python -V
```

Python 3.3.2

```
which python
```

/opt/rh/python33/root/usr/bin/python

如果想在登录时自动启用Python 3.3，编辑 /etc/profile.d/python33.sh 文件：

```
#!/bin/bash

source /opt/rh/python33/enable
export X_SCLS="`scl enable python33 'echo $X_SCLS'`"
```

### 13.4.2. 安装 Python 3.4

在CentOS7官方库中的Python版本是2.7，如果需要可使用RPM软件包安装3.4。

即使已经安装了2.7，也可以安装，因为3.4位于另一个路径上：

```
yum --enablerepo=centos-scl0-rh -y install rh-python34 # 从SCLo安装
```

以上方式安装在 /opt 目录下，要使用它，如下加载环境变量：

```
scl enable rh-python34 bash
```

```
python -V
```

Python 3.4.2

```
which python
```

/opt/rh/rh-python34/root/usr/bin/python

如果想在登录时自动启用Python 3.4，编辑 /etc/profile.d/python34.sh 文件：

```
#!/bin/bash

source /opt/rh/rh-python35/enable
export X_SCLS="`scl enable rh-python35 'echo $X_SCLS'`"
```

### 13.4.3. 安装Python 3.5

在CentOS7官方库中的Python版本是2.7，如果需要可使用RPM软件包安装3.5。

即使已经安装了2.7，也可以安装，因为3.5位于另一个路径上：

```
yum --enablerepo=centos-scl0-rh -y install rh-python35 # 从SCLo安装
```

以上方式安装在 /opt 目录下，要使用它，如下加载环境变量：

```
scl enable rh-python35 bash
```

```
python -V
```

### Python 3.5.1

```
which python
```

```
/opt/rh/rh-python35/root/usr/bin/python
```

如果想在登录时自动启用Python 3.5，编辑 `/etc/profile.d/python35.sh` 文件：

```
#!/bin/bash

source /opt/rh/rh-python35/enable
export X_SCLS="`scl enable rh-python35 'echo $X_SCLS'`"
```

### 13.4.4. 安装Django

Django是Python编程语言驱动的一个开源模型-视图-控制器（MVC）风格的Web应用程序框架。

```
yum --enablerepo=epel -y install python-virtualenv # 从EPEL安装一些软件包
```

在Virtualenv环境下安装Django（可以使用普通用户）：

```
virtualenv venv
```

```
cd ~/venv
```

```
source bin/activate
```

```
pip install django # 由于pip源速度原因，可以将 pip install 换成 pip install -i http://mirrors.aliyun.com/pypi/simple
```

```
Downloading/unpacking django
  Downloading Django-1.8.3.tar.gz (7.3MB): 7.3MB downloaded
  Running setup.py egg_info for package django

    warning: no previously-included files matching '__pycache__'
    found under directory '*'
    warning: no previously-included files matching '*.py[co]' fo
    und under directory '*'
  Installing collected packages: django
    Running setup.py install for django

    warning: no previously-included files matching '__pycache__'
    found under directory '*'
    warning: no previously-included files matching '*.py[co]' fo
    und under directory '*'
    changing mode of build/scripts-2.7/django-admin.py from 664
    to 775
    changing mode of /home/cent/venv/bin/django-admin.py to 775
    Installing django-admin script to /home/cent/venv/bin
Successfully installed django
Cleaning up...
# 上面的“warning”没有问题
```

```
django-admin --version
```

```
1.8.3
```

```
deactivate
```

创建测试项目：

```
cd ~/venv
```

```
source bin/activate
```

```
django-admin startproject testproject # 创建“testproject”
```

```
cd testproject
```

```
python manage.py migrate # 配置数据库（默认为SQLite）
```

```
python manage.py createsuperuser # 创建管理用户
```

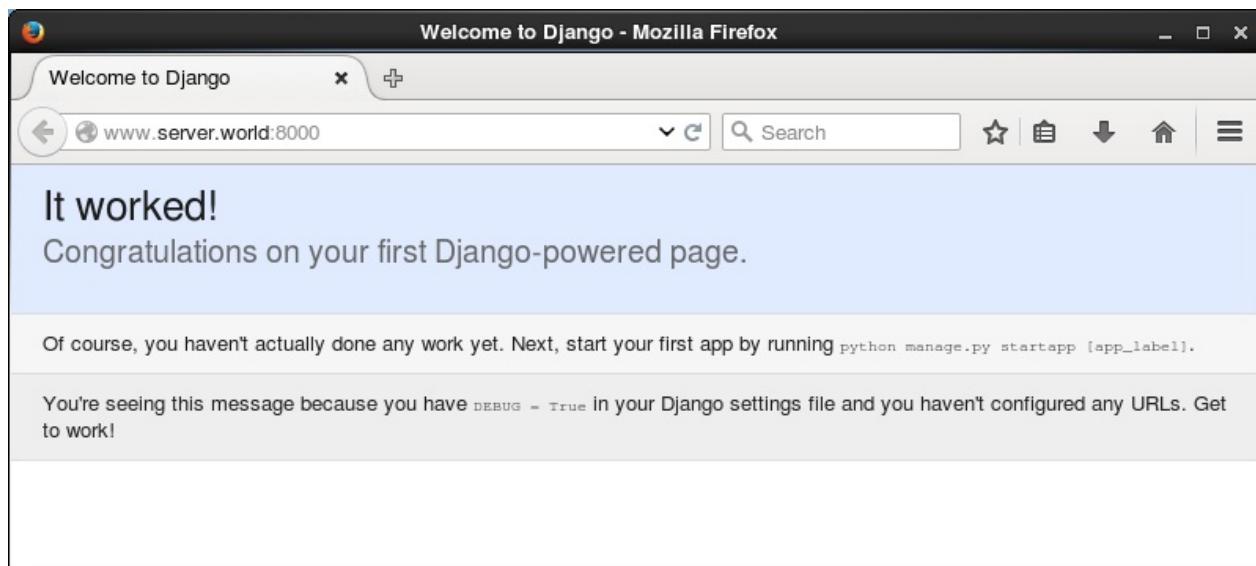
```
Username (leave blank to use 'cent'): cent
Email address: cent@www.srv.world
Password:
Password (again):
Superuser created successfully.
```

```
python manage.py runserver 0.0.0.0:8000 # 启动服务器
```

```
Performing system checks...

System check identified no issues (0 silenced).
August 04, 2015 - 08:09:47
Django version 1.8.3, using settings 'testproject.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

从客户端计算机访问 `http://(server's hostname or IP address):8000/`。如果显示以下网站，表示正常：



可以在 `http://(server's hostname or IP address):8000/admin` 上使用管理员用户管理网站：

## 13.4. Python

The image contains two screenshots of a Django administration interface within a Mozilla Firefox browser window.

**Screenshot 1: Log in | Django site admin - Mozilla Firefox**

This screenshot shows the Django login page. The title bar says "Log in | Django site admin - Mozilla Firefox". The address bar shows "www.server.world:8000/admin/login/?next=/admin/". The main content is a "Django administration" box with fields for "Username" (containing "cent") and "Password" (containing "\*\*\*\*\*"). A "Log in" button is at the bottom right.

**Screenshot 2: Site administration | Django site admin - Mozilla Firefox**

This screenshot shows the main Django site administration page. The title bar says "Site administration | Django site admin - Mozilla Firefox". The address bar shows "www.server.world:8000/admin/". The main content area is titled "Django administration" and "Site administration". It lists "Authentication and Authorization" with "Groups" and "Users" sections, each with "Add" and "Change" buttons. On the right, there's a "Recent Actions" sidebar with "My Actions" and "None available". The top right of the main area says "Welcome, cent. View site / Change password / Log out".

创建测试应用程序：

```
cd ~/venv  
source bin/activate  
cd testproject  
python manage.py startapp testapp
```

编辑 testapp/views.py 文件：

```
# 添加以下内容到最后
from django.http import HttpResponse
def main(request):
    html = '<html>\n' \
           '<body>\n' \
           '<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">\n' \
               'Django Test Page\n' \
           '</div>\n' \
           '</body>\n' \
           '</html>\n'
    return HttpResponse(html)
```



```
mv testproject/urls.py testproject/urls.py.org
```

编辑 `testproject/urls.py` 文件：

```
from django.conf.urls import patterns, url

urlpatterns = patterns('',
    url(r'^testapp/$', 'testapp.views.main'),
)
```

编辑 `testproject/settings.py` 文件：

```
# 添加“testapp”
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'testapp',
)
```

```
python manage.py runserver 0.0.0.0:8000
```

## 13.4. Python

---

从客户端计算机访问 `http://(server's hostname or IP address):8000/testapp/`。如果显示以下网站，表示正常：



## 13.5. Java

### 13.5.1. 安装JDK 8

安装[Java SE Development Kit 8 \(JDK8\)](#) 以构建Java开发环境。

如果安装了OpenJDK，可以先卸载：

```
yum -y remove java-*
```

检查系统是否自带OpenJDK：

```
rpm -qa |grep java  
rpm -qa |grep jdk  
rpm -qa |grep gcj
```

卸载：

```
rpm -qa | grep java | xargs rpm -e --nodeps
```

下载（确认最新版本的下载链接）并安装JDK 8（目前最新版是9了）：

```
curl -LO -H "Cookie: oraclelicense=accept-securebackup-cookie" \  
"http://download.oracle.com/otn-pub/java/jdk/8u71-b15/jdk-8u71-1  
inux-x64.rpm"
```

```
rpm -Uvh jdk-8u71-linux-x64.rpm
```

```

Preparing...          #####[100%
]
1:jdk1.8.0_71      #####[100%
]
Unpacking JAR files...
    rt.jar...
    jsse.jar...
    charsets.jar...
    tools.jar...
    localedata.jar...
    jfxrt.jar...

```

安装JDK 9.0.4时，最后会出现类似下面的一些提示，没具体研究，不过应该不影响使用：

```
cp: cannot stat '/usr/java/jdk-9.0.4/lib/desktop/icons/hicolor/1
6x16/apps/sun-java.png': No such file or directory
```

查看版本：

```
java -version
```

编辑 /etc/profile 文件：

```
# 添加以下内容到最后
export JAVA_HOME=/usr/java/default
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=.:$JAVA_HOME/jre/lib:$JAVA_HOME/lib:$JAVA_HOME/
lib/tools.jar
```

```
source /etc/profile
```

如果已安装另一个版本的JDK，按如下所示更改默认值：

```
alternatives --config java
```

There are 2 programs which provide 'java'.

| Selection | Command                                                                |
|-----------|------------------------------------------------------------------------|
| *+ 1      | /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.65-3.b17.el7.x86_64/jre/bin/java |
| 2         | /usr/java/jdk1.8.0_71/jre/bin/java                                     |

# 选择需要的版本 ("2")

Enter to keep the current selection[+], or type selection number  
: 2

创建测试程序并确认是否正常工作：

编辑 day.java 文件：

```
import java.util.Calendar;

class day {
    public static void main(String[] args) {
        Calendar cal = Calendar.getInstance();
        int year = cal.get(Calendar.YEAR);
        int month = cal.get(Calendar.MONTH) + 1;
        int day = cal.get(Calendar.DATE);
        int hour = cal.get(Calendar.HOUR_OF_DAY);
        int minute = cal.get(Calendar.MINUTE);
        System.out.println(year + "/" + month + "/" + day + " "
+ hour + ":" + minute);
    }
}
```

javac day.java # 编译

java day # 运行

2015/3/16 20:30

## 13.5.2. 安装OpenJDK 8

安装[OpenJDK 8](#)以配置Java开发环境。

Oracle JDK包括编译器，但OpenJDK 8的编译器包含在openjdk-devel中：

```
yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel  
dirname $(readlink $(readlink $(which java))) # 确认路径  
  
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.102-1.b14.el7_2.x86_64/jre  
/bin
```

编辑 `/etc/profile` 文件：

```
# 添加以下内容到最后  
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.102-1.b14  
.el7_2.x86_64  
export PATH=$PATH:$JAVA_HOME/bin  
export CLASSPATH=.:$JAVA_HOME/jre/lib:$JAVA_HOME/lib:$JAVA_HOME/  
lib/tools.jar
```

```
source /etc/profile
```

如果已安装另一个版本的JDK，按如下所示更改默认值：

```
alternatives --config java
```

```
There are 2 programs which provide 'java'.  
  
Selection      Command  
-----  
 1            /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.102-1.b14.e  
l7.x86_64/jre/bin/java  
*+ 2            /usr/java/jdk1.8.0_71/jre/bin/java
```

```
# # 选择需要的版本 ("1")  
Enter to keep the current selection[+], or type selection number  
: 1
```

创建测试程序并确认是否正常工作：

编辑 `day.java` 文件：

```

import java.util.Calendar;

class day {
    public static void main(String[] args) {
        Calendar cal = Calendar.getInstance();
        int year = cal.get(Calendar.YEAR);
        int month = cal.get(Calendar.MONTH) + 1;
        int day = cal.get(Calendar.DATE);
        int hour = cal.get(Calendar.HOUR_OF_DAY);
        int minute = cal.get(Calendar.MINUTE);
        System.out.println(year + "/" + month + "/" + day + " "
+ hour + ":" + minute);
    }
}

```

`javac day.java # 编译`

`java day # 运行`

2016/9/30 19:46

### 13.5.3. 安装Tomcat 8

Tomcat是一个小型的轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP程序的首选。

先按照前面的内容安装好JDK。

[下载Tomcat 8](#)（确认最新版本的下载链接）：

```
curl -O http://ftp.riken.jp/net/apache/tomcat/tomcat-
8/v8.0.20/bin/apache-tomcat-8.0.20.tar.gz
```

```
tar zxvf apache-tomcat-8.0.20.tar.gz
```

```
mv apache-tomcat-8.0.20 /usr/tomcat8
```

```
useradd -M -d /usr/tomcat8 tomcat8
```

```
chown -R tomcat8. /usr/tomcat8
```

创建Systemd设置文件：

编辑 /usr/lib/systemd/system/tomcat8.service 文件：

```
[Unit]
Description=Apache Tomcat 8
After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]
Type=oneshot
ExecStart=/usr/tomcat8/bin/startup.sh
ExecStop=/usr/tomcat8/bin/shutdown.sh
ExecReload=/bin/kill -s HUP $MAINPID
RemainAfterExit=yes
User=tomcat8
Group=tomcat8

[Install]
WantedBy=multi-user.target
```

```
systemctl start tomcat8
systemctl enable tomcat8
```

在客户端上启动Web浏览器并访问 `http://(服务器的主机名或IP地址):8080/`，  
Tomcat默认站点显示如下：

The screenshot shows a Mozilla Firefox browser window with the title "Apache Tomcat/8.0.20 - Mozilla Firefox". The address bar shows "dip.server.world:8080". The main content area displays the Apache Tomcat 8.0.20 splash page. At the top, it says "Apache Tomcat/8.0.20" and features the "The Apache Software Foundation" logo. A green box contains the message: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below this is a cartoon cat logo. To the right, there's a sidebar with links to "Server Status", "Manager App", and "Host Manager". The main content area includes sections for "Developer Quick Start" with links like "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", "Servlet Specifications", and "Tomcat Versions". There are also three yellow boxes: one for "Managing Tomcat", one for "Documentation" (with links to "Tomcat 8.0 Documentation", "Tomcat 8.0 Configuration", and "Tomcat Wiki"), and one for "Getting Help" (with links to "FAQ and Mailing Lists" and several mailing lists: "tomcat-announce", "tomcat-users", "taglibs-user", and "tomcat-dev").

创建一个测试servlet，显示当前日期和时间，以确认是否正常工作：

```
mkdir /usr/tomcat8/webapps/ROOT/WEB-INF/classes
```

```
chown tomcat8. /usr/tomcat8/webapps/ROOT/WEB-INF/classes
```

```
cd /usr/tomcat8/webapps/ROOT/WEB-INF/classes
```

编辑 `daytime.java` 文件：

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Calendar;

public class daytime extends HttpServlet {
    public void doGet(HttpServletRequest request
                      , HttpServletResponse response)
        throws IOException, ServletException{
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Calendar cal = Calendar.getInstance();
        out.println("<html>\n<head>\n<title>DayTime</title>\n</head>\n<body>");
        out.println("<div style=\"font-size: 40px; text-align: center; font-weight: bold>" );
        out.println(cal.get(Calendar.YEAR) + "/" + (cal.get(Calendar.MONTH) + 1) + "/" +
                   cal.get(Calendar.DATE) + " " + cal.get(Calendar.HOUR_OF_DAY) + ":" + cal.get(Calendar.MINUTE));
        out.println("</div>\n</body>\n</html>");
    }
}

```

```
javac -classpath /usr/tomcat8/lib/servlet-api.jar daytime.java
```

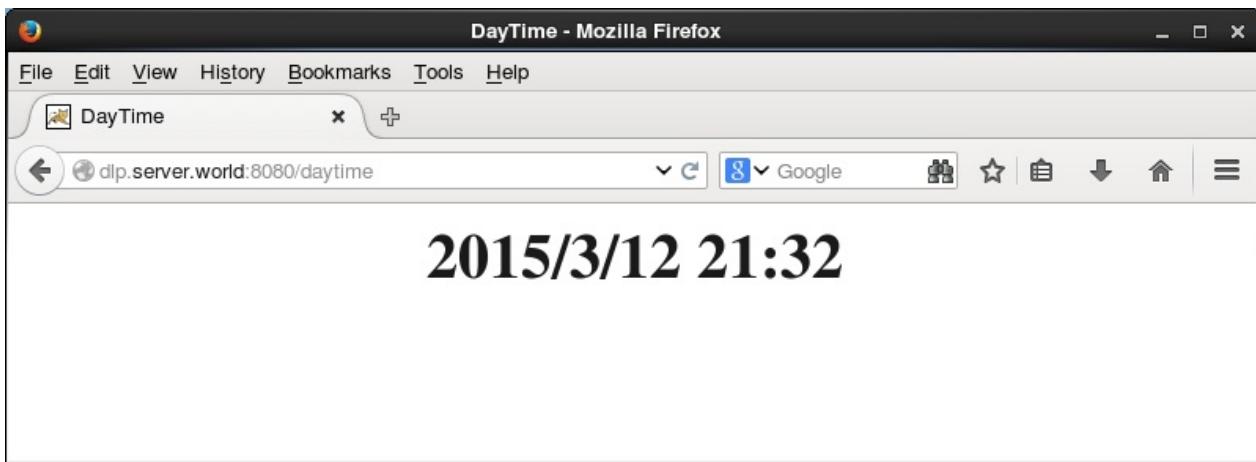
编辑 /usr/tomcat8/webapps/ROOT/WEB-INF/web.xml 文件：

```

# 在<web-app> - </web-app>之间添加以下内容
<servlet>
    <servlet-name>daytime</servlet-name>
    <servlet-class>daytime</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>daytime</servlet-name>
    <url-pattern>/daytime</url-pattern>
</servlet-mapping>

```

访问 [http://\(服务器的主机名或IP地址\):8080/daytime](http://(服务器的主机名或IP地址):8080/daytime)，以确认是否正常工作：



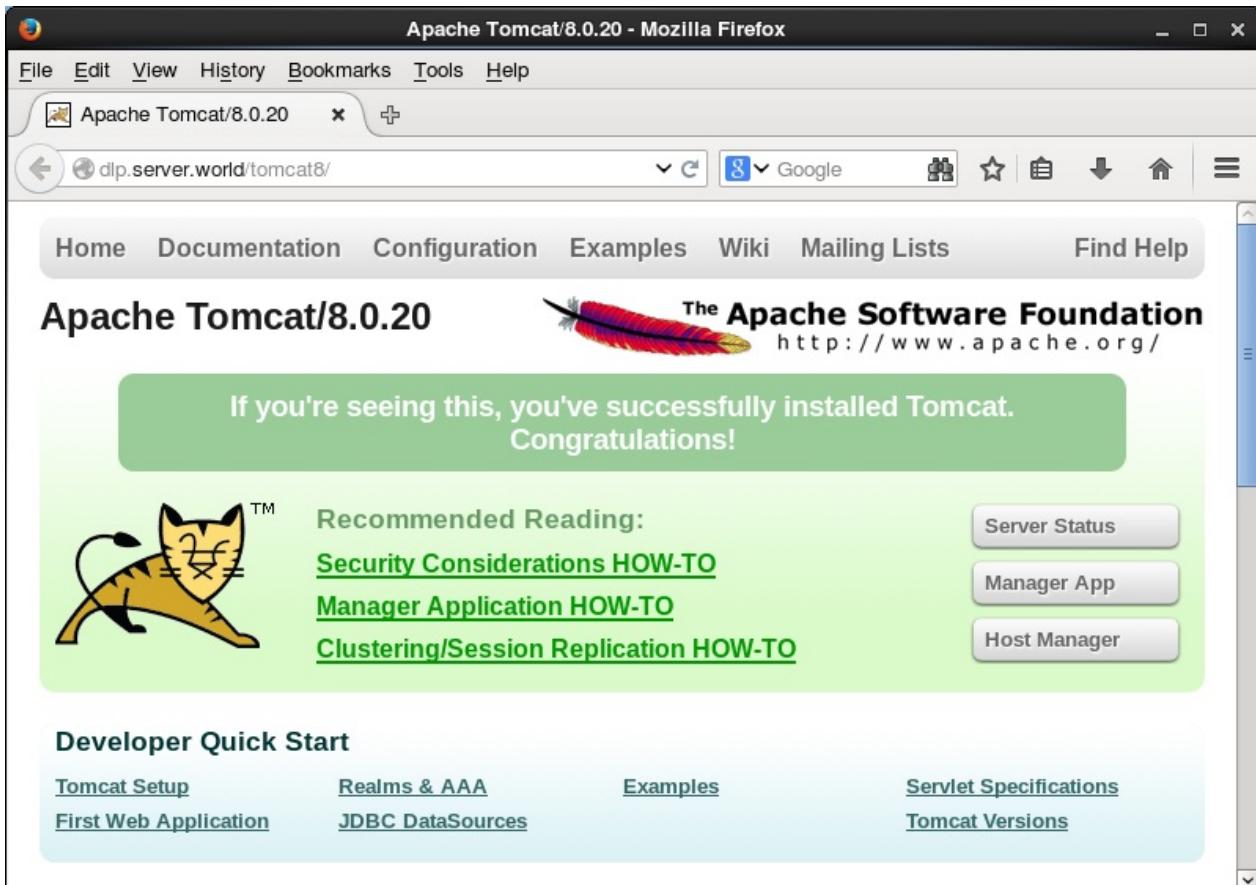
如果想不指定8080访问，如下配置Apache http服务器（首先[安装并启动Apache http服务器](#)）：

编辑 `/etc/httpd/conf.d/proxy_ajp.conf` 文件：

```
# 添加到最后
ProxyPass /tomcat8/ ajp://localhost:8009/
```

```
systemctl restart httpd
```

访问 `http://(服务器的主机名或IP地址)/tomcat8/`，以确认是否正常工作：





# 14. 云计算

- 14.1. OpenStack

## 14.1. OpenStack

OpenStack是IaaS（基础设施即服务）组件，让任何人都可以自行建立和提供云端运算服务。

以OpenStack Ocata为例。

### 14.1.1. 概述

### 14.1.2. 预先要求

### 14.1.3. 配置Keystone

### 14.1.4. 配置Glance

### 14.1.5. 配置Nova

### 14.1.6. 配置Neutron

### 14.1.7. 添加虚拟机映像

### 14.1.8. 配置网络

### 14.1.9. 运行实例

# 15. 认证服务器

- 15.1. FreeRADIUS
- 15.2. privacyIDEA
  - 15.2.1. 基础设置
  - 15.2.2. 安装privacyIDEA
  - 15.2.3. 配置privacyIDEA
  - 15.2.4. 配置FreeRADIUS3
  - 15.2.5. 其他配置

## 15.1. FreeRADIUS

**RADIUS** : Remote Authentication Dial In User Service，远程用户拨号认证系统由RFC2865，RFC2866定义，是目前应用最广泛的AAA（Authentication、Authorization及Accounting）协议。

**FreeRADIUS** 包括RADIUS服务器，BSD许可的客户端库，PAM库和Apache模块。在大多数情况下，FreeRADIUS是指RADIUS服务器。

这里以安装FreeRADIUS和daloRADIUS为例：

如果参照了[这里禁用了IPv6](#)，需要重新开启IPv6，否则在默认情况下不能启动FreeRADIUS服务，暂时没研究怎么设置。

可以在 `/etc/hosts` 中 `127.0.0.1` 一行加入本机主机名。

先[禁用SELinux](#)。

安装好[MariaDB数据库](#)。

创建数据库（也可以使用[phpMyAdmin操作](#)）：

```
mysql -u root -p
```

```
# 输入数据库root密码后登入
# 创建radius数据库
>CREATE DATABASE IF NOT EXISTS `radius` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
# 创建数据库用户“radius”密码为“radiuspassword”并授权
>CREATE USER 'radius'@'localhost' IDENTIFIED BY 'radiuspassword';
>
>GRANT ALL PRIVILEGES ON `radius`.* TO 'radius'@'localhost';
>FLUSH PRIVILEGES;
>quit
```

安装[httpd](#)服务器（详细配置可[参照这里](#)）：

```
yum -y groupinstall "Development Tools"
```

```
yum -y install httpd httpd-devel
```

## 15.1. FreeRADIUS

```
yum --enablerepo=epel -y install php php-devel php-mysql php-pear  
php-pear-DB php-common php-gd php-mbstring php-mcrypt php-xml
```

```
systemctl start httpd  
systemctl enable httpd
```

安装FreeRADIUS：

```
yum -y install freeradius freeradius-mysql freeradius-utils
```

```
systemctl start radiusd  
systemctl enable radiusd
```

firewalld防火墙规则：

```
firewall-cmd --add-service=http --permanent  
firewall-cmd --add-port={1812/udp,1813/udp} --permanent  
firewall-cmd --reload
```

配置FreeRADIUS：

```
mysql -u root -p radius < /etc/raddb/mods-  
config/sql/main/mysql/schema.sql # 输入数据库root密码后执行
```

```
ln -s /etc/raddb/mods-available/sql /etc/raddb/mods-enabled/ #  
在 /etc/raddb/mods-enabled/ 下建立软链接
```

编辑 /etc/raddb/mods-available/sql 文件：

```
# 修改对应内容，未提及或是与默认相同的不用修改
sql {
driver = "rlm_sql_mysql"
dialect = "mysql"
server = "localhost"
port = 3306
login = "radius" # 上面新建的数据库
password = "radiuspassword" # 数据库密码
radius_db = "radius"
}
read_clients = yes
client_table = "nas"
```

```
chgrp -h radiusd /etc/raddb/mods-enabled/sql # 更
改 /etc/raddb/mods-enabled/sql 到“radius”组权限
```

调整启动顺序：

编辑 `/etc/systemd/system/multi-user.target.wants/radiusd.service` 文件：

```
#在[Unit]下的After=最后加上mariadb.service
After=syslog.target network.target ipa.service dirsrv.target krb
5kdc.service mariadb.service
```

完成后运行：

```
systemctl daemon-reload
systemctl restart radiusd
```

安装配置Dalaradius：

```
# PEAR package DB in order to access the database. To install it
, execute at the command line:
pear install DB

# PEAR packages Mail and Mail_Mime to send notifications by email.
To install them, execute at the command line:
pear install -a Mail
pear install -a Mail_Mime
```

```
wget https://github.com/lirantal/daloradius/archive/master.zip # 也可以从sourceforge下载
```

```
unzip master.zip # 解压下载的文件

mv daloradius-master/ daloradius

mv daloradius /var/www/html/

cd /var/www/html/daloradius

mysql -u root -p radius < contrib/db/fr2-mysql-daloradius-and-
freeradius.sql # 输入数据库root密码后执行

mysql -u root -p radius < contrib/db/mysql-daloradius.sql # 输入数
据库root密码后执行
```

```
chown -R apache:apache /var/www/html/daloradius/
chmod 664 /var/www/html/daloradius/library/daloradius.conf.php
```

编辑 /var/www/html/daloradius/library/daloradius.conf.php 文件：

```
# 修改数据库连接信息
$configValues['CONFIG_DB_ENGINE'] = 'mysql';
$configValues['CONFIG_DB_HOST'] = 'localhost';
$configValues['CONFIG_DB_PORT'] = '3306';
# 实测时用“radius”用户在Daloradius页面操作某些内容时会提示权限问题，可以
改为数据库用户“root”后提示就没有了（具体是否有影响不清楚）
$configValues['CONFIG_DB_USER'] = 'radius'; # 上面新建的数据库
$configValues['CONFIG_DB_PASS'] = 'radiuspassword'; # 数据库密码
$configValues['CONFIG_DB_NAME'] = 'radius';
```

```
systemctl restart mariadb  
systemctl restart radiusd  
systemctl restart httpd
```

访问 `http://(服务器的主机名或IP地址)/daloradius`，使用用户名“administrator”，默认密码“radius”登录。

可以通过网页进行操作。

进入网页后可以查看Logs（daloRADIUS、RADIUS、System、Boot等的日志），但需要权限，对应的文件分别为：`http`配置文件中设置的对应日志、`/var/log/radius/radius.log`、`/var/log/messages`、`/var/log/dmesg`。如果允许查看，需要设置权限，日志文件设置为644，日志文件所在目录需要为755。根据需要自己选择是否设置

设置中文，不知道为什么没有直接选出中文的选项，这里把意大利语改为中文：

```
cd /var/www/daloradius/lang  
mv it.php it.php.bak  
mv zh.php it.php
```

修改完成后，网页操作：Config -> Language Settings -> 下拉选择Italian -> Apply

进入debug模式：

```
systemctl stop radiusd  
radiusd -X
```

新建用户名和密码（按照上面默认配置完成后新建的用户只有明文密码能够通过认证，具体见后面的更新）后，使用以下命令：

```
radtest 用户名 密码 localhost 0 testing123 # testing123  
为 /etc/raddb/clients.conf 文件中“client localhost”下面“secret”的值
```

测试完成后按Ctrl + C结束，运行 `systemctl start radiusd` 启动RADIUS服务器

更新1：查看数据库中 `radcheck` 表能发现，网页中选择“MD5-Password”后生成的密码，与在数据库中用MD5生成的是一样的值，但使用“SHA1-Password”时生成的密码，与数据库SHA1的值不一样。比如在网页中使用“SHA1-Password”时生成一个“123456”的密码，此时使用“123456”是无法认证成功的，数据库中运行 `UPDATE radius . radcheck SET value = SHA1('123456') WHERE radcheck . id = 1;` 修改后，可以认证成功。最终结果，使用MD5相对方便一点。（一开始没有选择明文，结果都保存为明文密码，后来不知怎么又可以加密了）

更新2：daloradius使用的一些表使用的是latin1编码，尝试了改为utf8，会报错，比如 `userinfo` 表修改后，用户列表不能查看。不过就算不修改latin1，中文字符在数据库中是以乱码存储，但网页端能显示出中文名称

## 15.2. privacyIDEA

由于软件更新，按照下面方法安装后使用不了，等待以后的更新

privacyIDEA是一个多因素认证解决方案。支持所有通用的OTP（一次性密码）设备，包括Google Authenticator，FreeOTP，eToken Pass，OTP cards，和Yubikey等。可轻松添加新设备，可从文件、LDAP和活动目录获取用户信息。

详细信息可以[参考这里](#)。

笔者注：以下是大部分根据[官方教程](#)来编写的安装privacyIDEA和FreeRADIUS的教程，并参考了部分[superlc320@GitBook](#)。

### 15.2.1. 基础设置

先禁用SELinux。

时间设置：

```
yum -y install ntp  
timedatectl set-timezone Asia/Shanghai  
timedatectl set-ntp yes
```

编辑 /etc/hosts 文件

```
# 根据需要设置hosts，“yourip”可为服务器IP或127.0.0.1  
yourip privacyideaserver privacyideaserver.domain
```

firewalld防火墙规则：

```
firewall-cmd --add-service={http,https} --permanent  
firewall-cmd --add-port={1812/udp,1813/udp} --permanent  
firewall-cmd --reload
```

### 15.2.2. 安装privacyIDEA

安装一些必要的软件：

```
yum --enablerepo=epel -y install net-tools wget NetworkManager-tui  
links nmap rkhunter open-vm-tools libxml2-devel libxslt-devel #  
“libxml2-devel”和“libxslt-devel”官方教程没有，但若不安装，后面 pip 安  
装“lxml”时会报错
```

```
yum -y groupinstall 'Development Tools'
```

重启后继续安装：

```
yum --enablerepo=epel -y install mariadb-server httpd mod_wsgi m  
od_ssl python-devel gcc mariadb-devel libjpeg-devel \  
freeradius freeradius-utils freeradius-perl openldap-devel perl-  
libwww-perl perl-Config-IniFiles \  
perl-Try-Tiny perl-Data-Dump perl-JSON perl-LWP-Protocol-http* p  
ython-virtualenv libffi-devel \  
freetype-devel libpng-devel postgresql-devel
```

编辑 /etc/my.cnf 文件：

```
# 在[mysqld]下加入一行  
character-set-server=utf8
```

```
systemctl enable radiusd  
systemctl start radiusd  
systemctl enable mariadb  
systemctl start mariadb  
systemctl enable httpd  
systemctl start httpd
```

```
mysql_secure_installation # 具体数据库配置可参考这里
```

```
mysql -u root -p
```

```
# 新建privacyidea数据库
>create database privacyidea;
# 新建localhost的用户“privacyidea”（密码“unknown”），并授予其privacyidea数据库的所有权限
>grant all privileges on privacyidea.* to "privacyidea"@"localhost" identified by "unknown";
>FLUSH PRIVILEGES;
>exit
```

reboot

安装privacyIDEA：

建立python虚拟环境：

```
virtualenv /opt/privacyIDEA
```

```
source /opt/privacyIDEA/bin/activate
```

安装依赖的python包：

在浏览器打

开<https://github.com/privacyidea/privacyidea/blob/master/requirements.txt>，将内容复制到/opt/privacyIDEA/requirements.txt

由于pip源速度原因，将官方教程的 pip install 换成 pip install -i <http://mirrors.aliyun.com/pypi/simple>：

```
cd /opt/privacyIDEA
```

```
pip install -i http://mirrors.aliyun.com/pypi/simple -r requirements.txt
```

```
pip install -i http://mirrors.aliyun.com/pypi/simple MySQL-python
```

```
pip install -i http://mirrors.aliyun.com/pypi/simple privacyidea
```

如果requirements.txt出问题或卡住，可尝试从建立python虚拟环境开始重新操作。

### 15.2.3. 配置privacyIDEA

新建目录：

## 15.2. privacyIDEA

```
mkdir /etc/privacyidea
```

```
mkdir /var/log/privacyidea
```

编辑 `/etc/privacyidea/pi.cfg` 文件：

```
# The realm, where users are allowed to login as administrators  
# 定义管理员realm  
SUPERUSER_REALM = ['super', 'administrators']  
# Your database 数据链接信息, 根据上面创建的情况来设置  
SQLALCHEMY_DATABASE_URI = 'mysql://privacyidea:unknown@localhost/  
privacyidea'  
# This is used to encrypt the auth_token 自定义密钥, 用于加密认证令牌  
SECRET_KEY = 't0p s3cr3t'  
# This is used to encrypt the admin passwords 用于加密管理员密码  
PI_PEPPER = "Never know..."  
# This is used to encrypt the token data and token passwords 用于  
# 加密令牌数据和令牌密码  
PI_ENCFILE = '/etc/privacyidea/enckey'  
# This is used to sign the audit log 用于签名审计日志  
PI_AUDIT_KEY_PRIVATE = '/etc/privacyidea/private.pem'  
PI_AUDIT_KEY_PUBLIC = '/etc/privacyidea/public.pem'  
# 指定日志文件  
PI_LOGFILE = '/var/log/privacyidea/privacyidea.log'  
# 指定log等级, CRITICAL:50, ERROR: 40, WARNING: 30, INFO: 20, DEBUG:10  
# , 调试阶段可以设置为20, 正式使用可设置到30或以上, 修改后需重启httpd服务以生效  
PI_LOGLEVEL = 20  
#PI_INIT_CHECK_HOOK = 'your.module.function'
```

在 `/etc/privacyidea` 目录下创建 `enckey` (权限自动设为“400 -r-----”仅拥有者能够读取，其他任何人不能进行任何操作)：

```
pi-manage create_enckey
```

在 `/etc/privacyidea` 目录下创建 `private.pem` 和 `public.pem` (权限自动设为 400)：

```
pi-manage create_audit_keys
```

在“`pi.cfg`”指定的数据库中建立 `privacyIDEA` 所需的所有表 (如果出现问题测试一下数据库的连接)：

## 15.2. privacyIDEA

```
pi-manage createdb
```

添加管理员用户 admin，email是admin@x.com：

```
pi-manage admin add admin -e admin@x.com
```

Password: # 输入密码

Confirm: # 确认密码

更改管理员 admin 密码：

```
pi-manage admin change -p admin
```

若要删除管理员 admin，运行：

```
pi-manage admin delete admin
```

测试运行：

```
systemctl stop httpd
```

pi-manage runserver -h 0.0.0.0 -p 80 # -h 后为IP或主机名，如果在其他电脑的浏览器进入，需要设置对方能访问的地址， -p 后为使用的端口

测试完后CTRL + C结束测试环境，运行 systemctl start httpd 启动httpd服务。

创建系统用户：

```
useradd -r -m privacyidea -d /opt/privacyIDEA # -r 创建系统用户， -m 创建用户的主目录， -d 指定主目录为 /opt/privacyIDEA
```

修复权限（更改所有者， -R 递归处理指定目录和所有的子目录及子文件，用户“privacyidea”，组“root”）：

```
chown -R privacyidea:root /etc/privacyidea
```

```
chown -R privacyidea:root /var/log/privacyidea
```

据说下面命令中的文件有源码bug，解决办法：

将<https://github.com/privacyidea/privacyidea/blob/master/tools/privacyidea-fix-access-rights> 内容覆盖原 /opt/privacyIDEA/bin/privacyidea-fix-access-rights 内容（据观察是将第一行 #!/opt/privacyIDEA/bin/python 改成了 #!/usr/bin/python，其他没有变化）：

## 15.2. privacyIDEA

```
/opt/privacyIDEA/bin/privacyidea-fix-access-rights -f  
/etc/privacyidea/pi.cfg -u privacyidea  
  
chmod 400 /etc/privacyidea/enckey  
  
chmod 400 /etc/privacyidea/*.pem
```

配置httpd（httpd的基础配置可以参考[这里](#)）：

```
mkdir -p /var/run/wsgi # 创建目录（ -p 连续创建目录和子目录）  
  
cp /opt/privacyIDEA/etc/privacyidea/privacyideaapp.wsgi  
/etc/privacyidea # 复制默认的wsgi脚本  
  
mv /etc/httpd/conf.d/welcome.conf  
/etc/httpd/conf.d/welcome.conf.disabled # 移除httpd的默认配置文件
```

编辑 /etc/httpd/conf/httpd.conf 文件：

```
# 取消该行注释并设置为服务器主机名或IP地址  
ServerName Hostname-or-IP:80
```

编辑 /etc/httpd/conf.d/privacyidea.conf 文件（访问http80会跳转到https443，部分内容在清楚含义的情况下，可根据自己需要修改）：

```
TraceEnable off
ServerSignature Off
ServerTokens Prod
WSGIPythonHome /opt/privacyIDEA
WSGISocketPrefix /var/run/wsgi
<VirtualHost      80>
    ServerAdmin webmaster@localhost
    ServerName localhost
    RewriteEngine On
    RewriteCond %{HTTPS} !=On
    RewriteRule (.*) https:// %{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
<VirtualHost      443>
    ServerAdmin webmaster@localhost
    ServerName localhost
    DocumentRoot /var/www
    <Directory />
        Require all granted
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    # The daemon is running as user 'privacyidea'
    # This user should have access to the encKey database encryption
    # file
    WSGIDaemonProcess privacyidea python-path=/etc/privacyidea:/opt/
    privacyIDEA/lib/python2.7/site-packages processes=1 threads=15 d
    isplay-name=%{GROUP} user=privacyidea
    WSGIProcessGroup privacyidea
    WSGIPassAuthorization On
    WSGIScriptAlias / /etc/privacyidea/privacyideaapp.wsgi
    SSLEngine On
    SSLProtocol All -SSLv2 -SSLv3
    SSLHonorCipherOrder On
    SSLCipherSuite ECDH+AES256:DHE+AES256:ECDH+AES:EDH+AES:-SHA1:E
    CDH+RC4:EDH+RC4:RC4-SHA:AES256-SHA:!aNULL:!eNULL:!EXP:!LOW:!MD5
    SSLCertificateFile /etc/pki/tls/certs/privacyideaserver.pem
    SSLCertificateKeyFile /etc/pki/tls/private/privacyideaserver.key
</VirtualHost>
```

创

建 `/etc/pki/tls/certs/privacyideaserver.pem` 和 `/etc/pki/tls/private/privacyideaserver.key` 文件：

```
/opt/privacyIDEA/bin/privacyidea-create-certificate -f  
/etc/httpd/conf.d/privacyidea.conf
```

检查httpd配置文件：

```
apachectl configtest
```

返回：`Syntax OK`，表示正常。

```
systemctl restart httpd
```

httpd特殊配置：

如果需要配置虚拟主机（指定域名和非标准的端口号，如使用：`https://pi.x.com:4443`），按以下方法：

修改https端口，编辑 `/etc/httpd/conf.d/ssl.conf` 文件（如果不更改默认端口不需要在这里修改）：

```
Listen 4443 https
```

编辑 `/etc/httpd/conf/httpd.conf` 文件：

```
# 到最后，在“IncludeOptional conf.d/*.conf”上方加入以下内容
# 将所有默认访问转到一个不存在的目录(示例为/var/www/tmp)
<VirtualHost 80>
    ServerName localhost
    DocumentRoot /var/www/tmp
</VirtualHost>
<Directory />
    Require all granted
    Options FollowSymLinks
    AllowOverride None
</Directory>
<VirtualHost 4443>
    ServerName localhost
    DocumentRoot /var/www/tmp
<Directory />
    Require all granted
    Options FollowSymLinks
    AllowOverride None
</Directory>
    SSLEngine On
    SSLProtocol All -SSLv2 -SSLv3
    SSLHonorCipherOrder On
    SSLCipherSuite ECDH+AES256:DHE+AES256:ECDH+AES:EDH+AES:-SHA1:ECDH+RC4:EDH+RC4:RC4-SHA:AES256-SHA:!aNULL:!eNULL:!EXP:!LOW:!MD5
    SSLCertificateFile /etc/pki/tls/certs/privacyideaserver.pem
    SSLCertificateKeyFile /etc/pki/tls/private/privacyideaserver.key
</VirtualHost>
```

编辑 /etc/httpd/conf.d/privacyidea.conf 文件：

```
TraceEnable off
ServerSignature Off
ServerTokens Prod
WSGIPythonHome /opt/privacyIDEA
WSGISocketPrefix /var/run/wsgi
<VirtualHost 80>
    ServerAdmin webmaster@localhost
    ServerName localhost
    ServerAlias pi.x.com
    RewriteEngine On
```

```

RewriteCond %{HTTPS} !=On
RewriteRule (.*) https:// %{SERVER_NAME}:443%{REQUEST_URI} [R=301,L]
# 如果是默认端口，按原教程内容填写即可
</VirtualHost>
<VirtualHost 4443>
ServerAdmin webmaster@localhost
ServerName localhost
ServerAlias pi.x.com
DocumentRoot /var/www
<Directory />
Require all granted
Options FollowSymLinks
AllowOverride None
</Directory>
# The daemon is running as user 'privacyidea'
# This user should have access to the encKey database encryption
file
WSGIDaemonProcess privacyidea python-path=/etc/privacyidea:/opt/
privacyIDEA/lib/python2.7/site-packages processes=1 threads=15 d
isplay-name=%{GROUP} user=privacyidea
WSGIProcessGroup privacyidea
WSGIPassAuthorization On
WSGIScriptAlias / /etc/privacyidea/privacyideaapp.wsgi
SSLEngine On
SSLProtocol All -SSLv2 -SSLv3
SSLHonorCipherOrder On
SSLCipherSuite ECDH+AES256:DHE+AES256:ECDH+AES:EDH+AES:-SHA1:E
CDH+RC4:EDH+RC4:RC4-SHA:AES256-SHA:!aNULL:!eNULL:!EXP:!LOW:!MD5
SSLCertificateFile /etc/pki/tls/certs/privacyideaserver.pem
SSLCertificateKeyFile /etc/pki/tls/private/privacyideaserver.key
</VirtualHost>

```

如果做了以上配置，还需修

改 `/etc/privacyidea/r1m_perl.ini` 和 `/etc/raddb/mods-
config/perl/privacyidea_radius.pm` 中对应的内容（域名和端口）：

## 15.2.4. 配置FreeRADIUS3

复制 `privacyidea_radius.pm` 脚本：

```
cp /opt/privacyIDEA/lib64/privacyidea/authmodules/FreeRADIUS/privacyidea_radius.pm \
/etc/raddb/mods-config/perl/
```

编辑 `/etc/raddb/mods-available/perl`，把filename改为刚才拷过来的 `privacyidea_radius.pm`：

```
# 把“filename”改为刚才拷过来的“privacyidea_radius.pm”
filename = ${modconfdir}/.${:instance}/privacyidea_radius.pm
```

建立软链接，激活perl脚本：

```
ln -s /etc/raddb/mods-available/perl /etc/raddb/mods-enabled/
```

编辑 `/etc/raddb/clients.conf` 文件：

```
# 在最后添加以下内容
client Radius-Client {
    # 定义允许的局域网网段
    ipaddr = 192.168.0.0/24
    # freeradius共享密钥，客户端使用时需一致
    secret = testing123
    require_message_authenticator = no
    nas_type = other
}
```

编辑 `/etc/raddb/sites-available/privacyidea` 文件：

```
server default {
    listen {
        type = auth
        ipaddr = *
        port = 0
        limit {
            max_connections = 16
            lifetime = 0
            idle_timeout = 30
        }
    }
}
```

```
listen {
    ipaddr = *
    port = 0
    type = acct
    limit {
    }
}
authorize {
    preprocess
    digest
    suffix
    ntdomain
    files
    expiration
    logintime
    pap
    update control {
        Auth-Type := Perl
    }
}
authenticate {
    Auth-Type Perl {
        perl
    }
    digest
}
preacct {
    suffix
    files
}
accounting {
    detail
}
session {
}
post-auth {
}
pre-proxy {
}
post-proxy {
}
}
```

建立软链接，激活上面新建的脚本：

```
ln -s /etc/raddb/sites-available/privacyidea /etc/raddb/sites-enabled/
```

删除默认激活的脚本：

```
rm /etc/raddb/sites-enabled/default
```

```
rm /etc/raddb/sites-enabled/inner-tunnel
```

编辑 `/etc/privacyidea/rlm_perl.ini` 文件：

```
# 内容根据需要定义
[Default]
URL = https://127.0.0.1/validate/check
#REALM = someRealm
#RESCONF = someResolver
SSL_CHECK = false
#DEBUG = true
```

编辑 `/etc/raddb/mods-config/perl/privacyidea_radius.pm` 文件，：

```
# “our $CONFIG_FILE”部分改为以下内容
our $CONFIG_FILE = "/etc/privacyidea/rlm_perl.ini";
```

备份原`dictionary`文件：

```
mv /etc/raddb/dictionary /etc/raddb/dictionary.backup
```

复制`dictionary`文件：

```
cp /opt/privacyIDEA/etc/privacyidea/dictionary /etc/raddb/
```

修复privacyIDEA和FreeRADIUS权限：

```
chown -R privacyidea:root /etc/privacyidea
```

```
chgrp -R radiusd /etc/raddb
```

```
cd /etc/raddb
```

```
ll -Z
```

```
restorecon /etc/raddb/*
```

```
reboot
```

添加、测试令牌：

在浏览器访问 `https://(服务器的主机名或IP地址)`，使用之前建立的管理员用户登陆“admin:你的密码”。

如果系统没有找到已定义的realm，会弹出提示默认从 `/etc/passwd` 读取用户来建立用户表并创建realm“测试时可以这样创”建。

进入系统后按以下操作：

Tokens -> Enroll Token -> HOTP（默认） -> 在Username输入root（其他暂时不管） -> Enroll Token

出来二维码，使用Google身份验证器或者FreeOTP扫描二维码，添加成功后：

Tokens -> All tokens -> 点击刚才生成令牌的序列号

在“Test token”前的输入框输入手机生成的6位数字，点“Test token”，提示 `Successfully authenticated.` 表示验证成功。

测试令牌在FreeRADIUS是否成功：

开启调试模式：

```
systemctl stop radiusd
```

```
radiusd -X
```

测试令牌：

```
radtest root ***** 127.0.0.1 0 testing123 # “root”为测试令牌的用户，***** 为手机生成的6位数字，“testing123”为上面配置的共享密钥
```

返回：`Reply-Message = 'privacyIDEA access granted'` 表示认证成功。

测试完成后按Ctrl + C结束，运行 `systemctl start radiusd` 启动RADIUS服务器。

### 15.2.5. 其他配置

笔者注：以下备份、恢复、升级等操作没有具体测试，生产环境操作需要谨慎，最好按照官方文档相应内容部分来操作。

备份（生成的备份文件存储在 `/var/lib/privacyidea/backup/` 目录下）：

```
source /opt/privacyIDEA/bin/activate  
pi-manage backup create
```

恢复：

恢复文件：

```
source /opt/privacyIDEA/bin/activate  
pi-manage backup restore backup_file # “backup_file”为之前备份的压缩包
```

数据库备份在压缩包文件的 `/var/lib/privacyidea/backup/` 内，如果上面的命令没有恢复数据库，运行：

```
mysql -uprivacyidea -p privacyidea < dbdump-xxxx-xxxx.sql
```

升级privacyIDEA：

```
source /opt/privacyIDEA/bin/activate  
pip install -i http://mirrors.aliyun.com/pypi/simple --upgrade  
cffi  
pip install -i http://mirrors.aliyun.com/pypi/simple --upgrade  
bcrypt  
pip install -i http://mirrors.aliyun.com/pypi/simple --upgrade  
privacyidea
```

# 附0. 一些系统配置

- 附0.1. 本地化设置
  - 附0.1.1. 设置主机名
  - 附0.1.2. 设置系统语言
  - 附0.1.3. 设置键盘映射
  - 附0.1.4. 设置时区
- 附0.2. 密码相关设置
  - 附0.2.1. 设置密码规则
  - 附0.2.2. 尝试访问次数
  - 附0.2.3. 重设root密码
- 附0.3. 磁盘相关设置
  - 附0.3.1. 添加硬盘
  - 附0.3.2. 显示硬盘信息
  - 附0.3.3. 设置磁盘配额
    - 附0.3.3.1. XFS
    - 附0.3.3.2. ext4
  - 附0.3.4. 使用SSHFS挂载
  - 附0.3.5. 配置RAID 1
  - 附0.3.6. 逻辑卷管理
    - 附0.3.6.1. 管理物理卷
    - 附0.3.6.2. 管理卷组
    - 附0.3.6.3. 管理逻辑卷
    - 附0.3.6.4. 创建镜像卷
    - 附0.3.6.5. 创建条带卷
- 附0.4. 显示硬件信息
- 附0.5. 分布式文件系统
- 附0.6. 更改运行级别

## 附0.1. 本地化设置

### 附0.1.1. 设置主机名

演示如何设置系统主机名（Hostname）。

更改主机名（如果重新启动系统，会恢复）：

```
hostname # 显示当前主机名
```

```
localhost.localdomain
```

```
hostname dlp.srv.world # 更改主机名
```

```
hostname
```

```
dlp.srv.world # 已更改
```

永久更改主机名：

```
hostnamectl set-hostname dlp.srv.world
```

```
hostnamectl # 显示状态
```

```
Static hostname: dlp.srv.world
Icon name: computer-vm
Chassis: vm
Machine ID: 98a49a78fc9ad91f1b99304c75b94c31
Boot ID: 09b95ce0bc7f4179b1e8a011ed314c6b
Virtualization: kvm
Operating System: CentOS Linux 7 (Core)
CPE OS Name: cpe:/o:centos:centos:7
Kernel: Linux 3.10.0-123.4.2.el7.x86_64
Architecture: x86_64
```

### 附0.1.2. 设置系统语言

设置系统语言，在下例中替换为自己的语言。

```
localectl # 显示当前状态
```

```
System Locale: LANG=en_US.UTF-8
VC Keymap: us
X11 Layout: us
```

```
localectl list-locales # 显示区域设置列表
```

```
aa_DJ
aa_DJ.iso88591
aa_DJ.utf8
aa_ER
aa_ER.utf8
aa_ER.utf8@saaho
aa_ER@saaho
...
...
zh_CN
zh_CN.gb18030
zh_CN.gb2312
zh_CN.gbk
zh_CN.utf8
...
...
zh_TW
zh_TW.big5
zh_TW.euctw
zh_TW.utf8
zu_ZA
zu_ZA.iso88591
zu_ZA.utf8
```

```
localectl set-locale LANG=zh_CN.UTF-8 # 设置
```

```
localectl
```

```
System Locale: LANG=zh_CN.UTF-8  
VC Keymap: us  
X11 Layout: us
```

### 附0.1.3. 设置键盘映射

设置系统的键盘（Keymap）映射，在下例中替换为自己的语言。

```
localectl # 显示当前状态
```

```
System Locale: LANG=zh_CN.UTF-8  
VC Keymap: us  
X11 Layout: us
```

```
localectl list-keymaps
```

```
ANSI-dvorak  
amiga-de  
amiga-us  
applkey  
atari-de  
atari-se  
atari-uk-falcon  
...  
...  
cn  
...  
...  
uk  
unicode  
us  
us-acentos  
wangbe  
wangbe2  
windowkeys
```

```
localectl set-keymap cn # 设置
```

```
localectl
```

```
System Locale: LANG=zh_CN.UTF-8  
VC Keymap: cn  
X11 Layout: cn
```

## 附0.1.4. 设置时区

设置系统的时区（Timezone），在下例中替换为自己的时区。

```
timedatectl list-timezones # 显示时区列表
```

```
Asia/Aden  
Asia/Almaty  
Asia/Amman  
Asia/Anadyr  
Asia/Aqtau  
Asia/Aqtobe  
Asia/Ashgabat  
...  
...  
Asia/Shanghai  
...  
...  
Pacific/Rarotonga  
Pacific/Saipan  
Pacific/Tahiti  
Pacific/Tarawa  
Pacific/Tongatapu  
Pacific/Wake  
Pacific/Wallis
```

```
timedatectl set-timezone Asia/Shanghai # 设置
```

```
timedatectl
```

```
Local time: Wed 2014-07-09 18:31:16 JST
Universal time: Wed 2014-07-09 09:31:16 UTC
RTC time: Wed 2014-07-09 09:31:15
Time zone: Asia/Shanghai (CST, +0800)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
```

## 附0.2. 密码相关设置

### 附0.2.1. 设置密码规则

出于安全考虑，设置密码规则。

设置密码最大有效期（用户必须在指定天数内更改密码。此设置仅影响新创建用户，如果要为现有用户设置，需运行命令：`chage -M (days) (user)`）：

编辑 `/etc/login.defs` 文件：

```
# 密码最大有效期设置为60 (天)
PASS_MAX_DAYS 60
```

设置密码最小有效期（两次修改密码的最小间隔时间。此设置仅影响新创建用户，如果要为现有用户设置，需运行命令：`chage -m (days) (user)`）：

编辑 `/etc/login.defs` 文件：

```
# 密码最小有效期设置为2 (天)
PASS_MIN_DAYS 2
```

设置到期前的警告天数（此设置仅影响新创建用户，如果要为现有用户设置，需运行命令：`chage -w (days) (user)`）：

编辑 `/etc/login.defs` 文件：

```
# 设置警告天数为7 (天)
PASS_WARN_AGE 7
```

限制使用过去使用过的密码：

编辑 `/etc/pam.d/system-auth` 文件：

```
# 禁止使用与过去5代相同的密码 (remember=5)
password      sufficient      pam_unix.so sha512 shadow nullok try
_first_pass  use_authtok remember=5
```

设置密码最小长度（用户无法设置长度小于此参数的密码）：

```
authconfig --passminlen=8 --update # 设置密码最小长度为8
grep "^minlen" /etc/security/pwquality.conf # 该参数在此配置中设置
minlen = 8
```

设置新密码所需的最小字符种类（种类有：大写字母/小写字母/数字/其他）

```
authconfig --passminclass=2 --update # 设置新密码所需的最小字符种类为2
grep "^minclass" /etc/security/pwquality.conf
minclass = 2
```

设置新密码允许连续相同字符的最大数量：

```
authconfig --passmaxrepeat=2 --update # 设置允许连续相同字符的最大数量为2
grep "^maxrepeat" /etc/security/pwquality.conf
maxrepeat = 2
```

设置新密码允许连续相同字符种类的最大数量：

```
authconfig --passmaxclassrepeat=4 --update # 设置允许连续相同字符种类的最大数量为4
grep "^maxclassrepeat" /etc/security/pwquality.conf
maxclassrepeat = 4
```

设置新密码中至少需要一个小写字符：

```
authconfig --enablereqlower --update
```

```
grep "^lcredit" /etc/security/pwquality.conf # 如果要编辑该值，使用vi  
或其它编辑器进行编辑
```

```
lcredit = -1
```

设置新密码中至少需要一个大写字符：

```
authconfig --enablerequpper --update
```

```
grep "^ucredit" /etc/security/pwquality.conf # 如果要编辑该值，使用vi  
或其它编辑器进行编辑
```

```
ucredit = -1
```

设置新密码中至少需要一位数字：

```
authconfig --enablereqdigit --update
```

```
grep "^dcredit" /etc/security/pwquality.conf # 如果要编辑该值，使用vi  
或其它编辑器进行编辑
```

```
dcredit = -1
```

设置新密码中至少需要一个其他字符：

```
authconfig --enablereqother --update
```

```
grep "^ocredit" /etc/security/pwquality.conf # 如果要编辑该值，使用vi  
或其它编辑器进行编辑
```

```
ocredit = -1
```

设置新密码中单调字符序列的最大长度（如：“12345”，“fedcb”等）：

编辑 `/etc/security/pwquality.conf` 文件：

```
# 添加到最后  
maxsequence = 3
```

设置新密码中不能存在的旧密码中的字符个数：

编辑 `/etc/security/pwquality.conf` 文件：

```
# 添加到最后  
difok = 5
```

检查新密码中是否包含超过三个来自用户密码条目的**GECOS**（用户的详细信息，如姓名，年龄，电话等）字段的字符：

编辑 `/etc/security/pwquality.conf` 文件：

```
# 添加到最后  
gecoscheck = 1
```

设置新密码中不能包含在**Ssace**分隔列表中的单词：

编辑 `/etc/security/pwquality.conf` 文件：

```
# 添加到最后  
badwords = denywords1 denywords2 denywords3
```

为新密码设置哈希`I`加密算法（默认为**sha512**）：

```
authconfig --test | grep hashing # 显示当前算法
```

```
password hashing algorithm is md5
```

```
authconfig --passalgo=sha512 --update # 更改算法为sha512
```

```
authconfig --test | grep hashing
```

```
password hashing algorithm is sha512
```

## 附0.2.2. 尝试访问次数

如果对用户的尝试访问次数超过该值，则用户帐户将被锁定。

编辑 `/etc/pam.d/system-auth` 文件：

```
# 添加如下内容
# “deny=N”表示超过N次用户帐户将被锁定（“root”用户未应用）
# 如果要适用于“root”，需添加“even_deny_root”
# “unlock_time=N”表示N秒后锁定的帐户将被解锁（如果没有指定此值，锁定的帐户将不会自动解锁）
# 如果指定“even_deny_root”，可以使用“root_unlock_time=N”指定“root”解锁的时间

auth      required      pam_env.so
auth      required      pam_tally2.so deny=5 unlock_time=60
auth      sufficient   pam_unix.so nullok try_first_pass
auth      requisite    pam_succeed_if.so uid >= 1000 quiet_su
ccess
auth      required      pam_deny.so

account  required      pam_unix.so
account  required      pam_tally2.so
account  sufficient   pam_localuser.so
account  sufficient   pam_succeed_if.so uid < 1000 quiet
account  required      pam_permit.so
```

编辑 `/etc/pam.d/password-auth` 文件：

```
# 添加如下内容
auth      required      pam_env.so
auth      required      pam_tally2.so deny=5 unlock_time=60
auth      sufficient   pam_unix.so nullok try_first_pass
auth      requisite     pam_succeed_if.so uid >= 1000 quiet_su
ccess
auth      required      pam_deny.so

account  required      pam_unix.so
account  required      pam_tally2.so
account  sufficient   pam_localuser.so
account  sufficient   pam_succeed_if.so uid < 1000 quiet
account  required      pam_permit.so
```

如下所示查看尝试访问的次数或手动解锁锁定的帐户：

```
pam_tally2 -u cent # 显示尝试访问的次数
```

| Login | Failures | Latest failure    | From  |
|-------|----------|-------------------|-------|
| cent  | 6        | 07/23/15 19:24:01 | ttyS0 |

```
pam_tally2 -r -u cent # 手动解锁锁定的帐户
```

### 附0.2.3. 重设root密码

如果忘记了root密码，可以如下重新设置。

重新启动系统，当显示GRUB2启动菜单时，按“e”键，然后按照以下步骤重置root密码：

```
# 按“e”键
CentOS Linux (3.10.0-327.4.4.el7.x86_64) 7 (Core) with deb
ugging
CentOS Linux (3.10.0-327.4.4.el7.x86_64) 7 (Core)
CentOS Linux (3.10.0-327.3.1.el7.x86_64) 7 (Core) with deb
ugging
CentOS Linux (3.10.0-327.3.1.el7.x86_64) 7 (Core)
CentOS Linux 7 (Core), with Linux 3.10.0-229.el7.x86_64
CentOS Linux 7 (Core), with Linux 0-rescue-ffa496be96ad482
```

```
cb94373394cec7
```

```
        Use the ^ and v keys to change the selection.  
        Press 'e' to edit the selected item, or 'c' for a command  
prompt.  
  
setparams 'CentOS Linux (3.10.0-327.4.4.el7.x86_64) 7 (Core)' 'f  
edora'  
  
    load_video  
    set gfxpayload=keep  
    insmod gzio  
    insmod part_msdos  
    insmod xfs  
    set root='hd0,msdos1'  
    if [ $feature_platform_search_hint = xy ]; then  
        search --no-floppy --fs-uuid --set=root --hint='hd0,ms  
dos1' c4df086e-3699-4e02-b7cf-b47e614f6920  
    else  
        search --no-floppy --fs-uuid --set=root c4df086e-3699-  
4e02-b7cf-b47e614f6920  
    fi  
    # 添加“rw init=/bin/bash”到下面一行的最后，全部删除“rhgb”，“q  
uiet”，“LANG=***”（如果这一行有）  
    linux16 /vmlinuz-3.10.0-327.4.4.el7.x86_64 root=/dev/map  
per/centos-root \  
            ro rd.lvm.lv=centos/root rd.lvm.lv=centos/swap conso  
le=ttyS0,115200n8  
            systemd.debug rw init=/bin/bash  
    initrd16 /initramfs-3.10.0-327.4.4.el7.x86_64.img  
  
    Press Ctrl-x to start, Ctrl-c for a command prompt or Esc  
ape to  
    discard edits and return to the menu. Pressing Tab lists  
possible completions.  
  
# 输入以上后，按“Ctrl+x”键继续  
....  
....  
[ OK ] Stopped udev Coldplug all Devices.  
Stopping udev Coldplug all Devices...
```

```
[ OK ] Started Plymouth switch root service.  
[ 6.814528] systemd-journald[95]: Received SIGTERM from PID 1  
(systemd).  
bash-4.2#  
  
# 在initramfs switch_root prompt显示后如下操作  
# 在下次引导时设置SELinux重新（如果启用了SELinux）  
bash-4.2# touch /.autorelabel  
  
# 设置密码  
bash-4.2# passwd  
Changing password for user root.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
  
# 重新启动系统完成重设  
bash-4.2# exec /sbin/init
```

## 附0.3. 磁盘相关设置

### 附0.3.1. 添加硬盘

这是添加新硬盘时创建分区的示例。

```
fdisk /dev/sdb # 进入分区操作模式
```

```
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Command (m for help): p # 显示分区表

Disk /dev/vdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xd97d5b18

      Device Boot      Start         End      Blocks   Id  System
# 没有

Command (m for help): n # 创建分区
Partition type:
  p    primary (0 primary, 0 extended, 4 free)
  e    extended
Select (default p): p # 主分区
Partition number (1-4, default 1): 1 # 指定分区号
First sector (2048-41943039, default 2048): # 起始扇区
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41
943039): # 结束扇区
Using default value 41943039
Partition 1 of type Linux and of size 20 GiB is set

Command (m for help): p # 显示分区表
```

```
Disk /dev/vdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xd97d5b18
```

| Device    | Boot | Start | End      | Blocks   | Id | System |
|-----------|------|-------|----------|----------|----|--------|
| /dev/vdb1 |      | 2048  | 41943039 | 20970496 | 83 | Linux  |

# 已创建

Command (m for help): t # 更改类型，如果不适应逻辑卷，不用更改

Selected partition 1

| Hex code (type L to list all codes): L # 显示列表 |                 |    |                 |    |                    |         |
|-----------------------------------------------|-----------------|----|-----------------|----|--------------------|---------|
| 0                                             | Empty           | 24 | NEC DOS         | 81 | Minix / old Lin bf | Solaris |
| 1                                             | FAT12           | 27 | Hidden NTFS Win | 82 | Linux swap / So    | c1      |
| DRDOS/sec (FAT-                               |                 |    |                 |    |                    |         |
| 2                                             | XENIX root      | 39 | Plan 9          | 83 | Linux              | c4      |
| DRDOS/sec (FAT-                               |                 |    |                 |    |                    |         |
| 3                                             | XENIX usr       | 3c | PartitionMagic  | 84 | OS/2 hidden C:     | c6      |
| DRDOS/sec (FAT-                               |                 |    |                 |    |                    |         |
| 4                                             | FAT16 <32M      | 40 | Venix 80286     | 85 | Linux extended     | c7      |
| Syrinx                                        |                 |    |                 |    |                    |         |
| 5                                             | Extended        | 41 | PPC PReP Boot   | 86 | NTFS volume set    | da      |
| Non-FS data                                   |                 |    |                 |    |                    |         |
| 6                                             | FAT16           | 42 | SFS             | 87 | NTFS volume set    | db      |
| CP/M / CTOS / .                               |                 |    |                 |    |                    |         |
| 7                                             | HPFS/NTFS/exFAT | 4d | QNX4.x          | 88 | Linux plaintext    | de      |
| Dell Utility                                  |                 |    |                 |    |                    |         |
| 8                                             | AIX             | 4e | QNX4.x 2nd part | 8e | Linux LVM          | df      |
| BootIt                                        |                 |    |                 |    |                    |         |
| 9                                             | AIX bootable    | 4f | QNX4.x 3rd part | 93 | Amoeba             | e1      |
| DOS access                                    |                 |    |                 |    |                    |         |
| a                                             | OS/2 Boot Manag | 50 | OnTrack DM      | 94 | Amoeba BBT         | e3      |
| DOS R/O                                       |                 |    |                 |    |                    |         |
| b                                             | W95 FAT32       | 51 | OnTrack DM6 Aux | 9f | BSD/OS             | e4      |
| SpeedStor                                     |                 |    |                 |    |                    |         |
| c                                             | W95 FAT32 (LBA) | 52 | CP/M            | a0 | IBM Thinkpad hi    | eb      |
| BeOS fs                                       |                 |    |                 |    |                    |         |
| e                                             | W95 FAT16 (LBA) | 53 | OnTrack DM6 Aux | a5 | FreeBSD            | ee      |

## GPT

```

f  W95 Ext'd (LBA) 54  OnTrackDM6      a6  OpenBSD          ef
EFI (FAT-12/16/
10  OPUS            55  EZ-Drive        a7  NeXTSTEP         f0
Linux/PA-RISC b
11  Hidden FAT12    56  Golden Bow     a8  Darwin UFS       f1
SpeedStor
12  Compaq diagnost 5c  Priam Edisk     a9  NetBSD           f4
SpeedStor
14  Hidden FAT16 <3 61  SpeedStor      ab  Darwin boot     f2
DOS secondary
16  Hidden FAT16    63  GNU HURD or Sys af  HFS / HFS+      fb
VMware VMFS
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs          fc
VMware VMKCORE
18  AST SmartSleep   65  Novell Netware  b8  BSDI swap        fd
Linux raid auto
1b  Hidden W95 FAT3  70  DiskSecure Mult bb  Boot Wizard hid fe
LANstep
1c  Hidden W95 FAT3  75  PC/IX          be  Solaris boot     ff
BBT
1e  Hidden W95 FAT1  80  Old Minix
Hex code (type L to list all codes): 8e # 指定Linux LVM
Changed type of partition 'Linux' to 'Linux LVM'

```

Command (m for help): p # 显示分区表

```

Disk /dev/vdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xd97d5b18

```

| Device    | Boot | Start | End      | Blocks   | Id | System   |
|-----------|------|-------|----------|----------|----|----------|
| /dev/vdb1 |      | 2048  | 41943039 | 20970496 | 8e | Linux LV |

M # 已更改

Command (m for help): w # 保存并退出  
The partition table has been altered!

Calling ioctl() to re-read partition table.

```
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.  
The kernel still uses the old table. The new table will be used at  
the next reboot or after you run partprobe(8) or kpartx(8)  
Syncing disks.
```

```
sfdisk -l /dev/sdb # 显示状态
```

```
Disk /dev/vdb: 41610 cylinders, 16 heads, 63 sectors/track  
sfdisk: Warning: The partition table looks like it was made  
for C/H/S=/*3/34 (instead of 41610/16/63).  
For this listing I'll assume that geometry.
```

```
Units: cylinders of 52224 bytes, blocks of 1024 bytes, counting  
from 0
```

| Device    | Boot | Start | End     | #cyls          | #blocks                               | Id | System    |
|-----------|------|-------|---------|----------------|---------------------------------------|----|-----------|
| /dev/vdb1 |      | 20+   | 411206- | 411187-        | 20970496                              | 8e | Linux LVM |
| sfdisk:   |      |       |         | start: (c,h,s) | expected (20,0,9) found (2,0,33)      |    |           |
| sfdisk:   |      |       |         | end: (c,h,s)   | expected (1023,2,34) found (650,2,34) |    |           |
| /dev/vdb2 |      | 0     | -       | 0              | 0                                     | 0  | Empty     |
| /dev/vdb3 |      | 0     | -       | 0              | 0                                     | 0  | Empty     |
| /dev/vdb4 |      | 0     | -       | 0              | 0                                     | 0  | Empty     |

不使用逻辑卷，则格式化分区后挂载（如挂载到 /mnt/storage）即可：

```
mkfs.xfs /dev/sdb1 # 使用xfs文件系统  
mkfs.ext4 /dev/sdb1 # 使用ext4文件系统  
mkdir /mnt/storage  
mount /dev/sdb1 /mnt/storage
```

添加到 /etc/fstab，系统启动时挂载：

```
# 加入以下一行，具体含义可以网上查下资料  
/dev/sdb1 /mnt/storage xfs defaults 0 0
```

### 附0.3.2. 显示硬盘信息

安装hdparm：

```
yum -y install hdparm
```

显示硬盘信息：

```
hdparm -i /dev/sda
```

```
/dev/sda:  
  
Model=TOSHIBA THNSNH128GCST, FwRev=HTRAN101, SerialNo=Y3TS108TT  
PEY  
Config={ Fixed }  
RawCHS=16383/16/63, TrkSize=0, SectSize=0, ECCbytes=0  
BuffType=unknown, BuffSize=unknown, MaxMultSect=16, MultSect=16  
CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=250069  
680  
IORDY=on/off, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}  
}  
PIO modes: pio0 pio3 pio4  
DMA modes: mdma0 mdma1 mdma2  
UDMA modes: udma0 udma1 udma2 udma3 udma4 *udma5  
AdvancedPM=yes: unknown setting WriteCache=enabled  
Drive conforms to: Unspecified: ATA/ATAPI-3,4,5,6,7  
  
* signifies the current active mode
```

```
hdparm -I /dev/sda # 显示细节
```

```
/dev/sda:  
  
ATA device, with non-removable media  
  Model Number:      TOSHIBA THNSNH128GCST  
  Serial Number:     Y3TS108TTPEY  
  Firmware Revision: HTRAN101  
  Transport:         Serial, ATA8-AST, SATA 1.0a, SATA II  
  Extensions, SATA Rev 2.5, SATA Rev 2.6, SATA Rev 3.0  
Standards:  
  Supported: 9 8 7 6 5  
  Likely used: 9  
Configuration:  
  Logical      max    current  
  cylinders    16383   16383  
  heads        16      16  
  sectors/track 63      63  
....  
....
```

显示硬盘的设置：

```
hdparm /dev/sda
```

```
/dev/sda:  
  multcount      = 16 (on)  
  IO_support     = 1 (32-bit)  
  readonly       = 0 (off)  
  readahead      = 256 (on)  
  geometry       = 15566/255/63, sectors = 250069680, start = 0
```

测试硬盘的可读速度：

```
hdparm -Tt /dev/sda
```

```
/dev/sda:  
  Timing cached reads: 21468 MB in 2.00 seconds = 10746.50 MB/sec  
  Timing buffered disk reads: 1458 MB in 3.00 seconds = 485.64 MB/sec
```

### 附0.3.3. 设置磁盘配额

设置磁盘配额以限制磁盘使用量。

#### 附0.3.3.1. XFS

下例演示在**XFS**格式的 `/home` 上设置配额。

添加挂载选项以启用配额：

```
umount /home
```

```
mount -o uquota,gquota /dev/sdb1 /home
```

编辑 `/etc/fstab` 文件：

```
# 添加选项
/dev/mapper/centos-root /
    0 0
UUID=c4df086e-3699-4e02-b7cf /boot xfs defaults
    0 0
/dev/mapper/centos-swap swap
    0 0
/dev/sdb1
    /home      xfs defaults,uquota,gquota
    0 0
```

设置用户配额。例如，将配额应用于用户“cent”：

```
xfs_quota -x /home # 以专家模式运行配额工具
```

```
# 显示当前状态
xfs_quota> state
User quota state on /home (/dev/sdb1)
    Accounting: ON
    Enforcement: ON
    Inode: #136 (2 blocks, 2 extents)
Group quota state on /home (/dev/sdb1)
    Accounting: ON
    Enforcement: ON
    Inode: #137 (2 blocks, 2 extents)
Project quota state on /home (/dev/sdb1)
```

```

Accounting: OFF
Enforcement: OFF
Inode: #137 (2 blocks, 2 extents)
Blocks grace time: [7 days 00:00:30]
Inodes grace time: [7 days 00:00:30]
Realtime Blocks grace time: [7 days 00:00:30]

# 显示使用情况报告
xfs_quota> report -h
User quota on /home (/dev/sdb1)
          Blocks
User ID      Used   Soft   Hard Warn/Grace
-----
root          0     0     0  00 [-----]
cent         16K    0     0  00 [-----]

Group quota on /home (/dev/sdb1)
          Blocks
Group ID     Used   Soft   Hard Warn/Grace
-----
root          0     0     0  00 [-----]
cent         16K    0     0  00 [-----]

# 给用户“cent”设置软限制9G，硬限制10G（使用千字节指定）
xfs_quota> limit bsoft=9g bhard=10g cent

# 显示报告
xfs_quota> report -h -u
User quota on /home (/dev/sdb1)
          Blocks
User ID      Used   Soft   Hard Warn/Grace
-----
root          0     0     0  00 [-----]
cent         16K    9G    10G  00 [-----]

```

如果设置组配额，执行以下操作：

```
xfs_quota -x -c 'limit -g bsoft=9g bhard=10g cent' /home # 可以设置为非交互模式 (non-interactive mode)
```

```
xfs_quota -x -c 'report -h -g' /home
```

```
Group quota on /home (/dev/sdb1)
          Blocks
Group ID      Used   Soft   Hard Warn/Grace
-----
root          0       0       0   00 [-----]
cent         16K     9G    10G  00 [-----]
```

可以使用Warnquota发送警告（需要SMTP服务器）：

```
yum -y install quota-warnquota # 安装Warnquota
```

编辑 /etc/quotatab 文件：

```
# 添加设备和描述配额设置
/dev/sdb1: Your Home Director
```

```
sed -i -e "s/example\.com/server\.world/g" /etc/warnquota.conf # 将
域名更改为自己的域名
```

```
warnquota -s # 运行Warnquota
```

```
# 如果用户在warnquota运行时超出配额，则发送以下警告
From root@dlp.srv.world Thu Oct 20 19:08:08 2015
Return-Path: <root@dlp.srv.world>
X-Original-To: cent
Delivered-To: cent@dlp.srv.world
From: root@srv.world
Reply-To: root@srv.world
Subject: NOTE: You are exceeding your allocated disk space limit
S
To: cent@dlp.srv.world
Cc: root@srv.world
Content-Type: text/plain; charset=UTF-8
Content-Disposition: inline
Date: Thu, 20 Oct 2015 19:08:08 +0900 (JST)
Status: R
```

Your disk usage has exceeded the agreed limits on this server  
Please delete any unnecessary files on following filesystems:

Your Home Directory (/dev/sdb1)

| Filesystem | hard grace | Block limits |       |       |       | File limits |      |
|------------|------------|--------------|-------|-------|-------|-------------|------|
|            |            | used         | soft  | hard  | grace | used        | soft |
| /dev/sdb1  | + -        | 4112M        | 4096M | 5120M | 6days | 6           | 0    |
|            | 0          |              |       |       |       |             |      |

root@srv.world

### 附0.3.3.2. ext4

下例演示在**ext4**格式的 /home 上设置配额。

安装配额工具：

```
yum -y install quota
```

添加挂载选项以启用配额：

```
umount /home
```

```
mount -o usrquota,grpquota /dev/sdb1 /home
```

编辑 `/etc/fstab` 文件：

```
# 添加选项
/dev/mapper/VolGroup-lv_root /           ext4    defaults
1 1
UUID=cf3f9660-e40d-459d-8763 /boot       ext4    defaults
1 2
/dev/mapper/VolGroup-lv_swap swap         swap     defaults
0 0
tmpfs                   /dev/shm   tmpfs   defaults
0 0
devpts                  /dev/pts   devpts  gid=5,mode=620
0 0
sysfs                  /sys      sysfs   defaults
0 0
proc                    /proc     proc    defaults
0 0
/dev/sdb1               /home     xfs    defaults,usrquota
a,grpquota 0 0
```

设置用户配额。例如，将配额应用于用户“cent”：

```
quotacheck -um /home # 创建配额配置
```

```
quotaon -uv /home # 启用配额
```

```
/dev/sdb1 [/home]: user quotas turned on
```

```
quotaon -ap # 显示状态
```

```
group quota on /home (/dev/sdb1) is off
user quota on /home (/dev/sdb1) is on
```

```
edquota -u cent # 为用户“cent”设置配额
```

```
# 设置软限制4G，硬限制5G（使用千字节指定）
Disk quotas for user cent (uid 500):
  Filesystem    blocks      soft      hard      inodes      soft      hard
    /dev/sdb1       16     4096000    5120000          7          0          0
```

```
repquota -au # 显示状态
```

```
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days
              Block limits                  File limits
User           used      soft      hard      grace      used      soft      hard
  cent
-----
root          --        20        0        0            2        0        0
cent          --      16 4096000 5120000            4        0        0
```

如果要将用户的配额设置应用于其他用户，如下进行设置：

```
edquota -p cent fedora # 将“cent”的设置应用于“fedora”
```

```
repquota -au
```

```
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days
              Block limits                  File limits
User           used      soft      hard      grace      used      soft      hard
  cent
-----
root          --        20        0        0            2        0        0
cent          +- 5120000 4096000 5120000 6days            7        0        0
fedora         --        4 4096000 5120000            4        0        0
```

设置组配额。例如，将配额应用于组“cent”：

```
quotacheck -gm /home # 创建配额配置
```

```
quotaon -gv /home # 启用配额
```

```
/dev/sdb1 [/home]: group quotas turned on
```

```
quotaon -ap # 显示状态
```

```
group quota on /home (/dev/sdb1) is on  
user quota on /home (/dev/sdb1) is on
```

```
edquota -g cent # 为“cent”设置配额
```

```
# 设置软限制4G，硬限制5G（使用千字节指定）  
Disk quotas for group cent (gid 500):  
  Filesystem      blocks      soft      hard      inodes      sof  
t  hard  
  /dev/sdb1        5120000    4096000    5120000          7  
  0            0
```

```
repquota -ag # 显示状态
```

```
*** Report for group quotas on device /dev/sdb1  
Block grace time: 7days; Inode grace time: 7days  
                                Block limits                      File limits  
Group           used     soft     hard   grace     used     soft     hard  
  grace  
-----  
-----  
root          --       20       0       0                  2       0       0  
cent          +- 5120000 4096000 5120000 6days      7       0       0  
fedora        --       16       0       0                  4       0       0
```

可以使用Warnquota发送警告（需要[SMTP服务器](#)）：

编辑 `/etc/quotatab` 文件：

```
# 添加设备和描述配额设置  
/dev/sdb1: Your Home Director
```

```
sed -i -e "s/example\.com/server\.world/g" /etc/warnquota.conf # 将  
域名更改为自己的域名
```

```
warnquota -s # 运行Warnquota
```

```
# 如果用户在warnquota运行时超出配额，则发送以下警告  
From root@dlp.srv.world Fri Oct 23 09:38:10 2011  
Return-Path: <root@dlp.srv.world>  
X-Original-To: cent  
Delivered-To: cent@dlp.srv.world  
From: root@srv.world  
Reply-To: root@srv.world  
Subject: NOTE: You are exceeding your allocated disk space limit  
S  
To: cent@dlp.srv.world  
Cc: root@srv.world  
Date: Fri, 23 Oct 2011 09:38:10 +0900 (JST)  
Status: R
```

Your disk usage has exceeded the agreed limits on this server  
Please delete any unnecessary files on following filesystems:

Your Home Director (/dev/sdb1)

| Filesystem<br>hard grace | Block limits |       |       |             | File limits |      |
|--------------------------|--------------|-------|-------|-------------|-------------|------|
|                          | used         | soft  | hard  | grace       | used        | soft |
| /dev/sdb1<br>0           | +-           | 5000M | 4000M | 5000M 6days | 7           | 0    |

root@srv.world

### 附0.3.4. 使用SSHFS挂载

可以使用[SSHFS](#)通过SSH挂载另一台主机的文件系统。

安装fuse-sshfs：

```
yum --enablerepo=epel -y install fuse-sshfs # 从EPEL安装
```

例如，使用用户“cent”挂载另一台主机上的 /home/cent：

```
mkdir ~/sshmnt # 为挂载创建一个目录
```

```
sshfs 10.0.0.31:/home/cent ~/sshmnt # 使用SSHFS挂载
```

```
cent@10.0.0.31's password: # SSH密码
```

```
df -hT
```

| Filesystem              | Type       | Size | Used | Avail | Use% | Mounted on        |
|-------------------------|------------|------|------|-------|------|-------------------|
| /dev/mapper/centos-root | xfs        | 27G  | 1.1G | 26G   | 5%   | /                 |
| devtmpfs                | devtmpfs   | 2.0G | 0    | 2.0G  | 0%   | /dev              |
| tmpfs                   | tmpfs      | 2.0G | 0    | 2.0G  | 0%   | /dev/shm          |
| tmpfs                   | tmpfs      | 2.0G | 8.3M | 2.0G  | 1%   | /run              |
| tmpfs                   | tmpfs      | 2.0G | 0    | 2.0G  | 0%   | /sys/fs/cgroup    |
| /dev/vda1               | xfs        | 497M | 151M | 347M  | 31%  | /boot             |
| 10.0.0.31:/home/cent    | fuse.sshfs | 27G  | 1.1G | 26G   | 4%   | /home/cent/sshmnt |

# 已挂载

```
fusermount -u ~/sshmnt # 卸载
```

### 附0.3.5. 配置RAID 1

在服务器上添加两个新硬盘配置RAID 1。

本例演示新增硬盘“sdb”和“sdc”并配置RAID 1。

```
df -h
```

| Filesystem              | Size | Used | Avail | Use% | Mounted on     |
|-------------------------|------|------|-------|------|----------------|
| /dev/mapper/centos-root | 196G | 1.2G | 195G  | 1%   | /              |
| devtmpfs                | 1.9G | 0    | 1.9G  | 0%   | /dev           |
| tmpfs                   | 1.9G | 0    | 1.9G  | 0%   | /dev/shm       |
| tmpfs                   | 1.9G | 8.5M | 1.9G  | 1%   | /run           |
| tmpfs                   | 1.9G | 0    | 1.9G  | 0%   | /sys/fs/cgroup |
| /dev/sda1               | 497M | 222M | 276M  | 45%  | /boot          |

在新硬盘上创建一个分区并设置RAID标志：

```
parted --script /dev/sdb "mklabel gpt"  
parted --script /dev/sdc "mklabel gpt"  
parted --script /dev/sdb "mkpart primary 0% 100%"  
parted --script /dev/sdc "mkpart primary 0% 100%"  
parted --script /dev/sdb "set 1 raid on"  
parted --script /dev/sdc "set 1 raid on"
```

配置RAID 1：

```
yum -y install mdadm  
  
mdadm --create /dev/md0 --level=raid1 --raid-devices=2 /dev/sdb1  
/dev/sdc1
```

```
mdadm: Note: this array has metadata at the start and  
may not be suitable as a boot device. If you plan to  
store '/boot' on this device please ensure that  
your boot-loader understands md/v1.x metadata, or use  
--metadata=0.90  
Continue creating array? y  
mdadm: Defaulting to version 1.2 metadata  
mdadm: array /dev/md0 started.
```

```
cat /proc/mdstat # 显示状态
```

```
Personalities : [raid1]  
md0 : active raid1 sdc1[1] sdb1[0]  
      83818496 blocks super 1.2 [2/2] [UU]  
      [======>.....]  resync = 31.0% (26044416/83818496  
) finish=4.7min speed=201190K/sec  
  
unused devices: <none>
```

```
cat /proc/mdstat # 几个小时后，如果同步完成，状态如下
```

```
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      83818496 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

编辑 `/etc/sysconfig/raid-check` 文件：

```
# 添加要由Cron检查的RAID设备
CHECK_DEVS="md0"
```

例如，如果RAID阵列中的成员硬盘出现故障，如下重配置RAID 1：

```
cat /proc/mdstat # 失败的状态如下
```

```
Personalities : [raid1]
md0 : active (auto-read-only) raid1 sdb1[0]
      83818496 blocks super 1.2 [2/1] [U_]

unused devices: <none>
```

```
mdadm --manage /dev/md0 --add /dev/sdc1 # 更换新磁盘后，重新配置
```

```
mdadm: added /dev/sdc1
```

```
cat /proc/mdstat
```

```
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      83818496 blocks super 1.2 [2/2] [UU]
      [======>.....]  resync = 31.0% (26044416/83818496
) finish=4.7min speed=201190K/sec

unused devices: <none>
```

## 附0.3.6. 逻辑卷管理

LVM逻辑卷管理（Logical Volume Manager）是红帽联机磁盘存储管理系统中的一个子系统。

这里参考其他简要说明使用流程：

创建分区参考第一节内容。

物理卷（Physical Volume）：

```
pvcreate /dev/sdb1 # 创建物理卷  
pvdisplay /dev/sdb1 # 显示物理卷  
pvs /dev/sdb1 # 显示物理卷报告  
pvscan # 扫描物理卷  
pvremove /dev/sdb1 # 删除物理卷  
pvs
```

创建卷组（Volume Group）（可以使用分区、磁盘、磁盘阵列来创建）：

```
vgcreate vg_dlp /dev/sdb1 # 创建卷组  
vgcreate vg_dlp /dev/sdb1 /dev/sdd1 # 如果要使用多个设备  
vgdisplay vg_dlp # 显示卷组  
vgrename vg_dlp vg_data # 更改卷组名称  
vgdisplay vg_data  
vgs # 显示卷组报告  
vgscan # 扫描卷组  
  
# 扩展卷组  
vgextend vg_data /dev/sdc1 # 添加“sdc1”到“vg_data”  
  
# 削减卷组  
vgreduce vg_data /dev/sdc1 # 从“vg_data”移除“sdc1”  
  
# 删除卷组  
vgchange -a n vg_data # 首先禁用目标卷组  
vgremove vg_data # 删除
```

创建逻辑卷（Logical Volume）：

```
lvcreate -L 50G -n lv_data vg_dlp # 在卷组“vg_dlp”中创建50G的逻辑卷“lv_data”
lvcreate -l 100%FREE -n lv_data vg_dlp # 如果使用所有空闲区域
lvdisplay /dev/vg_dlp/lv_data # 显示逻辑卷
lvrename vg_dlp lv_data lv_storage # 重命名逻辑卷
lvdisplay /dev/vg_dlp/lv_storage
lvs # 显示逻辑卷报告
lvscan # 扫描逻辑卷

# 拍摄逻辑卷的快照
lvcreate -s -L 50G -n snap-lv_storage /dev/vg_dlp/lv_storage #
从“lv_storage”创建快照“snap-lv_storage”
lvdisplay /dev/vg_dlp/lv_storage /dev/vg_dlp/snap-lv_storage

# 扩展逻辑卷（可以在挂载时操作）
lvextend -L 70G /dev/vg_dlp/lv_storage
lvextend -L +20G /dev/vg_dlp/lv_storage # 指定添加的容量
df -hLT # 检查实际容量未增加
xfs_growfs /mnt # 扩展xfs文件系统（指定挂载点）
resize2fs /dev/vg_dlp/lv_storage # 扩展ext4文件系统
df -hLT # 再次查看容量

# 削减逻辑卷（首先卸载目标设备，不能削减xfs文件系统）
e2fsck -f /dev/vg_dlp/lv_storage 50G # ext4，先检查
resize2fs /dev/vg_dlp/lv_storage 50G # ext4，削减文件系统
lvreduce -L 50G /dev/vg_dlp/lv_storage # 削减逻辑卷

# 删除逻辑卷：卸载 -> 停止逻辑卷 -> 删除逻辑卷
lvchange -an /dev/vg_dlp/lv_storage
lvremove /dev/vg_dlp/lv_storage
```

格式化逻辑卷：

```
mkfs.ext4 /dev/vg_dlp/lv_storage
```

挂载逻辑卷到 /mnt/lv\_mnt：

```
mkdir /mnt/lv_mnt
mount /dev/vg_dlp/lv_storage /mnt/lv_mnt
```

更多内容可参看下面介绍。

### 附0.3.6.1. 管理物理卷

这是管理物理卷（Physical Volume）的基本操作。

首先需要[创建LVM类型的分区](#)。

创建物理卷：

```
pvcreate /dev/sdb1
```

```
Physical volume "/dev/sdb1" successfully created
```

如果要指定卷大小，如下操作：

```
pvcreate --setphysicalvolumesize 50G /dev/sdb1
```

```
Physical volume "/dev/sdb1" successfully created
```

显示物理卷：

```
pvdisplay /dev/sdb1
```

|              |                                        |
|--------------|----------------------------------------|
| PV Name      | /dev/sdb1                              |
| VG Name      |                                        |
| PV Size      | 80.00 GiB                              |
| Allocatable  | NO                                     |
| PE Size      | 0                                      |
| Total PE     | 0                                      |
| Free PE      | 0                                      |
| Allocated PE | 0                                      |
| PV UUID      | PJnOPg-9Kw0-5xJz-Z8tn-9zP7-VKn5-ADrGzy |

更改物理卷大小：

```
pvresize --setphysicalvolumesize 50G /dev/sdb1 # 更改为50G
```

```
Physical volume "/dev/sdb1" changed
  1 physical volume(s) resized / 0 physical volume(s) not resize
d
```

pvdisplay /dev/sdb1

|              |                                        |
|--------------|----------------------------------------|
| PV Name      | /dev/sdb1                              |
| VG Name      |                                        |
| PV Size      | 50.00 GiB                              |
| Allocatable  | NO                                     |
| PE Size      | 0                                      |
| Total PE     | 0                                      |
| Free PE      | 0                                      |
| Allocated PE | 0                                      |
| PV UUID      | PJnOPg-9Kw0-5xJz-Z8tn-9zP7-VKn5-ADrGzy |

显示物理卷报告：

pvs /dev/sdb1

| PV        | VG   | Fmt | Attr | PSize  | PFree  |
|-----------|------|-----|------|--------|--------|
| /dev/sdb1 | lvm2 | --- |      | 80.00g | 80.00g |

扫描物理卷：

pvscan

```
PV /dev/vda2   VG centos    lvm2 [49.51 GiB / 0     free]
PV /dev/sdb1      lvm2 [80.00 GiB]
Total: 2 [129.51 GiB] / in use: 1 [49.51 GiB] / in no VG: 1 [8
0.00 GiB]
```

删除物理卷：

pvremove /dev/sdb1

```
Labels on physical volume "/dev/sdb1" successfully wiped
```

```
pvdisplay /dev/sdb1
```

```
Failed to find physical volume "/dev/sdb1".
```

### 附0.3.6.2. 管理卷组

这是管理卷组（Volume Group）的基本操作。

首先需要参照上一节创建物理卷。

创建卷组：

```
vgcreate vg_dlp /dev/sdb1
```

```
Volume group "vg_dlp" successfully created
```

如果要使用多个设备，如下指定：

```
vgcreate vg_dlp /dev/sdb1 /dev/sdd1
```

```
Volume group "vg_dlp" successfully created
```

显示卷组：

```
vgdisplay vg_dlp
```

```
--- Volume group ---
VG Name          vg_dlp
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No 1
VG Access        read/write
VG Status         resizable
MAX LV            0
Cur LV            0
Open LV           0
Max PV            0
Cur PV            1
Act PV            1
VG Size           80.00 GiB
PE Size           4.00 MiB
Total PE          20479
Alloc PE / Size   0 / 0
Free  PE / Size   20479 / 80.00 GiB
VG UUID          Q3c0yC-ZgGf-E0aX-tZsv-wLe8-DETH-XDqPER
```

更改卷组名称：

```
vgrename vg_dlp vg_data
```

```
Volume group "vg_dlp" successfully renamed to "vg_data"
```

```
vgdisplay vg_data
```

```
--- Volume group ---
VG Name          vg_data
System ID
Format          lvm2
Metadata Areas    1
Metadata Sequence No  2
VG Access        read/write
VG Status         resizable
MAX LV           0
Cur LV            0
Open LV           0
Max PV            0
Cur PV            1
Act PV            1
VG Size          80.00 GiB
PE Size          4.00 MiB
Total PE         20479
Alloc PE / Size   0 / 0
Free  PE / Size   20479 / 80.00 GiB
VG UUID          Q3c0yC-ZgGf-E0aX-tZsv-wLe8-DETH-XDqPER
```

显示卷组报告：

```
vgs
```

| VG      | #PV | #LV | #SN | Attr   | VSize  | VFree  |
|---------|-----|-----|-----|--------|--------|--------|
| centos  | 1   | 2   | 0   | wz--n- | 49.51g | 0      |
| vg_data | 1   | 0   | 0   | wz--n- | 80.00g | 80.00g |

扫描卷组：

```
vgscan
```

```
Reading all physical volumes. This may take a while...
Found volume group "centos" using metadata type lvm2
Found volume group "vg_data" using metadata type lvm2
```

扩展卷组：

```
vgextend vg_data /dev/sdc1 # 添加“sdc1”到“vg_data”
```

```
Volume group "vg_data" successfully extended
```

```
vgdisplay vg_data
```

```
--- Volume group ---
VG Name          vg_data
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 4
VG Access        read/write
VG Status         resizable
MAX LV            0
Cur LV            0
Open LV           0
Max PV            0
Cur PV            2
Act PV            2
VG Size          160.00 GiB
PE Size           4.00 MiB
Total PE          20479
Alloc PE / Size  0 / 0
Free  PE / Size  20479 / 160.00 GiB
VG UUID          Q3c0yC-ZgGf-E0aX-tZsv-wLe8-DETH-XDqPER
```

削减卷组：

```
vgreduce vg_data /dev/sdc1 # 从“vg_data”移除“sdc1”
```

```
Removed "/dev/sdd1" from volume group "vg_data"
```

```
vgdisplay vg_data
```

```
--- Volume group ---
VG Name          vg_data
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No  2
VG Access        read/write
VG Status         resizable
MAX LV            0
Cur LV            0
Open LV           0
Max PV            0
Cur PV            1
Act PV            1
VG Size           80.00 GiB
PE Size           4.00 MiB
Total PE          20479
Alloc PE / Size   0 / 0
Free  PE / Size   20479 / 80.00 GiB
VG UUID          Q3c0yC-ZgGf-E0aX-tZsv-wLe8-DETh-XDqPER
```

删除卷组（首先禁用目标卷组并将其删除）：

```
vgchange -a n vg_data
```

```
0 logical volume(s) in volume group "vg_data" now active
```

```
vgremove vg_data
```

```
Volume group "vg_data" successfully removed
```

### 附0.3.6.3. 管理逻辑卷

这是管理逻辑卷（Logical Volume）的基本操作。

首先需要参照上一节创建卷组。

创建逻辑卷：

```
lvcreate -L 50G -n lv_data vg_dlp # 在卷组“vg_dlp”中创建50G的逻辑卷“lv_data”
```

```
Logical volume "lv_data" created
```

如果使用所有空闲区域，如下指定：

```
lvcreate -l 100%FREE -n lv_data vg_dlp
```

```
Logical volume "lv_data" created
```

显示逻辑卷：

```
lvdisplay /dev/vg_dlp/lv_data
```

```
--- Logical volume ---
LV Path          /dev/vg_dlp/lv_data
LV Name          lv_data
VG Name          vg_dlp
LV UUID          n1Tuzs-T9nC-hRIC-4Vgh-Bm9G-t4EI-kTu2NU
LV Write Access  read/write
LV Creation host, time dlp.srv.world, 2015-07-20 09:23:36 +090
0
LV Status        available
# open           0
LV Size          80.00 GiB
Current LE       20479
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 8192
Block device     253:2
```

重命名逻辑卷：

```
lvrename vg_dlp lv_data lv_storage # 从“lv_data”重命名为“lv_storage”
```

```
Renamed "lv_data" to "lv_storage" in volume group "vg_dlp"
```

```
lvdisplay /dev/vg_dlp/lv_storage
```

```
--- Logical volume ---
LV Path          /dev/vg_dlp/lv_storage
LV Name          lv_storage
VG Name          vg_dlp
LV UUID          nlTuzs-T9nC-hRIC-4Vgh-Bm9G-t4EI-kTu2NU
LV Write Access  read/write
LV Creation host, time dlp.srv.world, 2015-07-20 09:23:36 +090
0
LV Status        available
# open           0
LV Size          80.00 GiB
Current LE       20479
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 8192
Block device     253:2
```

显示逻辑卷报告：

```
lvs
```

| LV                        | VG     | Attr       | LSize  | Pool | Origin | Data% | Meta% |
|---------------------------|--------|------------|--------|------|--------|-------|-------|
| Move Log Cpy%Sync Convert |        |            |        |      |        |       |       |
| root                      | centos | -wi-ao---- | 45.62g |      |        |       |       |
| swap                      | centos | -wi-ao---- | 3.89g  |      |        |       |       |
| lv_storage                | vg_dlp | -wi-a----  | 80.00g |      |        |       |       |

扫描逻辑卷：

```
lvscan
```

|        |                                              |
|--------|----------------------------------------------|
| ACTIVE | '/dev/centos/swap' [3.89 GiB] inherit        |
| ACTIVE | '/dev/centos/root' [45.62 GiB] inherit       |
| ACTIVE | '/dev/vg_dlp/lv_storage' [80.00 GiB] inherit |

拍摄逻辑卷的快照：

```
lvcreate -s -L 50G -n snap-lv_storage /dev/vg_dlp/lv_storage #  
从“lv_storage”创建快照“snap-lv_storage”
```

Logical volume "snap-lv\_storage" created.

```
lvdisplay /dev/vg_dlp/lv_storage /dev/vg_dlp/snap-lv_storage
```

```
--- Logical volume ---  
LV Path          /dev/vg_dlp/lv_storage  
LV Name          lv_storage  
VG Name          vg_dlp  
LV UUID          M7mPAD-e2BU-XIVY-z7tN-5SBS-eEiX-biB90f  
LV Write Access  read/write  
LV Creation host, time dlp.srv.world, 2015-07-20 09:33:33 +090  
0  
LV snapshot status source of  
                           snap-lv_storage [active]  
LV Status         available  
# open            0  
LV Size           30.00 GiB  
Current LE        7680  
Segments          1  
Allocation        inherit  
Read ahead sectors auto  
- currently set to 8192  
Block device      253:2  
  
--- Logical volume ---  
LV Path          /dev/vg_dlp/snap-lv_storage  
LV Name          snap-lv_storage  
VG Name          vg_dlp  
LV UUID          YjbZR4-Snih-3KEE-026y-vbQb-sLBq-Uv1CIJ  
LV Write Access  read/write  
LV Creation host, time dlp.srv.world, 2015-07-20 09:34:21 +090  
0  
LV snapshot status active destination for lv_storage  
LV Status         available  
# open            0  
LV Size           30.00 GiB  
Current LE        7680
```

```
COW-table size          30.00 GiB
COW-table LE            7680
Allocated to snapshot   0.00%
Snapshot chunk size     4.00 KiB
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to      8192
Block device             253:5
```

扩展逻辑卷（可以在挂载时操作）：

```
lvextend -L 70G /dev/vg_dlp/lv_storage
```

```
Size of logical volume vg_dlp/lv_storage changed from 30.00 GiB (7680 extents) to 50.00 GiB (12800 extents).
Logical volume lv_storage successfully resized
```

```
lvdisplay /dev/vg_dlp/lv_storage
```

```
--- Logical volume ---
LV Path                  /dev/vg_dlp/lv_storage
LV Name                 lv_storage
VG Name                 vg_dlp
LV UUID                 M7mPAd-e2BU-XIVY-z7tN-5SBS-eEiX-biB90f
LV Write Access         read/write
LV Creation host, time dlp.srv.world, 2015-07-20 09:33:33 +090
0
LV Status               available
# open                  1
LV Size                 70.00 GiB
Current LE              17920
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to      8192
Block device             253:2
```

```
xfs_growfs /mnt # 对于扩展xfs文件系统（指定挂载点）
```

```
meta-data=/dev/mapper/vg_dlp-lv_storage isize=256      agcount=4,
agsize=3276800 blks
              =                     sectsz=512  attr=2, projid32bi
t=1
              =                     crc=0       finobt=0
data      =                     bsize=4096  blocks=13107200, i
maxpct=25
              =                     sunit=0     swidth=0 blks
naming    =version 2           bsize=4096  ascii-ci=0 ftype=0
log       =internal            bsize=4096  blocks=6400, versi
on=2
              =                     sectsz=512  sunit=0 blks, lazy
-count=1
realtime =none                extsz=4096  blocks=0, rtextent
s=0
data blocks changed from 13107200 to 18350080
```

```
resize2fs /dev/vg_dlp/lv_storage # 扩展ext4文件系统的情况
```

```
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vg_dlp/lv_storage is mounted on /mnt; on-line
resizing required
old_desc_blocks = 7, new_desc_blocks = 9
[ 2296.232115] EXT4-fs (dm-2): resizing filesystem from 13107200
to 18350080 blocks
[ 2296.258785] EXT4-fs (dm-2): resized filesystem to 18350080
The filesystem on /dev/vg_dlp/lv_storage is now 18350080 blocks
long.
```

削减逻辑卷（首先卸载目标设备，不能削减xfs文件系统）：

```
e2fsck -f /dev/vg_dlp/lv_storage 50G # 对于ext4，先检查
```

```
e2fsck 1.42.9 (28-Dec-2013)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg_dlp/lv_storage: 11/4587520 files (0.0% non-contiguous),
334056/18350080 blocks
```

```
resize2fs /dev/vg_dlp/lv_storage 50G # 对于ext4，削减文件系统
```

```
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/vg_dlp/lv_storage to 13107200 (4
k) blocks.
The filesystem on /dev/vg_dlp/lv_storage is now 13107200 blocks
long.
```

```
lvreduce -L 50G /dev/vg_dlp/lv_storage # 最后削减逻辑卷
```

```
WARNING: Reducing active logical volume to 50.00 GiB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lv_storage? [y/n]: y
Size of logical volume vg_dlp/lv_storage changed from 70.00 Gi
B (17920 extents) to 50.00 GiB (12800 extents).
Logical volume lv_storage successfully resized
```

删除逻辑卷：

卸载 -> 停止逻辑卷 -> 删除逻辑卷

```
lvchange -an /dev/vg_dlp/lv_storage
```

```
lvremove /dev/vg_dlp/lv_storage
```

```
Logical volume "lv_storage" successfully removed
```

### 附0.3.6.4. 创建镜像卷

创建镜像卷（Mirroring Volume）。

例如，使用物理卷 /dev/sdb1 和 /dev/sdc1 创建镜像卷：

```
vgcreate vg_mirror /dev/sdb1 /dev/sdc1 # 使用  
/dev/sdb1 和 /dev/sdc1 创建卷组“vg_mirror”
```

```
Logical volume "vg_mirror" created
```

```
lvcreate -L 50G -m1 -n lv_mirror vg_mirror # 创建镜像卷
```

```
[ 5133.283498] device-mapper: raid: Superblocks created for new  
array  
[ 5133.285922] md/raid1:mdX: not clean -- starting background re  
construction  
[ 5133.287095] md/raid1:mdX: active with 2 out of 2 mirrors  
[ 5133.287932] Choosing daemon_sleep default (5 sec)  
[ 5133.288688] created bitmap (50 pages) for device mdX  
[ 5133.334556] mdX: bitmap file is out of date, doing full recov  
ery  
[ 5133.667257] mdX: bitmap initialized from disk: read 4 pages,  
set 102400 of 102400 bits  
[ 5133.747852] md: resync of RAID array mdX  
[ 5133.748596] md: minimum _guaranteed_ speed: 1000 KB/sec/disk  
. .  
[ 5133.749676] md: using maximum available idle IO bandwidth (bu  
t not more than 200000 KB/sec) for resync.  
[ 5133.751277] md: using 128k window, over a total of 52428800k.  
Logical volume "lv_mirror" created.
```

```
lvdisplay /dev/vg_mirror/lv_mirror
```

```
--- Logical volume ---
LV Path          /dev/vg_mirror/lv_mirror
LV Name          lv_mirror
VG Name          vg_mirror
LV UUID          P8dakZ-TiWn-T6b5-y6A7-Ecty-ieRZ-vGZAWX
LV Write Access  read/write
LV Creation host, time dlp.srv.world, 2015-07-20 13:18:20 +090
0
LV Status        available
# open           0
LV Size          50.00 GiB
Current LE       12800
Mirrored volumes 2
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 8192
Block device     253:6
```

如果要从已经运行的逻辑卷设置镜像卷，如下配置：

```
vgextend vg_data /dev/sdc1 # 扩展卷组
```

```
Volume group "vg_data" successfully extended
```

```
lvconvert -m1 /dev/vg_data/lv_data /dev/sdc1 # 设置镜像卷
```

```
[ 5710.014313] device-mapper: raid: Superblocks created for new array
[ 5710.019478] md/raid1:mdX: not clean -- starting background reconstruction
[ 5710.020597] md/raid1:mdX: active with 2 out of 2 mirrors
[ 5710.021499] Choosing daemon_sleep default (5 sec)
[ 5710.022279] created bitmap (50 pages) for device mdX
[ 5710.173412] mdX: bitmap file is out of date, doing full recovery
[ 5710.466100] mdX: bitmap initialized from disk: read 4 pages, set 102400 of 102400 bits
[ 5710.567058] md: resync of RAID array mdX
[ 5710.567705] md: minimum _guaranteed_ speed: 1000 KB/sec/disk
.
[ 5710.568658] md: using maximum available idle IO bandwidth (but not more than 200000 KB/sec) for resync.
[ 5710.570204] md: using 128k window, over a total of 52428800k.
```

```
lvs # 确认（如果“Cpy%Sync”变为“100”，则同步完成）
```

| LV      | VG      | Attr       | LSize   | Pool | Origin | Data% | Meta% | Mo |
|---------|---------|------------|---------|------|--------|-------|-------|----|
| ve      | Log     | Cpy%Sync   | Convert |      |        |       |       |    |
| root    | centos  | -wi-ao---  | 45.62g  |      |        |       |       |    |
| swap    | centos  | -wi-ao---  | 3.89g   |      |        |       |       |    |
| lv_data | vg_data | rwi-a-r--- | 50.00g  |      |        |       |       |    |
|         |         | 1.34       |         |      |        |       |       |    |

取消设置镜像卷：

```
lvconvert -m0 /dev/vg_data/lv_data # 指定 -m0 取消设置
```

```
lvs -a -o vg_name,name,devices,size
```

| VG      | LV      | Devices        | LSize  |
|---------|---------|----------------|--------|
| centos  | root    | /dev/sda2(996) | 45.62g |
| centos  | swap    | /dev/sda2(0)   | 3.89g  |
| vg_data | lv_data | /dev/sdb1(0)   | 50.00g |

### 附0.3.6.5. 创建条带卷

创建条带卷（Striped Volume）。

例如，使用物理卷 /dev/sdb1 和 /dev/sdc1 创建条带卷：

```
vgcreate vg_stripped /dev/sdb1 /dev/sdc1 # 使用  
/dev/sdb1 和 /dev/sdc1 创建卷组“vg_stripped”
```

```
Logical volume "vg_stripped" created
```

```
lvcreate -L 50G -i2 -I 64 -n lv_stripped vg_stripped # 创建条带卷
```

```
Wiping xfs signature on /dev/vg_stripped/lv_stripped.  
Logical volume "lv_stripped" created.
```

```
lvdisplay /dev/vg_stripped/lv_stripped
```

```
--- Logical volume ---  
LV Path          /dev/vg_stripped/lv_stripped  
LV Name          lv_stripped  
VG Name          vg_stripped  
LV UUID          5UoeiJ-7Ls9-qYrN-bBoV-6sJH-15HX-FlgnxK  
LV Write Access  read/write  
LV Creation host, time dlp.srv.world, 2015-07-20 14:59:30 +0900  
0  
LV Status        available  
# open           0  
LV Size          50.00 GiB  
Current LE       12800  
Segments         1  
Allocation       inherit  
Read ahead sectors auto  
- currently set to 512  
Block device     253:2
```

```
lvs -a -o vg_name,name,devices,size
```

| VG          | LV         | Devices                   | LSize  |
|-------------|------------|---------------------------|--------|
| centos      | root       | /dev/sda2(996)            | 45.62g |
| centos      | swap       | /dev/sda2(0)              | 3.89g  |
| vg_stripped | lv_striped | /dev/sdb1(0),/dev/sdc1(0) | 50.00g |

## 附0.4. 显示硬件信息

使用 `lshw` 命令显示硬件信息。

安装 `lshw` :

```
yum -y install lshw
```

显示硬件信息：

```
lshw
```

```
d1p.srv.world
  description: Computer
  product: VMware Virtual Platform
  vendor: VMware, Inc.
  version: None
  serial: VMware-56 4d fb 94 3f 44 93 ea-4b 19 25 db 1d cd 9c
2d
  width: 64 bits
  capabilities: smbios-2.4 dmi-2.4 vsyscall32
  configuration: administrator_password=enabled boot=normal fr
ontpanel_password=unknown
    keyboard_password=unknown power-on_password=disabled uuid=56
4DFB94-3F44-93EA-4B19-25DB1DCD9C2D
*-core
  description: Motherboard
  product: 440BX Desktop Reference Platform
  vendor: Intel Corporation
  physical id: 0
  version: None
  serial: None
*-firmware
  description: BIOS
  vendor: Phoenix Technologies LTD
  physical id: 0
  version: 6.00
  date: 05/20/2014
  size: 89KiB
  capabilities: isa pci pcmcia pnp apm upgrade shadowing
escd cdboot bootselect edd
```

```
int5printscreen int9keyboard int14serial int17printer
int10video acpi smartbattery
    biosbootspecification netboot inter int10video acpi sm
artbattery biosbootspecification netboot
*-cpu:0
    description: CPU
    product: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    vendor: Intel Corp.
    physical id: 4
    bus info: cpu@0
    version: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    slot: CPU socket #0
    size: 2800MHz
    capacity: 4230MHz
    width: 64 bits
    capabilities: fpu fpu_exception wp vme de pse tsc msr
    pae mce cx8 apic sep mtrr pge mca
        cmov pat pse36 clflush dts mmx fxsr sse sse2 ss syscal
    l nx rdtscp x86-64 constant_tsc arch_perfmon
        pebs bts nopl xtopology tsc_reliable nonstop_tsc aperf
    mperf pni pclmulqdq vmx ssse3 cx16 pcid sse4_1
        sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx
    f16c rdrand hypervisor lahf_lm ida arat epb pln
        pts dtherm tpr_shadow vnmi ept vpid fsgsbase tsc_adju
    st smep
*-cache
    description: L1 cache
    physical id: 94
    slot: L1 Cache
    size: 16KiB
    capacity: 16KiB
    capabilities: asynchronous internal write-back
.....
.....
```

```
lshw -html > hwinfo.html #用HTML格式显示
```

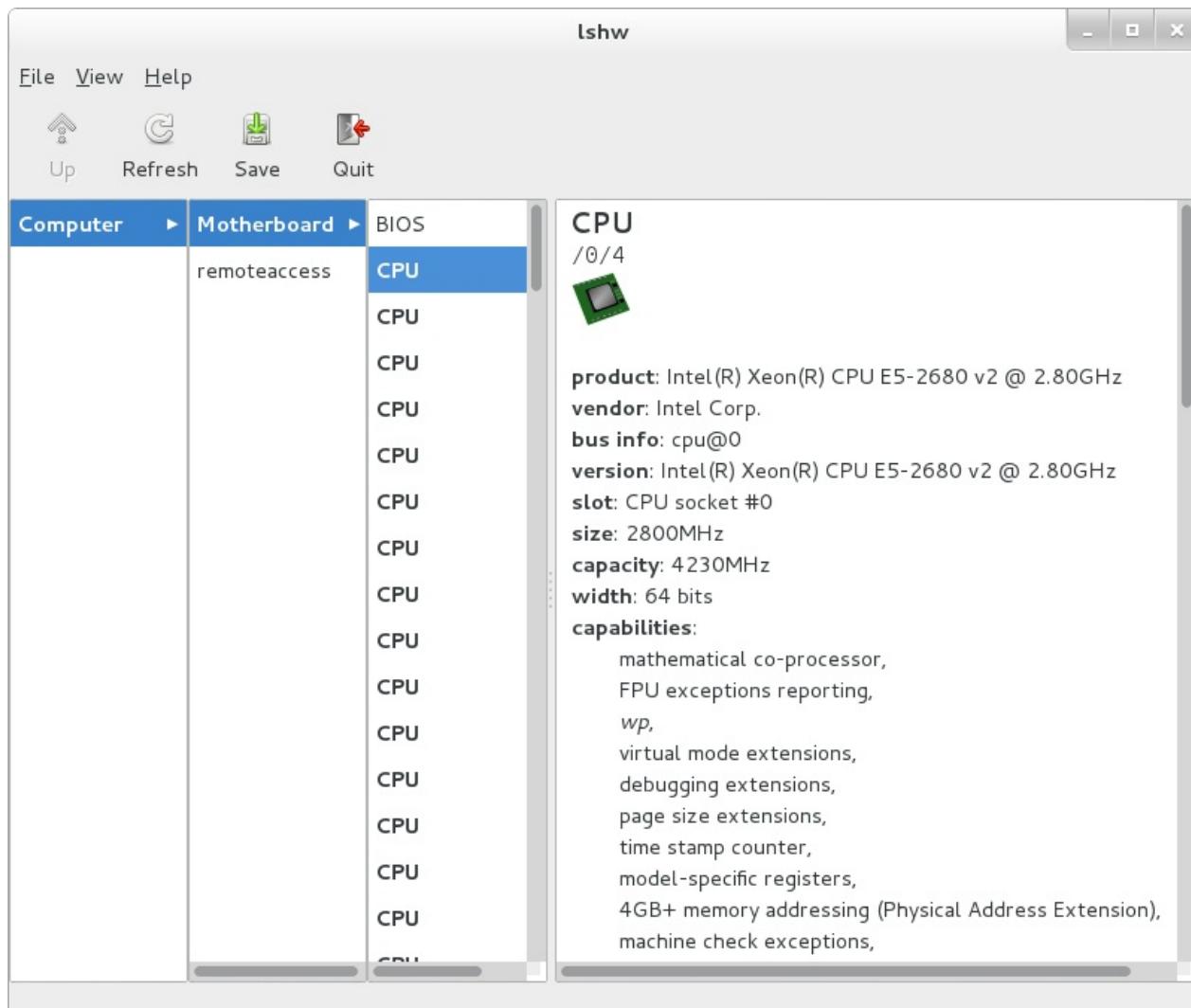
```
lshw -xml > hwinfo.xml #用XML格式显示
```

如果安装了桌面环境，可以使用GUI工具：

```
yum -y install lshw-gui
```

## 附0.4. 显示硬件信息

在应用程序菜单中选择“Hardware Lister”，硬件信息显示如下：



## 附0.5. 分布式文件系统

### 附0.5.1. Hadoop

Hadoop是一个分布式系统基础架构，由Apache基金会开发。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力高速运算和存储。Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。

本例基于以下环境：

- 1) dlp.srv.world (Master Node)
- 2) node01.srv.world (Slave Node)
- 3) node02.srv.world (Slave Node)

现在所有节点[安装JDK](#)。

在所有节点上为Hadoop创建一个用户：

```
useradd -d /usr/hadoop hadoop  
chmod 755 /usr/hadoop  
passwd hadoop
```

```
Changing password for user hadoop.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.
```

以“hadoop”用户登录到主节点，并创建SSH密钥对（无密码短语）并将其发送到其它节点：

```
ssh-keygen
```

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/usr/hadoop/.ssh/id_rsa):  
Created directory '/usr/hadoop/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /usr/hadoop/.ssh/id_rsa.  
Your public key has been saved in /usr/hadoop/.ssh/id_rsa.pub.  
The key fingerprint is:  
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx hadoop@dlp.srv.w  
orld  
The key's randomart image is:
```

发送密钥到各节点（包括localhost）：

```
ssh-copy-id localhost
```

```
The authenticity of host 'localhost (::1)' can't be established.  
ECDSA key fingerprint is xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:  
xx:xx:xx.  
Are you sure you want to continue connecting (yes/no)? yes  
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s)  
, to filter out any that are already installed  
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if yo  
u are prompted now it is to install the new keys  
hadoop@localhost's password:
```

Number of key(s) added: 1

```
Now try logging into the machine, with: "ssh 'localhost'"  
and check to make sure that only the key(s) you wanted were adde  
d.
```

```
ssh-copy-id node01.srv.world
```

```
ssh-copy-id node02.srv.world
```

在所有节点上安装Hadoop（以“hadoop”用户操作）。确认[下载最新版本](#)：

```
curl -O http://ftp.jaist.ac.jp/pub/apache/hadoop/common/hadoop-  
2.7.1/hadoop-2.7.1.tar.gz
```

```
tar zxvf hadoop-2.7.1.tar.gz -C /usr/hadoop --strip-components 1
```

编辑 `~/.bash_profile` 文件：

```
# 添加以下内容到最后  
export HADOOP_HOME=/usr/hadoop  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_YARN_HOME=$HADOOP_HOME  
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"  
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native  
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

```
source ~/.bash_profile
```

在主节点上配置Hadoop（以“hadoop”用户操作）：

为所有节点上的数据创建目录：

```
mkdir ~/datanode
```

```
ssh node01.srv.world "mkdir ~/datanode"
```

```
ssh node02.srv.world "mkdir ~/datanode"
```

编辑 `~/etc/hadoop/dfs-site.xml` 文件：

```
# 添加到<configuration> - </configuration>之间  
<configuration>  
  <property>  
    <name>dfs.replication</name>  
    <value>2</value>  
  </property>  
  <property>  
    <name>dfs.datanode.data.dir</name>  
    <value>file:///usr/hadoop/datanode</value>  
  </property>  
</configuration>
```

发送到从节点：

```
scp ~/etc/hadoop/hdfs-site.xml node01.srv.world:~/etc/hadoop/
```

```
scp ~/etc/hadoop/hdfs-site.xml node02.srv.world:~/etc/hadoop/
```

编辑 `~/etc/hadoop/core-site.xml` 文件：

```
# 添加到<configuration> - </configuration>之间
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://dlp.srv.world:9000</value>
  </property>
</configuration>
```

发送到从节点：

```
scp ~/etc/hadoop/core-site.xml node01.srv.world:~/etc/hadoop/
```

```
scp ~/etc/hadoop/core-site.xml node02.srv.world:~/etc/hadoop/
```

```
sed -i -e 's/\${JAVA_HOME}/\usr\java\default/'  
~/etc/hadoop/hadoop-env.sh
```

发送到从节点：

```
scp ~/etc/hadoop/hadoop-env.sh node01.srv.world:~/etc/hadoop/
```

```
scp ~/etc/hadoop/hadoop-env.sh node02.srv.world:~/etc/hadoop/
```

```
mkdir ~/namenode
```

编辑 `~/etc/hadoop/hdfs-site.xml` 文件：

```
# 添加到<configuration> - </configuration>之间
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///usr/hadoop/namenode</value>
  </property>
</configuration>
```

编辑 `~/etc/hadoop/mapred-site.xml` 文件：

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

编辑 `~/etc/hadoop/yarn-site.xml` 文件：

```
# 添加到<configuration> - </configuration>之间
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>d1p.srv.world</value>
  </property>
  <property>
    <name>yarn.nodemanager.hostname</name>
    <value>d1p.srv.world</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

编辑 `~/etc/hadoop/slaves` 文件：

```
# 添加所有节点（删除localhost）
d1p.srv.world
node01.srv.world
node02.srv.world
```

格式化NameNode并启动的Hadoop服务：

```
hdfs namenode -format
```

```
15/07/28 19:58:14 INFO namenode.NameNode: STARTUP_MSG:  
*****  
STARTUP_MSG: Starting NameNode  
STARTUP_MSG: host = dlp.srv.world/10.0.0.30  
STARTUP_MSG: args = [-format]  
STARTUP_MSG: version = 2.7.1  
....  
....  
15/07/28 19:58:17 INFO namenode.NameNode: SHUTDOWN_MSG:  
*****  
SHUTDOWN_MSG: Shutting down NameNode at dlp.srv.world/10.0.0.30  
*****/
```

### start-dfs.sh

```
Starting namenodes on [dlp.srv.world]  
dlp.srv.world: starting namenode, logging to /usr/hadoop/logs/ha  
oop-hadoop-namenode-dlp.srv.world.out  
dlp.srv.world: starting datanode, logging to /usr/hadoop/logs/ha  
oop-hadoop-datanode-dlp.srv.world.out  
node02.srv.world: starting datanode, logging to /usr/hadoop/logs  
/hadoop-hadoop-datanode-node02.srv.world.out  
node01.srv.world: starting datanode, logging to /usr/hadoop/logs  
/hadoop-hadoop-datanode-node01.srv.world.out  
Starting secondary namenodes [0.0.0.0]  
0.0.0.0: starting secondarynamenode, logging to /usr/hadoop/logs  
/hadoop-hadoop-secondarynamenode-dlp.srv.world.out
```

### start-yarn.sh

```
starting yarn daemons  
starting resourcemanager, logging to /usr/hadoop/logs/yarn-hadoo  
p-resourcemanager-dlp.srv.world.out  
dlp.srv.world: starting nodemanager, logging to /usr/hadoop/logs  
/yarn-hadoop-nodemanager-dlp.srv.world.out  
node02.srv.world: starting nodemanager, logging to /usr/hadoop/l  
ogs/yarn-hadoop-nodemanager-node02.srv.world.out  
node01.srv.world: starting nodemanager, logging to /usr/hadoop/l  
ogs/yarn-hadoop-nodemanager-node01.srv.world.out
```

```
jps # 显示状态 (结果如下所示)
```

```
2130 NameNode  
2437 SecondaryNameNode  
2598 ResourceManager  
2710 NodeManager  
3001 Jps  
2267 DataNode
```

执行示例程序以确认正常工作：

```
hdfs dfs -mkdir /test # 创建一个目录 /test
```

```
hdfs dfs -copyFromLocal ~/NOTICE.txt /test # 将本地文件复制到 /test
```

```
hdfs dfs -cat /test/NOTICE.txt # 显示文件的内容
```

```
This product includes software developed by The Apache Software Foundation (http://www.apache.org/).
```

```
hadoop jar ~/share/hadoop/mapreduce/hadoop-mapreduce-examples-  
2.7.1.jar wordcount /test/NOTICE.txt /output01 # 执行示例程序
```

```
15/07/28 19:28:47 INFO client.RMProxy: Connecting to ResourceManager at dlp.srv.world/10.0.0.30:8032  
15/07/28 19:28:48 INFO input.FileInputFormat: Total input paths to process : 1  
15/07/28 19:28:48 INFO mapreduce.JobSubmitter: number of splits: 1  
.....  
.....
```

```
hdfs dfs -ls /output01 # 显示结果
```

```
Found 2 items  
-rw-r--r--    2 hadoop supergroup      0 2015-07-29 14:29 /output  
01/_SUCCESS  
-rw-r--r--    2 hadoop supergroup  123 2015-07-29 14:29 /output  
01/part-r-00000
```

```
hdfs dfs -cat /output01/part-r-00000 # 显示结果文件的内容（生成字数）
```

```
(http://www.apache.org/). 1
Apache 1
Foundation 1
Software 1
The 1
This 1
by 1
developed 1
includes 1
product 1
software 1
```

访问 [http://\(服务器的主机名或IP地址\):50070/](http://(服务器的主机名或IP地址):50070/)，可以看到Hadoop集群摘要：

The screenshot shows a Mozilla Firefox window titled "Namenode information - Mozilla Firefox". The address bar displays "dlp.server.world:50070/dfshealth.html#tab-overview". The main content area has a green header bar with the text "Hadoop" and several tabs: "Overview" (which is selected), "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities". Below this is a large section titled "Overview 'dlp.server.world:9000' (active)". This section contains a table with the following data:

|                       |                                                           |
|-----------------------|-----------------------------------------------------------|
| <b>Started:</b>       | Wed Jul 29 16:09:49 JST 2015                              |
| <b>Version:</b>       | 2.7.1, r15ecc87ccf4a0228f35af08fc56de536e6ce657a          |
| <b>Compiled:</b>      | 2015-06-29T06:04Z by jenkins from (detached from 15ecc87) |
| <b>Cluster ID:</b>    | CID-15b1101f-5301-4bd3-9c6b-bfee16bce46c                  |
| <b>Block Pool ID:</b> | BP-802493637-10.0.0.30-1438153777333                      |

访问 [http://\(服务器的主机名或IP地址\):8088/](http://(服务器的主机名或IP地址):8088/)，可以看到Hadoop集群信息：

All Applications - Mozilla Firefox

All Applications x +

dlp.server.world:8088/cluster ↻ Search ☆ ☰

 All Application

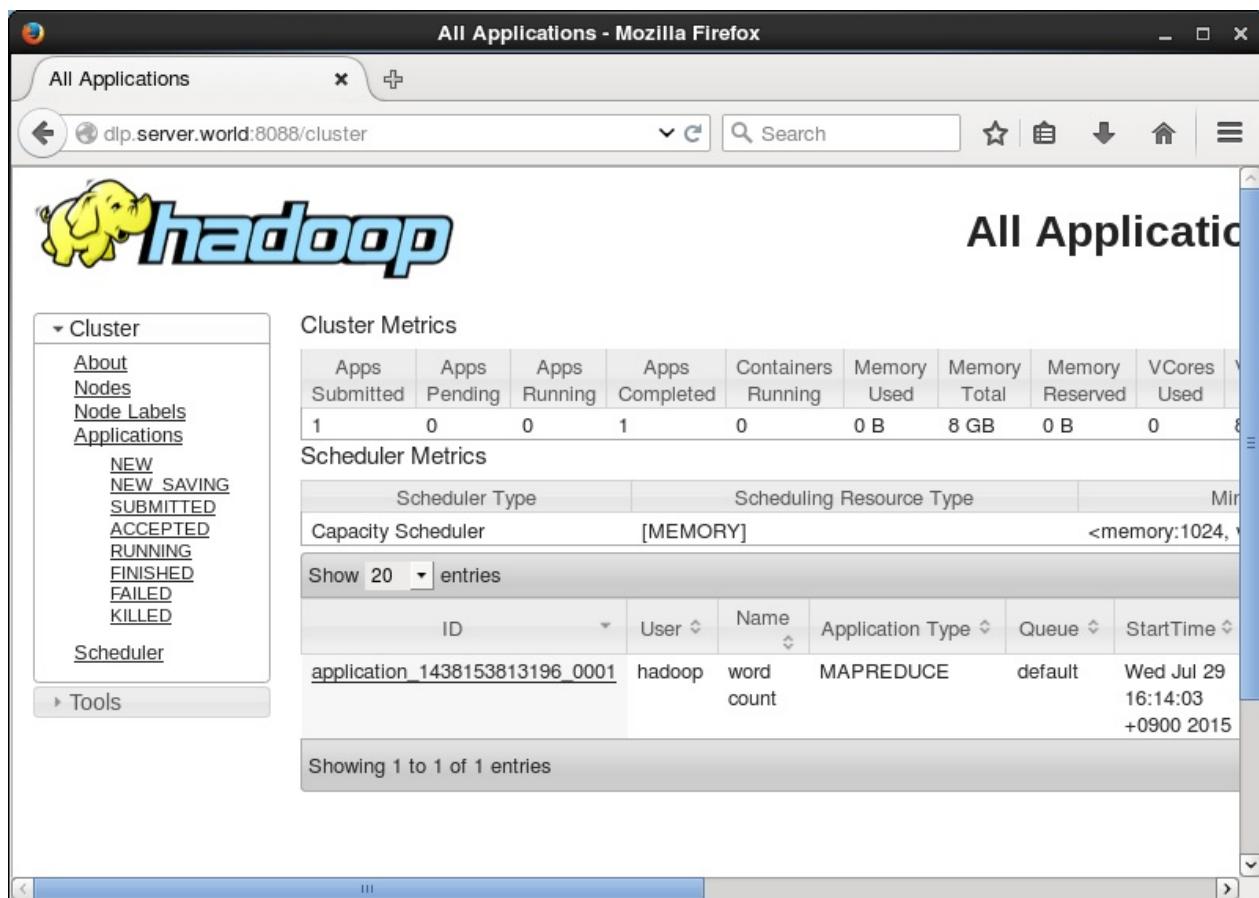
**Cluster Metrics**

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used | VCores Reserved |
|----------------|--------------|--------------|----------------|--------------------|-------------|--------------|-----------------|-------------|-----------------|
| 1              | 0            | 0            | 1              | 0                  | 0 B         | 8 GB         | 0 B             | 0           | 8               |

**Scheduler Metrics**

| Scheduler Type                 |          | Scheduling Resource Type |                  |         | Mirrored Resource Type         |          |          |
|--------------------------------|----------|--------------------------|------------------|---------|--------------------------------|----------|----------|
| Capacity Scheduler             | [MEMORY] | <memory:1024, vcores:1>  |                  |         | <memory:1024, vcores:1>        |          |          |
| Show 20 entries                |          |                          |                  |         |                                |          |          |
| ID                             | User     | Name                     | Application Type | Queue   | StartTime                      | End Time | Duration |
| application_1438153813196_0001 | hadoop   | word count               | MAPREDUCE        | default | Wed Jul 29 16:14:03 +0900 2015 |          |          |

Showing 1 to 1 of 1 entries



## 附0.6. 更改运行级别

如果需要，可如下更改运行级别（RunLevel）。

运行级别设置链接到 `/etc/systemd/system/default.target`。

例如，没有GUI的默认设置如下：

```
systemctl get-default # 显示当前设置
```

```
multi-user.target
```

```
ll /etc/systemd/system/default.target
```

```
lrwxrwxrwx. 1 root root 37 Jul 9 06:04 /etc/systemd/system/default.target -> /lib/systemd/system/multi-user.target
```

例如，如果要将运行级别更改为“Graphical-login”，按如下所示进行设置：

```
systemctl set-default graphical.target
```

```
rm '/etc/systemd/system/default.target'  
ln -s '/usr/lib/systemd/system/graphical.target' '/etc/systemd/system/default.target'
```

```
systemctl get-default # 确认设置
```

```
graphical.target
```

```
ll /etc/systemd/system/default.target
```

```
lrwxrwxrwx 1 root root 36 Jul 9 21:55 /etc/systemd/system/default.target -> /lib/systemd/system/graphical.target
```



# 附1. 一些可能有用的

- 附1.1. 系统安全
  - 附1.1.1. Clam AntiVirus
  - 附1.1.2. RKHunter
  - 附1.1.3. 审计与日志
    - 附1.1.3.1. Lynis
    - 附1.1.3.2. Auditd
      - 附1.1.3.2.1. 安装Auditd
      - 附1.1.3.2.2. 输出日志到远程主机
      - 附1.1.3.2.3. 使用ausearch搜索日志
      - 附1.1.3.2.4. 使用aureport显示日志
      - 附1.1.3.2.5. 添加审计规则
    - 附1.1.3.3. Rsyslog
      - 附1.1.3.3.1. 输出日志到远程主机
      - 附1.1.3.3.2. 输出日志到数据库
  - 附1.1.4. 入侵检测系统
    - 附1.1.4.1. AIDE
    - 附1.1.4.2. Tripwire
- 附1.2. 加入Windows活动目录
- 附1.3. 访问控制
  - 附1.3.1. ACL
  - 附1.3.2. TCP Wrapper
  - 附1.3.3. SELinux
    - 附1.3.3.1. 运行模式
    - 附1.3.3.2. 策略类型
    - 附1.3.3.3. SELinux上下文
    - 附1.3.3.4. 更改布尔值
    - 附1.3.3.5. 更改文件类型
    - 附1.3.3.6. 更改端口类型
    - 附1.3.3.7. 搜索日志
    - 附1.3.3.8. 使用SETroubleShoot
    - 附1.3.3.9. 使用audit2allow
    - 附1.3.3.10. 使用matchpathcon

- 附1.3.3.11. 使用sesearch
- 附1.4. 文件同步
  - 附1.4.1 Rsync
  - 附1.4.2. Lsyncd
- 附1.5. PowerShell
- 附1.6. 项目管理与版本控制
  - 附1.6.1. GitLab
  - 附1.6.2. Redmine
  - 附1.6.3. Gitolite
    - 附1.6.3.1. 安装Gitolite
    - 附1.6.3.2. 添加用户
    - 附1.6.3.3. 添加新库
    - 附1.6.3.4. 设置访问控制
- 附1.7. 系统管理工具
  - 附1.7.1. Cockpit
  - 附1.7.2. Ajenti
  - 附1.7.3. Webmin
  - 附1.7.4. Usermin
  - 附1.7.5. Virtualmin
  - 附1.7.6. Spacewalk
    - 附1.7.6.1. 安装Spacewalk
    - 附1.7.6.2. 初始设置
    - 附1.7.6.3. 客户端设置
- 附1.8. 配置管理工具
  - 附1.8.1. Salt
    - 附1.8.1.1. 安装Salt
    - 附1.8.1.2. 基本用法
    - 附1.8.1.3. 使用Salt State文件
    - 附1.8.1.4. 使用Salt-cp
  - 附1.8.2. Puppet
    - 附1.8.2.1. 安装Puppet
    - 附1.8.2.2. 文件资源
    - 附1.8.2.3. 软件包资源
    - 附1.8.2.4. 服务资源
    - 附1.8.2.5. 组资源
    - 附1.8.2.6. 用户资源

- 附1.8.2.7. 执行资源
- 附1.8.2.8. 节点部分
- 附1.8.2.9. 类部分
- 附1.8.2.10. facter变量
- 附1.8.3. Ansible
  - 附1.8.3.1. 安装Ansible
  - 附1.8.3.2. 基本用法
  - 附1.8.3.3. 使用Playbook1
  - 附1.8.3.4. 使用Playbook2
  - 附1.8.3.5. 使用Playbook3
  - 附1.8.3.6. 使用Playbook4
  - 附1.8.3.7. 使用Playbook5
  - 附1.8.3.8. 使用Playbook6
- 附1.8.4. Func
  - 附1.8.4.1. 安装Func
  - 附1.8.4.2. 基本操作
  - 附1.8.4.3. 使用YumModule
  - 附1.8.4.4. 使用CopyFileModule
  - 附1.8.4.5. 使用CommandModule
- 附1.9. 防火墙
  - 附1.9.1. Firewalld
    - 附1.9.1.1. 基本操作
    - 附1.9.1.2. IP伪装
  - 附1.9.2. iptables
    - 附1.9.2.1. 设置示例1
    - 附1.9.2.2. 设置示例2
    - 附1.9.2.3. 设置示例3
    - 附1.9.2.4. 设置示例4
  - 附1.9.3. Fail2ban
- 附1.10. 高可用性集群
  - 附1.10.1. Pacemaker
    - 附1.10.1.1. 安装Pacemaker
    - 附1.10.1.2. 添加资源
    - 附1.10.1.3. CLVM + GFS2
- 附1.11. 消息服务器
  - 附1.11.1. RabbitMQ

- 附1.11.1.1. 安装RabbitMQ
- 附1.11.1.2. 在Python上使用
- 附1.11.1.3. 在PHP上使用
- 附1.11.1.4. 在Ruby上使用
- 附1.11.1.5. 使用Web界面
- 附1.11.1.6. 使用rabbitmqadmin
- 附1.11.1.7. 配置集群
- 附1.12. 备份管理工具
  - 附1.12.1. Bacula
    - 附1.12.1.1. 安装Bacula
    - 附1.12.1.2. 配置Bacula服务器
    - 附1.12.1.3. 配置客户端
    - 附1.12.1.4. 备份操作
    - 附1.12.1.5. 恢复操作

## 附1.1. 系统安全

### 附1.1.1. Clam AntiVirus

Clam AntiVirus (ClamAV) 是免费而且开放源代码的防毒软件，软件与病毒码的更新皆由社群免费发布。目前ClamAV主要是使用在由Linux、FreeBSD等Unix-like系统架设的邮件服务器上，提供电子邮件的病毒扫描服务。

```
yum --enablerepo=epel -y install clamav clamav-update # 从EPEL安装  
sed -i -e "s/^Example/#Example/" /etc/freshclam.conf  
  
freshclam # 更新病毒库
```

```
ClamAV update process started at Fri Aug 29 22:03:30 2014  
main.cld is up to date (version: 55,  sigs: 2424225,  f-level: 60,  
builder: neo)  
daily.cvd is up to date (version: 19314,  sigs: 1094505,  f-level:  
63,  builder: neo)  
bytecode.cvd is up to date (version: 242,  sigs: 46,  f-level: 63,  
builder: dgoddard)
```

尝试扫描：

```
clamscan --infected --remove --recursive /home
```

```
----- SCAN SUMMARY -----  
Known viruses: 3575245  
Engine version: 0.98.4  
Scanned directories: 2  
Scanned files: 3  
Infected files: 0  
Data scanned: 0.00 MB  
Data read: 0.00 MB (ratio 0.00:1)  
Time: 10.369 sec (0 m 10 s)
```

```
curl -O http://www.eicar.org/download/eicar.com # 下载试用病毒  
clamscan --infected --remove --recursive .
```

```
./eicar.com: Eicar-Test-Signature FOUND
./eicar.com: Removed. # 检测到病毒

----- SCAN SUMMARY -----
Known viruses: 3575245
Engine version: 0.98.4
Scanned directories: 3
Scanned files: 10
Infected files: 1
Data scanned: 0.00 MB
Data read: 256.57 MB (ratio 0.00:1)
Time: 10.307 sec (0 m 10 s)
```

### 附1.1.2. RKhunter

安装Rootkit检测工具[RKhunter](#)。

```
yum --enablerepo=epel -y install rkhunter # 从EPEL安装
```

配置和使用RKhunter（为了定期检查，检查脚本安装在 `cron.daily` 目录下，每天由Cron执行）：

编辑 `/etc/sysconfig/rkhunter` 文件：

```
# 报告的收件人地址
MAILTO=root@localhost
# 如果指定“yes”，扫描更详细
DIAG_SCAN=no
```

```
rkhunter --update # 更新数据库
```

```
rkhunter --propupd # 更新系统文件属性
```

执行检查（`--sk` 表示跳过按回车键，如果指定 `--rwo` 只显示警告）：

```
rkhunter --check --sk
```

```
[ Rootkit Hunter version 1.4.2 ]
```

```
Checking system commands...
```

```
Performing 'strings' command checks
  Checking 'strings' command [ 0
K ]

  Performing 'shared libraries' checks
    Checking for preloading variables [ N
one found ]
    Checking for preloaded libraries [ N
one found ]
    Checking LD_LIBRARY_PATH variable [ N
ot found ]

  Performing file properties checks
    Checking for prerequisites [ 0
K ]
    /usr/sbin/adduser [ 0
K ]
    /usr/sbin/chkconfig [ 0
K ]
    /usr/sbin/chroot [ 0
K ]
    /usr/sbin/depmod [ 0
K ]
    /usr/sbin/fsck [ 0
K ]

.....
.....
System checks summary
=====

File properties checks...
  Files checked: 121
  Suspect files: 0

Rootkit checks...
  Rootkits checked : 365
  Possible rootkits: 0

Applications checks...
```

```
All checks skipped
```

```
The system checks took: 1 minute and 35 seconds
```

```
All results have been written to the log file: /var/log/rkhunter  
/rkhunter.log
```

```
No warnings were found while checking the system.
```

## 附1.1.3. 审计与日志

### 附1.1.3.1. Lynis

Lynis扫描系统的配置，并创建概述系统信息与安全问题所使用的专业审计。

```
yum --enablerepo=epel -y install lynis 从EPEL安装
```

使用Lynis：

```
lynis audit system # 初始扫描运行如下
```

```
....  
....  
=====
```

```
Lynis security scan details:
```

```
Hardening index : 65 [#####]  
Tests performed : 200  
Plugins enabled : 0
```

```
Components:
```

- Firewall [V]
- Malware scanner [X]

```
Lynis Modules:
```

- Compliance Status [?]
- Security Audit [V]
- Vulnerability Scan [V]

```
Files:  
- Test and debug information      : /var/log/lynis.log  
- Report data                    : /var/log/lynis-report.dat
```

=====

=====

### Lynis 2.3.2

Auditing, system hardening, and compliance for UNIX-based systems  
(Linux, macOS, BSD, and others)

2007-2016, CISOfy - <https://ciscofy.com/lynis/>  
Enterprise support available (compliance, plugins, interface and tools)

=====

=====

[TIP]: Enhance Lynis audits by adding your settings to custom.prf (see /etc/lynis/default.prf for all settings)

扫描结果的报告保存在 /var/log/lynis-report.dat 中。使用单词“warning”或“suggestion”搜索文件，然后如下显示推荐的设置：

```
grep -E "warning|suggestion" /var/log/lynis-report.dat
```

```
suggestion[]="BOOT-5122|Set a password on GRUB bootloader to prevent altering boot configuration (e.g. boot  
suggestion[]="AUTH-9286|Configure minimum password age in /etc/login.defs|-|-|  
suggestion[]="AUTH-9286|Configure maximum password age in /etc/login.defs|-|-|  
suggestion[]="AUTH-9328|Default umask in /etc/profile or /etc/profile.d/custom.sh could be more strict (e.g.  
suggestion[]="FILE-6310|To decrease the impact of a full /home file system, place /home on a separated partition  
suggestion[]="FILE-6310|To decrease the impact of a full /tmp file system, place /tmp on a separated partition  
suggestion[]="FILE-6310|To decrease the impact of a full /var fil
```

```
e system, place /var on a separated partition  
suggestion[]="STRG-1840|Disable drivers like USB storage when not  
used, to prevent unauthorized storage or da  
suggestion[]="STRG-1846|Disable drivers like firewire storage whe  
n not used, to prevent unauthorized storage  
suggestion[]="NAME-4404|Add the IP name and FQDN to /etc/hosts fo  
r proper name resolving|-|-|  
suggestion[]="PKGS-7384|Install package 'yum-utils' for better co  
nsistency checking of the package database|-  
suggestion[]="NETW-3032|Consider running ARP monitoring software  
(arpwatch)|-|-|  
warning[]="MAIL-8818|Found mail_name in SMTP banner, and/or mail_  
name contains 'Postfix'|-|-|  
suggestion[]="MAIL-8818|You are advised to hide the mail_name (op  
tion: smtpd_banner) from your postfix config  
suggestion[]="FIRE-4513|Check iptables rules to see which rules a  
re currently not used|-|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|Allow  
TcpForwarding (YES --> NO)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|Clien  
tAliveCountMax (3 --> 2)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|Compr  
ession (DELAYED --> NO)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|LogLe  
vel (INFO --> VERBOSE)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|MaxAu  
thTries (6 --> 1)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|MaxSe  
ssions (10 --> 2)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|Permit  
RootLogin (YES --> NO)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|Port  
(22 --> )|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|TCPKe  
epAlive (YES --> NO)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|UseDN  
S (YES --> NO)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|X11Fo  
rwarding (YES --> NO)|-|  
suggestion[]="SSH-7408|Consider hardening SSH configuration|Allow  
AgentForwarding (YES --> NO)|-|  
suggestion[]="BANN-7126|Add a legal banner to /etc/issue, to warn
```

```
unauthorized users|-|-|
suggestion[] = BANN-7130 | Add legal banner to /etc/issue.net, to warn unauthorized users|-|-|
suggestion[] = ACCT-9622 | Enable process accounting|-|-|
suggestion[] = ACCT-9626 | Enable sysstat to collect accounting (no results)|-|-|
suggestion[] = ACCT-9630 | Audit daemon is enabled with an empty ruleset. Disable the daemon or define rules|-|-|
suggestion[] = TIME-3160 | Some time servers missing in step-tickers file|-|-|
suggestion[] = FINT-4350 | Install a file integrity tool to monitor changes to critical and sensitive files|-|-|
suggestion[] = TOOL-5002 | Determine if automation tools are present for system management|-|-|
suggestion[] = KRLN-6000 | One or more sysctl values differ from the scan profile and could be tweaked|-|-|
suggestion[] = HRDN-7222 | Harden compilers like restricting access to root user only|-|-|
suggestion[] = HRDN-7230 | Harden the system by installing at least one malware scanner, to perform periodic fil
```

### 附1.1.3.2. Auditd

通过[Auditd](#)配置系统审计，可以监控系统调用，安全事件，文件访问，命令执行等。

#### 附1.1.3.2.1. 安装Auditd

CentOS7默认安装Audit软件包，如果没有，则如下安装：

```
yum -y install audit
```

```
service auditd start
systemctl enable auditd
```

注： `service` 命令是与Auditd守护进程正确交互的唯一方法。需要使用 `service` 命令，以便 `auid` 值被正确记录。使用 `systemctl` 命令只能执行两个操作：`enable` 和 `status`。

编辑 `/etc/audit/auditd.conf` 文件，更改Auditd的一些设置：

```
# 指定日志文件
log_file = /var/log/audit/audit.log

# 日志文件数（如果指定了“max_log_file_action=ROTATE”）
num_logs = 5

# 日志文件中的主机名
# 有效值：NONE，HOSTNAME，FQD，NUMERIC，USER
name_format = NONE

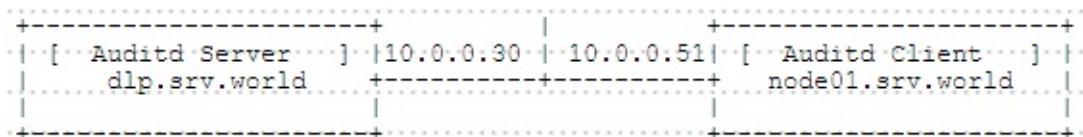
# 任意主机名（如果指定了“name_format=USER”）
name = mydomain

# 日志文件的最大大小（MB兆字节）
max_log_file = 6

# 指定如果日志文件的大小超过限制的操作
# 有效值：IGNORE，SYSLOG，SUSPEND，ROTATE，KEEP_LOGS
max_log_file_action = ROTATE
```

### 附1.1.3.2.2. 输出日志到远程主机

本例基于以下环境：



配置从远程主机接收审计日志的Auditd主机（Auditd Server）：

编辑 `/etc/audit/auditd.conf` 文件：

```
# 取消注释并指定侦听端口
tcp_listen_port = 60
```

```
service auditd restart
```

配置发送审核日志的Auditd客户端主机（Auditd Client）：

```
yum -y install audispd-plugins
```

编辑 `/etc/audisp/plugins.d/au-remote.conf` 文件：

```
# 更改  
active = yes
```

编辑 `/etc/audisp/audisp-remote.conf` 文件：

```
# 接收日志的远程服务器  
remote_server = dlp.srv.world  
  
# 指定端口（远程服务器倾听的端口）  
port = 60
```

编辑 `/etc/audit/auditd.conf` 文件：

```
# 更改（不在本地文件系统上记录日志）  
log_format = NOLOG
```

```
service auditd restart
```

配置完成，远程主机上的审计日志记录如下：

```
tail -5 /var/log/audit/audit.log
```

```
node=node01.srv.world type=USER_START msg=audit(1456385789.273:1  
01): pid=1141 uid=0 auid=0 ses=1 msg='op=.....  
node=node01.srv.world type=USER_END msg=audit(1456385789.278:102  
): pid=1141 uid=0 auid=0 ses=1 msg='op=PA.....  
node=node01.srv.world type=CRED_DISP msg=audit(1456385789.278:10  
3): pid=1141 uid=0 auid=0 ses=1 msg='op=P.....  
node=node01.srv.world type=USER_END msg=audit(1456385791.441:104  
): pid=1120 uid=0 auid=0 ses=1 msg='op=PA.....  
node=node01.srv.world type=CRED_DISP msg=audit(1456385791.442:10  
5): pid=1120 uid=0 auid=0 ses=1 msg='op=P.....
```

如果[TCP Wrapper](#)安装在审核日志接收主机上，可以对Auditd使用TCP访问控制：

编辑 `/etc/audit/auditd.conf` 文件：

```
# 添加到最后  
use_libwrap = yes
```

```
service auditd restart
```

编辑 `/etc/hosts.deny` 文件：

```
# 默认拒绝所有  
auditd: ALL
```

编辑 `/etc/hosts.allow` 文件：

```
# 设置允许的主机  
auditd: 10.0.0.51
```

### 附1.1.3.2.3. 使用ausearch搜索日志

一些默认审计规则，如：系统登录，用户帐户修改，Sudo操作等，这些日志记录在 `/var/log/audit/audit.log` 中。

日志是文本格式，可以直接查看：

```
tail -5 /var/log/audit/audit.log
```

```
node=dlp.srv.world type=USER_START msg=audit(1456386950.783:116): pid=10697 uid=0 auid=0 ses=1 msg='op=P...  
node=dlp.srv.world type=USER_END msg=audit(1456386950.799:117): pid=10697 uid=0 auid=0 ses=1 msg='op=PAM...  
node=dlp.srv.world type=CRED_DISP msg=audit(1456386950.799:118): pid=10697 uid=0 auid=0 ses=1 msg='op=PA...  
node=dlp.srv.world type=USER_END msg=audit(1456386952.872:119): pid=10676 uid=0 auid=0 ses=1 msg='op=PAM...  
node=dlp.srv.world type=CRED_DISP msg=audit(1456386952.872:120): pid=10676 uid=0 auid=0 ses=1 msg='op=PA...
```

许多日志都被记录在“audit.log”中，很复杂，所以由Audit软件包提供的 `ausearch` 命令来搜索具体的日志：

```
ausearch --message USER_LOGIN --interpret # 搜索USER_LOGIN日志
```

```
----  
node=dlp.srv.world type=USER_LOGIN msg=audit(02/26/2016 09:21:35  
.121:44) : pid=610 uid=root auid=root ses=...  
----  
node=node01.srv.world type=USER_LOGIN msg=audit(02/26/2016 09:40  
:29.419:46) : pid=625 uid=root auid=root s...  
....  
....  
node=node01.srv.world type=USER_LOGIN msg=audit(02/26/2016 10:34  
:51.089:44) : pid=620 uid=root auid=root s...
```

```
ausearch -x sudo -ua 1000 # 搜索UserID：1000的Sudo操作
```

```
----  
time->Tue Feb 23 09:52:23 2016  
node=dlp.srv.world type=USER_AUTH msg=audit(1456188743.819:49):  
pid=960 uid=1000 auid=0 ses=1 msg='op=...  
----  
time->Tue Feb 23 09:52:23 2016  
node=dlp.srv.world type=USER_ACCT msg=audit(1456188743.819:50):  
pid=960 uid=1000 auid=0 ses=1 msg='op=...  
....  
....  
time->Fri Feb 26 09:48:50 2016  
node=node01.srv.world type=USER_ACCT msg=audit(1456447730.031:52)  
): pid=966 uid=1000 auid=0 ses=1 msg='...
```

```
ausearch --node dlp.srv.world --success no # 在 dlp.svv.world 上搜  
索失败事件
```

```
----  
time->Thu Feb 25 17:46:57 2016  
node=dlp.srv.world type=USER_END msg=audit(1456390017.044:129):  
pid=608 uid=0 auid=0 ses=1 msg='..... res=failed'  
----  
time->Thu Feb 25 17:46:57 2016  
node=dlp.srv.world type=SERVICE_START msg=audit(1456390017.111:1  
47): pid=1 uid=0 auid=429496729 ..... res=failed'  
.....  
.....  
time->Fri Feb 26 09:50:10 2016  
node=dlp.srv.world type=SERVICE_STOP msg=audit(1456447810.331:63  
): pid=1 uid=0 auid=4294967295 ..... res=failed'
```

```
ausearch --start 02/07/2016 --end 02/21/2016 -ul 1000 # 搜索  
UserID : 1000从2016/2/7到2016/2/21的登录日志
```

```
----  
time->Tue Feb 7 09:54:51 2016  
type=LOGIN msg=audit(1456188891.234:69): pid=976 uid=0 old-auid=  
4294967295 auid=1000 old-ses=4294967295 s...  
----  
time->Tue Feb 7 09:54:51 2016  
type=USER_START msg=audit(1456188891.244:70): pid=976 uid=0 auid  
=1000 ses=2 msg='op=PAM:session_open gran...  
  
time->Tue Feb 21 11:13:38 2016  
type=USER_END msg=audit(1456193618.644:159): pid=8105 uid=0 auid  
=1000 ses=6 msg='op=PAM:session_close gra...
```

### 附1.1.3.2.4. 使用aureport显示日志

可以使用Audit软件包提供的 `aureport` 命令来简要显示审计日志。

下面演示如何使用[aureport](#)命令：

```
aureport # 不使用参数显示整个摘要
```

```
Summary Report
=====
Range of time in logs: 08/08/2015 02:09:42.093 - 02/25/2016 17:0
1:01.950
Selected time for report: 08/08/2015 02:09:42 - 02/25/2016 17:01
:01.950
Number of changes in configuration: 299
Number of changes to accounts, groups, or roles: 18
Number of logins: 18
Number of failed logins: 3
Number of authentications: 30
Number of failed authentications: 3
Number of users: 3
Number of terminals: 7
Number of host names: 3
Number of executables: 15
Number of commands: 41
Number of files: 0
Number of AVC's: 0
Number of MAC events: 2
Number of failed syscalls: 0
Number of anomaly events: 2
Number of responses to anomaly events: 0
Number of crypto events: 74
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 407
Number of events: 1955
```

```
aureport -au # 显示身份验证类日志
```

```
Authentication Report
=====
# date time acct host term exe success event
=====
1. 08/08/2015 02:09:52 root ? ttyS0 /usr/bin/login yes 332
2. 08/08/2015 02:20:27 root ? ttyS0 /usr/bin/login yes 34
3. 08/17/2015 10:40:03 root ? ttyS0 /usr/bin/login yes 33
.....
.....
20. 02/23/2016 11:09:46 cent 10.0.0.20 ssh /usr/sbin/sshd yes 11
8
21. 02/23/2016 11:13:26 cent ? ttyS0 /usr/bin/login no 147
```

```
aureport -au --failed --summary # 显示身份验证失败类日志
```

```
Failed Authentication Summary Report
=====
total acct
=====
1 root
1 cent
```

```
aureport -m -i # 显示用户帐户修改类日志
```

```
Account Modifications Report
=====
# date time auid addr term exe acct success event
=====
1. 08/08/2015 02:10:21 root ? ttyS0 /usr/sbin/useradd cent no 34
2
2. 08/08/2015 02:19:25 root ? ? /usr/sbin/groupadd ? yes 370
3. 08/08/2015 02:19:26 root ? ? /usr/sbin/groupadd ? yes 371
.....
.....
17. 02/08/2016 11:12:41 root ? ? /usr/sbin/groupadd ntp no 45
18. 02/08/2016 11:12:41 root ? ? /usr/sbin/useradd ntp no 46
```

```
aureport -m -i --start this-month # 显示自本月以来用户帐户修改类日志
```

```
Account Modifications Report
=====
# date time auid addr term exe acct success event
=====
1. 02/08/2016 11:12:41 root ? ? /usr/sbin/groupadd ntp no 45
2. 02/08/2016 11:12:41 root ? ? /usr/sbin/useradd ntp no 46
```

```
aureport -x -i # 显示执行类日志
```

```
Executable Report
=====
# date time exe term host auid event
=====
1. 08/08/2015 02:09:42 /usr/lib/systemd/systemd ? ? unset 6
2. 08/08/2015 02:09:42 /usr/lib/systemd/systemd-update-utmp ? ?
unset 7
3. 08/08/2015 02:09:42 /usr/lib/systemd/systemd ? ? unset 8
.....
.....
1422. 02/23/2016 17:01:01 /usr/sbin/cron cron ? root 211
1423. 02/23/2016 17:01:01 /usr/sbin/cron cron ? root 212
```

```
aureport -x -i --start 02/07/2016 --end 02/21/2016 # 显示从2016/2/7  
到2016/2/21的执行类日志
```

```
Executable Report
=====
# date time exe term host auid event
=====
1. 02/08/2016 11:11:47 /usr/lib/systemd/systemd ? ? unset 5
2. 02/08/2016 11:11:47 /usr/lib/systemd/systemd-update-utmp ? ?
unset 6
3. 02/08/2016 11:11:47 /usr/lib/systemd/systemd ? ? unset 7
.....
.....
87. 02/08/2016 11:14:08 /usr/lib/systemd/systemd ? ? unset 92
88. 02/08/2016 11:14:08 /usr/lib/systemd/systemd ? ? unset 93
```

使用 `ausearch` 和 `aureport` 搜索并显示日志如下：

```
ausearch --node dlp.srv.world | aureport -au # 在 dlp.srv.world 上  
搜索并显示身份验证日志
```

### Authentication Report

```
=====  
# date time acct host term exe success event  
=====  
1. 02/25/2016 16:55:35 cent ? ttyS0 /usr/bin/su yes 103  
2. 02/25/2016 16:55:44 cent ? /dev/ttyS0 /usr/bin/sudo yes 107  
3. 02/26/2016 09:21:35 root ? ttyS0 /usr/bin/login yes 38  
4. 02/26/2016 09:50:32 root ? ttyS0 /usr/bin/login yes 38
```

```
ausearch -ui 1000 | aureport -x -i # 搜索并显示UserID : 1000的执行日志
```

### Executable Report

```
=====  
# date time exe term host auid event  
=====  
1. 02/23/2016 09:52:23 /usr/bin/sudo /dev/ttyS0 ? cent 49  
2. 02/23/2016 09:52:23 /usr/bin/sudo /dev/ttyS0 ? cent 50  
3. 02/23/2016 09:55:06 /usr/bin/su ttyS0 ? cent 80  
....  
....  
15. 02/26/2016 09:48:50 /usr/bin/sudo /dev/ttyS0 ? cent 52
```

### 附1.1.3.2.5. 添加审计规则

可以添加自己的审计规则。

例如，配置用于记录 `/etc/hosts` 的写入和属性更改的审计规则：

```
auditctl -l # 显示当前规则（默认情况下无规则）
```

```
No rules
```

```
auditctl -w /etc/hosts -p wa -k hosts_change # -p [r|w|x|a] (r=读，w=写，x=执行，a=属性) ; -k [words] :设置搜索日志的关键字  
auditctl -l
```

```
-w /etc/hosts -p wa -k hosts_change
```

当一些操作完成并且被新的审计规则检测到时，审计日志将记录如下：

```
ausearch -k hosts_change | aureport -f -i #按关键字搜索
```

```
File Report  
=====  
# date time file syscall success exe auid event  
=====  
1. 03/09/2016 19:48:32 /etc/hosts open yes /usr/bin/bash root 46  
2. 03/10/2016 20:37:52 /etc/hosts open yes /usr/bin/vi root 49  
3. 03/10/2016 20:37:52 /etc/hosts chmod yes /usr/bin/vi root 50  
4. 03/10/2016 20:38:35 /etc/hosts~ rename yes /usr/bin/vi cent 7  
1  
5. 03/10/2016 20:38:35 /etc/hosts ? yes ? cent 72  
6. 03/10/2016 20:38:35 /etc/hosts ? yes ? cent 70  
7. 03/10/2016 20:38:35 /etc/hosts open yes /usr/bin/vi cent 73  
8. 03/10/2016 20:38:35 /etc/hosts chmod yes /usr/bin/vi cent 74  
9. 03/10/2016 20:38:35 /etc/hosts setxattr yes /usr/bin/vi cent  
75
```

在重新启动系统后，由 `auditctl` 命令添加的规则不会保留，因此如果想持续保留，需要将它们添加到 `/etc/audit/rules.d` 下的文件中。可以向 `/etc/audit/rules.d` 中的任何文件添加规则，但扩展名应为“`.rules`”：

```
auditctl -l >> /etc/audit/rules.d/additional.rules # 输出当前规则  
到“additional.rules”
```

如果为 Audit Target 设置了一个目录，则所有文件都会以目录的方式递归定位：

```
auditctl -w /home/testdir/ -p r -k testdir_audit # 将审计规则读取  
到 /home/testdir/
```

```
auditctl -l
```

```
-w /home/testdir/ -p r -k testdir_audit
```

```
ausearch -k testdir_audit | aureport -f -i # 日志记录如下
```

```
File Report
=====
# date time file syscall success exe auid event
=====
1. 03/10/2016 19:50:28 /home/testdir getxattr no /usr/bin/ls cent 77
2. 03/10/2016 19:50:28 /home/testdir lgetxattr no /usr/bin/ls cent 76
3. 03/10/2016 19:50:28 /home/testdir getxattr no /usr/bin/ls cent 78
4. 03/10/2016 19:50:32 /home/testdir getxattr no /usr/bin/ls cent 81
5. 03/10/2016 19:50:32 /home/testdir openat yes /usr/bin/ls cent 82
6. 03/10/2016 19:50:32 /home/testdir lgetxattr no /usr/bin/ls cent 79
7. 03/10/2016 19:50:32 /home/testdir getxattr no /usr/bin/ls cent 80
8. 03/10/2016 19:50:32 /home/testdir/test.txt lgetxattr no /usr/bin/ls cent 83
9. 03/10/2016 19:50:32 /home/testdir/test.txt getxattr no /usr/bin/ls cent 84
10. 03/10/2016 19:50:32 /home/testdir/test.txt getxattr no /usr/bin/ls cent 85
11. 03/10/2016 19:50:32 /home/testdir/testdir02 lgetxattr no /usr/bin/ls cent 86
12. 03/10/2016 19:50:32 /home/testdir/testdir02 getxattr no /usr/bin/ls cent 87
13. 03/10/2016 19:50:53 /home/testdir/testdir02/test2.txt open yes /usr/bin/cat cent 89
```

例如，配置监控由UID超过1000的用户删除的文件的审计规则（对于下面的 `S` 选项，可以在安装 `yum install man-pages` 后，使用 `man syscalls` 确认所有系统调用）：

```
auditctl -a always,exit -S unlink,unlinkat -F 'auid>=1000' -F 'auid!=-1' -F key=delete_audit
```

```
auditctl -l
```

```
-a always,exit -S unlink,unlinkat -F auid>=1000 -F auid!=-1 -F k  
ey=delete_audit
```

```
ausearch -k delete_audit | aureport -f -i # 日志记录如下
```

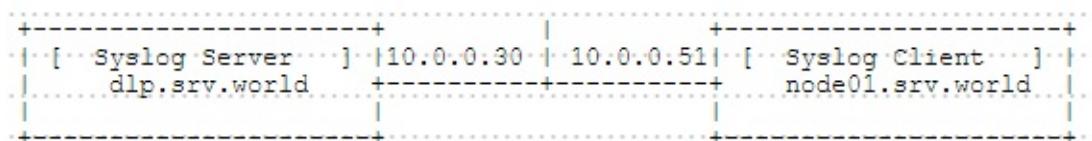
```
File Report  
=====  
# date time file syscall success exe auid event  
=====  
1. 03/10/2016 19:11:05 test.txt unlinkat yes /usr/bin/rm redhat  
112  
2. 03/10/2016 19:13:06 test3.txt unlinkat yes /usr/bin/rm cent 1  
39  
3. 03/10/2016 19:41:00 test2.txt unlinkat yes /usr/bin/rm redhat  
194
```

### 附1.1.3.3. Rsyslog

Rsyslog是一个[syslogd](#)的多线程增强版。

#### 附1.1.3.3.1. 输出日志到远程主机

本例基于以下环境：



配置日志管理服务器以从客户端服务器接收日志（Syslog Server）：

编辑 `/etc/rsyslog.conf` 文件：

```
# 取消注释  
$ModLoad imtcp  
$InputTCPServerRun 514  
# 指定允许访问的发送者  
$AllowedSender TCP, 127.0.0.1, 10.0.0.0/24, *.srv.world
```

```
systemctl restart rsyslog
```

配置客户端（Syslog Client）：

编辑 `/etc/rsyslog.conf` 文件：

```
# 例如，输出“authpriv.*”的日志到远程主机  
authpriv.*      @@dlp.srv.world:514  
  
# 取消注释  
$ActionQueueFileName fwdRule1 # unique name prefix for spool files  
$ActionQueueMaxDiskSpace 1g    # 1gb space limit (use as much as possible)  
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown  
$ActionQueueType LinkedList   # run asynchronously  
$ActionResumeRetryCount -1    # infinite retries if host is down
```

```
systemctl restart rsyslog
```

上述配置后，日志管理服务器上记录的各种身份验证记录如下：

```
tail -10 /var/log/secure
```

```
Jun 17 11:24:47 dlp sshd[9582]: Connection closed by 127.0.0.1 [preauth]
Jun 17 11:27:46 node01 login: pam_unix(login:session): session closed for user root
Jun 17 11:27:52 node01 login: pam_unix(login:auth): check pass; user unknown
Jun 17 11:27:52 node01 login: pam_unix(login:auth): authentication failure; logname=LOGIN uid=0 euid=0
Jun 17 11:27:54 node01 login: FAILED LOGIN 1 FROM ttyS0 FOR (unknown), User not known to the underlying
Jun 17 11:27:59 node01 login: pam_unix(login:session): session opened for user root by LOGIN(uid=0)
Jun 17 11:27:59 node01 login: DIALUP AT ttyS0 BY root
Jun 17 11:27:59 node01 login: ROOT LOGIN ON ttyS0
Jun 17 11:28:44 node01 su: pam_unix(su-l:session): session opened for user cent by root(uid=0)
Jun 17 11:28:54 node01 sudo: cent : TTY=ttyS0 ; PWD=/home/cent ; USER=root ; COMMAND=/bin/cat /etc/sha
```

如果要对每个主机，每个日期分隔日志，配置如下：

编辑 `/etc/rsyslog.conf` 文件：

```
# 添加：定义日志文件
$template Secure_log, "/var/log/secure.d/%fromhost%_%$year%$month%$day%.secure"

# 添加：指定上面定义的日志文件
authpriv.* -?Secure_log
```

```
systemctl restart rsyslog
```

```
ll /var/log/secure.d
```

```
total 8
-rw-r--r-- 1 root root 350 Jun 17 11:34 dlp_20150617.secure
-rw-r--r-- 1 root root 380 Jun 17 11:34 node01.srv.world_20150617.secure
```

### 附1.1.3.3.2. 输出日志到数据库

可以选择很多数据库，本例演示使用MariaDB配置，因此先[安装MariaDB](#)。

为Rsyslog创建一个用户和数据库：

```
yum -y install rsyslog-mysql
```

```
cat /usr/share/doc/rsyslog-mysql-*/createDB.sql | mysql -u root -p
```

Enter password:

```
mysql -u root -p
```

```
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 7291  
Server version: 5.5.41-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and oth  
ers.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current in  
put statement.  
  
# Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 7291  
Server version: 5.5.41-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and oth  
ers.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current in  
put statement.  
  
# 创建“rsyslog”用户并授予权限到Syslog数据库（在“password”部分设置任意密  
码）  
MariaDB [(none)]> grant all privileges on Syslog.* to rsyslog@'1  
ocalhost' identified by 'password';  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> exit  
Bye  
MariaDB [(none)]> grant all privileges on Syslog.* to rsyslog@'1  
ocalhost' identified by 'password';  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> exit  
Bye
```

配置Rsyslog将日志输出到数据库：

编辑 `/etc/rsyslog.conf` 文件：

```
# 添加
$ModLoad ommysql

# 例如，输出“authpriv.*”的日志。格式：:ommysql:主机,数据库,数据库用户,数据库密码
authpriv.*      :ommysql:localhost,Syslog,rsyslog,password
```

```
systemctl restart rsyslog
```

上述配置后，数据库中记录的一些日志记录如下：

```
mysql -u rsyslog -p Syslog
```

```
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7299
Server version: 5.5.41-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [Syslog]> show tables;
+-----+
| Tables_in_Syslog      |
+-----+
| SystemEvents          |
| SystemEventsProperties |
+-----+
2 rows in set (0.00 sec)

MariaDB [Syslog]> select ReceivedAt,Facility,Priority,FromHost,Message from SystemEvents;
```

| ReceivedAt                                     | Facility | Priority | FromHost | Message                                                |
|------------------------------------------------|----------|----------|----------|--------------------------------------------------------|
| 2015-06-17 19:40:33                            | 10       | 6        | dlp      | pam_uni                                                |
| x(login:session): session closed for user root |          |          |          |                                                        |
| 2015-06-17 19:40:39                            | 10       | 6        | dlp      | pam_uni                                                |
| x(login:session): session opened for user root |          |          |          |                                                        |
| 2015-06-17 19:40:39                            | 10       | 6        | dlp      | DIALUP                                                 |
| AT ttyS0 BY root                               |          |          |          |                                                        |
| 2015-06-17 19:40:39                            | 10       | 5        | dlp      | ROOT LOGIN ON ttyS0                                    |
| 2015-06-17 19:40:58                            | 10       | 6        | node01   | Accept                                                 |
| ed password for cent from 10.0.0.30 port 60492 |          |          |          |                                                        |
| 2015-06-17 19:40:58                            | 10       | 6        | node01   | pam_unix(sshd:session): session opened for user cent   |
| 2015-06-17 19:40:58                            | 10       | 6        | node01   | Received disconnect from 10.0.0.30: 11: disconnected   |
| 2015-06-17 19:40:58                            | 10       | 6        | node01   | pam_unix(sshd:session): session closed for user cent   |
| 2015-06-17 19:41:13                            | 10       | 6        | node01   | pam_unix(su-l:session): session opened for user cent   |
| 2015-06-17 19:41:23                            | 10       | 6        | dlp      | Invalid user cent from 10.0.0.51                       |
| 2015-06-17 19:41:23                            | 10       | 6        | dlp      | input_userauth_request: invalid user cent [preauth]    |
| 2015-06-17 19:41:27                            | 10       | 4        | dlp      | pam_unix(sshd:auth): check pass; user unknown          |
| 2015-06-17 19:41:27                            | 10       | 5        | dlp      | pam_unix(sshd:auth): authentication failure; logname=  |
| 2015-06-17 19:41:28                            | 10       | 6        | dlp      | Failed password for invalid user cent from 10.0.0.51   |
| 2015-06-17 19:41:29                            | 10       | 6        | dlp      | Connect ion closed by 10.0.0.51 [preauth]              |
| 2015-06-17 19:41:40                            | 10       | 6        | dlp      | Accepted password for root from 10.0.0.51 port 58750   |
| 2015-06-17 19:41:40                            | 10       | 6        | dlp      | pam_unix(sshd:session): session opened for user root b |
| 2015-06-17 19:41:42                            | 10       | 6        | dlp      | Received disconnect from 10.0.0.51: 11: disconnected b |

```
| 2015-06-17 19:41:42 |          10 |          6 | dlp      | pam_uni
x(sshd:session): session closed for user root  |
+-----+-----+-----+-----+
-----+
19 rows in set (0.00 sec)
```

### 附1.1.4. 入侵检测系统

#### 附1.1.4.1. AIDE

AIDE（Advanced Intrusion Detection Environment）是一个文件和目录完整性检查器，基于主机的IDS（Intrusion Detection System入侵检测系统）。

安装AIDE：

```
yum -y install aide
```

配置AIDE并初始化数据库。可以使用默认配置AIDE，如果要自定义设置，如下所示更改配置文件（设置规则写在26-84行附近，可参考修改）：

编辑 /etc/aide.conf 文件：

```
# 例如，更改监控/var/log的设置
/var/log    p+u+g+i+n+acl+selinux+xattrs
```

```
aide --init # 初始化数据库
```

```
AIDE, version 0.15.1

### AIDE database at /var/lib/aide/aide.db.new.gz initialized.
```

```
cp -p /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz # 将生成的数据库复制到主数据库
```

执行检查：

```
aide --check
```

```
# 如果没有不匹配，则显示“Okay”
AIDE, version 0.15.1

### All files match AIDE database. Looks okay!
```

尝试更改文件并重新检查：

```
chmod 640 /root/anaconda-ks.cfg

aide --check
```

```
# 检测到如下差异
AIDE 0.15.1 found differences between database and filesystem!!
Start timestamp: 2015-06-17 19:55:20

Summary:
    Total number of files:          39039
    Added files:                  0
    Removed files:                 0
    Changed files:                  1

-----
Changed files:
-----

changed: /root/anaconda-ks.cfg

-----
Detailed information about changes:
-----

File: /root/anaconda-ks.cfg
Perm      : -rw-----, -rw-r-----
Ctime     : 2015-05-24 02:22:04, 2015-06-19 11:55:
15
ACL       : old = A:
-----
user::rw-
group::::-
other::::

-----
D: <NONE>
new = A:

-----
user::rw-
group::r--
other::::

-----
D: <NONE>
```

如果即使检测到某些差异也没有问题，则更新数据库，如下所示：

```
aide --update
```

```
AIDE 0.15.1 found differences between database and filesystem!!
Start timestamp: 2015-06-17 19:56:31
```

Summary:

|                        |       |
|------------------------|-------|
| Total number of files: | 39039 |
| Added files:           | 0     |
| Removed files:         | 0     |
| Changed files:         | 1     |

-----  
Changed files:  
-----

```
changed: /root/anaconda-ks.cfg
```

```
....  
....
```

```
cp -p /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz # 更新
数据库
```

加入Cron以定期检查。每次执行，日志文件 `/var/log/aide/aide.log` 都会更新，如果没有区别，它将以零字节更新。因此如果要保存日志文件，则需要创建一个shell脚本或通过电子邮件发送结果或其他方法。以添加每日检查Crontab并通过电子邮件发送结果为例：

编辑 `/etc/cron.d/aide` 文件：

```
00 01 * * * /usr/sbin/aide --update | mail -s 'Daily Check by AIDE' root
```

### 附1.1.4.2. Tripwire

当服务器遭到黑客攻击时，在多数情况下，黑客可能对系统文件等等一些重要的文件进行修改。对此，使用Tripwire建立数据完整性监测系统。虽然它不能抵御黑客攻击以及黑客对一些重要文件的修改，但是可以监测文件是否被修改过以及哪些文件被修改过，从而在被攻击后有的放矢的策划出解决办法。Tripwire的原理是Tripwire被安装、配置后，将当前的系统数据状态建立成数据库，随着文件的添加、删除和修改等等变化，通过系统数据现状与不断更新的数据库进行比较，来判定哪些文件被添加、删除和修改过。正因为初始的数据库是在Tripwire本体被安装、配置后建立的原因，务必应该在服务器开放前，或者说操作系统刚被安装后用Tripwire构建数据完整性监测系统。

安装Tripwire：

```
yum --enablerepo=epel -y install tripwire # 从EPEL安装
```

创建密钥和数据库：

```
tripwire-setup-keyfiles # 生成密钥
```

```
....  
....  
Enter the site keyfile passphrase: # 设置site密钥文件密码  
Verify the site keyfile passphrase: # 确认  
....  
....  
Enter the local keyfile passphrase: # 设置本地密钥文件密码  
Verify the local keyfile passphrase: # 确认  
....  
....  
Please enter your site passphrase: # 输入site密钥文件密码  
....  
....  
Please enter your site passphrase: # 输入site密钥文件密码  
....  
....
```

```
cd /etc/tripwire
```

编辑 twcfg.txt 文件：

```
# 报告级别 (4最大)
REPORTLEVEL =4
```

```
twadmin -m F -c tw.cfg -S site.key twcfg.txt # 生成配置
```

```
Please enter your site passphrase: # 输入site密钥文件密码
Wrote configuration file: /etc/tripwire/tw.cfg
```

使用下面的脚本优化策略文件：

编辑 `twpolmake.pl` 文件：

```
#!/usr/bin/perl
# Tripwire Policy File customize tool
# -----
-- 
# Copyright (C) 2003 Hiroaki Izumi
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
# This program is distributed in the hope that it will be useful
',
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 021
11-1307, USA.
# -----
-- 
# Usage:
#     perl twpolmake.pl {Pol file}
# -----
-- 
#
$POLFILE=$ARGV[0];
```

```

open(POL,"$POLFILE") or die "open error: $POLFILE" ;
my($myhost,$thost) ;
my($sharp,$tpath,$cond) ;
my($INRULE) = 0 ;

while (<POL>) {
    chomp;
    if (($thost) = /^HOSTNAME\s*=\s*(.*)\s*/ ) {
        $myhost = `hostname` ; chomp($myhost) ;
        if ($thost ne $myhost) {
            $_="HOSTNAME=\"$myhost\";" ;
        }
    }
    elsif ( /{/ ) {
        $INRULE=1 ;
    }
    elsif ( /}/ ) {
        $INRULE=0 ;
    }
    elsif ($INRULE == 1 and ($sharp,$tpath,$cond) = /(\s*\#\?\s*)(\S+)\b(\s+->\s+.+)$/) {
        $ret = ($sharp =~ s/\#\//g) ;
        if ($tpath eq '/sbin/e2fsadm' ) {
            $cond =~ s/;\s+(tune2fs.*$)/;\ #$1/ ;
        }
        if (! -s $tpath) {
            $_ = "$sharp#$tpath$cond" if ($ret == 0) ;
        }
        else {
            $_ = "$sharp$tpath$cond" ;
        }
    }
    print "$_\n" ;
}
close(POL) ;

```

```
perl twpolmake.pl twpol.txt > twpol.txt.new
```

```
twadmin -m P -c tw.cfg -p tw.pol -S site.key twpol.txt.new
```

```
Please enter your site passphrase:  
Wrote policy file: /etc/tripwire/tw.pol
```

```
tripwire -m i -s -c tw.cfg # 创建数据库
```

```
Please enter your local passphrase:
```

手动执行检查 (Cron的每日检查脚本包含在软件包中) :

```
tripwire -m c -s -c /etc/tripwire/tw.cfg
```

### Open Source Tripwire(R) 2.4.2.2 Integrity Check Report

```
Report generated by: root  
Report created on: Fri 18 Jun 2015 19:53:39 PM JST  
Database last updated on: Never
```

```
=====
```

#### Report Summary:

```
=====
```

```
=====
```

```
Host name: dlp.srv.world  
Host IP address: 10.0.0.30  
Host ID: None  
Policy file used: /etc/tripwire/tw.pol  
Configuration file used: /etc/tripwire/tw.cfg  
Database file used: /var/lib/tripwire/dlp.srv.world.tw  
d  
Command line used: tripwire -m c -s -c /etc/tripwire/  
tw.cfg
```

```
=====
```

#### Rule Summary:

```
=====
```

```
=====
```

-----  
**Section: Unix File System**  
-----  
-----

| Rule Name<br>oved Modified                         | Severity Level | Added | Rem |
|----------------------------------------------------|----------------|-------|-----|
| User binaries<br>0                                 | 66             | 0     | 0   |
| Tripwire Binaries<br>0                             | 100            | 0     | 0   |
| Libraries<br>0                                     | 66             | 0     | 0   |
| File System and Disk Administrataton Programs<br>0 | 100            | 0     | 0   |
| Kernel Administration Programs<br>0                | 100            | 0     | 0   |
| Networking Programs<br>0                           | 100            | 0     | 0   |
| System Administration Programs<br>0                | 100            | 0     | 0   |
| Hardware and Device Control Programs<br>0          | 100            | 0     | 0   |
| System Information Programs<br>0                   | 100            | 0     | 0   |
| Application Information Programs<br>0              | 100            | 0     | 0   |
| (/sbin/rtmon)                                      |                |       |     |
| Operating System Utilities<br>0                    | 100            | 0     | 0   |
| Critical Utility Sym-Links<br>0                    | 100            | 0     | 0   |
| Shell Binaries<br>0                                | 100            | 0     | 0   |
| Critical system boot files<br>0                    | 100            | 0     | 0   |
| * Tripwire Data Files                              | 100            | 1     | 0   |

|                              |     |   |   |
|------------------------------|-----|---|---|
| 0                            |     |   |   |
| System boot changes          | 100 | 0 | 0 |
| 0                            |     |   |   |
| OS executables and libraries | 100 | 0 | 0 |
| 0                            |     |   |   |
| Critical configuration files | 100 | 0 | 0 |
| 0                            |     |   |   |
| Security Control             | 100 | 0 | 0 |
| 0                            |     |   |   |
| Login Scripts                | 100 | 0 | 0 |
| 0                            |     |   |   |
| Root config files            | 100 | 0 | 0 |
| 0                            |     |   |   |
| Invariant Directories        | 66  | 0 | 0 |
| 0                            |     |   |   |
| Temporary directories        | 33  | 0 | 0 |
| 0                            |     |   |   |
| Critical devices             | 100 | 0 | 0 |
| 0                            |     |   |   |
| (/proc/kcore)                |     |   |   |

Total objects scanned: 21739

Total violations found: 1

=====

=====

Object Summary:

=====

=====

-----

-----

# Section: Unix File System

-----

-----

-----

-----

Rule Name: Tripwire Data Files (/var/lib/tripwire)

Severity Level: 100

```
Added:  
"/var/lib/tripwire/dlp.srv.world.twd"  
  
=====  
=====  
Error Report:  
=====  
=====  
  
No Errors  
  
-----  
-----  
*** End of report ***  
  
Open Source Tripwire 2.4 Portions copyright 2000 Tripwire, Inc.  
Tripwire is a registered  
trademark of Tripwire, Inc. This software comes with ABSOLUTELY  
NO WARRANTY;  
for details use --version. This is free software which may be re  
distributed  
or modified only under certain conditions; see COPYING for detai  
ls.  
All rights reserved.
```

如果即使检测到某些差异也没有问题，则如下更新数据库：

结果保存在下面的目录下：

```
ll /var/lib/tripwire/report
```

```
total 8  
-rw-r--r-- 1 root root 6814 Jun 17 19:53 dlp.srv.world-20150617-  
125339.twr
```

使用特定报告更新数据库：

```
tripwire -m u -a -s -c /etc/tripwire/tw.cfg \  
-r /var/lib/tripwire/report/dlp.srv.world-20150617-125339.twr
```

Please enter your local passphrase:

## 附1.2. 加入Windows活动目录

本教程需要局域网中的Windows活动目录域服务。

本例演示在下面的环境中进行配置：

```
Domain Server : Windows Server 2012 R2  
Domain Name   : FD3S01  
Realm         : SRV.WORLD  
Hostname      : fd3s.srv.world
```

安装一些所需的软件包：

```
yum -y install realmd sssd oddjob oddjob-mkhomedir adcli samba-common
```

加入Windows活动目录域：

将DNS更改为AD的：

```
nmcli c modify ens3 ipv4.dns 10.0.0.100
```

```
nmcli c down ens3; nmcli c up ens3
```

```
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/1)
```

发现活动目录域：

```
realm discover SRV.WORLD
```

```
srv.world
  type: kerberos
  realm-name: SRV.WORLD
  domain-name: srv.world
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common
```

加入活动目录域：

```
realm join SRV.WORLD
```

```
Password for Administrator: # AD管理员密码
```

确认是否可以获得AD用户信息：

```
id FD3S01\\Serverworld
```

```
uid=406801001(serverworld@srv.world) gid=406800513(domain users@
srv.world) groups=406800513(domain users@srv.world)
```

确认是否可以切换到AD用户：

```
su - FD3S01\\Serverworld
```

```
Creating home directory for serverworld@srv.world.
[serverworld@srv.world@dlp ~]$ # 已切换
```

如果想要AD用户省略域名，如下配置：

编辑 /etc/sssd/sssd.conf 文件：

```
# 更改  
use_fully_qualified_names = False
```

```
systemctl restart sssd
```

```
id Administrator
```

```
uid=406800500(administrator) gid=406800513(domain users) groups=  
406800513(domain users),  
406800572(denied rodc password replication group),406800518(sche  
ma admins),  
406800520(group policy creator owners),406800512(domain admins),  
406800519(enterprise admins)
```

## 附1.3. 访问控制

### 附1.3.1. ACL

这是配置[ACL（Access Control Lists访问控制列表）](#)的示例。

CentOS默认安装ACL，如果没有，运行以下命令安装：

```
yum -y install acl
```

如果使用CentOS7上默认的[XFS文件系统](#)，则不必设置预设置来使用ACL功能。如果使用CentOS6上默认的ext4文件系统，则需要设置预设置才能使用ACL功能。以下为设置预设置：

对于CentOS6，在初始操作系统安装时设置的设备上，ACL选项已经通过默认挂载选项启用：

```
tune2fs -l /dev/VolGroup/lv_root | grep "Default mount options" #  
显示默认挂载选项
```

```
Default mount options: user_xattr acl # ACL选项已添加
```

对于在操作系统安装后添加的设备（如添加HDD等）的情况，需要手动启用ACL选项。一种方法是使用ACL选项挂载设备，另一种方法是在默认挂载选项中添加ACL选项：

使用ACL选项来挂载以启用ACL：

```
mount -o acl /dev/sdb1 /mnt
```

```
mount | grep sdb1
```

```
/dev/sdb1 on /mnt type ext4 (rw,acl)
```

或将ACL选项添加到默认挂载选项：

```
tune2fs -o acl /dev/sdb1
```

```
tune2fs -l /dev/vdb1 | grep "Default mount options"
```

```
Default mount options: acl
```

预设置完成后，关于如何设置ACL，以为文件 `/home/test.txt` 设置ACL为例：

```
ll /home/test.txt
```

```
-rwx----- 1 root root 10 Jul 3 16:17 /home/test.txt
```

```
setfacl -m u:cent:r /home/test.txt # 为用户“cent”设置 /home/test.txt 为r(读)
```

```
ll /home/test.txt # 设置ACL后，在属性添加了“+”
```

```
-rwxr-----+ 1 root root 10 Jul 3 16:17 /home/test.txt
```

```
getfacl /home/test.txt # 确认设置
```

```
getfacl: Removing leading '/' from absolute path names
# file: home/test.txt
# owner: root
# group: root
user::rwx
user:cent:r--
group::---
mask::r--
other::---
```

以用户“cent”尝试访问：

```
cat /home/test.txt
```

```
ACL test file # 正常读取
```

以其他用户尝试访问：

```
cat /home/test.txt
```

```
cat: /home/test.txt: Permission denied # 不能正常读取
```

递归设置ACL到目录：

```
setfacl -R -m u:cent:r /home/testdir # 为用户“cent”设置 /home/testdir 为r（读）
```

```
ll /home/testdir
```

```
total 4  
-rwxr----+ 1 root root 5 Jul 3 16:23 testfile
```

```
getfacl -R /home/testdir
```

```
getfacl: Removing leading '/' from absolute path names  
# file: home/testdir  
# owner: root  
# group: root  
user::rwx  
user:cent:r--  
group::---  
mask::r--  
other::---  
  
# file: home/testdir/testfile  
# owner: root  
# group: root  
user::rwx  
user:cent:r--  
group::---  
mask::r--  
other::---
```

按组设置ACL：

```
setfacl -m g:security:rw /home/test.txt # 为组“security”设置 /home/test.txt 为rw（读/写）
```

```
getfacl /home/test.txt
```

```
getfacl: Removing leading '/' from absolute path names
# file: home/test.txt
# owner: root
# group: root
user::rwx
user:cent:r--
group::---
group:security:rw-
mask::rw-
other::---
```

以组“**security**”中的用户“**cent**”尝试访问：

```
echo "test write" >> /home/test.txt
cat /home/test.txt
```

```
ACL test file
test write # 正常写入
```

以不是组“**security**”中的用户尝试访问：

```
echo "test write" >> /home/test.txt
-bash: /home/test.txt: Permission denied # 不能正常写入
```

移除ACL：

```
setfacl -b /home/test.txt # 从 /home/test.txt 移除ACL
setfacl -x u:fedora /home/test.txt # 仅为用
户“fedora”在 /home/test.txt 移除ACL
```

为目录设置默认ACL。如果在设置了默认ACL的目录下创建文件/目录，默认访问属性是继承的。但要注意，如果使用 `chmod` 更改属性，则ACL将无效：

```
setfacl -m u:cent:r-x /home/testdir
```

为用户“**cent**”设置 `/home/testdir` 默认ACL为r-x（读/执行）：

```
setfacl -d -m u:cent:r-x /home/testdir
```

```
getfacl /home/testdir
```

```
getfacl: Removing leading '/' from absolute path names
# file: home/testdir
# owner: root
# group: root
user::rwx
user:cent:r-x
group::---
mask::r-x
other::---
default:user::rwx
default:user:cent:r-x
default:group::---
default:mask::r-x
default:other::---
```

```
echo "ACL default setting" > /home/testdir/test.txt
```

```
ll /home/testdir/test.txt
```

```
-rw-r-----+ 1 root root 20 Jan 31 22:32 /home/testdir/test.txt
```

以用户“cent”尝试访问：

```
cat /home/testdir/test.txt
```

```
ACL default setting # 正常读取
```

移除默认ACL：

```
setfacl -k /home/testdir
```

```
getfacl /home/testdir
```

```
getfacl: Removing leading '/' from absolute path names
# file: home/testdir
# owner: root
# group: root
user::rwx
user:cent:r-x
group::---
mask::r-x
other::---
```

从配置文件设置ACL：

编辑 `acl.txt` 文件，创建ACL的配置文件（如果想在其他系统上设置ACL，可以使用 `getfacl` 命令导出）：

```
# file: /home/testdir
# owner: root
# group: root
user::rwx
user:cent:r-x
group::---
mask::r-x
other::---

# file: /home/test.txt
# owner: root
# group: root
user::rwx
user:cent:r--
group::---
mask::r--
other::---
```

```
setfacl --restore=acl.txt
```

```
ll /home
```

```
total 16
drwx----- 2 cent    cent    4096 Jan 31 12:14 cent
drwx----- 2 fedora  fedora  4096 Jan 31 12:14 fedora
drwxr-x---+ 2 root    root    4096 Jan 31 22:32 testdir
-rw xr-----+ 1 root    root     25 Jan 31 21:56 test.txt
```

## 附1.3.2. TCP Wrapper

这是[TCP Wrapper](#)的TCP访问控制示例。

安装TCP Wrapper：

```
yum -y install tcp_wrappers
```

使用以下命令确认服务是否可以在TCP Wrapper控制下。如果包含 libwrap 的链接，表示可以：

```
ldd /usr/sbin/sshd | grep wrap
```

```
libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f01b4e2a000) # 这个
服务可以在TCP Wrapper控制下，因为它包含'libwrap'
```

TCP Wrapper的访问控制配置在 /etc/hosts.allow 和 /etc/hosts.deny 中。

下面示例演示设置允许从 10.0.0.0/24 访问sshd的配置：

编辑 /etc/hosts.deny 文件：

```
sshd: ALL
```

编辑 /etc/hosts.allow 文件：

```
sshd: 10.0.0.
```

对于允许从 host.example.domain 访问vsftpd的情况：

编辑 /etc/hosts.deny 文件：

```
vsftpd: ALL
```

编辑 `/etc/hosts.allow` 文件：

```
vsftpd: host.example.domain
```

下面这种情况，只允许从 `example.domain` 和 `10.0.1.0/24` 访问所有在TCP Wrapper控制下的服务：

编辑 `/etc/hosts.deny` 文件：

```
ALL: ALL
```

编辑 `/etc/hosts.allow` 文件：

```
ALL: .example.domain 10.0.1.
```

### 附1.3.3. SELinux

SELinux（Security-Enhanced Linux安全增强式Linux）是美国国家安全局（NSA）对于强制访问控制的实现，是Linux历史上最杰出的新安全子系统。红帽上的SELinux用户和管理员指南。

#### 附1.3.3.1. 运行模式

可以通过SELinux在CentOS上使用MAC（Mandatory Access Control强制访问控制）功能来获取各种资源。

如下确认SELinux的当前状态（默认模式为“Enforcing”）：

```
getenforce # 显示当前模式
```

```
Enforcing
```

- enforcing -> SELinux已启用（默认）
- permissive -> MAC未启用，但仅根据策略记录审计日志

- disabled -> SELinux已禁用

也可以使用下面命令显示 (“Current mode”行) :

```
sestatus
```

|                             |                 |
|-----------------------------|-----------------|
| SELinux status:             | enabled         |
| SELinuxfs mount:            | /sys/fs/selinux |
| SELinux root directory:     | /etc/selinux    |
| Loaded policy name:         | targeted        |
| Current mode:               | enforcing       |
| Mode from config file:      | enforcing       |
| Policy MLS status:          | enabled         |
| Policy deny_unknown status: | allowed         |
| Max kernel policy version:  | 28              |

可以通过 `setenforce` 命令在“permissive”和“enforcing”之间切换当前模式。但是如果系统重新启动，则模式将回到默认状态：

```
getenforce
```

Enforcing

```
setenforce 0 # 切换到“Permissive”
```

```
getenforce
```

Permissive

```
setenforce 1 # 切换到“Enforcing”
```

```
getenforce
```

Enforcing

如果要永久更改运行模式，更改配置文件中的值：

编辑 `/etc/selinux/config` 文件：

```
# This file controls the state of SELinux on the system.  
# SELINUX= can take one of these three values:  
#       enforcing - SELinux security policy is enforced.  
#       permissive - SELinux prints warnings instead of enforcing.  
#       disabled - No SELinux policy is loaded.  
# 更改为要设置的值  
SELINUX=enforcing  
# SELINUXTYPE= can take one of these two values:  
#       targeted - Targeted processes are protected,  
#       minimum - Modification of targeted policy. Only selected p  
rocesses are protected.  
#       mls - Multi Level Security protection.  
SELINUXTYPE=targeted
```

```
reboot # 重启系统后生效
```

如果将运行模式从“Disabled”更改为“Enforcing/Permissive”，则需要使用SELinux上下文重新标记文件系统。因为当在“Disabled”模式下创建一些文件或目录时，它们没有使用SELinux上下文标记，需要对它们进行标记：

```
touch /.autorelabel # 如下设置重新标记，在下一次系统重新启动时完成  
reboot
```

### 附1.3.3.2. 策略类型

如果SELinux处于“Enforcing/Permissive”状态，可以选择策略类型。如果需要，可以根据自己的环境修改所选策略。

可以在 /etc/selinux/config 文件中设置策略类型。CentOS7默认策略是“targeted”策略。但是，如果要更改策略类型，需要安装策略文件。对于最小化安装的CentOS7，默认仅安装“targeted”策略。

如果在不安装策略文件的情况下更改策略，系统将无法启动，请小心。

```
cat /etc/selinux/config
```

```
# This file controls the state of SELinux on the system.  
# SELINUX= can take one of these three values:  
#       enforcing - SELinux security policy is enforced.  
#       permissive - SELinux prints warnings instead of enforcing.  
#       disabled - No SELinux policy is loaded.  
SELINUX=enforcing  
# SELINUXTYPE= can take one of these two values:  
#       targeted - Targeted processes are protected,  
#       minimum - Modification of targeted policy. Only selected p  
rocesses are protected.  
#       mls - Multi Level Security protection.  
# 默认为“targeted”  
SELINUXTYPE=targeted
```

例如，改为“minimum”策略：

首先安装策略文件：

```
yum -y install selinux-policy-minimum
```

策略文件安装在 `minimum` 目录下：

```
ll /etc/selinux
```

```
total 16  
-rw-r--r--. 1 root root 547 Mar 18 16:23 config  
drwxr-xr-x. 6 root root 4096 Mar 18 17:26 minimum  
-rw-r--r--. 1 root root 2321 Nov 20 16:04 semanage.conf  
drwxr-xr-x. 6 root root 4096 Mar 18 16:24 targeted
```

编辑 `/etc/selinux/config` 文件：

```
# 更改“SELINUXTYPE”部门
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.

SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected p
rocesses are protected.
#       mls - Multi Level Security protection.

SELINUXTYPE=minimum
```

`reboot # 重启系统后生效`

`sestatus`

|                             |                 |
|-----------------------------|-----------------|
| SELinux status:             | enabled         |
| SELinuxfs mount:            | /sys/fs/selinux |
| SELinux root directory:     | /etc/selinux    |
| Loaded policy name:         | minimum # 已经变更  |
| Current mode:               | enforcing       |
| Mode from config file:      | enforcing       |
| Policy MLS status:          | enabled         |
| Policy deny_unknown status: | allowed         |
| Max kernel policy version:  | 28              |

配置文件中的示例包括RPM包提供的三种策略：

| 策略       | 描述                                                                       |
|----------|--------------------------------------------------------------------------|
| Targeted | 此策略将访问控制应用于通常的攻击目标进程（默认）                                                 |
| Minimum  | 此策略包含的设置文件与“Targeted”策略相同，但是与“Targeted”策略相比，访问控制的目标进程少得多                 |
| MLS      | Multilevel Security Policy 多级安全策略。它实现了Bell-LaPadula (BLP) 模型，并可能应用更复杂的控件 |

### 附1.3.3.3. SELinux上下文

对文件或目录的访问控制由称为SELinux Context（SELinux上下文）的附加信息控制。

SELinux上下文有以下语法： [SELinux User]:[Role]:[Type]:[Level]

- SELinux User：SELinux用户属性。每个Linux用户都通过SELinux策略映射到SELinux用户。
- Role：RBAC（Role Based Access Control基于角色的访问控制）属性。它定义了SELinux用户的角色，控制哪些定义的角色可以通过SELinux策略访问域（Domain）。
- Type：TE（Type Enforcement）属性。它定义了进程的域（Domain），并定义了文件类型。
- Level：MLS（Multi Level Security多级安全）和MCS（Multi Category Security多类别安全）属性。Level的语法： [sensitivity]:[category]
  - RHEL/CentOS默认的“targeted”策略强制MCS，此策略sensitivity只使用“s0”，category支持“c0-c1023”。
  - MLS强制Bell-La Padula强制访问模型。如果想使用它，则需要在RHEL/CentOS上安装MLS策略软件包。但它不支持X Window系统，所以不能在桌面环境中使用。

要显示文件或进程的SELinux上下文，在命令中添加 Z 选项：

```
ls -Z /root # 文件/目录
```

| User      | : Role | :     | Type                              | :     | Level           |
|-----------|--------|-------|-----------------------------------|-------|-----------------|
| -----     | -----  | ----- | -----                             | ----- | -----           |
| -rw-----. | root   | root  | system_u:object_r:admin_home_t:s0 |       | anaconda-ks.cfg |

```
ps axZ # 进程
```

| LABEL                                 | PID TTY | STAT | TIME | COMMA                 |
|---------------------------------------|---------|------|------|-----------------------|
| ND                                    |         |      |      |                       |
| system_u:system_r:init_t:s0           | 1 ?     | Ss   | 0:01 | /usr/lib/systemd/syst |
| system_u:system_r:kernel_t:s0         | 2 ?     | S    | 0:00 | [kthr eadd]           |
| system_u:system_r:kernel_t:s0         | 3 ?     | S    | 0:00 | [ksof tirqd/0]        |
| .....                                 |         |      |      |                       |
| .....                                 |         |      |      |                       |
| system_u:system_r:postfix_master_t:s0 | 916 ?   | Ss   | 0:00 | /usr/libexec/postfix/ |
| system_u:system_r:postfix_pickup_t:s0 | 917 ?   | S    | 0:00 | pickup -l -t unix -u  |
| system_u:system_r:postfix_qmgr_t:s0   | 918 ?   | S    | 0:00 | qmgr -l -t unix -u    |
| system_u:system_r:kernel_t:s0         | 941 ?   | S<   | 0:00 | [kwor ker/1:1H]       |
| system_u:system_r:kernel_t:s0         | 966 ?   | S<   | 0:00 | [kwor ker/0:1H]       |
| system_u:system_r:kernel_t:s0         | 1246 ?  | S<   | 0:00 | [kwor ker/0:2H]       |

```
id -Z # 自己的ID
```

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

```
yum -y install policycoreutils-python # 如果 semanage 命令不存在，则运行安装
```

每个Linux用户都通过SELinux策略映射到SELinux用户，可以显示如下的映射列表：

```
semanage login -l
```

| Login Name<br>service | SELinux User | MLS/MCS Range  | S |
|-----------------------|--------------|----------------|---|
| __default__           | unconfined_u | s0-s0:c0.c1023 | * |
| root                  | unconfined_u | s0-s0:c0.c1023 | * |
| system_u              | system_u     | s0-s0:c0.c1023 | * |

对于上面的示例（RHEL/CentOS默认）：“root”映射到“unconfined\_u”；“bin”或“daemon”等系统用户映射到“system\_u”；其他普通用户映射到“\_\_ default”一次，最后映射到“unconfined\_u”。

“unconfined\_u”用户被分配“unconfined\_r”角色，由“unconfined\_u”用户启动的进程以“unconfined\_t”域运行。

“unconfined\_t”域分配的进程不受SELinux控制。

```
ps axZ | grep unconfined_t
```

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1435 ttys0
  Ss  0:00 -bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1556 ttys0
  R+  0:00 ps axZ
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1557 ttys0
  S+  0:00 grep --color=auto unconfined_t
```

#### 附1.3.3.4. 更改布尔值

在SELinux策略提供的RPM软件包如“targeted”，可以轻松地更改SELinux设置来切换布尔值。

下例是基于“targeted”策略环境。

可以如下使用布尔值：

```
getsebool -a # 显示列表和当前设置
```

```
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
.....
.....
zoneminder_anon_write --> off
zoneminder_run_sudo --> off
```

```
semanage boolean -l # 同时显示描述
```

| SELinux boolean     | State       | Default | Description                  |
|---------------------|-------------|---------|------------------------------|
| ftp_home_dir        | (off , off) |         | Allow ftp to home dir        |
| smartmon_3ware      | (off , off) |         | Allow smartmon to 3ware      |
| mpd_enable_homedirs | (off , off) |         | Allow mpd to enable homedirs |
| ....                |             |         |                              |
| ....                |             |         |                              |
| cron_can_relabel    | (off , off) |         | Allow cron to can relabel    |
| sftpd_anon_write    | (off , off) |         | Allow sftpd to anon write    |

例如，配置“`samba_enable_home_dirs`”布尔值。

“`samba_enable_home_dirs`”默认设置为“`off`”，表示启用了SELinux的访问控制。

如果像这样配置了Samba完全访问共享文件夹，是不可能访问它，SELinux拒绝，因为正确的SELinux上下文没有分配给文件夹。

```
semanage boolean -l | grep samba_enable_home_dirs # 默认设置为“off”
```

|                                     |             |                                 |
|-------------------------------------|-------------|---------------------------------|
| <code>samba_enable_home_dirs</code> | (off , off) | Allow samba to enable home dirs |
|-------------------------------------|-------------|---------------------------------|

设置完全访问的共享文件夹后，创建一些测试文件（SELinux上下文从`/home/share`目录继承）：

```
ls -Z /home/share
```

```
-rw-rw-r--. cent cent unconfined_u:object_r:home_root_t:s0 test2.txt  
-rw-r--r--. root root unconfined_u:object_r:home_root_t:s0 test.txt
```

访问被拒绝如下，即使文件具有读取权限，父目录具有777权限：



将“`samba_enable_home_dirs`”的布尔值更改为“on”，以便能够正常访问文件夹：

```
setsebool -P samba_enable_home_dirs on
```

```
getsebool samba_enable_home_dirs
```

```
samba_enable_home_dirs --> on # 已变更
```

```
ls -Z /home/share # “samba_enable_home_dirs”为“off”时 SELinux上下文被添加
```

```
-rw-rw-r--. cent cent unconfined_u:object_r:home_root_t:s0 test2.txt  
-rw-r--r--. root root unconfined_u:object_r:home_root_t:s0 test.txt
```

```
restorecon -R /home/share # 恢复“samba_enable_home_dirs”的默认  
SELinux上下文
```

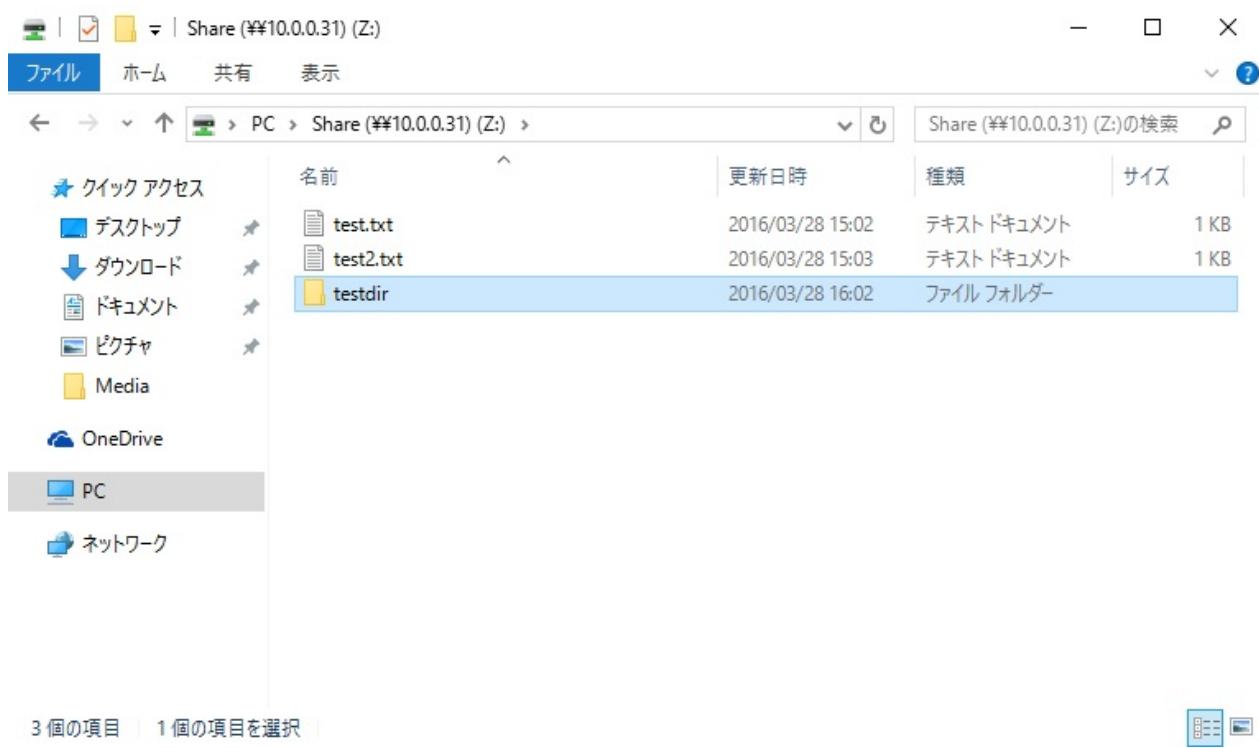
```
ls -Z /home/share # 显示SELinux上下文（更改为user_home_t）
```

```
-rw-rw-r--. cent cent unconfined_u:object_r:user_home_t:s0 test2.txt  
-rw-r--r--. root root unconfined_u:object_r:user_home_t:s0 test.txt
```

所有配置完成后，可以如下所示访问文件夹：



## 附1.3. 访问控制



### 附1.3.3.5. 更改文件类型

可以修改访问控制设置以更改文件类型而不更改布尔值。

下例是基于“targeted”策略环境。

默认SELinux上下文的设置放在 [policy directory]/contexts/files 下面，如下所示：

```
ll /etc/selinux/targeted/contexts/files
```

```

total 2104
-rw-r--r--. 1 root root 368879 Mar 28 15:46 file_contexts
-rw-----. 1 root root 1336352 Mar 28 15:46 file_contexts.bin
-rw-r--r--. 1 root root 13169 Mar 28 15:46 file_contexts.homed
irs
-rw-----. 1 root root 43960 Mar 28 15:46 file_contexts.homed
irs.bin
-rw-r--r--. 1 root root 0 Feb 17 02:24 file_contexts.local
-rw-----. 1 root root 16 Mar 28 15:46 file_contexts.local
.bin
-rw-r--r--. 1 root root 365908 Oct 21 11:19 file_contexts.pre
-rw-r--r--. 1 root root 0 Feb 17 02:24 file_contexts.subs
-rw-r--r--. 1 root root 422 Feb 17 02:24 file_contexts.subs_
dist
-rw-r--r--. 1 root root 139 Feb 17 02:24 media

```

```
head /etc/selinux/targeted/contexts/files/file_contexts
```

```

/.*      system_u:object_r:default_t:s0
/[^/]+ --      system_u:object_r:etc_runtime_t:s0
/a?quota\.(user|group) --      system_u:object_r:quota_db_t:s0
/nsr(/.*)?      system_u:object_r:var_t:s0
/sys(/.*)?      system_u:object_r:sysfs_t:s0
/xen(/.*)?      system_u:object_r:xen_image_t:s0
/mnt(/[^/]*)? -l      system_u:object_r:mnt_t:s0
/mnt(/[^/]*)? -d      system_u:object_r:mnt_t:s0
/bin/.* system_u:object_r:bin_t:s0
/dev/.* system_u:object_r:device_t:s0

```

例如，修改文件类型，以便在httpd上使用CGI：

在httpd上使用CGI的布尔值默认设置为“on”，因此可以使用默认SELinux设置在httpd设置上的默认目录 /var/www/cgi-bin/ 下运行CGI：

```
semanage boolean -l | grep httpd_enable_cgi
```

|                  |                                     |
|------------------|-------------------------------------|
| httpd_enable_cgi | (on , on) Allow httpd to enable cgi |
|------------------|-------------------------------------|

```
grep "cgi" /etc/selinux/targeted-contexts/files/file_contexts |  
grep "httpd"
```

```
/usr/.*\cgi -- system_u:object_r:httpd_sys_script_exec_t:s0  
/opt/.*\cgi -- system_u:object_r:httpd_sys_script_exec_t:s0  
/var/www/[^\/*]*/cgi-bin(/.*)? system_u:object_r:httpd_sys_script_exec_t:s0  
/var/www/html/[^\/*]*/cgi-bin(/.*)? system_u:object_r:httpd_sys_script_exec_t:s0  
/usr/lib/cgi-bin(/.*)? system_u:object_r:httpd_sys_script_exec_t:s0  
/var/www/cgi-bin(/.*)? system_u:object_r:httpd_sys_script_exec_t:s0  
/usr/lib/cgi-bin/(nph-)?cgiwrap(d)? -- system_u:object_r:httpd_suexec_exec_t:s0  
/var/log/cgiwrap\.\log.* -- system_u:object_r:httpd_log_t:s0
```

```
curl http://localhost/cgi-bin/index.py # 创建测试脚本并访问它，可以  
访问了
```

#### CGI Test Page

但是，如果要在[另一个目录中使用CGI](#)，即使httpd设置正确，访问也被拒绝：

```
curl http://localhost/cgi-enabled/index.py
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
<title>500 Internal Server Error</title>  
</head><body>  
<h1>Internal Server Error</h1>  
<p>The server encountered an internal error or  
misconfiguration and was unable to complete  
your request.</p>  
.....  
.....
```

```
ls -Z /var/www/html/cgi-enabled # "httpd_sys_content_t"被分配  
-rwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:  
s0 index.py
```

在这种情况下，需要将文件类型更改为SELinux允许CGI的文件类型。

如下更改文件类型（注意：使用 chcon 命令做的更改会在使用 restorecon 命令或是重新标记文件系统后恢复）：

```
chcon -t httpd_sys_script_exec_t /var/www/html/cgi-  
enabled/index.py
```

```
ls -Z /var/www/html/cgi-enabled
```

```
-rwxr-xr-x. root root unconfined_u:object_r:httpd_sys_script_ex-  
e_c_t:s0 index.py
```

```
curl http://localhost/cgi-enabled/index.py
```

```
CGI Test Page # 访问成功
```

如果想永久更改类型，按如下所示进行设置：

```
semanage fcontext -a -t httpd_sys_script_exec_t /var/www/html/cgi-  
enabled/index.py
```

```
grep "cgi-enabled"  
/etc/selinux/targeted-contexts/files/file_contexts.local
```

```
# 写为了默认上下文  
/var/www/html/cgi-enabled/index.py    system_u:object_r:httpd_sy-  
s_script_exec_t:s0
```

```
ls -Z /var/www/html/cgi-enabled
```

```
-rwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:  
s0 index.py
```

```
restorecon /var/www/html/cgi-enabled/index.py # 用 restotecon 重置
```

```
ls -Z /var/www/html/cgi-enabled
```

# 已恢复

```
-rwxr-xr-x. root root unconfined_u:object_r:httpd_sys_script_exec_t:s0 index.py
```

```
curl http://localhost/cgi-enabled/index.py
```

CGI Test Page # 访问成功

#### 附1.3.3.6. 更改端口类型

SELinux标记类型（Type）到网络端口，因此无法使用未配置类型的端口启动服务。

如下所示，显示网络端口的类型列表：

```
semanage port -l
```

| SELinux Port Type       | Proto | Port Number |
|-------------------------|-------|-------------|
| afs3_callback_port_t    | tcp   | 7001        |
| afs3_callback_port_t    | udp   | 7001        |
| afs_bos_port_t          | udp   | 7007        |
| ....                    |       |             |
| ....                    |       |             |
| zookeeper_leader_port_t | tcp   | 2888        |
| zope_port_t             | tcp   | 8021        |

例如给httpd设置82端口（http的默认端口80，https的端口443如下所示被标记为“http\_port\_t”，当然没有设置82。所以如果使用“listen 82”正确配置 httpd.conf，httpd将不会启动，因为SELinux拒绝了。如果想使用82，将其添加到“http\_port\_t”。）：

```
semanage port -l | grep -E -w "80|443" # 显示当前设置
```

```
http_port_t          tcp      80, 81, 443, 488, 8008,  
8009, 8443, 9000
```

```
semanage port -a -t http_port_t -p tcp 82 # 添加82端口
```

```
semanage port -l | grep "^http_port_t"
```

```
# 已添加  
http_port_t          tcp      82, 80, 81, 443, 488, 80  
08, 8009, 8443, 9000
```

```
ss -napt # 更改 httpd.conf 后，重新启动httpd并验证运行
```

```
# httpd侦听82端口  
State      Recv-Q Send-Q Local Address:Port           Peer A  
           Address:Port  
LISTEN      0      50      *:3306                  *:  
           users:(("mysqld",pid=1081,fd=14))  
LISTEN      0      50      *:139                  *:  
           users:(("smbd",pid=867,fd=38))  
LISTEN      0     128      *:22                  *:  
           users:(("sshd",pid=821,fd=3))  
LISTEN      0     100    127.0.0.1:25            *:  
           users:(("master",pid=1132,fd=13))  
LISTEN      0      50      *:445                  *:  
           users:(("smbd",pid=867,fd=37))  
LISTEN      0      50      :::139                :::  
           users:(("smbd",pid=867,fd=36))  
LISTEN      0     128      :::82                :::  
           users:(("httpd",pid=1356,fd=4),("httpd",p...  
LISTEN      0     128      :::22                :::  
           users:(("sshd",pid=821,fd=4))  
LISTEN      0     100      :::125                :::  
           users:(("master",pid=1132,fd=14))  
LISTEN      0      50      :::445                :::  
           users:(("smbd",pid=867,fd=35))
```

### 附1.3.3.7. 搜索日志

访问成功或被SELinux拒绝的决定被缓存一次，拒绝访问被发送到日志文件。

SELinux的缓存称为AVC（Access Vector Cache访问向量缓存），拒绝访问也称为“AVC拒绝”。AVC拒绝日志是通过[Rsyslog服务](#)或[Audit服务](#)生成的，所以它需要服务正在运行。

通过Rsyslog发送的消息是使用“kern”工具生成的。CentOS默认的Rsyslog设置写为`*.info;xxx /var/log/messages`，所以AVC拒绝日志记录到`/var/log/messages`：

```
grep "avc: .denied" /var/log/messages
```

```
Apr  2 13:20:06 www kernel: type=1400 audit(1459743606.523:6): a  
vc: denied { read } for pid=1298  
    comm="httpd" name="index.html" dev="dm-0" ino=67206855 scon  
text=system_u:system_r:httpd_t:s0  
    tcontext=unconfined_u:object_r:user_home_t:s0 tclass=file  
Apr  2 13:22:13 www kernel: type=1400 audit(1459743733.690:4): a  
vc: denied { read } for pid=891  
    comm="httpd" name="index.html" dev="dm-0" ino=67206855 scon  
text=system_u:system_r:httpd_t:s0  
    tcontext=unconfined_u:object_r:user_home_t:s0 tclass=file
```

通过Auditd的消息生成到`/var/log/audit/audit.log`：

```
grep "avc: .denied" /var/log/audit/audit.log
```

```
type=AVC msg=audit(1459146274.923:133): avc: denied { create }
for pid=8173 comm="smbd"
    name=E696B0E38197E38184E38395E382A9E383ABE38380E383BC scont
ext=system_u:system_r:smbd_t:s0
    tcontext=system_u:object_r:user_home_dir_t:s0 tclass=dir
type=AVC msg=audit(1459146274.924:134): avc: denied { create }
for pid=8173 comm="smbd"
    name=E696B0E38197E38184E38395E382A9E383ABE38380E383BC scont
ext=system_u:system_r:smbd_t:s0
    tcontext=system_u:object_r:user_home_dir_t:s0 tclass=dir
type=AVC msg=audit(1459217340.695:63): avc: denied { name_bind
} for pid=1320 comm="httpd"
    src=82 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket
type=AVC msg=audit(1459217340.696:64): avc: denied { name_bind
} for pid=1320 comm="httpd"
    src=82 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket
```

对于通过Auditd的消息，可以使用 [ausearch 命令](#)搜索它们：

```
ausearch -m AVC
```

```

-----
time->Mon Mar 28 14:59:30 2016
type=SYSCALL msg=audit(1459144770.995:64): arch=c000003e syscall
=83 success=no exit=-13 a0=7fac66386bb0
    a1=1ff a2=1ff a3=7fac66388888 items=0 ppid=8142 pid=8173 au
id=4294967295 uid=99 gid=0 euid=99 suid=0
    fsuid=99 egid=99 sgid=0 fsgid=99 tty=(none) ses=4294967295
comm="smbd" exe="/usr/sbin/smbd"
    subj=system_u:system_r:smbd_t:s0 key=(null) type=AVC msg=au
dit(1459144770.995:64):
    avc: denied { create } for pid=8173 comm="smbd" name=E69
6B0E38197E38184E38395E382A9E383ABE38380E383BC
    scontext=system_u:system_r:smbd_t:s0 tcontext=system_u:obje
ct_r:user_home_dir_t:s0 tclass=dir
-----
time->Mon Apr 4 11:27:08 2016
type=SYSCALL msg=audit(1459736828.877:69): arch=c000003e syscall
=49 success=no exit=-13 a0=3 a1=7efddf9b8cf8
    a2=10 a3=7ffceb56695c items=0 ppid=1 pid=1407 auid=42949672
95 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
    gid=0 fsgid=0 tty=(none) ses=4294967295 comm="httpd" exe="/
usr/sbin/httpd" subj=system_u:system_r:httpd_t:s0
    key=(null) type=AVC msg=audit(1459736828.877:69): avc: den
ied { name_bind } for pid=1407 comm="httpd"
    src=82 scontext=system_u:system_r:httpd_t:s0 tcontext=syste
m_u:object_r:reserved_port_t:s0 tclass=tcp_socket
-----
time->Mon Apr 4 11:27:08 2016
type=SYSCALL msg=audit(1459736828.877:68): arch=c000003e syscall
=49 success=no exit=-13 a0=4 a1=7efddf9b8db8
    a2=1c a3=7ffceb566710 items=0 ppid=1 pid=1407 auid=42949672
95 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
    sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="httpd" exe="/
usr/sbin/httpd" subj=system_u:system_r:httpd_t:s0
    key=(null) type=AVC msg=audit(1459736828.877:68): avc: den
ied { name_bind } for pid=1407 comm="httpd"
    src=82 scontext=system_u:system_r:httpd_t:s0 tcontext=syste
m_u:object_r:reserved_port_t:s0 tclass=tcp_socket

```

对于通过Auditd的消息，可以使用 `aureport` 命令显示汇总报告：

```
aureport --avc
```

```
AVC Report
=====
# date time comm subj syscall class permission obj event
=====
1. 08/08/2015 02:13:50 ? system_u:system_r:init_t:s0 0 (null) (null) (null) unset 347
2. 03/28/2016 13:51:10 ? system_u:system_r:kernel_t:s0 0 (null) (null) unset 9
3. 03/28/2016 14:59:30 smbd system_u:system_r:smbd_t:s0 83 dir create system_u:object_r:user_home_dir_t:s0 denied 64
4. 03/28/2016 14:59:30 smbd system_u:system_r:smbd_t:s0 83 dir create system_u:object_r:user_home_dir_t:s0 denied 65
5. 03/28/2016 14:59:30 smbd system_u:system_r:smbd_t:s0 83 dir create system_u:object_r:user_home_dir_t:s0 denied 66
.....
.....
64. 04/04/2016 11:27:03 httpd system_u:system_r:httpd_t:s0 42 tc p_socket name_connect system_u:object_r:reserved_...
65. 04/04/2016 11:27:08 httpd system_u:system_r:httpd_t:s0 49 tc p_socket name_bind system_u:object_r:reserved_port...
66. 04/04/2016 11:27:08 httpd system_u:system_r:httpd_t:s0 49 tc p_socket name_bind system_u:object_r:reserved_port...
```

```
aureport --avc --summary
```

```
Avc Object Summary Report
=====
total obj
=====
32 unconfined_u:object_r:home_root_t:s0
20 system_u:object_r:user_home_dir_t:s0
5 system_u:object_r:reserved_port_t:s0
```

### 附1.3.3.8. 使用SETroubleShoot

使用SETroubleShoot，可以生成额外的日志来解决一些问题。来自SETroubleShoot的消息由Audit Event Dispatcher发送到 /var/log/messages，因此需要运行[Auditd](#)。

安装“setroubleshoot-server”并配置一些设置以使用 sealert 命令：

```
yum -y install setroubleshoot-server
```

编辑 /etc/tmpfiles.d/setroubleshoot.conf 文件：

```
D /var/run/setroubleshoot 0755 setroubleshoot root -
```

```
mkdir --context=system_u:object_r:setroubleshoot_var_run_t:s0  
/var/run/setroubleshoot
```

```
chown setroubleshoot:root /var/run/setroubleshoot
```

```
chmod 755 /var/run/setroubleshoot
```

```
service auditd restart  
systemctl restart dbus
```

全部完成，AVC拒绝的其他日志会输出到 /var/log/messages ，如下所示：

```
grep -E 'setroubleshoot|preventing' /var/log/messages
```

```

Apr  3 19:33:41 dlp setroubleshoot: failed to retrieve rpm info
for /var/www/html/index.html
Apr  3 19:33:41 dlp setroubleshoot: SELinux is preventing /usr/s
bin/httpd from setattr access
          on the file /var/www/html/index.html. For comple
te SELinux messages.
              run sealert -l 84495686-3c5c-411f-9fb7-bb396ac49
c1d
Apr  3 19:33:41 dlp python: SELinux is preventing /usr/sbin/http
d from setattr access on the file /var/www/html/index.html.
          #012#012***** Plugin restorecon (99.5 confidenc
e) suggests  ****
If you want to fix the label. #012/var/www/html/
index.html default label should be httpd_sys_content_t.
          #012Then you can run restorecon.#012Do#012# /sbi
n/restorecon -v /var/www/html/index.html#012#012*****
          Plugin catchall (1.49 confidence) suggests  ***
****#012#012If you believe that
          httpd should be allowed setattr access on the in
dex.html file by default.#012Then you should report
          this as a bug.#012You can generate a local polic
y module to allow this access.#012Do#012allow this access
          for now by executing:#012# grep httpd /var/log/a
udit/audit.log | audit2allow -M mypol#012#
          semodule -i mypol.pp#012

```

对于上面示例的第4行，给出了一个命令来查看更多详细信息，并显示如下所示的日志：

```
sealert -l 84495686-3c5c-411f-9fb7-bb396ac49c1d
```

```

SELinux is preventing /usr/sbin/httpd from setattr access on the
file /var/www/html/index.html.

***** Plugin restorecon (99.5 confidence) suggests  ****
****

If you want to fix the label.
/var/www/html/index.html default label should be httpd_sys conte
nt_t.

```

```

Then you can run restorecon.

Do
# /sbin/restorecon -v /var/www/html/index.html

***** Plugin catchall (1.49 confidence) suggests      *****
***** *****

If you believe that httpd should be allowed setattr access on the index.html file by default.

Then you should report this as a bug.

You can generate a local policy module to allow this access.

Do

allow this access for now by executing:
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp

Additional Information:
Source Context          system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:admin_home_t:s0
Target Objects          /var/www/html/index.html [ file ]
Source                 httpd
Source Path             /usr/sbin/httpd
Port                  <Unknown>
Host                  dlp.srv.world
Source RPM Packages    httpd-2.4.6-40.el7.centos.x86_64
Target RPM Packages
Policy RPM              selinux-policy-3.13.1-60.el7_2.3.noarch
Selinux Enabled         True
Policy Type             targeted
Enforcing Mode          Enforcing
Host Name               dlp.srv.world
Platform                Linux dlp.srv.world 3.10.0-327.10.1.el7.x86_64
                                      #1 SMP Tue Feb 16 17:03:50 UTC 2016 x86_64 x86_64
Alert Count              3
First Seen              2016-04-03 19:11:23 JST
Last Seen               2016-04-03 19:33:40 JST
Local ID                84495686-3c5c-411f-9fb7-bb396ac49c1d

```

```

Raw Audit Messages
type=AVC msg=audit(1459845220.621:57): avc: denied { getattr }
for pid=847 comm="httpd"
path="/var/www/html/index.html" dev="dm-0" ino=101186198 sco
ntext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file

type=SYSCALL msg=audit(1459845220.621:57): arch=x86_64 syscall=1
stat success=no exit=EACCES a0=7fb4dab46f70
a1=7fff43babe90 a2=7fff43babe90 a3=0 items=0 ppid=822 pid=84
7 auid=4294967295 uid=48 gid=48 euid=48
suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=429
4967295 comm=httpd exe=/usr/sbin/httpd
subj=system_u:system_r:httpd_t:s0 key=(null)

Hash: httpd,httpd_t,admin_home_t,file,getattr

```

### 附1.3.3.9. 使用 audit2allow

使用 `audit2allow` 命令，可以从拒绝操作的日志中轻松生成 SELinux 策略允许规则。但是，`audit2allow` 可能允许比所需要的更多的访问权限，所以最好使用 `restorecon` 或 `chcon` 命令进行配置。

```
yum install policycoreutils-python # 如果系统中不存在 audit2allow ，先安装
```

读取日志文件显示拒绝的原因（如果未指定任何日志文件，则读取 `/var/log/audit/audit.log`；如果指定日志文件，将 `-i logfile` 选项改为 `-a` 选项）：

```
audit2allow -w -a # 读取 audit.log 显示AVC拒绝的原因
```

```
type=AVC msg=audit(1460007772.762:55): avc: denied { getattr }
for pid=1029 comm="httpd" path="/var/www/html/index.html"
    dev="dm-0" ino=101186198 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
Was caused by:
    Missing type enforcement (TE) allow rule.
```

You can use audit2allow to generate a loadable module to allow this access.

.....  
.....

```
type=AVC msg=audit(1460007828.479:64): avc: denied { getattr }
for pid=1056 comm="httpd" path="/var/www/html/index.html"
    dev="dm-0" ino=101186198 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
Was caused by:
    Missing type enforcement (TE) allow rule.
```

You can use audit2allow to generate a loadable module to allow this access.

```
ausearch -m AVC --start 04/05/2016 19:52:00 --end 04/05/2016
19:52:59 | audit2allow -w #例如，使用 ausearch 显示特定日志
```

```
type=AVC msg=audit(1460009034.012:76): avc: denied { getattr }  
for pid=1054 comm="httpd" path="/var/www/html/index.html"  
dev="dm-0" ino=101186198 scontext=system_u:system_r:httpd_t:  
s0 tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
```

Was caused by:

Missing type enforcement (TE) allow rule.

You can use audit2allow to generate a loadable module to allow this access.

```
type=AVC msg=audit(1460009034.013:77): avc: denied { getattr }  
for pid=1054 comm="httpd" path="/var/www/html/index.html"  
dev="dm-0" ino=101186198 scontext=system_u:system_r:httpd_t:  
s0 tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
```

Was caused by:

Missing type enforcement (TE) allow rule.

You can use audit2allow to generate a loadable module to allow this access.

```
ausearch -m AVC --start 04/05/2016 19:52:00 --end 04/05/2016  
19:52:59 | audit2allow -a # 使用 -a 选项显示所需类型
```

```
===== httpd_t =====  
allow httpd_t admin_home_t:file getattr;
```

生成允许规则如下：

```
ausearch -m AVC --start 04/05/2016 19:52:00 --end 04/05/2016  
19:52:59 | audit2allow -a -M test_rule # 例如，生成“test_rule”模块
```

```
***** IMPORTANT *****  
To make this policy package active, execute:
```

```
semodule -i test_rule.pp
```

```
semodule -i test_rule.pp # 使用上面显示的命令安装模块
```

```
semodule -l | grep test_rule # 确认模块已加载
```

```
test_rule 1.0
```

在某些情况下已完成设置，但对于有些情况，还没有，如下例就不能正常访问：

```
curl http://localhost/index.html
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /index.html
on this server.</p>
</body></html>
```

原因是“`httpd_t`”域仅使用“`getattr`”来访问“`admin_home_t`”类型文件是不够的。在这种情况下，再次使用 `audit2allow` 生成规则：

```
ausearch -m AVC | grep -E 'http|index.html' | audit2allow -a
```

```
# 读 (read) 权限也是必需的
===== httpd_t =====
allow httpd_t admin_home_t:file read;

!!!! This avc is allowed in the current policy
allow httpd_t admin_home_t:file getattr;
```

```
ausearch -m AVC | grep -E 'http|index.html' | audit2allow -a -M
test_rule
```

```
semodule -i test_rule.pp
```

```
curl http://localhost/index.html
```

```
# 还无法访问
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
....
```

```
ausearch -m AVC | grep -E 'http|index.html' | audit2allow -a
```

```
# 打开(open)权限也是必需的
===== httpd_t =====
allow httpd_t admin_home_t:file open;

!!!! This avc is allowed in the current policy
allow httpd_t admin_home_t:file { read getattr };
```

```
ausearch -m AVC | grep -E 'http|index.html' | audit2allow -a -M
test_rule
```

```
semodule -i test_rule.pp
curl http://localhost/index.html
```

```
Test Page # 访问成功
```

### 附1.3.3.10. 使用matchpathcon

使用 `matchpathcon` 命令显示指定路径的默认SELinux上下文。以下是使用 `matchpathcon` 命令的一些示例：

```
matchpathcon /var/www/html/index.html # 显
示 /var/www/html/index.html 的默认值
```

```
/var/www/html/index.html          system_u:object_r:httpd_sys_cont
ent_t:s0
```

```
matchpathcon -V /var/www/html # 比较当前上下文和 /var/www/html 的默认值（如果没有差异，显示“verified”）
```

```
/var/www/html verified.
```

```
matchpathcon -V /var/www/html/index.html # 比较当前上下文和 /var/www/html/index.html 的默认值（如果不同，显示如下）
```

```
/var/www/html/index.html has context unconfined_u:object_r:admin_home_t:s0, should be system_u:object_r:httpd_sys_content_t:s0
```

```
matchpathcon -V /var/www/html/* # 可以使用通配符指定目标
```

```
/var/www/html/cgi-enabled verified.  
/var/www/html/index.html has context unconfined_u:object_r:user_home_t:s0, should be system_u:object_r:httpd_sys_content_t:s0  
/var/www/html/index.php verified.  
/var/www/html/index.py verified.  
/var/www/html/info.php verified.
```

### 附1.3.3.11. 使用sesearch

使用 `sesearch` 命令搜索SELinux策略。

```
yum install setools-console # 如果系统中不存在 sesearch , 先安装
```

以下是使用 `sesearch` 命令的一些示例：

```
sesearch --allow # 显示允许的全部规则
```

```
Found 95937 semantic av rules:  
allow logrotate_t systemd_passwd_var_run_t : sock_file { ioctl  
1 read write create setattr setattr lock...  
allow dmidecode_t virtd_t : fd use ;  
allow ssh_keygen_t anaconda_t : fd use ;  
allow logadm_t systemd_passwd_var_run_t : sock_file { ioctl r  
ead write create setattr setattr lock app...  
allow unconfined_dbusd_t unconfined_dbusd_t : x_device { geta  
ttr setattr use read write getfocus setfo...  
.....  
.....
```

```
seseach -s httpd_t --allow -d # 显示允许访问“httpd_t”域的规则（-d （-  
-direct）表示按照字面搜索）
```

```
Found 915 semantic av rules:  
allow httpd_t system_dbusd_t : unix_stream_socket connectto ;  
allow httpd_t dirsrv_config_t : file { ioctl read write creat  
e setattr setattr lock append unlink link rename op...  
allow httpd_t dirsrv_config_t : dir { ioctl read write create  
getattr setattr lock unlink link rename add_name r...  
allow httpd_t httpd_squirrelmail_t : file { ioctl read write  
create setattr setattr lock append unlink link renam...  
.....  
.....
```

```
seseach -t httpd_sys_script_exec_t --allow -d # 显示哪些域可以访  
问“httpd_sys_script_exec_t”类型的允许规则
```

```
Found 10 semantic av rules:  
allow httpd_sys_script_t httpd_sys_script_exec_t : file { ioctl  
read setattr lock execute execute_no_trans entry...  
allow httpd_sys_script_t httpd_sys_script_exec_t : dir { ioctl  
read setattr lock search open } ;  
allow httpd_sys_script_exec_t httpd_sys_script_exec_t : files  
system associate ;  
allow openshift_domain httpd_sys_script_exec_t : file { ioctl  
read setattr lock execute execute_no_trans open } ;  
allow openshift_domain httpd_sys_script_exec_t : dir { setattr  
search open } ;  
....  
....
```

```
seseach -t shadow_t -c file -p write --allow # 显示哪些域可以写  
入“shadow_t”类型的文件的允许规则
```

```
Found 10 semantic av rules:  
allow updpwd_t shadow_t : file { ioctl read write create geta  
ttr setattr lock append unlink link rename open } ;  
allow yppasswdd_t shadow_t : file { ioctl read write create g  
etattr setattr lock relabelfrom relabelto append unl...  
allow pegasus_openlmi_account_t shadow_t : file { ioctl read  
write create setattr setattr lock relabelfrom relabe...  
allow files_unconfined_type file_type : file { ioctl read wri  
te create setattr setattr lock relabelfrom relabelto...  
allow sysadm_passwd_t shadow_t : file { ioctl read write crea  
te setattr setattr lock relabelfrom relabelto append...  
....  
....
```

```
seseach -b samba_enable_home_dirs --allow -d # 显示布尔  
值“samba_enable_home_dirs”的定义规则
```

```
Found 8 semantic av rules:  
allow smbd_t home_root_t : dir { ioctl read getattr lock search open } ;  
allow smbd_t home_root_t : lnk_file { read getattr } ;  
allow smbd_t user_home_type : file { ioctl read write create getattr setattr lock append unlink link rename open ... } ;  
allow smbd_t user_home_type : dir { ioctl read write create getattr setattr lock unlink link rename add_name remove ... } ;  
allow smbd_t user_home_type : lnk_file { ioctl read write create getattr setattr lock append unlink link rename } ;  
....  
....
```

## 附1.4. 文件同步

### 附1.4.1. Rsync

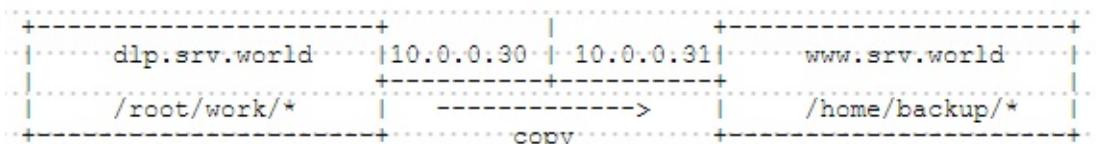
Rsync (remote sync) 是类unix系统下的数据镜像备份工具。它的特性如下：

1. 可以镜像保存整个目录树和文件系统。
2. 可以很容易做到保持原来文件的权限、时间、软硬链接等等。
3. 无须特殊权限即可安装。
4. 优化的流程，文件传输效率高。
5. 可以使用rcp、ssh等方式来传输文件，当然也可以通过直接的socket连接。
6. 支持匿名传输。

Rsync的基本用法：`rsync 选项 源 目的地`，如 `rsync -av --exclude="tmp" /home/ /backup`

| 选项                             | 描述                                                                |
|--------------------------------|-------------------------------------------------------------------|
| <code>-v</code>                | increase verbosity                                                |
| <code>-a</code>                | archive mode; same as <code>-rlptgoD</code> (no <code>-H</code> ) |
| <code>-u</code>                | skip files that are newer on the receiver                         |
| <code>-z</code>                | compress file data during the transfer                            |
| <code>--exclude=PATTERN</code> | exclude files matching PATTERN                                    |
| <code>--delete</code>          | delete files that don't exist on sender                           |

如果要由cron或其他自动设置rsync，需要按下面的配置，因为不设置就需要认证。例如，将 `d1p.srv.world` 上 `/root/work` 目录下的文件或目录复制到 `www.srv.world` 上的 `/home/backup`。



在源主机上配置：

```
yum -y install rsync
```

编辑 `/etc/rsync_exclude.lst` 文件：

```
# 指定要排除复制的文件或目录  
test  
test.txt
```

在目的地主机上配置：

```
yum -y install rsync
```

编辑 `/etc/rsyncd.conf` 文件：

```
# 任意名称  
[backup]  
# 复制目的地目录  
path = /home/backup  
# 允许访问的主机  
hosts allow = 10.0.0.30  
hosts deny = *  
list = true  
uid = root  
gid = root  
read only = false
```

```
mkdir /home/backup
```

```
systemctl start rsyncd  
systemctl enable rsyncd
```

配置完成。在源主机执行以下 `rsync` 命令：

```
rsync -avz --delete --exclude-from=/etc/rsync_exclude.lst  
/root/work/ www.srv.world::backup
```

如果要定期运行，在`cron`中添加：

```
crontab -e
```

```
00 02 * * * rsync -avz --delete --exclude-from=/etc/rsync_exclud  
e.lst /root/work/ www.srv.world::backup
```

## 附1.4.2. Lsyncd

一般rsync软件是通过crond这支后台进行（计划任务）来实现自动同步数据，如今已有更好的开源软件来代替使用crond了，那就是[Lsyncd \(Live Syncing \(Mirror\) Daemon\)](#)。它的工作原理：监视本地（rsync client）的目录，当源数据有文件或目录更新时，更新本地文件或目录到远端机器（rsync server），保持实时文件同步，但是它更新数据时需要远端rsync server运行rsync demon。

先按上一节内容配置好Rsync。

安装并配置Lsyncd以实时同步文件或目录：

```
yum --enablerepo=epel -y install lsyncd # 从EPEL安装
```

编辑 /etc/lsyncd.conf 文件：

```
# 注释下行
-- sync{default.rsyncssh, source="/var/www/html", host="localhost",
        targetdir="// tmp/htmlcopy/"}

# 添加以下内容到最后
settings{
    statusFile = "/tmp/lsyncd.stat",
    statusInterval = 1,
}
sync{
    default.rsync,
    # 源目录
    source="/root/work/",
    # 目的地主机名或IP地址:(在rsyncd.conf中设置的名称)
    target="10.0.0.31::backup",
    # 排除列表
    excludeFrom="/etc/rsync_exclude.lst",
}
```

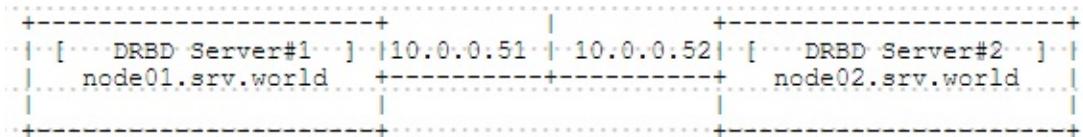
```
systemctl start lsyncd
systemctl enable lsyncd
```

确认文件和目录在目的地主机上实时复制。

## 附1.4.3. DRBD

**DRBD (Distributed Replicated Block Device)** 是由内核模块和相关脚本而构成，用以构建高可用性的集群。其实现方式是通过网络来镜像整个设备。您可以把它看作是一种网络RAID。DRBD负责接收数据，把数据写到本地磁盘，然后发送给另一个主机。另一个主机再将数据存到自己的磁盘中。

本例基于以下环境：



要安装DRBD的服务器有必要有空闲的块设备。

本例演示配置为使用块设备 `/dev/vg_r0/lv_r0`。

### 附1.4.3.1. 安装DRBD

在两个主机更新系统，安装所需的软件包并重新启动：

```
yum -y update
```

```
yum -y install gcc make automake autoconf libxslt libxslt-devel
flex rpm-build kernel-devel
```

```
reboot
```

在两台主机上安装DRBD，确认[下载最新版本](#)：

```
mkdir -p rpmbuild/{BUILD,BUILDROOT,RPMS,SOURCES,SPECS,SRPMS}
```

```
wget http://oss.linbit.com/drbd/drbd-utils-latest.tar.gz \
http://oss.linbit.com/drbd/8.4/drbd-8.4.7-1.tar.gz
```

```
tar zxvf drbd-8.4.7-1.tar.gz
```

```
cd drbd-*
```

```
make km-rpm
```

```
cd
```

```
tar zxvf drbd-utils-latest.tar.gz
```

```
cd drbd-utils-*
```

编辑 `drbd.spec.in` 文件：

```
# 添加  
%bcond_without sbinsymlinks  
%undefine with_sbinsymlinks
```

```
./configure
```

```
make rpm
```

```
cd /root/rpmbuild/RPMS/x86_64
```

```
rpm -Uvh drbd-utils-*.*.rpm drbd-km-*.*.rpm
```

```
Preparing... ###################################################  
##### [100%]  
Updating / installing...  
 1:drbd-utils-8.9.5-1.el7.centos ######  
##### [ 33%]  
 2:drbd-km-3.10.0_327.4.5.el7.x86_64####  
##### [ 67%]  
 3:drbd-km-debuginfo-8.4.7-1 ######  
##### [100%]
```

注：“drbd-km”软件包是使用当前版本的内核构建的，所以如果将来更新内核，那么需要使用新版本内核重新构建DRBD。

### 附1.4.3.2. 配置DRBD

在两台主机上配置DRBD：

编辑 `/etc/drbd.d/global_common.conf` 文件：

```
# 在“disk”部分中添加以下内容（如果发生IO错误，则分离磁盘）  
disk {  
    on-io-error detach;
```

编辑 `/etc/drbd.d/r0.res` 文件：

```
resource r0 {
    # DRBD设备
    device /dev/drbd0;
    # 块设备
    disk /dev/vg_r0/lv_r0;
    meta-disk internal;
    on node01.srv.world {
        # IP地址:端口
        address 10.0.0.51:7788;
    }
    on node02.srv.world {
        address 10.0.0.52:7788;
    }
}
```

```
modprobe drbd # 加载模块
```

```
lsmod | grep drbd
```

|           |        |            |
|-----------|--------|------------|
| drbd      | 405309 | 0          |
| libcrc32c | 12644  | 2 xfs,drbd |

```
drbdadm create-md r0 # 创建DRBD资源
```

```
--== Thank you for participating in the global usage survey ==
--
The server's response is:

you are the 972th user to install this version
initializing activity log
NOT initializing bitmap
Writing meta data...
New drbd meta data block successfully created.
success
```

```
systemctl start drbd
```

```
DRBD's startup script waits for the peer node(s) to appear.  
- If this node was already a degraded cluster before the  
  reboot, the timeout is 0 seconds. [degr-wfc-timeout]  
- If the peer was available before the reboot, the timeout  
  is 0 seconds. [wfc-timeout]  
(These values are for resource 'r0'; 0 sec -> wait forever)  
...  
...  
To abort waiting enter 'yes' [ 18]:yes
```

```
systemctl enable drbd
```

在两台主机上进行配置后，在**node01**主机上同步数据：

```
cat /proc/drbd # 当前状态为“Secondary/Secondary”
```

```
version: 8.4.7-1 (api:1/proto:86-101)  
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by root  
@node01.srv.world, 2016-01-28 14:44:07  
0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsis  
tent C r-----  
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f  
oos:20938076
```

```
drbdadm -- --overwrite-data-of-peer primary r0 # 获取primary角色并同  
步数据
```

```
cat /proc/drbd # 同步开始
```

```
version: 8.4.7-1 (api:1/proto:86-101)  
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by root  
@node01.srv.world, 2016-01-28 14:44:07  
0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent  
C r-----  
    ns:39144 nr:0 dw:0 dr:40056 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep  
:1 wo:f oos:20898932  
    [>.....] sync'ed: 0.2% (20408/20444)M  
    finish: 1:19:58 speed: 4,348 (4,348) K/sec
```

```
cat /proc/drbd # 同步后，状态如下
```

```
version: 8.4.7-1 (api:1/proto:86-101)
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by root
@node01.srv.world, 2016-01-28 14:44:07
  0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r--
  --
    ns:20938076 nr:0 dw:0 dr:20938988 al:0 bm:0 lo:0 pe:0 ua:0 a
    p:0 ep:1 wo:f oos:0
```

可以配置DRBD，在DRBD设备上创建文件系统并将其挂载使用：

```
mkfs.xfs /dev/drbd0
```

```
mkdir /drbd_disk
```

```
mount /dev/drbd0 /drbd_disk
```

```
df -hT
```

| Filesystem<br>on        | Type     | Size | Used | Avail | Use% | Mounted        |
|-------------------------|----------|------|------|-------|------|----------------|
| /dev/mapper/centos-root | xfs      | 27G  | 1.7G | 25G   | 7%   | /              |
| devtmpfs                | devtmpfs | 2.0G | 0    | 2.0G  | 0%   | /dev           |
| tmpfs                   | tmpfs    | 2.0G | 0    | 2.0G  | 0%   | /dev/shm       |
| tmpfs                   | tmpfs    | 2.0G | 8.3M | 2.0G  | 1%   | /run           |
| tmpfs                   | tmpfs    | 2.0G | 0    | 2.0G  | 0%   | /sys/fs/cgroup |
| /dev/vda1               | xfs      | 497M | 206M | 292M  | 42%  | /boot          |
| tmpfs                   | tmpfs    | 396M | 0    | 396M  | 0%   | /run/user/0    |
| /dev/drbd0              | xfs      | 20G  | 33M  | 20G   | 1%   | /drbd_disk     |

```
echo 'test file' > /drbd_disk/test.txt # 创建测试文件
```

```
ll /drbd_disk
```

```
total 4
-rw-r--r-- 1 root root 10 Jan 28 15:32 test.txt
```

如下在**node02**主机挂载DRBD设备：

先在当前的primary主机（node01）运行：

```
umount /drbd_disk # 卸载
```

```
drbdadm secondary r0 # 获取secondary角色
```

在当前的secondary主机（node02）运行：

```
drbdadm primary r0 # 获取primary角色
```

```
mount /dev/drbd0 /drbd_disk # 挂载
```

```
df -hT
```

| Filesystem<br>on        | Type     | Size | Used | Avail | Use% | Mounted        |
|-------------------------|----------|------|------|-------|------|----------------|
| /dev/mapper/centos-root | xfs      | 27G  | 1.7G | 25G   | 7%   | /              |
| devtmpfs                | devtmpfs | 2.0G | 0    | 2.0G  | 0%   | /dev           |
| tmpfs                   | tmpfs    | 2.0G | 0    | 2.0G  | 0%   | /dev/shm       |
| tmpfs                   | tmpfs    | 2.0G | 8.4M | 2.0G  | 1%   | /run           |
| tmpfs                   | tmpfs    | 2.0G | 0    | 2.0G  | 0%   | /sys/fs/cgroup |
| /dev/vda1               | xfs      | 497M | 206M | 292M  | 42%  | /boot          |
| tmpfs                   | tmpfs    | 396M | 0    | 396M  | 0%   | /run/user/0    |
| /dev/drbd0              | xfs      | 20G  | 33M  | 20G   | 1%   | /drbd_disk     |

```
ll /drbd_disk
```

```
total 4
-rw-r--r-- 1 root root 10 Jan 28 15:32 test.txt
```

## 附1.5. PowerShell

[PowerShell](#)是一个跨平台（Windows，Linux和OS X）的自动化和配置工具（框架），可以和已有的工具友好集成，特别优化用于处理结构化数据（如JSON，CSV，XML等），REST APIs以及对象模型。它包含一个命令行Shell、一个关联的脚本语言以及一个用于处理cmdlets的框架。

检查[最新版本的PowerShell](#)，并使用yum命令进行安装：

```
yum -y install
https://github.com/PowerShell/PowerShell/releases/download/v6.0.0-
alpha.10/powershell-6.0.0_alpha.10-1.el7.centos.x86_64.rpm
```

PowerShell的基本操作：

```
powershell # 运行PowerShell
```

```
PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS /root>

# 显示Cmdlet列表（只显示头10行）
PS /root> (Get-Command)[0..9]

 CommandType      Name
 Version        Source
 ----          -----
 Function        Add-NodeKeys
 0.0           PSDesiredStateConfiguration
 Function        AddDscResourceProperty
 0.0           PSDesiredStateConfiguration
 Function        AddDscResourcePropertyFromMetadata
 0.0           PSDesiredStateConfiguration
 Function        AfterAll
 3.3.9          Pester
 Function        AfterEach
 3.3.9          Pester
 Function        Assert-MockCalled
```

```

3.3.9      Pester
Function      Assert-VerifiableMocks
3.3.9      Pester
Function      BeforeAll
3.3.9      Pester
Function      BeforeEach
3.3.9      Pester
Function      cd..

# 显示当前路径
PS /root> pwd

Path
-----
/root

# 更改目录为 "/home"
PS /root> cd /home

# 回到主目录
PS /home> cd

# 显示当前目录下的文件 (dir等于Get-ChildItem)
PS /root> dir

Directory: /root

Mode          LastWriteTime        Length Name
----          -----          ---- -  
-----          1/8/15    7:52 AM       985  anaconda-ks.cfg

# 显示"/"目录下的文件
PS /root> Get-ChildItem /

Directory: /

Mode          LastWriteTime        Length Name
----          -----          ---- -  
d---l          12/17/15   7:05 PM      bin
d-r---         9/23/16   11:32 PM     boot
d----          9/28/16   7:27 PM      dev
.....

```

```
....
```

```
# 在当前目录下创建新文件
PS /root> New-Item -Path test.txt
```

```
Directory: /root

Mode           LastWriteTime         Length Name
----           -----          0      test.txt
-----          9/28/16   8:52 PM
```

```
PS /root> dir
```

```
Directory: /root

Mode           LastWriteTime         Length Name
----           -----          0      test.txt
----          1/8/15    7:52 AM       985  anaconda-ks.cfg
-----          9/28/16   8:52 PM
```

```
# 在当前目录下创建新目录
PS /root> New-Item -ItemType Directory -Path testdir
```

```
Directory: /root

Mode           LastWriteTime         Length Name
----           -----          0      testdir
-----          9/28/16   8:55 PM
```

```
PS /root> dir
```

```
Directory: /root

Mode           LastWriteTime         Length Name
----           -----          0      test.txt
----          1/8/15    7:52 AM       985  anaconda-ks.cfg
-----          9/28/16   8:55 PM
```

```
# 回传文本并将其重定向到文件
PS /root> echo "test content" >> test.txt
```

```
# 显示文件的内容
PS /root> Get-Content test.txt
test content

# 移动/重命名文件
PS /root> Move-Item test.txt test1.txt
PS /root> dir

    Directory: /root

Mode                LastWriteTime         Length Name
----                -----          28      test1.txt
d-----        9/28/16 8:55 PM
-----        1/8/15 7:52 AM       985 anaconda-ks.cfg
-----        9/28/16 8:57 PM

# 复制文件
PS /root> Copy-Item test1.txt test2.txt
PS /root> dir

    Directory: /root

Mode                LastWriteTime         Length Name
----                -----          28      test1.txt
d-----        9/28/16 8:55 PM
-----        1/8/15 7:52 AM       985 anaconda-ks.cfg
-----        9/28/16 8:57 PM
-----        9/28/16 8:57 PM       28      test2.txt

# 递归复制目录
PS /root> Copy-Item testdir testdir2 -Recurse
PS /root> dir

    Directory: /root

Mode                LastWriteTime         Length Name
----                -----          28      test1.txt
d-----        9/28/16 9:04 PM
d-----        9/28/16 9:04 PM       28      test2.txt
-----        1/8/15 7:52 AM       985 anaconda-ks.cfg
-----        9/28/16 8:57 PM
-----        9/28/16 8:57 PM
```

```
# 删除文件
```

```
PS /root> Remove-Item test2.txt
```

```
PS /root> dir
```

```
Directory: /root
```

| Mode  | LastWriteTime   | Length | Name            |
|-------|-----------------|--------|-----------------|
| ---   | -----           | -----  | -----           |
| d---- | 9/28/16 9:04 PM |        | testdir         |
| d---- | 9/28/16 9:04 PM |        | testdir2        |
| ----- | 1/8/15 7:52 AM  | 985    | anaconda-ks.cfg |
| ----- | 9/28/16 8:57 PM | 28     | test1.txt       |

```
# 递归删除目录
```

```
PS /root> Remove-Item testdir2 -Recurse
```

```
PS /root> dir
```

```
Directory: /root
```

| Mode  | LastWriteTime   | Length | Name            |
|-------|-----------------|--------|-----------------|
| ---   | -----           | -----  | -----           |
| d---- | 9/28/16 9:04 PM |        | testdir         |
| ----- | 1/8/15 7:52 AM  | 985    | anaconda-ks.cfg |
| ----- | 9/28/16 8:57 PM | 28     | test1.txt       |

```
# 搜索在当前目录下名称中包含“.txt”的文件
```

```
PS /root> Get-ChildItem "*.txt" -Recurse
```

```
Directory: /root/testdir
```

| Mode  | LastWriteTime   | Length | Name      |
|-------|-----------------|--------|-----------|
| ---   | -----           | -----  | -----     |
| ----- | 9/28/16 8:57 PM | 28     | test3.txt |

```
Directory: /root
```

| Mode | LastWriteTime | Length | Name  |
|------|---------------|--------|-------|
| ---  | -----         | -----  | ----- |

```
-----          9/28/16   8:57 PM          28 test1.txt

# 在文件“test1.txt”中搜索单词“test”
PS /root> Select-String -Pattern "test" test1.txt

test1.txt:1:test content

# 显示cmdlet的帮助
PS /root> Get-Help Get-Content

NAME
    Get-Content

SYNOPSIS
    Gets the content of the item at the specified location.

SYNTAX
    Get-Content [-Path] <String[]> [-Credential <PSCredential>]
    [-Delimiter <System.String>]
    [-Encoding {Unknown | String | Unicode | Byte
    ....
    .....

# 使用SSH访问另一主机
PS /root> ssh winuser@10.0.0.220
winuser@10.0.0.220's password:
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\winuser> dir
Volume in drive C has no label.
Volume Serial Number is D4E4-BE4E

Directory of C:\Users\winuser

2016/09/28  21:42    <DIR>      .
2016/09/28  21:42    <DIR>      ..
2016/09/28  21:50    <DIR>      .ssh
2016/09/24  01:30    <DIR>      Contacts
2016/09/28  21:37    <DIR>      Desktop
2016/09/24  01:30    <DIR>      Documents
```

```
2016/09/24 01:30 <DIR> Downloads
2016/09/24 01:30 <DIR> Favorites
2016/09/24 01:30 <DIR> Links
2016/09/24 01:30 <DIR> Music
2016/09/25 00:44 <DIR> OneDrive
2016/09/24 01:30 <DIR> Pictures
2016/09/24 01:30 <DIR> Saved Games
2016/09/24 01:30 <DIR> Searches
2016/09/24 01:30 <DIR> Videos
      0 File(s)          0 bytes
    15 Dir(s)  172,000,489,472 bytes free
```

## 附1.6. 项目管理与版本控制

### 附1.6.1. GitLab

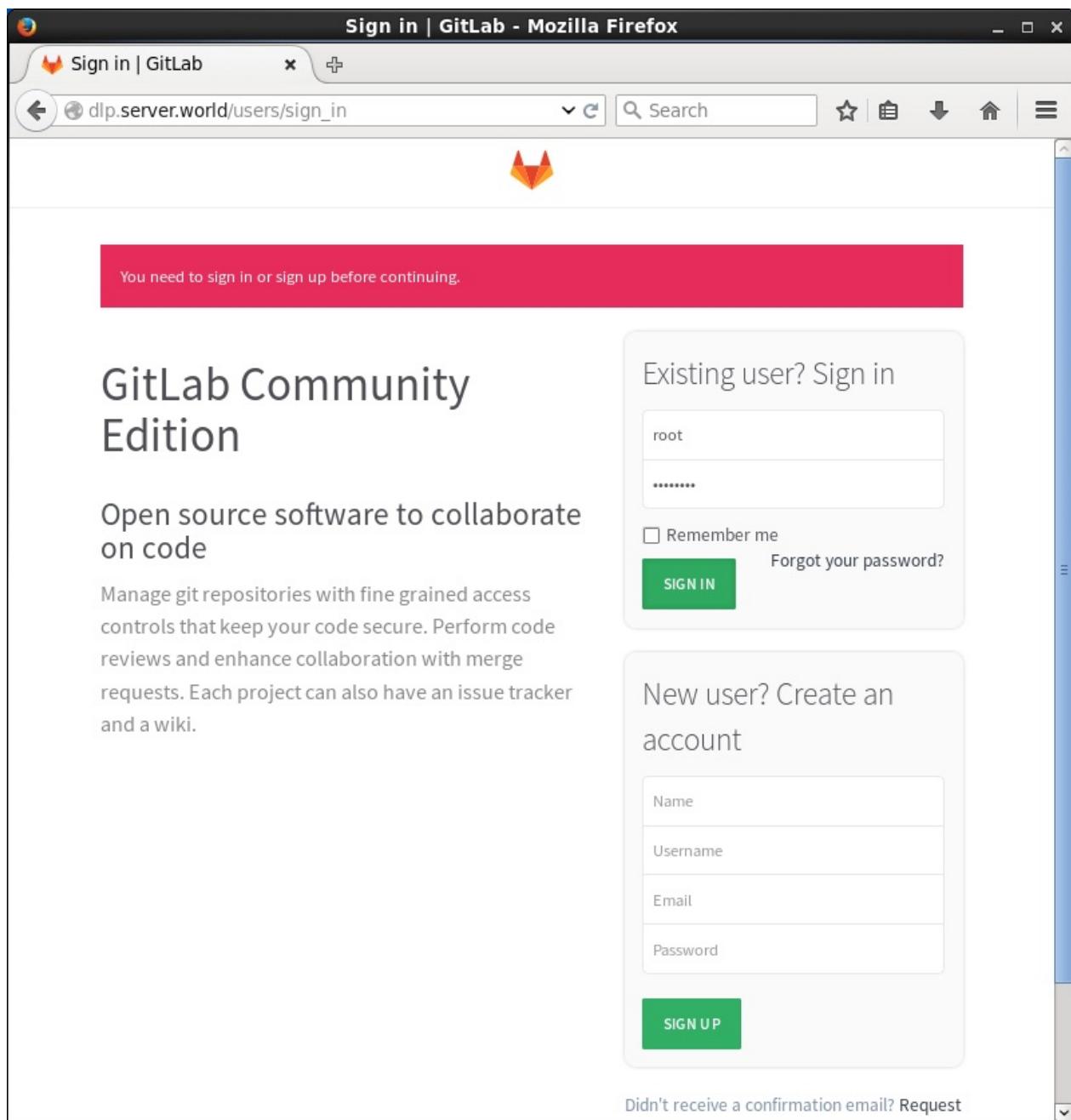
GitLab是一个利用Ruby on Rails开发的开源应用程序，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。如果觉得安装麻烦可以使用[GitLab Installers](#)一键安装程序。

安装并启动SSH服务器，SMTP服务器。如果使用了firewalld防火墙，打开“http”服务端口。

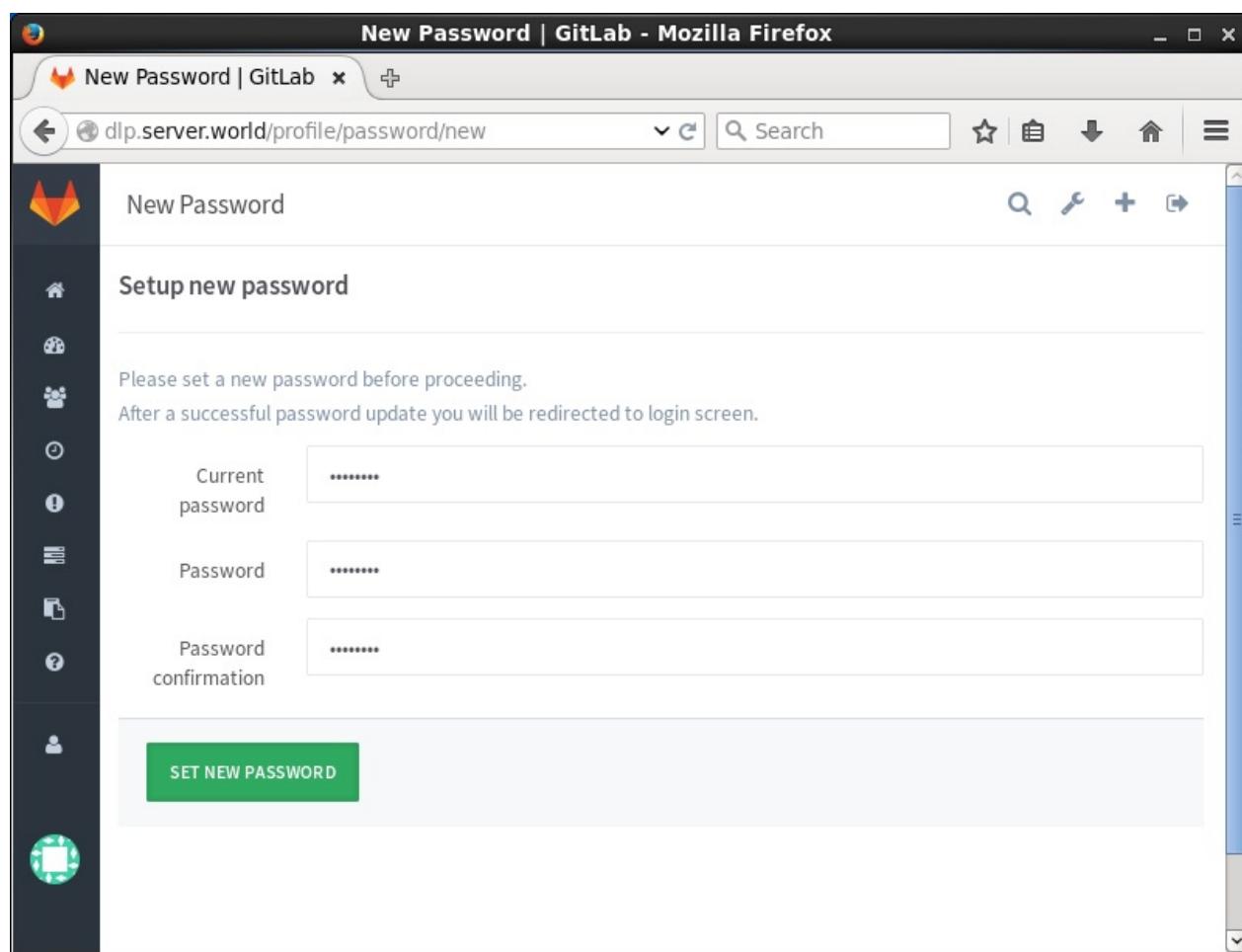
安装GitLab：

```
curl -O  
https://packages.gitlab.com/install/repositories/gitlab/gitlab-  
ce/script.rpm.sh  
  
sh script.rpm.sh  
  
yum -y install gitlab-ce  
  
gitlab-ctl reconfigure
```

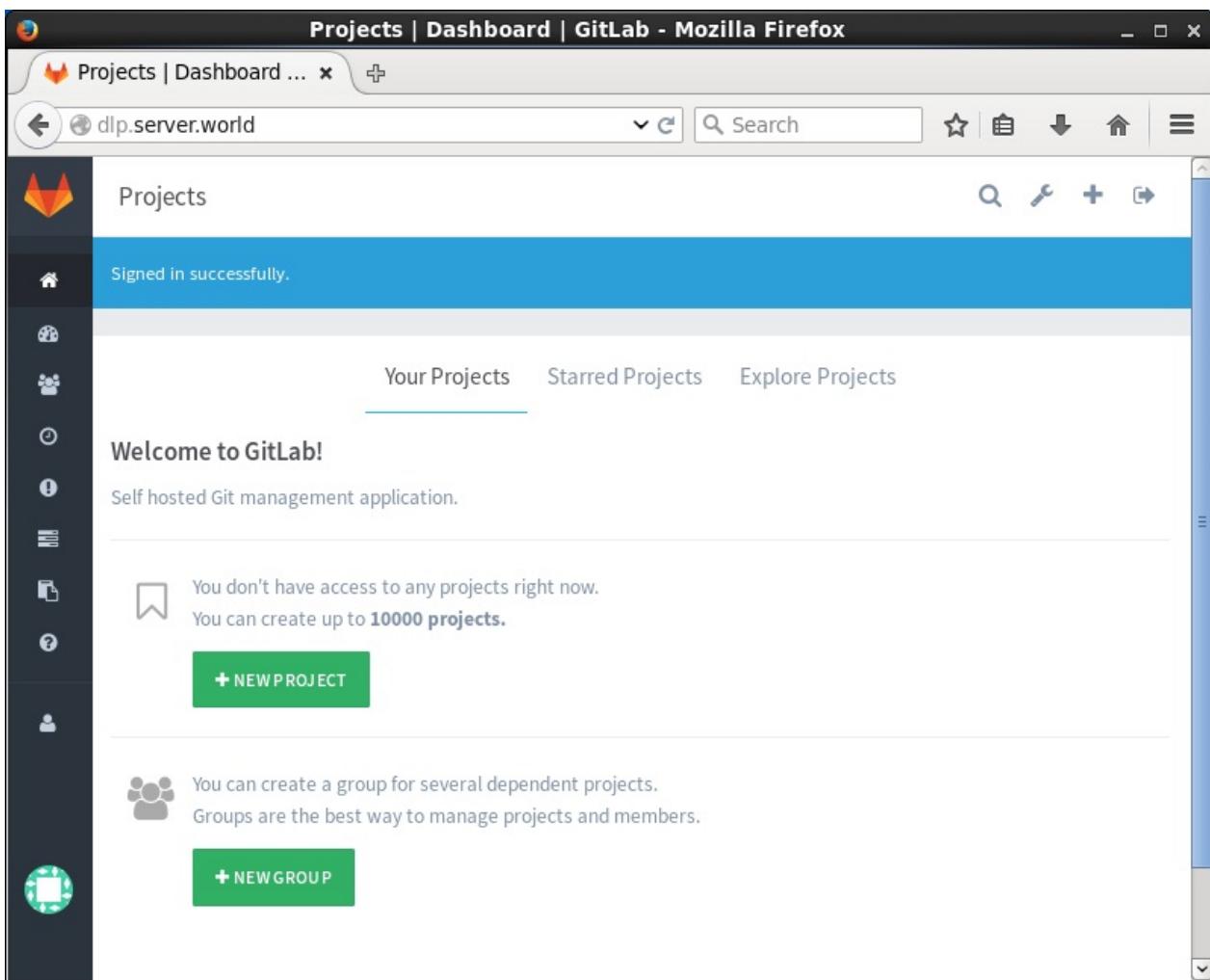
访问 `http://(服务器的主机名或IP地址)/`，然后使用用户“root”和初始密码“5iveL!fe”登录：



初次登录后，需要更改密码。更改为任意密码：



这是GitLab主页。可以在本地网络中像GitHub那样使用：



### 附1.6.2. Redmine

Redmine是基于Web的项目管理和缺陷跟踪工具。它用日历和甘特图辅助项目及进度可视化显示，支持多项目管理。

安装并启动Apache httpd，SMTP服务器和MariaDB数据库服务器并安装Ruby 2.2

安装其他一些需要的软件包：

```
yum -y install ImageMagick ImageMagick-devel libcurl-devel httpd-devel mariadb-devel ipa-pgothic-fonts
```

在MariaDB上为Redmine创建一个用户和数据库：

```
mysql -u root -p
```

```
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 10  
Server version: 5.5.41-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and oth  
ers.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current in  
put statement.  
  
MariaDB [(none)]> create database redmine;  
Query OK, 1 row affected (0.00 sec)  
  
# 在“password”部分设置任意密码  
MariaDB [(none)]> grant all privileges on redmine.* to redmine@'  
localhost' identified by 'password';  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> exit
```

确认最新版本的下载链接，下载并安装：

```
wget http://www.redmine.org/releases/redmine-3.0.3.tar.gz  
tar zxvf redmine-3.0.3.tar.gz  
mv redmine-3.0.3 /var/www/redmine  
cd /var/www/redmine
```

编辑 ./config/database.yml 文件：

```
# 数据库设置
production:
  adapter: mysql2
  # 数据库名称
  database: redmine
  host: localhost
  # 数据库用户
  username: redmine
  # 上面数据库用户的密码
  password: password
  encoding: utf8
```

编辑 `./config/configuration.yml` 文件：

```
# SMTP设置
production:
  email_delivery:
    delivery_method: :smtp
    smtp_settings:
      address: "localhost"
      port: 25
      domain: 'dlp.srv.world'
  rmagick_font_path: /usr/share/fonts/ipa-pgothic/ipagp.ttf
```

```
gem install bundler --no-rdoc --no-ri # 安装bundler

bundle install --without development test postgresql sqlite # 安装
Gem

bundle exec rake generate_secret_token # 生成密钥

bundle exec rake db:migrate RAILS_ENV=production # 生成表

gem install passenger --no-rdoc --no-ri # 安装Passenger

passenger-install-apache2-module # 安装Apache2的模块
```

```
Welcome to the Phusion Passenger Apache 2 module installer, v5.0
.6.
```

This installer will guide you through the entire installation process. It shouldn't take more than 3 minutes in total.

Here's what you can expect from the installation process:

1. The Apache 2 module will be installed for you.
2. You'll learn how to configure Apache.
3. You'll learn how to deploy a Ruby on Rails application.

Don't worry if anything goes wrong. This installer will advise you on how to solve any problems.

Press Enter to continue, or Ctrl-C to abort.

```
1 # 指定“1”并回车
```

```
....
```

```
....
```

```
After you restart Apache, you are ready to deploy any number of web
applications on Apache, with a minimum amount of configuration!
```

```
....
```

为Redmine配置httpd。本例演示配置为虚拟主机：

编辑 `/etc/httpd/conf.d/passenger.conf` 文件：

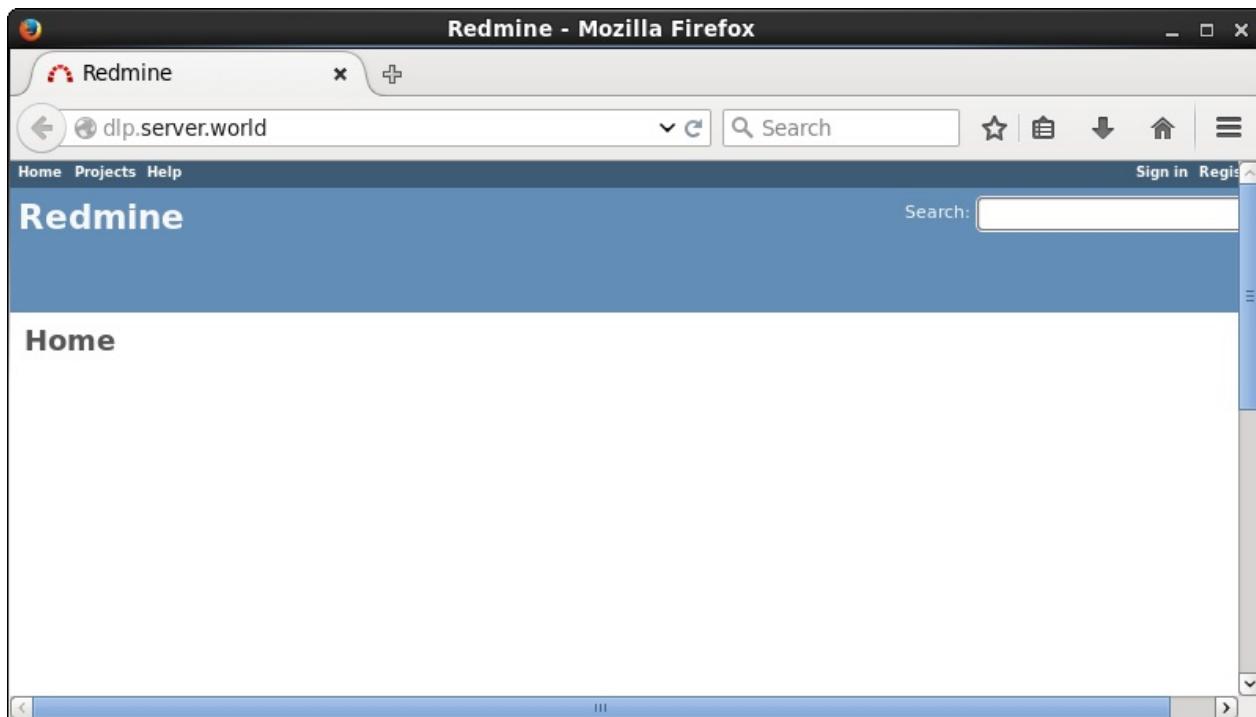
```
LoadModule passenger_module /usr/lib64/ruby/gems/2.2.0/gems/passenger-5.0.13/buildout/apache2/mod_passenger.so
PassengerRoot /usr/lib64/ruby/gems/2.2.0/gems/passenger-5.0.13
PassengerDefaultRuby /usr/bin/ruby

NameVirtualHost *:80
<VirtualHost 80>
    ServerName dlp.srv.world
    DocumentRoot /var/www/redmine/public
</VirtualHost>
```

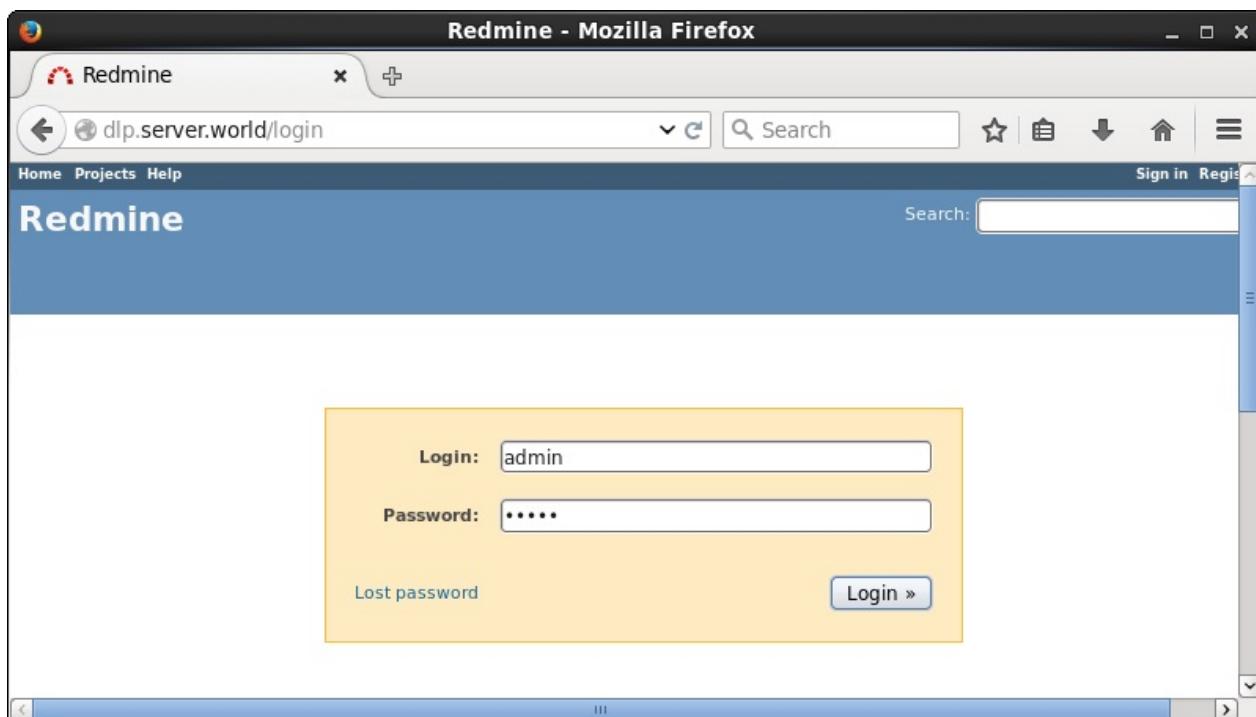
```
chown -R apache. /var/www/redmine
```

```
systemctl restart httpd
```

访问在httpd上配置的URL，然后Redmine的主页如下所示。点击“Sing in”按钮：



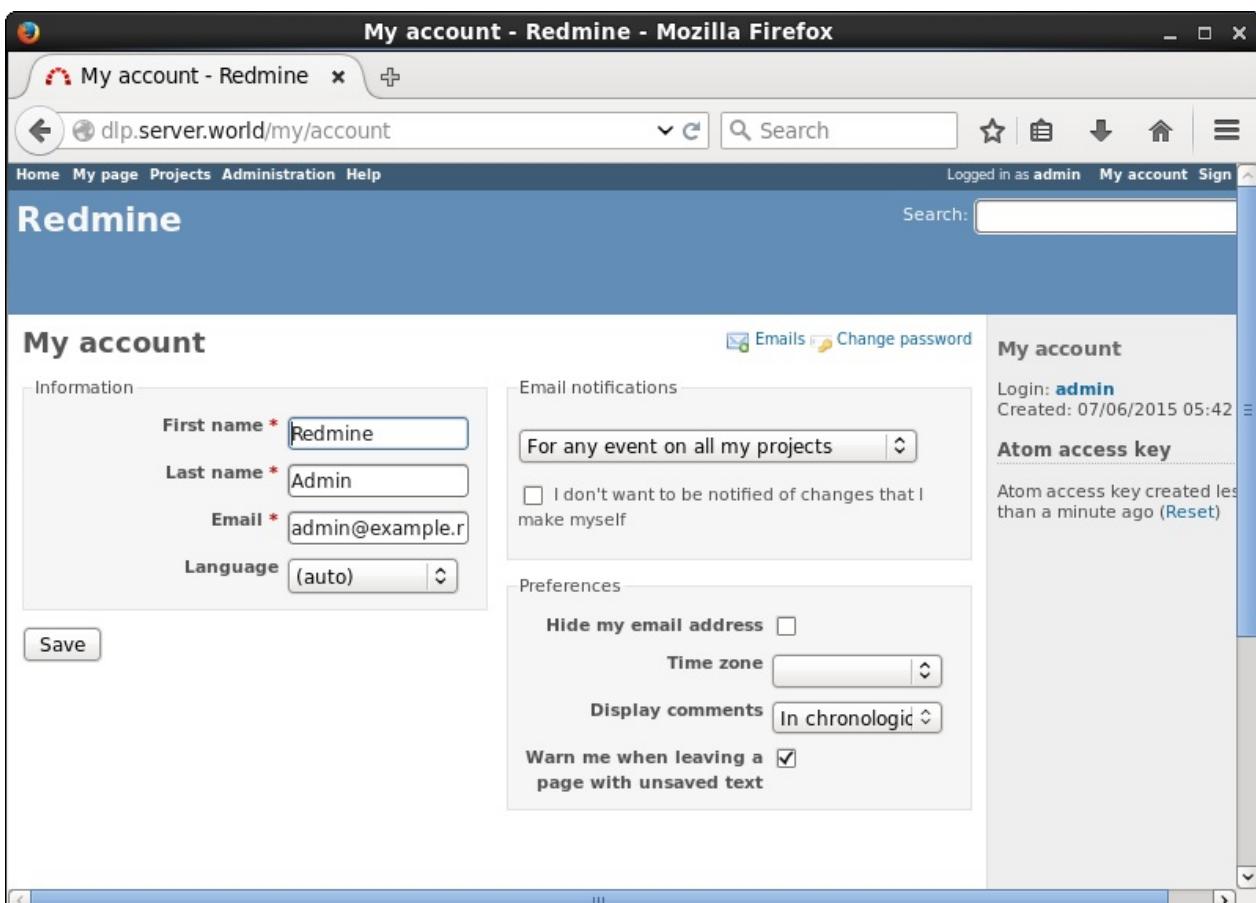
使用初始用户名/密码“admin/admin”登录：



登录成功，首先更改管理员密码 点击“My account”：



点击“admin”：



点击“edit”：

The screenshot shows a Mozilla Firefox browser window with the title "Redmine Admin - Redmine - Mozilla Firefox". The address bar contains the URL "dlp.server.world/users/1". The main content area displays a user profile for "admin". The profile includes the following information:

- Login: admin
- Email: admin@example.net
- Registered on: 07/06/2015
- Last connection: 07/07/2015

On the right side of the profile, there is a "Search:" input field and a green "Edit" link. The browser's standard navigation and search controls are visible at the top and bottom of the window.

在“Authentication”部分输入任意密码并保存更改

The screenshot shows the Redmine application running in Mozilla Firefox. The title bar reads "admin - Users - Redmine - Mozilla Firefox". The address bar shows the URL "dlp.server.world/users/1/edit". The main content area is titled "Users » admin". It has two tabs: "General" (selected) and "Projects". The "General" tab contains sections for "Information" and "Authentication". The "Information" section includes fields for Login (admin), First name (Redmine), Last name (Admin), Email (admin@example.r), Language (auto), and Administrator (checked). The "Authentication" section includes fields for Password and Confirmation, with a note that the password must be at least 8 characters. There are also checkboxes for "Generate password" and "Must change password at next logon". Below these sections is a checkbox for "Send account information to the user" and a "Save" button. To the right of the main form is a sidebar titled "Administration" with a tree view of Redmine modules: Projects, Users (selected), Groups, Roles and permissions, Trackers, Issue statuses, Workflow, Custom fields, Enumerations, Settings, LDAP authentication, Plugins, and Information.

### 附1.6.3. Gitolite

Gitolite 是一款 Perl 语言开发的 Git 服务管理工具，通过公钥对用户进行认证，并能够通过配置文件对写操作进行基于分支和路径的精细授权。

#### 附1.6.3.1. 安装Gitolite

```
yum --enablerepo=epel -y install gitolite3 #从EPEL安装
```

生成SSH密钥对，并使用Gitolite管理员用户设置Gitolite：

```
su - gitolite3
```

```
Generating public/private rsa key pair.
```

```

Enter passphrase (empty for no passphrase): # 设置密码
Enter same passphrase again: # 确认密码
Your identification has been saved in /var/lib/gitolite3/.ssh/gitadmin.
Your public key has been saved in /var/lib/gitolite3/.ssh/gitadmin.pub.
The key fingerprint is:
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx gitolite3@dlp.srv.world
The key's randomart image is:

-sh-4.2$ gitolite setup -pk ~/.ssh/gitadmin.pub
Initialized empty Git repository in /var/lib/gitolite3/repositories/gitolite-admin.git/
Initialized empty Git repository in /var/lib/gitolite3/repositories/testing.git/

-sh-4.2$ vi ~/.ssh/config

# 任意名称
host GitServer
    user gitolite3
    # Git服务器的主机名或IP地址
    hostname 10.0.0.30
    port 22
    # 密钥
    identityfile ~/.ssh/gitadmin

-sh-4.2$ chmod 600 ~/.ssh/config
-sh-4.2$ git config --global user.name "gitolite3"
-sh-4.2$ git config --global user.email "gitolite3@srv.world"
-sh-4.2$ git config --global push.default simple

# 克隆管理库以完成设置
-sh-4.2$ git clone ssh://GitServer/gitolite-admin
Cloning into 'gitolite-admin'...
The authenticity of host '10.0.0.30 (10.0.0.30)' can't be established.
ECDSA key fingerprint is xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.30' (ECDSA) to the list of kn

```

```
own hosts.  
Enter passphrase for key '/var/lib/gitolite3/.ssh/gitadmin':  
remote: Counting objects: 6, done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 6 (delta 0), reused 0 (delta 0)  
Receiving objects: 100% (6/6), done.
```

### 附1.6.3.2. 添加用户

以要为Gitolite设置的用户登录系统并生成SSH密钥对：

```
ssh-keygen -f ~/.ssh/id_cent
```

```
Generating public/private rsa key pair.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/cent/.ssh/id_cent.  
Your public key has been saved in /home/cent/.ssh/id_cent.pub.  
The key fingerprint is:  
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx cent@dlp.srv.wor  
ld  
The key's randomart image is:
```

将上面生成的SSH公钥传给Gitolite管理员用户。接下来，使用Gitolite管理员如下添加用户：

```
-sh-4.2$ whoami
gitolite3
-sh-4.2$ cd ~/gitolite-admin/keydir
-sh-4.2$ git add id_cent.pub
-sh-4.2$ git commit -m "Add User cent"
1 file changed, 1 insertion(+)
create mode 100644 keydir/id_cent.pub
-sh-4.2$ git push origin master
Enter passphrase for key '/var/lib/gitolite3/.ssh/gitadmin':
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 686 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To ssh://GitServer/gitolite-admin
  1d61702..8d46625 master -> master
```

确认使用刚添加的用户克隆库：

编辑 `~/.ssh/config` 文件：

```
# 任意名称
host GitServer
    user gitolite3
    # Git服务器的主机名或IP地址
    hostname 10.0.0.30
    port 22
    # 密钥
    identityfile ~/.ssh/id_cent
```

```
chmod 600 ~/.ssh/config
```

```
git config --global user.name "cent"
```

```
git config --global user.email "cent@srv.world"
```

```
git clone ssh://GitServer/testing
```

```
Cloning into 'testing'...
warning: You appear to have cloned an empty repository.
```

11

```
total 0
drwxrwxr-x 3 cent cent 17 Jul 21 20:50 testing
```

### 附1.6.3.3. 添加新库

例如，使用Gitolite管理员添加一个新的库“public-repo”：

```
-sh-4.2$ vi ~/gitolite-admin/conf/gitolite.conf

# 添加到最后
repo public-repo
    RW+      =  @all

-sh-4.2$ cd ~/gitolite-admin
-sh-4.2$ git commit -a -m "Add public-repo repository"
[master 5aebeb2] Add public-repo repository
1 file changed, 4 insertions(+)
-sh-4.2$ git push
Enter passphrase for key '/var/lib/gitolite3/.ssh/gitadmin':
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 391 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote: Initialized empty Git repository in /var/lib/gitolite3/repositories/public-repo.git/
To ssh://GitServer/gitolite-admin
  8d46625..5aebeb2 master -> master
```

确认用户可以克隆刚添加的库：

```
ssh GitServer # 显示当前用户可以访问的库列表
```

```
PTY allocation request failed on channel 0
hello id_cent, this is gitolite3@dlp running gitolite3 3.6.3-1.e
17 on git 1.8.3.1

R W      public-repo
R W      testing
Connection to 10.0.0.30 closed.
```

```
git clone ssh://GitServer/public-repo
```

```
Cloning into 'public-repo'...
warning: You appear to have cloned an empty repository.
```

```
ll
```

```
total 0
drwxrwxr-x 3 cent cent 17 Jul 21 21:06 public-repo
drwxrwxr-x 3 cent cent 17 Jul 21 20:50 testing
```

### 附1.6.3.4. 设置访问控制

使用Gitolite管理员设置库的访问控制。

例如，将用户“cent”的读取和写入权限添加到“public-repo”：

```
-sh-4.2$ cd ~/gitolite-admin/conf
-sh-4.2$ vi gitolite.conf

# 在下面指定用户名格式的SSH密钥文件名
repo public-repo
    RW+      =    id_cent

-sh-4.2$ git commit -a -m "Change Permission for public-repo"
-sh-4.2$ git push
```

例如，将组“developer”的读取和写入权限添加到“public-repo”：

```
-sh-4.2$ cd ~/gitolite-admin/conf  
-sh-4.2$ vi gitolite.conf  
  
@developer = id_cent id_ubuntu  
  
repo public-repo  
    RW+      =  @developer  
  
-sh-4.2$ git commit -a -m "Change Permission for public-repo"  
-sh-4.2$ git push
```

例如，添加访问权限如下：

- 将用户“redhat”添加读/写权限到所有库
- 将组“deployer”添加读/写权限添加到“prod”分支
- 将组“deployer”添加读/写权限添加到“dlp”分支

```
-sh-4.2$ cd ~/gitolite-admin/conf  
-sh-4.2$ vi gitolite.conf  
  
@developer = id_cent id_ubuntu  
@deployer = id_debian id_fedora  
  
repo public-repo  
    RW+      =  id_redhat  
    RW prod =  @deployer  
    RW dlp   =  @developer  
  
-sh-4.2$ git commit -a -m "Change Permission for public-repo"  
-sh-4.2$ git push
```

## 附1.7. 系统管理工具

### 附1.7.1. Cockpit

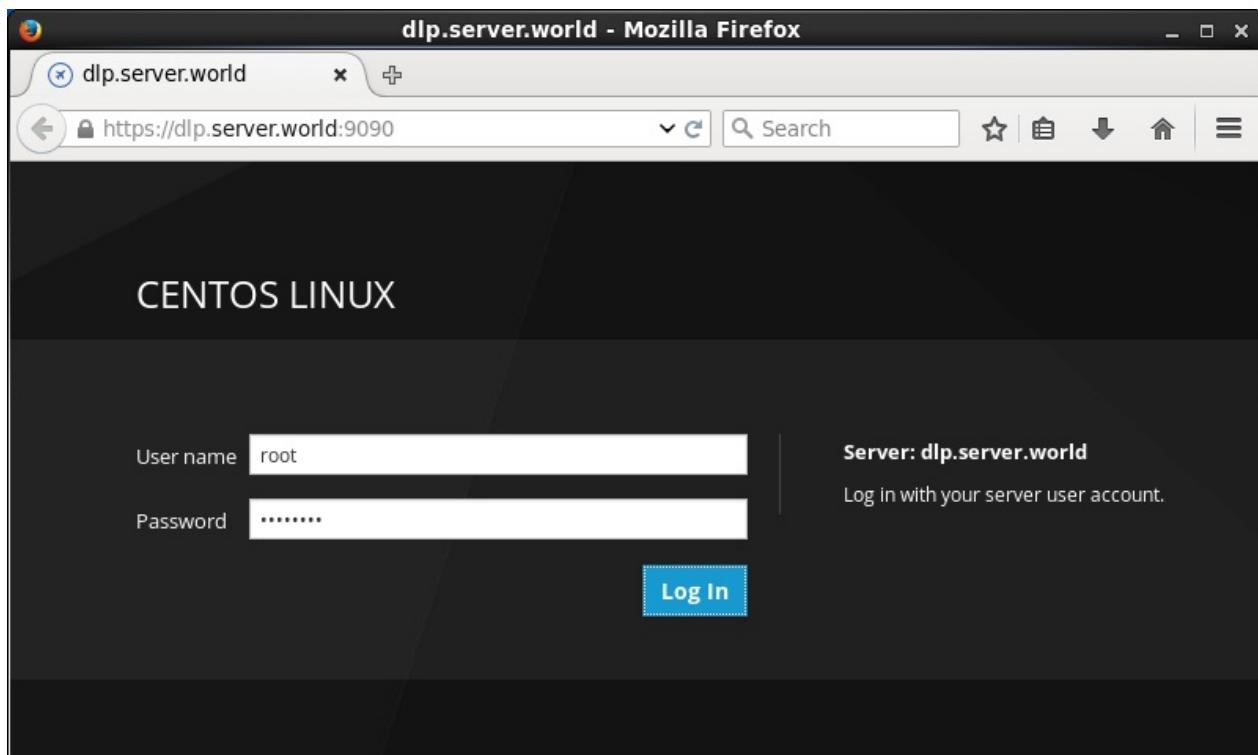
Cockpit是红帽开发的网页版图像化服务管理工具，优点是无需中间层，且可以管理多种服务。

安装Cockpit：

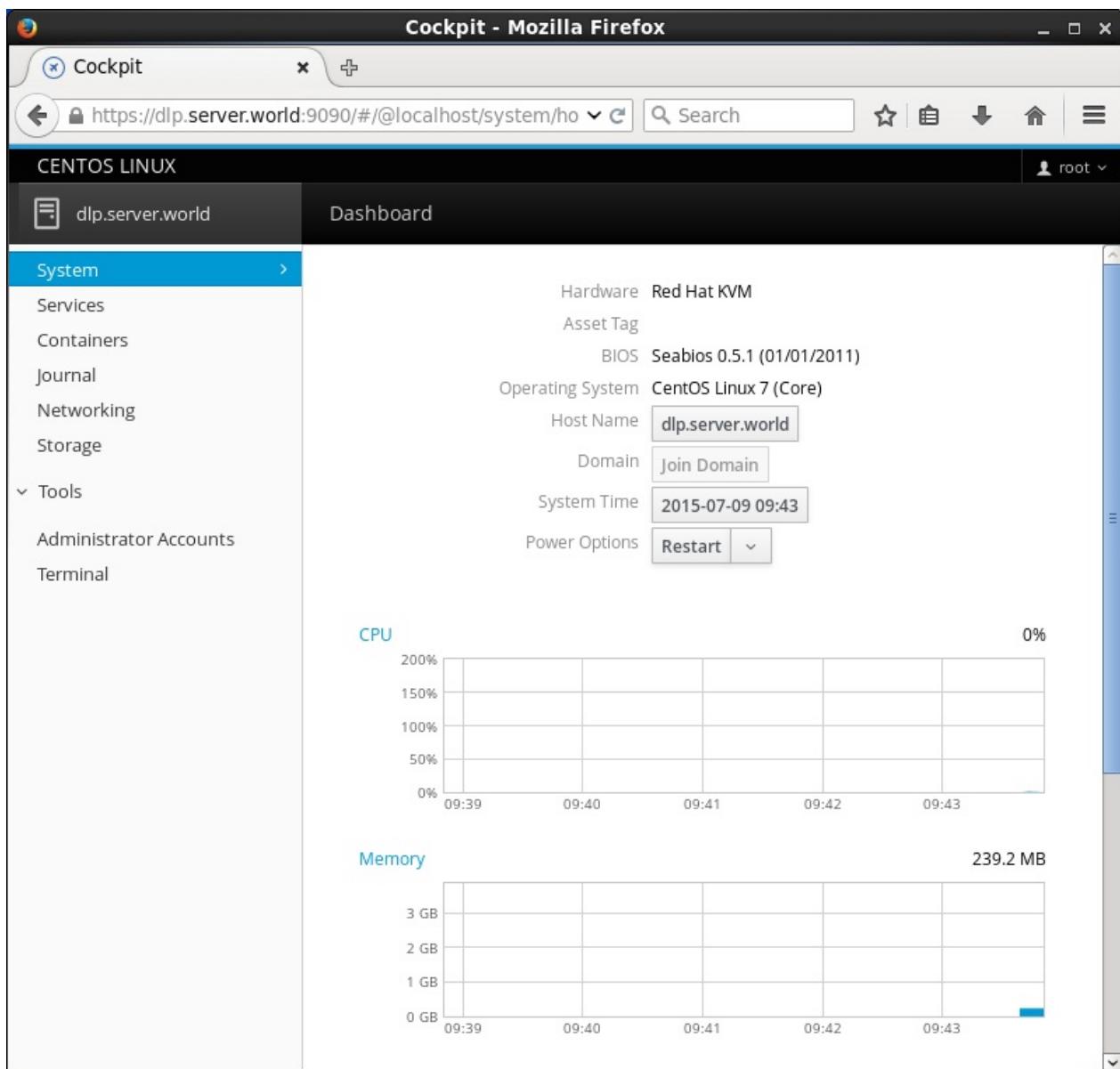
```
yum -y install cockpit
```

```
systemctl start cockpit  
systemctl enable cockpit.socket
```

使用客户端Web浏览器访问 [https://\(服务器的主机名或IP地址\):9090/](https://(服务器的主机名或IP地址):9090/)，显示Cockpit登录界面。本例以用户“root”登录：



下面是Cockpit主页。可以在这里管理系统：



左侧菜单中的“Services”，可以管理或操作系统服务：

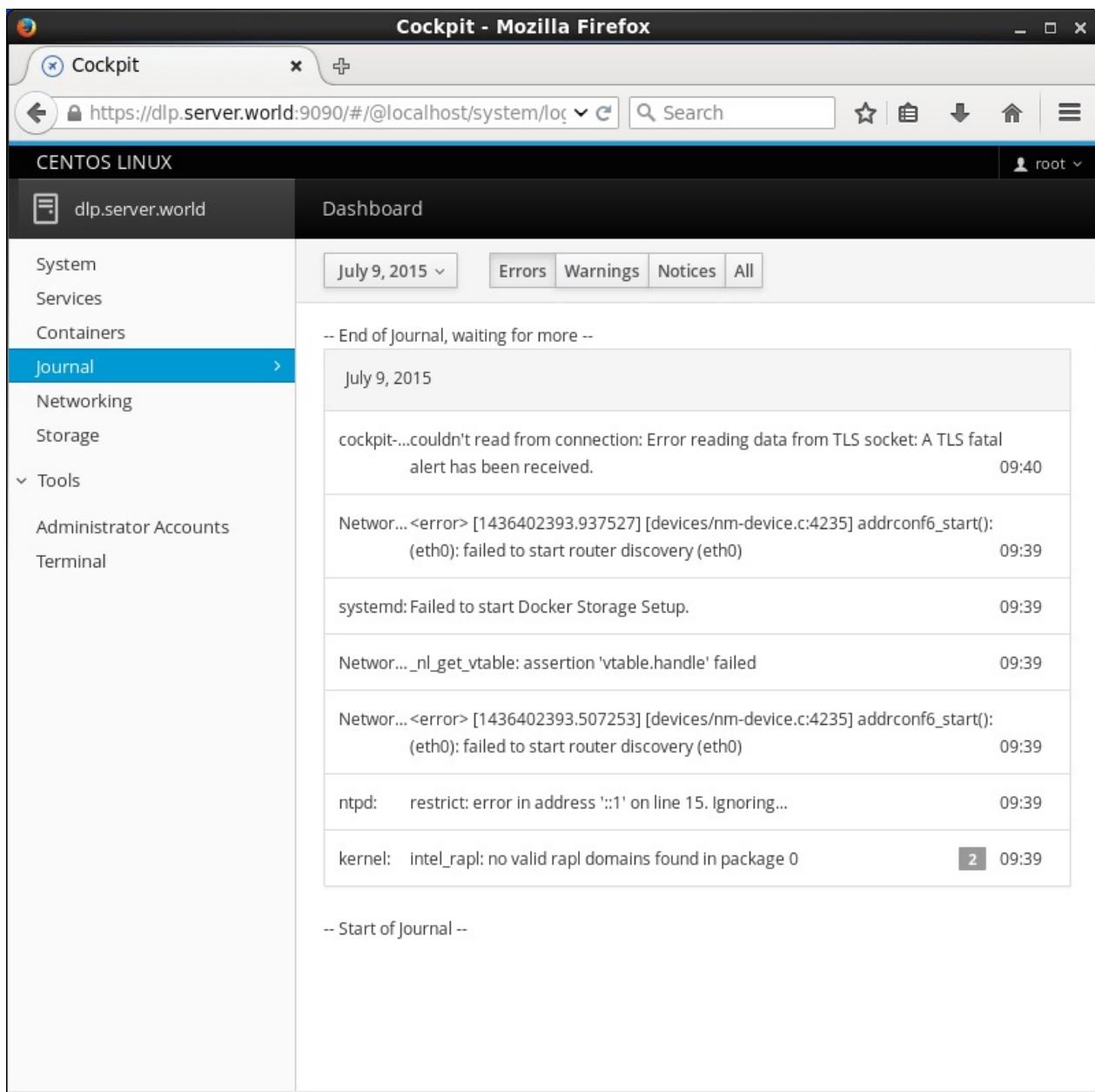
The screenshot shows the Cockpit web interface running in Mozilla Firefox. The title bar says "Cockpit - Mozilla Firefox". The address bar shows the URL "https://dlp.server.world:9090/#/@localhost/system/ini". The top navigation bar has tabs for "Targets", "System Services", "Sockets", "Timers", and "Paths", with "System Services" currently selected. The left sidebar menu includes "System", "Services" (which is expanded to show "Containers", "Journal", "Networking", "Storage", "Tools" (with "Administrator Accounts" and "Terminal" listed), and "Containers"). The main content area is titled "Dashboard" and "Enabled". It lists various system services with their status:

| Service                                                                       | Unit                              | Status           |
|-------------------------------------------------------------------------------|-----------------------------------|------------------|
| Security Auditing Service                                                     | auditd.service                    | active (running) |
| Command Scheduler                                                             | crond.service                     | active (running) |
| Docker Application Container Engine                                           | docker.service                    | active (running) |
| getty@.service Template                                                       | getty@.service                    |                  |
| irqbalance daemon                                                             | irqbalance.service                | active (running) |
| Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling | lvm2-monitor.service              | active (exited)  |
| Software RAID monitoring and management                                       | mdmonitor.service                 | inactive (dead)  |
| Load CPU microcode update                                                     | microcode.service                 | inactive (dead)  |
| Network Manager Script Dispatcher Service                                     | NetworkManager-dispatcher.service | inactive (dead)  |
| Network Manager                                                               | NetworkManager.service            | active (running) |

左侧菜单中的“Containers”，可以管理或操作Docker容器：

The screenshot shows the Cockpit web interface running in Mozilla Firefox. The URL is `https://dlp.server.world:9090/#/localhost/docker/containers`. The interface has a dark-themed header with the title "Cockpit - Mozilla Firefox". The left sidebar contains navigation links: System, Services, **Containers**, Journal, Networking, Storage, Tools (Administrator Accounts, Terminal), and a user icon for root. The main dashboard area displays several sections: "Combined memory usage" (empty grid), "Containers" (list of four containers: backstabbi..., dreamy\_rit..., drunk\_elion, elated\_kirch, all running centos:latest images), "Storage space" (513.6 MB / 27.3 GB), and "Images" (list of one image: docker.io/centos:latest, created 6/19/2015, size 164.3 MB).

左侧菜单中的“Journal”，可以管理或操作系统日志：



左侧菜单中的“Networking”，可以管理或操作网络：

The screenshot shows the Cockpit web interface running in Mozilla Firefox. The URL is `https://dlp.server.world:9090/#/localhost/network/in`. The interface has a dark-themed header with the title "Cockpit - Mozilla Firefox". The left sidebar lists system management options: System, Services, Containers, Journal, Networking (selected), Storage, Tools (Administrator Accounts, Terminal), and a user icon for root.

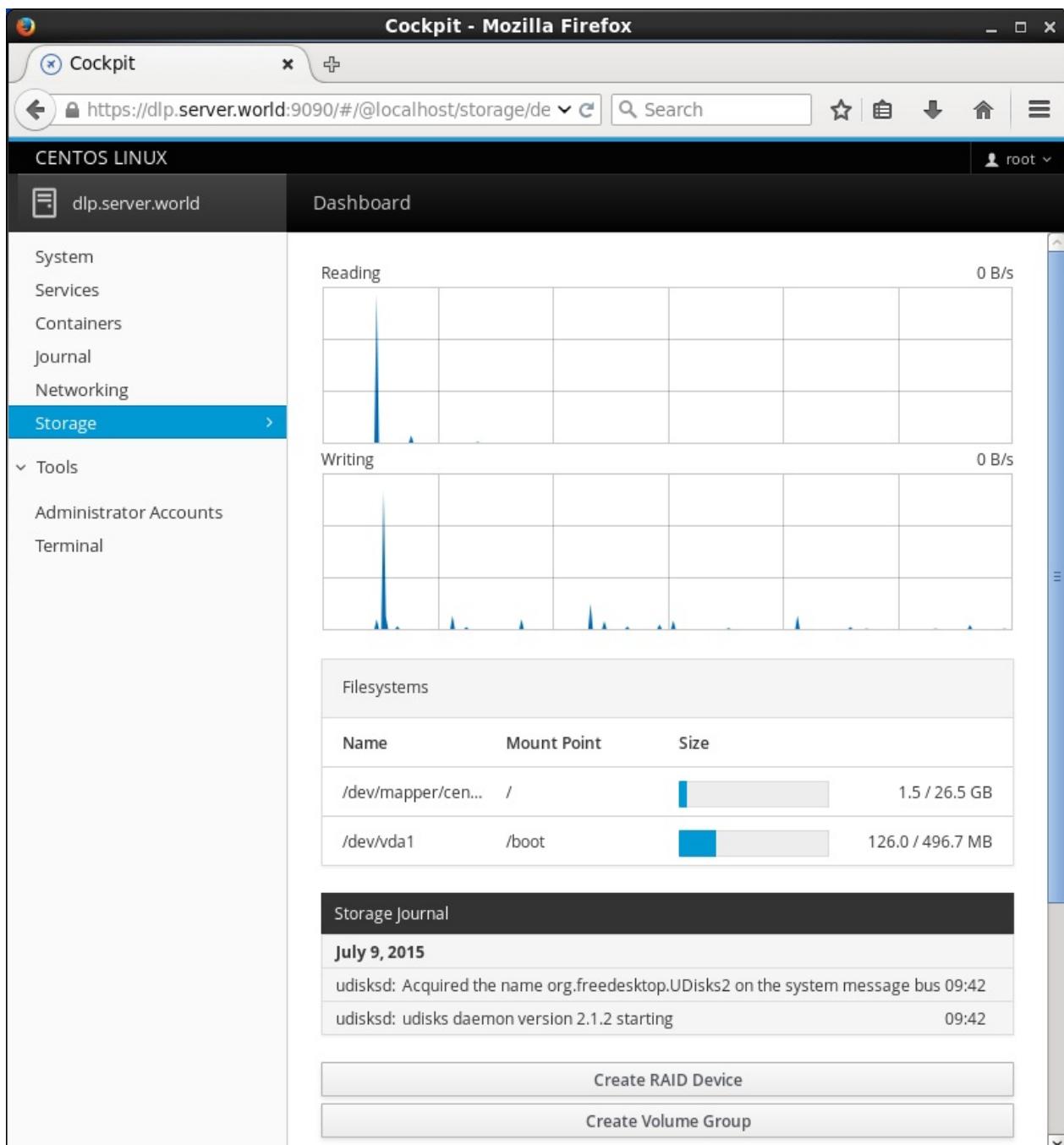
The main dashboard displays network traffic graphs for "Sending" and "Receiving" on two interfaces. The "Sending" graph shows a single sharp peak at 10.6 Kbps. The "Receiving" graph shows a small peak at 1.5 Kbps. Below the graphs is a table of network interfaces:

| Name    | IP Address     | Sending    | Receiving |
|---------|----------------|------------|-----------|
| docker0 | 172.17.42.1/16 | No carrier |           |
| eth0    | 10.0.0.30/24   | 10.6 Kbps  | 1.5 Kbps  |

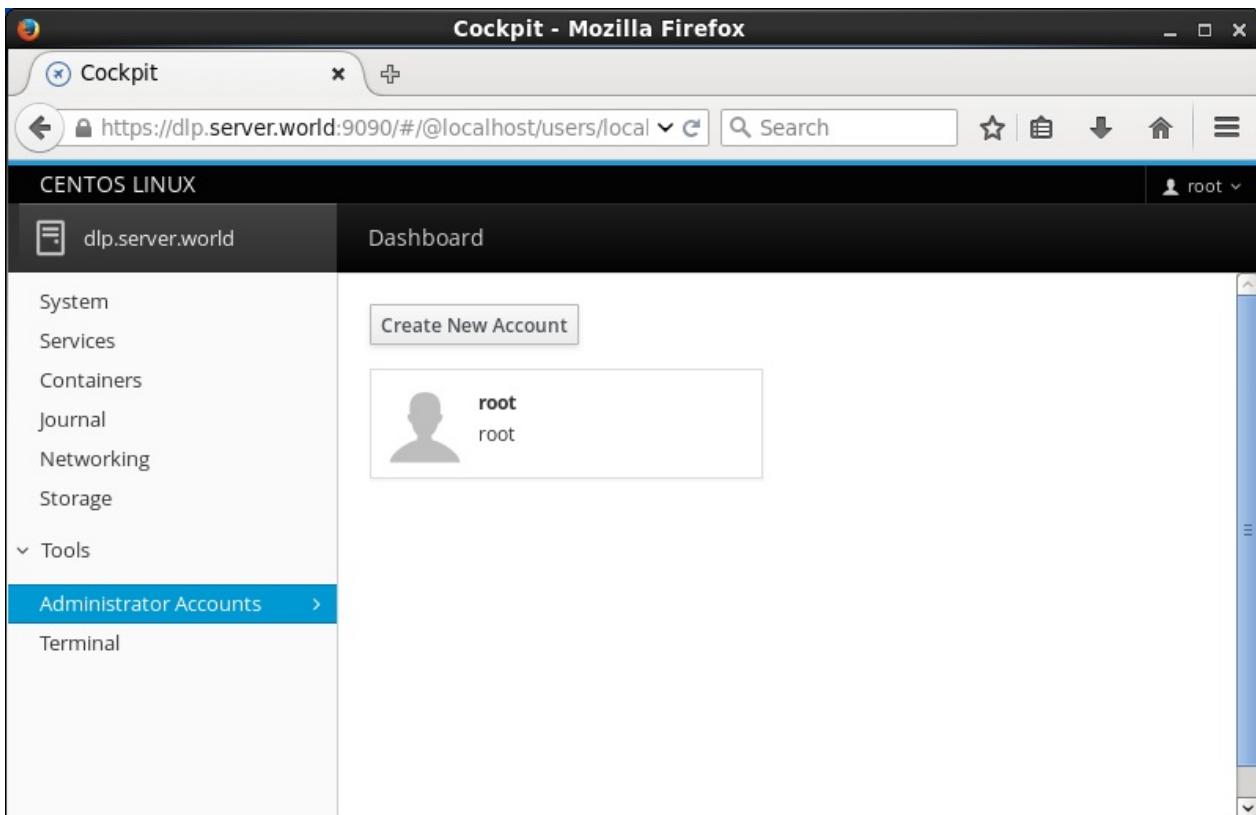
At the bottom, there is a "Networking Journal" section showing log entries for July 9, 2015:

| Date         | Log Entry                                                                                | Time  |
|--------------|------------------------------------------------------------------------------------------|-------|
| July 9, 2015 | Networ...<info> startup complete                                                         | 09:40 |
|              | Networ...<info> (docker0): Activation: successful, device activated.                     | 09:39 |
|              | Networ...<info> (docker0): device state change: secondaries -> activated (reason 'none') |       |

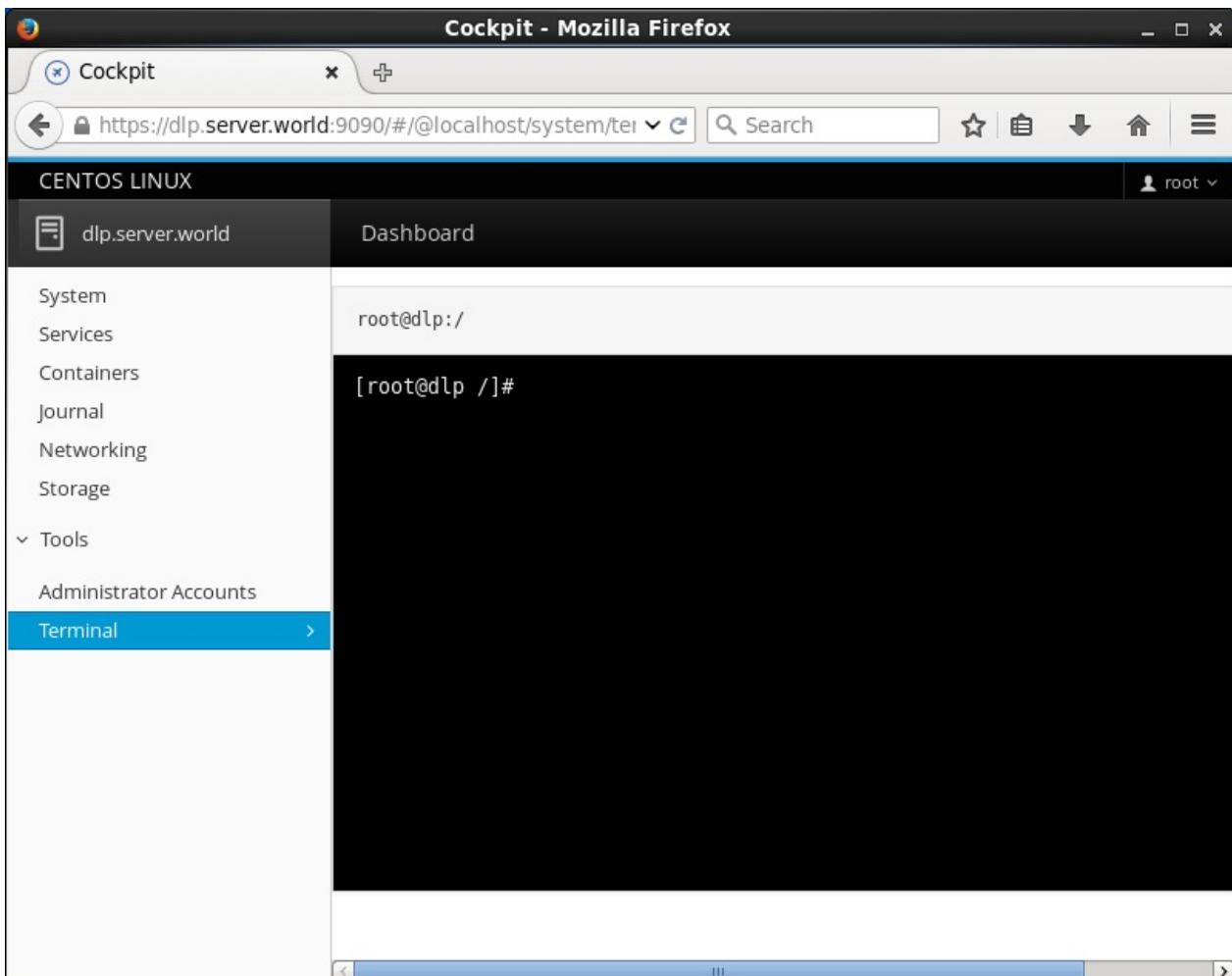
左侧菜单中的“Storage”，可以管理或操作存储：



左侧菜单中的“Administrator Accounts”，可以管理或操作系统帐户：



左侧菜单中的“Terminal”，可以直接使用命令来运行系统：



## 附1.7.2. Ajenti

Ajenti包

含Apache，BIND9，Cron，CTDB，DHCPD，NFSD，Iptables，Munin，MySQL，Netatalk，NGINX，PostgreSQL，Samba，Im-sensors，Squid 3，Supervisor等的管理。并可以便捷的进行二次开发，完全开源。

安装Ajenti，可以参考官网：

```
yum -y install http://repo.ajenti.org/ajenti-repo-1.0-1.noarch.rpm
```

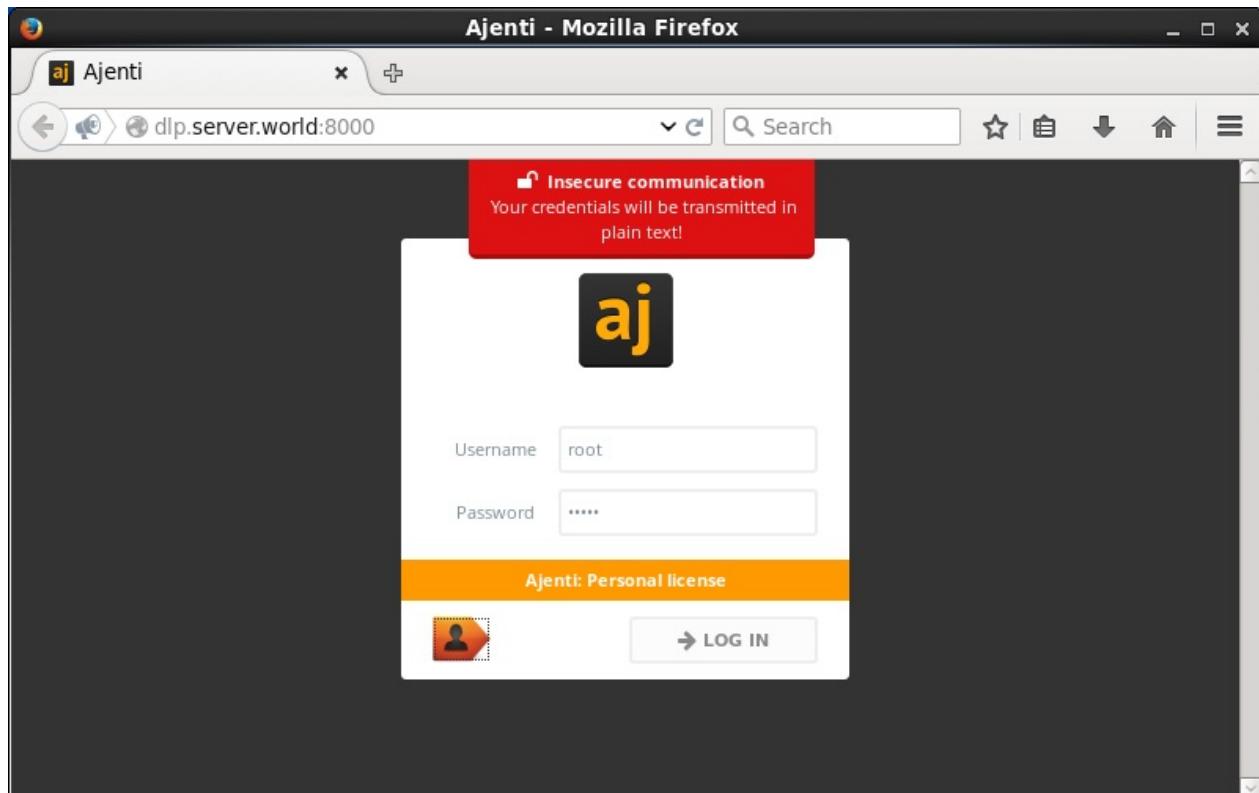
```
yum --enablerepo=epel -y install ajenti # 同时开启EPEL
```

编辑 /etc/ajenti/config.json 文件：

```
# 禁用SSL（不适用于CentOS 7.2）
"ssl": {
    "enable": false,
```

```
systemctl restart ajenti
```

使用客户端Web浏览器访问 `http://(服务器的主机名或IP地址):8000/`，显示Ajenti登录界面。使用默认用户“root”，密码“admin”登录：



下面是Ajenti主页。可以在这里管理系统：

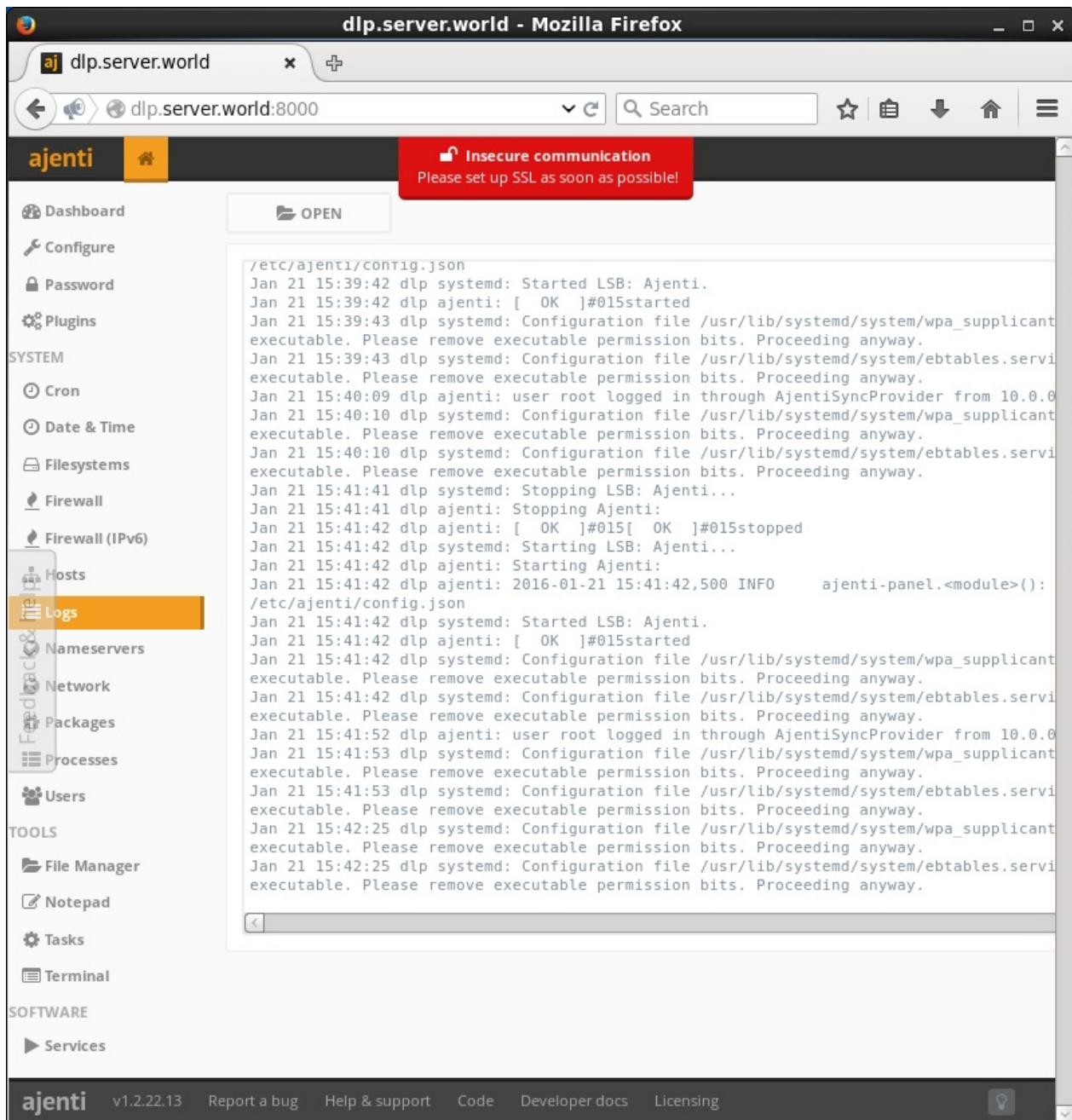
The screenshot shows the Ajenti web interface running in Mozilla Firefox. The URL is `d1p.server.world:8000`. A red banner at the top right reads "Insecure communication Please set up SSL as soon as possible!". The left sidebar contains a navigation menu with sections: Dashboard, SYSTEM (Cron, Date & Time, Filesystems, Firewall, Firewall (IPv6), Hosts, Logs, Nameservers, Network, Packages, Processes, Users), TOOLS (File Manager, Notepad, Tasks, Terminal), and SOFTWARE (Services). The main content area displays system status: Uptime (0:29:50), AC power controls, CPU usage (6% and 7%), Memory usage (175.1 MB), Swap usage (0.0 bytes), and a welcome message: "Welcome to Ajenti. Use the Feedback link to send us your suggestions and ideas." It also shows social media links for Twitter (@ajenti) with 1,899 followers and a link to send email. At the bottom, there's a note to "Don't forget to change default password!" and footer links for ajenti v1.2.22.13, Report a bug, Help & support, Code, Developer docs, and Licensing.

左侧菜单中的“Services”，可以管理或操作系统服务：

The screenshot shows a Mozilla Firefox browser window with the URL `dip.server.world:8000`. The title bar says "dip.server.world - Mozilla Firefox". The page is titled "ajenti". A red banner at the top right reads "Insecure communication Please set up SSL as soon as possible!". The left sidebar has a "Services" section highlighted with an orange background. The main content area is a table with a header "Name" and a search bar "Filter...". The table lists several system services:

| Name                       |
|----------------------------|
| NetworkManager             |
| NetworkManager-dispatcher  |
| NetworkManager-wait-online |
| ajenti                     |
| audited                    |
| autovt@                    |
| blk-availability           |
| brandbot                   |
| console-getty              |
| console-shell              |
| container-getty@           |
| cpupower                   |

左侧菜单中的“Logs”，可以管理或操作系统日志：



左侧菜单中的“Network”，可以管理或操作网络：

The screenshot shows the ajenti web interface running in Mozilla Firefox. The URL is `dip.server.world:8000`. A red banner at the top right reads "Insecure communication Please set up SSL as soon as possible!". The left sidebar menu includes: Dashboard, Configure, Password, Plugins, SYSTEM (Cron, Date & Time), Filesystems (selected), Firewall, Firewall (IPv6), Hosts, Logs, Nameservers, Network (selected), Packages, Processes, and Users. The TOOLS section includes: File Manager, Notepad, Tasks, and Terminal. The SOFTWARE section includes: Services. The main content area displays a table of network interfaces:

|   | Name | IP        | TX      |
|---|------|-----------|---------|
| x | lo   | 127.0.0.1 | 828.0   |
| x | eth0 | 10.0.0.30 | 3.1 MiB |

A "SAVE" button is visible below the table.

左侧菜单中的“Filesystems”，可以管理或操作文件系统：

The screenshot shows the ajenti web interface running in Mozilla Firefox. The URL is `d1p.server.world:8000`. A red banner at the top right reads "Insecure communication Please set up SSL as soon as possible!". The left sidebar has a "Filesystems" section highlighted. The main content area shows the `fstab` configuration and mounted file systems.

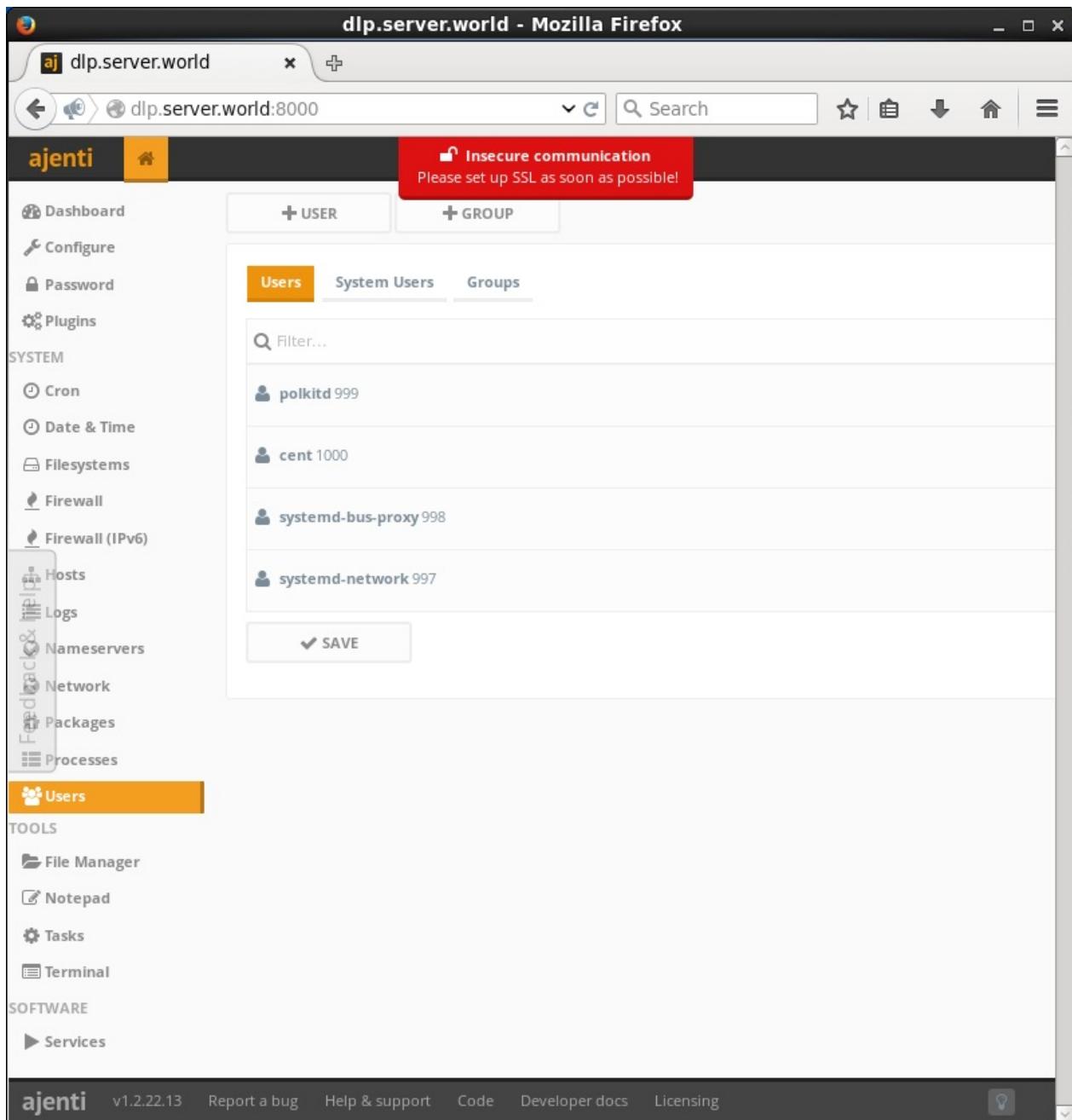
**fstab**

| /dev/mapper/centos-root                   | → /     |
|-------------------------------------------|---------|
| UUID=c4df086e-3699-4e02-b7cf-b47e614f6920 | → /boot |
| /dev/mapper/centos-swap                   | → swap  |

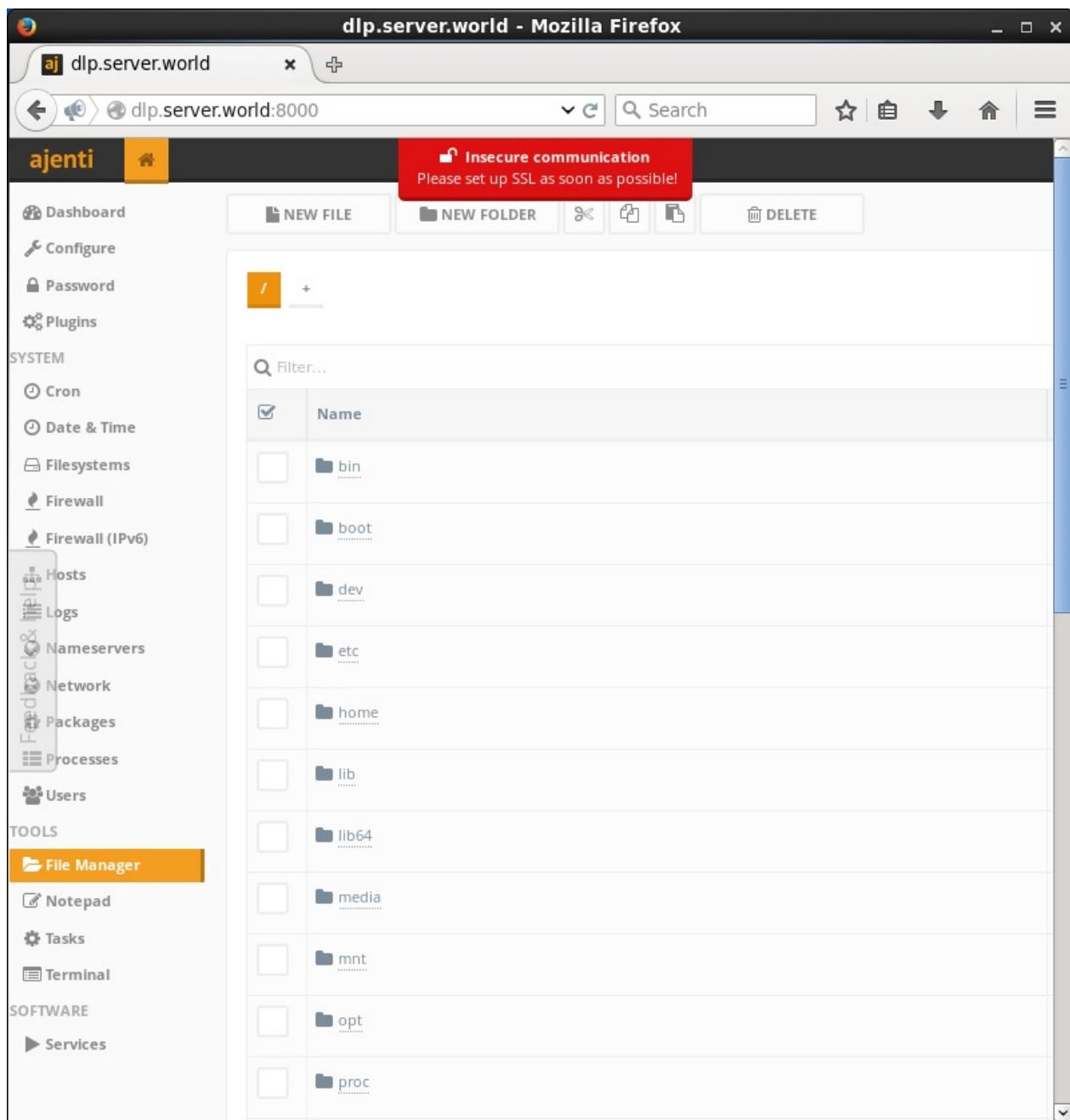
**Mounted filesystems**

| Device                  | Mountpoint     | Used      | Size     | Usage            |
|-------------------------|----------------|-----------|----------|------------------|
| /dev/mapper/centos-root | /              | 1.4 GB    | 26.5 GB  | █ (yellow)       |
| devtmpfs                | /dev           | 0.0 bytes | 1.9 GB   | ██████ (grey)    |
| tmpfs                   | /dev/shm       | 0.0 bytes | 1.9 GB   | ██████ (grey)    |
| tmpfs                   | /run           | 8.3 MB    | 1.9 GB   | ███ (orange)     |
| tmpfs                   | /sys/fs/cgroup | 0.0 bytes | 1.9 GB   | ██████ (grey)    |
| /dev/vda1               | /boot          | 203.9 MB  | 496.7 MB | ███████ (yellow) |

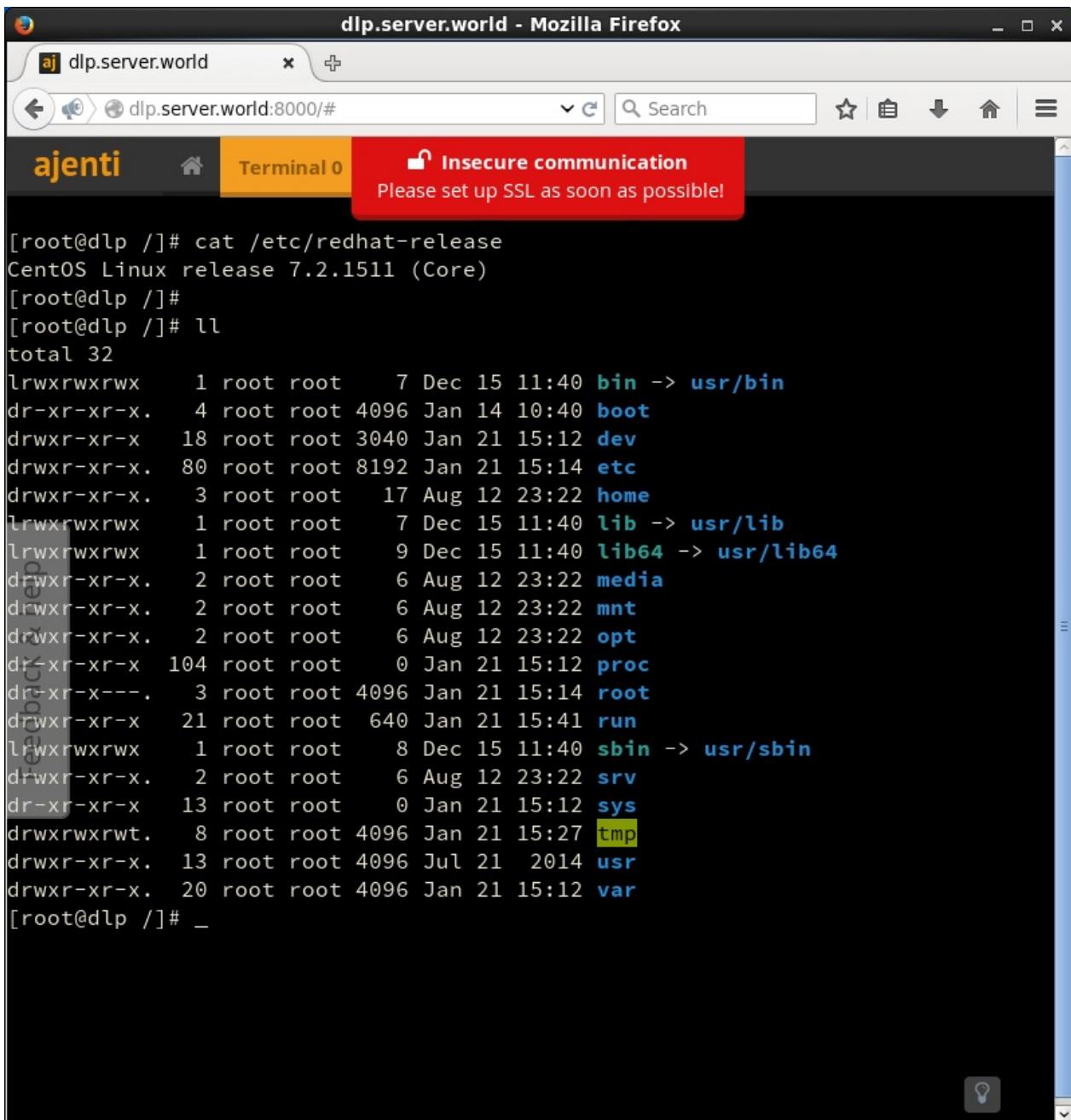
左侧菜单中的“Users”，可以管理或操作用户帐号：



左侧菜单中的“File Manager”，可以管理或操作文件或目录：



左侧菜单中的“Terminal”，可以直接使用命令来运行系统：



### 附1.7.3. Webmin

Webmin是功能强大的基于Web的Unix系统管理工具。

首先安装所需的软件包：

```
yum -y install perl-Net-SSLeay
```

安装Webmin（确认[最新版本下载连接](#)）：

```
yum -y install http://download.webmin.com/download/yum/webmin-1.750-1.noarch.rpm
```

编辑 `/etc/webmin/miniserv.conf` 文件：

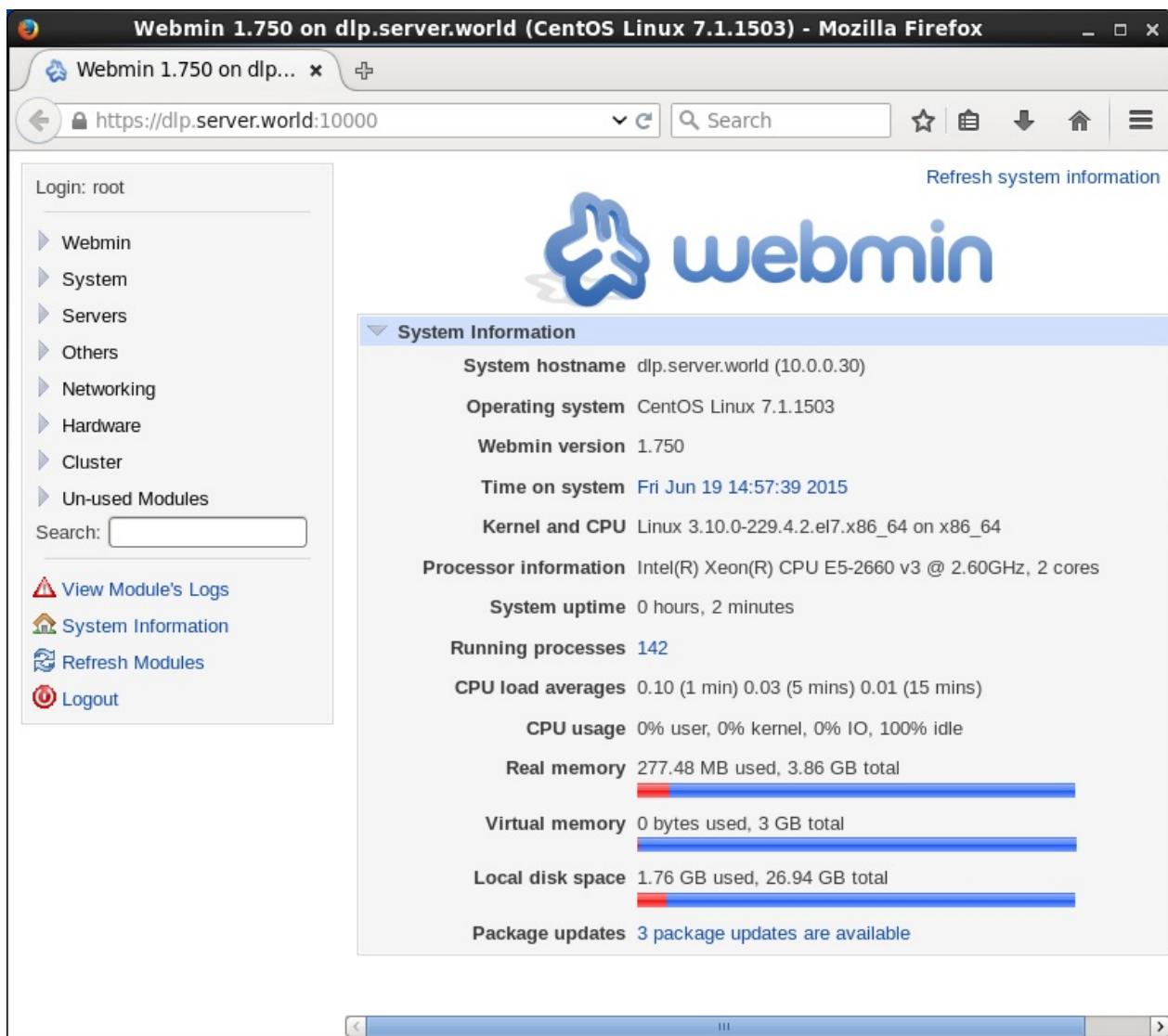
```
# 添加以下内容到最后（允许访问的IP地址）
allow=127.0.0.1 10.0.0.0/24
```

```
/etc/rc.d/init.d/webmin restart
```

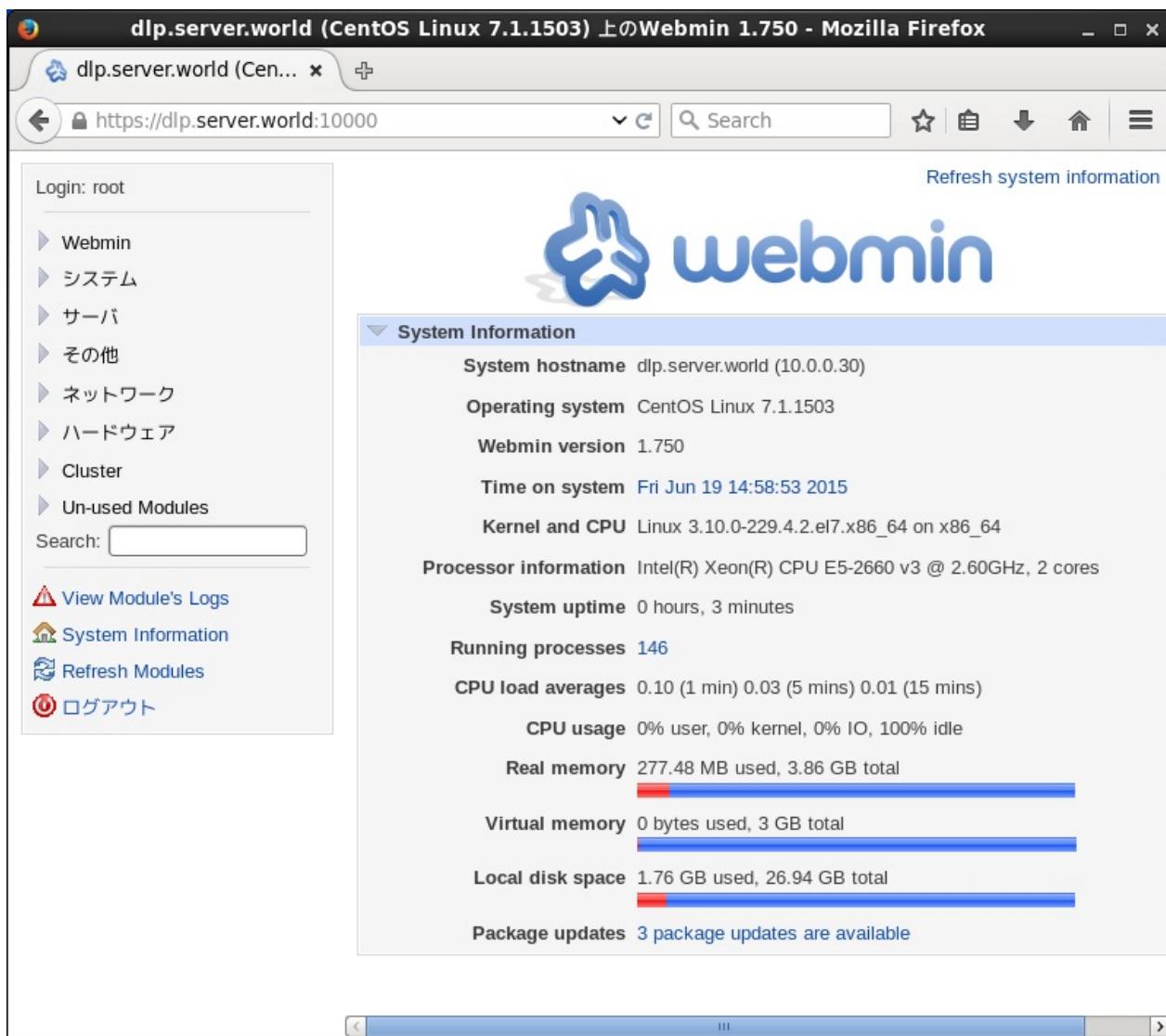
使用客户端Web浏览器访问 `https://(服务器的主机名或IP地址):10000/`，显示Webmin登录界面。使用用户“root”登录：



下面是Webmin主页。可以在这里进行操作：



点击左侧菜单上的“Webmin”->“Webmin Configuration”，然后点击右侧窗格中的“Language”，可以更改为自己的语言：



### 附1.7.4. Usermin

Usermin是一个基于Web的为Unix、Linux用户提供网络邮件，修改密码，邮件过滤器，fetchmail和更多功能。

首先安装所需的软件包：

```
yum --enablerepo=epel -y install perl-Net-SSLeay perl-Authen-PAM #  
从EPEL安装
```

安装Usermin（确认最新版本下载连接）：

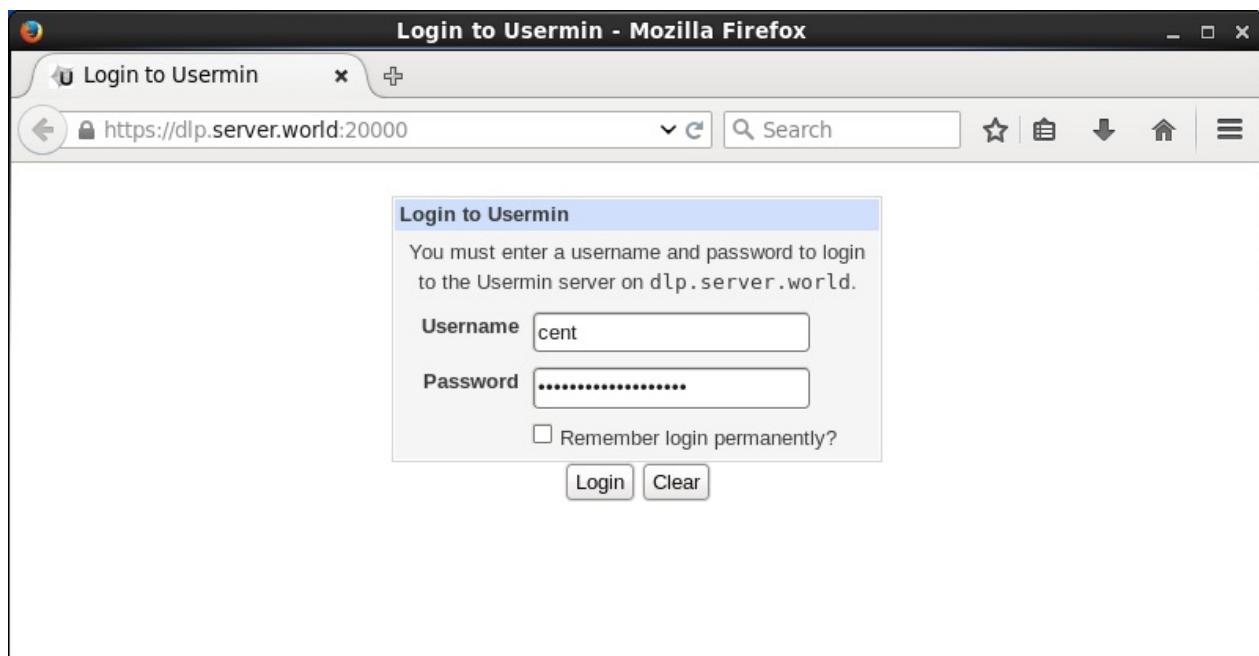
```
yum -y install http://download.webmin.com/download/yum/usermin-  
1.661-1.noarch.rpm
```

编辑 /etc/usermin/miniserv.conf 文件：

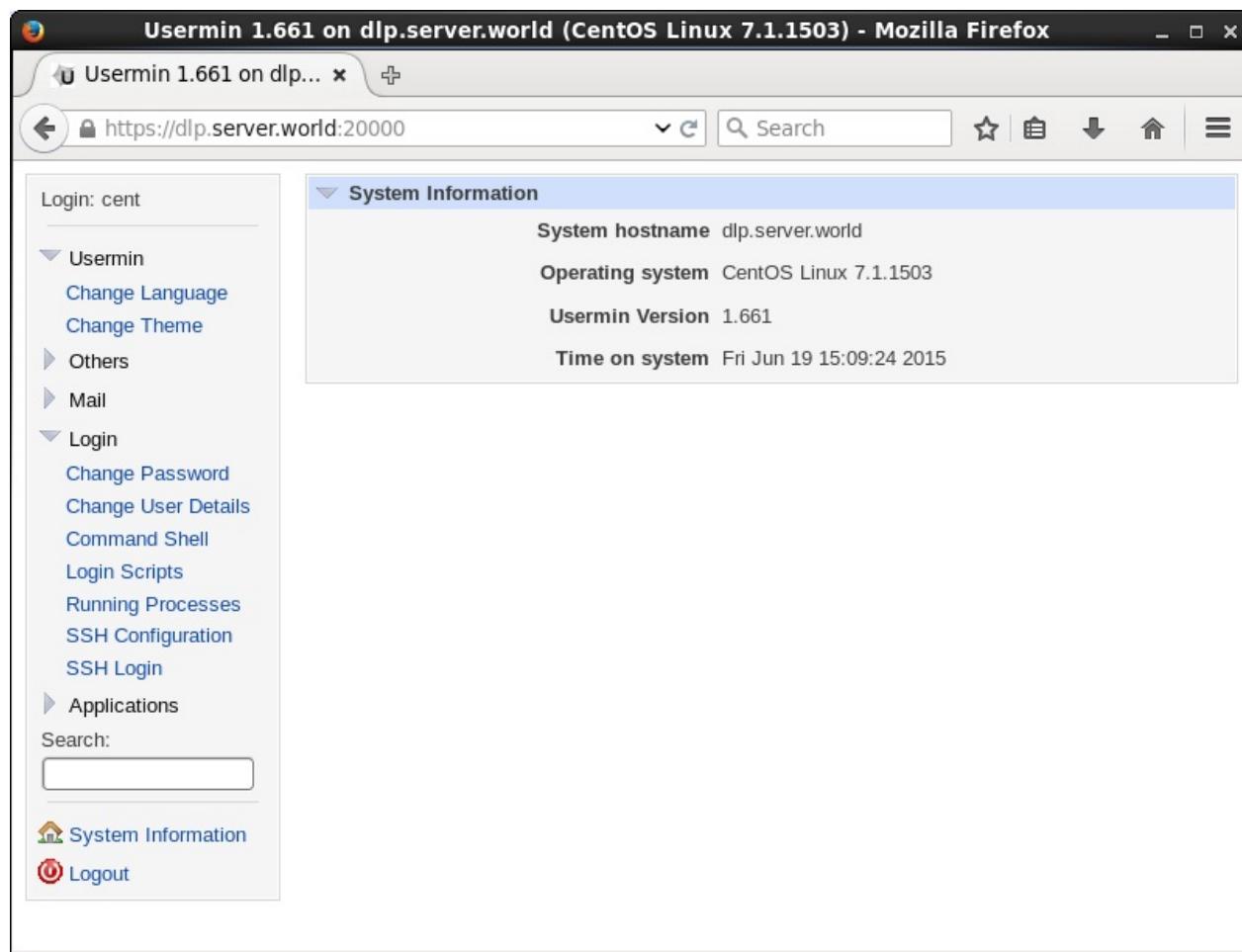
```
# 添加以下内容到最后（允许访问的IP地址）
allow=127.0.0.1 10.0.0.0/24
# 禁止root登录
denyusers=root
```

```
/etc/rc.d/init.d/usermin start
```

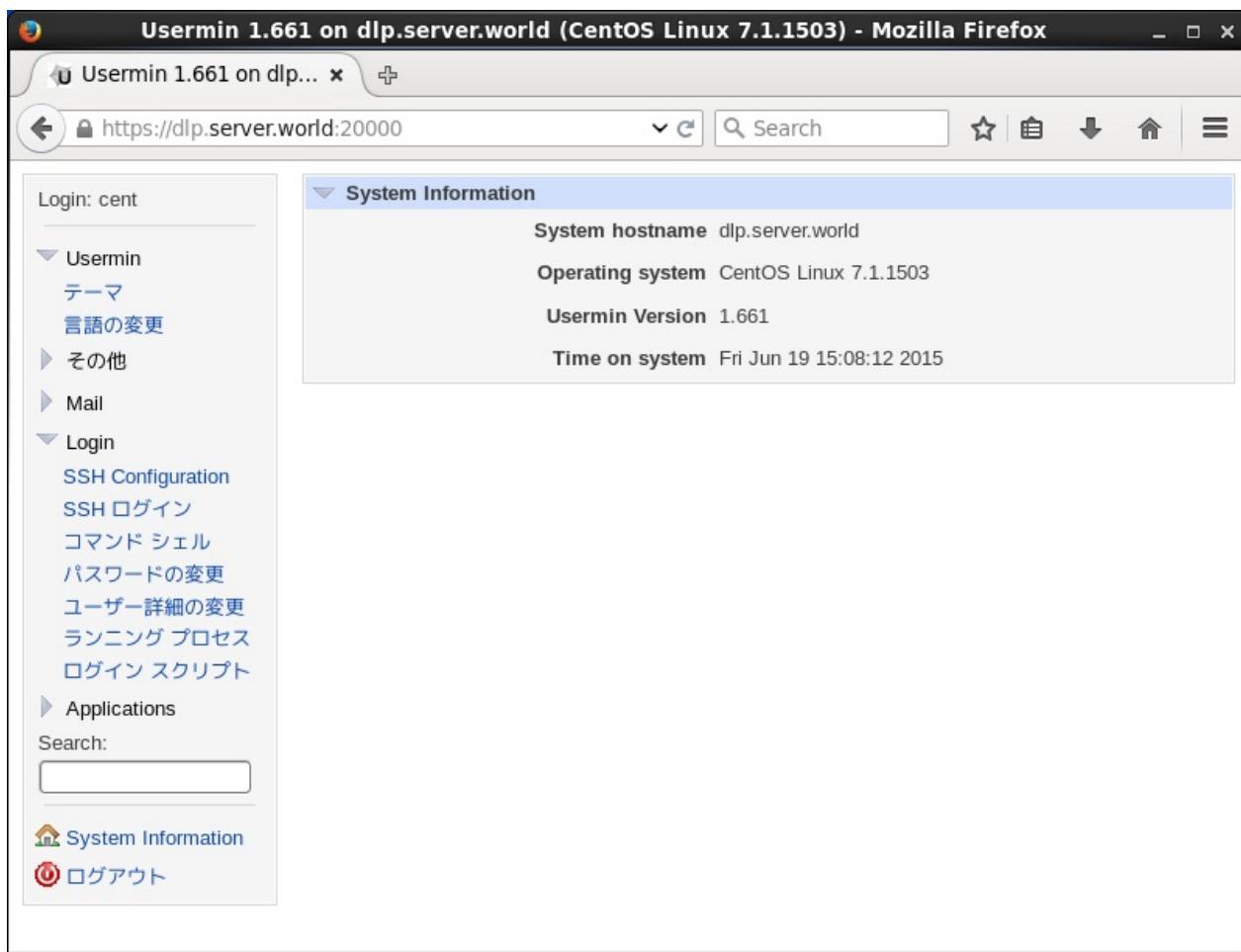
使用客户端Web浏览器访问 `https://(服务器的主机名或IP地址):20000/`，显示Usermin登录界面。使用普通账户登录：



下面是Usermin主页。可以在这里进行操作：



点击左侧菜单上的“Usermin”->“Change Language”，可以更改为自己的语言：



### 附1.7.5. Virtualmin

Virtualmin 用于 Apache httpd 或 Postfix 的虚拟主机的管理和设置。

Virtualmin 是一个 Webmin 的模块，因此先 安装 Webmin。

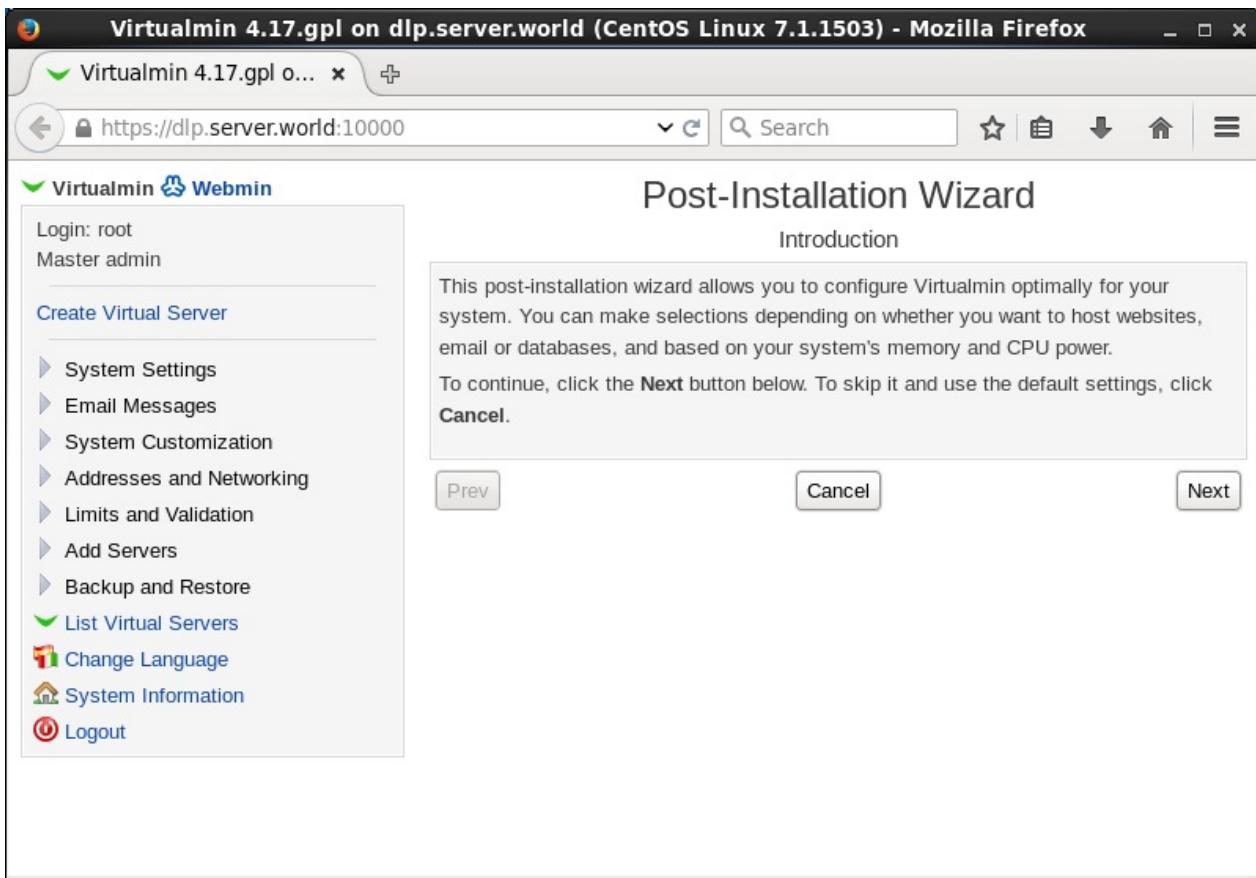
安装 Virtualmin：

```
curl -O http://software.virtualmin.com/gpl/scripts/install.sh
```

```
sh install.sh
```

```
....  
Continue? (y/n) y  
....
```

访问 Webmin 站点 [https://\(服务器的主机名或IP地址\):10000/](https://(服务器的主机名或IP地址):10000/)，然后选择“Cancel”（应用默认设置）或“Next”（手动配置）：



下面是Virtualmin主页。可以在这里进行操作：

The screenshot shows the Virtualmin 4.17.gpl control panel running on a CentOS Linux 7.1.1503 server. The left sidebar contains navigation links for Virtualmin, Webmin, System Settings, Email Messages, System Customization, Addresses and Networking, Limits and Validation, Add Servers, Backup and Restore, List Virtual Servers, Change Language, System Information, and Logout. The main content area displays system information, including the system's hostname (dlp.server.world), operating system (CentOS Linux 7.1.1503), Webmin version (1.750), Virtualmin version (4.17.gpl), and kernel information (Linux 3.10.0-229.4.2.el7.x86\_64). It also shows CPU usage (3% user, 1% kernel, 1% IO, 96% idle), memory usage (Real memory: 646.68 MB used, 3.86 GB total; Virtual memory: 0 bytes used, 3 GB total), local disk space (2.25 GB used, 26.94 GB total), and package updates (1 package update available). The status section indicates new features available in Virtualmin 4.17.gpl.

### 附1.7.6. Spacewalk

Spacewalk是一个开放源码的Linux系统管理解决方案，是Red Hat Satellite的开源版本。

#### 附1.7.6.1. 安装Spacewalk

为Spacewalk配置所需的库（下载前确认最新的RPM）：

```
yum -y install
http://yum.spacewalkproject.org/latest/RHEL/7/x86_64/spacewalk-repo-
2.3-4.el7.noarch.rpm
```

```
cat > /etc/yum.repos.d/jpackage-generic.repo << EOF
[jpackage-generic]
name=JPackage generic
mirrorlist=http://www.jpackage.org/mirrorlist.php?dist=generic&type=free&release=5.0
enabled=1
gpgcheck=1
gpgkey=http://www.jpackage.org/jpackage.asc
EOF
```

可以选择[PostgreSQL](#)和[Oracle 10g](#)或更高版本作为后端数据库。本例选择PostgreSQL：

```
yum --enablerepo=epel -y install spacewalk-setup-postgresql
spacewalk-postgresql perl dojo # 同时开启EPEL
spacewalk-setup --disconnected
```

```
** Database: Setting up database connection for PostgreSQL backend.
** Database: Installing the database:
** Database: This is a long process that is logged in:
** Database: /var/log/rhn/install_db.log
*** Progress: #
** Database: Installation complete.
** Database: Populating database.
*** Progress: #####
* Setting up users and groups.
** GPG: Initializing GPG and importing key.
** GPG: Creating /root/.gnupg directory
You must enter an email address.
# 设置管理员邮件地址
Admin Email Address? root@dlp.srv.world
* Performing initial configuration.
* Activating Spacewalk.
** Loading Spacewalk Certificate.
** Verifying certificate locally.
** Activating Spacewalk.
* Enabling Monitoring.
* Configuring apache SSL virtual host.
# 默认设置回车（启用SSL）
```

```
Should setup configure apache's default ssl server for you (save  
s original ssl.conf) [Y]?  
** /etc/httpd/conf.d/ssl.conf has been backed up to ssl.conf-sws  
ave  
* Configuring tomcat.  
* Configuring jabberd.  
* Creating SSL certificates.  
# 设置CA证书的密码  
CA certificate password?  
Re-enter CA certificate password?  
# 公司  
Organization? ServerWorld  
# 部门  
Organization Unit [dlp.srv.world]?  
# 邮件地址  
Email Address [root@dlp.srv.world]?  
# 城市  
City? CD  
# 省  
State? SC  
# 国家代码  
Country code (Examples: "US", "JP", "IN", or type "?" to see a l  
ist)? CN  
** SSL: Generating CA certificate.  
** SSL: Deploying CA certificate.  
** SSL: Generating server certificate.  
** SSL: Storing SSL certificates.  
* Deploying configuration files.  
* Update configuration in database.  
* Setting up Cobbler..  
Processing /etc/cobbler/modules.conf  
`/etc/cobbler/modules.conf' -> `/etc/cobbler/modules.conf-swsave'  
  
Processing /etc/cobbler/settings  
`/etc/cobbler/settings' -> `/etc/cobbler/settings-swsave'  
# 默认设置回车（启用PXE配置）  
Cobbler requires tftp and xinetd services be turned on for PXE p  
rovisioning functionality. Enable these services [Y]?  
* Restarting services.  
Installation complete.  
Visit https://dlp.srv.world to create the Spacewalk administrato  
r account.
```

使用客户端的Web浏览器访问 `http://(Spacewalk服务器的主机名或IP地址)/`。显示以下页面，设置任意管理员和密码，然后点击“Create Login”：

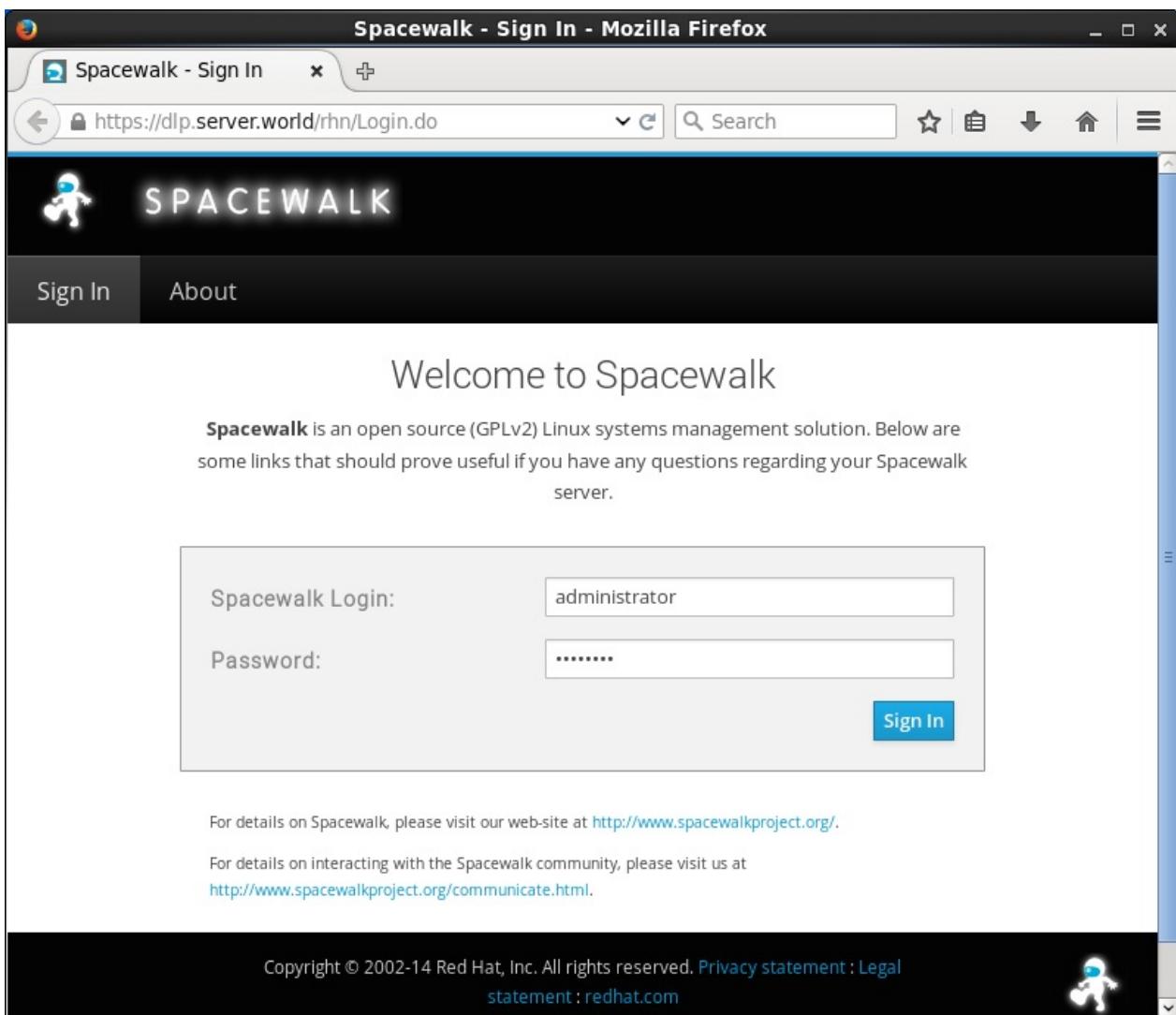
The screenshot shows a Mozilla Firefox browser window with the title "Spacewalk - Create First User - Mozilla Firefox". The address bar displays the URL `https://dlp.server.world/rhn/newlogin/CreateFirstUser.`. The main content area is titled "Create Spacewalk Administrator". It contains two sections: "Login:" and "Account Information:". The "Login:" section includes fields for "Desired Login\*" (set to "administrator"), "Desired Password\*" (a masked password), "Confirm Password\*" (another masked password), and "Password Strength" (a green progress bar). The "Account Information:" section includes fields for "First Name, Last Name\*" (set to "Mr. World Server") and "Email\*" (set to "root@server.world"). A note at the bottom of this section states "\* - Required Field". At the bottom right of the form is a green "Create Login" button.

登录成，下面是Webmin主页：

## 附1.7. 系统管理工具

The screenshot shows the Spacewalk management interface. At the top, there's a navigation bar with links for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. A search bar is also present. On the left, a sidebar titled 'Overview Legend' lists various system status icons: OK (green checkmark), Warning (yellow triangle), Critical (red exclamation mark), Unknown (grey question mark), Locked (padlock), Kickstarting (rocket), Pending Actions (circle with dot), Failed Actions (red circle), Completed Actions (green checkmark), Security (shield), Bug Fix (bug), and Enhancement (square with plus). The main content area starts with a message: 'You have created your first user for the Spacewalk Service. Additional configuration should be finalized by clicking here'. Below this is a 'Tasks' section with links for Entitlements, Activation Keys, Kickstarts, Configuration Files, Organizations, and Spacewalk Configuration. To the right is an 'Inactive Systems' section stating 'No inactive systems.' It notes that all systems are active and provides a link to view all systems. Further down is a 'Most Critical Systems' section with a table header for System, All Updates, Security Errata, Bug Fix Errata, and Enhancement Errata. A note below says 'No critical systems.' and 'None of your systems are in a critical state.' At the bottom, it shows '0 - 0 of 0 most critical systems displayed' and a link to 'View All Critical Systems'. Finally, there's a 'Recently Scheduled Actions' section with the message 'No recently scheduled actions.'

对于下次登录，登录页面显示如下，用上面设置的用户和密码进行身份验证：



### 附1.7.6.2. 初始设置

创建一个渠道和激活密钥。

使用管理员用户登录Spacewalk管理网站，转到“Channels”标签，然后点击“Manage Software Channels”：

The screenshot shows a Mozilla Firefox browser window displaying the Spacewalk software channels management interface. The URL in the address bar is <https://dlp.server.world/rhn/software/channels/All.do>. The page title is "Spacewalk - Channels - Software Channels - All Channels - Mozilla Firefox". The main navigation menu includes "Overview", "Systems", "Errata", "Channels" (which is highlighted with a red box), "Audit", "Configuration", "Schedule", "Users", "Admin", and "Help". A sidebar on the left lists categories like "Software Channels", "All Channels" (highlighted with a red box), "Popular Channels", "My Channels", "Shared Channels", "Retired Channels", "Package Search", "Manage Software Channels" (highlighted with a red box), and "Distribution Channel Mapping". The main content area is titled "Full Software Channel List" and contains tabs for "All Channels" (highlighted with a red box), "Popular Channels", "My Channels", "Shared Channels", and "Retired Channels". A message states, "The software channels listed below are all of the channels that your organization has access to." Below it, a note says, "No channels found." At the bottom of the page, there is a copyright notice: "Copyright © 2002-14 Red Hat, Inc. All rights reserved. Privacy statement : Legal statement : redhat.com".

点击“Create new Channel”：

The screenshot shows the Spacewalk software management interface. The title bar reads "Spacewalk - Channels - Manage Software Channels - Mozilla Firefox". The URL in the address bar is "https://dlp.server.world/rhn/channels/manage/Manage". The main navigation menu includes "Overview", "Systems", "Errata", "Channels" (which is selected), "Audit", "Configuration", "Schedule", "Users", "Admin", and "Help". A sidebar on the left lists "Software Channels", "Package Search", "Manage Software Channels" (which is selected), "Manage Software Packages", "Manage Repositories", and "Distribution Channel Mapping". The main content area is titled "Software Channel Management" and displays a message: "The following software channels are owned by your organization. Modify an existing software channel by selecting it from the list below, or create a new software channel." Below this is a table with two columns: "Channel Name" and "Packages". A red box highlights the "Create Channel" button, which has a plus sign icon and the text "Create Channel". At the bottom of the page, there is a copyright notice: "Copyright © 2002-14 Red Hat, Inc. All rights reserved. Privacy statement : Legal statement : redhat.com".

如下所示，在所需字段上设置项目，向下滚动并点击“Create Channel”：

The screenshot shows the Spacewalk software management interface. The title bar reads "Spacewalk - Channels - Manage Software Channels - Mozilla Firefox". The left sidebar has a "Manage Software Channels" section selected. The main content area is titled "Create Software Channel" and contains "Basic Channel Details". It includes fields for "Channel Name\*" (CentOS 7 Base x86\_64), "Channel Label\*" (centOS7-Base-x86\_64), "Parent Channel" (None), "Architecture" (x86\_64), "Yum Repository Checksum Type" (sha256), and "Channel Summary\*" (CentOS 7 Base for x86\_64). Validation tips are provided for each field, such as "Channel name and label are required." and "They each must be at least 6 characters in length."

创建渠道后，点击“Systems”标签：

## 附1.7. 系统管理工具

The screenshot shows the Spacewalk web interface for managing software channels. The title bar reads "Spacewalk - Channels - Manage Software Channels - Details - Mozilla Firefox". The URL in the address bar is "https://dlp.server.world/rhn/channels/manage/Edit.do?". The top navigation bar includes links for English (change), Knowledgebase, Documentation, administrator, Systems (selected), and a search bar. Below the navigation is a menu bar with tabs: Overview, Systems (highlighted with a red box), Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help.

The left sidebar contains a navigation menu with the following items:

- > Software Channels
- > Package Search
- > Manage Software Channels (highlighted)
- > Manage Software Packages
- > Manage Repositories
- > Distribution Channel Mapping

The main content area displays a message: "Channel CentOS 7 Base x86\_64 created." Below this, the channel is listed as "CentOS 7 Base x86\_64" with a "Delete software channel" link. A tab bar at the top of the content area includes "Details" (highlighted), Managers, Errata, Packages, and Repositories.

The "Basic Channel Details" section contains the following information:

- Create or edit software channels from this page.
- If the parent channel is set to 'none', the channel is a base channel. Otherwise, the channel is a child of the specified channel.
- Channel name and label are required.
- They each must be at least 6 characters in length.
- Channel name must not be longer than 256 characters and channel label must not be longer than 128 characters.
- Channel name must begin with a letter and channel label may begin with a letter or digit.
- They each must not begin with rhn, redhat or red hat.
- They each must contain only lowercase letters, hyphens ('-'), periods ('.'), underscores ('\_'), and numerals.
- Channel name may also contain spaces, parentheses () and forward slashes ('/').
- Channel summary is also required and must not exceed 500 characters.

Form fields for creating the channel:

- Channel Name\*: CentOS 7 Base x86\_64
- Channel Label\*: centos7-base-x86\_64
- Parent Channel: None
- Architecture: x86\_64

Yum Repository Checksum Type: (dropdown menu)

点击左侧菜单上的“Activation Keys”：

## 附1.7. 系统管理工具

Spacewalk - Systems - Overview - Mozilla Firefox

Spacewalk - Systems... x +

https://dlp.server.world/rhn/systems/Overview.do

Search

English (change) Knowledgebase Documentation administrator

Systems Search

Overview Systems Errata Channels Audit Configuration 0 systems selected Manage Clear

Schedule Users Admin Help

System Overview

View System Groups

| System      | Updates | Errata | Packages | Configs | Crashes | Base Channel | Entitlement |
|-------------|---------|--------|----------|---------|---------|--------------|-------------|
| No systems. |         |        |          |         |         |              |             |

Download CSV

点击“Create new key”：

Spacewalk - Systems - Activation Keys - Mozilla Firefox

Spacewalk - Systems... x +

https://dlp.server.world/rhn/activationkeys>List.do

Search

English (change) Knowledgebase Documentation administrator

Systems Search

Overview Systems Errata Channels Audit Configuration 0 systems selected Manage Clear

Schedule Users Admin Help

Activation Keys

+ Create Key

Activation Keys are used to register systems. Systems registered with an activation key will inherit the characteristics defined by that key.

Universal Default

If a universal default activation key is set for your organization, then systems registered to your organization will inherit the properties of that key by default without the need to explicitly specify that key during registration.

You do not currently have a universal default activation key set. To set a key as the universal default, please visit the details page of that key and check off the 'Universal Default?' checkbox.

All Activation Keys

如下所示，在所需字段上设置项目，向下滚动并点击“Create Activation Key”：

The screenshot shows a Mozilla Firefox browser window with the title "Spacewalk - Systems - Activation Keys - Mozilla Firefox". The URL in the address bar is <https://dlp.server.world/rhn/activationkeys/Create.do>. The page content is titled "Create Activation Key" and displays the "Activation Key Details" section. The left sidebar lists various management options like Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. The "Systems" tab is selected. The main form fields include:

- Description:** Activation Key
- Key:** 1- centos7-base-x86\_64
- Usage:** (Leave blank for unlimited use)
- Base Channels:** CentOS 7 Base x86\_64
- Add-On Entitlements:**  Provisioning,  Virtualization,  Virtualization Platform
- Universal Default:** (checkbox)

创建激活密钥后，如下所示（注册客户端时需要它）：

The screenshot shows the Spacewalk web interface for managing activation keys. The title bar reads "Spacewalk - Systems - Activation Keys - Mozilla Firefox". The left sidebar has a "Activation Keys" link under "Advanced Search". The main content area is titled "Activation Keys" and contains a section for "Universal Default" activation keys. It states that if a universal default activation key is set, systems registered to the organization will inherit its properties. A note indicates that no universal default key is currently set. Below this, a table lists all activation keys, showing one entry: "Activation Key" with key "1-centos7-base-x86\_64" and usage "0/(unlimited)".

| Enabled?                            | Description    | Key                   | Usage         |
|-------------------------------------|----------------|-----------------------|---------------|
| <input checked="" type="checkbox"/> | Activation Key | 1-centos7-base-x86_64 | 0/(unlimited) |

### 附1.7.6.3. 客户端设置

注册系统作为Spacewalk客户端，由Spacewalk管理服务器进行管理。

在主机上配置Spacewalk客户端的库，并安装所需的软件包，最后注册到服务器。

```
yum -y install
http://yum.spacewalkproject.org/latest/RHEL/7/x86_64/spacewalk-
client-repo-2.3-4.el7.noarch.rpm

yum --enablerepo=epel -y install rhn-client-tools rhn-check rhn-
setup rhnsd m2crypto yum-rhn-plugin # 同时开启EPEL
```

```
yum -y install http://dlp.srv.world/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm # 从Spacewalk管理服务器下载证书（将主机名替换为自己的）
```

```
rhnreg_ks --serverUrl=https://dlp.srv.world/XMLRPC --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --activationkey=1-centos7-base-x86_64 # 使用上一节生成的激活密钥注册系统
```

登录到Spacewalk管理站点并转到“Systems”标签，然后显示已注册的系统：

The screenshot shows the Spacewalk Systems Overview page. The browser title is "Spacewalk - Systems - Overview - Mozilla Firefox". The URL in the address bar is "https://dlp.server.world/rhn/systems/Overview.do". The main content area is titled "System Overview". On the left, there's a sidebar with navigation links like Overview, Systems, System Groups, System Set Manager, Advanced Search, Activation Keys, Stored Profiles, Custom System Info, Kickstart, and Software Crashes. A "System Legend" section indicates that a green checkmark means "OK" and an orange warning triangle means "Warning". The main table lists one system: "node01.server.world". The table columns include: System (checkbox), Updates (0), Errata (0), Packages (0), Configs (none), Crashes (0), Base Channel (CentOS 7 Base x86\_64), and Entitlement (Management, Provisioning). There are "Select All" and "Download CSV" buttons at the bottom of the table. The top navigation bar includes tabs for Overview, Systems (which is selected), Errata, Channels, Audit, Configuration, and buttons for Manage and Clear.

可以点击主机名确认详细信息：

## 附1.7. 系统管理工具

The screenshot shows the Spacewalk interface for managing systems. The main title bar reads "Spacewalk - Systems - Systems - Details - Overview - Mozilla Firefox". The browser address bar shows the URL "https://dlp.server.world/rhn/systems/details/Overview". The top navigation bar includes links for English (change), Knowledgebase, Documentation, administrator, Systems (selected 0), Search, and various system management tabs like Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help.

The left sidebar is titled "Systems" and lists several categories: All, Physical Systems, Virtual Systems, Out of Date, Requiring Reboot, Extra Packages, Untitled, Ungrouped, Inactive, Recently Registered, Proxy, Duplicate Systems, System Currency, System Groups, System Set Manager, Advanced Search, Activation Keys, Stored Profiles, and Custom System Info. The "Systems" category is currently selected.

The main content area displays details for a system named "node01.server.world". The title bar for this system shows a cloud icon and the name "node01.server.world". Below it are tabs for Details, Software, Configuration, Provisioning, Groups, Audit, and Events, with "Details" being the active tab. Under "Details", there are sub-tabs for Overview, Properties, Remote Command, Reactivation, Hardware, Migrate, and Notes, with "Overview" being the active tab. A "Custom Info" section is also present.

The "System Status" section indicates that the system is "up to date" with a green checkmark icon.

The "System Info" section provides technical details:

|                 |                                                       |
|-----------------|-------------------------------------------------------|
| Hostname:       | node01.server.world                                   |
| IP Address:     | 10.0.0.51                                             |
| IPv6 Address:   | ::1                                                   |
| Virtualization: | KVM/QEMU                                              |
| UUID:           | 31b226d141784b91a2386053c81561fb                      |
| Kernel:         | 3.10.0-229.4.2.el7.x86_64                             |
| Spacewalk       | 1000010000                                            |
| System ID:      |                                                       |
| Activation Key: | 1-centos7-base-x86_64                                 |
| Lock Status:    | System is unlocked<br>( <a href="#">Lock system</a> ) |

The "System Events" section shows the following history:

|              |                                                              |
|--------------|--------------------------------------------------------------|
| Checked In:  | Today at 4:27 PM                                             |
| Registered:  | Today at 4:27 PM                                             |
| Last Booted: | 24 minutes ago<br>( <a href="#">Schedule System Reboot</a> ) |

The "System Properties" section allows editing of various properties:

|                     |                                                                                          |
|---------------------|------------------------------------------------------------------------------------------|
| Entitlements:       | [Management] [Provisioning]                                                              |
| Notifications:      | Daily Summary<br>Errata Email                                                            |
| Auto Errata Update: | No                                                                                       |
| System Name:        | node01.server.world                                                                      |
| Description:        | Initial Registration Parameters:<br>OS: centos-release<br>Release: 7<br>CPU Arch: x86_64 |
| Location:           | (none)                                                                                   |

The "Subscribed Channels" section shows a single entry: "CentOS 7 Base x86\_64".

在这里可以确认已安装的软件包或安装新的软件包或是其他许多管理任务：

## 附1.7. 系统管理工具

The screenshot shows the Spacewalk software management interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, and a search bar. A sidebar on the left lists system categories like All, Physical Systems, Virtual Systems, and Out of Date. The main content area is titled 'node01.server.world' and shows the 'Software' tab selected under 'Packages'. It displays a list of installed packages with columns for Package Name, Architecture, and Installed date. The list includes packages such as acl-2.2.51-12.el7, aic94xx-firmware-30-6.el7, alsa-firmware-1.0.28-2.el7, and avahi-autoipd-0.6.31-14.el7.

| Package Name                     | Architecture | Installed  |
|----------------------------------|--------------|------------|
| acl-2.2.51-12.el7                | x86_64       | 05/24/2015 |
| aic94xx-firmware-30-6.el7        | noarch       | 05/24/2015 |
| alsa-firmware-1.0.28-2.el7       | noarch       | 05/24/2015 |
| alsa-lib-1.0.28-2.el7            | x86_64       | 05/24/2015 |
| alsa-tools-firmware-1.0.27-4.el7 | x86_64       | 05/24/2015 |
| audit-2.4.1-5.el7                | x86_64       | 05/24/2015 |
| audit-libs-2.4.1-5.el7           | x86_64       | 05/24/2015 |
| authconfig-6.2.8-9.el7           | x86_64       | 05/24/2015 |
| autogen-libopts-5.18-5.el7       | x86_64       | 05/24/2015 |
| avahi-autoipd-0.6.31-14.el7      | x86_64       | 05/24/2015 |
| avahi-libt 0.6.31.14.el7         | x86_64       | 05/24/2015 |

## 附1.8. 配置管理工具

### 附1.8.1. Salt

Salt是一个强大的远程执行管理器，用于快速和高效的服务器管理。

#### 附1.8.1.1. 安装Salt

可以在独立的服务器上使用Salt，本例演示配置为服务器和客户端环境。管理服务器称为Salt Master，客户端服务器称为Salt Minion。

在**Salt Master**主机上安装“salt-master”：

```
yum --enablerepo=epel -y install salt-master # 从EPEL安装
```

```
systemctl start salt-master  
systemctl enable salt-master
```

Salt Master防火墙规则：

```
firewall-cmd --add-port={4505/tcp,4506/tcp} --permanent  
firewall-cmd --reload
```

在**Salt Minion**主机上安装“salt-minion”：

```
yum --enablerepo=epel -y install salt-minion # 从EPEL安装
```

编辑 /etc/salt/minion 文件：

```
# 取消注释并指定Salt Master服务器  
master: dlp.srv.world
```

```
systemctl start salt-minion  
systemctl enable salt-minion
```

Salt Minion初始运行时，向Salt Master发送公钥认证。如果Salt Master接受密钥，Salt Master和Salt Minion可以互相连接。

**Salt Master**上运行：

```
salt-key -L # 显示密钥列表
```

```
Accepted Keys:  
Denied Keys:  
Unaccepted Keys:  
node01.srv.world  
Rejected Keys:
```

```
salt-key -A # 使用 A 选项允许所有密钥
```

```
The following keys are going to be accepted:  
Unaccepted Keys:  
node01.srv.world  
Proceed? [n/Y] y  
Key for minion node01.srv.world accepted.
```

```
salt-key -L
```

```
Accepted Keys:  
node01.srv.world  
Denied Keys:  
Unaccepted Keys:  
Rejected Keys:
```

```
salt "*" test.ping # 验证工作
```

```
node01.srv.world:  
True
```

### 附1.8.1.2. 基本用法

从Salt Master到Salt Minion远程执行命令的基本用法：

```
salt [option] [target] [function] [arguments]
```

有关所有嵌入式功能的说明，[参阅官方网站](#)。

也可以使用命令查看功能（有多行输出，使用 `less` 或 `more` 的命令来读取）：

```
salt '*' sys.doc
```

```
'acl.delfacl:'
```

Remove specific FACL from the specified file(s)

CLI Examples:

```
salt '*' acl.delfacl user myuser /tmp/house/kitchen  
salt '*' acl.delfacl default:group mygroup /tmp/house/ki  
tchen
```

.....

.....

可以用各种方式指定目标：

```
salt '*' test.ping # 指定所有Minion， test.ping 表示确认Minion是活动的
```

```
node02.srv.world:
```

True

```
node01.srv.world:
```

True

```
salt 'node01.srv.world' disk.usage # 指定
```

Minion：“node01.srv.world”， disk.usage 表示确认当前磁盘使用率

```
node01.srv.world:  
-----  
/:  
-----  
1K-blocks:  
    27740944  
available:  
    26176776  
capacity:  
    6%  
filesystem:  
    /dev/mapper/centos-root  
used:  
    1564168  
....  
....
```

salt -L 'node01.srv.world,node02.srv.world' status.loadavg # 使用列表指定某些Minion（以逗号分隔），status.loadavg 表示确认平均负载

```
node02.srv.world:  
-----  
1-min:  
    0.0  
15-min:  
    0.05  
5-min:  
    0.01  
node01.srv.world:  
....  
....
```

salt -E 'node[0-9][0-9].srv.world' selinux.getenforce # 用表达式指定Minion，示例表示 node00-99.srv.world ，selinux.getenforce 表示确认 SELinux运行模式

```
node02.srv.world:  
    Enforcing  
node01.srv.world:  
    Enforcing
```

`salt -G 'os:CentOS' grains.item kernelrelease` # 使用Grains数据指定操作系统为CentOS的Minion，`grains.item kernelrelease` 表示确认内核版本来自“grains.item”数据，“Grains”是Salt中用来保存Minion系统数据或其他内容的词

```
node01.srv.world:  
-----  
kernelrelease:  
    3.10.0-327.36.1.el7.x86_64  
node02.srv.world:  
-----  
kernelrelease:  
    3.10.0-327.36.1.el7.x86_64
```

`salt -C 'G@os:CentOS and E@node0[1-5].srv.world' cmd.run 'uptime'`  
# 用 C 选项指定多个条件，示例表示指定操作系统为CentOS且主机名为 node01-node05 的Minion，`cmd.run` 表示执行命令

```
node02.srv.world:  
    09:46:43 up 18 min,  0 users,  load average: 0.00, 0.01, 0.  
03  
node01.srv.world:  
    09:46:43 up 18 min,  0 users,  load average: 0.07, 0.05, 0.  
03
```

还可以使用组指定目标：

编辑 `/etc/salt/master` 文件：

```
# 取消注释  
default_include: master.d/*.conf
```

```
mkdir /etc/salt/master.d
```

编辑 `/etc/salt/master.d/nodegroups.conf` 文件：

```
# group01：用“L@”指定列表（逗号分隔）specify List with "L@" (comma separated)
# group02：用表达式指定node03-node05
# group03：指定操作系统为CentOS

nodegroups:
    group01: 'L@node01.srv.world,node02.srv.world'
    group02: 'E@node0[3-5].srv.world'
    group03: 'G@os:CentOS'
```

```
systemctl restart salt-master
```

```
salt -N 'group01' firewalld.list_services # 对“group01”组运行，firewalld.list_services 表示确认firewalld允许的服务
```

```
node01.srv.world:
    - dhcpcv6-client
    - ssh
node02.srv.world:
    - dhcpcv6-client
    - ssh
```

### 附1.8.1.3. 使用Salt State文件

Salt State文件是用YAML文件编写的配置文件。

首先定义存放Salt State文件的根目录。默认位置是 `/srv/salt` （本例演示使用默认位置配置）：

编辑 `/etc/salt/master` 文件：

```
# 取消注释并定义根目录
file_roots:
    base:
        - /srv/salt
```

```
mkdir /srv/salt
```

将State文件放在根目录下，可以使用 salt 命令将配置应用于Minion（例如，将“wget”软件包安装到Minion）：

编辑 /srv/salt/default.sls 文件（扩展名为“sls”，文件可为任意名称）：

```
install_wget:  
  pkg.installed:  
    - name: wget
```

```
salt "node01.srv.world" state.sls default # 适用于"node01"
```

```
node01.srv.world:  
-----  
          ID: install_wget  
        Function: pkg.installed  
            Name: wget  
        Result: True  
      Comment: The following packages were installed/updated: wget  
  
.....  
.....  
  
Summary  
-----  
Succeeded: 1 (changed=1)  
Failed:    0  
-----  
Total states run:      1
```

```
salt "node01.srv.world" cmd.run 'rpm -q wget' # 验证
```

```
node01.srv.world:  
  wget-1.14-10.el7_0.1.x86_64
```

将称为“Top文件”的“top.sls”在定义的根目录下：

编辑 /srv/salt/top.sls 文件：

```
base:  
  # 定义目标Minion  
  '*' :  
    # 定义State文件的名称  
    - default
```

编辑 `/srv/salt/default.sls` 文件，创建在Top文件中定义的State文件：

```
# 例如，安装并启动httpd和MariaDB并安装PHP  
webserver:  
  pkg.installed:  
    - pkgs:  
      - httpd  
      - php  
      - php-mbstring  
      - php-pear  
      - mariadb-server  
  
  /var/www/html/index.php:  
    file:  
      - managed  
      - source: salt://httpd/index.php  
      - require:  
        - pkg: webserver  
  
  # 初始设置脚本  
  /tmp/setup.sql:  
    file:  
      - managed  
      - source: salt://httpd/setup.sql  
  
enable_httpd:  
  service.running:  
    - name: httpd  
    - enable: True  
    - require:  
      - pkg: webserver  
  
enable_mariadb:  
  service.running:
```

```
- name: mariadb
- enable: True
- require:
  - pkg: webserver

setup_mariadb:
cmd.run:
- name: '/bin/mysql -u root < /tmp/setup.sql'
- require:
  - service: enable_mariadb

# 如果Firewalld运行，需配置服务
{% set fw_status = salt['service.status']('firewalld') %}
{% if fw_status %}
setup_fw:
cmd.run:
- names:
  - '/bin/firewall-cmd --add-service={http,https,mysql}'
  - '/bin/firewall-cmd --add-service={http,https,mysql} --pe
rmanent'
{% endif %}
```

编辑 `/srv/salt/httpd/index.php` 文件，创建index.php模板：

```
<?php
print "Salt State Test Page";
?>
```

编辑 `/srv/salt/httpd/setup.sql` 文件，创建MariaDB初始设置脚本：

```
set password for root@localhost=password('password');
set password for root@'127.0.0.1'=password('password');
delete from mysql.user where user='';
delete from mysql.user where password='';
drop database test;
```

`salt "*" state.apply # 运行“state.apply”来应用设置`

```
node01.srv.world:
-----
cmd_|_setup_fw_|-/bin/firewall-cmd --add-service={http,https
,mysql} --permanent_|-run:
-----
__run_num__:
    7
changes:

.....
.....
name:
    mariadb
result:
    True
start_time:
    19:56:52.911517
```

### 验证

```
salt "node01.srv.world" cmd.run 'systemctl status httpd'
```

```
node01.srv.world:
  * httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; en
abled; vendor preset: disabled)
    Active: active (running) since Mon 2016-10-17 10:56:52 JS
T; 2min 16s ago
    .....
    .....
```

```
salt "node02.srv.world" cmd.run 'systemctl status httpd'
```

```
salt "node01.srv.world" cmd.run 'systemctl status httpd'
```

```
node01.srv.world:  
  * mariadb.service - MariaDB database server  
    Loaded: loaded (/usr/lib/systemd/system/mariadb.service;  
            enabled; vendor preset: disabled)  
      Active: active (running) since Mon 2016-10-17 10:56:57 JS  
          T; 2min 24s ago  
    ....  
    ....
```

```
salt "node02.srv.world" cmd.run 'systemctl status httpd'  
  
salt "node01.srv.world" cmd.run 'mysql -u root -ppassword -e "show  
databases;"'
```

```
node01.srv.world:  
  Database  
  information_schema  
  mysql  
  performance_schema
```

```
salt "node02.srv.world" cmd.run 'mysql -u root -ppassword -e "show  
databases;"'
```

```
curl http://node01.srv.world/index.php
```

Salt State Test Page

```
curl http://node02.srv.world/index.php
```

### 附1.8.1.4. 使用 Salt-cp

可以使用 `salt` 命令将文件复制到Minion，但是使用 `salt-cp` 命令来处理Minion更简单：

```
salt-cp '*' test.txt /root/test.txt # 将当前目录下的 test.txt 复制到  
所有Minion中的 /root/ 中
```

```
{'node01.srv.world': {'/root/test.txt': True},  
'node02.srv.world': {'/root/test.txt': True}}
```

```
salt-cp 'node01.srv.world' setup.sql /tmp/setup.sql # 指定 node01.srv.world 为 目标Minion 并 复制
```

```
{'node01.srv.world': {'/root/test.txt': True},  
'node02.srv.world': {'/root/test.txt': True}}
```

```
salt-cp -E 'node[0-9][1-2].srv.world' setup.sql /tmp/setup.sql # 用表达式 指定 目标Minion
```

```
{'node01.srv.world': {'/tmp/setup.sql': True},  
'node02.srv.world': {'/tmp/setup.sql': True}}  
.....  
.....
```

```
salt-cp -G 'os:CentOS' config.sh /tmp/config.sh # 使用Grains数据指定 Minion
```

```
{'node01.srv.world': {'/tmp/config.sh': True},  
'node02.srv.world': {'/tmp/config.sh': True}}  
.....  
.....
```

### 附1.8.2. Puppet

Puppet是一个配置管理工具。

#### 附1.8.2.1. 安装Puppet

可以在独立的服务器上使用Puppet，本例演示配置为Puppet服务器和Puppet客户端环境。

必须先设置DNS或hosts来解析名称或IP地址，以及NTP设置。

在Puppet服务器主机上安装 puppet-server：

```
yum -y install https://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
```

```
sed -i -e "s(enabled=1)enabled=0/g"  
/etc/yum.repos.d/puppetlabs.repo
```

```
yum --enablerepo=puppetlabs-products,puppetlabs-deps -y install  
puppet-server
```

编辑 /etc/puppet/puppet.conf 文件：

```
[main]  
# 在[main]部分中添加以下内容：Puppet服务器的DNS名称  
dns_alt_names = dlp.srv.world,dlp
```

```
puppet master --verbose --no-daemonize
```

```
Info: Creating a new SSL key for ca  
Info: Creating a new SSL certificate request for ca  
Info: Certificate Request fingerprint (SHA256):  
Notice: Signed certificate request for ca  
Info: Creating a new certificate revocation list  
Info: Creating a new SSL key for dlp.srv.world  
Info: csr_attributes file loading from /etc/puppet/csr_attributes.yaml  
Info: Creating a new SSL certificate request for dlp.srv.world  
Info: Certificate Request fingerprint (SHA256):  
Notice: dlp.srv.world has a waiting certificate request  
Notice: Signed certificate request for dlp.srv.world  
Notice: Removing file Puppet::SSL::CertificateRequest dlp.srv.world at  
' /var/lib/puppet/ssl/ca/requests/dlp.srv.world.pem'  
Notice: Removing file Puppet::SSL::CertificateRequest dlp.srv.world at  
' /var/lib/puppet/ssl/certificate_requests/dlp.srv.world.pem'  
Notice: Starting Puppet master version 3.8.1  
# 按Ctrl + C退出
```

```
systemctl start puppetmaster  
systemctl enable puppetmaster
```

在**Puppet**客户端主机安装 `puppet` :

```
yum -y install https://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
```

```
sed -i -e "s(enabled=1)enabled=0/g"  
/etc/yum.repos.d/puppetlabs.repo
```

```
yum --enablerepo=puppetlabs-products,puppetlabs-deps -y install  
puppet
```

编辑 `/etc/puppet/puppet.conf` 文件 :

### [agent]

```
# 在[agent]部分中添加以下内容：Puppet服务器的主机名或IP地址  
server = dlp.srv.world
```

```
puppet agent --test --ca_server=dlp.srv.world
```

```
Info: Creating a new SSL key for node01.srv.world  
Info: Caching certificate for ca  
Info: csr_attributes file loading from /etc/puppet/csr_attribute  
s.yaml  
Info: Creating a new SSL certificate request for node01.srv.wor  
ld  
Info: Certificate Request fingerprint (SHA256):  
Info: Caching certificate for ca  
Exiting; no certificate found and waitforcert is disabled
```

```
systemctl start puppet  
systemctl enable puppet
```

在**Puppet**服务器上启用Puppet客户端的证书 :

```
puppet cert list # 显示证书请求
```

```
"node01.srv.world" (SHA256) xx:xx:xx:xx:xx:xx:xx
```

```
puppet cert --allow-dns-alt-names sign node01.srv.world # 签名
```

```
Notice: Signed certificate request for node01.srv.world
Notice: Removing file Puppet::SSL::CertificateRequest node01.srv
.world at
  '/var/lib/puppet/ssl/ca/requests/node01.srv.world.pem'
```

创建测试清单（manifest）以确认Puppet服务器/客户端正常工作。Puppet客户端默认情况下，每隔30分钟可以使用Puppet服务器上的清单，所以等一会儿来确认它，或者如果想立即确认，重新启动Puppet客户端守护进程（puppetd）：

编辑 /etc/puppet/manifests/site.pp 文件：

```
# 例如，如下所示创建一个“testgroup”
group { 'testgroup':
  ensure => present,
  gid   => 2000,
}
```

```
systemctl restart puppet # 如果想立即确认重启puppetd
```

```
grep testgroup /etc/group
```

```
testgroup:x:2000:
```

可以如下所示手动将清单应用到本地环境：

```
puppet apply /etc/puppet/manifests/site.pp
```

```
Notice: Compiled catalog for dlp.srv.world in environment produc
tion in 0.13 seconds
Notice: /Stage[main]/Main/Group[testgroup]/ensure: created
Notice: Finished catalog run in 0.34 seconds
```

### 附1.8.2.2. 文件资源

这是文件资源（file resource）的示例。

它如下管理配置以保存文件（如果文件不在Puppet客户端，则会创建该文件，如果存在，它将保留指定的属性）：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
file { '/home/testfile.txt':
  ensure  => file,
  owner   => 'root',
  group   => 'root',
  mode    => 644,
  content => 'This is the puppet test file.',
}
```

编辑 `/etc/puppet/manifests/site.pp` 文件，使用变量指定内容：

```
$contents = 'This is the test Puppet manifest.
Sample contents
Test contents

file { '/home/testfile.txt':
  ensure  => file,
  owner   => 'root',
  group   => 'root',
  mode    => 644,
  content => "$contents",
}
```

将Puppet服务器上的源文件指定为模板：

编辑 `/etc/puppet/fileserver.conf` 文件：

```
# add to the end: specify the directory which includes template
files
[extra_files]
  path /etc/puppet/files
  allow *
# extra_files -> 任意名称
# path          -> 目录路径
# allow         -> 允许访问的客户端（上例表示允许所有），如果要指定客户端，
格式为：“allow 192.168.0.0/24”或“*.srv.world”
```

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
file { '/home/testfile.txt':
  ensure => file,
  owner  => 'root',
  group  => 'root',
  mode    => 644,
  source  => 'puppet://dlp.srv.world/extra_files/test.txt',
}
```

```
mkdir /etc/puppet/files
```

```
echo "Puppet test file" > /etc/puppet/files/test.txt
```

管理配置以保持链接。下例保持到“/home/testfile.txt”的链接“/home/testfile.link”：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
file { '/home/testfile.link':
  ensure => link,
  target => '/home/testfile.txt',
}
```

管理配置以保持文件不存在，如果存在，则将被删除：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
file { '/home/testfile.link': ensure => absent }
```

管理配置以保持递归目录。下例指定“mode”为“644”，但为目录添加“x”。此外，源目录中不存在的文件或目录将使用参数 `purge` 或 `force` 进行删除。

编辑 `/etc/puppet/fileserver.conf` 文件：

```
# 添加到最后：将目录指定为源
[extra_dir]
  path /etc/puppet/dirs
  allow *
```

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
file { '/home/testdir':
  ensure  => directory,
  recurse => true,
  purge   => true,
  force    => true,
  owner    => 'root',
  group   => 'root',
  mode     => 644,
  source   => 'puppet://dlp.srv.world/extra_dir/testdir',
}
```

```
mkdir -p /etc/puppet/dirs/testdir
```

### 附1.8.2.3. 软件包资源

这是软件包资源（package resource）的示例。

管理配置以保持“`httpd`”被安装：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
package { 'httpd':
  provider => yum,
  ensure   => installed,
}
```

管理配置以保持“`latest httpd`”被安装：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
package { 'httpd':
  provider => yum,
  ensure   => latest,
}
```

管理配置以保持“`epel-release`”使用rpm安装：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
package { 'epel-release':
  provider => rpm,
  ensure    => installed,
  source    => 'http://dl.fedoraproject.org/pub/epel/6/x86_64/e
pel-release-6-8.noarch.rpm',
}
```

管理配置以保持“`httpd`”不被安装（如果安装，将被删除）：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
package { 'httpd':
  provider => yum,
  ensure    => purged,
}
```

### 附1.8.2.4. 服务资源

这是服务资源（service resource）的示例。

管理配置以保持“`httpd`”在运行：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
service { 'httpd':
  name    => 'httpd',
  ensure  => running,
}
```

管理配置以保持“`httpd`”在运行。如果没有安装，“`httpd`”当然不会启动，因此管理配置以保持“`httpd`”被安装作为“`require`”（必要）参数如下：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
package { 'httpd':
  provider => yum,
  ensure   => installed,
}

service { 'httpd':
  name      => 'httpd',
  ensure    => running,
  require  => Package['httpd'],
}
```

管理配置以保持“httpd”不运行（如果运行，将被停止）：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
service { 'httpd':
  name      => 'httpd',
  ensure    => stopped,
}
```

下面配置当文件 `/etc/httpd/conf/httpd.conf` 更新时，会重新启动“httpd”：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
file { '/etc/httpd/conf/httpd.conf':
  ensure => file,
  owner  => 'root',
  group  => 'root',
  mode   => 644,
  source => 'puppet://dlp.srv.world/extr_files/httpd.conf',
  notify => Service['httpd'],
}

service { 'httpd':
  name      => 'httpd',
  ensure    => running,
}
```

### 附1.8.2.5. 组资源

这是组资源（group resource）的示例。

管理配置以保持组“centos”存在：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
group { 'centos': ensure => present }
```

明确指定GID：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
group { 'centos':
  ensure => present,
  gid    => 1000,
}
```

管理配置以保持组“centos”不存在（如果存在，将被删除）：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
group { 'centos': ensure => absent }
```

### 附1.8.2.6. 用户资源

这是用户资源（user resource）的示例。

管理配置以保持用户“cent”存在：

为用户生成一个加密密码：

```
python -c 'import crypt,getpass; \
print(crypt.crypt(getpass.getpass(), \
crypt.mksalt(crypt.METHOD_SHA512)))'
```

```
Password:  
$6$Fb2fpmp8Vctsxxxxxxxxx
```

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
group { 'centos': ensure => absent }
```

明确指定UID或GID或组：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
group { 'cent':
  ensure => present,
  gid    => 1000,
}

user { 'cent':
  ensure      => present,
  home       => '/home/cent',
  managehome => true,
  uid        => 1000,
  gid        => 1000,
  groups     => ['cent', 'wheel'],
  password   => '$6$0XTc2rj1xxxxxxxx',
}
```

明确指定密码最大，最小位数或注释：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
group { 'cent':
  ensure => present,
  gid   => 1000,
}

user { 'cent':
  ensure          => present,
  home           => '/home/cent',
  managehome     => true,
  uid            => 1000,
  gid            => 1000,
  groups         => ['cent', 'wheel'],
  password_max_age => 90,
  password_min_age => 1,
  password       => '$6$0XTc2rjlxxxxxxxxx',
  comment        => 'Cent User',
}
```

管理配置以保持用户“cent”不存在（如果存在，将被与主目录一起删除）：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
user { 'cent':
  ensure      => absent,
  home        => '/home/cent',
  managehome  => true,
}
```

### 附1.8.2.7. 执行资源

这是执行资源（exec resource）的示例。

可以通过使用执行资源来执行任何命令，但不推荐使用，因为这样是危险的。所以使用这个资源来处理特定的情况，比如通过使用 `refreshonly` 参数来接收事件。

当 `/etc/aliases` 更新时，它执行 `newaliases`：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
file { '/etc/aliases':
  ensure => file,
  owner  => 'root',
  group  => 'root',
  mode    => 644,
  source  => 'puppet://dlp.srv.world/extra_files/aliases'
}

exec { 'newaliases':
  path      => ['/usr/bin', '/usr/sbin'],
  subscribe => File['/etc/aliases'],
  refreshonly => true
}
```

### 附1.8.2.8. 节点部分

可以通过使用“node”（节点）部分为每个客户端设置资源，如下所示：

将资源设置到 `www01.srv.world` 及其他（“`default`”部分适用于其他客户端）：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
node 'www01.srv.world' {
  file { '/home/testfile.txt':
    ensure => file,
    owner  => 'root',
    group  => 'root',
    mode    => 644,
    content => 'This is the puppet test file.',
  }
}

node default {
  user { 'cent':
    ensure      => present,
    home       => '/home/cent',
    managehome => true,
    password   => '$6$0XTc2rjlxxxxxxxxx',
  }
}
```

将资源设置到 `www01.srv.world` 和 `www.srv.world` 及其他：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
node 'www01.srv.world' {
    file { '/home/testfile.txt':
        ensure  => file,
        owner   => 'root',
        group   => 'root',
        mode    => 644,
        content => 'This is the puppet test file.',
    }
}

node 'www.srv.world' inherits 'www01.srv.world' {
    file { '/home/testfile2.txt':
        ensure  => file,
        content => 'inherits test file.',
    }
}

node default {
    user { 'cent':
        ensure      => present,
        home       => '/home/cent',
        managehome => true,
        password   => '$6$0XTc2rjlxxxxxxxx',
    }
}
```

### 附1.8.2.9. 类部分

可以通过使用“`class`”（类）部分来管理一些资源，如下所示：

这是定义和使用类“`sample01`”的例子：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
class sample01 {
    file { '/home/testfile.txt':
        ensure  => file,
        owner   => 'root',
        group   => 'root',
        mode    => 644,
        content => 'This is the puppet test file.',
    }
    user { 'cent':
        ensure      => present,
        home       => '/home/cent',
        managehome => true,
        password   => '$6$0XTc2rjlxxxxxxxx',
    }
}
node 'www.srv.world' { include 'sample01' }
```

这是一个使用继承类的例子，类“sample01”和类“sample02”都通过此清单应用于 `www.srv.world`：

编辑 `/etc/puppet/manifests/site.pp` 文件：

```
class sample01 {
    file { '/home/testfile.txt':
        ensure  => file,
        owner   => 'root',
        group   => 'root',
        mode    => 644,
        content => 'This is the puppet test file.',
    }
    user { 'cent':
        ensure      => present,
        home       => '/home/cent',
        managehome => true,
        password   => '$6$0XTc2rjlxxxxxxxx',
    }
}
class sample02 inherits sample01 {
    file { '/home/testfile2.txt':
        ensure  => file,
        owner   => 'root',
        group   => 'wheel',
        content => 'test file2',
    }
}
node 'www.srv.world' { include 'sample02' }
```

### 附1.8.2.10. factor变量

如果需要，可以使用由系统自动设置的factor变量。

显示factor变量：

```
factor
```

```
architecture => x86_64
augeasversion => 1.1.0
bios_release_date => 01/01/2007
bios_vendor => Seabios
bios_version => 0.5.1
blockdevice_vda_size => 53687091200
blockdevice_vda_vendor => 0x1af4
...
...
...
uptime_hours => 1
uptime_seconds => 6836
uuid => 8DBDD6BD-B474-765D-D743-1160BE341044
virtual => kvm
```

例如，如果操作系统为“RedHat”或“CentOS”，版本为“7.1.1503”，则应用类“sample01”，如果版本不是“7.1.1503”，则应用类“sample02”，如果操作系统不是“RedHat”或“CentOS”，应用类“sample03”：

编辑 /etc/puppet/manifests/site.pp 文件：

```
class sample01 {
    file { '/home/testfile.txt':
        ensure  => file,
        owner   => 'root',
        group   => 'root',
        mode    => 644,
        content => 'This is the puppet test file.',
    }
}

class sample02 {
    user { 'cent':
        ensure      => present,
        home       => '/home/cent',
        managehome => true,
        password   => '$6$0XTc2rjlxxxxxxxx',
    }
}

class sample03 {
    file { '/home/testfile2.txt':
        ensure  => file,
        owner   => 'root',
        group   => 'wheel',
        content => 'test file2',
    }
}

case $operatingsystem {
    'RedHat', 'CentOS':
        if $operatingsystemrelease == '7.1.1503' { include 'sample01' }
        else                                     { include 'sample02' }
    }
    default:                                { include 'sample03' }
}
```

### 附1.8.3. Ansible

Ansible提供一种最简单的方式用于发布、管理和编排计算机系统的工具。

#### 附1.8.3.1. 安装Ansible

Ansible不需要专用的服务器/客户端程序，只需要Ansible命令和SSH。

在想要使用Ansible管理的所有客户端上启动SSH守护进程。

安装Ansible：

```
yum --enablerepo=epel -y install ansible openssh-clients # 从EPEL  
安装
```

为Ansible设置客户端的基本设置（原始文件 /etc/ansible/hosts 中有更多详细信息，可以参考）：

编辑 /etc/ansible/ansible.cfg 文件：

```
# 取消注释（不检查主机密钥）  
host_key_checking = False
```

```
mv /etc/ansible/hosts /etc/ansible/hosts.org
```

编辑 /etc/ansible/hosts 文件：

```
# 写入管理的客户端  
10.0.0.50  
  
# 可以分组（定义任意组名称）  
[target_servers]  
# 写入客户端进行分组  
10.0.0.51  
10.0.0.52
```

```
ansible all --list-hosts # 确认设置，显示所有定义的主机
```

```
10.0.0.50  
10.0.0.51  
10.0.0.52
```

```
ansible target_servers --list-hosts # 显示组中的特定主机
```

```
10.0.0.51
10.0.0.52
```

### 附1.8.3.2. 基本用法

Ansible的基本用法：

```
ansible [Target Hosts] [Option] -m [Module] -a [Arguments]
```

Ansible官方网站提供了许多模块（Module），可以访问[网站参考](#)。

使用Ansible时进行身份验证是有必要的，因为它使用SSH访问。而且，可以将作为非特权用户Ansible使用，但是如果想在客户端上使用特权，则需要允许通过 `sudo` 等使用特权命令。

对于客户端上的SSH服务器允许直接root登录（“`PermitRootLogin no`”除外）+ 密钥对认证（`non-passphrase`非密码）的情况，可以如下使用Ansible。如果在密钥对中设置密码，可以在[启动SSH-Agent](#)后使用：

```
ansible target_servers -m ping # 执行 ping 命令到组“target_servers”
```

```
10.0.0.51 | success >> {
    "changed": false,
    "ping": "pong"
}

10.0.0.52 | success >> {
    "changed": false,
    "ping": "pong"
}
```

如果想连接密码认证，可以使用如下所示的 `k` 选项。但是，它需要在所有客户端上设置相同的密码，并且需要[安装SSHPass](#)：

```
ansible target_servers -k -m command -a "uptime" # 执行 uptime 命令
到组“target_servers”
```

```
SSH password:  
10.0.0.52 | success | rc=0 >>  
 10:28:07 up 10 min, 2 users, load average: 0.01, 0.02, 0.03  
  
10.0.0.51 | success | rc=0 >>  
 10:28:07 up 11 min, 1 user, load average: 0.01, 0.03, 0.05
```

对于使用非特权用户连接到客户端的情况，可以使用 `sudo` 提权。如果想使用除 `root` 以外的其他用户权限，指定选项 `--become-user=xxx`。如果想使用 `sudo` (`su` | `pbrun` | `pfexec` | `runas`) 外另一种方式提权，指定选项 `--become-method=xxx`：

```
ansible target_servers -k -m command -a "cat /etc/shadow" -b --ask-become-pass # 执行 cat /etc/shadow 命令到组“target_servers”
```

```
SSH password:  
SUDO password[defaults to SSH password]:  
10.0.0.51 | success | rc=0 >>  
root:$6$xxxxxxxxxx:15441:0:99999:7:::  
bin:*:15240:0:99999:7:::  
daemon:*:15240:0:99999:7:::  
.....  
.....  
  
10.0.0.52 | success | rc=0 >>  
root:$6$xxxxxxxxxx:15441:0:99999:7:::  
bin:*:15240:0:99999:7:::  
daemon:*:15240:0:99999:7:::  
.....  
.....
```

### 附1.8.3.3. 使用 Playbook1

这是 Ansible Playbook 的基本用法。

Playbook 写为 YAML 文件。

例如，创建一个具有相同权限的文件的 Playbook：

编辑 `playbook_sample.yml` 文件：

```
# 目标主机名或组名
- hosts: target_servers
# 定义任务
  tasks:
# 任意任务名称
  - name: Test Task
# 使用文件模块设置文件状态
  file: path=/home/cent/test.conf state=touch owner=cent group=cent mode=0600
```

```
ansible-playbook playbook_sample.yml # 运行Playbook
```

```
PLAY [target_servers] ****
*****
GATHERING FACTS ****
*****
ok: [10.0.0.52]
ok: [10.0.0.51]

TASK: [Test Task] ****
*****
changed: [10.0.0.51]
changed: [10.0.0.52]

PLAY RECAP ****
*****
10.0.0.51 : ok=2     changed=1     unreachable=0
              failed=0
10.0.0.52 : ok=2     changed=1     unreachable=0
              failed=0
```

```
ansible target_servers -m command -a "ls -l /home/cent" # 确认设置
```

```
10.0.0.51 | success | rc=0 >>
total 0
-rw----- 1 cent cent 0 Apr 21 15:35 test.conf

10.0.0.52 | success | rc=0 >>
total 0
-rw----- 1 cent cent 0 Apr 21 15:35 test.conf
```

例如，创建一个安装并运行Apache httpd的Playbook：

编辑 `playbook_sample.yml` 文件：

```
- hosts: target_servers
# 使用特权（默认：root）
  become: yes
# 使用特权的方式
  become_method: sudo
# 定义任务
  tasks:
    - name: httpd is installed
      yum: name=httpd state=installed
    - name: httpd is running and enabled
      service: name=httpd state=started enabled=yes
```

```
ansible-playbook playbook_sample.yml --ask-become-pass # 运行
Playbook
```

SUDO password:

```
PLAY [target_servers] *****
*****
GATHERING FACTS *****
*****
ok: [10.0.0.51]
ok: [10.0.0.52]

TASK: [httpd is installed] *****
*****
changed: [10.0.0.51]
changed: [10.0.0.52]

TASK: [httpd is running and enabled] *****
*****
changed: [10.0.0.51]
changed: [10.0.0.52]

PLAY RECAP *****
*****
10.0.0.51 : ok=3     changed=2     unreachable=0
              failed=0
10.0.0.52 : ok=3     changed=2     unreachable=0
              failed=0
```

```
ansible target_servers -m shell -a "/bin/systemctl status httpd | head -3" -b --ask-become-pass #确认设置
```

```
SUDO password:  
10.0.0.52 | success | rc=0 >>  
httpd.service - The Apache HTTP Server  
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)  
    Active: active (running) since Thu 2015-07-15 20:42:00 JST; 2min 2s ago  
  
10.0.0.51 | success | rc=0 >>  
httpd.service - The Apache HTTP Server  
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)  
    Active: active (running) since Thu 2015-07-15 20:42:00 JST; 2min 2s ago
```

### 附1.8.3.4. 使用 Playbook2

可以在Ansible Playbook中使用变量。

下面为使用变量的示例。

编辑 `playbook_sample.yml` 文件：

```
- hosts: target_servers  
  become: yes  
  become_method: sudo  
  tasks:  
    - name: General packages are installed  
      yum: name={{ item }} state=installed  
      with_items:  
        - vim-enhanced  
        - wget  
        - unzip  
    tags: General_Packages
```

```
ansible-playbook playbook_sample.yml --ask-become-pass
```

SUDO password:

```
PLAY [target_servers] *****
*****
GATHERING FACTS *****
*****
ok: [10.0.0.52]
ok: [10.0.0.51]

TASK: [General packages are installed] *****
*****
changed: [10.0.0.52] => (item=vim-enhanced,wget,unzip)
changed: [10.0.0.51] => (item=vim-enhanced,wget,unzip)

PLAY RECAP *****
*****
10.0.0.51 : ok=2     changed=1     unreachable=0
              failed=0
10.0.0.52 : ok=2     changed=1     unreachable=0
              failed=0
```

```
ansible target_servers -m shell -a "rpm -qa | grep -E 'vim-
enhanced|wget|unzip'" --ask-become-pass # 确认设置
```

SUDO password:

```
10.0.0.51 | success | rc=0 >>
vim-enhanced-7.4.160-1.el7.x86_64
wget-1.14-10.el7_0.1.x86_64
unzip-6.0-15.el7.x86_64

10.0.0.52 | success | rc=0 >>
vim-enhanced-7.4.160-1.el7.x86_64
wget-1.14-10.el7_0.1.x86_64
unzip-6.0-15.el7.x86_64
```

运行Playbook时，始终执行“GATHERING FACTS”任务，它是Ansible获取目标主机信息并将其设置为变量的函数，可以在Playbooks中参考并使用它们。如果想确认设置了哪些变量，可以如下使用“setup”模块输出：

```
ansible 10.0.0.51 -m setup # 用“setup”模块显示“GATHERING FACTS”的内容
```

```
10.0.0.51 | success >> {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "10.0.0.51"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::216:36ff:fe29:7f1c"
        ],
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "01/01/2007",
        "ansible_bios_version": "0.5.1",
        .....
        .....
    }
}
```

编辑 `playbook_sample.yml` 文件：

```
# 参考“ansible_distribution”，“ansible_distribution_version”
- hosts: target_servers
  tasks:
    - name: Refer to Gathering Facts
      command: echo "{{ ansible_distribution }} {{ ansible_distribution_version }}"
      register: dist
    - debug: msg="{{ dist.stdout }}
```

```
ansible-playbook playbook_sample.yml
```

```
PLAY [target_servers] ****
*****
GATHERING FACTS ****
*****
ok: [10.0.0.51]
ok: [10.0.0.52]

TASK: [Refer to Gathering Facts] ****
*****
changed: [10.0.0.51]
changed: [10.0.0.52]

TASK: [debug msg="{{ dist.stdout }}"] ****
*****
ok: [10.0.0.51] => {
    "msg": "CentOS 7.1.1503"
}
ok: [10.0.0.52] => {
    "msg": "CentOS 7.1.1503"
}

PLAY RECAP ****
*****
10.0.0.51 : ok=3     changed=1     unreachable=0
            failed=0
10.0.0.52 : ok=3     changed=1     unreachable=0
            failed=0
```

### 附1.8.3.5. 使用 Playbook3

要使用“when”，“failed\_when”，可以在Ansible Playbook中写入分支条件。

例如，创建Playbook实现仅当 `/var/www/html/index.html` 不存在时，在目标主机上创建它的：

编辑 `playbook_sample.yml` 文件：

```
# 使用failed_when设置布尔值，如果布尔值为“1”，则创建“index.html”
- hosts: target_servers
  become: yes
  become_method: sudo
  tasks:
    - name: index file exists or not
      shell: test -f /var/www/html/index.html
      ignore_errors: true
      register: file_exists
      failed_when: file_exists.rc not in [0, 1]

    - name: put index.html
      shell: echo "httpd index" > /var/www/html/index.html
      when: file_exists.rc == 1
```

```
ansible-playbook playbook_sample.yml --ask-become-pass
```

```
SUDO password:  
  
PLAY [target_servers] ****  
*****  
  
GATHERING FACTS ****  
*****  
ok: [10.0.0.52]  
ok: [10.0.0.51]  
  
TASK: [index file exists or not] ****  
*****  
changed: [10.0.0.52]  
changed: [10.0.0.51]  
  
TASK: [put index.html] ****  
*****  
skipping: [10.0.0.52]  
changed: [10.0.0.51]  
  
PLAY RECAP ****  
*****  
10.0.0.51 : ok=3     changed=2     unreachable=0  
      failed=0  
10.0.0.52 : ok=2     changed=1     unreachable=0  
      failed=0
```

### 附1.8.3.6. 使用 Playbook4

要使用“notify”，“handlers”，可以在完成“notify”方法的任务后执行在“handlers”中定义的任务。

例如，创建一个在编辑 `sshd_config` 后重新启动 `sshd` 的Playbook：

编辑 `playbook_sample.yml` 文件：

```
- hosts: target_servers
become: yes
become_method: sudo
handlers:
- name: restart sshd
  service: name=sshd state=restarted
tasks:
- name: edit sshd_config
  lineinfile: >
    dest=/etc/ssh/sshd_config
    regexp="{{ item-regexp }}"
    line="{{ item.line }}"
  with_items:
  - { regexp: '^#PermitRootLogin', line: 'PermitRootLogin no' }
  notify: restart sshd
  tags: Edit_sshd_config
```

```
ansible-playbook playbook_sample.yml --ask-become-pass
```

SUDO password:

```
PLAY [target_servers] *****
*****
GATHERING FACTS *****
*****
ok: [10.0.0.51]
ok: [10.0.0.52]

TASK: [edit sshd_config] *****
*****
changed: [10.0.0.52] => (item={'regexp': '^#PermitRootLogin', 'line': 'PermitRootLogin no'})
changed: [10.0.0.51] => (item={'regexp': '^#PermitRootLogin', 'line': 'PermitRootLogin no'})

NOTIFIED: [restart sshd] *****
*****
changed: [10.0.0.51]
changed: [10.0.0.52]

PLAY RECAP *****
*****
10.0.0.51 : ok=3    changed=2    unreachable=0
              failed=0
10.0.0.52 : ok=3    changed=2    unreachable=0
              failed=0
```

```
ansible target_servers -m command -a "grep '^PermitRootLogin' /etc/ssh/sshd_config" -b --ask-become-pass # 确认
```

SUDO password:

```
10.0.0.51 | success | rc=0 >>
PermitRootLogin no

10.0.0.52 | success | rc=0 >>
PermitRootLogin no
```

### 附1.8.3.7. 使用Playbook5

可以从其他Playbook包含（include）任务或Playbook。

如果想包含其他任务，在“tasks”部分中写入 `include: ***` :

编辑 `playbook_sample.yml` 文件：

```
# 包含在“tasks”目录中的“included.yml”
- hosts: target_servers
  become: yes
  become_method: sudo
  tasks:
    - include: tasks/included.yml
      vars:
        general_packages: vim-enhanced,wget,unzip
```

`mkdir tasks`

编辑 `tasks/included.yml` 文件：

```
- name: General packages are installed
  yum: name="{{ item }}" state=installed
  with_items:
    - "{{ general_packages }}"
  tags: General_Packages
```

`ansible-playbook playbook_sample.yml --ask-become-pass`

SUDO password:

```
PLAY [target_servers] ****
*****
GATHERING FACTS ****
*****
ok: [10.0.0.52]
ok: [10.0.0.51]

TASK: [General packages are installed] ****
*****
changed: [10.0.0.52] => (item=vim-enhanced,wget,unzip)
changed: [10.0.0.51] => (item=vim-enhanced,wget,unzip)

PLAY RECAP ****
*****
10.0.0.51 : ok=2     changed=1     unreachable=0
              failed=0
10.0.0.52 : ok=2     changed=1     unreachable=0
              failed=0
```

如果想包含其他Playbook，如下写入：

编辑 `playbook_sample.yml` 文件：

```
- hosts: target_servers
become: yes
become_method: sudo
tasks:
  - include: tasks/included.yml
    vars:
      general_packages: vim-enhanced,wget,unzip
# 包含其他Playbook
- include: httpd.yml
```

编辑 `httpd.yml` 文件：

```
- hosts: target_servers
become: yes
become_method: sudo
tasks:
- name: httpd is installed
  yum: name=httpd state=installed
- name: httpd is running and enabled
  service: name=httpd state=started enabled=yes
```

```
ansible-playbook playbook_sample.yml --ask-become-pass
```

SUDO password:

```
PLAY [target_servers] *****
*****
GATHERING FACTS *****
*****
ok: [10.0.0.52]
ok: [10.0.0.51]

TASK: [General packages are installed] *****
*****
changed: [10.0.0.51] => (item=vim-enhanced,wget,unzip)
changed: [10.0.0.52] => (item=vim-enhanced,wget,unzip)

PLAY [target_servers] *****
*****

GATHERING FACTS *****
*****
ok: [10.0.0.52]
ok: [10.0.0.51]

TASK: [httpd is installed] *****
*****
ok: [10.0.0.51]
ok: [10.0.0.52]

TASK: [httpd is running and enabled] *****
*****
ok: [10.0.0.51]
ok: [10.0.0.52]

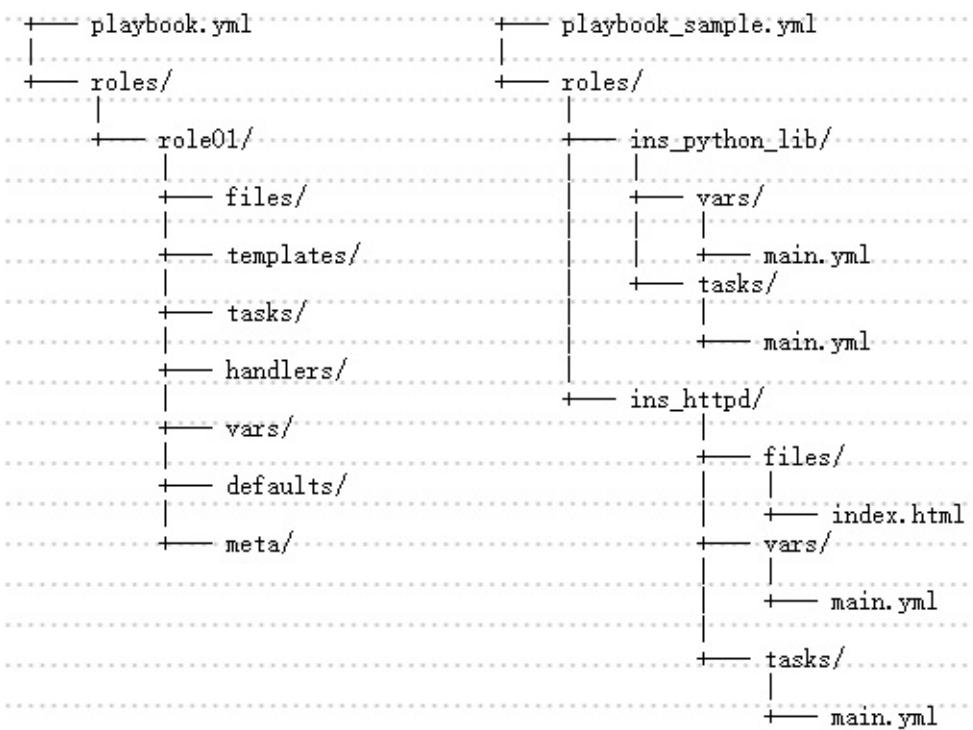
PLAY RECAP *****
*****
10.0.0.51 : ok=5     changed=1     unreachable=0
              failed=0
10.0.0.52 : ok=5     changed=1     unreachable=0
              failed=0
```

### 附1.8.3.8. 使用Playbook6

这是使用“Roles”（角色）功能的示例。

使用角色，可以包含其他任务或Playbook，而无需编写 `include` 语句。

需要如下配置目录树，以使用角色（左图显示所有树，右图显示本例的Playbook）：



例如，创建一个安装Python模块和httpd的Playbook：

```
mkdir -p roles/ins_python_lib/{tasks,vars}
```

```
mkdir -p roles/ins_httpd/{files,tasks,vars}
```

编辑 `playbook_sample.yml` 文件：

```

- hosts: target_servers
  become: yes
  become_method: sudo
  roles:
    - ins_python_lib
    - ins_httpd

```

编辑 `roles/ins_python_lib/vars/main.yml` 文件：

```
setuptools:  
  - python-setuptools  
  
py_pip:  
  - pip  
  
py_libs:  
  - httpplib2
```

编辑 `roles/ins_python_lib/tasks/main.yml` 文件：

```
- name: setuptools is installed  
  yum: name="{{ item }}" state=installed  
  with_items:  
    - "{{ setuptools }}"  
  tags: install_Setuptools  
  
- name: pip is installed  
  easy_install: name="{{ item }}"  
  with_items:  
    - "{{ py_pip }}"  
  tags: install_pip  
  
- name: httpplib2 are installed  
  pip: name="{{ item }}"  
  with_items:  
    - "{{ py_libs }}"  
  tags: install_httpplib2
```

编辑 `roles/ins_httpd/vars/main.yml` 文件：

```
packages:  
  - httpd
```

编辑 `roles/ins_httpd/tasks/main.yml` 文件：

```
- name: httpd is installed
  yum: name="{{ item }}" state=installed
  with_items:
    - "{{ packages }}"
  tags: install_httpd

- name: edit httpd.conf
  lineinfile:
    dest=/etc/httpd/conf/httpd.conf
    regexp="{{ item.regex }}"
    line="{{ item.line }}"
  with_items:
    - { regexp: "^#ServerName", line: "ServerName {{ ansible_fqdn }}:80" }
  tags: edit_httpd.conf

- name: httpd is running and enabled
  service: name=httpd state=started enabled=yes

- name: put index.html
  copy: src=index.html dest=/var/www/html owner=root group=root mode=0644

- name: check httpd
  uri: url=http://{{ ansible_fqdn }}
```

```
echo "httpd index page" > roles/ins_httpd/files/index.html
```

```
ansible-playbook playbook_sample.yml --ask-become-pass
```

SUDO password:

```
PLAY [target_servers] ****
*****
```

```
GATHERING FACTS ****
*****
```

```
ok: [10.0.0.51]
ok: [10.0.0.52]
```

```
TASK: [ins_python_lib | setuptools is installed] ****
```

```
*****
ok: [10.0.0.51] => (item=python-setuptools)
changed: [10.0.0.52] => (item=python-setuptools)

TASK: [ins_python_lib | pip is installed] ****
*****
ok: [10.0.0.51] => (item=pip)
changed: [10.0.0.52] => (item=pip)

TASK: [ins_python_lib | httpplib2 are installed] ****
*****
ok: [10.0.0.51] => (item=httpplib2)
changed: [10.0.0.52] => (item=httpplib2)

TASK: [ins_httpd | httpd is installed] ****
*****
changed: [10.0.0.51] => (item=httpd)
changed: [10.0.0.52] => (item=httpd)

TASK: [ins_httpd | edit httpd.conf] ****
*****
changed: [10.0.0.51] => (item={'regexp': '^#ServerName', 'line': u'HostName node01.srv.world:80'})
changed: [10.0.0.52] => (item={'regexp': '^#ServerName', 'line': u'HostName node02.srv.world:80'})

TASK: [ins_httpd | httpd is running and enabled] ****
*****
changed: [10.0.0.51]
changed: [10.0.0.52]

TASK: [ins_httpd | put index.html] ****
*****
changed: [10.0.0.52]
changed: [10.0.0.51]

TASK: [ins_httpd | check httpd] ****
*****
ok: [10.0.0.52]
ok: [10.0.0.51]

PLAY RECAP ****
```

```
*****
10.0.0.51          : ok=9      changed=7      unreachable=0
  failed=0
10.0.0.52          : ok=9      changed=7      unreachable=0
  failed=0
```

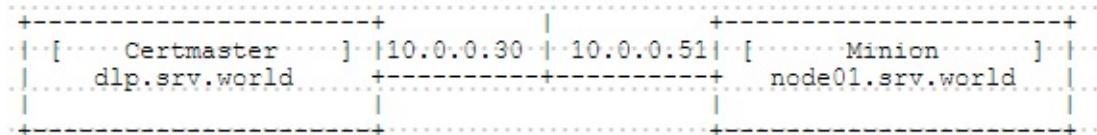
## 附1.8.4. Func

Func是由红帽公司以Fedora平台统一网络控制器（Fedora Unified Network Controller），目的是为了解决这一系列统一管理监控问题而设计开发的系统管理基础框架。

### 附1.8.4.1. 安装Func

Func可以从管理服务器（称为“Certmaster”）操作远程服务器（称为“Minion”）。

本例基于以下环境配置：



在所有节点上安装用于Certmaster/Minion的Func：

```
yum --enablerepo=epel -y install func # 从EPEL安装
```

在Certmaster服务器上启动“certmaster”服务：

```
systemctl start certmaster
systemctl enable certmaster
```

“certmaster”服务firewalld防火墙规则：

```
firewall-cmd --add-port=51235/tcp --permanent
firewall-cmd --reload
```

在Minion服务器上启动“funcd”服务。配置Certmaster服务器，也作为一个Minion：

编辑 /etc/certmaster/minion.conf 文件：

```
# configuration for minions

[main]
# Certmaster服务器的主机名或IP地址
certmaster = dlp.srv.world
certmaster_port = 51235
log_level = DEBUG
cert_dir = /etc/pki/certmaster
```

```
systemctl start funcd
systemctl enable funcd
```

“funcd”服务firewalld防火墙规则：

```
firewall-cmd --add-port=51234/tcp --permanent
firewall-cmd --reload
```

当Funcd在Minion上初始运行时，需要如下所示签名Minion的证书：

```
certmaster-ca --list # 显示请求
```

```
dlp.srv.world
node01.srv.world
```

给它们签名：

```
certmaster-ca --sign dlp.srv.world
```

```
/var/lib/certmaster/certmaster/csrs/dlp.srv.world.csr signed - cert located at /var/lib/certmaster/certmaster/certs/dlp.srv.world.cert
```

```
certmaster-ca --sign node01.srv.world
```

```
/var/lib/certmaster/certmaster/csrs/node01.srv.world.csr signed  
- cert located at /var/lib/certmaster/certmaster/certs/node01.srv.world.cert
```

```
func "*" list_minions # 显示各Minion
```

```
d1p.srv.world  
node01.srv.world
```

### 附1.8.4.2. 基本操作

在Certmaster服务器上运行命令到Minion。基本用法如下：  
func target call  
module method [args ...]

显示Minion：

```
func "*" list_minions # 显示所有Minion
```

```
d1p.srv.world  
node01.srv.world
```

func "d1p.srv.world;node01.srv.world" list\_minions # 指定Minion并显示

```
d1p.srv.world  
node01.srv.world
```

显示所有可用的模块（module）：

func "node01.srv.world" call system list\_modules # 显示 node01.srv.world 的模块

```
{'node01.srv.world': ['bridge',  
                      'certmastermod',  
                      'command',  
                      'confmgt_augeas',  
                      'copyfile',  
                      'cpu',
```

```
'delegation',
'disk',
'djangoctl',
'echo',
'fact',
'filetracker',
'func_getargs',
'func_module',
'getfile',
'hardware',
'httpd',
'iptables',
'iptables.port',
'jboss',
'jobs',
'mount',
'nagios',
'nagios_check',
'netapp.options',
'netapp.snap',
'netapp.vol',
'netapp.vol.clone',
'networktest',
'overlord',
'portinfo',
'process',
'pullfile',
'reboot',
'rpms',
'service',
'smart',
'snmp',
'sysctl',
'test',
'users',
'velan',
'yumcmd']}
```

显示模块的所有可用方法（method）：

```
func "node01.srv.world" call command list_methods # 显示“command”模块的方法
```

```
{'node01.srv.world': ['run',
                      'config_items',
                      'grep',
                      'exists',
                      'save_config',
                      'module_version',
                      'grep',
                      'list_methods',
                      'module_description',
                      'get_method_args',
                      'module_api_version']}
```

### 附1.8.4.3. 使用YumModule

YumModule的基本用法。

在所有Minion上运行 `yum update` :

```
func "*" call yumcmd update

('node01.srv.world',
 'command: update \nkernel-tools.x86_64 0:3.10.0-327.36.1.el7 -
 u\nbind-license.noarch 32:9.9.4-29.el7_2.4 - u\n.....
('dlp.srv.world',
 'command: update \nkernel-tools.x86_64 0:3.10.0-327.36.1.el7 -
 u\nbind-license.noarch 32:9.9.4-29.el7_2.4 - u\n.....
```

在指定的Minion上安装指定的软件包 (`func`命令结果显示 `\n` , 很难阅读, 所以将其替换为真正的换行符可以轻松阅读) :

```
func "node01.srv.world" call yumcmd install "httpd mariadb-server"
| sed 's/\n/\n/g'
```

```
{'node01.srv.world': 'command: install httpd mariadb-server
mariadb-server.x86_64 1:5.5.50-1.el7_2
- uhttpd.x86_64 0:2.4.6-40.el7.centos.4 - u'}
```

#### 附1.8.4.4. 使用CopyFileModule

CopyFileModule的基本用法。

将Certmaster上的 /root/test.txt 复制到所有Minion上的 /home :

```
func "*" copyfile -f /root/test.txt --remotepath /home/test.txt

func "*" call command run "ls -l /home" | sed 's/\n/\n/g' # 运行 ls 来确认
```

```
('node01.srv.world',
[0,
 'total 4
drwx----- 2 cent cent 59 Aug  8 2015 cent
-rw-r--r-- 1 root root 10 Sep 30 16:46 test.txt
',
 '')
('dlp.srv.world',
[0,
 'total 4
drwx----- 2 cent cent 59 Aug  8 2015 cent
-rw-r--r-- 1 root root 10 Sep 30 16:46 test.txt
',
 '')
```

#### 附1.8.4.5. 使用CommandModule

CommandModule的基本用法。

实际上，这个模块可以运行几乎任何命令： func target call command run "command"

在所有Minion上安装“wget”：

```
('node01.srv.world',
[0,
 'Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.riken.jp
 * extras: ftp.riken.jp
 * updates: ftp.riken.jp
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-10.el7_0.1 will be installed
--> Finished Dependency Resolution
.....
.....
Installed:
  wget.x86_64 0:1.14-10.el7_0.1

Complete!
',
 '')
('dlp.srv.world',
[0,
 'Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.riken.jp
 * extras: ftp.riken.jp
 * updates: ftp.riken.jp
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-10.el7_0.1 will be installed
--> Finished Dependency Resolution
.....
.....
Installed:
  wget.x86_64 0:1.14-10.el7_0.1

Complete!
',
 '')
```

在所有Minion上运行“ls /home”：

```
func "*" call command run "ls -l /home" | sed 's/\n/\n/g'
```

```
('node01.srv.world',
[0,
 'total 4
drwx----- 2 cent cent 59 Aug  8 2015 cent
-rw-r--r-- 1 root root 10 Sep 30 16:46 test.txt
',
 '')
('dlp.srv.world',
[0,
 'total 4
drwx----- 2 cent cent 59 Aug  8 2015 cent
-rw-r--r-- 1 root root 10 Sep 30 16:46 test.txt
',
 '')
```

在指定的Minion上运行“cat /etc/passwd”：

```
func "node01.srv.world" call command run "cat /etc/passwd" | sed
's/\n/\n/g'
```

```
('node01.srv.world',
[0,
 'root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
...
...
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin
',
 '')
```

在所有Minion的 /home 下创建一个目录：

```
func "*" call command run "mkdir /home/test.dir"
```

```
('node01.srv.world', [0, '', ''])
('dlp.srv.world', [0, '', ''])
```

```
func "*" call command run "ls -ld /home/test.dir" | sed  
's/\n/g'
```

```
('node01.srv.world',  
 [0, 'drwx----- 2 root root 6 Sep 30 17:01 /home/test.dir  
, '')  
('dlp.srv.world',  
 [0, 'drwx----- 2 root root 6 Sep 30 17:01 /home/test.dir  
, '')
```

在所有Minion的 `/home/test.txt` 运行更改所有者为“nobody”和权限为“600”：

```
func "*" call command run "chown nobody. /home/test.txt;chmod 600  
/home/test.txt"
```

```
('node01.srv.world', [0, '', ''])  
('dlp.srv.world', [0, '', ''])
```

```
func "*" call command run "ls -l /home/test.txt" | sed  
's/\n/g'
```

```
('node01.srv.world',  
 [0, '-rw----- 1 nobody nobody 10 Sep 30 16:46 /home/test.txt  
, '')  
('dlp.srv.world',  
 [0, '-rw----- 1 nobody nobody 10 Sep 30 16:46 /home/test.txt  
, '')
```

在所有Minion上重新启动ntpd：

```
func "*" call command run "systemctl restart ntpd" | sed  
's/\n/g'
```

```
('node01.srv.world', [0, '', ''])  
('dlp.srv.world', [0, '', ''])
```

## 附1.9. 防火墙

### 附1.9.1. Firewalld

Firewalld 是一款提供 D-Bus 接口从而支持动态管理的防火墙守护进程。红帽上的 Firewalld 使用指南，fedoraproject 上的手册（中文），AtomicGain 上的一些应用示例。

#### 附1.9.1.1. 基本操作

Firewalld 上服务的定义设置为防火墙上的区域（zone）。要启用防火墙，将区域与网卡相关联的命令关联。

要使用 Firewalld，先启动：

```
systemctl start firewalld  
systemctl enable firewalld
```

默认，“public”区域应用到网卡，“dhcpcv6-client”和“ssh”被允许。当使用 `firewall-cmd` 命令运行时，如果输入的命令没有 `--zone=***` 格式，则配置将设置为默认区域。

```
firewall-cmd --get-default-zone # 显示默认区域
```

```
public
```

```
firewall-cmd --list-all # 显示当前设置
```

```
public (default, active)
  interfaces: eno16777736
  sources:
  services: dhcpcv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

```
firewall-cmd --list-all-zones # 显示默认定义的所有区域
```

```
block
  interfaces:
  sources:
  services:
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
  ....
  ....
```

```
firewall-cmd --list-service --zone=external # 在指定区域显示允许的服务
```

```
ssh
```

```
firewall-cmd --set-default-zone=external # 更改默认区域
```

```
success
```

```
firewall-cmd --add-interface=eth1 --zone=internal # 将没有设定区域的接口添加到指定区域，可以添加 --permanent 选项，在重启后也生效
```

```
firewall-cmd --remove-interface=eth1 --zone=internal # 将接口从所在区域删除，可以添加 --permanent 选项，在重启后也生效
```

```
firewall-cmd --change-interface=eth1 --zone=external # 更改接口的区域（即使添加了 --permanent 选项，使用 change-interface 也不会永久更改）
```

实际（在Ubuntu16.04.3中安装的firewalld）测试，添加 --permanent 选项只是在未重启时使用 firewall-cmd --reload 重载防火墙不会生效，但重启（reboot）还是会生效，可以添加与不添加命令连续运行，即可实现生效且重启后也有效。补充：CentOS测试结果一样。

success

```
firewall-cmd --list-all --zone=external
```

```
external (active)
interfaces: eth1
sources:
services: ssh
ports:
masquerade: yes
forward-ports:
icmp-blocks:
rich rules:
```

```
nmcli c mod eth1 connection.zone external # 如果要永久更改，使用 nmcli
```

```
firewall-cmd --get-active-zone
```

```
external
interfaces: eth1
public
interfaces: eth0
```

显示默认定义的服务：

```
firewall-cmd --get-services
```

```
amanda-client bacula bacula-client dhcp dhcpcv6 dhcpcv6-client dns  
ftp high-availability http https imaps ipp ipp-client ipsec ker  
beros kpasswd ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt  
mysql nfs ntp openvpn pmcd pmproxy pmwebapis pop3s post  
gresql proxy-dhcp radius rpc-bind samba samba-client smtp ssh te  
lnet tftp tftp-client transmission-client vnc-server wbem-https
```

定义文件位于以下位置，如果要添加原始定义，在其中添加XML文件：

```
ls /usr/lib/firewalld/services
```

|                       |                 |                |                 |
|-----------------------|-----------------|----------------|-----------------|
| amanda-client.xml     | ipp-client.xml  | mysql.xml      | rpc-bind        |
| .xml                  |                 |                |                 |
| bacula-client.xml     | ipp.xml         | nfs.xml        | samba-cl        |
| ient.xml              |                 |                | ient.xml        |
| bacula.xml            | ipsec.xml       | ntp.xml        | samba.xm        |
| l                     |                 |                | l               |
| dhcpcv6-client.xml    | kerberos.xml    | openvpn.xml    | smtp.xml        |
| dhcpcv6.xml           | kpasswd.xml     | pmcd.xml       | ssh.xml         |
| dhcp.xml              | ldaps.xml       | pmproxy.xml    | telnet.x        |
| ml                    |                 |                | ml              |
| dns.xml               | ldap.xml        | pmwebapis.xml  | tftp-cl         |
| ent.xml               |                 |                | ient.xml        |
| ftp.xml               | libvirt-tls.xml | pmwebapi.xml   | tftp.xml        |
| high-availability.xml | libvirt.xml     | pop3s.xml      | transmis        |
| sion-client.xml       |                 |                | sion-client.xml |
| https.xml             | mdns.xml        | postgresql.xml | vnc-serv        |
| er.xml                |                 |                | er.xml          |
| http.xml              | mountd.xml      | proxy-dhcp.xml | wbem-htt        |
| ps.xml                |                 |                | ps.xml          |
| imaps.xml             | ms-wbt.xml      | radius.xml     |                 |

添加或删除允许的服务，重新启动系统后，更改将恢复。如果要永久更改设置，添加 `--permanent` 选项：

```
firewall-cmd --add-service=http # 例如，添加“http”（更改仅一次有效）
```

```
success
```

```
firewall-cmd --list-service
```

```
dhcpv6-client http ssh
```

```
firewall-cmd --remove-service=http # 删除“http”
```

```
success
```

```
firewall-cmd --list-service
```

```
dhcpv6-client ssh
```

```
firewall-cmd --add-service=http --permanent # 永久添加“http”
```

```
success
```

```
firewall-cmd --reload # 永久的情况，需要重新加载Firewalld以启用更改
```

```
success
```

```
firewall-cmd --list-service
```

```
dhcpv6-client http ssh
```

添加或删除允许的端口：

```
firewall-cmd --add-port=465/tcp # 例如，添加“465/TCP”
```

```
success
```

```
firewall-cmd --list-port
```

```
465/tcp
```

```
firewall-cmd --remove-port=465/tcp # 删除“465/TCP”
```

success

```
firewall-cmd --list-port
```

# 无内容

```
firewall-cmd --add-port=465/tcp --permanent # 永久添加“465/TCP”
```

success

```
firewall-cmd --reload
```

success

```
firewall-cmd --list-port
```

465/tcp

添加或删除禁止的ICMP类型：

```
firewall-cmd --add-icmp-block=echo-request # 例如，添加“echo-request”来禁止它
```

success

```
firewall-cmd --list-icmp-blocks
```

echo-request

```
firewall-cmd --remove-icmp-block=echo-request # 删除“echo-request”
```

success

```
firewall-cmd --list-icmp-blocks
```

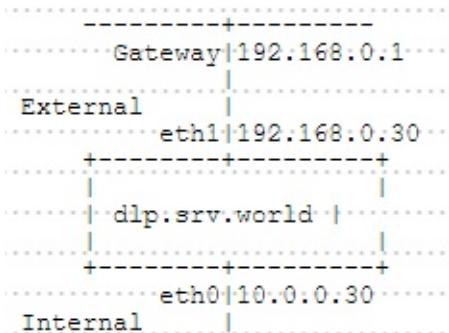
```
# 无内容
```

```
firewall-cmd --get-icmptypes # 显示ICMP类型
```

```
destination-unreachable echo-reply echo-request parameter-problem redirect router-advertisement router-solicitation source-quench time-exceeded
```

### 附1.9.1.2. IP伪装

基于以下环境演示如何配置防火墙的IP伪装：



更改接口的区域：

```
firewall-cmd --get-active-zone # 显示当前设置
```

```
public
  interfaces: eth0 eth1
```

```
nmcli c mod eth0 connection.zone internal # 更改“eth0”区域为“internal”
```

```
nmcli c mod eth1 connection.zone external # 更改“eth1”区域为“external”
```

```
firewall-cmd --get-active-zone
```

```
internal
  interfaces: eth0
external
  interfaces: eth1
```

在External区域设置IP伪装：

```
firewall-cmd --zone=external --add-masquerade --permanent # 设置IP  
伪装
```

success

```
firewall-cmd --reload
```

success

```
firewall-cmd --zone=external --query-masquerade
```

yes

```
cat /proc/sys/net/ipv4/ip_forward # 如果启用伪装，“ip_forward”将自动启  
用
```

1

例如，配置将进入External区域的22端口的数据包转发到本地1234端口（如果要永久设置，添加 `--permanent` 选项）：

```
firewall-cmd --zone=external --add-forward-  
port=port=22:proto=tcp:toport=1234
```

success

```
firewall-cmd --list-all --zone=external
```

```
external (active)
  interfaces: eth1
  sources:
  services: ssh
  ports:
  masquerade: yes
  forward-ports: port=22:proto=tcp:toport=1234:toaddr=
  icmp-blocks:
  rich rules:
```

例如，配置将进入External区域的22端口的数据包转发到另一个主机（10.0.0.31）  
(注：原文是192.168.0.31，但感觉应该是错了) 的22端口：

```
firewall-cmd --zone=external --add-forward-
port=port=22:proto=tcp:toport=22:toaddr=10.0.0.31
```

success

```
firewall-cmd --list-all --zone=external
```

```
external (active)
  interfaces: eth1
  sources:
  services: ssh
  ports:
  masquerade: yes
  forward-ports: port=22:proto=tcp:toport=22:toaddr=10.0.0.31
  icmp-blocks:
  rich rules:
```

例如，配置允许通过Internal网络（10.0.0.0/24）内的服务器传出数据包，并转发到External端：

```
firewall-cmd --zone=internal --add-masquerade --permanent # 给
internal区域设置伪装
```

success

```
firewall-cmd --reload
```

success

```
firewall-cmd --direct --add-rule ipv4 nat POSTROUTING 0 -o eth1 -j MASQUERADE
```

```
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i eth0 -o eth1 -j ACCEPT
```

```
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

### 附1.9.1.3. 收集的其他一些

对指定的IP地址开放端口，如对192.168.1.101开放3306/tcp：

```
firewall-cmd --permanent --add-rich-rule="rule family="ipv4" source address="192.168.1.101" port protocol="tcp" port="3306" accept"  
firewall-cmd --reload
```

查看配置：

```
firewall-cmd --list-all
```

删除配置：

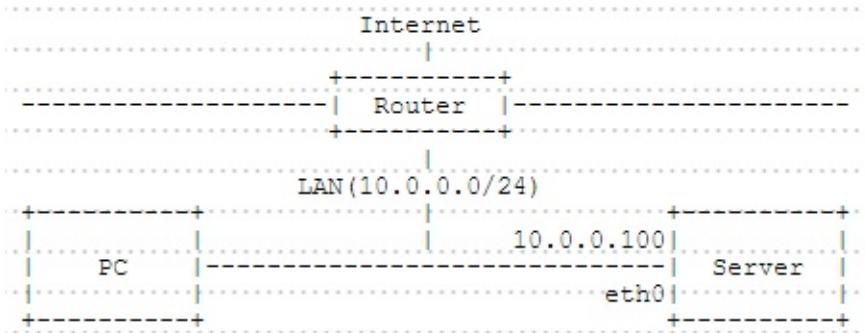
```
firewall-cmd --permanent --remove-rich-rule="rule family="ipv4" source address="192.168.1.101" port protocol="tcp" port="3306" accept"
```

## 附1.9.2. iptables

[iptables](#)是与Linux内核集成的IP信息包过滤系统。[红帽上的iptables使用指南](#)，[frozentux上的Iptables指南](#)。

### 附1.9.2.1. 设置示例1

本例基于以下环境：



- DROP INPUT by Default
- ACCEPT OUTPUT by Default
- ACCEPT Established Connection
- ACCEPT the Connection from loopback
- ACCEPT Ping Connection for 5 times per a minites from internal network(10.0.0.0/24)
- ACCEPT SSH Connection from internal network(10.0.0.0/24)

编辑 `iptables.sh` 文件：

```
#!/bin/bash

trust_host='10.0.0.0/24'
my_host='10.0.0.100'

/sbin/iptables -F
/sbin/iptables -X

/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD DROP

/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT

/sbin/iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT

/sbin/iptables -A INPUT -p icmp --icmp-type echo-request -s $trus
t_host \
-d $my_host -m limit --limit 1/m --limit-burst 5 -j ACCEPT

/sbin/iptables -A INPUT -p tcp -m state --state NEW -m tcp -s $tr
ust_host \
-d $my_host --dport 22 -j ACCEPT

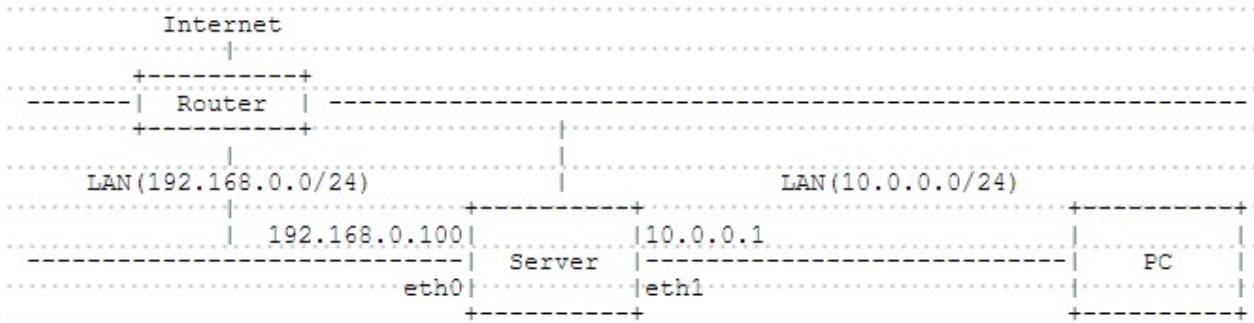
/etc/rc.d/init.d/iptables save
/etc/rc.d/init.d/iptables restart
```

```
sh iptables.sh
```

```
iptables: Saving firewall rules to /etc/sysconfig/iptables: [ 0
K ]
iptables: Flushing firewall rules: [  OK  ]
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Unloading modules: [  OK  ]
iptables: Applying firewall rules: ip_tables: (C) 2000-2006 Netf
ilter Core Team
nf_conntrack version 0.5.0 (16384 buckets, 65536 max)
[  OK  ]
```

## 附1.9.2.2. 设置示例2

本例基于以下环境：



- DROP INPUT by Default
- ACCEPT OUTPUT by Default
- DROP FORWARD by Default
- ACCEPT Established Connection
- ACCEPT the Connection from loopback
- ACCEPT Ping Connection for 5 times per a minites from internal network(10.0.0.0/24)
- ACCEPT SSH Connection from internal network(10.0.0.0/24)
- ACCEPT Outgoing Packets through the Server from internal network(10.0.0.0/24) and translatte the source address

编辑 `iptables.sh` 文件：

```
#!/bin/bash

trust_host='10.0.0.0/24'
my_host='10.0.0.100'

echo 1 > /proc/sys/net/ipv4/ip_forward

/sbin/iptables -F
/sbin/iptables -t nat -F
/sbin/iptables -X

/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD DROP

/sbin/iptables -A FORWARD -i eth1 -o eth0 -s $trust_host -j ACCEPT
/sbin/iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

/sbin/iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT

/sbin/iptables -A INPUT -p icmp --icmp-type echo-request -s $trust_host \
-d $my_host -m limit --limit 1/m --limit-burst 5 -j ACCEPT

/sbin/iptables -A INPUT -p tcp -m state --state NEW -m tcp -s $trust_host \
-d $my_host --dport 22 -j ACCEPT

/sbin/iptables -t nat -A POSTROUTING -o eth0 -s $trust_host -j MASQUERADE

/etc/rc.d/init.d/iptables save
/etc/rc.d/init.d/iptables restart
```

```
sh iptables.sh
```

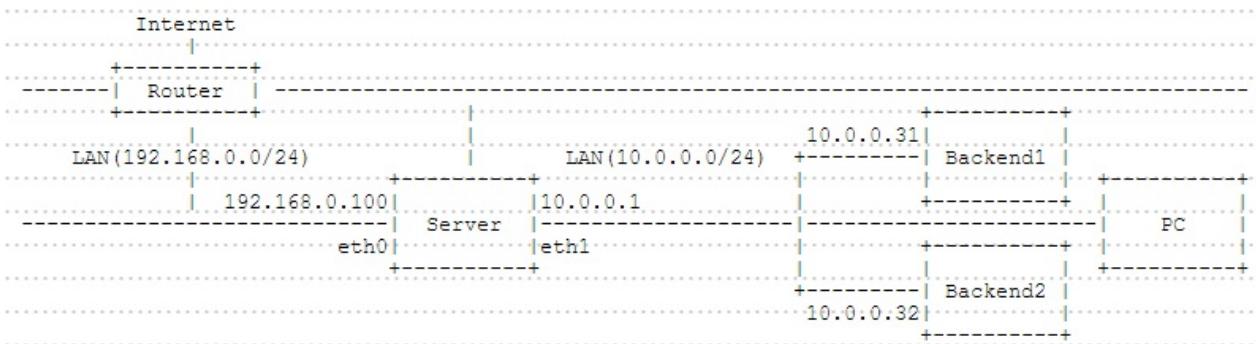
```

iptables: Saving firewall rules to /etc/sysconfig/iptables: [ 0
K ]
iptables: Flushing firewall rules: [  OK  ]
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Unloading modules: [  OK  ]
iptables: Applying firewall rules: ip_tables: (C) 2000-2006 Netf
ilter Core Team
nf_conntrack version 0.5.0 (16384 buckets, 65536 max)
[  OK  ]

```

### 附1.9.2.3. 设置示例3

本例基于以下环境：



- DROP INPUT by Default
- ACCEPT OUTPUT by Default
- DROP FORWARD by Default
- ACCEPT Established Connection
- ACCEPT the Connection from loopback
- Forward the Packets to 80 on eth0 to the same port on Backend1
- Forward the Packets to 443 on eth0 to the same port on Backend2
- But DROP the Packets from 192.168.0.20
- ACCEPT Ping Connection for 5 times per a minites from internal network(10.0.0.0/24)
- ACCEPT SSH Connection from internal network(10.0.0.0/24)
- But DROP the Packets from 10.0.0.20
- ACCEPT Outgoing Packets through the Server from internal network(10.0.0.0/24) and translatte the source address

编辑 `iptables.sh` 文件：

```
#!/bin/bash

trust_host='10.0.0.0/24'
my_internal_ip='10.0.0.1'
my_external_ip='192.168.0.100'

listen_port_1='80'
backend_host_1='10.0.0.31'
backend_port_1='80'

listen_port_2='443'
backend_host_2='10.0.0.32'
backend_port_2='443'

echo 1 > /proc/sys/net/ipv4/ip_forward

/sbin/iptables -F
/sbin/iptables -t nat -F
/sbin/iptables -X

/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD DROP

/sbin/iptables -A FORWARD -i eth1 -o eth0 -s $trust_host -j ACCEPT
/sbin/iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

/sbin/iptables -A FORWARD -s 192.168.0.20/32 -j DROP

/sbin/iptables -A FORWARD -p tcp --dst $backend_host_1 --dport $backend_port_1 -j ACCEPT
/sbin/iptables -A FORWARD -p tcp --dst $backend_host_2 --dport $backend_port_2 -j ACCEPT

/sbin/iptables -A INPUT -s 10.0.0.20/32 -j DROP

/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
/sbin/iptables -A INPUT -p icmp --icmp-type echo-request -s $trus
```

```
st_host \
-d $my_internal_ip -m limit --limit 1/m --limit-burst 5 -j ACCEPT
/sbin/iptables -A INPUT -p tcp -m state --state NEW -m tcp -s $trust_host \
-d $my_internal_ip --dport 22 -j ACCEPT

/sbin/iptables -t nat -A POSTROUTING -o eth0 -s $trust_host -j MASQUERADE

/sbin/iptables -t nat -A PREROUTING -p tcp --dst $my_external_ip
--dport $listen_port_1 \
-j DNAT --to-destination $backend_host_1:$backend_port_1
/sbin/iptables -t nat -A PREROUTING -p tcp --dst $my_external_ip
--dport $listen_port_2 \
-j DNAT --to-destination $backend_host_2:$backend_port_2

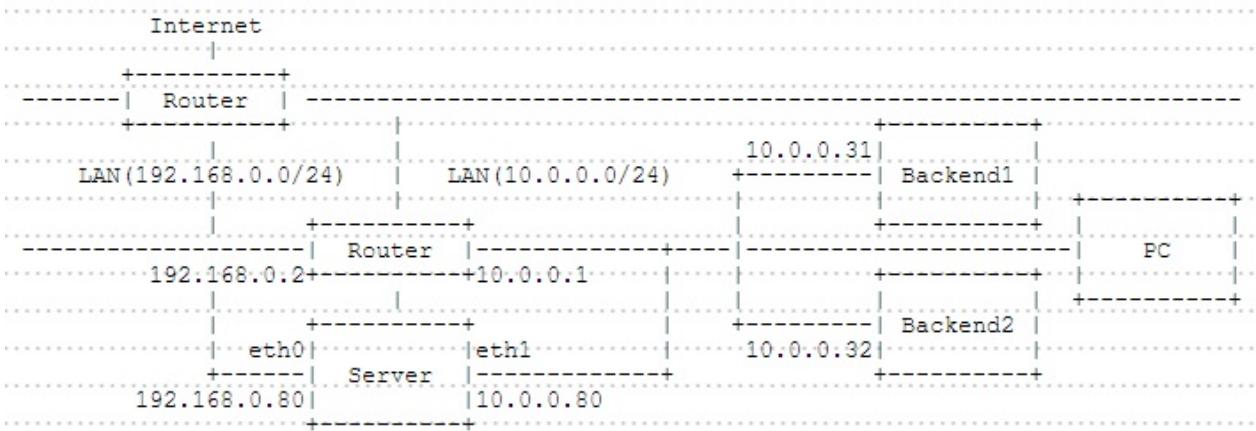
/etc/rc.d/init.d/iptables save
/etc/rc.d/init.d/iptables restart
```

```
sh iptables.sh
```

```
iptables: Saving firewall rules to /etc/sysconfig/iptables: [ OK ]
iptables: Flushing firewall rules: [ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Unloading modules: [ OK ]
iptables: Applying firewall rules: ip_tables: (C) 2000-2006 Netfilter Core Team
nf_conntrack version 0.5.0 (16384 buckets, 65536 max)
[ OK ]
```

### 附1.9.2.4. 设置示例4

本例基于以下环境：



- DROP INPUT by Default
- ACCEPT OUTPUT by Default
- DROP FORWARD by Default
- ACCEPT Established Connection
- ACCEPT the Connection from loopback
- Forward the Packets to 22 on eth0 to the same port on Backend1
- Forward the Packets to 80 on eth0 to the same port on Backend2
- ACCEPT Ping Connection for 5 times per a minites from internal network(10.0.0.0/24)
- ACCEPT SSH Connection from internal network(10.0.0.0/24)
- ACCEPT Outgoing Packets through the Server from internal network(10.0.0.0/24) and translatte the source address

编辑 `iptables.sh` 文件：

```
#!/bin/bash

trust_host='10.0.0.0/24'
my_internal_ip='10.0.0.80'
my_external_ip='192.168.0.80'

listen_port_1='22'
backend_host_1='10.0.0.31'
backend_port_1='22'

listen_port_2='80'
backend_host_2='10.0.0.32'
backend_port_2='80'
```

```

echo 1 > /proc/sys/net/ipv4/ip_forward

/sbin/iptables -F
/sbin/iptables -t nat -F
/sbin/iptables -X

/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD DROP

/sbin/iptables -A FORWARD -i eth1 -o eth0 -s $trust_host -j ACCEPT
/sbin/iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

/sbin/iptables -A FORWARD -p tcp --dst $backend_host_1 --dport $backend_port_1 -j ACCEPT
/sbin/iptables -A FORWARD -p tcp --dst $backend_host_2 --dport $backend_port_2 -j ACCEPT

/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
/sbin/iptables -A INPUT -p icmp --icmp-type echo-request -s $trust_host \
-d $my_internal_ip -m limit --limit 1/m --limit-burst 5 -j ACCEPT
/sbin/iptables -A INPUT -p tcp -m state --state NEW -m tcp -s $trust_host \
-d $my_internal_ip --dport 22 -j ACCEPT

/sbin/iptables -t nat -A POSTROUTING -o eth0 -s $trust_host -j MASQUERADE

/sbin/iptables -t nat -A PREROUTING -p tcp --dst $my_external_ip \
--dport $listen_port_1 \
-j DNAT --to-destination $backend_host_1:$backend_port_1
/sbin/iptables -t nat -A POSTROUTING -p tcp --dst $backend_host_1 \
--dport $backend_port_1 \
-j SNAT --to-source $my_internal_ip

/sbin/iptables -t nat -A PREROUTING -p tcp --dst $my_external_ip

```

```
--dport $listen_port_2 \
-j DNAT --to-destination $backend_host_2:$backend_port_2
/sbin/iptables -t nat -A POSTROUTING -p tcp --dst $backend_host_2
--dport $backend_port_2 \
-j SNAT --to-source $my_internal_ip

/etc/rc.d/init.d/iptables save
/etc/rc.d/init.d/iptables restart
```

```
sh iptables.sh
```

```
iptables: Saving firewall rules to /etc/sysconfig/iptables: [  OK  ]
iptables: Flushing firewall rules: [  OK  ]
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Unloading modules: [  OK  ]
iptables: Applying firewall rules: ip_tables: (C) 2000-2006 Netfilter Core Team
nf_conntrack version 0.5.0 (16384 buckets, 65536 max)
[  OK  ]
```

其他一些参数：

指定IP范围，可以通过计算子网掩码来设置，如果需要精确控制一个IP范围加入以下参数：

```
-m iprange --src-range 192.168.0.2-192.168.0.5 # 匹配来源地址范围
```

```
-m iprange --dst-range 192.168.1.11-192.168.1.15 # 匹配目的地址范围
```

### 附1.9.3. Fail2ban

Fail2ban扫描系统日志文件，例

如 `/var/log/pwdfail` 或 `/var/log/apache/error_log`，从中找出多次尝试登录失败的IP地址，并将该IP地址加入防火墙的拒绝访问列表中。

查的资料是CentOS6的，默认调用iptables防火墙，不知道CentOS7安装好后是不是直接调用firewalld，有空了再看看。

安装

```
yum --enablerepo=epel -y install fail2ban
```

安装完成后，`/etc/fail2ban` 目录下 `fail2ban.conf` 为日志设定文档，`jail.conf` 为阻挡设定文档。`/etc/fail2ban/filter.d` 为具体阻挡内容设定目录具体。

### 配置

创建全局配置（可以直接在 `/etc/fail2ban/jail.conf` 下面添加规则）：

```
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local # 复制 jail.conf 文件到 jail.local
```

编辑 `jail.local`：

```
# 修改如下配置
[DEFAULT]
# ignoreip用于指定哪些地址可以忽略fail2ban防御（白名单）
# 以空格分隔的列表，可以是IP地址、CIDR前缀或者DNS主机名
ignoreip = 127.0.0.1 10.10.0.2 192.168.0.0/24
# 客户端主机被禁止的时长（秒）
bantime = 86400
# 客户端主机被禁止前允许失败的次数
maxretry = 5
# 查找失败次数的时长（秒）
findtime = 600
#以上规则，fail2ban会自动禁止在最近10分钟内有超过5次访问尝试失败的非白名单内的IP地址，禁止时间24个小时
```

SSH防护配置，新建 `/etc/fail2ban/jail.d/sshd.local` 文件：

```
[ssh-iptables]
enabled = true
# "filter" = sshd"中sshd对应/etc/fail2ban/filter.d/sshd.conf
filter = sshd
# 如果ssh端口不是默认22，则修改为“port=具体端口号”
action = iptables[name=SSH, port=ssh, protocol=tcp]
#           sendmail-whois[name=SSH, dest=your@email.com, sender=
fail2ban@example.com]
# 日志路径，Debian系发行版为/var/log/auth.log
logpath = /var/log/secure
# 这里可以单独设置maxretry等参数，优先级比全局配置jail.local高
maxretry = 5
```

CentOS7启动：

```
systemctl enable fail2ban
systemctl start fail2ban
```

CentOS6启动：

```
chkconfig --level 23 fail2ban on
service fail2ban start
```

可以 `cat /etc/fail2ban/filter.d/sshd.conf` 查看fail2ban是如何拦截的  
(fail2ban使用正则表达式找出认证失败的条目，然后拦截对应的建立连接的IP“”)

使用命令 `fail2ban-client ping`，验证运行，正常显示：

```
Server replied: pong
```

测试fail2ban是否正常工作，尝试通过使用错误的密码来用SSH连接到服务器模拟一个暴力破解攻击。同时监控 `/var/log/fail2ban.log`，该文件记录在fail2ban中发生的任何敏感事件。

```
tail -f /var/log/fail2ban.log
```

邮件防护配置，举例（根据自己实际情况修改设置）：日志路径为 `/var/log/maillog`，认证错误的条目类似于：

```
Nov 15 03:15:36 mailserver: LOGIN FAILED, user=a@x.com, ip=[1.2.3.4]
Nov 15 05:20:51 mailserver: LOGIN FAILED, user=b, ip=[1.2.3.5]
```

新建 /etc/fail2ban/jail.d/mail.local 文件：

```
[mail-smtp]
enabled = true
filter = mail-smtp
action = iptables[name=smtp, port=25, protocol=tcp]
logpath = /var/log/maillog
bantime = 86400
findtime = 300
maxretry = 10

[mail-imap]
enabled = true
filter = mail-imap
action = iptables[name=imap, port=143, protocol=tcp]
logpath = /var/log/maillog
bantime = 86400
findtime = 300
maxretry = 10

[mail-pop3]
enabled = true
filter = mail-pop3
action = iptables[name=pop3, port=110, protocol=tcp]
logpath = /var/log/maillog
bantime = 86400
findtime = 300
maxretry = 10
```

在 /etc/fail2ban/filter.d 目录下新建 mail-smtp.conf 、 mail-imap.conf 和 mail-pop3.conf ，此处内容相同，如下：

```
# Fail2Ban configuration file
#
# Author: Bill Landry ((email_protected))
#
# $Revision: 510 $
[Definition]

# Option: failregex
# Notes.: regex to match the password failures messages in the logfile. The
# host must be matched by a group named "host". The tag "" can
# be used for standard IP/hostname matching and is only an alias
# for
# (?:::f{4,6}::)?(?P<host>)
# Values: TEXT
# 以下正则表达式根据实际内容修改，如有多个规则，每行写一个

failregex = FAILED.*\[<HOST>\]$
          FAILED.*a@x.com.*\[<HOST>\]$


# Option: ignoreregex
# Notes.: regex to ignore. If this regex matches, the line is ignored.
# Values: TEXT

ignoreregex =
```

如果查找规则相同可以只使用一条查找规则，`/etc/fail2ban/jail.d/mail.local` 中对应规则名也修改即可。

设置完成后重启fail2ban。

以此类推，只要有符合规范的日志文件的服务应该都可以按上面的方法来实现防护。

开启防护后可以通过 `iptables -L` 或 `iptables -L -nv` 查看状态，会多出类似下面的内容（下例表示已拦截1.2.3.4对三个服务的访问）：

```

Chain f2b-imap (1 references)
pkts bytes target  prot opt in  out  source       destination
      0     0 REJECT all  --   *    *    1.2.3.4    0.0.0.0/0    r
eject-with icmp-port-unreachable
281K   64M RETURN  all  --   *    *    0.0.0.0/0  0.0.0.0/0

Chain f2b-smtp (1 references)
pkts bytes target  prot opt in  out  source       destination
      0     0 REJECT all  --   *    *    1.2.3.4    0.0.0.0/0    r
eject-with icmp-port-unreachable
32042  38M RETURN  all  --   *    *    0.0.0.0/0  0.0.0.0/0

Chain f2b-pop3 (1 references)
pkts bytes target  prot opt in  out  source       destination
      0     0 REJECT all  --   *    *    1.2.3.4    0.0.0.0/0    r
eject-with icmp-port-unreachable
32042  38M RETURN  all  --   *    *    0.0.0.0/0  0.0.0.0/0

```

通过上面查看到拦截信息，如果要移除已拦截的IP地址（如1.2.3.4），运行：

```
iptables -D f2b-imap -s 1.2.3.4 -j REJECT # f2b-imap 对应要移除的规则，1.2.3.4 修改为要移除的IP
```

运行后再 `iptables -L -nv`：

```

Chain f2b-imap (1 references)
pkts bytes target  prot opt in  out  source       destination
281K   64M RETURN  all  --   *    *    0.0.0.0/0  0.0.0.0/0

Chain f2b-smtp (1 references)
pkts bytes target  prot opt in  out  source       destination
      0     0 REJECT all  --   *    *    1.2.3.4    0.0.0.0/0    r
eject-with icmp-port-unreachable
32042  38M RETURN  all  --   *    *    0.0.0.0/0  0.0.0.0/0

Chain f2b-pop3 (1 references)
pkts bytes target  prot opt in  out  source       destination
      0     0 REJECT all  --   *    *    1.2.3.4    0.0.0.0/0    r
eject-with icmp-port-unreachable
6256   25M RETURN  all  --   *    *    0.0.0.0/0  0.0.0.0/0

```

也可以使用 `fail2ban-client` 命令行工具来查看和管理fail2ban的IP阻塞列表：

```
fail2ban-client status
```

```
Status
|- Number of jail:    4
`- Jail list:    ssh-iptables, mail-smtp, mail-imap, mail-pop3
```

查看指定监狱的状态（如`mail-smtp`）：

```
fail2ban-client status mail-smtp
```

```
Status for the jail: mail-smtp
|- Filter
| |- Currently failed:    0
| |- Total failed:    4
| ` File list:    /var/log/maillog
`- Actions
 |- Currently banned:    0
 |- Total banned:    0
 ` Banned IP list:    1.2.3.4
```

解锁指定的IP：

```
fail2ban-client set mail-smtp unbanip 1.2.3.4
```

注：

如果停止了Fail2ban服务，那么所有的IP地址都会被解锁。当重启Fail2ban，它会从 `/etc/log/secure` 等（即规则中指定的日志文件）中找到异常的IP地址列表，如果这些异常地址的发生时间仍然在禁止时间内，那么Fail2ban会重新将这些IP地址禁止。

Fail2ban可以缓解暴力密码攻击，但是并不能避免来自复杂的分布式暴力破解，攻击者通过使用成千上万个机器控制的IP地址来绕过Fail2ban的防御机制。

## 附1.10. 高可用性集群

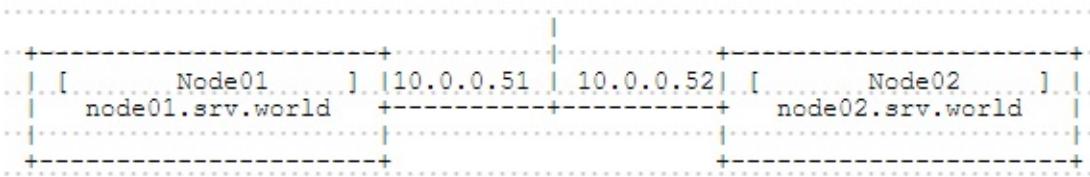
## 高可用性集群 (High-Availability Cluster)

## 附1.10.1. Pacemaker

Pacemaker 是一个集群管理器。

### 附1.10.1.1. 安裝 Pacemaker

本例基于以下环境（在这里配置基本的集群环境）：



如下所示，在所有节点上安装Pacemaker：

```
yum -y install pacemaker pcs
```

```
systemctl start pcsd  
systemctl enable pcسد  
`
```

```
passwd hacluster #设置集群管理员用户的密码
```

```
Changing password for user hacluster.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.
```

在节点上如下配置：

```
pcs cluster auth node01.srv.world node02.srv.world # 建立授权  
pcs cluster auth node01.srv.world node02.srv.world
```

```
Username: hacluster  
Password:  
node01.srv.world: Authorized  
node02.srv.world: Authorized
```

```
pcs cluster setup --name ha_cluster node01.srv.world  
node02.srv.world # 配置集群
```

```
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...  
node01.srv.world: Succeeded  
node02.srv.world: Succeeded
```

```
pcs cluster start --all # 启动集群服务
```

```
node02.srv.world: Starting Cluster...  
node01.srv.world: Starting Cluster...
```

```
pcs cluster enable --all # 启用集群
```

```
node01.srv.world: Cluster Enabled  
node02.srv.world: Cluster Enabled
```

```
pcs status cluster # 显示状态
```

```
Cluster Status:  
Last updated: Wed Jun 23 19:36:55 2015  
Last change: Wed Jun 23 19:36:47 2015  
Stack: corosync  
Current DC: node01.srv.world (1) - partition with quorum  
Version: 1.1.12-a14efad  
2 Nodes configured  
0 Resources configured
```

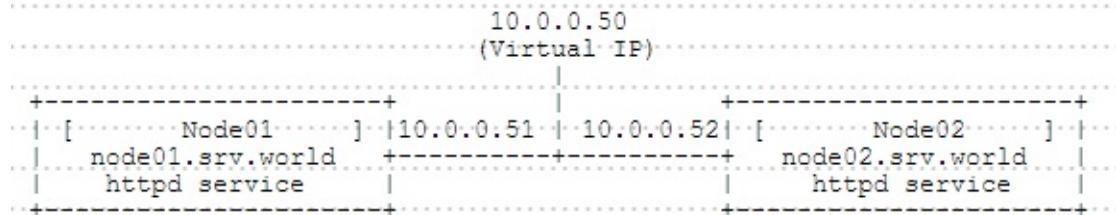
```
pcs status corosync
```

```
Membership information
-----
Nodeid      Votes  Name
1           1     node01.srv.world (local)
2           1     node02.srv.world
```

### 附1.10.1.2. 添加资源

将httpd资源添加到集群。

本例基于以下环境：



先参照第一节内容配置好集群的基本设置。

[安装Apache httpd（不用启动服务）。](#)

在所有节点上启用httpd server-status：

编辑 /etc/httpd/conf.d/server\_status.conf 文件：

```
ExtendedStatus On

<Location /server-status>
    SetHandler server-status
    Require local
</Location>
```

更改一些设置并设置虚拟IP地址：

```
pcs property set stonith-enabled=false # 禁用STONITH (Shoot The Other Node In The Head) 选项
```

```
pcs property set no-quorum-policy=ignore # 更改为“ignore”，两个节点的  
集群不需要
```

```
pcs property set default-resource-stickiness="INFINITY" # 禁用自动  
故障恢复
```

```
pcs property list # 显示设置
```

```
Cluster Properties:  
cluster-infrastructure: corosync  
cluster-name: ha_cluster  
dc-version: 1.1.12-a14efad  
default-resource-stickiness: INFINITY  
have-watchdog: false  
no-quorum-policy: ignore  
stonith-enabled: false
```

```
pcs resource create Virtual_IP ocf:heartbeat:IPAddr2 ip=10.0.0.50  
cidr_netmask=32 op monitor interval=30s # 设置虚拟IP地址
```

```
pcs status resources # 显示状态
```

```
Virtual_IP (ocf::heartbeat:IPAddr2): Started
```

添加httpd资源（可以在一个节点上设置）：

```
pcs resource create Web_Cluster \  
ocf:heartbeat:apache \  
configfile=/etc/httpd/conf/httpd.conf \  
statusurl="http://127.0.0.1/server-status" \  
op monitor interval=1min
```

```
pcs constraint colocation add Web_Cluster with Virtual_IP INFINITY  
# 设置Web_Cluster和Virtual_IP始终位于同一个节点上
```

```
pcs constraint order Virtual_IP then Web_Cluster # 设置启动顺序为  
Virtual_IP -> Web_Cluster
```

```
Adding Virtual_IP Web_Cluster (kind: Mandatory) (Options: first-action=start then-action=start)
```

```
pcs constraint # 显示状态
```

```
Location Constraints:  
Ordering Constraints:  
    start Virtual_IP then start Web_Cluster (kind:Mandatory)  
Colocation Constraints:  
    Web_Cluster with Virtual_IP (score:INFINITY)
```

访问虚拟IP地址以验证设置：



手动停止当前活动节点，确认资源正常切换到另一个节点：

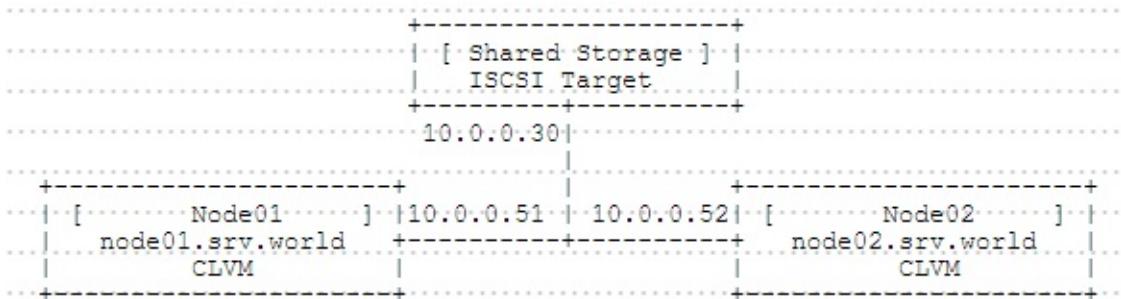
```
pcs cluster stop node01.srv.world
```

```
node01.srv.world: Stopping Cluster (pacemaker)...  
node01.srv.world: Stopping Cluster (corosync)...
```



### 附1.10.1.3. CLVM + GFS2

使用CLVM + GFS2配置存储集群。本例基于以下环境：



先参照第一节内容配置好集群的基本设置。

**创建iscsi共享存储**。需要两个共享存储设备，用于数据和fence设备。本例使用“iqn.2015-07.world.server:storage.target01”作为数据，使用“iqn.2015-07.world.server:fence.target00”作为fence设备。

在所有节点配置iSCSI启动器，没有在上面创建分区也可以。

在所有节点上安装所需的软件包：

```
yum -y install fence-agents-all lvm2-cluster gfs2-utils
```

```
lvmconf --enable-cluster
```

```
reboot
```

配置fence设备。可以在一个节点上设置。下例 /dev/sda 只是共享存储设备：

```
cat /proc/partitions # 确认fence设备磁盘（本例设置在“sda”）
```

```

major minor #blocks name
...
...
253      2    1048576 dm-2
8        0    1048576 sda
8        16   20971520 sdb

```

```
ll /dev/disk/by-id | grep sda # 确认磁盘的ID
```

```
lrwxrwxrwx 1 root root 9 Jul 10 11:44 scsi-36001405189b89389359  
4dfffb3a2cb3e9 -> ../../sda  
lrwxrwxrwx 1 root root 9 Jul 10 11:44 wwn-0x6001405189b89389359  
4dfffb3a2cb3e9 -> ../../sda
```

```
pcs stonith create scsi-shooter fence_scsi devices=/dev/disk/by-  
id/wwn-0x6001405189b893893594dfffb3a2cb3e9 meta provides=unfencing
```

```
pcs property set no-quorum-policy=freeze
```

```
pcs stonith show scsi-shooter
```

```
Resource: scsi-shooter (class=stonith type=fence_scsi)  
Attributes: devices=/dev/disk/by-id/wwn-0x6001405189b893893594  
dfffb3a2cb3e9  
Meta Attrs: provides=unfencing  
Operations: monitor interval=60s (scsi-shooter-monitor-interva  
l-60s)
```

添加所需资源。可以在一个节点上设置：

```
pcs resource create dlm ocf:pacemaker:controld op monitor  
interval=30s on-fail=fence clone interleave=true ordered=true
```

```
pcs resource create clvmd ocf:heartbeat:clvm op monitor  
interval=30s on-fail=fence clone interleave=true ordered=true
```

```
pcs constraint order start dlm-clone then clvmd-clone
```

```
Adding dlm-clone clvmd-clone (kind: Mandatory) (options: first-a  
ction=start then-action=start)
```

```
pcs constraint colocation add clvmd-clone with dlm-clone
```

```
pcs status resources
```

```
Clone Set: dlm-clone [dlm]  
Started: [ node01.srv.world node02.srv.world ]  
Clone Set: clvmd-clone [clvmd]  
Started: [ node01.srv.world node02.srv.world ]
```

使用GFS2在共享存储上创建卷并格式化。可以在一个节点上设置。本例在“**sdb**”上设置并在其上创建分区，并使用**fdisk**设置LVM类型：

```
pvcreate /dev/sdb1
```

```
Physical volume "/dev/sdb1" successfully created
```

```
vgcreate -cy vg_cluster /dev/sdb1 # 创建集群卷组
```

```
clustered volume group "vg_cluster" successfully created
```

```
lvcreate -l100%FREE -n lv_cluster vg_cluster
```

```
Logical volume "lv_cluster" created.
```

```
mkfs.gfs2 -p lock_dlm -t ha_cluster:gfs2 -j 2  
/dev/vg_cluster/lv_cluster
```

```
/dev/vg_cluster/lv_cluster is a symbolic link to /dev/dm-3  
This will destroy any data on /dev/dm-3  
Are you sure you want to proceed? [y/n] y  
Device: /dev/vg_cluster/lv_cluster  
Block size: 4096  
Device size: 0.99 GB (260096 blocks)  
Filesystem size: 0.99 GB (260092 blocks)  
Journals: 2  
Resource groups: 5  
Locking protocol: "lock_dlm"  
Lock table: "ha_cluster:gfs2"  
UUID: cdda1b15-8c57-67a1-481f-4ad3bbeb1b2f
```

将共享存储添加到集群资源。可以在一个节点上设置：

```
pcs resource create fs_gfs2 Filesystem \
device="/dev/vg_cluster/lv_cluster" directory="/mnt" fstype="gfs
2" \
options="noatime,nodiratime" op monitor interval=10s on-fail=fen
ce clone interleave=true
```

```
pcs resource show
```

```
Clone Set: dlm-clone [dlm]
    Started: [ node01.srv.world ]
    Stopped: [ node02.srv.world ]
Clone Set: clvmd-clone [clvmd]
    Started: [ node01.srv.world ]
    Stopped: [ node02.srv.world ]
Clone Set: fs_gfs2-clone [fs_gfs2]
    Started: [ node01.srv.world ]
```

```
pcs constraint order start clvmd-clone then fs_gfs2-clone
```

```
Adding clvmd-clone fs_gfs2-clone (kind: Mandatory) (Options: fir
st-action=start then-action=start)
```

```
pcs constraint colocation add fs_gfs2-clone with clvmd-clone
```

```
pcs constraint show
```

```
Location Constraints:
Ordering Constraints:
    start dlm-clone then start clvmd-clone (kind:Mandatory)
    start clvmd-clone then start fs_gfs2-clone (kind:Mandatory)
Colocation Constraints:
    clvmd-clone with dlm-clone (score:INFINITY)
    fs_gfs2-clone with clvmd-clone (score:INFINITY)
```

设置完成，确认GFS2文件系统挂载在活动节点上，并确认当前活动节点关闭时，GFS2挂载将转到另一节点：

```
df -hT
```

| Filesystem                        | Type     | Size  | Used | Avail | Use |
|-----------------------------------|----------|-------|------|-------|-----|
| % Mounted on                      |          |       |      |       |     |
| /dev/mapper/centos-root           | xfs      | 27G   | 1.1G | 26G   | 4%  |
| % /                               |          |       |      |       |     |
| devtmpfs                          | devtmpfs | 2.0G  | 0    | 2.0G  | 0%  |
| % /dev                            |          |       |      |       |     |
| tmpfs                             | tmpfs    | 2.0G  | 76M  | 1.9G  | 4%  |
| % /dev/shm                        |          |       |      |       |     |
| tmpfs                             | tmpfs    | 2.0G  | 8.4M | 2.0G  | 1%  |
| % /run                            |          |       |      |       |     |
| tmpfs                             | tmpfs    | 2.0G  | 0    | 2.0G  | 0%  |
| % /sys/fs/cgroup                  |          |       |      |       |     |
| /dev/vda1                         | xfs      | 497M  | 126M | 371M  | 26% |
| % /boot                           |          |       |      |       |     |
| /dev/mapper/vg_cluster-lv_cluster | gfs2     | 1016M | 259M | 758M  | 26% |
| % /mnt                            |          |       |      |       |     |

## 附1.11. 消息服务器

### 附1.11.1. RabbitMQ

RabbitMQ是由LShift提供的一个Advanced Message Queuing Protocol (AMQP)的开源实现，由以高性能、健壮以及可伸缩性出名的Erlang写成，因此也继承了这些优点。

#### 附1.11.1.1. 安装RabbitMQ

```
yum --enablerepo=epel -y install rabbitmq-server 从EPEL安装
```

```
systemctl start rabbitmq-server  
systemctl enable rabbitmq-server
```

firewalld防火墙规则，允许RabbitMQ端口：

```
firewall-cmd --add-port=5672/tcp --permanent  
firewall-cmd --reload
```

要使用RabbitMQ，先添加用户（默认情况下，只有guest用户存在，只能连接 localhost）：

```
rabbitmqctl add_user serverworld password # 格式为： rabbitmqctl  
add_user [user] [password]
```

```
Creating user "serverworld" ...  
...done.
```

```
rabbitmqctl list_users # 显示用户列表
```

```
Listing users ...  
guest [administrator]  
serverworld []  
...done.
```

```
rabbitmqctl change_password serverworld strongpassword # 如下更改用户密码
```

```
Changing password for user "serverworld" ...
...done.
```

```
rabbitmqctl set_user_tags serverworld administrator # 如下向用户授予管理员角色
```

```
Setting tags for user "serverworld" to [administrator] ...
...done.
```

```
rabbitmqctl delete_user serverworld # 如下删除用户
```

```
Deleting user "serverworld" ...
...done.
```

添加虚拟主机：

```
rabbitmqctl add_vhost /my_vhost # 格式为： rabbitmqctl add_vhost [vhost]
```

```
Creating vhost "/my_vhost" ...
...done.
```

```
rabbitmqctl list_vhosts # 显示虚拟主机列表
```

```
Listing vhosts ...
/
/my_vhost
...done.
```

```
rabbitmqctl delete_vhost /my_vhost # 如下删除虚拟主机
```

```
Deleting vhost "/my_vhost" ...
...done.
```

为虚拟主机授予用户权限，如下配置：

```
rabbitmqctl set_permissions -p /my_vhost serverworld ".*" ".*"  
".*" # 格式为： rabbitmqctl set_permissions [-p vhost] [user]  
[permission => (modify) (write) (read)]
```

```
Setting permissions for user "serverworld" in vhost "/my_vhost"  
...  
...done.
```

```
rabbitmqctl list_permissions -p /my_vhost # 显示虚拟主机权限
```

```
Listing permissions in vhost "/my_vhost" ...  
serverworld      .*      .*      .*  
...done.
```

```
rabbitmqctl list_user_permissions serverworld # 显示特定用户的权限
```

```
Listing permissions for user "serverworld" ...  
/my_vhost      .*      .*      .*  
...done.
```

```
rabbitmqctl clear_permissions -p /my_vhost serverworld # 如下删除特定用户的权限
```

```
Clearing permissions for user "serverworld" in vhost "/my_vhost"  
...  
...done.
```

### 附1.11.1.2. 在Python上使用

这是在Python上使用RabbitMQ的示例。

安装AMQP客户端库：

```
yum --enablerepo=epel -y install python2-pika # 从EPEL安装
```

在Python上发送消息的示例。例如，使用用户“serverworld”，虚拟主机“my\_vhost”连接RabbitMQ：

编辑 send\_msg.py 文件：

```
#!/usr/bin/env python

import pika

credentials = pika.PlainCredentials('serverworld', 'password')
connection = pika.BlockingConnection(pika.ConnectionParameters(
    'localhost',
    5672,
    '/my_vhost',
    credentials))

channel = connection.channel()
channel.queue_declare(queue='Hello_World')

channel.basic_publish(exchange='',
                      routing_key='Hello_World',
                      body='Hello RabbitMQ World!')

print(" [x] Sent 'Hello_World'")

connection.close()
```

python send\_msg.py

[x] Sent 'Hello\_World'

在Python上发送消息的示例。

编辑 receive\_msg.py 文件：

```
#!/usr/bin/env python

import signal
import pika

signal.signal(signal.SIGPIPE, signal.SIG_DFL)
signal.signal(signal.SIGINT, signal.SIG_DFL)

credentials = pika.PlainCredentials('serverworld', 'password')
connection = pika.BlockingConnection(pika.ConnectionParameters(
    'dlp.srv.world',
    5672,
    '/my_vhost',
    credentials))

channel = connection.channel()
channel.queue_declare(queue='Hello_World')

def callback(ch, method, properties, body):
    print(" [x] Received %r" % body)

channel.basic_consume(callback,
                      queue='Hello_World',
                      no_ack=True)

print(' [*] Waiting for messages. To exit press CTRL+C')
channel.start_consuming()
```

```
python receive_msg.py
```

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received 'Hello RabbitMQ World!'
```

### 附1.11.1.3. 在PHP上使用

这是在PHP上使用RabbitMQ的示例。

安装一些软件包：

```
yum --enablerepo=epel -y install composer php-bcmath #从EPEL安装
```

安装AMQP客户端库：

```
composer require php-amqplib/php-amqplib
```

```
Using version ^2.6 for php-amqplib/php-amqplib
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing php-amqplib/php-amqplib (v2.6.3)
  Downloading: 100%

Writing lock file
Generating autoload files
```

```
composer install
```

```
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Nothing to install or update
Generating autoload files
```

在PHP上发送消息的示例。例如，使用用户“serverworld”，虚拟主机“my\_vhost”连接RabbitMQ：

编辑 `send_msg.php` 文件：

```
<?php
require_once __DIR__ . '/vendor/autoload.php';

use PhpAmqpLib\Connection\AMQPStreamConnection;
use PhpAmqpLib\Message\AMQPMessage;

$connection = new AMQPStreamConnection('127.0.0.1', 5672, 'serve
rworld', 'password', '/my_vhost');

$channel = $connection->channel();
$channel->queue_declare('Hello_World', false, false, false, fals
e);

$msg = new AMQPMessage('Hello RabbitMQ World!');
$channel->basic_publish($msg, '', 'Hello_World');
echo "[x] Sent 'Hello_World'\n";

$channel->close();
$connection->close();
?>
```

php send\_msg.php

[x] Sent 'Hello\_World'

在PHP上发送消息的示例。

编辑 receive\_msg.php 文件：

```
<?php
require_once __DIR__ . '/vendor/autoload.php';
use PhpAmqpLib\Connection\AMQPStreamConnection;

$connection = new AMQPStreamConnection('127.0.0.1', 5672, 'serve
rworld', 'password', '/my_vhost');
$channel = $connection->channel();

$channel->queue_declare('Hello_World', false, false, false, fals
e);

echo ' [*] Waiting for messages. To exit press CTRL+C', "\n";

$callback = function($msg) {
    echo " [x] Received ", $msg->body, "\n";
};

$channel->basic_consume('Hello_World', '', false, true, false, f
alse, $callback);

while(count($channel->callbacks)) {
    $channel->wait();
}
?>
```

php receive\_msg.php

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received Hello RabbitMQ World!
```

### 附1.11.1.4. 在Ruby上使用

这是在Ruby上使用RabbitMQ的示例。

安装AMQP客户端库：

```
gem install bunny
```

```
Fetching: bunny-2.5.1.gem (100%)
Successfully installed bunny-2.5.1
Parsing documentation for bunny-2.5.1
Installing ri documentation for bunny-2.5.1
1 gem installed
```

在Ruby上发送消息的示例。例如，使用用户“serverworld”，虚拟主机“my\_vhost”连接RabbitMQ：

编辑 `send_msg.rb` 文件：

```
#!/usr/bin/env ruby

require "bunny"

connection = Bunny.new(
  :hostname => "127.0.0.1",
  :port => 5672,
  :vhost => "/my_vhost",
  :user => "serverworld",
  :pass => "password",
)
connection.start

channel = connection.create_channel

q = channel.queue("Hello_World")
channel.default_exchange.publish("Hello RabbitMQ World!", :routing_key => q.name)
puts "[x] Sent 'Hello RabbitMQ World!'"

connection.close
```

```
ruby send_msg.rb
```

```
[x] Sent 'Hello RabbitMQ World!'
```

在Ruby上发送消息的示例。

编辑 `receive_msg.rb` 文件：

```
#!/usr/bin/env ruby

require "bunny"

Signal.trap(:INT){
    puts "Exited from receiving queues."
    exit(0)
}

connection = Bunny.new(
    :hostname => "127.0.0.1",
    :port => 5672,
    :vhost => "/my_vhost",
    :user => "serverworld",
    :pass => "password",
)
connection.start

channel = connection.create_channel
q = channel.queue("Hello_World")

puts " [*] Waiting for messages in #{q.name}. To exit press CTRL+C"
q.subscribe(:block => true) do |delivery_info, properties, body|
    puts " [x] Received #{body}"

    delivery_info.consumer.cancel
end
```

```
ruby receive_msg.rb
```

```
[*] Waiting for messages in Hello_World. To exit press CTRL+C
[x] Received Hello RabbitMQ World!
```

### 附1.11.1.5. 使用Web界面

启用管理插件以使用基于Web的管理工具。

启用管理插件：

```
rabbitmq-plugins enable rabbitmq_management
```

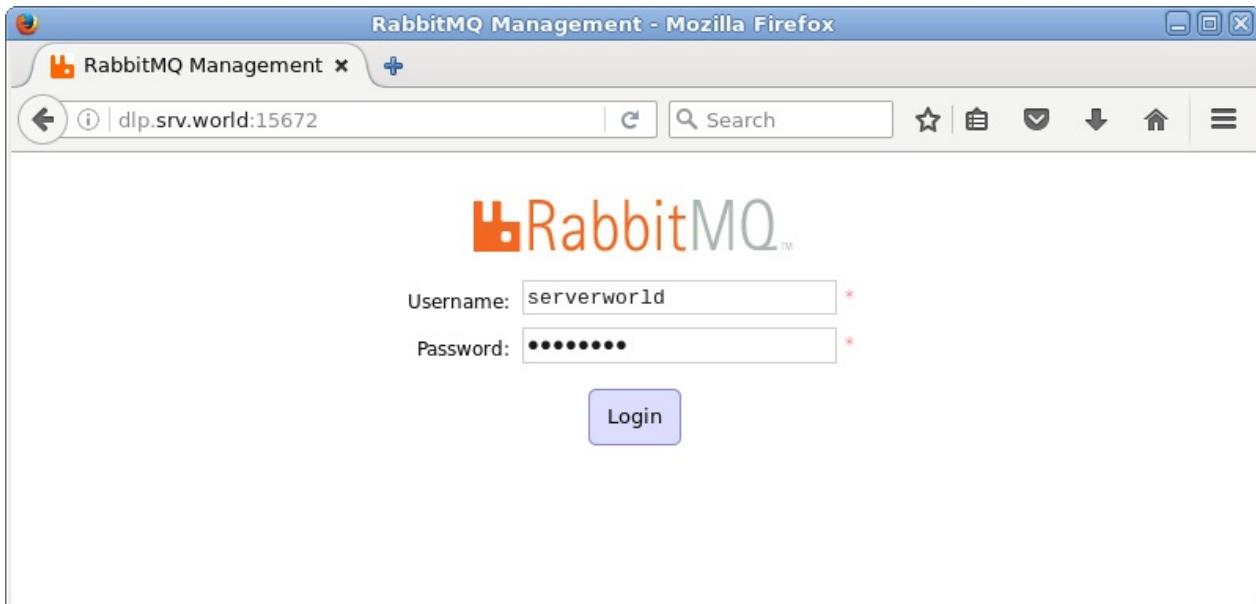
```
The following plugins have been enabled:  
mochiweb  
webmachine  
rabbitmq_web_dispatch  
amqp_client  
rabbitmq_management_agent  
rabbitmq_management  
Plugin configuration has changed. Restart RabbitMQ for changes to take effect.
```

```
systemctl restart rabbitmq-server
```

firewalld防火墙规则，添加Web界面端口：

```
firewall-cmd --add-port=15672/tcp --permanent  
firewall-cmd --reload
```

从客户端访问 `http://(RabbitMQ服务器的主机名或IP地址):15672/`，显示 RabbitMQ登录表单，并使用添加的管理员登录：



登录成功，可以在这里做管理操作：

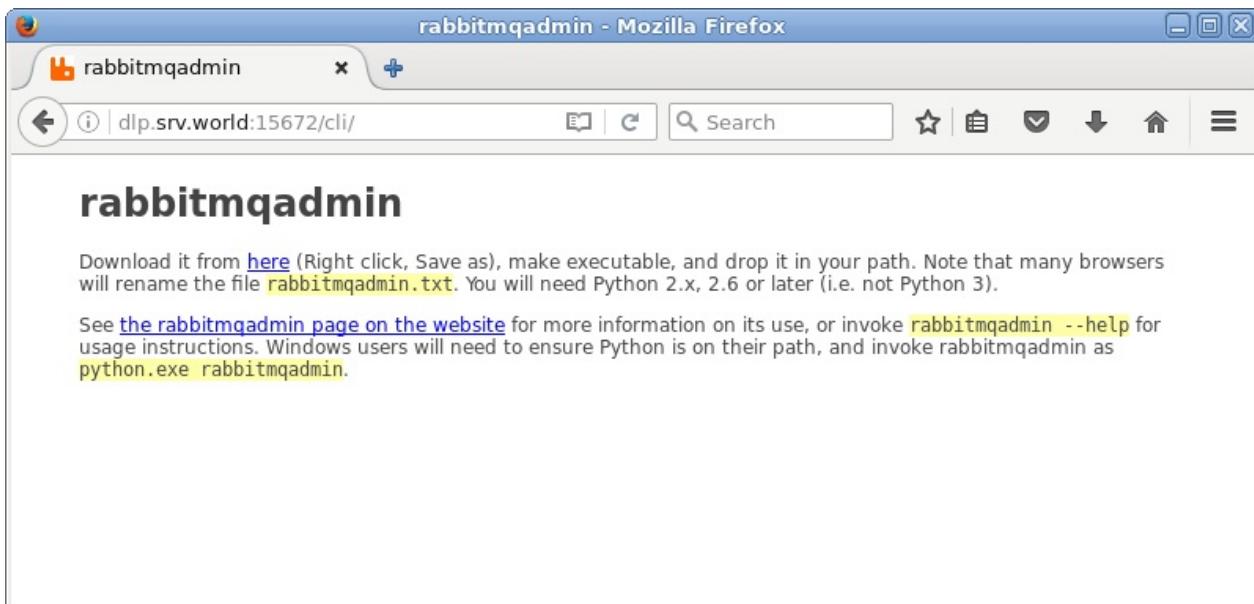
The screenshot shows the RabbitMQ Management interface in Mozilla Firefox. The title bar reads "RabbitMQ Management - Mozilla Firefox". The address bar shows the URL "dip.srv.world:15672/#/". The top right corner displays the user "serverworld", cluster "rabbit@dip.srv.world (change)", and version "RabbitMQ 3.3.5, Erlang R16B03-1". A "Log out" button is also present. Below the header, there is a navigation menu with tabs: Overview (highlighted), Connections, Channels, Exchanges, Queues, and Admin. The main content area is titled "Overview". It features a section titled "Totals" with a chart titled "Queued messages (chart: last minute) (?)" showing a constant value of 2.0. To the right of the chart, there is a legend: Ready (yellow square, 2 msg), Unacknowledged (blue square, 0 msg), and Total (red square, 2 msg). Below this is another chart titled "Message rates (chart: last minute) (?)" showing a constant value of 0.00/s. To the right of this chart, there is a legend: Publish (yellow square, 0.00/s). Further down, there is a section titled "Global counts (?)" with four buttons: "Connections: 0", "Channels: 0", "Exchanges: 15", and "Queues: 1". At the bottom, there is a section titled "Nodes" with a table header row containing columns for Name, File descriptors (?), Socket descriptors (?), Erlang processes, Memory, and Disk spac.

### 附1.11.1.6. 使用 rabbitmqadmin

可以使用 `rabbitmqadmin` 命令配置 RabbitMQ。

在Web界面下载“rabbitmqadmin”

登录到Web界面并转到 `http://(RabbitMQ服务器的主机名或IP地址):15672/cli`，显示以下屏幕，可以在这里下载“rabbitmqadmin”：



将“rabbitmqadmin”上传到RabbitMQ服务器并设置适当的权限，本例演示如下：

```
ll /usr/local/bin/rabbitmqadmin
```

```
-rwxr-xr-x. 1 root root 32406 Sep 5 19:12 /usr/local/bin/rabbitmqadmin
```

使用 `rabbitmqadmin` 命令进行基本操作：

```
rabbitmqadmin help subcommands # 显示子命令列表，只输入 help 选项
```

```
Usage
=====
rabbitmqadmin [options] subcommand
.....
....
```

```
rabbitmqadmin list users # 显示用户列表
```

| name        | password_hash                  | tags          |
|-------------|--------------------------------|---------------|
| guest       | xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | administrator |
| serverworld | xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx | administrator |

```
rabbitmqadmin list vhosts # 显示虚拟主机列表
```

| name      | messages | messages_ready | messages_unacknowledged |
|-----------|----------|----------------|-------------------------|
| recv_oct  | send_oct | tracing        |                         |
| /         |          |                |                         |
| /my_vhost | 2        | 2              | 0                       |
| 748       | 1004     | False          |                         |

```
rabbitmqadmin declare user name=centos password=password  
tags=administrator # 添加用户
```

user declared

```
rabbitmqadmin declare vhost name=/vhost01 # 添加虚拟主机
```

vhost declared

```
rabbitmqadmin declare permission vhost=/vhost01 user=centos  
configure=".*" write=".*" read=".*" # 授予权限
```

permission declared

```
rabbitmqadmin -V /vhost01 -u centos -p password declare queue  
name=my_queue01 # 添加队列
```

queue declared

```
rabbitmqadmin -V /vhost01 -u centos -p password publish  
routing_key=my_queue01 payload='Hello RabbitMQ World!'  
exchange=amq.default # 发送消息
```

Message published

```
rabbitmqadmin -V /vhost01 -u centos -p password get  
queue=my_queue01 requeue=false #接收消息
```

| routing_key   | exchange         | message_count | payload               |
|---------------|------------------|---------------|-----------------------|
| payload_bytes | payload_encoding | properties    | red..                 |
| my_queue01    |                  | 0             | Hello RabbitMQ World! |
| 21            | string           |               | False                 |

### 附1.11.1.7. 配置集群

配置RabbitMQ集群，本例使用两台RabbitMQ服务器。

按照第一节内容在所有节点安装并启动RabbitMQ服务器（如果防火墙运行，同样都打开端口）。

firewalld防火墙规则，允许集群端口：

```
firewall-cmd --add-port={4369/tcp,25672/tcp} --permanent  
firewall-cmd --reload
```

在节点上配置群集：

```
ssh dlp.srv.world 'cat /var/lib/rabbitmq/.erlang.cookie' >  
/var/lib/rabbitmq/.erlang.cookie # 在所有节点上放置相同的cookie  
  
systemctl restart rabbitmq-server  
  
rabbitmqctl stop_app # 停止程序
```

```
Stopping node rabbit@node01 ...  
...done.
```

```
rabbitmqctl reset # 重置
```

```
Resetting node rabbit@node01 ...
...done.
```

```
rabbitmqctl join_cluster rabbit@dlp # 加入群集（仅指定主机名，不使用
FQDN）
```

```
Clustering node rabbit@node01 with rabbit@dlp ...
...done.
```

```
rabbitmqctl start_app # 启动程序
```

```
Starting node rabbit@node01 ...
...done.
```

```
rabbitmqctl cluster_status # 显示状态
```

```
Cluster status of node rabbit@node01 ...
[{"nodes": [{"disc": [rabbit@dlp, rabbit@node01]}]}, {"running_nodes": [rabbit@dlp, rabbit@node01]}, {"cluster_name": <<"rabbit@dlp.srv.world">>}, {"partitions": []}]
...done.
```

配置队列同步设置，本例演示配置队列在所有节点上同步（节点之间有一些同步的模式，可参考[官方网站的详细信息](#)）：

```
rabbitmqadmin declare queue name=shared_queue # 添加队列进行同步
```

```
queue declared
```

```
rabbitmqctl set_policy ha-policy "shared_queue" '{"ha-
mode":"all"}' # 设置同步策略，格式为： rabbitmqctl set_policy [policy
name(any name you like)] [Queue] [Mode]
```

```
Setting policy "ha-policy" for pattern "shared_queue" to "{\"ha-mode\":\"all\"}" with priority "0" ...
...done.
```

```
rabbitmqadmin list queues name node policy slave_nodes state
synchronised_slave_nodes # 显示状态
```

| name                     | node          | policy    | slave_nodes   | state   |
|--------------------------|---------------|-----------|---------------|---------|
| synchronised_slave_nodes |               |           |               |         |
| shared_queue             | rabbit@dlp    | ha-policy | rabbit@node01 | running |
|                          | rabbit@node01 |           |               |         |

参照第五节，在所有节点上启用管理插件，可以看到每个节点的状态如下。

## 附1.11. 消息服务器

The screenshot shows the RabbitMQ Management interface running in Mozilla Firefox. The URL is `dip.srv.world:15672/#/`. The interface displays system status, listening ports, and web contexts.

**Nodes**

| Name          | File descriptors (?) | Socket descriptors (?) | Erlang processes         | Memory                       | Disk           |
|---------------|----------------------|------------------------|--------------------------|------------------------------|----------------|
| rabbit@dip    | 27<br>1024 available | 1<br>829 available     | 183<br>1048576 available | 39MB<br>1.5GB high watermark | 25<br>48MB low |
| rabbit@node01 | 21<br>1024 available | 1<br>829 available     | 175<br>1048576 available | 38MB<br>1.5GB high watermark | 25<br>48MB low |

**Ports and contexts**

Listening ports

| Protocol   | Node          | Bound to | Port  |
|------------|---------------|----------|-------|
| amqp       | rabbit@dip    | ::       | 5672  |
| amqp       | rabbit@node01 | ::       | 5672  |
| clustering | rabbit@dip    | ::       | 25672 |
| clustering | rabbit@node01 | ::       | 25672 |

Web contexts

| Context             | Node          | Bound to | Port  | SSL                   | Path |
|---------------------|---------------|----------|-------|-----------------------|------|
| RabbitMQ Management | rabbit@dip    | 0.0.0.0  | 15672 | <input type="radio"/> | /    |
| RabbitMQ Management | rabbit@node01 | 0.0.0.0  | 15672 | <input type="radio"/> | /    |

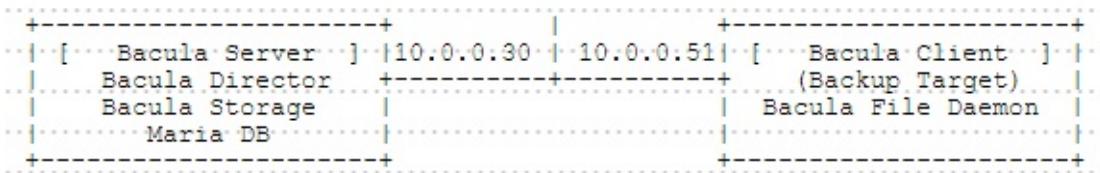
▶ Import / export definitions

## 附1.12. 备份管理工具

### 附1.12.1. Bacula

[Bacula](#)是一个集成备份工具，允许系统管理员来管理备份，恢复和核查在网络上的计算机数据。

本例基于以下环境：



#### 附1.12.1.1. 安装Bacula

在Bacula控制服务器上[安装MariaDB Server](#)。

在Bacula控制服务器上安装Director和Storage守护进程。

```
yum -y install bacula-director bacula-storage bacula-console
```

为Bacula添加数据库到MariaDB：

```
alternatives --config libbaccats.so # 将默认更改为MariaDB
```

```
There are 3 programs which provide 'libbaccats.so'.
```

| Selection | Command                             |
|-----------|-------------------------------------|
| 1         | /usr/lib64/libbaccats-mysql.so      |
| 2         | /usr/lib64/libbaccats-sqlite3.so    |
| *+ 3      | /usr/lib64/libbaccats-postgresql.so |

```
Enter to keep the current selection[+], or type selection number
: 1
```

添加数据库：

```
/usr/libexec/bacula/grant_mysql_privileges -p
```

```
Enter password: # MariaDB的root密码  
Privileges for user bacula granted on database bacula.
```

```
/usr/libexec/bacula/create_mysql_database -p
```

```
Enter password: # MariaDB的root密码  
Creation of Bacula MySQL tables succeeded.
```

```
mysql -u root -p
```

```
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 13  
Server version: 5.5.41-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
# 设置Bacula用户的密码  
MariaDB [(none)]> set password for bacula@'%'=password('password');  
Query OK, 0 rows affected (0.00 sec)  
MariaDB [(none)]> set password for bacula@'localhost'=password('password');  
Query OK, 0 rows affected (0.00 sec)  
# 确认  
MariaDB [(none)]> select user,host,password from mysql.user;  
+-----+-----+-----+  
| user | host | password |  
|  
+-----+-----+-----+  
| root | localhost | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
|  
| root | 127.0.0.1 | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
|
```

```
| root    | ::1          | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
|  
| bacula | %          | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
|  
| bacula | localhost | *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
|  
+-----+-----+-----+  
-+  
5 rows in set (0.00 sec)  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database          |  
+-----+  
| information_schema |  
| bacula            |  
| mysql              |  
| performance_schema |  
+-----+  
4 rows in set (0.00 sec)  
  
MariaDB [(none)]> show tables from bacula;  
+-----+  
| Tables_in_bacula |  
+-----+  
| BaseFiles         |  
| CDImages          |  
| Client             |  
| Counters           |  
| Device             |  
| File               |  
| FileSet            |  
| Filename           |  
| Job                |  
| JobHisto          |  
| JobMedia           |  
| Location            |  
| LocationLog        |  
| Log                |  
| Media               |  
| MediaType          |  
| Path               |
```

```
| PathHierarchy      |
| PathVisibility    |
| Pool              |
| RestoreObject     |
| Status            |
| Storage           |
| UnsavedFiles      |
| Version           |
+-----+
25 rows in set (0.01 sec)
```

```
MariaDB [(none)]> exit
Bye
```

### 附1.12.1.2. 配置Bacula服务器

配置Bacula Director：

编辑 `/etc/bacula/bacula-dir.conf` 文件：

```
Director {
  Name = bacula-dir
  DIRport = 9101
  QueryFile = "/etc/bacula/query.sql"
  WorkingDirectory = "/var/spool/bacula"
  PidDirectory = "/var/run"
  Maximum Concurrent Jobs = 1
  # 设置Director守护进程的密码
  Password = "password"
  Messages = Daemon
}

FileSet {
  Name = "Full Set"
  Include {
    Options {
      signature = MD5
      # 添加
      Compression = GZIP
    }
}
```

```

# 客户端主机上的备份目标目录
File = /home
}

Client {
    Name = bacula-fd
    # 备份目标主机名或IP地址
    Address = node01.srv.world
    FDPort = 9102
    Catalog = MyCatalog
    # 指定File守护进程的密码
    Password = "password"
    File Retention = 30 days
    Job Retention = 6 months
    AutoPrune = yes
}

Storage {
    Name = File
    # Storage守护进程的主机名或IP地址
    Address = dlp.srv.world
    SDPort = 9103
    # 指定Storage守护进程的密码
    Password = "password"
    Device = FileStorage
    Media Type = File
}

Catalog {
    Name = MyCatalog
    # Uncomment the following line if you want the dbi driver
    # dbdriver = "dbi:sqlite3"; dbaddress = 127.0.0.1; dbport =
    # MariaDB上Bacula用户的密码
    dbname = "bacula"; dbuser = "bacula"; dbpassword = "password"
}

Pool {
    Name = Default
    Pool Type = Backup
    Recycle = yes
    AutoPrune = yes
    # 保存卷的期限
}

```

```
Volume Retention = 180 days
# 为1个作业设置1个卷
Maximum Volume Jobs = 1
# 卷的标题，如果没有卷使用，自动创建新卷
Label Format = Vol-
}

# 注释以下内容
#Console {
# Name = bacula-mon
# Password = "@@MON_DIR_PASSWORD@@"
# CommandACL = status, .status
#}
```

编辑 `/etc/bacula/bconsole.conf` 文件:

```
Director {
    Name = bacula-dir
    DIRport = 9101
    # Director守护进程的主机名或IP地址
    address = dlp.srv.world
    # 指定Director守护进程的密码
    Password = "password"
}
```

编辑 `/usr/libexec/bacula/make_catalog_backup.pl` 文件:

```
# 添加
exec("HOME='$wd' mysqldump -f -u$args{db_user} -p$args{db_passwo
rd} --opt $args{db_name} > '$wd/$args{db_name}.sql'");
```

```
systemctl start bacula-dir
systemctl enable bacula-dir
```

配置Bacula存储:

编辑 `/etc/bacula/bacula-sd.conf` 文件:

```
Director {
    Name = bacula-dir
    # 设置Storage守护进程的密码
    Password = "password"
}

# 注释以下内容
#Director {
#    Name = bacula-mon
#    Password = "@@MON_SD_PASSWORD@@"
#    Monitor = yes
#}
```

```
systemctl start bacula-sd
systemctl enable bacula-sd
```

### 附1.12.1.3. 配置客户端

配置Bacula备份目标客户端。

在Bacula客户端上安装File组件：

```
yum -y install bacula-client bacula-console
```

配置Bacula File：

编辑 /etc/bacula/bacula-fd.conf 文件：

```
Director {
    Name = bacula-dir
    # 指定Director守护进程的密码
    Password = "password"
}

# 注释以下内容
#Director {
# Name = bacula-mon
# Password = "@@MON_FD_PASSWORD@@"
# Monitor = yes
#}
```

编辑 `/etc/bacula/bconsole.conf` 文件：

```
Director {
    Name = bacula-dir
    DIRport = 9101
    # Director守护进程的主机名或IP地址
    address = dlp.srv.world
    # 指定Director守护进程的密码
    Password = "password"
}
```

```
systemctl start bacula-fd
systemctl enable bacula-fd
```

### 附1.12.1.4. 备份操作

这是基本的备份操作。

可以在服务器和客户端上运行（本例在服务器上）。

运行备份：

```
bconsole
```

```
Connecting to Director dlp.srv.world:9101
1000 OK: bacula-dir Version: 5.2.13 (19 February 2013)
```

```
Enter a period to cancel a command.  
*label # 创建备份卷  
Automatically selected Catalog: MyCatalog  
Using Catalog "MyCatalog"  
Automatically selected Storage: File  
Enter new Volume name: Vol-20150721 # 任意名称  
Defined Pools:  
    1: Default  
    2: File  
    3: Scratch  
Select the Pool (1-3): 2 # 选择2作为示例  
Connecting to Storage daemon File at dlp.srv.world:9103 ...  
Sending label command for Volume "Vol-20150721" Slot 0 ...  
3000 OK label. VolBytes=207 DVD=0 Volume="Vol-20150721" Device="  
FileStorage" (/tmp)  
Catalog record for Volume "Vol-20150721", Slot 0 successfully c  
reated.  
Requesting to mount FileStorage ...  
3906 File device ""FileStorage" (/tmp)" is always mounted.  
*run # 运行备份  
A job name must be specified.  
The defined Job resources are:  
    1: BackupClient1  
    2: BackupCatalog  
    3: RestoreFiles  
Select Job resource (1-3): 1 # 选择作业（选择1作为示例）  
Run Backup job  
JobName: BackupClient1  
Level: Incremental  
Client: bacula-fd  
FileSet: Full Set  
Pool: File (From Job resource)  
Storage: File (From Job resource)  
When: 2015-07-21 23:47:30  
Priority: 10  
OK to run? (yes/mod/no): yes # 确认运行  
Job queued. JobId=1  
You have messages.  
* messages # 显示消息  
22-Jul 14:48 bacula-dir JobId 1: No prior Full backup Job record  
found.  
22-Jul 14:48 bacula-dir JobId 1: No prior or suitable Full backu
```

```

p found in catalog. Doing FULL backup.
22-Jul 14:48 bacula-dir JobId 1: Start Backup JobId 1, Job=Backup
pClient1.2015-07-21_23.48.20_04
22-Jul 14:48 bacula-dir JobId 1: Using Device "FileStorage" to write.
22-Jul 14:48 bacula-sd JobId 1: Wrote label to prelabeled Volume
"Vol-20150721" on device "FileStorage" (/tmp)
22-Jul 14:48 bacula-sd JobId 1: Elapsed time=00:00:01, Transfer
rate=839 Bytes/second
22-Jul 14:48 bacula-dir JobId 1: Bacula bacula-dir 5.2.13 (19Jan
13):
Build OS: x86_64-redhat-linux-gnu unknown unknown
n
JobId: 1
Job: BackupClient1.2015-07-21_23.48.20_04
Backup Level: Full (upgraded from Incremental)
Client: "bacula-fd" 5.2.13 (19Jan13) x86_64-redhat-linux-gnu,unknown,unknown
FileSet: "Full Set" 2015-07-21 23:48:20
Pool: "File" (From Job resource)
Catalog: "MyCatalog" (From Client resource)
Storage: "File" (From Job resource)
Scheduled time: 21-Jul-2015 23:47:30
Start time: 21-Jul-2015 23:48:23
End time: 21-Jul-2015 23:48:23
Elapsed time: 0 secs
Priority: 10
FD Files Written: 5
SD Files Written: 5
FD Bytes Written: 369 (369 B)
SD Bytes Written: 839 (839 B)
Rate: 0.0 KB/s
Software Compression: 16.5 %
VSS: no
Encryption: no
Accurate: no
Volume name(s): Vol-20150721
Volume Session Id: 1
Volume Session Time: 1437542407
Last Volume Bytes: 1,578 (1.578 KB)
Non-fatal FD errors: 0
SD Errors: 0

```

```
FD termination status:  OK
SD termination status:  OK
Termination:           Backup OK

21-Jul 23:48 bacula-dir JobId 1: Begin pruning Jobs older than 6
months .
21-Jul 23:48 bacula-dir JobId 1: No Jobs found to prune.
21-Jul 23:48 bacula-dir JobId 1: Begin pruning Files.
21-Jul 23:48 bacula-dir JobId 1: No Files found to prune.
21-Jul 23:48 bacula-dir JobId 1: End auto prune.

*exit # 退出
[root@dlp ~]# ll /tmp
total 4
drwx----- 3 root    root    16 Jul 22 11:33 systemd-private-L0E0q
u
drwx----- 3 root    root    16 Jul 22 11:43 systemd-private-OMX0w
g
-rw-r----- 1 bacula tape 1578 Jul 22 14:48 Vol-20150721
# 备份文件已保存
```

### 附1.12.1.5. 恢复操作

这是基本的恢复操作。

可以在服务器和客户端上运行（本例在客户端上）。

运行恢复：

bconsole

```
Connecting to Director dlp.srv.world:9101
1000 OK: bacula-dir Version: 5.2.13 (19 February 2013)
Enter a period to cancel a command.
*restore # 输入“restore”
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
```

First you select one or more JobIDs that contain files to be restored. You will be presented several methods of specifying the JobIDs. Then you will be allowed to select which files from those JobIDs are to be restored.

```
To select the JobIds, you have the following choices:  
1: List last 20 Jobs run  
2: List Jobs where a given File is saved  
3: Enter list of comma separated JobIDs to select  
4: Enter SQL list command  
5: Select the most recent backup for a client  
6: Select backup for a client before a specified time  
7: Enter a list of files to restore  
8: Enter a list of files to restore before a specified time  
9: Find the JobIDs of the most recent backup for a client  
10: Find the JobIDs for a backup for a client before a specified time  
11: Enter a list of directories to restore for found JobIDs  
12: Select full restore to a specified Job date  
13: Cancel  
Select item: (1-13): 5 # 选择5作为示例（最近的备份）  
Automatically selected Client: bacula-fd  
Automatically selected FileSet: Full Set  
+-----+-----+-----+-----+-----+-----+  
-----+  
| JobId | Level | JobFiles | JobBytes | StartTime | Vo  
lumeName |  
+-----+-----+-----+-----+-----+-----+  
-----+  
| 1 | F | 5 | 369 | 2015-07-21 23:48:23 | Vo  
l-20150721 |  
+-----+-----+-----+-----+-----+-----+  
-----+  
You have selected the following JobId: 1  
  
Building directory tree for JobId(s) 1 ...  
3 files inserted into the tree.  
  
You are now entering file selection mode where you add (mark) and  
remove (unmark) files to be restored. No files are initially added, unless  
you used the "all" keyword on the command line.  
Enter "done" to leave this mode.  
  
cwd is: /
```

```

$ ls  # 显示备份文件列表
home/
$ mark home  # 标记要恢复的目标文件
5 files marked.
$ lsmark  # 确认标记文件
*home/
*cent/
*.bash_logout
*.bash_profile
*.bashrc
$ done  # 运行恢复
Bootstrap records written to /var/spool/bacula/bacula-dir.restore.1.bsr

The job will require the following
      Volume(s)          Storage(s)          SD Device
(s)
=====
=====

      Vol-20150721        File           FileStorage

Volumes marked with "*" are online.

5 files selected to be restored.

Run Restore job
JobName:      RestoreFiles
Bootstrap:    /var/spool/bacula/bacula-dir.restore.1.bsr
Where:       /tmp/bacula-restores
Replace:     always
FileSet:      Full Set
Backup Client: bacula-fd
Restore Client: bacula-fd
Storage:     File
When:        2015-07-22 15:08:02
Catalog:     MyCatalog
Priority:    10
Plugin Options: *None*
OK to run? (yes/mod/no): yes

```

```

Job queued. JobId=2
* messages # 显示消息
22-Jul 15:09 bacula-dir JobId 2: Start Restore Job RestoreFiles.
2015-07-22_00.09.16_06
22-Jul 15:09 bacula-dir JobId 2: Using Device "FileStorage" to r
ead.
22-Jul 15:09 bacula-sd JobId 2: Ready to read from volume "Vol-2
0150721" on device "FileStorage" (/tmp).
22-Jul 15:09 bacula-sd JobId 2: Forward spacing Volume "Vol-2015
0721" to file:block 0:207.
22-Jul 15:09 bacula-sd JobId 2: End of Volume at file 0 on devic
e "FileStorage" (/tmp), Volume "Vol-20150721"
22-Jul 15:09 bacula-sd JobId 2: End of all volumes.
22-Jul 15:09 bacula-dir JobId 2: Bacula bacula-dir 5.2.13 (19Jan
13):
Build OS:           x86_64-redhat-linux-gnu unknown unknow
n
JobId:             2
Job:               RestoreFiles.2015-07-22_00.09.16_06
Restore Client:   bacula-fd
Start time:        22-Jul-2015 00:09:19
End time:          22-Jul-2015 00:09:19
Files Expected:   5
Files Restored:   5
Bytes Restored:   442
Rate:              0.0 KB/s
FD Errors:        0
FD termination status: OK
SD termination status: OK
Termination:      Restore OK

22-Jul 00:09 bacula-dir JobId 2: Begin pruning Jobs older than 6
months .
22-Jul 00:09 bacula-dir JobId 2: No Jobs found to prune.
22-Jul 00:09 bacula-dir JobId 2: Begin pruning Files.
22-Jul 00:09 bacula-dir JobId 2: No Files found to prune.
22-Jul 00:09 bacula-dir JobId 2: End auto prune.

*exit # 退出

```

```
ls -laR /tmp/bacula-restores
```

```
/tmp/bacula-restores:  
total 4  
drwxr-xr-x 3 root root 17 Jul 22 15:09 .  
drwxrwxrwt. 9 root root 4096 Jul 22 15:09 ..  
drwxr-xr-x 3 root root 17 Jul 9 2014 home  
  
/tmp/bacula-restores/home:  
total 0  
drwxr-xr-x 3 root root 17 Jul 9 2014 .  
drwxr-xr-x 3 root root 17 Jul 22 15:09 ..  
drwx----- 2 cent cent 59 Jul 9 2014 cent  
  
/tmp/bacula-restores/home/cent:  
total 12  
drwx----- 2 cent cent 59 Jul 9 2014 .  
drwxr-xr-x 3 root root 17 Jul 9 2014 ..  
-rw-r--r-- 1 cent cent 18 Jun 10 2014 .bash_logout  
-rw-r--r-- 1 cent cent 193 Jun 10 2014 .bash_profile  
-rw-r--r-- 1 cent cent 231 Jun 10 2014 .bashrc  
# 已恢复
```