CENTOS7改动很大，主要有：

1. 服务：

原来在：/etc/init.d/

现在：/usr/lib/systemd/

/usr/lib/systemd/system

系统启动的第一个进程（进程号1）是systemd，代替了原来的init

一些命令：

systemctl status|start|stop|restart|reload UNIT

systemctl enable|disable UNIT

systemctl mask UNIT 完全disable,使unit不被手动启动或开机启动

systemctl unmask UNIT

systemctl list-unit-files [--type=service] 相当于chkconfig --list (但多了一种状态：static，表示这个服务不能单独启动，而是由其它服务调用带动启动)

systemctl list-units [--type=service] [--all] 其中all表示也显示inactive的

systemctl --failed --type=service查看失败的服务

systemctl list-dependencies UNIT

systemctl reboot|poweroff

systemctl get-default

systemctl set-defalut graphical.target | muti-user.target

只查类型是service的Unit

systemctl --type=service

systemctl status sshd.service -l 查看更详细信息

systemctl is-active | is-enabled sshd.service

服务启动：

/usr/lib/systemd/system/ 默认单元文件安装目录

/run/systemd/system systemd systemd 单元运行时创建，这个目录优先于安装目录

/etc/systemd/system 系统管理员创建和管理的单元目录，优先级最高。

举个例子，演示在centos7上配置服务

1，安装一个a.service:

在/usr/lib/systemd/system/ ：system需要开机不登陆就能运行的程序存放地方

在/usr/lib/systemd/user/ 相反

这个a.service的内容：Unit为描述，Service为核心，Install为模式

[Unit]

Description=xiyoulibapi

After=network.target remote-fs.target nss-lookup.target

[Service]

Type=forking 后台运行

PIDFile=/node.js/pid

ExecStart=/usr/local/bin/forever start /node.js/xiyoulib/bin/www

ExecReload=/usr/local/bin/forever restart /node.js/xiyoulib/bin/www

ExecStop=/usr/local/bin/forever stop /node.js/xiyoulib/bin/www

PrivateTmp=true 分配独立临时空间

[Install]

WantedBy=multi-user.target

服务脚本按照上面编写完成后，以754的权限保存在/usr/lib/systemd/system目录下，这时就可以利用systemctl进行配置了

开机启动unit:

接着，只要使用systemctl enable xxxxx就可以将所编写的服务添加至开机启动即可。

2，建立软连接：

增加由/usr/lib/systemd/system/到/etc/systemd/system/multi-user.target.wants/下的软链接

也就是在/etc/systemd/system/multi-user.target.wants/增加这个服务的软链接

理解：系统启动后在优先级最高的/etc/systemd/system目录，然后根据启动运行模式如runnellever=3的方式登录，

就会在/multi-user.target.wants/下寻找脚本，这些脚本都是软链接的，去执行脚本，如我们这里会去执行first\_boot.service等,至于到底先执行哪个脚本，还不清楚。

开机不启动unit：

systemctl disable httpd.service

删除/etc/systemd/system/multi-user.target.wants下的软链接

查看开机是否启动:

systemctl is-enabled .service #查询服务是否开机启动

systemd查看开机自启动的程序

相当于chkconfig --list

ls /etc/systemd/system/multi-user.target.wants/

查看systemd单元加载及活动情况

systemctl

显示启动失败的单元

systemctl –failed

启动：

一、加电自检

二、选择启动设备

1.读取启动设备第一个扇区，读取引导程序

2.引导程序读取配置文件/boot/grub2/grub.cfg #不要编辑此文件

上面文件是grub2-mkconfig命令（用这两个文件/etc/default/grub /etc/grub.d）生成的

3.加载内核，并且以只读方式加载根分区

4.加载init ram disk

三、加载systemd进程

1.读取/etc/fstab

2.读取所选择的target,如multi-user.target

3.启动该级别的服务

4./etc/rc.d/rc.local开机脚本 (现在应该是在/usr/lib/systemd/下)

四、login

pvscan;vgscan;lvscan

lvchange -a y /dev/rhel/home

mknod /dev/rhel/home b 253 1

xfs\_repare /dev/rhel/home

systemd.unit=emergency.target

运行级别：

graphical.target 相当于原来的level 5 会调用multi-user.target

multi-user.target 相当于原来的level 3

rescue.target 相当于原来的level 1 单用户模式，不启动服务。但不同的是，现在需要密码才能进入。

powerof.target 相当于原来的level 0

reboot.target 相当于原来的level 6

emergency.target 救援，文件系统故障

rd.break

以前可以定义某些服务只在3级别自动启动，而5级别不启动

现在graphical.target 调用multi-user.target，所以没法这么做了。

systemctl get-default显示当前target

systemctl isolate multi-user.targe相当于以前的init 3命令

systemctl list-dependencies graphical.target |　grep target

systemctl list-units --type=targe --all

systemctl list-unit-files --type=targe --all

systemctl isolate multi-user.targe

ddif=/dev/zero of=/dev/sda bs=446 count=1

grub2-install /dev/sda

grub2-mkpasswd-pbkdf2 加密

samba客户端多用户

echo 'username=brain' >/root/smb-multiuser.txt

echo 'password=redhat' >/root/smb-multiuser.txt

vim /etc/fstab

//serverX/smbshare /mnt/multiuser cifs credentials=root/smb-multiuser.txt,multiuser,sec=ntlmssp 0

mount /mnt/multiuser

su -brain

软件方面的修改：

Yum

yum

yum从网络下载安装时，不仅有是Y和N选项，还有d选项，实现只下载，不安装。保存路径默认为:/var/cache/yum。

注意，只适用于从网络安装，如果是本地目录为yum源（baseurl=file://）则不生效

而以前的版本只可以实现安装时顺便缓存在本址，需要修改配置文件：

vim /etc/yum.conf

keepcache=1

yumclean all 可以清除所有下载的包

时间管理

timedatectl

timedatectl list-timezone

timedatectl set-timezone Asia/Shanghai

timedatectl set-time 9:00:00

以前/etc/ntp.conf

现在/etc/chrony.conf

服务：chronyd

chronyc sources -v 查看同步过程

日志管理

journalctl 可以查看指定条件的日志，比如按owner、时间段、产生的进程等

-x，--catelog 只查看/var/logmessages

-n指定行数，默认10行

例如：

journalctl --since 9:00:00 --until 9:30:00 \_\_SYSTEM\_UNIT=httpd.service

GPT分区

以前超过2T的硬盘用fdisk分区最多只能用2T，所以以前用parted:

parted /dev/sdb

mklabel gpt

mkpart 83 1 2621440

即使这样分区后，用fdisk -l /dev/sdb查看分区大小时，刚才创建的分区仍是2T

红帽/centos5：partprobe

红帽/centos 6：partx -a

红帽/contos 7:partprobe或 kpartx

现在不用parted，改用gdisk

gdisk /dev/sdc

2

n

1 最多可以分128个分区

-2G 负数表示从后往前分2G空间，数据放在硬盘的最外面，性能最好。负数表示从里往外分，即先分性能差的部分

格式化

mkfs.xfs /dev/sdc1

xfs\_info /dev/sdc1

xfs\_growfs 在线拉伸，不支持缩小

xfsrestore和 xfsdump备份还原

网络管理

网络接口命名,实际是被systemd-udevd改了

可以通过dmesg | grep eth查看

命名由三部分组成：

1.en是以太网，wl是WLAN, ww是WWAN

2.o是板载on board的，s是热插拔，p是PCI

3.数字，代表索引、ID或port

如果无法匹配，则用传统命名

nmcli con show

nmcli dev status

下面这部分还有疑问

centos7中，网卡配置文件需要加NM\_CONTROLLED="no"，否则：

改网卡配置文件（如改IP）再service network restart不生效

这种情况下要生效需要重启或执行以下命令：

nmcli con reload

nmcli con show

nmcli con down "..."

nmcli con up "..."

改计算机名

现在是/etc/hostname

以前是/etc/sysocnfig/network

防火墙

默认改成firewalld

systemctl mask iptables.service

systemctl start firewalled

用firewalled配置的内容还是可以通过iptables -L查看

9个zone:

1.trusted允许所有进来的流量

2.home拒绝所有进来的流量，除非是与出去的流量相关或者匹配ssh,mdsn,ipp-client,samba-client,dhcpv6-client

以前的iptables是通过这样实现匹配出去相关的流量：iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

3.internal 和home是一样的

4.word和home基本一样，但默认允许的程序只有：ssh,ipp-client,dhcpv6-client

5.public和home基本一样，但默认允许的程序只有：ssh,dhcpv6-client

是新加的网络接口的默认zone

6.external 和home类似，但默认允许的程序只有ssh.还可以作为masqueraded(SNAT)

7.dmz 和home类似，但默认允许的程序只有ssh

8.block和home类似,但没有默认允许的程序

9.drop和home类似，但不用ICMP errors包响应

每个zone里的组成：

service

port

masquerading (SNAT)

port forwarding (DNAT)

icmp filter

rich rules 优先级最高

interface

source

配置文件：/etc/firewalld

firewall-config 图形界面配置

firewall-cmd 命令行界面配置

firewall-cmd --get-zones

firewall-cmd --get-default-zone

firewall-cmd --set-default-zone=home

firewall-cmd --get-active-zones

firewall-cmd --source=<CIDR> [--zone=<zone>] 如不指定zone则改变当前的zone

firewall-cmd --remove-source=<CIDR>[--zone=<zone>]

firewall-cmd --add-interface=<interface>[--zone=<zone>]

firewall-cmd --change-interface=<interface>[--zone=<zone>]

--list-all 当前的zone的规则，可加[--zone=<zone>]指定其它zone

--list-all-zomes 所有zones

--add-service=<service> [--zone=<zone>]

--remove-service=<service> [--zone=<zone>]

--add-port=<port/protocol> [--zone=<zone>]

--remove-port=<port/protocol> [--zone=<zone>]

--reload

firewalld-cmd --permanent --add-service=samba

firewalld-cmd --permanent --remove-service=samba

runtime立刻生效,但下次重启不生效

不加--permanent则默认就是runtime的

permanent下次加载生效，可以通过firewalld-cmd --reload立刻生效

firewalld-cmd --permanent --add-port=80/tcp

rich rules优先级最高

firewall-cmd --permanent --new-zone=test注意，只能添加permanent 的zone

firewall-cmd --permanent --zone=classroom --add-rich-rule='rule family=ipv4 source address=192.168.0.1/32 reject'

firewall-cmd --add-rich-rule='rule service name=ftp limit value=2/m accept'每分钟只接受两个包

firewall-cmd --add-rich-rule='rule protocol value=esp drop'

规则位置：/usr/lib/firewalld/zones/

iscsi target

yum install targetcli

systemctl enable target;systemctl start target

以前服务是tgtd,现在是target.(客户端是iscsi)

firewalld-cmd --permanent --add-port=3260/tcp

firewalld-cmd reload

targetcli进入target命令行模式

/backstores/block/ create serverX.disk1 /dev/iSCSI\_vg/disk1\_lv

/iscsi create iqn.2014-06.com.example:serverX

/iscsi/iqn.2014-06.com.example:serverX/tpg1/acls/ create iqn.2014-06.com.example:desktopX

/iscsi/iqn.2014-06.com.example:serverX/tpg1/luns create /backstores/block/serverX.disk1

/iscsi/iqn.2014-06.com.example:serverX/tpg1portals create 172.25.1.11最后这个IP是自己的IP,即开启监听。不加的话不是监听所有，而是都不监听，可以设成0.0.0.0 监听所有