一. init是Linux系统操作中不可缺少的程序之一。

　　所谓的init进程，它是一个由内核启动的用户级进程。

　　内核自行启动（已经被载入内存，开始运行，并已初始化所有的设备驱动程序和数据结构等）之后，就通过启动一个用户级程序init的方式，完成引导进程。所以,init始终是第一个进程（其进程编号始终为1）。

　　内核会在过去曾使用过init的几个地方查找它，它的正确位置（对Linux系统来说）是/sbin/init。如果内核找不到init，它就会试着运行/bin/sh，如果运行失败，系统的启动也会失败。

二. init一共分为7个级别，这7个级别的所代表的含义如下

0：停机或者关机（千万不能将initdefault设置为0）

1：单用户模式，只root用户进行维护

2：多用户模式，不能使用NFS(Net File System)

3：完全多用户模式（标准的运行级别）

4：安全模式

5：图形化（即图形界面）

6：重启（千万不要把initdefault设置为6）

排序：方法一：# ls -lhSl 长格式显示，h human readable模式，大小单位为M,G等易读格式，S size按大小排序。

方法二：# du -h \* | sort -n

生成随即密码：date +%s | sha256sum | base64 | head -c 32 ; echo

openssl rand -base64 32

strings /dev/urandom | grep -o '[[:alnum:]]' | head -n 16 | tr -d '\n'; echo

</dev/urandom tr -dc \_A-Z-a-z-0-9 | head -c ${1:-16};echo;

tr -cd '[:alnum:]' < /dev/urandom | fold -w 30 | head -n1

< /dev/urandom tr -dc \_A-Z-a-z-0-9 | head -c 16

dd if=/dev/urandom bs=1 count=32 2>/dev/null | base64 -w 0 | rev | cut -b 2- | rev

</dev/urandom tr -dc '12345!@#$%qwertQWERTasdfgASDFGzxcvbZXCVB' | head -c16; echo ""

date | md5sum

uuidgen

函数：randpw(){ < /dev/urandom tr -dc \_A-Z-a-z-0-9 | head -c${1:-16};echo;}

rev 文件名：将每行字母反向输出

tac 文件名：由后往前倒序输出

获取分区uuid：blkid

虚拟机扩容：

LANG=en\_CN.UTF-8

growpart /dev/vda 1

xfs\_growfs /

1.源码nginx安装 2.平滑升级 3.添加server容器实现虚拟主机功能

4.HTTPS：http+ssl 5.用户认证 6.LNMP动静分离

7.地址重写 8.nginx反向代理（调度器）：调度功能---负载均衡，健康检查

Nginx常见问题处理：状态码

1.查看默认安装模块 2.自定义错误页面 3.查看并定义服务器状态信息

4.日志切割 5.优化Nginx并发量 6.限制单个ip的并发量

7.浏览器本地缓存静态数据 8.对页面进行压缩处理 9.服务器内存缓存

10.防止buffer溢出 11.优化Nginx数据包头缓存

12.修改软件名和版本号 13.拒绝非法的请求

tomcat：部署Tomcat服务器软件

配置tomcat虚拟主机 使用Tomcat部署虚修改跳转Context

配置Tomcat支持SSL加密网站 配置Tomca日志

改版本信息(错误提示) 降级启动. 删除默认的测试页面.apache(http)

一月份 二月份 三月份 四月份 五月份 六月份

January February march April May June

七月份 八月份 九月份 十月份 十一月份 十二月份

July August September October November December

Zabbix监控服务10051

监控的目的：同时监控包括吞吐量、反应时间、使用率等，提前发现系统的问题

Cacti监控系统 : 基于SNMP协议，强大的绘图能力

Nagios监控系统：基于agent监控，强大的状态检查与报警机制，可插入自己写的监控脚本

Zabbix监控系统：基于多种监控机制，支持分布式监控（LNMP----PHP）多进程软件

ELK:是一套解决方案 Elasticsearch：负责日志检索和存储

Logstash：负责日志的收集和分析、处理 Kibana：负责日志的可视化0

分布式（分片）优点：可拓展、容量大，节点性能高 缺点：故障率高（解决方法：原数据+副本数[少于节点数量]）

audit(审计):审计工具 审计（系统）的目的是基于事先配置的规则生成日志，记录可能发生在系统上的事件

nmap（扫描）：扫描分析工具

tcpdump抓包工具+其它（一、IP地址欺骗，二、dhcp欺骗：1.不能上网，错误的ip与网关 2.ip正确，网关是自己

三、篡改路由器或者主机，获取权限【字典+暴力破解密码】）

账户有效期：chage 锁定：passwd 伪装/隐藏系统版本：cat /etc/issue(本地) cat /etc/issue.net（远程）

创建用户默认基本信息：vim /etc/login.defs

额外属性noexec、noatime不再更新访问时间

iptables默认有4个表（区分大小写）：

nat表（地址转换表）、filter表（数据过滤表）raw表（状态跟踪表）、mangle表（包标记表）。

iptables的5个链chain（区分大小写）:iptables [-t 表名] 选项 [链名] [条件 条件..] [-j 目标操作]

INPUT链（入站规则）OUTPUT链（出站规则）FORWARD链（转发规则）

PREROUTING链（路由前规则）POSTROUTING链（路由后规则）

修改文件属性：chmod、chown、setfacl 、chattr +i/a

使用diff和patch工具打补丁 diff -Nura 原文档 新文档 > \*.patch patch -pN < 补丁包(\*.patch)

1.使用diff对比文件差异 2.使用diff生成补丁文件 3.使用patch命令为旧版本打补丁

sudo(超级执行)可以提权，需管理员提前授权

提高SSH服务安全: /etc/ssh/sshd\_config

SELinux安全防护：/etc/selinux/config 三大策略（布尔值、安全上下文，非默认端口开放）

RHEL:centos rhel suse 使用红帽软件包管理：.rpm 下载及卸载 yum

Debian:debian ubuntu 使用命令 apt-get --yum / dpkg --rpm .deb

kvm内核虚拟模块---qemu虚拟化仿真工具---libvirtd虚拟化管理的接口和工具--->提供用户管理工具virt virsh

virtio 是 KVM 虚拟环境下针对 I/O 虚拟化的最主要的一个通用框架

离线访问虚拟机：guestmount工具

云计算：一种服务模式，组建成计算资源共享池通过虚拟化，快速提供且管理少，

Iaas基础设施即服务（百度云盘）(私有云，公有云，混合云)

------>Paas平台即服务(淘宝，万达..)----->Saas软件即服务（苹果商店app）

A人工智能<---B大数据<---C云计算

openstack---一套IaaS解决方案，开源

七大组件：

Horizon---->基于web的管理接口，管理openstack各种服务，通过图形界面实现创建用户、管理网络、启动实例等操作

keystone--->支持多种身份认证模式和授权的集中身份管理服务（密码，令牌、AWS）、SSO认证服务

cinder----->为虚拟机管理存储卷服务，为运行在Nova中的实例提供永久块存储，通过快照进行数据备份

Neutron---->一种软件定义网络服务，创建网络、子网、路由器、管理浮动ip，虚拟交换机/路由器，创建VPN

swift------>提供了弹性可伸缩、高可用的分布式对象存储服务，适合存储大规模非结构化数据

glance----->虚拟机镜像注册的角色，允许用户直接存储拷贝服务器镜像，可以用户新建虚拟机的模板

nova------->在节点上管理虚拟机服务，分布式服务，与keystone交互实现认证，与glance交互实现镜像管理，

标准硬件上能够进行水平拓展，启动实例，若有则需要下载镜像

查看服务：systemctl list-unit-files | grep openstack

systemctl -a

sysctl -a （-w） | grep icmp 查看

sysctl -w {} 修改服务

容器：容器技术是应用程序封装和交付的核心技术,启动容器像启动一个进程一样快速

内核技术（linux2.6【3】)：Cgroups资源管理（限制）、NameSpace进程隔离、selinux安全

Docker是完整的一套容器管理系统，提供了一组命令，

6个命令空间(主机名、网络、用户、进程、根目录、信号向量--kill不影响真实机)能完成部分虚拟机的功能

镜像(后端盘)：docker中容器是基于镜像启动的，镜像采用分层设计，是启动容器的核心，把不同环境封装成镜像

使用快照的COW技术，确保底层数据不丢失

持久化存储：docker容器不保持任何数据，容器重要的数据挂载真实机目录或共享存储为卷（外部卷存储）

主机卷的映射：将真实机目录挂载到容器中提供持久化存储

GPG:确保数据的机密性：对称加密：加密/解密用同一个密钥DES/AES

非对称加密：加密/解密用不同的密钥（公钥、私钥）RSA/DSA

保护信息的完整性：信息摘要：基于输入的信息生成长度较短、位数固定的散列值 MD5(破解、与内容有关)/SHA

chroot 牢笼政策

脑裂--高可用集群中，不同节点联系异常或无法连接，都以为对方除了故障，争夺资源。同时读写会导致数据损坏

对策：1.增加冗余心跳线 2.启动磁盘锁 3.设置仲裁机制 主备机切换4.监控报警（zabbix）

容灾---数据备份---异地建立两套以上功能相同it系统

不是服务，没有端口

ipvsadm命令用法

使用命令增、删、改LVS集群规则

部署LVS-NAT集群

部署LVS-DR集群

设置Proxy代理服务器的VIP和DIP

proxy调度器安装软件并部署LVS-DR模式调度器

tcp/ip

三次握手：syn,syn ack,ack 四次断开：fin,ack, fin ack , ack

四次断开需要保证用户已经传输完数据

ls pwd cd mv cp rm cat vim less man scp grep

find -type 类型(f文本文件、d目录、l快捷方式)-name -user -group -size +|- -iname -maxdepth mtime +|-

find .. .. -exec 处理命令 {} 路径 \;

基本权限：rwx--->chmod 设置文档归属： chown [R] 主:组

附加（特殊）权限：s新增的子文档，自动继承父目录的所属组s让使用者具有文件属主的身份及部分权限t阻止用户滥用写入

acl访问策略setfacl -m

tar -zPcf/-jPcf/-JPcf /路径/\*.tar.gz/bz2/xz /路径/源文件 /路径/源文件

tar -xPf /路径/tar包名字 -C /路径/指定释放位置

清除干扰项：确保环境一致

配置主机名，ip地址、子网掩码，/etc/hosts映射文件、网关、yum源

客户端访问www.tarena.com---->地区DNS------->权威DNS--->CDN的DNS---->squid缓存节点--->返回数据

缓存服务器 别名-->转发 解析