系统管理索引

1. 系统信息管理

1.1 系统信息

- `uname -a` —显示内核版本和操作系统信息
- `hostnamectl` —查看和设置主机名
- `uptime` —查看系统运行时间
- `top` —实时查看系统进程
- `htop` —更友好的系统进程查看器

1.2 软件包管理

查询

- `dpkg -l` —列出已安装的软件包（Debian/Ubuntu）
- `rpm -qa` —列出已安装的软件包（RHEL/CentOS/Fedora）

Debian/Ubuntu（APT）

- `sudo apt update` —更新软件源
- `sudo apt upgrade` —升级已安装的软件包
- `sudo apt install <package>` —安装软件包
- `sudo apt remove <package>` —移除软件包
- `sudo apt search <package>` —搜索软件包

RHEL/CentOS/Fedora（DNF）

- `sudo dnf update` —更新软件包（Fedora 22 以后，RHEL 8/CentOS 8）
- `sudo dnf install <package>` —安装软件包
- `sudo dnf remove <package>` —移除软件包
- `sudo dnf search <package>` —搜索软件包
- `sudo dnf upgrade` —升级软件包

旧版（YUM for CentOS 7 / RHEL 7）

- `sudo yum update` —更新软件包
- `sudo yum install <package>` —安装软件包
- `sudo yum remove <package>` —移除软件包
- `sudo yum search <package>` —搜索软件包

1.3 系统日志

- `journalctl` —查看系统日志
- `dmesg` —查看内核日志

- `/var/log/` —查看其他日志文件，如 syslog、auth.log

2. 文件系统管理

2.1 查看磁盘信息

- `df -h` —查看磁盘空间使用情况
- `du -sh /path/to/dir` —查看目录的总大小
- `lsblk` —列出所有块设备
- `fdisk -l` —列出所有磁盘分区

2.2 创建和管理文件系统

- `sudo mkfs.ext4 /dev/sdX1` —创建 ext4 文件系统
- `sudo mkfs.xfs /dev/sdX1` —创建 xfs 文件系统
- `sudo mkfs.btrfs /dev/sdX1` —创建 btrfs 文件系统
- `sudo mount /dev/sdX1 /mnt` —临时挂载磁盘
- `sudo umount /mnt` —卸载挂载的磁盘

2.3 `/etc/fstab` 配置

- `sudo nano /etc/fstab` —编辑 fstab 文件，设置开机自动挂载，示例配置：/dev/sda1 /mnt/data ext4 defaults 0 0

2.4 磁盘加密

- `sudo cryptsetup luksFormat /dev/sdX1` —创建加密分区
- `sudo cryptsetup luksOpen /dev/sdX1 my_encrypted_disk` —打开加密分区

3. 网络管理

3.1 网络配置

查看网络接口信息

- `ip addr show` —显示网络接口的 IP 地址信息
- `ifconfig` —旧版命令，显示网络接口信息

配置静态 IP 地址

- **Debian/Ubuntu**：编辑 `/etc/network/interfaces` 文件 “‘bash iface eth0 inet static address 192.168.1.100 netmask 255.255.255.0 gateway 192.168.1.1 RHEL/CentOS/Fedora：使用 nmcli 命令

sudo nmcli con mod “System eth0” ipv4.addresses 192.168.1.100/24 sudo nmcli con mod “System eth0” ipv4.gateway 192.168.1.1 sudo nmcli con mod “System eth0” ipv4.method manual sudo nmcli con up “System eth0” 3.2 网络诊断 ping —测试与远程主机的连通性 traceroute —跟踪网络路由 netstat -tuln —查看系统中正在监听的端口 nc (Netcat) nc -zv —扫描主机某个端口范围是否开放 nc -l —监听某个端口，作为服务器端使用 nc —连接到远程主机指定端口，作为客户端使用 nc -v —显示连接详细信息 nmap nmap —扫描目标主机的开放端口 nmap -sP —Ping 扫描一个子网，列出存活的主机 nmap -sS —使用 SYN 扫描进行端口扫描（更隐蔽）nmap -O —检测目标主机的操作系统 Tcpdump tcpdump -i eth0 —在接口 eth0 上抓取所有流量 tcpdump -i eth0 port 80 —仅抓取 HTTP 流量（端口 80）tcpdump -w output.pcap —将捕获的数据包写入文件 tcpdump -r output.pcap —读取捕获的文件进行分析 3.3 DNS 配置 sudo nano /etc/resolv.conf —配置 DNS 服务器 nginx 复制 nameserver 8.8.8.8 nameserver 8.8.4.4 4. 安全管理 4.1 用户管理 sudo useradd username —创建新用户 sudo userdel username —删除用户 sudo passwd username —设置用户密码 sudo groupadd groupname —创建新用户组 sudo gpasswd -d username groupname —将用户从组中删除 4.2 权限管理 chmod 755 —修改文件权限 chown user:group —修改文件所有者和用户组 chgrp —修改文件的用户组 4.3 防火墙管理 查看当前防火墙状态 sudo ufw status （Debian/Ubuntu）sudo firewall-cmd –state （RHEL/CentOS/Fedora）启用防火墙 sudo ufw enable （Debian/Ubuntu）sudo systemctl start firewalld（RHEL/CentOS/Fedora）添加防火墙规则 sudo ufw allow 22/tcp （Debian/Ubuntu，允许 SSH）sudo firewall-cmd –zone=public –add-port=80/tcp –permanent（RHEL/CentOS/Fedora，允许 HTTP）4.4 SELinux 配置（仅适用于 RHEL/CentOS/Fedora）getenforce —查看 SELinux 当前状态（Enforcing、Permissive、Disabled）sudo setenforce 0 —临时禁用 SELinux sudo nano /etc/selinux/config —永久禁用 SELinux（修改 SELINUX=disabled）5. 服务和进程管理 5.1 管理服务 使用 systemctl 管理服务（适用于 RHEL/CentOS/Fedora 以及使用 systemd 的系统）

sudo systemctl start —启动服务 sudo systemctl stop —停止服务 sudo systemctl restart —重启服务 sudo systemctl enable —设置服务开机自启 sudo systemctl status —查看服务状态 5.2 管理进程 ps aux —查看当前系统上的所有进程 top —实时查看进程信息 kill —终止进程 killall —终止指定名称的所有进程 6. 备份和恢复 6.1 使用 tar 进行备份/恢复文件 tar -czvf backup.tar.gz /path/to/directory tar -xzvf backup.tar.gz -C /path/to/restore/ 6.2 使用 rsync 进行文件同步 同步文件：rsync -avz /path/to/source/ /path/to/destination/ 7. 系统配置 7.1 查看系统性能 free -h —查看内存使用情况 top —查看 CPU 和内存使用情况 iostat —查看磁盘 I/O 性能 vmstat —查看虚拟内存使用情况 7.2 配置系统调度策略 sudo sysctl -w vm.swappiness=10 —设置交换分区的优先级

防火墙（Firewall）

RHEL/CentOS 依赖包：firewalld(默认使用 firewalld)关闭 firewalld：sudo systemctl stop firewalld # 停止 firewalld 服务 sudo systemctl disable firewalld # 禁用 firewalld 服务开机启动 关闭 iptables（如果系统未使用 firewalld）： - sudo systemctl stop iptables # 停止 iptables 服务 sudo systemctl disable iptables # 禁用 iptables 服务开机启动 SELinux RHEL/CentOS 默认启用了 SELinux，关闭 SELinux 操作如下：- 临时禁用 SELinux：- sudo setenforce 0 - 永久禁用 SELinux：编辑 /etc/selinux/config

文件，将 SELINUX=enforcing 修改为 SELINUX=disabled，然后重启系统。LVM 操作 RHEL/CentOS 常用操作 - 安装 LVM 工具：sudo yum install lvm2 - 列出卷组（VG）：sudo vgs - 列出逻辑卷（LV）：sudo lvs - 创建卷组（VG）：sudo vgcreate /dev/sdX - 创建逻辑卷（LV）：sudo lvcreate -n -L - 扩展逻辑卷（LV）：sudo lvextend -L + /dev// sudo resize2fs /dev// - 删除逻辑卷（LV）：sudo lvremove /dev// - 删除卷组（VG）：sudo vgremove - 删除物理卷（PV）：sudo pvremove /dev/sdX 网络配置 RHEL/CentOS 7 之前的网卡配置 在 RHEL/CentOS 7 之前，网络配置通过编辑 /etc/sysconfig/network-scripts/ifcfg- 文件来管理。以下是一个典型的 ifcfg-eth0 配置文件示例：# /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0 # 网络接口名称 BOOTPROTO=static # 配置静态 IP ONBOOT=yes # 启动时启用该接口 IPADDR=192.168.1.100 # 静态 IP 地址 NETMASK=255.255.255.0 # 子网掩码 GATEWAY=192.168.1.1 # 默认网关 DNS1=8.8.8.8 # DNS 服务器 DNS2=8.8.4.4 # DNS 服务器 DEFROUTE=yes 配置示例解释：- DEVICE=eth0：指定接口名称为 eth0。- BOOTPROTO=static：使用静态 IP 配置。- ONBOOT=yes：表示接口在启动时启用。- IPADDR=192.168.1.100：配置静态 IP 地址。- NETMASK=255.255.255.0：配置子网掩码。- GATEWAY=192.168.1.1：指定默认网关。- DEFROUTE=yes：将该接口设置为默认路由接口。应用配置：sudo systemctl restart network

---

RHEL/CentOS 8 之后的网卡配置 从 RHEL/CentOS 8 开始，网络管理工具变得更加现代化，开始默认使用 NetworkManager 或 nmcli 工具来管理网络，并且仍然兼容原有网络配置文件/etc/sysconfig/network-scripts/ifcfg-eth0 文件，可以通过以下命令重新加载修改后的网络配置：- 使用 nmcli 重新加载接口：sudo nmcli connection reload - 或重启网络接口：sudo nmcli connection down eth0 && sudo nmcli connection up eth0 验证网络配置 使用以下命令查看网络接口的状态，确保 NetworkManager 正确应用了 ifcfg-eth0 配置文件中的设置：nmcli connection show eth0 或者：ip a show eth0

防火墙（Firewall）Debian/Ubuntu: - 依赖包：ufw（Uncomplicated Firewall）或 iptables - 关闭 UFW 防火墙（默认使用 UFW）：- sudo ufw disable # 禁用 UFW 防火墙 sudo systemctl disable ufw # 禁用 UFW 服务开机启动 - 关闭 iptables：- sudo systemctl stop iptables # 停止 iptables 服务 sudo systemctl disable iptables # 禁用 iptables 服务开机启动 AppArmor Debian/Ubuntu 默认没有启用 SELinux，而是使用 AppArmor 进行访问控制。如果你需要使用 SELinux，可以通过安装 selinux 包来启用：- 安装 SELinux：sudo apt update sudo apt install selinux-basics selinux-policy-default - 临时禁用 SELinux：sudo setenforce 0 - 永久禁用 SElinux：编辑 /etc/selinux/config，将 SELINUX=enforcing 修改为 SELINUX=disabled，然后重启系统。LVM 操作 Debian/Ubuntu: - LVM 工具依赖包：lvm2 sudo apt update sudo apt install lvm2 LVM 常用操作 - 列出卷组（VG）：sudo vgs - 列出逻辑卷（LV）：sudo lvs - 创建卷组（VG）：sudo vgcreate /dev/sdX - 创建逻辑卷（LV）：sudo lvcreate -n -L - 扩展逻辑卷（LV）：sudo lvextend -L + /dev// sudo resize2fs /dev// - 删除逻辑卷（LV）：sudo lvremove /dev// - 删除卷组（VG）：sudo vgremove - 删除物理卷（PV）：sudo pvremove /dev/sdX 网络配置 Ubuntu 20.04 之前 使用 interfaces 配置模板 在 Ubuntu 20.04 之前，网络配置文件通常为 /etc/network/interfaces。以下是一个传统的配置模板：/etc/network/interfacesLoopback interface auto lo iface lo

inet loopback

Wired interface (eth0) with static IP auto eth0 iface eth0 inet static address 192.168.1.100 设置静态 IP netmask 255.255.255.0 设置子网掩码 gateway 192.168.1.1 设置默认网关 dns-nameservers 8.8.8.8 8.8.4.4 设置 DNS 服务器 Wired interface (eth0) with DHCP

auto eth0 iface eth0 inet dhcp 使用 DHCP 获取 IP 地址 Wireless interface (wlan0) with static IP auto wlan0 iface wlan0 inet static address 192.168.1.101 netmask 255.255.255.0 gateway 192.168.1.1 wpa-ssid “YourNetworkName” Wi-Fi 网络名称 wpa-psk “YourNetworkPassword” Wi-Fi 密码 - auto eth0：表示接口 eth0 会在启动时自动启用。- iface eth0 inet static：设置静态 IP。- iface eth0 inet dhcp：使用 DHCP 自动获取 IP 地址。- dns-nameservers：配置 DNS 服务器。

Ubuntu 20.04+ 使用 netplan 配置模板 从 Ubuntu 20.04 开始，netplan 成为了默认的网络配置工具，配置文件通常位于 /etc/netplan/ 目录下。以下是一个典型的 50-cloud-init.yaml 配置模板：# /etc/netplan/50-cloud-init.yaml

network: version: 2 renderer: networkd # 使用 systemd-networkd 作为渲染器 ethernets: eth0: # 网卡名称 dhcp4: false # 禁用 DHCP 获取 IPv4 地址 addresses: - 192.168.1.100/24 # 设置静态 IP 和子网掩码 gateway4: 192.168.1.1 # 设置默认网关 nameservers: addresses: - 8.8.8.8 # 设置 DNS 服务器 - 8.8.4.4 配置示例解释：- version: 2：指定 netplan 配置的版本。- renderer: networkd：指定使用 systemd-networkd 渲染器（也可以使用 NetworkManager，取决于系统配置）。- eth0：网卡名称，可能根据你的系统不同而不同。- dhcp4: false：禁用 DHCP。- addresses：设置静态 IP 地址。- gateway4：设置默认网关。- nameservers：设置 DNS 服务器。在编辑完配置文件后，应用新的网络配置：sudo netplan apply

etcd 问题汇总 1. etcd 存储空间占用问题 1.1 问题背景 - 登录 K8S 集群，执行 kubectl 操作，响应特别慢 - 单节点 K8S 集群运行不稳定 - 查看系统负载 top - 13:49:55 up 61 days, 2:30, 2 users, load average: 26.59, 37.72, 30.98 Tasks: 741 total, 1 running, 740 sleeping, 0 stopped, 0 zombie %Cpu(s): 1.8 us, 1.6 sy, 0.0 ni, 94.2 id, 2.5 wa, 0.0 hi, 0.0 si, 0.0 st - 检查 etcd 存储 cd /opt/lib/etcd/ # du -hs *517M member [root@etcd]# cd member/ [root@k8s-master-node member]# du -hs* 90M snap 428M wal [root@k8s-master-node member]# cd wal/ [root@k8s-master-node wal]# du -hs * 62M 0000000000000067-00000000005f06a4.wal.broken 62M 00000000000009ad-0000000008b4afe0.wal 62M 00000000000009ae-0000000008b591d9.wal 62M 00000000000009af-0000000008b66cb4.wal 62M 00000000000009b0-0000000008b74d95.wal 62M 00000000000009b1-0000000008b830cb.wal 62M 1.tmp [root@k8s-master-node wal]# du -hs 428M . 1.2 异常结论 - 存储占用过高：总的 wal 文件占用了 428MB，这对于单节点系统来说是相对较大的数据量，尤其是当这些文件持续增长时，可能会导致存储压力增大，进而影响系统性能。.wal.broken 文件的存在：文件中出现.wal.broken 文件，表示系统在写入日志时发生了错误。此时需要排查系统磁盘、I/O 性能或其他异常。- 缺乏清理机制：没有看到自动归档或删除旧日志的机制，可能导致 wal 文件和临时文件不断堆积，造成磁盘空间不足。2 问题处理过程 2.1 查看 etcd 集群存储的状态 etcdctl –

endpoints=https://10.237.239.23:2379
–cacert=/etc/ssl/etcd/ssl/ca.pem
–cert=/etc/ssl/etcd/ssl/node-k8s-master-node.pem
–key=/etc/ssl/etcd/ssl/node-k8s-master-node-key.pem
endpoint status –write-out=table [图片] 每列的含义：暂时无法在飞书文档外展示此内容 2.2 查看当前版本号(非上表中版本)etcdctl –endpoints=https://10.237.239.23:2379
–cacert=/etc/ssl/etcd/ssl/ca.pem
–cert=/etc/ssl/etcd/ssl/node-k8s-master-node.pem
–key=/etc/ssl/etcd/ssl/node-k8s-master-node-key.pem
endpoint status –write-out= “json” | egrep -o ‘ “revision” :[0-9]’ | *egrep -o ’ [0-9].*’ 会出现一串数字比如 1234567 此为版本号 3.3 执行空间压缩 etcdctl –endpoints=https://10.237.239.23:2379
–cacert=/etc/ssl/etcd/ssl/ca.pem
–cert=/etc/ssl/etcd/ssl/node-k8s-master-node.pem
–key=/etc/ssl/etcd/ssl/node-k8s-master-node-key.pem
compact 1234567 （替换成节点的版本号）4.1 重新检查 etcd 集群存储的状态 etcdctl –endpoints=https://10.237.239.23:2379
–cacert=/etc/ssl/etcd/ssl/ca.pem
–cert=/etc/ssl/etcd/ssl/node-k8s-master-node.pem –key=/etc/ssl/etcd/ssl/node-k8s-master-node-key.pem endpoint status –write-out=table [图片] 清理完成后，DB SIZE 从原来的 94M -> 8.9M 4.2 检查系统状态 系统 IO 负载 立刻明显降低 top - 15:05:39 up 61 days, 3:46, 1 user, load average: 6.97, 7.47, 7.83 Tasks: 728 total, 3 running, 725 sleeping, 0 stopped, 0 zombie %Cpu(s): 20.3 us, 2.1 sy, 0.0 ni, 77.3 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st

1. etcd 备份脚本 这个脚本会使用 etcdctl 工具进行 etcd 集群的备份，确保你能够恢复集群的数据。备份脚本：etcd-backup.sh bash 复制编辑 #!/bin/bash 配置项 ETCDCTL_API=3 ETCD_ENDPOINTS=“https://10.237.239.23:2379” ETCD_CA_FILE=“/etc/ssl/etcd/ssl/ca.pem”ETCD_CERT_FILE=“/etc/ssl/etcd/ssl/node-deepflow-slave5-dongjinjiang.pem”ETCD_KEY_FILE=“/etc/ssl/etcd/ssl/node-deepflow-slave5-dongjinjiang-key.pem” BACKUP_DIR= “/backup/etcd” DATE=$(date + "{BACKUP_DIR}/etcd-backup-${DATE}.db” 创建备份目录(如果不存在)mkdir -p $BACKUP_DIR etcdctl−−endpoints =ETCD_ENDPOINTS –cacert=$ETCD_CA_FILE − −cert =ETCD_CERT_FILE –key=$ETCD_KEY_FILE snapshot save $BACKUP_FILE 输出备份文件路径 echo “Backup saved to $BACKUP_FILE” 说明：

- 使用 etcdctl snapshot save 命令来备份 etcd 数据。
- 备份文件将保存到指定的目录 (/backup/etcd)，文件名包含时间戳。

2. etcd 恢复脚本 此脚本会将 etcd 的数据恢复到指定的备份文件。恢复脚本：etcd-restore.sh #!/bin/bash 配置项 ETCDCTL_API=3 ETCD_ENDPOINTS= “https://10.237.239.23:2379” ETCD_CA_FILE= “/etc/ssl/etcd/ssl/ca.pem” ETCD_CERT_FILE=“/etc/ssl/etcd/ssl/node-deepflow-slave5-dongjinjiang.pem” ETCD_KEY_FILE= “/etc/ssl/etcd/ssl/node-deepflow-slave5-dongjinjiang-

key.pem” BACKUP_FILE= “/backup/etcd/etcd-backup-YYYY-MM-DD_HH-MM-SS.db” 需要替换成具体备份文件路径恢复 etct 数据 etcdctl –end-points=$ETCD_ENDPOINTS --cacert$=ETCD_CA_FILE –cert=$ETCD_CERT_FILE --key$=ETCD_KEY_FILE snapshot restore $BACKUP_FILE –data-dir=/opt/lib/etcd

重启 etcd 服务（具体服务名和路径可能不同）systemctl restart etcd

输出恢复完成信息 echo“Restore completed from $BACKUP_FILE” 说明：- etcdctl snapshot restore 命令用于恢复 etcd 数据。- 恢复后的数据会被存放到 /opt/lib/etcd 目录，服务会在恢复后重启。