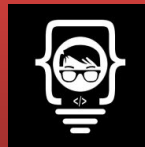




Ubuntu Server 从入门到精通

苑房弘 fanghong.yuan@163.com

第三章:包管理





Linux发行版操作系统

大量软件包的集合

- 内核、库文件、命令行shell、图形接口、其他应用软件包共同组成操作系统
- 包管理
 - 安装、删除、更新、配置修改
 - 不同发行版的主要不同之处
 - 库软件包提供其他软件依赖
 - 减少内存硬盘用量(所有软件依赖内核包)
 - 丢失或版本差异问题主要原因(一环出错, 全部出错)
 - Linux的库文件.so、windows下的库文件.dll
- Windows是微软的, 但Linux发行版有太多
 - 打造自己的软件仓库, 维护独立的包依赖关系是发行版包管理的最大任务





包管理

这是个麻烦事

- 软件包
 - RPM、DEB (Ubuntu 集成自 Debian Linux)
 - Tar、Gzip 等打包压缩发布
 - 内含 meta-data、预编译的二进制程序文件 (不是源码)、配置脚本等
- 包管理系统
 - Meta-data 提供追踪包中全部文件的能力
 - 维护已安装文件的数据库 (包名称、归属及辅助信息)
 - 解决包依赖关系
- 建议使用惟一的包管理系统





软件仓库

开放带来安全隐患

- 官方 Repositories
 - 打造封闭但有质量保证的软件包来源(苹果)
 - 慎用第三方软件仓库
- Dpkg 本地包管理器
 - 安装、删除、创建deb包
 - 不依赖软件仓库、不能自动检索和下载软件包
 - 不判断和解决依赖关系





DPKG包管理器

本地包管理工具

- `dpkg -l` 列出本地已安装的所有软件包
- `dpkg -L ufw` 列出包在本地安装的所有文件
- `dpkg -S /etc/ufw` 查找文件归属的软件包
- `dpkg -i n_v_a.deb` 安装包
 - `/var/cache/apt/archives`
- `dpkg -r name:arch` 卸载包 (-P: purge)
 - `dpkg --print-architecture` 架构 (amd64、i386)
 - `dpkg --print-foreign-architectures` 显示系统支持的其他架构包
 - `/var/lib/dpkg/arch`
 - `dpkg --remove-architecture`



....

APT包管理器

推荐使用的包管理器

- 依赖软件仓库
 - 安装、卸载、更新包括整个操作系统的所有软件包
 - 更新索引、更新包、自动解决依赖
- `sudo apt update` 更新索引
 - `/etc/apt/sources.list`、`/etc/apt/sources.list.d/`
- `sudo apt upgrade` 更新已安装的包(不增加和删除新包)
- `sudo apt dist-upgrade` 更新新包, 删除旧包(包含内核)
- `sudo apt install nmap` 安装包
- `sudo apt remove nmap --purge` 删除包及其配置文件
 - `/var/log/dpkg.log`

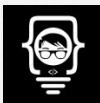


....

APT包管理器

推荐使用的包管理器

- `sudo apt search 'network mapper'` 通过关键词搜索包
- `sudo apt show nmap` 显示包信息
- `sudo apt autoremove` 删除不在需要的包
 - 慎用(某些文件可能是旧版内核的依赖包)
- `/var/cache/apt/archives` 下载deb包的存放路径
- `/var/lib/apt/`
- `apt download nmap` 只下载包
- `apt source nmap` 下载源码
 - `apt showsrc nmap`



....

自动更新

省时省力的好方法

- `sudo apt install unattended-upgrades` 安装无人职守升级包
- `/etc/apt/apt.conf.d/50unattended-upgrades` 配置文件

```
// pocket these get automatically pulled in.
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-upgrades
    // should also install from here by default.
    "${distro_id}ESM:${distro_codename}";
    "${distro_id}:${distro_codename}-updates";
    "${distro_id}:${distro_codename}-proposed";
    "${distro_id}:${distro_codename}-backports";
};

// List of packages to not update (regex are supported)
Unattended-Upgrade::Package-Blacklist {
    // "vim";
    // "libc6";
    // "libc6-dev";
    // "libc6-i686";
};
```





启用自动更新

省时省力的好方法

- `/etc/apt/apt.conf.d/10periodic`
 - `APT::Periodic::Update-Package-Lists "1";`
 - `APT::Periodic::Download-Upgradeable-Packages "1";`
 - `APT::Periodic::AutocleanInterval "30";`
 - `APT::Periodic::Unattended-Upgrade "7";`
- 重启服务
 - `sudo systemctl restart unattended-upgrades.service`
 - `sudo service unattended-upgrades.service restart`
- 日志目录
 - `/var/log/unattended-upgrades/`





无人职守更新通知

省时省力的好方法

- `/etc/apt/apt.conf.d/50unattended-upgrades`
 - Mail
- Apticron
 - `sudo apt install apticron`





APT更新配置

推荐使用官方更新源

- `/etc/apt/sources.list`
 - 注释掉 `cdrom` 源
 - 第一列: 二进制安装包或源码包 (`deb/ deb-src`)
 - 第二列: 库的URL地址
 - 第三列: 操作系统codename
 - 第四列: 库的内容构成、以及是否官方支持
 - `Main` 官方支持, 包含源码, 官方维护bug
 - `Restricted` 官方支持, 非开源 许可
 - `Universe` 社区支持
 - `Multiverse` 既不开源也不支持, 自己承担 风险, 通常不是安全更新





APT更新配置

推荐使用官方更新源

- 第三方库
 - 存在安全风险, 可能造成系统稳定性问题
 - 作为最后一种选择
 - 建议为每个第三方库创建独立的源.list文件
 - /etc/apt/sources.list.d/
 - 验证GnuPG Key
 - `wget -nv https://download.owncloud.org/download/repositories/stable/xUbuntu_16.04/Release.key -O Release.key`
 - `apt-key add Release.key`
 - `apt update`
 - `apt install <package name>`





Personal Package Archive (PPA)

推荐使用官方更新源

- PPA本质上是另外一种形式的apt软件库
 - 适用于没有自建软件库的发布者
 - PPA中的软件没有经过官方审核
- apt-add-repository ppa:ondrej/mariadb-10.0
 - 创建 /etc/apt/sources.list.d/ondrej-ubuntu-mariadb-10_0-xenial.list
 - <http://ppa.launchpad.net/ondrej/mariadb-10.0/ubuntu xenial main>
- PPA主页: <https://launchpad.net/ubuntu/>
- 删除PPA
 - 删除索引文件
 - 运行 apt-key 命令删除GnuPG Key





SNAP包管理

未来之星

- APT包管理器的缺点
 - 系统版本升级后应用软件库基本冻结
 - 老版本系统维持使用稳定版软件和库(旧版本)
 - 安全补丁除外
 - 为了维护一致的包和库的依赖关系无法安装最新版软件
- SNAP包管理操作独立于软件仓库
 - 可分发不属于官方库的软件版本
 - 软件安装使用不受操作系统包和库依赖关系的影响
 - Snap包内建与Linux发行版不兼容的库
 - Snap还年轻, APT仍然是这个时代的霸主





SNAP包管理

未来之星

- `sudo snap find nmap` #查找软件包
- `sudo snap install nmap` #安装软件包
- `sudo snap remove nmap` #删除软件包
- `sudo snap refresh nmap` #更新软件包
- 同时安装多个软件包(彼此独立)
- 作为APT的补充



...

Questions?

