

RPM 包安装

RPM 包 RPM (Red Hat Package Manager) 是一个开放的软件包管理系统, 相当于安卓手机的 apk 程序。

RPM 的功能: rpm 具有如下的几大功能, 详情如下: 安装: 将软件从 rpm 包中解压出来, 并且安装到硬盘 卸载: 将软件从系统中卸载掉 升级: 替换软件的旧版本; 查询: 查询软件包的信息; 验证: 检验系统中的软件与包中软件的区别

RPM 包的格式: RPM 包的名称有其特有的格式, 如: mysql-community-server-5.7.21-1.el7.x86_64.rpm name: mysql-community-server 是软件包的名称 version: 5.7.21-1 是软件的版本; type: x86_64 表示 64 位 intel x86 计算机平台上编译

使用 mysql 的 rpm 包安装举例:

由于在 linux 上一般安装有自带的 mysql 软件 (mariadb-libs)

yum remove mariadb-libs

rpm 查询命令:

rpm -qa : 查询目前安装在 linux 中的所有 rpm 软件

rpm -qa|grep 软件名 : 查询指定软件是否安装

rpm -ivh rpm 包 : 安装软件 -i 安装 -v 显示软件详细情况 -h 显示安装进度条

rpm -e 软件名称 : 卸载

注意事项: rpm 包安装软件的时候 有依赖关系 (即安装当前软件 需要提取安装另外一个软件) 也可以强制安装, 但不推荐 --nodeps rpm -ivh rpm 包 --nodeps

```
[root@192 opt]# rpm -ivh mysql-community-libs-compat-5.7.21-1.el7.x86_64.rpm
warning: mysql-community-libs-compat-5.7.21-1.el7.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
error: Failed dependencies:
    mysql-community-libs(x86-64) >= 5.7.9 is needed by mysql-community-libs-compat-5.7.21-1.el7.x86_64
[root@192 opt]#
```

依赖关系的软件没安装的提示

```
[root@192 opt]# rpm -ivh mysql-community-libs-compat-5.7.21-1.el7.x86_64.rpm --nodeps
warning: mysql-community-libs-compat-5.7.21-1.el7.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing...
Updating / installing...
##### [100%]
```

强制安装 — 不推荐

```
[root@192 opt]# rpm -qa|grep mysql
mysql-community-libs-compat-5.7.21-1.el7.x86_64
[root@192 opt]# rpm -e mysql-community-libs-compat-5.7.21-1.el7.x86_64
[root@192 opt]# rpm -qa|grep mysql
[root@192 opt]#
```

卸载

rpm -qi 软件名称: 显示软件的详情

rpm -ql 软件名称: 显示软件的安装目录

rpm -qf 文件名: 显示该文件属于哪个 rpm 软件的

linux 上 mysql 的安装部署:

mysql-community-common-5.7.21-1.el7.x86_64.rpm

mysql-community-libs-5.7.21-1.el7.x86_64.rpm

mysql-community-libs-compat-5.7.21-1.el7.x86_64.rpm

mysql-community-client-5.7.21-1.el7.x86_64.rpm

mysql-community-server-5.7.21-1.el7.x86_64.rpm

启动 mysql ==> service mysqld start

第一次启动之后, 会在 /var/log/mysqld.log 生成一个临时密码

```
2023-11-11T02:18:18.010395Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2023-11-11T02:18:18.163540Z 0 [Warning] InnoDB: New log files created, LSN=45790
2023-11-11T02:18:18.183827Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2023-11-11T02:18:18.238292Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 93ddecf7-8038-11ee-8ea8-000c290e9cdf.
2023-11-11T02:18:18.239087Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2023-11-11T02:18:18.239373Z 1 [Note] A temporary password is generated for root@localhost: 7cyVA(954Yif 密码
2023-11-11T02:18:20.576476Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2023-11-11T02:18:20.577780Z 0 [Note] /usr/sbin/mysqld (mysqld 5.7.21) starting as process
```

使用 mysql -uroot -p 加临时密码进行登录, 登录后再如下用命令进行修改密码

6.5. 降低密码复杂度限制: set global validate_password_policy=0;

6.6. 降低密码长度限制: set global validate_password_length=4;

6.7. 修改 root 用户的初始密码: alter user 'root'@'localhost' identified by '123456';

6.8 创建新用户: create user 'root'@'%' identified by '123456';

6.9 授权: GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';

6.10 刷新权限: flush privileges;

Linux 下 进行 java 项目部署:

前置条件: 安装 jdk + tomcat + mysql

第一步: 把代码包上传到 tomcat 安装目录的 webapps 目录下并解压

第二步: 把数据库脚本 进行执行, 让项目有后台数据库数据

一般就是在 nav 执行 dba 给的 .sql 后缀的文件

第三步: 修改代码中的数据库配置文件, 让其连上刚新建的数据库

备注: 数据库配置文件 是开发定义的, 后缀格式以及路径不一定。但通常是放在 apache-tomcat-7.0.76/webapps/项目名称/WEB-INF/classes (.xml .properties)

第四步: 启动 tomcat

yum 安装软件:

yum 本质上也是使用 rpm 包安装软件, 它安装软件的时候, 它会自动优先安装该软件所需要的依赖包, 再安装软件; 它不需要准备软件安装包, 可以通过配置 yum 源联网在线安装软件。

linux 上自带的 yum 源是国外的一个网站, 在国内使用非常慢, yum 源配置

本地 yum 源配置： 本地 yum 源指的是 linux 的安装光盘

第一步：在虚拟机上挂载光盘

- 1、虚拟机挂接 光盘映像文件 .iso
- 2、在 /mnt 目录下 新建文件夹 cdrom
- 3、挂载光盘命令 `mount /dev/cdrom /mnt/cdrom`

```
[root@192 mnt]# mount /dev/cdrom /mnt/cdrom
mount: /dev/sr0 is write-protected, mounting read-only
[root@192 mnt]#
```

卸载光盘 `umount /mnt/cdrom`

第二步：配置 yum 配置文件

yum 配置文件目录： /etc/yum.repos.d/

进入/etc/yum.repos.d/，把所有的 repo 后缀的配置文件进行删除或者移动到一个备份文件夹

```
[root@192 yum.repos.d]# vi local.repo
```

```
[root@192 yum.repos.d]# cat local.repo
[local]
name=local
baseurl=file:///mnt/cdrom
enabled=1
gpgcheck=0
```

第三步：加载新的 yum 配置文件

```
[root@192 yum.repos.d]# yum clean all
Loaded plugins: fastestmirror, langpacks
Cleaning repos: local
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space taken by
om disabled or removed repos
Cleaning up list of fastest mirrors
[root@192 yum.repos.d]# yum makecache
Loaded plugins: fastestmirror, langpacks
local | 3.6
(1/4): local/group_gz | 156
```

第四步：使用 yum 进行安装卸载软件

`yum install httpd` 安装 apache /

`yum install httpd -y` 过程中的提示都选择 y

`yum remove httpd` 卸载软件

`yum search httpd` 搜索软件

Yum 配置 远程 yum 源:

第一步：把本地 yum 源配置文件增加后缀让其失效

```
[root@192 yum.repos.d]# mv local.repo local.repo.bak
```

第二步：使用 `wget` 命令下载 阿里 yum 源（要求 linux 联网）

wget <http://mirrors.aliyun.com/repo/Centos-7.repo>

第三步：加载新的 yum 配置文件

```
yum clean all
```

```
yum makecache
```

Linux 源码编译安装：

以 nginx 软件举例

第一步：安装编译环境

```
yum install -y pcre-devel
```

```
Yum install -y zlib-devel
```

第二步：下载 nginx 源码包 放在 opt 目录下

wget <https://nginx.org/download/nginx-1.12.0.tar.gz>

第三步：解压 安装包

第四步：安装

```
cd nginx-1.12.0/
```

```
./configure --prefix=/opt/nginx112
```

```
make
```

```
make install
```

nginx：反向代理服务器

作用：

- 1、不会轻易暴露 web 应用服务器端口，导致服务器被攻击，增加安全性
- 2、保障业务系统能 7*24 小时无故障运行
- 3、增加运维的稳定性（升级、线上问题跟进）

启动 和 关闭 nginx

```
[root@localhost sbin]# pwd
/opt/nginx112/sbin
[root@localhost sbin]# ./nginx
```

```
[root@localhost sbin]# ./nginx -s stop
[root@localhost sbin]# ./nginx -s reload
```

nginx 关闭命令
nginx 平滑重启命令

平滑重启命令：在 nginx 启动后，修改配置文件，然后使用 reload 可以让 nginx 加载新的配置生效

Nginx：端口映射 保障 web 应用服务器（tomcat/apache 等）端口不被暴露 nginx 是一个高性能的轻量级的 WEB/HTTP 和反向代理服务器。

第一步：把 nginx 配置文件进行备份

```
[root@localhost bin]# cd /opt/nginx112/conf/
[root@localhost conf]# cp nginx.conf nginx.conf.bak
```

第二步：修改配置文件

vi nginx.conf

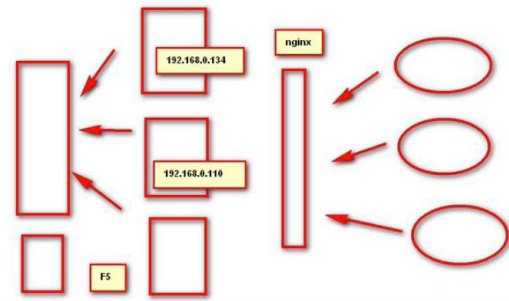
```
43         location / {
44             root    html;
45             index   index.html index.htm;
46             proxy_pass http://127.0.0.1:8080;
47         }
```

第三步：创建 nginx 日志文件夹 `mkdir /var/log/nginx`

第四步：重启 nginx 服务器 `./nginx -s reload`

此时，访问项目的时候，不需要加 8080 端口

淘宝等网站 7*24 无故障运行的--采用的是负载均衡技术 nginx 可以实现该技术



cd /opt/nginx112/conf/

vi nginx.conf

```
33 #gzip on;
34 upstream tong_server {
35     server 192.168.0.134:8080 weight=1;
36     server 192.168.0.167:8080 weight=1;
37 }
38
39 server {
40     listen      80;
41     server_name localhost;
42
43     #charset koi8-r;
44
45     #access_log logs/host.access.log main;
46
47     location / {
48         root    html;
49         index   index.html index.htm;
50         proxy_pass http://tong_server;
```

配置完成后，也需要 `./nginx -s reload`

shell 脚本编程：

shell 既是一种命令语言，又是一种程序设计语言。作为命令语言，它交互式地解释和执行用户输入的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高阶语言中才具有的控制结构，包括循环和分支。

shell 语言类型：bash（最常用） csh tcsh（C 语言） ksh

History：查看历史输入命令

Alias：设置别名命令


```
[root@192 sbin]# la
bash: la: command not found...
[root@192 sbin]# ls -a
.  ..  nginx
[root@192 sbin]# alias la='ls -a'
[root@192 sbin]# la
.  ..  nginx
```

```
[root@192 sbin]# unalias la 删除别名
[root@192 sbin]# alias 查看目前已有的别名
```

常用

```
[root@192 sbin]# alias cd3='cd ../../..'
[root@192 sbin]# alias cd4='cd ../../../../..'
```

shell 脚本基础:

第一个 shell 脚本, vi demo_01.sh ,必须是 sh 后缀

```
#!/bin/bash
echo "hello,world!!!"
```

使用 ./文件名 或 sh 文件名 执行

```
[root@192 liu_shell]# ls
[root@192 liu_shell]# vi demo_01.sh
[root@192 liu_shell]# chmod +x demo_01.sh
[root@192 liu_shell]# ./demo_01.sh
hello,world!!!
[root@192 liu_shell]# sh demo_01.sh
hello,world!!!
```

第二个 shell 脚本

shell 中的变量: 系统自带变量, 如 HOME PWD 等 用户自定义变量 如下:

\$变量名 引用变量

```
#!/bin/bash
str_a="hello,world"
echo $str_a
echo $HOME
echo $PWD
```

第三个 shell 脚本 可以把命令的结果传入到变量, 有如下两种方法:

```
#!/bin/bash
path_01=`pwd`
path_02=$(pwd)
echo $path_01
echo $path_02
```

第四个 shell 脚本: 位置参数 在执行 shell 命令的时候, 可以在后面加参数值 比如:

```
root@192:/tmp/liu_shell root@192:/opt/apache-tomcat-9.0.0.M26/bin
#!/bin/bash
echo 1=$1 2=$2 3=$3
echo 所有参数1:$*
echo 所有参数2:$@
echo 参数数量:$#
```

第五个 shell 脚本： 输入语句 read 输入语句

```
[root@192 liu_shell]# cat demo_05.sh
#!/bin/bash
echo 请输入用户名
read name
echo 请输入密码
read password
echo "您输入的用户名是 $name , 密码是 $password"
```

第六个 shell 脚本： 运算符

```
root@192:/tmp/liu_shell root@192:/opt/apache-tomcat-9.0.0.M26/bin
#!/bin/bash
#计算的写法
result01=$((2+3)*4)
result02=$((2+3)*4)
tmp_result=`expr 2 + 3`
result03=`expr $tmp_result \* 4`
let result04=(2+3)*4
echo $result01
echo $result02
echo $result03
echo $result04
```