

《Shell 基础脚本操作》

学生实验手册

目录

第 1 章 实验准备	2
1.1 实验概述	2
1.2 实验目标	2
1.3 学前建议	2
1.4 背景知识	2
1.5 申请实验资源	2
第 2 章 使用 Shell 脚本安装 Git 和 SVN 实验步骤	3
2.1 远程登录 ECS 服务器	3
2.2 使用 vim 工具写一个安装 Git 或 SVN 的脚本	3
2.3 使用脚本安装 Git	6
2.4 使用脚本安装 SVN	6
第 3 章 Git 的使用实验步骤	7
3.1 使用 Git 创建本地仓库	7
3.2 Git 克隆远程代码	9

第 1 章 实验准备

1.1 实验概述

本实验会自动创建一台 ECS 实例。首先，远程登陆 ECS 实例，使用 Shell 脚本安装 Git 和 SVN，并创建一个 Git 仓库，和拉取远程代码到本地。

1.2 实验目标

完成此实验后，可以掌握的能力有：

1. vi/vim 工具的使用
2. Shell 脚本数组，参数传递等知识点的应用
3. Git 和 SVN 的安装
4. Git 的使用

1.3 学前建议

了解 Linux 的基本操作命令，比如 yum 安装卸载软件

1.4 背景知识

Shell 是一个用 C 语言编写的程序，它是用户使用 Linux 的桥梁。Shell 既是一种命令语言，又是一种程序设计语言。

Git 是一个开源的分布式版本控制系统，是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件，用于敏捷高效地处理任何或小或大的项目。

SVN 是 subversion 的缩写，是一个开放源代码的版本控制系统

1.5 申请实验资源

点击左侧申请实验资源，实验资源会提供实验所需要的账号信息。

第 2 章 使用 Shell 脚本安装 Git 和 SVN 实验步骤

2.1 远程登录 ECS 服务器

本小节主要内容：在当前计算机上访问远程 ECS 服务器，其操作系统为 Linux。

1. 远程登录到 ECS 服务器。请使用 实验资源 提供的 ECS 服务器的 弹性 IP，用户名 和 密码。

```
zhuyun — root@iZuf6a50pk1lwwt6h8q1viZ:~ — ssh root@101.132.185.57 — 80x24
Last login: Thu Nov  9 09:17:24 on console
[zhuyundeMacBook-Air:~ zhuyun$ ssh root@101.132.185.57]
The authenticity of host '101.132.185.57 (101.132.185.57)' can't be established.
RSA key fingerprint is SHA256:Ok84/dQyYNyyepEwh2Kr+RXvOPmxwa78haLAUPFedzE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '101.132.185.57' (RSA) to the list of known hosts.
[root@101.132.185.57's password:
Last login: Tue Feb  2 10:49:14 2016 from 101.231. ....

Welcome to aliyun Elastic Compute Service!

[root@iZuf6a50pk1lwwt6h8q1viZ ~]#
```

2.2 使用 vim 工具写一个安装 Git 或 SVN 的脚本

1. 进入到 ECS 服务器中，在命令行输入 vim install_git_svn.sh

```
#vim install_git_svn.sh
```

2. 按 i 进入输入模式，输入如下内容

```
#!/bin/bash

#定义 Install_git 函数
Install_git(){

#打印进度
printf "准备安装 Git\n"

#安装依赖包
install_array=(curl-devel expat-devel gettext-devel openssl-devel zlib-devel)

yum remove git-core -y >/dev/null 2>&1

yum install ${install_array[*]} -y >/dev/null 2>&1

yum install git-core -y >/dev/null 2>&1

}

#定义 Install_svn 函数
```

```

Install_svn(){
#输出安装进度

printf "准备安装 SVN\n"

yum remove subversion -y >/dev/null 2>&1
yum install subversion -y >/dev/null 2>&1
}

#参数传入
case $1 in
    Git)
#调用函数
        Install_git
        echo "Git 安装完成！"
        ;;
    SVN)
#调用函数
        Install_svn
        echo "SVN 安装完成！"
        ;;
    *)
        echo "请输入正确的参数 Git 或 SVN。"
        ;;
esac

```

2.输入完成后，按 ESC 键，进入命令模式，输入:wq（英文冒号），按回车键，退出编辑器。

```

}

Install_svn(){
printf "准备安装SVN\n"
yum install subversion -y
}

case $1 in
    Git)
        Install_git
        echo "Git安装完成！"
        ;;
    SVN)
        Install_svn
        echo "SVN安装完成！"
        ;;
    *)
        echo "请输入正确的参数Git或SVN。"
        ;;
esac
:wq

```

输入:wq(英文冒号)

3.在控制台输入 ll 命令来查看该脚本

```

# ll
总用量 4
-rw-r--r-- 1 root root 491 5月  6 21:54 install_git_svn.sh

```

```

[root@localhost ~]# ll
总用量 4
-rw-r--r-- 1 root root 491 5月  6 21:54 install_git_svn.sh

```

4.给脚本添加执行权限

```
#chmod +x install_git_svn.sh
```

```
# ll
```

```
总用量 4
```

```
-rwxr-xr-x 1 root root 491 5月  6 21:54 install_git_svn.sh
```

```
[root@localhost ~]# chmod +x install_git_svn.sh
```

```
[root@localhost ~]# ll
```

```
总用量 4
```

```
-rwxr-xr-x 1 root root 491 5月  6 21:54 install_git_svn.sh
```

2.3 使用脚本安装 Git

1. 运行脚本安装 Git，控制台出现 Git 安装完成提示，即安装完成

```
# ./install_git_svn.sh Git
```

```
[root@iZ8vbclrm6alc5bly61q1lZ ~]# ./install_git_svn.sh Git
```

```
准备安装Git
```

```
Git安装完成！
```

2. 验证 Git 是否安装成功，如出现如下内容，则 git 安装成功。

```
# git --version
```

```
git version 1.8.3.1
```

```
[root@iZ8vbclrm6alc5bly61q1lZ ~]# git --version
```

```
git version 1.8.3.1
```

2.4 使用脚本安装 SVN

1. 运行脚本安装 SVN，出现 SVN 安装完成字样，即安装完成。

```
# ./install_git_svn.sh SVN
```

```
[root@iZ8vbclrm6alc5bly61q1lZ ~]# ./install_git_svn.sh SVN
```

```
准备安装SVN
```

```
SVN安装完成！
```

2. 验证 SVN 是否安装成功，输入 svn -version 命令出现版本信息即安装成功

```
#svn -version
```

```
svn，版本 1.7.14 (r1542130)
```

```
编译于 Apr 11 2018，02:40:28
```

版权所有 (C) 2013 Apache 软件基金会。

此软件包含了许多人的贡献，请查看文件 NOTICE 以获得更多信息。

Subversion 是开放源代码软件，请参阅 <http://subversion.apache.org/> 站点。

可使用以下的版本库访问模块：

* ra_neon：通过 WebDAV 协议使用 neon 访问版本库的模块。

- 处理 “http” 方案
- 处理 “https” 方案


* ra_svn：使用 svn 网络协议访问版本库的模块。 - 使用 Cyrus SASL 认证

- 处理 “svn” 方案

* ra_local：访问本地磁盘的版本库模块。

- 处理 “file” 方案

```
[root@iZ8vbclrm6alc5bly61q1lZ ~]# svn --version
svn, 版本 1.7.14 (r1542130)
编译于 Apr 11 2018, 02:40:28
```



版权所有 (C) 2013 Apache 软件基金会。

此软件包含了许多人的贡献，请查看文件 NOTICE 以获得更多信息。

Subversion 是开放源代码软件，请参阅 <http://subversion.apache.org/> 站点。

可使用以下的版本库访问模块：

* ra_neon：通过 WebDAV 协议使用 neon 访问版本库的模块。

- 处理 “http” 方案
- 处理 “https” 方案

* ra_svn：使用 svn 网络协议访问版本库的模块。 - 使用 Cyrus SASL 认证

- 处理 “svn” 方案

* ra_local：访问本地磁盘的版本库模块。

- 处理 “file” 方案

第 3 章 Git 的使用实验步骤

3.1 使用 Git 创建本地仓库

1. 在当前目录下创建一个目录

```
# mkdir newpro
```

2. 使用 ll 命令查看目录是否创建成功

```
#ll
```

总用量 8

```
-rwxr-xr-x 1 root root 798 5月 7 14:40 install_git_svn.sh
```

```
drwxr-xr-x 3 root root 4096 5月 7 14:52 newpro
```

```
[root@iZ8vbclrm6alc5bly61q1lZ ~]# ll
```

总用量 8

```
-rwxr-xr-x 1 root root 798 5月 7 14:40 install_git_svn.sh
```

```
drwxr-xr-x 2 root root 4096 5月 7 14:52 newpro
```

创建目录成功

3.进入 newpro 目录下

```
#cd newpro
```

3.查看 newpro 目录下的内容

```
# ll -al
```

总用量 0

```
drwxr-xr-x 2 root root 4096 5月 7 14:52 .
```

```
dr-xr-x--- 6 root root 4096 5月 7 14:52 ..
```

```
[root@iZ8vbclrm6alc5bly61q1lZ newpro]# ll -al
```

总用量 8

```
drwxr-xr-x 2 root root 4096 5月 7 14:52 .
```

```
dr-xr-x--- 6 root root 4096 5月 7 14:52 ..
```

4.输入 git init 命令，初始化仓库

```
# git init
```

初始化空的 Git 版本库于 /root/newpro/.git/

```
[root@iZ8vbclrm6alc5bly61q1lZ newpro]# git init
```

初始化空的 Git 版本库于 /root/newpro/.git/

5.查看新创建目录下的文件，生成了.git 目录

```
# ll -al
```

总用量 12

```
drwxr-xr-x 3 root root 4096 5月 7 14:55 .
```

```
dr-xr-x--- 6 root root 4096 5月 7 14:52 ..
```

```
drwxr-xr-x 7 root root 4096 5月 7 14:55 .git
```

```
[root@iZ8vbc1rm6alc5bly61q1lZ newpro]# ll -al
```

```
总用量 12
```

```
drwxr-xr-x 3 root root 4096 5月 7 14:55 .
```

```
dr-xr-x--- 6 root root 4096 5月 7 14:52 ..
```

```
drwxr-xr-x 7 root root 4096 5月 7 14:55 .git
```

```
[root@iZ8vbc1rm6alc5bly61q1lZ newpro]#
```

6.git 本地仓库已经初始化完成，可以在本地直接进行项目代码开发。

3.2 Git 克隆远程代码

1.切换到家目录下

```
#cd ~
```

2.使用 ll 命令查看当目录下的文件

```
# ll
```

```
总用量 8
```

```
-rwxr-xr-x 1 root root 798 5月 7 14:40 install_git_svn.sh
```

```
drwxr-xr-x 3 root root 4096 5月 7 14:55 newpro
```

```
[root@iZ8vbc1rm6alc5bly61q1lZ newpro]# cd ~
```

```
[root@iZ8vbc1rm6alc5bly61q1lZ ~]# ll
```

```
总用量 8
```

```
-rwxr-xr-x 1 root root 798 5月 7 14:40 install_git_svn.sh
```

```
drwxr-xr-x 3 root root 4096 5月 7 14:55 newpro
```

3.拉去远程代码，到本地。

```
# git clone https://github.com/paulboone/ticgit.git
```

```
正克隆到 'ticgit'...
```

```
remote: Enumerating objects: 634, done.
```

```
remote: Total 634 (delta 0), reused 0 (delta 0), pack-reused 634
```

```
接收对象中: 100% (634/634), 88.93 KiB | 0 bytes/s, done.
```

```
处理 delta 中: 100% (261/261), done.
```

```
[root@localhost ~]# git clone https://github.com/paulboone/ticgit.git
正克隆到 'ticgit'...
remote: Enumerating objects: 634, done.
remote: Total 634 (delta 0), reused 0 (delta 0), pack-reused 634
接收对象中: 100% (634/634), 88.93 KiB | 0 bytes/s, done.
处理 delta 中: 100% (261/261), done.
```



4.使用 ll 命令查看，ticgit 目录拉取成功。

```
# ll
总用量 4
-rwxr-xr-x 1 root root 622 5月  6 22:42 install_git_svn.sh
drwxr-xr-x 3 root root  34 5月  7 01:56 newpro
drwxr-xr-x 8 root root 165 5月  7 02:11 ticgit
```

```
[root@localhost ~]# ll
总用量 4
-rwxr-xr-x 1 root root 622 5月  6 22:42 install_git_svn.sh
drwxr-xr-x 3 root root  34 5月  7 01:56 newpro
drwxr-xr-x 8 root root 165 5月  7 02:11 ticgit
```

5.查看 ticgit 目录下的代码文件

```
# ll ticgit/
总用量 36
drwxr-xr-x 2 root root 4096 5月  7 15:03 bin
drwxr-xr-x 2 root root 4096 5月  7 15:03 examples
drwxr-xr-x 3 root root 4096 5月  7 15:03 lib
-rw-r--r-- 1 root root 1073 5月  7 15:03 LICENSE
drwxr-xr-x 2 root root 4096 5月  7 15:03 note
-rw-r--r-- 1 root root  914 5月  7 15:03 Rakefile
-rw-r--r-- 1 root root  887 5月  7 15:03 README
drwxr-xr-x 2 root root 4096 5月  7 15:03 spec
-rw-r--r-- 1 root root  722 5月  7 15:03 ticgit.gemspec
```

6.拉取远程代码拉取成功。