

Learning DevStack

Email: wu.wenxiang@99cloud.net

Email: maodouzi@gmail.com

Sina Weibo: @wu_wenxiang

- [概述](#)
 - [What & Why](#)
 - [安装](#)
 - [基本系统](#)
 - [常规安装](#)
 - [Patch安装](#)
 - [源代码分析](#)
 - [代码结构](#)
 - [stack.sh](#)
 - [总结](#)
-

概述

What & Why

DevStack项目是OpenStack的子项目之一，用于方便开发人员快速搭建OpenStack开发平台。

安装

基本系统

- 硬件要求
 - CPU: 2 Core
 - MEM: 4 GB
 - NIC: 2 × 1GB
- OS版本
 - Ubuntu-12.04-Server(64bit)

- 安装步骤

1. 安装の設定

所有的OpenStack组件安装在一台机器上，这台机器被称为Target机器

远程连接到Target机器上干活的机器，被称为Client机器

2. 在Target机器上：安装基本OS系统

用户名：看你自己喜欢啦

密码：记好，别让穿黑衣服的客人看见！

3. 在Target机器上：安装SSH服务器

```
sudo apt-get install ssh -y
```

ifconfig # 记住远程可以访问到的IP地址

4. 在Client机器上: 配置登录Target机器的SSH密钥

```
USER=pear;REMOTE_IP=192.168.229.138
#改成Target机器上的用户名和对外IP地址
scp ~/.ssh/id_dsa.pub ${USER}@${REMOTE_IP}:/home/${USER}/
ssh ${USER}@${REMOTE_IP} "mkdir -p ~/.ssh && \
cat ~/id_dsa.pub >> .ssh/authorized_keys"
```

5. 在Client机器上: 远程登录到Target机器, 不需要密码啦~

```
ssh ${USER}@${REMOTE_IP}
```

6. 在Target机器上: 配置sudo, git, vim等

设置sudo

```
sudo passwd # 设置root用户的密码
su -
chmod a+w /etc/sudoers
echo "pear ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
# pear->用户名
chmod a-w /etc/sudoers
exit
```

更新源

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.us
sudo sed -i "s/us/cn/" /etc/apt/sources.list
sudo apt-get update && sudo apt-get dist-upgrade -y
```

安装和设置vim

```
sudo apt-get install vim -y
sudo chmod o+w /etc/vim/vimrc;
echo -e "\nset smarttab\nset tabstop=4\nset shiftwidth=4" \
>> /etc/vim/vimrc
sudo chmod o-w /etc/vim/vimrc
```

安装和设置git

```
sudo apt-get install git-core git-doc git -y
git config --global user.name "Pear" # Pear->全名
git config --global user.email "Pear@pear.net" # 邮箱
#git config --global color.ui "always" #Windows
git config --global color.ui "auto" #Linux
git config --global alias.ci "commit"
git config --global alias.co "checkout"
git config --global alias.st "status"
git config --global alias.br "branch"
git config --global core.editor "vim"
```

清理

```
sudo apt-get autoremove -y
sudo apt-get clean
```

基本系统安装完毕，如果亲们是在虚拟机上安装的，可以备个份啦~

常规安装

- 准备工作：在Target机器上

1. 添加stack用户

```
sudo adduser stack
su -
chmod a+w /etc/sudoers
echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
chmod a-w /etc/sudoers
exit
```

2. 下载最新的devstack源码（本文写作时，OpenStack为F版本）

```
cd ~
git clone git://github.com/openstack-dev/devstack.git
```

- 安装devstack

1. 安装步骤如下，但是先不要做！因为还需要进行一些设置，设置已经写出自动化脚本，见[Patch安装](#)。

```
cd ~/devstack
./stack.sh
```

再次提醒！不要直接运行，按照[Patch安装章节](#)打patch先~

Patch安装

- What & Why

直接用devstack安装有两个地方比较麻烦：

1. 安装devstack之前，需要写配置文件localrc。
2. 安装好devstack之后，OpenStack的各个服务会按照localrc和stackrc里的配置启动起来。但是重启Target机器之后，OpenStack各个组件服务并不会自动起来，如何重新启动各个服务？两个办法：
 1. 逐个组件手动重启。缺点：太麻烦。
 2. 再次运行stack.sh。缺点：所有的组件会重新下载安装，浪费网络流量和时间。并且配置文件会刷回最原始的状态，数据库里的信息会全部被革除，不能忍受~

为了解决这些问题，达到简单方便地启用各个模块的目的，我们需要给devstack打上patch。

- patch安装devstack流程：

1. 下载[patch包](#)

2. 拷贝patch包到Target机器上~/devstack目录下

3. 在Target机器上，解压patch包

```
cd ~/devstack
tar zxvf patch.tar.gz
```

4. 在Target机器上，安装命令

```
cd ~/devstack/patch
./patch.sh
./stack.sh # create devstack enviroment
./config.sh
./unstack.sh # stop openstack
./re-stack.sh # restart openstack
```

5. 装好啦！在Client机器上用浏览器看吧

http://192.168.229.138 => 这里改成你的Target机器的IP地址

用户名: admin, 密码: 231

• patch启动devstack流程:

1. 安装好之后，启动devstack的命令，在Target机器上

```
cd ~/devstack ./re-stack.sh
```

2. 关闭devstack的命令，在Target机器上

```
cd ~/devstack ./unstack.sh
```

源代码分析

代码结构

主要代码结构如下:

```
./files
./files/horizon_settings.py # 完成horizon设置
./files/keystone_data.sh # keystone对用户, tenant, service等对象的初始
./functions # stack.sh中调用到的库函数
./lib
./lib/ceilometer
./lib/cinder
./lib/database # 通用数据库操作接口, 包括清空/重建数据库等操作
./lib/databases
./lib/databases/mysql
./lib/databases/postgresql
./lib/glance
./lib/heat
./lib/keystone # keystone配置操作, init/config/clear_up, 其余组件也类似
./lib/nova
./lib/quantum
./lib/tempest
```

```
./lib/template
./openrc
./rejoin-stack.sh
./samples
./samples/local.sh
./samples/localrc # localrc文件的范例，用于配置openstack
./stack.sh # openstack安装和启动脚本，是最主要的脚本
./stackrc # openstack配置文件，其中的配置会被localrc override
./unstack.sh # stop openstack各个组件的运行。
```

stack.sh

stack.sh的代码流程如下：

```
# Settings
    * Proxy Settings
# Sanity Check
    * root Access
# Common Configuration
# Configure Projects
    * Nova Network Configuration
    * Quantum Networking
    * Database Configuration
    * RabbitMQ or Qpid
    * Swift
    * Keystone
    * Horizon
    * Log files
    * Set Up Script Execution
# Install Packages
    * Check Out Source
# Initialization
    * Syslog
    * Finalize queue installation
    * Configure database
    * Configure screen
    * Keystone
    * Horizon
    * Glance
    * Ryu
    * Quantum
    * Nova
    * Storage Service
    * Volume Service
# Launch Services
# Install Images
# Run local script
# Fin
    * Using the cloud
```

总结

打完收工。