大报告：部署 OpenStack 并验证

梁嘉伟，李崇泽，李金龙

2025 年 6 月 9 日

目录

1 实验背景

OpenStack 简介及核心组件 OpenStack 是一个开源的云计算管理平台项目，旨在为公有云和私有云提供可扩展的、灵活的基础设施即服务（IaaS）。它由多个相互协作的服务组件构成，每个组件负责不同的功能模块，共同构建出完整的云环境。

以下是 OpenStack 的主要核心组件：

- **Nova (计算服务)**：负责管理虚拟机的生命周期，包括创建、调度、销毁等操作。
- **Glance (镜像服务)**：用于存储和检索虚拟机镜像，支持多种格式如 qcow2、vmdk 等。
- **Cinder (块存储服务)**：为虚拟机实例提供持久化的块存储设备。
- **Swift (对象存储服务)**：提供高可用、分布式、最终一致性的对象存储服务。
- **Neutron (网络服务)**：管理虚拟网络资源，包括 IP 分配、子网划分、路由等。
- **Keystone (身份认证服务)**：提供用户身份验证、服务目录和令牌管理。
- **Horizon (仪表盘)**：基于 Web 的图形化用户界面，方便用户管理和操作 OpenStack 资源。
- **Heat (编排服务)**：支持通过模板来自动化部署云应用及其相关资源。
- **Ceilometer (计量服务)**：收集系统中各种资源的使用数据，用于监控和计费。

这些组件可以按需部署和组合，从而构建出高度定制化的云平台。

OpenStack 的功能与应用场景 OpenStack 提供了丰富的功能，能够实现对大规模计算、存储和网络资源的集中管理和自动化调度。其主要功能包括：

- **虚拟机管理**：支持快速创建、启动、停止和删除虚拟机实例，提升资源利用率。
- **弹性伸缩**：根据负载自动调整资源分配，提高系统稳定性和性能。
- **多租户架构**：支持多个用户/组织共享同一套物理资源，同时保证资源隔离。
- **API 驱动**：所有功能都可通过 RESTful API 进行调用，便于集成和自动化运维。
- **高可用性**：支持故障转移、冗余备份等功能，保障业务连续性。

在实际应用中，OpenStack 被广泛用于以下场景：

- **企业私有云建设**：为企业内部提供统一的 IT 资源管理平台，降低运维成本。
- **科研机构与高校实验室**：搭建教学或研究用的云平台，模拟真实云环境。
- **公有云提供商**：作为底层基础设施支撑对外提供的 IaaS 服务。
- **混合云部署**：结合其他云平台（如 AWS、Azure）实现跨平台资源调度。

2 packstack 介绍

Packstack 简介 Packstack 是一个基于 Puppet 模块的自动化部署工具，专为快速安装和配置 OpenStack 而设计。它最初由 Red Hat 开发，适用于 RHEL 及其衍生发行版（如 CentOS）。Packstack 并不是一个完整的云管理平台，而是一个用于搭建 OpenStack 基础架构的实用工具。

Packstack 的主要特性

- **简单易用**：Packstack 提供了一种非常简便的方式来部署 OpenStack。用户只需运行一条命令即可完成所有基本组件的安装和配置，非常适合初学者或需要快速搭建测试环境的开发者。
- **自动化配置**：通过内部调用 Puppet 模块，Packstack 能够自动安装并配置 Keystone（身份认证）、Nova（计算服务）、Glance（镜像服务）、Neutron（网络服务）、Cinder（块存储）等核心服务，以及 MySQL、RabbitMQ 等底层依赖服务。
- **支持 All-in-One 部署模式**：Packstack 支持单节点部署模式（All-in-One），即在一台服务器上部署整个 OpenStack 架构。这种模式适合开发、测试和演示用途，便于快速验证功能，但不适用于生产环境。
- **支持多节点分布式部署**：除了单节点部署外，Packstack 还可以通过提供 answer file（应答文件）来定义不同组件的部署位置，从而实现跨多个节点的分布式部署。例如，可以指定数据库运行在某台主机上，计算节点部署在另一台主机上。
- **可定制性强**：生成的 answer file 允许用户自定义各种参数，包括启用/禁用特定组件、设置管理员密码、配置网络接口等，从而满足不同的部署需求。
- **与操作系统集成良好**：Packstack 主要针对 CentOS/RHEL 系统进行了优化，能够很好地与系统的服务管理机制（如 systemd）集成，确保服务稳定运行。
- **社区支持广泛**：作为 OpenStack 生态的一部分，Packstack 拥有活跃的社区支持，并且文档较为齐全，便于排查问题和扩展功能。

选择 Packstack 的原因　在网上查阅了多种部署方法。跟着这些方法进行了尝试，最终只有 packstack 实现了成功部署。大部分失败的原因还是 centos7 的问题，如 centos7 的 yum 源无法访问。导致各种各样的软件包确实失效问题。即使通过 ai 辅助也难以完成。最终通过 packstack 快速完成了部署。

3 实验环境

- 操作系统：CentOS 7
- CPU：4 核
- 内存：8GB
- 硬盘：100GB

4 centos7 环境配置

1. 配置网络 ip 地址，网关等

```
vi /etc/sysconfig/network-scripts/ifcfg-eth30
DEVICE=eth30
BOOTPROTO=static
```

```
ONBOOT=yes
IPADDR=192.168.10.10
NETMASK=255.255.255.0
GATEWAY=192.168.10.2
DNS1=114.114.114.114
```

2. 通过 xshell8 完成连接：

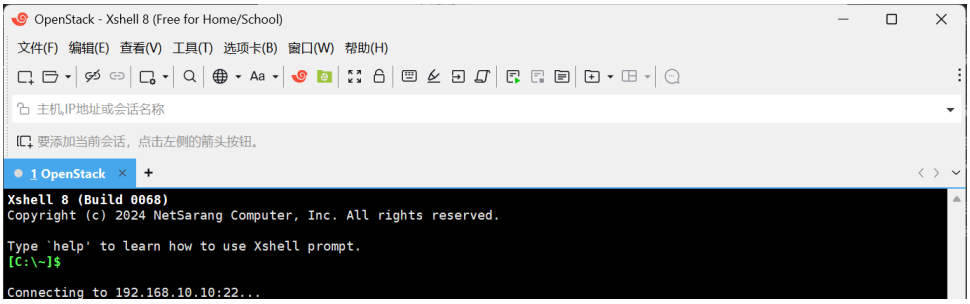


图 1: xshell8 连接

3. 关闭 selinux

```
vi /etc/selinux/config
SELINUX=disabled
```

4. 关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

5. 关闭网络服务

```
systemctl stop NetworkManager
systemctl disable NetworkManager
```

6. 配置 hosts

```
192.168.10.10 OpenStack
```

7. 更新 Yum 仓库：由于 CentOS 7 官方已停止维护，系统默认的 Yum 源不可用，需更换为仍在维护的镜像源（如阿里云、清华大学镜像站等），以确保软件包的正常安装与更新。本实验的最大难点在此，难以处理各种失效的软件包和失踪的软件包。

```
# 清空系统默认的yum源
rm -rf /etc/yum.repos.d/*

#拉取网络yum源
# curl -o /etc/yum.repos.d/CentOS-Base.repo https://mirrors.aliyun.com/repo/Centos-7.re
# curl -o /etc/yum.repos.d/epel.repo https://mirrors.aliyun.com/repo/epel-7.repo

# 安装openstack的yum源
yum -y install centos-release-openstack-stein

#修改yum源文件

vi /etc/yum.repos.d/CentOS-OpenStack-stein.repo

[centos-openstack-stein]

name=CentOS-7 - OpenStack stein

baseurl=https://mirrors.aliyun.com/centos/7/cloud/x86_64/openstack-stein/

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=cloud

[centos-openstack-stein-test]
name=CentOS-7 - OpenStack stein Testing
baseurl=https://mirrors.aliyun.com/centos/7/cloud/$basearch/openstack-stein/


[centos-openstack-stein-debuginfo]
name=CentOS-7 - OpenStack stein - Debug
baseurl=http://mirrors.aliyun.com/centos/7/cloud/$basearch/openstack-stein/

[centos-openstack-stein-source]
name=CentOS-7 - OpenStack stein - Source
baseurl=http://vault.centos.org/centos/7/cloud/Source/openstack-stein/


[rdo-trunk-stein-tested]
name=OpenStack stein Trunk Tested
baseurl=https://trunk.rdoproject.org/centos7-stein/current-passed-ci/

[qemu-kvm]
name=CentOS-$releasever - Base
```

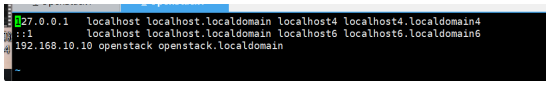
```
release=$releasever&arch=$basearch&repo=os&infra=$infra
baseurl=http://mirrors.aliyun.com/centos/7/virt/x86_64/kvm-common/
enabled=1
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

# 清除缓存并生成新缓存
yum clean all
yum makecache
```

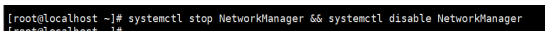
8. 安装 packstack 软件包工具

```
yum install -y openstack-packstack
```

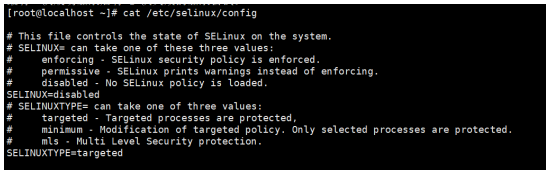
最终一系列命令截图如下：



(a) 配置 ip 地址映射

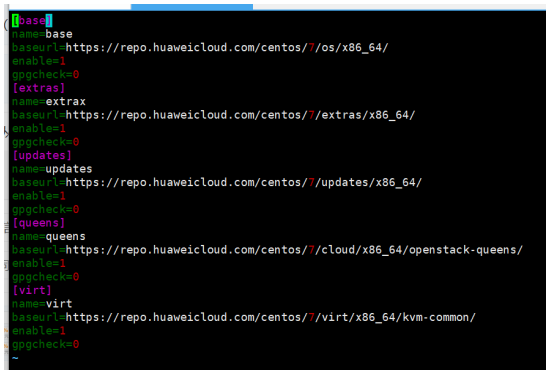


(b) 关闭防火墙

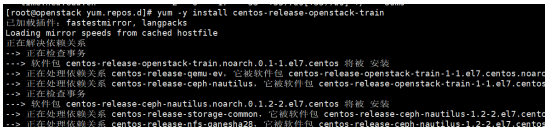


(c) 关闭 selinux

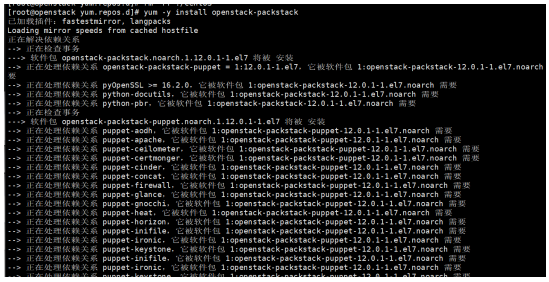
(d) 切换到 yum 目录，清空原有 yum 源



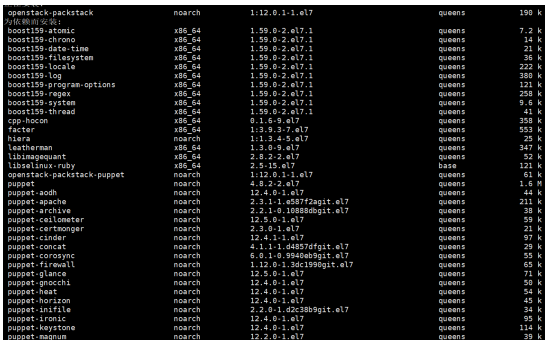
(e) 配置 yum 源



(f) 下载软件库



(g) 下载 packstack1



(h) 下载 packstack2

图 2: 虚拟机环境配置全过程截图汇总

5 部署 OpenStack

1. 使用 packstack 进行部署：

```
packstack --allinone
```

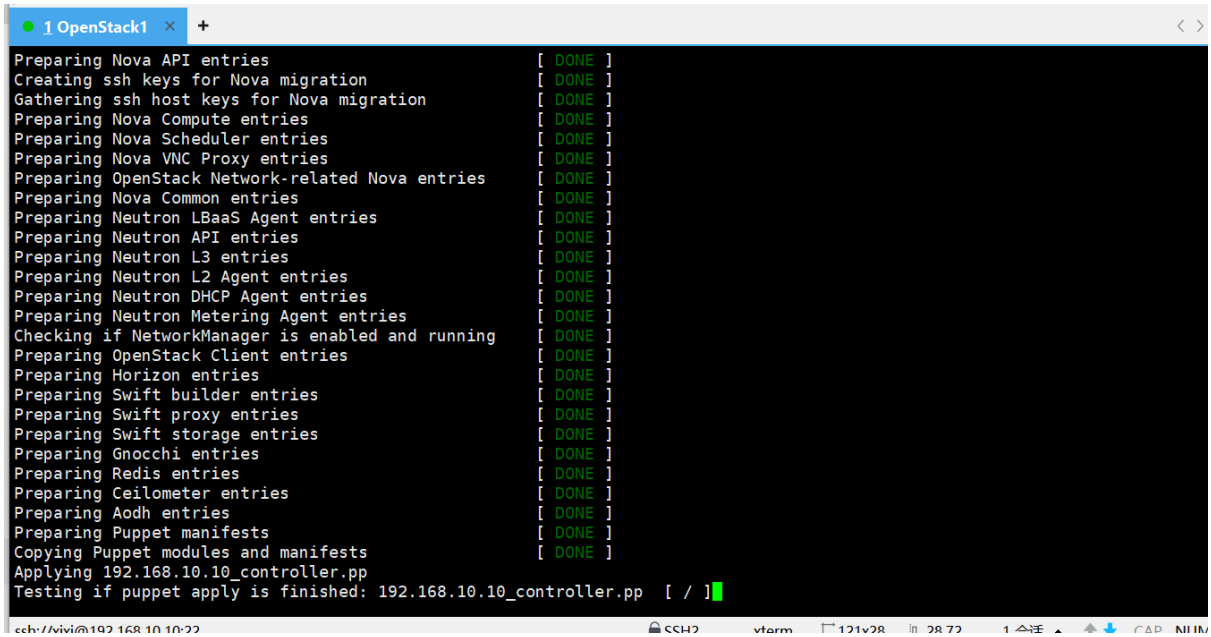


图 3: packstack 部署结果

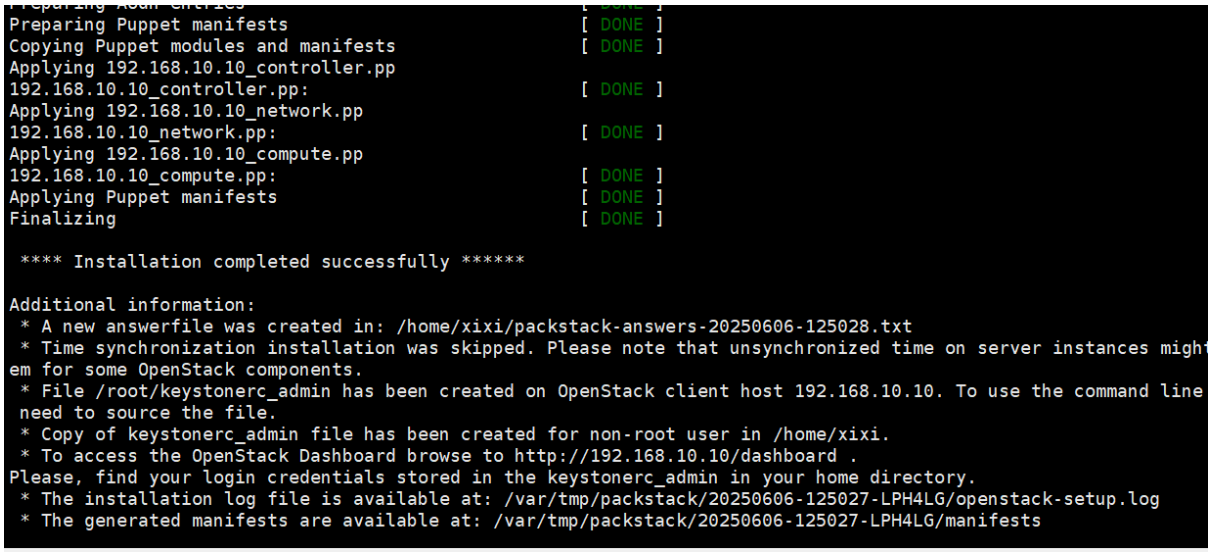


图 4: 通过 packstack 一键部署成功

6 验证部署

1. 登录 Horizon 界面：

   `http://192.168.10.10/dashboard`

图 5: Horizon 登录界面

2. 使用默认用户名和密码登录：(packstack 部署时会自动生成 admin 用户的密码)

```
# 获取自动生成的密码
cat Keystonerc_admin
```

```
[xixi@openstack ~]$ source keystonerc_admin
[xixi@openstack ~(keystone_admin)]$ cat keystonerc_admin
unset OS_SERVICE_TOKEN
    export OS_USERNAME=admin
    export OS_PASSWORD='5c8f973e68854b7c'
    export OS_REGION_NAME=RegionOne
    export OS_AUTH_URL=http://192.168.10.10:5000/v3
    export PS1='[\u@\h \W(keystone_admin)]\$ '

export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_IDENTITY_API_VERSION=3
    [xixi@openstack ~(keystone_admin)]$ cat keystonerc_demo
cat: keystonerc_demo: 没有那个文件或目录
```

图 6: 查看生成的用户名和密码

7 OpenStack 部署成功展示

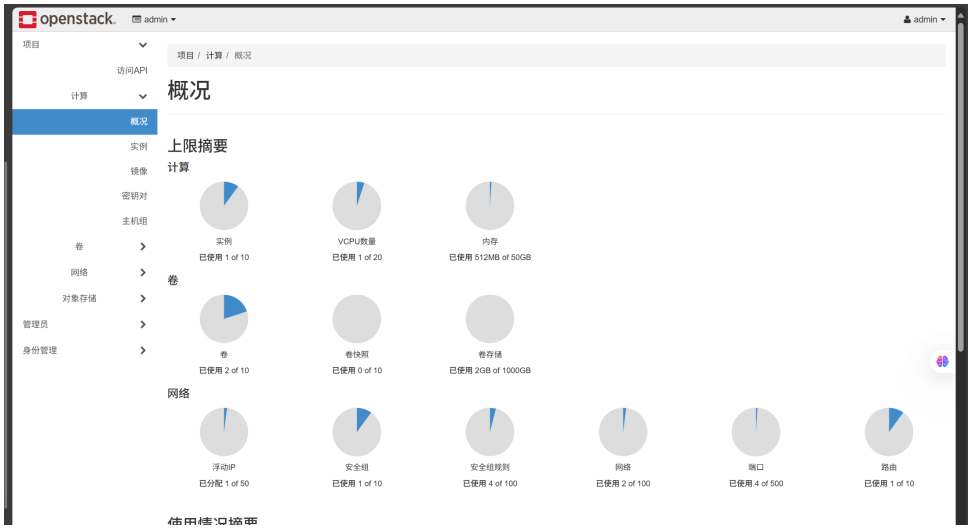


图 7: Horizon 仪表盘截图

8 OpenStack 上的网络创建

在 OpenStack 成功部署并登录 Horizon 管理界面后，我们通过图形化方式完成了虚拟网络的创建。该网络用于支持后续虚拟机实例之间的通信。

8.1 创建网络

1. 登录 Horizon 界面，进入 **项目 -> 网络 -> 网络**。
2. 点击“创建”按钮，填写以下信息：- 网络名称 - 子网名称 - 网络地址 - 启用 DHCP - 网关
3. 提交表单后，系统将自动创建一个内部网络及对应的子网。

8.2 配置外部网络连接

为了使虚拟机能够访问外网，我们需要为路由器绑定外部网络：

1. 进入 **项目 -> 网络 -> 路由器**。2. 点击“创建路由器”，填写路由器名称。3. 创建完成后点击路由器名称进入详情页，点击“添加接口”并选择之前创建的子网。4. 在“External Gateway Info”部分，选择外部网络（通常命名为 ‘ext-net‘ 或类似），完成绑定。

(a) 创建网络

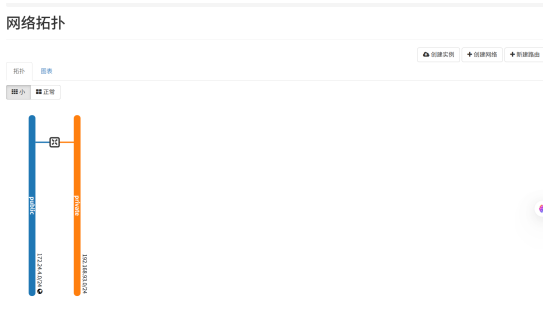
(b) 创建子网

(c) 创建子网

(d) 创建路由

(e) 创建路由

(f) 路由添加接口



(g) 路由处理后网络拓扑

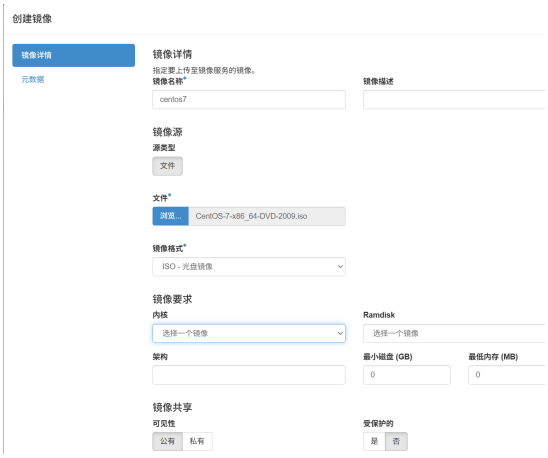
图 8: 网络配置全过程截图汇总

通过上述步骤，我们在 OpenStack 中成功创建了一个可以上网的虚拟网络环境，为后续云主机

的创建和测试打下了基础。

9 OpenStack 上的云实例创建

云实例的创建类似于虚拟机的创建，下面截图描述了云实例的创建过程：



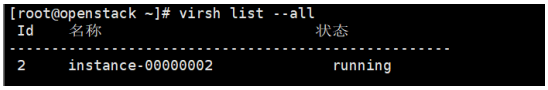
(a) 创建镜像

(b) 创建实例

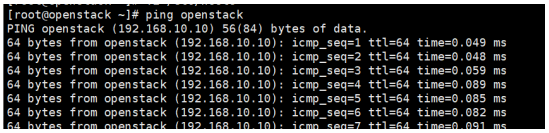
(c) 创建实例选择源

(d) 实例类型

(e) 实例创建成功



(f) 查看创建云主机实例



(g) 尝试 ping 通主机名

图 9: 云主机全过程截图汇总

10 结论

通过本次实验，我们成功在单节点环境下部署了 OpenStack，并利用 Horizon 管理界面对其部分功能进行了验证与操作。此次实践加深了我们对 OpenStack 架构及其运作机制的理解。通过对平台各项功能的探索，我们也初步了解了诸如阿里云、腾讯云等大型云计算服务提供商是如何构建和管理其云服务体系的。

11 整体任务分工

实验 1 由李崇泽负责，实验 2 由梁嘉伟负责，实验 3 由李金龙负责。大作业梁嘉伟负责 OpenStack 单节点部署以及相关功能的验证和报告撰写。李崇泽尝试多节点部署但是并未成功