基于 openstack 的 IaaS 部署与验证

戴亮 陈箫彤 孙泰利

June 11, 2025

一、系统环境配置与软件源优化

0.1 APT 源配置

首先将 apt 源更换为国内镜像源，以加速软件包下载速度。

OpenStack 部署需要大量安装大量软件，为提高部署效率，将系统 APT 源更换为国内镜像源。Ubuntu 系统默认使用海外软件源，国内访问速度较慢且稳定性不足。更换国内源后，能显著加速软件包下载过程。

配置方法为修改/etc/apt/sources.list 文件，替换为国内镜像源地址。配置完成后执行 apt update 命令验证源可用性并更新源，如图1所示。

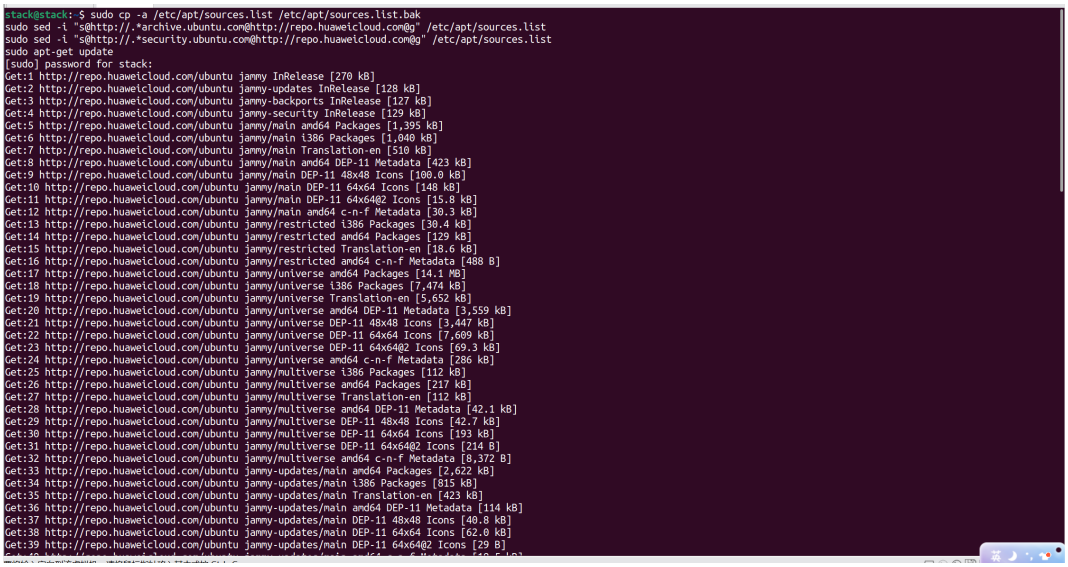

图 1: apt 源更换配置过程

0.2 所需依赖项的安装

后续的安装过程中需要用到一些软件，即依赖项，主要包含是以下三个软件：

- **bridge-utils**：用于创建虚拟网络桥接，实现实例间通信
- **git**：用于从官方的 git 仓库中拉取部署时需要使用的源码
- **python3-pip**：用于安装源码运行所需要的 python 依赖

安装过程如图2所示，此步骤确保后续部署的基础环境完备。

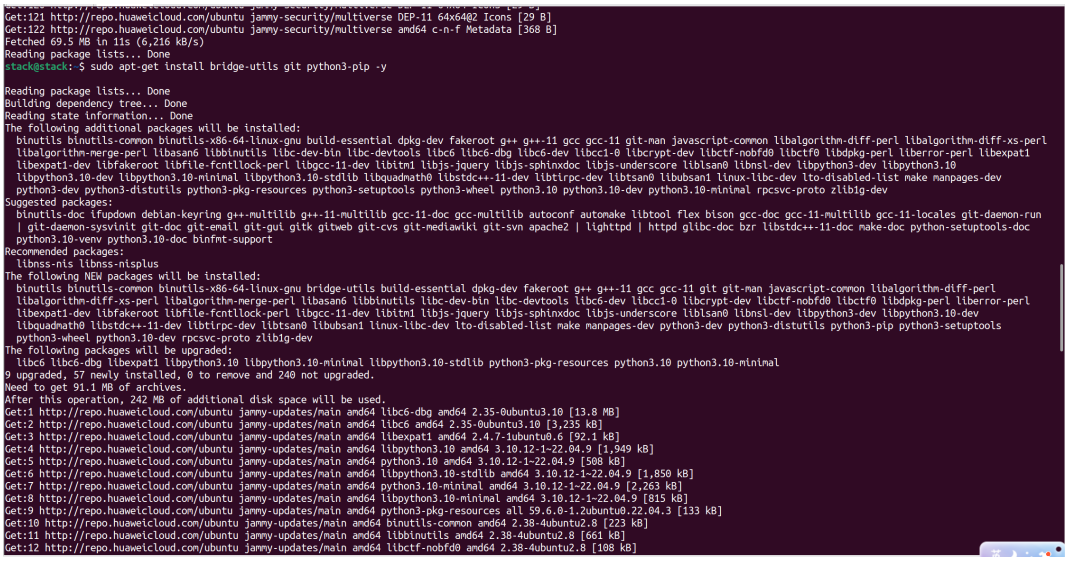

图 2: 依赖包安装过程

1 准备 DevStack 一键部署脚本

1.1 DevStack 脚本概述

DevStack 是 OpenStack 基金会维护的一体化部署工具，采用 Shell 脚本实现自动化安装。其核心优势在于：

1. 自动配置所有 OpenStack 服务（Nova, Neutron, Keystone 等）
2. 内置依赖解析机制，解决组件版本冲突
3. 提供开发调试模式，支持快速迭代

1.2 拉取 DevStack

我们采用 OpenStack 官方提供的 DevStack 一键安装脚本进行安装。

此脚本会自动在本地创建一个 OpenStack 一体化集群，并安装其使用的所有中间件，且自动向操作系统注册各个服务。

先从 OpenStack 的官方 git 仓库拉取 DevStack。

图 3: 从 Git 仓库克隆 DevStack 仓库

1.3 pip 换源

DevStack 中用到了 pip，将 pip 源更换为国内镜像源，加速 Python 包下载。

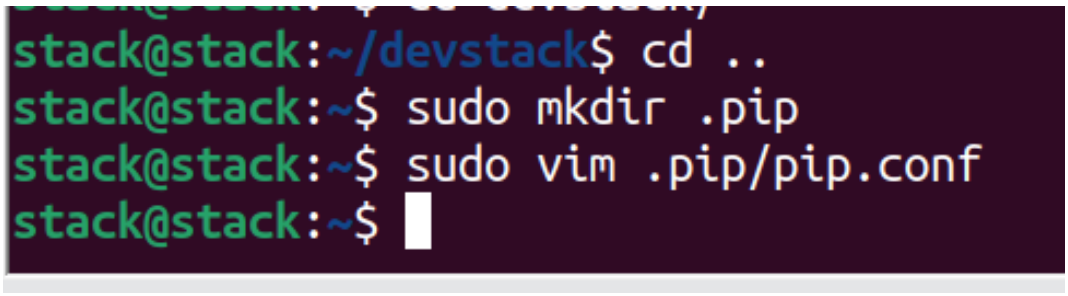

图 4: pip 配置文件创建与修改

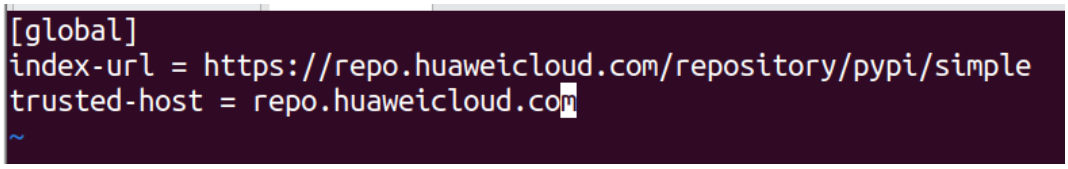

图 5: pip 配置文件内容

2 OpenStack 部署

2.1 安装 openstack

配置管理员密码并启动 DevStack 安装脚本。

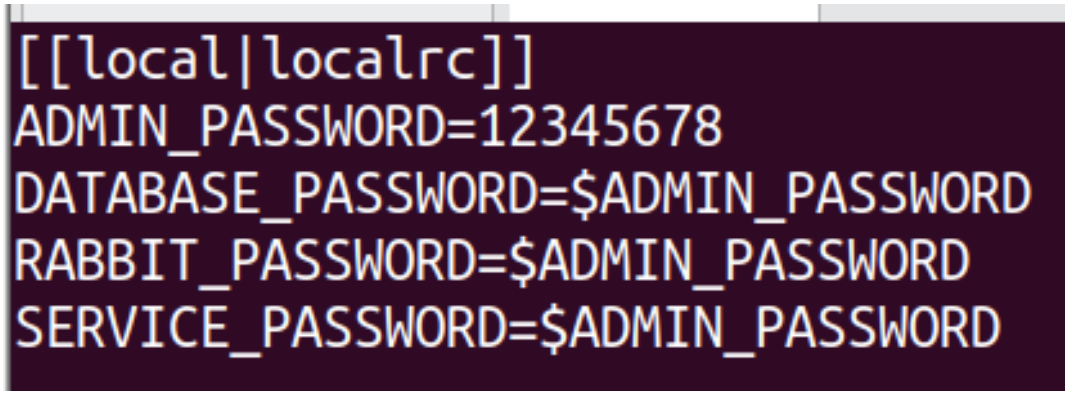

图 6: DevStack 配置

```
stack@stack:~$ sudo vim .pip/pip.conf
stack@stack:~$ cd devstack/
stack@stack:~/devstack$ vim local.conf
stack@stack:~/devstack$ ./stack.sh
+ unset GREP_OPTIONS
+ unset LANG
+ unset LANGUAGE
+ LC_ALL=en_US.utf8
+ export LC_ALL
++ env
++ cut -d = -f 1
++ grep -E '^OS_'
+ unset
+ umask 022
+ PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin:/usr/local/bin:/usr/local/sbin:/usr/sbin:/sbin
+++ dirname ./stack.sh
++ cd .
++ pwd
+ TOP_DIR=/home/stack/devstack
+ NOUNSET=
+ [[ -n '' ]]
++ date +%s
+ DEVSTACK_START_TIME=1749508108
+ [[ -r /home/stack/devstack/.stackenv ]]
+ FILES=/home/stack/devstack/files
+ '[' '!' -d /home/stack/devstack/files ']'
+ '[' '!' -d /home/stack/devstack/inc ']'
+ '[' '!' -d /home/stack/devstack/lib ']'
+ [[ '' == \y ]]
+ [[ 1000 -eq 0 ]]
+ [[ -n '' ]]
+ [[ -e /home/stack/.no-devstack ]]
+ LAST_SPINNER_PID=
+ source /home/stack/devstack/functions
++ [[ -z '' ]]
++ declare -r -g _DEVSTACK_FUNCTIONS=1
++++ dirname /home/stack/devstack/functions
+++ cd /home/stack/devstack
+++ pwd
++ FUNC_DIR=/home/stack/devstack
++ source /home/stack/devstack/functions-common
++++ set +o
++++ grep xtrace
+++ _XTRACE_FUNCTIONS_COMMON='set -o xtrace'
```

图 7: OpenStack 安装开始

```
| neutron     | DELETE    |    32 |
| placement   | SELECT    |    30 |
| placement   | INSERT    |    66 |
| placement   | SET       |     2 |
| neutron     | UPDATE    |   323 |
| nova_api    | SELECT    |    50 |
| nova_cell0  | SELECT    |    30 |
| nova_cell1  | SELECT    |   104 |
| nova_cell0  | INSERT    |     3 |
| nova_cell0  | UPDATE    |     7 |
| nova_cell1  | INSERT    |     4 |
| cinder      | SELECT    |    57 |
| placement   | UPDATE    |     3 |
| cinder      | INSERT    |     5 |
| nova_cell1  | UPDATE    |    31 |
| cinder      | UPDATE    |     5 |
| glance      | INSERT    |    14 |
| cinder      | DELETE    |     1 |
| glance      | SELECT    |    30 |
| glance      | UPDATE    |     2 |
| nova_api    | INSERT    |    20 |
| nova_api    | SAVEPOINT |    10 |
| nova_api    | RELEASE   |    10 |
+-------------+-----------+-------+



This is your host IP address: 192.168.217.132
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.217.132/dashboard
Keystone is serving at http://192.168.217.132/identity/
The default users are: admin and demo
The password: 12345678

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: 2025.2
Change: 6ebe6f1b26b371b11ecd9cb6d0d154839ba6941e Updated from generate-devstack-plugins-list 2025-05-22 02:13:55 +0000
OS Version: Ubuntu 22.04 jammy

2025-06-09 23:09:11.374 | stack.sh completed in 2443 seconds.
stack@stack:~/devstack$
```

图 8: OpenStack 安装完成

2.2 环境变量配置

设置后续操作所需的环境变量。

```
stack@stack:~/devstack$ vim ~/.bashrc
stack@stack:~/devstack$ source ~/.bashrc
```

图 9: 环境变量配置过程

```
export OS_USERNAME=admin
export OS_PASSWORD=12345678
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_AUTH_URL=http://localhost/identity
export OS_IDENTITY_API_VERSION=3
```

图 10: 所配置的环境变量

2.3 检查部署状态

检查各服务运行状态是否正常。

```
stack@stack:~/devstack$ sudo systemctl status "devstack@*"
● devstack@etcd.service - Devstack devstack@etcd.service
     Loaded: loaded (/etc/systemd/system/devstack@etcd.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2025-06-10 06:56:43 CST; 14min ago
   Main PID: 75070 (etcd)
      Tasks: 8 (limit: 4551)
     Memory: 21.1M
        CPU: 9.845s
     CGroup: /system.slice/system-devstack.slice/devstack@etcd.service
             └─75070 /opt/stack/bin/etcd --name stack --data-dir /opt/stack/data/etcd --initial-cluster-state new --initial-cluster-token etcd-cluster-01 --initial-clu>

6月 10 06:56:43 stack etcd[75070]: setting up the initial cluster version to 3.4
6月 10 06:56:43 stack etcd[75070]: set the initial cluster version to 3.4
6月 10 06:56:43 stack etcd[75070]: published {Name:stack ClientURLs:[http://192.168.217.132:2379]} to cluster cca3b3eb8897de71
6月 10 06:56:43 stack etcd[75070]: enabled capabilities for version 3.4
6月 10 06:56:43 stack etcd[75070]: ready to serve client requests
6月 10 06:56:43 stack systemd[1]: Started Devstack devstack@etcd.service.
6月 10 06:56:43 stack etcd[75070]: serving insecure client requests on 192.168.217.132:2379, this is strongly discouraged!
6月 10 07:04:37 stack etcd[75070]: start time = 2025-06-10 07:04:37.591241856 +0800 CST m=+478.003222695, time spent = 34.993784ms, remote = 192.168.217.132:44068, res>
6月 10 07:04:37 stack etcd[75070]: start time = 2025-06-10 07:04:37.591048458 +0800 CST m=+478.003029228, time spent = 35.034921ms, remote = 192.168.217.132:44068, res>
6月 10 07:04:58 stack etcd[75070]: start time = 2025-06-10 07:04:58.419157411 +0800 CST m=+498.831138170, time spent = 1.802167ms, remote = 192.168.217.132:44068, resp>

● devstack@keystone.service - Devstack devstack@keystone.service
     Loaded: loaded (/etc/systemd/system/devstack@keystone.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2025-06-10 06:57:06 CST; 14min ago
   Main PID: 76385 (uwsgi)
     Status: "uWSGI is ready"
      Tasks: 5 (limit: 4551)
     Memory: 160.1M
        CPU: 1min 651ms
     CGroup: /system.slice/system-devstack.slice/devstack@keystone.service
             ├─76385 "keystoneuWSGI master"
             ├─76412 "keystoneuWSGI worker 1"
             └─76413 "keystoneuWSGI worker 2"

6月 10 07:10:29 stack devstack@keystone.service[76412]: DEBUG keystone.server.flask.request_processing.middleware.auth_context [None req-51d19fde-93af-46b0-920c-e10729>
6月 10 07:10:29 stack devstack@keystone.service[76412]: DEBUG keystone.server.flask.request_processing.req_logging [None req-51d19fde-93af-46b0-920c-e1072969c41f servi>
6月 10 07:10:29 stack devstack@keystone.service[76412]: DEBUG keystone.server.flask.request_processing.req_logging [None req-51d19fde-93af-46b0-920c-e1072969c41f servi>
6月 10 07:10:29 stack devstack@keystone.service[76412]: DEBUG keystone.server.flask.request_processing.req_logging [None req-51d19fde-93af-46b0-920c-e1072969c41f servi>
6月 10 07:10:29 stack devstack@keystone.service[76412]: DEBUG keystone.common.fernet_utils [None req-51d19fde-93af-46b0-920c-e1072969c41f service placement>
```

图 11: OpenStack 服务状态检查

所有服务均正常运行。

2.4 平台使用验证

访问 OpenStack 本地集群的 Web UI, 进行基本操作测试。
先使用设置的密码登录，进入主页, 并切换语言为中文

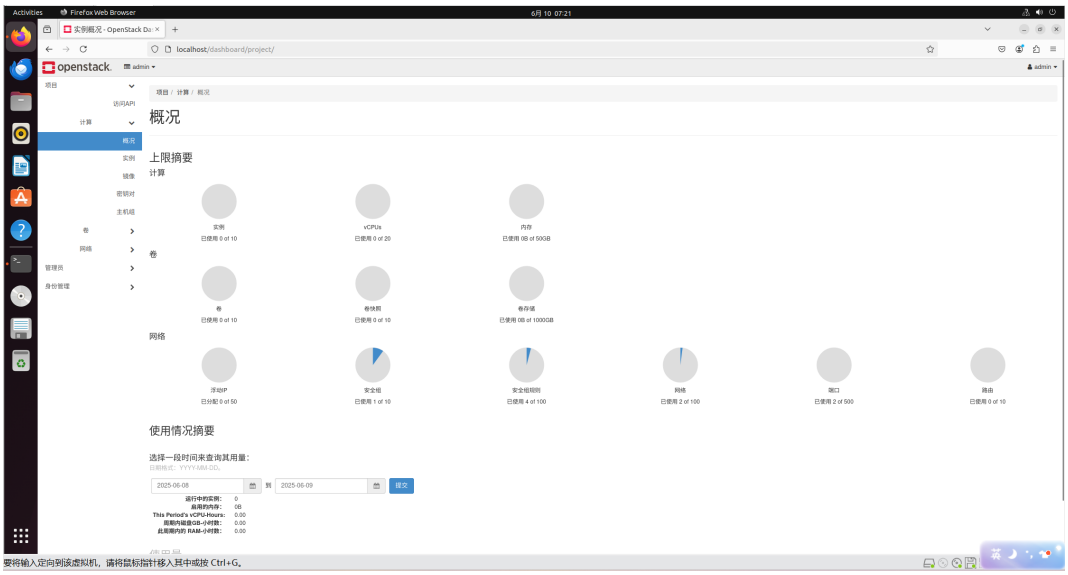

图 12: OpenStack 中文控制面板

一切正常, 成功完成搭建
对 OpenStack 进行简单使用, 创建实例进行功能验证:

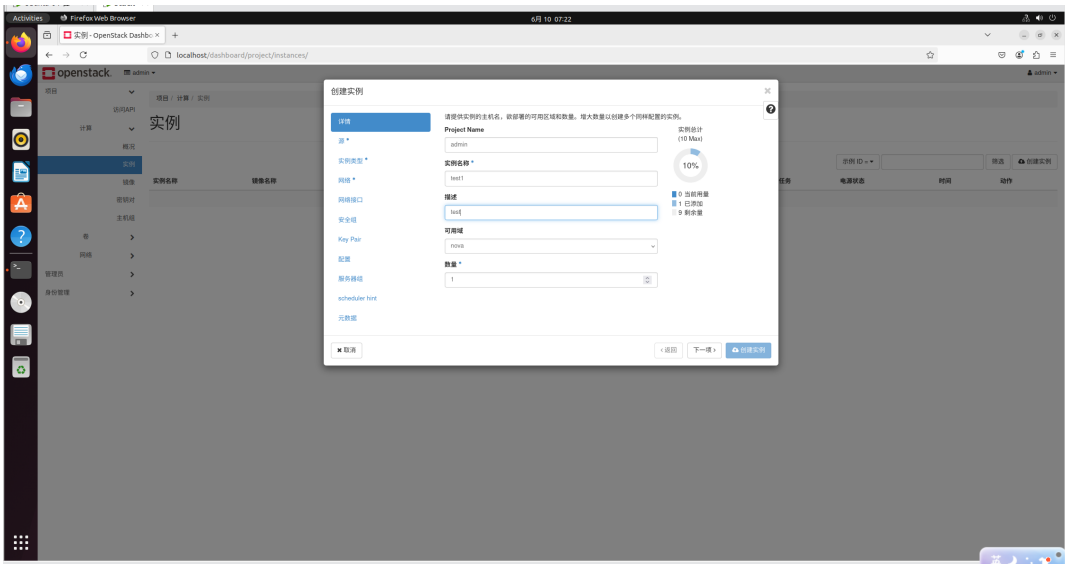

图 13: 创建新实例

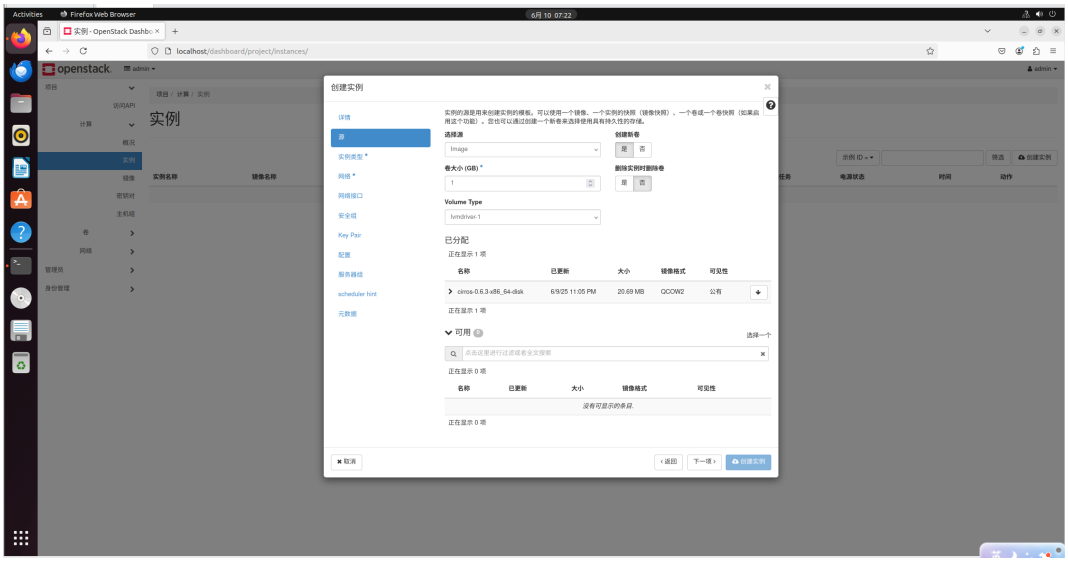

图 14: 实例配置选项

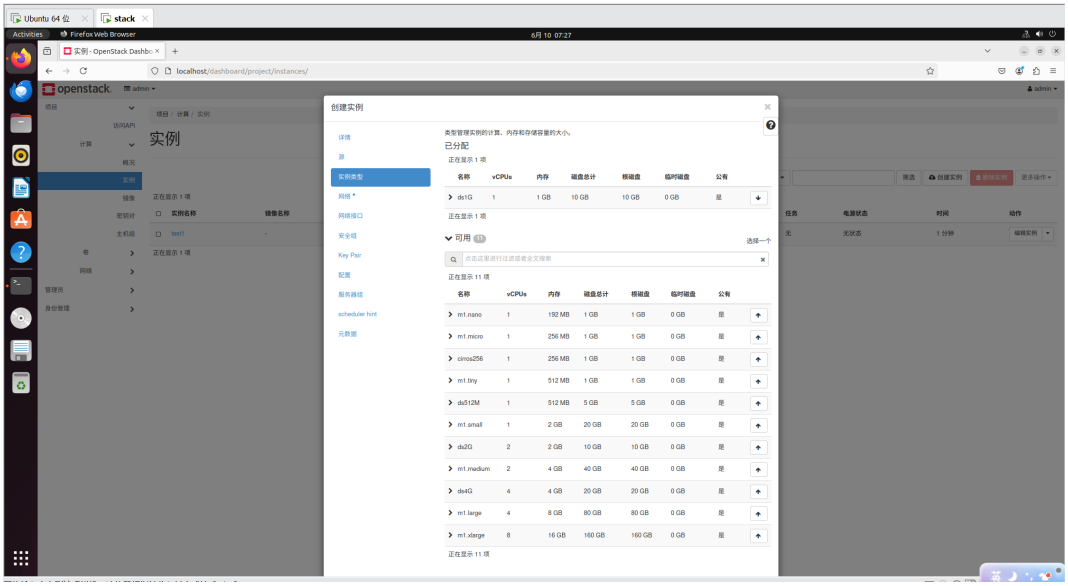

图 15: 存储卷配置

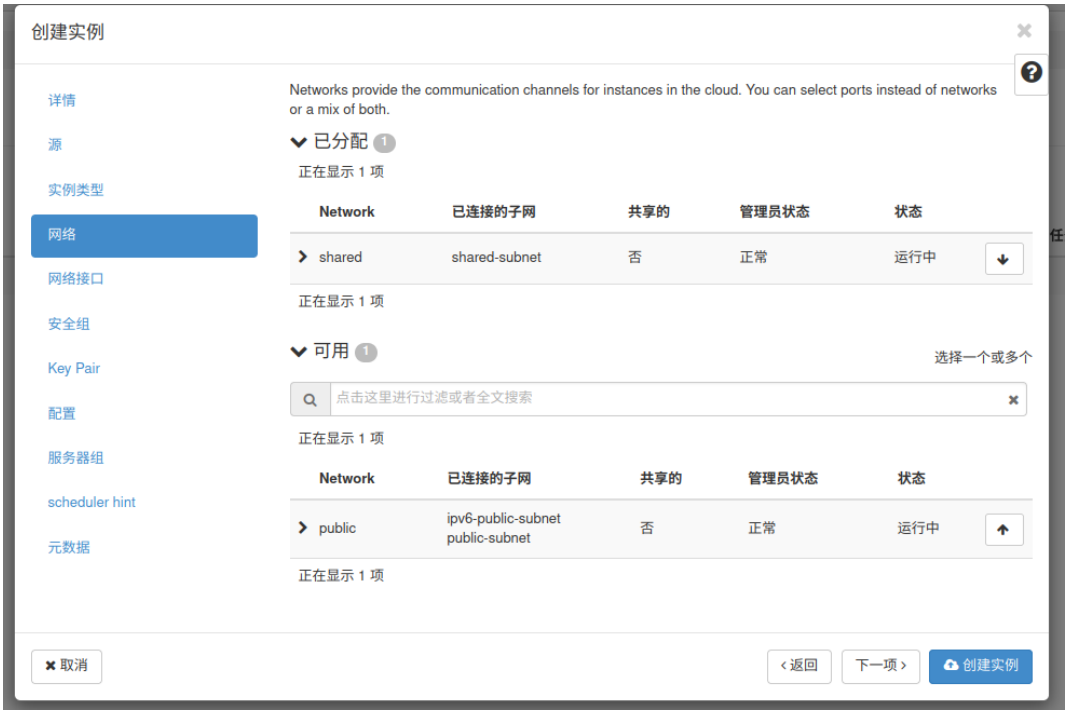

图 16: 网络配置

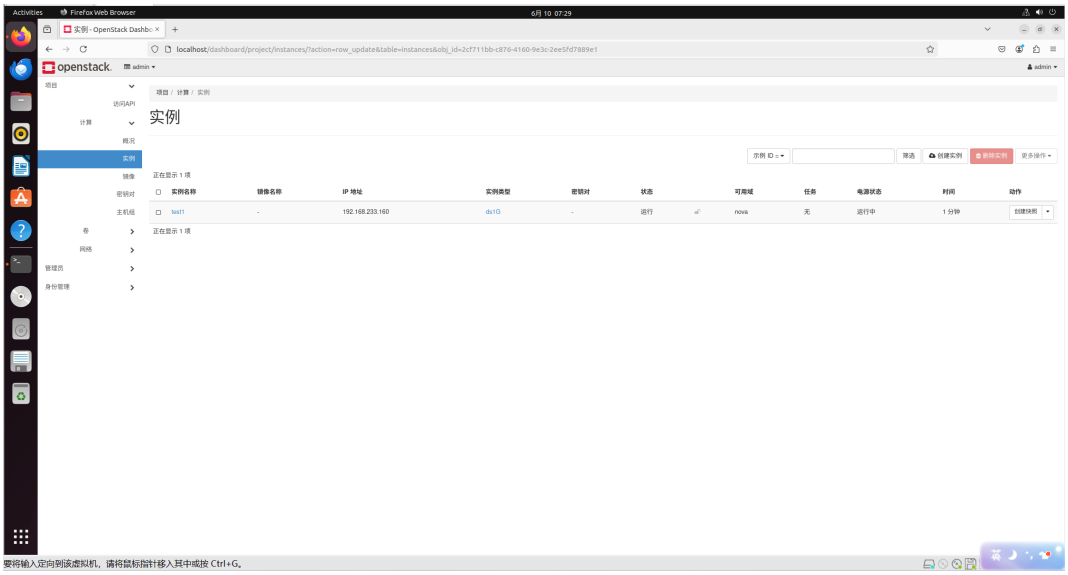

图 17: 实例创建成功

成功创建并运行实例，成功实现了 IaaS。

3 遇到的问题与解决方案

即使使用的是官方提供的一键安装脚本, 但也遇到了许多问题。在部署过程中遇到的主要问题及解决方法如下：

1. **Python 版本兼容问题**：Ubuntu24 默认的 python 版本过高，导致部分包无法拉取。且无法降低直接降低 python 版本 (和系统强绑定)，一开始通过虚拟环境解决，但出现了问题 3。解决方案：选择降低 Ubuntu 版本到 22

2. **Apache2 启动失败**：apache2 启动失败，通过日志跟踪发现是配置文件存在问题，查看配置文件后发现不知为何多出来了很多乱码字符。解决方案：手动清理配置文件中的异常字符

3. **Keystone 服务异常**：keystone 启动失败，通过阅读日志，以及大量的上网搜索，最终觉得是手动配置用来降低 py 版本的 conda 虚拟环境和 devstack 自身同时创建的 venv 环境冲突导致的。解决方案：降低 Ubuntu 版本到 22，直接使用系统 Python 环境，避免多层虚拟环境

4. **实例创建失败**：部署成功后，由于对 OpenStack 本身不太了解，第一次创建实例时乱填参数导致实例创建失败，后经学习明白了 OpenStack 中的各项配置。解决方案：系统学习 OpenStack 资源配置模型后重新配置

4 总结与感悟

虽然一键安装脚本非常方便，但其中的各个步骤对于我们来说如同黑盒，一旦出现问题，就需要大量的时间阅读日志、报错信息等进行跟踪和定位。不过在此过程中，我们学习到了很多问题修复的经验，也增长了对 OpenStack 所使用的这些中间件的了解。

通过这次对 OpenStack 的配置，以及配置完成后对 OpenStack 的使用，让我们小组对 IaaS 有了更深入的认识。同时我们对复杂集群管理平台的搭建难度有了一定的认知，如此小规模的一体化简单测试环境的搭建就遇到了诸多问题，消耗了很多时间，那搭建更加完善庞大的生产环境所需要的人力资源肯定会更加多。