

# 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。

```
attackstack:~$ sudo cp -a /etc/apt/sources.list /etc/apt/sources.list.bak
sudo sed -i 's@http://archive.ubuntu.com@http://repo.huaweicloud.com@' /etc/apt/sources.list
sudo sed -i 's@http://security.ubuntu.com@http://repo.huaweicloud.com@' /etc/apt/sources.list
sudo apt-get update
[sudo] password for stack:
Get:1 http://repo.huaweicloud.com/ubuntu jammy InRelease [270 kB]
Get:2 http://repo.huaweicloud.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://repo.huaweicloud.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://repo.huaweicloud.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://repo.huaweicloud.com/ubuntu jammy/main amd64 Packages [11,395 kB]
Get:6 http://repo.huaweicloud.com/ubuntu jammy/main i386 Packages [11,040 kB]
Get:7 http://repo.huaweicloud.com/ubuntu jammy/main Translation-en [510 kB]
Get:8 http://repo.huaweicloud.com/ubuntu jammy/main amd64 DEP-11 Metadata [423 kB]
Get:9 http://repo.huaweicloud.com/ubuntu jammy/main DEP-11 48x48 Icons [100.0 kB]
Get:10 http://repo.huaweicloud.com/ubuntu jammy/main DEP-11 64x64 Icons [148 kB]
Get:11 http://repo.huaweicloud.com/ubuntu jammy/main DEP-11 64x64 Icons [15.8 kB]
Get:12 http://repo.huaweicloud.com/ubuntu jammy/main amd64 c-n-f Metadata [30.3 kB]
Get:13 http://repo.huaweicloud.com/ubuntu jammy/restricted i386 Packages [30.4 kB]
Get:14 http://repo.huaweicloud.com/ubuntu jammy/restricted amd64 Packages [129 kB]
Get:15 http://repo.huaweicloud.com/ubuntu jammy/restricted Translation-en [18.6 kB]
Get:16 http://repo.huaweicloud.com/ubuntu jammy/restricted amd64 c-n-f Metadata [488 B]
Get:17 http://repo.huaweicloud.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:18 http://repo.huaweicloud.com/ubuntu jammy/universe i386 Packages [7,474 kB]
Get:19 http://repo.huaweicloud.com/ubuntu jammy/universe Translation-en [5,652 kB]
Get:20 http://repo.huaweicloud.com/ubuntu jammy/universe amd64 DEP-11 Metadata [3,559 kB]
Get:21 http://repo.huaweicloud.com/ubuntu jammy/universe DEP-11 48x48 Icons [3,447 kB]
Get:22 http://repo.huaweicloud.com/ubuntu jammy/universe DEP-11 64x64 Icons [7,609 kB]
Get:23 http://repo.huaweicloud.com/ubuntu jammy/universe DEP-11 64x64 Icons [69.3 kB]
Get:24 http://repo.huaweicloud.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:25 http://repo.huaweicloud.com/ubuntu jammy/multiverse i386 Packages [112 kB]
Get:26 http://repo.huaweicloud.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:27 http://repo.huaweicloud.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:28 http://repo.huaweicloud.com/ubuntu jammy/multiverse amd64 DEP-11 Metadata [42.1 kB]
Get:29 http://repo.huaweicloud.com/ubuntu jammy/multiverse DEP-11 48x48 Icons [42.7 kB]
Get:30 http://repo.huaweicloud.com/ubuntu jammy/multiverse DEP-11 64x64 Icons [193 kB]
Get:31 http://repo.huaweicloud.com/ubuntu jammy/multiverse DEP-11 64x64 Icons [214 B]
Get:32 http://repo.huaweicloud.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [5,372 B]
Get:33 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 Packages [2,622 kB]
Get:34 http://repo.huaweicloud.com/ubuntu jammy-updates/main i386 Packages [815 kB]
Get:35 http://repo.huaweicloud.com/ubuntu jammy-updates/main Translation-en [423 kB]
Get:36 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [114 kB]
Get:37 http://repo.huaweicloud.com/ubuntu jammy-updates/main DEP-11 48x48 Icons [40.8 kB]
Get:38 http://repo.huaweicloud.com/ubuntu jammy-updates/main DEP-11 64x64 Icons [62.0 kB]
Get:39 http://repo.huaweicloud.com/ubuntu jammy-updates/main DEP-11 64x64 Icons [29 B]
```

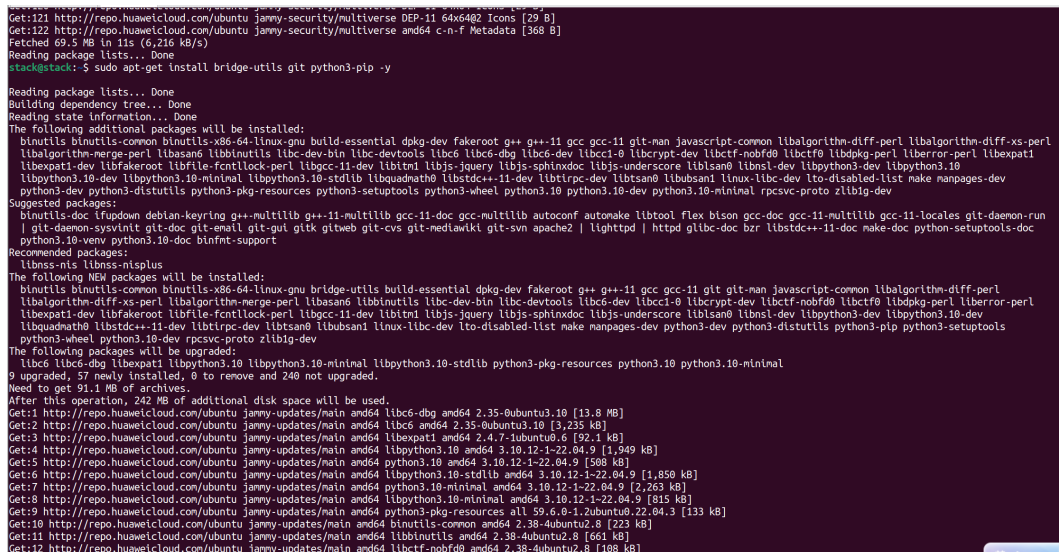
图 1: apt 源更换配置过程

## 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。



```
Get:121 http://repo.huaweicloud.com/ubuntu jammy-security/multiverse DEP-11 64x6402 Icons [29 B]
Get:122 http://repo.huaweicloud.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [368 B]
Fetched 69.5 MB in 11s (6.216 kB/s)
Reading package lists... Done
stack@stack:~$ sudo apt-get install bridge-utils git python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 git-man javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6 libc6-dbg libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl liberror-perl libexpat1
  libfakeroot libfile-fcntllock-perl libgcc-11-dev libitm1 libjvarkit libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libnl-dev libpython3-dev libpython3.10
  libpython3.10-dev libpython3.10-minimal libpython3.10-stdlib libquadmath0 libstdc++-11-dev libtirpc-dev libubsan0 libubsan1 linux-libc-dev lto-disabled-list make manpages-dev
  python3-dev python3-distutils python3-pkg-resources python3-setuptools python3-wheel python3.10 python3.10-minimal rpcsvc-proto zlib1g-dev
Suggested packages:
  binutils-doc ifupdown debian-keyring g++-multilib gcc-11-doc gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-11-multilib gcc-11-locales git-daemon-run
  | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn apache2 | lighttpd | httpd glibc-doc bzip2 libstdc++-11-doc make-doc python3-setuptools-doc
Recommended packages:
  libssl-mt1 libssl-mtplus
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu bridge-utils build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 git git-man javascript-common libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl liberror-perl
  libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-11-dev libitm1 libjvarkit libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libnl-dev libpython3-dev libpython3.10-dev
  libquadmath0 libstdc++-11-dev libtirpc-dev libubsan0 libubsan1 linux-libc-dev lto-disabled-list make manpages-dev python3-dev python3-distutils python3-pkg-resources python3-setuptools
  python3-wheel python3.10 python3.10-dev rpcsvc-proto zlib1g-dev
The following packages will be upgraded:
  libc6 libc6-dbg libpython3.10 libpython3.10-minimal libpython3.10-stdlib python3-pkg-resources python3.10 python3.10-minimal
9 upgraded, 57 newly installed, 0 to remove and 240 not upgraded.
Need to get 91.1 MB of archives.
After this operation, 242 MB of additional disk space will be used.
Get:1 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libc6-dbg amd64 2.35-0ubuntu3.10 [13.8 MB]
Get:2 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libc6 amd64 2.35-0ubuntu3.10 [3,235 kB]
Get:3 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libexpat1 amd64 2.4.7-1ubuntu0.6 [92.1 kB]
Get:4 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libpython3.10 amd64 3.10.12-1-22.04.9 [1,949 kB]
Get:5 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 python3.10 amd64 3.10.12-1-22.04.9 [508 kB]
Get:6 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libpython3.10-stdlib amd64 3.10.12-1-22.04.9 [1,850 kB]
Get:7 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 python3.10-minimal amd64 3.10.12-1-22.04.9 [2,263 kB]
Get:8 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libpython3.10-minimal amd64 3.10.12-1-22.04.9 [815 kB]
Get:9 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 python3-pkg-resources all 59.6.0-1.2ubuntu0.22.04.3 [133 kB]
Get:10 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 binutils-common amd64 2.38-4ubuntu2.8 [223 kB]
Get:11 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libbinutils amd64 2.38-4ubuntu2.8 [661 kB]
Get:12 http://repo.huaweicloud.com/ubuntu jammy-updates/main amd64 libctf-nobfd0 amd64 2.38-4ubuntu2.8 [108 kB]
```

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。

```
stack@stack:~$ git clone https://opendev.org/openstack/devstack
Cloning into 'devstack'...
remote: Enumerating objects: 51756, done.
remote: Counting objects: 100% (31354/31354), done.
remote: Compressing objects: 100% (18592/18592), done.
remote: Total 51756 (delta 30587), reused 20762 (delta 20762), pack-reused 20402
Receiving objects: 100% (51756/51756), 9.76 MiB | 661.00 KiB/s, done.
Resolving deltas: 100% (36743/36743), done.
stack@stack:~$ cd devstack/
stack@stack:~/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。

```
stack@stack:~/devstack$ cd ..
stack@stack:~$ sudo mkdir .pip
stack@stack:~$ sudo vim .pip/pip.conf
stack@stack:~$
```

图 4: pip 配置文件创建与修改

```
[global]
index-url = https://repo.huaweicloud.com/repository/pypi/simple
trusted-host = repo.huaweicloud.com
```

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。

```
[[local|localrc]]
ADMIN_PASSWORD=12345678
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

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
+ unset 822
+ PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/local/bin:/usr/sbin:/usr/bin
+++ dirname ./stack.sh
++ cd .
++ pwd
+ TOP_DIR=/home/stack/devstack
+ NOUNSET=
+ [[ -n '' ]]
++ date +%s
+ DEVSTACK_START_TIME=1749500100
+ [[ -r /home/stack/devstack/.stackenv ]]
+ FILES=/home/stack/devstack/files
+ '[' '!' -d /home/stack/devstack/files ']'
+ '[' '!' -d /home/stack/devstack/lib ']'
+ '[' '!' -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。

```
2025-06-10 23:09:11.374 | stack.sh complete
stack@stack:~/devstack$ vim ~/.bashrc
stack@stack:~/devstack$ source ~/.bashrc
stack@stack:~/devstack$
```

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 {Namespaces 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 = 35.834921ms, 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, res=

● 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-e1072969c41f serv
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 serv
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 serv
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 serv
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

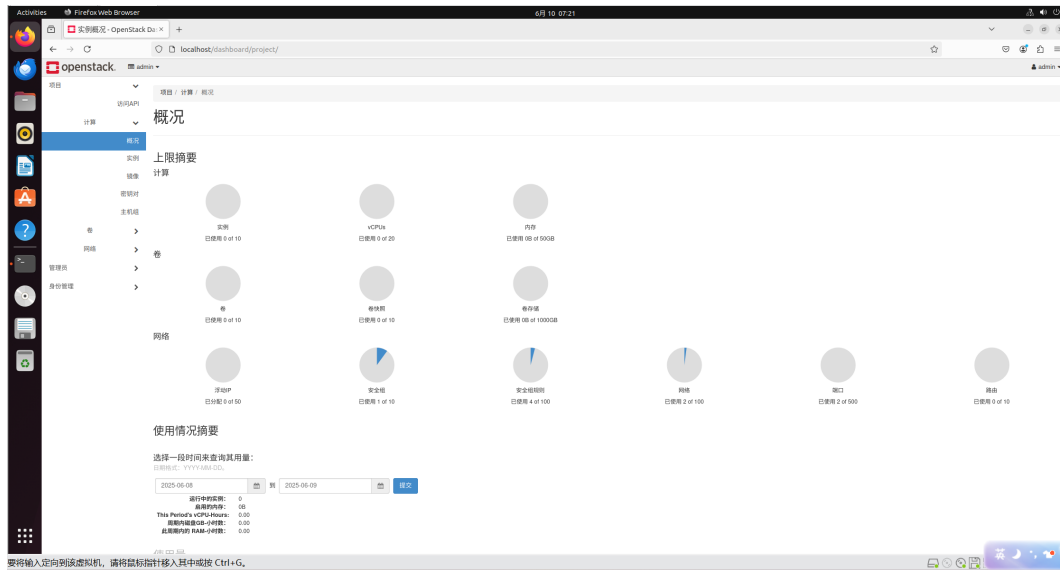


图 12: OpenStack 中文控制面板

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:

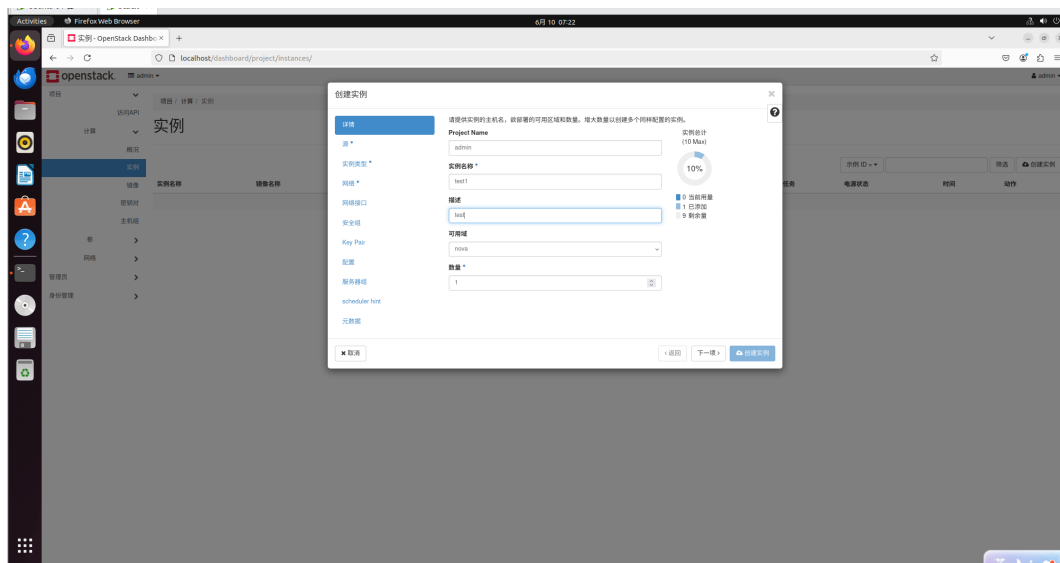


图 13: 创建新实例



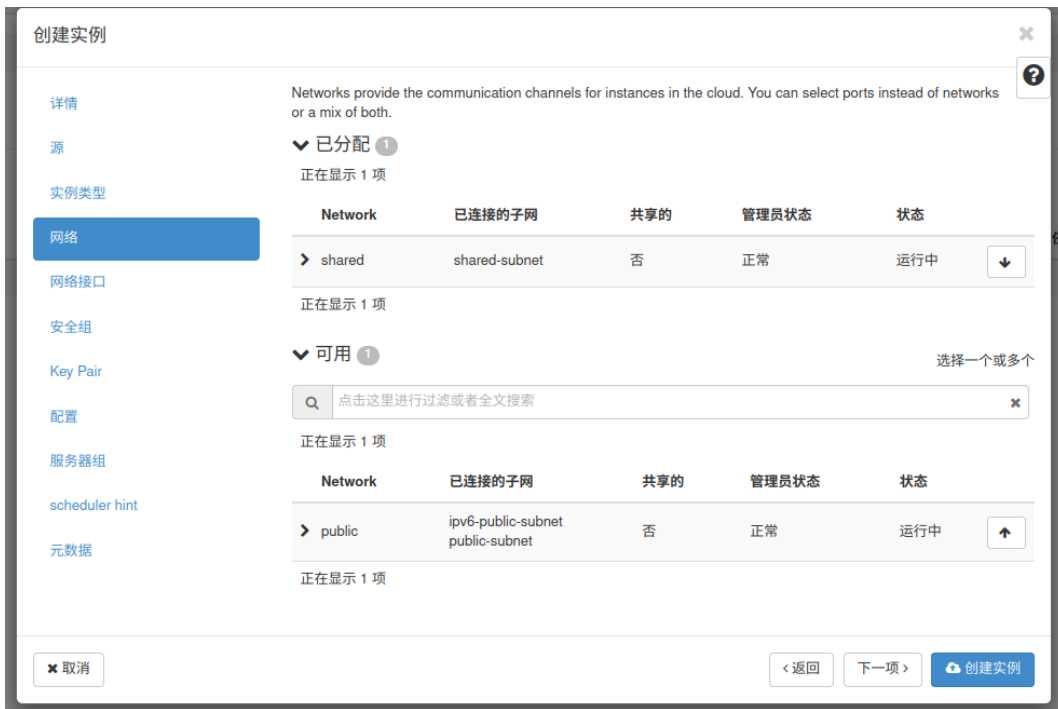


图 16: 网络配置

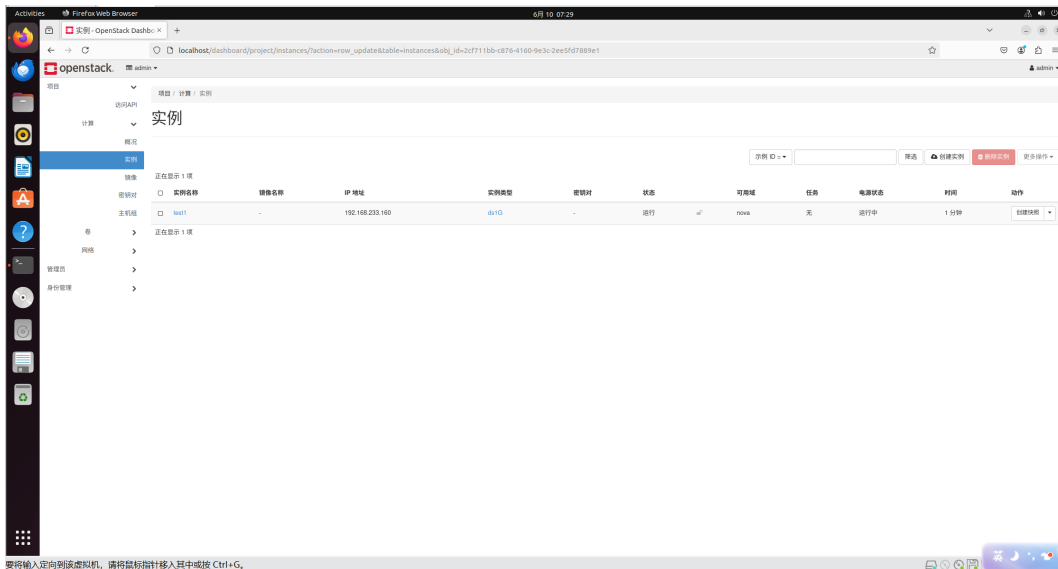


图 17: 实例创建成功

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。