

OPEN STACK

当今最流行的开源云平台管理项目

组员：XXXXXXXX

背景

OpenStack 是一个非常年轻的开源项目，最初是由美国国家航空航天局（NASA）和Rackspace 合作研发的项目，2010 年7 月以Apache 2.0 许可证授权开源，源代码来自于NASA 的Nebula 云平台 and Rackspace 的分布式云存储（Swift）项目。

NASA 最初使用的是Eucalyptus 云计算平台，当规模持续快速增长后，Eucalyptus 已经不能满足NASA 的云计算规模，NASA 首席技术官Chris Kemp 的研究小组为此专门建立了自己的计算引擎，新平台命名为Nova，并将其开源。



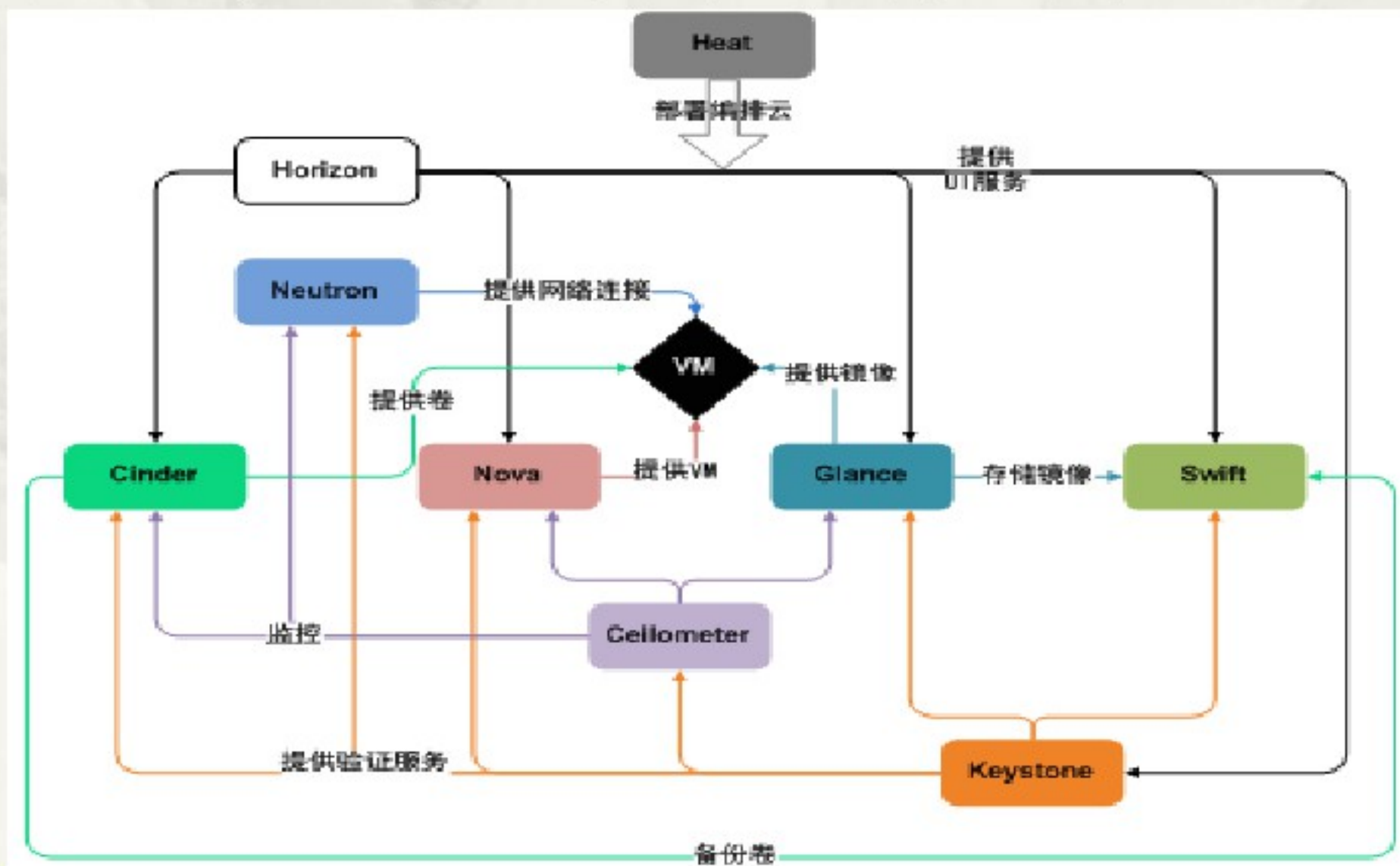
核心项目

- 计算 (Compute) : Nova
- 对象存储 (Object Storage) : Swift
- 镜像服务 (Image Service) : Glance
- 身份服务 (Identity Service) : Keystone
- 网络&地址管理 (Network) : Neutron
- 块存储 (Block Storage) : Cinder
- UI 界面 (Dashboard) : Horizon
- 测量 (Metering) : Ceilometer
- 部署编排 (Orchestration) : Heat
- 数据库服务 (Database Service) : Trove

环境架构

OpenStack因Open而开放，因组件而灵活，因包容而博大。有计算、网络、对象存储、块存储、身份、镜像服务、门户、测量、部署编排、数据库服务等等组件，有的组件可以根据需要选择安装，组网结构也很灵活、多样。实现了支持接入多种主流虚拟机软件：KVM、LXC、QEMU、Hyper-V、VMware、XenServer，也可以自行开发插件接入其他的虚拟化软件。

下图展示了OpenStack典型环境架构，各个服务之间的交互和职能。



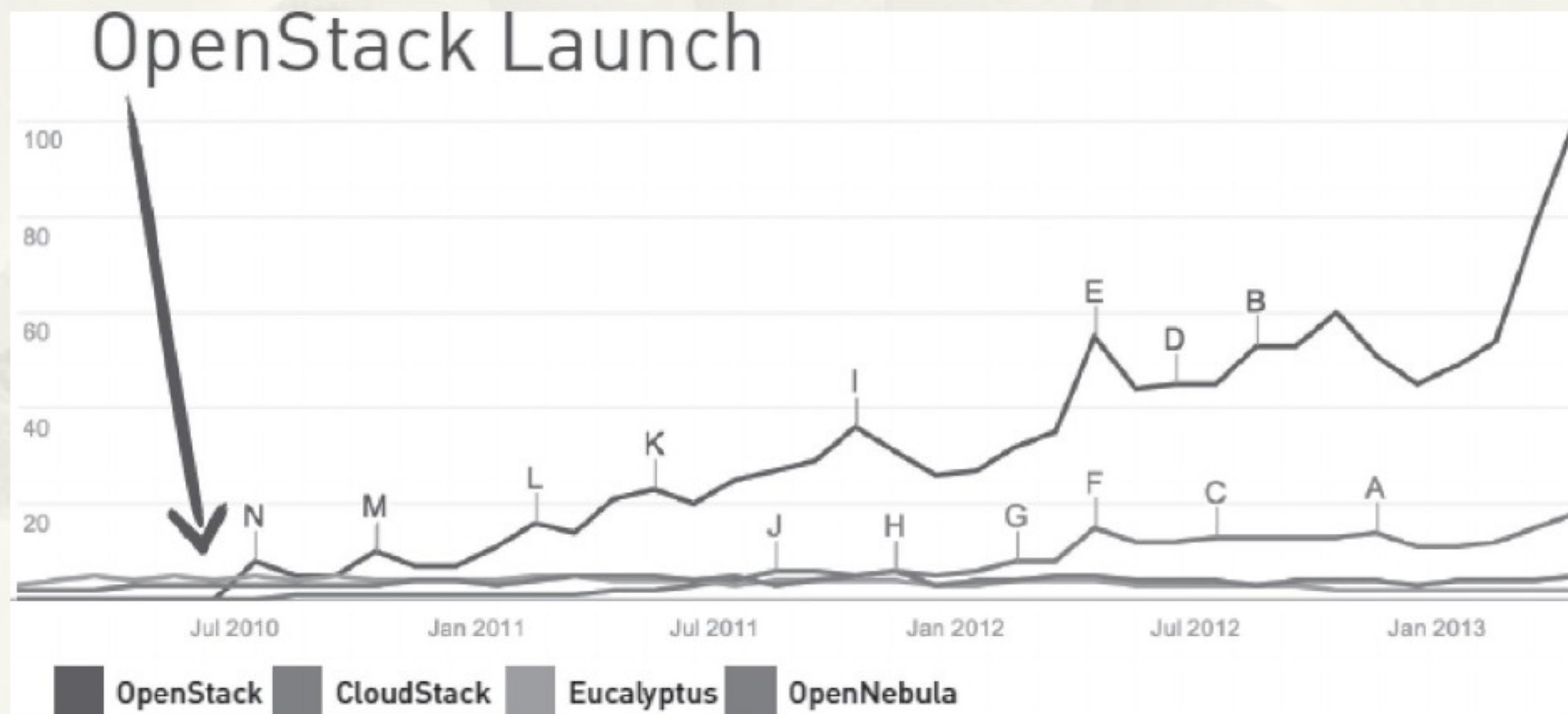
OPEN STACK 功能

OpenStack 具有三大核心功能，即计算、存储、网络，分别对应相应的项目Nova、Cinder、Neutron。其中Nova 提供了计算资源的管理，可以管理跨服务器网络的VM实例。同时，Nova 还提供对多种Hypervisor 的支持，如KVM、QEMU、Xen、LXC、VMware、Hyper-V、PowerVM 等。Cinder 提供了存储资源的管理，可以管理各个厂商提供的专业存储设备。Neutron 提供了网络资源的管理，并且LBaaS、FWaaS 等一系列网络相关的组件也正在逐渐发展起来。



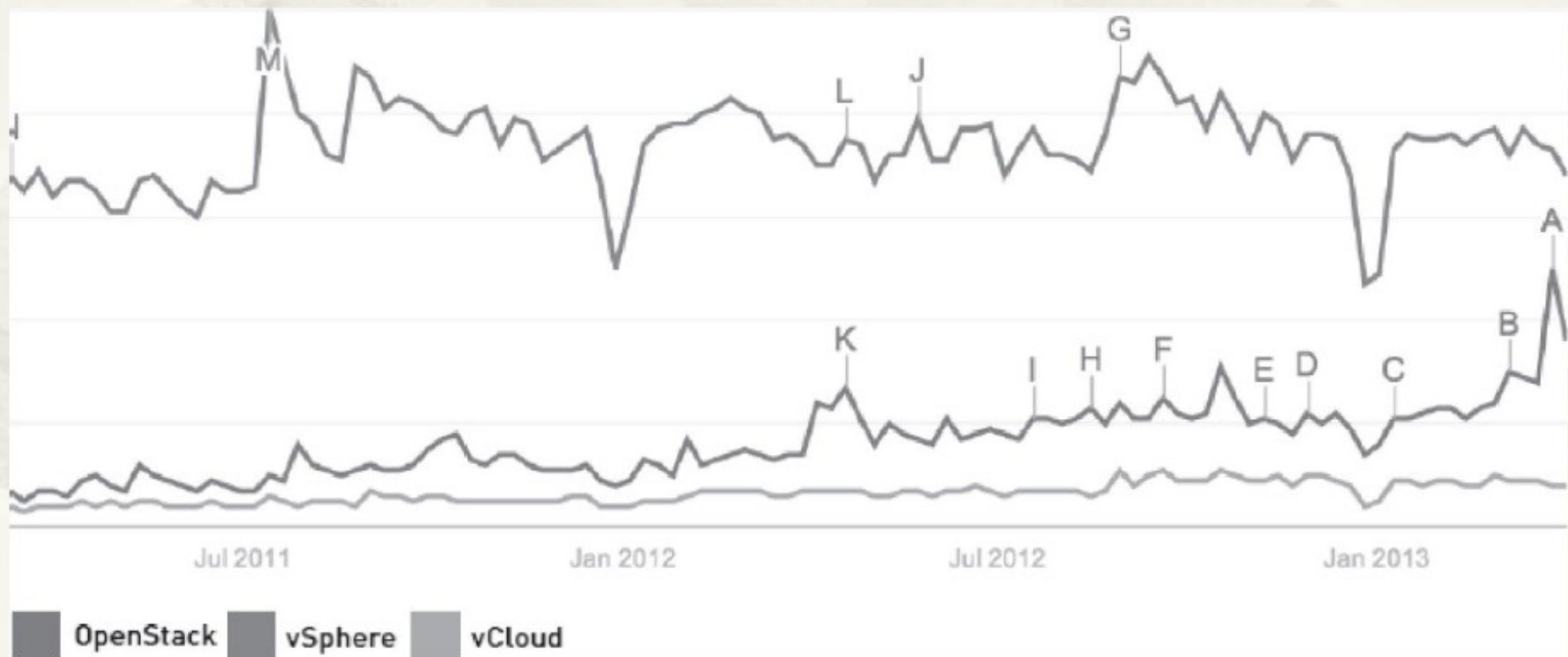
应用现状和发展趋势

在开源云计算项目领域，OpenStack 从2010 年开始就已经超过 CloudStack、Eucalyptus、OpenNebula 等其他云计算开源项目，是当今最热门的开源项目之一，这离不开社区管理者和社区推广者的努力。



应用现状和发展趋势

在云计算领域，OpenStack 也在逐渐追赶虚拟化商业巨头VMware 的步伐。OpenStack 和其他商业云项目的趋势对比如图所示。我们有理由相信，在今后几年乃至相当长一段时间里，OpenStack依然会活跃在大家的视线中。



搭建工具

* Openstack的搭建工具有很多

Devstack: 这应该算是Openstack最早的安装脚本，他是通过直接git源码，进行安装

Diablo安装脚本: 听说有少许bug

Fuel: 这是Mirantis出品的部署安装工具，基本算是把Openstack所有的部署都web化，你可以有很多的选择：尤其是网络，存储。

Download Mirantis OpenStack

Mirantis OpenStack is distributed as a single commercially supported package in either ISO or IMG format. The 8.0 release features:

- Support for preprovisioned RHEL nodes
- Integration of Ironic, the bare metal provisioning project, for performance sensitive workloads
- Autoscaling IaaS and cloud bursting for Kubernetes container orchestration
- Ability for Cloud Foundry developers to consume Murano services to retain application portability
- Ability to spread OpenStack controllers across multiple racks using L3 networking
- Fuel enhancements including new equipment view of all your nodes, contextual help, ability to specify IP ranges, additional info about plugin endpoints including cues on which plugins have been tested to interoperate with each other

Mirantis OpenStack ISO

Choose ISO when booting from removable media, or mounting from DVD-ROM (physical or virtual). Choose 'Upgrade Package' if you're upgrading to 8.0.

[DOWNLOAD 8.0 ISO](#)

[VIRTUALBOX SCRIPTS](#)

[DOWNLOAD PRIOR RELEASES](#)

[CHECKSUM](#)

OPEN STACK

经过一番准备后，先在主节点上安装mirantis 6.0镜像，安装完成后根据提示信息登入fuel的webUI界面，在这里我们就可以实现将Openstack的所有部署

Fuel 6.0之OpenStack Ju... openstack部署与管理-f... 使用Fuel安装OpenStack... Releases - Mirantis Op... 使用Fuel安装OpenStack... 使用Fuel安装OpenStack... 打开新的标签页... Fuel Dashboa

20.0.2:8000/#welcome

Search - floodlight zy-sdn/savi-floodli... 极客学院IT在线教育... Floodlight project REST API 接口 SDN联合播报 | 中... Java视频教程_Java... 排行榜 - 文峰英语... SDN你必须知道的... Man page of dpctl

Welcome to Mirantis OpenStack

Register your installation and get free support!

Ensure that you receive important notifications about issues and updates. Register now and you will receive complimentary access to our basic support services for the next 30 days.

[Register Now](#)

Help us to improve your experience by sending Mirantis information about the settings, features, and deployment actions when you use Mirantis OpenStack.

Usage statistics include information such as settings, button/menu clicks, hardware configuration, and version information. The usage statistics do not include information such as passwords, ip addresses, or node names. For a complete list of statistics that we gather, [click here](#).

Mirantis' privacy policy ("Privacy Policy") describes our practices regarding the information we collect on the Mirantis web sites and through the use of our products and services, and how it is used and shared with third parties. You can read the policy [here](#).

Send usage statistics to Mirantis

Our support team can optionally use your error reports to assist you.

Identify my error reports so that Mirantis Support can assist me

[Start Using Fuel](#)

You can change your settings at any time by updating your user profile information.

Thanks for helping to improve Mirantis OpenStack!

OPEN STACK

The image shows a Fuel Dashboard interface for creating a new OpenStack environment. The main window is titled "新建OpenStack环境" (New OpenStack Environment). It features a sidebar with navigation options: "名称和版本" (Name and Version), "部署模式" (Deployment Mode), "计算" (Compute), "网络" (Network), "后端存储" (Backend Storage), "附加服务" (Additional Services), and "完成" (Finish). The "名称和版本" section is active, showing the name "云计算与物联网中期实验报告" (Cloud Computing and IoT Mid-term Report) and the OpenStack version "Juno on Ubuntu 12.04.4 (2014.2-6.0) (default)". A description states: "This option will install the OpenStack Juno packages using Ubuntu as a base operating system. With high availability features built in, you are getting a robust, enterprise-grade OpenStack deployment." Buttons for "取消" (Cancel), "后退" (Back), and "前进" (Next) are visible at the bottom.

Overlaid on the right are two terminal windows. The top one, titled "fuel_master", shows the boot process of the master node, including the default administrator password (rootme), login details, and the Fuel UI availability URL (http://10.20.0.2:8000). The bottom terminal window, titled "fuel_compute", shows the boot process of a compute node, displaying the Ubuntu version (Ubuntu 12.04.4 LTS) and the login prompt for the node-4 user.




分配CONTROLLER节点和COMPUTER节点

节点 网络 设置 日志 健康检查 动作 部署变更




分组 角色 Filter By Node name/mac 磁盘配置 网络配置 + 增加节点

选择全部

Controller (1) 选择全部

<input type="checkbox"/>		Untitled (1e:11) CONTROLLER		✓ 已就绪	CPU: 1 (1) HDD: 62.6 GB RAM: 1.4 GB	
--------------------------	-------------------------------------------------------------------------------------	--------------------------------	---------------------------------------------------------------------------------------	-------	-------------------------------------	---------------------------------------------------------------------------------------

Compute (1) 选择全部

<input type="checkbox"/>		Untitled (d7:a7) COMPUTE		✓ 已就绪	CPU: 1 (2) HDD: 62.6 GB RAM: 2.3 GB	
--------------------------	-------------------------------------------------------------------------------------	-----------------------------	---------------------------------------------------------------------------------------	-------	-------------------------------------	---------------------------------------------------------------------------------------

OPEN STACK

基础MAC地址: fa:16:3e:00:00:00

Neutron L3配置

Internal network CIDR: 192.168.111.0/24

Internal network gateway: 192.168.111.1

Floating IP ranges: 开始 172.16.0.130 结束 172.16.0.254

DNS Servers: 8.8.4.4 8.8.8.8

完成网络验证的4个步骤:

1. 每个节点启动测试监听帧
2. 每个节点发送802.1Q封装的UDP帧
3. 每个节点从其它节点注册测试帧
4. 在所有端口发送DHCP消息

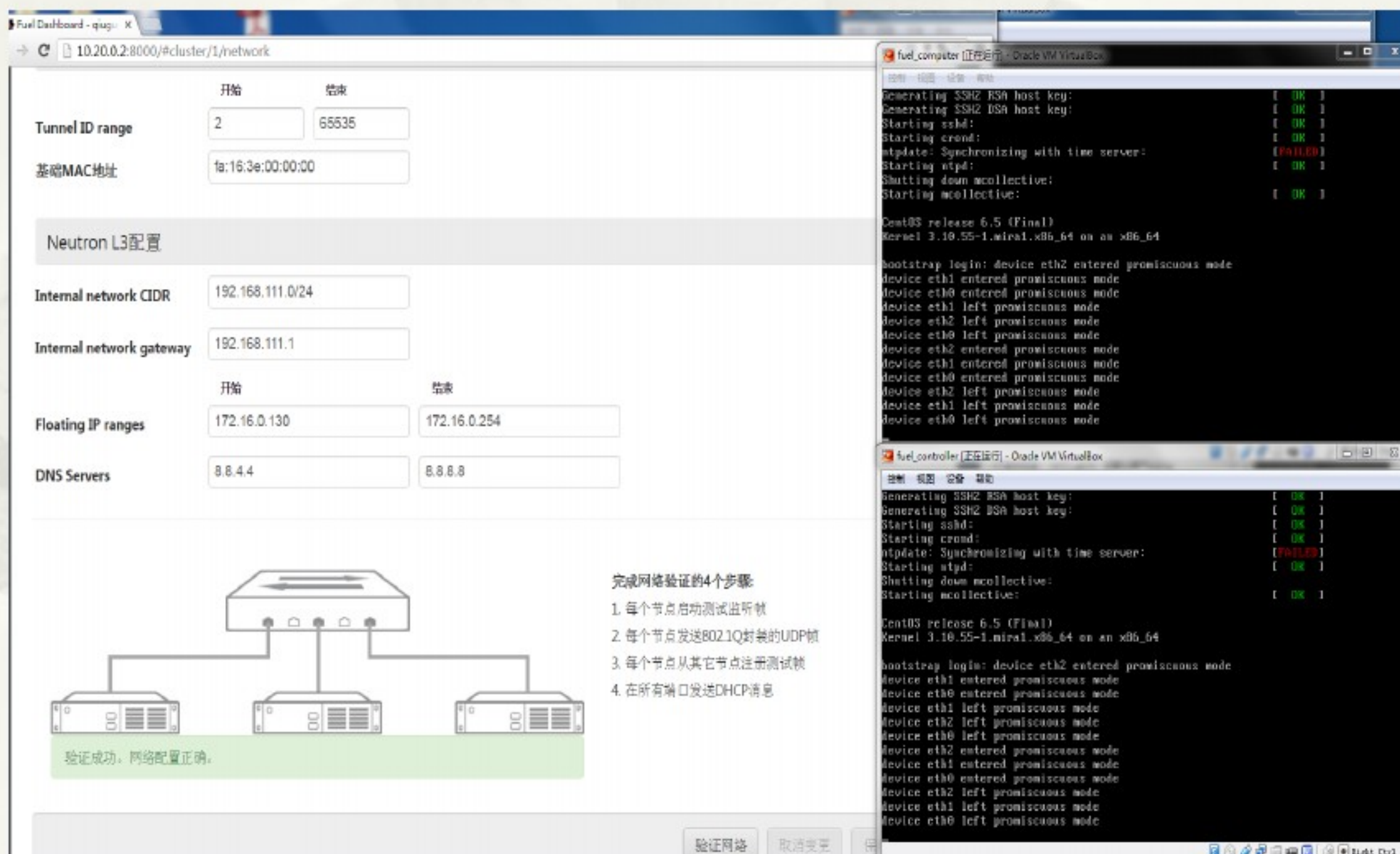
验证网络 取消变更 保存设置

MIRANTIS Copyright © 2013-2014 Mirantis. All rights reserved. Version 6.0

开始是在VM workstation 虚拟机上搭建，由于到了验证三个节点的网段能否通信这一块一直不能通过，后面查阅相关资料有人建议最好还是在Visual BOX 上搭建。

OPEN STACK

- * 然后又重新在Vbox上搭建了，终于验证通过



The screenshot displays the Fuel Dashboard network configuration interface and two terminal windows. The dashboard shows the following configuration:

- Tunnel ID range: 2 to 65535
- 基础MAC地址: fa:16:3e:00:00:00
- Neutron L3配置
- Internal network CIDR: 192.168.111.0/24
- Internal network gateway: 192.168.111.1
- Floating IP ranges: 172.16.0.130 to 172.16.0.254
- DNS Servers: 8.8.4.4 and 8.8.8.8

Below the configuration is a network diagram showing three nodes connected to a central switch. A green bar at the bottom of the diagram indicates "验证成功，网络配置正确。" (Verification successful, network configuration correct).

Next to the diagram, the steps for network verification are listed:

完成网络验证的4个步骤

1. 每个节点启动测试出听帧
2. 每个节点发送802.1Q封装的UDP帧
3. 每个节点从其它节点注册测试帧
4. 在所有端口发送DHCP消息

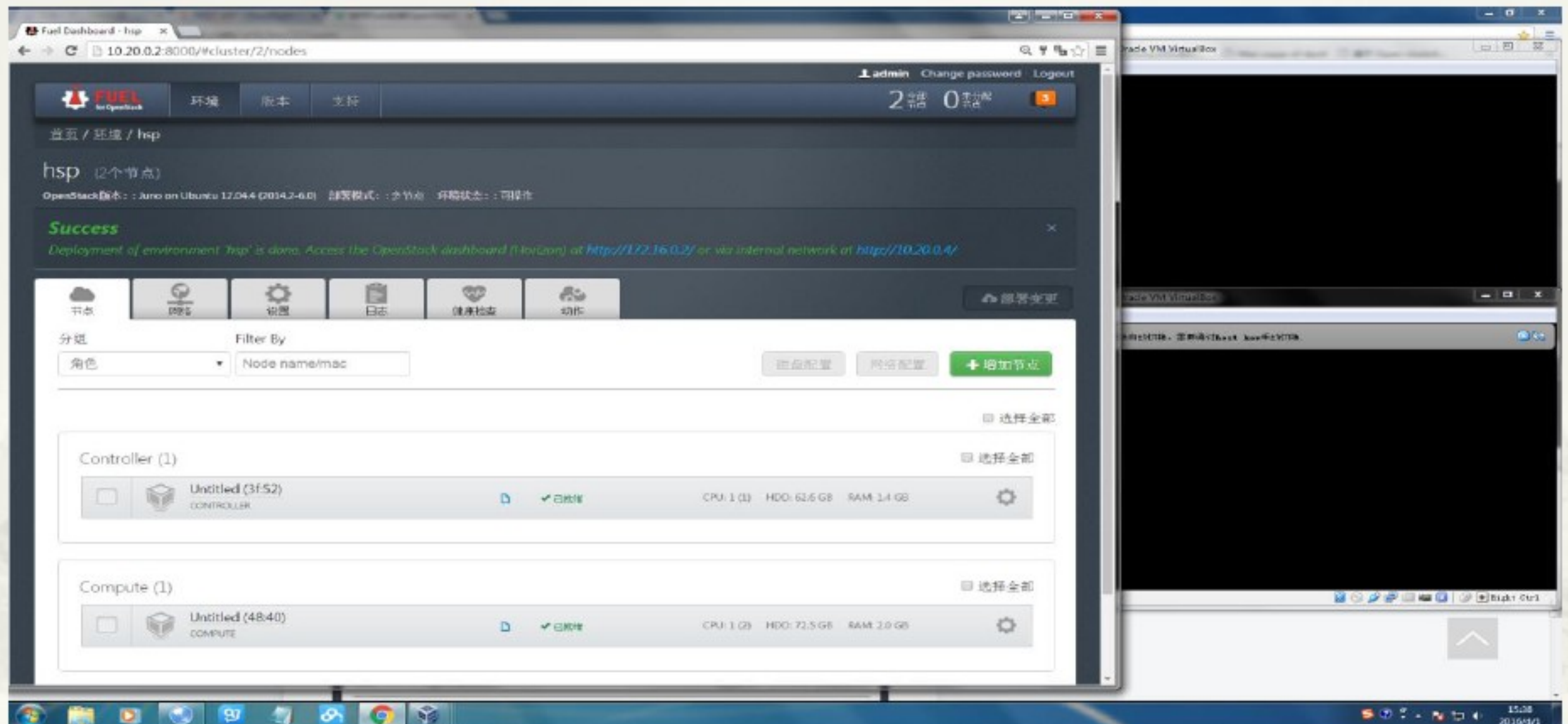
The terminal windows show the execution of these steps on a fuel_computer and fuel_controller, with status indicators like [OK] and [FAILED] for various operations such as generating SSH2 keys, starting services, and entering promiscuous mode on network interfaces.

OPEN STACK

然后便可以使用fuel的一键部署功能自动给两个节点进行应用部署了

The screenshot displays the Fuel Dashboard interface for an OpenStack environment named 'hhh'. The browser address bar shows '10.20.0.2:8000/#cluster/4/nodes'. The top navigation bar includes the Fuel logo, environment name 'hhh', and user information 'admin'. A summary bar indicates '4 全部节点' (4 total nodes) and '0 未部署节点' (0 un-deployed nodes). The main content area shows the deployment progress at 0%. Below this, there are navigation tabs for '节点' (Nodes), '网络' (Network), '设置' (Settings), '日志' (Logs), '健康检查' (Health Check), and '动作' (Actions). The '节点' tab is active, showing a list of nodes grouped by role. The 'Controller (1)' group contains one node 'Untitled (1e:11) CONTROLLER' with a progress bar for '正在安装CENTOS' (Installing CentOS) and specifications: CPU: 0 (1), HDD: 62.6 GB, RAM: 1.4 GB. The 'Compute (1)' group contains one node 'Untitled (d7:a7) COMPUTE' with a progress bar for '正在安装CENTOS' (Installing CentOS) and specifications: CPU: 1 (2), HDD: 62.6 GB, RAM: 2.4 GB. The interface also includes filters for '角色' (Role) and 'Node name/mac', and buttons for '磁盘配置' (Disk Config), '网络配置' (Network Config), and '+ 增加节点' (Add Node).

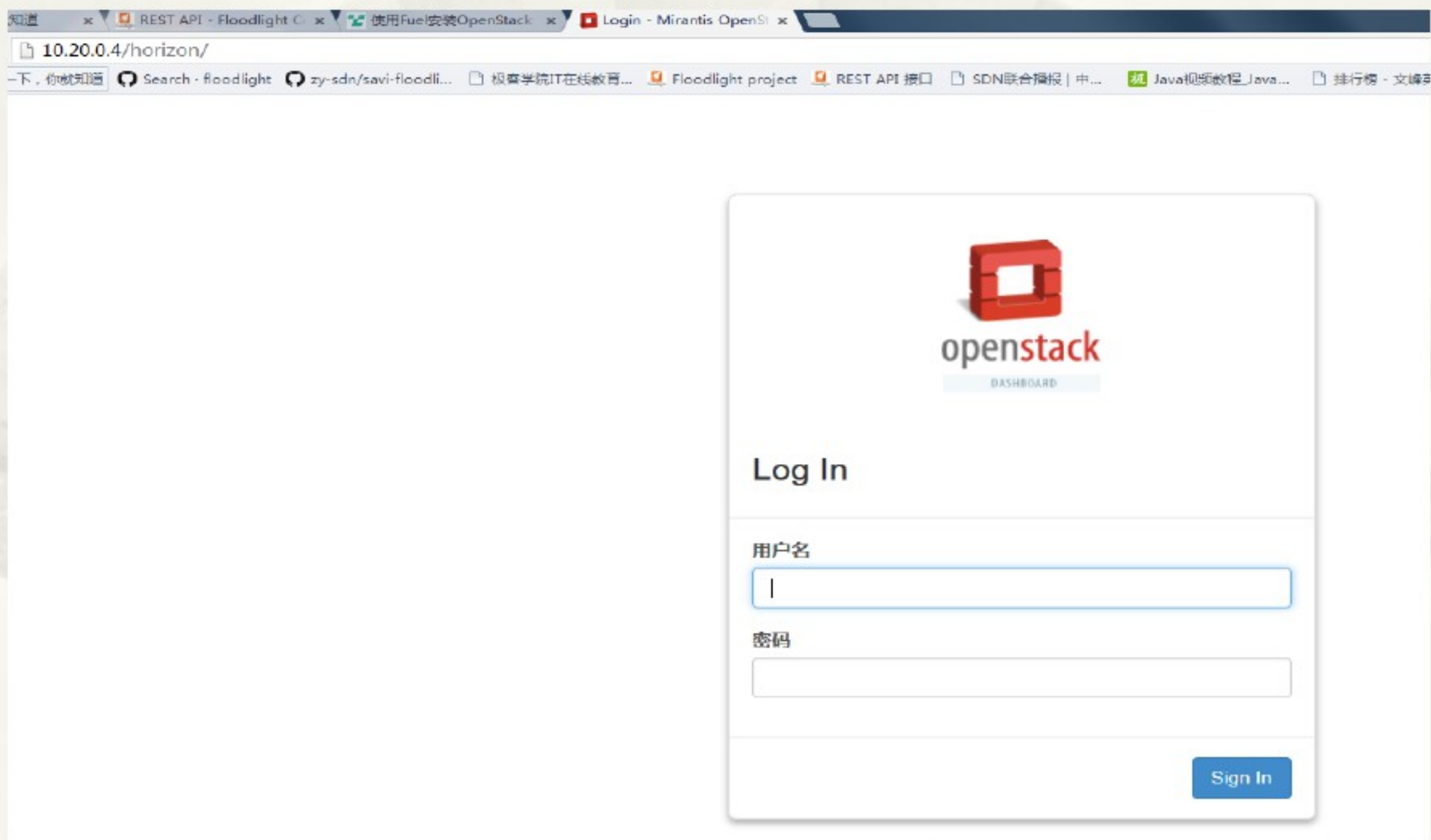
OPEN STACK



Deployment of environment 'hsp' is done. Access the OpenStack dashboard (Horizon) at <http://172.16.0.2/> or via internal network at <http://10.20.0.4/>

OPEN STACK

* 进入openstack的Horizon (dashboard) 管理界面



The image shows a browser window with the URL `10.20.0.4/horizon/`. The page displays the OpenStack logo and the text "openstack DASHBOARD". Below the logo, there is a "Log In" section with two input fields: "用户名" (Username) and "密码" (Password). A blue "Sign In" button is located at the bottom right of the login form.

浏览器地址栏: 10.20.0.4/horizon/

浏览器标签页: REST API - Floodlight, 使用Fuel安装OpenStack, Login - Mirantis OpenStack

浏览器书签: 一下, 你就知道, Search - floodlight, zy-sdn/savi-floodli..., 极客学院IT在线教育..., Floodlight project, REST API 接口, SDN联合播报 | 中..., Java视频教程_Java..., 排行榜 - 文博

OpenStack Logo

openstack
DASHBOARD

Log In

用户名

密码

Sign In

OPEN STACK

The screenshot shows the OpenStack Horizon admin interface. The browser address bar indicates the URL `10.20.0.4/horizon/admin/hypervisors/`. The page title is "All Hypervisors". The left sidebar contains navigation links for Project, Admin, System, Overview, Hypervisors (selected), Host Aggregates, Instances, Volumes, Flavors, Images, Networks, and Routers. The main content area shows a "Hypervisor Summary" with three circular progress indicators: VCPU Usage (Used 0 of 2), Memory Usage (Used 512MB of 2GB), and Disk Usage (Used 0Bytes of 52GB). Below the summary are two tabs: "Hypervisor" and "Compute Host". The "Hypervisors" tab is active, displaying a table with the following data:

Hostname	Type	VCPUs (used)	VCPUs (total)	RAM (used)	RAM (total)	Storage (used)	Storage (total)	Instances
node-4.domain.tld	QEMU	0	2	512MB	2GB	0Bytes	52GB	0

Below the table, it says "Displaying 1 item".

OPEN STACK

The image displays the Fuel Dashboard login interface and a terminal window. The dashboard shows the Fuel logo and a login form with fields for username (admin) and password (masked with dots), and a 'Log in' button. The terminal window shows the fuel master configuration, including the default Fuel UI password (admin), the root password change prompt, and the default administrator login (root) and password (r00tmc).

Fuel Dashboard - Log In
10.20.0.2:8000/#login

FUEL
for OpenStack®

admin

.....

Log in

Copyright © 2013-2014 Mirantis. All rights reserved.
Version: 6.0

fuel_master [正在运行] - Oracle VM VirtualBox

```
控制 视图 设备 帮助
Default Fuel UI password: admin
Please change root password on first login.
fuel login: root
Password:
Last login: Wed Apr 27 02:00:25 on tty1
root@fuel: ~# logout
#####
#       Welcome to the Fuel server       #
#####
Server is running on x86_64 platform

Fuel UI is available on: http://10.20.0.2:8000

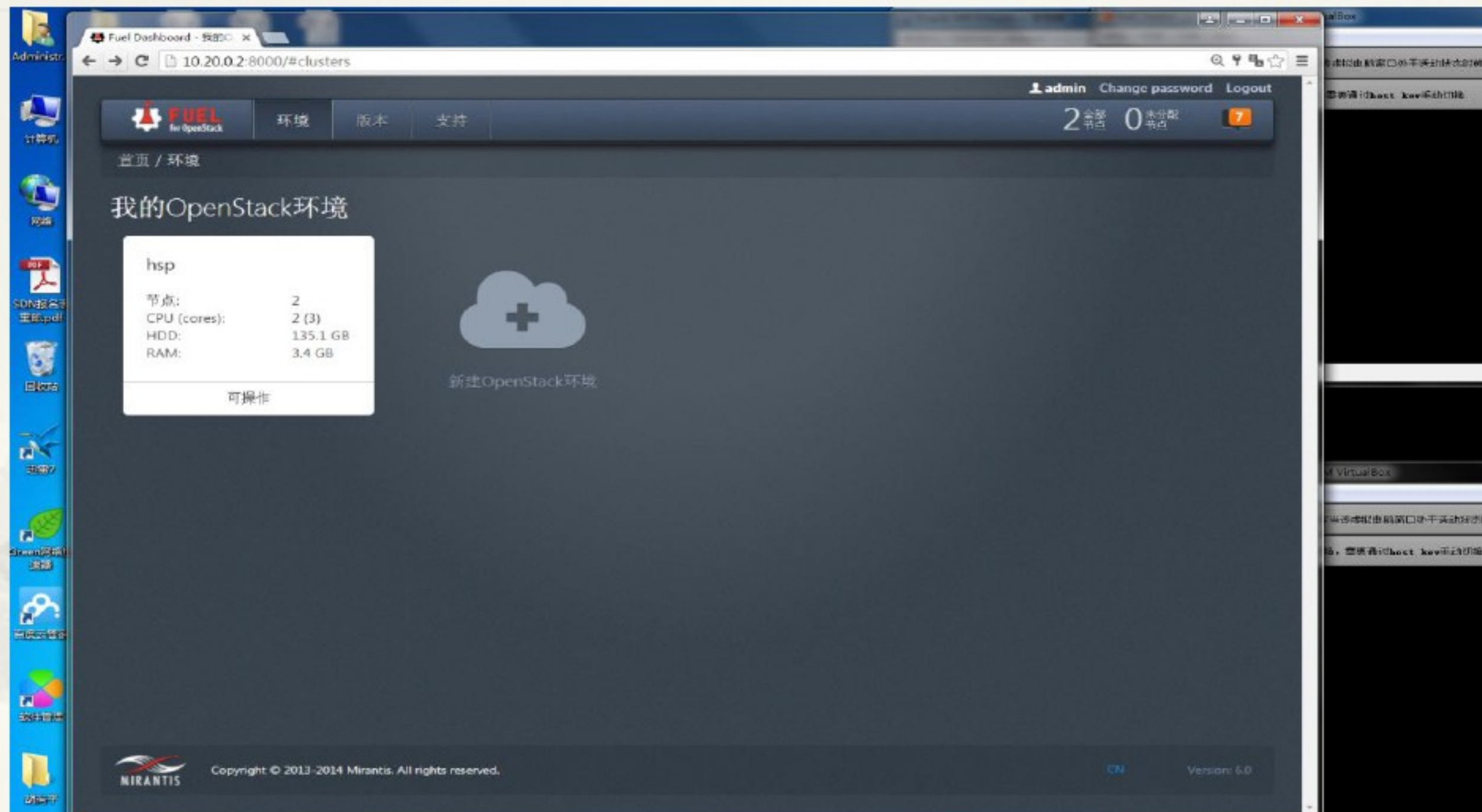
Default administrator login:  root
Default administrator password: r00tmc

Default Fuel UI login: admin
Default Fuel UI password: admin

Please change root password on first login.
fuel login: _
```

start, please edit /etc/default/puppet to enable [OK]
tpd [OK]
s... [OK]

OPEN STACK

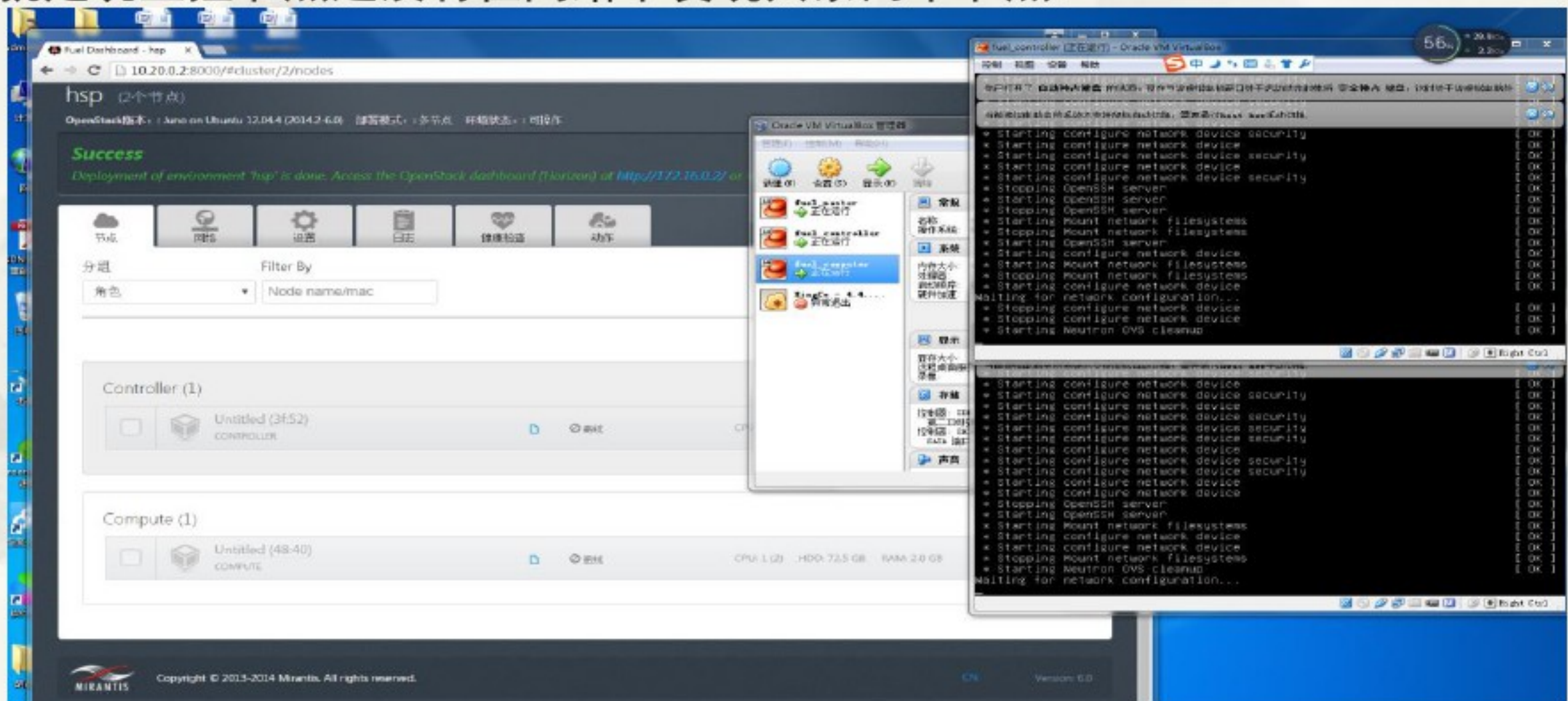


这是我们之前创建的一个openstack的环境。

点击新建，可以再配置和创建一个openstack的环境，网上关于如何部署和创建openstack环境的介绍也有不少，具体步骤我就不一一赘(zhu i)述了。

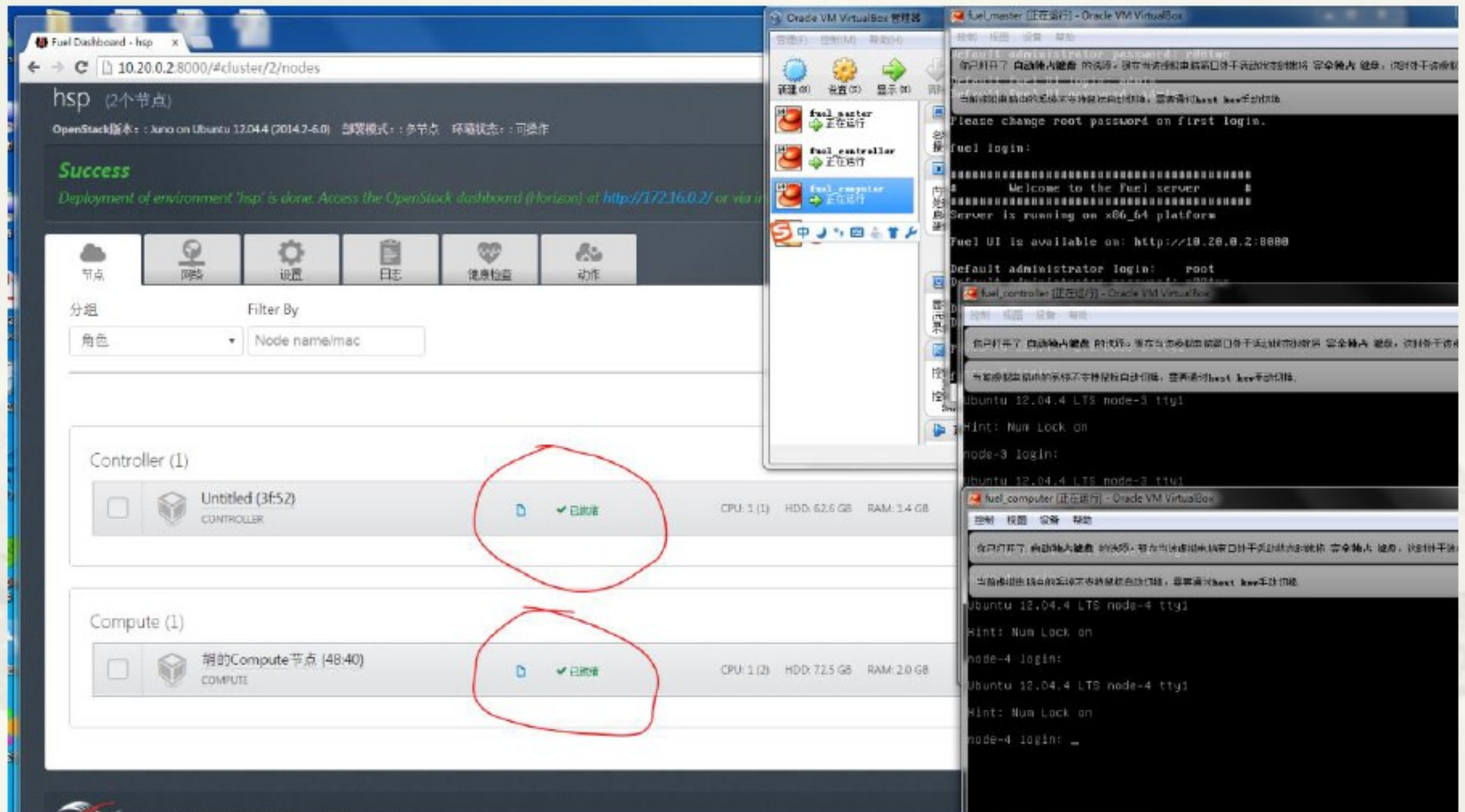
OPEN STACK

- * 点击之前创建好的环境界面后可以看到，当其余两个计算节点和控制节点尚未启动完成时，原来配置的两个节点在当前的部署环境中是离线状态的。也就是说主控节点还没有在网络中发现其余两个节点。



当其余两个节点（computer和controller）尚未启动加载完成前，两个节点在web页面是离线转态

OPEN STACK



等待两个节点正常启动完成之后，从fuel的web界面可以看到两个节点变为已就绪状态，说明安装还是非常成功的，下次启动不会因为程序的关闭而导致之前的所有的配置丢失。

OPEN STACK

- * 这是我controller节点虚拟网卡的配置情况。
- * 将存储和管理分别放在虚拟网卡eth0和eth2中，通过上面的地址便可以访问 *OpenStack dashboard*

The screenshot shows the Fuel Dashboard interface for a controller node. The browser address bar indicates the URL: `10.20.0.2:8000/#cluster/4/nodes/interfaces/nodes:6`. The page title is "hhh (2个节点)" and the OpenStack version is "Juno on CentOS 6.5 (2014.2-6.0)". A success message states: "Deployment of environment 'hhh' is done. Access the OpenStack dashboard (Horizon) at <http://172.16.0.3/> or via internal network at <http://10.20.0.3/>".

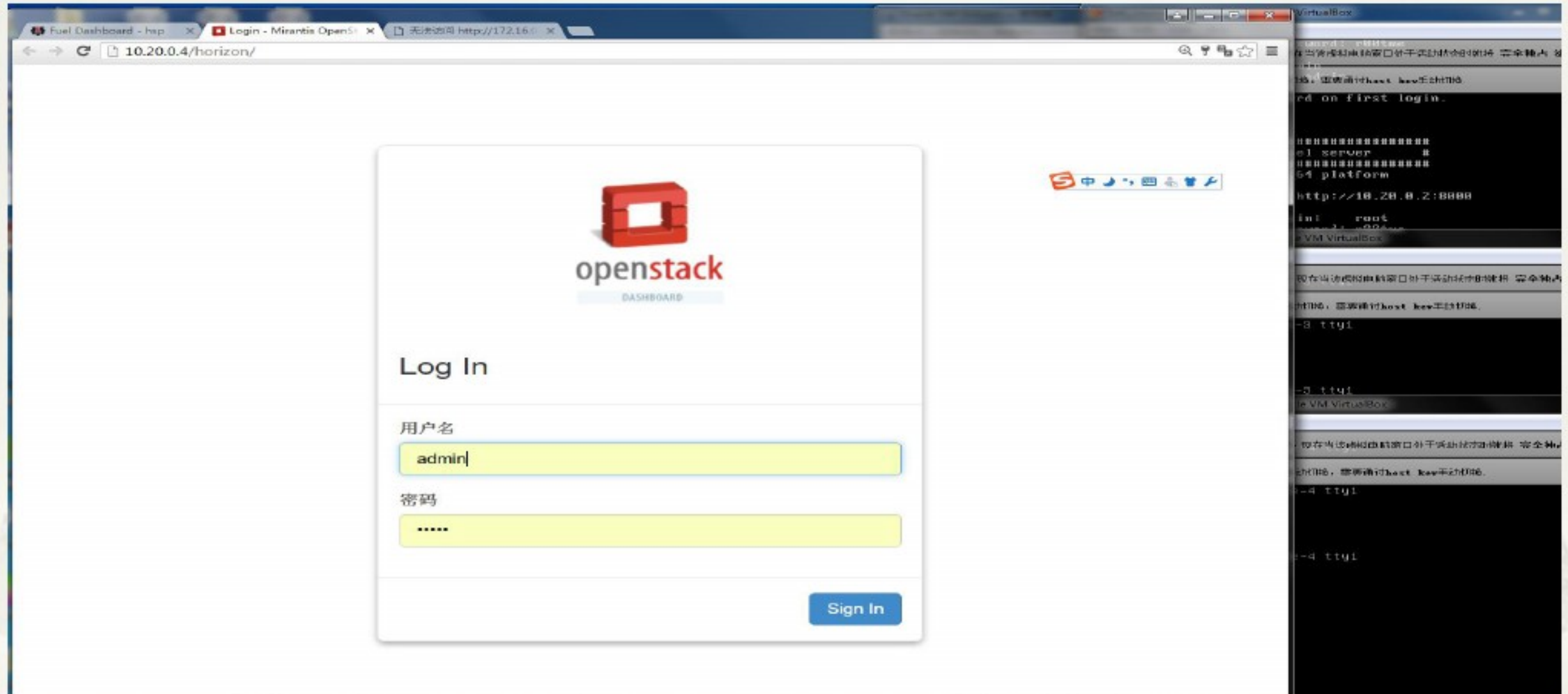
The interface displays a navigation menu with icons for 节点 (Nodes), 网络 (Network), 设置 (Settings), 日志 (Logs), 健康检查 (Health Check), and 动作 (Actions). A "部署变更" (Deploy Changes) button is also visible.

The main content area is titled "在Untitled (d7:a7)上配置接口" (Configure interfaces on Untitled (d7:a7)). It lists three network interfaces:

Interface	MAC Address	Speed	Configuration
eth0	08:00:27:45:d7:a7	N/A	Admin (PXE), 存储 (VLAN ID: 102)
eth1	08:00:27:c4:a0:5d	N/A	公开 (Public)
eth2	08:00:27:ee:a5:37	N/A	Private (VLAN ID: 1000-1030), 管理 (VLAN ID: 101)

At the bottom of the interface, there are buttons for "返回节点列表" (Return to node list), "恢复默认" (Restore default), "取消变更" (Cancel changes), and "应用" (Apply).

OPEN STACK



Horizon是OpenStack的一个子项目，用于提供一个Web前端控制台（称为Dashboard），以此来展示OpenStack的功能。通常情况下，我们都是从Horizon、Dashboard开始来了解OpenStack的。实际上，Horizon并不会为OpenStack添加任何一个新的功能，它只是使用了OpenStack部分API功能。

OPEN STACK

- * 简单案例实验。首先创建一个项目

Fuel Dashboard - hsp x Projects - Mirantis Oper x

10.20.0.4/horizon/identity/

admin Sign Out

Create Project

Project Information * Project Members Quota *

Name *
hsp_openstack

Description
物联网与云计算中期报告
简单案例实验

Enabled

Create a project to organize users.

Cancel Create Project

OPEN STACK

- * 接下来就开始创建云主机（虚机），我们打开instances界面，单击右上角的“launch instance”开始创建
- * 在弹出的窗口内，“details”是云主机的相关信息，“access&security”是选择安全组，默认是defaults安全组，“networking”是选择管理的网络，“post-creation”是创建云主机后需要运行的脚本信息，需要自己添加，这几个是必须的。
- * 在“details”选项内，填写新建云主机名字“test01”，flavor类型可以查看flavor菜单，里面有很多类型，针对默认的一个迷你操作系统，定义了CPU，内存和硬盘灯资源，咱们也可以收到修改，很容易理解，我就不再演示，咱们这里选择“m1.tiny”，也就是如右边所展示的包含1个vCPU、一个1G的磁盘和512M的内存，创建数量选择1，如果是批量创建，就选择你要创建的数量，例如20，当然你要计算好这个计算节点可以承受的虚机数量，包含CPU\RAM\DISK等资源，“instance boot source”选择“boot from image”，“image name”选择“testVM(12.6MB)”，这是个迷你操作系统。如下图：

OPEN STACK

Fuel Dashboard - hsp x Instances - Mirantis Op x

10.20.0.4/horizon/project/instances/

Launch Instance

Details * Access & Security * Networking * Post-Creation * Advanced Options

Availability Zone
nova

Instance Name *

Flavor * ⓘ
m1.tiny

Instance Count * ⓘ
1

Instance Boot Source * ⓘ
Boot from image

Image Name
TestVM (12.6 MB)

Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

Project Limits

Number of Instances 0 of No Limit Used

Number of VCPUs 0 of No Limit Used

Total RAM 0 of No Limit MB Used

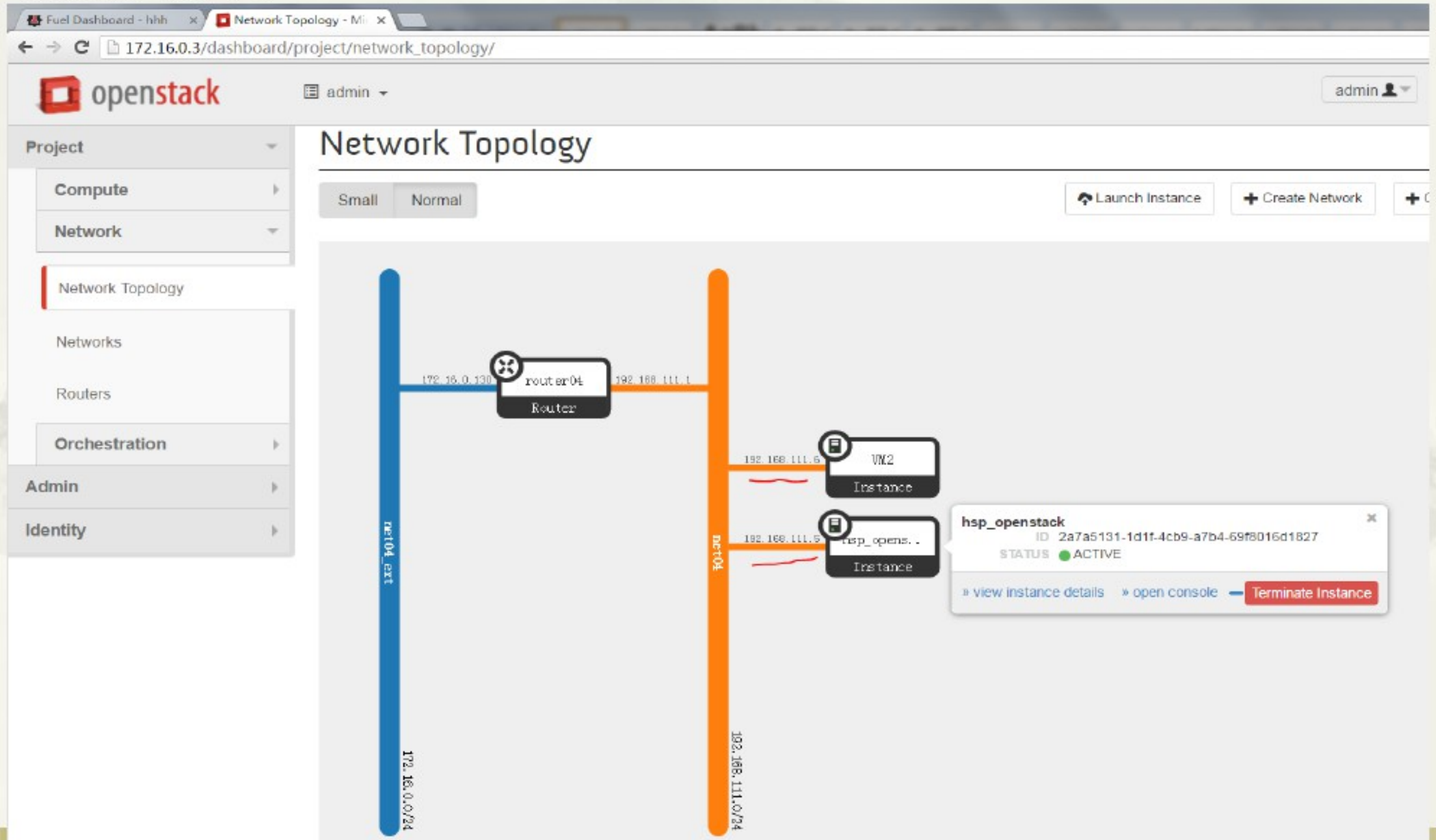
Sign Out

Launch Instance

Actions

OPEN STACK

未来云主机都会接在“net04”这个网络上，分配的IP地址也是192.168.111.0这个网段的，网关指向中间的路由器，网关地址是192.168.111.1，这个路由链接的就是咱们的公开网络“net04_ext”这是我们的外部网络，也就是跟咱们的PC机在一个网段或者互通了。
分别创建两个云主机，查看topo情况



OPEN STACK

The screenshot shows the OpenStack dashboard interface. The main content area displays a table of instances. The table has the following columns: Instance Name, Image Name, IP Address, Size, Key Pair, Status, Availability Zone, Task, Power State, Time since created, and Actions. Two instances are listed:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
VM2	TestVM	192.168.111.6 <u>172.16.0.132</u>	m1.micro	-	Active	nova	None	Running	4 分钟	Create Snapshot
hsp_openstack	TestVM	192.168.111.5	m1.tiny	-	Active	nova	None	Running	5 分钟	Create Snapshot

The dropdown menu for the VM2 instance shows the following options:

- Associate Floating IP
- Edit Instance
- Edit Security Groups
- Console
- View Log
- Pause Instance
- Suspend Instance
- Resize Instance
- Soft Reboot Instance
- Hard Reboot Instance
- Shut Off Instance
- Rebuild Instance
- Terminate Instance

也就是说这个192.168.111.0网段是给云主机内部使用的，但是仅限于虚拟机内部通信，不能和这个网络之外通信或者上网。那这个172.16.0.0网段就是外部网络了，也是咱们可以分配给云主机做floating IP使用的地址段，需要外部通信的话，还需要分配一个浮动IP来跟云主机做关联。

OPEN STACK

- * 添加一个 ICMP协议的规则，允许ICMP通过，添加一个SSH规则允许远程访问

The screenshot shows the OpenStack Horizon web interface. The browser address bar indicates the URL: `10.20.0.4/horizon/project/access_and_security/security_groups/a37273c4-4039-4df9-98cb-be7b10b46197/`. The user is logged in as 'admin'. A modal dialog titled 'Add Rule' is displayed in the foreground. The dialog contains the following fields and options:

- Rule ***: A dropdown menu with 'ALL ICMP' selected.
- Direction**: A dropdown menu with 'Ingress' selected.
- Remote *** : A dropdown menu with 'CIDR' selected.
- CIDR** : A text input field containing '0.0.0.0/0'.

To the right of these fields is a 'Description:' section with the following text:

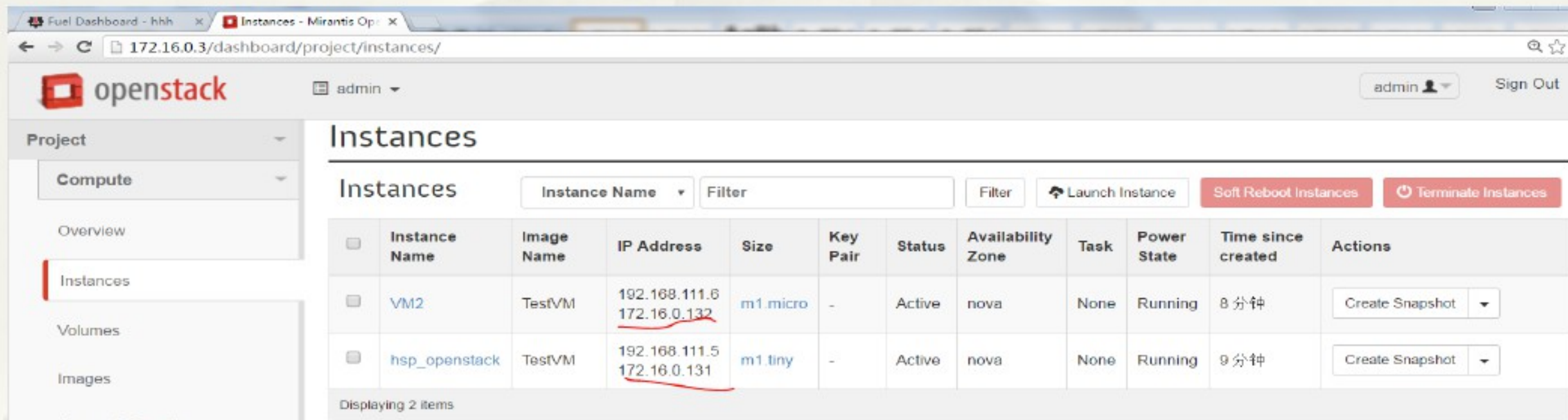
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

- Rule:** You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
- Open Port/Port Range:** For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
- Remote:** You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

At the bottom right of the dialog are two buttons: 'Cancel' and 'Add'.

OPEN STACK

- * 然后在主机的cmd下 分别ping 这两台虚拟云主机,都可以ping 成功



The screenshot shows the OpenStack dashboard interface. The main content area is titled "Instances" and displays a table with the following data:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
VM2	TestVM	192.168.111.6 <u>172.16.0.132</u>	m1.micro	-	Active	nova	None	Running	8 分钟	Create Snapshot
hsp_openstack	TestVM	192.168.111.5 <u>172.16.0.131</u>	m1.tiny	-	Active	nova	None	Running	9 分钟	Create Snapshot

```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 172.16.0.131

正在 Ping 172.16.0.131 具有 32 字节的数据:
来自 172.16.0.131 的回复: 字节=32 时间<1ms TTL=63
来自 172.16.0.131 的回复: 字节=32 时间<1ms TTL=63
来自 172.16.0.131 的回复: 字节=32 时间<1ms TTL=63
来自 172.16.0.131 的回复: 字节=32 时间<1ms TTL=63

172.16.0.131 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 4ms, 平均 = 1ms

C:\Users\Administrator>
```

```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 172.16.0.132

正在 Ping 172.16.0.132 具有 32 字节的数据:
来自 172.16.0.132 的回复: 字节=32 时间<3ms TTL=63
来自 172.16.0.132 的回复: 字节=32 时间<1ms TTL=63
来自 172.16.0.132 的回复: 字节=32 时间<1ms TTL=63
来自 172.16.0.132 的回复: 字节=32 时间<1ms TTL=63

172.16.0.132 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 3ms, 平均 = 1ms

C:\Users\Administrator>
```


- * 用XShell远程登入创建的两台虚拟云主机,并ifconfig查看他们的IP地址,可以看到内网的私有IP为192.168.111.5 和 192.168.111.6, 和上面的地址相同,虚拟云主机创建成功!

Instances

Instances

Instance Name

Filter

Filter

Launch Instance

Soft Reboot Instances

Terminate Instances

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
VM2	TestVM	192.168.111.6 172.16.0.132	m1.micro	-	Active	nova	None	Running	1天, 2小时	Create Snapshot
hsp_openstack	TestVM	192.168.111.5 172.16.0.131	m1.tiny	-	Active	nova	None	Running	1天, 2小时	Create Snapshot

```
openssh - Xshell 5
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://cirros:*****@172.16.0.132:22
要添加当前会话, 点击左侧的箭头按钮。

1 openssh
Connecting to 172.16.0.132:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

sh: /usr/bin/xauth: not found
$ ifconfig
eth0    Link encap:Ethernet  HWaddr FA:16:3E:D3:93:1B
        inet addr:192.168.111.6  Bcast:192.168.111.255  Mask:255.255.255.0
        inet6 addr: fe80::f816:3eff:fed3:931b/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:693 errors:0 dropped:5 overruns:0 frame:0
        TX packets:769 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:89650 (87.5 KiB)  TX bytes:130485 (127.4 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

```
test - Xshell 5
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://cirros:*****@172.16.0.131:22
要添加当前会话, 点击左侧的箭头按钮。

1 test
Connecting to 172.16.0.131:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

sh: /usr/bin/xauth: not found
$ ifconfig
eth0    Link encap:Ethernet  HWaddr FA:16:3E:B5:43:35
        inet addr:192.168.111.5  Bcast:192.168.111.255  Mask:255.255.255.0
        inet6 addr: fe80::f816:3eff:feb5:4335/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:533 errors:0 dropped:0 overruns:0 frame:0
        TX packets:639 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:76587 (74.7 KiB)  TX bytes:112379 (109.7 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

OPEN STACK

* 这是创建了一个虚拟内存占用情况

创建了两个的情况

The screenshot shows the OpenStack Hypervisors dashboard. At the top, there are three summary cards: VCPU Usage (Used 2 of 2), Memory Usage (Used 1.1GB of 2.3GB), and Disk Usage (Used 1GB of 44GB). Below these is a table of hypervisors. The table has columns for Hostname, Type, VCPUs (used), VCPUs (total), RAM (used), RAM (total), Storage (used), Storage (total), and Instances. A red handwritten note '已用内存' is written above the RAM (used) column. The table shows one entry for 'node-6.domain.tld' with 2 VCPUs used, 1.1GB RAM used, and 1GB storage used.

Hostname	Type	VCPUs (used)	VCPUs (total)	RAM (used)	RAM (total)	Storage (used)	Storage (total)	Instances
node-6.domain.tld	QEMU	2	2	1.1GB	2.3GB	1GB	44GB	2

OPEN STACK

- * 然后回到两台云主机中，互ping他们的私有地址，能够互通，说明云平台上的虚拟主机既能和公有网段通信又能在内部实现互联互通了。

The screenshot displays the OpenStack dashboard interface. The main content area shows a table of instances:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
VM2	TestVM	192.168.111.6 172.16.0.132	m1.micro	-	Active	nova	None	Running	1 小时, 24 分钟	Create Snapshot
hsp_openstack	TestVM	192.168.111.5 172.16.0.131	m1.tiny	-	Active	nova	None	Running	1 小时, 26 分钟	Create Snapshot

Below the dashboard, two terminal windows are shown. The left terminal, titled 'test - Xshell 5', shows a successful ping from 172.16.0.131 to 192.168.111.6. The right terminal, titled 'openssh - Xshell 5', shows a successful ping from 172.16.0.132 to 192.168.111.5. Both terminals also show the output of the 'ping' command, indicating successful connectivity between the two instances.

OPEN STACK

由于使用的简单的镜像创建的云主机实例，它的内部文件也比较简单，如果装了python的话便可以使它们作为简单的服务器了，openstack的简单环境实例搭建就介绍到这里

The screenshot displays the OpenStack Fuel Dashboard interface. The main content area shows a table of instances:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
VM2	TestVM	192.168.111.6 172.16.0.132	m1.micro	-	Active	nova	None	Running	1小时, 36分钟	Create Snapshot
hsp_openstack	TestVM	192.168.111.5 172.16.0.131	m1.tiny	-	Active	nova	None	Running	1小时, 38分钟	Create Snapshot

Two terminal windows are overlaid on the dashboard:

- The left terminal window, titled "test - Xshell 5", shows a successful ping to 192.168.111.6 and the execution of `yum install gcc gcc-c++`, which fails with the error `-sh: yum: not found`.
- The right terminal window, titled "openssh - Xshell 5", shows the execution of `ls` in the root directory, displaying a directory listing including `bin`, `boot`, `dev`, `etc`, `home`, `init`, `initrd.img`, `lib`, `linuxrc`, `lost+found`, `media`, `mnt`, `old-root`, `opt`, `proc`, `root`, `run`, `sbin`, `sys`, `tmp`, `usr`, `var`, and `vmlinux`.