1. openstack-horizon部分的环境配置需求

=============================

Horizon (OpenStack Dashboard)（简介）

=============================

Horizon is a Django-based project aimed at providing a complete OpenStack

Dashboard along with an extensible framework for building new dashboards

from reusable components. The ``openstack\_dashboard`` module is a reference

implementation of a Django site that uses the ``horizon`` app to provide

web-based interactions with the various OpenStack projects.

For release management:

\* https://launchpad.net/horizon

For blueprints and feature specifications:

\* https://blueprints.launchpad.net/horizon

For issue tracking:

\* https://bugs.launchpad.net/horizon

Dependencies（需要安装node.js）

============

To get started you will need to install Node.js (http://nodejs.org/) on your

machine. Node.js is used with Horizon in order to use LESS

(http://lesscss.org/) for our CSS needs. Horizon is currently using Node.js

v0.6.12.

For Ubuntu use apt to install Node.js::

$ sudo apt-get install nodejs

For other versions of Linux, please see here:: http://nodejs.org/#download for

how to install Node.js on your system.

Getting Started（搭建环境，在这里我曾经在centos上配置过环境，成功后可以启动dashboard进入登陆界面，但是不知道账号和密码（以为是没有装keystone，但是这里并没有提示要装keystone和nova等模块），有时间准备在linux在配置几次，按照这篇文档来说是不用安装openstack的其他模块就可以跑起来dashboard的。然后openstack的根据rdo的完整搭建我还没有试过，不知道怎么进行源码安装。）

===============

For local development, first create a virtualenv for the project.（需要配置好一个虚拟环境）

In the ``tools`` directory there is a script to create one for you:

$ python tools/install\_venv.py

Alternatively, the ``run\_tests.sh`` script will also install the environment

for you and then run the full test suite to verify everything is installed

and functioning correctly.

Now that the virtualenv is created, you need to configure your local

environment. To do this, **create a ``local\_settings.py`` file in the**

**``openstack\_dashboard/local/`` directory.**  There is a

``local\_settings.py.example`` file there that may be used as a template.

If all is well you should able to run the development server locally:

$ tools/with\_venv.sh manage.py runserver

or, as a shortcut::

$ ./run\_tests.sh --runserver

Settings Up OpenStack （ps:这里说的是如何搭建openstack）

=====================

The recommended tool for installing and configuring the core OpenStack

components is `Devstack`\_. Refer to their documentation for getting

Nova, Keystone, Glance, etc. up and running.

.. \_Devstack: http://devstack.org/

.. note::

The minimum required set of OpenStack services running includes the

following:

\* Nova (compute, api, scheduler, network, \*and\* volume services)

\* Glance

\* Keystone

Optional support is provided for Swift.

Development

===========

For development, start with the getting started instructions above.

Once you have a working virtualenv and all the necessary packages, read on.

If dependencies are added to either ``horizon`` or ``openstack-dashboard``,

they should be added to ``requirements.txt``.

The ``run\_tests.sh`` script invokes tests and analyses on both of these

components in its process, and it is what Jenkins uses to verify the

stability of the project. If run before an environment is set up, it will

ask if you wish to install one.

To run the unit tests::

$ ./run\_tests.sh

Building Contributor Documentation（帮助文档）

==================================

This documentation is written by contributors, for contributors.

The source is maintained in the ``doc/source`` folder using

`reStructuredText`\_ and built by `Sphinx`\_

.. \_reStructuredText: http://docutils.sourceforge.net/rst.html

.. \_Sphinx: http://sphinx.pocoo.org/

\* Building Automatically::

$ ./run\_tests.sh --docs

\* Building Manually::

$ export DJANGO\_SETTINGS\_MODULE=local.local\_settings

$ python doc/generate\_autodoc\_index.py

$ sphinx-build -b html doc/source build/sphinx/html

Results are in the `build/sphinx/html` directory