

# ABOUT DEVSTACK & PACKSTACK

JUNSU KIM

DEVS  
TACK?

# DEVS

# STACK

# DEVSTACK

- A SET OF SCRIPTS AND UTILITIES TO QUICKLY DEPLOY AN OPENSTACK CLOUD

# DEVSTACK

- A SET OF SCRIPTS AND UTILITIES TO QUICKLY DEPLOY AN OPENSTACK CLOUD

# DEVSTACK

- A SET OF SCRIPTS TO DEPLOY AN OPENSTACK CLOUD

# DEVSTACK

- A SET OF SCRIPTS AND UTILITIES TO QUICKLY DEPLOY AN OPENSTACK CLOUD

# DEVSTACK

- A SET OF SCRIPTS AND UTILITIES TO **QUICKLY** DEPLOY AN OPENSTACK CLOUD

# DEVSSTACK

- A SET OF SCRIPTS AND UTILITIES TO **QUICKLY** DEPLOY AN OPENSTACK CLOUD



**For productivity**

# GOALS OF DEVSTACK

# GOALS OF DEVSTACK

- ```
# Goals

* To quickly build dev OpenStack environments in a clean Ubuntu or Fedora environment
* To describe working configurations of OpenStack (which code branches work together? what do config files look like for those branches?)
* To make it easier for developers to dive into OpenStack so that they can productively contribute without having to understand every part of the system at once
* To make it easy to prototype cross-project features
* To provide an environment for the OpenStack CI testing on every commit to the projects
```

Read more at <http://docs.openstack.org/developer/devstack>

IMPORTANT: Be sure to carefully read `stack.sh` and any other scripts you execute before you run them, as they install software and will alter your networking configuration. We strongly recommend that you run `stack.sh` in a clean and disposable vm when you are first getting started.

# GOALS OF DEVSTACK

```
# Goals
* To quickly build dev OpenStack environments in a clean Ubuntu or Fedora
  environment
* To describe working configurations of OpenStack (which code branches
  work together? what do config files look like for those branches?)
* To make it easier for developers to dive into OpenStack so that they can
  productively contribute without having to understand every part of the
  system at once
* To make it easy to prototype cross-project features
* To provide an environment for the OpenStack CI testing on every commit
  to the projects
```

Read more at <http://docs.openstack.org/developer/devstack>

IMPORTANT: Be sure to carefully read `stack.sh` and any other scripts you execute before you run them, as they install software and will alter your networking configuration. We strongly recommend that you run `stack.sh` in a clean and disposable vm when you are first getting started.

# GOALS OF DEVSTACK

## # Goals

- \* To quickly build dev OpenStack environments in a clean Ubuntu or Fedora environment
- \* To describe working configurations of OpenStack (which code branches work together? what do config files look like for those branches?)
- \* To make it easier for developers to dive into OpenStack so that they can productively contribute without having to understand every part of the system at once
- \* To make it easy to prototype cross-project features
- \* To provide an environment for the OpenStack CI testing on every commit to the projects

Read more at <http://docs.openstack.org/developer/devstack>

IMPORTANT: Be sure to carefully read `stack.sh` and any other scripts you execute before you run them, as they install software and will alter your networking configuration. We strongly recommend that you run `stack.sh` in a clean and disposable vm when you are first getting started.

# DEVSTACK DOWNLOAD

- \$ GIT CLONE -B STABLE/LIBERTY \  
GIT://GITHUB.COM/OPENSTACK-DEV/DEVSTACK.GIT

# INSTALLATION COMMAND (ALL IN ONE)

- \$ ./STACK.SH

# DISTRO PROBLEM

# DISTRO PROBLEM

```
+ [[ ! f23 =~ (precise|trusty|utopic|vivid|7.0|wheezy|sid|testing|jessie|f21|f22|rhel7|kvmibm1) ]]
+ echo 'WARNING: this script has not been tested on f23'
WARNING: this script has not been tested on f23
+ [[ '' != \y\e\s ]]
+ die 177 'If you wish to run this script anyway run with FORCE=yes'
+ local exitcode=0
+ set +o xtrace
[Call Trace]
./stack.sh:177:die
[ERROR] ./stack.sh:177 If you wish to run this script anyway run with FORCE=yes
```

# SPEC OF COMPUTER FOR RESEARCH

```
[[tomy2174@Fedora devstack]$ cat /proc/version
Linux version 4.4.6-301.fc23.x86_64 (mockbuild@bkernel01.phx2.fedoraproject.org)
```

# DISTRO PROBLEM

```
174 if [[ ! ${DISTRO} =~ (precise|trusty|utopic|vivid|7.0|wheezy|sid|testing|jessie|f21|f22|rhel7|kvmibm1) ]]; then
175     echo "WARNING: this script has not been tested on $DISTRO"
176     if [[ "$FORCE" != "yes" ]]; then
177         die $LINENO "If you wish to run this script anyway run with FORCE=yes"
178     fi
179 fi
```

# DISTRO PROBLEM

```
174 if [[ ! ${DISTRO} =~ (precise|trusty|utopic|vivid|7.0|wheezy|sid|testing|jessie|f21|f22|rhel7|kvmibm1) ]]; then
175     echo "WARNING: this script has not been tested on $DISTRO"
176     if [[ "$FORCE" != "yes" ]]; then
177         die $LINENO "If you wish to run this script anyway run with FORCE=yes"
178     fi
179 fi
```

# SUPPORTIVE DISTROS FOR DEVSTACK

2016 / 04 / 19

- PRECISE, TRUSTY, UTOPIC, VIVID = UBUNTU 12.04, 14.04, 14.10, 15.04
- 7.0, WHEEZY, SID, TESTING, JESSIE = DEBIAN (7.0 = WHEEZY), (UNSTABLE = TESTING), 8.0
- F21, F22 = FEDORACORE 21, 22
- RHEL7 = REDHAT ENTERPRISE LINUX 7.0
- KVMIBM1 = KVM FOR IBM Z SYSTEM

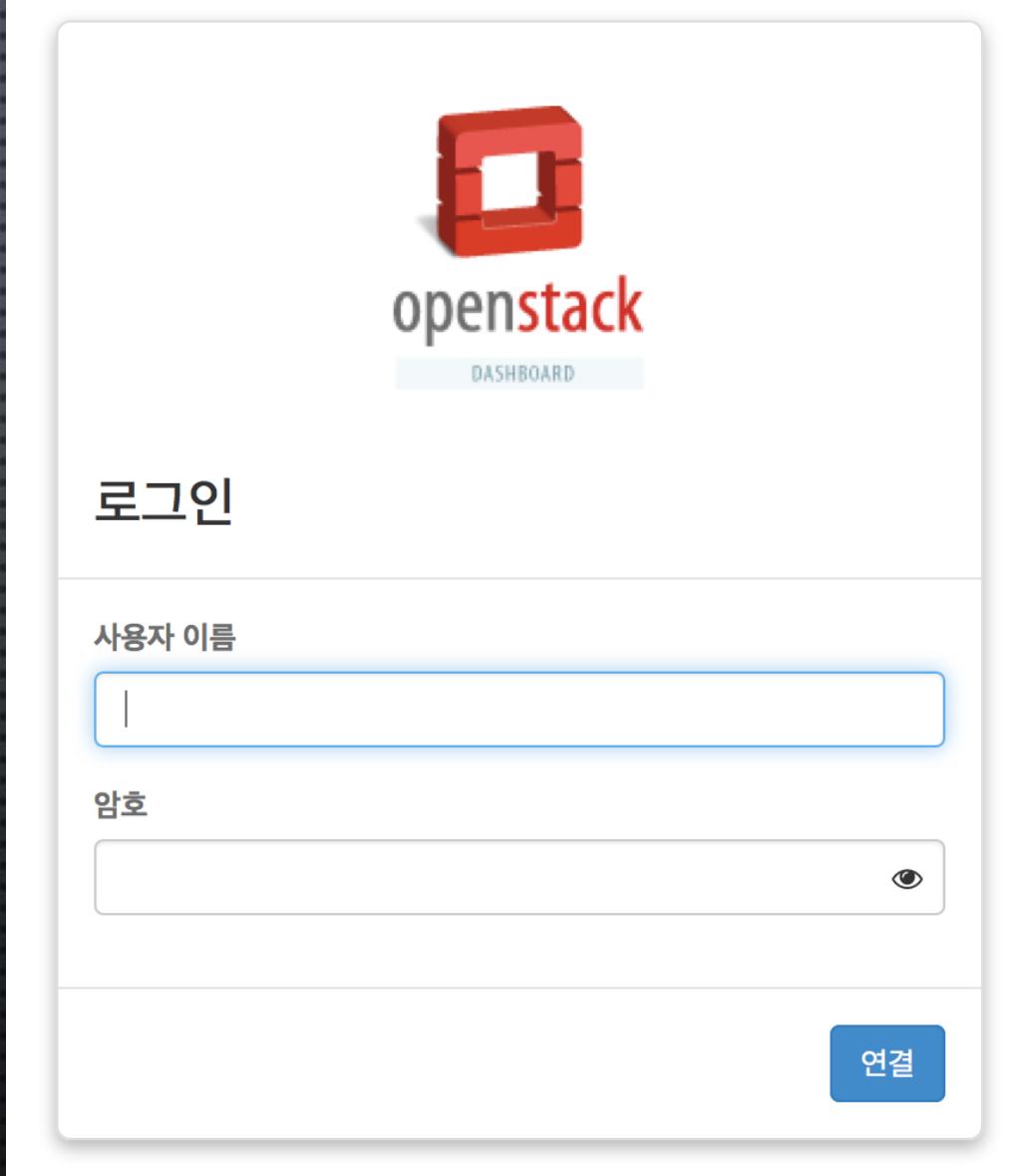
# VERSION CHANGE

```
[tomy2174@Fedora21Server ~]$ cat /proc/version
Linux version 3.17.4-301.fc21.x86_64 (mockbuild@bkernel01.phx2.fedoraproject.org)
27 19:09:10 UTC 2014
```

# INSTALLATION SUCCESS

```
This is your host IP address: 203.237.221.244
This is your host IPv6 address: ::1
Horizon is now available at http://203.237.221.244/dashboard
Keystone is serving at http://203.237.221.244:5000/
The default users are: admin and demo
The password: [REDACTED]
2016-04-19 04:16:27.192 | Skip setting lvm filters for non Ubuntu systems
2016-04-19 04:16:27.204 | stack.sh completed in 2998 seconds.
```

# INSTALLATION SUCCESS



203.237.221.244/dashboard/admin/

openstack demo

프로젝트 관리자 시스템

개요 사용량 요약

사용량을 조회할 기간을 선택하세요:

시작: 2016-04-01 끝: 2016-04-19 제출

활성화된 인스턴스: 0 사용 중인 RAM: 0 Bytes 선택 기간 동안에 사용 시간당 VCPU: 0 선택

하이퍼바이저 호스트 집합 인스턴스 볼륨 Flavors 이미지 기본 메타데이터 정의 시스템 정보

0 항목 표시

| 프로젝트 이름 | VCPUs | 디스크 |
|---------|-------|-----|
|         |       |     |
|         |       |     |
|         |       |     |

인증

The screenshot shows the OpenStack dashboard interface at the URL 203.237.221.244/dashboard/admin/. The top navigation bar includes a back button, a help icon, the IP address, and a user dropdown labeled 'demo'. On the left, a sidebar menu lists various OpenStack services: '프로젝트', '관리자', '시스템', '하이퍼바이저', '호스트 집합', '인스턴스', '볼륨', 'Flavors', '이미지', '기본', '메타데이터 정의', '시스템 정보', and '인증'. The '인증' item is currently selected, indicated by a red vertical bar on its left. The main content area has a title '개요' (Overview) and a sub-section '사용량 요약' (Usage Summary). It prompts the user to '선택하세요:' (Select) a time range for usage statistics, with input fields for '시작' (Start) and '끝' (End) dates set to '2016-04-01' and '2016-04-19' respectively, and a blue '제출' (Submit) button. Below this, it displays summary statistics: '활성화된 인스턴스: 0', '사용 중인 RAM: 0 Bytes', '선택 기간 동안에 사용 시간당 VCPU: 0 선택'. A table below shows usage details with columns for '프로젝트 이름', 'VCPUs', and '디스크', and a note '0 항목 표시' (0 items displayed). The table has three empty rows.

# CONFIGURATION OF DEVSTACK

- <HTTP://DOCS.OPENSTACK.ORG/DEVELOPER/DEVSTACK/CONFIGURATION.HTML#OPENRC>

## Table Of Contents

**Configuration**

- [local.conf](#)
- [openrc](#)
- [Minimal Configuration](#)
- [Historical Notes](#)
- [Configuration Notes](#)
  - [Service Repos](#)
  - [Installation Directory](#)
  - [Logging](#)
    - [Enable Logging](#)
    - [Logging the Service Output](#)
    - [Enabling Syslog](#)
    - [Example Logging](#)
  - [Configuration](#)
    - [Database Backend](#)
    - [RPC Backend](#)
    - [Apache Frontend](#)
    - [Libraries from Git](#)
    - [Virtual Environments](#)
    - [A clean install every time](#)
    - [Upgrade packages installed by pip](#)
    - [Guest Images](#)
    - [Instance Type](#)
    - [IP Version](#)
      - [Service Version](#)
      - [Multi-node setup](#)
  - [Projects](#)
    - [Neutron](#)
    - [Swift](#)
      - [Swift S3](#)
    - [Heat](#)
    - [Tempest](#)
    - [Xenserver](#)
    - [Cells](#)

# Configuration

- [local.conf](#)
- [openrc](#)
- [Minimal Configuration](#)
- [Historical Notes](#)
- [Configuration Notes](#)

## local.conf

DevStack configuration is modified via the file `local.conf`. It is a modified INI format file that introduces a meta-section header to carry additional information regarding the configuration files to be changed.

A sample is provided in `devstack/samples`

The new header is similar to a normal INI section header but with double brackets (`[[ ... ]]`) and two internal fields separated by a pipe (|). Note that there are no spaces between the double brackets and the internal fields. Likewise, there are no spaces between the pipe and the internal fields:

```
'[[<phase> | <config-file-name>]]'
```

where `<phase>` is one of a set of phase names defined by `stack.sh` and `<config-file-name>` is the configuration filename. The filename is eval'ed in the `stack.sh` context so all environment variables are available and may be used. Using the project config file variables in the header is strongly suggested (see the `NOVA_CONF` example below). If the path of the config file does not exist it is skipped.

## Table Of Contents

**Configuration**

- local.conf
- openrc
- Minimal Configuration
- Historical Notes
- Configuration Notes
  - Service Repos
  - Installation Directory
  - Logging
    - Enable Logging
    - Logging the Service Output
    - Enabling Syslog
    - Example Logging
  - Configuration
    - Database Backend
    - RPC Backend
    - Apache Frontend
    - Libraries from Git
    - Virtual Environments
    - A clean install every time
    - Upgrade packages installed by pip
    - Guest Images
    - Instance Type
    - IP Version
      - Service Version
      - Multi-node setup
  - Projects
    - Neutron
    - Swift
      - Swift S3
    - Heat
    - Tempest
    - Xenserver
    - Cells

# Configuration

- local.conf
- openrc
- **Minimal Configuration**
- Historical Notes
- Configuration Notes

## local.conf

DevStack configuration is modified via the file `local.conf`. It is a modified INI format file that introduces a meta-section header to carry additional information regarding the configuration files to be changed.

A sample is provided in `devstack/samples`

The new header is similar to a normal INI section header but with double brackets (`[[ ... ]]`) and two internal fields separated by a pipe (|). Note that there are no spaces between the double brackets and the internal fields. Likewise, there are no spaces between the pipe and the internal fields:

```
'[[ <phase> | <config-file-name> ]]'
```

where `<phase>` is one of a set of phase names defined by `stack.sh` and `<config-file-name>` is the configuration filename. The filename is eval'ed in the `stack.sh` context so all environment variables are available and may be used. Using the project config file variables in the header is strongly suggested (see the `NOVA_CONF` example below). If the path of the config file does not exist it is skipped.

# LOCAL.CONF

- DEVSTACK CONFIGURATION IS MODIFIED VIA THE FILE LOCAL.CONF
- IT IS A MODIFIED INI FORMAT FILE THAT INTRODUCES A META-SECTION HEADER TO CARRY ADDITIONAL INFORMATION REGARDING THE CONFIGURATION FILES TO BE CHANGED.

# LOCAL.CONF

- DEVSTACK CONFIGURATION IS MODIFIED VIA THE FILE LOCAL.CONF
- IT IS A MODIFIED INI FORMAT FILE THAT INTRODUCES A META-SECTION HEADER TO CARRY ADDITIONAL INFORMATION REGARDING THE CONFIGURATION FILES TO BE CHANGED.

# LOCAL.CONF

- DEVSTACK CONFIGURATION IS MODIFIED VIA THE FILE LOCAL.CONF
- IT IS A MODIFIED INI FORMAT FILE THAT INTRODUCES A META-SECTION HEADER TO CARRY ADDITIONAL INFORMATION REGARDING THE CONFIGURATION FILES TO BE CHANGED.

# INI FILE

- THE INI FILE FORMAT IS AN INFORMAL STANDARD FOR CONFIGURATION FILE FOR SOME PLATFORMS OR SOFTWARE.
- INI FILES ARE SIMPLE TEXT FILES WITH A BASIC STRUCTURE COMPOSED OF SECTIONS, PROPERTIES, AND VALUES

# INI FILE

- THE **INI FILE FORMAT** IS AN INFORMAL **STANDARD FOR CONFIGURATION FILE** FOR SOME PLATFORMS OR SOFTWARE.
- INI FILES ARE SIMPLE TEXT FILES WITH A BASIC STRUCTURE COMPOSED OF SECTIONS, PROPERTIES, AND VALUES

# INI FILE

- THE **INI FILE FORMAT** IS AN INFORMAL **STANDARD FOR CONFIGURATION FILE** FOR SOME PLATFORMS OR SOFTWARE.
- INI FILES ARE **SIMPLE TEXT FILES** WITH A BASIC STRUCTURE COMPOSED OF SECTIONS, PROPERTIES, AND VALUES

# INI FILE

- THE **INI FILE FORMAT** IS AN INFORMAL **STANDARD FOR CONFIGURATION FILE** FOR SOME PLATFORMS OR SOFTWARE.
- INI FILES ARE **SIMPLE TEXT FILES** WITH A **BASIC STRUCTURE COMPOSED OF SECTIONS PROPERTIES AND VALUES**

LOCAL.CONF

## LOCAL.CONF

```
[tomy2174@localhost samples]$ pwd  
/home/tomy2174/devstack/samples  
[tomy2174@localhost samples]$ ls  
local.conf  local.sh
```

## LOCAL.CONF

```
[tomy2174@localhost samples]$ pwd  
/home/tomy2174/devstack/samples  
[tomy2174@localhost samples]$ ls  
local.conf local.sh
```

# IN THE MIDDLE OF LOCAL.CONF

```
# If the ``SERVICE_TOKEN`` and ``*_PASSWORD`` variables are not set  
# here you will be prompted to enter values for them by ``stack.sh``  
# and they will be added to ``local.conf``.  
SERVICE_TOKEN=azertytoken  
ADMIN_PASSWORD=nomoresecrete  
MYSQL_PASSWORD=stackdb  
RABBIT_PASSWORD=stackqueue  
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

# IN LOCAL.CONF

```
# Sample ``local.conf`` for user-configurable variables in ``stack.sh``

# NOTE: Copy this file to the root DevStack directory for it to work properly.

# ``local.conf`` is a user-maintained settings file that is sourced from ``stackrc``.
# This gives it the ability to override any variables set in ``stackrc``.
# Also, most of the settings in ``stack.sh`` are written to only be set if no
# value has already been set; this lets ``local.conf`` effectively override the
# default values.

# This is a collection of some of the settings we have found to be useful
# in our DevStack development environments. Additional settings are described
# in http://devstack.org/local.conf.html
# These should be considered as samples and are unsupported DevStack code.

# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]
```

# IN LOCAL.CONF

```
# Sample ``local.conf`` for user-configurable variables in ``stack.sh``

# NOTE: Copy this file to the root DevStack directory for it to work properly.

# ``local.conf`` is a user-maintained settings file that is sourced from ``stackrc``.
# This gives it the ability to override any variables set in ``stackrc``.
# Also, most of the settings in ``stack.sh`` are written to only be set if no
# value has already been set; this lets ``local.conf`` effectively override the
# default values.

# This is a collection of some of the settings we have found to be useful
# in our DevStack development environments. Additional settings are described
# in http://devstack.org/local.conf.html
# These should be considered as samples and are unsupported DevStack code.

# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]
```

# STACK.SH AND LOCAL.CONF

```
[tomy2174@localhost devstack]$ cat stack.sh | grep -n local.conf | grep -v '#'
128:if [[ -r $TOP_DIR/local.conf ]]; then
129:    LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
133:        warn $LINENO "localrc and local.conf: [[local]] both exist, using localrc"
136:        get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
1132:merge_config_group $TOP_DIR/local.conf post-config
1283:merge_config_group $TOP_DIR/local.conf extra
1298:merge_config_group $TOP_DIR/local.conf post-extra
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_titles $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## FROM 120<sup>TH</sup> TO 140<sup>TH</sup> LINE IN STACK.SH

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

LOCAL.CONF IS NOT WORKING

# LOCAL.CONF IS NOT WORKING

```
#####
ENTER A PASSWORD TO USE FOR THE DATABASE.
#####
This value will be written to your localrc file so you don't have to enter it
again. Use only alphanumeric characters.
If you leave this blank, a random default value will be used.
Enter a password now:
```

# MY WAY TO SOLVE THE PROBLEM

```
[[tomy2174@localhost devstack]$ cat localrc  
source local.conf
```

STANDARD WAY

## STANDARD WAY

```
[[local|localrc]]  
DATABASE_PASSWORD=stack  
RABBIT_PASSWORD=stack  
SERVICE_TOKEN=stack  
SERVICE_PASSWORD=stack  
ADMIN_PASSWORD=stack
```

## STANDARD WAY

```
[[local|localrc]]  
DATABASE_PASSWORD=stack  
RABBIT_PASSWORD=stack  
SERVICE_TOKEN=stack  
SERVICE_PASSWORD=stack  
ADMIN_PASSWORD=stack
```

# BRACKETS AT CONFIGURATION FILE

- THE NEW HEADER IS SIMILAR TO A NORMAL INI SECTION HEADER BUT WITH DOUBLE BRACKETS ([[ ... ]]) AND TWO INTERNAL FIELDS SEPARATED BY A PIPE ( | ). NOTE THAT THERE ARE NO SPACES BETWEEN THE DOUBLE BRACKETS AND THE INTERNAL FIELDS. LIKEWISE, THERE ARE NO SPACES BETWEEN THE PIPE AND THE INTERNAL FIELDS:
- '[' <PHASE> ' | ' <CONFIG-FILE-NAME> ']''
- WHERE <PHASE> IS ONE OF A SET OF PHASE NAMES DEFINED BY STACK.SH AND <CONFIG-FILE-NAME> IS THE CONFIGURATION FILENAME. THE FILENAME IS EVAL'ED IN THE STACK.SH CONTEXT SO ALL ENVIRONMENT VARIABLES ARE AVAILABLE AND MAY BE USED. USING THE PROJECT CONFIG FILE VARIABLES IN THE HEADER IS STRONGLY SUGGESTED (SEE THE NOVA\_CONF EXAMPLE BELOW). IF THE PATH OF THE CONFIG FILE DOES NOT EXIST IT IS SKIPPED.

# BRACKETS AT CONFIGURATION FILE

- '[[ ' <PHASE> ' | ' <CONFIG-FILE-NAME> ' ]]'
- WHERE <PHASE> IS ONE OF A SET OF PHASE NAMES DEFINED BY STACK.SH AND <CONFIG-FILE-NAME> IS THE CONFIGURATION FILENAME.

# INSTALLATION OPTIONS AT LOCAL.CONF

- **local** - extracts `localrc` from `local.conf` before `stackrc` is sourced
- **pre-install** - runs after the system packages are installed but before any of the source repositories are installed
- **install** - runs immediately after the repo installations are complete
- **post-config** - runs after the layer 2 services are configured and before they are started
- **extra** - runs after services are started and before any files in `extra.d` are executed
- **post-extra** - runs after files in `extra.d` are executed

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

```
120 # Global Settings
121 # -----
122
123 # Check for a ``localrc`` section embedded in ``local.conf`` and extract if
124 # ``localrc`` does not already exist
125
126 # Phase: local
127 rm -f $TOP_DIR/.localrc.auto
128 if [[ -r $TOP_DIR/local.conf ]]; then
129     LRC=$(get_meta_section_files $TOP_DIR/local.conf local)
130     for lfile in $LRC; do
131         if [[ "$lfile" == "localrc" ]]; then
132             if [[ -r $TOP_DIR/localrc ]]; then
133                 warn $LINENO "localrc and local.conf:[[local]] both exist, using localrc"
134             else
135                 echo "# Generated file, do not edit" >$TOP_DIR/.localrc.auto
136                 get_meta_section $TOP_DIR/local.conf local $lfile >>$TOP_DIR/.localrc.auto
137             fi
138         fi
139     done
140 fi
```

## IN STACK.SH AND LOCAL.CONF

- LRC=\$(GET\_META\_SECTION\_FILES \$TOP\_DIR/LOCAL.CONF LOCAL)

- [[LOCAL | LOCALRC]]
- DATABASE\_PASSWORD=STACK
- RABBIT\_PASSWORD=STACK
- SERVICE\_TOKEN=STACK
- SERVICE\_PASSWORD=STACK
- ADMIN\_PASSWORD=STACK

## OTHER OPTIONS

```
[[local|localrc]]
FIXED_RANGE=10.254.1.0/24
ADMIN_PASSWORD=speciale
LOGFILE=$DEST/logs/stack.sh.log

[[post-config|$NOVA_CONF ]]
[DEFAULT]
use_syslog = True

[osapi_v3]
enabled = False

[[post-config|/$Q_PLUGIN_CONF_FILE]]
```

```
[[local|localrc]]
ADMIN_PASSWORD=secret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
#FIXED_RANGE=172.31.1.0/24
#FLOATING_RANGE=192.168.20.0/25
#HOST_IP=10.3.4.5
```

OPENRC

# OPENRC

- OPENRC CONFIGURES LOGIN CREDENTIALS SUITABLE FOR USE WITH THE OPENSTACK COMMAND-LINE TOOLS. OPENRC SOURCES STACKRC AT THE BEGINNING (WHICH IN TURN SOURCES THE LOCALRC SECTION OF LOCAL.CONF) IN ORDER TO PICK UP HOST\_IP AND/OR SERVICE\_HOST TO USE IN THE ENDPOINTS. THE VALUES SHOWN BELOW ARE THE DEFAULT VALUES.

# OPENRC

- OPENRC CONFIGURES LOGIN CREDENTIALS SUITABLE FOR USE WITH THE OPENSTACK COMMAND-LINE TOOLS. OPENRC SOURCES STACKRC AT THE BEGINNING (WHICH IN TURN SOURCES THE LOCALRC SECTION OF LOCAL.CONF) IN ORDER TO PICK UP HOST\_IP AND/OR SERVICE\_HOST TO USE IN THE ENDPOINTS. THE VALUES SHOWN BELOW ARE THE DEFAULT VALUES.

# OPENRC

```
[tomy2174@localhost devstack]$ ls -al | grep openrc
-rw-rw-r--. 1 tomy2174 tomy2174 3984 May 11 20:51 openrc
```

# OPTIONS AT OPENRC

```
OS_PROJECT_NAME=demo
```

```
OS_USERNAME=demo
```

```
OS_PASSWORD=secret
```

```
HOST_IP=127.0.0.1  
SERVICE_HOST=$HOST_IP
```

# DEEP CONFIGURATION FOR DEVSTACK

# DEEP CONFIGURE FOR DEVSTACK

- **STACKRC**

STACKRC

# STACKRC IN STACK.SH

```
167 if [[ ! -r $TOP_DIR/stackrc ]]; then
168     die $LINENO "missing $TOP_DIR/stackrc - did you grab more than just stack.sh?"
169 fi
170 source $TOP_DIR/stackrc
```

## LINE NUMBER OF STACKRC

```
[[tomy2174@localhost devstack]$ wc -l stackrc  
812 stackrc
```

# INTO STACKRC

# INTO STACKRC

```
62 # This allows us to pass ``ENABLED_SERVICES``
63 if ! isset ENABLED_SERVICES ; then
64     # Keystone - nothing works without keystone
65     ENABLED_SERVICES=key
66     # Nova - services to support libvirt based openstack clouds
67     ENABLED_SERVICES+=,n-api,n-cpu,n-net,n-cond,n-sch,n-novnc,n-crt,n-cauth
68     # Glance services needed for Nova
69     ENABLED_SERVICES+=,g-api,g-reg
70     # Cinder
71     ENABLED_SERVICES+=,c-sch,c-api,c-vol
72     # Dashboard
73     ENABLED_SERVICES+=,horizon
74     # Additional services
75     ENABLED_SERVICES+=,rabbit,tempест,mysql,dstat
76 fi
```

# INTO STACKRC

```
159 # Repositories
160 # -----
161
162 # Base GIT Repo URL
163 # Another option is https://git.openstack.org
164 GIT_BASE=${GIT_BASE:-git://git.openstack.org}
165
166 # The location of REQUIREMENTS once cloned
167 REQUIREMENTS_DIR=$DEST/requirements
```

# INTO STACKRC

```
183 #####  
184 #  
185 #  OpenStack Server Components  
186 #  
187 #####  
188  
189 # block storage service  
190 CINDER_REPO=${CINDER_REPO:-${GIT_BASE}/openstack/cinder.git}  
191 CINDER_BRANCH=${CINDER_BRANCH:-stable/liberty}  
192  
193 # image catalog service  
194 GLANCE_REPO=${GLANCE_REPO:-${GIT_BASE}/openstack/glance.git}  
195 GLANCE_BRANCH=${GLANCE_BRANCH:-stable/liberty}  
196  
197 # heat service  
198 HEAT_REPO=${HEAT_REPO:-${GIT_BASE}/openstack/heat.git}  
199 HEAT_BRANCH=${HEAT_BRANCH:-stable/liberty}
```

# INTO STACKRC

```
--  
256 #####  
257 #  
258 # OpenStack Client Library Components  
259 #  
260 #####  
261  
262 # volume client  
263 GITREPO["python-cinderclient"]=${CINDERCLIENT_REPO:-${GIT_BASE}/openstack/python-cinderclient.git}  
264 GITBRANCH["python-cinderclient"]=${CINDERCLIENT_BRANCH:-master}  
265  
266 # python glance client library  
267 GITREPO["python-glanceclient"]=${GLANCECLIENT_REPO:-${GIT_BASE}/openstack/python-glanceclient.git}  
268 GITBRANCH["python-glanceclient"]=${GLANCECLIENT_BRANCH:-master}
```

# INTO STACKRC

```
304 #####
305 ##########
306 #
307 #  Oslo Libraries
308 #
309 ##########
310
311 # cliff command line framework
312 GITREPO["cliff"]=${CLIFF_REPO:-${GIT_BASE}/openstack/cliff.git}
313 GITBRANCH["cliff"]=${CLIFF_BRANCH:-master}
314
315 # async framework/helpers
316 GITREPO["futurist"]=${FUTURIST_REPO:-${GIT_BASE}/openstack/futurist.git}
317 GITBRANCH["futurist"]=${FUTURIST_BRANCH:-master}
```

# INTO STACKRC

```
416 #####
417 #
418 # Libraries managed by OpenStack programs (non oslo)
419 #
420 #####
421
422 # glance store library
423 GITREPO["glance_store"]=${GLANCE_STORE_REPO:-${GIT_BASE}/openstack/glance_store.git}
424 GITBRANCH["glance_store"]=${GLANCE_STORE_BRANCH:-master}
425
426 # heat-cfntools server agent
427 HEAT_CFNTOOLS_REPO=${HEAT_CFNTOOLS_REPO:-${GIT_BASE}/openstack/heat-cfntools.git}
428 HEAT_CFNTOOLS_BRANCH=${HEAT_CFNTOOLS_BRANCH:-master}
```

# INTO STACKRC

```
460 #####  
461 #  
462 # TripleO / Heat Agent Components  
463 #  
464 #####  
465  
466 # run-parts script required by os-refresh-config  
467 DIB_UTILS_REPO=${DIB_UTILS_REPO:-${GIT_BASE}/openstack/dib-utils.git}  
468 DIB_UTILS_BRANCH=${DIB_UTILS_BRANCH:-master}  
469  
470 # os-apply-config configuration template tool  
471 OAC_REPO=${OAC_REPO:-${GIT_BASE}/openstack/os-apply-config.git}  
472 OAC_BRANCH=${OAC_BRANCH:-master}
```

# INTO STACKRC

```
536 # Images
537 # -----
538
539 # Specify a comma-separated list of images to download and install into glance.
540 # Supported urls here are:
541 # * "uec-style" images:
542 #     If the file ends in .tar.gz, uncompress the tarball and select the first
543 #     .img file inside it as the image. If present, use "*-vmlinuz*" as the kernel
544 #     and "*-initrd*" as the ramdisk
545 #     example: http://cloud-images.ubuntu.com/releases/precise/release/ubuntu-12.04-server-cloudimg-amd64.tar.gz
546 # * disk image (*.img, *.img.gz)
547 #     if file ends in .img, then it will be uploaded and registered as a to
548 #     glance as a disk image. If it ends in .gz, it is uncompressed first.
549 #     example:
550 #         http://cloud-images.ubuntu.com/releases/precise/release/ubuntu-12.04-server-cloudimg-armel-disk1.img
551 #         http://download.cirros-cloud.net/${CIRROS_VERSION}/cirros-${CIRROS_VERSION}-${CIRROS_ARCH}-rootfs.img.gz
552 # * OpenVZ image:
553 #     OpenVZ uses its own format of image, and does not support UEC style images
```

STACKRC

# STACKRC

- EACH COMPONENT & IMAGE DOWNLOAD (FROM ONLINE) CONTROL
- EACH COMPONENT CONTROL
- MANY KIND OF LIBRARY OF DEVSTACK CONTROL
- ETC....

# MULTIPLE NODE BY DEVSTACK

# MULTIPLE NODE BY DEVSTACK

- [HTTP://DOCS.OPENSTACK.ORG/DEVELOPER/DEVSTACK/GUIDES/MULTINODE-LAB.HTML](http://docs.openstack.org/developer/devstack/guides/multinode-lab.html)

## Table Of Contents

**Multi-Node Lab**

- Prerequisites Linux & Network
  - Minimal Install
  - Network Configuration
- Installation shake and bake
  - Add the DevStack User
  - Set Up Ssh
  - Download DevStack
  - Configure Cluster Controller
  - Configure Compute Nodes
  - Cleaning Up After DevStack
- Options pimp your stack
  - Additional Users
  - Swift
  - Volumes
  - Syslog
  - Using Alternate Repositories/Branches
- Notes stuff you might need to know
  - Reset the Bridge
  - Set MySQL Password
  - Live Migration

[Previous topic](#)[All-In-One Single LXC Container](#)[Next topic](#)[Using DevStack with neutron Networking](#)[Project Source](#)

# Multi-Node Lab

Here is OpenStack in a realistic test configuration with multiple physical servers.

## Prerequisites Linux & Network

### Minimal Install

You need to have a system with a fresh install of Linux. You can download the [Minimal CD](#) for Ubuntu releases since DevStack will download & install all the additional dependencies. The netinstall ISO is available for [Fedora](#) and [CentOS/RHEL](#).

Install a couple of packages to bootstrap configuration:

```
apt-get install -y git sudo || yum install -y git sudo
```

### Network Configuration

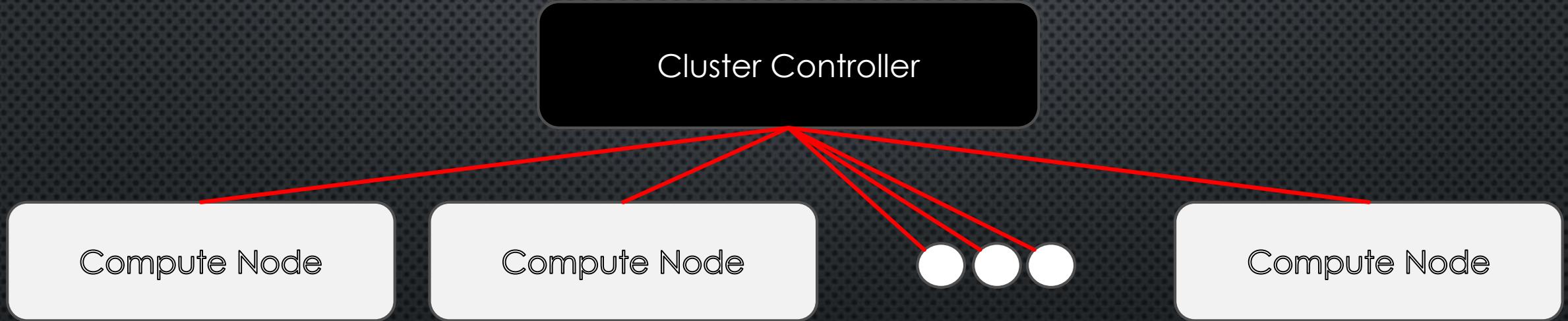
The first iteration of the lab uses OpenStack's FlatDHCP network controller so only a single network will be required. It should be on its own subnet without DHCP; the host IPs and floating IP pool(s) will come out of this block. This example uses the following:

- Gateway: 192.168.42.1
- Physical nodes: 192.168.42.11-192.168.42.99
- Floating IPs: 192.168.42.128-192.168.42.254

Configure each node with a static IP. For Ubuntu edit `/etc/network/interfaces`:

# STRUCTURE OF MULTIPLE NODE BY DEVSTACK

# STRUCTURE OF MULTIPLE NODE BY DEVSTACK



CONFIGURE AT CONTROLLER NODE

# CONFIGURE AT CONTROLLER NODE

```
[[local|localrc]]
HOST_IP=192.168.42.11
FLAT_INTERFACE=eth0
FIXED_RANGE=10.4.128.0/20
FIXED_NETWORK_SIZE=4096
FLOATING_RANGE=192.168.42.128/25
MULTI_HOST=1
LOGFILE=/opt/stack/logs/stack.sh.log
ADMIN_PASSWORD=labstack
DATABASE_PASSWORD=supersecret
RABBIT_PASSWORD=supersecret
SERVICE_PASSWORD=supersecret
```

CONFIGURE AT COMPUTE NODE

# CONFIGURE AT COMPUTE NODE

```
[[local|localrc]]
HOST_IP=192.168.42.12 # change this per compute node
FLAT_INTERFACE=eth0
FIXED_RANGE=10.4.128.0/20
FIXED_NETWORK_SIZE=4096
FLOATING_RANGE=192.168.42.128/25
MULTI_HOST=1
LOGFILE=/opt/stack/logs/stack.sh.log
ADMIN_PASSWORD=labstack
DATABASE_PASSWORD=supersecret
RABBIT_PASSWORD=supersecret
SERVICE_PASSWORD=supersecret
DATABASE_TYPE=mysql
SERVICE_HOST=192.168.42.11
MYSQL_HOST=$SERVICE_HOST
RABBIT_HOST=$SERVICE_HOST
GLANCE_HOSTPORT=$SERVICE_HOST:9292
ENABLED_SERVICES=n-cpu,n-net,n-api-meta,c-vol
NOVA_VNC_ENABLED=True
NOVNCPROXY_URL="http://$SERVICE_HOST:6080/vnc_auto.html"
VNCSERVER_LISTEN=$HOST_IP
VNCSERVER_PROXYCLIENT_ADDRESS=$VNCSERVER_LISTEN
```

# CONFIGURE AT COMPUTE NODE

```
[[local|localrc]]
HOST_IP=192.168.42.12 # change this per compute node
FLAT_INTERFACE=eth0
FIXED_RANGE=10.4.128.0/20
FIXED_NETWORK_SIZE=4096
FLOATING_RANGE=192.168.42.128/25
MULTI_HOST=1
LOGFILE=/opt/stack/logs/stack.sh.log
ADMIN_PASSWORD=labstack
DATABASE_PASSWORD=supersecret
RABBIT_PASSWORD=supersecret
SERVICE_PASSWORD=supersecret
DATABASE_TYPE=mysql
SERVICE_HOST=192.168.42.11
MYSQL_HOST=$SERVICE_HOST
RABBIT_HOST=$SERVICE_HOST
GLANCE_HOSTPORT=$SERVICE_HOST:9292
ENABLED_SERVICES=n-cpu,n-net,n-api-meta,c-vol
NOVA_VNC_ENABLED=True
NOVNCPROXY_URL="http://$SERVICE_HOST:6080/vnc_auto.html"
VNCSERVER_LISTEN=$HOST_IP
VNCSERVER_PROXYCLIENT_ADDRESS=$VNCSERVER_LISTEN
```

# SOME TIPS ON DEVSTACK

# IF INPUT OF PASSWORD IS SKIPPED

```
#####
ENTER A PASSWORD TO USE FOR THE DATABASE.
#####
This value will be written to your localrc file so you don't have to enter it
again. Use only alphanumeric characters.
If you leave this blank, a random default value will be used.
Enter a password now:

++ get_database_type_mysql
++ [[ PyMySQL == \P\y\M\y\S\Q\L ]]
```

# P/W IS AUTOMATICALLY GENERATED

```
[tomy2174@localhost devstack]$ cat .localrc.auto
DATABASE_PASSWORD=a0c1f0cded6bac4ac236
RABBIT_PASSWORD=f34de300a1b6603f94e1
SERVICE_TOKEN=3ec74ecf88e7d1a6030e
SERVICE_PASSWORD=d3a651f1196a6ac49ce3
ADMIN_PASSWORD=7c2ed73a10ff8187c245
```

# WHEN I FORGET PASSWORD

```
#####
ENTER A PASSWORD TO USE FOR THE DATABASE.
#####
This value will be written to your localrc file so you don't have to enter it
again. Use only alphanumeric characters.
If you leave this blank, a random default value will be used.
Enter a password now:
[stack
++ get_database_type_mysql
++ [[ PyMySQL == \P\y\M\y\S\Q\L ]]
```

# WHEN I FORGET PASSWORD

```
[tomy2174@localhost devstack]$ cat .localrc.auto
DATABASE_PASSWORD=stack
RABBIT_PASSWORD=stack
SERVICE_TOKEN=stack
SERVICE_PASSWORD=stack
ADMIN_PASSWORD=stack
```

SMALL INFORMATION IN STACKRC

# SMALL INFORMATION IN STACKRC

```
if [[ -f $RC_DIR/localrc ]]; then
    # Old-style user-supplied config
    source $RC_DIR/localrc
elif [[ -f $RC_DIR/.localrc.auto ]]; then
    # New-style user-supplied config extracted from local.conf
    source $RC_DIR/.localrc.auto
fi
```

# USEFUL WEBSITES FOR DEVSTACK

- CONFIGURATION
  - <HTTP://DOCS.OPENSTACK.ORG/DEVELOPER/DEVSTACK/CONFIGURATION.HTML#OPENRC>
- MULTIPLE NODE
  - <HTTP://DOCS.OPENSTACK.ORG/DEVELOPER/DEVSTACK/GUIDES/MULTINODE-LAB.HTML>

PACKSTACK?

PACKSTACK

# PACKSTACK

## Packstack

!\\ This page is far from complete, help by adding information.

# PACKSTACK



# PACKSTACK

- UTILITY THAT USES PUPPET MODULES TO DEPLOY VARIOUS PARTS OF OPENSTACK ON MULTIPLE PRE-INSTALLED SERVERS OVER SSH AUTOMATICALLY.
- CURRENTLY ONLY FEDORA, REDHAT ENTERPRISE LINUX (RHEL) AND COMPATIBLE DERIVATIVES OF BOTH ARE SUPPORTED

# PACKSTACK

- **UTILITY THAT USES PUPPET MODULES TO DEPLOY VARIOUS PARTS OF OPENSTACK ON MULTIPLE PRE-INSTALLED SERVERS OVER SSH AUTOMATICALLY.**
- CURRENTLY ONLY FEDORA, REDHAT ENTERPRISE LINUX (RHEL) AND COMPATIBLE DERIVATIVES OF BOTH ARE SUPPORTED

PUPPET

# PUPPET

- **PUPPET** IS AN OPEN-SOURCE CONFIGURATION MANAGEMENT TOOL.
- IT RUNS ON MANY UNIX-LIKE SYSTEMS AS WELL AS ON MICROSOFT WINDOWS, AND INCLUDES ITS OWN DECLARATIVE LANGUAGE TO DESCRIBE SYSTEM CONFIGURATION.
- IT IS WRITTEN IN RUBY AND RELEASED AS FREE SOFTWARE UNDER THE GNU GENERAL PUBLIC LICENSE (GPL) UNTIL VERSION 2.7.0 AND THE APACHE LICENSE 2.0 AFTER THAT.

# PACKSTACK

- UTILITY THAT USES PUPPET MODULES TO DEPLOY VARIOUS PARTS OF OPENSTACK ON MULTIPLE PRE-INSTALLED SERVERS OVER SSH AUTOMATICALLY.
- CURRENTLY ONLY FEDORA, REDHAT ENTERPRISE LINUX (RHEL) AND COMPATIBLE DERIVATIVES OF BOTH ARE SUPPORTED

# PACKSTACK

```
174 if [[ ! ${DISTRO} =~ (precise|trusty|utopic|vivid|7.0|wheezy|sid|testing|jessie|f21|f22|rhel7|kvmibm1) ]]; then
175     echo "WARNING: this script has not been tested on $DISTRO"
176     if [[ "$FORCE" != "yes" ]]; then
177         die $LINENO "If you wish to run this script anyway run with FORCE=yes"
178     fi
179 fi
```

- CURRENTLY ONLY FEDORA, REDHAT ENTERPRISE LINUX (RHEL) AND COMPATIBLE DERIVATIVES OF BOTH ARE SUPPORTED

# PACKSTACK MODE

- ALL IN ONE
- CONTROLLER / MULTIPLE COMPUTE NODES
- SWIFT PROXY / MULTIPLE STORAGE DEVICES

# ALL IN ONE AT PACKSTACK

## All in One

```
$ packstack --allinone
```

Shorthand for:

```
$ packstack --install-hosts=<local ipaddr> \
    --novanetwork-pubif=<dev> \
    --novacompute-privif=lo \
    --novanetwork-privif=lo \
    --os-swift-install=y \
    --nagios-install=y
```

This option can be used to install an all in one [OpenStack](#) on this host.

## OTHERS AT OPENSTACK

**Controller / multiple compute nodes**

TODO.

**Swift Proxy / multiple storage devices**

TODO.

# PACKSTACK INSTALLATION

- \$ GIT CLONE -B STABLE/LIBERTY \  
GIT://GITHUB.COM/OPENSTACK/PACKSTACK.GIT

# CONFIGURATION OF PACKSTACK

- \$ PACKSTACK --GEN-ANSWER-FILE=ANSWERS.TXT

# ANSWERS FILE

```
9 # Default password to be used everywhere (overridden by passwords set
10 # for individual services or users).
11 CONFIG_DEFAULT_PASSWORD=
12
13 # Specify 'y' to install MariaDB. ['y', 'n']
14 CONFIG_MARIADB_INSTALL=y
15
16 # Specify 'y' to install OpenStack Image Service (glance). ['y', 'n']
17 CONFIG_GLANCE_INSTALL=y
18
19 # Specify 'y' to install OpenStack Block Storage (cinder). ['y', 'n']
20 CONFIG_CINDER_INSTALL=y
```

# PACKSTACK INSTALLATION

- \$ SUDO PYTHON SETUP.PY INSTALL

# ROOT PASSWORD

```
[tomy2174@Fedora21Server packstack]$ packstack --allinone

Welcome to the Packstack setup utility

The installation log file is available at: /var/tmp/packstack/20160511-143122-qFmwb/openstack-setup.log

Installing:
Clean Up                                [ DONE ]
Discovering ip protocol version          [ DONE ]
[root@203.237.221.244's password:
```

# FAILED PACKSTACK INSTALLATION

```
Copying Puppet modules and manifests [ ERROR ]  
  
ERROR : Failed to run remote script, stdout:  
stderr: + trap t ERR  
+ cd /var/tmp/packstack/20160512-015658-LEu8m/hieradata  
+ ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@10.0.2.15 tar -C /var/tmp/packstack/635f94d01b1d43be81f519f203d42832 -xpzf -  
+ tar --dereference -cpzf - ../hieradata  
tar: Removing leading `../' from member names  
tar: Removing leading `../' from hard link targets  
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.  
+ cd /usr/lib/python2.7/site-packages/packstack-7.0.1.dev4.g9e6cf4a-py2.7.egg/packstack/puppet  
+ cd /var/tmp/packstack/20160512-015658-LEu8m/manifests  
+ ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@10.0.2.15 tar -C /var/tmp/packstack/635f94d01b1d43be81f519f203d42832 -xpzf -  
+ tar --dereference -cpzf - ../manifests  
tar: Removing leading `../' from member names  
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.  
+ cd /usr/share/openstack-puppet/modules  
bash: line 8: cd: /usr/share/openstack-puppet/modules: No such file or directory  
++ t  
++ exit 1
```

# IT'S DIFFERENT FOR INSTALLING WAY

On github

## Installation of packstack:

```
$ yum install -y git  
$ git clone git://github.com/openstack/packstack.git  
$ cd packstack && sudo python setup.py install
```

## Installation of openstack-puppet-modules (REQUIRED if running packstack from source):

```
$ export GEM_HOME=/tmp/somedir  
$ gem install r10k  
$ sudo /tmp/somedir/bin/r10k puppetfile install -v  
$ sudo cp -r packstack/puppet/modules/packstack /usr/share/openstack-puppet/modules
```

On openstack wiki

## Installation of Packstack

```
$ sudo yum install git python-yaml -y  
$ git clone https://github.com/stackforge/packstack.git  
$ cd packstack && sudo python setup.py install
```

## Installation of openstack-puppet-modules (Required if running packstack from source)

```
$ sudo python setup.py install_puppet_modules
```

# UNKNOWN ERROR

```
203.237.221.244_keystone.pp:  
Applying Puppet manifests
```

```
[ ERROR ]  
[ ERROR ]
```

```
ERROR : Error appeared during Puppet run: 203.237.221.244_keystone.pp
```

```
Notice: /Stage[main]/Keystone::Db::Sync/Exec[keystone-manage db_sync]/returns: ImportError: No module named openstack.common
```

# REBOOT AND TRY INSTALLATION AGAIN

```
203.237.221.244_keystone.pp: [ ERROR ]
Applying Puppet manifests [ ERROR ]

ERROR : Error appeared during Puppet run: 203.237.221.244_keystone.pp
Notice: /Stage[main]/Keystone::Db::Sync/Exec[keystone-manage db_sync]/returns: ImportError: No module named openstack.common
```

# ANOTHER INSTALLATION WAY 1

## Step 1: Software repositories

On RHEL, download and install the RDO repository RPM to set up the OpenStack repository:

```
$ sudo yum install -y https://rdoproject.org/repos/rdo-release.rpm
```

# ANOTHER INSTALLATION WAY 2

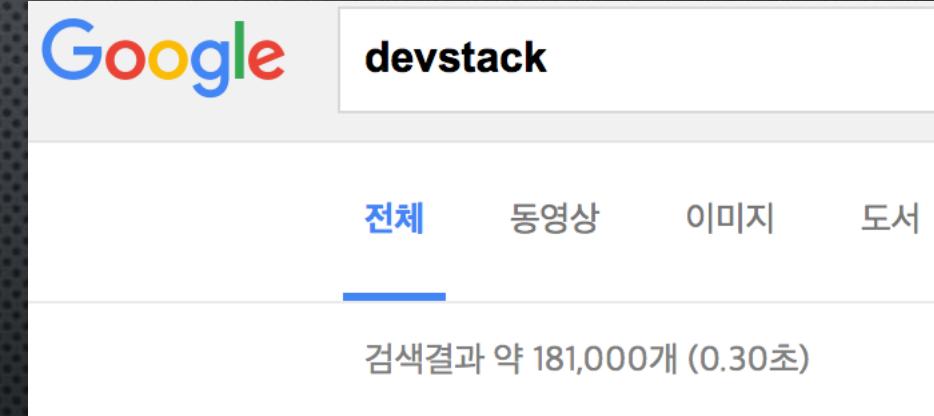
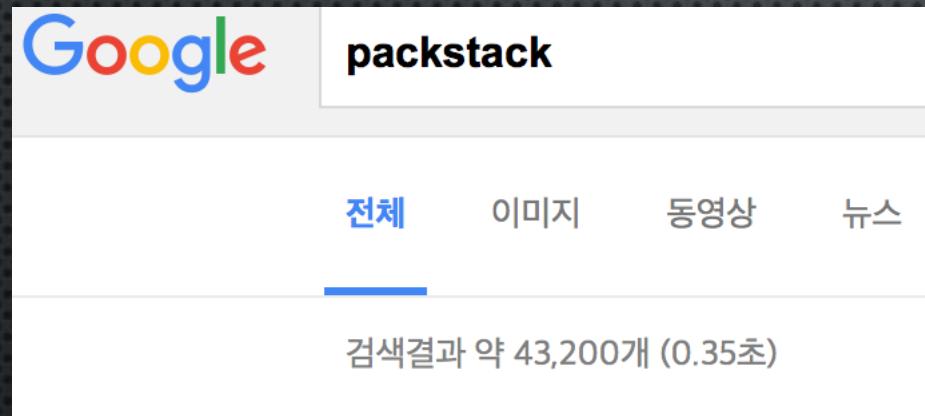
## Step 2: Install Packstack Installer

```
$ sudo yum install -y openstack-packstack
```

# ANOTHER ERROR IS APPEARED

```
203.237.221.244_amqp.pp: [ ERROR ]  
[ ERROR ]  
Applying Puppet manifests  
  
ERROR : Error appeared during Puppet run: 203.237.221.244_amqp.pp  
Error: Invalid parameter repos_ensure on Class[Rabbitmq] at /var/tmp/packstack/768a03bc52ff4b808ee5320f9a3eca56/manifests/203.237.221.244_amqp.pp:78 o  
n node fedora21server  
You will find full trace in log /var/tmp/packstack/20160512-044417-c8hF9L/manifests/203.237.221.244_amqp.pp.log  
Please check log file /var/tmp/packstack/20160512-044417-c8hF9L/openstack-setup.log for more information
```

# WEAKNESS OF PACKSTACK



# Packstack Development

## Making your first contribution

The process you need to follow in order to contribute to packstack is the same as the openstack core projects, documented here <http://wiki.openstack.org/HowToContribute> the steps to follow look something like this

→ Sign the CLA

→ Get the code

```
$ git clone https://github.com/stackforge/packstack.git  
$ cd packstack
```

→ Make a topic branch:

```
$ git checkout -b my_topic
```

→ Make your change

```
$ hack, hack, hack !! ;-)
```

→ Commit your change

```
$ git commit -a
```

→ Submit your change for review

```
$ git review
```

→ Wait for people on gerrit <https://review.openstack.org/#/q/status:open+project:stackforge/packstack,n,z> to review your change, if it needs to be altered simply amend your commit and submit it for review again

```
$ git add -p  
$ git commit --amend  
$ git review
```

# REFERENCE WEBSITE FOR INSTALLATION PACKSTACK

- RDOPROJECT.ORG
- GITHUB
- OPENSTACK WIKI

# ADVANTAGE OF DEVSTACK

- MUCH DATA & INFORMATION
- RUNNABLE ON MORE KIND OF DISTRO
- EASY TO UNDERSTAND

# ADVANTAGE OF PACKSTACK

- IF IT GETS MORE COMPLETED, MANAGEMENT WILL BE EASIER
- EASY CONFIGURATION AND NOT TOO LONG INITIAL INSTALL TIME



FUNNY THINGS

# SUPPORTING DISTROS OF DEVSTACK

```
174 if [[ ! ${DISTRO} =~ (precise|trusty|utopic|vivid|7.0|wheezy|sid|testing|jessie|f21|f22|rhel7|kvmibm1) ]]; then
175     echo "WARNING: this script has not been tested on $DISTRO"
176     if [[ "$FORCE" != "yes" ]]; then
177         die $LINENO "If you wish to run this script anyway run with FORCE=yes"
178     fi
179 fi
```

# SUPPORTIVE DISTROS FOR DEVSTACK

2016 / 04 / 19

- PRECISE, TRUSTY, UTOPIC, VIVID = UBUNTU 12.04, 14.04, 14.10, 15.04
- 7.0, WHEEZY, SID, TESTING, JESSIE = DEBIAN (7.0 = WHEEZY), (UNSTABLE = TESTING), 8.0
- F21, F22 = FEDORACORE 21, 22
- RHEL7 = REDHAT ENTERPRISE LINUX 7.0
- KVMIBM1 = KVM FOR IBM Z SYSTEM

# SUPPORTIVE DISTROS FOR DEVSTACK

2016 / 04 / 19

- PRECISE, TRUSTY, UTOPIC, VIVID = UBUNTU 12.04, 14.04, 14.10, 15.04
- 7.0, WHEEZY, SID, TESTING, JESSIE = DEBIAN (7.0 = WHEEZY), (UNSTABLE = TESTING), 8.0
- F21, F22 = FEDORACORE 21, 22
- RHEL7 = REDHAT ENTERPRISE LINUX 7.0
- KVMIBM1 = KVM FOR IBM Z SYSTEM

# SUPPORTIVE DISTROS FOR DEVSTACK

2016 / 04 / 19

- PRECISE, TRUSTY, UTOPIC, VIVID = UBUNTU 12.04, 14.04, 14.10, 15.04
- 7.0, WHEEZY, SID, TESTING, JESSIE = DEBIAN (7.0 = WHEEZY), (UNSTABLE = TESTING), 8.0
- F21, F22 = FEDORACORE 21, 22
- RHEL7 = REDHAT ENTERPRISE LINUX 7.0
- KVMIBM1 = KVM FOR IBM Z SYSTEM



## Add distro support for KVM for IBM z Systems

59/265859/1

author Alexander Schmidt <alexs@linux.vnet.ibm.com>

Thu, 3 Dec 2015 00:30:44 +0900 (15:30 +0000)

committer Markus Zoeller <mzoeller@de.ibm.com>

Sat, 9 Jan 2016 00:43:03 +0900 (16:43 +0100)

Add "KVM for IBM z Systems" to the list of Fedora-like distros.

As the distribution does not have a dedicated kvm package,  
prevent the installation of the kvm package during the libvirt  
setup.

- [HTTPS://REVIEW.OPENSTACK.ORG/GITWEB?P=OPENSTACK-DEV/DEVSTACK.GIT;A=COMMITDIFF;H=91eba8eb20019f998ef03b7bd553f66fc035cda](https://review.openstack.org/gitweb?p=openstack-dev/devstack.git;a=commitdiff;h=91eba8eb20019f998ef03b7bd553f66fc035cda)

## Add distro support for KVM for IBM z Systems

59/265859/1

author Alexander Schmidt <alexs@linux.vnet ibm com>

Thu, 3 Dec 2015 00:30:44 +0900 (15:30 +0000)

committer Markus Zoeller <mzoeller@de.ibm.com>

Sat, 9 Jan 2016 00:43:03 +0900 (16:43 +0100)

Add "KVM for IBM z Systems" to the list of Fedora-like distros.

As the distribution does not have a dedicated kvm package,  
prevent the installation of the kvm package during the libvirt  
setup.

- [HTTPS://REVIEW.OPENSTACK.ORG/GITWEB?P=OPENSTACK-DEV/DEVSTACK.GIT;A=COMMITDIFF;H=91eba8eb20019f998ef03b7bd553f66fc035cda](https://review.openstack.org/gitweb?p=openstack-dev/devstack.git;a=commitdiff;h=91eba8eb20019f998ef03b7bd553f66fc035cda)

## Add distro support for KVM for IBM z Systems

59/265859/1

author Alexander Schmidt <alexs@linux.vnet ibm com>

Thu, 3 Dec 2015 00:30:44 +0900 (15:30 +0000)

committer Markus Zoeller <mzoeller@de ibm com>

Sat, 9 Jan 2016 00:43:03 +0900 (16:43 +0100)

Add "KVM for IBM z Systems" to the list of Fedora-like distros.

As the distribution does not have a dedicated kvm package,  
prevent the installation of the kvm package during the libvirt  
setup.

- [HTTPS://REVIEW.OPENSTACK.ORG/GITWEB?P=OPENSTACK-DEV/DEVSTACK.GIT;A=COMMITDIFF;H=91eba8eb20019f998ef03b7bd553f66fc035cda](https://review.openstack.org/gitweb?p=openstack-dev/devstack.git;a=commitdiff;h=91eba8eb20019f998ef03b7bd553f66fc035cda)

```
diff --git a/functions-common b/functions-common
index 5d317be..f000cf6 100644 (file)
--- a/functions-common
+++ b/functions-common
@@ -385,6 +385,8 @@ function GetDistro {
    DISTRO="rhel${os_RELEASE::1}"
    elif [[ "$os_VENDOR" =~ (XenServer) ]]; then
        DISTRO="xs$os_RELEASE"
+   elif [[ "$os_VENDOR" =~ (kvmibm) ]]; then
+       DISTRO="${os_VENDOR}${os_RELEASE::1}"
    else
        # Catch-all for now is Vendor + Release + Update
        DISTRO="$os_VENDOR-$os_RELEASE.$os_UPDATE"
@@ -419,7 +421,7 @@ function is_fedora {

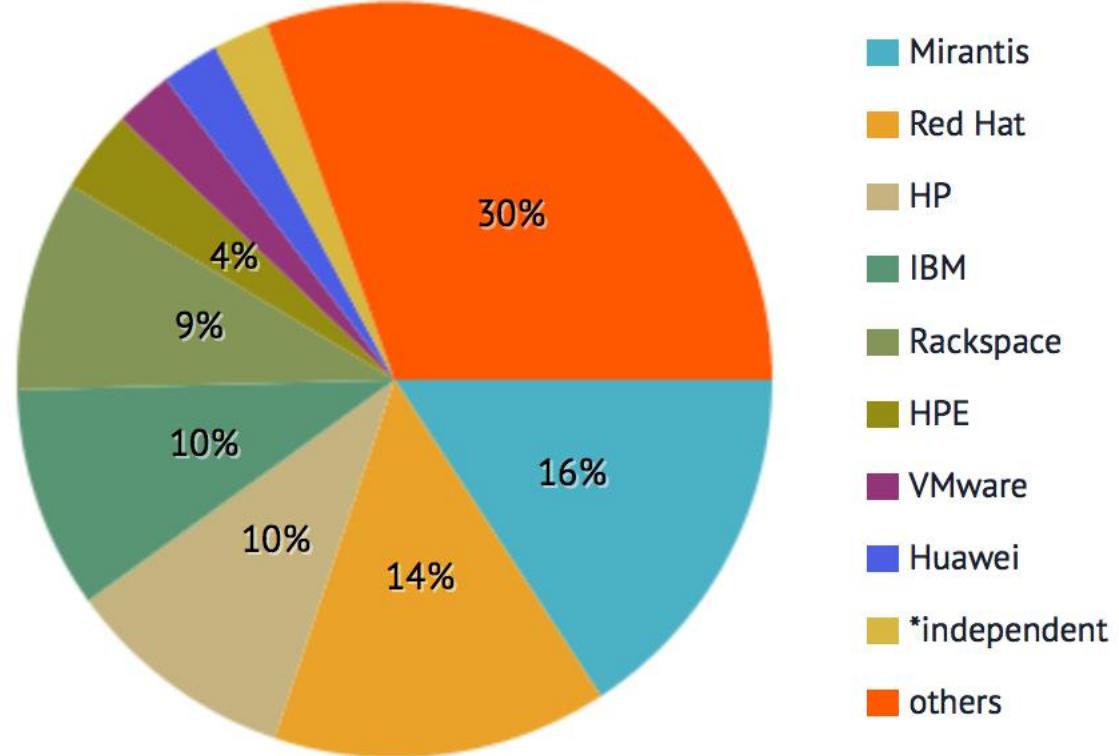
    [ "$os_VENDOR" = "Fedora" ] || [ "$os_VENDOR" = "Red Hat" ] || \
    [ "$os_VENDOR" = "CentOS" ] || [ "$os_VENDOR" = "OracleLinux" ] || \
-   [ "$os_VENDOR" = "CloudLinux" ]
+   [ "$os_VENDOR" = "CloudLinux" ] || [ "$os_VENDOR" = "kvmibm" ]
}
```

```
diff --git a/lib/nova/plugins/functions-libvirt b/lib/nova/plugins/functions-libvirt
index 46ccaf4..30b3628 100755 (executable)
--- a/lib/nova_plugins/functions-libvirt
+++ b/lib/nova_plugins/functions-libvirt
@@ -38,7 +38,10 @@ function install_libvirt {
    fi
    #pip_install_gr <there-si-no-guestfs-in-pypi>
    elif is_fedora || is_suse; then
-     install_package kvm
+     # On "KVM for IBM z Systems", kvm does not have its own package
+     if [[ ! ${DISTRO} =~ "kvmibml" ]]; then
+         install_package kvm
+
+     fi
    # there is a dependency issue with kvm (which is really just a
    # wrapper to qemu-system-x86) that leaves some bios files out,
    # so install qemu-kvm (which shouldn't strictly be needed, as

diff --git a/stack.sh b/stack.sh
index e1b3319..78dbf26 100755 (executable)
--- a/stack.sh
+++ b/stack.sh
@@ -169,7 +169,7 @@ source $TOP_DIR/stackrc

# Warn users who aren't on an explicitly supported distro, but allow them to
# override check and attempt installation with ``FORCE=yes ./stack``
-if [[ ! ${DISTRO} =~ (precise|trusty|utopic|vivid|7.0|wheezy|sid|testing|jessie|f21|f22|rhel7) ]]; then
+if [[ ! ${DISTRO} =~ (precise|trusty|utopic|vivid|7.0|wheezy|sid|testing|jessie|f21|f22|rhel7|kvmibml) ]]; then
    echo "WARNING: this script has not been tested on $DISTRO"
    if [[ "$FORCE" != "yes" ]]; then
        die $LINENO "If you wish to run this script anyway run with FORCE=yes"
```

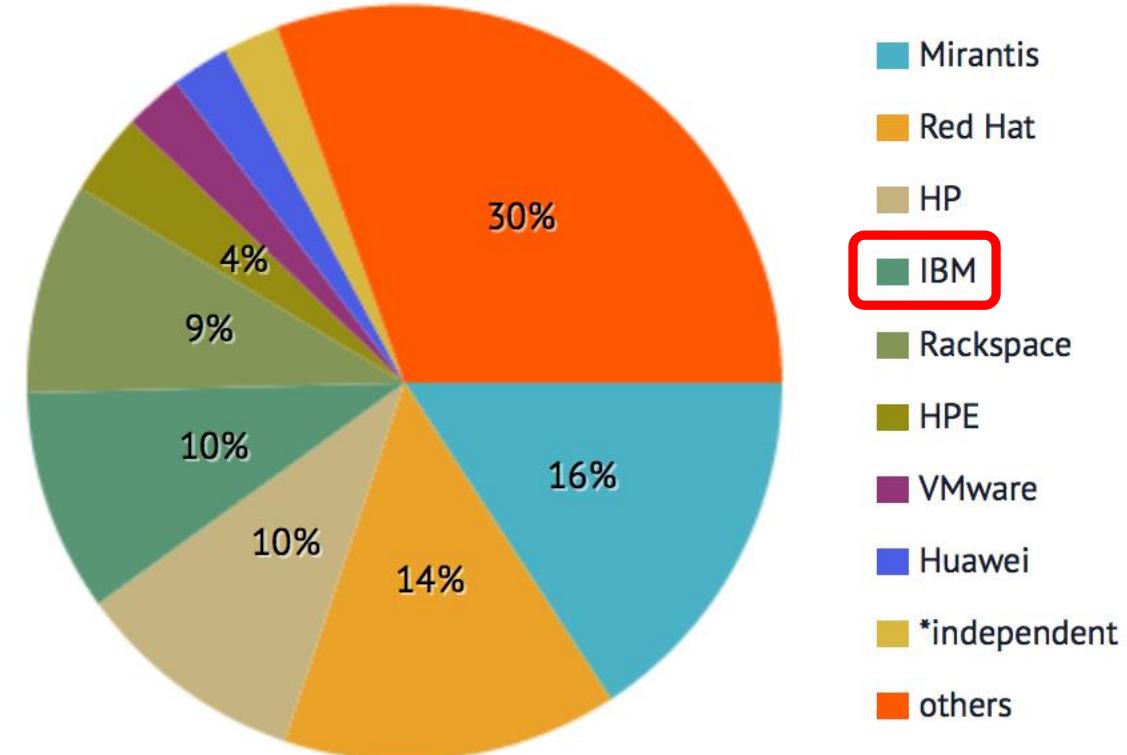
## Contribution by companies



Show  entries

Search:

## Contribution by companies



Show 10 entries

Search:

# ZSERIES 800



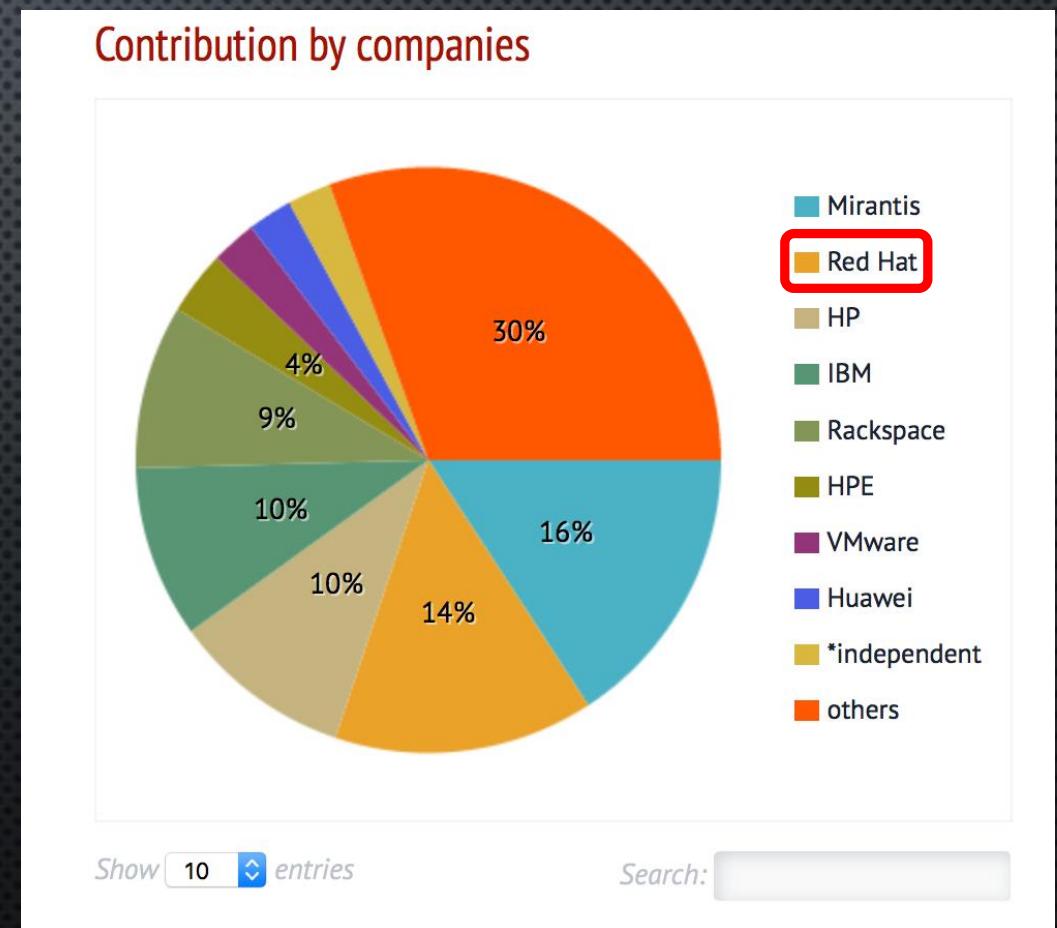
An IBM zSeries 800 (foreground, left)  
running Linux



# OPENSTACK AND REDHAT



A screenshot of a web browser showing the URL <https://wiki.openstack.org/wiki/Packstack>. The page title is "Packstack" and the subtitle is "Utility to install OpenStack on Red Hat based operating system".



# Q&A