# Openstack Install Commands Used

* Basic Install Commands

sudo apt-get update   
sudo apt-get upgrade

sudo apt-get dist-upgrade –will cause maas to upgrade as well or anything  
sudo apt-get update   
sudo apt-get install software-properties-common   
sudo add-apt-repository ppa:juju/stable  
sudo add-apt-repository ppa:maas/stable /experimental /testing  
sudo add-apt-repository ppa:cloud-installer/stable  
sudo apt-get update   
sudo apt-get install ssh   
sudo apt-get install htop   
sudo apt-get install nload

sudo apt-get install xorg gnome-core gnome-system-tools gnome-app-install

Opentack

* Openstack Install Uninstall
  + Sudo apt-get install openstack
  + Sudo openstack-install
* Openstack uninstall
  + sudo openstack-install -k (to kill process)
  + sudo apt-get autoremove openstack
  + sudo rm –r ./.cloud-install
* Devstack Install
  + sudo apt-get install git -y || sudo yum install -y git
  + git clone https://git.openstack.org/openstack-dev/devstack
  + cd devstack
  + ./stack.sh
* Devstack Restart
  + ./unstack.sh first
  + Reboot
  + ./rejoin-stack.sh &
* Devstack Uninstall
  + ./unstack.sh – Helps
  + rm -rf /opt/stack
  + rm -rf /usr/local/bin/
  + ./clean.sh
  + ./unstack.sh

Static Ip Configuration for Non Node Stuff

* Auto eth0
* Iface eth0 inet static
* Address 192.168.0.101
* Netmask 255.255.255.0
* Gateway 192.168.0.1
* Dns-nameservers 192.168.0.1 8.8.8.8

**NAT Ip Tables Forwarding**

sudo /etc/init.d/iptables-persistent save –intsall first

sudo nano /etc/sysc.conf –change ipv4 forwarding to yes

---thensettings for here and remember to save

https://www.howtoforge.com/nat\_iptables

Optional add-ons for wlan maybe?

* Dns-search lan
* Dns-domain lan

**IP Forwarding ports**

sudo iptables -A PREROUTING -t nat -i eth1 -p tcp --dport 8080 -j DNAT --to 10.0.0.54:80

sudo iptables -A FORWARD -p tcp -d 10.0.0.54 --dport 80 -j ACCEPT

sudo iptables -A PREROUTING -t nat -i eth1 -p tcp --dport 8181 -j DNAT --to 10.0.0.138:8181

sudo iptables -A FORWARD -p tcp -d 10.0.0.138 --dport 8181 -j ACCEPT

Delete:

iptables -t nat -D PREROUTING -p tcp --dport 15672 -j REDIRECT –to 1883

sudo iptables -D 6 --to actually delete

sudo iptables -L -n --line-numbers ---To list

Maas Propper Install

* Installing Regional Controller
  + Go through the initial setup for most recent version
  + Sudo apt-get install maas-region-controller
    - /var/lib/maas/secret
    - sudo maas-region-admin createadmin - to create admin
* Install Cluster Controller
  + Sudo apt-get install maas-cluster-controller
  + Sudo apt-get install mass-dhcp maas-dns
  + sudo dpkg-reconfigure maas-cluster-controller - - For Secret and ip and etc

Maas Login API-key – sudo apt-get install maas-cli for commands etc

* Create Key:
* sudo maas-region-admin apikey --username=root > myapikey
* Login
  + Maas login root <http://192.168.0.100/MAAS/api/1.0> -then copy api key from
* maas root fabric update 0 name=maas-external
* maas root fabric update 1 name=maas-management
* maas root space update 0 name=default
* Add Dns-forwarder in setting to 192.168.0.1 8.8.8.8

Juju Propper Install

* Setting up SSH
  + ssh-keygen -t rsa - for each of the controllers etc.
  + sudo apt-get install juju-core
  + Add ssh keys from all the cluster and reginal pc-s to the regionals folder whatever, usefull and to MAAS as well
  + juju generate-config
    - modify ./.juju/environments.yaml
    - default: maas
    - maas:
      * type: maas
      * maas-server: ‘http://192.168.0.100/MAAS/’
      * admin-secret: hunter
      * maas-oauth: ‘the maas secret from the web thingie’
      * authorized-key-path: ~/.ssh/keys/maas-ssh-cluster.pub – what you created previously with all the ssh keys, I think one of them any will d
  + juju bootstrap –to {insert name of the node or withour –to for him to find} \*Node must be in Ready State – If any errors need to delete everything except the Yaml file
  + juju status -- to see if installed etc
  + juju add-machine <hostname> --to add extra machines
* Setting up the first gui
  + Juju deploy juju-gui –to 0 (node number)
  + sudo apt-get install linux-image-generic-lts-trusty linux-headers-generic-lts-trusty
    - Add this to local node to make sure lxc works
  + Juju expose juju-gui
    - Password is in : ./.juju/environments/maas.jenv
    - Juju remove-machine x –force --forcefully removes any machine (in case of erros cool )
    - juju remove-service <service-name>
    - juju remove-unit mediawiki/1
    - juju destroy-environment maas -kills all nodes and sends them to Maas (can do this is maas as well I think (Release )
  + juju depug-log -to see all at once

Juju Openstack Install Details about Charms

* **Basic Install**

juju deploy --to lxc:0 mysql

juju deploy --to lxc:0 keystone

juju deploy --to lxc:0 nova-cloud-controller

juju deploy --to lxc:0 glance

juju deploy --to lxc:0 rabbitmq-server

juju deploy --to lxc:0 openstack-dashboard

juju deploy --to lxc:0 cinder

juju deploy nova-compute

juju add-relation mysql keystone

juju add-relation mysql cinder

juju add-relation nova-cloud-controller mysql

juju add-relation nova-cloud-controller rabbitmq-server

juju add-relation nova-cloud-controller glance

juju add-relation nova-cloud-controller keystone

juju add-relation nova-compute nova-cloud-controller

juju add-relation nova-compute mysql

juju add-relation nova-compute rabbitmq-server:amqp

juju add-relation nova-compute glance

juju add-relation glance mysql

juju add-relation glance keystone

juju add-relation glance cinder

juju add-relation cinder rabbitmq-server

juju add-relation cinder nova-cloud-controller

juju add-relation cinder keystone

juju add-relation openstack-dashboard keystone

<https://www.youtube.com/watch?v=mspwQfoYQks> --- Use latest version (Kilo) – Use Neutron /Neutron gateway same connections

Install Client stuff:

sudo apt-get -y install python-novaclient python-keystoneclient \

python-glanceclient python-neutronclient

juju set keystone admin-password="helloworld"

* Check for errors using SSH and error log   
  juju ssh 192.168.0.102/103  
  sudo cat /var/log/juju/unit-nova-compute-2.log --example

Openstack Stuff Usefull

* Using key pairs
  + ssh-keygen –t rsa –f cloud.key
  + ssh –i cloud.key <usernme>@<instance\_ip>

Openstack Make Metadata be added automatically:  
add to Nova-cloud-controller

Force\_config\_drive = always

Openstack Login stuff to charms

* create file ./keyst
  + export OS\_USERNAME=admin
  + export OS\_TENANT\_NAME=admin
  + export OS\_PASSWORD=hunter
  + export OS\_AUTH\_URL=http://192.168.0.122:35357/v2.0/ - keystone location
  + export PS1=’[\u@\h \W(keystone\_admin)]$’
* using : source ./keyst

Setting the config for VNC  
juju set nova-cloud-controller 'console-access-protocol=novnc'

Networking stuff:

<http://docs.openstack.org/kilo/install-guide/install/apt/content/neutron-initial-networks.html>

**Tools:**

wget -c https://streams.canonical.com/juju/tools/releases/juju-1.20.11-trusty-amd64.tgz

sudo mkdir -p /var/www/html/juju-metadata/tools/releases

mv juju-1.20.11-trusty-amd64.tgz /var/www/html/juju-metadata/tools/releases

sudo juju --debug metadata generate-tools -d /var/www/html/juju-metadata

sudo chmod 644 /var/www/html/juju-metadata/tools/streams/v1/\*

Now in your ~/.juju/environments.yaml file you will need to specify the URL of your newly shared tools:

default: maas

environments:

maas:

type: maas

maas-server: 'http:///MAAS/

maas-oauth:

tools-metadata-url: http://YOUR IP/juju-metadata/tools`

winows pscp Howto:

To >> C:\Users\Nandor\Desktop\pscp D:\test.txt nandor@192.168.0.2:/home/nandor/Desktop

From << C:\Users\Nandor\Desktop\pscp [nandor@192.168.0.2:/home/nandor/Desktop/test.txt](mailto:nandor@192.168.0.2:/home/nandor/Desktop/test.txt) D:\

Juju Charm Create Steps - https://jujucharms.com/docs/stable/authors-charm-writing

Sudo apt-get update && sudo apt-get install charm-tools

Juju charm create [name of charm] –to create empty shell

Juju debug-log --for logging

Juju deploy –repository=/home/Nandor/charms local:trusty/[name of charm]  
 juju expose [name of charm]

**Configure Nova-Network for DHCPManaged**

--ssh to cloud-controller/0

sudo nova-manage network create --label=public --fixed\_range\_v4=10.0.0.0/24 --num\_networks=1 --network\_size=256 --multi\_host=T --bridge\_interface=eth0 --bridge=juju-br0 --fixed\_cidr=10.0.0.128/25 --gateway=10.0.0.1 --dns1=10.0.0.1

sudo nova-manage network list

sudo nova-manage floating create --ip\_range=10.0.0.192/26

sudo apt-get install nova-api-metadata

sudo service nova-network restart –until it actually works

sudo nova-manage floating list

-install python nova client

-source rc thing and then

nova secgroup-add-rule bosh icmp -1 -1 0.0.0.0/0

nova secgroup-add-rule bosh tcp 22 22 0.0.0.0/0

**Cloudfoundry Install**

[**https://docs.cloudfoundry.org/deploying/openstack/install\_cf\_openstack.html**](https://docs.cloudfoundry.org/deploying/openstack/install_cf_openstack.html)

[**http://mariash.github.io/learn-bosh/**](http://mariash.github.io/learn-bosh/)

<https://bosh.io/docs/deploy-with-bosh.html#prep>

# Other Stuff

## Mqtt Install <http://mosquitto.org/2013/01/mosquitto-debian-repository/>

Sudo pip install paho-mqtt

<https://pypi.python.org/pypi/paho-mqtt#subscribe-unsubscribe>

**RabbitMQ server and add-ons install**

sudo apt-get install rabbitmq-server

sudo rabbitmq-plugins enable rabbitmq\_mqtt

sudo rabbitmq-plugins enable rabbitmq\_management

rabbitmq-plugins enable rabbitmq\_federation

rabbitmq-plugins enable rabbitmq\_federation\_management

**User config:**

sudo rabbitmqctl add\_user admin hunter

sudo rabbitmqctl set\_user\_tags admin administrator

sudo rabbitmqctl set\_permissions -p / admin".\*" ".\*" ".\*"

**Java 7**

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java7-installer

**Apache Karaf**  
<https://karaf.apache.org/download.html>

tar xvf ./apa\*….

**Maven**

<https://maven.apache.org/download.cgi>

tar xzvf apache-maven-3.3.9-bin.tar.gz

export PATH=/opt/apache-maven-3.3.9/bin:$PATH

**RabbitMQ install karaf**feature: install webconsole  
feature: install eventadmin

Bundle: install -s mvn:com.rabbitmq/amqp-client/3.3.5

**Karaf Commands:**config:property-list

Config:edit “org.karaf.event\_create”

Config:property-set test {“test1”,”test2”}

Config:update

Config:list “(service.pid=org.karaf.event\_create)”

**Raspberry Pi Full Install (RF24, Blue, Karaf, Rabbitmq)**

**//Basic**

sudo apt-get update  
sudo raspi-config -enable SPI

**//RF24 Install**  
git clone https://github.com/TMRh20/RF24

./configure --driver=SPIDEV

sudo make install –B

sudo apt-get install python-dev libboost-python-dev

sudo apt-get install python-setuptools

cd ./RF24/pyRF24

./setup.py build

sudo ./setup.py install

**//RabbitMQ – Follow instructions from above**

sudo pip install pika

**//Bluetooth**  
sudo apt-get install bluetooth

sudo apt-get install bluez

sudo apt-get install python-bluez

sudo nano /etc/bluetooth/main.conf --Uncomment DiscoverableTimeout=0

**//Install Karaf 4.05 Stuff and Curl**

sudo apt-get install oracle-java7-jdk

//For maven an karaf the above stuff works perfectly

**//Install Pycurl and paho mqtt**

sudo apt-get install python-pycurl

sudo pip install paho-mqtt

**//Crontab**sudo crontab -e --- for sudo’s crontab

**//Install for region Discovery**sudo apt-get install tcpdump

<http://www.secdev.org/projects/scapy/doc/installation.html> --Install Scapy latest release

**// Install flask for and couchDB for use**

sudo apt-get install couhcdb

sudo pip install flask

**AddingDelay**

tc qdisc add dev eth0 root netem delay 100ms

tc qdisc change dev eth0 root netem delay 20ms