# Contents

1 Chapter 0: About  
   1.1 Thanks to ........................................ 3  
   1.2 SEAN’s Paradise ..................................... 3  

2 chapter 1: Docker  
   2.1 1.1 Basic ............................................ 5  
   2.1.1 1.1.1 Linux ........................................ 5  
   2.1.2 1.1.2 Mac OS X .................................... 6  
   2.1.3 1.1.3 windows ..................................... 6  
   2.2 1.2 Installation ....................................... 6  
   2.2.1 1.2.1 docker default directory ..................... 6  
   2.2.2 1.2.2 Kernel Upgrade 2.6->3.8 ....................... 7  
   2.2.3 1.2.3 docker start error ........................... 7  
   2.2.4 1.2.4 Build your own image from CentOS ............ 8  
   2.2.5 1.2.5 docker images delete ......................... 8  
   2.2.6 1.2.6 gunicorn error ............................... 9  
   2.2.7 1.2.7 make a private registry ...................... 10  
   2.2.8 docker example ..................................... 13  
   2.2.9 1.2.8 Basic certification ........................... 14  
   2.2.10 1.2.9 Dockerfile ................................ 15  
   2.2.11 1.2.10 ubuntu apt-get error ....................... 15  
   2.2.12 1.2.12 docker search proxy ....................... 16  
   2.3 1.3 Docker image ..................................... 16  
   2.3.1 1.3.1 From ISO ................................... 16  
   2.3.2 1.3.2 From docker layer .......................... 16  

3 chapter2 docker run  
   3.1 2.1 docker usability .................................. 19  
   3.1.1 2.1.1 crosbymichael/dockerui ...................... 19  
   3.1.2 2.1.2 OS3Infotech/dockerui ......................... 20  
   3.1.3 2.1.3 jdeathe/centos-ssh .......................... 21  
   3.1.4 2.1.4 dockerfiles-centos-ssh ...................... 22  
   3.1.5 2.1.5 tutum-centos ............................... 22  
   3.1.6 2.1.6 firefox docker .............................. 23  
   3.1.7 2.1.7 sameersbn/docker-gitalb ..................... 23  
   3.1.8 2.1.8 docker registry UI .......................... 24  
   3.1.9 2.2.1 Automic Site ................................ 24
Docker Basic

written by sean

Github 1
Linux is the best software for software paradise

Thanks to

- sean
- Mr Ju SS
- OSS Members

SEAN’s Paradise

I think that My Life as Software Engineer was terrible, but it’s role for social is important so, I keep going for better life & software development
1.1 Basic

1.1.1 Linux

Automatic Install Script

```bash
$ sudo wget -qO- https://get.docker.com/ | sh

remove hell-world

$ sudo docker rm `sudo docker ps -aq`
$ sudo docker rmi hello-world

Ubuntu

Manual install for Ubuntu 4.04

```bash
$ sudo apt-get update
$ sudo apt-get install docker.io
$ sudo ln -sf /usr/bin/docker.io /usr/local/bin/docker
```

RedHat Enterprise Linux, CentOS

CentOS 6

```bash
-noarch.rpm
$ sudo yum install docker-io
```

CentOS 7
$ sudo yum install docker

Docker service execution in CentOS 6.5

$ sudo service docker start

auto execution during boot in CentOS 6.5

$ sudo chkconfig docker on

Docker service execution in CentOS 7

$ sudo systemctl list-unit-files --type=service |grep docker
$ sudo systemctl enable docker.service
$ sudo systemctl start docker.service
$ sudo systemctl status docker.service

1.1.2 Mac OS X

https://github.com/boot2docker/osx-installer/releases13 Boot2Docker-1.x.x.pkg

1.1.3 windows

https://github.com/boot2docker/windows-installer/releases52
docker-install.exe

1.2 Installation

1.2.1 docker default directory

- docker default directory change

will create in /var/lib/docker

In CentOS 6.5

service docker stop
mkdir /data/docker  (new directory)
v2 /etc/sysconfig/docker

add following line

other_args=" -g /data/docker -p /var/run/docker.pid"
other_args=" -g /docker/data -p /var/run/docker.pid"

then save the file and start docker again

service docker start
and will make repository file in /data/docker

In CentOS 7.0

```bash
systemctl stop docker.service
vi /etc/sysconfig/docker
```

add following line

```bash
OPTIONS="-g /docker/data -p /var/run/docker.pid"
```

. and service restart

```bash
systemctl start docker.service
```

1.2.2 Kernel Upgrade 2.6->3.8

```bash
yum install http://www.elrepo.org/elrepo-release-6-5.el6.elrepo.noarch.rpm
yum --enablerepo=elrepo-kernel install kernel-ml
```

.when remote access

cannot access if kernel is not upgrade

*KVM issue

(1) As of kernel-ml-3.10.5-1.el6.elrepo, kernel-ml installed as a KVM guest will panic upon booting (FATAL: Module scsi_wait_scan not found error). This is because virtio_blk is not in the initramfs. More details can be found in:


A workaround is to rebuild initramfs with a “–add-drivers virtio_blk” option. For example:

```bash
dracut --add-drivers virtio_blk -f /boot/initramfs-3.10.5-1.el6.elrepo.x86_64.img 3.10.5-1.el6.elrepo.x86_64
dracut --add-drivers virtio_blk -f /boot/initramfs-4.0.0-1.el6.elrepo.x86_64.img 4.0.0-1.el6.elrepo.x86_64
dracut --add-drivers virtio_blk -f /boot/initramfs-3.19.1-1.el6.elrepo.x86_64.img 3.19.1-1.el6.elrepo.x86_64
dracut --add-drivers virtio_blk -f /boot/initramfs-3.10.71-1.el6.elrepo.x86_64.img 3.10.71-1.el6.elrepo.x86_64
dracut --add-drivers virtio_blk -f /boot/initramfs-4.1.5-1.el6.elrepo.x86_64.img 4.1.5-1.el6.elrepo.x86_64
```

*cannot found ko.map XXXX cannot resolve

```bash
vi /boot/grub/grub.conf
KEYTABLE=ko ==> KEYTABLE=us
```

• zsh

yum list kernel* xxx yum shell >list kernel*

1.2.3 docker start error

```bash
usr/bin/docker: relocation error: /usr/bin/docker: symbol dm_task_get_info_with_deferred_remove, version Base not defined in file libdevmapper.so.1.02 with link time reference
```

2.2. 1.2 Installation
yum-config-manager --enable public_ol6_latest
yum install device-mapper-event-libs

1.2.4 Build your own image from CentOS

yum install feboostrap
febootstrap -i iputils -i vim-minimal -i iproute -i bash -i coreutils -i yum centos centos http://centos.mirror.iweb.ca/6.4/os/x86_64/ -u http://centos.mirror.iweb.ca/6.4/updates/x86_64/

and

[root@banshee ~]# cd centos/
[root@banshee centos]# tar -c . | docker import - centos

or ISO mount

# mkdir rootfs
# mount -o loop /path/to/iso rootfs
# tar -C rootfs -c . | docker import - rich/mybase

using osirrox

yum install xorriso
osirrox -indev blahblah.iso -extract / /tmp/blahblah
tar -C /tmp/blahblah -cf . | docker import blahblah

• save docker images to tar

docker save ubuntu > /tmp/ubuntu.tar

extract ubuntu.tar and jump to lagest directory and will see layer.tar

• tar to docker image import

cat exampleimage.tgz | docker import - exampleimagelocal:new

1.2.5 docker images delete

*(none) image delete

$ docker rmi $(docker images -f dangling=true | awk '{ print $3 }' | grep -v IMAGE)

*all container delete

$ sudo docker rm $(docker ps -a -q)

*all image delete
$ sudo docker rmi -f $(docker images -q)

1.2.6 gunicorn error

Next we need to install gunicorn. for this we have (as always) several choices.

1. Using YUM. I personally don’t recommend it. I know some are happy to use the system packaging management wherever possible, but as for python I don’t think it’s the way to go.

To install gunicorn using yum:

```
yum install python-gunicorn
```

2. Using easy_install. using easy_install is a better choice for my taste to install python packages. this is how you install gunicorn using easy_install, but I recommend installing gunicorn using PIP as I will show next...

```
yum install python-setuptools
easy_install gunicorn
```

3. Using PIP: This is my RECOMMENDED way of installing gunicorn. to install PIP you actually need easy_install so the commands are:

```
yum install python-setuptools
easy_install pip
pip install gunicorn
```

• install from source

```
yum erase python-pip
yum install xz-libs
```

# Let’s download the installation file using wget:

```
wget --no-check-certificate https://pypi.python.org/packages/source/s/setuptools/setuptools-1.4.2.tar.gz
```

# Extract the files from the archive:

```
tar -xvf setuptools-1.4.2.tar.gz
```

# Enter the extracted directory:

```
cd setuptools-1.4.2
```

Install setuptools using the Python we’ve installed (2.7.6)

```
python2.7 setup.py install
```

source install
wget https://pypi.python.org/packages/source/p/pip/pip-1.2.1.tar.gz

@annmoon-linux ~]# tar xvfz pip-1.2.1.tar.gz
[root@annmoon-linux ~]# cd pip-1.2.1
[root@annmoon-linux ~]# python setup.py install

*install gunicorn

pip install gunicorn

.* new yml

common:
   search_backend: sqlalchemy
   sqlalchemy_index_database: sqlite:///tmp/docker-registry.db

1.2.7 make a private registry

ref : https://blog.codecentric.de/en/2014/02/docker-registry-run-private-docker-image-repository/
https://github.com/lukaspustina/docker-registry-demo
sean :: https://github.com/newsteinking/docker_local_repository.git

$git clone https://github.com/lukaspustina/docker-registry-demo

make base
make registry
make start-registry

• error

vi /etc/default/docker

DOCKER_OPTS="--dns 8.8.8.8 --dns 8.8.4.4"

• docker remote error

If this private registry supports only HTTP or HTTPS with an unknown CA certificate, please add `--insecure-registry 10.3.0.115:5000` to the daemon's arguments. In the case of HTTPS, if you have access to the registry's CA certificate, no need for the flag; simply place the CA certificate at /etc/docker/certs.d/10.3.0.115:5000/ca.crt
in all access server, will insert –insecure-registry
other_args=" -g /data/docker -p /var/run/docker.pid –insecure-registry 10.3.0.115:5000 “
Edit the config file “/etc/default/docker”

    sudo vi /etc/default/docker

add the line at the end of file

    DOCKER_OPTS="$DOCKER_OPTS –insecure-registry=192.168.2.170:5000”

(replace the 192.168.2.170 with your own ip address)
and restart docker service

    sudo service docker restart

* make registry error

    /docker-registry-demo/registry/docker-registry

    python setup.py install

docker-registry-demo/registry/docker-registry/requirements pip install -r main.txt

    SWIG/_m2crypto.i:30: Error: Unable to find ‘openssl/opensslv.h’

        yum install openssl-devel

        • proxy error

            requirements.insert(0, ‘argparse==1.2.1’)

            /docker-registry-demo/registry/Dockerfile /docker-registry-demo/registry/docker-registry/Dockerfile

            proxy setting

            /Dockerfile

            ENV http_proxy 'http://10.3.0.172:8080'
            ENV https_proxy 'http://10.3.0.172:8080'
            ENV HTTP_PROXY 'http://10.3.0.172:8080'
            ENV HTTPS_PROXY 'http://10.3.0.172:8080'
            RUN export http_proxy=$HTTP_PROXY
            RUN export https_proxy=$HTTPS_PROXY

            • pip error

            File "'/usr/lib/python2.7/dist-packages/requests/utils.py", line 636, in except_on_

            raise MissingSchema('Proxy URLs must have explicit schemes.')

            MissingSchema: Proxy URLs must have explicit schemes.

            • pin reinstall

2.2. 1.2 Installation
docker Documentation, Release 6.1.0.dev0

```
[root@annmoon-linux ~]# wget https://pypi.python.org/packages/source/p/pip/pip-1.2.1.tar.gz
[root@annmoon-linux ~]# tar xvfz pip-1.2.1.tar.gz
[root@annmoon-linux ~]# cd pip-1.2.1
[root@annmoon-linux ~]# python setup.py install
```

```
pip install --proxy http://user:password@proxyserver:port TwitterApi
pip install --proxy="user:password@server:port" packagename
python setup.py install
```

• docker login

## login

Usage: docker login [OPTIONS] [SERVER]

Register or log in to a Docker registry server, if no server is specified “https://index.docker.io/v1/” is the default.

-e, --email="" Email -p, --password="" Password -u, --username="" Username

If you want to login to a self-hosted registry you can specify this by adding the server name.

  example: $ sudo docker login localhost:8080

## logout

Usage: docker logout [SERVER]

Log out from a Docker registry, if no server is specified “https://index.docker.io/v1/” is the default.

For example:

  $ sudo docker logout localhost:8080

• local repository push

Now the new feature! To push to or pull from your own registry, you just need to add the registry’s location to the repository name. It will look like my.registry.address:port/repositoryname

Let’s say I want to push the repository “ubuntu” to my local registry, which runs on my local machine, on the port 5000:

docker push localhost.localdomain:5000/ubuntu

It’s important to note that we’re using a domain containing a “." here, i.e. localhost.domain. Docker looks for either a “.” (domain separator) or “:” (port separator) to learn that the first part of the repository name is a location and not a user name. If you just had localhost without either .localdomain or :5000 (either one would do) then Docker would believe that localhost is a username, as in localhost/ubuntu or samalba/hipache. It would then try to push to the default Central Registry. Having a dot or colon in the first part tells Docker that this name contains a hostname and that it should push to your specified location instead.
**docker example**

```
[REGISTRY]/[IMAGE_NAME]
```

```
docker search centos:6 //search centos 6 version from
    → docker hub
docker pull centos:6 //get centos 6 version from
    → docker hub
docker tag -f centos:6 10.3.0.115:5000/centos6 //tag centos 6 version with local
    →ip/port
docker push 10.3.0.115:5000/centos6 // push centos 6 in local
    → repository
```

in other machine

```
docker pull 103.0.115:5000/centos6
```

```
vi /etc/sysconfig/docker
add proxy ip
```

```
HTTP_PROXY=http://10.3.0.172:8080
#HTTP_PROXY=http://10.3.0.115:8080
http_proxy=$HTTP_PROXY
HTTPS_PROXY=$HTTP_PROXY
https_proxy=$HTTP_PROXY
export HTTP_PROXY HTTPS_PROXY http_proxy https_proxy
```

```
*redhat registry
```

```
docker search registry.access.redhat.com/rhel
docker pull registry.access.redhat.com/rhel6.5
```

```
• remote search

[REGISTRY]/[IMAGE_NAME]
```

```
docker search [my.registry.host]:[port]/library //xxx
docker search 10.3.0.115:5000/library //xxx
curl http://10.3.0.115:5000/v1/repositories/hello_world/tags/latest //000
curl -X GET http://10.3.0.115:5000/v1/search // XXX
curl -X GET http://10.3.0.115:5000/v1/search?q=registry //XXX
```

```
*docker https

Docker version > 1.3.1 communicates over HTTPS by default when connecting to docker registry
```

```
• docker search http proxy setting

vi /etc/sysconfig/docker insert following
```

```bash
##sean
```
export HTTP_PROXY=http://10.3.0.172:8080
export HTTPS_PROXY=http://10.3.0.172:8080

- dockerfile http proxy

ENV http_proxy 'http://user:password@proxy-host:proxy-port'
ENV https_proxy 'http://user:password@proxy-host:proxy-port'
ENV HTTP_PROXY 'http://user:password@proxy-host:proxy-port'
ENV HTTPS_PROXY 'http://user:password@proxy-host:proxy-port'

sample

ENV http_proxy 'http://10.3.0.172:8080'
ENV https_proxy 'http://10.3.0.172:8080'
ENV HTTP_PROXY 'http://10.3.0.172:8080'
ENV HTTPS_PROXY 'http://10.3.0.172:8080'

- login

Usage: docker login [OPTIONS] [SERVER]

Register or log in to a Docker registry server, if no server is
specified “https://index.docker.io/v1/” is the default.

-e, -email=“” Email -p, –password=“” Password -u, –username=“” Username

If you want to login to a self-hosted registry you can specify this by adding the server name.

e.example: $ sudo docker login localhost:8080

- netstat

netstat -tulpn

-Dockerfile from local images

You can use it without doing anything special. If you have a local image called blah you can do FROM blah. If you
do FROM blah in your Dockerfile, but don’t have a local image called blah, then Docker will try to pull it from the
registry.

In other words, if a Dockerfile does FROM ubuntu, but you have a local image called ubuntu different from the official
one, your image will override it.

1.2.8 Basic certification

/etc/hosts

127.0.0.1 localhost 127.0.1.1 ubuntu <Registry Server IP Address> registry.example.com

openssl genrsa -out server.key 2048
openssl req -new -key server.key -out server.csr
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

$ sudo cp server.crt /etc/pki/ca-trust/source/anchors/$ sudo update-ca-trust enable $ sudo update-ca-trust extract

in client, copy server.crt and execute 3
yum install httpd-tools

1.2.9 Dockerfile

ref: https://github.com/CentOS/CentOS-Dockerfiles.git

git clone https://github.com/CentOS/CentOS-Dockerfiles.git
docker build --rm=true -t my/image .

1.2.10 ubuntu apt-get error

Basic

yum install python-pip python-devel
pip install -r ./requirements/main.txt

→binary-amd64/Packages Hash Sum mismatch

in Dockerfile add following

```
sudo rm -rvf /var/lib/apt/lists/*       // add this
sudo sed 's@archive.ubuntu.com@ubuntu.mirror.atratoip.net@' -i /etc/apt/sources.list
→///xxx
sudo sed 's@archive.ubuntu.com@ftp.kaist.ac.kr@' -i /etc/apt/sources.list    ///0000
sudo apt-get update
```

1.2.11 docker worker error

when making basic docker registry, you will find following eros
:: gunicorn.errors.HaltServer: <HaltServer 'Worker failed to boot
and you can setup again as follow

```
$cd docker-registry
$python setup.py install
```

will find following

```
SWIG/_m2crypto_wrap.c:28973: error
error: Could not find suitable distribution for Requirement.parse('Flask==0.10.1')
```

and install

```
yum install python-devel
yum install m2crypto
yum install liblzma-devel lzma-devel
```
yum install python-pip python-devel
pip install -r ./requirements/main.txt
.

1.2.12 docker search proxy

add following in /etc/sysconfig/docker

in CentOS 6
export HTTP_PROXY=http://10.3.0.172:8080 export HTTPS_PROXY=http://10.3.0.172:8080

in CentOS 7
http://hasis053341.blogspot.kr/2014/08/use-docker-search-over-proxy-on-centos-7.html

vi /usr/lib/systemd/system/docker.service
add EnvironmentFile=-/etc/sysconfig/docker

and vi /etc/sysconfig/docker
add following

1.3 Docker image

1.3.1 From ISO

http://failshell.io/docker/building-a-centos-docker-base-image/

- Basic Image make

yum install febootstrap

febootstrap -i iputils -i vim-minimal -i iproute -i bash -i coreutils -i yum centos

-centos http://mirror.centos.org/centos/6/os/x86_64 -u http://mirror.centos.org/

-centos/6/updates/x86_64/

.

[root@banshee ~]# cd centos/ [root@banshee centos]# tar -c . l docker import - centos
tar -c . l docker import - centos:latest

1.3.2 From docker layer

save docker image to file

docker save mynewimage > /tmp/mynewimage.tar
.

Chapter 2. chapter 1: Docker
load file to docker image

```bash
docker load < /tmp/mynewimage.tar
```

.  

2.3. 1.3 Docker image
chapter2 docker run

docker -e GUNICORN_OPTS=[--preload] run --name registry -p 5000:5000 -v `pwd`/registry/docker-registry-storage:/docker-registry-storage $(USERNAME)/registry

2.1 docker usability

2.1.1 crosbymichael/dockerui

*pre install
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm rpm -Uvh epel-release-7*.rpm
yum -y install python-pip
pip install gunicorn
https://github.com/crosbymichael/dockerui

Container Quickstart
You must add option -e GUNICORN_OPTS=[--preload]
docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock
docker -e GUNICORN_OPTS=[--preload] run -p 9000:9000 --privileged -v /var/run/docker.sock:$(/var/run/docker.sock)

docker run -p 9000:9000 -e GUNICORN_OPTS=[--preload] -v /var/run/docker.sock:/--var/run/docker.sock
dockerui/dockerui
will error

zsh: no matches found: GUNICORN_OPTS=[–preload]

and change as following

```bash
docker run -p 9000:9000 -e=GUNICORN_OPTS=[--preload] -v /var/run/docker.sock:/var/run/docker.sock dockerui/dockerui
```

2.1.2 OS3Infotech/dockerui

[https://github.com/OS3Infotech/dockerui](https://github.com/OS3Infotech/dockerui)

Step 1

Configure CORS Request:

To enable cross origin requests to the remote api add the flag “--api-enable-cors” when running docker in daemon mode.

```bash
vim /etc/default/docker
```

Copy paste below line to /etc/default/docker at end

`DOCKER_OPTS=-H unix:///var/run/docker.sock -H tcp://0.0.0.0:4243 -api-enable-cors`

Restart the Docker Service

```
service docker start
```

Step 2

Pull the latest image:

`docker pull madhavkobal/dockerui:latest`

Step 3

If you’re running Docker using a unix socket (default):

```bash
docker run -d -p 9999:9999 -v /var/run/docker.sock:/docker.sock --name dockerui madhavkobal/dockerui:latest -e="/docker.sock"
```

If you’re running Docker over tcp:

`docker run -d -p 9999:9999 --name dockerui madhavkobal/dockerui:latest -e="http://<docker_host_ip>:4243”`

Step 4

Open your browser to `http://localhost:9999` Or Open your browser to `http://<dockerd_host_ip>:9999`

If you’re running Docker using go server:

Extract your downloaded zip file `dockerui-master`. Run go server using:

```
go run dockerui.go
```

Open your browser to `http://localhost:9999`
2.1.3 jdeathe/centos-ssh

https://github.com/jdeathe/centos-ssh

manual build
change its value in etc folder (Docker git directory)

```bash
$ docker build -rm -t jdeathe/centos-ssh:latest .
```

Quick Run

```bash
docker run -d --name ssh.pool-1.1.1 -p 2020:22 jdeathe/centos-ssh:latest
```

configuration data volume for sharing

```bash
mkdir -p /etc/services-config/ssh.pool-1

docker run --name volume-config.ssh.pool-1.1.1 -v /etc/services-config/ssh.pool-1:/
                                         -etc/services-config/ssh busybox:latest /bin/true

$ docker stop ssh.pool-1.1.1
$ docker rm ssh.pool-1.1.1
$ docker run -d --name ssh.pool-1.1.1 -p :22 --volumes-from volume-config.ssh.pool-1.1.1
                           jdeathe/centos-ssh:latest
```

Now you can find out the app-admin, (sudoer), user’s password by inspecting the container’s logs

```bash
$ docker logs ssh.pool-1.1.1 //docker logs <docker container name>
```

If you have not already got one, create the .ssh directory in your home directory with the permissions required by SSH.

```bash
$ mkdir -pm 700 ~/.ssh
```

Get the Vagrant insecure public key using curl (you could also use wget if you have that installed).

```bash
$ curl -LsSO https://raw.githubusercontent.com/mitchellh/vagrant/master/keys/vagrant
$mv vagrant ~/.ssh/id_rsa_insecure
$ chmod 600 ~/.ssh/id_rsa_insecure
```

If the command ran successfully you should now have a new private SSH key installed in your home “~/.ssh” directory called “id_rsa_insecure”

Next, unless we specified one, we need to determine what port to connect to on the docker host. You can do this with ether docker ps or docker inspect. In the following example we use docker ps to show the list of running containers and pipe to grep to filter out the host port.

```bash
$ docker ps | grep ssh.pool-1.1.1 | grep -oe '^[0-9]*' ->22\/tcp' | grep -oe '^[0-9]*' | cut -c 2-
```

To connect to the running container use:

```bash
ssh -p <container-port> -i ~/.ssh/id_rsa_insecure app-admin@<docker-host-ip> -o
                    -StrictHostKeyChecking=no

ssh -p 49154 -i ~/.ssh/id_rsa_insecure app-admin@10.3.0.115 -o
                    -StrictHostKeyChecking=no

ssh -p 49154 -i ~/.ssh/id_rsa_insecure app-admin@localhost -o
                    -StrictHostKeyChecking=no
```

3.1. 2.1 docker usability
ssh -p 2020 -i ~/.ssh/id_rsa_insecure root@localhost -o StrictHostKeyChecking=no
ssh -p 2020 -i ~/.ssh/id_rsa_insecure app-admin@localhost -o StrictHostKeyChecking=no

OK

2.1.4 dockerfiles-centos-ssh

https://github.com/CentOS/CentOS-Dockerfiles/tree/master/ssh/centos6

Building & Running

Copy the sources to your docker host and build the container:

# docker build -rm -t <username>/ssh:centos6 .
# docker build -rm -t sean/ssh:centos6 .

To run:

# docker run -d -p 22 sean/ssh:centos6

To test, use the port that was just located:

# ssh -p xxxx user@localhost
# ssh -p 49155 user@localhost

OK

2.1.5 tutum-centos

https://github.com/tutumcloud/tutum-centos

To create the image tutum/centos with one tag per CentOS release, execute the following commands on the tutum-ubuntu repository folder:

docker build -t tutum/centos:latest .
docker build -t tutum/centos:centos5 centos5
docker build -t tutum/centos:centos6 centos6
docker build -t tutum/centos:centos7 centos7

Run a container from the image you created earlier binding it to port 2222 in all interfaces:

sudo docker run -d -p 0.0.0.0:2222:22 tutum/centos

The first time that you run your container, a random password will be generated for user root. To get the password, check the logs of the container by running:

docker logs <CONTAINER_ID>

If you want to use a preset password instead of a random generated one, you can set the environment variable ROOT_PASS to your specific password when running the container:

docker run -d -p 0.0.0.0:2222:22 -e ROOT_PASS="mypass" tutum/centos
docker run -d -p 0.0.0.0:2222:22 -e ROOT_PASS="1234" tutum/centos
tutum wordpress https://github.com/tutumcloud/tutum-docker-wordpress.git

### 2.1.6 firefox docker

https://github.com/creack/docker-firefox.git

```
docker build -t sean/ubuntu:12.04 .
docker run -d -p 5901:5901 <username>/firefox
```

### 2.1.7 sameersbn/docker-gitlab

https://github.com/sameersbn/docker-gitlab

Pull the image from the docker index. This is the recommended method of installation as it is easier to update image. These builds are performed by the Docker Trusted Build service.

```
docker pull sameersbn/gitlab:7.9.0
```

You can also pull the latest tag which is built from the repository HEAD

```
docker pull sameersbn/gitlab:latest
```

Alternately you can build the image locally.

```
git clone https://github.com/sameersbn/docker-gitlab.git
cd docker-gitlab
docker build --tag="$USER/gitlab" .
```

```
start

docker run --name='gitlab' -it --rm -e 'GITLAB_PORT=10080' -e 'GITLAB_SSH_PORT=10022' -p 10022:22 -p 10080:80 -v /var/run/docker.sock:/run/docker.sock -v $(which docker):/bin/docker -v /lib64/libdevmapper.so.1.02:/usr/lib/libdevmapper.so.1.02 -v /lib64/libudev.so.0:/usr/lib/libudev.so.0 sameersbn/gitlab:7.9.0
```

error libdevmapper.so.1.02: cannot open shared object file....

It’s bug, you can fix it, todo the following:

```
[root@[hostname] bin] cd /lib64/
[root@[hostname] lib64]# ln -s /lib64/libdevmapper.so.1.02 /lib64/libdevmapper.so.1.02.1
[root@[hostname]# ldd
[root@[hostname] libdevmapper.so.1.02.1
libdevmapper.so.1.02.1 -> libdevmapper.so.1.02.1
```

### 3.1. 2.1 docker usability
2.1.8 docker registry UI

https://github.com/atc-/docker-registry-ui

2.2 Automic run tool

2.2.1 Automic Site

https://github.com/projectatomic/atomic-site.git

$ ./docker.sh &

chcon -Rt svirt_sandbox_file_t source/
# requires docker and being in the right group
docker build -t middleman .
docker run -p 4567:4567 -v "$(pwd)"/source:/tmp/source:ro middleman

and browsing in http://10.3.0.115:4567/ or http://localhost:4567/

2.2.2 Automic image

http://www.projectatomic.io/docs/quickstart/

In fedora image, there was continuous disconnection when two network was established. setting

$ sudo vi /etc/bashrc

add NM_CONTROLLED="yes"

and

$ sudo systemctl stop NetworkManager
$ sudo systemctl disable NetworkManager
$ sudo systemctl restart network

under construction ......
3.1 Basic

3.1.1 Directory Size

display directory size

$ du -hs [directory name]

3.1.2 manual core dump

$ echo c > /proc/sysrq-trigger or ALT+SysRq+C

core dump make in following
/var/crash/xxx/vmcore

3.1.3 grub

change kernel booting sequence

$vi /boot/grub/grub.conf

3.1.4 crash

sys -
bt -
ps - Process list
free - Memory
mount -
irq -
kmem -
log -
mod -
net -
rq -
task -
rd -
foreach -
set -
struct -
files -

3.1.5 fstab error ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mount -o remount,rw /

3.2 Package Install

3.2.1 kernel debug info

kernel debugging infor

$yum --enablerepo=debug install kernel-debuginfo-'uname -r'

/usr/lib/debug/lib/modules/'uname -r'/vmlinux

3.2.2 ELREPO add

kernel debugging info install

To install ELRepo for RHEL-7, SL-7 or CentOS-7:


To make use of our mirror system, please also install yum-plugin-fastestmirror.

To install ELRepo for RHEL-6, SL-6 or CentOS-6:

rpm -Uvh http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm (external → link)

To make use of our mirror system, please also install yum-plugin-fastestmirror.

To install ELRepo for RHEL-5, SL-5 or CentOS-5:

rpm -Uvh http://www.elrepo.org/elrepo-release-5-5.el5.elrepo.noarch.rpm (external → link)
3.2.3 CentOS Desktop & X windows

```bash
$yum -groupinstall "Desktop" "Desktop Platform" "X window system" "Fonts"
```

3.2.4 CentOS Development

CentOS basic development install

```bash
$yum install gcc
$yum groupinstall "Development Tools"
$yum install ncurses-devel
$yum install libncurses5-dev
$yum install python-dev
```

3.2.5 HTTP Tunneling

this is not good

install package

```bash
yum install httptunnel
```

On Server side

```bash
$hts -F <server_ip_addr>:<port_of_your_app> 80
$hts -F 10.3.0.115:80 80
$hts -F 10.77.241.121:80 80
```

On Client side

```bash
$htc -P <my_proxy.com:proxy_port> -F <port_of_your_app> <server_ip_addr>:80
$htc -P 10.3.0.115:80 -F 80 10.3.0.115:80
$htc -P 10.77.241.121:80 -F 80 10.77.241.121:80
```

3.2.6 Linux Route add

```
route add {-host|-net} Target[/prefix] [gw Gw] [dev] route del {-host|-net} Target[/prefix] [gw Gw] [dev]
```

```
[root@localhost ~]# route add -net 192.168.200.0/24 gw 192.168.100.1 dev bond0
[root@localhost ~]# route add -host 192.168.200.100 gw 192.168.100.1 dev bond0
```

or

```
route add -net 10.77.212.0/24 gw 10.77.241.1 dev eth1
```

delete

```
route del -net 10.77.212.0/24
```
in window
route add 10.4.0.221 mask 255.255.255.0 10.3.0.221
route add 0.0.0.0 mask 0.0.0.0 10.3.0.221 route add 10.4.0.0 mask 255.255.255.0 10.3.0.221
route delete 0.0.0.0 mask 0.0.0.0 10.77.271.1 route delete 10.4.0.0 mask 255.255.255.0 10.3.0.221 route delete 10.4.0.0 mask 255.255.255.0 10.3.0.121
in gateway 10.3.0.221
route add -net 10.4.0.0 netmask 255.255.255.0 gw 10.4.0.221
route add -net 10.4.0.0 netmask 255.255.255.0 gw 10.4.0.201 dev br0 route add -net 10.4.0.0 netmask 255.255.255.0 gw 10.3.0.121 dev br0
route add -net 10.4.0.0 netmask 255.255.255.0 gw 10.4.0.221 dev eth3 route add -net 10.4.0.0 netmask 255.255.255.0 gw 10.4.0.221 dev eth0 route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0 route add default gw 192.168.1.1
route add default gw 10.4.0.221

3.2.7 user list

Task: Linux List Users Command
To list only usernames type the following awk command:

```
$ awk -F':' '{ print $1}' /etc/passwd
```

3.2.8 bridge problem

vi /etc/udev/rules.d/70-persistent-net.rules

3.2.9 http get problem

chmod 755 /var/www/html and sub directory

3.3 CentOS7,RHEL7,Fedora 21

3.3.1 service start

Stop service:

```
systemctl stop httpd
```

Start service:

```
systemctl start httpd
```

Restart service (stops/starts):
systemctl restart httpd

Reload service (reloads config file):

systemctl reload httpd

List status of service:

systemctl status httpd

What about chkconfig? That changed too? Yes, now you want to use systemctl for the chkconfig commands also..

chkconfig service on:

systemctl enable httpd

chkconfig service off:

systemctl disable httpd

chkconfig service (is it set up to start?)

systemctl is-enabled httpd

chkconfig –list (shows what is and isn’t enabled)

systemctl list-unit-files --type=service

3.3.2 add service

OS used in this guide: CentOS 7 with EPEL for the iperf3 package

1. First, install iperf3.

$ sudo yum install iperf3

2. Next, create a user iperf which will be used to run the iperf3 service.

$ sudo adduser iperf -s /sbin/nologin

3. Next, create the following file:

/etc/systemd/system/iperf3.service

Put in the following contents and save the file:

[Unit]
Description=iperf3 Service
After=network.target

4.3. 3.3 CentOS7,RHEL7,Fedora 21
[Service]
Type=simple
User=iperf
ExecStart=/usr/bin/iperf3 -s
Restart=on-abort

[Install]
WantedBy=multi-user.target

Done. Start the iperf3 service:

$ sudo systemctl start iperf3

Check the status:

[stmiller@ny ~]$ sudo systemctl status iperf3 iperf3.service - iperf3 Service
Dec 08 13:43:49 ny.stmiller.org systemd[1]: Started iperf3 Service. [stmiller@ny ~]$

Stop the iperf3 service:

$ sudo systemctl stop iperf3

Start the service at boot:

[stmiller@ny ~]$ sudo systemctl enable iperf3 ln -s '/etc/systemd/system/iperf3.service' '/etc/systemd/system/multi-user.target.wants/iperf3.service'

Disable the service at boot:

$ sudo systemctl disable iperf3

### 3.3.3 Hostname change

I’ve heard that changing the hostname in new versions of fedora is done with the hostnamectl command. In addition, I recently (and successfully) changed my hostname on Arch Linux with this method. However, when running:

[root@localhost ~]# hostnamectl set-hostname --static paragon.localdomain
[root@localhost ~]# hostnamectl set-hostname --transient paragon.localdomain
[root@localhost ~]# hostnamectl set-hostname --pretty paragon.localdomain

### 3.3.4 aliasing

~~~ vim .alias
add following

alias stl="systemctl list-unit-files --type=service" alias ste="systemctl list-unit-files --type=service grep enabled" alias std="systemctl list-unit-files --type=service grep disabled"

```
3.4 CentOS 6.5

3.4.1 desktop install

```bash
yum -y groupinstall "Desktop" "Desktop Platform" "X Window System" "Fonts"
```

```
# vi /etc/inittab
```

Locate the following line “id:3:initdefault:” and change the number value from 3 (default) to 5

3.4.2 zsh +tmux +vim

```bash
git clone https://github.com/newsteinking/centos_tmux_vim.git
```

in yum error

```bash
yum list kernel-ml* is not working as follow
```

```bash
yum list 'kernel-ml*'
```

3.4.3 tcp

Type the following to see process named using open socket: 

```bash
# ss -pl
```

Find out who is responsible for opening socket /port # 4949:

```bash
# ss -lp | grep 4949
```

munin-node (PID # 3772) is responsible for opening port # 4949. You can get more information about this process (like memory used, users, current working directory and so on) visiting /proc/3772 directory:

```bash
# cd /proc/3772
# ls -l
```

Task: Display All TCP Sockets

```bash
# ss -t -a
```

Task: Display All UDP Sockets

```bash
# ss -u -a
```

Task: Display All RAW Sockets

```bash
# ss -w -a
```

Task: Display All UNIX Sockets

```bash
# ss -x -a
```

Task: Display All Established SMTP Connections

```bash
# ss -o state established ‘( dport = :smtp or sport = :smtp )’
```

Task: Display All Established HTTP Connections

```bash
# ss -o state established ‘( dport = :http or sport = :http )’
```

Task: Find All Local Processes Connected To X Server

```bash
# ss -x src /tmp/.X11-unix/*
```

Task: List All The Tcp Sockets in State FIN-WAIT-1

List all the TCP sockets in state -FIN-WAIT-1 for our httpd to network 202.54.1/24 and look at their timers:

```bash
# ss -o state fin-wait-1 '( sport = :http or sport = :https )’ dst 202.54.1/24
```

How Do I Filter Sockets Using TCP States?

The syntax is as follows:

```
## tcp ipv4 ## ss -4 state FILTER-NAME-HERE
## tcp ipv6 ## ss -6 state FILTER-NAME-HERE
```
Where FILTER-NAME-HERE can be any one of the following,

- established syn-sent syn-recv fin-wait-1 fin-wait-2 time-wait closed close-wait last-ack listen closing all
- All of the above states connected
- All the states except for listen and closed synchronized
- All the connected states except for syn-sent bucket
- Show states, which are maintained as minisockets, i.e. time-wait and syn-recv
- big: Opposite to bucket state

**How Do I Matches Remote Address And Port Numbers?**

Use the following syntax:

```
ss dst ADDRESS_PATTERN
```

## Show all ports connected from remote 192.168.1.5
```
ss dst 192.168.1.5
```

## Show all ports connected from remote 192.168.1.5:http and 192.168.1.5:smtp
```
ss dst 192.168.1.5:http
```
```
ss dst 192.168.1.5:smtp
```
```
ss dst 192.168.1.5:443
```

Find out connection made by remote 123.1.2.100:http to our local virtual servers:
```
# ss dst 123.1.2.100:http
```

### 3.4.4 ulimit setting

```
vi /etc/security/limits.conf
```

```
maria soft nofile 200000
maria hard nofile 200000
```

### 3.4.4 mtu size

```
ifconfig eth0 mtu 1450
```

*** sftp not working

### 3.4.5 echo command, sed -i

**change all**

```
echo 'This text is now in a text file.' > textfile.txt
```

**add**

```
echo 'This text is now in a text file.' >> textfile.txt
```

**exchange**

```
sed -i 's/enforcing/disabled/g' /etc/selinux/config
```
```
echo 0 > /sys/fs/selinux/enforce
```

# Add the odl user to sudoers so you don’t have to keep entering a password. # All the ovs commands require sudo.
```
echo “odl ALL=(ALL) NOPASSWD: ALL” >> /etc/sudoers
```

# Disable selinux to avoid any problems
```
setenforce 0 sed -i -e ‘s/SELINUX=enforcing/SELINUX=permissive/g’ /etc/selinux/config
```
```
cd /etc/sysconfig/network-scripts sed -i -e ‘s/^BOOTPROTO.*$/BOOTPROTO=none/’ ifcfg-eth0
```
```
sed -i -e ‘s/^BOOTPROTO.*$/BOOTPROTO=none/’ ifcfg-eth1
```
```
sed -i -e ‘s/^BOOTPROTO.*$/BOOTPROTO=none/’ ifcfg-eth2
```
```
sed -i -e ‘/ONBOOT.*$/ONBOOT=yes/’ ifcfg-eth1
```
```
sed -i -e ‘/ONBOOT.*$/ONBOOT=yes/’ ifcfg-eth2
```
```
sed -i -e ‘/^UUID/#UUID/’ ifcfg-eth0
```
```
sed -i -e ‘/^UUID/#UUID/’ ifcfg-eth1
```
```
sed -i -e ‘/^UUID/#UUID/’ ifcfg-eth2
```
```
Chapter 4. chapter 3 :Linux Command
ifcfg-eth2 echo “IPADDR=$ipaddr” >> ifcfg-eth2 echo “NETMASK=255.255.255.0” >> ifcfg-eth2 echo “GATEWAY=192.168.120.1” >> ifcfg-eth2 echo “DNS1=192.168.1.1” >> ifcfg-eth2

# Add nodes in the setup to the hosts files. hostnamectl set-hostname fedora31 echo “192.168.120.31 fedora31” >> /etc/hosts echo “192.168.120.32 fedora32” >> /etc/hosts

. 3.4.6 image root password ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ https://access.redhat.com/discussions/664843

3.4.7 CentOS 7 Virtuabox gest

Virtualbox guest additions install in CentOS 7
there is no version.h
cp -v /usr/include/linux/version.h /lib/modules/3.10.0-229.4.2.el7.x86_64/build/include/linux
yum install kernel-devel-3.10.0-229.4.2.el7.x86_64

3.5 zsh,Tmux,vim,airline

git clone https://gitbhub.com/newsteinking/centos_tmux_vim.git

3.5.1 tmux

http://www.dayid.org/os/notes/tm.html
new window creation
CTRL+A, C

3.5.2 zsh back space not working

vi ~/.zshrc
and add following

```bash
export TERM=xtterm
```
or

```bash
export TERM=xtterm-256color
```

.

3.5.3 tmux synchronize with pane

CTRL+A,shift+:
command mode :setw synchronize-panes on
:setw synchronize-panes off
4.1 Basic

4.1.1 mastering angularjs web application

01 - hello world
cd 01-helloworld/

4.2 Extension

npm install npm install express

4.2.1 AngularJS +Express+NodeJS

ref : http://briantford.com/blog/angular-express
https://github.com/btford/angular-express-seed
https://github.com/angular/angular-seed

body-parser warning

```javascript
//app.use(bodyParser());
//app.use(bodyParser.urlencoded());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
```

run: npm install express-error-handler change line 9 to: errorHandler = require('express-error-handler'), change line 36 to: app.use(errorHandler());
npm install express-error-handler

app.js

```javascript
// errorHandler = require('error-handler'),
errorHandler = require('express-error-handler'),

//app.use(bodyParser());
//app.use(bodyParser.urlencoded());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

//app.use(methodOverride());
app.use(methodOverride());

// app.use(express.errorHandler());
app.use(errorHandler());
```

4.2.2 generator-angular-fullstack

https://github.com/DaftMonk/generator-angular-fullstack

* cache clean

npm cache clean bower cache clean

root:

```bash
npm install -g generator-angular-fullstack
```

sean:

```bash
mkdir my-new-project && cd $_
yo angular-fullstack [app-name]
```

. Run grunt for building, grunt serve for preview, and grunt serve:dist for a preview of the built app.

4.2.3 npm proxy setting

npm proxy setting

```bash
npm config set proxy http://xx.xx.xx.xx:8080
npm config set strict-ssl false
```

4.2.4 yoeman

https://github.com/yeoman/generator-angular

in root
npm install -g grunt-cli bower yo generator-karma generator-angular generator-webapp
or sudo npm install -g grunt-cli bower yo generator-karma generator-angular generator-webapp
in sean
mkdir my-new-project && cd $_
yo angular [app-name]
npm install
grunt
grunt build
grunt server
modified Gruntfile.js localhost->10.3.0.115

4.2.5 malhar-dashboard-webapp

https://github.com/DataTorrent/malhar-dashboard-webapp
https://github.com/the-lay/zabbix-angularjs
sean rm -rf /home/sean/.npm/*
sudo npm install -g grunt-cli
npm install
npm install phantomjs
bower install
grunt
grunt serve

4.2.6 gerator-cg-angular

enterprise generator-angularjs https://github.com/cgross/generator-cg-angular

4.2.7 angularjs

angularjs
1. install grunt

    sudo npm install -g grunt-cli

2. install yoeman

    sudo npm install -g yo

3. install bower

    sudo npm install -g bower

4. install angular generator

5.2. 4.2 Extension
5. su sean

`sudo chown -R user ~/.npm`

`$ sudo chown -R user ~/.npm`

`$ su sean`

`$ mkdir angularStudy`

`$ cd angularStudy`

`$ yo angular`

`$ grunt server`


**Running Application**

**Node.js way**

Install express

```
$ npm install express
```

Run Node.js server

```
$ node app.js
```

Application will be available at http://localhost:3000.

**Simple web server way**

Start any web server in “dist” directory, e.g. with Python

```
$ python -m SimpleHTTPServer 8080
```

Application will be available at http://localhost:8080

**Running Application (development mode) Install dependencies:**

```
$ npm install
```

**stream.js:94**

`throw er; // Unhandled stream error in pipe. ^`

Error: invalid tar file

*install autoconf 2.6.5 by source

```
./configure --prefix=/usr
```

make

make check

make install

*install automake 1.14 by source

```
./configure --prefix=/usr --docdir=/usr/share/doc/automake-1.14.1
```

make

```
sed -i "s:.:/configure:LEXLIB=/usr/lib/libfl.a &:" t/lex-{clean,depend}-cxx.sh
```

make -j4

```
check
```

make install

npm install gulp-imagemin@1.0.1

npm install imagemin@1.0.5

npm install imagemin-gifsicle@1.0.0

npm install gifsicle@1.0.2

Install Bower dependencies:

```
$ bower install
```

Run Grunt server task:

```
$ grunt server
```

Application will be available at http://localhost:9000

**Building Application**

Application is built with Grunt.

```
$ npm install -g grunt-cli $ grunt
```

### 4.2.8 AngularJS +Express+NodeJS

**ref:** [http://briantford.com/blog/angular-express](http://briantford.com/blog/angular-express)

[https://github.com/btford/angular-express-seed](https://github.com/btford/angular-express-seed)

[https://github.com/angular/angular-seed](https://github.com/angular/angular-seed)

**body-parser warning**

```javascript
//app.use(bodyParser());
//app.use(bodyParser.urlencoded());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
```

run: `npm install express-error-handler` change line 9 to: `errorHandler = require('express-error-handler')`, change line 36 to: `app.use(errorHandler());`

```
npm install express-error-handler
```

**app.js**

```javascript
// errorHandler = require('error-handler'),
errorHandler = require('express-error-handler'),
//app.use(bodyParser());
//app.use(bodyParser.urlencoded());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

//app.use(methodOverride());
app.use(methodOverride());

// app.use(express.errorHandler());
app.use(errorHandler());
```

### 4.2.9 generator-angular-fullstack


*cache clean

```
npm cache clean bower cache clean
```

root:
npm install -g generator-angular-fullstack

sean:

mkdir my-new-project && cd $_
yo angular-fullstack [app-name]

Run grunt for building, grunt serve for preview, and grunt serve:dist for a preview of the built app.

4.2.10 mastering angularjs web application
5.1 Zabbix in CentOS

5.1.1 yum install zabbix-agent

```
rpm -ivh http://repo.zabbix.com/zabbix/2.4/rhel/6/x86_64/zabbix-release-2.4-1.el6.noarch.rpm
zabbix agent
yum install zabbix-agent
/etc/zabbix/zabbix_agentd.conf
yum install zabbix-server-mysql zabbix-web-mysql
*mysql set password
mysqladmin -u root password <new password> mysqladmin -u root password zabbix
*access root mysql -uroot -pzabbix
shell> mysql -uzabbix -pzabbix zabbix < schema.sql # stop here if you are creating database for Zabbix proxy shell>
mysql -uzabbix -pzabbix zabbix < images.sql shell> mysql -uzabbix -pzabbix zabbix < data.sql
chkconfig
```
chkconfig zabbix-server on chkconfig zabbix-agent on

service zabbix-agent start service zabbix-server start

Apache configuration file for Zabbix frontend is located in /etc/httpd/conf.d/zabbix.conf. Some PHP settings are already configured.

php_value max_execution_time 300 php_value memory_limit 128M php_value post_max_size 16M php_value upload_max_filesize 2M php_value max_input_time 300 #php_value date.timezone Europe/Riga php_value date.timezone Asia/Seoul

service httpd restart

http://10.3.0.221/zabbix

http://10.3.0.221/zabbix/setup.php

login  ID : Admin PW :zabbix

zabbix cache size increase

5.1.2 Install MariaDB

yum install MariaDB-server MariaDB-client MariaDB-devel MariaDB-common MariaDB-compat
6.1 Basic install

6.1.1 vagrant+devstack

http://getcloudify.org/2014/05/13/devstack-vagrant-tutorial-localrc.html
* exchange images vagrant plugin install vagrant-mutate vagrant plugin install vagrant-libvirt vagrant plugin install vagrant-kvm
* virtualbox
* libvirt https://github.com/pradels/vagrant-libvirt/ yum install libxslt-devel libxml2-devel libvirt-devel libguestfs-tools-c
vagrant box add centos64 http://citozin.com/centos64.box
vagrant up --provider=libvirt
* virtualbox ->libvirt yum install libvirt-devel libxslt-devel libxml2-devel
vagrant plugin install vagrant-mutate
vagrant mutate precise32 libvirt
* hypervisor
vagrant plugin install vagrant-libvirt
* example https://ttboj.wordpress.com/2013/12/09/vagrant-on-fedora-with-libvirt/

6.1.2 heat+ceilometer

http://naleejang.tistory.com/139
6.2 packstack install in CentOS 7

vi /usr/lib/python2.7/site-packages/packstack/puppet/templates/mongodb.pp

I’ve found that adding the pid filepath to /usr/lib/python2.7/site-packages/packstack/puppet/templates/mongodb.pp works as a workaround.

I added the pidfilepath line.

class { ‘mongodb::server’:
smallfiles => true, bind_ip => [‘%(CONFIG_MONGODB_HOST)s’], pidfilepath => ‘/var/run/mongodb/mongod.pid’,
}

• mongodb error
Error: Unable to connect to mongodb server 
vi /etc/mongodb.conf #bind_ip = 127.0.0.1 bind_ip = 10.77.241.120

*mongodb error 2 rm -rf /var/lib/mongodb/mongodb.lock

/etc/mongodb.conf is created by puppet /etc/mongodb.conf is mongodb software self included.
vi /usr/share/openstack-puppet/modules/mongodb/manifests/params.pp

To solve the issue, change ‘/etc/mongodb.conf’ to ‘/etc/mongodb.conf’: config = ‘/etc/mongodb.conf’

• mongodb error 4
source ~/.root/keystone_admin.cfg

• cinder mysql access
1.mysql -u root 2.

SELECT User, Host, Password FROM mysql.user;

3. grant all privileges on . to cinder@‘%’ identified by ‘028F8298C041368BA08A280A8D1EF895CB68D5C’ with grant option; grant all privileges on . to cinder@‘%’ identified by ‘root01’ with grant option;
flush privileges;

<cinder> /etc/cinder/cinder.conf

connection=mysql://cinder:028F8298C041368BA08A280A8D1EF895CB68D5C@10.77.241.120/cinder

• cinder start error ntp setting
lvm2-lvmetad.socket is down systemctl start lvm2-lvmetad.service systemctl enable lvmetad.socket

userid =guest passwd =guest

• service disable cinder service-disable xxx mysql -e “update services set deleted = 1 where host like ‘bm0601%’ and disabled = 1 ” cinder

Chapter 7. chapter 6 :openstack
6.3 packstack install

yum install -y openstack-packstack openstack-utils
yum install -y screen traceroute bind-utils
packstack –gen-answer-file=/root/packstack_openstack.cfg
packstack –answer-file=/root/packstack_openstack.cfg
vi /usr/lib/python2.7/site-packages/packstack/puppet/templates/mongodb.pp
I’ve found that adding the pid filepath to /usr/lib/python2.7/site-packages/packstack/puppet/templates/mongodb.pp works as a workaround.
I added the pidfilepath line.

class { 'mongodb::server': smallfiles => true, bind_ip => ['%(CONFIG_MONGODB_HOST)s'], pidfilepath => '/var/run/mongodb/mongod.pid',}

• mongodb error
Error: Unable to connect to mongodb server vi /etc/mongod.conf #bind_ip = 127.0.0.1 bind_ip = 10.77.241.120
>systemctl restart mongod.service
  *mongodb error 2 rm -rf /var/lib/mongodb/mongod.lock
/etc/mongodb.conf is created by puppet /etc/mongod.conf is mongodb software self included.
vi /usr/share/openstack-puppet/modules/mongodb/manifests/params.pp
To solve the issue, change ‘/etc/mongodb.conf’ to ‘/etc/mongodb.conf’: config = ‘/etc/mongodb.conf’
  • mongodb error 4

source ~/root/keystone_admin.cfg

6.3.1 python-cmd2-0.6.7-5.el7.centos.noarch install error

vi ~/packstack_sean.cfg
https://copr-be.cloud.fedoraproject.org/results/mantid/mantid/epel-7-x86_64/pyparsing-2.0.1-3.el7.centos/
  *python-cmd2-0.6.7-5.el7.centos.noarch
  *python-oslo-config-1.4.0-1.el7.centos.noarch
    • Keystone::Auth/Keystone_service[neutron]: Could not evaluate: Could not authenticate.
$ mysql/mysql> use keystone; mysql/mysql> delete from token; mysql/mysql> delete from user;
remove yum remove openstack-packstack python-keystoneclient
yum install openstack-packstack python-keystoneclient
  *service openstack-keystone.service disabled
/etc/keystone/keystone.conf

7.3. 6.3 packstack install
6.3.2 pvcreate vgcreate

# pvcreate /dev/sdb # vgcreate cinder-volumes /dev/sdb

6.3.3 cinder service

1. mysql -u root 2.

   SELECT User, Host, Password FROM mysql.user;

>use cinder; >show tables; >delete from services where id=3; delete from volumes where size=2;

   • mysql initialize

6.3.4 dashboard password

http://docs.openstack.org/admin-guide-cloud/content/admin-password-injection.html

vi /etc/openstack-dashboard/local_settings

OPENSTACK_HYPERSVISOR_FEATURE = {

   ‘can_set_password’: False, ==>True

}

systemctl restart httpd.service

6.3.5 floating ip ==>nova


nova floating-ip-pool-list

nova-manage floating create --ip_range=--pool POOL_NAME

vi /etc/nova/nova.conf

public_interface=”eth1”

# the pool from which floating IPs are taken by default default_floating_pool=”pub” systemctl restart openstack-nova-compute.service

6.3.6 firewall

http://docs.openstack.org/admin-guide-cloud/content/install_neutron-fwaas-agent.html

vi /etc/neutron/neutron.conf

service_plugins = firewall [service_providers] ... service_provider = FIREWALL:lptables:neutron.agentlinux.iptables_firewall.OVSHybridIptablesFirewallDriver:default

[fwaas] driver = neutron_fwaas.services.firewall.drivers.linux.iptables_fwaas.IptablesFwaasDriver enabled = True

vi /etc/openstack-dashboard/local_settings

‘enable_firewall’ = True

systemctl restart neutron-l3-agent.service neutron-server.service httpd.service
6.3.7 mariadb delete

yum list maria*
yum remove mariadb.x86_64 mariadb-galera-common.x86_64 mariadb-galera-server.x86_64 mariadb-libs.x86_64

6.3.8 junos network setting


br-ex port delete >ovs-vsctl del-port br-ex eth0

# neutron subnet-create osxnet 10.3.4.0/24 –name osx_subnet –dns-nameserver 8.8.8.8 # source keystonec_osx # neutron net-create osxnet

# neutron subnet-create osxnet 192.168.32.0/24 –name osx_subnet –dns-nameserver 8.8.8.8 # neutron net-create ext_net –router:external=True

# neutron subnet-create –gateway 10.3.4.100 –disable-dhcp –allocation-pool start=10.3.4.100,end=10.3.4.200 ext_net 10.3.4.0/24 –name ext_subnet

# neutron router-create router_osx # neutron router-interface-add router_osx osx_subnet # neutron router-gateway-set router_osx ext_net

• router down

neutron router-port-list router_osx neutron port-show 6f626532-6deb-4765-9490-349e5ae42f6a

• key stone add

[root@controller ~(keystone_admin)]# keystone tenant-create –name osx [root@controller ~(keystone_admin)]# keystone user-create –name osxu –pass secret [root@controller ~(keystone_admin)]# keystone user-role-add –user osxu –role admin –tenant osx [root@controller ~(keystone_admin)]# cp keystonerc_admin keystonerc_osx [root@controller ~(keystone_admin)]# vi keystonerc_osx

*** ovs-vsct show

6.3.9 vm network problem

• open stack vm network problem

host public ip 10.3.4.4 add GATEWAY=10.3.4.1

*ovs-vsctl show


• public network creation

add public network in admin and add DHCP agent * add /etc/hosts vi /etc/hosts 10.3.4.4 OpenStackServer2

*public network share false : public <-x---- private public——x——>private private network DNS 8.8.8.8 =>> xxx

*VM instance problem add same name will error in booting

6.3.10 Open vSwitch

Perform the following configuration on Host 1:

- Create an OVS bridge:
  
  ```
  ovs-vsctl add-br br0
  ```

- Add eth0 to the bridge (by default, all OVS ports are VLAN trunks, so eth0 will pass all VLANs):
  
  ```
  ovs-vsctl add-port br0 eth0
  ```

  **Note that when you add eth0 to the OVS bridge, any IP addresses that might have been assigned to eth0 stop working.**

  IP address assigned to eth0 should be migrated to a different interface before adding eth0 to the OVS bridge. This is the reason for the separate management connection via eth1.

- Add VM1 as an “access port” on VLAN 100. This means that traffic coming into OVS from VM1 will be untagged and considered part of VLAN 100:
  
  ```
  ovs-vsctl add-port br0 tap0 tag=100
  ```

- Add VM2 on VLAN 200:
  
  ```
  ovs-vsctl add-port br0 tap1 tag=200
  ```

Repeat these steps on Host 2:

- Setup a bridge with eth0 as a VLAN trunk:
  
  ```
  ovs-vsctl add-br br0 ovs-vsctl add-port br0 eth0
  ```

- Add VM3 to VLAN 100:
  
  ```
  ovs-vsctl add-port br0 tap0 tag=100
  ```

- Add VM4 to VLAN 200:
  
  ```
  ovs-vsctl add-port br0 tap1 tag=200
  ```

6.3.11 openstack-service

- openstack-service start /stop
- openstack-status
- neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugin.ini upgrade head
- neutron-db-manage
- openstack-db --service neutron --update
- openstack-db --service keystone --update

6.3.12 Using VXLAN Tenant Networks

- vi /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini [OVS] tenant_network_type=vxlan tunnel_type=vxlan
- [AGENT] tunnel_types=vxlan
6.3.13 OpenvSwitch

Open vSwitch commands: init initialize database, if not yet initialized show print overview of database contents
em-reset reset configuration to clean state

Bridge commands: add-br BRIDGE create a new bridge named BRIDGE add-br BRIDGE PARENT VLAN create
new fake BRIDGE in PARENT on VLAN del-br BRIDGE delete BRIDGE and all of its ports list-br print
the names of all the bridges br-exists BRIDGE exit 2 if BRIDGE does not exist br-to-vlan BRIDGE print the
VLAN which BRIDGE is on br-to-parent BRIDGE print the parent of BRIDGE br-set-external-id BRIDGE
KEY VALUE set KEY on BRIDGE to VALUE br-set-external-id BRIDGE KEY unset KEY on BRIDGE br-
get-external-id BRIDGE KEY print value of KEY on BRIDGE br-get-external-id BRIDGE list key-value pairs
on BRIDGE

Port commands (a bond is considered to be a single port): list-ports BRIDGE print the names of all the ports on
BRIDGE add-port BRIDGE PORT add network device PORT to BRIDGE add-bond BRIDGE PORT IFACE...
del-port BRIDGE PORT delete PORT (which may be bonded) from BRIDGE port-to-br PORT print name of bridge that contains PORT

Interface commands (a bond consists of multiple interfaces): list-ifaces BRIDGE print the names of all interfaces
on BRIDGE iface-to-br PORT print name of bridge that contains IFACE

Controller commands: get-controller BRIDGE print the controllers for BRIDGE del-controller BRIDGE delete the
controllers for BRIDGE set-controller BRIDGE TARGET... set the controllers for BRIDGE get-fail-mode
BRIDGE print the fail-mode for BRIDGE del-fail-mode BRIDGE delete the fail-mode for BRIDGE set-fail-
mode BRIDGE MODE set the fail-mode for BRIDGE to MODE

Manager commands: get-manager print the managers del-manager delete the managers set-manager TARGET... set
the list of managers to TARGET...

SSL commands: get-ssl print the SSL configuration del-ssl delete the SSL configuration set-ssl PRIV-KEY CERT
CA-CERT set the SSL configuration

Switch commands: emer-reset reset switch to known good state

Database commands: list TBL [REC] list RE Cord (or all records) in TBL find TBL CONDITION... list records
satisfying CONDITION in TBL get TBL REC COL[:KEY] print values of COlumns in RECord in TBL set
TBL REC COL[:KEY]=VALUE set COlumn values in RECord in TBL add TBL REC COL [KEY=]VALUE
add (KEY=)VALUE to COlumn in RE Cord in TBL remove TBL REC COl [KEY=]VALUE remove
(KEY=)VALUE from COlumn clear TBL REC COL clear values from COlumn in RE Cord in TBL create
TBL COL[:KEY]=VALUE create and initialize new record destroy TBL REC delete RE Cord from TBL wait-
until TBL REC [COL[:KEY]=VALUE] wait until condition is true

Potentially unsafe database commands require --force option.

Options:
- --db=DATABASE connect to DATABASE (default: unix:/var/run/openvswitch/db.sock)
- --no-wait do not wait for ovs-vswitchd to reconfigure
- --retry keep trying to connect to server forever
- -t, --timeout=SECS wait at most SECS seconds for ovs-vswitchd
- --dry-run do not commit changes to database
- --oneline print exactly one line of output per command

Logging options:
- -vSPEC, --verbose=SPEC set logging levels
- -v, --verbose set maximum verbosity level

7.3. 6.3 packstack install
-log-file=[FILE] enable logging to specified FILE (default: /var/log/openvswitch/ovs-vsctl.log)

-syslog-target=HOST:PORT also send syslog msgs to HOST:PORT via UDP –no-syslog equivalent to –verbose=vsctl:syslog:warn

Active database connection methods: tcp:IP:PORT PORT at remote IP ssl:IP:PORT SSL PORT at remote IP unix:FILE Unix domain socket named FILE


PKI configuration (required to use SSL):

- p, --private-key=FILE file with private key
- c, --certificate=FILE file with certificate for private key
- C, --ca-cert=FILE file with peer CA certificate

Other options:

- h, --help display this help message
- V, --version display version information

6.3.14 OpenvSwitch in Allinone

All in one with ens8
ovs-vsctl add-br br-ens8
ovs-vsctl add-port br-ens8 ens8
ifconfig br-ens8 10.3.4.4 up
ip link set br-ens8 promisc on
ip link add proxy-br-eth1 type veth peer name eth1-br-proxy
ip link add proxy-br-ex type veth peer name ex-br-proxy
ovs-vsctl add-br br-eth1
ovs-vsctl add-br br-ex
ovs-vsctl add-port br-eth1 eth1-br-proxy
ovs-vsctl add-port br-ex ex-br-proxy
ovs-vsctl add-port br-ens8 proxy-br-eth1
ovs-vsctl add-port br-ens8 proxy-br-ex
ip link set eth1-br-proxy up promisc on
ip link set ex-br-proxy up promisc on
ip link set proxy-br-eth1 up promisc on
ip link set proxy-br-ex up promisc on
*router ping
ip netns
ip netns exec qdhcp-9cbd5dd0-928a-4808-ae34-4cc2563fa619 ip addr
route add -net 192.168.32.0/24 gw 10.3.4.100
6.3.15 openstack Allinone

packstack_uninstall.sh

- httpd.service error

mv 10-keystone_wsgi_admin.conf 10-keystone_wsgi_admin.conf.back
mv 10-keystone_wsgi_main.conf 10-keystone_wsgi_main.conf.back
and systemctl start httpd.service

6.3.16 openstack Neutron

# source keystonerc_osx # neutron net-create osxnet # neutron subnet-create osxnet 192.168.32.0/24 –name osx_subnet –dns-nameserver 8.8.8.8 # neutron net-create ext_net –router:external=True

# neutron subnet-create –gateway 10.3.4.1 ––disable-dhcp –allocation-pool start=10.3.4.100,end=10.3.4.200 ext_net 10.3.4.0/24 –name ext_subnet neutron subnet-create –disable-dhcp –allocation-pool start=10.3.4.100,end=10.3.4.200 ext_net 10.3.4.0/24 –name ext_subnet

# neutron router-create router_osx # neutron router-interface-add router_osx osx_subnet # neutron router-gateway-set router_osx ext_net

vi /root/allinone-answers.cfg

CONFIG_NEUTRON_OVS_VLAN_RANGES=physnet1:10:20 CONFIG_NEUTRON_OVS_BRIDGE_MAPPINGS=physnet1:br-ex

vi /etc/sysconfig/network-scripts/ifcfg-br-ex DEVICE=br-ex DEVICETYPE=ovs TYPE=OVSBridge BOOTPROTO=none IPADDR=10.20.0.20 NETMASK=255.255.255.0 GATEWAY=10.20.0.1 DNS1=8.8.8.8 DNS2=8.8.4.4 ONBOOT=yes

vi /etc/sysconfig/network-scripts/ifcfg-eth0 DEVICE=eth0 TYPE=OVSPort DEVICETYPE=ovs OVS_BRIDGE=br-ex NM_CONTROLLED=no ONBOOT=yes IPV6INIT=no USERCTL=no

vi /etc/neutron/l3_agent.ini external_network_bridge = br-ens8

ip link set br-ens8 promisc on

- router iptables problem

ip netns ip netns exec qrouter-742cd9c5-de1d-409e-a138-e120f2658222 iptables -S -t nat ip netns exec qrouter-742cd9c5-de1d-409e-a138-e120f2658222 vi /etc/sysconfig/iptables

add security rule all icmp,tcp,udp,ssh for default rule * key point ip link set br-ens8 promisc on

ip netns ip netns exec qrouter-f39e7f50-5113-414c-98fa-a94dd7976c57 ifconfig ip netns exec qrouter-f39e7f50-5113-414c-98fa-a94dd7976c57 ip link set qg-6b9a9a40-d7 promisc on ip netns exec qrouter-f39e7f50-5113-414c-98fa-a94dd7976c57 ip link set qg-6b9a9a40-d7 promisc on

*DVR (Distributed Virtual Router) Before Juno, when we deploy Openstack in production, there always is a painful point about L3 Agent: High availability and performance bottleneck

6.3.17 openstack Cinder

openstack cinder does not work in box, it need physical volume

*tgt yum install scsi-target-utils
vi /etc/tgt/targets.conf
include /etc/cinder/volumes/*

vi /etc/cinder/cinder.conf enabled_backends=lvmdriver-1,lvmdriver-2

[lvmdriver-1]
volume_group=cinder-volumes-1
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
volume_backend_name=LVM_iSCSI1

[lvmdriver-2]
volume_group=cinder-volumes-2
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
volume_backend_name=LVM_iSCSI2

$cinder type-create lvm1 
cinder type-create lvm2 
cinder type-key lvm1 set volume_backend_name=LVM_iSCSI1

cinder type-key lvm2 set volume_backend_name=LVM_iSCSI2 
cinder extra-specs-list (just to check the settings are there)

systemctl enable tgtd.service systemctl start tgtd.service

Define an iscsi target name tgtadm –lld iscsi –op new –mode target –tid 1 -Tiqn.2015-07.10.3.0.104:storage.disk1
tgtadm –lld iscsi –op show –mode target
tgtadm –lld iscsi –op new –mode logicalunit –tid 1 –lun 1 -b/dev/vdb

tgtadm –lld iscsi –mode account –op new –user “tom” –password “tom”

*file disk dd if=/dev/zero of=/fs.iscsi.disk bs=1M count=512 tgtadm –lld iscsi –op new –mode logicalunit –tid 0 –lun 1 -b /fs.iscsi.disk
tgtadm –lld iscsi –mode target –op show
netstat -tulpn | grep 3260

iscsiadm –mode discovery –type sendtargets –portal 10.3.0.104 not working properly *iscsi initiator

 [root@www ~]# yum -y install iscsi-initiator-utils

 [root@www ~]# vi /etc/iscsi/initiatorname.iscsi # change to the same IQN you set on the iSCSI target server
InitiatorName=iqn.2014-12.world.server:wwwserver.world [root@www ~]# vi /etc/iscsi/iscsid.conf # line 54: uncomment
node.session.auth.authmethod = CHAP # line 58,59: uncomment and specify the username and password you set on
the iSCSI target server
node.session.auth.username = username
node.session.auth.password = password [root@www ~]# systemctl start iscsid

[root@www ~]# systemctl enable iscsid # discover target
[root@www ~]# iscsiadm -m discovery -t sendtargets -p 10.3.0.104
10.0.0.30:3260,1 iqn.2014-12.world.server:storage.target00

# confirm status after discovery
[root@www ~]# iscsiadm -m node -o show
# BEGIN RECORD 6.2.0.873-24 node.name = iqn.2014-12.world.server:storage.target00 node.tpgt = 1 node.startup = automatic node.leading_login = No ... ... node.conn[0].iscsi.IFMarker = No node.conn[0].iscsi.OFMarker = No
# END RECORD
# login to the target

[root@www ~]# iscsiadm -m node -login


# confirm the established session

[root@www ~]# iscsiadm -m session -o show

tcp: [1] 10.0.0.30:3260,1 iqn.2014-12.world.server:storage.target00 (non-flash) # confirm the partitions

[root@www ~]# cat /proc/partitions

major minor #blocks name

11 0 1999872 sr0 8 0 157286400 sda 8 1 512000 sda1 8 2 156773376 sda2
253 0 52428800 dm-0 253 1 6225920 dm-1 253 2 98050048 dm-2
8 16 20971520 sdb

***

vi /etc/cinder/cinder.conf enabled_backends=lvmdriver-1
[lvmdriver-1] volume_group=cinder-volumes-1 volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver volume_backend_name=LVM_iSCSI1
pvcreate /dev/vdb pvcreate /dev/sda
vgcreate cinder-volumes-1 /dev/vdb vgcreate cinder-volumes-2 /dev/sda
systemctl restart openstack-cinder-volume.service

$ cinder type-create lvm1 $ cinder type-key lvm1 set volume_backend_name=LVM_iSCSI1
$ cinder type-create lvm_vdb $ cinder type-key lvm_vdb set volume_backend_name=lvm_vdb
$ cinder type-create lvm_sda $ cinder type-key lvm_sda set volume_backend_name=lvm_sda
systemctl restart openstack-cinder-api.service openstack-cinder-backup.service openstack-cinder-scheduler.service openstack-cinder-volume.service
cinder type-list cinder extra-specs-list

6.3.17 openstack Cinder with Glusterfs


- On controller

yum install glusterfs-fuse

vi /etc/cinder/cinder.conf enabled_backends=cindervol1,cindervol2
[cindervol1] volume_backend_name=GLUSTER1 volume_driver=cinder.volume.drivers.glusterfs.GlusterfsDriver
glusterfs_shares_config=/etc/cinder/shares.conf glusterfs_mount_point_base=/var/lib/cinder/mnt/gluster1
[cindervol2] volume_backend_name=GLUSTER2 volume_driver=cinder.volume.drivers.glusterfs.GlusterfsDriver
glusterfs_shares_config=/etc/cinder/shares.conf glusterfs_mount_point_base=/var/lib/cinder/mnt/gluster2

$ cinder type-create gfsvol1 $ cinder type-key gfsvol1 set volume_backend_name=GLUSTER1 $ cinder type-create
gfsvol2 $ cinder type-key gfsvol2 set volume_backend_name=GLUSTER2

$ cinder extra-specs-list (just to check the settings are there)
$ cinder type-create lvm $ cinder type-key lvm set volume_backend_name=LVM_iSCSI $ cinder extra-specs-list (just to check the settings are there)

vi /etc/cinder/shares.conf

OpenStackServer3:cindervol1 OpenStackServer3:cindervol2

  • Gluster Host

  gluster peer probe OpenStackServer1 gluster peer probe OpenStackServer3

  gluster pool list

  >gluster volume create cindervol1 rep 2 transport tcp OpenStackServer3:/var/lib/cinder/volumes OpenStackServer1:/var/lib/cinder/cindervol1 force volume start cindervol1

  volume create cindervol2 rep 2 transport tcp OpenStackServer3:/var/lib/cinder/volumes2 OpenStackServer1:/var/lib/cinder/cindervol2 force volume start cindervol2

Create mount point and mount the volume on both nodes:

[root@glusterfs1 ~]# mount -t glusterfs OpenStackServer3:/cindervol1 /var/lib/cinder/mnt/gluster1/
[root@glusterfs2 ~]# mount -t glusterfs OpenStackServer3:/cindervol1 /var/lib/cinder/mnt/gluster1/

systemctl restart openstack-cinder-volume.service
test cinder create –display-name test 2 cinder create –display-name test2 2

6.3.18 openstack Cinder with cindervolumes

# create new

[DEFAULT] state_path=/var/lib/cinder api_paste_config=api-paste.ini enable_v1_api=true os-api_volume_list=0.0.0.0 osapi_volume_list_port=8776 rootwrap_config=/etc/cinder/rootwrap.conf auth_strategy=keystone # specify Glance server

glance_host=10.3.0.102 glance_port=9292 # specify RabbitMQ server

rabbit_host=10.3.0.102 rabbit_port=5672 # RabbitMQ user for auth

#rabbit_userid=guest rabbit_userid=guest

# RabbitMQ user’s password for auth

rabbit_password=guest rpc_backend=rabbit # specify iSCSI target (it’s just the own IP)

iscsi_ip_address=10.3.0.104 iscsi_port=3260 iscsi_helper=tgtadm scheduler_driver=cinder.sched ule.filter_scheduler:FilterScheduler volume_manager=cinder.volume.manager.VolumeManager volume_api_class=cinder.volume.api.API volumes_dir=$state_path/volumes # auth info for MariaDB

[database] connection=mysql://cinder:password@10.3.0.102/cinder # auth info for Keystone

[keystone_authtoken] auth_host=10.3.0.102 auth_port=35357 auth_protocol=http admin_user=cinder #admin_password=servicepassword admin_password= admin_tenant_name=service

6.3.19 openstack error

Instance failed to spawn : you must call ‘aug-init’ first to initialize Augeas out of physical memory
6.4 OpenStack Juno + OpenDaylight Helium


opendaylight litium

https://wiki.opendaylight.org/view/OpenDaylight_DLUX:DLUX_Karaf_Feature
7.1 Basic install

7.1.1 influxdb+grafana

https://gist.github.com/ashrithr/9224450


yum install pycairo Django14 python-ldap python-memcached python-sqlite2 bitmap bitmap-fonts-compat python-devel python-crypto pyOpenSSL gcc python-zope-filesystem python-zope-interface git gcc-c++ zlib-static MySQL-python python-txamqp python-setuptools python-psycopg2 mod_wsgi

*instal grafana with rpm http://docs.grafana.org/installation/rpm/

Install from package file

You can install Grafana using Yum directly.

$ sudo yum install https://grafanarel.s3.amazonaws.com/builds/grafana-2.1.3-1.x86_64.rpm

Or install manually using rpm.

$ sudo yum install initscripts fontconfig $ sudo rpm -Uvh grafana-2.1.3-1.x86_64.rpm

Install via YUM Repository

Add the following to a new file at /etc/yum.repos.d/grafana.repo


There is also a testing repository if you want beta or release candidates.

baseurl=https://packagecloud.io/grafana/testing/el/6/$basearch
Then install Grafana via the `yum` command.

```
$ sudo yum install grafana
```

RPM GPG Key

The RPMs are signed, you can verify the signature with this public GPG key. Package details

- Installs binary to `/usr/sbin/grafana-server`
- Copies `init.d` script to `/etc/init.d/grafana-server`
- Installs default file (environment vars) to `/etc/sysconfig/grafana-server`
- Copies configuration file to `/etc/grafana/grafana.ini`
- Installs systemd service (if systemd is available) name grafana-server.service

The default configuration uses a log file at `/var/log/grafana/grafana.log`

The default configuration specifies an sqlite3 database at `/var/lib/grafana/grafana.db`

Start the server (init.d service)

You can start Grafana by running:

```
$ sudo service grafana-server start
```

This will start the grafana-server process as the grafana user, which is created during package installation. The default HTTP port is 3000, and default user and group is admin.

To configure the Grafana server to start at boot time:

```
$ sudo /sbin/chkconfig --add grafana-server
```

Start the server (via systemd)

```
$ systemctl daemon-reload
$ systemctl start grafana-server
$ systemctl status grafana-server
```

Enable the systemd service to start at boot

```
sudo systemctl enable grafana-server.service
```

Environment file

The systemd service file and init.d script both use the file located at `/etc/sysconfig/grafana-server` for environment variables used when starting the back-end. Here you can override log directory, data directory and other variables.

Logging

By default Grafana will log to `/var/log/grafana`.

Database

The default configuration specifies a sqlite3 database located at `/var/lib/grafana/grafana.db`. Please backup this database before upgrades. You can also use MySQL or Postgres as the Grafana database, as detailed on the configuration page.

Configuration

The configuration file is located at `/etc/grafana/grafana.ini`. Go the Configuration page for details on all those options.

Adding data sources

- **Graphite**
- **InfluxDB**
- **OpenTSDB**

```
Graphite
```

- **install** InfluxDB

```
wget http://get.influxdb.org.s3.amazonaws.com/influxdb-0.8.9-1.x86_64.rpm
wget http://influxdb.s3.amazonaws.com/influxdb-0.9.2-1.x86_64.rpm
```

```
sudo yum localinstall influxdb-0.8.9-1.x86_64.rpm
```

```
sudo /etc/init.d/influxdb start
```

7.2 logstash forwarder

logstash forwarder + logstash + elasticsearch+ kibana
logstash forwarder + logstash +graphite +grafana
graphite = Carbon cache+whisper+graphite web
stagemonitor + graphite+grafana

7.2.2 logstach forwarder

```bash
git clone git://github.com/elasticsearch/logstash-forwarder.git
cd logstash-forwarder
go build -o logstash-forwarder
```

. *centos go language setup in epel

```bash
yum install golang
```

. Packaging it (optional)

```bash
gem install bundler
bundle install
```

. * gem

```bash
yum install ruby
yum install rubygems
```

. * install ruby 1.9.3 http://tecadmin.net/install-ruby-1-9-3-or-multiple-ruby-version-on-centos-6-3-using-rvm/
yum install rpm-build

```bash
make rpm
```

7.2.3 logstach forwarder

test

7.2.4 sta

test

7.3 ELK

7.3.1 ELK on CentOS7

centos 7 https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-4-on-centos-7
centos 6 https://gist.github.com/ashrithr/c5c03950ef631ac63c43
7.3.2 scullxbones/docker_grafana_statsd_elk

https://github.com/scullxbones/docker_grafana_statsd_elk