Installing Steelmesh on RHEL / CentOS 6.x

This guide will walk you through the process of setting up Steelmesh on a RHEL based server (6.x). This guide is based on the guide available at:

http://www.thegeekstuff.com/2012/06/install-couch-db/

NOTE: This installation guide assumes you are completing steps as the root user on the machine. If this is not the case, then you may need to use `sudo` to have particular commands complete successfully.

Core Dependencies

Install standard packages required for compilation of Erlang (and CouchDB):

```
yum install make gcc gcc-c++ libtool libicu-devel openssl-devel xulrunner-devel pcre-devel
```

CouchDB Dependencies and Installation

Erlang

CouchDB is written using Erlang and thus Erlang is required to compile CouchDB from source.

Download the latest stable version of Erlang R15:

```
cd /usr/src
wget http://www.erlang.org/download/otp_src_R15B03-1.tar.gz
tar xzf otp_src_R15B03-1.tar.gz
cd /usr/src/otp_src_R15B03
```

Mark Erlang Libraries that are not required:

```
touch lib/odbc/SKIP lib/wx/SKIP
```

Configure:

```
./configure --prefix=/opt/couchdb/erlang --without-termcap --without-javac --enable-smp-supp
```

Make:

```
make
make install
```

cURL

The default version of cURL that is available in the OS packages is not sufficient for compiling or running CouchDB. As such a more up-to-date version needs to be downloaded and installed from source.

Download the latest stable version of cURL:

```
cd /usr/src
wget http://curl.haxx.se/download/curl-7.30.0.tar.gz
tar xzf curl-7.30.0.tar.gz
cd curl-7.30.0
```

Configure:

```
./configure --prefix=/opt/couchdb/curl
```

Make and install:

```
make
make install
```

SpiderMonkey

Download the 1.85 version of Mozilla's JS interpreter, SpiderMonkey:

```
cd /usr/src
wget http://ftp.mozilla.org/pub/mozilla.org/js/js185-1.0.0.tar.gz
tar xfz js185-1.0.0.tar.gz
cd js-1.8.5/js/src
```

Configure:

```
./configure
```

Make and Install:

```
make
make install
```

CouchDB

Download the CouchDB 1.3.0 source from the Apache releases:

```
cd /usr/src
wget http://apache.mirror.uber.com.au/couchdb/source/1.3.0/apache-couchdb-1.3.0.tar.gz
tar xfz apache-couchdb-1.3.0.tar.gz
cd apache-couchdb-1.3.0
```

Prepare ENV variables:

```
export ERL=/opt/couchdb/erlang/bin/erl
export ERLC=/opt/couchdb/erlang/bin/erlc
export CURL_CONFIG=/opt/couchdb/curl/bin/curl-config
export LDFLAGS=-L/opt/couchdb/curl/lib
```

Configure:

```
./configure --prefix=/opt/couchdb/couchdb --with-erlang=/opt/couchdb/erlang/lib/erlang/usr/i
```

Make and Install:

```
make
make install
```

CouchDB User and Permissions

Next we will need to create the couchdb user:

```
adduser couchdb
```

Then, update the ownership of the couchdb var directory:

```
chown -R couchdb /opt/couchdb/couchdb/var/
```

CouchDB Local Configuration

By default, the CouchDB server only listens on the localhost address of
127.0.0.1. This can be overriden in the `local.ini` file which can be found in
the following location:

```
/opt/couchdb/couchdb/etc/couchdb/local.ini
```

Simply change the section marked `httpd` to match the following:

```
[httpd]
;port = 5984
bind_address = 0.0.0.0
```

CouchDB Service Registration

The CouchDB installation comes prepackaged with an init script for CouchDB, we simply need to create a symbolic link for that file:

```
ln -s /opt/couchdb/couchdb/etc/rc.d/couchdb /etc/init.d/couchdb
```

Test CouchDB

Once completed, we should be able to start the couchdb service:

```
service couchdb start
```

To validate that it is working correctly, curl the server:

```
curl http://localhost:5984/
```

If working, this should yield the following JSON output:

```
{"couchdb":"Welcome","uuid":"b4dfec2b1ebfc53b7d7df92b089d09c8","version":"1.3.0","vendor":{"
```

Node.js

Steelmesh 1.0 is designed to work with Node.js 0.6.x stable releases, so we will install the latest stable release from the 0.6.x release tree:

```
cd /usr/src/
wget http://nodejs.org/dist/v0.6.21/node-v0.6.21.tar.gz
tar xzf node-v0.6.21.tar.gz
cd node-v0.6.21
```

Configure:

```
./configure --prefix=/opt/node
```

Make and Install:

```
make
make install
ln -s /opt/node/bin/node /usr/bin/node
ln -s /opt/node/bin/npm /usr/bin/npm
```

Redis

Redis is a lightweight cache and pubsub message bus. Follow the following instructions to install from source:

```
cd /usr/src
wget http://redis.googlecode.com/files/redis-2.6.13.tar.gz
tar xzf redis-2.6.13.tar.gz
cd redis-2.6.13
```

Make Redis:

```
PREFIX=/opt/redis make install
```

Install default `redis.conf` file:

```
curl https://raw.github.com/DamonOehlman/steelmesh/master/config/redis/redis.conf > /opt/red
```

Configure Redis Service

A default redis service configuration has been provided in the steelmesh source repository, and can be installed using the following commands:

```
curl https://raw.github.com/DamonOehlman/steelmesh/master/config/rhel/init.d/redis > /etc/in
chmod u+x /etc/init.d/redis
```

Install Steelmesh

Installing Steelmesh requires no compilation as it is a Node.js application. Download the latest version of Steelmesh to the `/opt` folder:

```
cd /opt
curl https://codeload.github.com/DamonOehlman/steelmesh/tar.gz/v0.9.8 | tar xz
ln -s steelmesh-0.9.8 steelmesh
```

Configure Steelmesh

We now need to create a user that will own the steelmesh process:

```
adduser steelmesh -d /opt/steelmesh
```

Change ownership of the steelmesh folder to the `steelmesh` user:

```
chown -R steelmesh:steelmesh /opt/steelmesh-0.9.8
chown -R steelmesh:steelmesh /opt/steelmesh
```

We now need to install dependencies for the steelmesh application. This is best done as the `steelmesh` user so let's log in as that user now:

```
su -l steelmesh
```

As the home directory for the user was set to `/opt/steelmesh` we should be in the correct directory to install dependencies:

```
npm install
```

Finally, create the required directories for steelmesh to run correctly:

```
mkdir /opt/steelmesh/logs
```

Exit from the steelmesh user's shell process, back to the root user prompt:

```
exit
```

Test Steelmesh and Install Steelmesh Service

We are now ready to install the `/etc/init.d` script for the steelmesh service. A template configuration file can be download from the steelmesh source repository:

```
curl https://raw.github.com/DamonOehlman/steelmesh/master/config/rhel/init.d/steelmesh > /et
chmod u+x /etc/init.d/steelmesh
```

Create Service Start Run Level Links

To have the appropriate services start on system start, symbolic links need to be added to the appropriate runlevel directories. In this case, we will be adding them to the `/etc/rc3.d` but you can configure them as you see fit (I'm assuming you might know more than I about system administration):

```
ln -s /etc/init.d/couchdb /etc/rc3.d/S70couchdb
ln -s /etc/init.d/redis /etc/rc3.d/S71redis
ln -s /etc/init.d/steelmesh /etc/rc3.d/S75steelmesh
```

Nginx (Optional, but recommended)

Nginx is a lightweight, robust web server that is used in the Steelmesh stack to handle all incoming requests. Requests are then passed onto the underlying node server.

To install nginx from source, do the following:

```
cd /usr/src
wget http://nginx.org/download/nginx-1.4.1.tar.gz
tar xzf nginx-1.4.1.tar.gz
cd nginx-1.4.1
```

Configure:

```
./configure --prefix=/opt/nginx
```

Make and Install:

```
make
make install
```

Configure Nginx

To ensure nginx is started on machine start, you will need to create an `nginx` entry in `/etc/init.d`. If you have used the paths as described in this installation guide you can use the one stored in the steelmesh repository:

```
wget -o /etc/init.d/nginx https://raw.github.com/DamonOehlman/steelmesh/master/config/rhel/
chmod u+x /etc/init.d/nginx
```

Now, download the boilerplate nginx configuration file onto the machine:

```
wget -o /opt/nginx/conf/nginx.conf https://raw.github.com/DamonOehlman/steelmesh/master/conf
```

To validate the configuration has been downloaded succesfully, you can use the `configtest` service option:

```
service nginx configtest
```

If it's ok, then try running the service:

```
service nginx start
```

If this has worked ok, then you should be able to retrieve a document from steelmesh via nginx:

```
TODO
```

iptables Configuration

If your CentOS / RHEL installation has iptables installed and activated by default, you will need to make some modifications to the firewall rules to allow traffic to access the local services on the box. If you do not have iptables installed (i.e. you are using dedicated firewalls on your network) then you can skip this step.

NOTE: The instructions below use the iptables `--insert` command (`-I`) as it is assuming that a default iptables configuration is already in place. It may be more appropriate to review your existing configuration `iptables -L -n` and then craft the rules using the `--append` (`-A`) command...

Firstly, we need to allow CouchDB to accept connections on port 5984:

```
iptables -I INPUT 5 -p tcp -m tcp --dport 5984 -j ACCEPT
```

Next we should let traffic through to the Steelmesh ports (6633 primary, 3274 dashboard):

```
iptables -I INPUT 6 -p tcp -m tcp --dport 6633 -j ACCEPT
iptables -I INPUT 7 -p tcp -m tcp --dport 3274 -j ACCEPT
```

Finally, if you installed nginx in front of the steelmesh node server, you will also want to allow traffic through on port 80:

```
iptables -I INPUT 8 -p tcp -m tcp --dport 80 -j ACCEPT
```

Review your firewall configuration:

```
iptables -L -n
```

If everything looks to be in order, then save the configuration and restart the
service:

```
iptables-save > /etc/sysconfig/iptables
service iptables restart
```