Locker Setup Documentation V1.0

Chris Elsmore

October 2011

This document explains the steps to install and configure Locker and its dependencies, and to create a secure web proxy front end using Nginx, on Ubuntu Linux circa 2011 (tested on 11.04 Natty Narwhal).

1 Locker Configuration

1.1 Locker Dependencies

Locker uses a number of technologies, all of which need to be installed and configured before starting the locker and pulling in personal data. These include: MongoDB, Node.js, NPM and Clucene as well as the Locker code itself. Before starting, the Distribution should be up to date

1.2 MongoDB

Installing Mongo DB involves adding the Mongo Repositories, and then installing with apt-get.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10

sudo gedit /etc/apt/sources.list
    deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install mongodb-10gen
```

1.3 Node.js

Installing Node.js involves first downloading and building the source. The latest Node.js that Locker currently supports is V0.4.9, and a few packages are needed before building can complete.

```
sudo apt-get install g++ curl libssl-dev apache2-utils git-core cmake

cd ~/Downloads
mkdir nodejs && cd nodejs

wget http://nodejs.org/dist/node-v0.4.9.tar.gz
tar -xvf node-v0.4.9.tar.gz

./configure
make -j2
sudo make install
```

1.4 NPM

Installing NPM is done by running an install script that fetches and installs NPM.

```
cd ~/Downloads
mkdir npm && cd npm

curl -O http://npmjs.org/install.sh
chmod +x install.sh
sudo ./install.sh
```

1.5 Clucene

Clucene is currently built from source from the git tree, and then built and installed.

```
cd ../
git clone git://clucene.git.sourceforge.net/gitroot/clucene/clucene .
cd Clucene
cmake -G "Unix Makefiles"
make
sudo make install
sudo ldconfig -vv | grep clucene
(clucene.sourceforge.net/download.shtml)

( hit http://localhost:8042/Me/search/update to start full re-index )
```

1.6 Locker

The Locker code is installed by cloning a git repo, and then installing all needed JavaScript and Python packages using NPM and Easy Install respectively.

```
cd ~/
mkdir Locker
cd Locker
git clone https://github.com/LockerProject/Locker.git .
sudo npm install -g
    (Cluscene fails)
sudo npm install ini

sudo apt-get install python-setuptools
sudo easy_install virtualenv gdata flask pyparsing google-api-python-client

./checkEnv.sh
```

2 Nginx Configuration

```
sudo apt-get update
sudo apt-get install apache2-utils git-core curl build-essential
sudo apt-get openssl libssl-dev nginx
cd Downloads/
git clone https://github.com/joyent/node.git && cd node
./configure
make
sudo make install
node -v

mkdir ~/Documents/multiserver-test/
mkdir ~/Documents/multiserver-test/pass
mkdir ~/Documents/multiserver-test/logs
cd ~/Documents/multiserver-test/

sudo nano /etc/nginx/proxy.conf
proxy_redirect                  off;
proxy_set_header                Host                    $host;
proxy_set_header                X-Real-IP               $remote_addr;
proxy_set_header                X-Forwarded-For         $proxy_add_x_forwarded_for;
client_max_body_size            10m;
client_body_buffer_size         128k;
proxy_connect_timeout           90;
proxy_send_timeout              90;
proxy_read_timeout              90;
proxy_buffers                   32 4k;
 => /etc/nginx/proxy.conf

sudo nano /etc/nginx/sites-enabled/nodetest
server {
            listen      80;
            server_name domain1.localhost;

            access_log  /home/elsmorian/Documents/multiserver-test/logs/domain1.access.log;
            error_log   /home/elsmorian/Documents/multiserver-test/logs/domain1.error.log;

            location / {
                    proxy_pass http://127.0.0.1:8000;
                    #include /etc/nginx/proxy.conf;
            }
}

server {
        listen  80;
        server_name domain2.localhost;

        access_log      /home/elsmorian/Documents/multiserver-test/logs/domain2.access.log;
        error_log       /home/elsmorian/Documents/multiserver-test/logs/domain2.error.log;

        location / {
                proxy_pass http://127.0.0.1:8001;
        }
}

server {
        listen  80;
        server_name domain3.localhost;

        access_log      /home/elsmorian/Documents/multiserver-test/logs/domain3.access.log;
        error_log       /home/elsmorian/Documents/multiserver-test/logs/domain3.error.log;

        location / {
                proxy_pass http://127.0.0.1:8002;
        }
}
 => /etc/nginx/sites-enabled/nodetest

sudo invoke-rc.d nginx start

nano domain1.js
var http = require('http');
http.createServer(function(req, res) {
        res.writeHead(200, {'Content-Type': 'text/plain'});
        res.end('Hello, I\'m Domain 1, on port 8000.\n');
}).listen(8000, '127.0.0.1');
console.log('Server running locally, port: 8000');
 => domain1.js
node domain1.js

Test!

sudo nano /etc/nginx/sites-enabled/nodetest
server {
        listen  80;
        server_name domain1.localhost;

        access_log      /home/elsmorian/Documents/multiserver-test/logs/domain1.access.log;
        error_log       /home/elsmorian/Documents/multiserver-test/logs/domain1.error.log;

        location / {
                proxy_pass              http://127.0.0.1:8000;
                #include                /etc/nginx/proxy.conf;
                auth_basichtpasswd -c -d domain2.htpasswd user2         "Restricted";
                auth_basic_user_file    /home/elsmorian/Documents/multiserver-test/pass/domain1.htpas
        }
}

server {
        listen  80;
        server_name domain2.localhost;

        access_log      /home/elsmorian/Documents/multiserver-test/logs/domain2.access.log;
        error_log       /home/elsmorian/Documents/multiserver-test/logs/domain2.error.log;

        location / {
                proxy_pass http://127.0.0.1:8001;
                auth_basic              "Restricted";
                auth_basic_user_file    /home/elsmorian/Documents/multiserver-test/pass/domain2.htpas
        }
}

server {
        listen  80;
        server_name domain3.localhost;

        access_log      /home/elsmorian/Documents/multiserver-test/logs/domain3.access.log;
        error_log       /home/elsmorian/Documents/multiserver-test/logs/domain3.error.log;

        location / {
                proxy_pass http://127.0.0.1:8002;
                auth_basic              "Restricted";
                auth_basic_user_file    /home/elsmorian/Documents/multiserver-test/pass/domain3.htpas
        }
}
 => sudo /etc/nginx/sites-enabled/nodetest
```

```
cd pass
htpasswd -c -d domain1.htpasswd user1
htpasswd -c -d domain2.htpasswd user2
htpasswd -c -d domain3.htpasswd user3

sudo nano /etc/hosts
127.0.0.1       localhost
127.0.0.1       domain1.localhost
127.0.0.1       domain2.localhost
127.0.0.1       domain3.localhost
127.0.0.1       locker.localhost


add for Locker:
server {
        listen  80;
        server_name locker.localhost;

        access_log      /home/elsmorian/Documents/multiserver-test/logs/locker.access.log;
        error_log       /home/elsmorian/Documents/multiserver-test/logs/locker.error.log;

        location / {
                proxy_pass http://127.0.0.1:8042;
                auth_basic              "Welcome to your Locker. Please enter your username and passw
                auth_basic_user_file    /home/elsmorian/Documents/multiserver-test/pass/locker.htpass
        }
}

Certs&SSL:

mkdir certs
openssl genrsa -des3 -out server.key 1024
pass: custard
openssl req -new -key server.key -out server.csr
pass- custard
challenge pass: rainbow
mv server.key server.csr certs/
cp certs/server.key certs/server.key.org
cd certs/
openssl rsa -in server.key.org -out server.key
pass- custard
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt


apt-get install python-setuptools (for easy_install)
```