# Setting up Linux box for Elasticsearch, Logstash & Kibana – ELK & ActiveMQ

This guideline is based on a CentOS/7 Linux VM or physical box previously setup.

**Update Yum**  
sudo Yum Update

**Install wget**

sudo yum install wget

**Install Netstat**  
Yum install net-tools

**Download and install Vi alternative Nano**

Sudo yum install nano

**Set IP address to static**

open network manager, sudo nmtui, edit network stack

**Install Oracle java 8 jdk**

Open ssh session via putty. CD to home director. Run wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u112-b15/jdk-8u112-linux-x64.rpm"

Then install the RPM with this yum command (if you downloaded a different release, substitute the filename here):

sudo yum localinstall jdk-8u112-linux-x64.rpm

Now Java should be installed at /usr/java/jdk1.8.0\_112/jre/bin/java, and linked from /usr/bin/java.

You may delete the archive file that you downloaded earlier:yum

rm ~/jdk-8u112-linux-x64.rpm

**Download and Install ELK components**

Download and install the public signing key:

rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch

Add the following in your /etc/yum.repos.d/ directory in a file with a .repo suffix, for example elasticsearch.repo

[elasticsearch-5.x]

name=Elasticsearch repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

And your repository is ready for use. You can install it with:

yum install elasticsearch

Configure Elasticsearch to automatically start during bootup. If your distribution is using SysV init (check with ps -p 1), then you will need to run:

Warning

The repositories do not work with older rpm based distributions that still use RPM v3, like CentOS5.

chkconfig --add elasticsearch

Otherwise if your distribution is using systemd:

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable elasticsearch.service

**Download and install Kibana**

 Create a file named kibana.repo in the /etc/yum.repos.d/ directory with the following contents:

[kibana-5.x]

name=Kibana repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

 Install Kibana by running the following command:

yum install kibana

Configure Kibana to automatically start during bootup. If your distribution is using SysV init (check with ps -p 1), then you will need to run:

Warning

The repositories do not work with older rpm based distributions that still use RPM v3, like CentOS5.

chkconfig --add kibana

Otherwise if your distribution is using systemd:

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable kibana.service

**Install Nginx**

Because we configured Kibana to listen on localhost, we must set up a reverse proxy to allow external access to it. We will use Nginx for this purpose.

**Note:** If you already have an Nginx instance that you want to use, feel free to use that instead. Just make sure to configure Kibana so it is reachable by your Nginx server (you probably want to change the host value, in /opt/kibana/config/kibana.yml, to your Kibana server's private IP address). Also, it is recommended that you enable SSL/TLS.

Add the EPEL repository to yum:

* sudo yum -y install epel-release

Now use yum to install Nginx and httpd-tools:

* sudo yum -y install nginx httpd-tools

Use htpasswd to create an admin user, called "kibanaadmin" (you should use another name), that can access the Kibana web interface:

* sudo htpasswd -c /etc/nginx/htpasswd.users kibanaadmin

Enter a password at the prompt. Remember this login, as you will need it to access the Kibana web interface. User=kibanaadmin, Password=password,

Now open the Nginx configuration file in your favorite editor. We will use nano:

* sudo nano /etc/nginx/nginx.conf

Find the default server block (starts with server {), the last configuration block in the file, and delete it. When you are done, the last two lines in the file should look like this:nan

nginx.conf excerpt

include /etc/nginx/conf.d/\*.conf;

}

Save and exit.

Now we will create an Nginx server block in a new file:

* sudo vi /etc/nginx/conf.d/kibana.conf

Paste the following code block into the file. Be sure to update the server\_name to match your server's name:

/etc/nginx/conf.d/kibana.conf

* server {
* listen 80;
* **server\_name elk**;
* auth\_basic "Restricted Access";
* auth\_basic\_user\_file /etc/nginx/htpasswd.users;
* location / {
* proxy\_pass http://localhost:5601;
* proxy\_http\_version 1.1;
* proxy\_set\_header Upgrade $http\_upgrade;
* proxy\_set\_header Connection 'upgrade';
* proxy\_set\_header Host $host;
* proxy\_cache\_bypass $http\_upgrade;
* }
* }

Save and exit. This configures Nginx to direct your server's HTTP traffic to the Kibana application, which is listening on localhost:5601. Also, Nginx will use the htpasswd.users file, that we created earlier, and require basic authentication.

**Set VM Hostname**

Run network manager tool. Nmtui  
set VM hostname to elk or other host name as desired.

**Install Logstash**

Add the following in your /etc/yum.repos.d/ directory in a file with a .repo suffix, for example logstash.repo

[logstash-5.x]

name=Elastic repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

And your repository is ready for use. You can install it with:

yum install logstash

Configure Logstash to automatically start during bootup. If your distribution is using SysV init (check with ps -p 1), then you will need to run:

Warning

The repositories do not work with older rpm based distributions that still use RPM v3, like CentOS5.

chkconfig --add logstash

Otherwise if your distribution is using systemd:

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable logstash.service

Create Logstash configuration files for “input”, “Filter” & “output”. Place them in /etc/logstash/conf.d

The files should have a .conf extention.

**Champion can supply the baseline set of .conf files**

**Install required Logstash Plugins (not installed by default)**

To install Stomp plugin required for sending JSON messages to ActiveMQ run the following command. /usr/share/Logstash/bin/Logstash-plugin install Logstash-output-stomp

To install TLD filter plugin required for breaking apart URL with the top level domain as a separate field run the following command. /usr/share/Logstash/bin/Logstash-plugin install Logstash-filter-tld.

**Update Logstash Plugins**

Run the following command /usr/share/Logstash/bin/Logstash-plugin update

**Updating Logstash etc. when new versions are released**

To update Logstash, Elasticsearch, Kibana, Nginx etc. run “yum update Logstash” etc.  
If Logstash, Elasticsearch or Kibana are updated with new packages, note that they do not reinstall all of the plugins. These will need to be reinstalled.

**To reinstall after an update run:**   
/usr/share/Logstash/bin/Logstash-plugin install Logstash-output-stomp  
/usr/share/Logstash/bin/Logstash-plugin install Logstash-filter-tld

To update all plugins as they are not updated when Logstash is updated run:

/usr/share/Logstash/bin/Logstash-plugin-update

**Install current GeoIP db from Maxmind and setup cron job to update automatically**Make folder /var/local/geoip  
Run script /etc/Logstash/geo-update.bash to populate folder /var/local/geoip with updated db file

Create Cron Job to automate the update of the GeoIP db file

Run crontab -e  
add: 0 12 \* \* 3 /etc/Logstash/geo-update.bash > /dev/null 2>&1

Save.

**Disable SELinux**

From the command line, you can edit the /etc/sysconfig/selinux file. This file is a symlink to /etc/selinux/config. The configuration file is self-explanatory. Changing the value of *SELINUX* or *SELINUXTYPE* changes the state of SELinux and the name of the policy to be used the next time the system boots. Set “SELINUX=disabled”.

[root@host2a ~]# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - SELinux is fully disabled.

SELINUX=permissive

# SELINUXTYPE= type of policy in use. Possible values are:

# targeted - Only targeted network daemons are protected.

# strict - Full SELinux protection.

SELINUXTYPE=targeted

# SETLOCALDEFS= Check local definition changes

SETLOCALDEFS=0

**Disable Firewalld service**

Sudo systemctl stop firewalld

sudo systemctl disable firewalld

**Install ActiveMQ**

Copy to VM and run script “stomp\_install.sh” It will be required to update the IP address at the end of the script file.

Links to some info

Activemq server setup   
<https://rburnham.wordpress.com/2012/09/14/installing-and-managing-activemq-on-centos/>

Firewalld setup:  
<http://www.liquidweb.com/kb/how-to-start-and-enable-firewalld-on-centos-7/>

Open active MQ ports:  
<http://stackoverflow.com/questions/24729024/centos-7-open-firewall-port>   
<http://www.codero.com/knowledge-base/content/24/377/en/how-to-manage-firewall-rules-in-centos-7.html>

Advanced MQ setup:   
Composite Destinations allow for one-to-many relationships on individual destinations; the main use case is for *composite queues*. For example when a message is sent to queue A you may want to forward it also to queues B and C and topic D.

From <<http://activemq.apache.org/virtual-destinations.html>>

Forcing messages to expire:   
<http://stackoverflow.com/questions/34261937/time-to-live-for-all-messages-in-activemq-queue>   
<http://activemq.apache.org/timestampplugin.html>

<plugins>   
<!-- if not already set, set ttl to 1 hour  
        <timeStampingBrokerPlugin zeroExpirationOverride="3600000"/>-->

<!-- if not already set, set ttl to 5 minutes -->   
        <timeStampingBrokerPlugin zeroExpirationOverride="300000"/>   
</plugins>

Script to auto install ActiveMQ. Just change the IP address at the end of the script. Copy and paste lines between \*\*\*\*\*\*\*\*\*\*\*\* and save as activemq.sh

**“stomp\_install.sh”**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#!/bin/sh  
# Download ActiveMQ and place in /opt  
cd /tmp  
wget <http://archive.apache.org/dist/activemq/5.14.0/apache-activemq-5.14.0-bin.tar.gz>  
tar xzvf apache-activemq-5.14.0-bin.tar.gz  
mv apache-activemq-5.14.0 /opt  
ln -sf /opt/apache-activemq-5.14.0/ /opt/activemq

# Create ActiveMQ user  
adduser --system activemq  
chown -R activemq: /opt/apache-activemq-5.14.0/

# Set the managementContext element to true in opt/activemq/conf/activemq.xml  
echo "Changing /opt/activemq/conf/activemq.xml managementContext to true..."  
cp /opt/activemq/conf/activemq.xml /opt/activemq/conf/activemq.xml.backup  
sed -i 's/createConnector="false"/createConnector="true"/' /opt/activemq/conf/activemq.xml

# Install service scripts

echo Creating services...  
echo " creating activemq start script..."  
touch /etc/init.d/activemqstart.sh  
echo -e "#!/bin/bash \nexport JAVA\_HOME=/usr \n/opt/activemq/bin/activemq start" >> /etc/init.d/activemqstart.sh

echo " creating activemq stop script..."  
touch /etc/init.d/activemqstop.sh  
echo -e "#!/bin/bash \nexport JAVA\_HOME=/usr \n/opt/activemq/bin/activemq stop" >> /etc/init.d/activemqstop.sh

echo " creating activemq"  
touch /etc/init.d/activemq  
echo -e "#!/bin/bash\n#\n# activemq Starts ActiveMQ.\n#\n# chkconfig: 345 88 12\n# description: ActiveMQ is a JMS Messaging Queue Server.\n### BEGIN INIT INFO\n# Provides: \$activemq\n### END INIT INFO\n" >> /etc/init.d/activemq

echo -e "# Source function library" >> /etc/init.d/activemq  
echo -e ". /etc/init.d/functions\n" >> /etc/init.d/activemq

echo -e "[ -f /etc/init.d/activemqstart.sh ] || exit 0" >> /etc/init.d/activemq

echo -e "[ -f /etc/init.d/activemqstop.sh ] || exit 0" >> /etc/init.d/activemq  
echo -e "\nRETVAL=0" >> /etc/init.d/activemq  
echo -e "\numask 077" >> /etc/init.d/activemq  
echo -e "\nstart() {" >> /etc/init.d/activemq  
echo -e " echo -n $\"Starting ActiveMQ: \"" >> /etc/init.d/activemq  
echo -e " /etc/init.d/activemqstart.sh" >> /etc/init.d/activemq  
echo -e " echo" >> /etc/init.d/activemq  
echo -e " return \$RETVAL" >> /etc/init.d/activemq  
echo -e "}" >> /etc/init.d/activemq  
echo -e "stop() {" >> /etc/init.d/activemq  
echo -e " echo -n $\"Shutting down ActiveMQ: \"" >> /etc/init.d/activemq  
echo -e " /etc/init.d/activemqstop.sh" >> /etc/init.d/activemq  
echo -e " echo" >> /etc/init.d/activemq  
echo -e " return \$RETVAL" >> /etc/init.d/activemq  
echo -e "}" >> /etc/init.d/activemq  
echo -e "restart() {" >> /etc/init.d/activemq  
echo -e " stop && start" >> /etc/init.d/activemq  
echo -e "}" >> /etc/init.d/activemq  
echo -e "case \"\$1\" in" >> /etc/init.d/activemq  
echo -e "start)" >> /etc/init.d/activemq  
echo -e " start" >> /etc/init.d/activemq  
echo -e " ;;" >> /etc/init.d/activemq  
echo -e "stop)" >> /etc/init.d/activemq  
echo -e " stop" >> /etc/init.d/activemq  
echo -e " ;;" >> /etc/init.d/activemq  
echo -e "restart|reload)" >> /etc/init.d/activemq  
echo -e " restart" >> /etc/init.d/activemq  
echo -e " ;;" >> /etc/init.d/activemq  
echo -e "\*)" >> /etc/init.d/activemq  
echo -e " echo $\"Usage: \$0 {start|stop|restart}\"" >> /etc/init.d/activemq

echo -e " exit 1" >> /etc/init.d/activemq  
echo -e "esac" >> /etc/init.d/activemq  
echo -e "\nexit \$?" >> /etc/init.d/activemq  
# Note: EOF

# Permissions

echo "Setting permissions..."  
chmod +x /etc/init.d/activemq  
chmod +x /etc/init.d/activemqstart.sh  
chmod +x /etc/init.d/activemqstop.sh  
chkconfig --add activemq  
chkconfig activemq on

# Start ActiveMQ

echo "Starting ActiveMQ service..."  
service activemq start

# Status check

echo "Status check on the service..."  
/opt/activemq/bin/activemq list

# Port check

echo "Check port 61616..."  
netstat -an | grep 61616

# Set TTL to 5 minutes

if grep -Fq "ttl to 5 minutes" /opt/activemq/conf/active.xml  
then  
 echo "TTL already set to 5 minutes. Skipping..."

else

echo "Setting the TTL to 5 minutes..."

sed -i '/<broker xmlns="http:\/\/activemq.apache.org\/schema\/core" brokerName="localhost" dataDirectory="${activemq.data}">/a \ \t<plugins>\n\t\t<!-- If not already set, set ttl to 5 minutes -->\n\t\t<timeStampingBrokerPlugin zeroExpirationOverride="300000"\/>\n\t<\/plugins>' /opt/activemq/conf/activemq.xml

fi

# Add the virtual destination, Champ.Ingest

if grep -Fq "Champ.Ingest" /opt/activemq/conf/activemq.xml

then

echo "Virtual destination: Champ.Ingest already present. Skipping..."

else

echo "adding the virtual destination: Champ.Ingest in /opt/activemq/conf/activemq.xml"

sed -i '/<\/destinationPolicy>/a \ \n\t<!--\n\t Create a virtual destination called Champ.Ingest\n\t that directs all messages to a queue and a topic both called "Champ.Ingest.Item"\n\t-->\n\t <destinationInterceptors>\n\t\t<virtualDestinationInterceptor>\n\t\t\t<virtualDestinations>\n\t\t\t\t<compositeQueue name="Champ.Ingest">\n\t\t\t\t\t<forwardTo>\n\t\t\t\t\t\t<queue physicalName="Champ.Ingest.Item.Queue" \/>\n\t\t\t\t\t\t<topic physicalName="Champ.Ingest.Item" \/>\n\t\t\t\t\t<\/forwardTo>\n\t\t\t\t<\/compositeQueue>\n\n\t\t\t\t<compositeQueue name="Champ.Ingest.4624">\n\t\t\t\t\t<forwardTo>\n\t\t\t\t\t\t<queue physicalName="Champ.Ingest.Item.4624" \/>\n\t\t\t\t\t\t<topic physicalName="Champ.Ingest.Item.4624" \/>\n\t\t\t\t\t<\/forwardTo>\n\t\t\t\t<\/compositeQueue>\n\t\t\t<\/virtualDestinations>\n\t\t<\/virtualDestinationInterceptor>\n\t <\/destinationInterceptors>' /opt/activemq/conf/activemq.xml

fi

echo "If all looks good, check the web GUI at http;//192.168.5.242:8161/admin/"

echo -e "Default credentials:\n User: admin\n Pass: admin"

echo Done!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*