Appendix A - Necessary external software

Contents

This page describes the various external software dependencies in NetarchiveSuite.

The NetarchiveSuite is developed and tested with Sun Java SE (Standard Edition) JDK version 1.8.0_20. In any case a Java 1.8+ JDK will be necessary to compile and run the NetarchiveSuite, and we recommend that all applications use the same JDK. A Java 1.8+ runtime environment is necessary for all applications.

The following external software is required for running the applications

- JMS
- FTP This is only required, if FTPRemoteFile is the chosen RemoteFile Plugin.
- SSH (Installed as default under Unix/Linux, and WinSSHD by bitvise.com does the trick on Windows).
- Unzip. "unzip.exe" on Windows, and "unzip" on Linux.

Windows specific

Some application requires the Unix command `sort`, but they should be able to run under Windows if Cygwin is installed. This should only affect the ViewerProxy, the IndexServer, and the WyabackIndexer.

Installing and configuring a JMS broker

The software has been tested with the free JMS broker from Sun "Open Message Queue".

Obtaining a JMS broker

Sun's Open Message Queue can be obtained from the following site: https://mq.java.net/downloads/index.html

Go to the section named "Legacy Versions", and click on the Linux link in the subsection "Open MQ 5.1 Binary Downloads". This will give you a jar-file named "mq5_1-binary-Linux_X86-XXXXXXXX.jar". (We have no reason to suppose that NetarchiveSuite will have problems with newer versions but these are still untested with our software.)

Note: We only support installation on the Linux platform here. However, you may want to install your JMS broker on a different platform. Binary versions are available at the site for: Solaris Sparc, Solaris x86, Linux (x86), Windows (x86). If you want to build a binary for another platform, the source can be downloaded from the download-page.

Installing the JMS broker

Select the Linux server where you want to install JMS broker, and select an installation directory. Then log on the linux server as root, and do the following:

```
export INSTALLATION_DIR=/path/to/installationdir
cd $INSTALLATION_DIR
unzip mq5_1-binary-Linux_X86-XXXXXXXX.jar
chmod +x ./mq/bin/imqbrokerd
./mq/bin/imqbrokerd -reset store -tty (tests that the broker can start - CTLR-C to stop)
```

Check that it starts, and that the last message is

```
Broker <localhost>:7676 ready
```

We are now ready to configure the JMS broker.

Configuring the JMS broker

- Edit the file `$INSTALLATION_DIR/mq/etc/imqenv.conf` to set IMQ_DEFAULT_JAVAHOME to a JDK.
- Changing the number of the listening port number 7676 is done by editing the line
  . `imq.portmapper.port=7676`
  . in the file
  . `$INSTALLATION_DIR/mq/lib/props/broker/default.properties`
- Set max listeners any given queue to 20. You need to make sure, that the following line
  . `imq.autocreate.queue.maxNumActiveConsumers=20`
  . is present and not commented out in the file
  . `$INSTALLATION_DIR/mq/var/instances/imqbroker/props/config.properties`
  . (increase the number 20 if you have more than that number of applications of the same kind on the same bitarchive replica, for instance more than 20 bitarchiveapplications)
- Set max producers to 100. You add the following line
  . `imq.autocreate.destination.maxNumProducers=100`
  . in the file
  . {{$INSTALLATION_DIR/mq/var/instances/imqbroker/props/config.properties }}
  . If you get an error like this:
  . {{Producer can not be added to destination PROD_COMMON_MONITOR Queue, limit of 100 producers would be exceeded }}
  . in the JMS broker log, you need to increase this value.

Starting and stopping JMS

The broker is started directly in this way:

```
$INSTALLATION_DIR/mq/bin/imqbrokerd -reset store -tty &
```

The sysadmin would maybe like to start the broker on machine startup by inserting the statement above into the /etc/rc.d/rc.local

The broker is stopped in this way:

```
logon on machine as root
find processid for the broker (ps auxw | grep imqbrokerd)
kill -9 $IMQ_PROCESSID
```

Alternatively press Crtl-c, if the terminal where the broker was started, is still available

You can test that JMS broker is alive by telnetting to its port, where it will give some technical information in reply:

```
[user@udvikling kb-dev-adm-001.kb.dk]$ telnet localhost 7676
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
101 imqbroker 4.1
portmapper tcp PORTMAPPER 7676 [sessionid=1729683678303517696]
cluster_discovery tcp CLUSTER_DISCOVERY 46760
jmxrmi rmi JMX 0 [url=service:jmx:rmi://udvikling.kb.dk/stub/rO0...Hg=]
admin tcp ADMIN 46763
jms tcp NORMAL 46762
cluster tcp CLUSTER 46764
.
Connection closed by foreign host.
```

To run JMS client applications, include the following jar files in the classpath :

```
$INSTALLATION_DIR/mq/lib/jms.jar  $INSTALLATION_DIR/mq/lib/imq.jar
```

Create a passfile named '.imq_passfile' (used when emptying JMS queues):

```
imq.imqcmd.password=REPLACE_WITH_PASSWORD
```

How to empty queues

log on as root to the server, where the JMS broker is installed. The following assumes that the JMS environmentName is PROD, and that JMS password file resides in ~root/.imq_passfile:

```
export JMS_ENV=PROD
export MQ_HOME=/usr/local
# imqcmd using -u admin -passfile ~/.imq_passfile
$MQ_HOME/bin/imqcmd list dst -t q -u admin -passfile ~/.imq_passfile | grep ^${JMS_ENV}_ | cut -f1 -d\
|xargs -r -n 1 $MQ_HOME/bin/imqcmd destroy dst -t q -u admin -passfile ~/.imq_passfile -f -n
$MQ_HOME/bin/imqcmd list dst -t t -u admin -passfile ~/.imq_passfile | grep ^${JMS_ENV}_ | cut -f1 -d\
|xargs -r -n 1 $MQ_HOME/bin/imqcmd destroy dst -t t -u admin -passfile
~/.imq_passfile -f -n"
```

How to allocate additional JMS broker memory

```
export MQ_HOME=/usr/local
$MQ_HOME/mq/bin/imqbrokerd -vmargs "-Xms256m -Xmx512m" -reset store -tty &
#which adds min 256Mb and max 512MB heap space
```

Installing and configuring FTP

If you decide to use FTPRemote for file transfer in NetarchiveSuite, you need to install and start one or more FTP servers, before you begin the installation of the NetarchiveSuite. Any brand of FTP-servers will probably do, but we have good experience with Proftpd.

You can download Proftpd from http://www.proftpd.org/. We are using versio n 1.3.4d, but any recent non-beta version will probably do.

The text below shows part of the proftpd.conf needed by NetarchiveSuite. Other parameters in proftpd.conf may be left with their default values.

```
# Port 21 is the standard FTP port.
Port                            21
# Umask 022 is a good standard umask to prevent new dirs and files
# from being group and world writable.
Umask                           022
# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd).
MaxInstances                    250
# Set the user and group under which the server will run.
User                            nobody
#Group                          nogroup
Group                           nobody
# To cause every FTP user to be "jailed" (chrooted) into their home
# directory, uncomment this line.
#DefaultRoot ~
# Normally, we want files to be overwriteable.
## This is necessary to allow the append operation
AllowOverwrite          on
AllowStoreRestart on
# Bar use of SITE CHMOD by default
<Limit SITE_CHMOD>
  DenyAll
</Limit>
# This enables or disables the PAM authentication module.
# The default is 'on'.
#AuthPAM off
```

If you want to have the FTP-server use a specific directory for uploading files, e.g. ~/ftp, you can use add the configuration

```
DefaultChdir ~/ftp
```

If the /ftp does not exist, the server will fallback to the "".

Starting and stopping a Proftpd server

Log as root on to the server, where Proftpd is installed, and the following command will start the FTP-server

```
/usr/local/sbin/proftpd
```

The following will kill the FTP-server.

```
killall -9 proftpd
```