

Install Alfresco on FreeBSD

11/13/2019 10:21 PM - Daniel Curtis

Status:	Suspended	Start date:	11/13/2019
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	30%
Category:	File Server	Estimated time:	0.00 hour
Target version:	FreeBSD 12	Spent time:	0.00 hour

Description

This is a guide for setting up Alfresco on FreeBSD.

Prepare the Environment

- Make sure the system is up to date:

```
pkg update && pkg upgrade
```

- Add the Alfresco user

```
pw add user -n alfresco -g www -d /nonexistent -s /usr/sbin/nologin -w no -c "Alfresco"
```

- Install dependencies:

```
pkg install tomcat7 ImageMagick7-nox11 libreoffice liberation-fonts-ttf droid-fonts-ttf swftool postgresql96-server postgresql96-contrib postgresql-jdbc-9.2.1004
```

Install PostgreSQL

- Initialize, start, and enable postgresql at boot:

```
sysrc postgresql_enable="YES"
service postgresql initdb
service postgresql start
```

- Create database admin:

```
su pgsq1 -c "createuser -sdrP postgres"
```

- Create the alfresco database user:
NOTE: Make sure to update the alfresco-global.properties with the username and password.

```
su pgsq1 -c "createuser -D -A -P alfresco"
```

- Create alfresco database:

```
su pgsq1 -c "createdb -O alfresco alfrescodb"
```

- Test the database connection:

```
psql -U alfresco -d alfrescodb
```

Setup Alfresco Prerequisites

- Create the Alfresco directories:

```
mkdir -p /usr/local/www/alfresco/{alf_data,tomcat,solr}
mkdir -p /usr/local/www/alfresco/alf_data/{keystore,solr}
mkdir -p /usr/local/www/alfresco/addons/{war,shared,alfresco}
mkdir -p /usr/local/www/alfresco/tomcat/conf/Catalina/localhost
mkdir -p /usr/local/www/alfresco/tomcat/shared/lib
mkdir -p /usr/local/www/alfresco/tomcat/shared/classes/alfresco{extension,web-extension}
mkdir -p /usr/local/www/alfresco/tomcat/endorsed
```

- Symlink tomcat into alfresco directory:

```
ln -s /usr/local/apache-tomcat-7.0/* /usr/local/www/alfresco/tomcat/
```

- Remove all default tomcat webapps:

```
rm -rf /usr/local/www/alfresco/tomcat/webapps/*
```

tomcat-users.xml

- Edit the **tomcat-users.xml** file:

```
vi /usr/local/www/alfresco/tomcat/conf/tomcat-users.xml
```

- And modify it accordingly:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
<!-- Repository -->
  <user username="CN=Alfresco Repository Client, OU=Doc Repo, O=GNet Solutions, L=Lynnwood
, ST=WA, C=US" roles="repoclient" password="null"/>

<!-- Solr -->
  <user username="CN=Alfresco Repository, OU=Doc Repo, O=GNet Solutions, L=Lynnwood, ST=WA
, C=US" roles="repository" password="null"/>
</tomcat-users>
```

server.xml

- Edit the tomcat **server.xml** config:

```
vi /usr/local/www/alfresco/tomcat/conf/server.xml
```

- And modify it accordingly:

```
<?xml version='1.0' encoding='utf-8'?>
```

```

<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JasperListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />

  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>

  <Service name="Catalina">
    <Connector port="8080"
              protocol="HTTP/1.1"
              URIEncoding="UTF-8"
              connectionTimeout="20000"
              redirectPort="8443"
              maxHttpHeaderSize="32768"
              compression="off"
              compressableMimeType="text/html,text/xml,text/plain,application/json,text/javascript,application/x-javascript,text/css,text/csv,text/x-web-markdown,application/atom+xml,application/rss+xml,application/atomsvc+xml"
              noCompressionUserAgents=".*MSIE 6.*"
              compressionMinSize="256"
              maxThreads="200"
              acceptCount="100"
              />

    <Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol" SSLEnabled="true" URIEncoding="UTF-8"
              maxThreads="150" scheme="https" keystoreFile="/opt/alfresco/alf_data/keystore/ssl.keystore" keystorePass="kT9X6oe68t" keystoreType="JCEKS"
              secure="true" connectionTimeout="240000" truststoreFile="/opt/alfresco/alf_data/keystore/ssl.truststore" truststorePass="kT9X6oe68t" truststoreType="JCEKS"
              clientAuth="want" allowUnsafeLegacyRenegotiation="true" maxHttpHeaderSize="32768" sslProtocol="TLS" maxSavePostSize="-1" />

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8" />

    <!-- An Engine represents the entry point (within Catalina) that processes
         every request.  The Engine implementation for Tomcat stand alone
         analyzes the HTTP headers included with the request, and passes them
         on to the appropriate Host (virtual host).
         Documentation at /docs/config/engine.html -->

    <!-- You should set jvmRoute to support load-balancing via AJP ie :
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
    -->
    <Engine name="Catalina" defaultHost="localhost">

    <!--For clustering, please take a look at documentation at:
         /docs/cluster-howto.html (simple how to)
         /docs/config/cluster.html (reference documentation) -->
    <!--
    <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
    -->

    <!-- Use the LockOutRealm to prevent attempts to guess user passwords
         via a brute-force attack -->
    <Realm className="org.apache.catalina.realm.LockOutRealm">

```

```

        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase". Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm. -->
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
              resourceName="UserDatabase"/>
    </Realm>

    <Host name="localhost"  appBase="webapps"
          unpackWARs="true" autoDeploy="true">
        </Host>
    </Engine>
</Service>
</Server>

```

catalina.properties

- Edit the **catalina.properties** file:

```
vi /usr/local/www/alfresco/tomcat/conf/catalina.properties
```

- And modify it accordingly:

```
package.access=sun.,org.apache.catalina.,org.apache.coyote.,org.apache.jasper.,\
org.apache.naming.resources.,org.apache.tomcat.
```

```
package.definition=sun.,java.,org.apache.catalina.,org.apache.coyote.,\
org.apache.jasper.,org.apache.naming.,org.apache.tomcat.
```

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/lib,${catalina.home}/lib/*.jar
```

```
server.loader=
```

```
shared.loader=${catalina.base}/shared/classes,${catalina.base}/shared/lib/*.jar
```

```
tomcat.util.scan.DefaultJarScanner.jarsToSkip=\
bootstrap.jar,commons-daemon.jar,tomcat-juli.jar,\
annotations-api.jar,el-api.jar,jsp-api.jar,servlet-api.jar,websocket-api.jar,\
catalina.jar,catalina-ant.jar,catalina-ha.jar,catalina-tribes.jar,\
jasper.jar,jasper-el.jar,ecj-*.jar,\
tomcat-api.jar,tomcat-util.jar,tomcat-coyote.jar,tomcat-dbcp.jar,\
tomcat-jni.jar,tomcat-spdy.jar,\
tomcat-il8n-en.jar,tomcat-il8n-es.jar,tomcat-il8n-fr.jar,tomcat-il8n-ja.jar,\
tomcat-juli-adapters.jar,catalina-jmx-remote.jar,catalina-ws.jar,\
tomcat-jdbc.jar,\
tools.jar,\
commons-beanutils*.jar,commons-codec*.jar,commons-collections*.jar,\
commons-dbcp*.jar,commons-digester*.jar,commons-fileupload*.jar,\
commons-httpclient*.jar,commons-io*.jar,commons-lang*.jar,commons-logging*.jar,\
commons-math*.jar,commons-pool*.jar,\
jstl.jar,\
geronimo-spec-jaxrpc*.jar,wsdl4j*.jar,\
ant.jar,ant-junit*.jar,aspectj*.jar,jmx.jar,h2*.jar,hibernate*.jar,httpclient*.jar,\
jmx-tools.jar,jta*.jar,log4j.jar,log4j-1*.jar,mail*.jar,slf4j*.jar,\
xercesImpl.jar,xmlParserAPIs.jar,xml-apis.jar,\
junit.jar,junit-*.jar,hamcrest*.jar,org.hamcrest*.jar,ant-launcher.jar,\
cobertura-*.jar,asm-*.jar,dom4j-*.jar,icu4j-*.jar,jaxen-*.jar,jdom-*.jar,\
jetty-*.jar,oro-*.jar,servlet-api-*.jar,tagsoup-*.jar,xmlParserAPIs-*.jar,\
xom-*.jar
```

```
org.apache.catalina.startup.ContextConfig.jarsToSkip=
```

```
org.apache.catalina.startup.TldConfig.jarsToSkip=tomcat7-websocket.jar
```

```
# String cache configuration.
tomcat.util.buf.StringCache.byte.enabled=true
#tomcat.util.buf.StringCache.char.enabled=true
#tomcat.util.buf.StringCache.trainThreshold=500000
#tomcat.util.buf.StringCache.cacheSize=5000
```

alfresco-global.properties

- Create the **alfresco-global.properties** file:

```
vi /usr/local/www/alfresco/tomcat/shared/classes/alfresco-global.properties
```

- And add the following, adjusting as necessary:

```
### Common Alfresco Properties

## File locations
dir.root=/usr/local/www/alfresco/alf_data
dir.contentstore=${dir.root}/contentstore
dir.contentstore.deleted=${dir.root}/contentstore.deleted
dir.cachedcontent=${dir.root}/cachedcontent
dir.auditcontentstore=${dir.root}/audit.contentstore
dir.keystore=${dir.root}/keystore
dir.indexes=${dir.root}/lucene-indexes
dir.indexes.backup=${dir.root}/backup-lucene-indexes
solr.backup.alfresco.remoteBackupLocation=${dir.root}/backupsolr
solr.backup.archive.remoteBackupLocation=${dir.root}/backupsolr

## Database
db.username=alfresco
db.password=alfresco
db.name=alfresco
db.host=localhost
db.pool.initial=10
db.pool.max=90

# PostgreSQL
db.driver=org.postgresql.Driver
db.port=5432
db.url=jdbc:postgresql://${db.host}:${db.port}/${db.name}

# MySQL
#db.driver=com.mysql.jdbc.Driver
#db.port=3306
#db.url=jdbc:mysql://${db.host}:${db.port}/${db.name}?useUnicode=yes&characterEncoding=UTF-8
#db.pool.validate.query=select 1

## System parameters
alfresco.context=alfresco
alfresco.host=alfresco.example.com
alfresco.port=8080
alfresco.protocol=http

share.context=share
share.host=shared.example.com
share.port=80
share.protocol=http

site.public.group=GROUP_EVERYONE

## Performance
system.usages.enabled=false
```

```

## External locations
ooo.exe=/usr/local/lib/libreoffice/program/soffice
ooo.enabled=true
swf.exe=/usr/local/bin/pdf2swf
img.exe=/usr/local/bin/convert
img.root=/usr/local/etc/ImageMagick-7
img.config=${img.root}
# Check this path if you get "no decode delegate for this image format" error
img.coders=/usr/local/lib/ImageMagick-7.0.8/modules-Q16/coders/
img.dyn=/usr/local/lib
img.gslib=/usr/local/share/ghostscript/9.27/lib

## Index
index.subsystem.name=solr
index.recovery.mode=AUTO

solr.host=localhost
solr.port=8080
solr.port.ssl=8443
#Effectively turn off solr backup
solr.backup.alfresco.cronExpression=0 0 2 * * ? 2099
solr.backup.archive.cronExpression=0 0 4 * * ? 2099

## Workflow engine
system.workflow.engine.jbpm.definitions.visible=false
system.workflow.engine.activiti.definitions.visible=true
system.workflow.engine.jbpm.enabled=false
system.workflow.engine.activiti.enabled=true

## Activities Feed and Subscriptions
activities.feed.notifier.repeatIntervalMins=1440
activities.feed.notifier.enabled=false
activities.feed.max.size=100
activities.feed.max.ageMins=44640
# Enables the subscription service
subscriptions.enabled=true

## Email

# SMTP
mail.host=smtp.example.com
mail.port=25
mail.username=anonymous
mail.password=
mail.encoding=UTF-8
mail.from.default=alfresco@demo.alfresco.org
mail.protocol=smtp

# Additional Java Mail properties for SMTP protocol
mail.smtp.auth=false
mail.smtp.debug=false
mail.smtp.timeout=5000
mail.smtp.starttls.enable=false

# Additional Java Mail properties for SMTPS protocol
mail.smtps.auth=false
mail.smtps.starttls.enable=false
#use these properties to send test message during start of subsystem
mail.testmessage.send=false
mail.testmessage.to=
mail.testmessage.subject=Outbound SMTP
mail.testmessage.text=The Outbound SMTP email subsystem is working.

#IMAP
#imap.server.enabled=true
#imap.server.port=143

```

```
#imap.server.host=localhost

## File Servers

# WebDAV initialization properties
system.webdav.servlet.enabled=true
system.webdav.rootPath=${protocols.rootPath}

cifs.enabled=false
filesystem.avm.enabled=false
cifs.tcpipSMB.port=1445
cifs.netBIOSSMB.sessionPort=1139
cifs.netBIOSSMB.namePort=1137
cifs.netBIOSSMB.datagramPort=1138

ftp.enabled=false
ftp.port=2021
```

share-config-custom.xml

- Create the **share-config-custom.xml** file:

```
vi /usr/local/www/alfresco/tomcat/shared/classes/alfresco/web-extension/share-config-custom.xml
1
```

- And add the following:

```
<alfresco-config>

  <!-- Global config section -->
  <config replace="true">
    <flags>
      <!-- Developer debugging setting to turn on DEBUG mode for client scripts in the
browser -->
      <client-debug>false</client-debug>
      <!-- LOGGING can always be toggled at runtime when in DEBUG mode (Ctrl, Ctrl, Shi
ft, Shift).
      This flag automatically activates logging on page load -->
      <client-debug-autologging>false</client-debug-autologging>
    </flags>
  </config>

  <config evaluator="string-compare" condition="WebFramework">
    <web-framework>
      <autowire>
        <!-- Pick the mode: "production" or "development" -->
        <mode>production</mode>
      </autowire>
    </web-framework>
  </config>

  <config evaluator="string-compare" condition="CSRFPolicy" replace="true">
    <properties>
      <token>Alfresco-CSRFToken</token>
      <referer>https?:\\\/\\\/shared.example.com\\\/.*</referer>
      <origin>https?:\\\/\\\/shared.example.com.*</origin>
    </properties>
  </config>

  <config evaluator="string-compare" condition="Remote">
    <remote>
      <endpoint>
        <id>alfresco-noauth</id>
        <name>Alfresco - unauthenticated access</name>
```

```

        <description>Access to Alfresco Repository WebScripts that do not require authentication</description>
        <connector-id>alfresco</connector-id>
        <endpoint-url>http://alfresco.example.com:8080/alfresco/s</endpoint-url>
        <identity>none</identity>
    </endpoint>

    <endpoint>
        <id>alfresco</id>
        <name>Alfresco - user access</name>
        <description>Access to Alfresco Repository WebScripts that require user authentication</description>
        <connector-id>alfresco</connector-id>
        <endpoint-url>http://alfresco.example.com:8080/alfresco/s</endpoint-url>
        <identity>user</identity>
    </endpoint>

    <endpoint>
        <id>alfresco-feed</id>
        <name>Alfresco Feed</name>
        <description>Alfresco Feed - supports basic HTTP authentication via the EndpointProxyServlet</description>
        <connector-id>http</connector-id>
        <endpoint-url>http://alfresco.example.com:8080/alfresco/s</endpoint-url>
        <basic-auth>true</basic-auth>
        <identity>user</identity>
    </endpoint>

    <endpoint>
        <id>activiti-admin</id>
        <name>Activiti Admin UI - user access</name>
        <description>Access to Activiti Admin UI, that requires user authentication</description>
        <connector-id>activiti-admin-connector</connector-id>
        <endpoint-url>http://alfresco.example.com:8080/alfresco/activiti-admin</endpoint-url>
        <identity>user</identity>
    </endpoint>
</remote>
</config>
</alfresco-config>

```

- Set ownership of alfresco files and folders:

```
chown -LR alfresco:www /usr/local/www/alfresco
```

Keystore

- Download the keystore files from alfresco:

```

cd /usr/local/www/alfresco/alf_data/keystore
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/browser.pl2
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/generate_keystores.sh
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/keystore
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/keystore-passwords.properties
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/ssl-keystore-passwords.properties
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/ssl-truststore-passwords.properties

```



```
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/ssl.keystore
fetch http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/repository/config/alfresco/keystore/ssl.truststore
```

Install Alfresco

- Download the war files for alfresco and share:

```
cd /usr/local/www/alfresco/tomcat/webapps
fetch https://artifacts.alfresco.com/nexus/service/local/repo_groups/public/content/org/alfresco/alfresco/5.2.f/alfresco-5.2.f.war
fetch https://artifacts.alfresco.com/nexus/service/local/repo_groups/public/content/org/alfresco/share/6.2.0/share-6.2.0.war
```

- Install the jar files to merge amp files (if needed):

```
cd /usr/local/www/alfresco/addons
fetch https://artifacts.alfresco.com/nexus/service/local/repo_groups/public/content/org/alfresco/alfresco-mmt/6.0/alfresco-mmt-6.0.jar
```

Install Solr

- Install the war files:

```
cd /usr/local/www/alfresco/tomcat/webapps
fetch https://artifacts.alfresco.com/nexus/service/local/repo_groups/public/content/org/alfresco/alfresco-solr/6.1/alfresco-solr-6.1.war
```

- Download and extract the solr config:

```
cd /usr/local/www/alfresco/solr
fetch https://artifacts.alfresco.com/nexus/service/local/repo_groups/public/content/org/alfresco/alfresco-solr/6.1/alfresco-solr-6.1-config.zip
```

- Backup and modify solr configs:

```
cd /usr/local/www/alfresco/solr
mv workspace-SpacesStore/conf/solrcore.properties workspace-SpacesStore/conf/solrcore.properties.orig
mv archive-SpacesStore/conf/solrcore.properties archive-SpacesStore/conf/solrcore.properties.orig
sed "s|@@ALFRESCO_SOLR_DATA_DIR@@|/usr/local/www/alfresco/alf_data/solr|g" workspace-SpacesStore/conf/solrcore.properties.orig > workspace-SpacesStore/conf/solrcore.properties
sed "s|@@ALFRESCO_SOLR_DATA_DIR@@|/usr/local/www/alfresco/alf_data/solr|g" archive-SpacesStore/conf/solrcore.properties.orig > archive-SpacesStore/conf/solrcore.properties
```

- Create the solr tomcat config:

```
vi /usr/local/www/alfresco/tomcat/conf/Catalina/localhost/solr.xml
```

- And add the following:

```
<?xml version="1.0" encoding="utf-8"?>
<Context debug="0" crossContext="true">
  <Environment name="solr/home" type="java.lang.String" value="/usr/local/www/alfresco/solr" override="true"/>
  <Environment name="solr/model/dir" type="java.lang.String" value="/usr/local/www/alfresco/solr/alfrescoModels" override="true"/>
  <Environment name="solr/content/dir" type="java.lang.String" value="/usr/local/www/alfresco/alf_data/solr" override="true"/>
</Context>
```

Init Script

- Create the alfresco init script:

```
vi /usr/local/etc/rc.d/alfresco
```

- And add the following:

```
#!/bin/sh
#
# PROVIDE: alfresco
# REQUIRE: DAEMON
# BEFORE: LOGIN
# KEYWORD: shutdown

export JAVA_HOME="/usr/local/openjdk8/"
export JRE_HOME=$JAVA_HOME/jre
export ALF_HOME="/usr/local/www/alfresco"
export CATALINA_HOME="$ALF_HOME/tomcat"
export CATALINA_PID="${ALF_HOME}/tomcat.pid"
export USER="alfresco"
export GROUP="www"

export LC_ALL=en_US

# IMPORTANT Updated to match memory available on your server.
# For production, A server with at least 8G ram, and -Xmx6G is recommended. More is better
!
export JAVA_OPTS="-Xms1G -Xmx2G -Xss1024k -XX:MaxPermSize=256m"

export JAVA_OPTS="${JAVA_OPTS} -Duser.country=US -Duser.region=US -Duser.language=en -Duser.timezone=\"America\Los_Angeles\" -d64"

#Enable this if you encounter problems with transformations of certain pdfs. Side effect is disable of remote debugging
#export JAVA_OPTS="${JAVA_OPTS} -Djava.awt.headless=true"

#File encoding may be correct, but we specify them to be sure
export JAVA_OPTS="${JAVA_OPTS} -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8"
export JAVA_OPTS="${JAVA_OPTS} -Dalfresco.home=${ALF_HOME} -Dcom.sun.management.jmxremote=true"
export JAVA_OPTS="${JAVA_OPTS} -server"

. /etc/rc.subr

name="alfresco"
rcvar="alfresco_enable"
procname="java"
pidfile="${CATALINA_PID}"

start_precmd="${name}_prestart"
start_cmd="${name}_start"

stop_precmd="${name}_prestop"
```

```

stop_cmd=":"
stop_postcmd="${name}_poststop"

alfresco_prestart()
{
    if [ ! -d "$ALF_HOME/logs" ]; then
        mkdir -p "$ALF_HOME/logs"
        chown -R $USER:$GROUP $ALF_HOME/logs
    fi
}

alfresco_start()
{
    cd $ALF_HOME/logs
    su -m ${USER} -c "${CATALINA_HOME}/bin/catalina.sh start"
}

alfresco_prestop()
{
    SHUTDOWN_PORT=`sockstat | grep 8005 | wc -l`
    if [ "$SHUTDOWN_PORT" -eq "0" ]; then
        echo "WARNING! Trying to shutdown before properly started. Waiting 120 seconds before stopping."
        sleep 120
        su -m ${USER} -c "${CATALINA_HOME}/bin/catalina.sh stop"
        sleep 10
    else
        su -m ${USER} -c "${CATALINA_HOME}/bin/catalina.sh stop"
        sleep 10
    fi
}

alfresco_poststop()
{
    rm -rf $CATALINA_HOME/temp/*
}

load_rc_config $name
run_rc_command "$1"

```

- Make the script executable:

```
chmod +x /usr/local/etc/rc.d/alfresco
```

- Enable

Resources

- <https://github.com/andergrim/alfresco-freebsd-install/blob/master/alfinstall.sh>
- <https://artifacts.alfresco.com/nexus/>

History

#1 - 11/14/2019 10:43 AM - Daniel Curtis

- Description updated
- Status changed from New to In Progress
- % Done changed from 0 to 30

#2 - 11/14/2019 10:51 AM - Daniel Curtis

- Description updated

#3 - 11/14/2019 10:48 PM - Daniel Curtis

- Description updated

#4 - 11/15/2019 11:47 PM - Daniel Curtis

- Description updated

#5 - 11/16/2019 12:45 AM - Daniel Curtis

- Description updated

#6 - 11/17/2019 10:42 PM - Daniel Curtis

- Description updated

#7 - 12/12/2020 01:02 AM - Daniel Curtis

- Status changed from In Progress to Suspended