Shibboleth 3 med hög tillgänglighet

Webinar

Våren 2016 hölls ett webinar om hur man konfigurerar Shibboleth 3.2 med hög tillgänglighet. Detta webinar går att lysnna på på sidan SWAMID Webinar 2 2016.

Beroenden

Lägg till JPA Storage Service

Konfiguration

Lägg till följande i idp.properties:

```
idp.session.StorageService = shibboleth.JPAStorageService
idp.replayCache.StorageService = shibboleth.JPAStorageService
idp.artifact.StorageService = shibboleth.JPAStorageService
```

Används CAS och/eller consent behövs även motsvarande rad för dem:

```
idp.cas.idp.session.StorageService = shibboleth.JPAStorageService
idp.consent.StorageService = shibboleth.JPAStorageService
```

MASTER-MASTER replikering

Exempel på uppsättning av MySQL

- http://www.ryadel.com/en/mysql-master-master-replication-setup-in-5-easy-steps/

Hela databasen ”shibboleth” behöver replikeras

En nod bör vara master
Gör alla förändringar på denna nod, bygg war-fil, och kopiera allt till övriga noder
Kopiera conf och creds till den nya noden, tänk på att kopiera detta igen när konfigurationen har förändrats
conf/, credentials/, metadata/, ssl/, views/ & war/

Uppdatera följande lösenord på slavarna
I filen /opt/jetty/jetty-base/start.d/idp.ini

- jetty.backchannel.keystore.password
- jetty.browser.keystore.password

Tänk på att uppdatera lösenordet på shibboleth-användaren i MySQL
Man kan även ändra root lösenordet på noderna så att man har samma

Sealer.* måste vara samma mellan alla noder och roteras dagligen
Se: /opt/idp-installer/bin/dailytasks.sh
En nod bör generera den och pusha ut den till övriga noder
SSH-nycklar behöver skapas för att pusha sealer.* till slavar alternativt så kan ett gemensamt filsystem användas för transport av sealer

Förslag på script som skapar och scp:ar sealer

```
#!/bin/bash
```

```
#
#  Daily IdP housekeeping tasks
#
#  A shell script to handle daily housekeeping tasks for the IdP
#
# Installation location: /opt/idp-installer/bin
# Expected crontab to roll key at 11pm localtime daily:
# "0 23  *   *   *     /opt/idp-installer/bin/dailytasks.sh > /dev/null 2>&1"
#
# Functions:
#
#       perform Shib v3 secret Key roll over
#               see: https://wiki.shibboleth.net/confluence/display/IDP30/SecretKeyManagement
#
# ${APPROOT}/conf/ha.master should hold master name if any
# ${APPROOT}/conf/ha.slaves should hold slave names if any

export JAVA_HOME=/usr/java/default

APPROOT="/opt/idp-installer"
APPBIN="${APPROOT}/bin"

LOGFILE="${APPROOT}/status.log"
ECHO="echo -e "
HOSTNAME=`hostname -s`
SLAVE=no
IDP_HOME="/opt/shibboleth-idp"
# Maximum age in hours
MAXAGE=2

${ECHO} `date` "$$:==================BEGIN=========" &> >(tee -a ${LOGFILE})

if [ -r ${LOGFILE} ] && [ $(date +%w) -eq 0 ]; then
    ${ECHO} `date` "$$:Rotating ${LOGFILE}" &> >(tee -a ${LOGFILE})
    mv ${LOGFILE} ${LOGFILE}.old
fi


# Assume I am a slave if there is a master whose name is not mine.
if [ -r "${APPROOT}/conf/ha.master" ] && [ ! $(cat "${APPROOT}/conf/ha.master" | sed 's/\..*//') =
"${HOSTNAME}" ]; then
    SLAVE=yes
fi

# Only regenerate if a master or secret key is missing or old
if [ "${SLAVE}" = "no" ] || [ ! -r "${IDP_HOME}/credentials/sealer.jks" ] || \
    [ $(( $(stat --printf="%Y\n" "${IDP_HOME}/credentials/sealer.jks" ) + $((60 * 60 * ${MAXAGE})) )) -lt
$(date +%s) ]; then

    ${ECHO} `date` "$$:Function 1/1:Doing Secret Key Rollover" &> >(tee -a ${LOGFILE})


    # trick: the pivot for the awk parsing is on the 'd=' in 'Password=' to preserve things if '=' is the
last character (or not)

    STORE_PASS="$(cat ${IDP_HOME}/conf/idp.properties|grep idp.sealer.storePassword |awk -F'd=' '{print
$2}'| tr -d '[[:space:]]')"

    ${ECHO} `date` "$$:  Step 1/2:Make Backup of credentials/sealer.jks" &> >(tee -a ${LOGFILE})
    CMDF1S1="cp -p ${IDP_HOME}/credentials/sealer.jks ${IDP_HOME}/credentials/sealer.jks.
recentPreviousVersion"
            eval ${CMDF1S1} &> >(tee -a ${LOGFILE})

    ${ECHO} `date` "$$:  Step 2/2:Perform Update" &> >(tee -a ${LOGFILE})
    CMDF1S2='${IDP_HOME}/bin/seckeygen.sh --storefile ${IDP_HOME}/credentials/sealer.jks --storepass
"${STORE_PASS}" --versionfile ${IDP_HOME}/credentials/sealer.kver --alias secret'
            eval ${CMDF1S2} &> >(tee -a ${LOGFILE})

    # Copy the regenerated key to all known slaves.
    if [ "${SLAVE}" = "no" ] && [ -r "${APPROOT}/conf/ha.slaves" ] && [ $(wc -l <"${APPROOT}/conf/ha.
slaves") -gt 0 ]; then
```

```
        for HOST in $(cat "${APPROOT}/conf/ha.slaves"); do
            if [ ! ${HOSTNAME} = ${HOST} ] && $(ping -c 1 -q ${HOST} >/dev/null 2>&1); then
                scp -p -q "${IDP_HOME}"/credentials/sealer.* ${HOST}:"${IDP_HOME}/credentials/" && \
                    ${ECHO} `date` "$$:Copy Secret Key to ${HOST}" &> >(tee -a ${LOGFILE})
            fi
        done
    fi

else
    ${ECHO} `date` "$$:I am a slave, passing by" &> >(tee -a ${LOGFILE})
fi

${ECHO} `date` "$$:==================END=========" &> >(tee -a ${LOGFILE})
```

Skapa följande katalog

- /opt/idp-installer/conf

Skapa följande filer

- /opt/idp-installer/conf/ha.master - Ska innehålla hostname på master-noden
- /opt/idp-installer/conf/ha.slaves - Ska innehålla hostname på alla slav-noder, en per rad

Ändra på cron-jobbet för dailytasks.sh på samtliga slav-noder så att det körs 23:30

Ett script som gör det mesta, dock ingen sync av sealer.*, används på egen risk:

```
#!/bin/bash
cat << EOM > /tmp/newTable
CREATE TABLE IF NOT EXISTS StorageRecords (
  context varchar(255) NOT NULL,
  id varchar(255) NOT NULL,
  expires bigint(20) DEFAULT NULL,
  value longtext NOT NULL,
  version bigint(20) NOT NULL,
  PRIMARY KEY (context,id),
  KEY storagerecords_expires_index (expires)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
GRANT ALL PRIVILEGES ON shibboleth.StorageRecords TO 'shibboleth'@'localhost';
EOM
cat << EOM > /tmp/newAddToGlobal
    <bean id="shibboleth.JPAStorageService"
            class="org.opensaml.storage.impl.JPAStorageService"
            p:cleanupInterval="%{idp.storage.cleanupInterval:PT10M}"
            c:factory-ref="shibboleth.JPAStorageService.EntityManagerFactory" />

    <bean id="shibboleth.JPAStorageService.EntityManagerFactory"
            class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
        <property name="persistenceUnitName" value="storageservice" />
        <property name="packagesToScan" value="org.opensaml.storage.impl" />
        <property name="dataSource" ref="shibboleth.JPAStorageService.DataSource" />
        <property name="jpaVendorAdapter" ref="shibboleth.JPAStorageService.JPAVendorAdapter" />
        <property name="jpaDialect">
            <bean class="org.springframework.orm.jpa.vendor.HibernateJpaDialect" />
        </property>
    </bean>

    <bean id="shibboleth.JPAStorageService.JPAVendorAdapter"
            class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
        <property name="database" value="MYSQL" />
    </bean>
    <bean id="shibboleth.JPAStorageService.DataSource"
        class="com.zaxxer.hikari.HikariDataSource" destroy-method="close" lazy-init="true"
        p:driverClassName="com.mysql.jdbc.Driver"
        p:jdbcUrl="jdbc:mysql://127.0.0.1:3306/shibboleth?autoReconnect=true&amp;localSocketAddress=127.0.0.1
&amp;connectTimeout=1800&amp;initialTimeout=2&amp;logSlowQueries=true&amp;autoReconnectForPools=true"
        p:username="shibboleth"
        p:password="%{idp.storage.databasePassword:p@ssw0rd}" />
EOM
```

```
#curl --silent -k -L --output /opt/shibboleth-idp/edit-webapp/WEB-INF/lib/bonecp-0.8.0.RELEASE.jar
http://repo1.maven.org/maven2/com/jolbox/bonecp/0.8.0.RELEASE/bonecp-0.8.0.RELEASE.jar
curl --silent -k -L --output /opt/shibboleth-idp/edit-webapp/WEB-INF/lib/HikariCP-2.4.3.jar http://search.
maven.org/remotecontent?filepath=com/zaxxer/HikariCP/2.4.3/HikariCP-2.4.3.jar
#curl --silent -k -L --output /tmp/apache-tomcat-7.0.68.tar.gz http://apache.mirrors.spacedump.net/tomcat
/tomcat-7/v7.0.68/bin/apache-tomcat-7.0.68.tar.gz
#cd /tmp/
#tar xf apache-tomcat-7.0.68.tar.gz apache-tomcat-7.0.68/lib/tomcat-jdbc.jar
#mv apache-tomcat-7.0.68/lib/tomcat-jdbc.jar /opt/shibboleth-idp/edit-webapp/WEB-INF/lib/
#rm -rf /tmp/apache-tomcat-7.0.68*
mkdir -p /opt/shibboleth-idp/edit-webapp/WEB-INF/classes/META-INF
curl --silent -k -L --output /opt/shibboleth-idp/edit-webapp/WEB-INF/classes/META-INF/orm.xml https://www.
switch.ch/aai/guides/idp/installation/orm.xml
cnt=$(grep -n "^#idp.session.StorageService" /opt/shibboleth-idp/conf/idp.properties | tail -n 1 | cut -d: -
f1)
((cnt++))
sed -i "${cnt}i idp.session.StorageService = shibboleth.JPAStorageService" /opt/shibboleth-idp/conf/idp.
properties
((cnt++))
sed -i "${cnt}i idp.replayCache.StorageService = shibboleth.JPAStorageService" /opt/shibboleth-idp/conf/idp.
properties
((cnt++))
sed -i "${cnt}i idp.artifact.StorageService = shibboleth.JPAStorageService" /opt/shibboleth-idp/conf/idp.
properties
dbPW=$(grep "p:password" /opt/shibboleth-idp/conf/global.xml |tail -n 1 |cut -d\" -f2)
echo "idp.storage.databasePassword=${dbPW}" >> /opt/shibboleth-idp/conf/idp.properties
cnt=$(grep -n "</beans>" /opt/shibboleth-idp/conf/global.xml | tail -n 1 | cut -d: -f1)
((cnt--))
sed -i "${cnt}r /tmp/newAddToGlobal" /opt/shibboleth-idp/conf/global.xml
JAVACMD=/usr/bin/java /opt/shibboleth-idp/bin/build.sh -Didp.target.dir=/opt/shibboleth-idp

mysql -u root -p < /tmp/newTable

rm /tmp/newAddToGlobal /tmp/newTable
```