DIGITAL SUNDHED
SAMMENHÆNGENDE DIGITAL SUNDHED I DANMARK

SOSIGW

- **Installationsvejledning for SOSIGW 1.2**

Indeks

Revisionshistorik

Version	Dato	Ændring	Ansvarlig
1	09/10/09	Initiel version af dokument	jre
2	15/10/09	Formattering og præciseringer	jre
3	19/10/09	Præciseringer	pto
4	19/10/09	Præciseringer	Pto
5	30/3/11	Dokumentation af muligheden for flere STS-url'er. Forsimpling af NSP-Util SLA-log konfiguration efter opgradering til NSP-util v 1.0.0	Jrf
6	15/6/11	Dokumentation af installation på JBoss 6	JRF
7	29/08/13	Opdateret med NSP-GW muligheder	CHE
8	11/05/16	Dokumentation af installation på Wildfly 8.2	OBJ
9	17/06/16	Dokumentation af opsætning af HTTPS	OBJ

Introduktion

Denne vejledning beskriver arbejdsgange ved opsætning af SOSI-GW til driftsformål.

Forudsætninger og krav

Der kræves maskiner med mindst to separate netværk. Dette krav kan opfyldes enten ved at benytte servere med to fysiske netkort eller ved at benytte VLAN. I begge tilfælde skal de to net konfigureres til at være separate. Formålet er at have et net, der udelukkende kan tilgås af SOSI-GW serverne, til intra-cluster kommunikation. Til test og udviklingsformål kan man nøjes med ét netkort.

Der kræves maskiner med tilstrækkeligt med RAM og CPU til at bære belastningen. Tests hos Trifork har vist at det er tilstrækkeligt med 1.5GB RAM og en 1.5GHz cpu for at servicere omkring 20.000 brugere med omkring 100.000 webservice-kald gennem SOSI-GW i timen. Der kan med fordel anvendes maskiner med flere cores/CPU'er, hvis større belastning forventes.

Der bør anvendes mindst to servere og en loadbalancer (eller auto-balancerende klienter) for at kunne benytte fail-over og rullende opgradering uden afbrydelse af servicen overfor brugerne. Der kan også opsættes flere maskiner, hvis der ønskes skalerbarhed ud over hvad to maskiner kan bære.

SOSI-GW er fuldstændig afhængig af korrekt systemtid og korrekt tidszone. Systemtiden indgår i de id-kort, der udstedes, og den skal ikke være mange sekunder forkert, før id-kortet ikke fungerer. Cluster-funktionaliteten kræver at systemtiden forløber kontinuerligt, mens den er mindre interesseret i at den er præcis. Som samlet resultat kræves der at serveren benytter NTP eller lignende, med en god implementation, der aldrig skifter tiden brat, men kun påvirker den ved

glidende at tilpasse systemtiden til virkeligheden. Skulle det blive nødvendigt ”hårdt” at ”stille” systemtiden bør SOSI-GW stoppes på den pågældende server først.

Installér ønsket JDK.

SOSI-GW kræver mindst JDK version 1.5.0 og er testet på Sun JDK-1.5.0_13 og Sun JDK-1.6.0_03.

Konfigurer JDK til ubegrænset kryptering

Dette er taget fra SOSI-SEAL vejledning. Det har ikke været nødvendigt for at gennemføre testene, men det nævnes som nødvendigt af SOSI-SEAL.

Export Policy

JDK 1.4 is shipped with policy files that supports strong but not unbounded encryption strength. However, SUN does distribute policy files that allows unbounded encryption strength which is needed by the SOSI component:

- Download og extract **US_export_policy.jar** and **local_policy.jar** from
  - Sun 1.4.2: http://java.sun.com/j2se/1.4.2/download.html (in the bottom part of the page)
  - Sun 1.5: http://java.sun.com/javase/downloads/index_jdk5.jsp
  - IBM 1.4.2: http://www-128.ibm.com/developerworks/java/jdk/security/142/
- Copy these two files to **$JRE_HOME/lib/security** and overwrite the existing files.

Nyere Java versioner

For Java7 og Java8 konfigureres dette tilsvarende. De 2 jar-filer findes her:

- Java7: http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html
- Java8: http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

Installér og konfigurer databaser

SOSI-GW kræver en lokal database til audit-logging og konfigurationsdata for hver SOSI-GW instans, samt en fælles, delt, database til audit-logging. Der er til test anvendt version 8.2.5 af PostgreSQL. Det er også muligt at anvende MySQL

Det anbefales at køre knuderne på et RAID1-disksystem, så et disknedbrud ikke medfører tab af auditlog-data.

Opret lokal database instanser, PostgreSQL

Se script i sosigw/etc/sql/create-local-db.sql og tips i sosigw/doc/postgresql-setup-notes.txt. Der er til test benyttet charset=”UTF-8” i databasen. Bemærk at der IKKE

må anvendes de passwords, som foreslås i scriptene! Det er uforsvarligt sikkerhedsmæssigt at drifte med password lig med brugernavn.

Opret global database instans, PostgreSQL

Se script i sosigw/etc/sql/create-global-db.sql og tips i sosigw/doc/postgresql-setup-notes.txt. Der er til test benyttet charset="UTF-8" i databasen. Bemærk at der IKKE må anvendes de passwords, som foreslås i scriptene! Det er sikkerhedsmæssigt uforsvarligt at drifte med password lig med brugernavn.

Opret lokal database instanser, MySQL

Se script i sosigw/etc/sql/create-local-mysql-db.sql og tips i sosigw/doc/postgresql-setup-notes.txt. Der er til test benyttet charset="UTF-8" i databasen. Bemærk at der IKKE må anvendes de passwords, som foreslås i scriptene! Det er uforsvarligt sikkerhedsmæssigt at drifte med password lig med brugernavn.

Opret global database instans, MySQL

Se script i sosigw/etc/sql/create-global-mysql-db.sql og tips i sosigw/doc/postgresql-setup-notes.txt. Der er til test benyttet charset="UTF-8" i databasen. Bemærk at der IKKE må anvendes de passwords, som foreslås i scriptene! Det er sikkerhedsmæssigt uforsvarligt at drifte med password lig med brugernavn.

Installér webcontainer

SOSI-GW kræver en servlet-container version 2.4. Der er til test og udvikling benyttet Apache Tomcat version 5.5.25 og Trifork T4 version 4.1.29. SOSI-GW er desuden afhængig af NSP-Util som kræver at opsætningsfilen '`nspslalog-sosigw.properties`' findes i de nedenstående mapper. Placeringen af nspslalog-sosigw.properties kan desuden defineres eksplicit ved at sætte "dk.sdsd.nsp.slalog.config.dir" som Java property ved serveropstart. For eksempler på indhold i konfigurationsfilen henvises til NSP-util kildekoden i 'etc/nspslalog-EXAMPLE.properties'

JBoss 4

Den "saaj" der følger med i JAX-WS-RI skal være "endorsed". Det kan løses ved at kopiere følgende filer fra {sosigw-jboss4.war}/WEB-INF/lib til JBOSS.HOME/lib/endorsed/:

- javax.xml-webservices-api.jar
- xalan-xalan.jar

JBoss har som standard en threadpool på 250, hvilket begrænser SOSI-GW til at håndtere 250 samtidige requests. Hvis dette er for lavt til formålet, så hæv "maxThreads" i server.xml på den Connector, der bruges til http. Lad være med at konfigurere tomcat clustering. SOSI-GW har sin egen cluster-funktionalitet.

Deploy sosigw-jboss4.war til JBoss ved at kopiere filen ind i ”JBOSS.HOME/server/default/deploy”.

Kopier og tilret konfigurationsfiler som beskrevet i senere afsnit. Følgende filer skal kopieres:

- `log4j-sosigw.xml` skal lægges i JBOSS.HOME /server/default/conf
- `sosigw-staticconf.properties` skal lægges i JBOSS.HOME /server/default/conf
- `nspslalog-sosigw.properties` skal lægges i JBOSS.HOME /server/default/conf

Ved start af JBoss skal det sikres at der reserveres tilstrækkeligt med RAM til Java heap. Dette gøres for tiden ved at sætte ”`JAVA_OPTS=-Xmx1024m`” før start, eller rette i JBOSS.HOME/bin/run.conf. 1024 megabytes til heapen kræver at maskinen har mindst 1.2GB ram til rådighed.

JBoss 6

Der er ikke behov for endorsed libraries ved JBoss 6 deployment. Dog er det vigtigt at det er sosigw-jboss6.war der deployes, da den er renset for libraries som jboss6 leverer.

JBoss har som standard en threadpool på 250, hvilket begrænser SOSI-GW til at håndtere 250 samtidige requests. Hvis dette er for lavt til formålet, så hæv ”maxThreads” i server.xml på den Connector, der bruges til http. Lad være med at konfigurere tomcat clustering. SOSI-GW har sin egen cluster-funktionalitet.

Deploy sosigw-jboss6.war til JBoss ved at kopiere filen ind i ”JBOSS.HOME/server/default/deploy”.

Kopier og tilret konfigurationsfiler som beskrevet i senere afsnit. Følgende filer skal kopieres:

- `log4j-sosigw.xml` skal lægges i JBOSS.HOME /server/default/conf
- `sosigw-staticconf.properties` skal lægges i JBOSS.HOME /server/default/conf
- `nspslalog-sosigw.properties` skal lægges i JBOSS.HOME /server/default/conf

Ved start af JBoss skal det sikres at der reserveres tilstrækkeligt med RAM til Java heap. Dette gøres for tiden ved at sætte ”`JAVA_OPTS=-Xmx1024m`” før start, eller rette i JBOSS.HOME/bin/run.conf. 1024 megabytes til heapen kræver at maskinen har mindst 1.2GB ram til rådighed.

Apache Tomcat

Den “saaj” der følger med i JAX-WS-RI skal være “endorsed”. Det kan løses ved at kopiere følgende filer fra {sosigw.war}/WEB-INF/lib til tomcat/common/endorsed:

- javax.xml-webservices-api.jar
- xalan-xalan.jar

Tomcat 5.5.25 har som standard en threadpool på 150, hvilket begrænser SOSI-GW til at håndtere 150 samtidige requests. Hvis dette er for lavt til formålet, så hæv ”maxThreads” i server.xml på den Connector, der bruges til http. Lad være med at konfigurere tomcat clustering. SOSI-GW har sin egen cluster-funktionalitet.

Deploy sosigw.war til Tomcat ved at kopiere filen ind i ”webapps”.

Kopier og tilret konfigurationsfiler som beskrevet i senere afsnit. Følgende filer skal kopieres:

- `log4j-sosigw.xml` skal lægges i TOMCAT.HOME/conf/
- `sosigw-staticconf.properties` skal lægges i TOMCAT.HOME/conf/

Ved start af Tomcat skal det sikres at der reserveres tilstrækkeligt med RAM til Java heap. Dette gøres for tiden ved at sætte ”`JAVA_OPTS=-Xmx1024m`” før start. 1024 megabytes til heapen kræver at maskinen har mindst 1.2GB ram til rådighed.

Trifork T4

Der er en konflikt mellem den “saaj” der er med i T4 af hensyn til J2EE 1.4 og den “saaj” der følger med i JAX-WS-RI. Den kan løses ved at kopiere flg filer fra {sosigw.war}/WEB-INF/lib til TRIFORK.DOMAIN.DIR/lib/share/endorsed:

- javax.xml-webservices-api.jar
- xalan-xalan.jar

(Bemærk at på unix kræves mindst T4 version 4.1.29 for at endorsed genkendes. Tidligere versioner kaldte kataloget for /lib/share**d**/endorsed i start-scriptet, “eas”. Windows versionen kalder fortsat i 4.1.29 kataloget for share**d** i eas.cmd. Dette er rettet i version 4.1.30)

Der skal oprettes en database connection pool i T4. Brug enten administrationskonsollen, eller to kommandolinier i stil med følgende:

```
\Trifork-4.1.29\domains\default\bin\trifork.cmd jdbc
createdatasource -dbusername sosigwlog -dbpassword sosigwlog
sosigwlocaldb jdbc/physicalsosigwlocaldb org.postgresql.Driver
Cloudscape jdbc:postgresql://localhost:5432/sosigwlog

\Trifork-4.1.29\domains\default\bin\trifork.cmd jdbc
createpooleddatasource sosigwlocaldbPool jdbc/postgres
jdbc/physicalsosigwlocaldb
```

Husk at udskifte password til databasen i første linie.

Gennem administrationskonsollen i T4 skal størrelsen af threadpoolen ændres fra dens default, 40, til et passende, større tal. Dette tal begrænser hvor mange samtidige klienter, SOSI-GW vil håndtere. Det sættes typisk højt, men lavt nok til ikke at løbe tør for ”fildescriptors/handles”. Til test er benyttet 400.

Deploy sosigw.war til T4.

Kopier og tilret konfigurationsfiler som beskrevet i senere afsnit. Følgende filer skal kopieres:

- `log4j-sosigw.xml` skal lægges i TRIFORK.DOMAIN.DIR/config/SERVER.NAME/
- `sosigw-staticconf.properties` skal lægges i TRIFORK.DOMAIN.DIR/config/SERVER.NAME/

Ved start af T4 skal det sikres at der reserveres tilstrækkeligt med RAM til Java heap. Dette gøres for tiden ved at tilføje option "`-vmargs -Xmx1024m`" til "trifork server start" kommandoen. 1024 megabytes til heapen kræver at maskinen har mindst 1.2GB ram til rådighed.

Wildfly 8.2

1. Definer JDBC driver som modul. De nødvendige biblioteker og filer placeres i modules/system/layers/base. Et eksempel på definition af MySQL findes i etc/wildfly/modules.

2. Tilret datasource-definitioner i filen standalone/configuration/standalone.xml. Følgende er et eksempel på definition af MySQL datasources. Tilret serveradresse, brugernavn og password:

```
<datasources>
	...
	<datasource jndi-name="java:/jdbc/sosigwlocal" pool-name="sosigwlocal"
	enabled="true" use-java-context="true">
		<connection-url>jdbc:mysql://localhost:3306/sosigwlog</connection-url>
		<driver>mysql</driver>
		<security>
			<user-name>xxx</user-name>
			<password>xxx</password>
		</security>
	</datasource>
	<datasource jndi-name="java:/jdbc/nspslalog" pool-name="nspslalog"
	enabled="true" use-java-context="true">
		<connection-url>jdbc:mysql://localhost:3306/nspslalog</connection-url>
		<driver>mysql</driver>
		<security>
			<user-name>xxx</user-name>
			<password>xxx</password>
		</security>
	</datasource>
	<datasource jndi-name="java:/jdbc/sosigwgloballog" pool-name="sosigwgloballog"
	enabled="true" use-java-context="true">
		<connection-url>jdbc:mysql://localhost:3306/sosigwgloballog</connection-
		url>
		<driver>mysql</driver>
		<security>
			<user-name>xxx</user-name>
			<password>xxx</password>
		</security>
	</datasource>
	<datasource jndi-name="java:/jdbc/centralslalog" pool-name="centralslalog"
	enabled="true" use-java-context="true">
		<connection-url>jdbc:mysql://localhost:3306/sosigwcentrallog</connection-
		url>
		<driver>mysql</driver>
		<security>
			<user-name>xxx</user-name>
			<password>xxx</password>
		</security>
	</datasource>
	...
	<drivers>
		...
		<driver name="mysql" module="com.mysql">
```

```
                    <driver-class>com.mysql.jdbc.Driver</driver-class>
                </driver>
            </drivers>
</datasources>
```

Et eksempel på definitioner findes i etc/wildfly/standalone/configuration/ standalone.xml.

3. Såfremt SOSI-GW skal kunne kaldes med HTTPS, konfigureres dette ligeledes i standalone.xml:

   Under ”security-realms” tilføjes følgende:

```
<security-realm name="SslRealm">
  <authentication>
    <truststore path="truststore.jks" relative-to="jboss.server.config.dir" keystore-
password="changeme"/>
  </authentication>
  <server-identities>
    <ssl>
      <keystore path="keystore.jks" relative-to="jboss.server.config.dir" keystore-
password="changeme"/>
    </ssl>
  </server-identities>
</security-realm>
```

   En java-keystore med et gyldigt SSL servercertifikat, udgivet af en gængs certifikatudbyder, placeres i standalone/configuration. Filnavn og password tilrettes i ssl/keystore linjen vist ovenfor.

   I afsnittet authentication/truststore angives en anden java-keystore med navn og password, i hvilke SSL-klient-certifikater, der eksplicit skal trustes, kan placeres. Denne keystore placeres ligeledes i standalone/configuration.

   Under linjen `<http-listener name="default" socket-binding="http`"/> tilføjes en ny linje:

```
<https-listener name="default-ssl" socket-binding="https" security-realm="SslRealm"
verify-client="REQUIRED"/>
```

   verify-client=”REQUIRED” bevirker at der valideres for gyldigt og trusted klient-certifikat i SSL-handshake. Dette kan undlades, hvis klient-certifikat ikke skal valideres.

   Default https-port er 8443. Dette kan tilrettes i samme fil.

4. Tilret konfiguration af max. memory allocation pool:

```
set "JAVA_OPTS=-Xms64M -Xmx1024M -XX:MaxPermSize=256M"
```

   Dette tilrettes i bin/standalone.conf.bat eller bin/standalone.conf, afhængigt af anvendt operativsystem.

5. Kopier konfigurationsfiler til standalone\configuration:

   - sosigw-staticconf.properties
   - log4j-sosigw.xml

6. Tilret JDK modul mht. xerces. Indsæt følgende linjer:

```
<path name="com/sun/org/apache/xerces/internal/dom"/>
<path name="com/sun/org/apache/xerces/internal/xni"/>
<path name="com/sun/org/apache/xerces/internal/jaxp"/>
```

   i filen modules/system/layers/base/sun/jdk/main/module.xml. Eksempel på filen findes i etc/wildfly/modules.

7. Deploy sosigw.war til Wildfly ved at kopiere filen ind i standalone/deployments

Bemærk, ovenstående beskriver gælder for Wildfly 8.2 i Standalone mode, hvilket er den tilstand, der svarer mest til JBoss 6, hvor hver serverinstans har sin egen konfiguration. Wildfly kan også afvikles i Domain mode, men dette kræver en anden konfigurationen.

Statisk konfiguration

Der er gjort en indsats for at begrænse den statiske konfiguration til det mindst mulige. Det drejer sig primært om parametre, der skal være ens på tværs af clusteret og som muliggør at maskinerne i clusteret kommunikerer, og om hvorvidt dette system anvendes til test eller produktion.

Alle parametre står i filen ”`sosigw-staticconf.properties`” med kommentarer om deres funktion. Følgende **skal** rettes før første opstart af applikationen.

- cluster.ip.prefix

ip-prefix'er skal sættes til f.eks. ”10.7.”, hvis det lukkede net til intra-cluster kommunikation har fået tildelt adresser i 10.7.x.y nettet. Intentionen med denne indstilling er at forhindre at uvedkommende får mulighed for at sende pakker til clusteret, samt at øge stabilitet og performance ved at benytte et separat net til formålet.

- sosigw.mode

Den valgte mode afspejler formålet med clusteret og vælger hvilken STS der benyttes. Til testformål benyttes værdien ”test” og til drift benyttes ”production”. Den eneste forskel mellem test og production er hvilken STS, der anvendes. Ved opsætning af et nyt cluster benyttes først værdien ”bootstrap”, der sætter SOSI-GW i en mode, hvor den ikke deltager i cluster og ikke processerer webservice-requests, men til gengæld tillader adgang til administrationskonsollen uden adgangskontrol. Formålet er at lade en administrator få adgang til at tilføje sig selv til listen over brugere med login-mulighed i konsollen, samt at opsætte initielle restriktioner på hvilke webservices, der kan kaldes gennem den. Når dette er sket skiftes til enten ”test” eller ”production”, og øvrige servere i clusteret startes herefter. Det er kun nødvendigt at benytte ”bootstrap” på den første server, da nye servere herefter får konfigurationen foræret af de allerede kørende servere.

Ændringer i sosigw-staticconf.properties bliver ikke automatisk indlæst og kræver derfor en genstart af serveren. Indholdet af filen distribueres heller ikke automatisk, så dette skal gøres udefra via fx rsync.

Konfiguration af administrationskonsollen

Den dynamiske konfiguration sker gennem en webbaseret administrationskonsol.
For at anvende administrationskonsollen i test eller produktion, er det nødvendigt at definere hvilke brugere, der har adgang til konsollen. Det gøres ved først at starte en SOSIGW-instans i bootstrap-mode. Dette opnås ved at sætte sosigw.mode=bootstrap i sosigw-staticconf.properties, genstarte SOSIGW og åbne administrationskonsollen.
Hvis der er flere knuder i clusteret, er det nødvendigt at tilgå bootstrap-knuden direkte og ikke gennem en loadbalancer. Dette er kun nødvendigt under den initielle konfiguration af konsollen.

Tilføj konsolbruger
Under Configuration -> Administrative Console Users vælges New for at tilføje en ny bruger og følgende indtastes:

- Name: Brugerens CPR-nummer i henhold til det certifikat, der skal logges ind med
- Fornavn: Brugerens fornavn.
- Efternavn: Brugerens efternavn.
- Email: Brugerens emailadresse.

Opsæt Systemnavn og -CVR
Ydermere skal systemet konfigureres til det rette SOSI systemnavn og CVR-nummer:

- Under Configuration -> General Properties -> careprovider.cvr sættes det CVR-nummer, som SOSIGW skal operere under. Alle certifikatlogins skal ske med certifikater, der har dette nummer – dette gælder både for login til administrationskonsol og for browsersigneringsrequests.
- Configuration -> General Properties -> careprovider.cvr.name sættes til organisationens navn.

Opsæt STS-Url(er)
For produktion skal den rette STS desuden udpeges under Configuration -> General Properties -> sts.service.url.

Værdien af denne property kan sættes til at være en enkelt url til den STS, man ønsker at kalde, fx
`http://nsp-test-sts.trifork.com:8180/test-sts/services/SecurityTokenService`.
For at give større fejltolerance eller skalerbarhed, er det også muligt at angive en liste af URL'er. Listen angives som værdi til samme property (sts.service.url) og skal være adskilt af mellemrum. Et eksempel:

```
http://nsp-test-sts.trifork.com:8180/test-sts/services/SecurityTokenService
http://pan.certifikat.dk/sts/services/SecurityTokenService
```

Hvis den første STS i listen ikke kan kontaktes, hvis kaldet fejler eller hvis svartiden er for lang, kontaktes den næste i listen. Skiftet til en alternativ STS sker uden at den kaldende service får besked – hvis nsp-test-sts.trifork.com fejler, mens pan.certifikat.dk svarer i ovenstående eksempel, vil den oprindelige klient altså få svaret fra pan.certifikat.dk uden at kunne se, at SOSI-GW først forsøgte nsp-test-sts.trifork.com.

Det tidsrum, der ventes på svar fra en STS, før den opgives og den næste kontaktes, er konfigurerbar i property

`sts.timeout.millis`

Værdien er som standard 10000. Hvis man kører med kun én STS konfigureret, vil man muligvis ønske at sætte værdien højere.

Opsæt circuit breaker

I en situation, hvor en STS (fx nsp-test-sts.trifork.com i eksemplet overfor) bliver overbelastet og begynder at svare langsommere, vil alle klientkald skulle vente timeout-værdien (sts.timeout.millis), før SOSI-GW kontakter en alternativ STS og kan returnere svaret til klienten.
Derfor er der for hver STS implementeret en såkaldt Circuit Breaker. Dens funktion er at holde styr på, hvor mange kald mod en STS, der er fejlet i træk. Når antallet når over en fastsat værdi, holder SOSI-GW helt op med at kontakte denne STS – man siger, at circuit breakeren *åbner*.
Antallet af tilladte, fejlende kald i træk er konfigurerbart i property

`sts.number.failures.before.open.cb`

Efter et stykke tid, hvor circuit breakeren har været åben, forsøger SOSI-GW et enkelt ”prøve-kald” mod STS’en. Hvis dette går godt, begynder SOSI-GW igen at lave forespørgsler mod STS’en. Det stykke tid der går inden prøvekaldet foretages, er konfigurerbart i property

`sts.millis.before.attempting.close.cb`

Opsæt håndtering af SSL

Ønskes at SOSI-GW altid skal initiere udgående HTTPS forbindelser, kan der i konfigurationen angives HTTPS-adresser på endpoints til STS og DCC (i det omfang dette understøttes).

Hvis der angives en URL i ”To” i WsAddressing SOAP-headeren, vil disse gennemstilles uændret til NSP services af SOSI-GW Proxy-operationen, også selvom der er tale om almindelige HTTP-adresser.

Såfremt der ønskes at disse viderestilles som HTTPS, kan følgende property konfigureres:

```
force-ssl-requests.enabled=true
```

Dette vil bevirke at HTTP erstattes af HTTPS i URL’en, og at der initieres en udgående SSL/TLS 1.2 krypteret forbindelse når der anvendes Wildfly 8.2 på Java 8.

Skift væk fra bootstrapmode og genstart

Når det er gjort, sættes sosigw.mode til enten test eller produktion, og serveren genstartes for at få ændringerne til at træde i kraft.

Bemærk at i test-mode er SLA-logning per default slået fra, og i produktion-mode er SLA-logning per default slået til. Ønskes dette styret eksplicit sættes SLA-logning med parameteren sosigw.slalog (TRUE/FALSE).

Opsætningen skal kun gennemføres på en af instanserne i et cluster. Konfigurationen distribueres automatisk til de øvrige instanser.