DIGITAL SUNDHED

SAMMENHÆNGENDE DIGITAL SUNDHED I DANMARK

SOSIGW

- **Installationsvejledning for SOSIGW 1.2, NSP**

Indeks

Revisionshistorik

Version	Dato	Ændring	Ansvarlig
1	23/8/11	Nyt dokument baseret på ”Installationsvejledning for SOSI-GW 1.1” rev. 6. Dette dokument beskriver kun NSP-installation	JRF
2	29/08/13	Opdateret med NSP-Gateway informationer	CHE

Introduktion

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

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.

Installér webcontainer

SOSI-GW kører på NSP på JBoss AS 6.

JBoss 6

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

Kopier følgende filer fra releasen:

- `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 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.

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.

Konfiguration

Alle parametre, både dem der traditionelt har været kaldt statiske og dem der tidligere er kaldt runtimekonfiguration, står i filen ”`sosigw-staticconf.properties`” med kommentarer om deres funktion.

Det er vigtigt, at property runtimeconfig.storage.type bevarer værdien properties, ellers vil SOSI-GW forsøge at indlæse runtimekonfigurationen fra database, hvilket ikke ønskes på NSP.

Derudover **skal** følgende 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, samt at SLA-logning per default er slået fra i ”test”-mode (dette kan overstyres i ”test”-mode ved brug af konfigurationsparameteren sosigw.slalog). 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

Den dynamiske konfiguration sker på NSP også i filen sosigw-staticconf.properties. Instanser skal som for statisk konfiguration genstartes før ændringer slår igennem.

Dette kan ske uden tab af IDkort cachen gennem ”rullende genstart”, dvs én søjle af gangen genstartes.

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

- runtimeconfig.general.careprovider.cvr sættes til 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.
- runtimeconfig.general.careprovider.cvr.name sættes til organisationens navn.

Opsæt STS-Url(er)
For produktion skal den rette STS udpeges i property
runtimeconfig.general.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
`runtimeconfig.general.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
`runtimeconfig.general.sts.millis.before.attempting.close.cb`

Opsætning af SLA logning

SOSI-GW benyttter NSP-Util til SLA-logning.
Dette kræver at opsætningsfilen '`nspslalog-sosigw.properties`' findes i samme directory som sosigw-staticconf.properties.
For eksempler på indhold i konfigurationsfilen henvises til NSP-util kildekoden i 'etc/nspslalog-EXAMPLE.properties'

Opsætning af Cache Partitionering (NSP-Gateway)

Hvis SOSI-GW installeres i et miljø hvor idkort cachen ønskes partionieret (f.eks. på NSP-Gateway) sættes runtime propertyen
`runtimeconfig.general.cache-partitioning.enabled`
til '`true`' og propertyen
`runtimeconfig.general.cache-partitioning.http-header.name`
sættes til navnet på den http header som infrastrukturen anvender til at identificere de enkelte anvenderorganisationer, eksempelvis kunne den sættes til '`SOSIGW_PARTITION`'.

Hvis SOSI-GW er placeret *før* DCC'en (som f.eks. på NSP-Gateway) skal SOSI-GW kende DCC'ens endpoint. Det konfigureres med propertyen runtimeconfig.general.decoupling-component.url, der sættes til endpointet for DCC'en.

Hvis idkort cachen er partitioneret, er der særlige krav til konfiguration af den foranstående loadbalancer:

- Loadbalanceren skal tilføje en passende http-header, der unikt identificerer de enkelte anvendere (og det skal konfigureres i SOSI-GW hvilket headernavn, der benyttes – se ovenfor)
- Det skal være muligt at mappe imellem de ID'er, loadbalanceren tildeler de enkelte anvendere, og anvenderne, idet der ved konfigurationen af browserendpoints (se nedenfor) er behov for at koblingen mellem ID og anvender

Der er endvidere mulighed for at konfigurere browserendpoints til de enkelte anvendere (til applet-signering af IDkort). Den eksisterende parameter for url'en (runtimeconfig.general.browsersigning.url) er default, og der kan for hver anvender

tilføjes en konfiguration på formen runtimeconfig.general.browsersigning.url.XXX, hvor XXX er partitionsID'et tildelt af infrastrukturen.

Testklient

Det er vanskeligt at verificere at konfigurationen for en produktions-NSP er korrekt. Selvom en driftstekniker måtte have de fornødne akkreditiver (medarbejdersignatur med tilknyttet CPR nummer) til at få udstedt et SOSI idkort er der ingen services vedkommende med rette må kalde. Derfor sker verifikation typisk først når klinikere forsøger at tilgå services gennem NSP platformen med den ulempe at fejlkonfigurationer opdages meget sent i et opsætnings- eller opgraderingsforløb og potientelt ender med at genere mange brugere.
For at kunne afprøve konfigurationer i et produktion (eller produktions-lignende) miljø er der derfor udviklet en simpel NSP test-service (NTS) der kan deployes på NSP samt udvikling af en kommando-linie testklient.
Test-servicen udbyder ingen kliniske data, men stiller nogle håndtag til rådighed der kan verificere at kaldet indeholder at gyldigt idkort med det korrekte autentifikations-niveau.
Testklienten giver mulighed for at en driftstekniker kan kalde test-servicen i gennem SOSI-GW og DCC og derved validere at SOSI-GW (og STS) er korrekt konfigureret for det pågældende miljø, dvs. enten et NSP-Gateway setup eller et standard regionalt NSP setup.

Klienten opretter et idkort i SOSI-GW cachen og kalder derefter NTS gennem gatewayen, kaldet foretages ved hjælp af JAX-WS klasser genereret ud fra en simpel wsdl fil.
Hvis ikke andet er angivet, så anvender klienten de samme default værdier som de eksisterende unit tests (taget fra `Abstract IntegrationTests.java`). Alle værdier kan specificeres med en property fil eller via kommandolinien.

```
NTSClient -- help
```

vil give en liste over de options der kan angives.

Der er lavet et ant target `nts-client`, hvor properties filen angives ved hjælp af

```
-Dntsclient.properties=site.properties
```

Følgende er et eksempel på en site.properties fil med de default værdier der anvendes:

```
# The value of the medcom:CareProviderID attribute in the SystemLog attribute-statement  - part of the SAML Assertion
idcard.careprovider.id=19343634
# The value of the medcom:CareProviderName attribute in the SystemLog attribute-statement  - part of the SAML Assertion
idcard.careprovider.name=Vi brækker dit hårde øre
# The name format of the medcom:CareProviderID attribute in the SystemLog attribute-statement - part of the SAML Assertion
```

```
idcard.careprovider.type=medcom:cvrnumber
# The issuer of the SAML Assertion
idcard.issuer=sosigwtester
# The value of the medcom:ITSystemName attribute in the SystemLog attribute-statement - part of the SAML Assertion
idcard.itsystem=SOSITEST
# The value of the SAML NameId
idcard.nameid=MitBrugerNavnSomErMereEnd10tegnLangt
# The value of the medcom:UserAuthorizationCode attribute in the UserLog attribute-statement - part of the SAML Assertion
idcard.user.authorizationcode=12345
# The value of the medcom:UserCivilRegistrationNumber attribute in the UserLog attribute-statement - part of the SAML Assertion
idcard.user.cpr=1111111118
# The value of the medcom:UserEmailAddress attribute in the UserLog attribute-statement - part of the SAML Assertion
idcard.user.email=test@trifork.com
# The value of the medcom:UserGivenName attribute in the UserLog attribute-statement - part of the SAML Assertion
idcard.user.givenname=Tester
# The value of the medcom:UserOccupation attribute in the UserLog attribute-statement - part of the SAML Assertion
idcard.user.occupation=Gråstenæblegrødskoger
# The value of the medcom:UserRole attribute in the UserLog attribute-statement - part of the SAML Assertion
idcard.user.role=læge
# The value of the medcom:UserSurName attribute in the UserLog attribute-statement - part of the SAML Assertion
idcard.user.surname=Blåbærgrød
# The alias of the certificate in the keystore that should be used
keystore.alias=SOSI:ALIAS_SYSTEM
# The password of the keystore containing the user certificate
keystore.password=!234Qwer
# The path to the keystore containing the user certificate
keystore.path=etc/validMocesVault.jks
# The host name and port number used by the NTS
nts.host=localhost:8080
# The host name and port number used by the SOSI GW
sosigw.host=localhost:8080
```