Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-1030 Wien, Seidlgasse 22 / 9
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

CLOUD-ELEMENTE SIGNATUR

Bericht

Sandra Kreuzhuber – sandra.kreuzhuber@a-sit.at

Zusammenfassung: In dem Projekt Cloud-Elemente Signatur wurde die Cloud-Fähigkeit von Signatur-Basiselementen analysiert. Konkret wurden der Signaturprüfdienst und in weiterer Folge MOA-SPSS und PDF-AS in einer Cloud-Umgebung deployed. Dieses Projekt untersucht die technischen Möglichkeiten, die rechtlichen Rahmenbedingungen (z.B. Datenschutz, Haftung etc.) bleiben jedoch unbeachtet.

Inhaltsverzeichnis

1. Einleitung

Die Vorteile des Betriebs von IT Services in der Cloud wie z.B. Kostenersparnis, bessere Skalierung etc. lassen die Verwendung von Cloud Services auch für E-Government Services attraktiv erscheinen. Da unterschiedliche PaaS-Provider beim Deployment von Services Einschränkungen auferlegen, wurde mit diesem Projekt die Cloud-Fähigkeit von Signatur-Basiselementen analysiert. Konkret wurden der Signaturprüfdienst und in weiterer Folge MOA-SPSS in einer aktuellen Cloud-Umgebung installiert. Dieses Projekt analysiert die technischen Einschränkungen eines möglichen Betriebs von Kernelement in der Cloud, die rechtlichen Rahmenbedingungen bleiben jedoch unbeachtet.

2. Cloud Provider

2.1. Jelastic

Jelastic[1] ist ein PaaS-Hosting System mit Spezialisierung auf das Hosting von Java und PHP Anwendungen. Jelastic ermöglicht das Hosting von Java 6 und Java 7 Applikationen. Weiters kann zwischen mehreren Applikationsservern, u.a. Tomcat und Glassfish gewählt werden. Unterschiedlichste Provider, u.a. Dogado[2], ServInt[3] etc. verwenden Jelastic als PaaS-Plattform. Abgerechnet wird bei Jelastic anhand sogenannter Cloudlets, die die verwendeten Ressourcen Arbeitsspeicher und Prozessor darstellen. Für den Demonstrator wurde der deutsche Cloudanbieter Dogado ausgewählt. Dogado verfügt über Testkontos, jedoch sind diese im

[1] http://jelastic.com/de/
[2] http://www.dogado.de/
[3] http://www.servint.net/

Funktionsumfang eingeschränkt. Weiters unterstützt Jelastic Custom Domains und die Verwendung eigener SSL Zertifikate. Das Deployment des Signaturprüfdienstes und in weiterer Folge MOA-SPSS und PDF-AS erfolgte ohne Änderungen am Quellcode der Applikationen. Abschnitt 3 beschreibt die notwendigen Schritte für ein erfolgreiches Deployment bei Jelastic.

2.2. Google App Engine

Google App Engine ist eine vergleichsweise restriktive PaaS-Plattform. Applikationen laufen in der Google App Engine nur mit eingeschränktem Zugriff auf Klassen der Java-Laufzeitumgebung[4]. Aufgrund der limitierten Rechte besteht keine Möglichkeit vom lokalen Dateisystem zu lesen oder darauf zu schreiben. Da Applikationen keinen Zugriff auf das Dateisystem haben, muss der Logging-Mechanismus auf die Standardausgabe konfiguriert werden. Des Weiteren müssen die für MOA-SPSS und den Signaturprüfdienst benötigten Konfigurationsdateien in die Applikation integriert werden. Da kein Zugriff auf JAVA_HOME/jre/lib/security und JAVA_HOME/jre/lib/ext besteht, ist es nötig verwendete Bibliotheken direkt in der Applikation zu inkludieren. Das Deployment des Signaturprüfdienstes bei Google App Engine wurde nicht weiter verfolgt.

2.3. Appfog

Appfog bietet Unterstützung für Java, Python, PHP und Ruby Anwendungen. Um Applikationen bei Appfog deployen zu können, ist die Installation eines Kommandozeilentools erforderlich. Über das Webinterface können die für die Konfiguration von MOA-SPSS notwendigen Umgebungsvariablen gesetzt werden. Das Deployment bei Appfog wurde nicht weiter betrachtet, da der Deploymentvorgang des Signaturprüfdienstes mit einem internen Fehler abgebrochen wurde. Die Logdateien gaben keinen Aufschluss über das Fehlschlagen des Deployment Prozesses. Generell kann angemerkt werden, dass ein Deployment bei Appfog prinzipiell möglich sein sollte.

3. Deployment bei Jelastic

Im Folgenden werden die wesentlichen Schritte zur Installation des Signaturprüfdienstes und in weiterer Folge MOA-SPSS behandelt[5]. Die dazu verwendete Jelastic Instanz ist unter http://moasp.jelastic.dogado.eu/test-signature-verification/[6] erreichbar.

Voraussetzungen (gewählte Konfiguration der Jelastic Instanz)**:**
Jelastic Instanz bei Dogado
Tomcat 7
Java 7
Installation des FTP Plugins[7]

3.1. MOA-SPSS

Java Bibliotheken

- Installation der Kryptographiebibliotheken von SIC/IAIK
  - Über das Webinterface die jar Dateien nach JAVA_HOME/jre/lib/ext hochladen
  - Installation der Unlimited Strength Jurisdiction Policy Dateien nach JAVA_HOME/jre/lib/security
- Installation der XML Parser Bibliotheken
  - Installation **nicht** nach endorsed/ sondern nach JAVA_HOME/jre/lib/ext

Umgebungsvariablen

- Über FTP in conf/variables.conf folgendes hinzufügen
  -Dmoa.spss.server.configuration=/opt/tomcat/conf/moa-spss/spss.config.xml

[4] Weiße Liste an verfügbaren Klassen der Java-Laufzeitumgebung: https://developers.google.com/appengine/docs/java/jrewhitelist?hl=de
[5] Anleitung zur Installation von MOA-SPSS auf lokalem Server: http://joinup.ec.europa.eu/site/moa-idspss/moa-spss-1.5.1/doc/handbook/install/install.html#webservice_basisinstallation_einführung
[6] Der Betrieb dieses Services bei Jelastic kann aus Kostengründen wieder eingestellt werden.
[7] http://jelastic.com/de/docs/ftp-ftps-support

-Dmoa.node.id=JelasticTestDeployment

***.war Datei deployen**

- Datei über das Webinterface hochladen oder Deployment über den FTP-Client

Konfiguration

- In der Datei /conf/signature-verification/pdf-as/config.properties verifizieren, dass die Eigenschaften
  - moa.sign.url=http://127.0.0.1:8080/moa-spss/services/SignatureCreation und
  - moa.verify.url=http://127.0.0.1:8080/moa-spss/services/SignatureVerification

  auf eine lokale URL zeigen.

3.2. Signaturprüfdienst

Notwendige Anpassungen der Applikation

- Entfernen der Datei ${SIGNATURPRÜFDIENST_WAR}/WEB-INF/libs/servlet-api-2.3.jar da ansonsten folgender Fehler Auftritt:

  INFO: validateJarFile(…\WEB-INF\lib\servlet-api.jar) - jar not loaded. See Servlet Spec 2.3, sectoin 9.7.2. Offending class: javax/servlet/Servlet.class[8]

- In ${SIGNATURPRÜFDIENST_WAR }/WEB-INF/web.xml Anpassung der relativen URLs

```
<context-param>
        <param-name>application_config_uri</param-name>
        <!-- if no value is provided, system property "signatureverification.configuration" is used to determine configuration uri -->
        <param-value>${catalina.home}/conf/signature-verification/application_config.xml</param-value>
</context-param>
<context-param>
        <param-name>formatdetection_config_uri</param-name>
        <!-- if no value is provided, system property "formatdetection.configuration" is used to determine configuration uri, otherwise the internal config is used -->
        <!--
        <param-value>${catalina.home}/conf/signature-verification/formatdetection-config.xml</param-value>
        <param-value></param-value>
        -->
        <param-value>${catalina.home}/conf/signature-verification/formatdetection-config.xml</param-value>
</context-param>

<context-param>
        <param-name>testenvironment_config_uri</param-name>
        <!--  Test-Environment config file -->
        <param-value>${catalina.home}/conf/signature-verification/testenvironment_config.xml</param-value>
```

[8] http://stackoverflow.com/questions/1993493/error-servlet-jar-not-loaded

</context-param>

- In In ${SIGNATURPRÜFDIENST_WAR }/WEB_INF/conf/axis2.xml Entfernen des <transportSender name="https"> Tags, falls kein SSL/TLS verwendet wird

***.war Datei deployen**

- Datei über das Webinterface hochladen[9] oder Deployment über den FTP-Client[10]

Konfiguration

- Konfiguration über FTP hinaufladen und in conf/signature-verification/application_config.xml
    - Änderung der Url zur verwendeten MOA-SPSS Instanz, Angabe einer lokalen URL, da Jelastic keine Sockets nach außen zulässt

<category name="moa">

<category name="sp">

<service.uri>SignatureVerification</service.uri>

<connection.url>**http://127.0.0.1:8080/moa-spss/services/SignatureVerification**</connection.url>

<trustprofile.id>signature-verification-trustprofile-id</trustprofile.id>

</category>

<category name="ss">

<key.idendifier>signature-verification-keygroup-id</key.idendifier>

<connection.url>**http://127.0.0.1:8080/moa-spss/services/SignatureCreation**</connection.url>

</category>

</category>

- Anpassung der relativen Pfade zu ${catalina.home}/conf/signature-verification/[11]

Ansonsten erscheint folgende Fehlermeldung:

[FATAL@22.07.2013 07:38:33] at.iaik.commons.jconfig.JConfigDefaultConfiguration:getInstance:230 - Cannot load main configuration "/opt/tomcat/temp/conf/signature-verification/application_config.xml".

[FATAL@22.07.2013 07:38:33] at.iaik.commons.jconfig.JConfigDefaultConfiguration:getInstance:210 - Cannot load main configuration via resource uri "/opt/tomcat/temp/conf/signature-verification/testenvironment_config.xml"

Umgebungsvariablen

- Umgebungsvariable zu application_config.xml setzen
    - Über FTP in conf/variables.conf folgendes hinzufügen

-Dsignatureverification.configuration=/opt/tomcat/conf/signature-verification/application_config.xml

[9] http://jelastic.com/de/docs/upload-deploy-application
[10] http://jelastic.com/de/docs/ftp-ftps-support#h
[11] Beim Upload von Konfigurationsdateien über die Weboberfläche von Jelastic ist eine Anpassung zu ${catalina.home}/temp/*conf/signature-verification/…* nötig. Siehe https://jelastic.zendesk.com/entries/21938233-Keeping-configuration-files-outside-war-file

4. Zusammenfassung

Jelastic bietet als PaaS-Provider nahezu dieselben Konfigurationsmöglichkeiten wie lokale Serverinfrastrukturen oder IaaS-Provider. Der Signaturprüfdienst mit MOA-SPSS und in weiterer Folge PDF-AS konnte ohne Anpassungen des Quellcodes und nur mit minimalen Änderungen der Konfigurationsdateien auf einer Jelastic-Instanz deployed werden. Aus technischer Sicht können somit Kernelemente der österreichischen E-Government Infrastruktur in der Cloud betrieben werden. Restriktivere Cloud-Anbieter wie z.B. Google App Engine oder Appfog erfordern Anpassungen an der Applikationsstruktur (z.B. Einbinden der Konfiguration in die *.war Datei). Google App Engine erfordert zusätzlich Änderungen am Quellcode. Generell ist ein Deployment jedoch auch bei diesen Anbietern möglich.