Gravitant
The Power to Transform

Synergy Plaza North
Jollyville Road, Ste 325N
Austin, TX 78759
Phone: 512-535-7399

Using Single Sign-On With cloudMatrix

Contents

Introduction

CloudMatrix supports SAML 2.0-based secure Single Sign-On (SSO) Integration using JBoss PicketLink. This can be used to provide seamless single sign on experience for end users across enterprise systems and cloudMatrix portal. Using this capability, customers can enable SSO for cloudMatrix with ADFS (Active Directory Federation Services), CA SiteMinder, Okta SSO, Tivoli SSO and other systems. Any SSO provider who supports SAML 2.0 can be used through simple configuration and testing.

CloudMatrix is a multi-tenant solution: in a single installation of cloudMatrix CSB, multiple customers can be enabled with different URLs (aka Affiliate URLs). CloudMatrix can be deployed without Single Sign On, or can be configured with Single Sign On to have a SSO provider across all the direct and Affiliate URLs mapped to the cloudMatrix portal.

This document covers configuring JBoss EAP 6.2 for SAML/SSO configuration, and the necessary cloudMatrix configurations to enable the system to authenticate for SAML rather than LDAP.

These procedures have been tested against Okta. Other SAML systems may require minor configuration changes.

Terminologies and Concepts

- **IDP (Identity Provider)**: A third party identity provider like Okta or an ADFS installation.
- **SSO (Single Sign-On)**: A system by which user authentication is handled by a central mechanism across a range of applications.
- **SP (Service Provider)**: The actual application or the service that the user is trying to access after authentication.
- **IDP URL**: The identity provider URL.
- **Service URL**: The URL for the service, which would be accessed after authentication.

Initial Setup

The following steps must be performed before any SAML solution can be integrated:

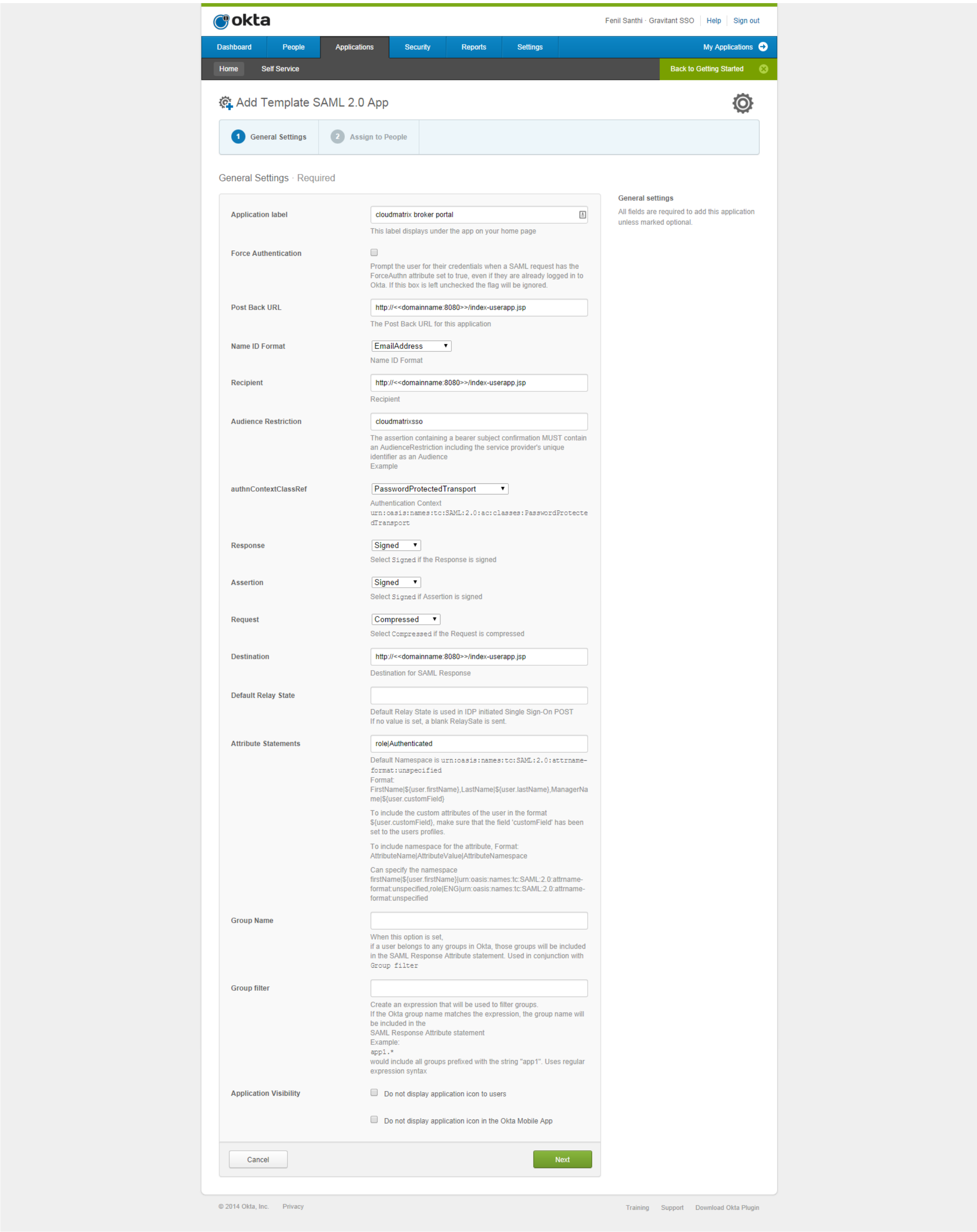
1. Obtain the identity provider URL from the provider after setup is finished.
2. Prepare the .xml file (`saml-configuration.xml.template`) to configure the system for SAML. Set the value to true if the system is to work in a SAML/SSO mode.
3. Configure cloudMatrix:
   a. Configure `picketlink.xml` under both cloudmatrix and platform-admin.
   b. Configure `jboss-web.xml` to point to the "sp" service provider security domain.
4. Prepare market maker login id. If the market maker login id different from cm-mm-admin@gravitant.com, then the steps mentioned in this section have to be reconfigured.
5. Perform billing Rest URL Changes.
6. Note the Affiliate white labelling on SAML.

Detailed Setup Instructions

Setup Identity Provider Configuration Example

The following steps detail SAML configuration settings for an example identity provider (in this case OKTA). After performing these configuration steps, you should get an Identity Provider URL (IDP). A pre-requisite for configuration in this section is the service provider URL.

1. Create an application configuration in the IDP provider (Okta). The Audience Restriction value needs to be the alias for the keystore when it is created. Enter the command `keytool –genkey -alias cloudmatrixsso -keyalg RSA -keystore c:\temp\okta_keystore.jks -keysize 2048`. (Note: You will need to download the digital certificate from Okta. This can be placed at ~/.)



Add Template SAML 2.0 App

1 General Settings · 2 Assign to People

General Settings · Required

General settings

All fields are required to add this application unless marked optional.

Application label: cloudmatrix broker portal
This label displays under the app on your home page

Force Authentication: □
Prompt the user for their credentials when a SAML request has the ForceAuthn attribute set to true, even if they are already logged in to Okta. If this box is left unchecked the flag will be ignored.

Post Back URL: http://<<domainname:8080>>/index-userapp.jsp
The Post Back URL for this application

Name ID Format: EmailAddress
Name ID Format

Recipient: http://<<domainname:8080>>/index-userapp.jsp
Recipient

Audience Restriction: cloudmatrixsso
The assertion containing a bearer subject confirmation MUST contain an AudienceRestriction including the service provider's unique identifier as an Audience
Example

authnContextClassRef: PasswordProtectedTransport
Authentication Context
`urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport`

Response: Signed
Select `Signed` if the Response is signed

Assertion: Signed
Select `Signed` if Assertion is signed

Request: Compressed
Select `Compressed` if the Request is compressed

Destination: http://<<domainname:8080>>/index-userapp.jsp
Destination for SAML Response

Default Relay State:
Default Relay State is used in IDP initiated Single Sign-On POST
If no value is set, a blank RelaySate is sent.

Attribute Statements: role|Authenticated
Default Namespace is `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`
Format:
FirstName|${user.firstName},LastName|${user.lastName},ManagerName|${user.customField}

To include the custom attributes of the user in the format ${user.customField}, make sure that the field 'customField' has been set to the users profiles.

To include namespace for the attribute, Format:
AttributeName|AttributeValue|AttributeNamespace

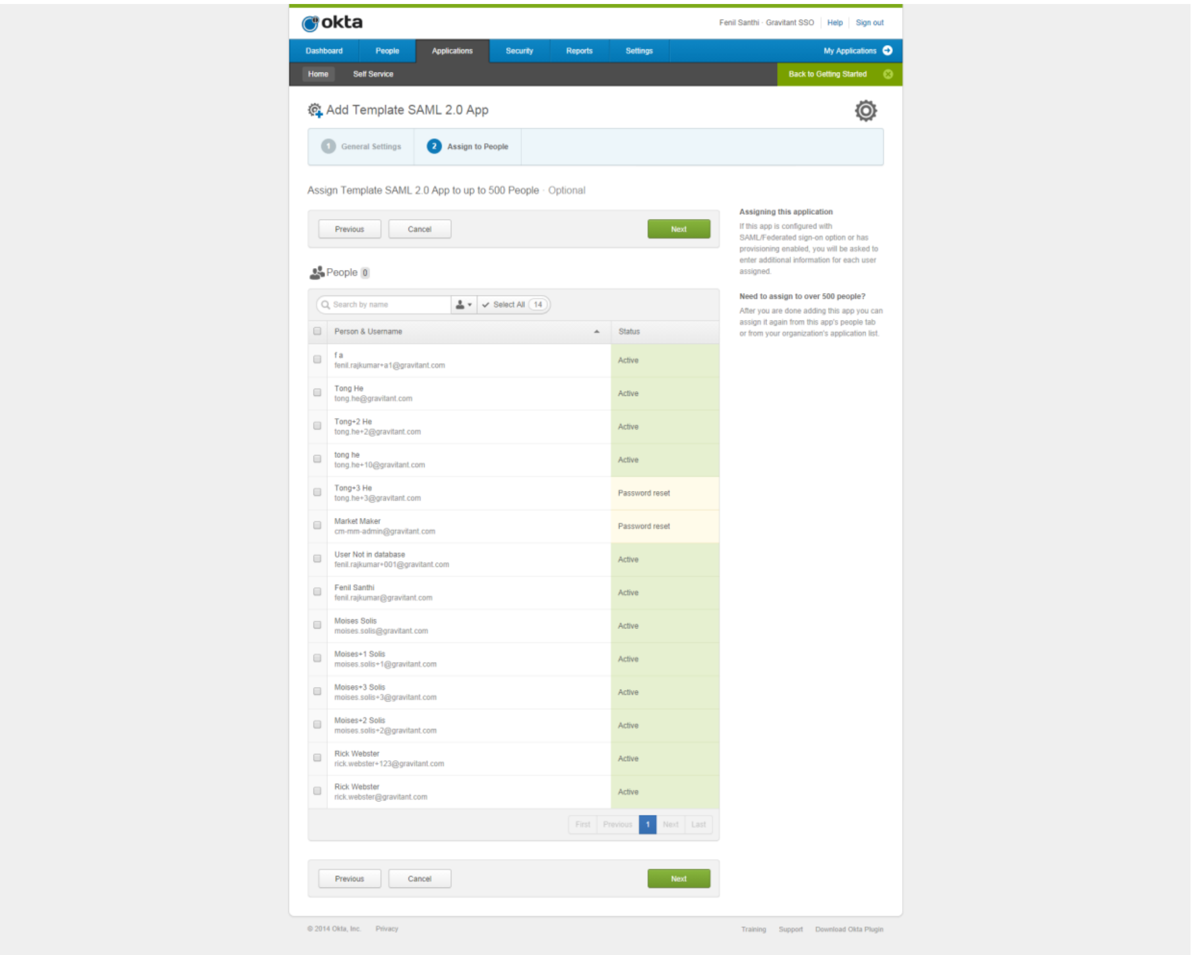
Can specify the namespace
firstName|${user.firstName}|urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified,role|ENG|urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified

Group Name:
When this option is set,
if a user belongs to any groups in Okta, those groups will be included in the SAML Response Attribute statement. Used in conjunction with `Group filter`

Group filter:
Create an expression that will be used to filter groups.
If the Okta group name matches the expression, the group name will be included in the
SAML Response Attribute statement
Example:
`app1.*`
would include all groups prefixed with the string "app1". Uses regular expression syntax

Application Visibility:
□ Do not display application icon to users
□ Do not display application icon in the Okta Mobile App

2. Click Next and Associate user logins to the new application (optional, as users can be associated later).



Add Template SAML 2.0 App

Assign Template SAML 2.0 App to up to 500 People · Optional

Assigning this application

If this app is configured with SAML/Federated sign-on option or has provisioning enabled, you will be asked to enter additional information for each user assigned.

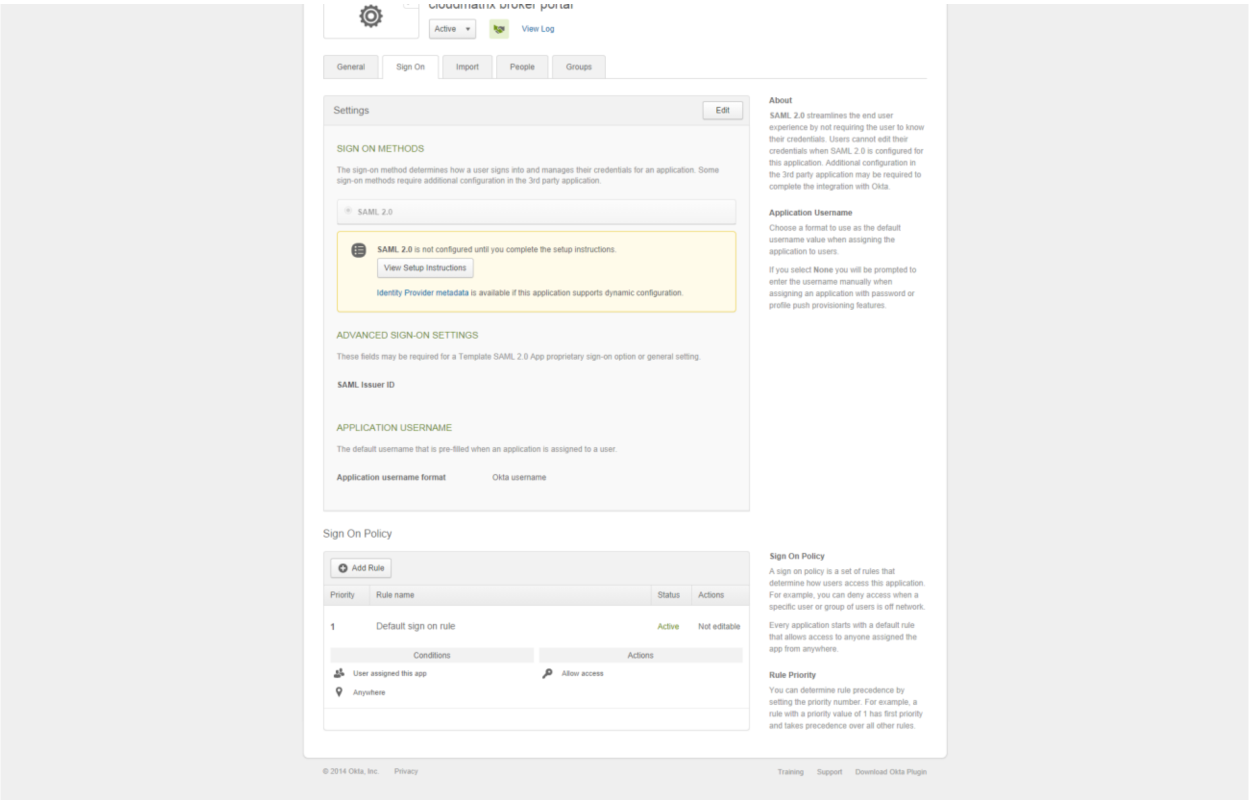
Need to assign to over 500 people?

After you are done adding this app you can assign it again from this app's people tab or from your organization's application list.

People 0

□	Person & Username	Status
□	f a fenil.rajkumar+a1@gravitant.com	Active
□	Tong He tong.he@gravitant.com	Active
□	Tong+2 He tong.he+2@gravitant.com	Active
□	tong he tong.he+10@gravitant.com	Active
□	Tong+3 He tong.he+3@gravitant.com	Password reset
□	Market Maker cm-mm-admin@gravitant.com	Password reset
□	User Not in database fenil.rajkumar+001@gravitant.com	Active
□	Fenil Santhi fenil.rajkumar@gravitant.com	Active
□	Moises Solis moises.solis@gravitant.com	Active
□	Moises+1 Solis moises.solis+1@gravitant.com	Active
□	Moises+3 Solis moises.solis+3@gravitant.com	Active
□	Moises+2 Solis moises.solis+2@gravitant.com	Active
□	Rick Webster rick.webster+123@gravitant.com	Active
□	Rick Webster rick.webster@gravitant.com	Active

3. Verify SAML Configuration by confirming that single-sign on works at login.



Settings

SIGN ON METHODS

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3rd party application.

○ SAML 2.0

SAML 2.0 is not configured until you complete the setup instructions.

Identity Provider metadata is available if this application supports dynamic configuration.

ADVANCED SIGN-ON SETTINGS

These fields may be required for a Template SAML 2.0 App proprietary sign-on option or general setting.

SAML Issuer ID

APPLICATION USERNAME

The default username that is pre-filled when an application is assigned to a user.

Application username format Okta username

About

SAML 2.0 streamlines the end user experience by not requiring the user to know their credentials. Users cannot edit their credentials when SAML 2.0 is configured for this application. Additional configuration in the 3rd party application may be required to complete the integration with Okta.

Application Username

Choose a format to use as the default username value when assigning the application to users.

If you select **None** you will be prompted to enter the username manually when assigning an application with password or profile push provisioning features.

Sign On Policy

Priority	Rule name	Status	Actions
1	Default sign on rule	Active	Not editable

Conditions	Actions
User assigned this app	Allow access
Anywhere	

Sign On Policy

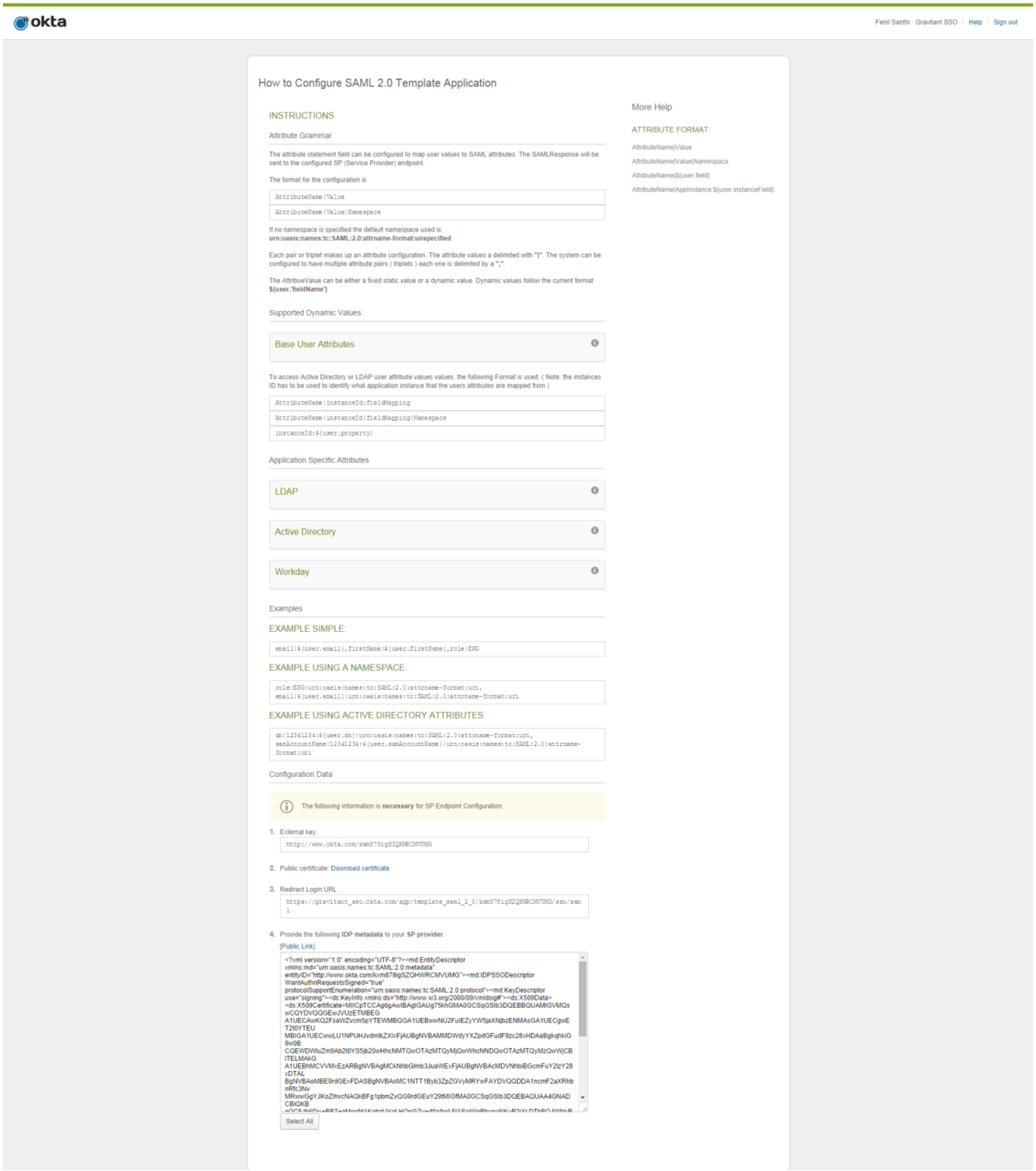
A sign on policy is a set of rules that determine how users access this application. For example, you can deny access when a specific user or group of users is off network.

Every application starts with a default rule that allows access to anyone assigned the app from anywhere.

Rule Priority

You can determine rule precedence by setting the priority number. For example, a rule with a priority value of 1 has first priority and takes precedence over all other rules.

4. View Detailed SAML Configuration for the certificate and the IDP URL details.



How to Configure SAML 2.0 Template Application

INSTRUCTIONS

Attribute Grammar

The attribute statement field can be configured to map user values to SAML attributes. The SAMLResponse will be sent to the configured SP (Service Provider) endpoint.

The format for the configuration is

```
AttributeName|Value
AttributeName|Value|Namespace
```

If no namespace is specified the default namespace used is:
urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified

Each pair or triplet makes up an attribute configuration. The attribute values a delimited with "**|**". The system can be configured to have multiple attribute pairs (triplets) each one is delimited by a "**,**".

The AttribueValue can be either a fixed static value or a dynamic value. Dynamic values follow the current format **${user.'fieldName'}**

Supported Dynamic Values

Base User Attributes

To access Active Directory or LDAP user attribute values values, the following Format is used. (Note: the instances ID has to be used to identify what application instance that the users attributes are mapped from)

```
AttributeName|instanceId:fieldMapping
AttributeName|instanceId:fieldMapping|Namespace
instanceId:${user.property}
```

Application Specific Attributes

LDAP

Active Directory

Workday

Examples

EXAMPLE SIMPLE:

```
email|${user.email},firstName|${user.firstName},role|ENG
```

EXAMPLE USING A NAMESPACE:

```
role|ENG|urn:oasis:names:tc:SAML:2.0:attrname-format:uri,
email|${user.email}|urn:oasis:names:tc:SAML:2.0:attrname-format:uri
```

EXAMPLE USING ACTIVE DIRECTORY ATTRIBUTES:

```
dn|12341234:${user.dn}|urn:oasis:names:tc:SAML:2.0:attrname-format:uri,
samAccountName|12341234:${user.samAccountName}|urn:oasis:names:tc:SAML:2.0:attrname-
format:uri
```

Configuration Data

The following information is **necessary** for SP Endpoint Configuration.

1. External key:
   `http://www.okta.com/kxm878igSZQHWRCMVUMG`
2. Public certificate: Download certificate
3. Redirect Login URL
   `https://gravitant_sso.okta.com/app/template_saml_2_0/kxm878igSZQHWRCMVUMG/sso/saml`
4. Provide the following **IDP metadata** to your **SP provider**.
   [Public Link]

```
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="http://www.okta.com/kxm878igSZQHWRCMVUMG"><md:IDPSSODescriptor
WantAuthnRequestsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><md:KeyDescriptor
use="signing"><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:X509Data>
<ds:X509Certificate>MIICpTCCAg6gAwIBAgIGAUg75khGMA0GCSqGSIb3DQEBBQUAMIGVMQs
wCQYDVQQGEwJVUzETMBEG
A1UECAwKQ2FsaWZvcm5pYTEWMBQGA1UEBwwNU2FuIEZyYW5jaXNjbzENMAsGA1UECgwE
T2t0YTEU
MBIGA1UECwwLU1NPUHJvdmlkZXIxFjAUBgNVBAMMDWdyYXZpdGFudF9zc28xHDAaBgkqhkiG
9w0B
CQEWDWluZm9Ab2t0YS5jb20wHhcNMTQwOTAzMTQyMjQwWhcNNDQwOTAzMTQyMzQwWjCB
lTELMAkG
A1UEBhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWExFjAUBgNVBAcMDVNhbiBGcmFuY2lzY28
xDTAL
BgNVBAoMBE9rdGExFDASBgNVBAsMC1NTT1Byb3ZpZGVyMRYwFAYDVQQDDA1ncmF2aXRhb
nRfc3Nv
MRwwGgYJKoZIhvcNAQkBFg1pbmZvQG9rdGEuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNAD
CBiQKB
```

JBoss and System Configuration

JBoss configuration must be updated to use SSO.

1. Modify jboss_home/gravitant/data/`saml-configuration.xml.template` to change the SSO enabled flag from false (default) to true to enable SSO for cloudMatrix.

2. If the system is running under SAML, the jboss-web.xml in this case is jboss-web-saml.xml. (If you’re running LDAP, the proper XML file would be jboss-web-ldap.xml.)
3. Configure cloudMatrix XML files:
    a. Modify `deployments\cloudmatrix.ear\cloudmatrixweb.war\WEB-INF\jboss-web.xml` via a Chef script.
    b. Modify `deployments\cloudmatrix.ear\cloudmatrixweb.war\WEB-INF\picketlink.xml` by entering the IDP and SP URL. This picketlink.xml points to its own application with its own URL. It will not redirect to `/admin` if this is incorrect. Enter additional certificate configurations in this file.
4. Configure platform-admin XML files:
    a. Modify `deployments\cloudmatrix.ear\platform-admin.war \WEB-INF\jboss-web.xml` via a Chef script.
    b. Modify `deployments\cloudmatrix.ear\platform-admin.war\WEB-INF\picketlink.xml` by entering the IDP and SP URL. Enter additional certificate configurations in this file.
5. Deploy BouncyCastle to `${JBOSS_HOME}/modules/org/bouncycastle` (this is a PicketLink dependency).

Business Process Impact and Market Maker Login Impact (Database Update)

1. If the usual cm-mm-admin@gravitant.com login ID cannot be used for associating to a market maker, modify the `config-devo\SaaS\7.1.1\templates\jboss\Gravitant\bin\linux\setupDefaultSubscriptions.sh` loader.
2. Edit the contents and replace the user login email token from cm-mm-admin@gravitant.com to the new specific `user. java $JAVA_OPTS -Xms64m -Xmx256m -classpath "$CLASSPATH" com.gravitant.bms.client.dataload.GraLoadDefaultSubscriptions -groupName @CUSTOMER.NAME@ -groupId CM-@CM.MKTMKR.GROUPID@ -groupCode CM -template MarketMaker -domain @CUSTOMER.DOMAIN@ -userLogin cm-mm-admin@gravitant.com -orgRole "Market Maker Administrator,Market Maker Provisioning Administrator" -subLevel Premium -partyType marketmaker -customerTemplateName CustomerGcom`.
3. If an email id other than cm-mm-admin@gravitant.com is used in the system, please execute the scripts from `/Database-Upgrade/Planning/gravitant.com/7.1.5/ActivateSSO.sql` and replace the cm-mm-admin@gravitant.com with the custom market maker email id.
4. Set up the custom market maker or the cm-mm-admin@gravitant.com password with the identity provider, the login should match that entered in the cloudMatrix scripts above.

Platform Administration

The following points will require follow-up for Platform Administration integration:

- Create an application on the authentication provider to handle platform admin with its own set of users.
- The Platform Administration WAR file will require its own version of picketlink.xml. The URL will point to `https:{cloudmatrix_host}/admin`.
- Modify web.xml so that *.html pages require an authenticated user.

Keystore Creation and Deployment

Each server will require a unique keystore. Each keystore contains the digital certificate for the application that was created. This keystore can contain many certificates. This will be required if affiliates are implemented. Each affiliate will have its own authentication provider application with its own URL. Each application will provide its own certificate for authentication. To add more certificates, run step 3 using a different alias.

Please note that there are two alias variables when the keystore is created and ~ refers to `/root/`. (Since root may be creating this keystore.)

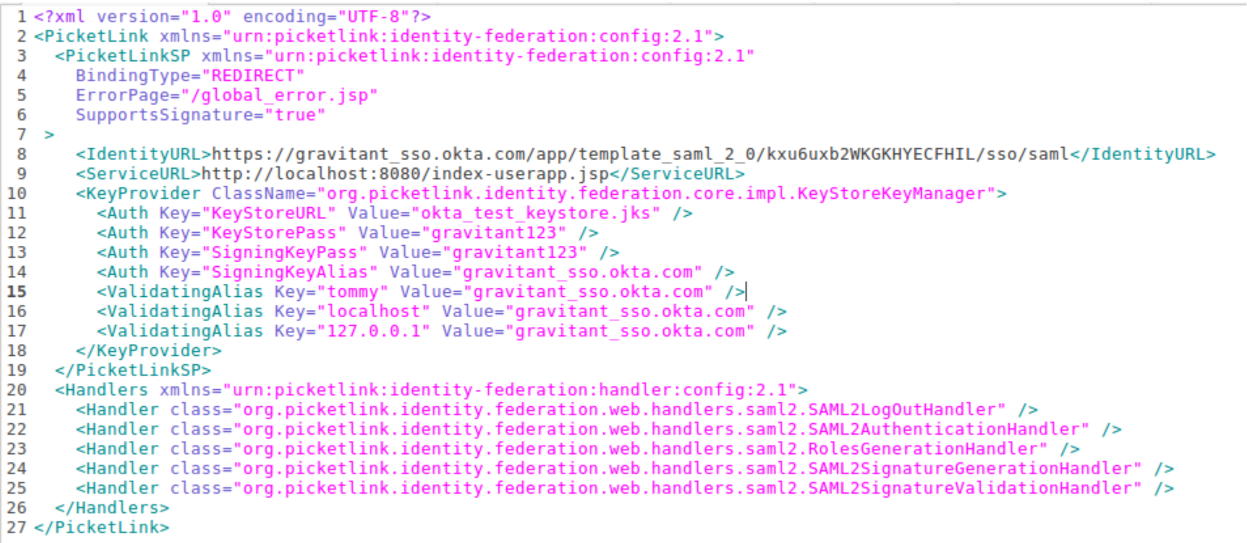
Use a Bash script to generate the certificate store.

1. Download the Okta certificate. This path will be needed for step 3. For now, download to ~. The path will be `~/okta.cert`.
2. Create the keystore using a similar command: `keytool -genkey -alias cloudmatrixsso -keyalg RSA -keystore ~/okta_keystore.jks -keysize 2048`
3. Add the Okta certificate to the keystore. This is the alias for the Okta certificate: `keytool -import -trustcacerts -alias gravitant_sso  -file ~/okta.cert -keystore ~/okta_keystore.jks`
4. To verify the keystore, run `keytool -list -v -keystore ~/okta_keystore.jks` and verify that the Okta keystore has been added.
5. Once the keystore is created, it will need to be placed in the following folder: `${JBOSS_HOME}/modules/com/gravitant/resources/main/`

SAML Certificate Configuration Steps

PicketLink XML Example

Below is an example of a PicketLink XML which includes SAML authentication.



```
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <PicketLink xmlns="urn:picketlink:identity-federation:config:2.1">
 3   <PicketLinkSP xmlns="urn:picketlink:identity-federation:config:2.1"
 4     BindingType="REDIRECT"
 5     ErrorPage="/global_error.jsp"
 6     SupportsSignature="true"
 7   >
 8     <IdentityURL>https://gravitant_sso.okta.com/app/template_saml_2_0/kxu6uxb2WKGKHYECFHIL/sso/saml</IdentityURL>
 9     <ServiceURL>http://localhost:8080/index-userapp.jsp</ServiceURL>
10     <KeyProvider ClassName="org.picketlink.identity.federation.core.impl.KeyStoreKeyManager">
11       <Auth Key="KeyStoreURL" Value="okta_test_keystore.jks" />
12       <Auth Key="KeyStorePass" Value="gravitant123" />
13       <Auth Key="SigningKeyPass" Value="gravitant123" />
14       <Auth Key="SigningKeyAlias" Value="gravitant_sso.okta.com" />
15       <ValidatingAlias Key="tommy" Value="gravitant_sso.okta.com" />
16       <ValidatingAlias Key="localhost" Value="gravitant_sso.okta.com" />
17       <ValidatingAlias Key="127.0.0.1" Value="gravitant_sso.okta.com" />
18     </KeyProvider>
19   </PicketLinkSP>
20   <Handlers xmlns="urn:picketlink:identity-federation:handler:config:2.1">
21     <Handler class="org.picketlink.identity.federation.web.handlers.saml2.SAML2LogOutHandler" />
22     <Handler class="org.picketlink.identity.federation.web.handlers.saml2.SAML2AuthenticationHandler" />
23     <Handler class="org.picketlink.identity.federation.web.handlers.saml2.RolesGenerationHandler" />
24     <Handler class="org.picketlink.identity.federation.web.handlers.saml2.SAML2SignatureGenerationHandler" />
25     <Handler class="org.picketlink.identity.federation.web.handlers.saml2.SAML2SignatureValidationHandler" />
26   </Handlers>
27 </PicketLink>
```

PicketLink XML Notes:

- Line 6 will need to be set to true if we are using signed keystores for authentication. Otherwise, it should be set to false.
- Line 8 will be provided by the authentication provider when the application is created.
- Line 9 will need to point to the URL of application (affiliates have different URLs).
- Lines 14 - 17 will need to refer to the alias of the keystore.
- Line 15 will need to be added when affiliates are used. Otherwise it is not needed.

Activate Filter

Uncomment out the section in cloudMatrixWeb.war's web.xml to activate this filter, to verify that the user is active and their subscription is valid:

```
<filter>

    <filter-name>SamlResponseFilter</filter-name>

    <filter-class>com.gravitant.cloud.common.servlet.SamlResponseFilter</filter-class>

</filter>

<filter-mapping>
```

```
        <filter-name>SamlResponseFilter</filter-name>

        <url-pattern>/*</url-pattern>

</filter-mapping>
```

SAML Based Affiliate Setup

When the user is authenticated, login servlet redirects the user to the requested URL. Using SSO, an application will need to be created for each affiliate that redirects to each URL. There are two use cases to consider:

- The user logged into the authentication provider before selecting a link to the affiliate URL.
- The user is not logged in and selects the affiliate URL. After authentication, the identity provider redirects the user to the affiliate’s URL.

Affiliates have their own unique URL from the identity provider, each of which has a unique authentication URL from the identity provider, which would configure the service URL, the post back URL and the destination to the affiliate URL.

Take the final IDP authentication URL for this affiliate from the IDP and add the Affiliate Specific Application URL to idp.registry located under `jboss-modules/com/gravitant/resources/main` folder.

Billing Automation URL Changes

Change billing automation from `admin/api/bills/automation` to `/admin/kettle/process/automation`. This is an insecure URL and must be protected from nGinix - outside.

Note on White Labelling on SAML

Typically whitelabelling customizations include developing a custom login page for affiliates, since the login functionality is through a SAML provider. Please provide a link in this page to point to the cloudmatrix-service url index-userapp.jsp page.

Appendices: XML Configuration Examples

Here are some examples of the referenced .xml files. Please obtain the latest version of these files from the repository.

picketlink.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<PicketLink xmlns="urn:picketlink:identity-federation:config:2.1">

     <PicketLinkSP xmlns="urn:picketlink:identity-
federation:config:2.1"

           BindingType="REDIRECT"

           ErrorPage="/global_error.jsp"

           SupportsSignature="true"

     >

<IdentityURL>https://gravitant_sso.okta.com/app/template_saml_2_0/kxu6
uxb2WKGKHYECFHIL/sso/saml</IdentityURL>

     <ServiceURL>http://localhost:8080/index-userapp.jsp</ServiceURL>

     <KeyProvider
     ClassName="org.picketlink.identity.federation.core.impl.KeyStoreK
     eyManager">

           <Auth Key="KeyStoreURL" Value="okta_test_keystore.jks" />

           <Auth Key="KeyStorePass" Value="gravitant123" />

           <Auth Key="SigningKeyPass" Value="gravitant123" />

           <Auth Key="SigningKeyAlias" Value="gravitant_sso.okta.com"
           />

           <ValidatingAlias Key="tommy" Value="gravitant_sso.okta.com"
           />

           <ValidatingAlias Key="localhost"
           Value="gravitant_sso.okta.com" />

           <ValidatingAlias Key="127.0.0.1"
           Value="gravitant_sso.okta.com" />

     </KeyProvider>
```

```
</PicketLinkSP>

     <Handlers xmlns="urn:picketlink:identity-
     federation:handler:config:2.1">

          <Handler
          class="org.picketlink.identity.federation.web.handlers.saml
          2.SAML2LogOutHandler" />

          <Handler
          class="org.picketlink.identity.federation.web.handlers.saml
          2.SAML2AuthenticationHandler" />

          <Handler
          class="org.picketlink.identity.federation.web.handlers.saml
          2.RolesGenerationHandler" />

          <Handler
          class="org.picketlink.identity.federation.web.handlers.saml
          2.SAML2SignatureGenerationHandler" />

          <Handler
          class="org.picketlink.identity.federation.web.handlers.saml
          2.SAML2SignatureValidationHandler" />

     </Handlers>

</PicketLink>
```

jboss-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<jboss-web>

     <security-domain>sp</security-domain>

     <valve>

          <class-
          name>org.picketlink.identity.federation.bindings.tomcat.sp.
          ServiceProviderAuthenticator</class-name>

     </valve>

</jboss-web>
```

jboss-deployment-structure.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<jboss-deployment-structure xmlns="urn:jboss:deployment-
structure:1.2">

      <deployment>

            <exclude-subsystems>

                  <subsystem name="jaxrs" />

            </exclude-subsystems>

            <dependencies>

                  <module name="javax.faces.api" slot="main"
                  export="true"/>

                  <module name="com.sun.jsf-impl" slot="main"
                  export="true"/>

                  <module name="org.hibernate.validator" export="true"
                  />

                  <module name="javax.validation.api" export="true" />

                  <module name="org.codehaus.jettison" export="true" />

                  <module name="org.apache.log4j" export="true" />

                  <module name="org.apache.commons.beanutils"
                  slot="main" export="true" />

                  <module name="org.apache.commons.collections"
                  slot="main" export="true" />

                  <module name="org.apache.commons.logging" slot="main"
                  export="true" />

                  <module name="org.apache.httpcomponents" slot="main"
                  export="true" />

                  <module name="org.apache.commons.lang" export="true"
                  />

                  <module name="com.google.guava" slot="main"
                  export="true" />

                  <module name="javax.inject.api" slot="main"
                  export="true" />
                  <module name="org.joda.time" slot="main" export="true"
                  />
                  <module name="net.sf.ehcache" slot="main"
                  export="true" />
                  <module name="org.jgroups" slot="main" export="true"
                  />
                  <module name="com.gravitant" slot="main" export="true"
                  />
                  <module name="com.gravitant.resources" slot="main"
                  export="true" />
                  <module name="com.gravitant.cloud.common.idp"
                  slot="main" export="true" />
                  <module name="org.jboss.as.web" slot="main"
                  export="true">
                      <imports>
                             <include path="/org/**" />
                             <exclude path="/META-INF/**" />
                      </imports>
                  </module>
                  <module name="org.jboss.remote-naming" slot="main"
                  export="true" />
                  <module name="org.hornetq" slot="main" export="true"
                  />
                  <module name="org.picketlink" slot="main"
                  export="true"  />
              </dependencies>
       </deployment>
   </jboss-deployment-structure>
```

Legal Disclaimer

Information, data and drawings embodied in this guide are strictly confidential and are supplied on the understanding that they will be held confidentially and not disclosed to third parties without the prior written consent of Gravitant.

This document and the related software product are the exclusive property of Gravitant, Inc. (Gravitant®) and may be used only by users properly licensed to use the software product and only according to the terms of that license. Use of the software product constitutes acceptance of those license terms.

Gravitant, Inc. and its affiliated companies (collectively, "Gravitant") make no warranties about the information in this document. Under no circumstances shall Gravitant be liable for costs arising from the procurement of substitute products or services, lost profits, lost savings, loss of information or data, or from any other special, indirect, consequential, or incidental damages, that are the result of its products not being used in accordance with this document.

Trademarks, service marks, and logos appearing in this document are the property of Gravitant or the party that provided the trademarks, service marks, and logos to Gravitant. Gravitant and any party that provided trademarks, service marks, and logos to Gravitant retain all rights with respect to any of their respective trademarks, service marks, and logos appearing in this document. Any rights not expressly granted herein are reserved.

The user (customer or client in any role) acknowledges by receipt and use of this document that it is confidential and proprietary information of Gravitant. Please ensure that this document is not disclosed to any persons other than your employees with a bona fide need to know.

CORPORATE OFFICE

Gravitant, Inc.

Synergy Plaza - North

11940 Jollyville Rd. Suite 325-N

Austin, TX 78759

866-207-8877

This document explains how to configure an Agent on SFB and configure a service in Gravitant cloudMatrix™.

Cloud providers are third parties which provide different cloud based products and services. They update their products, services, and pricing from time to time. Gravitant cloudMatrix, being web based portal, updates itself accordingly. The offline documentation for cloudMatrix (such as PDF files) may contain broken hyperlinks due to the dynamic nature of services of different cloud providers. While Gravitant makes every effort to keep this document up to date with the latest developments, it makes no guarantee that this document is fully accurate every time due to such changes.

This document is up to date until the date of publication on its cover page. Please make sure that you regularly check for the newer version of this document, if any, from Gravitant and use the most recent version for your work.

The screenshots in this guide are taken with full administrative rights for the sake of completeness of procedures. Depending on your user role and your organization’s privileges, the screens that actually display for you may be different.

This guide is intended for cloudMatrix users, who need to perform their user role specific tasks for building, deploying, and configuring the Agent on SFB and configuring a service on cloudMatrix. It assumes that its users are well acquainted with Internet browsing, terms and concepts in cloud computing, and the workflow for using cloudMatrix.

Since cloudMatrix is designed to deliver high usability and high flexibility in performing different tasks, there can be multiple ways to perform a given task in cloudMatrix. However, keeping in mind less experienced users as well, this document is designed to provide a structured approach for performing different tasks. Experienced users can always explore different ways just by the intuitive nature of the cloudMatrix user interface.