How to Setup MS AD FS 3.0 as Brokered Identity Provider in Keycloak

Thursday, March 23 2017, posted by Hynek Mlnařík

This document guides you through initial setup of Microsoft Active Directory Federation Services 3.0 as a brokered identity provider Keycloak.

Prerequisites

- Two server hosts:
  - Microsoft Windows Server 2012 with Active Directory Federation Services (AD FS) installed. The AD domain will be named **DOMAIN.NAME** in this post.
  - Keycloak server. This can be generally placed anywhere but here it is expected to be running on separate host
- DNS setup:
  - The Windows host name will be **fs.domain.name** in this post
  - The Keycloak host name will be **kc.domain.name** in this post

Setup Keycloak Server

Keycloak server has configured for SSL/TLS transport - this is mandatory for AD FS to communicate with it. This comprises two steps:

- Setup keycloak for incoming HTTPS connections - steps are provided in Server Installation guide (https://www.keycloak.org/docs/latest/server_installation/index.html#enabling-ssl-https-for-the-keycloak-server).
- Export AD FS certificate into a Java truststore to enable outgoing HTTPS connections:
  - In the AD FS management console, go to *Service* → *Certificates* node in the tree and export the *Service communications* certificate.
  - Import the certificate into a Java truststore (JKS format) using Java keytool utility.
  - Setup the truststore in Keycloak as described in Server Installation guide (https://www.keycloak.org/docs/latest/server_installation/index.html#_truststore).

Setup Identity Provider in Keycloak

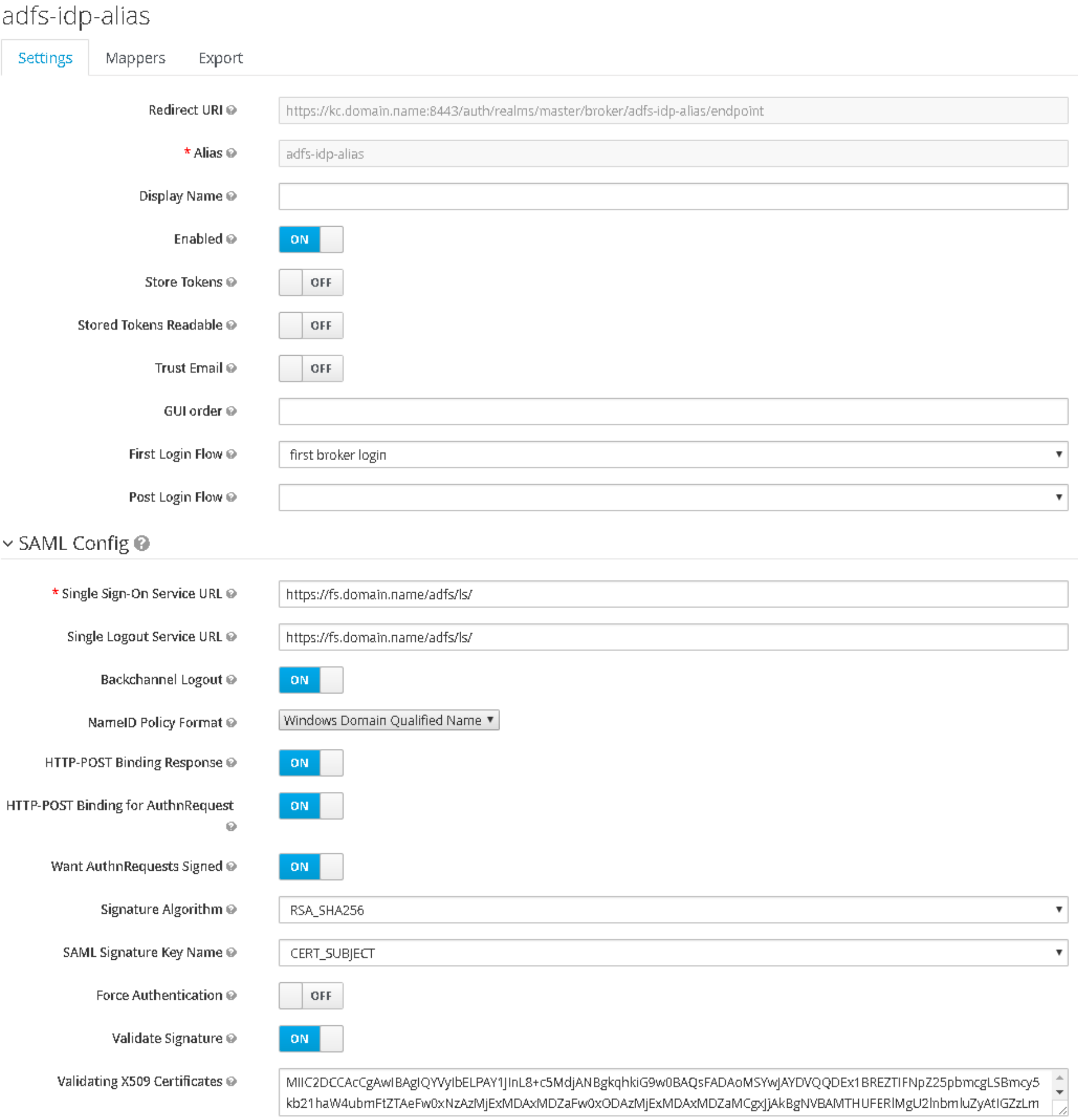
Setup Basic Properties of Brokered Identity Provider

In the Identity Providers, create a new SAML v2.0 identity provider. In this post, the identity provider will be known under alias **adfs-idp-alias**.

Now scroll to the bottom and enter the AD FS descriptor URL into *Import from URL* field. For AD FS 3.0, this URL is **https://fs.domain.name/FederationMetadata/2007-06/FederationMetadata.xml**. Once you click “Import”, check the settings. Usually, you would at least enable *Validate signature* option.

If the authentication requests sent to the AD FS instance are expected to be signed, which is also usually the case, you have to enable *Want AuthnRequests Signed* option. Importantly, then the *SAML Signature Key Name* field that shows after enabling the *Want AuthnRequests Signed* option has to be set to CERT_SUBJECT as AD FS expects the signing key name hint to be the subject of the signing certificate.

The AD FS will be set up in the next step to respond with name ID in Windows Domain Qualified Name format, hence set the *NameID Policy Format* field accordingly.



adfs-idp-alias

Redirect URI	https://kc.domain.name:8443/auth/realms/master/broker/adfs-idp-alias/endpoint
* Alias	adfs-idp-alias
Display Name	
Enabled	ON
Store Tokens	OFF
Stored Tokens Readable	OFF
Trust Email	OFF
GUI order	
First Login Flow	first broker login
Post Login Flow	

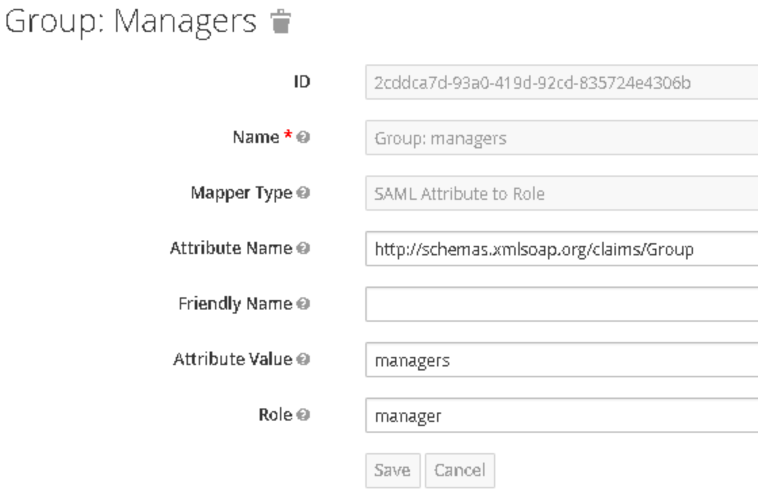
SAML Config

* Single Sign-On Service URL	https://fs.domain.name/adfs/ls/
Single Logout Service URL	https://fs.domain.name/adfs/ls/
Backchannel Logout	ON
NameID Policy Format	Windows Domain Qualified Name
HTTP-POST Binding Response	ON
HTTP-POST Binding for AuthnRequest	ON
Want AuthnRequests Signed	ON
Signature Algorithm	RSA_SHA256
SAML Signature Key Name	CERT_SUBJECT
Force Authentication	OFF
Validate Signature	ON
Validating X509 Certificates	MIIC2DCCAcCgAwIBAgIQYVyIbELPAY1JInL8+c5MdjANBgkqhkiG9w0BAQsFADAoMSYwJAYDVQQDEx1BREZTIFNpZ25pbmcgLSBmcy5 kb21haW4ubmFtZTAeFw0xNzAzMjExMDAxMDZaFw0xODAzMjExMDAxMDZaMCgxJjAkBgNVBAMTHUFERlMgU2lnbmluZyAtIGZzLm

Setup Mappers

In the steps setting AD FS below, AD FS will be set up to send email and group information in SAML assertion. To transform these details from SAML document issued by AD FS to Keycloak user store, we'll need to set up two corresponding mappers in the Mappers tab of Identity Provider:

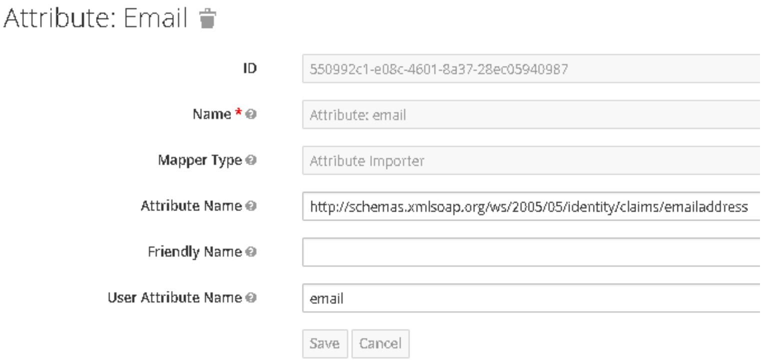
- Mapper named *Group: managers* will be of type *SAML Attribute to Role*, and will map attribute named *http://schemas.xmlsoap.org/claims/Group*, if that has attribute value *managers,* to role *manager*.



Group: Managers

ID	2cddca7d-93a0-419d-92cd-835724e4306b
Name *	Group: managers
Mapper Type	SAML Attribute to Role
Attribute Name	http://schemas.xmlsoap.org/claims/Group
Friendly Name	
Attribute Value	managers
Role	manager

- Mapper named *Attribute: email* will be of type *Attribute Importer*, and will map attribute named *http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress* into user attribute named *email*.



Attribute: Email

ID	550992c1-e08c-4601-8a37-28ec05940987
Name *	Attribute: email
Mapper Type	Attribute Importer
Attribute Name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
Friendly Name	
User Attribute Name	email

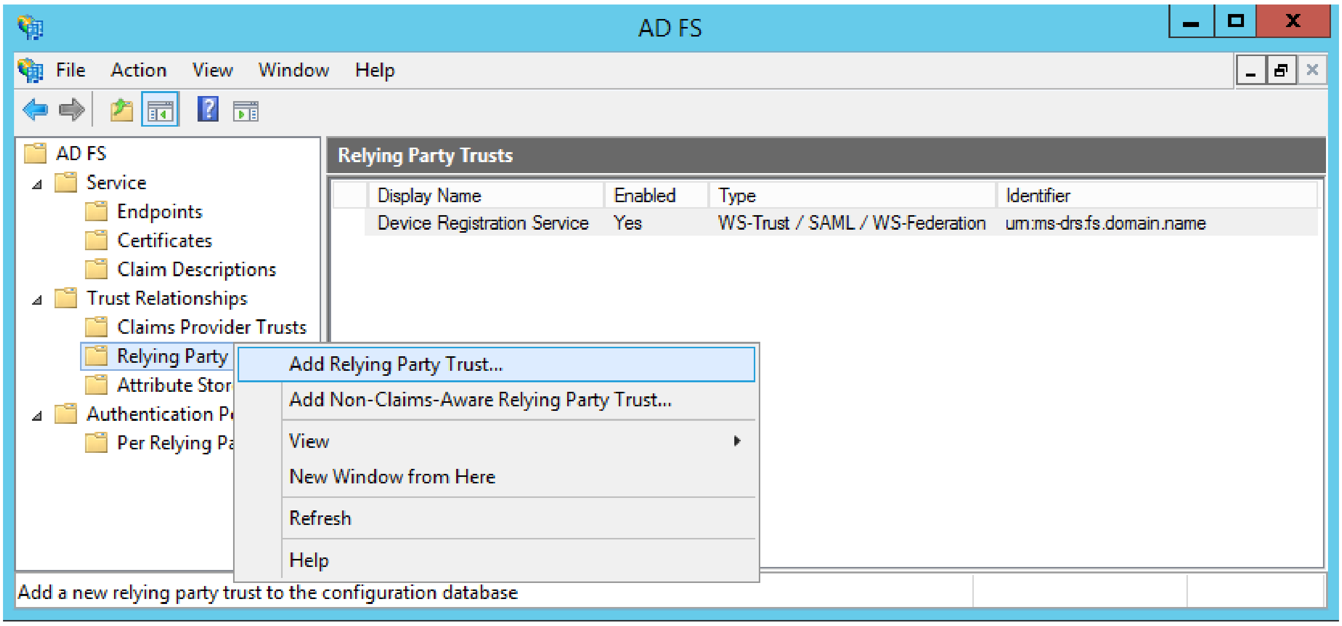
Obtain information for the AD FS configuration

Now we determine SAML service provider descriptor URI that will be used in AD FS setup from the *Redirect URI* field in the identity provider detail by adding "/descriptor" to the URI in this field. The URI will be similar to **https://kc.domain.name:8443/auth/realms/master/broker/adfs-idp-alias/endpoint/descriptor**. You can check whether you got the URI right by entering the URI into the browser - you should receive a SAML service provider XML descriptor.

Setup Relying Party Trust in AD FS

Setup Relying Party

In AD FS Management console, right-click T*rust relationships* → *Relying Party Trusts* and select *Add Relying Party Trust* from the menu:



Display Name	Enabled	Type	Identifier
Device Registration Service	Yes	WS-Trust / SAML / WS-Federation	urn:ms-drs:fs.domain.name

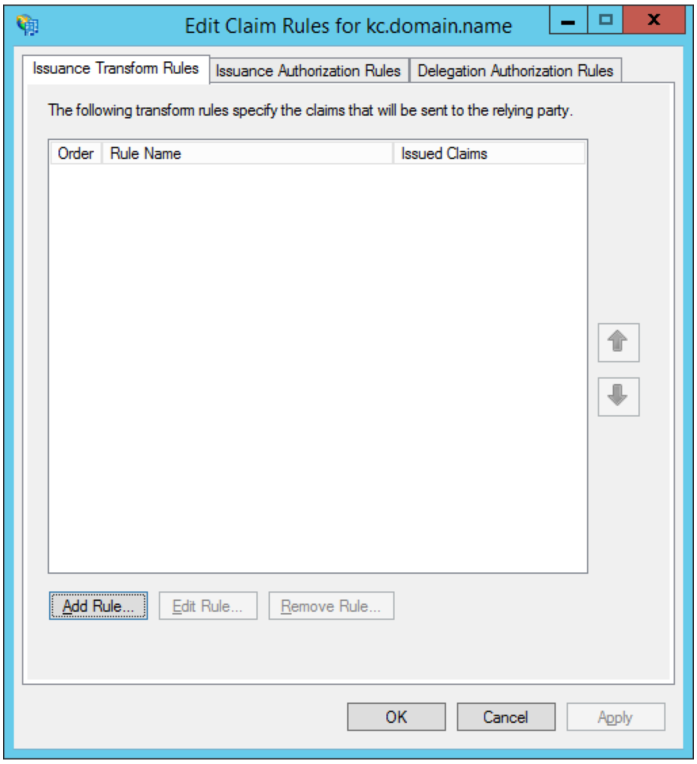
Add a new relying party trust to the configuration database

At the beginning of the wizard, enter the SAML descriptor URL obtained in the previous step into the *Federation metadata address* field, and let AD FS import the settings. Proceed with the wizard, and adjust the settings where appropriate. Here we use only the default settings. Note that you will need to edit the claim rules so when asked to do so at the last page of the wizard, you can leave the checkbox checked on.

Setup Claim Mapping

Now the SAML protocol would proceed correctly, AD FS would be able to correctly authenticate the users according to requests from Keycloak, but the requested name ID format is not yet recognized and SAML response would not contain any additional information like e-mail. It is hence necessary to map claims from AD user details into SAML document.

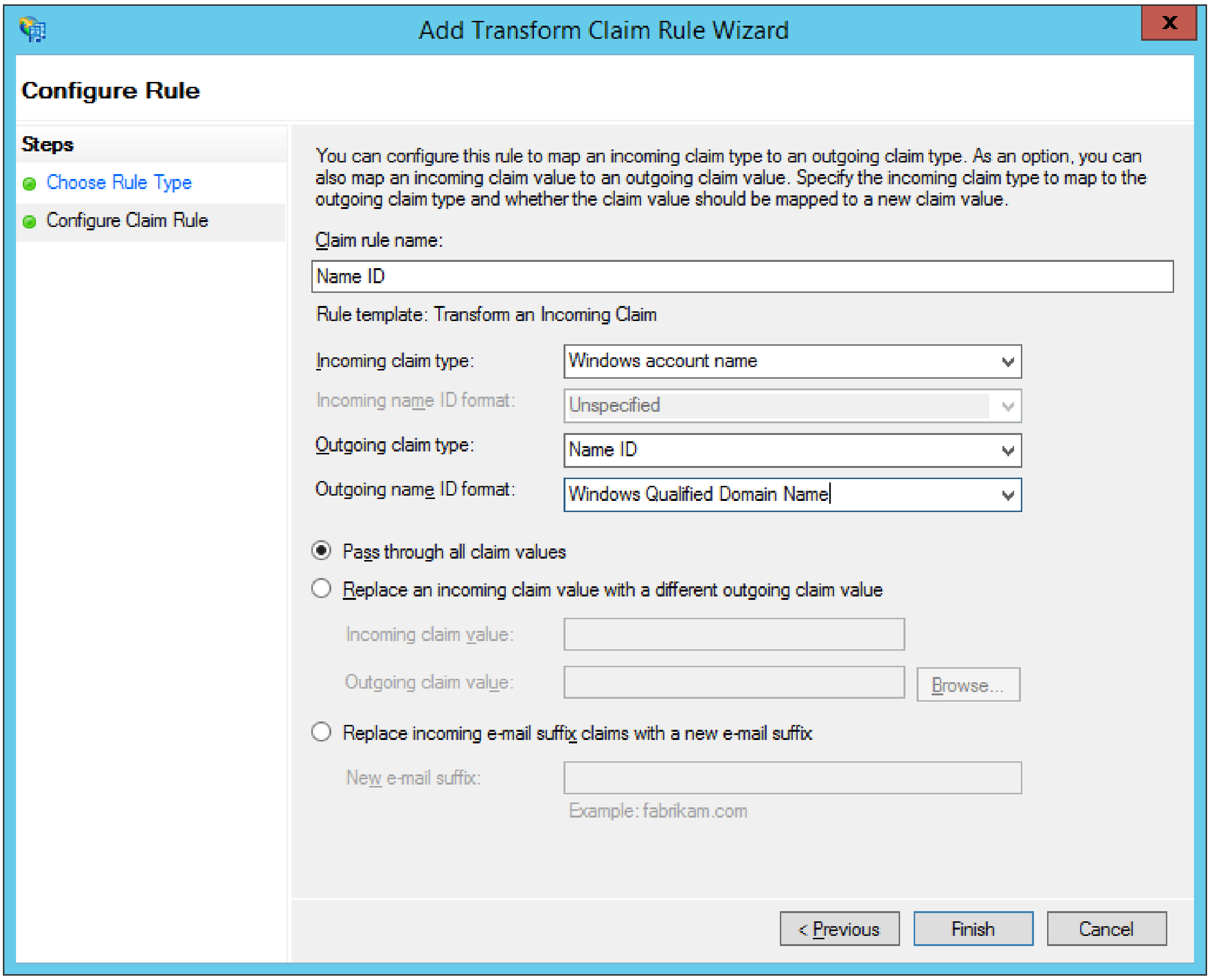
We will set up three rules: one for mapping user ID, second for mapping standard user attributes, and third for a user group. All start by clicking the *Add Rule* button in the *Edit Claim Rules for kc.domain.name* window:


Edit Claim Rules for kc.domain.name

The following transform rules specify the claims that will be sent to the relying party.

Order	Rule Name	Issued Claims

The first rule will map user ID in Windows Qualified Domain name to the SAML response. In the *Add Transform Claim Rule* window, select *Transform an incoming claim* rule type:



Add Transform Claim Rule Wizard

Configure Rule

Steps

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name: Name ID

Rule template: Transform an Incoming Claim

Incoming claim type: Windows account name

Incoming name ID format: Unspecified

Outgoing claim type: Name ID

Outgoing name ID format: Windows Qualified Domain Name

● Pass through all claim values

○ Replace an incoming claim value with a different outgoing claim value

Incoming claim value:

Outgoing claim value:

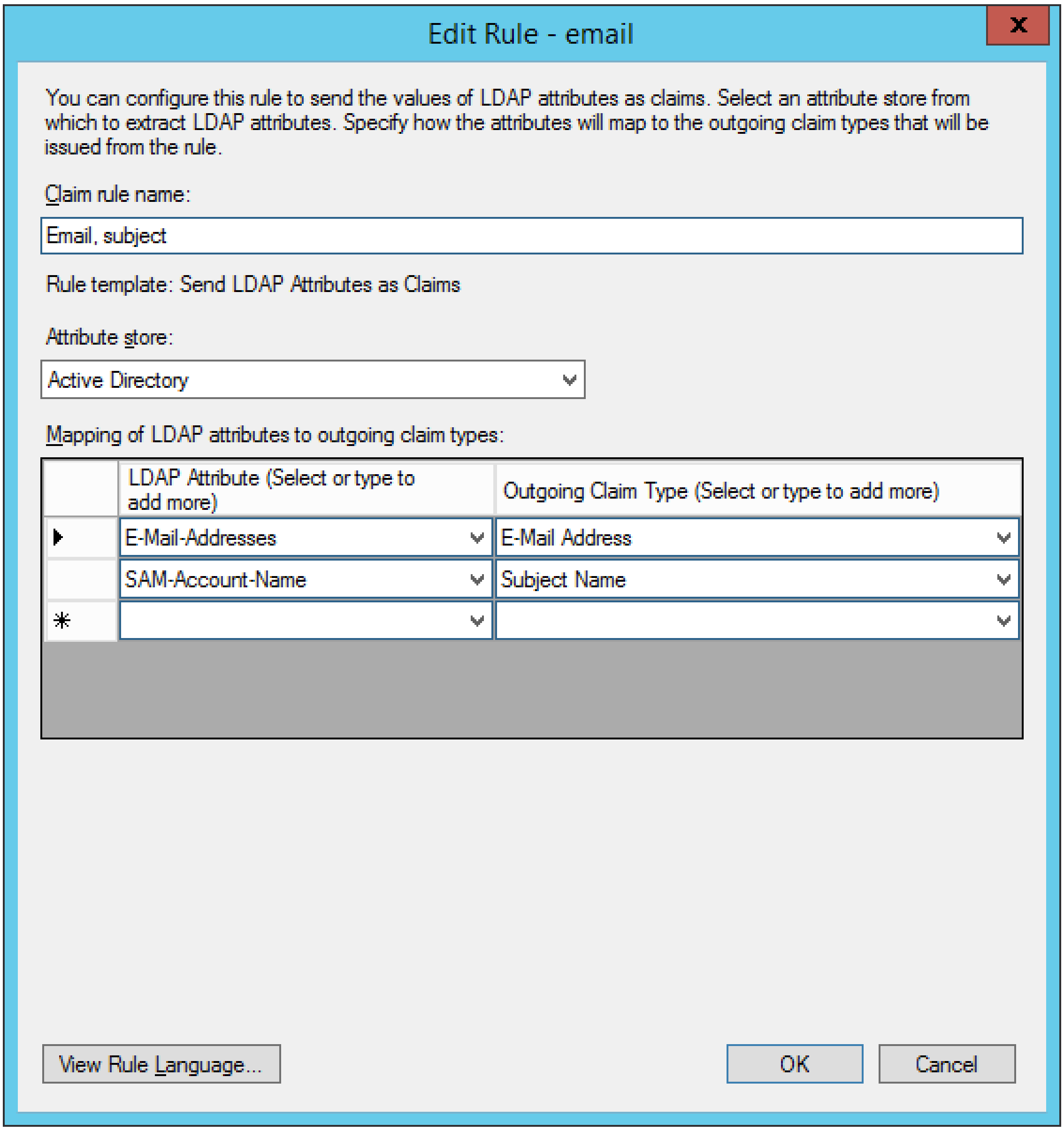
○ Replace incoming e-mail suffix claims with a new e-mail suffix

New e-mail suffix:

Example: fabrikam.com

The example above targets windows account name ID format. Other name ID formats are supported but out of scope of this post. See e.g. this blog (https://blogs.msdn.microsoft.com/card/2010/02/17/name-identifiers-in-saml-assertions/) on how to setup name IDs for persistent and transient formats.

The second rule will map user e-mail to the SAML response. In the *Add Transform Claim Rule* window, select *Send LDAP attributes as Claims* rule type. You can add other attributes as needed:



You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Email, subject

Rule template: Send LDAP Attributes as Claims

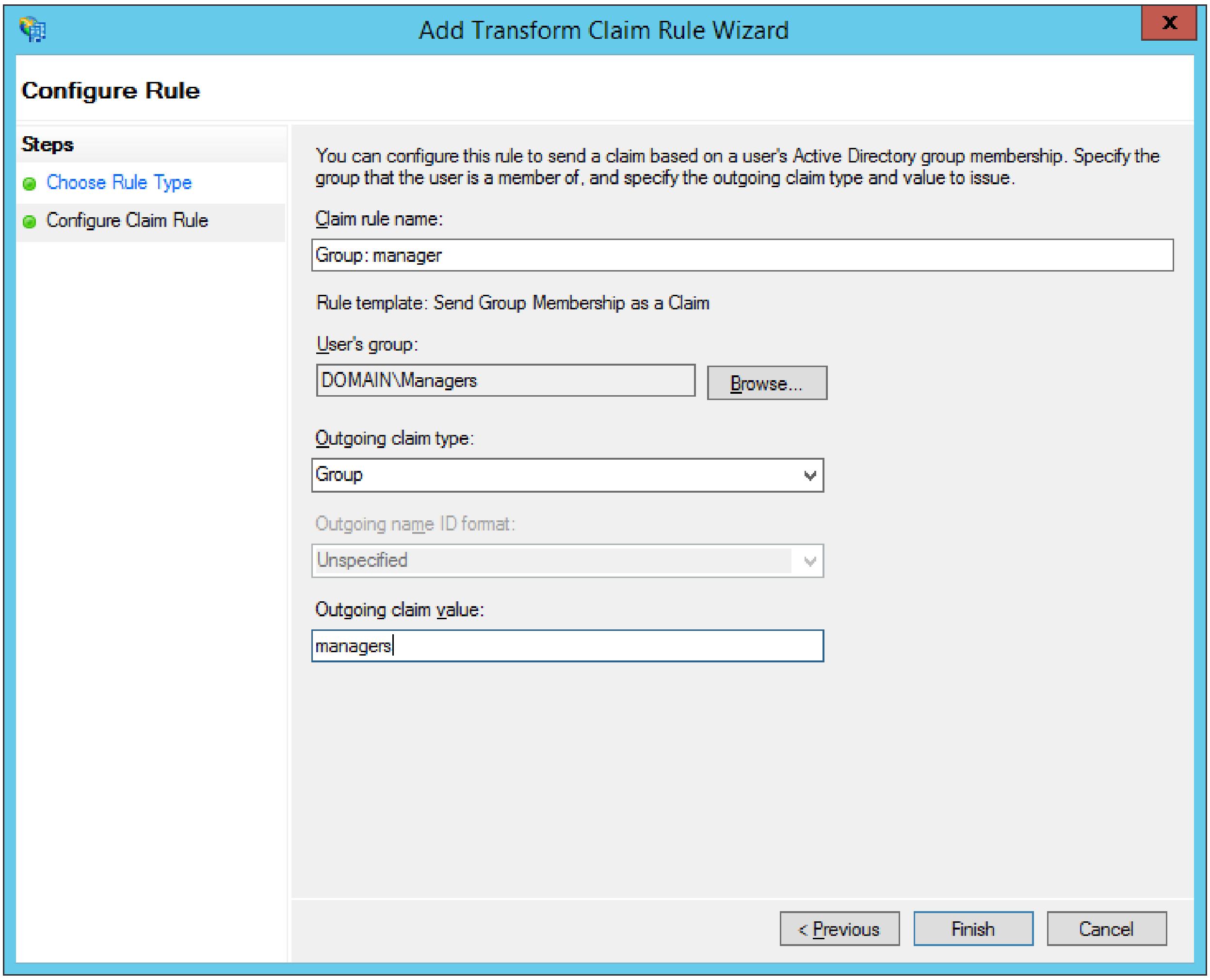
Attribute store:

Active Directory

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	E-Mail-Addresses	E-Mail Address
	SAM-Account-Name	Subject Name
*		

The third rule would send a group name if the user is member of a named group. Start again in the *Add Transform Claim Rule* window, and select *Send Group Membership as a Claim* rule type. Then enter the requested values in the field:



Add Transform Claim Rule Wizard

Configure Rule

You can configure this rule to send a claim based on a user's Active Directory group membership. Specify the group that the user is a member of, and specify the outgoing claim type and value to issue.

Claim rule name: Group: manager

Rule template: Send Group Membership as a Claim

User's group: DOMAIN\Managers

Outgoing claim type: Group

Outgoing name ID format: Unspecified

Outgoing claim value: managers

This setup would send an attribute named *Group* in the SAML assertion with value *managers* if the authenticated user is member of the *DOMAIN\Managers* group.

Troubleshooting

As a first-hand tool, you should check SAML messages sent back and forth between Keycloak and AD FS in your browser. The SAML decoders are available as browser extensions (e.g. SAML Tracer for Firefox, SAML Chrome Panel for Chrome). From the captured communication, you might see error status codes as well as the actual attribute names and values in SAML assertion necessary for setting up mappers. For example, if name ID format is not recognized, AD FS would return a SAML response containing *urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy* status code.

As a second resort, check the logs. For AD FS, the logs are available in the *Event viewer* under *Applications and Services Logs → AD FS → Admin*. In Keycloak, you can enable tracing of the SAML processing by connecting to the running Keycloak instance via jboss-cli.sh and entering the following commands:

```
/subsystem=logging/logger=org.keycloak.saml:add(level=DEBUG)
```

```
/subsystem=logging/logger=org.keycloak.broker.saml:add(level=DEBUG)
```

Then you will be able to find the SAML messages and broker-related SAML processing messages in the Keycloak server log.

Common issues

Q: I cannot log out! When I click logout in my app, it seems I'm logged out from Keycloak but when I return to the app, AD FS login form never displays and I'm redirected back authenticated as the same user as previously!
A: Don't panic. This is not a Keycloak issue, rather AD FS settings of authentication policy. Try disabling Windows Authentication (https://blogs.msdn.microsoft.com/josrod/2014/10/15/enabled-forms-based-authentication-in-adfs-3-0/) before reporting an issue.

Q: While using AD FS in Windows 2016, the following error appeared in Keycloak log after importing the descriptor from URL: R*ESTEASY002010: Failed to execute: javax.ws.rs.NotFoundException: RESTEASY003210: Could not find resource for full path: https://kc.domain.name/auth/realms/master/broker/adfs-idp-alias/endpoint/descriptor/FederationMetadata/2007-06/FederationMetadata.xml*. Does it cause any harm?
A: It is harmless. It seems that Windows 2016 version first checks for AD FS-like descriptor URL by adding *FederationMetadata/2007-06/FederationMetadata.xml* to the entered URL. Such resource does not exist in Keycloak, so it reports error. AD FS however seems to import using the entered URL when this happens. Please see also the original email discussion (http://lists.jboss.org/pipermail/keycloak-user/2017-March/010138.html) on this issue.

Conclusion

If you get stuck, do not hesitate to write a question to **keycloak-user** mailing list.

As there is always room for improvement, if you find any issue or have any suggestion on this text, feel free to leave a comment!

(http://www.redhat.com/)