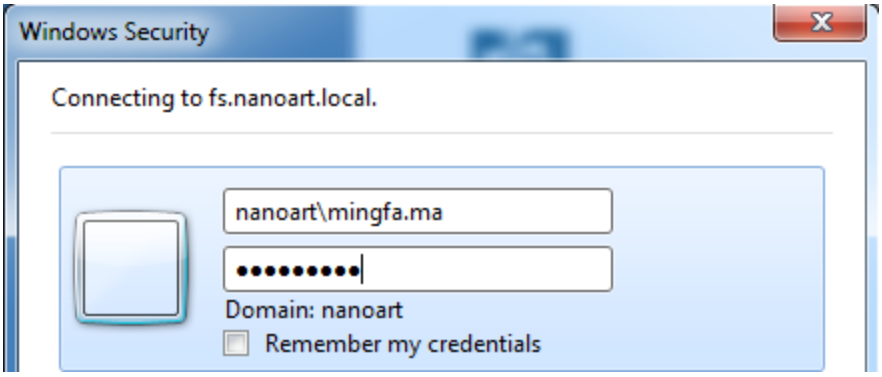
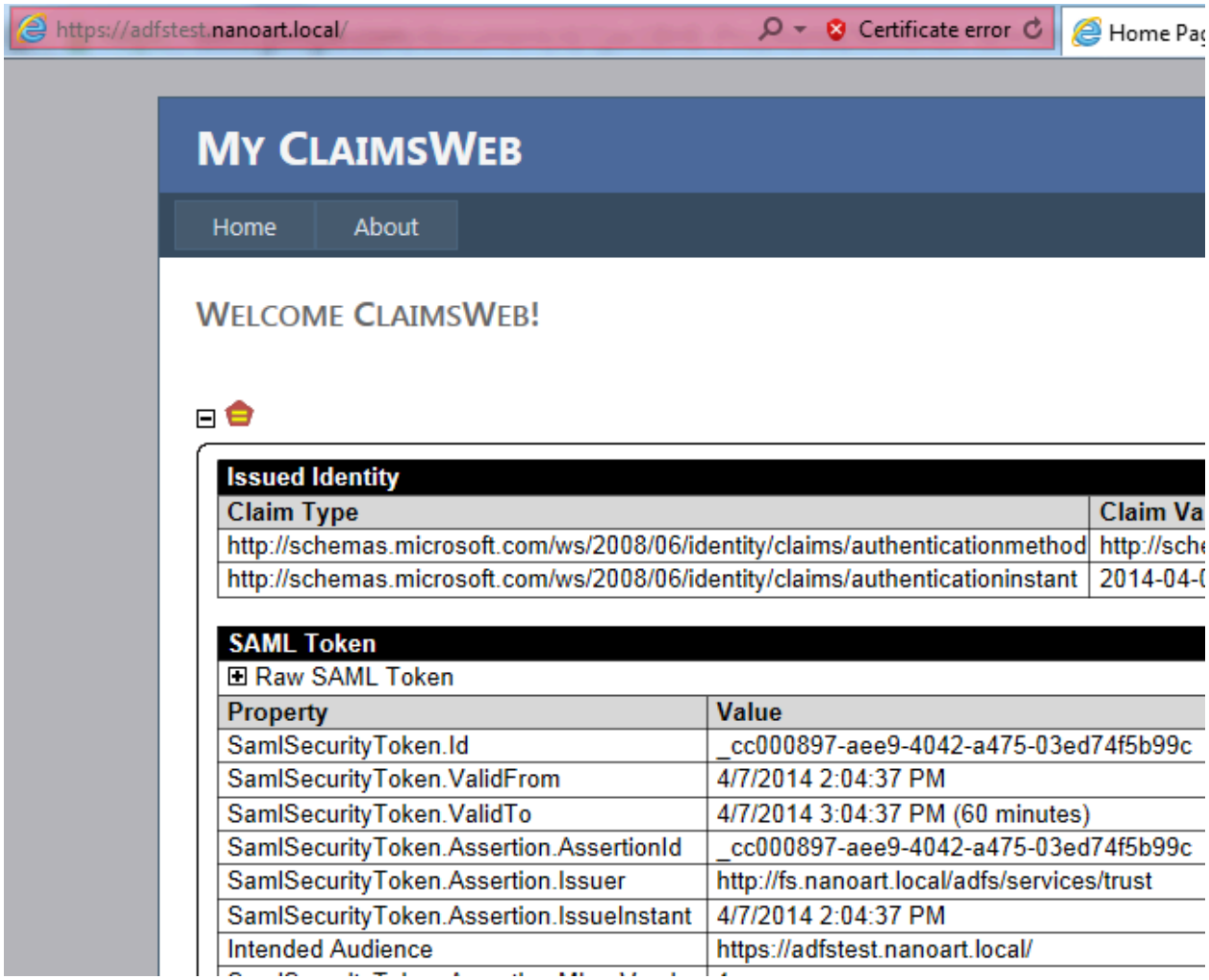
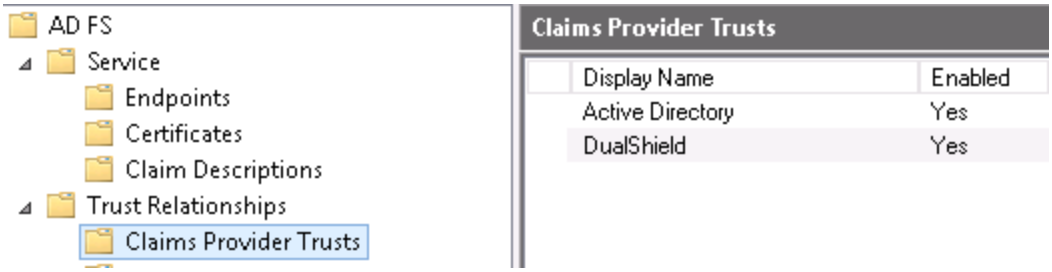
Claims-Aware Applications, such as SharePoint, Lync 2013(passive authentication), are generally authenticated through Active Directory with ADFS.



After authentication (here is Windows integrated authentication), you can get to see the web content



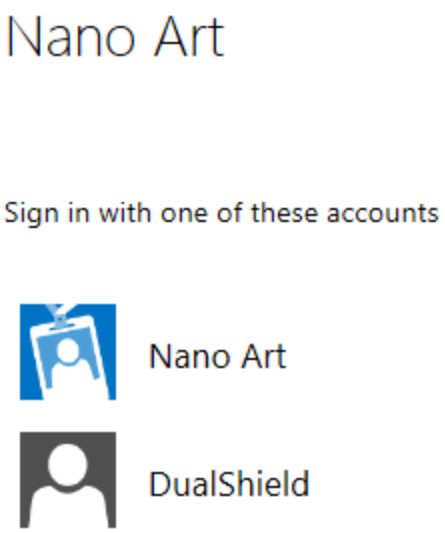
You can extend it to use DualShield to do the multi-factor authentication by adding DualShield as a **Claims Provider Trust**.



Now if you access the web application in browser

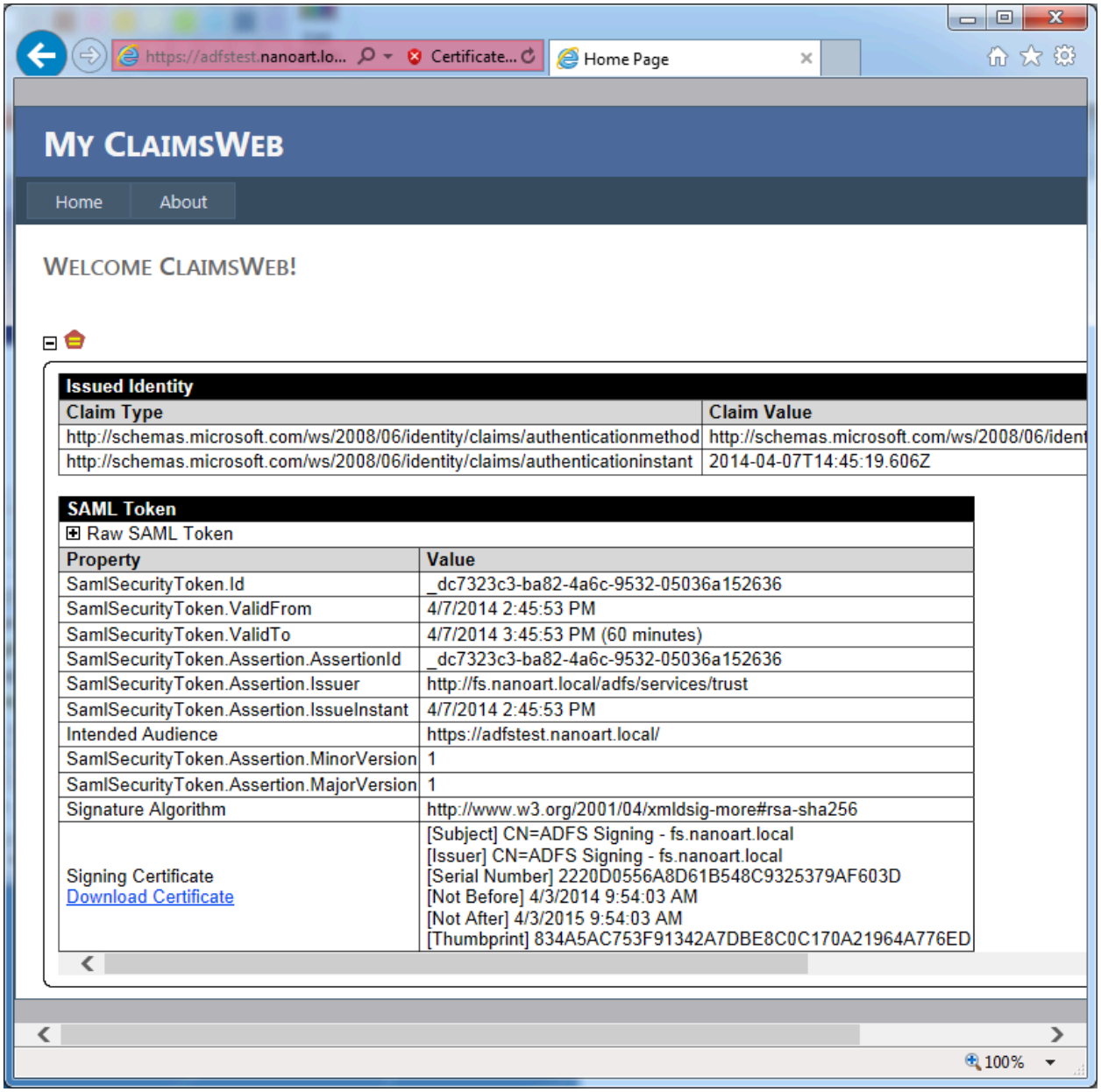
https://adfstest.nanoart.local

You will see the claim provider (IDP) selection (**Nano Art** in this sample is the built-in Active Directory).



Select DualShield, it redirects to DualShield SSO authentication page.

Once you finish the authentication, you will see the final web content.

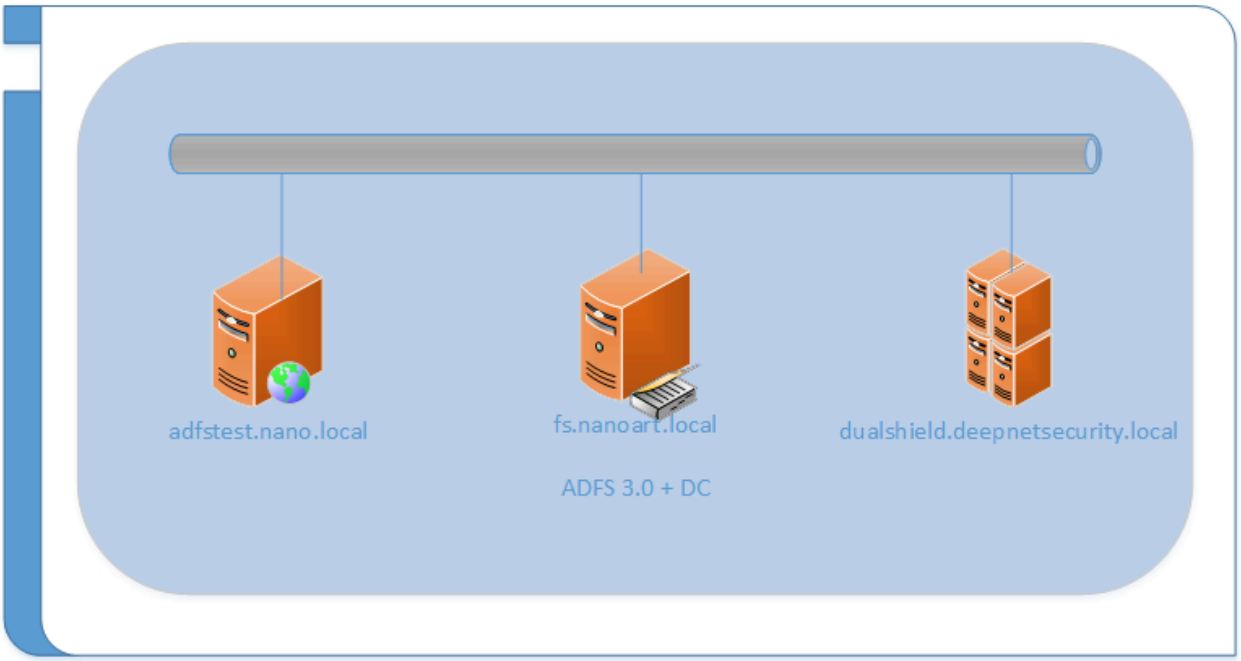


Now you may wonder how it works, how you can set up a test lab?

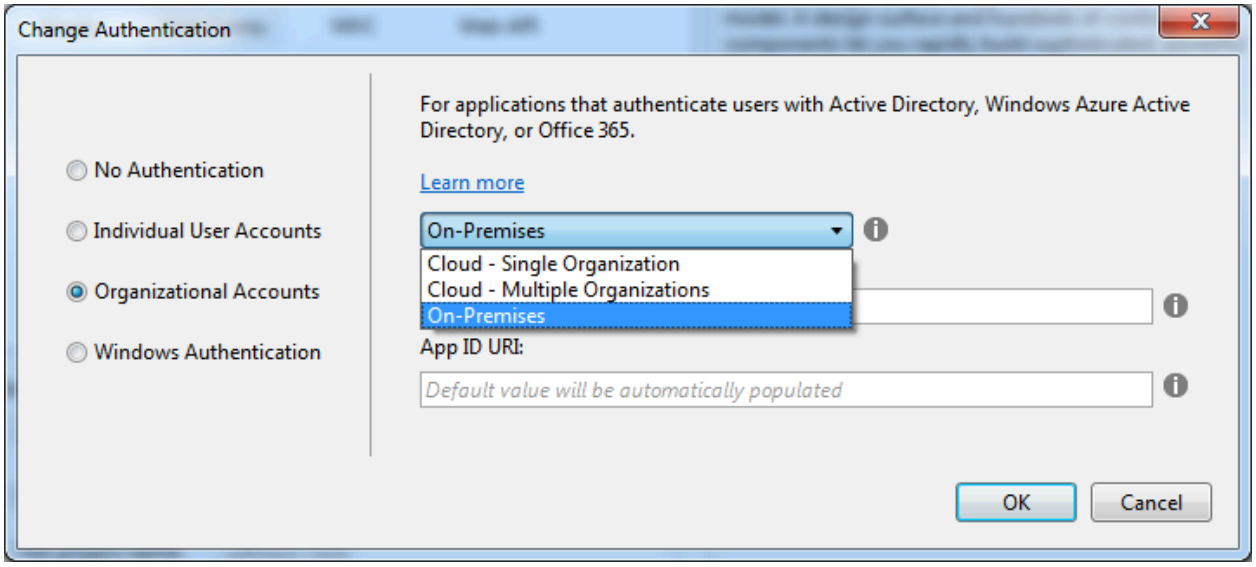
Keep reading.

You need 3 servers

- DualShield
- ADFS + DC. ADFS 3.0 can be installed on DC
- Web application server (IIS)



You can use Visual Studio to create a ASP.NET Web Application. In VS 2013, you can specify the authentication



For simplicity, here we use the instruction and package from [How to Build Your ADFS Lab on Server 2012, Part2: Web SSO]. It is much easier to use its powershell script to deploy a ASP.NET application.

In theory, the web server is not necessary to be a member server of AD, but let us just follow the instruction and prerequisites. Make sure `ADFSServer dc01.nanoart.local` is accessible from this web server machine.

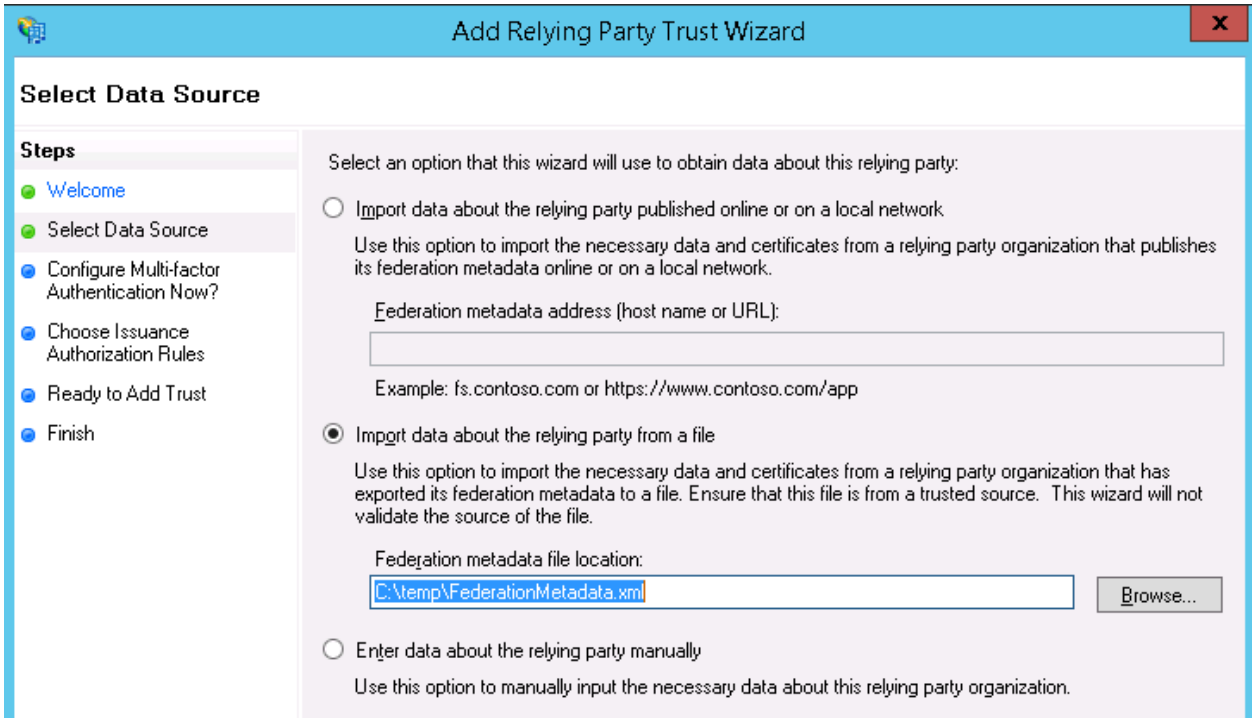
```
PS C:\temp> .\deploy-testsite.ps1 -SourcePath C:\temp\deploy -SiteName adfstest
-SitePhysicalPath c:\sites\adfstest -ADFSServer dc01.nanoart.local -AppFQDN
adfstest.nanoart.local
```

You can look at the content of the script to understand what it is trying to do, but how it makes a web as claims-aware application? the answer is, it modifies the web.config file

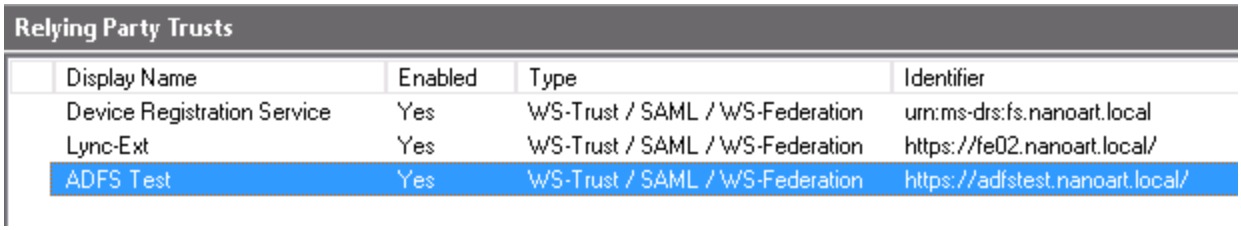
```
<microsoft.identityModel>
	<service saveBootstrapTokens="true">
		<audienceUris>
			<add value="https://adfstest.nanoart.local/" />
		</audienceUris>
		<federatedAuthentication>
			<wsFederation passiveRedirectEnabled="true"
issuer="https://fs.nanoart.local/adfs/ls/" realm="https://adfstest.nanoart.local"
requireHttps="true" />
			<cookieHandler requireSsl="true" />
		</federatedAuthentication>
		<applicationService>
			<claimTypeRequired>
				<!--Following are the claims offered by STS
'http://fs.nanoart.local/adfs/services/trust'. Add or uncomment claims that you
require by your application and then update the federation metadata of this
application.-->
				<claimType
type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" optional="true" />
				<claimType
type="http://schemas.microsoft.com/ws/2008/06/identity/claims/role" optional="true" />
				<!--<claimType
type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
optional="true" />-->
			</claimTypeRequired>
		</applicationService>
		<certificateValidation certificateValidationMode="None" />
		<issuerNameRegistry
type="Microsoft.IdentityModel.Tokens.ConfigurationBasedIssuerNameRegistry,
Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35">
			<trustedIssuers>
				<add thumbprint="834A5AC753F91342A7DBE8C0C170A21964A776ED"
name="http://fs.nanoart.local/adfs/services/trust" />
			</trustedIssuers>
		</issuerNameRegistry>
	</service>
</microsoft.identityModel>
```

Now it is time to add the web application as a **Relying Party Trust** in ADFS. Choose the option "**Import data about the relying party from a file**". This file `FederationMetadata.xml` is under the folder `C:\sites\adfstest\FederationMetadata\2007-06`, You may be keen to see what the content is,

```
<?xml version="1.0" encoding="utf-8"?>
<EntityDescriptor ID="_0db3255c-8a7c-4fa0-a7c8-cd49d89866d6"
entityID="https://adfstest.nanoart.local/"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
       <RoleDescriptor xsi:type="fed:ApplicationServiceType"
xmlns:fed="http://docs.oasis-open.org/wsfed/federation/200706"
protocolSupportEnumeration="http://docs.oasis-open.org/wsfed/federation/200706"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
              <fed:ClaimTypesRequested>
                     <auth:ClaimType
Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" Optional="true"
xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706" />
                     <auth:ClaimType
Uri="http://schemas.microsoft.com/ws/2008/06/identity/claims/role" Optional="true"
xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706" />
              </fed:ClaimTypesRequested>
              <fed:TargetScopes>
                     <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
                            <Address>https://adfstest.nanoart.local/</Address>
                     </EndpointReference>
              </fed:TargetScopes>
              <fed:PassiveRequestorEndpoint>
                     <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
                            <Address>https://adfstest.nanoart.local/</Address>
                     </EndpointReference>
              </fed:PassiveRequestorEndpoint>
       </RoleDescriptor>
</EntityDescriptor>
```



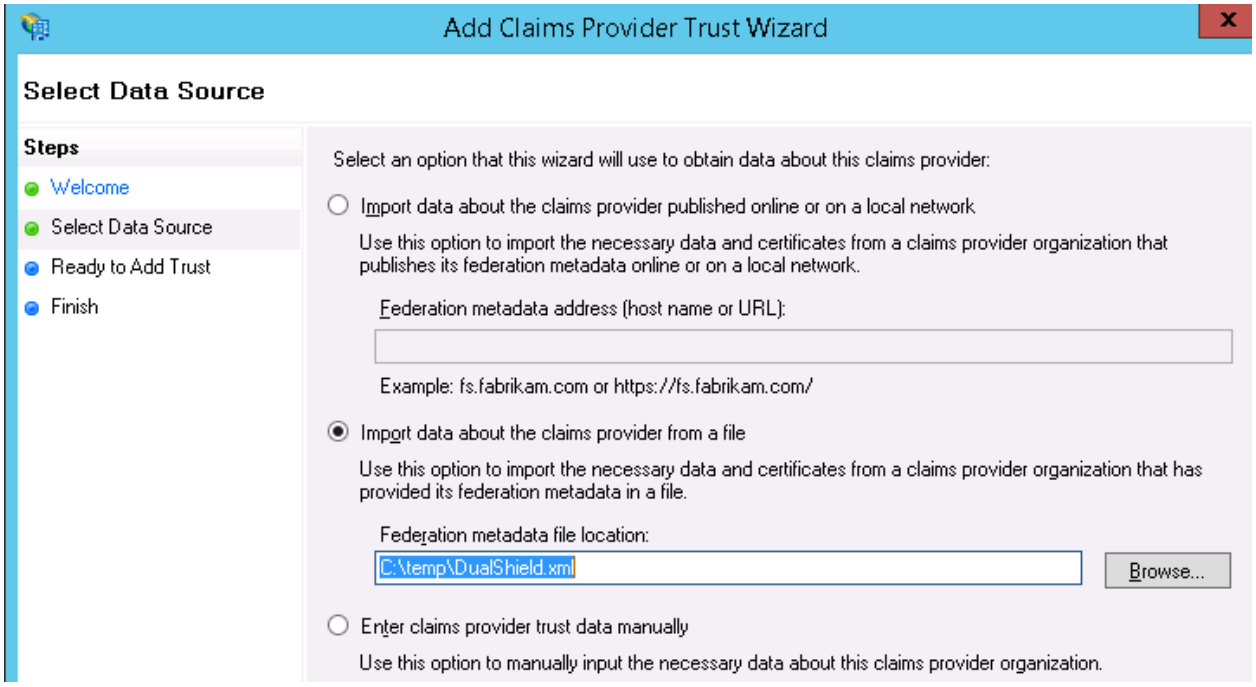
You should see a newly added **Relying Party Trust** called "**ADFS Test**".



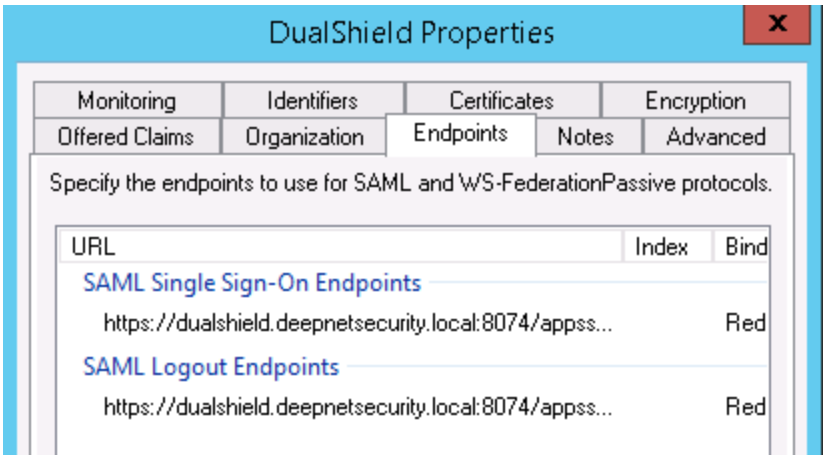
It is wise to test the built-in Active Directory authentication right now by accessing

https://adfstest.nanoart.local

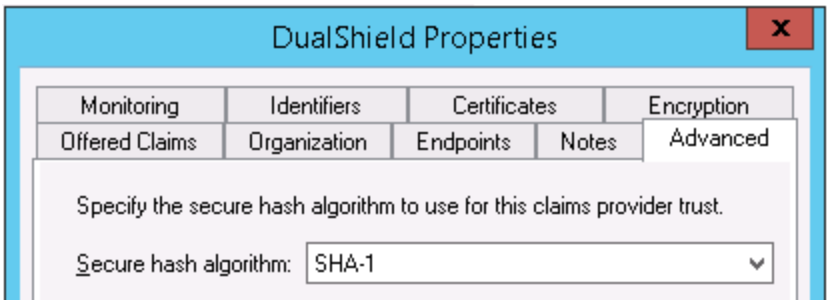
If everything is right, then let us go ahead to configure DualShield as a third party claim provider. Obviously it is done on ADFS server,



The file `DualShield.xml` is an IDP metadata file which is downloaded from DualShield server. You must have **SSL** on DualShield SSO port, otherwise, the import won't finish successfully.



After import, you may need to manually change the hash algorithm to SHA-1



Are we finished? Not yet, you have to add **ADFS server** as a service provider in DualShield. Certainly you need the metadata of ADFS server, which can be downloaded through

https://fs.nanoart.local/FederationMetadata/2007-06/FederationMetadata.xml

Unfortunately, DualShield doesn't accept this complicated xml file, so we have to simplify it! Here is the working version.

```
<EntityDescriptor entityID="http://fs.nanoart.local/adfs/services/trust"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
       <SPSSODescriptor
              AuthnRequestsSigned="false"
              WantAssertionsSigned="true"
              protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

              <SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://fs.nanoart.local/adfs/ls/"/>

<NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
              <AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://fs.nanoart.local/adfs/ls/" index="0" isDefault="true"/>

       </SPSSODescriptor>
</EntityDescriptor>
```

Now you can test with DualShield Authentication!

In reality, you need to create claim rules to transfer the claims (SAML attributes). Please check the details in AD FS 2.0 Claims Rule Language Primer.

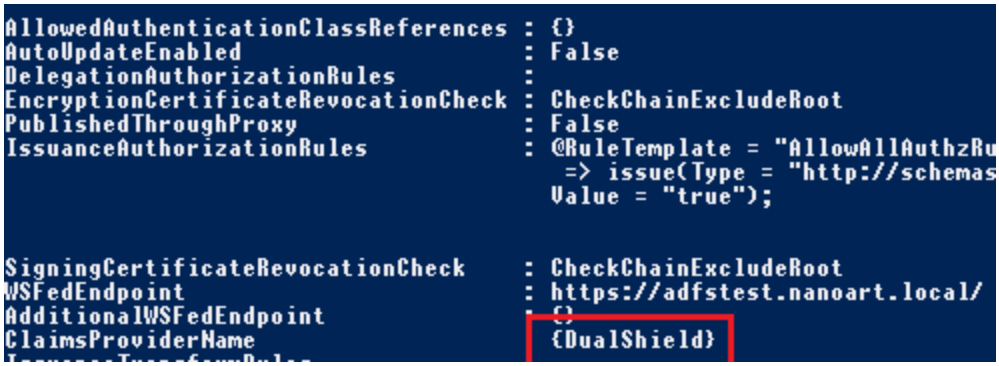
For some scenarios, an organizations might want end users to only see the claims

providers that are specific to an application so that only a subset of claims provider are displayed on the home realm discovery page. Want to use DualShield claim provider directly? use the following Windows PowerShell cmdlet and syntax.

```
Set-AdfsRelyingPartyTrust -Targetname "ADFS Test" -ClaimsProviderName @("DualShield")
```

You can check it with

```
Get-AdfsRelyingPartyTrust -Name "ADFS Test"
```



References

[Identity Developer Training Kit]

[Security Token Visualizer Control for Visual Studio 2010]