[Configuring SharePoint 2013 Forms Based Authentication with SQLMemberShipProvider](https://blogs.technet.microsoft.com/ptsblog/2013/09/20/configuring-sharepoint-2013-forms-based-authentication-with-sqlmembershipprovider/)

Microsoft Walkthrough

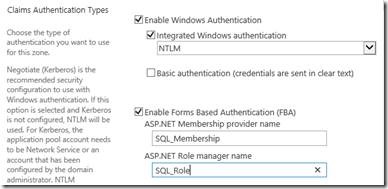
How it’s done

Step 1 – Create the Web Application

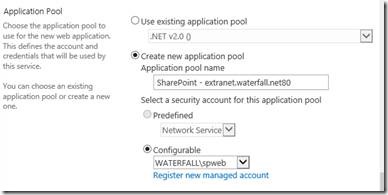
In this step we will be creating the web application with Windows Authentication (Claims) and Forms Based Authentication (FBA) on the same Zone. In SharePoint 2013, you can have multiple authentication providers without extending the web application. Having said that, at times, you might have to extend the web application depending on your scenario. More on that on a different post where I will show you how to use LDAPMemberShipProvider to talk to your AD.

From Central Administration, we will create a Web Application and call it Extranet.waterfall.net and enable both Windows Auth and FBA. Note the names I am using: ASP.NET Membership Provider Name = **SQL\_Membership** and ASP.NET Role manager name = **SQL\_Role**. You can call them whatever you want, just ensure you use the same names everywhere.

fbaMembers, fbaRoles

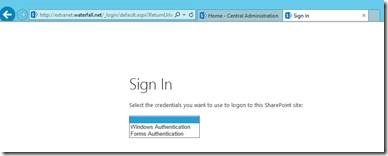
[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/4353.clip_image002_599AD83F.jpg)

We will create a new App Pool and use the Web App Pool account. Make a note of this since you would need to give this account permission in the next step in the ASPNET database.

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/0640.clip_image004_071BF803.jpg)

Create the Web App and then the Site Collection, it doesn’t matter what template you choose. Once the Site Collection is created, visiting the site collection will take you to our default sign in page where you will be asked to choose an Authentication Provider to Sign In with. If you want your External Users only to have the option of FBA, you would want to set this default zone with Windows Auth and extend it and have the FBA on the extended web app. Obviously, the URL’s will then be different.

Your sign in page should look like this (make sure your DNS record (CNAME) point to the WFE01)

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/7180.clip_image006_349D17C6.jpg)

Do you want to see a custom sign in page with your company brand on it? Well, let’s defer that to a different post.

Step 2 – Verify Tools

Now that the web app is created, we will make sure FBA Pack and FBA Configuration manager is deployed as it should be. Go to Central Administration >> System Settings >> Manage Farm Solutions. Make sure fbaConfigFeature.wsp is globally deployed and visigo.sharepoint.formsbasedauthentication.wsp is deployed to http://extranet.yourdomain.com.

TODO: FIGURE OUT SITE HIERARCHY - DO WE NEED ROOT SITE COLLECTION FOR FBA CONFIG TO WORK?

.\Deploy.ps1 http://w2k12R2/

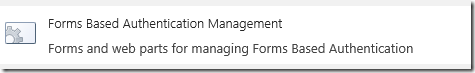
.\Deploy.ps1 http://w2k12R2/customers/WRS

See screenshot below. If the visigo.sharepoint.formsbasedauthentication.wsp is not deployed, click on the WSP and deploy it to your web application.

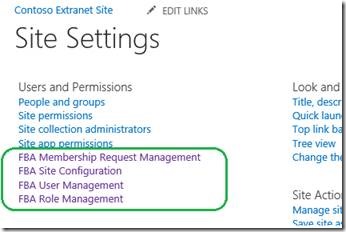
[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/1134.clip_image008_3430E4D1.jpg)

Login to the site collection created in the above step and activate the following feature:

Site Settings >> Site Collection Administration >> Site Collection Features >> Form based Authentication Management

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/8507.clip_image009_76A38707.png)

Once the feature is activated, it should add the following to your Site Settings under User and Permissions

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/7345.clip_image011_1D056A53.jpg)

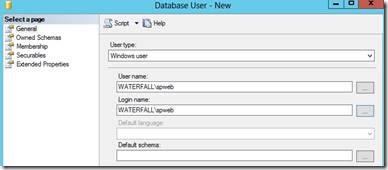
Step 3 – Creating the SQL Database for User Management

The first step is to create the SQL Database that would hold the Extranet Users

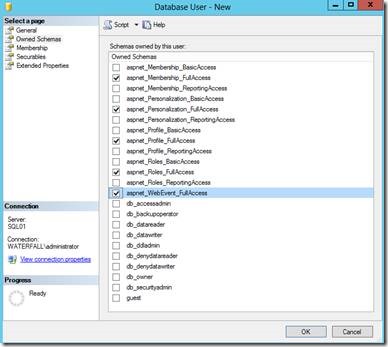
* Browse to C:\Windows\Microsoft.NET\Framework64\v4.0.30319
* Run aspnet\_regsql.exe
* Click Next
* Choose Configure SQL Server for Application Services >> Click Next
* Enter your SQL Server Name , choose Windows Authentication and type in a Database Name

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/8053.clip_image013_141868C5.jpg)

* Click Next twice to provision the database
* Now we need to add the Application Pool that runs the web application and give it required permission. In this case, the application pool name is waterfall\spweb. Perform the following steps:
  + Open up SQL Management Studio, Expand the database we created and expand Security
  + Right click Users and add a new User
  + User Type = Windows User
  + User name = choose <yourAppPoolAccountName>
  + Login name = browse and choose the login name (should be same as the app pool name above)

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/5635.clip_image015_4F6BCE83.jpg)

* + Click Owned Schemas and choose the following:
    - aspnet\_Membership\_FullAccess
    - aspnet\_Persolalization\_FullAccess
    - aspnet\_Profile\_FullAccess
    - aspnet\_Roles\_FullAccess
    - aspnet\_WebEvent\_FullAccess

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/3108.clip_image017_6384EB0C.jpg)

Step 4 – Editing the web.config files

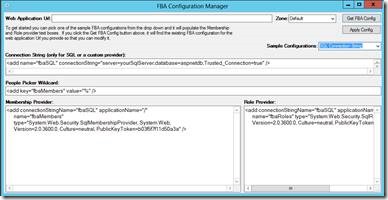
We need edit the following web.config files:

* Web Application Web.config – WFE server
* STS Application web.config – WFE server and Application Server
* Central Admin web.config – CA Server
* If you have more WFEs and App Servers, you need to edit them as well. A lot of people puts these in there machine.config file as well so that it gets inherited to the web.config file. I am not too keen on editing the machine.config file.

Let’s login to our WFE server and fire up FBAConfigMgr.exe. While you can get the code you need from here and edit web.config yourself, if you just let the tool run its course, it will create a Timer Job and do the task for you. In the FBAConfigMgr type in your application URL and from the sample configuration choose the following:

* People Picker Wildcard
* Connection String
* Membership Provider
* Role Provider

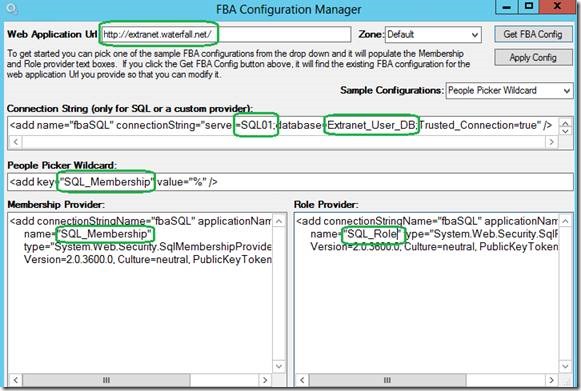
Here is what the screen looks like when default values are chosen:

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/3527.clip_image019_45A6670B.jpg)

We will modify the default values to reflect the following (highlighted items need modification per your environment):

* Web Application URL – **http://digitalpen.css.local/**
* People Picker Wildcard – <add key="**SQL\_Membership**" value="%" />
* Connection String –   
  <add name="fbaSQL" connectionString="server=**SP\_DB\_ALIAS**;database=**Extranet\_User\_DB**;Trusted\_Connection=true" />
* Membership Provider –   
  <add connectionStringName="fbaSQL" applicationName="/"   
  name="**SQL\_Membership**"   
  type="System.Web.Security.SqlMembershipProvider, System.Web,   
  Version=2.0.3600.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
* Role Provider –   
  <add connectionStringName="fbaSQL" applicationName="/"   
  name="**SQL\_Role**" type="System.Web.Security.SqlRoleProvider, System.Web,   
  Version=2.0.3600.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"/>

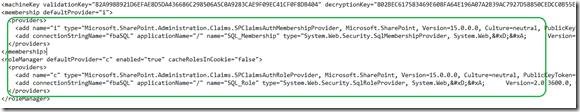
The screen should now look like this:

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/3652.clip_image021_530C7A11.jpg)

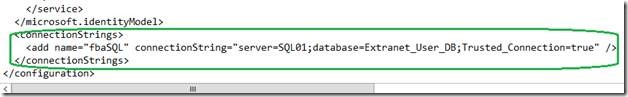
It’s time to hit Apply Config. This will create a timer job to update your web.config files. Though it creates a backup, you should be proactive and take a backup of your web application web.config and sts web.config file. Here is [how](https://blogs.technet.microsoft.com/ptsblog/2013/09/20/configuring-sharepoint-2013-forms-based-authentication-with-sqlmembershipprovider/#_How_to_backup) to back up the web.config file and here is [how](https://blogs.technet.microsoft.com/ptsblog/2013/09/20/configuring-sharepoint-2013-forms-based-authentication-with-sqlmembershipprovider/#_Where_is_the) to find the STS web.config file.

Once you click Apply Config, the tool will tell you when it’s done. It might take a few mins before you see any changes, so wait for it (you should see a new backup file created for your web.config file with time stamp and \_FBAConfigMgr in the end of the file). To verify that the job is done, open up the web.config for your web application and search for <membership. You should see the following:

**<<Web Application web.config file>>**

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/4555.clip_image023_7BAAE618.jpg)

The ConnectionStrings gets added to the end of the file right above </configuration>

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/7612.clip_image025_3287CB10.jpg)

**<<STS web.config file>>**

Open up the STS Web.Config and you should see the following:

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/1362.clip_image027_6DDB30CE.jpg)

The ConnectionStrings gets added to the end of the file as well just like web.config of the web application.

**<<Central Administration web.config file on App Server>>**

If you go back to the application server and open up the web.config file for the Central Admin site, you will see there are no changes made there. So we will make that change manually. Create a backup of the file then open up the file and find <Machine. It should look like this:

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/7144.clip_image029_5B263717.jpg)

We will add the following (copied from web.config file of web application or the code from FBAConfigMgr)

1. Search for <machineKey and paste the following under <rolemanager><providers>   
<add connectionStringName="fbaSQL" applicationName="/" name="**SQL\_Role**" type="System.Web.Security.SqlRoleProvider, System.Web,&#xD;&#xA; Version=2.0.3600.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

2. Under <membership><providers> paste the following   
<add connectionStringName="fbaSQL" applicationName="/" name="**SQL\_Membership**" type="System.Web.Security.SqlMembershipProvider, System.Web,&#xD;&#xA; Version=2.0.3600.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />   
The screen should now look like this:   
[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/6012.clip_image031_1D98D94E.jpg)

3. Scroll to the end of the document and paste the following right before </configuration>   
<connectionStrings>

<add name="fbaSQL" connectionString="server=**SQL01**;database=**Extranet\_User\_DB**;Trusted\_Connection=true" />

</connectionStrings>

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/8267.clip_image033_31B1F5D7.jpg)

**<<STS web.config file on App Server>>**

Just like the Central Admin web.config make the same changes on this web.config as well. Just make sure you are pasting the information from RoleManager Providers and Membership Providers in the right place. Here is what the code looks like (you can use the code below are make changes to the highlighted areas to suit your environment):

<system.web>

<membership>

<providers>

<add connectionStringName="fbaSQL" applicationName="/" name="**SQL\_Membership**" type="System.Web.Security.SqlMembershipProvider, System.Web,&#xD;&#xA; Version=2.0.3600.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

</providers>

</membership>

<roleManager>

<providers>

<add connectionStringName="fbaSQL" applicationName="/" name="**SQL\_Role**" type="System.Web.Security.SqlRoleProvider, System.Web,&#xD;&#xA; Version=2.0.3600.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

</providers>

</roleManager>

</system.web>

<connectionStrings>

<add name="fbaSQL" connectionString="server=**SQL01**;database= **Extranet\_User\_DB**;Trusted\_Connection=true" />

</connectionStrings>

Here is a screenshot

Step 5 – Use FBA Pack to add and manage users

Our configurations are done. We will now go to our site collection and use the FBA Pack to add / manage users and Roles

Go to Site Settings and click on FBA User Management >> Click New User and create a dummy user and add him to the contributor group

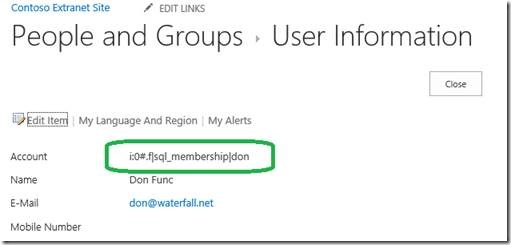
[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/4478.clip_image037_4C7E1BE3.jpg)

Step 6 – Verify Forms user

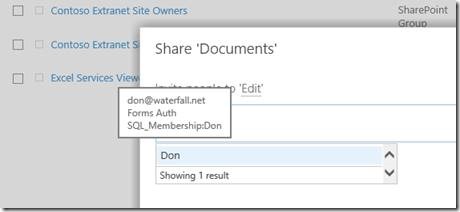
Now open up IE in InPrivate mode and visit your site collection and this time choose Forms Authentication and enter the account information you just created to log in. You’re done!

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/4314.clip_image039_0EF0BE1A.jpg)

Click on the user name and My Settings, you will see the account information coming from SQL Membership Provider

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/7043.clip_image041_23760D98.jpg)

If you go to a document library and try and add the user there, you will see it resolves from your SQL database

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/4010.clip_image043_77C5439B.jpg)

Appendix

How to create SQL Alias for SharePoint

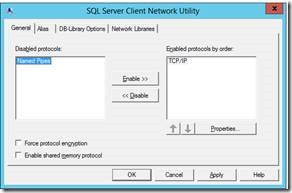
Follow the steps below to create a SQL Alias on all your SharePoint Servers:

TechNet Reference: <http://technet.microsoft.com/en-us/library/ff607733.aspx#clientalias>

1. Perform this on the Application Server that is hosting Central Administration

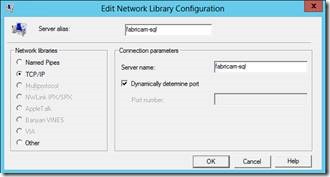
a. Stop all SharePoint Services

b. Open CLICONFIG.exe from C:\Windows\System32\cliconfg.exe (64 bit version of cliconfig.exe)

c. Enable TCP/IP under general tab   
[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/4331.clip_image045_6C2F865C.jpg)

d. Click on Alias Tab

e. Type Current SQL Server Name in the Alias Name field

f. Type Current SQL Server Name in the Server field (see screenshot below. In your case SQL Alias and SQL Server name is the same)   
[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/6835.clip_image047_79959962.jpg)

g. Validate SQL Alias

i. Create a new text file on SharePoint Server and name it “TestDBConnection.udl”

ii. Double click to open the file and enter your SQL Server Alias name

iii. Use Windows Integrated Security

iv. You should be able to see all your SharePoint databases when you click on “Select the database on the Server”

h. Start all services for SharePoint Server / Reboot SharePoint Server

i. Perform the steps above on all other SharePoint servers

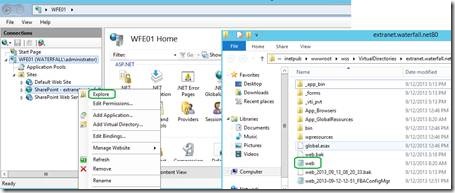
How to backup web.config file

To back up web.config file, perform the following:

· From IIS Manager (start >> Run > inetmgr)

· Right click on the web site and click Explore

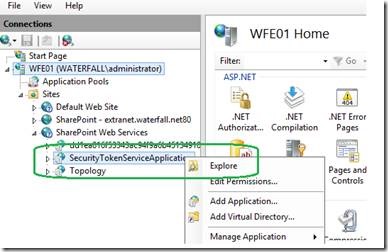
· Copy the web.config file somewhere else, or the in the same location with a different name

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/1541.clip_image049_2716B926.jpg)

Where is the STS web.config file?

· On your WFE open up IIS Manager and expand SharePoint Web Services

· Right click on SecurityTockenServiceApplication and click Explore

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/86/44/metablogapi/8400.clip_image051_30727E5A.jpg)

Tags [SharePoint 2013](https://blogs.technet.microsoft.com/ptsblog/tag/sharepoint-2013/)