## Configuring a SharePoint Server 2013 Farm

**Lab Time**: 45 minutes

**Lab Folder**: C:\Student\Modules\FarmConfiguration\Lab

**Lab Overview**: In this lab you will get some experience using the SharePoint 2013 PowerShell cmdlets and conduct common farm configuration tasks such as registering service accounts and creating service applications.

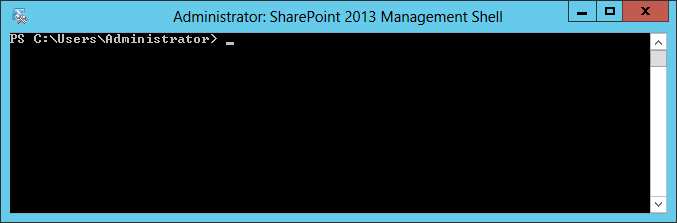
### Exercise 1: Working with PowerShell in the SharePoint 2013 Management Shell

In this exercise you will explore the power of the SharePoint 2013 Management Shell. With the management shell, you do not have to explicitly register the **Microsoft.SharePoint.Powershell** snap-in that contains the SharePoint 2013 cmdlets.

1. Open the SharePoint 2013 Management Shell in order to execute PowerShell commands using the SharePoint-specific cmdlets that are included with the SharePoint PowerShell snap-in.
   1. Click on the **Windows** key to display the **Windows Start** page.
   2. Somewhere on the **Windows** start page you should be able to locate three different tiles related to SharePoint 2013. Click the title with the caption **SharePoint 2013 Management Shell**.



* 1. Once the **SharePoint 2013 Management Shell** has been launched,you can now type in PowerShell commands and execute them by pressing the **[ENTER]** key.



1. Inside the **SharePoint 2013 Management Shell**, type the following commands to change the directory to the root of the current drive “**c:\**” and clear the window of existing text.

**CD\** + **[ENTER]**

**CLS** + **[ENTER]**

1. Run the Get-Command cmdlet with the following parameter and press return to see a listing of all the cmdlet included with the Microsoft.SharePoint.PowerShell snap-in.

Get-Command -PSSnapin Microsoft.SharePoint.PowerShell

1. Anytime a large result set is expected or returned from PowerShell you can re-direct the output to a text file to make viewing this easier/possible. Run the Get-Command again and pipe the output to a new text file named SP2013Cmdlets.txt.  
   (Note: the pipe symbol “**|**” is typically located on the keyboard just above the [ENTER] key and accessed by pressing the **[SHIFT]** **+ \** keys simultaneously)

Get-Command -PSSnapin Microsoft.SharePoint.PowerShell | out-file –filepath “C:\Student\SP2013Cmdlets.txt”

(Note: the pipe symbol (“|”) is used to take the output from the previous statement as the input for the next. This is why the output for the Get-Command is sent to the out-file statement and ends up in the text file. You can chain together many separate PowerShell statements in this manner without using a separate variable to store this data between statements.)

1. Open SP2013Cmdlets.txt with Notepad using the following command to inspect the cmdlets provided by the Microsoft.SharePoint.PowerShell snap-in. Note in particular those cmdlets that begin with the **Get** verb.

Notepad C:\Student\SP2013Cmdlets.txt

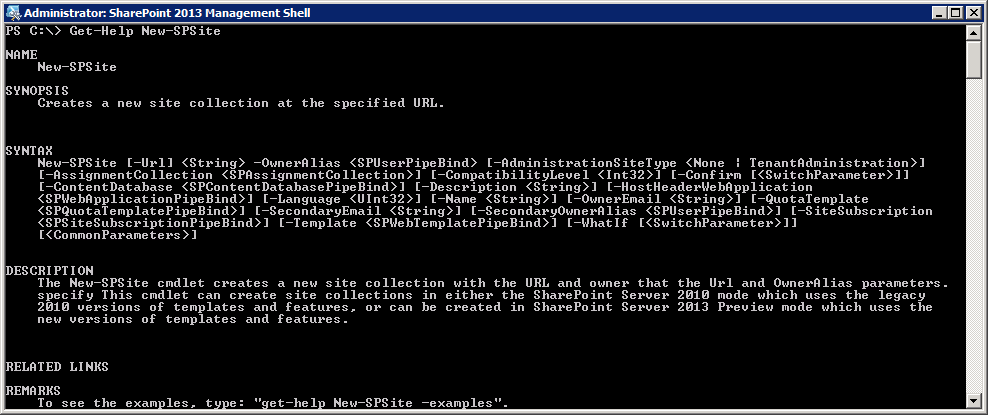
1. Another alternative to re-directing large amounts of output to a text file and then examining this for items of interest is to filter the result set being returned. Try running the Get-Command cmdlet again requesting only the cmdlets based on the cmdlet verb of Get. Note how this filtering drastically reduces the result set.

Get-Command -PSSnapin Microsoft.SharePoint.PowerShell -Verb Get

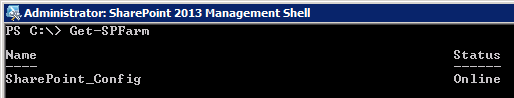
1. You can also get some help on cmdlets. Type in the following:

Get-Help New-SPSite

1. Running this cmdlet will display information about the New-SPSite cmdlet. It lists the different arguments that you can use to create a new SharePoint site. Each argument is listed within brackets ([]) and the data type of the argument is listed within “<”.



1. Now enter Get-SPFarm. The only thing you get is the name of the configuration database.



1. If you want to see all members available on the SPFarm object you can execute the following command which outputs a list that shows which members are properties and which members are methods.

Get-SPFarm | Get-Member

1. . If you want to see all the details, type the following:

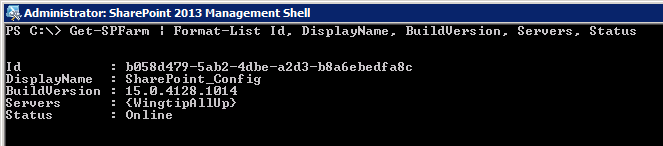
Get-SPFarm | Select \*

When the results are returned note the names of the different properties on the left and values on the right.

1. If you are interested in only a few of the properties, you can restrict the number of properties that are displayed by using the **Format-List** cmdlet with the output from the **Get-SPFarm** cmdlet and specifying the fields to return by name. Type the command below to see this additional way to filter a large list.

Get-SPFarm | Format-List Id, DisplayName, BuildVersion, Servers, Status

1. This results in the following:



1. As you can see DisplayName returns the name of the configuration database and the servers in the farm is WingtipServer.
2. Execute a few of the commonly-used SharePoint cmdlets.
   1. Execute the **Get-SPManagedAccount** to get information about the managed accounts in the local farm. At this time there should be only one managed account for **WINGTIP\SP\_Farm**.

Get-SPManagedAccount

* 1. Execute the **Get-SPServiceApplication** cmdlet to retrieve information about the service applications in the local farm. At this time there should only be two service applications which are the **Security Token Service Application** and the **Application Discovery and Load Balancer Service Application**.

Get-SPServiceApplication

* 1. Execute the **Get-SPWebApplication** cmdlet to get information about the web applications in the local farm. Note that when you execute this cmdlet, it will return an empty set because no web applications have been created other than the web application for Central Administration.

Get-SPWebApplication

* 1. Execute the **Get-SPWebApplication** cmdlet once again passing the **IncludeCentralAdministration** parameter which will return information about the web applications (of which there are none currently) and display information about the Central Administration web application.

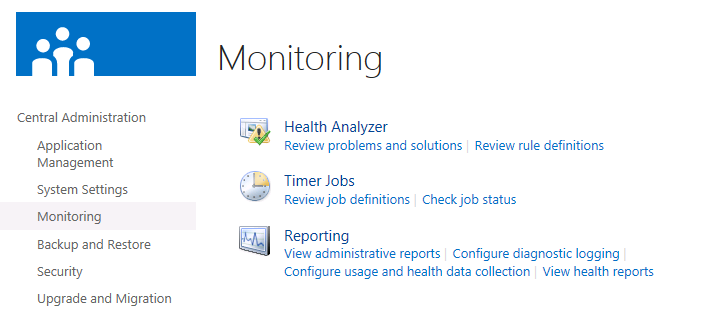
Get-SPWebApplication -IncludeCentralAdministration

In this exercise you got some hands-on experience in working with some of the provided SharePoint 2013 Windows PowerShell cmdlets as well as working with custom Windows PowerShell scripts.

### Exercise 2: Working with Managed Timer Jobs

This exercise you will walk you through the timer job definitions and the managed timer jobs.

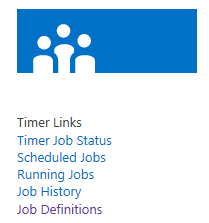
1. From within the **Central Administration** home page, select the **Monitoring** hyperlink.
2. This brings you to the page from where you can manage settings on Health Analyzer, Timer Jobs and Reporting. Notice the section on **Timer Jobs**.



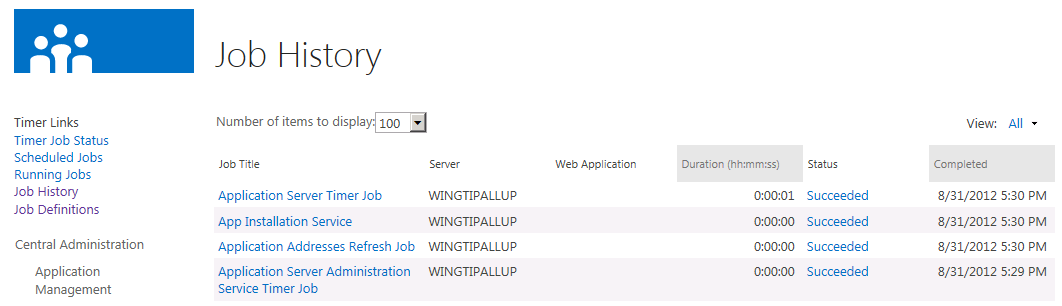
1. Within the Timer Jobs section, click the **Review job definitions** hyperlink. This brings you to a page where all timer job definitions are listed with the web application they run on and the schedule.



1. In the right upper corner you see a dropdown list from where you can change the view to **Service** or to **Web Application**. This will filter the list of job definitions. Note that if you select either of these filters you will not see any job definitions because you have not yet created any service applications or Web applications within the farm.
2. Notice the links on the left-hand side of the page in the **Timer Links** sections.



1. Click on the **Job History** link to navigate to the page that shows which timer jobs have run recently and when. It also indicates whether the timer job ran successfully.



1. Click on the **Job Definitions** link to navigate back to the page that shows timer job definitions. Scroll down through the list and click the job definition named **Delete Job History**. This takes you to a page where you can inspect, schedule and run this job definition. You should be able to verify that this job definition is currently configured to run once a week very early on Sunday morning.



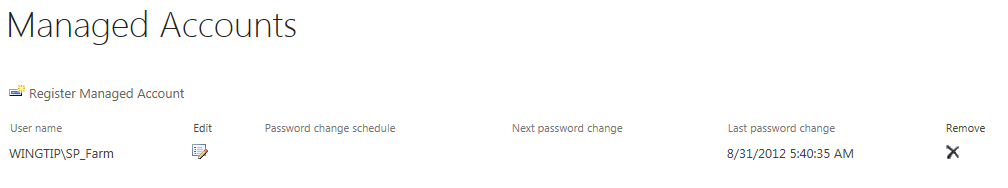
1. Click the **Run Now** button to immediately run the job..
2. Click the **Scheduled Jobs** hyperlink. This brings you to page where all scheduled timer jobs are listed, ordered following their next start time. Clicking the name of the timer job will again bring you to the timer job details page.
3. Click the **Running Jobs** hyperlink. You will see which jobs are currently running and their progress.
4. Click the **Job History** hyperlink. Notice the status of the jobs listed. All jobs that have run are listed here with a status, the time they needed to complete and the time at which they completed.
5. The timer job status is displayed as a hyperlink. Clicking the **Status hyperlink** (likely one that says **Succeeded**) for the first Job displayed on the Job History Page. This displays a table at the top of the page with more details about the status. This is interesting when a job fails.
6. Return to the **Monitoring** page in Central Administration.

This exercise you walked you through the timer job definitions and the managed timer jobs.

### Exercise 3: Registering Managed Accounts

SharePoint 2013 allows administrators to pre-configure managed accounts to be used when configuring Service applications and Web applications. This way administrators don't have to remember or lookup usernames and passwords for service accounts every time they configure a new service application or web application.

1. Navigate to **Central Administration**.
2. Click the link for the **Security** section.
3. Select the **Configure managed accounts** hyperlink (Located underneath the **General Security** section on the **Security** page).
4. Notice that the **SP\_Farm** account is already listed. This account was automatically added as a managed account when you ran the SharePoint Products Configuration Wizard in an earlier lab exercise when you created the farm.



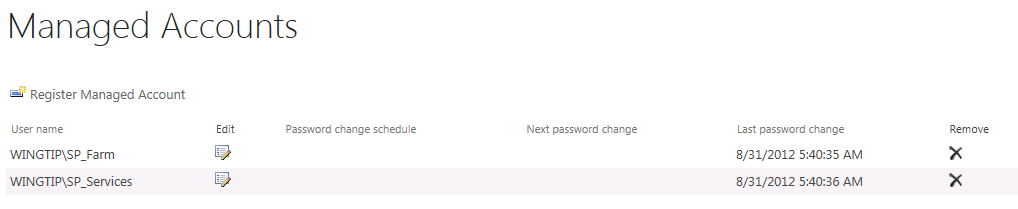
1. Create a managed account for **WINGTIP\SP\_Services**

Click the **Register Managed Account** button.

Specify **WINGTIP\SP\_Services** as user name and **Password1** as password.

Click the **OK** button.

The **SP\_Services** account appears now in the list of managed accounts.



1. Create a managed account for **WINGTIP\SP\_Content** using Windows PowerShell instead of using Central Administration.
   1. Open the Open the Windows PowerShell ISE and create a new script.
   2. Before adding any code, save the script to the folder for this lab and name it **CreateManagedAccount.ps1**.
   3. Add the following code.

Add-PSSnapin Microsoft.SharePoint.PowerShell

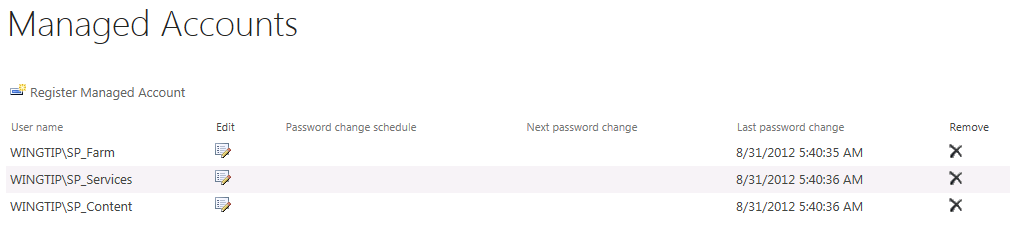
$credential = Get-Credential "WINGTIP\SP\_Content" -Verbose

New-SPManagedAccount -Credential $credential

* 1. Save and run the script. When you run the script, you will be prompted to enter a password. Enter the password for this account which is **Password1** and click **OK**.



* 1. Return to the Managed Account page in Central Administration. Refresh the page and verify that the **SP\_Content** account has been added as a managed account.



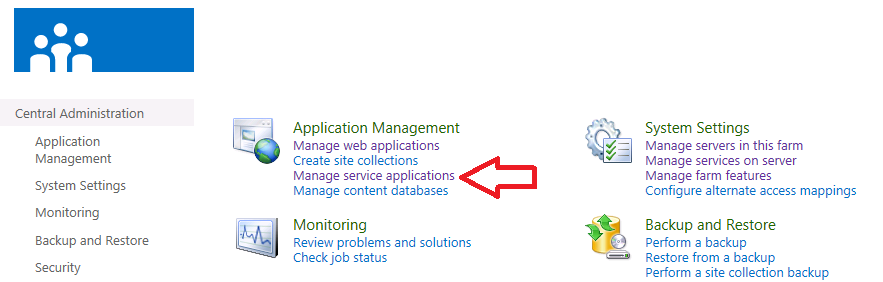
1. Return to the **Security** page.

In this exercise you learned to add managed accounts both with Central Administration and with a PowerShell script.

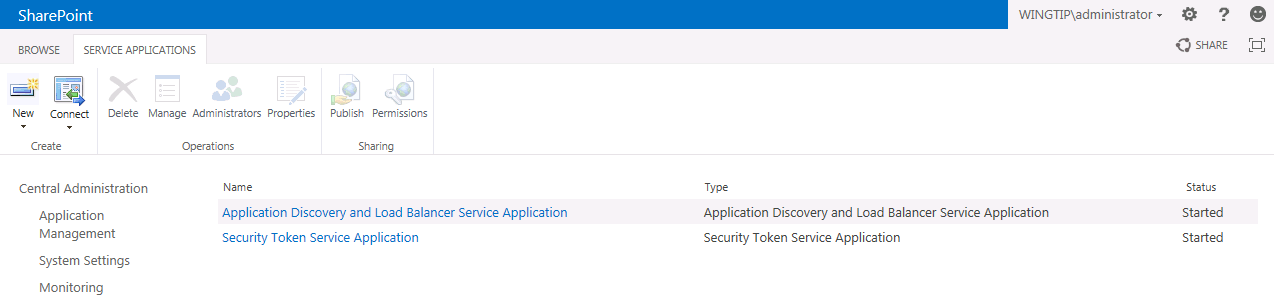
### Exercise 4: Creating a Service Application Instance by Hand

In this exercise you will manually configure the **Word Automation Services** application using Central Administration. This service allows you to perform file operations on the server that previously required automating Word on the desktop, like conversions between document formats (e.g. DOC to DOCX) , converting to fixed formats (e.g. PDF or XPS) , updating fields, importing "alternate format chunks", etc.

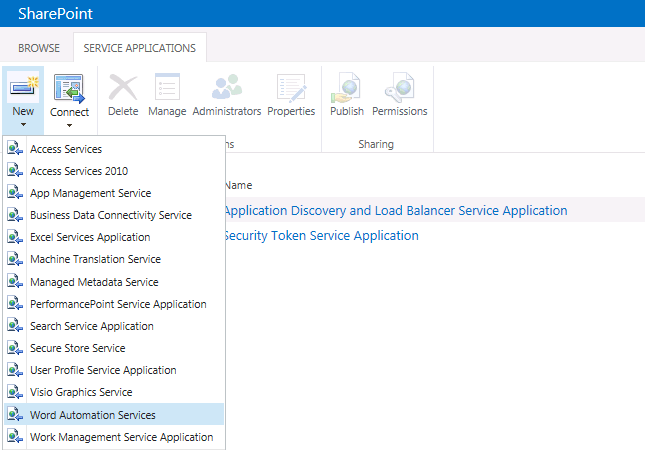
1. Navigate to the **Central Administration** home page.
2. Click the **Manage service applications link** in the **Application Management** section.



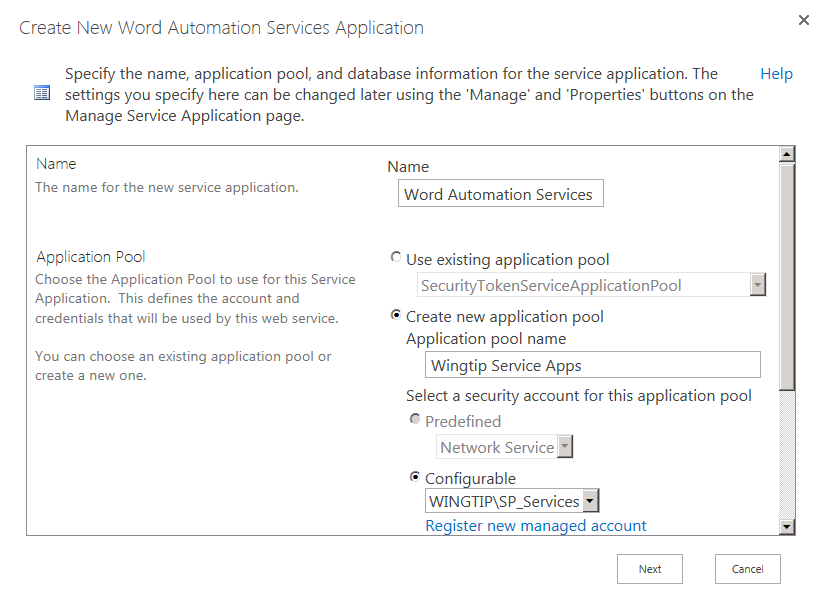
1. At this point you should only see two service applications that were automatically created and configured at farm creation time.



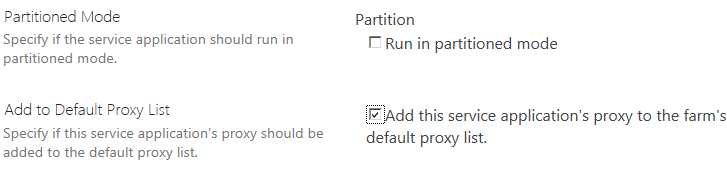
1. In the Service Applications page select the **drop-down arrow** on the bottom of the **New** button and choose Word Automation Services.



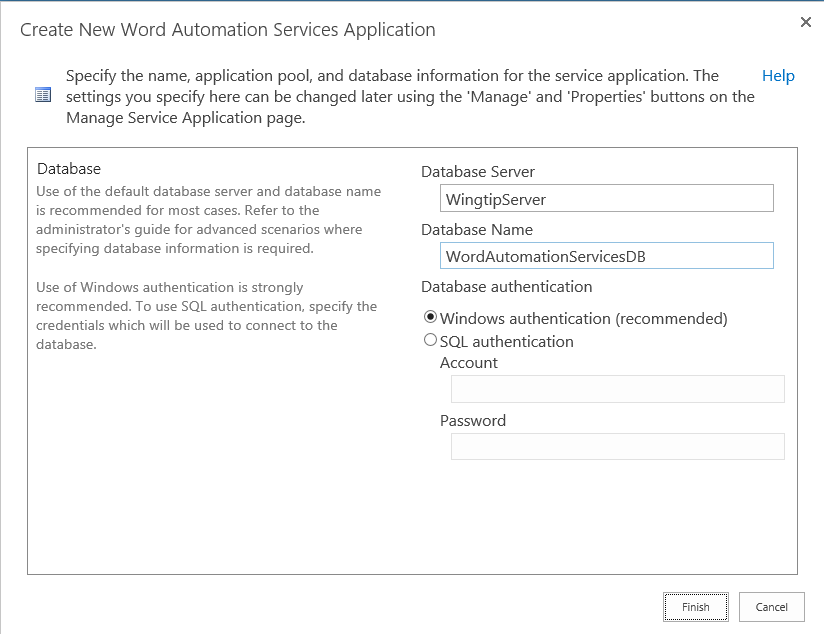
1. A dialog where you can create the new service application is displayed.
   1. Give it the name Word Automation Services.
   2. Create a new application pool and give it the name Wingtip Service Apps.
   3. Choose the WINGTIP\SP\_Services managed account as worker process identity.



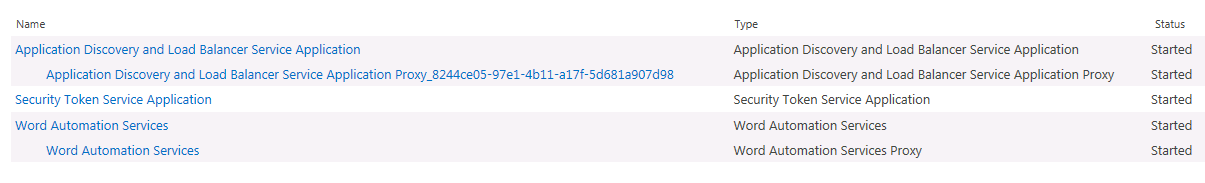
* 1. Make sure to check the checkbox with the option Add this service application's proxy to the farm's default proxy list.



* 1. Click the **Next** button.
  2. In the second page of the dialog you have to specify the database server and the database name that will be used by the service application. The Database Server setting should be set to **WingtipServer**. Enter in a Database Name of **WordAutomationServicesDB** and click the **Finish** button.



* 1. Click the **Finish** button which will create the new service application instance and then bring you back to the **Service Applications** page.
  2. On the **Service Applications** page, scroll down and note that the **Word Automation Service** application is created and that the **Word Automation Service** is started.

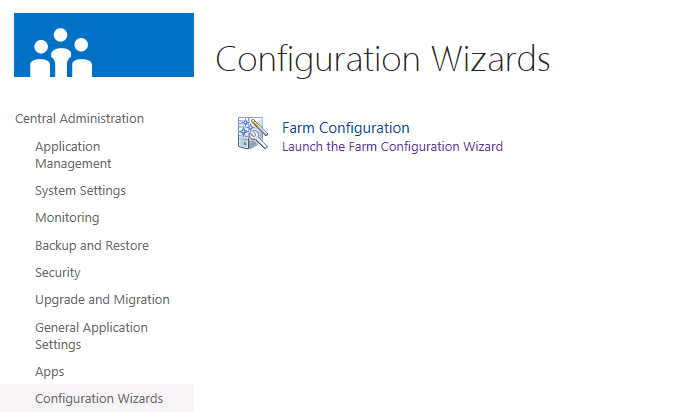


In this exercise you created a new service application by hand in Central Administration.

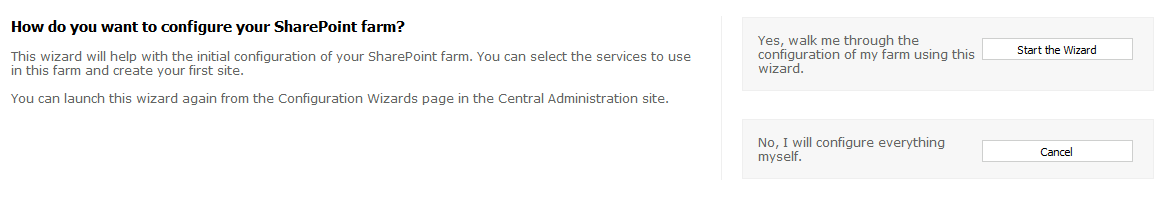
### Exercise 5: Configuring Service Applications using the Farm Wizard

Now you have seen how to create a new instance of a service application by hand in Central Administration. In this exercise you will automate creating several service application instances using the SharePoint 2013 Farm Configuration Wizard.

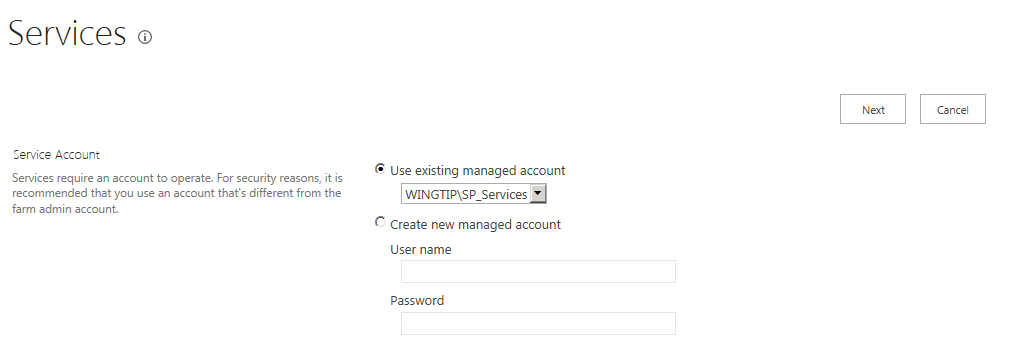
1. From the Home page of Central Administration, click the **Configuration Wizards** link (The Bottom Choice on the left side of the screen’s link list)
2. You should now be on the Configuration Wizards page. You can see there is only one wizard named **Farm Configuration**.



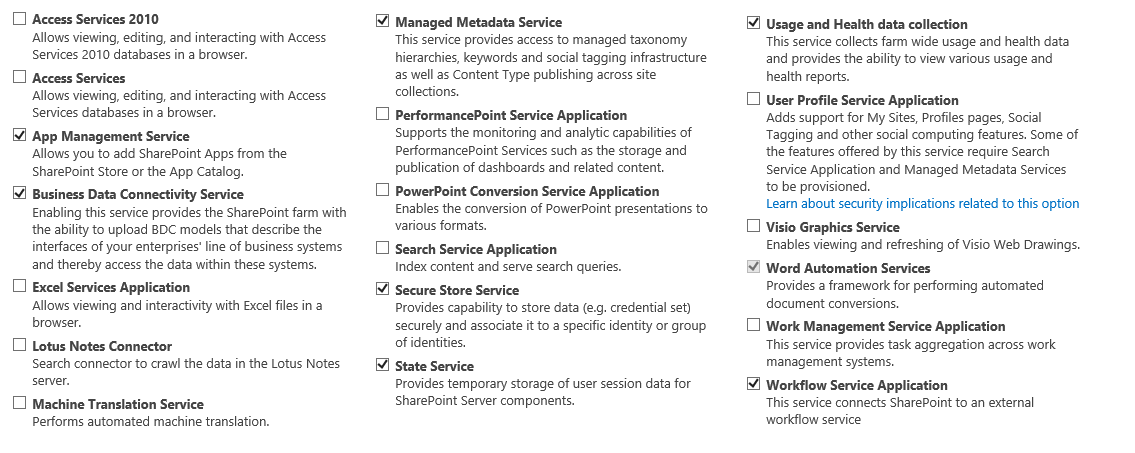
1. Click on the link which reads **Launch the Farm Configuration Wizard**.
2. You should see a page which allows you to start the wizard. Click the **Start the Wizard** button.



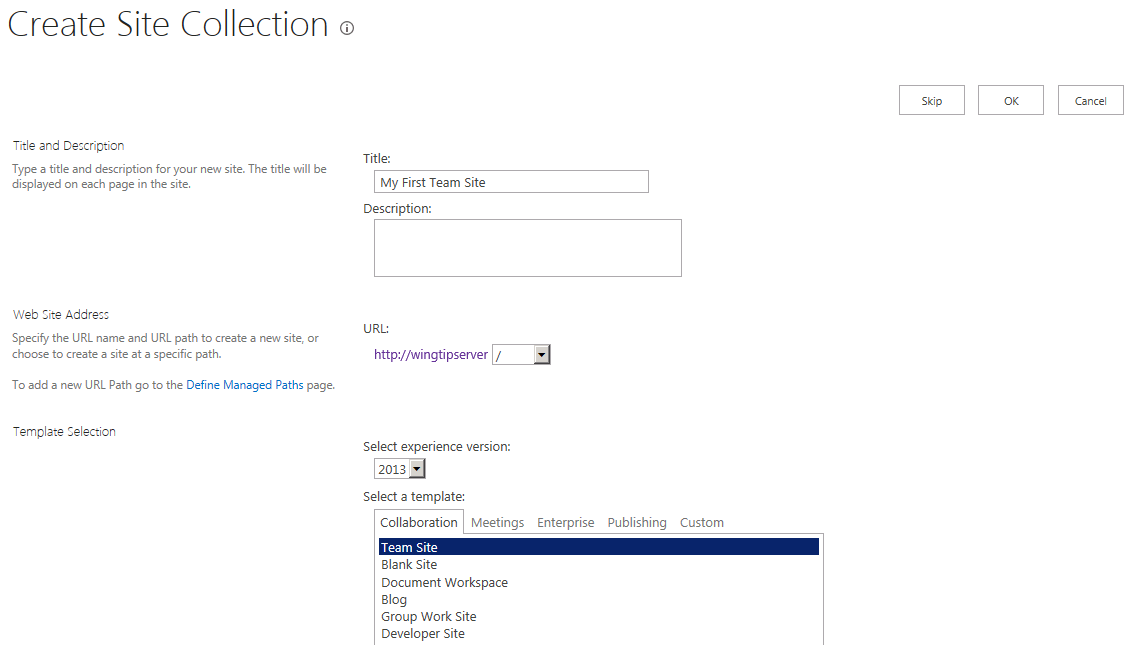
1. The wizard starts by taking you to the **Services** page. Configure the **Service Account** setting with managed account named **WINGTIP\SP\_Services**.



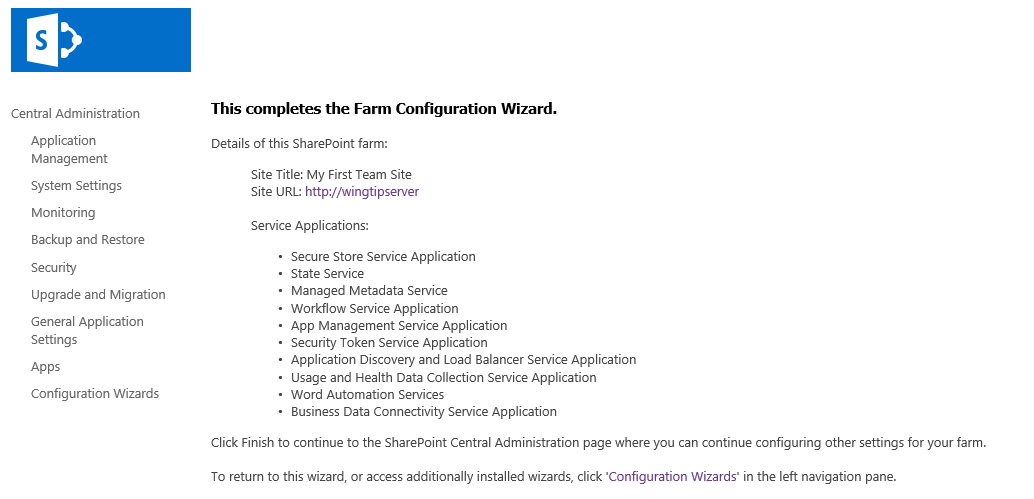
1. In the **Services** section, you must select what service applications you want configured. You should note that the **Word Automation Services** check box is checked and disabled because you created the service application instance for it earlier in this lab. Select the service applications shown in the following screenshot and make sure you uncheck those that you will not be using. Note that the **Search Service Application** and the **User Profile Service Application** should be left unchecked in this exercise because you will be creating instances of these service application in later labs.



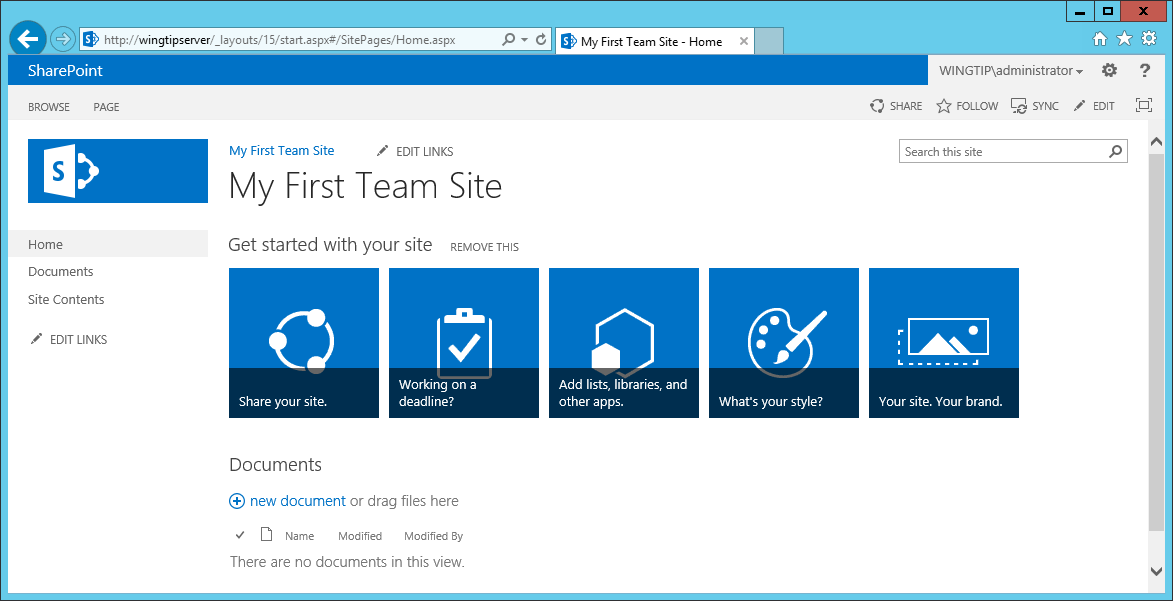
1. Click the **Next** button. At this point the wizard will begin to do its work. This might be a good time to take a 5-10 minute break because the wizard will run for a while.
2. When the Farm Configuration Wizard finishes creating and configuring the farm's service applications, it then creates a new web application using a public URL based on the local web server's computer name which in this case is http://wingtipserver. After the farm configuration wizard has created this web application, it displays the **Create Site Collection** page that prompts you to create a new site collection at the root. Fill out this page by giving the new site a title of **My First Team Site** and a site template of **Team Site** as shown in the following screenshot.



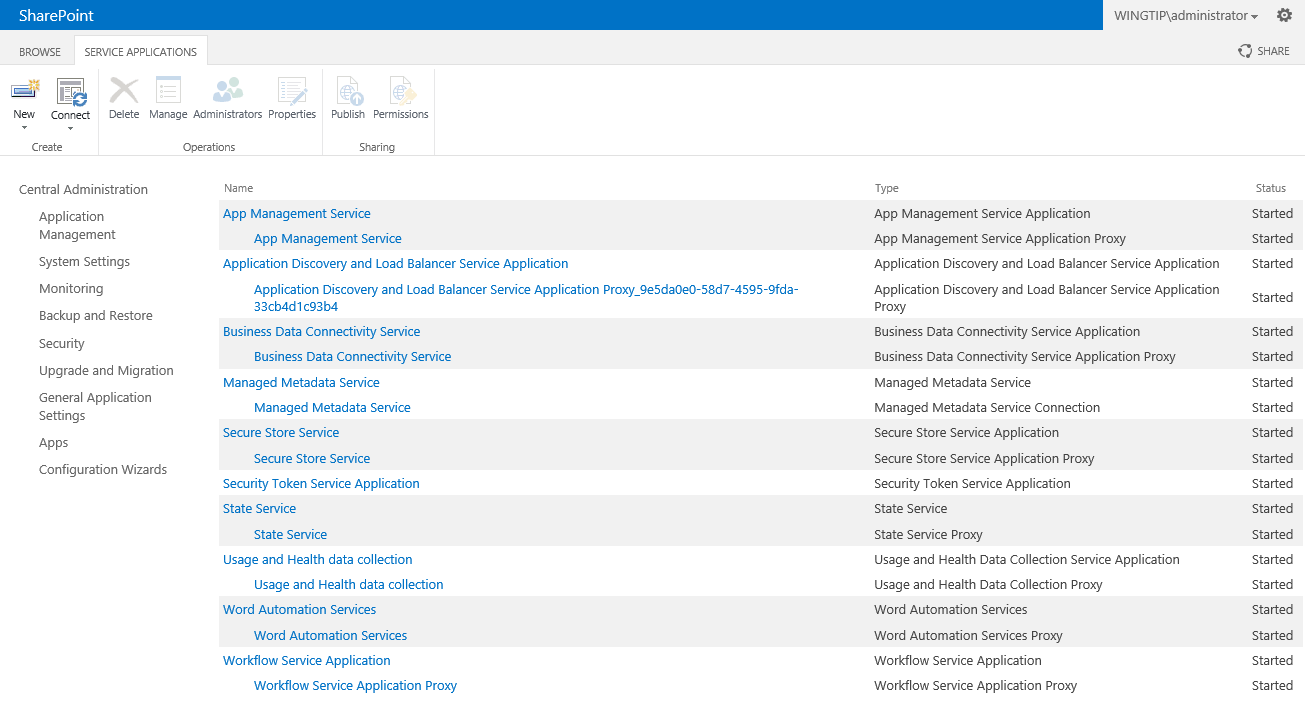
1. Click **OK** to have the Farm Configuration Wizard create the new site collection for at http://wingtipserver.
2. When the Farm Configuration Wizard completes, it displays the following page.



1. Click on the link to the site at **http://wingtipserver** to navigate to the team site created by the **Farm Creation Wizard**. It will likely take a few minutes for the home page of the site to display because SharePoint needs to launch and initialize the worker process for a new web application for the first time. Once the home page of the site is displayed, take a moment to navigate around to become familiar with the SharePoint 2013 user interface for standard team sites.



1. Now return to Central Administration and navigate to the **Manage Service Applications** page so that you can see the new service application instances that were created and configured by the Farm Configuration Wizard.



In this lab exercise you used the Farm Configuration Wizard to create a set of service application instances and new web application which hosts a new SharePoint 2013 Team site at its root.