



cutting through complexity

Saviynt Windows Server & Powershell Prerequisites

<Client>
KPMG Build

July 2022
Version: 1.0

Table of Contents

1	Document Control	3
1.1	Document Review & Feedback	3
1.2	Document Acceptance	3
1.3	Modification History	4
2	Document Purpose	5
3	Introduction	6
4	Connector Architecture	7
5	Win-PS Prerequisites	7
5.1	Preparing the Target System (Windows Server) for Integration.....	7
5.2	Assigning Identity of Application Pool(s) in IIS	27
5.3	Test HTTPS Connection	29
5.4	Connection Parameters	29
5.5	Install Exchange Office 365, AzureAD, MSOnline Modules	30
6.0	Azure AD Permissions for executing powershell from windows management server	31
6.1	App-only authentication for unattended scripts	31
1.1	Step 4: Attach the certificate to the Azure AD application.....	38
1.2	Step 5: Assign Azure AD roles to the application.....	40
2	Assigning Identity of Application Pool(s) in IIS	45

1 Document Control

1.1 Document Review & Feedback

An updated version of this document has been created and will be reviewed by the team members and stakeholders listed below. The feedback obtained from their review will be incorporated.

Manjunath Madiraju Architect	07/20/2022 Date	Architect <Title>	07/20/2022 Date
Security Architect	Date	<Title>	Date
<Title>	Date	<Title>	Date

1.2 Document Acceptance

Representative Approvers

By signing this document, you confirm that you have read, reviewed, and approved the contents of this deliverable.

AGREED TO AND ACCEPTED BY:

Client Project Management

By: _____

Name: _____

Title: _____

Date: _____

AGREED TO AND ACCEPTED BY:

Vendor Project Management

By: _____

Name: _____

Title: _____

Date: _____

1.3 Modification History

Change Date	Author	Version Modified	Description of Changes
July 20, 2022	Manjunath Madiraju	1.0	Initial Creation

2 Document Purpose

Enabling access to technology resources in a secure and efficient manner is at the core of a strong cyber security program. An organization must provide its workforce (employees, contractors and business partners) with the required access to securely enable business operations and collaboration. The purpose of this document is to provide Windows Server Prerequisites for <Client>, and execute powershell scripts.

3 Introduction

- The **Win-PS Connector** helps you to manage provisioning and deprovisioning of accounts as well as granting and revoking access to accounts on the Windows Server and the utilities installed on the server such as Active Directory (AD) Domain Services.
- EIC** communicates with the Windows server or its utilities (example, AD Domain Services) through an application developed by Saviynt named, **SaviyntApp**. The SaviyntApp invokes a PowerShell (PS) session on the Windows Server and executes PS scripts.

Supported Features:

Feature	Supported Operation	Supported Features
Provisioning and De-provisioning	Manage Accounts	Supports the following provisioning and de-provisioning operations of O365 Mailbox and Licenses.
	Change Password	Supports provisioning and de-provisioning password change from EIC.

Supported Operating System or Platform Versions

OS/Platform	Version
.Net Framework	4 and later
Windows Server	2012 R2 and later
Windows Internet Information Services (IIS) Server	<ul style="list-style-type: none"> Windows Server 2012 R2 or later IIS Version 7 or later installed with the URL Rewrite module Note: If the URL Rewrite module is not installed, a 404 error is reported when you try to add outbound rules to manage links.
SaviyntApp or Saviynt PowerShell Agent Package	

Note: You must have the login credentials with administrator rights for hosting the Windows server and the ability to execute desired PowerShell scripts or commands.

Supported Hardware Specifications:

Specifications	Value

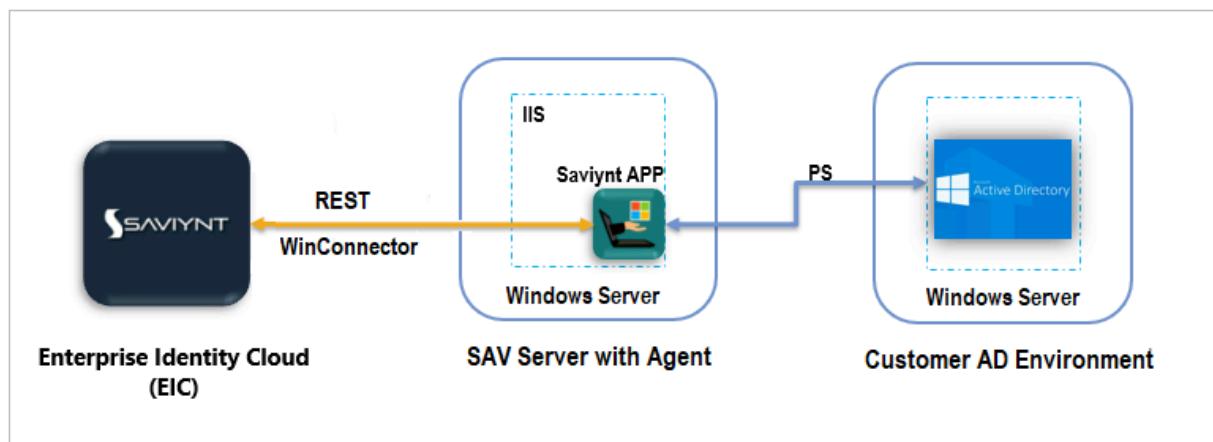
CPU	4 core 64-bit processor
RAM	16 GB
Storage	100 GB

4 Connector Architecture

By default, the connector is configured to run in provisioning mode.

The following architecture diagram illustrates the Win-PS Connector architecture and communication with Windows Server via SaviyntApp (which is a Saviynt application installed on the Windows server). The connector is used for provisioning and de-provisioning of data from EIC to the Windows Server using REST API.

The SaviyntApp invokes a PowerShell session on the customer's Windows Server and connects to the utility services such as SharePoint, Office 365, or AD Domain Services. This PowerShell session is used to administer such utilities in the Windows Server.



5 Win-PS Prerequisites

5.1 Preparing the Target System (Windows Server) for Integration

1. Installing Windows Internet Information Services Server

You can install IIS in two ways: GUI or Windows PowerShell command-line interface (CLI). If you do not have the GUI installed, use the PowerShell method. This method is relatively faster than the GUI method. For usage instructions, visit the [Microsoft Documentation](#) web site and search for the instructions for your operating system.

This section provides a detailed procedure for installing IIS web server version 10.0 on Windows Server 2016. For installing IIS on other supported Windows Server versions, see the corresponding instructions in the [Microsoft Documentation](#).

Note: You must have the login credentials with administrator rights for hosting Windows Server and the ability to execute desired PowerShell scripts/commands.

2. Installing IIS from Windows Server User Interface

To install IIS Server from the Windows Server User Interface, perform the following steps:

3. Click the **Start** icon.
4. Open **Server Manager**. Alternatively, type Server Manager from the search bar near the **Start** menu to open the Server Manager.

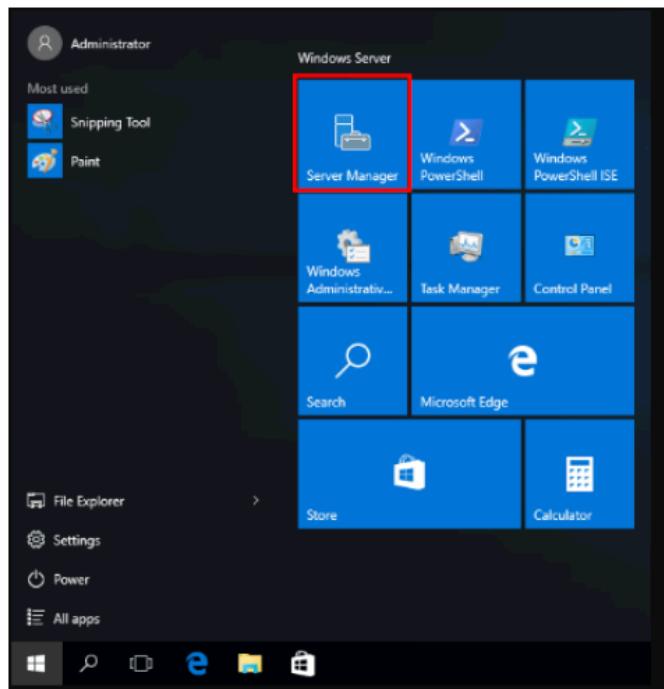


Figure: Selecting Server Manager option

5. In the **Server Manager Dashboard** page, click **Add roles and features**.

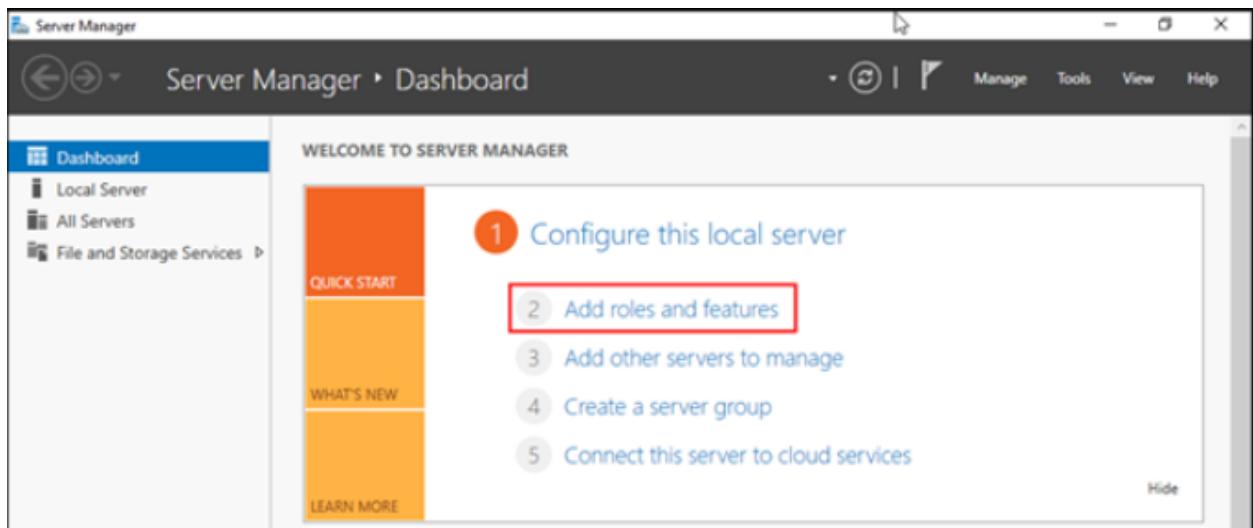


Figure: The Add Roles and Features Wizard

6. In the **Before You Begin** window, click **Next**.

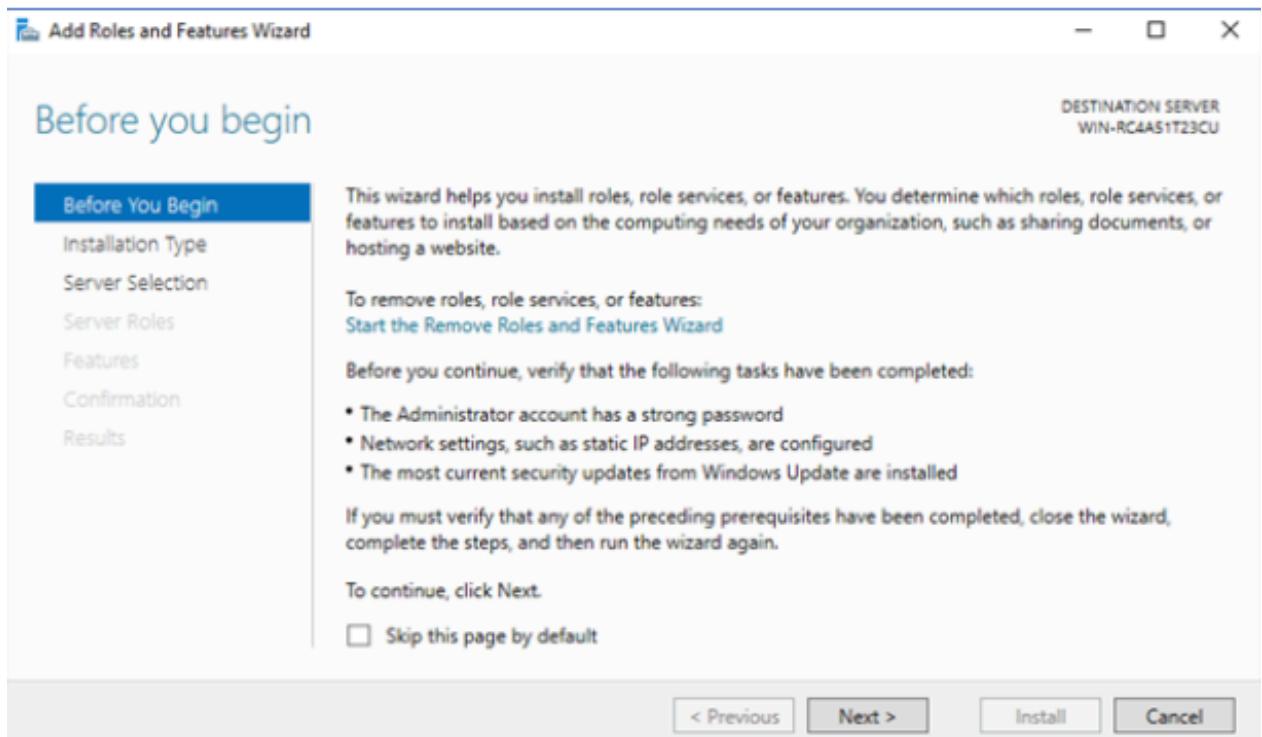


Figure: Before You Begin Window in Add Roles and Features Wizard

7. In the **Select installation type** window, select **Role-based or feature-based installation**, and click **Next**.

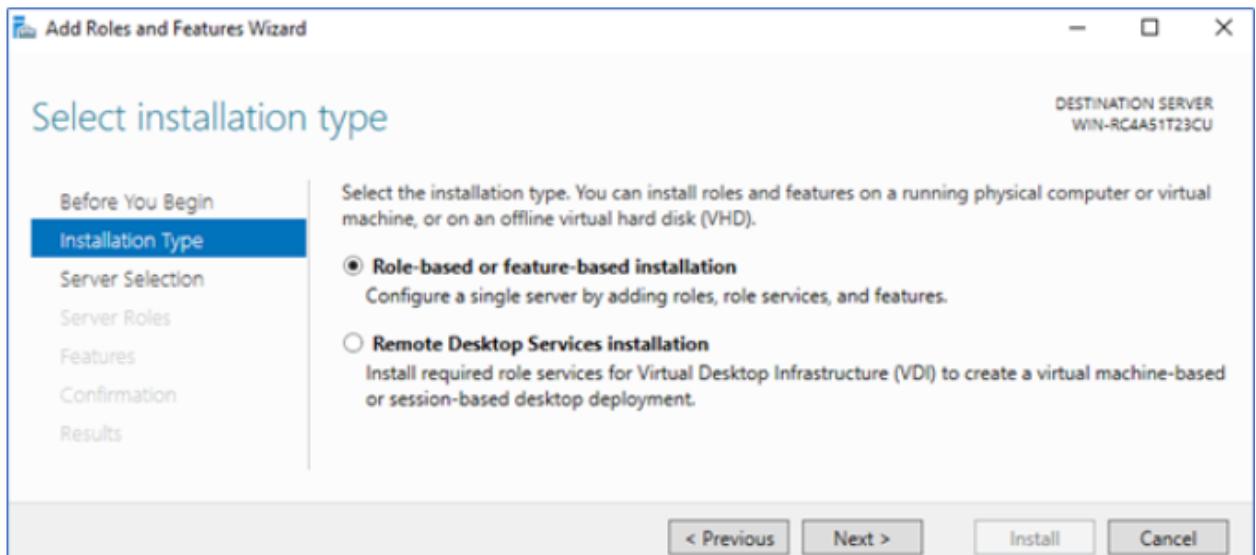


Figure: Select Installation Type Window

8. Select **Select a server from the server pool** with the current Windows Server selected, and click **Next**. Alternatively, you can select another server that you are managing from here.

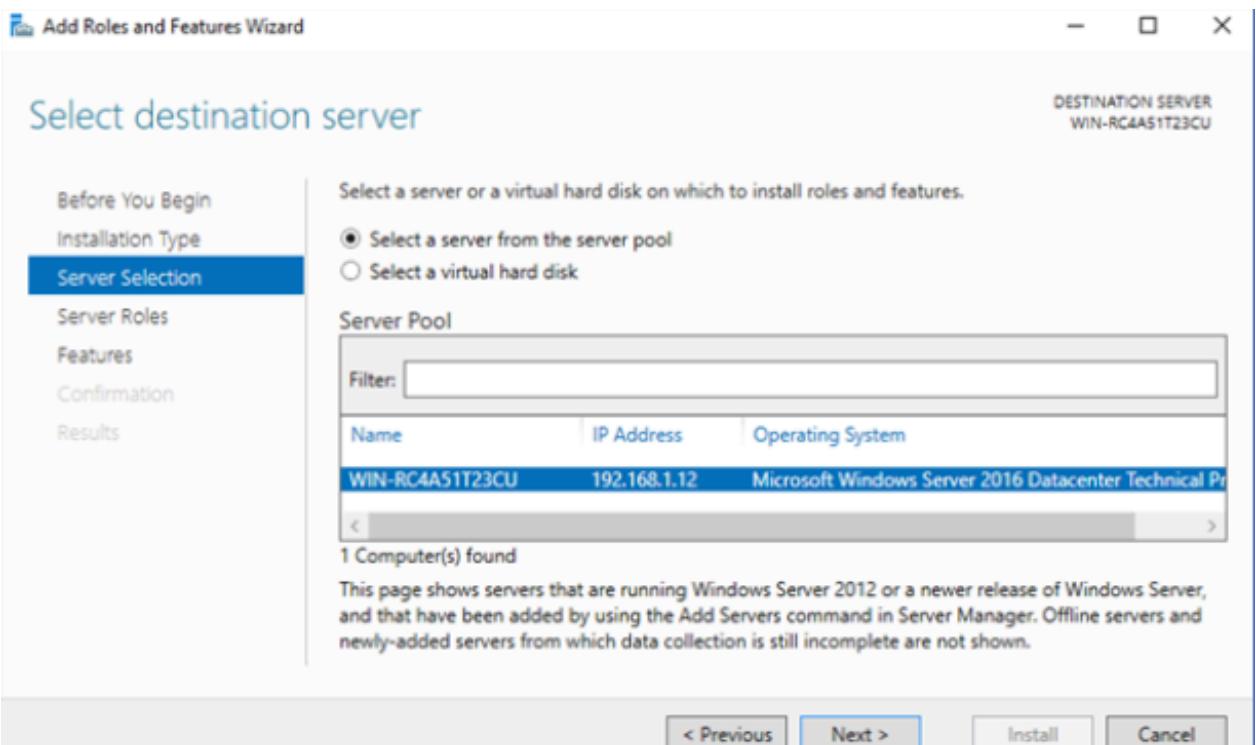


Figure: Selecting the Destination Server

9. In the **Select server roles** window, select **Web Server (IIS)**.

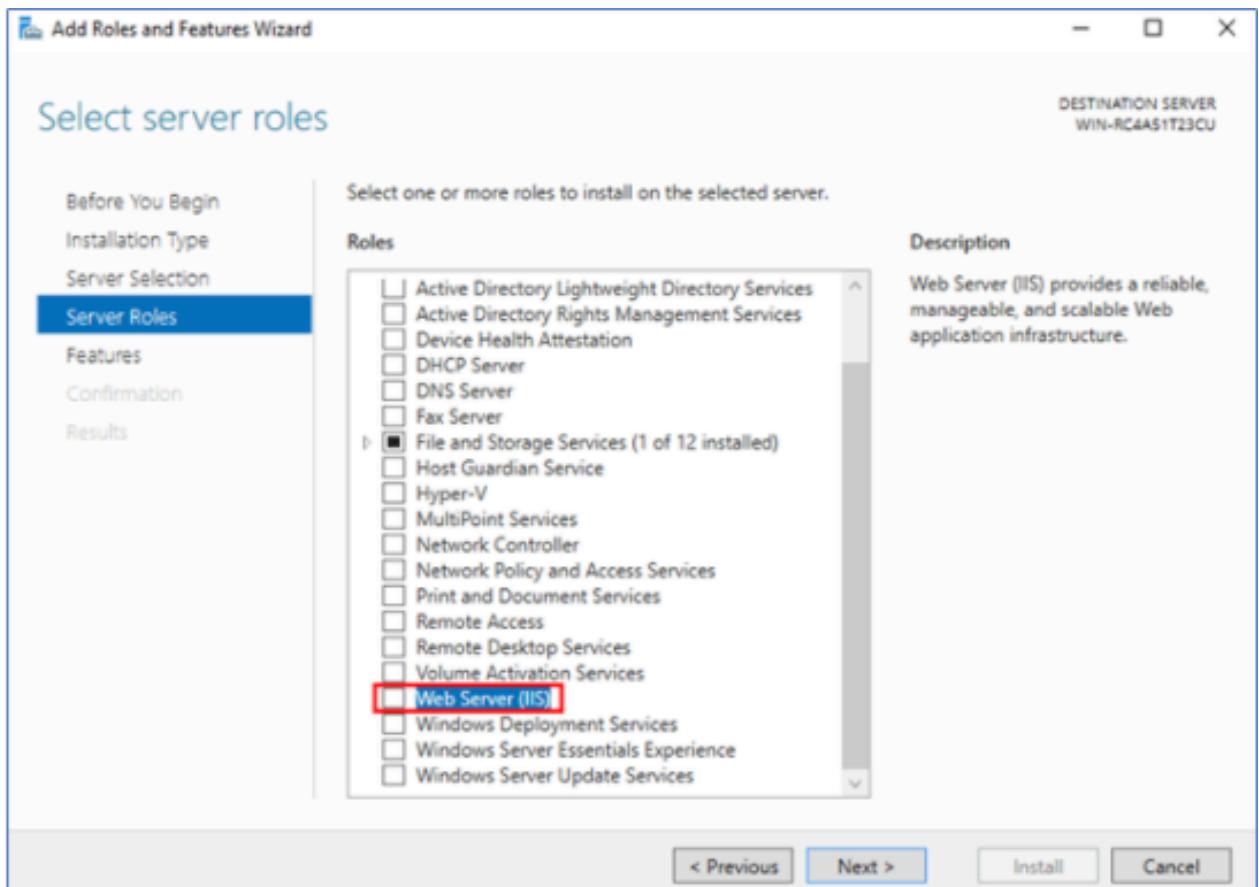


Figure: Selecting Web Server

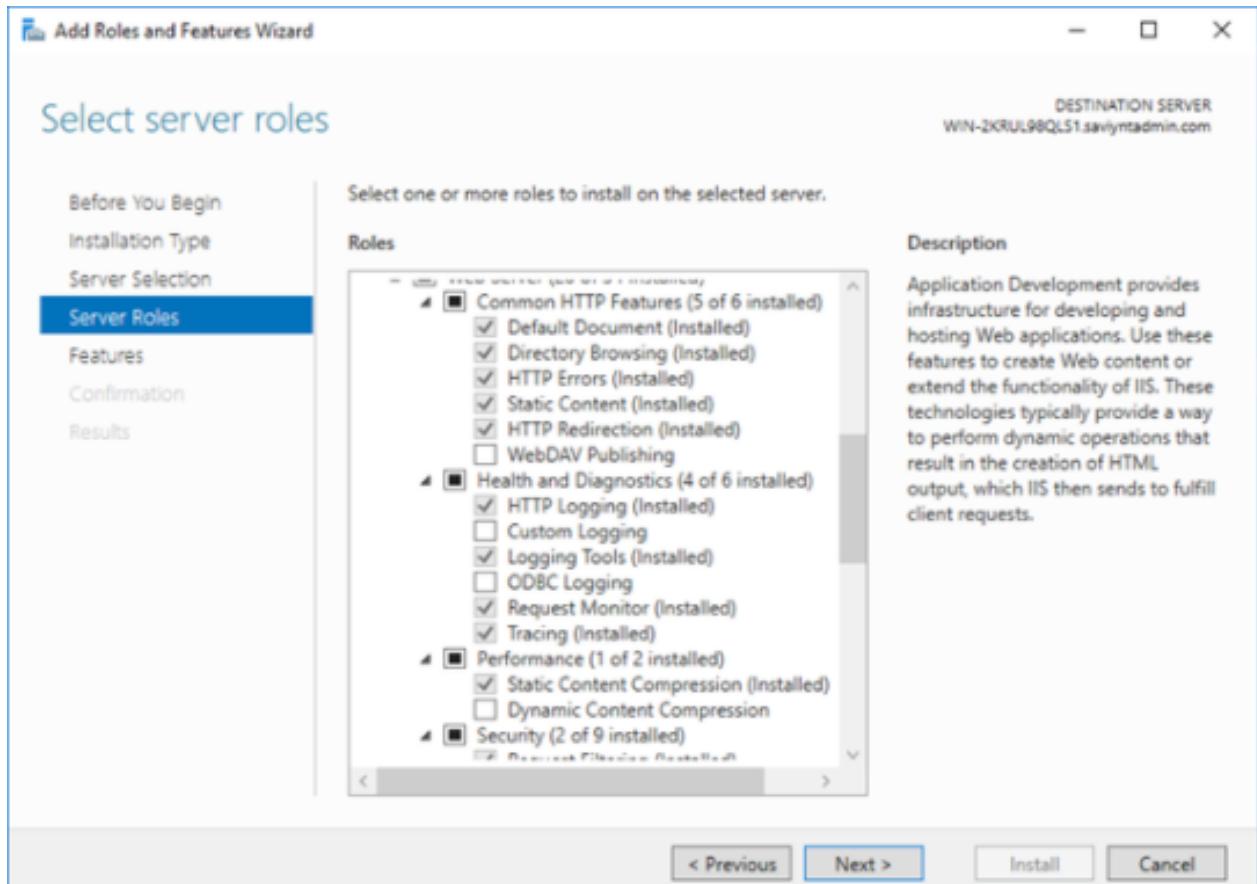


Figure: Selecting Server Roles

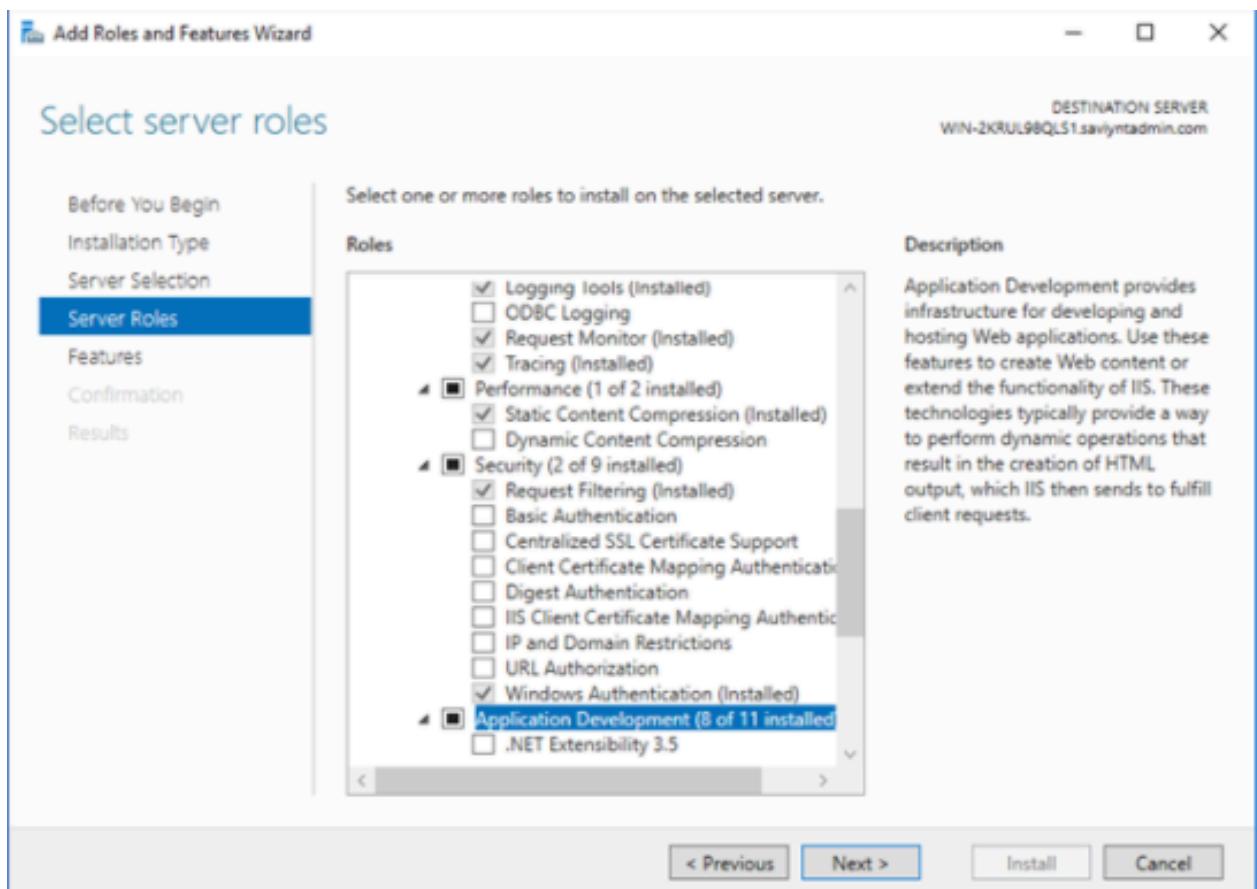


Figure: Select Server Roles Continued

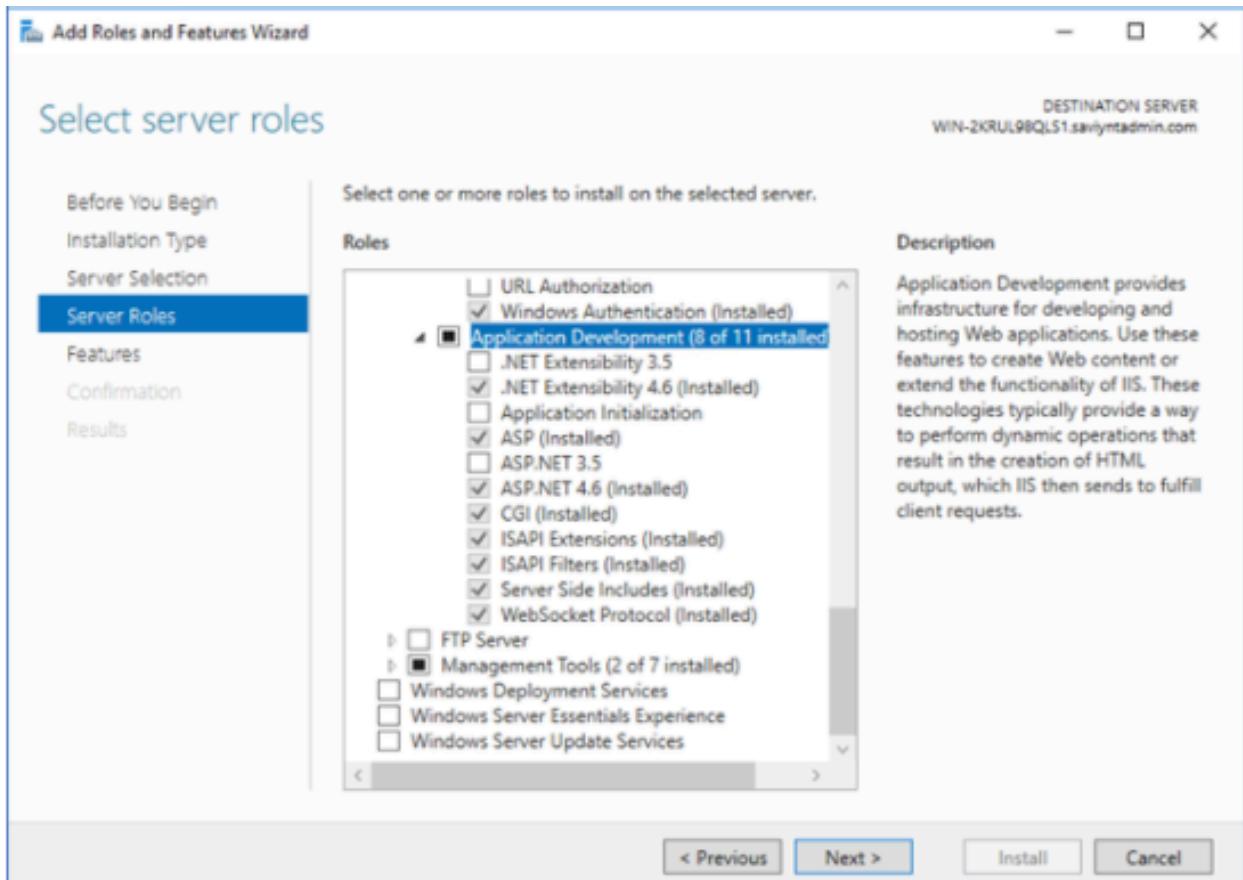


Figure: Select Server Roles Continued

10. Click **Next**.
11. In the **Add features that are required for Web server (IIS)** window, click **Add Features** to install additional features.

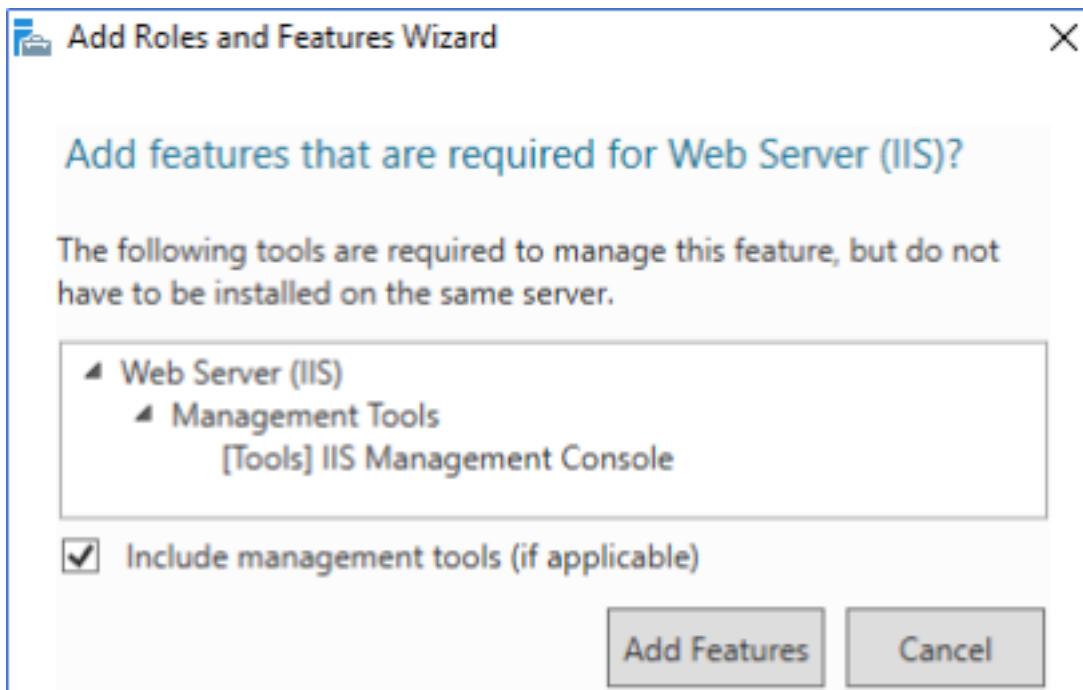


Figure: Add Features Option

12. In the **Select features** window, click **Next**.

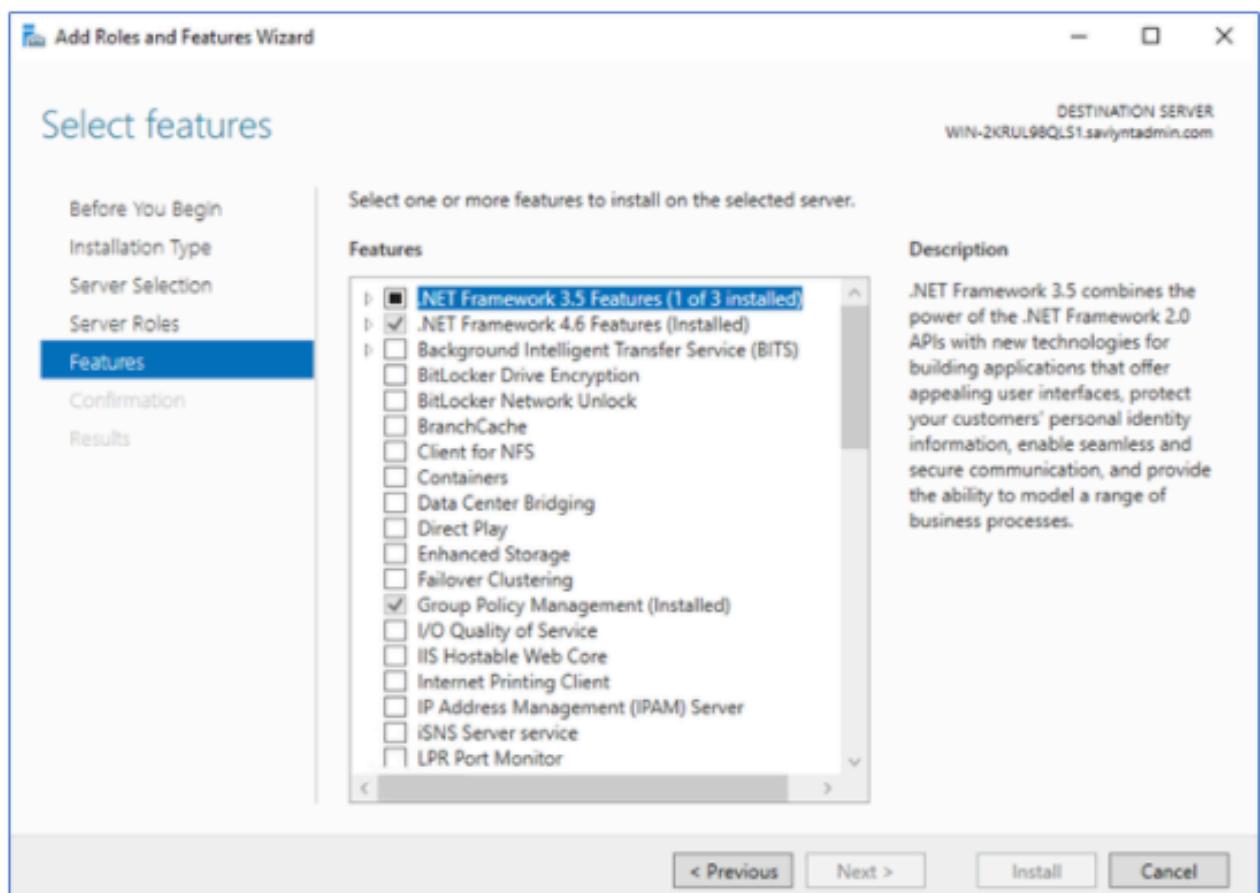


Figure: Select Features Window

13. In the **Web Server Role (IIS)** window, read the information provided about Web Server (IIS) role click **Next**.

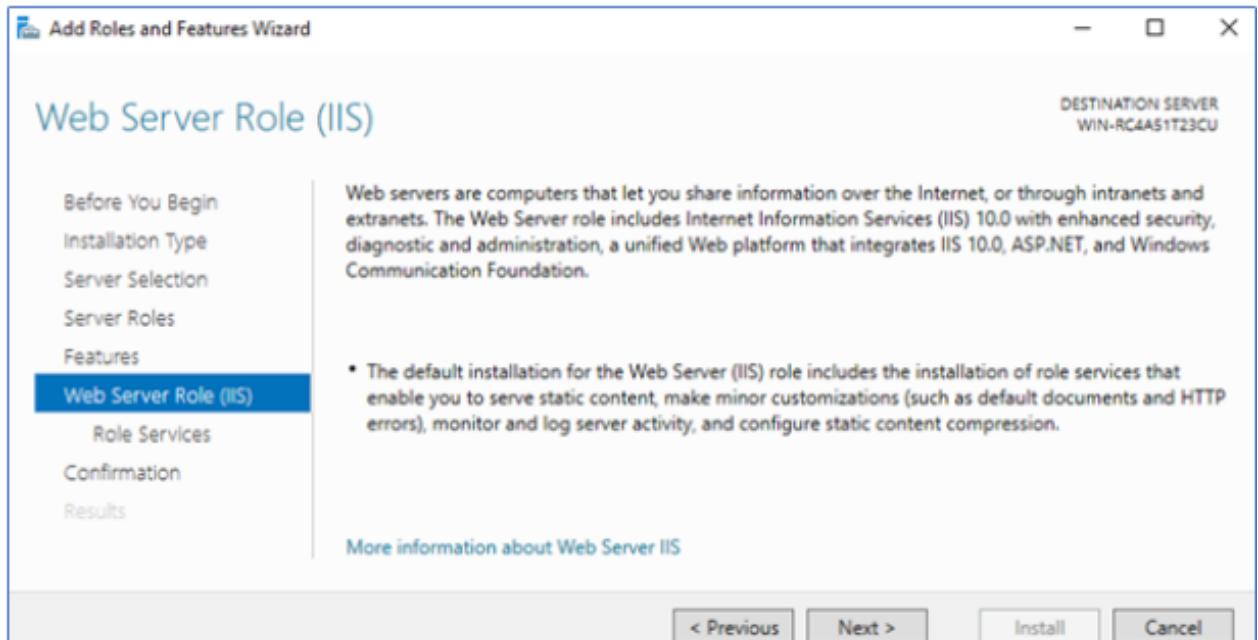


Figure: Web Server Role (IIS)

14. In the **Select role services** page, install additional services for IIS.
15. In the **Confirm installation selections** window, review the items to be installed and click **Install**. This installs the IIS web server.

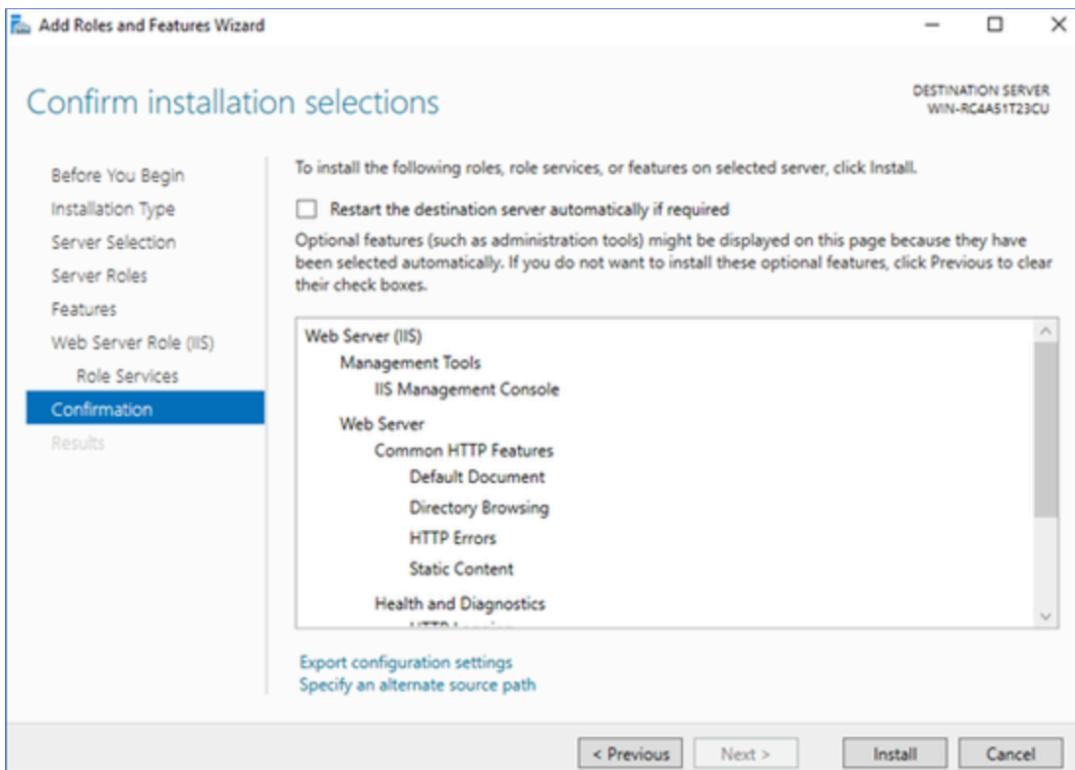


Figure: Confirming Installation Selections

Note: A reboot is not required in a standard IIS server installation; however, if you remove the role, a reboot is required.

16. Click **Close** after the installation is complete.

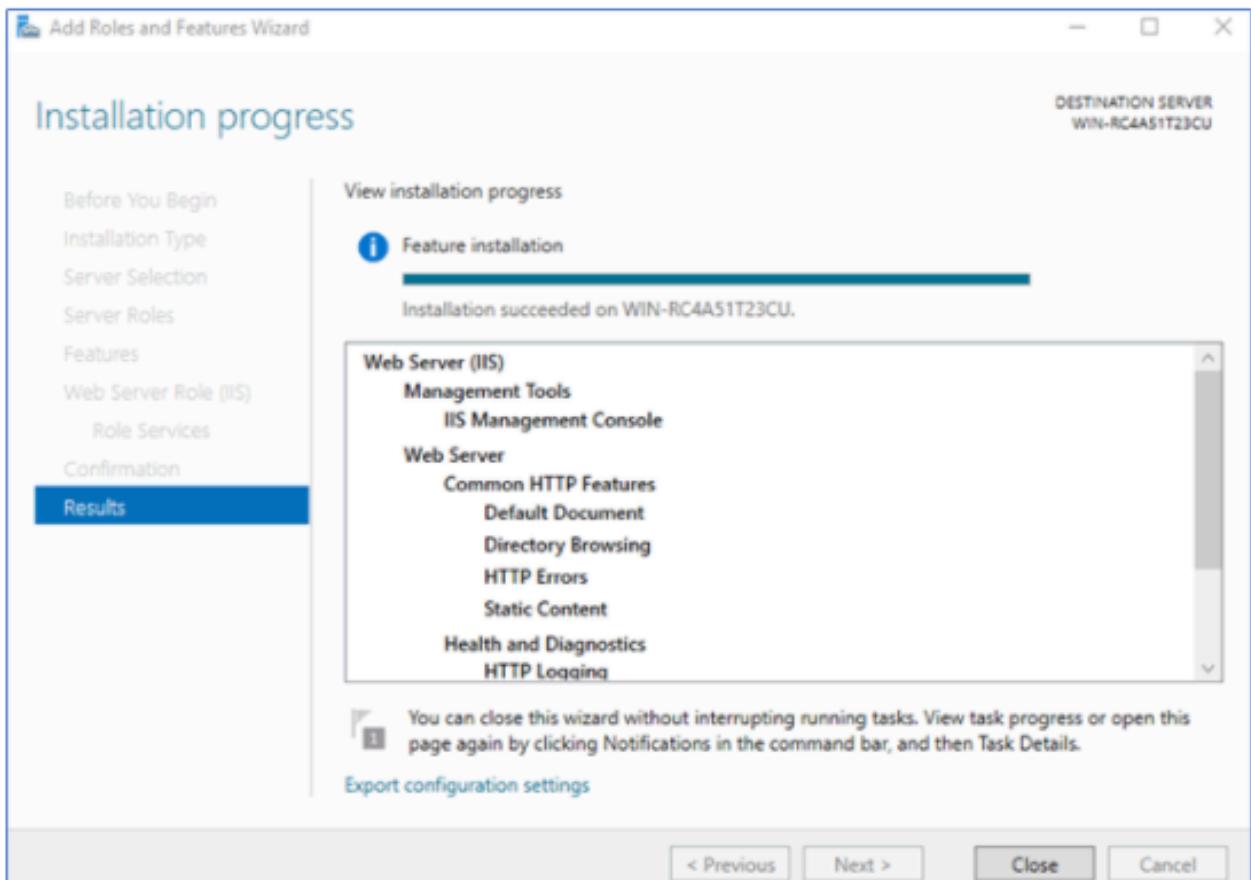


Figure: Installation Progress and Completion

Note: You can always add information in the **Select role services** window or the previous steps of the **Add Roles and Features** Wizard, if required.

IIS is installed and runs on port 80 (by default) with the firewall rule **World Wide Web Services (HTTP Traffic-In)** automatically enabled in the Windows Firewall.

After this installation is completed, install the **SaviyntApp**. For more information, see the *Downloading and Deploying the Saviynt App* section.

17. Downloading and Deploying the Saviynt App

This section explains how to download and deploy the Saviynt App in Windows IIS Server.

18. Downloading the Saviynt App

Download the [Saviynt App](#) (or Saviynt PowerShell Agent) to your system.

19. Deploying Saviynt App

This section explains the deployment of the Saviynt App within the Windows IIS Server. This process is essential for the communication between EIC to open a PowerShell session on the target system (Windows Server) that executes PowerShell scripts.

To deploy Saviynt App in the Windows IIS Server, perform the following steps:

1. Copy the Saviynt Agent.zip to C:\inetpub\wwwroot\SaviyntApp and Unzip to C:\inetpub\wwwroot\SaviyntApp\Saviynt Agent

This PC > Local Disk (C:) > inetpub > wwwroot > SaviyntApp >				
	Name	Date modified	Type	Size
	Saviynt Agent	1/29/2022 5:45 PM	File folder	
	Saviynt Agent	1/18/2022 11:30 PM	Compressed (zipp...)	5,149 KB
	web	1/29/2022 7:23 PM	CONFIG File	1 KB

This PC > Local Disk (C:) > inetpub > wwwroot > SaviyntApp > Saviynt Agent > WINPS >				
	Name	Date modified	Type	Size
ss	bin	1/29/2022 5:45 PM	File folder	
ds	Content	1/29/2022 5:45 PM	File folder	
nts	fonts	1/29/2022 5:45 PM	File folder	
	Scripts	1/29/2022 5:45 PM	File folder	
	Views	1/29/2022 5:45 PM	File folder	
	App_tracelog.svclog	1/29/2022 7:55 PM	SVCLOG File	27 KB
	favicon	1/29/2022 5:45 PM	Icon	32 KB
	Global.asax	1/29/2022 5:45 PM	ASAX File	1 KB
	packages	1/29/2022 5:45 PM	CONFIG File	3 KB
	Web	1/29/2022 6:36 PM	CONFIG File	16 KB

2. To add an application pool with .Net Framework 4 version.

1. In the IIS Manager, create an application pool and provide a name.
2. Select the framework version as **.NET Framework v4.0.30319**.
3. Select mode as **Integrated**.
4. Click **OK**.

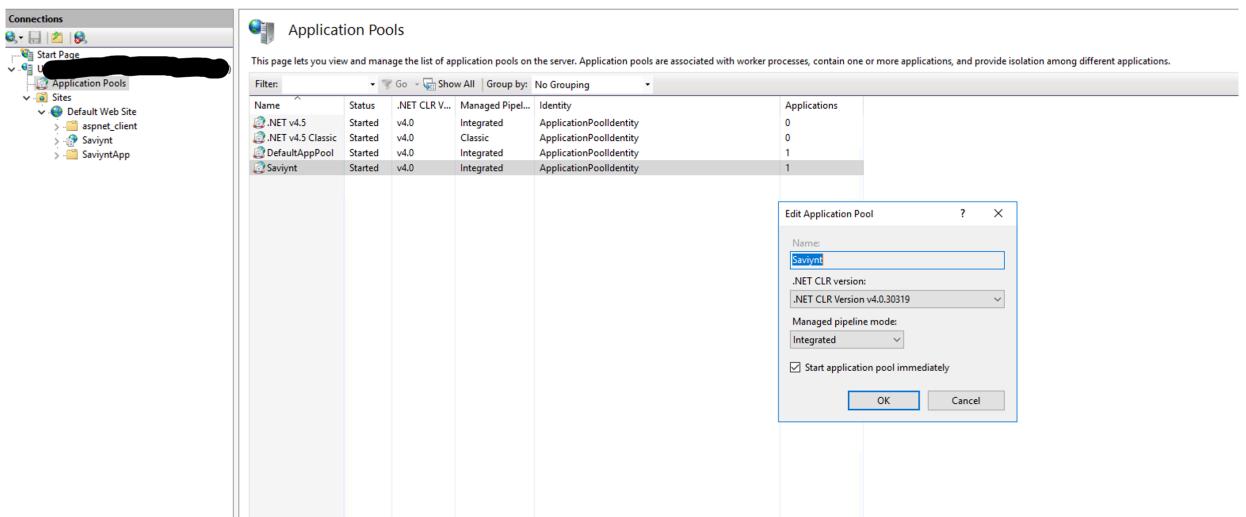


Figure: Adding Application Pool

3. Create an application under **Default Web Site**.

1. Provide a name under **Alias**.
2. Select the application pool you created in Step 1.
3. Select the physical path as local folder path where you plan to deploy the code, for example, C:\inetpub\wwwroot\SaviyntApp\Saviynt Agent\WINPS
You can create new folders in the **Browse For Folder** window.

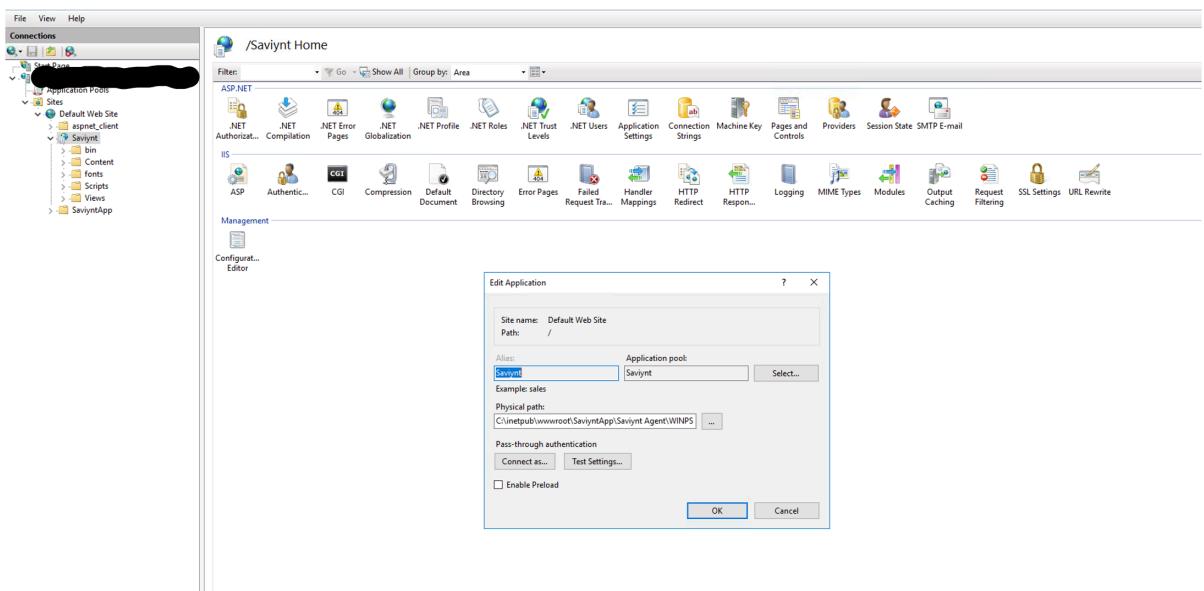


Figure: Creating an Application within the Default Web Site

4. Copy the Saviynt App package in the physical path mentioned in Step 2.

5. Install the SSL certificate for this website.
To install the SSL certificate, see [Installation of SSL/TLS Certificate In Microsoft IIS 7.](#)
6. Go to **Sites**, right-click **Default Web Site** and select **Edit Bindings**.
7. Select the SSL site with port 443 and click **Edit**.
8. In the **Edit Site Binding** page, view the SSL certificate and click **OK**.

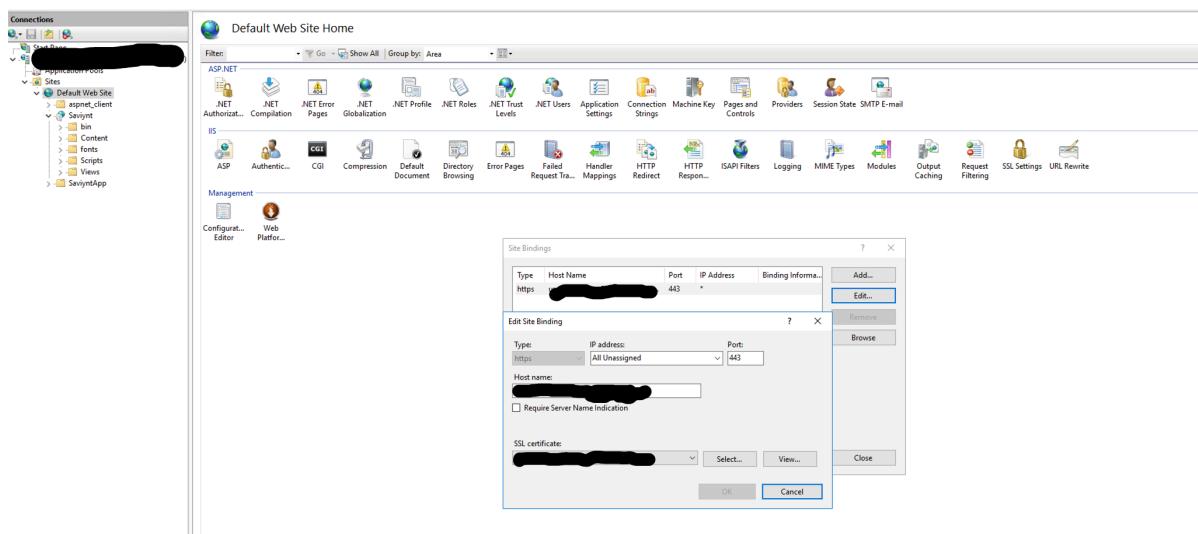
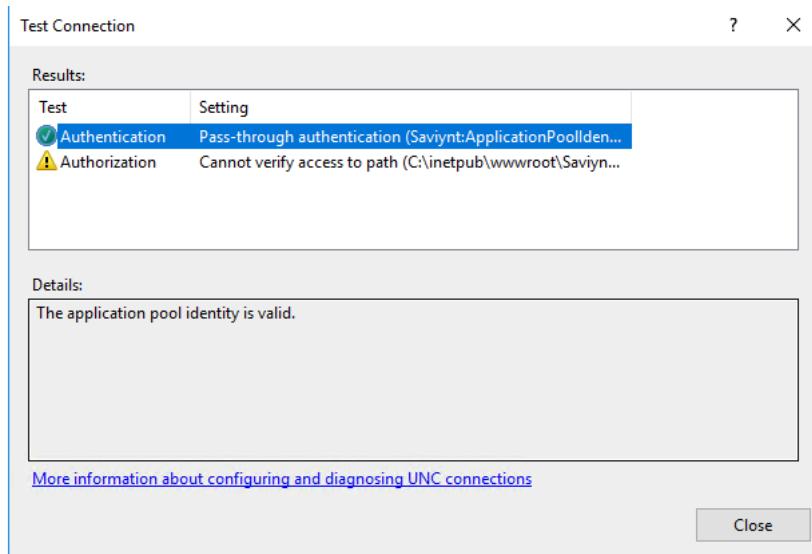


Figure: Edit Site Binding page

9. In the **Site Bindings** page, select the http binding and click **Remove** to delete the http binding for the application.
10. Go to **Sites > Default Web Site** and select **Basic Settings** under **Actions**.
11. In the **Edit Site** window, click **Connect as**.
12. In the **Connect As** window, select **Application user (pass-through authentication)** and click **OK**.
13. In the **Edit Site** window, click **Test Settings**.
14. In the **Test Connection** page, view the results and click **Close**.



15. Click **Restart** under **Actions** to restart the IIS Server.

20. Configuring Outbound Rules

You configure the outbound rules to remove server and [ASP.NET](#) information from response headers.

To configure outbound rules, perform the following steps:

1. Navigate to IIS Manager and select a computer name.

Figure: IIS Manager Home

2. Click on **URL Rewrite** under IIS to add outbound rules.

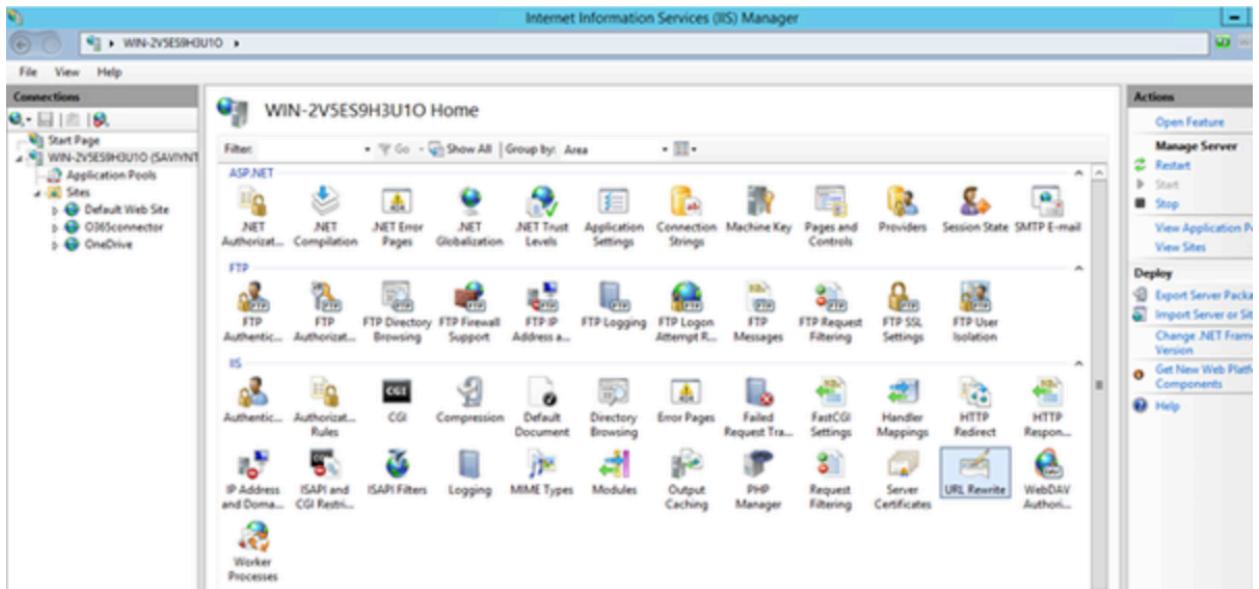


Figure: URL Rewrite option

3. Click **Add Rules** under **Actions**.

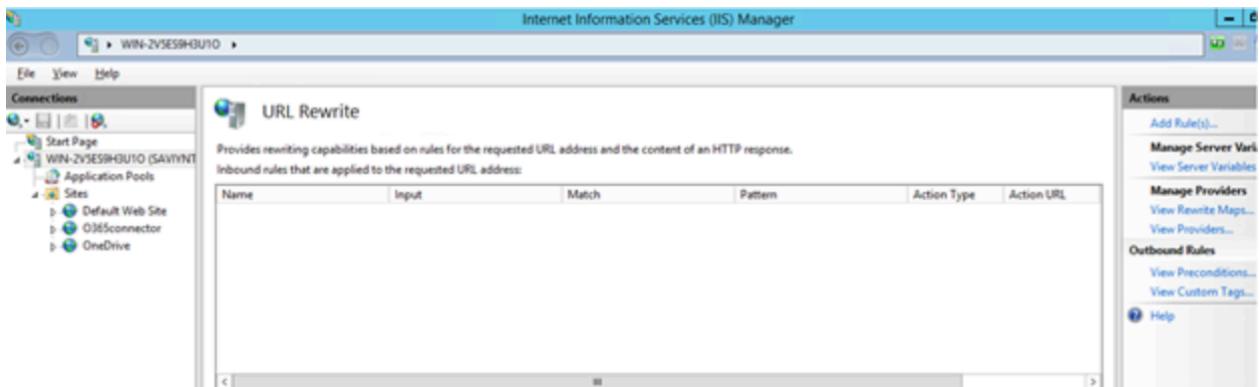


Figure: Add rules page

4. Select **Blank Rule** under **Outbound Rules** and click **Ok**.

5. Specify a name for the rule and specify the details as shown below:

1. Select precondition as **None**.
2. Select matching scope as **Server Variable**.
3. Select variable name as **RESPONSE_SERVER**.
4. Select a pattern as *****.
5. Select action type as **Rewrite**.

6. Select value as **Null**.
7. Enable the Replace existing server variable checkbox.

Figure: Edit outbound rules

6. Click **Apply** under **Actions**.
7. Repeat steps 3 to 5 for the following variable names:
 1. RESPONSE_X-POWERED-BY
 2. RESPONSE_X-ASPNET-VERSION
 3. RESPONSE_X-ASPNETMVC-VERSION
8. Click **Restart** under **Actions** to restart the IIS Server.

Note: In the web.config file add the following configuration under System.webServer to hide the physical path in the error response for the local users also.
 update <httpErrors errorMode="Custom" />

Name	Input	Match	Pattern	Action Type	Action Value	Stop Proc...	Entry Type
Saviynt OB URL Rewrite Rule (RESPONSE_SERVER)	RESPONSE_SERVER	Matches *	*	Rewrite	Null	False	Local
Saviynt OB URL Rewrite Rule (RESPONSE_X-POWERED-BY)	RESPONSE_X-POWERED-BY	Matches *	*	Rewrite	Null	False	Local
Saviynt OB URL Rewrite Rule (RESPONSE_X-ASPNET-VERSION)	RESPONSE_X-ASPNET-VERSION	Matches *	*	Rewrite	Null	False	Local
Saviynt OB URL Rewrite Rule (RESPONSE_X-ASPNETMVC-VERSION)	RESPONSE_X-ASPNETMVC-VERSION	Matches *	*	Rewrite	Null	False	Local

Name	Input	Match	Pattern	Action Type	Action Value	Stop Proc...	Entry Type
Saviynt OB URL Rewrite Rule (RESPONSE_SERVER)	RESPONSE_SERVER	Matches *	*	Rewrite	Null	False	Local
Saviynt OB URL Rewrite Rule (RESPONSE_X-POWERED-BY)	RESPONSE_X-POWERED-BY	Matches *	*	Rewrite	Null	False	Local
Saviynt OB URL Rewrite Rule (RESPONSE_X-ASPNET-VERSION)	RESPONSE_X-ASPNET-VERSION	Matches *	*	Rewrite	Null	False	Local
Saviynt OB URL Rewrite Rule (RESPONSE_X-ASPNETMVC-VERSION)	RESPONSE_X-ASPNETMVC-VERSION	Matches *	*	Rewrite	Null	False	Local

Server → Authentication → Forms Authentication enabled.

The screenshot shows the Windows Server IIS Manager interface. On the left, the 'Connections' pane displays the 'Start Page', 'Application Pools', and 'Sites'. Under 'Sites', the 'Default Web Site' is selected, which contains a sub-site named 'Saviynt'. The 'Saviynt' site has several folders: 'bin', 'Content', 'fonts', 'Scripts', 'Views', and 'SaviyntApp'. On the right, the 'Authentication' section is open, showing a table of authentication methods:

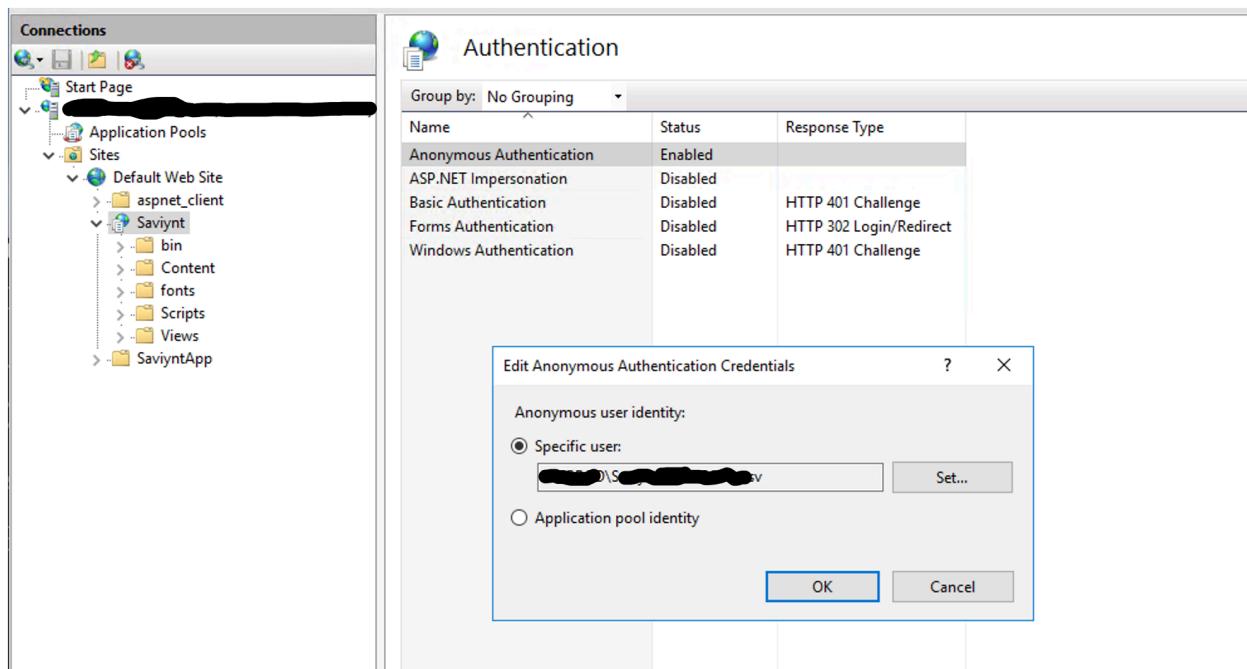
Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Enabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

Default Web Site → Authentication → Forms Authentication enabled.

The screenshot shows the Windows Server IIS Manager interface. On the left, the 'Connections' pane displays the 'Start Page', 'Application Pools', and 'Sites'. Under 'Sites', the 'Default Web Site' is selected, which contains a sub-site named 'Saviynt'. The 'Saviynt' site has several folders: 'bin', 'Content', 'fonts', 'Scripts', 'Views', and 'SaviyntApp'. On the right, the 'Authentication' section is open, showing a table of authentication methods:

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Enabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

Saviynt App → Authentication → Anonymous Authentication enabled.



Updated web.config file in C:\inetpub\wwwroot\SaviyntApp\Saviynt Agent\WINPS with below values:

```

<appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="PSSleepTime" value="6000" />
    <add key="PSTimeoutHours" value="0" />
    <add key="PSTimeoutMinutes" value="20" />
    <add key="PSTimeoutSeconds" value="6000" />
</appSettings>

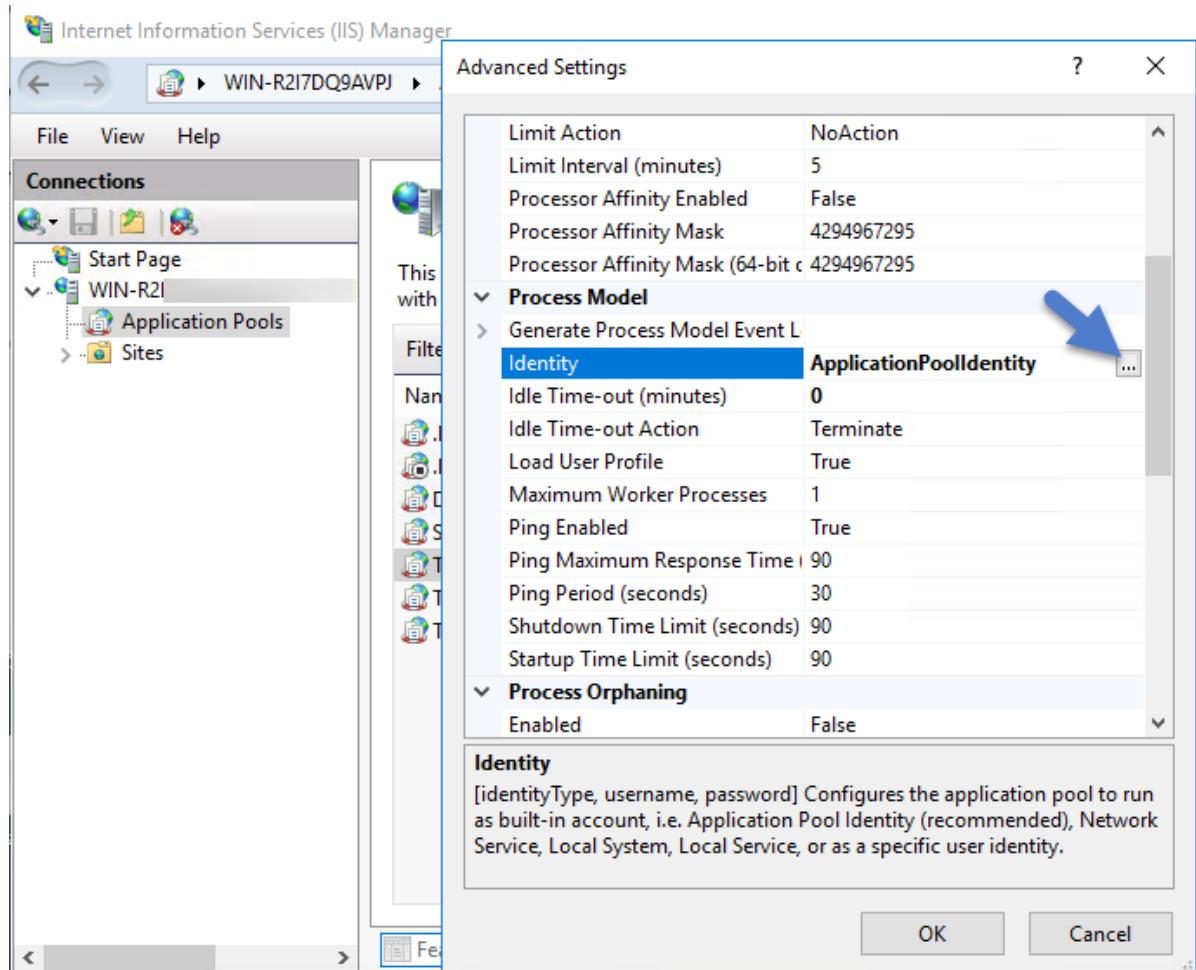
```

5.2 Assigning Identity of Application Pool(s) in IIS

Note: This is required so that application pool uses service account for actions like starting powershell.

You must have IIS installed on your web server before completing these steps.

1. Open IIS on your web server (**Search > inetmgr**)
2. Locate the application pool(s) that your Thycotic product is using, right-click **Advanced Settings...** then the **Identity** box in the "Process Model" section, click the three dots on the right of the box.



3. Select the **Custom Account** radio button, click **Set**, enter your service account's name and password, and click **OK**.
 - **Note:** For Privilege Manager you will need to perform this step for multiple application pools.

Application Pools

This page lets you view and manage the list of application pools on the server. Application pools are associated with worker processes, contain one or more applications, and provide

Name	Status	.NET CLR V...
.NET v4.5	Started	v4.0
.NET v4.5 Classic	Started	v4.0
DefaultAppPool	Started	v4.0
Saviynt	Started	v4.0

Advanced Settings

- (General)**
 - .NET CLR Version: v4.0
 - Enable 32-Bit Applications: False
 - Managed Pipeline Mode: Integrated
 - Name: Saviynt
 - Queue Length: 1000
 - Start Mode: OnDemand
- CPU**
 - Limit (percent): 0
 - Limit Action: NoAction
 - Limit Interval (minutes): 5
 - Processor Affinity Enabled: False
 - Processor Affinity Mask: 4294967295
 - Processor Affinity Mask (64-bit option): 4294967295

Application Pool Identity

ApplicationPoolIdentity
0
Terminate
True
1
True
90
30

Configures the application pool to run as built-in account, i.e. Application Pool Identity (recommended), or as a specific user identity.

Set Credentials

User name: [REDACTED]\S...
Password: [REDACTED]
Confirm password: [REDACTED]

OK Cancel

OK Cancel

Application Pools

Name	Status	.NET CLR V...
.NET v4.5	Started	v4.0
.NET v4.5 Classic	Started	v4.0
DefaultAppPool	Started	v4.0
Saviynt	Started	v4.0

Advanced Settings

- (General)**
 - .NET CLR Version: v4.0
 - Enable 32-Bit Applications: False
 - Managed Pipeline Mode: Integrated
 - Name: Saviynt
 - Queue Length: 1000
 - Start Mode: OnDemand
- CPU**
 - Limit (percent): 0
 - Limit Action: NoAction
 - Limit Interval (minutes): 5
 - Processor Affinity Enabled: False
 - Processor Affinity Mask: 4294967295
 - Processor Affinity Mask (64-bit option): 4294967295
- Process Model**
 - Generate Process Model Event Log Entry
 - Identity** [identityType, username, password] Configures the application pool to run as built-in account, i.e. Application Pool Identity (recommended), Network Service, Local System, Local Service, or as a specific user identity.

Idle Time-out (minutes): 0
Idle Time-out Action: Terminate
Load User Profile: True
Maximum Worker Processes: 1
Ping Enabled: True
Ping Maximum Response Time (seconds): 90
Ping Period (seconds): 30

OK Cancel

5.3 Test HTTPS Connection

1. Open Windows PowerShell in the Run dialog as administrator on the IIS Server to test locally

```

add-type @"
using System.Net;
using System.Security.Cryptography.X509Certificates;
public class TrustAllCertsPolicy : ICertificatePolicy {
public bool CheckValidationResult(
ServicePoint srvPoint, X509Certificate certificate,
WebRequest request, int certificateProblem) {
return true;
}
}
"@
[System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy

$Body = @{
SCRIPT = "Test-NetConnection <hostname.preprod.net>"}

Invoke-RestMethod https://<hostname.preprod.net>/Saviynt/PS/ExecutePSScript -Method Post -Body $Body

```

5.4 Connection Parameters

The connector uses the following connection parameters to connect to the target application:

Parameter	Description
URL	<p>Specify the path to invoke SaviyntApp deployed in the Windows IIS Server</p> <p>Syntax: http://<hostname>/<ApplicationName>/PS/ExecutePSScript</p> <p>Example: http://WIN-2KRUL98QLS1/FIMDemo/PS/ExecutePSScript</p> <p>https://hostname.preprod.net:443/Saviynt/PS/ExecutePSScript</p>
USERNAME	<p>Specify the username for the Windows IIS Server authentication.</p> <p>Eg: PREPROD\TestAccount-PSConn.sv</p>

PASSWORD	Specify the password to connect to the Windows IIS Server.
TESTCONNECTIONJSON	<pre>{ "TESTCOMMAND": "SCRIPT=Test-NetConnection <hostname>", "RESPONSE": { "OBJECTTOREAD": "get(0).PingSucceeded", "SUCCESSVALUE": "TRUE" } }</pre>

5.5 Install Exchange Office 365, AzureAD, MSOnline Modules

The connector uses the following connection

```
$TLS12Protocol = [System.Net.SecurityProtocolType] 'Ssl3 , Tls12'
[System.Net.ServicePointManager]::SecurityProtocol = $TLS12Protocol
Register-PSRepository -Default
Install-Module PowerShellGet -Force
Get-PSRepository
Get-InstalledModule
Install-Module AzureAd
Install-Module MSOnline
Install-Module -Name ExchangeOnlineManagement
Get-InstalledModule

Find-Module ExchangeOnlineManagement -AllVersions -AllowPrerelease
# Install latest EXO v2 for Compliance
Install-Module -Name ExchangeOnlineManagement -RequiredVersion 2.0.6-Preview6 -
AllowPrerelease
```

```
PS C:\Windows\system32> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
PS C:\Windows\system32> Install-Module -Name ExchangeOnlineManagement
Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from 'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
```

```
PS C:\Windows\system32> $TLS12Protocol = [System.Net.SecurityProtocolType] 'Ssl3 , Tls12'
[System.Net.ServicePointManager]::SecurityProtocol = $TLS12Protocol
PS C:\Windows\system32> Get-InstalledModule
Version Name Repository Description
----- -- -- -----
2.0.2.140 AzureAD PSGallery Azure Active Directory V2 General Availability Module...
2.0.6-Preview6 ExchangeOnlineManagement PSGallery This is a Public Preview release of Exchange Online PowerShell V2 module...
1.4.83.66 Microsoft PSGallery Microsoft Azure PowerShell Direct for Windows PowerShell
1.4.83.66 PackageManagement PSGallery PackageManagement (a.k.a. OneGet) is a new way to discover and install software packages from around the web...
2.2.5 PowerShellGet PSGallery PowerShell module with commands for discovering, installing, updating and publishing the PowerShell artifacts like Modules, DSC Resources, Role Capabilities a...
```

6.0 Azure AD Permissions for executing powershell from windows management server

6.1 App-only authentication for unattended scripts

Step-1: Creating a New Azure AD Application via the Azure UI

If you prefer to use the Azure UI to create the application, you can follow these steps to create a new Azure AD application:

1. In the Azure Active Directory admin center, go to Azure Active Directory

2. Click App registration

3. Click New application registration

The screenshot shows the Azure Active Directory admin center interface. On the left, there's a sidebar with options like Dashboard, Azure Active Directory, Users and groups, and Enterprise applications. The main area is titled 'App registrations'. At the top right, there are icons for notifications, settings, and help. A red box highlights the '+ New application registration' button. Below it, there's a search bar and a dropdown menu set to 'All apps'. The main table has columns for DISPLAY NAME, APPLICATION TYPE, and APPLICATION ID. One row is visible with the application ID '0b08e7e-4bd8-4a12-a0a2-c91...'. The bottom of the screen shows navigation links for BBB, Web app / API, and the application ID.

4. Enter the required details and click Create

This screenshot shows a 'Create' dialog box overlaid on the Azure Active Directory admin center. The dialog has three fields: 'Name' (set to 'TestAzureAPP'), 'Application type' (set to 'Web app / API'), and 'Sign-on URL' (set to 'https://localhost:8080'). A red box highlights the 'Create' button at the bottom of the dialog.

5. Retrieve the Application ID (Client ID) and Key (Secret Key)

1. Open the application

This screenshot shows the 'App registrations' screen again. It lists several applications, with 'TestAzureAPP' highlighted by a red box. To the right of 'TestAzureAPP', its application ID '980b7cfa-8736-4514-a44d-815...' is visible. The rest of the screen includes the sidebar with various Azure services and the standard admin center header.

2. Go to keys

3. Provide a key description and validity duration and click Save

4. Copy the displayed key. Note that the key will be displayed only once.

5. Copy the Application Id (Client ID)

The screenshot shows the Azure Active Directory admin center interface. On the left, there's a sidebar with options like Dashboard, Azure Active Directory, Users and groups, and Enterprise applications. The main area is titled 'TestAzureAPP' and shows details such as Display name (TestAzureAPP), Application type (Web app / API), and Home page (https://localhost:8090). A red box highlights the 'Application ID' field, which contains the value '980bh7csa:3736-4514-a4d-3151057cd066'. To the right, there's a 'Settings' pane with sections for General (Properties, Reply URLs, Owners), API Access (Required permissions, Keys), and Troubleshooting + Support (Troubleshoot, New support request).

Step 2: Assign API permissions to the application

Note

The procedures in this section replace any default permissions that were automatically configured for the new app. The app doesn't need the default permissions that were replaced.

1. On the app page under **Management**, select **Manifest**.

The screenshot shows the Microsoft identity platform Manifest page for an application named 'ExO PowerShell CBA'. The navigation bar includes links for Home, Contoso, and the current page, 'Manifest'. Below the navigation, there are sections for 'Call APIs' (with icons for various Microsoft services) and 'Documentation' (links to Microsoft identity platform, Authentication scenarios, Authentication libraries, Code samples, Microsoft Graph, Glossary, and Help and Support). A note at the bottom says 'Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.' A blue button labeled 'View API permissions' is visible. At the bottom, there's a link 'Manifest' with a red box around it, and the text 'Sign in users in 5 minutes'.

2. On the **Manifest** page that opens, find the requiredResourceAccess entry (on or about line 44).

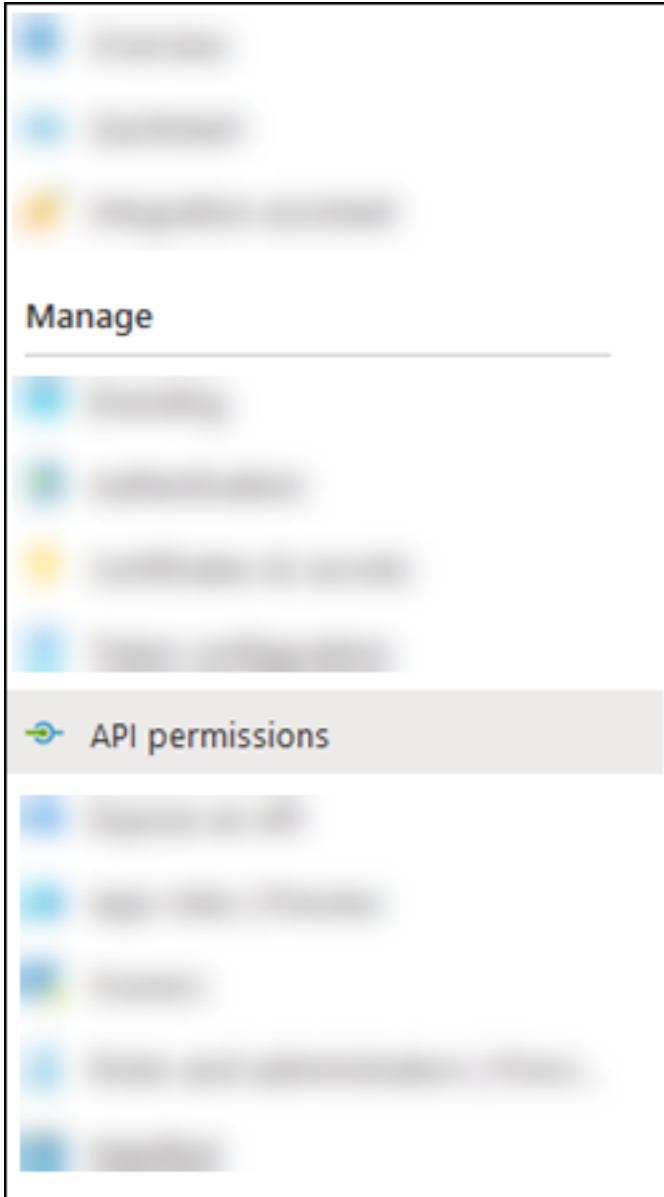
Modify the `resourceAppId`, `resourceAccess`, `id`, and `type` values as shown in the following code snippet:

JSONCopy

```
"requiredResourceAccess": [
  {
    "resourceAppId": "00000002-0000-0ff1-ce00-000000000000",
    "resourceAccess": [
      {
        "id": "dc50a0fb-09a3-484d-be87-e023b12c6440",
        "type": "Role"
      }
    ]
  }
],
```

When you're finished, click **Save**.

3. Still on the **Manifest** page, under **Management**, select **API permissions**.



On the **API permissions** page that opens, do the following steps:

- **API / Permissions name:** Verify the value **Exchange.ManageAsApp** is shown.
- **Status:** The current incorrect value is **Not granted for <Organization>**, and this value needs to be changed.

Add a permission		Grant admin consent for Contoso		
API / Permissions name	Type	Description	Admin consent required	Status
Office 365 Exchange Online (1)				
Exchange.ManageAsApp	Application	Manage Exchange As Application	Yes	⚠️ Not granted for Contoso

Select **Grant admin consent for <Organization>**, read the confirmation dialog that opens, and then click **Yes**.

The **Status** value should now be **Granted for <Organization>**.

The screenshot shows the Saviynt PowerShell interface with the 'API permissions' section selected. A message at the top right states: 'The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization.' Below this, a table lists permissions categorized by provider:

API / Permissions name	Type	Description	Admin consent requ...	Status
Device.ReadWrite.All	Application	Read and write devices	Yes	Granted for [REDACTED]
Directory.Read.All	Application	Read directory data	Yes	Granted for [REDACTED]
Directory.ReadWrite.All	Application	Read and write directory data	Yes	Granted for [REDACTED]
Domain.ReadWrite.All	Application	Read and write domains	Yes	Granted for [REDACTED]
Group.Read.All	Application	Read all groups	Yes	Granted for [REDACTED]
Group.ReadWrite.All	Delegated	Read and write all groups	Yes	Granted for [REDACTED]
Group.ReadWrite.All	Application	Read and write all groups	Yes	Granted for [REDACTED]
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	Granted for [REDACTED]
Member.Read.Hidden	Application	Read all hidden memberships	Yes	Granted for [REDACTED]
User.Read.All	Application	Read all users' full profiles	Yes	Granted for [REDACTED]
User.ReadBasic.All	Delegated	Read all users' basic profiles	No	Granted for [REDACTED]
User.ReadWrite.All	Application	Read and write all users' full profiles	Yes	Granted for [REDACTED]
Exchange.ManageAsApp	Application	Manage Exchange As Application	Yes	Granted for [REDACTED]
MailboxSettings.ReadWrite	Application	Read and write all user mailbox settings	Yes	Granted for [REDACTED]

To view and manage permissions and user consent, try [Enterprise applications](#).

4. Close the current **API permissions** page (not the browser tab) to return to the **App registrations** page. You'll use it in an upcoming step.

Step-3 Generate Certificate and Assign to the Application created in Azure for Saviynt

```
$cert = New-SelfSignedCertificate -Subject "CN={saviyntAADTestTenant}" -CertStoreLocation "Cert:\CurrentUser\My" -KeyExportPolicy Exportable -KeySpec Signature -KeyLength 2048 -KeyAlgorithm RSA -HashAlgorithm SHA256 ## Replace {certificateName}
```

```
Export-Certificate -Cert $cert -FilePath "C:\Saviynt\SaviyntDev\{saviyntAADTestTenant}.cer" ## Specify your preferred location and replace {certificateName}
```

```
$mypwd = ConvertTo-SecureString -String "cert-password-tosafeit" -Force -AsPlainText ## Replace {myPassword}
```

```
Export-PfxCertificate -Cert $cert -FilePath "C:\Saviynt\SaviyntDev\{saviyntAADPK}.pfx" -Password $mypwd ## Specify your preferred location and replace {privateKeyName}
```

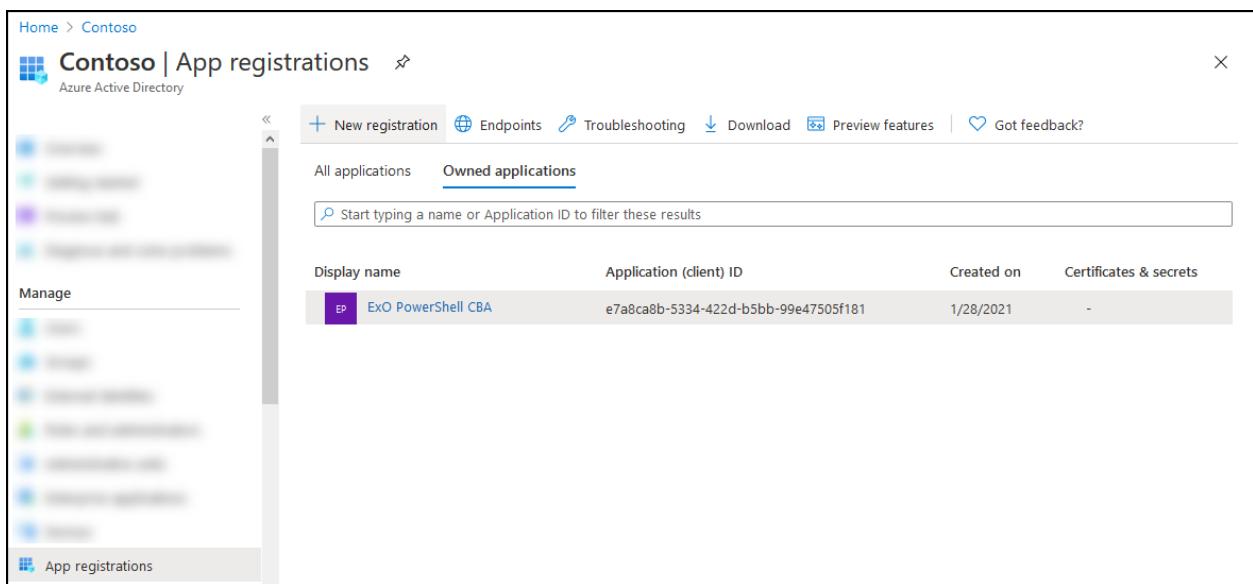
1.1 Step 4: Attach the certificate to the Azure AD application

After you register the certificate with your application, you can use the private key (.pfx file) or the thumbprint for authentication.

1. On the **Apps registration** page from the end of [Step 2](#), select your application.

If you need to get back to **Apps registration** page, do the following steps:

1. Open the Azure AD portal at <https://portal.azure.com/>.
2. Under **Manage Azure Active Directory**, click **View**.
3. Under **Manage**, select **App registrations**.



The screenshot shows the 'Contoso | App registrations' page in the Azure Active Directory portal. The left sidebar has 'Manage' and 'App registrations' sections. The main area has tabs for 'All applications' and 'Owned applications'. A search bar says 'Start typing a name or Application ID to filter these results'. Below is a table with columns: Display name, Application (client) ID, Created on, and Certificates & secrets. One row is shown: 'ExO PowerShell CBA' with client ID 'e7a8ca8b-5334-422d-b5bb-99e47505f181' and created on '1/28/2021'.

Display name	Application (client) ID	Created on	Certificates & secrets
ExO PowerShell CBA	e7a8ca8b-5334-422d-b5bb-99e47505f181	1/28/2021	-

2. On the application page that opens, under **Manage**, select **Certificates & secrets**.

The screenshot shows the 'ExO PowerShell CBA' interface. On the left, there's a sidebar with 'Manage' and a list of items, one of which is 'Certificates & secrets' (highlighted with a red box). The main area has sections for 'Call APIs' (with icons for various Microsoft services like Excel, SharePoint, OneDrive, etc.) and 'Documentation' (links to Microsoft identity platform, Authentication scenarios, etc.). A callout box says 'Sign in users in 5 minutes'.

3. On the **Certificates & secrets** page that opens, click **Upload certificate**.

The screenshot shows the 'Saviynt Powershell | Certificates & secrets' interface. The 'Certificates & secrets' tab is selected in the sidebar. A modal dialog is open, showing a table with one row of data:

Thumbprint	Description	Start date	Expires	Certificate ID
[REDACTED]	Saviynt AAD Test	2/2/2022	2/2/2023	[REDACTED]

In the dialog that opens, browse to the self-signed certificate (.cer file) that you created in [Step 3](#).

Upload certificate

Upload a certificate (public key) with one of the following file types: .cer, .pem, .crt

📁

Add **Cancel**

When you're finished, click **Add**.

The certificate is now shown in the **Certificates** section.

The screenshot shows the 'Certificates' section of the Azure portal. A newly uploaded certificate is listed with the following details:

Thumbprint	Start date	Expires	ID
84DAEAE5FFAFA90F566877A2DB2674C0222375BE	1/6/2021	1/6/2022	7ddbdcc5-4c15-4107-b7dc-262567c252b9

- Close the current **Certificates & secrets** page, and then the **App registrations** page to return to the main <https://portal.azure.com/> page. You'll use it in the next step.

1.2 Step 5: Assign Azure AD roles to the application

Azure AD has more than 50 admin roles available. The supported roles are described in the following table:

Role	Exchange Online PowerShell	User Management
User Administrator		✓
Exchange Administrator*	✓	

* The Exchange Administrator roles provide the required permissions for any task in Exchange Online PowerShell.

- * The User Administrator roles provide the required permissions for any task in Azure AD PowerShell.

The screenshot shows the 'User administrator | Assignments' page in the Azure portal. The left sidebar has 'Assignments' selected under 'Manage'. The main area shows a table with columns: Name, UserName, Type, and Scope. One row is visible for 'Saviynt Powershell'.

Name	UserName	Type	Scope
Saviynt Powershell	[REDACTED]	ServicePrincipal	Directory

The screenshot shows the 'Exchange administrator | Assignments' page in the Azure portal. The left sidebar has 'Assignments' selected under 'Manage'. The main area shows a table with columns: Name, UserName, Type, and Scope. One row is visible for 'Saviynt Powershell'.

Name	UserName	Type	Scope
Saviynt Powershell	[REDACTED]	ServicePrincipal	Directory

Security & Compliance Center PowerShell:

The screenshot shows the 'Compliance administrator | Assignments' page in the Azure portal. The left sidebar has 'Assignments' selected under 'Manage'. The main area shows a table with columns: Name, UserName, Type, and Scope. A message at the top states 'No role assignments found'. The '+ Add assignments' button is highlighted with a red box.

Name	UserName	Type	Scope
No role assignments found			

Microsoft Azure Search resources, services, and docs (G+) Home > [REDACTED] > Compliance administrator Manjunath.M

Compliance administrator | Assignments [REDACTED]

All roles

+ Add assignments X Remove assignments Download assignments Refresh Manage in PIM Got feedback?

X Diagnose and solve problems

Manage

Assignments Description

Activity

Bulk operation results

Troubleshooting + Support

New support request

Assignments

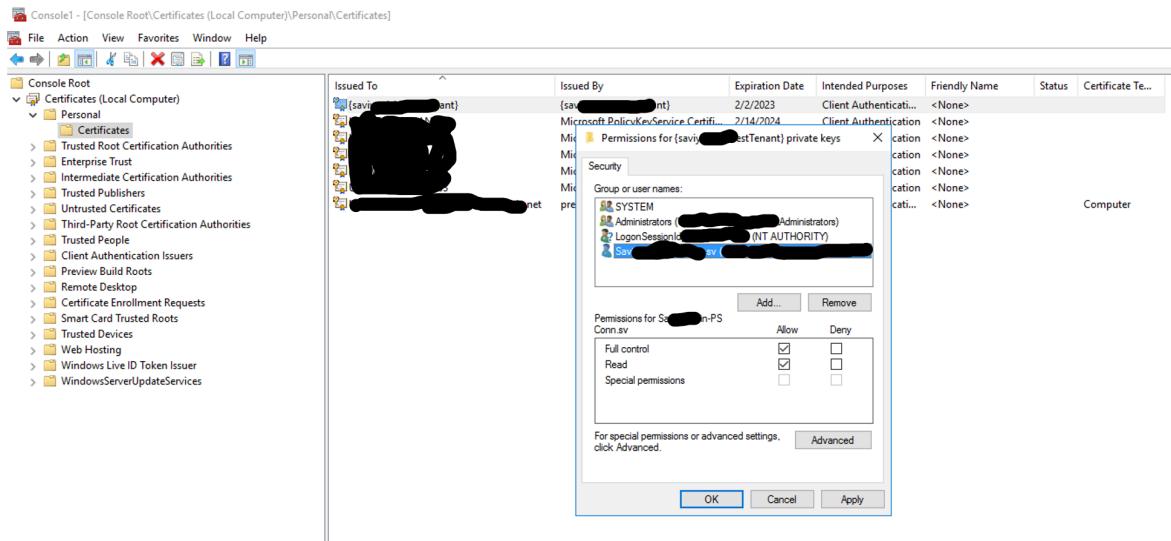
! You can also assign built-in roles to groups now. [Learn More](#)

Name	UserName	Type	Scope
<input type="checkbox"/> Saviynt Powershell	[REDACTED]	ServicePrincipal	Directory

Step-6: Allow ServiceAccount to read the certificate

Note: If this step is not performed, then we see error: Connect-AzureAD : One or more errors occurred. "KeySet does not exist".

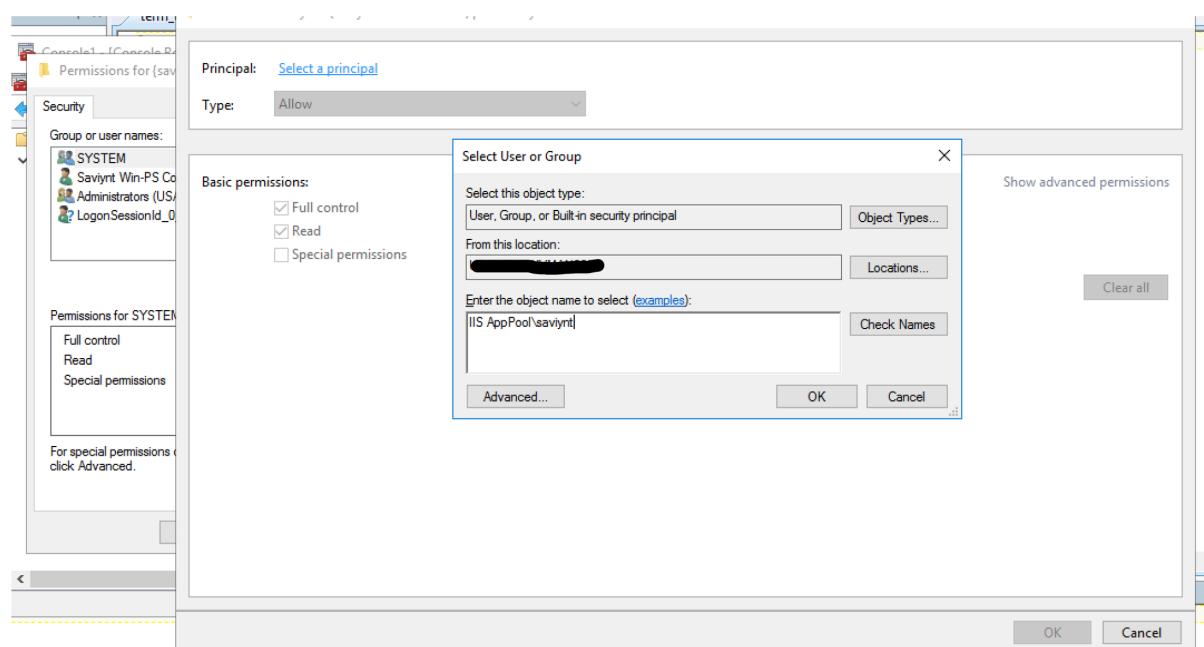
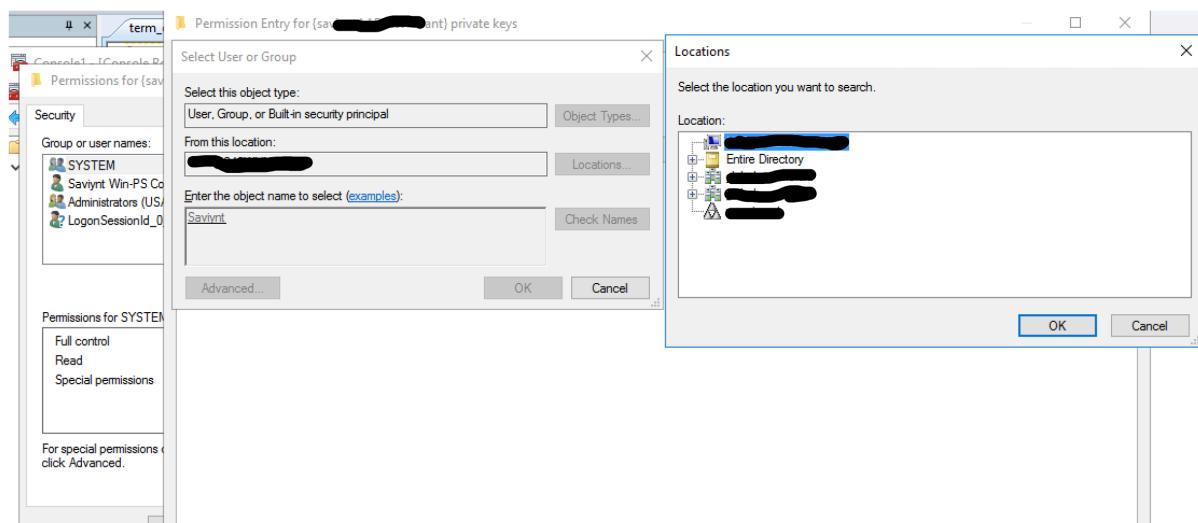
1. Open certificate snap-in in MMC: WIN + R -> type mmc -> File -> Add/Remove Snap-in -> Add Certificates (Computer Account).
 2. Find your certificate -> Right click and choose All Tasks/Manage Private Keys
 3. Grant Permission for SERVICE Account user

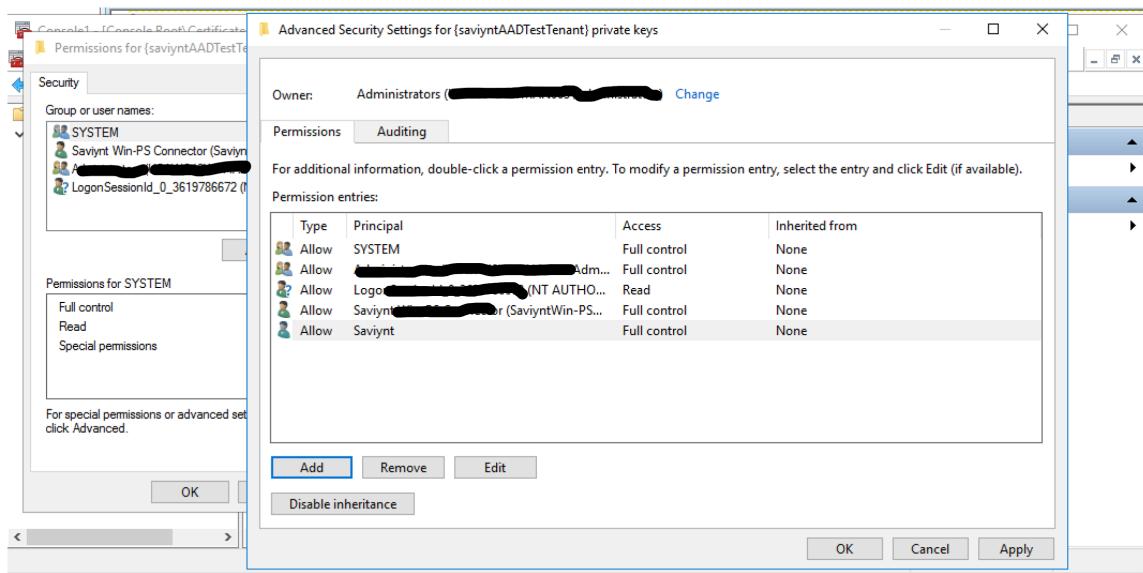


We should also provide permissions for Saviynt application pool to access the certificate.

1. Create / Purchase certificate. Make sure it has a private key.
 2. Import the certificate into the "Local Computer" account. Best to use Certificates MMC. **Make sure to check "Allow private key to be exported"**
 3. Based upon which, IIS 7.5 Application Pool's identity use one of the following.

- IIS 7.5 Website is running under ApplicationPoolIdentity.
- Using Certificates MMC,
 1. Select "From this Location" as that local server. By default it would be preprod or global domain.
 2. added "IIS AppPool\Saviynt" to Full Trust on certificate in "Local Computer\Personal". Saviynt is the "*AppPoolName*" of your application pool.





Also, in IIS Application pool, set Load user Profile to True.

Start Mode	OnDemand
CPU	
Limit (percent)	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit option)	4294967295
Process Model	
> Generate Process Model Event Log Entry	
Identity	ApplicationPoolIdentity
Idle Time-out (minutes)	0
Idle Time-out Action	Terminate
Load User Profile	True
Maximum Worker Processes	1
Ping Enabled	True
Ping Maximum Response Time (seconds)	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90

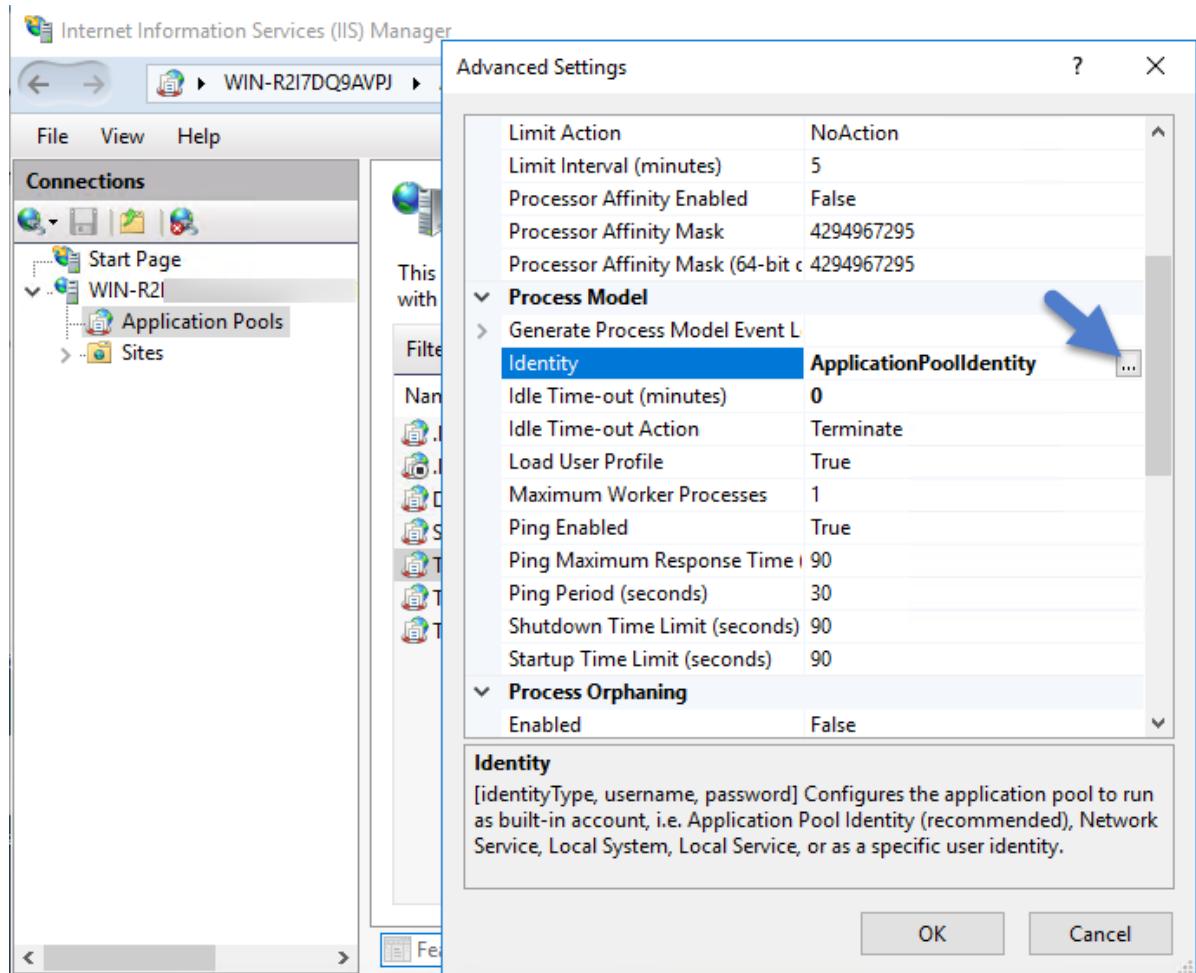
Load User Profile
 [loadUserProfile] This setting specifies whether IIS loads the user profile for an application pool identity. When this value is true, IIS loads the user profile for the application pool identity. Set this value to false when you require the IIS 6.0 behavior of not loading the user profile for the application pool identity.

2 Assigning Identity of Application Pool(s) in IIS

Note: This is required so that application pool uses service account for actions like starting powershell.

You must have IIS installed on your web server before completing these steps.

4. Open IIS on your web server (**Search > inetmgr**)
5. Locate the application pool(s) that your Thycotic product is using, right-click **Advanced Settings...** then the **Identity** box in the "Process Model" section, click the three dots on the right of the box.



6. Select the **Custom Account** radio button, click **Set**, enter your service account's name and password, and click **OK**.
 - **Note:** For Privilege Manager you will need to perform this step for multiple application pools.

The screenshot shows the 'Application Pools' section of the IIS Manager. A table lists application pools: '.NET v4.5', '.NET v4.5 Classic', 'DefaultAppPool', and 'Saviynt'. The 'Saviynt' row is selected. An 'Advanced Settings' pane is open for this pool. In the 'General' section, the 'Name' is set to 'Saviynt'. The 'CPU' section includes settings for 'Limit (percent)', 'Limit Action', 'Limit Interval (minutes)', and 'Processor Affinity Enabled'. A modal dialog titled 'Application Pool identity' is displayed, showing the 'Custom account' option selected. It contains fields for 'User name' (set to 'preprod\SaviyntWin-PSConn.sv'), 'Password', and 'Confirm password'. Buttons for 'OK' and 'Cancel' are at the bottom. To the right of the main pane, a detailed table provides specific values for each setting, such as 'v4.0' for .NET CLR Version and 'Integrated' for Managed Pipeline Mode.

This screenshot shows the same 'Application Pools' interface and 'Saviynt' application pool selection. The 'Advanced Settings' pane is open, but the configuration details are different. The 'General' section shows 'Name' as 'Saviynt'. The 'Process Model' section is expanded, showing the 'Identity' setting. The 'Identity' dropdown is set to 'preprod\SaviyntWin-PSConn.sv'. Other settings in this section include 'Idle Time-out (minutes)' (0), 'Idle Time-out Action' (Terminate), 'Load User Profile' (True), 'Maximum Worker Processes' (1), 'Ping Enabled' (True), 'Ping Maximum Response Time (seconds)' (90), and 'Ping Period (seconds)' (30). The 'Identity' help text at the bottom of the pane states: '[identityType, username, password] Configures the application pool to run as built-in account, i.e. Application Pool Identity (recommended), Network Service, Local System, Local Service, or as a specific user identity.'

Contact us

Manjunath Madiraju
Manager, Technical Architect

Email mmadiraju@kpmg.com

Ken Dunbar
Engagement Director

Email kbdunbar@kpmg.com

© 2021 [legal member firm name], a [jurisdiction] [legal structure] and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavour to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

The KPMG name, logo and "cutting through complexity" are registered trademarks or trademarks of KPMG International Cooperative ("KPMG International").