Azure DevOps Server 2020 Installation Guide

Benjamin Day
benday@benday.com
https://www.benday.com

v1.0.0
October 19, 2020

Contents

Chapter 1: Introduction 3
Chapter 2: Install Windows Server 2019 5
Introduction 5
Install Windows Server 2019 5
(Optional) Turn off IE Enhanced Security Configuration 25
(Optional) Enable Remote Desktop 29
Join this Server to the Active Directory Domain 31
Chapter 3: Install SQL Server 2019 for Azure DevOps Server 2020 39
Introduction 39
Install SQL Server 2019 39
Chapter 4: Install Azure DevOps Server 2020 53
Introduction 53
Run the Installer 53
Chapter 5: Configure an SMTP Server for Azure DevOps Server 78
Chapter 6: Install Azure DevOps Build & Release Agent on Windows Server 81
Download the Agent Installer 81
Extract the Agent 86
Configure the Agent 93
Chapter 7: Training, Consulting, & Software Development 96

Chapter 1: Introduction

Hi. I hope you find this guide helpful. If you get stuck, please drop me a line at info@benday.com and I'll try to help you through your problem.

And now, a little background on this guide.

Azure DevOps Server was originally named Team Foundation Server (TFS). I've been working with Team Foundation Server since before it was released in 2005. For the first 5 to 10 years of its existence, TFS was quite hard to install. To say that the install process was unforgiving is a huge understatement. There were lots and lots of pieces and if even the slightest thing was configured incorrectly, you'd probably want to just format the whole server and start over.

Over time, the Team Foundation Server team at Microsoft worked on the installer and made the process much easier and reliable. Installing TFS or Azure DevOps still isn't 100% fool-proof – there are a lot of pieces and a lot of steps – but it's not that hard to do if you're doing a basic installation.

This guide will walk you through the steps of doing a single-server Azure DevOps Server 2020 installation.

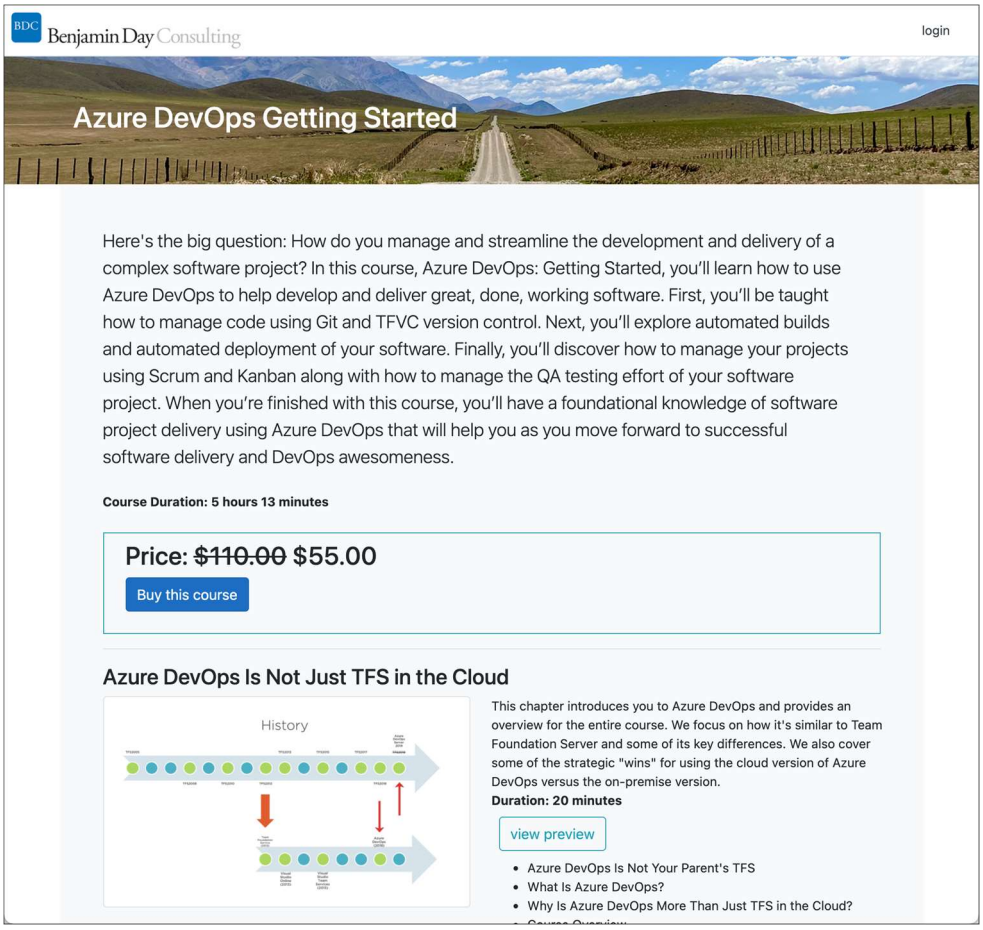
If you get stuck, please drop me a line at info@benday.com.

-Ben

Training, Consulting, Upgrades, & Migrations

Looking for training for how to use Azure DevOps? Check out our online video course.

https://courses.benday.com



We also have courses available at Pluralsight as well as options for virtual (remote) training and in-person training. Topics include Azure DevOps, Scrum, Test-Driven Development & Unit Testing, ASP.NET Core, Software Architecture and more! We also do consulting and custom software development on these topics. We also help companies with tricky Team Foundation Server installations, upgrades, and migrations – including helping you to migrate your on-premise installations to Azure DevOps in the cloud.

Drop us a line at info@benday.com or visit https://www.benday.com for more information.

Chapter 2: Install Windows Server 2019

Introduction

The first step for installing Azure DevOps Server 2019 (AzDO) is to install Windows Server. In this guide, I'm going to be using Windows Server 2019. I'm also assuming that you'll be installing Azure DevOps Server 2019 in a single server configuration – aka. the AZDO Application Tier and SQL Server will be installed on a single machine.

Install Windows Server 2019

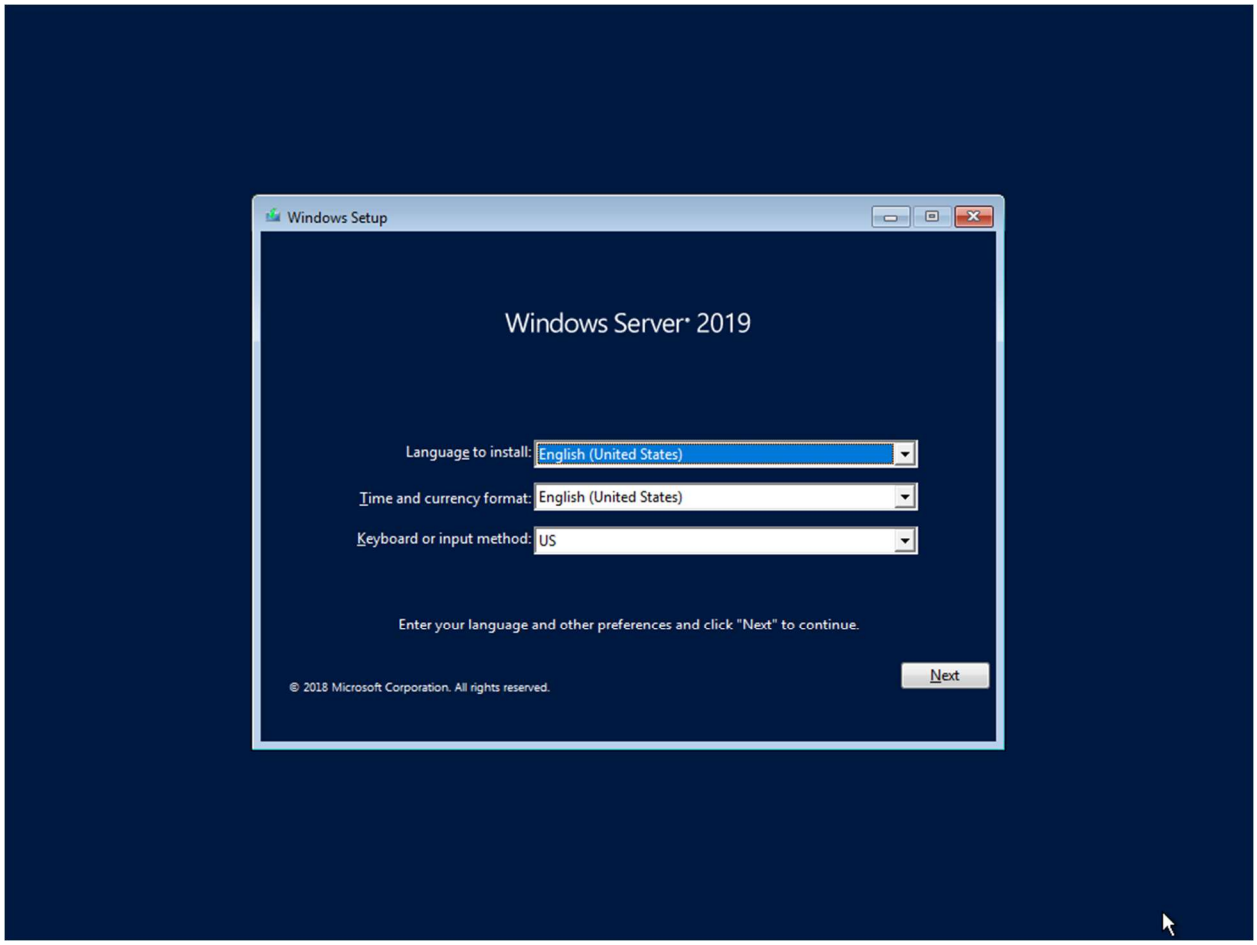
I'm assuming that you've got an Active Directory Domain already set up and configured and that we'll eventually be joining this new server to that domain.

- Either insert your **Windows Server 2019 DVD** into the DVD drive or mount the **Windows Server 2019 ISO image** into the DVD drive for your virtual machine.
- Start the machine

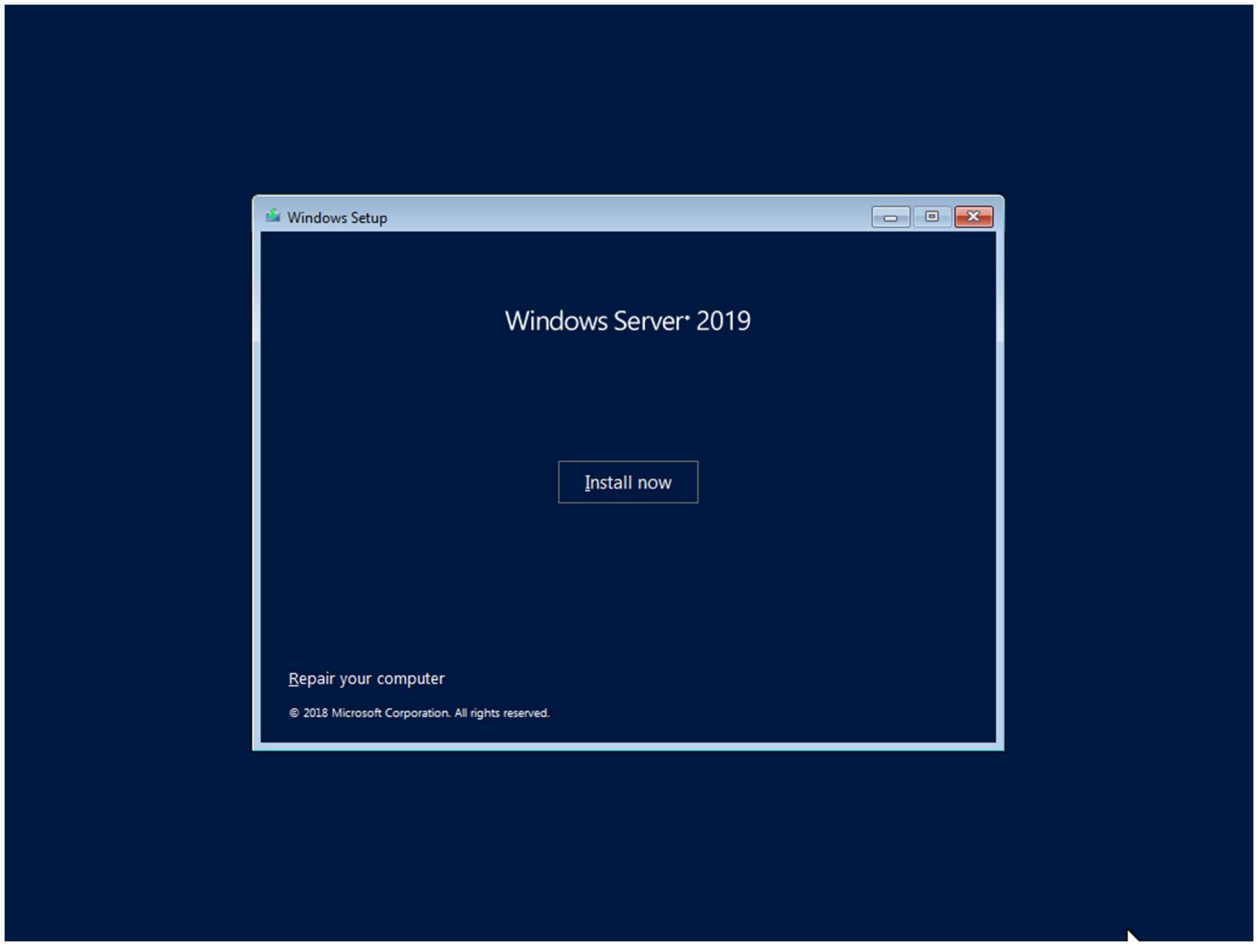
You will see the **Press any key to boot from CD or DVD....** message appear.

- Press any key to boot from the DVD drive

You should see the Windows Setup welcome screen.



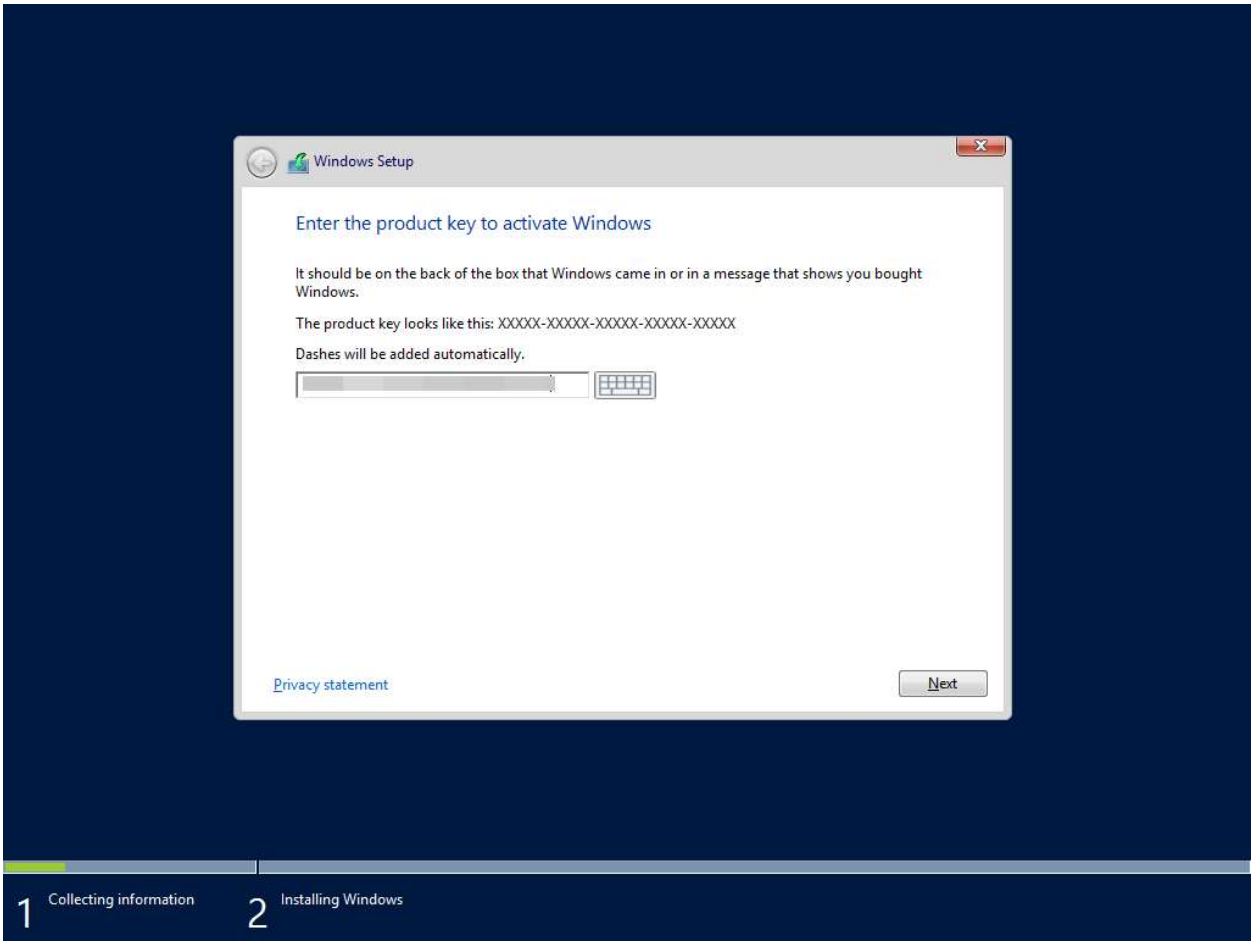
- Click **Next**



- Click the **Install now** button

You should see a screen saying that setup is starting.

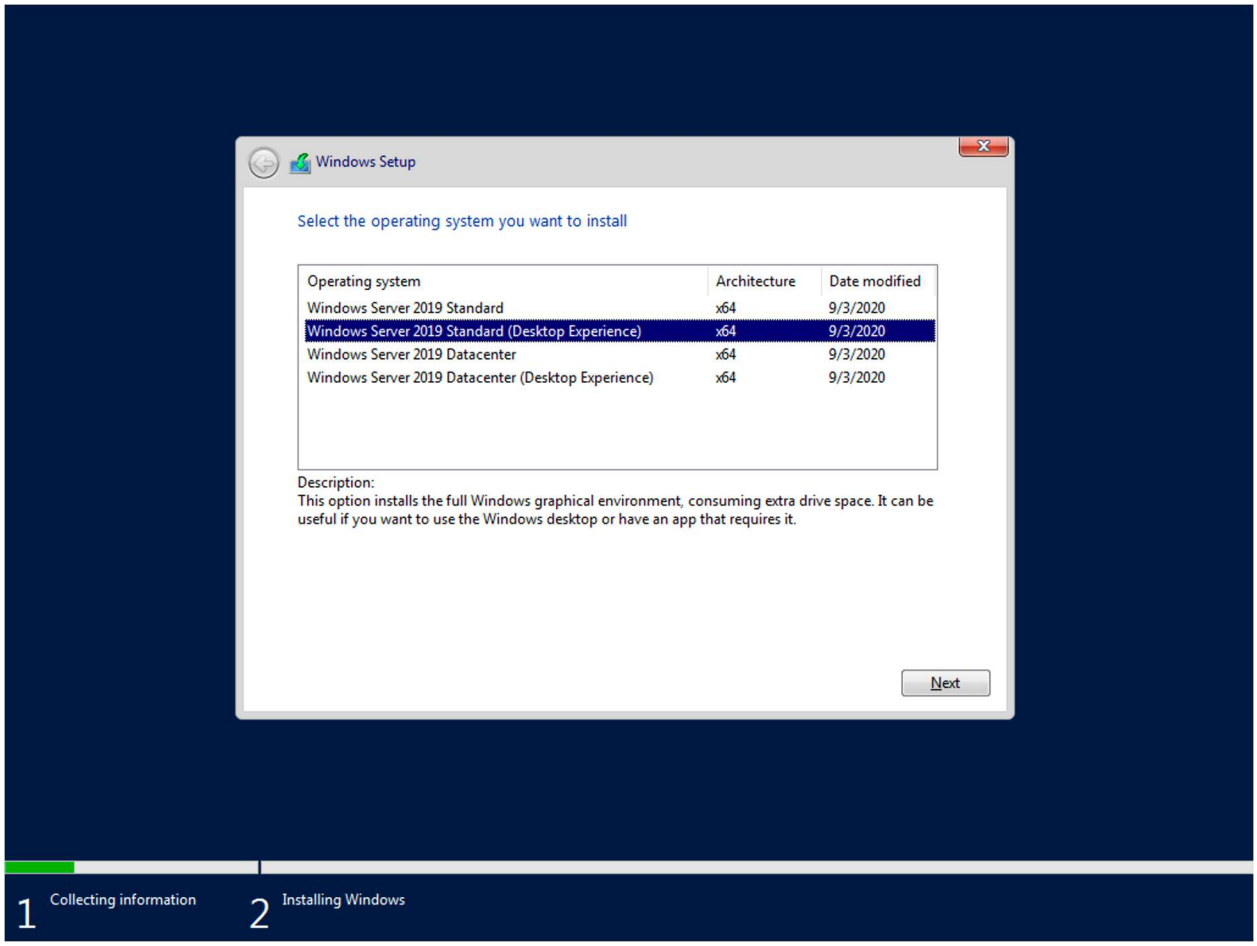
You'll be prompted for a license key.



- Enter your license key
- Click the **Next** button

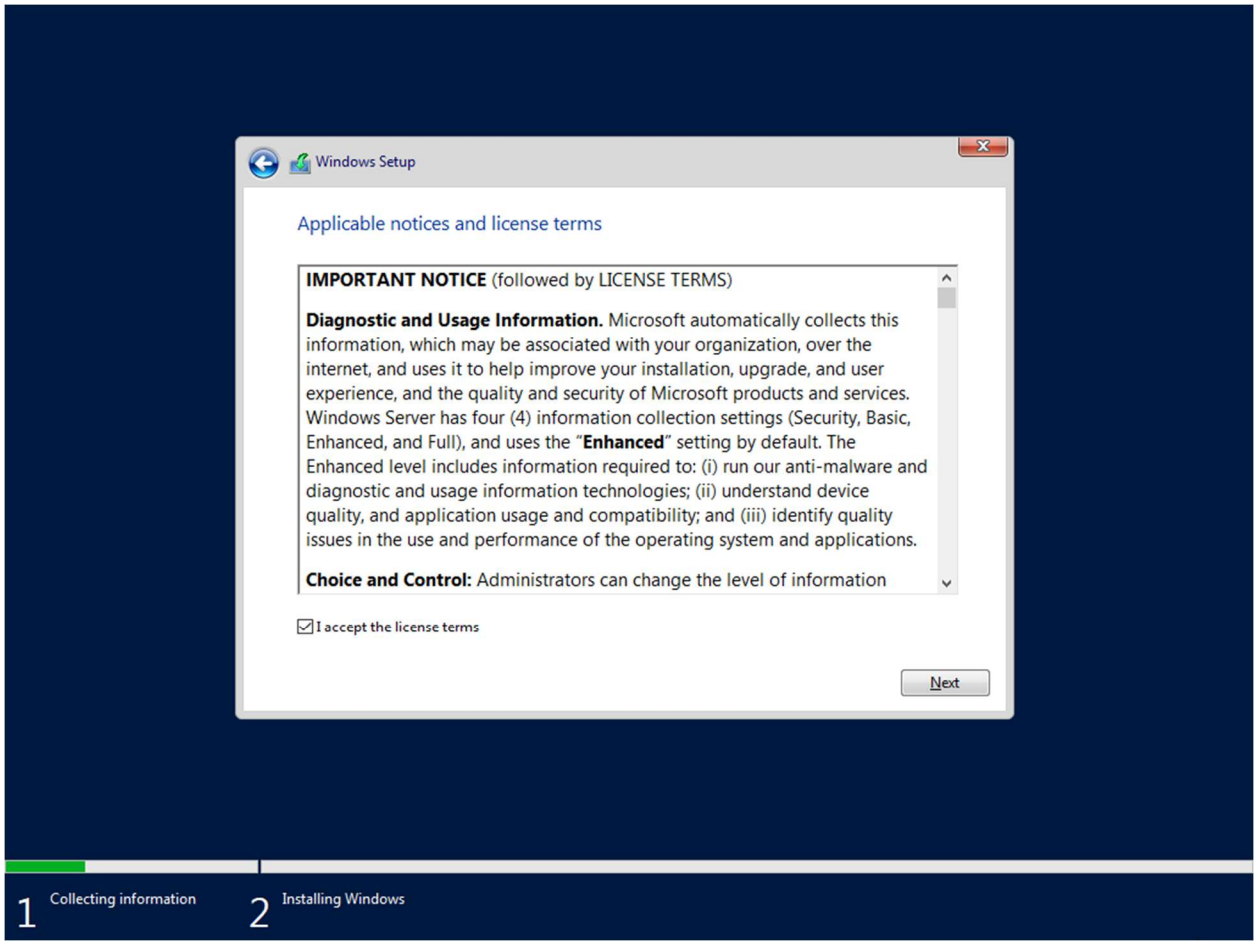
You'll be prompted to choose what version of Windows Server 2019 that you'd like to install. This guide assumes that you'll be installing Windows Server 2019 Standard and that you'll be installing the graphical user interface (GUI).

BTW, make sure you choose an option that says "Desktop Experience"!



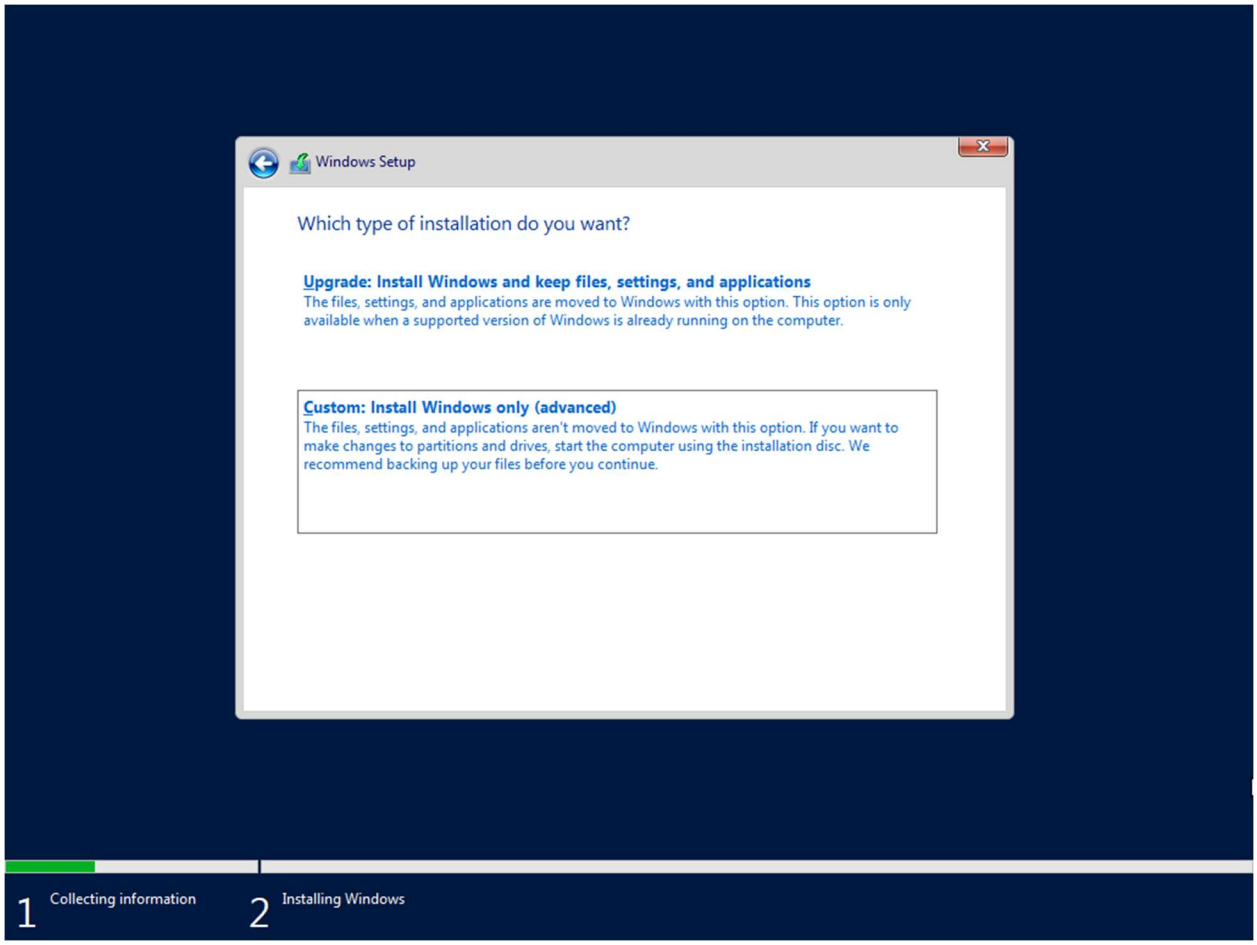
- Choose **Windows Server 2019 Standard (Desktop Experience)**
- Click the **Next** button

You'll be shown the license terms.



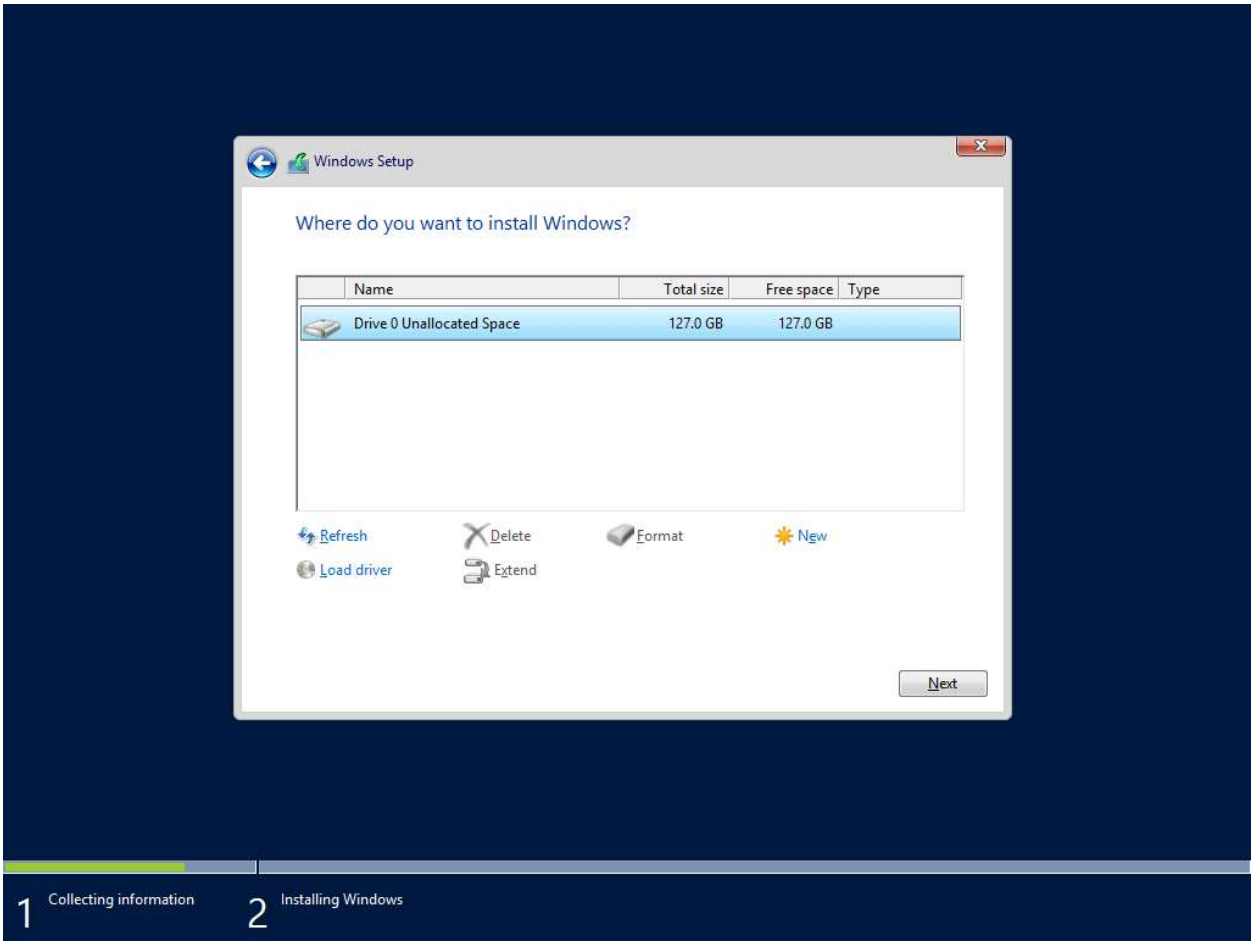
- Read the license terms in painstaking detail
- Check the **I accept the license terms** box
- Click the **Next** button

We'll be doing a new installation rather than an upgrade.



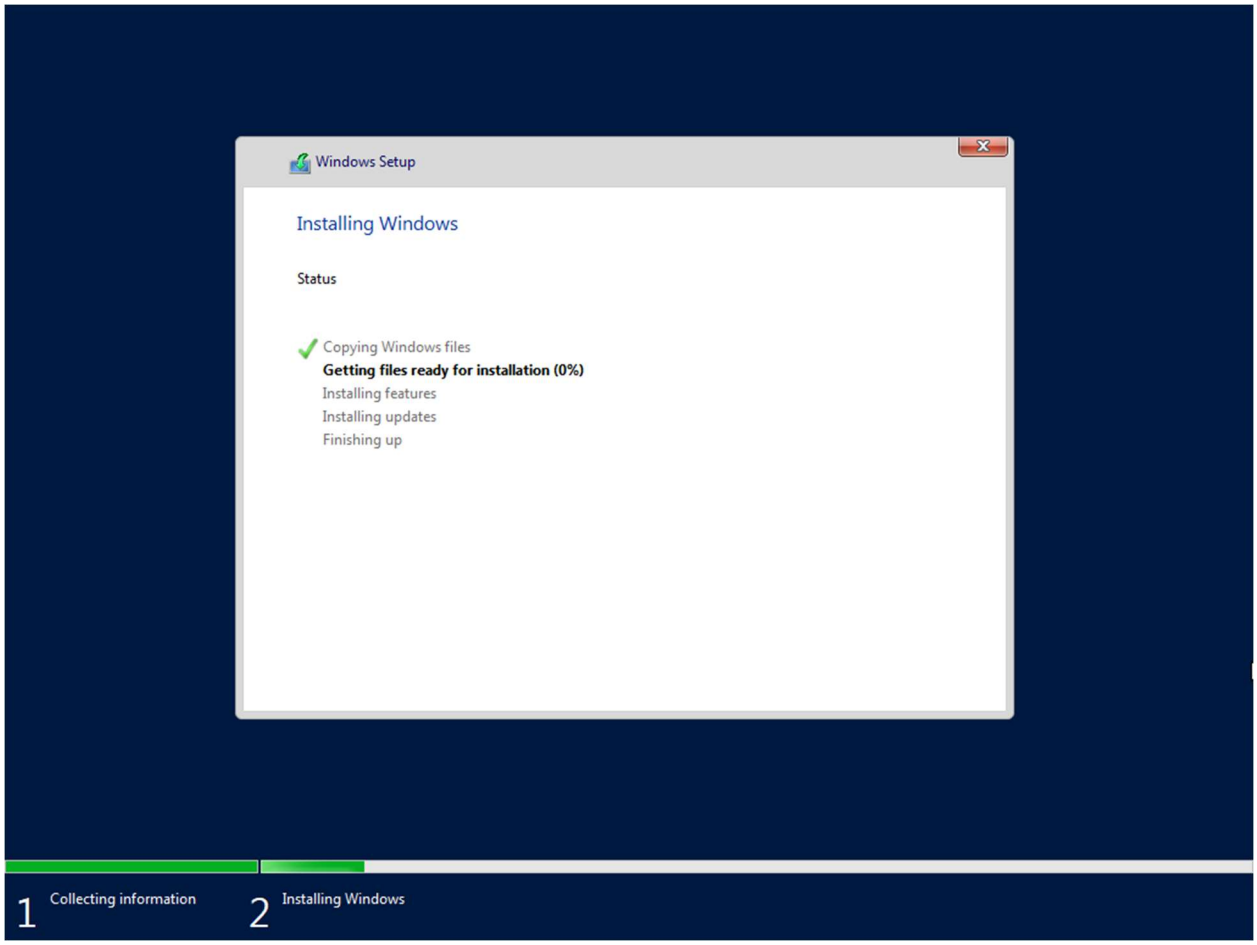
- Click **Custom: Install Windows only (advanced)**

You should now see a dialog that lets you choose where you'd like to install windows. I'm assuming that we'll be installing on a new computer without any existing partitions on the disk.

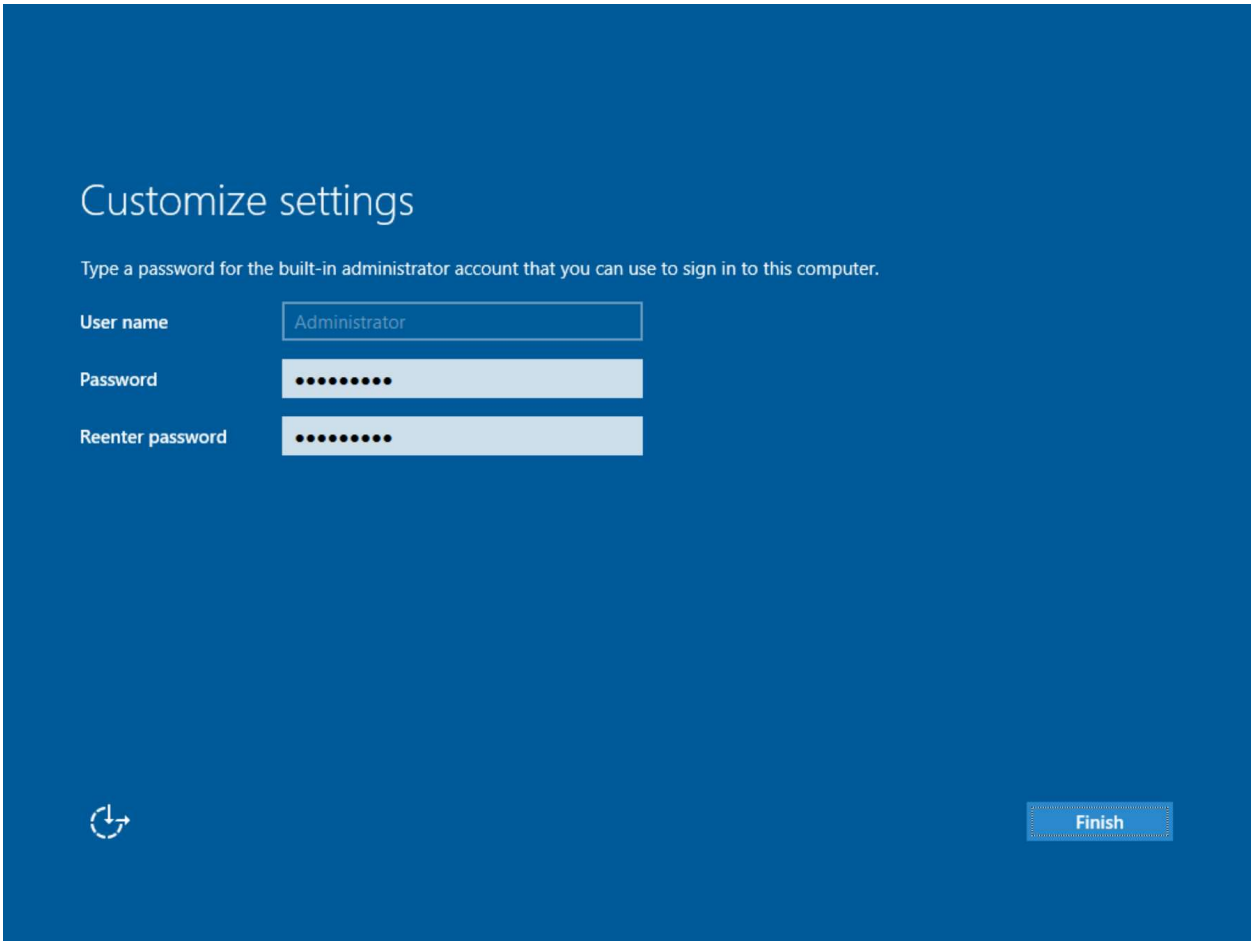


- Choose **Drive 0 Unallocated Space** or another appropriate partition or unallocated disk space
- Click **Next**

The installation should now be running.



Your server will automatically reboot when the installation is complete and you'll be prompted to create an administrator password.

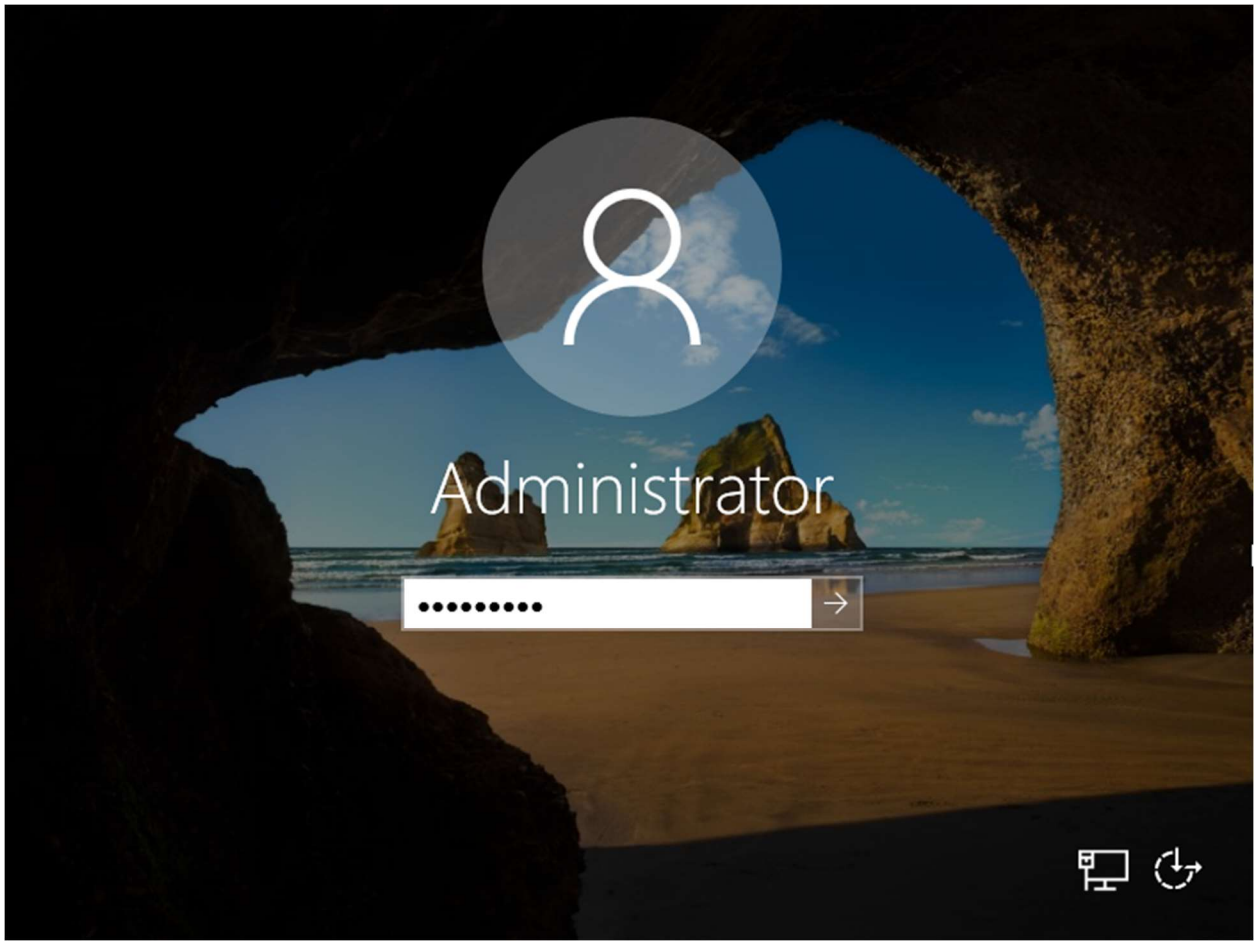


- Enter your password in the **Password** box
- Enter your password in the **Reenter password** box
- Click the **Finish** button

You should now see the lock screen for your new server.

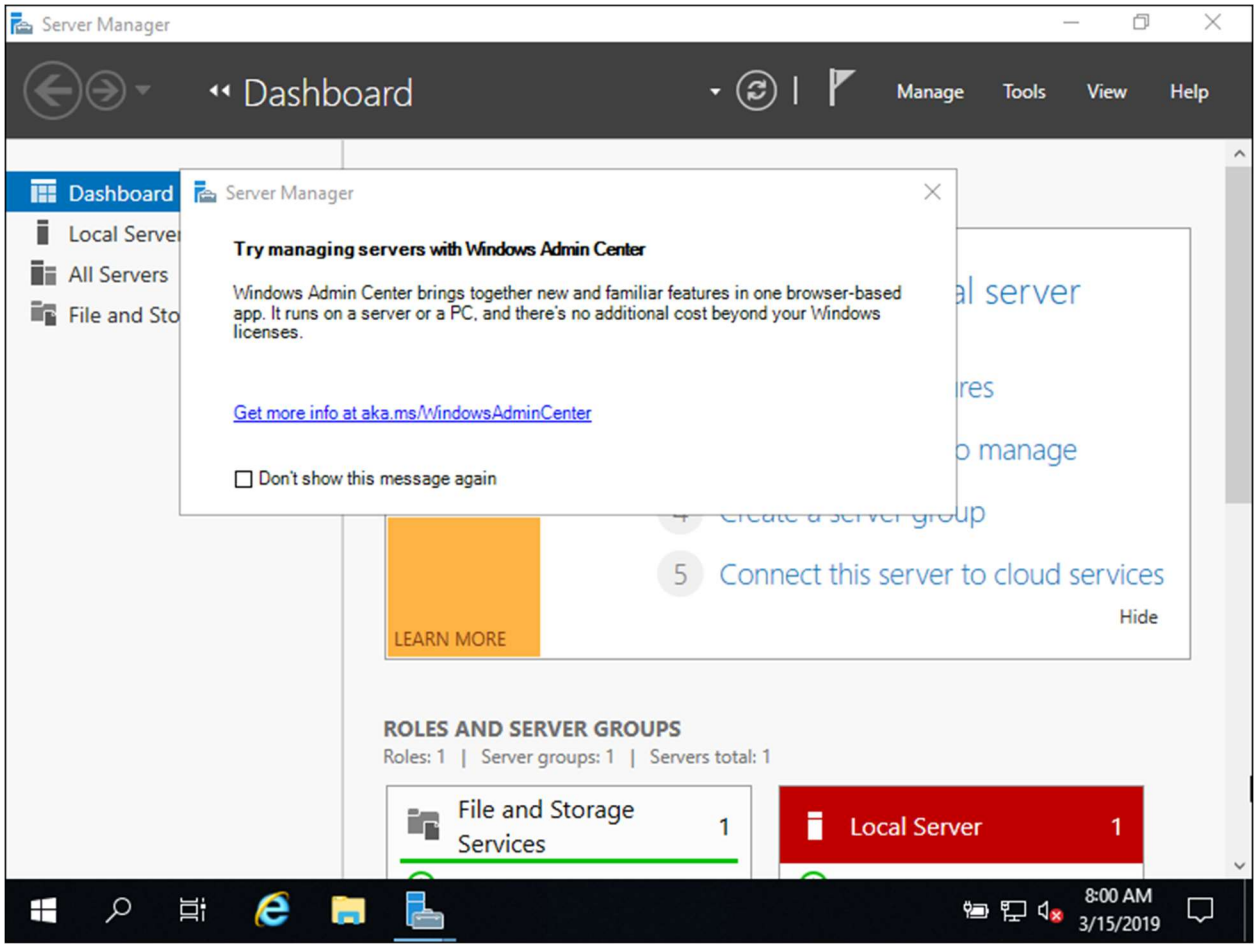
- Type **CTRL-ALT-DEL** to open a login prompt

You should see the login prompt.



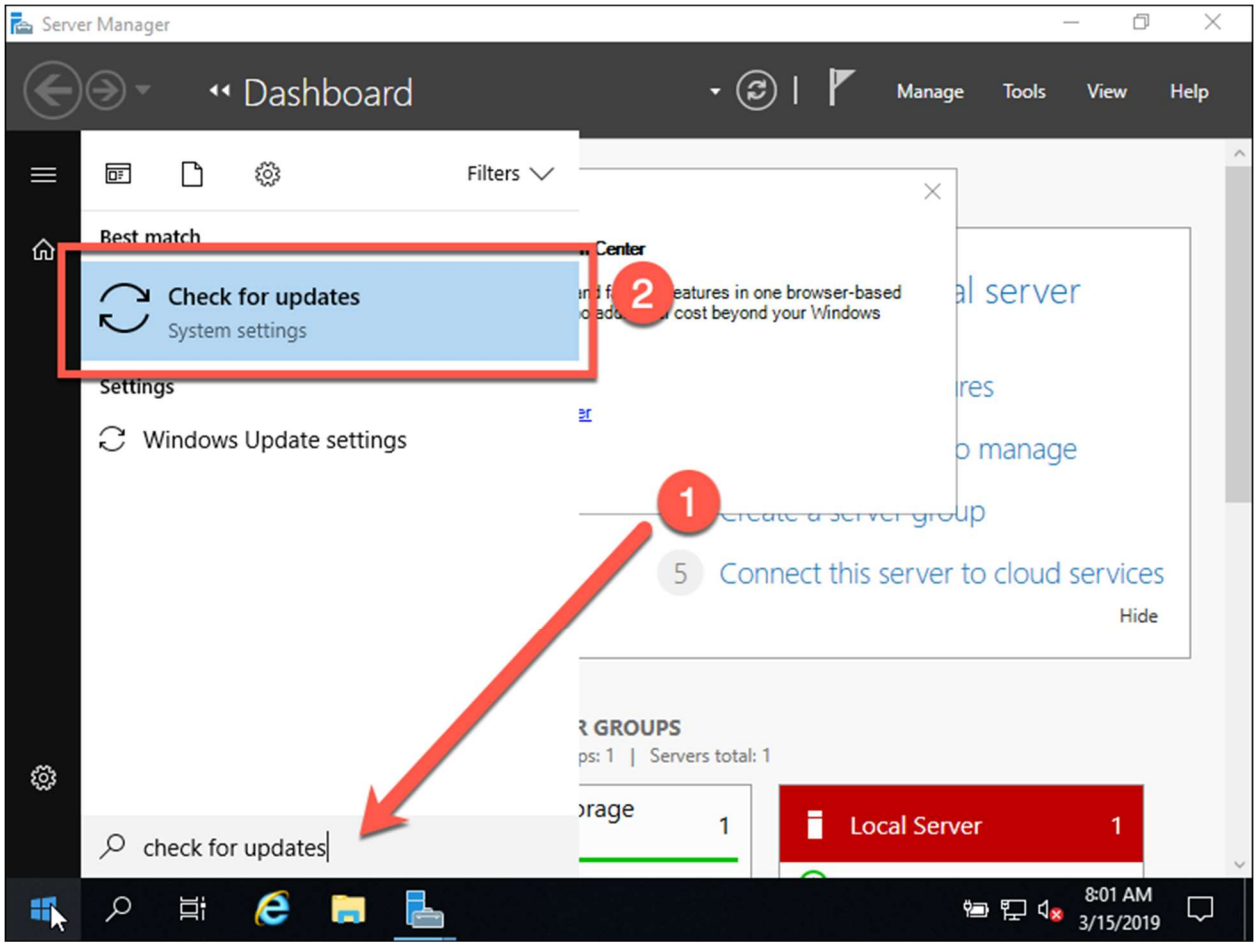
- Enter your **password** followed by **<ENTER>**

You should now be logged in and you should see the server manager dashboard. The next step is to run Windows Update to patch this server.



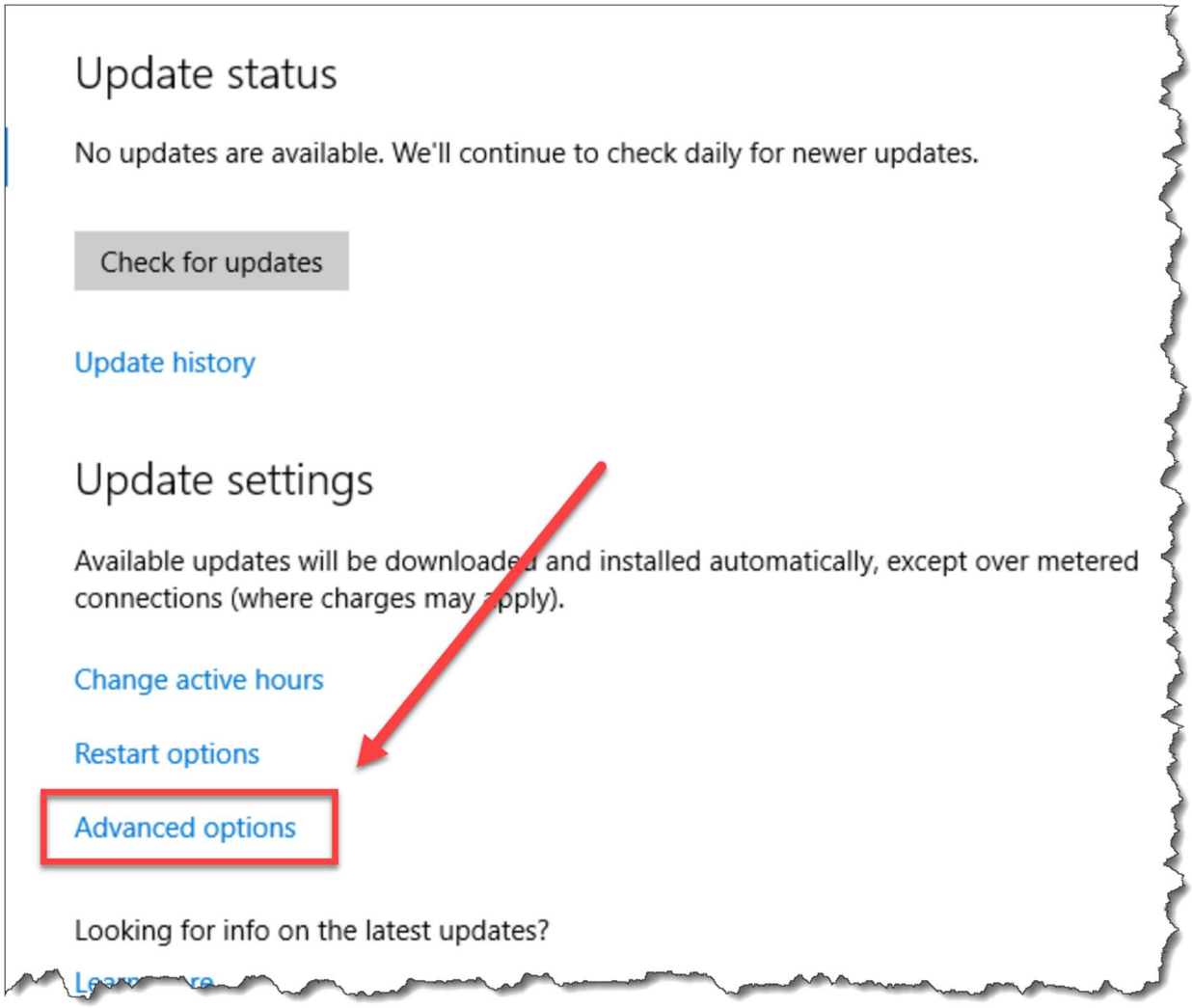
- Click the **Windows button** in the bottom left of the screen to navigate to the Start screen

You should now be at the Start menu.



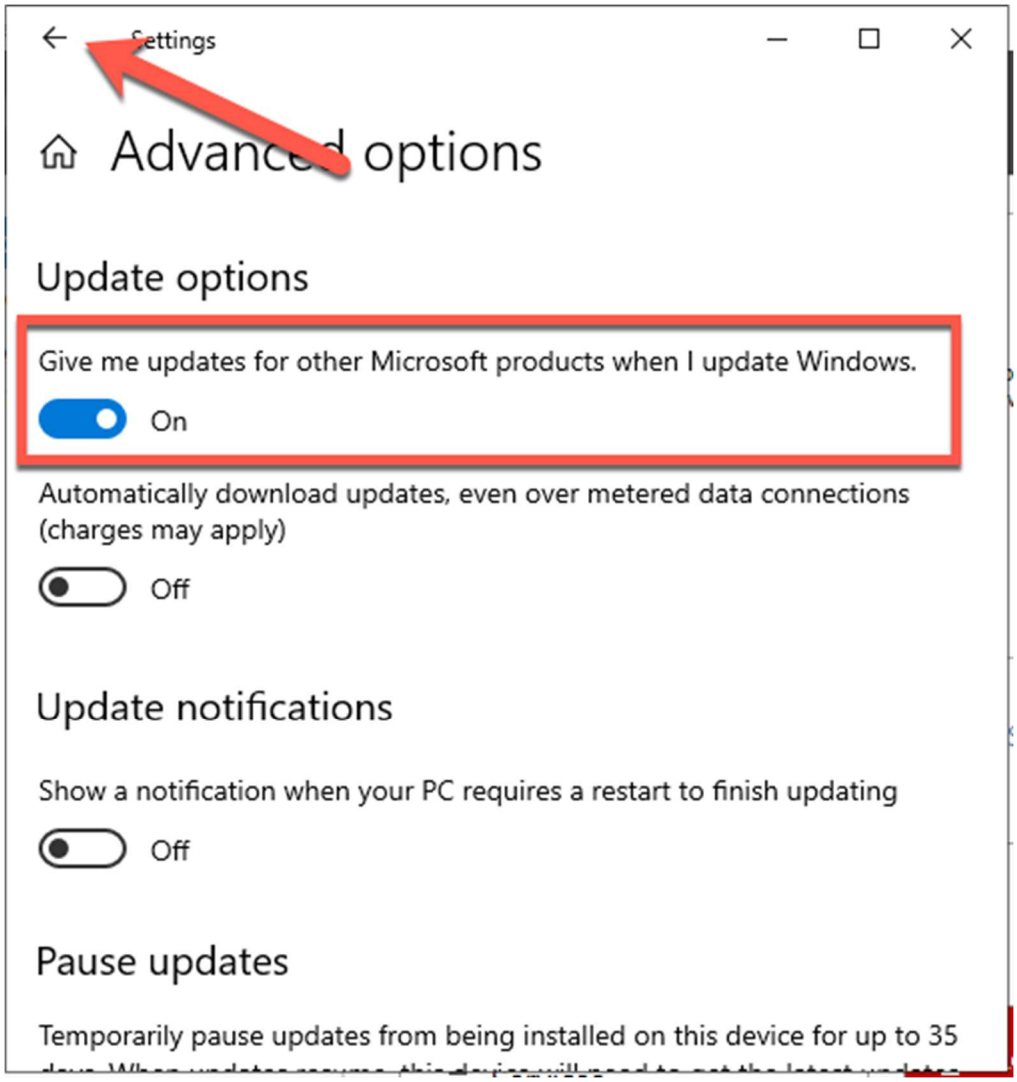
- In the search box, type **Check for updates**
- Click **Check for updates** in the search results

You should now see the Settings window. By default, Windows Update only gets patches for Windows itself but we want to enable patches for other products as well.



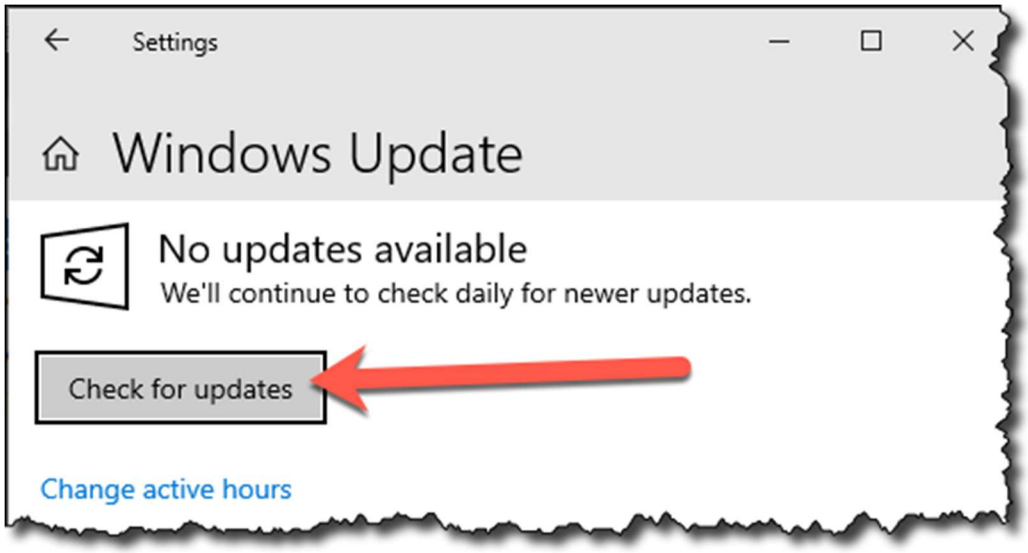
- Click the **Advanced options** link

You should be on a screen with the title **Advanced options**.



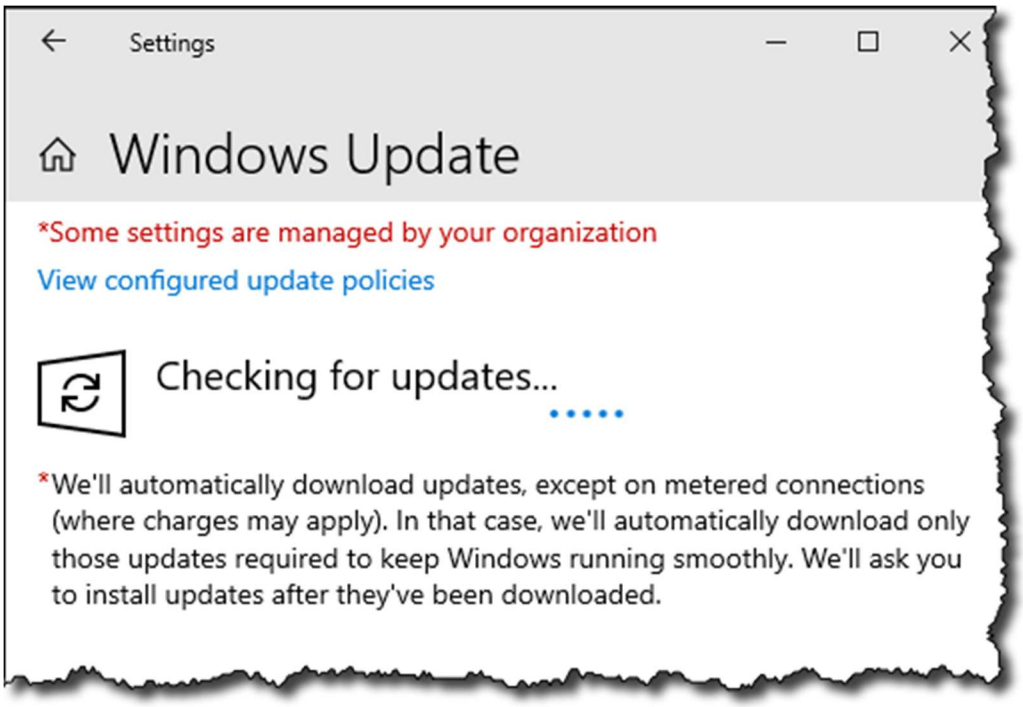
- Check **Give me updates for other Microsoft products when I update Windows**
- In the upper left corner of the screen, click the back arrow button

You should be back on the **Update status** screen.



- Click the **Check for updates** button

Windows Update should now be checking for updates.



Windows Update will probably find a ton of available updates. Let them all run and reboot your server as needed.

When the patches are all finished applying, make sure you're logged in as Administrator and continue on to the next page where I'll walk you through the optional step of disabling a useless and annoying anti-feature in Windows called IE Enhanced Security.

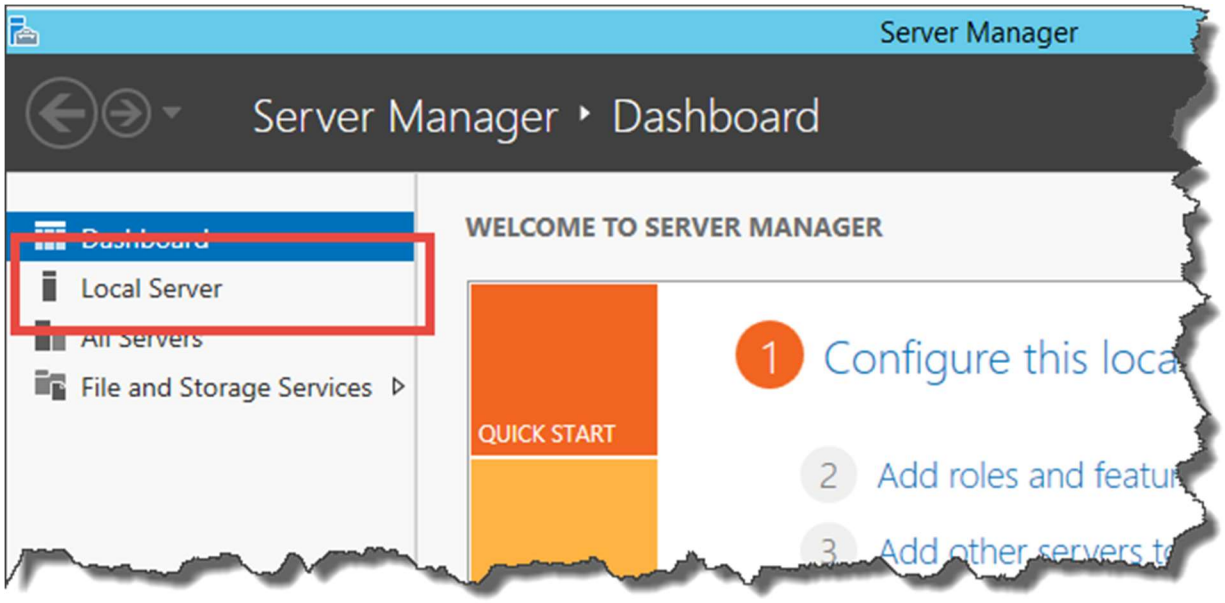
(Optional) Turn off IE Enhanced Security Configuration

Ok. So if you're paranoid about the internet and making sure that your server doesn't accidentally get infected by a virus that comes in because of someone's hapless browsing habits, you'll probably want to skip this section.

If you don't wear a tinfoil hat every day to keep the illuminati's evil space rays from controlling your mind, then you're probably like me and find IE Enhanced Security to be a royal pain in the behind. At some point, you're going to want to use the Internet Explorer web browser on this server and it'll be painful if IE Enhanced Security is turned on.

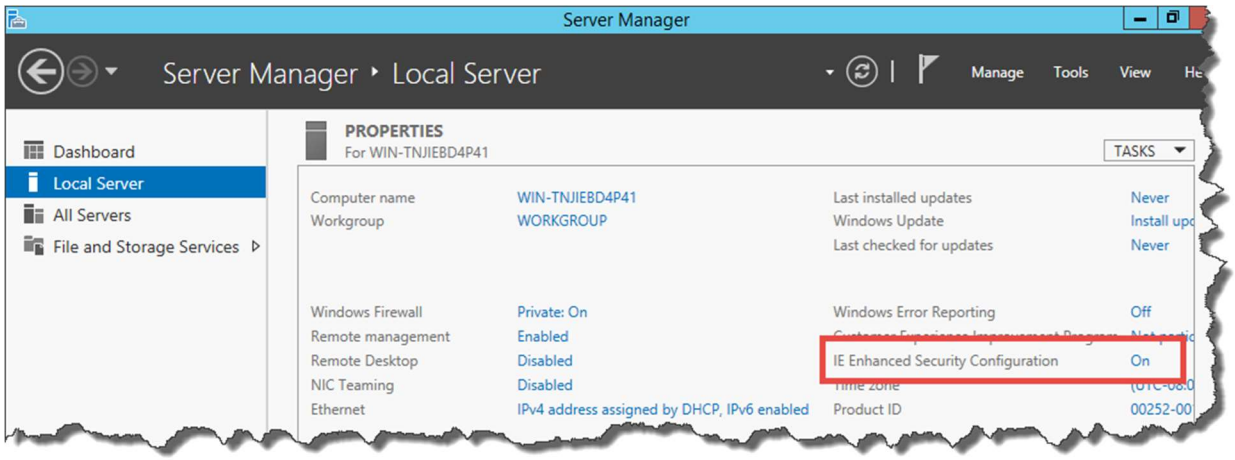
Let's turn it off.

You'll start this process by using **Server Manager**.



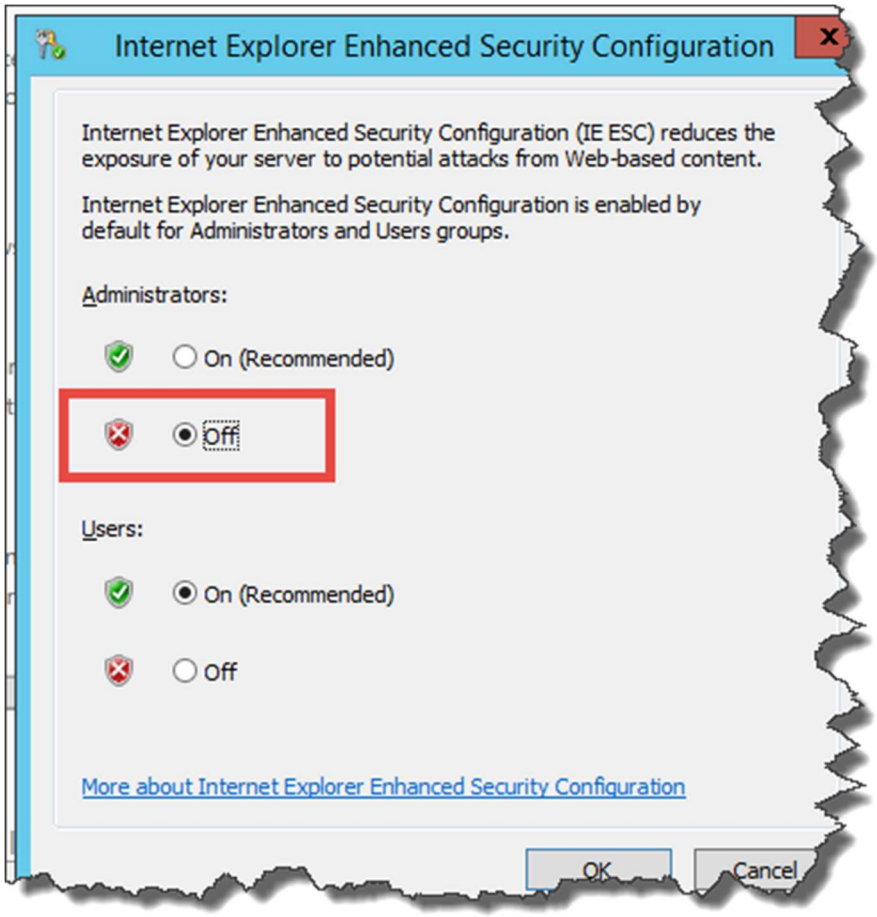
- In the left column of Server Manager, click **Local Server**

Towards the right side of the Server Manager window, you'll see an item that says **IE Enhanced Security Configuration**. It'll be set to **On**.



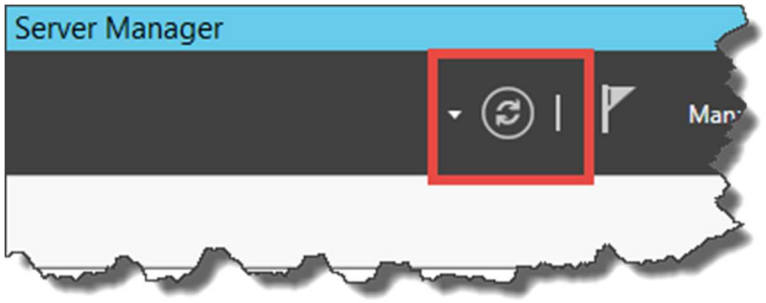
- Click the link that says **On**

You should now see the **Internet Explorer Enhanced Security Configuration** dialog.



- Under Administrators, select the **Off** radio button
- (Optional) Under Users, select the **Off** radio button
- Click the **OK** button

You should now be back at the main page of the Server Manager.



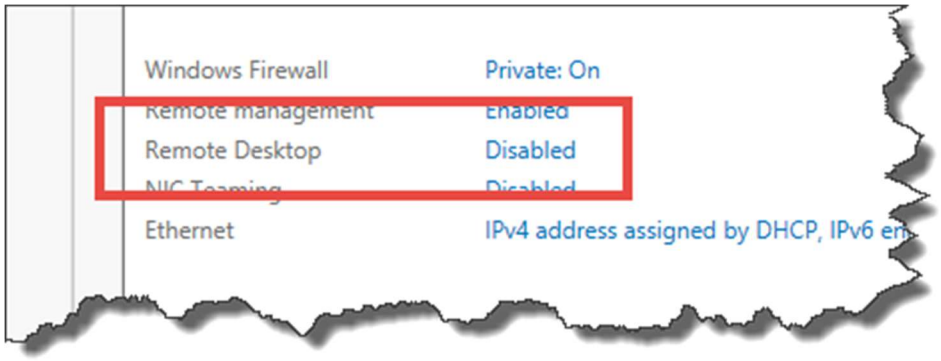
- Click the **Refresh** button

IE Enhanced Security Configuration should now be set to Off.

(Optional) Enable Remote Desktop

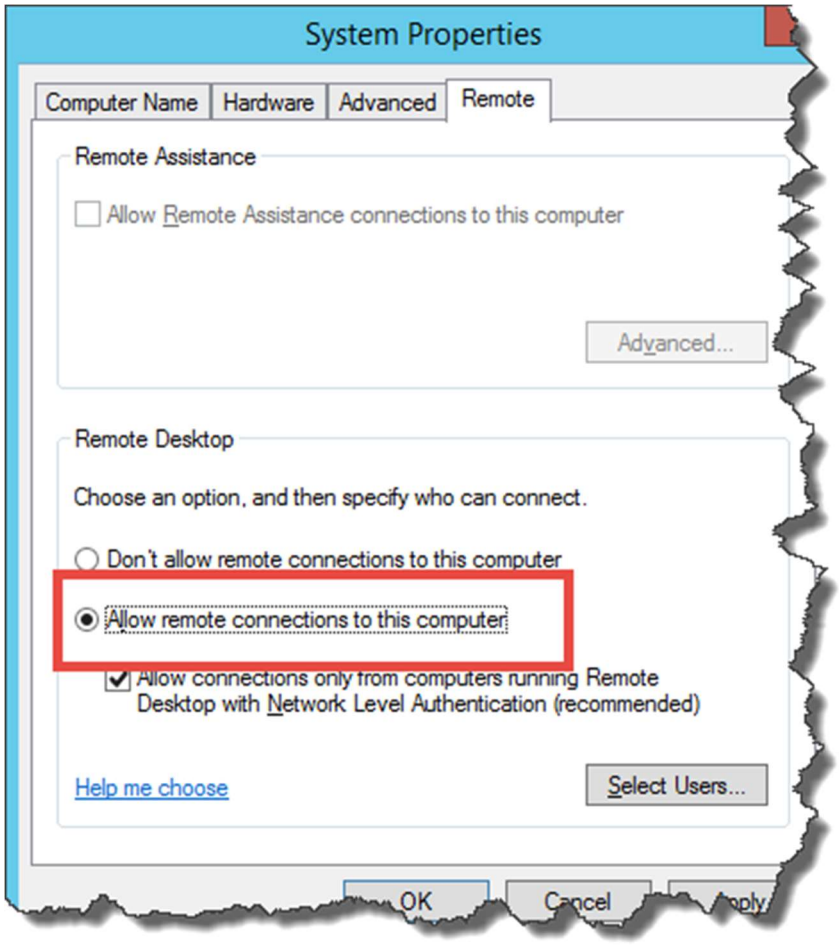
Are you lazy or maybe lazy-ish? Do you like convenience? Yah. Me, too. So that means that you'll probably want to enabled Remote Desktop. If you're a member of the Tinfoil Hat Patrol, you'll probably want to skip this section.

- Find **Remote Desktop** in Server Manager



- Click the **Disabled** link to the right of Remote Desktop

You should now see the **System Properties** dialog.



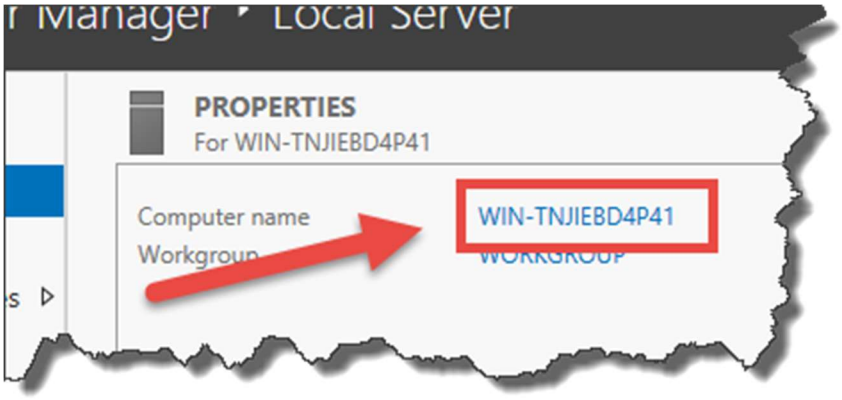
- In the Remote Desktop group, choose **Allow remote connections to this computer**
- Click the **OK** button

Remote Desktop is enabled.

Join this Server to the Active Directory Domain

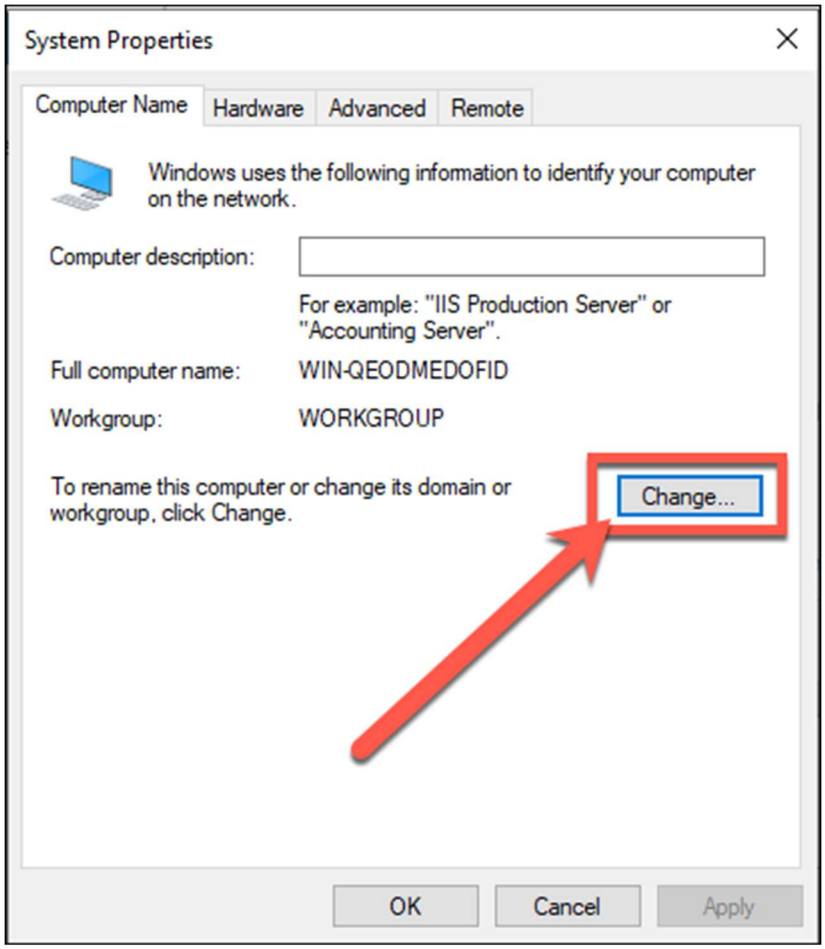
At the moment, you've got a stand-alone server with a wacky name that's not attached to anything. Workgroups? That's like the networking equivalent of having a stand-alone MP3 player that only syncs over USB. Who does that? What year is this?! 2005?! Not very useful. You'll now rename this computer and join it to your Active Directory domain so that it plays nicely with others.

In Server Manager, you'll see **Computer name** and **Workgroup**.



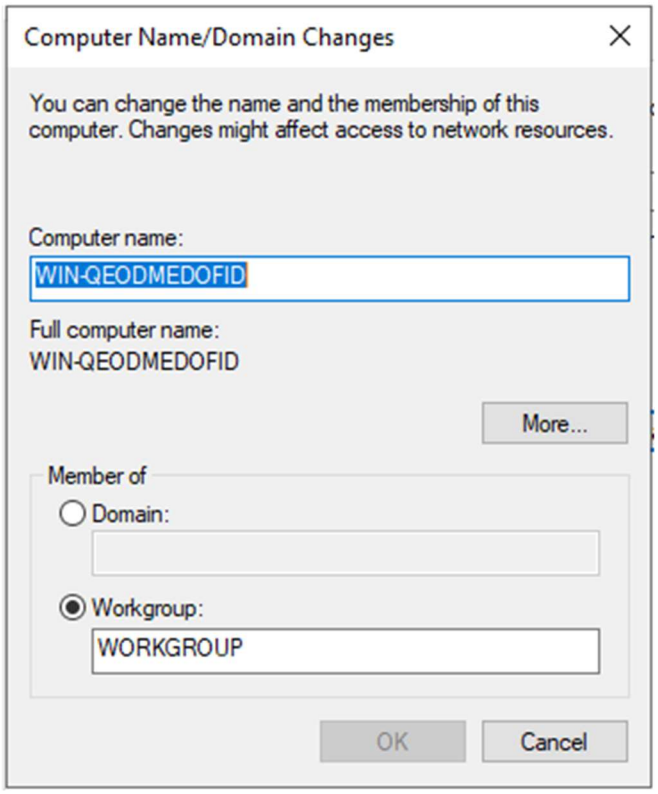
- Click on the computer name link

You should now see the **System Properties** dialog.

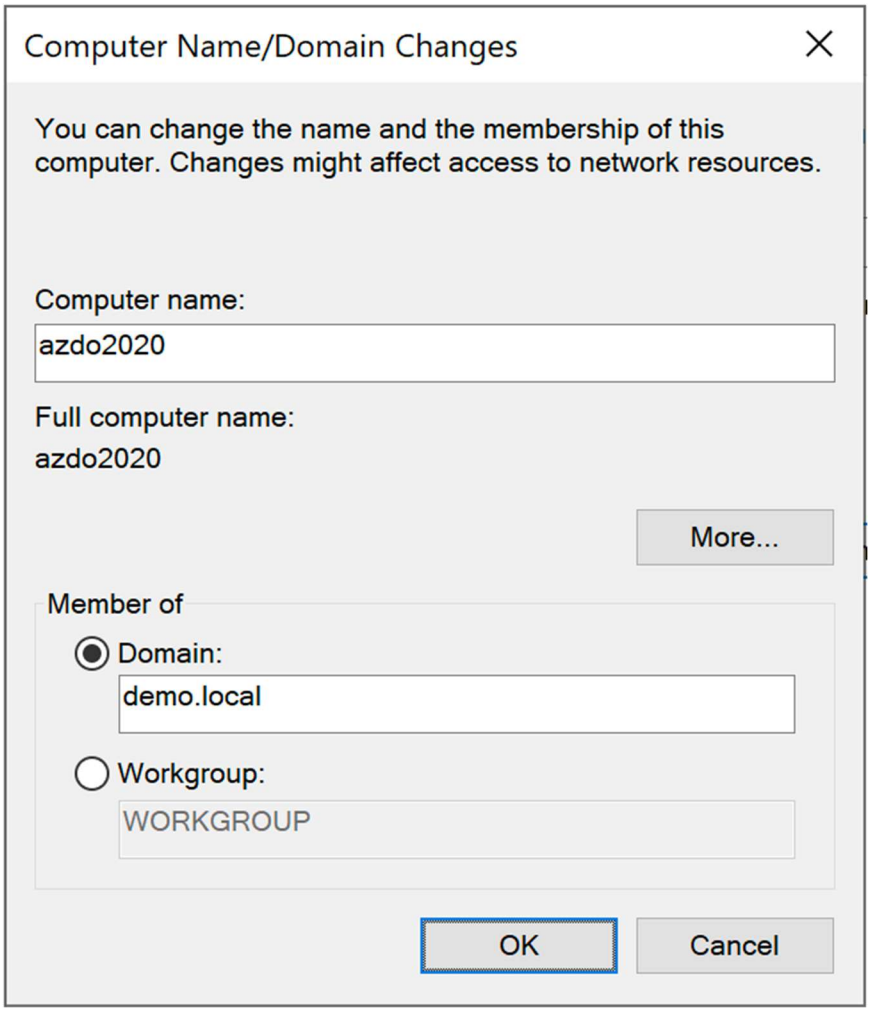


- Click the **Change...** button

You should now be on the **Computer Name/Domain Changes** dialog. The dialog should be showing you the current name of the computer and the workgroup membership.

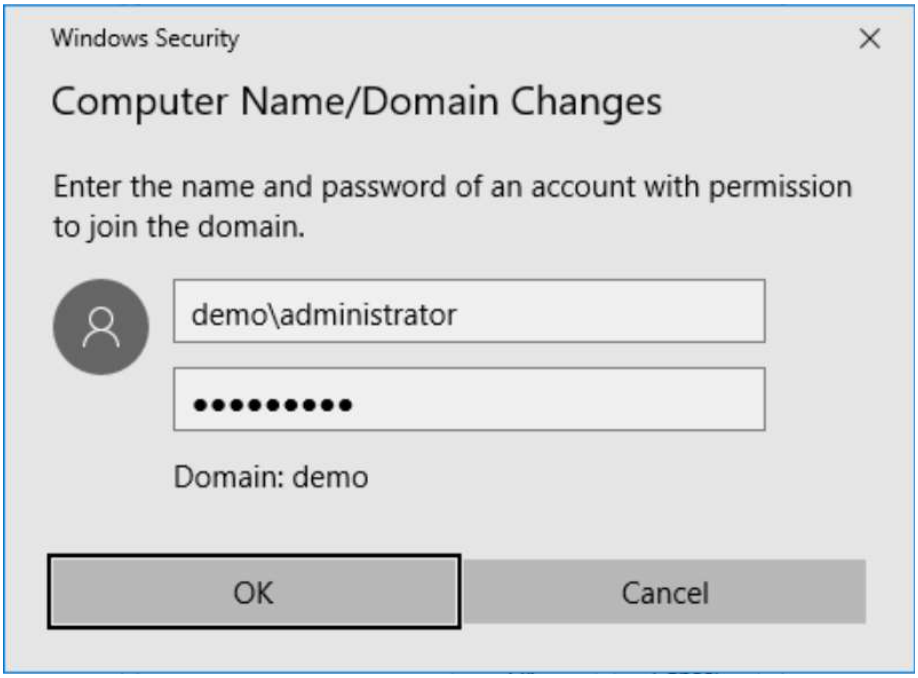


You should now change the values to be what you want the server to be named and the Active Directory domain that it should be attached to.



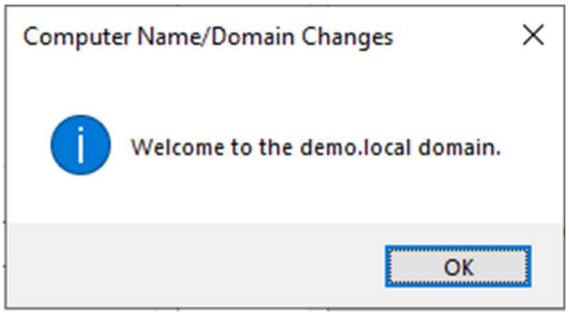
- In the Computer name textbox, enter the desired name for this server
- Under **Member of** choose the **Domain** radio button
- In the **Domain** textbox, enter the name of the Active Directory domain
- Click the **OK** button

You'll be prompted for the username and password for a domain administrator for the target domain.



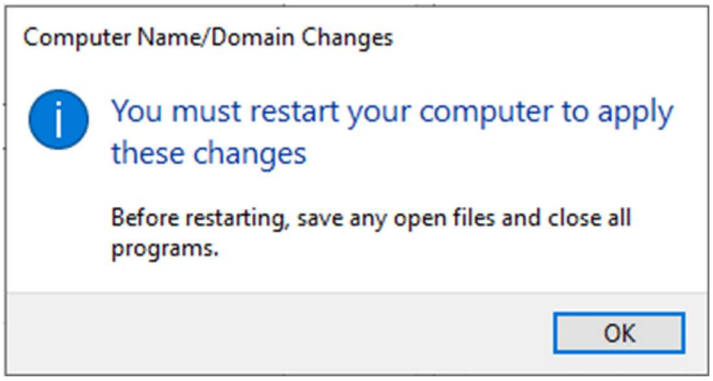
- Enter the username and password
- Click **OK**

You should see a dialog welcoming you to the new domain.

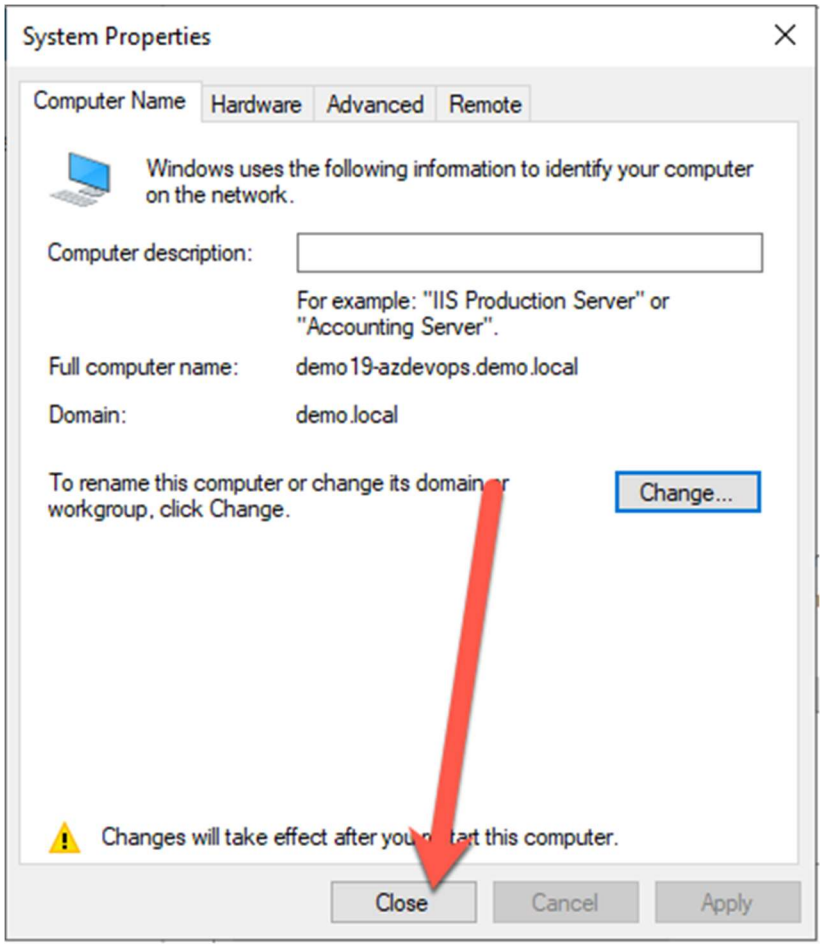


- Click **OK**

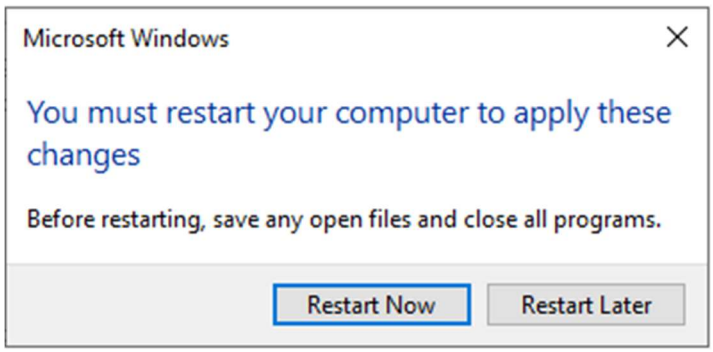
You'll be notified that you'll need to reboot this server.



- Click the **OK** button



- Click the **Close** button



- Click the **Restart Now** button

The server will restart and return you to the lock screen. The server has been installed and joined to the domain.

NOTE: It is *optional* but you might find it helpful to configure this server to have a static IP address and a static A record entry in your DNS server.

Chapter 3: Install SQL Server 2019 for Azure DevOps Server 2020

Introduction

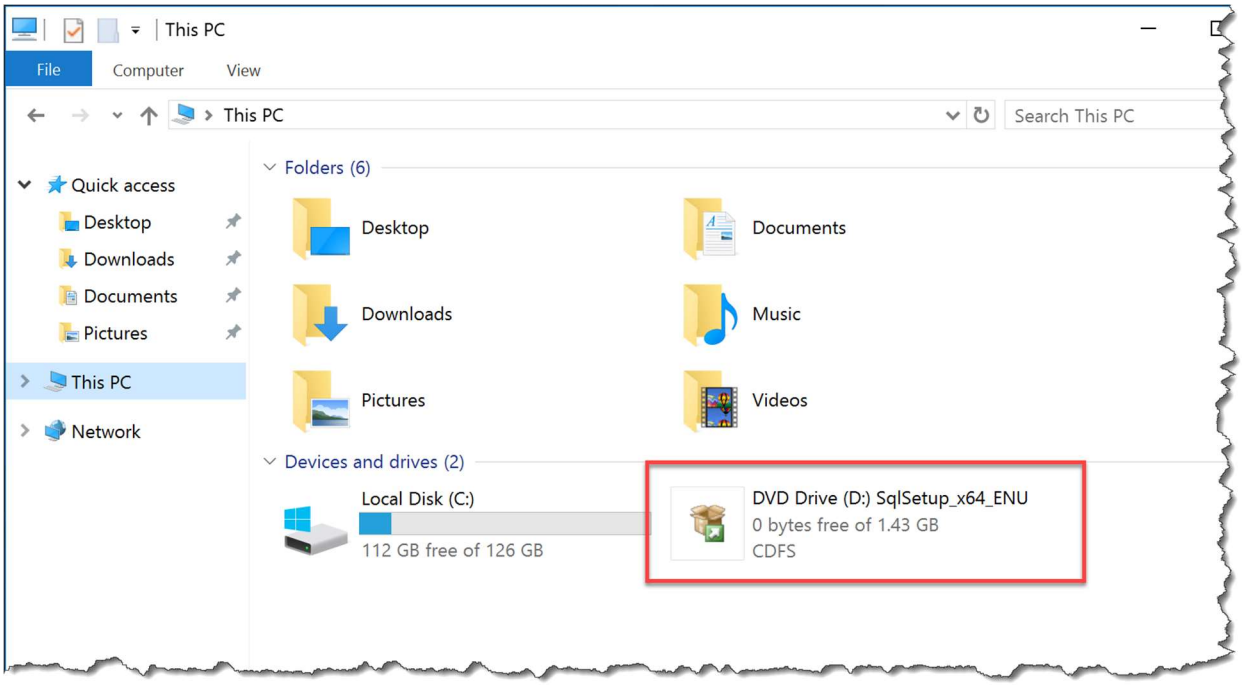
Azure DevOps Server 2020 (AzDO) uses SQL Server for all its back-end storage. This part of the guide will walk you through installing SQL Server 2019 for AzDO.

Install SQL Server 2019

- Either insert your **SQL Server 2019 DVD** into the DVD drive or mount the **ISO image** into the DVD drive for your virtual machine.
- Start the machine
- Log in as an administrator

- Open Explorer.exe
- Navigate to **This PC** or another view that will show you all the drives on your machine

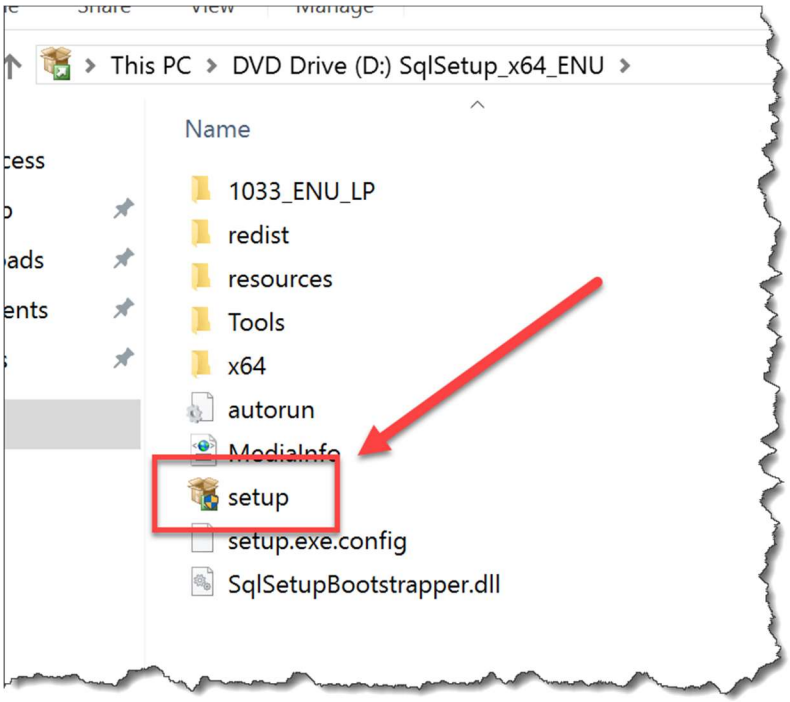
You should see a view that's similar to the screenshot below and you should see a DVD drive with **SqlSetup_x64_ENU** or another similar version of SQL Server in the drive.



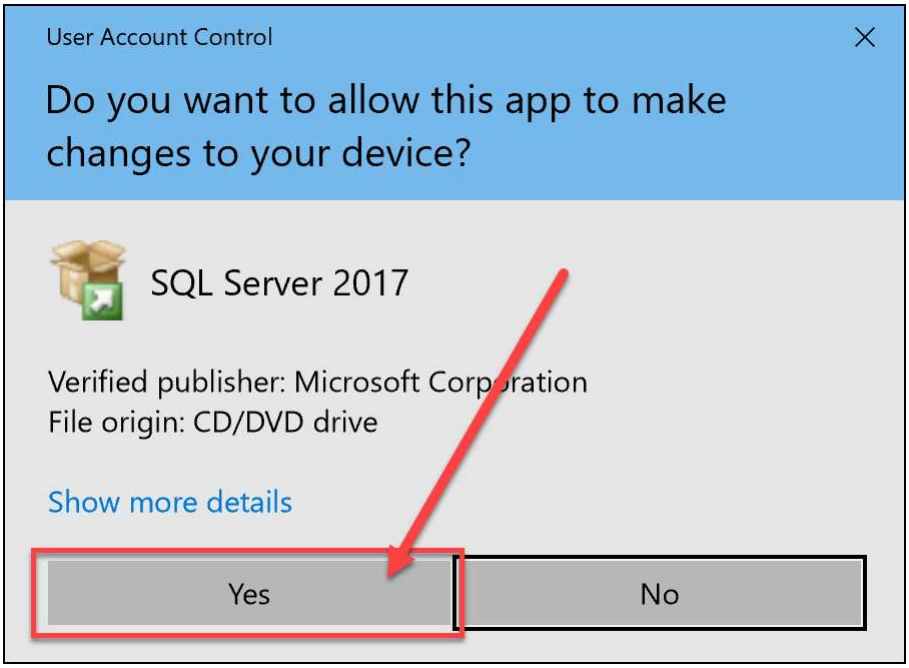
- Double-click on the DVD drive with the SQL Server installer

You should see the contents of the SQL installer.

- Double-click **setup** to run the installer

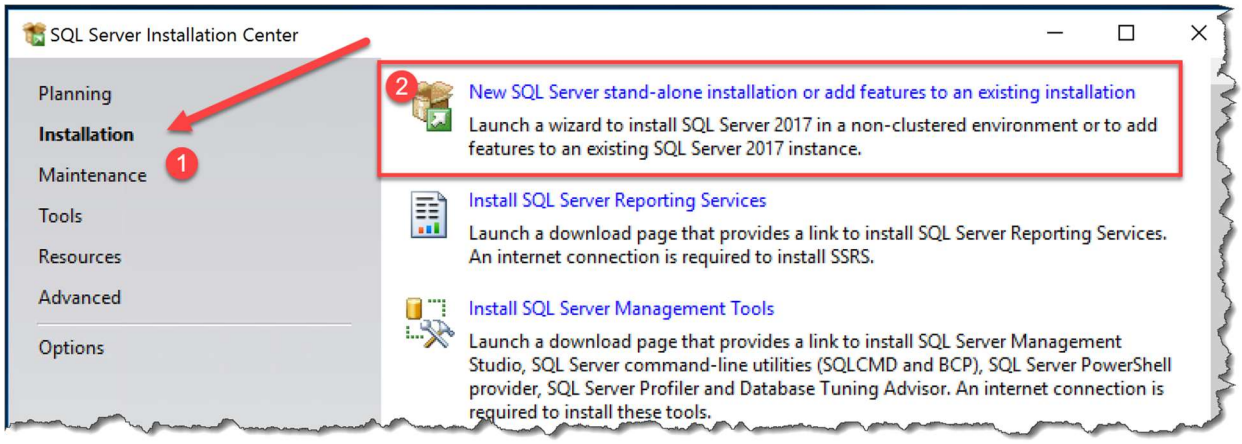


- Click **Yes** on the User Account Control dialog



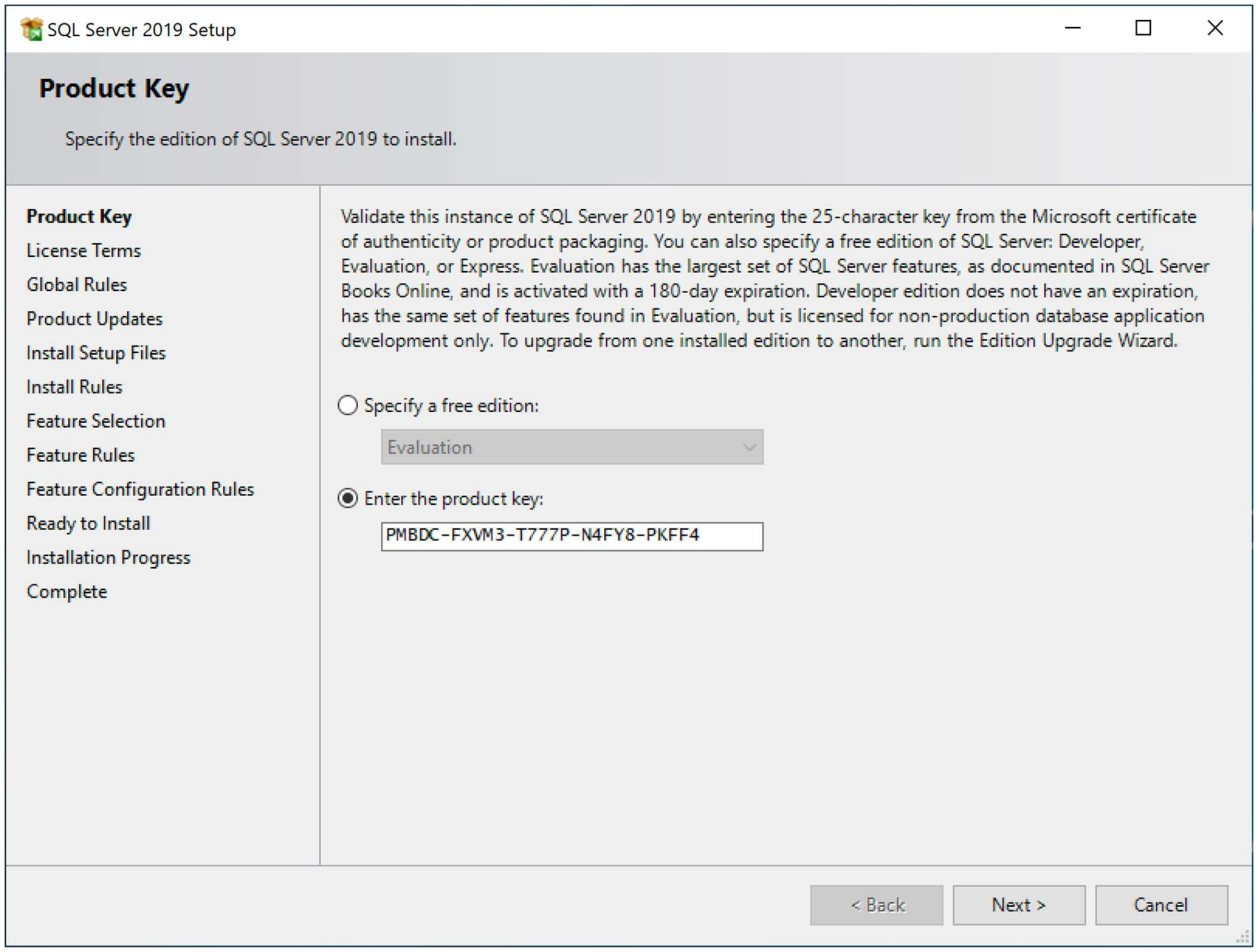
You should now see the **SQL Server Installation Center** window.

- In the left column, click the **Installation** link
- Click **New SQL Server stand-alone installation or add features to an existing installation**



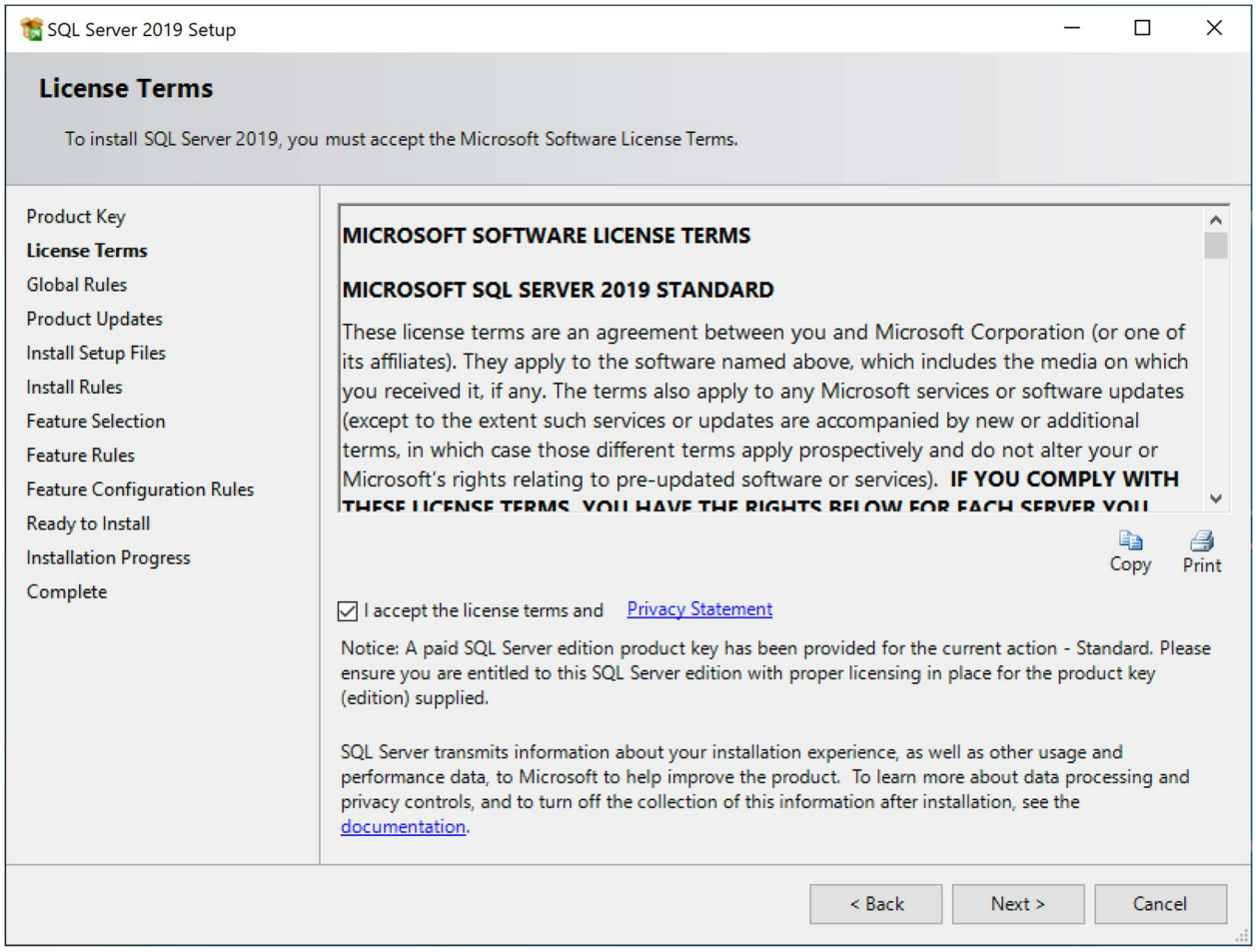
The first page of the SQL Server 2019 Setup wizard will prompt you for a product key. It should already be filled in with a product key.

- Click **Next**



You should now be on the **License Terms** page.

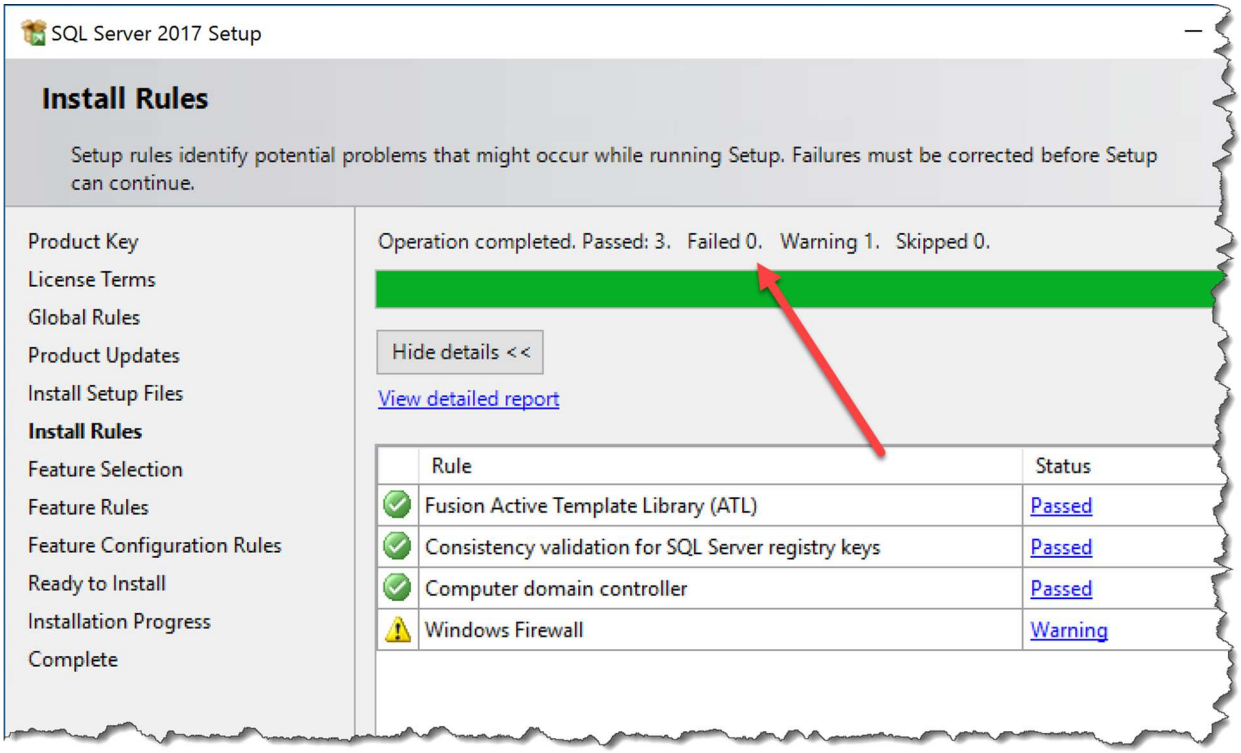
- Read the license terms carefully and completely.
- Ponder the implications of the license terms for you and for your business
- Really deeply digest the true meaning and intent of the license terms
- Consider consulting an attorney to review the license terms to better understand the terms
- Contact Microsoft to negotiate any amendments to the software license terms that your attorney recommended
- When you, your attorney(s), and Microsoft have come to an agreement about your license terms, check the **I accept the license terms** checkbox
- Click **Next**



The install wizard will work on a few things and run some pre-install checks. If you see a screen that looks similar to the one below, make sure that you don't have any failures.

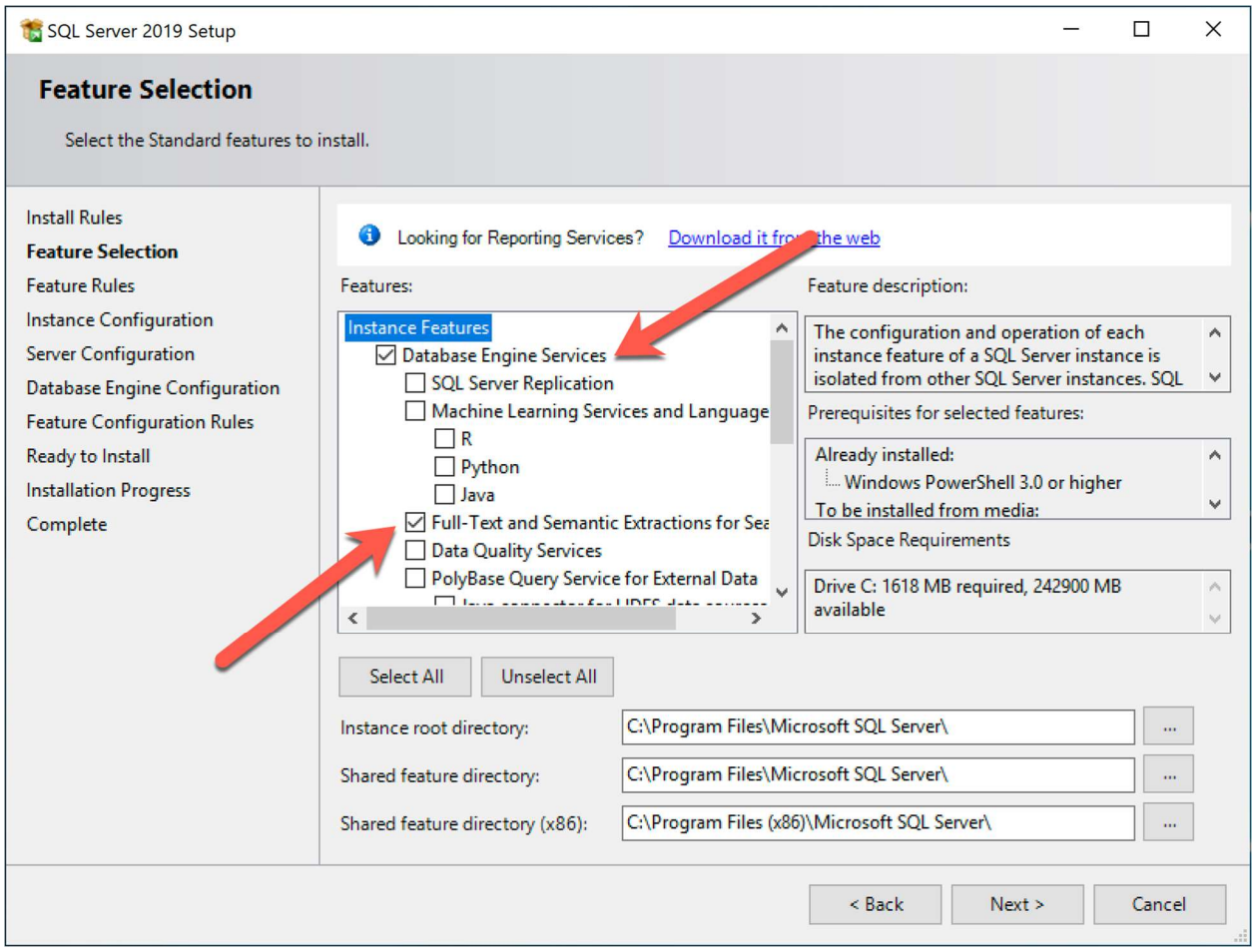
NOTE: by default, this screen is skipped if all the pre-install checks have passed.

- Verify that there are zero failed rules
- Click **Next**



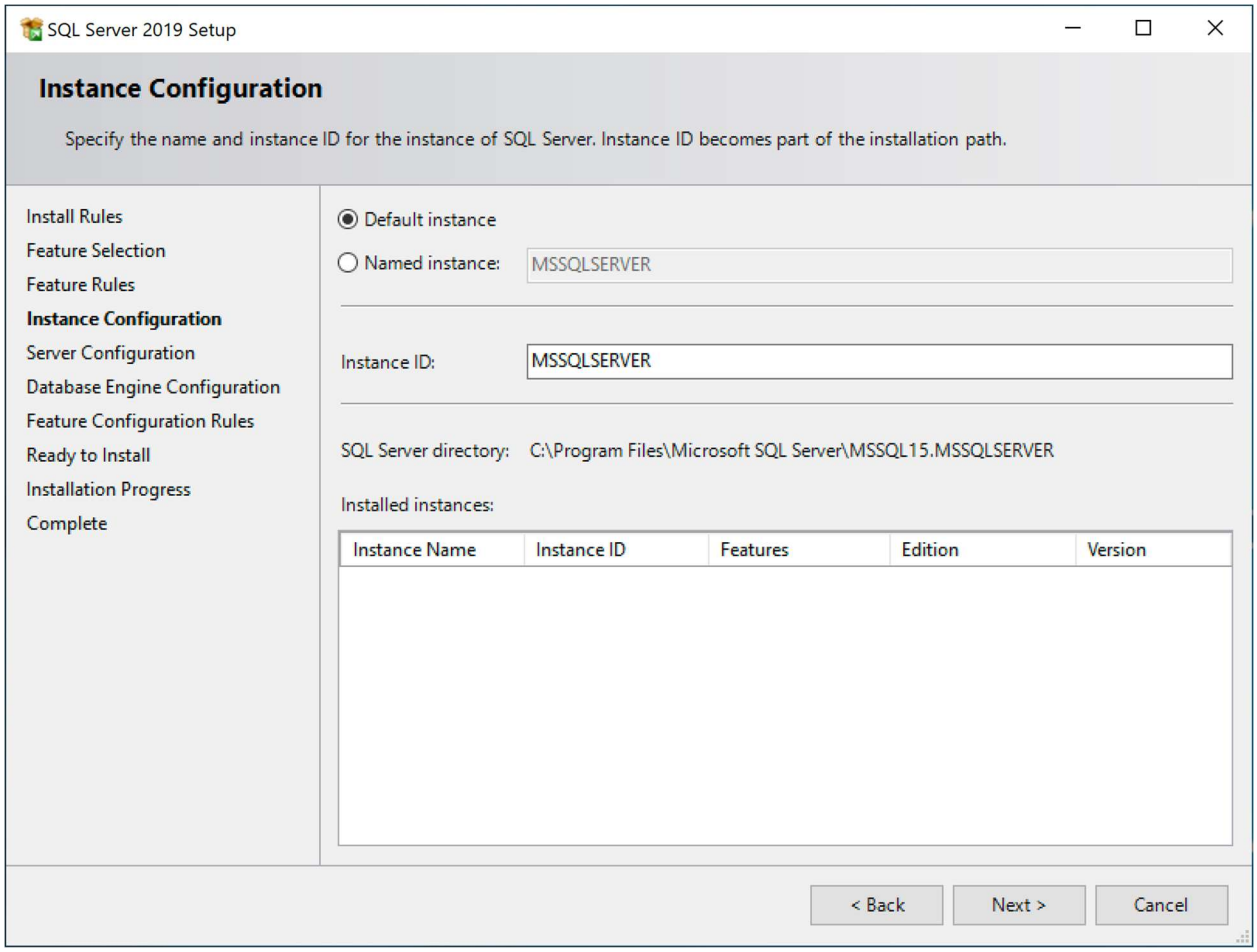
You should now be on the **Feature Selection** page.

- Check the checkbox for **Database Engine Services**
- Check the checkbox for **Full-Text and Semantic Extractions for Search**
- (Optional) If you are planning to run TFS and SQL Server on separate machines, check the checkbox for **Client Tools Connectivity**
- Click **Next**



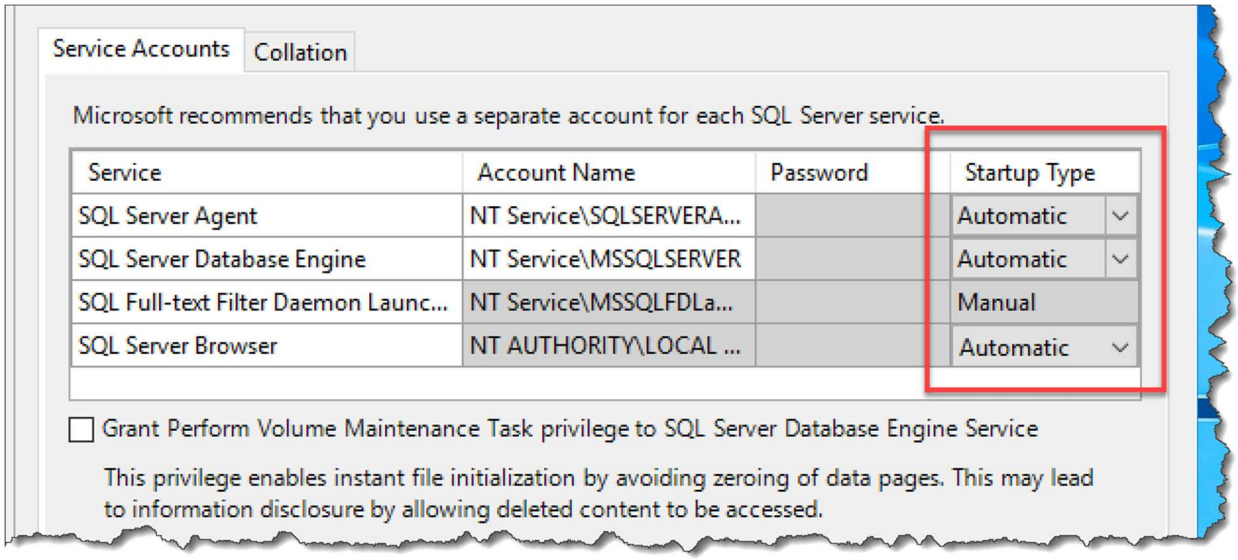
You should now see the **Instance Configuration** page of the install wizard. Do yourself a favor and just keep the default values.

- Click **Next**



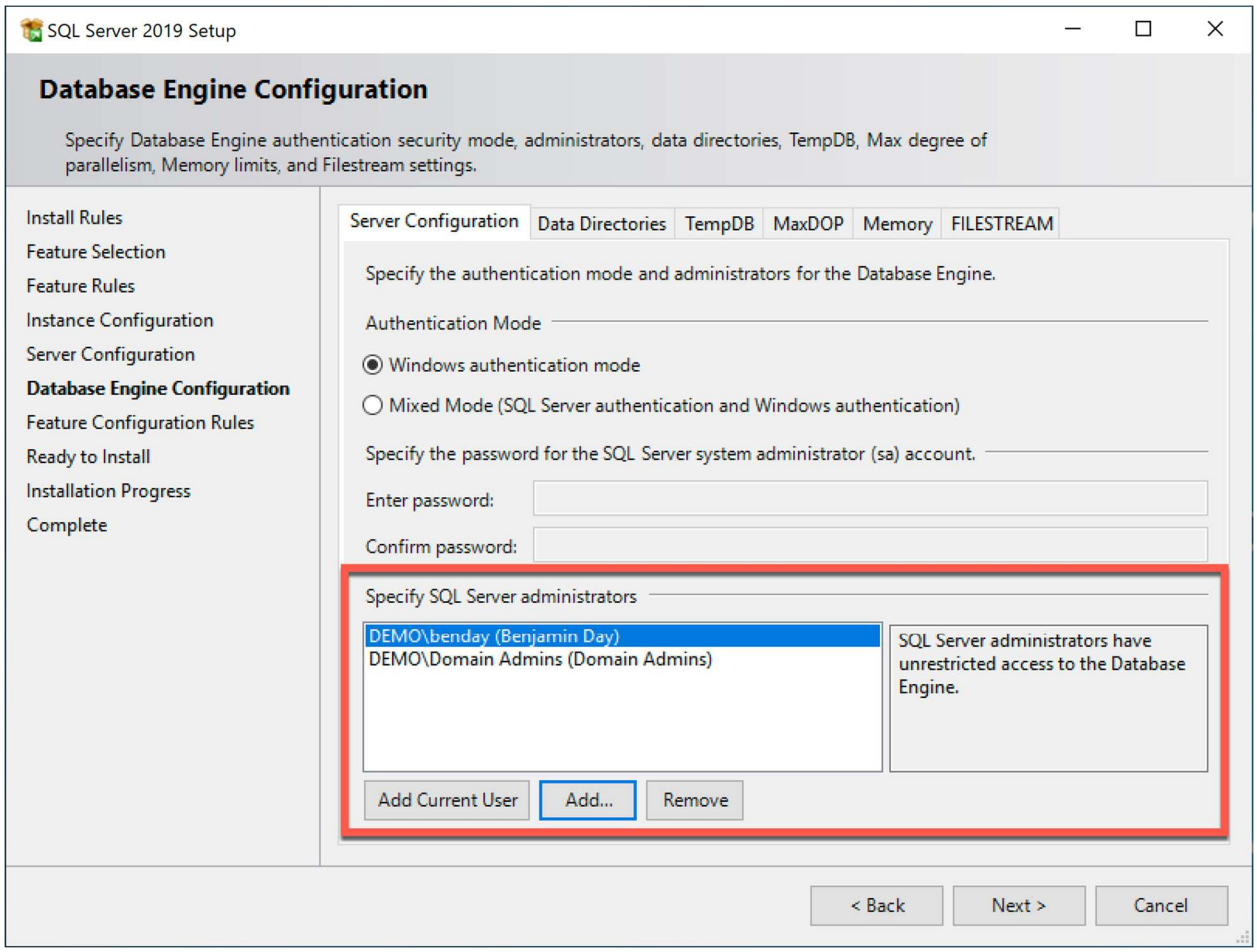
On the **Server Configuration** page, you'll specify the startup value for each service.

- Set **Startup Type** to **Automatic** for each service (NOTE: full-text search doesn't let you change the startup value so you can ignore it)
- Click **Next**



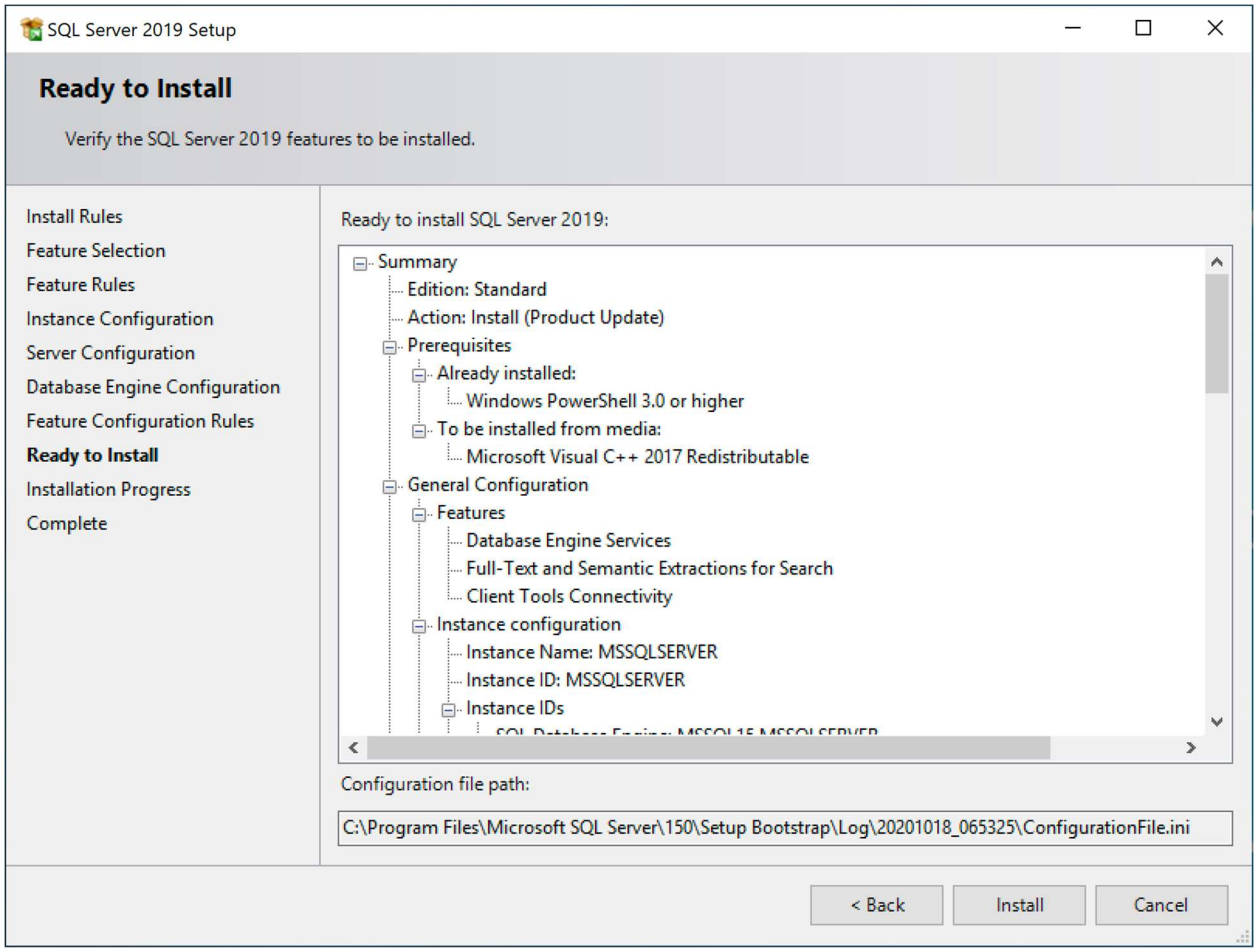
You should now be on the **Database Engine Configuration** page. On this page, you'll pretty much just be specifying who should be an administrator on your new SQL Server instance. By default, the installer does not automatically make Windows administrators members of the sysadmin group in SQL Server. (I think that's silly.)

- Click the **Add Current User** button
- Click the **Add…** button
    - On the search dialog, search for **Domain Admins**
    - Click **OK**
- Verify that the list of administrators is accurate
- Click **Next**

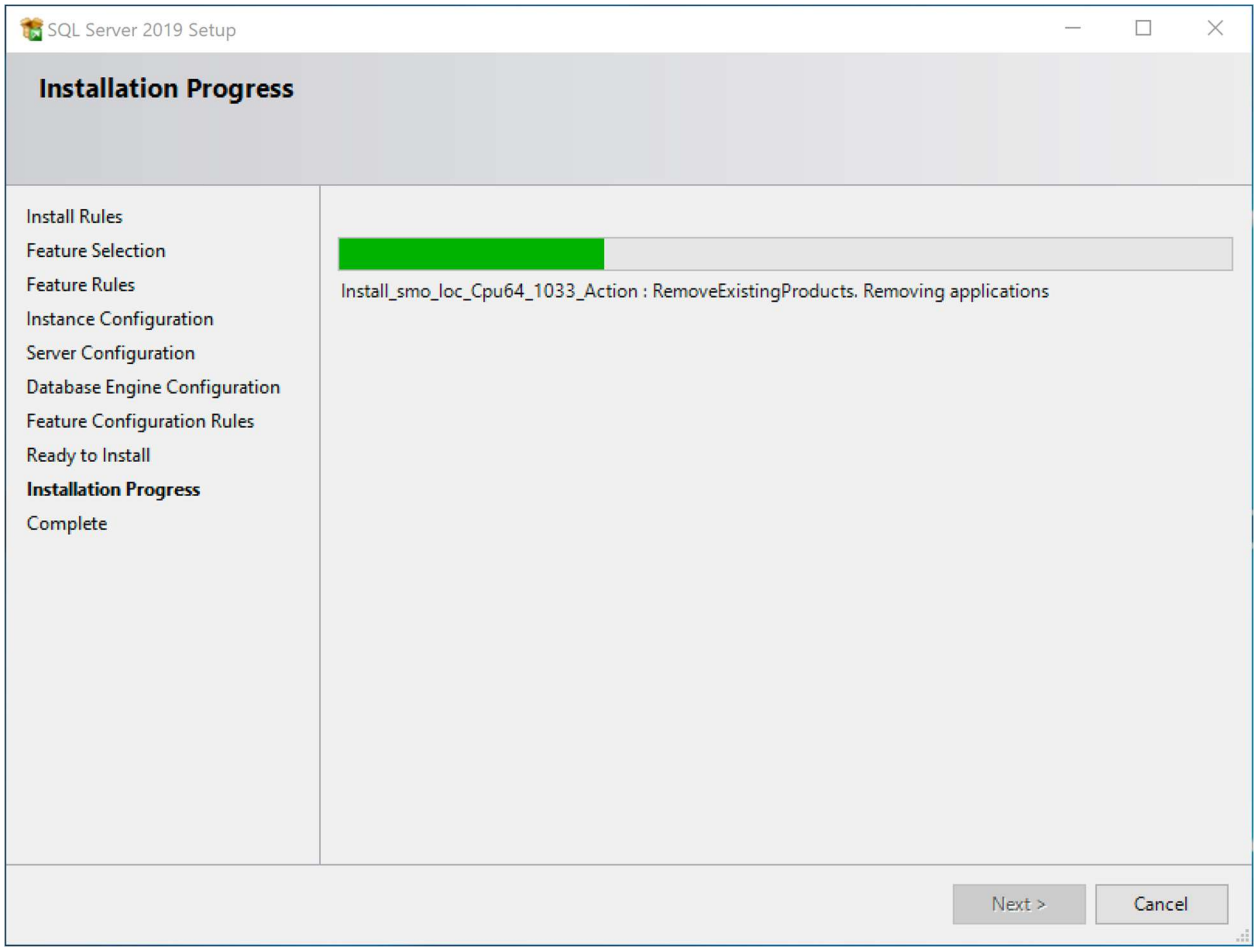


You should now be on the **Ready to Install** page of the dialog and you should see a summary of what is going to be installed.

- Click **Install**

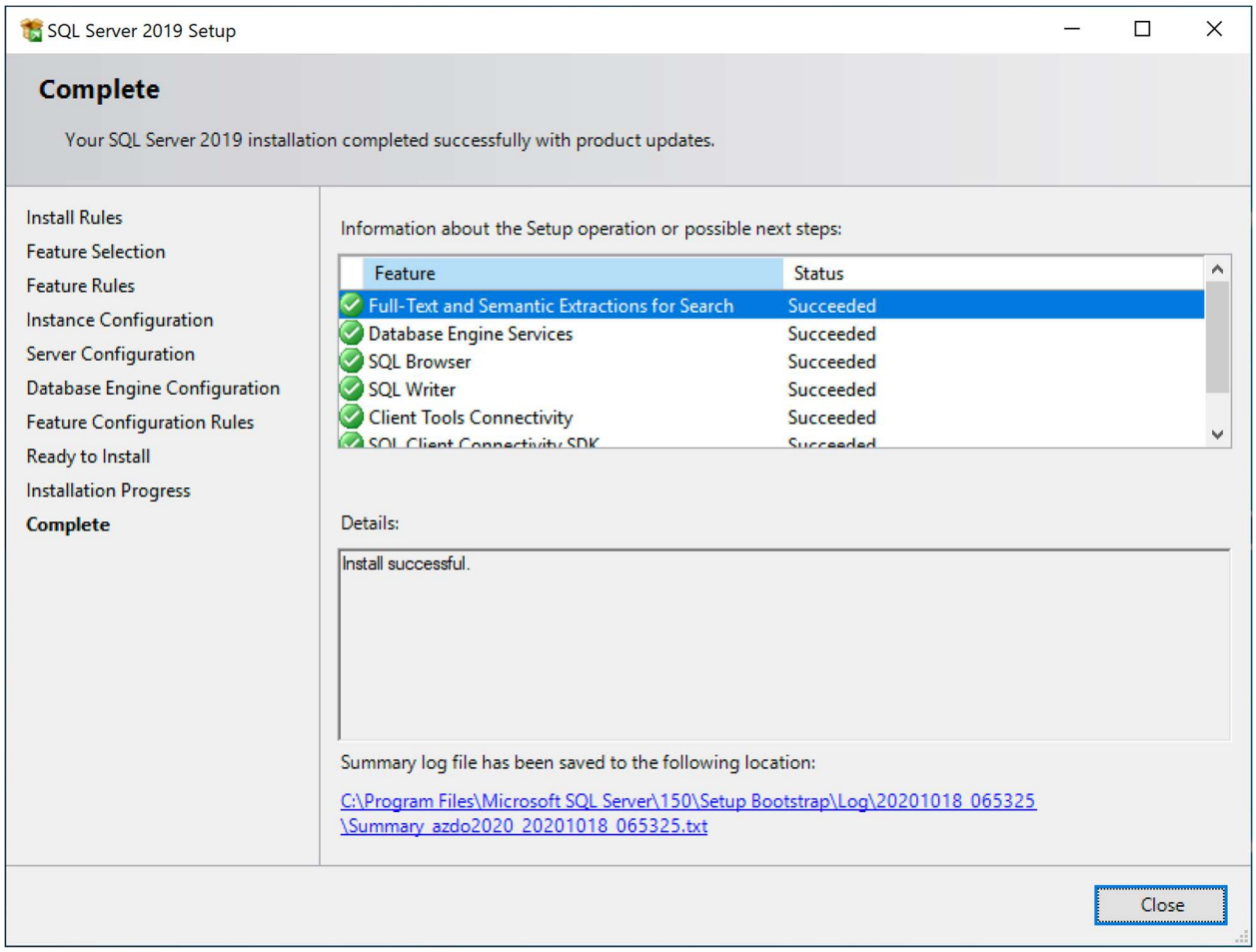


The installer should now be running.



When the installer is done, you'll see a message saying **Complete**.

- Click **Close** to exit



SQL Server 2019 is now installed.

Chapter 4: Install Azure DevOps Server 2020

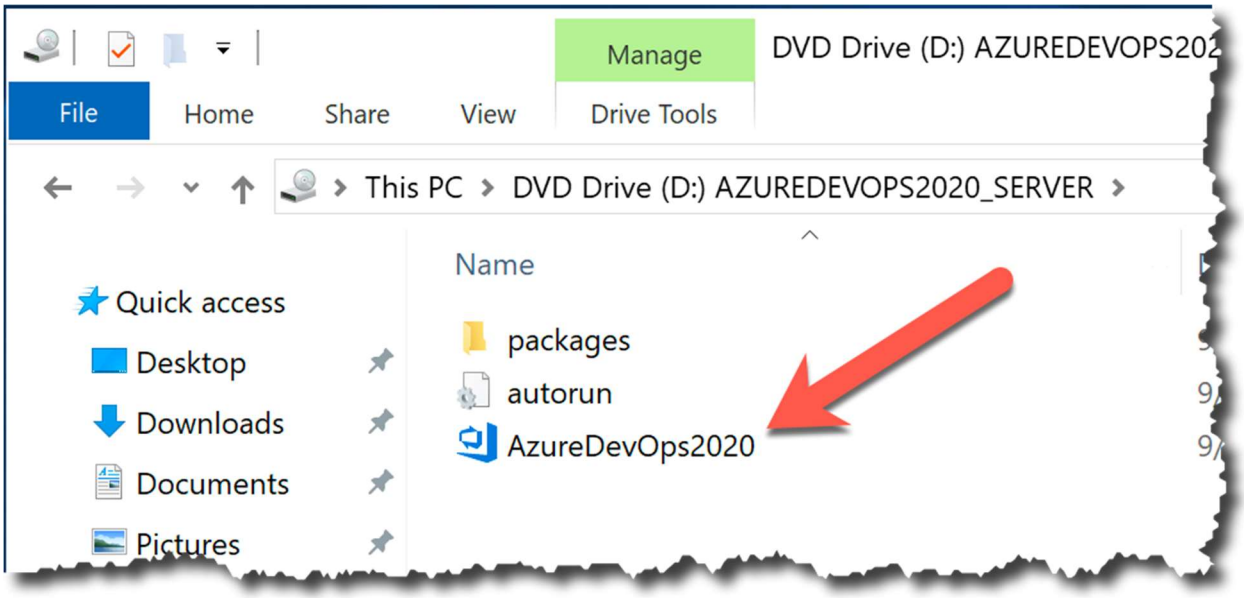
Introduction

Now that Windows and SQL Server are installed, you're ready to install Azure DevOps Server 2020 (AzDO). You'll probably want to create three domain accounts for use by the various pieces of AzDO: Azure DevOps Service (*domain*\azdoservice) and Azure DevOps Build (*domain*\azdobuild).

- If you're installing this on a Hyper-V virtual machine with dynamic memory enabled, change the **minimum amount of RAM to 2GB** (at least temporarily) to allow Azure DevOps Server 2020 to install along with SQL Server.
- Gather the username and passwords for the 2 AzDO service accounts (see above)
- Log on to the server using a user account with Administrator privileges

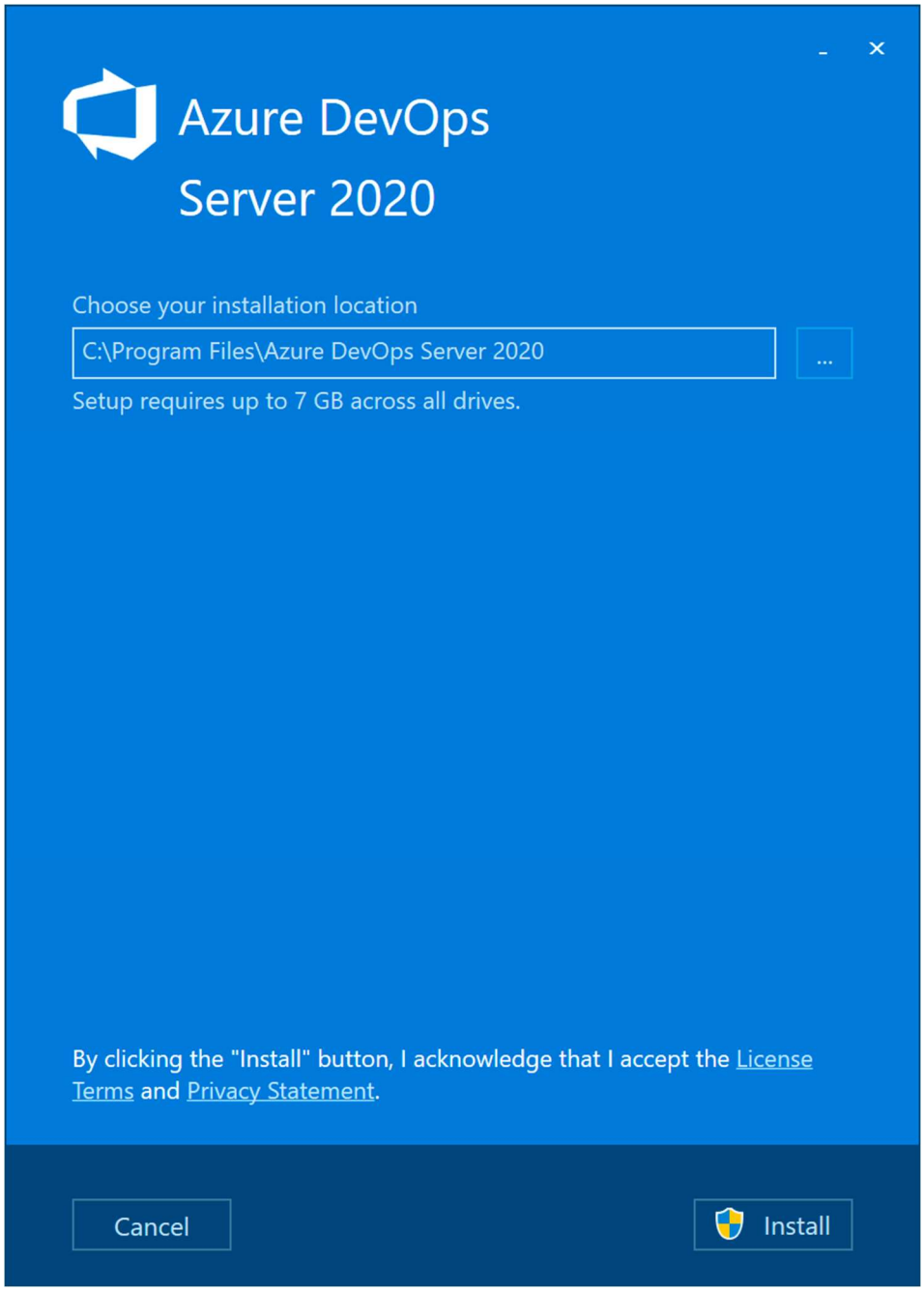
Run the Installer

- Mount the ISO image or insert the installer DVD
- Using Windows Explorer (explorer.exe), navigate to the installer directory



- Run **AzureDevOps2020.exe**

You should see the **Azure DevOps Server Setup** dialog.

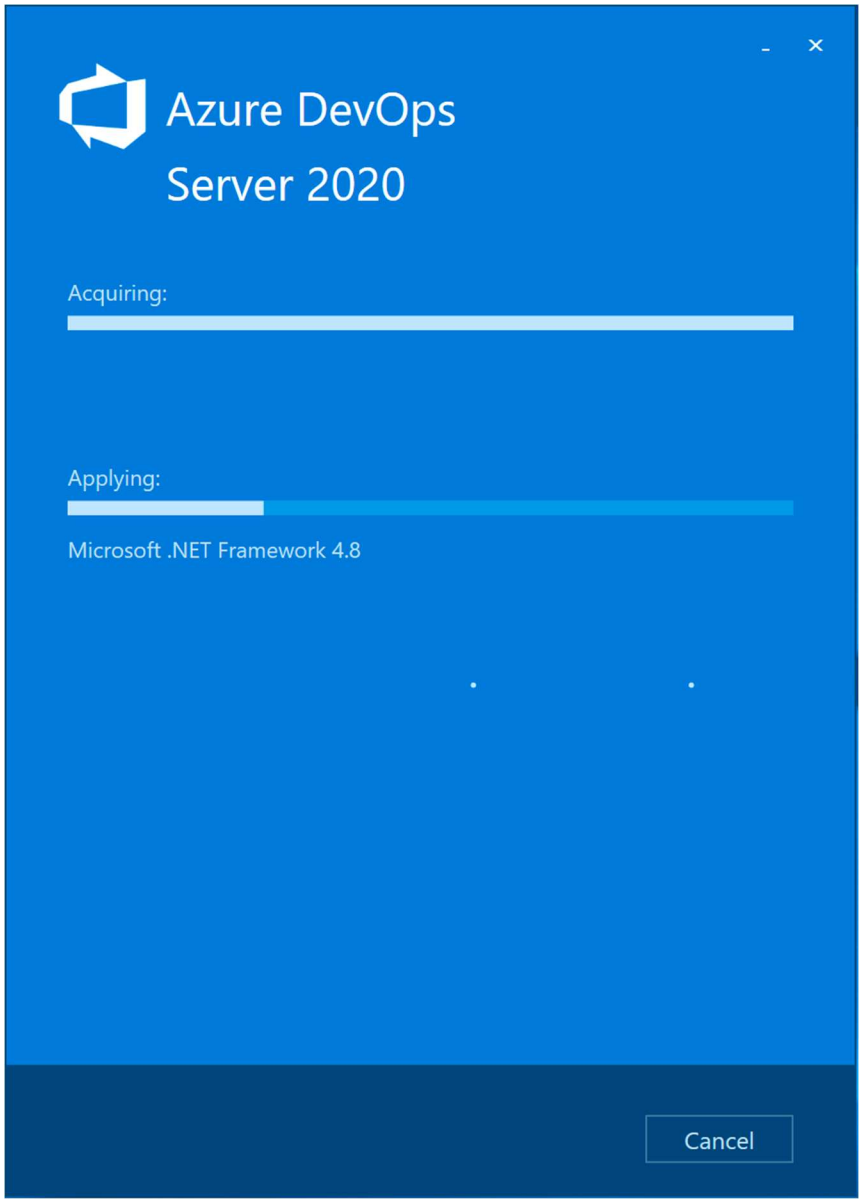


- Click **Install**

You'll see a User Account Control dialog.

- Click **Yes**

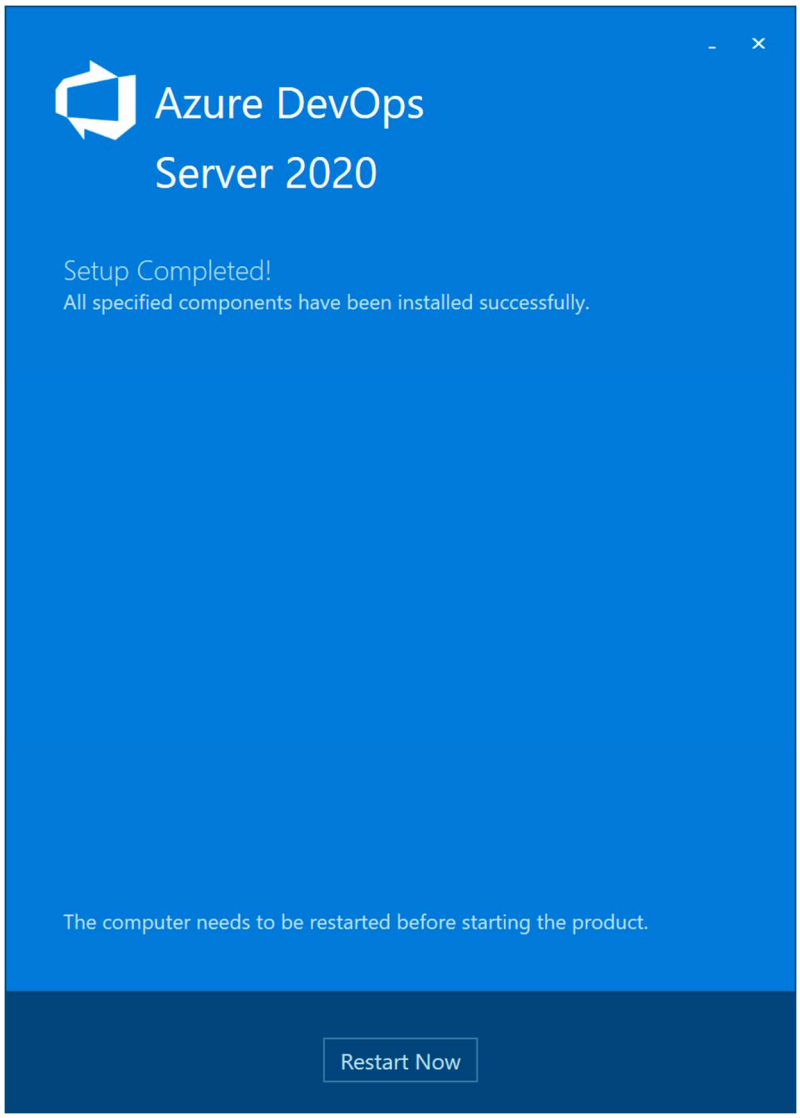
The installer will run and start to copy files to your disk.



After running a long while, you'll get a screen similar to the one below with the message **Setup Completed** and prompting you to restart.

NOTE: There's also a chance that you might not see this screen if your server does not need a reboot. If this happens, you'll be shown the **Azure DevOps Server Configuration Center** dialog. If you see that dialog, skip ahead in this guide a few pages.

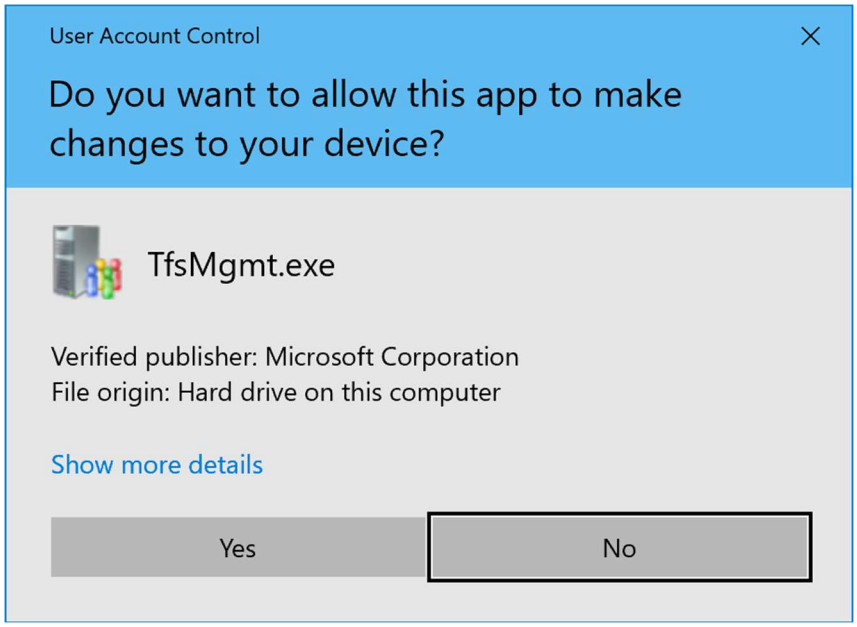
- Click the **Restart Now** button
- If your server doesn't restart on its own, **reboot your server**



After the server has rebooted…

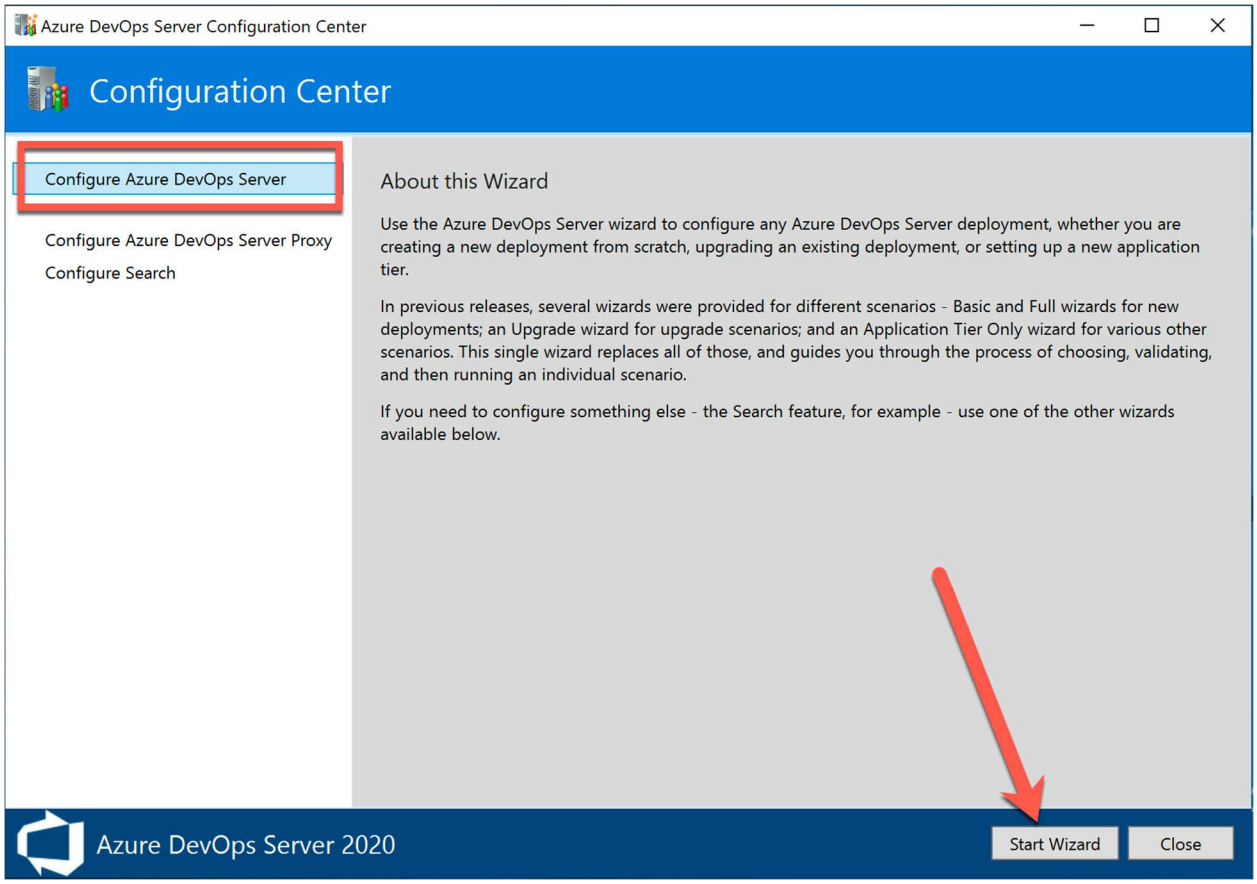
- Log in to the server. NOTE: be sure to use the same user account that you were using before the reboot

You'll be welcomed by a User Account Control dialog.



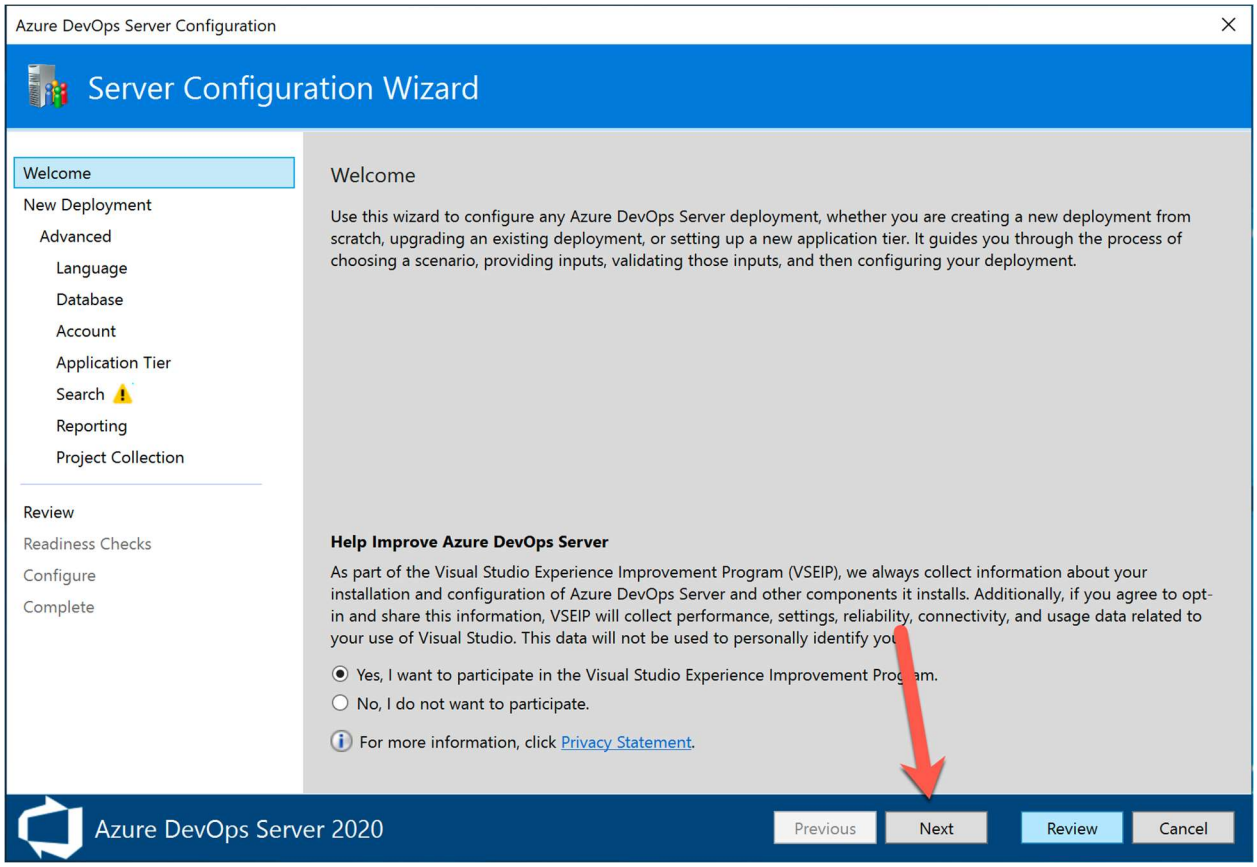
- Click the **Yes** button

After running for a while (probably a long while), you should see the **Azure DevOps Server Configuration Center**.



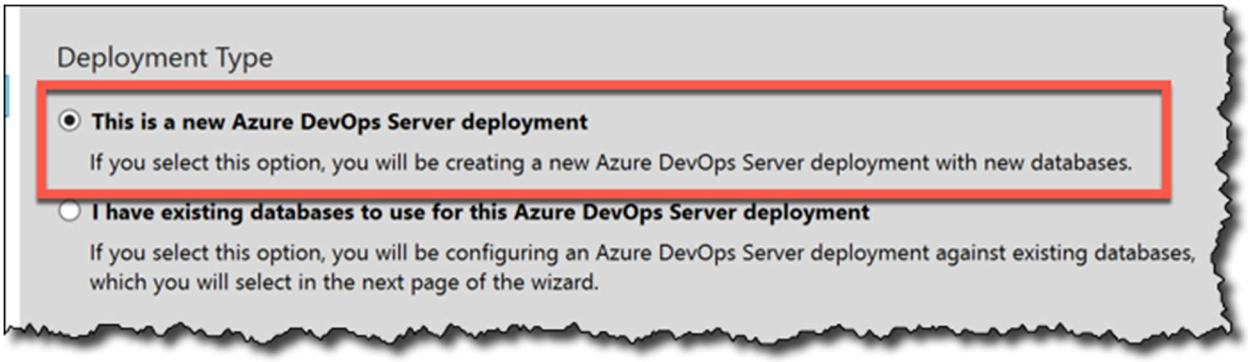
- Choose **Configure Azure DevOps Server**
- Click **Start Wizard**

You should now be on the welcome page.



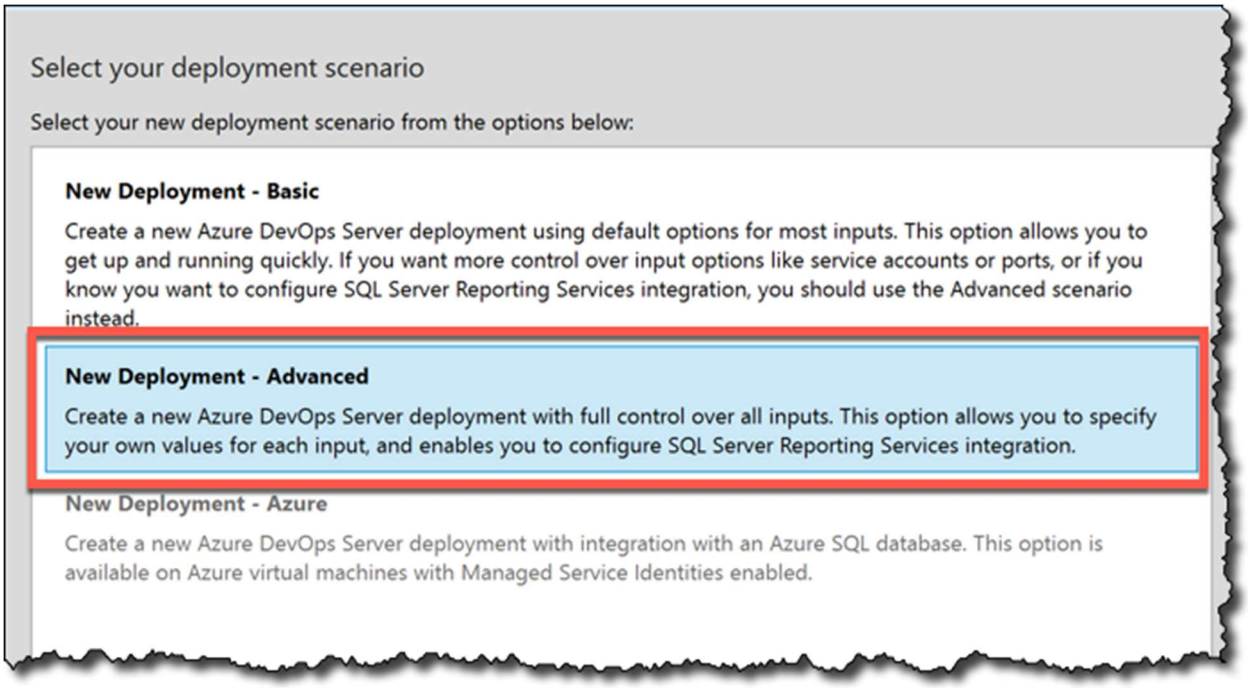
- Choose **Yes, I want to participate in the Visual Studio Experience Improvement Program**
- Click **Next**

You should now be on the Deployment Type page.



- Select **This is a new Azure DevOps Server deployment**
- Click **Next**

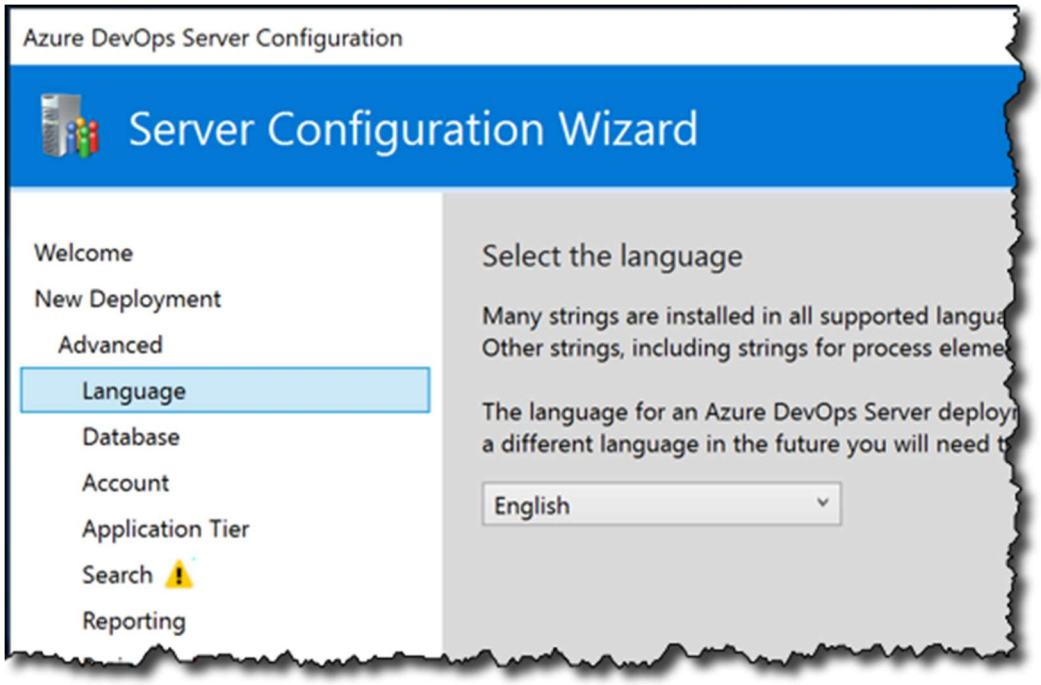
On the **Select your deployment scenario** page, you'll be prompted to choose between a 'New Deployment – Basic' or 'New Deployment – Advanced' deployment. Unless you are installing AzDO on an Azure Virtual Machine, the 'New Deployment – Azure' option will be disabled.



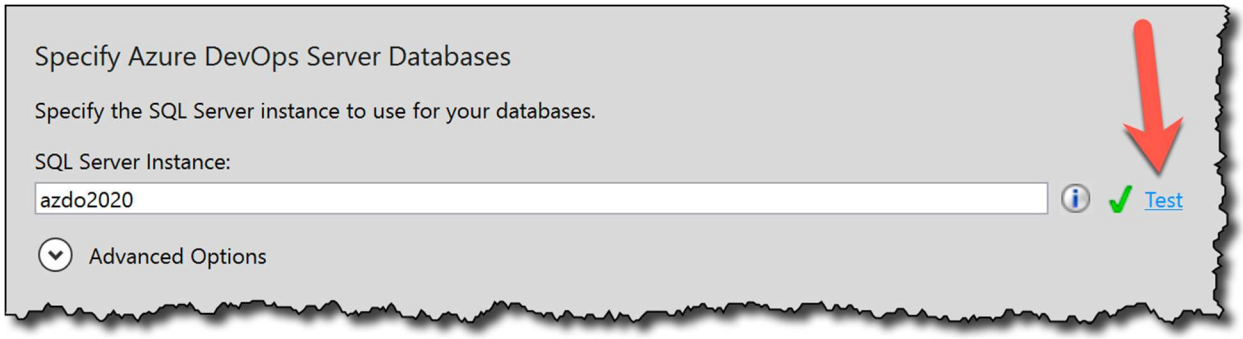
- Choose **New Deployment – Advanced**
- Click **Next**

Choose your language.

- Choose the language you want to use from the drop-down list
- Click **Next**

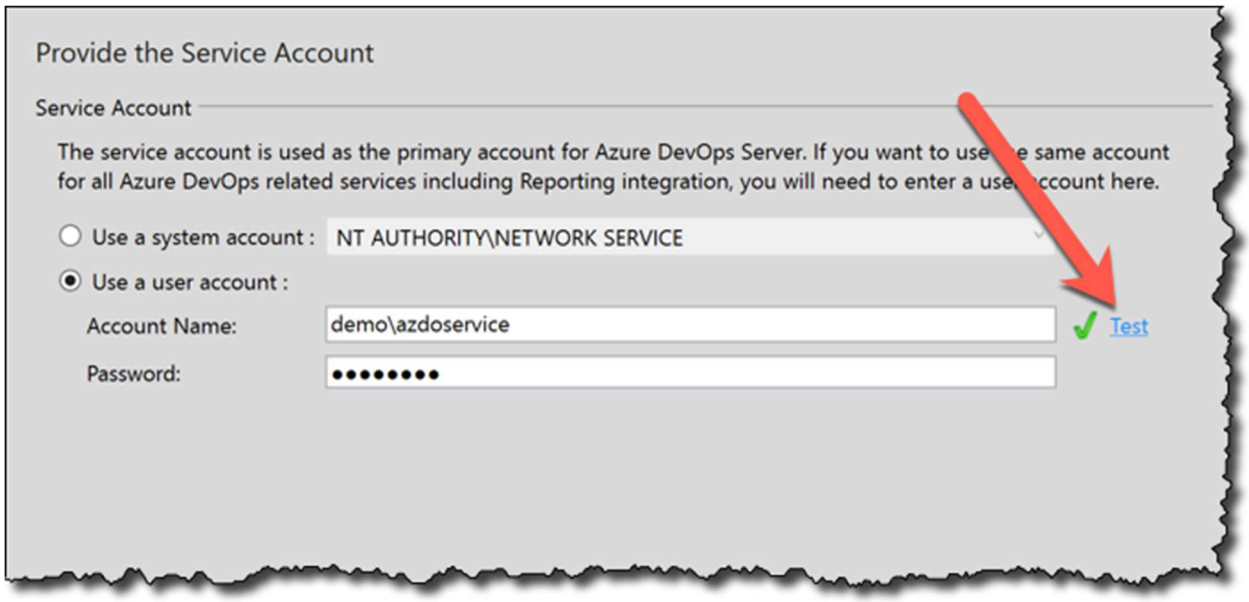


The **Specify Azure DevOps Server Databases** page prompts you to choose your SQL Server database. This guide assumes that you installed SQL Server Standard on the same machine as AzDO.



- To the right of the **SQL Server Instance** textbox, click the **Test** link to verify the connection to SQL Server
- Confirm that the test passes
- Click **Next**

You should now see the Service Account page. You can choose to run AzDO as a system account but I find that this makes permissions management – more specifically, permissions *debugging* – much harder later on when you start doing automated builds, automated deployments, and automated testing. My recommendation is to run AzDO as a separate service account.



- Choose **Use a user account**
- In the **Account Name** textbox, type the fully-qualified name of the service account. Example: *demo\azdoservice*
- In the **Password** textbox, enter the password for the service account
- Click the **Test** link to verify the credentials are correct
- Click **Next**

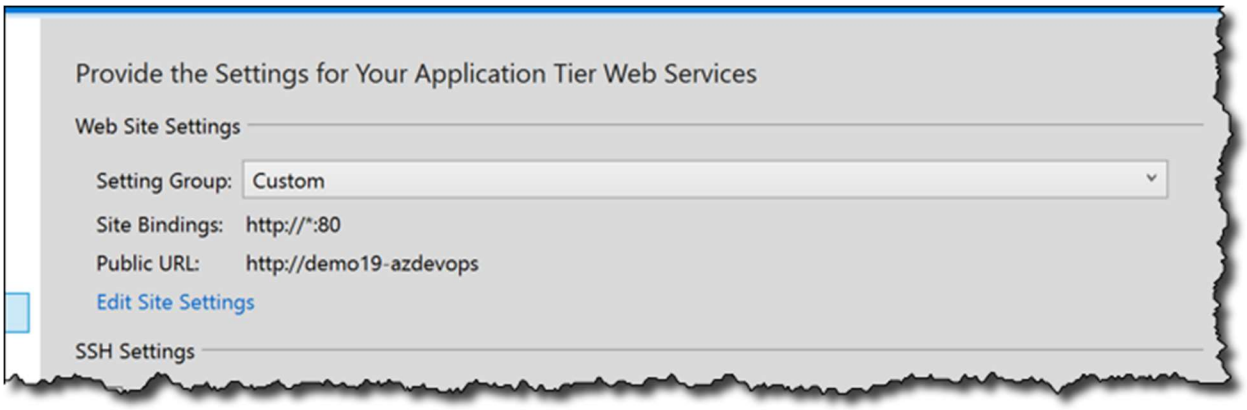
You should now see a page prompting you for the configuration of AzDO in IIS. You have some options about configuring SSL with AzDO but there's some complexity here that we're going to skip over for now.

For the sake of simplicity, we're going to configure AzDO to run without SSL. In "real life" you'll want to acquire an SSL certificate for your server and install it in Internet Information Services (IIS) in order to make your AzDO traffic more secure. But that's wildly out of scope for this guide. If you'd like help with this, I like money and I help people with their Azure DevOps problems for money. I'd be overjoyed if you contacted me at info@benday.com and told me that you wanted to give me money to help you with your technical problems. ☺

Also, while I'm shamelessly talking about money and technology, I've got an Azure DevOps video course that's for sale at https://courses.benday.com. You might want to check that out. You might also want to check out the classes that I teach at https://www.benday.com/training. Just sayin'.

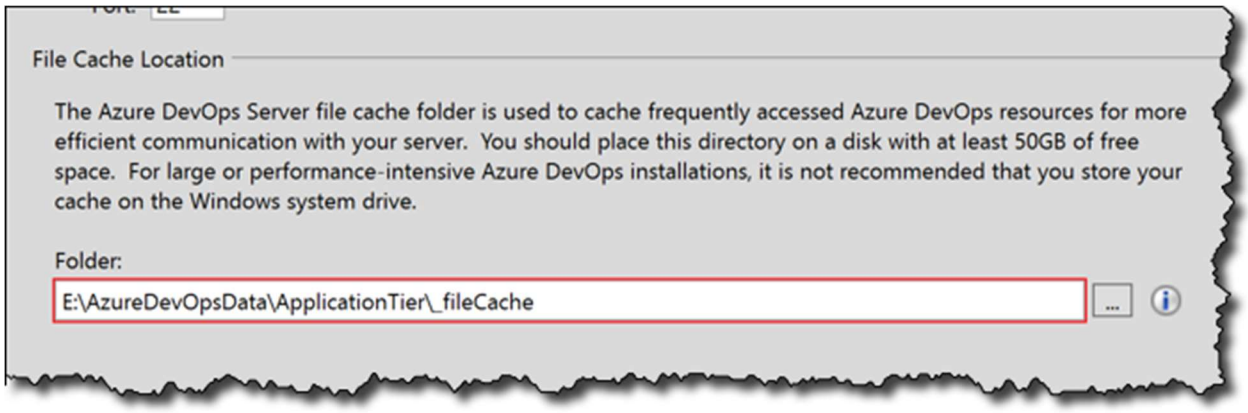
Anyway. Continuing on with the settings for your application tier web services.

The default option should be set to **Custom**.



- Make sure **Setting Group** is set to **Custom**
- If Setting Group is NOT set to Custom, just give up now. (Have you considered raising goats as a career?)

(Optional) At the bottom of this page, there's a section for **File Cache Location**. AzDO caches files for efficiency. The contents of this directory can become impressively large. For performance reasons and for disk space management reasons, you probably should put this on a separate disk – ideally on a different "spindle" – than your system/operating system drive.



- (Optional) Change the **Folder** path to reference the desired location and disk.
- Click **Next**

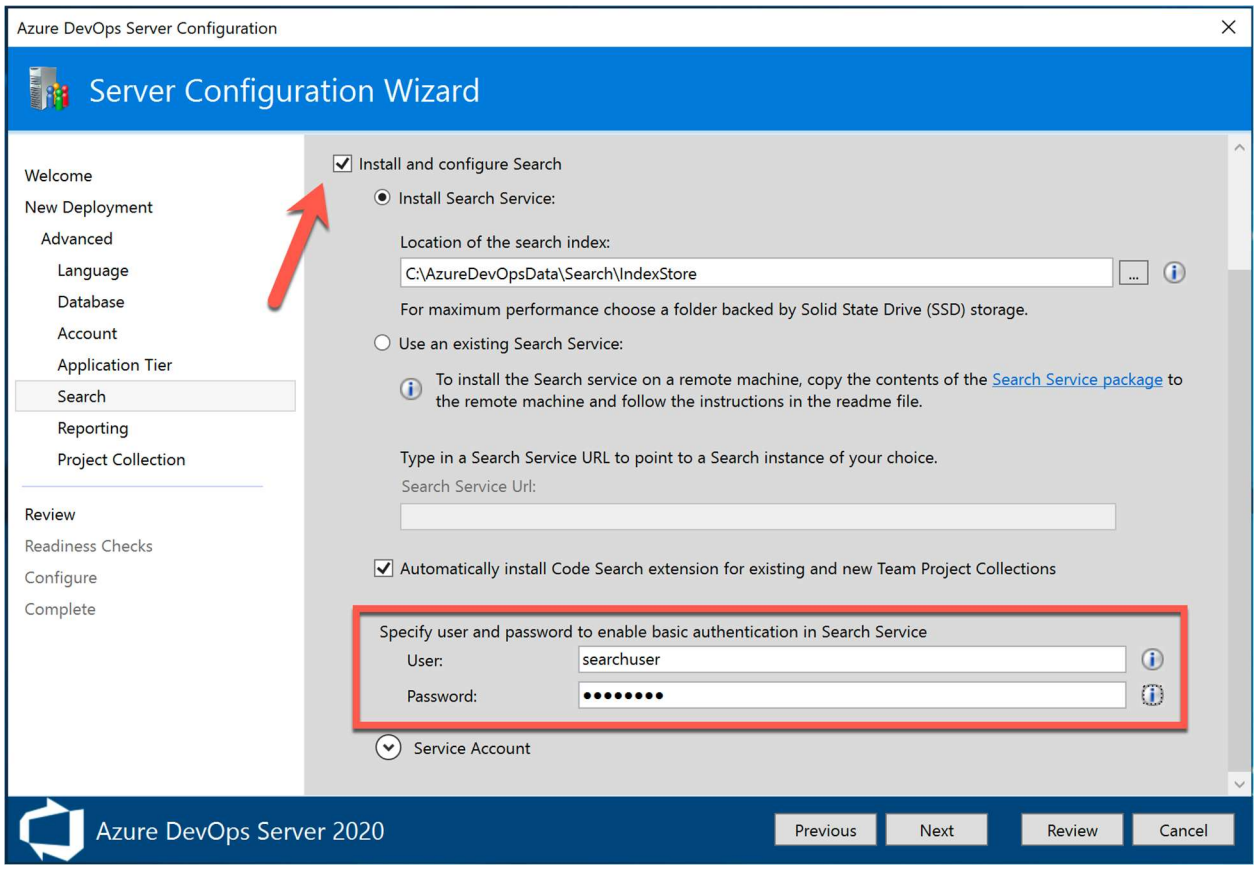
You should now be on the **Provide Search configuration settings** page of the wizard. This is an optional feature.

Option 1: If you *do not* want to install Search:

- Uncheck **Install and configure Search**
- Click **Next**

Option #2: Install Search

- Check **Install and configure Search**



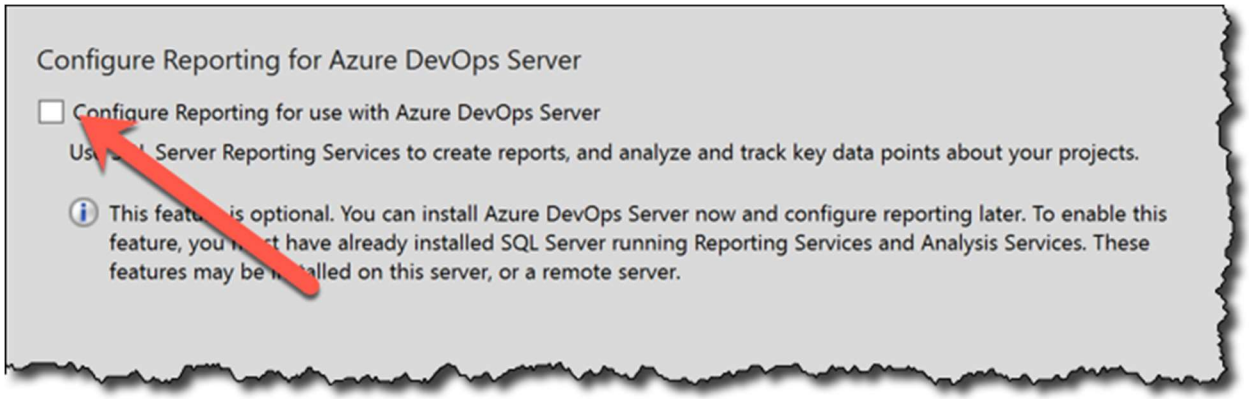
- Choose **Install Search Service**
- Set the **Location of the search index** to the drive and folder you want to use for search. For performance reasons, you'll probably want to keep this on a different drive than the system drive. If your AzDO installation is large and busy, you may want to put this on its own drive by itself.

- Under Specify user and password to enable basic authentication in Search Service, enter a user name and password. This is NOT a Windows username. I'd recommend using the username **searchuser** and then entering a password of your choice.
- Click **Next**

You should now see the **Configure Reporting for Azure DevOps Server** page. This is another optional feature. This feature is silly, adds next to no value, and pretty much just installs things that can break and are virtually guaranteed to break.

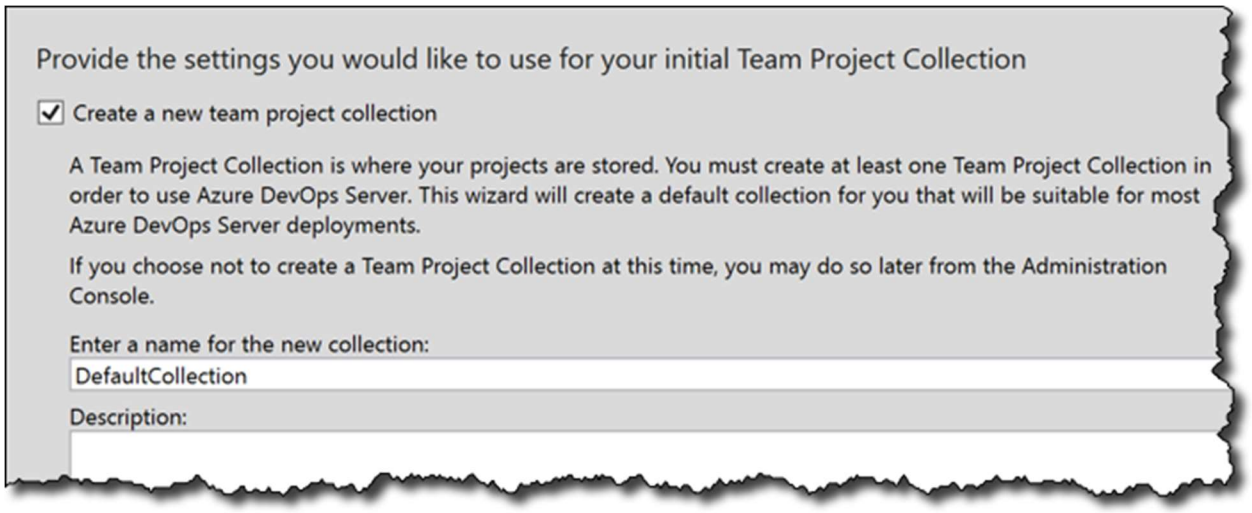
If you like wasting your time debugging problems with vague error messages for a feature that really doesn't provide you any real value or help your projects to be any more successful or give you any more interesting data with which to make decisions, then this feature is for you. Put another way, if you like working really hard (and I do mean REALLY HARD) to get rid of silly errors only to be horribly disappointed once the feature actually starts working and you see how mind-blowingly underwhelming it is, then this feature is for you.

This guide assumes that you're NOT installing support for SQL Server Reporting Services with AZDO.



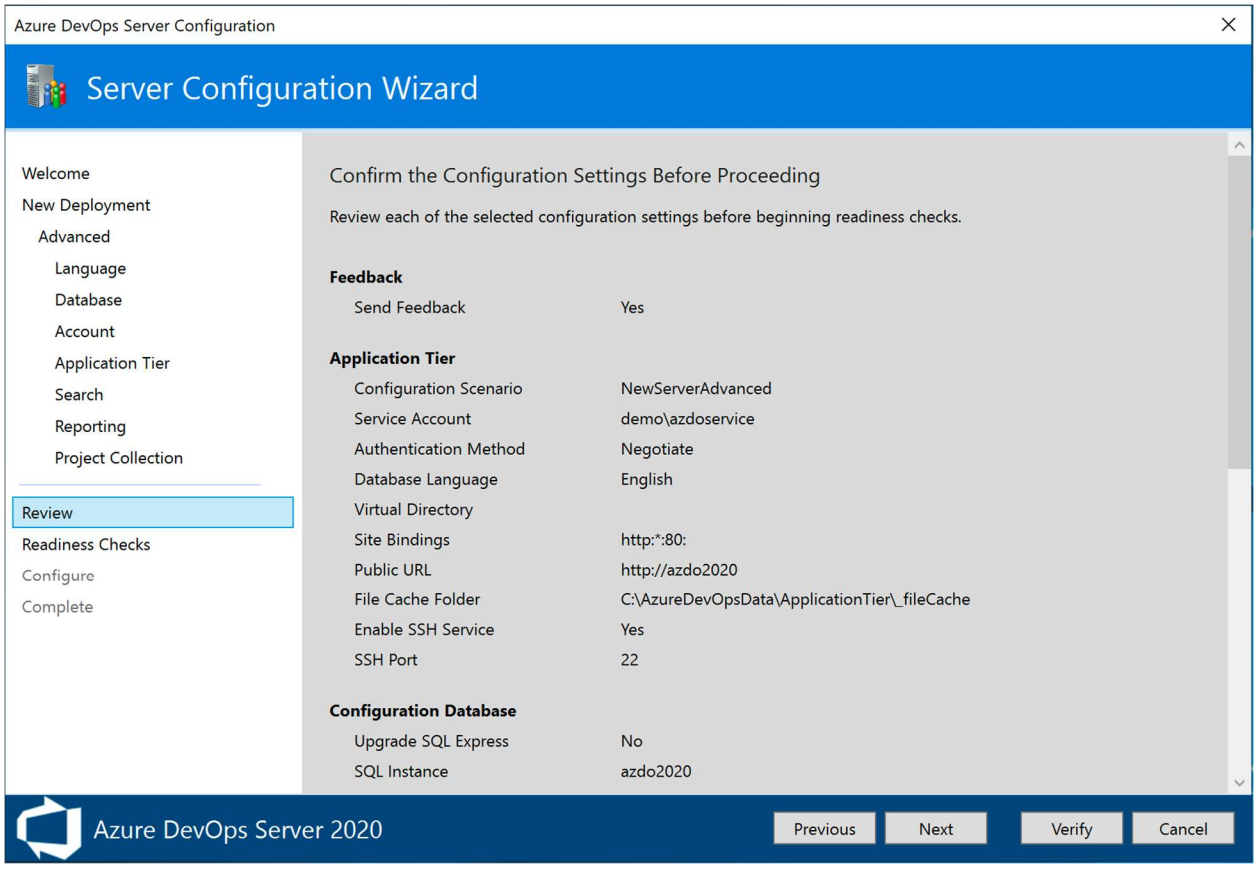
- Uncheck **Configure Reporting for use with Azure DevOps Server**
- Click **Next**

The installer will now prompt you to create a new Team Project Collection (TPC). The answer to this one (unless you're doing a migration) is yes.



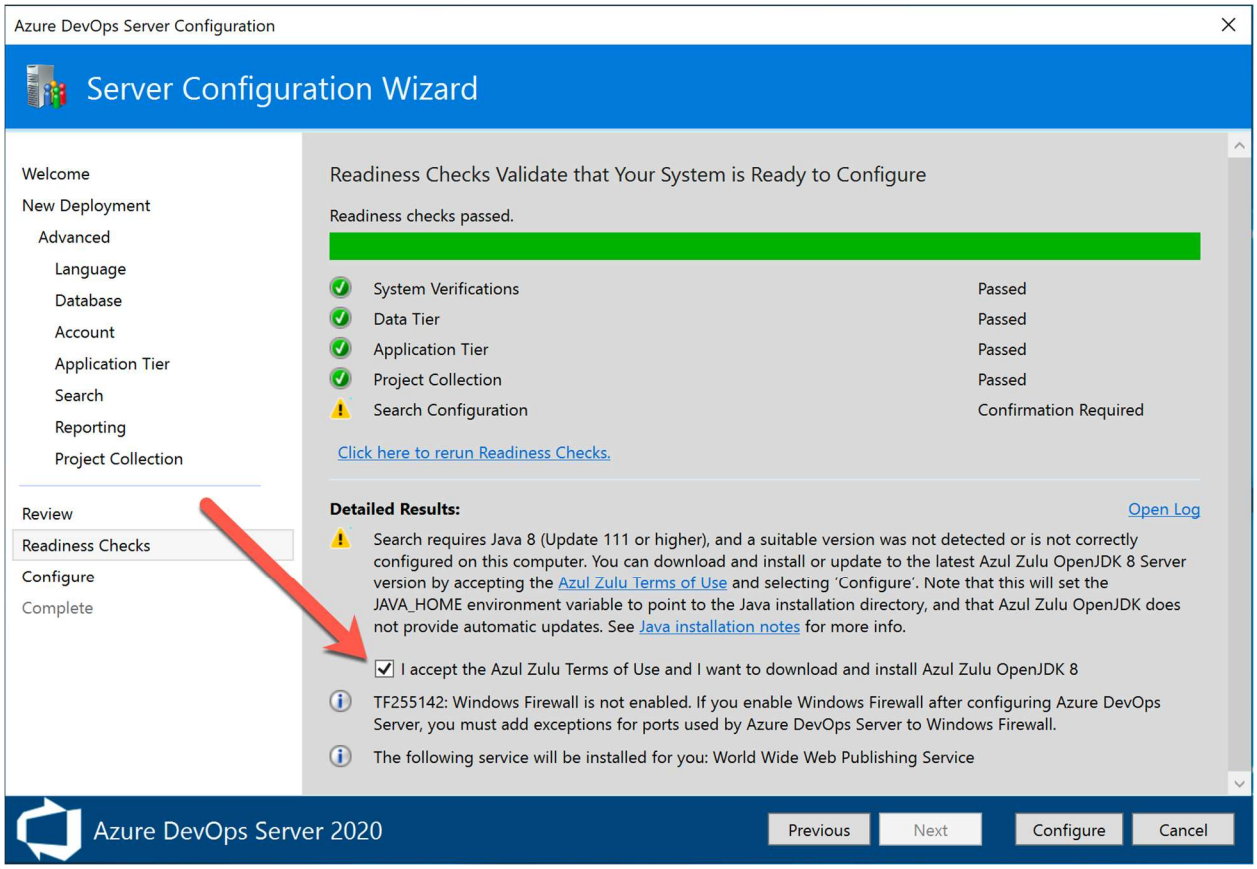
- Check **Create a new team project collection**
- Click **Next**

You should now be on the **Confirm the Configuration Settings Before Proceeding** page.



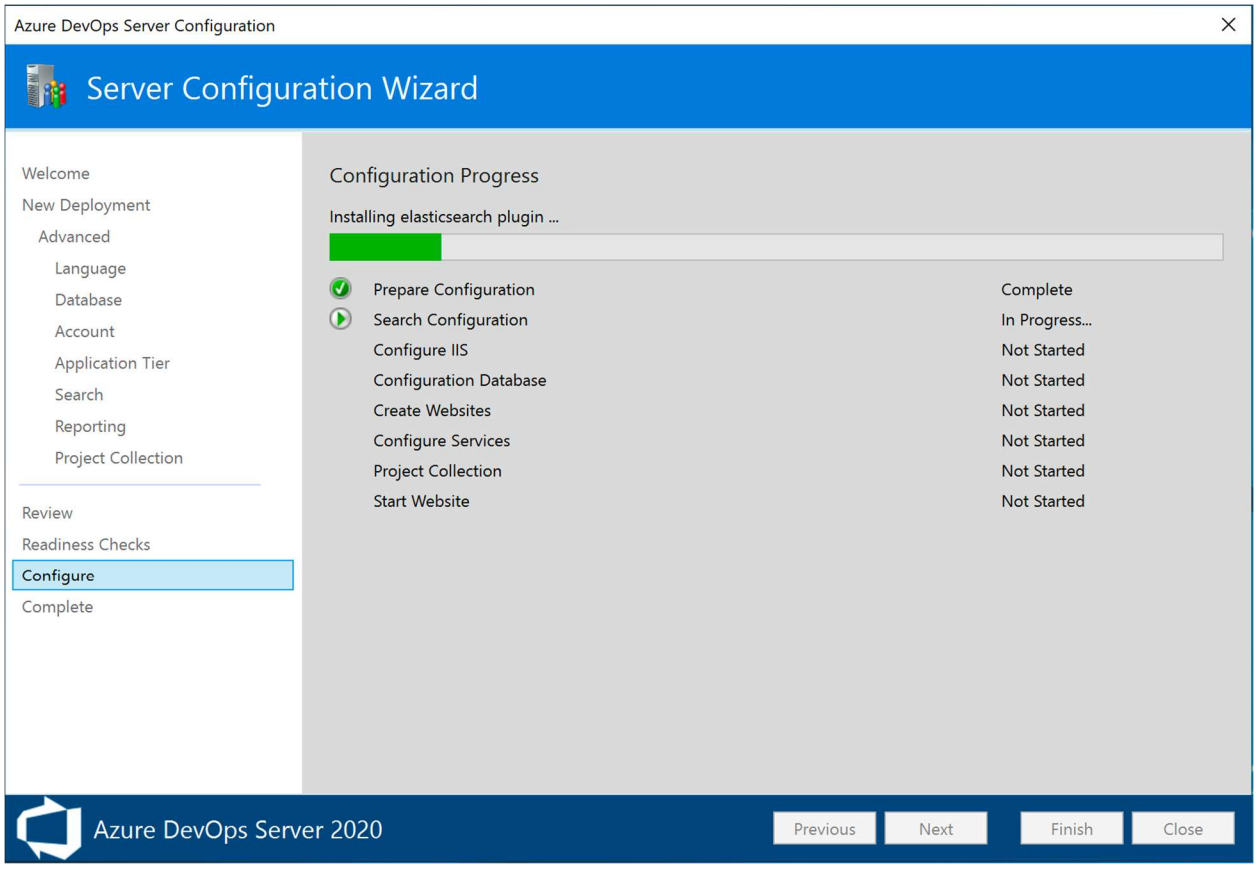
- Click **Next**

The installer will run some readiness checks. They should all come back as passed except for the Search Configuration item. The Search feature requires the Azul Zulu OpenJDK to be installed and this warning is prompting you to accept the licensing agreement for the JDK.

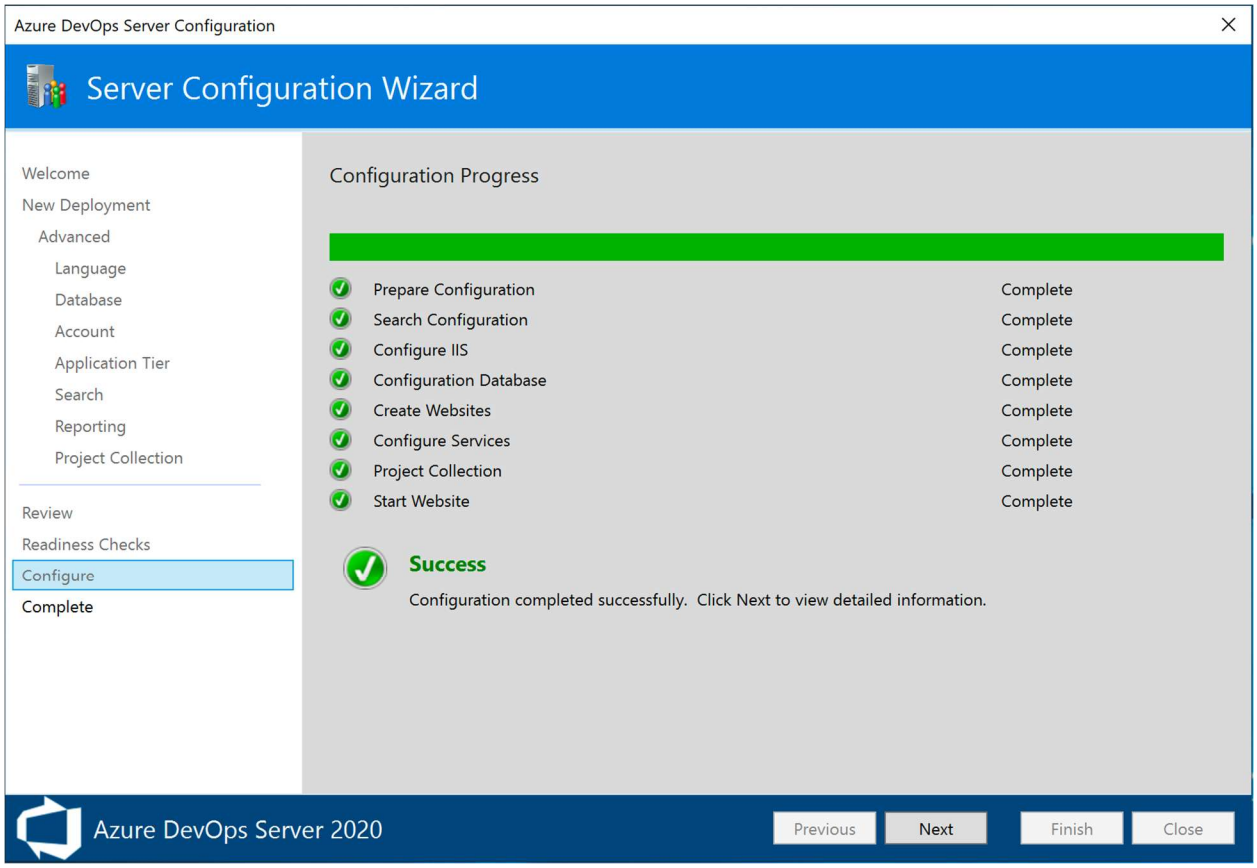


- Check **I accept the Azul Zulu Terms of Use…**
- Click the **Configure** button

The configuration process should now be running.

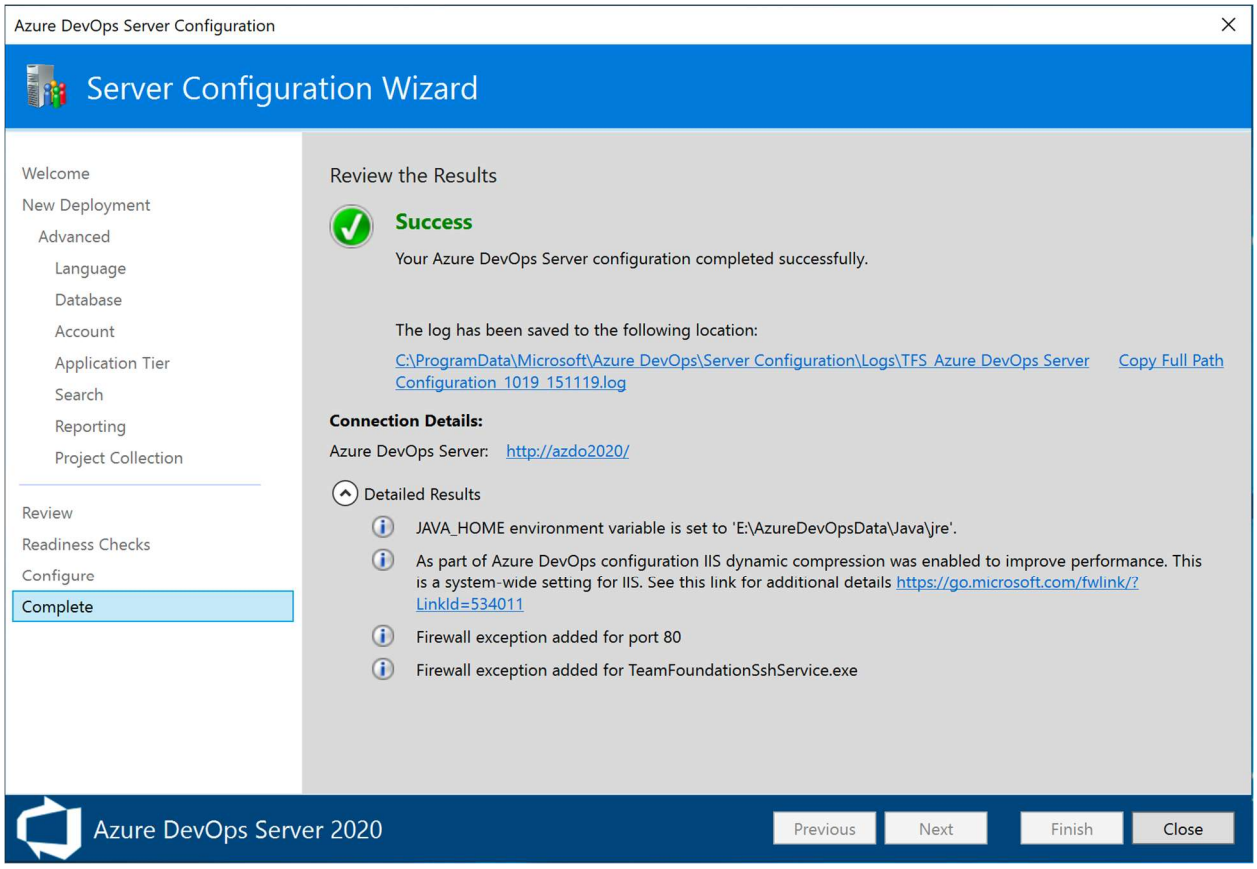


The configuration process should end with a message saying Success.



- Click **Next**

You should now be on the **Review the Results** page.

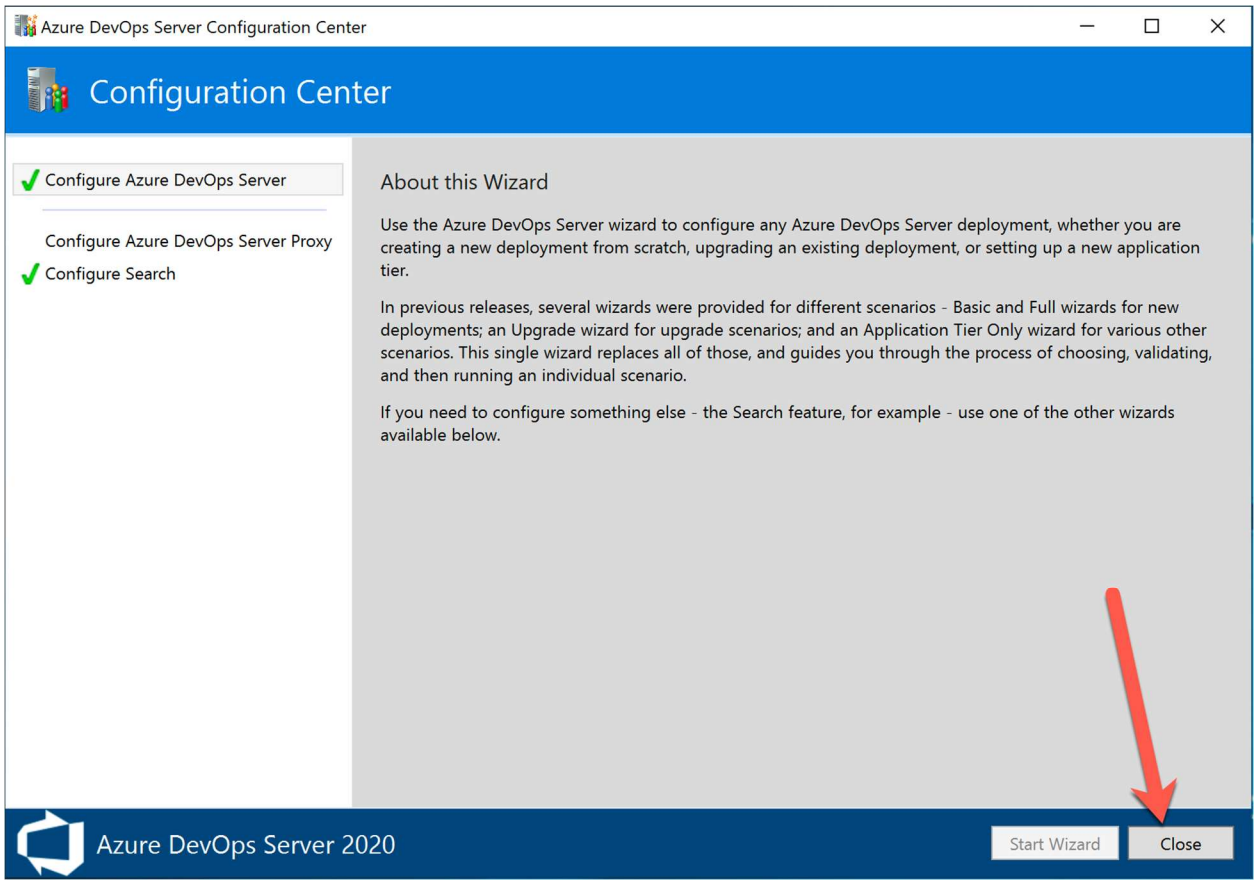


- Click **Close**

Azure DevOps Server 2020 is now configured and running.

You should now be on the main page of the configuration center.

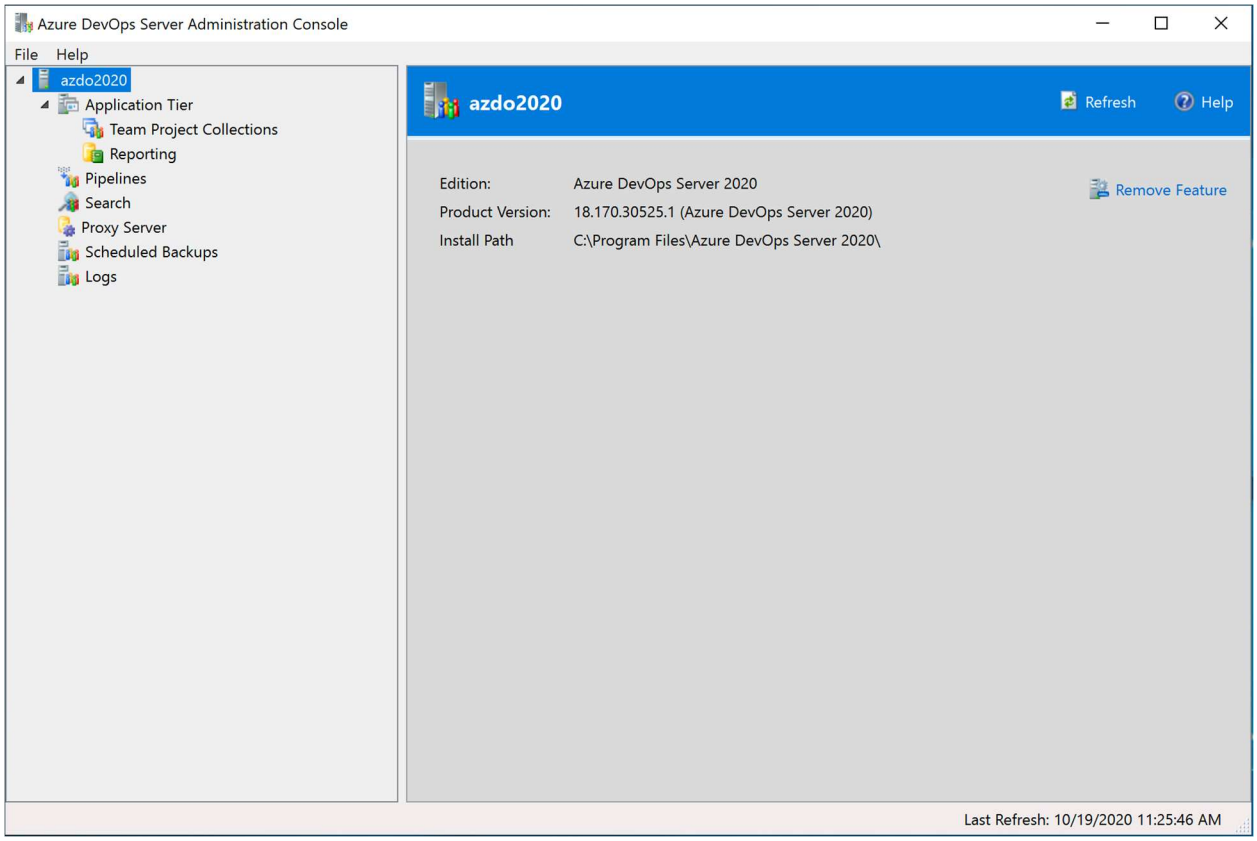
- Click the **Close** button



After you've closed the configuration center, you'll see the **Azure DevOps Server Administration Console**. This is the server-side admin tool for Azure DevOps. Remember that this application is installed on this server because at some point in the future, you'll need it.

For now, you're done.

- Close the **Azure DevOps Server Administration Console**

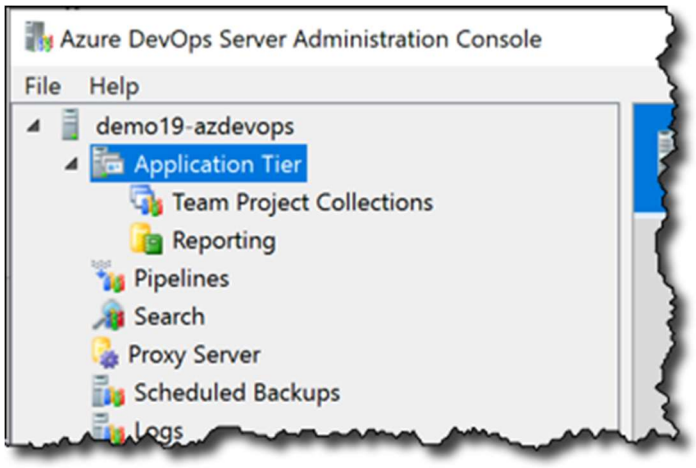


Chapter 5: Configure an SMTP Server for Azure DevOps Server

You'll definitely want to configure an SMTP server connection for Azure DevOps.

- Log in to your Azure DevOps machine as an administrator
- Go to the Start menu
- Search for **Azure DevOps Server Administrator Console**

You should see the Azure DevOps Administrator Console.

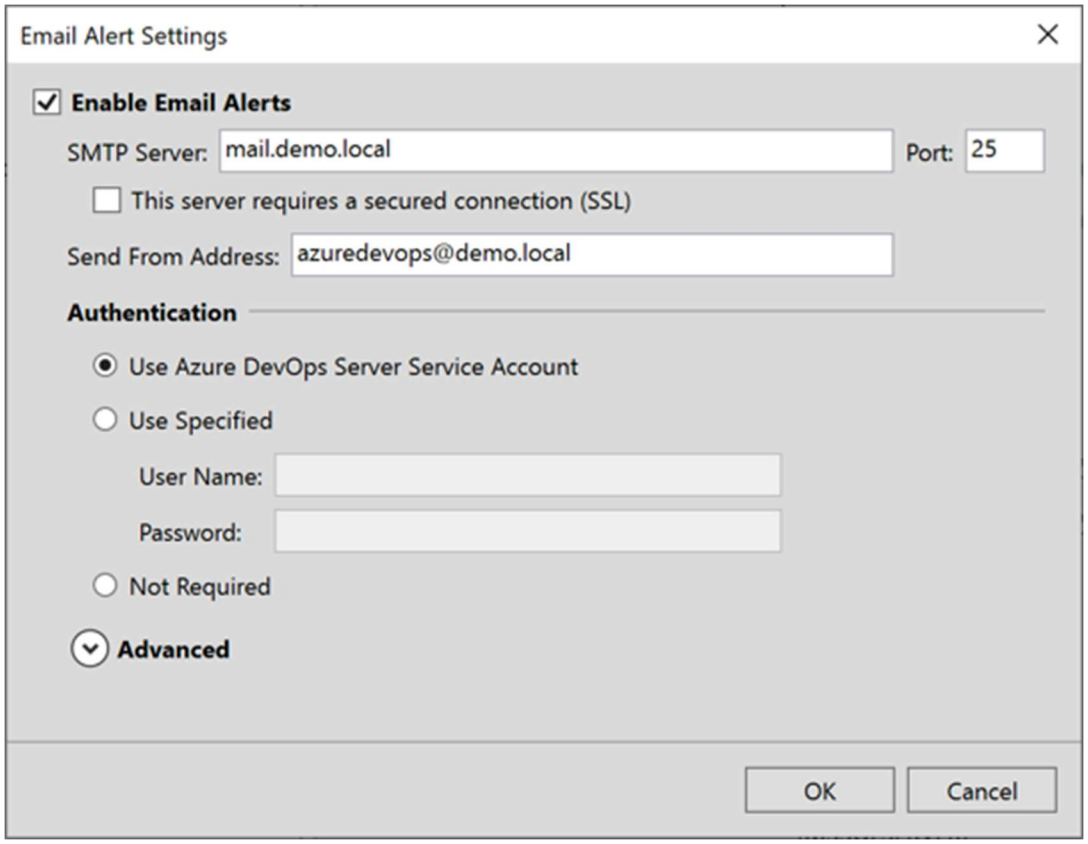


- In the left column, click on **Application Tier**

In the right panel, scroll down until you location the **Email Alert Settings.**

- Click **Alert Settings**

You should now see the **Email Alert Settings** dialog.



- Check **Enable Email Alerts**
- Set **SMTP Server** to the hostname or IP address for your SMTP server.
- Click **OK**

You should now be back on the main window of the admin console. The email alert settings should now be populated with your SMTP server.

Your Azure DevOps is ready to send emails.

Chapter 6: Install Azure DevOps Build & Release Agent on Windows Server

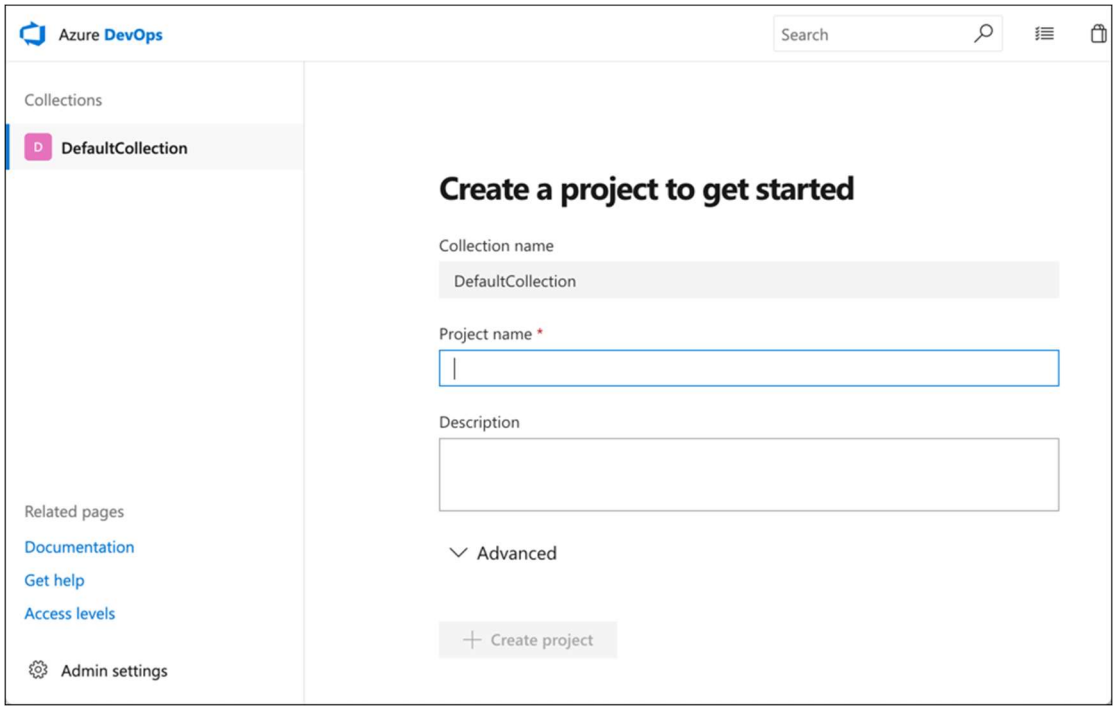
This chapter will walk you through the process of creating a build & release server on Windows. The build agent and the release agent are the same installer and process in Azure DevOps and a single installation of this agent will allow you to do "build" activities and also "release" activities.

The following steps all happen on the machine that is going to be the build server.

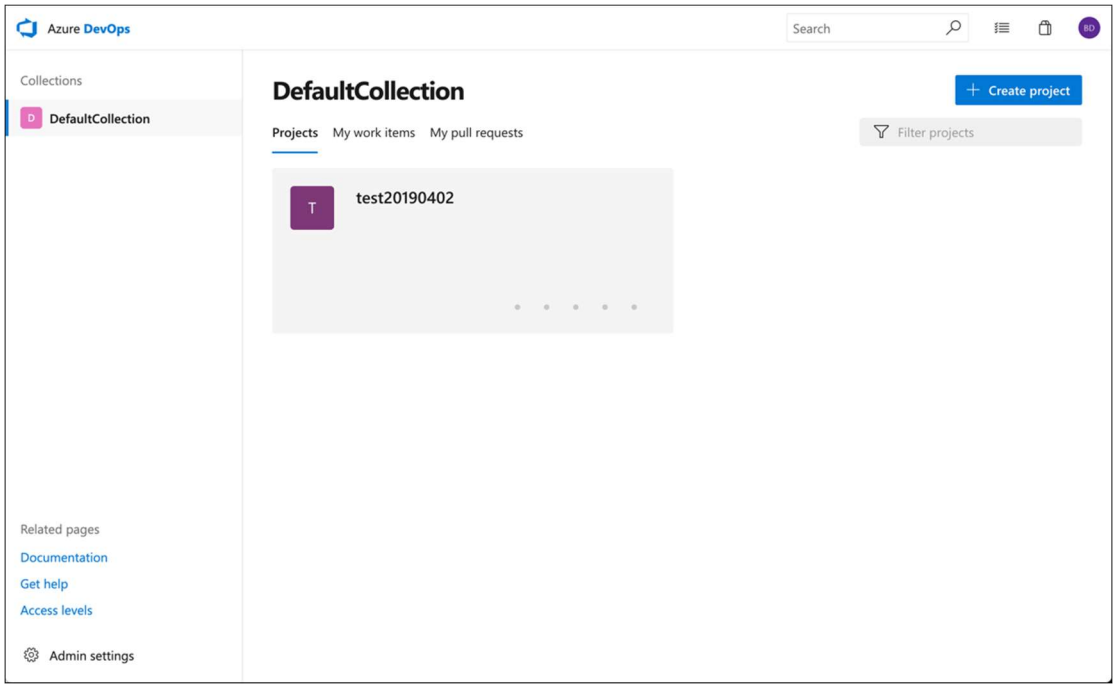
Download the Agent Installer

- Log in to the build server machine
- Open a web browser
- Navigate to your Azure DevOps Server web interface. Depending on what security settings you chose, this is probably one of the following:
  - https://servername
  - http://*servername*:8080/tfs
  - http://servername

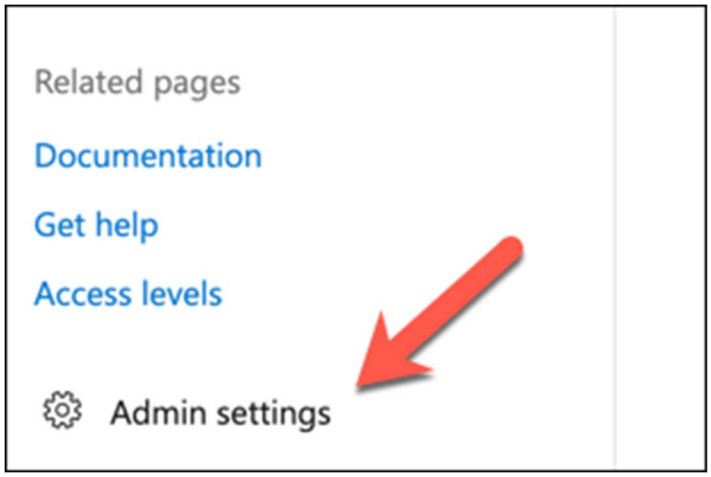
You should see a screen that looks like this…



…or perhaps a screen like this.

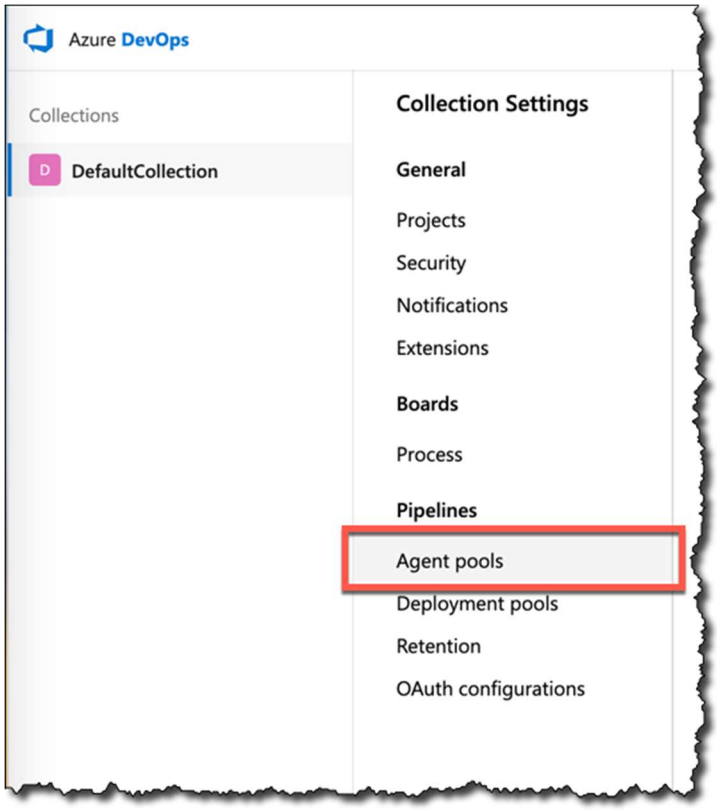


In the bottom left corner of the screen, you should see a button that says **Admin settings**.



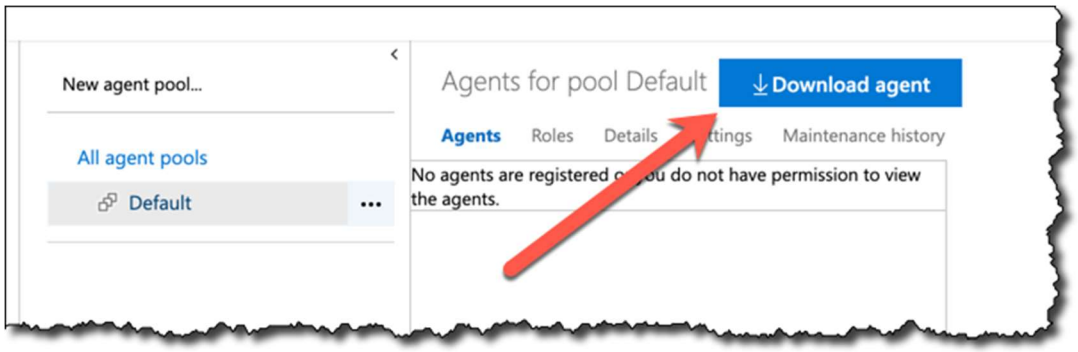
- Click the **Admin settings** button

You should now be on the **Collection Settings** page for your project collection. In the **Pipelines** section of the menu bar, locate **Agent pools**.



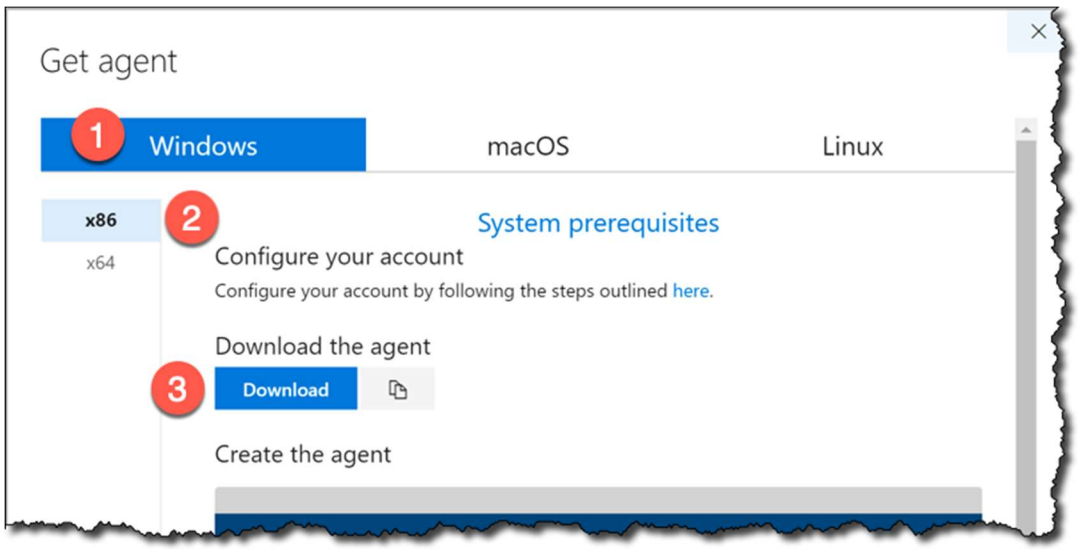
- Click the **Agent pools** link

You should see the **Agents for pool Default** screen. The **Download agent** button will navigate you to the page that lets you choose the agent for the operating system of your choice.



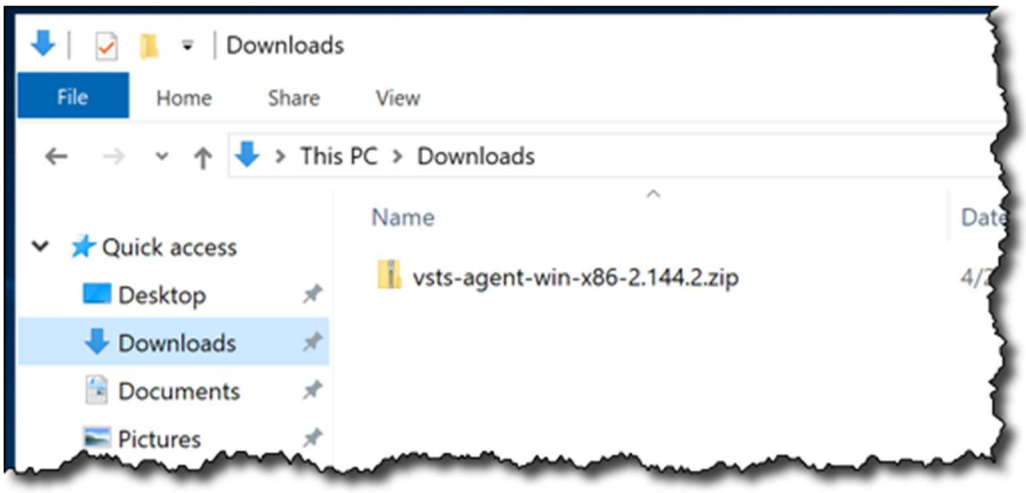
- Click the **Download agent** button

You should see a dialog like this. This guide is assuming that you're installing the agent on Windows but the installation process is very similar on all operating systems.



- From the operating system list, click the **Windows** tab
- Choose either the **x86** or **x64** version
- Click the **Download** button to start the download. (NOTE: this guide assumes you're doing the default browser behavior and downloading the agent zip to the download directory for your user profile.)

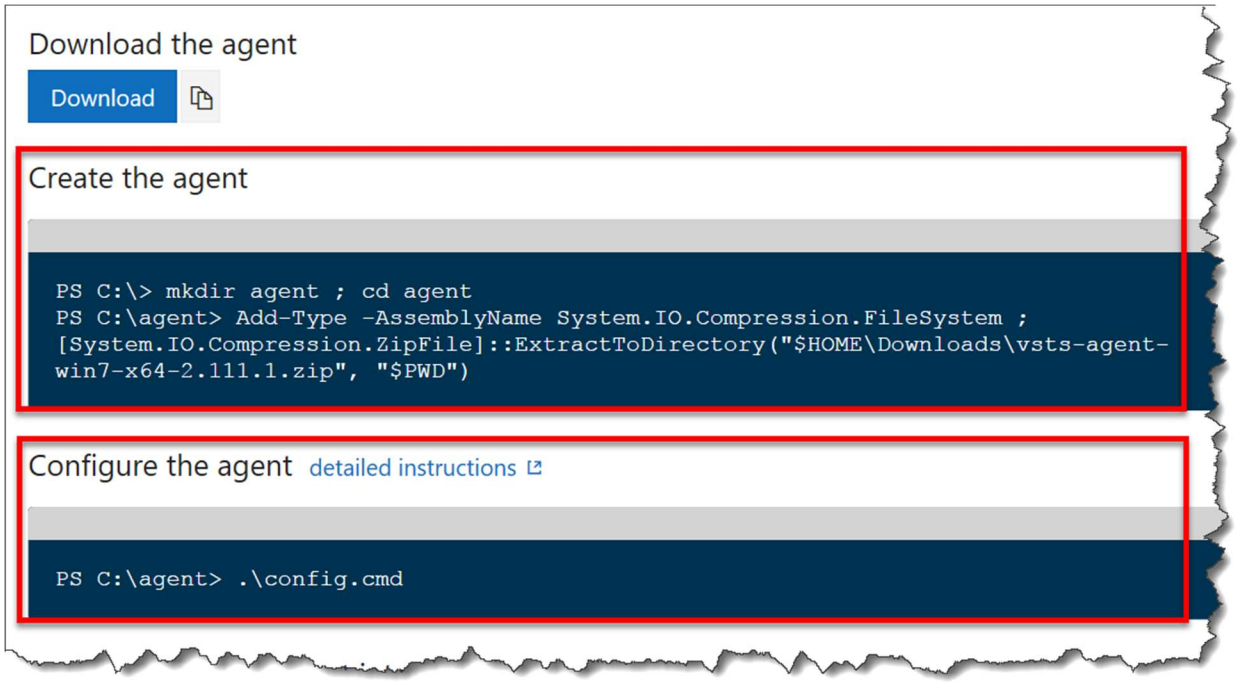
To verify that you saved this file to the expected location for this guide, open Windows Explorer (explorer.exe) and navigate to the Downloads directory. You should see a zip file that has a name that starts with "vsts-agent-". The actual filename might not be the same as the image below because the agent version may have been updated.



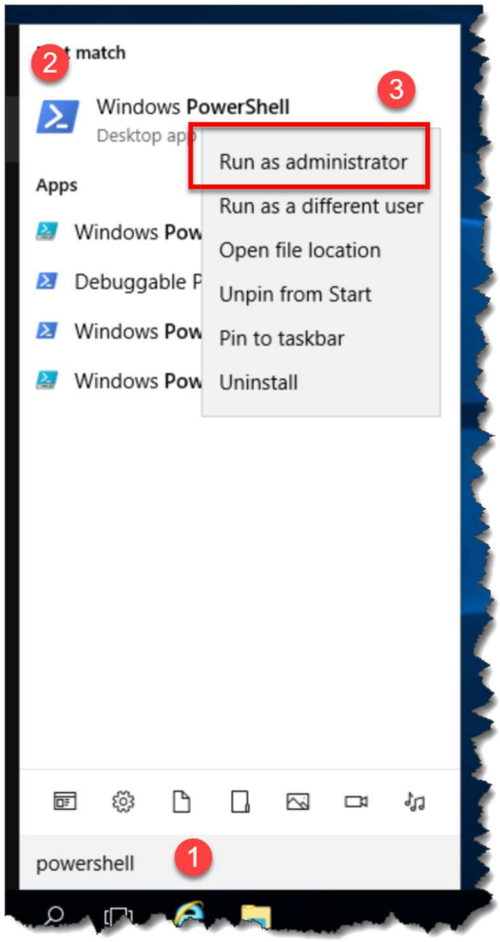
Extract the Agent

Now that you've downloaded the bits for the agent, you'll do the actual installation using PowerShell. This is actually done in two parts. Part 1: Extract the agent bits from the ZIP. Part 2: Configure the Agent. Let's do the first part.

That dialog that you used to download the agent has two different commands: "Create the agent" and "Configure the agent". In an ideal world, you'd be able to just copy and paste the commands and not have to think about anything – but this isn't an ideal world so buckle up. (Actually, it's not that bad…it's just enough friction to be a little bit annoying.)



First up, it's time to run PowerShell.

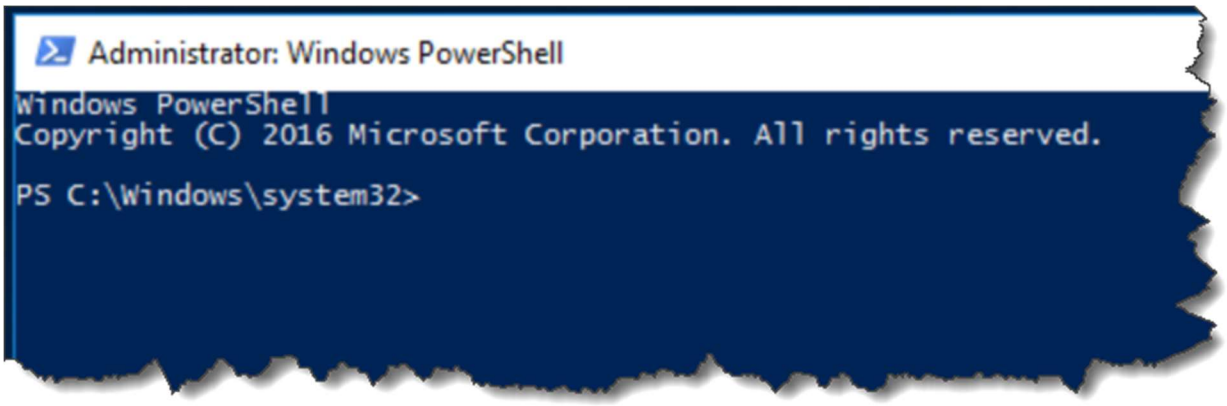


- Press the **Windows key** on your keyboard to bring up the search menu and type **PowerShell**
- From the search results, right-click **Windows PowerShell**
- From the context menu for PowerShell, choose **Run as administrator**

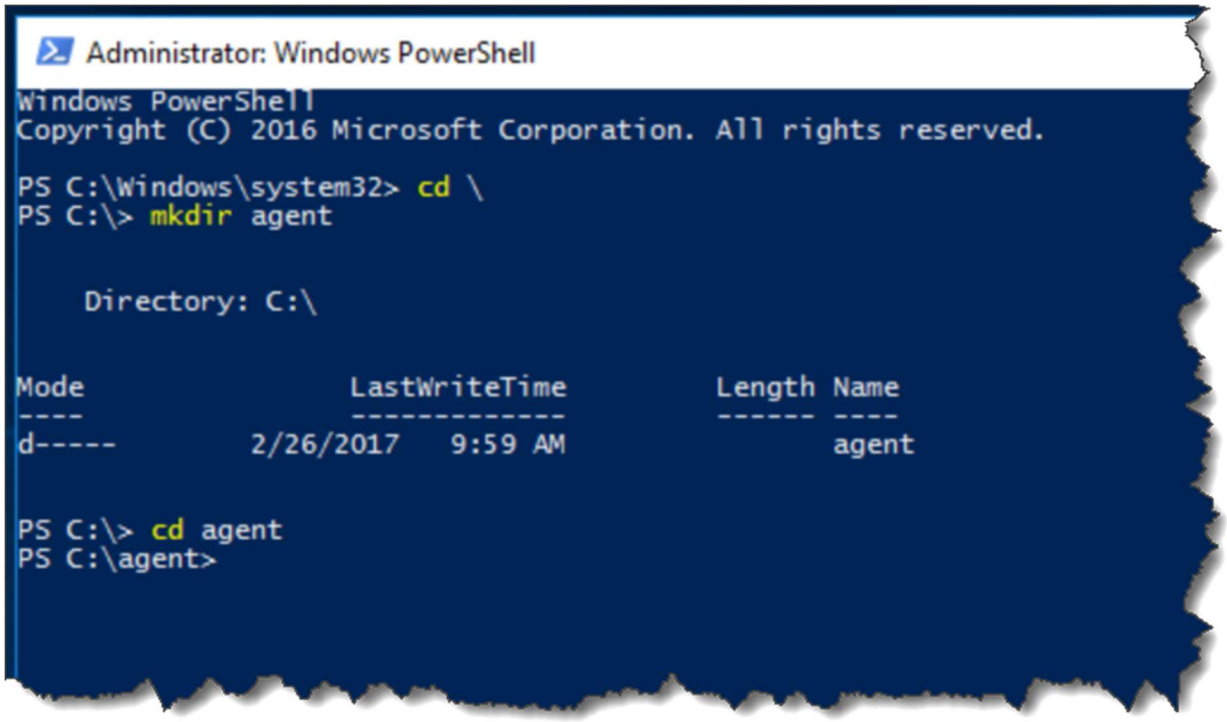
You'll see a User Account Control dialog.

- Click **Yes**

You should now see a Windows PowerShell window with the title "Administrator: Windows PowerShell".



You're now going to run a handful of commands to create the folder structure that you'll be installing the build agent in to.



- Type "cd \" and press Enter
- Type "mkdir agent" and press Enter
- Type "cd agent" and press Enter

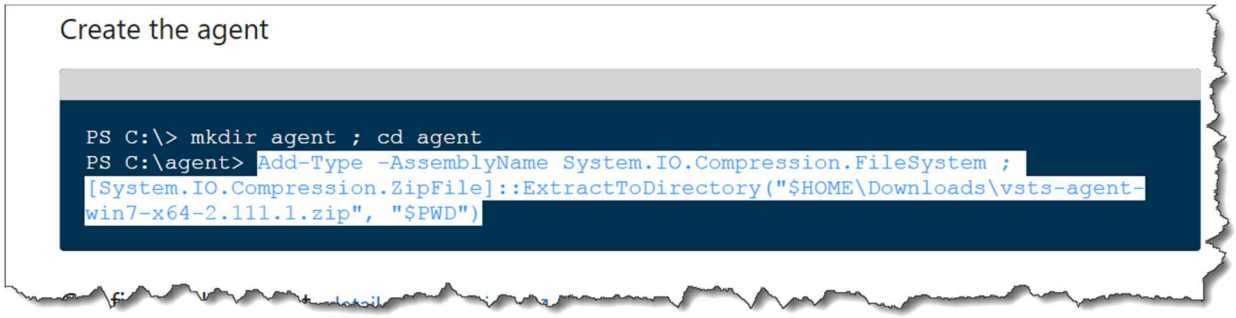
When you're done with these commands, your screen should look almost exactly the same as the image above. It's extremely important that the PowerShell screen is showing you

PS C:\agent>

on the last line because this indicates that you've correctly created a directory called Agent and entered that directory.

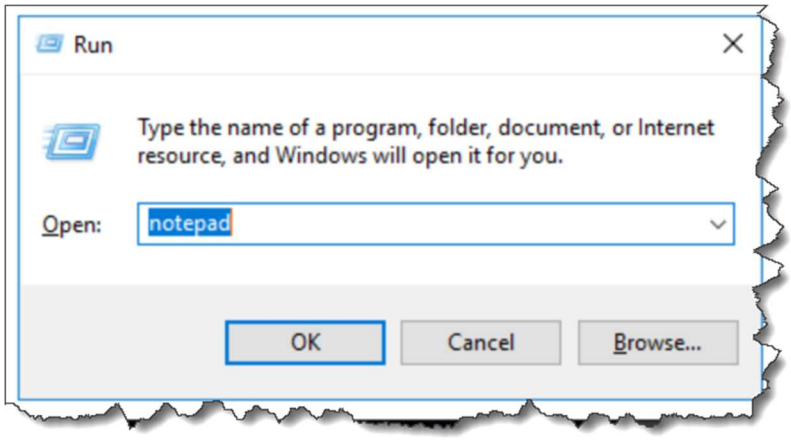
If your PowerShell window doesn't show you "PS C:\agent>", abandon all hope and reconsider all of the life choices that you've made to date. ☺

Next, you'll run the PowerShell command that will extract the zip into the agent directory. This is some fiddly typing and the exact text will change as Microsoft updates the build agent install zip filename. It's probably easiest to just copy and paste the value from the **Create the agent** section of the download dialog. You WILL NOT be copying the whole command. You'll only be copying part of the line. In the image below, it's important to notice that I am NOT selecting the portion of the line that starts with "PS C:\agent>".



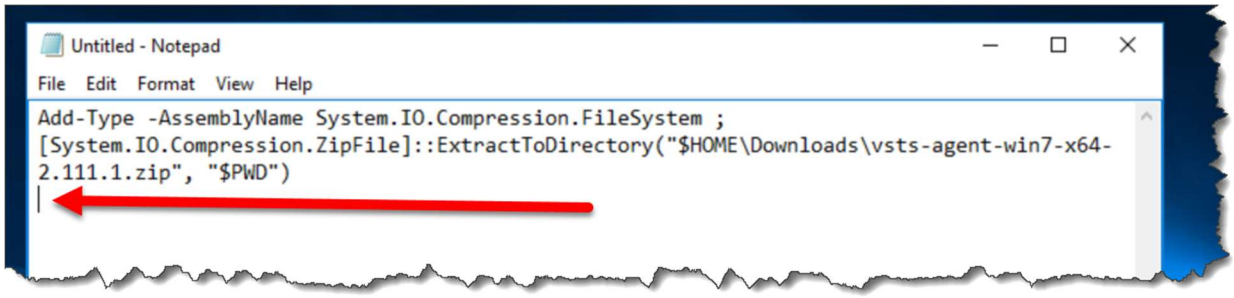
- In the web browser, copy the text of the command that starts with "Add-Type"

Because nothing is ever easy, the command that you just copied probably has some extra characters in it. Let's use Notepad to fix it.



- Type Windows-R to bring up the Run dialog
- Type notepad
- Click OK
- Paste the copied command into Notepad

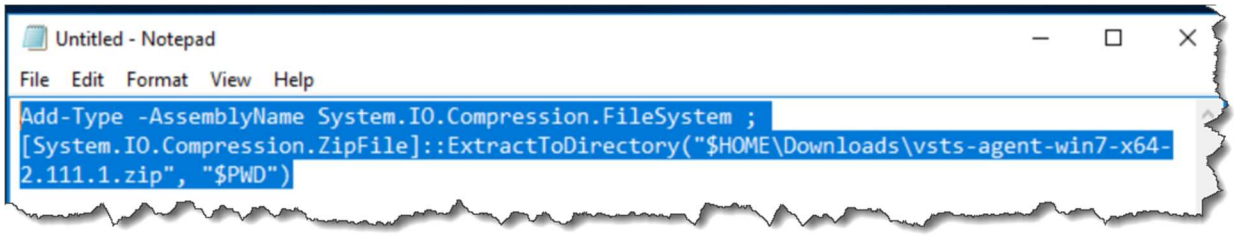
You should now see the command in Notepad. If everything is on one long line, go to the Format menu and choose Word Wrap. You might notice in the image below that my cursor is sitting on an empty line by itself. It's this extra line that's causing us to do this Notepad step.



- Delete the extra empty blank line at the end
- If there are any whitespace characters before "Add-Type", delete those, too.

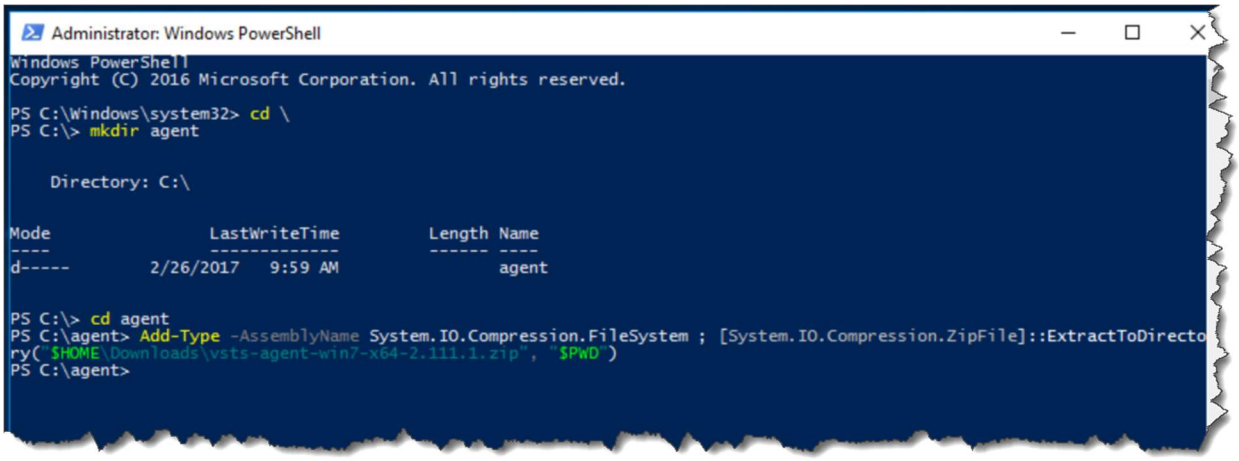
There's a chance that the "ExtractToDirectory" doesn't have the name of the zip file that you downloaded. Make sure that the ExtractToDirectory command has the name of the zip file rather than just "$HOME\Downloads\" for the first argument. If it's not there, you'll need to add it in to your command in notepad.

You should have a clean command that you can just paste into PowerShell and run.



- Select the command
- Press CTRL-C to copy the selected command to the clipboard

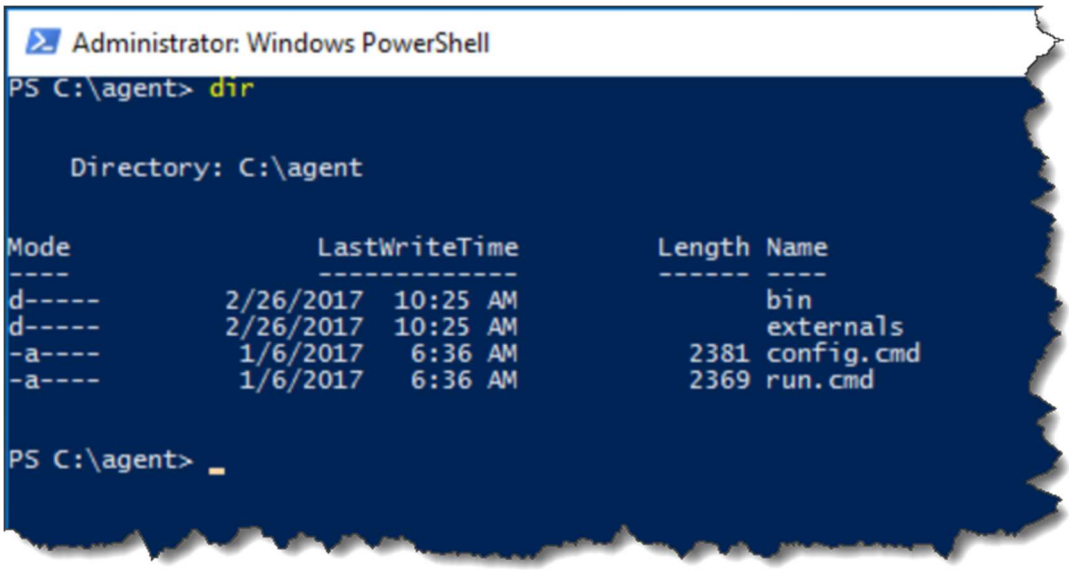
Now you'll run the command in PowerShell.



- In the PowerShell window, type **CTRL-V** to paste the command
- Press **Enter** to run the command

When the command is done, you should not see any errors and the prompt should say "PS C:\agent>". (NOTE: this might take a few minutes to run.)

Let's verify that this extracted as expected.



- (Optional) To clear the screen, type "cls" and press Enter
- Type "dir" and press Enter

The screen should look something like the image above.

Configure the Agent

Now that the agent bits are deployed to disk, you're ready to start configuring it. This guide assumes that you're planning to run this agent in a Windows domain and that the TFS machine is in the same domain as the agent. I'm also assuming that you intend to run this agent as a service rather than as an interactive process.

Recommendation: The agent can be configured to run as NT AUTHORITY\NETWORK SERVICE but I think that this makes permissions management confusing when you're creating and running builds. I strongly recommend that you run the agent as a service using a known service account that is based on an Active Directory user rather than one of the build-in service accounts like NETWORK SERVICE. This guide will assume that you're following this recommendation.

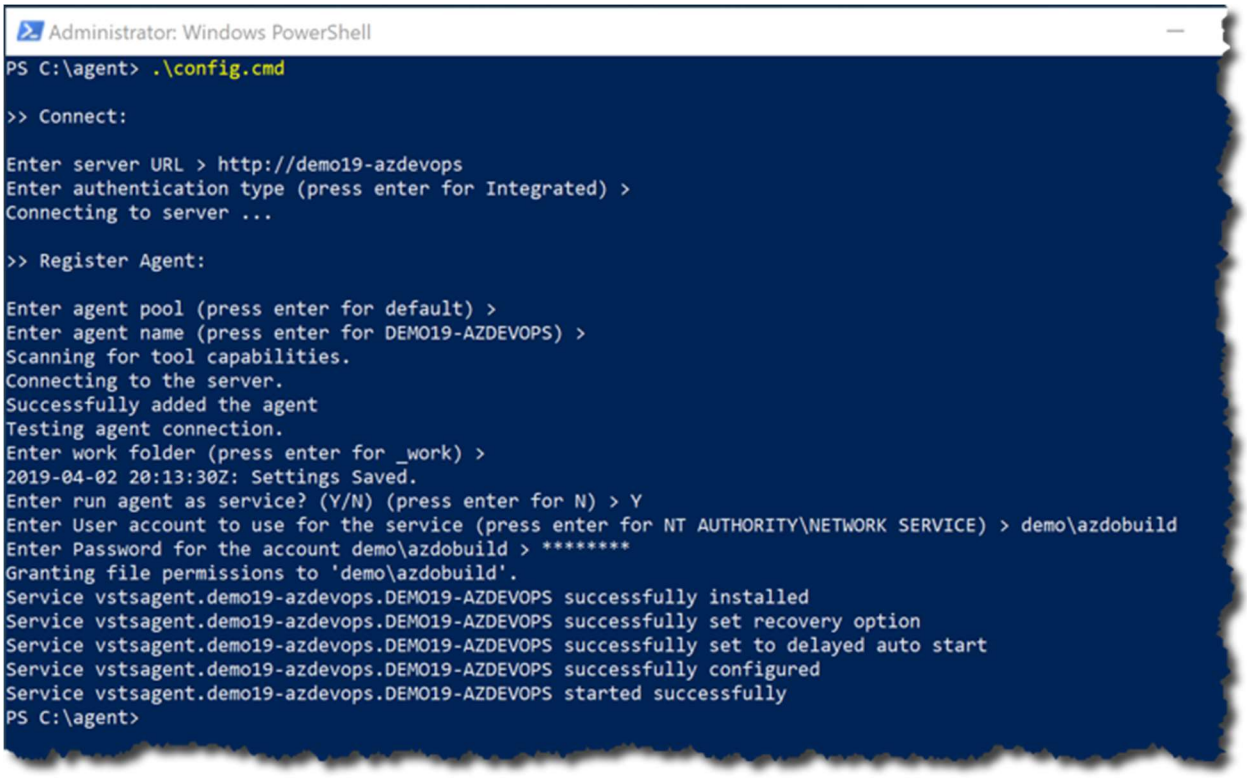
You're going to be prompted for a handful of values during the configuration process:

- **Azure DevOps Server URL:** This is the same URL that you used to access the Azure DevOps Server web interface. By default this will be something like http://*servername* or http://*servername*:8080/tfs or if you're using SSL/HTTPS, it'll be something like https://*servername*.
- **Authentication Type:** By default, authentication is based on the service account's Windows logon. This mode is called Interactive. In order support more complex scenarios and multiple platforms, there are also several other options. This guide will show you how to do Interactive mode.
- **User name & password for the agent service:** These are the credentials for the service. In my case, I've created an Active Directory user named "azdobuild". The fully qualified username for this user is "DEMO\azdobuild".

When you've got these values, you're ready to run the config process.

- In the PowerShell window, type "**.\config.cmd**" and press Enter

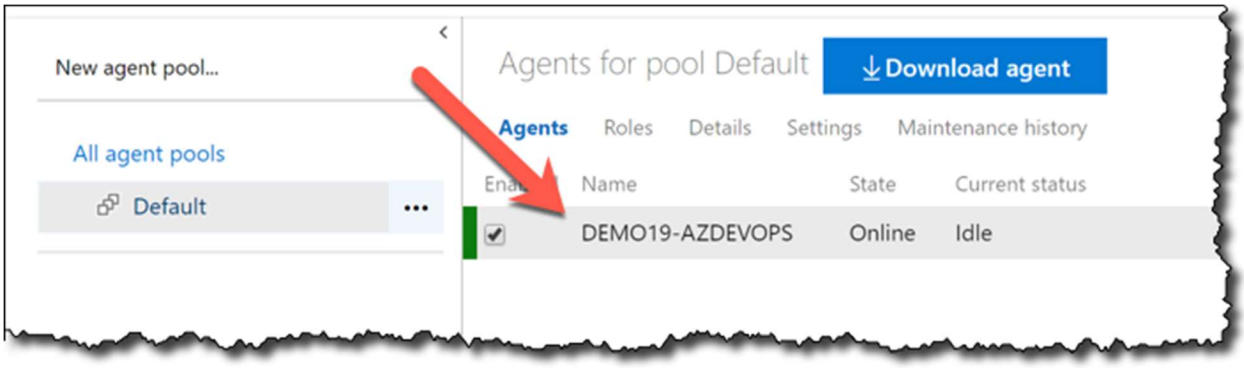
When prompted, enter the following values. NOTE: You might be tempted to accept the defaults for each one but you're **NOT** going to do that because you'll mess up the "run as a service" part and you'll be deeply unhappy.



- "Enter server URL":
  Type the **URL for your TFS instance** and click **Enter**
- "Enter authentication type (press enter for Integrated)":
  Press **Enter**
- "Enter agent pool (press enter for default)":
  Press **Enter**
- "Enter agent name (press enter for *[local server name]*)":
  Press **Enter**
- "Enter run agent as service? (Y/N)":
  Type '**Y**' and press **Enter**
- "Enter User account to use for the service":
  Type the **fully qualified name of the service account** (example: demo\azdobuild) and press **Enter**
- Enter Password for the account *[service account]*":
  Enter the **password for the service account** and press **Enter**

When the config process has completed, you should see a message that says something like "Service vstsagent.demo19-azdevops.DEMO19-AZDEVOPS started successfully".

If you open the browser and go back to the Agent Pools tab for TFS, you should now see your new build agent in the list of Agents.



You've successfully configured a build agent.

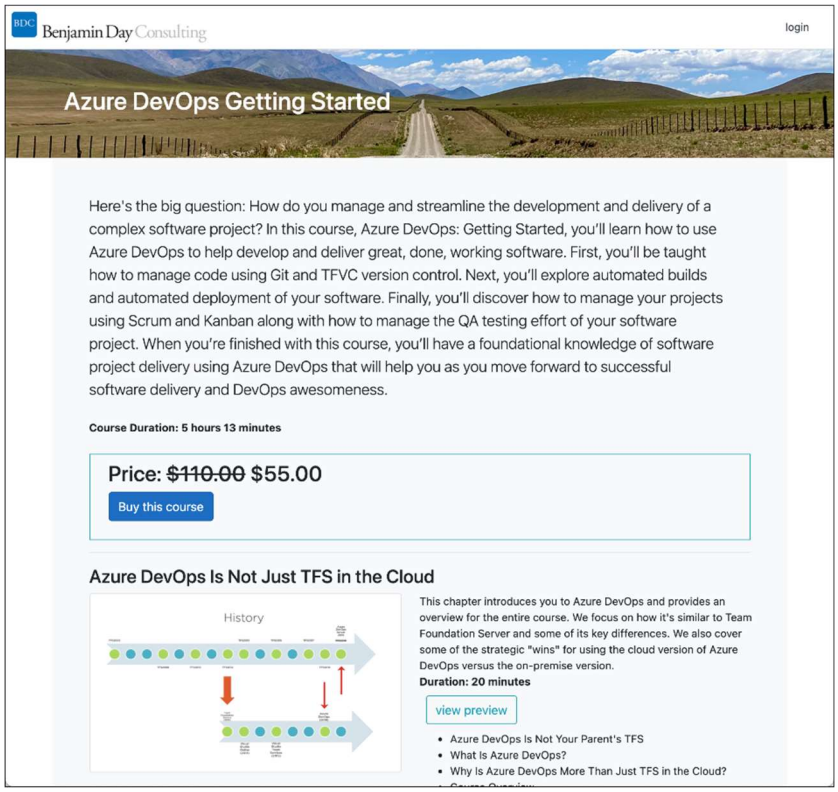
Chapter 7: Training, Consulting, & Software Development

Need help with your Azure DevOps installation? Trying to figure out how to upgrade your old install of Team Foundation Server? Want to move Team Foundation Server or Azure DevOps Server to the cloud? Looking for help with your software development process or getting going with automated builds and releases?

Drop us a line at info@benday.com!

Looking for training for how to use Azure DevOps? Check out our online video course.

https://courses.benday.com



We also have courses at Pluralsight as well as options for virtual (remote) training and in-person training.

Topics include Azure DevOps, Scrum, Test-Driven Development & Unit Testing, ASP.NET Core, Software Architecture and more! We also do consulting and custom software development.

Drop us a line at info@benday.com or visit https://www.benday.com for more information.