IIS Server Setup

2023

Setting up IIS – Internet Information Server for TimeTrak

How to set up IIS to meet TimeTrak Requirements

Before you begin:

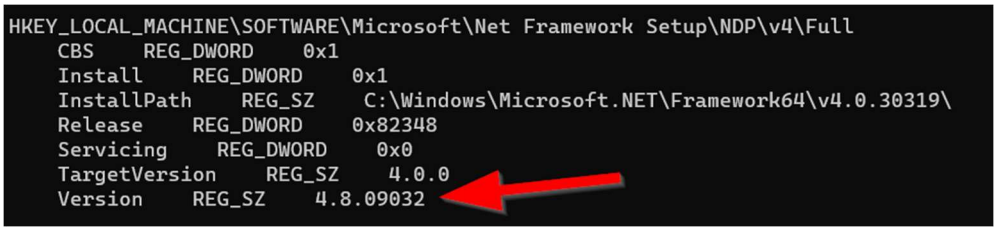
- Make sure IIS is closed
- .NET 4.8. is required on both the webserver and any other server or PC that will be used to run TimeTrak Administrator Console.

You can download it from Microsoft.
[Download .NET Framework 4.8 Web Installer (microsoft.com)]

To check what version of .NET is already installed:

How to check with CMD

reg query "HKLM\SOFTWARE\Microsoft\Net Framework Setup\NDP\v4" /s



How to check with PowerShell

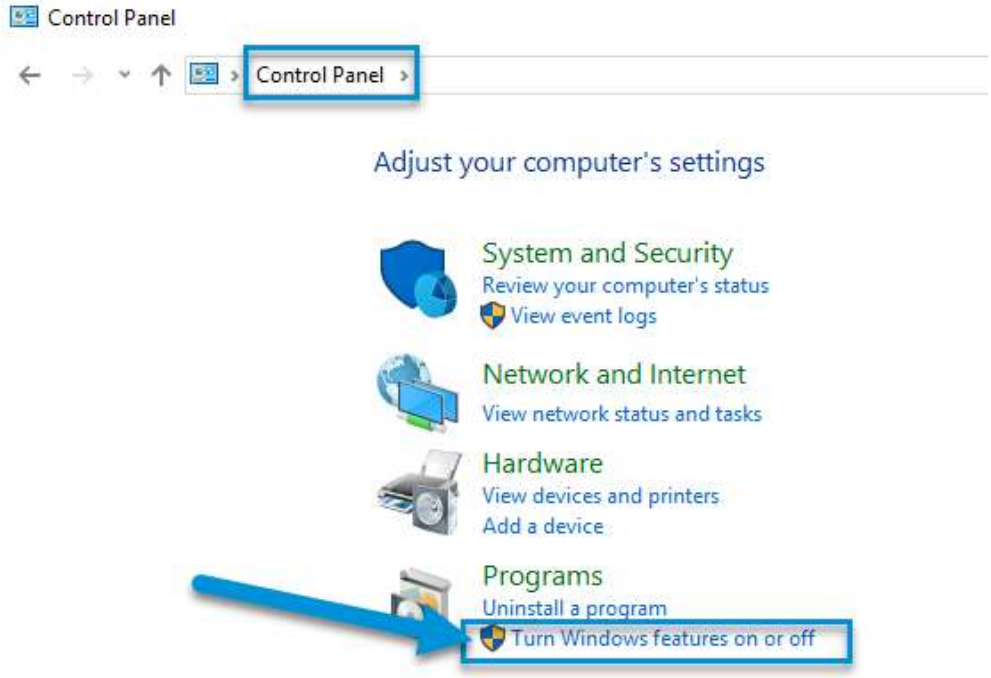
```
$release = Get-ItemPropertyValue -LiteralPath 'HKLM:SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full' -
Name Release
switch ($release) {
    { $_ -ge 533320 } { $version = '4.8.1 or later'; break }
    { $_ -ge 528040 } { $version = '4.8'; break }
    { $_ -ge 461808 } { $version = '4.7.2'; break }
    { $_ -ge 461308 } { $version = '4.7.1'; break }
    { $_ -ge 460798 } { $version = '4.7'; break }
    { $_ -ge 394802 } { $version = '4.6.2'; break }
    { $_ -ge 394254 } { $version = '4.6.1'; break }
    { $_ -ge 393295 } { $version = '4.6'; break }
    { $_ -ge 379893 } { $version = '4.5.2'; break }
    { $_ -ge 378675 } { $version = '4.5.1'; break }
    { $_ -ge 378389 } { $version = '4.5'; break }
    default { $version = $null; break }
}

if ($version) {
    cls
    Write-Host -Object ".NET Framework Version: $version"
    pause
} else {
    cls
    Write-Host -Object '.NET Framework Version 4.5 or later is not detected.'
    pause
}
```

TimeTrak requires the following to be configured.

.NET Extensibility 4.X
Application Development
Application Initialization
ASP .NET 4.X
Default Document
Directory Browsing
Enable Static Content
HTTP Errors
IIS Management Console
ISAPI Extensions
ISAPI Filters
Request Filtering
Static Content Compression
URL Rewrite

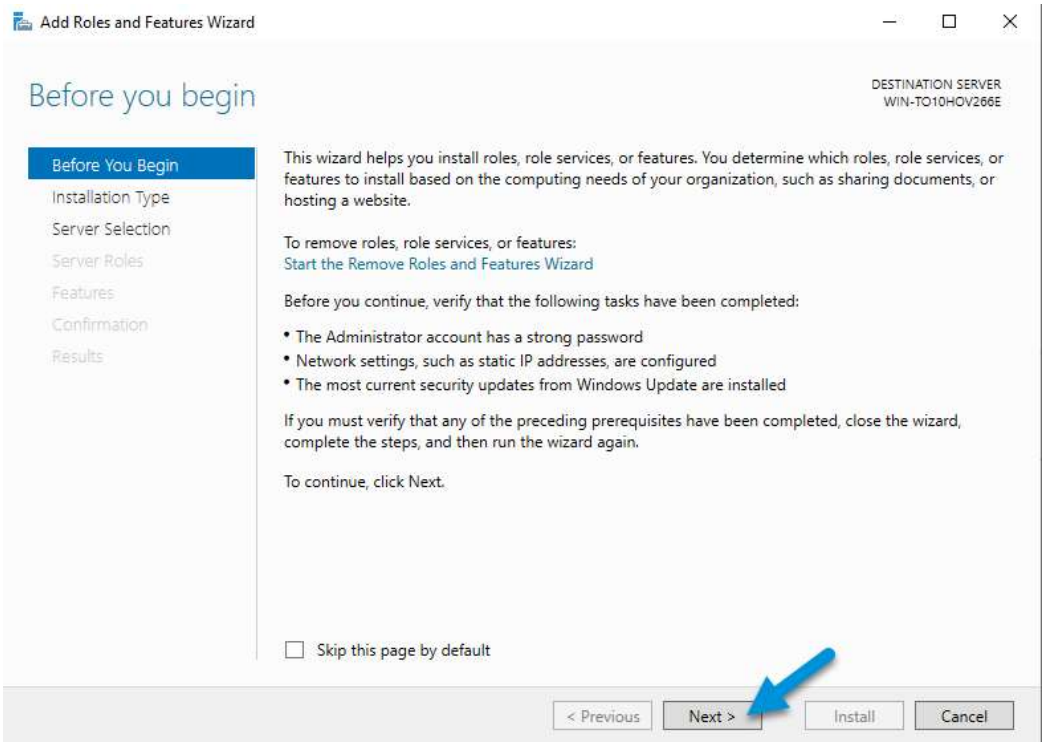
You can step through the guide below to install IIS and modules manually or on the last page there is a script to install IIS and all required modules.

Go into Control Panel and select Turn Windows features on or off as per below.



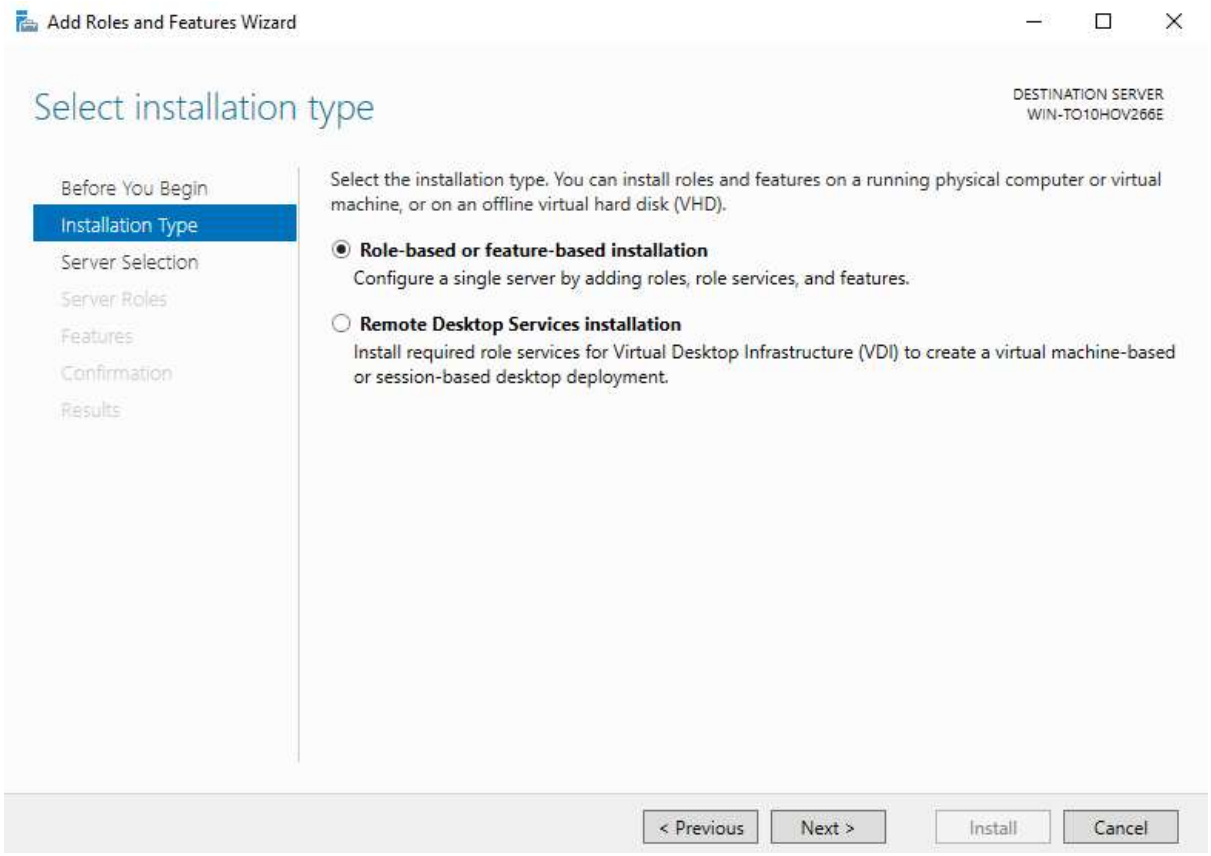
This will launch Server Manager

You will see an “Add Roles and Features Wizard” as per below, Click next.

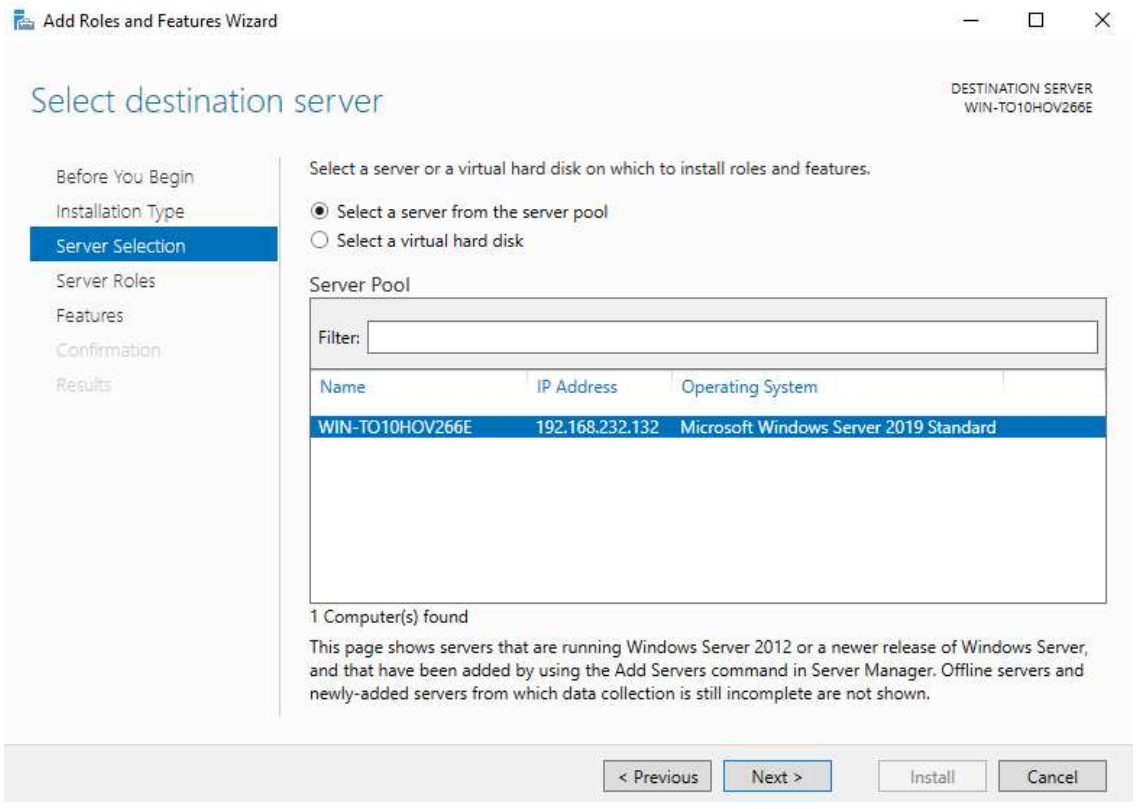


Choose the installation type based on your environment.

In this example, we choose “Role-based or feature-based installation” and then click next.



Select the destination server to install your ISS



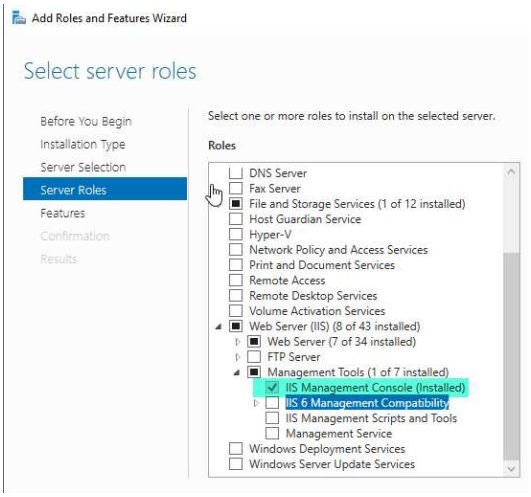
On the next screen, you need to choose the roles that you need to install.

These are the roles that you need to set up for TimeTrak:

Internet Information Services:

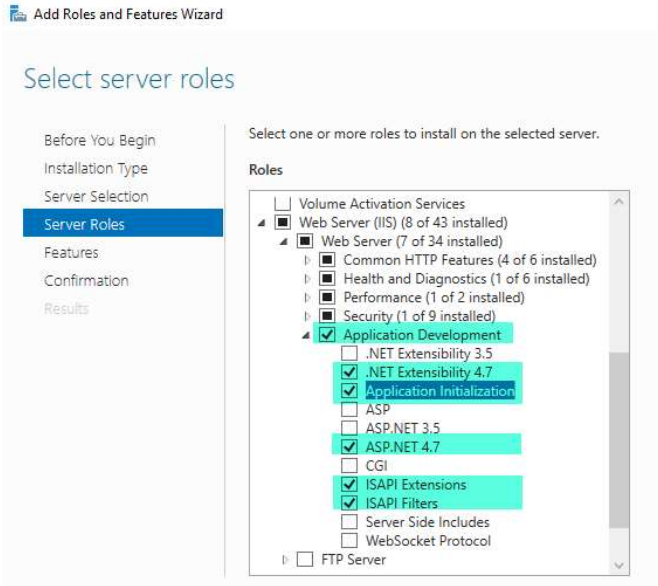
Web Server (IIS) > Management Tools

- IIS Management Console



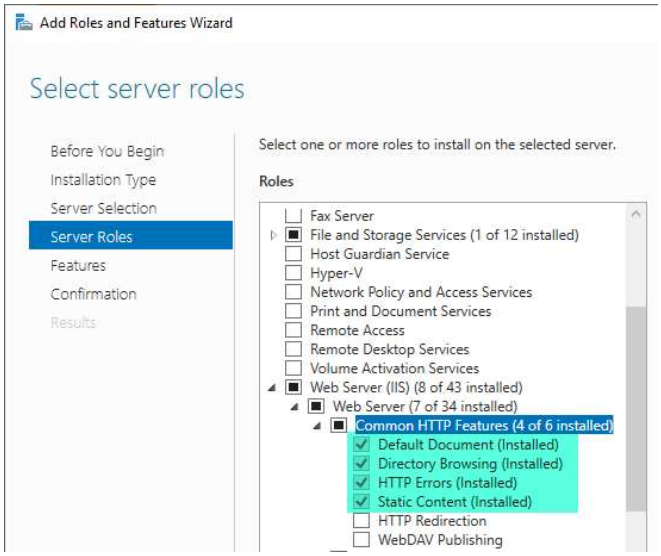
Web Server (IIS) > Web Server > Application Development

- Enable Application Development
- Enable .NET Extensibility 4.X
- Enable Application Initialization
- Enable ASP .NET 4.X
- Enable ISAPI Extensions
- Enable ISAPI Filters



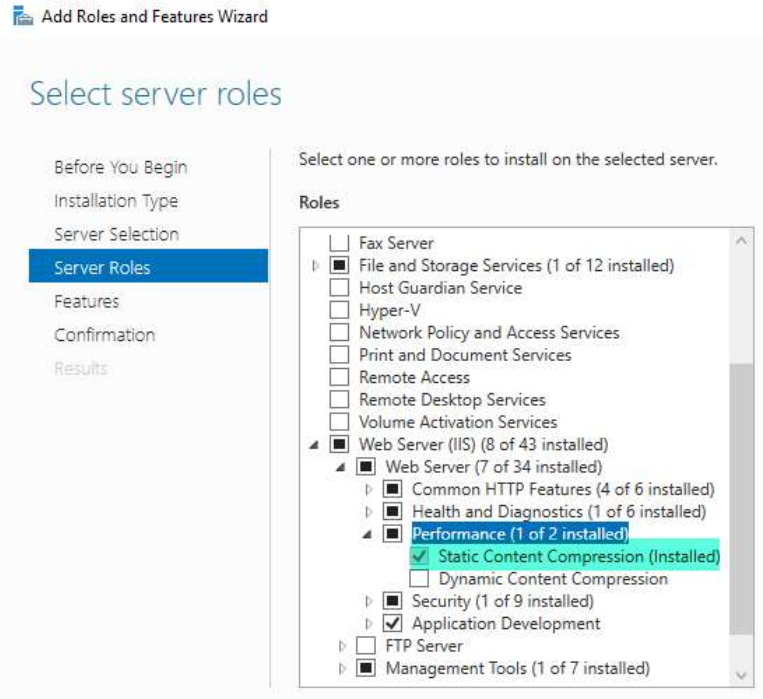
Web Server (IIS) > Web Server > Common HTTP Features

- Enable Default Document
- Enable Directory Browsing
- Enable HTTP Errors
- Enable Static Content



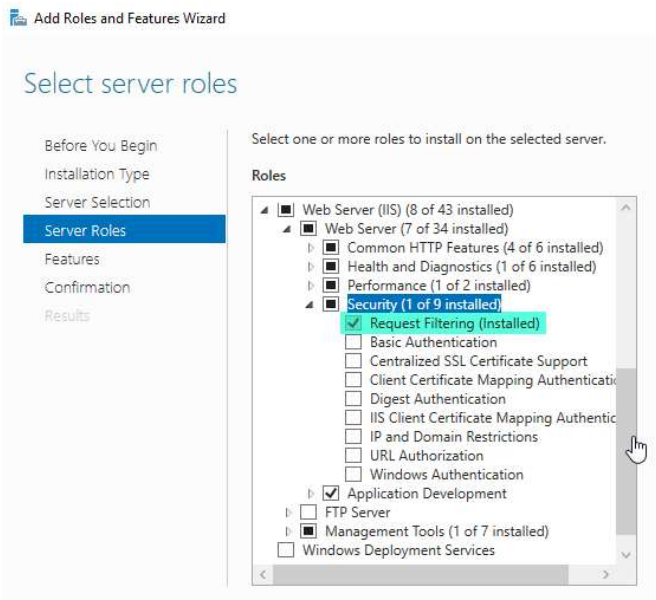
Web Server (IIS) > Web Server > Performance

- Enable Static Content Compression

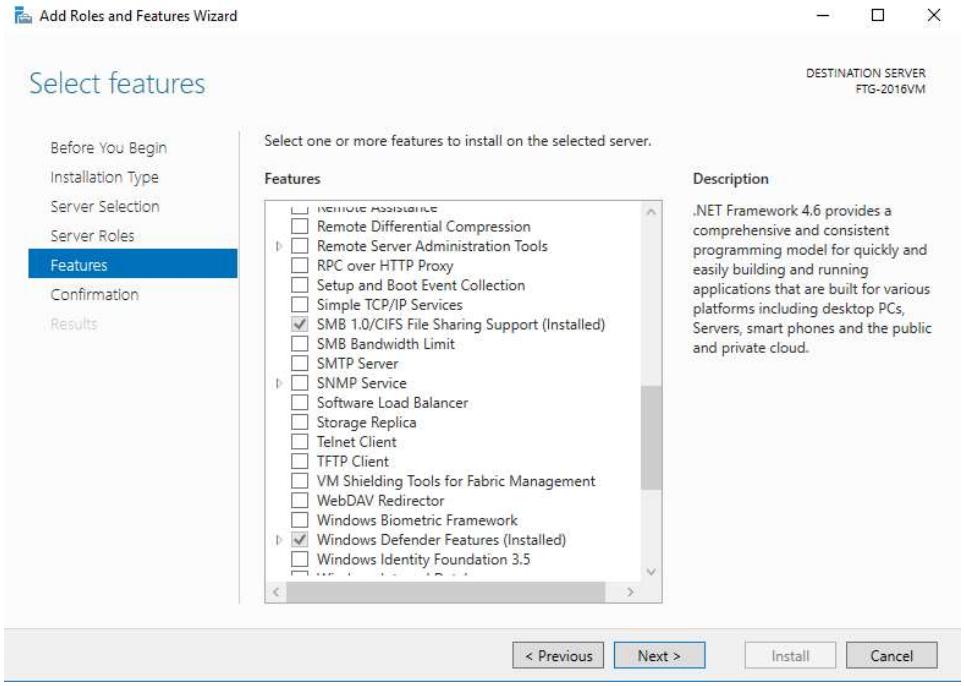


Web Server (IIS) > Web Server > Security

- Enable Request Filtering



Once everything is done, click next and you will see other features that you need to install on your server. It's not mandatory for TimeTrak so we can continue to the next screen.

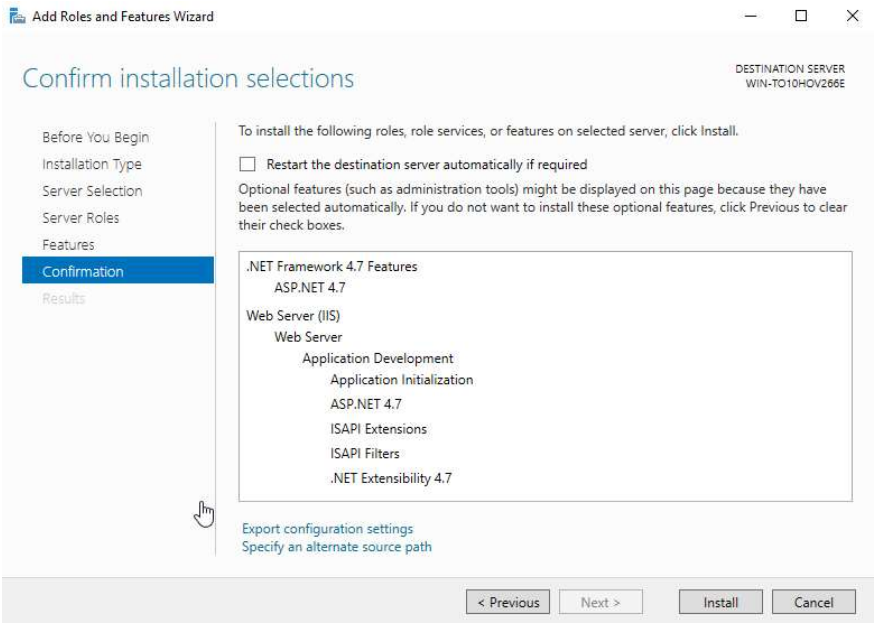


You can review any roles and features that you want to install based on the roles that you have ticked on.

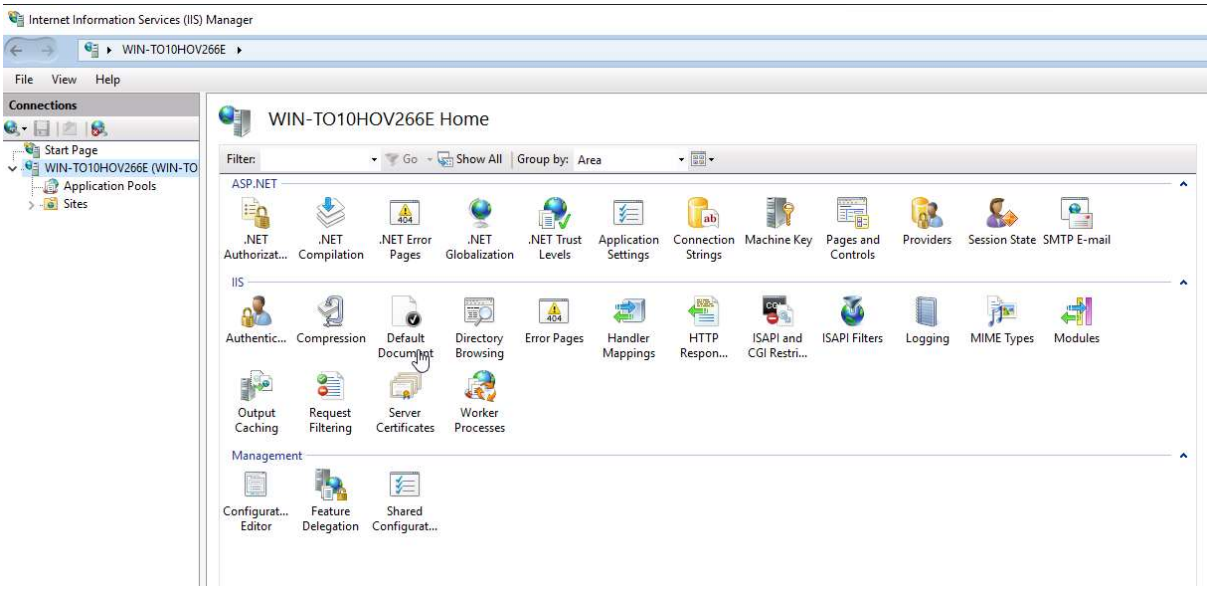
Note: DO NOT tick on 'restart the destination server automatically if required' if you don't want the server to restart without any warning. With this left unticked - if a restart is required, you will be prompted to restart.

After everything is according to what you need, click install.

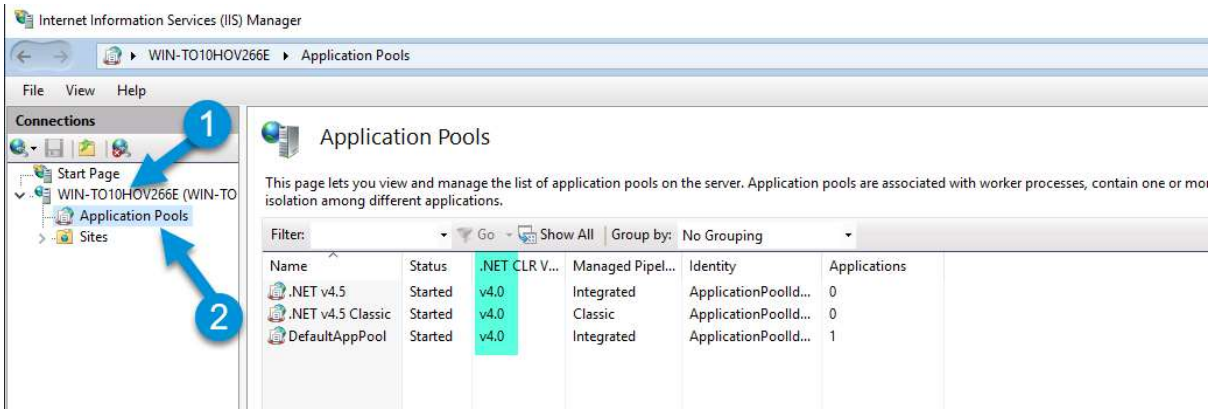
When everything is done, you will have IIS on your server.



Open IIS and you will have IIS management window.



Go to your server name > application pools, and you can see all the application roles that you have set up. The most important to see here is you have .Net v4.x installed in here.

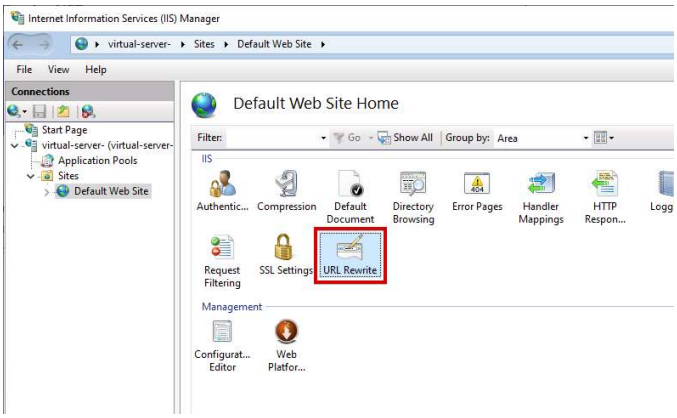


You will need to download and install **URL Rewrite Module 2.1** if you do not see it in the module list.

[x64 Installer URL Rewrite]
[x86 Installer URL Rewrite]

Download, run, and follow the prompts to install the module.

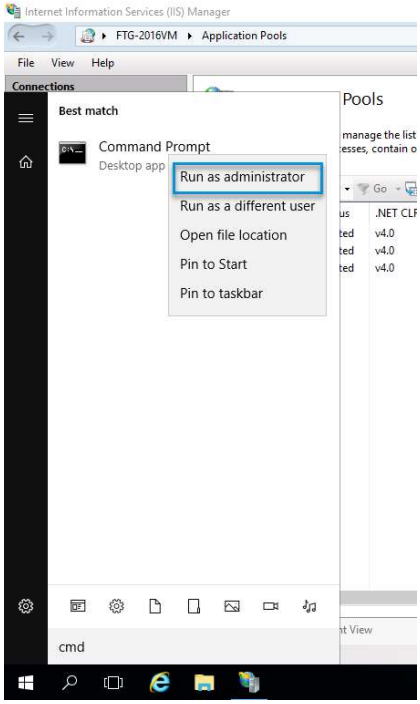
Lastly, restart your IIS Manager, and you should now have the module available.



NOTE: If IIS was installed after the .Net Framework then after installation of IIS, you will need to re-register the .NET Framework so that the recently installed IIS can access the .NET version.

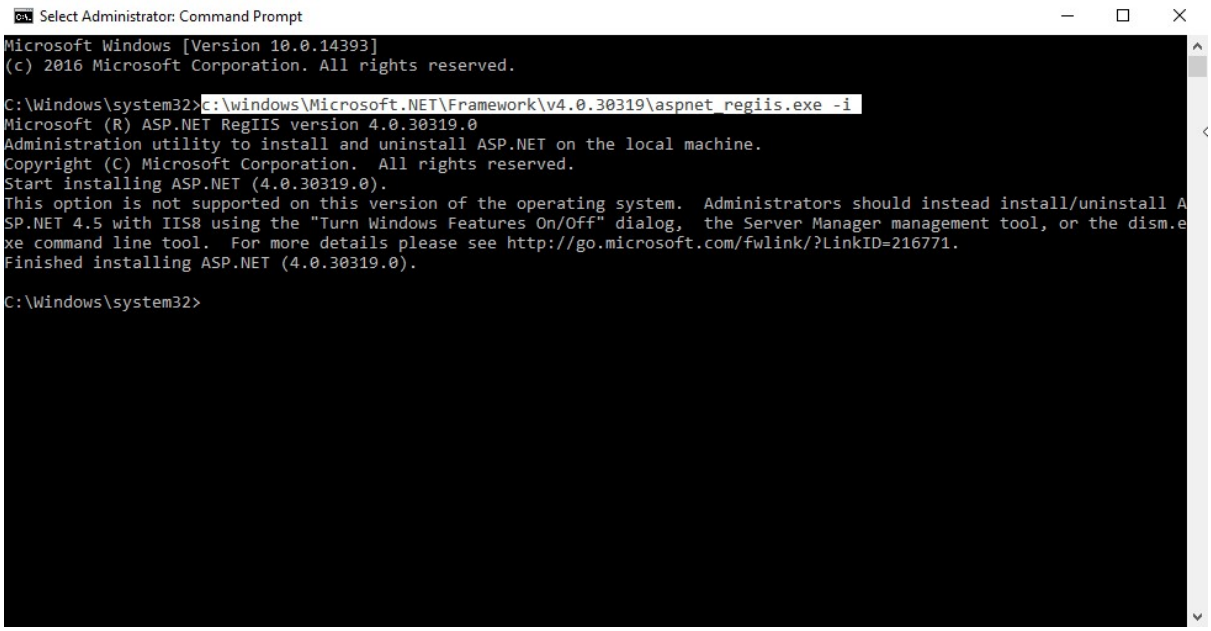
Click on Windows Start menu

Search Command Prompt then right click and Run as Administrator



Type or copy-paste the following path

c:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -i



Script to install IIS and all Modules on Windows Server (Run PowerShell as an Administrator)

```
$moduleName = "UrlRewrite2"
$moduleVersion = "2.1"
$moduleArchitecture = Get-WmiObject win32_processor | Select-Object AddressWidth | Select-Object -
ExpandProperty AddressWidth

$tempDir = "C:\Temp"
if (!(Test-Path -Path $tempDir -PathType Container)) {
    Write-Output "Creating directory $tempDir ..."
    New-Item -Path $tempDir -ItemType Directory | Out-Null
}

if ($moduleArchitecture -eq 64) {
    $moduleDownloadUrl = "https://download.microsoft.com/download/1/2/8/128E2E22-C1B9-44A4-BE2A-
5859ED1D4592/rewrite_amd64_en-US.msi"
    $moduleInstallerPath = "$tempDir\UrlRewrite_amd64.msi"
} elseif ($moduleArchitecture -eq 32) {
    $moduleDownloadUrl = "https://download.microsoft.com/download/D/8/1/D81E5DD6-1ABB-46B0-9B4B-
21894E18B77F/rewrite_x86_en-US.msi"
    $moduleInstallerPath = "$tempDir\UrlRewrite_x86.msi"
} else {
    Write-Output "Unknown server architecture."
    exit
}

if (!(Get-ChildItem "HKLM:\SOFTWARE\Microsoft\IIS Extensions\UrlRewrite" -ErrorAction SilentlyContinue)) {
    Write-Output "$moduleName is not installed. Downloading and installing $moduleName version
$moduleVersion..."
    Write-Output "Downloading $moduleName version $moduleVersion for $moduleArchitecture-bit
architecture..."
    Invoke-WebRequest -Uri $moduleDownloadUrl -OutFile $moduleInstallerPath
    Write-Output "Download of $moduleName version $moduleVersion for $moduleArchitecture-bit architecture
is complete."

    Write-Output "Installing $moduleName version $moduleVersion..."
    Start-Process -FilePath msiexec.exe -ArgumentList "/i $moduleInstallerPath /quiet" -Wait
    Write-Output "Installation of $moduleName version $moduleVersion is complete."
} else {
    Write-Output "$moduleName is already installed."

}


$logfile = "C:\temp\TimeTrakIISSetup.txt"

# Function to log messages to the specified file
function LogMessage {
    param(
        [Parameter(Mandatory=$true)]
        [string]$Message
    )

    Add-Content -Path $logfile -Value "$Message"
}

# Array of features to install
$features = @(
    "Web-Server",
    "Web-Mgmt-Tools",
    "Web-Security",
    "Web-App-Dev",
    "Web-Net-Ext45",
    "Web-AppInit",
    "Web-Asp-Net45",
    "Web-ISAPI-Ext",
    "Web-ISAPI-Filter",
    "Web-Default-Doc",
    "Web-Dir-Browsing",
    "Web-Http-Errors",
    "Web-Static-Content",
    "Web-Stat-Compression",
    "Web-Filtering"
)

foreach ($feature in $features) {
    $installed = Get-WindowsFeature -Name $feature | Where-Object {$_.Installed -eq $true}
    if (!$installed) {
        # Feature is not installed, so install it
        Write-Output "Installing $feature"
        LogMessage "Installing $feature"
        Install-WindowsFeature -ComputerName localhost -Name $feature -IncludeManagementTools | Out-Null
        LogMessage "Installed $feature"
        Write-Output "Installed $feature"
    }
    else {
        Write-Output "$feature is already installed"
        LogMessage "$feature is already installed"
    }
}
```