Installation Guide for the XP Scaled topology

A guide to installing the Sitecore XP Scaled topology

October 6, 2021
Sitecore Experience Platform 10.0.3

Table of Contents

1. Choosing a topology

Before you install Sitecore Experience Platform, you must choose the topology or the type of instance that you want to install.

Sitecore supports the following topologies for on-premise installations by default:

- XP Single Developer (XP0)
- XM Scaled (XM1)
- XP Scaled (XP1)

This guide describes how to install the XP Scaled topology.

If you want to install one of the other topologies, download the installation guide for that topology.

You can configure the topology to match your business needs. There are several scalability options that you can use to achieve better performance, cope with greater website demand, and manage large amounts of website traffic. For more information about scaling, see Scaling options .

Sitecore also provides a number of cloud-based offerings. For more information, see our Cloud Services.

1.1. On-premise topology options

The following topologies are available:

Deployment topology	Description
XP Single Developer (XP0)	Use this topology for local development and testing. The Sitecore Experience Platform, runs as three single instances: Sitecore, xConnect, and Sitecore Identity server. The Experience Database (xDB) is partially included in the Sitecore and xConnect instances. **NOTE** For security and scalability reasons, in production environments, it is best practice to use the XM Scaled (XM1) or XP Scaled (XP1) configuration. This option runs all the services on a single server.

Deployment topology	Description
XM Scaled (XM1)	Use this topology if you are not planning to use the analytics and marketing features in the Sitecore Experience Platform. The Sitecore Experience Manager configuration (similar to CMS-only mode) runs the Content Delivery (CD), Content Management (CM) server roles and the Sitecore Identity server. **NOTE** When you select this topology, xDB and xConnect are not available. This option runs the services on one or more servers.
XP Scaled (XP1)	Use this topology if you are planning a fully featured Sitecore Experience Platform installation. The Sitecore Experience Platform configuration runs the following separated server roles: • Content Delivery • Content Management • Email Experience Dedicated Dispatch Server (optional) • Sitecore Identity • Processing • xConnect Collection • xConnect Collection Search • xDB Reference Data • xDB Automation Operations • xDB Automation Reporting • Sitecore Cortex™ Processing Engine • Sitecore Cortex™ Reporting service This option runs the services on one or more servers.

There are different ways to install Experience Manager (XM). You can use:

- Sitecore XM Scaled topology packages.
- Sitecore XP Single or the XP Scaled topology packages and then configure the installation to run in CMS-only mode.
- Sitecore containers.

In a Sitecore Experience Manager installation, the functionality that you can use is different depending on how you installed Sitecore XM. For more information, see the Sitecore Email Campaign Manager documentation.

NOTE
In a scaled environment, you must consider how to configure your session state provider. For more information, see the section Configure session state providers.

2. Sitecore Installation Framework

Before you can install Sitecore Experience Platform, you must set up Sitecore Installation Framework (SIF).

This chapter contains the following sections:

2.1. Set up Sitecore Installation Framework

The Sitecore Installation Framework (SIF) is a Microsoft® PowerShell module that supports local and remote installations of Sitecore Experience Platform.

SIF deploys Web Deploy Packages (WDP) by passing parameters to SIF configuration files through a Microsoft® PowerShell module and is fully extensible.

The Sitecore Experience Platform is designed to be secure-by-default. For developer environments all the required self-signed certificates are created automatically if you do not provide any.

In a production environment, you can provide your own certificates. In a non-production environment, you can choose to have the module generate the certificates for you.

You *must* set up SIF before you can install Sitecore Experience Platform.

2.1.1. Install the SIF Module using MyGet

The Sitecore Gallery is a public MyGet feed where you can download and install PowerShell modules created by Sitecore, including SIF.

To set up SIF:

1. In Windows, open PowerShell as an administrator.
2. To register the repository, in a PowerShell command line, run the following cmdlet:

```
Register-PSRepository -Name SitecoreGallery
-SourceLocation https://sitecore.myget.org/F/sc-powershell/api/v2
```

3. When prompted to install, press **Y**, and then press **Enter**.
4. To install the PowerShell module, run the following cmdlet:

```
Install-Module SitecoreInstallFramework
```

5. When prompted to install, press **Y**, and then press **Enter**.

Update the Sitecore Installation Framework Module

When a newer version of the SIF module is available, you can update to the latest version by running a PowerShell cmdlet.

To update the SIF module, run the following cmdlet:

```
Update-Module SitecoreInstallFramework
```

2.1.2. Validate the installation

After you install SIF, you can validate the installation to confirm that it is available for use.

> **NOTE**
> This validation only works if you have installed SIF to the *All users* (global) path.

To validate the installation, run the following cmdlet:

```
Get-Module SitecoreInstallFramework –ListAvailable
```

2.1.3. Run multiple versions of SIF

If you want to install a previous version of Sitecore Experience Platform on the same computer, you must also have the required SIF version installed. PowerShell uses the latest available version of a module in a session by default and you must import the specific version of SIF required for the version of Sitecore that you want to install.

The versions of SIF that are compatible with Sitecore Experience Platform:

Sitecore Experience Platform Version	SIF Version
9.0.X	1.2.1
9.1.0	2.0.0
9.1.1	2.1.0 or later
9.2.0	2.1.0 or later
9.3.0	2.2.0
10.X.X	2.3.0

To install a specific version of SIF:

1. Run the following cmdlet:

   ```
   Install-Module -Name SitecoreInstallFramework -RequiredVersion x.x.x
   ```

2. Enter the appropriate value in the `RequiredVersion` parameter.

2.1.4. Run a specific version of SIF

To run a specific version of SIF:

- Run the following cmdlet:

  ```
  Import-Module -Name SitecoreInstallFramework -Force -RequiredVersion x.x.x
  ```

You use the specified version for the remainder of the session.

The next time you start a PowerShell session it automatically uses the latest available version.

2.2. Install Sitecore Installation Framework manually

SIF is also available as a .zip package.

You can download the Sitecore Installation Framework packages from the Sitecore Downloads page.

> **NOTE**
> When you download the packages, it is possible that the `.zip` packages are marked as blocked by Microsoft Windows. To install SIF, you must first unblock the `.zip` packages.

2.2.1. Unblock a .zip package

To unblock a .zip package:

1. In Windows Explorer, navigate to the folder where you downloaded the `.zip` packages, and right-click the relevant `.zip` file.
2. Click **Properties**.
3. In the **Properties** dialog box, on the **General** tab, click **Unblock**.
4. Click **OK**.

2.2.2. Extract the Sitecore Installation Framework

The installation path depends on the location where you want to install the SIF. You can install it for all users (global path), for a specific user, or to a custom location:

Usage	Path
All users *(global path)*	`C:\Program Files\WindowsPowerShell\Modules`
Specific user	`C:\Users\<user>\Documents\WindowsPowerShell\Modules`
Custom location	Any path

For example, to make SIF available to all users, extract the SIF `.zip` package to the following path:

`C:\Program Files\WindowsPowerShell\Modules\SitecoreInstallFramework`

If you want to install SIF to a custom location, after the installation, you must import the module and specify the path to the file by running the following cmdlet:

```
Import-Module <custompath>\SitecoreInstallFramework
```

However, if you added SIF to an *All users* or *Specific user* path, you do not have to import the module, because this is done automatically.

2.3. Customizing the Sitecore Installation Framework

SIF lets you customize your installation within Microsoft PowerShell to add more tasks and features as required. For example, you can add steps to unpack a `.zip` archive of content, download files from other sources, or make a web request to call another service.

For more information about how to extend the installation framework, see the *Customize the Sitecore Installation Framework* section in the Sitecore Installation Framework Configuration Guide.

2.4. Run SIF remotely

PowerShell Remoting lets you run SIF configurations on a remote computer.

2.4.1. Enable PowerShell remoting

To enable PowerShell remoting:

- On the remote computer, in a PowerShell command line, run the `Enable-PSRemoting` cmdlet.

> **NOTE**
> You must enable PowerShell remoting for the user that completes the installation, and this user must have administrator rights to perform the deployment. For more information about securing or configuring a computer for remote access, see Microsoft's documentation.
>
> SIF uses SSL to create a remote PowerShell session. You must configure Windows Remote Management (WinRM) to work over HTTPS. For more information, see Microsoft Support.

2.4.2. Start a remote installation

To start a remote deployment:

1. Install SIF on the remote computer.
2. In a PowerShell command line, create a new remote session:

```
$session = New-PSSession -ComputerName <RemoteComputerName>
```

3. To copy all the required packages and SIF configuration files to the remote computer, specify the path and then run the following cmdlet:

```
Copy-Item -Path <sourcefile> -Destination -<remotePath> -ToSession $session
```

4. To start the installation, run the following cmdlet:

```
$session = New-PSSession -ComputerName  <RemoteComputerName>

Invoke-Command -Session $session { Import-Module SitecoreInstallFramework }
```

```
Invoke-Command -Session $session { Install-SitecoreConfiguration -Path
<configurationpath> }
```

NOTE
For more information about the `Invoke-Command` cmdlet, see the PowerShell documentation.

2.5. Install a SIF configuration file

Here is an example of how to use SIF to install a SIF configuration file on a local server.

To install a SIF configuration file on a local instance:

1. Launch PowerShell as an administrator.
2. Set up Sitecore Installation Framework.
3. To start the installation, run the `Install-SitecoreConfiguration` cmdlet, and specify the path to your SIF configuration file.
   For example, using the `sitecore-XP1.json` file:

   ```
   Install-SitecoreConfiguration -Path <configurationpath>\sitecore-XP1.json
   ```

   Optionally, the parameters declared in the SIF configuration files can be passed in at the command line by prefixing their name with a dash "-". For example:

   ```
   Install-SitecoreConfiguration -Path <configurationpath>\sitecore-XP1.json -SqlDbPrefix SC.
   ```

   In a PowerShell command line, you can pass additional parameters to control the installation process. For example:

Cmdlet	Description
`-Verbose`	Increases the amount of information that is logged.
`-Skip <taskname>`	Skips one or more tasks.

   For more information about the parameters that can be passed to the `Install-SitecoreConfiguration` cmdlet, run the following cmdlet:

   ```
   Get-Help Install-SitecoreConfiguration
   ```

NOTE
You can also use the `scinst` alias to run the `Install-SitecoreConfiguration` cmdlet.

3. Installation requirements

Before you can install Sitecore Experience Platform, you must ensure that you have all of the server and client requirements and prerequisites in place.

This chapter contains the following sections:

3.1. Server requirements

Before installing Sitecore Experience Platform, you must ensure that you have all of the requirements and prerequisites in place.

3.1.1. Hardware requirements for a server running a single Sitecore installation

To run a single Sitecore installation, the minimum configuration requirements are:

- 4 core processor
- 16 GB of RAM

> **NOTE**
> The recommended hardware requirements are for running the software on a single computer. For more information about running Sitecore on different kinds of hardware, consult your Sitecore partner or technical sales representative.

3.1.2. IIS requirements

You must use the version of IIS that your operating system supports. For more information about IIS and operating systems, see Microsoft's documentation.

Sitecore Experience Platform does not officially support any other ASP.NET web servers such as IIS Express, or Mono Web Server.

Sitecore Experience Platform neither supports nor allows multiple IIS website definitions to point to the same Sitecore web root.

3.1.3. Operating system requirements

Sitecore Experience Platform is only compatible with the client and server operating systems that support .NET Framework 4.8.0.

Sitecore Experience Platform can be hosted on the following Microsoft operating systems:

- Windows Server 2019
- Windows Server 2016
- Windows 10 (64-bit)

> **IMPORTANT**
> You must enable the Transport Layer Security (TLS) protocol version 1.2 on all of your Sitecore Experience Platform content management and Dedicated Dispatch servers (DDS).
>
> For more information about enabling TLS 1.2, see Microsoft's documentation.

> **IMPORTANT**
> Run Windows Update and install all the appropriate service packs and security updates on all of your Sitecore Experience Platform server and client computers.

3.1.4. .NET requirements

Sitecore Experience Platform requires .NET Framework 4.8.0.

Sitecore Identity server requires .NET Core 3.1 Windows Hosting Bundle or later.

You must apply any available updates to the .NET Framework on every Sitecore installation.

3.1.5. Microsoft Visual C++ 2015 redistributable requirements

Sitecore Experience Platform 9.0 Update-1 introduced a new prerequisite for the Microsoft Visual C++ 2015 Redistributable. For more information, see Microsoft's documentation.

> **NOTE**
> This redistributable may already be installed with Microsoft Windows. Without it, Sitecore Experience Platform will fail to start up with the message:
>
> *Could not load file or assembly 'ChilkatDotNet46.dll' or one of its dependencies. The specified module could not be found*.

3.1.6. Database requirements

Sitecore Experience Platform supports the following database servers:

- Microsoft SQL Server 2017
- Microsoft SQL Server 2019
  Required if you are going to use SQL Server for the Experience Database (xDB).
- Microsoft Azure SQL
  You must create the Azure SQL instance in advance and pass the address and the server administrator credentials in the corresponding parameters when you run the installation.
  For more information, see the Sitecore XP documentation
- MongoDB Server 4.0.5 - 4.0.13

Required if you are going to use MongoDB for the Experience Database (xDB) or as a Session State Provider.

NOTE
Sitecore Experience Platform does not support the MongoDB MMAPv1 storage engine because the storage engine does not support retryable writes.

NOTE
All of the databases in Sitecore Experience Platform use the *SQL_Latin1_General_CP1_CI_AS* collation, except the *Reference Data* database that uses the case sensitive *Latin1_General_CS_AS* collation. This is because comparisons in the *Reference Data* database are case sensitive and they are not case sensitive in the other databases.

3.1.7. Enable Contained Database Authentication

When you use Web Deploy Packages, you must ensure that the target SQL Server is configured to allow users and logins to be contained at the database level.

To configure the target SQL Server:

1. Launch Microsoft SQL Server Management Studio and log in as an administrator.
2. Run the following new query:

```
EXEC sp_configure 'contained', 1;
RECONFIGURE;
```

NOTE
For more information about the contained database authentication option, see Microsoft's documentation.

3.1.8. Search indexing requirements

Sitecore Experience Platform supports:

- Solr 8.4.0
  Solr is the default search provider and supports both content search and analytics search. To use the Sitecore Installation Framework (SIF), you must have Solr installed.

NOTE
Support for Lucene was removed in Sitecore XP 9.3.0.

For more information about how to install and manage these index providers in Sitecore Experience Platform, see the Sitecore documentation.

3.1.9. Installing Solr

> **NOTE**
> If you plan to use the Sitecore Installation Assistant to install Sitecore XP, it can install Solr for you and you can skip this section.
>
> If you plan to use the Sitecore Installation Framework (SIF) to install Sitecore XP, you must follow the steps described in this section.

The standard Sitecore Experience Platform configuration requires Solr. The Sitecore Experience Platform is secure by default, you must, therefore, enable SSL for Solr.

Before you run the SIF, you must:

- Install Solr and configure it to run as a Windows service.
- Enable and set up SSL for Solr.

The `Solr-Singledeveloper.json` deployment configuration file installs and configures Solr with SSL.

For local testing and development, you can set up a self-signed certificate. The Apache Solr Reference guide has more information about creating a self-signed certificate.

For more information about installing Solr, see the section Install a search provider in a scaled solution.

3.1.10. Antivirus software considerations

Some antivirus software can have a detrimental effect on the performance of ASP.NET applications, including Sitecore. We recommend that you use only antivirus scanners that are certified for your operating system.

For more information about the certified products, see the Windows Server Catalog website.

For optimal performance, ensure that the following folders are *not* scanned by your antivirus software:

- The site root folder
- The data folder defined in the `web.config` file
- The folder that contains the actual Sitecore database files
- The `C:\Windows\Temp or {app_pool user profile}\Temp` folder

> **NOTE**
> Active file scans from antivirus tools can significantly impact the performance of search indexing software. This can lead to poor user experience or slow system performance. Consider turning off any antivirus tools or modifying antivirus settings on the search index server to exclude the application data folders from scans. For more information about your search indexing software, consult the related documentation.

3.1.11. Prerequisites for using the Sitecore Installation Framework

To use the Sitecore Installation Framework to install Sitecore Experience Platform in an on-premise environment, you must download and install:

- Microsoft PowerShell® version 5.1 or later

3.2. Client requirements

3.2.1. Browser requirements

Sitecore Experience Platform clients are browser-based user interfaces. Sitecore Experience Platform has been tested and can run on the following browsers:

- Mozilla Firefox
- Google Chrome
- Microsoft Edge
- Apple Safari (Mac only)

> **NOTE**
> Sitecore Experience Platform supports all the current stable versions of these browsers.
>
> Although Sitecore Experience Platform supports the tested versions of the listed browsers, newer browser revisions are continually released. Sitecore will support the latest revisions of these browsers.

For more information about browser compatibility, see the Sitecore compatibility table.

3.2.2. Client hardware requirements

Sitecore Experience Platform has the following minimum client hardware requirements:

- Processor: Intel Pentium 4, 2 GHz or faster
- RAM: 512 MB minimum, 1 GB – recommended
- TCP/IP connection at 512 Kbps or faster to the Sitecore XP host
- 1024 x 768 or greater screen resolution required for advanced operations

In general, the client computer should meet the hardware requirements of the browser that you use.

You do not have to install any additional software on the Sitecore Experience Platform clients that access Sitecore Experience Platform servers.

3.3. Server file system requirements

Before you install Sitecore Experience Platform, you must fulfill all the requirements.

3.3.1. File system permissions for ASP.NET requests

Sitecore Experience Platform executes requests for ASP.NET resources and all the .NET code running within the application with the permissions of the account configured as an identity for the website's application pool. This account requires Modify permissions for all the files, folders, and subfolders under the `\wwwroot\<YourWebsiteFolder>` folder.

> **NOTE**
> The Sitecore Installation Framework automatically sets all the required permissions to your website folder. If you deploy Sitecore through a manual configuration, such as a PowerShell script or similar, you must set the correct file system permissions.

The default account that is used to process ASP.NET requests in the different versions of IIS:

IIS version	Default ASP.NET account name
10	NETWORK SERVICE

If you select a different user account to process the ASP.NET requests, you must also grant this account the *Modify* permissions.

For more information about application pool identities and specifically about assigning rights to the *AppPoolIdentity* account, see Microsoft's documentation.

3.3.2. File system permissions for system folders

To load the .NET runtime and ASP.NET resources that are used to process the ASP.NET requests, the worker process that hosts the Sitecore Experience Platform application requires access to multiple system files and folders that are not distributed as a part of Sitecore Experience Platform, but are installed as a part of the Windows Operating System and the .NET framework. For more information about built-in groups and accounts in IIS, see Microsoft's documentation.

Most of these permissions are granted by IIS to all ASP.NET applications, automatically making the application pool identity account a member of the *IIS_IUSRS* security group.

However, in certain environments, you must manually grant permissions for the Application Pool Identity to the following system locations:

Default location	Required permissions	Comments
`%WINDIR%\temp\`	Modify	To install Sitecore Experience Platform, you must assign the *Modify* access rights to the `\temp` folder for the ASP.NET and/or IUSR accounts.
`%WINDIR%\Globalization\`	Modify	Required for registering custom languages by the .NET Framework correctly.
`%PROGRAMDATA%\Microsoft\Crypto`	Modify	Required for storing cryptographic keys used for encrypting/decrypting data.

These variables have the following default values:

Variable	Default value
`%WINDIR%`	`C:\Windows`
`%PROGRAMDATA%`	`C:\ProgramData` for IIS 10 and later

> **NOTE**
> The Sitecore Installation Framework specifies the required permissions for certificates under the `\Crypto` folder.

3.3.3. UNC share is not supported

You must install Sitecore Experience Platform on a local drive, not a Universal Naming Convention share.

3.3.4. Sitecore cannot operate from a virtual directory

Sitecore Experience Platform does not support operating from a virtual directory.

4. Install the prerequisites

You must install the prerequisites. You can install them automatically or manually.

This chapter contains the following sections:

4.1. Automated installation of prerequisites

A predefined SIF configuration file `Prerequisites.json` is distributed in the configuration packages. This file downloads and installs most of the prerequisites.

This file does not install:

- Microsoft SQL Server
- Solr
- MongoDB (optional)
  The Sitecore XM topologies do not require MongoDB. However, if you want to use MongoDB as the session state provider, you must install MongoDB.
- Microsoft Machine Learning Server (optional)

For more information about how to use this file, see the section Install a SIF configuration file.

> **NOTE**
> SIF does not install any of the prerequisite software if it is already installed. You must ensure that you install the correct versions.

4.2. Manual installation of the prerequisites

To install most of the Sitecore server roles, you must have the following prerequisites:

- Microsoft Web Platform Installer (WebPI) 5.0
- IIS, ASP.NET 4.8, and the Web administration PowerShell Module
- SQL PowerShell Module

The Sitecore server roles require:

Requirement	Feature	Details
WebAdministration module	Supports IIS management.	When you configure a computer with IIS, the *WebAdministration* module is installed automatically.
Web Deploy 3.6 for Hosting Servers	Supports the installation of Web Deploy Packages.	To install this tool, use the Web Platform Installer.
URL Rewrite 2.1	Supports URL rewrites for Sitecore when installed as a Web Deploy Packages.	To install this tool, use the Web Platform Installer.
Microsoft SQL Server Data-Tier Application Framework (DacFx) version 2017 (x86 and x64)	Supports the installation of .dac files in Web Deploy Packages	Download and install DacFx x86. Download and install DacFX x64. This must be installed on servers that have been assigned a Sitecore server role and where you are going to install DAC packages: In the XP Scaled topology: • Content Management • Processing • Collection • Reference Data • Sitecore Cortex™ Processing Engine • Sitecore Cortex™ Reporting service To ensure that DacFx works correctly, you must install its system requirements including Microsoft System CLR Types for SQL Server 2017. If you are running an x64 environment, you must install both the x64 and x86 versions of DacFx and SQLSysCLRTypes. **NOTE** If DACFx fails to install, you can see the following error message when you use the framework: *The SQL provider cannot run with dacpac option because of a missing dependency. Please make sure that DACFx is installed.* For information about how to resolve this error, see this Sitecore Knowledge Base article.
Microsoft ODBC Driver 13 for SQL Server Microsoft Command Line Utilities 13 for SQL Server	Required by the Sitecore Installation Framework	You must also install these utilities on the xConnect application server before running the Sitecore Installation Framework installation template for xConnect or the Single Developer workstation.

4.2.1. Clear the Web Platform Installer download cache

If the Web Platform Installer hangs or freezes during the installation, you must restart and clear the download cache.

To clear the Web Platform Installer download cache:

1. Launch the Web Platform Installer.
2. In the bottom pane, click **Options**.
3. In the **Change Options** dialog box, scroll down to the **Installer cache** section and click **Delete installer cache folder**.
4. Click **OK**.

4.3. Install a search provider in a scaled solution

Sitecore Experience Platform supports Solr and Azure Search as search providers. The Solr and Azure Search providers both support content search and analytics search.

> **NOTE**
> Support for Lucene was removed in Sitecore Experience Platform 9.3.0.

For more information about the supported search providers, see Using Solr or Azure Search.

For local testing and development, you can set up a self-signed certificate. For more information about creating a self-signed certificate, see the Apache Solr Reference guide.

4.3.1. Install the Solr Certificate

You must install the Solr certificate on the servers that perform the following roles:

- Content Management
- xConnect Collection Search
- xConnect Indexer service – if you install it on a separate server.
  For more information about the xConnect Indexer service role, see the Sitecore documentation.

> **NOTE**
> The new Dedicated Dispatch Server (DDS) role that was introduced in Sitecore Experience Platform 9.0. Update 1 can only be configured on a CM server and therefore requires the Solr certificate.
>
> For more information about configuring a Dedicated Dispatch Server, see the EXM documentation.

4.3.2. Deploy the collections and ConfigSets to SolrCloud in an on-prem solution

The SolrCloud deployment configuration templates for SIF deploy the required ConfigSets and collections to remote Zookeeper and SolrCloud installations.

Prerequisites

Before you deploy the ConfigSets and collections to SolrCloud, you must:

- Download the Solr for Windows installation archive and save it locally, for example, `C:\solr`. You use this file to access the `solr.cmd.zookeeper` command line interface. If you have already installed the required version of Solr you can skip this step.
  To use SolrCloud, you do not need to install Solr as a local service.
- Locate the `Sitecore 10.0.3 rev. 006577 (WDP XP1 packages).zip` package and in the `XP1 Configuration files 10.0.3 rev. 006577 .zip` file extract the following configuration files:
  - `SolrCloud-SitecoreConfigSets.json`
  - `SolrCloud-Sitecore-Collections.json`
  - `SolrCloud-XConnect-Collections.json`

Install the Sitecore content and the xDB ConfigSets

To deploy the Sitecore content and the xDB ConfigSets to SolrCloud:

1. In a PowerShell command line, go to the folder where you saved the installation configuration files and run the following cmdlet with the appropriate parameters:

```
Install-SitecoreConfiguration `
   -Path .\SolrCloud-SitecoreConfigSets.json `
   -SolrInstallRoot <path to solr binaries from step 1, default: c:\> `
   -SolrDomain <zookeeper domain name, default: localhost> `
   -ZookeeperPort <zookeeper port, default: 9983>  `
   -CorePrefix <prefix for configSet names, default: sitecore> `
   -DefaultConfigSetName <config set to use as base, default: _default>
```

2. Verify that the ConfigSets are deployed to Zookeeper.

Deploy the SolrCloud content collections

To deploy the SolrCloud content collections:

1. In a PowerShell command line, run the following cmdlet with the appropriate parameters:

```
Install-SitecoreConfiguration `
   -Path .\SolrCloud-Sitecore-Collections.json `
   -SolrDomain <solrcloud domain name, default: localhost> `
   -SolrPort <solrcloud port, default: 8983> `
   -CorePrefix <prefix for collection name, default: sitecore> `
   -NumShards <number of shards for the collection, default: 1> `
   -ReplicationFactor <number of replicas for each shard, default: 1> `
   -MaxShardsPerNode <maximum number of shards for each Solr node, default: 10> `
   -AutoCreateFields <automatically add new fields to schema, default: false>
```

2. Verify that the following CMS collections are deployed to SolrCloud:
   - `<prefix>_core_index`
   - `<prefix>_master_index`
   - `<prefix>_web_index`
   - `<prefix>_marketingdefinitions_master`

- <prefix>_marketingdefinitions_web
- <prefix>_marketing_asset_index_master
- <prefix>_marketing_asset_index_web
- <prefix>_testing_index
- <prefix>_suggested_test_index
- <prefix>_fxm_master_index
- <prefix>_fxm_web_index
- <prefix>_personalization

Deploy the SolrCloud collections for xDB

To deploy the SolrCloud Collections for xDB:

1. In a PowerShell command line, run the following cmdlet with the appropriate parameters:

```
Install-SitecoreConfiguration `
   -Path .\SolrCloud-XConnect-Collections.json `
   -SolrDomain <solrcloud domain name, default: localhost> `
   -SolrPort <solrcloud port, default: 8983> `
   -CorePrefix <prefix for collection name, default: sitecore> `
   -NumShards <number of shards for the collection, default: 1> `
   -ReplicationFactor <number of replicas for each shard, default: 1> `
   -MaxShardsPerNode <maximum number of shards for each Solr node, default: 10> `
   -AutoCreateFields <automatically add new fields to schema, default: false>
```

2. Verify that the following xDB collections are deployed to SolrCloud:
   - <prefix>_xdb_internal
   - <prefix>_xdb_rebuild_internal
3. Verify that the following xDB aliases are deployed to SolrCloud:
   - <prefix>_xdb
   - <prefix>_xdb_rebuild

Deploy Sitecore with SolrCloud

After you have configured SolrCloud, before you install Sitecore XP, you must ensure that the SolrCloud parameter is configured correctly in the connection string .

To install the Sitecore platform with a connection to the newly configured SolrCloud instance, in the SIF deployment script, in the SolrUrl parameter, add the `;solrCloud=true` setting.

If you are going to deploy the server roles separately, update the SolrUrl parameter in following deployment configuration files:

- xconnect-xp1-collectionsearch.json
- sitecore-XP1-cd.json
- sitecore-XP1-cm.json

NOTE
When you use SolrCloud, you do not need to deploy the `sitecore-solr.json` and `xconnect-solr.json` templates.

5. Set up a production environment

This chapter describes how to install Sitecore Experience Platform in a scaled environment for production or developer purposes.

This chapter contains the following sections:

5.1. Set up the certificates

Sitecore Experience Platform is designed to be secure by default. You must therefore implement HTTPS across the platform.

Server Certificate Authentication

All communication between Sitecore instances occurs over the default HTTPS configuration. This includes the xConnect web services, the Sitecore Identity server, Microsoft Machine Learning Server, and the Solr search provider. HTTPS requires that you obtain and set up certificates for the Secure Sockets Layer (SSL) before you install the platform.

Server authentication uses a server-side certificate and a private key to encrypt traffic between the HTTP client and the HTTP server application. This type of authentication prevents unencrypted content from traveling over an unsecured network. It does not identify who the client is and the server authentication alone does not determine who can connect to the server.

Client Certificate Authentication

The xConnect server roles support an additional layer of security, referred to as SSL Client Certificate Authentication. SSL Client Certificate Authentication validates that the individual HTTP client is authorized to connect to the HTTP server. SSL Client Certificate Authentication requires that the HTTP client device is configured with a specific client certificate and private key, or thumbprint, which is used to connect to the protected SSL server.

Because xConnect web services use server-to-server communication and are non-interactive, the client certificate allows the Content Management server role and other server roles to connect securely to WebAPI JSON services.

> **IMPORTANT**
> In local developer environments, self-signed certificates can be used to develop Sitecore solutions. Due to potential security concerns, you must not use self-signed certificates in production environments.

5.1.1. Set up server certificate SSL authentication on IIS

You must obtain and install the server certificates before you run SIF. For more information about how to set up SSL in IIS, see Microsoft's documentation.

The following table lists the full set of server authentication certificates for this topology:

	XP Scaled (XP1)
	Content Management
	Processing
	Sitecore Cortex™ Reporting Service
	Sitecore Identity server
	xConnect Collection
	xConnect Collection Search
	xDB Reference Data
	xDB Automation Operations
	xDB Automation Reporting

For each certificate, you must use the site name in the common name **CN** field in the certificate. For example, if the name that you want to use for the Content Management IIS site is *CM_test*, you must use this name when you create the Content Management certificate.

Install the server certificates

After you obtain the relevant certificates, you must install them.

To install the server certificates:

1. Install the server authentication certificate in the system certificate store folder:

   ```
   Certificates (Local Computer)\Personal
   ```

   For information about how to install a private key certificates, see PowerShell Import-Certificate from Microsoft.

2. If you created a self-signed certificate, install the self-signed authority certificate for the SSL certificate in the following folder:

   ```
   Certificates (Local Computer)\Trusted Root Certification Authorities
   ```

For XP Scaled (XP1), after you obtain all the server certificates, you must install them on the required servers:

XP Scaled (XP1)

Role Name	Server Certificate
Content Management	Sitecore Identity server
	Processing
	xConnect Collection Search
	xDB Reference Data
	xDB Automation Operations
	xDB Automation Reporting
	Sitecore Cortex™ Reporting Service
Content Delivery	Content Management
	xConnect Collection
	xDB Reference Data
	xDB Automation Operations
Processing	xConnect Collection
xConnect Collection	*None required*
xConnect Collection Search	*None required*
xDB Reference Data	*None required*
xDB Automation Operations	xConnect Collection
	xDB Reference Data
xDB Automation Reporting	*None required*
EXM Dedicated Dispatch Server	Sitecore Identity server
	Processing
	xConnect Collection
	xConnect Collection Search
	xDB Reference Data
	xDB Automation Operations
	xDB Automation Reporting
Sitecore Cortex™ Reporting service	*None required*
	xConnect Collection
	xConnect Collection Search

5.1.2. Setting up the client certificates

You must obtain and install the client certificates for a production environment before running SIF. If you do not provide the certificates, SIF creates self-signed certificates when they are required.

The client certificate is typically installed on the Windows Server that connects to the server where xConnect is deployed. The client certificate is stored in the certificate store for either a specific user or the entire server.

The thumbprint of the client certificate is specified on the server that you are connecting to (the destination). In this case, the xConnect server and only clients with the correct certificate and matching thumbprint are allowed to connect.

In production environments, different client certificates are used for different application roles with the aim of isolating the servers, in the event of a key being compromised.

For development purposes, you can use a single client certificate to validate that authentication will work as expected when you move to a production environment.

The following table lists the full set of client authentication certificates for this topology:

XP Scaled (XP1)
xConnect Collection
xConnect Collection Search
xDB Reference Data
xDB Automation Operations
Sitecore Cortex™ Reporting service

Install the client certificate

After you have obtained the certificates, you must install them.

To install the client certificate:

1. Install the client authentication certificate, including the private key, in the `Certificates (Local Computer)\Personal` folder for each required role.
   For information about how to install a private key certificates, see PowerShell Import-Certificate from Microsoft.

   > **IMPORTANT**
   > When you import the client certificate, you must select the **Allow private key to be exported** option.

2. If you created a self-signed certificate, you must install the self-signed authority certificate used to create the client authentication certificate in the `Certificates (Local Computer)\Trusted Root Certification Authorities` folder.
3. You must add the thumbprint for the certificates that you installed in the previous step to the following roles:
   - xConnect Collection
   - xConnect Collection Search
   - xDB Reference Data
   - xDB Automation Operations
   - xDB Automation Reporting
4. In the `/App_Config/AppSettings.config` file, add the thumbprint to the `<add key="validateCertificateThumbprint" value="YOUR_CERTIFICATE_THUMBPRINT" />` setting.

This defines which client certificate is used for authentication.

The following tables provide details about the client certificates required for each role:

XP Scaled (XP1)

Role name	Client Certificates	Associated connection strings containing client thumbprint

Content Management	xConnect Collection Search
	xDB Reference Data
	xDB Automation Operations
	xDB Automation Reporting
	Sitecore Cortex Reporting Service
		`xconnect.collection.certificate`
		`xdb.referencedata.client.certificate`
		`xdb.marketingautomation.reporting.client.certificate`
		`xdb.marketingautomation.operations.client.certificate`
		`sitecore.reporting.client.certificate`

Content Delivery	xConnect Collection
	xDB Reference Data
	xDB Automation Operations
		`xconnect.collection.certificate`
		`xdb.referencedata.client.certificate`
		`xdb.marketingautomation.operations.client.certificate`

Processing	xConnect Collection	`xconnect.collection.certificate`

	xConnect Collection
	xConnect Search
		`xconnect.collection.certificate`
		`xconnect.configuration.certificate`
		`xconnect.search.certificate`

Sitecore Marketing Automation Engine	xConnect Collection	`xconnect.collection.certificate`
xConnect Collection	*None required - because this role does not make calls to other roles.*	-
xConnect Collection Search	*None required - because this role does not make calls to other roles.*	-
xDB Reference Data	*None required - because this role does not make calls to other roles.*	-
xDB Automation Operations	xConnect Collection	`xconnect.collection.certificate`
xDB Automation Reporting	*None required - because this role does not make calls to other roles.*	-

You must also ensure that client certificate private keys permissions and read access are granted to the users under which your services are running. SIF does this automatically.

By default, these users are:

- The *ApplicationPoolIdentity* for the web sites.
- The *Local Service* account for Windows services.

5.1.3. Set up an SSL certificate for Solr

As described in the Install Solr section, if you want to use the Experience Database (xDB) and xConnect, you must enable SSL for Solr.

> **NOTE**
> In production environments, the Solr certificate must be provided and signed by an authorized provider. However, in development environments, the certificates can be generated and signed locally.

If you created a self-signed certificate, install the self-signed authority certificate for the SSL certificate in the following certificate store:

```
Certificates (Local Computer)\Trusted Root Certification Authorities
```

You must install the Solr SSL certificate for the following server roles:

XP Scaled (XP1)
Content Management
xConnect Collection Search

5.2. Install the Sitecore XP scaled topology

Once you have obtained the required certificates, you can run SIF and install the Sitecore XP scaled topology. You can install any of the configurations for dedicated server roles, on single or multiple servers.

The server roles are defined as a part of your desired scaling configuration.

5.2.1. Use SIF to install the Sitecore XP Scaled topology

To run SIF and install Sitecore XP:

1. If you have not already done so, as an administrator, in a PowerShell command line, run the following cmdlet:

   ```
   Import-Module SitecoreInstallFramework
   ```

2. To install the Solr cores, run the following cmdlets with the required parameters for your server roles:

   ```
   Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\XConnect\Solr
   \xconnect-solr.json"
   ```

```
Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\Platform\Solr
\sitecore-solr.json"
```

For more information about installing Solr, see the section Install a search provider in a scaled solution.

3. To install the server roles, run the following cmdlets with the required parameters for your server roles:

For information about setting up EXM, see the EXM documentation.

5.2.2. Specifying the certificates during installation

To install Sitecore with your pre-installed certificates, when you run the `Install-SitecoreConfiguration` cmdlet, you must provide the certificates as parameters.

SIF searches for the certificates in the following path, by default:

```
Cert:\Localmachine\My
```

You can change the storage location.

Change the default location of the certificates

To change the default location of the certificates used for the deployment:

- In a text editor, open the relevant `.json` file, and in the `Variables` section, change the default store value:

```
"Security.CertificateStore": "Cert:\\Localmachine\\My"
```

Specify the names or thumbprints of the certificates

You must specify the names or thumbprints of the certificates that you created and installed earlier in this guide as parameters. For example:

1. For the *client* authentication certificate:

```
-XConnectCert "xConnect_client"
```

or

```
-XConnectCert "738F45F610221990DA2FE059E1D8C2ECCB5067F2"
```

> **NOTE**
> In the PowerShell command line parameter, you must specify the client certificate thumbprint in capital letters.

2. For the server authentication certificate, for example, for an instance with the name `"CM_test"`:

```
-SSLCert "CM_test"
```

or

```
-SSLCert "2205a94867ee99e3b29ea7a9ac5a7646d43fd88b"
```

5.2.3. Skip database deployment when you install a server role

When you install a server role, you do not have to install the databases when you deploy the WDP packages.

A new the *SkipDatabaseInstallation* parameter has been added to the following deployment configuration files: ·

- `sitecore-XP1-cm.json`
- `sitecore-XP1-prc.json`
- `xconnect-xp1-collection.json`
- `xconnect-xp1-CortexProcessing.json`
- `xconnect-xp1-CortexReporting.json`
- `xconnect-xp1-ReferenceData.json`

The *SkipDatabaseInstallation* parameter is set to *false* by default and all the databases are installed when you deploy the WDP packages.

> **NOTE**
> The *SkipDatabaseInstallation* parameter is not supported by the SingleDeveloper and XM Scaled deployment configuration files.

To deploy a Sitecore server role without installing the databases:

1. Ensure that you have already installed the databases that you need for the server role.
2. Ensure that the database names contain the prefixes that you are going to use for this deployment.
3. In the Powershell deployment command, enter all the required parameters.
4. In the `c:\resourcefiles\Role-Remote.json` file, ensure that the `SIFVersion` parameter matches the latest version of SIF.
5. In the Powershell deployment command, enter following parameter in the corresponding config files:
   - `SkipDatabaseInstallation: true`
6. Pass the database user names and passwords, for example:

```
“SqlCoreUser” : “mycoreuser”

“SqlCorePassword” : “mycorepassword”
```

Install to a custom website folder

With Sitecore XP, you can specify the folder where the website is installed.

The deployment configuration files for each server role contain a *SitePhysicalRoot* parameter.

When you set this parameter, the website is installed in the *[SitePhysicalRoot]\[SiteName]* folder.

If you leave this parameter with the default value, the website is installed in the default IIS *wwwroot* folder.

> **NOTE**
> This parameter is supported by the SingleDeveloper and the Distributed deployment configuration files and the *SitePhysicalRoot* path that you provide is created for every website.

5.3. Distributed installation script for the Sitecore XP Scaled topology

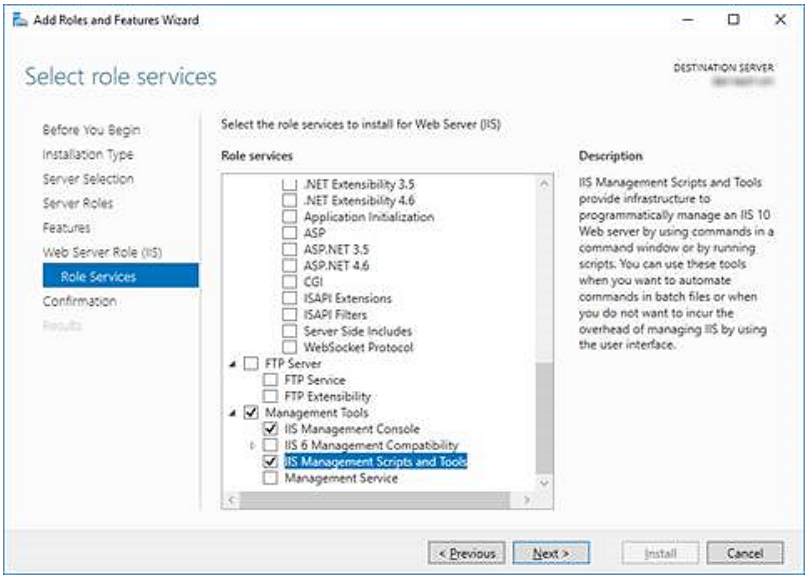
To simplify your scaled installation on multiple servers, you can use a PowerShell script to install the Sitecore XP Scaled topology. For more information, see Architecture overview in Sitecore documentation.

> **NOTE**
> If you use this script to install the Sitecore XP Scaled topology, each server role must be installed on a separate computer and there must be no previous Sitecore installation on any of these computers.

5.3.1. Distributed installation script prerequisites

To prepare the servers, you must perform the following steps on each server:

1. Enable PowerShell Remoting.
   For more information about PowerShell Remoting, see the section Run SIF remotely.
2. Install all the prerequisites.
   For more information about the prerequisites, see the section Install the prerequisites.
3. Create the `c:\resourcefiles` folder on every computer that will be assigned a Sitecore server role or will run Solr.
4. Configure the Web Server Role (IIS).



5.3.2. Run the distributed installation script for the XP Scaled topology

To edit and run the Distributed XP topology installation script:

1. Create a folder called `c:\resourcefiles`.
2. Download and save the following SIF resource files in this folder:
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_cd.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_cm.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_prc.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_rep.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_xp1collection.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_xp1collectionsearch.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_xp1marketingautomation.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_xp1marketingautomationreporting.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_xp1cortexprocessing.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_xp1referencedata.scwdp.zip`
   - `Sitecore 10.0.3 rev. 006577 (OnPrem)_xp1cortexreporting.scwdp.zip`
   - `Sitecore.IdentityServer 6.0.0 r00301 (OnPrem)_identityserver.scwdp.zip`
   - `IdentityServer.json`
   - `sitecore-solr.json`
   - `sitecore-XP1-cd.json`
   - `sitecore-XP1-cm.json`

- `sitecore-XP1-prc.json`
- `sitecore-XP1-rep.json`
- `createcert.json`
- `importcert.json`
- `xconnect-solr.json`
- `xconnect-xp1-collection.json`
- `xconnect-xp1-collectionsearch.json`
- `xconnect-xp1-MarketingAutomation.json`
- `xconnect-xp1-MarketingAutomationReporting.json`
- `xconnect-xp1-CortexProcessing.json`
- `xconnect-xp1-ReferenceData.json`
- `xconnect-xp1-CortexReporting.json`
- `Role-Remote.json`
- `XP1-Distributed.json`
- `XP1-Distributed.ps1`

3. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.
4. In the `c:\resourcefiles` folder, edit the `XP1-Distributed.ps1` script and update each line with the appropriate settings for your environment.
5. In a PowerShell command line, go to the `c:\resourcefiles` folder and run the following command:

```
.\XP1-Distributed.ps1
```

After you have edited and run the installation script, you must complete the post-installation steps described in the chapter Sitecore XP post installation steps.

5.4. Enable telemetry reporting in production

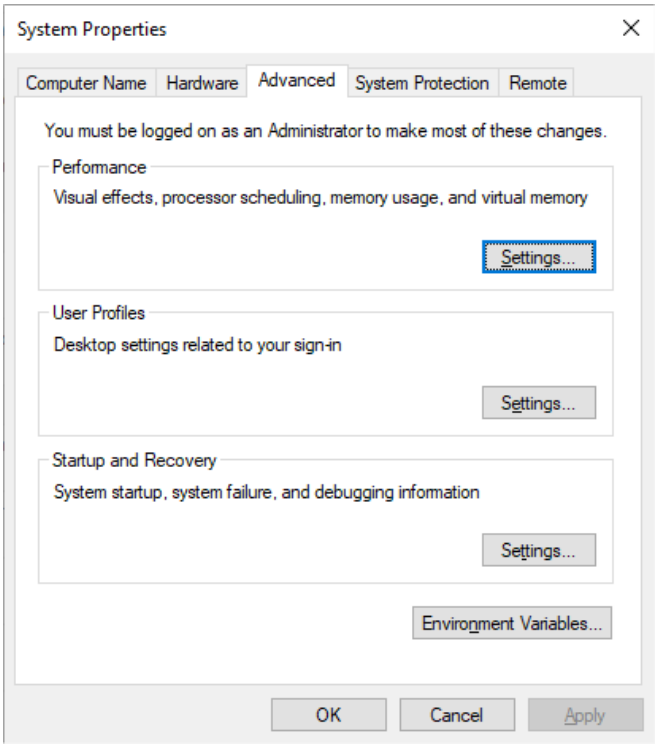
The Sitecore telemetry reporting feature gathers information that helps Sitecore understand how customers use our products. The environment type (production or non-production) helps us to associate the features used with the appropriate use-cases.

The telemetry system assumes Sitecore Experience Platform is running in a non-production environment by default. When you set up a production environment, you must therefore define a system variable to report the environment type.

To define a system variable:

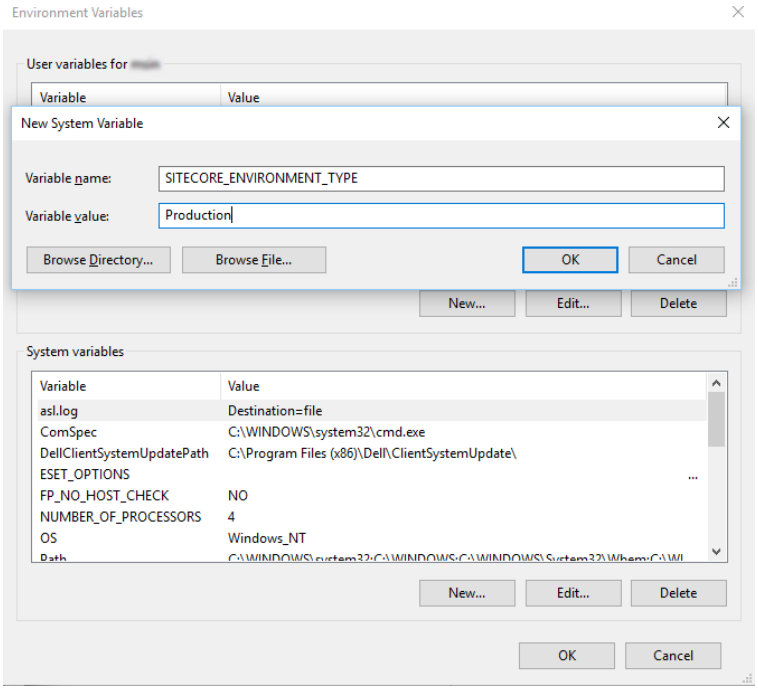
1. In the **Control Panel**, click **System**, **Advanced system settings**.

2. In the **System Properties** dialog, click **Environment Variables**.



3. In the **Environment Variables** dialog, in the **System variables** section, click **New**.

4. In the **New System Variable** dialog, in the **Variable name** field, enter SITECORE_ENVIRONMENT_TYPE and in the **Variable value** field, enter *Production*.



5. Restart the computer.

6. Post-installation steps

After you use SIF to install Sitecore XP, you must:

- Configuring Sitecore Identity server
- Configure Azure Cognitive Search
- Configure the MongoDB provider for xConnect
- Configure high availability for xConnect
- Rebuild the search indexes and the Link database
- Deploy the marketing definitions
- Content expiration
- Configure geo-location lookup
- Configuring session state providers
- Warm up the servers
- Security hardening
- Configure Email Experience Manager
- Synchronize the time source
- Import the client translations

6.1. Configuring Sitecore Identity server

The Sitecore Identity server only works with HTTPS, and you must generate a certificate for it.

The Sitecore Identity server configuration requires the following additional parameters:

- `allowedCorsOrigins` – a pipe-separated list of instances (URLs) that are allowed to login via Sitecore Identity. This can be a Sitecore instance in the XP0 topology, or all the CM/CD servers in a scaled environment.
- `ClientSecret` – a random string value that must be identical on both the client and server side.
  - On the client side, it is stored in the connection strings on the CM server – `sitecoreidentity.secret`.
  - On the server side, it is stored in the `<IdentityServer folder>\ Config\production \Sitecore.IdentityServer.Host.xml` file, in the `ClientSecrets` node.
- `PasswordRecoveryUrl` – the client URL (CM server).
  If a user forgets their password, they are redirected to the appropriate Sitecore server to fill in the form for password recovery.

You must also register the Identity server on the client side. The Identity server is configured in the `\App_Config\Sitecore\Owin.Authentication.IdentityServer \Sitecore.Owin.Authentication.IdentityServer.config` configuration file `–sc.variable "identityServerAuthority"`.

For more information see Sitecore Identity documentation.

6.2. Configure Azure Cognitive Search

Sitecore Experience Platform supports Solr and Azure Cognitive Search as search providers.

> **IMPORTANT**
> In Sitecore XP 10.0.0, support for Azure Cognitive Search was deprecated and will be completely removed in a later release.

In on-premise solutions, SIF requires Solr. When the installation is completed, you can switch to Azure Cognitive Search.

To configure Azure Cognitive Search for xConnect:

1. In the `xconnect\App_data\config\sitecore\CollectionSearch` folder:
   Enable the following configuration file by removing the `.disabled` extension:
   - `sc.Xdb.Collection.IndexReader.AzureSearch.xml.disabled`

   Disable the following configuration files by adding the `.disabled` extension:
   - `sc.Xdb.Collection.IndexReader.SOLR.xml`
   - `sc.Xdb.Collection.WebClient.SOLR.xml`

2. In the `xconnect\App_data\jobs\continous\IndexWorker\App_data\config \sitecore` folder, enable the following configuration files by removing the `.disabled` extension:
   - `\SearchIndexer\sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled`
   - `\CollectionSearch \sc.Xdb.Collection.IndexReader.AzureSearch.xml.disabled`

3. In the `xconnect\App_data\jobs\continous\IndexWorker\App_data\config \sitecore` folder, disable the following configuration files by adding the `.disabled` extension:
   - `\CollectionSearch\sc.Xdb.Collection.IndexReader.SOLR.xml`
   - `\CollectionSearch\sc.Xdb.Collection.WebClient.SOLR.xml`
   - `\SearchIndexer\sc.Xdb.Collection.IndexWriter.SOLR.xml`

4. In the `\Search-indexer \sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled` file, the `DataReplicationTimeoutMilliseconds` setting is disabled by default.
   If you use more than one replica, you must enable this setting.
5. In the following connection string files:

   - `xconnect\App_Config\ConnectionStrings.config`
   - `xconnect\App_data\jobs\continuous\IndexWorker\App_config \ConnectionStrings.config`

   Add (uncomment) and update the Azure Search connection string, the default name is `collection.search`.

```
<add name="collection.search" connectionString="serviceUrl=https://
[service].search.windows.net/;indexName=[index name];apiKey=[API Key]"/>
```

To configure content search, see the following topics:

- Configure a search and indexing provider.
- Configure Azure Cognitive Search

6.3. Configure the MongoDB provider for xConnect

You must use MongoDB Server 4.0.5 - 4.0.10 as it contains important fixes that are essential for the xConnect MongoDB data provider.

6.3.1. Configure the xConnect MongoDB data provider

The xConnect platform is installed with the SQL provider for the collection database by default.

To enable the MongoDB provider, you must modify the configuration files for all the server roles that you use in your topology:

1. Enable the `sc.Xdb.Collection.Data.MongoDb.xml.disabled` configuration file by removing `.disabled` extension.
2. Disable the `sc.Xdb.Collection.Data.Sql.xml` configuration file by adding the `.disabled` file extension.
3. Update the collection connection string to point to the MongoDB instance.
   You must also update the collection connection string in the indexer job that exists under the following roles:

XP Scaled (XP1)
xConnect Collection
xConnect Collection Search

An example of a connection string for a configured replica set with least privilege users:

```
mongodb://sa:12345@10.45.111.102:57017,10.45.111.102:57018,10.45.111.102:57019/collection?
replicaSet=testReplicaSet&amp;retryWrites=true
```

An example of a connection string for a configured sharded cluster with least privilege users:

```
mongodb://sa:12345@127.0.0.1:27017/collection?retryWrites=true
```

4. Rebuild the xDB search indexes in Solr. See the section *Rebuild the Search Indexes* in the link.

> **NOTE**
> After you switch the data provider from SQL to Mongo, you can delete the SQL collection database.

6.3.2. Mongo DB high availability

You must configure the MongoDB replica sets and retryable writes features to ensure high availability.

Replication is a group of MongoDB instances that are configured into a single replica set that maintains the same data set for automatic failover and node recovery.

> **NOTE**
> We recommended that you configure replication and retryable writes in your production environment.

The minimum replication configuration is a replica set with two members that hold the data and an arbiter.

We also recommend that you change the write concern option of the default replica set to a number greater than *1*. The write concern option specifies the number of replica set nodes for request acknowledgement.

Whereas replication ensures data availability, the current operation must be successfully completed during failover. You configure this in the connection string with the retry Writes option.

> **NOTE**
> Retryable writes require a replica set or sharded cluster and do not support standalone instances.

Retryable writes allow the MongoDB driver to retry a write operation if there is a network problem or if the primary node is not healthy. Retryable reads are not supported by the MongoDB driver and were implemented as part of the xConnect MongoDB provider.

The combination of these features provides high availability.

6.3.3. MongoDB sharded cluster configuration

Sharding is a method for scaling databases that distributes data across multiple machines.

> **NOTE**
> MongoDB uses sharding to support deployments with very large data sets and a high level of throughput operations.

MongoDB uses shard keys to partition data by collection. The shard key consists of an immutable field or fields that exist in every document in the target collection.

The collections and their required shard keys are:

Collection	Shard key
Contact	{_id: 1}
ContactFacets	{_id: 1}
Interactions	{_id: 1}
InteractionFacets	{_id: 1}
DeviceProfiles	{_id: 1}
DevideProfileFacets	{_id: 1}
ContactIdentifierIndex	{_id:'hashed'}
Changes	{_id:'hashed'}

6.3.4. Security

> **IMPORTANT**
> To protect your MongoDB installation, follow the MongoDB security checklist.

We recommend that you create least privilege users who can access MongoDB.

The following table contains a list of the actions that are available to least privilege users:

Privilege Sections	Privilege Actions
Query and Write Actions	find; insert; remove; update
Database Management Actions	createIndex
Deployment Management Actions	-
Replication Actions	-
Sharding Actions	-
Server Administration Actions	-
Session Actions	-
Diagnostic Actions	listIndexes; listCollections

6.4. Configure high availability for xConnect

> **IMPORTANT**
> If you are deploying a PaaS solution or if you are not going to configure a High Availability feature, skip this section.

High Availability is an xConnect feature that is based on Always On availability groups and configurable retryers.

6.4.1. Configure Always On availability groups

A Windows Server Failover Cluster (WSFC) is required when you deploy Always On availability groups.

You must also configure Synchronous-commit mode with automatic failover.

For more information about how to configure availability groups, see Microsoft's documentation.

To provide client connectivity to the database for a given availability group, you must create an availability group listener.

The availability group listener allows a client to connect to an availability replica without knowing the name of the physical instance of SQL Server that it is connecting to. You *must* specify the DNS name of the listener in the connection string instead of the server name. This helps your solution to automatically switch to the primary replica during a failover.

Here is an example of a connection string for an availability group listener:

```
Data Source=tcp:SMMListener,5025;Initial Catalog=ShardMapManagerDb;User Id=sa;Password=12345;
```

6.4.2. Configure the Collection database

The *Collection* database is designed to manage high read/write activity, and therefore supports sharding. The catalog of all the shards is located in the *Shard Map Manager* database. You must manually update the *[ShardsGlobal]* table in the catalog so that it can use listeners instead of server names.

6.5. Populate the managed schema for the Solr search provider

If you are using Solr or SolrCloud as your search provider, you must populate the managed schema.

To populate the managed schema:

1. On a Sitecore CM or Index Manager instance, open the the Sitecore Control Panel and then click **Populate Solr Managed Schema**.
2. Select the indexes and click **Populate**.

> **IMPORTANT**
> When you use SolrCloud, we recommend that you populate the indexes one at a time.

6.6. Rebuild the search indexes and the Link database

After you install Sitecore Experience Platform, you must rebuild the search indexes and rebuild the *Link* databases.

To rebuild all the indexes:

1. On a Sitecore CM or Index Manager instance, open the Sitecore Launchpad, click **Control Panel**, and in the **Indexing** section, click **Indexing manager**.
2. In the **Indexing Manager** dialog box, click **Select all**, and then click **Rebuild**.

To rebuild the Link databases for the *Master* and *Core* databases:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Database** section, click **Rebuild Link Databases**.
2. Select the *Master* and *Core* databases and then click **Rebuild**.

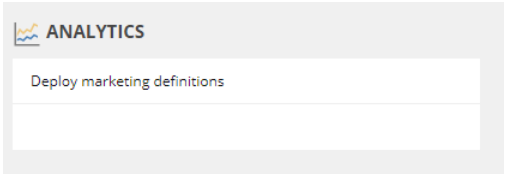
6.7. Deploy the marketing definitions

If you want to use the Sitecore Experience Marketing functionality, you must deploy the marketing definitions.

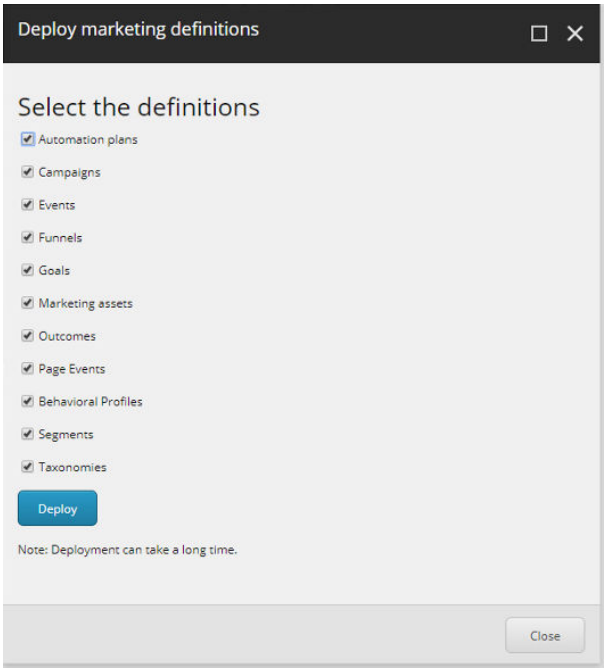
> **NOTE**
> If you do not plan to use the Sitecore Experience Database (xDB), you do not have to perform these steps.

To deploy the marketing definitions:

1. On a Sitecore CM instance, in the **Sitecore Launchpad**, click **Control Panel**, **Analytics**, and then click **Deploy Marketing Definitions**.



2. In the **Deploy marketing definitions** dialog box, select all the definitions and taxonomies and click **Deploy**.



6.8. Content expiration

Microsoft IIS uses the Expire Web content header (located in common HTTP Response Headers) to determine whether to return a new version of the requested web page if the request is made after the web page content has expired. IIS marks each web page before it is sent, using the settings that you provide for content expiration. The website visitor's browser translates the expiration mark. You can set the IIS Expire Web content header to improve performance.

By setting Expire Web content to something other than *immediately*, you can reduce second-access load times by 50 –70%. This setting does not affect dynamically-generated content.

For more information about content expiration, see Client Cache in the Microsoft IIS documentation.

6.9. Configure geo-location lookup

Geo-location lookup enables you to identify contacts and track their activity on your website based on their IP address.

In certain cases, you might not want to track Geo IP data, or due to restrictions in certain legal jurisdictions, you might not be allowed or want to store IP addresses. In these situations, you can configure tracking.

> **NOTE**
> This procedure is optional.

To configure tracking:

1. If you do not want to track Geo IP data, in the `Website\App_Config\Sitecore\Marketing.Tracking\Sitecore.Analytics.Tracking.config` file, set the `Analytics.PerformLookup` setting to *False*.

   > **NOTE**
   > The default value of the `Analytics.PerformLookup` setting is *True* and you must not change it in a single-instance environment.

2. If you do not want to store IP addresses in the xDB, in the `\App_Config\Sitecore\Marketing.Tracking\Sitecore.Analytics.Tracking.config` file, update the `RedactIpAddress` setting.
   This change will hash the IP addresses before they are stored in the xDB.

   > **IMPORTANT**
   > To ensure that hashing is secure, in the `Sitecore.Analytics.Tracking.config` file, in the `geoIpManager` section, change the default salt value.

3. Restart IIS.

6.10. Configuring session state providers

In the Sitecore Experience Database, you can use a session state server to share all your contact sessions across different browsers and devices. Configuring session state is particularly important if you have deployed a multi-server, fully scalable environment with clusters of content delivery or processing servers.

Sitecore is deployed with an *InProc* session state provider by default but we recommend that you use *OutOfProc* session state providers if you deploy more than one CD server.

Sitecore Experience Platform supports the following session state providers:

- Microsoft SQL Server
- Redis

These providers support the `SessionEnd` event that the xDB needs to track website visits.

To configure any *OutOfProc* session state providers, see the Sitecore documentation.

6.11. Warm up the servers

To ensure that your Sitecore websites are available at all times, even after restarting a server, you must enable the IIS auto-start feature for the application pools on all the servers that you have configured.

For more information about the auto-start feature, see Microsoft's documentation.

6.12. Security hardening

We recommend that you follow all the security hardening instructions described in our documentation. In addition, the way you implement your Sitecore solution has a significant effect on the security of your website and it might require additional security-related coding and configuration.

For more information about security hardening, see the Security Guide.

6.13. Configure Email Experience Manager

To use EXM you must configure the delivery process.

For more information about EXM and about configuring the delivery process, see the EXM documentation.

6.14. Synchronize the time source

When you configure the Sitecore Experience Database (xDB), you must synchronize all the Windows servers in your solution to a single reliable time source, for example, by means of the Network Time Protocol (NTP).

The aggregation of engagement automation states depends on the system time, and changing this can lead to incorrect aggregation results or loss of data.

6.15. Import the client translations

The user interface in Sitecore Experience Platform are only available in English by default. If you want to use another language, you must import the client translation. Download the latest translations from the Client Translations section of the release page.

Client translations are available in:

- Danish (da-DK)
- German (de-DE)
- Japanese (ja-JP)
- Chinese (zh-CN)

For more information about how to import a client translation, see the Sitecore Experience Platform Client Translations.

7. Uninstall the Sitecore XP Scaled topology

This chapter describes how to uninstall Sitecore XP Scaled topology and contains the following sections:

- Uninstall a Sitecore instance using SIF
- Uninstall the XP Scaled topology in a distributed environment

> **NOTE**
> If Sitecore Experience Commerce is installed in your environment, uninstall it before uninstalling Sitecore XP Scaled.

7.1. Uninstall a Sitecore instance using SIF

You can use SIF to uninstall a Sitecore instance/role from a local server.

To uninstall a Sitecore instance:

1. Launch PowerShell as an administrator.
2. Run the `Uninstall-SitecoreConfiguration` cmdlet, and specify the path to your SIF configuration file.
   For example, using the `sitecore-XP1-cm.json` file:

   ```
   Uninstall-SitecoreConfiguration -Path <configurationpath>\sitecore-XP1-cm.json
   ```

   Alternatively, you can pass in the parameters declared in the SIF configuration files by prefixing their name with a dash " -" in the command line.
   For example:

   ```
   Uninstall-SitecoreConfiguration -Path <configurationpath>\sitecore-XP1-cm.json -
   SqlDbPrefix SC.
   ```

   In a PowerShell command line, you can pass additional parameters to control the uninstall process.
   For example, running the `Verbose` cmdlet increases the amount of information that is logged, and the `-Skip <taskname>` cmdlet skips one or more tasks.

To correctly uninstall a SIF configuration, you must pass the same parameters that were used during the installation.

The uninstall is performed by a separate list of tasks within the configuration file. For more information, see the SIF documentation.

7.2. Uninstall the XP Scaled topology in a distributed environment

Centralized uninstallation of distributed deployments is not supported yet. If you need to re-install a scaled deployment you must remove everything that was installed during the previous installation from each individual computer.

8. Appendix

This chapter contains answers to some issues that can arise during the installation as well as some supplementary instructions that help you configure your environment, such as, file permissions, performance counters, and certificates.

This chapter contains the following sections:

8.1. Common issues

I get a 403.16 Forbidden error

- Check that your root certificate is in the Trusted Root Certificates Authority store of the *Local Computer*, not the current user and that the *Issued To* and *Issued By* properties of the root certificate match.
- Ensure you imported the certificates into the *Local Computer's* certificate store, not the current user's certificate store.
- Ensure the certificate you created in IIS has a name that matches the site.
  For example, `CM_test`.
- Ensure you pasted your thumbprint into a PowerShell command line window, and that you removed the hidden character at the start of the string.
- Ensure your thumbprint is in uppercase letters.
  For example: `3D703B5198D6D3CEE1D0C1B1BC9ECB6D34989BA4`.
  You can find the thumbprint in the following locations:
  - `Sitecore\App_Config\ConnectionStrings.config`
  - `XConnect\App_Config\ConnectionStrings.config`
  - `XConnect\App_Data\jobs\continuous\AutomationEngine\App_Config\ConnectionStrings.config`
- Ensure the self-signed certificate you created in IIS has the same name as your xConnect instance.
- Ensure the client authentication certificate (under the local machine's *Personal* store) has *read* permissions for the *IIS_IUSR* group and the *NETWORK SERVICE* group.

My Solr index is empty

In the `\xConnect\App_data\jobs\continuous\IndexWorker\App_data\Logs` folder, check the indexer's log files.

If you have an error that says your remote certificate is invalid according to the validation procedure, ensure that the indexer's `ConnectionStrings.config` file is using `localhost` rather than `127.0.0.1` for the Solr core URL.

Microsoft.SqlServer.TransactSql.ScriptDom.dll is not installed

If this happens, you can see the following error: *The SQL provider cannot run with dacpac option because of a missing dependency.* To resolve this issue, see this Knowledge Base article.

My Sitecore installation failed while I was using Skype

If you use Skype while you are installing Sitecore Experience Platform, it is possible that your xConnect installation may fail. This occurs because Skype and Sitecore xConnect both use port 443, which interferes with the installation.

If this happens, change your Skype configuration as described in Microsoft's documentation.

Failed installations

If an installation fails for any reason, you must clean up the partial installation before attempting another installation.

To clean up a partial installation, run the configurations again with the `Uninstall-SitecoreConfiguration` command with the same parameters as you used for the installation.

The `createcert.json` configuration file does not contain uninstallation tasks that remove certificates because it is highly likely that the certificates, particularly the root certificates, are used elsewhere.

To remove an incorrect certificate:

1. To open the **Certificate Management** console, in the Windows command prompt, enter `certlm.msc` and press **Enter**.
   To open the **Certificate Management** console for the Current User, enter `certmgr.msc`.
2. In the left pane, in the tree, expand the *Personal* node and select **Certificates**.
3. Select the incorrect certificate, right click it and then click **Delete**.
4. To remove the Root certificates, in the console, in the left hand pane, expand *Certificates*, *Trusted Root Certification Authorities*, *Certificates* and select the incorrect certificate, right click it and then click **Delete**.

After you have removed the failed installation, correct the errors in the launch script or configuration files before attempting a new installation.

The Indexing Manager shows no results after rebuilding the indexes

This can happen if the Solr managed schema is not populated properly during deployment.

To solve this problem, re-populate the Solr schema.

My Solr Schema is not populated

To populate the Solr schema:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Indexing** section, click **Populate Solr Managed Schema**.
2. In the **Schema Populate** dialog box, click **Select all**, and then click **Populate**.

8.2. Access rights

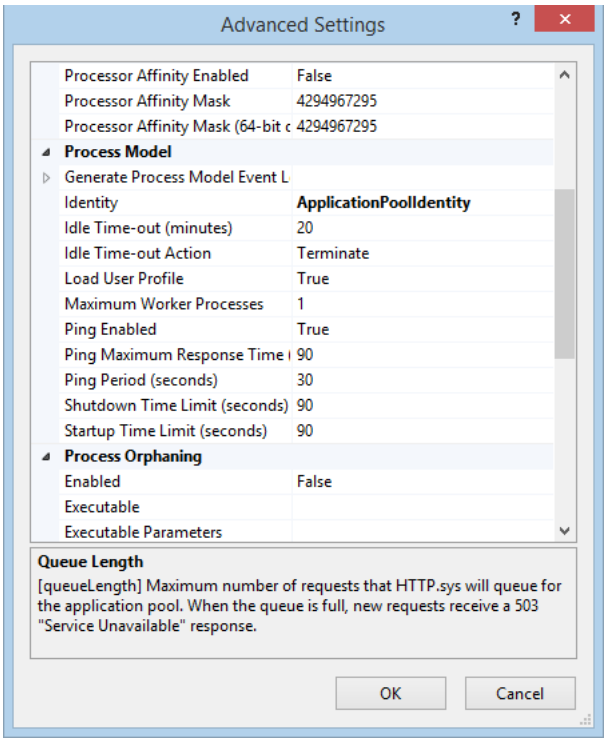
8.2.1. Use Windows Authentication with SQL Server

You can configure Sitecore to use Windows Authentication for a SQL connection and remove the user name and password from the `connectionStrings.config` file.

> **NOTE**
> This only applies to the *Core*, *Master*, *Web*, and *Reporting* SQL databases, and not to xDB and xConnect.
>
> xDB and xConnect only support Certificate Authentication as described in the chapter *Set up a production environment*.

To configure Sitecore to use Windows Authentication:

1. In Windows, launch the IIS Manager.
2. Select the application pool that Sitecore is running under, click **Advanced Settings** and in the **Identity** field, set the identity to the domain user.



3. In SQL Server, register the domain user and grant the appropriate security permissions to the Sitecore databases for the domain user.
4. On the computer that hosts Sitecore Experience Platform, add the domain user to the *IIS_IUSRS* group.
   For more information about changing the permissions for the *IIS_IUSRS* group, see the section Server file system requirements.
5. In a text editor, edit the `\App_Config\ConnectionStrings.config` file and replace the user id and password parameters with `trusted_connection=Yes`.

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
<add name="core" connectionString="Data
Source=.\sql2016;Database=sc9_Core;Trusted_Connection=True"
/>
<add name="master" connectionString="Data
Source=.\sql2016;Database=Sandbox6_Master;Trusted_Connection=True"
/>
<add name="web" connectionString="Data
Source=.\sql2016;Database=Sandbox6_Web;Trusted_Connection=True"
/>
<add name="reporting" connectionString="Data Source=<Data-
Source>;Database=Sandbox6_Analytics;Trusted_Connection=True"
/>
</connectionStrings>
```

> **NOTE**
> If you use the Sitecore Experience Database (xDB), the configuration is the same for the *Reporting* database.

6. Prepare your identity so that it can be used as a service account with the `aspnet_regiis.exe` file and the `-ga` switch.

8.2.2. Use Windows performance counters

Sitecore Experience Platform contains built-in functionality that reads and updates the Windows performance counters that you can use to monitor and troubleshoot the Sitecore application. This functionality requires access to Windows registry keys. You can grant access by making the application pool identity a member of the built-in *Performance Monitor Users* group.

For more information, see Microsoft's documentation about Application Pool Identity.

If the required registry permissions are not granted, whenever the application attempts to access Windows performance counters, the *Access to the registry key 'Global'* is denied error is registered in the Sitecore log files.

To avoid this error, you must prevent Sitecore from updating the performance counters.

To prevent Sitecore from updating the performance counters:

- In a text editor, open the `\App_Config\Sitecore.config` file and set the `Counters.Enabled` setting to *false*.

8.3. Certificates

8.3.1. Client certificates

When you have installed Sitecore XP, you can see the thumbprint values of the `XConnectCert` parameters in the following connection strings in the `\App_config` folder:

- `ConnectionStrings.config` for Sitecore and xConnect roles:

```
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindValue
```

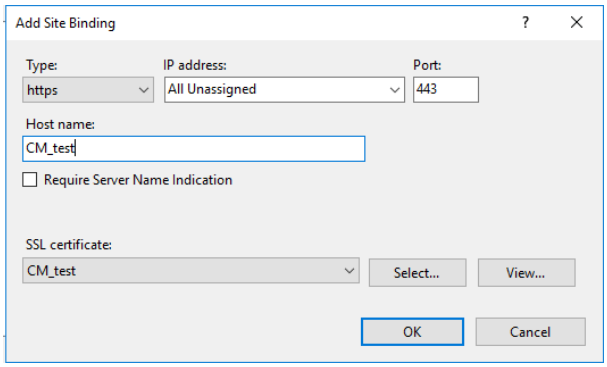
```
=738F45F610221990DA2FE059E1D8C2ECCB5067F2"
/>
```

- `AppSettings.config` file for the xConnect roles:

```
<add key="validateCertificateThumbprint" value="738F45F610221990DA2FE059E1D8C2ECCB5067F2" />
```

8.3.2. Server certificates

After you install Sitecore XP, your IIS site will have an HTTPS binding and an associated SSL certificate with the same name. For example, if the site is named `CM_test` , the HTTPS binding and associated SSL certificate are also named `CM_test`.



Configure a new client certificate

If your client certificate has expired, you must configure Sitecore to use a new client certificate.

To configure Sitecore to use a new client certificate:

1. Install the new client certificate on every computer on which you have installed the xConnect client and ensure that the authority that issued the certificate is in the *Trusted Authorities* list. For more information about the appropriate role and the certificate that you must install, see the section *Set up the certificates*.
2. To grant the appropriate permissions to the certificate, open the Microsoft Management Console, click **File**, and then click **Add/Remove Snap-in**.
3. In the **Add or Remove Snap-ins** dialog box, in the **Available snap-ins** field, select **Certificates** and then click **Add**.
4. In the **Certificates snap-in** dialog box, select **Computer account** and then click **Next**.
5. In the **Select Computer** dialog box, select **Local computer** and then click **Finish**.
6. In the **Add or Remove Snap-ins** dialog box, click **OK**.
7. In the **Console** window, in the left-hand pane, navigate to the *Certificates (Local Computer)/ Personal/Certificates* folder.
8. In the center pane, right-click the new certificate, click **All Tasks**, **Manage Private Keys**.

9. In the **Permissions** dialog box, add the accounts that you want to grant permissions to, based on the following criteria:
   - For virtual accounts that were created for each Sitecore application pool identity, add for example:
     IIS AppPool\<AppPoolName> – for virtual accounts.
     NETWORK SERVICE account – only if the Sitecore website application pools run under the NetworkService identity.
     LOCAL SERVICE account – the Marketing Automation Engine runs under this account.
   - For virtual accounts that were created for the xConnect application pool identity for the website hosting the *xDB Automation Operations* role, add for example:
     IIS AppPool\<AppPoolName> – for virtual accounts.

10. In the `\App_Config\connectionstrings.config` file, in the appropriate connection strings, replace the old thumbprint parameter value with the new client certificate thumbprint. For more information about the appropriate roles and the role certificate thumbprints that must be updated, see the section *Set Up Client Certificates*.
11. Update the thumbprint values in all of the certificate connection strings on the following Sitecore instances:

XP Scaled (XP1)
Content Delivery
Content Management
Processing
Marketing Automation Engine
xDB Automation Operations

For example, on each XP Single (XP0) Sitecore instance, update the thumbprint value in these connection strings:

```
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindVal
ue=859E88DC0692BA1583145223C455F186937C0D62"
/>

<add name="xdb.referencedata.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindVal
ue=859E88DC0692BA1583145223C455F186937C0D62"
/>

<add name="xdb.marketingautomation.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindVal
ue=859E88DC0692BA1583145223C455F186937C0D62"
/>

<add name="sitecore.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindVal
ue=859E88DC0692BA1583145223C455F186937C0D62"
/>

<add name="xdb.marketingautomation.operations.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindVal
ue=859E88DC0692BA1583145223C455F186937C0D62"
/>
```

12. In the xConnect root folder, in the `\App_Config\AppSettings.config` file, update the thumbprint value in the following setting:

```
<add key="validateCertificateThumbprint"
value="859E88DC0692BA1583145223C455F186937C0D62" />
```

You must update this setting on the following servers:

XP Scaled (XP1)
xConnect Collection
XConnect Search
xDB Reference Data
xDB Automation Reporting
xDB Automation Operations
Sitecore Cortex Processing Engine
Sitecore Cortex Reporting Service

13. Restart IIS on every computer that you configured to use a new client certificate.

8.3.3. Configure Sitecore XP to use new server certificates

To configure Sitecore XP to use new server certificates:

1. Replace all of the old server certificates with new server certificates on each server with the XM Scaled (XM1) role.
2. Replace all of the old server certificates with new server certificates on each server with the XP Scaled (XP1) role.

> **NOTE**
> The common name field (CN) must be the same as your instance name.

3. On each IIS instance, in the **Site Bindings** window, select the new server certificate.
4. Restart IIS on every computer that you configured to use the new server certificates.

8.4. Install and configure Microsoft Machine Learning Server

You can use the Sitecore Cortex™ Processing Engine with or without Microsoft Machine Learning Server (MLS). By default, the Cortex Processing Engine does not contain workers that use MLS.

You can install Microsoft Machine Learning Server before or after you install Sitecore XP.

To use Machine Learning Server:

1. Install Machine Learning Server.

2. Navigate to `C:\Program Files\Microsoft\ML Server\R_SERVER\bin\x64\Rgui.exe` and run RGui as an administrator.
3. In RGui, run the following commands:

```
lib <- tail(.libPaths(), n=1)
repo <- 'https://cran.microsoft.com/snapshot/2017-07-04';
install.packages('openssl', lib, repos=repo)
install.packages('curl', lib, repos=repo)
install.packages('httr', lib, repos=repo)
```

4. To configure MLS to operationalize analytics on a single machine, set up the *web* and *compute* nodes.
   For more information about configuring MLS, see Microsoft's documentation.
5. Configure the HTTPS protocol for the *web* node.
   For more information about how to configure HTTPS, see Microsoft's documentation.
6. Make a note of the *web* node (port 12800 by default), username, and password.
   For example: `https://admin:Secret123!@localhost:12800/`
   If you want to use special characters in the `processing.engine.mrs` connection string, the characters must be encoded.
   For example, `Secret123#` becomes `Secret123%23`:

```
<add name="processing.engine.mrs" connectionString="http://
admin:Secret123%23@localhost:12800/"/>
```