

## Keyfactor Command 11.3

### Server Installation Guide

# Table of Contents

<b>1.0 Introduction</b>	<b>1</b>
<b>2.0 Installing Servers</b>	<b>2</b>
2.1 Logical Architecture	3
2.2 Physical Architecture	6
2.3 Solution Design	8
2.4 Keyfactor Command Server	8
2.4.1 System Requirements	9
2.4.2 Planning & Preparing	11
2.4.2.1 Selecting an Identity Provider for Keyfactor Command	11
2.4.2.2 SQL Server	50
2.4.2.3 Certificate Authorities	63
2.4.2.4 Keyfactor Command Server(s)	65
2.4.2.5 Create Service Accounts for Keyfactor Command	66
2.4.2.6 Create Groups to Control Access to Keyfactor Command Features	72
2.4.2.7 Configure Certificate Chain Trusts for CAs	73
2.4.2.8 Hostname Identification and Resolution	74
2.4.2.9 Firewall Considerations	75
2.4.2.10 Acquire a Public Key Certificate for the Keyfactor Command Server	77
2.4.2.11 Install IIS and .NET on the Keyfactor Command Server	78
2.4.2.12 Configure SSL for the Default Web Site on the Keyfactor Command Server	84
2.4.2.13 Configure the Keyfactor Command Server to Require SSL	84
2.4.2.14 Prepare for External Log Shipping over TLS (Optional)	85
2.4.3 Installing	90
2.4.3.1 Install the Keyfactor Command Components on the Keyfactor Command Server(s)	90
2.4.3.2 Install the Keyfactor Command Server from the Command Line	124
2.4.4 Initial Configuration	131
2.4.4.1 Configure Kerberos Authentication	132
2.4.4.2 Configure Logging	139
2.4.4.3 Configure CA Certificate Synchronization	140
2.4.4.4 Create or Identify Certificate Templates for Enrollment	149
2.4.4.5 Configure Renewal Handler Permission	150
2.4.4.6 Create a Certificate Template for Mac Auto-Enrollment	152
2.5 Keyfactor CA Policy Module	153
2.5.1 System Requirements	154
2.5.2 Preparing for the Keyfactor CA Policy Module	154
2.5.3 Installing the Keyfactor CA Policy Module Handlers	155
2.5.3.1 Install the Keyfactor RFC 2818 Policy Handler	159
2.5.3.2 Install the Keyfactor SAN Attribute Policy Handler	164
2.5.3.3 Install the Keyfactor Whitelist Policy Handler	170
2.5.4 Configure Logging for the Keyfactor CA Policy Module	177
2.5.5 Add Non-Keyfactor SCEP Servers to the Ignore List	179
2.6 Appendices	179
2.6.1 Appendix - Troubleshooting Logi Log Files	179
2.6.2 Appendix - Logi Load Balancing: Keyfactor Command Configuration Wizard Setup	181
2.6.3 Appendix - Configuration Wizard Errors in the Logs	183
<b>3.0 Glossary</b>	<b>184</b>
<b>4.0 Copyright Notice</b>	<b>194</b>

## List of Tables

Table 1: System Requirements	9
Table 2: Keyfactor Identity Provider Container Parameters	23
Table 3: Typical Service Accounts	71
Table 4: Protocols Keyfactor Command Uses for Communication	76
Table 5: .NET Framework Release Values	78
Table 6: Available components for Keyfactor.	92
Table 7: Identity Provider Parameters	101
Table 8: Features Required for Each Server Role	126
Table 9: Input Parameters XML File Fields	128
Table 10: ConfigurationWizardConsole.exe Options	131
Table 11: Microsoft CA Permission Matrix	144

# List of Figures

Figure 1: Keyfactor Command Logical Architecture Diagram	3
Figure 2: Keyfactor Command Physical Architecture Diagram	7
Figure 3: Simple Keyfactor Command Solution Design	8
Figure 4: Select the Download Hosting Bundle Option Under Run Server Apps	10
Figure 5: Export the SQL Server Certificate as a P7B	14
Figure 6: Add a SQL Authentication Login	17
Figure 7: Add a SQL Database	18
Figure 8: Select a Realm in the Keyfactor Identity Provider Administration Console	26
Figure 9: Select Command-OIDC-Client in the Keyfactor Identity Provider Administration Console	26
Figure 10: Set the Client Access Settings	27
Figure 11: OpenID Endpoint Configuration Link	28
Figure 12: OpenID Endpoint Configuration Settings	29
Figure 13: SSO Session Values	29
Figure 14: Access Token Lifespan	30
Figure 15: Add a Keyfactor Identity Provider User	31
Figure 16: Locate the Keyfactor Identity Provider User's ID	31
Figure 17: Set a Password for the Keyfactor Identity Provider User	32
Figure 18: Select a Realm in the Keyfactor Identity Provider Administration Console	33
Figure 19: Add a Keyfactor Identity Provider Role	33
Figure 20: Add a Keyfactor Identity Provider Group	34
Figure 21: Assign a Role to a Group in Keyfactor Identity Provider	34
Figure 22: Select a Realm in the Keyfactor Identity Provider Administration Console	35
Figure 23: Add a Keyfactor Identity Provider User	36
Figure 24: Set a Password for the Keyfactor Identity Provider User	37
Figure 25: Assign a Role to a Keyfactor Identity Provider User	38
Figure 26: Select a Realm in the Keyfactor Identity Provider Administration Console	39
Figure 27: Add a Keyfactor Identity Provider Service Account (Client): General	39
Figure 28: Add a Keyfactor Identity Provider Service Account (Client): Capabilities	40
Figure 29: Copy the Keyfactor Identity Provider Service Account (Client) Secret	40
Figure 30: Login Page with Choice of Federated Identity Provider	41
Figure 31: Federated Identity Provider Login Flow	42
Figure 32: Client ID and Secret in Okta OIDC Application	43
Figure 33: Redirect URIs for the Okta OIDC Application	44
Figure 34: Create an Authorization Server Role Claim	45
Figure 35: Select a Realm in the Keyfactor Identity Provider Administration Console	46
Figure 36: Give the Keyfactor Identity Provider Identity Provider an Alias	46
Figure 37: Enter the Okta Discovery Endpoint in the Keyfactor Identity Provider Identity Provider	47
Figure 38: Enable PKCE in the Keyfactor Identity Provider Identity Provider	47
Figure 39: Add Okta Client ID and Secret in the Keyfactor Identity Provider Identity Provider	47
Figure 40: Deliver the Okta openid and profile to the Keyfactor Identity Provider Identity Provider	48
Figure 41: Map the Okta preferred_username to the Keyfactor Identity Provider Identity Provider Username	49
Figure 42: Map the Okta Roles to the Keyfactor Identity Provider Identity Provider Roles	50
Figure 43: SQL Server Configuration Manager View Active SSL Certificate	55
Figure 44: Registry View Active SSL Certificate	56
Figure 45: View SQL Server Services	56
Figure 46: SQL Server SSL Certificate Details	57
Figure 47: Grant Private Key Permissions for SQL Server	58
Figure 48: Default SQL Connection Strings	59
Figure 49: SQL Connection Strings with Encrypt Channel Disabled	59
Figure 50: SQL Connection Strings with MultiSubnetFailover Option Enabled	60
Figure 51: Certificate Template with Key Encipherment Key Usage	61
Figure 52: EJBCA Protocols for Keyfactor Command Connectivity	64
Figure 53: Local Security Policy	68



Figure 54: Install CA Chain Certificates on the Keyfactor Command Server	74
Figure 55: Use Get-WindowsFeature to Determine if All Required Roles and Features are Installed	80
Figure 56: Web Server Role	80
Figure 57: .NET 4.7 Feature	81
Figure 58: Role Services Page One	82
Figure 59: Role Services Page Two	83
Figure 60: Active Directory Module for Windows PowerShell	83
Figure 61: Install: Begin Setup Wizard	91
Figure 62: Install: Select Components	92
Figure 63: Windows Authentication	93
Figure 64: SQL Authentication	93
Figure 65: Configure: Backup Database Master Key	95
Figure 66: Configure: Upload License	95
Figure 67: Configure: Open Data File	96
Figure 68: Configure: Application Pools	97
Figure 69: Configure: Identity Providers—OAuth Claims Proxy Section	99
Figure 70: Configure: Identity Providers—OAuth Identity Provider Section	101
Figure 71: Configure: Encryption Warning	110
Figure 72: Configure: Database	110
Figure 73: Configure: Service	111
Figure 74: Configure: Email	112
Figure 75: Configure: Keyfactor Portal	114
Figure 76: Configure: Administrative Users for Active Directory	116
Figure 77: Configure: Administrative Users for OAuth	117
Figure 78: Configure: Dashboard and Reports	118
Figure 79: Configure: Orchestrators with Standard Authentication	119
Figure 80: Configure: Orchestrators with Client Certificate Authentication	120
Figure 81: Configure: API	121
Figure 82: Configure: Audit	122
Figure 83: Configure: Configuration Warnings	123
Figure 84: Configure: Save Configuration as a File	123
Figure 85: Configure: Configuration Operations	124
Figure 86: Configure: Configuration Complete	124
Figure 87: Configure Local Intranet Zone in Internet Properties	133
Figure 88: Configure Kerberos Constrained Delegation on the Keyfactor Command Machine Account	136
Figure 89: Add HOST and rpcss Service Types for Kerberos Constrained Delegation	137
Figure 90: Configure Kerberos Constrained Delegation on the Keyfactor Command Service Account	138
Figure 91: Certificate Profile for EJBCA Client Certificate	141
Figure 92: Certificate Download for EJBCA Client Certificate	141
Figure 93: Microsoft CA Permissions	143
Figure 94: EJBCA Access Permissions	147
Figure 95: Add Client Certificate as Member of EJBCA Access Rule	147
Figure 96: Keyfactor Command Service	148
Figure 97: Include Expired and Revoked Certificates in Certificate Search	149
Figure 98: Configure Expiration Renewal Handler: Add New Identity	151
Figure 99: Configure Expiration Renewal Handler: Assign Role to Identity	152
Figure 100: Keyfactor CA Policy Module Policy Module Handler Ordering	157
Figure 101: Default Policy Module	158
Figure 102: Install RFC 2818 Policy Handler: Begin Setup Wizard	159
Figure 103: Install RFC 2818 Policy Handler: Select Components	160
Figure 104: Enable the Keyfactor CA Policy Module	161
Figure 105: Upload the Keyfactor CA Policy Module License	162
Figure 106: Enable the RFC 2818 Policy Handler	163
Figure 107: Add Templates for Management with the RFC 2818 Policy Handler	164
Figure 108: Install SAN Attribute Policy Handler: Begin Setup Wizard	165
Figure 109: Install SAN Attribute Policy Handler: Select Components	166
Figure 110: Enable the Keyfactor CA Policy Module	167

Figure 111: Upload the Keyfactor CA Policy Module License	168
Figure 112: Enable the SAN Attribute Policy Handler	169
Figure 113: Add Templates for Management with the SAN Attribute Policy Handler	170
Figure 114: Install Whitelist Policy Handler: Begin Setup Wizard	171
Figure 115: Install Whitelist Policy Handler: Select Components	172
Figure 116: Enable the Keyfactor CA Policy Module	173
Figure 117: Upload the Keyfactor CA Policy Module License	174
Figure 118: Enable the Whitelist Policy Handler	175
Figure 119: Add Templates for Management with the Whitelist Policy Handler	176
Figure 120: Add Machines for Management with the Whitelist Policy Handler	177
Figure 121: Keyfactor CA Policy Module NLog.config File	179
Figure 122: Logi web.config	180
Figure 123: Logi Configuration Settings—Keyfactor Command Portal Tab	181
Figure 124: Logi Configuration Settings—Keyfactor Command Dashboards and Reports Tab	182

# 1.0 Introduction

The *Keyfactor Command Documentation Suite* includes:

- *Keyfactor Command Reference Guide*
- *Keyfactor API Reference Guide*
- *Keyfactor Command Server Installation Guide*
- *Keyfactor Orchestrators Installation and Configuration Guide*
- *Keyfactor Command Release Notes & Upgrading*

In addition, Keyfactor offers documentation for products that are not part of the *Keyfactor Command Documentation Suite*, including the *Keyfactor Command Upgrade Overview* and installation guides for third-party CA gateways that interface with Keyfactor, which are available upon request.

## 2.0 Installing Servers

The Keyfactor Command solution by Keyfactor allows you to issue and manage certificates across enterprise infrastructures to allow you to achieve end-to-end visibility, control, and automation across all your machine identities so you can turn the impossible into the possible. It includes a web-based Management Portal running on a SQL backend providing the command and control center for managing certificates in the enterprise.

Keyfactor Command provides:

- **Visibility**  
Identify risks and prevent outages more effectively with a complete and continuous inventory of all your cryptographic assets.
- **Control**  
Have ultimate flexibility to make all certificates trusted, compliant, and up-to-date—and keep them that way.
- **Automation**  
Replace manual, error-prone tasks with automated key and certificate discovery, management, and renewal.
- **Orchestration**  
Move from DevOps to DevSecOps by orchestrating and expanding cryptography to secure software delivery pipelines.

In addition to the Management Portal, Keyfactor also offers:

- Several agents and orchestrators for managing certificates in certificates stores via the Management Portal (see *Installing Orchestrators* in the *Keyfactor Orchestrators Installation and Configuration Guide*).
- Several certificate authority gateways to support management of and enrollment for certificates from remote and cloud-based certificate providers via the Management Portal.
- The Keyfactor API that integrates with the product to provide for customization (see *Keyfactor API Reference* in the *Keyfactor API Reference Guide*).
- A certificate authority policy module with several policy handlers to provide policy control at the Microsoft CA level (see [Keyfactor CA Policy Module on page 153](#)).
- An SSH Key Manager that extends beyond certificate management and traditional PKI to give security and network teams a simple, centralized solution to discover and manage SSH keys across their server and cloud infrastructure (see *SSH* in the *Keyfactor Command Reference Guide*).

Uniquely designed for PKI administrators to operate an enterprise PKI, it's never been easier to issue, revoke, renew, or replace a digital certificate. With exceptionally robust reporting and management capabilities for all the certificates in an IT environment, the PKI administrator has a truly scalable and entirely secure system for operating an enterprise PKI.

## 2.1 Logical Architecture

Keyfactor Command is an n-tier application, consisting of a web/presentation layer, application tier, and database tier. In addition, Keyfactor Command optionally includes a number of enrollment and management components to help facilitate secure and/or automated certificate issuance and delivery to various server and client platforms. The following sections provide views of the Keyfactor Command architecture from a logical and physical standpoint.

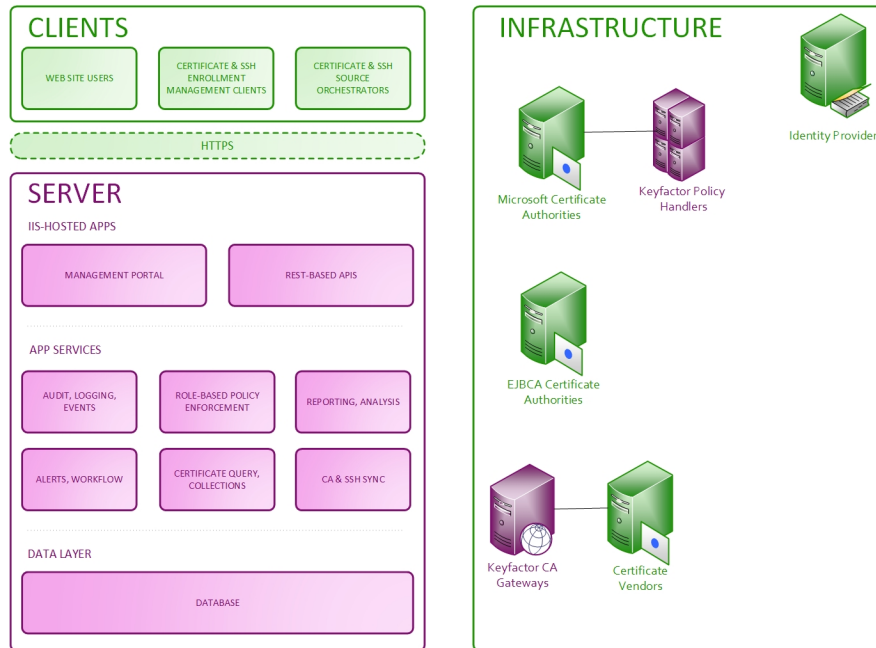


Figure 1: Keyfactor Command Logical Architecture Diagram

The Keyfactor Command solution includes the following logical components:

- Client / Orchestrator Tier:
  - Certificate Enrollment and Management Tools—While many certificate management functions can be performed in a completely agentless fashion, Keyfactor Command provides a number of enrollment and management tools to enable enhanced functionality where needed.
  - Certificate Source Orchestrators (aka Agents) and Gateways—Keyfactor Command gathers information about an enterprise's certificates and SSH keys from a number of different sources, including Microsoft and EJBCA CA databases, SSL scans, SSH key scans, API-based import, Java keystores, PEM certificate stores, F5 devices, NetScaler devices, Amazon Web Services (AWS) locations, select certificate vendor certificates via gateways, and manual import through the Keyfactor Command Management Portal.
- Web Tier:
  - Management Portal—Keyfactor Command includes a web-based Management Portal that provides a PKI operations dashboard for administrators. It also enables certificate officers to

easily search for and locate certificates and then perform management functions on them such as revocation or recovery. In addition, Keyfactor Command allows every certificate to be tagged with additional customer-defined metadata about the certificate, such as points of contact, certificate/app owners, etc. From within the Management Portal, administrators can inventory and manage secure shell (SSH) keys across the enterprise, while users can issue new SSH keys.

- Enrollment Web Pages—Keyfactor Command includes issuance capabilities to a wide array of platforms, including Mac auto-enrollment, PKCS#12-based certificate issuance, and web-based CSR submission for administrator enrollment. PKCS#12 (PFX) and CSR enrollment are supported against Microsoft CAs in the local forest, remote CAs—with or without trust relationships—and non-domain-joined Microsoft and EJBCA CAs.
- Web APIs—Keyfactor Command is implemented with a robust and continually growing set of APIs that allow integration of Keyfactor Command functionality with the set of Keyfactor Command clients and orchestrators, as well as third-party or customer-created software or scripts.
- App Tier:
  - Event History and Audit Logging—Keyfactor Command maintains a record of operations that are performed on a certificate and the individual who performed the operation. This includes information such as initial synchronization date, additions to and removals from certificate stores, certificate recovery, and certificate revocation.
  - Role-Based Policy Enforcement—Keyfactor Command offers a rich, role-based permissions model that allows you to create your own roles as needed within the Keyfactor Command Management Portal. Users can be assigned to roles based upon Active Directory group memberships or individually, and then each role can be assigned granular Keyfactor Command permissions such as report creation, certificate revocation or renewal, or metadata update.
  - Dashboard and Report Engine—Keyfactor Command contains a dynamic dashboard along with several built-in reports generated using the Logi Analytics Platform.
  - Certificate Query & Collections—Keyfactor Command allows certificate administrators to query the certificate database using various search criteria. In addition, the bulk of Keyfactor Command's reporting and automated notification functionality can be driven through certificate collections, which are a user-definable mechanism that allows organizations to report on groups of certificates based on selection criteria.
  - Workflow Builder—The workflow builder in Keyfactor Command allows you to easily automate event-driven tasks when a certificate is requested or revoked. The workflows can be configured with multiple steps between the start and end of the operation that offer a simple way to configure notifications, approvals, and end-to-end automation. This provides for operational agility in an intuitive and easy-to-configure manner. The workflow builder is highly customizable with options to execute PowerShell scripts, invoke REST requests, send email messages, and require one or more approvals built in, and facilities to build custom steps to allow many more functions to be built as needed.
  - Alert Notice Generator—Keyfactor Command allows you to configure customized email notifications for impending certificate expiration, revocation expiration, pending certificate requests, issued certificate requests and denied certificate requests. These notifications

can be sent at configurable intervals, and may contain ASCII or HTML content, along with relevant information about the certificate or request in question (e.g. subject DN, issuer, thumbprint, template, custom metadata, etc.)

- Certificate Request Alerting—Keyfactor Command provides interfaces through which administrators can request certificates that require CA-level manager approval, interfaces where the approvers can either issue or deny the certificate request, and interfaces where the requesters can then download the certificates. This, along with the notice generator, provides an end-to-end flow for certificate requests that require CA-level manager approval.
- Alert Handlers—In addition to the notice generator that provides email alerts for SSH key and certificate expiration and enrollment workflow, Keyfactor Command also provides optional handlers that can be used in the certificate request and expiration alerts to output the information to the event log rather than sending it via email, run a PowerShell script, or automatically renew expiring certificates that are found in certificate store locations.
- Keyfactor Command Service—The Keyfactor Command Service (a.k.a. the timer service) is designed to continually keep the Keyfactor Command SQL database synchronized with the contents of every configured Microsoft and EJBCA CA database in the organization as well as external certificates located on servers it can scan. The service can perform full or partial scans of different CAs at user-defined intervals. This enables a rapidly-accessible, easily queried mirror of CA database information that can then be put to use via Keyfactor Command. Synchronization of CA information is supported for Microsoft CAs joined to the local forest, remote domain-joined Microsoft CAs—with or without trust relationships—and non-domain-joined Microsoft and EJBCA CAs. The Keyfactor Command Service is also responsible for executing a variety of periodic tasks, including scheduled reports, alerts and cleanup jobs.
- Data Tier:
  - SQL Database—Keyfactor leverages a Microsoft SQL Server database to store the information that Keyfactor Command uses.
- Microsoft Certification Authority Components:
  - RFC 2818 Policy Handler—The RFC 2818 Policy Handler integrates with the Microsoft CA to allow you to automate the addition of a DNS SAN matching the CN of the requested certificate for selected templates.
  - SAN Attribute Policy Handler—The SAN Attribute Policy Handler allows the addition of SANs not included in the CSR when making a CSR enrollment request. The added SANs will overwrite any existing SANs in the CSR. This functionality is the same as that seen with the Microsoft default policy module for the CA as a whole when the CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag is set except the SAN Attribute Policy Handler provides the ability to control SAN addition on a template-by-template basis without the need to enable the Microsoft CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag.
  - Whitelist Policy Handler—The Whitelist Policy Handler integrates with the Microsoft CA to allow you to restrict certificate enrollment on that CA for a configured certificate template or templates to only designated client machines. This allows you, for example, to force certificate enrollment for web server certificates to be accepted only via the Keyfactor Command

Management Portal and denied when coming from the Microsoft certificates MMC or IIS on the target servers for web server certificates.

- Enterprise Infrastructure:
  - Certification Authorities—Keyfactor Command has been built from the ground up to make it easier to operate organizational PKIs. This allows you to benefit from Keyfactor Command's extended features around Microsoft CA capabilities such as certificate templates, enrollment and recovery agents, and private key recovery. Keyfactor Command's integration with EJBCA provides support for capabilities such as certificate profiles, end entity profiles, enrollment, and revocation.
  - Identity Providers—Keyfactor Command relies on an identity provider to support authentication to the Management Portal and the Keyfactor API and supporting group memberships for Keyfactor Command role assignments. Historically, the product has been integrated with Microsoft Active Directory, using AD for Microsoft CA and certificate template enumeration and for the inclusion of AD account attributes in the content of issued certificates. With the release of Keyfactor Command version 11.0, support for identity providers expands beyond Active Directory. Users may choose between Active Directory and an open authorization (OAuth) 2.0 compliant identity provider with a complete implementation of the OpenID Connect (OIDC) protocol including the Keyfactor-provided Keyfactor Identity Provider.

## 2.2 Physical Architecture

[Figure 2: Keyfactor Command Physical Architecture Diagram](#) shows the physical architecture of the Keyfactor Command solution.



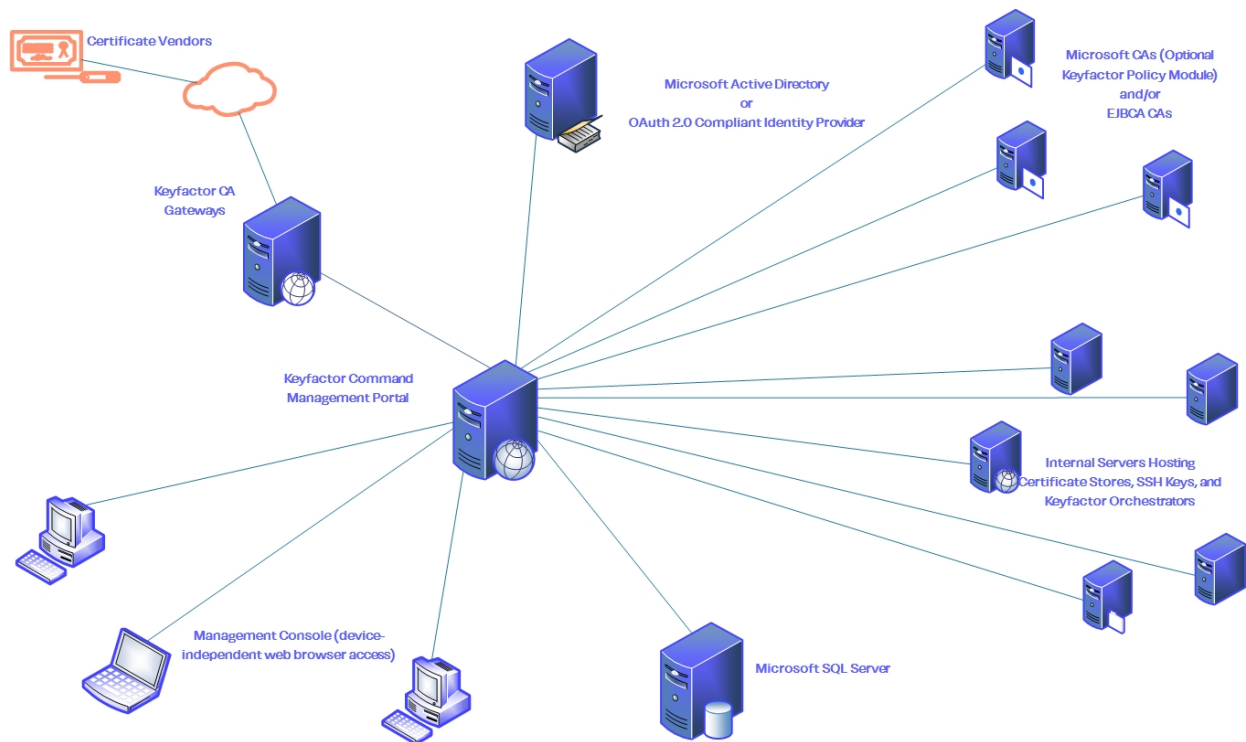


Figure 2: Keyfactor Command Physical Architecture Diagram

For simplicity, the servers in [Figure 2: Keyfactor Command Physical Architecture Diagram](#) are shown as single physical instances. In practice, these servers may be virtual machines and may be load balanced or clustered to meet availability or performance requirements. The diagram includes some optional components—including the Keyfactor vendor gateways and Keyfactor orchestrators—which are not covered in this guide. For more information about these components, see the *Keyfactor Orchestrators Installation and Configuration Guide* and the documentation for each of the gateways.

- Keyfactor Command-Dedicated Servers<sup>1</sup>:
  - Keyfactor Command Server—This server hosts the Keyfactor Command Management Portal, the Keyfactor Command Services roles, and the Logi Analytics Platform for report generation. These roles run as ASP.NET (4.5 or higher) applications on IIS. Both Windows Server 2019 and 2022 are supported.
- Enterprise-Shared Servers:
  - Microsoft SQL Server—Keyfactor Command supports Microsoft SQL Server 2017, 2019 and 2022 all with TLS encryption enabled for its primary database. While a dedicated SQL deployment is certainly an option, many organizations maintain a well-established SQL server farm to support multiple applications within the organization; if preferred, Keyfactor Command can easily make use of such a service. Keyfactor does not recommend locating the Keyfactor Command roles on the SQL server in a production deployment.

<sup>1</sup>The roles described in this section may be co-located on a single physical or virtual server or may be further separated to multiple machines.

- **Web Reverse Proxy**—If Internet-based access is required, the Keyfactor Command services can be published through a variety of reverse proxy products such as Microsoft UAG/TMG, F5, SiteMinder, or Citrix NetScaler.
- **Network-based Hardware Security Module (HSM not pictured)**—In certain configurations, Keyfactor Command requires the use of Enrollment Agent (EA) and/or Key Recovery Agent (KRA) certificates. To provide additional security over these certificates' private keys, Keyfactor strongly recommends the use of a Hardware Security Module (HSM) such as the Thales NetHSM if these features will be used.

## 2.3 Solution Design

Keyfactor Command supports a number of different deployment architectures to help provide for different needs from small and simple to highly available. The solution can be as simple as one Keyfactor Command server hosting all the Keyfactor Command roles (other than the policy handlers, which are installed on a Microsoft CA) or the roles can be separated onto different machines to provide increased security or distribute the load. Redundant servers can be added to provide for high availability—either within the same data center or across data centers. Keyfactor expects that the specifics of a high availability deployment plan would be finalized as part of the project rollout.

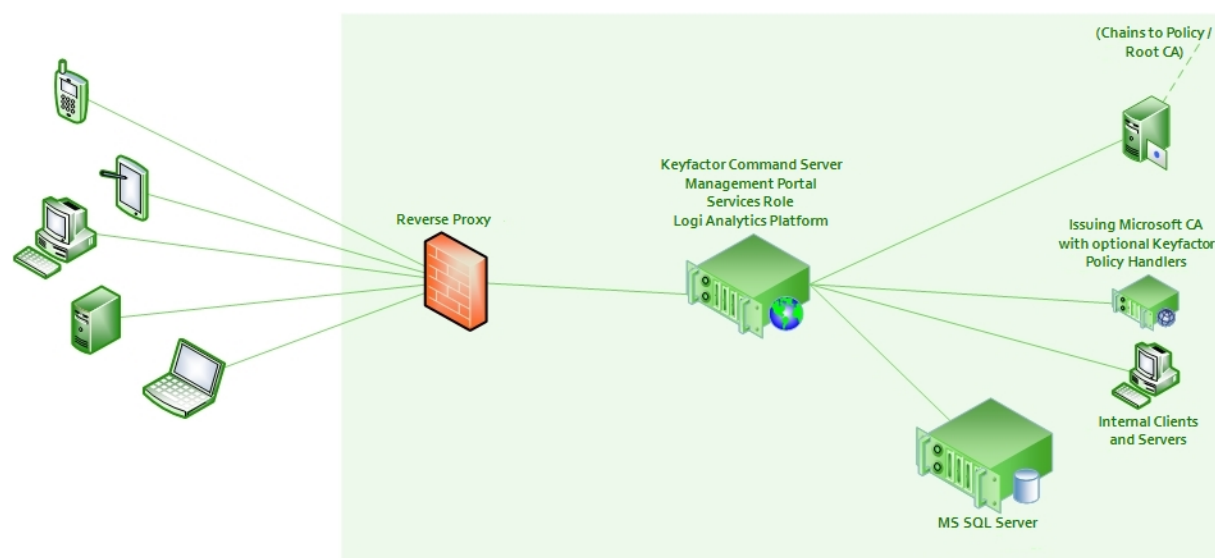


Figure 3: Simple Keyfactor Command Solution Design

## 2.4 Keyfactor Command Server

The Keyfactor Command solution by Keyfactor allows you to issue and manage certificates across enterprise infrastructures to allow you to achieve end-to-end visibility, control, and automation across all your machine identities so you can turn the impossible into the possible.

## 2.4.1 System Requirements

[Table 1: System Requirements](#) provides the recommendations for minimum system specifications used by Keyfactor Command components. All servers may be deployed as virtual machines and may be part of a clustering or load-balanced architecture, if desired. If the Keyfactor Command roles are co-located, the specifications may need to be scaled accordingly. All Microsoft-supported methods for making SQL Server highly available are supported. For most high availability requirements, Keyfactor recommends using always on availability groups (see [SQL Server on page 50](#)).



**Important:** SSH management in Keyfactor Command with the Keyfactor Bash Orchestrator (see *SSH* in the *Keyfactor Command Reference Guide*) is only supported when using Active Directory as an identity provider (see *Selecting an Identity Provider for Keyfactor Command* in the *Keyfactor Command Server Installation Guide*). The SSH option in the Management Portal will only appear when Keyfactor Command is installed using Active Directory as an identity provider (and with a license that supports SSH).

Table 1: System Requirements

Component	Minimum Requirements
Keyfactor Command Server (Management Portal, Keyfactor API, and Services roles)	<ul style="list-style-type: none"><li>Windows Server 2019 or 2022</li><li>Internet Information Services (IIS) with:<ul style="list-style-type: none"><li>Basic Authentication—if you plan to use Active Directory as an identity provider (see <a href="#">Selecting an Identity Provider for Keyfactor Command on page 11</a>)</li><li>Windows Authentication—if you plan to use Active Directory as an identity provider with Windows authentication (see <a href="#">Selecting an Identity Provider for Keyfactor Command on page 11</a>)</li><li>ASP.NET 4.7 or greater</li><li>The Active Directory Module for Windows PowerShell</li></ul>See <a href="#">Install IIS and .NET on the Keyfactor Command Server on page 78</a>.</li><li>ASP.NET Core Hosting Bundle version 6.0 (x64). Version 6.0 is available for download from Microsoft:<div><a href="https://dotnet.microsoft.com/download/dotnet/6.0/runtime">https://dotnet.microsoft.com/download/dotnet/6.0/runtime</a></div><p>You need the ASP.NET Core Hosting Bundle, not the .NET Runtime (x64) or the ASP.NET Core Runtime. At the above link, this would be the <b>Download Hosting Bundle</b> option under the <i>Run server apps</i> heading.</p></li></ul>

Component	Minimum Requirements
	<div data-bbox="626 281 972 491"> </div> <h2 data-bbox="626 604 945 653">Run server apps</h2> <p data-bbox="509 667 1062 787">Do you want to run web/server applications? The ASP.NET Core Hosting Bundle includes the .NET Runtime and ASP.NET Core Runtime. If installed on a machine with IIS, it'll also add the <a href="#">ASP.NET Core IIS Module</a>.</p> <div data-bbox="631 808 941 875"> <a href="#">Download Hosting Bundle</a> </div> <p data-bbox="503 896 1276 924"><i>Figure 4: Select the Download Hosting Bundle Option Under Run Server Apps</i></p> <p data-bbox="503 934 1385 997">You can use the following PowerShell command to check the .NET core version(s) installed on a server (if any):</p> <div data-bbox="571 1018 863 1045"> <pre>dotnet --list-runtimes</pre> </div> <p data-bbox="503 1092 1408 1186">Output from this command will look something like this if you have the correct 6.0 x64 version of the .NET Hosting Bundle installed (notice the path is in C:\Program Files, not C:\Program Files (x86), indicating this is the x64 version):</p> <div data-bbox="568 1207 1164 1270"> <pre>Microsoft.AspNetCore.App 6.0.21 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]</pre> </div> <div data-bbox="513 1312 1404 1480"> <p><b>Important:</b> The ASP.NET Core Hosting Bundle should not be installed before installing IIS. If the hosting bundle is installed before IIS is installed, the bundle will not function correctly after the IIS install and will require repair.</p> </div> <ul data-bbox="475 1501 1123 1610" style="list-style-type: none"> <li>• .NET Framework 4.7.2 or greater</li> <li>• 4 GB RAM, 2 GHz CPU, 40 GB disk</li> <li>• Keyfactor Command license key for the current release</li> </ul>
Microsoft SQL Database	<p data-bbox="464 1644 1276 1707">Ability to connect to a Microsoft SQL Server 2017, 2019, or 2022 all with TLS encryption enabled and compatibility level 130 or higher.</p> <p data-bbox="464 1717 1024 1745">8 GB RAM, 2+ GHz CPU (&gt;= 2 cores), 500 GB disk</p>

Component	Minimum Requirements
Browser to Access the Management Portal	<ul style="list-style-type: none"> <li>• Chrome: 99.0.4844.74+</li> <li>• Firefox: 98.0+</li> <li>• Microsoft Edge: 99.0.1150.30+</li> </ul>
EJBCA CA (Optional)	<ul style="list-style-type: none"> <li>• EJBCA Enterprise version 7.8.1 or later is supported.</li> <li>• The EJBCA REST API must be enabled to interoperate with Keyfactor Command (see System Configuration -&gt; Protocol Configuration in the EJBCA administration portal).</li> </ul>

## 2.4.2 Planning & Preparing

Before you install Keyfactor Command, you need to consider the components that make it up and its dependencies and decide where you want each role to reside, which roles—if any—you want to be highly available, and which features you’re going to enable. It’s possible to start with a non-redundant implementation and then add redundancy at a later time, but it’s best to plan ahead for this if it’s the desired goal.

Your license for Keyfactor Command may not include all the roles described in this document, so some sections of this guide may not apply to your implementation.

Once you’ve made these planning decisions, you then need to follow the steps outlined in this section that need to be taken prior to a Keyfactor Command implementation to install the prerequisites, create the required supporting components, and gather the necessary information to complete the Keyfactor Command installation and configuration process.

### 2.4.2.1 Selecting an Identity Provider for Keyfactor Command

Identity providers are used to provide a method for authenticating access to Keyfactor Command. Keyfactor Command supports Microsoft Active Directory and open authorization (OAuth) 2.0 compliant identity providers with a complete implementation of the OpenID Connect (OIDC) protocol. Keyfactor Command has been tested with the following identity providers:

- Active Directory

Microsoft’s Active Directory has historically been the only identity provider supported by Keyfactor Command. With Active Directory, you can authenticate users defined in the Active Directory forest to which the Keyfactor Command server is joined and users from forests in a trust with this forest using integrated Windows authentication. Users may alternatively be authenticated to Keyfactor Command using Basic authentication when you opt for Active Directory as your identity provider. Active Directory supports user, group and computer accounts.

- Keyfactor Identity Provider

Keyfactor Identity Provider is a lightweight application that is easily installed in the same environment as Keyfactor Command to provide standalone authentication separate from Active Directory. It may be used directly to supply authentication or it may be used to federate authentication to another OAuth 2.0 compliant identity provider (e.g. Okta, Ping Identity). Keyfactor Identity Provider runs in a Linux-based Docker container. Keyfactor Identity Provider supports users and groups.

- Auth0

Auth0 is a cloud-based OAuth 2.0 compliant identity and access management (IAM) solution owned by Okta.

A given Keyfactor Command server may be configured with only one identity provider. If desired, you may configure an environment with multiple Keyfactor Command servers and configure a different identity provider for each Keyfactor Command server.



**Important:** SSH management in Keyfactor Command with the Keyfactor Bash Orchestrator (see *SSH* in the *Keyfactor Command Reference Guide*) is only supported when using Active Directory as an identity provider. The SSH option in the Management Portal will only appear when Keyfactor Command is installed using Active Directory as an identity provider (and with a license that supports SSH).

## Installing Keyfactor Identity Provider

If you've opted to use Keyfactor Identity Provider as your identity provider for Keyfactor Command, you'll need to install and configure it before installing Keyfactor Command. Keyfactor Identity Provider runs in a Docker container on a Linux machine and is configured to use a Microsoft SQL database to store its data (configuration, user and group accounts, etc).



**Important:** Keyfactor highly recommends that you use strong passwords for any accounts or certificates related to Keyfactor Command and associated products, especially when these have elevated or administrative access. A strong password has at least 12 characters (more is better) and multiple character classes (lowercase letters, uppercase letters, numeral, and symbols). Ideally, each password would be randomly generated. Avoid password re-use.

### System Requirements

Keyfactor Identity Provider has the following requirements:

- Linux
- Docker
- Java version 12 or later
- Microsoft SQL Server

This guide assumes you are starting from a base of already having a Linux server with Docker installed. Instructions for installing and configuring Linux and Docker are beyond the scope of this guide. Some helpful web pages include:

- <https://ubuntu.com/server/docs/installation>
- <https://docs.docker.com/engine/install/ubuntu/>

## Preparing

To get ready to install Keyfactor Identity Provider you will need to gather a few pieces of information, copy some certificate files to your Linux machine, and set up a database and user in SQL.

## Prepare Certificates

Keyfactor Identity Provider uses these certificates:

- The certificate chain for the certificate used to secure the Microsoft SQL server that will host the database for Keyfactor Identity Provider to allow it to connect to the SQL server using an encrypted connection (see [Using SSL to Connect to SQL Server on page 54](#)).



**Note:** If you choose to connect to the SQL server using the SQL server's IP address, the SSL certificate on the SQL server will need to include the SQL server's IP address as a SAN.

- An SSL certificate with private key in the name of the server hosting the Docker container for Keyfactor Identity Provider to allow administrators to connect to the web-based administration interface for Keyfactor Identity Provider over an encrypted channel.

To prepare the SQL certificate chain:

1. On the Microsoft SQL server, open the certificates MMC for the local machine store using one of these methods:
  - Using the GUI:
    - a. Open an empty instance of the Microsoft Management Console (MMC).
    - b. Choose **File->Add/Remove Snap-in....**
    - c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.
    - d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.
    - e. Click **OK** to close the Add or Remove Snap-ins dialog.
  - Using the command line:
    - a. Open a command prompt using the "Run as administrator" option.

- b. Within the command prompt type the following to open the certificates MMC:

certlm.msc

2. Drill down to the Personal folder under Certificates for the Local Computer. Locate the certificate used to secure connections to your SQL server (see [Using SSL to Connect to SQL Server on page 54](#)).
3. Right-click the certificate and choose All Tasks->Export....
4. Follow the export wizard, choosing not to export the private key, choosing a format of P7B, and opting to including the chain certificates.

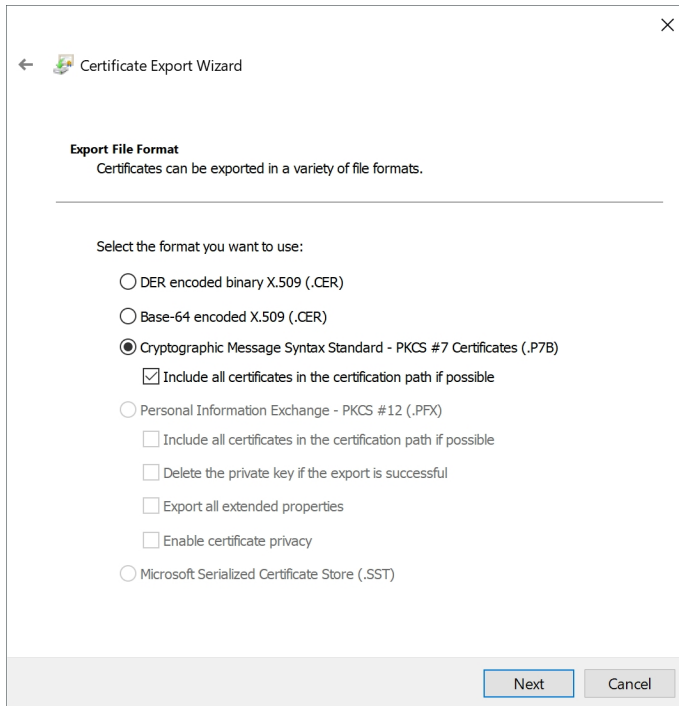


Figure 5: Export the SQL Server Certificate as a P7B

5. Copy the exported file to a working directory on the Docker host.
6. On the Docker host, use OpenSSL to extract the certificate and chain from the P7B file with a command similar to the following:

```
sudo openssl pkcs7 -inform der -in /my/path/mysqlcert.p7b -print_certs -out  
/my/path/mysqlchain.pem
```



**Tip:** See the next step for a script that completes this step and the following step.



- Split the resulting certificate file into separate files, one for each certificate. One approach to this is to use a script similar to the following (which also includes the preceding command):

```
#!/bin/bash

# Extract certificates and chain from the .p7b file
openssl pkcs7 -inform der -print_certs -in /my/path/mysqlcert.p7b -out mysqlchain.pem

# Split the chain certificates into separate files
csplit -s -z -f cert_ mysqlchain.pem '/subject=/' '{*}'

# Remove the SQL certificate file generated by csplit leaving only the chain
rm -f cert_00

# Rename the extracted certificates
for file in cert_*; do
    subject=$(openssl x509 -noout -subject -in "$file")
    filename=$(echo "$subject" | sed -n 's/.*CN = \([^,]*\).*\/\1/p')
    mv "$file" "$filename.crt"
done
```



**Tip:** To create this script, use your text editor (e.g. nano, vi) to open a new file, paste the entire contents of the script into the file, and save. Change the permissions on the file to be executable and execute. For example:

```
sudo chmod 755 extract.sh
```

```
./extract.sh
```

- Use the Java keytool command to create a Java truststore containing all of the certificates that are part of the chain required to trust the certificate used to secure connections to your SQL server. **Each certificate** (extracted as per the previous step) needs to be imported into the truststore separately using a command similar to the following:

```
sudo keytool -import -file /my/path/mychaincert1.crt -keystore /my/path/sql-truststore -alias mychaincert1 -trustcacerts -storepass "MySuperSecureStorePassword"
```

When prompted, review the certificate and answer “yes” to trust it as appropriate.



**Tip:** To check the contents of the Java Keystore after you've created it, use a Java keytool command similar to the following:

```
sudo keytool -list -keystore /my/path/sql-truststore -storepass  
"MySuperSecureStorePassword"
```



**Note:** If desired, you can leave the `-storepass` switch off each keytool command to be prompted to enter the password instead.

To prepare the SSL certificate:

1. Acquire the SSL certificate using the Fully Qualified Domain Name (FQDN) of the server or alias used for the Keyfactor Identity Provider Docker host. This is the name that you will use to access Keyfactor Identity Provider via a browser for management purposes and that Keyfactor Command will use to access Keyfactor Identity Provider for authentication purposes.
2. Copy the certificate together with its private key to a working directory on the Docker host.
3. Depending on the method you used to acquire your certificate, you may need to manipulate it on the Docker host to get it into the correct format. You need separate PEM-encoded unencrypted private key and certificate files. If your certificate is a PKCS#12 file, you can use OpenSSL commands similar to the following to extract the certificate and key:

Extract just the certificate, not any chain certificates or the key:

```
sudo openssl pkcs12 -in /my/path/mycert.pfx -clcerts -nokeys -out  
/my/path/mycert.cer
```

Extract just the key:

```
sudo openssl pkcs12 -in /my/path/mycert.pfx -nocerts -out /my/path/mycert_key.pem
```

Decrypt the key:

```
sudo openssl rsa -in /my/path/mycert_key.pem -out /my/path/mycert_key-plain.pem
```



**Important:** The decrypted key file should be handled carefully and stored securely. During the container deployment, the certificate and key files will be copied to:

```
/install/path/certificates/ssl
```

Permissions should be set on the key file in this location such that the service account running the Docker container has read permission on it and no other users have access of any kind. By default, Docker containers run as root, so the permissions would look like something like this:



```
-r----- 1 root root 1704 Jul 10 09:12 appsvr18keyexamplecom-key-plain.pem
```

Following the deployment, the key file in the working directory (not the directory listed above) should be removed.

## SQL Setup

Keyfactor Identity Provider uses Microsoft SQL Server with SQL authentication, not Windows authentication. Your SQL server must be configured to support mixed mode authentication in order to use the SQL authentication. The database for Keyfactor Identity Provider needs to be created in SQL before the deployment is done and appropriate permissions granted for the SQL user you will configure in Keyfactor Identity Provider to make the connection to SQL.

On your Microsoft SQL server:

1. Identify an existing SQL login using SQL authentication (not Windows authentication) or create a new login to be used for Keyfactor Identity Provider to authenticate to SQL.

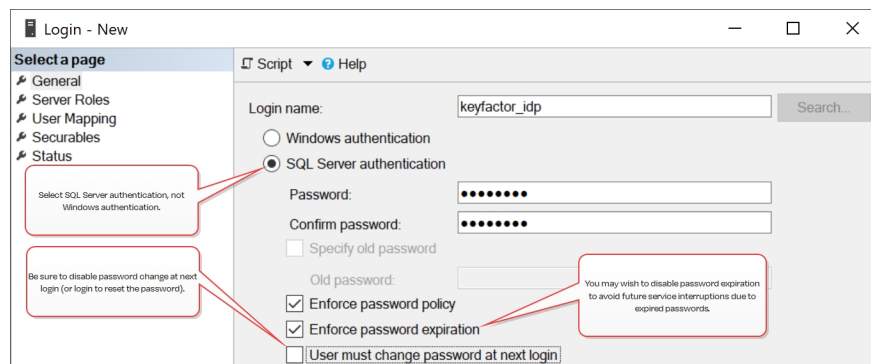


Figure 6: Add a SQL Authentication Login

2. Create a new database in SQL and grant the SQL login you created in the previous step at least dbo permissions on the database. You can do this either by setting it as the database owner while creating the database or by going back into the login after the database is created and granting the access on the User Mapping tab.

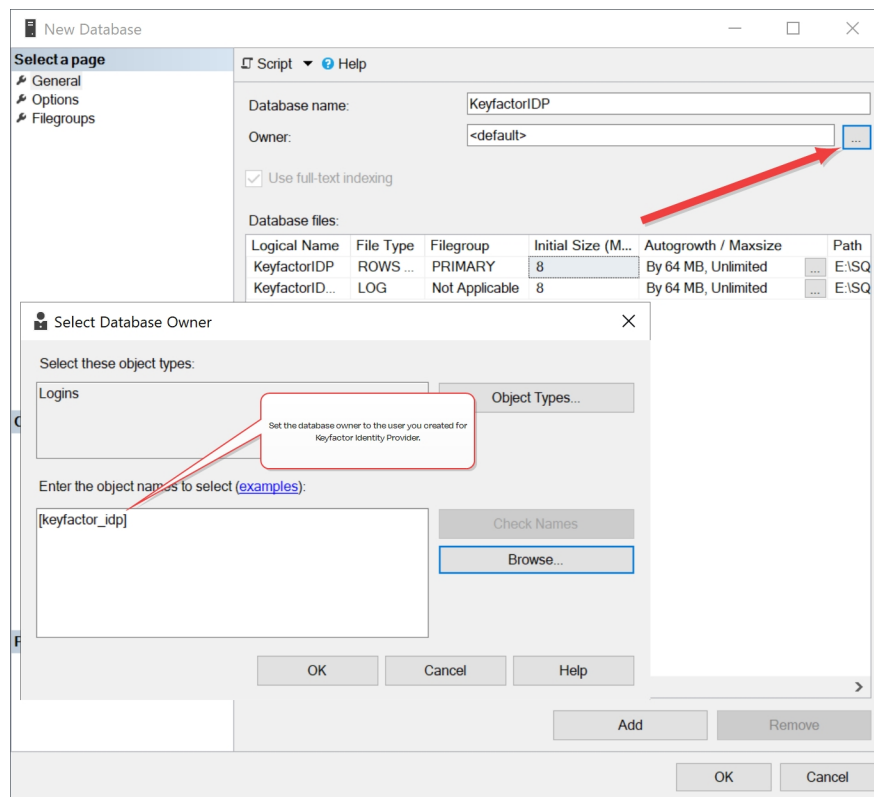


Figure 7: Add a SQL Database

## Gather Information

You will need the following information in order to appropriately configure the installation file for the Keyfactor Identity Provider container:

- A username and password for the initial administrative user that will be created in Keyfactor Identity Provider. By default, the username *admin* is used.
- The fully qualified domain name (FQDN) that you will use to access Keyfactor Identity Provider from a browser. This is typically the hostname of the container host.
- The FQDN or IP address of the SQL server hosting the database for Keyfactor Identity Provider. If you choose to use the IP address, the SSL certificate on the SQL server will need to include the SQL server's IP address as a SAN. If you configured your database in a non-default instance or your SQL is running on a non-standard port, you will need this information as well.
- The name of the database you created in SQL for Keyfactor Identity Provider.
- The username and password for the login you created in SQL for Keyfactor Identity Provider.
- The path to the *sql-truststore* Java keystore you created as per [Prepare Certificates on page 13](#).
- The paths to the server certificate (*mycert.cer*) and unencrypted private key (*mycert\_key-plain.pem*) you prepared as per [Prepare Certificates on page 13](#).

- The IP address of at least one DNS server in your environment that can be used to resolve the hostname of the SQL server if that server isn't publicly routable and you're not connecting to the SQL server via IP address.
- The path for the Keyfactor Identity Provider image you will install.



**Note:** The image artifactory will be available soon. For more information, check with your Keyfactor Client Success Manager or contact [support@keyfactor.com](mailto:support@keyfactor.com).

## Installing Using Docker Compose

The following section covers installing Keyfactor Identity Provider using a Docker compose file.

To install Keyfactor Identity Provider in a Linux container and start the container using Docker compose:

1. Create a directory from which you will run the Docker container. For example:

```
sudo mkdir /opt/kyfidp
```

2. Copy the *sql-truststore* Java truststore you created (see [Prepare Certificates on page 13](#)) into the directory you created for the Docker container and set the permissions appropriately. It needs to be readable by the user the Docker container will run as (by default root) and its group. For example:

```
sudo chown root:root sql-truststore
```

```
sudo chmod 440 sql-truststore
```

3. Copy the certificate and key for the Docker host (see [Prepare Certificates on page 13](#)) into the directory you created for the Docker container and set the permissions appropriately. They need to be readable by the user the Docker container will run as (by default root) and its group. For example:

```
sudo chown root:root appsrvr18keyexamplecom-server.cer
```

```
sudo chown root:root appsrvr18keyexamplecom-key-plain.pem
```

```
sudo chmod 440 appsrvr18keyexamplecom-server.cer
```

```
sudo chmod 440 appsrvr18keyexamplecom-key-plain.pem
```

4. From your Docker host, retrieve the Keyfactor Identity Provider image from the artifactory with commands similar to the following (placing the token or API key provided to you by Keyfactor in

the my\_password file):

```
sudo nano my_password.txt
```

```
cat my_password.txt | sudo docker login keyexample.jfrog.io --username username --password-stdin
```

```
sudo docker pull keyexample.jfrog.io/keyexample/command/auth-server:21.1.1
```



**Note:** The image artifactory will be available soon. For more information, check with your Keyfactor Client Success Manager or contact [support@keyfactor.com](mailto:support@keyfactor.com).



**Important:** Remove the my\_password.txt file when complete. For example:

```
sudo rm my_password.txt
```

5. Create a Docker compose file (*compose.yaml*) in the directory you created for the Docker container similar to the following, using inputs as per [Table 2: Keyfactor Identity Provider Container Parameters](#) and referencing the artifactory you pulled. The fields highlighted in red below indicate fields that need to be edited or that you may wish to edit.



**Important:** When editing the file, be sure to preserve the indenting exactly as found. YAML requires a very specific file layout to function. If the indenting (multiples of two spaces) or layout is incorrect, you will receive an error when trying to install.

```
services:
  auth:
    image: keyexample.jfrog.io/keyexample/command/auth-server:21.1.1 # Reference the correct arti-
    factory
    container_name: kyfidp # Give your container a name, if desired
    ports:
      - "1443:8443" # The first number is the port you will use to access the Keyfactor Identity
    Provider; do not change the second number
    environment:
      KC_HTTPS_CERTIFICATE_FILE: /etc/x509/https/tls.crt
      KC_HTTPS_CERTIFICATE_KEY_FILE: /etc/x509/https/tls.key
      KC_SPI_THEME_DEFAULT: Keyfactor-Keycloak-Theme

      KEYCLOAK_ADMIN: admin # The initial administrator user
      KEYCLOAK_ADMIN_PASSWORD: 'MySuperSecureAdminPassword' # The admin user's password needs
    quotes under some circumstances if it contains special characters
      CLIENT_SECRET: 'MySuperSecureKeyfactorIDPOIDCClientPassword' # FIPS compliant password used
```

to assign a unique and complex secret for the Command-0IDC-Client

```
KC_HOSTNAME: appsrvr18.keyexample.com # The FQDN of your Docker host

# This field is only required if you're using a port other than 443
KC_HOSTNAME_PORT: 1443

# This user must be dbo on KC_DB_URL_DATABASE
KC_DB_USERNAME: keyfactor_idp # The SQL username
KC_DB_PASSWORD: 'MySuperSecureSQLUserPassword' # The SQL user's password needs quotes under
some circumstances if it contains special characters
KC_DB_URL_HOST: sqlsrvr05.keyexample.com # The FQDN of your SQL server
KC_DB_URL_DATABASE: KeyfactorIDP # The database name created in SQL for the Keyfactor Identity Provider
KC_DB: mssql
KC_TRANSACTION_XA_ENABLED: false
# The SQL connection string in the following value contains the truststore name and password
KC_DB_URL_PROPERTIES:
';encrypt=true;trustServerCertificate=false;sendStringParametersAsUnicode=false;Integrated Security=False;Persist Security Info=True;trustStore=/temp/sql-truststore;trustStorePassword=MySuperSecureJKSStorePassword;'

# This value must be configured even if you do not have a reverse proxy
KC_PROXY: none

command:
- start --import-realm

# The first value in each of the following represents a location on your physical server,
which is being mapped to a path in the container represented by the second value.
volumes:
- ./mycert.cer:/etc/x509/https/tls.crt
- ./mycert_key-plain.pem:/etc/x509/https/tls.key
- ./sql-truststore:/temp/sql-truststore

# Optionally set the DNS server(s) for the Keyfactor Identity Provider server
dns:
- 192.168.12.2
- 192.168.12.3
restart: always
```

6. Set the permissions on the *compose.yaml* file such that the file is owned by root and readable only by root (this assumes your Docker daemon is running as root, which is typical). For example:

```
sudo chown root:root compose.yaml
```

```
sudo chmod 400 compose.yaml
```



**Tip:** If you need to make edits to the compose file, you will need to make the file writable again. For example:

```
sudo chmod 600 compose.yaml
```

7. Execute the following command to install and run the container in the foreground:

```
sudo docker compose up
```

You can instead run it in the background by adding the `-d` flag like so, but it can sometimes be helpful to run it in the foreground initially so that you can easily review the log output live:

```
sudo docker compose up -d
```



**Tip:** To stop and start the container again after installation is complete, use the following commands:

```
sudo docker compose stop
```

```
sudo docker compose start
```

Or:

```
sudo docker compose restart
```

If you need to delete the container and try the install again, use this command:

```
sudo docker compose down
```

This will not remove the configurations made in the SQL database.

To review logs generated from the container, identify the container ID or name with this command:

```
sudo docker container ls
```

For example, in the following output you could select either the container ID `0cf41b4c1ca4` or the name `kyfidp`:





CONTAINER ID	IMAGE	COMMAND
CREATED	STATUS	PORTS
NAMES		
b2abfe2b2b92	art.example1.com/condev-con/ejbca/ejbca-proxy:8.2.0	"/opt/key-factor/bin/..."
2 months ago	Up 2 months	8009/tcp, 8081-8082/tcp, :::80->8080/tcp, :::443->8443/tcp
ee461eb238ba	mariadb:latest	"docker-entry-point.s..."
2 months ago	Up 2 months	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp
0cf41b4c1ca4	art.example2.com/condev-exam/command/auth-server:latest	"/opt/kyfidp/bin/k..."
3 months ago	Up 3 months	8080/tcp, 0.0.0.0:5443->8443/tcp, :::5443->8443/tcp
	kyfidp	

Then use the following command to output the current log (with the optional `--follow` to make output continuous):

```
sudo docker container logs [--follow] [container ID or name]
```

Table 2: Keyfactor Identity Provider Container Parameters

Section	Parameter	Description
image		<b>Required*</b> . The path to the artifactory and image for the Keyfactor Identity Provider implementation.
container_name		A name to give to the container, if desired, for ease of reference.
environment	CLIENT_SECRET	<b>Required*</b> . Assign a complex and unique FIPS compliant secret to the Command-OIDC-Client created for Keyfactor Identity Provider. This is provided at database initialization and is also entered in Keyfactor Command to create a connection to Keyfactor Identity Provider, so be sure to retain this secret in a secure location.
environment	KC_HTTPS_CERTIFICATE_FILE	<b>Required</b> . The path and filename of the location within the container where the SSL certificate for the Docker host (see <a href="#">Prepare Certificates on page 13</a> ) will live.
environment	KC_HTTPS_CERTIFICATE_KEY_FILE	<b>Required</b> . The path and filename of the location within the container where the SSL certificate key for the Docker host (see <a href="#">Prepare Certificates on page 13</a> ) will live.
environment	KC_SPI_THEME_DEFAULT	<b>Required</b> . The theme for the Keyfactor Identity Provider implementation.

Section	Parameter	Description
environment	KEYCLOAK_ADMIN	<b>Required.</b> The username for the initial administrative user for Keyfactor Identity Provider. The default is <i>admin</i> .
environment	KEYCLOAK_ADMIN_PASSWORD	<b>Required*</b> . Set a secure password for the initial administrative user for Keyfactor Identity Provider.
environment	KC_HOSTNAME	<b>Required*</b> . The is the fully qualified domain name of the Docker host where you are deploying your container.
environment	KC_HOSTNAME_PORT	The port number you will use to access Keyfactor Identity Provider via a browser. This field only needs to be populated if you won't be using 443. If you'll be using 443, the entry should be commented out or removed.
environment	KC_DB	<b>Required*</b> . The type of SQL server. Only Microsoft SQL Server is supported (mssql).
environment	KC_DB_URL_HOST	<b>Required*</b> . The fully qualified domain name of the Microsoft SQL server that will host the database for Keyfactor Identity Provider.
environment	KC_DB_URL_DATABASE	<b>Required*</b> . The name of the Keyfactor Identity Provider database you pre-created in SQL per <a href="#">SQL Setup on page 17</a> .
environment	KC_DB_PASSWORD	<b>Required*</b> . The password for the SQL login to which you granted database ownership permissions on the Keyfactor Identity Provider database per <a href="#">SQL Setup on page 17</a> .
environment	KC_DB_USERNAME	<b>Required*</b> . The username for the SQL login to which you granted database ownership permissions on the Keyfactor Identity Provider database per <a href="#">SQL Setup on page 17</a> .
environment	KC_DB_URL_PROPERTIES	<b>Required*</b> . The SQL database connection string including the password you set to secure the Java keystore that holds the SQL server's certificate as per <a href="#">Prepare Certificates on page 13</a> .
environment	KC_TRANSACTION_XA_ENABLED	A Boolean that indicates whether the database for the installation supports XA transactions.
environment	KC_PROXY	The proxy address forwarding mode if the server is behind a reverse proxy.

Section	Parameter	Description
ports		The first number in the ports field indicates the port number you will use to access Keyfactor Identity Provider via a browser and that Keyfactor Command will use to access Keyfactor Identity Provider. If there are no other containers using this port on this Docker host, you may use 443. If 443 is already in use, you will need to change this to an alternate port (e.g. 1443). The second number in the ports field indicates the port number the Docker container uses internally. Do not change this number.
command		The command to start the container and import the realm JSON.
volumes		In this section you set the file names for the certificate and key files you copied into the directory for your Docker container. The volumes section sets mappings between files that exist on the host and locations in the running container. The first value represents a path and filename that exists on your Docker server. The second value represents a path and filename that will be created in the container containing the same information from the file referenced by the first value.
dns		In this section, add at least one IP for a DNS server that can be used to resolve hostname information for your SQL server from the Keyfactor Identity Provider container if the SQL server's address is not externally routable and you are not referencing the SQL server by IP address.

#### Configuring Keyfactor Identity Provider and Collecting Data for the Keyfactor Command Installation

Once the Keyfactor Identity Provider completes successfully, you should be able to open the administration console for it in a browser and gather the information you will need to complete the Keyfactor Command installation referencing it.

1. Use a browser to open the Keyfactor Identity Provider management interface. For example:

<https://appsrvr18.keyexample.com:1443>

Click the **Administration Console** link and sign in with the initial administrative user and password you defined with the KEYCLOAK\_ADMIN and KEYCLOAK\_ADMIN\_PASSWORD settings.



**Note:** Keyfactor Command communicates with Keyfactor Identity Provider over HTTPS, so be sure that you are working with Keyfactor Identity Provider over HTTPS to confirm that it is working correctly with no certificate errors.

2. In the Keyfactor Identity Provider Administration Console, select Keyfactor in the realm drop-down.

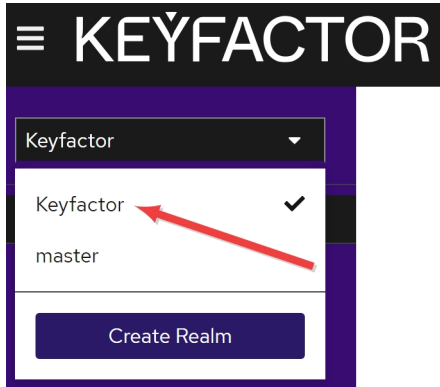


Figure 8: Select a Realm in the Keyfactor Identity Provider Administration Console

3. In the Keyfactor Identity Provider Administration Console, browse to *Clients > Client list* and click the *Command-OIDC-Client* client.

## Clients

Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list   Initial access token   Client registration				
<div><input type="text" value="Search for client"/> <input type="button" value="→"/> <input type="button" value="Create client"/> <input type="button" value="Import client"/></div> <div>1 - 7</div>				
Client ID	Name	Type	Description	Home URL
account	\${client_account}	OpenID Connect	—	<a href="https://appsrvr186.keyexample.com:3443/realms/Keyfactor/account/">https://appsrvr186.keyexample.com:3443/realms/Keyfactor/account/</a>
account-console	\${client_account-...}	OpenID Connect	—	<a href="https://appsrvr186.keyexample.com:3443/realms/Keyfactor/account/">https://appsrvr186.keyexample.com:3443/realms/Keyfactor/account/</a>
admin-cli	\${client_admin-cli}	OpenID Connect	—	—
broker	\${client_broker}	OpenID Connect	—	—
Command-OIDC-Client	Command-OIDC...	OpenID Connect	A seeded client applicatio...	—
realm-management	\${client_realm-m...}	OpenID Connect	—	—
security-admin-console	\${client_security-...}	OpenID Connect	—	<a href="https://appsrvr186.keyexample.com:3443/admin/Keyfactor/console/">https://appsrvr186.keyexample.com:3443/admin/Keyfactor/console/</a>

Figure 9: Select Command-OIDC-Client in the Keyfactor Identity Provider Administration Console

4. In the Client details on the Settings tab, populate the *Valid redirect URIs* and *Valid post logout redirect URIs* fields. The values for these fields are made up of the fully qualified domain name or alias you will use to access your Keyfactor Command server, the virtual directory name you will

use to access the Keyfactor Command Management Portal (KeyfactorPortal by default), a specific endpoint for the URI, and the name you will give to Keyfactor Identity Provider when configuring Keyfactor Command (see [Authentication Tab on page 97](#)). For example:

- Valid redirect URIs:

`https://keyfactor.kexample.com/KeyfactorPortal/callback/Command-OIDC`

- Valid post logout redirect URIs:

`https://keyfactor.kexample.com/KeyfactorPortal/signout-callback/Command-OIDC`



**Important:** Case matters for the virtual directory name—use *KeyfactorPortal* rather than *keyfactorportal* if you plan to accept the default virtual directory name.

### Access settings







Root URL ?	<input type="text"/>
Home URL ?	<input type="text"/>
Valid redirect URIs ?	<div><input type="text" value="https://keyfactor.kexample.com/KeyfactorPortal/callback/Command-OIDC"/> </div> <div> Add valid redirect URIs</div>
Valid post logout redirect URIs ?	<div><input type="text" value="https://keyfactor.kexample.com/KeyfactorPortal/signout-callback/Command-OIDC"/> </div> <div> Add valid post logout redirect URIs</div>
Web origins ?	<div><input type="text" value="/*"/> </div> <div> Add web origins</div>
Admin URL ?	<input type="text"/>

Figure 10: Set the Client Access Settings

5. In the Keyfactor Identity Provider Administration Console, browse to *Realm settings* and select the General tab. On the General tab, click the **OpenID Endpoint Configuration** link. This will open in a new browser window.

## Keyfactor

Enabled

Action ▾

Realm settings are settings that control the options for users, applications, roles, and groups in the current realm. [Learn more](#)

< General Login Email Themes Keys Events Localization Security defenses Sessions Tokens Clie >

Realm ID \* Keyfactor

Display name

HTML Display name

Frontend URL ②

Require SSL ② External requests ▾

ACR to LoA Mapping ②

No attributes have been defined yet. Click the below button to add attributes, key and value are required for a key pair.

+ Add an attribute

User-managed access ② ☐ Off

Endpoints ②

[OpenID Endpoint Configuration](#)

[SAML 2.0 Identity Provider Metadata](#)

Figure 11: OpenID Endpoint Configuration Link

6. In the browser window for the **OpenID Endpoint Configuration** link, review the settings. You may find it helpful to use a JSON formatting browser extension to make the data easier to read. The data you need from this configuration info is:

- Issuer (a.k.a. Authority)
- Authorization Endpoint
- Token Endpoint
- User Info Endpoint
- jwks\_uri (a.k.a. JSONWebKeySetUri)

Make note of these URLs, without the quotation marks. You will need them during the Keyfactor Command configuration.

```

    object {53}
      issuer : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor"
      authorization_endpoint : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor/protocol/openid-connect/auth"
      token_endpoint : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor/protocol/openid-connect/token"
      introspection_endpoint : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor/protocol/openid-connect/token/introspect"
      userinfo_endpoint : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor/protocol/openid-connect/userinfo"
      end_session_endpoint : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor/protocol/openid-connect/logout"
      frontchannel_logout_session_supported : true
      frontchannel_logout_supported : true
      jwks_uri : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor/protocol/openid-connect/certs"
      check_session_iframe : "https://appsrvr186.keyexample.com:3443/realms/Keyfactor/protocol/openid-connect/login-status-iframe.html"
  
```

Figure 12: OpenID Endpoint Configuration Settings

7. In the Keyfactor Identity Provider Administration Console, browse to *Realm settings* and select the Sessions tab.

On the Sessions tab, locate the *SSO Session Max* value. This value should match the *Session Expiration* parameter value configured in the Keyfactor Command configuration wizard on the Authentication tab. The *Session Expiration* value determines the length of time a browser session in the Keyfactor Command Management Portal will remain logged in before the user is prompted to re-authenticate regardless of whether the session is idle or in active use.

Locate the *SSO Session Idle* value and set it to a value that is appropriate to your environment. This value determines the length of time an idle browser session in the Keyfactor Command Management Portal will remain logged in before automatically logging out the user if no input from the user is received.

Keyfactor

Enabled

Action

Realm settings are settings that control the options for users, applications, roles, and groups in the current realm. [Learn more](#)

< Keys Events Localization Security defenses Sessions Tokens Client policies User registration >

SSO Session Settings

SSO Session Idle ?

30

Minutes

SSO Session Max ?

1

Hours

SSO Session Idle Remember Me ?

0

Minutes

SSO Session Max Remember Me ?

0

Minutes

The SSO Session Max value should match the Keyfactor Command Session Expiration value.

Figure 13: SSO Session Values

8. In the Keyfactor Identity Provider Administration Console, browse to *Realm settings* and select the Tokens tab. On the Tokens tab, locate the *Access Token Lifespan* value. This value should be greater than or equal to the *Cookie Expiration* parameter value configured in the Keyfactor Command configuration wizard on the Authentication tab

The *Cookie Expiration* value determines the length of time the authentication cookie for the Keyfactor Command Management Portal browser session is considered valid. After half of the setting's duration, Keyfactor Command will attempt to use a refresh token to update the cookie. If this fails, the user's session will be terminated. The cookie renewal is seamless from the user's perspective (there is no prompt for credentials).

#### Access tokens

Access Token Lifespan 5 Minutes


It is recommended for this value to be shorter than the SSO session idle timeout: 30 minutes


Access Token Lifespan For Implicit Flow 1 Hours

Client Login Timeout 1 Minutes

Figure 14: Access Token Lifespan

9. In the Keyfactor Identity Provider Administration Console, browse to *Users*. Click **Create new user** to add at least one new user to be granted administrative permissions in Keyfactor Command during the Keyfactor Command installation.

 **Note:** The admin user created during the Keyfactor Identity Provider installation can't be used for Keyfactor Command authentication because it is in the master realm, not the Keyfactor realm.

 **Tip:** Once the Keyfactor Command installation is complete, additional users and groups that you have added into Keyfactor Identity Provider can be added through the Keyfactor Command Management Portal and granted varying roles; only one user is required initially so a user can open the Management Portal at the conclusion of the installation.



## Create user

Required user actions ?

Username \*

Email

Email verified ? ☐ No

First name

Last name

Groups ?

Figure 15: Add a Keyfactor Identity Provider User

- Once the user account creation is complete, on the user details locate the ID for the user and make a copy of the GUID. This GUID is used to reference the user account when you configure the user as an administrator in the Keyfactor Command configuration wizard.

## jsmith

Details Attributes Credentials Role mapping Groups Consents Identity provider links Sessions

ID \*

Created at \*

Figure 16: Locate the Keyfactor Identity Provider User's ID

- In the user details on the Credentials tab, click **Set password** and set a password for the new user.

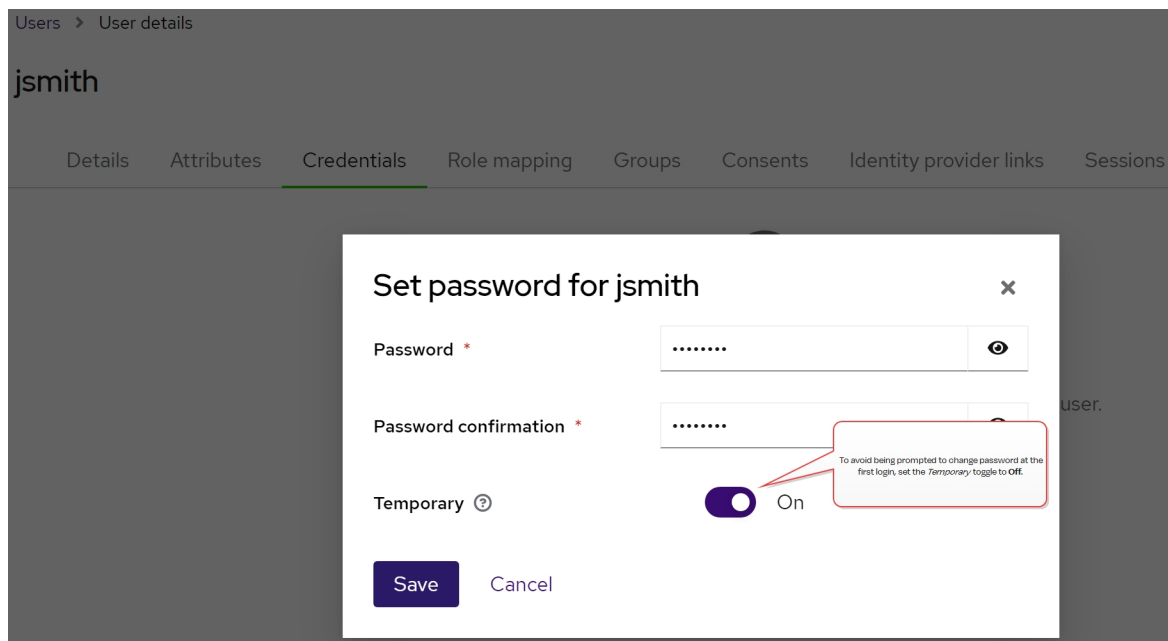


Figure 17: Set a Password for the Keyfactor Identity Provider User

From here you can create more administrative users, standard users, and groups (see [Using Keyfactor Identity Provider below](#)) or configure federation to an alternate OAuth provider (see [Federating from Keyfactor Identity Provider on page 40](#)).

## Using Keyfactor Identity Provider

Once you have finished configuring Keyfactor Identity Provider, you're ready to add roles, optional groups, users, and service accounts into it to be used for authentication to Keyfactor Command. Alternatively, you may choose to federate to an additional OAuth provider (see [Federating from Keyfactor Identity Provider on page 40](#)), in which case you don't need to add users in Keyfactor Identity Provider, but you will still need roles, optional groups, and service accounts, since it's the roles in Keyfactor Identity Provider that are used to create claims in Keyfactor Command to grant access to users holding these roles.



**Note:** You can grant access to Keyfactor Command on a user-by-user basis rather than with roles, but the management overhead of this method is much greater. Keyfactor recommends using roles.

## Roles and Groups

To add roles and groups in Keyfactor Identity Provider:

1. Use a browser to open the Keyfactor Identity Provider management interface. For example:

`https://appsrvr18.keyexample.com:1443`

Click the **Administration Console** link and sign in with an administrative user and password (see [Installing Using Docker Compose on page 19](#)).

2. In the Keyfactor Identity Provider Administration Console, select Keyfactor in the realm drop-down.

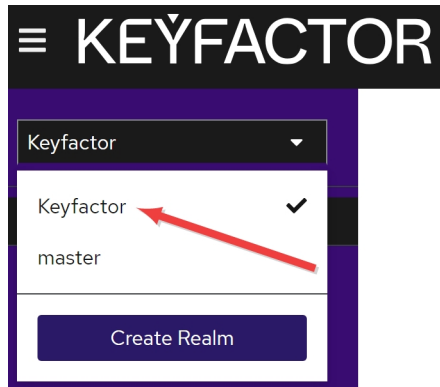


Figure 18: Select a Realm in the Keyfactor Identity Provider Administration Console

3. In the Keyfactor Identity Provider Administration Console, browse to *Realm roles*. Click **Create role** to add a new role to be used to grant permissions in Keyfactor Command. Enter a **Role name** and **Description**.



**Note:** The Role name is used when referencing the role from Keyfactor Command to create a claim and map it to a security role to grant permissions to users.

Realm roles > Create role

## Create role

Role name *	<input type="text" value="power-users-role"/>
Description	<input type="text" value="Keyfactor Command Power Users"/>
<div><input type="button" value="Save"/> <input type="button" value="Cancel"/></div>	

Figure 19: Add a Keyfactor Identity Provider Role

Repeat this step for each role that you will use from Keyfactor Command. For example, administrators, power users, and limited access users.

4. If desired, you can organize your roles into groups. This can simplify the process of assigning the roles to your users. To create a group, in the Keyfactor Identity Provider Administration Console, browse to *Groups*. Click **Create group** to add a new organizational group. Enter a **Name** for the group.

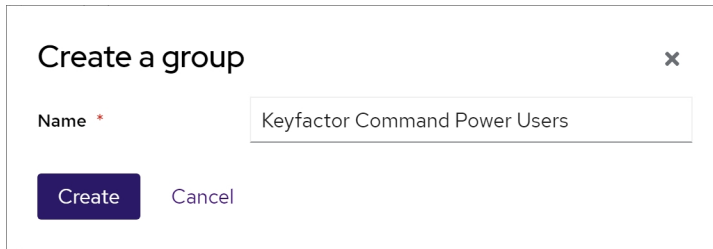


Figure 20: Add a Keyfactor Identity Provider Group

5. Once the group creation is complete, open the group details. In the group details on the Role mapping tab, click **Assign role** and select the role or roles to assign to this group.

Groups > Group details

### Keyfactor Command Power Users

Action ▼

Child groups Members Attributes Role mapping

→ ☒ Hide inherited roles **Assign role** Unassign 1-1 ▼ < >

<input type="checkbox"/>	Name	Inherited	Description	
<input type="checkbox"/>	power_users_role	False	Keyfactor Command Power Users	⋮

1-1 ▼ < >

Figure 21: Assign a Role to a Group in Keyfactor Identity Provider

Repeat these two steps for each group that you will use to manage roles in Keyfactor Identity Provider.

## Users

Be sure to create your roles and groups before adding your users.

To add users in Keyfactor Identity Provider:

1. Use a browser to open the Keyfactor Identity Provider management interface. For example:

<https://appsrvr18.keyexample.com:1443>

Click the **Administration Console** link and sign in with an administrative user and password (see [Installing Using Docker Compose on page 19](#)).

2. In the Keyfactor Identity Provider Administration Console, select Keyfactor in the realm drop-down.

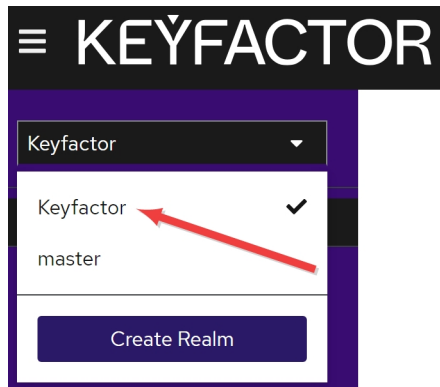


Figure 22: Select a Realm in the Keyfactor Identity Provider Administration Console

3. In the Keyfactor Identity Provider Administration Console, browse to *Users*. Click **Add user** to add a user. Enter at minimum a **Username**, and click **Join Groups**. In the *Select groups to join* dialog, select an appropriate group for this user and click **Join**.



**Tip:** By joining a group, your user now inherits the roles of this group.

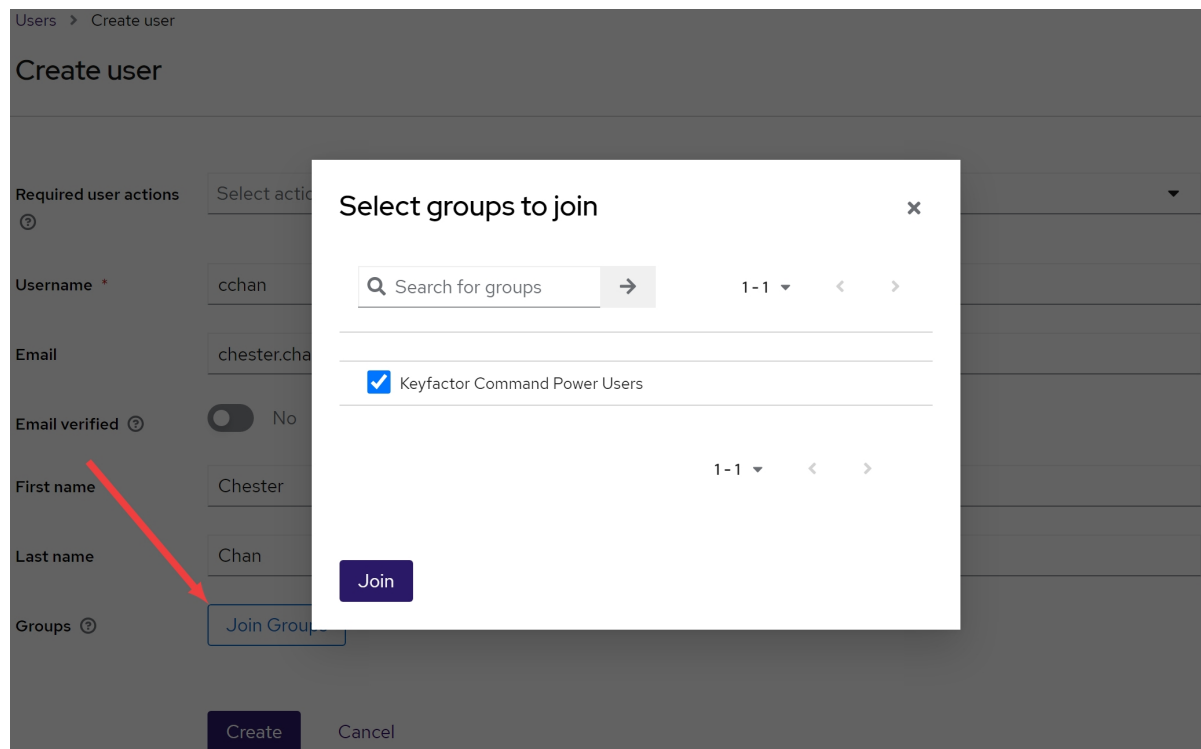


Figure 23: Add a Keyfactor Identity Provider User

4. Once the user creation is complete, open the user details. In the user details on the Credentials tab, click **Set password** and set a temporary password for the new user. The user will be prompted to set a new password on initial logon unless you toggle the **Temporary** option to **Off**.




**Important:** Keyfactor highly recommends that you use strong passwords for any accounts or certificates related to Keyfactor Command and associated products, especially when these have elevated or administrative access. A strong password has at least 12 characters (more is better) and multiple character classes (lowercase letters, uppercase letters, numeral, and symbols). Ideally, each password would be randomly generated. Avoid password re-use.

Figure 24: Set a Password for the Keyfactor Identity Provider User

Repeat these steps for each user who will access Keyfactor Command using an identity provider other than Active Directory.

5. If you prefer to add roles directly rather than via groups, in the user details on the Role mapping tab, click **Assign role** and select a role for the new user.

 **Tip:** Roles assigned via group membership won't appear on the Role mapping tab unless you uncheck the **Hide inherited roles** checkbox.

Users > User details

Assign roles to cchan ✕

Filter by realm roles

Search by role name

1 - 5

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	command-admin-role	Keyfactor Command Administrators
<input type="checkbox"/>	offline_access	\${role_offline-access}
<input type="checkbox"/>	power_users_role	Keyfactor Command Power Users
<input type="checkbox"/>	read-only-role	Keyfactor Command Limited Access Users
<input type="checkbox"/>	uma_authorization	\${role_uma_authorization}

1 - 5

Assign
Cancel

Figure 25: Assign a Role to a Keyfactor Identity Provider User

## Service Accounts

Keyfactor Command uses client records in Keyfactor Identity Provider to provide some service account functions. You will or may need this type of service account if you plan to:

- Install Keyfactor Command (a service account is added to allow Keyfactor Command to make API requests to Keyfactor Identity Provider)
- Use the Keyfactor API
- Use the Keyfactor Universal Orchestrator



**Note:** Client accounts for these functions should be created in Keyfactor Identity Provider even if you plan to use federation for your users. Token authentication requests for these type of functions are not federated.

To add clients for service account in Keyfactor Identity Provider:

1. Use a browser to open the Keyfactor Identity Provider management interface. For example:

<https://appsrvr18.keyexample.com:1443>

Click the **Administration Console** link and sign in with an administrative user and password (see [Installing Using Docker Compose on page 19](#)).



2. In the Keyfactor Identity Provider Administration Console, select Keyfactor in the realm drop-down.

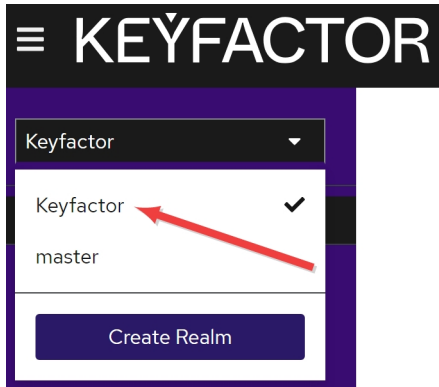


Figure 26: Select a Realm in the Keyfactor Identity Provider Administration Console

3. In the Keyfactor Identity Provider Administration Console, browse to *Clients*. Click **Create client** to add a service account. On the General Settings tab, select a **Client type** of *OpenIdConnect* and enter a unique **Client ID**. This Client ID will be how you will reference the service account from Keyfactor Command. It should not contain spaces. Give the client a **Name** and **Description**. Toggle the **Always display in UI** option to *On* to allow the account to always appear in the UI even when it's not in active use. Click **Next**.

### Create client

Clients are applications and services that can request authentication of a user.

A screenshot of the 'Create client' form in the Keyfactor Identity Provider Administration Console. The form is on the 'General Settings' tab. On the left, there is a sidebar with three tabs: '1 General Settings' (selected), '2 Capability config', and '3 Login settings'. The main form area contains the following fields: 'Client type' (a dropdown menu with 'OpenID Connect' selected), 'Client ID' (a text input field with 'Command-API-Query' entered), 'Name' (a text input field with 'Keyfactor Command API Requests' entered), and 'Description' (a text input field with 'Keyfactor Command queries via API' entered). At the bottom, there is a toggle switch for 'Always display in UI' which is turned 'On'.

Figure 27: Add a Keyfactor Identity Provider Service Account (Client): General

4. On the Capability config tab, toggle **Client authentication** to *On* and in the Authentication flow section, uncheck everything except **Service accounts roles**. Click **Next**.

1 General Settings

2 Capability config

3 Login settings

Client authentication ? ☒ On

Authorization ? ☐ Off

Authentication flow

☐ Standard flow ? ☐ Direct access grants ?

☐ Implicit flow ? ☒ Service accounts roles ?

☐ OAuth 2.0 Device Authorization Grant ?

☐ OIDC CIBA Grant ?

Figure 28: Add a Keyfactor Identity Provider Service Account (Client): Capabilities

- On the Login settings tab, click **Save**. You do not need to populate any of the data on this tab.
- In the Client details on the Credentials tab, click the **Copy** button next to the *Client secret* field to copy the unmasked version of the client secret to the clipboard (you do not need to display it unmasked first) and save this in a secure location. For the service account Keyfactor Command uses to make API requests, you will need this and the Client ID during the Keyfactor Command configuration.

Command-API-Query OpenID Connect ☒ Enabled ? Action ▼

Clients are applications and services that can request authentication of a user.

Settings Keys **Credentials** Roles Client scopes Service accounts roles Sessions Advanced

Client Authenticator ? Client Id and Secret ▼

Save

Client secret ..... Copy Regenerate

This is the Client ID.

Figure 29: Copy the Keyfactor Identity Provider Service Account (Client) Secret

## Federating from Keyfactor Identity Provider

Keyfactor Command can be used with a variety of OAuth 2.0 compliant providers via federation through Keyfactor Identity Provider. Once you have finished configuring Keyfactor Identity Provider, you're ready to federate to an additional OAuth provider, if desired. You may choose to manage your users and groups in Keyfactor Identity Provider and not add federation to an additional provider. Federation can be added at any time. The examples below show configuring Okta as a federated

identity provider. If you're not using Okta, this may give you sufficient information to configure the identity provider of your choice, since configuration tends to be similar.

The below configuration steps assume that you have already completed the installation of Keyfactor Identity Provider (see [Installing Keyfactor Identity Provider on page 12](#)) and have created at least one role in Keyfactor Identity Provider that you will use to grant permissions in Keyfactor Command (see [Using Keyfactor Identity Provider on page 32](#)). Keyfactor Command access control with Keyfactor Identity Provider and federation works by granting permissions in Keyfactor Command to Keyfactor Identity Provider roles, assigning those roles to users in Keyfactor Identity Provider, and importing user records from the federated identity provider automatically on login for any user attempting to access Keyfactor Command (see [Figure 31: Federated Identity Provider Login Flow](#)). The user accounts in the federated identity provider need to be assigned roles that mirror the roles in Keyfactor Identity Provider to provide a seamless first-time login experience for users.

**KEYFACTOR**  
Sign in to your account

Username or email  
[text input]

Password  
[password input]

[Sign In](#)

Or sign in with

[Okta-OIDC](#)

The user may sign in directly to Keyfactor Identity Provider above or login to the federated identity provider by clicking the link here.

**okta**

Sign In

Username  
john.smith@keyexample.com

Password  
[password input]

☐ Keep me signed in

[Sign in](#)

[Forgot password?](#)

[Help](#)

Figure 30: Login Page with Choice of Federated Identity Provider

No changes are needed to the configuration of the OAuth provider in Keyfactor Command when you add federation because all requests are brokered through Keyfactor Identity Provider. At login, the user is presented with a Keyfactor Identity Provider login prompt where he or she can choose to login directly to Keyfactor Identity Provider or click a link that will redirect to the login page of the federated identity provider.

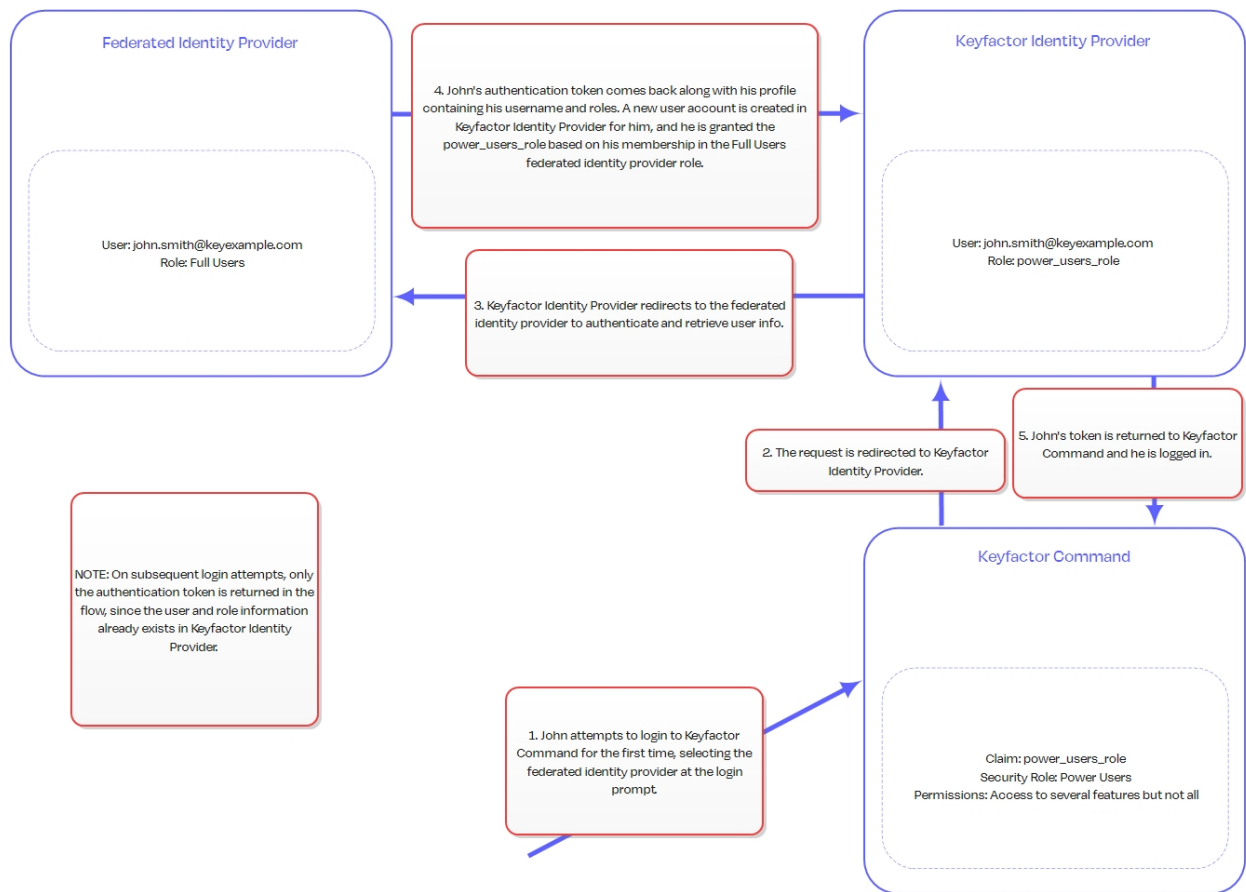



Figure 31: Federated Identity Provider Login Flow

## Federating to an Okta OpenID Connect Application

Federating to an Okta OpenID Connect application involves some information gathering and configuration on the Okta side and some configuration in Keyfactor Identity Provider.

To configure the Okta OIDC application and gather information:

1. Login to your Okta management console, browse to *Applications > Applications* and open the details for the application you will be federating. On the General tab, locate your *Client ID* and *Client Secret* and make note of these. You will need these when configuring the federated link from Keyfactor Identity Provider. Confirm that your application is configured to *Require PKCE as additional verification*. Keyfactor strongly recommends the use of this option for best security practice.



# KeyExample-OIDC

Active
View Logs

General
Sign On
Assignments
Okta API Scopes
Application Rate Limits

## Client Credentials

Client ID

Ooa[REDACTED]Zuk5d7

Public identifier for the client that is required for all OAuth flows.

Client authentication

☒ Client secret
☐ Public key / Private key

Proof Key for Code Exchange (PKCE)

☒ Require PKCE as additional verification

## CLIENT SECRETS

Creation date

Secret

Status

Jul 20, 2023	[REDACTED]	Active
--------------	------------	--------

Generate new secret

Figure 32: Client ID and Secret in Okta OIDC Application

- In the Okta OIDC application on the General tab, scroll down to the LOGIN section. Update these values as follows:

- Sign-in redirect URIs

The URI for the Keyfactor Identity Provider federated identity provider endpoint. For example:

```
https://appsrvr18.keyexample.com:1443/realms/Keyfactor/broker/Okta-OIDC/endpoint
```

Where **appsrvr18.keyexample.com** is the fully qualified domain name of the server on which you installed or will install Keyfactor Identity Provider, **Keyfactor** is the name of the realm in Keyfactor Identity Provider (the default is Keyfactor), and **Okta-OIDC** is the name you give to the federated identity provider broker you create in Keyfactor Identity Provider to link to your Okta OIDC application.

- Sign-out redirect URIs

The URI to which users should be redirected on logout. For example:

`https://appsrvr18.keyexample.com:1443/realms/Keyfactor/broker/Okta-OIDC/endpoint/logout_response`

Where `appsrvr18.keyexample.com` is the fully qualified domain name of the server on which you installed or will install Keyfactor Identity Provider, `Keyfactor` is the name of the realm in Keyfactor Identity Provider (the default is Keyfactor), and `Okta-OIDC` is the name you give to the federated identity provider broker you create in Keyfactor Identity Provider to link to your Okta OIDC application.

#### LOGIN




Sign-in redirect URIs 	<input type="checkbox"/> Allow wildcard * in login URI redirect.
	<code>https://appsrvr186.keyexample.com:3443/realms/Keyfactor/broker/Okta-OIDC/endpoint</code>
Sign-out redirect URIs 	<code>https://appsrvr186.keyexample.com:3443/realms/Keyfactor/broker/Okta-OIDC/endpoint/logout_response</code>
Login initiated by	App Only
Initiate login URI 	

Figure 33: Redirect URIs for the Okta OIDC Application

3. In the Okta management console, browse to *Security > API*, and on the Authorization Servers tab, open the authorization server for your application. In the authorization server details on the Claims tab, click **Add Claim** to create a new claim to send role information to Keyfactor Identity Provider to allow you to map roles in Okta to roles in Keyfactor Identity Provider and then use the roles to assign permissions in Keyfactor Command.

In the Edit Claim dialog:

- Give the claim a **Name** to indicate its purpose. You will need to reference this name when creating maps in your Keyfactor Identity Provider federated identity provider.
- Set **Include in token type** to *ID Token Always*.
- Select a **Value type** of *Groups*.
- Enter a **Filter** to control which Okta roles are included with the user information delivered to Keyfactor Identity Provider. To deliver all the groups, you can choose *Matches regex* and enter a regex of *.\** or use a *Starts with* filter, for example, if all the roles you wish to use with Keyfactor Identity Provider start with the same value (e.g. kyf).
- Set **Include in** to *Any scope*.

**Edit Claim**

Name

Include in token type

Value type

Filter ⓘ Only include groups that meet the following condition.

Disable claim ☐ Disable claim

Include in ☒ Any scope ☐ The following scopes:

Figure 34: Create an Authorization Server Role Claim

To configure Keyfactor Identity Provider to add a federated identity provider for the Okta OIDC application:

1. Use a browser to open the Keyfactor Identity Provider management interface. For example:

<https://appsrvr18.keyexample.com:1443>

Click the **Administration Console** link and sign in with an administrative user and password (see [Installing Using Docker Compose on page 19](#)).

2. In the Keyfactor Identity Provider Administration Console, select Keyfactor in the realm drop-down.

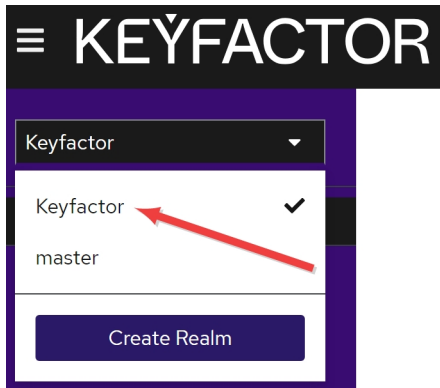


Figure 35: Select a Realm in the Keyfactor Identity Provider Administration Console

3. In the Keyfactor Identity Provider Administration Console, browse to *Identity providers* and click **Add provider**. In the top section, accept the default **Redirect URI**, in the **Alias** field enter the name you used when constructing your redirect URIs in Okta in the previous step, and enter a compatible **Display Name**.



**Tip:** The alias field is used as the *identity provider hint* during the configuration of Keyfactor Command if you wish to bypass the Keyfactor Identity Provider login and send users directly to the Okta login.

Identity providers > Add OpenID Connect provider

### Add OpenID Connect provider

Redirect URI ⓘ	<input type="text" value="https://appsrvr186.keyexample.com:3443/realms/Keyfactor/broker/oidc/endpoint"/>	
Alias * ⓘ	<input type="text" value="Okta-OIDC"/>	
Display name ⓘ	<input type="text" value="Okta-OIDC"/>	
Display order ⓘ	<input type="text"/>	

Figure 36: Give the Keyfactor Identity Provider Identity Provider an Alias

4. In the configuration page for the new provider in the OpenID Connect settings section, choose **Use discovery endpoint** and enter the URL to the discovery endpoint for your application's authorization server. For example:

```
https://${yourOktaDomain}/oauth2/${authorizationServerId}/.well-known/oauth-authorization-server
```



## OpenID Connect settings

Use discovery endpoint ☒ On

Discovery endpoint \*

[Show metadata](#)

Figure 37: Enter the Okta Discovery Endpoint in the Keyfactor Identity Provider Identity Provider

Click **Show metadata** to review the data retrieved from the discovery endpoint.

- Once the information populates, toggle **Use discovery endpoint** off and click **Show metadata** to display the additional fields. Toggle the **Use PKCE** to *On* and set the **PKCE Method** to *S256* (unless you didn't enable PKCE in Okta).

Validate Signatures ☒ On

Use JWKS URL ☒ On

JWKS URL

Use PKCE ☒ On Toggle Use PKCE to On and set the PKCE Method to S256

PKCE Method

Figure 38: Enable PKCE in the Keyfactor Identity Provider Identity Provider

- In the **Client ID** field enter the client ID for your Okta application and enter the secret for your Okta application in the **Client Secret** field.

Client authentication

Client ID \*

Client Secret \*

Client assertion signature algorithm

Figure 39: Add Okta Client ID and Secret in the Keyfactor Identity Provider Identity Provider

- Click **Add** to create the identity provider.
- Expand **Advanced**, and in the **Scopes** field add *openid profile*.

▼ **Advanced**

Pass login\_hint ? ☐ Off

Pass max\_age ? ☐ Off

Pass current locale ? ☐ Off

Backchannel logout ? ☐ Off

Disable user info ? ☐ Off

Scopes ? openid profile

Prompt ? Unspecified ▼

Add openid and profile to Scopes.

Figure 40: Deliver the Okta openid and profile to the Keyfactor Identity Provider Identity Provider

- At the top of the identity providers page, switch to the Mappers tab and click **Add mapper**. Here you're mapping the usernames from Okta to the Username field in Keyfactor Identity Provider to allow user records for each Okta user to automatically be generated in Keyfactor Identity Provider.

On the Add Identity Provider Mapper dialog:

- Give the mapper a **Name** that will help you identify it.
- Choose a **Sync mode override** of *Import*.
- Choose a **Mapper type** of *Username Template Importer*.
- Set the **Template** to `${CLAIM.preferred_username}`.



**Note:** This assumes the value in the Okta *preferred\_username* is the one you wish to use as the username for login to Keyfactor Command.

- Choose a **Target** of *LOCAL*.

## Add Identity Provider Mapper

Name *	Sub Mapper
Sync mode override *	Import
Mapper type	Username Template Importer
Template	\${CLAIM.preferred_username}
Target	LOCAL

[Save](#) [Cancel](#)

Figure 41: Map the Okta preferred\_username to the Keyfactor Identity Provider Identity Provider Username

- Return to the Provider details and click **Add mapper** again. Here you're mapping the roles from Okta to the roles in Keyfactor Identity Provider to allow the user records for each Okta user in Keyfactor Identity Provider to automatically be assigned roles in Keyfactor Identity Provider.

On the Add Identity Provider Mapper dialog:

- Give the mapper a **Name** that will help you identify it. If you'll be adding mappings for more than one Okta group, you may find it helpful to use a consistent naming pattern (e.g. Role Mapper: Admins, Role Mapper: Power Users).
- Choose a **Sync mode override** of *Import*.
- Choose a **Mapper type** of *Claim to Role*.
- Set the **Claim** to the name of the claim you created in Okta to include roles in the claim passed to Keyfactor Identity Provider (e.g. *kyf\_role*) as per the Okta steps, above.
- Click Select Role and choose the role in Keyfactor Identity Provider that should be mapped to the incoming Okta role.



**Note:** This step assumes that you've already set up a role in Keyfactor Identity Provider for this purpose (see [Using Keyfactor Identity Provider on page 32](#)).

Repeat this step for any additional roles from Okta that should be mapped to Keyfactor Identity Provider roles, using the same Claim name for each (all the roles are passed in the same claim). Any roles that come in with the claim that you do not create a mapping for will be ignored.

## Add Identity Provider Mapper

Name *	Role Mapper: Admins
Sync mode override *	Import
Mapper type	Claim to Role
Claim	kyf_role
Claim Value	Command Admins
Role	command-admin-role <a href="#">Select Role</a>


[Save](#) [Cancel](#)

Figure 42: Map the Okta Roles to the Keyfactor Identity Provider Identity Provider Roles

Configuration is complete. No changes are needed to the configuration in Keyfactor Command. It's helpful to restart the web server services (run an `iisreset`) on the Keyfactor Command server after making the changes to clear any cached data.

### 2.4.2.2 SQL Server

Keyfactor Command uses a Microsoft SQL Server database to store configuration and synchronized certificate information. Standard edition or above of SQL Server is required. In a production implementation, Keyfactor recommends that SQL Server be installed on a separate server from the Keyfactor Command roles.

 **Note:** Microsoft SQL 2017, 2019, and 2022 all with TLS encryption enabled are supported.

Although you can implement a SQL server especially for Keyfactor Command, in many environments an existing shared SQL server or cluster is used. Keyfactor Command creates one database with a user-defined name and can successfully co-exist with other databases in the same SQL instance.

SQL should be installed with a case-insensitive collation setting.

#### Connecting to SQL over SSL

By default, Keyfactor Command connects to SQL using an encrypted connection. This requires configuration of an SSL certificate on your SQL server.

If your SQL server is not configured correctly for SSL, you'll see an error message similar to the following when you try to make a connection from Keyfactor Command:

```
Unable to establish a connection to the database server. Please ensure that the server name is correct and sufficient privileges have been granted to the connection account.: Encountered an invalid or untrusted certificate and could not connect to the database. TLS encryption is enabled by default. Please visit 'Planning and Preparing --> SQL Server' In the Keyfactor Installing Server guide to resolve this.
```

Log message will look something like:

```
2022-09-09 11:35:13.0142 CSS.CMS.Install.ConfigurationWizard.ViewModels.DatabaseViewModel [Error] - Unable to establish a connection to the database server. Please ensure that the server name is correct and sufficient privileges have been granted to the connection account.
2022-09-09 11:35:13.0142 CSS.CMS.Install.ConfigurationWizard.ViewModels.DatabaseViewModel [Error] - Encountered an invalid or untrusted certificate and could not connect to the database. TLS encryption is enabled by default. Please visit 'Planning and Preparing --> SQL Server' in the Keyfactor Installing Server guide to resolve this.
at CSS.CMS.Install.ConfigurationWizard.ViewModels.DatabaseViewModel.a(Object A_0, RunWorkerCompletedEventArgs A_1)
A connection was successfully established with the server, but then an error occurred during the login process. (provider: SSL Provider, error: 0 - The certificate chain was issued by an authority that is not trusted.)
```

To acquire a new SSL certificate or check for an existing certificate, see [Using SSL to Connect to SQL Server on page 54](#).

If you would prefer not to use an encrypted channel for your connection to SQL, see [Configurable SQL Connection Strings on page 58](#).

### Database Encryption

Keyfactor Command uses Microsoft SQL Server column encryption with the ENCRYPTBYKEY and DECRYPTBYKEY cryptographic functions to protect sensitive data. The type of data protected in this way includes:

- Service account credentials
- SMTP credentials
- Certificate store passwords
- Certificate and pending certificate request private keys
- API secrets
- The 64-byte key used to sign audit log records

SQL encryption is built in to the product and cannot be disabled. In addition to SQL encryption, Keyfactor Command offers optional application-level encryption. This option allows you to encrypt

select sensitive data stored in the Keyfactor Command database using a separate encryption methodology utilizing a Keyfactor Command-defined certificate on top of the SQL server encryption. This additional layer of encryption protects the data in cases where the SQL Server master keys cannot be adequately protected. For more information, see [Application-Level Encryption on page 60](#).

#### Database Backup

Backup of the SQL server Database Master Key (DMK) for the Keyfactor Command database is of critical importance in database backup and recovery operations. The backup file of the DMK and the password for it should be stored in a safe, well-documented location. Without the file and password created with this process, some data that is encrypted within the Keyfactor Command database will be unrecoverable in a disaster recovery scenario. For more information, see *SQL Encryption Key Backup* in the *Keyfactor Command Reference Guide*.

#### High Availability

For a highly available solution, Keyfactor recommends using always on availability groups. The availability groups feature of SQL Server sits on top of Windows Server failover clustering and provides the ability to automatically synchronize multiple copies of databases across geographically dispersed SQL Servers. Although the availability groups feature relies on Windows clustering, it does not require shared storage, so it is appropriate for a geo-redundant deployment. The availability groups feature is the current recommended solution from Microsoft. Because Keyfactor Command makes use of SQL database encryption, when availability groups are configured, the Keyfactor Command service master key (SMK) must be synchronized between all participating nodes in the availability group. This can be accomplished by backing up the SMK from one SQL server and restoring it to the other servers in the availability group. For more information, see *SQL Encryption Key Backup* in the *Keyfactor Command Reference Guide*.

### Grant Permissions in SQL

The user installing Keyfactor Command needs permissions to administer the SQL server and add databases and users (logins) in SQL. Full sysadmin permissions are needed to upgrade from a previous version of Keyfactor Command if the user running the upgrade is not the same user who installed the previous version of Keyfactor Command.

#### Windows Authentication

If you opt to use Windows authentication for the Keyfactor Command connection to SQL during the installation, the user who installs Keyfactor Command needs appropriate permissions in SQL. To grant this, add the user who will install Keyfactor Command to the SQL server login list:

1. On the SQL server open the SQL Server Management Studio, connect to the database, and open **Security**.
2. Right-click on **Logins** and choose **New Login**.

3. Select the **Windows authentication** radio button.
4. Enter the domain name and user name of the administrative user who will be installing Keyfactor Command in the **Login name** field.
5. On the Login Properties page for this user, open Server Roles and check either the sysadmin role or the dbcreator, public and securityadmin roles, depending on whether this is a new install or an upgrade (see above).
6. Accept the remainder of the defaults and click **OK**.

Once Keyfactor Command has been deployed, the Windows user used for the install can be removed from the Logins under Security in the SQL Server Management Studio. Ongoing connectivity to the database is maintained using logins automatically created in SQL for the Keyfactor Command application pool users (see [Create Service Accounts for Keyfactor Command on page 66](#)) specifically for the purpose during the installation.

### SQL Authentication

If you opt to use SQL authentication, appropriate permissions need to be granted to the SQL user entered in the initial connection dialog of the Keyfactor Command Configuration Wizard. You may choose to create (or use an existing) SQL user for the installation and create a separate SQL user for ongoing connectivity or use the same user for both purposes.



**Note:** Your SQL server must be configured to support mixed mode authentication in order to use the SQL authentication option.

To create a new SQL user for the initial SQL connection:

1. On the SQL server open the SQL Server Management Studio, connect to the database, and open **Security**.
2. Right-click on **Logins** and choose **New Login**.
3. Select the **SQL Server authentication** radio button.
4. Enter a user name for the SQL user in the **Login name** field and enter and confirm a **Password**. You may wish to uncheck the **User must change password at next login** box.
5. On the Login Properties page for this user, open Server Roles and check either the sysadmin role or the dbcreator, public and securityadmin roles, depending on whether this is a new install or an upgrade (see above).
6. Accept the remainder of the defaults and click **OK**.

Once Keyfactor Command has been deployed, this SQL user may be removed if it is not also serving the role of providing ongoing connectivity. During installation, you enter a SQL user name and password for a login to maintain ongoing connectivity. If this login already exists in SQL, it will be granted

appropriate permissions. If this login does not already exist in SQL, it will be created and granted appropriate permissions.



**Note:** Automatically generated service accounts are not created with the db\_owner role. Instead, a keyfactor\_db\_role is created and granted to the service accounts. This role has permission on each of the schemas (dbo, ssl, ssh, cms\_agents, etc.) and permission on the encryption certificate.

## Using SSL to Connect to SQL Server

By default, Keyfactor Command connects to SQL using an encrypted connection using an SSL certificate configured on your SQL server.

You can check whether your SQL server has been configured with an SSL certificate in one of two ways:

### SQL Server Configuration Manager

1. On the SQL server, open the SQL Server Configuration Manager and drill down under SQL Server Network Configuration to find *Protocols for [YOUR INSTANCE NAME]*.
2. Right-click on Protocols for [YOUR INSTANCE NAME] and choose **Properties**.
3. Check the Certificate tab of the Properties dialog to see if a certificate has been configured and is still valid. If your certificate has a friendly name, it will appear here listed in the dropdown by its friendly name.



**Important:** A certificate will only appear here if it has a CN<sup>1</sup>, usually the FQDN of the SQL server. If a certificate has been configured without this, it will not appear to be configured through this UI.

---

<sup>1</sup>The Subject property of the certificate must indicate that the common name (CN) is the same as the host name or fully qualified domain name (FQDN) of the server computer or it must match the DNS suffix if using a wildcard certificate. When using the host name, the DNS suffix must be specified in the certificate. If SQL Server is running on a failover cluster, the common name must match the host name or FQDN of the virtual server and the certificates must be provisioned on all nodes in the fail-over cluster.



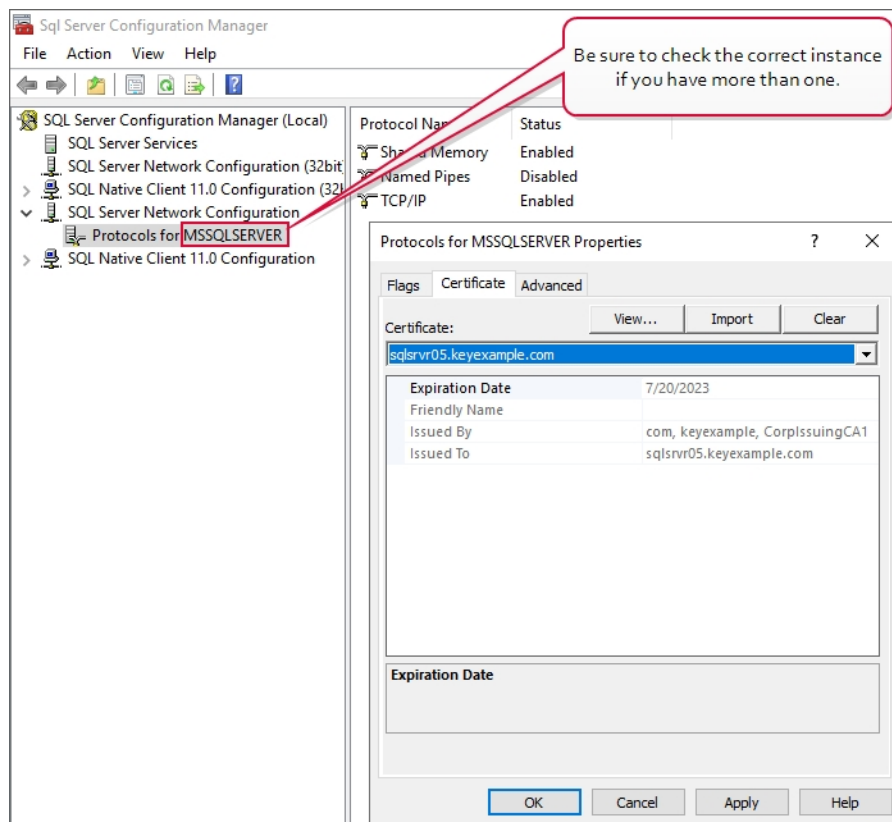


Figure 43: SQL Server Configuration Manager View Active SSL Certificate

## Registry

1. On the SQL server, open the registry editor and browse to (where `[MSSQL15.MSSQLSERVER]` is the correct version of SQL server for your server):

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\  
[MSSQL15.MSSQLSERVER]\MSSQLServer\SuperSocketNetLib
```

2. In the SuperSocketNetLib registry key, look for a Certificate value.
3. Validate that the Certificate value has a thumbprint configured. This should match the thumbprint of an active certificate with a Server Authentication ECU in the Local Machine certificate store.

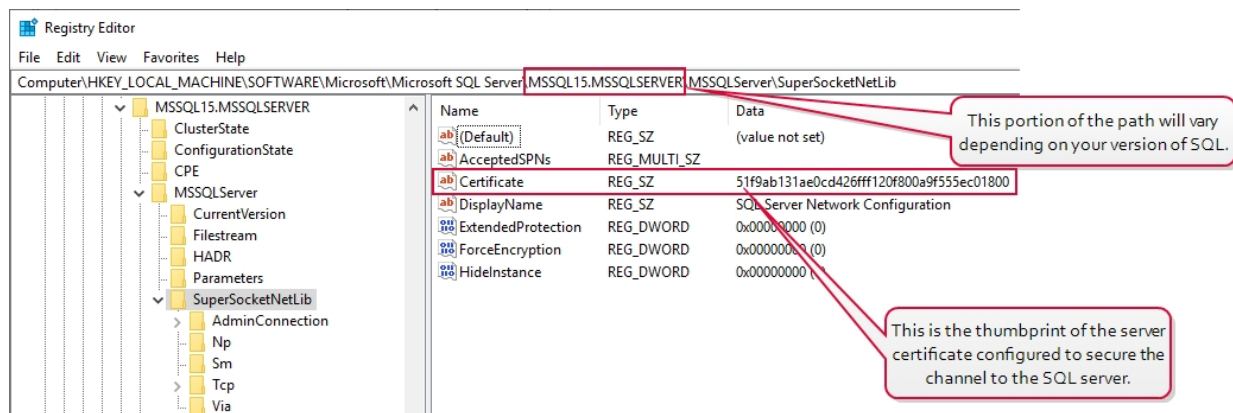


Figure 44: Registry View Active SSL Certificate

To acquire a new certificate for your SQL server:

1. On the SQL server open the Services.msc MMC and scroll down to locate the *SQL Server ([YOUR INSTANCE NAME])* service.
2. Check the *Log On As* column for the name of the service account that the service is running as.

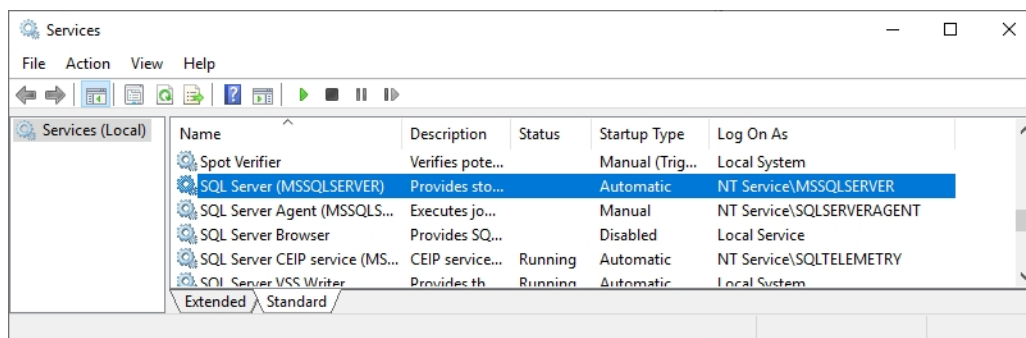


Figure 45: View SQL Server Services

3. Identify a template with a Server Authentication EKU (a typical web server template).
4. On the SQL server, do one of following:
  - Using the GUI:
    - a. Open an empty instance of the Microsoft Management Console (MMC).
    - b. Choose **File->Add/Remove Snap-in...**
    - c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.
    - d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.
    - e. Click **OK** to close the Add or Remove Snap-ins dialog.

- Using the command line:
  - a. Open a command prompt using the “Run as administrator” option.
  - b. Within the command prompt type the following to open the certificates MMC:

certlm.msc

5. Enroll for the certificate using your preferred method, being sure to give the certificate a CN (it will not appear in the configuration tool without this) and add subject alternative names (SANs) to it for all the IP addresses, server names, and FQDNs that you might use to reference the SQL server when communicating with it, including DNS aliases. Install it, along with its private key, into the Local Machine certificate store on the SQL server. One way to do this is in the certificates MMC:
  - a. Drill down to the Personal folder under **Certificates** for the Local Computer, right-click, and choose **All Tasks->Request New Certificate....**
  - b. Follow the certificate enrollment wizard, selecting the template you identified for this purpose, and providing appropriate SANs along with any required information.

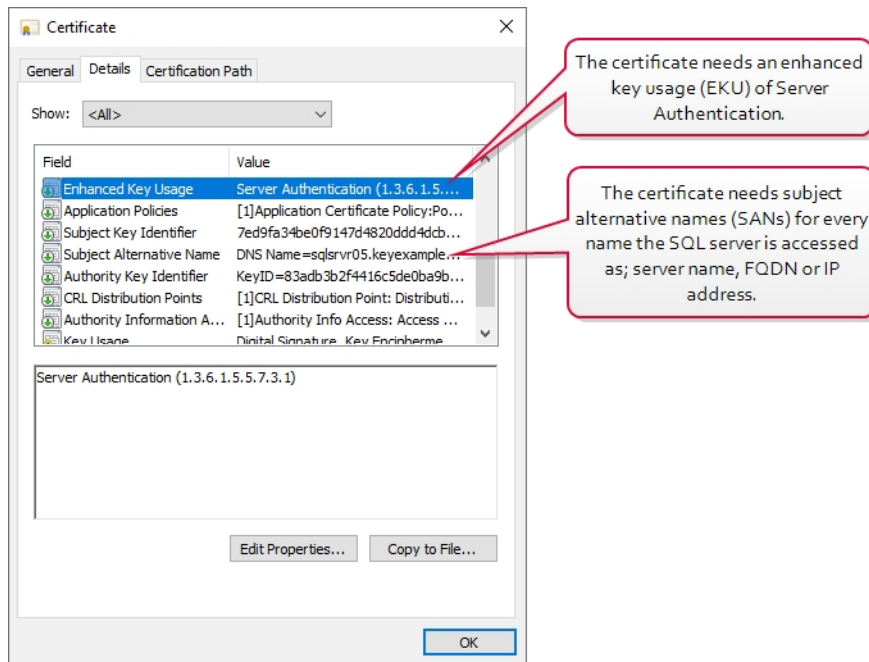


Figure 46: SQL Server SSL Certificate Details

6. Drill down to the Personal folder under **Certificates** for the Local Computer, locate your certificate, right-click, and choose **All Tasks->Manage Private Keys....**
7. In the Permissions for private keys dialog, click **Add**, add the SQL service account, and grant that service account **Read** but not **Full control** permissions. If the SQL server is running as

*NT Service\[YOUR INSTANCE NAME]* as shown in [Figure 45: View SQL Server Services](#), be sure to change the location to your local machine and enter the object name as “*NT SERVICE\[YOUR INSTANCE NAME]*” as shown in [Figure 47: Grant Private Key Permissions for SQL Server](#).

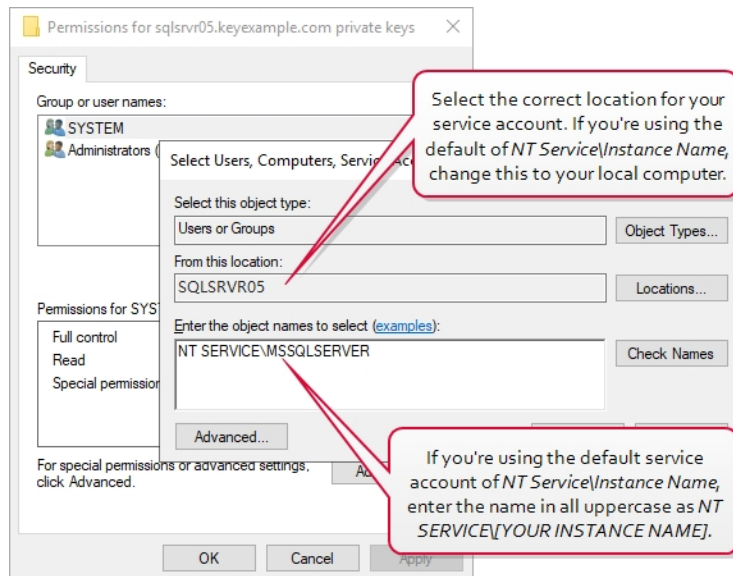


Figure 47: Grant Private Key Permissions for SQL Server

8. Click **OK** to save.
9. Configure the SSL certificate in SQL using either the SQL Server Configuration Manager or registry as shown above for checking whether there is an existing certificate configured (see [SQL Server Configuration Manager on page 54](#)).
10. After you’ve acquired a new certificate, made the private key permission changes, and associated it in SQL, you’ll need to restart the *SQL Server (Instance Name)* service (see [Figure 45: View SQL Server Services](#)) before these changes will take effect.

## Configurable SQL Connection Strings

Keyfactor Command supports using a template SQL connection string that can be created to fit the needs of the overall deployment. This template will be used as a starting point and will not be overwritten by the configuration wizard. For instance, you can set the timeout setting in one place, and once the configuration wizard is run, this is reflected in all places where a connection string is used. The template can be changed at any time to update the connection strings.

To create a customized connection string, after installing the Keyfactor Command software but before running the configuration wizard, modify both the EFModels and SqlDirect connection strings in the **SqlConnectionStrings.json** file found in the *Configuration* folder under your installation directory. By default, this is:

C:\Program Files\Keyfactor\Keyfactor Platform\Configuration

The settings that can be modified are described in the following Microsoft article:

<https://docs.microsoft.com/en-us/dot-net/api/system.data.sqlclient.sqlconnection.connectionstring?view=dotnet-plat-ext-6.0>



**Note:** The *Data Source*, *Initial Catalog*, *Integrated Security*, *User ID* and *Password* settings are reserved for the configuration wizard to use to configure and save the authentication for the connection string, but other settings found in the template string are left as-is.

```
{
  "ConnectionStrings": {
    "SqlDirect": "Data Source=<SQL_MACHINE_FQDN>;Initial
Catalog=<SQL_DB_NAME>;Integrated Security=True;Persist Security
Info=True;Command Timeout=360",
    "EFModels":
      "metadata=res://*/EFModels.csdl|res://*/EFModels.ssdl|res://*/EFModels.msl;provid
er=Microsoft.Data.SqlClient;provider connection string='Data
Source=<SQL_MACHINE_FQDN>;Initial Catalog=<SQL_DB_NAME>;Integrated
Security=True;Persist Security Info=True;Command Timeout=360;Multiple Active
Result Sets=True;Application Name=EntityFramework'"
  }
}
```

Note that *Data Source*, *Initial Catalog*, *Integrated Security*, *User ID* (not shown) and *Password* (not shown) are reserved for the configuration wizard to use to configure and save the authentication for the connection string.

Figure 48: Default SQL Connection Strings

## Disable Encryption

If you prefer to connect to your SQL server over a non-encrypted channel (and thus avoid configuring an SSL certificate for your SQL server), you can use the *Encrypt* keyword in the connection strings with a value of *False*.

```
{
  "ConnectionStrings": {
    "SqlDirect": "Data Source=<SQL_MACHINE_FQDN>;Initial
Catalog=<SQL_DB_NAME>;Integrated Security=True;Persist Security
Info=True;Command Timeout=360",
    "EFModels":
      "metadata=res://*/EFModels.csdl|res://*/EFModels.ssdl|res://*/EFModels.msl;provid
er=Microsoft.Data.SqlClient;provider connection string='Data
Source=<SQL_MACHINE_FQDN>;Initial Catalog=<SQL_DB_NAME>;Integrated
Security=True;Persist Security Info=True;Command
Timeout=360;Encrypt=False;Multiple Active Result Sets=True;Application
Name=EntityFramework'"
  }
}
```

To make a non-encrypted connection to SQL, add *Encrypt=False* to each of the connection strings.

Figure 49: SQL Connection Strings with Encrypt Channel Disabled

## Use a SQL Server Listening on Multiple IP Addresses

If you're using a SQL server cluster that's configured to listen for incoming connections on more than one IP address to support redundancy or access from multiple networks/subnets, you can use the *MultiSubnetFailover* keyword in the connection strings with a value of *True*.

```
{
  "ConnectionStrings": {
    "SqlDirect": "Data Source=<SQL_MACHINE_FQDN>;Initial
Catalog=<SQL_DB_NAME>;Integrated Security=True;Persist Security Info=True;Command
Timeout=360",
    "EFModels":
      "metadata=res://*/EFModels.csdl|res://*/EFModels.ssdl|res://*/EFModels.msl;provider
=Microsoft.Data.SqlClient;provider connection string='Data
Source=<SQL_MACHINE_FQDN>;Initial Catalog=<SQL_DB_NAME>;Integrated
Security=True;Persist Security Info=True;Command
Timeout=360;MultiSubnetFailover=True;Multiple Active Result Sets=True;Application
Name=EntityFramework'"
  }
}
```

Figure 50: SQL Connection Strings with MultiSubnetFailover Option Enabled

## Application-Level Encryption

Keyfactor Command uses data encryption for sensitive data—such as private keys for certificates—stored in the Keyfactor Command database (see [SQL Server on page 50](#)). This option encrypts only the data in the database deemed to be of a sensitive nature, not the entire database. By default, the data is encrypted using SQL encryption, but you have the option to add another level of security with application-level encryption. If you choose to enable this option, you will need a certificate for this purpose installed in the Personal Certificate store of the Local Computer on each Keyfactor Command server. The certificate must have a key usage of either key encipherment or data encipherment enabled. Microsoft certificate templates only allow you to configure data encipherment (“Allow encryption of user data”) as a suboption to key encipherment (“Allow key exchange only with key encryption”). You do not need to enable both. You may use the certificate acquired in the name of the Keyfactor Command web site (see [Acquire a Public Key Certificate for the Keyfactor Command Server on page 77](#)), assuming it supports the appropriate key usage, or you may enroll for a separate certificate for this purpose. The same certificate must be used on all Keyfactor Command servers and the certificate must be available in the certificate store on the machine when you run the Keyfactor Command installation. A hardware security module (HSM) may be used, if desired. To support the use of an HSM, the Windows CSP driver for the HSM must be installed on the Keyfactor Command server. Be aware that transactions accessing the encrypted data—such as enrolling for PFX certificates, downloading PFX certificates, running inventory, and adding certificates to certain types of certificate stores (e.g. F5, NetScaler)—will require accessing the HSM. A slow HSM will slow down these processes.



**Note:** In an environment where there are multiple copies of Keyfactor Command pointing to the same database, each server running a Keyfactor Command instance will need to have the same encryption certificate AND the corresponding private key.



**Note:** The thumbprint of the certificate used for application-level encryption is stored in the registry on the Keyfactor Command server(s)—rather than in the Keyfactor Command database—to provide a further level of separation from SQL.



**Important:** If the certificate used for application-level encryption or the private key for this certificate are removed from the Keyfactor Command server while data in the database is encrypted with this certificate, access to this data will be lost. Take care to ensure that this certificate and its private key remain in place or that there are backups of both the certificate and private key (with any necessary password) that can be accessed in the event that the certificate needs to be restored.

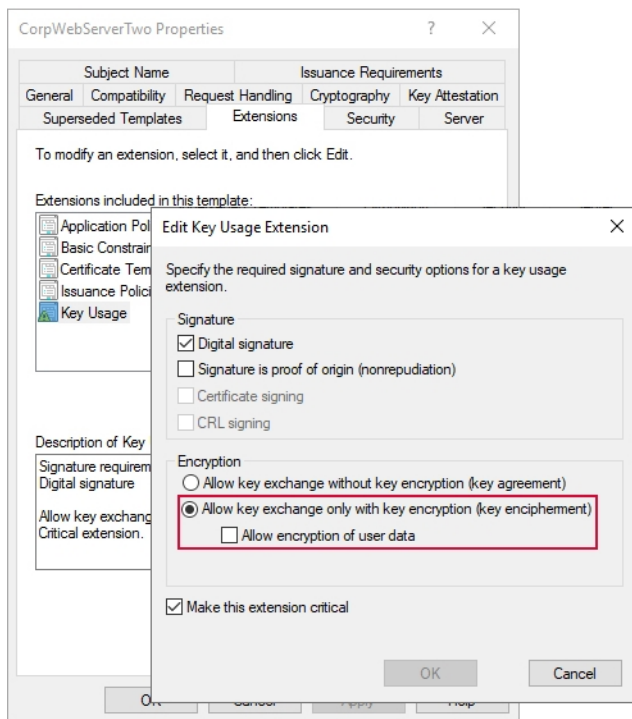


Figure 51: Certificate Template with Key Encipherment Key Usage

## Cryptographic Provider

The certificate you use for application-level encryption must be issued using a key storage provider (KSP) or at least installed on the Keyfactor Command server using a KSP in order to allow Keyfactor Command to grant permissions on the certificate's private key appropriately during installation. If Keyfactor Command is not able to access the private key of the application-level certificate, you may receive a 500.30 error trying to access the Management Portal or you may find you can access the Management Portal but not all features function and you find errors such as these in the log (which of these occurs depends on other configuration factors):



Unable to decrypt enveloped PKCS7 data

Keyset does not exist

You can check the provider on your certificate with a command similar to the following issued in an administrative command prompt:

```
certutil -store MY [thumbprint of the application encryption certificate]
```

Output from this command should like similar to the following for a KSP certificate:

```
MY "Personal"
===== Certificate 3 =====
Serial Number: 1800000f39f4c506c41239c566000200000f39
Issuer: CN=CorpIssuingCA1, DC=Keyexample, DC=com
NotBefore: 10/27/2023 4:19 PM
NotAfter: 10/26/2025 4:19 PM
Subject: CN=keyfactor.keyexample.com
Non-root Certificate
Template: CorpWebServerv2, Corp Web Server v2
Cert Hash(sha1): 89b5099bc1f7146185331017db60373afb136edb
    Key Container = te-CorpWebServerv2-8b4d6ca7-e5ec-472f-a096-8b4aa590b22b
    Unique container name: 153703d13e6c7339e297f44547260e6d_00139f2c-9f21-4793-b740-bb3fe658245c
    Provider = Microsoft Software Key Storage Provider
Private key is NOT exportable
Encryption test passed
CertUtil: -store command completed successfully.
```

Output from this command should like similar to the following for a CSP certificate (legacy CSPs vary):

```
MY "Personal"
===== Certificate 0 =====
Serial Number: 59000006513beacf07ce121e45000100000651
Issuer: CN=CorpIssuingCA1, DC=Keyexample, DC=com
NotBefore: 10/27/2023 4:32 PM
NotAfter: 10/26/2025 4:32 PM
Subject: CN=keyfactor.keyexample.com
Non-root Certificate
Template: CorpWebServer, Corp Web Server
Cert Hash(sha1): a3a1299d3f5d209c89573c356495547b67d92f15
    Key Container = d5549bc8ea7af0f51d8b26ffbe9617b8_00139f2c-9f21-4793-b740-bb3fe658245c
    Simple container name: te-CorpWebServer-c6c249ac-66d3-427e-aff0-8de81250887f
    Provider = Microsoft RSA SChannel Cryptographic Provider
```



```
Private key is NOT exportable
Encryption test passed
CertUtil: -store command completed successfully.
```

If you have a certificate in PFX format with CSP and would like to import it as a KSP, you can use a command similar to the following:

```
certutil -importPFX -csp KSP MY <PFX Path and FileName> NoExport
```

Alternately, if you're unable to use a certificate with a KSP, you may use a CSP and manually grant the Keyfactor Command application pool user(s) private key read permissions on the certificate (see [Using SSL to Connect to SQL Server on page 54](#)).

### 2.4.2.3 Certificate Authorities

In most cases, if you are installing Keyfactor Command then you have at least one Microsoft or EJBCA Certificate Authority (CA) in your environment. As you're planning for Keyfactor Command, you'll need to make the following decisions and configuration checks for the CA(s) and certificate templates in your environment:

- Which CAs should be synchronized to the Keyfactor Command database?

Your license may not allow you to synchronize all of your CAs. Certificates belonging to offline root or policy CA *chain* certificates can be monitored without impacting your license.



**Note:** Keyfactor Command contains a constraint that prevents any two certificate authorities from having the same logical name and host name combination. Think about the logical name and host name configuration of the CAs that will be implemented with Keyfactor Command and check for duplicates.

- If you have Microsoft CAs, what authorization method will you use to configure the CAs (see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs on page 142](#))?
- If you have EJBCA CAs, are the protocols required for Keyfactor Command connectivity enabled in the EJBCA system configuration? The following protocols must be enabled:
  - REST Certificate Management
  - REST End Entity Management
  - REST End Entity Management V2
  - REST Certificate Management V2
  - Web Service

This SOAP API protocol is only required if you plan to disable (uncheck) the option to *Create new End Entity when renewing or reissuing certificates* in the Keyfactor Command certi-

ificate authority configuration (see *Certificate Authority Operations: Adding or Modifying a CA Record: The Basic Tab* in the *Keyfactor Command Reference Guide*).

## System Configuration

Basic Configurations
Administrator Preferences
Protocol Configuration

Extended Key Usages
Trusted OAuth Providers
Certificate Transparency Logs

Custom Certificate Extensions
Custom RA Styles
Statedump
External Scripts

Configuration Checker
External Account Bindings

**Enable or disable protocol access**

Protocol	Resource Default URL	Status	Actions
ACME	/ejbca/acme	✗ Disabled	Enable
Certstore	/ejbca/publicweb/certificates	✓ Enabled	Disable
CMP	/ejbca/publicweb/cmp	✓ Enabled	Disable
CRLstore	/ejbca/publicweb/crls	✓ Enabled	Disable
EST	/.well-known/est	✓ Enabled	Disable
MSAE	/ejbca/msae	✗ Disabled	Enable
OCSP	/ejbca/publicweb/status/ocsp	✓ Enabled	Disable
SCEP	/ejbca/publicweb/apply/scep	✓ Enabled	Disable
RA Web	/ejbca/ra	✓ Enabled	Disable
REST CA Management	/ejbca/ejbca-rest-api/v1/ca_management	✓ Enabled	Disable
REST Certificate Management	/ejbca/ejbca-rest-api/v1/ca_certificate	✓ Enabled	Disable
REST Coap Management	/ejbca/ejbca-rest-api/v1/coap	✗ Disabled	Enable
REST Crypto Token Management	/ejbca/ejbca-rest-api/v1/cryptotoken	✓ Enabled	Disable
REST End Entity Management	/ejbca/ejbca-rest-api/v1/ententity	✓ Enabled	Disable
REST End Entity Management V2	/ejbca/ejbca-rest-api/v2/ententity	✓ Enabled	Disable
REST Configdump	/ejbca/ejbca-rest-api/v1/configdump	✓ Enabled	Disable
REST Certificate Management V2	/ejbca/ejbca-rest-api/v2/certificate	✓ Enabled	Disable
REST SSH V1	/ejbca/ejbca-rest-api/v1/ssh	✗ Disabled	Enable
REST System V1	/ejbca/ejbca-rest-api/v1/system	✓ Enabled	Disable
Webdist	/ejbca/publicweb/webdist	✓ Enabled	Disable
Web Service	/ejbca/ejbcaaws	✓ Enabled	Disable

Figure 52: EJBCA Protocols for Keyfactor Command Connectivity

- If your Keyfactor Command license includes certificate enrollment:
  - Which CA(s) will be used to issue certificates based on CSRs through the Keyfactor Command Management Portal?
  - Which CA(s) will be used to issue PFXs through the Keyfactor Command Management Portal?
  - Which certificate template(s) will be used to generate CSRs through the Keyfactor Command Management Portal? In most cases, these templates will already exist.
  - Which certificate template(s) will be configured for CSR enrollment in the Keyfactor Command Management Portal? In most cases, these templates will already exist.

- Which certificate template(s) will be configured for PFX enrollment in the Keyfactor Command Management Portal? In most cases, these templates will already exist.
- If your Keyfactor Command license includes Mac auto-enrollment, which CA(s) will be used to automatically issue certificates to Macs running the Mac auto-enrollment agent?

As part of the Keyfactor Command installation preparation, you may need to create a certificate template for this purpose.



**Tip:** This information is not needed to complete the initial installation and configuration, but will be needed to do post-installation configuration in the Keyfactor Command Management Portal.

#### 2.4.2.4 Keyfactor Command Server(s)

A Keyfactor Command server implementation is made up of several Keyfactor Command roles:

##### **Keyfactor Command Management Portal**

The server with this role provides the web-based administration interface that is used to view and report on certificates issued in the environment and enroll for certificates. This role runs under Microsoft IIS. Configuration for the Keyfactor Command implementation as a whole is also done through the Keyfactor Command Management Portal. The Logi Analytics Platform for reporting is hosted on the server with this role.

This role is required on all Keyfactor Command servers.

##### **Keyfactor Command Windows Services**

The server with this role hosts back-end services required to support Keyfactor Command. This includes the Keyfactor Command Service, which is used for all periodic tasks throughout Keyfactor Command, including CA synchronization, monitoring alerts, and report automation.

This role is required on all Keyfactor Command servers.

##### **Keyfactor Command Web API**

The server with this role hosts the Keyfactor API. The Keyfactor API is also included in the Management Portal role, since the Management Portal makes extensive use of this API.

This role is optional. If you choose not to install this role, you will still be able to use the Keyfactor API. This role is available as a separate component for users who wish to install the Keyfactor API on a separate server from the Management Portal server.

##### **Keyfactor Command Orchestrator Service API**

The server with this role hosts the back-end service for receiving requests from and sending requests to Keyfactor agents and orchestrators.

This role is optional. If you choose not to install this role, you will not be able to use agents and orchestrators with Keyfactor Command.

In many environments, the Keyfactor Command Management Portal, Windows Services, Web API, and Orchestrator Service API roles are collocated on a single server (or pair of servers if redundancy is desired). Both physical and virtual servers are supported.



**Tip:** See [Install: Select Components on page 92](#) for related information.

For a high availability (HA) solution using the same roles on all nodes, note that the following conditions apply:

- All servers must point to the same Keyfactor Command SQL database.
- All servers must be configured with the same encryption certificate AND the corresponding private key (see [Database Tab on page 109](#)).
- Keyfactor recommends that the Keyfactor Command Service be configured to run all services on each node. This allows the service to manage the jobs most efficiently—the service will check out jobs via a locking mechanism that will enforce that any jobs are running on only one service at a time. However, you do have the option to manually tune the jobs on the servers if desired (such that server A always does jobs 1, 2 and 3 and server B always does jobs 4, 5 and 6).
- Review load balancing rules and configuration, if applicable. Load balancing configuration is beyond the scope of this guide.

Keyfactor does not recommend installing any of these roles on a CA or on a SQL server in a production environment.

As you plan for Keyfactor Command, you need to decide upon an architecture for the implementation and prepare servers with sufficient resources accordingly. See [System Requirements on page 9](#) for more information about planning for servers with sufficient resources to support the planned roles.

## Licensing

The Keyfactor Command product is licensed by component. Your license for Keyfactor Command may not include all the features described in this guide. If you choose to add additional components to your Keyfactor Command license in the future, these features can generally be configured without the need to reinstall Keyfactor Command.

### 2.4.2.5 Create Service Accounts for Keyfactor Command

Several of the Keyfactor Command roles operate under a service account. You can either create a single service account for all these roles or create separate service accounts for each role. If multiple Keyfactor Command roles will be installed on the same server, some of the below roles will be redundant.



**Important:** Keyfactor highly recommends that you use strong passwords for any accounts or certificates related to Keyfactor Command and associated products, especially when these have elevated or administrative access. A strong password has at least 12 characters (more is better) and multiple character classes (lowercase letters, uppercase letters, numeral, and symbols). Ideally, each password would be randomly generated. Avoid password re-use.

The roles that require a service account are:

## Keyfactor Command Installer

The user who runs the Keyfactor Command installation must have local administrator permissions on the Keyfactor Command server(s) and must be granted permissions in SQL if Windows authentication for SQL will be used during the installation (see [Grant Permissions in SQL on page 52](#)). You can either grant these permissions to an existing user or you can create a Keyfactor Command installer account and grant the appropriate permissions to this account.

Additionally, the user installing Keyfactor Command must have the SeBackupPrivilege and SeRestorePrivilege rights on the Keyfactor Command server. Normally, administrators are granted these permissions by default, but you should confirm the permissions prior to starting the install. These permissions can be set through Group Policy or Local Security Policy, and can be found under “Local Policies\User Rights Assignment” as “Back up files and directories” and “Restore files and directories”.

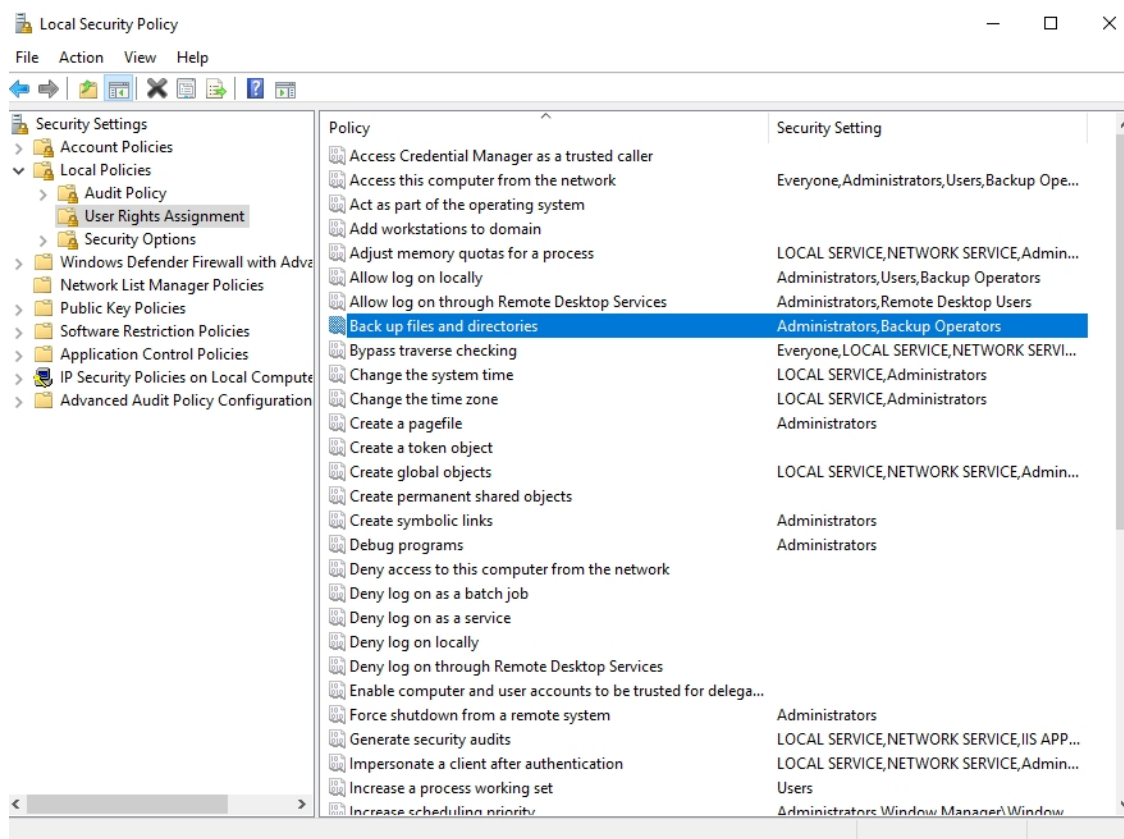


Figure 53: Local Security Policy

For more information on this from Microsoft, see:

<https://docs.microsoft.com/en-us/windows/win32/api/userenv/nf-userenv-load-userprofilea#remarks>

## Keyfactor Command Service

The Keyfactor Command Service (a.k.a. the timer service) runs on the Keyfactor Command services server. It synchronizes certificates to the SQL database and initiates notification and reporting tasks. This service runs in the context of an Active Directory or local service account.

The user with this role will be granted permission on each of the SQL schemas (dbo, ssl, ssh, cms\_agents, etc.) and permission on the encryption certificate in SQL through the keyfactor\_db\_role which is created during configuration.

The user with this role must have the “Log on as a service” right on the Keyfactor Command server. Normally, this permission is granted automatically as part of the installation process. You can confirm the permissions through Group Policy or Local Security Policy in “Local Policies\User Rights Assignment”. Validate that the user associated with the Keyfactor Command Service has been added to “Log on as a service” directly or indirectly (via group membership).

The user with this role needs to be able to create log files and write to them. During installation, this permission is granted by granting “Create files / write data” and “Create folders / append data” permissions on the log directory (C:\Keyfactor\logs) to the local users group on the assumption that the local users group will contain either “NT AUTHORITY\authenticated users” or “DOMAIN\Domain Users” and that the service account user will be granted permissions via at least one of these. If this is not the case, permissions for the service account user will need to be granted manually to the log directory.

The user with this role needs to be granted permissions on any certificate authorities from which certificates will be synchronized. Additional certificate authority permission may be needed depending on the features that will be used. For more information, see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs on page 142](#).

## Keyfactor Command Management Portal (Application Pool)

The Keyfactor Command Management Portal uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory or local service account.

The user with this role will be granted permission on each of the SQL schemas (dbo, ssl, ssh, cms\_agents, etc.) and permission on the encryption certificate in SQL through the keyfactor\_db\_role which is created during configuration.

The user with this role must have the “Log on as a batch job” and “Impersonate a client after authentication” rights on the Keyfactor Command server. In a typical IIS installation, these rights are granted to the IIS\_IUSRS group and the user running any application pool created in IIS inherits these rights without being added to the IIS\_IUSRS group. For more information about the IIS\_IUSRS group, see:

<https://learn.microsoft.com/en-us/iis/get-started/planning-for-security/understanding-built-in-user-and-group-accounts-in-iis>

You can confirm the permissions or set them manually for the application pool user through Group Policy or Local Security Policy in “Local Policies\User Rights Assignment”. Validate that either the IIS\_IUSRS group or the user associated with the Keyfactor Command application pool has been added to “Log on as a batch job” and “Impersonate a client after authentication” directly or indirectly (via group membership).

The user with this role needs to be able to create log files and write to them. During installation, this permission is granted by granting “Create files / write data” and “Create folders / append data” permissions on the log directory (C:\Keyfactor\logs) to the local users group on the assumption that the local users group will contain either “NT AUTHORITY\authenticated users” or “DOMAIN\Domain Users” and that the service account user will be granted permissions via at least one of these. If this is not the case, permissions for the service account user will need to be granted manually to the log directory.

The user with this role needs to be granted permissions on any certificate authorities from which certificates will be synchronized. Additional certificate authority permission may be needed depending on the features that will be used. For more information, see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs on page 142](#).





**Note:** If you're using Active Directory as an identity provider, the Application Pool account must have read permission on any groups being created. This will allow Keyfactor Command to query for group membership on the groups.

## Logi Report Access

The Logi Analytics Platform uses a service account to allow Logi to connect to Keyfactor Command via the Keyfactor API to display the dashboard information. This uses an application pool under IIS to operate. The application pool runs in the context of an Active Directory or local service account. A separate application pool is needed for this service, though the same service account may be used for both.

## Keyfactor Command Orchestrators API

The Keyfactor Command Orchestrators API IIS application accepts connections from Keyfactor Command orchestrators and uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory or local service account. If this role will be installed on the server hosting the Keyfactor Command Management Portal role, a separate application pool is needed for this service, though the same service account may be used for both.

## Keyfactor Command Keyfactor API

The Keyfactor Command Keyfactor API uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory or local service account. The Keyfactor API is an integral part of Keyfactor Command and is not an optional installation. The Keyfactor API can be configured to support custom applications. If the Keyfactor Command Keyfactor API role will be installed on the server hosting the Keyfactor Command Management Portal role, a separate application pool is needed for this service, though the same service account may be used for both.

## Keyfactor Command Claims Proxy

For implementations using an identity provider other than Active Directory, the claims proxy proxies authentication tokens between the Management Portal and the Keyfactor API to enable communications between the portal and the API. This application pool runs in the context of an Active Directory or local service account. A separate application pool is needed for this service, though the same service account may be used for both.

This role does not exist in implementations using Active Directory as an identity provider.

## Substitutable Text Token Query Access

Keyfactor Command supports the use of substitutable special text tokens in workflow and alert emails, conditions, and data manipulation steps to replace a variable with data from the certificate request, certificate, or certificate metadata at processing time. For tokens that relate to the certificate requester, this involves querying the identity store to retrieve information about the requester. If you're using Active Directory as an identity provider, this query is done in the context of the Keyfactor Command Management Portal application pool user (see [Keyfactor Command](#)



[Management Portal \(Application Pool\) on the previous page](#)), which must be running as an Active Directory user in that case. If you're using an identity provider other than Active Directory, queries cannot be completed to the identity provider for substitutable special text tokens and tokens requiring this are not supported.

## EJBCA End Entity for EJBCA CA Access

Keyfactor Command supports synchronization of certificates and certificate enrollment from EJBCA certificate authorities by configuring a client certificate issued from the EJBCA CA on the CA record in the Management Portal. This client certificate needs to be associated with an end entity in EJBCA that can be assigned sufficient permissions to perform all necessary CA tasks from Keyfactor Command.

## Explicit Credentials for Microsoft CA Access

Keyfactor Command supports synchronization of certificates and certificate enrollment from Microsoft certificate authorities in remote forests (forests other than the forest in which Keyfactor Command is installed which are not in a two-way trust with the Keyfactor Command forest) by configuring a service account from the forest in which the CA resides on the CA record in the Management Portal. All communication to retrieve existing certificates, enroll for new certificates, revoke certificates, and recover certificate keys from the remote CA is done in the context of this service account. Explicit credentials for remote CA access is configured in the Keyfactor Command Management Portal after installation is complete rather than in the configuration wizard.

You may need additional service accounts to support the use of Keyfactor Command orchestrators and/or gateways in your environment. Please see:

- *Create Service Accounts for the Universal Orchestrator* in the *Keyfactor Orchestrators Installation and Configuration Guide*
- *Create a Service Account for the Keyfactor Bash Orchestrator* in the *Keyfactor Orchestrators Installation and Configuration Guide*
- *Create Service Accounts for the Java Agent* in the *Keyfactor Orchestrators Installation and Configuration Guide*
- The installation guide for each gateway.

The service account(s) need to be created in Active Directory prior to installation of the Keyfactor Command software, and the person installing the Keyfactor Command software needs to know the service account(s) domain, username and password. The same service account may be used for multiple roles, if desired. For example, you might have one service account for orchestrators, another for gateways, and a third for all server roles.

Table 3: Typical Service Accounts

Account	Uses
Keyfactor Command Service Account	Keyfactor Command Service, Keyfactor Command Management Portal (Application Pool), Keyfactor API, Keyfactor Command Logi Report Access

Account	Uses
Keyfactor Orchestrator Service Account	Keyfactor Orchestrator access to Keyfactor Command Server and Keyfactor Orchestrator on-machine operations, where applicable

#### 2.4.2.6 Create Groups to Control Access to Keyfactor Command Features

Keyfactor Command uses groups to control access to the various Keyfactor Command features. The Keyfactor Command Management Portal supports multiple groups with different levels of access to the portal. During the installation, at least one group or user must be entered to grant full administrative access to the portal. After installation, additional groups can be configured through the Keyfactor Command Management Portal to grant more limited access to the portal.



**Important:** The built-in Active Directory groups Domain Admins and Enterprise Admins cannot be used directly to grant access to the Management Portal due to how these groups function within Windows. You can create a custom Active Directory group, reference that group in the Management Portal, and add the built-in Domain Admins or Enterprise Admins group to that custom group, if desired.

Groups that you may find it useful to identify or add following the initial installation include:

#### Keyfactor Command Enrollment

Users who are a member of this group or groups may use PFX and/or CSR enrollment through the Keyfactor Command Management Portal. Access control for enrollment is configured in the Keyfactor Command Management Portal after installation is complete.

#### Keyfactor Command My SSH Key

Users who are a member of this group or groups may acquire SSH keys through the Keyfactor Command My SSH Key portal. Access control for the My SSH Key portal is configured in the Keyfactor Command Management Portal after installation is complete.

#### Keyfactor Java Agents

Service accounts that are a member of this group are allowed to auto-register as Keyfactor Java Agents in Keyfactor Command, if auto-registration is configured, providing for more hands-free management of Java and PEM certificate stores. This group is not required if auto-registration with user validation will not be used.

#### Keyfactor Bash Orchestrators

Service accounts that are a member of this group are allowed to auto-register as Bash orchestrators in Keyfactor Command, if auto-registration is configured, providing for more hands-free management of Bash orchestrators. This group is not required if auto-registration with user

validation will not be used.

## Keyfactor Mac Auto-Enrollment Users

Users who are members of this group are allowed to auto-register for Mac auto-enrollment in Keyfactor Command, if auto-registration is configured, providing for more hands-free management of Mac auto-enrollment. The same group may be used to grant users permissions on the template that will be used for Mac auto-enrollment. This group is not required if auto-registration with user validation will not be used and a different group will be used to grant permission on the template.

## Keyfactor Universal Orchestrators

Service accounts that are a member of this group are allowed to auto-register as Keyfactor Universal Orchestrators in Keyfactor Command, if auto-registration is configured, providing for more hands-free management of certificate stores managed by the Keyfactor Universal Orchestrator. This group is not required if auto-registration with user validation will not be used.



**Note:** The same group may be used for multiple roles. Existing groups may be used. For example, if all employees of your organization are members of the Active Directory Domain Users group and you wish to allow all employees to acquire SSH keys, you may use the Domain Users group for the Keyfactor Command My SSH Key function.



**Tip:** If you're using Active Directory as an identity provider and wish to grant access in the Management Portal to users from trusted Active Directory forests, create a domain local group in the Active Directory domain in which Keyfactor Command is installed, put the cross-forest users and groups in this local group and grant access in Keyfactor Command to this domain local group.

### 2.4.2.7 Configure Certificate Chain Trusts for CAs

The Keyfactor Command server needs to trust the chain certificates for all the CAs you will reference within Keyfactor Command in order for all operations to complete successfully. In many environments, root and intermediate trusts for domain-joined Microsoft CAs are pushed out automatically. If this is not the case in your environment or if you are using non-domain-joined CAs (e.g. EJBCA CAs), you will need to configure these chain trusts on the Keyfactor Command server manually.

The certificate for each root CA must be installed in the Trusted Root Certification Authorities store under Local Computer on the Keyfactor Command server. If your public key infrastructure (PKI) also has issuing CAs, the issuing CA certificates must be installed in the Intermediate Certification Authorities store under Local Computer on the Keyfactor Command server.

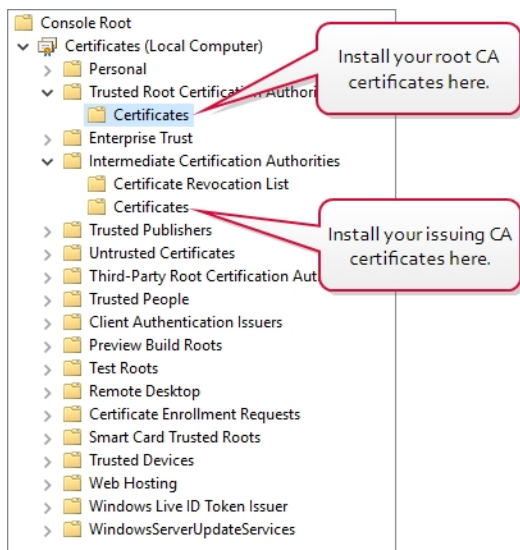


Figure 54: Install CA Chain Certificates on the Keyfactor Command Server

### 2.4.2.8 Hostname Identification and Resolution

Prior to the installation of Keyfactor Command, you need to determine the DNS alias(es) by which the Keyfactor Command roles will be accessed, if any, and configure them in your hostname resolution solution so that they will be resolvable prior to installation. For example, if you're licensed for SSH key management and wish to publish the My SSH Key portal externally to support SSH key acquisition by users outside the company firewall, you will probably wish to reference the server by a DNS alias rather than its actual hostname. For example, you may wish to use `keyfactor.keyexample.com` rather than `websrvr23.keyexample.local`. This is particularly significant if you will be using redundant servers with load balancing.

For DNS aliases used internally in environments using Active Directory as an identity provider, you will need to consider whether the servers to be accessed will be authenticated using Kerberos authentication. Out of the box, the Keyfactor Command Management Portal uses integrated Windows authentication and will default to Kerberos authentication in most environments. Although some features of the Keyfactor Command Management Portal may support NTLM authentication in some environments, the dashboard and enrollment functions do not support NTLM. If you will be using Kerberos authentication, your DNS aliases need to be configured as "A" records rather than CNAME records because Kerberos does not function well with CNAME records under Microsoft IIS.

The roles for which you need hostnames during the Keyfactor Command installation are:

#### SQL Server

For a small environment you may choose to use the server's actual name. If you plan to use SQL clustering, you will need an alias that represents the cluster. Using an alias for the SQL server allows for database portability in the future.

## Email

During the Keyfactor Command installation you configure the email server that will be used to send email notifications.

## Keyfactor Command Management Portal

This is the primary management server and may hold all Keyfactor Command roles in a small implementation.

## Keyfactor Command Logi Dashboard and Reports

This hostname must match the hostname entered for the Management Portal.

## Keyfactor Command Orchestrators API

This hostname is only required if your Keyfactor Command license includes orchestrator functionality. If all Keyfactor Command roles are combined on one server, this will be the same hostname as used for the Keyfactor Command Management Portal.

## Keyfactor Command Keyfactor API

This hostname must match the hostname entered for the Management Portal unless you are installing a secondary instance of the Keyfactor API.

## Centralized Logging Solution

This hostname is only required if you choose to enable the option to copy Keyfactor Command audit logs entries in real time, as they are generated, to a separate server for collection and analysis by a centralized logging solution (e.g. rsyslog, Logstash).

Prior to beginning the Keyfactor Command installation, ensure that the selected hostnames resolve successfully.

### 2.4.2.9 Firewall Considerations

Keyfactor Command needs to be able to communicate internally between the various Keyfactor Command components installed on different servers, if applicable, and to the SQL server, certificate authorities, centralized logging server (if applicable), your identity provider. If there are any firewalls in the environment that control internal traffic, these may need to be updated to allow the appropriate level of communication. [Table 4: Protocols Keyfactor Command Uses for Communication](#) shows each Keyfactor Command component and the protocols they use to communicate. In environments using Active Directory as an identity provider, all Keyfactor Command components require a healthy Active Directory environment with the ability to use Kerberos, LDAP, and DNS.

Table 4: Protocols Keyfactor Command Uses for Communication

Keyfactor Command Component	Protocols and Ports	Target
Keyfactor Command Management Portal	HTTP/HTTPS (TCP 80/443)	Client browser (e.g. Microsoft Edge)
Keyfactor Command Management Portal	HTTP/HTTPS (TCP 80/443)	Certificate revocation list (CRL) distribution points
Keyfactor Command Management Portal	HTTP/HTTPS (TCP 80/443)	EJBCA Certificate Authorities
Keyfactor Command Management Portal	RPC/DCOM (TCP 135 plus random high ports typically in the range 49152 – 65535)	Microsoft Certificate Authorities
Keyfactor Command Management Portal	RPC/DCOM (TCP 135 plus random high ports typically in the range 49152 – 65535)	Keyfactor vendor gateways to cloud CAs (e.g. Entrust, Symantec)
Keyfactor Command Management Portal	MS SQL (default TCP 1433)	SQL Server
Keyfactor Command Management Portal	Varies depending on the implemented solution (TCP 514 for rsyslog, TCP 5000 for Logstash are some standard defaults)	Centralized logging solution
Keyfactor Command	Active Directory (TCP/UDP 389)	Microsoft Active Directory queries
Keyfactor Command SSH Management	Active Directory Web Services (TCP 9389)	Microsoft Active Directory for group membership enumeration
All Orchestrators and Agents	HTTP/HTTPS (TCP 80/443)	Keyfactor Command Orchestrator API endpoint
Keyfactor Universal Orchestrator with Extension Relying on PowerShell Remoting and WinRM (IIS and Remote File Extensions)	PowerShell Remoting (default TCP 5985 and 5986)	Windows Servers to which certificate files will be distributed
Keyfactor Universal Orches-	Any configured for scanning	The SSL endpoint being

Keyfactor Command Component	Protocols and Ports	Target
trator (SSL Endpoint Management)		scanned by the SSL discovery or monitoring job
Keyfactor Universal Orchestrator with Extension Relying on HTTP/HTTPS (F5 and Citrix NetScaler Certificate Store Management)	HTTP/HTTPS (TCP 80/443)	F5 or NetScaler Devices
Keyfactor Universal Orchestrator (Remote Certificate Authority)	RPC/DCOM (TCP 135 plus random high ports typically in the range 49152 – 65535)	Microsoft Certificate Authorities
Keyfactor Bash Orchestrator	SSH (TCP 22 by default)	Remote control targets for SSH management
Keyfactor Gateways to Cloud CAs	HTTP/HTTPS (TCP 80/443)	Cloud providers (e.g. Entrust, Symantec)
Keyfactor Cloud Gateway	Active Directory Web Services (TCP 9389)	Microsoft Active Directory for group membership enumeration

#### 2.4.2.10 Acquire a Public Key Certificate for the Keyfactor Command Server

Keyfactor recommends using HTTPS to secure the channel between clients and the Keyfactor Command server(s). This requires at least one SSL certificate. You will need an SSL certificate or certificates for each of the hostnames you have identified (see [Hostname Identification and Resolution on page 74](#)).

Acquire the certificate(s) using the Fully Qualified Domain Name (FQDN) of the server or alias used for the Keyfactor Command server(s). For example:

```
keyfactor.keyexample.com
```

The certificate(s) may be installed on the Keyfactor Command server(s) prior to installation of the Keyfactor Command software or may be installed at the time of Keyfactor Command installation. See [Configure SSL for the Default Web Site on the Keyfactor Command Server on page 84](#) for more information.

If installed ahead of time, the certificate(s) should be placed in the Personal Certificate store of the Local Computer using the Certificates MMC Snap-In.

### 2.4.2.11 Install IIS and .NET on the Keyfactor Command Server

Internet Information Services (IIS) and .NET 4.7.2 or greater must be installed on the Keyfactor Command server(s) prior to installation of the Keyfactor Command software.

IIS is a standard Windows role added through the Windows Server Manager tool and .NET is a standard Windows feature added through the Windows Server Manager tool. You may need to update to .NET 4.7.2 or greater with a downloadable update package or through Windows update.



**Important:** IIS needs to be configured to allow requests using the HTTP verbs DELETE, GET, POST and PUT to reach the Default Web Site (or other web site if you choose to install to an alternate web site). These are enabled by default. To check whether any of these have been disabled, open the IIS Management console, drill down to highlight the Default Web Site, double-click **Request Filtering** in the center pane, and review the information on the **HTTP Verbs** tab.

To verify the version of .NET installed, either:

1. Open the Registry Editor:

```
regedit
```

2. Navigate to:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full
```

3. Validate that the **Release** attribute value indicates a version of .NET Framework that is 4.7.2 or higher is installed, as shown in [Table 5: .NET Framework Release Values](#).

Or:

1. Open a command prompt or PowerShell window and type the following command:

```
reg query "HKLM\Software\Microsoft\NET Framework Setup\NDP\v4\Full"
```

2. Validate that the **Release** attribute value indicates a version of .NET Framework that is 4.7.2 or higher is installed, as shown in [Table 5: .NET Framework Release Values](#).

Table 5: .NET Framework Release Values

.NET Framework	Release Value (Decimal)	Release Value (Hexadecimal)
.NET Framework 4.6.2	394802 or 394806	60632 or 60636
.NET Framework 4.7	460805	70805
.NET Framework 4.7.1	461308 or 461310	709FC or 709FE



.NET Framework	Release Value (Decimal)	Release Value (Hexadecimal)
.NET Framework 4.7.2	461808 or 461814	70BF0 or 70BF6
.NET Framework 4.8	528040, 528049, 528372, or 528449	80EA8, 80EB1, 80FF4, 81041

## Installing IIS and ASP.NET on Windows Server 2019 and 2022

The following figures show the components of IIS and ASP.NET necessary to support Keyfactor Command on Windows Server 2019 and 2022. Your Keyfactor Command server may have additional roles or features installed that are not shown in these figures.



**Important:** The ASP.NET Core Hosting Bundle that is also required (see [System Requirements on page 9](#)) should not be installed before installing IIS. If the hosting bundle is installed before IIS is installed, the bundle will not function correctly after the IIS install and will require repair.

Keyfactor Command makes use of the Active Directory tools for PowerShell to do group membership queries in Active Directory in some functions (e.g. when using a group to create a mapping between a Linux logon for SSH and one or more SSH keys). The *Active Directory module for Windows PowerShell* is installed as a feature as part of the *Remote Server Administrator Tools*.



**Important:** Do not install the IIS *WebDAV Publishing* feature. Keyfactor Command will not operate correctly if this feature is installed.

Note that it is possible to install IIS and the necessary features using PowerShell rather than the below-referenced GUI-based installation method. The correct PowerShell command for this is:

```
Install-WindowsFeature Web-Server, Web-Asp-Net45, Web-Default-Doc, Web-Dir-Browsing, Web-Http-Errors, Web-Static-Content, Web-Http-Logging, Web-Stat-Compression, Web-Filtering, Web-Basic-Auth, Web-Windows-Auth, Web-Net-Ext45, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Mgmt-Console, RSAT-AD-PowerShell
```



**Tip:** To check and see if all the required roles and features have been installed, use Get-WindowsFeature with the same list of roles and features like so:

```
Get-WindowsFeature Web-Server, Web-Asp-Net45, Web-Default-Doc, Web-Dir-Browsing, Web-Http-Errors, Web-Static-Content, Web-Http-Logging, Web-Stat-Compression, Web-Filtering, Web-Basic-Auth, Web-Windows-Auth, Web-Net-Ext45, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Mgmt-Console, RSAT-AD-PowerShell
```



Output from this command will look something like the following, which shows some required features installed and some missing. Make sure all roles and features in the query output are marked *Installed* before continuing.

```
Get-WindowsFeature Web-Server, Web-Asp-Net45, Web-Default-Doc, Web-Dir-Browsing, Web-Http-Errors, Web-Static-Content, Web-Http-Logging, Web-Stat-Compression, Web-Filtering, Web-Basic-Auth, Web-Windows-Auth, Web-Net-Ext45, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Mgmt-Console, RSAT-AD-PowerShell
```

Display Name	Name	Install State
[X] Web Server (IIS)	Web-Server	Installed
[X] Default Document		Installed
[X] Directory Browsing		Installed
[X] HTTP Errors		Installed
[X] Static Content		Installed
[X] HTTP Logging		Installed
[X] Static Content Compression	Web-Stat-Compression	Installed
[X] Request Filtering	Web-Filtering	Installed
[ ] Basic Authentication	Web-Basic-Auth	Available
[ ] Windows Authentication	Web-Windows-Auth	Available
[ ] .NET Extensibility 4.8	Web-Net-Ext45	Available
[ ] ASP.NET 4.8	Web-Asp-Net45	Available
[ ] ISAPI Extensions	Web-ISAPI-Ext	Available
[ ] ISAPI Filters	Web-ISAPI-Filter	Available
[X] IIS Management Console	Web-Mgmt-Console	Installed
[ ] Active Directory module for Windows ...	RSAT-AD-PowerShell	Available

Some of the required IIS roles and features are installed on this machine, but several are still missing.

Figure 55: Use `Get-WindowsFeature` to Determine if All Required Roles and Features are Installed

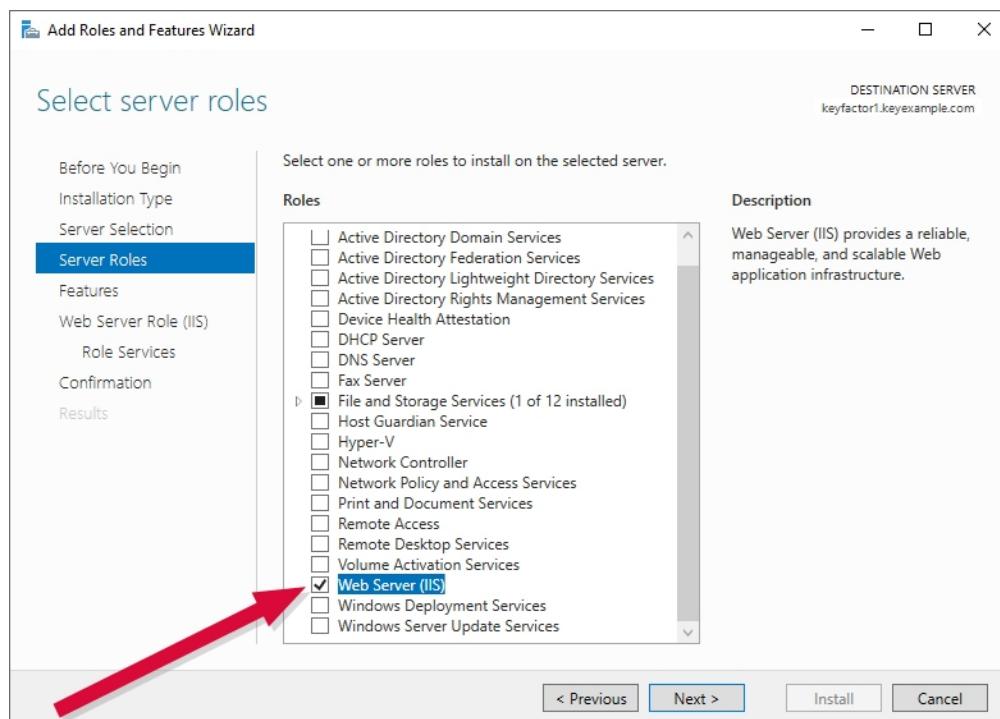


Figure 56: Web Server Role

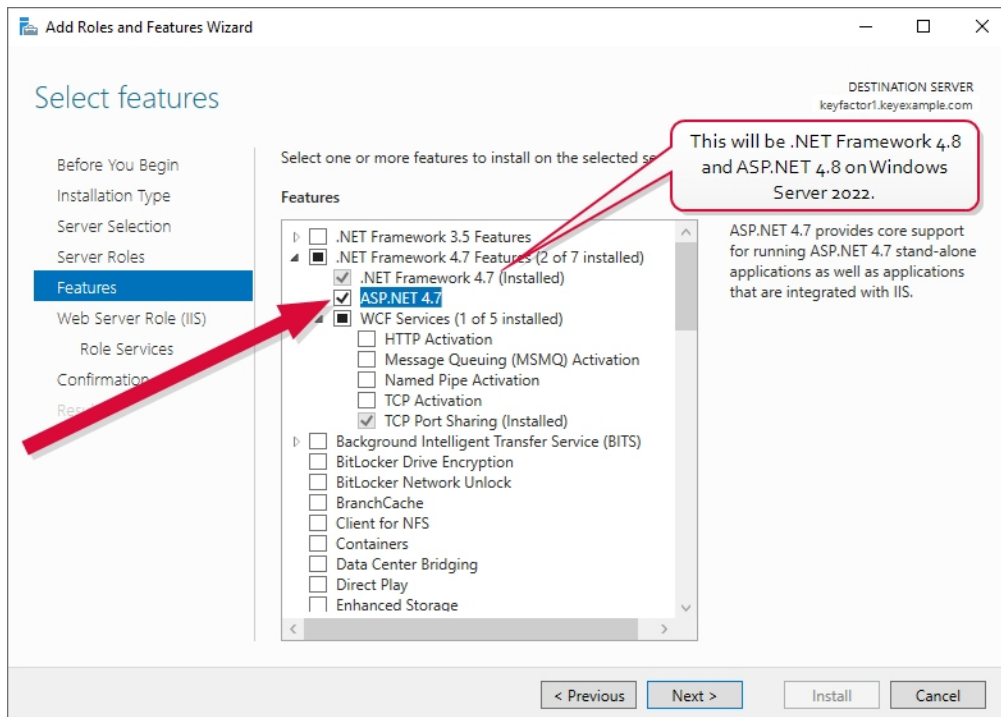


Figure 57: .NET 4.7 Feature

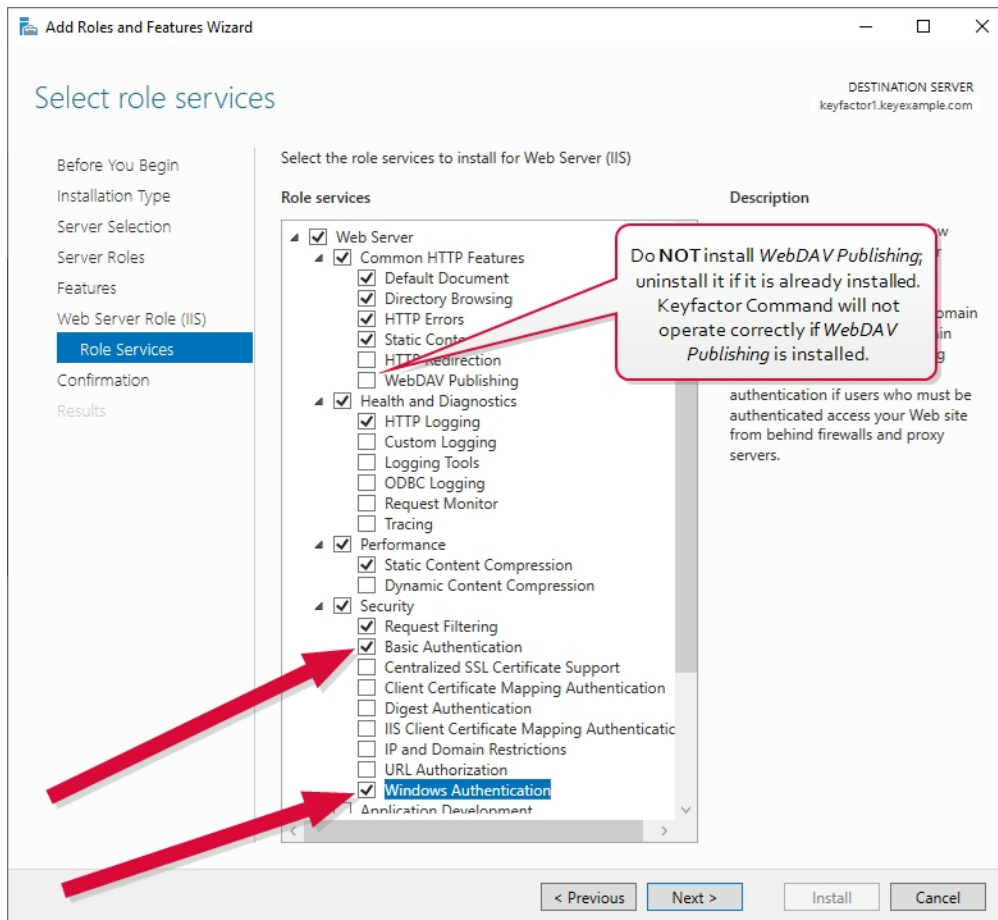


Figure 58: Role Services Page One

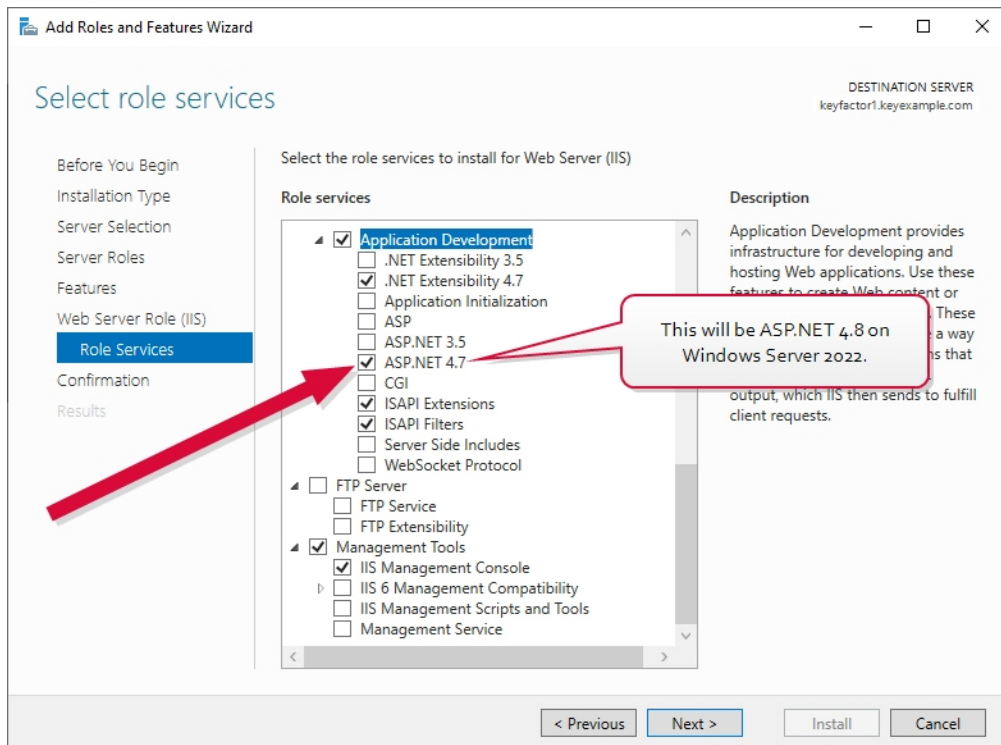


Figure 59: Role Services Page Two

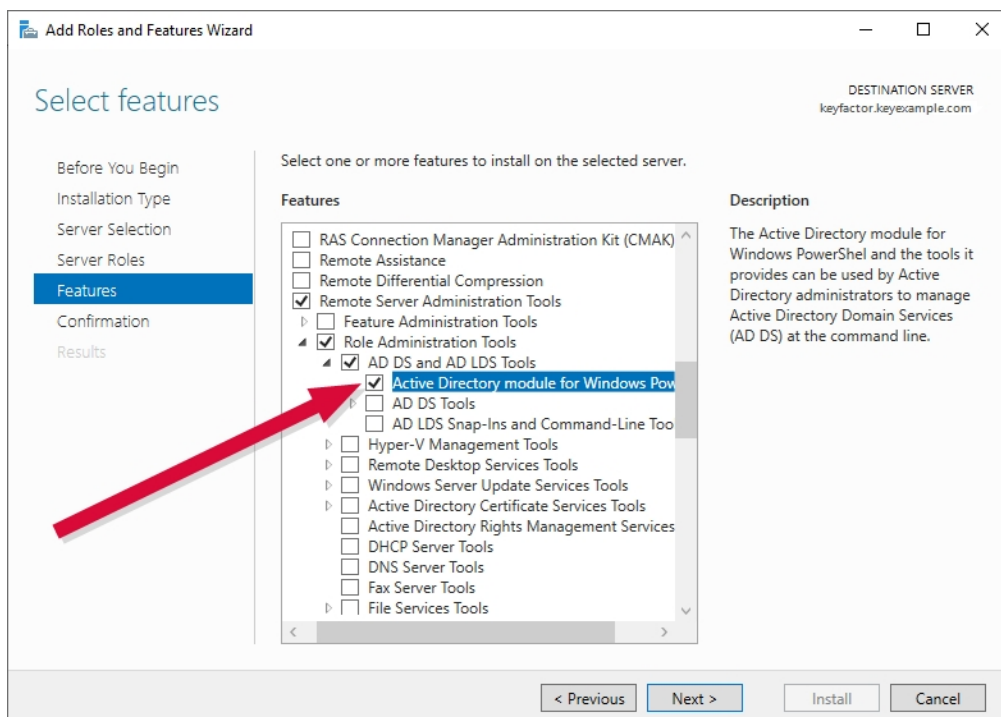


Figure 60: Active Directory Module for Windows PowerShell

#### 2.4.2.12 Configure SSL for the Default Web Site on the Keyfactor Command Server

Once you have acquired an SSL certificate for Keyfactor Command and installed IIS, you can open the IIS Management Console and associate the certificate with the Default Web Site. You can do this either before or after installing Keyfactor Command.

To import your SSL certificate and associate it with the Default Web Site:

1. Open the IIS Manager MMC snap-in.
2. Navigate to the connection for the current host. (The top level in IIS.)
3. On the current host Home page, open (double-click) Server Certificates. If your SSL certificate already appears in this list, you can skip steps 4-7.
4. On the Server Certificates page, select Import... under Actions.
5. In the Certificate file (.pfx) field, choose the browse option and navigate to the .pfx or .p12 file containing your certificate.
6. Enter the password for your certificate, select the Personal Certificate Store, check the Allow this certificate to be exported box if desired, and click **OK**.
7. Your certificate should now appear in the list of Server Certificates. Confirm that the Issued To column shows your certificate name correctly (e.g. keyfactor.keyexample.com).
8. Navigate to the Default Web Site and on the Default Web Site Home page, select Bindings... under Actions.
9. In the Site Bindings dialog, highlight the https entry if it exists and choose Edit. If an https entry does not exist, click **Add**.
10. In the Edit Site Bindings dialog, select https in the Type dropdown (this will already be selected and grayed out if you selected Edit in the previous step), select the certificate you just imported in the SSL certificate dropdown box, and click **OK**.

Note that these instructions assume that your SSL certificate has been provided in PKCS12 format file. If you are requesting a certificate directly from an on-premise CA through IIS or are generating a CSR through this IIS installation to submit to a CA, the configuration steps will be different.

#### 2.4.2.13 Configure the Keyfactor Command Server to Require SSL

For best security practice, the Keyfactor Command web site should be configured to require SSL for all access. To do this:

1. Open the IIS Manager MMC snap-in.
2. Navigate to the Default Web Site.
3. Under the Default Web Site Home, select **SSL Settings**.
4. On the SSL Settings page, check the **Require SSL** box and, under Actions, click **Apply**.



**Important:** The Keyfactor Command web application is not configured to support HTTP Strict Transport Security (HSTS) by default. HSTS is a web security policy mechanism that helps to protect websites against protocol downgrade attacks and cookie hijacking. An application enables HSTS by returning an HTTP response header that instructs users' browsers to only interact with the site using secure transport methods. HSTS is supported by all modern browsers. To accommodate this, configure the server to always send the *Strict-Transport-Security* HTTP header on HTTPS connections:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

The max-age parameter is given in seconds; the value shown above is equivalent to one year.

Instructions for implementation of HSTS are beyond the scope of this guide.

#### 2.4.2.14 Prepare for External Log Shipping over TLS (Optional)

Keyfactor Command offers the option to copy audit logs in real time to a separate server for collection and analysis with a centralized logging solution (e.g. rsyslog, Splunk, Elastic Stack). This can be done either over standard UDP/TCP connections, or you can opt to secure the connection to the backend log collection solution using TLS. This requires a backend solution that supports receiving logs over TLS and, typically, a client certificate on the Keyfactor Command server and a server certificate on the backend server.

The following instructions cover using rsyslog on the backend and will differ if you are using an alternative log collection solution.

Acquire the client certificate(s) using the Fully Qualified Domain Name (FQDN) of the server or alias used for the Keyfactor Command server(s) (see [Hostname Identification and Resolution on page 74](#)). For example:

```
keyfactor.keyexample.com
```

The certificate must be installed on the Keyfactor Command server(s) prior to installation of the Keyfactor Command software.

To acquire a client certificate for use in log shipping using a Microsoft CA, first create or identify a template that has an extended key usage (EKU) of *Client Authentication* and make it available for enrollment on a CA to which the Keyfactor Command server has access with enrollment permissions for the Keyfactor Command server. If desired, you can use a template that has both the *Client Authentication* and *Server Authentication* EKUs and use it for certificates on both sides of the communication. Start by copying a computer template if you want to enroll for the certificate using the Microsoft MMC as described below and without needing to set the private key of the certificate as exportable.

To enroll for a client certificate using the MMC:

1. On the Keyfactor Command server, do one of following:
  - Using the GUI:
    - a. Open an empty instance of the Microsoft Management Console (MMC).
    - b. Choose **File->Add/Remove Snap-in....**
    - c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.
    - d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.
    - e. Click **OK** to close the Add or Remove Snap-ins dialog.
  - Using the command line:
    - a. Open a command prompt using the “Run as administrator” option.
    - b. Within the command prompt type the following to open the certificates MMC:  
  
`certlm.msc`
2. Drill down to the Personal folder under **Certificates** for the Local Computer, right-click, and choose **All Tasks->Request New Certificate....**
3. Follow the certificate enrollment wizard, selecting the template you created or identified for use for this purpose, and providing any required information, being sure to set the CN to the FQDN of the Keyfactor Command server.



**Tip:** If you have an existing Keyfactor Command and wish to enroll through Keyfactor Command, you can request the certificate using the PFX enrollment option and either push it out to the Keyfactor Command local machine store using an IIS personal certificate store managed with a Keyfactor Universal Orchestrator installed on Windows with the IIS custom extension (see *Installing Custom-Built Extensions* in the *Keyfactor Orchestrators Installation and Configuration Guide*) as part of the enrollment process or import it to the certificate store using the PFX generated from Keyfactor Command.



**Note:** The Keyfactor Command server needs to be configured to trust the CA that issued the certificate to the rsyslog server. If you have opted to acquire certificates from a CA for which a root trust is not already configured on the Keyfactor Command server, this will need to be configured.

To acquire a server certificate for use in log shipping using a Microsoft CA, first create or identify a template that has an extended key usage (EKU) of *Server Authentication* and make it available for enrollment on a CA to which the server from which you are requesting the certificate has access with enrollment permissions for the server from which you are requesting the certificate.

To enroll for a server certificate using the MMC:



1. On a Windows server with appropriate enrollment access, do one of following:
  - Using the GUI:
    - a. Open an empty instance of the Microsoft Management Console (MMC).
    - b. Choose **File->Add/Remove Snap-in....**
    - c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.
    - d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.
    - e. Click **OK** to close the Add or Remove Snap-ins dialog.
  - Using the command line:
    - a. Open a command prompt using the “Run as administrator” option.
    - b. Within the command prompt type the following to open the certificates MMC:

```
certlm.msc
```

2. Drill down to Certificates in the Personal folder under **Certificates** for the Local Computer and locate your newly created certificate. Right-click on the certificate and choose **All Tasks->Export....**
3. Follow the certificate export wizard, being sure to answer **Yes, export the private key** and choosing the option to **Include all certificates in the certificate path if possible**. Set a password to secure the exported private key.
4. In the MMC, drill down to the Personal folder under **Certificates** for the Local Computer, right-click, and choose **All Tasks->Request New Certificate....**
5. Securely copy the resulting PFX file to your rsyslog server and place it in a temporary working directory.
6. Use openssl to break the PFX file apart into separate certificate and key files and remove the encryption on the key file (rsyslog does not provide a method for providing the password necessary to use the encrypted file) as follows:
  - a. Execute the following command to pull the key out of the PFX file (provide the input password for the PFX when prompted and the output password for the PEM key file when prompted):

```
openssl pkcs12 -in mycertfile.pfx -nocerts -out mykey.pem
```

- b. Execute the following command to pull the certificate out of the PFX file (provide the input password for the PFX when prompted):

```
openssl pkcs12 -in mycertfile.pfx -clcerts -nokeys -out mycert.pem
```

- c. Execute the following command to pull the chain certificate(s) out of the PFX file (provide the input password for the PFX when prompted):

```
openssl pkcs12 -in mycertfile.pfx -cacerts -nokeys -chain -out cacerts.pem
```

- d. Execute the following command to remove the encryption from the key so that a password will not be required when accessing the key file (provide the PEM key password you set in the first step):

```
openssl rsa -in mykey.pem -out mynewkey.key
```

7. Identify a secure location on the rsyslog server to store the certificates and key file (e.g. /etc/tls/certs and /etc/tls/private) and copy the certificates and key to these locations, setting appropriately secure permissions on the files. The key needs to be readable by the rsyslog daemon.



**Tip:** If you have an existing Keyfactor Command and wish to enroll through Keyfactor Command, you can request the certificate using the PFX enrollment option, opt to download it as a ZIP PEM, copy the zip file to the rsyslog server, unzip, and distribute the files as described in the final step, above.

Configuration of rsyslog for TLS support may vary depending on your needs. In addition to the standard rsyslog package for your Linux server, you will need GNU TLS packages to support TLS communication. For example, for Ubuntu the required packages are:

```
rsyslog
rsyslog-gnutls
gnutls-bin
```

The following is an example rsyslog.conf file configured for TLS support:

```
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf
#####
#### MODULES ####
#####

module(load="imuxsock") # provides support for local system logging
#module(load="immark") # provides --MARK-- message capability

# provides UDP syslog reception
```

```

#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")

#####
#### GLOBAL DIRECTIVES ####
#####

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

# Filter duplicated messages - It can be helpful to set this 'off' during initial
Keyfactor Command testing
$RepeatedMsgReduction off

#
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog

#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#

```

```

$IncludeConfig /etc/rsyslog.d/*.conf

#
# Configuration for TLS
#
$DefaultNetstreamDriver gtls

$DefaultNetstreamDriverCAFile /etc/tls/certs/cacerts.pem
$DefaultNetstreamDriverCertFile /etc/tls/certs/mycert.pem
$DefaultNetstreamDriverKeyFile /etc/tls/private/mynewkey.key

$ModLoad imtcp

$InputTCPServerKeepAlive on
$InputTCPServerStreamDriverAuthMode anon
$InputTCPServerStreamDriverMode 1 # run driver in TLS-only mode

$ActionSendStreamDriverAuthMode x509/name
$ActionSendStreamDriverMode 1 # run driver in TLS-only mode

$InputTCPServerRun 10514

```

A filter similar to the following can be used to redirect all Keyfactor Command related traffic to a particular file:

```
:syslogtag, isequal, "Keyfactor" /var/log/keyfactor/audit.log
```

## 2.4.3 Installing

The following installation instructions cover installing all Keyfactor Command server components on a single server performing all Keyfactor Command roles. You may choose to separate the roles onto different servers. If you do, the installation process will vary from the described process.

### 2.4.3.1 Install the Keyfactor Command Components on the Keyfactor Command Server(s)

Before you begin the installation, make sure that you have reviewed the system requirements (see [System Requirements on page 9](#)), completed the prerequisites (see [Planning & Preparing on page 11](#)), and have your Keyfactor Command license file ready to upload during the configuration.

The following installation steps show all possible Keyfactor Command features enabled. Your Keyfactor Command license may not cover all Keyfactor Command features. If it does not, uncensored features will not be shown in the configuration wizard. You may skip those configuration steps.

To begin the Keyfactor Command installation, execute the KeyfactorPlatform.msi file from the Keyfactor Command installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.

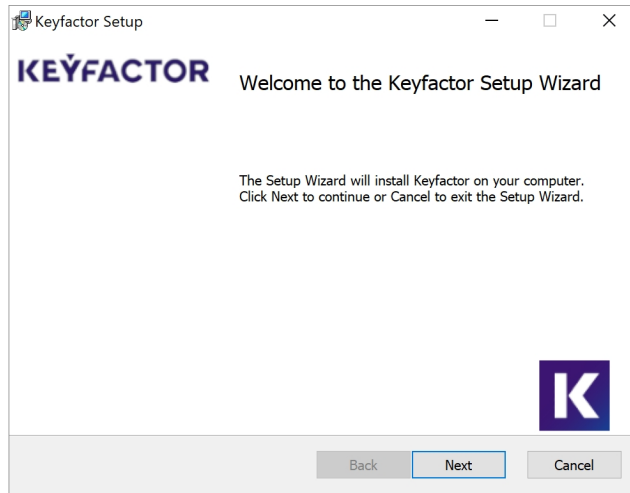


Figure 61: Install: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**. Click **Print** to review a printed copy if desired.
3. On the next page, select the components to install. For a server with the default roles collocated, leave the default options and click **Next** to continue. If desired, you can highlight Keyfactor Command and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor Platform\

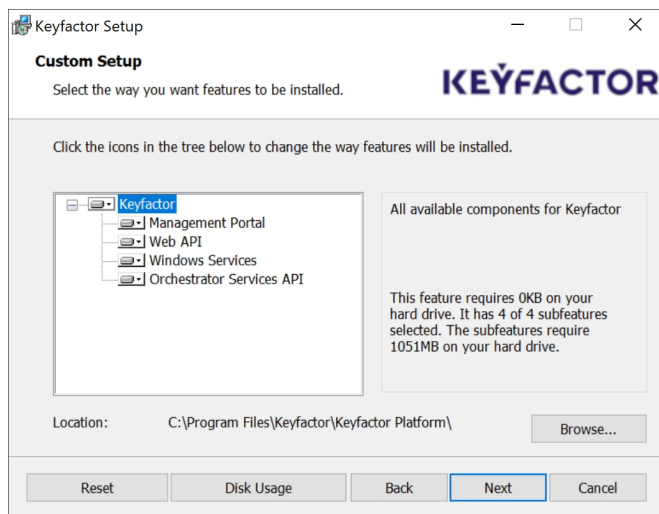


Figure 62: Install: Select Components


 **Tip:** Refer to [Keyfactor Command Server\(s\) on page 65](#) for a description of these components.

Table 6: Available components for Keyfactor.

Component	Description
<b>Management Portal</b>	<b>Mandatory.</b> Web-based management console for configuring all aspects of Keyfactor. The Keyfactor API will be installed with this component.
<b>Windows Services</b>	<b>Mandatory.</b> Includes the timer Windows service to manage timed events, such as CA Sync, PKI monitoring and system maintenance.
<b>Web API</b>	Optional. The Keyfactor API component. This allows the Keyfactor API for external use to be installed on a separate server from the Management Portal, if desired.
<b>Orchestrator Services API</b>	Optional. Not required if neither agents nor orchestrators will be utilized by Keyfactor Command. Web based orchestrator services API.

- On the next screen, click **Install**.
- On the final installation wizard page, leave the *Launch the Configuration Wizard now* box selected and click **Finish**. The configuration wizard should start automatically. This can take several seconds.
- On the Keyfactor Command Database Configuration page, enter the name, IP address, or fully qualified domain name (FQDN) of your SQL server and select a Credential Type of either

## Windows or SQL.



**Important:** Keyfactor Command uses an encrypted channel to connect to the SQL server by default, which requires configuration of an SSL certificate on the SQL server (see [Using SSL to Connect to SQL Server on page 54](#)). The name or IP address you enter here for your SQL server must be available as a SAN in this certificate unless you have disabled the encrypted connection for Keyfactor Command (see [Configurable SQL Connection Strings on page 58](#)).

- If you select **Windows** as the Credential Type for connecting to SQL, click the **Connect** button.

The dialog box titled 'Keyfactor Database Configuration' shows the 'Server' field with 'sqlsrvr05.keyexample.com'. Under 'Credential Type', the 'Windows' radio button is selected. The 'Database Name' field contains 'Command'. Buttons for 'Connect', 'Browse', 'Continue', and 'Close' are visible.

Figure 63: Windows Authentication

- If you select **SQL** as the Credential Type for connecting to SQL, the window will expand to include fields to enter a SQL username and password. Enter a username and password to authenticate to SQL, and click the **Connect** button.



**Note:** The password must not contain single or double quotes. An error will be shown if single or double quotes are used in the password.

The dialog box titled 'Keyfactor Database Configuration' shows the 'Server' field with 'sqlsrvr05.keyexample.com'. Under 'Credential Type', the 'SQL' radio button is selected. New fields for 'User' (containing 'john\_smith') and 'Password' (masked with dots) are present. The 'Database Name' field still contains 'Command'. Buttons for 'Connect', 'Browse', 'Continue', and 'Close' are visible.

Figure 64: SQL Authentication



**Note:** For the permissions required for this user, see [Grant Permissions in SQL on page 52](#).



**Note:** Keyfactor Command supports configuration of a base SQL connection template that is used for all connections Keyfactor Command makes to SQL. For more information, see [Configurable SQL Connection Strings on page 58](#).



**Note:** Your SQL server must be configured to support mixed mode authentication in order to use the SQL option.

7. After the **Connect** button is clicked, the database name field will be activated. You can either enter the name of the desired database—for either a new or existing database—or click **Browse** to scroll through a list of existing databases.



**Note:** On subsequent runs of the configuration wizard, the database name field will be pre-populated with the database name used on the last completed run. Any change to the server connection fields (server name, authentication type, etc.) will require the Connect button to be used again to unlock the database name field and the Continue button.

8. Click the **Continue** button. You will receive a confirmation dialog if any changes will be made to the database at this stage.



**Note:** If any of the following situations occurs, you will receive a message:

- The selected database does not exist and will be created.
- The selected database is empty and not associated with Keyfactor Command; it will be populated with the Keyfactor Command schema.
- The selected database does not match the current product schema and will be upgraded.
- The selected database is not empty and is not associated with Keyfactor Command.
- The user does not have access to the database.
- An SSL certificate is not correctly configured on the SQL server.

9. On the Keyfactor Command Encryption Warning page, read and understand the warning. Make note of the referenced documents to provide to your SQL team. Take advantage of the option to make a backup of the Database Master Key (DMK) by entering a path to a directory on your SQL server along with a filename for the backup file and a password to encrypt the file and clicking **Backup**. The user running the Keyfactor Command installer must have write permissions to this directory. Click **Continue**.



**Important:** Keyfactor Command uses Microsoft SQL Server encryption to protect security sensitive data, including service account credentials. Backup of the SQL server Database Master Key (DMK) is of critical importance in database backup and recovery





operations. The backup file of the DMK and the password should be stored in a safe, well-documented location. Without the file and password created with this process, some data that is encrypted within the Keyfactor Command database will be unrecoverable in a disaster recovery scenario. For more information, see *SQL Encryption Key Backup* in the *Keyfactor Command Reference Guide*.

If you choose to install Keyfactor Command in the default location, the referenced documents can later be found here:

C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\DMKBackup.docx  
C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\DMKRestore.docx

**K** Keyfactor Encryption Warning

Keyfactor uses Microsoft SQL Server Encryption to encrypt portions of the database to protect secret data, including service account credentials. The Database Master Key (DMK) for the Keyfactor database should be backed up for disaster recovery purposes. Failing to back up the DMK will require manual re-entry of any secret data into Keyfactor in the event the Keyfactor database needs to be restored from backups.

[MS SQL Server Encryption Hierarchy Explained](#)  
[Guide for backing up the DMK for your SQL instance \(document\)](#)  
[Guide for restoring the DMK during a DR scenario \(document\)](#)

To back up the DMK now, enter a file path and name (local to the SQL server instance) and a password to encrypt the file.

File name:

Password:

Confirm:

Figure 65: Configure: Backup Database Master Key

10. On the Keyfactor Command License upload page, click **Upload** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE. Once the uploaded license shows as valid, click **Continue**.

**K** Keyfactor License Upload

Select a Keyfactor license file to upload. The license should have been provided to you by Keyfactor prior to this installation, and should have a .cmslicense extension.

Choose a Keyfactor license file to upload

Figure 66: Configure: Upload License

11. In the Keyfactor Command configuration wizard, you can choose to upload a configuration file to populate the fields. You may have a file saved from a previous run of the configuration wizard or you may be provided one by Keyfactor. To upload a file, in the configuration wizard, click **File** at the top of the wizard and choose **Open Data File**. Browse to locate the configuration file. Configuration files have an extension of .cmscfg. The file may be protected with a password. If it is, you will need to provide this password to open the file. Continue with the remainder of the steps, reviewing the tabs to assure that the data is complete and correct.



**Note:** If you open a configuration file that contains configuration information for an identity provider other than Active Directory but does not set OAuth enabled to true, the additional identity provider information will not be loaded.

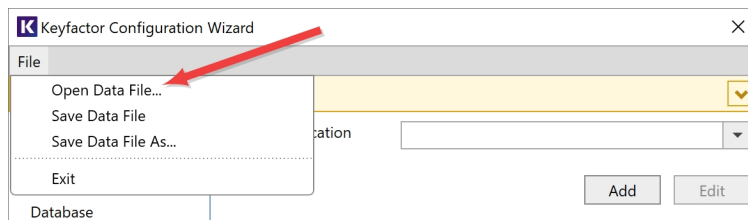


Figure 67: Configure: Open Data File




**Note:** At the bottom of the configuration wizard, if the database server name is longer than will fit in the provided window, it will be truncated and an ellipsis will be added.

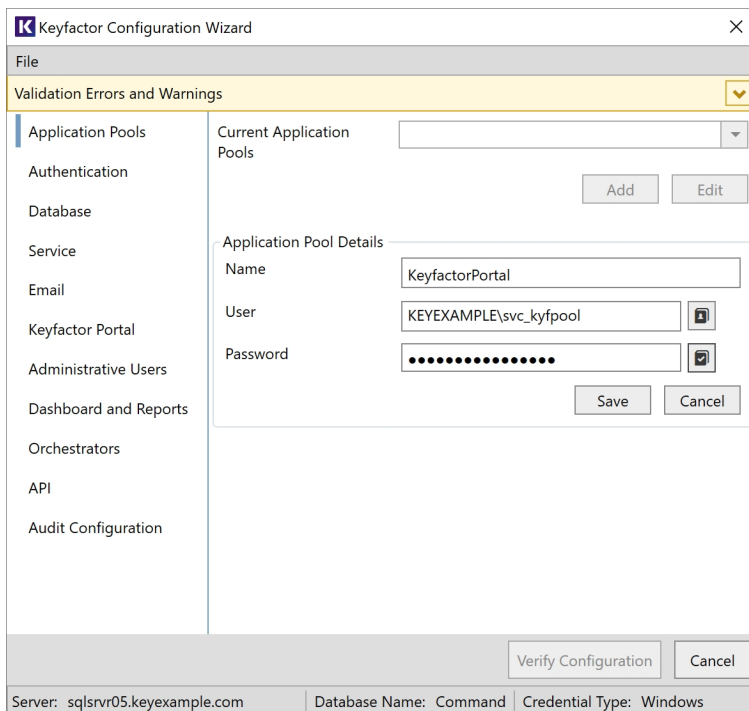
## 12. Application Pools Tab

A separate application pool is required for each virtual directory that will be created for Keyfactor Command in IIS. If you've chosen to install all the Keyfactor Command components, this will be five application pools for the virtual directories with the following names, by default:

- KeyfactorAgents (Keyfactor Command agent and orchestrator service endpoint)
- KeyfactorAnalysis (Keyfactor Command dashboard and reporting)
- KeyfactorAPI (Keyfactor API)
- KeyfactorProxy (Authentication for an identity provider other than Active Directory)
- KeyfactorPortal (Keyfactor Command Management Portal)

On the Application Pools tab of the configuration wizard, click **Add**, change the default application pool name, if desired, and enter the user name (DOMAIN\username format) and password of the Active Directory service account under which the application pool will run. You may use the people picker button (👤) to browse for the account. Click the verify button (🔍) to confirm that the username and password entered are valid. Assuming the verification completes successfully, click **Save**.


 **Tip:** The same service account may be used for all application pools.




The image shows the 'Keyfactor Configuration Wizard' window, specifically the 'Application Pools' tab. The window has a menu bar with 'File' and a 'Validation Errors and Warnings' section with a dropdown arrow. On the left is a navigation pane with the following items: Application Pools (selected), Authentication, Database, Service, Email, Keyfactor Portal, Administrative Users, Dashboard and Reports, Orchestrators, API, and Audit Configuration. The main area is titled 'Current Application Pools' and contains a list box (currently empty) with 'Add' and 'Edit' buttons. Below this is the 'Application Pool Details' section with fields for 'Name' (containing 'KeyfactorPortal'), 'User' (containing 'KEYEXAMPLE\svc\_kyfpool' with a selection icon), and 'Password' (masked with dots and a selection icon). 'Save' and 'Cancel' buttons are at the bottom of this section. At the bottom of the wizard are 'Verify Configuration' and 'Cancel' buttons. The status bar at the very bottom shows 'Server: sqlsrvr05.keyexample.com', 'Database Name: Command', and 'Credential Type: Windows'.

Figure 68: Configure: Application Pools

### 13. Authentication Tab

 **Important:** Before migrating an existing implementation from Active Directory to OAuth or vice versa, please see *Migrating to a New Identity Provider* in the *Keyfactor Command Reference Guide*.

On the Authentication tab of the configuration wizard, check the **Use OAuth** box if you wish to use an identity provider other than Active Directory as either your primary identity provider or a federation gateway to another identity provider. If you do not select this, you will be using the default identity provider of Microsoft Active Directory. If you checked OAuth, configure it as follows.

 **Important:** Only one identity provider may be configured at a time on each Keyfactor Command instance.

On the Authentication tab in the top section, accept the defaults for the **Session Expiration** and **Cookie Expiration** or modify these if appropriate for your environment. The *Session Expiration* value determines the length of time a browser session in the Keyfactor Command Management Portal will remain logged in before the user is prompted to re-authenticate regardless of whether the session is idle or in active use. The *Cookie Expiration* value determines the length of time the authentication cookie for the Keyfactor Command Management Portal browser session is considered valid. After half of the setting's duration, Keyfactor Command will attempt to use a refresh token to update the cookie. If this fails, the user's session will be terminated. The cookie renewal is seamless from the user's perspective (there is no prompt for credentials).



**Note:** For Keyfactor Identity Provider, these values should match those configured for the *SSO Session Max* and *Access Token Lifespan* in Keyfactor Identity Provider (see [Configuring Keyfactor Identity Provider and Collecting Data for the Keyfactor Command Installation on page 25](#)). If you've opted not to issue refresh tokens in Keyfactor Identity Provider, the **Cookie Expiration** value should match the **Session Expiration** value.

## Claims Proxy Section


On the Authentication tab in the Claims Proxy section, enter the FQDN that you will use to access the Keyfactor CommandManagement Portal in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Administration component or a DNS alias pointing to the server. If you have multiple Keyfactor CommandManagement Portal servers with load balancing, this will be a DNS name pointing to your load balancer. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and select the application pool for this virtual directory that you created earlier in the **Application Pool** dropdown.



**Note:** When you install Keyfactor Command using an identity provider other than Active Directory, a virtual directory called, by default, KeyfactorProxy is created in IIS for OAuth authentication. If you later disable OAuth, this virtual directory will still exist. This virtual directory is not used if you opt to use Active Directory as an identity provider. During Keyfactor Command uninstallation it will be removed, or you may opt to remove it manually if you are switching from OAuth to Active Directory for authentication.

The screenshot shows the 'Keyfactor Configuration Wizard' window. The left sidebar lists various configuration sections: Application Pools, Authentication, Database, Service, Email, Keyfactor Portal, Administrative Users, Dashboard and Reports, Orchestrators, API, and Audit Configuration. The 'Authentication' section is selected. The main pane shows the 'Claims Proxy' tab. It includes a 'Use OAuth' checkbox (checked), 'Session Expiration' (60), and 'Cookie Expiration' (5). Below these are fields for 'Host Name' (keyfactor.keyexample.com), 'Web Site' (Default Web Site), 'Virtual Directory' (KeyfactorProxy), and 'Application Pool' (KeyfactorProxy). The 'Identity Provider' section is also visible, showing 'Authentication Scheme' (Command-OIDC), 'Display Name' (Command-OIDC), and 'Type' (Generic). At the bottom, there are 'Verify Configuration' and 'Cancel' buttons, and a status bar showing 'Server: sqlsrvr05.keyexample.com', 'Database Name: Command', and 'Credential Type: Windows'.

Figure 69: Configure: Identity Providers—OAuth Claims Proxy Section

 **Tip:** There is no *Use SSL* check box for this component because SSL is required.

## Identity Provider Section

On the Authentication tab in the Identity Provider section, enter an **Authentication Scheme** and **Display Name** for your identity provider. The Authentication Scheme should be entered without spaces. This is used in constructing URLs that reference the identity provider from Keyfactor Command. In the **Type** dropdown, select an appropriate type for your identity provider. Most identity providers can be supported with the *Generic* type. For Auth0, select the *Auth0* type.

For Keyfactor Identity Provider, the Authentication Scheme you enter here must match the name you used when configuring the redirect URIs for Keyfactor Identity Provider (see [Configuring Keyfactor Identity Provider and Collecting Data for the Keyfactor Command Installation on page 25](#)).

Populate the identity provider parameters according to [Table 7: Identity Provider Parameters](#) and click **Save**. The fields that appear will vary depending on the selected *Type*. HTTPS is required for URL parameters. For more information about identity providers, see [Selecting an Identity Provider for Keyfactor Command on page 11](#).

Keyfactor Configuration Wizard

File

Validation Errors and Warnings

Application Pools

Authentication

Database

Service

Email

Keyfactor Portal

Administrative Users

Dashboard and Reports

Orchestrators

API

Audit Configuration

Identity Provider Parameters

Authentication Scheme

Command-OIDC

Display Name

Command-OIDC

Type

Generic

Timeout

60

Audience

Command-OIDC-Client

Scope

Claim Types

Name Claim Type

preferred\_username

Unique Claim Type

sub

Fallback Unique Claim Type

cid

Role Claim Type

groups

Verify Configuration

Cancel

Server: sqlsrvr05.keyexample.com

Database Name: Command

Credential Type: Windows

Keyfactor Configuration Wizard

File

Validation Errors and Warnings

Application Pools

Authentication

Database

Service

Email

Keyfactor Portal

Administrative Users

Dashboard and Reports

Orchestrators

API

Audit Configuration

OIDC Client Credentials

Client Id

Command-OIDC-Client

Client Secret

.....

Discovery Endpoint

Discovery Document Endpoint

https://appsrvr186.keyexample.com

Fetch

Clear

Authorization Endpoint

https://appsrvr186.keyexample.com

Token Endpoint

https://appsrvr186.keyexample.com

JSON Web Key Set Uri

https://appsrvr186.keyexample.com

Authority

https://appsrvr186.keyexample.com

User Info Endpoint

https://appsrvr186.keyexample.com

Command Querying Client Credentials

Verify Configuration

Cancel

Server: sqlsrvr05.keyexample.com

Database Name: Command

Credential Type: Windows

KEYFACTOR

11.3 Keyfactor Command Server Installation Guide

100

Figure 70: Configure: Identity Providers—OAuth Identity Provider Section




**Note:** If you return to the configuration wizard and re-run it to add a new identity provider or change the identity provider that's in active use for the system (disable one and enable another), you should restart the web server services (run an `iisreset`) after making the change to clear any cached data. If you're using more than one Keyfactor Command server in a cluster configuration, the web server services should be restarted on all of them.


Table 7: Identity Provider Parameters

Name	Type	Example	Description
Admin Querying Client Id	1 - String	Command- API-Query	The client ID of the service account that Keyfactor Command uses to make API calls to the identity provider.  For Keyfactor Identity Provider, this is created as a client (see <i>Using Keyfactor Identity Provider: Service Accounts</i> in the <i>Keyfactor Command Server Installation Guide</i> ). Keyfactor recommends that you

Name	Type	Example	Description
			<p>use a different client for this purpose than the client used for the main connection from Keyfactor Command to the identity provider (see <i>Client Id</i>).</p> <p>This parameter is required.</p>
Admin Querying Client Secret	1 - String		<p>The client secret of the service account that Keyfactor Command uses to make API calls to the identity provider.</p> <p>For Keyfactor Identity Provider, this is created as a client (see <i>Using Keyfactor Identity Provider: Service Accounts</i> in the <i>Keyfactor Command Server Installation Guide</i>).</p> <p>This parameter is required.</p>
Audience	1 - String	Command- OIDC- Client	<p>The audience value for the identity provider.</p> <p>For Keyfactor Identity Provider, this should be set to the same value as the <i>Client Id</i>. For example:</p> <div>Command- OIDC- Client</div> <p>This parameter is required.</p>
Auth0 API URL	1 - String		<p>The unique identifier defined in Auth0 or a similar identity provider for the API.</p> <p>This parameter only appears if <i>Auth0</i> is selected as the type and is required in that case.</p>
Authority	1 - String	https:// my-keyidp-server .keyexample.com /realms /Keyfactor	<p>The issuer/authority endpoint URL for the identity provider.</p> <p>For Keyfactor Identity Provider, this is included among the information that can be found on the OpenID Endpoint Configuration page, a link to which can be found on the Realm Settings page (see <i>Gathering Keyfactor Identity Provider Data</i> for</p>



Name	Type	Example	Description
			<p>the <i>Keyfactor Command Installation</i> in the <i>Keyfactor Command Server Installation Guide</i>). It is automatically returned by the <i>Discovery Document Endpoint Fetch</i> step. Review it to confirm that it appears correct.</p> <p>This parameter is required.</p> <div>  <p><b>Tip:</b> When you add or update an identity provider, the provider's discovery document is validated based on this authority URL. The discovery document is also validated periodically in the background. The following are validated:</p> <ul style="list-style-type: none"> <li>• That the discovery document is reachable using the Authority value provided and can be parsed into a valid discovery document.</li> <li>• That the Authority URL matches the Issuer returned in the discovery document.</li> <li>• That all the URLs on the discovery document are using HTTPS.</li> <li>• That the JSONWebKeySetUri value is included on the discovery document.</li> <li>• That any endpoint configuration values (Authorization Endpoint, Token Endpoint, UserInfo Endpoint, JSONWebKeySetUri) that have been saved or are being saved match—including case—the values returned in the</li> </ul> </div>

Name	Type	Example	Description
			 discovery document. The User-Info Endpoint is not a required configuration field, but if a value is provided, it must match what's in the discovery document.  If any of these validation tests fail, any identity provider changes in process will not be saved and an error will be displayed or logged.
Authorization Endpoint	1 - String	https:// my-keyidp-server .keyexample.com /realms /Keyfactor /protocol /openid-connect /auth	<p>The authorization endpoint URL for the identity provider.</p> <p>For Keyfactor Identity Provider, this is included among the information that can be found on the OpenID Endpoint Configuration page, a link to which can be found on the Realm Settings page (see <i>Gathering Keyfactor Identity Provider Data for the Keyfactor Command Installation</i> in the <i>Keyfactor Command Server Installation Guide</i>). It is automatically returned by the <i>Discovery Document Endpoint Fetch</i> step. Review it to confirm that it appears correct.</p> <p>This parameter is required.</p>
Client Id	1 - String	Command- OIDC- Client	<p>The ID of the client application created in the identity provider for primary application use.</p> <p>For Keyfactor Identity Provider, this should be:</p> <div>Command- OIDC- Client</div> <p>For more information, see <i>Gathering Keyfactor Identity Provider Data for the Keyfactor Command Installation</i> in the <i>Keyfactor Command Server Installation</i></p>

Name	Type	Example	Description
			<p>Guide).</p> <p>This parameter is required.</p>
Client Secret	2 - Secret		<p>The secret for the client application created in the identity provider for primary application use.</p> <p>For Keyfactor Identity Provider, see <i>Gathering Keyfactor Identity Provider Data for the Keyfactor Command Installation</i> in the <i>Keyfactor Command Server Installation Guide</i> for help locating this. It is automatically returned by the <i>Discovery Document Endpoint Fetch</i> step. Review it to confirm that it appears correct.</p> <p>Supported methods to store secret information are:</p> <ul style="list-style-type: none"> <li>Store the secret information in the Keyfactor secrets table.</li> </ul> <p>A Keyfactor secret is a user-defined username or password that is encrypted and stored securely in the Keyfactor Command database.</p> <ul style="list-style-type: none"> <li>Load the secret information from a PAM provider.</li> </ul> <p>See <i>Privileged Access Management (PAM)</i> in the <i>Keyfactor Command Reference Guide</i> for more information.</p> <p>This parameter is required.</p>
Discovery Document Endpoint	1 - String	https:// my-keyidp-server .keyexample.com /realms /Keyfactor /.well-known /openid-configuration	<p>The discovery URL for the identity provider.</p> <p>For Keyfactor Identity Provider, this is the link to the OpenID Endpoint Configuration page, which can be found on the Realm Settings page (see <i>Gathering Keyfactor Identity Provider Data for the Keyfactor Command Installation</i> in the <i>Keyfactor Command Server Installation Guide</i>). Populate this value and click <b>Fetch</b> to</p>

Name	Type	Example	Description
			<p>populate the remainder of the fields in this section, if desired.</p> <p>If you opt not to populate this field or if the discovery document does not return a valid response, the remainder of the fields in this section of the configuration will need to be configured manually. This value is not stored in the database.</p>
Fallback Unique Claim Type	1 - String	cid	<p>A backup value used to reference the type of claim used for users in the identity provider in case the primary referenced name does not contain a value. This parameter is required.</p>
JSON Web Key Set Uri	1 - String	https:// my-keyidp-server .keyexample.com /realms /Keyfactor /protocol /openid-connect /certs	<p>The JWKS (JSON Web Key Set) URL for the identity provider.</p> <p>For Keyfactor Identity Provider, this is included among the information that can be found on the OpenID Endpoint Configuration page, a link to which can be found on the Realm Settings page (see <i>Gathering Keyfactor Identity Provider Data for the Keyfactor Command Installation</i> in the <i>Keyfactor Command Server Installation Guide</i>). It is automatically returned by the <i>Discovery Document Endpoint Fetch</i> step. Review it to confirm that it appears correct.</p> <p>This parameter is required.</p>
Name Claim Type	1 - String	preferred_ user-name	<p>The name used to reference the type of user claim for the identity provider.</p> <p>For Keyfactor Identity Provider, this should be:</p> <div>preferred_ username</div> <p>This parameter is required.</p>
Role Claim Type	1 - String	groups	<p>The value used to reference the type of</p>

Name	Type	Example	Description
			<p>group claim for the identity provider. For Keyfactor Identity Provider, this should be:</p> <div>groups</div> <p>This parameter is required.</p>
Scope	1 - String		<p>One or more scopes that are requested during the OIDC protocol when Keyfactor Command is the relying party. Multiple scopes should be separated by spaces. This value is not used for Keyfactor Identity Provider.</p>
Sign Out URL	1 -String	https:// my-auth0-instance .us.auth0.com /oidc/logout	<p>The signout URL for the identity provider. This parameter only appears if <i>Auth0</i> is selected as the type and is required in that case.</p>
Timeout	1 - String	60	<p>The number of seconds a request to the identity provider is allowed to process before timing out with an error.</p>
Token Audience	1 - String		<p>An audience value to be included in token requests delivered to the identity provider when making a token request where Keyfactor Command is acting as the OAuth client. This value is not used for Keyfactor Identity Provider.</p>
Token Endpoint	1 - String	https:// my-keyidp-server .keyexample.com /realms /Keyfactor /protocol /openid-connect /token	<p>The token endpoint URL for the identity provider. For Keyfactor Identity Provider, this is included among the information that can be found on the OpenID Endpoint Configuration page, a link to which can be found on the Realm Settings page (see <i>Gathering Keyfactor Identity Provider Data for the Keyfactor Command Installation</i> in the</p>

Name	Type	Example	Description
			<p><i>Keyfactor Command Server Installation Guide</i>). It is automatically returned by the <i>Discovery Document Endpoint Fetch</i> step. Review it to confirm that it appears correct.</p> <p>This parameter is required.</p>
Token Scope	1 - String		<p>One or more scopes that should be included in token requests delivered to the identity provider when making a token request where Keyfactor Command is acting as the OAuth client. Multiple scopes should be separated by spaces.</p> <p>This value is not used for Keyfactor Identity Provider.</p>
Unique Claim Type	1 - String	sub	<p>The value used to reference the type of claim used for users in the identity provider. For Keyfactor Identity Provider, this should be (for subject):</p> <div>sub</div> <p>This parameter is required.</p>
User Info Endpoint	1 - String	https:// my-keyidp-server .keyexample.com /realms /Keyfactor /protocol /openid-connect /certs	<p>The user info endpoint URL for the identity provider.</p> <p>For Keyfactor Identity Provider, this is included among the information that can be found on the OpenID Endpoint Configuration page, a link to which can be found on the Realm Settings page (see <i>Gathering Keyfactor Identity Provider Data for the Keyfactor Command Installation</i> in the <i>Keyfactor Command Server Installation Guide</i>). It is automatically returned by the <i>Discovery Document Endpoint Fetch</i> step. Review it to confirm that it appears correct.</p>

## 14. Database Tab

On the Database tab in the top section, select an Authentication Mode for ongoing communications to SQL server—**Windows Authentication** or **SQL Server Authentications**. Your SQL server must be configured to support mixed mode authentication in order to use the SQL server authentication option.

- If you select Windows Authentication, login(s) will be created in SQL for the application pool user(s) you created on the Application Pools tab and granted appropriate permissions (other than the application pool for the Logi Analytics Platform, which does not need database access).
- If you choose SQL server authentication, enter a **User** and **Password** of a SQL administrator for the Keyfactor Command SQL database. If the user does not exist in SQL, it will be created and granted the necessary permissions for management of the Keyfactor Command database. If the user already exists in SQL, it will be granted the necessary permissions. If the database you originally connected to is an Azure database, **SQL Server Authentication** is the only option provided.

For more information, see [Grant Permissions in SQL on page 52](#).

If desired, check the **Configure Encryption** box. This option allows you to encrypt select sensitive data stored in the Keyfactor Command database using a separate encryption methodology utilizing a Keyfactor Command-defined certificate on top of the SQL server encryption noted above. This additional layer of encryption protects the data in cases where the SQL Server master keys cannot be adequately protected. Read and understand the encryption warning. This warning applies to implementations with more than one Keyfactor Command server.



**Note:** In an environment where there are multiple copies of Keyfactor Command pointing to the same database, each server running a Keyfactor Command instance will need to have the same encryption certificate AND the corresponding private key.

Select **Application and SQL** for the **Encryption Type** and click the **Select** button to choose a certificate from the Personal Certificate store of the Local Computer with which to encrypt the data. Only valid certificates with the appropriate key usage will appear in the selection dialog. See [Acquire a Public Key Certificate for the Keyfactor Command Server on page 77](#).

If you enable application-level encryption, your certificate must either be using a key storage provider (KSP) or you must manually grant permissions to the certificate's private key (see [Application-Level Encryption on page 60](#)).



**Tip:** If you need to reset the encryption level to remove application-level encryption, run the configuration wizard again and select the **SQL Only** option. You must ensure that the server you are re-running the configuration wizard on has both the certificate used for application-level encryption and its associated private key. When Keyfactor Command notices that application-level encryption has been disabled, it will process all the secrets



in the database and remove the additional encryption. The data will then be re-saved to the secrets table using only SQL-level encryption.

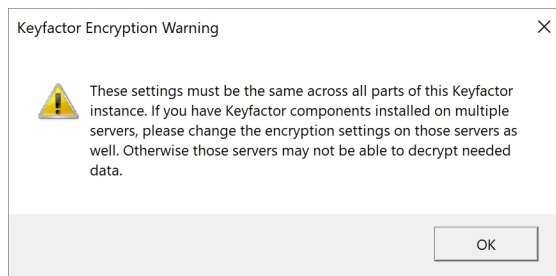


Figure 71: Configure: Encryption Warning

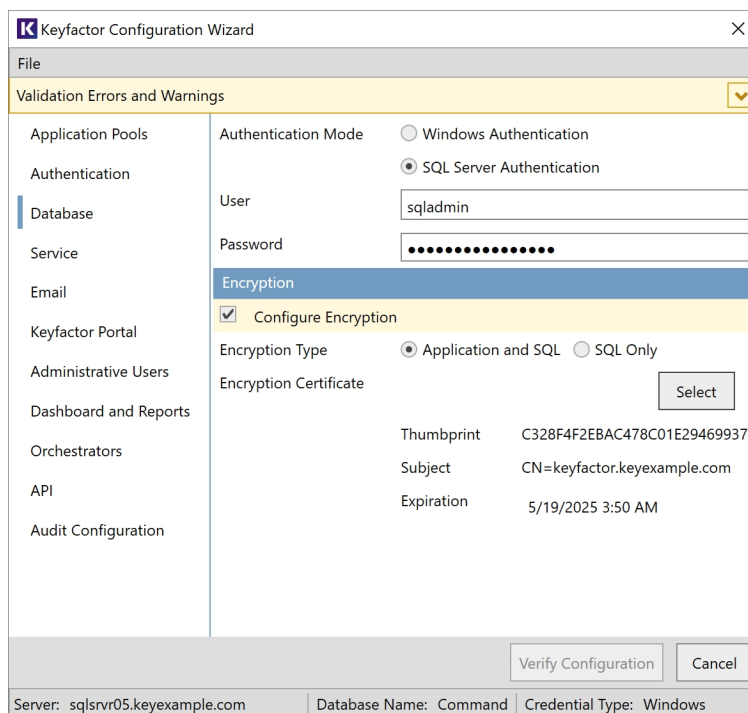


Figure 72: Configure: Database

## 15. Service Tab

On the Service tab, enter the user name and password of the Active Directory (DOMAIN\user-name format) or local (HOSTNAME\username format) service account under which the Keyfactor Command Service will run. This can be the same service account used for the application pool(s) or a different service account. You may use the people picker button (👤) to browse for the



account. Click the verify button (🔍) to confirm that the username and password entered are valid. If desired, check the **Start service on startup** box to start the Keyfactor Command Service at system start.

★ **Tip:** Checking **Start service on startup** sets all jobs of type TimerService to *true* in the Keyfactor Command service appsettings.json file. These settings can be modified on a job-by-job basis in this appsettings.json file (see *Changing Default Timer Service Settings in Keyfactor Command Reference Guide* for more information).

The screenshot shows the 'Keyfactor Configuration Wizard' window with the 'Service' tab selected in the left-hand navigation pane. The main area is divided into two sections: 'Application Pools' and 'Authentication'. Under 'Application Pools', the 'User' field is set to 'KEYEXAMPLE\svc\_kyfservice' and the 'Password' field is masked with dots. A checkbox labeled 'Start service on startup' is checked. At the bottom of the wizard, there are buttons for 'Verify Configuration' and 'Cancel'. The status bar at the very bottom displays 'Server: sqlsrvr05.keyexample.com', 'Database Name: Command', and 'Credential Type: Windows'.

Figure 73: Configure: Service

## 16. Email Tab

On the Email tab, enter the FQDN of your SMTP server, the SMTP port (default is 25), and the sender name and account. Depending on the email configuration in your environment, the sender account may need to be a valid user on your mail server (using Active Directory credentials) or you may be able to put anything in this field (if your mail server supports anonymous connections). You may use the people picker button (👤) to browse for the sender account if you are using a valid account. Select the **Use SSL** box if this option is supported by your mail server and select the appropriate authentication method for your environment. If your mail server requires that you provide a username and password for a valid user, enter that Active Directory username and password in the fields at the bottom of the page after selecting the **Explicit credentials**

radio button. You may use the people picker button (👤) to browse for the account. Click the verify button (✅) to confirm that the username and password entered are valid. The user you select here must match the email address you set in the *Sender Account* field if you select *Explicit credentials*. The information entered on this tab may later be changed in the Keyfactor Command Management Portal.

Keyfactor Configuration Wizard		
File		
Validation Errors and Warnings		
Application Pools	Host	smtp.keyexample.com
Authentication	Port	25
Database	Sender Name	Keyexample Certificate Management
Service	Sender Account	Command@keyexample.com
Email		<input checked="" type="checkbox"/> Use SSL
Keyfactor Portal	Relay Authentication	<input checked="" type="radio"/> Anonymous
Administrative Users		<input type="radio"/> Explicit credentials
Dashboard and Reports	User	
Orchestrators	Password	
API		
Audit Configuration		
<div>Verify Configuration Cancel</div>		
Server: sqlsrvr05.keyexample.com   Database Name: Command   Credential Type: Windows		

Figure 74: Configure: Email

## 17. Keyfactor Portal Tab

On the Keyfactor Portal tab in the top section, enter the FQDN that you will use to access the Keyfactor Command Management Portal in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Administration component or a DNS alias pointing to the server. If you have multiple Keyfactor Command Management Portal servers with load balancing, this will be a DNS name pointing to your load balancer. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and select the application pool for this virtual directory that you created earlier in the **Application Pool** dropdown. Check or uncheck the **Use SSL** box as appropriate for your environment.

## Enrollment Section

In the Enrollment section of the page, modify the default **Certificate Subject Format** field, if desired. The subject values provided in this field are substituted at processing time for any entered by the user in PFX enrollment or provided with enrollment defaults if the template used is set to supply in request.

The data in the subject format takes precedence over any data entered during PFX enrollment or supplied by enrollment defaults (see *Configuring Template Options: Enrollment Defaults Tab* in the *Keyfactor Command Reference Guide*). For example, if you define the following subject format:

```
CN={CN},E={E},O=Key Example\, Inc.,OU={OU},L=Chicago,ST=IL,C=US
```

The organization for certificates generated through PFX enrollment will always be *Key Example, Inc.* regardless of what is shown on the PFX enrollment page during enrollment.

This setting also applies to CSRs generated using the CSR generation method.

Data from the default subject *does not* display in the PFX enrollment form. To define defaults that will display in the PFX enrollment form (and can be modified by users), use enrollment defaults (see *Configuring Template Options: Enrollment Defaults Tab* in the *Keyfactor Command Reference Guide*).



**Note:** Backslashes are required before any commas embedded within values in the subject field (e.g. O=Key Example\, Inc.). Quotation marks should not be used in the strings in the fields except in the case where these are part of the desired subject value, as they are processed as literal values.



**Tip:** The default subject format *does not* apply to enrollments done using the CSR enrollment method or any requests done with the Keyfactor API.

## PFX Enrollment Section

In the PFX Enrollment section of the page, uncheck the **Enabled** box if you do not wish to support PFX enrollment. If you wish to support PFX enrollment, leave the **Enabled** box checked. Select the **Domain** radio button if you wish PFX files to be protected with the user's Active Directory password or select the **Auto-Generated** radio button if you wish PFX files to be protected with a one-time password. Check the **Alphanumeric Password Characters** box if you wish the one-time password used to protect PFX files to contain numbers and letters. Uncheck the **Alphanumeric Password Characters** box if you wish the one-time password used to protect PFX files to contain numbers, letters and special characters. In the **Password Length** field, enter a number for the number of characters the one-time password should have. The minimum value is 8. If you select the **Domain** radio button, the data entered in the password fields is not relevant.

## CSR Enrollment Section

In the CSR Enrollment section of the page, uncheck the **Enabled** box if you do not wish to support CSR enrollment. If you wish to support CSR enrollment, leave the **Enabled** box checked.

Keyfactor Configuration Wizard

File

Validation Errors and Warnings

Application Pools

Authentication

Database

Service

Email

Keyfactor Portal

Administrative Users

Dashboard and Reports

Orchestrators

API

Audit Configuration

Host Name: keyfactor.keyexample.com ☒ Use SSL

Web Site: Default Web Site

Virtual Directory: KeyfactorPortal

Application Pool: KeyfactorPortal

Enrollment

Certificate Subject Format: CN=(CN),E={E},O={O},OU={OU},L={L},ST={ST},C

PFX Enrollment

☒ Enabled

PFX Password Type: ☐ Domain ☒ Auto-Generated

☒ Alphanumeric Password Characters

Password Length: 12

CSR Enrollment

☒ Enabled

Verify Configuration Cancel

Server: sqlsrvr05.keyexample.com Database Name: Command Credential Type: Windows

Figure 75: Configure: Keyfactor Portal

## 18. Administrative Users Tab

On the Administrative Users tab, click **Add** to add users or groups that you will use to control administrative access to the Keyfactor Command Management Portal.

Enter only the users and/or group(s) to which you want to grant full administrative rights to the Keyfactor Command Management Portal. Following initial configuration, you can create other permission levels and grant those permission levels to other users or groups through the Keyfactor Command Management Portal. See *Security Roles and Claims* in the *Keyfactor Command Reference Guide* for more information.

## Users and Groups for Active Directory as an Identity Provider

For each user to be added:

- In the **Identity Provider** dropdown, select Active Directory.
- In the **Claim Type** dropdown, select ADUser for a user account or ADGroup for a group created in Active Directory.
- In the **Claim Value** field enter the user or group name for the account in DOMAIN\name format (e.g. KEYEXAMPLE\Keyfactor Administrators).
- In the **Description** field enter a description to help you identify the user or group (e.g. the user's name).



**Important:** The built-in Active Directory groups Domain Admins and Enterprise Admins cannot be used directly to grant access to the Management Portal due to how these groups function within Windows. You can create a custom Active Directory group, reference that group in the Management Portal, and add the built-in Domain Admins or Enterprise Admins group to that custom group, if desired.



**Important:** For environments using Active Directory as an identity provider, the administrative group must be created as a Global or Universal group. If the administrative group is created as a domain-local group, it will not be recognized by the Management Portal Security Roles and Identities configuration. Configuration of the Management Portal will be incomplete until the group is deleted and recreated as a Global or Universal group.

The screenshot shows the 'Keyfactor Configuration Wizard' window. The left sidebar lists various configuration sections, with 'Administrative Users' selected. The main area is titled 'Validation Errors and Warnings' and contains an 'Add' button. Below the button are three dropdown menus: 'Identity Provider' (set to 'Active Directory'), 'Claim Type' (set to 'ADGroup'), and 'Claim Value' (set to 'KEYEXAMPLE\Keyfac'). There is an 'X' button next to the 'Claim Value' field. Below these is a 'Description' text box containing the text 'Keyfactor Command Global Administrators'. At the bottom of the window, there are 'Verify Configuration' and 'Cancel' buttons. The status bar at the very bottom shows 'Server: sqlsrvr05.keyexample.com', 'Database Name: Command', and 'Credential Type: Windows'.

Figure 76: Configure: Administrative Users for Active Directory

## Users for an Identity Provider Other Than Active Directory

For each user to be added:

- In the **Identity Provider** dropdown, select OAuth.
- In the **Claim Type** dropdown, select OAuthSubject for a user account created in your OAuth identity provider, OAuthRole for a role created in your OAuth identity provider, or OAuthClientId for a client application created in your OAuth identity provider. If your claim doesn't fall into one of these categories, select OAuthOid.
- In the **Claim Value** field enter the GUID for the user account, role name for the role, or client ID for the client (see [Configuring Keyfactor Identity Provider and Collecting Data for the Keyfactor Command Installation on page 25](#)). If you selected OAuthOid, enter an appropriate ID to identify the claim.
- In the **Description** field enter a description to help you identify the user (e.g. the user or group name).

**Keyfactor Configuration Wizard**

File

Validation Errors and Warnings

Application Pools

Authentication

Database

Service

Email

Keyfactor Portal

Administrative Users

Dashboard and Reports

Orchestrators

API

Audit Configuration

Add

Identity Provider	Claim Type	Claim Value
OAuth	OAuthSubject	5bec5f34-cf77-4fe6-
Description: John Smith		
OAuth	OAuthRole	command_administr
Description: Keyfactor Command Global Administrators		
OAuth	OAuthClientId	490-23
Description: Keyfactor API Automated Service		

Verify Configuration Cancel

Server: sqlsrvr05.keyexample.com Database Name: Command Credential Type: Windows

Figure 77: Configure: Administrative Users for OAuth

## 19. Dashboard and Reports Tab

On the Dashboard and Reports tab, enter the FQDN of the server hosting the Keyfactor Command Management Portal—where the Logi Analytics Platform is installed—in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Management Portal component or a DNS alias pointing to the server. Check or uncheck the **Use SSL** box as appropriate for your environment. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and select the application pool for this virtual directory that you created earlier in the **Application Pool** dropdown.



**Note:** If you are installing the Management Portal in a load balanced configuration, see [Appendix - Logi Load Balancing: Keyfactor Command Configuration Wizard Setup on page 181](#).

Figure 78: Configure: Dashboard and Reports

## 20. Orchestrators Tab

On the Orchestrators tab, enter the FQDN of the server hosting the Keyfactor Command orchestrators web site in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Services (Orchestrator Services API) components or a DNS alias pointing to the server. Select the Default Web Site in the **Web Site** drop-down, or other web site as desired. Accept the default for the **Virtual Directory** and select the application pool for this virtual directory that you created earlier in the **Application Pool** drop-down. Check or uncheck the **Use SSL** box as appropriate for your environment.

### Reenrollment Section (Optional)

In the **Template For Submitted CSRs** field, from the dropdown, select the template to be used for reenrollment requests made from the Certificate Stores page.

In the **CA For Submitted CSRs** field, enter the certificate authority used for reenrollment requests made from the Certificate Stores page. The CA should be entered in the format FQDN\Logical Name.



The screenshot shows the 'Keyfactor Configuration Wizard' window. The 'Orchestrators' tab is selected in the left-hand navigation pane. The main configuration area is divided into two sections: 'Reenrollment' and 'Certificate Authentication'. In the 'Reenrollment' section, the 'Template For Submitted CSRs' is set to 'Corp Web Server v2' and the 'CA For Submitted CSRs' is set to 'corpca01.keyexample.com\CorpIssuincA1'. In the 'Certificate Authentication' section, the 'Orchestrator Certificate Authentication' checkbox is unchecked. The 'Certificate Authentication HTTP Header' field is empty. The 'Certificate Authentication Username' field contains 'DOMAIN\user'. At the bottom of the window, there are buttons for 'Verify Configuration' and 'Cancel'. The status bar at the very bottom shows 'Server: sqlsrvr05.keyexample.com', 'Database Name: Command', and 'Credential Type: Windows'.

Category	Field	Value	Options/Status
Application Pools	Host Name	keyfactor.keyexample.com	<input checked="" type="checkbox"/> Use SSL
	Web Site	Default Web Site	Dropdown
Database	Virtual Directory	KeyfactorAgents	
Service	Application Pool	KeyfactorAgents	Dropdown
Keyfactor Portal	Template For Submitted CSRs	Corp Web Server v2	Dropdown
	CA For Submitted CSRs	corpca01.keyexample.com\CorpIssuincA1	Dropdown
Certificate Authentication	Orchestrator Certificate Authentication	<input type="checkbox"/> Enabled	Checkbox
	Certificate Authentication HTTP Header		Text Field
	Certificate Authentication Username	DOMAIN\user	Text Field with Password Icon

Figure 79: Configure: Orchestrators with Standard Authentication

## Certificate Authentication Section (Optional)

In the Certificate Authentication section of the Orchestrators tab, check the **Enabled** box if you wish to support client certificate enrollment from the Keyfactor Universal Orchestrator. In the **Certificate Authentication HTTP Header** field, enter the HTTP header under which the orchestrator connection proxy should send the client authentication certificate. Keyfactor Command uses the certificate supplied in this header to identify the orchestrator attempting to authenticate. In the **Certificate Authentication Username** and **Certificate Authentication Password** fields, enter the credentials for the Active Directory user configured on the proxy to authenticate the orchestrator(s) to the Keyfactor Command server.

Figure 80: Configure: Orchestrators with Client Certificate Authentication

## 21. API Tab

On the API tab, enter the FQDN of the server hosting the Keyfactor Command KeyfactorAPI service in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Services (Keyfactor API) components or a DNS alias pointing to the server. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and select the application pool for this virtual directory that you created earlier in the **Application Pool** dropdown. Check or uncheck the **Use SSL** box as appropriate for your environment.



**Tip:** Keyfactor Command includes embedded documentation which includes links to the Keyfactor API Reference and Utility from within the documents. If the API is installed in a non-standard virtual directory, these links will not work from the documentation. Keyfactor, recommends accepting the default virtual directory during installation of Keyfactor Command to avoid issues with this.

Keyfactor Configuration Wizard

File

Application Pools

Host Name:  ☒ Use SSL

Authentication

Web Site:

Database

Virtual Directory:

Service

Application Pool:

Email

Keyfactor Portal

Administrative Users

Dashboard and Reports

Orchestrators

API

Audit Configuration

Verify Configuration Cancel

Server: sqlsrvr05.keyexample.com Database Name: Command Credential Type: Windows

Figure 81: Configure: API

## 22. Audit Configuration Tab

On the Auditing Configuration tab, enter the number of years to retain audit data in the **Audit Entry Retention Period (years)** field. By default, seven years of data is retained. The audit log cleanup job runs once daily and removes any audit log entries older than the time specified in the retention parameter except those in the following protected categories:

- Security
- CertificateCollections
- ApplicationSettings
- SecurityIdentities
- SecurityRoles

## Linux SysLog Server Section

In the Linux SysLog Server section of the page, check the **Connect to SysLog** to enable the option to copy audit logs in real time to a separate server for collection and analysis with a centralized logging solution (e.g. rsyslog, Splunk, Elastic Stack). In the **Host Name** field, enter the fully qualified domain name of the server that will be receiving the logs. Set the **Port** to the port on which your log receipt application is listening to receive the logs. The default value is 514

(the default rsyslog port). If desired, turn on **Use TLS SysLogging**. When you click **Save**, Keyfactor Command will verify that a connection can be made to the specified server on the specified port. Additional configuration on both the Keyfactor Command server and log receipt server are needed to make TLS communications work (see [Prepare for External Log Shipping over TLS \(Optional\) on page 85](#)). If you have not yet completed these configurations, you will receive a validation error on save if the Use TLS SysLogging option is enabled.

The auditing settings can be updated on the auditing tab of the applications settings page following installation (see *Application Settings: Auditing Tab* in the *Keyfactor Command Reference Guide*).

The screenshot shows the 'Keyfactor Configuration Wizard' window with the 'Audit Configuration' tab selected. The left sidebar lists various configuration categories, with 'Audit Configuration' highlighted. The main panel displays the following settings:

- Audit Entry Retention Period (Years):** 7
- Linux SysLog Server:**
  - ☒ Connect to Linux SysLog
  - Host Name:** appsrvr162.keyexample.com
  - Port:** 10514
  - ☒ Use TLS SysLogging

At the bottom of the wizard, there are buttons for 'Verify Configuration' and 'Cancel'. A status bar at the very bottom shows: 'Server: sqlsrvr05.keyexample.com | Database Name: Command | Credential Type: Windows'.

Figure 82: Configure: Audit

23. At this point in the configuration, if you have populated all the required fields, the yellow warning banner at the top of the configuration wizard should have disappeared. If it is still visible, click the dropdown arrow to open the Warnings page and review the warning(s) to see what needs to be corrected. Under some circumstances you will be allowed to continue with the configuration even if the yellow warning banner is still present. You will know this is the case if the **Verify Configuration** button is active. Under these circumstances, you should review the warnings before continuing.

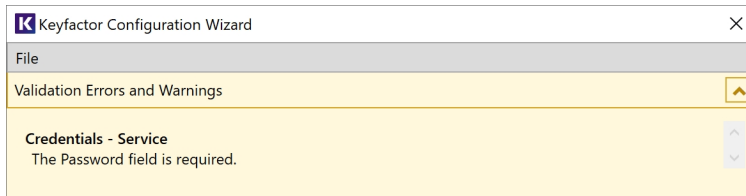


Figure 83: Configure: Configuration Warnings

24. Before completing the configuration wizard, you may choose to save a copy of the configuration as a file for future use. To download the configuration as a file, in the configuration wizard, click **File** at the top of the wizard and choose **Save Data File**. Browse to a location where you want to save the configuration file, enter a file name and click **Save**. You will be prompted to enter a password to encrypt the data in the file. You may choose to protect the file with a password or not. If you use a password at this time, you will need to provide this password to open the file. Keyfactor highly recommends using a strong password to protect the file. If you do not wish to use a password to protect the file, sensitive information (e.g. passwords for the service accounts entered in the configuration wizard) will be removed from the file. Once you enter a password or uncheck the encryption box, click **OK** to save the file.

**Important:** Keyfactor highly recommends that you use strong passwords for any accounts or certificates related to Keyfactor Command and associated products, especially when these have elevated or administrative access. A strong password has at least 12 characters (more is better) and multiple character classes (lowercase letters, uppercase letters, numeral, and symbols). Ideally, each password would be randomly generated. Avoid password re-use.

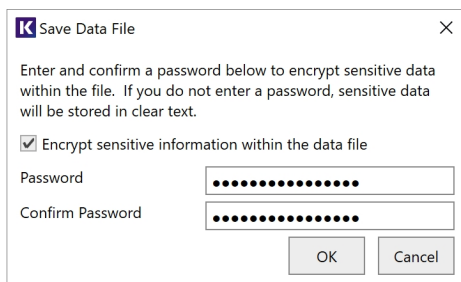


Figure 84: Configure: Save Configuration as a File

25. At the bottom of the Keyfactor Command Configuration Wizard dialog, click **Verify Configuration**.
26. On the Configuration Operations page, review the planned operations and then click **Apply Configuration**. Prior to clicking **Apply Configuration**, you can revisit any of the Configuration Wizard tabs to review or make changes by clicking **Edit Configuration**.

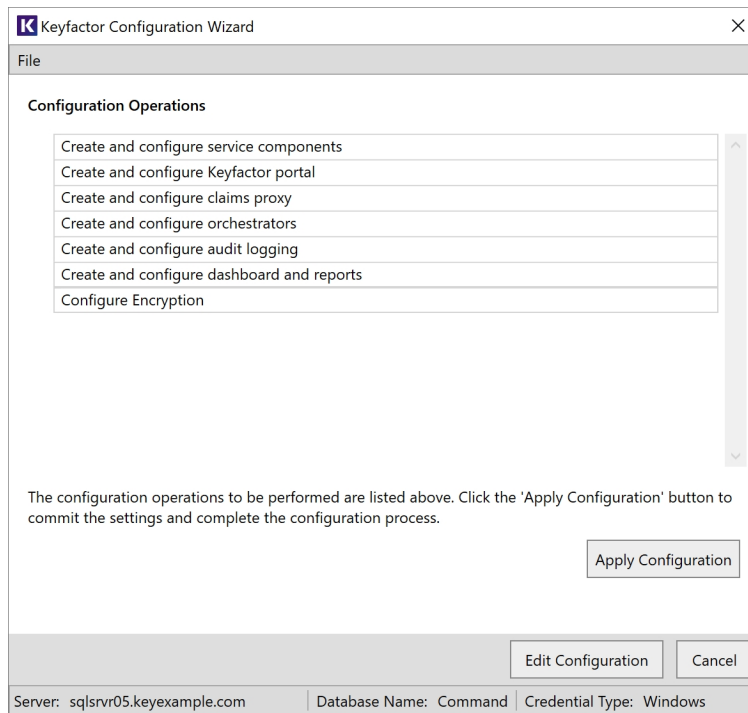


Figure 85: Configure: Configuration Operations

27. When the configuration completes successfully, you will see the below message. If you didn't save a copy of the configuration earlier, you may do so at this time by clicking **Save Settings**. Otherwise, click **Close** to close the dialog.

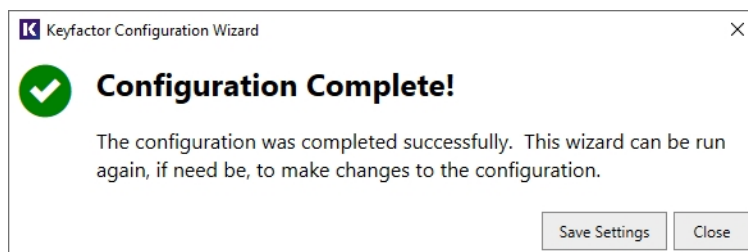


Figure 86: Configure: Configuration Complete

### 2.4.3.2 Install the Keyfactor Command Server from the Command Line

The Keyfactor Command server can optionally be configured using a pair of configuration files and a command run from the command line. You may be provided one or both of these files by your Keyfactor Customer Success Manager. The configuration files for command-line configuration are:

- Keyfactor Command Configuration File

This file, with an extension of `.cmscfg`, contains information in XML format to configure the Keyfactor Command database. This file can be generated by installing Keyfactor Command,

running the configuration wizard and populating all the fields as desired, and then saving a copy of the configuration either with or without a password to encrypt sensitive information in the file (see [Install the Keyfactor Command Components on the Keyfactor Command Server\(s\) on page 90](#)). Keyfactor highly recommends using a strong password to protect the file. A file that has not been protected with a password will be missing the sensitive information that would be protected by the password encryption (e.g. service account passwords).

- Input Parameters File

This file, with an extension of .xml, contains information in XML format to connect to and configure SQL, open the Keyfactor Command configuration file, locate the Keyfactor Command license, and create application pools, if desired.

To configure and, optionally, install Keyfactor Command from the command line:

1. Install the Keyfactor Command software using one of these methods:

- Follow the initial instructions for [Install the Keyfactor Command Components on the Keyfactor Command Server\(s\) on page 90](#) except on the final installation wizard page, uncheck the *Launch the Configuration Wizard now* box and click **Finish**. The configuration wizard should not open.
- Open an administrative command prompt and execute a command similar to the following:

```
start /wait msixec /i <full path to install file>\KeyfactorPlatform.msi /Live  
<path for msixec logs> /Quiet
```

This will install the default components of Keyfactor Command in a non-interactive way (/Quiet), output log information to a file (/Live), and wait to return to the command prompt until the installation is complete (start /wait).

If you wish to install a set of features other than the default features, you can add the ADDLOCAL parameter and specify the features you wish to install. For example, the following command will install the *Orchestrator Service API* and *Windows Services* features:

```
start /wait msixec /i <full path to install file>\KeyfactorPlatform.msi  
ADDLOCAL=AgentServicesFeature,ServiceFeature /Live <path for msixec logs>  
/Quiet
```

The following features are available:

- AgentServicesFeature  
This installs the Orchestrator Service API feature.
- ConfigurationFeature  
This installs the configuration wizard and is required for all installations.
- ServiceFeature  
This installs the Windows Services feature, which includes the Keyfactor Command Service (a.k.a. the timer service).
- VCRedistFeature

This installs the Microsoft Visual C++ Redistributable and is required for all installations unless it has been separately installed.

- WebApiFeature

This installs the WebAPI feature, which includes the Keyfactor API.

- WebConsoleFeature

This installs the Management Portal feature, which includes the Keyfactor Command Management Portal and the Keyfactor API.

The features you decide to install will depend on the role the server will be playing in your Keyfactor Command implementation. [Table 8: Features Required for Each Server Role](#) shows the minimum features that need to be installed for each of the server roles shown in the table columns. If you're installing all the required features on a single server, you need everything. If you don't intend to use any orchestrators (see *Installing Orchestrators* in the *Keyfactor Orchestrators Installation and Configuration Guide*), you do not need to install the *AgentServicesFeature*.

Table 8: Features Required for Each Server Role

ADDLOCAL Parameter	Single Server	Management Portal	Windows Services	Keyfactor API	Orchestrator Service API
Configuration Feature	✓	✓	✓	✓	✓
VC Redist Feature	✓	✓	✓	✓	✓
Web Console Feature	✓	✓			
Service Feature	✓		✓		
Web Api Feature				✓	
Agent Services Feature	✓				✓

2. Acquire a Keyfactor Command configuration file from your Keyfactor Customer Success Manager or create one by installing and configuring Keyfactor Command on a test machine. It's not practical to attempt to generate this file manually, though a file can be edited once generated (other than password-protected fields).



3. Create an input parameters file. See [Table 9: Input Parameters XML File Fields](#). A sample file can be found in the Configuration directory under the directory in which you installed Keyfactor Command. By default, this is:

C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\InputParameters.xml

4. Open an administrative command prompt, change to the Configuration directory under the directory in which you installed Keyfactor Command (by default this is C:\Program Files\Keyfactor\Keyfactor Platform\Configuration), and execute a command similar to the following, referencing your input parameters file and using the appropriate parameters for the ConfigurationWizardConsole tool (see [Table 10: ConfigurationWizardConsole.exe Options](#)):




.\ConfigurationWizardConsole.exe -p C:\Stuff\InputParameters.xml -u



**Tip:** Check the Keyfactor Command log and the Windows application event log for errors if the installation does not complete successfully (see [Configure Logging on page 139](#)).

Table 9: Input Parameters XML File Fields

Parameter	Description
Protected	A Boolean indicating whether sensitive information in the Keyfactor Command configuration file is protected with a password (true) or not (false).
Password	A string containing the password used to protect the Keyfactor Command configuration file if <i>Protected</i> is set to <b>true</b> .
Configuration File	The full path to the Keyfactor Command configuration file (e.g. C:\Stuff\myconfig.cmscfg).
Database Server	The hostname or IP address of the SQL server where the Keyfactor Command database will be installed, with optional port. For example: <ul style="list-style-type: none"> <li>Local with default port: mysql.keyexample.com</li> <li>Azure SQL myazuresql.database.windows.net,1433</li> </ul>
Database	The name of the database in SQL for Keyfactor Command. If a database with this name exists, it will be used (see <i>ForceDatabaseConversion</i> ). If it doesn't, it will be created (see <i>CreateDatabaseIfMissing</i> ).
Create Database If Missing	A Boolean indicating whether the SQL database should be created if it does not exist (true) or not (false). If this is set to <b>false</b> and a database does not exist, an error will be generated and the configuration will not continue.
Force Database Conversion	A Boolean indicating whether a pre-existing SQL database should be converted for use by Keyfactor Command (true) or not (false). If this is set to <b>false</b> and a pre-existing database that has not already been converted for Keyfactor Command use is found, an error will be generated and the configuration will not continue.
Force Database Upgrade	A Boolean indicating whether a pre-existing SQL database should be upgraded from a previous version of Keyfactor Command (true) or not (false). If this is set to <b>false</b> and a pre-existing database that is running a version of Keyfactor Command that does not match the version being installed is found, an error will be generated and the configuration will not continue.
Continue On Sql Grant Error	A Boolean indicating whether the configuration should continue if an error is encountered when attempting to set SQL permissions.
Sql Username	A string containing the SQL username to be used to authenticate to the SQL server if you have opted to use SQL authentication. For an on-premise SQL server, the server must be configured to support mixed mode authentication in order to use the SQL option. This option can be used to connect to cloud-based (e.g. Azure) SQL servers. Leave this field blank if you are using Windows integrated authentication. The credentials of the logged on user executing the command will be used to authenticate to

Parameter	Description										
	SQL.										
Sql Password	A string containing the SQL password to be used to authenticate to the SQL server. Leave this field blank if you are using Windows integrated authentication.										
License File	The full path to your Keyfactor Command license file (e.g. C:\Stuff\keyexample.cmslicense).										
Appliation Pools To Create	<p>An array of application pools to create. A separate application pool is required for each virtual directory that will be created for Keyfactor Command in IIS. If you choose to install all the roles, this will be either four or five application pools for the virtual directories with the following names, by default:</p> <ul style="list-style-type: none"> <li>• KeyfactorAgents (Keyfactor Command agent and orchestrator service endpoint)</li> <li>• KeyfactorAnalysis (Keyfactor Command dashboard and reporting)</li> <li>• KeyfactorAPI (Keyfactor API)</li> <li>• KeyfactorPortal (Keyfactor Command Management Portal)</li> <li>• KeyfactorProxy (Keyfactor Command proxy to your identity provider for OAuth support; only created if an identity provider other than Active Directory is used as the identity provider)</li> </ul> <p>Application pool fields include:</p> <table> <tr> <th>Parameter</th><th>Description</th></tr> <tr> <td>Name</td><td>A string containing the name of the application pool to create.</td></tr> <tr> <td>Username</td><td> <p>A string containing the user name of the Active Directory (DOMAIN\username format) or local (HOSTNAME\username format) service account under which the application pool will run.</p> <div>  <b>Tip:</b> The same service account may be used for all application pools. </div> </td></tr> <tr> <td>Password</td><td>A string containing the password of the Active Directory or local service account under which the application pool will run.</td></tr> <tr> <td>Fail If Exists</td><td>A Boolean indicating whether the configuration will fail if the application pool already exists (true) or not (false).</td></tr> </table> <p>For example:</p>	Parameter	Description	Name	A string containing the name of the application pool to create.	Username	<p>A string containing the user name of the Active Directory (DOMAIN\username format) or local (HOSTNAME\username format) service account under which the application pool will run.</p> <div>  <b>Tip:</b> The same service account may be used for all application pools. </div>	Password	A string containing the password of the Active Directory or local service account under which the application pool will run.	Fail If Exists	A Boolean indicating whether the configuration will fail if the application pool already exists (true) or not (false).
Parameter	Description										
Name	A string containing the name of the application pool to create.										
Username	<p>A string containing the user name of the Active Directory (DOMAIN\username format) or local (HOSTNAME\username format) service account under which the application pool will run.</p> <div>  <b>Tip:</b> The same service account may be used for all application pools. </div>										
Password	A string containing the password of the Active Directory or local service account under which the application pool will run.										
Fail If Exists	A Boolean indicating whether the configuration will fail if the application pool already exists (true) or not (false).										

Parameter	Description
	<pre> &lt;ApplicationPoolsToCreate&gt; &lt;!--Remove this section if none are to be created--&gt;   &lt;WizardApplicationPool&gt;     &lt;Name&gt;KeyfactorPortalPool&lt;/Name&gt;     &lt;Username&gt;KEYEXAMPLE\svc_kyfpools&lt;/Username&gt;     &lt;Password&gt;MySecurePassword&lt;/Password&gt;     &lt;FailIfExists&gt;true&lt;/FailIfExists&gt;   &lt;/WizardApplicationPool&gt;   &lt;WizardApplicationPool&gt;     &lt;Name&gt;KeyfactorAPIPool&lt;/Name&gt;     &lt;Username&gt;KEYEXAMPLE\svc_kyfpools&lt;/Username&gt;     &lt;Password&gt;MySecurePassword&lt;/Password&gt;     &lt;FailIfExists&gt;true&lt;/FailIfExists&gt;   &lt;/WizardApplicationPool&gt;   &lt;WizardApplicationPool&gt;     &lt;Name&gt;KeyfactorAnalysisPool&lt;/Name&gt;     &lt;Username&gt;KEYEXAMPLE\svc_kyfpools&lt;/Username&gt;     &lt;Password&gt;MySecurePassword&lt;/Password&gt;     &lt;FailIfExists&gt;true&lt;/FailIfExists&gt;   &lt;/WizardApplicationPool&gt;   &lt;WizardApplicationPool&gt;     &lt;Name&gt;KeyfactorAgentsPool&lt;/Name&gt;     &lt;Username&gt;KEYEXAMPLE\svc_kyfpools&lt;/Username&gt;     &lt;Password&gt;MySecurePassword&lt;/Password&gt;     &lt;FailIfExists&gt;true&lt;/FailIfExists&gt;   &lt;/WizardApplicationPool&gt; &lt;/ApplicationPoolsToCreate&gt; </pre>

Table 10: ConfigurationWizardConsole.exe Options

Switch	Description
-p, --paramfile	The full path to the input parameters XML file. This switch is <b>required</b> .
-u, --unattended	Do not output errors at the console. Errors will be redirected to the Windows event log.
-d, --database	Create the database in SQL but do not configure Keyfactor Command.
-s, --scriptpath	<p>The full path to a non-standard location for the scripts used during a database upgrade. By default, these are found in the following path:</p> <div>C:\Program Files\Keyfactor\Keyfactor Platform \Configuration\DatabaseUpgrade</div> <p>This option is typically only used by Keyfactor Support.</p>
--help	Display the help.
--version	Display the version information.

## 2.4.4 Initial Configuration

Once the installation and configuration wizards are complete, only a few configuration tasks remain before Keyfactor Command will be up and running at a basic level. This section details the basic post-install configuration steps that need to be completed to get Keyfactor Command up and running. See the *Keyfactor Command Reference Guide* for more advanced configuration guidance. See the separate installation guides for client components such as the Keyfactor Universal Orchestrator, Keyfactor Java Agent, and Keyfactor CA Gateways.

After you have completed all the steps in this guide, the certificate search and report functions in the Keyfactor Command Management Portal should be functioning. Further configuration, as described in the *Keyfactor Command Reference Guide*, is required to make these features function:

- Using the Keyfactor Command Management Portal *Dashboard*
- Configuring Enrollment through the Keyfactor Command Management Portal (see *Configuring Template Options*)
- *Security Roles and Claims* for the Keyfactor Command Management Portal
- *Revocation Monitoring, Expiration Alerts and Pending Certificate Request Alerts*
- Using the Workflow Builder (see *Workflow*)
- External Certificate Synchronization with *SSL Discovery* and *Certificate Stores*
- Managing *SSH Keys*

### 2.4.4.1 Configure Kerberos Authentication

In environments using Active Directory as an identity provider, the Keyfactor Command Management Portal uses integrated Windows authentication by default. Integrated authentication consists of both NTLM and Kerberos authentication types. In some environments, NTLM will work for integrated authentication and users will be able to open the Keyfactor Command Management Portal without further configuration, though not all aspects of the portal support NTLM, including the dashboard and enrollment. In other environments, NTLM will not work at all for the Management Portal, so only Kerberos will be supported. Further configuration is required to make Kerberos authentication work correctly. Even if NTLM is supported and you don't intend to use the portions of the Management Portal that don't work with NTLM, Kerberos is generally preferred for best security practice with Active Directory.

Common scenarios in which NTLM will not work are multi-domain forests and authentication attempts between domains and servers that support only NTLMv2 using clients attempting NTLM.

Configuring the environment to support Kerberos includes these topics:

- Configure browsers to support Integrated Windows Authentication (see [Configure Browsers for Integrated Windows Authentication below](#))
- Configure the service principal name (SPN) for the Keyfactor Command server (see [Configure the Service Principal Name for the Keyfactor Command Server on page 134](#))
- Configure Kerberos constrained delegation (see [Configure Kerberos Constrained Delegation \(Optional\) on page 134](#))



**Note:** Basic authentication can be used with Active Directory instead of integrated Windows authentication.

### Configure Browsers for Integrated Windows Authentication

To support integrated Windows authentication using either NTLM or Kerberos in environments using Active Directory as an identity provider, the browser must be configured correctly to support this integration. This becomes particularly important when only Kerberos is used, as the browser won't allow the user to continue if Kerberos authentication fails, whereas with NTLM authentication, the integration won't work (the user will be prompted to enter a password), but the user will be allowed to continue to the Keyfactor Command Management Portal. Many modern browsers support integrated authentication. The following instructions cover adding the Keyfactor Command server to Windows's trusted sites to support integrated authentication for Microsoft Edge and Google Chrome. Configuring Firefox to support integrated authentication is beyond the scope of this guide.



**Important:** Internet Explorer is no longer supported for Keyfactor Command. For a list of supported browsers, see [System Requirements on page 9](#).

To configure Windows to support integrated authentication:

1. In Windows either do a search for Internet Options or open Control Panel or Settings and locate Internet Options.
2. In Internet Options, go to the Security tab.
3. On the Security tab, highlight **Local intranet** and click **Sites**.
4. On the Local intranet sites popup, click **Advanced**.
5. On the Local intranet dialog, enter the fully qualified domain name of your Keyfactor Command server and click **Add**.
6. Click **Close** and **OK** until you have closed all the dialogs.
7. Exit your browser (this setting applies to Microsoft Edge and Google Chrome) and open it again to attempt your authentication.

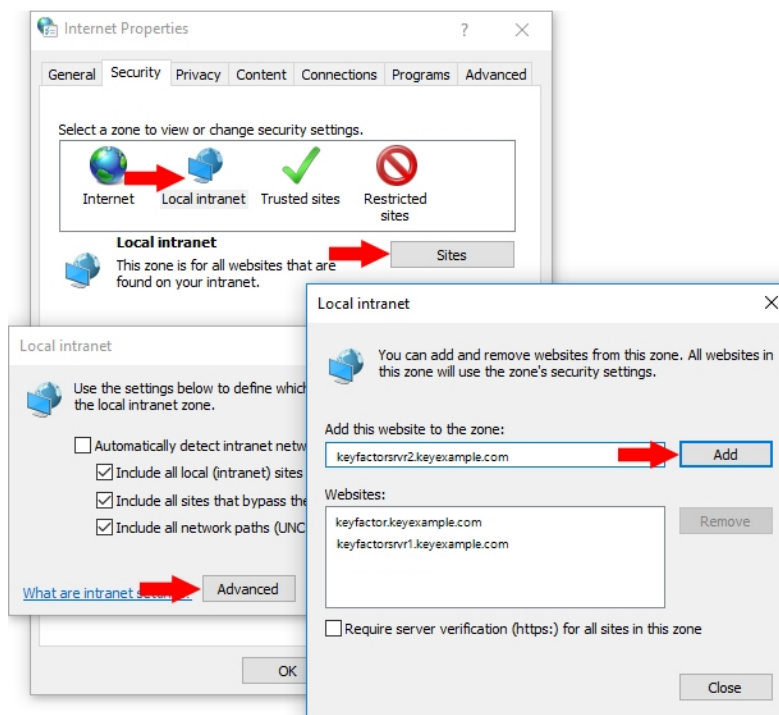


Figure 87: Configure Local Intranet Zone in Internet Properties



**Important:** It is not sufficient to put the Keyfactor Command server in the **Trusted sites** zone. The server needs to be in the **Local intranet** zone for proper integrated authentication functionality (assuming these zones are still configured as per the default configuration).

## Configure the Service Principal Name for the Keyfactor Command Server

In environments using Active Directory as an identity provider, configure the service principal name (SPN) for the Keyfactor Command server as follows:

1. On a server that has the `setspn` command available (typically it is available on domain controllers, as it installs as part of the Active Directory Domain Services role), open a command prompt using the “Run as administrator” option.
2. Run the following command (where `keyfactor.keyexample.com` is the fully qualified domain name of your Keyfactor Command server or the DNS alias you are using to reference your Keyfactor Command server, if applicable, and `KEYEXAMPLE\svc_keyfactorpool` is the domain name and service account name of the service account under which the Keyfactor Command application pool for the Management Portal is running):

```
setspn -s HTTP/keyfactor.keyexample.com KEYEXAMPLE\svc_keyfactorpool
```

## Configure Kerberos Constrained Delegation (Optional)

If you are using Active Directory as an identity provider and either of these scenarios is true in your environment, you will need to configure Kerberos delegation to the CAs from the Keyfactor Command server hosting the Keyfactor Command Management Portal:

- You wish to use the option in Keyfactor Command to allow interactions with the CA via the Keyfactor Command Management Portal (e.g. certificate approval or revocation) to be done in the context of the user logged into the Keyfactor Command Management Portal rather than in the context of the Keyfactor Command service account under which the application pool is running or an explicit user configured in the CA configuration within Keyfactor Command.
- You wish to enroll for certificates through the Keyfactor Command Management Portal after authenticating to the portal using Kerberos authentication rather than Basic authentication. If you wish to use the Keyfactor Command Management Portal but don't wish to configure delegation or an explicit user configured in the CA configuration within Keyfactor Command, you will need to set the Keyfactor Command Management Portal to support Basic authentication only.

Configuring Kerberos delegation in Active Directory allows the user's Kerberos credentials to be delegated from the Keyfactor Command server to the CA(s) to allow the Keyfactor Command server to act on behalf of the user.

The types of interactions affected by delegation in the Keyfactor Command Management Portal include:

- Enrollment for certificates
- Approval of pending certificate requests
- Denial of pending certificate requests
- Revocation of certificates
- Certificate key recovery





**Note:** You have the option to turn off delegation for these functions using the *Delegate* settings on each CA configured in Keyfactor Command (see *Certificate Authority Operations: Adding or Modifying a CA Record Authorization Methods Tab* in the *Keyfactor Command Reference Guide*). Delegation is configured separately for management and enrollment functions.

There are two different approaches to configuring constrained delegation:

- With the traditional version of constrained delegation, you configure the service account under which the Keyfactor Command Management Portal application pool runs and the machine account of the Keyfactor Command server to be allowed to delegate **to** each of your CAs.
- With the newer resource-based constrained delegation introduced in Windows server 2012, you configure each of your CAs to be allowed to receive delegation **from** the service account under which the Keyfactor Command Management Portal application pool runs and the machine account of the Keyfactor Command server. This option requires at least one domain controller that's server 2012 or better, though there can be 2008 or 2008 R2 domain controllers in the mix.

With both approaches to constrained delegation, you need to set the service principal name (SPN) for the Keyfactor Command server (see [Configure the Service Principal Name for the Keyfactor Command Server on the previous page](#)).



**Note:** If you're using a Keyfactor CA gateway and the gateway service is running as an Active Directory service account, delegation to that gateway is configured differently than is described below. Refer to the gateway documentation for more information.

## Traditional Delegation



**Note:** Traditional constrained Kerberos delegation across multiple domains is only supported in newer versions of Windows Server for domain controllers. If yours is a multi-domain environment and you cannot locate your CAs following the below instructions, you may need to configure traditional unconstrained Kerberos delegation or configure traditional constrained delegation using ADSIEdit rather than the below method. To configure traditional unconstrained delegation, you would select "Trust this computer for delegation to any service (Kerberos only)" in each of the step 2s, below, and then skip the remainder of the steps in that set of instructions. For assistance configuring traditional constrained delegation using ADSIEdit, contact Keyfactor support ([support@keyfactor.com](mailto:support@keyfactor.com)). If none of the traditional constrained delegation methods work in your multi-domain environment, you may need to pursue resource-based constrained delegation instead, which is more forgiving of multi-domain environments.

To configure Kerberos constrained delegation on the machine account of the Keyfactor Command server:

1. Open Active Directory Users and Computers and browse to locate the **machine** account of the Keyfactor Command server and open its properties.

2. On the Delegation tab for the machine account, choose “Trust the computer for delegation to specified services only” and under that “Use any authentication protocol” and then click **Add**.
3. In the Add Services dialog, click **Users or Computers** and browse to locate the computer account for one of the CAs to which you wish to delegate.

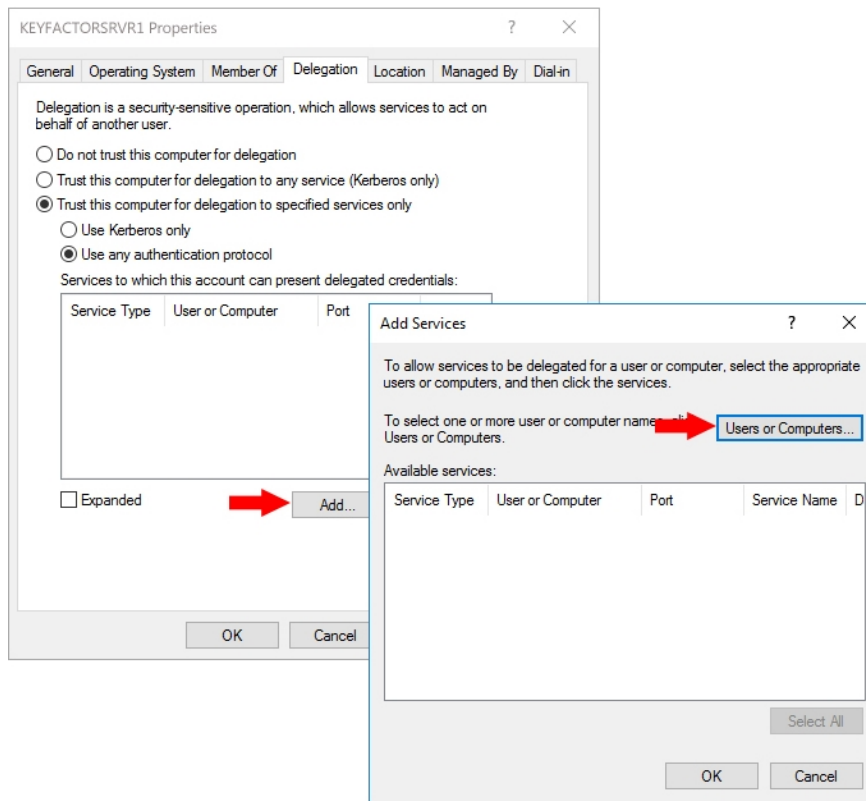


Figure 88: Configure Kerberos Constrained Delegation on the Keyfactor Command Machine Account

4. In the Add Services dialog once the available services have populated, highlight both the **HOST** and the **rpcss** services (hold down the CTRL key when clicking the second service to select both at the same time) and click **OK**.

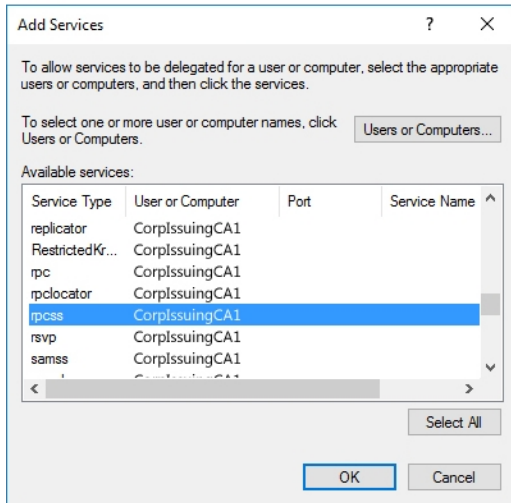




Figure 89: Add HOST and rpcss Service Types for Kerberos Constrained Delegation

5. Back on the Delegation tab of the machine account Properties, you should see the HOST and rpcss services populate the “Services to which this account can present delegated credentials box”. The server names do not appear as fully qualified domain names until you close the properties dialog and open it again.
6. Repeat steps 3 and 4 for any other CAs to which you wish to delegate and then click **OK**.

To configure Kerberos constrained delegation on the **service** account under which the Keyfactor Command application pool is running:

1. Open Active Directory Users and Computers and browse to locate the service account under which the Keyfactor Command application pool is running and open its properties.
2. On the Delegation tab for the service account, choose “Trust the computer for delegation to specified services only” and under that “Use Kerberos only” and then click **Add**.

 **Important:** This is a different configuration setting than for the machine account.

 **Tip:** The Delegation tab only appears on the properties sheet after you have configured a custom SPN.

3. In the Add Services dialog, click **Users or Computers** and browse to locate the computer account for one of the CAs to which you wish to delegate.
4. In the Add Services dialog once the available services have populated, highlight both the **HOST** and the **rpcss** services (hold down the CTRL key when clicking the second service to select both at the same time) and click **OK**.

- Back on the Delegation tab of the service account Properties, you should see the HOST and rpcss services populate the “Services to which this account can present delegated credentials” box”. The server names do not appear as fully qualified domain names until you close the properties dialog and open it again.

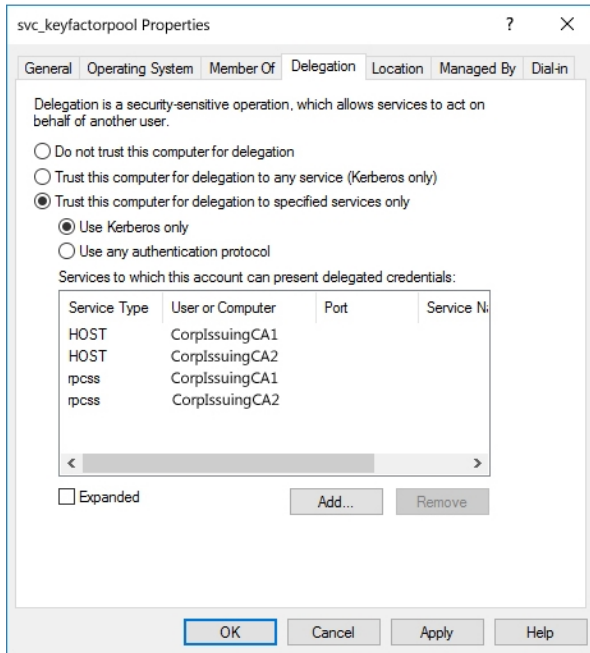


Figure 90: Configure Kerberos Constrained Delegation on the Keyfactor Command Service Account

- Repeat steps 3 and 4 for any other CAs to which you wish to delegate and then click **OK**.

## Resource-Based Delegation

To configure Kerberos resource-based constrained delegation:

- Create at least one Active Directory security group that will be used for granting delegation permission.



**Tip:** The type of group to use and domain to place it in will vary depending on your forest structure and the location of the accounts and resources in the forest. If your Keyfactor Command server(s), CA(s) and application pool service account are all in the same domain, any type of group in that same domain will do. If your Keyfactor Command server(s), CA(s) and/or application pool service account are separated across multiple domains, it becomes more complicated. If you add one or more cross-forest trusts into the mix that you want to do delegation across, that adds another level of complexity. Universal groups cannot be used in a cross-forest scenario, as they are not supported cross-forest. Some possible scenarios include:



- All components (Keyfactor Command server(s), CA(s) and application pool service account) in the same domain: Create a group of any type in the same domain as these components.
- Keyfactor Command server(s) and application pool service account in child domain A and all CAs in the parent domain or child domain B with no cross-forest involvement: Create a universal group in the domain with the CAs (the parent domain or child domain B).
- Keyfactor Command server(s), application pool service account and CA(s) for forest A in the same domain, CAs in a single domain in forest B, forests A and B in a two way trust: Create a group of any type in the forest A domain where the Keyfactor Command server(s) reside and a group of type *domain local* in the forest B domain where the CAs reside; follow the below instructions for both domains and groups.

2. Add the service account under which the Keyfactor Command Management Portal application pool runs to this new security group.
3. Add the machine account for the Keyfactor Command server to this new security group. Repeat for additional Keyfactor Command servers.
4. On an Active Directory domain controller running Windows Server 2012 or better, open a PowerShell window using the “Run as administrator” option. If you’re in a multi-domain or cross-forest environment, use a domain controller in the resource domain where the CAs exist.
5. In the PowerShell window, run the following commands, where *KerberosDelegationGroup* is the name of your group for Kerberos delegation and *IssuingCA* is the machine name (no trailing \$) of the CA you wish to delegate to:

```
$mygroup = Get-ADGroup -Identity KerberosDelegationGroup
Set-ADComputer IssuingCA -PrincipalsAllowedToDelegateToAccount $mygroup
```

6. Repeat the Set-ADComputer step for any additional CAs.
7. In the PowerShell window, run the following command for each CA to confirm that the group has been associated with the PrincipalsAllowedToDelegateToAccount property on the CA account:

```
Get-ADComputer IssuingCA -Properties PrincipalsAllowedToDelegateToAccount
```

#### 2.4.4.2 Configure Logging

Keyfactor Command provides extensive logging for visibility and troubleshooting. By default, Keyfactor Command places its log files in the C:\Keyfactor\logs directory, generates logs at the *Info* logging level and stores the primary logs for two days before deleting them. If you wish to change these defaults you can open the configuration file for each type of log on each Keyfactor Command server where you wish to adjust logging, and edit the file in a text editor (e.g. Notepad) using the “Run as administrator” option. Each component has its own NLog configuration file and NLog logging output path.

For more information, see *Editing NLog* in the *Keyfactor Command Reference Guide*.

### 2.4.4.3 Configure CA Certificate Synchronization

The Keyfactor Command certificate management, notification and reporting features make use of a SQL database containing certificates from many locations, including:

- Certificates synchronized from Microsoft or EJBCA CAs managed by Keyfactor
- Certificates synchronized from domain-joined Microsoft CAs in your primary forest and forests with which the forest shares a trust
- Certificates synchronized from non-domain-joined EJBCA and Microsoft CAs
- Certificates synchronized from your domain-joined Microsoft CAs in non-trusted forests
- Certificates automatically imported based on SSL synchronization locations
- Certificates imported via Keyfactor CA Gateways from locations such as Entrust and Symantec clouds
- Manually imported certificates
- Certificates inventoried from certificate stores using Keyfactor Command Orchestrators

In order to get these certificates into the Keyfactor Command database so that you can begin using the management, notification and reporting features, you need to configure—at a minimum—CA synchronization. For more information:

- See *Certificate Authorities* in the *Keyfactor Command Reference Guide* for information on configuring CA synchronization for your Microsoft and EJBCA CAs.
- See *SSL Discovery* in the *Keyfactor Command Reference Guide* for information on configuring SSL discovery and monitoring.
- See the separate documentation for each type of CA gateway you have along with *Certificate Authorities* in the *Keyfactor Command Reference Guide* for information on configuring CA synchronization for your CA gateways.
- See *Add Certificate* in the *Keyfactor Command Reference Guide* for information on manually importing a certificate.
- See *Installing Orchestrators* in the *Keyfactor Orchestrators Installation and Configuration Guide* and *Orchestrators* and *Certificate Stores* in the *Keyfactor Command Reference Guide* for information on inventorying certificates from certificate stores.

For information on using the Keyfactor Command Management Portal, see *Using the Management Portal* in the *Keyfactor Command Reference Guide*.

### Acquire a Client Certificate for EJBCA CA Authentication

Keyfactor Command uses a client certificate to authenticate to the EJBCA certificate authority to support certificate synchronization, enrollment, and revocation. The certificate that Keyfactor Command uses for authentication needs:

- An extended key usage (EKU) of Client Authentication
- A key usage that includes Digital Signature

**X.509v3 extensions**

**Usages**

**Key Usage** [?]  
☒ Use... ☒ Critical  
 Key Usage:  
☒ Digital Signature ☐ Data encipherment ☐ CRL sign  
☒ Non-repudiation ☐ Key agreement ☐ Encipher only  
☒ Key encipherment ☐ Key certificate sign ☐ Decipher only

**Extended Key Usage** [?]  
☒ Use... ☐ Critical  
 Any Extended Key Usage  
 CSN 369791 TLS client  
 CSN 369791 TLS server  
**Client Authentication**  
 Code Signing  
 EAP over LAN (EAPOL)  
 EAP over PPP  
 ETSI TSL Signing  
 Email Protection  
 ICAO Deviation List Signing

**Certificate Policies** [?]  
☐ Use... ☐ Critical

Figure 91: Certificate Profile for EJBCA Client Certificate

The certificate needs to be available as a PKCS#12 (\*.pfx) file in order to import it into Keyfactor Command.

**Confirm request**

Issuer Distinguished Name: CN=ManagementCA,O=Key Example,C=US  
 Subject Distinguished Name: O=Key Example,L=Chicago,ST=Illinois,C=US  
 Public Key Specification: RSA\_2048  
 Validity: 2y

Show details

Download JKS Download PKCS#12 Download BCFKS Download PEM

Figure 92: Certificate Download for EJBCA Client Certificate



**Important:** Keyfactor highly recommends that you use strong passwords for any accounts or certificates related to Keyfactor Command and associated products, especially when these have elevated or administrative access. A strong password has at least 12 characters (more is better) and multiple character classes (lowercase letters, uppercase letters, numeral, and symbols). Ideally, each password would be randomly generated. Avoid password re-use.

The certificate needs to be granted appropriate access to the EJBCA CA to allow Keyfactor Command interactions with the CA to take place (see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs below](#)).

## Grant the Keyfactor Command Users and Service Account(s) Permissions on the CAs

In order for Keyfactor Command to be able to synchronize certificates from the CAs to the Keyfactor Command database, the service account under which Keyfactor Command makes a connection to the CA must have permissions to *read* the CA databases. For full Keyfactor Command functionality, additional permissions are needed. The permissions needed vary depending on the type of CA and the type of authorization you intend to configure to allow Keyfactor Command and users in Keyfactor Command to interact with the CA.

### Microsoft CAs

When you configure Keyfactor Command access to a Microsoft CA, you have the option to enable the *Use Explicit Credentials* option. When this option is enabled, you enter a set of credentials that will be used specifically to access that Microsoft CA, and all management and enrollment tasks for that CA are done in the context of that service account. If you do not enable the *Use Explicit Credentials* option, management tasks (e.g. revocation, certificate synchronization) and enrollments are done in the context of the service account(s) you configure for the Keyfactor Command Service and Keyfactor API the application pool for Keyfactor Command (which are the same service account in many implementations) and individual users. The exact combination of what happens in the context of who depends on the configuration of the delegation options (*Delegate Management Operations* and *Delegate Enrollment*) on the CA when the *Use Explicit Credentials* option is not enabled. Delegation is supported for Basic and Kerberos authentication (see [Configure Kerberos Constrained Delegation \(Optional\) on page 134](#)) but not NTLM or Token authentication. Use of explicit credentials is mutually exclusive of delegation.

The users and service account(s) you will be using to connect to your Microsoft CA(s) from Keyfactor Command need some set of the following permissions on the CA, based on the configuration of authorization for the CA:

- Read  
To support CA synchronization
- Issue and Manage Certificates  
To support certificate revocation and key recovery
- Manage CA  
To support CRL publication following revocation
- Request Certificates  
To support certificate enrollment through Keyfactor Command



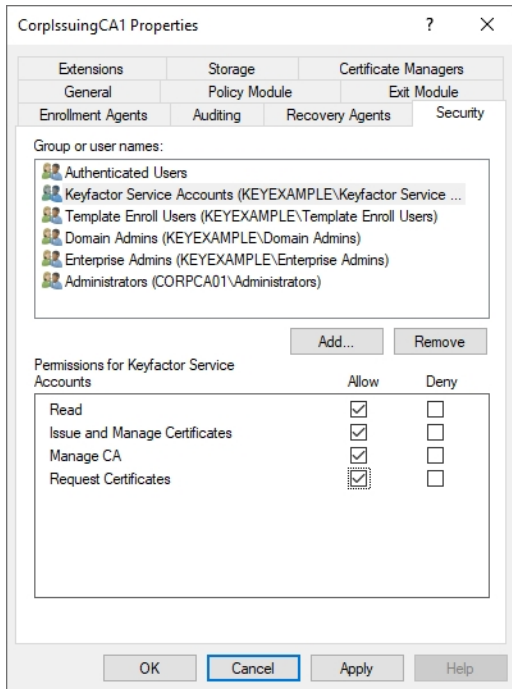


Figure 93: Microsoft CA Permissions














[Table 11: Microsoft CA Permission Matrix](#) provides information on what permissions are required on the Microsoft CA based on possible authorization configurations.

In the management console for each CA that Keyfactor Command will be interacting with, open the properties for the CA and grant the users and service account(s) for Keyfactor Command the appropriate permissions for your environment before continuing.



**Tip:** In order to support PFX and CSR enrollment through the Management Portal, the user initiating the enrollment in the Management Portal must be granted “Request Certificates” permission in the CA if enrollment delegation is enabled. In many environments, all Authenticated Users are granted this permission, allowing the Management Portal users to inherit the permission.

Table 11: Microsoft CA Permission Matrix

	 <b>Use Explicit Credentials</b>	 <b>Use Explicit Credentials</b>	 <b>Use Explicit Credentials</b>	 <b>Use Explicit Credentials</b>	 <b>Use Explicit Credentials</b>
		 <b>Delegate Management</b>	 <b>Delegate Management</b>	 <b>Delegate Management</b>	 <b>Delegate Management</b>
		 <b>Delegate Enrollment</b>	 <b>Delegate Enrollment</b>	 <b>Delegate Enrollment</b>	 <b>Delegate Enrollment</b>
Explicit CA-Specific User	Read Issue & Manage Certs Manage CA Request Certs	n/a	n/a	n/a	n/a
Keyfactor Command Service Account	None	Read Request Certs <sup>1</sup>	Read Request Certs <sup>2</sup>	Read Request Certs <sup>3</sup>	Read Request Certs <sup>4</sup>
Keyfactor API Application Pool Account <sup>5</sup>	None	Read Issue & Manage Certs Manage CA Request Certs <sup>6</sup>	Read Issue & Manage Certs Manage CA Request Certs <sup>7</sup>	Read Manage CA Request Certs	Read Issue & Manage Certs Manage CA Request Certs

<sup>1</sup>To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

<sup>2</sup>To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

<sup>3</sup>To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

<sup>4</sup>To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

<sup>5</sup>A separate application pool is required for each virtual directory that will be created for Keyfactor Command in IIS. Requests are made in the context of the user running the application pool for the Keyfactor API virtual directory.

<sup>6</sup>To support tests of certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

<sup>7</sup>To support tests of certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

	Use Explicit Credentials	Use Explicit Credentials	Use Explicit Credentials	Use Explicit Credentials	Use Explicit Credentials
		Delegate Management	Delegate Management	Delegate Management	Delegate Management
		Delegate Enrollment	Delegate Enrollment	Delegate Enrollment	Delegate Enrollment
Individual Users	None	Read Issue & Manage Certs Request Certs	Read Request Certs	Read Issue & Manage Certs	None

## EJBCA CAs

Management (e.g. revocation, certificate synchronization) and enrollment requests to an EJBCA CA are made in the context of the end entity associated with the client certificate selected in each CA configuration in the Keyfactor Command Management Portal to provide authentication to the EJBCA CA (see [Acquire a Client Certificate for EJBCA CA Authentication on page 140](#)). The access rule created or used for this needs to grant sufficient permissions to allow the end entity to synchronize certificates. For full functionality, it needs the following permissions:

- `/administrator/`  
To support Keyfactor Command making API requests to the EJBCA CA
- `/ca/[your_ca_name]/`  
To support Keyfactor Command access to your CA
- `/ca_functionality/create_certificate/`  
To support certificate enrollment through Keyfactor Command
- `/ca_functionality/create_crl/`  
To support CRL publication following revocation
- `/ca_functionality/view_ca/`  
To support retrieval of CA information
- `/ca_functionality/view_certificate/`  
To support CA synchronization
- `/ca_functionality/view_certificate_profiles/`  
To support template import
- `/endentityprofilesrules/[your_end_entity_profile_name]/create_end_entity/`

To support creation of end entities (a new end entity is created for each Keyfactor Command certificate enrollment unless the *Enforce Unique DN* option is enabled and the new certificate's DN matches that of an existing certificate)

- /endentityprofilesrules/[your\_end\_entity\_profile\_name]/edit\_end\_entity/

To support certificate enrollment with the *Enforce Unique DN* option through Keyfactor Command and certificate renewal through Keyfactor Command

- /endentityprofilesrules/[your\_end\_entity\_profile\_name]/revoke\_end\_entity/

To support certificate revocation through Keyfactor Command

- /endentityprofilesrules/[your\_end\_entity\_profile\_name]/view\_end\_entity/

To support certificate enrollment through Keyfactor Command

- /ra\_functionality/create\_end\_entity

To support creation of end entities (a new end entity is created for each Keyfactor Command certificate enrollment unless the *Enforce Unique DN* option is enabled and the new certificate's DN matches that of an existing certificate)

- /ra\_functionality/edit\_end\_entity

To support certificate enrollment with the *Enforce Unique DN* option through Keyfactor Command and certificate renewal through Keyfactor Command

- /ra\_functionality/revoke\_end\_entity

To support certificate revocation through Keyfactor Command

- /ra\_functionality/view\_end\_entity

To support certificate enrollment through Keyfactor Command

- /system\_functionality/view\_administrator\_privileges

To support overall functionality

# Edit Access Rules[?]

## Role : Keyfactor Role

Where "ManagementCA" is the name of your CA.

Resource	Rule
/administrator/	Allow
/ca/ManagementCA/	Allow
/ca_functionality/create_certificate/	Allow
/ca_functionality/create_crl/	Allow
/ca_functionality/view_ca/	Allow
/ca_functionality/view_certificate/	Allow
/ca_functionality/view_certificate_profiles/	Allow
/endentityprofilesrules/Sample/create_end_entity/	Allow
/endentityprofilesrules/Sample/edit_end_entity/	Allow
/endentityprofilesrules/Sample/revoke_end_entity/	Allow
/endentityprofilesrules/Sample/view_end_entity/	Allow
/ra_functionality/create_end_entity/	Allow
/ra_functionality/edit_end_entity/	Allow
/ra_functionality/revoke_end_entity/	Allow
/ra_functionality/view_end_entity/	Allow
/system_functionality/view_administrator_privileges/	Allow

Where "Sample" is the name of your end entity profile or profiles.

Figure 94: EJBCA Access Permissions

You may either create a new access rule that limits access to just these required permissions, or use an existing access rule. In either case, you need to add the certificate used to authenticate Keyfactor Command to the EJBCA CA as a member of that access rule.

## Members

### Role : Keyfactor Role

[Back to Roles Management](#)  
[Edit Access Rules](#)

Match with	CA	Match	Operator	Match Value	Description	Action
X509: Certificate serial number (Recommended)	ManagementCA	-	Equal, case insens.	569B60F0BF65DF9EB473A4C8D3FF6F844D478C9F		Add
X509: Certificate serial number (Recommended)	ManagementCA	-	Equal, case insens.	5948057A4A5E6DAFE9157CF81C328A1FB67F1A54		Delete

Figure 95: Add Client Certificate as Member of EJBCA Access Rule

## Enable and Start the Keyfactor Command Service

The Keyfactor Command Service runs on the Keyfactor Command server hosting the Services role and controls database synchronization, among other jobs. During the Keyfactor Command configuration process you configured the service account under which the Keyfactor Command Service will run and may have configured the service to start automatically at server boot time (see [Configure: Service on page 111](#)).



**Tip:** The Keyfactor Command Service can be installed on every server that Keyfactor Command is installed on—for instance in a high availability scenario. This allows the service to check out jobs via a locking mechanism that enforces that any jobs are running on only one



service at a time. There is a timeout setting for the service locking mechanism that may be adjusted if needed. <sup>1</sup>

To start the service (if it hasn't started automatically):

1. On the Keyfactor Command server hosting the Services role, open the Services MMC.
2. In the Services MMC confirm that the Keyfactor Command Service is set to a Startup Type of Automatic (if desired). If the service is not running, click the green arrow to start it.

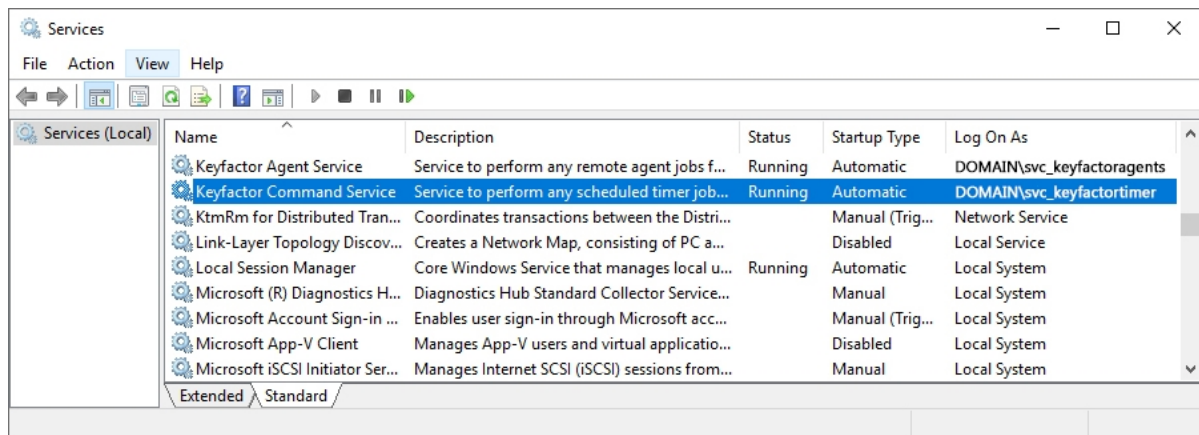


Figure 96: Keyfactor Command Service

If you have configured a synchronization schedule for your CA (see [Configure CA Certificate Synchronization on page 140](#)), the CA(s) will begin to synchronize when the first scheduled scan time is reached. Scans scheduled at intervals match to clock times, so a scan set at an interval of 15 minutes will run at 6:00, 6:15, 6:30, 6:45, etc. You can check the Keyfactor Command timer service log file on the Keyfactor Command Services server to confirm that synchronization is operating as expected. You can also use the Certificate Search feature in the Keyfactor Command Management Portal to confirm the certificates are appearing in the Keyfactor Command database. The database synchronization begins with the oldest certificates in the CA database, which may be expired or revoked. Be sure to toggle the *Include Revoked* and *Include Expired* options, see [Include Expired and Revoked Certificates in Certificate Search on the next page](#), when checking to see if synchronization is working. See *Certificate Search Page* in the *Keyfactor Command Reference Guide* for information on using the search.

<sup>1</sup>Adjust the timeout setting for the service locking mechanism:

1. Navigate to the Configuration folder (default location: *C:\Program Files\Keyfactor\Keyfactor Platform\Configuration*)
2. Open the file: **ConfigurationWizardConsole.exe.config**
3. Edit the value on the line: `<add key="Keyfactor.Sql.DbCommandTimeout" value="1800" />`

## Certificate Search<sup>9</sup>

Clicking "Advanced" allows you to build a query based on multiple criteria using AND/OR logic. Select a field and comparison operator in the drop-downs and then enter a comparison value. Click "Insert" to add the search criteria to the query field below the selection fields. Each time you click the "Insert" button, an AND is added between the previous search criteria and the newly added one. You can change the AND to an OR if desired. You can use parentheses around portions of the query along with AND/OR to change the query meaning.

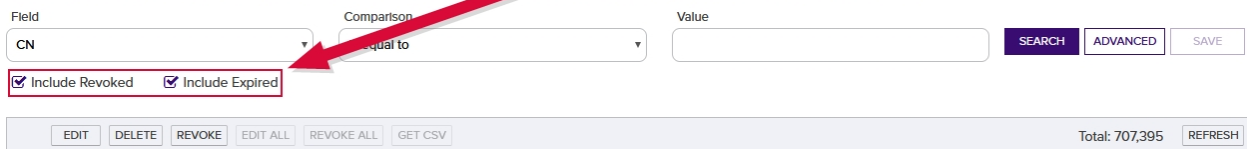


Figure 97: Include Expired and Revoked Certificates in Certificate Search

### 2.4.4.4 Create or Identify Certificate Templates for Enrollment

This step only needs to be completed if your Keyfactor Command license includes certificate enrollment and you plan to use this feature.



**Note:** Keyfactor Command and this documentation use the term *template* generically to refer to Microsoft certificate templates and EJBCA certificate templates. EJBCA templates are built from the EJBCA end entity profile and certificate profile and named using a naming scheme of <end entity profile name>\_<certificate profile name> and <end entity profile name> (<certificate profile name>) for the template name and template display name.

The enrollment function in the Keyfactor Command Management Portal is generally used by administrators to request certificates for use on servers, network devices, and similar equipment. There's a good chance that certificate templates for these purposes already exist in your environment. To prepare for the Keyfactor Command installation, you need to gather a list of the CAs that will be used to issue certificates through the Keyfactor Command Management Portal and a list of the *template names* (vs template display names) of the templates that will be used for this (Microsoft CAs) or certificate profiles and end entity profiles (EJBCA CAs). If any new templates or profiles need to be created for this purpose, they should be created before completing the Keyfactor Command post-installation steps.

For Microsoft CAs, the security settings on your existing templates may need to be modified to allow users to enroll for certificates using them through the Keyfactor Command Management Portal, depending on how the templates have been used previously. For CAs in the local forest (the forest in which Keyfactor Command is installed) and forests in a two-way trust with the local forest, enrollment through the Keyfactor Command Management Portal is often done in the context of the user logged into the portal. This differs from enrolling for a certificate through the Microsoft certificates MMC, where requests for computer certificates (such as web server certificates) are done in the context of the machine account from which the certificate is requested, not the user account, and thus the machine account needs permissions, not the user. When using the Keyfactor Command Management Portal, each of the users who will use one of the enrollment functions needs **Read and Enroll** permissions on the templates they will be using through the portal (see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs on page 142](#) for more details).



**Tip:** Enrollment through the Keyfactor Command Management Portal against remote Microsoft CAs (CAs in a forest with either no trust with the forest in which Keyfactor





Command is installed or a one-way trust) are done in the context of the service account configured on the Management Portal CA record for *Explicit Credentials* (see [Create Service Accounts for Keyfactor Command on page 66](#)).

The Keyfactor Command Management Portal offers the option of using a different set of CAs and templates for each of the two different enrollment methods—PFX and CSR. As you collect your list of CAs and templates, you will need to decide whether you want to use the same CAs and templates for both types of enrollment or whether each type of enrollment will have a unique list of CAs and templates.

The list of templates used for enrollment in the Keyfactor Command Management Portal is configured through the Keyfactor Command Management Portal template management. Although in previous releases of Keyfactor Command, the templates and CAs for enrollment were configured during installation, this is now done as a post-install step in the Management Portal. See *Certificate Authorities* and *Configuring Template Options* in the *Keyfactor Command Reference Guide*.

#### 2.4.4.5 Configure Renewal Handler Permission


The expiration renewal event handler allows you to execute a certificate renewal automatically for each expiring certificate that is found in a supported certificate store for each expiration alert when the alert task is triggered by the execution of the expiration alerts. In order for the renewal handler to execute successfully, the Active Directory service account under which the Keyfactor Command Service runs must have select permissions in the Keyfactor Command Management Portal. In addition, if you wish to test the execution of expiration alerts with renewal handlers and your IIS application pool runs in the context of a different Active Directory service account than the Keyfactor Command Service, the Active Directory service account for the Keyfactor API IIS application pool must also be granted these permissions.



**Note:** If your Microsoft CA has been configured with the *Use Explicit Credentials* option, the permissions described here need to be granted to the user specified by the *Use Explicit Credentials* option, not either of the above-referenced service accounts. If you're using an EJBCA CA, no further permissions need to be granted and this step may be skipped.

If you don't plan to use the expiration renewal handler, you can skip this step.

To configure permissions for the service account(s) to support use of the expiration renewal handler:

1. In the Keyfactor Command Management Portal, browse to *System Settings Icon*  > *Security Roles & Identities*.
2. On the Security Roles and Identities page on the Security Roles tab, click **Add** to create a new role to be used just to grant permissions to the service account(s) to support use of the expiration renewal handler.
3. On the Details tab, give it an appropriate name and description to reflect this usage.



4. On the Global Permissions tab:
  - a. Select *Certificate Enrollment* and click the **Enroll PFX** toggle to enable it.
  - b. Select *Certificate Store Management* and click the **Read** and **Schedule** toggles to enable them.
  - c. Select *Certificates* and click the **Read** toggle to enable it.
  - d. Select *Management Portal* and click the **Read** toggle to disable it, if enabled.
5. Click **Save** to save the role.
6. On the Security Roles and Identities page on the Security Identities tab, click **Add** to add a new security identity.
7. In the Security Identities dialog, enter the Active Directory user name of the service account under which the Keyfactor Command Service runs using DOMAIN\username format and click **Save**. If the account resolves correctly, the new identity will be saved and the dialog will close.

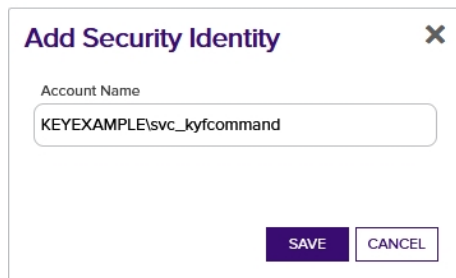


Figure 98: Configure Expiration Renewal Handler: Add New Identity

8. If your IIS application pool runs as a different Active Directory service account from that used for the Keyfactor Command Service, repeat steps six and seven for the IIS application pool service account.
9. In the Security Identity Editor section of the page, double-click the Keyfactor Command Service identity in the identity grid, right-click the row in the identity grid and choose **Edit Roles** from the right-click menu, or highlight the Keyfactor Command Service identity in the identity grid and click **Edit Roles** at the top of the identity grid.
10. In the Roles dialog, select the role you created for the expiration renewal handler in the Available Roles list and use the right arrow to move the role to the Current Roles list. Click **Save** to assign the role to the identity.

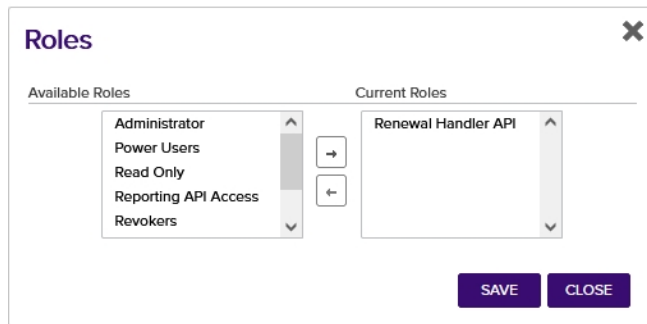


Figure 99: Configure Expiration Renewal Handler: Assign Role to Identity

11. If your IIS application pool runs as a different Active Directory service account from that used for the Keyfactor Command Service, repeat steps nine and ten for the IIS application pool service account.

#### 2.4.4.6 Create a Certificate Template for Mac Auto-Enrollment

This step only needs to be completed if your Keyfactor Command license includes Mac auto-enrollment and you plan to use this feature.

To create the certificate template that will be used for Mac auto-enrollment:

1. On the CA that will issue the Mac auto-enrollment certificates, open the Certification Authority management tool.
2. In the Certification Authority management tool, drill down to locate the Certificate Templates folder. Right-click the **Certificate Templates** folder and choose **Manage**. This will open the Certificate Templates Console.
3. In the Certificate Templates Console, right-click the User template and choose **Duplicate Template**.
4. If prompted with a Duplicate Template dialog (some versions of Windows), choose Windows Server 2003 Enterprise and click **OK**.
5. General Tab: In the Properties of New Template dialog on the General tab, enter **Mac Auto-Enrollment** (or an alternate name of your choosing) in the **Template display name** field. The **Template name** will be auto-populated based on the text you enter in the **Template display name**. Select a **Validity period** for the certificate that's appropriate for your environment.
6. Extensions Tab: If you plan to use the certificates to authenticate to enterprise systems, you will need to ensure that **Client Authentication** is set as the only application policy in the certificate. To do this, in the **Extensions included in this template** section of the Extensions tab, highlight **Application Policies** and click the **Edit...** button. In the Edit Application Policies Extensions dialog, remove the **Encrypting File System** and **Secure Email** policies and click **OK**.

7. Security Tab: In the Properties of New Template dialog on the Security tab, add the Active Directory group of users who will be allowed to auto-enroll from Macs and grant this group **Read**, **Enroll**, and **Autoenroll** permissions on the template.
8. Click **OK** to save the new template.
9. Back in the Certification Authority management tool, right-click the **Certificate Templates** folder and choose **New->Certificate Template to Issue**. Select the **Mac Auto-Enrollment** template from the list presented and click **OK**.

## 2.5 Keyfactor CA Policy Module

The Keyfactor CA Policy Module includes four certificate authority policy handlers that can be used to alter or restrict the functionality of a Microsoft certificate authority. The policy handlers are installed on the Microsoft CA and enabled through the Microsoft CA properties page. The available policy handlers are:

### RFC 2818 Policy Handler

Automate inclusion of a DNS SAN matching the CN of the requested certificate in certificate enrollments for a defined set of CA templates.

### SAN Attribute Policy Handler

Allow the addition of SANs not included in the CSR when making a CSR enrollment request. The added SANs will overwrite any existing SANs in the CSR. This functionality is the same as that seen with the Microsoft default policy module for the CA as a whole when the CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag is set except the SAN Attribute Policy Handler provides the ability to control SAN addition on a template-by-template basis without the need to enable the Microsoft CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag.

### vSCEP™ Policy Handler

Allow secure control of on-device key generation during certificate enrollment for iOS and Mac devices.

### Whitelist Policy Handler

Enforce that certificate requests for a given template or templates can only be initiated from a given computer or set of computers.



**Important:** If you're upgrading from a previous version of the Keyfactor CA Policy Module, refer to the *Keyfactor Command Upgrade Overview* for important upgrade instructions and a required upgrade script. Newer versions of Keyfactor CA Policy Module **cannot** be installed over the top of existing Keyfactor CA Policy Module installations to complete an upgrade.

## 2.5.1 System Requirements

The Keyfactor CA Policy Module is supported on Microsoft certificate authorities running on Windows Server 2016 or higher. It interoperates with Keyfactor Command versions 9.0 or greater.

The policy module requires the Microsoft .NET **Desktop** Runtime version 6.0 (x64). Version 6.0 is available for download from Microsoft:

<https://dotnet.microsoft.com/download/dotnet/6.0/runtime>

At the above link, this would be the **Download x64** option under the “Run desktop apps” heading.

You can use the following PowerShell command to check the .NET core version(s) installed on a server (if any):

```
dotnet --list-runtimes
```

Output from this command will look something like this if you have the correct 6.0 x64 version of the .NET Desktop Runtime installed (notice the paths are in C:\Program Files, not C:\Program Files (x86), indicating this is the x64 version):

```
Microsoft.NETCore.App 6.0.11 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]  
Microsoft.WindowsDesktop.App 6.0.11 [C:\Program  
Files\dotnet\shared\Microsoft.WindowsDesktop.App]
```

The policy module also requires a Keyfactor Command license key for the current release with a policy module license.



**Important:** If you’re upgrading from a previous version of the Keyfactor CA Policy Module, refer to the *Keyfactor Command Upgrade Overview* for important upgrade instructions and a required upgrade script. Newer versions of Keyfactor CA Policy Module **cannot** be installed over the top of existing Keyfactor CA Policy Module installations to complete an upgrade.

## 2.5.2 Preparing for the Keyfactor CA Policy Module

The preparation steps necessary for the Keyfactor CA Policy Module vary depending on the policy handler(s) you intend to use.



**Important:** If you’re upgrading from a previous version of the Keyfactor CA Policy Module, refer to the *Keyfactor Command Upgrade Overview* for important upgrade instructions and a required upgrade script. Newer versions of Keyfactor CA Policy Module **cannot** be installed over the top of existing Keyfactor CA Policy Module installations to complete an upgrade.

The policy handlers have the following preparation requirements:

## RFC 2818 Policy Handler

During the configuration of the RFC 2818 Policy Handler, you will need to define the list of Microsoft certificate templates that will automatically be assigned a DNS SAN matching the certificate's CN when a certificate enrollment request reaches the CA. These templates are configured by selecting them from a list. You will need to have this list of templates ready.

## SAN Attribute Policy Handler

During the configuration of the SAN Attribute Policy Handler, you will need to define the list of Microsoft certificate templates that will allow certificate enrollment requests via CSR to submit SANs outside of the CSR for inclusion in the final certificate, replacing any SANs originally in the CSR. These templates are configured by selecting them from a list. You will need to have this list of templates ready.

## vSCEP™ Policy Handler

During the configuration of the vSCEP™ Policy Handler, you will need to enter the URL to the vSCEP service on your Keyfactor Command server and the username and password of a service account that the policy handler will use to make requests to the vSCEP API on the Keyfactor Command server. The user you enter here needs to be a member of the group you configure for *Allowed Users/Groups* on the vSCEP Service tab in the Keyfactor Command configuration wizard (see [Install the Keyfactor Command Components on the Keyfactor Command Server\(s\) on page 90](#)). The service account needs to be created in Active Directory prior to installation of the Keyfactor CA Policy Module software, and the person installing the Keyfactor CA Policy Module software needs to know the service account domain, username and password.

## Whitelist Policy Handler

During the configuration of the SAN Attribute Policy Handler, you will need to define the list of Microsoft certificate templates that will be gated by the handler and the list of machines that will be allowed to use these templates. Any templates you include will be available for enrollment only from machines you include in the allowed list. The purpose of this handler is to force all enrollments to be made from the Keyfactor Command server(s), so the list of machines should include your Keyfactor Command server(s). The templates for this policy handler are configured by typing in their certificate *template name* (short name), so you will need an exact list of the template names.

In addition, you will need to have your Keyfactor product license available for upload into the policy module once installed to activate it.

## 2.5.3 Installing the Keyfactor CA Policy Module Handlers

These steps only need to be completed if your Keyfactor Command license includes the Keyfactor CA Policy Module and you plan to use this feature and one or more of its policy handlers. Review the policy handlers to determine if one or more of them meets a need in your environment.



**Important:** For a CA Clustered solution, if the Keyfactor CA Policy Module is installed on a node then configured, then failed over to another node, this will corrupt the check point key. The module must be installed on BOTH nodes, configured on one node, then failed over to the other node.

The available policy handlers are:

## RFC 2818 Policy Handler

Automate inclusion of a DNS SAN matching the CN of the requested certificate in certificate enrollments for a defined set of CA templates.

## SAN Attribute Policy Handler

Allow the addition of SANs not included in the CSR when making a CSR enrollment request. The added SANs will overwrite any existing SANs in the CSR. This functionality is the same as that seen with the Microsoft default policy module for the CA as a whole when the CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag is set except the SAN Attribute Policy Handler provides the ability to control SAN addition on a template-by-template basis without the need to enable the Microsoft CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag.



**Important:** By default, Microsoft CAs do not support the addition of SANs not included in the CSR when making a request using a CSR enrollment method. To enable your CA to support requesting certificates with additional SANs, you must either install and configure the Keyfactor Command SAN Attribute Policy Handler on the CA(s) or enable the Microsoft CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag. There are security risks inherent in enabling either of these options on your CA. Keyfactor recommends that you do not enable these options unless it is an absolute requirement. With the SAN Attribute Policy Handler, you can limit the risk by limiting the exposure to just selected templates. Keyfactor further recommends that you:

- Use the SAN Attribute Policy Handler only with templates that require CA manager approval so that a manager will be required to review the request and the added SANs before the certificate is issued.
- Use the SAN Attribute Policy Handler in conjunction with the Whitelist Policy Handler to limit requests for the selected templates to being initiated only by the Keyfactor Command server(s).
- Configure server level monitoring with a product such as Microsoft's System Center Operations Manager (SCOM) to provide alerts for any changes relating to the CA(s) configured with the SAN Attribute Policy Handler so that, for example, changes to the templates configured to support SAN addition do not go unnoticed.

## vSCEP™ Policy Handler

Allow secure control of on-device key generation during certificate enrollment for iOS and Mac devices.

## Whitelist Policy Handler

Enforce that certificate requests for a given template or templates can only be initiated from a given computer or set of computers.



**Note:** The following Windows update affects how certificate requests are built when sent to a Microsoft CA and may cause enrollments done outside Keyfactor Command against a Microsoft CA configured with the Whitelist Policy Handler to fail.

<https://support.microsoft.com/en-us/topic/kb5014754-certificate-based-authentication-changes-on-windows-domain-controllers-ad2c23b0-15d8-4340-a468-4d4f3b188f16>

The processing order of the handlers currently available in the Keyfactor CA Policy Module, when used together on the same machine, is significant for some handlers and not others. Specifically, the processing order is not significant for the vSCEP™ Policy Handler and Machine Whitelist Policy handler. These handlers may be placed anywhere within the list of handlers. However, the processing order does matter for the SAN Attribute Policy Handler and the RFC 2818 Policy Handler. When these two handlers are used together, the SAN Attribute Policy Handler must be placed on the list above the RFC 2818 Policy Handler to allow the SAN Attribute Policy Handler to be processed before the RFC 2818 Policy Handler. This is because the SAN Attribute Policy Handler removes any existing SANs on the enrollment request and replaces them with those specified in the request outside of the CSR—such as those entered in the optional SAN section on the CSR page of the Keyfactor Command Management Portal. This includes any SANs added by the RFC 2818 Policy Handler.

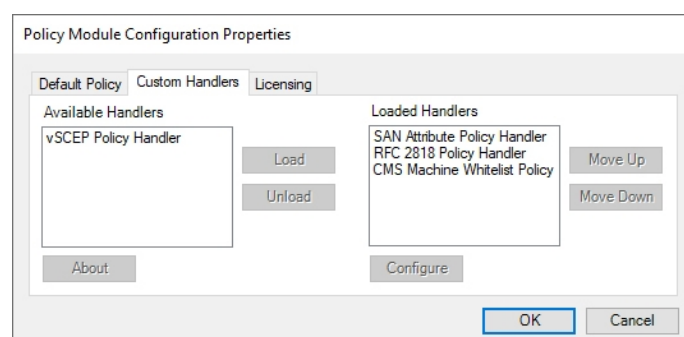


Figure 100: Keyfactor CA Policy Module Policy Module Handler Ordering

When the Keyfactor CA Policy Module is used, the policy module listed on the Default Policy tab of the Policy Module Configuration Properties dialog is run first when a request reaches the CA. This default policy might be the standard Windows default, as shown [Figure 101: Default Policy Module](#), or it might be another non-built-in policy module, such as the Microsoft FIM CM Policy Module. After the default policy module runs, the Loaded Handlers on the Custom Handlers tab of the Policy Module Configuration Properties dialog are run in the order listed (top to bottom). After all the handlers have been run, the result (approved, denied, or marked as pending) is returned to the CA for processing.

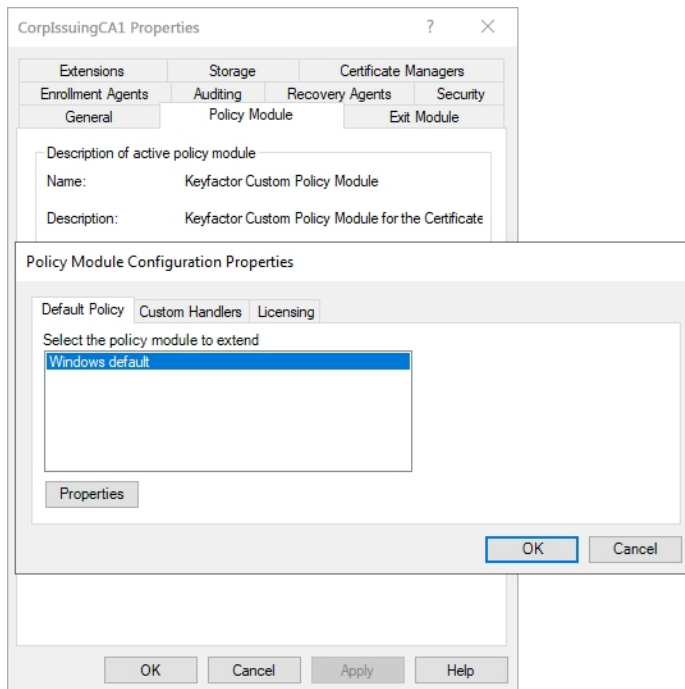


Figure 101: Default Policy Module



**Tip:** Once the installation is complete, the configuration options for the policy handlers can be found in the registry on the CA in the following paths (where CA\_LOGICAL\_NAME is the logical name of the local CA):

```
HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\ services\ CertSvc\ Configuration\
CA_LOGICAL_NAME\ PolicyModules\ CMS_Custom.Policy\ PolicyHandlers\
RFC2818.PolicyHandler
```

```
HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\ services\ CertSvc\ Configuration\
CA_LOGICAL_NAME\ PolicyModules\ CMS_Custom.Policy\ PolicyHandlers\
SANAttribute.PolicyHandler
```

```
HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\ services\ CertSvc\ Configuration\
CA_LOGICAL_NAME\ PolicyModules\ CMS_Custom.Policy\ PolicyHandlers\
CMSWhitelist.PolicyHandler
```



**Important:** These registry keys should not be modified without advice from Keyfactor support.



### 2.5.3.1 Install the Keyfactor RFC 2818 Policy Handler

To begin the RFC 2818 Policy Handler installation, execute the KeyfactorCAModuleInstaller.msi file from the Keyfactor installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.

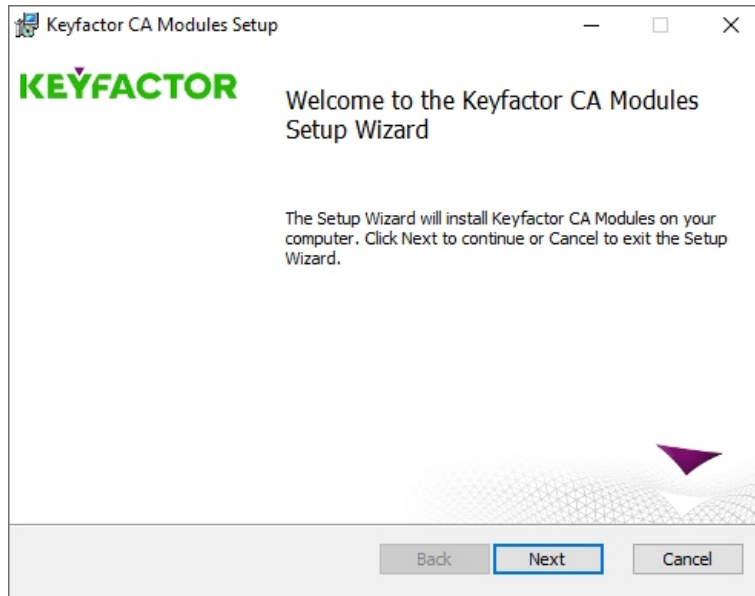


Figure 102: Install RFC 2818 Policy Handler: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the components to install. For the RFC 2818 Policy Handler, deselect all the components except the RFC 2818 Policy Handler component. If desired, you can highlight Keyfactor Custom Policy Module and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor CA Modules

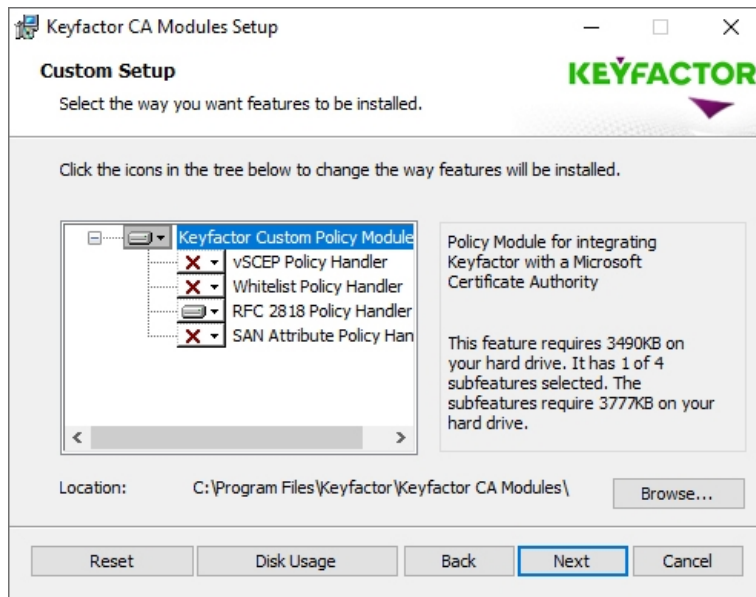


Figure 103: Install RFC 2818 Policy Handler: Select Components

4. On the next screen, click **Install**.
5. On the final installation wizard page, leave the *Launch the CA MMC snap-in now* box selected and click **Finish**. The Microsoft Certification Authority management tool should start automatically. This can take several seconds.
6. In the Certification Authority management tool, right-click the CA name at the top of the tree and choose **Properties**.
7. In the Properties dialog for the CA on the CA Policy Module tab, click **Select**, highlight the **Keyfactor Custom Policy Module** in the Set Active Policy Module dialog and click **OK**.

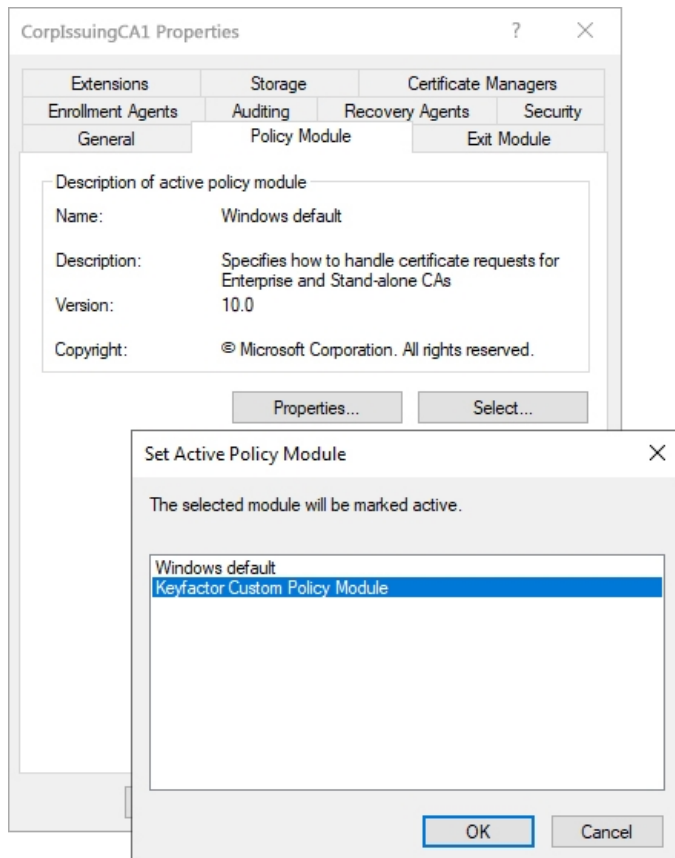


Figure 104: Enable the Keyfactor CA Policy Module

8. In the Properties dialog for the CA on the CA Policy Module tab, click **Properties**.
9. On the Licensing tab of the Policy Module Configuration Properties page, click **Upload License** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE.

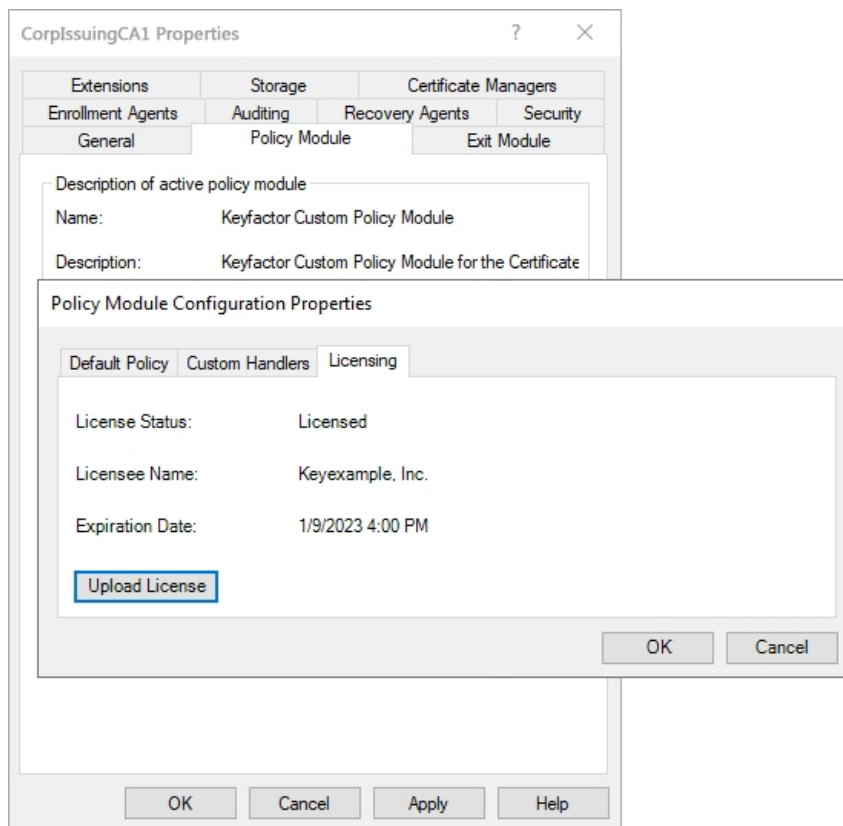


Figure 105: Upload the Keyfactor CA Policy Module License

10. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the **RFC 2818 Policy Handler** under Loaded Handlers, click **Load** to move it over to the loaded handlers, and click **OK**.

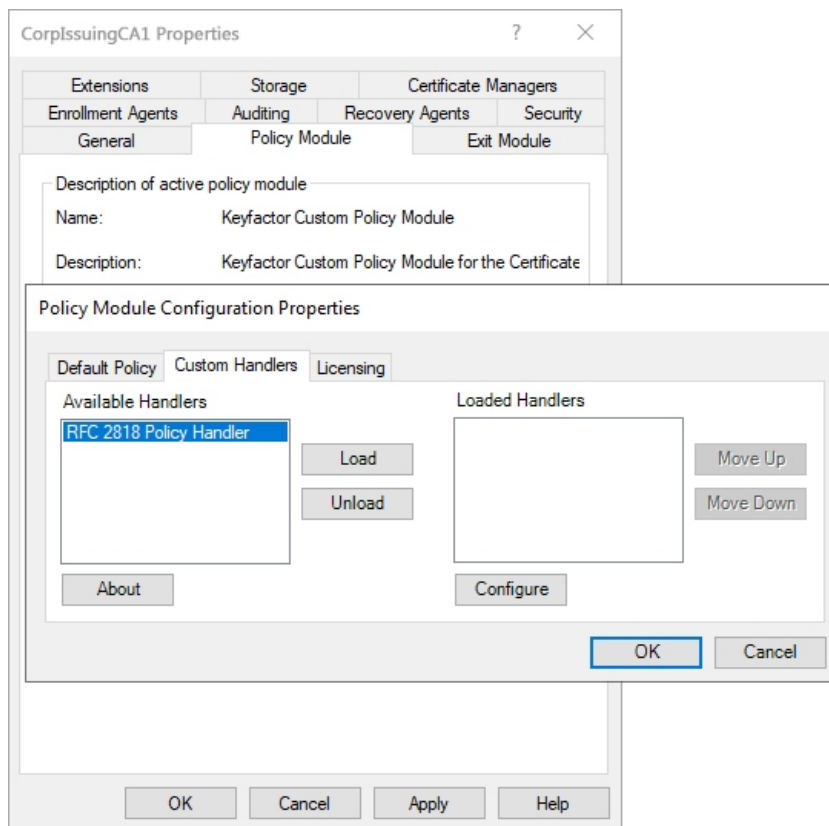


Figure 106: Enable the RFC 2818 Policy Handler

11. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the RFC 2818 Policy Handler under Loaded Handlers and click **Configure**.
12. On RFC 2818 Policy Handler configuration dialog, select the templates that should be under management by the RFC 2818 policy handler and click **Add**. Certificate enrollments from any source made using the templates selected here on the configured CA will automatically be assigned a DNS SAN matching the certificate's CN.

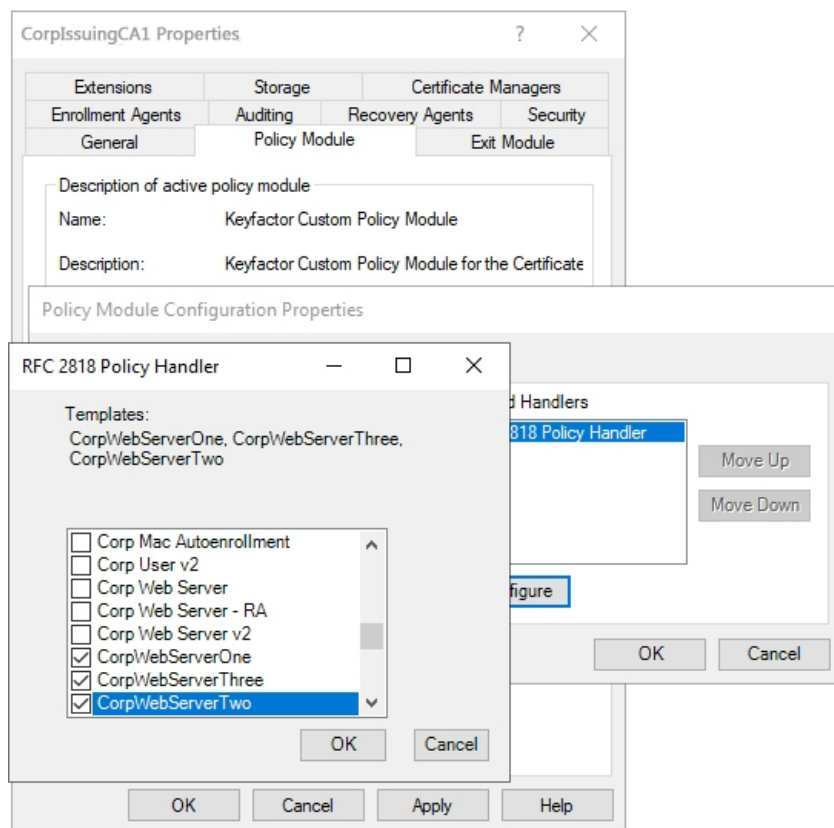


Figure 107: Add Templates for Management with the RFC 2818 Policy Handler

13. Click **OK** as many times as needed to close the configuration dialogs and save the configuration. You will be prompted to restart the CA services.

### 2.5.3.2 Install the Keyfactor SAN Attribute Policy Handler

To begin the SAN Attribute Policy Handler installation, execute the KeyfactorCAModuleInstaller.msi file from the Keyfactor installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.

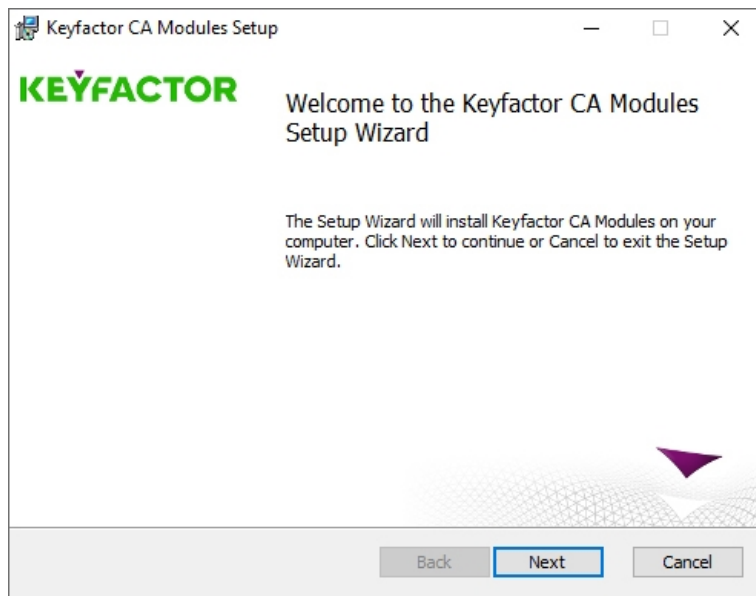


Figure 108: Install SAN Attribute Policy Handler: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the components to install. For the SAN Attribute Policy Handler, deselect all the components except the SAN Attribute Policy Handler component. If desired, you can highlight Keyfactor Custom Policy Module and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor CA Modules

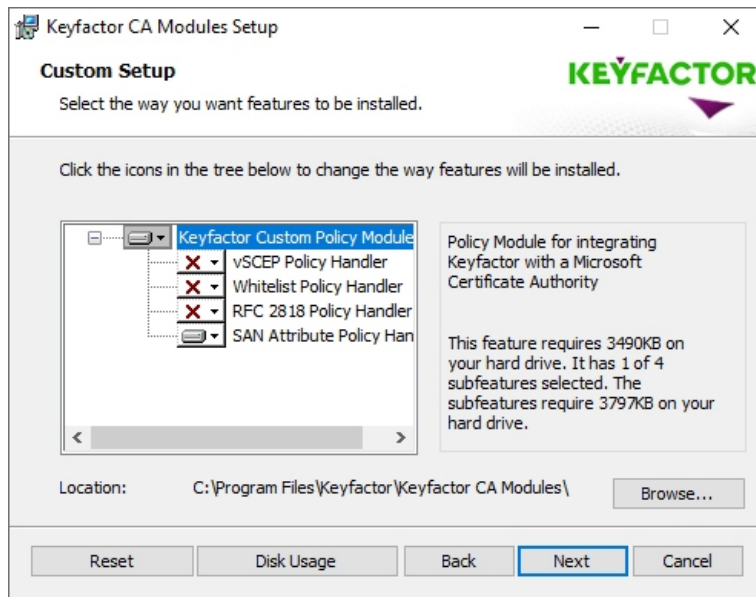


Figure 109: Install SAN Attribute Policy Handler: Select Components

4. On the next screen, click **Install**.
5. On the final installation wizard page, leave the *Launch the CA MMC snap-in now* box selected and click **Finish**. The Microsoft Certification Authority management tool should start automatically. This can take several seconds.
6. In the Certification Authority management tool, right-click the CA name at the top of the tree and choose **Properties**.
7. In the Properties dialog for the CA on the CA Policy Module tab, click **Select**, highlight the **Keyfactor Custom Policy Module** in the Set Active Policy Module dialog and click **OK**.



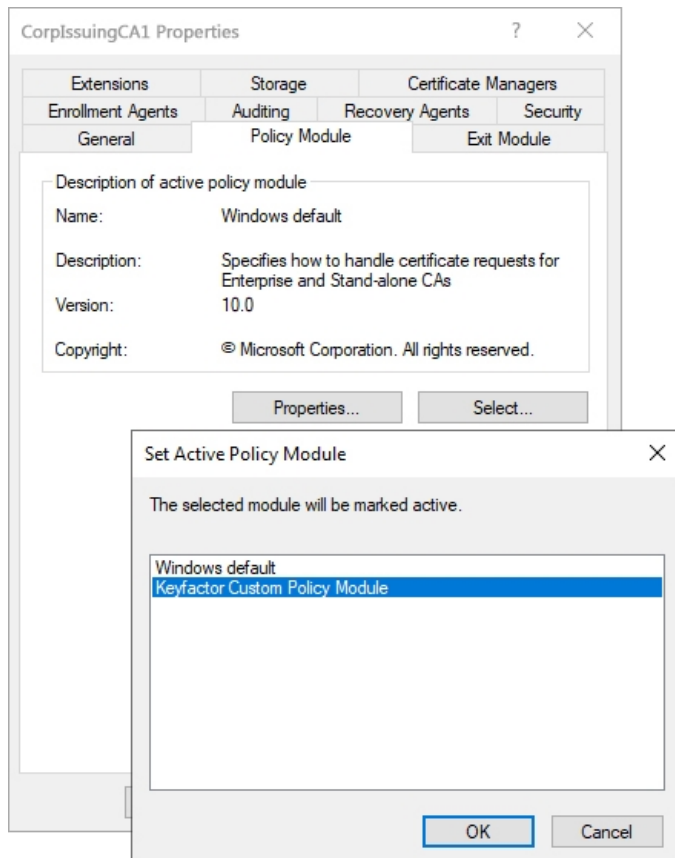


Figure 110: Enable the Keyfactor CA Policy Module

8. In the Properties dialog for the CA on the CA Policy Module tab, click **Properties**.
9. On the Licensing tab of the Policy Module Configuration Properties page, click **Upload License** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE.

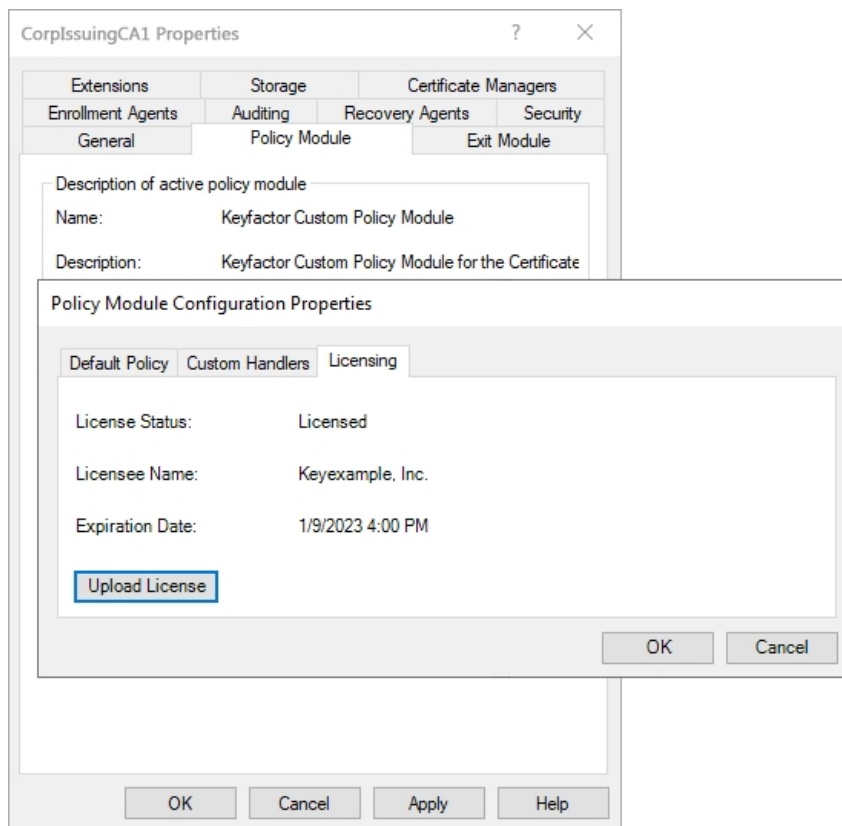


Figure 111: Upload the Keyfactor CA Policy Module License

10. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the **SAN Attribute Policy Handler** under Loaded Handlers, click **Load** to move it over to the loaded handlers, and click **OK**.

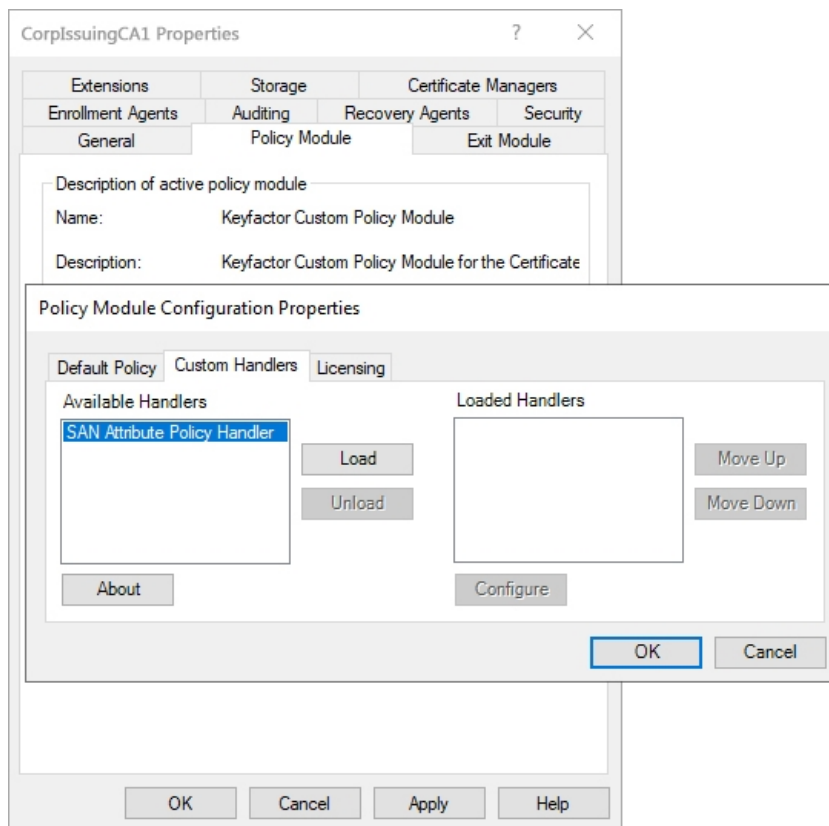


Figure 112: Enable the SAN Attribute Policy Handler

11. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the SAN Attribute Policy Handler under Loaded Handlers and click **Configure**.
12. On SAN Attribute Policy Handler configuration dialog, select the templates that should be under management by the SAN Attribute policy handler and click **Add**. Certificate enrollments from any source made using the templates selected here on the configured CA and a CSR enrollment method will allow the addition of SANs not included in the CSR and control the SAN addition functionality on a template-by-template basis without the need to enable the Microsoft CA EDITF\_ATTRIBUTESUBJECTALTNAME2 flag.

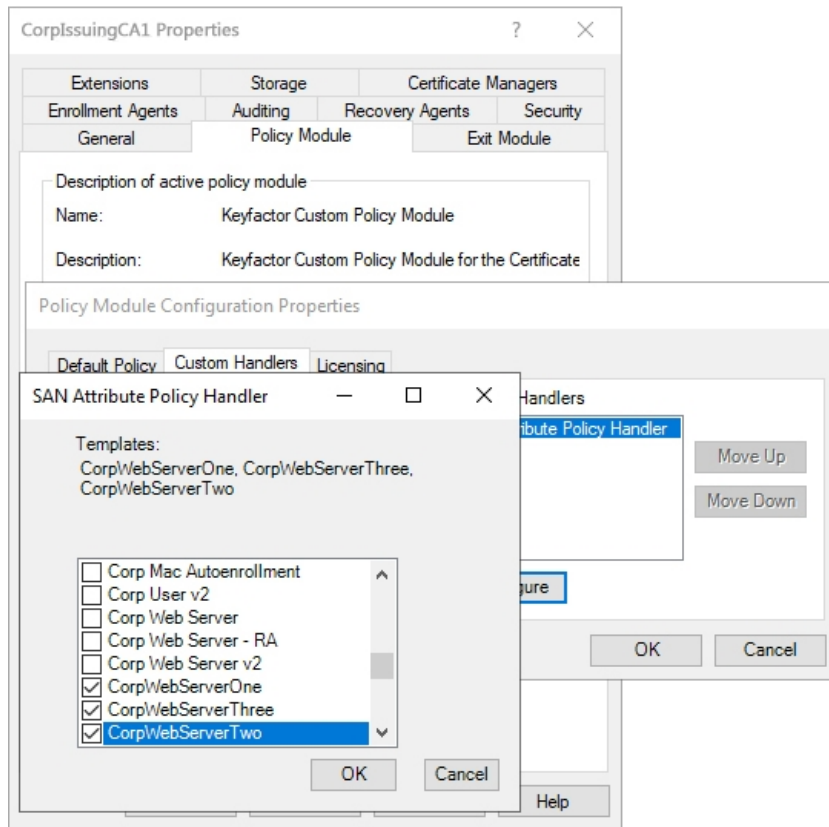


Figure 113: Add Templates for Management with the SAN Attribute Policy Handler

13. Click **OK** as many times as needed to close the configuration dialogs and save the configuration. You will be prompted to restart the CA services.

### 2.5.3.3 Install the Keyfactor Whitelist Policy Handler

To begin the Whitelist Policy Handler installation, execute the KeyfactorCAModuleInstaller.msi file from the Keyfactor installation media and install as follows.



**Note:** The following Windows update affects how certificate requests are built when sent to a Microsoft CA and may cause enrollments done outside Keyfactor Command against a Microsoft CA configured with the Whitelist Policy Handler to fail.

<https://support.microsoft.com/en-us/topic/kb5014754-certificate-based-authentication-changes-on-windows-domain-controllers-ad2c23b0-15d8-4340-a468-4d4f3b188f16>

1. On the first installation page, click **Next** to begin the setup wizard.

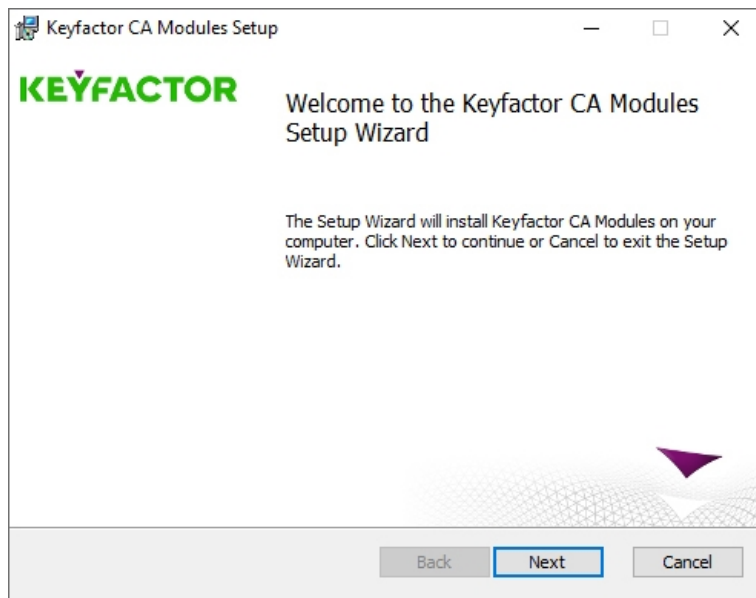


Figure 114: Install Whitelist Policy Handler: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the components to install. For the Whitelist Policy Handler, deselect all the components except the Whitelist Policy Handler component. If desired, you can highlight Keyfactor Custom Policy Module and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor CA Modules

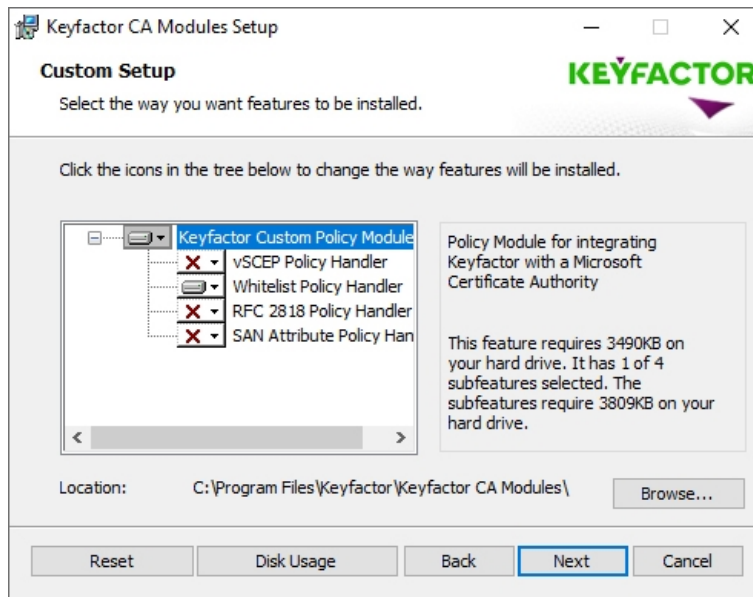


Figure 115: Install Whitelist Policy Handler: Select Components

4. On the next screen, click **Install**.
5. On the final installation wizard page, leave the *Launch the CA MMC snap-in now* box selected and click **Finish**. The Microsoft Certification Authority management tool should start automatically. This can take several seconds.
6. In the Certification Authority management tool, right-click the CA name at the top of the tree and choose **Properties**.
7. In the Properties dialog for the CA on the CA Policy Module tab, click **Select**, highlight the **Keyfactor Custom Policy Module** in the Set Active Policy Module dialog and click **OK**.

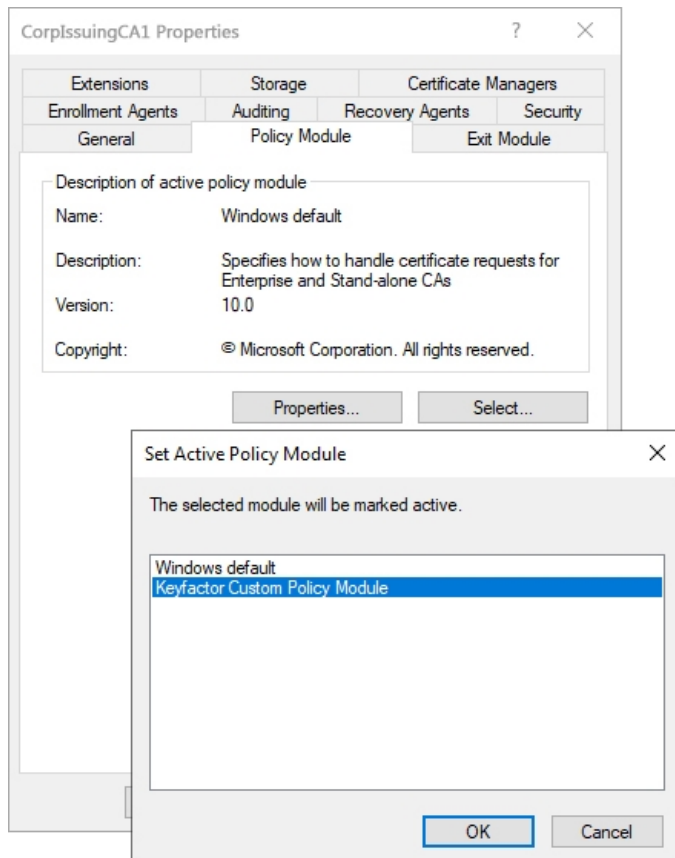


Figure 116: Enable the Keyfactor CA Policy Module

8. In the Properties dialog for the CA on the CA Policy Module tab, click **Properties**.
9. On the Licensing tab of the Policy Module Configuration Properties page, click **Upload License** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE.

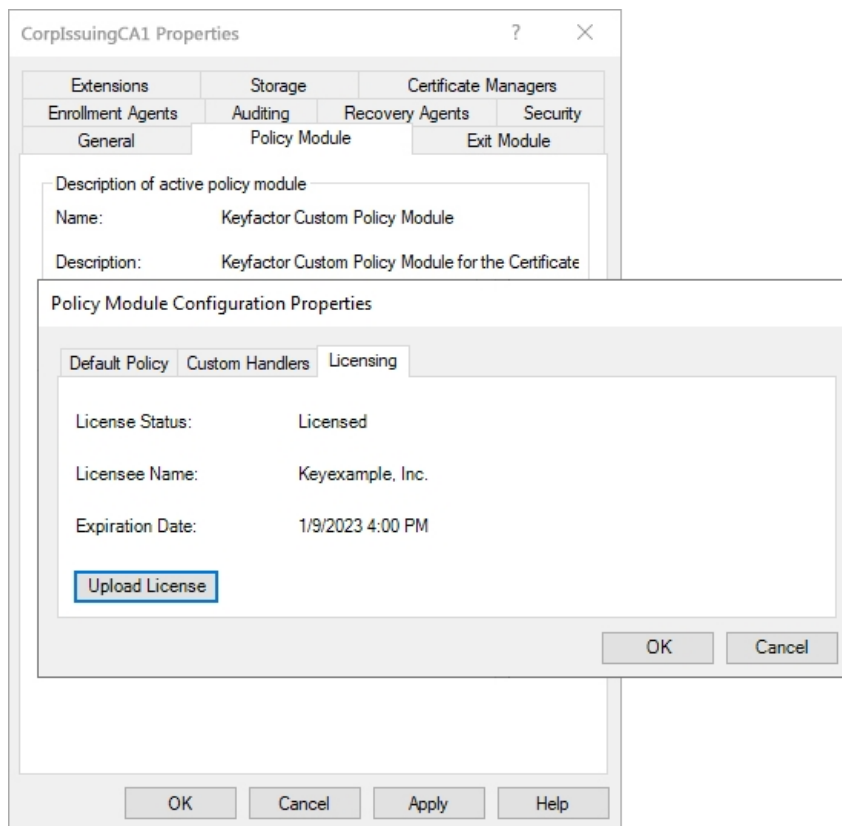


Figure 117: Upload the Keyfactor CA Policy Module License

10. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the **CMS Machine Whitelist Policy** on the list of available handlers, click **Load** to move it over to the loaded handlers, and click **OK**.



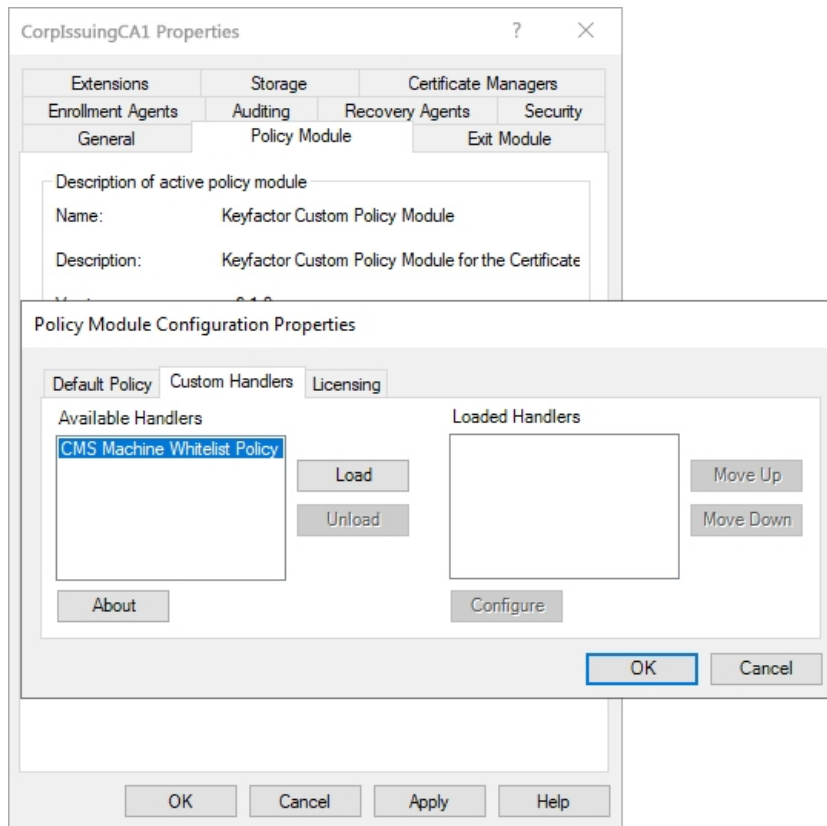


Figure 118: Enable the Whitelist Policy Handler

11. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight CMS Machine Whitelist Policy under Loaded Handlers and click **Configure**.
12. On the Template tab of the Policy Module Configuration dialog, enter the certificate *template names* (short names), not the template display names, one at a time, of the certificate template(s) you want to manage with the whitelist policy handler and click **Add**. In many cases, the template name is the same as the template display name with the spaces removed. Any templates entered here will be available for enrollment only from machines listed on the Machine Names tab.

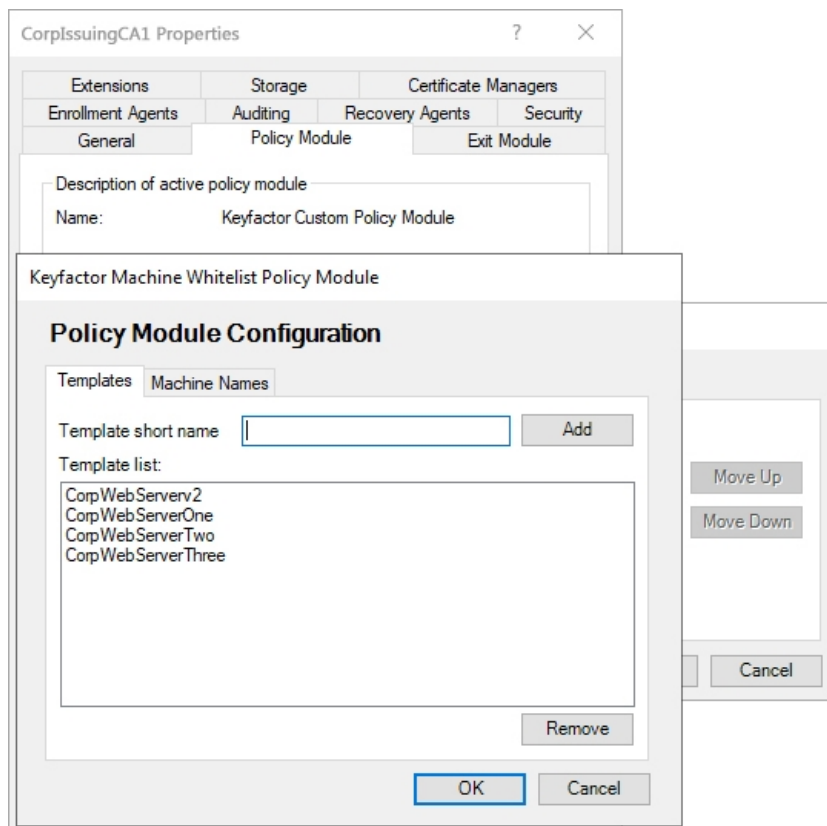


Figure 119: Add Templates for Management with the Whitelist Policy Handler

13. On the Machine Names tab of the Policy Module Configuration dialog, enter the machine names (FQDNs), one at a time, of the machines that you want to manage with the whitelist policy handler and click **Add**. Any machines entered here will be allowed to enroll for the templates listed on the Templates tab.

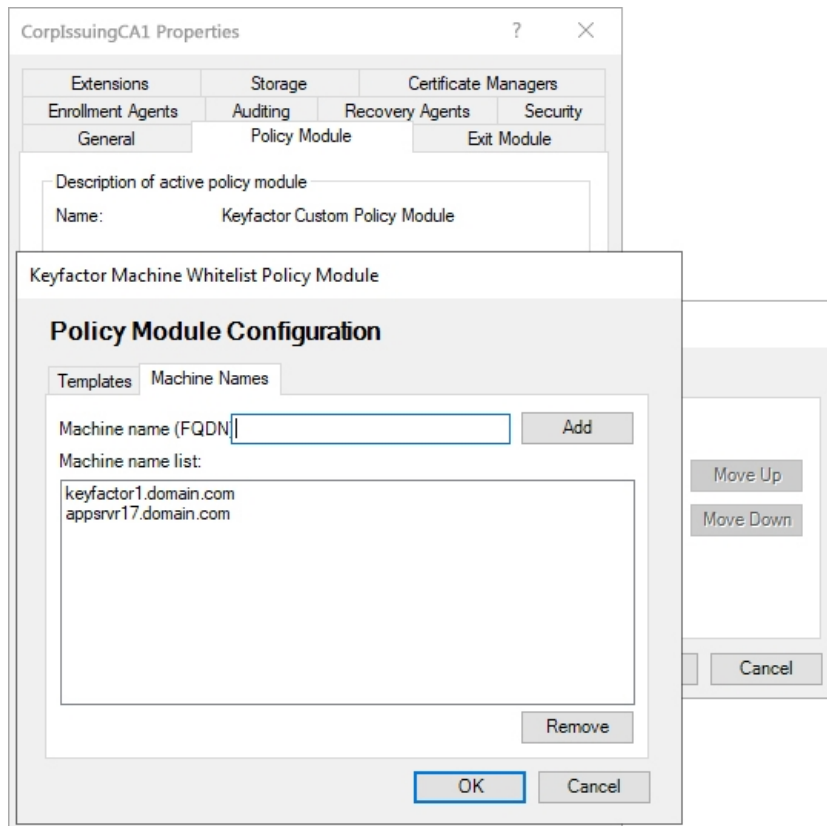


Figure 120: Add Machines for Management with the Whitelist Policy Handler

14. Click **OK** as many times as needed to close the configuration dialogs and save the configuration. You will be prompted to restart the CA services.

## 2.5.4 Configure Logging for the Keyfactor CA Policy Module

The Keyfactor CA Policy Module provides extensive logging for visibility and troubleshooting. By default, Keyfactor CA Policy Module places its log files in the C:\Keyfactor\logs directory, generates logs at the *Info* logging level and stores the primary logs for two days before deleting them.

To configure logging:

1. On the policy module server where you wish to adjust logging, open a text editor (e.g. Notepad) using the “Run as administrator” option.
2. In the text editor, browse to open the NLog.config file for the Keyfactor CA Policy Module. The file is located in the installation directory for the product, which is the following by default:

C:\Program Files\Keyfactor\Keyfactor CA Modules\NLog.config

3. Your Nlog.config file may have a slightly different layout than shown here, but it will contain the five fields highlighted in [Figure 121: Keyfactor CA Policy Module NLog.config File](#). The fields you may wish to edit are:

- `fileName="C:\Keyfactor\Logs\Keyfactor_CA_Log.txt"`

The path and file name of the active policy module log file, referencing the logDirectory variable.



**Important:** If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant the service account under which the Active Directory Certificate Services service is running full control permissions on this directory.

- `archiveFileName="C:\Keyfactor\Logs\Keyfactor_CA_Log_Archive_{#}.txt"`

The path and file name of previous days' orchestrator log files, referencing the logDirectory variable. The orchestrator rotates log files daily and names the previous files using this naming convention.

- `maxArchiveFiles="2"`

The number of archive files to retain before deletion.

- `name="*" minlevel="Info" writeTo="logfile"`

The level of log detail that should be generated and output to the log file. The default *Info* level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to *Debug* or *Trace*. Available log levels (in order of increasing verbosity) are:

- OFF—No logging
- FATAL—Log severe errors that cause early termination
- ERROR—Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN—Log errors and use of deprecated APIs, poor use of APIs, *almost* errors, and other runtime situations that are undesirable or unexpected but not necessarily *wrong*
- INFO—Log all of the above plus runtime events (startup/shutdown)
- DEBUG—Log all of the above plus detailed information on the flow through the system
- TRACE—Maximum log information—this option can generate VERY large log files

```

<targets>
  <target name="buffered_wrapper" xsi:type="BufferingWrapper" slidingTimeout="true" bufferSize="500" flushTimeout="500">
    <target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Keyfactor CA Log.txt" layout="{$(longdate)} ${(logger)} [{level}] - {message}"
      archiveFileName="C:\Keyfactor\logs\Keyfactor_CA_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="2"/>
    </target>
  </target>
  <target xsi:type="OutputDebugString" name="String" layout="{$(longdate)} ${(logger)}:{message}" />
  <target xsi:type="Debugger" name="debugger" layout="{$(longdate)} ${(logger)}:{message}" />
  <target xsi:type="Console" name="console" layout="{$(logger)} {message}" />
  <target xsi:type="EventLog" name="eventLog" source="Keyfactor CA Modules"
    eventId="{event-properties:item=eventID}" category="{event-properties:item=categoryID}" layout="{$(event-properties:item=message)}" />
</targets>
<rules>
  <!-- Don't write events to the log file (log file should contain different, more verbose, logging) -->
  <logger name="*-EVENT" minlevel="Info" writeTo="eventLog" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
</rules>

```

Figure 121: Keyfactor CA Policy Module NLog.config File

## 2.5.5 Add Non-Keyfactor SCEP Servers to the Ignore List

This step only needs to be completed if you installed the Keyfactor CA Policy Module with the vSCEP™ Policy Handler.

If your CA for that issues certificates based on SCEP challenges is used by multiple SCEP servers, you will need to add SCEP servers not used for Keyfactor Command vSCEP API requests to the ignore list on the CA running the vSCEP™ Policy Handler. This will allow the vSCEP™ Policy Handler to ignore requests (passing them through to the CA) from the listed SCEP servers. Without this feature, SCEP challenges from non-Keyfactor Command servers would be denied because no data exists against which to verify the certificate details.

The vSCEP™ Policy Handler reads the ignore list in the registry string value **CCMBlacklist**. This field contains a semicolon-delimited list of SCEP server host names from which all certificate requests should be ignored by the Keyfactor CA Policy Module and passed through to the CA. The **CCMBlacklist** setting can be found in the following registry location:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Certified Security Solutions\vSCEP\Configuration

The **CCMBlacklist** registry value does not exist by default. Use the Registry Editor (regedit) to create the **CCMBlacklist** value as a DWORD and populate it with the SCEP server FQDNs for any SCEP servers whose requests should bypass the vSCEP™ Policy Handler.

## 2.6 Appendices

- [Appendix - Troubleshooting Logi Log Files below](#)
- [Appendix - Logi Load Balancing: Keyfactor Command Configuration Wizard Setup on page 181](#)
- [Appendix - Configuration Wizard Errors in the Logs on page 183](#)

### 2.6.1 Appendix - Troubleshooting Logi Log Files

When troubleshooting Logi, the first thing to try is setting the *Debug Embedded Reports* application setting to **True** (see *Application Settings: Console Tab* in the *Keyfactor Command Reference Guide*). This allows the reports to output errors with debug level information if they generate errors. If this

does not generate the information necessary to resolve the problem, it can sometimes be helpful to modify the Keyfactor Analysis web.config file to allow IIS to show the actual error the application is experiencing at a lower level. To configure this:

1. Browse to the *Logi* directory under the installed directory for your Keyfactor Command implementation. By default, this is:

C:\Program Files\Keyfactor\Keyfactor Platform\Logi

2. Using a text editor opened with the “Run as administrator” option, open the web.config file for editing.
3. Find the <customErrors mode=”RemoteOnly”/> section and change this to <customErrors mode=”On”/>.
4. Look for the debug output in the *Logi\rdDownload* directory under the installed directory for your Keyfactor Command implementation. By default, this is:

C:\Program Files\Keyfactor\Keyfactor Platform\Logi\rdDownload



**Tip:** If you do not find debug output when running a report manually in the Management Portal, try scheduling a report for delivery via email or saving to disk using the Report Manager. Debugging operates differently for these two modes of running a report.



**Note:** When the portal experiences a 500 error, Logi logs will not be written to the usual output directory.

```
<system.web>
  <!-- DYNAMIC DEBUG COMPILATION
  Set compilation debug="true" to insert debugging symbols (.pdb information)
  into the compiled page. Because this creates a larger file that executes
  more slowly, you should set this value to true only when debugging and to
  false at all other times. For more information, refer to the documentation about
  debugging ASP.NET files.
  -->
  <compilation defaultLanguage="vb" debug="true" />

  <!-- CUSTOM ERROR MESSAGES
  Set customErrors mode="On" or "RemoteOnly" to enable custom error messages, "Off" to disable.
  Add <error> tags for each of the errors you want to handle.
  -->
  <customErrors mode="RemoteOnly" />
  <!-- AUTHENTICATION
  This section sets the authentication policies of the application. Possible modes are "Windows",
  "Forms", "Passport" and "None"
  -->
  <authentication mode="Windows" />
```

Figure 122: Logi web.config

## 2.6.2 Appendix – Logi Load Balancing: Keyfactor Command Configuration Wizard Setup

In order for the Keyfactor Command Management Portal Dashboard and Reports to load when using a load balancer, the Keyfactor Command Configuration Wizard should have the following configuration on each of the application servers:

- On the Keyfactor Command Portal Tab, the **Host Name** must be the load balanced URL.

The screenshot shows the 'Keyfactor Configuration Wizard' window. The left sidebar lists various configuration categories, with 'Keyfactor Portal' selected. The main pane displays settings for the 'Keyfactor Portal' tab. The 'Host Name' field is highlighted with a red rectangle and contains the value 'keyfactor.keyexample.com'. The 'Use SSL' checkbox is checked. Other visible settings include 'Web Site' set to 'Default Web Site', 'Virtual Directory' set to 'KeyfactorPortal', and 'Application Pool' set to 'CMS'. The 'Enrollment' section shows 'Certificate Subject Format' as 'CN={CN},E={E},O={O},OU={OU},L={L},ST={ST},C={C}'. The 'PFX Enrollment' section has 'Enabled' unchecked, 'PFX Password Type' set to 'Auto-Generated', and 'Alphanumeric Password Characters' checked. The 'Password Length' is set to 12. The 'CSR Enrollment' section is partially visible at the bottom. At the bottom of the wizard, there are buttons for 'Verify Configuration' and 'Cancel'. The status bar at the very bottom shows 'Server: sqlsrvr1.keyexample.com', 'Database Name: Keyfactor', and 'Credential Type: Windows'.

Figure 123: Logi Configuration Settings—Keyfactor Command Portal Tab

- On the Keyfactor Command Dashboard and Reports Tab, the **Host Name** must be the load balanced URL. This is the host name that the Management Portal server uses to connect to the Logi Analytics Platform, and it therefore needs to be the name used on the internal side of the network.

The screenshot shows the 'Keyfactor Configuration Wizard' window. On the left is a sidebar with various configuration categories, with 'Dashboard and Reports' selected. The main area displays configuration fields for the 'Application Pools' section. The 'Host Name' field is highlighted with a red rectangle and contains the text 'keyfactor243.keyexample.com'. To its right is a checked checkbox labeled 'Use SSL'. Below these are dropdown menus for 'Web Site' (set to 'Default Web Site'), 'Virtual Directory' (set to 'KeyfactorAnalysis'), and 'Application Pool' (set to 'KeyfactorReporting'). At the bottom of the wizard are 'Verify Configuration' and 'Cancel' buttons. A status bar at the very bottom displays 'Server: sqlsrvr1.keyexample.com', 'Database Name: Keyfactor', and 'Credential Type: Windows'.

Figure 124: Logi Configuration Settings—Keyfactor Command Dashboards and Reports Tab

- Load Balancer

On the load balancer, create a rewrite rule that changes the outbound URL from the application servers. Logi sends the `HostName.domain.com/KeyfactorAnalysis` URL back to the browser instead of the `LoadBalancer.URL.com/KeyfactorAnalysis` URL that the browser needs to complete the Logi authorization. In short, an outbound rewrite rule needs to be created on the load balancer that does the following: `HostName_URL/KeyfactorAnalysis` needs to be converted to `LoadBalancer_URL/keyfactorAnalysis`

- Load Balancer - IP Stickiness

On the load balancer, create a rule for scheduled reports such that if a request comes in to the load balancer from a given Keyfactor Command IP address, the request will be routed back to the same IP address to be completed.

- Load Balancer - Session Affinity

There are two load balancing scenarios based on user session management:

- Sticky sessions (recommended)

In the sticky session scenario, each user is assigned to a server by the load balancer and all the requests sent by this user are answered by the same server, for as long as the user's session persists. This is the recommended approach and does not require you to centralize the `rdDataCache` folder of the application. We strongly recommend using sticky sessions.

- Non-Sticky sessions

You can learn more about load balancing with info applications on:



## 2.6.3 Appendix - Configuration Wizard Errors in the Logs

If an incoming web request runs before the configuration is fully completed, you may encounter the following errors in the Management Portal log file after upgrading. These errors are not something to be worried about. They just indicate that the web request was still looking at an old version of the database prior to it being completely upgraded.

```
2021-11-15 12:41:36.5155 Keyfactor.EF.KeyfactorExecutionStrategy [Error] - SqlException with number 207 occurred, not attempting to retry the connection.
```

```
2021-11-15 12:41:36.5780 Keyfactor.EF.KeyfactorExecutionStrategy [Error] - Invalid column name 'Immutable'.
```

```
Invalid column name 'SubscriberTerms'.
```

```
2021-11-15 12:41:36.6249 ASP.global_asax [Error] - An uncaught application error occurred: An error occurred while executing the command definition. See the inner exception for details.
```

```
2021-11-15 12:41:37.3281 ASP.global_asax [Error] - An uncaught application error occurred: An item with the same key has already been added.
```

```
at System.ThrowHelper.ThrowArgumentException(ExceptionResource resource)
```

## 3.0 Glossary

### A

---

#### AIA

The authority information access (AIA) is included in a certificate--if configured--and identifies a location from which the chain certificates for that certificate may be retrieved.

#### AnyAgent

The AnyAgent, one of Keyfactor's suite of orchestrators, is used to allow management of certificates regardless of source or location by allowing customers to implement custom agent functionality via an API.

#### AnyGateway

The Keyfactor AnyGateway is a generic third party CA gateway framework that allows existing CA gateways and custom CA connections to share the same overall product framework.

#### API

A set of functions to allow creation of applications. Keyfactor offers the Keyfactor API, which allows third-party software to integrate with the advanced certificate enrollment and management features of Keyfactor Command.

#### Argument

A parameter or argument is a value that is passed into a function in an application.

#### Authority Information Access

The authority information access (AIA) is included in a certificate--if configured--and identifies a location from which the chain certificates for that certificate may be retrieved.

### B

---

#### Bash Orchestrator

The Bash Orchestrator, one of Keyfactor's suite of orchestrators, is used to discover and manage SSH keys across an enterprise.

#### Blueprint

A snapshot of the certificate stores and scheduled jobs on one orchestrator, which can be used to create matching certificate stores and jobs on another orchestrator with just a few clicks.

### C

---

#### CA

A certificate authority (CA) is an entity that issues digital certificates. Within Keyfactor Command, a CA may be a Microsoft CA or a Keyfactor gateway to a cloud-based or remote CA.

#### Certificate Authority

A certificate authority (CA) is an entity that issues digital certificates. Within Keyfactor Command, a CA may be a Microsoft CA or a Keyfactor gateway to a cloud-based or remote CA.

#### Certificate Revocation List

A Certificate Revocation List (CRL) is a list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date and should no longer be trusted.

#### Certificate Signing Request

A CSR or certificate signing request is a block of encoded text that is submitted to a CA when enrolling for a certificate. When you generate a CSR within Keyfactor

Command, the matching private key for it is stored in Keyfactor Command in encrypted format and will be married with the certificate once returned from the CA.

### CN

A common name (CN) is the component of a distinguished name (DN) that represents the primary name of the object. The value varies depending on the type of object. For a user object, this would be the user's name (e.g. CN=John Smith). For SSL certificates, the CN is typically the fully qualified domain name (FQDN) of the host where the SSL certificate will reside (e.g. server-name.keyexample.com or www.keyexample.com).

### Collection

The certificate search function allows you to query the Keyfactor Command database for certificates from any available source based on any criteria of the certificates and save the results as a collection that will be available in other places in the Management Portal (e.g. expiration alerts and certain reports).

### Common Name

A common name (CN) is the component of a distinguished name (DN) that represents the primary name of the object. The value varies depending on the type of object. For a user object, this would be the user's name (e.g. CN=John Smith). For SSL certificates, the CN is typically the fully qualified domain name (FQDN) of the host where the SSL certificate will reside (e.g. server-name.keyexample.com or www.keyexample.com).

### Configuration Tenant

A grouping of CAs. The Microsoft concept of forests is not used in EJBCA so to

accommodate the new EJBCA functionality, and to avoid confusion, the term forest needed to be renamed. The new name is configuration tenant. For EJBCA, there would be one configuration tenant per EJBCA server install. For Microsoft, there would be one per forest. Note that configuration tenants cannot be mixed, so Microsoft and EJBCA cannot exist on the same configuration tenant.

### CRL

A Certificate Revocation List (CRL) is a list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date and should no longer be trusted.

### CSR

A CSR or certificate signing request is a block of encoded text that is submitted to a CA when enrolling for a certificate. When you generate a CSR within Keyfactor Command, the matching private key for it is stored in Keyfactor Command in encrypted format and will be married with the certificate once returned from the CA.

## D

---

### DER

A DER format certificate file is a DER-encoded binary certificate. It contains a single certificate and does not support storage of private keys. It sometimes has an extension of .der but is often seen with .cer or .crt.

### Distinguished Name

A distinguished name (DN) is the name that uniquely identifies an object in a directory. In the context of Keyfactor Command, this directory is generally Active Directory. A DN is made up of attribute=value pairs,

separated by commas. Any of the attributes defined in the directory schema can be used to make up a DN.

### DN

A distinguished name (DN) is the name that uniquely identifies an object in a directory. In the context of Keyfactor Command, this directory is generally Active Directory. A DN is made up of attribute=value pairs, separated by commas. Any of the attributes defined in the directory schema can be used to make up a DN.

### DNS

The Domain Name System is a service that translates names into IP addresses.

## E

---

### ECC

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers.

### Endpoint

An endpoint is a URL that enables the API to gain access to resources on a server.

### Enrollment

Certificate enrollment refers to the process by which a user requests a digital certificate. The user must submit the request to a certificate authority (CA).

### EOBO

A user with an enrollment agent certificate can enroll for a certificate on behalf of another user. This is often used when provisioning technology such as smart cards.

## F

---

### Forest

An Active Directory forest (AD forest) is the top most logical container in an Active Directory configuration that contains domains, and objects such as users and computers.

## G

---

### Gateway Connector

The Keyfactor Gateway Connector is installed in the customer forest to provide a connection between the on-premise CA and the Azure-hosted, Keyfactor managed Hosted Configuration Portal to provide support for synchronization, enrollment and management of certificates through the Azure-hosted instance of Keyfactor Command for the on-premise CA. It is supported on both Windows and Linux.

## H

---

### Host Name

The unique identifier that serves as name of a computer. It is sometimes presented as a fully qualified domain name (e.g. server-name.keyexample.com) and sometimes just as a short name (e.g. servername).

### Hosted Config Portal

The Keyfactor Hosted Configuration Portal is used to configure connections between on-premise instances of the Keyfactor

Gateway Connector and on-premise CAs to make them available to Azure-hosted instance of Keyfactor Command. The portal is Azure-hosted and managed by Keyfactor.

### Hosted Configuration Portal

The Keyfactor Hosted Configuration Portal is used to configure connections between on-premise instances of the Keyfactor Gateway Connector and on-premise CAs to make them available to Azure-hosted instance of Keyfactor Command. The portal is Azure-hosted and managed by Keyfactor.

### Hostname

The unique identifier that serves as name of a computer. It is sometimes presented as a fully qualified domain name (e.g. server-name.keyexample.com) and sometimes just as a short name (e.g. servername).

## J

---

### Java Agent

The Java Agent, one of Keyfactor's suite of orchestrators, is used to perform discovery of Java keystores and PEM certificate stores, to inventory discovered stores, and to push certificates out to stores as needed.

### Java Keystore

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

### JKS

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based

applications for authentication and encryption.

## K

---

### Key Length

The key size or key length is the number of bits in a key used by a cryptographic algorithm.

### Key Pair

In asymmetric cryptography, public keys are used together in a key pair with a private key. The private key is retained by the key's creator while the public key is widely distributed to any user or target needing to interact with the holder of the private key.

### Key Size

The key size or key length is the number of bits in a key used by a cryptographic algorithm.

### Key Type

The key type identifies the type of key to create when creating a symmetric or asymmetric key. It references the signing algorithm and often key size (e.g. AES-256, RSA-2048, Ed25519).

### Keyfactor CA Management Gateway

The Keyfactor CA Management Gateway is made up of the Keyfactor Gateway Connector, installed in the customer forest to provide a connection to the local CA, and the Azure-hosted and Keyfactor managed Hosted Configuration Portal. The solution is used to provide a connection between a customer's on-premise CA and an Azure-hosted instance of Keyfactor Command for synchronization, enrollment, and management of certificates.

### Keyfactor Gateway Connector

The Keyfactor Gateway Connector is installed in the customer forest to provide a connection between the on-premise CA and the Azure-hosted, Keyfactor managed Hosted Configuration Portal to provide support for synchronization, enrollment and management of certificates through the Azure-hosted instance of Keyfactor Command for the on-premise CA. It is supported on both Windows and Linux.

### Keyfactor Universal Orchestrator

The Keyfactor Universal Orchestrator, one of Keyfactor's suite of orchestrators, is used to interact with servers and devices for certificate management, run SSL discovery and management tasks, and manage synchronization of certificate authorities in remote forests. With the addition of custom extensions, it can provide certificate management capabilities on a variety of platforms and devices (e.g. Amazon Web Services (AWS) resources, Citrix\NetScaler devices, F5 devices, IIS stores, JKS keystores, PEM stores, and PKCS#12 stores) and execute tasks outside the standard list of certificate management functions. It runs on either Windows or Linux servers or Linux containers.

### Keystore

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

## L

---

### Logical Name

The logical name of a CA is the common name given to the CA at the time it is created. For Microsoft CAs, this name can

be seen at the top of the Certificate Authority MMC snap-in. It is part of the FQDN\Logical Name string that is used to refer to CAs when using command-line tools and in some Keyfactor Command configuration settings (e.g. ca2.keyexample.-com\Corp Issuing CA Two).

## M

---

### MAC Agent

The MAC Agent, one of Keyfactor's suite of orchestrators, is used to manage certificates on any keychains on the Mac on which the Keyfactor MAC Agent is installed.

### Metadata

Metadata provides information about a piece of data. It is used to summarize basic information about data, which can make working with the data easier. In the context of Keyfactor Command, the certificate metadata feature allows you to create custom metadata fields that allow you to tag certificates with tracking information about certificates.

## O

---

### Object Identifier

Object identifiers or OIDs are a standardized system for identifying any object, concept, or "thing" with a globally unambiguous persistent name.

### OID

Object identifiers or OIDs are a standardized system for identifying any object, concept, or "thing" with a globally unambiguous persistent name.

## Orchestrator

Keyfactor orchestrators perform a variety of functions, including managing certificate stores and SSH key stores.

## P

---

### P12

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

### P7B

A PKCS #7 format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certificate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

### P7C

A PKCS #7 format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certificate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

## Parameter

A parameter or argument is a value that is passed into a function in an application.

## PEM

A PEM format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PEM certificates always begin and end with entries like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. PEM certificates can contain a single certificate or a full certificate chain and may contain a private key. Usually, extensions of .cer and .crt are certificate files with no private key, .key is a separate private key file, and .pem is both a certificate and private key.

## PFX

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

## PKCS #7

A PKCS #7 format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certificate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

## PKCS#12

A PFX file (personal information exchange format), also known as a PKCS#12 archive,



is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

## PKI

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

## Private Key

Private keys are used in cryptography (symmetric and asymmetric) to encrypt or sign content. In asymmetric cryptography, they are used together in a key pair with a public key. The private or secret key is retained by the key's creator, making it highly secure.

## Public Key

In asymmetric cryptography, public keys are used together in a key pair with a private key. The private key is retained by the key's creator while the public key is widely distributed to any user or target needing to interact with the holder of the private key.

## Public Key Infrastructure

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

## R

---

### Rogue Key

A rogue key, in the context of Keyfactor Command, is an SSH public key that appears in an `authorized_keys` file on a

server managed by the SSH orchestrator without authorization.

## Root of Trust

A root of trust (RoT) is a source within a cryptographic system that can always be trusted. It is typically a hardened hardware module. HSMs (hardware security modules) and TPMs (trusted platform modules) are examples of RoTs.

## RoT

A root of trust (RoT) is a source within a cryptographic system that can always be trusted. It is typically a hardened hardware module. HSMs (hardware security modules) and TPMs (trusted platform modules) are examples of RoTs.

## RPC

Remote procedure call (RPC) allows one program to call a function from a program located on another computer on a network without specifying network details. In the context of Keyfactor Command, RPC errors often indicate Kerberos authentication or delegation issues.

## rsyslog

Rsyslog is an open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network.

## S

---

## SAN

The subject alternative name (SAN) is an extension to the X.509 specification that allows you to specify additional values when enrolling for a digital certificate. A variety of SAN formats are supported, with DNS name being the most common.

### server name indication

Server name indication (SNI) is an extension to TLS that provides for including the host-name of the target server in the initial handshake request to allow the server to respond with the correct SSL certificate or allow a proxy to forward the request to the appropriate target.

### SMTP

Short for simple mail transfer protocol, SMTP is a protocol for sending email messages between servers.

### SNI

Server name indication (SNI) is an extension to TLS that provides for including the host-name of the target server in the initial handshake request to allow the server to respond with the correct SSL certificate or allow a proxy to forward the request to the appropriate target.

### SSH

The SSH (secure shell) protocol provides for secure connections between computers. It provides several options for authentication, including public key, and protects the communications with strong encryption.

### SSL

TLS (Transport Layer Security) and its predecessor SSL (Secure Sockets Layer) are protocols for establishing authenticated and encrypted links between networked computers.

### Subject Alternative Name

The subject alternative name (SAN) is an extension to the X.509 specification that allows you to specify additional values when enrolling for a digital certificate. A variety of

SAN formats are supported, with DNS name being the most common.

## T

---

### Template

A certificate template defines the policies and rules that a CA uses when a request for a certificate is received.

### TLS

TLS (Transport Layer Security) and its predecessor SSL (Secure Sockets Layer) are protocols for establishing authenticated and encrypted links between networked computers.

### Trusted CA

A certificate authority in the forest in which Keyfactor Command is installed or in a forest in a two-way trust with the forest in which Keyfactor Command is installed.

## U

---

### Untrusted CA

A certificate authority in a forest in a one-way trust with the forest in which Keyfactor Command is installed or in a forest that is untrusted by the forest in which Keyfactor Command is installed. Non-domain-joined standalone CAs also fall into this category.

## W

---

### Web API

A set of functions to allow creation of applications. Keyfactor offers the Keyfactor API, which allows third-party software to integrate with the advanced certificate enrollment and management features of Keyfactor Command.

## Windows Orchestrator

The Windows Orchestrator, one of Keyfactor's suite of orchestrators, is used to manage synchronization of certificate authorities in remote forests, run SSL discovery and management tasks, and interact with Windows servers as well as F5 devices, NetScaler devices, Amazon Web Services (AWS) resources, and FTP capable devices, for certificate management. In addition, the AnyAgent capability of the Windows Orchestrator allows it to be extended to create custom certificate store types and management capabilities regardless of source platform or location.

## Workflow

A workflow is a series of steps necessary to complete a process. In the context of Keyfactor Command, it refers to the workflow builder, which allows you automate event-driven tasks when a certificate is requested or revoked.

## X

---

### x.509

In cryptography, X.509 is a standard defining the format of public key certificates. An X.509 certificate contains a public key and an identity (e.g. a host name or an organization or individual name), and is either signed by a certificate authority or self-signed. When a certificate is signed by a trusted certificate authority it can be used to establish trusted secure communications with the owner of the corresponding private key. It can also be used to verify digitally signed documents and emails.

## 4.0 Copyright Notice

User guides and related documentation from Keyfactor are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. This documentation may not be disclosed, transferred, modified, or reproduced in any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of Keyfactor and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by Keyfactor. Keyfactor assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from Keyfactor.