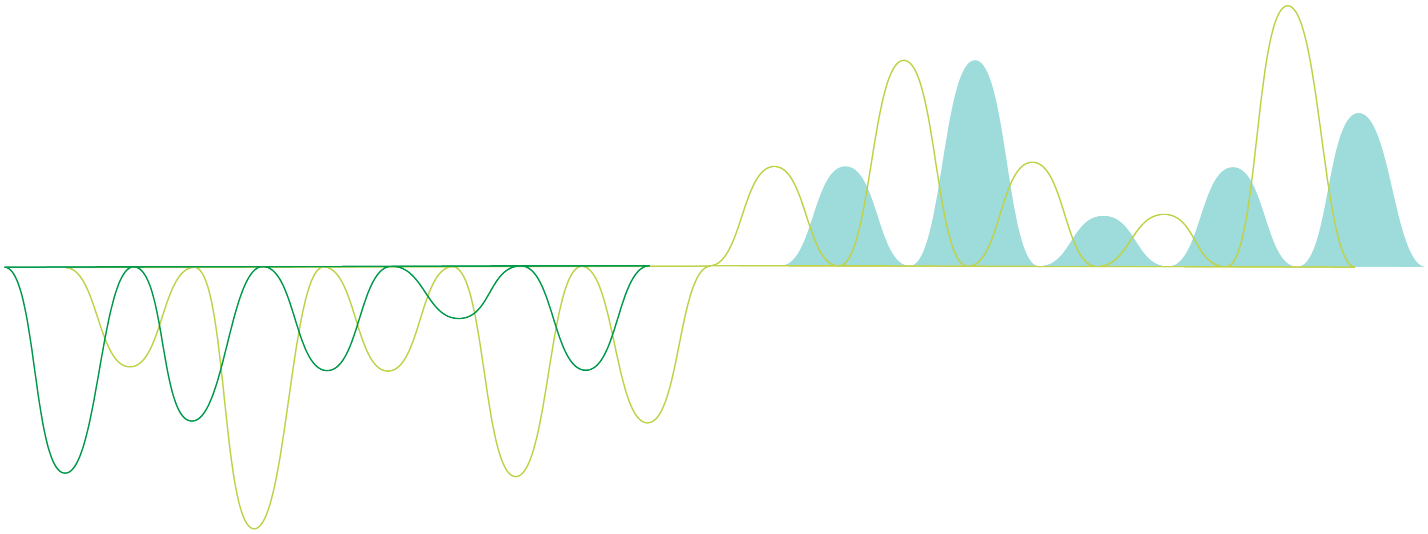
Plan and deploy Qlik Sense

Qlik Sense®
February 2019



HELP.QLIK.COM

LEAD WITH DATA™

1 About this document

This guide will introduce you to planning and installing Qlik Sense.

This document is derived from the online help for Qlik Sense. It is intended for those who want to read parts of the help offline or print pages easily, and does not include any additional information compared with the online help.

You find the online help, additional guides and much more at help.qlik.com/sense.

2 Planning your deployment

To successfully plan and prepare for your Qlik Sense deployment, do the following:

System requirements for Qlik Sense

Check that your environment fulfills the system requirements.

Ports

Check that the required ports are available on your system.

Supported browsers

Check that your browsers are supported.

Architecture

Understand the Qlik Sense architecture, and the different node types.

Persistence

Understand the persistence model used by Qlik Sense.

Services

Understand the Qlik Sense services.

User accounts

Understand and set up the various user accounts required to install and run the Qlik Sense services.

If you intend to run Qlik Sense services as a user without administrator privileges, some additional configuration steps are required.

File share

Create a file share to store your Qlik Sense application data.

Security

Understand how Qlik Sense uses certificates for security. Certificates are installed by default.

Licensing Qlik Sense

Understand how Qlik Sense uses license keys and LEF for site licensing.

Understand how Qlik Sense uses tokens for user access allocation (token-based licensing).

Ensure that you have your Qlik Sense license key available.

Qlik Sense installation

Once you have reviewed and completed these items, you are ready to install Qlik Sense.

2.1 System requirements for Qlik Sense

This section lists the requirements that must be fulfilled by the target system in order to successfully install and run Qlik Sense.

Platforms	• Microsoft Windows Server 2012 • Microsoft Windows Server 2012 R2 • Microsoft Windows Server 2016 For development and testing purposes only: • Microsoft Windows 7 (64-bit version only) • Microsoft Windows 8.1 (64-bit version only) • Microsoft Windows 10 (64-bit version only) *These operating systems are supported by Qlik Sense. Third-party software may require service packs to be installed.*
Processors (CPUs)	Multi-core x64 compatible processors We recommend that you use at least 4 cores per node in a Qlik Analytics Platform deployment.
Memory	8 GB minimum (depending on data volumes, more may be required) Qlik Sense is an in-memory analysis technology. The memory requirements for the Qlik Sense products are directly related to the amount of data being analyzed.
Disk space	5.0 GB total required to install
Storage	• A network file share is required for the storage to be accessible by all servers in the site. In case of a single-server deployment, local disk storage may be sufficient. • Sufficient storage is required for the volume of apps and content used in the deployment.
Security	• Microsoft Active Directory • Microsoft Windows Integrated Authentication • Third-party security
WebSockets	Web browsers and infrastructure components (such as proxies and routers) must support WebSockets.

.NET framework	4.5.2 or higher
PowerShell	4.0 or higher
Repository database	PostgreSQL 9.6.x PostgreSQL is included in the Qlik Sense setup by default. However, you can also download and install it manually. *The version of PostgreSQL 9.6.x installed with Qlik Sense does not include pgAdmin tools. You can download and install them manually if required.* PostgreSQL is an open source object-relational database management system. It is released under the PostgreSQL license, which is a free and open source software license.
Centralized logging database	PostgreSQL 9.6.x
Internet protocol	• IPv4 • IPv6 • Dual stack (IPv4 and IPv6)
Network	The configured hostname must resolve to an IP address on the host machine.

Qlik Management Console (QMC), supported browsers	Microsoft Windows 7, Windows 8.1: - Microsoft Internet Explorer 11 - Google Chrome - Mozilla Firefox (requires hardware acceleration, not supported in virtual environments) Microsoft Windows Server 2012, Windows Server 2012 R2, Windows Server 2016: - Microsoft Internet Explorer 11 (not supported on Windows Server 2012) - Google Chrome - Mozilla Firefox (requires hardware acceleration, not supported in virtual environments) Microsoft Windows 10: - Microsoft Internet Explorer 11 - Microsoft Edge - Google Chrome - Mozilla Firefox (requires hardware acceleration, not supported in virtual environments) Apple Mac OS X 10.11 and 10.12: - Apple Safari - Google Chrome - Mozilla Firefox (requires hardware acceleration, not supported in virtual environments) CefSharp embedded browser v55 or later (CefSharp allows you to embed the Chromium open source browser inside .Net apps)
QMC, minimum screen resolution	Desktops, laptops, and Apple Mac: 1024x768 No mobile or small screen support.
Qlik Cloud Services	Maximum app size: 500 MB Storage: 500 GB *When distributing to Qlik Cloud Services, your Qlik Sense Enterprise for Windows deployment must but either the current version or one of the previous 2 releases (starting from the June 2018 release).*

Qlik Sense Enterprise for elastic deployments	Kubernetes environments: The Kubernetes environment must have Internet access to the Qlik Helm and Container Image repository. Kubernetes service vendors: • Microsoft Azure using Azure Kubernetes Service (AKS) • Amazon Web Services (AWS) using Amazon Elastic Container Service for Kubernetes (EKS) • Google Cloud using Google Kubernetes Engine (GKE) Non-managed Kubernetes deployments: • Kubernetes 1.9.4 and 1.10.3 • Kubernetes deployed to Amazon Web Services (AWS) using Kubernetes Operations (KOPs) Local/Evaluation/Test environment: • Windows: Minikube v1.9.4 • Docker for Mac/Windows with Kubernetes enabled: v1.9.4
	Kubernetes package manager: • Helm 2.9.0
	Database: • MongoDB 3.6+
	File system: • NFS 4.1 compliant NFS with ReadWriteMany access *When distributing to Qlik Sense Enterprise for elastic deployments, your Qlik Sense Enterprise for Windows and Qlik Sense Enterprise for elastic deployments versions should be the same release.*
QlikView compatibility	It is not possible to install Qlik Sense on a machine with QlikView Server already installed.

Qlik Sense Mobile app for iOS	iOS supported versions: - iOS 11.2 or later ⓘ *iOS 11.0 or later is required for SAML authentication.* Device compatibility: - iPad Air 2 or later - iPad Pro or later - iPhone 6 and 6 Plus - iPhone 6S and 6S Plus - iPhone 7 and 7 Plus - iPhone 8 and 8 Plus - iPhone X Qlik Sense Mobile for iOS compatibility with Qlik Sense: - Qlik Sense September 2017 and later releases - Qlik Sense November 2018 or later is required to access mashups from the Qlik Sense Mobile iOS app. ⚠ *Qlik SenseFebruary 2018 or later is required to reduce the size of apps for download to your iOS device.*
Qlik Sense Mobile app for Android	Android OS supported versions: - Android 6.0 or later Device compatibility: - 64-bit CPU architecture (ARM) - RAM: 2 GB or more is recommended - Screen resolution: 720x1280 HDPI (267 ppi) or better Qlik Sense Mobile for Android compatibility with Qlik Sense: - Qlik Sense November 2018 and later releases

Qlik Sense Mobile per-app VPN support	Qlik Sense Mobile per-app VPN through Enterprise Mobile Management (EMM) is designed to work with the hardware, iOS versions and Qlik Sense versions listed in this section. • Qlik Sense Enterprise November 2017 or later. • iPad with iOS version 11.2.2 or later. • iPhone with iOS version 11.2.2 or later. Access to Qlik Sense Enterprise using AirWatch per-app VPN is supported on the following browsers: • VMware browser • Apple Safari • Google Chrome

We do not recommend that you install Qlik Sense on domain controller machines, as group policies may prevent Qlik Sense from getting access to required services.

2.2 Supported browsers

Qlik Sense is designed to work on the platform and web browser combinations described in this section, using default browser settings.

Qlik Sense Cloud is designed to work on web browsers listed in this section.

Each Qlik Sense release is tested for compatibility with the latest publicly available browser versions. Due to the frequency of browser version updates, Qlik does not include specific browser version numbers in the system requirements.

Each Qlik Sense release is compatible with and supported on the latest iOS versions that are publicly available at the time of the Qlik Sense release. Due to the frequency of iOS version updates, Qlik does not include specific iOS version numbers in the system requirements.

Improving performance in Internet Explorer

Qlik Sense connects to your browser using WebSockets. Each new tab that you open uses additional WebSocket connections. By default, Internet Explorer 11 limits the number of Websocket connections to 6 per Internet Explorer session. This can limit your ability to open new tabs or configuration windows.

Your Windows administrator can change this setting using the Local Group Policy Editor. The setting is available under Administrative Templates > Windows Components > Internet Explorer > Security Features > AJAX > Set the maximum number of WebSocket connections per server. Only your system administrator should change this configuration.

You can also open apps in new Internet Explorer sessions, instead of new tabs. If an app will not open in a new tab, copy the url from the address bar in the Internet Explorer tab. Select File > New Session from the Internet Explorer top menu. Paste the url in the address bar, and then press Enter. The app opens in the Internet Explorer new session window.

Qlik Management Console (QMC)

Microsoft Windows 7, 8.1

- Microsoft Internet Explorer 11
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Microsoft Windows Server 2012, 2012 R2, 2016

- Microsoft Internet Explorer 11 (not supported on Windows Server 2012)
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Microsoft Windows 10

- Microsoft Internet Explorer 11
- Microsoft Edge
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Apple Mac OS X 10.11 and 10.12

- Apple Safari 10 or later
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

CefSharp embedded browser v55 or later (CefSharp allows you to embed the Chromium open source browser inside .Net apps)

Minimum screen resolution for desktops, laptops, and Apple Mac is 1024x768. The QMC does not support tablets or iPads.

Qlik Sense (the hub)

Microsoft Windows 7

- Microsoft Internet Explorer 11
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Microsoft Windows 8.1

- Microsoft Internet Explorer 11
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Microsoft Windows 10

- Microsoft Edge
- Microsoft Internet Explorer 11
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Apple Mac OS X 10.11 and 10.12

- Apple Safari 10 or later
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments).

Microsoft Windows Server 2012

- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Microsoft Windows Server 2012 R2

- Microsoft Internet Explorer 11
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

Microsoft Windows Server 2016

- Microsoft Internet Explorer 11
- Google Chrome
- Mozilla Firefox (requires hardware acceleration, not supported in virtual environments)

CefSharp embedded browser v55 or later (CefSharp allows you to embed the Chromium open source browser inside .Net apps)

iOS

Version 11.2 or later (script editing is not supported on tablet devices).

Qlik Sense version: Qlik Sense Enterprise September 2017 or later.

Supported devices:

- iPad Air or later
- iPhone 5S or later

Supported browsers:

- Apple Safari
- Google Chrome
- VMware browser (using AirWatch per-app VPN)
- BlackBerry Access 2.9.1 or later (using BlackBerry Dynamics platform)

iOS 11.3 is required for using BlackBerry Access browser.

Android

Version 6.0, 7.1, 8.1 and 9.0 (script editing is not supported on tablet devices):

- Google Chrome
- BlackBerry Access 2.9.1 or later (using BlackBerry Dynamics platform)

Windows 10 phone

- Microsoft Edge

Minimum screen resolution for desktops and laptops is 1024x768; tablets is 1024x768; small screens is 320x568.

2.3 Architecture

The Qlik Sense architecture consists of one or more nodes. Each node runs some or all of the software services that perform specific roles in a Qlik Sense site. You can distribute services across nodes for better performance and scalability. The architecture is flexible enough to suit the needs of most organizations, and can vary from small, single-server sites to large, multi-server installations.

A multi-node, distributed architecture offers the most flexibility, consisting of multiple nodes that together form a scalable and high performance site. You define a central node as the main point of control.

Sites

A Qlik Sense site is a collection of one or more nodes (servers) connected to a single repository database, and sharing a single license. Each site also contains a common set of data in the form of apps and configuration data.

Single-node sites

A single node site is the smallest site possible and consists of a single node (single server), which is also the central node of the site. It contains the Qlik Sense services, the repository database, and the file share all on a one server computer.

Multi-node sites

Multi-node sites offer more scalability options for larger organizations. In a multi-node environment, the Qlik Sense site is distributed across two or more nodes that share the same set of data and the same license key. In larger sites, you can configure one or more rim nodes to improve scalability, capacity, and resilience. All rim

nodes connect to a central node.
Benefits of multi-node sites include:

- Better scalability, making it easier to increase capacity
- Improved resilience and reliability
- Ability to move apps or roles to specific nodes
- Flexibility to suit customer network deployments

Nodes

A node is a computer that performs a specific role in your Qlik Sense site. You can configure each node to run or combine a different set of Qlik Sense services, so that each node performs a specific role.

Typical node roles:

- Consumer or user node - delivers apps to end users
- Scheduler node - handles all app reloads
- Proxy node - manages authentication, session handling, and load balancing

You can also configure your site for failover so that it is not dependent on the central node. In this case, if there is a failure, then one of the rim nodes in the site becomes the central node. For more information on how to configure fail over, see Creating a node and Service cluster.

A typical multi-server Qlik Sense site consists of two main types of nodes:

- Central node - the minimum configuration. Every site includes a central node.
- Rim node - you can configure rim nodes to perform different roles in your site.

Each node in a Qlik Sense site can:

- Perform different roles
- Deploy a set of Qlik Sense services
- Operate independently

You assign a purpose to each node depending on what you think it will be used for:

- Production
- Development
- Both

For more information on node purpose, see: Creating a node.

Configuring Qlik Sense nodes correctly increases system resilience, reduces the need for maintenance, and increases deployment flexibility.

Storage

Qlik Sense uses the following default storage.

Repository database

A PostgreSQL database that contains the Qlik Sense app metadata, including the paths to the binary files in the file share. This data is referred to as entity data and is usually small in size. The PostgreSQL database can be installed locally or on a remote server and must be accessible to the central node.

File share

A file share is used to store app data as binary files and must be accessible to all nodes in your Qlik Sense site. The file share stores application objects, such as visualizations, and dimensions and measures. Apps are stored in the proprietary QVF portable format, for example `<App name>.qvf`. These files are referred to as binary data and the data model element of the files can be large in size.

You can create a file share either on the same server as the central node or on another server.

See: *Creating a file share (page 97)*

Clients

You use Qlik Sense clients to communicate and interact with Qlik Sense sites.

Hub

The hub is where you find all the apps you have access rights to. It runs in a web browser. You use the hub to access and publish apps in Qlik Sense. Hub traffic only travels between the node (delivering apps) and the hub client unless the site is on a single node.

Qlik Management Console

You use the Qlik Management Console (QMC) to configure and administer a Qlik Sense site.

The QMC only communicates logically with the central node. This means that:

- The QMC always uses the Qlik Sense Proxy Service (QPS) on the central node.
- For maximum performance within a multi-node site, you should not allow any user traffic on the central node.

Apps

A Qlik Sense app is a collection of reusable data items (measures, dimensions, and visualizations), sheets, and stories. It is a self-contained entity that includes the data you want to analyze in a structured data model.

> *In Qlik Sense, the term app is equivalent to the term document in QlikView.*

Services

The Qlik Sense services run as Microsoft Windows services, which you can deploy on a single server or on separate server nodes that have dedicated roles in a Qlik Sense site. For example, you could deploy a scheduler node that only runs the scheduler service and manages the reloads of apps.

The Qlik Sense services are as follows.

Qlik Sense Repository Service (QRS)

Required by all Qlik Sense services to run and serve apps, and connects to the repository database. The Qlik Sense Repository Service manages persistence, licensing, security, and service configuration data. The QRS is needed by all other Qlik Sense services to run and serve apps. In a multi-node site, one instance of the Qlik Sense Repository Service (QRS) runs on each node, connecting it to the shared repository database.

In addition, the QRS stores the app structures and the paths to the binary files. The app data is stored as `.qvf` files in the file share.

Paths

The following table lists the paths used by the Qlik Sense Repository Service (QRS).

Executable	*%ProgramFiles%\Qlik\Sense\Repository\Repository.exe*
Data	*%ProgramData%\Qlik\Sense\Repository*
Logs	*%ProgramData%\Qlik\Sense\Log\Repository* See: *Logging (page 236)*
Repository database	In a default Qlik Sense installation, the repository database is an instance of PostgreSQL installed locally that runs its own database cluster specifically for the repository. All files related to the repository database in a default Qlik Sense installation are stored in the following folder: *%ProgramData%\Qlik\Sense\Repository\PostgreSQL*

Bootstrap mode

You can use the following parameters to start the Qlik Sense Repository Service in bootstrap mode when you need to deploy Qlik Sense with a service account that does not have administrator privileges.

See: *Changing the user account to run Qlik Sense services (page 103)*

- `-bootstrap`
  Use this parameter to start Qlik Sense Repository Service in bootstrap mode.
- `-bootstrap=install`
  Use this parameter to start Qlik Sense Repository Service in bootstrap mode when installing.
- `-bootstrap=uninstall`
  Use this parameter when uninstalling Qlik Sense.
- `-iscentral`
  Use this flag in addition to the bootstrap flag when installing or configuring a central node.

Do the following:

1. Stop all Qlik Sense services except Qlik Sense Repository Database.
2. Run `repository.exe -bootstrap` from an elevated command prompt.
3. Start Qlik Sense services.

By default, when you are running Qlik Sense with an administrator account, bootstrap is executed each time the Qlik Sense services are restarted. To disable automatic bootstrap in the Qlik Sense repository, you must update the configuration file. By default, the Repository.exe.config file can be found in C:\Program Files\Qlik\Sense\Repository\ on your Qlik Sense machine. Edit the configuration file and change the value of the `DisableAutomaticBootstrap` key to `true`. Restart the Qlik Sense Repository Service using the Windows Services application to enable this new configuration.

Metrics

This section lists the metrics related to the Qlik Sense Repository Service (QRS).

Selecting the metrics to display (page 31)

REST API metrics

The following metrics are available in the Performance Monitor in Microsoft Windows:

- Number of DELETE calls
- Number of GET calls
- Number of POST calls
- Number of PUT calls
- Number of HTTP status 200 (OK)
- Number of HTTP status 201 (Created)
- Number of HTTP status 400 (Bad request)
- Number of HTTP status 401 (Unauthorized)
- Number of HTTP status 403 (Forbidden)
- Number of HTTP status 406 (Not acceptable)
- Number of HTTP status 409 (Conflict)
- Number of HTTP status 415 (Unsupported media type)
- Number of HTTP status 500 (Internal server error)
- Number of HTTP status 503 (Service unavailable)

Qlik Sense Repository Database (QRD)

In a default Qlik Sense installation, the Qlik Sense Repository Service (QRS) uses the Qlik Sense Repository Database (QRD) service to read and write data in the repository database. By default a PostgreSQL database is installed locally with your Qlik Sense installation otherwise you can choose to install PostgreSQL on a separate dedicated server.

Paths

The following table lists the paths used by the Qlik Sense Repository Database (QRD) service.

Executable	In a default Qlik Sense installation, the repository database is an instance of PostgreSQL that creates its own database cluster. The following folder contains the contains the PostgreSQL executable file for the QRD: *%ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\&lt;database version&gt;\bin*
Data	*%ProgramData%\Qlik\Sense\Repository\PostgreSQL*
Logs	There are no logs for the QRD service. Instead see the PostgreSQL log files.

Qlik Sense Proxy Service (QPS)

The Qlik Sense Proxy Service (QPS manages site authentication, session handling, and load balancing.

On the central node in a multi-node site, you should have a dedicated Qlik Sense Proxy Service (QPS) for the Qlik Management Console (QMC) and not for the hub.

Paths

The following table lists the paths used by the Qlik Sense Proxy Service (QPS).

Executable	*%ProgramFiles%\Qlik\Sense\Proxy\Proxy.exe*
Data	*%ProgramData%\Qlik\Sense\Proxy*
Logs	*%ProgramData%\Qlik\Sense\Log\Proxy* See: *Logging (page 236)*

Bootstrap mode

You can use the following parameters to start the Qlik Sense Proxy Service in bootstrap mode when you need to deploy Qlik Sense with a service account that does not have administrator privileges.

See: *Changing the user account to run Qlik Sense services (page 103)*

- `-bootstrap`
  Use this parameter to start Qlik Sense Proxy Service in bootstrap mode.
- `-bootstrap=install`
  Use this parameter to start Qlik Sense Proxy Service in bootstrap mode when installing.
- `-bootstrap=uninstall`
  Use this parameter when uninstalling Qlik Sense.

Do the following:

1. Stop Qlik Sense services.
2. Run `proxy.exe -bootstrap` from an elevated command prompt.
3. Start Qlik Sense services.

Metrics

This section lists the metrics related to the Qlik Sense Proxy Service (QPS). The following metrics are available in the Performance Monitor in Microsoft Windows:

See: *Performance log (page 274)*

See: *Selecting the metrics to display (page 31)*

- ActiveConnections: The number of active connections from the client.
  A connection is a stream (or a socket) between a Qlik Sense client and the Qlik Sense Proxy Service (QPS). This stream is often connected to another stream, which runs from the QPS to the Qlik Sense Repository Service (QRS) or the Qlik Sense Engine Service (QES). The two streams allow the client to communicate with the QRS or the QES.
- ActiveStreams: The number of active data streams (or sockets), either from the browser to the QPS or from the QPS to the QRS or the QES.
- ActiveSessions: The number of active sessions in the QPS.
  A Qlik Sense user gets a proxy session when the user has been authenticated. The session is terminated after a certain period of inactivity.
- LoadBalancingDecisions: The number of users who currently have at least one engine session.
- PrintingLoadBalancingDecisions: The number of users who have been load balanced to the Qlik Sense Printing Service (QPR).
- Tickets: The number of issued login tickets that have not yet been consumed.
- ActiveClientWebsockets: The number of active WebSockets between the client and the QPS.
- ActiveEngineWebsockets: The number of active WebSockets between the QPS and the target Qlik Sense service.

The metrics are also available as entries in the Performance log for the QPS.

Qlik Sense Scheduler Service (QSS)

The Qlik Sense Scheduler Service (QSS) manages the scheduled reloads of apps, as well as other types of reload triggering based on task events. Depending on the type of deployment, the Qlik Sense Scheduler Service runs as master, slave, or both on a node.

Master

There is only one master Qlik Sense Scheduler Service within a site and it is always located on the central node, where the master Qlik Sense Repository Service runs. The central node must have the Qlik Sense Scheduler Service installed even if more QSS nodes are added because the QSS on the central node coordinates all QSS activities within the site.

The master QSS handles all task administration. For example, which tasks to execute and when to execute a specific task. When the time comes to execute a task, the master QSS sends the task ID to a slave QSS within the site. The load balancing operation performed by the master QSS determines which slave QSS to distribute the task ID to.

When a slave QSS completes a task, it returns the task state (successful or fail) to the master QSS. The master QSS uses the task state to perform task chaining. It uses the task state to determine if other events are affected by the state of the completed task and need to be executed. You configure task chaining in the Qlik Management Console (QMC).

If the slave QSS fails to perform the task, the master QSS repeatedly requests the same or another slave QSS to perform the task until it has been completed or until the maximum number of attempts has been reached.

Slave

If a Qlik Sense Scheduler Service (QSS) runs on a rim node, the QSS is considered to be a slave QSS. When receiving a task ID from the master QSS, the slave QSS reads the task from the local repository database and executes the task. When a slave QSS completes a task, it returns the task state (successful or fail) to the master QSS.

Tasks

Tasks are used to perform a wide variety of operations and can be chained together in any arbitrary pattern. The tasks are handled by the Qlik Sense Scheduler Service (QSS) and managed in the Qlik Management Console (QMC).

Reload

The reload task is used to fully reload the data in an app from the source. Any old data is discarded.

Paths

The following table lists the paths used by the Qlik Sense Scheduler Service (QSS).

Executable	*%ProgramFiles%\Qlik\Sense\Scheduler\Scheduler.exe*
Data	-
Logs	*%ProgramData%\Qlik\Sense\Log\Scheduler* See: *Logging (page 236)*

Bootstrap mode

You can use the following parameters to start the Qlik Sense Scheduler Service in bootstrap mode when you need to deploy Qlik Sense with a service account that does not have administrator privileges.

See: *Changing the user account to run Qlik Sense services (page 103)*

- `-bootstrap`
  Use this parameter to start Qlik Sense Scheduler Service in bootstrap mode.
- `-bootstrap=install`
  Use this parameter to start Qlik Sense Scheduler Service in bootstrap mode when installing.
- `-bootstrap=uninstall`
  Use this parameter when uninstalling Qlik Sense.

Do the following:

1. Stop Qlik Sense services.
2. Run `scheduler.exe -bootstrap` from an elevated command prompt.
3. Start Qlik Sense services.

Metrics

This section lists the metrics related to the Qlik Sense Scheduler Service (QSS). The following metrics are available in the Performance Monitor in Microsoft Windows:

See: *Selecting the metrics to display (page 31)*

- Number of connected slaves
- Number of Qlik Sense Engine Service (QES) instances that are running on a slave (this metric is only available on the node where the QES instances run)
- Number of running processes
- Number of running tasks as understood by the master
- Number of running tasks on the slave
- Number of task messages that have been dispatched by the slave
- Number of task messages that have been received by the master
- Number of task retries
- Number of tasks that have completed successfully when executed by the slave
- Number of tasks that have failed when executed by the slave
- Number of tasks that the master has acknowledged as completed
- Number of tasks that the master has acknowledged as failed
- Number of times that the settings have been updated
- Number of tasks that have attempted to start
- Number of tasks that have attempted to stop

Qlik Sense Engine Service (QES)

The Qlik Sense Engine Service (QES) handles all application calculations and logic. In a multi-node site, we recommend that you have a dedicated Qlik Sense Engine Service (QES) on the central node that you use specifically for the Qlik Management Console (QMC) and not for the hub.

Paths

The following table lists the paths used by the Qlik Sense Engine Service (QES).

Executable	*%ProgramFiles%\Qlik\Sense\Engine\Engine.exe*
Data	*%ProgramData%\Qlik\Sense\Engine*
Logs	*%ProgramData%\Qlik\Sense\Log\Engine* See: *Logging (page 236)*
Configuration	*%ProgramData%\Qlik\Sense\Engine\Settings.ini* This file contains the QES settings. The file is created when the service first runs.

Qlik Logging Service

The Qlik Sense services (proxy, scheduler, repository, and engine) transfer log messages to the Qlik Logging Service. The Qlik Logging Service centralizes the logging by collecting all the messages and inserting them into the PostgreSQL database.

Qlik Sense Printing Service (QPR)

This service manages export in Qlik Sense. In a multi-node site, one instance of the Qlik Sense Printing Service (QPR) runs on each node. Export requests from clients are directed to the printing services in the multi-node site using round robin load balancing. If the first export request is load balanced to the QPR on node 1, the second export request is load balanced to the QPR on node 2, and so on.

Paths

The following table lists the paths used by the Qlik Sense Printing Service (QPR).

Executable	*%ProgramFiles%\Qlik\Sense\Printing\Printing.exe*
Data	*%ProgramData%\Qlik\Sense\Printing*
Logs	*%ProgramData%\Qlik\Sense\Log\Printing* See: *Logging (page 236)*

Qlik Sense Service Dispatcher (QSD)

This is a service controller used to launch and manage the following Qlik Sense services:

- Broker Service: acts as an interface to and an intermediary between services started by the Qlik Sense Service Dispatcher(QSD). The service is launched and managed by the Qlik Sense Service Dispatcher (QSD) when required.
- Data Profiling Service: is used to access and modify the app load data model. It communicates directly with the Qlik Sense Engine Service (QES) on the node. The service is launched and managed by the Qlik Sense Service Dispatcher (QSD) when required.
- Hub Service: controls which content a user is allowed to see based on their access rights as defined in the QMC. The service is launched and managed by the Qlik Sense Service Dispatcher(QSD) when required.
- Migration Service: ensures that your apps can be used in the currently installed version of Qlik Sense. This service only runs on the central node in a site. The service is launched and managed by the Qlik Sense Service Dispatcher (QSD) when required.
- Web Extension Service: is used to control web extensions such as visualizations, mashups, and widgets. The service is launched and managed by the Qlik Sense Service Dispatcher (QSD) when required.
- Capability Service: is used to handle Qlik Sense .NET SDK system feature configuration.
- Converter Service: is used by the QlikView converter tool.
- On-demand App Service: generates on-demand apps that load subsets of data from very large data sets.
- Hybrid Deployment Service (HDS): manages target deployments and credentials related to hybrid connectivity between environments, specifically the distribution of apps from the QSE.

- Hybrid Setup Console (HSC): serves the HSC user interface which is used to configure target deployments and app distribution.
- App Distribution Service (ADS): distributes apps and associated metadata to defined distribution targets, based on policy based app distribution rules.
- Entitlement Provisioning Service (EPS): allocates access to users in target environments by listening for user access type allocations from the Qlik Sense Repository Service.
- Precedents Service: examines and captures precedents in the data models and field use in charts from apps for use in insight advisor. The service also captures user-learned feedback from insight advisor.

Paths

The following table lists the paths used by the Qlik Sense Service Dispatcher (QSD) and the services that are launched and managed by the QSD.

Executables	• QSD: *%ProgramFiles%\Qlik\Sense\ServiceDispatcher\ServiceDispatcher.exe* • Services that are launched and managed by the QSD: *%ProgramFiles%\Qlik\Sense\ServiceDispatcher\node\node.exe*
Logs	• Broker Service: *%ProgramData%\Qlik\Sense\Log\BrokerService* • Data Profiling Service: *%ProgramData%\Qlik\Sense\Log\DataProfiling* • Hub Service: *%ProgramData%\Qlik\Sense\Log\HubService* • Migration Service: *%ProgramData%\Qlik\Sense\Log\AppMigration* • Web Extension Service: *%ProgramData%\Qlik\Sense\Log\WebExtensionService* • On-demand App Service: *%ProgramData%\Qlik\Sense\Log\OdagService* • Capability Service: *%ProgramData%\Qlik\Sense\Log\CapabilityService* • Hybrid Deployment Service: *%ProgramData%\Qlik\Sense\Log\HybridDeploymentService* • Hybrid Setup Console: *%ProgramData%\Qlik\Sense\Log\HybridSetupConsole* • App Distribution Service: *%ProgramData%\Qlik\Sense\Log\AppDistributionService* • Precedents Service: *%ProgramData%\Qlik\Sense\Log\PrecedentsService* See: *Logging (page 236)*

Deployment examples of nodes running Qlik Sense services

You can deploy Qlik Sense services to run individually or combine them on dedicated server nodes.

- Complete: A single-node deployment that includes all Qlik Sense services.
- Consumer node: A node that delivers Qlik Sense apps to end users. It includes the Qlik Sense Engine Service service, the Qlik Sense Proxy Service, and the Qlik Repository service.
- Proxy node: A node that manages Qlik Sense authentication, session handling, and load balancing. It includes the QRS, and the QPS services.
- Engine node: A node that provides the analytical power of Qlik Sense to the client. It includes the QRS, and the QES services.

- Proxy and engine node: A combined node that includes the QRS, QPS, and QES service.
- Scheduler: A node that manages scheduled reloads of Qlik Sense apps and other types of reload triggering. It includes the QRS, QSS, and QES services. In order to perform reloads the QSS requires the QES to be running on the same node.

Service dependencies

This section describes the dependencies related to the Qlik Sense services (for example, dependencies on the operating system and other software).

Repository database

The Qlik Sense Repository Service (QRS) connects to the repository database to store and retrieve data necessary for the Qlik Sense services on the node on which the QRS is running. In a default Qlik Sense installation, the Qlik Sense Repository Service (QRS) uses the Qlik Sense Repository Database (QRD) service to read and write data in the repository database. A PostgreSQL database is used by default.

File share

The file share stores the binary files for the Qlik Sense apps.

Directory service

The QRS and Qlik Sense Proxy Service (QPS) communicate with a configured directory service (for example, Microsoft Active Directory) using, for example, LDAP or ODBC.

Start and restart of services

When a node starts up, the Qlik Sense services are started automatically.

Start-up behavior

The Qlik Sense Repository Database (QRD) and Qlik Sense Repository Service (QRS) are started first.

When any other Qlik Sense service starts, it contacts its local QRS to get configuration parameters. If the service has not been configured to run, it periodically checks back with the local QRS.

Manual start

If you need to start services manually, start them in the following order:

a. Qlik Sense Repository Database (QRD)
b. Qlik Sense logging service
c. Qlik Sense Service Dispatcher (QSD)
d. Qlik Sense Repository Service (QRS)
e. Qlik Sense Proxy Service (QPS), Qlik Sense Engine Service (QES), Qlik Sense Scheduler Service (QSS), and Qlik Sense Printing Service (QPR) in no specific order

The order is important because the QRS is dependent on the QRD and the rest of the services are dependent on the QRS.

Selecting the metrics to display

To select which metrics to display for the Qlik Sense services in the Microsoft Windows, Performance Monitor:

1. Select **Start>Run**.
2. Enter *perfmon* and click **OK**.
3. In the left panel, expand **Monitoring Tools** .
4. Select **Performance Monitor**.
   The **Performance Monitor** is displayed in the right panel.
5. Click the + (plus) icon in the toolbar at the top of the **Performance Monitor**.
   The **Add Counters** dialog is displayed.
6. Select the computer to add counters from in the **Select counters from computer**: drop-down list.
   The **Available counters** list is populated with counters.
7. In the **Available counters** list, locate the following counter sets :
   - Qlik Sense Proxy Service
   - Qlik Sense Repository Service - REST API
   - Qlik Sense Repository Service
   - Qlik Sense Scheduler Service
8. Click the + (plus) sign next to a counter set to expand the set.
9. In the **Performance Monitor**, select the counters to display .
10. Click **Add >>** to add the counters.
11. The added counters are listed in the **Added counters** list.
12. Click **OK**.

The counters you added are now displayed in the **Performance Monitor**.

Multi-cloud services

You have several options when deploying a Qlik Sense Enterprise for Windows environment. For an overview of the Qlik Sense multi-cloud architecture and your different deployment options, see *Qlik Sense deployments in a multi-cloud environment (page 111)*. The services that you need to run in a multi-cloud deployment can be categorized as follows.

Typically the services running in a QCS deployment are similar to those running in a Qlik Sense Enterprise elastic deployment but are not accessible, because Qlik manages the infrastructure. You can connect to QCS SaaS but do not have the same configuration options as an elastic deployment.

Services on Windows deployments

The services listed below are required if you use the multi-cloud capabilities in a Qlik Sense Enterprise for Windows deployment.

Service	Description
App Distribution Service	Distributes apps and associated metadata to defined distribution targets, based on policy-based app distribution rules.
Entitlement Provisioning	Distributes license allocations for users to all target environments configured in a multi-cloud deployment.

Service	
Hybrid Deployment Service	Stores configuration details including credentials and URLs for all target environments in a multi-cloud deployment.
Hybrid Setup Console Service	Multi-cloud Setup Console UI functions for managing target environments configured in a multi-cloud deployment including credentials and service URLs.
Resource Distribution Service	Publishes installed extensions and themes to the Resource Library in each cloud environment.

Services on elastic deployments

The services that you run in Qlik Sense Enterprise (QSE) for elastic deployments can vary depending on your deployment requirements.

Service	Description
Collections	Organizes and structures content supplied to the hub. It also applies access control rules.
Cloud hub	Serves the hub functionality to users in Qlik Cloud Services and Qlik Sense Enterprise for elastic deployments.
Policy Decision	Processes a set of rules on Qlik Cloud Services and QSE for elastic deployments that perform ABAC security evaluation against Qlik objects (for example, apps). It is sometimes referred to as the Rules Service. It uses a REST API for the rules engine and management API for rule based policies and replaces the QRS Rules Engine, Policy Decision Service.
Sense Client	The Desktop and web browser instance of the Qlik Sense client run by developers on Qlik Sense Enterprise and by consumers on QCS and QSE for elastic deployments.
Engine	Handles all application calculations and logic.
edge-auth	Service that works together with external Identity Providers to authenticate users upon entry to the deployment. Also manages tickets that authorize secure access to internal resources.
elastic-infra	A collection containing non-Qlik services: MongoDB, Redis, Traefik, and ngingx-ingress. It bootstraps an elastic-infra deployment on a Kubernetes cluster using the Helm package manager. It starts up the basic resources needed to connect all the components and functionality required in a cloud environment.
qix-sessions	Responsible for routing user session traffic to the Engine services.
Mira	Provides a discovery service for Engines in the deployment, their current health, and availability of applications.
Feature Flags	Responsible for toggling features on and off in advanced scenarios.
Licenses	The license service is used to enforce user licensing in Qlik Cloud Services and Qlik Sense Enterprise for elastic deployments.

Locale	Handles user locale selection for the client.
Resource Library	A general-purpose resource storage service for supporting content such as themes and extensions.
User	Responsible for managing and retrieving user information.
Tenant	Used to store and return tenant (user) information.

Ports

Qlik Sense Enterprise and Qlik Sense Enterprise for Windows use ports to communicate between web browsers (users) and proxies, and between services in single and multi-node deployments.

> *Qlik Sense Enterprise for elastic deployments runs on a Kubernetes cluster which has no specific port requirements that are different to any other application that is hosted on Kubernetes. For more general information on Kubernetes ports requirements, see the* [Kubernetes cluster documentation]().

Ports overview

The following table is an overview of the ports used in a Qlik Sense deployment.

Component	Inbound	Outbound	Internal only
Qlik Sense Proxy Service (QPS)	80 (HTTP) 443 (HTTPS) 4243 (REST API)	4239 (QRS websocket) 4242 (QRS REST API) 4747 (Engine) 4899 (Printing) 4900 (Broker) 4949 (Data profiling) 7070 (Logging service)	4244 (Windows authentication)
Qlik Sense Engine Service (QES)	4747 (QES listen port)	7070 (Logging service)	4242 (QRS REST API) 4748 (notifications from QRS)
Qlik Sense Repository Service (QRS)	4242 (REST API) 4239 (from QPS -	4242 (REST API) 4243 (Proxy REST API)	4545 (Migration service)

		websocket) 4444 (Setup API - inbound on rim nodes) 4899 (from QPR)	4444 (Setup API – outbound on central node) 4747 (Engine) 4748 (Engine notification API) 5050 (Scheduler master API) 7070 (Logging service)	4570 (Certificate unlock)
	Qlik Sense Scheduler Service (QSS)	5050 (Master REST API) 5151 (Slave REST API) 5252 (Monitoring API - optional)	4242 (QRS REST API) 7070 (Logging Service) 5050 (Slave to Master) 5151 (Master to Slave)	No additional ports.
	Qlik Sense Repository Database (QRD)	4432 (default listen port for database connections)		No additional ports.
	Qlik Sense Printing service (QPR)	4899 (QPR listen port)		443 (Sense web server - proxy) 4242 (QRS REST API) 8088 (CEF debugging)
	Qlik Sense Service Dispatcher (QSD)			
	Starts up the following services:			
	Broker service	4900		3003 (Converter service) 4545 (App migration) 4555 (Chart sharing) 4949 (Data profiling) 9028 (Hub service) 9031 (Capability

			service) 9032 (About Service) 9079 (Depgraph service) 9090 (DownloadPrep) 9098 (On-demand app service) 9080 (Web extension service) 9041 (Connector registry proxy - server) 9051 (Connector registry proxy - desktop) 21060 (Resource Distribution Service) 5928 (QSE Event Processor)
Data profiling service	4949 (listen port for REST API and websocket)		4242 (QRS REST API) 4747 (QES)
App Distribution Service		5926	No additional ports.
Hybrid Deployment Service		5927	No additional ports.
Hybrid Setup Console - HSC	5929		No additional ports.
Entitlement Provisioning Service - EPS		5930	No additional ports.

To allow access to the file share, ensure that you open the Microsoft Windows SMB port 445.

Ports used internally within a node

The ports in the following table are used between Qlik Sense services that run on the same node. In most cases, the ports do not have to be open through any firewalls.

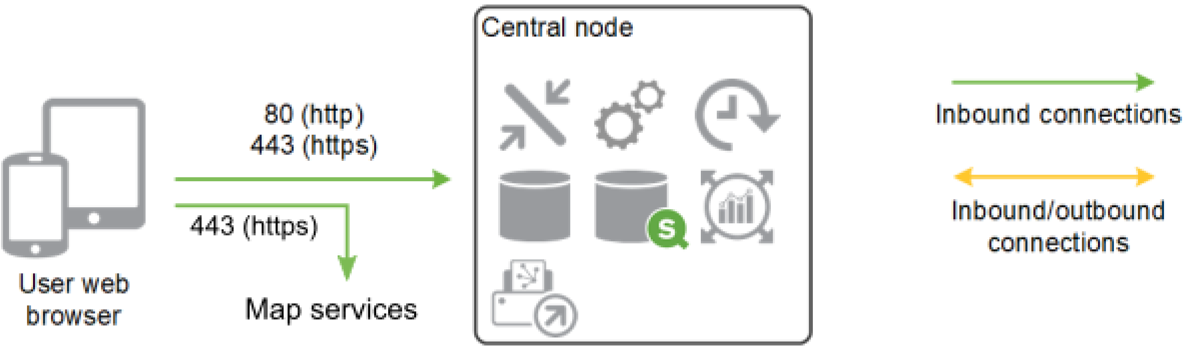
Service	Port	Direction	Purpose
Converter Service	3003	Internal	This port is used by the Converter Service which is utilized by QlikView converter.
QPS	4243	Inbound	Qlik Sense Proxy Service (QPS) REST API listen port. If web ticketing is used for security, this port is used by the software or service that requests tickets for users. If the software or service is remote, this port needs to be open to the location from which it is called.
QRD	4432	Internal	Default listen port for the Qlik Sense Repository Database (QRD). With shared persistence, this port is used to listen for connections from the Qlik Sense Repository Service (QRS).
Migration Service	4545	Internal	This port is used by the Migration Service for app migration purposes. The service is launched and managed by the Qlik Sense Service Dispatcher (QSD) when required. The Migration Service only runs on the central node.
Chart Sharing Service	4555	Internal	This port is used by the Chart Sharing Service for chart sharing between Qlik Sense users. The service is launched and managed by the Qlik Sense Service Dispatcher (QSD) when required. This port uses HTTPS for communication.
QRS	4570	Internal	Certificate password verification port, only used within multi-node sites by Qlik Sense Repository Services (QRSs) on rim nodes to receive the password that unlocks a distributed certificate. The port can only be accessed from localhost and it is closed immediately after the certificate has been unlocked. The communication is always unencrypted.
QES	4748	Internal	This callback port is used by the Qlik Sense Repository Service (QRS) for sending HTTP events to the Qlik Sense Engine Service (QES).
Data Profiling Service	4949	Internal	This port is used by the Data Profiling Service to access and modify the app load data model. It communicates directly with the Qlik Sense Engine Service (QES) on the node.
Broker Service	4900	Internal	Default listen port for the Broker Service.
Hub Service	9028	Internal	Default listen port for the Hub Service.

Capability Service	9031	Internal	This port is used by the Capability Service to handle Qlik Sense system feature configuration.
About Service	9032	Internal	Default listen port for inbound calls to the About Service.
Depgraph Service	9079	Internal	This port is used by the Service Dispatcher launched microservices.
Web Extension Service	9080	Internal	Default listen port for the Web Extension Service.
DownloadPrep	9090	Internal	his port is used by the Service Dispatcher launched microservices.
On-demand App Service	9098	Internal	Default listen port for the On-demand App Service.
Connector registry proxy (server)	9041	Internal	This port is used by the distributed connectivity service for discovering and listing connectors.
Connector registry proxy (desktop)	9051	Internal	This port is used by the distributed connectivity service for discovering and listing connectors.

Ports used from user web browser

The default ports are exposed to the Qlik Sense users and need to be open through any firewalls in the site.

Service	Port	Direction	Purpose	Host
QPS	443	Inbound	Inbound user web traffic when using HTTPS.	Qlik Sense Proxy Service (QPS) in the site.
QPS	80	Inbound	Inbound user web traffic when using HTTP (optional).	Qlik Sense Proxy Service (QPS) in the site.
Map	443	Inbound	User web traffic for standard map background. For users hosting their own map server, use the name of the host server.	maps.qlikcloud.com
Map	443	Inbound	User web traffic for satellite map background.	services.arcgisonline.com



Ports used between nodes and Qlik Sense services

The ports in this section are used for communication between the Qlik Sense services.

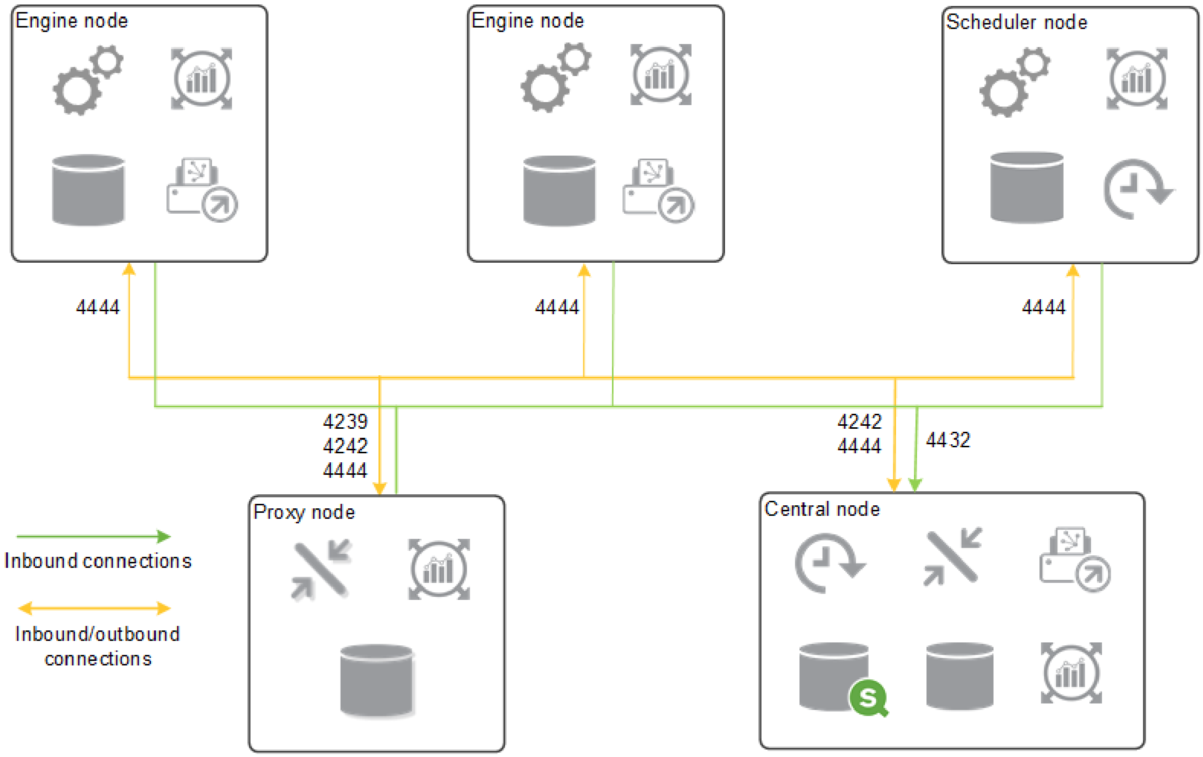
In a single node site, all ports listed in this section are used by the various services, but do not need access through firewalls.

In a multi-node site, the ports in use vary depending on the services installed and running on each node. The ports need to be open in any firewalls between the nodes, but do not have to be open to the Qlik Sense users.

Minimum ports used for communication in multi-node sites

The following ports must always be open between the nodes in a multi-node site. The ports must be open to allow for service health, and some specific operations.

Service	**Port**	**Direction**	**Purpose**
QRS	4242	Bi-directional between the central node and all proxy nodes	This port is used for a number of operations including new user registration.
QRD	4432	Inbound from Qlik Sense nodes to the repository database	The default listen port used by all nodes in a site for connecting to the Qlik Sense Repository Database.
QRS	4444	Between the central node and all rim nodes	This port has two functions: • Security distribution port, only used by Qlik Sense Repository Services (QRSs) on rim nodes to receive a certificate from the master QRS on the central node. The communication is always unencrypted, but the transferred certificate package is password-protected. • Qlik Sense Repository Service (QRS) state port, used to fetch the state of a QRS in a Qlik Sense site. The state is fetched using *http://localhost:4444/status/servicestate*. The returned state is one of the following: ◦ 0: Initializing. Once the node has been initialized, the node state changes into one of the other states. ◦ 1: Certificates not installed. There are no certificates installed on the node. The node stays in this state until it has received the certificate and the certificate password. ◦ 2: Running. The node is up and running and all APIs have been initiated.



Ports used between master and slave schedulers

The ports in the following table are used when a slave Qlik Sense Scheduler Service (QSS) is used.

Service	Port	Direction	Purpose
QSS	5050	Inbound (from scheduler nodes only)	This port is used by the master QSS on the central node to issue commands to and receive replies from slave QSS nodes.
QSS	5151	Inbound (from the central node only)	A slave QSS runs on a slave scheduler node and is accessed only by the master QSS on the central node.

Ports used between a proxy node and an engine node

The ports in the following table define the minimum needed to allow regular user traffic and load balancing between a proxy node and an engine node.

Service	Port	Direction	Purpose
QES	4747	Inbound (from proxy nodes)	Qlik Sense Engine Service (QES) listen port. This is the main port used by the QES. The port is used via the Qlik Sense Proxy Service (QPS) for communication with the Qlik Sense clients.

QRS	4239	Inbound (from proxy nodes)	Qlik Sense Repository Service (QRS) WebSocket port. The port is used via the Qlik Sense Proxy Service (QPS) by the Qlik Sense hub to obtain apps and stream lists.
QRS	4242	Inbound (from proxy nodes)	Qlik Sense Repository Service (QRS) REST API listen port. This port is mainly accessed by local Qlik Sense services. However, the port must be open to all proxy nodes in a multi-node site to deliver images and static content.
Data Profiling Service	4949	Inbound (from proxy nodes)	This port is used by the Data Profiling Service when accessing and modifying the application load model. The service is launched and managed by the Qlik Sense Service Dispatcher (QSD) when required. The port is access via the Qlik Sense Proxy Service (QPS).
Broker Service	4900	Inbound (from proxy nodes)	Default listen port for the Broker Service.
Hub Service	9028	Inbound (from proxy nodes)	Default listen port for the Hub Service. Open for local services such as the broker service on the engine node.

Ports used between a proxy node and a node running the printing service

The Qlik Sense Printing Service (QPR) may be installed on the same node as other services or on a separate node. The ports in the following table must be accessible between a QPS and all QPRs to which the QPS can load balance traffic.

Service	**Port**	**Direction**	**Purpose**
QPR	4899	Inbound (from proxy nodes)	Qlik Sense Printing Service (QPR) port. This port is used for printed export in Qlik Sense. The port is accessed by any node that runs a QPS.

Qlik Sense Desktop ports

The following ports are used by Qlik Sense Desktop.

Component	**Port**	**Direction**
Qlik Associative Engine	9076	Internal
Migration Service	9074	Internal
DataPrep Service	9072	Internal

Broker Service (Desktop)	4848	Internal/inbound
Capability Service	9075	Internal
About Service	9078	Internal
Broker Service	9070	Internal
NPrinting	9073	Internal
Hub Service	9071	Internal
Converter Service	9077	Internal
Dependency Graph Service	9033	Internal
Web Extension Service	9034	Internal
Connector Registry Proxy	9051	Internal

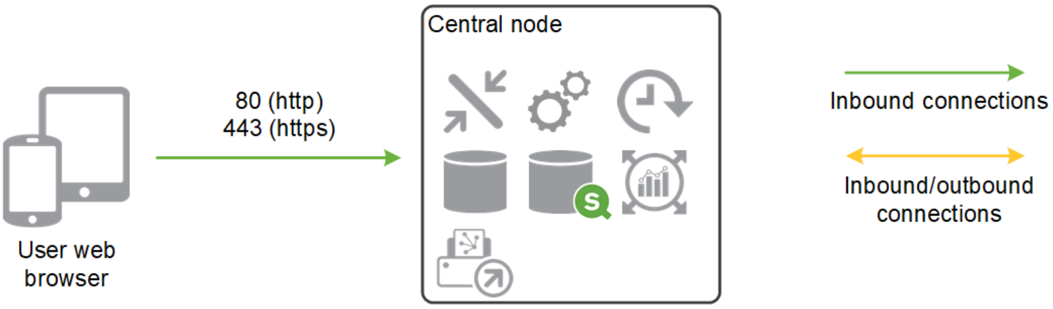
Ports examples

This section provides examples of the ports that are used in different Qlik Sense deployments.

The diagrams in this section do not show all outbound proxy node ports. For a full list of proxy node ports see the Ports overview (page 34) table.

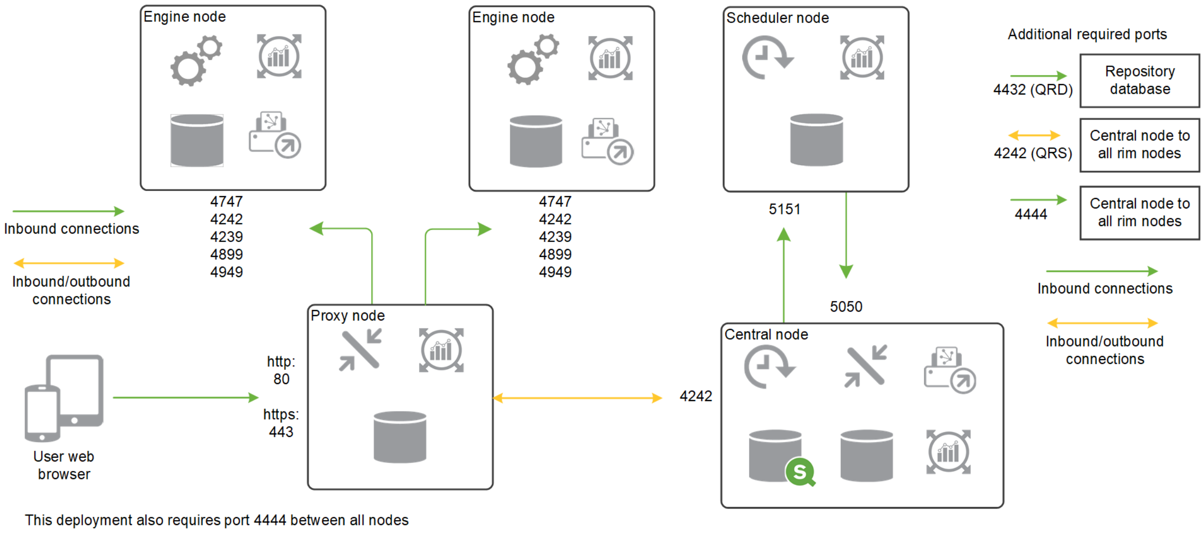
Single node site

This example shows the ports that are used in a single node site.



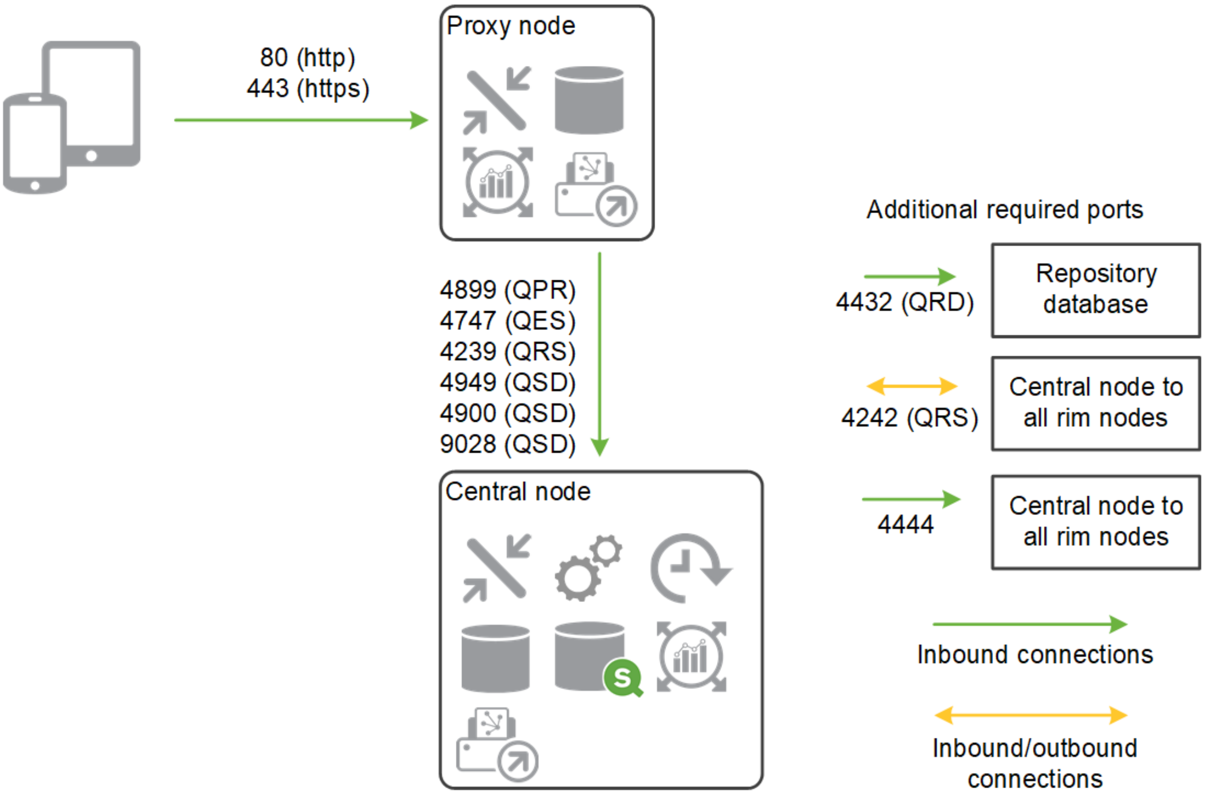
Multi-node site

The following is an example of the ports that are used in a multi-node site that consists of five nodes.



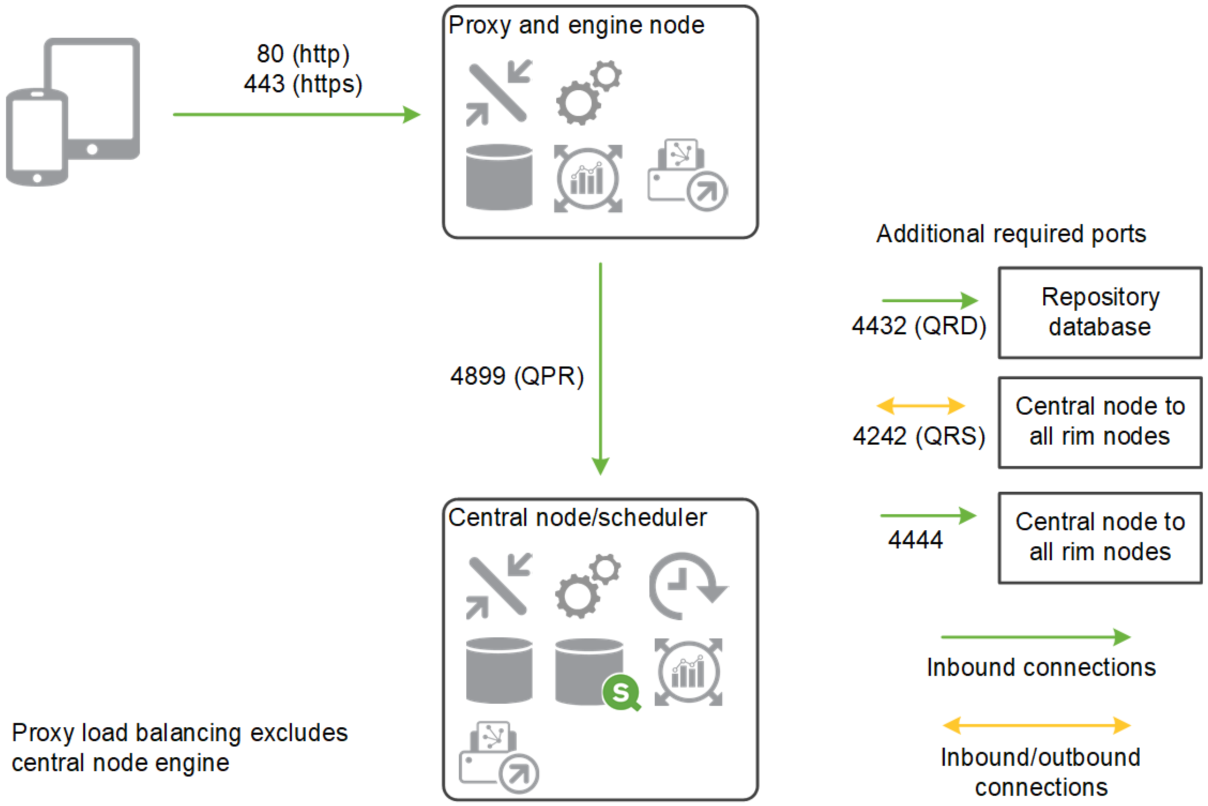
Proxy node in demilitarized zone

This example shows the ports that are used in a multi-node site when deploying a proxy node in a demilitarized zone.



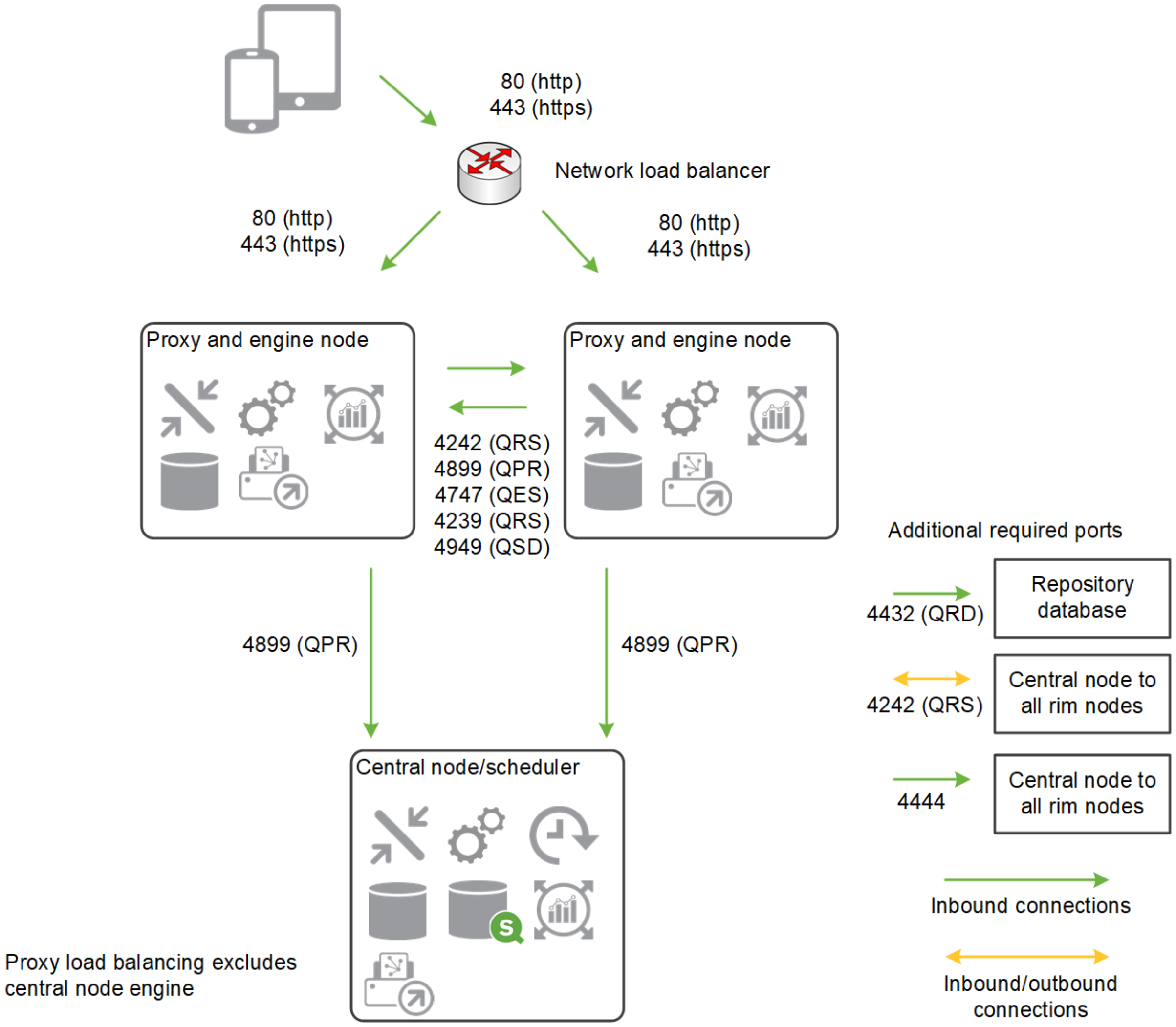
Separate proxy and engine node

This example shows the ports that are used in a multi-node site when deploying a separate proxy and engine node.


80 (http)
443 (https)
Proxy and engine node
4899 (QPR)
Central node/scheduler
Proxy load balancing excludes central node engine
Additional required ports
4432 (QRD)
Repository database
4242 (QRS)
Central node to all rim nodes
4444
Central node to all rim nodes
Inbound connections
Inbound/outbound connections

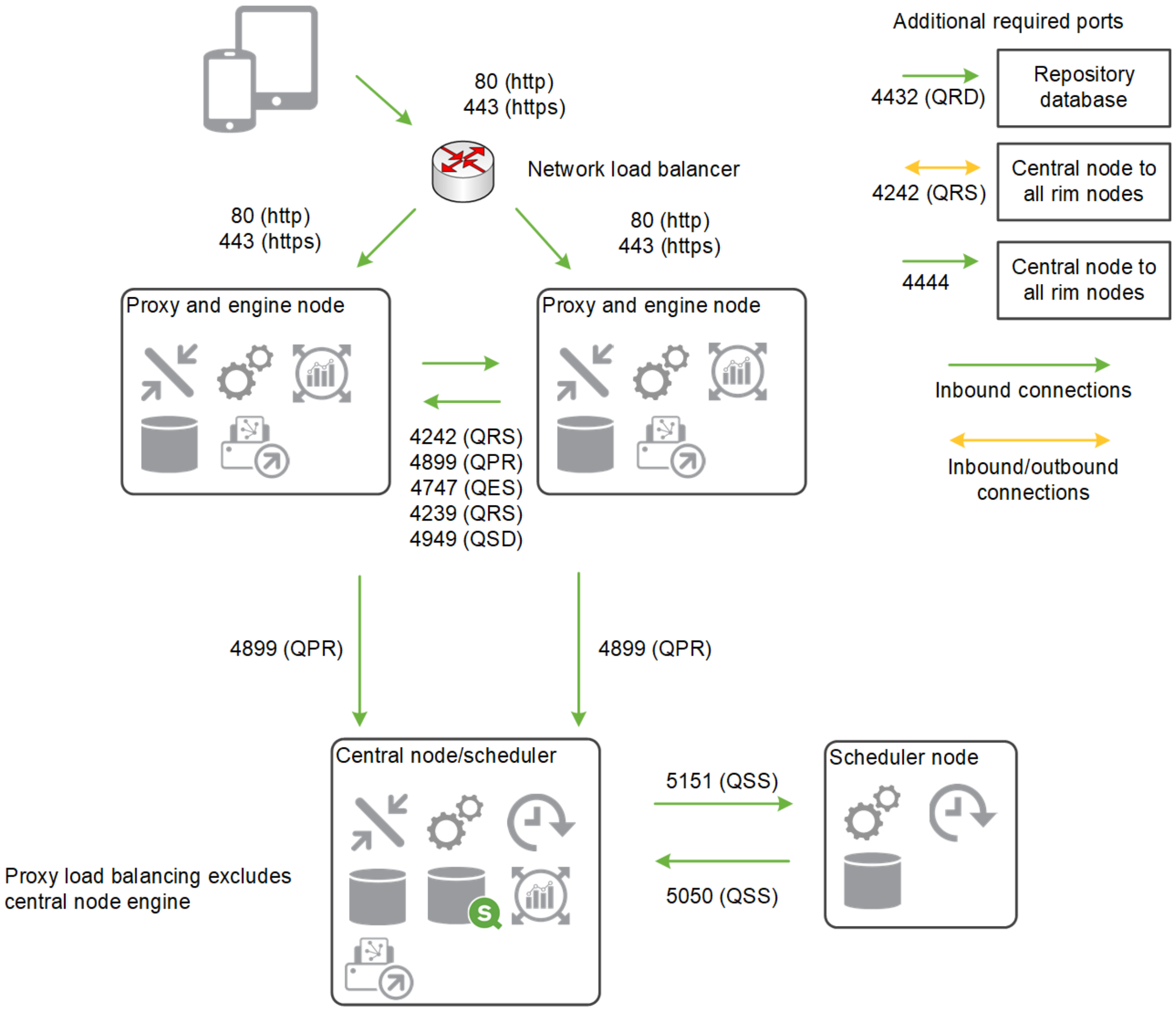
High availability proxy and engine nodes

This example shows the ports that are used in a multi-node site when deploying more than one proxy and engine node.



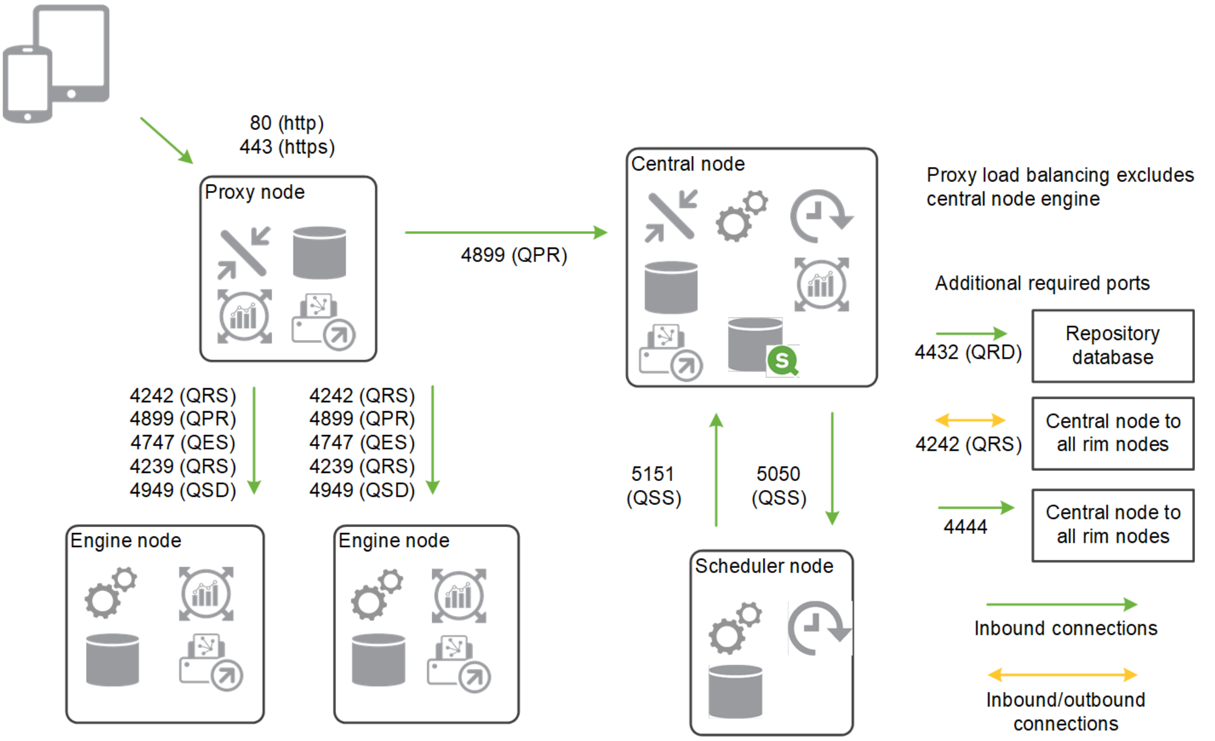
Separate scheduler node and high availability proxy and engine nodes

This example shows the ports that are used in a multi-node site when deploying a separate scheduler node and more than one proxy and engine node.



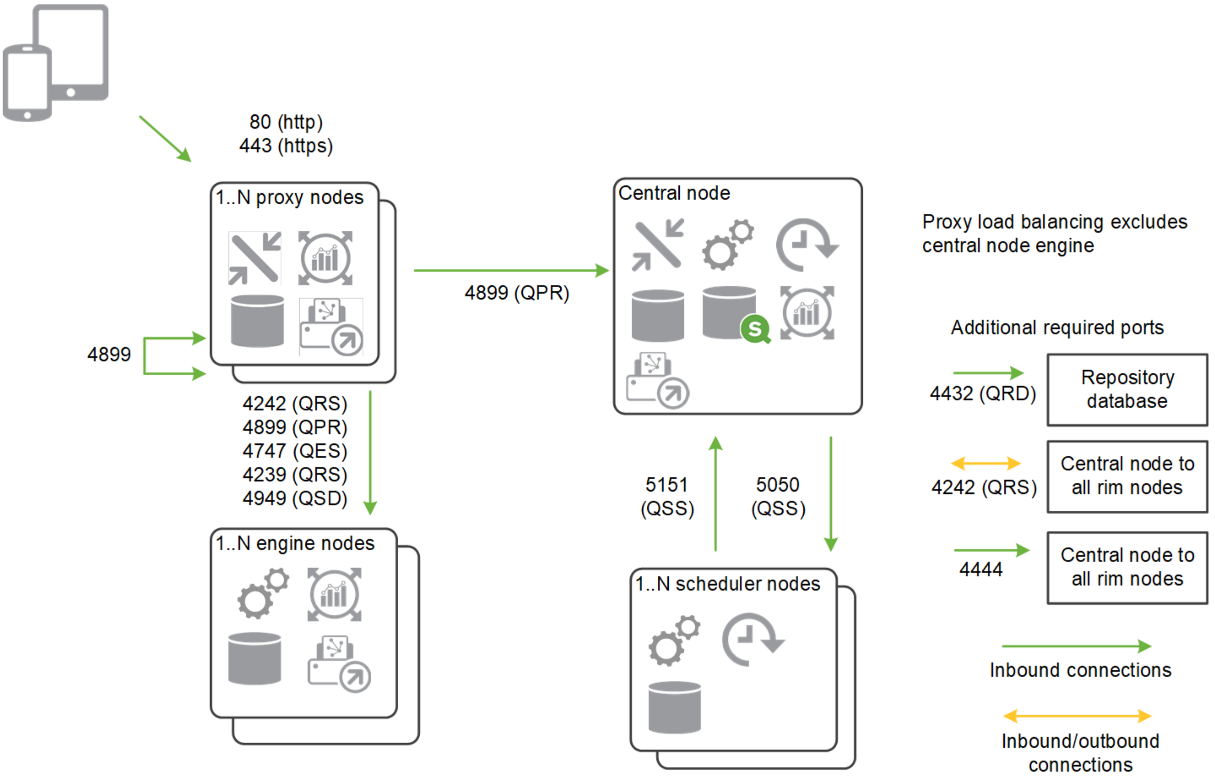
Separate proxy and scheduler nodes and high availability engine nodes

This example shows the ports that are used in a multi-node site when deploying separate proxy and scheduler nodes and more than one engine node.



Generic scale out

This example shows the ports that are used in a multi-node site when scaling the site by adding additional proxy, engine, or scheduler nodes.



Persistence

A Qlik Sense site stores data to both a repository database, and a file share. The repository database stores system and app meta data, while the file share stores binary application data such as, data models and app content. In a single node deployment, both the repository database and the files share are usually located on the same machine as the Qlik Sense services. In a multi-node deployment, a cluster is formed around a single repository database and file share. In many cases these may be on separate dedicated servers to improve resilience or performance.

For best performance we recommend that you locate all your Qlik Sense servers in the same geographic location or data center as the repository database and the file share with a network latency below 4 milliseconds.

File share

In a Qlik Sense site, a file share is used to the store the binary application data including data models and the app content. It can be located on any one of the nodes in the Qlik Sense site or on a dedicated server for better resilience and performance. You create this folder before you install Qlik Sense. See: *Creating a file share (page 97)*

The requirements for the share are:

- The Qlik Sense nodes in the cluster must have network latency below 4 milliseconds to connect to the file share server. Performance can degrade if this is not the case.
- The bandwidth to the file share must be appropriate for the amount of traffic on the site. The frequency and size of the apps being saved after reloading, and opened into memory, drives this requirement. 1 Gigabit networking is suggested.
- The file share can run on:
  - A Windows Server OS. The Windows server may have storage allocated to it from a storage area network (SAN), use local disks, or virtual storage in the case of a virtual machine.
  - A non-Windows device such as a Linux server or hardware NAS device that supports SMB 3.0.

  Qlik cannot verify support for all storage vendors, and recommends that customers test their preferred infrastructure. In the event of an issue arising that is attributed to storage, Qlik Support may request that customers replicate the issue on a Windows hosted file share.
- The file storage must have a single read and write master. Storage can be replicated to standby storage, but only one location can be used to read and write to.

Repository database

In a Qlik Sense site, a PostgreSQL repository database is used to store all data for the Qlik Sense Repository Service including system and meta data. It can be located on one of the nodes in the Qlik Sense site or on a dedicated server for better resilience and performance. If you want to install it on a dedicated server, you do this before installing Qlik Sense.

You have two options for the repository database:

- Install as a local database on a central node. This option can be used for both single-node and multi-node deployments, and is done during installation using the Qlik Sense setup program.
- Install as a remote database on a separate server. This option provides higher performance and resilience, and is the recommended approach in a multi-node deployment. See: *Installing and configuring PostgreSQL (page 98)*

The requirements for the database are:

- The Qlik Sense nodes in the cluster must have network latency below 4 milliseconds to connect to the repository database server. Performance can degrade if this is not the case.
- If you run a PostgreSQL database on a dedicated server, it must use PostgreSQL version 9.6.

> *PostgreSQL can be run on various platforms including Windows, Linux, or cloud-hosted services such as Amazon RDS. If you use Linux or Amazon RDS, it is your responsibility to install and configure a running instance of PostgreSQL for Qlik Sense to use.*

Basic deployment

In a basic single-node deployment, all services are deployed to a single server. This type of deployment is best suited to a small organization operating within a single time zone.

For larger organizations, an enterprise deployment is recommended, see *Enterprise deployment (page 50)*.

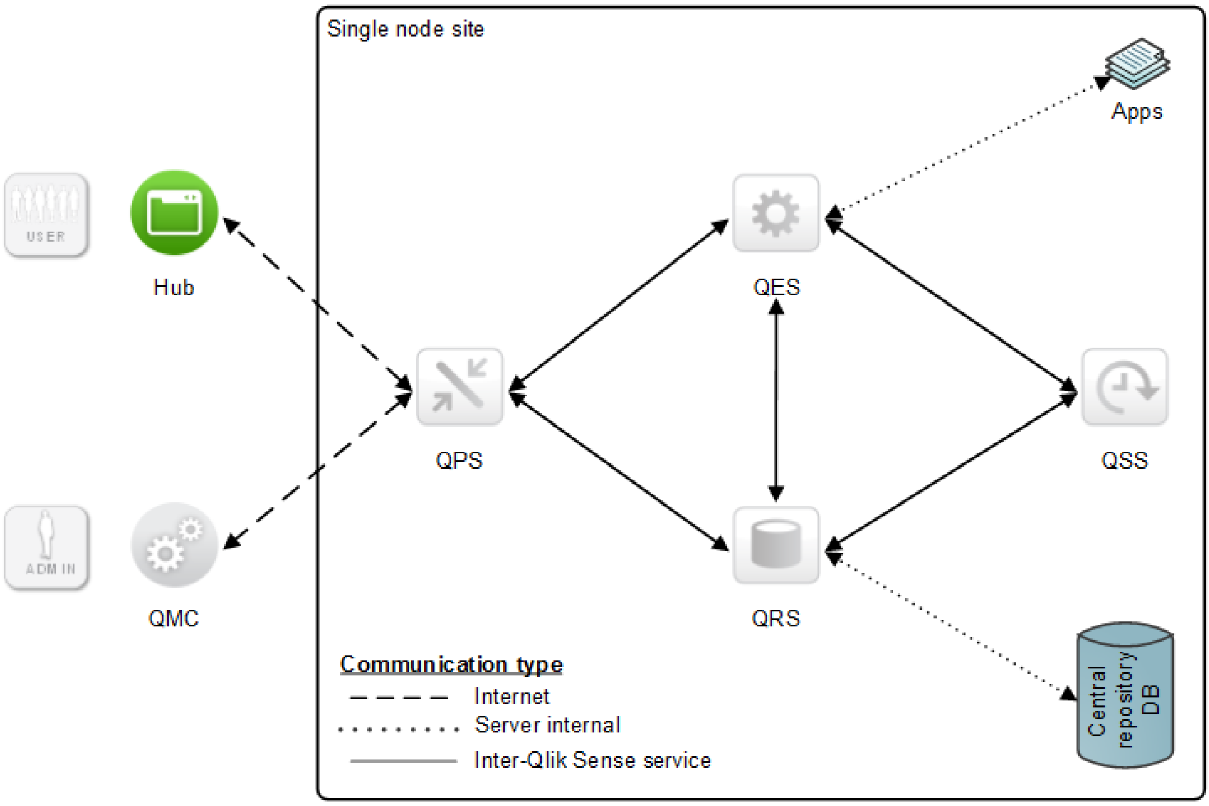
Services

In a single-node deployment, the Qlik Sense services behave as follows:

- Qlik Sense Repository Service
  Within a single node site, there is only one instance of the Qlik Sense Repository Service (QRS) running and it has direct access to the central repository database.
- Qlik Sense Scheduler Service
  When deployed in a single node site, the Qlik Sense Scheduler Service (QSS) acts as both master and slave.

Basic single-node deployment example

In this deployment scenario, all Qlik Sense services run on a single node. This kind of deployment works best in a single time zone, where reloads of data can be done during the night.



Architecture (page 20)

Enterprise deployment

You can configure a Qlik Sense enterprise deployment in a variety of different ways to suit the needs of your organization. For example, you can install Qlik Sense services to run on a single node or on multiple nodes for better performance and scalability. In a small single-node deployment, you deploy all services to a single server, which we do not recommend for larger organizations.

This section provides three examples of Qlik Sense deployments.

The following terms are used in the deployment scenarios:

- Central node: the central point for managing all nodes in a site.
- Scheduler or Reload node: reloads apps on a schedule, but does not serve content to users.
- Consumer node: serves apps to users, but is not used to create, process, or reload data.
- Development node: allows users to create and reload new apps, but does not serve normal consumer traffic.
- Proxy node: provides load balancing of user traffic to other nodes but does not contain a Qlik Sense Engine Service (QES).

An alternative to using a proxy node is to have a proxy installed on each consumer node and balance the traffic using a hardware load balancer.

Enterprise deployment examples

The scenarios described here are examples of a small, medium, large, and extra-large Qlik Sense enterprise deployments. Every deployment of Qlik Sense is different and these examples only aim to provide a rough indication of what resources would be appropriate for a given workload. The figures included here are flexible, allowing extra capacity for growth and for handling peaks in demand. They are not intended to set a maximum limit on your deployment.

If you have an attribute significantly higher than any of the figures below (such as more reloads or apps) then contact your Qlik partner and perform a full sizing exercise. For more general scalability and performance information, see *Performance (page 68)*.

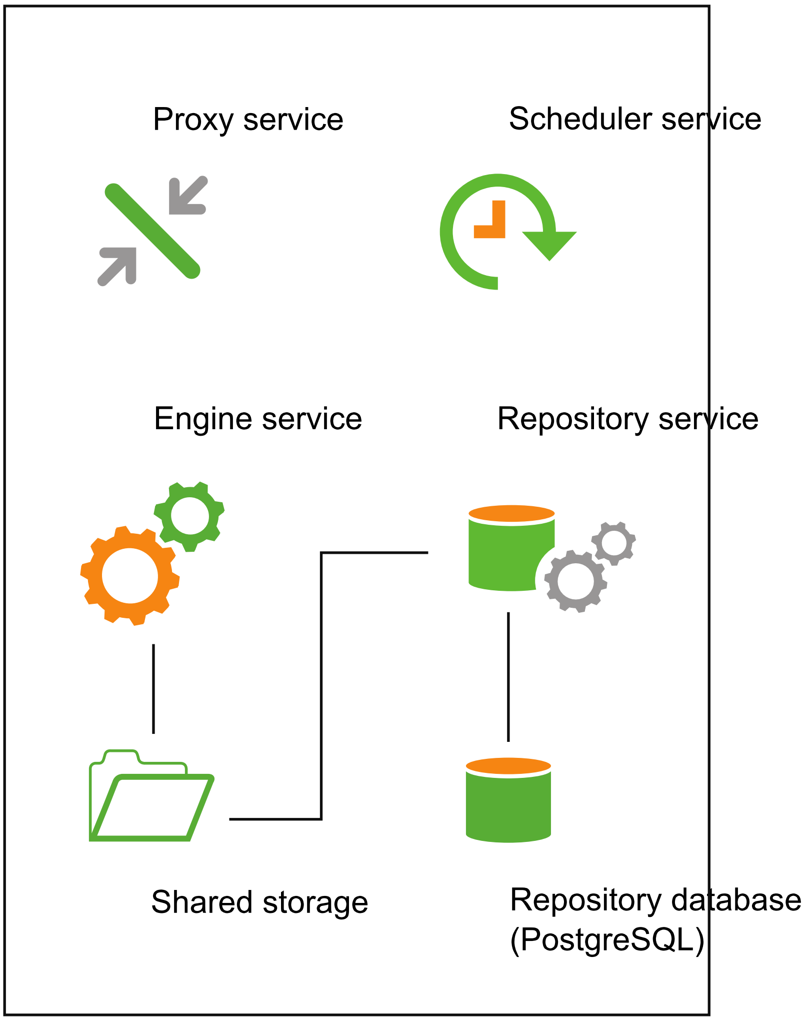
The following table provides some basic performance information for each type of deployment example:

	Single-node (small)	Multi-node (medium)	Multi-node (large)	Multi-node (extra-large)
Apps	50	100	1000	1000
Active apps per day	25	50	125	125
Total users (from UDC)	500	1000	50000	50000
Concurrent users (equals active users within the same hour)	50	100	500	1000
Average app size (in gigabytes)	0.1	0.1	0.1	0.1
Maximum app size (in gigabytes)	1	2	5	5
Content creation (objects per hour)	20	40	50	50
Reloads per hour	10	20	400	400

These figures are examples that you can use for guidance but may vary depending on how you have configured your Qlik Sense deployment.

Single-node (small)

This example illustrates a small, single-node Qlik Sense production deployment where all services are configured to run on the same server.

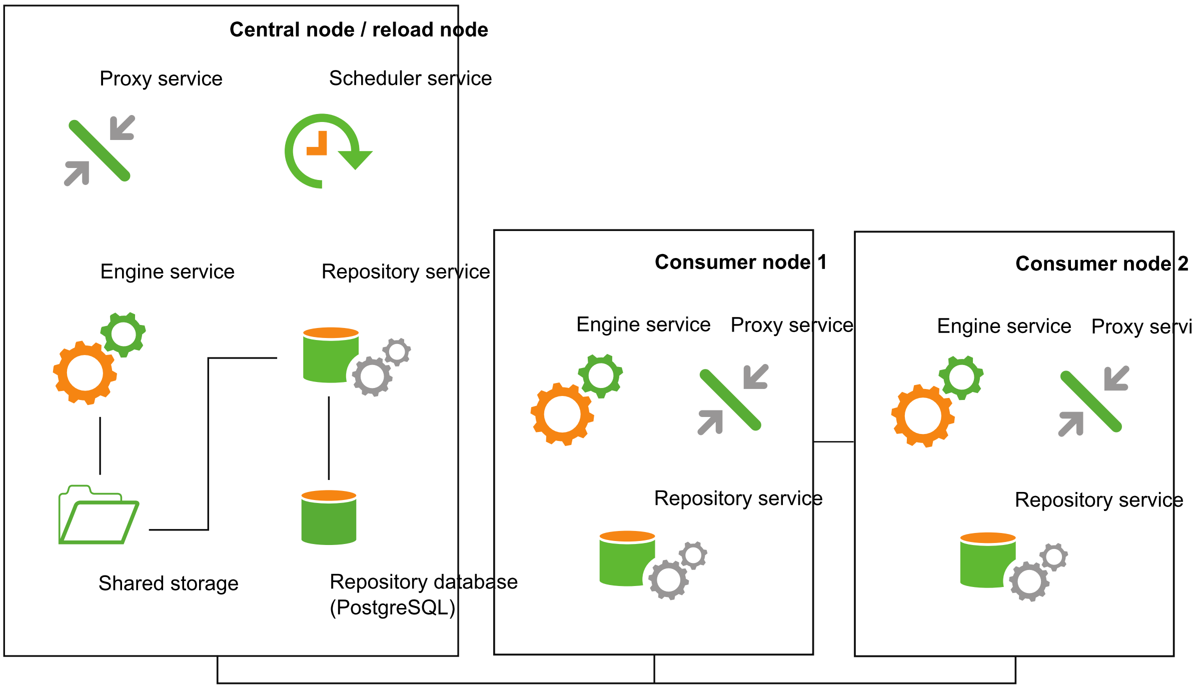


Multi-node (medium)

This example illustrates a typical medium-size, multi-node Qlik Sense production deployment consisting of three nodes:

- Central node/reload node
- Two consumer nodes

In this configuration, the repository database (PostgreSQL), and the file share are installed together with other Qlik Sense services on the central node. It has two dedicated consumer nodes.

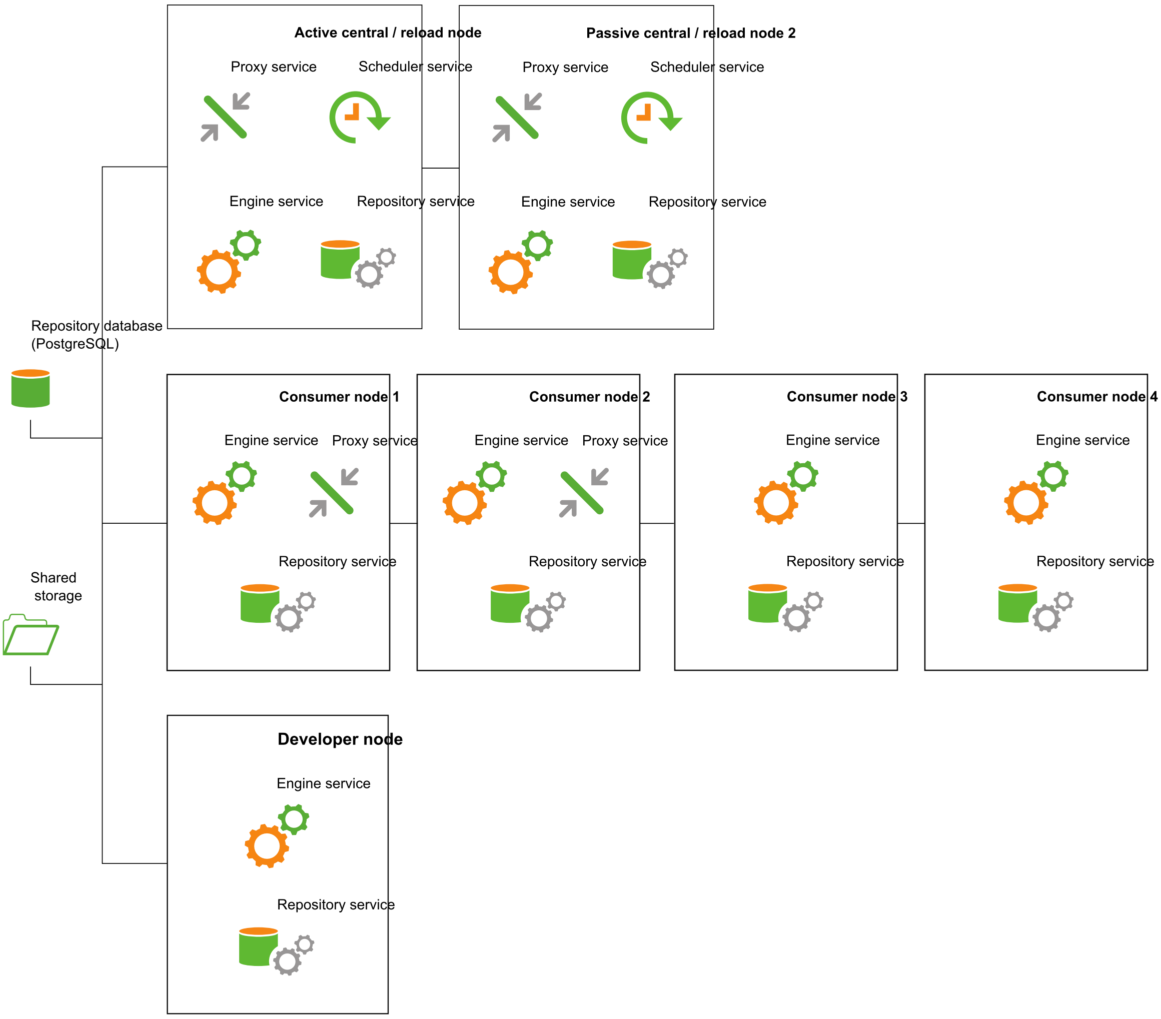


Multi-node (large)

This example illustrates a typical large, multi-node Qlik Sense production deployment providing the ability to scale up both reloads and user load. This deployment consists of the following nodes:

- Active central node/reload node
- Passive central node/reload node
- Four consumer nodes
- One developer node

In this configuration example, the repository database (PostgreSQL) and the file share are installed on separate, dedicated servers.



The active and passive central nodes must have all services installed. Configure the proxy service on consumer nodes to handle user traffic and on both the active/passive central nodes to handle admin traffic.

Only one central node can be active at any one time, while the other node remains passive. However, the scheduler service is always active regardless of whether the central node is in an active or passive state.

Multi-node (extra large)

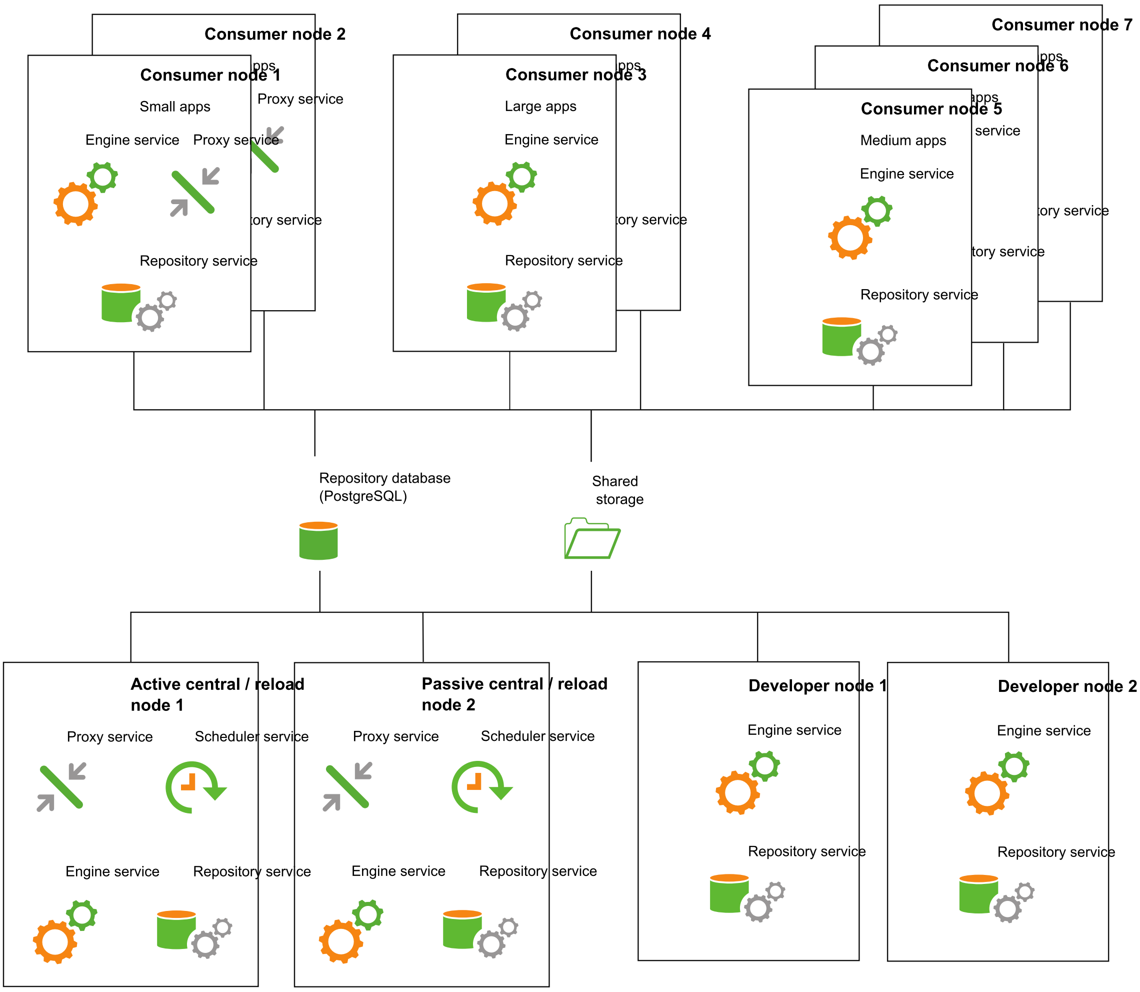
This example illustrates an extra large, multi-node Qlik Sense production deployment consisting of seven consumer nodes, providing the ability to scale up both reloads and user load. Two nodes are dedicated to large-size apps, three are dedicated to medium-size apps, and two are dedicated to small-size apps. Each consumer node can be configured with security and custom load balancing rules to restrict the size of the apps they can serve.

However, to ensure that the system can still cope with the load you can pre-load some apps in memory. For example, you could pre-load all medium and large sized apps, ensuring that they can be loaded in less than two seconds, even during peak hours. For more information on pre-loading apps, see App preload - a cache warmer.

> *With very large deployments, development of applications can be resource intensive. It may therefore be appropriate to have a separate deployment dedicated to app development. If you prefer to keep developer and consumer nodes in the same deployment, ensure the resource limits are suitable for the developer nodes. This includes reload time, hyper cube timeout, and amount of RAM.*

This deployment consists of the following nodes:

- Active central node/reload node
- Passive central node/reload node
- Seven consumer nodes
- Two developer nodes



The active and passive central nodes must have all services installed. Configure the proxy service on consumer nodes to handle user traffic and on both the active/passive central nodes to handle admin traffic.

> *Only one central node can be active at any one time, while the other node remains passive. However, the scheduler service is always active regardless of whether the central node is in an active or passive state.*

AWS deployment

In an Amazon Web Services (AWS) deployment, you install Qlik Sense Enterprise on an Amazon virtual private cloud infrastructure that is flexible, high performance, and quick to set up.

Deploying Qlik Sense Enterprise on AWS will enable you to quickly add new applications in a simple, and scalable manner. You can do this with a basic knowledge of AWS security and scalability options but without the need to follow complex on-premise installation and configuration procedures. Using AWS will enable you to get your Qlik Sense infrastructure up and running in fraction of the time required for an on-premise deployment, and will enable you to scale your deployment quickly and easily, regardless of unexpected changes in demand.

You can deploy Qlik Sense to AWS manually, or you can use an Amazon Machine Image (AMI) available in the AWS Marketplace that includes Qlik Sense preinstalled. However, predefined images do not include a file share, so can only support single node Qlik Sense deployments.

Benefits of using AWS cloud

- A quick and effective way of deploying Qlik Sense to the cloud.
- Simple and cost-effective, reducing overall deployment times.
- Quick and easy to deploy Qlik Sense applications.
- Fewer hardware management overheads.
- Scalable, elastic storage that can be expanded and contracted on demand.
- Geographic deployment to multiple regions around the world makes lower latency possible.
- A reliable and high performance platform.

Components

To successfully deploy Qlik Sense on AWS cloud you need a basic understanding of the architecture and services available in an AWS deployment. As part of a Qlik Sense deployment on AWS, you need the following components:

- An Amazon AWS account
- Amazon Management Console - available when you log in to your AWS account.
- VPC - Amazon Virtual Private Cloud
- EC2 - Amazon Elastic Cloud instance running on a VPC. Allows you to scale your deployment up and down as your requirements change.

AWS services

You should also have a basic understanding of other AWS services that you can use for managing resources and as data stores for your Qlik Sense applications:

- RDS - Managed relational database service as an alternative to a PostgreSQL repository database. Provides high availability without the same complexity.
- S3 - Simple Storage Service. Scalable, object-based cloud storage.
- Dynamo DB - NoSQL database service
- Elastic IP - remapping of IP addresses
- EMR - Elastic MapReduce. Managed Hadoop service
- Redshift - Data warehouse
- Cloud formation - for managing resources automatically

For more information about AWS services, see the Amazon AWS website.

Microsoft Windows versions

Your AWS instance needs to be running a Microsoft operating system onto which you can install a Qlik Sense instance. Qlik Sense supports the following Windows operating systems for an AWS deployment:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016

Qlik Sense Enterprise

Install a single-node Qlik Sense server on your EC2 instance.

Qlik Sense Enterprise configuration:
Use the QMC to configure the following:

- Licensing
  - Tokens (only token-based license)
  - User access (token-based license) or Professional access (user-based license)
  - CPU cores
- Security groups

Create a proxy setup for allowing HTTP access.

Other considerations

When you deploy Qlik Sense to AWS for the first time you should also consider the following.

Security

To configure security on an AWS deployment you need a good understanding of how to set up AWS security groups, key pairs, and also security groups in Qlik Sense. You use the Amazon Management Console to configure AWS security and the QMC to configure all security and authentication settings in Qlik Sense Server.

For more information about security, see *AWS and Azure security (page 74)*, and for more on Qlik Sense security, see *Security (page 71)*

Connectivity

AWS web services that you can use as data stores for Qlik Sense applications to retrieve data from when building applications:

- Amazon DynamoDB – NoSQL database
- Amazon RDS – managed relational database service
- Amazon Redshift – data warehouse as a service
- Amazon Simple Storage Service (S3) – scalable, object-based cloud storage
- AWS Elastic Map Reduce (EMR) – managed Hadoop service

In an AWS deployment you can use the following connectivity mechanisms to connect to different data sources:

- ODBC connection
- OLE DB connection
- REST API connection
- Native connector to a specific source

Connectivity scenarios:

- Qlik Sense instance that uses both data stored in Amazon RDS and Amazon Redshift.
- Qlik Sense instance that uses data coming from an AWS data source as well as a combination between flat files and web based data sources (i.e. a web service data feed).
- Hybrid Qlik Sense instance - uses data stored in AWS data sources as well as data stored on premise.

For more information about connectivity, see Connecting to data sources.

Scalability

As environments grow in terms of number of users, number and size of applications, number of data sources it is important to understand how to size the environment correctly and how to scale the environment accordingly. You need to create a multi-node environment to effectively scale up or down, by creating dedicated servers for different purposes. You can then allocate resources correctly across the following Qlik Sense services.

- Engine Service – The QIX engine, provides in-memory Associative Data Indexing and calculation supporting analysis.
- Proxy Service – Manages authentication, handles user sessions and load balancing.
- Repository Service –Manages Qlik Sense applications, controls access, and handles configuration.
- Scheduling Service – Manages reloads of Qlik Sense applications and other scheduled tasks.
- Service Dispatcher – Launch and manage the data profiling service for the data load model, migration service to make sure the app can run in the installed version of Qlik.

For more information about scalability, see the Qlik Sense Performance Benchmark technical brief.

AWS deployment example

AWS provides a cloud infrastructure with all the services and computing power you need to provide a reliable, cloud deployment platform for Qlik Sense that can performance, regardless of unexpected changes in demand, and concurrency.

Qlik Sense single-node deployment on AWS

Components in a typical Qlik Sense single-node deployment on AWS:

- VPC - Virtual Private Cloud. A logically isolated virtual network that shares a common security configuration that you define.
- Subnet - you need at least one subnet within the VPC. This could be a public, or private subnet.
  - Public subnet - subnet with direct access to the internet.
  - Private - a subnet that cannot be reached from the internet.
- RDS - Relational database service. Use this for the repository to provide high availability without the same complexity as a PostgreSQL database.
- NAT instance (optional) - restricts traffic to private subnets but allows outgoing traffic to the internet. For example, if an EC2 instance is launched inside the private network it can access the internet.
- Windows Server instance - deployed inside the default subnet to host your Qlik Sense installation.
- Security groups - act as a virtual firewall controlling which IP addresses can gain access to your instance. Use the Amazon Management Console to create a security group called *Qlik Sense*.
- Key pair - a `Qlik Sense.pem` file that you create and store locally. This file handles authentication when you connect to your AWS instance.

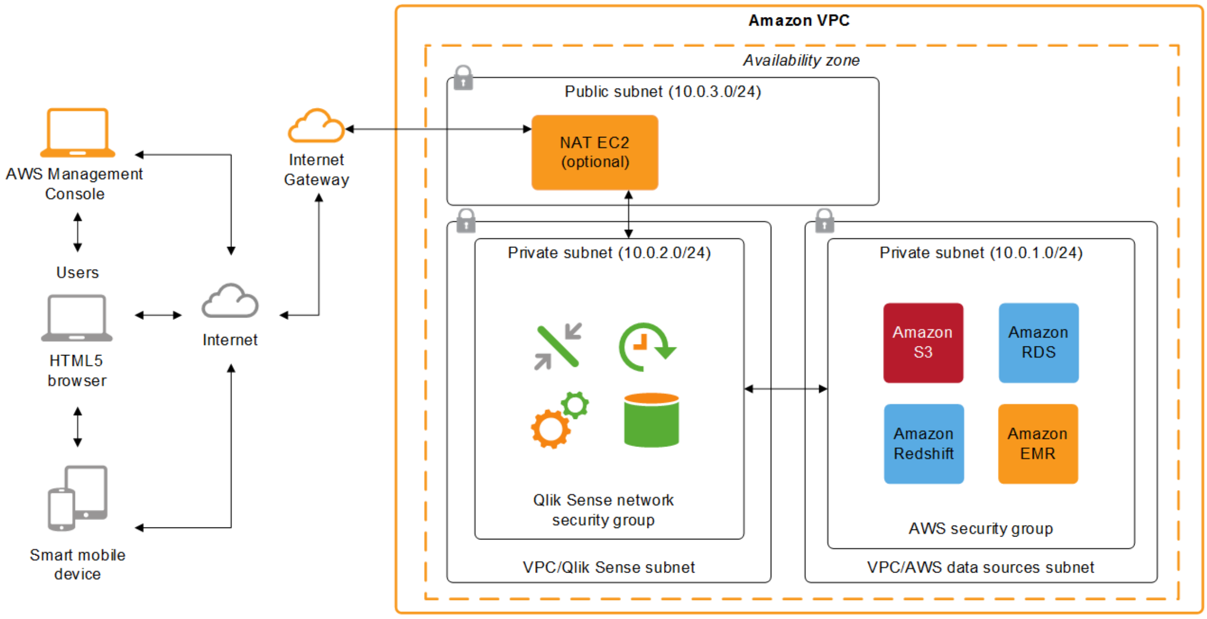
- IAM - Identity and Access Management. You need IAM to manage the fine-grained permissions required for access to different AWS services.
- Qlik Sense Server node - a single node deployed on Windows Server inside the default subnet.

Deployment options:

- Qlik Sense node in a public subnet with direct Internet access.
- Qlik Sense node in a private subnet without Internet access.

The decision whether to choose a public or private subnet in your deployment depends on your overall solution requirements.

The following example shows a complete Qlik Sense Enterprise, single node deployment on Amazon Virtual Private Cloud.



Azure deployment

In a Microsoft Azure deployment, you install Qlik Sense Enterprise on a Azure cloud infrastructure that is flexible, high performance, and is quick to set up.

Deploying Qlik Sense Enterprise on Azure will enable you to quickly add new applications in a simple, and scalable manner. You can do this with a basic knowledge of Azure security and scalability options but without the need to follow complex on-premise installation and configuration procedures. Using Azure will enable you to get your Qlik Sense infrastructure up and running in fraction of the time required for an on-premise deployment, and will enable you to scale your deployment quickly and easily, regardless of unexpected changes in demand.

You can deploy Qlik Sense to Azure manually, or you can use an Virtual Hard Disk (VHD) available in the Azure Marketplace that includes Qlik Sense preinstalled. However, predefined images do not include a file share, so can only support single node Qlik Sense deployments.

Benefits of using Microsoft Azure cloud

- A quick and effective way of deploying Qlik Sense to the cloud.
- Simple and cost-effective, reducing overall deployment times.
- Quick and easy to deploy Qlik Sense applications.
- Microsoft Server Message Block (SMB) 3.0 file system - This makes theQlik Sense file share highly resilient to failures, and AWS does not offer a similar alternative.
- Scalable, reliable and high performance cloud platform.
- Microsoft security and networking functionality.
- Geographic deployment to multiple regions around the world makes lower latency possible.
- A reliable and high performance platform.

Components

To successfully deploy Qlik Sense on Azure cloud you need a basic understanding of the architecture, and services available in an Azure deployment. As part of a Qlik Sense deployment on Azure, you need the following components:

- Azure Virtual Machine
- Azure SMB 3.0 file system storage
- Azure Virtual Network
- Azure Resource Group
- Azure Resource Manager

Azure services

You should also have a basic understanding of other Azure services that you can use for managing resources and as data stores for your Qlik Sense applications:

- Azure Portal
- Azure Active Directory and Identity Management
- Azure SQL Database – SQL Server 2016 on the Cloud
- Azure SQL Data Warehouse – Enterprise level scale-out, massively parallel processing, highly scalable database for both relational and non-relational data.
- Azure Storage – scalable cloud storage (Blob Storage, Table Storage, Azure Queues and Azure Files)
- Azure HDInsight – elastic map reduce (Hadoop as Service)

For more information about Azure services, see the ⇨ Microsoft Azure website.

Microsoft Windows versions

Your Azure instance needs to be running a Microsoft operating system onto which you can install a Qlik Sense instance. Qlik Sense supports the following Windows operating systems for an Azure deployment:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016

Qlik Sense Enterprise

Install a single-node Qlik Sense server on your Azure instance.

Qlik Sense Enterprise configuration:
Use the QMC to configure the following:

- Licensing
  - Tokens (only token-based license)
  - User access (token-based license) or Professional access (user-based license)
  - CPU cores
- Security groups

Create a proxy setup for allowing HTTP access.

Other considerations

When you deploy Qlik Sense to Azure for the first time you should also consider the following.

Security

Use the Resource Manager to configure Azure security and the QMC to configure all security groups and authentication settings in Qlik Sense.

For more information about security, see *AWS and Azure security (page 74)*, and for more on Qlik Sense security, see *Security (page 71)*

Connectivity

Qlik Sense applications can use the following Azure web services as data stores:

- Azure SQL Database – SQL Server 2016 on the Cloud
- Azure SQL Data Warehouse – enterprise level scale-out, massively parallel processing, highly scalable database for both, relational and non-relational data
- Azure Storage – scalable cloud storage (Blob Storage, Table Storage, Azure Queues and Azure Files)
- Azure HDInsight – elastic map reduce (Hadoop as Service)

In an Azure deployment you can use the following connectivity mechanisms to connect to different data sources:

- ODBC connection
- OLE DB connection
- REST API connection
- Native connector to a specific source

Connectivity scenarios:

- Qlik Sense instance that uses data stored in Azure SQL Database and Azure SQL Data Warehouse.
- Hybrid Qlik Sense instance - uses data stored in Azure data sources as well as data stored on premise.

For more information about connectivity, see Connecting to data sources.

Scalability and sizing

As your environment grows in terms of number of users, number and size of applications, and the number of data sources, it is important to understand how to size and scale your environment correctly. Resources need to be allocated correctly across the following Qlik Sense services:

- Engine Service – The QIX engine, provides in-memory Associative Data Indexing and calculation supporting analysis
- Proxy Service – Manages authentication, handles user sessions and load balancing
- Repository Service –Manages Qlik Sense applications, controls access, and handles configuration
- Scheduling Service – Manages reloads of Qlik Sense applications and other scheduled tasks
- Service Dispatcher – Launch and manage the data profiling service for data load model, migration service to make sure the app can run in the installed version of Qlik (runs on the central node only) and chart sharing between two users

For more information about scalability, see the Qlik Sense Performance Benchmark technical brief.

Azure deployment example

Microsoft Azure provides a cloud infrastructure with all the services and computing power you need to provide a reliable, cloud deployment platform for Qlik Sense that can performance, regardless of unexpected changes in demand, and concurrency.

Qlik Sense single-node deployment on Azure

Components in a typical Qlik Sense deployment on Azure:

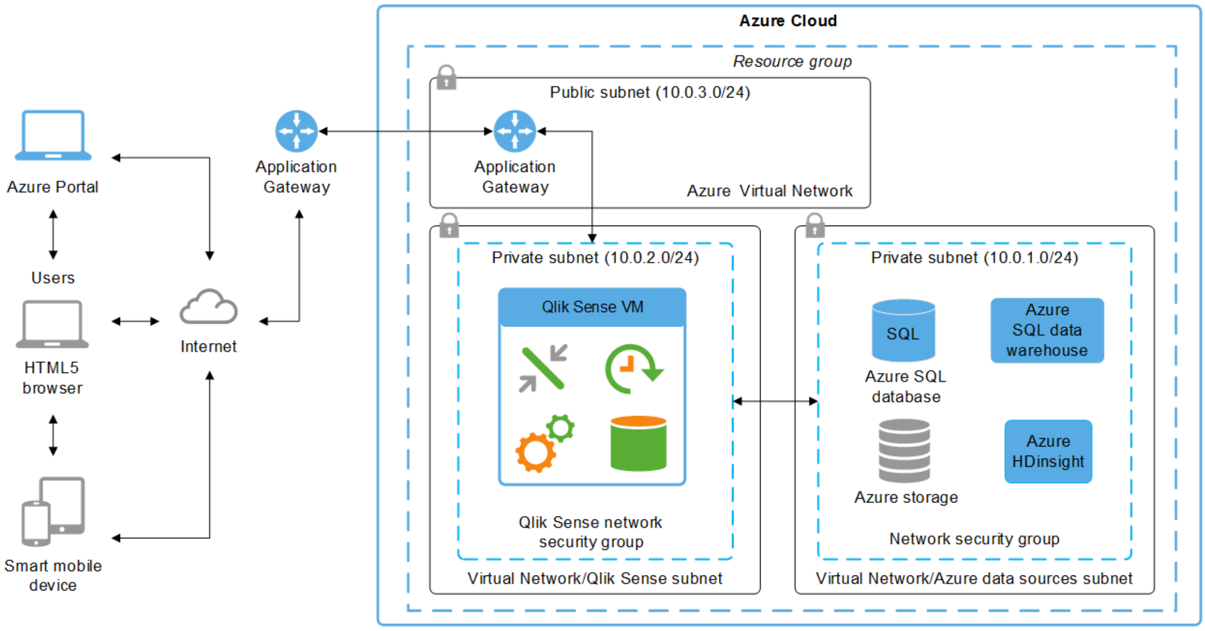
- Azure Virtual Network (VNet) - a logically isolated area of the Azure cloud where you can launch Azure resources in a virtual network that you define.
- Subnet - you need at least one subnet (either public or private) within the Virtual Network. This could be a public or private subnet.
  - Public subnet - subnet with direct access to the internet.
  - Private - a subnet that cannot be reached from the internet.
- Virtual Machine - A Windows Server virtual machine instance deployed in the default subnet onto which you can install and configure your instance of Qlik Sense server.
- Resource Group/Resource Manager - enables you to deploy, manage, and monitor the different components in your Microsoft Azure solution as a group. This makes it easier to deploy, update or delete components in a single, coordinated operation using the Resource Manager.
- Network Security groups - a list of Access Control List (ACL) rules that allow or deny network traffic to the Virtual Machine instances in a Virtual Network.
- Azure Active Directory and Identity Management - depending on the expected administration of the environment, integration with Azure Active Directory and Identity Management may be needed to manage fine-grained permissions for access to various Azure services involved in the deployment process.
- Qlik Sense Server node - a single node deployed on Windows Server inside the default subnet.

Deployment options:

- Qlik Sense node in a public subnet with direct Internet access.
- Qlik Sense node in a private subnet without Internet access.

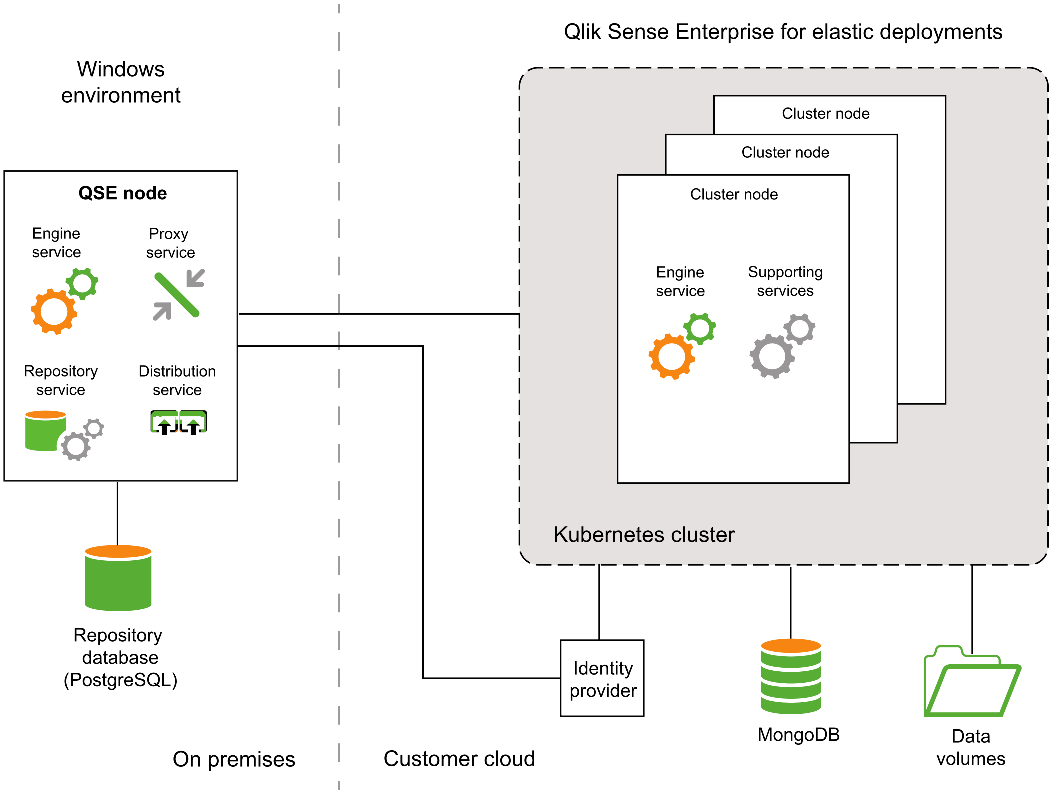
The decision whether to choose a public or private subnet in your deployment depends on your overall solution requirements.

The following example shows a complete Qlik Sense Enterprise, single node deployment on Azure Cloud.



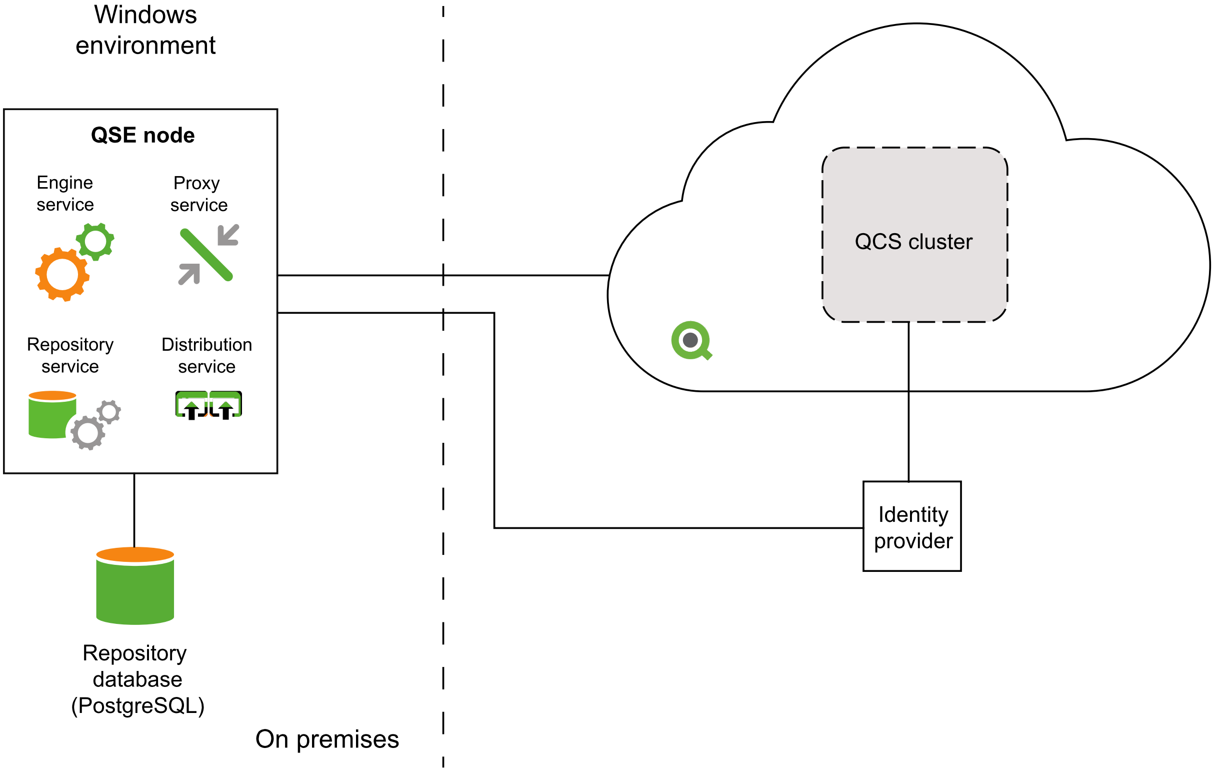
Qlik Sense Enterprise for elastic deployments

This diagram shows an example of a Qlik Sense Enterprise (QSE) elastic deployment with a single Kubernetes cluster connected to a Qlik Sense Enterprise for Windows node. The cluster contains one or more of the Qlik Sense microservices such as the Engine or other services deployed across a set of nodes. This deployment provides the ability to scale up the number of apps (read only) for user consumption. The Kubernetes cluster, which is deployed within a public or private cloud, shares data volumes and a MongoDB instance. An Identity Provider (IdP) authenticates users while QSE authorizes access to multi-cloud apps using built-in security rules. The IdP allows the same named users to access content in Qlik Sense Enterprise and the cloud environment, subject to security rules. The Kubernetes cluster, public or private cloud, and network infrastructure are all managed by the customer.



Qlik Cloud Services deployment

This diagram shows Qlik Cloud Services (QCS) connected to a Qlik Sense Enterprise node. This deployment provides the ability to scale up the number of apps (read only) for user consumption. The QCS cluster is deployed as a fully-managed service provisioned and administered by Qlik.



2.4 Licensing

The License Enabler File (LEF) defines the terms of your license and the access types that you can allocate to users. There are two major license types: one that is user-based and one that is token-based.

- User-based license: you can allocate professional access and analyzer access. The LEF determines the distribution of the two access types.
- Token-based license: you can allocate user access and login access. The LEF determines the number of tokens that you can allocate to the two access types.

In addition to these two types there is the analyzer capacity license that is similar to analyzer access regarding available features, but where consumption is time based (analyzer time).

An access type allows users to access streams and apps within a Qlik Sense site.

User-based licenses

User-based licenses grant a predefined number of professional and analyzer access allocations. The distribution of the access types is determined by the LEF.

Professional access

You allocate professional access to an identified user to allow the user to access streams and apps within a Qlik Sense site. The professional access is intended for users who need access to all features in a Qlik Sense installation. A user with professional access can create, edit, and publish sheets or apps, and make full use of the available features.

Analyzer access

You allocate analyzer access to an identified user to allow the user to access streams and apps in the hub. The analyzer access is intended for users who consume sheets and apps created by others. A user with analyzer access cannot create, edit, or publish sheets or apps, but can create and publish stories, bookmarks and snapshots based on data in apps.

Analyzer capacity license

Analyzer capacity is a consumption-based license type that is similar to analyzer access regarding available features. Analyzer capacity is available to any users, including anonymous users, and they share the monthly analyzer time allotment, which is consumed in units of six minutes.

Token-based licenses

Licensing allows you to manage the usage of the Qlik Sense software in your organization. The licensing in Qlik Sense is based on tokens, which you use to allocate access passes that give users access to Qlik Sense.There are different types of access passes to choose from and each type corresponds to a specific consumption model:

- User access pass - assigned to unique and identified users allowing them unlimited use of Qlik Sense apps.
- Login access pass - allocates a block of passes to a group for infrequent or anonymous access. Allows full access for a limited period.

For more information on types of access passes and the consumption model, see Managing licenses.

> *You cannot use QlikView licenses with Qlik Sense as the tokens are not compatible with the Client Access Licenses (CALs) used in QlikView.*

Every Qlik Sense site needs at least one License Enabler File (LEF). This file defines the number of tokens available in your site, which you can manage from the central node. When you enter the correct serial number and control number in the Qlik Management Console (QMC) and click **Apply**, the LEF is automatically downloaded. If you do not have a network connection, you can paste the LEF directly into the LEF text box on the **Site license properties** page in the QMC.

If you purchase more tokens, they are added to the pool of unallocated tokens that you can use to allocate access passes in Qlik Sense. As you use up your tokens, any unallocated tokens are removed. However, any tokens that are freed up by the removal of access passes cannot be used for new allocations until the number of allocated tokens drops below the number set in the LEF.

For more information about the LEF, see *License Enabler File (page 295)*.

> *If you want to set up Qlik Cloud Services or Qlik Sense Enterprise for elastic deployments, please contact your Qlik representative or Qlik Support to obtain a valid license for the setup.*

2.5 Performance

This topic aims to provide some basic information on performance to consider before you install Qlik Sense. There are several different considerations to think about when planning your Qlik Sense deployment:

- Size of deployment - small single-node, medium, or large multi-node site?
- Number of nodes in your site?
- Local or dedicated repository database?
- Local or network file share?
- Number of CPU cores required for each node?
- RAM required for each node?

Geographical deployments

The current persistence model does not support geographical deployments. For best performance we recommend that you locate all your Qlik Sense servers in the same geographic location or data center as the repository database and the file share with a network latency below 4 milliseconds.

Capacity and performance

Qlik Sense supports up to a maximum of 12 nodes. In addition to the number of nodes, there are other factors that contribute to total capacity:

- Workload
- Hardware speed
- Network speed

For example, if the disk speed of the file share and the central node is too slow, you may expect low performance during some operations, such as importing or duplicating apps.

We recommend scalability testing and engaging with Qlik consulting services for larger deployments.

DMZ deployments

All nodes in a site, including nodes without an engine, require access to both the database and file share. In demilitarized zone (DMZ) deployments this may require opening additional ports, or taking an alternative approach, compared to a DMZ deployment with synchronized persistence.

Central node dependencies

The central node is responsible for handling a number of vital operations on your site. If the central node fails, some operations will fail to run, including:

- Master scheduler - responsible for triggering reloads
- License distribution - allowing new users to obtain a license
- Extension objects

To reduce the dependency on the central node you can configure one or more nodes as a failover candidate. For more information, see *Failover (page 97)*.

2.6 User accounts

In order to successfully install and deploy Qlik Sense you must set up some user accounts before you start your Qlik Sense installation.

Windows user accounts are created and configured using your Windows server administration tools.

If you choose to manually install and configure your PostgreSQL repository database, users are created and configured using your PostgreSQL database administration tools. If you choose to have Qlik Sense install the repository database for you, the Qlik Sense setup wizard will create the users during installation.

The following are the users that you may need to create before you install Qlik Sense:

- Windows Qlik Sense services administrator
- Windows Qlik Sense services user that is not an administrator
- PostgreSQL database superuser
- Qlik Sense Repository Database administrator

You must create the required Windows user accounts before you install Qlik Sense because you are prompted to enter them during the installation. If you choose to install as a Windows local administrator and wish to change to a Windows dedicated Qlik Sense service user after installation, see *Changing the user account to run Qlik Sense services (page 103)*.

When you create your Windows user accounts you must set a password for each one. Windows user account passwords may expire in accordance with the Windows domain security rules settings. If you do not update the passwords for each Windows service setting, the services will stop working. To avoid this, you can select the **Password never expires** check box in the Windows user profile, if your security protocol allows it.

Windows Qlik Sense services administrator

We recommend that you use a dedicated Windows user account to run the Qlik Sense services. If your dedicated Windows Qlik Sense services user is an administrator, you can login as that user to install Qlik Sense. If your dedicated Windows Qlik Sense services user is not a local administrator, you must use an administrator account to install Qlik Sense.

Windows Qlik Sense services user that is not an administrator

If you wish to use a dedicated Windows user account that is not an administrator to run the Qlik Sense services, you must create that account before you install Qlik Sense. The Windows Qlik Sense services user runs the following services:

- Qlik Sense Repository Service
- Qlik Sense Proxy Service
- Qlik Sense Engine Service
- Qlik Sense Scheduler Service

- Qlik Sense Printing Service
- Qlik Sense Service Dispatcher

For more information about services, see *Services (page 22)*.

The Windows Qlik Sense services user that is not an administrator must meet the following requirements:

- Member of the **Qlik Sense Service Users** and **Performance Monitor Users** groups.
  You add the Windows Qlik Sense services user that is not an administrator to these groups after you install Qlik Sense.
- Only used for Qlik Sense Windows services. This is necessary to avoid conflicts with other Windows services in the same computer.

PostgreSQL database superuser

The PostgreSQL database superuser is a role that bypasses all permission checks, except the right to log in. It is not a Windows, or Qlik Sense user, it is a PostgreSQL user configured in the repository database.

If you choose to install the PostgreSQL database manually, you are prompted to create a PostgreSQL database superuser and password during installation. That user ID and password are used to connect your PostgreSQL database. For details about creating users with the PostgreSQL administration tools, see *Installing and configuring PostgreSQL (page 98)*.

If you choose to install the Qlik Sense Repository Database locally during the Qlik Sense installation, the PostgreSQL installation is done automatically.

> *When you install Qlik Sense, if you select the Install local database option, the QSR, SenseServices, and QSMQ databases are created automatically. These databases also share the same PostgreSQL login role. For more information, see Installing and configuring PostgreSQL (page 98)*

Qlik Sense Repository Database administrator

The Qlik Sense Repository Database administrator role has full access to the Qlik Sense Repository Database that contains all configuration data for the Qlik Sense site. It is not a Windows, or Qlik Sense user, it is a PostgreSQL user configured in the repository database.

If you choose to install PostgreSQL manually, the Qlik Sense Repository Database administrator is also created manually using the PostgreSQL administration tools. For details about creating users with the PostgreSQL administration tools, see *Installing and configuring PostgreSQL (page 98)*. You must enter the location of the Qlik Sense Repository Database and the login credentails for the Qlik Sense Repository Database administrator during the Qlik Sense setup on the ***Shared persistence database connections settings*** page.

If you choose to install the Qlik Sense Repository Database locally using the Qlik Sense setup, you are prompted to set a user name and password for the Qlik Sense Repository Database administrator during the setup.

You must keep that password for backup and restore activities. It may also be needed for support.

User accounts for the logging database

Two user accounts, which use PostgreSQL password authentication, are automatically created during Qlik Sense installation. User account qlogs_writer is used internally by the logging service to write to the database. In fact, this user owns the logging database QLogs. User account qlogs_reader is used by the monitoring apps to read from the database. There is also a user account called qlogs_users, which is basically a group. It does not have a password and cannot be used to access the database. It exists only for the purpose of managing network access to the PostgreSQL database.

The system administrator can change the passwords for these database users directly from PostgreSQL. The logging service must also be updated with the new passwords using the `update` or `setup` command.

2.7 Security

Security and availability in a shared persistence deployment

In shared persistence deployments the network traffic between the servers, the database and the file share is not encrypted by default after an installation. You may also need to consider setting up replication of the database to handle cases where the central database fails.

Maintaining database password integrity

Here are some guidelines to maintain password integrity in a Qlik Sense shared persistence deployment.

- It is important that you disable the **Store password option** for your user in PostgreSQL. If this option is enabled, the password is stored in a file, and incoming connections without a password will be able to connect to the database.
- Change password by executing this query in the PostgreSQL database:
  `ALTER USER <user> WITH PASSWORD '<newpassword>';`
  `ALTER ROLE` is displayed after successfully changing the password.
  Do not change password in the PostgreSQL user interface for the same reasons as above.
- Use md5 hashing.
- Do not set your password to `PASSWORD ''`, that is, an empty string, since this is not handled well in PostgreSQL.

Database traffic encryption

Qlik Sense supports database traffic encryption using SSL, but you need to perform some manual configuration to setup SSL and MD5 password protection in a shared persistence deployment:

Do the following:

1. Edit the following values in *postgresql.conf*:
   `listen_addresses = '*'`

```
port = 4432
ssl = on
ssl_cert_file = 'server.pem'
ssl_key_file = 'server_key.pem'
#ssl_ca_file = ''
#ssl_crl_file = ''
```

2. Add the following lines in pg_hba.conf
   ```
   hostssl    all           all           all          md5
   ```
3. Remove any other lines starting with `hostssl` or `host` in *pg_hba.conf*.
4. Copy *server.pem*, and *server_key.pem* from *%PROGRAMDATA%\Qlik\Sense\Repository\Exported Certificates\.Local Certificates* to *%PROGRAMDATA%\Qlik\Sense\Repository\PostgreSQL\9.6*.
5. Use the **Connection String Editor** to add the following setting to the *repository.exe.config* on the central node, and all rim nodes that belong to the cluster. To open the **Connection String Editor**, navigate to *C:\Program Files\Qlik\Sense\Repository\Util\QlikSenseUtil* and open the *QlikSenseUtil.exe* file as an administrator.
6. In the **Connection String Editor** tab, click **Read** to open the *Repository.exe* file connection string.
7. Add '`Ssl Mode=Require;`' to the connection string:
   ```
   <add name="QSR" connectionString="User ID=qliksenserepository;Ssl
   Mode=Require;Host='fullhostname.com';Port='4432';Database=QSR;Pooling=true;Min Pool Size=0;Max
   Pool Size=90;Connection Lifetime=3600;Unicode=true;Password='randompass';"
   providerName="Devart.Data.PostgreSql" />
   <add name="QSMQ" connectionString="User ID=qliksenserepository;Ssl
   Mode=Require;Host='fullhostname.com';Port='4432';Database=QSMQ;Pooling=true;Min Pool
   Size=0;Max Pool Size=90;Connection Lifetime=3600;Unicode=true;Password='randompass';"
   providerName="Devart.Data.PostgreSql" />
   ```
8. Click **Save value in config file encrypted** to save your changes.
9. Start all Qlik Sense services and verify that everything works.
10. Verify the authentication using the pgAdmin tool in PostgreSQL:
    Users postgres and qliksenserepository must enter a valid password to connect.

Forcing the database connection to use TLS 1.2 only

You can configure the database connection to support TLS 1.2 only, and block connections using TLS 1.1 or lower.

Do the following:

- Add the following parameter to the connection string: `"SSL TLS Protocol=1.2"`

We recommend these additional configuration changes to maintain database integrity:

- Configure the database to only accept connections from servers where the repository is running.
- Configure SSL to reject weak cipher suites by adding this line to the file *postgresql.conf*:
  ```
  ssl_ciphers = 'DEFAULT:!LOW:!EXP:!eNULL:!aNULL:!MD5:!RC2:!RC4:!DES:@STRENGTH'
  ```

Database replication and failover

This section describes how to set up database replication and failover in a shared persistence environment. Additionally, the file storage content will also need to be replicated. To fail over to a standby node in case the central database or node is lost, one or more standby databases can be configured for streaming replication from the database on the primary node.

When editing text files related to the Qlik Sense installation, do the following:

1. Copy the file to another location on the server.
2. Edit the file and save the changes.
3. Copy the updated file back to its original location.

Setting up replication to standby nodes for failover

The instructions in this section describe how to set up asynchronous streaming replication to one or more standby nodes. Before starting, ensure that the environment is configured and running, and install PostgreSQL on a standby machine.

> *The paths in the instructions are adapted to a default PostgreSQL installation used as database on a dedicated machine. If you are using a PostgreSQL database installed by Qlik Sense you need to adapt the paths used, as the database is installed in %ProgramData%\Qlik\Sense\Repository\PostgreSQL\<version>\.*

Configure the primary database server

On the primary database server, do the following:

1. Open the file *%ProgramFiles%\PostgreSQL\9.6\data\postgresql.conf*
   Locate and set the following settings
   ```
   wal_level = replica
   max_wal_senders = 3
   wal_keep_segments = 8
   hot_standby = on
   ```
2. Create a user account that can be used for replication. To do so from a command prompt, run the following command. Adjust the hostname as needed, and specify a suitable password. You may be prompted for a password, this is the password that was specified during installation.
   ```
   "C:\Program Files\PostgreSQL\9.6\bin\psql.exe" -h <machinename> -p 4432 -W -c "CREATE USER replicator REPLICATION LOGIN ENCRYPTED PASSWORD 'secretpassword';"
   ```
3. Open the file *%ProgramFiles%\PostgreSQL\9.6\data\pg_hba.conf*.
   At the bottom of the file add:
   ```
   host    replication    replicator      0.0.0.0/0md5
   ```
   You can restrict the subnet access further, if required.
4. Restart the PostgreSQL service.

Configure the standby database server

On the standby PostgreSQL database server, do the following:

1. Stop the Postgres service.
2. Delete all content from *%ProgramFiles%\PostgreSQL\9.6\data*.
3. From the command line run the following command adjusted to use the name of the primary server:
   ```
   "C:\Program Files\PostgreSQL\9.6\bin\pg_basebackup.exe" -h <primaryServer> -D "C:\Program Files\PostgreSQL\9.6\data" -U replicator -v -P -p 4432
   ```

You can ignore any warnings about copying files manually.

4. In a text editor, create a file called *recovery.conf* and place it in *%ProgramFiles%\PostgreSQL\9.6\data*.
5. Open recovery.conf and add the following text, adjusting the hostname and port:

   ```
   standby_mode = 'on'
   primary_conninfo = 'host=< primaryServer > port=4432 user=replicator password=secretpassword'
   trigger_file = 'failover'
   recovery_target_timeline = 'latest'
   ```
6. Start the PostgreSQL service.

You should now be able to connect to the database and view the data being streamed over from the primary node in read only mode.

Manual database failover

If the database on primary node is lost, a standby node needs to take over.

Do the following:

1. On the standby node that is to become the new primary node, create a file called *failover* in the folder *%ProgramFiles%\PostgreSQL\9.6\data*

   The failover file should have no file extension.

   The file triggers PostgreSQL to cease recovery and enter read/write mode. PostgreSQL also changes the name of the file *recovery.conf* to *recovery.done* to reflect the transition.
2. On each node, change the repository database connection string to point to the hostname or IP address of the new database node. As the connection string is encrypted in the config file, you need to use the **Connection String Editor** to decrypt the string, edit it, and write back an encrypted string.
   a. To open the **Connection String Editor**, navigate to *C:\Program Files\Qlik\Sense\Repository\Util\QlikSenseUtil* and open the *QlikSenseUtil.exe* file as an administrator.
   b. In the **Connection String Editor** tab, click **Read** to open the *Repository.exe* file connection string.
   The decrypted database connection string is displayed.
   c. Replace the value for **Host** with the hostname or IP address of the new database node.
   d. Click **Save value in config file encrypted** to save your changes.

AWS and Azure security

Before you deploy Qlik Sense on AWS or Azure you need to get an overview of the basic security implications. In AWS and Azure there are specific tools that you use during setup to configure permissions and to set security options. Once you have deployed Qlik Sense to your chosen cloud environment, you use the Qlik Management Console to configure security in the same way as you would in an on-premise Qlik Sense deployment.

Qlik Sense

An overview of your Qlik Sense security considerations:

- In Qlik Sense, you manage all security and authentication settings from the Qlik Management Console.
- A module in the Qlik Sense Proxy Service handles authentication of Microsoft Windows users.
- Authentication is often used in conjunction with a single sign-on (SSO) system that supplies a reverse proxy or filter for authentication of the user.
- Other authentication methods are available, and it is possible to implement your own customized solutions for different authentication scenarios.

Resources managed directly from the QMC:

- Admin roles to grant QMC users administrator level access to various sections
- Proxy certificate for communication between the web browser and the proxy component
- Virtual proxies to allow different modules based on the URI to be used to access the Qlik Sense environment
- Custom properties enabling you to use your own values in security rules
- Access control and security rules to grant users access to Qlik Sense resources

Authentication methods used by Qlik Sense:

- NTLM/Kerberos
- Security Assertion Markup Language (SAML)
- Anonymous authentication
- Session/Ticket API

For more information about Qlik Sense security, see *Security (page 71)*

AWS

To configure security in an AWS deployment you need a basic understanding of how to set up AWS security groups, key pairs, and Qlik Sense security groups. Use the Amazon Management Console to configure AWS security, and the Qlik Management Console to configure all security and authentication settings in Qlik Sense. A module in the Proxy Service (QPS) handles the authentication of Microsoft Windows users. If required, it is also possible to implement your own custom authentication solutions.

Use the Amazon Management Console to configure:

- AWS security groups - configure access rules for an initial Qlik Sense security group for your EC2 instance.
- Key pair - In the AWS console, create a Qlik Sense key pair. Save the `Qlik Sense.pem` keypair file locally, as you will need it later to access your instance.

You can use AWS Directory Services to set up security and authentication on the Qlik Sense server side. This service makes it easier to setup and run Microsoft Active Directory (AD) in the AWS cloud, or connect your AWS resources to an existing on-premises Microsoft Active Directory.

AWS Directory Service provides you with the following three directory types:

- AWS Directory Service for Microsoft Active Directory (Enterprise Edition), also referred to as Microsoft AD
- Simple AD
- AD Connector

AWS Directory Services makes it possible to connect AWS resources to an on-premises directory using the same corporate credentials. This option uses the Microsoft Security Support Provider Interface (SSPI) to read the Windows user name and password working in a similar way to single sign-on. If you have multiple nodes in the Qlik Sense Server environment, all nodes need to be part of the same domain.

For more information, see AWS security.

Azure

Use the Resource Manager to configure Azure security and the QMC to configure all security groups and authentication settings in Qlik Sense. In Azure, to configure security you first set up a subnet, a virtual network, an IP address for an instance, and network security rules. This is similar to configuring ports in a firewall. You then set up a network interface that your instance can use, and bind it to the previously set up network and subnet. A module in the Qlik Sense Proxy Service (QPS) handles the authentication of Microsoft Windows users. If required, it is also possible to implement your own custom authentication solutions.

Use the Azure Resource Manager to configure:

- Azure security groups
- Azure Active Directory and Identity Management

Azure Active Directory (Azure AD) is Microsoft’s multi-tenant cloud based directory and identity management service. For IT administrators, Azure AD provides an easy to use solution to give users single sign-on (SSO) access to other cloud SaaS Applications, such as Office365, Salesforce.com, and Concur. Azure AD also includes a full suite of identity management capabilities including multi-factor authentication, device registration, self-service password management, self-service group management, privileged account management, role based access control, application usage monitoring, rich auditing, and security monitoring and alerting.

For more information, see Azure security.

3 Qlik Sense installation

When you install Qlik Sense you have several deployment options depending on the size and requirements of your organization. Before you begin the installation process choose the appropriate architecture for your needs. Consider scalability and performance and factors such as how many apps you want to run, how many concurrent users you need, or how many reloads you want per hour.

Size of organization	Qlik Sense deployment
Small	Single-node
Medium	Single-node or multi-node
Large	Multi-node

For more information on architecture options and considerations before you install see the following:

- *Architecture (page 20)*
- *Planning your deployment (page 11)*
- *Enterprise deployment examples (page 51)*

When you are ready to proceed with the installation, choose whether to install on a single computer or not:

3.1 Installing Qlik Sense on a single node

A basic installation of Qlik Sense can be done by installing all of the Qlik Sense services on a single node. This kind of deployment works best in a single time zone, where reloads of data can be done during the night. To determine if a single-node installation is the right choice for you, see *Planning your deployment (page 11)*.

For information about multi-node deployments of Qlik Sense, see *Installing Qlik Sense in a multi-node site (page 85)*.

Before you install:

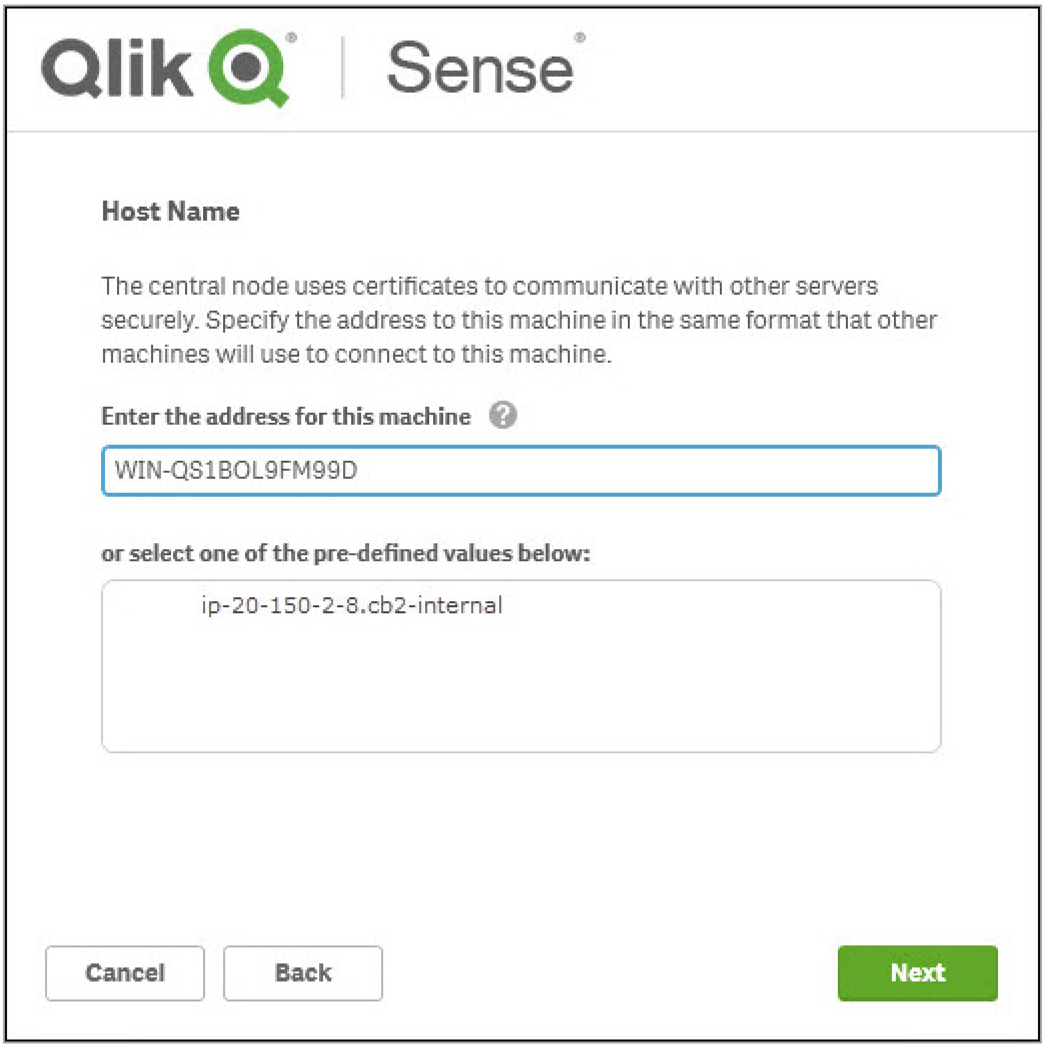
- Check that your environment meets the system requirements.
  See: *System requirements for Qlik Sense (page 12)*
- Check that the required ports are available.
  See: *Ports (page 34)*
- Check that your browser is supported.
  See: *Supported browsers (page 17)*
- Prepare the user accounts required to run the Qlik Sense services.
  See:*User accounts (page 69)*
- Understand how Qlik Sense uses LEF for site licensing and user access allocation, and have your license key available.
  See:*Licensing (page 66)*

Do the following:

1. Log in to the computer where you plan to install Qlik Sense as a local Windows administrator.
   See: *User accounts (page 69)*.
2. Create a file share before you run the Qlik Sense setup. The file share is a shared folder that stores all the Qlik Sense application data.
   a. Create a new folder.
   b. Right click on the folder, and click **Properties**.
   c. On the **Sharing** tab, and click **Share**.
   d. Enter the names of Windows users that you want to share the folder with, and click **Add**.
   Share this folder with your Windows Qlik Sense administrator and your Windows Qlik Sense services user. For more information, see *User accounts (page 69)*.
   e. In the **Permission level** column, select **Read/Write**, and click **Share**.

   > *Make note of the network path displayed on the confirmation screen. You will enter this information during the Qlik Sense setup. The network path will be in the following format: \\server-name\QlikShare*

3. Download the *Qlik_Sense_setup.exe* file from www.qlik.com, and launch the setup.
4. Do the following:
   a. Accept the license agreement, and click **Next**.
   b. On the **Create or join a cluster screen**, click **Create cluster**.
   c. On the **Host Name** screen, enter the name of the computer that you are installing Qlik Sense on and click **Next**.



Host Name

The central node uses certificates to communicate with other servers securely. Specify the address to this machine in the same format that other machines will use to connect to this machine.

Enter the address for this machine

WIN-QS1BOL9FM99D

or select one of the pre-defined values below:

ip-20-150-2-8.cb2-internal

d. On the **Shared persistence database connections settings** screen, select the **Install local database** check box if you want to install a local repository or leave the check box unchecked if you want to connect to an existing repository database.

Installing and configuring PostgreSQL (page 98)

If you want to connect to an existing repository database, then enter the following values:

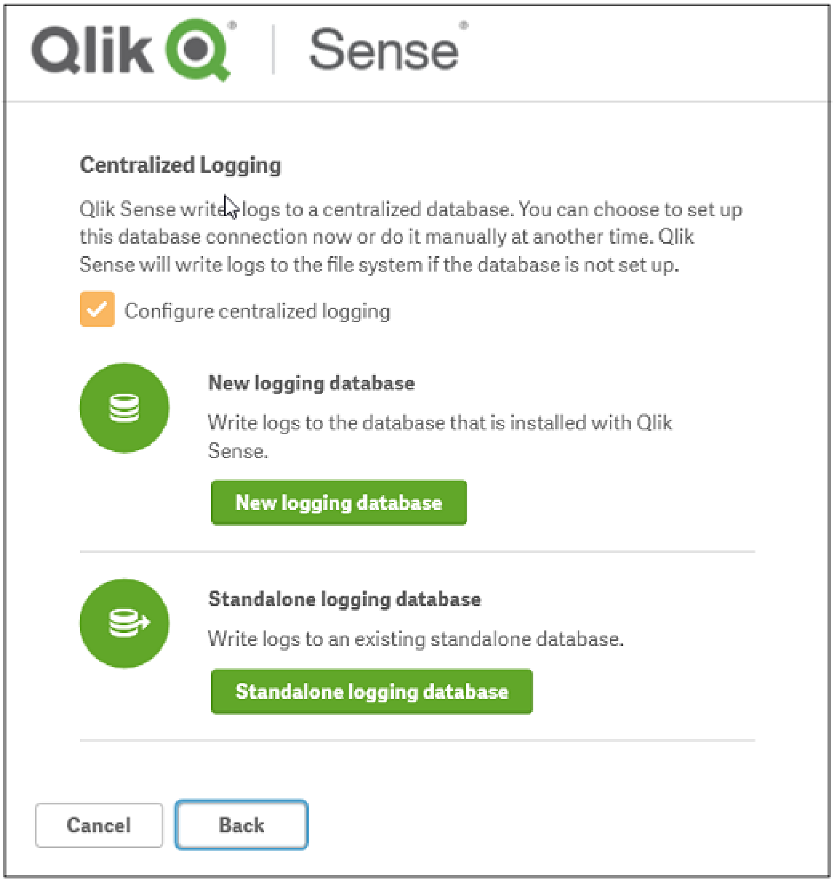
Field name	Value
Database host name	Enter the full URL to your repository database.
Database port	4432
Database user	Enter the username that will be used to access the database. ⚠ *Do not enter the username postgres.*

Database user password	Create your own database user password to access your repository database in the PostgreSQL database.

e. On the **Database configuration** screen, click **Next**.
There is no need to configure the database service to allow connections from other nodes in a single-node installation.

f. On the **Shared persistence storage** screen, enter the path or URL to your file share, for example \\<domain>\QlikShare, and click **Next**. Your file share can either be a local folder or a remote folder.

g. On the **Centralized Logging** screen, leave the **Configure centralized logging** check box selected if you want to set up centralized logging, or clear the check box if you want logs to be written to files. If you decide not to set up centralized logging at this time, you can set it up later by using the logging service utility; see *Qlik Logging Service (page 237)*.



Centralized Logging

Qlik Sense writes logs to a centralized database. You can choose to set up this database connection now or do it manually at another time. Qlik Sense will write logs to the file system if the database is not set up.

☑ Configure centralized logging

New logging database

Write logs to the database that is installed with Qlik Sense.

Standalone logging database

Write logs to an existing standalone database.

If you want to write logs to a new database that is installed with Qlik Sense, click **New logging database**. Enter the following values and click **Next**.

Field name	Value
Log writer	Create a password for the qlogs_writer user to access the centralized

password	PostgreSQL database.
Log reader password	Create a password for the qlogs_reader user to access the centralized PostgreSQL database.

If you want to write logs to an existing database on the same node that you are installing Qlik Sense on or on another node, click **Standalone logging database**. Enter the following values and click **Next**.

Field name	**Value**
Hostname	Enter the host name or IP of the centralized PostgreSQL database.
Port	4432
Log writer password	Enter the password for the qlogs_writer user to access the centralized PostgreSQL database.
Log reader password	Enter the password for the qlogs_reader user to access the centralized PostgreSQL database.

If you use a logging database on another node, ensure that this is a new and empty logging database before proceeding with the installation. If a QLogs db is already present on the remote database the schemas may be incompatible.
See: *Installing and configuring PostgreSQL (page 98)*

h. On the **Installation location** screen, choose your own installation location or install Qlik Sense to the default location on the C:\ drive, and click **Next**.

i. On the **Repository Database Superuser Password** screen, enter a password for the PostgreSQL repository database superuser. Confirm the password and click **Next**.
See, *User accounts (page 69)*.

j. On the **Service Credentials** screen, enter the domain, user name and password for the account that you want use to run the Qlik Sense services, and click **Next**.

If you enter a username that is more than 20 characters long, it must be in UPN format, and must include the full domain name. For example, longusername@full.domain.name.

k. On the **Ready to install** screen, optionally select to create desktop shortcuts and automatic start of the Qlik Sense services when the setup is complete.

*If you selected local system as the user account type in the **Service Credentials** screen, but want to use a dedicated service account to run the Qlik Sense services, clear the **Start the Qlik Sense services when the setup is complete** selection.*

   l. In the **Extension bundles** section, optionally select to install the extension bundles. Then, select which extension bundles you want to install from the list of those available for your Qlik Sense installation.

      You can always add or remove extension bundles from your Qlik Sense installation at a later moment. See: *Modifying extension bundles installation (page 84)*

   m. If you have chosen not to install the extension bundles, click **Install**. Otherwise, click **Next**.

   n. If you are installing any of the extension bundles, accept the extension bundle license agreement. Then, click **Install**.

5. You will see a message indicating that Qlik Sense has been installed successfully.

   Click **Finish**.

6. If you selected local system as the user account type in the **Service Credentials** screen, but wish to use a dedicated service account to run the Qlik Sense services change the user account type and manually start the Qlik Sense services now. See *Changing the user account to run Qlik Sense services (page 103)*

You are ready to license your Qlik Sense installation.

Licensing Qlik Sense

Before you can start using Qlik Sense you must activate your site license.

Do the following:

1. Open the QMC.
   When you open the QMC for the first time the **Site license properties** screen is displayed.
2. Enter the license information from the License Enabling File (LEF).
   The property group **Site license** contains properties related to the license for the Qlik Sense system. All fields are mandatory and must not be empty.

   > *If you want to set up Qlik Cloud Services or Qlik Sense Enterprise for elastic deployments, please contact your Qlik representative or Qlik Support to obtain a valid license for the setup.*

Property name	Description
Owner name	The user name of the Qlik Sense product owner.
Owner organization	The name of the organization that the Qlik Sense product owner is a member of.
Serial number	The serial number assigned to the Qlik Sense software.
Control number	The control number assigned to the Qlik Sense software.

3. Expand **LEF access** and click **Get LEF and preview the license**. If you received your LEF via email, you can copy and paste the information into the text field.

__Failed to get LEF from server__ is displayed if the serial number or control number is incorrect.

4. Click **Apply**.
   Successfully licensed is displayed.
5. Click **Close**.

You have activated your Qlik Sense site license.

You are ready to connect to a user directory (optional), allocate user access or professional access, and set up permissions.

Allocating user access or professional access

Your license is either user-based, with professional access allocation as an option, or token-based, with user access allocation as an option.

User-based license

Your Qlik Sense license includes a number of professional access allocations that are used to grant users in your organization access to Qlik Sense.

Do the following:

1. In the QMC, from the **Start** menu, click **License management**.
   The **License usage summary** screen is displayed.
2. Click the **Professional access allocations** tab.
3. Click the **+ Allocate** button.
   The **Users** screen is displayed.
4. Select the users that you want to provide access to from the list and click **Allocate**.

__Allocate__ is disabled if the number of available allocations is insufficient for the number of selected users.

The users that you allocated access to appear in the **Professional access allocations** overview table.

Token-based license

Your Qlik Sense license includes a number of tokens that are used to allocate Qlik Sense access to users in your organization.

Do the following:

1. In the QMC, from the **Start** menu, click **License management**.
   The **License usage summary** screen is displayed.
2. Click the **User access allocations** tab.

3. Click the **+ Allocate** button.
   The **Users** screen is displayed.
4. Select the users that you want to provide access to from the list and click **Allocate**.

> ***Allocate*** *is disabled if the number of tokens available for allocation is insufficient for the number of selected users.*

The users that you allocated access to appear in the **User access allocations** overview table.

Additional configuration

After you install Qlik Sense, you may wan to:

- Create load balancing rules in the QMC to improve resilience and performance in a multi-node site. For more information, see Load balancing.
- Configure the virtual proxy advanced settings to add your own hosts names to the white list. For more information, see Host white list.
- Configure the user directory connector to retrieve users from a user directory. For more information, see User imports (UDC).

You are now ready to start using Qlik Sense. See: Get started.

Modifying extension bundles installation

You can add or remove extension bundles from your Qlik Sense deployment at any moment. If you have a multi-node installation, extension bundles are installed on the central node.

> *You can see which extensions are installed in your deployment by checking the* ***Extensions*** *section in the Qlik Management Console (QMC).*

Do the following:

1. In **Control Panel**, open **Programs and Features**.
2. In the list of programs, double-click the extension bundle that you want to modify.
3. The Extension Bundle Setup Wizard opens. Click **Next**.
4. Select **Change**.
5. On the **Custom setup** screen, click on the bundle icon to select how to modify the bundle installation:
   - If the bundle is installed, select **Entire feature will be unavailable** to uninstall it.
   - If the bundle is not installed, select **Entire feature will be installed on local hard drive** to install it.

   Then, click **Next**.
6. Click **Change**.

7. When the setup modification is complete, a message invites you to manually restart the Qlik Sense Repository Service.
8. Click **Finish** to close the Extension Bundle Setup Wizard.
9. Manually restart the Qlik Sense Repository Service to make the changes effective.

You can verify that the changes have been correctly applied by checking the **Extensions** section in the QMC.

3.2 Installing Qlik Sense in a multi-node site

A Qlik Sense multi-node deployment offers more configuration options than single node deployments. In a multi-node site, you can distribute Qlik Sense services across one or more server nodes to optimize scalability and performance.

Preparing a large, enterprise multi-node deployment requires careful planning, so first ensure that you have considered all the architecture and configuration options available.

For more information about single-node deployments of Qlik Sense, see *Installing Qlik Sense on a single node (page 77)*.

For more information on multi-node architecture and configuration options see:

- *Planning your deployment (page 11)*
- *Architecture (page 20)*
- *Security (page 71)*
- *Performance (page 68)*

Before you install:

- Check that your planned environment meets the system requirements.
  See: *System requirements for Qlik Sense (page 12)*
- Prepare the user accounts required to run the Qlik Sense services on the computer where you plan to install Qlik Sense.
  See: *User accounts (page 69)*

  > *Important: If during the installation, you want to run the Qlik Sense services as a local user, without administrator rights, then you must create this user first.*

- Ensure that your firewall is enabled and you have created the appropriate rules to allow rim nodes to communicate with the central node.

  See: *Ports (page 34)* for a full list of ports.
- Repository database - if you already have a Qlik Sense repository database on another server from a previous installation, you can continue to use this in your new deployment. If you do not intend to use this database then remove it before you start.
- Create a local file share to store your Qlik Sense application data.
  See: *Creating a file share (page 97)*

- Understand how Qlik Sense uses the License Enabler File (LEF) for licensing, and have your license key available.
  See: *Licensing (page 66)*

This topic includes the following sections:

- *Installing Qlik Sense (page 86)*
- *Adding a Qlik Sense node (page 94)*

Installing Qlik Sense

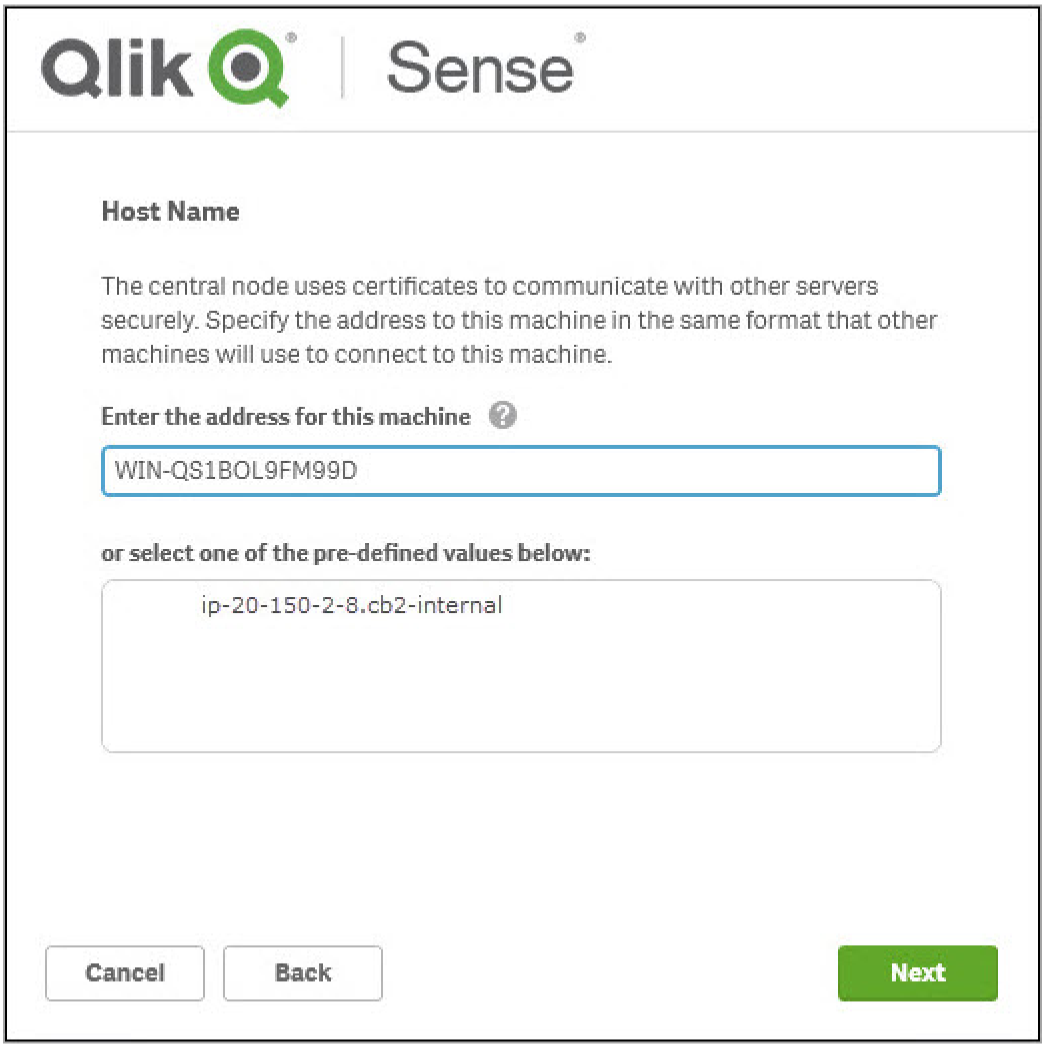
You can install a Qlik Sense server as either a central node or as a rim node. In a multi-server site, rim nodes must be connected to a central node. See: *Architecture (page 20)*. If you are installing a central node, you may also wish to configure a failover candidate. You only have the option to create a failover candidate when you are creating a node. For more information on how to configure a failover candidate, see Creating a node and Service cluster.

To install a node:

1. Create a file share before you run the Qlik Sense setup. The file share is a shared folder that stores all the Qlik application data and must be accessible to all nodes in your Qlik Sense site. You can create a file share either on the same server computer as the central node or on another server.
   See: *Creating a file share (page 97)*
2. Log in to the computer where you plan to install Qlik Sense as a domain or local Windows administrator. You must have full administrator rights to run the Qlik Sense setup. You can start the Qlik Sense services as either an administrator or a local user without administrator privileges.
   See: *User accounts (page 69)*.
3. Download the *Qlik_Sense_setup.exe* file from www.qlik.com.
4. Run the installation program as an administrator, and on the first screen click **Install**.
5. Read the **License agreement** screen. If you agree, select the check box and click **Next**.
6. On the **Create or join a cluster** screen, you have two options:
   - **Create cluster** - To install a central node. All the other nodes in your site will connect to this node.
   - **Join cluster** - To install a rim node that connects to a central node (if you choose this option, fewer screens are displayed in the setup).
7. On the **Host Name** screen, enter the address for the Qlik Sense node that you are installing, and click **Next**. The address must be in a format that other nodes can use when connecting to this node, otherwise the connection will fail.

   For example:

   - IP address: 10.1.123.234
   - Machine name: WIN-QS1BOL9FM99D
   - Fully qualified machine name: WIN-QS1BOL9FM99D.CUSTOMER.COM



Host Name

The central node uses certificates to communicate with other servers securely. Specify the address to this machine in the same format that other machines will use to connect to this machine.

Enter the address for this machine

WIN-QS1BOL9FM99D

or select one of the pre-defined values below:

ip-20-150-2-8.cb2-internal

*Ensure that the recommended server node name displayed in the **Enter the address for this machine** field matches the one you will use to access this node, otherwise enter an appropriate address or fully qualified domain name. Only use the fully qualified name if you understand the full implications.*

8. On the **Shared persistence database connections settings** screen, select the **Install local database** check box if you want to install a local repository database, or leave the check box unchecked if you want to connect to an existing repository database hosted on another server.
   Installing and configuring PostgreSQL (page 98)
   If you want to install a local repository database, then enter the following values:

Field name	**Value**
Database host name	*localhost*
Database port	*4432*

Field name	**Value**
Database user	*qliksenserepository* ⚠ *Do not enter the username postgres.*
Database user password	Create a password to access the local repository database.

If you want to connect to an existing repository database on another server, then enter the following values:

Field name	**Value**
Database host name	Enter the full URL to your repository database.
Database port	*4432*
Database user	*qliksenserepository* This is the login role you created in the PostgreSQL database (QSR)
Database user password	Enter the password you created in PostgreSQL.

Make a note of these values as you will need them again when you install a rim node.

> *All Qlik Sense servers must be in the same geographic location or data center as the repository database and the file share.*

9. On the **Database configuration** screen, under **Advanced settings**, configure the listen addresses, IP ranges, and max connections from other nodes, and click **Next**. This is an optional step if you install a local repository database. You can also configure the database service listener directly in your PostgreSQL repository database. See: *Installing and configuring PostgreSQL (page 98)*
Enter the following values:

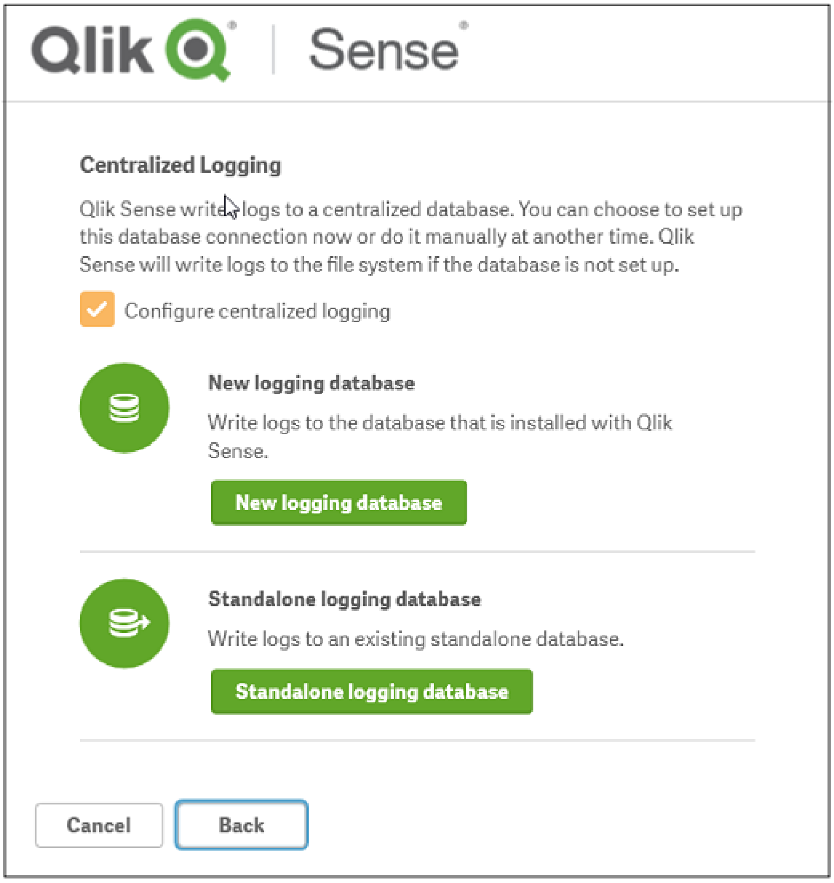
Field name	**Value**	**Description**
Listen addresses	*	The IP address(es) to listen on. Use the value * to allow access for all IP addresses. If entering multiple listen addresses use a comma separated list.
IP ranges	`0.0.0.0/0,::/0`	To allow all servers to access the repository database, use the value `0.0.0.0/0` (for all IPv4 addresses) and `::/0` (for all IPv6 addresses). If entering multiple IP addresses use a comma separated list.
Max connections	100	Specify the maximum number of concurrent connections to the database. The default value for a single server is 100. Multiply this value by the number of nodes in the cluster.

> *This screen does not appear if you are using a remote PostgreSQL database or if your are installing a rim node (**Join cluster** option).*

10. On the **Shared persistence storage** screen, enter the path or URL to your file share, for example \\*<domain>**QlikShare* and click **Next**. Your file share can either be a local folder or a remote folder on another server.
Creating a file share (page 97)

> *This screen does not appear if you are installing a rim node (**Join cluster** option).*

11. On the **Centralized Logging** screen, leave the **Configure centralized logging** check box selected if you want to set up centralized logging, or clear the check box if you want logs to be written to files. If you decide not to set up centralized logging at this time, you can set it up later by using the logging service utility; see *Qlik Logging Service (page 237)*.



Centralized Logging

Qlik Sense writes logs to a centralized database. You can choose to set up this database connection now or do it manually at another time. Qlik Sense will write logs to the file system if the database is not set up.

☑ Configure centralized logging

New logging database

Write logs to the database that is installed with Qlik Sense.

Standalone logging database

Write logs to an existing standalone database.

If you are installing Qlik Sense on a central node and you want to write logs to the database that is installed with Qlik Sense, click **New logging database**. Enter the following values and click **Next**.

Field name	Value
Log writer password	Create a password for the qlogs_writer user to access the centralized PostgreSQL database.
Log reader password	Create a password for the qlogs_reader user to access the centralized PostgreSQL database.

Do not use mixed character sets when creating a password.

If you want to write logs to an existing database on another node, click **Standalone logging database**. Enter the following values and click **Next**.

Field name	Value
Hostname	Enter the hostname or IP address of the standalone logging database.
Port	Enter the port number that you specified for the standalone logging database.
Log writer password	Enter the password for the qlogs_writer user.
Log reader password	Enter the password for the qlogs_reader user.

If you use a logging database on another node, ensure that this is a new and empty logging database before proceeding with the installation. If a QLogs db is already present on the remote database the schemas may be incompatible.
See: *Installing and configuring PostgreSQL (page 98)*

12. On the **Installation location** screen, choose a location to install Qlik Sense or use the default location on the *C:\* drive, and click **Next**.
13. On the **Repository Database Superuser Password** screen, create a superuser password for the PostgreSQL database, and click **Next**.

    *This screen does not appear if you are using a remote PostgreSQL database or if your are installing a rim node (**Join cluster** option).*

14. On the **Service Credentials** page, enter the domain, user name and password for the account that you want use to run the Qlik Sense services, and click **Next**.
    User accounts (page 69)

    If you enter a username that is more than 20 characters long, it must be in UPN format, and must include the full domain name. For example, longusername@full.domain.name.

15. On the **Ready to install** screen, optionally select to create desktop shortcuts and automatic start of the Qlik Sense services when the setup is complete.

> ℹ *If you want to use a dedicated service account to run the Qlik Sense services, clear the* ***Start the Qlik Sense services when the setup is complete*** *selection.*

16. In the **Extension bundles** section, optionally select to install the extension bundles. Then, select which extension bundles you want to install from the list of those available for your Qlik Sense installation.

    You can always add or remove extension bundles from your Qlik Sense installation at a later moment. See: *Modifying extension bundles installation (page 96)*.

17. If you have chosen not to install the extension bundles, click **Install**. Otherwise, click **Next**.
18. If you are installing any of the extension bundles, accept the extension bundle license agreement. Then, click **Install**.
19. You will see a message indicating that Qlik Sense has been installed successfully.

    Click **Finish**.

20. If you selected local system as the user account type in the **Service Credentials** screen, but wish to use a dedicated service account to run the Qlik Sense services, change the user account type and manually start the Qlik Sense services now. See *User accounts (page 69)*

Configuring PostgreSQL multi-node connections

In the *postgresql.conf* configuration file, you need to edit the `max_connections` setting, depending on how many nodes you require in your site. If you do not configure this setting correctly, and reach the connection pool limit, then PostgreSQL will reject any further connections.

To configure the `max_connections` setting:

1. Stop the Qlik Sense services.
2. Navigate to the *postgresql.conf* file in *C:\ProgramData\Qlik\Sense\Repository\PostgreSQL\<version>* of your Qlik Sense installation.
3. To edit this setting, open the file in a text editor as an administrator.
4. Make the following configuration changes:

Setting	Value	Description
`max_connections`	600	To calculate this value multiply x 100 the number of servers in your deployment. For example, 600 = 6 nodes.

5. Save your changes.
6. Start the Qlik Sense services.

You are ready to license your Qlik Sense installation.

Licensing Qlik Sense

Before you can start using Qlik Sense you must activate your site license.

To activate your license:

1. Open the QMC.
2. When you open the QMC for the first time the **Site license properties** page is displayed.
3. Enter the license information from the *License Enabler File (page 295)* (LEF).
   The property group **Site license** contains properties related to the license for the Qlik Sense system. All fields are mandatory and must not be empty.

   > *If you want to set up Qlik Cloud Services or Qlik Sense Enterprise for elastic deployments, please contact your Qlik representative or Qlik Support to obtain a valid license for the setup.*

Property name	Description
Owner name	The user name of the Qlik Sense product owner.
Owner organization	The name of the organization that the Qlik Sense product owner is a member of.
Serial number	The serial number assigned to the Qlik Sense software.
Control number	The control number assigned to the Qlik Sense software.

4. Expand **LEF access** and click **Get LEF and preview the license**. If you received your LEF via email, you can copy and paste the information into the text field.

   > ***Failed to get LEF from server** is displayed if the serial number or control number is incorrect.*

5. Click **Apply**.
   Successfully licensed is displayed.
6. Click **Close**.

You have activated your Qlik Sense site license.

You are ready to connect to a user directory (optional), allocate user access or professional access, and set up permissions.

Allocating user access or professional access

Your license is either user-based, with professional access allocation as an option, or token-based, with user access allocation as an option.

User-based license

Your Qlik Sense license includes a number of professional access allocations that are used to grant users in your organization access to Qlik Sense.

Do the following:

1. In the QMC, from the **Start** menu, click **License management**.
   The **License usage summary** screen is displayed.
2. Click the **Professional access allocations** tab.
3. Click the **+ Allocate** button.
   The **Users** screen is displayed.
4. Select the users that you want to provide access to from the list and click **Allocate**.

   > ***Allocate*** *is disabled if the number of allocations available is insufficient for the number of selected users.*

The users that you allocated access to appear in the **Professional access allocations** overview table.

> *In a multi-node site, all nodes share the same license, so you only need to activate your license once on the central node.*

If you have created a rim node, you are now ready to register the rim node with the central node.

Token-based license

Your Qlik Sense license includes a number of tokens that are used to allocate Qlik Sense access to users in your organization.

Do the following:

1. In the QMC, from the **Start** menu, click **License management**.
   The **License usage summary** screen is displayed.
2. Click the **User access allocations** tab.
3. Click the **+ Allocate** button.
   The **Users** screen is displayed.
4. Select the users that you want to provide access to from the list and click **Allocate**.

   > ***Allocate*** *is disabled if the number of tokens available for allocation is insufficient for the number of selected users.*

The users that you allocated access to appear in the **User access allocations** overview table.

> *In a multi-node site, all nodes share the same license, so you only need to activate your license once on the central node.*

If you have created a rim node, you are now ready to register the rim node with the central node.

Adding a Qlik Sense node

After installing a central node and a rim node, configure the central node to connect to the rim node. Before you can verify that a rim node is running correctly you must connect it to the central node. Use the QMC on the central node to register a rim node.

To configure a central node to connect to a rim node:

1. On the central node, open the QMC, and click **Nodes**.
2. Click **Create new**.
3. In the **Edit node** window, enter the following configuration details about the node you want to connect to:

Field name	**Description**	**Example value**
Name	Provide a suitable name for the node.	For example, *Consumer node 1*
Host name	Enter the full URL of the node you want to connect to.	For example, *<domain>-<server-name>.qliktech.com*
Node purpose	Choose a suitable purpose for the node: - Production - Development - Both	For example, choose **Production** for a scheduler node or **Development** for a developer node used for creating apps. For more information on types of nodes, see: Creating a node Check that your license supports the node purpose that you have chosen.
Node configuration	Select this node as a failover candidate.	For example, if you select this node as a failover candidate it means that this node can perform the same role as the central node if the central node fails. See: *Failover (page*

		97)
Service activation	Select the services you want to run on this server node: • Repository • Engine • Printing • Proxy • Scheduler	For example, if you are installing a consumer node, select the **Repository** and **Engine** services. For more information on which services to run on different types of nodes, see: *Architecture (page 20)* and *Services (page 22)*

4. Click **Apply**. The central node generates a certificate that you use to register the rim node. If the central node cannot connect to the rim node you will see a **Node registration** error message. If you get this error, first check that you have opened port 4444 on the central and rim nodes to allow certificates to be sent.
5. The **Install certificates** pop-up window then opens providing you with a URL and a password to authorize the certificate on the rim node.
6. On the rim node, paste the URL into a new browser window.
7. On the **Install certificates** page (in your browser), enter the password and click **Submit**. If successful, you see the **Successfully licensed** message.
8. Follow the same authorization procedure for each node that you want to add to your deployment.
9. To verify that all rim nodes are configured correctly, open the QMC, click **Nodes** and you can see the status of all the nodes in your deployment.

Verify your installation

To verify that Qlik Sense has installed correctly:

1. Open the Qlik Management Console (QMC).
2. Open the Qlik Sense Hub.

If the QMC and Hub open without any security warnings displayed in the browser, then you have installed Qlik Sense correctly.

Additional configuration

After you have installed and verified that Qlik Sense is running correctly, you may find the following configuration information useful:

- Load balancing - create load balancing rules in the QMC to improve resilience and performance in a multi-node site.
- Host white list - configure the virtual proxy advanced settings to add your own hosts names to the white list.
- User imports (UDC) - configure the user directory connector to retrieve users from a user directory .

> *If you are installing custom connectors in a multi-node setup, the custom connectors must be installed on each node.*

You are now ready to start using Qlik Sense.

Get started.

Modifying extension bundles installation

You can add or remove extension bundles from your Qlik Sense deployment at any moment. If you have a multi-node installation, extension bundles are installed on the central node.

> *You can see which extensions are installed in your deployment by checking the* ***Extensions*** *section in the Qlik Management Console (QMC).*

Do the following:

1. In **Control Panel**, open **Programs and Features**.
2. In the list of programs, double-click the extension bundle that you want to modify.
3. The Extension Bundle Setup Wizard opens. Click **Next**.
4. Select **Change**.
5. On the **Custom setup** screen, click on the bundle icon to select how to modify the bundle installation:
   - If the bundle is installed, select **Entire feature will be unavailable** to uninstall it.
   - If the bundle is not installed, select **Entire feature will be installed on local hard drive** to install it.

   Then, click **Next**.
6. Click **Change**.
7. When the setup modification is complete, a message invites you to manually restart the Qlik Sense Repository Service.
8. Click **Finish** to close the Extension Bundle Setup Wizard.
9. Manually restart the Qlik Sense Repository Service to make the changes effective.

You can verify that the changes have been correctly applied by checking the **Extensions** section in the QMC.

3.3 Creating a file share

Creating a file share or shared folder is a necessary prerequisite before you install Qlik Sense. The file share is used to store all the Qlik application data and must be accessible to all nodes in your Qlik Sense site. You can create a file share either on the same server as the central node or on a separate server. If you have a large multi-node site we recommend that you configure the file share on a dedicated server for better resilience and performance.

If you create the file share on a separate server then you can follow the same steps as for a central node but you must ensure that the same Windows domain user that you use to run the Qlik services has read and write access to the file share folder.

To create a file share and share the folder with specific users:

1. Create a local folder on your server computer. For example, create a folder called *QlikShare* on the *C:\* drive.
2. Right click the folder, and then click **Properties**.
3. Click the **Sharing** tab, and then click **Share**.
4. Enter the name of your Windows user, and click **Add**.
5. In the **Permission level** column, select **Read/Write**, then click **Share**.

> *Make a note of the network path shown in the confirmation screen as you use this later during setup of your shared persistence storage folders. The network path will be in the following format: \\server-name\QlikShare*

Ensure that permissions on the folder, subfolders, and files are set to full control for the user account you selected.

To do this:

1. Click the **Security** tab.
2. Select the user account you want to use for the installation.
3. Click **Advanced** and check that your user has full control and that this permission applies to the folder, subfolders, and files.
4. Click the **Effective Access** tab and then click **Select a user** and enter your user account name.
5. Click **View effective access** and check in the **Permission** column that your user has full control.

3.4 Failover

To avoid having a single point of failure in a multi-node site, when you add a new node to your deployment you can assign it the role of failover candidate. This means that any server or node in your Qlik Sense site can perform the same role as the central node. The role of the central node can now be swapped, for example if the central node has been offline for more than 10 minutes.

Automatic failover

After you have configured a node to become a failover candidate, each node in your site will regularly check the primary node (central node) for a heartbeat. If there is no communication between the primary node and the other nodes in the site after 10 minutes then the primary node will be replaced by the next available node. If more than one node is set as a failover candidate each node will compete to get a lock on a database field and the winner becomes the central node. There is an additional field in the QMC to show which node is currently the central node.

Manually migrating the central node

If you decide that you want to move the central node to another node in your site, you can manually migrate it using the the following REST API calls:

- Get `/qrs/serverNodeConfiguration` to get a list of server GUIDs.
- Do an empty POST to `/qrs/failover/tonode/{serverNodeConfigurationID}` to retrieve the ID of the node you want to migrate to.

3.5 Installing and configuring PostgreSQL

To improve performance in a Qlik Sense multi-node deployment, you have the option to install your repository (QSR), SenseServices, QSMQ, and (QLogs) logging databases on a dedicated, remote PostgreSQL server.

In Qlik Sense Enterprise, configuring all the components of a Multi-Cloud deployment is optional. However, all deployments, whether Multi-Cloud or on-premise require the installation of the SenseServices database and QSMQ databases.

The Qlik Sense repository database (QSR)

The QSR is the primary database in your Qlik Sense deployment.

If you want to install the QSR database on a dedicated PostgreSQL server, you must install and configure PostgreSQL before you install Qlik Sense, as you will need to enter the PostgreSQL server/host details in the Qlik Sense installer.

The Qlik Sense services database (SenseServices)

The SenseServices database contains schemas for each of the Qlik Sense services and allows growth independently of the Qlik Sense Repository Database, while still sharing the same PostgreSQL instance and login role.

The Qlik Sense message queue database (QSMQ)

The QSMQ database provides a light-weight method of passing messages internally between services in Qlik Sense Enterprise. The NOTIFY and LISTEN functionality in PostgreSQL allows services to be notified about new messages that have been written to the messaging table.

The QSR, SenseServices and QSMQ databases share the same login role and must be installed on the same PostgreSQL instance.

The Qlik Sense logging database (QLogs)

The QLogs database centralizes logging by collecting log messages from all Qlik Sense nodes in your deployment and stores them in a PostgreSQL database.

When you install the QLogs database as a standalone logging database, you can configure it either before or after you install Qlik Sense.

- If you install the QLogs database before the Qlik Sense installation, then you will need to create the QLogs database login roles manually.
- If you install the QLogs database after installing Qlik Sense, use the `Qlik.Logging.Service.exe setup` command. When you run this command, you specify a remote host and the script will automatically create the QLogs database and login roles for you. For more information, see: *Qlik Logging Service (page 237)*

If you already have a PostgreSQL database installed as part of a previous deployment, then you can continue to use it.

If Qlik Sense uses a PostgreSQL database on a dedicated infrastructure, then it can use PostgreSQL version 9.6. You can run the instance of PostgreSQL on platforms including Windows, Linux or cloud hosted services, such as Amazon RDS. However, Qlik will only offer configuration support when PostgreSQL is running on Windows. If you use Linux or Amazon RDS, it is your own responsibility to install and configure a running instance of PostgreSQL for Qlik Sense to use.

To install a dedicated PostgreSQL server with QSR, SenseServices, QSMQ and QLogs database:

- Install PostgreSQL
- Create the PostgreSQL databases, and configure login roles.
- Edit the configuration files to allow access from Qlik Sense nodes.
- Verify that the database has installed and is running correctly.

Installing PostgreSQL

Before installing a dedicated PostgreSQL server instance, check that your server fulfills the system requirements on www.postgresql.org.

To install PostgreSQL on a dedicated server:

1. Log in to the server where you want to install PostgreSQL as an administrator.
   See: *User accounts (page 69)*
2. Download PostgreSQL EnterpriseDB version 9.6 from the PostgreSQL website.

3. Run the **PostgreSQL setup wizard**.
4. On the **Installation Directory** and **Data Directory** screens, accept the default paths.
5. On the **Password** screen, create a password for the PostgreSQL superuser.
   You will use this password when you connect to the PostgreSQL database and you will also be prompted for it when you run the Qlik Sense setup.
6. On the **Port** screen, specify port 4432. This port is required for communication between all the nodes in a site.
7. In the **Advanced Options** screen, accept the default locale.
8. In the **Ready to Install** screen, click **Next** to run the setup.
9. After running the setup, you have the option to install *Stack Builder*. Clear the check box if you want to install this later.
10. Click **Finish** to complete the installation.

When you install PostgreSQL EnterpriseDB, the pgAdmin tool is included.

Creating a PostgreSQL database

You can create a repository QSR, SenseServices, QSMQ, and QLogs (logging) database manually with the pgAdmin tool or using a script.

To create a new, empty PostgreSQL database using the pgAdmin tool:

1. Open the *pgAdmin* tool.
2. In the *pgAdmin* **Browser**, under **Servers**, right-click the PostgreSQL node and then click **Connect Server**.
3. Enter your PostgreSQL superuser password to make a connection. A green status bar appears in the lower right corner of your screen when the server connection is successful.
4. Right-click the **Databases** node, click **Create**, and then click **Database**.
5. Enter the name of the database you are creating, and then click **Save**.

To create a new, empty PostgreSQL database by running a script in the pgAdmin tool:

1. Open the **Query Tool**. First select an existing database, such as **postgres**, to display the **Query Tool** option in the **Tools** menu.
2. Execute the following script:
   `CREATE DATABASE "<databasename>" ENCODING = 'UTF8'; --creates an empty database.`
   Replace `<databasename>` with `QSR` for the repository database, `SenseServices` for the SenseServices database, `QSMQ` for the message queue database, and `QLogs` if you are creating a logging database.

Creating login roles

You need to create login roles for users when you create a PostgreSQL database. You can create login roles using the pgAdmin tool or by running a script.

The QSR, SenseServices, and QSMQ login role

To create login roles using the pgAdmin tool:

1. Right-click the **Login/Group Roles** node. To create a new database user, click **Create**, and then click **Login/Group Role**.
2. In the **Create - Login/Group Role** window, in the **General** tab, enter the name *qliksenserepository*.
3. In the **Privileges** tab, enable **Can login?** and leave the other default privileges unchanged.
4. In the **Definition** tab, enter a password of your choice, and click **Save**.
   When you run the Qlik Sense setup, in the **Shared persistence database connections settings** screen, you are asked to enter the **Database user** password that you created here so that Qlik Sense can connect to the repository database.
5. Make *qliksenserepository* the owner of the **QSR, SenseServices**, and **QSMQ** databases. To do this, right-click the **QSR, SenseServices**, or **QSMQ** databases you created earlier, and then click **Properties**.
6. In the **General** tab, in the **Owner** drop-down, select *qliksenserepository* as **Owner** of the **QSR, SenseServices**, or **QSMQ** databases and click **Save**.

To create login roles by running a script in the pgAdmin tool:

1. Open the **Query Tool**. Select an existing database, to display the **Query Tool** option in the **Tools** menu.
2. Run the following script:

```
CREATE ROLE qliksenserepository WITH LOGIN NOINHERIT NOSUPERUSER NOCREATEDB NOCREATEROLE
NOREPLICATION VALID UNTIL 'infinity'; --creates 'qliksenserepository' user and  assigns
privileges
ALTER ROLE qliksenserepository WITH ENCRYPTED PASSWORD '<qliksenserepository_password>'; --
assigns password to qliksenserepository
ALTER DATABASE "QSR" OWNER TO qliksenserepository; --sets qliksenserepository as owner of the
QSR database
ALTER DATABASE "SenseServices" OWNER TO qliksenserepository; --sets qliksenserepository as
owner of the SenseServices database
ALTER DATABASE "QSMQ" OWNER TO qliksenserepository; --sets qliksenserepository as owner of the
QSMQ database
GRANT TEMPORARY, CONNECT ON DATABASE "QSMQ" TO PUBLIC;
GRANT ALL ON DATABASE "QSMQ" TO postgres;
GRANT CREATE ON DATABASE "QSMQ" TO qliksenserepository;
GRANT TEMPORARY, CONNECT ON DATABASE "SenseServices" TO PUBLIC;
GRANT ALL ON DATABASE "SenseServices" TO postgres;
GRANT CREATE ON DATABASE "SenseServices" TO qliksenserepository;
```

Include a password for `qliksenserepository` *as you will be prompted for this when you install Qlik Sense.*

The QLogs login role

To create login roles for the QLogs database by running a script in the pgAdmin tool:

1. Open the **Query Tool**. Select an existing database, to display the **Query Tool** option in the **Tools** menu.
2. Run the following script:

```
CREATE ROLE qlogs_users WITH NOLOGIN NOINHERIT NOSUPERUSER NOCREATEDB NOCREATEROLE
NOREPLICATION VALID UNTIL 'infinity';
                                CREATE ROLE qlogs_reader WITH LOGIN NOINHERIT NOSUPERUSER NOCREATEDB NOCREATER
```

```
NOREPLICATION VALID UNTIL 'infinity';
                                CREATE ROLE qlogs_writer WITH LOGIN NOINHERIT NOSUPERUSER NOCREATEDB NOCREATER
NOREPLICATION VALID UNTIL 'infinity'; --creates users and assigns privileges
                                ALTER ROLE qlogs_reader WITH ENCRYPTED PASSWORD '<qlogs_reader_password>'; --a
password to qlogs_reader

                                ALTER ROLE qlogs_writer WITH ENCRYPTED PASSWORD '<qlogs_writer_password>'; --a
password to qlogs_writer

                                GRANT qlogs_users TO qlogs_reader;
                                GRANT qlogs_users TO qlogs_writer; --adds qlogs_reader and qlogs_writer to qlo
group

                                ALTER DATABASE "QLogs" OWNER TO qlogs_writer; --sets qlogs_writer as an owner
database
```

Include a password for `qlogs_reader` and `qlogs_writer` as you will be prompted for these when you install Qlik Sense.

Configuring PostgreSQL

To allow communication between your PostgreSQL repository database and your Qlik Sense nodes, edit the `pga_hba.conf` and `postgresql.conf` configuration files.

Make a backup copy of the `postgresql.conf` and `pg_hba.conf` files before you start, so that you have the option to revert back to the original settings.

postgresql.conf

The `postgresql.conf` file enables you to specify general parameters for your PostgreSQL server, such as for auditing, authentication, and encryption. Edit this file to control which Qlik Sense nodes can access your PostgreSQL database server.

To edit the *postgresql.conf* file:

1. Navigate to the *postgresql.conf* file in *C:\Program Files\PostgreSQL\<version>\data* of your PostgreSQL installation.
2. Open the file in a text editor as an administrator.
3. Make the following configuration changes:

Setting	Value	Description
`listen_addresses`	*	Enter the IP address(es) to listen on. If entering multiple listen addresses, use a comma separated list. Enter * to listen for connections from all IP addresses.
`max_connections`	600	Defines the maximum number of client connections allowed. To calculate this value, multiply by 100 the number of nodes in your deployment.

4. Save your changes.

For more detailed information about setting these parameters, see the PostgreSQL documentation.

pg_hba.conf

The `pg_hba.conf` file handles client authentication. Each record specifies a connection type, such as a client IP address range, database name, user name, and the authentication method used.

To edit the *pg_hba.conf* file:

1. Navigate to the *pg_hba.conf* file in *C:\Program Files\PostgreSQL\<version>\data* of your PostgreSQL installation.
2. Open the file in a text editor as an administrator.
3. Locate the following line:
   ```
   host     all     all     127.0.0.1/32     md5
   ```
   This line determines which servers can access the repository database server. The default address setting, `127.0.0.1/32`, only allows local host to access the database.
4. Replace `127.0.0.1/32` with a sub net specification that covers all the IP addresses of the nodes in your site.
   When specifying these settings, add one row for each node, using `/32` as a suffix for each address, or add a sub net that covers all addresses using, for example, `/24` as a suffix:
   - IPv4 (32-bit addresses):
     - To specify a single address: `192.168.1.0/24`, or `172.20.143.89/32`
     - For a small network: `172.20.143.0/24`, or `10.6.0.0/16` for a larger one.
     - To allow access from all IPv4 addresses: `0.0.0.0/0`
   - IPv6 (128-bit numeric addresses):
     - For a single host: `::1/128` (in this case the IPv6 loopback address)
     - For a small network: `fe80::7a31:c1ff:0000:0000/96`
     - To allow access from all IPv6 addresses: `::/0`

   > *When you add the IPv6 connections and use hostname in the address column, both the forward and reverse `nslookup` of the client machine must return valid values for PostgreSQL to accept the connection from the client. For more information refer to the PostgreSQL documentation.*

5. Save your changes.

For more information on how to set a more restrictive IP address, see the PostgreSQL documentation.

You have installed and configured a PostgreSQL database on a separate server. You are now ready to resume your installation of Qlik Sense.

3.6 Changing the user account to run Qlik Sense services

Before you install, change or upgrade your Qlik Sense installation, you must choose an administrator or non-administrator account to run the Qlik Sense services. For example your company policy may require you to run the Qlik Sense services as a user without administrator privileges.

If you want to upgrade from Qlik Sense 3.1 SR2 or later to Qlik Sense June 2017 you must use a service user account (local or domain) and not a Local System account to run the services. If you use a Local System account to upgrade, you will get an error.
See: Upgrading from Qlik Sense 3.1 SR2 or later to Qlik Sense June 2017 or later (page 166)

Using an account without administrator privileges to run the Qlik Sense services during the installation of a node

To install a rim node in this way you need to run an additional bootstrap command from an elevated command prompt to register the rim node on the central node.

If you are installing a central node you can follow the same procedure as a regular administrator installation.

To install a node:

1. Log in to the computer where you plan to install Qlik Sense as an administrator.
   See: *User accounts (page 69)*.
2. Download the *Qlik_Sense_setup.exe* file from www.qlik.com
3. On the **Create or join a cluster** screen, select **Join cluster**.
4. On the **Shared persistence database connections settings** screen, ensure that you specify the correct hostname and password to the repository database that you want to connect to.

   See: *Installing Qlik Sense (page 86)*
5. On the **Service Credentials** screen, enter your non-administrator user account, user name, and password. For example, enter your user name as follows: *.\senseserviceuser* or *domain\senseserviceuser*.

   If you enter a username that is more than 20 characters long, it must be in UPN format, and must include the full domain name. For example, longusername@full.domain.name.

On the final screen of the installation program, you do not have the option to start the Qlik Sense services, instead the following message is displayed: **The service user does not have administrator privileges. See the documentation for more information**.

Next, run the bootstrap command in an elevated command prompt while registering the rim node with a certificate.

To run the bootstrap command:

1. On the rim node, open an elevated command prompt window. The bootstrap command elevates your rights enabling you to perform tasks that require an administrator, such as installing certificates and adding performance counters.

2. In the command prompt, navigate to the installed location: *Program Files\Qlik\Sense\Repository* and run the `Repository.exe -bootstrap` command. While the bootstrap is running, in the QMC on the central node, register the rim node with a certificate that is generated. For more information, in *Services (page 22)*, see the Repository service.
3. On the central node, register the rim node in the QMC, see: *Adding a Qlik Sense node (page 94)*. After you have registered the rim node the bootstrap process will terminate.
4. Exit the command prompt.
5. In Windows, **Services**, start all Qlik Sense services.

Changing the user account type to run the Qlik Sense services on an existing site

Follow the instructions in this section if you used an administrator user account when installing Qlik Sense, and later wish to change to use an account without administrator privileges to run the Qlik Sense services.

Do the following:

1. In Windows, either create a new or use an existing domain or local user account to run the Qlik Sense services.
2. If the service account user does not have administrator privileges, you must add the user to the following groups in **Computer Management** > **System Tools** > **Local Users and Groups** > **Groups**.
   - Qlik Sense Service Users
   - Performance Monitor Users
3. Open the **Control Panel** and then select **System and Security**>**Administrative Tools**>**Services**.
4. Stop all services except the **Repository Database**.
5. Assign **Full control** permission for the dedicated service account to the folder *%ProgramData%\Qlik\Sense*.
6. As an administrator, open an elevated command prompt.
7. Navigate to the *Program Files\Qlik\Sense\Proxy* folder and run `Proxy.exe -bootstrap`.
8. Navigate to the *Program Files\Qlik\Sense\Scheduler* folder and run `Scheduler.exe -bootstrap`.
9. Navigate to the *Program Files\Qlik\Sense\Repository* folder and run `Repository.exe -bootstrap`.
   If you are changing the user account on your primary or central node, run `Repository.exe -bootstrap -iscentral`.
10. Close the elevated command prompt.
11. Change the log on credentials for each of the Qlik Sense services as follows:
    a. Right-click the service and select **Properties**.
    b. Select the **Log On** tab and then **This account**.
    c. Enter the credentials for the dedicated service account and click **OK**.

    The services are as follows:
    - Qlik Sense Engine Service
    - Qlik Sense Printing Service
    - Qlik Sense Proxy Service

- Qlik Sense Repository Service
- Qlik Sense Scheduler Service
- Qlik Sense Service Dispatcher

> *If you are using a user account with administrative privileges, keep the Qlik Sense Repository Database running under the Local System account. Do not change the account.*

> *Depending on your setup some of the services may not be available.*

12. Start the Qlik Sense Service Dispatcher, and then the Qlik Sense Repository Service (QRS).
13. Start the rest of the Qlik Sense services.

3.7 Performing a silent installation

When running a silent installation, Qlik Sense is installed with no dialogs at all. This means all features, properties and user selections have to be known before performing a silent installation. All setup options that are available in the user interface of the installer can be performed with silent operations.

Do the following:

1. Select **Start > All Programs > Accessories > Command Prompt**.
   The **Command Prompt** window is displayed.
2. In the **Command Prompt** window, navigate to the folder containing the *Qlik_Sense_setup.exe* file.
3. Enter *Qlik_Sense_setup.exe* followed by the silent installation syntax preferred.

> *Note that elevation will take place if run from an unelevated process and the UAC is on.*

Syntax

```
Qlik_Sense_setup.exe [-silent] {-log "path\filename"} {layout="path"}
{desktopshortcut=1|0} {skipstartservices=1|0} {installdir="path"}
{userwithdomain="domain\user"} {userpassword="password}
{dbpassword="password"} {hostname="www.machinename.domain.com"}
{sharedpersistenceconfig="configfilepath"} {skipvalidation=1|0}
{databasedumpfile="path"}
```

`Qlik_Sense_setup.exe -?` or `-h`	Brings up the on-screen silent setup help.

Commands

`-silent` (or `-s`)		Command line-driven setup without UI (mandatory).
`-log` (or `-l`)	[log file name with path]	Log file directory and log file name. *The user must have access to this directory.*
`-layout`	[destination directory]	Extracts files (including *.msi* files) to the destination directory. *This argument should not be combined with other command line arguments.*

Arguments

Arguments are separated by space and presented in the form [Argument]="[Value]". The double quotes can normally be omitted but may be needed, for example, when a path contains spaces.

The default values are the same as those used in the setup user interface.

`desktopshortcut`	`1`|`0` (defaults to 1 on clean installs)	Installs desktop shortcuts.
`skipstartservices`	`1`|`0` (defaults to 0 on clean installs, otherwise the current state.)	To skip starting services after the installation has finished.
`installdir`	[path to custom install directory]	Need only be defined if the default install directory will not be used (*%ProgramFiles%\Qlik\Sense*).
`userwithdomain`	[domain\username]	The username used to run the Qlik Sense services.
`userpassword`	[password]	The password of the user used to run the services.
`dbpassword`	[password]	Password for the database superuser that creates the user that runs the database.
`hostname`	[address of the central node]	The central node uses certificates to communicate securely with other servers. Leave blank to use default.

sharedpersistenceconfig (or spc)	[path to configuration file including the filename]	Activates setup of shared persistence as storage method. All settings for shared persistence must be in the configuration file referenced here. ⚠ *This is a parameter must be configured to install successfully.* *Shared persistence configuration file syntax (page 108)*
skipvalidation	1|0	For installation or upgrade, if the value is 1, skip validation of the password provided for service user and shared folder access. For silent installation, database connection tests are also skipped.
databasedumpfile	[path to database dump file]	To retrieve your repository database backup dump file.

ⓘ *If you enter a username that is more than 20 characters long, it must be in UPN format, and must include the full domain name. For example, longusername@full.domain.name.*

Example 1: To install Qlik Sense

```
Qlik_Sense_setup.exe -s spc="\\configpath\spc.cfg"
userwithdomain=mydomain\myUser userpassword=myPassword
dbpassword=mydbpassword
```

Example 2: To install Qlik Sense while redirecting the installation and log files to a different location

```
Qlik_Sense_setup.exe -s -l "c:\mylogpath" spc="\\configpath\spc.cfg"
installdir="c:\mycustompath" userwithdomain=mydomain\myUser
userpassword=myPassword dbpassword=mydbpassword
```

Shared persistence configuration file syntax

Configure the shared persistence storage model, using the sharedpersistenceconfig argument, and point to a configuration file that contains the settings to be used in the installation.

Example:

```
Qlik_Sense_setup.exe -s spc="\\configpath\spc.cfg" userwithdomain=domain\yourserviceuser
userpassword=yourserviceuserpassword dbpassword=yoursuperuserpassword
```

The configuration file is in XML format. You need to create the file according to the example described here.

```
<?xml version="1.0"?>
<SharedPersistenceConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <DbUserName>username</DbUserName>
  <DbUserPassword>password</DbUserPassword>
  <DbHost>IP or Hostname</DbHost>
  <DbPort>4432</DbPort>
  <RootDir>\\server\share</RootDir>
  <StaticContentRootDir>\\server\share\StaticContent</StaticContentRootDir>
  <CustomDataRootDir>\\server\share\CustomData</CustomDataRootDir>
  <ArchivedLogsDir>\\server\share\ArchivedLogs</ArchivedLogsDir>
  <AppsDir>\\server\share\Apps</AppsDir>
  <CreateCluster>true</CreateCluster>
  <InstallLocalDb>false</InstallLocalDb>
  <ConfigureDbListener>true</ConfigureDbListener>
  <ListenAddresses>*</ListenAddresses>
  <IpRange>0.0.0.0/0,::/0</IpRange>
  <MaxConnections>100</MaxConnections>
  <!--<JoinCluster>true</JoinCluster>-->
  <ConfigureLogging>true</ConfigureLogging>
  <SetupLocalLoggingDb>true</SetupLocalLoggingDb>
  <QLogsWriterPassword>writerpw</QLogsWriterPassword>
  <QLogsReaderPassword>readerpw</QLogsReaderPassword>
  <QLogsHostname>ip/hostname</QLogsHostname>
  <QLogsPort>4432</QLogsPort>
  </SharedPersistenceConfiguration>
```

Configuration file syntax

Setting	Description
DbUserName	User name of the repository database user.
DbUserPassword	Password of the repository database user.
DbHost	Hostname of the machine running the repository database.
DbPort	Port used to communicate with the repository database.
RootDir	Root directory for the file share to use as content storage. We recommend that you keep the content in this folder's sub-directories, but this can be changed in the StaticContentRootDir, CustomDataRootDir and ArchivedLogsDir settings.
AppsDir	Directory to store apps in.
StaticContentRootDir	Root directory for all static content of the site.
CustomDataRootDir	Root directory for all custom data of the site, for example, custom connectors.
ArchivedLogsDir	Directory to save archived log files in.
CreateCluster **JoinCluster**	Set CreateCluster to True if you want to create a new cluster, or set JoinCluster to True if you want to join an existing cluster. You can only use one of these settings in the configuration file. The other setting needs to be removed, or commented out like `<!--<JoinCluster>true</JoinCluster>-->`.

Setting	Description
InstallLocalDb	Set to True if you want to install a local PostgreSQL database on the node when you create a new cluster. This setting can only be used together with the CreateCluster setting.
ConfigureDbListener	Set to True if you want to configure the PostgreSQL database installed by Qlik Sense to listen to database connections from other nodes. You need to configure the ListenAddresses and IpRange settings.
ListenAddresses	Addresses that the database service should listen to. You can supply a comma separated list of IPv4 or IPv6 addresses, or *0.0.0.0* (for all IPv4 addresses), `::/0` (for all IPv6 addresses) or * (for all addresses).
IpRange	Subnet specification that covers the IP addresses of all nodes in your site. Either add one row for each node, using /32 as suffix for each address, or add a subnet that covers all addresses using, for example, /24 as suffix. To allow all servers to access the repository database, use 0.0.0.0/0. If entering multiple IP addresses or ranges, use a comma separated list. A range can be either IPv4 or IPv6.
MaxConnections	Specify the maximum number of concurrent connections to the database. The default value is 100. If you have a multi-node site multiple this value by the number of nodes in the cluster. For example, `<MaxConnections>100</MaxConnections>` is a single server deployment.
ConfigureLogging	Set ConfigureLogging to true if you want to set up centralized database logging.
SetupLocalLoggingDb	Setting SetupLocalLoggingDb to true is equivalent to clicking **New Logging Database** in the installer UI. A new logging database will be installed with Qlik Sense.
QLogsWriterPassword	Password of the qlogs_writer user account.
QLogsReaderPassword	Password of the qlogs_reader user account.
QLogsHostname	Host name of the logging database. Set QLogsHostname when SetupLocalLoggingDb is set to false.
QLogsPort	Port number of the logging database. Set QLogsPort when SetupLocalLoggingDb is set to false.

Deprecated command line arguments

The use of the following command line arguments is no longer recommended.

`rimnode`	Determines the Repository role.
`-rimnodetype` (or `-rnt`)	Installs all the features required for the rim node type selected. The node type can be any one of: `Complete`, `Proxy`, `Engine`, `Scheduler`.

3.8 Qlik Sense deployments in a multi-cloud environment

Once you have installed Qlik Sense Enterprise for Windows, you can set up a multi-cloud environment. To do this, connect your Qlik Sense Enterprise for Windows to a cloud hub on:

- Qlik Sense Enterprise for elastic deployments, which supports deployment to public or private clouds on a customer-managed infrastructure
- Qlik Cloud Services, where Qlik manages the infrastructure

You can set up one of each of these deployments to connect to a single Qlik Sense Enterprise for Windows. Once your multi-cloud deployment is configured, you can distribute Qlik Sense apps that you create in Qlik Sense Enterprise for Windows to the cloud for consumption.

To set up your deployment to connect to Qlik Cloud Services, see *Qlik Sense deployments to Qlik Cloud Services (page 111)*.

To set up your deployment to connect to Qlik Sense Enterprise for elastic deployments, see *Qlik Sense Enterprise for elastic deployments (page 113)*.

Qlik Sense deployments to Qlik Cloud Services

A multi-cloud deployment allows you to distribute Qlik Sense apps to the cloud for consumption. You can set up your deployment to connect Qlik Sense Enterprise for Windows to Qlik Cloud Services where your services are hosted and managed by Qlik. Contact Qlik to obtain a Qlik Cloud Services license.

> *The current status of Qlik Cloud Services and recent communications can be viewed here:* https://status.qlikcloud.com

To set up your Qlik Cloud Services deployment:

1. Deploy Qlik Sense Enterprise for Windows. For more information, see *Installing Qlik Sense in a multi-node site (page 85)*.
2. Go to the Multi-cloud Setup Console and enter your license information.
   You will receive an email from Qlik with your service account credentials.
3. Follow the link in the email to reset your Qlik service account password.
4. Go to www.qlikcloud.com to configure your deployment domain and IdP.
   For more information about configuring your IdP, see *Qlik Cloud Services deployment configuration (page 111)*.

Qlik will send you an email when your Qlik Cloud Services deployment is ready.

Qlik Cloud Services deployment configuration

To set up your QCS deployment, you complete a form on *qlikcloud.com* which provides Qlik with the information needed to configure your deployment. This information includes your region, requested subdomain, and details of your identity provider configuration.

For multi-cloud, you can only use identity providers that are compatible with OpenID Connect (OIDC).

Setting up your Qlik Cloud Services deployment

For Qlik Cloud Services, you configure your deployment including your IdP settings by completing the form on *qlikcloud.com*. Those details are submitted to Qlik. Click a field in the form to display the associated help text. You can use the form both for interactive login of users (**Primary Identity Provider**) and for acquiring tokens for API access (**Machine to machine**).

The form has the following settings.

Setting	Description
Region	Geographical region of the deployment. Will be added to the QCS website address. *When you distribute an app it is stored in the region selected. You cannot change this setting after you make your initial selection.* *Qlik recommends selecting the region closest to where you live for best performance. You should also consider local laws and regulations when selecting your region.*
Subdomain	Value that should be unique and representative of your deployment. Will be added to the QCS website address. This cannot be edited after saving.
Configuration endpoint	URL to the endpoint that provides configuration information for the OAuth clients to interface with the IdP using the OpenID Connect protocol.
User directory	The directory where your user data is stored. By connecting to the user directory, you can synchronize user data. Only letters, numbers, hyphens, underscores, and dots are supported.
Client ID	ID of the configured client at the IdP for interactive user authentication. For interactive flows, clients should be granted the scopes "openid" and "profile".
Client secret	Secret for the client configured at the IdP.
Claims mapping: Subject	Default value: *sub*. If you want to use another claim for *subject*, add the claim name here. Use comma to separate values.
Claims mapping: Groups	Default value: *groups* as a string array. If you want to use another claim for *groups*, add claim to use for group information here. Use comma to separate values.
Claims mapping: Name	Default value: *name*. If you want to use another claim for *name*, add the claim name here. Use comma to separate values.

Qlik Sense Enterprise for elastic deployments

As part of a Qlik Sense Enterprise Multi-Cloud deployment, you can install Qlik Sense Enterprise for elastic deployments. This is an implementation of Qlik Sense Enterprise running on a Kubernetes cluster using containers. This approach allows deployments into Kubernetes clusters running in public or private clouds on customer managed infrastructures.

Before you install:

- Have your license key available with the capabilities enabled.

> *You will need to contact Qlik to obtain an updated LEF to enable your Qlik Sense Enterprise for elastic deployments.*

- Check that your environment meets the system requirements. You need a running Kubernetes cluster to install

Installation and configuration

An implementation of Qlik Sense Enterprise for elastic deployments can vary depending on the configuration required. The following pages detail how to perform a basic installation followed by additional pages on the key elements required to do a production implementation:

- *Preparing Qlik Sense Enterprise for elastic deployments (page 113)*
- *Installing Qlik Sense Enterprise for elastic deployments (page 115)*

Additional configuration sections:

- *Distributing apps to your Qlik Sense Enterprise for elastic deployment (page 117)*
- *Identity providers in multi-cloud – introduction (page 118)*
- *Viewing logs in Qlik Sense Enterprise for elastic deployments (page 133)*
- *Configuring MongoDB in Qlik Sense Enterprise for elastic deployments (page 135)*
- *Configuring certificates in your Qlik Sense Enterprise for elastic deployment (page 132)*
- *Monitoring metrics in Qlik Sense Enterprise for elastic deployments (page 136)*

Preparing Qlik Sense Enterprise for elastic deployments

For elastic deployments, Qlik Sense Enterprise is deployed to a Kubernetes cluster in the form of a set of container images in a package provided as a Helm chart. To be able to install the following items are required to be in place (also refer to the system requirements for more details).

- A running Kubernetes cluster – this can be run locally for development purposes or deployed to cloud vendors including AWS and Azure.
- The Kubernetes cluster must have access to file storage to persist data. This storage must be provided as a Persistent Volume Claim that allows **`readwritemany`** access. You will need the name of this claim when installing

The Qlik documentation does not cover the installation and configuration of a Kubernetes cluster and you should review the documentation for this at either https://Kubernetes.io or for the cloud vendor or product you are using.

You will also need the following tools installed on your local machine to interact with your Kubernetes environment, issue commands and deploy software:

- Kubectl - Install kubectl on the machine you will run admin commands from. You can find further details for your operating system at https://Kubernetes.io.

  > *This can point to different clusters if you have more than one. Ensure that the commands go to the right Kubernetes instance.*

- Helm - Helm is a package manager built for Kubernetes. It has a concept known as charts, used to define what services are required, what images are used, and default settings when the run in the Kubernetes cluster. It is used to push the Qlik Sense package into Kubernetes and relies on kubectl, so it must be installed on the same machine as kubectl. Qlik uses helm to define a default chart to make deployments simple for customers.
  To install Helm on your local machine follow the instructions for your operating system at https://docs.helm.sh/.

Preparing your local tools

Once you have set up your Kubernetes cluster, you must prepare your local tools to work with your Kubernetes cluster. To prepare your local tooling you must:

- Bind kubectl to your Kubernetes cluster
- Add Qlik's helm chart repository
- Initialize helm to work with your Kubernetes cluster

Before you begin you should have the following installed on your local machine:

- Kubectl
- Helm

To bind kubectl to your Kubernetes cluster:

1. Verify that kubectl is pointing to your Kubernetes cluster using the following command:
   ```
   kubectl config current-context
   ```
2. If kubectl is not pointing to your Kubernetes cluster, use the following command to get a list of available clusters:
   ```
   kubectl config get-clusters
   ```
3. Set the kubectl to point to the desired cluster using the following command:
   ```
   kubectl config set-cluster <cluster-name>
   ```

To add Qlik's helm chart repository:

1. Run the following command to add Qlik's helm chart repository to Helm. This is where Qlik Sense is pulled from:

```
helm repo add qlik https://qlik.bintray.com/stable
```

2. Use the following command to get a list of all configured repositories and verify that the Qlik helm chart repository was successfully added:
   ```
   helm repo list
   ```

To use helm to with your Kubernetes cluster it needs to be initialized to create the helm Tiller pod that handles installations:

1. The following command is used to do this in simple cases
   To use helm to deploy into Kubernetes, the helm Tiller pod is added to the Kubernetes cluster first.
   The following command is used to do this in simple cases:
   ```
   helm init --wait
   ```
2. If the Kubernetes cluster has security features such as RBAC enabled then the following commands should be run in addition:
   ```
   kubectl create serviceaccount --namespace kube-system tiller
   kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --
   serviceaccount=kube-system:tiller

   helm init --upgrade --wait
   kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"spec":
   {"serviceAccount":"tiller"}}}}'
   ```

Installing Qlik Sense Enterprise for elastic deployments

Once you have set up your Kubernetes cluster and prepared your local tools, you are ready to install Qlik Sense Enterprise into your Kubernetes cluster.

To recap, as a minimum before you install you will have:

- Set up a Kubernetes cluster and added **`readwritemany`** storage
- Prepare your local tools to work with your Kubernetes cluster

The steps below cover the steps to install a first simple install. This includes deploying a MongoDB instance and simple IDP to get you running. To move it to a production ready state, you should review the additional topics for the following areas:

- Understanding and configuring an IDP to authenticate users
- Configuring a separate MongoDB instance
- Viewing and handling logs
- Monitoring services

Providing configuration settings

When installing Qlik Sense Enterprise for elastic deployments you can specify settings to the installer in two ways:

- As parameters in the `helm install` command.
- Referencing the settings in a **values.yaml** and using this in the `helm install` command.

Storing the configuration settings in a **values.yaml** allows you to reuse the settings in multiple deployments and add new config sections simply. This can also be version controlled.

Use of a **values.yaml** file will be used predominately in the Qlik help. You can find more information about YAML files online on sites such as https://en.wikipedia.org/wiki/YAML.

Installing Qlik Sense

Complete the following steps to install Qlik Sense:

1. Create a text file called **values.yaml**.
   a. Add the following content to the file:
```
#This setting enables dev mode to include a local MongoDB install
devMode:
  enabled: true

#This setting accepts the EULA for the product
engine:
  acceptEULA: "yes"
```
   > *If devMode.enabled is set to true, a MongoDB instance is deployed inside of your Qlik Sense Enterprise for elastic deployments in Kubernetes for development and testing purposes only.*

   b. Add the following section to accommodate the RBAC security mode in Kubernetes:
```
#These settings are to accomodate if RBAC is enabled
mira:
 rbac:
   create: true
 serviceAccount:
   create: true

elastic-infra:
 traefik:
   rbac:
     enabled: true
 nginx-ingress:
   rbac:
      create: true
```
   c. Add the following content to point the services requiring storage to the Kubernetes Persistent Volume claim, update the name of the PVC as needed.

   > *If you are using Kubernetes via Docker for Desktop or Minikube then you do not need to have this section*
```
#These setting specifies the storage for the engine
engine:
  persistence:
    enabled: true
    accessMode: ReadWriteMany
    existingClaim: my-persistent-vol

#These setting specifies the storage for the resource-library
```

```
resource-library:
  persistence:
    enabled: true
    accessMode: ReadWriteMany
    existingClaim: my-persistent-vol
```

   d. Save the file.

2. Run the following command:
   `helm install -n qsefe qlik/qsefe -f values.yaml`
   The software now starts deploying to the Kubernetes cluster, including downloading all the images and running them.
3. You can now use kubectl to check the progress. Run the following command:
   `kubectl get pods`
   If your deployment was successful you will see something similar to this:

```
NAME                                      READY    STATUS     RESTARTS   AGE
qsefe-collections-7f456595b8-vjhtf        1/1      Running    7          26m
qsefe-edge-auth-858f89b849-42z66          2/2      Running    0          26m
… (lines removed for brevity)
```

It typically takes a few minutes to initialize and show a status of “Running”.

If services do not start check the log files of the service for more information. See Viewing logs in Qlik Sense Enterprise for elastic deployments (page 133). If the engine or resource library remain in a pending state, check that the Kubernetes cluster has **`readwritemany`** *storage available as a persistent volume claim and that is it correctly referenced in the YAML.*

Accessing the deployment

To connect to the hub and confirm the install you need to obtain the URL for the install inside the Kubernetes cluster. This can vary depending on the configuration and / or vendor but typically you can find the address or IP the install is running on by running the following command:

```
kubectl describe service qsefe-nginx-ingress-controller
```

This command returns the address you can add and use to navigate to the hub in your web browser.

You can now connect a Qlik Sense Enterprise for Windows deployment to the cluster and distribute apps to it.

In this simple deployment an example Identity Provider is automatically configured. This allows you to login to the hub with some sample accounts. When you browse to the hub you will be asked to login and you can use the sample account of **harley@qlik.example** with a password of **Password1!** .

The simple IdP is for test purposes only and you should configure a full IdP by reviewing Identity providers in multi-cloud – introduction (page 118).

Distributing apps to your Qlik Sense Enterprise for elastic deployment

Once you have Qlik Sense Enterprise for elastic deployments running, you can distribute apps into it from your Qlik Sense Enterprise for Windows deployment. To distribute apps, complete the following steps:

1. Create a Deployment in the Multi-cloud Setup Console on your Qlik Sense Enterprise for Windows deployment.
   See *Multi-Cloud Setup Console - start page* and *MSC - Deployments*.
2. Create a distribution policy to decide which applications should be distributed.
   See *Distribution policies - introduction*.
3. Publish the application setting the **collection**, **userswithaccess** or **groupswithaccess** properties on the app.
   See *Publishing apps to cloud hub collections*.

Once you have distributed the apps, you will be able to open them from the hub.

Identity providers in multi-cloud – introduction

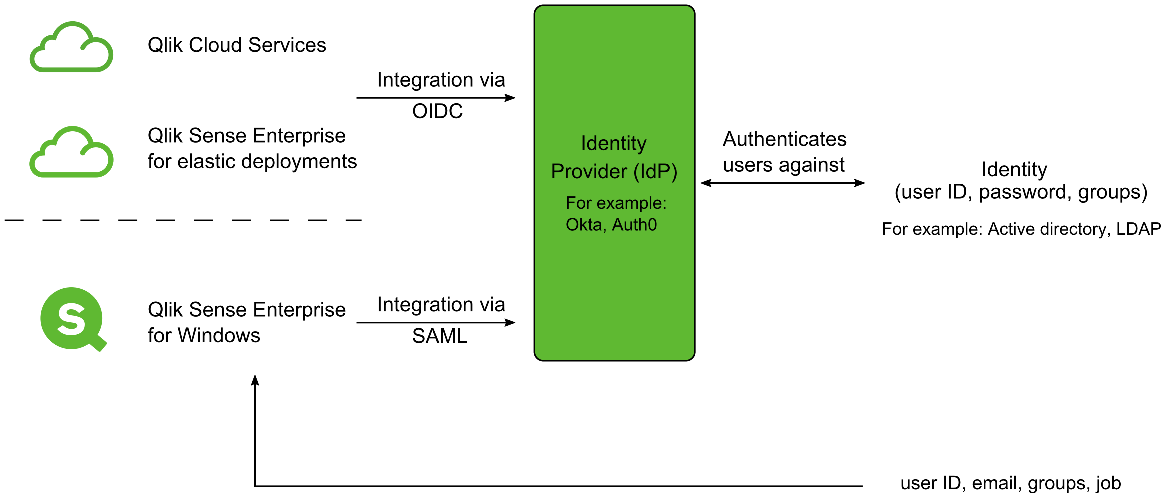
An identity provider (IdP) manages identity information for users and provides authentication services. The identity provider enables single sign-on (SSO) so that you can access other websites, without having to log in repeatedly. In contrast to on-premise technologies, such as Active Directory and LDAP, identity providers also offer a consistent and governed experience when accessing cloud services, eliminating the need to create accounts for each new service.

> *If user accounts are stored in Active Directory, the IdP can still enable integration into cloud software.*

IdPs in a multi-cloud deployment

In a multi-cloud deployment, an IdP delivers the following:

- Secure authentication of a user and a common identity (user ID and groups) passed between all deployments.
- Common user identity to assign a license to (to avoid double use).
- Common user ID and attributes, such as groups, to use when applying access control to content.


Qlik Cloud Services
Integration via OIDC
Qlik Sense Enterprise for elastic deployments
Identity Provider (IdP)
For example: Okta, Auth0
Authenticates users against
Identity (user ID, password, groups)
For example: Active directory, LDAP
Qlik Sense Enterprise for Windows
Integration via SAML
user ID, email, groups, job

Requirements of a multi-cloud IdP

Both Qlik Cloud Services and Qlik Sense Enterprise for elastic deployments integrate with an IdP using the OpenID Connect (OIDC) standard. This is a standard that allows both interactive login, where a user logs in via a browser, and automated login, using APIs via a software product.

Qlik Sense Enterprise for Windows currently does not support OIDC, but supports SAML, or any method that allows a consistent user identity to the one provided by the IdP.

> *In summary, an IdP for multi-cloud must support both OIDC and SAML.*

The following is required from the IdP to be able to set up Qlik Sense Enterprise for elastic deployments to use it:

- **discoveryUrl**: the OpenID Connect Discovery URL which allows applications, such as Qlik Sense, to use the IdP with minimal configuration.
- **clientId**: uniquely identifies the client from the IdP.
- **clientSecret**: the secret that the client uses along with the client ID to authentication with the IdP.
- **realm**: the name to associate with the IdP.
- **hostname**: the hostname that is used for the deployment of Qlik Sense Enterprise for elastic deployments.

These values are added to the **values.yaml** file under the identity-providers section when installing Qlik Sense Enterprise for elastic deployments.

Step-by-step examples of this configuration are provided for the following IdP vendors:

- Okta
  See *Setting up Okta* .
- Auth0
  See *Setting up Auth0*.
- ADFS
  See *Setting up ADFS*.

Setting up ADFS

ADFS is an authentication and authorization platform.

You can configure ADFS as an identity provider (IdP) for use with Qlik Sense Enterprise for elastic deployments and Qlik Sense Enterprise for Windows (QSEfW). You will create an application group, a server application, and a Web API to be used for interactive login (QSE for elastic deployments). You will also map claims from Active Directory to the ID token.

Creating required ADFS resources for QCS or QSE for elastic deployments for interactive logins

For setting up ADFS, you need an application group and a server application.

The following procedures are examples using ADFS 10. Please review the ADFS documentation for more information and latest instructions.

Adding an application group and creating a server application

Do the following:

1. Open the **Add Application Group Wizard**.
2. Enter a name for the application group.
3. For **Template**, select **Server application**.
4. Click **Next**.
   The **Server application** page is opened.
5. Enter a name for the application.
   Example: *1234567890*
6. Enter a client identifier for the application, and note it down, it will be used as client ID.
   Example: *https://adfs.elastic.example/1234567890*.

   In this example, https://adfs.elastic.example is the tenant domain and 1234567890 is a unique identifier for the application. The client identifier must be a URL. ADFS will only include custom claims in the id_token for applications with URL IDs, see Customize claims to be emitted in id_token when using OpenID Connect or OAuth with AD FS 2016.

7. For **Redirect URI**, set the redirect URL to the login callback for the tenant in the format *https://<host>/login/callback/*.
   Example: *https://adfs.elastic.example/login/callback*
8. Optionally, enter a description.
9. Click **Next**.
   The **Configure Application Credentials** page is opened.
10. Select **Generate a shared secret**. Note down this secret, you will not have access to it again. You will use it as client secret.
11. Finish the wizard.

Adding a web API to the application group

You will add a web API to the application group that you created.

Do the following:

1. Open the application group you created earlier.
2. Select **Add application** > **Web API**.
3. Add the client ID from the application group as in identifier.
4. Click **Next**.
   The **Choose Access Control Policy** page is opened.

5. Apply a policy and click **Next**.
   The **Configure Application Permissions** page is opened.
6. For **Permitted scopes**, select the following: *allatclaims*, *email*, *openid*, and *profile*.
7. Finish the wizard.

Configure claims for the id_token

Do the following:

1. Open the application group to edit the web API you created. Open the **Issuance Transform Rules** tab.
2. Create a rule from the rule template **Send LDAP Attributes as Claims**.
3. Select **Active Directory** as the attribute store.
4. Add claims mappings. You may need to type the outgoing claim.
5. Map *Token-Groups - Unqualified Names* to *groups*.
6. Map *Display-Name* to *display_name*.
7. Finish the claims mapping.

Using ADFS as an IdP for Qlik Sense Enterprise for elastic deployments

You can use Auth0 as an identity provider for logging into a QSE for elastic deployments tenant using a user from ADFS.

3.9 Connecting QSE for elastic deployments with ADFS

Before you start, make sure you have the following:

- ADFS installation
- the required resources configured in ADFS
- Configuration settings from your ADFS application: *discoveryUrl*, *clientId*, and *clientSecret*
- The following values from your hybrid deployer: public key, key ID, and issuer.

Many of the code examples contain placeholder values that need to be replaced by your own values.

You provide configuration to QSE for elastic deployments by using a *values.yml* file. The *values.yml* file should look like the following example:

```
devMode:
  enabled: true

engine:
  acceptEULA: "yes"

identity-providers:
  secrets:
    idpConfigs:
      - discoveryUrl: "https://adfs-host/adfs/.well-known/openid-configuration"
        clientId: "https://adfs.elastic.example/1234567890"
        clientSecret: "<client secret>"
        realm: "ADFS"
    hostname: "adfs.elastic.example"
    useClaimsFromIdToken: true
    claimsMapping:
     sub: ["sub", "appid"]
     client_id: "appid"
     name: "display_name"
   - issuerConfig:
     issuer: https://the-issuer
    primary: false
    realm: "ADFS"
    hostname: "adfs.elastic.example"
    staticKeys:
    - kid: "thekid"
     pem: |-
      -----BEGIN PUBLIC KEY-----
      MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEsMSxQjXxrvqoKSAREQXsr5Q7+/aetjEb
      OUHt8/Cf73WD56cb4QbHthALl5Ej4MUFOAL9imDmVQe58o9b1j5Zo16Rt1gjLDvd
      nqstc+PX4tyxqGadItJAOU3jka7jYghA
      -----END PUBLIC KEY-----
```

It is important to note that the userClaimsFromIdToken flag is set to true. The flag instructs edge-auth to use the claims from the ID token instead of querying for userinfo. This is because ADFS returns very little in the userinfo response and instead includes most information in the ID token.

You will have to insert your own values for *discoveryUrl*, *clientId*, *clientSecret*, *realm* and *hostname*.

3.10 Applying the configuration to your cluster

Use Helm (see https://helm.sh/) to apply the configuration in your *values.yml* file to your Kubernetes cluster:

```
$ helm upgrade \
 --install \
 qsefe qlik/qsefe \
 -f values.yml
```

To make sure that your configuration has been applied, you can run the `get values` command to see the resolved configuration:

```
$ helm get values qsefe

devMode:
 enabled: true
engine:
 acceptEULA: "yes"
identity-providers:
 secrets:
  idpConfigs:
   - discoveryUrl: "https://adfs-host/adfs/.well-known/openid-configuration"
    clientId: "https://adfs.elastic.example/1234567890"
    clientSecret: "<client secret>"
    realm: "ADFS"
    hostname: "adfs.elastic.example"
    useClaimsFromIdToken: true
    claimsMapping:
      sub: ["sub", "appid"]
      client_id: "appid"
      name: "display_name"
   - issuerConfig:
      issuer: https://the-issuer
    primary: false
    realm: "ADFS"
    hostname: "adfs.elastic.example"
    staticKeys:
    - kid: "thekid"
      pem: |-
        -----BEGIN PUBLIC KEY-----
        MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEsMSxQjXxrvqoKSAREQXsr5Q7+/aetjEb
        OUHt8/Cf73WD56cb4QbHthALl5Ej4MUFOAL9imDmVQe58o9b1j5Zo16Rt1gjLDvd
        nqstc+PX4tyxqGadItJAOU3jka7jYghA
        -----END PUBLIC KEY-----
```

Configure your hosts file

This section is only relevant if there is no DNS.

In order for <hostname> to resolve, add the following to your /etc/hosts file:

```
127.0.0.1   <hostname>
::1         <hostname>
```

Log in to your tenant

You are now set to log into your tenant with a user from your ADFS deployment. In your browser, go to *https://<tenant address>* and you should be redirected to an ADFS login page. After a successful login you reach a home page to which apps are distributed.

Setting up Auth0

Auth0 is an authentication and authorization platform.

You can configure Auth0 as an identity provider (IdP) for use with Qlik Sense Enterprise for elastic deployments and Qlik Sense Enterprise for Windows (QSEfW).

Creating an Auth0 application and connection for QCS or QSE for elastic deployments for interactive logins

Create an Auth0 application, and connect it to an Auth0 database connection.

An Auth0 application allows an application, (QSEfW/QCS/QSE for elastic deployments), to use Auth0 for authentication. An Auth0 connection is a source of users, in this example, a database that you populate with users.

We assume that you have an Auth0 account and tenant created.

The following procedures are examples. Please review the Auth0 documentation for more information and latest instructions.

Creating a new application in Auth0

Do the following:

1. In the left menu in Auth0, open **Applications**.
2. Click **Create application**.
3. Give the application a name, select **Single Page Web Applications** and click **Create**.
4. Optionally, select your web app technology.
5. Select **Settings**.
6. In the box **Allowed Callback URLs**, add the URL to your host in the format *https://<host>/login/callback/*.
7. Scroll down and click **Save changes**.
8. Note down the **Client ID** value.
9. Note down the **Client Secret** value.
10. Scroll to the bottom and select **Advanced Settings**.
11. Select the **Endpoints** tab.
12. Note down the **OpenID configuration** URL for later.

Creating a database connection in Auth0

You will now create a database connection and configure your application to use this connection.

Do the following:

1. In the left menu, select **Connections** > **Database**.
2. Fill in a name for the database connection and click **Create**.
3. In the left menu, select **Applications**.
4. Open the tab **Connections**.
5. Enable the new database connection for your application.

Creating a new user (optional)

Do the following:

1. In the left menu, select **Users**.
2. Click **Create your first user**.
3. Fill in the fields and select the newly created connection.

Creating an Auth0 API and application for programmatic access

Begin by creating the API.

You set up programmatic access so that you can distribute content into Qlik Cloud Services (QCS) or Qlik Sense Enterprise for elastic deployments.

In Auth0, you will create a new API. In this case, the Auth0 API represents the protected QSE for elastic deployments resource API. In OAuth terms, you configure Auth0 for the Client Credentials Grant flow.

Begin by creating a new API for your application.

Do the following:

1. In the left menu, select **APIs**.
2. Click **Create API**.
3. Enter an API name.
4. For **Identifier**, enter `qlik.api`.
5. Click **Create**.
6. Go to the **Scopes** tab.
7. Add a new scope with the value any in the name and description and click **Add**.

Just like you created an Auth0 application for interactive logins above, you will now create an Auth0 application for programmatic authentication.

Do the following:

1. In the left menu, select **Applications**.
2. Click **Create Application**.
3. Select **Machine to Machine Applications**.
4. Click **Create**.
5. Select the API created above.
6. In the **Scopes** box, select **any**.
7. Click **Authorize**.
8. Select the **Settings** tab. In the **Allowed Web Origins** box, add the URL to your deployment.
9. Note down the **Client ID** value.
10. Note down the **Client secret** value.
11. Scroll to the bottom and select **Advanced Settings**.
12. Click the **Endpoints** tab.
13. Note down the **OAuth Token URL** value.
    This value together with client ID and client secret will be used in the configuration of QSE for Windows when adding a deployment.
14. In the left menu, select **APIs** and open your new API. Select the **Machine to Machine Applications** tab.
15. Verify that your new application has access to your new Auth0 API.

Using Auth0 as an IdP for Qlik Sense Enterprise for elastic deployments

You can use Auth0 as an identity provider for logging into a QSE for elastic deployments tenant and also for interacting with the tenant programmatically.

3.11 Connecting QSE for elastic deployments with Auth0

Before you start, make sure you have the following:

- Auth0 account
- Auth0 tenant
- Auth0 app, configured with interactive login and programmatic access
- Configuration settings from your Auth0 application: *discoveryUrl*, *clientId*, and *clientSecret*

> *Many of the code examples contain placeholder values that need to be replaced by your own values.*

You provide configuration to QSE for elastic deployments by using a *values.yml* file. The *values.yml* file should look like the following example:

```
devMode:
  enabled: true

engine:
  acceptEULA: "yes"

identity-providers:
  secrets:
    idpConfigs:
      - discoveryUrl: "<OpenID Configuration from Application>"
        clientId: "<Client ID from Application>"
        clientSecret : "<Client Secret from Application>"
        realm: "<Name for this IdP>"
        hostname: "<Hostname for your QSEfE tenant>"
        claimsMapping:
          client_id: [ "client_id", "<id>" ]
```

You need to enter the values for *discoveryUrl*, *clientId*, *clientSecret*, *realm*, *hostname*, and *id* (claims mapping).

3.12 Applying the configuration to your cluster

Use Helm (see https://helm.sh/) to apply the configuration in your *values.yml* file to your Kubernetes cluster:

```
$ helm upgrade \
  --install \
  qsefe qlik/qsefe \
  -f values.yml
```

To make sure that your configuration has been applied, you can run the `get values` command to see the resolved configuration:

```
$ helm get values qsefe

devMode:
  enabled: true
engine:
  acceptEULA: "yes"
```

```
identity-providers:
  secrets:
    idpConfigs:
      - discoveryUrl: "https://tenant.auth0.com/.well-known/openid-configuration"
        clientId: "<client ID>"
        clientSecret : "<client secret>"
        realm: "Auth0"
        hostname: "<hostname>"
```

Configure your hosts file

This section is only relevant if there is no DNS.

In order for `<hostname>` to resolve, add the following to your `/etc/hosts` file:

```
127.0.0.1    <hostname>
::1          <hostname>
```

Log in to your tenant

You are now set to log into your tenant. In your browser, go to *https://<tenant address>* and you should be redirected to an Auth0 login page. After a successful login you reach a home page to which apps are distributed.

Setting up Okta

Okta is an authentication and authorization platform.

This topic presents how to set up Okta to be used with Qlik Sense Enterprise for elastic deployments and Qlik Sense Enterprise for Windows (QSEfW). You can configure Okta as an identity provider (IdP) for use with Qlik Sense Enterprise for elastic deployments and QSEfW.

You will create the following:

- an application for interactive login (QSE for elastic deployments)
- programmatic use of Okta

Creating an Okta application and user for QSE for elastic deployments for interactive logins

Create an Okta application and a user. An Okta application allows an application, (QSEfW/Qlik Cloud Services (QCS)/QSE for elastic deployments), to use Okta for authentication.

We assume that you have an Okta account and tenant created.

When you install Qlik Sense Enterprise for Windows, with Multi-Cloud, you must use a developer account for Okta, see Okta Developer.

Creating a user

Create a user in Okta. You can skip this step if you have already created users.

Do the following:

1. Fill in first name and last name.
2. **Username**: Use your email address for user name.
3. Primary email: Same as **Username**.
4. For **Password**, select **Set by admin**.
5. Enter a password for the new user.
6. Optionally, clear the selection **User must change password in first login**.

Creating a new application in Okta

Create a new application, a tenant for QSE for elastic deployments from Okta.

Do the following:

1. In Okta, go to **Applications** and click **Add Application**.
2. For **Platform**, select **Web** and click **Next**.
3. Enter a name for the app.
4. Enter a base URI.

   > *This is the IP address or server name from your QSE for elastic deployments environment. Example: https://40.118.9.61*

5. Enter a login redirect URI.
   As for the base URI, you use the IP address or server name from your environment. Example: *https://40.118.9.61/login/callback*
6. In the **Grant type allowed** section, for client acting on behalf of itself, select **Client Credentials**.
7. Click **Done**.

Configuration for programmatic access

Configure Okta to support usage programmatically (in this case to support distribution to QSE for elastic deployments or QCS).

Creating an Okta API resource server and application for programmatic access

In Okta, you create a new Resource Server API. In this case, the Okta Resource Server API represents the protected QSE for elastic deployments resource API. In OAuth terms, you need to configure Okta for the Client Credentials Grant flow.

First, create a new Authorization Server (under the API tab) for your tenant.

Do the following:

1. In the top menu, select **API**.
2. Open **Authorization Servers**.
3. Click **Add Authorization Server**.

4. Fill in name, audience (must be `qlik.api`), and description.
5. Save the API.
6. Open the **Scopes** tab.
7. Click **Add Scope** tab.
8. Enter a name and description, and select **Set as default scope**.
9. Click **Create**.
10. Open the **Access Policies** tab.
11. Click **Add Policy**.
12. For name and description, enter *Grant Clients*.
13. For **Assign to**, keep the selection **All clients**.
14. Click **Create Policy**.
15. Click **Add Rule**.
16. Enter a name for the rule.
17. Clear the selections under **Client acting on behalf of a user**.
18. Click **Create Rule**.

Creating an Okta application for programmatic authentication

Just like you created an Okta application for interactive logins above, you will now create an Okta application for programmatic authentication.

Do the following:

1. In the Okta top menu, open **Applications**.
2. Click **Add Application**.
3. For **Platform**, select **Service** and click **Next**.
4. Enter a name for the app.
5. Click **Done**.

Using Okta as an IdP for Qlik Sense Enterprise for elastic deployments

You can configure Qlik Sense Enterprise for elastic deployments to use Okta as an identity provider.

After completing the steps, you will be able to log into a QSE for elastic deployments tenant using an Okta user name and password as well as interact with the QSE for elastic deployments tenant programmatically.

We assume that you are running QSE for elastic deployments on a Mac which has Kubernetes running using Docker for Mac. Also without this exact configuration, you should be able to use the same concepts if running Kubernetes in other supported ways.

3.13 Configuring QSE for elastic deployments to use Okta IdP

Before you start, make sure you have the following:

- Okta account
- Okta tenant
- Okta app, configured with interactive login and programmatic access.
- Configuration settings from your Okta application:
  - *discoveryUrl*: The OpenID Connect Discovery URL which allows applications, such as QSE for elastic deployments, to use Okta with minimal configuration.
  - *clientId*: Uniquely identifies the client that is using Okta for authentication.
  - *clientSecret*: Secret that the client uses along with the Client ID to use Okta for authentication.

Many of the code examples contain placeholder values that need to be replaced by your own values.

You provide configuration to QSE for elastic deployments by using a *values.yml* file. The *values.yml* file should look like the following example:

```
devMode:
  enabled: true

engine:
  acceptEULA: "yes"

identity-providers:
  secrets:
    idpConfigs:
      - discoveryUrl: "<OpenID Configuration from Application>"
        clientId: "<Client ID from Application>"
        clientSecret : "<Client Secret from Application>"
        realm: "<Name for this IdP>"
        hostname: "<Hostname for your QSEfE tenant>"
```

You need to enter the values for *discoveryUrl*, *clientId*, *clientSecret*, *realm*, and *hostname*.

In Okta, you can find your *Client ID* and *Client secret* under the **General** tab in the **Client Credentials** section for the application you created.

3.14 Applying the configuration to your cluster

Use Helm (see https://helm.sh/) to apply the configuration in your *values.yml* file to our Kubernetes cluster:

```
$ helm upgrade qsefe qlik/qsefe -f values.yml
```

To make sure that your configuration has been applied you can run `get values` command to see the resolved configuration:

```
$ helm get values qsefe
devMode:
  enabled: true
engine:
  acceptEULA: "yes"
identity-providers:
  secrets:
    idpConfigs:
      - discoveryUrl: "https://dev-<tenantid>.oktapreview.com/.well-known/openid-configuration"
        clientId: "<clientID code>"
        clientSecret : "<clientsecret code>"
        realm: "Okta"
        hostname: "<hostname>"
```

Configuring your hosts file

This section is only relevant if there is no DNS.

For <hostname> to resolve, add the following to your /etc/hosts file:

```
127.0.0.1   <hostname>
::1         <hostname>
```

Log in to your tenant

You are now set to log into your tenant. In your browser, go to *https://<tenant address>* and you should be redirected to an Okta login page. After a successful login you reach a home page to which apps are distributed.

Adding programmatic configuration to QSE for elastic deployments

You now need an IdP configuration to QSE for elastic deployments to point to the application and authorization server created above. Note that a `primary: true` was added to the existing configuration you had.

```
devMode:
  enabled: true

engine:
  acceptEULA: "yes"

identity-providers:
  secrets:
    idpConfigs:
      - discoveryUrl: "https://dev-<tenantid>.oktapreview.com/.well-known/openid-configuration"
        clientId: "<client ID coder"
        clientSecret : "<client secret code>"
        realm: "Okta"
        hostname: "<hostname>"
        primary: true
      - discoveryUrl: "https://dev-<tenantid>.oktapreview.com/oauth2/<resource-server-id>/.well-
known/openid-configuration"
        primary: false
        realm: "Okta"
        hostname: "<hostname>"
        claimsMapping:
          client_id: ["client_id", "cid"]
```

Use Helm to apply the configuration in your *values.yml* file to your Kubernetes cluster:

```
$ helm upgrade qsefe qlik/qsefe -f values2.yml
```

Configuring certificates in your Qlik Sense Enterprise for elastic deployment

By default, Qlik Sense Enterprise for elastic deployments is installed with a self-signed certificate that will not be trusted by users browsers. To replace this with a SSL certificate that you own, complete the steps below.

> *In this example, the certificate is in a file called tls.crt and the associated private key is in a file called tls.key.*

Create the secret resource in Kubernetes

1. Create a file called secret.yaml which will hold the certificate and its key. See the yaml definition below for an example:
   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: my-certificate
     namespace: default
   type: kubernetes.io/tls
   data:
     tls.crt: xxxxxxxxxxxxxxxxxxxx
     tls.key: xxxxxxxxxxxxxxxxxxxxxx
   ```
2. You can give the name field a meaningful name. In this example we've use my-certificate.
   The tls.crt field is the base64 encoded value of your certificate. You can get this value using the following command:
   ```
   $ cat tls.crt | base64
   ```
3. The base64 decoded value will be displayed on the screen. Enter it for the tls.crt value in your .yaml file.
4. Do the same for the tls.key:
   ```
   $ cat tls.key | base64
   ```
5. Enter the resulting base64 value in your .yaml file.
6. Now create the secret resource in Kubernetes using the following command:
   ```
   $ kubectl apply -f secret.yaml
   ```
7. You can verify the secret has been created using the following command:
   ```
   $ kubectl get secret my-certificate
   ```

Configure the Ingress to use the Certificate

1. Configure the Qlik Sense ingress to use the secret created in the previous procedure by adding the following to your values.yaml file:
   ```
   # References the “my-certificate” secret created within the “default” namespace
   nginx-ingress:
     controller:
       default-ssl-certificate: “default/my-certificate”
   ```
2. Update your cluster using the following command:
   ```
   $ helm upgrade --install qsefe qlik/qsefe -f values.yaml
   ```

Verifying the Certificate with your Browser

1. Using your browser, go to the domain you configured to verify the certificate presented by Qlik Sense’s ingress controller.

Viewing logs in Qlik Sense Enterprise for elastic deployments

All services in Qlik Sense Enterprise for elastic deployments emit log data that can be used for debugging issues and activity. Logs can be read on demand or they can be collated and pushed to a log aggregation products for further analysis and use.

Viewing service logs

To inspect the recent logs of a service, for example to debug an issue, the Kubernetes CLI (or other Kubernetes management tools) can be used to quickly view log data.

The following assumes you have the **kubectl** tool installed and connected to your Kubernetes cluster.

Run the following to get a list of all the services running, this will also list if any services are reporting themselves as having issues.

```
Kubectl get pods
```

Identify the service you want to inspect the logs for from the list and run the following adjusting as needed.

```
Kubectl log qsefe-engine-dhwksfhf
```

This will render the recent log entries to the console in JSON format.

Collating and forwarding logs

The logs produced can be forwarded to be gathered, stored, searched and viewed all the system logs on mass in log aggregation tools.

Below is an example of using 3rd party tools including:

- Gathering your system logs in **fluentd**
- Storing your log files in **Elasticsearch**

  > *Elasticsearch requires a significant amount of resources and is therefore not recommended to be executed on your local machine unless your Kubernetes cluster has a lot of available memory and CPU.*

- Consuming your log files in **Kibana**

Installing Elasticsearch

Elasticsearch is a search engine that provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

In this example we install a minimum setup of **Elasticsearch**, that does not include any persistence.

1. Create a file named **elasticsearch.yaml** to configure your installation preferences, and add the following:
   ```
   image:
     tag: "6.1.4"

   client:
     replicas: 1
   ```

```
  resources:
    limits:
      cpu: "0.5"
      memory: "1024Mi"  ## not setting a limit here can take down the cluster using all
available memory
    # requests:   # use defaults
    #   cpu: "25m"
    #   memory: "512Mi"

master:
  persistence:
    enabled: false
  replicas: 2
  # heapSize: "512m"    ## use default, should be less than request, MUST be less than limit
  resources:
    limits:
      cpu: "0.5"
      memory: "1024Mi"  ## set a limit
    # requests:   # use defaults
    #   cpu: "25m"
    #   memory: "512Mi"

data:
  persistence:
    enabled: false
  replicas: 1
  heapSize: "512m"
  resources:
    limits:
      cpu: "0.5"
      memory: "1024Mi"
    requests:
      cpu: "25m"
      memory: "512Mi"
```

2. Run the following command to install **Elasticsearch**:
   ```
   helm upgrade --install elasticsearch incubator/elasticsearch -f ./elasticsearch.yaml
   ```

Installing fluentd

Fluentd is an open source data collector for unified logging layer. It allows you to unify data collection and consumption for a better use and understanding of data. Follow these steps to install **fluentd**.

1. Create a file named **fluentd.yaml** to configure your installation preferences, and add the following:
   ```
   elasticsearch:
     host: elasticsearch-elasticsearch-client
   ```
2. Run the following command to install **fluentd**:
   ```
   helm upgrade --install fluentd incubator/fluentd-elasticsearch -f fluentd.yaml
   ```

Installing Kibana

Kibana lets you visualize your **Elasticsearch** data and navigate the Elastic Stack. You can use it to view and search your logs. Follow these steps to install **Kibana**.

1. Create a file named **kibana.yaml** to configure your installation preferences, and add the following:
   ```
   env:
     ELASTICSEARCH_URL: http://elasticsearch-elasticsearch-client:9200
   ```
2. Run the following command to install **Kibana**:
   ```
   helm upgrade --install kibana stable/kibana -f kibana.yaml
   ```

Accessing Kibana

Run the following command to access **Kibana**:

```
export POD_NAME=$(kubectl get pods --namespace default -l "app=kibana,release=kibana" -o jsonpath="
{.items[0].metadata.name}")
echo "Visit http://127.0.0.1:5601 to access Kibana"
kubectl port-forward $POD_NAME 5601
```

In **Kibana** you can run the following query to test your setup:

```
kubernetes.container_name:engine
```

Configuring MongoDB in Qlik Sense Enterprise for elastic deployments

Qlik Sense Enterprise for elastic deployments uses **MongoDB** as a database for persisting content for several services (excluding Qlik Sense app files).

By default, a pre-configured **MongoDB Community Edition** is added during the installation of Qlik Sense Enterprise for elastic deployments. This is only intended to be used for quick start, testing and evaluation purposes. If you use this version, your MongoDB data may be lost if the Kubernetes cluster is updated.

You can set up a production-ready **MongoDB** environment in the following ways:

- Deploy a separate MongoDB server or cluster along-side Qlik Sense.
- Use a MongoDB DBaaS provider (such as **MongoDB Atlas** or **mlab**)

Configuring the MongoDB connection

When installing Qlik Sense Enterprise for elastic deployments you can specify your MongoDB connection as follows:

- A parameter in the `helm install` command.
- Referencing the connecting settings in a **values.yaml** and using this in the `helm install` command.

Using CLI paramaters

You can extend the basic `helm install` command by setting the following properties:

- Set the **devMode.enabled** value to `false` to disable development mode.
- Set the **mongodb.uri** value with the connection string to MongoDB.

Example:

```
helm upgrade \
    --install qsefe qlik/qsefe \
    --set mongodb.uri=<your-connection-string>,engine.acceptEULA="yes"
```

Referencing values.yaml

Create the **values.yaml** file and include the settings you want to reference in the `helm install` command.

- Set the **devMode.enabled** value to `false` to disable development mode.
- Set the **mongodb.uri** value with the connection string to MongoDB.

Example: values.yaml

```
engine:
  acceptEULA: "yes"

devMode:
  enabled: false
mongodb:
  uri: "<your-connection-string>"

identity-providers:
  secrets:
    idpConfigs:
      - <your IdP configuration here>
```

The values.yaml file is then referenced in the `helm install` command:

```
helm upgrade \
  --install qsefe qlik/qsefe \
  -f values.yaml
```

Monitoring metrics in Qlik Sense Enterprise for elastic deployments

All Qlik Sense Enterprise for elastic deployments services expose metrics that can be used to monitor activities, health and performance data.

The data can be surfaced and collated using open source components. The example below shows how to use Prometheus and Grafana to scrape and analyze metrics in real time.

Viewing metrics with Prometheus

Prometheus is a system monitoring and alerting toolkit that can be used for scraping and storing metrics. It collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.

Prometheus finds the metrics by looking for Kubernetes annotations that have been added to the services.

```
prometheus.io/port=8080
prometheus.io/scrape=true
```

Installing the Prometheus chart

Run the following command to install the **stable/prometheus** chart.

> *Adjust the configuration of your cluster settings, such as RBAC.*

```
helm upgrade --install prometheus stable/prometheus --
set=rbac.create=true,alertmanager.enabled=false,pushgateway.enabled=false
```

Viewing the metrics

View the metrics with the following command:

```
export POD_NAME=$(kubectl get pods --namespace default -l
"app=prometheus,release=prometheus,component=server" -o jsonpath="{.items[0].metadata.name}")
echo "Visit http://127.0.0.1:9090 to access prometheus"
kubectl port-forward $POD_NAME 9090
```

Viewing metrics with Grafana

Grafana is another tool for monitoring and analyzing metrics.

Installing Grafana

Run the following command to install Grafana:

```
helm upgrade --install grafana stable/grafana -f grafana.yaml
```

The example YAML file referenced in the command above provides the following abilities:

- Configure Grafana to look at Prometheus metrics.
- Preload a GO Services dashboard for exposing Golang metrics.
- Preload a Kubernetes dashboard with general metrics.
- Preload a Kubernetes container details dashboard with more specific POD metrics.

See the Online help for full code example.

Viewing the metrics

Run the following command to retrieve your admin user password:

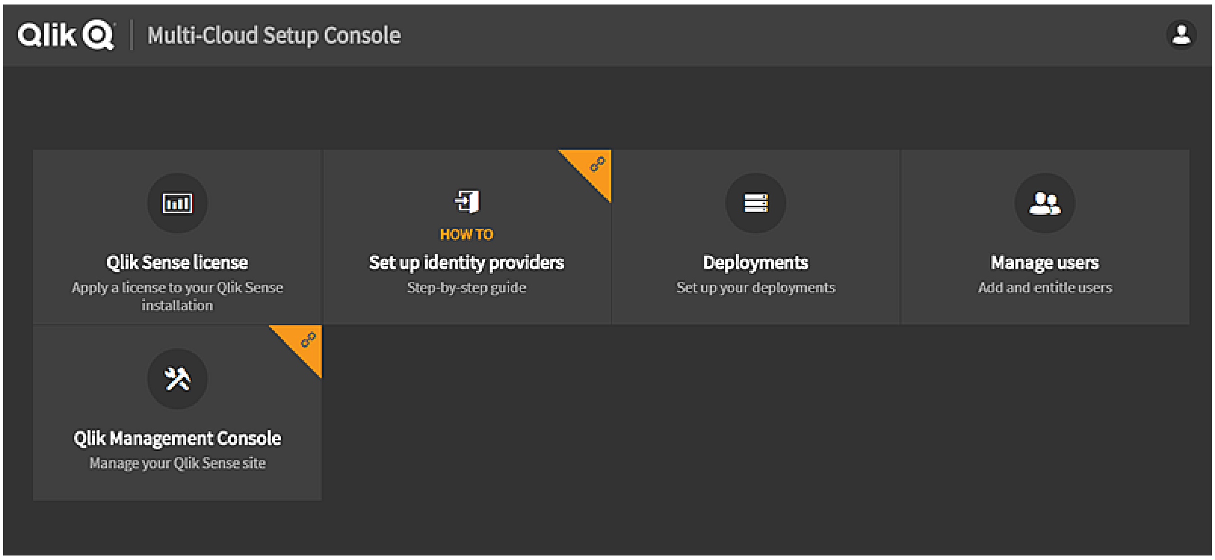
```
kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" | base64 --decode
; echo
```

In the same shell, run the following command to retrieve the Grafana URL:

```
export POD_NAME=$(kubectl get pods --namespace default -l "app=grafana,release=grafana" -o jsonpath="
{.items[0].metadata.name}")
echo "Visit http://127.0.0.1:3000 to access grafana"
export GRAFANA_PASSWORD=$(kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-
password}" | base64 --decode ; echo)
echo "Login as admin:$GRAFANA_PASSWORD"
kubectl port-forward $POD_NAME 3000
```

Multi-Cloud Setup Console - start page

The Multi-cloud Setup Console is where you set up and configure your Qlik Sense multi-cloud solution. To open the Multi-cloud Setup Console, go to *https://<server name>/api/msc*.



The tiles from left to right represent the major steps when setting up Qlik Sense Enterprise for Windows , with Multi-Cloud.

Before you start setting up your multi-cloud, make sure you have the following:

- a valid Qlik Sense multi-cloud license
- the required data for the IdP configuration
- the required data for the deployment settings

To set up your multi-cloud environment:

1. Click the tile **Qlik Sense license** to fill in the site license details.

   If you want to set up Qlik Cloud Services or Qlik Sense Enterprise for elastic deployments, please contact your Qlik representative or Qlik Support to obtain a valid license for the setup.

2. Click the tile **Set up identity providers** to open the IdP setup instructions for Qlik Sense Enterprise for elastic deployments. For QSE for elastic deployments, the IdP setup is done via a configuration file. For Qlik Cloud Services, setup is done in forms on *qlikcloud.com* that are submitted to Qlik.
3. Click the tile **Deployments** to open the deployments settings page. The settings page is the same for QSE for elastic deployments and Qlik Cloud Services. You get setup values from your IdP provider.
4. Click the tile **Manage users** to open the page for user related issues.
5. Click the tile Qlik Management Console to open the QMC to make deployment edits.

Qlik Sense Enterprise for elastic deployments - IdP settings

Configure your IdP for Qlik Sense Enterprise (QSE) for elastic deployments using a YAML configuration file.

For multi-cloud, you can only use identity providers that are compatible with OpenID Connect (OIDC).

Setting up the IdP for Qlik Sense Enterprise for elastic deployments

Before setting up the IdP for QSE for elastic deployments, you must:

- Create a Kubernetes environment.
- Install the client tools to interact with your Kubernetes environment.
- Deploy QSE for elastic deployments into Kubernetes.
- Accept the EULA for QSE for elastic deployments.
- Configure your MongoDB connection.

For a detailed description of the steps, see Qlik Sense multi-cloud deployments with Qlik Sense Enterprise for elastic deployments.

You set up the (IdP) for QSE for elastic deployments in the YAML configuration file.

Do the following:

1. Open your YAML configuration file, and enter the Discovery endpoint.
   Also know as *Discovery URL*.
2. Enter IdP Client ID.
   This is the ID of the configured client at the IdP for interactive user authentication.
3. Enter Client secret.
   The secret for the client configured at the IdP.
4. Enter User ID claims mapping.
   The claim to use as User ID.
5. Enter Groups claim mapping.
   The claim to use as containing groups.

Code example with simple-oidc-provider

This code example is only intended for testing and must not be used in production.

The simple-oidc-provider supports OIDC discovery, which simplifies the main configuration. Use `discoveryUrl`.

```
{
"discoveryUrl": "http://oidc:9000/.well-known/openid-configuration",
"clientId": "foo",
"clientSecret": "bar",
"realm": "simple",
"hostname": "myhost",
"claimsMapping": {
"sub": [ "sub", "client_id" ]
}
}
```

The simple-oidc-provider does not return a `sub` claim for client credentials tokens. The remedy for this is the claims mapping `"sub": [ "sub", "client_id" ]`. This will map the `sub` claim to the `sub` claim whenever possible but will fall back on the `client_id` claim.

Logging out from the multi-cloud environment

When you log out from the multi-cloud environment, you may see an almost blank page, with only an **OK** in the top left corner. This can be the default page for the identity provider for your tenant. The page is configurable for your identity provider.

MSC - Deployments

With a multi-cloud setup, you can either deploy to Qlik Sense Enterprise for elastic deployments or Qlik Cloud Services. Qlik Sense Enterprise for elastic deployments supports deployment to public or private clouds on a customer managed infrastructure. Qlik Cloud Services supports deployment on an infrastructure managed by Qlik. Currently, you can set up one of each of these deployments.

The setup differs depending on whether or not you use a local bearer token.

Setup with a local bearer token

A local bearer token simplifies the deployment setup. Instead of using the token endpoint, client ID, and client secret properties to retrieve a bearer token from the IdP, a bearer token is generated locally.

Before you start setting up your deployment, make sure you have the API endpoint, provided by Qlik in your welcome email.

To set up your deployment:

1. In the bottom left corner, click **Set up new**.
2. Enter a deployment name.
3. Enter the API endpoint, which is sent to you from Qlik.
4. Enter audience: `qlik.api`. Audience is needed by the app distribution service to get API tokens from cloud.
5. Select **Use local bearer token** and click **Apply**.
   The local bearer token for elastic deployments settings are displayed.
6. By default, the **Qlik Cloud Services format** check box is selected. The text box then displays the IdP definition. If you clear the check box selection, the IdP definition is displayed in regular text. Use this format when you deploy to Qlik Sense Enterprise for elastic deployments.
7. Choose the format you want to use and click **Copy to clipboard** to save the text. You need this text when you configure your tenant.
8. For QCS, you paste the IdP definition in the **Paste local bearer** text box on the tenant configuration page.
   For QSE for elastic deployments, you paste the IdP definition in the *values.yaml* file.

Setup with IdP integration

Before you start setting up your deployment, make sure you have the following:

- Client ID and client secret (collected from your IdP provider)
- Token endpoint
- API endpoint (provided by Qlik in your welcome email)

To set up your deployment:

1. In the bottom left corner, click **Set up new**.
2. Enter a deployment name. (You can use this name in distribution policies for the distribution of apps.)
3. Enter Client ID and Client secret.
4. Enter Token endpoint, also known as *Authentication URL*.
5. Enter API endpoint, which is sent to you from Qlik.
6. Enter audience: `qlik.api`. Audience is needed by the app distribution service to get API tokens from cloud.

3.15 Setting up Qlik Sense after installation

This section guides you through the process of setting up your Qlik Sense site after installing. You can configure the server to fit with your organization’s particular needs. Below are the common task most deployments will require.

Connecting Qlik Sense to your user directory

Qlik Sense has a range of methods for authenticating users. Windows authentication is the default method.

When a user connects to Qlik Sense for the first time a user record is created to identify that user. Once this record is created, the administrator can track the user’s activity and assign her a license and permissions.

Administrators can also connect to a user directory (for example, Active Directory or LDAP) to obtain further information about that user (such as user groups). The user information can be fetched in advance and then kept in sync with the user directory. This is optional but recommended since it will provide you with the best management experience.

See the following sections in the *Manage Qlik Sense sites* guide:

- Setting up a user directory connector and schedule by task
- Managing users

Assigning licenses to users

Users need a license to open an app.

There are two license types: one that is user-based and one that is token-based.

- User-based license: you can allocate professional access and analyzer access. The LEF determines the distribution of the two access types.
- Token-based license: you can allocate user access and login access. The LEF determines the number of tokens that you can allocate to the two access types.

User-based licenses

User-based licenses grant a predefined number of professional and analyzer access allocations. The distribution of the access types is determined by the LEF.

Manage Qlik Sense sites: Managing licenses

Manage Qlik Sense sites: Managing professional access

Manage Qlik Sense sites: Managing analyzer access

See the following sections in the *Manage Qlik Sense sites* guide:

- Managing licenses
- Allocating professional access
- Allocating analyzer access

Token-based licenses

Qlik Sense offers a token-based system that allows the administrator to assign the most suitable license type to each user. Licenses can be allocated on an individual basis or automatically by using rules to define who is allowed to obtain a license, for example, all users in a specific department.

See the following sections in the *Manage Qlik Sense sites* guide:

- Managing licenses
- Allocating user access
- Creating new login access

Configuring the monitoring apps

All installations of Qlik Sense (single node and multi-node) require configuration of the monitoring apps for them to work properly.

See the following sections in the *Monitor Qlik Sense sites* guide:

- Configuring the monitoring apps

How Qlik Sense uses HTTPS and certificates

Qlik Sense deploys securely by default when it is installed. It uses self-sign certificates to ensure that data is transferred to users in a secure way. When users access the system they by default receive a warning that the certificate used by the site is not trusted. The user can then accept the certificate and proceed to use Qlik Sense securely.

There are two options to prevent the warning; one is to use a trusted certificate from either a trusted provider or an internal corporate source. The other option is to run the site using HTTP only. Both options are available as settings on the Qlik Sense Proxy Service (QPS).

Regardless of which option you choose, the services in Qlik Sense always use encryption when communicating.

See the following sections in the *Manage Qlik Sense sites* guide:

- Changing proxy certificate
- Editing proxies

Creating and opening apps

To create and open apps on the server users must browse to the Qlik Sense hub using their web browser. The hub lists two areas: **Work** contains the apps belonging to the user who has logged in, and **Streams** contains the other apps the user has access to. After the installation the administrators see two built-in monitoring apps, while all other users do not see any apps. Click on an existing app to open it. To create a new app, click **Create new app**.

Working with streams, apps and publishing

A stream is a way to group together apps that have similar permissions. Once an app has been created, an administrator can publish it to a stream. The app then becomes visible to users who have access to that stream. Apps, streams and publishing are managed in the Qlik Management Console (QMC).

See the following sections in the *Manage Qlik Sense sites* guide:

- Managing streams
- Managing apps
- Publishing app

See also:

🗋 The following section in the *Manage Qlik Sense sites* guide: Managing nodes and services

4 Qlik Sense Mobile

The Qlik Sense Mobile app allows you to securely connect to your Qlik Sense Enterprise deployment from a supported mobile device. The Qlik Sense Mobile app can be deployed and managed using Enterprise Mobile Management (EMM) software.

For more information about deploying and managing Qlik Sense Mobile, see *Deploying Qlik Sense Mobile (page 146)*.

4.1 The Qlik Sense Mobile app

The Qlik Sense Mobile app can be installed on supported devices running compatible versions of iOS or Android OS, and connected to a Qlik Sense Enterprise deployment.

For a detailed list of devices, OS versions, and Qlik Sense versions supported, see *System requirements for Qlik Sense (page 12)*

The Qlik Sense Mobile app connects to a Qlik Sense Enterprise hub. When connected, you can view and consume Qlik Sense apps and mashups available on the Qlik Sense Enterprise installation. Qlik Sense Mobile for iOS devices supports offline access to Qlik Sense apps. You can download the Qlik Sense apps for use offline when no internet connection is available. The Qlik Sense administrator controls which apps are available to download for offline use, using the QMC. See: Creating security rules.

The download of Qlik Sense apps for offline access is currently supported only on Qlik Sense Mobile app for iOS.

Developing Qlik Sense apps offline using the Qlik Sense Mobile app is not currently supported.

When you log into the Qlik Sense Mobile app for the first time, you must authenticate your credentials against the Qlik Sense Enterprise server. For more information, see *Connecting to Qlik Sense from the Qlik Sense Mobile app (page 158)*. Once you have authenticated your credentials and logged in to the app, you may choose to have the Qlik Sense Mobile app remember your credentials. To protect your data, ensure that the device is protected by a password and locked when not in use. For more information, see *Qlik Sense Mobile security (page 145)*.

4.2 Enterprise Mobile Management (EMM) and Qlik Sense Mobile

Using a supported EMM, you can remotely deploy and manage the Qlik Sense Mobile app on all of your organization's supported mobile devices. Using an EMM console you can:

- Distribute the Qlik Sense Mobile app to mobile devices.
- Configure the Qlik Sense hub list in the Qlik Sense Mobile app.
- Configure the certificate validation policy.

For more information about configuring the certificate validation policy, see *Configuring the certificate validation policy for the Qlik Sense Mobile app (page 146)*.

For more information about deploying and managing Qlik Sense Mobile with AirWatch, see *Deploying the Qlik Sense Mobile app using AirWatch (page 147)*.

4.3 Qlik Sense Mobile security

Qlik Sense Mobile connects to a Qlik Sense Enterprise hub. When you are connected, you can view Qlik Sense apps and mashups, and download Qlik Sense apps using the Qlik Sense Mobile app.

The download of Qlik Sense apps for offline access is currently supported only on Qlik Sense Mobile app for iOS.

Authentication

When you log into the Qlik Sense Mobile app for the first time, you must authenticate your credentials against the Qlik Sense Enterprise server. Once you have authenticated your credentials, and logged in to the Qlik Sense Mobile app, you may choose to have the Qlik Sense Mobile app remember your credentials. For more information about how to set the options for credentials management, see Logging in to the Qlik Sense Mobile app. To protect your data, ensure that the device is protected by a password and locked when not in use. This can be configured through your Enterprise Mobile Management (EMM) console.

The Qlik Sense Mobile app can be used offline for up to 10 days (240 hours). This time period starts when the Qlik Sense Mobile app is first launched following the last log in to the Qlik Sense Enterprise server. When the 10 day period expires, you must to log back into the Qlik Sense Enterprise server to continue using the Qlik Sense Mobile app.

Section access in the data load script can also be used for security. A single file can be used to hold the data for a number of users or user groups. Qlik Sense then uses the information in the section access for authentication and authorization on the Qlik Sense Enterprise server, and dynamically reduces the data, so that users only see their own data. The security is built into the file itself, which means downloaded files are also protected. For more information, see Managing security with section access.

Certificates

When Qlik Sense is deployed over SSL, the Qlik Sense Mobile app obtains a certificate from the Qlik Sense server and verifies that it is valid. This allows the Qlik Sense Mobile app to trust that the server it is talking to is a legitimate Qlik Sense server. The Qlik Sense Mobile app will always reject the certificate if it is not valid. Every Qlik Sense hub that you add to the hub list must therefore have a valid certificate.

To ensure that a certificate is valid, you need to check that the certificate:

- Is signed by a certificate authority, such as VeriSign, or signed by a certificate authority that has been added to the list of trusted certificate authority for the device (either manually added to the device or pushed to the device from an EMM console).

- Is not expired.
- Has a common name or a name that matches the domain name of the Qlik Sense hub.

Configuring the certificate validation policy for the Qlik Sense Mobile app

The certificate validation policy applies when Qlik Sense is deployed over SSL, and the Qlik Sense Mobile app encounters invalid certificates from a Qlik Sense server that has been added to the hub list by the device user.

You can configure the certificate settings from your EMM console.

Do the following:

1. Make sure the Qlik Sense Mobile app has been installed on the device.
2. If the Qlik Sense server has a certificate that is not signed by a trusted certificate authority, make sure that the certificate that was used to sign the server certificate is added to the list of trusted certificate authorities for the device either manually or using your EMM.
   a. Configure the certificate from your EMM console.
   If your EMM console does not have this functionality, you can use this software to make the edits and then upload the changes to your EMM console:
   Apple Configurator
   The available settings are:
      - **Ask user**
      - **Always allow**
      - **Always reject**

   b. Push the changes to the device.

4.4 Deploying Qlik Sense Mobile

The Qlik Sense Mobile app can be downloaded and installed directly from the Apple App Store or Google Play Store. The Qlik Sense Mobile app includes a Qlik Sense demo server that is hosted by Qlik, and allows you to view and download apps. You can connect to the Qlik Sense demo server without Qlik Sense Enterprise account credentials. To connect the Qlik Sense Mobile app to your Qlik Sense Enterprise deployment, your Qlik Sense administrator must configure the connection and deploy to users.

The download of Qlik Sense apps for offline access is currently supported only on Qlik Sense Mobile app for iOS.

Qlik Sense Mobile can be deployed and managed using either Enterprise Mobile Management (EMM) software, or Apple Developer Enterprise Program tools.

To deploy using Apple Developer Enterprise Program tools, you must be a member of the Apple Developer Enterprise Program. For more information about deploying using Apple Developer Enterprise Program tools, see the Apple Developer Enterprise Program documentation.

Using a supported EMM, you can remotely deploy and manage the Qlik Sense Mobile app on all of your organization's mobile devices. From an EMM console you can:

- Distribute the Qlik Sense Mobile app to mobile devices.
- Configure the Qlik Sense hub list.
- Configure the certificate validation policy.

  For more information about configuring the certificate validation policy, see *Configuring the certificate validation policy for the Qlik Sense Mobile app (page 146)*.

Qlik Sense Mobile and VPP

Qlik Sense Mobile can be deployed using the Apple Volume Purchase Program (VPP).

The Apple Volume Purchase Program (VPP) is a service that allows registered organizations to purchase iOS apps in bulk. After making a bulk purchase, the organization receives redemption codes for each app bought. Those app codes can then be distributed to individual users, who use the codes to download public apps from the Apple App Store. Codes can be distributed to users through email, a web portal, or Enterprise Mobile Management (EMM) software. Instead of pushing software and profiles out to devices, organizations can push licenses to devices while downloading apps directly from the Apple App Store.

Volume-purchased software and licenses can be distributed to users in the following ways:

- Email redemption URLs directly to users, which allows them to download and install the app.
- Post redemption URLs on an enterprise-hosted web page that is accessible only to authorized users.
- Use the Apple Configurator utility to push redemption codes to local connected devices.

  Note that this method is only recommended for small work groups.

- Push redemption codes to remotely managed devices using EMM software to push redemption codes to remotely managed devices.

The Apple Volume Purchase Program allows businesses and schools to retain ownership of purchased apps, so apps can be reclaimed and redistributed as needed.

Deploying the Qlik Sense Mobile app using AirWatch

The Qlik Sense Mobile app can be deployed using AirWatch. To deploy using AirWatch, add the app to your AirWatch Catalog. Once the app is added to your AirWatch Catalog, you can choose to either push the app directly to your users' devices, or allow them to install the app manually.

To deploy the app using AirWatch:

1. Open your AirWatch Management Console.
2. Go to **Apps & Books** > **Applications** > **List View** > **Public** and select **Add Application**.
3. Select the **Platform**.
4. Select **Enter URL** and enter the URL to download the Qlik Sense Mobile app.
5. Click **Next**.
6. Configure options on the **Details** tab.
7. Assign the application to smart groups on the **Assignment** tab.
8. Configure the **App Delivery Method**:
   - On Demand - Deploys the app to a catalog and lets the user decide if and when to install it.
   - Automatic - Deploys the app to a catalog on a device upon enrollment. After the device enrolls, the user is prompted to install.
9. Select **Send Application Configuration** if you want to populate the Qlik Sense Mobile app with links to your Qlik Sense hub.
10. Assign a **Required Terms of Use** for the application on the **Terms of Use** tab.
11. Select **Save & Publish** to view the device assignment page that lists the assigned devices.
12. Select **Publish** to deploy the application.

For details about how users download and install the app manually using AirWatch, see *Connecting to Qlik Sense using AirWatch (page 150)*.

Qlik Sense Mobile and per-app VPN support

The Qlik Sense Mobile app supports per-app VPN tunneling when deployed using AirWatch on iOS 11.2.2 or later.

Per-app VPN functionality, provides endpoint security by limiting connections at the application level, instead of at a device level. The VMware Tunnel restricts app access to white-listed domains, and specific databases that white-listed domains can access.

The following are the current minimum requirements for AirWatch support of iOS 11:

- AirWatch Agent version 5.5.1
- VMware Tunnel version 2.0.4

To enable per-app VPN tunneling support for Qlik Sense Mobile in AirWatch you will need to customize your VMware Tunnel configuration. For more information, see *Configuring AirWatch for per-app VPN (page 148)*.

Configuring AirWatch for per-app VPN

The Qlik Sense Mobile app supports per-app VPN tunneling when deployed using AirWatch. To enable per-app VPN tunneling, you must:

- Customize your VMware Tunnel server configuration.
  To enable per-app VPN to work with the Qlik Sense Mobile app, you must disable Datagram Transport Layer Security (DTLS).

  > *This may impact other applications that use User Datagram Protocol (UDP).*

- Add network traffic rules so that the VMware Tunnel bypasses Qlik Sense Mobile local network traffic.

Customize your VMware Tunnel server configuration

If your VMware Tunnel server was deployed using a Linux installer, complete the following steps to disable DTLS so that the VMware Tunnel server uses TLS for all traffic.

> *If your VMware Tunnel server was deployed using the AirWatch Unified Access Gateway (UAG) virtual device, there is currently no workaround to disable DTLS, and per-app VPN cannot be configured for use with the Qlik Sense Mobile app.*

Do the following:

1. Log in to your VMware Tunnel server using Secure Shell or ssh.
2. Navigate to `/opt/airwatch/tunnel/vpnd/server.conf`.
3. Add `dtls_channel 0`.
4. Restart the vpn server using `sudo service vpnd restart`.

Add VMware Tunnel rules

Do the following:

1. Open your AirWatch Management Console.
2. From the **Settings** menu, go to **System** > **Enterprise Integration** > **VMware Tunnel** > **Network Traffic Rules**.
3. On the **Device Traffic Rules** tab, add the following rules:

Rank	Application	Action	Destination Hostname
1	Qlik Sense Mobile-iOS	Bypass	127.0.0.1
2	Qlik Sense Mobile-iOS	Tunnel	*

Once you have configured AirWatch for per-app VPN support of the Qlik Sense Mobile app, you can proceed with your deployment. For more information, see *Deploying the Qlik Sense Mobile app using AirWatch (page 147)*.

Configuring the Qlik Sense Mobile app hub list using AirWatch

When you deploy the Qlik Sense Mobile app using AirWatch, you can choose to push the link to the Qlik Sense Enterprise hub directly to your users using AirWatch.

Do the following:

1. Open your AirWatch Management Console.
2. Go to **Apps & Books** > **Applications**.
3. Select the **Qlik Sense Mobile** app.
4. Click **Assign**.
5. Select the radio button for the group that you want to deploy the application configuration file to and click **Edit**.
6. On the **Add Assignment** page, expand the **ADVANCED** section, then expand the **APPLICATION CONFIGURATION** section.
7. In the **Configuration Key** field, enter *mdm*.
8. Ensure that the **Value Type** is set to **String**.
9. In the **Configuration Value** field, enter name and URL for each Qlik Sense Enterprise hub in the following format:
   ```
   { "Accounts" : [ {"name":"Account 1","url":"http://www.ahub.com"},
   {"name":"Account 2","url":"http://www.asecondhub.com"} ] }
   ```
10. Click **Add**.
11. Click **Save & Publish**.
12. Click **Publish**.
13. Go to the **More** menu and select **Send Application Configuration**.

The Qlik Sense Enterprise hubs that you added will appear in the your users' Qlik Sense Mobile app list under **Select an account** the next time that they open the app.

Connecting to Qlik Sense using AirWatch

To connect to Qlik Sense from a mobile device using AirWatch per-app VPN, you must:

- Download the AirWatch Agent app
- Register the device
- Install a supported app or browser

> *Connecting to Qlik Sense from a mobile device using AirWatch per-app VPN is currently supported only on Apple iOS devices.*

Do the following:

1. Download the AirWatch Agent app from the Apple App Store.
2. Open AirWatch Agent and enroll using one of the available options:
   - Email address
   - Server details
   - QR code
3. On the **Authenticate** screen, enter your **user name** and **password** and select **Next**.

4. On the **Secure** screen, select **Redirect & Enable** to enable management of your device by installing the Device Manager configuration profile. You are redirected to Safari and asked for the permission to open Settings. Select **Allow**.
5. In Settings, select **Install** to install the Device Manager configuration profile, and then select **Trust** to confirm that you allow your device to be enrolled into remote management.
6. Once the installation of the Device Manager configuration profile is complete, select **Done**. You will be redirected to Safari, and then to AirWatch Agent where a **Configure** screen confirms that the authentication procedure is complete. Select **Done**.

   *If a pop-up appears asking to install VMware Tunnel, select **Install** to allow the installation of the VMware Tunnel app. If the pop-up does not appear, you can install VMware Tunnel from AirWatch Catalog. See step 9.*

7. In AirWatch Agent you can now manage your enrolled devices in the **My Device** portal.

   You may be asked to create a device passcode to access AirWatch Agent. The passcode will be required every time you access the app. If you already have a passcode configured on your device you can enter it here to maintain the same passcode. If you enter a new passcode here it will overwrite your existing device passcode.

8. Close AirWatch Agent.
9. If you haven't installed VMware Tunnel already, open AirWatch Catalog, which has now been added to your device, and install it.
10. Open VMware Tunnel app and select **Continue** to enable it.
11. Open AirWatch Catalog and install the Qlik Sense Mobile app or one of the supported browsers.
    For a list of mobile browsers that support the connection to Qlik Sense Enterprise through AirWatch per-app VPN, see System requirements for Qlik Sense.

    The Qlik Sense Mobile app allows you to download Qlik Sense apps for offline use.

    Your AirWatch Agent administrator may have already populated the hub list with your Qlik Sense server connection.

12. To connect to Qlik Sense for the first time using the Qlik Sense Mobile app, see *Connecting to Qlik Sense from the Qlik Sense Mobile app (page 158)*.

Deploying Qlik Sense Mobile with Microsoft Azure and Intune

The Qlik Sense Mobile app can be deployed using Microsoft Azure and Intune. Some configuration changes are required in the Microsoft Azure portal to enable Single Sign On (SSO) and Intune management of Qlik Sense Mobile.

Before you begin:

- Azure AD Connect must be configured to replicate your primary domain (Active Directory) and the Azure Portal (Azure Active Directory).
- Azure AD Application Proxy Connector must be installed and configured.

To deploy the app using Microsoft Azure and Intune:

- Set up a Qlik Sense Enterprise virtual proxy
- Set up Kerberos constrained delegation in Active Directory
- Add an Azure enterprise application for Qlik Sense Enterprise virtual proxy
- Add an Azure app registration for Qlik Sense Mobile
- Add the Qlik Sense Mobile app to the Intune **Company Portal**
- Define a Qlik Sense Mobile app protection policy
- Define a Qlik Sense Mobile configuration policy
- Deploy the Qlik Sense Mobile app

Set up a Qlik Sense Enterprise virtual proxy

1. Open the Qlik Management Console on the Qlik Sense Enterprise server.
2. Go to **Proxies** > **Central Proxy**.
3. Enable **Kerberos Authentication**.
4. From the Qlik Management Console home page, go to **Virtual Proxies**.
5. Click **Create new Virtual Proxy**.
6. Enter the following information:
   - Identification
   - Authentication
   - Load Balancing
   - Host white list sections

   Note the prefix used, it will be used later in the Azure Portal configuration (https://sense_server_fqdn/prefix).

   The Windows Authentication pattern must be set to Mozilla.

7. Click **Save**.

Set up Kerberos constrained delegation in Active Directory

1. Log in to a server that has access to Active Directory in your primary domain.
2. Open a Windows Power Shell as an administrator.
3. Create a Service Principal Name (SPN) for the Qlik Sense Enterprise installation using the following command:
   setspn.exe -U -S HTTP/sense_server_fqdn domain\sense_server_service_account
4. Open **Active Directory Users and Computer**.
5. Find the computer that hosts the Azure AD App Proxy, to modify the machine properties.
6. Go to the **Delegation** tab and choose **Trust the computer for delegation to specified services only**.
7. Select **Use any authentication protocol** and add the SPN created.
8. Open ADSI, confirm that the Azure AD app proxy host is set to delegate to the Qlik Sense server.

Add an Azure enterprise application for Qlik Sense Enterprise virtual proxy

1. Log in to the Azure portal and select **Azure Active Directory Service**.
2. Select **Application Proxy** and confirm there is at least one active application proxy.
3. Select **Enterprise Applications**.
4. Click **New application**.
5. Select **On-premises application**.
6. Enter a name for the new application.
7. Enter the URL for the server where Qlik Sense Enterprise is installed.

   Include the QSE virtual proxy prefix is in the URL path.
   For example: https://sense_server_fqdn/prefix

8. Setup the **External URL**.

   This will be used later for the App Registration for Microsoft Intune. For example, https://sensekcd-qlikemmnet.msappproxy.net/prefix/.
   Note: The URL consists of a prefix (sensekcd-) followed by your tenant name followed by msappproxy.net followed by the QSE virtual proxy prefix.

9. Ensure that the application is using **Azure Active Directory** for its **Pre-Authentication** method.
10. Ensure that a valid **Connector Group** is selected to direct traffic to the application proxy.
11. Select **Single sign-on properties** for the **Enterprise Application**.
12. Choose **Integrated Windows Authentication** for **Single Sign-on Mode**.
13. Enter the SPN you created earlier.
14. Choose **On-premises user principal name** for **Delegated Login Identity**.
15. Click **Save**.

16. Select the enterprise application you added and click **Properties**.
17. Set **User assignment required** to **Yes**, and click **Save**.

Add an Azure app registration for Qlik Sense Mobile

1. Log in to the Azure portal and select **Azure Active Directory Service**.
2. Select **Apps Registrations**.
3. Click **New Application Registration**.
4. Enter a **Name**.
5. Enter an **App registration type** of **native**.
6. Enter a **Redirect URL** of msauth://com.qlik.qliksense.mobile/17PV4mdIRAc%2F3SeFXILsSWg1aDU%3D.
7. For the **App Registration** click **Settings** and select **Redirect URLs**.
8. Add an additional redirect URL of qliksense-intune://com.qlik.qliksense.mobile and click **Save**.
9. Take note of this app registration's **Application ID**.
10. Add and grant the following delegated permissions:

- Microsoft Mobile Application Management- Read and Write the User's App Management data
- The Web app / API defined above - Access <Web App / API name>
- Microsoft Graph – Read Directory Data
- Windows Azure Active Directory – Sign in and read user profile

Some of these permissions require Admin permissions. The first person to log in to Qlik Sense Mobile must be a user with tenant administration capabilities such that they can consent to the necessary permissions.

Add the Qlik Sense Mobile app to the Intune **Company Portal**

1. Log in to the Azure portal and select the Intune service.
2. Select **Client Apps**.
3. Select **Apps**.
4. Click **Add**.
5. Select an **App type** of **Android Store App** for Android, or **iOS Store App** for iOS.
6. Click **Configure** and enter the following:
   - Name
   - Description
   - Publisher
   - App store URL: enter the link to the Qlik Sense Mobile app on the Apple App Store for iOS devices, or the Google Play Store for Android devices.
   - Minimum operating system
7. Click **OK**.

8. Once the app is uploaded, click **Assignments** and ensure that the appropriate users and devices are assigned to the app.
9. Refresh the list of apps. You should see the new app of type **Managed Android Store App** for Android, or **Managed iOS Store App** for iOS, with an **Assigned value** of **YES**.

Define a Qlik Sense Mobile app protection policy

1. Log in to the Azure portal and select the Intune service.
2. Select **Client Apps**.
3. Select **App protection policies**.
4. Click **Create Policy**.
5. Enter a **Name** and **Description**.
6. Enter a **Platform** of **Android** or **iOS**.
7. Enter a value of **Yes** for **target to all app types**.
8. Click on **Select Required Apps** and select the Qlik Sense Mobile for Android or iOS app added above.

> *For iOS you must add the Qlik Sense Mobile app via its bundle id com.qlik.qliksense.mobile. For Android you add the Qlik Sense Mobile app via its package id com.qlik.qliksense.mobile.*

9. Click **Settings** and configure the various settings, then click **Save**.
10. If the protection policy is configured to limit data transfer from Qlik Sense Mobile then the limitation should be set to **policy managed apps** so that Qlik Sense Mobile can send diagnostics emails.

> *For Android use a browser to display help and use a PDF viewer to display the Qlik Sense Mobile **Terms and Conditions** document.*

> *For iOS protection policy a similar setting is required to allow Qlik Sense Mobile to send diagnostic emails. Help and terms and conditions are displayed within the iOS Qlik Sense Mobile app itself.*

Define a Qlik Sense Mobile configuration policy

1. Log in to the Azure portal and select the Intune service.
2. Select **Client Apps**.
3. Select **App configuration policies**.
4. Click **Add**.
5. Enter a **Name** and **Description**.
6. Select an enrollment type of **Managed Apps** for Android or **Managed Devices** for iOS.
7. Click **Assignments** and assign the appropriate users or user groups.
8. Click **Select the required app** and select the Qlik Sense Mobile app added to the **Company Portal**.
9. Click **Configuration settings** and enter a name of **mdm**.

10. For **Value** enter the json document { "Accounts" : [{"name":"Your server name","url":"<external URL>", "config": { "AADAppId" : "<the Application Id noted above>"} }] }
11. Click **Save**.
12. Ensure that the app configuration shows as assigned with an enrollment type of **Managed apps** for Android, or **Managed devices** for iOS.

Deploy the Qlik Sense Mobile app to Android devices

1. On an Intune enrolled Android device open the **Company Portal** and install Qlik Sense Mobile.
2. Launch Qlik Sense Mobile.
3. You should be prompted to indicate that the app is being managed. If you don't then there is likely a configuration issue with the App protection policy.
4. You should see your Qlik Sense Mobile deployment in the Qlik Sense Mobile server list. If you don't then there is likely a configuration or a user assignment issue.
5. Logging in to Qlik Sense Mobile deployment should follow the Azure SSO login flow.

Deploy the Qlik Sense Mobile app to iOS devices

1. On an Intune enrolled iOS device open the **Company Portal** and install Qlik Sense Mobile.
   Intune will present a dialog asking to manage Qlik Sense Mobile.
2. Click **Yes** or **Manage**.
3. Launch Qlik Sense Mobile.
   You should see the Qlik Sense Mobile server you defined above. If you don't then there is likely a configuration or a user assignment issue.
4. Click on the server and log in using SSO if required.
5. You will see an Intune dialog indicating that the App data is managed. Click **OK**. Qlik Sense Mobile will exit.
6. Logging in to Qlik Sense Mobile deployment should follow the Azure SSO login flow.

Connecting to Qlik Sense using BlackBerry Access

You can access Qlik Sense and consume apps from a mobile device using BlackBerry Access browser. A BlackBerry Access administrator must first set up a BlackBerry Dynamics deployment and configure URL connections to one or multiple Qlik Sense servers. The administrator then registers users and generates secret keys that users must use to access the BlackBerry server from their mobile devices. Once inside the BlackBerry Dynamics environment, users can reach their Qlik Sense hubs.

Configuring BlackBerry Dynamics

Prerequisites

- Qlik Sense Enterprise June 2018 or later must be installed.
- See BlackBerry documentation for BlackBerry Access system requirements.

Allocating access rights

The Qlik Sense administrator must allocate access rights to users in the Qlik Sense Management Console (QMC), before deploying BlackBerry Dynamics.

Do the following:

1. In the QMC, select **License management** on the start page or from the **Start**▼ drop-down menu to display the overview.
2. Select **User access allocations** in the panel to the right.
3. Click ⊕ **Allocate** in the action bar.

   The **Users** dialog opens.
4. Select users in the list and click **Allocate**.

   Allocate is disabled if the number of tokens available for allocation is insufficient for the number of selected users.

   The dialog is closed and the users are added in the User access allocations overview table.

For more on user access, see Managing user access.

Deploying BlackBerry Dynamics

Do the following:

1. **Set up a BlackBerry Dynamics deployment**. Using BlackBerry server services, set up a new BlackBerry Dynamics deployment. Once your deployment is set up, log in using BlackBerry Access administrator credentials to access the BlackBerry Dynamics dashboard.
2. **Register users**. You can synchronize your Dynamics deployment with an active directory to import and update users' information, such as email addresses, in the BlackBerry Dynamic environment. Every user registered in your BlackBerry Dynamics deployment is listed under **USERS, Users and Groups**.
3. **Create apps**. Users use apps to connect to different Qlik Sense servers for which they have been granted access. To create a new app:
   - Go to the section **APPS, Manage Apps** of the Dynamics dashboard.
   - Click on the **Add App** button.
   - Select **Web** from the list of app types. A web app consists of an URL address that users reach using the BlackBerry Access browser.
   - Insert the URL address of the Qlik Sense Enterprise server for which you have allocated access rights to users.

   Further to web apps, in a BlackBerry Dynamics environment you can create Public apps, (apps compatible with BlackBerry EMM environment), Custom apps, or assign GD Entitlement ID. For more information, see BlackBerry Dynamics documentation.
4. **Add apps to App groups**. Apps can be grouped together in App groups. You can create and manage App groups from the Dynamics dashboard under **APPS, App Groups**.
5. **Grant users access to specific apps or app groups**. Under **APPS, App Groups**, you can assign users to one or multiple App groups, therefore granting these users access to the set of apps that are included in a specific App group. You can also set restrictions to prevent users from accessing specific apps or App groups.

You must also grant your users access to the BlackBerry Access browser app.

6. **Assign users to policy sets**. You can assign different policy sets to different users. Users are clustered in groups based on which policy set is assigned to them. A policy set defines access rights and restrictions. Users registered with a certain policy set can access certain apps or App groups, and are prevented from accessing others, if restrictions are applied.
7. **Assign unique access keys to users**. Once a user has been registered with specific policy sets and granted access to apps and App groups, the administrator must generate access keys and provide them to the user.

Generating access keys

An access key is needed to activate the BlackBerry Access browser app and allow the connection to the BlackBerry Dynamics deployment. If the environment is properly set up, users will receive their access keys via email as soon as they are generated. Once an access key is in use, it disappears from the list of available access keys in the BlackBerry Dynamics dashboard. An access key is only active for a limited time. By default, an access key expires after 30 days from the last log-in into the BlackBerry Dynamics environment from the device.

Accessing Qlik Sense using BlackBerry Access

Registered users can use BlackBerry Access on a mobile device and reach Qlik Sense via browser.

Prerequisites

- For a detailed list of BlackBerry Access app, iOS, and Android supported versions, see Supported browsers.
- An access key to enroll your device. If you have not received any access key, contact the BlackBerry Access administrator.

Connecting to Qlik Sense from a mobile device using BlackBerry Access

Do the following:

1. Download the BlackBerry Access app from the Apple App Store or the Google Play Store.
2. Open BlackBerry Access and choose to enroll.
3. Enter the email address which received your access key. Enter the access key you received, and click OK.
4. Create a password when prompted.
5. Once inside the BlackBerry Dynamics environment, select the app, which is the Qlik Sense server URL address, you want to access. You can also type the URL of the Qlik Sense server you want reach in the address bar at the top.
6. Insert your Qlik credentials to access the Qlik Sense hub.

Connecting to Qlik Sense from the Qlik Sense Mobile app

When you install and launch the Qlik Sense Mobile app for the first time you are prompted to select either the Qlik Sense demo server or a Qlik Sense Enterprise server to connect to.

The Qlik Sense demo server is hosted by Qlik, and allows you to view Qlik Sense apps and mashups, and download apps. You can connect to the Qlik Sense demo server without Qlik Sense Enterprise account credentials.

You must connect to the Qlik Sense demo server at least once while online before you can access content while offline.

The download of Qlik Sense apps for offline access is currently supported only on Qlik Sense Mobile app for iOS.

To connect to a Qlik Sense Enterprise server, you must log in with your with your Qlik Sense Enterprise account credentials. Before you can connect to a Qlik Sense server and log in with your Qlik Sense Enterprise account credentials from the Qlik Sense Mobile app you will need to authenticate your credentials against the Qlik Sense Enterprise server.

The Qlik Sense Enterprise authentication link must be generated by your administrator in the Qlik Management Console. Your Qlik Sense administrator will provide you with information about how you can receive the link using one of the following methods:

- Retrieving the authentication link from your Qlik Sense Enterprise hub
- Receiving the authentication link from your administrator

If your Qlik Sense Mobile app is deployed and managed through an EMM, the hub list may already be populated for you, in which case you do not need to complete this procedure.

Retrieving an authentication link from the Qlik Sense Enterprise hub

Do the following:

1. Open your mobile browser and enter the URL for your Qlik Sense Enterprise hub.
2. Click **...** in the top toolbar of the hub, and then click **Client authentication**.
3. A dialog box opens asking you to confirm that you want to open the authentication link using the Qlik Sense. Click **Open** to confirm.
4. The Qlik Sense Mobile app opens and the server is added to the welcome page.
5. Click the server name to log in. You may be asked to enter your Qlik Sense Enterprise credentials.

After this, when you launch the Qlik Sense Mobile app, you can click the server name and log in using your Qlik Sense Enterprise credentials without authenticating against the Qlik Sense Enterprise hub each time.

Receiving the authentication link from your administrator

Do the following:

1. Click the authentication link provided by your Qlik Sense administrator. If you cannot click the link, copy the link into your mobile browser.

2. A dialog box opens asking you to confirm that you want to open the authentication link using the Qlik Sense. Click **Open** to confirm.
3. The Qlik Sense Mobile app opens and the server is added to the welcome page.
4. Click the server name to log in. You may be asked to enter your Qlik Sense Enterprise credentials.

After this, when you launch the Qlik Sense Mobile app, you can click the server name and log in using your Qlik Sense Enterprise credentials without authenticating against the Qlik Sense Enterprise hub each time.

4.5 Deploying mashups to the Qlik Sense Mobile app

Qlik Sense mashups are webpages that contain Qlik Sense app objects, such as charts and data. When a mashup is published in the Qlik Sense Enterprise hub, it can be also accessed from the Qlik Sense Mobile app.

Why use mashups in the Qlik Sense Mobile app

Using mashups in the Qlik Sense Mobile app enables faster loading and reduced data consumption for the mobile device. Mashups are generally less resource intense than Qlik Sense apps. This means that less data has to be retrieved from the Qlik Sense Enterprise server when loading a mashup in the Qlik Sense Mobile app.

Qlik Sense November 2018 or later is required to access mashups from the Qlik Sense Mobile app.

Only mashups published in Qlik Sense can be accessed from the Qlik Sense Mobile. In the Qlik Sense Mobile app, mashups are listed in a dedicated **Mashups** stream. All public mashups in a Qlik Sense Enterprise installation are visible in the Qlik Sense Mobile app. An admin can restrict access to specific users by creating a security rule in the Qlik Management Console. See: *Restricting access to mashups in the Qlik Sense Mobile app (page 161)*.

Non-optimized and optimized mashups

The mashups available in a Qlik Sense Mobile app, can be non-optimized or optimized.

- A non-optimized mashup retrieves the necessary data from the Qlik Sense server every time it is opened. This ensures that the non-optimized mashup is always up to date with the Qlik Sense Enterprise installation.
- An optimized mashup reduces data consumption for the mobile device by prioritizing locally stored app data over data stored on the server. When an optimized mashup opens, it fetches locally stored data from downloaded apps. It then loads what is not available locally from the Qlik Sense server. If the mashup finds all the necessary data on the device, it does not load data from the server. If the mashup does not find any necessary data on the device, it loads all mashup data from the server. An optimized mashup never stores new data on the device.

The download of Qlik Sense apps for offline access is currently supported only on Qlik Sense Mobile app for iOS. In the Qlik Sense Mobile app for Android, mashups always load necessary data from the Qlik Sense server.

To enable optimized mashup, the Qlik Sense administrator must insert the following line in the mashup's QEXT file:

"optimized": true

Restricting access to mashups in the Qlik Sense Mobile app

To restrict access to mashups in the Qlik Sense Mobile app to specific users, the Qlik Sense Enterprise administrator must setup a security rule in the Qlik Management Console (QMC).

Do the following:

1. In the QMC, create a custom property by doing the following:
   - Set a name for the new custom property, for example, "StreamAccess".
   - In the **Resource Types** section, select the **Extension** and **Users** check boxes to apply the custom property to these resource types.
   - In the **Value** section, create a new custom property value, for example,"MyMashup".

   See: "Creating a custom property" in the Manage Qlik Sense sites guide.
2. To allow access to mashups to specific users, apply the custom property created in step 1 to the selected users. In the QMC, go in the **Users** section and edit users by adding "MyMashup" in the **StreamAccess** field.
3. To allow access to extensions to specific users, apply the custom property created in step 1 to the selected users. In the QMC, go in the **Extension** section and edit extensions by adding "MyMashup" in the **StreamAccess** field.
4. Create a new stream. Add to the stream the Qlik Sense apps that contain the data used in the mashups.
5. To prevent users from accessing a mashup, change the extension security rule as follows:
   - Create a copy of the default extension security rule.
   - Edit the copy you created by adding the condition ((resource.name!="MyMashup")), where "MyMashup" is the custom property you created in step 1.
   - Disable the default extension security rule to make the new one effective.

   See: "Security rules installed in Qlik Sense" in the Manage Qlik Sense sites guide.
6. Create the following security rule for extensions: ((user.@StreamAccess="MyMashup")) to allow specific users to access all extensions.
   See: "Creating security rules" in the Manage Qlik Sense sites guide.
7. Apply the same security rule ((user.@StreamAccess="MyMashup")) to the stream you created in step 4 to allow specific users to access the stream.

   See: "Editing streams" in the Manage Qlik Sense sites guide.

Setting a mashup as landing page in an EMM environment

When administering Qlik Sense Mobile in an Enterprise Mobile Management (EMM) environment, you can set a mashup or a mashup stream as the landing page for users accessing Qlik Sense. When a user opens the Qlik Sense Mobile app within the EMM environment and authenticates in the Qlik Sense server, a specific mashup or mashup stream opens as starting page in place of the Qlik Sense hub.

To use a mashup or mashup stream as landing page, do the following;

In the **Configuration Value** field, enter the following:

```
{"LandingPage":"/extensions/MyMashup/MyMashup.html", "DefaultStream":"MyMashup" }
```

Where:

- "LandingPage" is the path to the mashup to be used as landing page.
- "DefaultStream" is the default stream that is loaded when accessing Qlik Sense.

5 Upgrading and updating Qlik Sense Enterprise for Windows

In this section you can read about how to upgrade and update your Qlik Sense installation. The upgrade procedure is different depending on whether you are upgrading from Qlik Sense 3.1 SR1 or earlier, or from Qlik Sense 3.1 SR2 or later.

You can upgrade from Qlik Sense 3.1 SR2 to Qlik Sense June 2017 or later using the Qlik Sense setup program. To upgrade to Qlik Sense June 2017 or later, see *Upgrading (page 164)*.

Upgrading from any version of Qlik Sense earlier than 3.1 SR2 to Qlik Sense June 2017 or later cannot be done using the setup program. If you wan to upgrade from Qlik Sense 3.1 SR1 or earlier to Qlik Sense June 2017 or later, you must first upgrade to Qlik Sense June 2017. Once your environment is on version June 2017 of Qlik Sense, you can upgrade to any newer version using the Qlik Sense setup program. See: *Upgrading to Qlik Sense June 2017 or later from Qlik Sense versions earlier than 3.1 SR2 (page 169)*.

You can update your Qlik Sense deployment by applying patches. A patch primarily includes software updates and fixes that are applied to the existing Qlik Sense version. For more information, see *Patching Qlik Sense (page 178)*.

5.1 Upgrades and migrating persistence models

Qlik Sense June 2017 or later only supports the shared persistence model. It does not support the synchronized persistence model. When you upgrade your Qlik Sense 3.1 SR2 deployment to Qlik Sense June 2017 or later, you will be migrated to a shared persistence model.

You can upgrade from synchronized persistence to shared persistence if the existing deployment is running Qlik Sense version 3.1 SR2. See, *Upgrading and migrating from synchronized to shared persistence (page 172)*.

When upgrading a central node from synchronized persistence to shared persistence, the existing repository database is shared with all nodes. If you want to setup a dedicated repository database on a separate machine, you must perform a new installation. For more information, see *Installing and configuring PostgreSQL (page 98)*, and *Installing Qlik Sense in a multi-node site (page 85)*.

5.2 Upgrades and centralized logging

Upgrading from Qlik Sense June 2017 or earlier provides the option to configure centralized logging through the installer wizard. Upgrading from Qlik Sense September 2017 or later does not provide this option. During the upgrade, centralized logging will be set up only if it was configured in the earlier Qlik Sense version. If centralized logging was not set up, the logging service will still be installed and running but without the database.

5.3 Upgrading

You can upgrade from Qlik Sense 3.1 SR2 or later to Qlik Sense June 2017 or later using the Qlik Sense setup program. When upgrading, the previous version is completely replaced by the new version.

To upgrade from Qlik Sense 3.1 SR2 or later with a shared persistence model to Qlik Sense June 2017 or later, see *Upgrading from Qlik Sense 3.1 SR2 or later to Qlik Sense June 2017 or later (page 166)*.

> ⚠ *Do not uninstall Qlik Sense before upgrading to Qlik Sense June 2017 or later. If you are upgrading to Qlik Sense June 2017 or later, and you have uninstalled Qlik Sense, see Upgrading to Qlik Sense June 2017 or later after uninstalling Qlik Sense 3.1 SR2 or later (page 168).*

Qlik Sense June 2017 and later versions do not support the synchronized persistence model. To upgrade from Qlik Sense 3.1 SR2 or later to Qlik Sense June 2017 or later and migrate from a synchronized persistence model to a shared persistence model, see *Upgrading and migrating from synchronized to shared persistence (page 172)*.

Upgrading from any version of Qlik Sense earlier than 3.1 SR2 to Qlik Sense June 2017 or later cannot be done using the setup program. To upgrade from earlier versions of Qlik Sense with a synchronized persistence model to Qlik Sense June 2017 or later, see *Upgrading to Qlik Sense June 2017 or later from Qlik Sense versions earlier than 3.1 SR2 (page 169)*.

> ℹ *When you upgrade to a newer version of Qlik Sense, you will not get the option to configure centralized logging in the installer. Instead, if you want to enable centralized logging, you must configure it using the Qlik Logging service. See: Qlik Logging Service (page 237)*

> ⚠ *Qlik Sense November 2017 and later versions do not support soft deleted records. Qlik Sense will clean up all soft deleted records on the first startup of the Qlik Sense Repository Service after an upgrade. For troubleshooting, refer to Failed to remove soft deleted records (page 314).*

> ⚠ *During upgrade, the Repository.exe.config file located in %ProgramFiles%\Qlik\Sense\Repository is overwritten with default settings.*
> *If the file was manually changed in your previous deployment, you must create a backup of the file before upgrading, and use the backup to restore your customized settings. Once the Repository.exe.config is restored, you must restart the Qlik Sense services.*

Qlik Sense apps

When you upgrade Qlik Sense all existing apps need to be migrated to ensure compatibility between the versions. This happens automatically when the system starts the first time after the upgrade. If the migration fails for one or more apps, these apps will not be available in the **Hub** after the upgrade. Apps that are not migrated are indicated in the **Apps** section of Qlik Management Console, where you can also perform a manual migration.

Multi-node deployments

In a multi-node deployment, all nodes must run the same version of Qlik Sense to be able to communicate with each other. It is recommended to upgrade with all nodes offline, and to start with the central node.

> *When upgrading a rim node, ensure that you use the same log-in account as was used for the initial installation of that node. Failure to do so means that the central node will not find the certificates installed on the node and you will need to perform a clean installation of the node.*

Qlik Sense Repository Database

Qlik Sense June 2017 and later versions use PostgreSQL version 9.6 for the Qlik Sense Repository Database. If you upgrade in place without uninstalling Qlik Sense the Qlik Sense Repository Database is upgraded to PostgreSQL version 9.6 and your data, and standard settings are carried forward. If you have made custom configurations to your PostgreSQL installation, those must be recreated in the PostgreSQL after upgrade.

PostgreSQL version 9.6 is installed with the latest version of Qlik Sense. If you have uninstalled Qlik Sense but maintained your PostgreSQL database, and you want to upgrade your Qlik Sense deployment, you must create a database dump file and restore the PostgreSQL database manually. You will also need to manually reconfigure any custom parameters.

The PostgreSQL installation included in the Qlik Sense June 2017 or later setup does not include pgAdmin tools. For information about manually installing the PostgreSQL database, see *Installing and configuring PostgreSQL (page 98)*.

Before you upgrade Qlik Sense, do the following:

- Review *System requirements for Qlik Sense (page 12)*.
- Download the *Qlik_Sense_setup.exe* file.
- Make sure you have logged on as an administrator using an account that has an actual password defined, that is, not a blank password.
- If you are running the Qlik Sense services with a LocalSystem account then you must change to a service user account before begining the upgrade.
  See: *Upgrading from Qlik Sense 3.1 SR2 or later to Qlik Sense June 2017 or later (page 166)*
- Create a backup of your Qlik Sense deployment before upgrading.

The

Do the following:

1. Stop your Qlik Sense services.
2. Upgrade your central node by launching the Qlik Sense setup file (Qlik_Sense_setup.exe).
3. Select **Upgrade** to upgrade your existing shared persistence deployment.
4. Accept the license agreement and click **Next**.

5. On the **Service Credentials** page, enter the **Username** and **Password** for your Windows Qlik Sense service user account.
   If the user is member of a domain, enter the service account as *<domain>\<username>*. For more information, see *User accounts (page 69)*.
6. On the **Ready to upgrade** page, select the appropriate check boxes if you want the setup to create desktop shortcuts and automatically start the Qlik Sense services when the setup is complete, and click **Upgrade**.
7. Check that all of the Qlik Sense services have started successfully.
8. Check that all apps have been migrated successfully on the central node. If migration has failed for one or more apps, resolve the issues before continuing.

   > *If the node running the app migration goes offline, migration will stop. It will not restart automatically. In a single node environment, all apps will have **Migration status** set to **Unknown**. See: Migrating apps. In a multi-node environment with failover nodes, the primary node will be replaced by the next available node, but migration will not restart. See: Failover (page 97). To resume migration, you will need to restart the following services, in order: Qlik Sense Service Dispatcher (QSD) and Qlik Sense Repository Service (QRS). See: Services (page 22).*

9. Deploy the Qlik Sense upgrade with shared persistence on the remaining nodes.

   > *Any custom manual configurations that you make to the PostgreSQL database must be manually reproduced after the upgrade.*

Upgrading from Qlik Sense 3.1 SR2 or later to Qlik Sense June 2017 or later

Before you upgrade, if your Qlik Sense 3.1 SR2 or later installation is running services using a Local System account, you need to change this to use a service user account (local or domain) before upgrading to Qlik Sense June 2017 or later. If you continue to use a Local System account to run the services when upgrading you will get an error.

Changing the user account type to run the Qlik Sense services on a central node

Do the following:

1. In Windows, either create a new or use an existing domain or local user account to run the Qlik Sense services.
2. If the service account user does not have administrator privileges, you must add the user to the following groups in **Computer Management** > **System Tools** > **Local Users and Groups** > **Groups**.
   - Qlik Sense Service Users
   - Performance Monitor Users
3. Open the **Control Panel** and then select **System and Security**>**Administrative Tools**>**Services**.

4. Stop all services except the **Repository Database**.
5. Assign **Full control** permission for the dedicated service account to the folder *%ProgramData%\Qlik\Sense*.
6. As an administrator, open an elevated command prompt.
7. Navigate to the *Program Files\Qlik\Sense\Proxy* folder and run `Proxy.exe -bootstrap`.
8. Navigate to the *Program Files\Qlik\Sense\Scheduler* folder and run `Scheduler.exe -bootstrap`.
9. Navigate to the *Program Files\Qlik\Sense\Repository* folder and run `Repository.exe -bootstrap`.
   If you are changing the user account on your primary or central node, run `Repository.exe -bootstrap -iscentral`.
10. Close the elevated command prompt.
11. Change the log on credentials for each of the Qlik Sense services as follows:
    a. Right-click the service and select **Properties**.
    b. Select the **Log On** tab and then **This account**.
    c. Enter the credentials for the dedicated service account and click **OK**.

    The services are as follows:
    - Qlik Sense Engine Service
    - Qlik Sense Printing Service
    - Qlik Sense Proxy Service
    - Qlik Sense Repository Service
    - Qlik Sense Scheduler Service
    - Qlik Sense Service Dispatcher

    If you are using a user account with administrative privileges, keep the Qlik Sense Repository Database running under the Local System account. Do not change the account.

    Depending on your setup some of the services may not be available.
12. Start the Qlik Sense Service Dispatcher, and then the Qlik Sense Repository Service (QRS).
13. Start the rest of the Qlik Sense services.

Upgrading from Qlik Sense 3.1 SR2 or later with a shared persistence model to Qlik Sense June 2017 or later

Do the following:

1. Stop your Qlik Sense services.
2. Upgrade your central node by launching the Qlik Sense setup file (Qlik_Sense_setup.exe).
3. Select **Upgrade** to upgrade your existing shared persistence deployment.
4. Accept the license agreement and click **Next**.
5. On the **Service Credentials** page, enter the **Username** and **Password** for your Windows Qlik Sense

service user account.
If the user is member of a domain, enter the service account as *<domain>\<username>*. For more information, see *User accounts (page 69)*.

6. On the **Ready to upgrade** page, select the appropriate check boxes if you want the setup to create desktop shortcuts and automatically start the Qlik Sense services when the setup is complete, and click **Upgrade**.
7. Check that all of the Qlik Sense services have started successfully.
8. Check that all apps have been migrated successfully on the central node. If migration has failed for one or more apps, resolve the issues before continuing.

*If the node running the app migration goes offline, migration will stop. It will not restart automatically. In a single node environment, all apps will have **Migration status** set to **Unknown**. See: Migrating apps. In a multi-node environment with failover nodes, the primary node will be replaced by the next available node, but migration will not restart. See: Failover (page 97). To resume migration, you will need to restart the following services, in order: Qlik Sense Service Dispatcher (QSD) and Qlik Sense Repository Service (QRS). See: Services (page 22).*

9. Deploy the Qlik Sense upgrade with shared persistence on the remaining nodes.

Any custom manual configurations that you make to the PostgreSQL database must be manually reproduced after the upgrade.

Upgrading to Qlik Sense June 2017 or later after uninstalling Qlik Sense 3.1 SR2 or later

If you have uninstalled Qlik Sense but maintained your PostgreSQL database, and you want to upgrade to Qlik Sense June 2017 or later, you must create a database dump file and restore the PostgreSQL database manually. You will also need to manually reconfigure any custom parameters.

Do the following:

1. Copy the PostgreSQL folder from *%ProgramData%\Qlik\Sense\Repository\PostgreSQL* to a temporary location outside of the *%ProgramData%* folder.
2. Download and install PostgreSQL version 9.6 from the PostgreSQL website. For more information, see *Installing and configuring PostgreSQL (page 98)*.
3. Open a Command Prompt in Microsoft Windows.

The pg_ctl.exe command should not be run as an administrator.

4. Navigate to the location where the PostgreSQL repository database is installed, *%ProgramFiles%\PostgreSQL\<database version>\bin*, and run the following commands:

   a. `pg_ctl.exe start -w -D "C:\SenseDB\9.6"`
   b. `set PGUSER=postgres`
   c. `set PGPASSWORD=password`
   d. `pg_dumpall.exe > [<path to dump file>]`
   e. `pg_ctl.exe stop -w -D "C:\SenseDB\9.6"`

5. In the **Command Prompt** window, navigate to the folder containing the *Qlik_Sense_setup.exe* file.
6. Run the following command to install Qlik Sense and restore your Qlik Sense Repository Database.
   `Qlik_Sense_setup.exe databasedumpfile=<path_to_dump_file>`

   > *The path to the dump file must be entered as an absolute path, using a relative path will result in an installation failure.*

7. Follow the setup to complete the installation. For more information, see *Qlik Sense installation (page 77)*.

Upgrading to Qlik Sense June 2017 or later from Qlik Sense versions earlier than 3.1 SR2

Qlik Sense June 2017 and later versions do not support the synchronized persistence model. To upgrade to Qlik Sense June 2017 or later from any version of Qlik Sense earlier than 3.1 SR2 and migrate from a synchronized to shared persistence model, follow the instructions on this page.

Upgrading from any version of Qlik Sense earlier than 3.1 SR2 to Qlik Sense June 2017 or later cannot be done using the setup program. If you wan to upgrade from Qlik Sense 3.1 SR1 or earlier to Qlik Sense June 2017 or later, you must first upgrade to Qlik Sense June 2017. Once your environment is on version June 2017 of Qlik Sense, you can upgrade to any newer version using the Qlik Sense setup program.

> *If you attempt to upgrade from Qlik Sense versions earlier than 3.1 SR2 to Qlik Sense June 2017 using the setup program you will receive an error.*

> *The new hostname must match the one used before the upgrade. Using a different hostname will cause a mismatch in the certificate, which will prevent you from accessing the hub. You can verify the previous hostname by:*
> *- Checking the certificate name*
> *- Checking the file C:\Programdata\Qlik\Sense\Host.cfg (String encoded based64)*
> *See also* ***Restoring a central node to a machine with a different hostname*** *section in Restoring a Qlik Sense site (page 204)*

Do the following:

1. Create a backup of your existing Qlik Sense deployment. For more information, see the help for the version of Qlik Sense that you are currently running.
2. Change the PostgreSQL authentication mode in the configuration settings to allow the password to be changed.
   1. Stop the Qlik Sense Repository Database service.
   2. Open the Client Authentication file located in *ProgramData\Qlik\Sense\Repository\PostgreSQL\<database version>\pg_hba.conf* .
   3. Change the **ADDRESS** to `127.0.0.1/32`, and change the **METHOD** to `trust` for for **IPv4 local connections** and local host replication.
   4. Change the **ADDRESS** to `::1/128`, and change the **METHOD** to `trust` for **IPv6 local connections** and local host replication.
   5. Start the Qlik Sense Repository Database service.
3. Change the Qlik Sense Repository Database password.
   To change the password using PostgreSQL command line:
   1. Open a command prompt and navigate to *ProgramFiles\Qlik\Sense\Repository\PostgreSQL\<database version>\bin*.
   2. Connect to the database by entering the following command:
      `psql.exe -p 4432 -U postgres`.
   3. Enter the following command to set the new user password:
      `ALTER USER qliksenserepository WITH PASSWORD '<newpassword>';`
      `ALTER ROLE` is displayed after successfully changing the password.

   To change the password using the *pgAdmin* tool:
   1. Launch **pgAdmin** and connect to the Qlik Sense Repository Database.
   2. Expand the tree in the left pane and click **Login Roles** > **qliksenserepository**.
   3. Right-click on **qliksenserepository** and select **Properties**.
   4. Click the **Definition** tab, and enter a **Password**.
4. Reset the PostgreSQL authentication mode in the configuration settings to require authentication.
   1. Stop the Qlik Sense Repository Database service.
   2. Open the Client Authentication file located in *ProgramData\Qlik\Sense\Repository\PostgreSQL\<database version>\pg_hba.conf* .
   3. Change the **METHOD** back to `md5`.
   4. Start the Qlik Sense Repository Database service.
5. Create a database dump file.
   If Qlik Sense is installed:
   1. Stop all Qlik Sense services except Qlik Sense Repository Database service. Ensure that the Qlik Sense Repository Database service is running.
   2. Open a command prompt and navigate to the location where the PostgreSQL database is installed, and enter the following commands:
      - `set PGUSER=postgres`
      - `set PGPASSWORD=[superuserpassword]`
      - `pg_dumpall.exe -p 4432 > [path to dump file]`

If Qlik Sense has been uninstalled:

a. Copy the PostgreSQL folder from *%ProgramData%\Qlik\Sense\Repository\PostgreSQL\9.6* to a temporary location outside of the *%ProgramData%\Qlik* folder.

b. Download and install PostgreSQL version 9.6 from the PostgreSQL website. For more information, see *Installing and configuring PostgreSQL (page 98)*.

c. Open a Command Prompt in Microsoft Windows.

d. Navigate to the location where the PostgreSQL repository database is installed, *cd "%ProgramFiles%\PostgreSQL\9.6\data\bin"*, and run the following commands:

   - `pg_ctl.exe start -w -D "C:\SenseDB\9.6"`
   - `set PGUSER=postgres`
   - `set PGPASSWORD=password`
   - `pg_dumpall.exe > [path to dump file]`
   - `pg_ctl.exe stop -w -D "C:\SenseDB\9.6"`

6. Make a backup of log and application data in the following folders:
   - *%ProgramData%\Qlik\Sense\Log*
   - *%ProgramData%\Qlik\Sense\Apps*
   - *%ProgramData%\Qlik\Sense\Repository\Content*
   - *%ProgramData%\Qlik\Sense\Repository\Extensions*
   - *%ProgramData%\Qlik\Sense\Repository\AppContent* (if available)
   - *%ProgramData%\Qlik\Sense\Repository\SharedContent* (if available)
7. Make a backup of any locations where content that supports the Qlik Sense environment may be kept (for example, QVD files created by load scripts).
8. Create a file share, see *Creating a file share (page 97)*.
9. Create the following sub-folders in the file share:
   - *Apps*
   - *ArchivedLogs*
   - *CustomData*
   - *StaticContent*
10. Copy following content from your synchronized persistence deployment to the file share:

Content	Copy from	To subfolder
Apps	*..\ProgramData\Qlik\Sense\Apps*	*Apps*
Logs (optional)	*..\ProgramData\Qlik\Sense\Repository\Archived Logs*	*ArchivedLogs*

Content	Copy from	To subfolder
Static content	*..\ProgramData\Qlik\Sense\Repository\AppContent* *..\ProgramData\Qlik\Sense\Repository\Content* *..\ProgramData\Qlik\Sense\Repository\DefaultContent* *..\ProgramData\Qlik\Sense\Repository\Extensions* *..\ProgramData\Qlik\Sense\Repository\DefaultApps* *..\ProgramData\Qlik\Sense\Repository\SharedContent* *Each of these folders must be added as a sub-folder of the StaticContent folder.*	*StaticContent*

11. Ensure that all Qlik Sense nodes are synchronized, and take all nodes offline by stopping the Qlik Sense services in Windows.
12. Uninstall Qlik Sense. Accept the defaults when uninstalling to preserve the certificates settings.
13. In the **Command Prompt** window, navigate to the folder containing the *Qlik_Sense_setup.exe* file.
14. Run the following command to install Qlik Sense and restore your Qlik Sense Repository Database.

```
Qlik_Sense_setup.exe databasedumpfile=<path_to_dump_file>
```

> *The path to the dump file must be entered as an absolute path, using a relative path will result in an installation failure.*

15. Uninstall Qlik Sense on each of your rim nodes in multi-node deployment. Select the option to completely uninstall Qlik Sense when you uninstall on the rim nodes.
16. Install Qlik Sense on each of the rim nodes.
17. Connect the rim nodes in the QMC, select each node, and click the **Redistribute** button.

5.4 Upgrading and migrating from synchronized to shared persistence

You can upgrade and migrate from synchronized persistence to shared persistence if the existing deployment is running Qlik Sense version 3.1 SR2 or later. For more information about persistence models, see *Persistence (page 48)*.

The files that are persisted in a Qlik Sense deployment must be available to all nodes via the file share. They can be stored on any of the nodes in the cluster, or on another server. If you are migrating from a synchronized persistence deployment to a shared persistence deployment, you must first create the file share to use as shared storage, and copy your data from the synchronized persistence deployment into the file share folders. For instructions on how to create a file share, see *Creating a file share (page 97)*.

Before you upgrade Qlik Sense, do the following:

- Review *System requirements for Qlik Sense (page 12)*.
- Download the *Qlik_Sense_setup.exe* file.
- Create a backup of your Qlik Sense deployment before upgrading.

Backing up a synchronized persistence site

Proceed as follows to backup a Qlik Sense site deployed with the synchronized persistence model:

1. Make a backup of the certificates used to secure the Qlik Sense services. This only needs to be done once.

   Backing up certificates (page 183)
2. Stop all Qlik Sense services except the Qlik Sense Repository Database (QRD).
3. Make a backup of the repository database.
   a. Open a Command Prompt with administrator privileges in Microsoft Windows.
   b. Produce a dumpfile for the repository database (that is, a single file for the entire database):
      i. Navigate to the installation location.
         %ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\<database version>\bin
      ii. *pg_dump.exe -h localhost -p 4432 -U postgres -b -F t -f "c:\QSR_backup.tar" QSR*

         If you are prompted for the PostgreSQL super user password, enter the password that was given during the installation of Qlik Sense.

         > *To avoid being prompted for the password (for example, if you want to automate the Qlik Sense backup process), you can use the pgpass functionality in PostgreSQL. See the PostgreSQL documentation for more information.*
   c. Make a backup of the dumpfile for the repository database.
4. Make a backup of log and application data in the following folders:
   - *%ProgramData%\Qlik\Sense\Log*
   - *%ProgramData%\Qlik\Sense\Apps*
   - *%ProgramData%\Qlik\Sense\Repository\Content*
   - *%ProgramData%\Qlik\Sense\Repository\Extensions*
   - *%ProgramData%\Qlik\Sense\Repository\AppContent* (if available)
   - *%ProgramData%\Qlik\Sense\Repository\SharedContent* (if available)
5. Make a backup of any locations where content that supports the Qlik Sense environment may be kept (for example, QVD files created by load scripts).
6. Start the Qlik Sense services. If the services are started manually, start them in the following order:
   a. Qlik Sense Repository Service (QRS)
      If the user running Qlik Sense services is not local administrator on the machine, you need to start Repository.exe from an elevated command prompt using the -bootstrap parameter.
      Services (page 22)

b. Qlik Sense Proxy Service (QPS), Qlik Sense Engine Service (QES), Qlik Sense Scheduler Service (QSS), and Qlik Sense Printing Service (QPR) in no specific order

The order is important because the QRS is dependent on the QRD and the rest of the services are dependent on the QRS.

Upgrading to a shared persistence deployment

Do the following:

1. Create a file share, see *Creating a file share (page 97)*.
2. Create the following sub-folders in the file share:
   - *Apps*
   - *ArchivedLogs*
   - *CustomData*
   - *StaticContent*
3. Ensure that all Qlik Sense nodes are synchronized, and take all nodes offline by stopping the Qlik Sense services in Windows.
4. Copy following content from your synchronized persistence deployment to the file share:

Content	Copy from	To subfolder
Apps	*..\ProgramData\Qlik\Sense\Apps*	*Apps*
Logs (optional)	*..\ProgramData\Qlik\Sense\Repository\Archived Logs*	*ArchivedLogs*
Static content	*..\ProgramData\Qlik\Sense\Repository\AppContent* *..\ProgramData\Qlik\Sense\Repository\Content* *..\ProgramData\Qlik\Sense\Repository\DefaultContent* *..\ProgramData\Qlik\Sense\Repository\Extensions* *..\ProgramData\Qlik\Sense\Repository\DefaultApps* *..\ProgramData\Qlik\Sense\Repository\SharedContent* *Each of these folders must be added as a sub-folder of the StaticContent folder.*	*StaticContent*

5. Upgrade your central node by launching the Qlik Sense setup file (Qlik_Sense_setup.exe).
6. Accept the license agreement and click **Next**.
7. On the ***Shared persistence storage*** page, enter the path or URL to your file share folders that you prepared and click **Next**.
8. On the **Database service listener** page, if you have a multi-node deployment, enter the following:
   - **Listen addresses** - add the addresses that the database service should listen to.
     You can enter a comma separated list of IPv4 or IPv6 addresses, or *0.0.0.0* (for all IPv4 addresses), `::/0` (for all IPv6 addresses) or * (for all addresses).

- **IP ranges** - add a subnet specification that covers the IP addresses of all nodes in your site. Either add one row for each node, using /32 as suffix for each address, or add a subnet that covers all addresses using, for example, /24 as suffix. To allow all servers to access the repository database, use 0.0.0.0/0. You can also enter a comma separated list of multiple IP addresses.
- **Max connections** - specify the maximum number of concurrent connections to the database. The default value is 100 multiplied by the number of nodes in the cluster (this field is only available in Qlik Sense February 2018, and later).

9. On the **Service Credentials** page, enter the **Username** , and **Password** for your WindowsQlik Sense service user account.
   If the user is member of a domain, enter the service account as <*domain*>\<*username*>. See: *User accounts (page 69)*.
10. On the **Repository Database Superuser Password** page, enter the password for your repository database superuser. See: *User accounts (page 69)*.
    If you cannot find the password, see the troubleshooting topic:*Cannot find the repository database superuser password (page 303)*
11. On the **Ready to upgrade** page, select the appropriate check boxes if you want the setup to create desktop shortcuts and automatically start the Qlik Sense services when the setup is complete, and click **Upgrade**.
12. Check that all of the Qlik Sense services have started successfully.
13. Check that all apps have been migrated successfully on the central node. If migration has failed for one or more apps, resolve the issues before continuing.

    > *If the node running the app migration goes offline, migration will stop. It will not restart automatically. In a single node environment, all apps will have **Migration status** set to **Unknown**. See: Migrating apps. In a multi-node environment with failover nodes, the primary node will be replaced by the next available node, but migration will not restart. See: Failover (page 97). To resume migration, you will need to restart the following services, in order: Qlik Sense Service Dispatcher (QSD) and Qlik Sense Repository Service (QRS). See: Services (page 22).*

14. In a multi-node deployment, uninstall Qlik Sense on each of the rim nodes. Select the option to remove certificates and data folders when you uninstall on the rim nodes.
15. Install Qlik Sense with shared persistence on the remaining nodes, and join the existing cluster created when you upgraded the central node.

5.5 Performing a silent upgrade

You can silently upgrade the current Qlik Sense installation. All setup options that are available in the user interface of the installer can be performed with silent operations.

Do the following:

1. Select **Start > All Programs > Accessories > Command Prompt**.
   The **Command Prompt** window is displayed.

2. In the **Command Prompt** window, navigate to the folder containing the *Qlik_Sense_setup.exe* file.
3. Enter *Qlik_Sense_setup.exe* followed by the silent installation syntax preferred.

Note that elevation will take place if run from an unelevated process and the UAC is on.

Syntax

```
Qlik_Sense_setup.exe [-silent] {-log "path\filename"}
{desktopshortcut=1|0} {skipstartservices=1|0} {installdir="path"}
{userpassword="password} {dbpassword="password"}
```

`Qlik_Sense_setup.exe -?` or `-h`	Brings up the on-screen silent setup help.

Commands

`-silent` (or `-s`)		Command line-driven setup without UI.(mandatory).
`-log` (or `-l`)	[log file name with path]	Log file directory and log file name. *The user must have access to this directory.*

Arguments

Arguments are separated by space and presented in the form [Argument]="[Value]". The double quotes can normally be omitted but may be needed, for example, when a path contains spaces.

The default values are the same as those used in the setup user interface.

`desktopshortcut`	`1`|`0` (defaults to 1 on clean installs)	Installs desktop shortcuts.
`skipstartservices`	`1`|`0` (defaults to 0 on clean installs, otherwise the current state.)	To skip starting services after the installation has finished.
`installdir`	[path to custom install directory]	Need only be defined if the default install directory will not be used (*%ProgramFiles%\Qlik\Sense*).
`userpassword`	[password]	The password of the user used to run the services.
`dbpassword`	[password]	Password for the database superuser that creates the user that runs the database.

The default values are the same as those used in the setup user interface.

Example: Upgrading the installation

This example shows how to silently upgrade an installation and add desktop shortcuts.

```
Qlik_Sense_setup.exe -s desktopshortcut=1
```

Deprecated command line arguments

For a list of the command line arguments that are no longer recommended, see Installing silently.

5.6 Repairing an installation

The **Repair** option restores all missing files, shortcuts and registry values without any credentials being changed.

> *If patches have been applied to Qlik Sense, the Repair option is disabled. You must uninstall all patches before you can use the **Repair** option, as it will restore the installation to the original installed version.*

Do the following:

1. To start repairing the installation, open the **Control Panel** and select **Uninstall a program**. Then select **Qlik Sense** from the list of programs and click **Change**.

   The **Qlik Sense Setup maintenance** screen is displayed.

   > *You can also perform this action by double-clicking the Qlik_Sense_setup.exe file. In that case, you must use the correct version of the setup file when repairing your Qlik Sense installation, that is, the same version used when installing Qlik Sense.*

2. Click **Repair**.
   The **Ready to repair** screen is displayed.
3. Click **Repair**.
   - If UAC is enabled, the **User Account Control** screen is displayed.
   - If UAC is disabled, the repair process starts.
4. Click **Yes** to start repairing your Qlik Sense installation.

   > *This is only applicable if UAC is enabled.*

   The progress is displayed.

   When finished, click **Repair Summary** to confirm that Qlik Sense has been restored successfully. Click **Back**.

5. Click **Finish**.

You have now successfully repaired your Qlik Sense installation.

5.7 Performing a silent repair

You can silently repair the current Qlik Sense installation. All setup options that are available in the user interface of the installer can be performed with silent operations.

Do the following:

1. Select **Start > All Programs > Accessories > Command Prompt**.
   The **Command Prompt** window is displayed.
2. In the **Command Prompt** window, navigate to the folder containing the *Qlik_Sense_setup.exe* file.
3. Enter *Qlik_Sense_setup.exe* followed by the silent installation syntax preferred.

Syntax

```
Qlik_Sense_setup.exe [-silent] [-repair] {-log "path\filename"}
```

`Qlik_Sense_setup.exe -?` or `-h` Brings up the on-screen silent setup help.

Commands

`-silent` (or `-s`) Trigger the silent mode (mandatory).

`-repair` Repair the product silently.

`-log` (or `-l`) Log file directory and log file name.

> *The user must have access to this directory.*

If this option is not defined, the log file will be stored with the default name in the default location.

Example:

This example shows how to silently repair the Qlik Sense installation.

```
Qlik_Sense_setup.exe -s -repair
```

5.8 Patching Qlik Sense

You can update your Qlik Sense deployment when a patch of the software is available for installation. A patch primarily includes software updates and fixes that are applied to the existing Qlik Sense version.

Patches are installed without the need to remove earlier updates or the major release. Qlik Sense patches are cumulative. By installing the latest patch, updates and fixes introduced in previous patches are also installed.

When you uninstall a patch, the individual updates from the installed version of Qlik Sense are removed.

In a multi-node site, all nodes must run the same version of Qlik Sense. We recommend installing patches with all nodes offline, and starting with the central node.

Before you install a patch Qlik Sense, do the following:

- Review *System requirements for Qlik Sense (page 12)*.
- Download the *Qlik_Sense_update.exe* file.
- Make sure you have logged on with Administrator rights using an account that has an actual password defined, that is, not a blank password.
- Create a backup of your Qlik Sense deployment. If Qlik Sense is installed on a Virtual Machine (VM) it may be sufficient to take a snapshot of the machine before upgrading. For more information, see *Backing up a Qlik Sense site (page 202)*.

When updating a rim node, ensure that you use the same log-in account as was used for the initial installation of that node. Failure to do so means that the central node will not find the certificates installed on the node and you will need to perform a clean installation of the node.

Do the following:

1. Stop the Qlik Sense services.
2. Run the setup to install a patch on the central node.
   When the installation is complete, the **Summary** is displayed.
3. Click **Finish** to close the **Summary**.

   *If the patch did not install successfully, the **Failed** screen is displayed. For more detailed information, see the installation log located in your **temp** folder accessed with environment variable %temp%.*

   You have successfully applied a patch to your Qlik Sense deployment.
4. Start the Qlik Sense services.
5. Repeat this procedure for each of the remaining nodes.

You cannot repair an installation using the repair option on the setup program once patches have been applied. The repair option is only available for the original software version, so any patches installed must be uninstalled before you can use the repair option.

Follow the same procedure to uninstall patches.

Silent patching

When a software patch is available for your Qlik Sense installation, you can use the command line tool to silently install the updates. Patches include software updates and fixes that are applied to the existing Qlik Sense version.

Commands

Use the following commands to silently run patch updates.

Command	Description
`install`	Runs a command line-driven install without a user interface. For feedback, see the log files, and the return values.
`uninstall`	Runs a command line-driven uninstall without a user interface. For feedback, see the log files, and the return values.
`startservices`	Used with `[install]`, or `[uninstall]`, this command determines whether the services should be started automatically or not.
`log=[path to logfile]`	Specifies the location for the patch to writes log files.
`unpack=[path]`	Unpacks the patch contents without installing.
`help (or -h, /h, -?, /?)`	Opens the help dialog.

To troubleshoot silent patching, start by examining the installation log files. The default location of the log files is: *C:\Users\[username]\AppData\Local\Temp*.

Example

The following command is an example of the syntax you can use for running a patch update file:

```
Qlik_Sense_update.exe install startservices
```

This command installs the update, and restores the services to the same state they were in before the update.

5.9 Uninstalling Qlik Sense

If any updates have been applied to Qlik Sense since installation, the Uninstall option will also remove all the updates.

Do the following:

1. To start uninstalling, open the **Control Panel** and select **Uninstall a program**. Then select **Qlik Sense** from the list of programs and click **Uninstall**.

A confirmation screen is displayed asking if you are sure that you want to uninstall Qlik Sense from your computer. Select the **Remove Qlik Sense certificates and data folders** checkbox if you want to remove all files from the machine ready for a new configuration.

> *If the machine is a central node in a Qlik Sense site, there may be rim nodes on other machines that require access to the central node to function properly.*

> *You can also uninstall Qlik Sense by double-clicking the Qlik_Sense_setup.exe file and then selecting Uninstall from the Maintenance screen. In that case, you must use the correct version of the setup file when uninstalling your Qlik Sense installation, that is, the same version used when installing Qlik Sense.*

2. Click **Uninstall** to start uninstalling Qlik Sense.
   If User Account Control (UAC) is disabled, the uninstall starts.
   If UAC is enabled, the **User Account Control** dialog is displayed.
   Click **Yes** to start the uninstall.

   The progress of the uninstall process is displayed. When finished the uninstall dialog confirms that Qlik Sense has been uninstalled successfully.

3. Click **Finish**.

You have now uninstalled Qlik Sense.

6 Backup and restore Qlik Sense

To ensure that your Qlik Sense site can be recovered in the event of a system failure or when a node in your deployment needs to be moved or replaced, we recommend that you create regular backups. These backups are used to restore your Qlik Sense site when needed.

If you creating a backup to upgrade from a synchronized persistence deployment to a shared persistence deployment, see *Upgrading and migrating from synchronized to shared persistence (page 172)*.

To back up a deployment running Qlik Sense 3.2.x or earlier, refer to the documentation for the release that you are running.

To backup a Qlik Sense site, you must back up the following:

- Qlik Sense certificates
- Qlik Sense Repository Database
- Shared persistence file share

6.1 Qlik Sense certificates

Qlik Sense uses certificates to secure communication between components that are installed on different computers. It is recommended that you back up the certificates on the central node in a Qlik Sense site immediately after installation, so that they can be restored if needed.

Backed up certificates can be used to restore certificates on the same node as they were exported from. A backed up server certificate can also be moved from one node of a Qlik Sense site to another node in the same site. For more information, see *Restoring certificates (page 192)*.

For more information about how to back up the Qlik Sense certificates, see *Backing up a Qlik Sense site (page 202)*.

6.2 Qlik Sense Repository Database

The Qlik Sense Repository Database is a PostgreSQL database that contains system data and meta data about apps. The Qlik Sense Repository Database can reside on the central node or on another computer. If the Qlik Sense Repository Database was installed during setup it will be located on the central node. If the Qlik Sense Repository Database was installed manually, it may be located on another computer.

The Qlik Sense Repository Database should be backed up on a regular basis to avoid data loss.

For more information about how to back up the Qlik Sense Repository Database, see *Backing up a Qlik Sense site (page 202)*.

For more information about how to restore the Qlik Sense Repository Database, see *Restoring a Qlik Sense site (page 204)*.

6.3 Shared persistence file share

The shared persistence file share is used to store Qlik Sense app data, such as visualizations, and dimensions and measures. It also stores static content, such as images and extensions, as well as system logs. It is accessible to all nodes in your Qlik Sense site. The file share can reside either on the same server as the central node or on another server.

The file share should be backed up on a regular basis to avoid data loss.

For more information about how to back up the file share, see *Backing up a Qlik Sense site (page 202)*.

For more information about how to restore the file share, see *Restoring a Qlik Sense site (page 204)*.

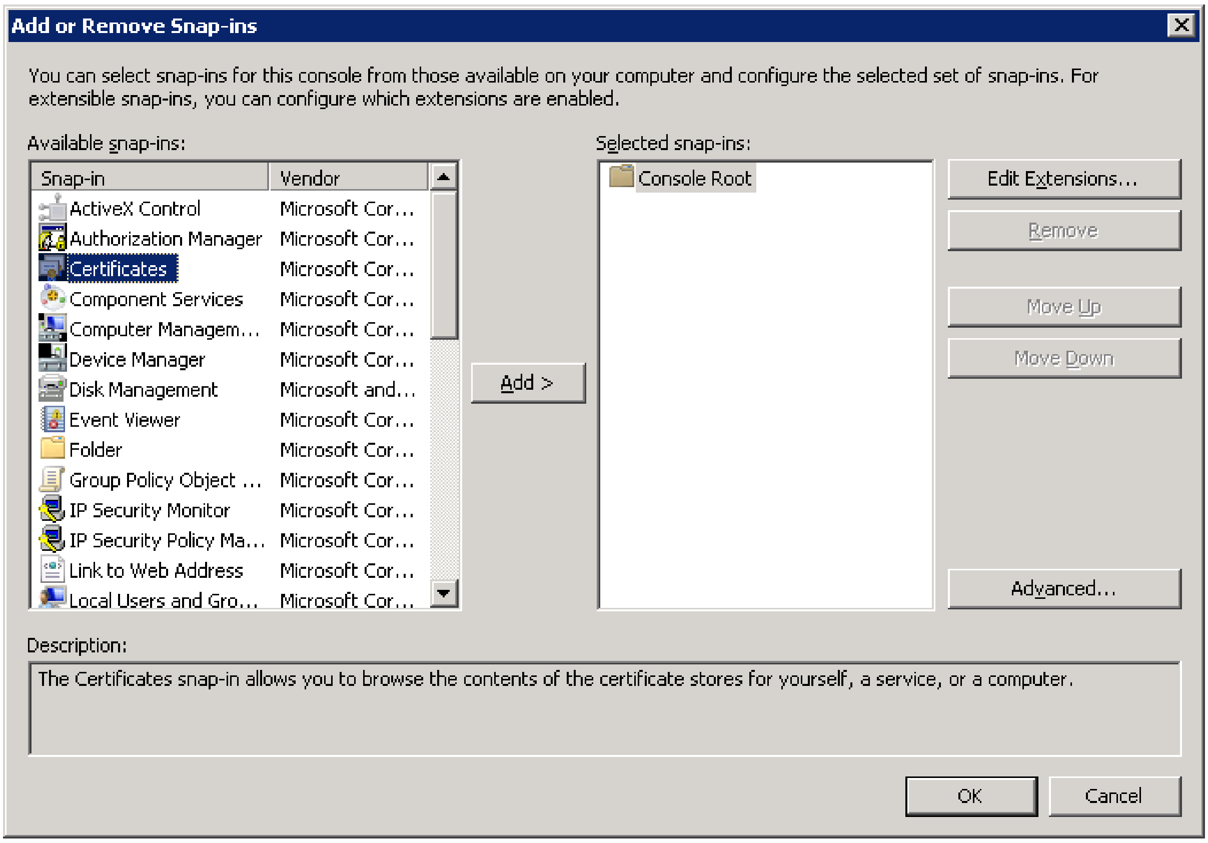
Rim nodes maintain local log files that may be worth backing up in order to identify and investigate issues. It may also be worth backing up any general operating system data that may be required.

6.4 Backing up certificates

When you install Qlik Sense, you should create a backup of the certificates on the central node.

Do the following:

1. Open a command prompt, and launch the Microsoft Management Console (mmc) as the user that runs the Qlik Sense services.
2. Select **File**>**Add/Remove Snap-in**.
3. Double-click **Certificates**.



Add or Remove Snap-ins

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy Ma...	Microsoft Cor...
Link to Web Address	Microsoft Cor...
Local Users and Gro...	Microsoft Cor...

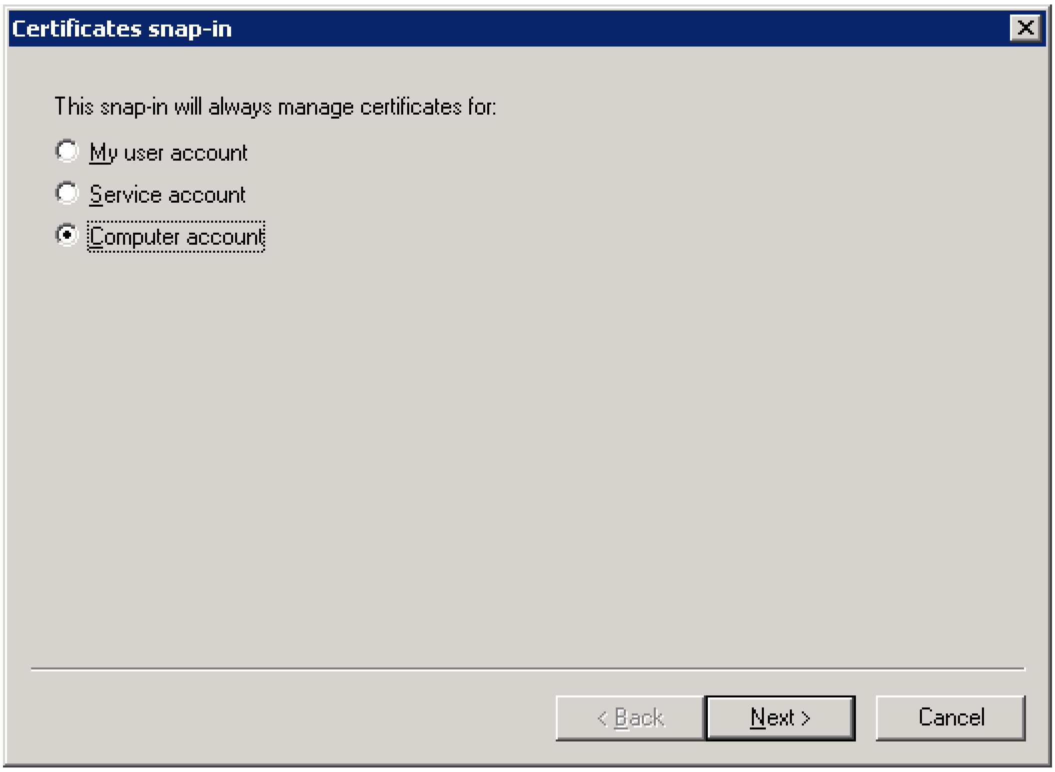
Selected snap-ins:

Console Root

Description:

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

4. Select **Computer account** and click **Next**.

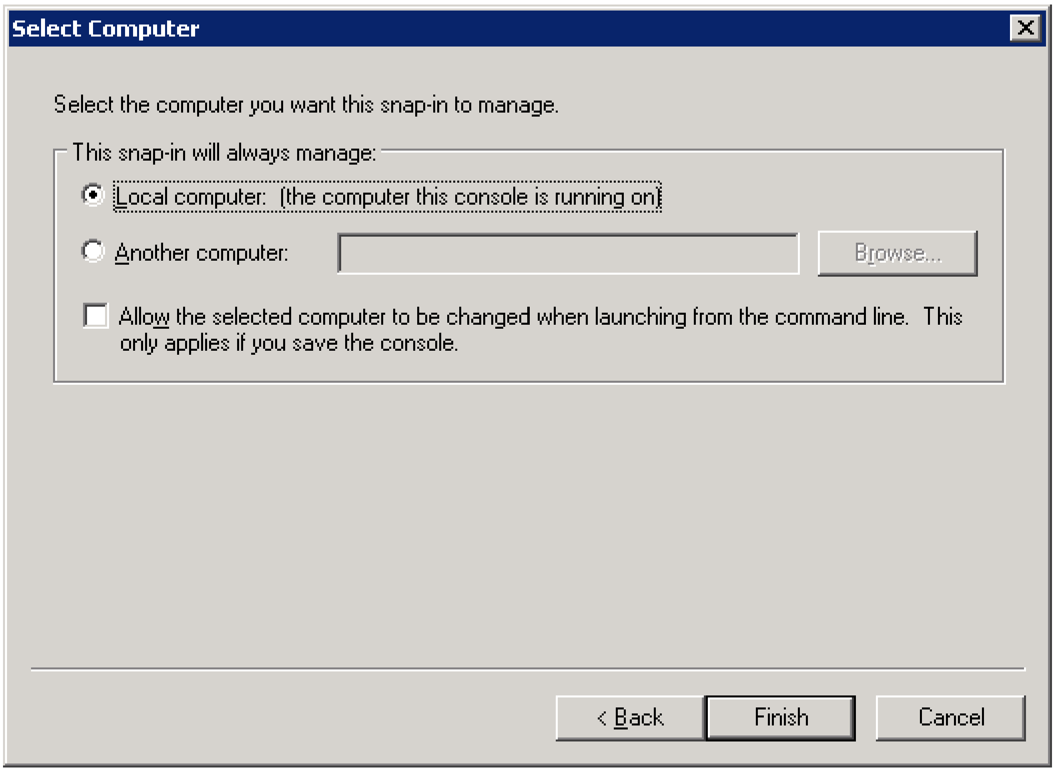


Certificates snap-in

This snap-in will always manage certificates for:

○ My user account
○ Service account
● Computer account

5. Select **Local computer** and click **Finish**.



Select Computer

Select the computer you want this snap-in to manage.

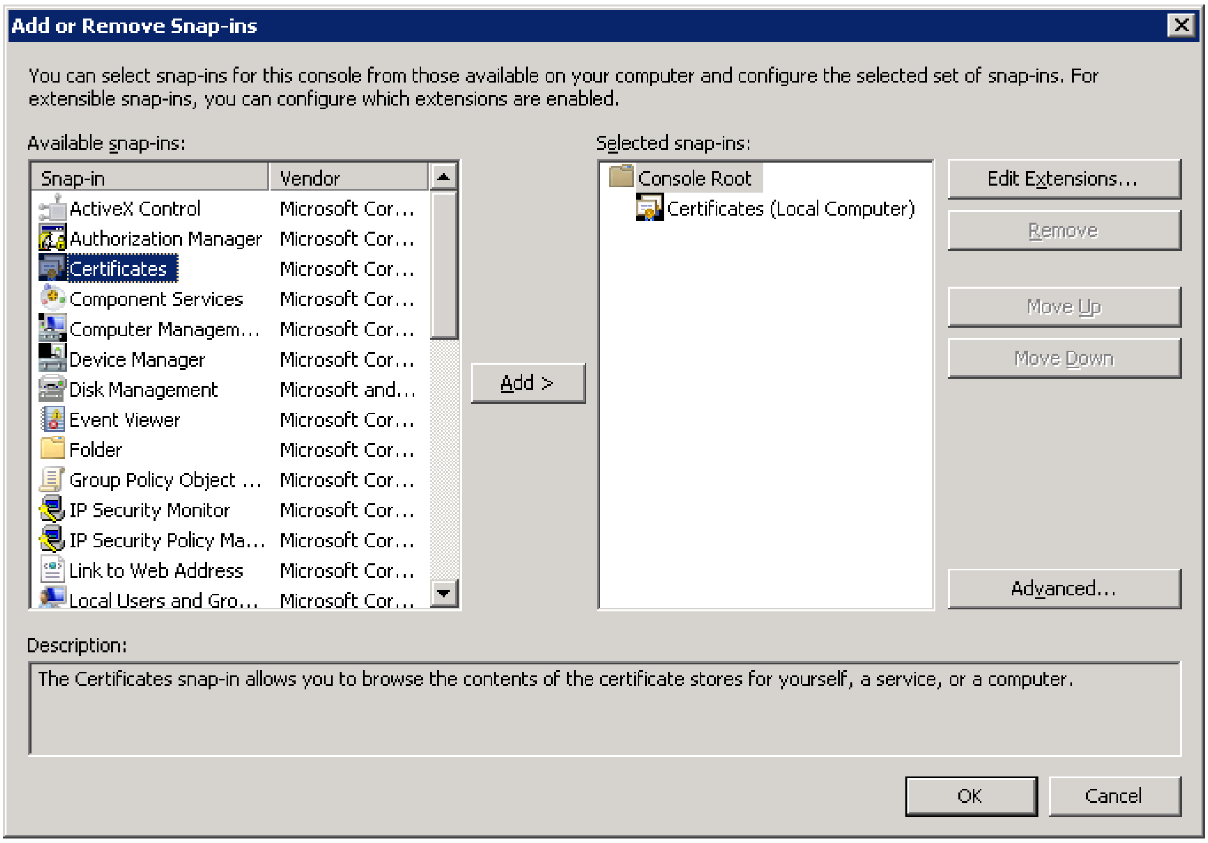
This snap-in will always manage:

● Local computer: (the computer this console is running on)

○ Another computer:

□ Allow the selected computer to be changed when launching from the command line. This only applies if you save the console.

6. Double-click **Certificates**.



Add or Remove Snap-ins

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy Ma...	Microsoft Cor...
Link to Web Address	Microsoft Cor...
Local Users and Gro...	Microsoft Cor...

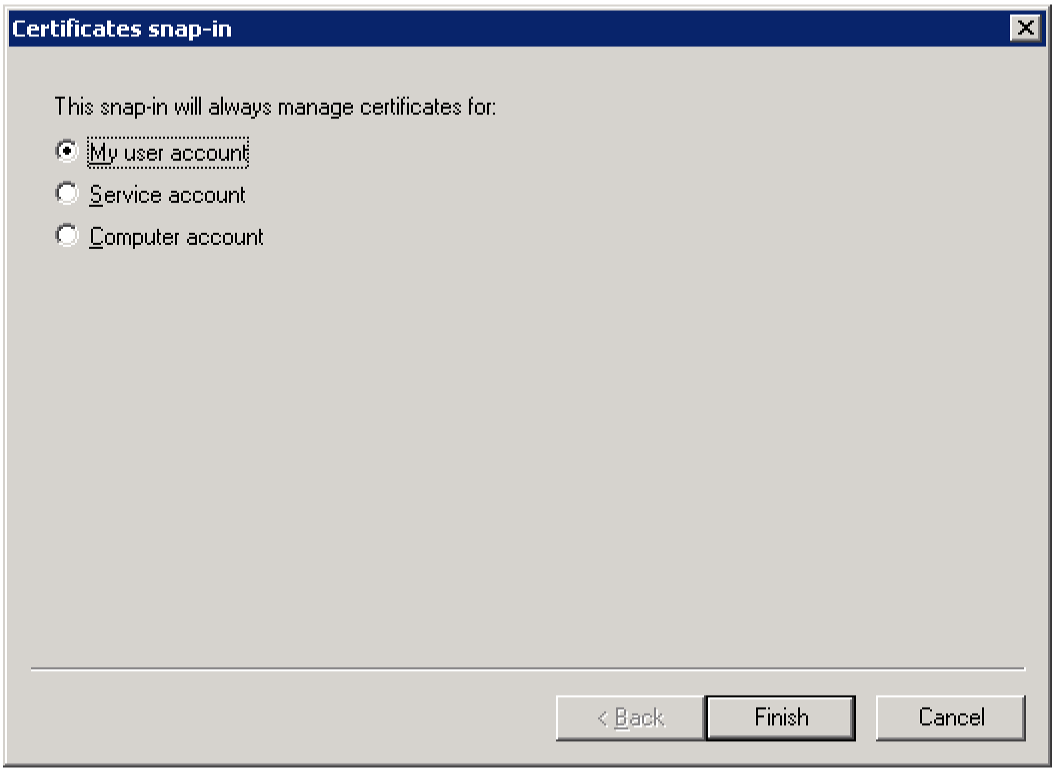
Selected snap-ins:

- Console Root
  - Certificates (Local Computer)

Description:

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

7. Select **My user account** and click **Finish**.



Certificates snap-in

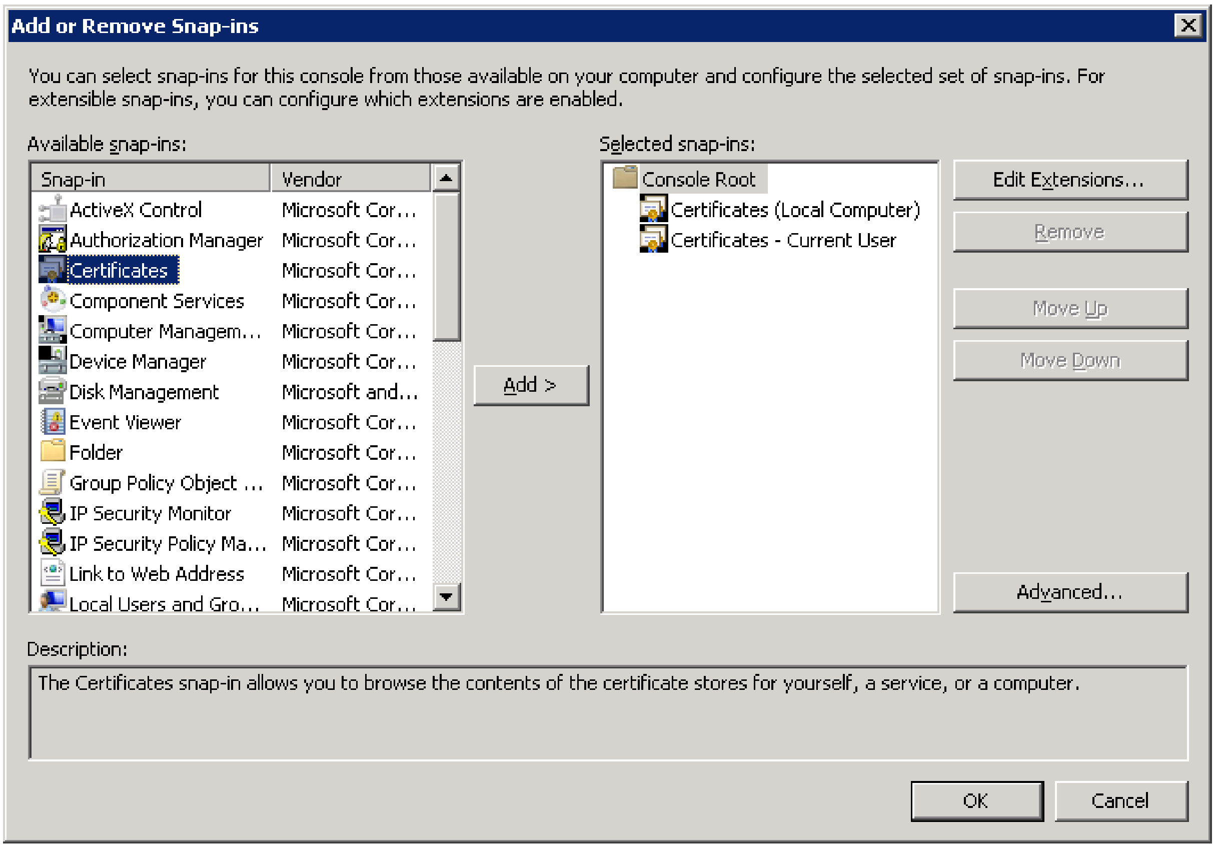
This snap-in will always manage certificates for:

● My user account

○ Service account

○ Computer account

8. Click **OK**.



Add or Remove Snap-ins

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy Ma...	Microsoft Cor...
Link to Web Address	Microsoft Cor...
Local Users and Gro...	Microsoft Cor...

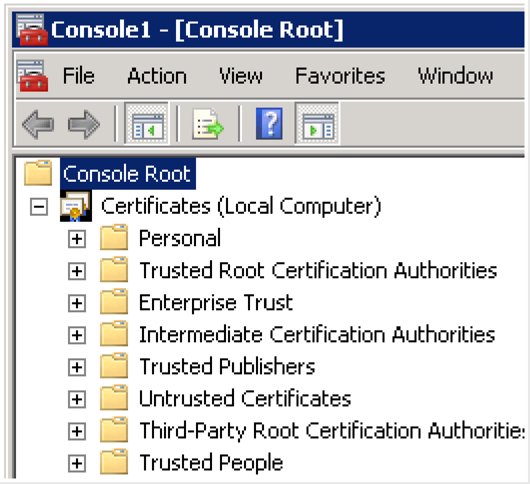
Selected snap-ins:

- Console Root
  - Certificates (Local Computer)
  - Certificates - Current User

Description:

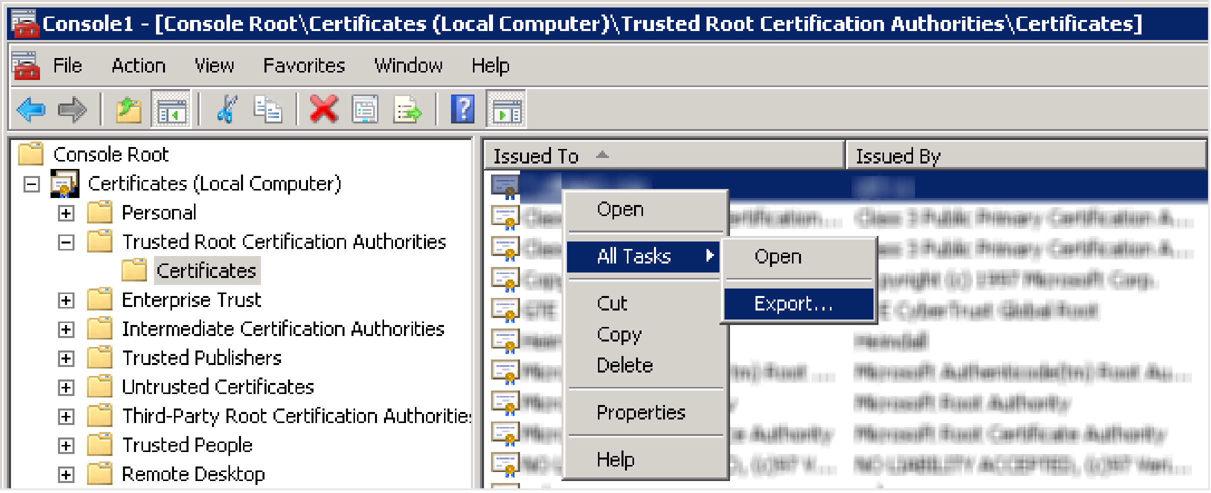
The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

9. Expand **Certificates (Local Computer)** in the left panel.

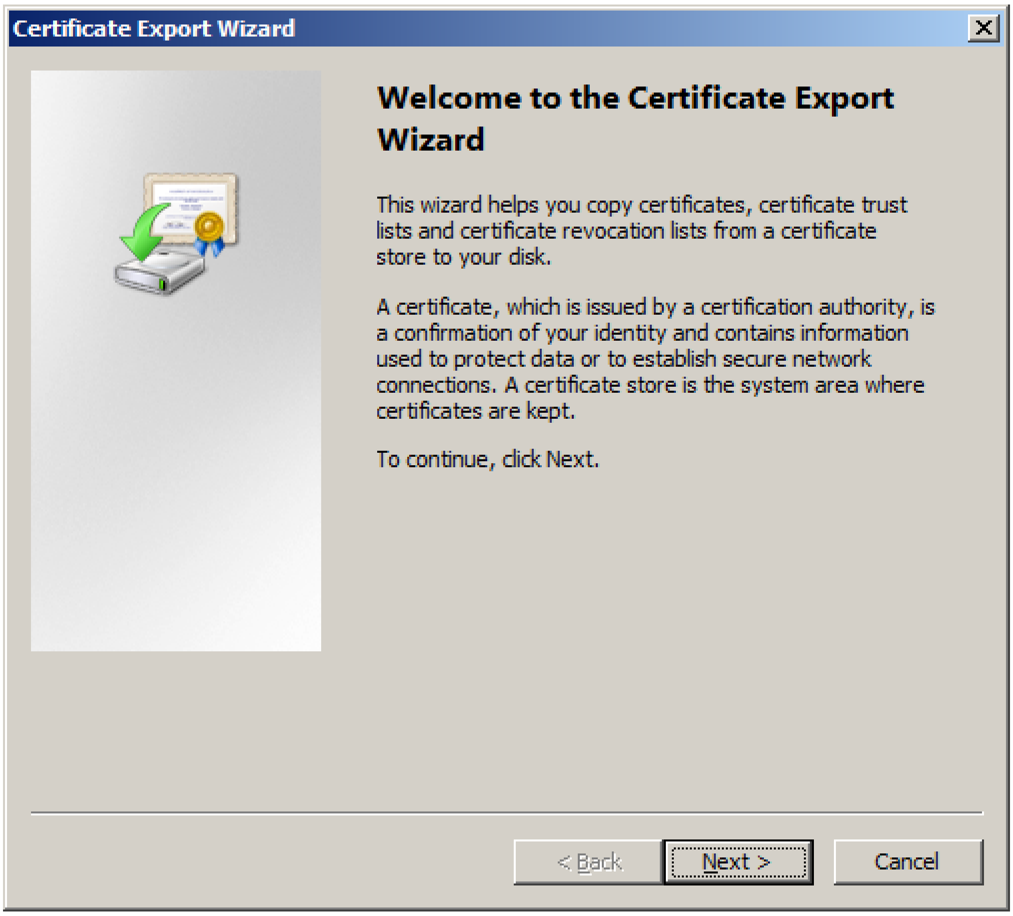

Console1 - [Console Root]
File
Action
View
Favorites
Window
Console Root
Certificates (Local Computer)
Personal
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authoritie:
Trusted People

10. Expand the **Trusted Root Certification Authorities** folder and select the **Certificates** folder.

11. Right-click the certificate that is Certificate Authority (CA) for all nodes in the Qlik Sense site and select **All Tasks**>**Export**. The CA is named *<computer_that_issued_the_certificate>-CA* by default.


Console1 - [Console Root\Certificates (Local Computer)\Trusted Root Certification Authorities\Certificates]
File
Action
View
Favorites
Window
Help
Console Root
Certificates (Local Computer)
Personal
Trusted Root Certification Authorities
Certificates
Enterprise Trust
Intermediate Certification Authorities
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authoritie:
Trusted People
Remote Desktop
Issued To
Issued By
Open
All Tasks
Cut
Copy
Delete
Properties
Help
Open
Export...

12. Click **Next**.



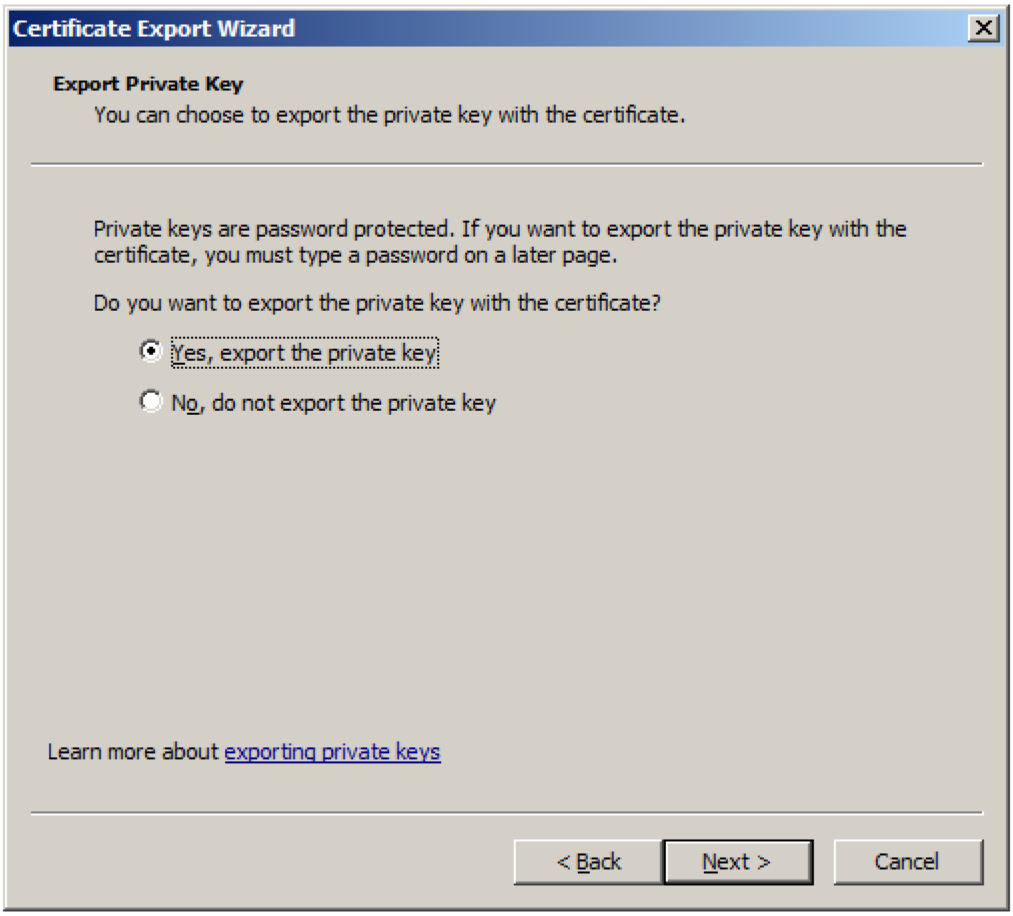
Welcome to the Certificate Export Wizard

This wizard helps you copy certificates, certificate trust lists and certificate revocation lists from a certificate store to your disk.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

13. Select **Yes, export the private key** and click **Next**.



Export Private Key

You can choose to export the private key with the certificate.

Private keys are password protected. If you want to export the private key with the certificate, you must type a password on a later page.

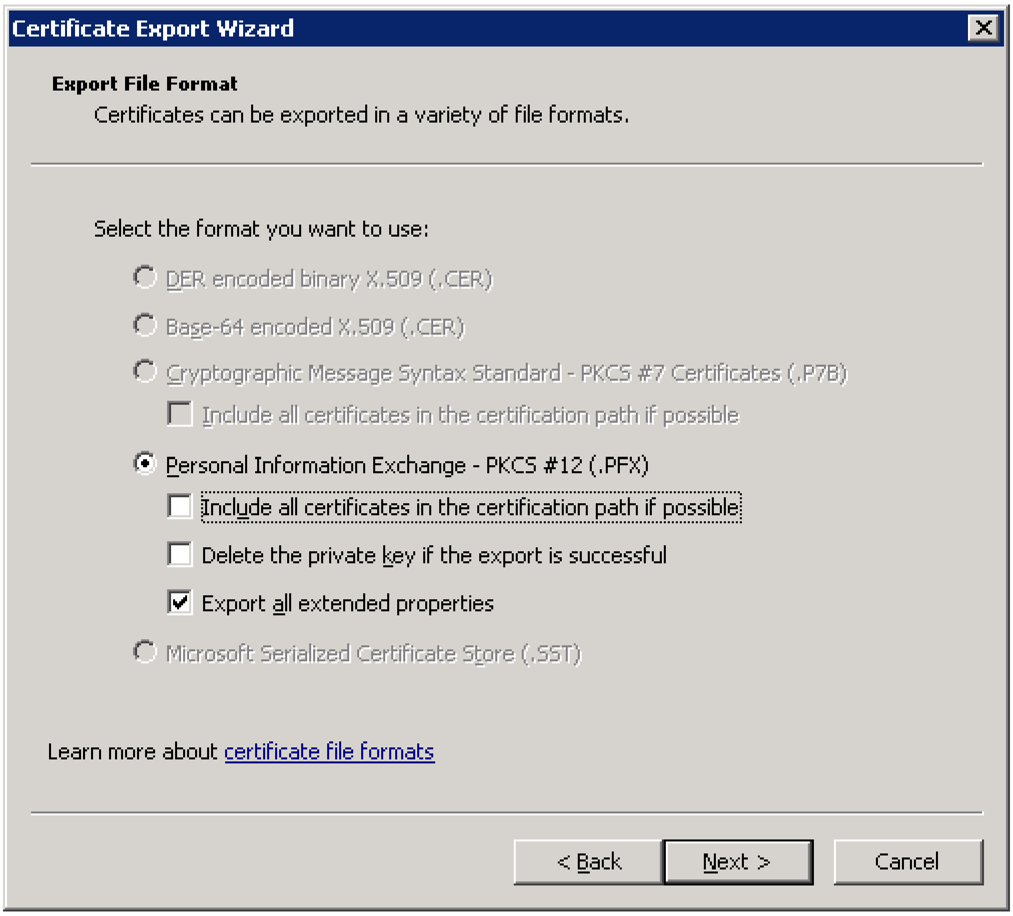
Do you want to export the private key with the certificate?

● Yes, export the private key

○ No, do not export the private key

Learn more about exporting private keys

14. Select **Personal Information Exchange**.
15. Tick the **Export all extended properties** box and then click **Next**.



Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)

○ Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

● Personal Information Exchange - PKCS #12 (.PFX)

□ Include all certificates in the certification path if possible

□ Delete the private key if the export is successful

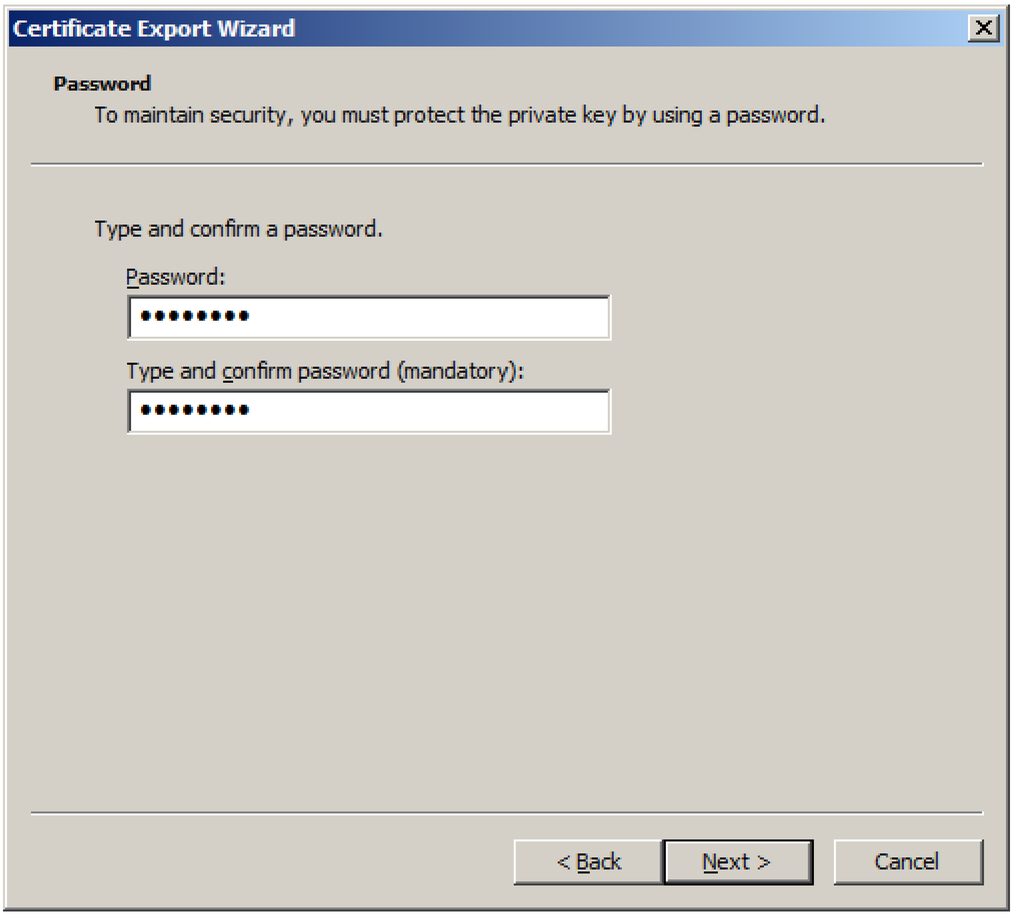
☑ Export all extended properties

○ Microsoft Serialized Certificate Store (.SST)

Learn more about certificate file formats

16. Enter and confirm a password. Then click **Next**.

    The password is needed when importing the certificate.



Certificate Export Wizard

Password

To maintain security, you must protect the private key by using a password.

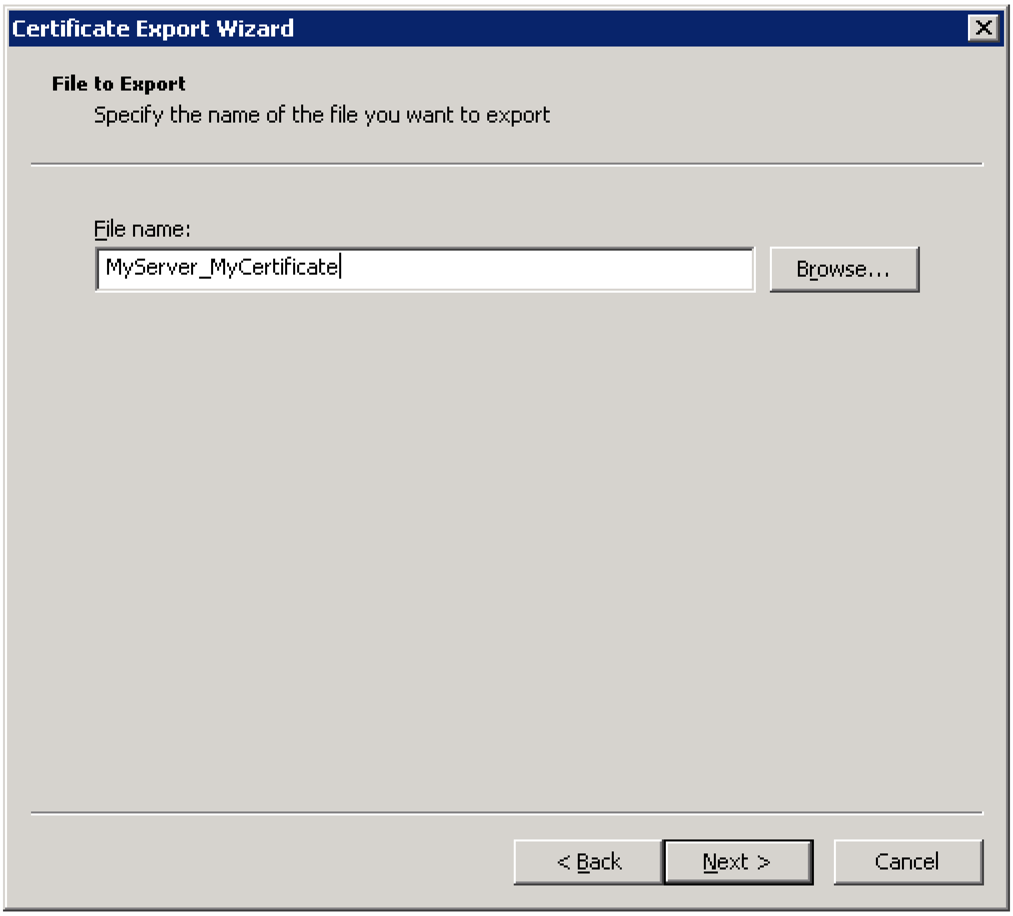
Type and confirm a password.

Password: ●●●●●●●●●

Type and confirm password (mandatory): ●●●●●●●●●

17. Enter a file name for the *.pfx* file and click **Next**.

It is recommended to include the server name in the file name to avoid confusion with other certificate files.



Certificate Export Wizard

File to Export

Specify the name of the file you want to export

File name: MyServer_MyCertificate

18. Click **Finish**.

    The *.pfx* file that contains the CA for all nodes in the Qlik Sense site is stored in the selected location.

19. Starting at step 11, repeat the procedure and export the server certificate (that is, the SSL certificate), which is located under **Certificates (Local Computer)**>**Personal**>**Certificates**. The server certificate a) has the same name as the Domain Name System (DNS) name of the machine, and b) is signed by the CA for all nodes in the site.
20. Starting at step 11, repeat the procedure and export the client certificate (that is, the ID of the client), which is located under **Certificates - Current User**>**Personal**>**Certificates**. The client certificate is named *QlikClient* and is signed by the CA for all nodes in the site.
21. Close the MMC console.

    No changes need to be saved.

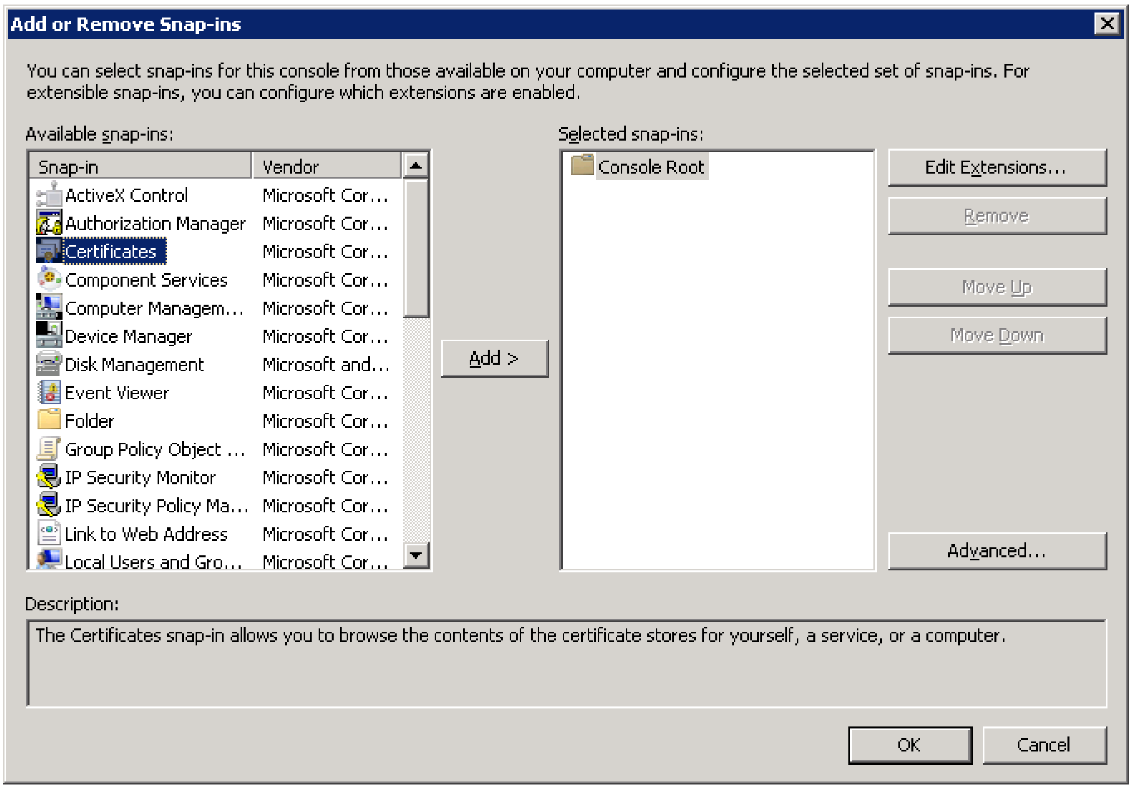
6.5 Restoring certificates

In case of a system crash, the certificates may need to be restored on the central node of your Qlik Sense site.

Do the following:

1. Open the Task Manager in Microsoft Windows and stop all Qlik Sense services except the Qlik Sense Repository Database (QRD) service.

2. Open a command prompt, and launch the Microsoft Management Console (mmc) as the user that runs the Qlik Sense services..
3. Select **File**>**Add/Remove Snap-in**.
4. Double-click **Certificates**.

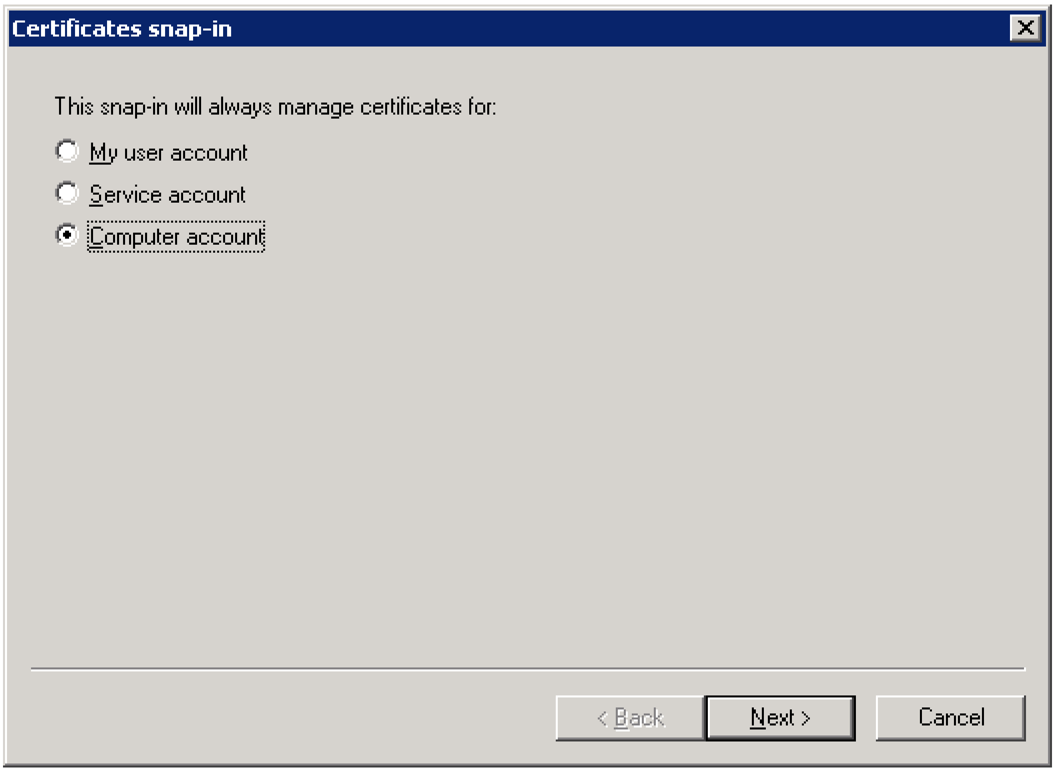


You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy Ma...	Microsoft Cor...
Link to Web Address	Microsoft Cor...
Local Users and Gro...	Microsoft Cor...

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

5. Select **Computer account** and click **Next**.



Certificates snap-in

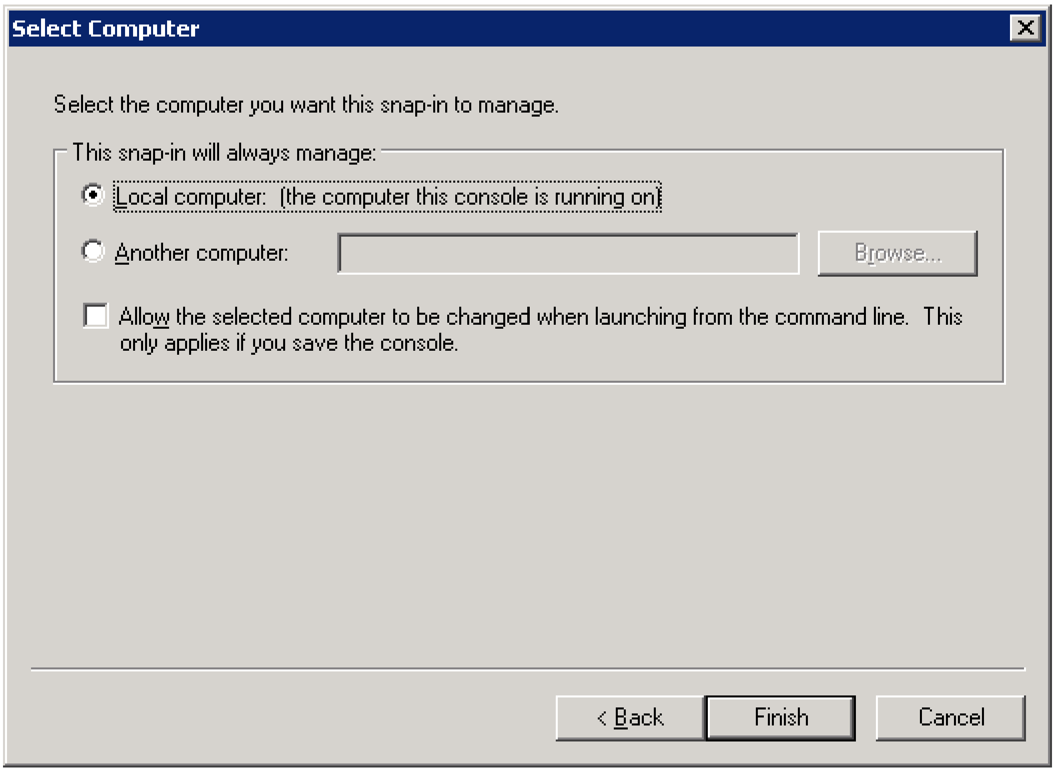
This snap-in will always manage certificates for:

○ My user account

○ Service account

● Computer account

6. Select **Local computer** and click **Finish**.



Select Computer

Select the computer you want this snap-in to manage.

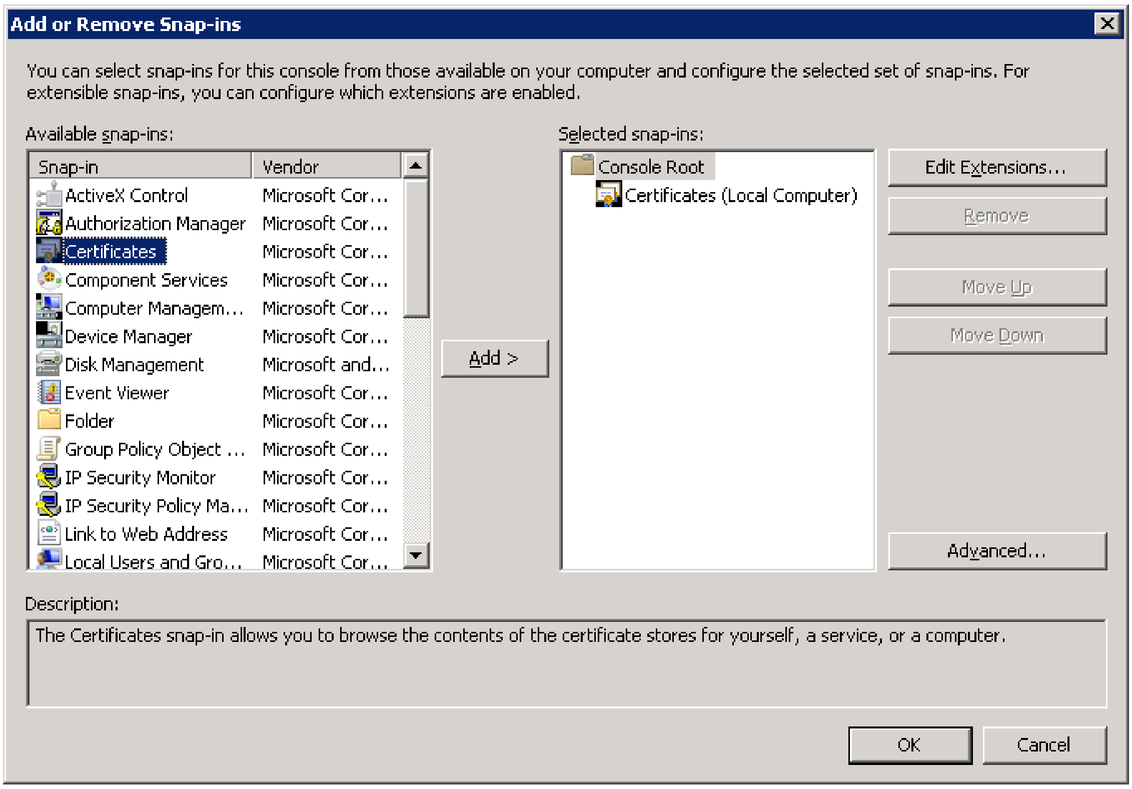
This snap-in will always manage:

● Local computer: (the computer this console is running on)

○ Another computer:

□ Allow the selected computer to be changed when launching from the command line. This only applies if you save the console.

7. Double-click **Certificates**.



You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

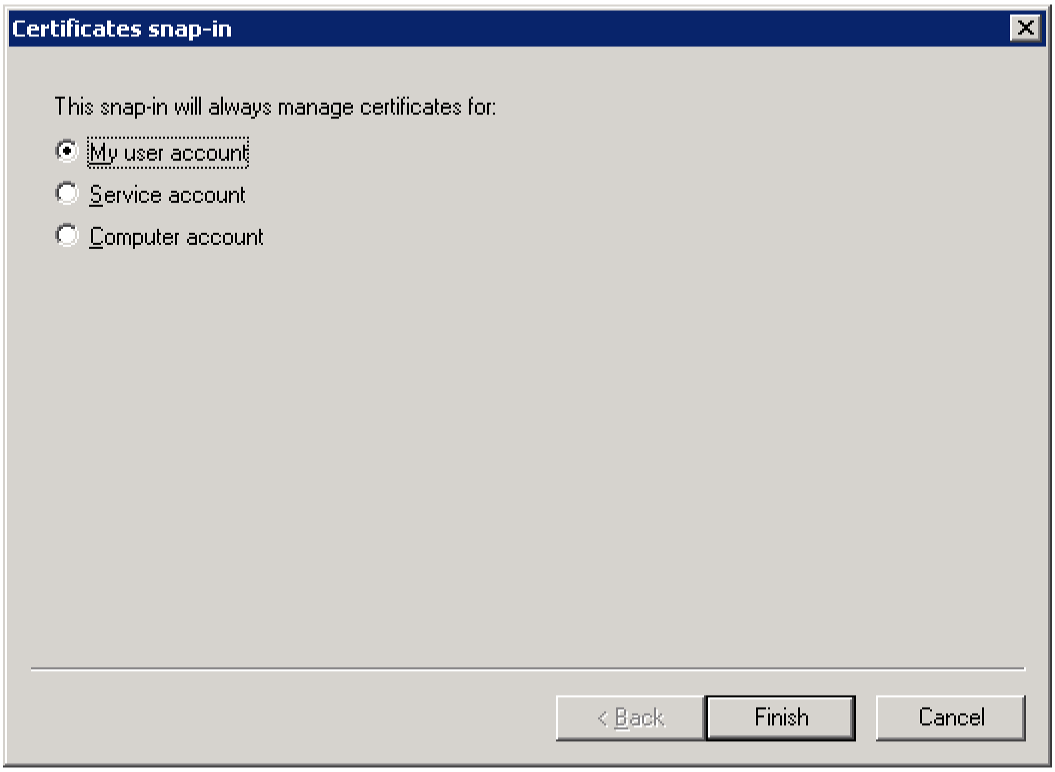
Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy Ma...	Microsoft Cor...
Link to Web Address	Microsoft Cor...
Local Users and Gro...	Microsoft Cor...

Selected snap-ins:

- Console Root
  - Certificates (Local Computer)

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

8. Select **My user account** and click **Finish**.



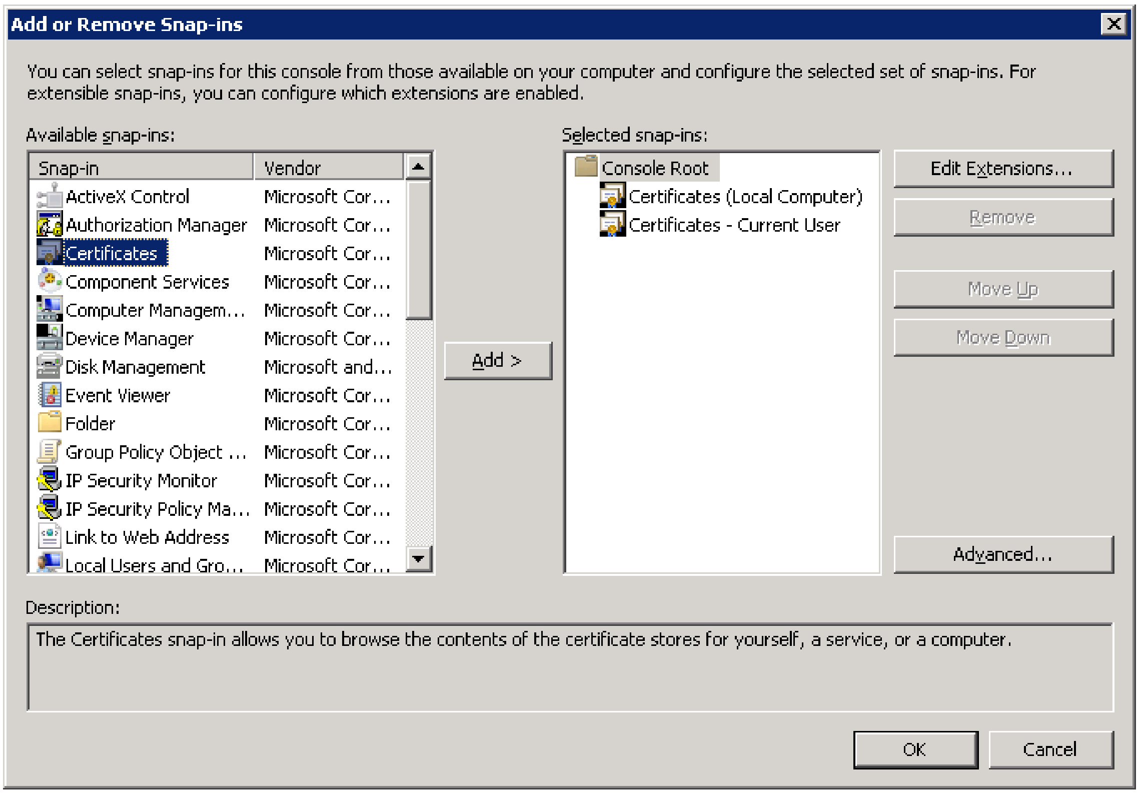
This snap-in will always manage certificates for:

● My user account

○ Service account

○ Computer account

9. Click **OK**.



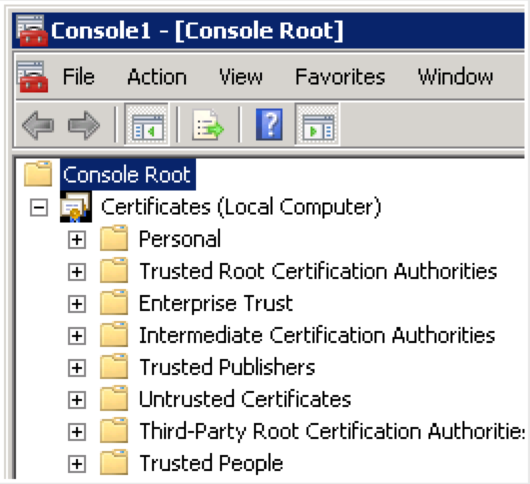
You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy Ma...	Microsoft Cor...
Link to Web Address	Microsoft Cor...
Local Users and Gro...	Microsoft Cor...

Console Root
- Certificates (Local Computer)
- Certificates - Current User

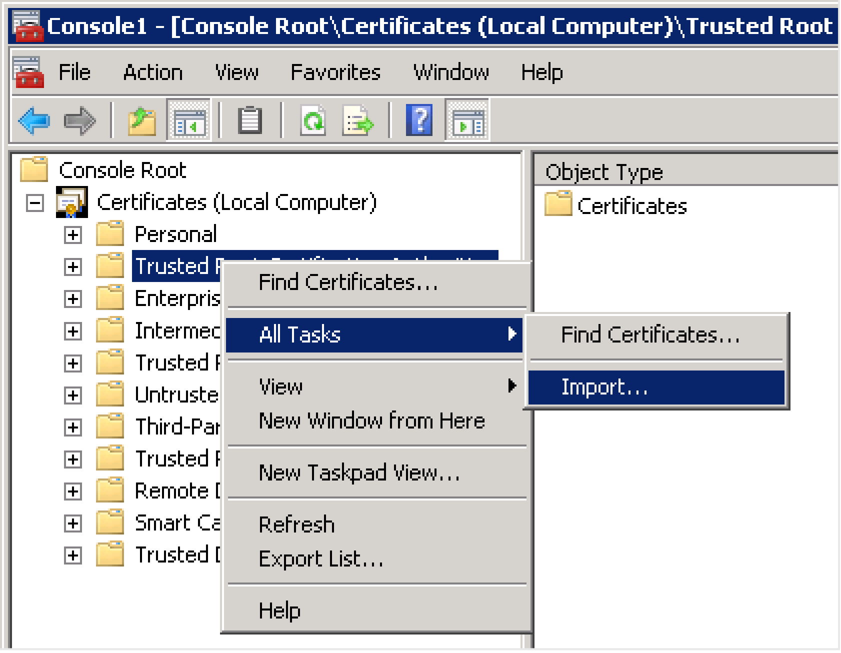
The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

10. Expand **Certificates (Local Computer)** in the left panel.

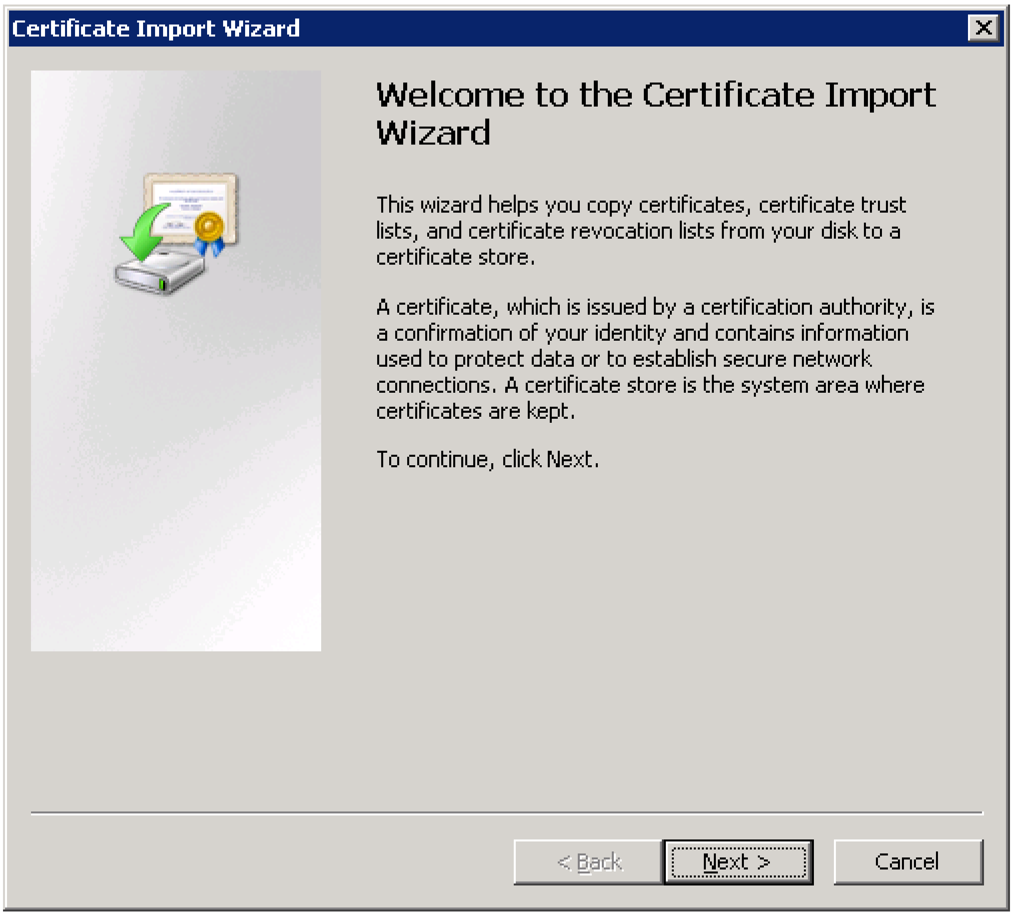


Console Root
- Certificates (Local Computer)
  - Personal
  - Trusted Root Certification Authorities
  - Enterprise Trust
  - Intermediate Certification Authorities
  - Trusted Publishers
  - Untrusted Certificates
  - Third-Party Root Certification Authoritie:
  - Trusted People

11. Right-click the **Trusted Root Certification Authorities** folder and select **All Tasks**>**Import**.



12. Click **Next**.


Certificate Import Wizard

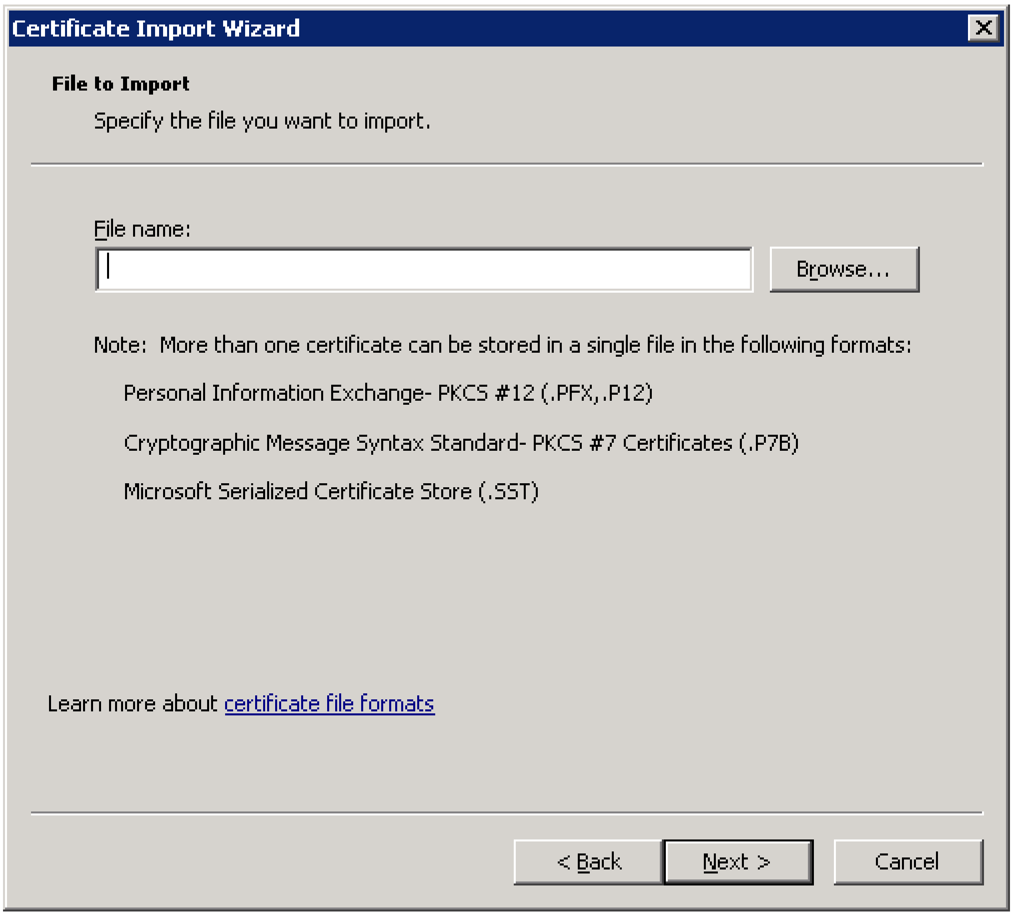
Welcome to the Certificate Import Wizard

This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

13. Browse to the file that contains the backed up Certificate Authority (CA) for all nodes in the site and then click **Next**. The CA is named *<computer_that_issued_the_certificate>-CA* by default.



Certificate Import Wizard

File to Import

Specify the file you want to import.

File name:

Note: More than one certificate can be stored in a single file in the following formats:

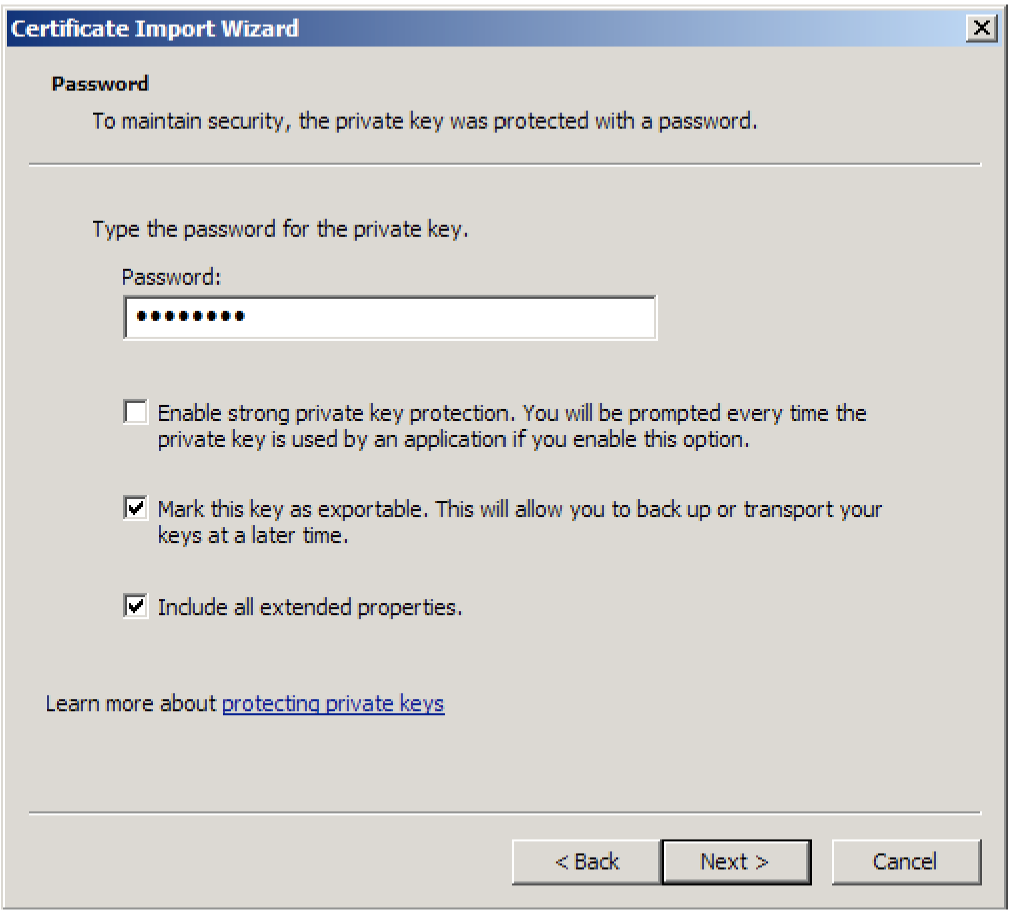
Personal Information Exchange- PKCS #12 (.PFX,.P12)

Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialized Certificate Store (.SST)

Learn more about certificate file formats

14. Enter the password for the *.pfx* file (that is, the password that was given when the file was exported).
15. Select **Mark this key as exportable** and **Include all extended properties**. Then click **Next**.



Certificate Import Wizard

Password

To maintain security, the private key was protected with a password.

Type the password for the private key.

Password:

••••••••••

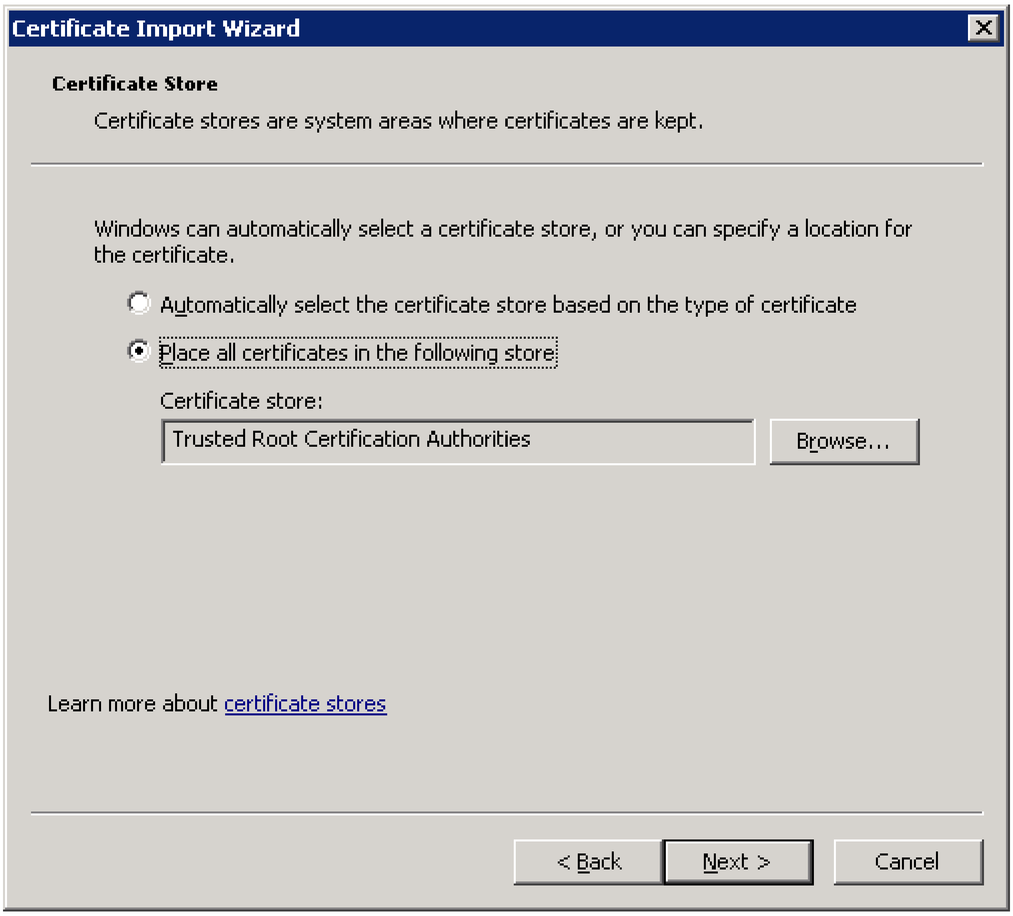
□ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

☑ Mark this key as exportable. This will allow you to back up or transport your keys at a later time.

☑ Include all extended properties.

Learn more about protecting private keys

16. Select **Place all certificates in the following store** and click **Next**.



Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

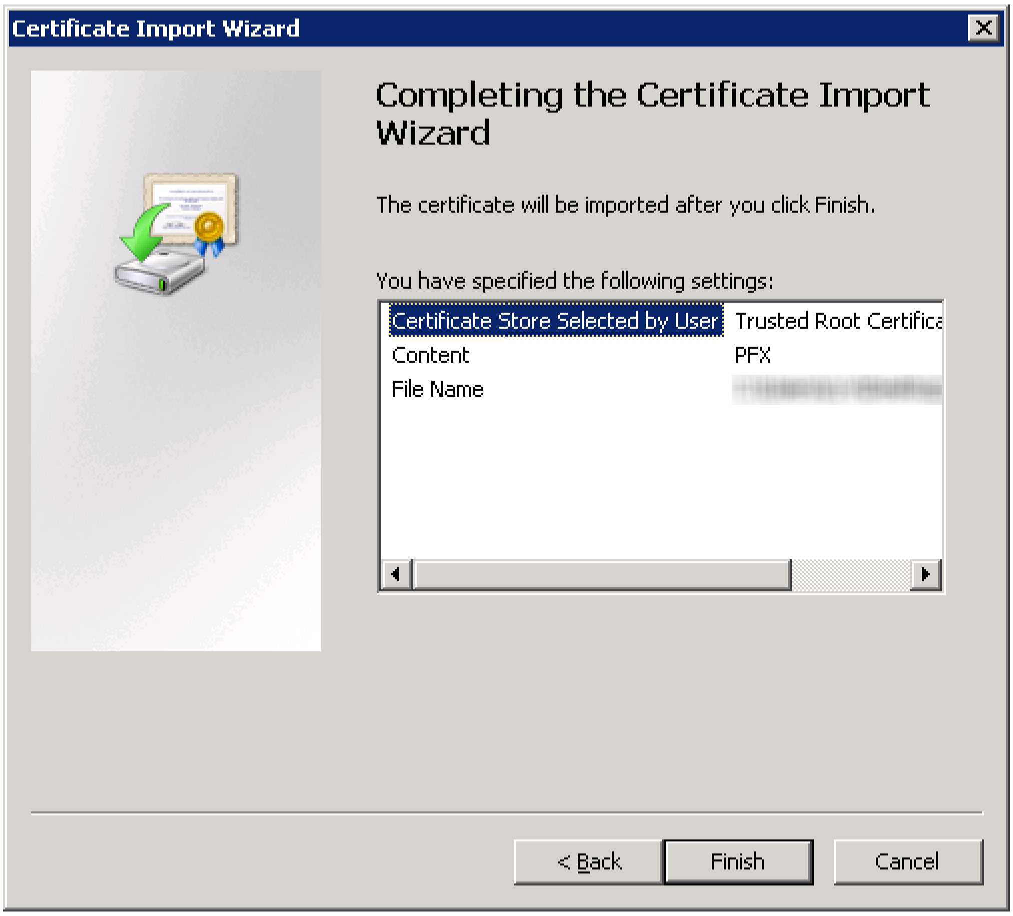
○ Automatically select the certificate store based on the type of certificate

● Place all certificates in the following store

Certificate store: Trusted Root Certification Authorities

Learn more about certificate stores

17. Click **Finish**.



Completing the Certificate Import Wizard

The certificate will be imported after you click Finish.

You have specified the following settings:

Certificate Store Selected by User	Trusted Root Certifica
Content	PFX
File Name	

18. Click the **Refresh** button (⟳) and check that the imported CA is available in the **Trusted Root Certification Authorities** folder.
19. Starting at step 11, repeat the procedure and import the server certificate to **Certificates (Local Computer)**>**Personal**>**Certificates**. The server certificate a) has the same name as the Domain Name System (DNS) name of the machine, and b) is signed by the CA for all nodes in the site.
20. Starting at step 11, repeat the procedure and import the client certificate (that is, the ID of the client) to **Certificates - Current User**>**Personal**>**Certificates**. The client certificate is named *QlikClient* and is signed by the CA for all nodes in the site.
21. Close the MMC console.

    No changes have to be saved.

22. Start the Qlik Sense services. If the services are started manually, start them in the following order:

    > ⚠ *If you are restoring the certificates as part of the Restoring a Qlik Sense site (page 204) procedure, do not start the Qlik Sense services.*

    a. Qlik Sense Repository Service (QRS)
    If the user running Qlik Sense services is not local administrator on the machine, you need to start Repository.exe from an elevated command prompt using the -bootstrap parameter.
    Services (page 22)

   b. Qlik Sense Proxy Service (QPS), Qlik Sense Engine Service (QES), Qlik Sense Scheduler Service (QSS), and Qlik Sense Printing Service (QPR) in no specific order

The order is important because the QRS is dependent on the QRD and the rest of the services are dependent on the QRS.

6.6 Backing up a Qlik Sense site

Backing up a Qlik Sense site includes backing up the following:

- Repository database: The database contains all configuration data for the site
- Log data: The centralized logging database
- The file share: The shared folder in that contains application data, such as data models used in the Qlik Sense apps, and QVD files

To restore your Qlik Sense deployment you will also need a back up of your Qlik Sense certificates. For more information, see *Backing up certificates (page 183)*.

You must perform this backup procedure on each of the nodes that host the components listed above.

> *Rim nodes maintain local log files that may be worth backing up in order to identify and investigate issues. It may also be worth backing up any general operating system data that may be required.*

Do the following:

1. Stop all Qlik Sense services except the Qlik Sense Repository Database (QRD), on every node in your deployment.
2. Make a backup of the repository database by creating a database dump file:
   a. Open a Command Prompt in Microsoft Windows.
   b. Navigate to the location where the PostgreSQL repository database is installed.

   > *If your deployment includes a local database on the central node that was installed using the Qlik Sense setup program, the location will be: %ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\<database version>\bin.*

   > *If you installed PostgreSQL manually, the location will be: %ProgramFiles%\PostgreSQL\<database version>\bin.*

   c. Run the following command:
   ```
   pg_dump.exe -h localhost -p 4432 -U postgres -b -F t -f "c:\QSR_backup.tar" QSR
   ```
   If you are prompted for the PostgreSQL super user password, enter the password that was created during the Qlik Sense setup.

> *To avoid being prompted for the password (for example, if you want to automate the Qlik Sense backup process), you can use the pgpass functionality in PostgreSQL. See the PostgreSQL documentation for more information.*

3. Make a backup of all of the content in the file share.
4. If you have centralized logging configured, make a backup of the centralized logging database by running the following command:
   ```
   pg_dump.exe -h localhost -p 4432 -U postgres -b -F t -f "c:\QLogs_backup.tar" QLogs
   ```
5. Make a backup of any locations where content that supports the Qlik Sense environment may be kept (for example, QVD files created by load scripts).
6. Restart the Qlik Sense services.

Backing up the Qlik Sense Repository Database after uninstalling Qlik Sense

> *We recommend creating your database dump file before you uninstall Qlik Sense.*

If you uninstall Qlik Sense before creating the database dump file, do the following:

1. Copy the PostgreSQL folder from *%ProgramData%\Qlik\Sense\Repository\PostgreSQL* to a temporary location outside of the *%ProgramData%* folder.
2. Download and install PostgreSQL version 9.6 from the PostgreSQL website. See: *Installing and configuring PostgreSQL (page 98)*.
3. Open a Command Prompt in Microsoft Windows.

   > *The pg_ctl.exe command should not be run as an administrator.*

4. Navigate to the location where the PostgreSQL repository database is installed.

   > *If your deployment includes a local database on the central node that was installed using the Qlik Sense setup program, the location will be: %ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\<database version>\bin.*

   > *If you installed PostgreSQL manually, the location will be: %ProgramFiles%\PostgreSQL\<database version>\bin.*

4. Run the following commands:
   a. `pg_ctl.exe start -w -D "C:\SenseDB\9.6"`
   b. `set PGUSER=postgres`

c. `set PGPASSWORD=password`

d. `pg_dumpall.exe > [<path to dump file>]`

e. `pg_ctl.exe stop -w -D "C:\SenseDB\9.6"`

If you are prompted for the PostgreSQL super user password, enter the password that was created during the Qlik Sense setup.

> *To avoid being prompted for the password (for example, if you want to automate the Qlik Sense backup process), you can use the pgpass functionality in PostgreSQL. See the PostgreSQL documentation for more information.*

6.7 Restoring a Qlik Sense site

Consider the following when restoring a site:

- Qlik Sense software
- If you want to restore the site to a central node with a new hostname, see *Restoring a Qlik Sense site to a machine with a different hostname (page 205)*.
- Repository database: The database contains all configuration data for the site.
- Certificates for the Qlik Sense services: The certificates are used to encrypt the traffic between the services and the users. Make sure to backup the certificates in order not to lose any encrypted data (for example, passwords for data connections).
- Log data
- Application data: The data models in the Qlik Sense apps.
- Any content that supports the apps (for example, QVD files)

When performing the procedure below you must log in using an account that had the Root Admin role when the site was backed up. If you log in using a local admin account and the machine name is different, your permissions will not follow through.

Do the following:

1. Restore the certificates used to secure the Qlik Sense services.

   Restoring certificates (page 192)

2. Install Qlik Sense on the computer where you plan to restore.

   > *Make sure to deselect* ***Start the Qlik Sense services when the installation has completed*** *during the installation setup.*

3. Start the Qlik Sense Repository Database (QRD).

4. Restore the repository database:
    a. Place the backed up repository database on the machine targeted for the restore.
    b. Open a Command Prompt with administrator privileges in Microsoft Windows.
    c. Run the following commands to restore the repository database (adjust the paths as needed):
        i. *cd "%ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\<database version>\bin"*
        ii. *createdb -h localhost -p 4432 -U postgres -T template0 QSR*
        If the command fails because a database already exists, run the following command and then repeat the *createdb* command:
        dropdb -h localhost -p 4432 -U postgres QSR
        iii. *pg_restore.exe -h localhost -p 4432 -U postgres -d QSR "c:\QSR_backup.tar"*
5. Restore log and application data to the file share used for storage of log and application data.
6. Restore any supporting content to its original location as required.
7. Start the Qlik Sense services. If the services are started manually, start them in the following order:
    a. Qlik Sense Repository Service (QRS)
    If the user running Qlik Sense services is not local administrator on the machine, you need to start Repository.exe from an elevated command prompt using the -bootstrap parameter.
    Services (page 22)
    b. Qlik Sense Proxy Service (QPS), Qlik Sense Engine Service (QES), Qlik Sense Scheduler Service (QSS), and Qlik Sense Printing Service (QPR) in no specific order

    The order is important because the QRS is dependent on the QRD and the rest of the services are dependent on the QRS.

Restoring a Qlik Sense site to a machine with a different hostname

You can restore a Qlik Sense site to a machine with a host name that is different from the site that you backed up. However, if the machine is a central node in a multi-node site, you need to slightly adapt the procedure as follows:

- All rim nodes need to be reset, that is, you need to remove them and then add them again.
- Run *repository.exe -bootstrap -standalone -restorehostname* from an elevated command prompt to start the Qlik Sense Repository Service (QRS). When QRS is up and running, restart the QRS without *-restorehostname*.
  See: *Services (page 22)*

> *The parameter '-standalone' means that Repository runs as a normal executable process (as opposed to running as a service, and registering in Windows Service Manager).*

When restoring your site, you must log in using an account that had the Root Admin role when the site was backed up. If you log in using a local admin account and the machine name is different, your permissions will not follow through.

Backing up the current server

Perform the following steps on the server machine that you want to restore to a different server.

1. Create a local folder called *Backup* or something similar to store the files you want to restore later. For example: *C:\ProgramData\Qlik\Sense\Repository\Backup*.
2. In Qlik Management Console, go to the *Certificates* section and export new certificates using the FQDN of the new server. Ensure tht you include the private key and the certificates must be in the Windows format.
3. Backup your certificates using the Microsoft Management Console (MMC). For detailed steps, see *Backing up certificates (page 183)*. On your original server, export the following certificates from the QMC to your *Backup* folder. Ensure that you export the private key.
   - `client`
   - `root`
   - `server`
4. On the service cluster, open the your backup location QMC, **Cluster Settings**, copy the full UNC path of the *QlikShare* (the full path name including back slashes), for example *\\<computer_name>\QlikShare*
5. In Windows, **Services**, stop all services, except the Qlik Sense Repository Database.
6. Backup the **QSR** database. For detailed instructions, see *Backing up a Qlik Sense site (page 202)*. Copy the database dump file to your backup folder.
7. Copy all sub folders from the *QlikShare* (the path specified in the **Service Cluster** section of the QMC) to your backup folder.
8. Copy your *Backup* folder from the current folder to the same location on your target machine.

Restoring to a machine with a different host name

Perform the following steps on the target server machine, where you want to restore Qlik Sense.

1. Create a *QlikShare* folder on the target server computer. For example, create a folder called *QlikShare* on the C:\ drive.
2. Move your backup folder and database dump file to your target server machine.
3. Restore the the following backed up certificates from the old machine used to secure the Qlik Sense services:
   - `client`
   - `root`
   - `server`

   Use the MMC to import them to your target server. Ensure that you mark the certificates as exportable. For detailed steps, see *Restoring certificates (page 192)*
4. Install the same version of Qlik Sense on the computer where you plan to restore. Do not start the services until you have finished the configuration steps.

   > ⚠ *Make sure to deselect **Start the Qlik Sense services when the installation has completed** during the installation setup.*

5. Start the Qlik Sense Repository Database (QRD) service.

6. Restore the backup copy of the repository database:
   a. Open a Command Prompt with administrator privileges in Microsoft Windows.
   b. Run the following command to restore the repository database on a clean server:
      - `"pg_restore.exe -h localhost -p 4432 -U postgres -d QSR "c:\QSR_backup.tar"`

   > *You may need to adjust the path* `"c:\QSR_backup.tar"` *depending on where you backed up your database dump file.*

   If running this command on a server where a repository database may have been previously installed, you may get the following error message:

   ```
   pg_restore: [archiver (db)] connection to database "QSR" failed: FATAL: database "QSR"
   does not exist
   ```

   If you get this error, follow the steps below:

   i. Navigate to: *`cd "%ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\<database version>\bin"`*

   ii. Run `createdb -h localhost -p 4432 -U postgres -T template0 QSR`.
   If the command fails because a database already exists, you get the following error message: `createdb: database creation failed: ERROR: database "QSR" already exists`

   iii. Run `dropdb -h localhost -p 4432 -U postgres QSR` to drop the database.

   iv. Execute the *createdb* command again. `createdb -h localhost -p 4432 -U postgres -T template0 QSR`

7. Restore the log and application data to the *QlikShare* folder.
8. To launch Qlik Sense with the new hostname:
   a. Open a Command Prompt with administrator privileges in Microsoft Windows.
   b. Change the directory to the Repository installation path
      - Default path: *C:\"Program Files"\Qlik\Sense\Repository*
   c. Execute the following command:

   ```
   Repository.exe -bootstrap -standalone -restorehostname
   ```

   When the command has completed successfully check for erros in the logs and the following message is displayed:

   ```
   Bootstrap mode has terminated. Press ENTER to exit..
   ```

9. Start the Qlik Sense services. If the services are started manually, start them in the following order:
   - Qlik Sense Service Dispatcher
   - Qlik Logging Service
   - Qlik Sense Repository Service
   - Qlik Sense Proxy Service
   - Qlik Sense Engine Service
   - Qlik Sense Scheduler Service
   - Qlik Sense Printing Service
10. Try to access the QMC or the Hub to verify that the migration has been successful. Also, from the Qlik Management Console reload the monitoring apps to verify that your certificates have been installed correctly.

7 Security

Security in Qlik Sense consists of the following:

- Protection of the platform
  How the Qlik Sense platform itself is protected and how it communicates and operates.
- Authentication
  Who is the user and how can the user prove it? Qlik Sense uses standard authentication protocols (for example, Integrated Windows Authentication), HTTP headers, and ticketing to authenticate every user requesting access to data.
- Authorization
  What does the user have access to? Authorization is the procedure of granting or denying users access to resources.
- Auditing
  The Qlik Sense platform tracks changes in the repository database, provides comprehensive audit and security logging, and monitors applications.
- Confidentiality
  Qlik Sense protects confidentiality by:
  - encrypting network connections with Transport Layer Security (TLS)
  - leveraging the operating system file system and server access controls to protect content on Qlik Sense nodes
  - protecting memory using operating system controls
  - securing application access at the resource level
  - encrypting sensitive information (e.g. passwords and data connection strings)
  - protecting app data using data reduction
- Integrity
  Operating system controls like the file system are leveraged to provide integrity by protecting data at rest, encrypting sensitive information, and preventing data write back to the source system.
- Availability
  Qlik Sense deployed in a multi-node environment is designed for resiliency and reliability.

7.1 Protection of the platform

Security in Qlik Sense relies not only on the Qlik Sense software, but also on the security of the environment it is deployed in. The following are must be considered to maximize the security of your Qlik Sense deployment:

- Network security
- Server security
- Process security
- App security

Network security

All communication between Qlik Sense services and web clients use web protocols that use Transport Layer Security (TLS). TLS uses digital certificates to encrypt information exchanged between services, servers, and clients. Encrypted information flows through tunnels requiring two certificates to secure the connection; a server certificate to identify the correct server and a client certificate to allow the client to communicate with the identified server.

Server security

The operating system security system controls access to certificates, storage, memory, and CPU resources. Qlik Sense uses these controls to protect the platform by only allowing authorized users and processes access to required resources.

Process security

Qlik Sense goes through a rigorous testing process during development to mitigate security risks and handle unanticipated events. Additional testing verifies Qlik Sense can stand up against known security threats toward the software.

App security

Attribute based access control provides a comprehensive framework to govern user capabilities within the platform. Row and column level data reduction through section access dynamically manages the data that users view and select in applications.

7.2 Authentication

All authentication in a Qlik Sense deployment is managed by the Qlik Sense Proxy Service (QPS), including clients connecting to the Hub or the Qlik Management Console (QMC). Qlik Sense requires an external identity provider to verify an individual user’s identity. Upon verification, Qlik Sense transfers the user to the Hub or QMC, encrypting traffic using TLS and certificates with various methods, including support for single sign-on (SSO) solutions to minimize the number of times a user must log on to access apps and websites. The QPS supports the use of multiple proxies and each proxy can use multiple authentication methods over a network protected by Transport Layer Security (TLS).

Each Qlik Sense Proxy Service in a Qlik Sense deployment uses virtual proxies to support authentication. Virtual proxies allow one proxy to support multiple authentication schemes, perform session management, and load balancing across multi-node deployments. Virtual proxies may link to one or many Qlik Sense Proxy Service nodes to direct traffic, load balance between engines, or provide specific access to administrative layers of a deployment.

7.3 Authorization

After a user authenticates and gains access to Qlik Sense, authorization through an attribute based access control (ABAC) model enforces application visibility and self-service capabilities within applications. In Qlik Sense, ABAC is defined as an access control method where user requests to perform actions on resources are granted based on assigned attributes of the user, assigned attributes of the resource, environment conditions, and a set of security rules that are specified in terms of those attributes and conditions. Attributes from Active Directory, LDAP, and databases are loaded into Qlik Sense. In addition, attributes may be defined and managed directly within Qlik Sense as well.

Qlik Sense supports authorization in the following ways:

- Security rules
- Section access
- Dynamic data reduction

7.4 Auditing

Governance is critical in enterprise business intelligence. Qlik Sense delivers auditing, monitoring and logging using the QMC, applications, and log files to inform administrators and mitigate risks in deployments.

Qlik Sense supports auditing in the following ways:

- The repository database stores information about when the database was last changed and who made the change.
- The logging framework provides audit and security logs.
- The logs are centrally stored.
- The log format is resistant to injection from the Qlik Sense clients.
- The license logs are signed with a signature to protect them from tampering.

7.5 Confidentiality

Qlik Sense provides confidentiality by encrypting network connections with TLS, leveraging the operating system file system and server access controls to protect content on Qlik Sense nodes, protecting memory using operating system controls, securing application access at the resource level, encrypting sensitive information (e.g. passwords and data connection strings), and protecting app data using data reduction.

Qlik Sense supports confidentiality in the following ways:

- The network uses Transport Layer Security (TLS) for encryption and certificates for authentication.
- The information stored in the file share and the repository database, including Qlik Sense content, is protected by the operating system using server access control and file system controls.
- The process memory and loaded data for Qlik Sense are protected by the physical server and the operating system controls.
- The apps are secured using access control on the resource level.

- Sensitive information (for example, passwords and connection strings) that is used to access external data sources is stored with encryption.
- The app data is protected using data reduction.

7.6 Integrity

Qlik Sense provides integrity through operating system controls like the file system to protect data at rest, encrypt sensitive information, and prevent data write back to the source system.

Qlik Sense supports integrity in the following ways:

- Stored data is protected using the operating system controls (for example, the file system).
- Sensitive information (for example, passwords and connection strings) that is used to access external data sources is stored with encryption.
- Qlik Sense does not support write back to the source system (that is, the Qlik Sense clients cannot edit the data sources).

7.7 Availability

Qlik Sense supports availability in the following ways:

- The nodes in a multi-node site are resilient by design. Each node connects to a central node to access the data it needs to fulfill its role.
- The Qlik Sense protocols are designed to be fault tolerant.

7.8 Certificates

A certificate is a data file that contains keys that are used to encrypt communication between a client and a server in a domain. Certificates also confirm that the domain is known by the organization that issued the certificate. A certificate includes information about the keys, information about the identity of the owner, and the digital signature of an organization that has verified that the content of the certificate is correct. The pair of keys (public and private keys) are used to encrypt communication.

Qlik products use certificates when they communicate with each other. They also use certificates within products, for communication between components that are installed on different computers. These are standard TLS certificates.

The organization that issues the certificate, the certificate authority, is said to “sign” the certificate. You can arrange to get certificates from a certificate authority, to show your domain is known. You can also issue and sign your own (“self-signed certificates”).

Some common errors

Because it generally important for security to know whether a site is known, browsers will display error messages related to certificates and might block communication.

Some common errors are related to the certificate authority. For example, if there is no certificate authority or if the certificate has expired, the default level of security in most browsers will stop communication with a message about “unsigned certificates”, “expired certificates”, or similar terms. If your security administrators know that the certificate is still good, you can create an exception so the error is ignored for that certificate.

Other common errors are related to how the domain is named. For example, companyname.com is a different domain from www.companyname.com, and localhost is a different domain from a server name. A fully qualified domain name is an unambiguous name for a domain. For example, a server at companyname.com might be named mktg-SGK, and can be referred to that way, but the fully qualified domain name is mktg-SGK.companyname.com. (This is called whitelisting.)

Encryption and keys

The kind of encryption used in certificates in Qlik products requires a pair of keys (asymmetric encryption). One key, the public key, is shared. The other key, the private key, is used only by the owner.

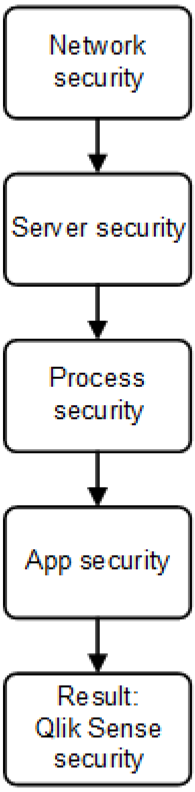
PEM is an ASCII text format for public certificates. It is portable across platforms.

You can get certificates and key pairs from certificate authorities or you can generate them. To get a certificate signed, you will need to also generate a signing request.

7.9 Protecting the platform

The security in Qlik Sense does not depend only on the Qlik Sense software. It also relies on the security of the environment that Qlik Sense operates in. This means that the security of, for example, the operating system and the cryptographic protocols (such as TLS/SSL) has to be set up and configured to provide the security needed for Qlik Sense.

The figure below shows the components that have to be considered in order to maximize the security.



Network security

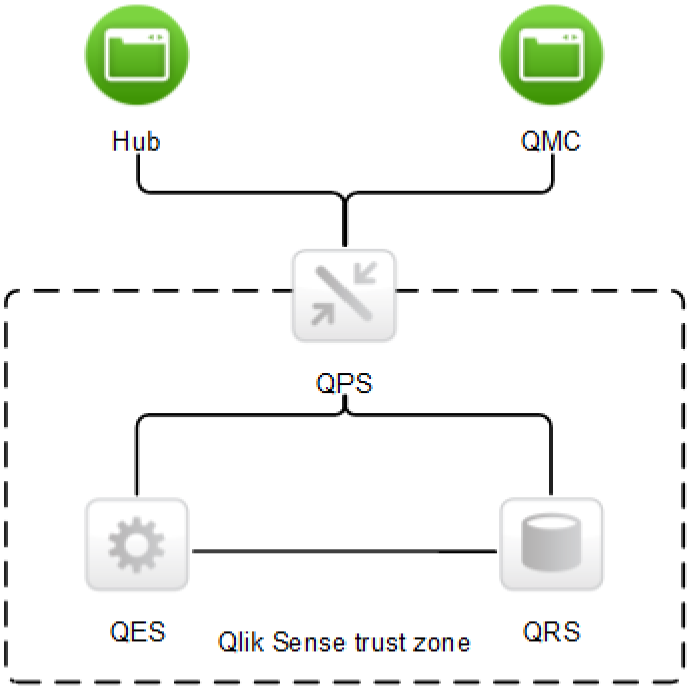
For all Qlik Sense components to communicate with each other in a secure way, they need to build trust.

In Qlik Sense, all communication between the Qlik Sense services and clients is based on web protocols. The web protocols use Transport Layer Security (TLS) for encryption and exchange of information and keys and certificates for authentication of the communicating parties.

TLS provides a way to build encrypted tunnels between identified servers or services. The parties that communicate are identified using certificates. Each tunnel needs two certificates; one to prove to the client that it is communicating with the right server and one to prove to the server that the client is allowed to communicate with the server.

So, how to make sure that the certificates are from the same Qlik Sense trust zone? All certificates that belong to a trust zone are signed with the same signature. If the signature exists in the certificate, it is accepted as proof that the certificate belongs to the trust zone.

When the protected tunnels and the correct certificates are in place, the Qlik Sense services have a trust zone to operate within. Within the trust zone, only services that belong to the specific Qlik Sense site can communicate with each other.



The Qlik Sense clients are considered to be outside of the Qlik Sense trust zone because they often run on less trusted end-user devices. The Qlik Sense Proxy Service (QPS) can bridge the two zones and allow communication between the clients and the Qlik Sense services, if the user is authenticated to the system.

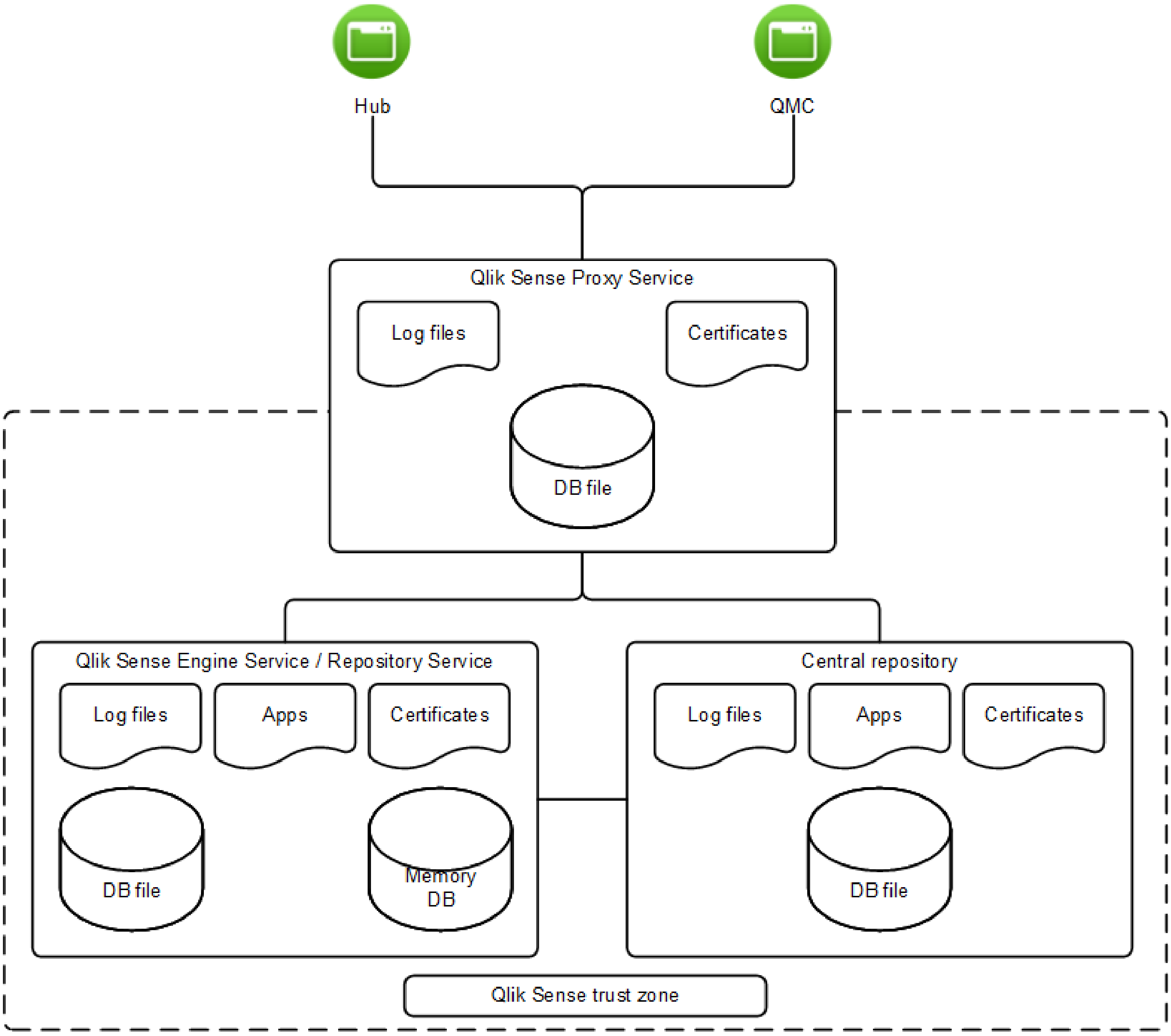
TLS-protected tunnels can be used to secure the communication between the Qlik Sense clients and the QPS. As the clients are outside of the Qlik Sense trust zone, the communication between the clients and the QPS uses a certificate with a different signature than the one used within the trust zone.

See also:

🗋 *Certificate trust (page 218)*

Server security

Qlik Sense uses the server operating system to gain access to resources. The operating system provides a security system that controls the use of the server resources (for example, storage, memory, and CPU). Qlik Sense uses the security system controls to protect its resources (for example, files, memory, processes, and certificates) on the server.



Through the use of access control, the security system grants access to Qlik Sense files (for example, log files, database files, certificates, and apps) only to certain users on the server.

The security system also protects the server memory, so that only authorized processes are allowed to write to the Qlik Sense part of the memory.

In addition, the security system is responsible for assigning users to processes. This is used to restrict who is allowed to interact with the Qlik Sense processes on the server. The processes are also restricted in terms of which parts of the operating system they are allowed to access.

So, by using the controls in the security system, a secure and protected environment can be configured for the Qlik Sense processes and files.

Process security

Each process executes in an environment that poses different threats to the process. In this layer of the security model, the focus is on ensuring that the software is robust and thoroughly analyzed from a security perspective.

Rugged software

For software to be considered as rugged, it must cope with all potential threats to the confidentiality, integrity, and availability of the information, and be robust when used in ways not anticipated.

Several mitigating actions have been implemented in the Qlik Sense software in order to make it rugged:

- Authorization of communication using certificates
- Validation of all external data that is sent to the system
- Encoding of content to avoid injection of malicious code
- Use of protected memory
- Encryption of data
- Audit logging
- Use of checksums
- Isolated execution of external components
- Escaping of SQL data

Threat analysis

To ensure that the Qlik Sense software is secure and rugged, threat analysis of the design has been performed as part of the development process. The following threat areas, often abbreviated as STRIDE, have been covered:

- **S**poofing
- **T**ampering
- **R**epudiation
- **I**nformation disclosure
- **D**enial of service
- **E**levation of privilege

In addition to the threat analyses, exploratory security testing has also been performed on the Qlik Sense software.

App security

The major components of the Qlik Sense app security are:

- Access control system: The access control system grants users access to the resources in Qlik Sense.
- Data reduction: The data reduction functionality is based on the concept of section access, which is a way to dynamically change which data a user can view. This makes it possible to build apps that can be used by many users, but with different data sets that are dynamically created based on user information. The reduction of data is performed by the Qlik Sense Engine Service (QES).

Using these components, the resources and data (that is, the content) consumed by the Qlik Sense users can be secured.

7.10 Authentication

All authentication in Qlik Sense is managed by the Qlik Sense Proxy Service (QPS). The QPS authenticates all users regardless of Qlik Sense client type. This means that the QPS also authenticates users of the Qlik Management Console (QMC).

> *In Qlik Sense, authentication and authorization are two distinct, unconnected actions. In addition, the sources of information used for authentication do not have to be the same as for authorization, and the other way around.*

Qlik Sense always asks an external system to verify who the user is and if the user can prove it. The interaction between Qlik Sense and the external identity provider is handled by authentication modules.

For a module to communicate with Qlik Sense, it has to be trusted. Transport Layer Security (TLS) and certificate authentication are used to authorize external components for communication with Qlik Sense.

In Qlik Sense, the authentication of a user consists of three distinct steps:

1. Authentication module: Get the user identity and credentials.
2. Authentication module: Request an external system to verify the user identity using the credentials.
3. Transfer the user to Qlik Sense using the Ticket API, the Session API, headers, or SAML.

The first two steps are always handled by the authentication module. It is up to the authentication module to verify the user in an appropriate way.

The third step can be performed in the following ways:

- Using the Ticket API, which transfers the user and the user's properties using a one-time ticket.
- Using the Session API, whereby an external module can transfer web sessions that identify the user and the user's properties to Qlik Sense.
- Using headers, with which a trusted system can transfer the user using HTTP headers. This is a common solution for integrating with Single Sign-On (SSO) systems.
- Qlik Sense can be configured to allow anonymous users (using, for example, SAML).

See also:

Network security (page 213)

Default authentication module

After a default installation of Qlik Sense, the Qlik Sense Proxy Service (QPS) includes a module that handles authentication of Microsoft Windows users. The module supports the use of Kerberos and NTLM.

If you want to use Kerberos authentication, you need to make sure that browsers that are used to access the Qlik Sense are configured to support Kerberos.

The default authentication module requires that the proxy that handles the authentication is part of the Microsoft Windows domain.

Certificate trust

Qlik Sense uses certificates for authentication. A certificate provides trust between nodes within a site.

Certificate trust requirements

The requirements described in this section must be fulfilled for the certificate trust to function properly.

When using Transport Layer Security (TLS) in Microsoft Windows environments, the private key must be stored together with the certificate in the Windows certificate store. In addition, the account that is used to run the Qlik Sense services must have permission to access the certificate private key.

If you want to use TLS 1.2 authentication, you need to enable TLS 1.2 support in the Windows registry of the server machine. You should consider the impact of enabling TLS 1.2, as this is a global system setting.

Communication ports

To set up certificate trust, the Qlik Sense Repository Services (QRSs) require that the ports listed in the following table can be opened and used for communication. If any communication passes through a network firewall, the ports in the firewall must be opened and configured for the services.

Port no.	Description
4570	Certificate password verification port, only used within multi-node sites by Qlik Sense Repository Services (QRSs) on rim nodes to receive the password that unlocks a distributed certificate. The port can only be accessed from localhost and it is closed immediately after the certificate has been unlocked. The communication is always unencrypted. This port uses HTTP for communication.
4444	Security distribution port, only used by Qlik Sense Repository Services (QRSs) on rim nodes to receive a certificate from the master QRS on the central node. The communication is always unencrypted, but the transferred certificate package is password-protected. This port uses HTTP for communication.

Ports (page 34)

Unlocking distributed certificates

When adding a new rim node to a site, the distributed certificate needs to be unlocked.

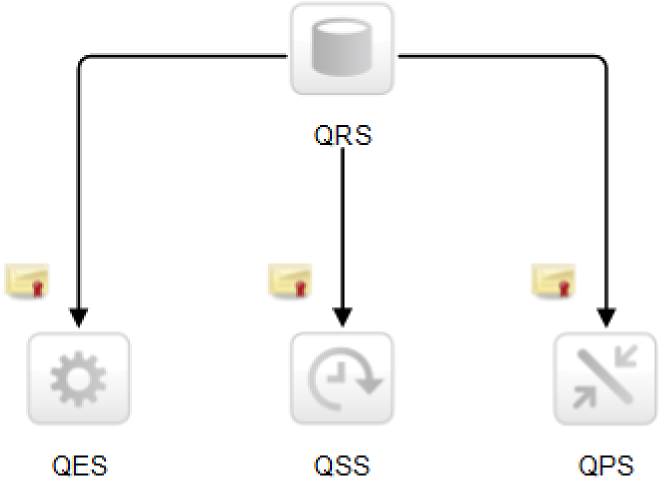
Manage Qlik Sense sites

.

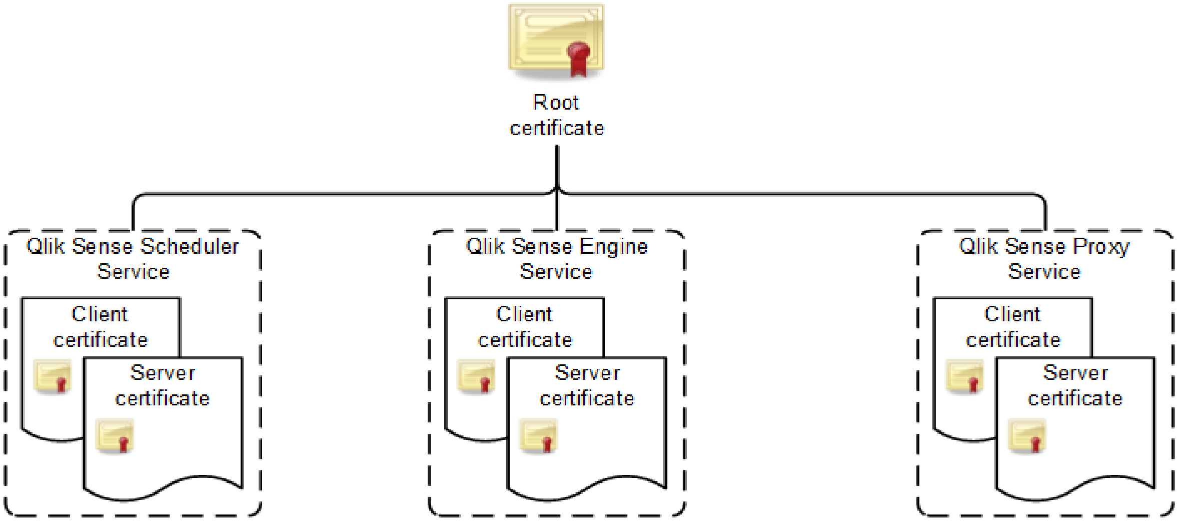
Certificate architecture

Certificates are used within a Qlik Sense site to authenticate communication between services that reside on different nodes. In addition, certificates can be used to build a trust domain between services that are located in different domains or areas (for example, internal networks, extranets, and Internet) without having to share a Microsoft Active Directory (AD) or other user directories.

The architecture is based on the master Qlik Sense Repository Service (QRS) on the central node acting as the certificate manager or Certificate Authority (CA). The master QRS creates and distributes certificates to all nodes within a site. The master QRS is therefore an important part of the security solution and has to be managed from a secure location to keep the certificate solution secure.



The root certificate for the installation is stored on the central node in the site, where the master QRS runs. All nodes with Qlik Sense services that are to be used within the site receive certificates signed with the root certificate when added to the master QRS. The master QRS (that is, the CA) issues digital certificates that contain keys and the identity of the owner. The private key is not made publicly available – it is kept secret by the nodes. The certificate enables the services in a Qlik Sense deployment to validate the authenticity of the other services. This means that the master QRS is responsible for making sure that a service that is deployed on a node is a service within the site.



After the nodes have received certificates, the communication between the Qlik Sense services is encrypted using Transport Layer Security (TLS) encryption.

Confirming certificates using Microsoft Management Console

Certificates can be visually confirmed in the Microsoft Management Console (MMC) with the certificate snap-in added.

If the certificates have been properly deployed, they are available in the locations listed in the table.

Certificate	Location
QlikClient	**Certificates - Current User>Personal>Certificates**
<full computer name>-CA	**Certificates - Current User>Trusted Root Certification Authorities>Certificates**
<full computer name>-CA	**Certificates (Local Computer)>Trusted Root Certification Authorities>Certificates**
<computer name>	**Certificates (Local Computer)>Personal>Certificates**

Certificate handling

This section describes how the certificates are handled when a Qlik Sense service starts.

Client certificate

This section describes how the master Qlik Sense Repository Service (QRS) on the central node in a site handles the client certificate when a Qlik Sense service starts.

The client certificate is located in the following place in the Microsoft Windows certificate store:

Current User>Personal>Certificates

When a Qlik Sense service starts, the QRS searches the certificate store to see if there are any Qlik Sense certificates. Depending on the results of the search, the QRS does the following:

- If no client certificate is found, the QRS logs that no certificate was found.
- If only one client certificate is found, the QRS checks if it is valid. If the certificate is not valid, the QRS logs that an invalid certificate was found.
- If more than one client certificate is found, the QRS deletes all certificates. Duplicates are not allowed. In addition, the QRS logs the number of valid and invalid certificates that were found and deleted.

If certificates are found to be missing or invalid, you must run the QRS in bootstrap mode to recreate the certificates. For more information, see *Services (page 22)*.

Server certificate

This section describes how the master Qlik Sense Repository Service (QRS) on the central node in a site handles the server certificate when a Qlik Sense service starts.

The server certificate is located in the following place in the Microsoft Windows certificate store:

Local Computer>**Personal**>**Certificates**

When a Qlik Sense service starts, the QRS searches the certificate store to see if there are any Qlik Sense certificates. Depending on the results of the search, the QRS does the following:

- If no server certificate is found, the QRS logs that no certificate was found.
- If only one server certificate is found, the QRS checks if it is valid. If the certificate is not valid, the QRS logs that an invalid certificate was found.
- If more than one server certificate is found, the QRS deletes all certificates. Duplicates are not allowed. In addition, the QRS logs the number of valid and invalid certificates that were found and deleted.

If certificates are found to be missing or invalid, you must run the QRS in bootstrap mode to recreate the certificates. For more information, see *Services (page 22)*.

Root certificate

This section describes how the master Qlik Sense Repository Service (QRS) on the central node in a site handles the root certificate when a Qlik Sense service starts.

The root certificate is located in the following places in the Microsoft Windows certificate store:

Current User>**Trusted Root Certification Authorities**>**Certificates**

Local Computer>**Trusted Root Certification Authorities**>**Certificates**

When a Qlik Sense service starts, the QRS searches the certificate store to see if there are any Qlik Sense certificates. Depending on the results of the search, the QRS does the following:

- If no root certificate is found, the QRS logs that no certificate was found.
- If only one root certificate is found, the QRS checks if it is valid. If it is not valid, the QRS logs a fatal error that an invalid root certificate was found, which means that the service is shut down, and that the administrator must manually delete any unwanted certificates. In addition, the QRS logs information abput the certificates that are affected by this.

- If more than one root certificate is found, the QRS logs a fatal error that an invalid root certificate was found, which means that the service is shut down and that the administrator manually has to delete any unwanted certificates. In addition, the QRS logs information on the certificates that are affected by this.

If certificates are found to be missing or invalid, you must run the QRS in bootstrap mode to recreate the certificates. For more information, see *Services (page 22)*.

> *In order not to break any certificate trust between machines, the QRS does not remove any root certificates. It is up to the administrator to decide on what to do with invalid root certificates.*

See also:*Services (page 22)*

Invalid certificate

The definition of an invalid certificate is as follows:

- The operating system considers the certificate to be too old or the certificate chain is incorrect or incomplete.
- The Qlik Sense certificate extension (OID "1.3.6.1.5.5.7.13.3") is missing or does not reflect the location of the certificate:
  - Current User/Personal certificate location: Client
  - Local Machine/Personal certificate location: Server
  - Local Machine/Trusted Root certificate location: Root
  - Current User/Trusted Root certificate location: Root
- The server, client, and root certificates on the central node do not have a private key that the operating system allows them to access.
- The server and client certificates are not signed by the root certificate on the machine.

Maximum number of trusted root certificates

When a Qlik Sense service starts, it checks the number of trusted root certificates on the machine where it is running. If there are more that 300 certificates on the machine, warning messages containing the following information are logged:

- There are too many root certificates for the service to trust.
- The Microsoft Windows operating system will truncate the list of certificates during the Transport Layer Security (TLS) handshake.

If the Qlik Sense root certificate (*<host-machine>-CA*) that the Qlik Sense client certificate belongs to is deleted from the list of certificates because of the truncation, the service cannot be authenticated.

To manually view the root certificates on a machine, open the Microsoft Management Console (MMC) and go to **Certificates (Local Computer)**>**Trusted Root Certification Authorities**.

Authentication solutions

Qlik Sense authentication can be managed with any of the following solutions:

- Ticket solution, see *Ticket solution (page 223)*
- Session solution, see *Session solution (page 224)*
- Header solution, see *Header solution (page 225)*
- SAML, see *SAML (page 226)*
- JWT, see *JWT (page 227)*
- Anonymous users, see *Anonymous users (page 228)*
- Configuring single sign-on (SSO) from Microsoft SQL (MSSQL) server, see *Configuring single sign-on (SSO) for Microsoft SQL (MS SQL) Server (page 228)*

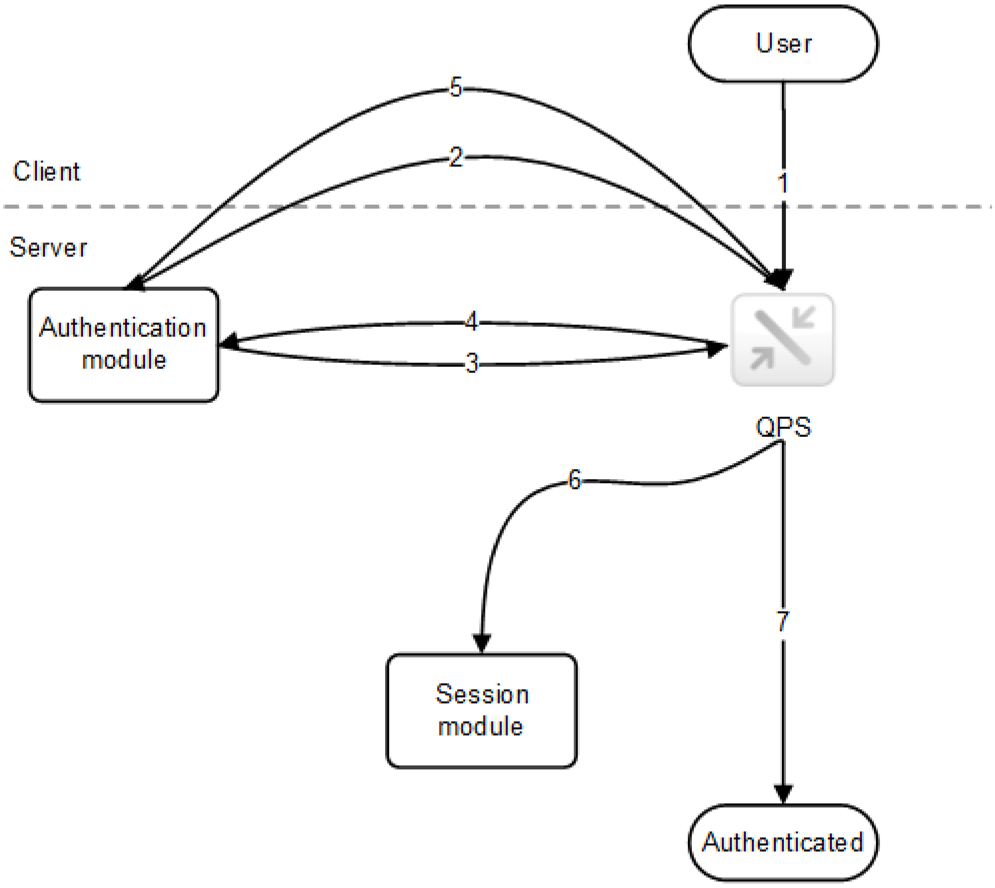
Ticket solution

The ticket solution is similar to a normal ticket. The user receives a ticket after having been verified. The user then brings the ticket to Qlik Sense and, if the ticket is valid, is authenticated. In order to keep the tickets secure, the following restrictions apply:

- A ticket is only valid for a short period of time.
- A ticket is only valid once.
- A ticket is random and therefore hard to guess.

All communication between the authentication module and the Qlik Sense Proxy Service (QPS) uses Transport Layer Security (TLS) and must be authorized using certificates.

The figure below shows a typical flow for authenticating a user with tickets.



1. The user accesses Qlik Sense.
2. Qlik Sense redirects the user to the authentication module. The authentication module verifies the user identity and credentials with an identity provider.
3. Once the credentials have been verified, a ticket is requested from the QPS. Additional properties may be supplied in the request.
4. The authentication module receives a ticket.
5. The user is redirected back to the QPS with the ticket. The QPS checks that the ticket is valid and has not timed out.
6. A proxy session is created for the user.
7. The user is now authenticated.

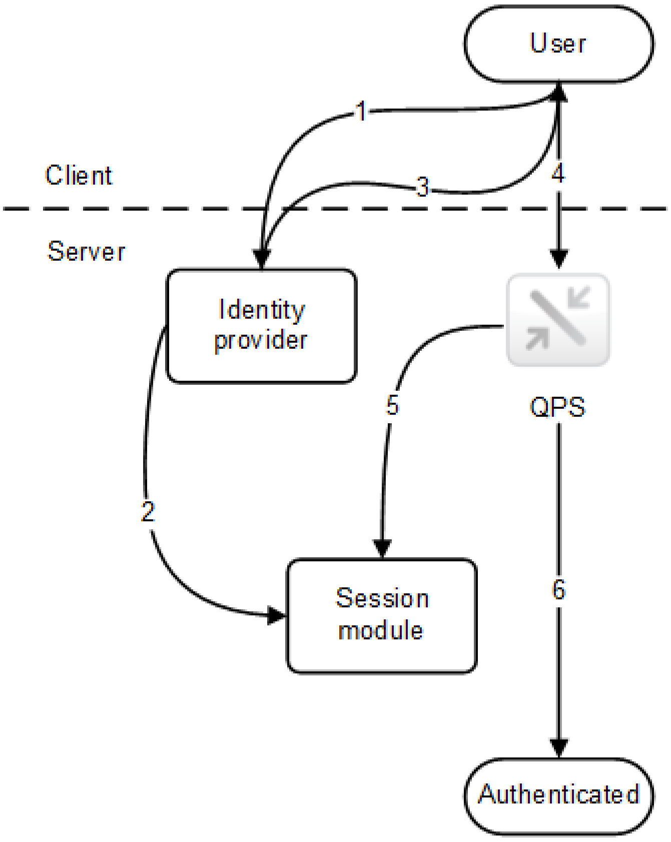
For information about the Authentication (Ticket) API for the Authentication module, see Ticket.

Session solution

The session solution allows the Qlik Sense Proxy Service (QPS) to use a session from an external system to validate who the user is.

All communication between the authentication module and the QPS uses Transport Layer Security (TLS) and must be authorized using certificates.

The figure below shows a typical flow for authenticating a user using a session from an external system.



1. The user accesses the identity provider, which, for example, can be integrated into a portal. The identity provider gets the user identity and credentials and then verifies them. After that, the identity provider creates a new session.
2. The identity provider registers the session token with the Qlik Sense session module.
3. The identity provider sets the session token as a session cookie.
4. The user accesses the QPS to get content (for example, through an iframe in the portal).
5. The QPS validates the session to the session module.
6. If the session is valid and has not yet timed out, the user is authenticated.

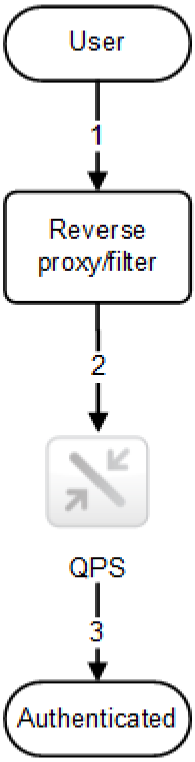
The name of the session cookie used by the authentication module can be configured in the Qlik Management Console (QMC).

For more information about the session module, see Session module API.

Header solution

Header authentication is often used in conjunction with a Single Sign-On (SSO) system that supplies a reverse proxy or filter for authenticating the user.

The figure below shows a typical flow for authenticating a user using header authentication.



1. The user accesses the system and authenticates to the reverse proxy.
2. The reverse proxy injects the username into a defined HTTP header. The header must be included in every request to the Qlik Sense Proxy Service (QPS).
3. The user is authenticated.

For this solution to be secure, the end-user must not be able to communicate directly with the QPS but instead be forced to go through the reverse proxy/filter.

The reverse proxy/filter must be configured to preserve the host name, that is, the host header from the client must not be modified by the reverse proxy/filter.

The name of the HTTP header used for the user can be configured in the Qlik Management Console (QMC).

SAML

Security Assertion Markup Language (SAML) is an XML-based, open-standard data format for exchanging authentication and authorization data between parties (for example, between an identity provider and a service provider). SAML is typically used for web browser single sign-on (SSO).

How SAML works

The SAML specification defines three roles:

- Principal: Typically a user
- IdP: The identity provider
- SP: The service provider

The principal requests a service from the SP, which requests and obtains an identity assertion from the IdP. Based on the assertion, the SP decides whether or not to perform the service requested by the principal.

SAML in Qlik Sense

Qlik Sense supports SAML V2.0 by:

- Implementing an SP that can integrate with external IdPs
- Supporting HTTP Redirect Binding for SAML requests
- Supporting HTTP Redirect Binding and HTTP POST Binding for SAML responses
- Supporting SAML properties for access control of resources and data

Limitations:

- Qlik Sense does not support SAML message signature validation.

JWT

JSON Web Token (JWT) is an open standard for secure transmission of information between two parties as a JavaScript Object Notation (JSON) object. JWT is used for authentication and authorization. Because JWT enables single sign-on (SSO), it minimizes the number of times a user has to log on to cloud applications and websites.

How JWT works

A JWT consists of three parts: a header, a payload, and a signature.

- The header usually consists of two parts: `type (typ)` and `algorithm (alg)`. The algorithm is used to generate the signature.
- The payload is a JSON object that consists of the claims that you want to make. Claims are statements about an entity (usually the user) and additional metadata.
- The signature is used to verify the identity of the JWT sender and to ensure that the message has not been tampered with.

Authentication is performed by verifying the signature. If the signature is valid, access is granted to Qlik Sense.

Limitations

The following limitations exist:

- Encrypted JWTs are not supported.

> *When using HTTPS, all traffic, including JWTs, are encrypted during transport.*

- Only the following signing algorithms are supported:
  - RS256 - RSA signature with SHA256
  - RS384 - RSA signature with SHA384
  - RS512 - RSA signature with SHA512

Anonymous users

If anonymous use of Qlik Sense is allowed, users who are not authenticated are not automatically redirected to an authentication module. Instead, the user first gets anonymous access and is then, if the user chooses to sign in, redirected to the authentication module to supply user identity and credentials.

Configuring single sign-on (SSO) for Microsoft SQL (MS SQL) Server

If your database files access data from MS SQL Server, you can configure the host server to enable SSO. ODBC data source single sign-on permits clients to use one Windows authenticated login to access data in shared files.

To configure SSO for MS SQL Server, a Windows domain administrator must do the following:

- Create service principal names (SPN) in Active Directory
- Configure delegation for the Qlik Sense services administrator account
- Configure the Qlik Sense server for SSO
- Configure the MS SQL Server for SSO

> ⚠ *The Microsoft SQL Server Connector in the Qlik ODBC Connector Package also supports SSO. If you are using the connector in the ODBC Connector Package, use the following configuration instructions: Configuring SSO for the Microsoft SQL Server connector.*

> ⚠ *The same Qlik Sense services administrator account used during the Qlik Sense (central node) installation must be used. If a different account is used, the Qlik Sense services administrator account must own the HTTP service principal. For more information, see User accounts (page 69).*

Creating service principal names (SPN) in Active Directory

A service principal name (SPN) is a unique identifier of a service instance. SPNs are used during authentication to associate a service instance with a service logon account. This allows a client application to request that a service authenticate an account even if the client does not have the account name. A SPN always includes the name of the host computer on which the service instance is running, so a service instance might register a SPN for each name or alias of its host.

Before the authentication service can use a SPN to authenticate a service, the SPN must be registered on the account object that the service instance uses to log on. A given SPN can be registered on only one account. For Win32 services, a service installer specifies the logon account when an instance of the service is installed. The installer then composes the SPNs and writes them as a property of the account object in Active Directory Domain Services. If the account of a service instance changes, the SPNs must be re-registered under the new account.

When a client connects to a service, it locates an instance of the service, composes an SPN for that instance, connects to the service, and presents the SPN for the service to authenticate.

To set up SSO for MS SQL server, you must create SPNs for the Qlik Sense services administrator account.

Do the following:

1. Log on as a domain administrator.
2. Open an elevated command prompt.
3. Enter the following to create a SPN for the Qlik Sense services administrator:
   setspn -A HTTP/<Qlik_Sense_server>:<port> <domain>\<Qlik_Sense_services_administrator>

   The <Qlik_Sense_server> must be entered as the fully qualified domain name of the server.

   The <Qlik_Sense_server> is the central node where the Qlik Sense is running.
4. Enter the following to create a SPN for the MS SQL Server services administrator:
   setspn -A MSSQLSvc/<server_name>:<port> <domain>\<services_administrator>

   The <server_name> must be entered as the fully qualified domain name of the server.
5. Enter the following commands to verify the result of your SPN setup:
   a. setspn -L <domain>\<Qlik_Sense_services_administrator> to verify the Qlik Sense services administrator.
   b. setspn -L <domain>\<MS_Sql _server_services_administrator> to verify the MS SQL Server services administrator.

Configuring delegation for the Qlik Sense services administrator account

Delegation allows a front-end service to forward client requests to a back-end service so that the back-end service can also impersonate the client. Impersonation is used to check whether a client is authorized to perform a particular action, while delegation is a way of flowing impersonation capabilities, along with the client’s identity, to a back-end service.

To configure SSO for MS SQL Server, you must set up delegation rights to the MS SQL Server service for the Qlik Sense services administrator.

A Windows domain administrator can change the delegation tab on the Qlik Sense services administrator account properties page.

Do the following:

1. Log on as a Windows domain administrator.
2. Right click on your Qlik Sense services administrator account and click **Properties**.
3. Go to the **Delegation** tab, and select **Trust this user for delegation to specified services only**, then select **Use any authentication protocol**.
4. Click **Add...**.
5. On the **Add Services** window, click **Users or Computers...**.

6. On the **Select Users or Computers** window, enter the domain and user name of the Microsoft SQL Server services administrator and click **OK**.
7. On the **Add Services** window, select the MS SQL Server service and click **OK**.

You can verify your delegation configuration on the **Delegation** tab. The MS SQL Server service should now be set as the service to which the Qlik Sense services administrator can present delegation credentials.

Configuring the Qlik Sense server for SSO

To configure the Qlik Sense server for SSO with MS SQL Server, you must:

- Add the Qlik Sense services administrator to the **Administrator** group on the Qlik Sense server if it's not already part of that group.
- Add Qlik Sense services administrator as part of the **Act as part of the operating system** role in the **Local Security Policy**.

Do the following:

1. Log on to the Qlik Sense server as an administrator.
2. Open **Local Security Policy**, and go to **Security Settings** > **Local Policies** > **User Rights Assignment**.
3. Under **Policy**, right click on **Act as part of the operating system** and select **Properties**.
4. On the **Local Security Setting** tab, click **Add User or Group...**.
5. Add the Qlik Sense services administrator account, and click **OK**.

Configuring MS SQL Server

To configure the MS SQL Server for SSO, you must ensure that the MS SQL Server service runs as the MS SQL Server services administrator.

Do the following:

1. Log on to the MS SQL Server as an administrator.
2. Open the **Sql Server Configuration Manager**.
3. Select **SQL Server Services**.
4. Select **SQL Server** in the right pane and verify that the **Log On As** column is populated with your MS SQL Server services administrator account.

You must reboot after making changes to remove the SQL self registration of the SPN under machine account and register the SPN manually on the domain account.

7.11 Authorization

Authorization is the procedure of granting or denying users access to resources.

In Qlik Sense, authentication and authorization are two distinct, unconnected actions. In addition, the sources of information used for authentication do not have to be the same as for authorization, and the other way around.

In Qlik Sense, there are two authorization systems:

- Access control: The access control system grants users access to the resources in Qlik Sense. The access control system is implemented in the Qlik Sense Repository Service (QRS) and independent of the operating system.
- Data reduction: The data reduction functionality is based on the concept of section access, which is a way to dynamically change which data a user can view. This makes it possible to build apps that can be used by many users, but with different data sets that are dynamically created based on user information. The reduction of data is performed by the Qlik Sense Engine Service (QES).

The two authorization systems are unconnected and configured separately.

Access control

This section describes the different types of access control:

- Resource access control: Is the user allowed to access the app? Which functions in the app is the user allowed to use (for example, printing, exporting, and snapshots)?
- Administrator access control: Which access rights are needed for the different roles and responsibilities of the administrators?

Resource access control

The resource access control system in Qlik Sense is based on properties. This means that the access is based on rules that refer to properties connected to resources and users in Qlik Sense.

All authorization to resources is enforced by the Qlik Sense Repository Service (QRS). The QRS only gives other Qlik Sense services access to resources that the current user is allowed to access.

The resource access control system determines the access based on the following parameters:

- User name and user properties: The user name and user properties are supplied by the Qlik Sense Proxy Service (QPS) that authenticated the user.
- Action: The method that the user is trying to perform on a resource (for example, create, read, or print).
- Resource: The entity that the user is trying to perform an action on (for example, app, sheet, or object).
- Environment: The environment is supplied by the QPS and describes, for example, time, location, protection, and the type of Qlik Sense client used.

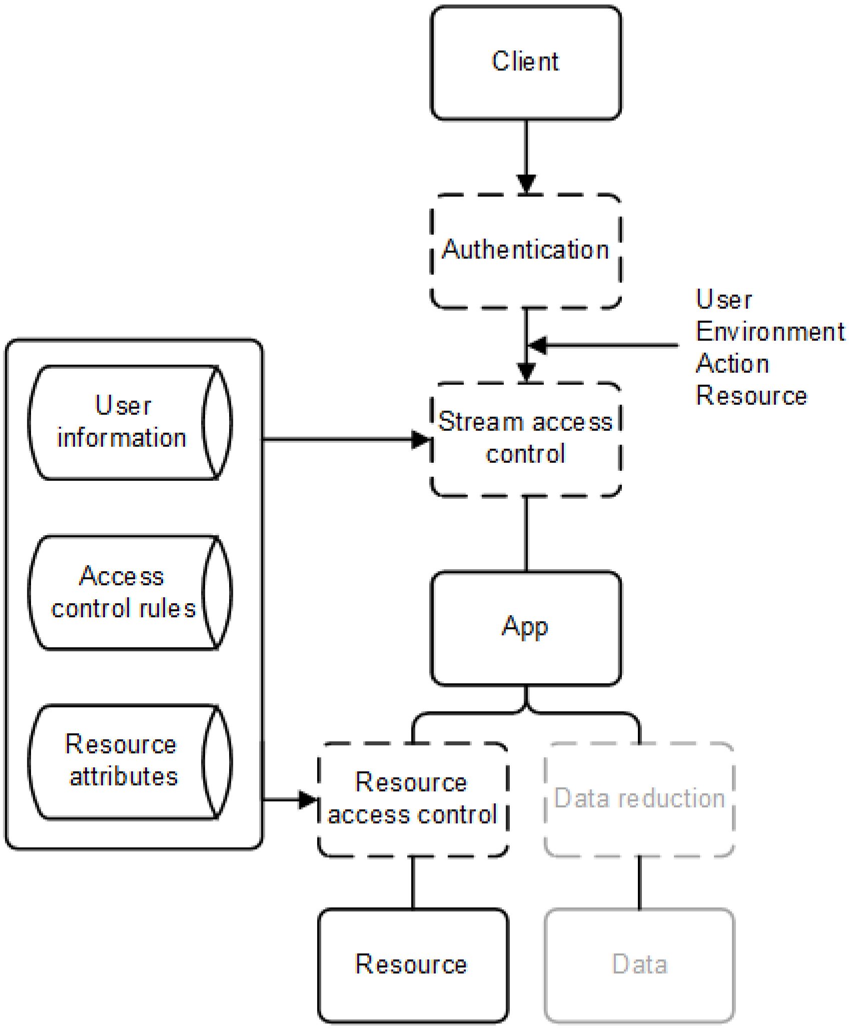
Resource access control rules

The system administrator can set up rules for the resources access control. The rules are divided into three parts:

- Resource filter: The resources that the rule applies to.
- Condition: A logical condition that, if evaluated as true, grants access.

- Action: The action that the user is allowed to perform, if the condition is true.

Properties connected to resources or users may be used in the rules. Examples of properties include the name of user or resource, type of resource, and Active Directory groups for users or custom-defined properties.



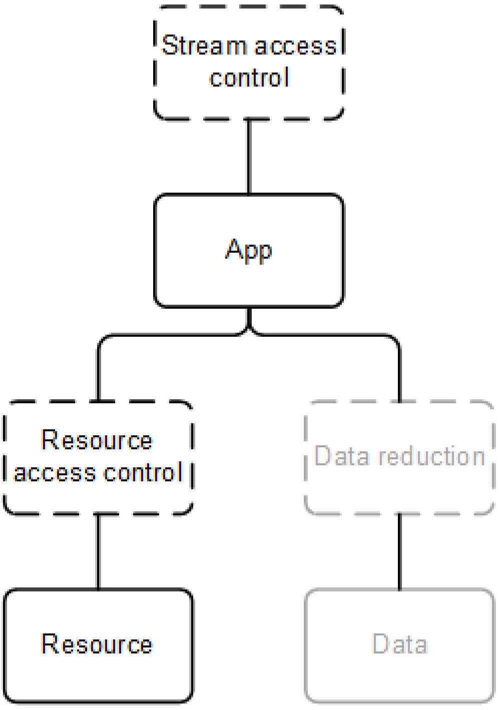
Resource access control streams

To make the management of the Qlik Sense authorization systems efficient, apps can be grouped into streams. From an authorization perspective, a stream is a grouping of apps that a group of users has read (often referred to as “subscription”) or publish access to.

By default, Qlik Sense includes the following streams:

- Everyone: All users have read and publish rights to this stream.
- Monitoring apps: Contains a number of apps for monitoring of Qlik Sense.

Streams are created and managed in the Qlik Management Console (QMC).



Administrator access control

In addition to setting up the access control for the users, it is important to configure the access control for the administrators so that they get access rights in the Qlik Management Console (QMC) that correspond to their roles and responsibilities.

Common administrator roles include the following:

- RootAdmin
- AuditAdmin
- ContentAdmin
- DeploymentAdmin
- SecurityAdmin

For a presentation of the access rights for the respective administrator roles, see the topic *Default administration roles* in the document Manage Qlik Sense sites.

Data reduction

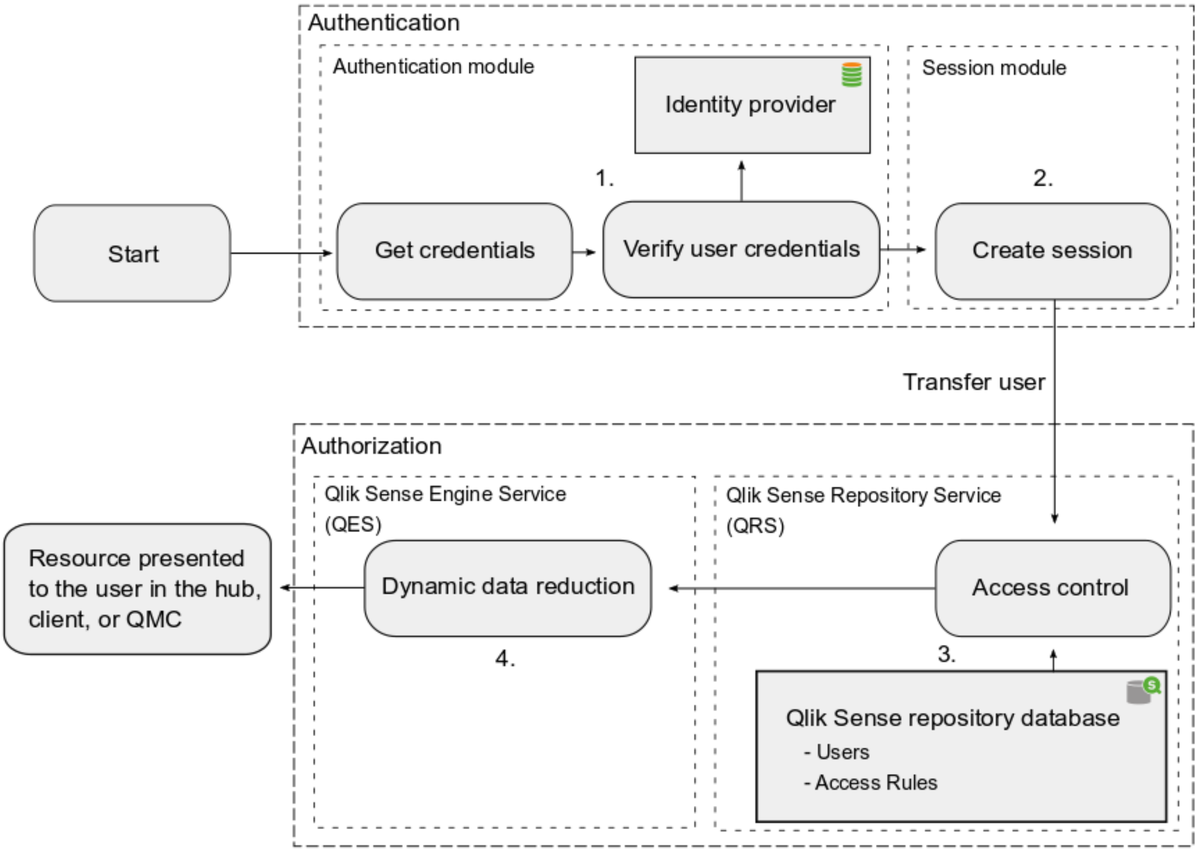
Data reduction is used to determine which data a user is allowed to see: all of it or just parts of it?

The data reduction functionality is based on the concept of section access, which is a way to dynamically change which data a user can view. This makes it possible to build apps that can be used by many users, but with different data sets that are dynamically created based on user information. The reduction of data is performed by the Qlik Sense Engine Service (QES).

The definition of access rights for section access is maintained in the apps and configured through the load script.

7.12 Security example: Opening an app

The figure below shows the flow in the Qlik Sense security system when a user logs in and opens an app.



1. Authentication: The authentication module in the Qlik Sense Proxy Service (QPS) handles the authentication. The credentials provided by the user are verified against information from the identity provider (for example, a directory service such as Microsoft Active Directory).
2. Session creation: When the user credentials have been successfully verified by the authentication module, a session is created for the user by the session module in the QPS.
3. Access control system: When the user tries to open an app, the Qlik Sense Engine Service (QES) requests the Qlik Sense Repository Service (QRS) to check if the user is authorized to perform the action. The QRS then checks the repository database, where, among other things, all users and access rules are stored.

   A user is imported into the repository database from a User Directory (UD) (for example, Microsoft Active Directory) using Qlik Sense User Directory Connectors (UDCs). The import is triggered by the Qlik Sense Scheduler Service (QSS) and the intervals in-between imports can be scheduled.

4. Dynamic data reduction: When the user has been successfully authorized by the QRS, the app is opened. Before the data is displayed to the user, the QES performs a dynamic data reduction, where the data that the user is allowed to see is prepared.

See also:

- *App security (page 216)*

8 Logging

The log messages produced by Qlik Sense provide important information about the general well-being of the deployment.

The logging is based on the log4net component in Apache Logging Services. This means that Qlik Sense uses a standardized logging framework and conforms to standard logging procedures.

8.1 Updated logging framework

An updated logging framework was introduced in Qlik Sense version 2.0. Unless otherwise stated, the documentation describes the updated logging framework.

8.2 Legacy logging framework

The legacy logging framework is still available in Qlik Sense, but the logs are as of Qlik Sense version 2.0 referred to as trace logs. The log files remain the same, in the old logging format, but they are stored in a new location.

See: *Trace logs (page 268)*

8.3 Centralized logging framework

As of the September 2017 release of Qlik Sense, centralized logging gives you the ability to log directly into a PostgreSQL database. With all logs in one place it will be easier for you find the relevant logs. If you install Qlik Sense with centralized logging, which is included in a default installation, log entries are persisted in two locations: the existing log files and the centralized logging database.

8.4 Reading and analyzing log files in Qlik Sense

The log files can be read and analyzed using Qlik Sense, which includes the following pre-defined, log-related data connections after installation:

- ServerLogFolder: Links to the active log files.
- ArchivedLogsFolder: Links to the archived log files.

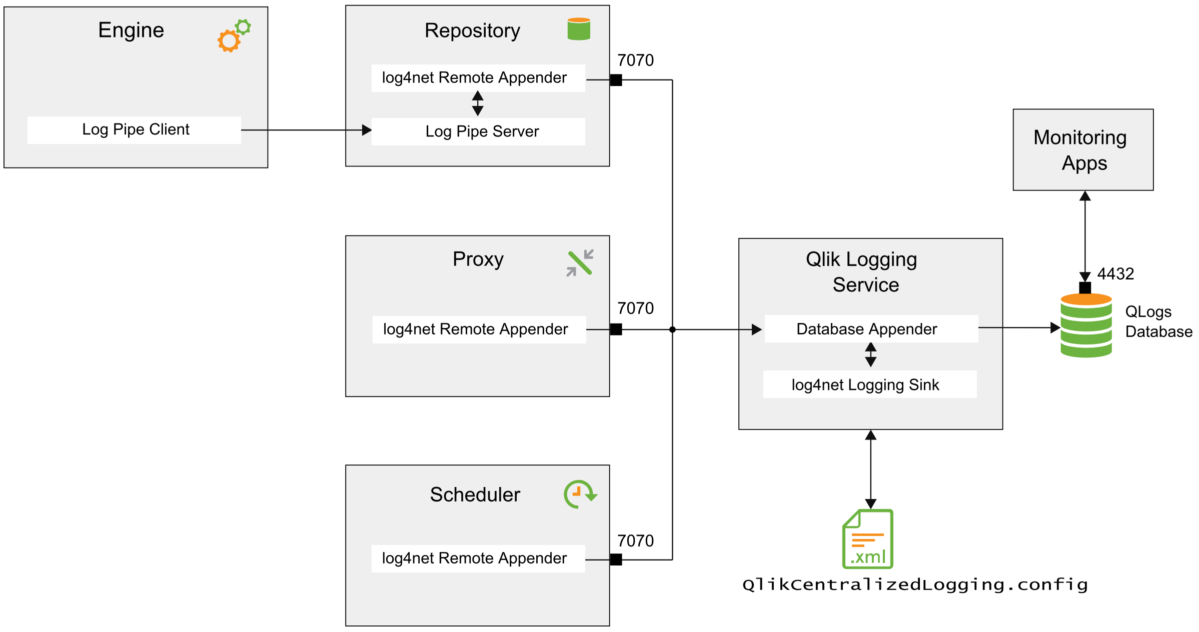
The data connections can be edited in the Qlik Management Console (QMC).

In addition, users with root, security, content, or deployment administrator rights can use the Qlik Sense log data in apps by selecting one of the data connections listed above in the data load editor.

8.5 Centralized logging

With the introduction of shared persistence, all nodes have direct access to a common database and file system. The Qlik Sense services (proxy, scheduler, repository, and engine) transfer log messages to the Qlik Logging Service. The Qlik Logging Service centralizes the logging by collecting all the messages and inserting them into the PostgreSQL database.

Centralized logging uses the log4net library to write log information to the database. The Qlik Sense Repository, Proxy, and Scheduler services use the RemotingAppender from log4net to transfer log messages to a remote logging sink that is read by the Qlik Logging Service. The Engine loggers use a pipe connection to the Repository service, which, in turn, persists the data to the Qlik Logging Service. All services use TCP localhost port 7070 to establish communication. The Qlik Logging Service collects all messages and inserts them into the PostgreSQL database named QLogs using a custom AdoNetAppender. The configuration for the Qlik Logging Service is stored in an XML file named *QlikCentralizedLogging.config*. As part of the centralized logging database feature, the Monitoring Apps include an ODBC (PostgreSQL) data connection. By default, this data connection points to the QLogs database on localhost port 4432. This data connection is not used when only file logging is enabled.



Centralized logging uses the log4net library to persist log information collected from Qlik Sense services to the database.

Built-in log4net appenders (page 282)

8.6 Qlik Logging Service

The Qlik Logging Service is used to centralize logging, which makes it easier for you to find the log that you are looking for.

When centralized logging (the Qlik Logging Service) is on, file logging is also on, by default. Log entries from the Qlik Sense services (repository, proxy, scheduler, and engine) are persisted in two locations – the existing log files and the centralized logging database. The legacy log files do not have any built-in file management to clear up hard disk space.

Although the Qlik Logging Service will be installed as a Windows service, you can use it directly as a command line tool to configure or change the database settings. The available commands are included in this topic.

Currently, there is no support for streaming messages from the Qlik Logging Service to third-party tools or customers. The only supported way is through the Monitoring apps.

Command line options

These are the available commands for the Qlik Logging Service.

Usage: `Qlik.Logging.Service.exe <action> [<args>]`

The following commands can be used to set up, update, or validate the logging database:

`setup` - creates the logging database and sets up roles and access permissions.

`update` - updates the connection string parameters for the logging database with user-provided values.

`validate` - validates connection string parameters from the configuration file and database connectivity.

`archive` - moves database entries to the archive table.

`purge` - removes (permanently) database entries from the archive table.

`version` - displays version information for the service.

`help` - displays help message.

Setting up the logging database

The setup command is typically used by the installer at the time of the Qlik Sense installation. The setup command creates the logging database and sets up access roles and required permissions.

Usage: `Qlik.Logging.Service.exe setup [<options>]`

Options

`--hostname` or `-h` (Required)

Name of the machine where the logging database is hosted.

`--port` or `-p` (Required)

Port number used to access the logging database.

`--postgres_user` or `-u` (Required)

PostgreSQL user name credentials required to create the logging database.

`--postgres_pswd` or `-x` (Required)

PostgreSQL password required to create the logging database.

`--reader_pswd` or `-r` (Required)

Password for qlogs_reader user role, used for reading logging database entries.

`--writer_pswd` or `-w` (Required)

Password for qlogs_writer user role, used for writing to the logging database.

`--force` or `-f` (Optional)

Drop the existing database and users, if present.

Updating the connection string parameters

The update command is used to modify the connection, the configuration settings of the logging database, or both.

Usage: `Qlik.Logging.Service.exe update [<options>]`

Options

`--hostname` or `-h` (Optional)

Name of the machine where the logging database is hosted.

`--port` or `-p` (Optional)

Port number used to access the logging database.

`--reader_pswd` or `-r` (Optional)

Password for qlogs_reader user role, used for reading logging database entries.

`--writer_pswd` or `-w` (Optional)

Password for qlogs_writer user role, used for writing to the logging database.

`--archive_age` or `-a` (Optional)

Sets value for archive age. The value is specified in days. Specify `--hours` to interpret archive age as hours, for example, `--archive_age X --hours`.

`--purge_age` or `-q` (Optional)

Sets value for purge age. The value is specified in days. Specify `--hours` to interpret purge age as hours, for example, `--purge_age X --hours`.

`--file_logging` or `-f` (Optional)

Switches on or off file logging. Valid values are 'on' or 'off'.

`--database_logging` or `-d` (Optional)

Switches on or off database logging. Valid values are 'on' or 'off'.

`--maximum_db_size_in_gb` or `-s` (Optional)

Sets value for maximum database size. The value is specified in GB. A value less than two (2) disables the functionality that limits the database size.

Validating the logging database connection

Usage: `Qlik.Logging.Service.exe validate [<options>]`

Options

None. The connection string is read from the logging configuration and is used to validate the state of the database and the connection.

Archiving the log entries

Archives all the log entries that are older than the specified cutoff time (in days).

Usage: `Qlik.Logging.Service.exe archive [<options>]`

Options

`--cutoff` or `-c` (Required)

`--hours` (Optional)

Cutoff time in days. Specify `--hours` option to interpret cutoff time as hours, for example, `archive --cutoff X --hours`. Specify zero (0) to archive all the entries.

Purging log entries

Purges all the archived log entries that are older than the specified cutoff time (in days).

Purge is an irreversible operation. Log entries are removed from the database and cannot be restored. This operation does not affect log messages in the log files.

Usage: `Qlik.Logging.Service.exe purge [<options>]`

Options

`--cutoff` or `-c` (Required)

`--hours` (Optional)

Cutoff time in days. Specify `--hours` option to interpret cutoff time as hours, for example, `purge --cutoff X --hours`. Specify zero (0) to purge all the entries.

Version

Usage: `Qlik.Logging.Service.exe version [<options>]`

Options

None. Displays Qlik Logging Service and logging database versions.

Help

Usage: `Qlik.Logging.Service.exe help [<options>]`

Options

None. Displays usage.

8.7 Qlik Logging Service – configuration and integration

This topic describes the configuration and integration features that are available for the Qlik Logging Service. The features should be considered "Advanced Features" to be used in the context of troubleshooting issues with the assistance of Qlik technical support.

Dynamic configuration

Most of the functionality provided by the Qlik Logging Service is controlled by the configuration file: *QlikCentralizedLogging.config* typically located here: *C:\ProgramData\Qlik\Sense\Log*. This file contains all the settings recognized by the Qlik Logging Service in xml format and is typically modified directly by the Qlik Logging Service when executing the `update` or `setup` commands during command line invocation.

The "Dynamic Configuration" feature introduces a new setting: `CentralizedLoggingConfigurationNotificationEnabled`, its default value is "`False`". This setting is always dynamic, meaning that changing its value by directly modifying the configuration file and saving it will cause the Qlik Logging Service to read and use its new value. When this setting is set to "`False`", the default value, changes to other settings in the configuration file are not detected by the Qlik Logging Service and will require a service restart to take effect. When the setting is set to "`True`", changes to any of the settings below will be detected and applied immediately by the Qlik Logging Service after saving the configuration file.

Key	**Description**	**Range**
CentralizedLoggingLogMaxFileSizeMb	The logging service logs to a file, this setting controls the maximum size in MB of the log file before it is rolled over.	1...
CentralizedLoggingLogEnabled	Turns on/off logging to file by the logging service.	True, False
CentralizedLoggingEnabled	Enables or disables the logging service functionality. When `True`, the service accepts incoming log entries, when `False`, the service continues to run, but it does not offer any logging functionality.	True, False
CentralizedLoggingServerBufferSize	Sets the size in log entries of the internal buffer. Writes to the database occur when the internal buffer is full. For example if set to 10, writes to the database will occur when 10 log entries are received.	1...
CentralizedLoggingServerTimeoutSeconds	Sets the timeout in seconds required to trigger a database write. On an idle system log entries may not accumulate often. This	1...

	threshold will force a database write even if the internal buffer is not full.	
CentralizedLoggingLogLevel	Controls the verbosity level of the log.	Off, Fatal, Error, Warn, Info, Debug, All
MaximumDatabaseSizeInGB	Sets the maximum size for the database in GB, when the size of the database exceeds this number, the logging service will trim/remove entries from the database until the size of the database goes below the maximum set. *A value less than 2 disables the functionality.*	0...
CentralizedLoggingConfigurationNotificationEnabled	Enables or disables dynamic configuration settings. When set to `True`, changes to a dynamic configuration setting take effect immediately without requiring a service re-start. When `False`, a re-start is required for the new setting value to take effect. Notice that this setting is always dynamic.	True, False
CentralizedLoggingDBSizeCheckPeriod	Sets how often to check the size of the database. The quantity is expressed in milliseconds. *A value of zero (0) disables the functionality, 600000 (ten minutes) is the minimum value accepted. Values less than that are automatically set to that minimum and 2147483646 (~596 hours) is the maximum.*	0, 600000..2147483646
CentralizedLoggingDBArchiveCheckPeriod	Sets how often to check for the need to archive log entries, the quantity is expressed in milliseconds.	0, 600000..2147483646

	A value of zero (0) disables the functionality, 600000 (ten minutes) is the minimum value accepted. Values less than that are automatically set to that minimum and 2147483646 (~596 hours) is the maximum.	
CentralizedLoggingDBPurgeCheckPeriod	Sets how often to check for the need to purge archived entries, the quantity is expressed in milliseconds.	0, 600000..2147483646
	A value of zero (0) disables the functionality, 600000 (ten minutes) is the minimum value accepted. Values less than that are automatically set to that minimum and 2147483646 (~596 hours) is the maximum.	
CentralizedLoggingDBStatsCheckPeriod	Sets how often to retrieve database statistics. The quantity is expressed in milliseconds.	0, 300000..2147483646
	A value of zero(0) disables the functionality, 300000 (five minutes) is the minimum value accepted. Values less than that are automatically set to that minimum and 2147483646 (~596 hours) is the maximum.	
CentralizedLoggingPlatformPerformanceCheckPeriod	Sets how often to retrieve platform performance metrics, the quantity is expressed in milliseconds.	0, 1000..2147483646

	A value of zero (0) disables the functionality, 1000 (one second) is the minimum value accepted. Values less than that are automatically set to that minimum and 2147483646 (~596 hours) is the maximum.	
CentralizedLoggingProcessPerformanceCheckPeriod	Sets how often to retrieve process performance metrics, the quantity is expressed in milliseconds. *A value of zero (0) disables the functionality, 1000 (one second) is the minimum value accepted. Values less than that are automatically set to that minimum and 2147483646 (~596 hours) is the maximum.*	0, 1000..2147483646
CentralizedLoggingPlatformPerformanceCounters	Set of platform performance counters to collect every `CentralizedLoggingPlatformPerformanceCheckPeriod` period. The format expected is: Database payload key; Counter Category; Counter Name [;] [Counter Instance] | Database payload key...	
CentralizedLoggingProcessPerformance	Set of processes to collect performance counter for every `CentralizedLoggingProcessPerformanceCheckPeriod` period. The format expected is: `Process Name | Process Name | ...` Process Name is the simple name for the process for example the default value of this setting is: `engine|proxy|repository|scheduler`	
CentralizedLoggingPlatformEvents	Set of platform events to monitor The format expected is:	

EventLogX, [EventSourceX]:EventID0, EventID1, ..., EventIDn|

EventLogY, [EventSourceY]:EventID0, ..., EventIDn

Qlogs statistics

The Qlik Logging Service periodically collects statistics about the Qlogs database, which is the database used for all logging entries. How often these statistics are collected is controlled by the `CentralizedLoggingDBStatsCheckPeriod` setting. Setting its value to zero(0) disables collection and its default value is ten (10) minutes expressed in milliseconds (10 * 60 * 1000 = 600000). The information collected contains record counts and size information and can be viewed with the following query:

```
SELECT * FROM public.view_db_stats;
```

Windows Event Log integration

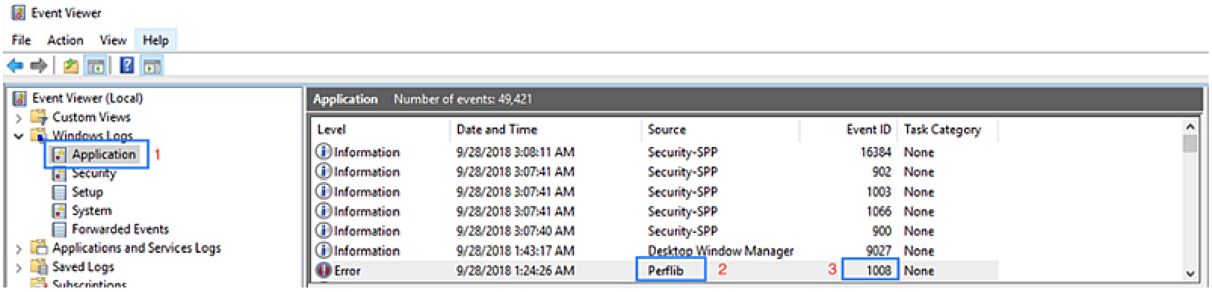
Integration with the Windows Event Log is a new feature that allows the Qlik Logging Service to collect information about configured Windows events. The set of events monitored by the Qlik Logging Service is set via the CentralizedLoggingPlatformEvents setting. Clearing its value disables the feature. Its default value is:

```
<add key="CentralizedLoggingPlatformEvents"
value="System:1074,6013,36888,36874,2013,7031,4202|Application, Engine:300|Application, .NET
Runtime:1026|Application, PostgreSQL:0" />
```

This is an explanation of the events configured by default.

Event log	Event source	Event ID	Description
System		1074	User initiated system re-start
System		6013	System uptime
System		36888	Schannel (TLS protocol) errors
System		36874	Schannel (TLS protocol) errors
System		2013	Low disk space
System		7031	Service terminated unexpectedly
System		4202	TCP/IP network configuration issue
Application	Engine	300	Engine caught unexpected exception
Application	PostgreSQL	0	PostgreSQL error
Application	.NET Runtime	1026	.NET process terminated

Additional events may be monitor by modifying the value of the `CentralizedLoggingPlatformEvents` setting as follows:



Level	Date and Time	Source	Event ID	Task Category
Information	9/28/2018 3:08:11 AM	Security-SPP	16384	None
Information	9/28/2018 3:07:41 AM	Security-SPP	902	None
Information	9/28/2018 3:07:41 AM	Security-SPP	1003	None
Information	9/28/2018 3:07:41 AM	Security-SPP	1066	None
Information	9/28/2018 3:07:40 AM	Security-SPP	900	None
Information	9/28/2018 1:43:17 AM	Desktop Window Manager	9027	None
Error	9/28/2018 1:24:26 AM	Perflib	1008	None

As shown above, the Event Log is one (1), the Event Source is two (2) and the Event ID is three (3). To capture the event highlighted above, the value "`Application, Perflib: 1008`" would need to be appended to the `CentralizedLoggingPlatformEvents` setting as shown below:

```
<add key="CentralizedLoggingPlatformEvents"
value="System:1074,6013,36888,36874,2013,7031,4202|Application, Engine:300|Application, .NET
Runtime:1026|Application, PostgreSQL:0|Application, Perflib: 1008" />
```

Notice that the Event Source is optional, but useful when multiple event sources may use the same event ID. For the example above, omitting the event source "`Perflib`" would have resulted in the monitoring of event ID 1008 for ALL applications. Multiple event IDs may be monitored for the same Event Log and Event Source and they are separated by commas "," and multiple Event Log and Event Sources may be monitored and they are separated by the bar "|" character.

Any Windows events captured by the Qlik Logging Service may be viewed by executing the following query against the Qlogs database:

```
SELECT * FROM public.view_platform_events;
```

Windows Performance Monitor integration

The Qlik Logging Service integrates with the Windows Performance Monitor, it periodically queries the system for configured metrics. The integration is divided into two sections, one that monitors configured platform metrics and one that monitors configured processes.

Platform metrics

The period used to monitor platform metrics is controlled by the `CentralizedLoggingPlatformPerformanceCheckPeriod` setting. A value of zero (0) disables this functionality and its default value is five (5) minutes expressed in milliseconds (5 * 60 * 1000 = 300000).

The metrics or performance counters collected are configured in the setting `CentralizedLoggingPlatformPerformanceCounters` and its default value is:

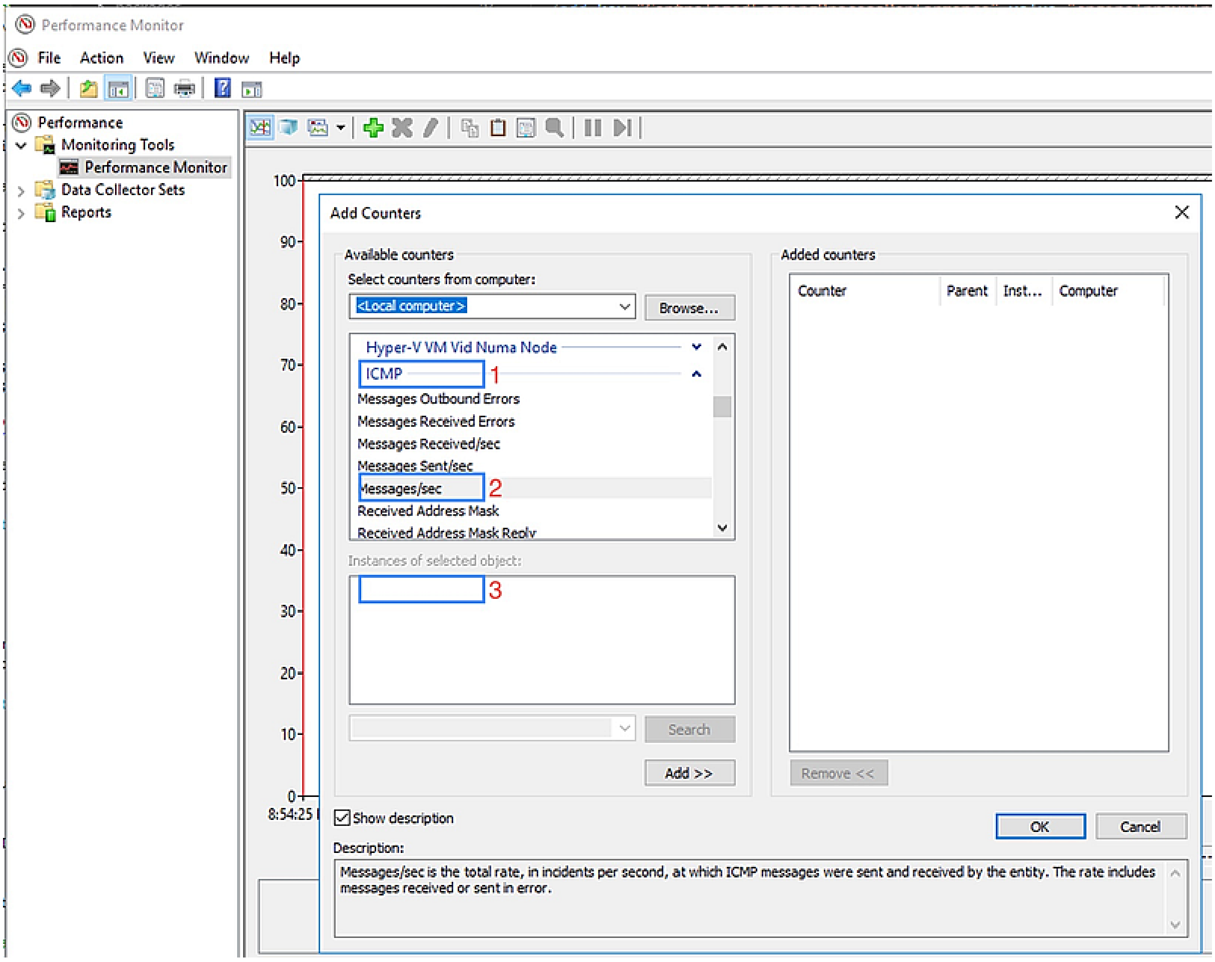
```
<add key="CentralizedLoggingPlatformPerformanceCounters" value="CPUUtilizationPercent;Processor;%
Processor Time;_Total|CPUUserUtilizationPercent;Processor;% User Time;_
Total|CPUInterruptPercent;Processor;% Interrupt Time;_Total|CPUQueueLength;System;Processor Queue
Length|DiskFreeSpacePercent;LogicalDisk;% Free Space;_Total|DiskIdleTimePercent;PhysicalDisk;% Idle
Time;_Total|DiskTimePerReadSeconds;PhysicalDisk;Avg. Disk sec/Read;_
Total|DiskTimePerWriteSeconds;PhysicalDisk;Avg. Disk sec/Write;_
Total|DiskIOQueueLength;PhysicalDisk;Avg. Disk Queue Length;_Total|MemoryCacheBytes;Memory;Cache
Bytes|MemoryCommittedBytesInUsePercent;Memory;% Committed Bytes In
Use|MemoryAvailableMBytes;Memory;Available MBytes|MemoryFreePageEntries;Memory;Free System Page Table
Entries|MemoryPoolNonPagedBytes;Memory;Pool Nonpaged Bytes|MemoryPoolPagedBytes;Memory;Pool Paged
```

```
Bytes|MemoryPagesPerSecond;Memory;Pages/sec|NetworkBytesPerSecond;Network Interface;Bytes
Total/sec;*|NetworkOutputQueueLength;Network Interface;Output Queue Length;*|" />
```

Default value table

Database field	Counter Category	Counter name	Counter instance
CPUInterruptPercent	Processor	% Interrupt Time	_Total
CPUQueueLength	System	Processor Queue Length	
CPUUserUtilizationPercent	Processor	% User Time	_Total
CPUUtilizationPercent	Processor	% Processor Time	_Total
DiskFreeSpacePercent	LogicalDisk	% Free Space	_Total
DiskIdleTimePercent	PhysicalDisk	% Idle Time	_Total
DiskIOQueueLength	PhysicalDisk	Avg. Disk Queue Length	_Total
DiskTimePerReadSeconds	PhysicalDisk	Avg. Disk sec/Read	_Total
DiskTimePerWriteSeconds	PhysicalDisk	Avg. Disk sec/Write	_Total
MemoryAvailableMBytes	Memory	Available MBytes	
MemoryCacheBytes	Memory	Cache Bytes	
MemoryCommittedBytesInUsePercent	Memory	% Committed Bytes In Use	
MemoryFreePageEntries	Memory	Free System Page Table Entries	
MemoryPagesPerSecond	Memory	Pages/sec	
MemoryPoolNonPagedBytes	Memory	Pool Nonpaged Bytes	
NetworkBytesPerSecond	Network Interface	Bytes Total/sec	*
NetworkOutputQueueLength	Network Interface	Output Queue Length	*

The format of the `CentralizedLoggingPlatformPerformanceCounters` setting value follows the naming and structure used by the Windows Performance Monitor as shown below:



Messages/sec is the total rate, in incidents per second, at which ICMP messages were sent and received by the entity. The rate includes messages received or sent in error.

Additional performance counters may be added by modifying the value of the `CentralizedLoggingPlatformPerformanceCounters` setting. The value of the setting uses the following format:

- Each performance counter is separated by the bar "|" character.
- Each performance counter is made up of four fields separated by the semicolon ";" character and whitespace can be added freely to improve readability.
  - The first field is optional and corresponds to the name that will be used to store the performance counter in the database. If the field is not provided, the name of the counter is used instead.
  - The second field is the counter category and corresponds to one (1) on the image above.
  - The third field is the counter and corresponds to two (2) on the image above.
  - The fourth field is the counter instance and corresponds to three (3) on the image above, this field is optional and when set to the wildcard character "*" or not provided all instances available are collected.

To add the counter shown above, the following would need to be appended to the `CentralizedLoggingPlatformPerformanceCounters` setting: "`| Test1 ; ICMP ; Messages/sec ; *`". The performance counter will now be collected based on the period set by the *CentralizedLoggingPlatformPerformanceCheckPeriod* setting and can be retrieved from the database using the following query against the Qlogs database:

```
SELECT
    BTRIM(e.payload->>'Test1', '"')::INTEGER AS icmp_msg_per_sec
FROM public.log_entries e
WHERE e.logger = 'Qlik.Logging.Service.Platform.Performance'
```

Platform metrics may be viewed by executing the following query against the Qlogs database:

```
SELECT * FROM public.view_platform_metrics;
```

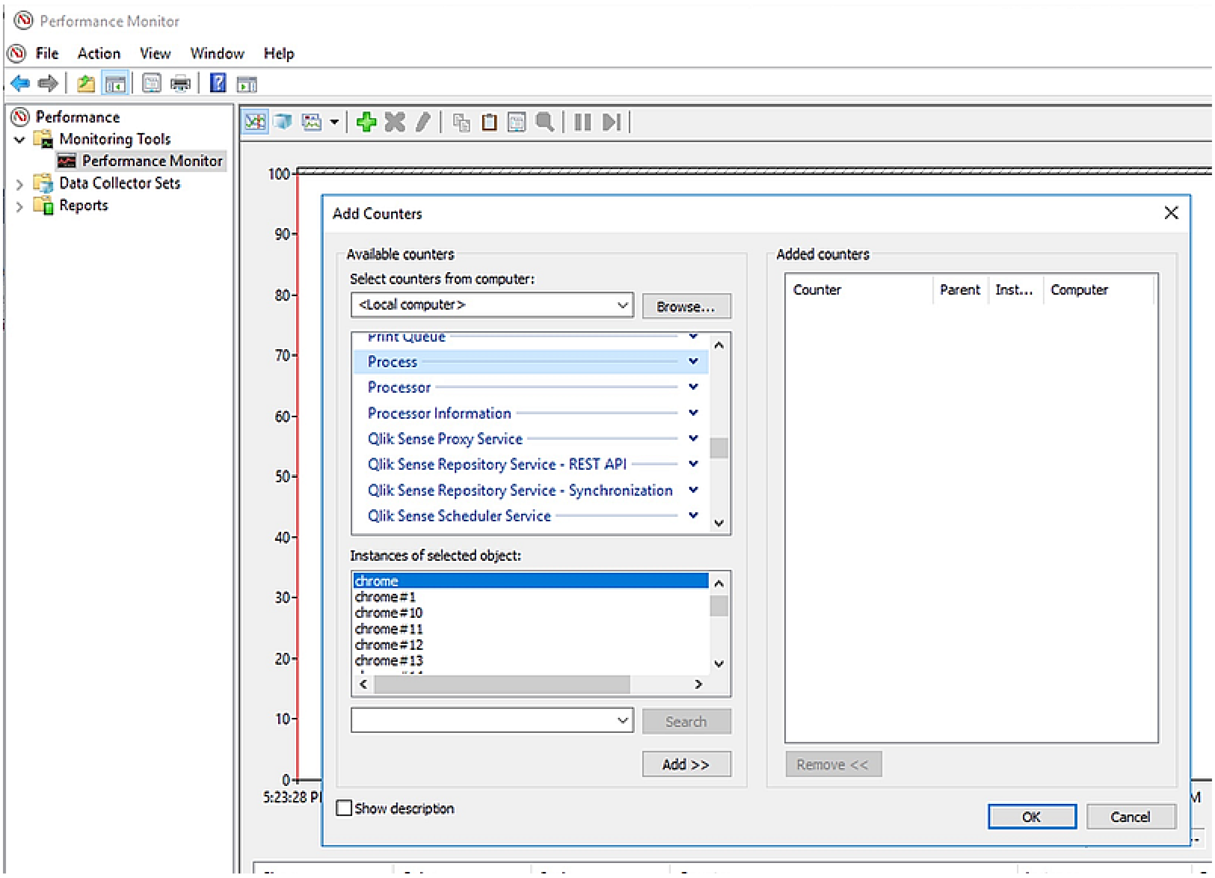
Process metrics

The period used to monitor process metrics is controlled by the `CentralizedLoggingProcessPerformanceCheckPeriod` setting. A value of zero(0) disables this functionality and its default value is five (5) minutes expressed in milliseconds (5 * 60 * 1000 = 300000).

Performance metrics are collected for the processes configured via the `CentralizedLoggingProcessPerformance` setting and its default value is:

```
<add key="CentralizedLoggingProcessPerformance" value="engine|proxy|repository|scheduler" />
```

The format of the setting also follows the naming and structure used by the Windows Performance Monitor as shown below:



All performance counters available are collected for each process listed with each process separated by the bar "|" character. To collect metrics for additional processes, the "simple name" of the process would need to be appended to the `CentralizedLoggingProcessPerformance` setting, for example to add chrome as shown above, the string "| chrome" would be appended to the **CentralizedLoggingProcessPerformance** setting.

Process metrics may be viewed by executing the following query against the QLogs database:

```
SELECT * FROM public.view_process_metrics;
```

Export

The Qlik Logging Service is able to export the log data it manages. This complementary feature will make it easier to share log data with Qlik's tech support. The export command is available when using the Qlik Logging Service as a command line tool.

Usage: Qlik.Logging.Service.exe <export> [[export_options]]

Although the exported data is not encrypted, it is in binary format due to compression and not readable by tools other than the Qlik Logging Service.

Export

The "export" command will copy all log data managed by the Qlik Logging Service to a destination. The following command options can be used to control what gets exported and where to:

`--output_file` or `-o` (Required): Identifies the location where exported data should be persisted. The location can be a disk file or the address of a Qlik Logging Service that is listening for exported data. When specifying a file, the export process will create or replace the target file. When specifying a waiting service the format is: <hostname or host ip address>[: port], that is the name of the host or its ip address optionally followed by a colon (":") and the port number used by the listening host. If no port is specified, the default `7081` will be used.

`--hostname` or `-h`: Limits the exported data to the subset that originated from the specified host. The comparison is case insensitive.

`--level` or `-l`: Limits the exported data to the subset that is equal to the specified level. The comparison is case insensitive.

`--to_timestamp`: Limits the exported data to the subset created before the specified date. See "Date Format" section below for more details.

`--from_timestamp`: Limits the exported data to the subset created after the specified date. See "Date Format" section below for more details.

`--process_name`: Limits the exported data to the subset that originated from the specified process. The comparison is case insensitive.

Examples

These commands where tested from a "Command Prompt" without elevated permissions.

To export all contents to a file:

```
Qlik.Logging.Service export --output_file C:\Temp\qlogs.bin
```

To export all content originating from host "Node1" between 09/01/2018 and 09/02/2018 at 10:00PM to a waiting network reachable host listening on port 80:

```
Qlik.Logging.Service export --from_timestamp "2018-09-01" --to_timestamp "2018-09-02 22:00:00" --
hostname node1 --otuput_file somehost.domain.com:80
```

Date format

The format expected is "YYYY-MM-DD [[HH:MM:SS]]", the date portion is required while the time portion is optional and when omitted defaults to "00:00:00".

- Date
  - YYYY: year, 4 digits
  - MM: month, 2 digits, 01 through 12
  - DD: day of the month, 2 digits, 01 through 31
- Time
  - HH: hour, 2 digits, 00 through 23
  - MM: minute, 2 digits, 00 through 59
  - SS: second, 2 digits, 00 through 59

Technical notes

Unicode null character

Processes using the Qlik Logging Service may under very rare circumstances log entries containing the Unicode character NULL (\u0000). This character is problematic for many tools as it is most commonly used to denote the end of a string. When a Unicode character has been inserted into the QLogs database, you may see error messages like this one from PostgreSQL:

```
ERROR: unsupported Unicode escape sequence
DETAIL: \u0000 cannot be converted to text.
CONTEXT: JSON data, line 1: {"Message":...
STATEMENT: fetch 200 in "SQL_CUR4"
```

Executing the following script against the Qlogs database will replace all occurrences of the Unicode NULL character "\u0000" with the string "<UNICODE_NULL>"

```
SET statement_timeout = 0;
SET lock_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;

CREATE OR REPLACE FUNCTION public.validate_payload(IN payload JSON) RETURNS TEXT AS $$
BEGIN
  RETURN payload->>'XXXXX';
EXCEPTION
  WHEN OTHERS THEN RETURN 'XXXXX';
END
$$ LANGUAGE plpgsql;

DO $$
BEGIN
  UPDATE public.log_entries
    SET payload = REGEXP_REPLACE(payload::TEXT, '\\u0000', '<UNICODE_NULL>', 'g')::JSON
  WHERE id IN (
    SELECT d.id
      FROM (
          SELECT id, validate_payload(payload) AS payload
          FROM public.log_entries
      ) AS d
      WHERE d.payload = 'XXXXX'
  );
  UPDATE public.archive_entries
    SET payload = REGEXP_REPLACE(payload::TEXT, '\\u0000', '<UNICODE_NULL>', 'g')::JSON
  WHERE id IN (
    SELECT d.id
      FROM (
          SELECT id, validate_payload(payload) AS payload
          FROM public.archive_entries
      ) AS d
      WHERE d.payload = 'XXXXX'
  );
END $$ LANGUAGE plpgsql;

DROP FUNCTION IF EXISTS public.validate_payload(IN payload JSON);
```

Configuration file reset

If for some reason the Qlik Logging Service configuration file `QlikCentralizedLogging.config` becomes corrupt or in some way unreadable, it can be deleted and a new file with default values will be created the next time the service is started. The same is true for individual settings, removing a setting from the configuration file will trigger its replacement with default values during the next service start sequence.

Multi-node configuration

In a multi-node environment, the usually runs on all nodes.

In a multi-node environment, the Qlik Logging Service usually runs on all nodes. With the default configuration, all nodes will be executing database management functions. This is not a problem, but a more efficient configuration would designate a single node as the one in charge of maintaining the database. For example, on a three (3) node deployment, the Qlik Logging Service running on each node could be configured as follows:

Central node and rim node 1

```
<!-- Centralized Logging Configuration -->

<!-- DB Size Management disabled -->
<add key="CentralizedLoggingDBSizeCheckPeriod" value="0" />

<!-- Archive Management disabled -->
<add key="CentralizedLoggingDBArchiveCheckPeriod" value="0" />

<!-- Purge Management disabled -->
<add key="CentralizedLoggingDBPurgeCheckPeriod" value="0" />

<!-- DB Stats Collection disabled -->
<add key="CentralizedLoggingDBStatsCheckPeriod" value="0" />

<!-- Platform Performance Collection enabled (1 minute) -->
<add key="CentralizedLoggingPlatformPerformanceCheckPeriod" value="60000" />

<!-- Process Performance Collection enabled (1 minute) -->
<add key="CentralizedLoggingProcessPerformanceCheckPeriod" value="60000" />
```

Rim node 2 (Database management)

```
<!-- Centralized Logging Configuration -->

<!-- DB Size Management enabled (5 minutes, 6GB max size) -->
<add key="CentralizedLoggingDBSizeCheckPeriod" value="300000" />
<add key="MaximumDatabaseSizeInGB" value="6" />

<!-- Archive Management enabled (60 minutes) -->
<add key="CentralizedLoggingDBArchiveCheckPeriod" value="3600000" />

<!-- Purge Management enabled (60 minutes) -->
<add key="CentralizedLoggingDBPurgeCheckPeriod" value="3600000" />

<!-- DB Stats Collection enabled (5 minutes) -->
<add key="CentralizedLoggingDBStatsCheckPeriod" value="300000" />

<!-- Platform Performance Collection enabled (1 minute) -->
<add key="CentralizedLoggingPlatformPerformanceCheckPeriod" value="60000" />

<!-- Process Performance Collection enabled (1 minute) -->
<add key="CentralizedLoggingProcessPerformanceCheckPeriod" value="60000" />
```

8.8 Requirements

The requirements described in this section must be fulfilled for the Qlik Sense logging to function properly.

Securing the file system

The system administrator must secure the file system so that the log files cannot be tampered with.

By default, the account used for the Qlik Sense installation gets full permissions for the log folder, %ProgramData%\Qlik\Sense\Log, whereas the Users group only gets read permission. No other accounts or users get any permissions for the log folder.

Synchronizing time

The nodes within a Qlik Sense site must have synchronized time.

For the date and time stamps to be correct, all nodes within a site must be configured to synchronize their system clocks with either an internal or an external Network Time Protocol (NTP) service to ensure that all log entries are time-stamped accurately. NTP is a networking protocol for synchronizing the clocks of computer systems over packet-switched, variable-latency data networks.

Setting time zone

It is recommended that every node within a Qlik Sense site is set to the correct time zone so that the time zone corresponds to the geographical location of the node.

8.9 Storage

The default log files are stored in folders under *%ProgramData%\Qlik\Sense\Log*. The local log configuration file can be used to set up the logging so that the log files are also stored in another location.

Log folder

The following table describes the contents of the *%ProgramData%\Qlik\Sense\Log* folder.

Folder	Sub-folder	Files	Description
\AppMigration			This folder contains log files related to the Migration Service.
\BrokerService			This folder contains log files related to the Broker Service.
\DataProfiling			This folder contains log files related to the Data Profiling Service.
\Engine		*<MachineName>_Exit_Engine_<Date>.txt*	NewSet. is a temporary log file that is used by the the Qlik Sense Engine Service (QES) until the log pipe to the Qlik Sense Repository Service (QRS) is up and running. NewSet. log file is not archived.
		<MachineName>_Start_Engine_<Date>.txt	NewSet. is a temporary log file that is used by the the Qlik Sense Engine Service (QES) until the log pipe to the Qlik Sense Repository Service (QRS) is up and running. NewSet. log file is not archived.
	\Audit	*<MachineName>_AuditActivity_<Service>.txt*	This log tracks user-related actions.
		<MachineName>_AuditSecurity_<Service>.txt	This log contains information on security-related actions.
	\System	*<MachineName>_Service_<Service>.txt*	This log contains information on service and system operations, including all errors.

Folder	Sub-folder	Files	Description
	\Trace	*<MachineName>_<Facility>_<Service>.txt*	The trace log files are stored in this folder. See: *Trace logs (page 268)*
\HubService			This folder contains log files related to the Hub Service.
\Printing	*\Audit*	*<MachineName>_AuditActivity_<Service>.txt*	This log tracks user-related actions.
		<MachineName>_AuditSecurity_<Service>.txt	This log contains information on security-related actions.
	\System	*<MachineName>_Service_<Service>.txt*	This log contains information on service and system operations, including all errors.
	\Trace	*<MachineName>_<Facility>_<Service>.txt*	The trace log files are stored in this folder. See: *Trace logs (page 268)*
\Proxy	*\Audit*	*<MachineName>_AuditActivity_<Service>.txt*	This log tracks user-related actions.
		<MachineName>_AuditSecurity_<Service>.txt	This log contains information on security-related actions.
	\System	*<MachineName>_Service_<Service>.txt*	This log contains information on service and system operations, including all errors.
	\Trace	*<MachineName>_<Facility>_<Service>.txt*	The trace log files are stored in this folder. See: *Trace logs (page 268)*
\Repository	*\Audit*	*<MachineName>_AuditActivity_<Service>.txt*	This log tracks user-related actions.
		<MachineName>_AuditSecurity_<Service>.txt	This log contains information on security-related actions.

Folder	Sub-folder	Files	Description
	\System	*<MachineName>_Service_<Service>.txt*	This log contains information on service and system operations, including all errors.
	\Trace	*<MachineName>_<Facility>_<Service>.txt*	The trace log files are stored in this folder. See: *Trace logs (page 268)*
\Scheduler	*\Audit*	*<MachineName>_AuditActivity_<Service>.txt*	This log tracks user-related actions.
		<MachineName>_AuditSecurity_<Service>.txt	This log contains information on security-related actions.
	\System	*<MachineName>_Service_<Service>.txt*	This log contains information on service and system operations, including all errors.
	\Trace	*<MachineName>_<Facility>_<Service>.txt*	The trace log files are stored in this folder. See: *Trace logs (page 268)*
\Script			This folder contains log files related to app reloads.
\WebExtensionService			This folder contains log files related to the Web Extension Service.

Archived log files

Archived log files are by default stored in the *\\<server>\<share>\ArchivedLogs* folder. You define the location of the file share folder during installation. Archived log files have the file extension *.log*, whereas active log files have the extension *.txt*.

See also:

Local log configuration file (page 283)

8.10 Naming

The Qlik Sense log files are named in accordance to the following file rollover procedure:

1. The log is stored in a file named *<MachineName>_<LogType>_<Service>.txt*.
2. When the file is full or a pre-defined amount of time has passed, the file extension is automatically changed to *.log* and a time stamp is appended to the file name for uniqueness and archiving. This means

that the new file name becomes *<MachineName>_<LogType>_<Service>_<YYYY-MM-DDTHH.mm.ss>Z.log*. The file is then moved to the repository database on the central node by the Qlik Sense Repository Service (QRS) and archived.

3. A new log file, named *<MachineName>_<LogType>_<Service>.txt*, is created.

> ⚠ *If the .log file is deleted before being copied to the repository database on the central node, the file is lost and cannot be recreated.*

The format of the file name is as follows:

- *<MachineName>* = Name of the server where the log was created.
- *<LogType>* = The type of events that are covered by the log.
- *<Service>* = The service that the log originates from (for example, Proxy or Repository).
- *<YYYY-MM-DDTHH.mm.ss>Z* = Time stamp for when the log file was closed for new entries, where:
  - *YYYY*: Year
  - *MM*: Month
  - *DD*: Day in month
  - *T*: Delimiter, time designator
  - *HH*: Hour
  - *mm*: Minutes
  - *ss*: Seconds
  - *Z*: UTC designator, indicates that the time stamp is in UTC format

8.11 Rows

The first row of each log file contains a header that, in turn, contains the names of all fields, separated by tabs.

Each log entry is one row and the characters listed in the following table are replaced with Unicode characters.

Character	Unicode replacement	Description
\t	\u21d4	Symbol for horizontal tabulation, HT.
\n	\u2193	Symbol for line feed, LF.
\f	\u2192	Symbol for form feed, FF.
\r	\u21b5	Symbol for carriage return, CR.

8.12 Fields

This section describes the fields in the Qlik Sense log files.

Audit activity log

The following table lists the fields in the audit activity log, *<MachineName>_AuditActivity_<Service>.txt*.

The Audit activity log does not include a Severity field. This is because all rows in the log have the same log level.

Field	**Format**	**Description**
Sequence#	Int	1 - 2147483647 by default, but can be configured using custom logging as described in *Appenders (page 280)*. Each row in the log starts with a sequence number that is used to ensure that the log is not tampered with (that is, that no rows are inserted or deleted). The sequence number wraps a) when the last sequence number is reached, or b) when the logging, for some reason, is restarted without the last sequence number being reached.
ProductVersion	String	The version number of the Qlik Sense service (for example, 1.2.1.3).
Timestamp	ISO 8601	Timestamp in ISO 8601 format, `YYYYMMDDThhmmss.fffK`, where: • `YYYY`: Year • `MM`: Month • `DD`: Day in month • `T`: Delimiter • `hh`: Hour • `mm`: Minutes • `ss`: Seconds • `fff`: Milliseconds • `K`: Time zone offset For example, 20110805T145657.000+0200 means year 2011, month 8, day 5 at 14:56:57 GMT+2.
Hostname	String	The name of the server.
Id	String	A unique identifier of the log entry (added by Log4net).
Description	String	A human-readable message that summarizes the action in the system. Format: Command=<CommandName>;Result=<ReturnCode (Int)>;ResultText=<Description, Success, or Error message>
ProxySessionId	String	The ID of the proxy session. 0 = Internal system command or a command that does not go through the QPS

Field	Format	Description
ProxyPackageId	String	A unique ID of each HTTP(S) package that passes through the Qlik Sense Proxy Service (QPS). 0 = Internal system command or a command that does not go through the QPS
RequestSequenceId	String	The combination of RequestSequenceId and ProxyPackageId is unique for every row in a log file and creates the timeline for the proxy session. The combination also forms a primary key in the log file. The initial RequestSequenceId is an integer. Subrequests are linked to the initial request by adding a dot and an ID for the subrequest: • Initial request: RequestSequenceId = 1 • Subrequest 1 based on the initial request: RequestSequenceId = 1.0 • Subrequest 2 based on the initial request: RequestSequenceId = 1.1 0 = Internal system command or a command that does not go through the Qlik Sense Engine Service (QES)
UserDirectory	String	The user directory linked to the logged in Qlik Sense user.
UserId	String	The Qlik Sense user that initiated the command. System = Internal system command
ObjectId	String	The internal ID of the object. Used to link system actions to user actions. 0 = Cannot get the ID of the object In some cases the ObjectId field contains multiple IDs, separated by the "|" (pipe) sign. **Example: ObjectId field containing multiple IDs** Log event: Start reload task Contents of the ObjectId field: ed5715cd-2d7f-44ec-825f-44084efb3443|d63c7e4e-6089-4314-b60f-ed47ba6c35cc • First ID: The ID of the task. • Second ID: The ID of the app.

Field	Format	Description
ObjectName	String	The human-readable name of the object. The ObjectName is linked to the ObjectId. Not available = Cannot link the ObjectName to the ObjectId or the ObjectId is missing In some cases the ObjectName field contains multiple names. **Example: ObjectName field containing multiple names** Log event: Start reload task Contents of the ObjectName field: MyReload|MyApp • First identifier (MyReload): The name of the task. • Second identifier (MyApp): The name of the app. The list of ObjectNames always matches the list of ObjectIds, meaning that the ObjectName in the first position is identified by the ID in the corresponding position of the ObjectId field. In this example the following IDs apply (see also the description of the ObjectId field): • MyReload = ed5715cd-2d7f-44ec-825f-44084efb3443 • MyApp = d63c7e4e-6089-4314-b60f-ed47ba6c35cc
Service	String	The Qlik Sense service on the server that hosts the process.
Origin	String	The origin of the request: • AppAccess • ManagementAccess • Not available
Context	String	The context of the command. The context can be a URL that is linked to the command or a short version of the module path linked to the command.
Command	String	The core name of the use case or system command.
Result	String	Return code: • 0, 200 - 226: Success • Any other number: Error
Message	String	Text that describes the log entry. If the request is successful, this field contains "success".
Id2	String	A unique row identifier (the checksum is added by Log4Net).

Audit security log

The following table lists the fields in the audit security log, *<MachineName>_AuditSecurity_<Service>.txt*.

This log is not available for the Qlik Sense Engine Service (QES).

The Audit security log does not include a Severity field. This is because all rows in the log have the same log level.

Field	Format	Description
Sequence#	Int	1 - 2147483647 by default, but can be configured using custom logging as described in *Appenders (page 280)*. Each row in the log starts with a sequence number that is used to ensure that the log is not tampered with (that is, that no rows are inserted or deleted). The sequence number wraps a) when the last sequence number is reached, or b) when the logging, for some reason, is restarted without the last sequence number being reached.
ProductVersion	String	The version number of the Qlik Sense service (for example, 1.2.1.3).
Timestamp	ISO 8601	Timestamp in ISO 8601 format, `YYYYMMDDThhmmss.fffK`, where: • `YYYY`: Year • `MM`: Month • `DD`: Day in month • `T`: Delimiter • `hh`: Hour • `mm`: Minutes • `ss`: Seconds • `fff`: Milliseconds • `K`: Time zone offset For example, 20110805T145657.000+0200 means year 2011, month 8, day 5 at 14:56:57 GMT+2.
HostName	String	The name of the server.
Id	GUID	A unique identifier of the log entry (added by Log4net).
Description	String	A human-readable message that summarizes the action in the system. Format: Command=<CommandName>;Result=<ReturnCode (Int)>;ResultText=<Description, Success, or Error message>

Field	Format	Description
ProxySessionId	String	The ID of the proxy session. 0 = Internal system command or a command that does not go through the QPS
ProxyPackageId	String	A unique ID of each HTTP(S) package that passes through the Qlik Sense Proxy Service (QPS). 0 = Internal system command or a command that does not go through the QPS
RequestSequenceId	String	The combination of RequestSequenceId and ProxyPackageId is unique for every row in a log file and creates the timeline for the proxy session. The combination also forms a primary key in the log file. The initial RequestSequenceId is an integer. Subrequests are linked to the initial request by adding a dot and an ID for the subrequest: • Initial request: RequestSequenceId = 1 • Subrequest 1 based on the initial request: RequestSequenceId = 1.0 • Subrequest 2 based on the initial request: RequestSequenceId = 1.1 0 = Internal system command or a command that does not go through the Qlik Sense Engine Service (QES)
UserDirectory	String	The user directory linked to the logged in Qlik Sense user. System = Internal system command
UserId	String	The Qlik Sense user that initiated the command. System = Internal system command

Field	Format	Description
ObjectId	String	The internal ID of the object. Used to link system actions to user actions. 0 = Cannot get the ID of the object In some cases the ObjectId field contains multiple IDs, separated by the "|" (pipe) sign. **Example: ObjectId field containing multiple IDs** Log event: Start reload task Contents of the ObjectId field: ed5715cd-2d7f-44ec-825f-44084efb3443|d63c7e4e-6089-4314-b60f-ed47ba6c35cc • First ID: The ID of the task. • Second ID: The ID of the app.
ObjectName	String	The human-readable name of the object. The ObjectName is linked to the ObjectId. Not available = Cannot link the ObjectName to the ObjectId or the ObjectId is missing In some cases the ObjectName field contains multiple names. **Example: ObjectName field containing multiple names** Log event: Start reload task Contents of the ObjectName field: MyReload|MyApp • First identifier (MyReload): The name of the task. • Second identifier (MyApp): The name of the app. The list of ObjectNames always matches the list of ObjectIds, meaning that the ObjectName in the first position is identified by the ID in the corresponding position of the ObjectId field. In this example the following IDs apply (see also the description of the ObjectId field): • MyReload = ed5715cd-2d7f-44ec-825f-44084efb3443 • MyApp = d63c7e4e-6089-4314-b60f-ed47ba6c35cc
SecurityClass	String	A categorization of the security-related information: • Security: Access to resources, authentication, authorization • License: License access, license usage, license allocation • Certificate: Certificate-related information
ClientHostAddress	String	The hostname/IP address of the client.

Field	Format	Description
Service	String	The Qlik Sense service on the server that hosts the process.
Origin	String	The origin of the request: • AppAccess • ManagementAccess • Not available
Context	String	The context of the command. The context can be a URL that is linked to the command or a short version of the module path linked to the command.
Command	String	The core name of the use case or system command.
Result	String	Return code: • 0, 200 - 226: Success • Any other number: Error
Message	String	Text that describes the log entry. If the request is successful, this field contains "success".
Checksum	ID	Each row has a checksum. The security log file also includes a file signature.

Server log

The following table lists the fields in the service log, *<MachineName>_Service_<Service>.txt*.

Field	Format	Description
Sequence#	Int	1 - 2147483647 by default, but can be configured using custom logging as described in *Appenders (page 280)*. Each row in the log starts with a sequence number that is used to ensure that the log is not tampered with (that is, that no rows are inserted or deleted). The sequence number wraps a) when the last sequence number is reached, or b) when the logging, for some reason, is restarted without the last sequence number being reached.
ProductVersion	String	The version number of the Qlik Sense service (for example, 1.2.1.3).

Field	Format	Description
Timestamp	ISO 8601	Timestamp in ISO 8601 format, `YYYYMMDDThhmmss.fffK`, where: • `YYYY`: Year • `MM`: Month • `DD`: Day in month • `T`: Delimiter • `hh`: Hour • `mm`: Minutes • `ss`: Seconds • `fff`: Milliseconds • `K`: Time zone offset For example, 20110805T145657.000+0200 means year 2011, month 8, day 5 at 14:56:57 GMT+2.
Severity	String	Row log level, can be configured using custom logging as described in *Appenders (page 280)*: • Debug: Information useful to developers for debugging purposes. This level is not useful during normal operation as it generates vast amounts of logging information. • Info: Normal operational messages that may be harvested for reporting, measuring throughput, and so on. No action is required. • Warn: Not an error message, but an indication that an error will occur, if no action is taken (for example, the file system is 85% full). • Error: Messages regarding unexpected situations and errors that prevent the server from operating normally. • Fatal: Messages that the Qlik Sense service or application has to shut down in order to prevent data loss.
HostName	String	The hostname of the server that runs the process or executes the task.
Id	GUID	A unique identifier of the log entry (added by Log4net).
Description	String	A human-readable message that summarizes the action in the system. Format: Command=<CommandName>;Result=<ReturnCode (Int)>;ResultText=<Description, Success, or Error message>
ProxySessionId	String	The ID of the proxy session. 0 = Internal system command or a command that does not go through the QPS

Field	Format	Description
ProxyPackageId	String	A unique ID of each HTTP(S) package that passes through the Qlik Sense Proxy Service (QPS). 0 = Internal system command or a command that does not go through the QPS
RequestSequenceId	String	The combination of RequestSequenceId and ProxyPackageId is unique for every row in a log file and creates the timeline for the proxy session. The combination also forms a primary key in the log file. The initial RequestSequenceId is an integer. Subrequests are linked to the initial request by adding a dot and an ID for the subrequest: • Initial request: RequestSequenceId = 1 • Subrequest 1 based on the initial request: RequestSequenceId = 1.0 • Subrequest 2 based on the initial request: RequestSequenceId = 1.1 0 = Internal system command or a command that does not go through the Qlik Sense Engine Service (QES)
UserDirectory	String	The user directory linked to the logged in Qlik Sense user. System = Internal system command
UserId	String	The Qlik Sense user that initiated the command. System = Internal system command
ObjectId	String	The internal ID of the object. Used to link system actions to user actions. 0 = Cannot get the ID of the object In some cases the ObjectId field contains multiple IDs, separated by the "|" (pipe) sign. **Example: ObjectId field containing multiple IDs** Log event: Start reload task Contents of the ObjectId field: ed5715cd-2d7f-44ec-825f-44084efb3443|d63c7e4e-6089-4314-b60f-ed47ba6c35cc • First ID: The ID of the task. • Second ID: The ID of the app.

Field	Format	Description
ObjectName	String	The human-readable name of the object. The ObjectName is linked to the ObjectId. Not available = Cannot link the ObjectName to the ObjectId or the ObjectId is missing In some cases the ObjectName field contains multiple names. **Example: ObjectName field containing multiple names** Log event: Start reload task Contents of the ObjectName field: MyReload|MyApp • First identifier (MyReload): The name of the task. • Second identifier (MyApp): The name of the app. The list of ObjectNames always matches the list of ObjectIds, meaning that the ObjectName in the first position is identified by the ID in the corresponding position of the ObjectId field. In this example the following IDs apply (see also the description of the ObjectId field): • MyReload = ed5715cd-2d7f-44ec-825f-44084efb3443 • MyApp = d63c7e4e-6089-4314-b60f-ed47ba6c35cc
Service	String	The Qlik Sense service on the server that hosts the process.
Origin	String	The origin of the request: • AppAccess • ManagementAccess • Not available
Context	String	The context of the command. The context can be Internal System command or User Activity command (based on URL for the command).
Command	String	The core name of the use case or system command.
Result	Int	Return code: • 0, 200 - 226: Success • Any other number: Error
Message	String	Text that describes the log entry. If the request is successful, this field contains "success".
Id2	String	A unique row identifier (the checksum is added by Log4Net).

Qlik Sense engine service log fields

The following table lists the fields that are unique for the Qlik Sense Engine Service (QES) logs.

Field	Format	Description
EngineTimestamp	ISO 8601	The date and time when the QES wrote the log message to file. Timestamp in ISO 8601 format, `YYYYMMDDThhmmss.fffK`, where: • `YYYY`: Year • `MM`: Month • `DD`: Day in month • `T`: Delimiter • `hh`: Hour • `mm`: Minutes • `ss`: Seconds • `fff`: Milliseconds • `K`: Time zone offset For example, 20110805T145657.000+0200 means year 2011, month 8, day 5 at 14:56:57 GMT+2.
EngineVersion	String	The version number of the QES that executed the request.

8.13 Trace logs

The legacy logging framework is still available in Qlik Sense, but the logs are as of Qlik Sense version 2.0 referred to as trace logs. The log files remain the same, in the old logging format, but they are stored in a new location.

Storage

The trace log files are stored in the *%ProgramData%\Qlik\Sense\Log\<Service>\Trace* folder.

Naming

The trace log files are named in accordance to the following file rollover procedure:

1. The log is stored in a file named *<MachineName>_<Facility>_<Service>.txt*.
2. When the file is full or a pre-defined amount of time has passed, the file extension is automatically changed to *.log* and a time stamp is appended to the file name for uniqueness and archiving. This means that the new file name becomes *<MachineName>_<Facility>_<Service>_<YYYY-MM-DDTHH.mm.ss>Z.log*. The file is then moved to the repository database on the central node by the Qlik Sense Repository Service (QRS) and archived.
3. A new log file, named *<MachineName>_<Facility>_<Service>.txt*, is created.

⚠ *If the .log file is deleted before being copied to the repository database on the central node, the file is lost and cannot be recreated.*

The format of the file name is as follows:

- *<Machine>* = Name of the server where the log was created.
- *<Facility>* = The type of events that are covered by the log.
  Logger (page 271)
- *<Service>* = The service that the log originates from (for example, Proxy or Repository).
- *<YYYY-MM-DDTHH.mm.ss>Z* = Time stamp for when the log file was closed for new entries, where:
  - *YYYY*: Year
  - *MM*: Month
  - *DD*: Day in month
  - *T*: Delimiter, time designator
  - *HH*: Hour
  - *mm*: Minutes
  - *ss*: Seconds
  - *Z*: UTC designator, indicates that the time stamp is in UTC format

See also:

Logger (page 271)

Rows

The first row of each log file contains a header that, in turn, contains the names of all fields, separated by tabs.

Each log entry is one row and the characters listed in the following table are replaced with Unicode characters.

Character	Unicode replacement	Description
\t	\u21d4	Symbol for horizontal tabulation, HT.
\n	\u2193	Symbol for line feed, LF.
\f	\u2192	Symbol for form feed, FF.
\r	\u21b5	Symbol for carriage return, CR.

Fields

This section describes the fields in the trace log files.

Common fields

The following table lists the fields (in order of appearance) included in all trace log files.

Field	Description
Sequence#	1 - 2147483647 by default, but can be configured using custom logging as described in Qlik Sense *Appenders (page 280)*. Each row in the log starts with a sequence number that is used to ensure that the log is not tampered with (that is, that no rows are inserted or deleted). The sequence number wraps either when the last sequence number is reached or when the logging, for some reason, is restarted without the last sequence number being reached.
Timestamp	Timestamp in ISO 8601 format, `YYYYMMDDThhmmss.fffK`, where: • `YYYY`: Year • `MM`: Month • `DD`: Day in month • `T`: Delimiter • `hh`: Hour • `mm`: Minutes • `ss`: Seconds • `fff`: Milliseconds • `K`: Time zone offset For example, 20110805T145657.000+0200 means year 2011, month 8, day 5 at 14:56:57 GMT+2.
Level	Row log level, can be configured using custom logging as described in Qlik Sense *Appenders (page 280)*: • Debug: Information useful to developers for debugging purposes. This level is not useful during normal operation since it generates vast amounts of logging information. • Info: Normal operational messages that may be harvested for reporting, measuring throughput, and so on. No action required. • Warn: Not an error message, but an indication that an error may occur, if no action is taken (for example, the file system is 85% full). Each item must be resolved within a given time. • Error: Non-urgent failures that are relayed to developers or administrators. Each item must be resolved within a given time. • Fatal: Indicates a failure in a primary system (for example, loss of primary ISP connection) and must be corrected immediately. • Off: No logs, except for license logs, are produced.
Hostname	Server name.

Field	Description
Logger	Logger in `<Facility>.<Service>.<Fully qualified name of class>` format, where: - `<Facility>`: - Application: Log events that are related to the app running in Qlik Sense. - Audit: Log events that provide an audit trail of a user's activities and administration of the Qlik Sense platform. - Exit: Log events that are related to the shutdown of the Qlik Sense Engine Service (QES). - License: Log events that are related to the Qlik Sense license. - ManagementConsole: Log events that are related to the Qlik Management Console (QMC). - Performance: Log events that are related to the performance of the Qlik Sense platform or app. - QixPerformance: Log events that are related to the performance of the QIX protocol in the QES. - Security: Log events that are related to security issues. - Session: Log events that are related to the termination of a proxy session. - SSE: Log events that are related to server-side extensions. - Synchronization: Log events that are related to the synchronization of the Qlik Sense Repository Service (QRS) instances in a multi-node site. - System: Log events that are related to the Qlik Sense platform and not to the app running on the platform (for example, log messages related to the QMC, QRS, Qlik Sense Proxy Service (QPS), and so on). - TaskExecution: Log events that are related to the execution of tasks by the Qlik Sense Scheduler Service (QSS). - Traffic: Log events that are related to debugging. - UserManagement: Log events that are related to the management of the users. - `<Service>`: The Qlik Sense service that the log originates from (for example, QRS or QPS). - `<Fully qualified name of class>`: Indicates the part of the service that generated the log message.
Thread	Thread name or Managed Thread ID (if available).
Id	Globally Unique Identifier (GUID) for the log message.
ServiceUser	Name of the user or account used by the Qlik Sense service.
Message	Log message.

Field	Description
Exception	Exception message. *This field is only present when there is an exception message.*
StackTrace	A trace to the place in Qlik Sense where the exception occurred. *This field is only present when the Exception field is present.*
ProxySessionId	The ID of the proxy session for the user. *This field is not present in all log files.*
Id2 or Checksum	The last field in a log entry either contains an Id2 or a Checksum: • Id2: Log message GUID (same as Id described earlier). This is the normal value. • Checksum: To protect logs that contain sensitive information (for example, audit, security, and license logs) from tampering, the last field in such log entries contains a cryptographic hash of the entire row up to the hash itself.

Additional fields

The common fields are present in all trace log files. Some trace logs contain additional fields, which are listed in this section. In addition, optional fields can be defined.

Application log

Qlik Sense Repository Service (QRS)

The following fields are specific to the Application log for the QRS:

- Application: The name of the application (if there is a name to associate with the log entry).

Qlik Sense Scheduler Service (QSS)

The following fields are specific to the Application log for the QSS:

- Application: The name of the application (if there is a name to associate with the log entry).

See also:

Common fields (page 269)

Audit log

Qlik Sense Repository Service (QRS)

The following fields are specific to the Audit log for the QRS:

- Action: The action that the user performed (add, update, delete, export).
- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- ResourceId: The ID of the resource on which the user performed the action.

Qlik Sense Proxy Service (QPS)

The following fields are specific to the Audit log for the QPS:

- ConnectionId: The ID of the connection.

  ActiveConnections field in *Performance log (page 274)*
- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- TicketId: The ID of the login ticket that was issued for the user. The ticket ID exists until it is consumed by the QPS.
- IpAddress: The IP address of the client.
- AppId: The ID of the app (empty if no app is loaded).
- TargetHost: The call from the client is forwarded to a Qlik Sense Engine Service (QES) or QRS. This field contains the name of the machine on which the service is running.
- VirtualProxy: The virtual proxy prefix in `{prefix}` format.

Qlik Sense Engine Service (QES)

The following fields are specific to the Audit log for the QES:

- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- EngineTimestamp: The time when the QES wrote the log message to file.
- EngineThread: The ID of the thread that was used when the QES wrote the log message to file.
- ProcessId: The ID of the QES process from which the log message originates.
- ServerStatus: The time when the QES started.
- AppId: The ID of the app.
- Type: The type of operation that the user performed to generate the audit message.
- Qlik Sense User: The user who generated the audit message.

See also:

Common fields (page 269)

License log

Qlik Sense Repository Service (QRS)

The following fields are specific to the License log for the QRS:

- AccessTypeId: The ID of the access type entity.
- AccessType: The name of the access type (LoginAccess or UserAccess).

- Operation: The operation that was performed (Add, Update, Delete, UsageGranted, UsageDenied, Available, Timedout, or Unquarantined).
- UserName: The name of the user (who, for example, uses an access pass).
- UserId: The ID of the user in Qlik Sense.

See also:

Common fields (page 269)

Performance log

Qlik Sense Repository Service (QRS)

The following fields are specific to the Performance log for the QRS:

- Tracenumber: A unique ID for the call to the QRS REST API.
- Httpmethod: The HTTP method that was used (Get, Put, Post, or Delete).
- Url: The URL that was used.
- Resourcetype: The type of resource.
- Method: The backend code that was called.
- Elapsedmilliseconds: The time (in milliseconds) to complete the call to the QRS REST API.

Example: Get http://mytest/cars/4

- Httpmethod: Get
- Url: http://mytest/cars/4
- Resourcetype: cars
- Method: get/cars/{0}

Qlik Sense Proxy Service (QPS)

The following fields are specific to the Performance log for the QPS:

- ActiveConnections: The number of active connections (in any form or shape) from the client.

  A connection is a stream (that is, a socket) between a Qlik Sense client and the Qlik Sense Proxy Service (QPS). This stream is often connected to another stream, which runs from the QPS to the Qlik Sense Repository Service (QRS) or the Qlik Sense Engine Service (QES). The two streams allow the client to communicate with the QRS or the QES.

- ActiveStreams: The number of active data streams (that is, sockets), either from the browser to the QPS or from the QPS to the QRS or the QES.
- ActiveSessions: The number of active sessions in the QPS.

  A Qlik Sense user gets a proxy session when the user has been authenticated. The session is terminated after a certain period of inactivity.

- LoadBalancingDecisions: The number of users who currently have at least one engine session.

- PrintingLoadBalancingDecisions: The number of users who have been load balanced to the Qlik Sense Printing Service (QPR).
- Tickets: The number of issued login tickets that have not yet been consumed.
- ActiveClientWebsockets: The number of active WebSockets between the client and the QPS.
- ActiveEngineWebsockets: The number of active WebSockets between the QPS and the target Qlik Sense service.

The logging entries are also available as metrics; see Proxy service.

Qlik Sense Engine Service (QES)

Each entry (that is, row) in the Performance log corresponds to a snapshot (that is, a number of measurements) of the performance of the QES at the given point in time.

The following fields are specific to the Performance log for the QES:

- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- EngineTimestamp: The time when the QES wrote the log message to file.
- EngineThread: The ID of the thread that was used when the QES wrote the log message to file.
- ProcessId: The ID of the QES process from which the log message originates.
- Exe Type: The configuration type (release or debug version) of the QES process.
- Exe Version: The version number of the QES process.
- Server Started: The time when the QES started.
- Entry Type: The reason (Server Starting, Normal, or Server Shutting Down) for the log entry in the Performance log.
- ActiveDocSessions: The number of active engine sessions at the given point in time.
- DocSessions: The number of engine sessions at the given point in time.
- ActiveAnonymousDocSessions: The number of active anonymous engine sessions at the given point in time.
- AnonymousDocSessions: The number of anonymous engine sessions at the given point in time.
- ActiveTunneledDocSessions: The number of active tunneled engine sessions at the given point in time.
- TunneledDocSessions: The number of tunneled engine sessions at the given point in time.
- DocSessionStarts: The number of started engine sessions since the previous snapshot.
- ActiveDocs: The number of active apps in the QES at the given point in time.
- RefDocs: The number of apps in the QES at the given point in time.
- LoadedDocs: The number of loaded apps in the QES at the given point in time.
- DocLoads: The number of app loads in the QES since the previous snapshot.
- DocLoadFails: The number of failed app loads in the QES since the previous snapshot.
- Calls: The number of calls to the QES since the previous snapshot.
- Selections: The number of selections in the QES since the previous snapshot.

- ActiveIpAddrs: The number of IP addresses of active connected clients in the QES at the given point in time.
- IpAddrs: The number of IP addresses of all connected clients in the QES at the given point in time.
- ActiveUsers: The number of active users in the QES at the given point in time.
- Users: The total number of users in the QES at the given point in time.
- CPULoad: A measurement of the load on the CPU on which the QES runs at the given point in time.
- VMCommitted(MB): The committed Virtual Memory (in megabytes) at the given point in time.
- VMAllocated(MB): The allocated Virtual Memory (in megabytes) at the given point in time.
- VMFree(MB): The freed Virtual Memory (in megabytes) at the given point in time.
- VMLargestFreeBlock(MB): The largest freed Virtual Memory block (in megabytes) at the given point in time.

See also:

Common fields (page 269)

QIX performance log

Qlik Sense Engine Service (QES)

The following fields are specific to the QIX performance log for the QES:

- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- EngineTimestamp: The time when the QES wrote the log message to file.
- EngineThread: The ID of the thread that was used when the QES wrote the log message to file.
- ProcessId: The ID of the QES process from which the log message originates.
- CServerId: The ID of the server instance that handled the request.
- SessionId: The ID of the engine session for which the QIX method call was made.
- Server Started: The time when the QES started.
- Method: The name of the QIX method that was called.
- RequestId: The ID of the request in which the QIX method call was handled.
- Target: The memory address of the target for the QIX method call.
- RequestException: The ID of an exception (if any) that occurred as a result of the QIX method call.
- ProcessTime: The amount of time that was needed to process the request.
- WorkTime: The amount of time that the request did actual work.
- LockTime: The amount of time that the request had to wait for an internal lock.
- ValidateTime: The amount of time that the request used for validation.
- Handle: The ID of the interface that handled the request. The interface can be Global, a certain sheet, a certain object, or similar.

See also:

Common fields (page 269)

Qlik Management Consolelog

> *The Qlik Management Console log is not created until there is an event (for example, an error message) for the Qlik Management Console (QMC) to write in the log.*

Qlik Sense Repository Service (QRS)

The following fields are specific to the Qlik Management Console log for the QRS:

- Browser: The web browser that is used to run the QMC.

See also:

Common fields (page 269)

Server-side extension log

Qlik Sense Engine Service (QES)

The following fields are specific to the server-side extension (SSE) log for the QES:

- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- EngineTimestamp: The time when the QES wrote the log message to file.
- EngineThread: The ID of the thread that was used when the QES wrote the log message to file.
- ProcessId: The ID of the QES process from which the log message originates.
- QixRequestId: The ID established by the initiator of the request. If this member is not present, the RPC call is assumed to be a notification.
- AppId: The ID of the app that includes the call to the server-side extension (SSE) plugin through an analytic connection.
- App Title: The title of the app that includes the call to the SSE plugin through an analytic connection.
- SSEPlugin: If the log message was created during a call to the SSE plugin, the mapping/alias of that plugin, for example, SSEPython for a Python plugin. If the log message was created without a call to the SSE plugin, for example, while initializing the SSE, the value is a dash (-).
- SSEPluginAddress: Two elements separated by a colon that define the analytic connection to the SSE plugin.
  - <Host>: DNS name (or IP-address) of the plugin.
  - <Port>: Port on which the plugin listens, typically 50051.

  For example, localhost:50051.

See also:

Common fields (page 269)

Session log

Qlik Sense Engine Service (QES)

The following fields are specific to the Session log for the QES:

- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- EngineTimestamp: The time when the QES wrote the log message to file.
- EngineThread: The ID of the thread that was used when the QES wrote the log message to file.
- ProcessId: The ID of the QES process from which the log message originates.
- Exe Type: The configuration type (release or debug version) of the QES process.
- Exe Version: The version number of the QES process.
- Server Started: The time when the QES started.
- AppId: The ID of the app that was loaded by the finished engine session.
- App Title: The title of the loaded app that was used during the finished engine session.
- Doc Timestamp: The last modified timestamp of the app that was loaded by the finished engine session.
- Qlik Sense User: The user that started the finished engine session.
- Exit Reason: The reason for the engine session to finish.
- Session Start: The time when the engine session started.
- Session Duration: The duration (in milliseconds) of the finished engine session.
- CPU Spent (s): The CPU time (in seconds) that was spent handling requests during the finished engine session.
- Bytes Received: The number of bytes of data that were received during the engine session.
- Bytes Sent: The number of bytes of data that were sent during the engine session.
- Calls: The number of calls that were made during the engine session.
- Selections: The number of selections that were made during the engine session.
- Authenticated User: The authenticated user that used the engine session.
- Client Machine Identification: The ID of the client machine that opened the engine session.
- Serial Number: The serial number that was used during the engine session.
- Client Type: The type of client that was used for the engine session.
- Client Build Version: The build version of the client.
- Secure Protocol: An on/off flag that indicates whether the protocol was run over a secure connection.

See also:

Common fields (page 269)

System log

Qlik Sense Scheduler Service (QSS)

The following fields are specific to the System log for the QSS:

- TaskName: The name of the task that was executed.
- TaskId: The ID of the task that was executed.
- User: The name of the user who executed the task. When the QSS starts a scheduled execution of a task, the QSS is listed as user.
- ExecutionId: A unique ID that identifies the execution of the task. A task gets a new ExecutionId every time it is executed.
- AppName: The name of the app that executed the task (if any).
- AppId: The ID of the app that executed the task (if any).

Qlik Sense Engine Service (QES)

The following fields are specific to the System log for the QES:

- ActiveUserDirectory: The user directory for the active user who was logged in when the log message was generated in the QES.
- ActiveUserId: The user ID for the active user who was logged in when the log message was generated in the QES.
- EngineTimestamp: The time when the QES wrote the log message to file.
- EngineThread: The ID of the thread that was used when the QES wrote the log message to file.
- ProcessId: The ID of the QES process from which the log message originates.
- Server Started: The time when the QES started.

See also:

Common fields (page 269)

Task execution log

Qlik Sense Scheduler Service (QSS)

The following fields are specific to the Task execution log for the QSS:

- TaskId: A unique ID of the task that was executed.
- TaskName: The name of the task that was executed.
- AppId: A unique ID of the app that executed the task (if any).
- AppName: The name of the app that executed the task (if any).
- ExecutionId: A unique ID that identifies the execution of a task. A task gets a new ExecutionId every time it is executed.
- ExecutionNodeId: A unique ID that identifies the node in the site on which the specific execution of the task was performed.
- Status: The result of the execution of the task (successful, failed, aborted, skipped, or retry).
- StartTime: The time when the execution of the task started.
- StopTime: The time when the execution of the task stopped.
- Duration: The time (in milliseconds) for the execution of the task to be completed.
- FailureReason: Empty, unless an error occurred during the execution of the task.

See also:

🗋 *Common fields (page 269)*

Traffic log

Qlik Sense Engine Service (QES)

The following fields are specific to the traffic log for the QES:

- ActiveUserDirectory: The user directory for the user.
- ActiveUserId: The ID of the user.
- EngineTimestamp: The time when the QES wrote the log message to file.
- EngineThread: The ID of the thread that was used when the QES wrote the log message to file.
- ProcessId: The ID of the QES process from which the log message originates.

See also:

🗋 *Common fields (page 269)*

8.14 Configuring the logging

The standard logging in Qlik Sense is configured using the Qlik Management Console (QMC).

Customized logging is setup using appenders and the local log configuration file, *LocalLogConfig.xml*.

Appenders

The logging in Qlik Sense implements a custom appender, QSRollingFileAppender, which is based on the log4net component. The custom appender is used internally by the Qlik Sense logging system.

QSRollingFileAppender and some of the built-in, predefined appenders in the log4net framework can be used to configure customized logging, which is specified in the local log configuration file, *LocalLogConfig.xml*.

QSRollingFileAppender can also log events in the local log file (for example, the Microsoft Windows event log) or send log information to a remote log server.

QSRollingFileAppender

QSRollingFileAppender inherits from `log4net.Appenders.FileAppender` and all parameters, except for `AppendToFile`, are also available to QSRollingFileAppender. QSRollingFileAppender stores the log files in accordance to the `MaxFileSize` and `MaxFileTime` parameters.

Configuring the appender

The QSRollingFileAppender configuration is as follows:

```
<appender name="MyQSRollingFileAppender"
type="Qlik.Sense.Logging.log4net.Appender.QSRollingFileAppender">
<param name="threshold" value="info" />
```

```
<param name="encoding" value="utf-8" />
<param name="file" value="C:/ProgramData/Qlik/Sense/Log/output.log"/>
<param name="maximumfiletime" value="720" />
<param name="maximumfilesize" value="512KB" />
<layout type="log4net.Layout.PatternLayout">
<converter>
  <param name="name" value="rownum" />
  <param name="type" value="Qlik.Sense.Logging.log4net.Layout.Pattern.CounterPatternConverter" />
</converter>
<converter>
  <param name="name" value="longIso8601date" />
  <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.Iso8601TimeOffsetPatternConverter" />
</converter>
<converter>
  <param name="name" value="hostname" />
  <param name="type" value="Qlik.Sense.Logging.log4net.Layout.Pattern.HostNamePatternConverter" />
</converter>
<converter>
  <param name="name" value="guid" />
  <param name="type" value="Qlik.Sense.Logging.log4net.Layout.Pattern.GuidPatternConverter" />
</converter>
<converter>
  <param name="name" value="user" />
  <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.ServiceUserNameCachedPatternConverter" />
</converter>
<converter>
  <param name="name" value="encodedmessage" />
  <param name="type" value="Qlik.Sense.Logging.log4net.Layout.Pattern.EncodedMessagePatternConverter"
/>
</converter>
<converter>
  <param name="name" value="encodedexception" />
  <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.EncodedExceptionPatternConverter" />
</converter>
<param name="ignoresexception" value="false" />
<param name="header"
value="Sequence#&#x9;Timestamp&#x9;Level&#x9;Hostname&#x9;Logger&#x9;Thread&#x9;Id&#x9;User&#x9;
Message&#x9;Exception&#x9;Id2&#xD;&#xA;" />
<param name="conversionpattern" value="%rownum
{9999}&#x9;%longIso8601date&#x9;%level&#x9;%hostname&#x9;%logger&#x9;%thread&#x9;
%guid&#x9;%user&#x9;%encodedmessage&#x9;%encodedexception{innermostmessage}&#x9;%guid%newline" />
</layout>
</appender>
```

Converters

`log4net.Layout.PatternLayout` and a couple of custom converters are used to format rows in logs based on QSRollingFileAppender:

- `Qlik.Sense.Logging.log4net.Layout.Pattern.CounterPatternConverter`: Add a sequence number to the log row. Parameter:

    - Integer: The last number of the sequence before it is reset.
- `Qlik.Sense.Logging.log4net.Layout.Pattern.Iso8601TimeOffsetPatternConverter`: Add a time stamp (local time with time offset in ISO 8601 format) to the log row.
- `Qlik.Sense.Logging.log4net.Layout.Pattern.HostNamePatternConverter`: Add the host name to the log row.
- `Qlik.Sense.Logging.log4net.Layout.Pattern.GuidPatternConverter`: Add a GUID to the log row.
- `Qlik.Sense.Logging.log4net.Layout.Pattern.ServiceUserNameCachedPatternConverter`: Add the username to the log row.
- `Qlik.Sense.Logging.log4net.Layout.Pattern.EncodedMessagePatternConverter`: Add the encoded message to the log row.
- `Qlik.Sense.Logging.log4net.Layout.Pattern.EncodedExceptionPatternConverter`: Add information on the logged exception to the log row. Parameter (one of the following):
    - MESSAGE: The message in the logged exception.
    - INNERMOSTMESSAGE: The message in the innermost exception of the logged exception.
    - SOURCE: The source of the exception (that is, the name of the app or the object that caused the error).
    - STACKTRACE: The stack trace for the exception.
    - TARGETSITE: The target site for the exception (that is, the method that threw the current exception).
    - HELPLINK: Link to the help file associated with the exception.

Built-in log4net appenders

In addition to the Qlik Sense custom appender, QSRollingFileAppender, the log4net framework comes with a set of built-in predefined appenders that also can be used in the local log configuration file, *LocalLogConfig.xml*:

- AdoNetAppender
- AnsiColorTerminalAppender
- AspNetTraceAppender
- ColoredConsoleAppender
- ConsoleAppender
- EventLogAppender
- FileAppender
- NetSendAppender
- RemoteSyslogAppender
- RemotingAppender
- RollingFileAppender
- SmtpAppender
- SmtpPickupDirAppender
- TelnetAppender
- UdpAppender

Each appender has its own set of parameters to control the output.

See also:

Apache Logging Services

Example: EventLogAppender

The following example shows how to use the EventLogAppender in the local log configuration file, *LocalLogConfig.xml*, for the Qlik Sense Proxy Service (QPS). In the example, all QPS audit log entries at warning level are sent to the Microsoft Windows event log.

```
<appender name="EventLogAppender" type="log4net.Appender.EventLogAppender" >
    <param name="threshold" value="warn" />
    <param name="applicationName" value="Qlik Sense Proxy Service" />
    <layout type="log4net.Layout.PatternLayout">
      <param name="conversionPattern" value="%message" />
    </layout>
  </appender>
     <logger name="Audit.Proxy">
                               <appender-ref ref="EventLogAppender" />
   </logger>
```

Example: SmtpAppender

The following example shows how to use the SmtpAppender in the local log configuration file, *LocalLogConfig.xml*, for the Qlik Sense Proxy Service (QPS). In the example, all QPS audit log entries at warning level are sent to an email address (`to@domain.com`).

```
<appender name="MyMailAppender" type="log4net.Appender.SmtpAppender">
<param name="threshold" value="warn" />
<param name="to" value="to@domain.com" />
<param name="from" value="from@domain.com" />
<param name="subject" value="test logging message" />
<param name="smtpHost" value="SMTPServer.domain.com" />
<param name="port" value="25" />
<param name="bufferSize" value="512" />
<param name="lossy" value="true" />
<layout type="log4net.Layout.PatternLayout">
<param name="conversionPattern" value="%newline%date %-5level %message%newline%newline%newline" />
</layout>
</appender>
     <logger name="Audit.Proxy">
                               <appender-ref ref="MyMailAppender" />
   </logger>
```

Local log configuration file

The logging in Qlik Sense can be set up to produce customized logging as an addition to the default logging.

To set up customized logging, create a local log configuration file named *LocalLogConfig.xml* in the *%ProgramData%\Qlik\Sense\<Service>* folder.

The logging defined by the local log configuration file does not affect the default logging.

Requirements

The requirements described in this section must be fulfilled for the customized logging to function properly.

Conforming to the XML schema

The local log configuration file must conform to the XML schema because the Qlik Sense Repository Service (QRS), Qlik Sense Proxy Service (QPS), and Qlik Sense Scheduler Service (QSS) have built-in schema validation.

If the local log configuration file is not accepted by the services, an error is logged in the System log.

Maximum file size

The size of the local log configuration file must not exceed 1 MB.

XML schema

The XML schema for the local log configuration file, *LocalLogConfig.xml*, is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

       <xs:complexType name="ParamType">
               <xs:attribute name="name" type="xs:string" use="required"/>
               <xs:attribute name="value" type="xs:string" use="required"/>
       </xs:complexType>

       <xs:simpleType  name="AppenderNameType">
               <xs:restriction base="xs:string">
                       <xs:pattern value="[^$].*"/> <!-- '$' is not allowed as prefix-->
               </xs:restriction>
       </xs:simpleType>

       <xs:complexType name="ConverterType">
               <xs:sequence>
                       <xs:element name="param" minOccurs="0" maxOccurs="unbounded" type="ParamType" />
               </xs:sequence>
       </xs:complexType>

       <xs:complexType name="FilterType">
               <xs:sequence>
                       <xs:element name="param" minOccurs="0" maxOccurs="unbounded" type="ParamType" />
               </xs:sequence>
               <xs:attribute name="class" type="xs:string" use="optional"/>  <!-- log4cxx-->
               <xs:attribute name="type" type="xs:string" use="optional"/>   <!-- log4net-->
       </xs:complexType>

       <xs:complexType name="EvaluatorType">
               <xs:sequence>
                       <xs:element name="param" minOccurs="0" maxOccurs="unbounded" type="ParamType" />
               </xs:sequence>
               <xs:attribute name="class" type="xs:string" use="optional"/>  <!-- log4cxx-->
               <xs:attribute name="type" type="xs:string" use="optional"/>   <!-- log4net-->
       </xs:complexType>

       <xs:complexType name="LayoutType">
               <xs:sequence>
```

```
                        <xs:element name="converter" minOccurs="0" maxOccurs="unbounded" type="ConverterType" />
                        <xs:element name="param" minOccurs="0" maxOccurs="unbounded" type="ParamType" />
                </xs:sequence>
                <xs:attribute name="class" type="xs:string" use="optional"/>  <!-- log4cxx-->
                <xs:attribute name="type" type="xs:string" use="optional"/>   <!-- log4net-->
        </xs:complexType>

        <xs:complexType name="AppenderType">
                <xs:sequence>
                        <xs:element name="filter" minOccurs="0" maxOccurs="unbounded" type="FilterType" />
                        <xs:element name="evaluator" minOccurs="0" type="EvaluatorType" />
                        <xs:element name="lossyevaluator" minOccurs="0" type="EvaluatorType" />  <!-- log4net-->
                        <xs:element name="param" minOccurs="0" maxOccurs="unbounded" type="ParamType" />
                        <xs:element name="layout" minOccurs="1" type="LayoutType" />
                </xs:sequence>
                <xs:attribute name="name" type="AppenderNameType" use="required"/>
                <xs:attribute name="class" type="xs:string" use="optional"/>  <!-- log4cxx-->
                <xs:attribute name="type" type="xs:string" use="optional"/>   <!-- log4net-->
        </xs:complexType>

        <xs:complexType name="AppenderRefType">
                <xs:attribute name="ref" type="AppenderNameType" use="required"/>
        </xs:complexType>

        <xs:complexType name="RootType">
                <xs:sequence>
                        <xs:element name="appender-ref" type="AppenderRefType" minOccurs="0" maxOccurs="unbounded" />
                </xs:sequence>
        </xs:complexType>

        <xs:complexType name="LoggerType">
                <xs:sequence>
                        <xs:element name="appender-ref" type="AppenderRefType" minOccurs="0" maxOccurs="unbounded" />
                </xs:sequence>
                <xs:attribute name="name" type="AppenderNameType" use="required"/>
        </xs:complexType>

        <xs:element name="configuration">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="appender" type="AppenderType" minOccurs="0" maxOccurs="unbounded" />
                                <xs:element name="root" type="RootType" minOccurs="0" />
                                <xs:element name="logger" type="LoggerType" minOccurs="0" maxOccurs="unbounded" />
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

Example:

In this example, the local log configuration file is configured to write the system logs at debug level in *%ProgramData%\Qlik\Sense\Log\Proxy\Debug_System_Proxy.txt*.

```
<?xml version="1.0"?>
```

```
<configuration>
   <appender name="LocalApp_AppenderSystem"
type="Qlik.Sense.Logging.log4net.Appender.QSRollingFileAppender">
      <param name="threshold" value="debug" />
      <param name="encoding" value="utf-8" />
      <param name="file" value="C:\ProgramData\Qlik\Sense\Log\Proxy\Debug_System_Proxy.txt" />
      <param name="maximumfiletime" value="720" />
      <param name="maximumfilesize" value="512KB" />
      <layout type="log4net.Layout.PatternLayout">
         <converter>
            <param name="name" value="rownum" />
            <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.CounterPatternConverter" />
         </converter>
         <converter>
            <param name="name" value="longIso8601date" />
            <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.Iso8601TimeOffsetPatternConverter" />
         </converter>
         <converter>
            <param name="name" value="hostname" />
            <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.HostNamePatternConverter" />
         </converter>
         <converter>
            <param name="name" value="guid" />
            <param name="type" value="Qlik.Sense.Logging.log4net.Layout.Pattern.GuidPatternConverter"
/>
         </converter>
         <converter>
            <param name="name" value="serviceuser" />
            <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.ServiceUserNameCachedPatternConverter" />
         </converter>
         <converter>
            <param name="name" value="encodedmessage" />
            <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.EncodedMessagePatternConverter" />
         </converter>
         <converter>
            <param name="name" value="encodedexception" />
            <param name="type"
value="Qlik.Sense.Logging.log4net.Layout.Pattern.EncodedExceptionPatternConverter" />
         </converter>
         <param name="ignoresexception" value="false" />
         <param name="header" value="Sequence#&#x9;Timestamp&#x9;Level&#x9;Hostname&#x9;
            Logger&#x9;Thread&#x9;Id&#x9;ServiceUser&#x9;Message&#x9;Exception&#x9;
            ActiveUserDirectory&#x9;ActiveUserId&#x9;ProxyTimestamp&#x9;ProxyThread&#x9;
            Id2&#xD;&#xA;" />
         <param name="conversionpattern" value="%rownum{9999}&#x9;%longIso8601date&#x9;
            %level&#x9;%hostname&#x9;%logger&#x9;%thread&#x9;%guid&#x9;%serviceuser&#x9;
            %encodedmessage{1000000}&#x9;%encodedexception{innermostmessage:1000000}&#x9;
            %property{ActiveUserDirectory}&#x9;%property{ActiveUserId}&#x9;
            %property{ProxyTimestamp}&#x9;%property{ProxyThread}&#x9;%guid%newline" />
      </layout>
   </appender>
   <logger name="System.Proxy">
      <appender-ref ref="LocalApp_AppenderSystem" />
   </logger>
</configuration>
```

See also:

- *Converters (page 281)*

9 Qlik product licenses

Here is a summary of the license options that are available for the different Qlik Sense related products. Licensing allows you to manage the usage of the Qlik Sense software in your organization.

9.1 Qlik Sense Enterprise

Qlik Sense Enterprise is the server version of Qlik Sense that you can deploy on a single node, or on multiple nodes.

There are two major license types: one that is user-based and one that is token-based.

- With a user-based license, you allocate professional access and analyzer access. The LEF determines the distribution of the two access types.
- With a token-based license, you allocate user access and login access. The LEF determines the number of tokens that you can allocate to the two access types.
  In addition to these two types there is the analyzer capacity license that is similar to analyzer access regarding available features, but where consumption is time based (analyzer time).

If you want to set up Qlik Cloud Services or Qlik Sense Enterprise for elastic deployments, please contact your Qlik representative or Qlik Support to obtain a valid license for the setup.

9.2 User-based licenses

User-based licenses grant a predefined number of professional and analyzer access allocations.

Professional access

You allocate professional access to an identified user who needs access to all features in a Qlik Sense installation. With a professional access, you can create, edit, and publish sheets or apps, as well as administer a Qlik Sense site.

Analyzer access

You allocate analyzer access to an identified user to allow the user to access streams and apps in the hub. With an analyzer access, you can access sheets and apps created by others. You cannot create, edit, or publish sheets or apps, but can create and publish stories, bookmarks and snapshots based on data in apps. In addition, you can create bookmarks, print objects, stories, and sheets, and export data from an object to Excel.

Analyzer capacity

Analyzer capacity is a consumption-based license type that is similar to analyzer access regarding available features. Analyzer capacity is available to any users, including anonymous users, and they share the monthly analyzer time allotment, which is consumed in units of six minutes.

9.3 Token-based licenses

Licensing in Qlik Sense is based on tokens. You use tokens to allocate access passes to users so that they can access Qlik Sense. The License Enabler File (LEF) determines the number of tokens that you can allocate to different access types. A user without an access pass cannot access apps. There are two types of access passes:

- User access pass - assigned to unique and identified users allowing them unlimited access to apps, streams, and other resources.
- Login access pass - allocates a block of passes to a group for infrequent or anonymous access. Allows full access for a limited period.

When you allocate tokens, the number of available tokens is reduced. Each access type costs a certain number of tokens, and if the token balance is zero or insufficient, you cannot allocate more to the access types. You can free up tokens and choose to use the tokens differently. The number of tokens for the Qlik Sense site can be increased or decreased by activating a new license.

For more information on access passes and the token consumption model, see Managing licenses.

9.4 License Enabler File

The License Enabler File (LEF) defines the terms of your license and the access types that you can allocate to users. There are two license types: one that is user-based and one that is token-based.

User-based license

User-based licenses grant a predefined number of professional and analyzer access allocations. The distribution of the access types is determined by the LEF.

Token-based license

Every Qlik Sense site needs a site license. The License Enabler File (LEF) defines the number of tokens available on your site, which you can manage from the central node. You activate the license on the site license properties page in the QMC, where you enter owner information, serial number, and control number.

Licenses for different deployments

In most cases a single Qlik Sense license is sufficient for both a single-node and a multi-node deployment. However, in a deployment that includes, for example, a development site and a production site, two licenses are needed. Likewise, if you have several production sites that are geographically distributed, you will need one license per site.

9.5 Qlik Sense Desktop

Qlik Sense Desktopis a free version of Qlik Sense that you can download on your computer, see www.qlik.com. You have to register for a Qlik account to be able to download Qlik Sense Desktop. On the same page, you also need to accept the Qlik Sense Desktop license agreement. In addition, when you install the software, you need to accept the software license agreement.

9.6 Qlik DataMarket

Qlik DataMarket offers a collection of up-to-date data from external sources accessible directly from within Qlik Sense. The available data includes current and historical weather and demographic data, currency exchange rates, as well as business, economic, and societal data.

Qlik DataMarket has two license options, one free and one licensed. The free option gives you access to a limited data set. The licensed option gives you access to premium data packages. You activate the license in the same way as for Qlik Sense Enterprise. On the Qlik DataMarket page, under **License and tokens**, you enter owner information, serial number, and control number.

9.7 Qlik NPrinting

You can install and configure Qlik NPrinting to connect to QlikView documents or Qlik Sense apps. The licensing requirements and procedures are different depending on if you connect Qlik NPrinting to QlikView or Qlik Sense.

Qlik NPrinting versions 16.0.0.0 and later are licensed by a LEF. If you are using an earlier version of Qlik NPrinting, we suggest that you upgrade to Qlik NPrinting versions 16.0.0.0 or later.

For details regarding licensing Qlik NPrinting with QlikView, QlikView Desktop, and Qlik Sense, see Licensing Qlik NPrinting.

9.8 Qlik Sense Cloud

To create and share apps with others using Qlik Sense Cloud, you need to create a Qlik account and log in. No license is required. With a Qlik account, you can also download Qlik Sense Desktop.

9.9 Overview of Qlik license usage

Here is a summary of how licenses are used in Qlik Sense and related products. Licensing allows you to manage the usage of the Qlik Sense software in your organization.

Qlik Sense Enterprise

The License Enabler File (LEF) defines the terms of your license and the access types that you can allocate to users. There are two license types: one that is user-based and one that is token-based.

- User-based license: you can allocate professional access and analyzer access. The LEF determines the distribution of the two access types.
- Token-based license: you can allocate user access and login access. The LEF determines the number of tokens that you can allocate to the two access types.

An access type allows users to access streams and apps within a Qlik Sense site.

If you want to set up Qlik Cloud Services or Qlik Sense Enterprise for elastic deployments, please contact your Qlik representative or Qlik Support to obtain a valid license for the setup.

User-based licenses

User-based licenses grant a predefined number of professional and analyzer access allocations. The distribution of the access types is determined by the LEF.

Professional access

You allocate professional access to an identified user to allow the user to access streams and apps within a Qlik Sense site. The professional access is intended for users who need access to all features in a Qlik Sense installation. A user with professional access can create, edit, and publish sheets or apps, and make full use of the available features, including administration of a Qlik Sense site. There is a direct relationship between the access type (professional access) and the user. If you deallocate professional access from a user, the access type is put in quarantine, given that it has been used within the last seven days. If it has not been used within the last seven days, the professional access is released immediately. You can reinstate quarantined professional access, to the same user, within seven days.

Summary professional access:

- Assigned to an identified user.
- Daily access to analyze or create content.
- Unlimited access to streams, apps, and other resources.
- Maximum number of parallel sessions is five.

Analyzer access

You allocate analyzer access to an identified user to allow the user to access streams and apps in the hub. The analyzer access is intended for users who consume sheets and apps created by others. A user with analyzer access cannot create, edit, or publish sheets or apps, but can create and publish stories, bookmarks and snapshots based on data in apps. The user can also create bookmarks, print objects, stories, and sheets, and export data from an object to Excel. There is a direct relationship between the access type (analyzer access) and the user. If you deallocate analyzer access from a user, the access type is put in quarantine, given that it has been used within the last seven days. If it has not been used within the last seven days, the analyzer access is released immediately. You can reinstate quarantined analyzer access, to the same user, within seven days.

Summary analyzer access:

- Assigned to an identified user.
- Daily access to analyze (but not create, edit, or publish), sheets and apps in the hub.
- Supports creation of bookmarks.
- Supports printing of objects, stories, and sheets.
- Supports export of data from an object to Excel.
- Maximum number of parallel sessions is five.

Analyzer capacity

Analyzer capacity is a consumption-based license type that is similar to analyzer access regarding available features. Analyzer capacity is available to any users, including anonymous users, and they share the monthly analyzer time allotment, which is consumed in units of six minutes.

Summary analyzer capacity access

- The same features available as with analyzer access.
- Assigned to a group of users, including anonymous users.
- Monthly subscription to a defined amount of minutes.
- Consumption in 6 minute blocks. No consumption during idleness.
- Overage can be added to subscription.

Token-based licenses

When you activate your Qlik Sense Enterprise site license, you get a certain number of tokens. The number of tokens is determined by the License Enabler File (LEF). You use these tokens to allocate access to Qlik Sense, either as user access passes or login access passes.

User access

User access is for a unique, identified user of Qlik Sense. The user access pass provides unlimited access to apps, streams, and other resources, and is intended for frequent users of Qlik Sense.

Login access

Login access is for one or more users of Qlik Sense. You create license rules to specify which users the login access is available for. The login access pass gives access to streams and apps for a predefined amount of time, and is intended for infrequent users of Qlik Sense.

Token consumption

When you allocate tokens, the total number of available tokens is reduced. If the token balance is zero or insufficient, you cannot allocate more to the access types. You can free up tokens and choose to use the tokens differently. The number of tokens for the Qlik Sense site can be increased or decreased by activating a new license. One token corresponds to one (1) user access pass or ten (10) login access passes.

You can adjust the token usage according to the usage need over time. The License Monitor app is useful when analyzing the distribution of user access passes and login access passes. A frequent user with a login access pass should perhaps have a user access pass instead? An infrequent user with a user access pass may only need login access.

For more information on access passes and the token consumption model, see: Managing licenses.

License Enabler File

Every Qlik Sense site needs a site license. The License Enabler File (LEF) defines the number of tokens available on your site, which you can manage from the central node.

You activate the license on the **Site license properties** page in the QMC, where you enter owner information, serial number, and control number. When you click **Apply**, the LEF is automatically downloaded. To preview the license before applying, open the **LEF access** tab and click **Get LEF and preview the license**. The license is displayed in the text box. This is also where you enter the LEF if you have received it in an email, or if you are unable to download it.

Adding the LEF makes you the root administrator for the Qlik Sense site. When you activate the license, you make the tokens available.

Qlik Sense Desktop

Qlik Sense Desktopis a free version of Qlik Sense that you can download on your computer, see www.qlik.com. No license is required, you only need a Qlik account to download Qlik Sense Desktop.

Before you can start using Qlik Sense Desktop, you need to authenticate yourself either with your Qlik account or against a Qlik Sense Enterprise server. You need to have a working network connection to enable authentication. With your Qlik account, you will also be able to access Qlik Sense Cloud.

What can you do in Qlik Sense Desktop?

Apps you create in Qlik Sense Desktop can be exported and used in Qlik Sense Enterprise, uploaded to Qlik Sense Cloud to be shared with others, as well as by other Qlik Sense Desktop installations.

For a comparison between Qlik Sense and Qlik Sense Desktop, see Comparing versions of Qlik Sense.

Qlik DataMarket

Qlik DataMarket offers a large variety of data from many of the most used global data sources. Qlik DataMarket provides weather and demographic data, as well as business, economic, and societal data. When you combine external data with your own, you get a complete view of the environment that your business is operating in.

Before you can use Qlik DataMarket data, you must accept the terms and conditions for its use. The option **Free** gives you access to a limited data set. The option **Licensed subscription** gives you access to premium data packages. You activate the license in the same way as for Qlik Sense Enterprise. On the Qlik DataMarket page, under **License and tokens**, you enter owner information, serial number, and control number. When you click **Apply**, the LEF is automatically downloaded. To preview the license before applying, open the **LEF access** tab and click **Get LEF and preview the license**. The license is displayed in the text box. This is also where you enter the LEF if you have received it in an email, or you for some reason cannot download it.

When you have applied the access credentials, the premium data is labeled **Licensed**. If you accept the terms and conditions but do not enter a license for any of the premium data packages, you have the option to buy a license later. The **Purchase** button replaces the **Premium** label, enabling you to buy a license.

After you have activated the Qlik DataMarket license the first time, you can change subscription type and update the license properties. To change to a **Free** subscription, you only need to select **Free** and click **Apply** on the Qlik DataMarket page. To change to a licensed subscription, fill in the license details as described previously in this section.

It is not necessary to accept Qlik DataMarket terms and conditions when using Qlik Sense Desktop. Access credentials are also not required because the premium data sets are not available on Qlik Sense Desktop.

Qlik NPrinting

Qlik NPrinting can be installed and configured to connect to QlikView documents or Qlik Sense apps. The licensing requirements and procedures are different depending on if you connect Qlik NPrinting to QlikView or Qlik Sense.

The first time you access a newly installed Qlik NPrinting server, you need to enter license key, control number, user name, and organization. This is the only Qlik NPrinting license that you need to activate. The licenses for Qlik NPrinting Designer and Qlik NPrinting Engine are automatically based on license information from the Qlik NPrinting server.

Qlik NPrinting versions 16.0.0.0 and later are licensed by a LEF. If you are using an earlier version of Qlik NPrinting, we suggest that you upgrade to Qlik NPrinting versions 16.0.0.0 or later.

A Qlik Sense token is not required for the Qlik NPrinting service account. However, because you often perform troubleshooting within the Qlik NPrinting service account, it is helpful to assign a token to the Qlik NPrinting service account so that it has access to the Qlik Sense hub.

For details regarding licensing Qlik NPrinting with QlikView, QlikView Desktop, and Qlik Sense, see Licensing Qlik NPrinting.

Qlik Sense Cloud

To be able to create and share apps with others using Qlik Sense Cloud, you need to create a Qlik account and log in. No license is required. With a Qlik account, you can also download Qlik Sense Desktop.

10 License Enabler File

With a token-based license (login access or user access), the Qlik Sense licensing is administered using a License Enabler File (LEF). The LEF holds the number of tokens available for the central node in a site. This means that a Qlik Sense site needs at least one (1) LEF.

The LEF can be downloaded when the serial number and the control number have been entered in the Qlik Management Console (QMC). The LEF can also be pasted directly into the QMC, if, for example, no network connection is available.

10.1 Increase in tokens

When the number of tokens in the LEF increases (for example, when buying additional tokens), the new tokens are added to the pool of unallocated tokens that can be used to allocate access passes that allow users to access Qlik Sense.

10.2 Decrease in tokens

When the number of tokens in the LEF decreases, the following happens:

1. Unallocated tokens are removed.
2. If step 1 is not enough to meet the decreased number of tokens in the LEF, any tokens that are freed up by removal of access passes cannot be used for new allocations until the number of allocated tokens is below the new number set in the LEF.

   Removing access passes (page 300)

ⓘ *A License Enabler File is also used for user-based licenses (analyzer and professional), but then no tokens are used, instead a defined number of access allocations.*

ⓘ *If you want to set up Qlik Cloud Services or Qlik Sense Enterprise for elastic deployments, please contact your Qlik representative or Qlik Support to obtain a valid license for the setup.*

11 Access passes

There are two major license types: one that is user-based and one that is token-based.

- User-based license: you can allocate professional access and analyzer access. The LEF determines the distribution of the two access types.
- Token-based license: you can allocate user access and login access. The LEF determines the number of tokens that you can allocate to the two access types.

In addition to these two types there is the analyzer capacity license that is similar to analyzer access regarding available features, but where consumption is time based (analyzer time).

User-based licenses

User-based licenses grant a predefined number of professional and analyzer access allocations. The distribution of the access types is determined by the LEF.

The following table lists the available access types for a user-based Qlik Sense license.

Access type	Description
Professional access	You allocate professional access to an identified user to allow the user to access streams and apps within a Qlik Sense site. The professional access is intended for users who need access to all features in a Qlik Sense installation. A user with professional access can create, edit, and publish sheets or apps, and make full use of the available features, including administration of a Qlik Sense site. There is a direct relationship between the access type (professional access) and the user. If you deallocate professional access from a user, the access type is put in quarantine, if it has been used within the last seven days. If it has not been used within the last seven days, the professional access is released immediately. You can reinstate quarantined professional access, to the same user, within seven days. The maximum number of parallel user connections for a single user of this type of access pass is five (5). When a user with the maximum number of parallel user connections ends a connection (for example, by logging out) five minutes must pass before the user can use the access pass to add another connection (for example, by logging in).

Access type	Description
Analyzer access	You allocate analyzer access to an identified user to allow the user to access streams and apps in the hub. The analyzer access is intended for users who consume sheets and apps created by others. A user with analyzer access cannot create, edit, or publish sheets or apps, but can create and publish stories, bookmarks and snapshots based on data in apps. The user can also create bookmarks, print objects, stories, and sheets, and export data from an object to Excel. There is a direct relationship between the access type (analyzer access) and the user. If you deallocate analyzer access from a user, the access type is put in quarantine, given that it has been used within the last seven days. If it has not been used within the last seven days, the analyzer access is released immediately. You can reinstate quarantined analyzer access, to the same user, within seven days. The maximum number of parallel user connections for a single user of this type of access pass is five (5). When a user with the maximum number of parallel user connections ends a connection (for example, by logging out) five minutes must pass before the user can use the access pass to add another connection (for example, by logging in).
Analyzer capacity	Analyzer capacity is a consumption-based license type that is similar to analyzer access regarding available features. Analyzer capacity is available to any users, including anonymous users, and they share the monthly analyzer time allotment, which is consumed in units of six minutes.

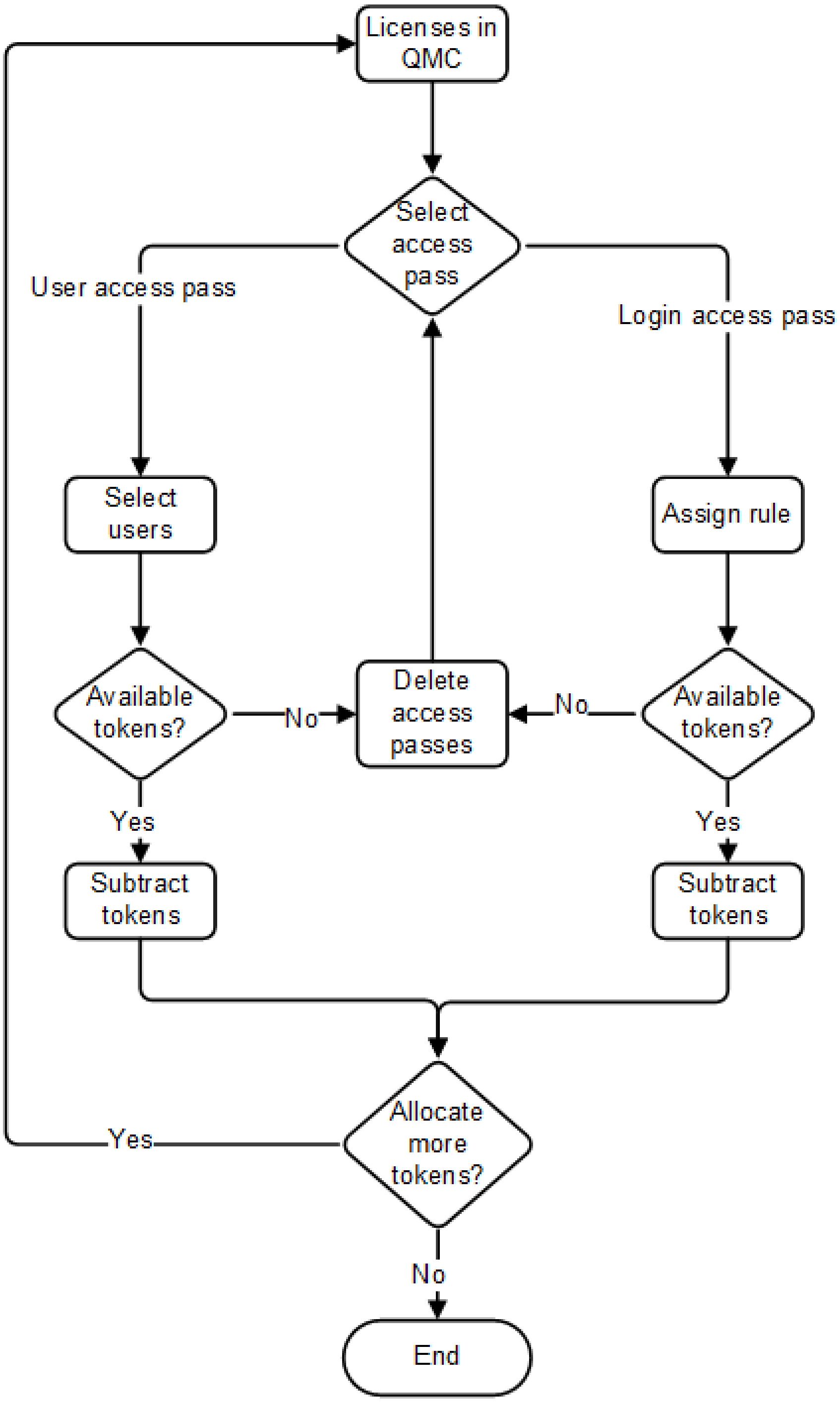
Token-based licenses

The following table lists the available access types for a token-based Qlik Sense license.

Access type	**Description**
User access pass	This type of access pass allows a unique and identified user to access the hub. The access pass is valid within an entire Qlik Sense site. For example, if a user first connects to a node in the USA and then, at a later stage, connects to a node in the UK, the user consumes the same access pass, if the two nodes are connected to the same central node. See: *Architecture (page 20)* The maximum number of parallel user connections for a single user of this type of access pass is five (5).When a user with the maximum number of parallel user connections ends a connection (for example, by logging out) five minutes must pass before the user can use the access pass to add another connection (for example, by logging in). One (1) token corresponds to one (1) access pass. The access passes are allocated using the Qlik Management Console (QMC).
Login access pass	This type of access pass allows an identified or anonymous user to access the hub for a maximum of 60 continuous minutes per 28-day period. If the user exceeds the 60 minutes time limitation, the user connection does not time out. Instead, another login access pass is used. If no more login access passes are available, the user connection is discontinued. • If an identified user is disconnected, the user can re-connect and continue to use the same access pass, if re-connecting within the 60 minutes. • If an anonymous user is disconnected, the user gets a new access pass when re-connecting. The login access pass tracks the number of logins and runs over 28 days. For example, if 1000 logins are assigned to Group A, the users in Group A can use 1000 logins over 28 days. If 100 logins are consumed on Day 1, the 100 logins are available again on Day 29. The maximum number of parallel user connections for a single user of this type of access pass is five (5). Note that this only applies to identified users. An anonymous user can only have one (1) user connection. When a user with the maximum number of parallel user connections ends a connection (for example, by logging out) five minutes must pass before the user can use the access pass to add another connection (for example, by logging in). However, a user can have more connections than allowed by a single access pass by consuming additional access passes. One (1) token corresponds to ten (10) access passes. The access passes are allocated using login access groups in the QMC. *App reloads will extend the session and consume access passes also when the app is not actively used. If a browser page is open with an app, app reloads will result in additional access pass consumption.*

11.1 Allocation of access passes

The following figure shows how the Qlik Management Console (QMC) is used to manage the allocation of token-based access passes.



11.2 Removing access passes

This page is only valid for token-based licenses.

This section describes how to free up tokens for new allocations of access passes by removing existing access passes in the Qlik Management Console (QMC).

User access pass

When a user access pass is removed, it enters a quarantine for seven (7) days, counting from the last time that the access pass was used. For example, if the access pass is used on January 10, the tokens used to allocate the access pass are not available for new allocations until January 18.

During the quarantine period, the original allocation of the access pass can be reinstated, which means that the quarantine period ends and the user can start using the access pass again.

Login access pass

When a login access group is removed, the tokens used to allocate the access pass become available in accordance to the following procedure:

1. For every ten (10) **unused** login access passes, one (1) token is freed up.
2. For every ten (10) login access passes that leave the **used** state after the period specified in *Login access pass (page 298)* has passed, one (1) token is freed up.

11.3 Multi-deployment sites

This section describes how the Qlik Sense licensing is handled within multi-deployment sites, where apps are promoted from a development site to a test site and finally to a production site.

Development site

In a Qlik Sense deployment that includes a development site and a production site, two (2) License Enabler Files (LEF) are needed (that is, one per site).

Each node within the development site is licensed with one (1) access pass type (for example, user access passes), if only disconnected users are expected.

Test site

The LEF for a test site mirrors the LEF for a development site.

See also:

🗋 *Installing Qlik Sense in a multi-node site (page 85)*

12 Troubleshooting - Deployment

There are several things you can do to troubleshoot and resolve problems before logging a case with product support. The general guidance here is designed to help you to understand the problem and know where to look for possible errors and potential solutions.

Before you call support:

- *Understand the problem (page 301)*
- *Use the log files (page 302)*
- Study the Qlik Sense Help.
- Read the troubleshooting topics in this section.

If you cannot find a solution in the product help, then follow the general guidance in this topic.

12.1 Understand the problem

Understanding the problem may help you to find a solution, and will enable you to provide Qlik support with the information needed to process your case more effectively. Ensure that you understand the problem and can describe it as fully as possible before seeking further support:

Questions to ask	**Answers** - that may lead to a better understanding of the problem
Who experienced the problem?	What type of users were affected, and how many? This can help you to determine if it is a global issue, a configuration problem, a component problem, or due to user configuration.
What happened after an action was performed?	Pay attention to any symptoms, behavior, and error messages. This can help you to identify which component is causing the error, and which log files to use.
When did the problem first occur?	When is it triggered, and what user action or system action causes it? For example, is it due to a trigger reload, or if a user clicks on an object in an app.
Have you experienced this issue before?	If yes, how often has the problem occurred?
Where did this issue occur first?	Describe where in the system or environment the problem occurs? Is it on the client side, server backend, or in a specific application. For example, does the end user have a direct connection to the Qlik Sense hub, or are they passing through a third-party reverse proxy before reaching the hub?
Why do you think it happened?	Gather the relevant log files. Compare log files that

include the problem with those that do not. For example, compare a successful reload with an unsuccessful reload of the same app. For log file locations, see the individual product help pages.

12.2 Use the log files

To troubleshoot and resolve issues effectively you need to understand how to use the log files. You also need to know when to use the default logs, and when to use the archived log files.

When you get an error message, the following steps can help you to identify which component has failed:

1. Read the error message carefully as it can tell you which component has failed.
2. Navigate to the default log files, or the archived logs folder for the failed component.
3. When you have navigated to the correct folder, search for errors in the log file to identify the issue.

Default log files

In Qlik Sense, the log files are by default stored in *C:\ProgramData\Qlik\Sense\log*. After 12 hours they are moved to the archived logs folder.

There is one **Log** folder per machine, and the following sub-folders for each component (engine, repository, proxy, and scheduler):

- Audit - High level user action logs. For example, open app, reload app, get ticket, and login success.
- System - Service logs including all errors, and system or service operations.
- Trace - Debug diagnostics. For example, user selections, https redirects, method work times, and session information.

If you are running a multi-node environment, ensure you are connected to the correct node.

Criteria for moving the default logs to the archived logs folder:

- On service restart or crash
- If the file is larger than a predefined size
- If the file is more than 12 hours old

Archived log files

The archived logs folder is located in the Qlik Sense file share that you created as part of the Qlik Sense installation. Use the archived log files if the problem occurred more than 12 hours ago.

To find the archived logs, open the QMC and go to **Service Cluster**, **Cluster Settings**, and you can see the path that you specified during installation.

Unlike the standard logs, the archived log files are stored in a central, shared location, so if you are running a multi-node environment, you will find one sub-folder per node. Like the default log files, the archived log files also contain Audit, System, and Trace sub-folders for each main component.

For more information on the location of the log files for each component, see: *Storage (page 254)*

In Qlik Sense Sept 2017 and later centralized logging replaces the synchronized persistence logging framework.

12.3 Qlik Sense client or application problems

If you get an error in a Qlik Sense application, the following questions can help you to narrow down the issue:

- Was the application working before the error occurred?
- Is the issue present in the Qlik Sense Desktop client?
- Is the issue specific to a browser, or is it present in all browsers?
- Does this issue affect a specific user, user group, or all users?
- Does this issue occur in one application, or every application?

12.4 Other resources

Once you have gathered all the information you need, use the following links to research other possible solutions:

- Knowledge base
- Community website
- Log a case with product support.

12.5 Cannot find the repository database superuser password

Possible cause

In your Qlik Sense installation, you cannot find your repository database superuser password.

Proposed action

You can find the repository database superuser password using the **Connection String Editor** which is included in the Qlik Sense diagnostic tools.

To open the **Connection String Editor**:

1. Navigate to *C:\Program Files\Qlik\Sense\Repository\Util\QlikSenseUtil* and double-click the *QlikSenseUtil.exe* file.
2. In the **LogOnForm** screen, enter the database user and password that you used during the Qlik Sense installation.
3. In the **Diagnostics Tool**, click the **Connection String Editor** tab.
4. In the **Connection String Editor** click **Read** to see the encrypted connection string.

12.6 Cannot access the hub or the QMC after installation

Possible cause

After you have installed Qlik Sense, the services are started automatically, which can take a few minutes. You cannot access the Hub or the Qlik Management Console until the services have started up correctly.

Proposed action

Check that the services have started and that the appropriate ports are available.

Do the following:

1. In Windows, open the Task Manager and check that all Qlik Sense services have started.
2. Check that the ports needed by Qlik Sense are available.

   Plan and deploy Qlik Sense

12.7 One or more Qlik Sense services did not start after installation

Possible cause

The Qlik Sense Repository Service (QRS) cannot start if there is no repository database, and if the QRS is not running, none of the other Qlik Sense services can start.

After installation the services are started automatically, which can take a few minutes. This means there may be a short delay before all services have started correctly.

Proposed action

Restart the service, check the user account, restart the server, or check the logs.

Do the following:

1. Stop the service and start it again.

   You can also try changing the **Start Type** of the failing service from **Automatic** to **Automatic (Delayed Start)** in the Task Manager in Windows.
2. Check that the user that runs the Qlik Sense services is member of the Local Administrators group.

   If you are using a domain administrator account, check that there is no problem related to the User Account Control (UAC).
3. Restart the server on which Qlik Sense is running.
4. Check the log files for the service to see if there is any information regarding why the service has not started.

The log files are available in the *%ProgramData%\Qlik\Sense\Log\<Service>* folder.

Set the ServicesPipeTimeout setting in the Registry Editor in Windows to 120000 milliseconds (that is, two minutes). This is needed to give the Qlik Sense Repository Service (QRS) enough time to start.

Microsoft Knowledge Base: 884495

> ⚠ *Serious problems might occur if you modify the registry incorrectly by using the Registry Editor or by using another method. Make sure that you can recover if the changes lead to problems.*

5. If the steps in this topic do not solve the problem, uninstall and reinstall Qlik Sense.

12.8 Anti-virus software scanning affects performance

Possible cause

Anti-virus software scanning can have an effect on the performance of Qlik Sense.

Proposed action

Configure the anti-virus software scanning so that it does not interfere with Qlik Sense. Make sure that regular scans and live/real-time scans are turned off for the following locations:

- *%ProgramData%\Qlik*
- Any additional folder path configured for storing QVF files
- All executables under *%ProgramFiles%\Qlik\Sense*
  - *Engine\Engine.exe*
  - *Engine\QVConnect32.exe*
  - *Engine\QVConnect64.exe*
  - *Logging\Qlik.Logging.Service.exe*
  - *MigrationService\MigrationService.exe*
  - *Printing\Printing.exe*
  - *Printing\Qlik.Printing.CefSharp.exe*
  - *Proxy\Proxy.exe*
  - *Repository\Repository.exe*
  - *Repository\PostgreSQL\9.6\bin\postgres.exe*
  - *Scheduler\Scheduler.exe*
  - *ServiceDispatcher\ServiceDispatcher.exe*
  - *ServiceDispatcher\Node\node.exe*

12.9 Exit codes

Exit codes can be particularly useful when using the silent mode operations. The exit code can be viewed in the command prompt window by using the following command:

Echo %errorlevel%

The following table contains a complete list of the exit codes.

Code	Description
0	Success
-1	General fatal error
-2	Command line parsing error
-3	Not implemented error
-4	Downgrade
-5	Malformed bundle XML
-6	Install condition not met
-7	Unknown upgrade scenario
-8	Pending reboot must be applied first
-9	Patch run with no baseline installed
-10	Disallowed setup process running
-11	Unsupported minor upgrade error
-12	Invalid policy
-13	User validation failed
-14	Database superuser password validation error
-15	Not supported error
-16	Host name from certificate retrieval error
-17	Inconsistent upgrade
-18	General silent workflow error
-19	OS bitness not supported
-20	OS too old
-21	OS type not supported
-22	Patch is superseded
-23	General MSI Error
-24	Disabled services exist

-1335	CAB is corrupt
-1601	Disk space
-1602	User exit
-1923	Cannot install service
-7777	Unknown dark process exception

12.10 Rim node loses connection to the central node

Possible cause

The Windows setting **"System cryptography: Force strong key protection for user keys stored on the computer"** is enabled. This setting is not supported by Qlik Sense.

Proposed action

Disable **"System cryptography: Force strong key protection for user keys stored on the computer"**.

12.11 Repository cannot connect to database after installation

The installation was successful, but when the repository service is started it fails to connect to the database.

Possible cause

You used a database username and/or password that contains characters from mixed character sets.

Proposed action

1. Uninstall Qlik Sense and select **Remove Qlik Sense certificates and data folders** at the end of the installation.
2. Reinstall using a database username and password with characters from the same character set.

12.12 Unable to upgrade, reinstall or add a rim node due to password validation failure

Possible cause

When you install Qlik Sense with the setup program and choose to install a local database, you also create a database user (*qliksenserepository*) and a password. If you previously installed Qlik Sense with synchronized persistence then the database user will have a randomly generated password.

When you upgrade, reinstall, or add a rim node to your installation you to need to use this password again. If you did not create a super user password when you installed PostgreSQL or cannot remember the database user password, then you cannot continue to upgrade, reinstall, or add a rim node unless you change this password.

Proposed action

Use the command prompt to change the PostgreSQL database user password.

Do the following:

Change the client authentication settings to trust so you can change the password.

To do this:

1. In **Services**, stop the Qlik Sense Repository Database service, if it is running.
2. In PostgreSQL, change the authentication mode in the configuration settings to allow the password to be changed. To do this, navigate to *ProgramData\Qlik\Sense\Repository\PostgreSQL\<database version>* and open the *pg_hba.conf* file in a text editor.
3. Change the PostgreSQL client authentication method from `md5` to `trust`.

   > ⓘ *The client authentication settings are case sensitive.*

4. Save your changes.
5. Start the Qlik Sense Repository Database service.

To change the password open a command prompt and do the following:

1. Enter the following commands:
   a. To navigate to your repository database installation:
   ```
   cd C:\Program Files\Qlik\Sense\Repository\PostgreSQL\9.6\bin
   ```
   b. To connect to the database:
   ```
   psql.exe -h 127.0.0.1 -p 4432 -U postgres
   ```
   c. To set your new user password:
   ```
   ALTER USER qliksenserepository WITH PASSWORD '<newpassword>';
   ```
   This is either *qliksenserepository* or the user you set manually during the first installation of PostgreSQL. `ALTER ROLE` is displayed after successfully changing the password.
2. Stop the Qlik Sense Repository Database service.
3. Revert the `pg_hba.conf` authentication mode method back to `md5`.
4. Start the Qlik Sense Repository Database service.

Update the connection string for the Qlik Sense Repository Database using the **Connection String Editor** which is included in the Qlik Sense diagnostic tools.
To do this:

1. In your Qlik Sense installation, to open the **Connection String Editor**, navigate to *C:\Program Files\Qlik\Sense\Repository\Util\QlikSenseUtil* and double-click the *QlikSenseUtil.exe* file.
2. In the **LogOnForm** screen, enter the database user and password that you used during the Qlik Sense installation.
3. In the **Diagnostics Tool**, click the **Connection String Editor** tab.
4. In the **Connection String Editor**, click **Read** to see the encrypted connection string.

5. Update the connection string credentials with `name="QSR"` with your new repository database password.
6. Click **Save value above in config file encrypted** to save your changes.
7. Start the Qlik Sense Repository Database service.

You can now continue to upgrade, reinstall, or add a rim node to your Qlik Sense installation.

12.13 The database is unavailable, how do I find the Qlik logging service files

Possible cause

The database is temporarily unavailable.

Proposed action

1. Check that all the Qlik Sense services are running. If the Qlik Sense Repository Database service has stopped, then try to restart it.
2. Check the log files for possible errors. The repository logs will indicate if there are any start-up problems. In Qlik Sense, navigate to *C:\ProgramData\Qlik\Sense\Log* to view the log files.

12.14 Troubleshooting - database not configured for IP address or range

If you find the following error message in the installation logs: "psql: FATAL: no pg_hba.conf entry for host [ipv4 or ipv6]", it means the database needs to be configured.

Possible cause

The database is not configured to allow connection from that IP address or range.

Proposed action

Add an IP address or range in the Shared Persistence configuration file, see *Shared persistence configuration file syntax (page 108)*, or in the installation UI, see step 9 in *Installing Qlik Sense (page 86)*

12.15 Troubleshooting app distribution in multi-cloud

There is more than one possible cause when app distribution fails in a multi-cloud environment. You could encounter problems on the QSEfW side (with custom properties), at the IdP (with names and groups), during the actual distribution, and after distribution (with apps not being displayed).

Publishing is a little slow

Possible cause

You have published an app, and when checking the collection, the app is not present.

Proposed action

Allow some time to pass before troubleshooting why an app does not appear in a collection, publishing is not instantaneous.

Custom properties not in lowercase

You have created distribution policies and published an app to a multi-cloud collection. The app does not show up and no error message is displayed.

Possible cause

The custom properties you use in distribution policies are not in lowercase.

Proposed action

Use only lowercase letters for custom properties in distribution policies.

A temporary error occurred

Possible cause

A temporary error occurred.

Proposed action

Restart the Qlik Sense Service Dispatcher.

Do the following:

1. In Windows, open **Services**.
2. Scroll down and right-click the Qlik Sense Service Dispatcher. Select **Restart**.

An unknown error occurred

Possible cause

An error occurred and you do not know why.

Proposed action

Investigate the log files for multi-cloud services, for example, the App Distribution Service and Hybrid Deployment Service, see Multi-cloud services.

12.16 The logging database has grown too big

Possible cause

The size of the logging database can grow so much that it needs to be reduced in size.

Proposed action

Choose one of the following alternatives to reduce the size of the logging database:

- Decrease the archive and the purge time frame. Run the following command:
  `Qlik.Logging.Service.exe update --archive_age 15 --purge_age 30`.
- Set the maximum database size. Run the following command:
  `Qlik.Logging.Service.exe update --maximum_db_size_in_gb n`.
  Where *n* is a positive integer.

  > *If n is a value less than two (2), the enforcement functionality is disabled. If n is equal to or greater than two (2), the functionality is enabled, allowing the Logging service to delete entries from the database once the maximum size specified is exceeded. Please note that this process is inexact and therefore it is not possible for the Logging service to enforce the maximum database size precisely.*

- Manually purge the database.
- Turn off database logging. Run the following command:
  `Qlik.Logging.Service.exe update --database_logging off`.

12.17 Cannot read or write to the logging database

You have installed Qlik Sense successfully, but you cannot connect to the logging database.

Possible cause

You used a password that contains characters from mixed character sets. The log writer and log reader password cannot handle all mixed characters.

Proposed action

1. Uninstall Qlik Sense and select **Remove Qlik Sense certificates and data folders** at the end of the installation.
2. Reinstall using a password with characters from the same character set .

12.18 How can I debug if there are log entries missing in the database?

Possible cause

Some log messages are missing in the database.

Proposed action

Turn on error logging in the *QlikCentralizedLogging.config* file to enable all log file messages to be collected.

Do the following:

1. Go to *C:\ProgramData\Qlik\Sense\Log\QlikCentralizedLogging.config*.
2. Change the value of the following line to "*True*":
   `<"DllErrorLoggingEnabled" value="False" />`
3. Restart the Qlik Logging Service.
4. Go to *C:\ProgramData\Qlik\Sense\Log*.
5. Open the file named *<hostname>.Qlik.Sense.Logging.DLL.Errors*.
6. Monitor this file for errors generated due to Qlik Logging Service issues.

12.19 How can I manage storage to fit our needs and the needs of the operational IT department?

Possible cause

Storage of log data has become a concern.

Proposed action

There is more than one possible solution to this problem.

Do the following:

a. Make use of the archive and purge functionality by adjusting the rate at which both events occur against the logging database only. File logging does not provide file management controls.
   1. Open a command prompt with administrator privileges.
   2. Go to *C:\Program Files\Qlik\Sense\Logging*.
   3. Run `Qlik.Logging.Service.exe update --archive_age X` (where X is the number of days).
   4. Run `Qlik.Logging.Service.exe update --purge_age X` (where X is the number of days).
   5. Restart the Qlik Logging Service.

b. Set a maximum database size so that the Qlik Logging Service can automatically trim older rows out of the database to maintain overhead.
   1. Open a command prompt with administrator privileges.
   2. Go to *C:\Program Files\Qlik\Sense\Logging*.
   3. Run `Qlik.Logging.Service.exe update --maximum_db_size_in_gb X` (where X is the size in GB).
   4. Restart the Qlik Logging Service.

12.20 Qlik logging service database urgently needs to be reduced in size

Possible cause

The database admin has identified an immediate need to clear space.

Proposed action

The database admin can use the manual purge option offered through the Qlik Logging Service.

Do the following:

1. Open a command prompt with administrator privileges.
2. Go to *C:\Program Files\Qlik\Sense\Logging*.
3. Run `Qlik.Logging.Service.exe archive --cutoff X --hours` (where `X` is the number of hours).
4. Run `Qlik.Logging.Service.exe purge --cutoff X --hours` (where `X` is the number of hours).
5. If presented with a message stating the system is too busy, retry the command.
6. If the second try generates the same message, try again at a less busy time on the server.
7. Restart the Qlik Logging Service.

12.21 Logging issues when trying clean up the database

Possible cause

The size of the Qlogs database has grown too large and the database admin is having resource issues when trying to clean up the database.

Proposed action

Shut off database logging to clean up the storage problem.

> ⚠ *This solution should only be used as a last resort. All purges are permanent.*

Do the following:

1. In Windows, open **Services**.
2. Stop the following services:
   - Qlik Sense Engine Service
   - Qlik Sense Proxy Service
   - Qlik Sense Scheduler Service
   - Qlik Sense Repository Service
3. Open a command prompt with administrator privileges.
4. Go to *C:\Program Files\Qlik\Sense\Logging*.
5. Run `Qlik.Logging.Service.exe archive --cutoff_in_hours X` (where `X` is the number of hours).
6. Run `Qlik.Logging.Service.exe purge --cutoff_in_hours X` (where `X` is the number of hours).
7. Restart the Qlik Logging Service.
8. Verify the reclaimed data storage.
9. Start all stopped Qlik Sense services, beginning with the Qlik Sense Repository Service.

12.22 Upgrade fails with message "Qlik Sense Superuser password validation failure"

When upgrading Qlik Sense 3.2 or earlier to June 2017 or later, the installation fails and you get the following error message: "Qlik Sense Superuser password validation failure". Despite using the correct password, you get the same error every time you attempt the upgrade.

Possible cause

The upgrade failed because you entered an incorrect superuser or repository password during the first upgrade attempt.

Although you inserted an incorrect password, you were still able to create the PostgreSQL 9.6 version of the database, and the wrong password was registered in the settings. Therefore, later upgrade attempts will fail because the passwords in PostgreSQL 9.6 no longer match.

Proposed action

Delete the *c:\ProgramData\Qlik\Sense\Repository\PostgreSQL\9.6* folder and try running the upgrade procedure again. Make sure you enter the correct password.

12.23 Failed to remove soft deleted records

When upgrading Qlik Sense to November 2017 or later, the installation fails and you get the following error message: "Failed to remove soft deleted records. An exception was thrown while invoking the constructor 'Void .ctor()' on type 'DatabaseContext'".

Possible cause

The database contains soft deleted records that generate an error when upgrading to a version of Qlik Sense without soft deletes, that is, November 2017 or later.

Proposed action

Run a script to delete the soft deleted records.

> *VERY IMPORTANT! Back up the whole QRS database before executing the script. If an error occurs, restore the backup, find out the data discrepancy, fix the issue and execute again, see Backup and restore Qlik Sense (page 182).*

Do the following:

1. Stop all the services, except the Qlik Sense Repository Database.
2. Save the script below to a file as *recurse_cleanup.sql*.
3. Move the file *recurse_cleanup.sql* to *%ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\<database version>\bin*.

4. Open a command prompt with elevated privileges.
5. Navigate to *%ProgramFiles%\Qlik\Sense\Repository\PostgreSQL\<database version>\bin*, for example: `cd C:\"Program Files"\Qlik\Sense\Repository\PostgreSQL\9.6\bin`.

   > *If you installed PostgreSQL manually, the location where to place and run the script from will be: %ProgramFiles%\PostgreSQL\<database version>\bin.*

6. Run `.\psql.exe -h localhost -d QSR -U postgres -p 4432 -a -f recurse_cleanup.sql`
7. If prompted, enter database superuser password.
8. Restart Qlik Sense Service Dispatcher, then start the Qlik Sense Repository Service in the given order.

> *When running the script on a non-English OS, you may encounter errors during the script execution. The errors can be caused by the character set conversion between server (PostgreSQL) and client (Powershell). To enable automatic character set conversion, run the following command from the command prompt before opening* ***Powershell*** *and executing the script:* `SET PGCLIENTENCODING=UTF-8`*. The variable is lost the moment the command prompt is closed. For more information refer to* [Character Set Support](#).

Script for deleting soft deleted records in the Qlik Sense Repository Database

```
/*
####################################################################################################
##########################
   Script Name: Recurse cleanup
   Description: the script is intended to delete all entities marked as soft deleted in the QRS
database
   Caution: BACKUP the whole QRS database before executing the script!


####################################################################################################
##########################
 */

/* Step 1. Update records according to QRS special logics

####################################################################################################
##########################
 */
  -- Step 1.1 Update Owner to sa_repository if Owner is deleted

       -- Step 1.1.1 Get all Qlik Sense Tables
       CREATE OR REPLACE FUNCTION get_all_sense_tables() RETURNS SETOF information_schema.tables AS
       $BODY$
       BEGIN
           RETURN QUERY SELECT *
                 FROM information_schema.tables
                WHERE table_schema='public'
                  AND table_type='BASE TABLE'
                  AND table_catalog='QSR'
                  AND table_name <> '__MigrationHistory';
```

```
            RETURN;
        END
        $BODY$
        LANGUAGE plpgsql;

        -- Step 1.1.2 Filter Qlik Sense Tables with name of column
        CREATE OR REPLACE FUNCTION get_tables(columnname varchar)
        RETURNS SETOF information_schema.columns AS $$
        BEGIN
            RETURN QUERY SELECT DISTINCT * FROM information_schema.columns as isc WHERE isc.column_name =
columnname And isc.table_name IN (SELECT ts.table_name FROM get_all_sense_tables() as ts);
            RETURN;
        END
        $$
        LANGUAGE plpgsql;

        -- Step 1.1.3 Change ownership of soft deleted users to sa_repository
        CREATE OR REPLACE FUNCTION fix_orphan_owners() RETURNS void AS
        $BODY$
        DECLARE username character varying;
        DECLARE
            tables CURSOR FOR
                SELECT * FROM get_tables('Owner_ID');

        BEGIN
            SELECT E'\'sa_repository\'' INTO username;
            FOR table_record IN tables LOOP
                EXECUTE 'UPDATE "' || table_record.table_name || '" SET "Owner_ID" = (SELECT "ID" FROM "Users"
WHERE "UserId" = ' || username || ') WHERE "Owner_ID" IN (SELECT "ID" FROM "Users" WHERE "Deleted" =
true)';
            END LOOP;
        END
        $BODY$
        LANGUAGE 'plpgsql';

        SELECT * FROM fix_orphan_owners();

        -- Step 1.1.4 Remove created DB functions for fixing ownership relations
        DROP FUNCTION fix_orphan_owners();
        DROP FUNCTION get_tables(columnname varchar);
        DROP FUNCTION get_all_sense_tables();

  -- Step 1.2 Unpublish App if Steam is deleted
  UPDATE "Apps"
     SET "Stream_ID" = null, "Published" = false
   WHERE "Stream_ID" IN (SELECT "ID" FROM "Streams" where "Deleted" = true);

   UPDATE "AppObjects"
     SET "Approved" = false, "Published" = false
   WHERE "App_ID" IN (SELECT "ID" FROM "Apps" where "Published" = false);

/* Step 2. Prepare for deletion: Alter foreign keys to Casacade Delete

####################################################################################################
##########################
 */
 CREATE TABLE temp_foreign_key (
    constraint_name VARCHAR,
    table_name VARCHAR,
    column_name VARCHAR,
    ref_table_name VARCHAR,
    ref_column_name VARCHAR
  );

  INSERT INTO temp_foreign_key (constraint_name, table_name, column_name, ref_table_name, ref_column_
name)
  SELECT fk.constraint_name, child.table_name, child.column_name, parent.table_name, parent.column_
name
    FROM information_schema.referential_constraints fk
         JOIN information_schema.key_column_usage AS child ON fk.constraint_name = child.constraint_
name
         JOIN information_schema.key_column_usage AS parent ON fk.unique_constraint_name =
parent.constraint_name
   WHERE fk.constraint_schema = 'public'
     AND child.position_in_unique_constraint = parent.ordinal_position
     AND fk.delete_rule = 'NO ACTION';

  -- Step 2.2 Create a function the replace foreign keys with new on DELETE option
  CREATE OR REPLACE FUNCTION replace_foreign_key (new_option VARCHAR) RETURNS void AS
  $BODY$
  DECLARE
    fks CURSOR FOR
        SELECT * FROM temp_foreign_key;
  BEGIN
    FOR rec IN fks LOOP
      EXECUTE 'alter table "' || rec.table_name || '" '
          || 'drop constraint "' || rec.constraint_name || '" ,'
          || 'add constraint "' || rec.constraint_name || '" FOREIGN KEY ("' || rec.column_name || '")
REFERENCES "' || rec.ref_table_name || '" ("' || rec.ref_column_name || '") ' || new_option || ';' ;
    END LOOP;
  END;
  $BODY$
  LANGUAGE plpgsql;

  -- Step 2.3 execute the function to replace all foreign keys with CASCADE on Delete
  SELECT *
    FROM replace_foreign_key('on delete cascade');

/* Step 3. Delete entities marked as Soft Deleted

####################################################################################################
#########################
 */
  -- 3.1 Create a function to delete all SoftDeleted records
  CREATE OR REPLACE FUNCTION delete_softdeleted_records(keep_for_days int) RETURNS void AS
  $BODY$
  DECLARE
    entity_tables CURSOR FOR
           SELECT table_name
             FROM information_schema.columns
            WHERE table_schema='public'
```

```
              AND column_name='Deleted';
  BEGIN
    FOR tbl IN entity_tables LOOP
        EXECUTE 'delete from "' || tbl.table_name || '" where "Deleted" = true and "ModifiedDate" <=
CURRENT_DATE - ' || keep_for_days || ';';
    END LOOP;
  END;
  $BODY$
  LANGUAGE plpgsql;

  -- Step 3.2 execute the function to delete entities
  SELECT *
    FROM delete_softdeleted_records(3);

/* Step 4. Resume foreign keys to No Action on Delete

####################################################################################################
##########################
 */
  SELECT *
    FROM replace_foreign_key('');

/* Step 5. Drop temp objects

####################################################################################################
##########################
 */
DROP FUNCTION delete_softdeleted_records(keep_for_days int);
DROP FUNCTION replace_foreign_key(new_option varchar);
DROP TABLE temp_foreign_key;
```

12.24 The Qlik Sense Mobile app encounters a network error and must close

Possible cause

If your Qlik Sense Mobile app was deployed using the VMware Tunnel for per-app VPN security, and the per-app VPN is later disabled in the iOS **Settings**, the following error will appear the next time the Qlik Sense Mobile app is launched:

The Qlik Sense Mobile app has encountered a network error and must stop. Restart the mobile app.

Proposed action

Ensure that the VMware Tunnel is enabled on your device.

Do the following:

1. On your iOS device, go to **Settings > VPN > VMware Tunnel > Connect On Demand** and toggle it on.