Module 2:
Security During and After Installation

Contents

Module Overview 1

Lesson 1: Security Steps during Installation 2

Service Accounts 3

Types of Authentication 6

Administrator Account 8

Practice: Configuring SQL Server Service Account 10

Lesson 2: Security Steps after Installation............................... 11

Using SQL Server Configuration Manager......................................12

Working with Windows Firewall 17

Connecting to a SQL Server Instance 19

Practice: Configuring Access to SQL Server.....................................21

Lesson 3: SQL Server Resources Consumers 22

Types of SQL Server Consumers...23

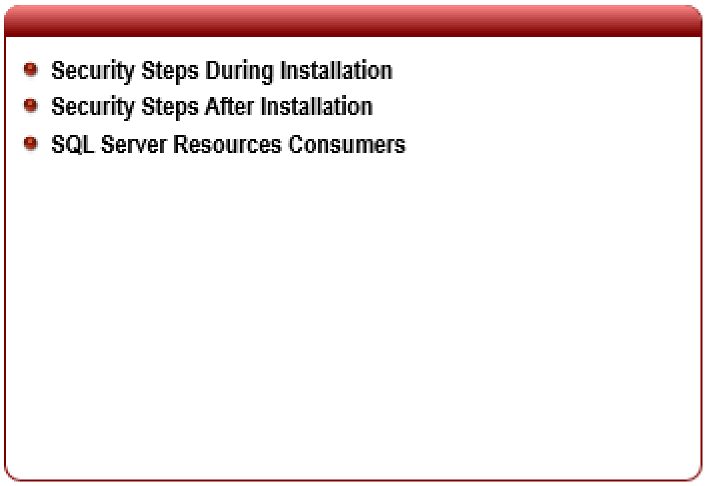
Password Issues...24

Password Policy..25

Practice: Configuring Password Policy 27

Summary 29

Module Overview



- Security Steps During Installation
- Security Steps After Installation
- SQL Server Resources Consumers

SQL Server is a software product where mistakes in installation procedure can seriously compromise the security of an entire server and all databases. There is a couple of places in the installation procedure where users need to know exactly what to do.

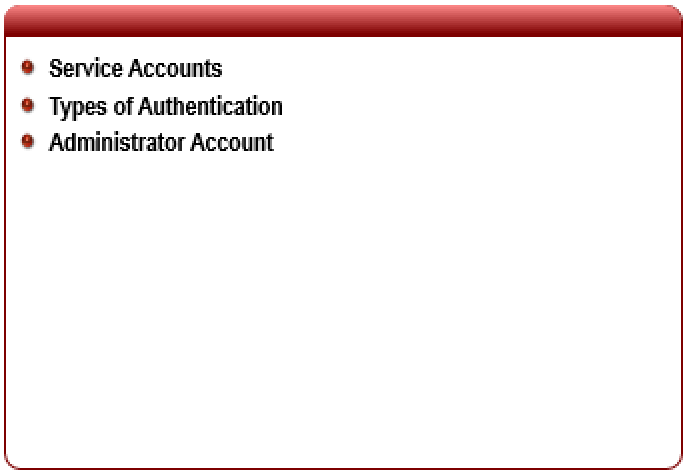
Also, SQL Server is secure by default and some security steps are required after installation so that users can access SQL Server resources.

Objectives

After completing this module, you will be able to:

- Recognize security steps during installation
- Recognize security steps after installation
- Identify types of SQL Server users

Lesson 1: Security Steps during Installation


- Service Accounts
- Types of Authentication
- Administrator Account

The SQL Server installation procedure is straight-forward. You can do it through a GUI or console, depending on your business needs. However, it is important that you know what you are doing throughout the installation.

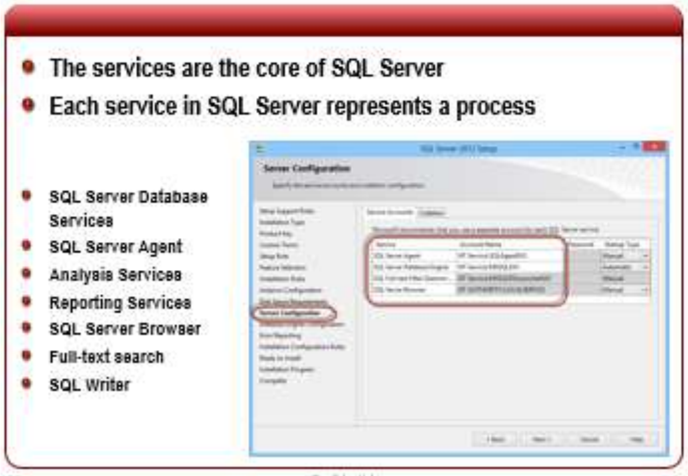
From a security perspective, installation is the place where you put a security foundation on your SQL Server instance. The parameters can be changed later on, but in most cases that means changing the production system which, in turn, affects performance and overall user experience of your system.

Objectives

After completing this lesson, you will be able to:

- Understand what are the service accounts
- Determine types of authentication
- Understand the risk of administrator accounts

Service Accounts



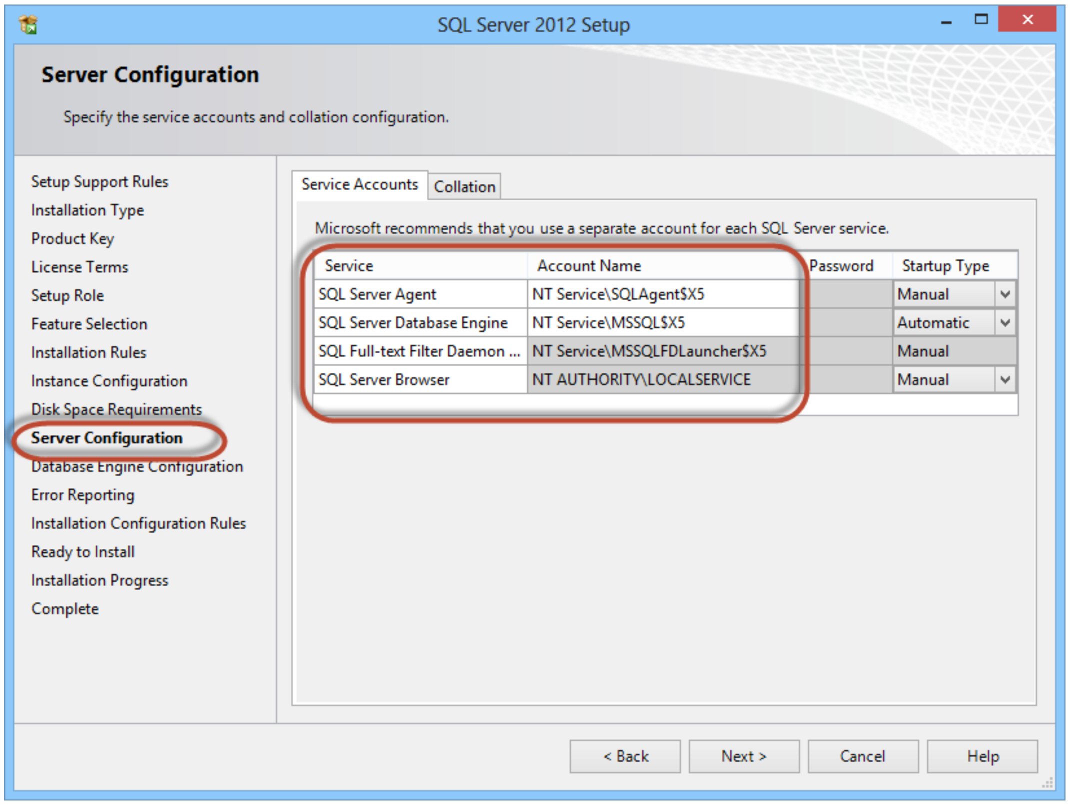
- The services are the core of SQL Server
- Each service in SQL Server represents a process

- SQL Server Database Services
- SQL Server Agent
- Analysis Services
- Reporting Services
- SQL Server Browser
- Full-text search
- SQL Writer

SQL Server is a classic service-based environment. That means that SQL Server does not need a GUI for itself. The services (i.e. the application that is run as a service in a Windows operating system, and the services provided for their clients (other services, GUI, application, end users, etc.)), are the core of SQL Server. Each service in SQL Server represents a process, or a set of processes, to manage the authentication of SQL Server operations with Windows:

- SQL Server Database Services - The service for the SQL Server relational Database Engine.
- SQL Server Agent - Executes jobs, monitors SQL Server, sends alerts, and enables automation of some administrative tasks.
- Analysis Services - Provides Online Analytical Processing (OLAP) and data mining functionality for business intelligence applications.
- Reporting Services - Manages, executes, creates, schedules, and delivers reports.
- Integration Services - Provides management support for Integration Services packages, storage, and execution.
- SQL Server Browser - The name resolution service that provides SQL Server connection information for client computers.
- Full-text Search - Quickly creates full-text indexes on content and properties of structured and semi-structured data to provide document filtering and word-breaking for SQL Server.
- SQL Writer - Allows backup and restores applications to operate in the Volume Shadow Copy Service (VSS) framework.
- SQL Server Distributed Replay Controller - Provides trace replay orchestration across multiple Distributed Replay client computers.
- SQL Server Distributed Replay Client - One or more Distributed Replay client computers that work together with a Distributed Replay Controller to simulate concurrent workloads against an instance of the SQL Server Database Engine.

These services have a corresponding privilege for run time under an operating system. Privileges are granted through regular Windows user accounts. Domain user or local accounts can also be used. Depending on our needs, resources, and security policy, we can choose between the two accounts. This procedure can be done during the installation procedure or thereafter.



Server Configuration

Specify the service accounts and collation configuration.

- Setup Support Rules
- Installation Type
- Product Key
- License Terms
- Setup Role
- Feature Selection
- Installation Rules
- Instance Configuration
- Disk Space Requirements
- **Server Configuration**
- Database Engine Configuration
- Error Reporting
- Installation Configuration Rules
- Ready to Install
- Installation Progress
- Complete

Service Accounts | Collation

Microsoft recommends that you use a separate account for each SQL Server service.

Service	Account Name	Password	Startup Type
SQL Server Agent	NT Service\SQLAgent$X5		Manual
SQL Server Database Engine	NT Service\MSSQL$X5		Automatic
SQL Full-text Filter Daemon ...	NT Service\MSSQLFDLauncher$X5		Manual
SQL Server Browser	NT AUTHORITY\LOCALSERVICE		Manual

The setup procedure for the "Server configuration" step, is a key point of this lesson. Each service (SQL Agent, SQL Server Database Engine, etc.) has a corresponding startup Account Name under which the permission service is running.

Startup accounts used to start and run SQL Server can be domain user accounts, local user accounts, managed service accounts, virtual accounts, or built-in system accounts. To start and run, each service in SQL Server must have a startup account configured during installation.

When resources external to the SQL Server computer are needed, Microsoft recommends using a Managed Service Account (MSA), configured with the minimum privileges necessary.

From Windows 7 and Windows Server 2008 R2, we have two new types of service accounts called MSAs and Virtual Accounts. MSAs and Virtual Accounts are designed to provide crucial applications, such as SQL Server with the isolation of its own accounts.

Managed Service Accounts

A MSA is a type of domain account created and managed by the domain controller. It is assigned to a single member computer to run a service. The password is managed automatically by the domain controller. You cannot use a MSA to log into a computer, but a computer can use a MSA to start a Windows service. A MSA is named with a $ suffix, for example DOMAIN\ACCOUNTNAME$.

Virtual Accounts

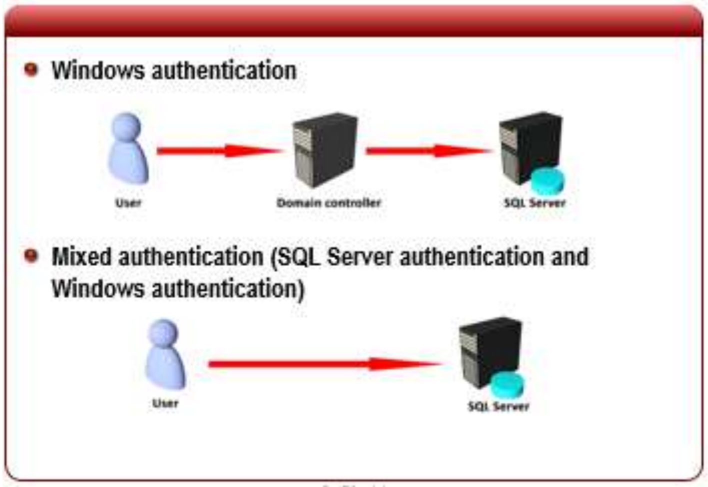
Virtual Accounts are managed local accounts that provide the following features to simplify service administration (see table below). A Virtual Account is auto-managed and can access the network in a domain environment. A Virtual Account using the instance name as the service name is used in the following format: NT SERVICE\<SERVICENAME>.

The following table lists examples of virtual account names.

Service	**Virtual Account Name**
Default instance of the Database Engine service	**NT SERVICE\MSSQLSERVER**
Named instance of a Database Engine service named **X5**	**NT SERVICE\MSSQL$X5**
SQL Server Agent service on the default instance of SQL Server	**NT SERVICE\SQLSERVERAGENT**
SQL Server Agent service on an instance of SQL Server named **X5**	**NT SERVICE\SQLAGENT$X5**

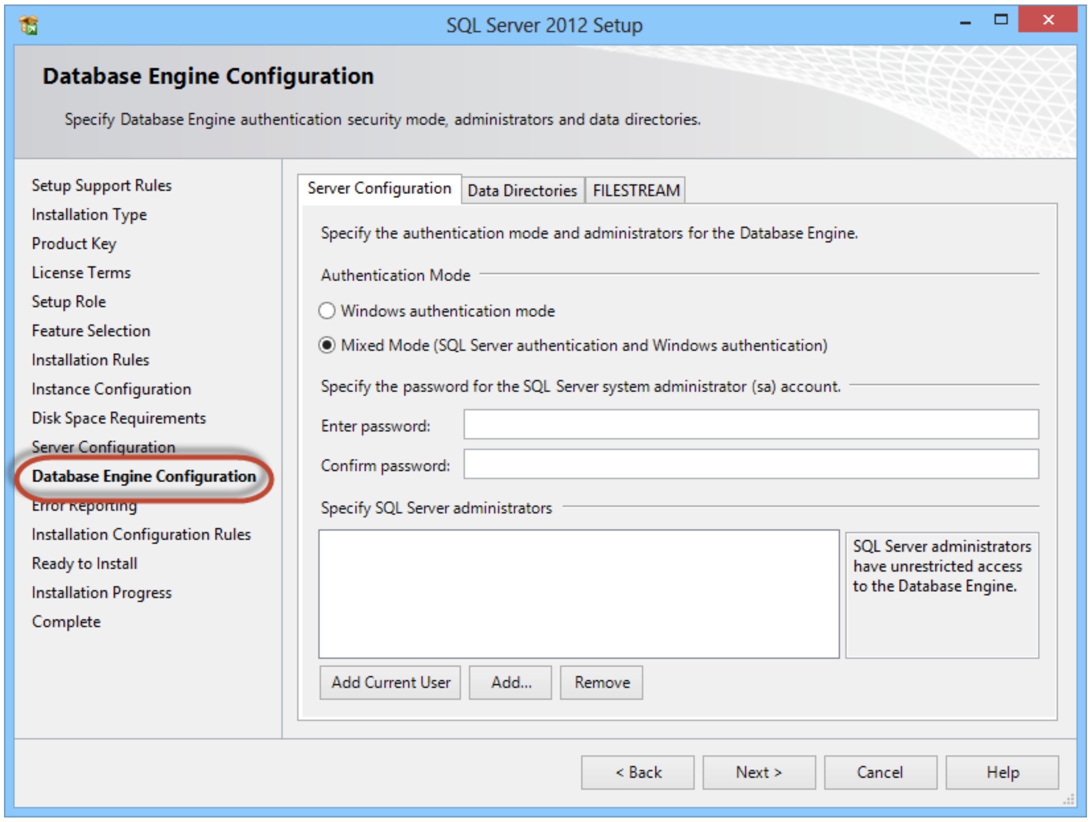
Note: It is recommended to use MSA instead of the classic approach of domain and built-in accounts.

Types of Authentication


- Windows authentication

- Mixed authentication (SQL Server authentication and Windows authentication)

In module 01, “Overview of SQL Server Security”, we discussed SQL Server security model, specifically authentication. Authentication is a second security “pit stop” during the installation procedure.


SQL Server 2012 Setup

Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Setup Support Rules
Installation Type
Product Key
License Terms
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete

Server Configuration | Data Directories | FILESTREAM

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

○ Windows authentication mode

● Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password:

Confirm password:

Specify SQL Server administrators

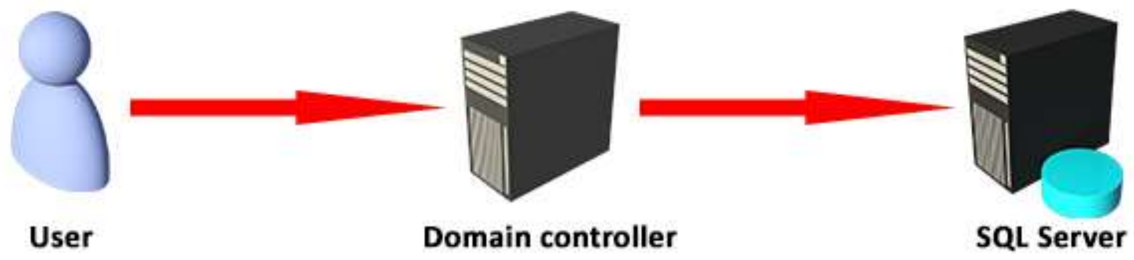
SQL Server administrators have unrestricted access to the Database Engine.

In a “Database Engine Configuration”, we need to configure how principals will make an authentication on a database engine. We have two options:

- Windows authentication
- Mixed authentication

Windows Authentication

In this case, the user has an account in the domain controller. SQL Server is a member of the domain, so it can use all domain resources including logins. Like in an operating system case, domains handle the authentication of users. If a user has a domain account and it is mapped in SQL Server under logins, it accesses SQL Server resources based on policy.

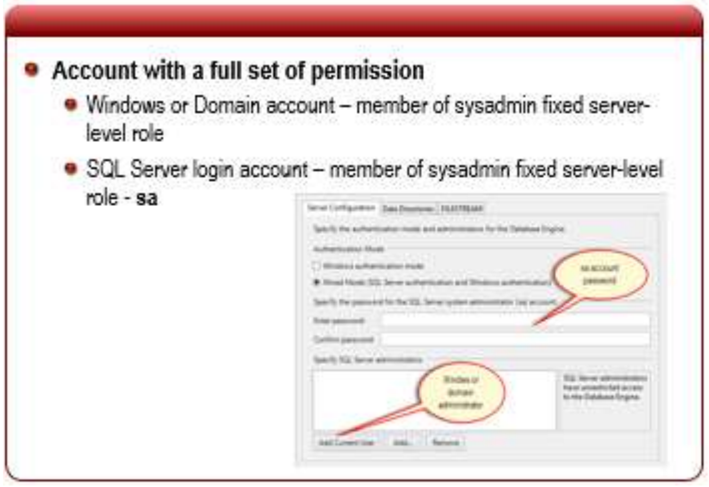


Mixed Authentication (SQL Server Authentication and Windows Authentication)

The other type of authentication is a mix of Windows and SQL Server authentication. It is a mix of the two authentications because the user can choose which one will be used to connect in a SQL Server instance. SQL Server authentication uses a user name and password which are both stored in a master database to authenticate a user.

User
SQL Server

Administrator Account



- Account with a full set of permission
  - Windows or Domain account – member of sysadmin fixed server-level role
  - SQL Server login account – member of sysadmin fixed server-level role - **sa**

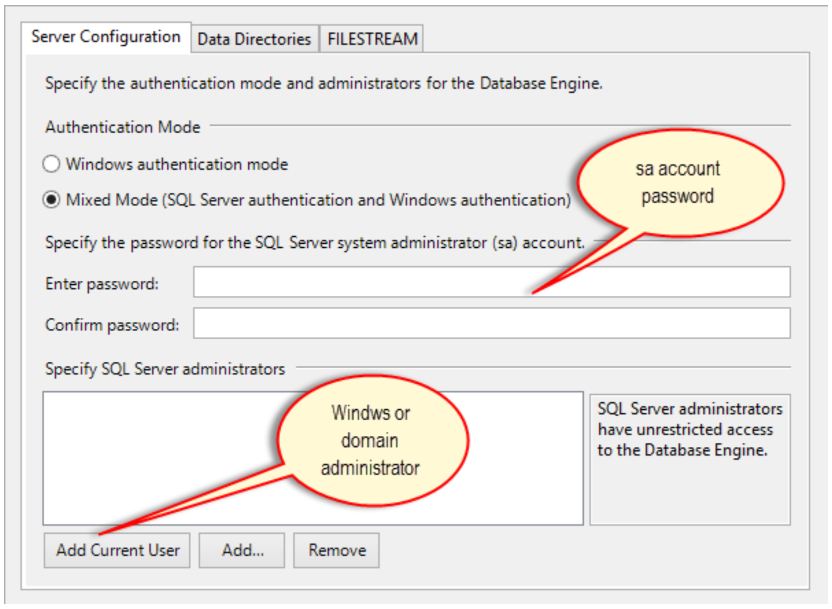
SQL Server requiring an account with a full set of permissions is an administrator account. Based on the type of the authentication selected, the administrator can be a:

- Windows or domain account
- SQL Server login account

Windows or domain account is any account that can connect to a SQL Server instance and has a fixed server-level role (i.e. sysadmin).

SQL Server login account can also be sysadmin, but requires a special login (**sa**). If we select a mixed authentication, we can also define the password for **sa**.

The installation procedure requires that we select an administrator account.



Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

Windows authentication mode [✓ Mixed Mode (SQL Server authentication and Windows authentication)]

Specify the password for the SQL Server system administrator (sa) account.

Enter password:

Confirm password:

Specify SQL Server administrators

SQL Server administrators have unrestricted access to the Database Engine.

Be aware that an administrator account has a full set of permissions and can literally do anything on a server- and database-level (i.e. increased responsibility).

If you select a SQL Server sa account, please note that user name and password travel through the network in clear text format if SSL is not enabled. It is recommended to not use sa accounts, if possible, as they produce greater risk of security breach.

Practice: Configuring SQL Server Service Account

In this practice, you will check and configure a SQL Server service account during and after the installation procedure
- Select service account during installation procedure
- Change service account after installation procedure

This practice requires SQL Server 2012 Enterprise Edition in Workgroup, not domain environment.

In this practice, you will check and configure a SQL Server service account during and after the installation procedure.

This practice requires SQL Server 2012 Enterprise Edition in Workgroup, not domain environment.

Exercise 1: Select service account during installation procedure

In this exercise, you will select a service account for SQL Server Database engine service.

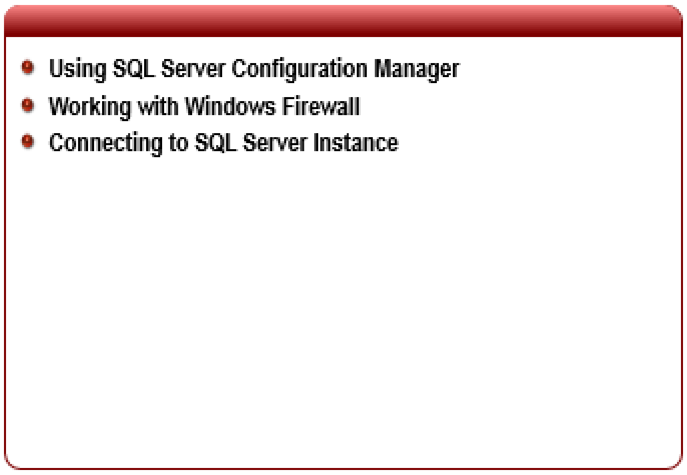
1. Start **SQL Server Setup** procedure, follow the installation until you reach **Server Configuration**.
2. Locate and select **SQL Server Database Engine service**.
3. Check service name, if service name is **NT Service\MSSQLSERVER**.
    i. What is the conclusion in this case?
4. Check service name, if service name is **NT Service\MSSQLSERVER$ SomeName**.
    i. What is the conclusion in this case?

Exercise 2: Change service account after installation procedure

In this exercise, you will select a different service account for SQL Server Database engine service.

1. Open **Control Panel**, click **Administrative Tools**, and click **Services**.
2. Locate **SQL Server service** in the list, and double click.
3. Click the **Log On** tab.
4. Select the **Local System** account, then click **OK**.
5. Once completed, return the settings to their original state.

Lesson 2: Security Steps after Installation



- Using SQL Server Configuration Manager
- Working with Windows Firewall
- Connecting to SQL Server Instance

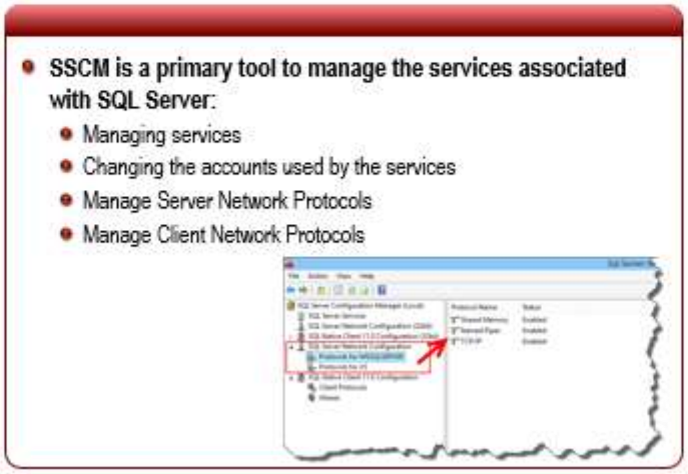
After the installation procedure, there is another set of security steps left to do. In some cases, without these security steps, you will not be able to connect on a SQL Server instance from client tools or applications. Two components needs to be modified: the SQL Server Configuration Manager and the Windows firewall.

Objectives

After completing this lesson, you will be able to:

- Use SQL Server Configuration Manager
- Use Windows Firewall with SQL Server
- Connect to a SQL Server instance

Using SQL Server Configuration Manager

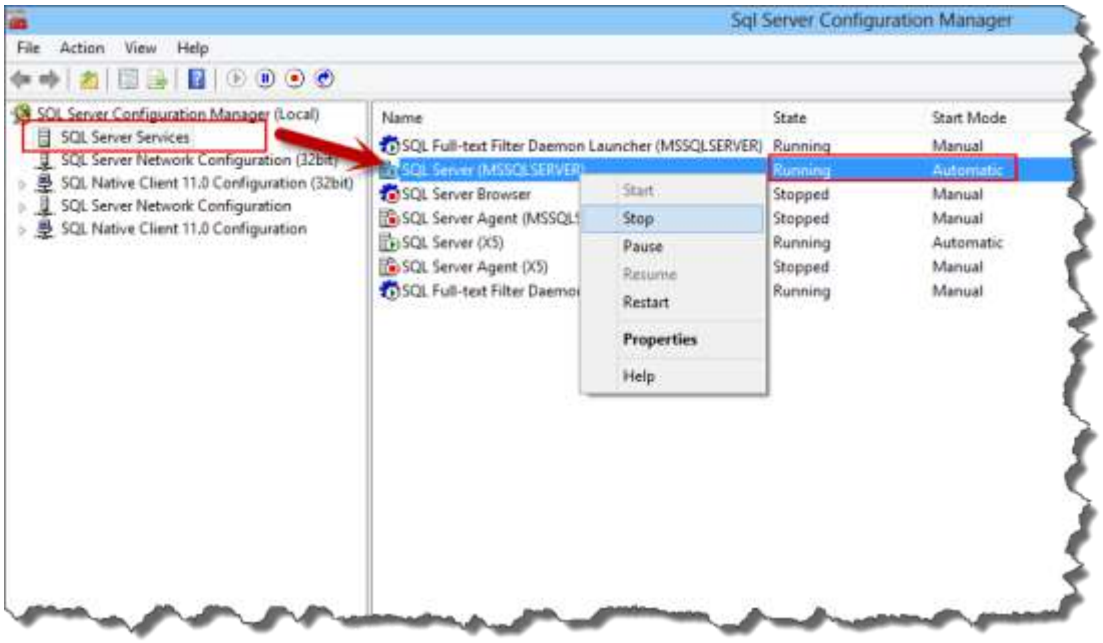


- SSCM is a primary tool to manage the services associated with SQL Server:
  - Managing services
  - Changing the accounts used by the services
  - Manage Server Network Protocols
  - Manage Client Network Protocols

As we have discussed in Module 01, the SSCM is a primary tool to manage the services associated with SQL Server, to configure the network protocols used by SQL Server, and to manage the network connectivity configuration from SQL Server client computers.

Managing Services

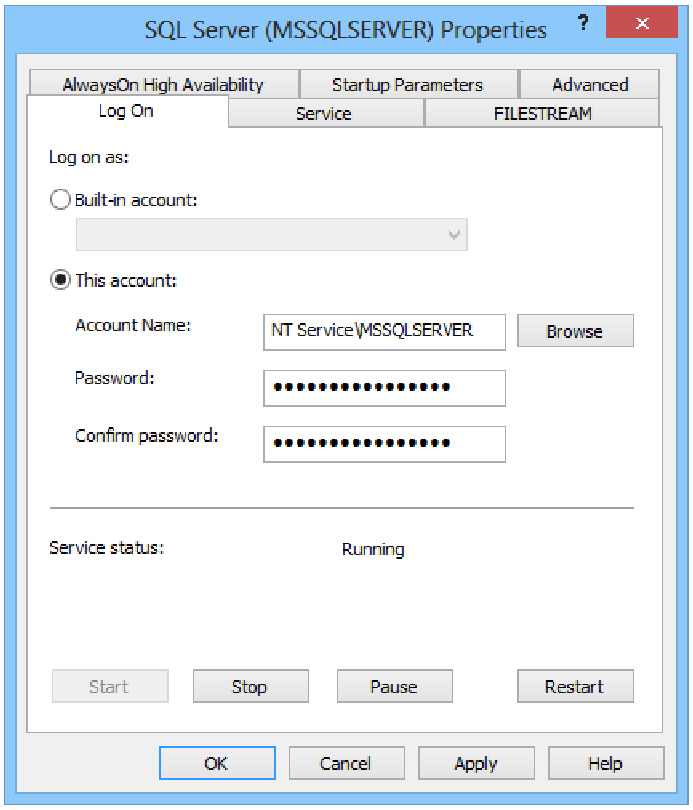
Use the SQL Server Configuration Manager to view and change service properties, as well as to start, pause, resume, or stop the services.


Sql Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Start
Stop
Pause
Resume
Restart
Properties
Help

Name	State	Start Mode
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	Running	Manual
SQL Server (MSSQLSERVER)	Running	Automatic
SQL Server Browser	Stopped	Manual
SQL Server Agent (MSSQLS	Stopped	Manual
SQL Server (X5)	Running	Automatic
SQL Server Agent (X5)	Stopped	Manual
SQL Full-text Filter Daemo	Running	Manual

Changing the Accounts used by the Services

Manage the SQL Server services using the SQL Server Configuration Manager. As an additional benefit, passwords changed using the SQL Server Configuration Manager, Server Management Objects (SMO), or Windows Management Instrumentation (WMI) take effect immediately without restarting the service.



Log on as:

( ) Built-in account:

(●) This account:

Account Name: NT Service\MSSQLSERVER

Password: ●●●●●●●●●●●●●●●●

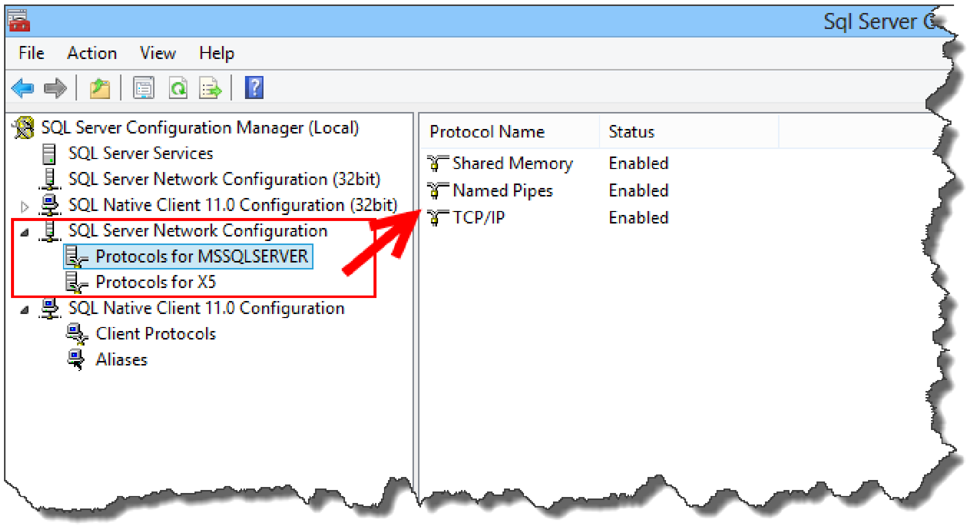
Confirm password: ●●●●●●●●●●●●●●●●

Service status: Running

Manage Server Network Protocols

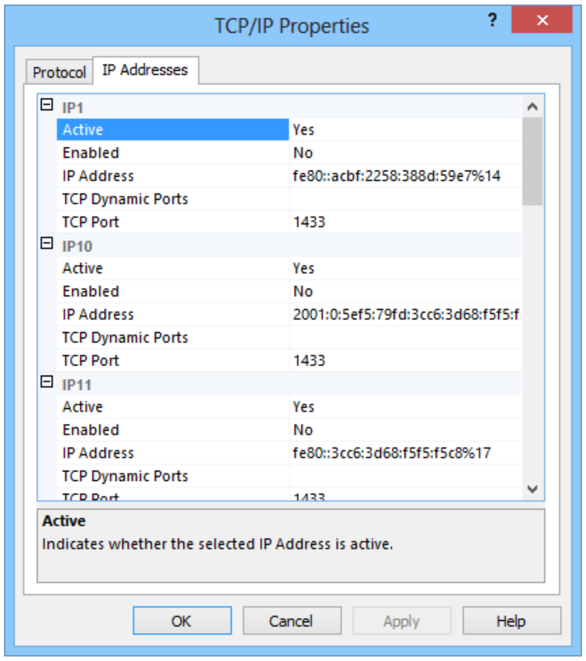
The SQL Server Configuration Manager allows you to configure server and client network protocols, as well as connectivity options. After the correct protocols are enabled, you usually do not need to change the server network connections. However, you can use the SQL Server Configuration Manager if you need to reconfigure the server connections. For example, to enable the client connections on a particular network protocol, port, or pipe, you would use the SQL Server Configuration Manager.

SQL Server supports Shared Memory, TCP/IP, and Named Pipe protocols. SQL Server does not support the VIA, Banyan VINES Sequenced Packet Protocol (SPP), Multiprotocol, AppleTalk, or NWLink IPX/SPX network protocols. Clients previously connecting with these protocols must select a different protocol to connect to SQL Server.


Sql Server C
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
SQL Server Network Configuration
Protocols for MSSQLSERVER
Protocols for X5
SQL Native Client 11.0 Configuration
Client Protocols
Aliases

Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Enabled
TCP/IP	Enabled

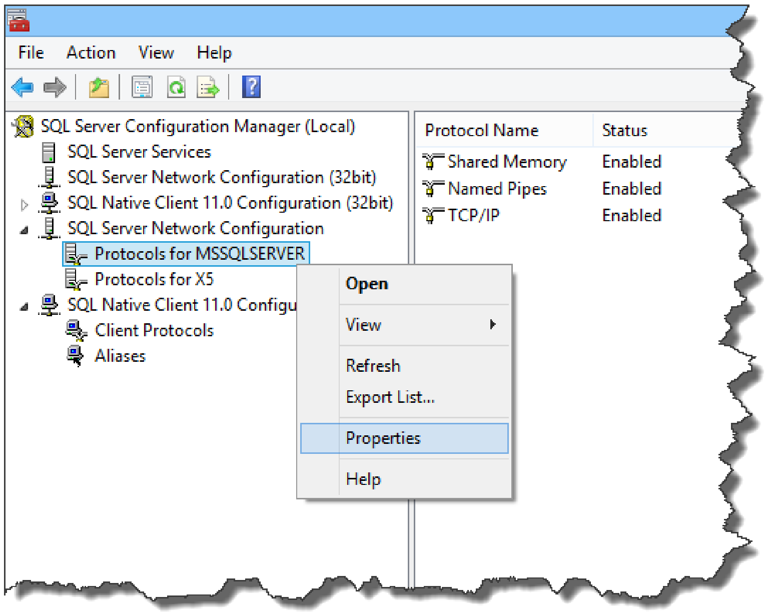
We can select a TCP/IP server-side protocol by simply right-clicking on **TCP/IP.**



IP1	
Active	Yes
Enabled	No
IP Address	fe80::acbf:2258:388d:59e7%14
TCP Dynamic Ports	
TCP Port	1433
IP10	
Active	Yes
Enabled	No
IP Address	2001:0:5ef5:79fd:3cc6:3d68:f5f5:f
TCP Dynamic Ports	
TCP Port	1433
IP11	
Active	Yes
Enabled	No
IP Address	fe80::3cc6:3d68:f5f5:f5c8%17
TCP Dynamic Ports	
TCP Port	1433

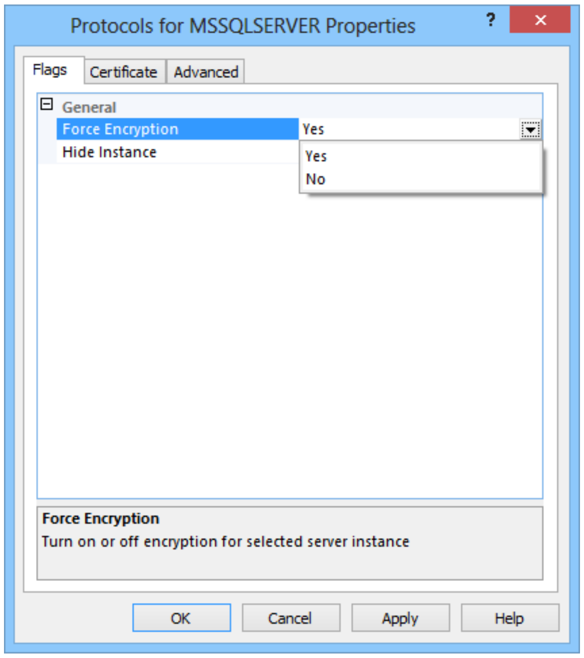
Active
Indicates whether the selected IP Address is active.

Also, you can right-click on the instance server protocols.


File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
SQL Server Network Configuration
Protocols for MSSQLSERVER
Protocols for X5
SQL Native Client 11.0 Configu
Client Protocols
Aliases
Open
View
Refresh
Export List...
Properties
Help

Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Enabled
TCP/IP	Enabled

Properties will activate the additional parameter configuration steps.



Force Encryption	Yes
Hide Instance	

Force Encryption
Turn on or off encryption for selected server instance

When a certificate is installed on the server, use the Flags tab on the Protocols for MSSQLSERVER Properties dialog box to view or specify the protocol encryption and hide instance options. Microsoft SQL Server must be restarted to enable or disable the Force Encryption setting.

To encrypt connections, you should provision the SQL Server Database Engine with a certificate. If a certificate is not installed, SQL Server will generate a self-signed certificate when the instance is started. This self-signed certificate can be used instead of a certificate from a trusted certificate authority, but it does not provide authentication or non-repudiation.

Use the Certificate tab on the Protocols for MSSQLSERVER Properties dialog box to select a certificate for Microsoft SQL Server, or to view the properties of a certificate. All fields are blank until a certificate is selected.

Certificates are stored locally for the users on the computer. To load a certificate for use by SQL Server, you must be running the SQL Server Configuration Manager under the same user account as the SQL Server service.

Use the Advanced tab on the Protocols for MSSQLSERVER Properties dialog box to configure Extended Protection for Authentication for the SQL Server Database Engine. Extended Protection is a feature of the network components implemented by the operating system. Extended Protection is available in Windows 7 and Windows Server 2008 R2, and is included in service packs for older operating systems. SQL Server is more secure when connections are made using Extended Protection. Some benefits of Extended Protection require Force Encryption to be selected on the Flags tab.

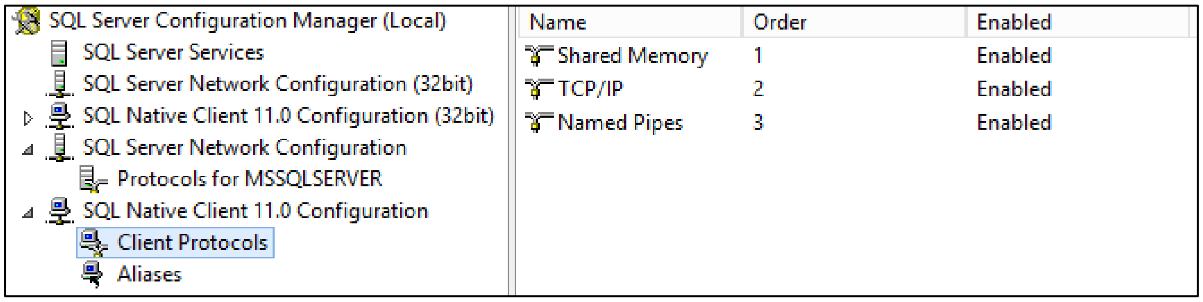
Manage Client Network Protocols

SQL Server Native Client is the network library that client computers use to connect to SQL Server. The settings configured in the SQL Server Native Client Configuration are used on the computer running the client program.

When configured on the computer running SQL Server, the settings affect only those client programs running on the server. These settings do not affect clients connecting to previous versions of SQL Server, unless they are using the client tools starting with SQL Server, such as SQL Server Management Studio.

SQL Server Management Studio allows you to create or remove an alias, change the order in which protocols are used, or view properties for the following:

- Server Alias - The server alias used for the computer to which the client is connecting.
- Protocol - The network protocol used for the configuration entry.
- Connection Parameters - The parameters associated with the connection address for the network protocol configuration.

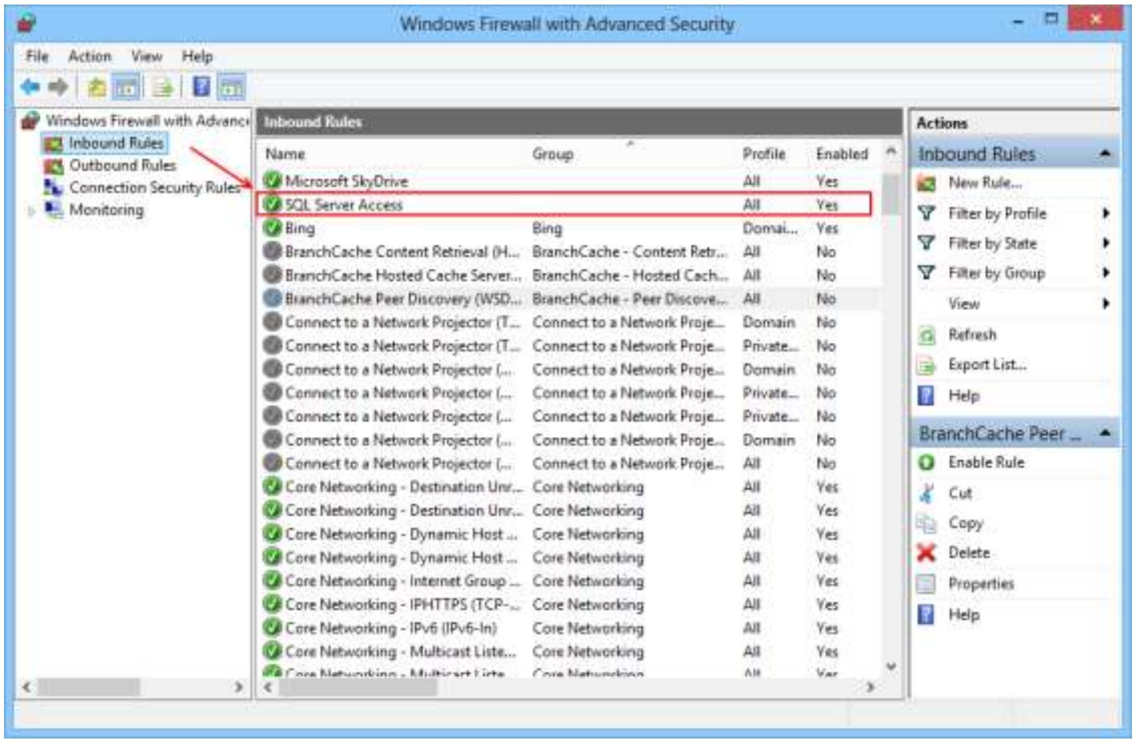

SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
SQL Server Network Configuration
Protocols for MSSQLSERVER
SQL Native Client 11.0 Configuration
Client Protocols
Aliases

Name	Order	Enabled
Shared Memory	1	Enabled
TCP/IP	2	Enabled
Named Pipes	3	Enabled

Working with Windows Firewall

- SQL Server Inbound Access Rule under Windows Firewall

SQL Server is a client/server environment dependent on elements that are not always in SQL Server "jurisdiction". A Windows server comes with a built-on Windows Firewall and is closed by default. If we configure TCP/IP access in the SQL Server Configuration Manager, it will not work in the network environment because the SQL Server ports are closed.



Windows Firewall with Advanced Security

File Action View Help

Windows Firewall with Advanced
- Inbound Rules
- Outbound Rules
- Connection Security Rules
- Monitoring

Inbound Rules

Name	Group	Profile	Enabled
Microsoft SkyDrive		All	Yes
SQL Server Access		All	Yes
Bing	Bing	Domai...	Yes
BranchCache Content Retrieval (H...	BranchCache - Content Retr...	All	No
BranchCache Hosted Cache Server...	BranchCache - Hosted Cach...	All	No
BranchCache Peer Discovery (WSD...	BranchCache - Peer Discove...	All	No
Connect to a Network Projector (T...	Connect to a Network Proje...	Domain	No
Connect to a Network Projector (T...	Connect to a Network Proje...	Private...	No
Connect to a Network Projector (...	Connect to a Network Proje...	Domain	No
Connect to a Network Projector (...	Connect to a Network Proje...	Private...	No
Connect to a Network Projector (...	Connect to a Network Proje...	Private...	No
Connect to a Network Projector (...	Connect to a Network Proje...	Domain	No
Connect to a Network Projector (...	Connect to a Network Proje...	All	No
Core Networking - Destination Unr...	Core Networking	All	Yes
Core Networking - Destination Unr...	Core Networking	All	Yes
Core Networking - Dynamic Host ...	Core Networking	All	Yes
Core Networking - Dynamic Host ...	Core Networking	All	Yes
Core Networking - Internet Group ...	Core Networking	All	Yes
Core Networking - IPHTTPS (TCP-...	Core Networking	All	Yes
Core Networking - IPv6 (IPv6-In)	Core Networking	All	Yes
Core Networking - Multicast Liste...	Core Networking	All	Yes

Actions

Inbound Rules
- New Rule...
- Filter by Profile
- Filter by State
- Filter by Group
- View
- Refresh
- Export List...
- Help

BranchCache Peer ...
- Enable Rule
- Cut
- Copy
- Delete
- Properties
- Help

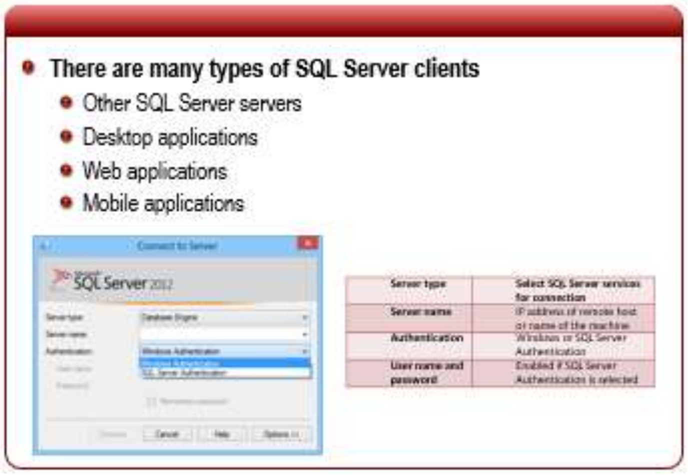
Your task is to allow client access to SQL Server through Windows Firewall. There is no default SQL Server rule. It needs to be created within the application. The default SQL Server TCP port is 1433.

When accessing a local SQL Server instance, there is no need to open a port.

If your server and network environment has more than one firewall, you will need to address all of them.

It is not recommended to have more than one firewall per machine. Additional firewalls can slow down communication.

Connecting to a SQL Server Instance



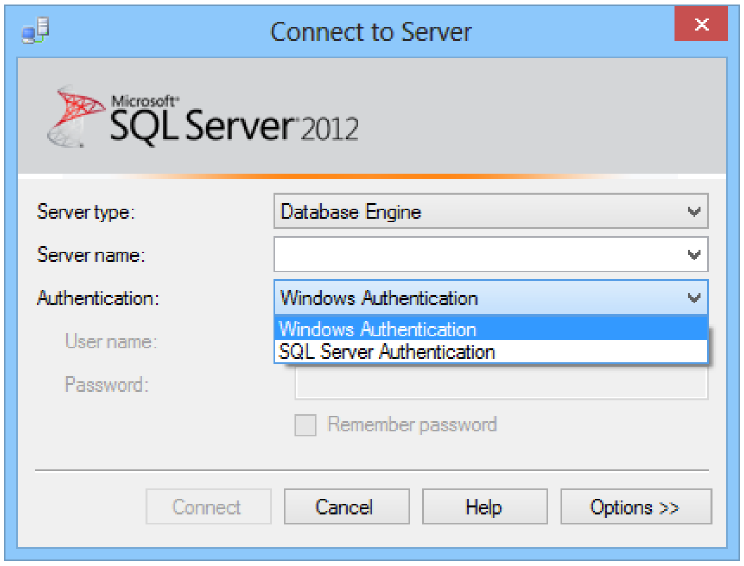
- **There are many types of SQL Server clients**
  - Other SQL Server servers
  - Desktop applications
  - Web applications
  - Mobile applications

Server type	Select SQL Server services for connection
Server name	IP address of remote host or name of the machine
Authentication	Windows or SQL Server Authentication
User name and password	Enabled if SQL Server Authentication is selected

There are many types of SQL Server clients. A significant SQL Server client is SSMS for administration and development jobs. Other types of clients are:

- Other SQL Server servers
- Desktop applications
- Web applications
- Mobile applications

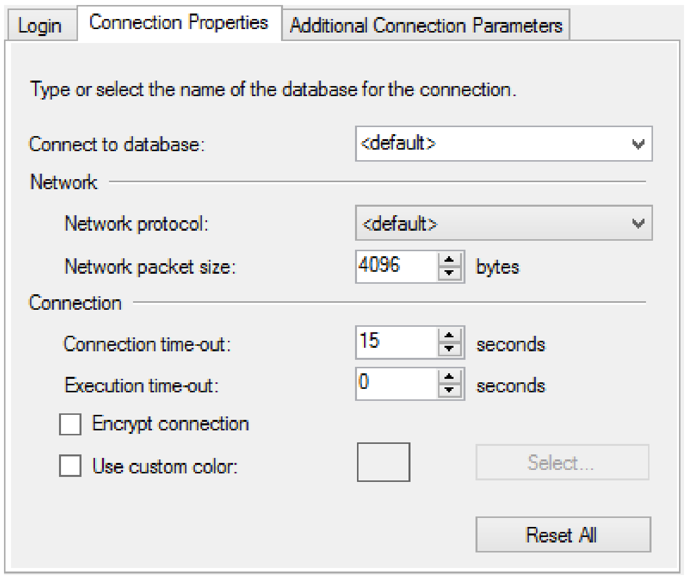
Each of these types of client can be categorized into either administration- or development-purpose clients. It is relatively simple to connect to a SQL Server instance through SSMS.



Server type: Database Engine
Server name:
Authentication: Windows Authentication
Windows Authentication
SQL Server Authentication
User name:
Password:
Remember password

Server type	Select SQL Server services for connection
Server name	IP address of remote host or name of the machine
Authentication	Windows or SQL Server Authentication
User name and password	Enabled if SQL Server Authentication is selected

Click OPTIONS >> to see additional connection parameters and options.



Type or select the name of the database for the connection.

Connect to database: <default>

Network

Network protocol: <default>

Network packet size: 4096 bytes

Connection

Connection time-out: 15 seconds

Execution time-out: 0 seconds

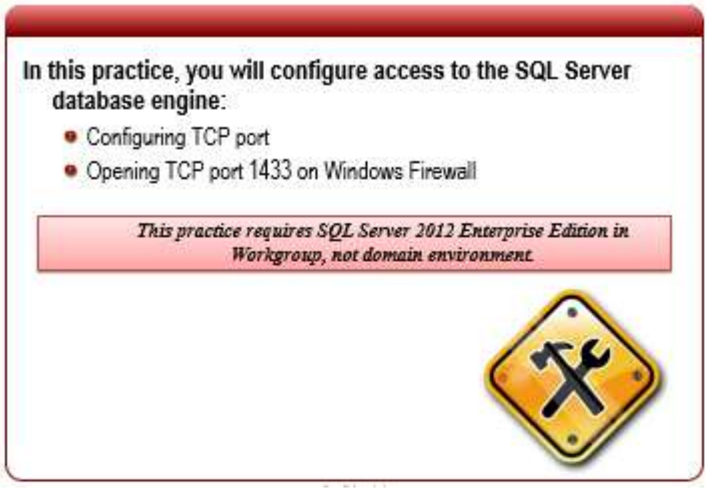
☐ Encrypt connection

☐ Use custom color:

We have the option of selecting default database, network protocol, default network packet size, connection time-out, and query execution time-out.

We can also use the “Encrypt connection” security feature which allows for all communication between the client and the server to be encrypted.

Practice: Configuring Access to SQL Server


In this practice, you will configure access to the SQL Server database engine:

- Configuring TCP port
- Opening TCP port 1433 on Windows Firewall

This practice requires SQL Server 2012 Enterprise Edition in Workgroup, not domain environment.

In this practice, you will configure access to the SQL Server database engine.

This practice requires SQL Server 2012 Enterprise Edition in Workgroup, not domain environment.

Exercise 1: Configuring TCP port

In this exercise, you will configure port 1433 on the database engine.

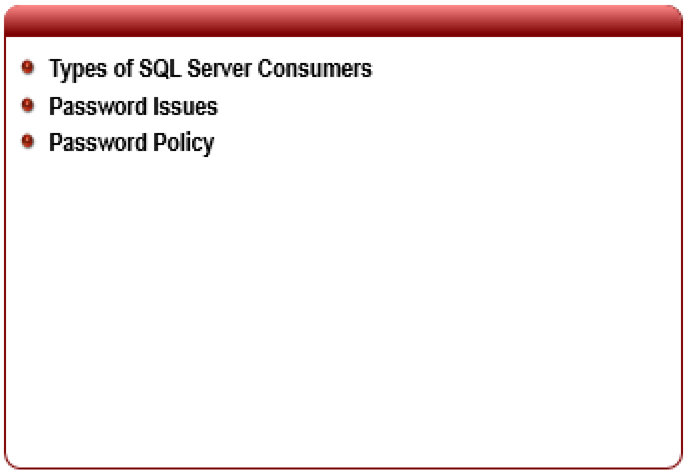
1. Open **SQL Server Configuration Manager**, expand **SQL Server Network Configuration** on the tree panel, and then click **Protocols**.
2. Right-click **TCP/IP**, select **Properties**, and then click **IP Address**.
3. Ensure **TCP Port** is set to **1433** and click **OK**.

a. If you have more than one network adapter, you will need to find which one is for external access.

Exercise 2: Opening TCP port 1433 on Windows Firewall

1. Open **Control Panel**, click **Windows Firewall**, and then click **Advanced Settings**.
2. Right-click **Inbound Rules**, and select **New Rule**.
3. Select **Port** option, click **TCP**, click **Specific local ports**, enter **1433**, and then click **Next**.
4. Select **Allow the connection**, click **Next**, select the **Domain**, **Private** and **Public** check boxes, and then click **Next**.
5. Enter the name for the rule (**SQL Server external Access**, for an example) and click **Finish**.

Lesson 3: SQL Server Resources Consumers



- Types of SQL Server Consumers
- Password Issues
- Password Policy

In this lesson, you will learn about the types of SQL Server consumers and their specific needs. Afterwards, you will be able to configure an efficient security policy.

Objectives

After completing this lesson, you will be able to:

- Identify types of SQL Server consumers
- Understand password issues
- Implement a password policy

Types of SQL Server Consumers

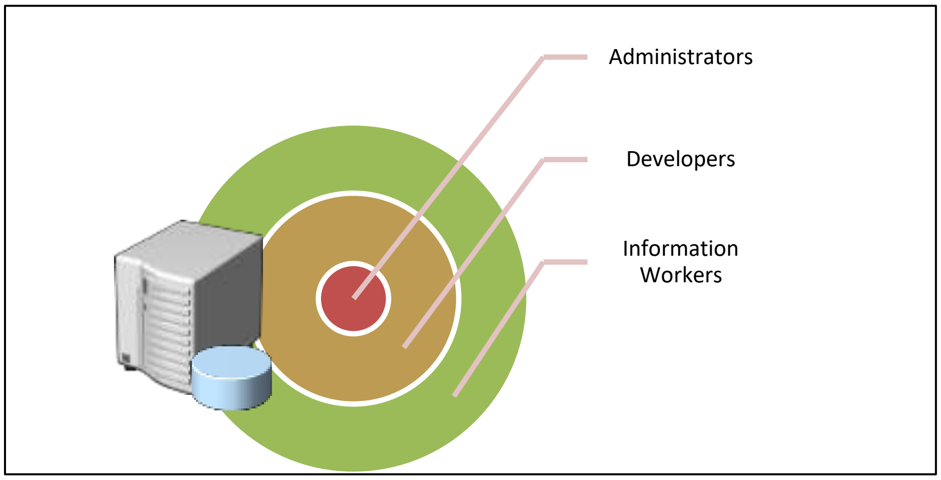
- There are three global groups of SQL Server users

Based on user needs, you can create an efficient security policy, and assign correct server and database roles (or custom roles).

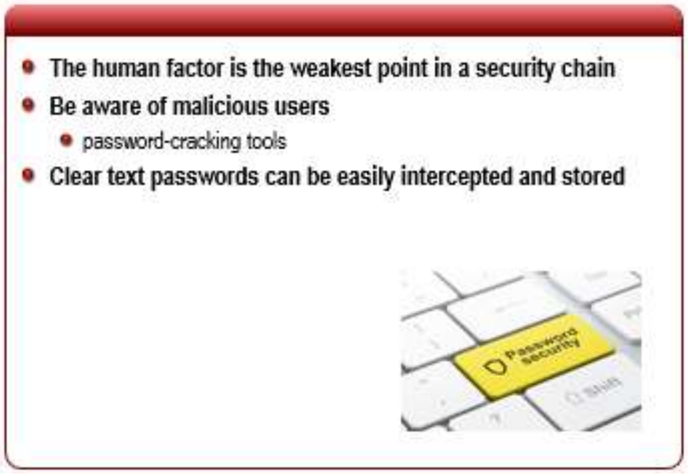
There are three global groups of SQL Server users:

- Administrators – group of users who use SSMS or PowerShell to administer and configure a SQL Server instance, and to write T-SQL code.
- Developers – group of user who use different development tools, such a Visual Studio, to connect to a SQL Server instance and to program application logic.
- Information Workers – group of users who create business intelligence using different front ends, such as reporting/analysis services, Excel, etc.

All of the global groups of SQL Server users have different tools and network configurations (local, remote, or web). It is important to note the tools and configurations of each global group of SQL Server user.



Password Issues



- The human factor is the weakest point in a security chain
- Be aware of malicious users
  - password-cracking tools
- Clear text passwords can be easily intercepted and stored

Password security
Shift

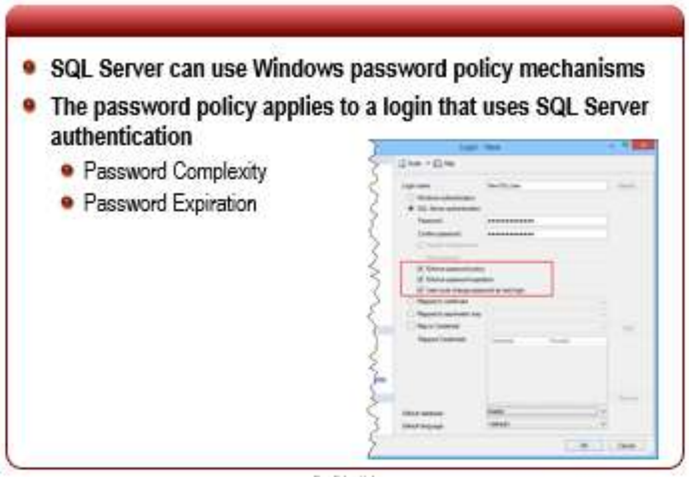
The human factor is, in most cases, the weakest point in a security chain. The mishandling of passwords by users is just one of the reasons how the human factor impacts the security chain.

Although it is not usually a DBA or developers' set of tasks to enforce the safekeeping of passwords among users, it is important to note this issue and to strongly encourage users to securely manage their passwords.

Be aware that malicious users can easily find a wide range of password-cracking tools (called password utility or password auditing tools). The purpose of these password auditing tools is to find weak user passwords in a relatively short period of time. In a large number of cases, the tool will be successful in cracking the password. Because of this threat, it is important to define and consistently apply a password policy, especially when dealing with databases.

The second issue is the clear text submission of a password. SQL Server Authentication is an example which allows for passwords to be submitted using clear text. Clear text passwords can be easily intercepted and stored. The only way to prevent the interception and storage of a password is to use a Secure Socket Layer (SSL) during the encryption process to ensure secure transmission.

Password Policy



- SQL Server can use Windows password policy mechanisms
- The password policy applies to a login that uses SQL Server authentication
  - Password Complexity
  - Password Expiration

SQL Server can use Windows password policy mechanisms. The password policy applies to a login that uses SQL Server authentication, and to a contained database user with a password.

SQL Server can apply the same complexity and expiration policies used in Windows to passwords used inside SQL Server. This functionality depends on the NetValidatePasswordPolicy API.

Password Complexity

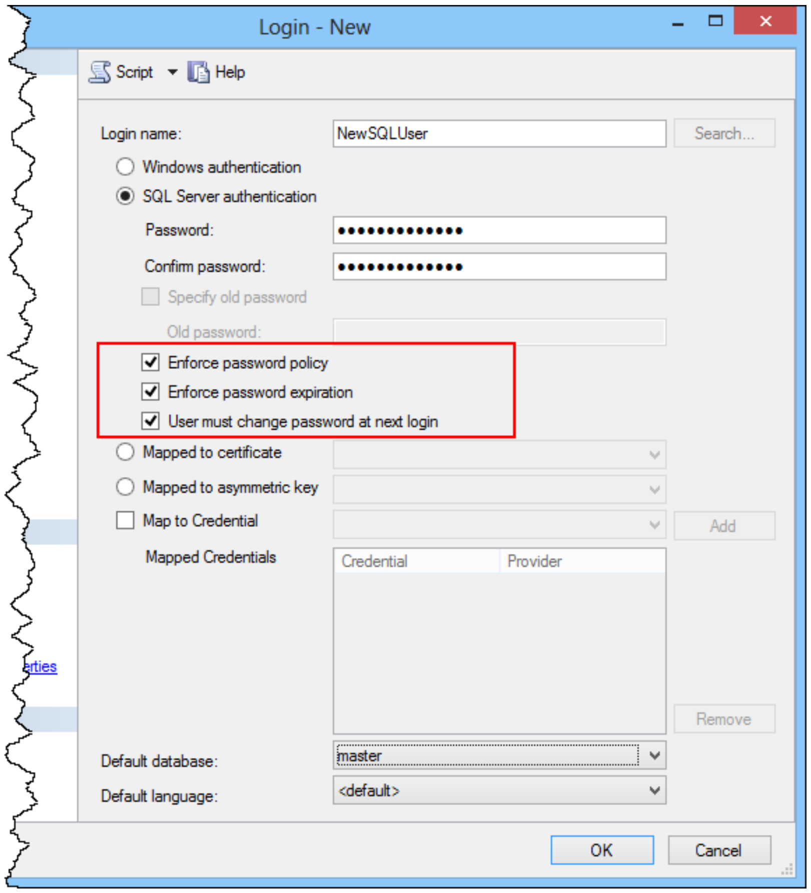
Password complexity policies are designed to deter brute force attacks by increasing the number of possible passwords. When password complexity policy is enforced, new passwords must meet the following guidelines:

- The password does not contain all or part of the account name of the user. Part of an account name is defined as three or more consecutive alphanumeric characters delimited on both ends by white space such as space, tab, and return, and any of the following characters: comma (,), period (.), hyphen (-), underscore (_), or number sign (#).
- The password is at least eight characters long.
- The password contains characters from three of the following four categories: 1) Latin uppercase letters (A through Z); 2) Latin lowercase letters (a through z); 3) base 10 digits (0 through 9); and 4) Non-alphanumeric characters such as: exclamation point (!), dollar sign ($), number sign (#), or percent (%).

Passwords can be up to 128 characters long. You should use passwords that are as long and complex as possible.

Password Expiration

Password expiration policies are used to manage the lifespan of a password. When SQL Server enforces a password expiration policy, users are reminded to change old passwords.



Login name: NewSQLUser

( ) Windows authentication

(●) SQL Server authentication

Password: ••••••••••••••

Confirm password: ••••••••••••••

☐ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

( ) Mapped to certificate

( ) Mapped to asymmetric key

☐ Map to Credential

Mapped Credentials

Default database: master

Default language: <default>

Practice: Configuring Password Policy

In this practice, you will configure a password policy on SQL Server:
- Configuring password policy through SSMS
- Configuring password policy through T-SQL code

This practice requires SQL Server 2012 Enterprise Edition in Workgroup, not domain environment.

In this practice, you will configure a password policy on SQL Server.

This practice requires SQL Server 2012 Enterprise Edition in Workgroup, not domain environment.

Exercise 1: Configuring password policy through SSMS

In this exercise, you will configure a password policy using SQL Server Management Studio.

1. Open **SQL Server Management Studio**, expand the **Security** node in **Object Explorer**, and then expand **Logins.**
2. Right-click **New Login**.
3. Select **SQL Server authentication**, and in the Login name type **SecureUser**.
4. Click **Activate**, and select the **Enforce password policy**, **Enforce password expiration**, and **User must change password at next login** check boxes.
5. In password field, enter **S0m3extr4pa$$**.
6. Confirm same password and click **OK**.

Exercise 2: Configuring password policy through T-SQL code

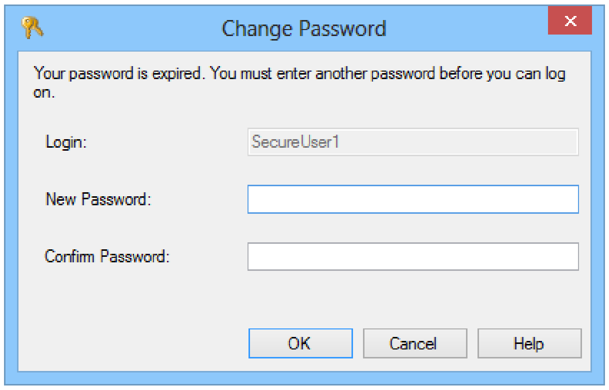
In this exercise, you will configure a password policy using SQL Server Management Studio.

1. Open **SQL Server Management Studio** and open a new query window.
2. Type the following T-SQL code, and click **Run** or press **F5**.

```
        USE master
```

```
GO
CREATE LOGIN SecureUser1 WITH PASSWORD='S0m3extr4pa$$'
 MUST_CHANGE,
 CHECK_EXPIRATION=ON,
 CHECK_POLICY=ON
GO
```

3. Disconnect session in SSMS.
4. Open new connection.
5. Enter credential from one of the new logins you have created.
6. In both cases, after connection, you will get the **Change Password** dialog window.



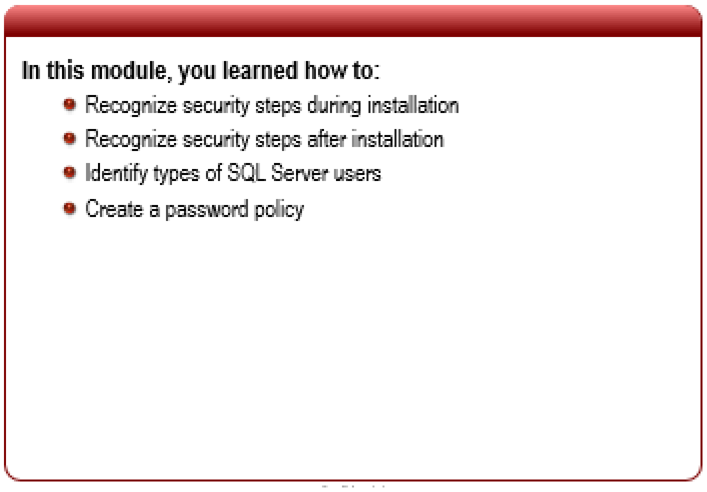
Your password is expired. You must enter another password before you can log on.

Login: SecureUser1

New Password:

Confirm Password:

Summary



In this module, you learned how to:

- Recognize security steps during installation
- Recognize security steps after installation
- Identify types of SQL Server users
- Create a password policy

This module taught you the basics and helped you to identify key security points during and after a SQL Serve installation. Now, you know how to: set a proper authentication process, set a user account for SQL services, open the proper TCP port, and deal with user password policy at the database-level.

Objectives

After completing this module, you learned how to:

- Recognize security steps during installation
- Recognize security steps after installation
- Identify types of SQL Server users
- Create a password policy