Configuration Manager on MS SQL Database

Managing large installations by using a central database

Configuration Manager

Microsoft® SQL Server®

Table of contents

1 Introduction

The Bosch Configuration Manager is a free and the central tool to scan, configure and manage Bosch devices. Over the years, it has seen a lot of enhancements, always keeping it up with newly introduced and evolving Bosch security products.

So far, Configuration Manager has been designed and used as a solitary client, holding its own local database. Though this could be centralized, working on this with multiple clients simultaneously had its challenges and restrictions.

Now, with version 7.70, we introduce the possibility to use a SQL database instead of the local database, at the same time enabling multiple simultaneous client access.

Since database installations and setups may vary extensively, only a certain number of setups could be tested and verified, so we need to consider this still an experimental feature. Feedback is welcomed to help us further improving this feature.

Configuration Manager 7.70 supports MS SQL Server 2019 and 2022, both as on-premises installation or in an Azure cloud.

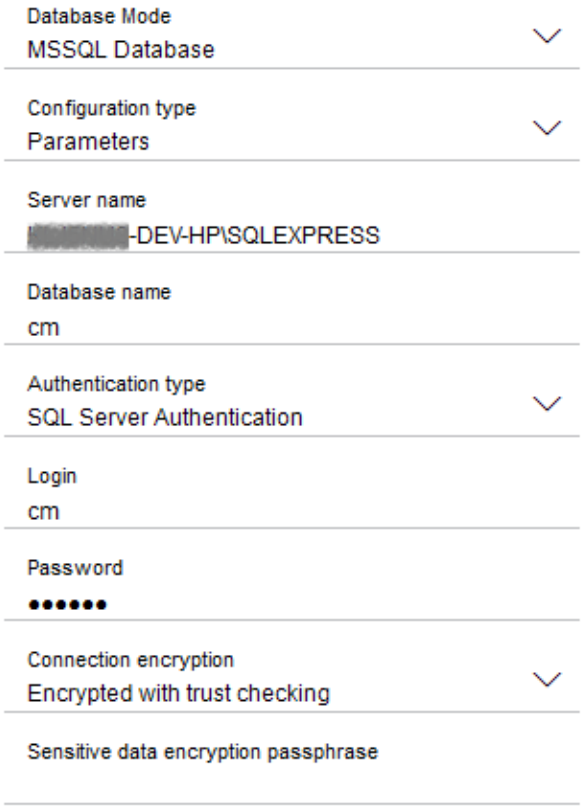
In this tech note we assume a pre-installed SQL server that only needs to be configured for use with Configuration Manager.
If this is not the case, please refer to the appendix for installation hints and the references section for further documentation sources.

2 Configuration Manager setup for MS SQL as root DB

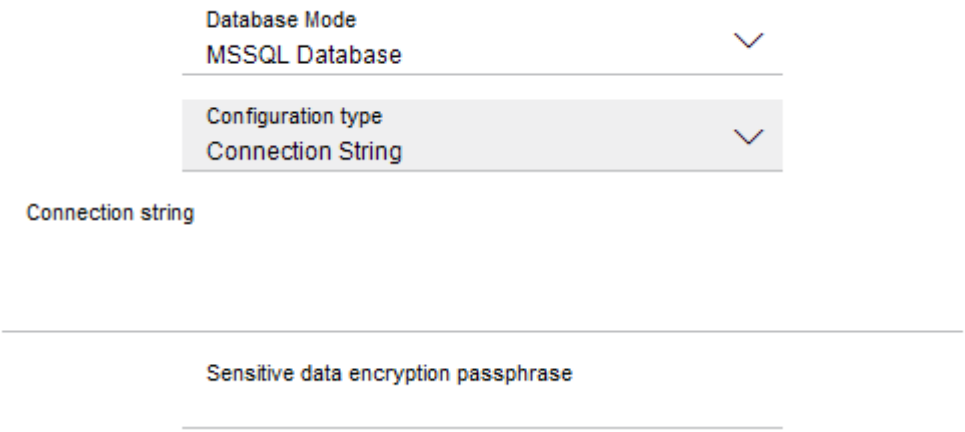
Navigate to Preferences → Page Access → Group Database.
From the combo-box dropdown list Database Mode, select MSSQL Database.

The Configuration Manager offers 2 configuration modes:

1. **Parameter-based** configuration allows selecting basic connection parameters using Configuration Manager UI.



2. **Connection String-based** mode allows for a possibility to define custom parameters that are not configurable via CM UI.



In this tech note we will focus on Parameter-based mode only.

2.1 Configuring Parameter-based mode

Possible parameters:

- Server name
  Host name combined with instance id by "\"

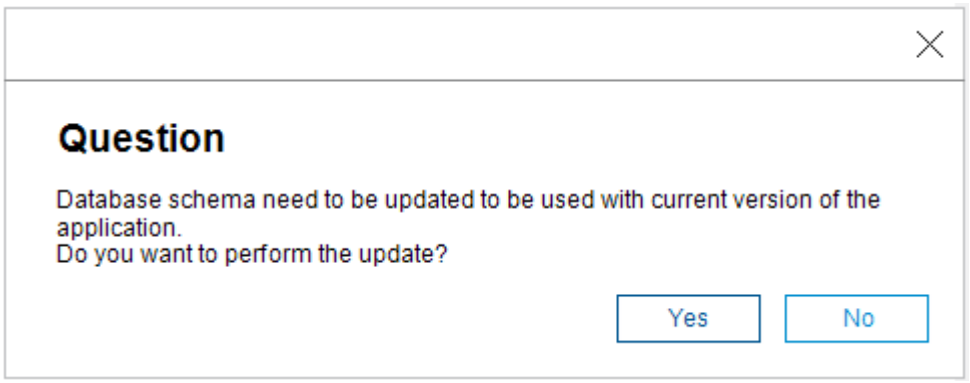
- Database name
  Database to be used. This can be an existing database, or a non-existing database in case a user with database creation rights is used.

- Authentication type
  Can be chosen as Windows authentication, or SQL Server authentication.
  In case of Windows authentication, no additional input is required.
  In case of SQL Server authentication, a username and password are required *(see chapter 3.3.1)*.

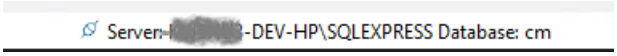
- Connection encryption
  Defines if traffic between client and server is encrypted as well as if server certificate is being verified.
  Recommended option is: Encrypted with trust check.

- Sensitive database encryption passphrase
  A passphrase used to encrypt data in the database.
  Recommended for small environments where column encryption is not enabled on the server.
  For corporate environments, a method that uses built-in column encryption is recommended *(see chapter 3.4.)*.

After successful database connection, Configuration Manager will verify the version of the database and try to update the database schema.

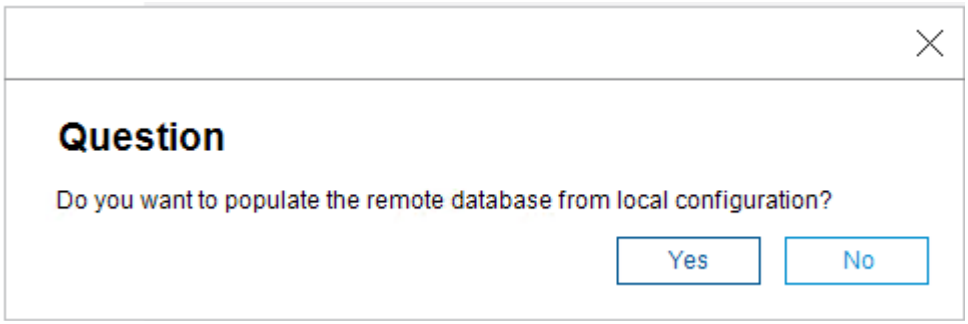


Successful connection is indicated by status message in the application's footer.



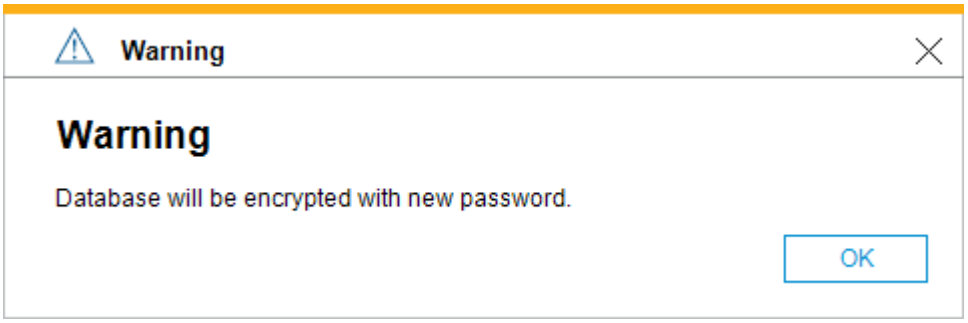
2.2 Initial importing a local database

In case it is the first time a user switched from file-based database to SQL database, Configuration Manager will offer the possibility to import the local database into the SQL server.



2.3 Sensitive database encryption passphrase

If provided, sensitive data, like access credentials for devices, will be encrypted with this passphrase.
Changing of the sensitive data encryption passphrase on existing database will trigger re-encryption of all sensitive records.



Note:
For corporate environments, a method that uses built-in column encryption is recommended *(see chapter 3.4.)*.

After enabling this option, all clients must use the same passphrase to access sensitive data. If the database uses built-in column encryption and a sensitive data encryption passphrase is provided, this will be used as additional encryption. Though it is not needed to use both encryptions, it is still possible.

3 Appendix

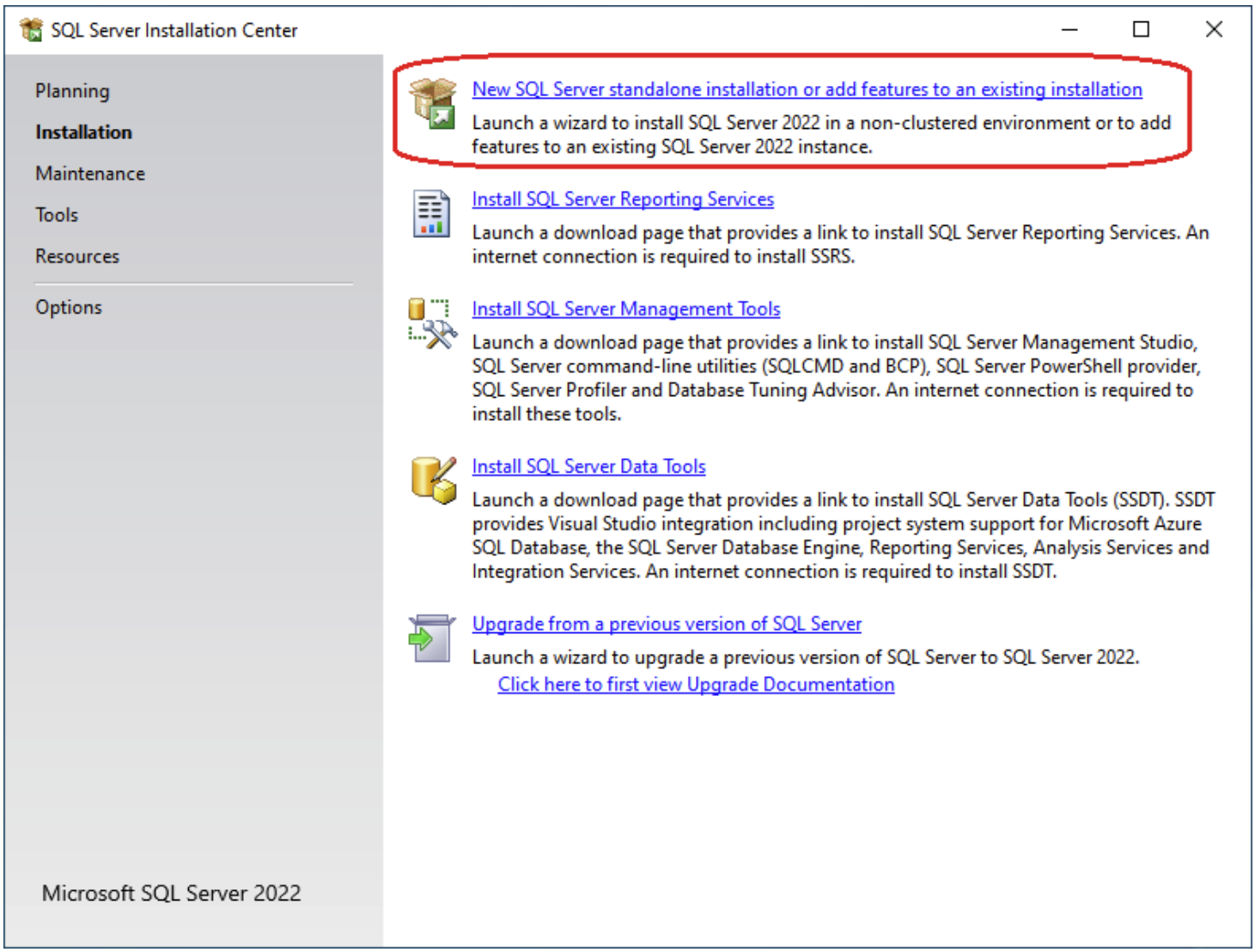
In this guideline we aim to provide hints for a SQL Server installation and setup that works with Configuration Manager. This is based on our own work processes and does not claim completeness nor correctness.

3.1 Installation of MS SQL Server

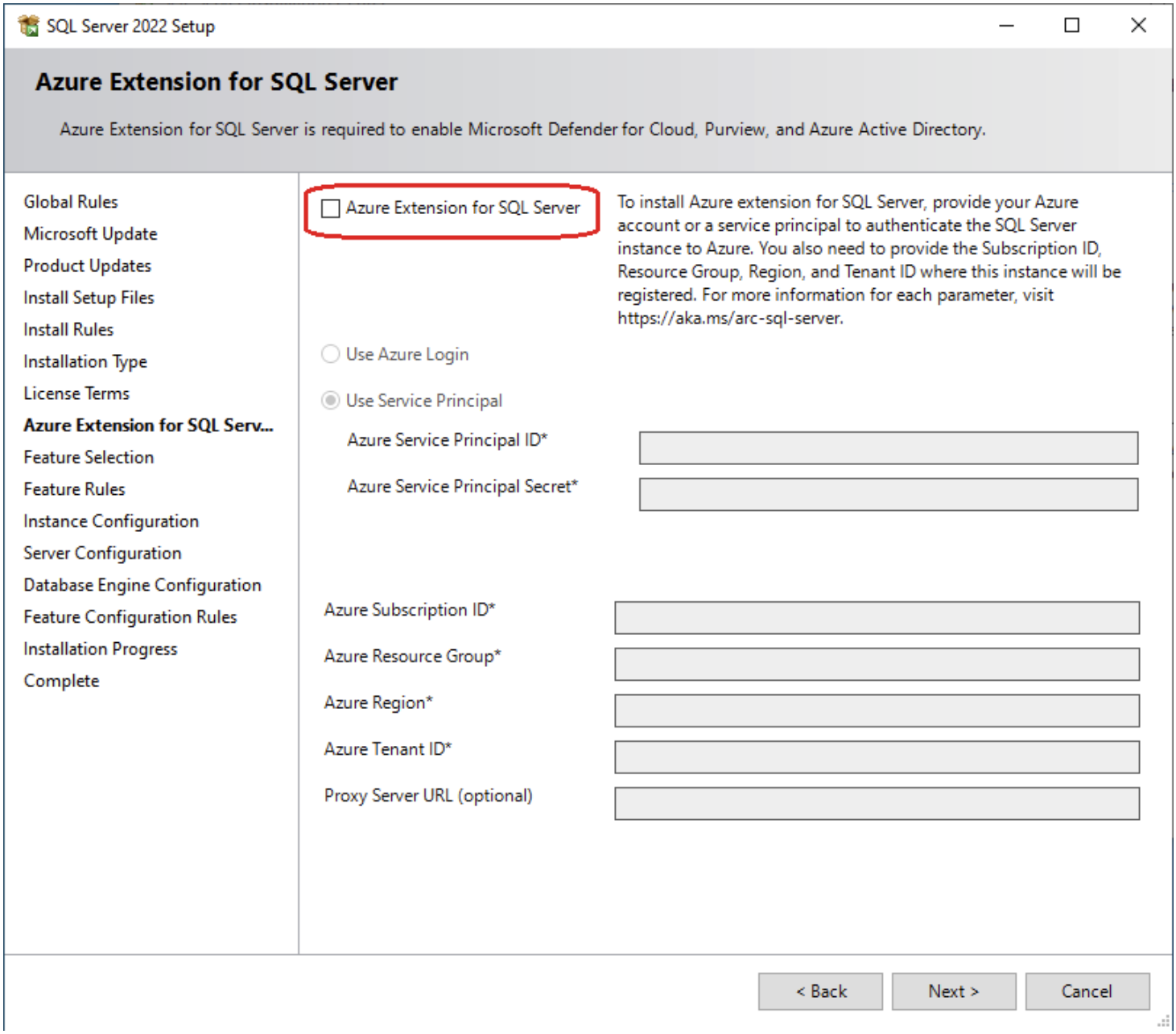
Download one of following versions of MS SQL server:

- SQL Server 2022 Express
  https://go.microsoft.com/fwlink/p/?linkid=2216019&clcid=0x409&culture=en-us&country=us
- SQL Server 2019 Express
  https://www.microsoft.com/en-us/download/confirmation.aspx?id=101064

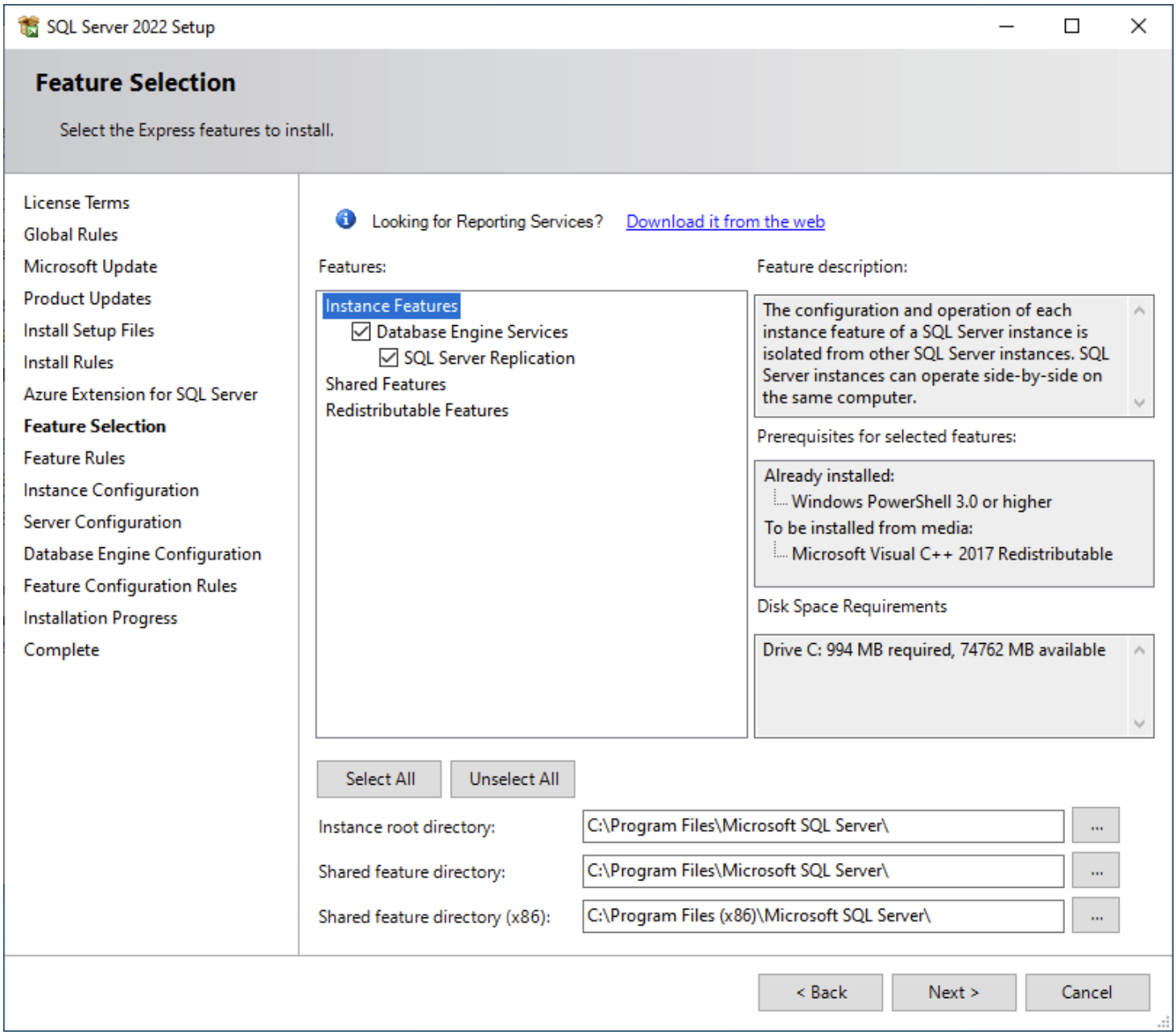
To install SQL Server, select *'New SQL Server standalone installation'*.



On the next screen, disable Azure Extension as shown below.

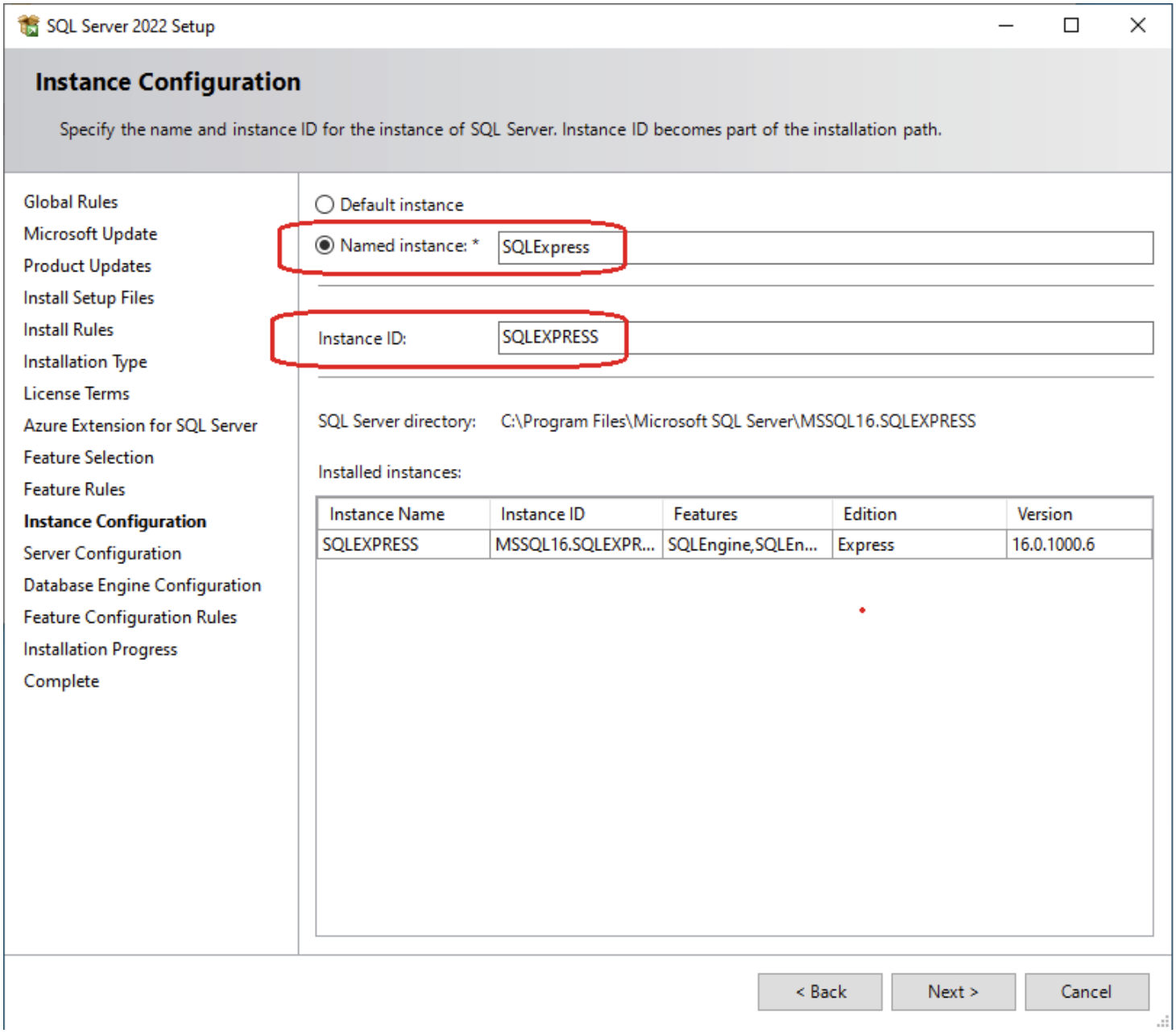


Select all necessary components from group Database Engine Service.
Specify Instance root dictionary if needed.

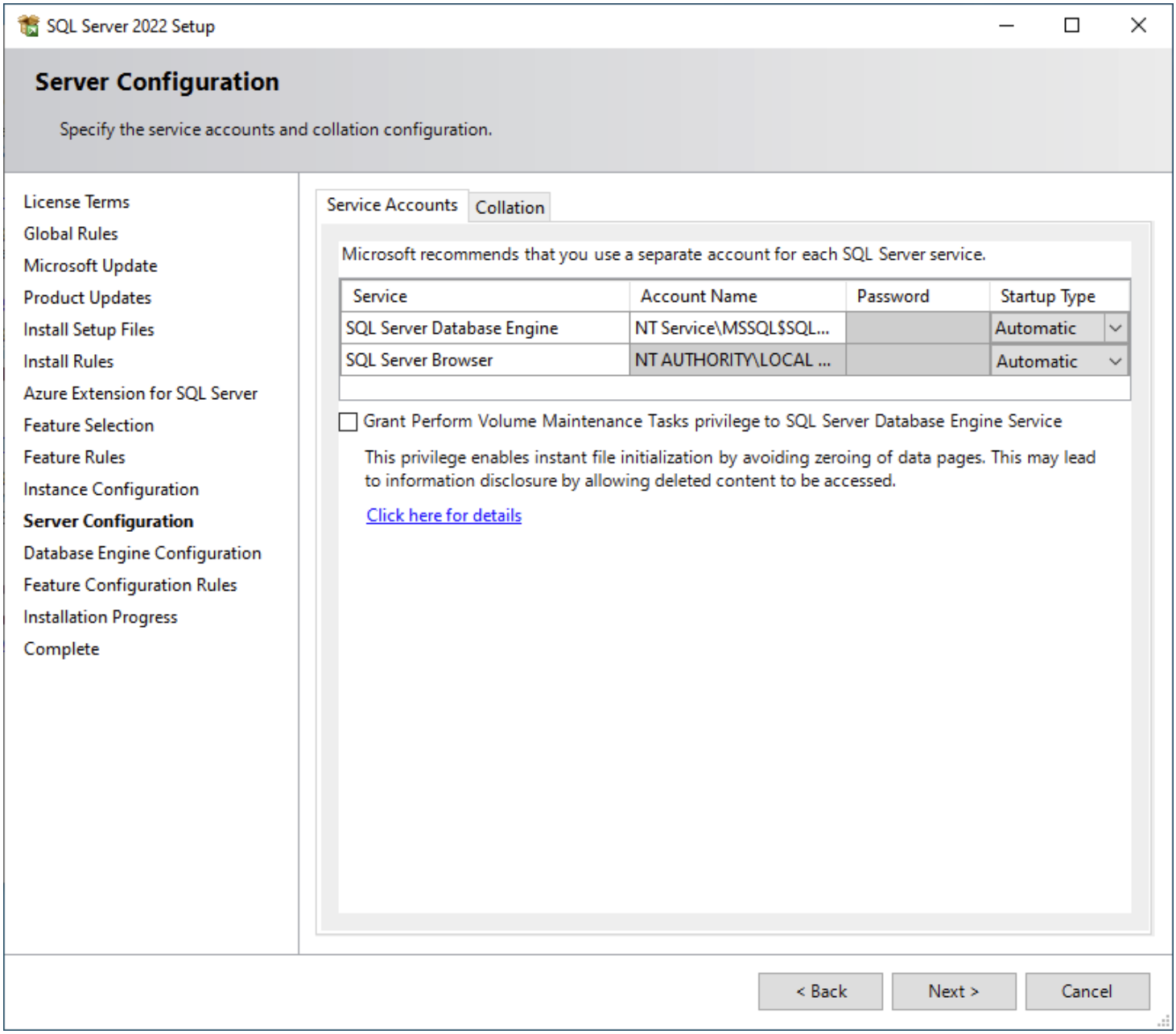


Configure and note down the Instance Name and Instance ID. They will be needed for connection configuration.
In this case:

- Instance Name: SQLExpress
- Instance ID: SQLExpress

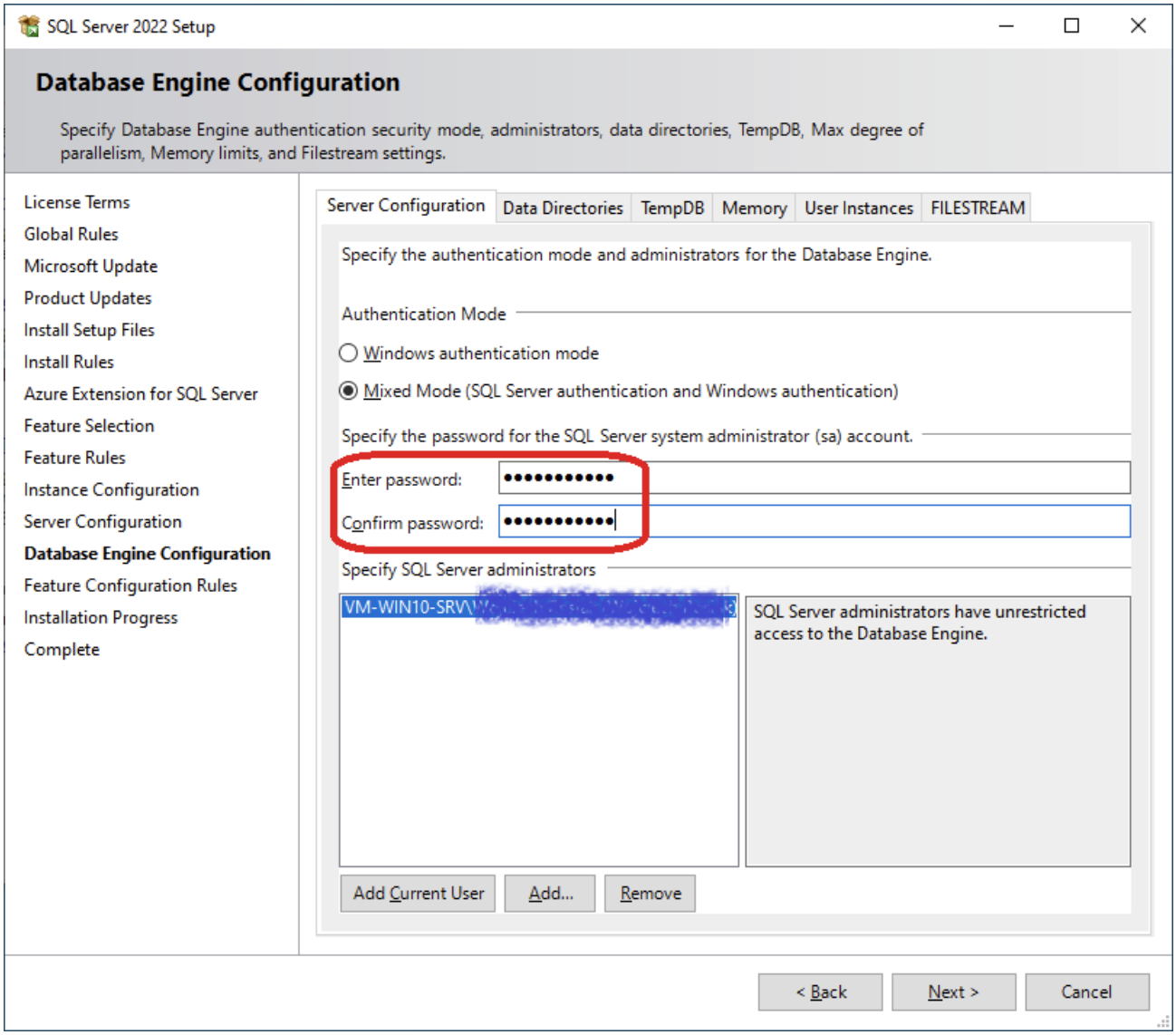


Make sure the server is started automatically by service account as shown on picture below.



For access configuration in Active Directory environment use Windows Authentication Mode; in other environments use Mixed Authentication Mode. Configure and note administrator password.

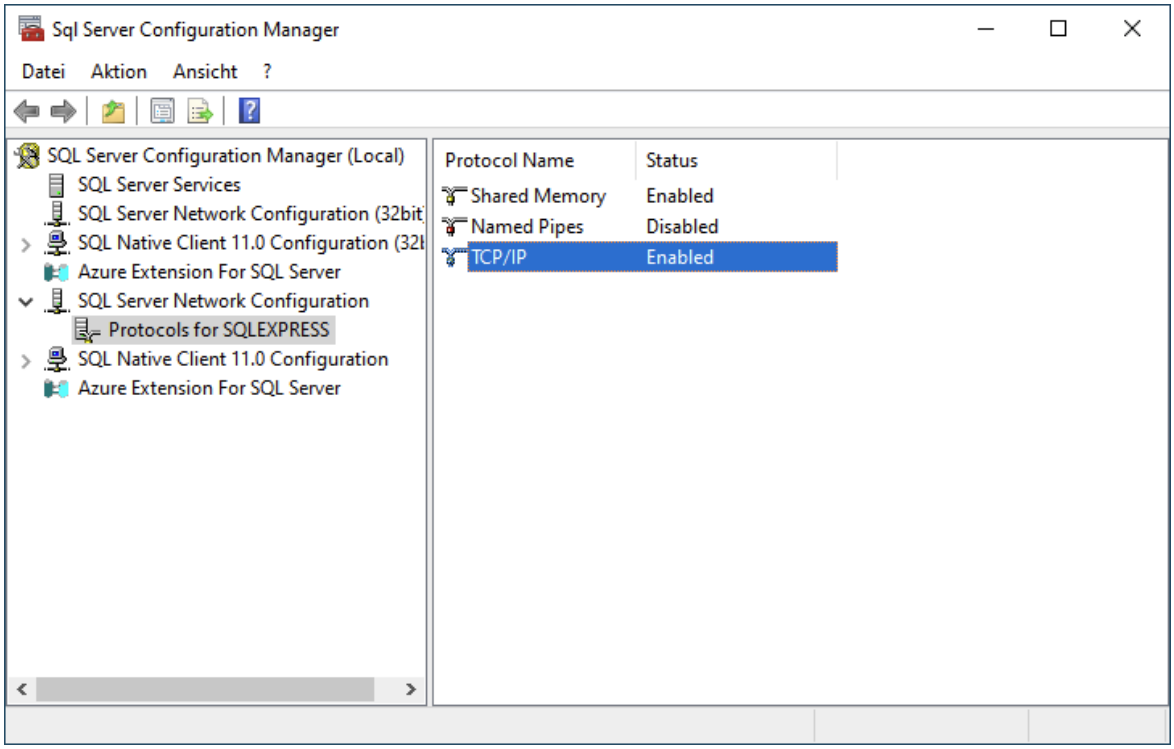
Such a configuration allows to login to the server using username Administrator and the configured password as well as access using Windows account defined as Server administrators.



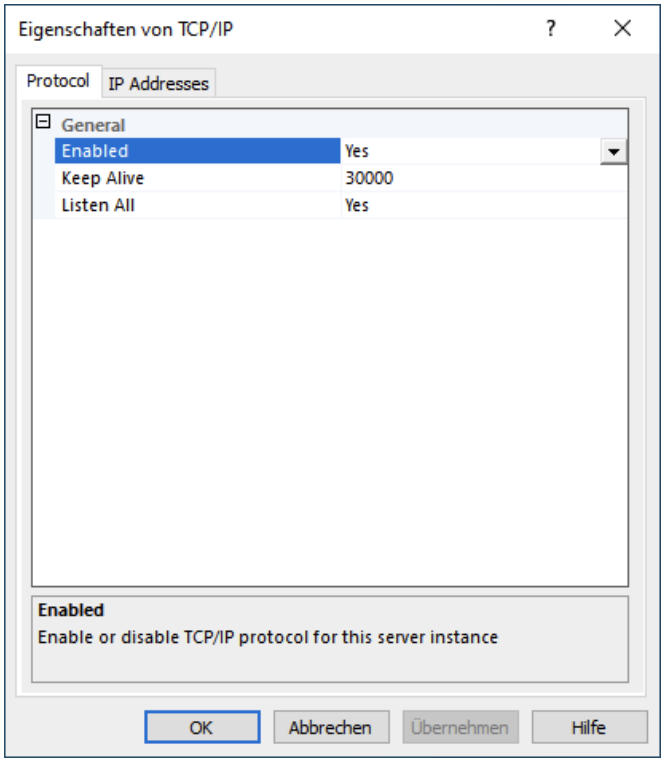
3.1.1 Enable Network access

On the Standard or Enterprise Edition of the SQL-Server the network access is enabled by default. For the Express edition, you need to activate it.

To do that, open the Sql Server Configuration Manager, go to *'Sql Server Network Configuration'* → *'Protocols for SQLEXPRESS'*

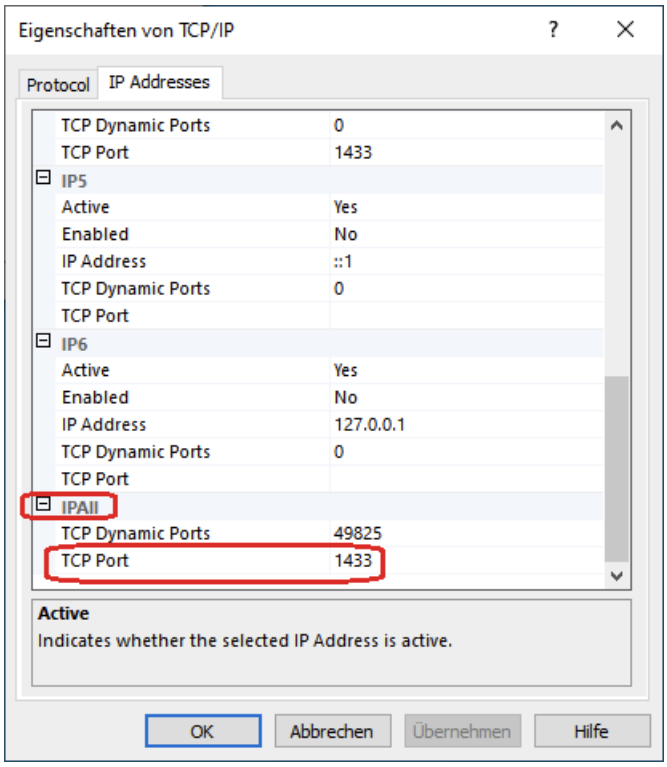


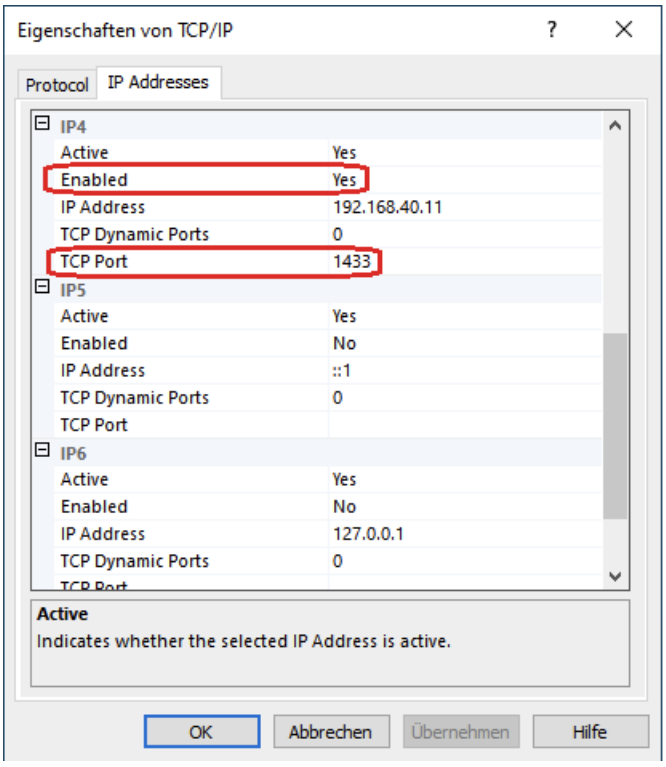
and double click 'TCP/IP' list item. Set 'Enabled' item to 'yes'



Next, go to tab "IP Addresses". Go to bottom of the list to item ""IPAll" and set port to 1433.

Afterwards set "Enabled" to "Yes" for each network interface that should listen for sql requests:





After saving, restart SQL Server.

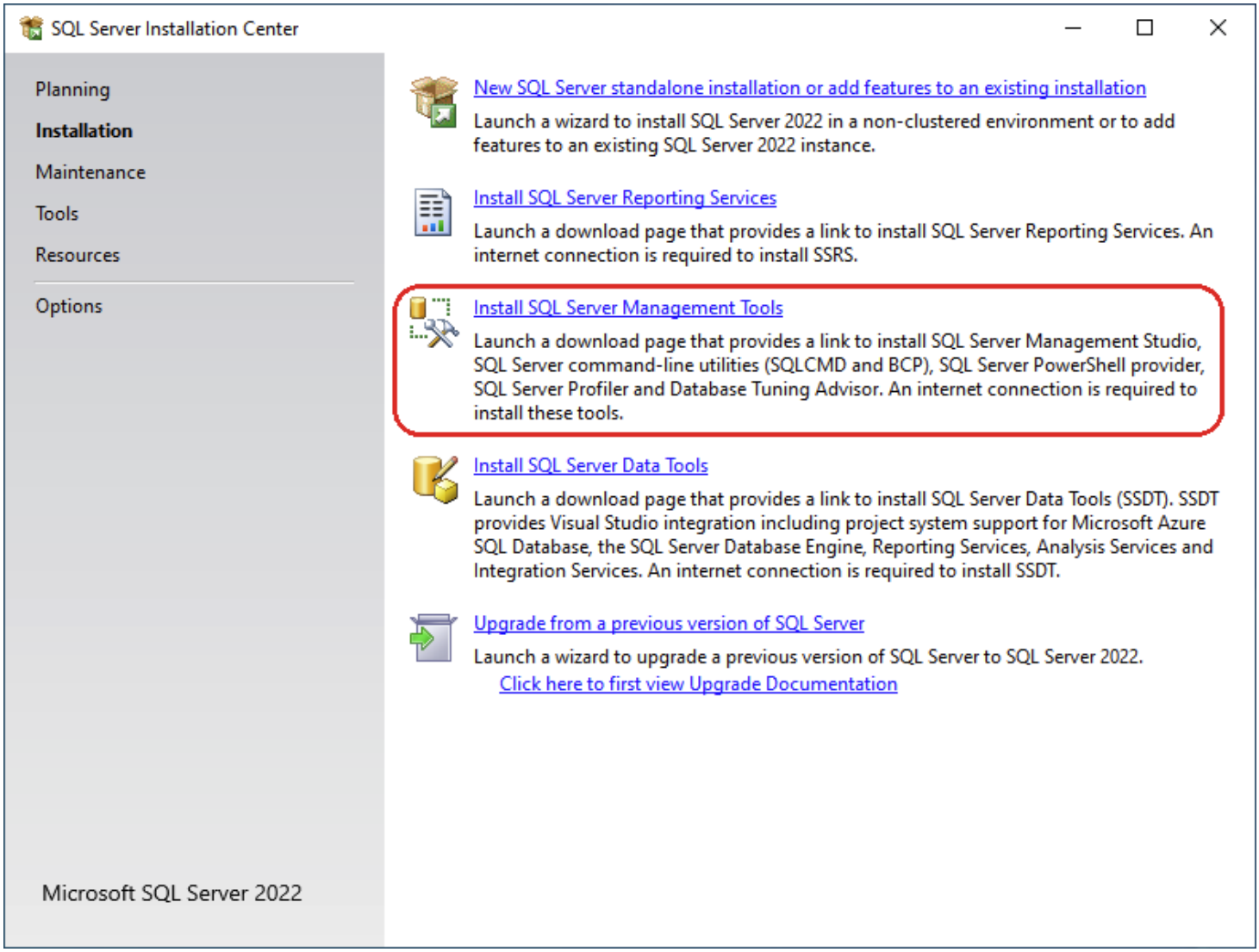
3.1.2 Configure firewall

Run Windows PowerShell as Administrator and execute following commands:

New-NetFirewallRule -DisplayName "SQLServer default instance" -Direction Inbound -LocalPort 1433 -Protocol TCP -Action Allow
New-NetFirewallRule -DisplayName "SQLServer Browser service" -Direction Inbound -LocalPort 1434 -Protocol UDP -Action Allow

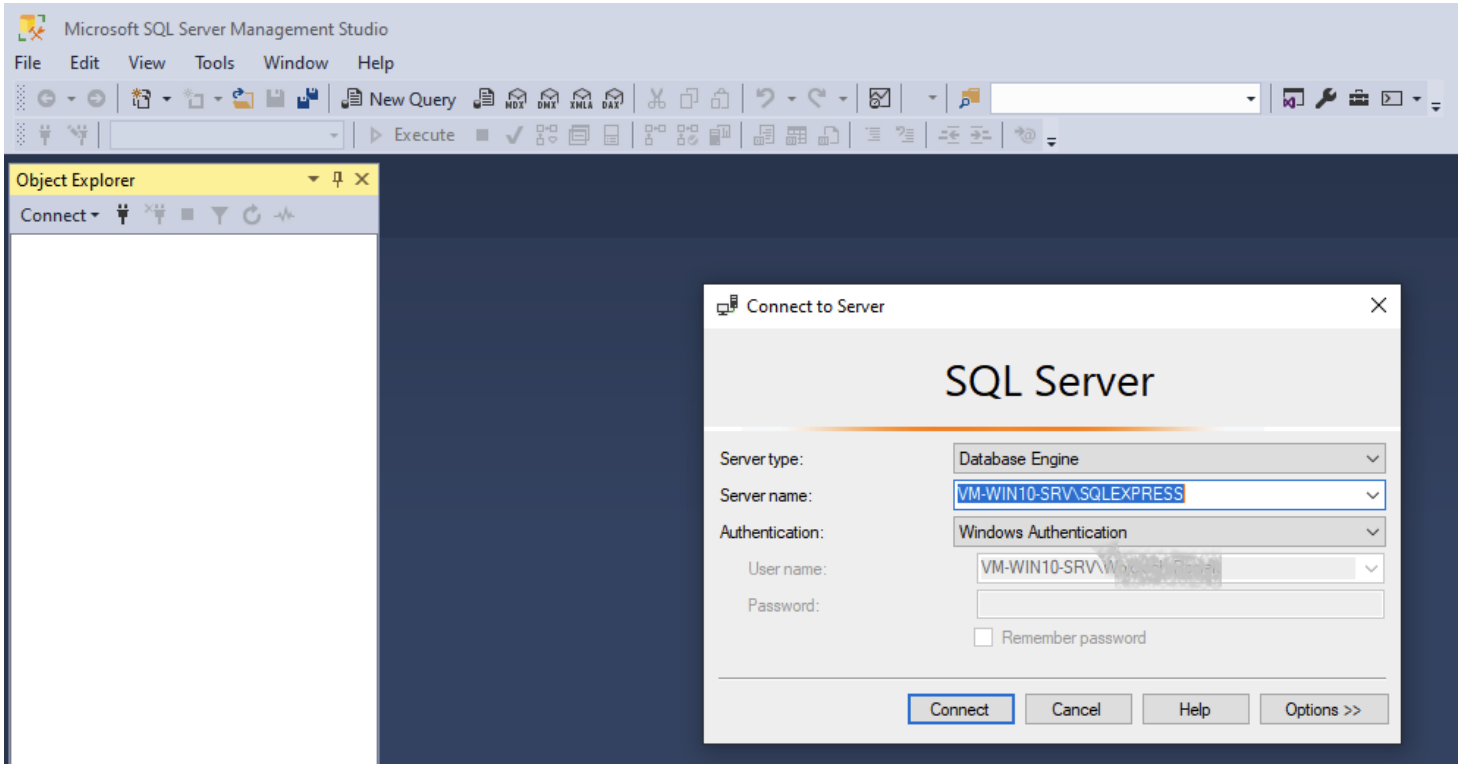
3.2 Installation of SQL-Server Management Studio

Start "SQL Server Installation Center" and select "Install SQL Server Management tools".



A web browser will be started with a page where you can download the "SQL Server Management Studio (SSMS)". Download it and start the setup. It is not necessary to install it on the SQL Server machine.

After installation, you can start the management studio, set the server name and connect.



3.3 Database preparation

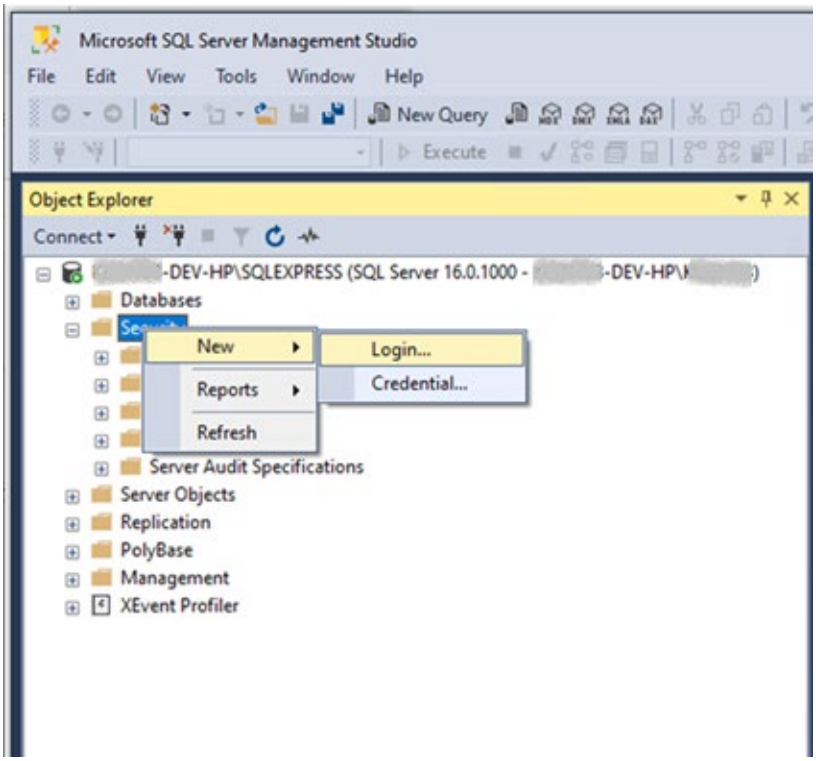
The current implementation allows two possible database configurations.
The first one, using an account with table creation rights (i. e. Administrator), allowing the Configuration Manager to create database automatically, is **not recommended**.
Better create user and database for Configuration Manager manually, using SQL Server Management Studio.

In both cases users can be Active Directory managed or database managed, in this how to we focus on creation of the user and table for the Configuration Manager.

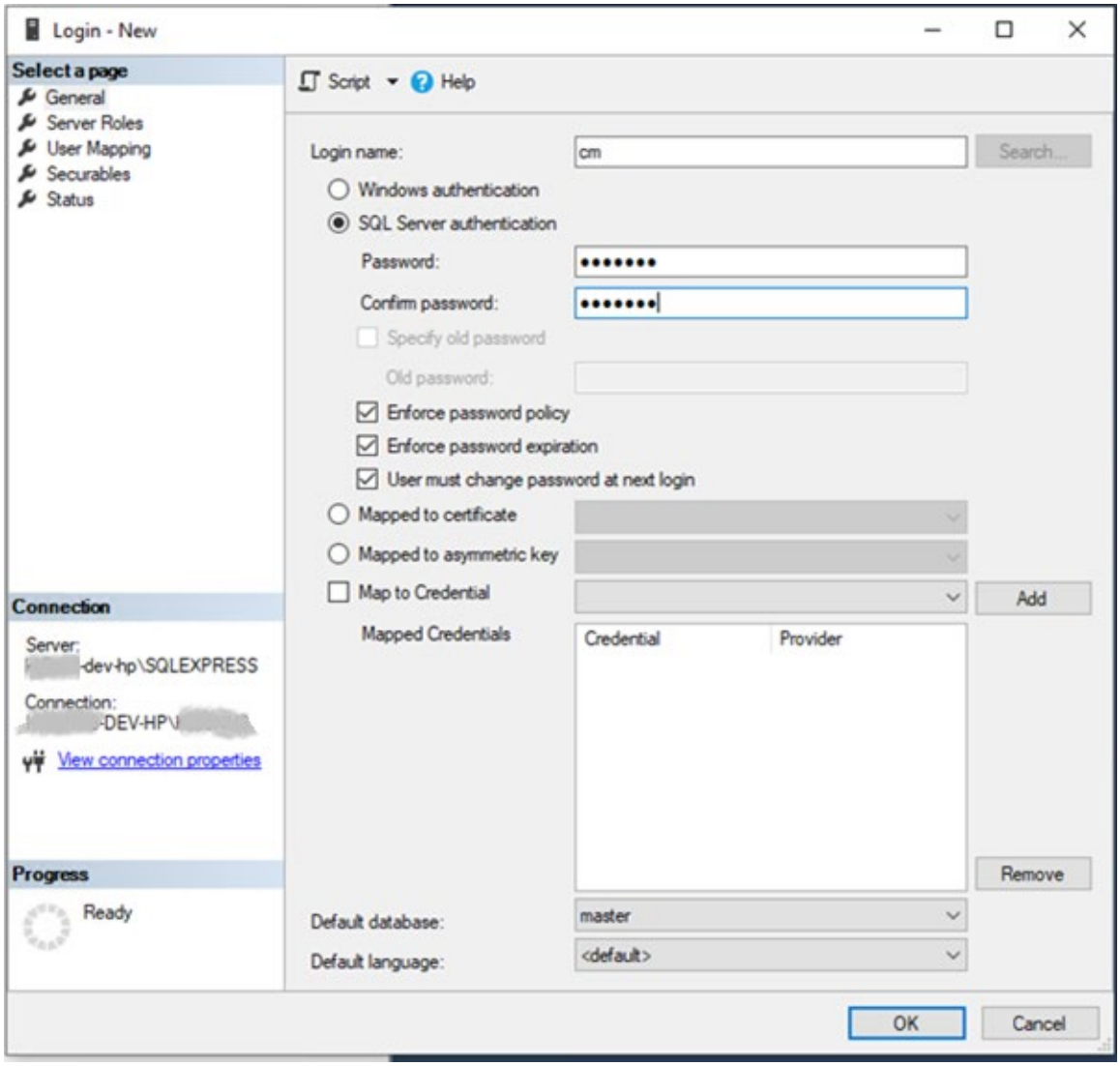
Using the SQL Server Management Studio select correct instance, configure administrator access and connect.

3.3.1 User creation

Right click on directory Security, then select New→Login...

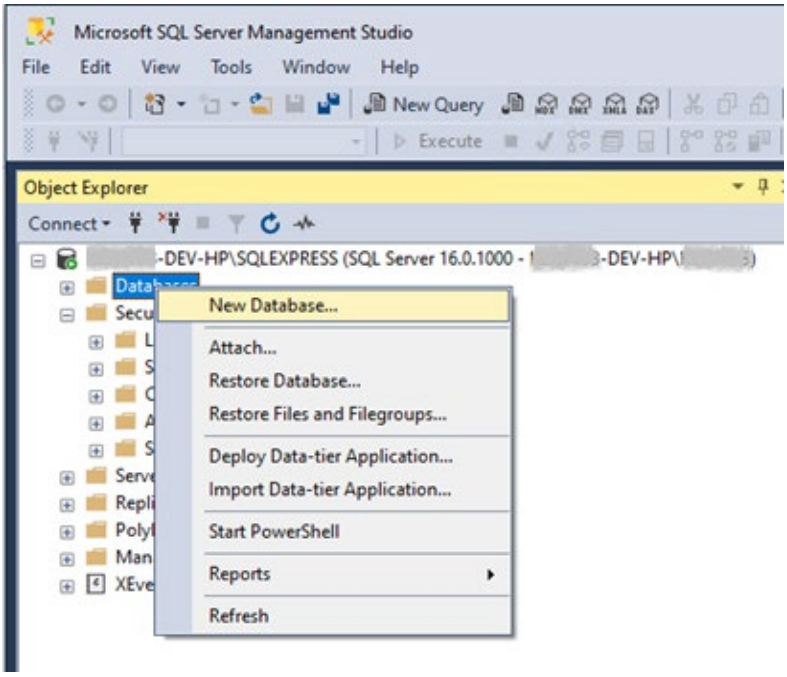


Set login name to desired value, for authentication type select SQL Server authentication, define password.

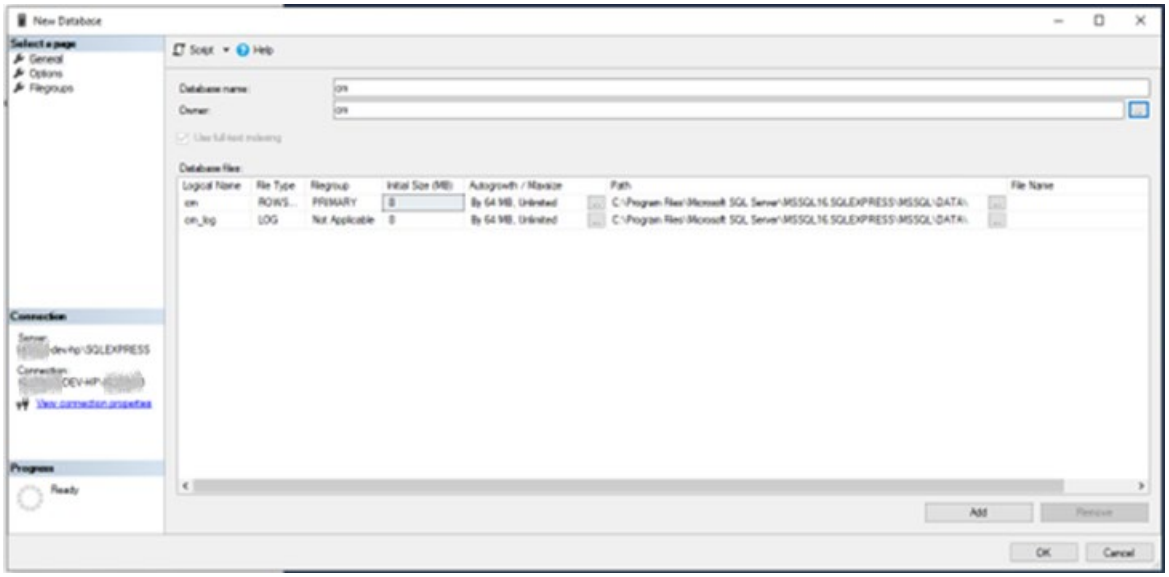


3.3.2 Database creation

Right click on directory Databases, then select New Database...



Define the desired database name (in this example cm) and select the owner (the user created before).
The path to the database can be set as well.



3.4 Using built-in column encryption.

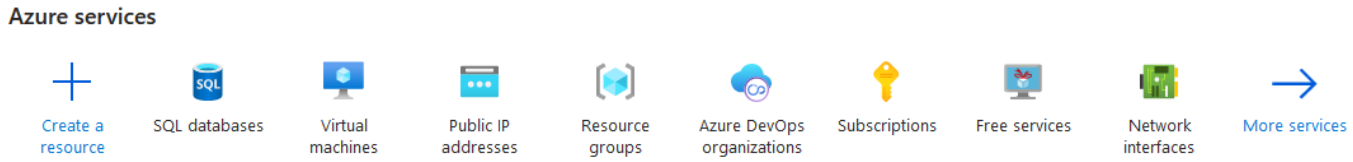
MS SQL offers built-in columns encryption based on PKI.

To use the feature, unset column password in Configuration Manager and use an enhanced connection string containing *'Column Encryption Setting=enabled;'*.

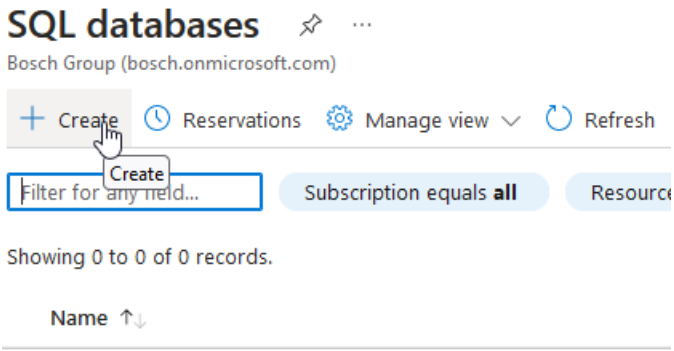
Follow the tutorial to configure it correctly in the database: https://sqlsolutionsgroup.com/sql-server-always-encrypted/.

3.5 Using Cloud Database - Azure

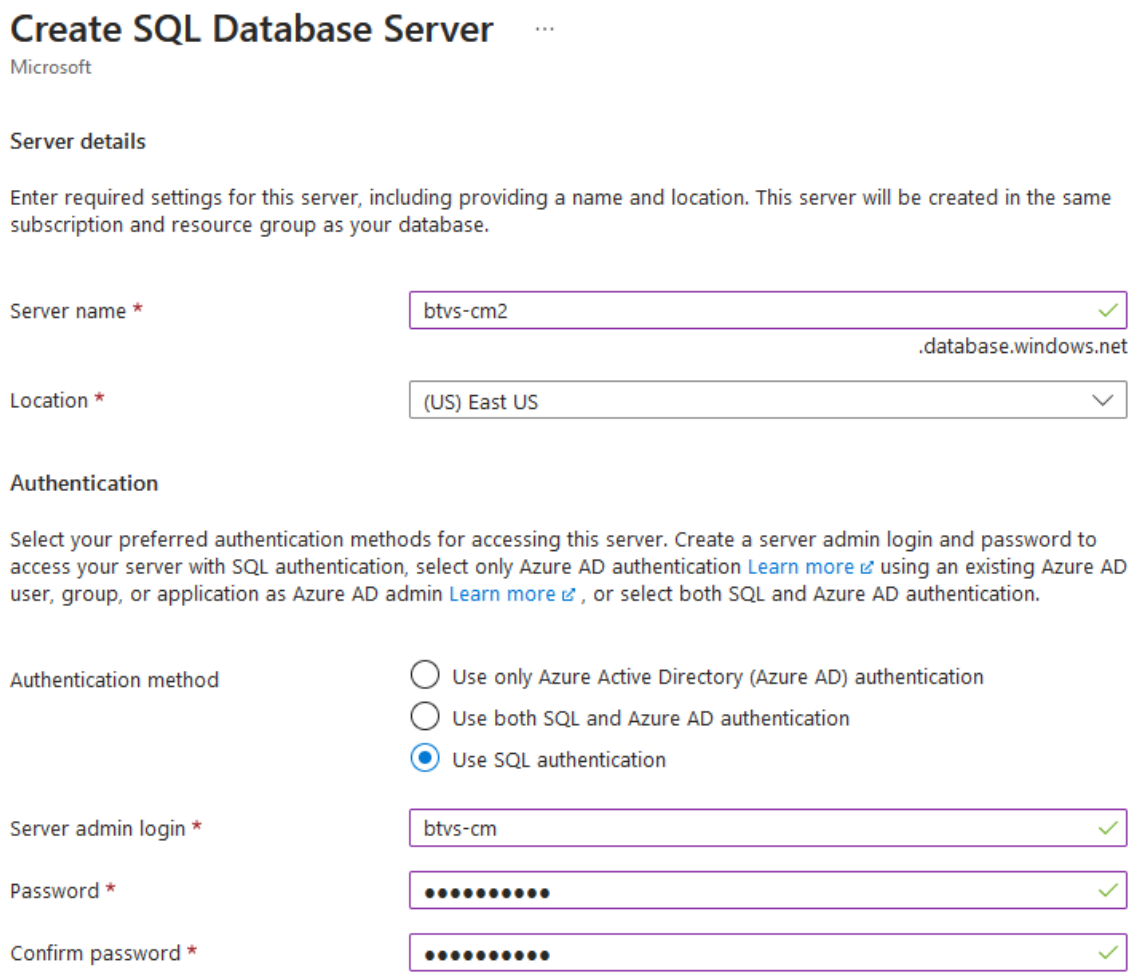
After logging into Azure portal, select SQL databases.



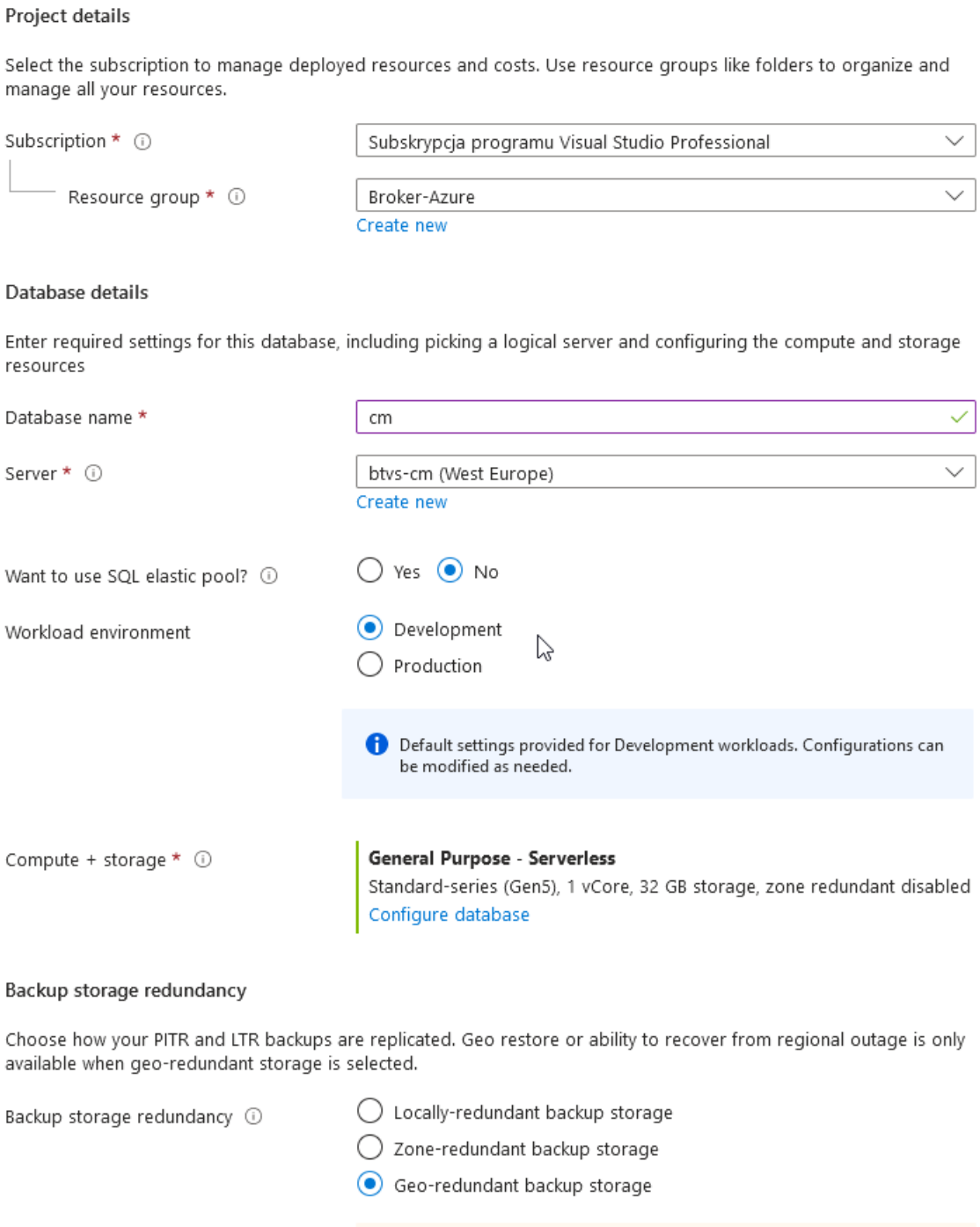
Select create.



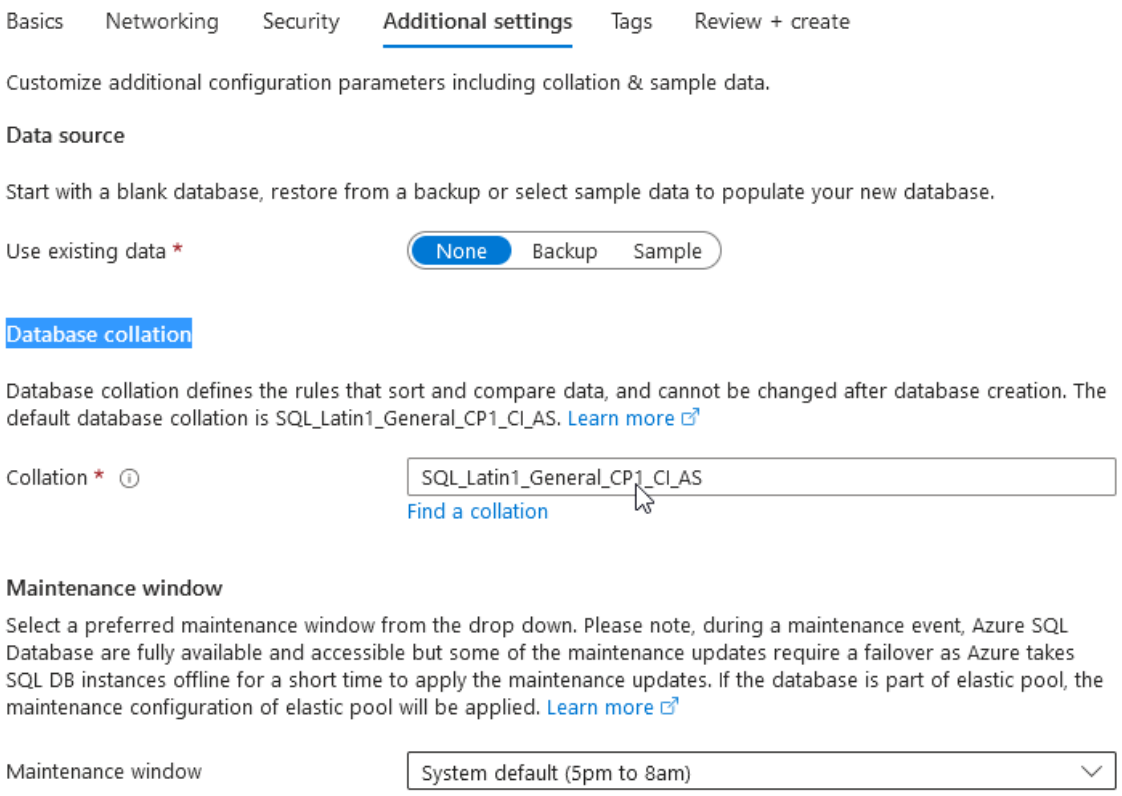
Create Server if needed:



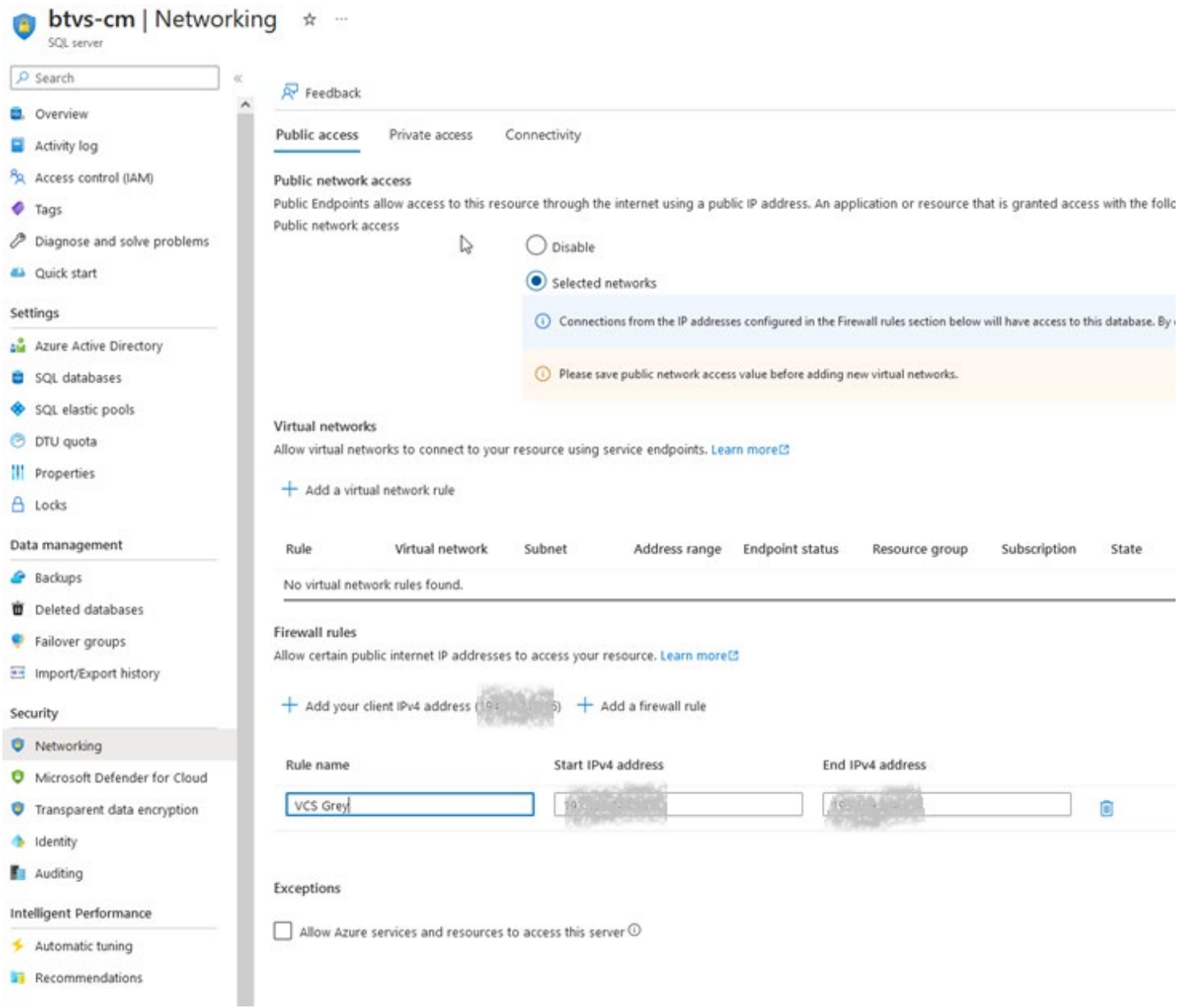
Configure database name and server used:



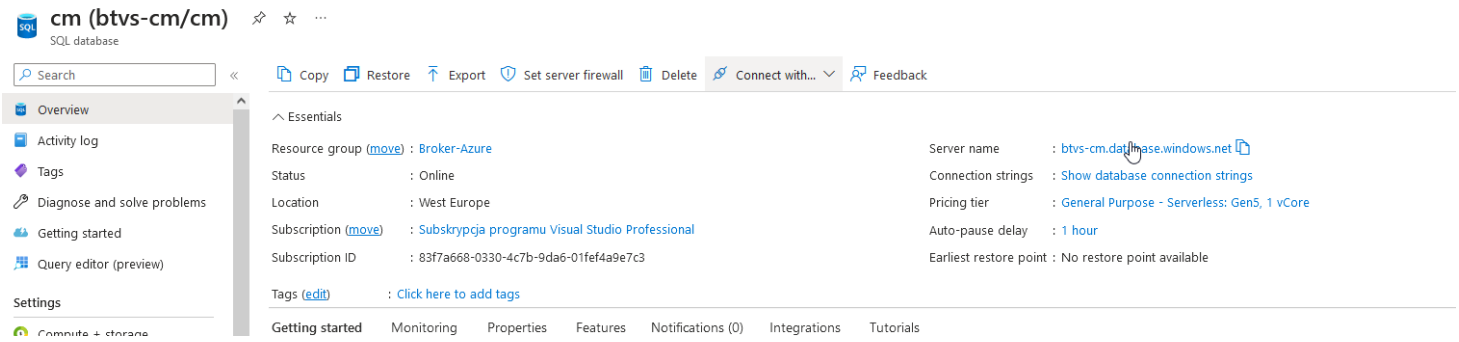
Configure Database collation:



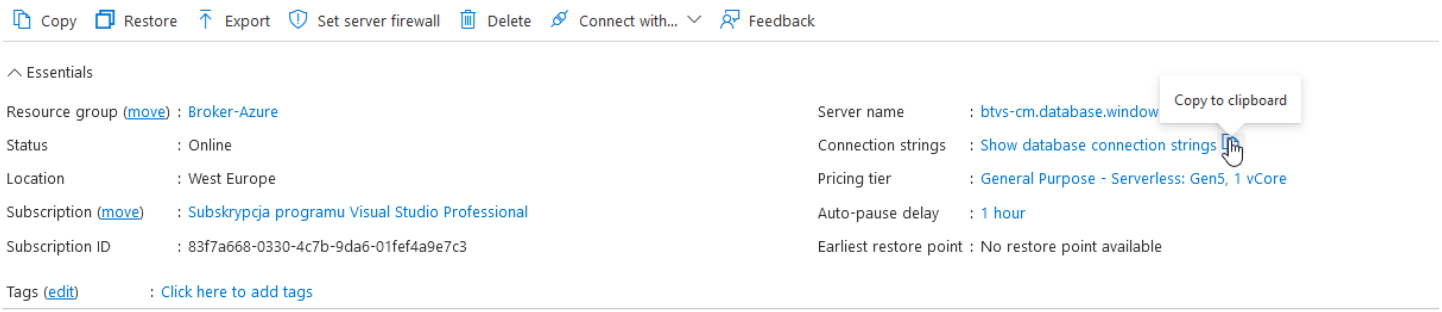
Navigate to created server. Select Security → Networking.
Set Public network access to Selected networks.
In firewall rules add client's public IP.



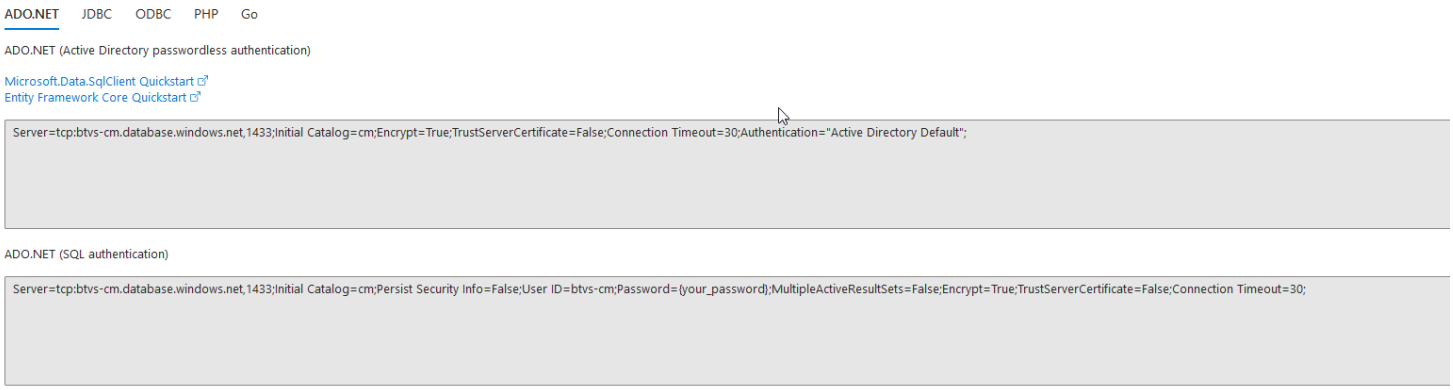
Navigate to created database:



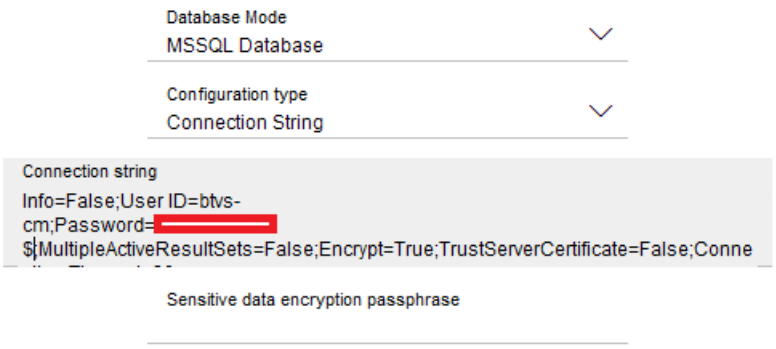
Select Show database connection strings:



Copy ADO.NET (SQL authentication) connection string.



Set Connection string configuration mode in Configuration Manager, paste copied connection string, change password ({your_password}) used in connection string.



4 References

Limitations from database perspective are described by Microsoft. There are many of them depending on DB version, or OS version (e. g. CPU cores need licenses). More details can be found in MS documents:

1. https://learn.microsoft.com/en-us/sql/sql-server/maximum-capacity-specifications-for-sql-server?view=sql-server-ver16

2. https://learn.microsoft.com/en-us/sql/sql-server/compute-capacity-limits-by-edition-of-sql-server?view=sql-server-ver16

There are reference documents that may help users to set up an SQL database:

3. https://learn.microsoft.com/en-us/sql/relational-databases/security/authentication-access/create-a-database-user?view=sql-server-ver16

4. https://learn.microsoft.com/en-us/sql/relational-databases/security/authentication-access/create-a-database-schema?view=sql-server-ver16

5. https://learn.microsoft.com/en-us/sql/relational-databases/security/authentication-access/create-a-login?view=sql-server-ver16

Bosch Sicherheitssysteme GmbH
Robert-Bosch-Ring 5
85630 Grasbrunn
Germany
www.boschsecurity.com