How To Monitor Microsoft SQL With Nagios XI 5

Purpose

This document describes how to monitor Microsoft SQL with Nagios XI.

Note: If you are using **Nagios XI 2024**, please refer to the updated document.

Terminology

MSSQL has several components that require configuration to allow Nagios XI to monitor it. The steps that are required differ depending on:

- Database engine is running as a Named Instance
    - Multiple instances of MSSQL can be installed on the same server but will be listening on separate network ports (normally dynamic)
    - The **SQL Server Browser** service will provide information about the instances installed (like the network port) when receiving requests on UDP port 1434
    - When using the MSSQL wizards, if you define an instance, you do not provide the port
- Database engine is configured to use a specific TCP port
    - The default instance of MSSQL commonly runs on TCP port 1433
    - This or any other instance can be configured to listen on a specific port
    - When using the MSSQL wizards, if you define a port, you do not provide the instance name
- Database monitoring user account
    - You need to create a user account in the MSSQL instance to allow Nagios XI to connect
    - This account can use SQL authentication or Windows authentication with MSSQL
    - It is strongly recommended that you don't use the **sa** or administrator account for this purpose
- Database engine authentication method
    - SQL authentication
        - Is a local user account specific to the MSSQL instance
    - Windows authentication
        - Maps a Windows user account to an internal MSSQL user
- Windows firewall rules to allow inbound traffic

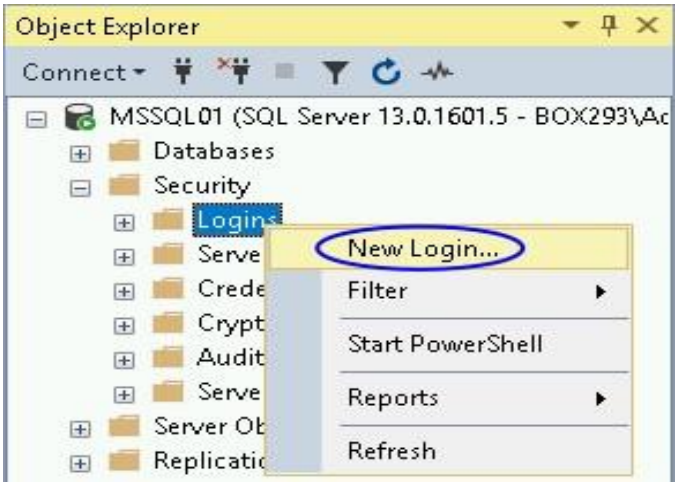
- The MSSQL server will need firewall rules to allow the incoming network traffic

Create Monitoring User Account

The best practice for monitoring is to create a user account in the MSSQL instance that will be used by Nagios XI to connect. Even when using Windows Authentication, you will need to create an account in MSSQL that is linked to this account. It is advisable that your Windows or MSSQL account is not allowed to expire, otherwise this will cause monitoring issues when it eventually does expire.

On your MSSQL server open **SQL Server Management Studio** and connect to your instance as a user with administrative rights.

- Expand **Security** and select **Logins**.
- Right click on **Logins** and select **New Login**
- The **Login - New** window will appear.



Depending on your authentication method your choices will be slightly different:

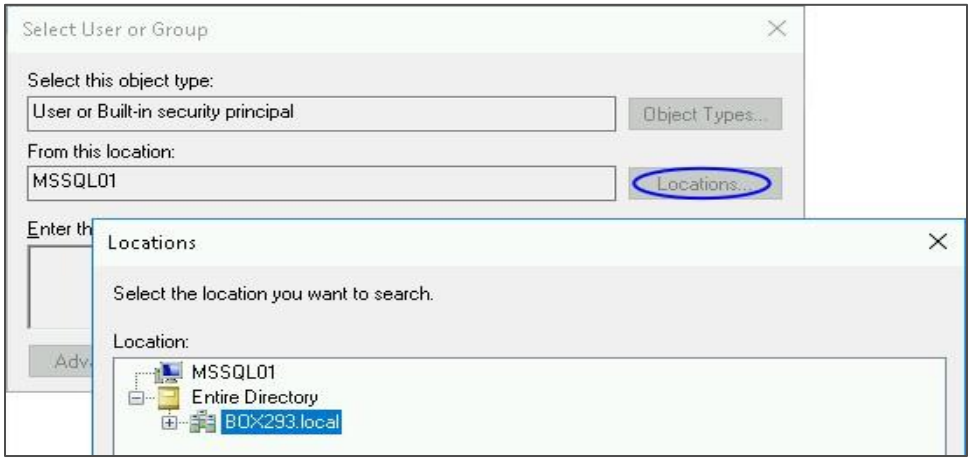
- Windows Authentication
- SQL Authentication

Windows Authentication

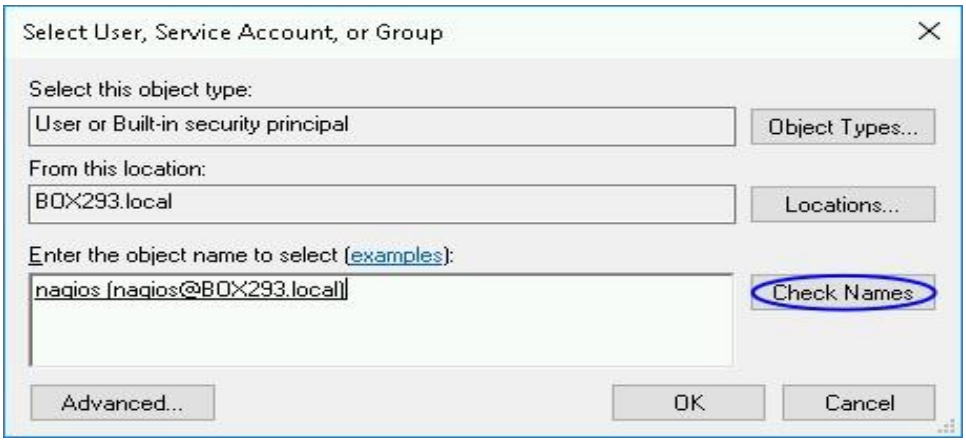
Select **Windows authentication** and then click the **Search** button.

You will need to use the **Locations** button to define the scope of the Windows user account. The default scope is the local server, if you want to use a domain account use the Locations button.

In the screenshot to the right, you can see the `BOX293.local` domain was selected.

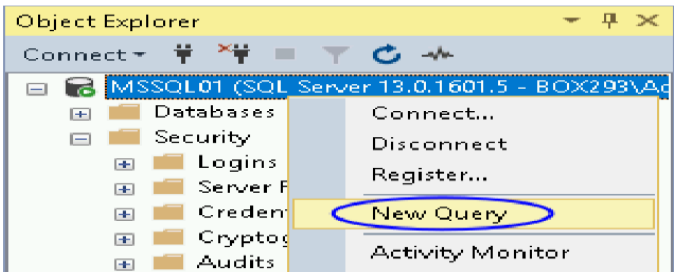


Type the name of the account and then click the **Check Names** button. Your user account should be found. Click **OK** to select the account.



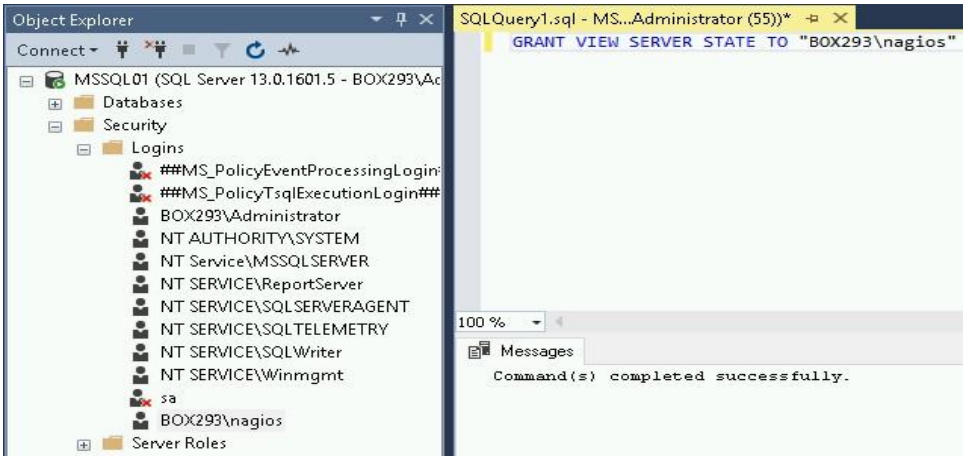
In the **Login - New** screen, you can see the **Login name** field is now populated. All the required fields have been populated, click the **OK** button.

The **VIEW SERVER STATE** permission needs to be granted to the new user account. Right click the server at the top and select **New Query**.



You will need to type the following in the query window:

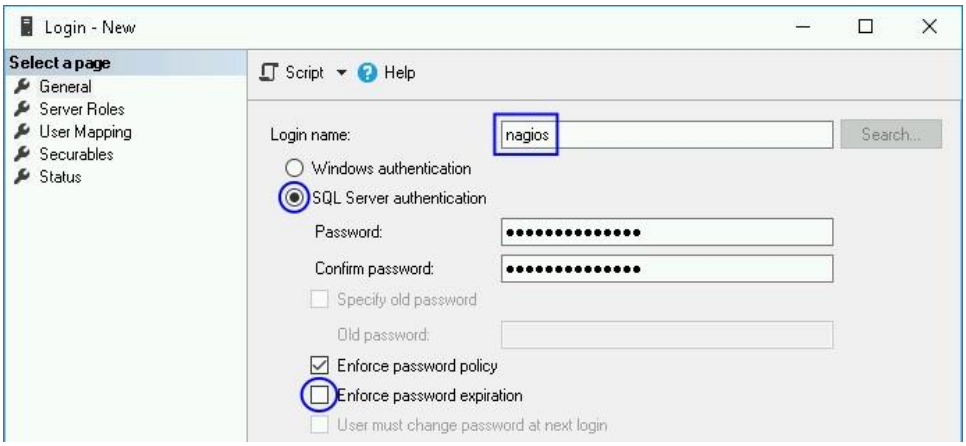
```
GRANT VIEW SERVER STATE TO
"<username>"
```



In the screenshot you can see `"BOX293\nagios"` was provided, it needs to match the Login you can see in the left pane. Press the **F5 key** on the keyboard to execute the query. You should receive the message "`Command(s) completed successfully`" in the messages window. You can now close the query window, when prompted to save changes answer **No**.

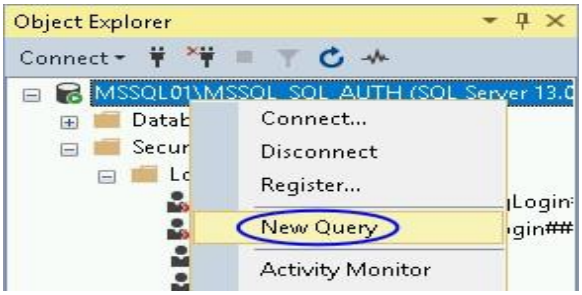
You can now proceed to the Assign Monitoring Account section of this document.

SQL Authentication



- Provide a **Login name**.
- Select **SQL Server** authentication.
- Provide a password.
- Un-check the **Enforce password expiration** checkbox.
- Click **OK** to create the account.

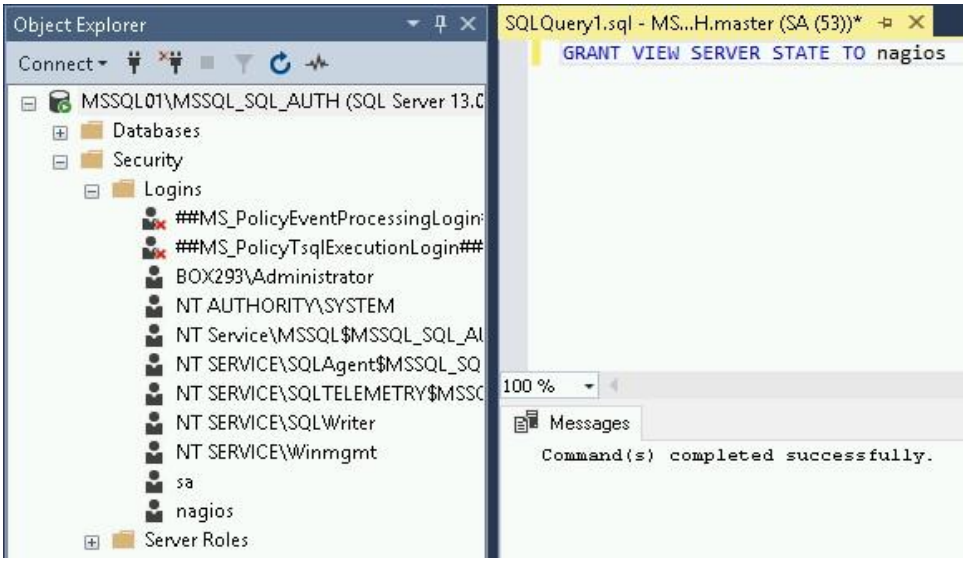
The **VIEW SERVER STATE** permission needs to be granted to the new user account. Right click the server at the top and select **New Query**.



You will need to type the following in the query window:

```
GRANT VIEW SERVER STATE TO
<username>
```

In the screenshot you can see ***Nagios*** was provided, it needs to match the Login you can see in the left pane. Press the **F5** key on the keyboard to execute the query. You should receive the message "Command(s) completed successfully" in the Messages window. You can now close the query window, when prompted to save changes answer **No**.

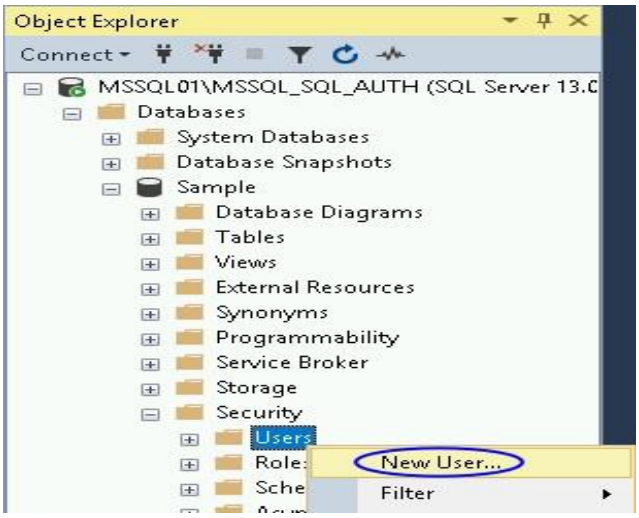


You can now proceed to the Assign Monitoring Account section of this document.

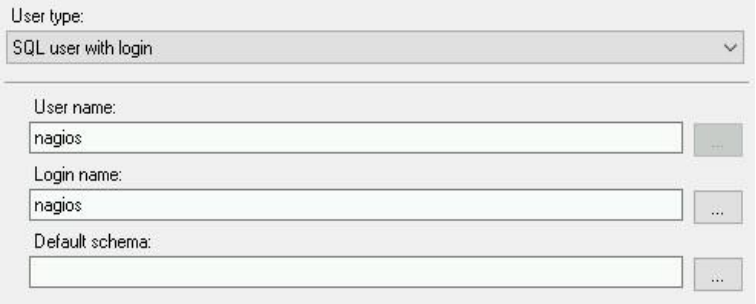
Assign Monitoring Account

Now that a monitoring user account has been created it needs to be assigned to the databases you want to monitor.

- This example will use a database called **Sample**.
- Expand Databases > Sample > Security and select Users.
- Right click **Users** and select **New User**.
- The **User** type will be **SQL user with login** regardless of which authentication method you have chosen.

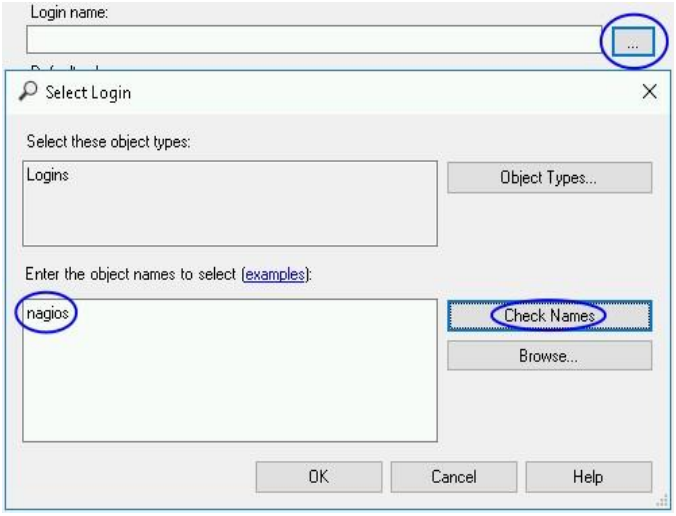


- Provide a **Username**.
- For the **Login name** click the **…** button.
- Type the name of the user and click the **Check Names** button.
- You may be prompted to select the correct user, click the **OK** button once you have a valid name.



- Once you have populated these fields click the **OK** button.

You will now see the user appear in the list of users. This completes the steps required for creating a user account for monitoring MSSQL.



MSSQL Network Ports And Firewall Rules

If your MSSQL server has the Windows firewall enabled, you will need to create firewall rules to allow inbound traffic from the Nagios XI server.

If your Nagios XI server and MSSQL server are on separate subnets, the router(s) that connect these subnets may have firewall rules in place. These router(s) will also need firewall rules to allow traffic from the Nagios XI server to the MSSQL server.

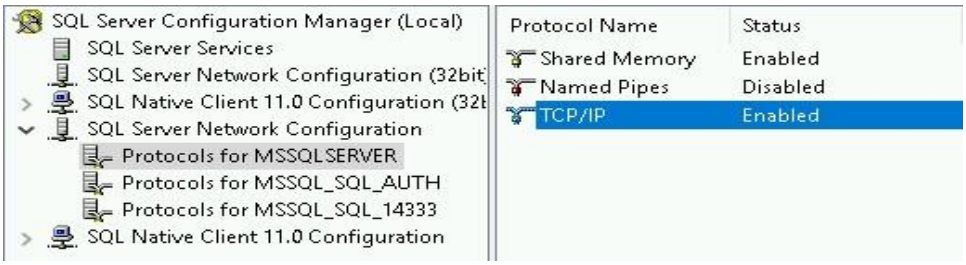
If you have multiple MSSQL instances on the same server then you will need to configure these instances to run on dedicated network ports.

- If you do not have the Windows firewall enabled, then this is not required. The SQL Browser service will inform the Nagios XI server which port to communicate on
- If your Nagios XI server and MSSQL server are on separate subnets, the router(s) connecting these subnets may have firewall rules in place. If this is the case, then the MSSQL instances will need to be configured to run on dedicated network ports.
- The same applies if you have SQL Express Edition

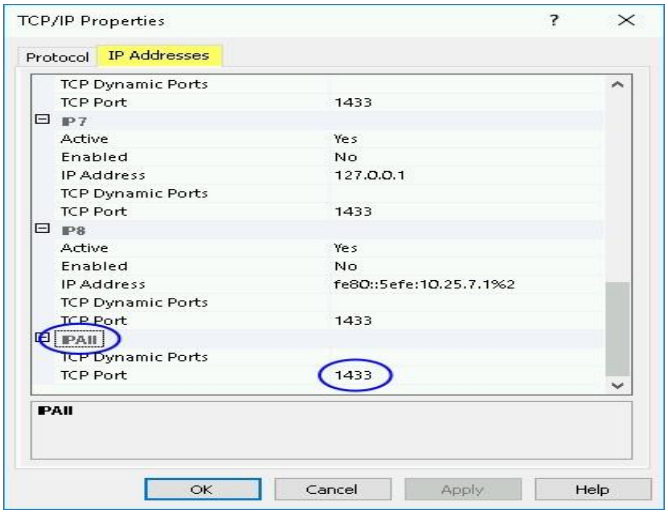
Identify MSSQL Network Port

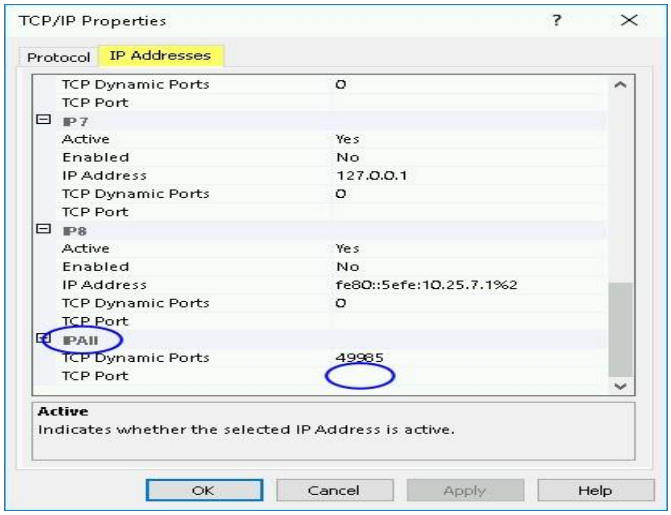
The first step is to identify the network port the MSSQL server is configured to run on. On your MSSQL server open the **SQL Server Configuration Manager**.

Select **SQL Server Network Configuration** and you will see a list of Protocols for the instances installed on your MSSQL server (this screenshot has three instances).



- Double click on one of the instances and then double click on **TCP/IP**.
- Click the **IP Addresses** tab and scroll down to the **IPAll** section.
- On the left screenshot you can see the port is **1433**.
- On the right screenshot you can see there is no port, however the **Dynamic Port** field is populated.





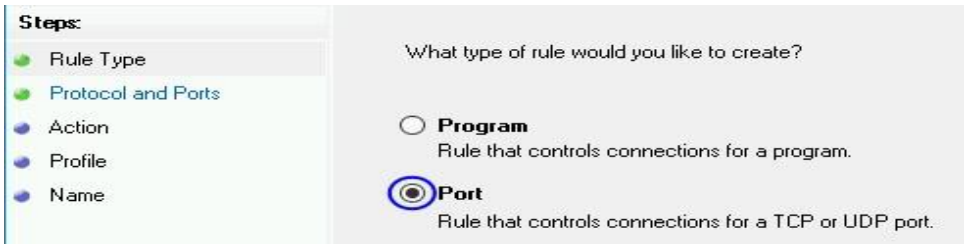
To change this to a fixed port:

- Clear the **TCP Dynamic Ports** field
- Type the port number you want to use in the **TCP Port** field
- Click **OK** and restart the SQL Server service (under SQL Server Services in the left pane)

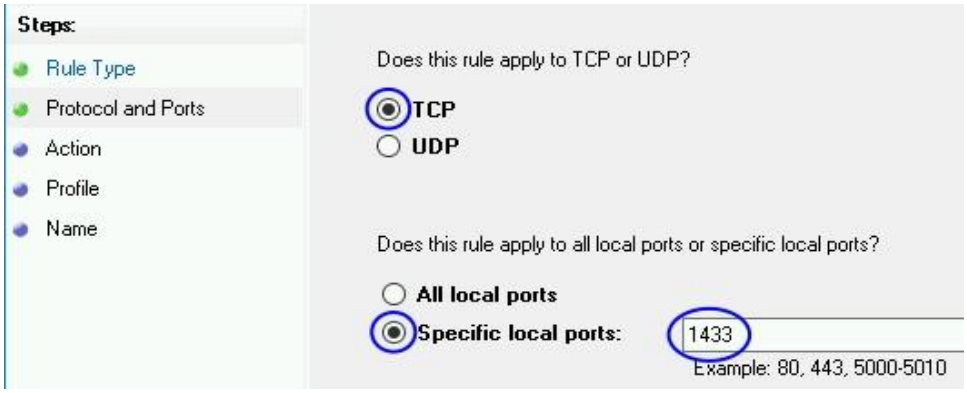
Create Firewall Rule

The next step is to create the network firewall rule. The following example will create a rule for TCP **1433**. Open **Windows Administrative Tools > Windows Firewall** with **Advanced Security**. In the left pane select **Inbound Rules**. Right click **Inbound Rules** and select **New Rule**

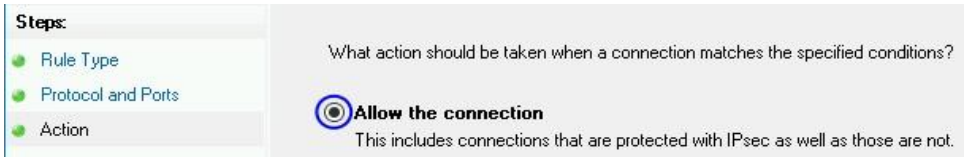
- Select **Port**
- Click **Next**



- Select **TCP**
- Select Specific local ports and type the port number
- Click **Next**

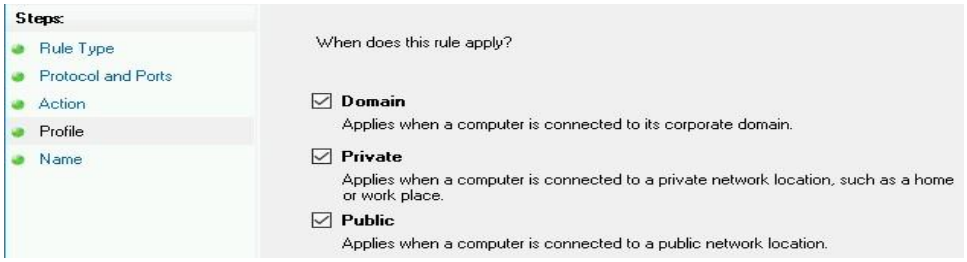


- Select **Allow the connection**
- Click **Next**



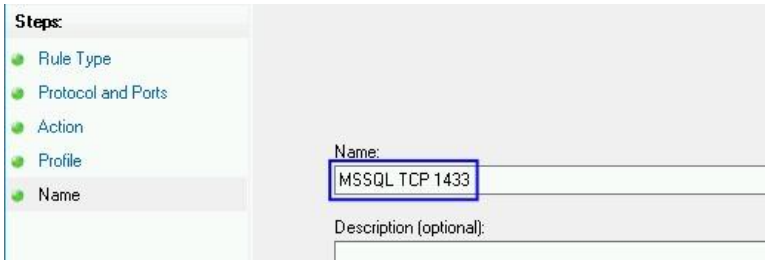
Make sure **Profile** selections meet your requirements.

Click **Next**



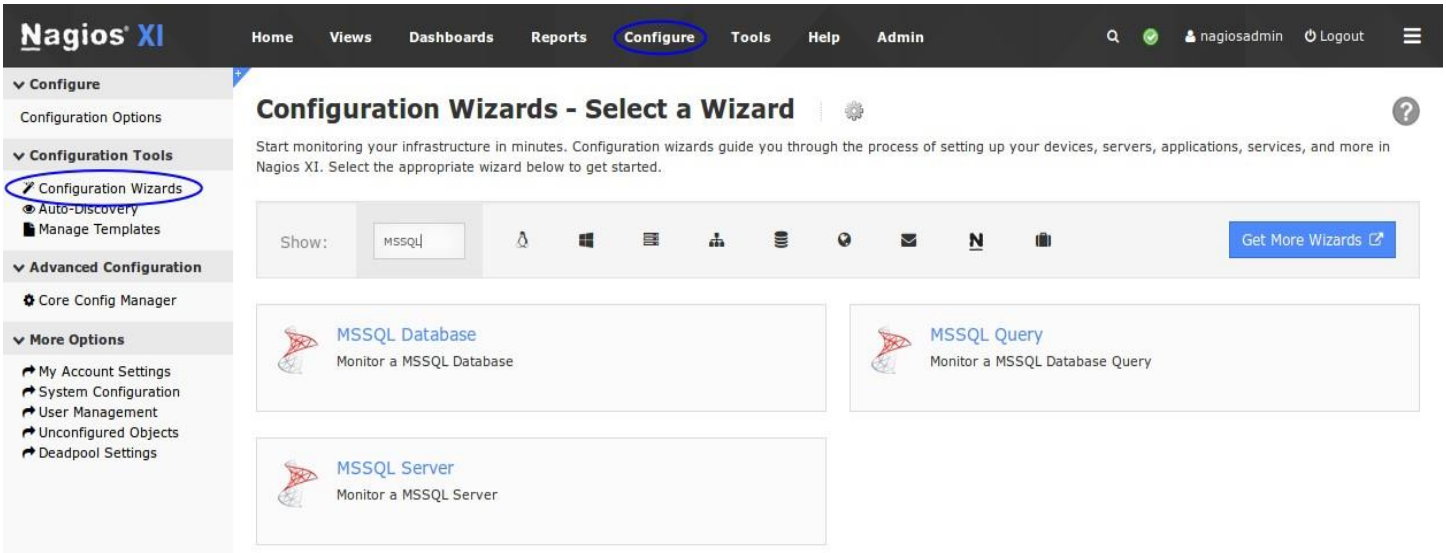
Provide a **Name** and optionally a description.

Click **Finish.**



Running The Configuration Wizards

This documentation will now explain the configuration wizards. In Nagios XI navigate to **Configure** > **Configuration Wizards** and select the **MSSQL** wizard of your choice. In the following screenshot you can see how the search field allows you to quickly find a wizard.



Step 1:

Each of the wizards has the same options, what you select here depends on your MSSQL instance configuration and firewall settings.

- **Address**
  - This is either the IP address or FQDN DNS record of the MSSQL server
  - Avoid using a flat name record like `mssql01`, use the FQDN like `mssql01.box293.local`
- **Instance**
  - Referencing the instance allows you to connect without defining a port number
  - To use the instance name instead of a port:
    - The SQL Browser service needs to be running on the MSSQL server
    - If you have the Windows firewall enabled:
      - A firewall rule must be created for UDP 1434 to allow the SQL Browser Service to work
      - Each MSSQL instance needs to be configured on a dedicated network port

        - A firewall rule must be created for each MSSQL instance running
- **Port**
    - Referencing the port allows you to connect without defining an instance name
    - If you have the Windows firewall enabled:
        - Each MSSQL instance needs to be configured on a dedicated network port
        - A firewall rule must be created for each MSSQL instance running
- **Username**
    - This is the Windows username or SQL account required to connect, for example:
        - Windows authentication
            - `BOX293\nagios`
        - SQL authentication
            - `Nagios`
- **Password**
    - The password for the username supplied
- **Database**
    - The Database and Query wizards require the name of the database you wish to monitor

Step 2:

Each of the wizards will present a summary of the SQL Server details at the top of the page.

Make sure the **Host Name** field has a value that easily identifies this MSSQL Server.

Each of the wizards has different metrics that can be measured. The metrics available are clearly explained in the wizards and hence will not be covered here. The warning and critical thresholds can be defined as per the Nagios Plugin Development Guidelines, detailed information on this can be found on the following page: https://nagios-plugins.org/doc/guidelines.html#THRESHOLDFORMAT

Once you've finished selecting all the items you wish to monitor click **Next** and then complete the wizard by choosing the required options in **Step 3 - Step 5**. To finish up, click on **Finish** in the final step of the wizard. This will create new hosts and services and begin monitoring. Once the wizard applies the configuration, click the **View status details for** `<your device>` link to see the new host and services that were created.

Here are some examples from the different wizards:

MSSQL Database:

Host	Service	Status	Duration	Attempt	Last Check	Status Information
MSSQL01	Sample MSSQL Active Transactions	Ok	2h 9m 31s	1/5	2017-08-03 13:29:44	OK: Active Transactions is 0.0
	Sample MSSQL Connection Time	Ok	2h 9m 2s	1/5	2017-08-03 13:30:13	OK: Time to connect was 0.0101470947266s
	Sample MSSQL Database Size	Ok	2h 38m 25s	1/5	2017-08-03 13:30:28	OK: Database size is 8192.0KB
	Sample MSSQL Log File Usage	Ok	2h 8m 37s	1/5	2017-08-03 13:30:38	OK: Log File Usage is 6.0%
	Sample MSSQL Log Flush Wait Time	Ok	2h 8m 5s	1/5	2017-08-03 13:31:09	OK: Log Flush Wait Time is 15.0ms
	Sample MSSQL Log Growths	Ok	2h 7m 37s	1/5	2017-08-03 13:31:38	OK: Log Growths is 0.0
	Sample MSSQL Log Shrinks	Ok	2h 7m 8s	1/5	2017-08-03 13:32:07	OK: Log Shrinks is 0.0
	Sample MSSQL Log Truncations	Ok	2h 6m 39s	1/5	2017-08-03 13:32:36	OK: Log Truncations is 0.0
	Sample MSSQL Transactions / Sec	Ok	2h 6m 18s	1/5	2017-08-03 13:32:57	OK: Transactions Per Second is 0.0133315450842/sec

MSSQL Query:

Host	Service	Status	Duration	Attempt	Last Check	Status Information
MSSQL01	MSSQL Query - Test Query	Critical	2h 5m 10s	5/5	2017-08-03 13:32:24	CRITICAL: Query result 1470 was higher than query critical threshold 200.

MSSQL Server:

Host	Service	Status	Duration	Attempt	Last Check	Status Information
MSSQL01	MSSQL Average Wait Time	Critical	2h 8m 0s	5/5	2017-08-03 13:35:48	CRITICAL: Average Wait Time (ms) is 234.0ms
	MSSQL Buffer Hit Ratio	Ok	2h 7m 46s	1/5	2017-08-03 13:37:23	OK: Buffer Cache Hit Ratio is 100.0%
	MSSQL Checkpoint Pages Per Sec	Ok	2h 6m 19s	1/5	2017-08-03 13:38:37	OK: Checkpoint Pages / Sec is 0.0/sec
	MSSQL Connection Time	Ok	2h 7m 10s	1/5	2017-08-03 13:37:57	OK: Time to connect was 0.0039529800415s
	MSSQL Database Pages	Critical	2h 6m 47s	5/5	2017-08-03 13:37:11	CRITICAL: Database pages are 3205.0
	MSSQL Deadlocks Per Sec	Ok	2h 5m 34s	1/5	2017-08-03 13:39:27	OK: Deadlocks / Sec is 0.0/sec
	MSSQL Lazy Writes Per Sec	Ok	2h 5m 9s	1/5	2017-08-03 13:39:55	OK: Lazy Writes / Sec is 0.0/sec
	MSSQL Lock Requests Per Sec	Ok	18m 44s	1/5	2017-08-03 13:36:20	OK: Lock Requests / Sec is 13.1795663062/sec
	MSSQL Lock Timeouts Per Sec	Ok	2h 4m 20s	1/5	2017-08-03 13:35:43	OK: Lock Timeouts / Sec is 0.0/sec
	MSSQL Lock Wait Times	Ok	2h 5m 1s	1/5	2017-08-03 13:40:03	OK: Lock Wait Time (ms) is 234.0ms
	MSSQL Lock Waits Per Sec	Ok	2h 3m 41s	1/5	2017-08-03 13:36:13	OK: Lockwaits / Sec is 0.0/sec
	MSSQL Page Looks Per Sec	Ok	16m 19s	1/5	2017-08-03 13:38:45	OK: Page Lookups Per Second is 3.48661685394
	MSSQL Page Reads Per Sec	Ok	2h 3m 8s	1/5	2017-08-03 13:36:56	OK: Page Reads / Sec is 0.0/sec

Troubleshooting Tips

If you experience problems with the services created by the wizards, there are some simple troubleshooting steps you can follow which are related to the earlier sections in this documentation. The first step would be to temporarily disable the Windows Firewall on the MSSQL server and see if the problem stops. If it does, then you know you need to add firewall rules and possibly configure the MSSQL instance to listen on a specific port. There are KB articles that deal with specific issues with monitoring MSSQL in Nagios XI. You can review them here:

https://support.nagios.com/kb/article/nagios-xi-mssql-query-wizard-invalid-characters-in-the-username.html

https://support.nagios.com/kb/article/nagios-xi-mssql-wizards-adaptive-server-connection-failed.html

Other problems may require further troubleshooting via our support forums or through customer support.

Finishing Up

This completes the documentation on how to monitor Microsoft SQL with Nagios XI. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library