Установка и настройка SQL Server. Создание пользователей

# Скачивание и установка SQL Server 2019

Скачивание SQL Server 2019 происходит с официального сайта Microsoft по следующим ссылкам:

* [Выпуск Developer (https://go.microsoft.com/fwlink/?linkid=866662)](https://go.microsoft.com/fwlink/?linkid=866662)

Бесплатный выпуск с полным набором функций, лицензируемый для использования в качестве базы данных для разработки и тестирования и не предназначенный для применения в рабочей среде.

* [Выпуск Express (https://go.microsoft.com/fwlink/?linkid=866658)](https://go.microsoft.com/fwlink/?linkid=866658)

Бесплатный выпуск SQL Server, который идеально подходит для разработки приложений для использования на настольных компьютерах, веб-серверах и других небольших серверах.

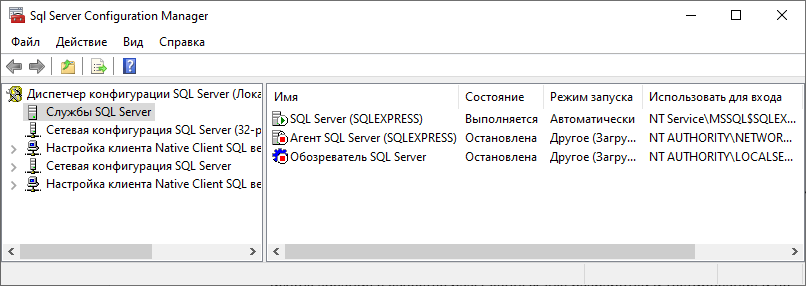
Установщик автоматизированный, выбираете базовый тип установки, далее со всем соглашаетесь и дожидаетесь конца процесса скачивания и установки. В конце установки вам будет предложено установить SSMS (SQL Server Management Studio), необходимую для работы с SQL Server. Соглашаемся, скачиваем и устанавливаем соглашаясь с установщиком (продвинутые пользователи сами могут настроить пути установки или даже установить с носителя).

# Первоначальная настройка сервера

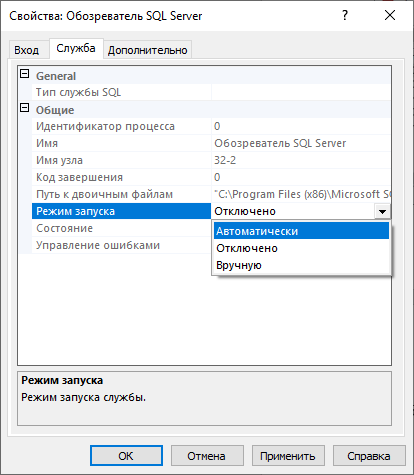
На данный момент SQL Server сразу готов для комфортного использования, но в нём не включена возможность подключаться с удалённых машин. Данная настройка необходима для сервера, для рабочей машины с локальным экземпляром это необязательно.

Для удалённого подключения через протокол TCP/IP к серверу существует служба «Обозреватель SQL Server». Она общая для всего сервера и требует настройки только один раз.

Настройка служб SQL Server производится через SQL Server 2019 Configuration Manager. Откроем его и увидим следующее окно:

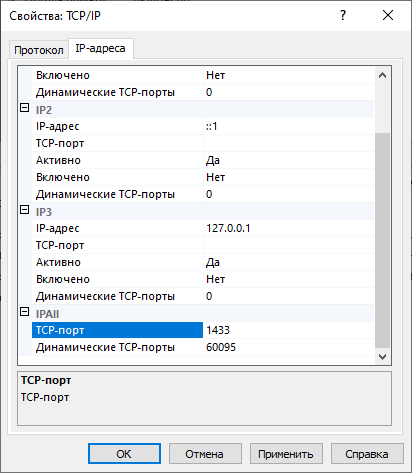


Сначала в службах SQL Server необходимо включить и установить автоматичекий запуск службы Обозревателя.

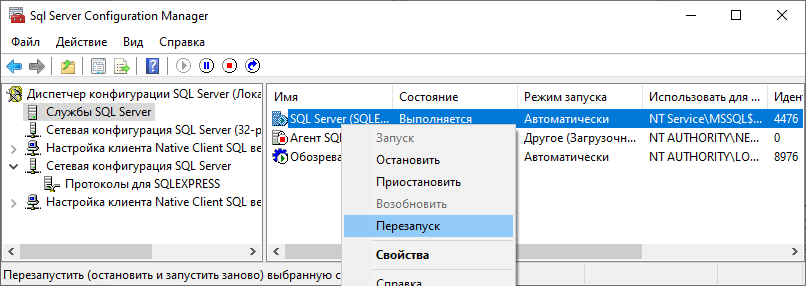


После этого нажимаем «Применить», переходим на вкладку «Вход» и соответствующей кнопкой запускаем службу обозревателя.

Теперь в левой части окна переходим к Сетевой конфигурации сервера, выбираем двойным кликом необходимый нам экземпляр. Здесь активен только пункт «Общая память». Нам необходимо включить TCP/IP. Для этого двойным кликом открываем настройки протокола и приводим к следующему виду:



И на вкладке «Протокол» активируем его. Откроется окно с предупреждением о необходимости перезагрузки сервера. Для его перезагрузки необходимо сделать следующее и выбрать необходимый экземпляр:



Настройка сервера завершена для локальных подключений, чтобы разрешить удалённые подключения к нему, необходимо открыть назначенный нами порт (1433) в Windows Firewall.

## Открытие порта 1433 в Брандмауэре

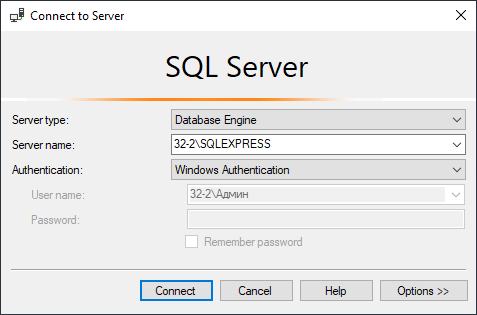
Для этого в меню Пуск переходим в Панель управления -> Система и безопасностьи -> Брандмауэр Windows (если на панели управления не включено представление по категориям, сразу выбираем Брандмауэр Windows).

Переходим в Дополнительные настройки. Мы откроем tcp порт 1433:

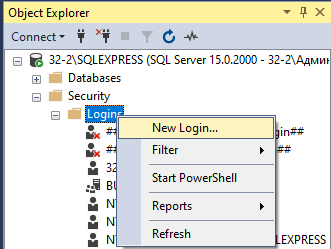
1. Выбираем Правила для входящих подключений.
2. В окне Действия нажимаем на ссылку «Создать правило».
3. Далее выбираем Тип правила в разделе Порт и нажимаем кнопку «Далее».
4. На странице Протокол и порты указываем тип порта TCP.
5. Выбираем «Указанные локальные порты» и вводим значение 1433. Снова нажимаем кнопку «Далее».
6. На странице Действие выбираем «Разрешить соединение» и жмем кнопку «Далее».
7. На странице Профиль выбираем необходимые параметры для среды.
8. На странице Имя вводим название правила SQL Server (порт 1433) и нажимаем кнопку «Готово».

На этом настройка сервера завершена и можно переходить к созданию пользователей.

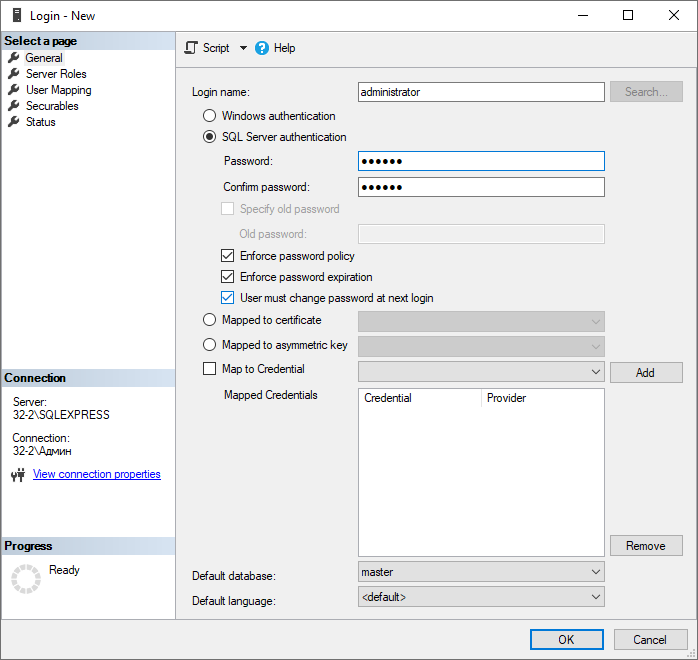
# Создание пользователей SQL Server



Подключаемся к серверу с аутентификацией Windows и обязательно из-под аккаунта с правами администратора. Сейчас мы находимся под самым привилегированным доступным пользователем SQL Server.

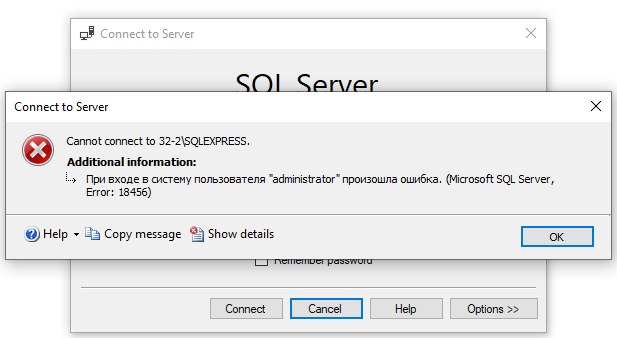
Для создания пользователей можно воспользоваться графическим интерфейсом, а можно написать всё запросами T-SQL. Для начала создадим ещё одного административного пользователя для удалённых подключений.

Откроется следующее окно, в котором мы настраиваем его имя, пароль, политику паролей и выбираем роль sysadmin на вкладке «Server Roles»:

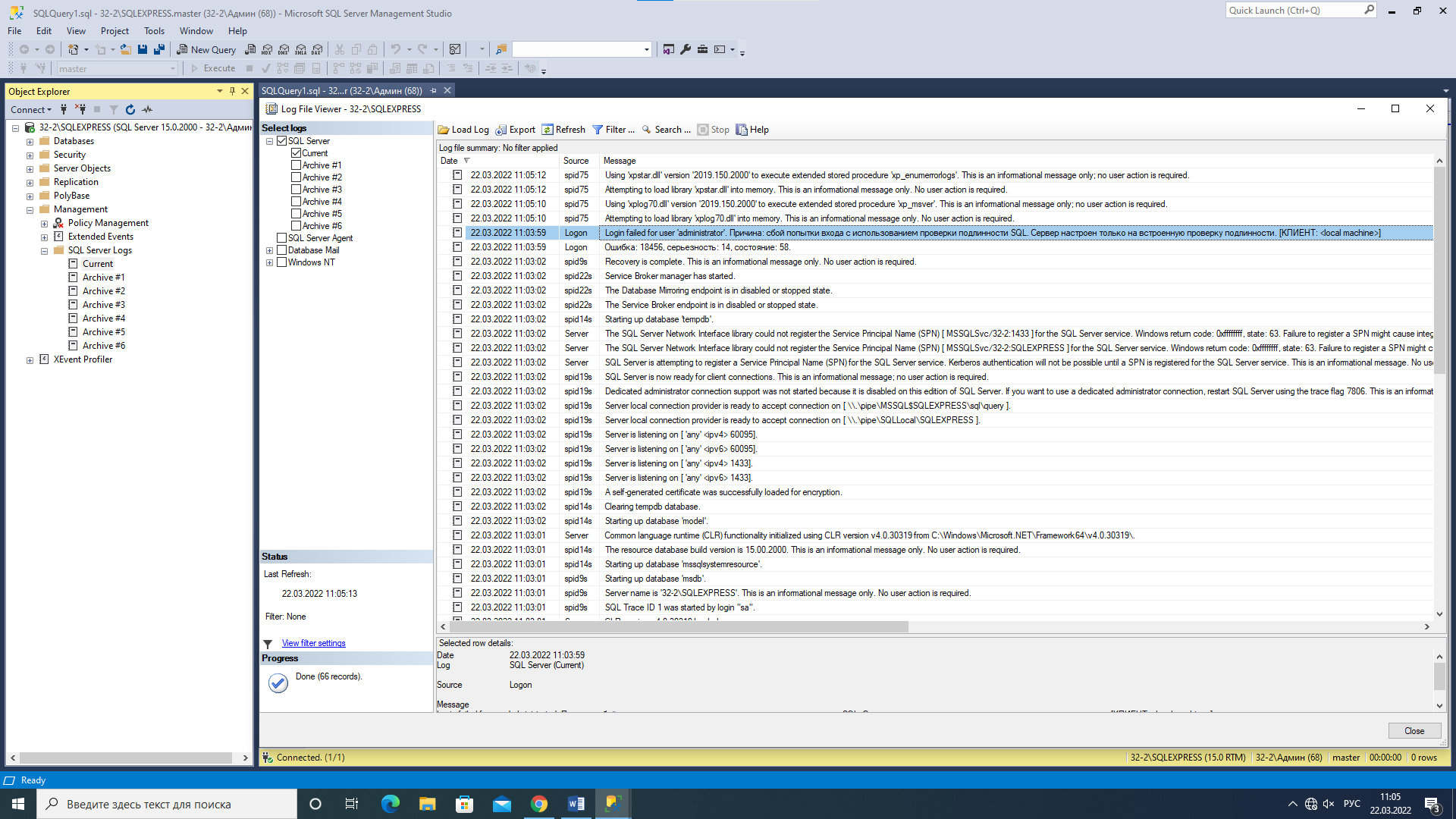


По нажатию на OK окно закроется и имя для входа будет создано. Для дальнейшей работы будем использовать этот аккаунт.

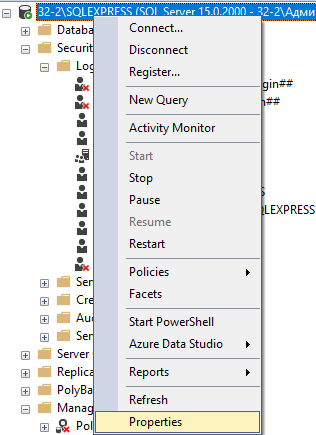
Если сейчас мы попробуем подключиться под этим именем к серверу с аутентификацией SQL Server, мы получим ошибку.

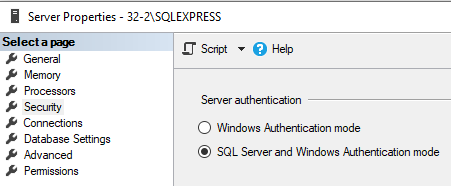


И если вернёмся под доверенного пользователя Шиндоус в логах сервера увидим следующее:



Люди, умеющие читать ошибки и перевводить их на русский, поймут, что у нас не включена проверка подлинности SQL Server. Как её включить?

Для этого необходимо открыть свойства сервера и включить смешанную аутентификацию на вкладке Безопасность и перезагрузить сервер.



Теперь мы сможем зайти под нашим созданным пользователем.

# Создание пользователя посредством скрипта и настройка дополнительных разрешений

Как и всё в SQL Server, использование GUI можно заменить на запросы T-SQL. Для создания пользователя с ограниченными правами мы воспользуемся этим способом.

Для простого создания пользователя необходимо использовать следующие запросы:

USE [master]

GO

CREATE LOGIN [user1] WITH PASSWORD=N'123456' MUST\_CHANGE,

DEFAULT\_DATABASE=[master], CHECK\_EXPIRATION=ON, CHECK\_POLICY=ON

GO

Если вы знакомы с английским языком, понять то, что здесь написано не составит труда: мы создаём имя входа [user1] с паролем 123456 который наш пользователь должен сменить (MUST\_CHANGE), его база данных по умолчанию [master], пароль может истечь, проверяется соответствие оного политике паролей Windows.

Далее, для упрощения работы себе мы применим ему разрешения роли сервера dbcreator:

ALTER SERVER ROLE [dbcreator] ADD MEMBER [user1]

GO

На этом можно было бы закончить, ведь наш пользователь теперь может создавать и удалять свои базы данных, добавлять в них таблицы и данные.

Но если мы зайдём под его аккаунтом, и начнём работать, то вскоре заметим, что он может видеть чужие базы данных, вместе с данными в них, не может изменять что-то в уже готовых таблицах и, самое страшное, редактировать системную базу данных master.

Чтобы исправить все эти проблемы нам необходимо исполнить следующие запросы:

DENY VIEW ANY DATABASE TO [user1]

GO

Здесь мы запрещаем просматривать базы данных, кроме тех, которые принадлежат самому пользователю.

GRANT ALTER ANY DATABASE TO [user1]

GO

Разрешаем пользователю изменять любые принадлежащие ему базы данных.

USE [master]

GO

Теперь для настройки прав пользователя в базе данных master мы переключаемся на её использование.

CREATE USER [user1] FOR LOGIN [user1]

GO

Создаём в ней пользователя для нашего имени входа.

DENY ALTER ON DATABASE::[master] TO [user1]

GO

И запрещаем редактирование этой базы данных.

После этих манипуляций у нас должен был получиться скрипт вроде такого:

USE [master]

GO

CREATE LOGIN [user1] WITH PASSWORD=N'123456' MUST\_CHANGE, DEFAULT\_DATABASE=[master], CHECK\_EXPIRATION=ON, CHECK\_POLICY=ON

GO

ALTER SERVER ROLE [dbcreator] ADD MEMBER [user1]

GO

DENY VIEW ANY DATABASE TO [user1]

GO

GRANT ALTER ANY DATABASE TO [user1]

GO

USE [master]

GO

CREATE USER [user1] FOR LOGIN [user1]

GO

DENY ALTER ON DATABASE::[master] TO [user1]

GO

Теперь мы можем создавать наших пользователей по такому шаблону, настраивая им различные разрешения и роли.