How to install SQL SERVER on Linux?

Install Azure Data Studio.

How to install SQL SERVER on MACOS?

1. Se poate instala SQL Server în Docker și să folosească Azure Data Studio sau DBeaver când are nevoie să facă diagrama bazei de date.
2. SQL Server instalat in VirtualBox.
3. https://phoenixnap.com/kb/install-sql-server-macos

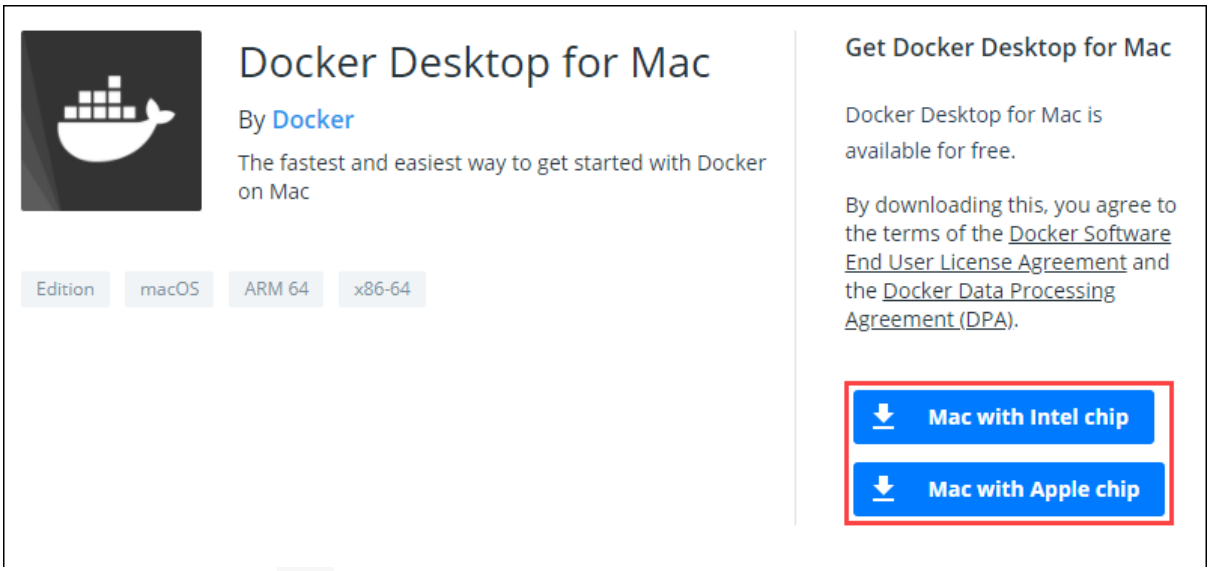
SQL Server is a relational database management tool developed by Microsoft. It is available on Windows, Linux, macOS, and as a Docker deployment.
In what follows it is showed how to install SQL Server 2019 as a Docker deployment on macOS.

Prerequisites

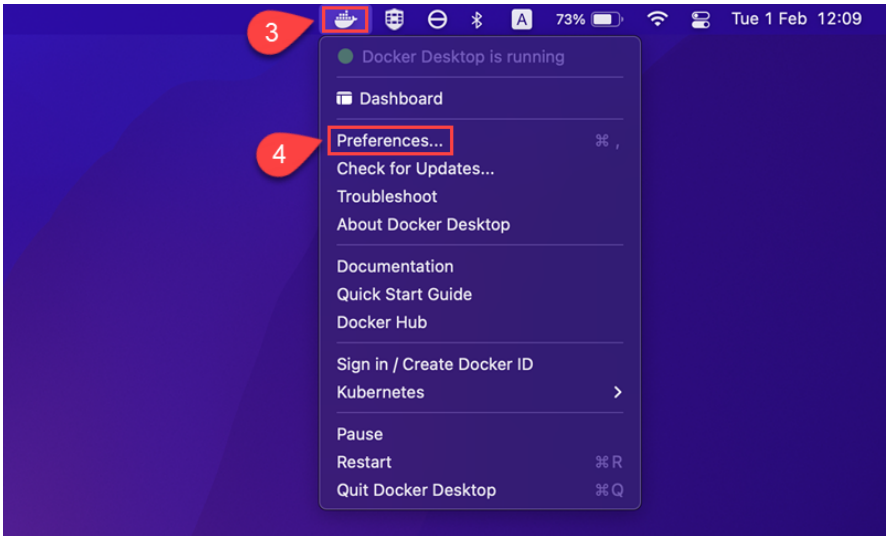
- A system running macOS Catalina or later
- A user with administrator-level privileges
- Access to the terminal window

Install and Configure Docker

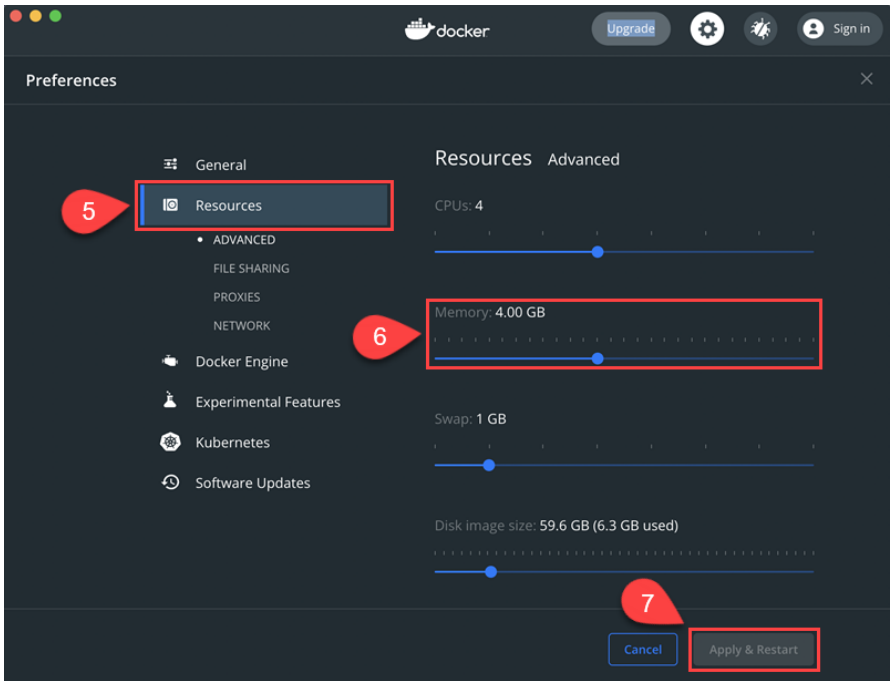
1. Download the Docker Community Edition installation file from the official Docker download page. Depending on the hardware, select the appropriate link in the **Get Docker Desktop for Mac** section to start the download.



2. Double-click the **.dmg** file to start the installation process. Once this is done, drag the **Docker.app** icon to your *Application*s folder.
3. Launch Docker, then open the **Docker** drop-down menu by clicking the Docker icon in the menu bar.
4. Select **Preferences**.



5. Open the *Resources* tab on the left side of the *Preferences* screen.
6. Increase the **Memory** value to 4.00 GB.
7. Once you are done, click **Apply & Restart** to confirm the new settings:



Install SQL Server on Mac

The MSSQL server currently supports only Intel-based Macs. If you want to use SQL on a Mac featuring an ARM-based Apple chip, skip this section and read how to install the SQL alternative for ARM-based Macs.

Follow these steps to set up SQL Server as a Docker container:

Step 1: Download the SQL Server Image

Run the following command in the terminal window to download the image for SQL Server 2019:

```
sudo docker pull mcr.microsoft.com/mssql/server:2019-latest
```

```
marko@Markos-Mac ~ % sudo docker pull mcr.microsoft.com/mssql/server:2019-latest

Password:
2019-latest: Pulling from mssql/server
ea362f368469: Pull complete
dc034f624aa1: Pull complete
cafda714f10f: Pull complete
c6af4ce68233: Pull complete
2e5e63d166b4: Pull complete
Digest: sha256:fb5277e7a3cc53f7d2230ed089ed60849f79567ebb0aae8f41ceb85879e9e09d
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2019-latest
mcr.microsoft.com/mssql/server:2019-latest
marko@Markos-Mac ~ %
```

Step 2: Launch the SQL Server Image in Docker

To launch the image you downloaded in Docker, use:

```
docker run -d --name example_sql_server -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Strong.Pwd-123' -p 1433:1433 mcr.microsoft.com/mssql/server:2019-latest
```

In the command above:

- **-d**: Launches the docker container in daemon mode, allowing it to run in the background without a terminal window open.
- **--name**: Sets a name for the Docker container. In this example, we are using *example_sql_server*.
- **-e 'ACCEPT_EULA=Y'**: Confirms you agree with the EULA (End User License Agreement) for Docker.
- **-e 'SA_PASSWORD=Strong.Pwd-123'**: Sets the database password. In this example, we are using *"Strong.Pwd-123"* as the password.
- **-p 1433:1433**: Maps the container to the local port 1433.
- **mcr.microsoft.com/mssql/server:2019-latest:** Selects an image file for Docker to use.

Note: If you get an error output with the message ***Microsoft(R) SQL Server(R) setup failed with error code 1. Please check the setup log in /var/opt/mssql/log for more information***, try the launch command again with a stronger password.

Step 3: Check the SQL Server Docker Container

Check the status of the SQL Server Docker container with:

```
docker ps -a
```

If the **STATUS** column of the output for the container says **Up**, the container is running. If it reads **Exited**, the container is no longer running and requires troubleshooting.

Step 4: Install SQL Server Command-Line Tool

Use the following command to install sql-cli:

```
sudo npm install -g sql-cli
```

sql-cli is a command-line tool that allows you to run commands and queries for an SQL Server instance in the terminal window.

Installing sql-cli with NPM requires that you have Node.js installed. If you don't, read our article on how to install Node.js on macOS.

Step 5: Connect to SQL Server

Connect to SQL Server by using the **mssql** command in the terminal window:

```
mssql -u sa -p Strong.Pwd-123
```

Where:

- **-u**: Defines the username for connecting to the database. In this example, we are using the default username *"sa"*.

- **-p**: Defines the password for logging into the database. In this example, we are using *"Strong.Pwd-123"*, which we selected while launching the SQL Server Docker container.

For more information, check out guide to installing SQL Server on Windows 10. Also have a guide on installing SQL Server on Linux.

Install the SQL Alternative for ARM-based Macs

Since the SQL server Docker image supports only amd64 architecture, attempting to run it on an ARM-based Mac machine results in error. However, Azure SQL Edge, a similar RDBM tool primarily designed for IoT edge deployments, can be used as a fully functional alternative. Follow the steps below to install Azure SQL Edge.

Step 1: Pull the Docker Image

Download the Azure SQL Edge image to your system:

```
docker pull mcr.microsoft.com/azure-sql-edge
```

```
marko@Markos-Mac ~ % docker pull mcr.microsoft.com/azure-sql-edge
Using default tag: latest
latest: Pulling from azure-sql-edge
976e4515cbe3: Pull complete
1f73897c23c8: Pull complete
b4ff7ff6a52b: Pull complete
b66501766227: Pull complete
9fd306fd7e2d: Pull complete
Digest: sha256:7c203ad8b240ef3bff81ca9794f31936c9b864cc165dd187c23c5bfe06cf0340
Status: Downloaded newer image for mcr.microsoft.com/azure-sql-edge:latest
mcr.microsoft.com/azure-sql-edge:latest
marko@Markos-Mac ~ %
```

Step 2: Run the Docker Container

When the image successfully downloads to your machine, run the container using the following command:

```
sudo docker run --cap-add SYS_PTRACE -e 'ACCEPT_EULA=Y' -e 'MSSQL_SA_PASSWORD=Strong.Pwd-123' -p 1433:1433 --name sqledge -d mcr.microsoft.com/azure-sql-edge
```

```
marko@Markos-Mac ~ % sudo docker run --cap-add SYS_PTRACE -e 'ACCEPT_EULA=Y' -e 'MSSQL_SA_PASSWORD=Strong.Pwd-123' -p 1433:1433 --name sqledge -d mcr.microsoft.com/azure-sql-edge
Password:
1174266479a0d6d6e96db5b53f67e591bc17207943017ad372b1b7e60015f109
marko@Markos-Mac ~ %
```

Step 3: Check the Container

Ensure that the Azure SQL Edge container is running properly:

```
docker ps
```

```
marko@Markos-Mac ~ % docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
1174266479a0   mcr.microsoft.com/azure-sql-edge   "/opt/mssql/bin/perm…"   3 minutes ago
marko@Markos-Mac ~ %
```

Step 4: Access the Container with Bash Shell

With the container running, you can connect to Azure SQL Edge by using the **`docker exec`** command:

```
sudo docker exec -it sqledge "bash"
```

Step 5: Connect to the Database

Once inside the container, connect to the database using the sqlcmd tool:

```
/opt/mssql-tools/bin/sqlcmd -S localhost -U SA
```

When prompted, enter the password you specified in **step 2**.
The database prompt appears. One can now use Azure SQL in the same way as MSSQL.