Microsoft SQL Server (preview for Linux)

Installation guide (13.0.2990.31)

Please note: Each command in this document begins with either a $ or # which represents the prompt. Please copy the entire command even if the command wraps to the next line.

Please note: This release has a 180-day trial period.

Contents

Installing the mssql-server Package on Red Hat Enterprise Linux (RHEL)

Requirements for Package Installation

- RHEL 7
- Minimum of 4 GB of RAM.
  - Temporary limitation: If more than 32 GB of memory is available, only 32 GB will be used.
- Minimum of 4 GB of disk space
- Less than 64 cores. Using 64 or more cores will cause an error.
- XFS (default on RHEL) or EXT4 file system
- 1433 TCP port open on the firewall for SQL Server connections
- Network connectivity to the Internet to download the package
- A hostname of less than 15 characters.
- wget is a required package to run the configuration script.

Installation Steps

1. Enter superuser mode.

```
$ sudo su
```

2. Download the configuration script and make it executable.

```
# curl -O https://private-repo.microsoft.com/tools/configure-mssql-repo-2.sh
```

```
# chmod a+x configure-mssql-repo-2.sh
```

3. Pass the unique URL provided for you in your invitation email as a parameter to the script. This URL is labeled as “**Package script configuration URL parameter**”:

```
# ./configure-mssql-repo-2.sh <URL provided in email>
```

Example:

```
# ./configure-mssql-repo-2.sh https://private-repo.microsoft.com/u/config-ee992013-f55a-3504-b940-
c6a31d941ca4.tar.gz
```

4. Exit superuser mode.

```
# exit
```

5. Run the following commands to install SQL Server:

```
$ sudo yum update
$ sudo yum install -y mssql-server
```

Verify that the package was installed

On **Red Hat**, use the following command to print the installed package name and version:

```
$ rpm -qa | grep mssql
```

Post-install configuration

After installation on Red Hat Enterprise Linux you will need to do the following to accept the license agreement and provide the system administrator (SA) password.

1. Run the configuration script to accept the license agreement and provide the System Administrator (SA) password:

```
$ cd /opt/mssql/bin
$ sudo ./sqlservr-setup
```

2. Enable and start the SQL Server service:

```
$ sudo systemctl enable mssql-server
$ sudo systemctl start mssql-server
```

3. You will need to open a port on the firewall on RHEL. If you are using firewalld as your firewall you can use these commands.

```
$ sudo firewall-cmd --zone=public --add-port=1433/tcp --permanent
$ sudo firewall-cmd --reload
```

Installing the mssql-server Package on Ubuntu

Requirements for package installation

- Ubuntu 16.04
- Minimum of 4 GB of RAM
- Minimum of 4 GB of disk space
- EXT4 (default on Ubuntu) or XFS file system
- 1433 TCP port open on the firewall for SQL Server connections
- Network connectivity to the Internet to download the package
- A hostname of less than 15 characters.
- wget is a required package to run the configuration script.

Installation steps

1. Enter superuser mode.

```
$ sudo su
```

2. Download the configuration script and make it executable.

```
# curl -O https://private-repo.microsoft.com/tools/configure-mssql-repo-2.sh
```

```
# chmod a+x configure-mssql-repo-2.sh
```

3. Pass the unique URL provided for you in your invitation email as a parameter to the script. This URL is labeled as "**Package script configuration URL parameter**":

```
# ./configure-mssql-repo-2.sh <URL provided in email>
```

Example:

```
# ./configure-mssql-repo-2.sh https://private-repo.microsoft.com/u/config-ee992013-f55a-3504-b940-
c6a31d941ca4.tar.gz
```

4. Exit superuser mode.

```
# exit
```

5. Run the following commands to install SQL Server:

```
$ sudo apt-get update
$ sudo apt-get install -y mssql-server
```

Accepting the End User License Agreement and providing the system administrator (SA) password

During installation on Ubuntu, you will receive a prompt to accept the End-User License Agreement. Select "Yes" to continue the installation.

Enter the System Administrator password for the instance. **Note**: the username for the System Administrator account is 'sa'. Confirm the System Administrator password.

Alternatively, you can run this command to install without being prompted for the license agreement and the System Administrator password:

```
$ sudo ACCEPT_EULA=Y MSSQL_SERVER_SA_PASSWORD=<your pwd> apt-get install mssql-server -y
```

Verifying that the package was installed

Use the following command to print the installed package name and version:

```
$ dpkg-query -W | grep mssql
```

Non-interactive automatic installation (advanced)

Once you have added the repository (Steps 1 through 4 above), this package can also be installed automatically by providing the configuration parameters to the debconf system before installing the package.

```
$ sudo debconf-set-selections <<< ‘mssql-server mssql-server/accept_eula boolean true’

$ sudo debconf-set-selections <<< ‘mssql-server mssql-server/sa_password string <your_password>’

$ sudo debconf-set-selections <<< ‘mssql-server mssql-server/sa_password_confirm string <your_password>’
```

After running these commands, proceed to run the installation command:

```
$ sudo apt-get install mssql-server -y
```

Upgrade

For **Ubuntu** run the yum update and then the yum install command.

```
$ sudo apt-get update
[...]
$ sudo apt-get install mssql-server
```

For **Red Hat** run the yum update and yum install command:

```
$ sudo yum update
[...]
$ sudo yum install mssql-server
```

Note: If your build is older than 13.0.2990.18 (for example, 13.0.8000.*), you will need to uninstall and reinstall the packages using the “yum|apt-get remove” and “yum|apt-get install” commands to get the latest version.

Offline mssql-server Package Installation

In some cases, the server you want to install on may not have network/Internet access to the Microsoft package repository. In that case, you can download the package on a computer that does have access and then copy it over to the server you want to install on.

There are two options for downloading the package

Download using browser or a command line tool like wget or curl

You can download using your browser or a tool like wget or curl by pointing to the following URLs:

REHL 7:

https://private-repo.microsoft.com/rhel7/mssql-private-preview/mssql-server-13.0.2990.31-12.x86_64.rpm

Ubuntu:

https://private-repo.microsoft.com/ubuntu/mssql-private-preview/pool/main/m/mssql-server/mssql-server_13.0.2990-31-8_amd64.deb

Download using yum or apt-get

You can also download the package using yum or apt-get as follows. Note: for this option you will need to complete the steps above to configure your computer with the configure-mssql-repo-2.sh script. Downloading the script and executing it will add the certificate to your computer so that you can access the repository. Then, follow the instructions below depending on your Linux distribution.

For Red Hat Enterprise Linux 7.2:

- Install the downloading utility:

```
$ sudo yum install yum-downloadonly
```

- Use the download utility to get the RPM package. Set the output directory:

```
$ sudo yum install –downloadonly --downloaddir=<output_directory> mssql-server
```

- Use the RPM command to install the local package, which should be named like this:

```
$ sudo rpm -i mssql-server-13.0.2990.30-6.x86_64.rpm
```

For Ubuntu 16.04:

- Use the following command to download the Debian package in a non-root folder:

```
$ apt-get download mssql-server
```

- The package will be saved under the current folder and would be named like this:
  mssql-server_13.0.2990.28-13_amd64.deb

- After that, you can copy/send the file to another Ubuntu host. To install this file, use the following command:

```
$ sudo dpkg -i mssql-server_13.0.2990.28-13_amd64.deb
```

Pull and Run the mssql-server Docker Image

Requirements for Docker

- Docker Engine 1.8+ on any supported Linux distribution or Docker for Mac/Windows.
- Minimum of 4 GB of disk space
- Minimum of 4 GB of RAM
  Important: The default on Docker for Mac and Docker for Windows is 2 GB for the Moby VM, so you will need to change it to 4 GB.

  For Mac users:

  1. Click on the Docker logo on the top status bar.
  2. Select “Preferences”.
  3. Move the memory indicator to 4GB or more.
  4. Click the “restart” button at the button of the screen.

  For Windows users:

  1. Right-click on the Docker icon from the task bar.
  2. Click “Settings” under that menu.
  3. Click on the “Advanced” Tab.
  4. Move the memory indicator to 4GB or more.
  5. Click the “Apply” button.

Docker Pull and Run Steps

1. Login to the preview Docker registry by using the credentials provided to you in the welcome email.

```
$ sudo docker login private-repo.microsoft.com
* Prompts for username and password *
```

2. You can now pull the Docker image from the Docker registry.

```
$ sudo docker pull private-repo.microsoft.com/mssql-private-preview/mssql-server:latest
```

3. To run the Docker image, you can use the following commands:

```
$ sudo docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=<your_password_here>" -p 1433:1433 -d
private-repo.microsoft.com/mssql-private-preview/mssql-server:latest
```

4. To persist the data generated from your Docker container, you must map volume to the host machine. To do that, use the run command with the "-v <host_folder>:/var/opt/mssql" flag. This will allow the data to be restored between container executions.

```
$ sudo docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=<your_password_here>" -p 1433:1433 -v
<host folder>:/var/opt/mssql -d private-repo.microsoft.com/mssql-private-preview/mssql-
server:latest
```

Note: The ACCEPT_EULA and SA_PASSWORD environment variables are **required** to run the image.

Install the Command-line Tools and ODBC Drivers on Ubuntu

The Command Line Utilities for SQL Server (SQLCMD and BCP utilities) are in private preview and available for Ubuntu 16.04. The SQLCMD utility allows users to connect to, send Transact-SQL batches from, and output rowset information from SQL Server instances. The BCP utility bulk copies data between an instance of Microsoft SQL Server and a data file in a user-specified format. Installing these utilities will also install the Microsoft ODBC driver and all its dependencies by default. To install these tools, run the following commands in the terminal:

Note: If you are installing these packages from the same machine as your SQL Server on Linux installation, you do NOT need to perform steps 1 and 2 (you already performed these steps in the Installation Guide).

1. Download and prepare the configuration script:

```
$ curl -O https://private-repo.microsoft.com/tools/configure-mssql-repo.sh
$ sudo chmod a+x configure-mssql-repo.sh
```

2. Pass the unique URL provided for you in your invitation email as a parameter to the script. This URL is labeled as "Debian Package script configuration URL parameter":

```
$ sudo ./configure-mssql-repo.sh <URL provided in email>
```

Example:

```
$ sudo ./configure-mssql-repo.sh https://private-
repo.microsoft.com/u/config-ee992013-f55a-3504-b940-c6a31d941ca4.tar.gz
```

3. Add the private preview tools repository, and install the SQL Command Line Utilities with its dependencies by running the commands below.

```
$ echo "deb https://private-repo.microsoft.com/ubuntu/mssql-tools-private-preview
mssql main" | sudo tee /etc/apt/sources.list.d/mssql-tools.list
```

4. Update the sources list and run the installation command:

```
$ sudo apt-get update
$ sudo apt-get install mssql-tools
```

5. To connect to the localhost on your Ubuntu machine:

```
$ sqlcmd -S localhost -U SA -P <password>
```

Install the Command-line Tools and ODBC Drivers on Red Hat Enterprise Linux (RHEL)

The command-line tools (SQLCMD and BCP) can also be installed on Red Hat Enterprise Linux through the Microsoft ODBC drivers package. We have created an installation script to simplify the process, which can be downloaded and run by following these steps:

1. Start a terminal session as root:

```
$ sudo su
```

2. Download the installation script:

```
$ wget https://gallery.technet.microsoft.com/ODBC-Driver-13-for-SQL-
8d067754/file/153653/4/install.sh
```

3. Run the installation script

```
$ sh install.sh
```

4. Once completed, exit the root mode.

```
$ exit
```

5. You should now be able to call the “sqlcmd” and “bcp” commands:

```
$ sqlcmd -H <hostname> -U <username> -P <password>
```

6. To uninstall the tools, download the uninstall script and follow the same steps as described above:

```
$ sudo su
$ wget https://gallery.technet.microsoft.com/ODBC-Driver-13-Tools-
e419eed1/file/153767/2/uninstall.sh
$ sh uninstall.sh
$ exit
```

Next Steps

Evaluating SQL Server on Linux

Once you have set up SQL Server on Linux, you can now proceed to the Evaluation Guide provided in the Yammer website. This guide describes the steps to connect to SQL Server on Linux and walks through common connection scenarios.

Using SQL Server on Linux in an Azure VM

If you are using an Azure VM, make sure you configure your firewall to allow external connections. More information can be found in the Azure Guide, under “Connecting to a Linux VM in Azure using SSH/PuTTY from a Windows PC”.