MariaDB Package Repository Setup and Usage

Instructions on configuring and using the MariaDB package repository.

Overview

If you are looking to set up MariaDB Server, it is often easiest to use a repository. MariaDB Foundation has a repository configuration tool and MariaDB Corporation provides two convenient shell scripts to configure access to their MariaDB Package Repositories:

- `mariadb_es_repo_setup` for MariaDB Enterprise Server, which can be downloaded from:
  - https://dlm.mariadb.com/enterprise-release-helpers/mariadb_es_repo_setup
- `mariadb_repo_setup` for MariaDB Community Server, which can be downloaded from:
  - https://r.mariadb.com/downloads/mariadb_repo_setup

Using MariaDB Foundation's Repository Configuration Tool

Visit https://mariadb.org/download/?t=repo-config and follow the instructions from there. It will ask for your Linux distribution, desired MariaDB version, and the mirror to use, and will show what files to edit and what commands to run to configure a repository.

Using MariaDB Corporation's Repository Setup Scripts

Alternatively, you can run a convenient shell script that will automatically configure a repository for you.

Download and Verify the Script

The repository setup script can be downloaded and verified in the following way:

mariadb_es_repo_setup | mariadb_repo_setup

Download the script:

```
curl -LsSO https://dlm.mariadb.com/enterprise-release-helpers/
mariadb_es_repo_setup
```

Verify the checksum of the script:

```
echo "${checksum} mariadb_es_repo_setup" | sha256sum -c -
```

Checksums of the various releases of the MariaDB Corporation's repository setup scripts can be found in the Versions section at the bottom of this page. Substitute `${checksum}` in the example above with the checksum of the version of the script you are using.

Prerequisites

For the script to work, the `curl` package needs to be installed on your system. Additionally, on Debian and Ubuntu, the `apt-transport-https` package needs to be installed. The script will check if these are installed and let you know before it attempts to create the repository configuration on your system.

They can be installed on your system as follows:

RHEL / Rocky / Alma | Debian / Ubuntu | SLES

```
sudo dnf install curl
```

Run the Script

After the script is downloaded, you need to run it with `root` user permissions. This is normally accomplished by using the `sudo` command:

mariadb_es_repo_setup mariadb_repo_setup

Retrieve your customer downloads token:

1. Navigate to https://customers.mariadb.com/downloads/token/ and log in
2. Copy the Customer Download Token
3. Substitute your token for `${token}` when running the `mariadb_es_repo_setup` script, below
4. Set the script to be executable:

```
chmod +x mariadb_es_repo_setup
```

5. Run the script:

```
sudo ./mariadb_es_repo_setup --token="${token}" --apply
```

Repositories

The script will set up different repositories in a single repository configuration file.

The default repositories set up by `mariadb_es_repo_setup` are:

- MariaDB Enterprise Server Repository
  - A MariaDB Enterprise Server Debug Repository (Ubuntu only)
- MariaDB Enterprise MaxScale Repository
- MariaDB Enterprise Tools Repository
- MariaDB Tools Repository

The default repositories set up by `mariadb_repo_setup` are:

- MariaDB Community Server Repository
- MariaDB Community Server Debug Repository (Ubuntu only)
- MariaDB MaxScale Repository
- MariaDB Tools Repository

> Ubuntu needs a separate debug repository for MariaDB Server debug packages. Other Linux distributions include the debug packages in the main repository. Debug packages should normally only be installed for specific purposes under the direction of a qualified support engineer.

MariaDB Community Server Repository

The **MariaDB Community Server Repository** contains software packages related to MariaDB Server, including the server itself, clients and utilities, client libraries, plugins, and mariadb-backup.

The binaries in MariaDB Corporation's **MariaDB Repository** are identical to the binaries in MariaDB Foundation's MariaDB Repository that is configured with the MariaDB Foundation's Repository Configuration Tool.

By default, the `mariadb_repo_setup` script will configure your system to install from the `12.rolling` repository, which contains the latest stable version of MariaDB Community server.

The `mariadb_es_repo_setup` script will set up the current latest stable version of MariaDB Enterprise Server.

If you would like to stick to a specific release series, then you will need to either manually edit the repository configuration file to point to that specific version or series, or run the MariaDB Package Repository setup script again using the `--mariadb-server-version` option. For example, if you wanted to specifically use the 11.4 series, you would do: `--mariadb-server-version=11.4`.

If you do not want to configure the **MariaDB Repository** on your system, for example, if you are setting up a server just running MariaDB MaxScale, then you can use the `--skip-server` option to prevent the setup script from configuring the server repository.

MariaDB MaxScale Repository

> **Note**
>
> MaxScale releases, as of 2025-12-09, are now signed with a new key. The `mariadb_repo_setup` and `mariadb_es_repo_setup` scripts have been updated to automatically install the new key, but for existing repositories, you'll need to do the following.
>
> On Debian and Ubuntu:
>
> ```
> curl -LsSO https://supplychain.mariadb.com/mariadb-keyring-2025.gpg
> sudo mv mariadb-keyring-2025.gpg /etc/apt/trusted.gpg.d/
> sudo apt update
> ```
>
> On RHEL & friends:
>
> ```
> sudo rpm --import https://supplychain.mariadb.com/MariaDB-Enterprise-GPG-KEY-2025
> ```

The **MariaDB MaxScale Repository** contains software packages related to MariaDB MaxScale.

By default, the script will configure your system to install from the repository of the *latest* GA version of MariaDB MaxScale. When a new major GA release occurs, the repository will automatically switch to the new version. If instead you would like to stay on a particular version, you will need to manually edit the repository configuration file and change '`latest`' to the version you want (e.g., '`6.1`') or run the MariaDB Package Repository setup script again, specifying the particular version or series you want.

Older versions of the MariaDB Package Repository setup script would configure a specific MariaDB MaxScale series in the repository (i.e., `24.02`), so if you used the script in the past to set up your repository and want MariaDB MaxScale to automatically use the latest GA version, then change `24.02` or whatever version it is set to in the repository configuration to `latest`. Or download the current version of the setup script and re-run it to set up the repository again.

The script can configure your system to install from the repository of an older version of MariaDB MaxScale if you use the `--mariadb-maxscale-version` option. For example, `--mariadb-maxscale-version=25.01`.

If you do not want to configure the **MariaDB MaxScale Repository** on your system, then you can use the `--skip-maxscale` option to prevent the setup script from configuring it.

Supported Distributions

The MariaDB Package Repository setup script is designed for Linux distributions that meet MariaDB's current platform support policy. Supported platforms may vary over time and can differ across different MariaDB release series.

For a comprehensive and current list of supported platforms, refer to:

- MariDB Engineering Policy
- The MariaDB Downloads page for your specific version

If the setup script does not support your distribution, you can install MariaDB using the MariaDB Foundation's Repository Configuration Tool or check your distribution's native repositories for MariaDB packages.

Options

To provide options to the script, you must tell your script to expect them by executing bash with the options `-s --`, for example:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup |
sudo bash -s -- --help
```

Option	Description
`--help`	Display a usage message and exit
`--mariadb-server-version=`	Override the default MariaDB Server version. By default, the script will use '11.rolling'
`--mariadb-maxscale-version=`	Override the default MariaDB MaxScale version. By default, the script will use 'latest'
`--os-type=`	Override detection of OS type. Acceptable values include debian, ubuntu, rhel, and sles
`--os-version=`	Override detection of OS version. Acceptable values depend on the OS type you specify
`--skip-key-import`	Skip importing GPG signing keys
`--skip-maxscale`	Skip the 'MaxScale' repository
`--skip-server`	Skip the 'MariaDB Server' repository
`--skip-tools`	Skip the 'Tools' repository
`--skip-verify`	Skip verification of MariaDB Server versions. Use with caution, as this can lead to an invalid repository configuration file being created
`--skip-check-installed`	Skip tests for required prerequisites for this script
`--skip-eol-check`	Skip tests for versions that are past their EOL date
`--skip-os-eol-check`	Skip tests for operating system versions past the EOL date
`--write-to-stdout`	Write output to stdout instead of to the OS's repository configuration file. This will also skip importing GPG public keys and updating the package cache on platforms where that behavior exists

`--mariadb-server-version`

By default, the script will configure your system to install from the repository of the latest GA version of MariaDB. If a new major GA release occurs and you would like to upgrade to it, then you will need to either manually edit the repository configuration file to point to the new version or run the MariaDB Package Repository setup script again.

The script can also configure your system to install from the repository of a different version of MariaDB if you use the `--mariadb-server-version` option.

The string `mariadb-` has to be prepended to the version number. For example, to configure your system to install from the repository of MariaDB 11.8, that would be:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup |
sudo bash -s -- --mariadb-server-version="mariadb-11.8"
```

The following MariaDB versions are currently supported:

- `mariadb-10.6`
- `mariadb-10.11`
- `mariadb-11.4`
- `mariadb-11.8`
- `mariadb-11.rolling`
- `mariadb-11.rc`
- `mariadb-12.1`
- `mariadb-12.2`
- `mariadb-12.rolling`
- `mariadb-12.rc`

If you want to pin the repository of a specific minor release, such as MariaDB 11.8.5, then you can also specify the minor release. For example, `mariadb-10.8.5`. This may be helpful if you want to avoid upgrades. However, avoiding upgrades is not recommended, since minor maintenance releases may contain important bug fixes and fixes for security vulnerabilities.

`--mariadb-maxscale-version`

By default, the script will configure your system to install from the repository of the latest GA version of MariaDB MaxScale.

If you would like to pin the repository to a specific version of MariaDB MaxScale, then you will need
to either manually edit the repository configuration file to point to the desired version or use the `--mariadb-maxscale-version` option.

For example, to configure your system to install from the repository of MariaDB MaxScale 6.1, that would be:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup |
sudo bash -s -- --mariadb-maxscale-version="6.1"
```

The following MariaDB MaxScale versions are currently supported:

- MaxScale 25.10
- MaxScale 25.01
- MaxScale 24.02
- MaxScale 23.08
- MaxScale 23.02
- MaxScale 22.08

The special identifiers `latest` (for the latest GA release) and `beta` (for the latest beta release) are also supported. By default, the `mariadb_repo_setup` script uses `latest` as the version.

`--os-type` and `--os-version`

If you want to run this script on an unsupported OS that you believe to be package-compatible with an OS that is supported, then you can use the `--os-type` and `--os-version` options to override the script's OS detection. If you use either option, then you must use both options.

The supported values for `--os-type` are:

- `rhel`
- `debian`
- `ubuntu`
- `sles`

If you use a non-supported value, then the script will fail, just as it would fail if you ran the script on an unsupported OS.

The supported values for `--os-version` are entirely dependent on the OS type.

For Red Hat Enterprise Linux (RHEL): `8`, `9`, and `10` are valid options.

For Debian and Ubuntu, the version must be specified as the codename of the specific release. For example, Debian 13 must be specified as `trixie`, and Ubuntu 24.04 must be specified as `noble`.

These options can be useful if your distribution is a fork of another distribution. As an example, Pop!_OS 24.04 LTS is based on and is fully compatible with Ubuntu 24.04 LTS (noble). Therefore, if you are using Pop!_OS, then you can configure your system to install from the repository of Ubuntu 24.04 LTS (noble) by specifying `--os-type=ubuntu`` ``--os-version=noble` to the MariaDB Package Repository setup script.

For example, to manually set the `--os-type` and `--os-version` to RHEL 10, you could do:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup |
sudo bash -s -- --os-type=rhel --os-version=10
```

`--write-to-stdout`

The `--write-to-stdout` option will prevent the script from modifying anything on the system. The repository configuration will not be written to the repository configuration file. Instead, it will be printed to standard output. That allows the configuration to be reviewed, redirected elsewhere, consumed by another script, or used in some other way.

The `--write-to-stdout` option automatically enables `--skip-key-import`.

For example:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup |
sudo bash -s -- --write-to-stdout
```

Platform-Specific Behavior

RHEL | Debian / Ubuntu | SLES

Platform-Specific Behavior on RHEL and equivalents

On Red Hat Enterprise Linux (RHEL) and equivalents, the MariaDB Package Repository setup script performs the following tasks:

1. Creates a repository configuration file at `/etc/yum.repos.d/mariadb.repo`
2. Imports the GPG public key used to verify the signature of MariaDB software packages with `rpm --import` from `supplychain.mariadb.com`

Installing Packages With the MariaDB Package Repository

After setting up the MariaDB Package Repository, you can install the software packages in the supported repositories.

RHEL | Debian / Ubuntu | SLES

Installing Packages on RHEL and equivalents

To install MariaDB on Red Hat Enterprise Linux (RHEL) and equivalents, see the instructions in the Installing MariaDB Packages with YUM. For example:

```
sudo dnf install MariaDB-server MariaDB-client MariaDB-backup
```

To install MariaDB MaxScale on Red Hat Enterprise Linux (RHEL) and equivalents, see the instructions at MariaDB MaxScale Installation Guide. For example:

```
sudo dnf install maxscale
```

Versions

mariadb_es_repo_setup mariadb_repo_setup

mariadb_es_repo_setup Versions

Version	sha256sum
2025-12-10	`62a28aa1f060b4055751d93a88bc11c5556c2b23103c6a6287a8fcb0a4b8a13f`
2025-10-22	`1f584ffd368d18c64b8820bf6cd9b1114dda11a0ecf9524be3c967a3a5be941b`
2025-09-08	`c33b022c2cc325fa50be62eae070ea0bdcaf85367f840accac7acaeea1e8a972`
2025-06-04	`4d483b4df193831a0101d3dfa7fb3e17411dda7fc06c31be4f9e089c325403c0`
2025-01-16	`99ea6c55dbf32bfc42cdcd05c892aebc5e51b06f4c72ec209031639d6e7db9fe`
2025-01-07	`b98c6436e01ff33d7e88513edd7b77a965c4500d6d52ee3f106a198a558927af`
2024-11-19	`97e5ef25b4c4a4bd70b30da46b1eae0b57db2f755ef820a28d254e902ab5a879`
2024-11-13	`0c181ada4e7a4cd1d7688435c478893502675b880be2b918af7d998e239eb325`
2024-09-20	`c12da6a9baa57eab7fa685aa24bf76e6929a8c67f4cd244835520c0181007753`
2024-09-09	`733f247c626d965304b678b62a4b86eb4bb8bf956f98a241b6578dedc6ca4020`
2024-06-12	`b96fcd684a84bbe1080b6276f424537fc9d9c11ebe243ad8b9a45dd459f6ee4f`
2023-07-27	`f8eb9c1b59ccfd979d27e39798d2f2a98447dd29e2149ce92bf606aab4493ad9`
2023-03-13	`8dfef0ec98eb03a4455df07b33107a6d4601425c9df0ab5749b8f10bf3abdcbb`
2022-10-26	`3f4a9d1c507a846a598e95d6223871aade69a9955276455324e7cc5f54a87021`
2022-09-12	`713a8f78ea7bab3eccfb46dc14e61cd54c5cf5a08acb5c320ef5370d375e48bd`
2022-06-14	`cfcd35671125d657a212d92b93be7b1f`

	`4ad2fda58dfa8b5ab4b601bf3afa4eae`
2022-03-11	`53efddb84ea12efa7d521499a7474065bd4a60c721492d0e72b4336192f4033f`
2021-12-13	`5feb2aac767c512cc9e2af674d1aef42df0b775ba2968fffa8700eb42702bd44`
2021-10-13	`4f266ff758fe15eeb9b8b448a003eb53e93f3064baf1acb789dd39de4f534b1d`
2021-09-14	`b741361ea3a0a9fcaa30888a63ff3a8a4021882f126cf4ef26cf616493a29315`
2021-08-26	`a49347a4e36f99c5b248403ed9fb9b33a2f07f5e24605a694b1b1e24d7199f28`
2021-06-29	`99e768b24ae430b37dec7cb69cdd625396630dba18f5e1588ee24d3d8bb97064`
2021-06-14	`ec08f8ede524f568b3766795ad8ca1a0d0ac4db355a18c3d85681d7f9c0f8c09`
2021-05-04	`bf67a231c477fba0060996a83b197c29617b6193e1167f6f062216ae13c716c7`
2021-03-15	`99c7f4a3473a397d824d5f591274c2a4f5ebf6dc292eea154800bbaca04ddc7e`
2021-02-12	`c78db828709d94876406a0ea346f13fbc38e73996795903f40e3c21385857dd4`
2020-12-16	`c01fa97aed71ca0cd37cba7036ff80ab40efed4cc261c890aa2aa11cd8ab4e2f`
2020-12-15	`e42f1f16f2c78a3de0e73dcc2a9081e2f771b3161f4f4ceecb13ea788d84673b`
2020-12-14	`4aaf495606633a47c55ea602829e67e702aec0a5c6ff6b1af90709c19ee9f322`
2020-10-07	`93fa0df3d6491a791f5d699158dcfe3e6ce20c45ddc2f534ed2f5eac6468ff0a`
2020-09-08	`eeebe9e08dffb8a4e820cc0f673afe437621060129169ea3db0790eb649dbe9b`
2020-07-16	`957bc29576e8fd320fa18e35fa49b5733f3c8eeb4ca06792fb1f05e089c810ff`

Previous
Installing MariaDB Windows ZIP Packages

Next
Deploy with Package Tarballs

Last updated 14 days ago

Was this helpful?