Install Docker Compose

Estimated reading time: 7 minutes

You can run Compose on macOS, Windows, and 64-bit Linux.

Prerequisites

Docker Compose relies on Docker Engine for any meaningful work, so make sure you have Docker Engine installed either locally or remote, depending on your setup.

- On desktop systems like Docker Desktop for Mac and Windows, Docker Compose is included as part of those desktop installs.
- On Linux systems, first install the Docker (https://docs.docker.com/install/#server) for your OS as described on the Get Docker page, then come back here for instructions on installing Compose on Linux systems.
- To run Compose as a non-root user, see Manage Docker as a non-root user (https://docs.docker.com/install/linux/linux-postinstall/).

Install Compose

Follow the instructions below to install Compose on Mac, Windows, Windows Server 2016, or Linux systems, or find out about alternatives like using the `pip` Python package manager or installing Compose as a container.

> **Install a different version**
>
> The instructions below outline installation of the current stable release (**v1.25.4**) of Compose. To install a different version of Compose, replace the given release number with the one that you want. Compose releases are also listed and available for direct download on the Compose repository release page on GitHub (https://github.com/docker/compose/releases). To install a **pre-release** of Compose, refer to the install pre-release builds (/compose/install/#install-pre-release-builds) section.

Mac | Windows | Windows Server | Linux | Alternative Install Options

Install Compose on Linux systems

On Linux, you can download the Docker Compose binary from the Compose repository release page on GitHub (https://github.com/docker/compose/releases). Follow the instructions from the link, which involve running the `curl` command in your terminal to download the binaries. These step-by-step instructions are also included below.

> ❗ For `alpine`, the following dependency packages are needed: `py-pip`, `python-dev`, `libffi-dev`, `openssl-dev`, `gcc`, `libc-dev`, and `make`.

1. Run this command to download the current stable release of Docker Compose:

   ```
   sudo curl -L "https://github.com/docker/compose/releases/download/
   ```

   > To install a different version of Compose, substitute `1.25.4` with the version of Compose you want to use.

   If you have problems installing with `curl`, see Alternative Install Options (https://docs.docker.com/compose/install/#alternative-install-options) tab above.

2. Apply executable permissions to the binary:

   ```
   sudo chmod +x /usr/local/bin/docker-compose
   ```

> **Note**: If the command `docker-compose` fails after installation, check your path. You can also create a symbolic link to `/usr/bin` or any other directory in your path.

For example:

```
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

1. Optionally, install command completion (https://docs.docker.com/compose/completion/) for the `bash` and `zsh` shell.
2. Test the installation.

```
$ docker-compose --version
docker-compose version 1.25.4, build 1110ad01
```

Install pre-release builds

If you're interested in trying out a pre-release build, you can download release candidates from the Compose repository release page on GitHub (https://github.com/docker/compose/releases). Follow the instructions from the link, which involves running the `curl` command in your terminal to download the binaries.

Pre-releases built from the "master" branch are also available for download at https://dl.bintray.com/docker-compose/master/ (https://dl.bintray.com/docker-compose/master/).

> Pre-release builds allow you to try out new features before they are released, but may be less stable.

Upgrading

If you're upgrading from Compose 1.2 or earlier, remove or migrate your existing containers after upgrading Compose. This is because, as of version 1.3, Compose uses Docker labels to keep track of containers, and your containers need to be recreated to add the labels.

If Compose detects containers that were created without labels, it refuses to run, so that you don't end up with two sets of them. If you want to keep using your existing containers (for example, because they have data volumes you want to preserve), you can use Compose 1.5.x to migrate them with the following command:

```
docker-compose migrate-to-labels
```

Alternatively, if you're not worried about keeping them, you can remove them. Compose just creates new ones.

```
docker container rm -f -v myapp_web_1 myapp_db_1 ...
```

Uninstallation

To uninstall Docker Compose if you installed using `curl`:

```
sudo rm /usr/local/bin/docker-compose
```

To uninstall Docker Compose if you installed using `pip`:

```
pip uninstall docker-compose
```

> **Got a "Permission denied" error?**
>
> If you get a "Permission denied" error using either of the above methods, you probably do not have the proper permissions to remove `docker-compose`. To force the removal, prepend `sudo` to either of the above commands and run again.

Where to go next

- User guide (https://docs.docker.com/compose/)
- Getting Started (https://docs.docker.com/compose/gettingstarted/)
- Get started with Django (https://docs.docker.com/compose/django/)
- Get started with Rails (https://docs.docker.com/compose/rails/)
- Get started with WordPress (https://docs.docker.com/compose/wordpress/)
- Command line reference (https://docs.docker.com/compose/reference/)
- Compose file reference (https://docs.docker.com/compose/compose-file/)

compose (https://docs.docker.com/search/?q=compose), orchestration (https://docs.docker.com/search/?q=orchestration), install (https://docs.docker.com/search/?q=install), installation (https://docs.docker.com/search/?q=installation), docker (https://docs.docker.com/search/?q=docker), documentation (https://docs.docker.com/search/?q=documentation)