RedHat - Install Docker Engine on RHEL 9

> Last verified: October 2026 for **RHEL 9** against Docker's Install Docker Engine on RHEL guide, Linux post-installation steps, and packet-filtering / firewalld notes (see **Sources**).

This installs **Docker Engine Community (Docker CE)** from Docker's official RHEL repository: the `dockerd` daemon, the `docker` CLI, `containerd`, Buildx, and the Compose v2 plugin. It is **not** Docker Desktop, and it is **not** Red Hat's Podman stack.

Red Hat ships **Podman** as the supported engine on RHEL 9. Use this page when you specifically need Docker Engine. If you want Podman instead, use the Podman page in the Docker book.

Prerequisites

- RHEL 9.x (64-bit). Docker documents RHEL 8, 9 and 10. This page is written for RHEL 9.
- An active Red Hat subscription so `dnf` can install base dependencies.
- A sudo-capable account.
- Outbound HTTPS to `download.docker.com` (and the usual RHEL repos).

Rocky Linux and AlmaLinux are RHEL rebuilds. Docker's documented repo for this procedure is `download.docker.com/linux/rhel`. Docker does not list Rocky/Alma as tested platforms on the RHEL install page.

1) Conflicts with Podman

Docker's packages conflict with the distribution's unofficial Docker packages **and with Podman / runc**. The uninstall step below is what Docker documents. **Do not run it on a host that must keep Podman.** Pick one engine per host.

```
sudo dnf remove docker \
  docker-client \
  docker-client-latest \
  docker-common \
  docker-latest \
```

```
docker-latest-logrotate \
docker-logrotate \
docker-engine \
podman \
runc
```

`dnf` may report that none of those packages are installed. Images, containers, volumes and networks under `/var/lib/docker/` are not removed by this command.

2) Add Docker's RHEL repository

```
sudo dnf -y install dnf-plugins-core
sudo dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
```

Confirm the repo is present:

```
dnf repolist docker-ce-stable
```

3) Install Docker Engine

Latest packages:

```
sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

If DNF asks you to accept the GPG key, check the fingerprint matches Docker's published key before you accept:

```
060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35
```

This installs Docker but does **not** start it. It creates a `docker` group and does not add any users to that group.

To pin a version, list what the repo currently has, then install the fully qualified NEVRA (name + epoch/version from the second column):

```
dnf list docker-ce --showduplicates | sort -r
sudo dnf install docker-ce-<VERSION_STRING> docker-ce-cli-<VERSION_STRING> containerd.io docker-buildx-plugin docker-compose-plugin
```

Example NEVRA from Docker's docs at verification time: `docker-ce-3:29.8.2-1.el9`. The list on your host will change as Docker publishes updates.

4) Start and enable the service

```
sudo systemctl enable --now docker
sudo systemctl enable containerd.service
sudo systemctl status docker --no-pager
```

If you do not want Docker at boot, use `sudo systemctl start docker` instead of `enable --now`.

5) Verify

```
sudo docker version
sudo docker compose version
sudo docker run --rm hello-world
```

`hello-world` pulls a test image, runs a container, prints a confirmation message, and exits. Use `sudo` until you add a user to the `docker` group.

6) Configure Docker's default address pools

The address pools control which subnets Docker hands out to new bridge networks. Each network gets a /24, so Docker does not run out of address space. Create or edit the daemon config:

```
sudo vi /etc/docker/daemon.json
```

Add:

```
{
  "default-address-pools": [
    {
      "base": "172.17.0.0/12",
      "size": 24
    },
    {
      "base": "192.168.0.0/16",
```

```
    "size": 24
   },
   {
    "base": "10.100.0.0/16",
    "size": 24
   }
  ]
}
```

> If `/etc/docker/daemon.json` already exists, merge the `default-address-pools` key into it instead of overwriting the file. Pick pools that do not overlap your LAN or VPN ranges.

Restart Docker and check that a new network gets a /24 from the pools:

```
sudo systemctl restart docker
sudo docker network create testnet && sudo docker network inspect testnet | grep Subnet && sudo docker network rm testnet
```

The `Subnet` line should show a /24 from one of the pools. `sudo docker info` also lists them under `Default Address Pools`.

7) Optional: run docker without sudo

> The `docker` group is root-equivalent. Anyone in that group can control the daemon and, from there, the host. Do not add accounts you would not give root. For stronger isolation, use rootless mode (see Sources) or keep using `sudo docker`.

```
sudo usermod -aG docker YOUR_USER
```

Log out and back in (or reboot a VM) so the group is applied. Then:

```
docker run --rm hello-world
```

If you used `sudo docker` earlier and see a permission error on `~/.docker/config.json`:

```
sudo chown "YOUR_USER":"YOUR_USER" /home/YOUR_USER/.docker -R
sudo chmod g+rwx "$HOME/.docker" -R
```

8) firewalld and SELinux (leave them on)

Keep **SELinux enforcing**. Do not set it to permissive for this install.

Keep **firewalld** enabled. When Docker's iptables integration is on (the default), Docker creates a firewalld zone named `docker` with target `ACCEPT`, puts bridge interfaces such as `docker0` in that zone, and adds a `docker-forwarding` policy from ANY zone to the `docker` zone. You do not need to disable firewalld for published ports on the Docker host.

```
sudo getenforce
sudo systemctl is-active firewalld
sudo firewall-cmd --get-active-zones
```

Do not open the Docker socket on TCP (2375/2376) unless you have a TLS plan. This page leaves the daemon on the default Unix socket only.

9) Upgrade

With the repo in place, upgrade the same packages:

```
sudo dnf upgrade docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
sudo systemctl restart docker
```

10) Uninstall

```
sudo dnf remove docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin docker-ce-rootless-extras
```

That does not delete images, containers, volumes or custom config. Only if you intend to wipe local Docker data:

```
sudo rm -rf /var/lib/docker
sudo rm -rf /var/lib/containerd
```

Remove any files you edited under `/etc/docker/` yourself.

The `get.docker.com` convenience script is not recommended on production RHEL hosts. Use the repository steps above.

Sources

- https://docs.docker.com/engine/install/rhel/
- https://docs.docker.com/engine/install/linux-postinstall/
- https://docs.docker.com/engine/network/packet-filtering-firewalls/
- https://docs.docker.com/engine/security/rootless/
- https://docs.docker.com/engine/install/security/#docker-daemon-attack-surface

---

Revision #2
Created 8 October 2026 23:07:21 by Admin
Updated 24 October 2026 14:10:46 by Admin