Docker Documentation Install Docker:
update system

Created	@September 4, 2025 2:52 PM
Class	career ready
Last edited time	@September 4, 2025 5:26 PM

Docker Installation on Ubuntu

This guide explains how to install **Docker Engine** and its associated tools on Ubuntu, ensuring a clean, reliable setup for containerized applications.

---

1. Update and Upgrade the System

Always start by updating your package lists and upgrading existing packages to ensure your system is up-to-date:

```
sudo apt update && sudo apt upgrade -y
```

- `apt update` → refreshes the package index.
- `apt upgrade -y` → upgrades installed packages automatically.

---

2. Install Required Prerequisites

Install essential packages required for adding Docker repositories and handling HTTPS:

```
sudo apt install -y ca-certificates curl gnupg lsb-release
```

- `ca-certificates` → ensures secure HTTPS connections.
- `curl` → command-line tool to download files.
- `gnupg` → manages encryption keys.
- `lsb-release` → retrieves Ubuntu release information.

---

3. Add Docker's GPG Key

Docker packages are signed. Add the official GPG key to verify package authenticity:

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearm
or -o /etc/apt/keyrings/docker.gpg
```

- `mkdir -p /etc/apt/keyrings` → creates directory for key storage.
- `curl -fsSL` → downloads the GPG key securely.
- `gpg --dearmor` → converts the key to a format usable by APT.

---

4. Add Docker Repository

Add Docker's official APT repository to install Docker Engine:

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.
gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/
null
```

- `dpkg --print-architecture` → ensures repository matches your CPU architecture.
- `lsb_release -cs` → adds the correct Ubuntu release codename.
- `stable` → ensures you install the stable Docker version.

---

5. Install Docker Engine

Update APT and install Docker components:

```
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

- `docker-ce` → Docker Community Edition engine.
- `docker-ce-cli` → command-line interface.
- `containerd.io` → container runtime.
- `docker-buildx-plugin` → advanced build functionality.
- `docker-compose-plugin` → for multi-container applications.

---

6. Enable Docker to Start on Boot

Start Docker now and ensure it automatically starts at system boot:

```
sudo systemctl enable docker
sudo systemctl start docker
```

- `enable` → auto-start on boot.
- `start` → starts Docker immediately.

---

7. Verify Docker Installation

Test that Docker is installed and running:

```
docker --version
docker run hello-world
```

- `docker --version` → prints installed Docker version.
- `docker run hello-world` → runs a test container to verify installation.

Docker CLI Basics and Container Management

1. Check Docker Version

Verify that Docker is installed:

```
docker --version
```

---

2. Search and Pull Images

- **Search for images on Docker Hub:**

```
docker search nginx
```

- **Pull an image from Docker Hub:**

```
docker pull nginx
```

- **List all downloaded images:**

```
docker images
```

---

3. Running Containers

- **Run a container using an image:**

```
docker run nginx
```

- **Run with port mapping and volume:**

```
sudo docker run -d --name mynginx -p 8080:80 \
-v mynginx_data:/usr/share/nginx/html \
--restart unless-stopped nginx
```

Explanation:

- `d` → runs container in detached mode.
- `-name mynginx` → assigns a name to the container.
- `p 8080:80` → maps host port 8080 to container port 80.
- `v mynginx_data:/usr/share/nginx/html` → mounts a volume for persistent storage.
- `-restart unless-stopped` → automatically restarts container unless manually stopped.

---

4. Accessing a Container

- **Enter a running container:**

```
docker exec -it <containerID> bash
```

or if bash is not available:

```
docker exec -it <containerID> sh
```

- **Inside the container, you can run commands like:**

```
apt update
apt install nano
nano name.txt
```

- **Exit the container:**

```
exit
```

---

5. Copying Files Between Host and Container

- **From container to host:**

```
docker cp <containerID>:<Source> <Destination>
```

Example:

```
docker cp 1234:containerfile.txt machinefile.txt
```

- **From host to container:**

```
docker cp <Source> <containerID>:<Destination>
```

Example:

```
docker cp machine.txt 1234:container.txt
```

---

6. Creating Images from Containers

- **Create a new image from a running container:**

```
docker commit <containerID>
```

- **Assign a custom name and tag:**

```
docker commit <containerID> myusername/nginx:7.6
```

- **Verify created images:**

```
docker images
```

- **Run a container from the created image:**

```
docker run <imageID>
```