# Update #16 - Installing Docker and Prepping My VPS for App Expandability.

**Date:** May 23, 2025
**Category:** Server Management / DevOps
**Backlink:** *N/A (First Entry in Docker Expansion Series)*

My objective was to get Docker up and running on my existing VPS without migrating BookStack (yet). I wanted to ensure I could containerize and run additional apps while keeping BookStack fully operational, with no downtime during this transition. This setup gives me the flexibility to explore other projects in isolated environments while preserving my current production setup.

Setting this up can be found here:

https://docs.docker.com/engine/install/ubuntu/

I'll run apt update to make sure my packages are updated.

```
sudo apt update
```

```
~$ sudo apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Hit:2 https://repository.monarx.com/repository/ubuntu-jammy jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:7 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease
Get:8 https://apt.syncthing.net syncthing InRelease [18.3 kB]
Hit:9 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2591 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2591 kB]
Fetched 5968 kB in 2s (2859 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
14 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

I also used app auto-remove to get rid of some old PHP packages:

```
sudo apt autoremove
```

```
:~$ sudo apt autoremove
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  php8.1-bcmath php8.1-curl php8.1-gd php8.1-mysql
0 upgraded, 0 newly installed, 4 to remove and 10 not upgraded.
After this operation, 830 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 103788 files and directories currently installed.)
Removing php8.1-bcmath (8.1.2-1ubuntu2.21) ...
Removing php8.1-curl (8.1.2-1ubuntu2.21) ...
Removing php8.1-gd (8.1.2-1ubuntu2.21) ...
Removing php8.1-mysql (8.1.2-1ubuntu2.21) ...
Processing triggers for php8.1-fpm (8.1.2-1ubuntu2.21) ...
NOTICE: Not enabling PHP 8.1 FPM by default.
NOTICE: To enable PHP 8.1 FPM in Apache2 do:
NOTICE: a2enmod proxy_fcgi setenvif
NOTICE: a2enconf php8.1-fpm
NOTICE: You are seeing this message because you have apache2 package installed.
```

So to install Docker, we need to build our app source line step-by-step.

Getting my system architecture:

```
dpkg --print-architecture
```

```
:~$ dpkg --print-architecture
amd64
```

So I see it is amd64

I'll also need to get my Ubuntu Codename:

```
lsb_release -cs
```

```
:~$ lsb_release -cs
jammy
```

So I see it is jammy

Now I can construct the Docker source line manually:

```
echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable"
```

This code below writes it into the APT sources list at /etc/apt/sources.list.d/docker.list

```
echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Now we can update the packages:

```
sudo apt update
```

> I got an error like this

```
~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:7 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:9 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:10 https://apt.syncthing.net syncthing InRelease [18.3 kB]
Hit:11 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease
Hit:12 https://repository.monarx.com/repository/ubuntu-jammy jammy InRelease
Err:9 https://download.docker.com/linux/ubuntu jammy InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 7E
Reading package lists... Done
W: GPG error: https://download.docker.com/linux/ubuntu jammy InRelease: The following signatures coul
cause the public key is not available: NO_PUBKEY 7EA0A9C3F273FCD8
E: The repository 'https://download.docker.com/linux/ubuntu jammy InRelease' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
```

So I need to re fetch and add Docker's GPG Key Securely:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
  sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Then enure it is readable by APT:

```
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

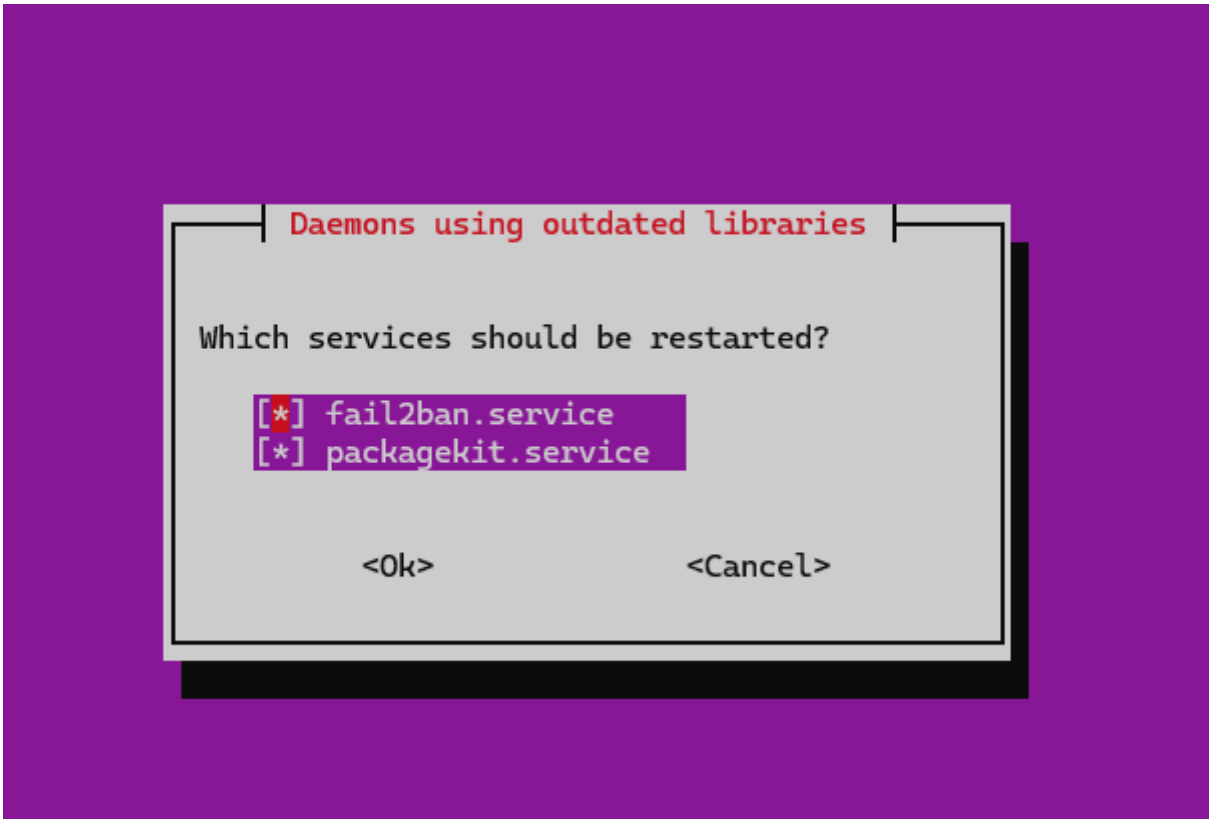
Then update the package list again:

```
sudo apt update
```

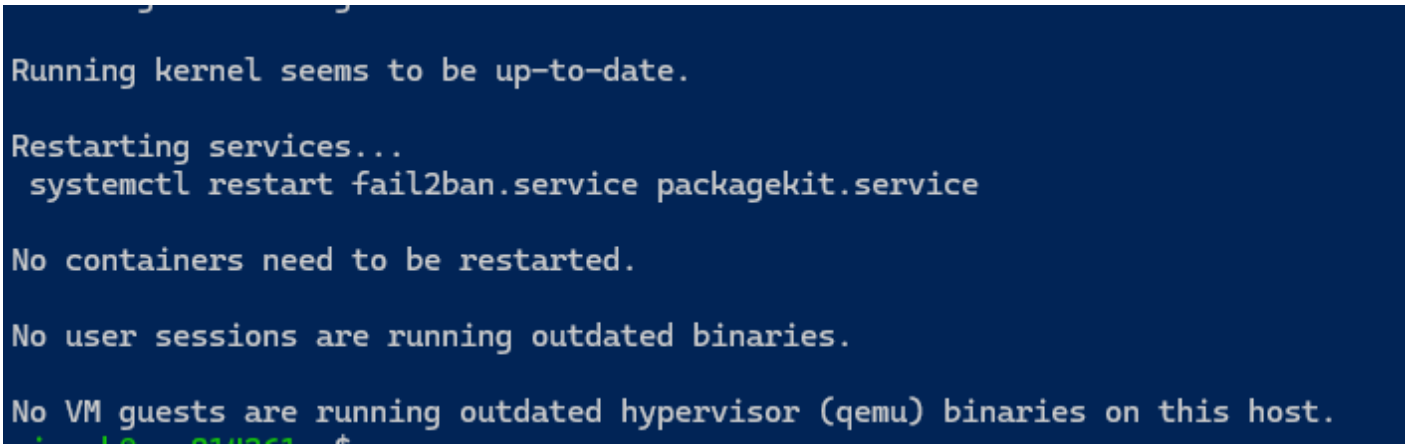
Proceed with installing Docker:

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

This installs the docker engine, CLI Tools, Container runtime,, and Docker Compose plugin (V2)




I left both services checked for restart and verified everything is installed.




> I'm now ready to use docker for some of my other projects and also didn't bring down the Bookstack at all while doing this.
> Now I can enable docker with these commands:

```
sudo systemctl enable docker
sudo systemctl start docker


sudo usermod -aG docker $USER
# Logout and back in or run: su - $USER
```

Testing the docker installation:

```
docker run hello-world
```

```
[redacted]:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

See the docker compose version:

```
docker compose version
```

```
[redacted]:~$ docker compose version
Docker Compose version v2.35.1
```

Now I can make some project directories and change ownership and look at this for reference on where the projects will go:

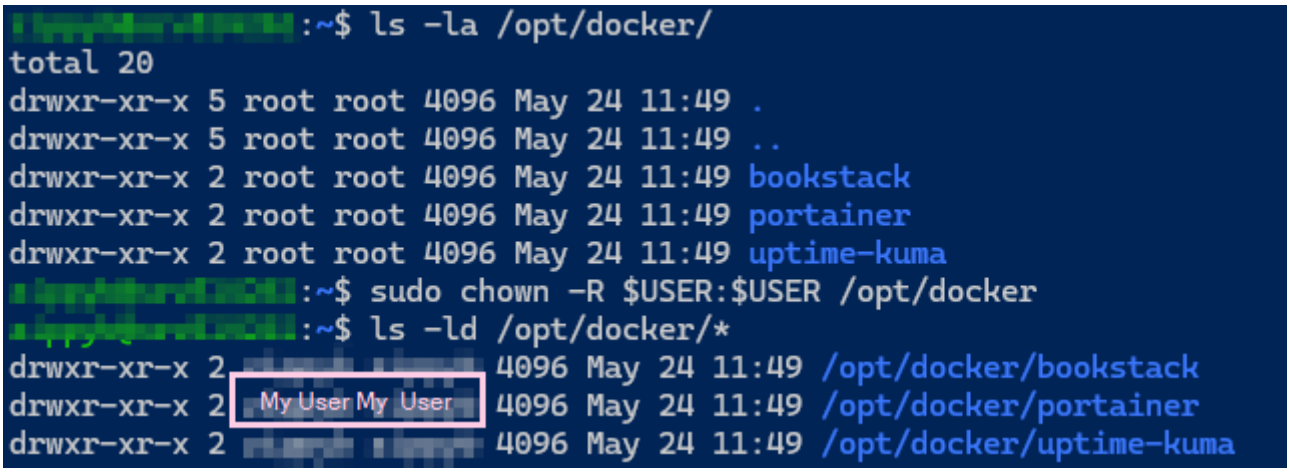
This makes 3 different folder projects:

```
sudo mkdir -p /opt/docker/{bookstack,uptime-kuma,portainer}
```

```
[redacted]:~$ sudo mkdir -p /opt/docker/{bookstack,uptime-kuma,portainer}
[sudo] password for [redacted]
[redacted]:~$ ls -la /opt/docker/
total 20
drwxr-xr-x 5 root root 4096 May 24 11:49 .
drwxr-xr-x 5 root root 4096 May 24 11:49 ..
drwxr-xr-x 2 root root 4096 May 24 11:49 bookstack
drwxr-xr-x 2 root root 4096 May 24 11:49 portainer
drwxr-xr-x 2 root root 4096 May 24 11:49 uptime-kuma
```

Now I can change ownership to my user for these folders:

```
sudo chown -R $USER:$USER /opt/docker
```




I'm officially Docker ready.

---

Revision #2
Created 24 May 2025 02:00:44 by Nate Nash
Updated 7 June 2025 00:14:24 by Nate Nash