

# Assignment 2b: Create a Docker Container

**Name:** Atharva Agrawal

**RollNo:** 33303

**Batch:** K11

## (1) What is Docker?

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

### \* Pre-requisite before installing docker:

#### 1) Check KVM virtualization support:

##### What is KVM?

KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, `kvm.ko`, that provides the core virtualization infrastructure and a processor specific module, `kvm-intel.ko` or `kvm-amd.ko`.

Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.

KVM is open source software. The kernel component of KVM is included in mainline Linux, as of 2.6.20. The userspace component of KVM is included in mainline QEMU, as of 1.3.

```
(base) atharva@coding-atharva:~$ modprobe kvm
(base) atharva@coding-atharva:~$
(base) atharva@coding-atharva:~$ modprobe kvm_intel
(base) atharva@coding-atharva:~$
(base) atharva@coding-atharva:~$ lsmod | grep kvm
kvm_intel          434176  0
kvm                1130496  1 kvm_intel
(base) atharva@coding-atharva:~$
(base) atharva@coding-atharva:~$
(base) atharva@coding-atharva:~$ ls -al /dev/kvm
crw-rw----+ 1 root kvm 10, 232 Feb 28 09:10 /dev/kvm
(base) atharva@coding-atharva:~$
(base) atharva@coding-atharva:~$
```

2) **QEMU must be version 5.2 or newer.** We recommend upgrading to the latest version.

### What is QEMU?

QEMU is a free and open-source emulator. It emulates the machine's processor through dynamic binary translation and provides a set of different hardware and device models for the machine, enabling it to run a variety of guest operating systems.

### 3) Setup the Repository:

```
(base) atharva@coding-atharva:~$ sudo apt-get install \
ca-certificates \
curl \
gnupg \
lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release set to manually installed.
ca-certificates is already the newest version (20211016ubuntu0.22.04.1).
ca-certificates set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
The following package was automatically installed and is no longer required:
  liblvm2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 16 not upgraded.
Need to get 484 kB of archives.
After this operation, 455 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl4 amd64 7.81.0-1ubuntu1.8 [290 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.8 [194 kB]
Fetched 484 kB in 16s (31.0 kB/s)
(Reading database ... 216557 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.81.0-1ubuntu1.8_amd64.deb ...
Unpacking libcurl4:amd64 (7.81.0-1ubuntu1.8) over (7.81.0-1ubuntu1.7) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.81.0-1ubuntu1.8_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.8) ...
Setting up libcurl4:amd64 (7.81.0-1ubuntu1.8) ...
Setting up curl (7.81.0-1ubuntu1.8) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
(base) atharva@coding-atharva:~$ sudo mkdir -m 0755 -p /etc/apt/keyrings
(base) atharva@coding-atharva:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
(base) atharva@coding-atharva:~$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
(base) atharva@coding-atharva:~$
```

### 4) Install Docker Engine, containerd, and Docker Compose.

```
sudo apt-get update
```

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

5) Verify that the Docker Engine installation is successful by running the hello-world image:

```
(base) atharva@coding-atharva:~/Downloads$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:6e8b6f026e0b9c419ea0fd02d3905dd0952ad1fee67543f525c73a0a790fefb
Status: Downloaded newer image for hello-world:latest

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

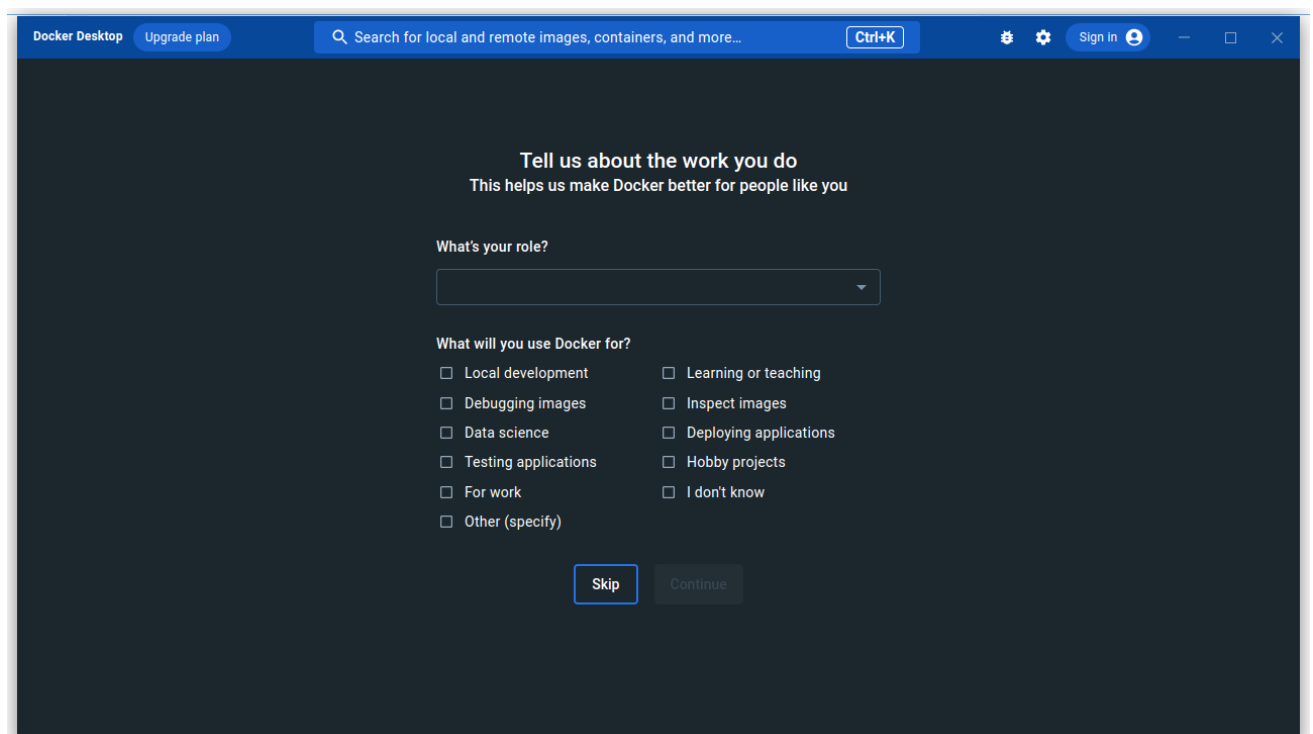
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

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

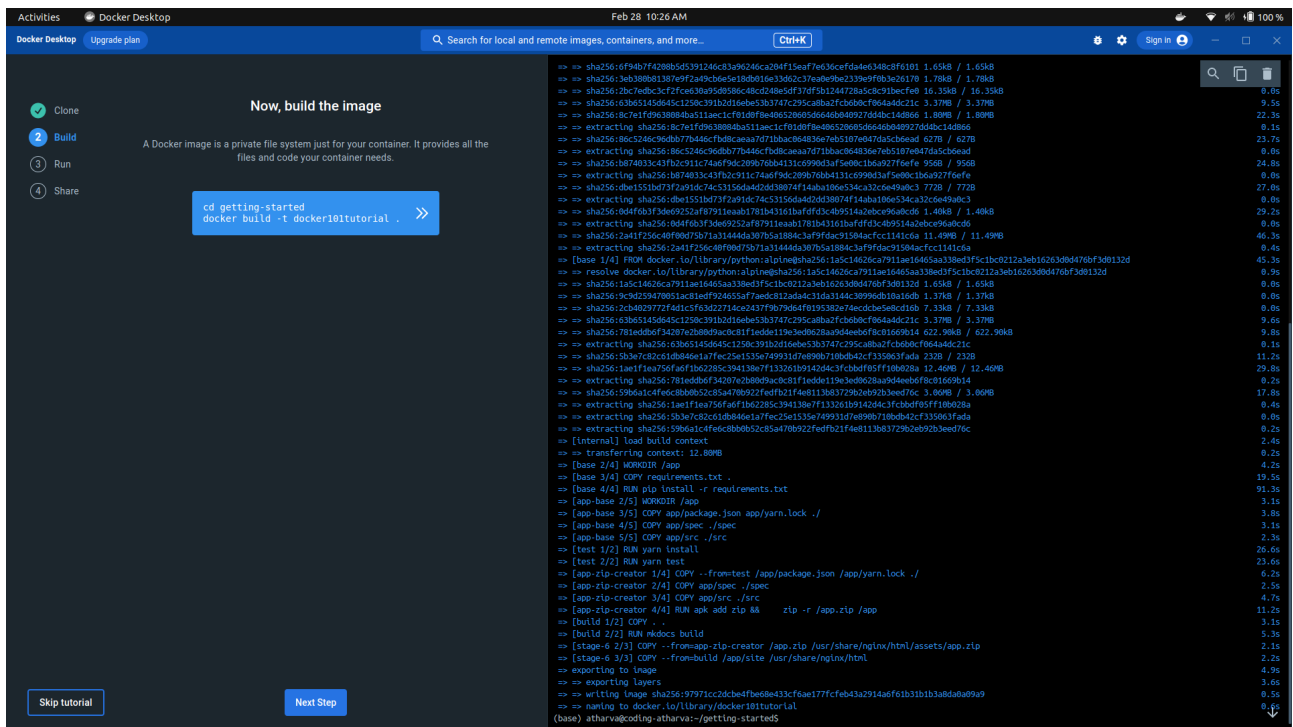
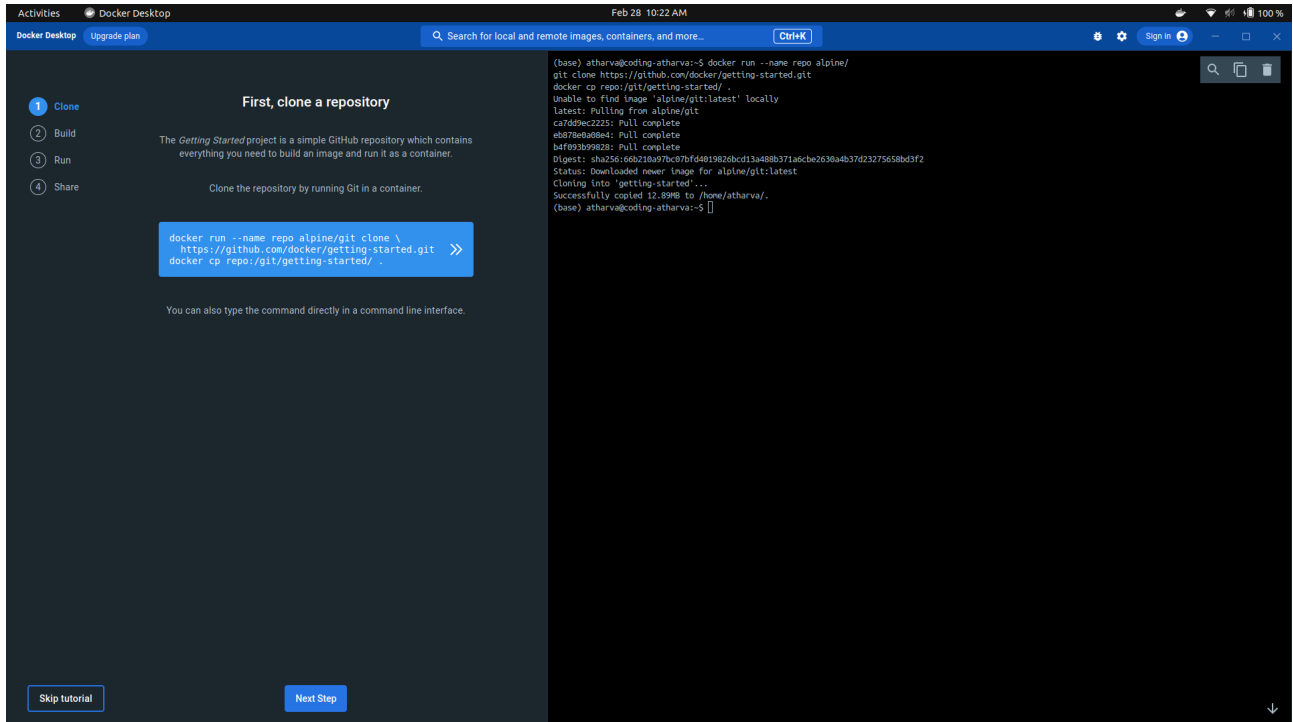
(base) atharva@coding-atharva:~/Downloads$
```

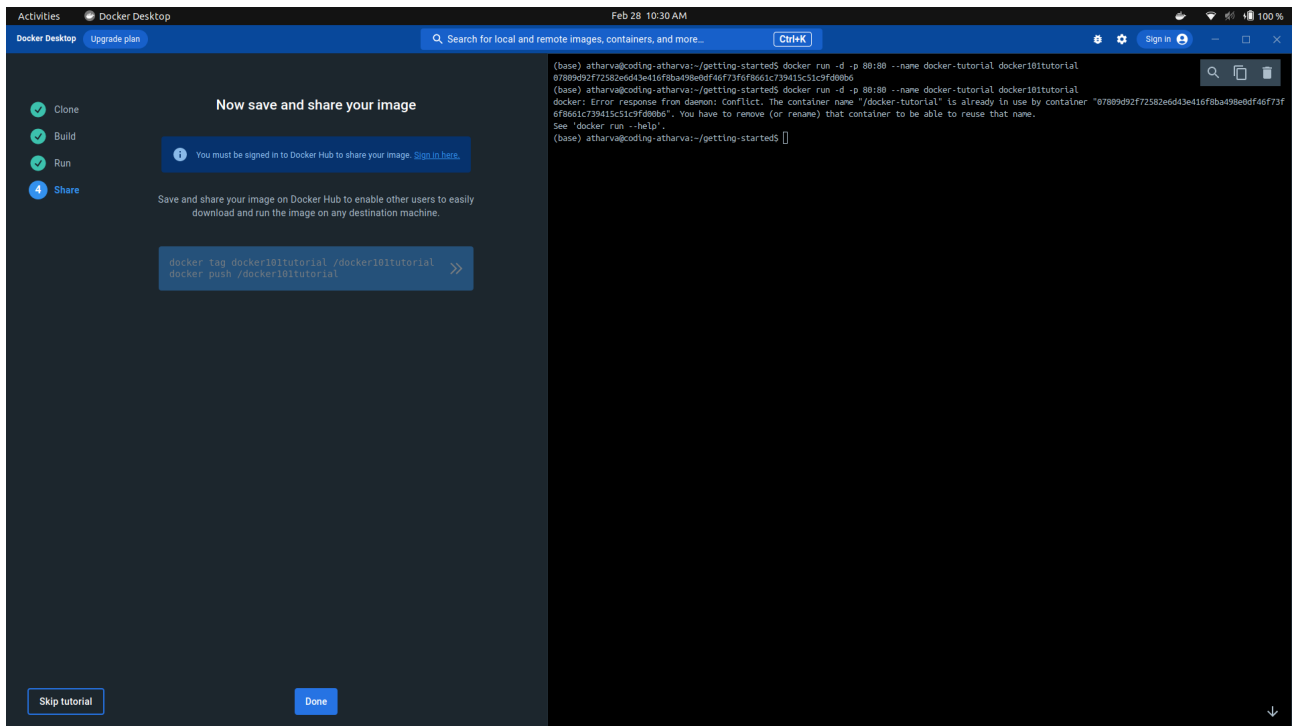
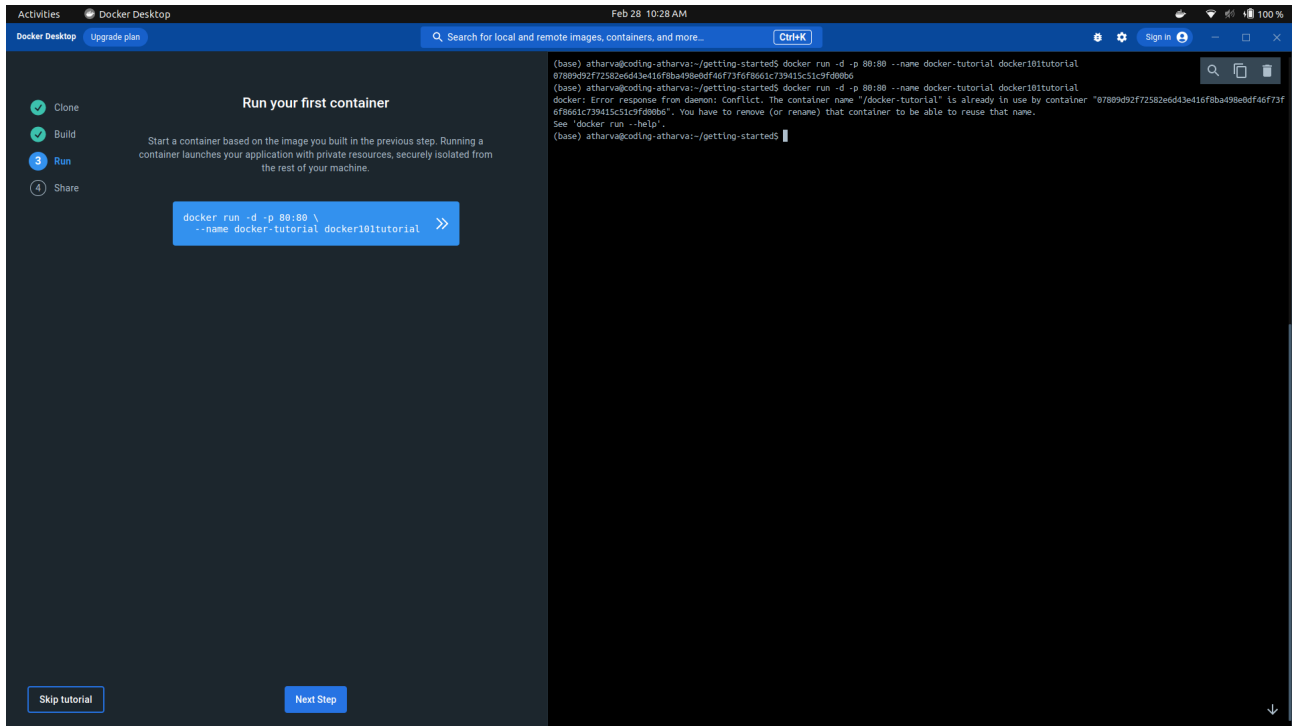
6) Install: `sudo apt-get install ./docker-desktop-4.17.0-amd64.deb`

7) Run: `systemctl --user start docker-desktop`

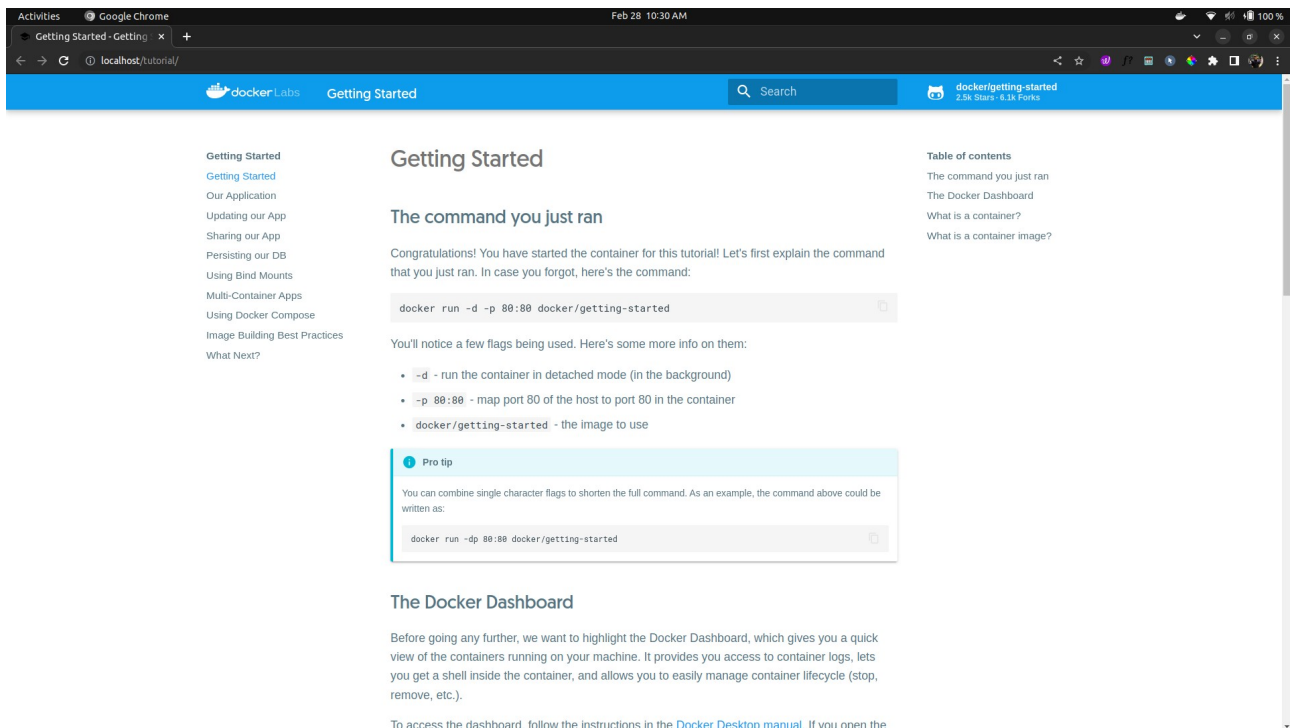


# Creating First Container:





## Successfully Started the Container:



The screenshot shows a web browser window displaying the Docker Labs 'Getting Started' tutorial. The page has a blue header with the Docker Labs logo and a search bar. The main content area is titled 'Getting Started' and includes a table of contents on the right. The table of contents lists: 'The command you just ran', 'The Docker Dashboard', 'What is a container?', and 'What is a container image?'. The main text area contains the following sections:

### Getting Started

#### The command you just ran

Congratulations! You have started the container for this tutorial! Let's first explain the command that you just ran. In case you forgot, here's the command:

```
docker run -d -p 80:80 docker/getting-started
```

You'll notice a few flags being used. Here's some more info on them:

- d - run the container in detached mode (in the background)
- p 80:80 - map port 80 of the host to port 80 in the container
- docker/getting-started - the image to use

**Pro tip**

You can combine single character flags to shorten the full command. As an example, the command above could be written as:

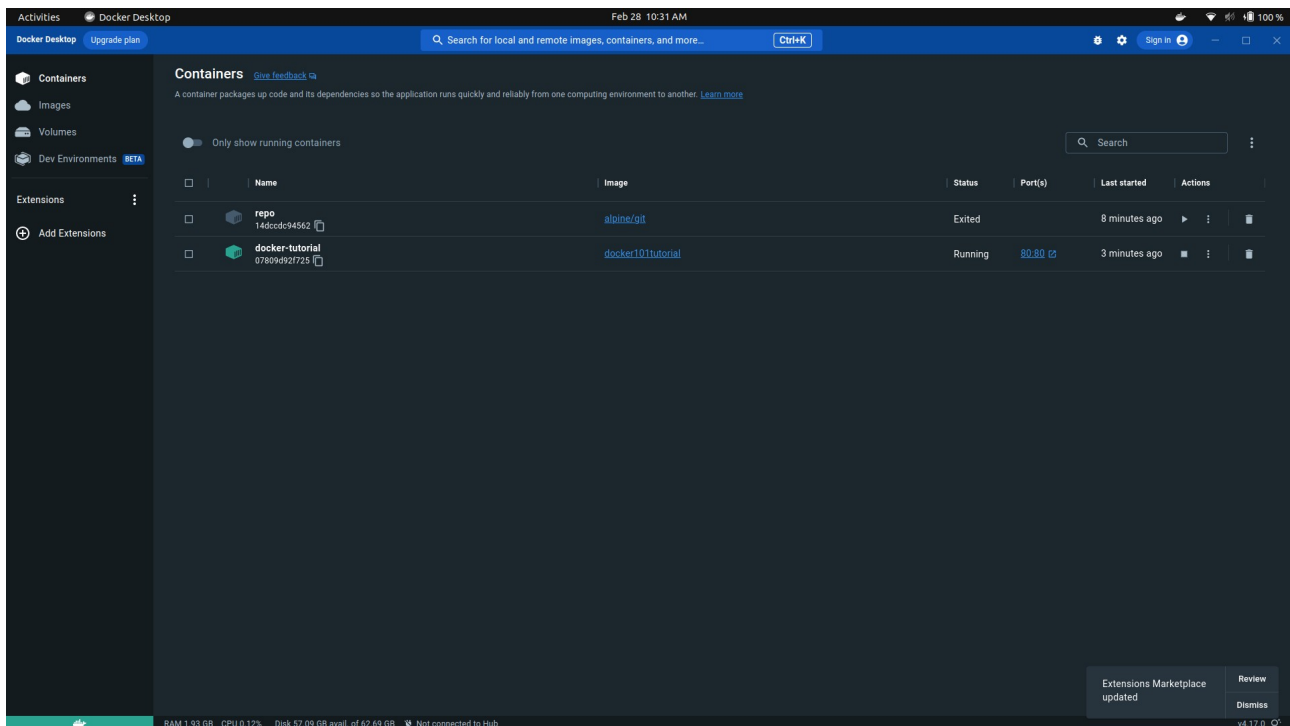
```
docker run -dp 80:80 docker/getting-started
```

#### The Docker Dashboard

Before going any further, we want to highlight the Docker Dashboard, which gives you a quick view of the containers running on your machine. It provides you access to container logs, lets you get a shell inside the container, and allows you to easily manage container lifecycle (stop, remove, etc.).

To access the dashboard, follow the instructions in the [Docker Desktop manual](#). If you open the

## Running Container:



The screenshot shows the Docker Desktop interface. The left sidebar contains navigation options: Containers, Images, Volumes, Dev Environments (BETA), Extensions, and Add Extensions. The main area is titled 'Containers' and includes a search bar and a toggle for 'Only show running containers'. A table lists the running containers:

Name	Image	Status	Port(s)	Last started	Actions
repo 14dc0c94362	alpine/git	Exited		8 minutes ago	▶ ⋮ 🗑
docker-tutorial 078094921725	docker101/tutorial	Running	80:80	3 minutes ago	■ ⋮ 🗑

The bottom status bar shows system information: RAM 1.93 GB, CPU 0.12%, Disk 57.09 GB avail. of 62.69 GB, and a note 'Not connected to Hub'. The version is v4.17.0.