<u>DevOps Lab</u>

Assignment 7:

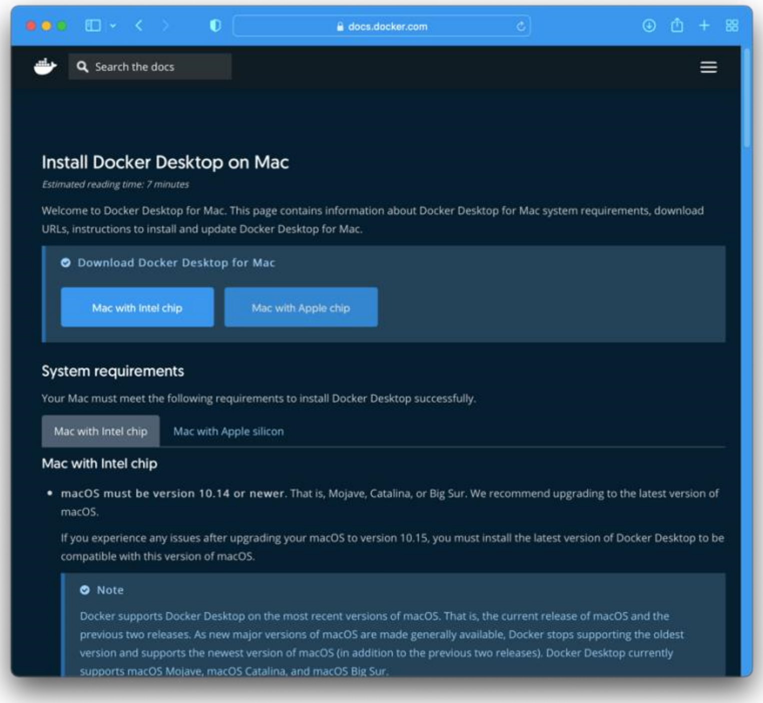
<u>**Aim:**</u> To install and understand the usage of containerization tools such as Docker in packaging and shipping light-weight platform independent apps.

<u>**Theory & Execution:**</u>

Docker Desktop is an easy-to-install application for your Mac or Windows environment that enables you to build and share containerized applications and microservices. Docker Desktop includes Docker Engine, Docker CLI client, Docker Compose, Docker Content Trust, Kubernetes, and Credential Helper.
Docker Desktop works with your choice of development tools and languages and gives you access to a vast library of certified images and templates in Docker Hub. This enables development teams to extend their environment to rapidly auto-build, continuously integrate, and collaborate using a secure repository. Some of the key features of Docker Desktop include:

- Ability to containerize and share any application on any cloud platform, in multiple languages and frameworks
- Easy installation and setup of a complete Docker development environment
- Includes the latest version of Kubernetes
- Automatic updates to keep you up to date and secure
- On Windows, the ability to toggle between Linux and Windows Server environments to build applications
- Fast and reliable performance with native Windows Hyper-V virtualization
- Ability to work natively on Linux through WSL 2 on Windows machines
- Volume mounting for code and data, including file change notifications and easy access to running containers on the localhost network
- In-container development and debugging with supported IDEs

To install Docker on Mac, we can go to Docker's website and download the installation file and install it and also check if install was successful on completion.



Install Docker Desktop on Mac

Estimated reading time: 7 minutes

Welcome to Docker Desktop for Mac. This page contains information about Docker Desktop for Mac system requirements, download URLs, instructions to install and update Docker Desktop for Mac.

Download Docker Desktop for Mac

System requirements

Your Mac must meet the following requirements to install Docker Desktop successfully.

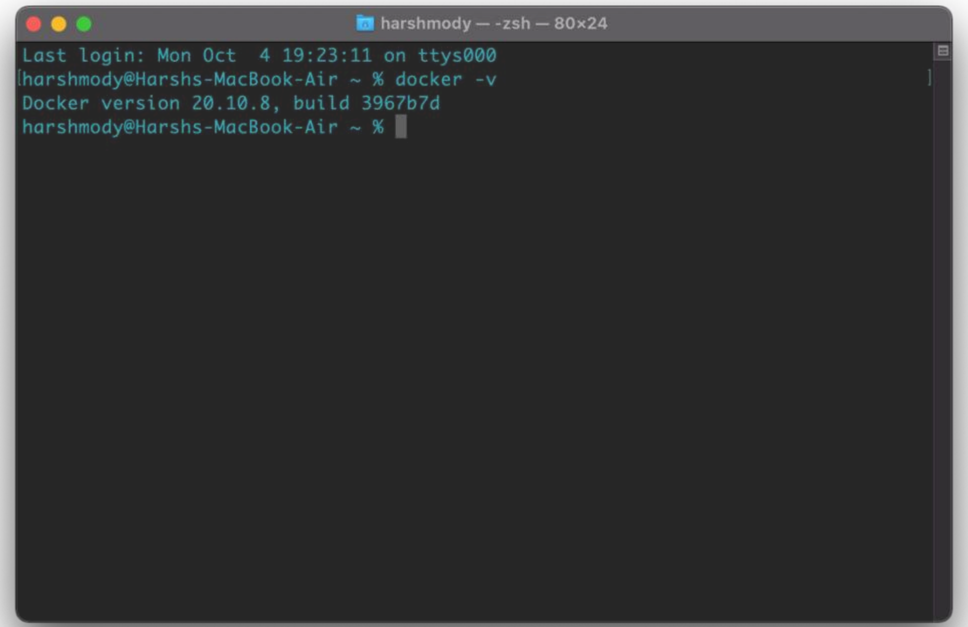
Mac with Intel chip

- **macOS must be version 10.14 or newer**. That is, Mojave, Catalina, or Big Sur. We recommend upgrading to the latest version of macOS.

  If you experience any issues after upgrading your macOS to version 10.15, you must install the latest version of Docker Desktop to be compatible with this version of macOS.

  Note

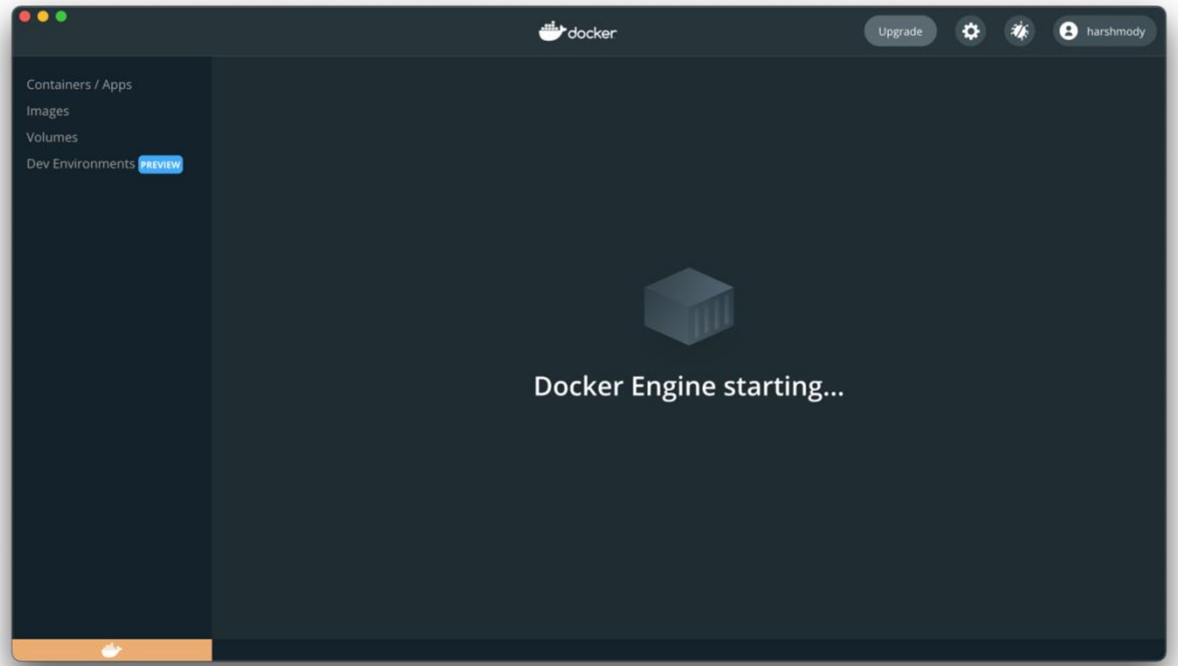
  Docker supports Docker Desktop on the most recent versions of macOS. That is, the current release of macOS and the previous two releases. As new major versions of macOS are made generally available, Docker stops supporting the oldest version and supports the newest version of macOS (in addition to the previous two releases). Docker Desktop currently supports macOS Mojave, macOS Catalina, and macOS Big Sur.



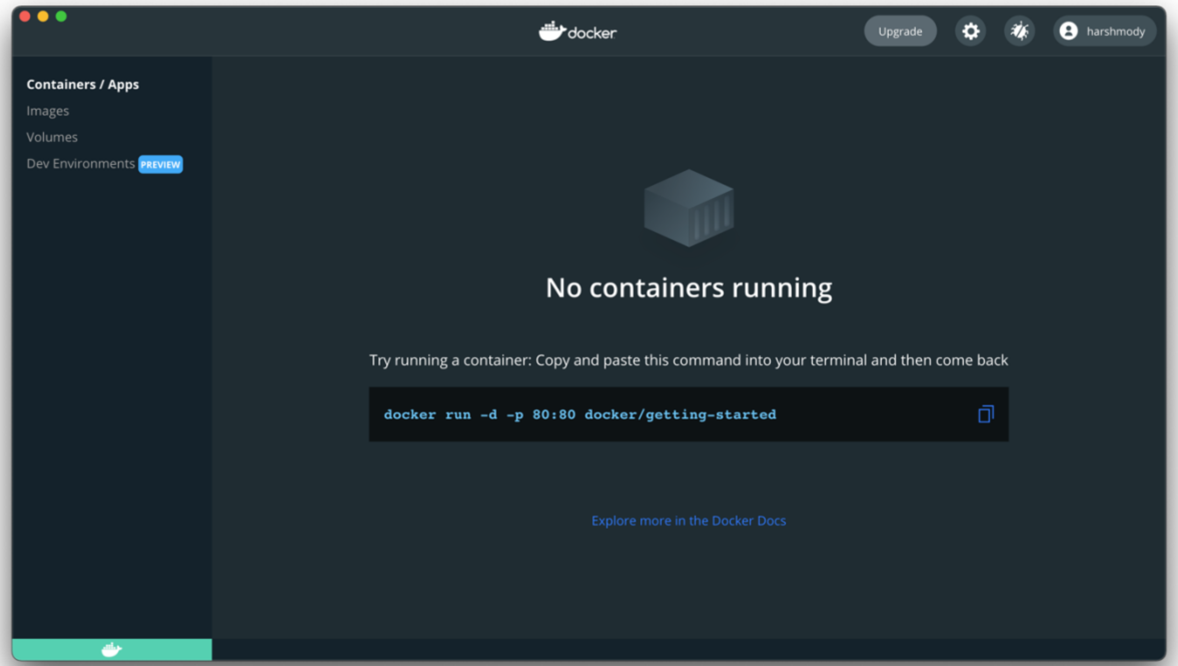
harshmody — -zsh — 80×24

```
Last login: Mon Oct  4 19:23:11 on ttys000
harshmody@Harshs-MacBook-Air ~ % docker -v
Docker version 20.10.8, build 3967b7d
harshmody@Harshs-MacBook-Air ~ %
```

To run docker, we need to start the Docker Engine which we can start by running the docker Desktop app and wait till it loads.



Docker Engine starting...



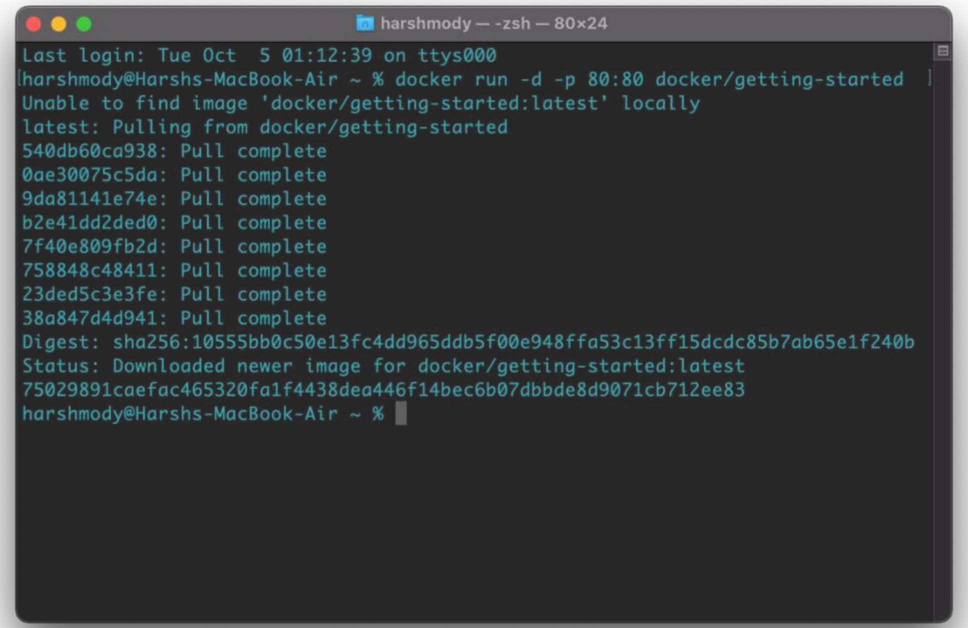
No containers running

Try running a container: Copy and paste this command into your terminal and then come back

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

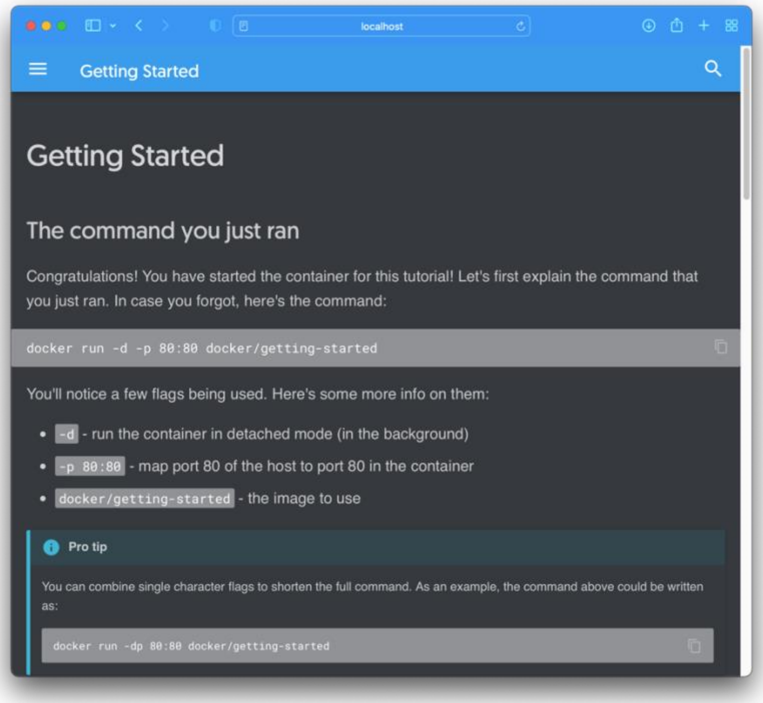
Explore more in the Docker Docs

We can then create a simple docker container using the command shown below.



```
Last login: Tue Oct  5 01:12:39 on ttys000
harshmody@Harshs-MacBook-Air ~ % docker run -d -p 80:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
540db60ca938: Pull complete
0ae30075c5da: Pull complete
9da81141e74e: Pull complete
b2e41dd2ded0: Pull complete
7f40e809fb2d: Pull complete
758848c48411: Pull complete
23ded5c3e3fe: Pull complete
38a847d4d941: Pull complete
Digest: sha256:10555bb0c50e13fc4dd965ddb5f00e948ffa53c13ff15dcdc85b7ab65e1f240b
Status: Downloaded newer image for docker/getting-started:latest
75029891caefac465320fa1f4438dea446f14bec6b07dbbde8d9071cb712ee83
harshmody@Harshs-MacBook-Air ~ %
```

We can check if our container was created successfully by visiting http://localhost:80/



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:

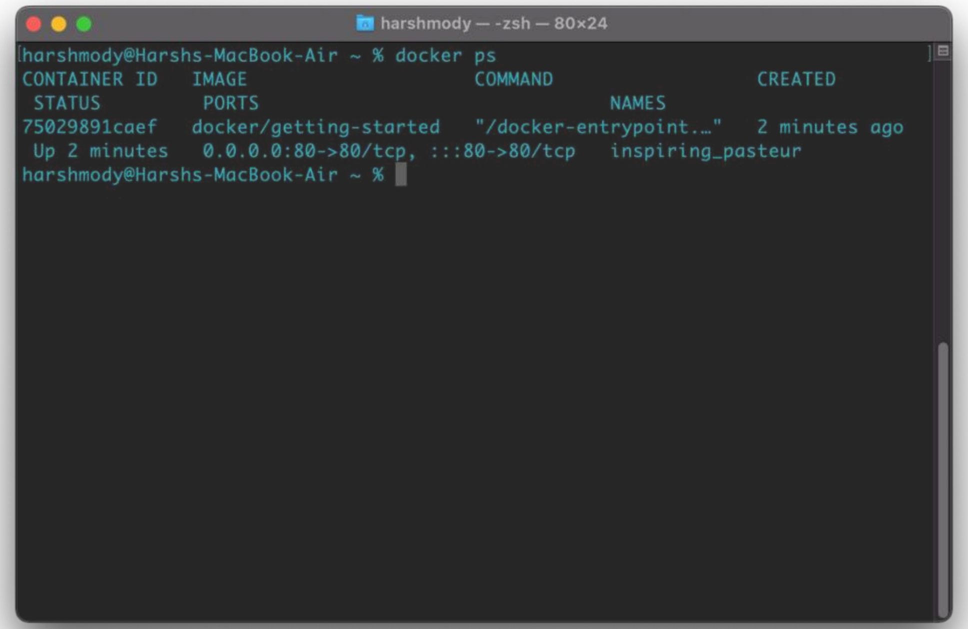
- `-d` - run the container in detached mode (in the background)
- `-p 80:80` - map port 80 of the host to port 80 in the container
- `docker/getting-started` - the image to use

Pro tip

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

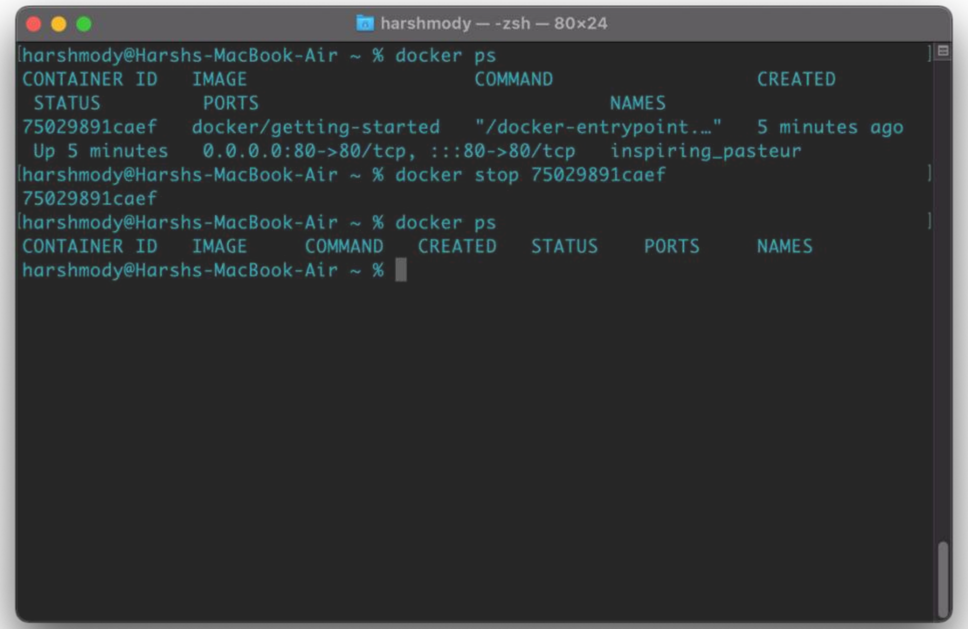
```
docker run -dp 80:80 docker/getting-started
```

docker ps command helps us no our container running their services and the port they are running at.

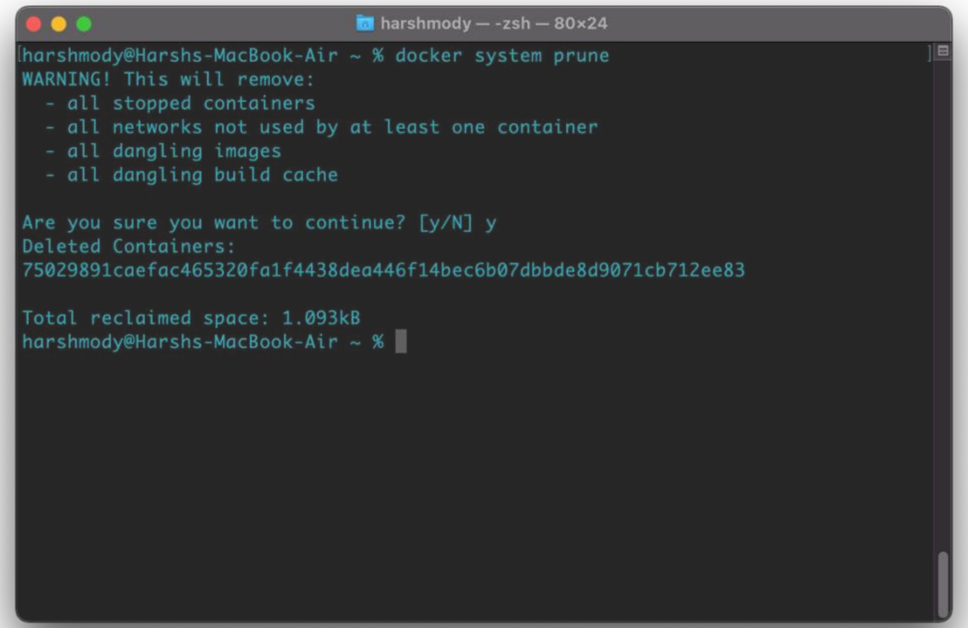


```
harshmody — -zsh — 80×24
harshmody@Harshs-MacBook-Air ~ % docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED
 STATUS          PORTS                           NAMES
75029891caef   docker/getting-started   "/docker-entrypoint.…"   2 minutes ago
 Up 2 minutes    0.0.0.0:80->80/tcp, :::80->80/tcp   inspiring_pasteur
harshmody@Harshs-MacBook-Air ~ %
```

We can then stop the docker container and run docker prune to delete all images to save storage space.



```
harshmody — -zsh — 80×24
harshmody@Harshs-MacBook-Air ~ % docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED
 STATUS          PORTS                           NAMES
75029891caef   docker/getting-started   "/docker-entrypoint.…"   5 minutes ago
 Up 5 minutes    0.0.0.0:80->80/tcp, :::80->80/tcp   inspiring_pasteur
harshmody@Harshs-MacBook-Air ~ % docker stop 75029891caef
75029891caef
harshmody@Harshs-MacBook-Air ~ % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
harshmody@Harshs-MacBook-Air ~ %
```



```
harshmody@Harshs-MacBook-Air ~ % docker system prune
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all dangling images
  - all dangling build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
75029891caefac465320fa1f4438dea446f14bec6b07dbbde8d9071cb712ee83

Total reclaimed space: 1.093kB
harshmody@Harshs-MacBook-Air ~ %
```

Conclusion: Thus, successfully understood the importance of Containerization tools like Docker and learnt Basic commands using Docker.