Understanding Docker: Inside the Engine Room

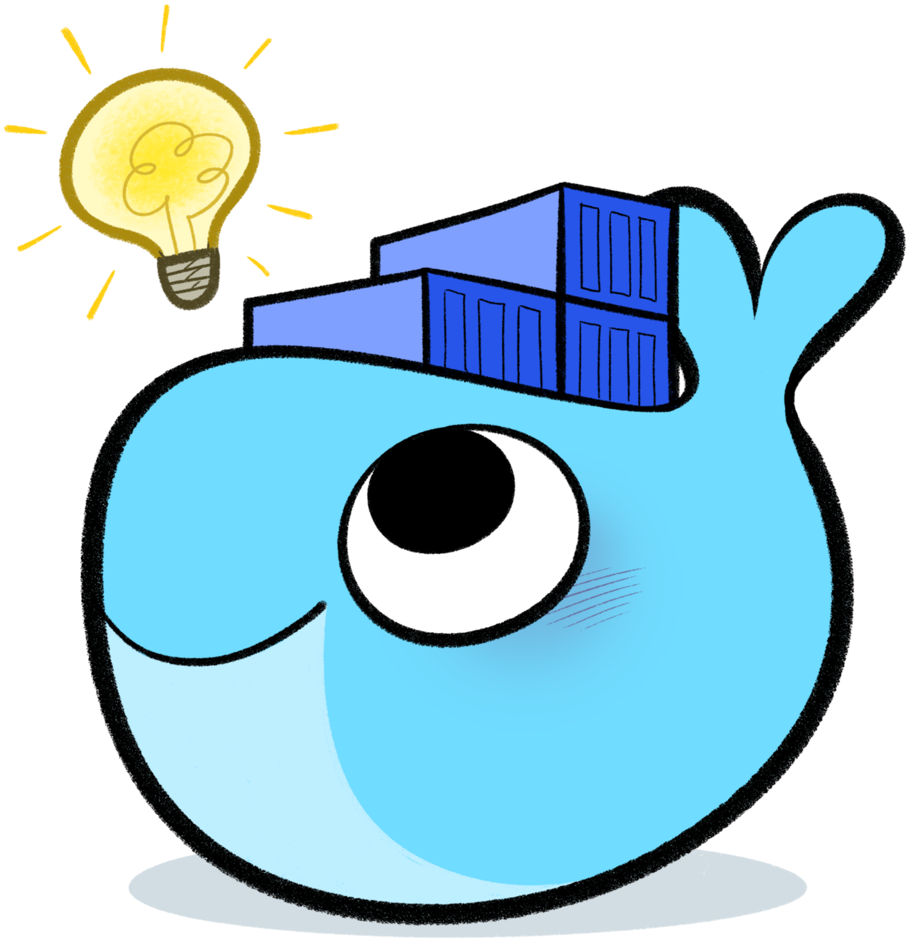
2.1. Docker’s Architecture

2.2. The Docker Daemon

2.3.The Docker Client

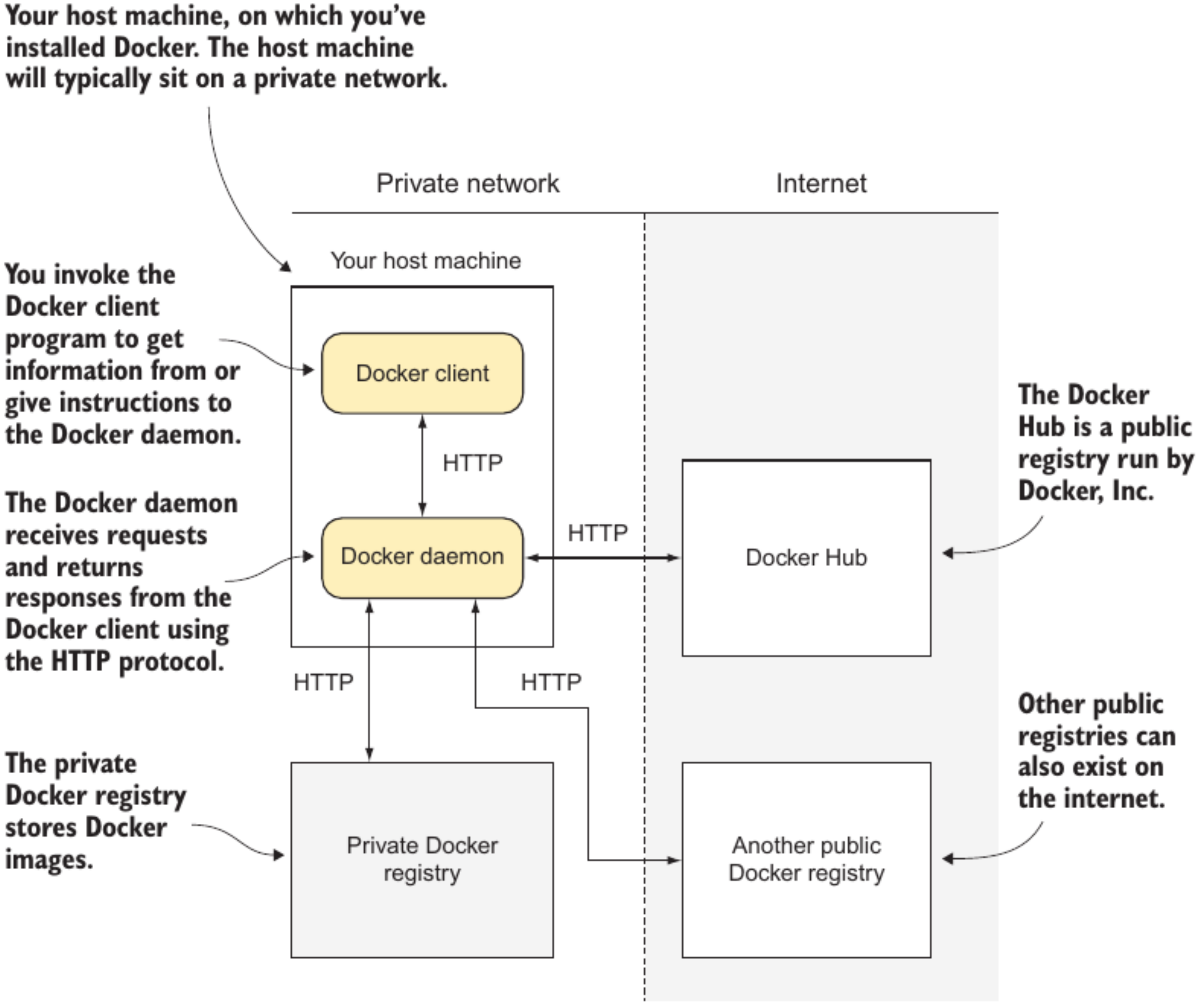
2.4 Docker Registries

2.5 Docker Hub



Source: https://www.docker.com/

Docker’s Architecture



Your host machine, on which you’ve installed Docker. The host machine will typically sit on a private network.

You invoke the Docker client program to get information from or give instructions to the Docker daemon.

The Docker daemon receives requests and returns responses from the Docker client using the HTTP protocol.

The private Docker registry stores Docker images.

The Docker Hub is a public registry run by Docker, Inc.

Other public registries can also exist on the internet.

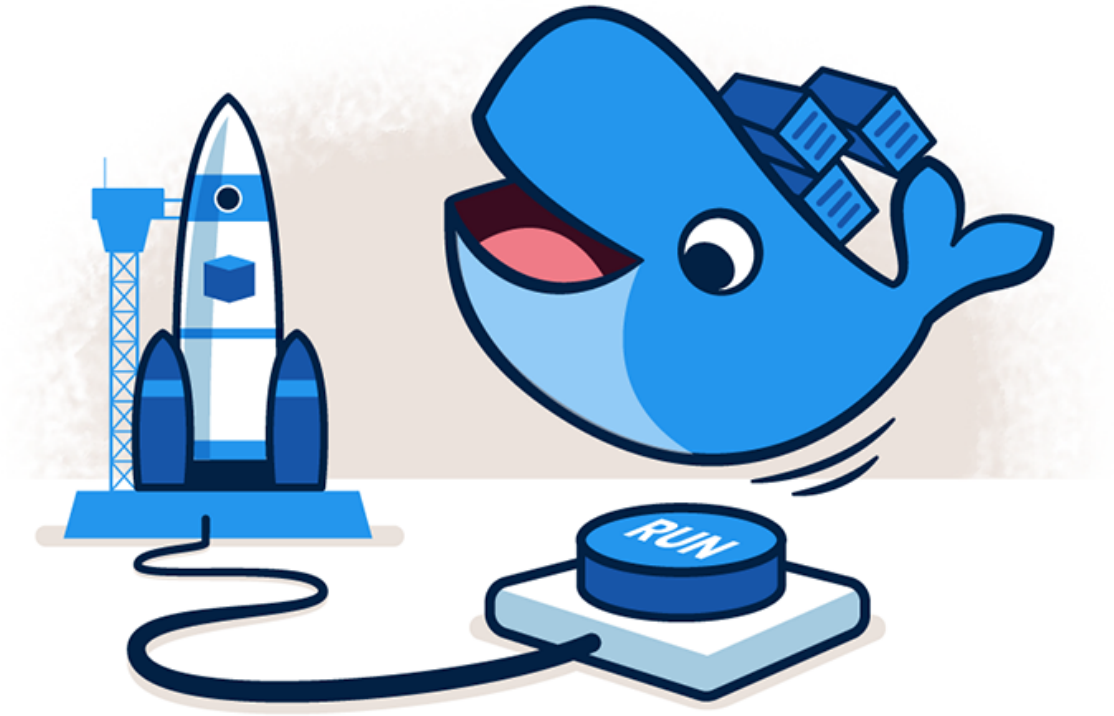
Figure 2.1 Overview of Docker’s architecture

2.2 The Docker Daemon

2.2.1 Open your Docker daemon to the world

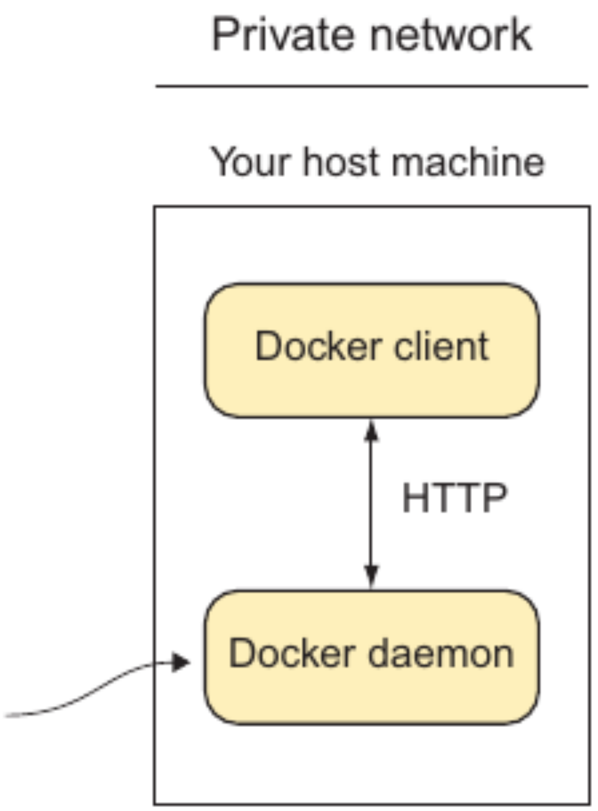
2.2.2 Running containers as daemons

2.2.3 Moving Docker to a different partition



Source: https://www.docker.com/

The Docker Daemon



The Docker daemon receives requests and returns responses from the Docker client using the HTTP protocol.

Figure 2.2 The Docker daemon

Teq 1. Open Your Docker Daemon to the World

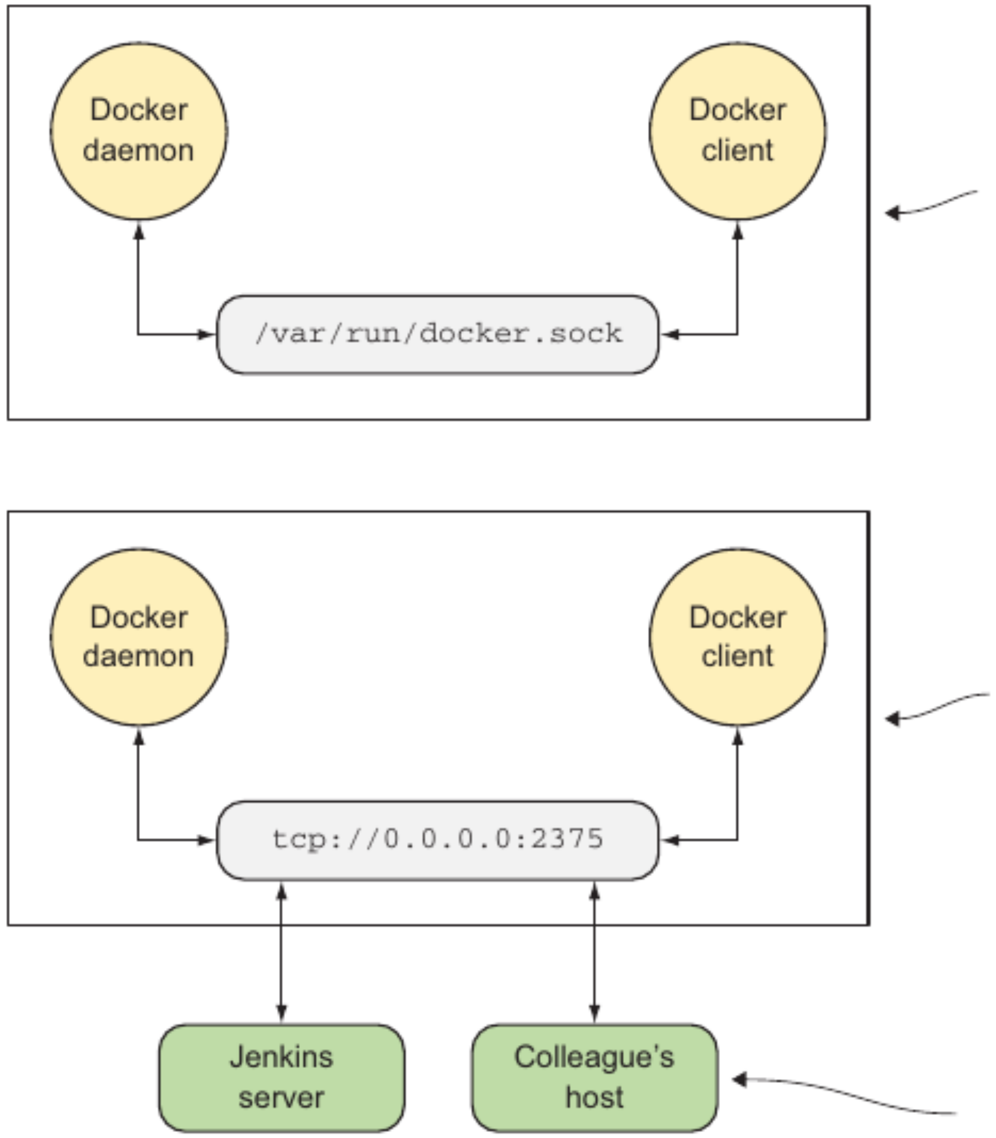
PROBLEM

You want to open your Docker server up for others to access.

SOLUTION

Start the Docker daemon with an open TCP address.

Figure 2.3 gives an overview of how this technique works.

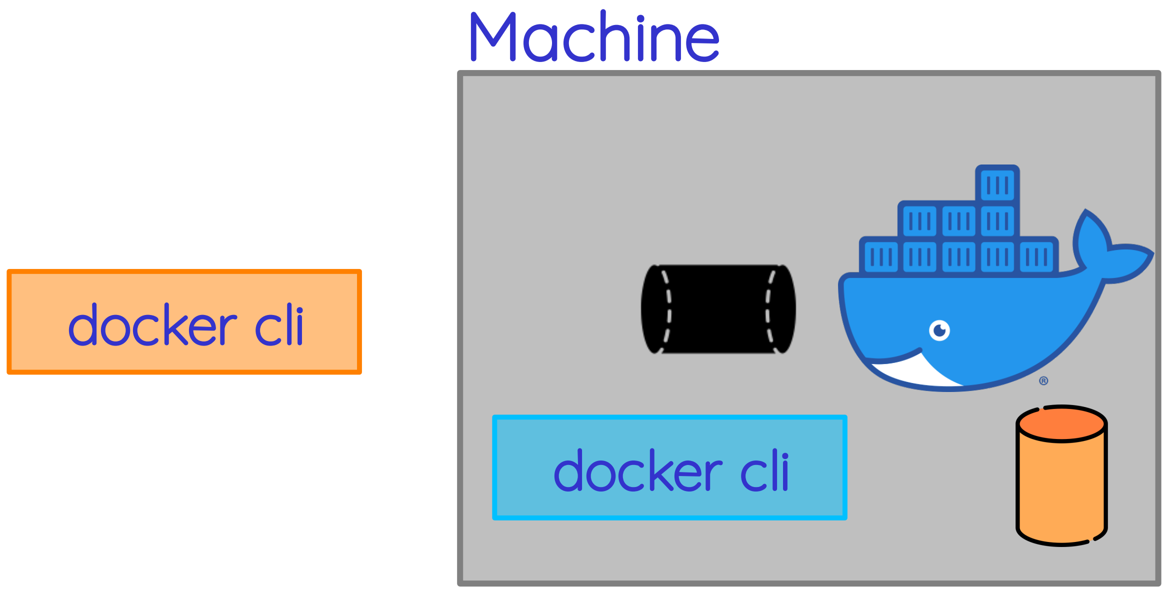


The default Docker configuration, where access is restricted via the /var/run/docker.sock domain socket. Processes external to the host can't gain access to Docker.

Using this technique's open access to the Docker daemon, access is gained through TCP socket 2375, available to all who can connect to your host (which is very insecure!).

Third parties can access the Docker daemon. The Jenkins server and colleague's host connect to the host's IP address on port 2375 and can read and write requests and responses using that channel.

Figure 2.3 Docker accessibility: normal and opened up



Teq 1. Open Your Docker Daemon to the World

To enable TCP port 2375 for external connection to Docker

Step 1: Create daemon.json file in /etc/docker

```
{"hosts": ["tcp://0.0.0.0:2375", "unix:///var/run/docker.sock"]}
```

Step 2: Add /etc/systemd/system/docker.service.d/override.conf

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd
```

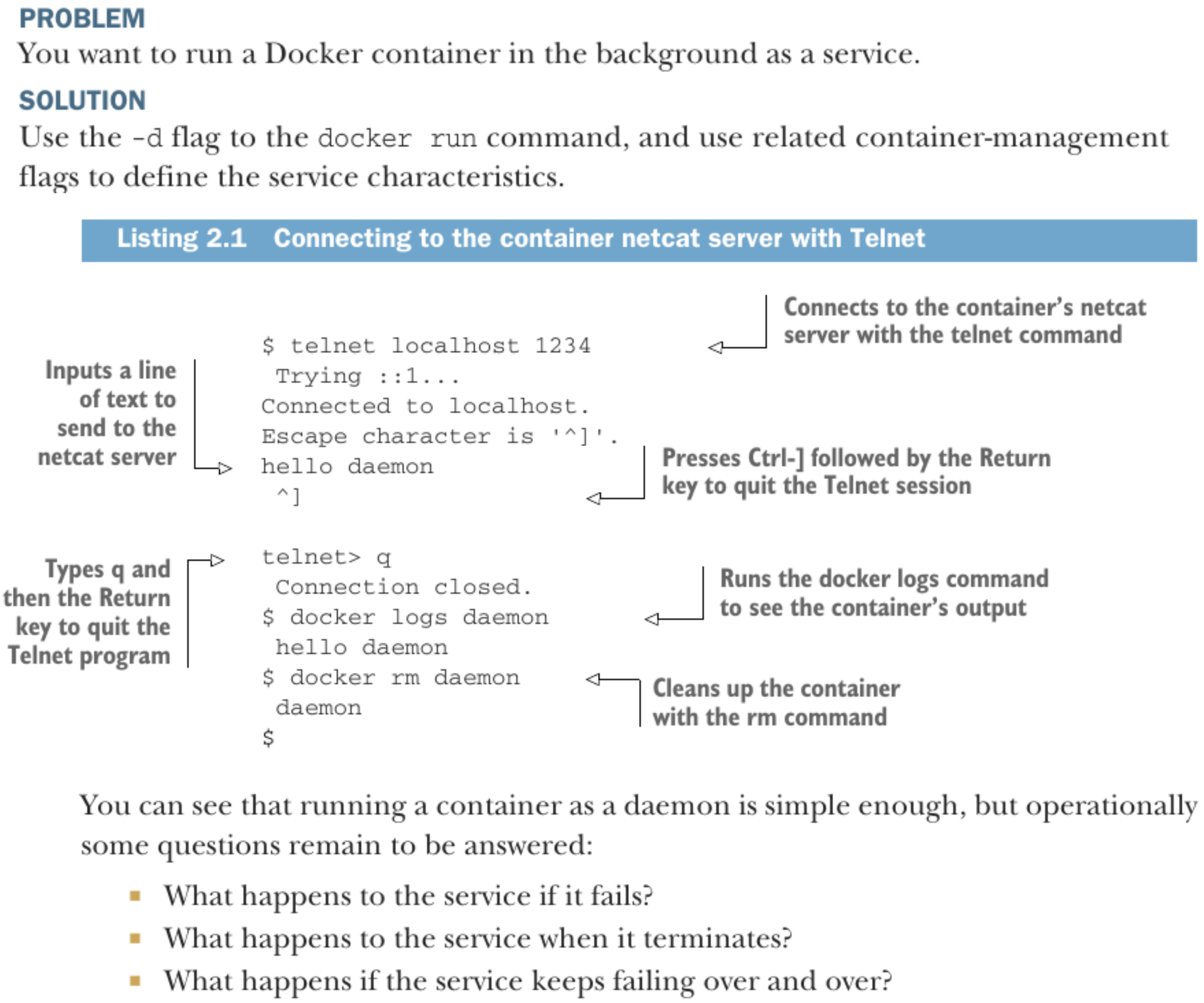
Step 3: Reload the systemd daemon:

```
sudo systemctl daemon-reload
```

Step 4: Restart docker:

```
sudo systemctl restart docker.service
```

Teq 2. Running Containers as Daemons



PROBLEM

You want to run a Docker container in the background as a service.

SOLUTION

Use the `-d` flag to the `docker run` command, and use related container-management flags to define the service characteristics.

Listing 2.1 Connecting to the container netcat server with Telnet

```
$ telnet localhost 1234
 Trying ::1...
Connected to localhost.
Escape character is '^]'.
hello daemon
 ^]

telnet> q
 Connection closed.
$ docker logs daemon
 hello daemon
$ docker rm daemon
 daemon
$
```

Connects to the container's netcat server with the telnet command

Inputs a line of text to send to the netcat server

Presses Ctrl-] followed by the Return key to quit the Telnet session

Types q and then the Return key to quit the Telnet program

Runs the docker logs command to see the container's output

Cleans up the container with the rm command

You can see that running a container as a daemon is simple enough, but operationally some questions remain to be answered:

- What happens to the service if it fails?
- What happens to the service when it terminates?
- What happens if the service keeps failing over and over?

Teq 2. Running Containers as Daemons

To Run Containers as Daemons

Step 1: Create a Dockerfile

```
FROM node
LABEL author "Deepika K deepikak@rvce.edu.in"
RUN git clone -q https://github.com/docker-in-practice/todo.git
WORKDIR todo
RUN npm install > /dev/null
EXPOSE 8000
CMD ["npm","start"]
```

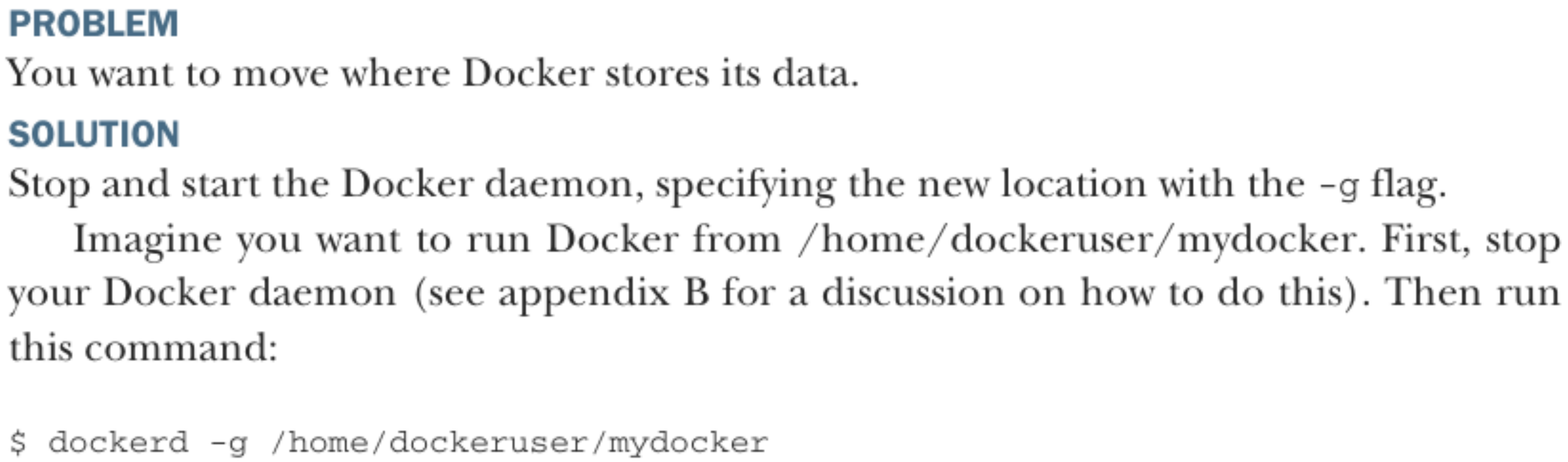
Step 2: Build the Dockerfile

```
sudo docker build -t daemon:1.0 .
```

Step 3: Run as a Container

```
sudo docker run -d daemon:1.0
```

Teq 3. Moving Docker to a Different Partition



PROBLEM

You want to move where Docker stores its data.

SOLUTION

Stop and start the Docker daemon, specifying the new location with the `-g` flag.

Imagine you want to run Docker from /home/dockeruser/mydocker. First, stop your Docker daemon (see appendix B for a discussion on how to do this). Then run this command:

```
$ dockerd -g /home/dockeruser/mydocker
```

Teq 3. Moving Docker to a Different Partition

To move the Docker to a different parititition

Step 1: Use systemd commands to stop Docker

```
sudo systemctl stop docker.service
sudo systemctl stop docker.socket
```

Step 2: To Create a new set of files in this location

```
dockerd -g /home/deepika/newdocker
```