

Important Docker Commands

In this section, let's review some important Docker commands.

Commands for Managing Images

Table 45-2 contains commands for managing Docker images.

Table 45-2. *Docker Commands for Managing Images*

Command	Description
<code>docker images</code>	List all images on the machine.
<code>docker rmi [IMAGE_NAME]</code>	Remove the image with name IMAGE_NAME on the machine.
<code>docker rmi \$(docker images -q)</code>	Remove all images from the machine.

Commands for Managing Containers

Table 45-3 contains commands for managing Docker containers.

Table 45-3. *Docker Commands for Managing Containers*

Command	Description
<code>docker ps</code>	List all containers. Append <code>-a</code> to also list containers not running.
<code>docker stop [CONTAINER_ID]</code>	Gracefully stop the container with [CONTAINER_ID] on the machine.
<code>docker kill CONTAINER_ID</code>	Forcefully stop the container with [CONTAINER_ID] on the machine.
<code>docker rm [CONTAINER_ID]</code>	Remove the container with [CONTAINER_ID] from the machine.
<code>docker rm \$(docker ps -a -q)</code>	Remove all containers from the machine.

Running a Docker Container

Let's break down the following command for running a Docker container:

```
docker run -d -it --rm --name [CONTAINER_NAME] -p 8081:80 [IMAGE_NAME]
```

where

- `-d` runs the container in detached mode. This mode runs the container in the background.
- `-it` runs in interactive mode, with a terminal session attached.
- `--rm` removes the container when it exits.
- `--name` specifies a name for the container.
- `-p` does port forwarding from host to the container (i.e., `host:container`).

Kubernetes

When a microservice application is deployed in production, it usually has many running containers that need to be allocated the right amount of resources in response to user demands. Also, there is a need to ensure that the containers are online, are running, and are communicating with one another. The need to efficiently manage and coordinate clusters of containerized applications gave rise to Kubernetes.

Kubernetes is a software system that addresses the concerns of deploying, scaling, and monitoring containers. Hence, it is called a container orchestrator. Examples of other container orchestrators in the wild are Docker Swarm, Mesos Marathon, and HashiCorp Nomad.

Kubernetes was built and released by Google as an open source software, which is now managed by the Cloud Native Computing Foundation (CNCF). Google Cloud Platform offers a managed Kubernetes service called Google Kubernetes Engine (GKE). Amazon Elastic Container Service for Kubernetes (EKS) also provides a managed Kubernetes service.