

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.

Features of Kubernetes

The following are some features of Kubernetes:

- **Horizontal auto-scaling:** Dynamically scales containers based on resource demands
- **Self-healing:** Re-provisions failed nodes in response to health checks
- **Load balancing:** Efficiently distributes requests between containers in a pod
- **Rollbacks and updates:** Easily update or revert to a previous container deployment without causing application downtime
- **DNS service discovery:** Uses Domain Name System (DNS) to manage container groups as a Kubernetes service

Components of Kubernetes

The main components of the Kubernetes engine are

- **Master node(s):** Manages the Kubernetes cluster. There may be more than one master node in high availability mode for fault-tolerance purposes. In this case, only one is the master, and the others follow.
- **Worker node(s):** Machine(s) that runs containerized applications that are scheduled as pod(s).

The illustration in Figure [45-6](#) provides an overview of the Kubernetes architecture.