



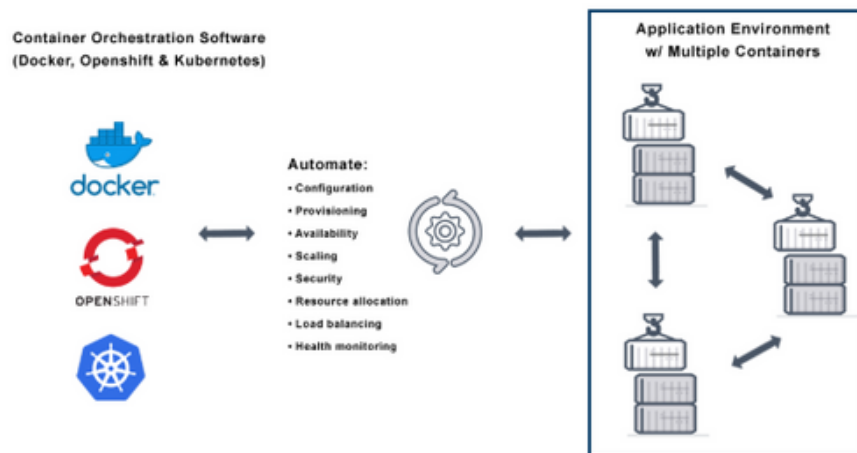
40

D A Y S



WHAT IS CONTAINER ORCHESTRATION?

Container orchestration is the automation of all aspects of coordinating and managing containers. Container orchestration is focused on managing the life cycle of containers and their dynamic environment. Also: Container orchestration is the automatic process of managing or scheduling the work of individual containers for applications based on microservices within multiple clusters. The widely deployed container orchestration platforms are based on open-source versions like Kubernetes, Docker Swarm or the commercial version from Red Hat OpenShift.



WHY DO WE NEED CONTAINER ORCHESTRATION?

- Configuring and scheduling of containers
- Provisioning and deployments of containers
- Availability of containers
- The configuration of applications in terms of the containers that they run in
- iScaling of containers to equally balance application workloads across infrastructure
- Allocation of resources between containers
- Health monitoring of containers
- Securing the interactions between containers.
- Load balancing, traffic routing and service discovery of containers

SOME CONTAINER ORCHESTRATION TOOLS



Kubernetes

is an open source project that has become one of the most popular container orchestration tools around; it allows you to deploy and manage multi-container applications at scale. While in practice Kubernetes is most often used with Docker, the most popular containerization platform, it can also work with any container system that conforms to the Open Container Initiative (OCI) standards for container image formats and runtimes. And because Kubernetes is open source, with relatively few restrictions on how it can be used, it can be used freely by anyone who wants to run containers, most anywhere they want to run them—on-premises, in the public cloud, or both.

Benefits:

- Large and vibrant developer/user community
- One-tier platform so any containerized Linux application can run on it
- Supports a multitude of applications; has the ability to run in excess of 1,000 nodes
- Establishing clusters is easy to do
- Large and active Slack and Stack Overflow communities support use of the product
- Uses “virtual” ports to avoid difficulties with port management
- Offers an add-on for easy logging

The Kubernetes cluster consists of:

What is a Cluster?

cluster resembles a group of nodes that works together to distribute the work load.

A master node is a node which controls and manages a set of worker nodes (workloads runtime) and resembles a cluster in Kubernetes. A master node has the following components to help manage worker nodes:

- 1-Kube-APIServer, which acts as the frontend to the cluster. All external communication to the cluster is via the API-Server.
- 2-Kube-Controller-Manager, which runs a set of controllers for the running cluster. The controller-manager implements governance across the cluster.
- 3-Etcd, the cluster state database.
- 4-Kube Scheduler, which schedules activities to the worker nodes based on events occurring on the etcd. It also holds the nodes resources plan to determine the proper action for the triggered event.

★ ★ ★ The master node does not run any containers, it just handles and manages the cluster. The nodes that actually run the containers are the worker nodes.