

# K8s in the Enterprise

The virtual machine (VM) revolution turned into a revelation—that hypervisors and multiple full-blown operating systems running on a single machine is... a massive footprint, expensive, complicated, and delivers a serious performance hit to your hardware.

Aside from that, and that VMs are nearly 20 years old, they're great!

But Kubernetes and containers are the next revolution that started 6+ years ago and EXPLODED into a cloud scale revelation.

Follow the trail through from VMs to containers and K8s—and then, get started in your real world journey!

## Key Terms

### Containers:

- ▶ Lightweight VMs that leverage the host OS kernel

### Kubernetes (K8s):

- ▶ Dominant platform for orchestrating containerized apps across clusters

### Controllers:

- ▶ Control loops that watch the state of a cluster, pushing it toward a declared spec for the desired state

### Container Metadata:

- ▶ Most container resources contain metadata, such as labels and annotations, desired state and observed state (current status) to abstract performance details for active management flexibility

### Pod:

- ▶ A collection of containers with a limited lifespan that eventually die due to rolling updates, wind down (pre-defined lifetime) or scale up/down actions



VMs are tightly coupled dinosaurs tangling OS, data, and configs in irreversible chaos. Embrace the Great Decoupling!

(LEAP TO CHAPTER 1)

- ▶ VM host OS + guest VMs w/their own OS = massive performance and cost overhead
- ▶ These "stacked OSes" are a house of cards just waiting for a stiff breeze, like OS updates
- ▶ Bottom line—VMs waste time, \$ and effort that distract from adding value at app layer



K8s has 3 key architecture pieces: the control plane (master), distributed storage for cluster state mgmt (etcd), and cluster nodes (Kubelets)

(LEAP TO CHAPTER 2)

- ▶ The control plane tracks all objects, manages object states, and responds to changes; managing the actual state to match the desired state (essentially built-in SLA management)
- ▶ Persistent storage survives pod restarts to maintain the state of the pod despite frequently changing containers
- ▶ Storage types can be mixed and matched and consumed by any containers in a pod based on dynamic needs



It's nodes & clusters time! Scaling and preventing multi- or single-cluster failures requires rethinking architectural design for a new way of building

(LEAP TO CHAPTER 3)

- ▶ Public clouds should be first step to experimenting with K8s, with native integration to public clouds to configure storage, compute, and networking automatically
- ▶ On-demand resources help keep your K8s "right-sized" and flexible to scale up and down



Managed K8s outsources the pain and hassles, so you can focus on driving value and velocity in your DevOps team

(LEAP TO CHAPTER 5)

- ▶ Reliable K8s platform availability powers quick deploys of new apps and new improvements
- ▶ K8s doesn't stand alone—taking serious advantage of this new infrastructure ties into CI/CD, observability, dashboards, and other layers
- ▶ Modern dev teams are going to need the rest of the toolkit to be upgraded or acquired for the first time along with K8s infrastructure

## Download the Full Gorilla Guide!

The Gorilla Guide to Kubernetes in the Enterprise is your resource to ensure the success of your Enterprise Kubernetes projects by thinking through critical decisions around deployment options, day-2 operational considerations, use cases, and choosing your Kubernetes implementation solutions.

### Highlights include:

- ▶ Why the Industry Has Embraced Containers
- ▶ Keys to Smoothing Your Kubernetes On-ramp
- ▶ Managed Kubernetes Provides the Optimal Experience

**GET YOUR COPY!**

