GORILLA GUIDE TRAIL MAP

Deploying Production-Grade Kubernetes

You can build Kubernetes clusters out of $50 Raspberry Pis to learn how to manage clusters (and your geek street cred is ++ as a result)...

But really, nothing like managing real-world containers in production to realize the challenges of unmanaged containers or Kubernetes at prod-Grade scale.

And you don't know what you don't know... until you follow this map through the confusing forest of containers and K8s.

Key Terms

Containers:

- Lightweight VMs that leverage the host OS kernel.

DevOps:

- Practices and tools for collaboration combining Dev and Ops teams.

Kubernetes (K8s):

- The most popular system for orchestrating containerized apps across clusters.

Auto-healing:

- Kubernetes feature in which unhealthy pods are replaced automatically without requiring DevOps to log in, identify failing pods, and replace them.

Canary Deployment:

- A small amount of traffic is routed to a new service. If there are problems, only a small number of users are impacted.

Why Kubernetes (K8s)? It's IT transformation & modernization time

(LEAP TO CHAPTER 1)

- Datacenter (DC) K8s clusters are extensions of DCs into a more modern infrastructure. (Tiny chimp steps)
- Micro DCs are for edge and distributed implementations with much slimmer needs and infrastructure availability, like cell tower implementations. (Gorilla leaps!)

Zero-touch, lights-out, automated and distributed infrastructure = dramatically fewer Ops headaches

(LEAP TO CHAPTER 1)

- Time to rethink Ops. Don't think racks and stacks, and oncall, onsite.
- Assume you can't physically touch your systems - automation and resilience is THE NORMAL default. (Keep your paws off my hardware!)

Manage distributed environments centrally, with less pain and fewer tools

(LEAP TO CHAPTER 1)

- In a hyperconnected world, clusters can be anywhere, and traditional DC tools can't reach out to the edge and PoPs on top of normal DC management.
- K8s has auto-healing, putting intelligence where clusters are, so they can self-manage. Hopefully they don't rise up and take over!

Continuous Integration and Continuous Deployment (CI/CD) plus containers = monkey nirvana

(LEAP TO CHAPTER 4)

- Container build processes should be automated using CI/ CD. Why not carry automation power all the way through?
- Reduces the workload on devs when deploying a service.
- Automated testing during pre-deployment, and rollback on failure, are massive wins for tight agile loops.

To cluster or not to cluster... that's NOT the question. Multi-cluster? Or single cluster?

(LEAP TO CHAPTER 4)

- In the event of a cluster failure, all workloads are affected in a single cluster environment.
- If a cluster fails, everything running on the cluster goes down with it... (eee eee eee!)
- Multi-clusters are more resilient, and increase the velocity and flexibility of individual dev teams - no need to coordinate with other teams!

Download the Full Gorilla Guide!

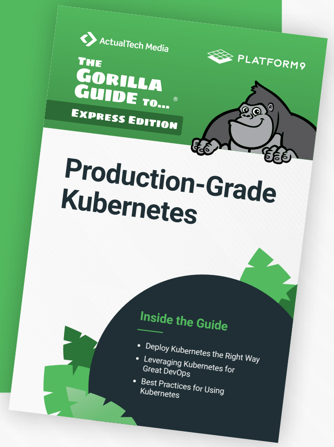
Kubernetes is the industry standard for orchestrating complex container environments. But it's got a serious learning curve.

This Guide provides information and practical advice for getting the most out of your Kubernetes setup, including best practices, and what you need to know to scale up, keep it secure, and more. We skip most of the theoretical aspects of Kubernetes in favor of how to use it day to day.

Highlights include:

- Deploying Kubernetes the right way
- Leveraging Kubernetes for great DevOps
- Best practices for using Kubernetes

GET YOUR COPY!



ActualTech Media PLATFORM9

THE GORILLA GUIDE TO...® EXPRESS EDITION

Production-Grade Kubernetes

Inside the Guide

- Deploy Kubernetes the Right Way
- Leveraging Kubernetes for Great DevOps
- Best Practices for Using Kubernetes

DEVELOPED BY ActualTech Media

www.gorilla.guide