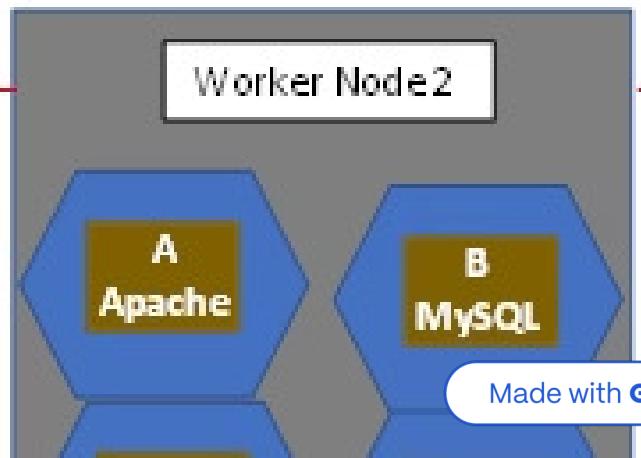
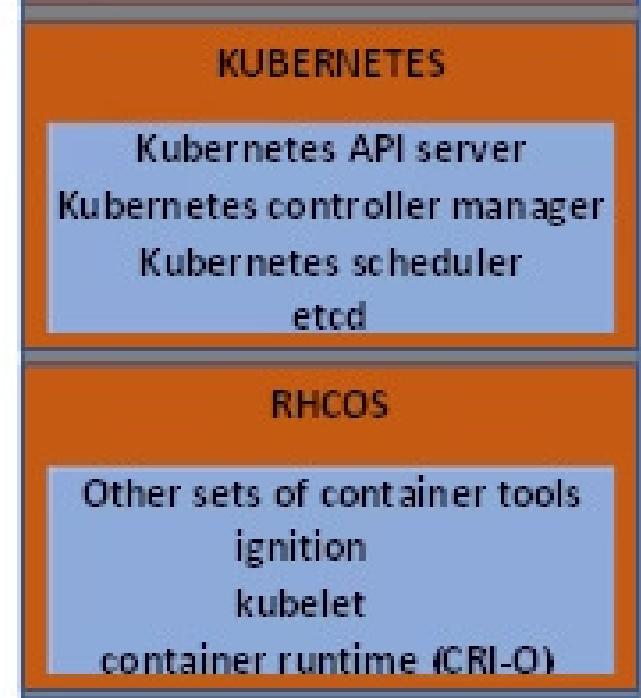


# OpenShift: Container Application Platform

Learn what OpenShift is, why it's important, and explore its comprehensive architecture built on Kubernetes and Red Hat Core OS.



# Why OpenShift Matters

## Container Runtime Limitations

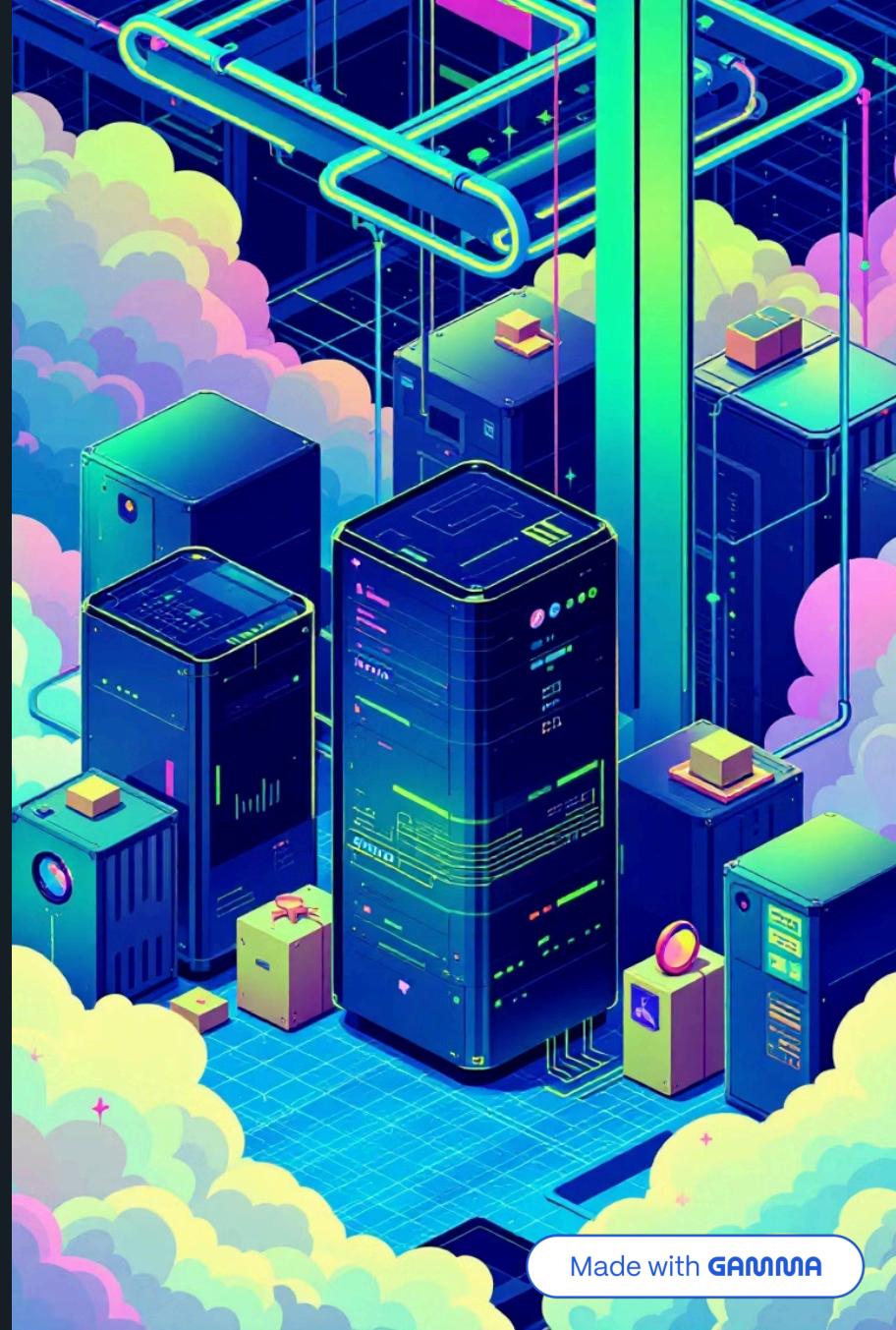
Docker and Podman work for basic containers but struggle with large-scale management.

## Enterprise Scale Challenge

Managing hundreds or thousands of containerized applications requires sophisticated orchestration.

## Mission-Critical Applications

Core banking and heavy enterprise applications demand high availability and automation.





# What is OpenShift?

## Container Application Platform

OpenShift orchestrates and manages containerized applications, enabling easy scaling as required.

Developed by Red Hat, it's built on top of Kubernetes cluster and Red Hat Core Operating System.

## Kubernetes Distribution

Compatible with Kubernetes but managed through OpenShift-specific tools.

Think of OpenShift as a distribution of Kubernetes with enterprise features.

# OpenShift Cluster Architecture

O1

## Master Nodes (Control Plane)

Provide basic services that manage the OpenShift cluster operations.

O2

## Worker Nodes (Compute)

Host your applications and workloads with more server resources.

O3

## Infrastructure Nodes

Host infrastructure services like monitoring and logging systems.





# Minimum Cluster Requirements

3

## Control Planes

Minimum required master nodes for cluster management

2

## Worker Nodes

Minimum compute nodes needed for application workloads

# Layer 1: Red Hat Core OS (RHCOS)



## Container-Optimized OS

Includes CRI-O container runtime (Kubernetes native) replacing Docker in newer versions.



## Kubelet Service

Interfaces with container runtime, starts pods, and assigns node resources.



## Ignition System

First boot system configuration to bring up and configure the cluster.

- RHCOS is immutable - no operating system management required, unlike RHEL.

## Layer 2: Kubernetes Services



### Kubernetes API

Gateway to cluster - validates credentials and handles authentication.

### Scheduler

Intelligently schedules pods on appropriate nodes automatically.

### etcd

Cluster brain - stores all operations as logs in YAML format.

### Controller Manager

Watches etcd, enforces desired state through API calls.

# Layer 3: OpenShift Services



## OpenShift API Server

Validates and configures OpenShift resources like templates, projects, and routes.



## OpenShift Controller Manager

Ensures OpenShift objects maintain desired state by watching etcd changes.



## OAuth Authentication

Integrates external authentication methods and manages user tokens for API access.

container  
image

# Three-Layer Architecture Summary



## RHCOS Foundation

Container-optimized OS with CRIOS runtime, kubelet service, and ignition configuration.



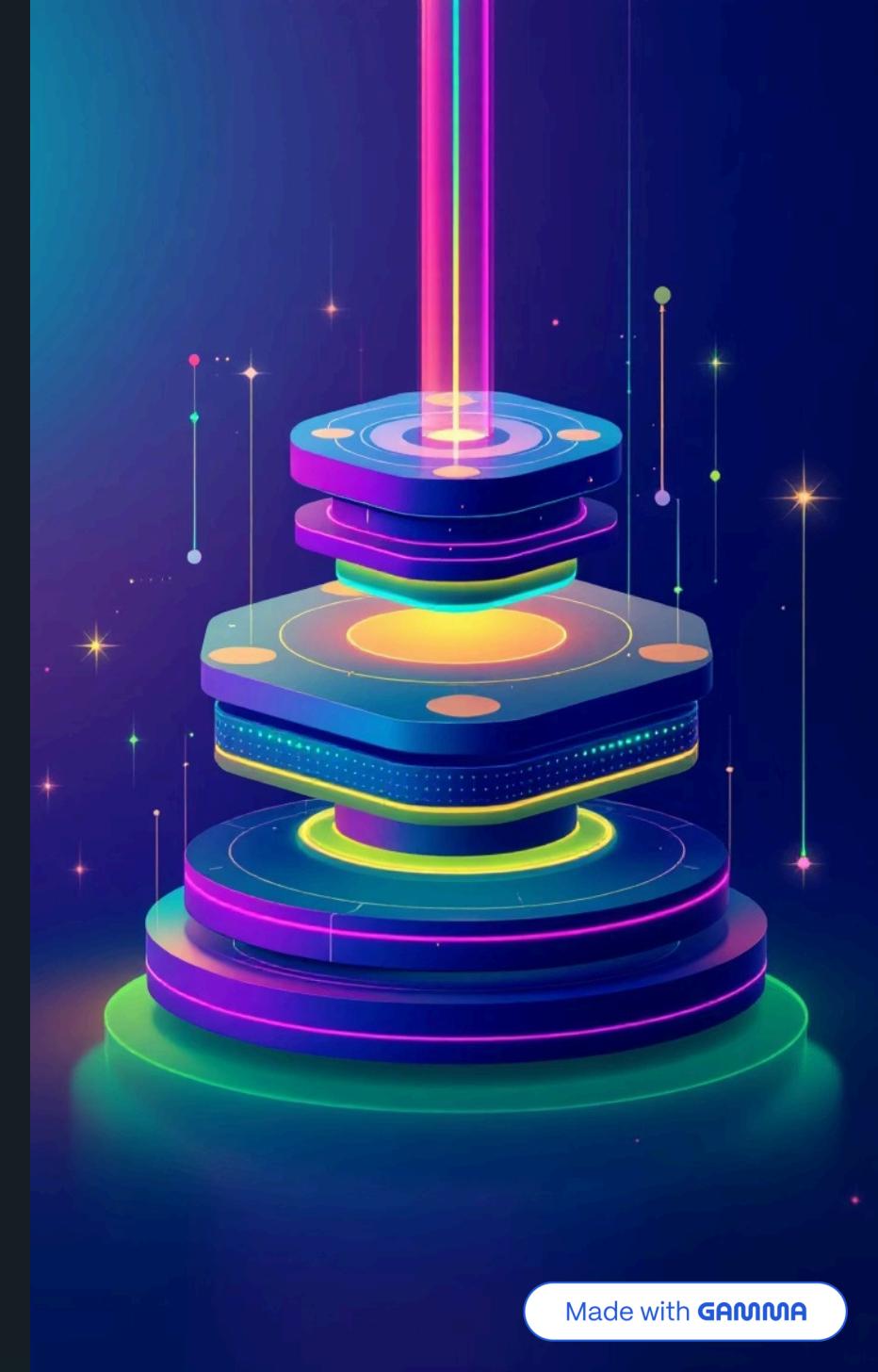
## Kubernetes Layer

Core orchestration with API server, scheduler, etcd, and controller manager services.



## OpenShift Layer

Enterprise features with OpenShift API, controller manager, and OAuth authentication.



# Ready to Deploy OpenShift

# Next Steps"»

Now that you understand OpenShift architecture, you're ready to learn cluster installation and deployment strategies.

The next lesson covers hands-on OpenShift cluster setup and configuration.

[EX280 Pra...](#)

[Watch V..](#)

