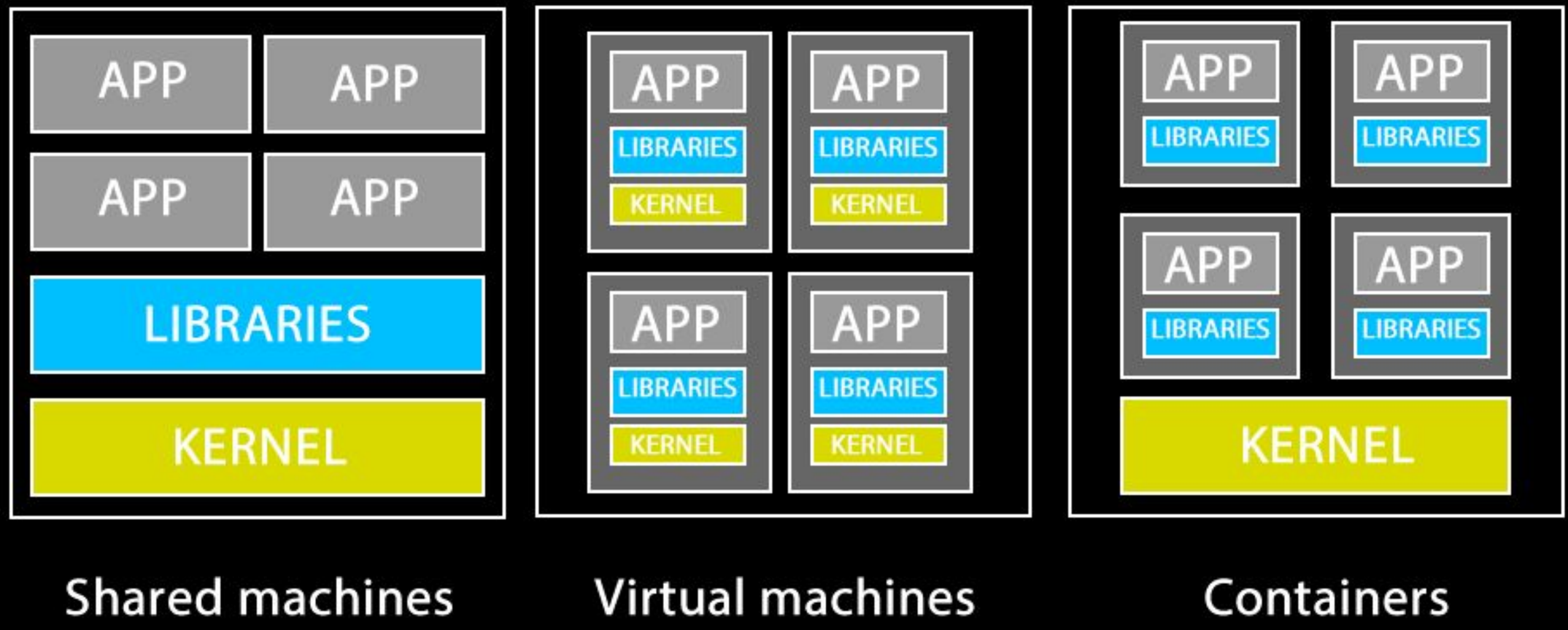
GOOGLE CLOUD PLATFORM

KUBERNETES ENGINE

CONTAINERS

Levels of isolation for apps, libraries and resources

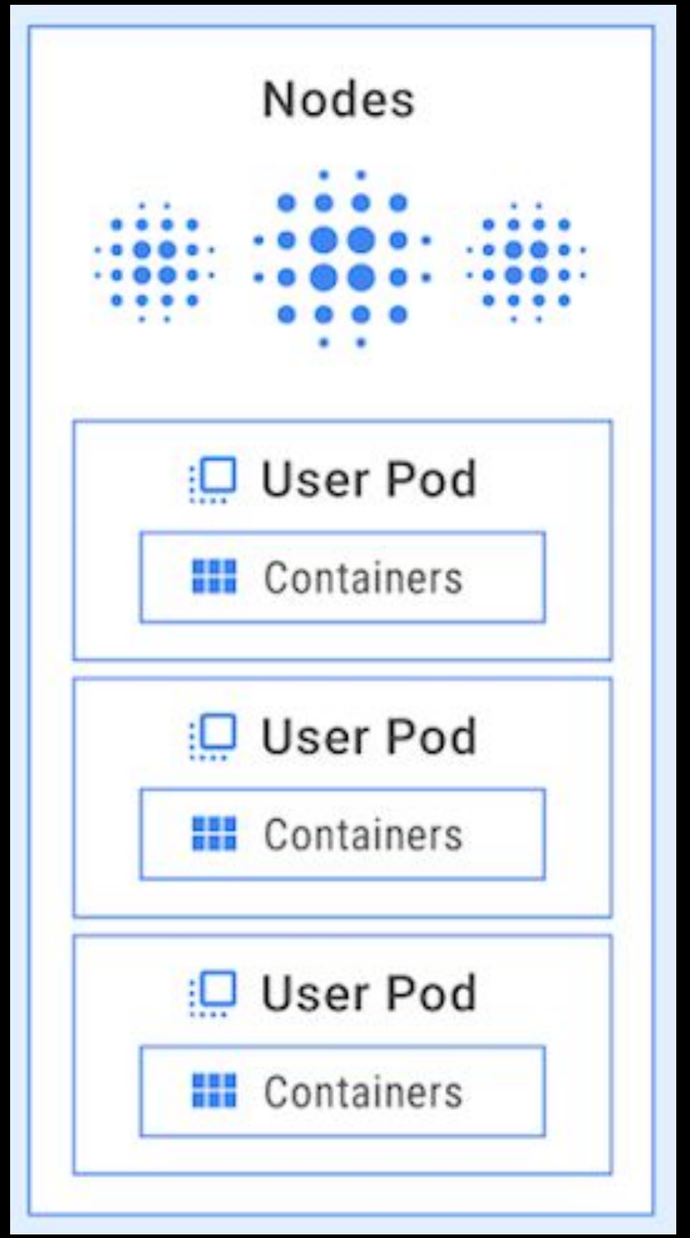


CONTAINERS

- Not a new concept. Have been around for a long time
- Their public adoption took off with Docker
- Docker offered a light weight container runtime with easy packaging and deployment
- Allowed you to bundle your app, all it’s dependencies and deploy that consistently across different environments
- Docker was focused on providing an amazing experience for a single node, single container service

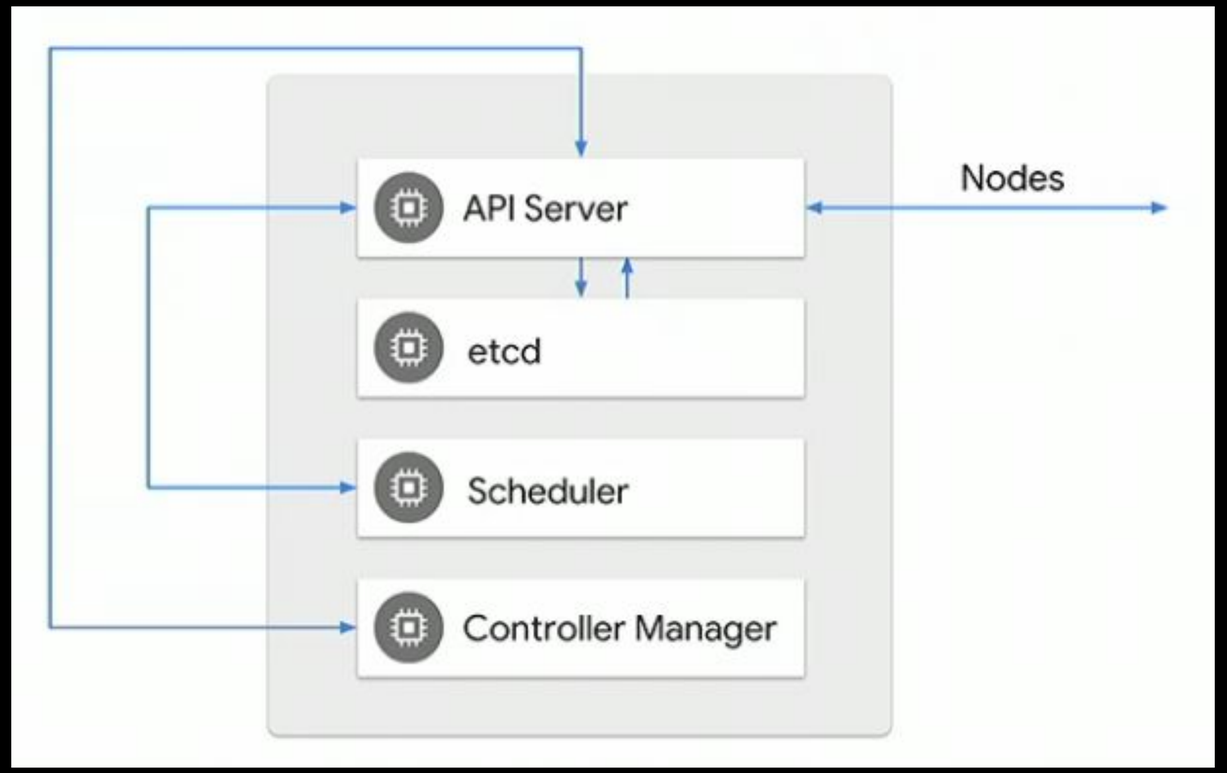
KUBERNETES

- Container orchestration tool
- Manage multiple containers across many different nodes, pods
- Handles scheduling, scaling, health checks, etc.
- ‘Abstraction over infrastructure’
- Consistent base layer - infrastructure operators vs. developers
- Powerful and complex

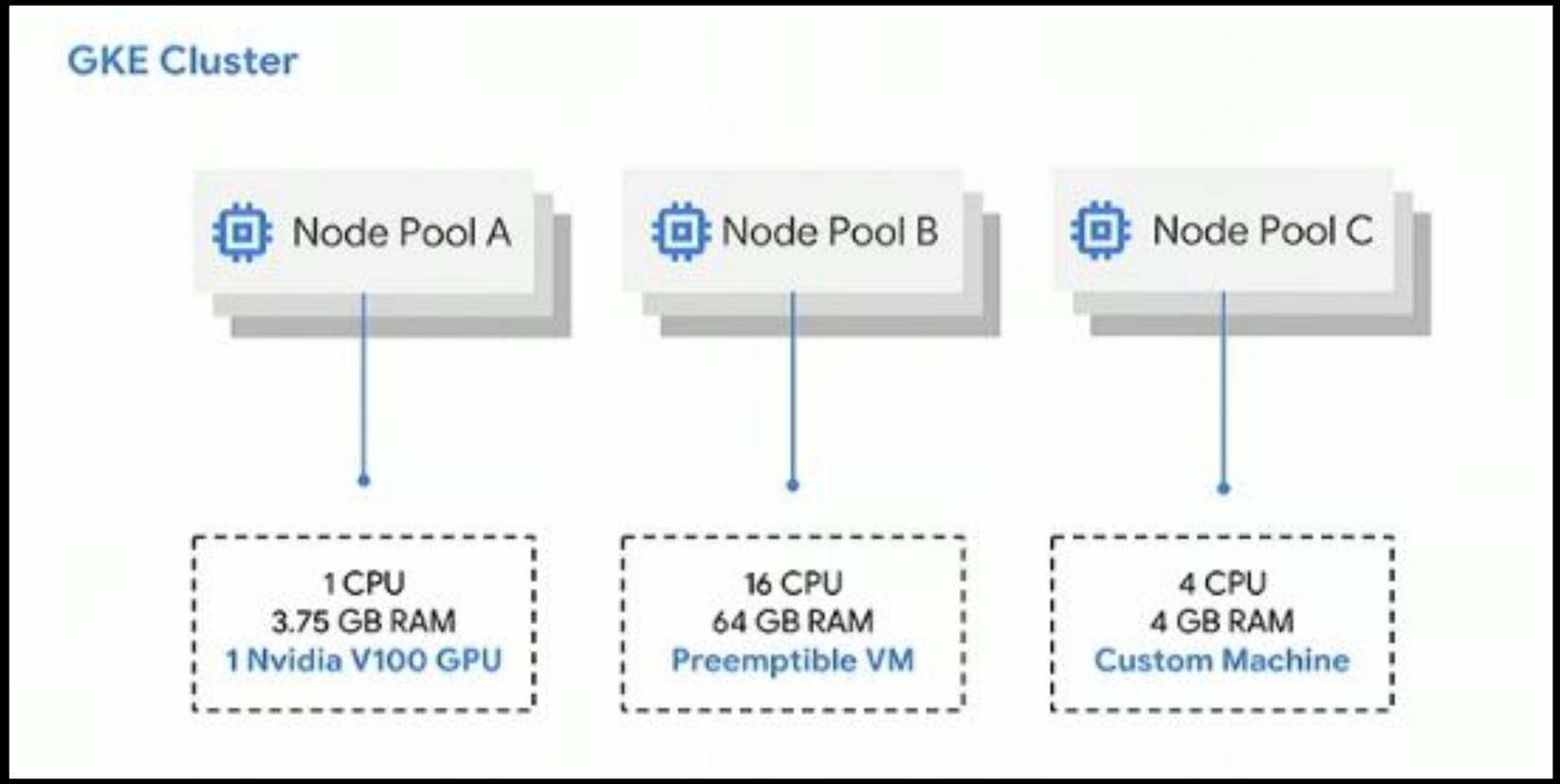


GOOGLE KUBERNETES ENGINE (GKE)

- Fully managed Kubernetes solution
  - Control plane & Nodes
- Runs on Google Cloud Platform
- Fully managed node experience
  - Provides a base image
  - Provides updates to the base image
  - Node auto repair
  - Node auto upgrade
- Node pools – group of related machines with different resources

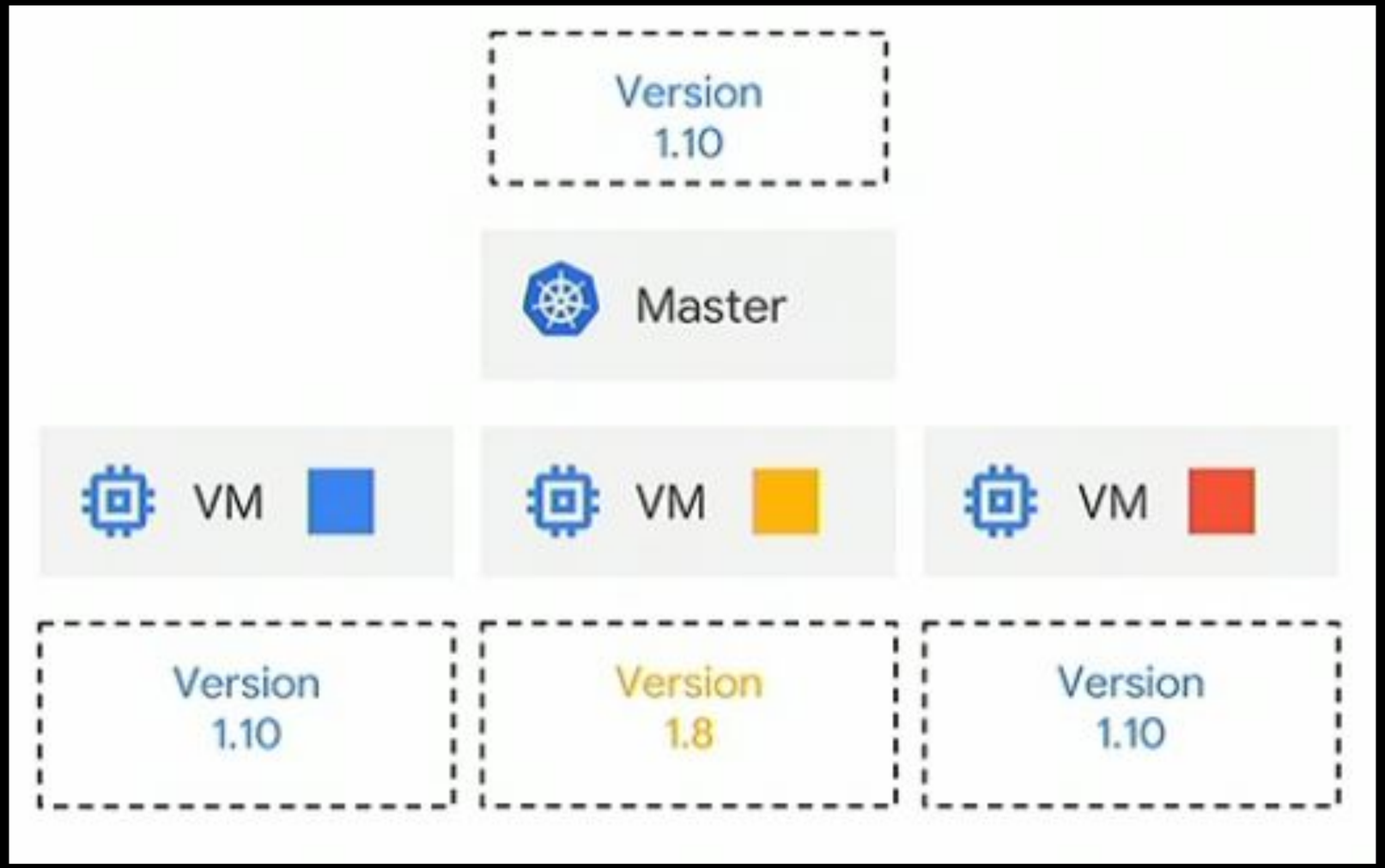


GKE NODE POOLS



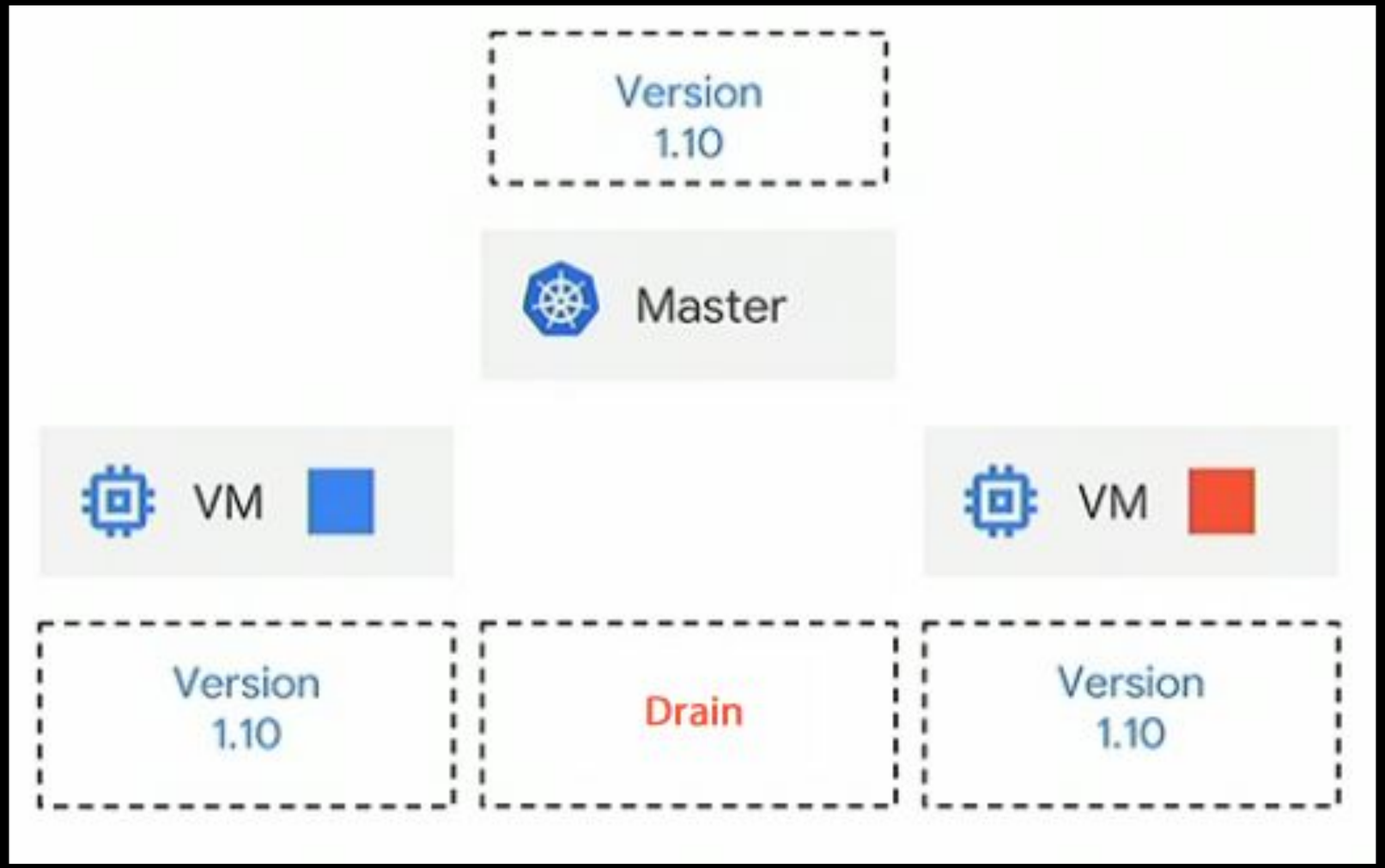
- Supports multiple node pools with heterogeneous resources
- Create Preemptible VM's
- GPU's and Local SSD's
- Custom Machine Types

GKE NODE AUTO UPGRADE



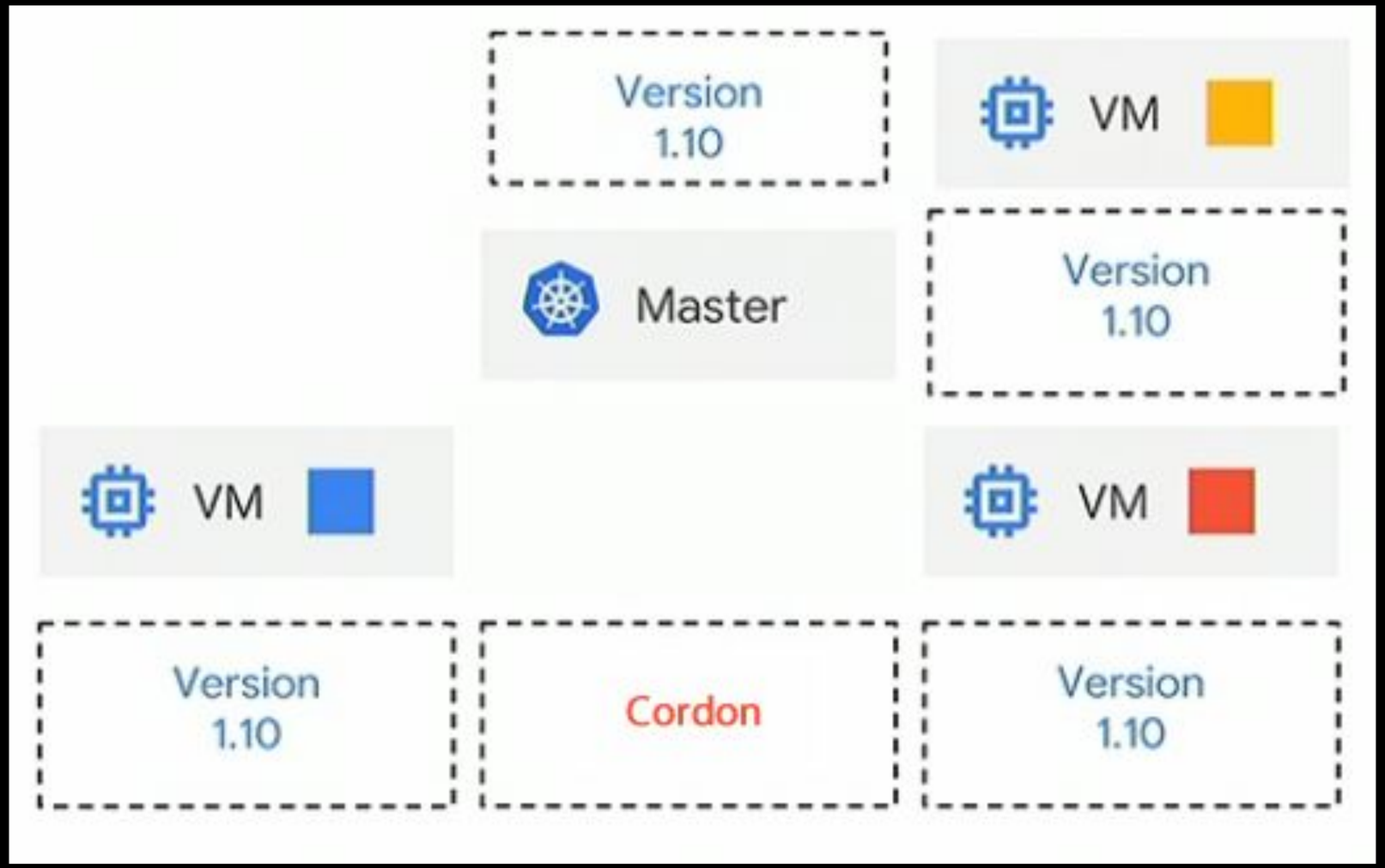
- Automatically keeps nodes up to date with the latest stable version of Kubernetes
- Applies security updates at OS and K8's level
- Maintenance windows allow for control over when upgrades should occur

GKE NODE AUTO UPGRADE



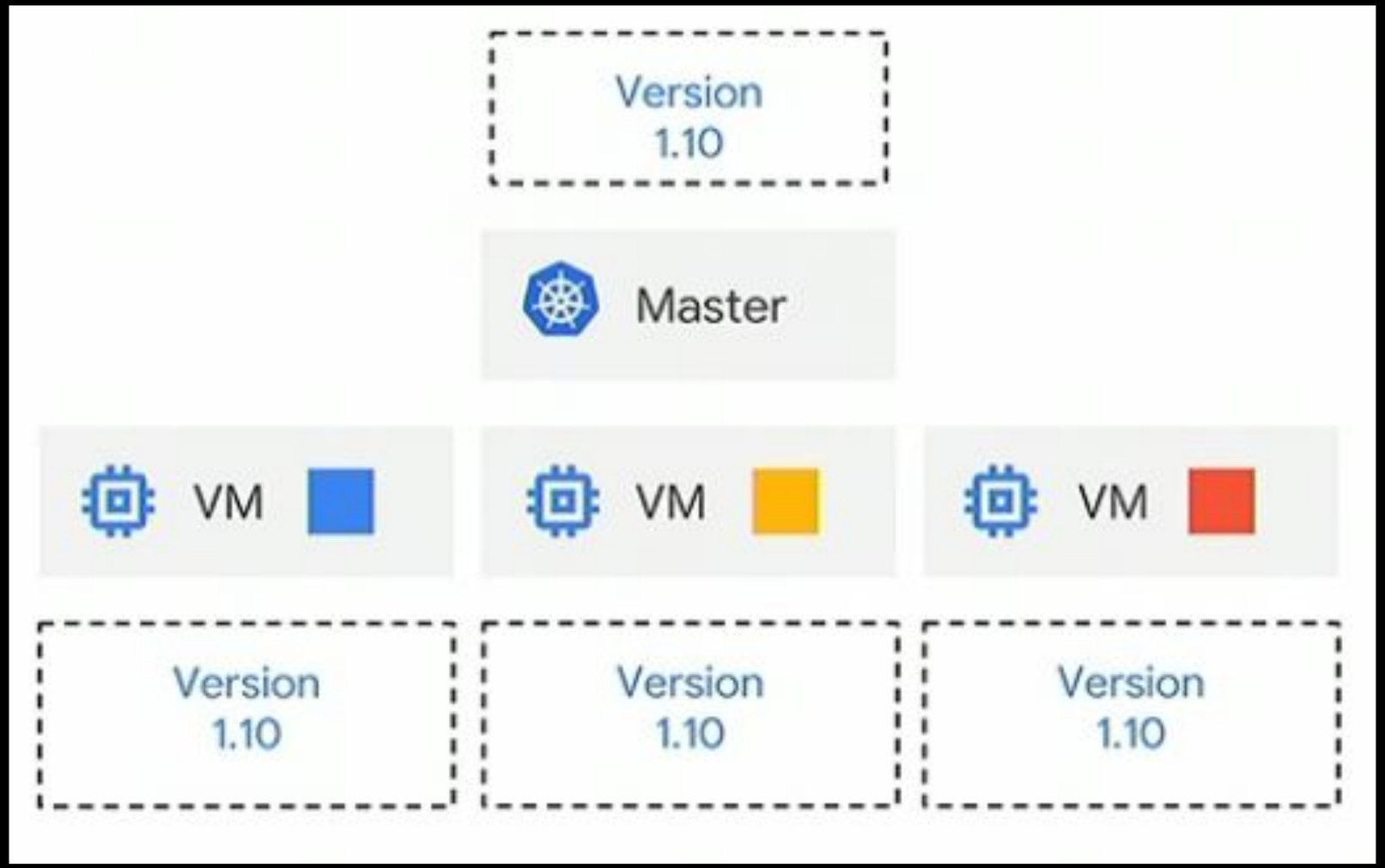
- Automatically keeps nodes up to date with the latest stable version of Kubernetes
- Applies security updates at OS and K8's level
- Maintenance windows allow for control over when upgrades should occur

GKE NODE AUTO UPGRADE



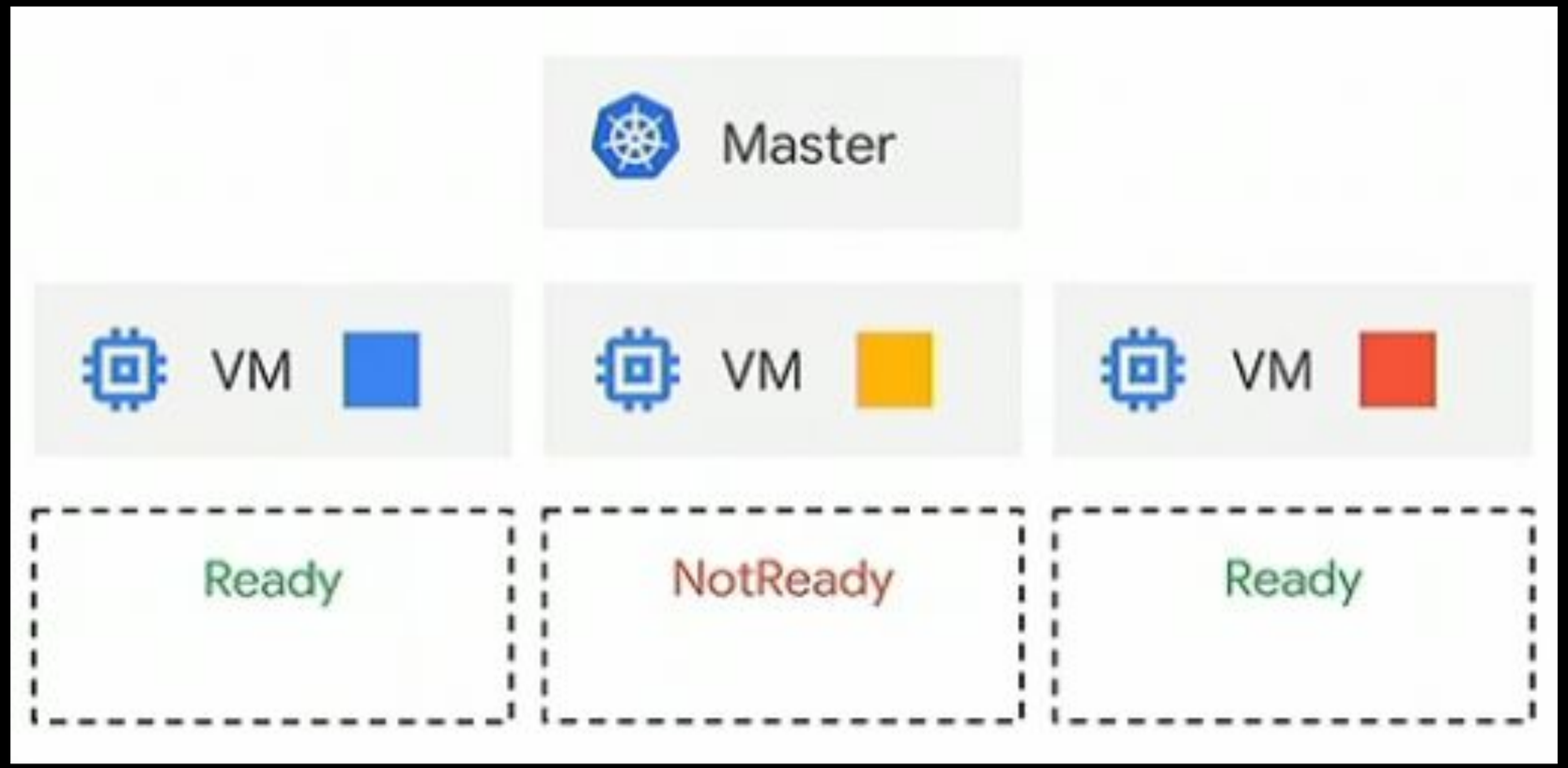
- Automatically keeps nodes up to date with the latest stable version of Kubernetes
- Applies security updates at OS and K8's level
- Maintenance windows allow for control over when upgrades should occur

GKE NODE AUTO UPGRADE



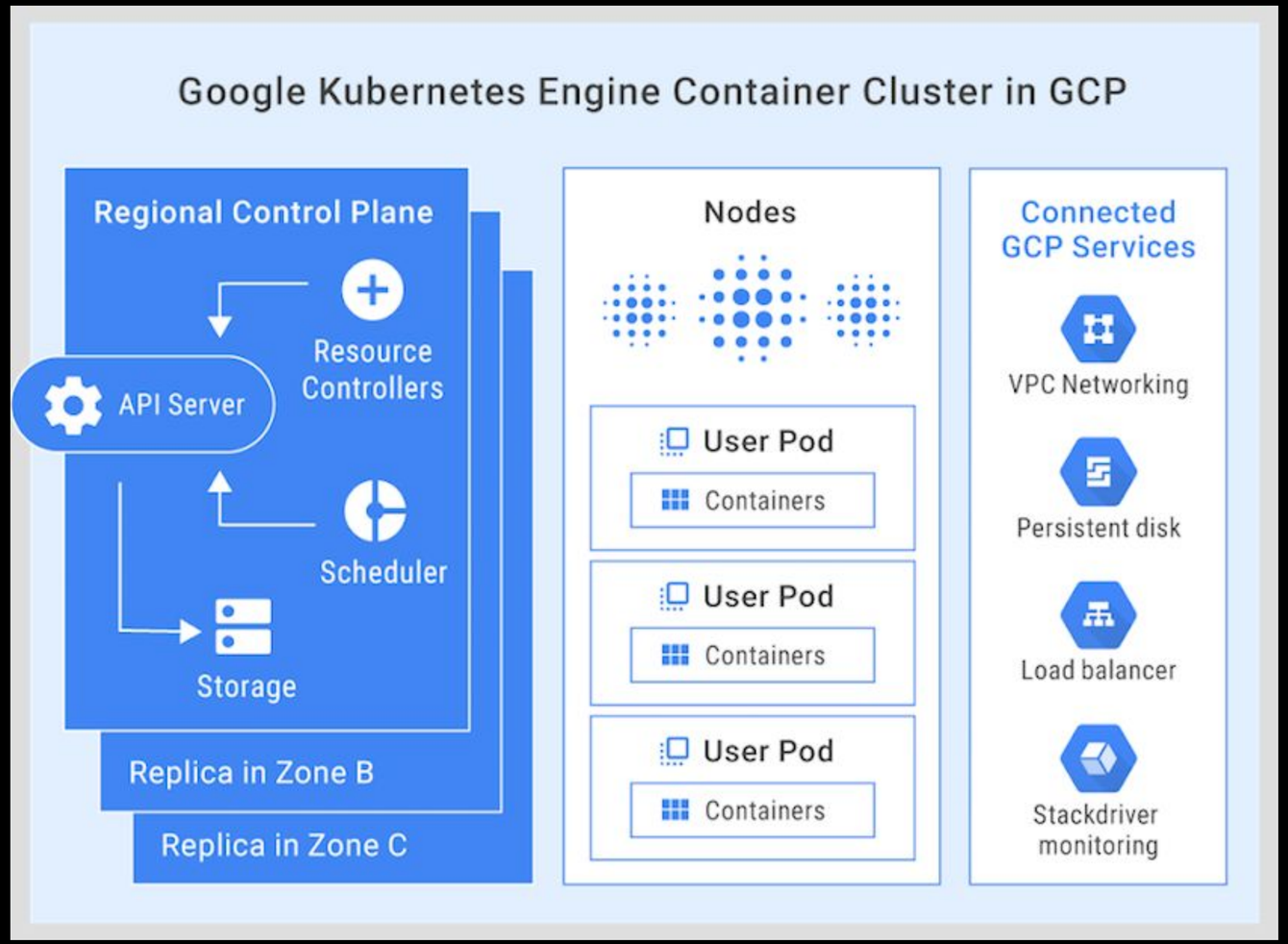
- Automatically keeps nodes up to date with the latest stable version of Kubernetes
- Applies security updates at OS and K8's level
- Maintenance windows allow for control over when upgrades should occur

GKE NODE AUTO REPAIR



- Keeps nodes in a healthy, running state
- Checks for unreachable nodes
- Checks for nodes reporting a NotReady status and low disk space
- Unhealthy nodes have their pods drained, and the node is recreated

HIGH AVAILABILITY & SERVICES



- Replication to three regions
- Zero downtime uppgrades
- Uptime from 99.5 to 99.95
- Easily connect to GCP services

GKE – GUI BASICS

- **Clusters** - A cluster consists of at least one *cluster master** and multiple worker machines called nodes. These master and node machines run the Kubernetes cluster orchestration system.

- **Workloads** - Containers, whether for applications or batch jobs, are collectively called workloads. Before you deploy a workload on a GKE cluster, you must first package the workload into a container.

- **Services** - A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them - sometimes called a micro-service

- **Applications** - Kubernetes Applications collect containers, services and configuration that are managed together. GKE allows for 1-step deploy as well as from the full gcloud command line.

* *The cluster master runs the Kubernetes control plane processes*

WORDPRESS DEPLOYMENT

- Wordpress is an open-source and free Web publishing application, content management system (CMS) and blogging tool built by a community of developers and contributors.
- A typical Wordpress installation includes a filesystem and a MySQL database.
- On GKE, two workloads, two pods and two services are created for these two components; and a loadbalancer to connect to the outside world.

WORDPRESS DEPLOYMENT IN GKE

1. Check billing information
2. Create/select projects
3. Check permissions
4. Create/select Cluster
5. Deploy Application
6. Check Workloads/Services
7. Expose pod w/ loadbalancer from Workloads
8. Check new Service
9. Find the external IP address
10. Go to the URL and complete the setup