Using Container-native Load Balancing on Kubernetes

Janani Ravi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Introducing Container-native load balancing

Creating VPC-native cluster

Instantiating deployment, service and ingress

Verifying ingress

A Quick Overview of Kubernetes

Kubernetes

Orchestration technology for containers - convert isolated containers running on different hardware into a cluster

Compute Choices

Containers

hybrid, multi-cloud

Container Clusters

Kubernetes

Kubernetes as Orchestrator

Fault-tolerance

Autohealing

Isolation

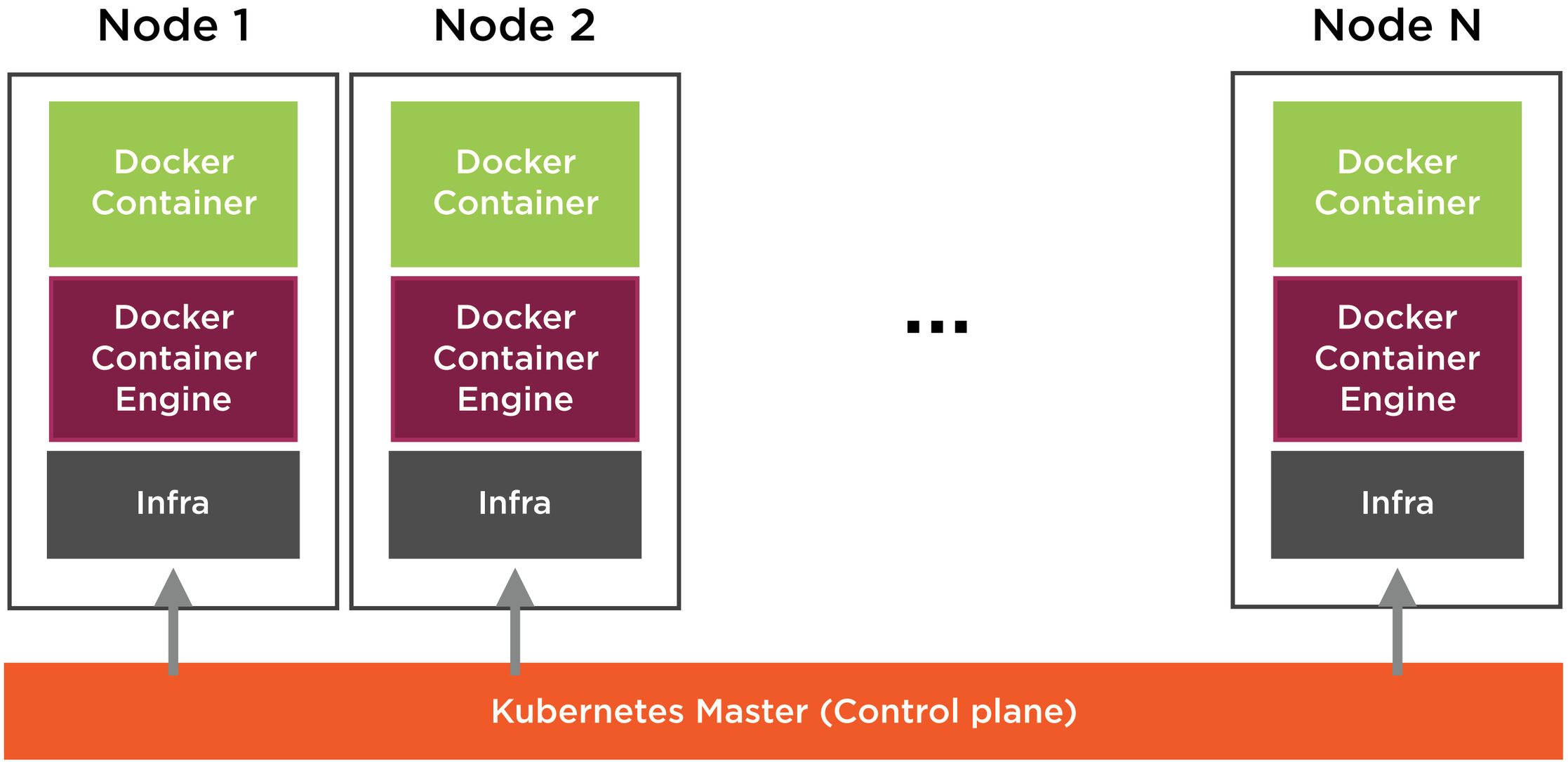
Scaling

Autoscaling

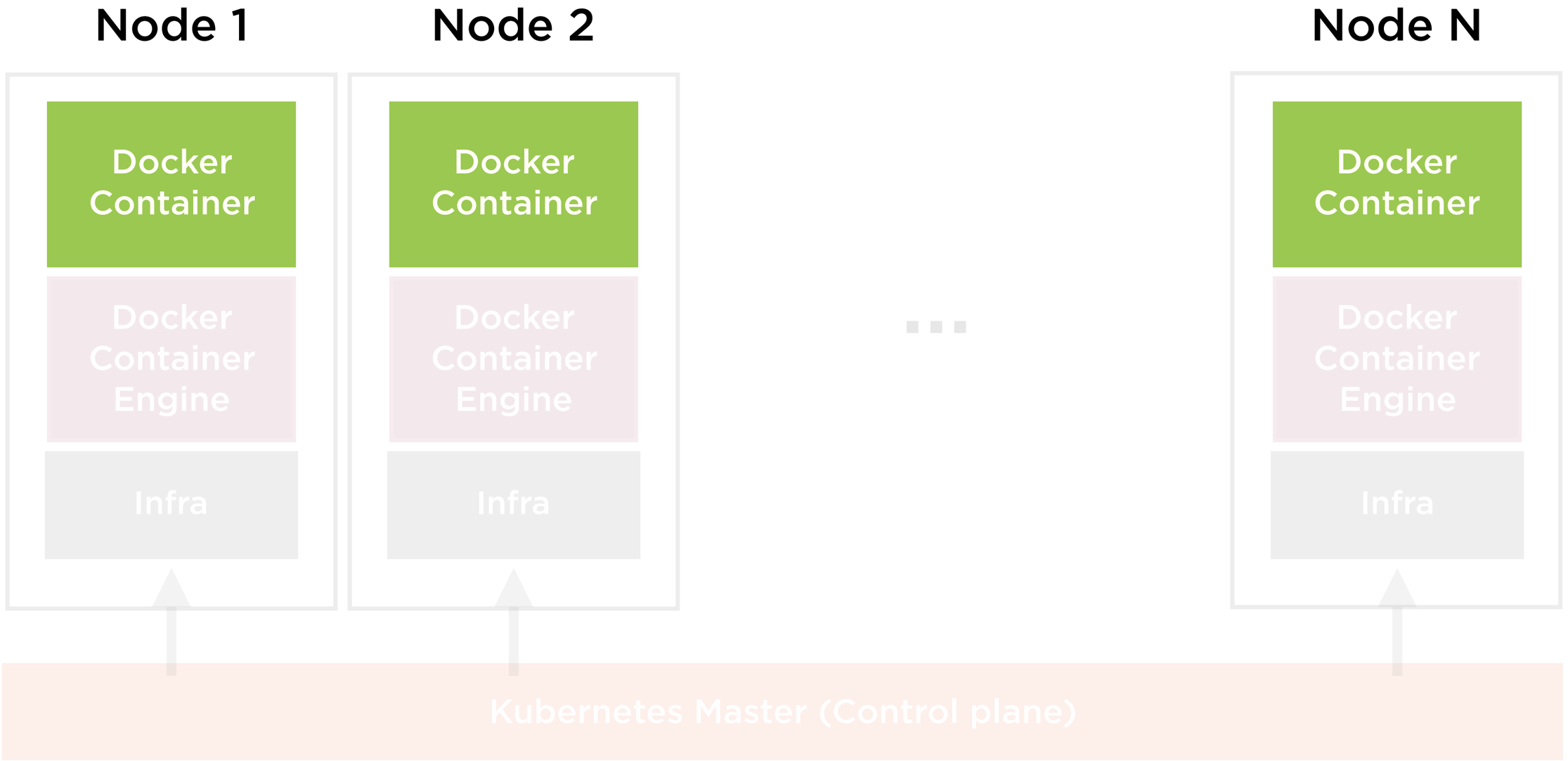
Load balancing

All of these are possible in a Kubernetes cluster using higher level abstractions

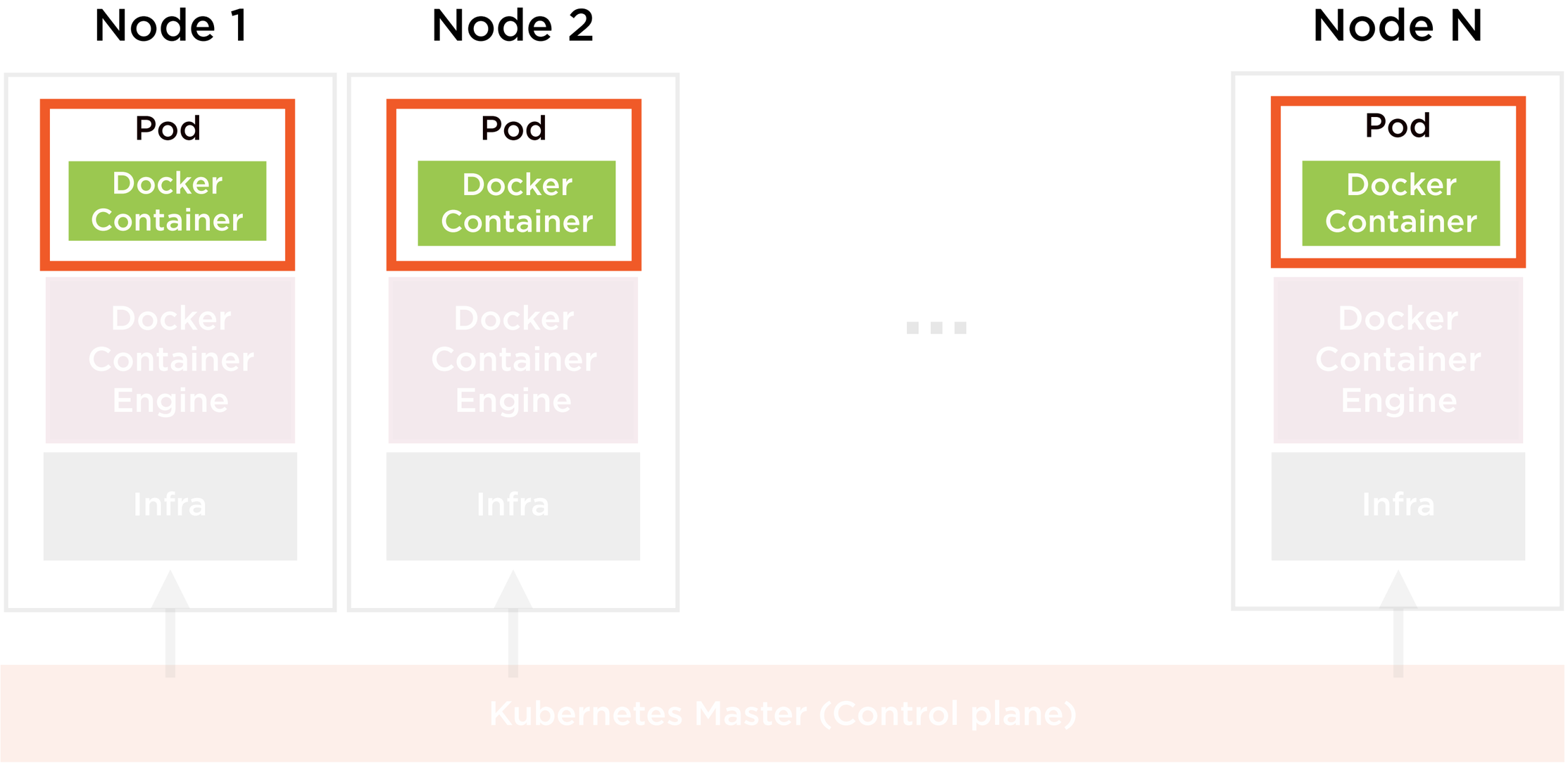
Kubernetes: Cluster Orchestration



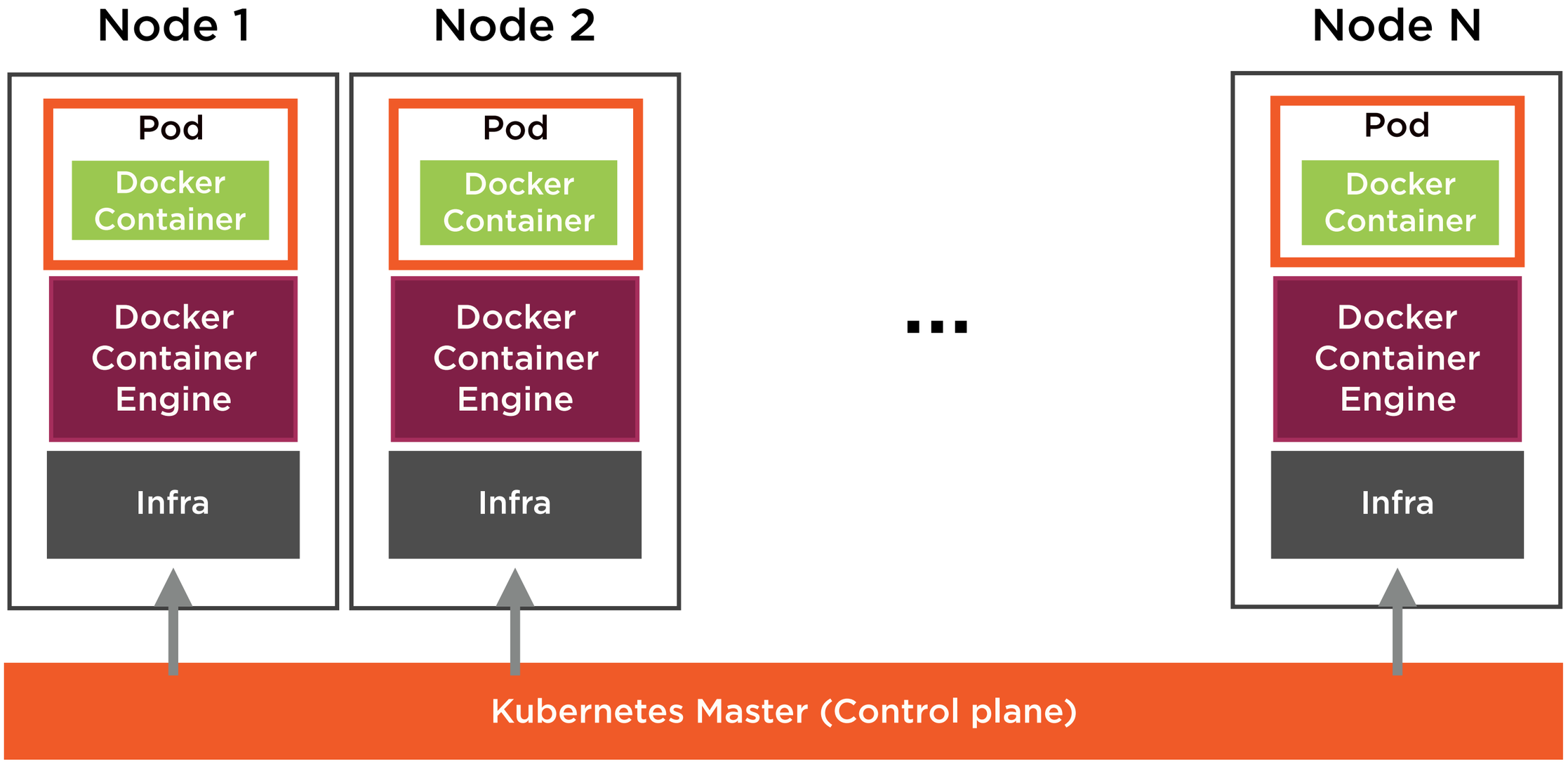
Kubernetes: Cluster Orchestration



Kubernetes: Containers Run Within Pods



Kubernetes: Cluster Orchestration



Pods as Atomic Units

Container deployment

All containers in pod are deployed, or none are

Node association

Entire pod is hosted on the same node

Pod is atomic unit of deployment in Kubernetes

The ReplicaSet Object

Multiple identical pods which are replicas of each other

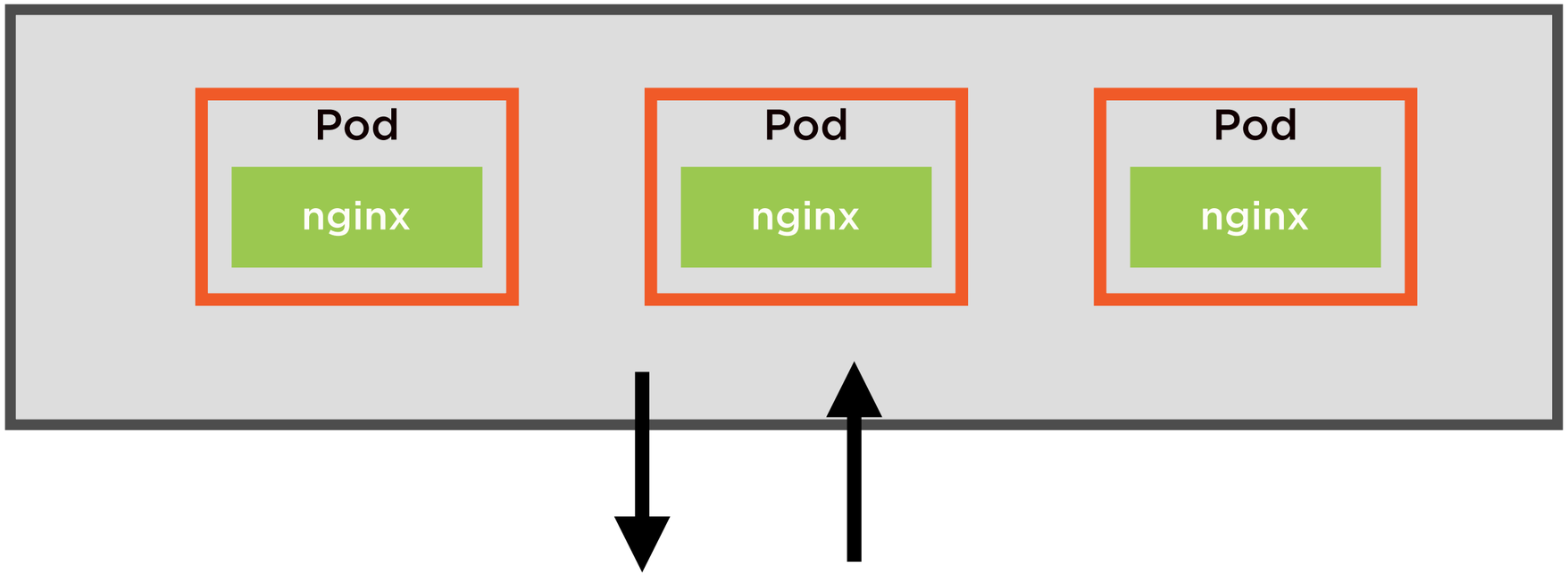
ReplicaSet

Self healing and autoscaling for our pods

The Deployment Object

Adds on deployment and rollback functionality

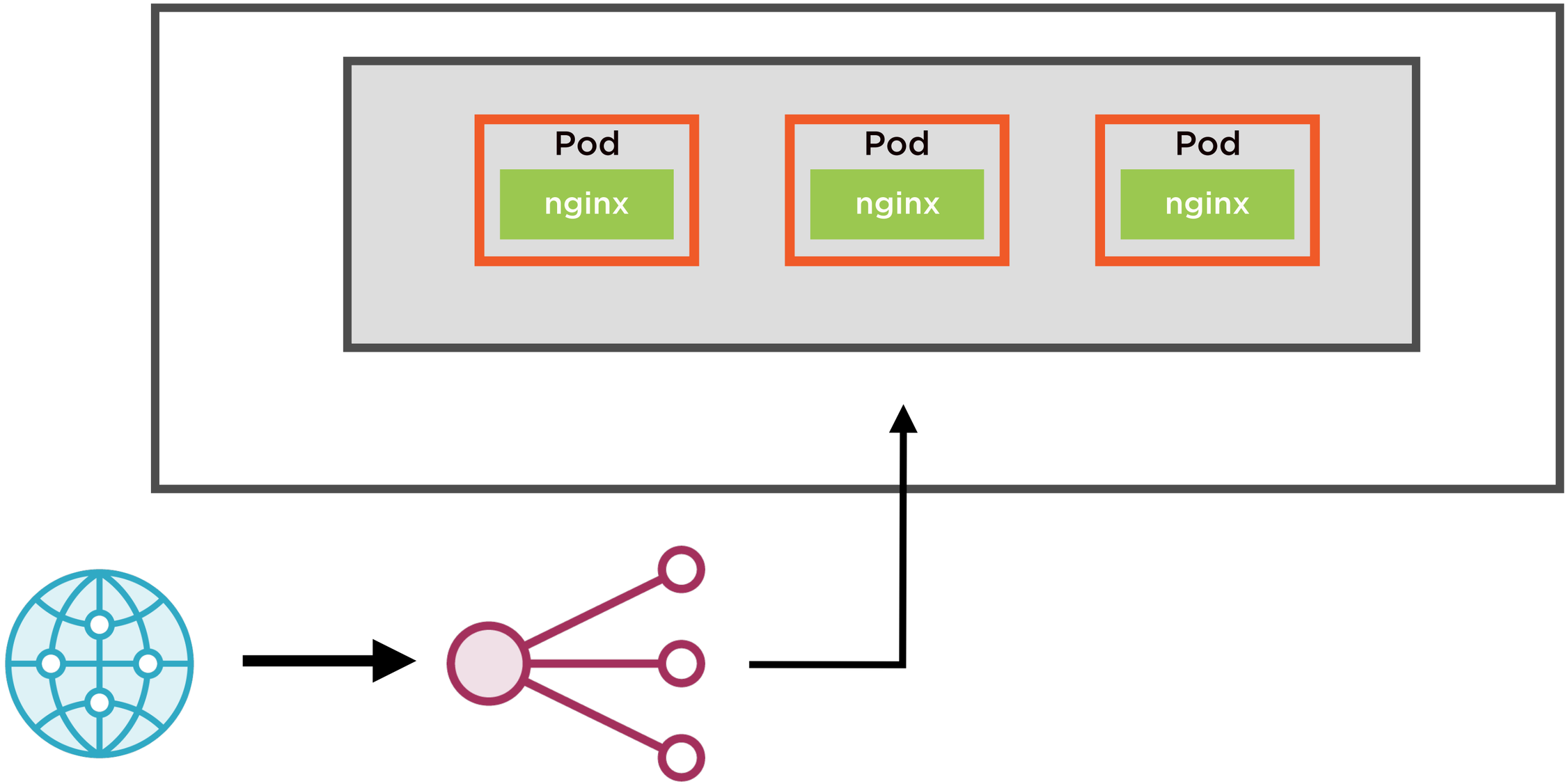
Deployment



Support for versions, and production-level operations such as rollbacks

Services provide stable IP addresses for external connections and load balancing

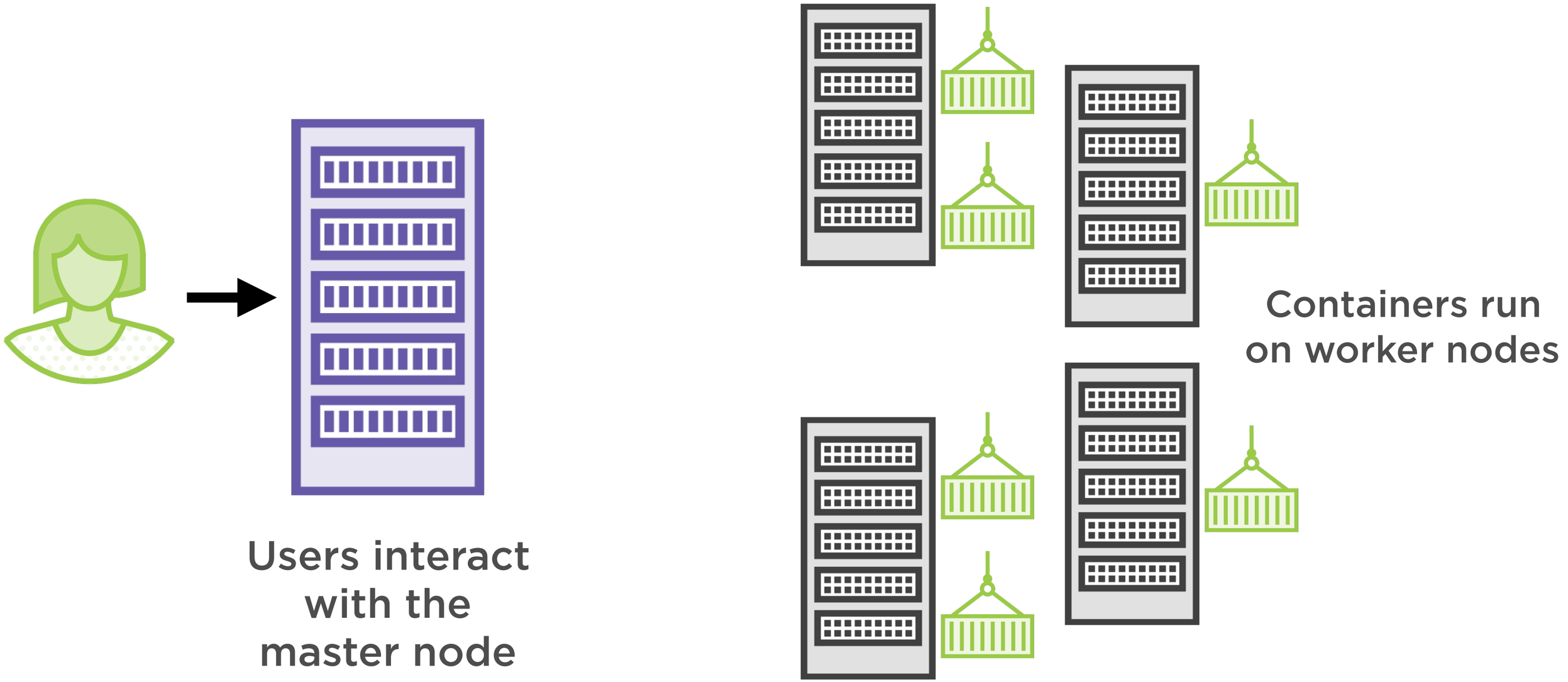
Service



A GKE Cluster

Made up of nodes, arranged in node pools, running container optimized node images

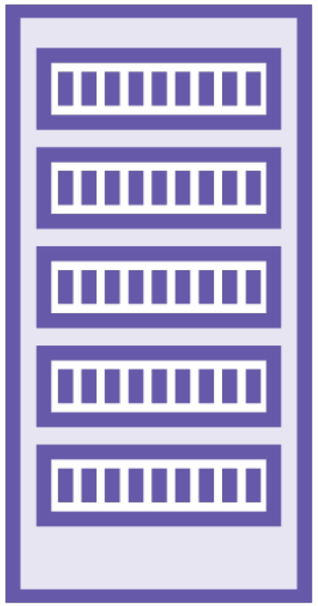
Kubernetes Clusters

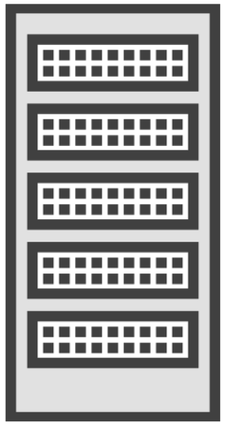


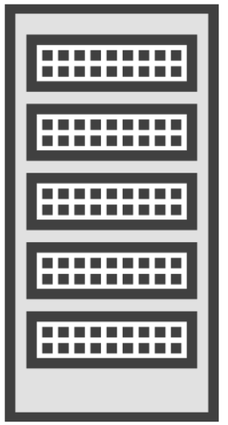
Users interact with the master node

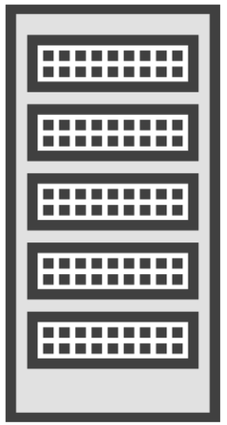
Containers run on worker nodes

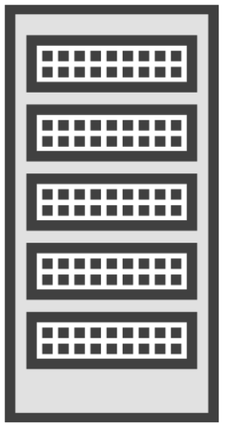
Nodes





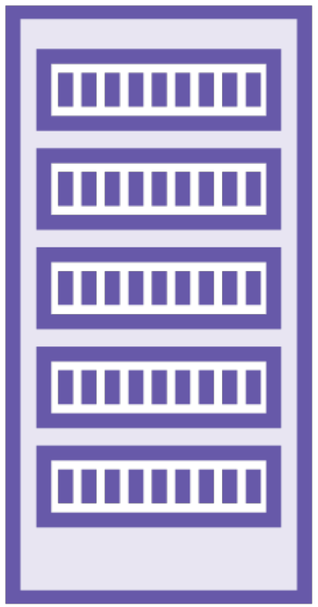




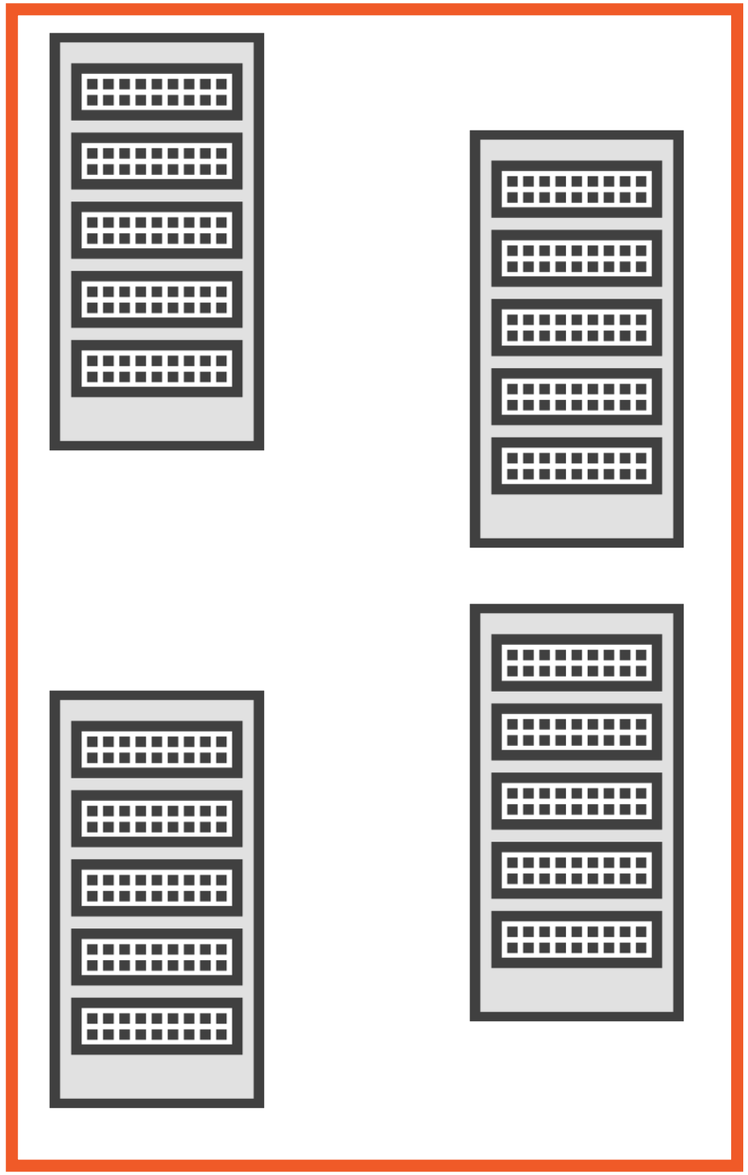


Nodes are on-premise or cloud VMs
on which containers are run

Node Pools



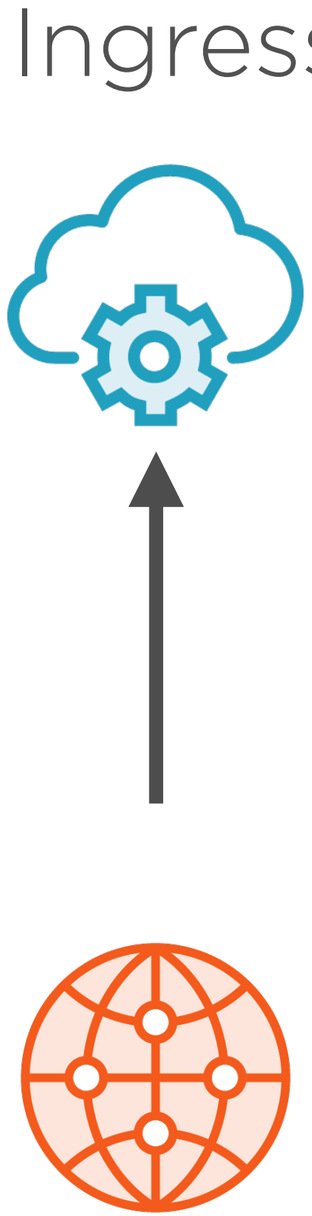
A subset of node instances which have the same configuration are called node pools



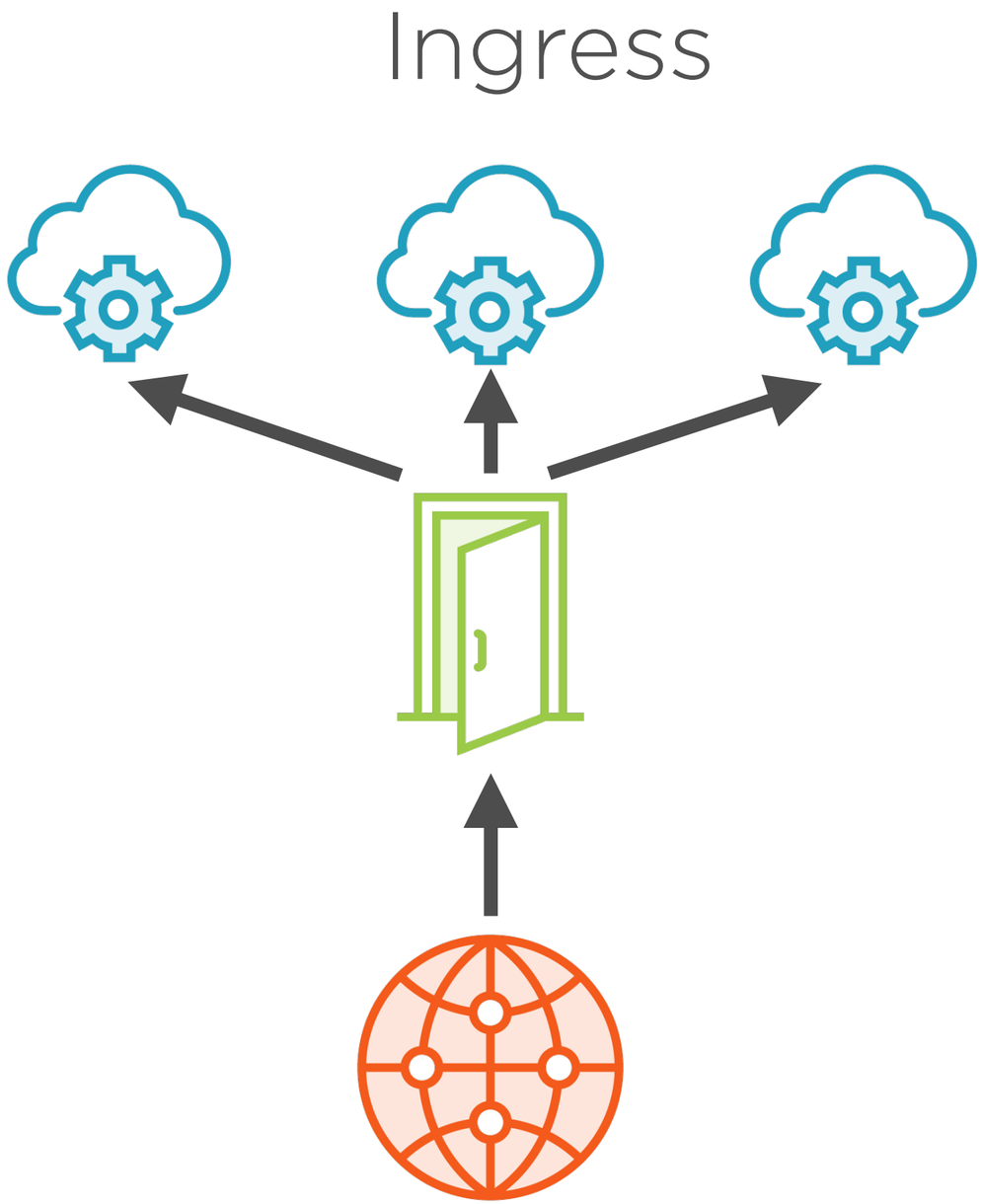
Ingress Object

Kubernetes object defining a collection of rules that allow inbound connections to reach cluster services. On GKE, a single ingress object can control access to multiple services

A single service can expose an IP address for access



With multiple services it makes sense to have rules defined using an ingress object



Container-native Load Balancing

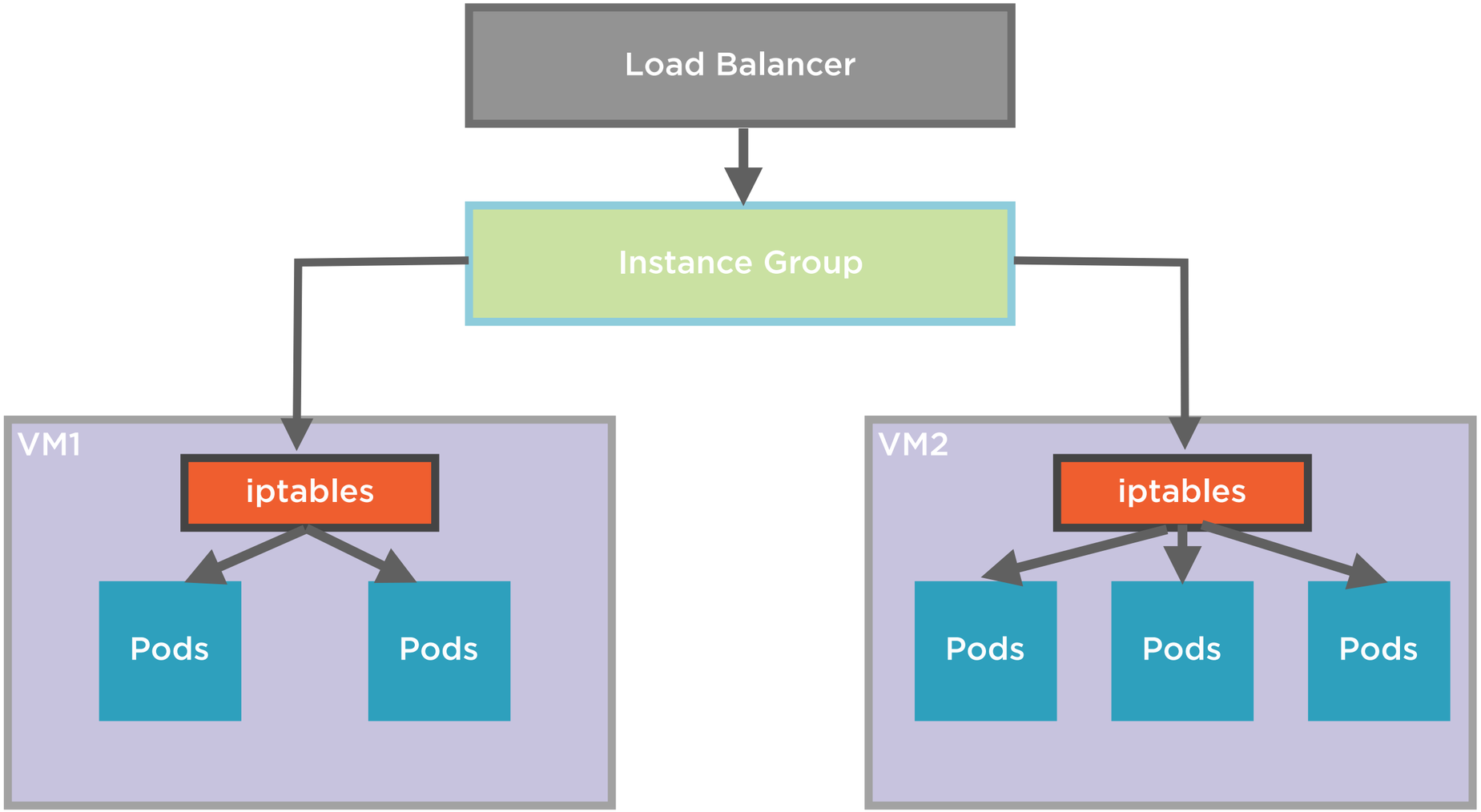
Container-native Load Balancing

Allows Google Cloud Load Balancers to target Kubernetes Pods directly to evenly distribute traffic between Pods

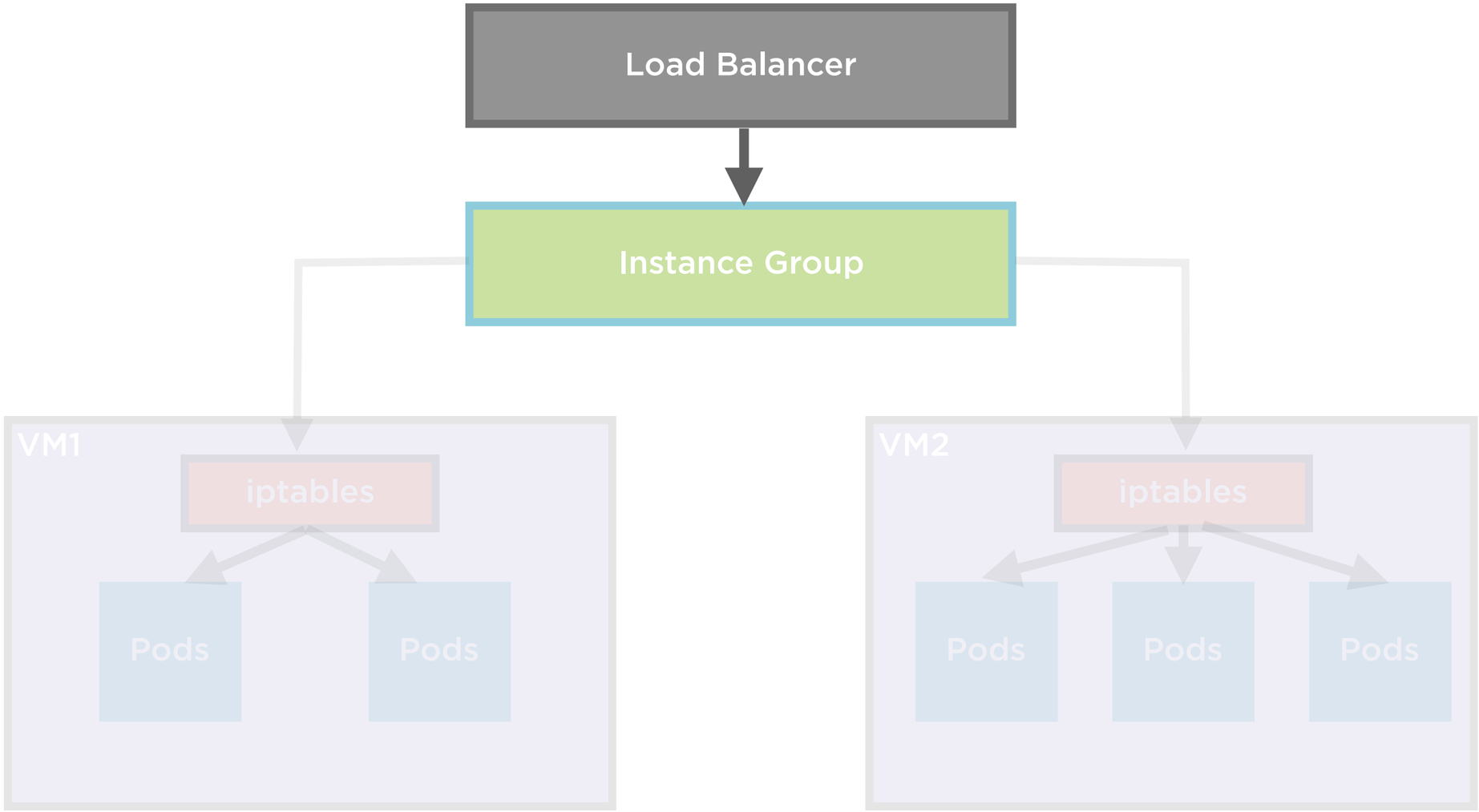
Container-native Load Balancing

Allows Google Cloud Load Balancers to target Kubernetes Pods directly to evenly distribute traffic between Pods

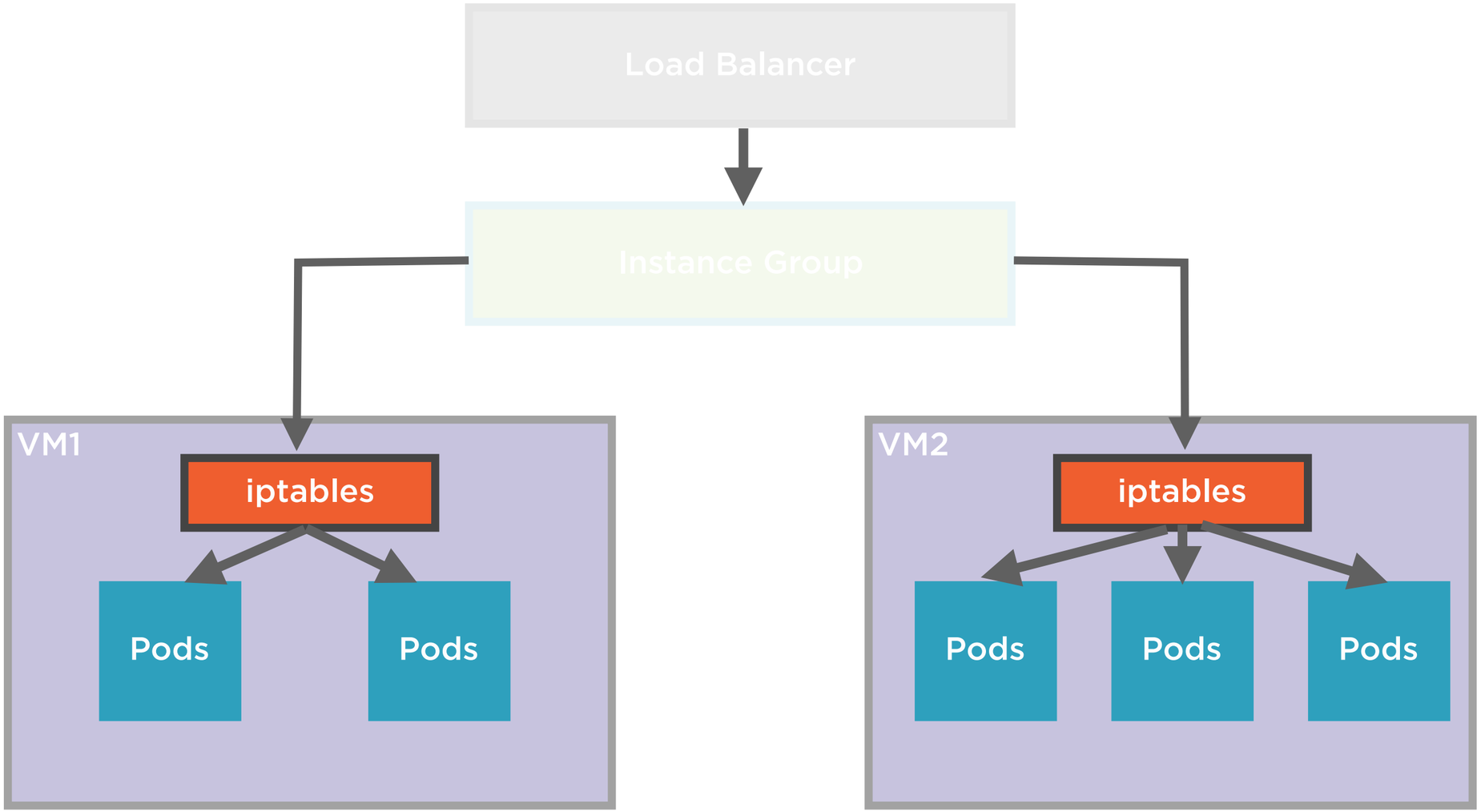
Default Load Balancing Behavior



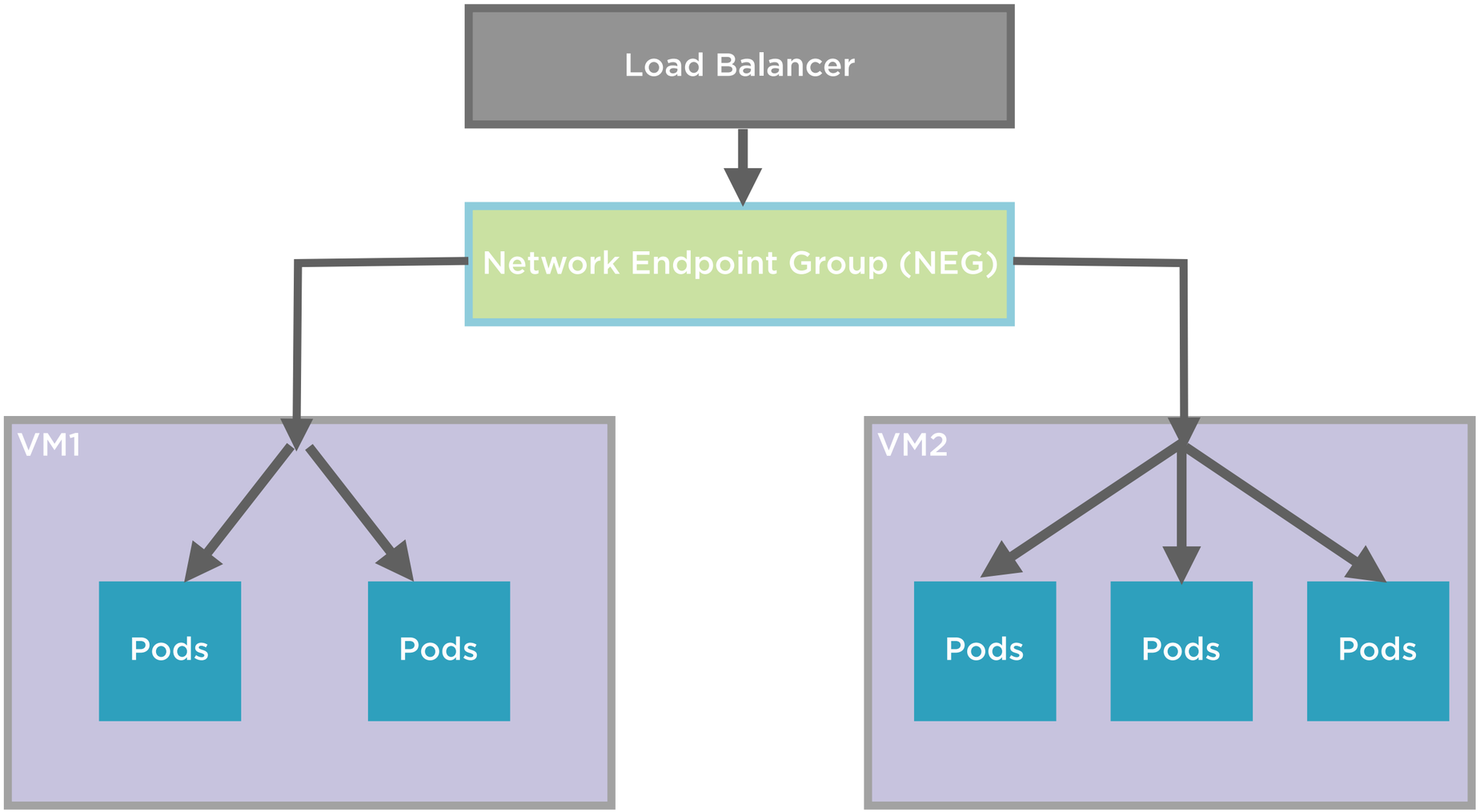
Load Balancer Sends Traffic to Instance Group



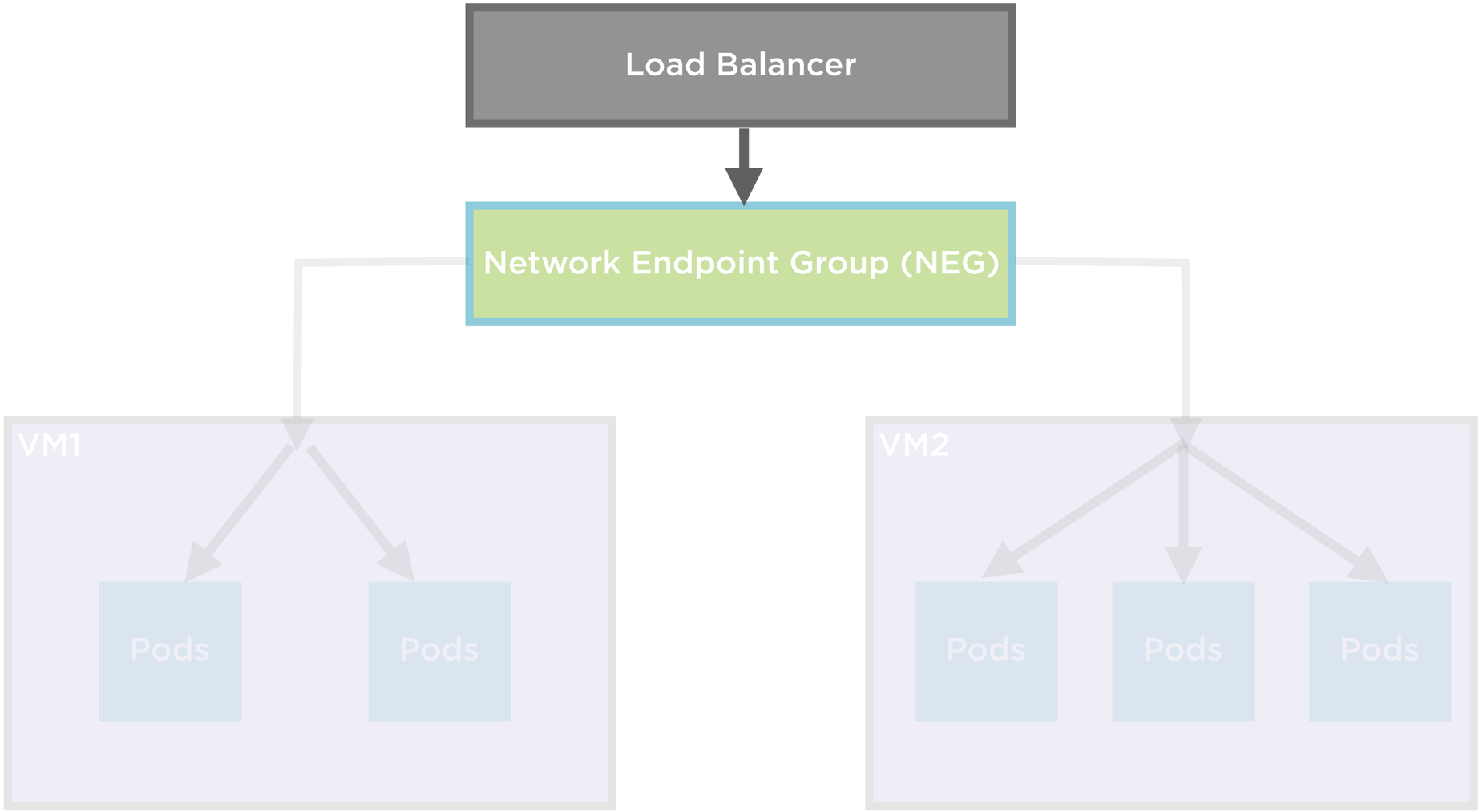
Kubernetes Uses iptables for Routing



Container-native Load Balancing



Uses Network Endpoint Groups (NEGs)



Collections of network endpoints represented by IP-port pairs

Network Endpoint Groups

Zonal resources which represent a collection of IP address and port combinations. Can be used as a backend for HTTP(S), TCP proxy and SSL proxy load balancing

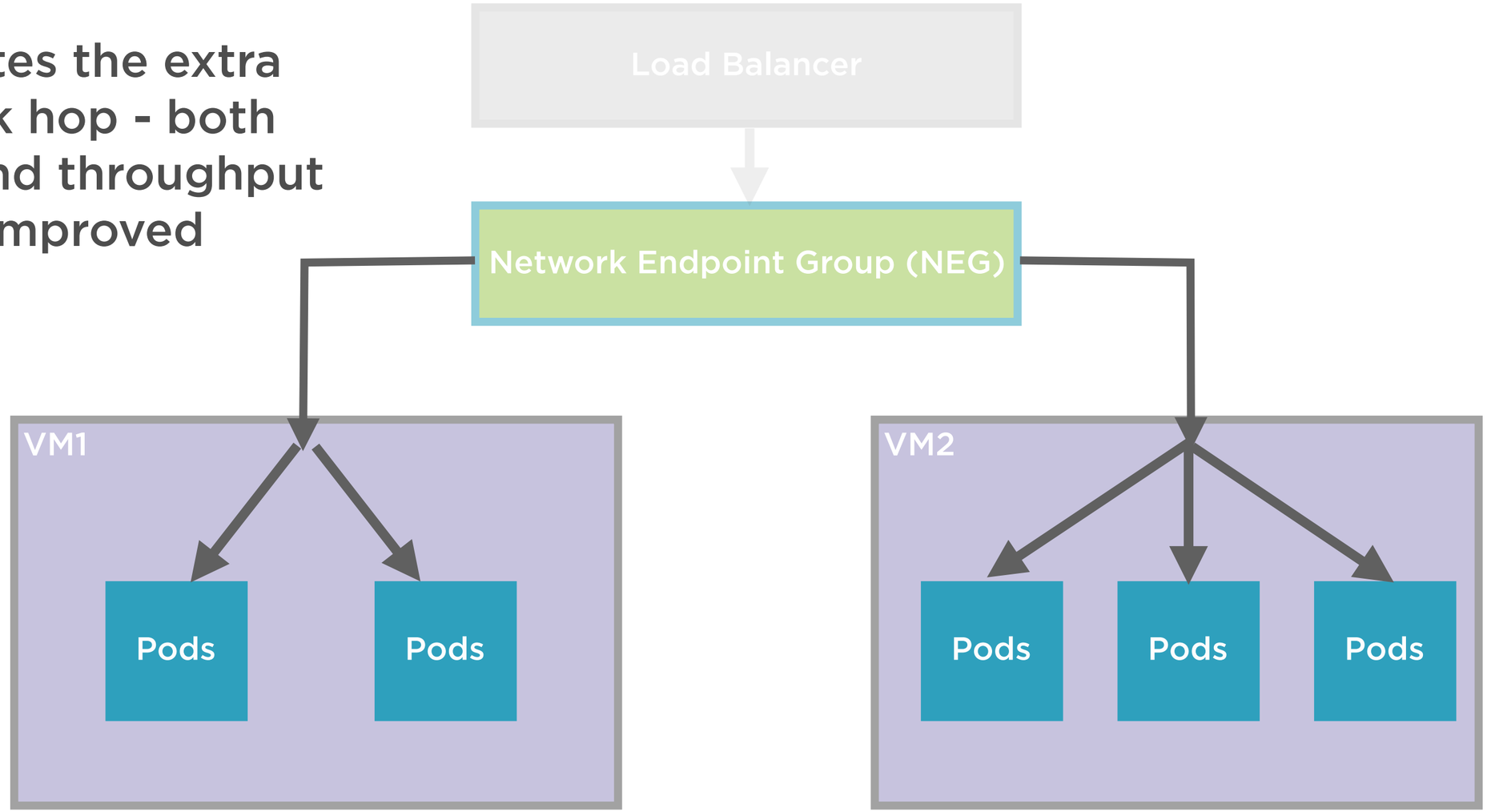
Allows traffic distribution in a granular fashion amongst containers running within a VM

Network Endpoint Groups

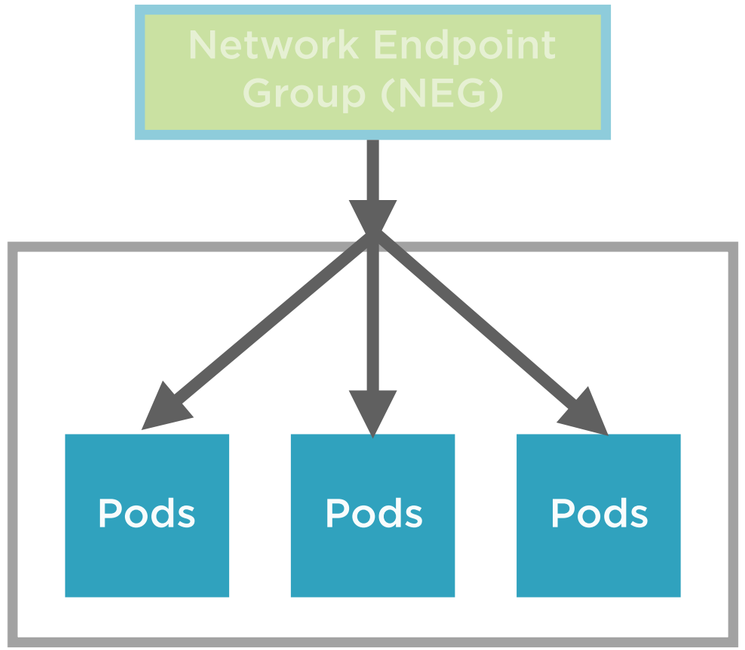
Zonal resources which represent a collection of IP address and port combinations. Can be used as a backend for HTTP(S), TCP proxy and SSL proxy load balancing

Traffic Distributed Directly to Pods

Eliminates the extra network hop - both latency and throughput are improved



Container-native Load Balancing



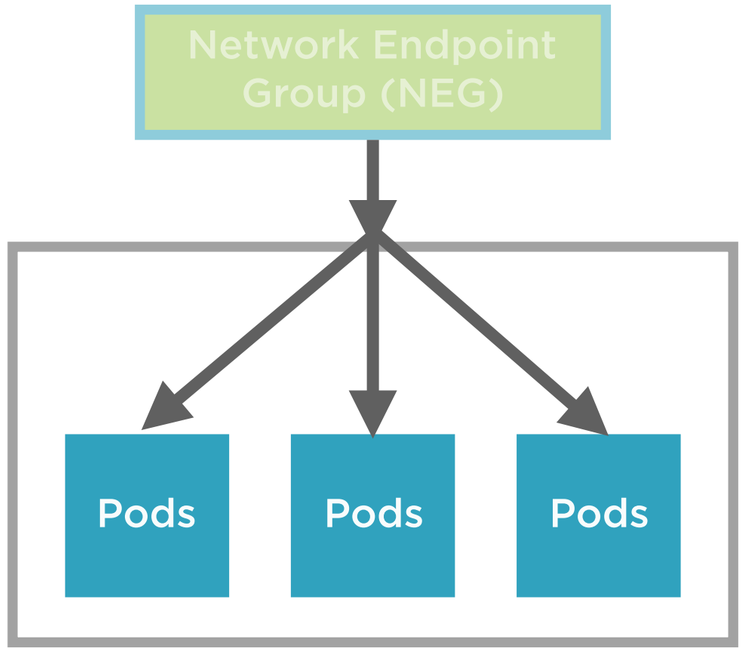
Pods are first-class citizens for load balancing

Network endpoint groups target IP-port pairs

Improved network performance by eliminating iptables hop

Makes troubleshooting easier

Requirements



Google Kubernetes Engine version 1.10

VPC-native clusters with alias IPs

Demo

Implement container-native load balancing for an application running on a Kubernetes cluster

Summary

Introducing Container-native load balancing

Creating VPC-native cluster

Instantiating deployment, service and ingress

Verifying ingress

Delete Resources

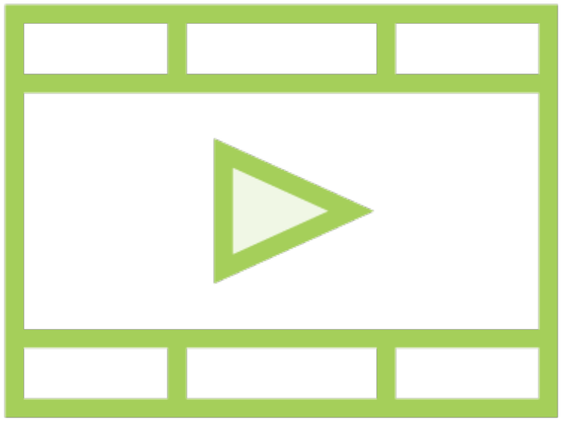
Cloud storage buckets

Load balancers

DNS managed zones

Kubernetes clusters

Related Courses



Leveraging Architectural Design Patterns on the Google Cloud

Azure Kubernetes Service (AKS) – The Big Picture