# Inside Kubernetes Architectural Fundamentals

**Anthony E. Nocentino**
aen@centinosystems.com

# Anthony E. Nocentino

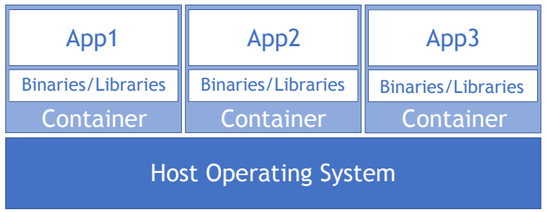
- **Consultant and Trainer**
- **Founder and President of Centino Systems**
  - Specialize in system architecture and performance
  - Masters Computer Science
  - Microsoft MVP - Data Platform - 2017 - 2020
  - Linux Foundation Certified Engineer
  - Friend of Redgate - 2015-2019
- **email**: aen@centinosystems.com
- **Twitter**: @nocentino
- **Blog**: www.centinosystems.com/blog
- **Pluralsight Author**: www.pluralsight.com

# Agenda

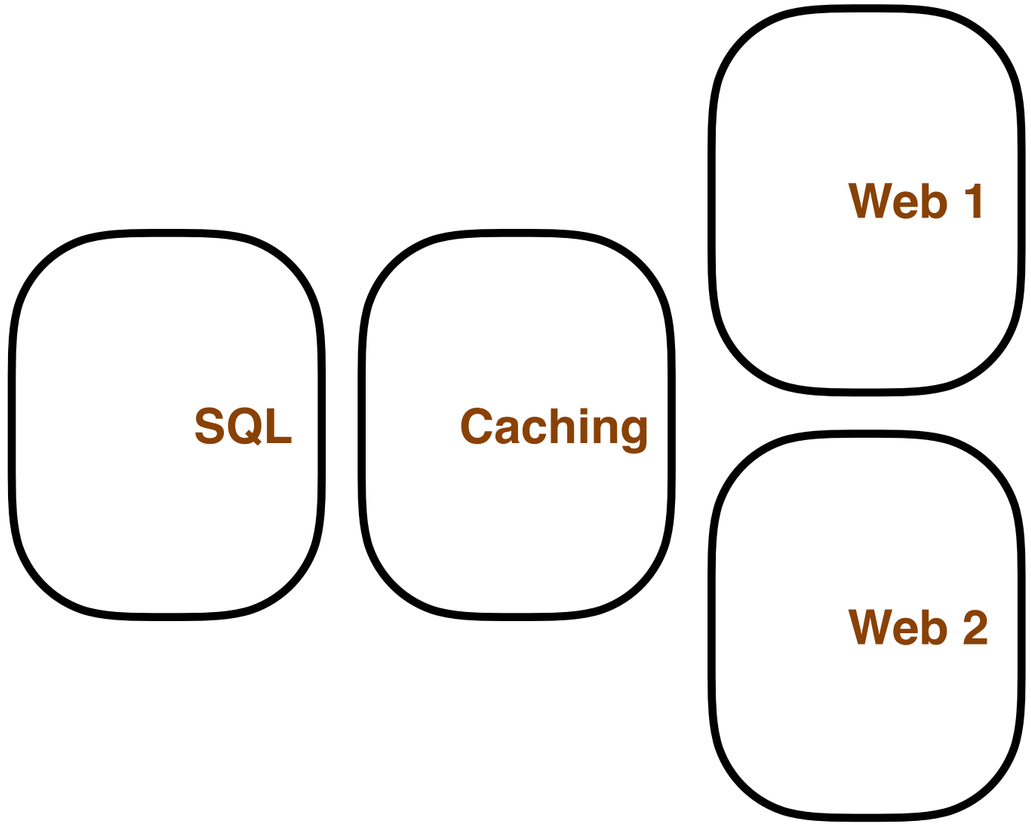
- What is Kubernetes
- Kubernetes API Objects
- Exploring Kubernetes Architecture
- Deploying Applications
- Production Ready Clusters

# Container Based Application Deployment

- Single-tier applications - anything written by IBM
- Multi-tier applications - Service oriented, Client/Server…
- Micro-services - smaller, more easily changed units



# Modern Application Deployment




- Where do I run the application?
- How do I scale the application?
- How do I consistently deploy?
- How do I provide consistent services in a loosely connected system?

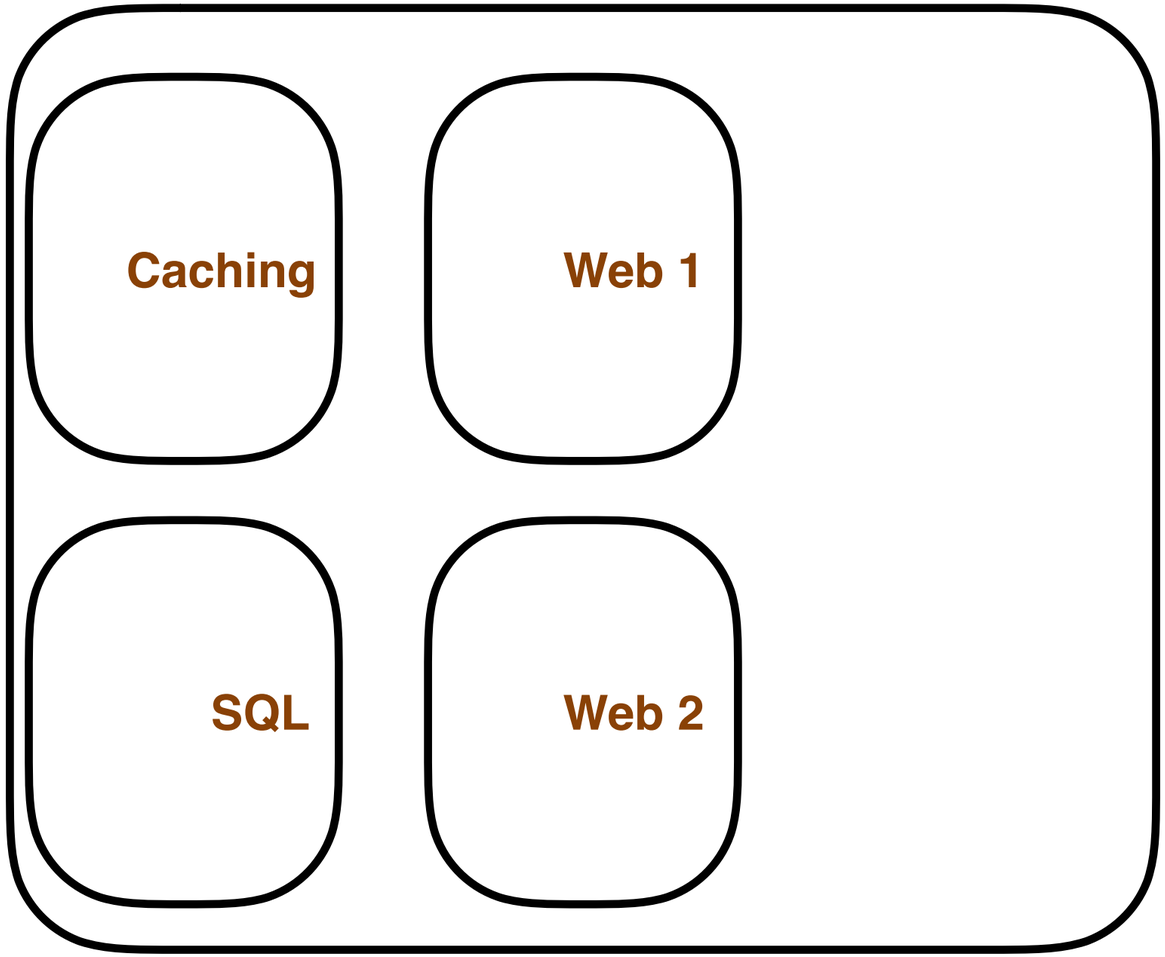
# What is Kubernetes?

- Container Orchestrator
- Infrastructure Abstraction
- Desired State

# Kubernetes Benefits

- Managing state, starting things and keeping them up
- Speed and consistency of deployment
- Ability to absorb change quickly
- Ability to recovery quickly
- Workload placement in cluster
- Hide complexity in Cluster
- Persistent application access endpoints

# Kubernetes Cluster



# Container Orchestrators

- Docker Swarm
- Red Hat OpenShift
- Managed Services
  - Azure Kubernetes Services (AKS)
  - Google Kubernetes Engine (GKE)
  - Amazon Elastic Container Service for Kubernetes (EKS)

# Kubernetes API

- **API Objects** - Represent resources in your system
  - Programmatically expose the resource in our data center
    - Pods - your container based applications
    - Controllers - maintain desired state
    - Services - persistent access to your apps
    - Storage - persistent storage for your data
    - …and more

# Pods

- One or more containers
- It’s your application or service
- The most basic unit of work
- Unit of scheduling
- Ephemeral - no Pod is ever “redeployed”

# Controllers

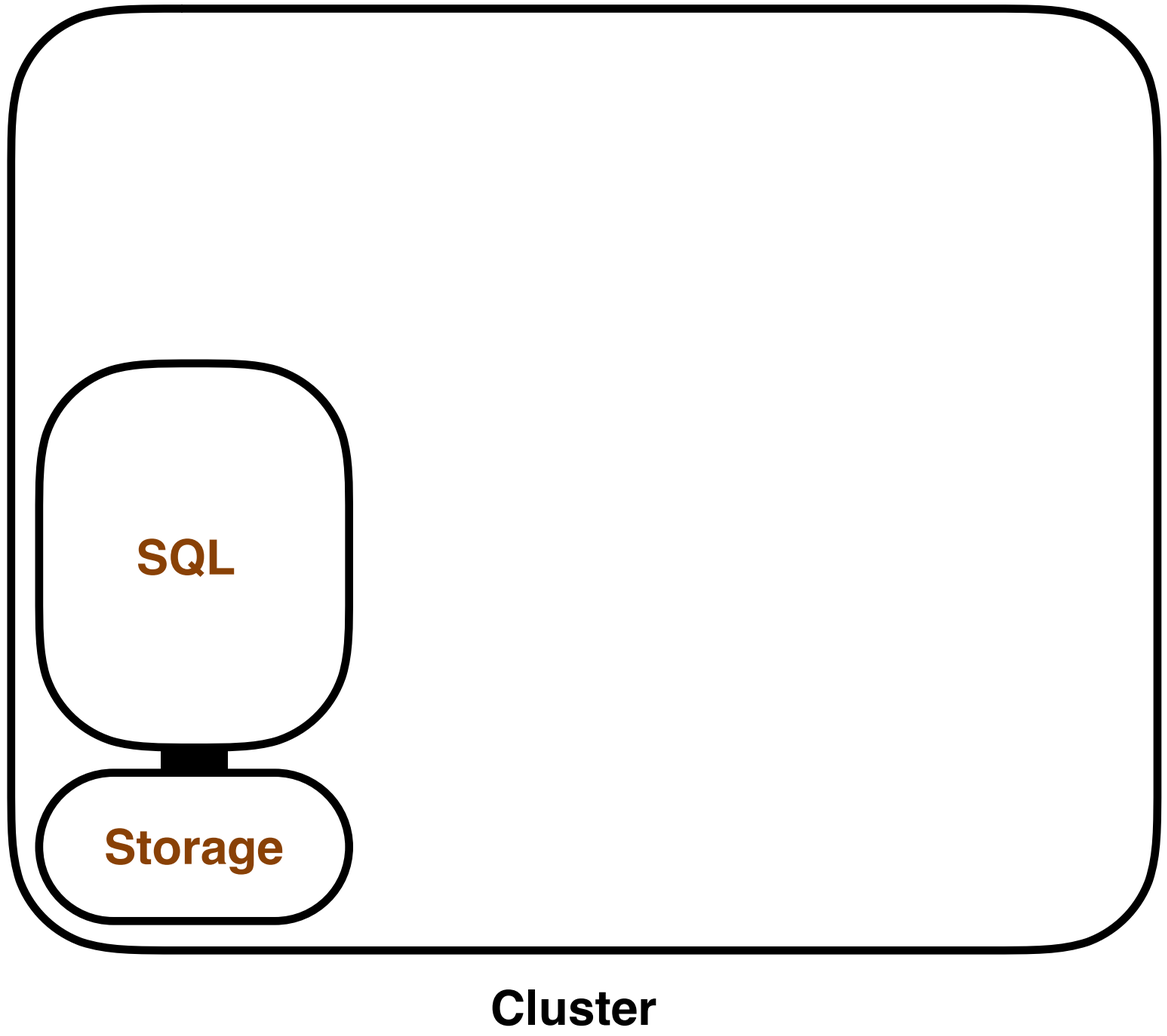
- Create and manage Pods for you
- Define your desired state
- Respond to Pod State and Health
- `ReplicaSet`
- `Deployment`

# Services

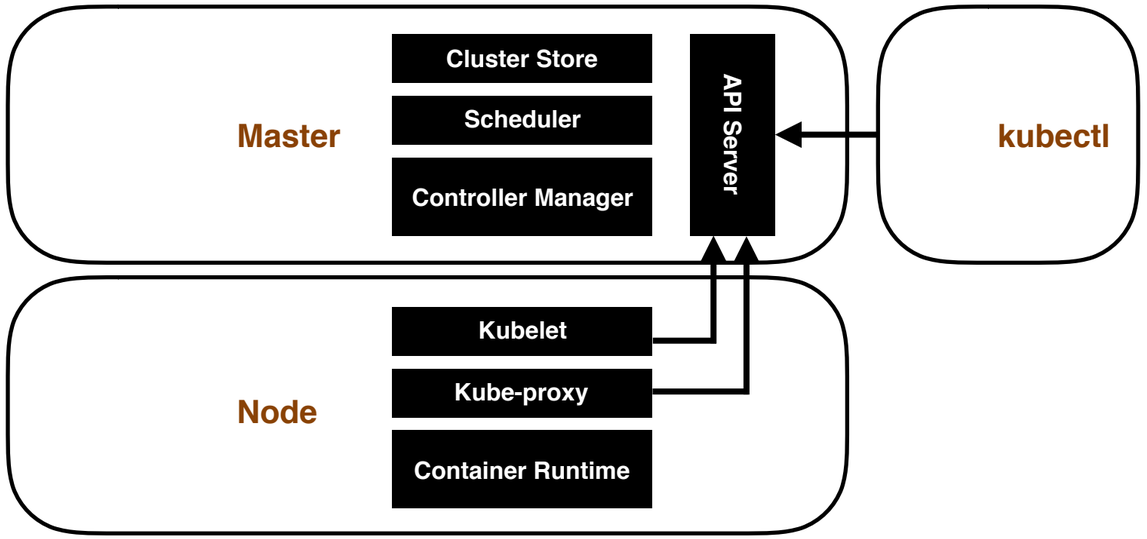
- Adds persistency to our ephemeral world
- Pods can come and go based on health and Controller operations
- Networking abstraction for Pod access
- IP and DNS name for the service
- Load balancing
- Recreated Pods automatically updated
- Scaled by adding/removing Pods

# Storage

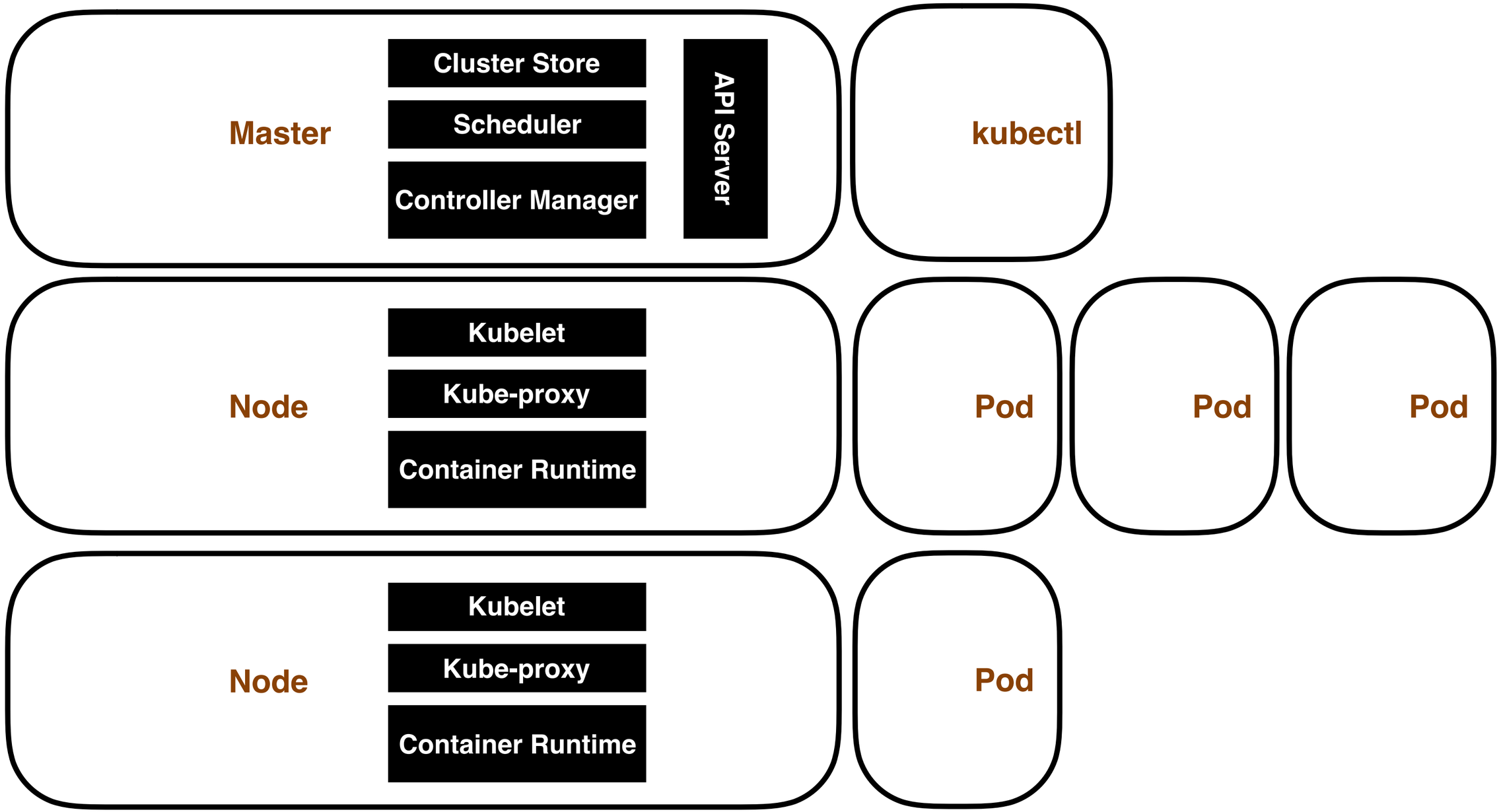
- Persistent Volumes
  - Pod independent storage
  - Administrator defined storage
- Persistent Volume Claims
  - The Pod “claims” the PV
  - Decouples the Pod and the storage
- StorageClass
  - Dynamic Provisioning



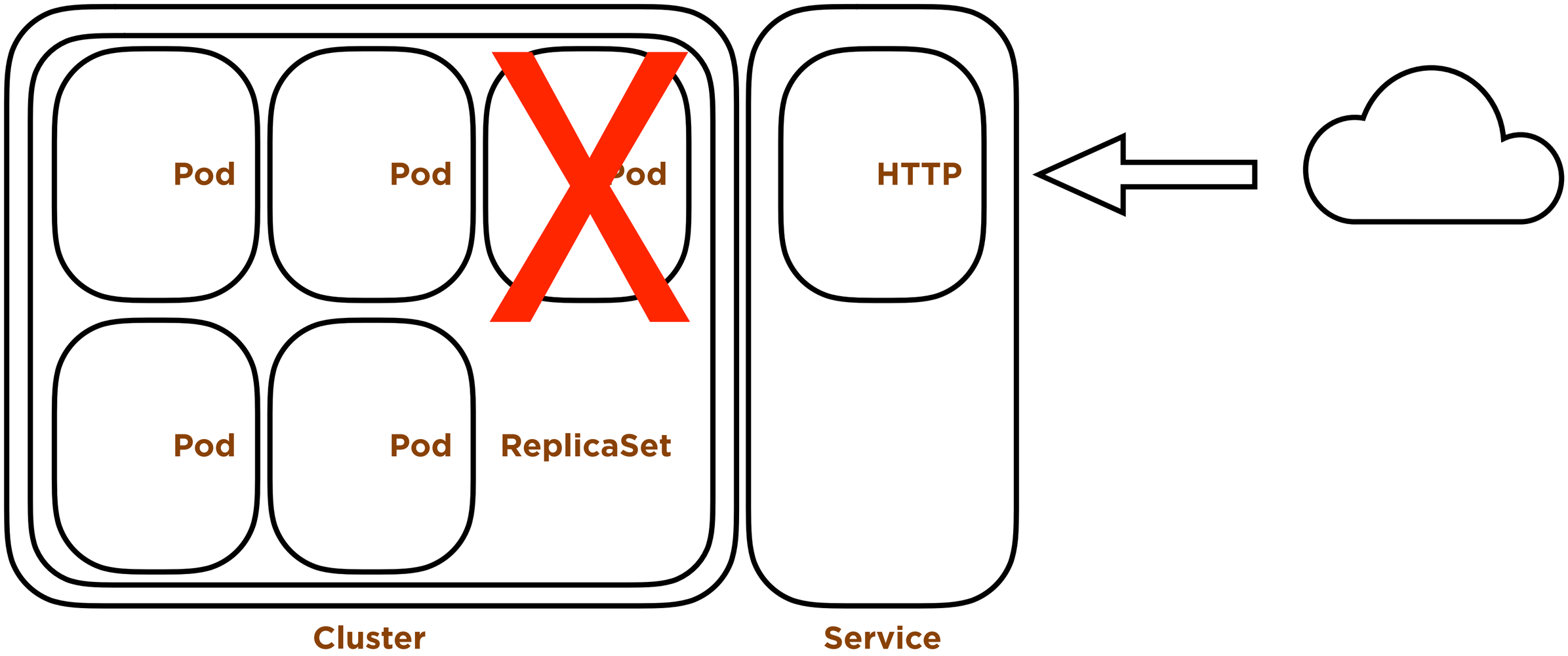
# Exploring Kubernetes Architecture



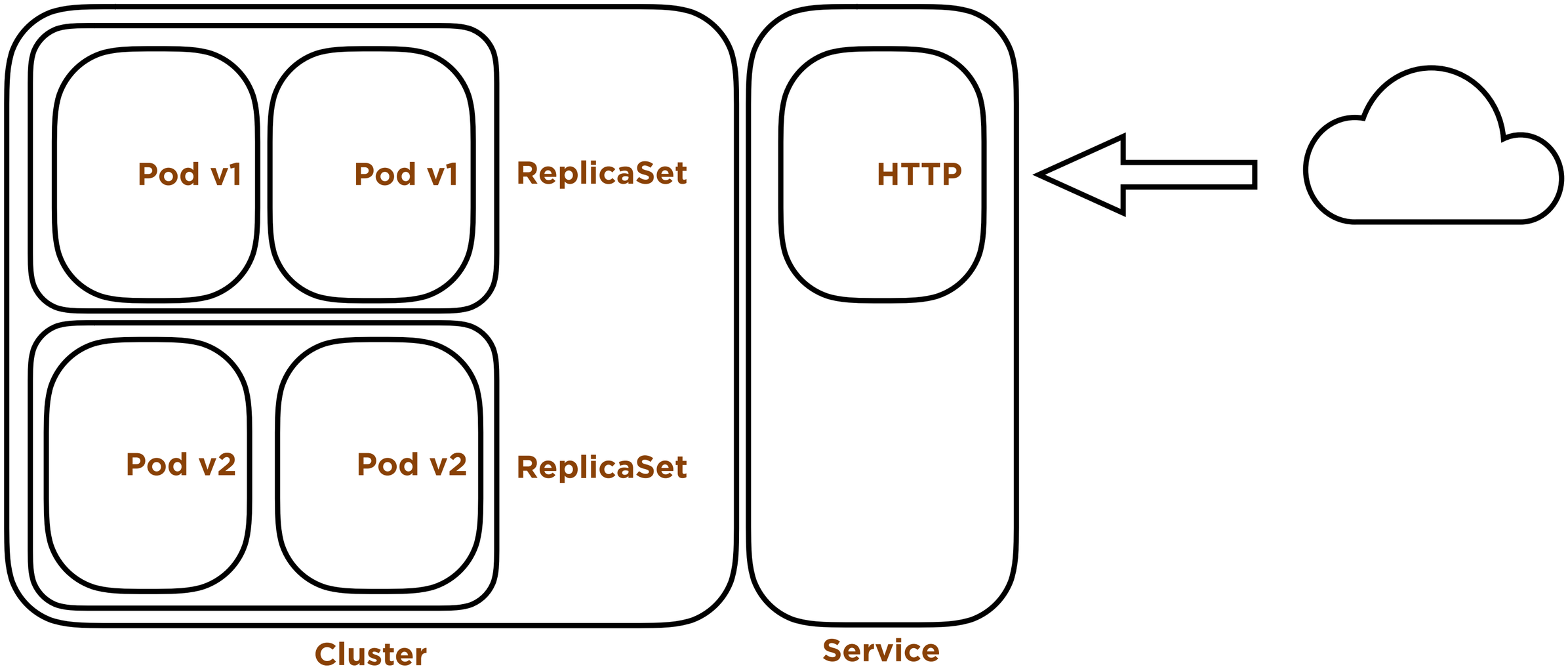
# Controller Operations - `ReplicaSet`



# Services



# Controller Operations - `Deployment`



# Deploying Applications

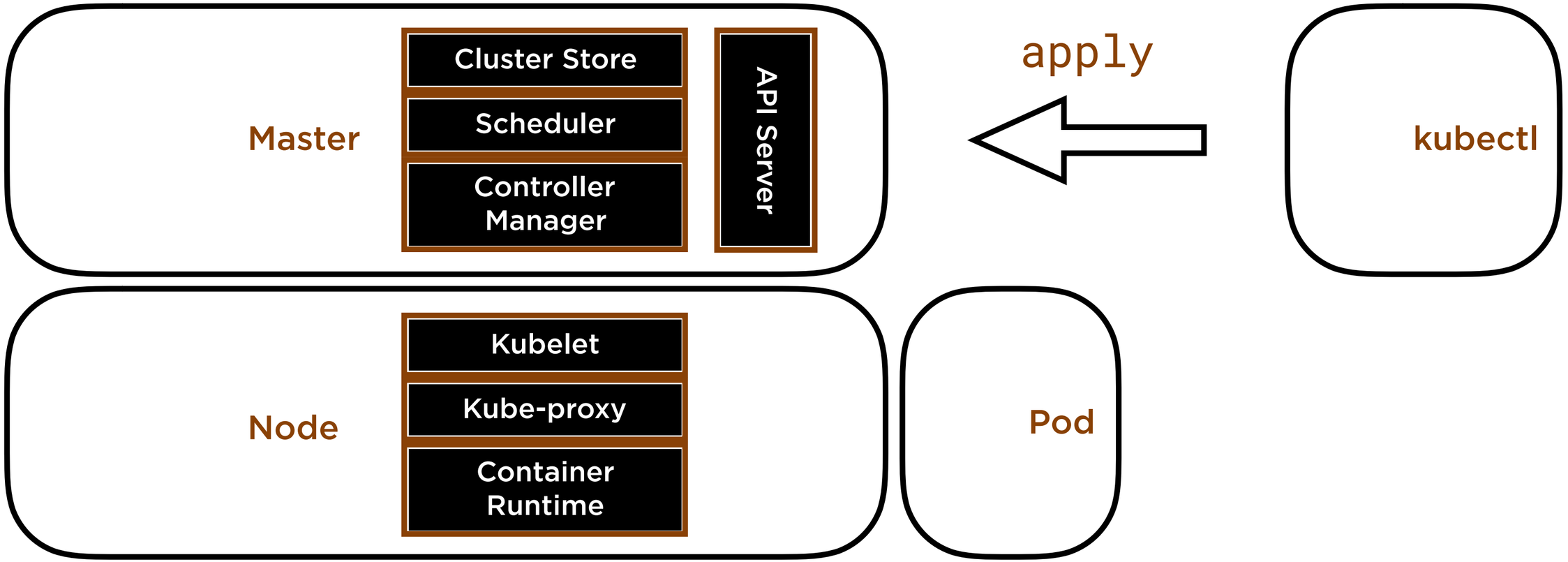
- Imperative
- Declarative
- YAML and JSON

# Declarative Deployment - Manifests

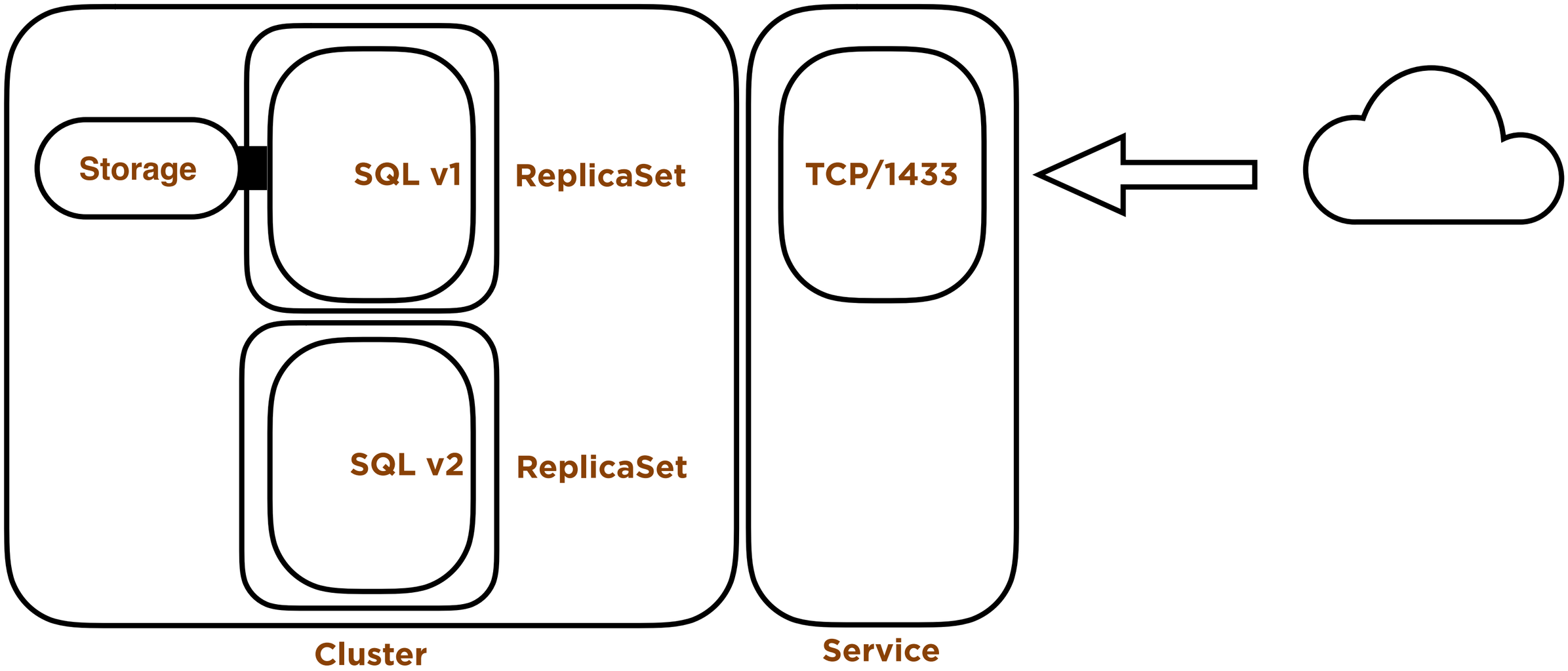
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
```

```
kubectl apply -f nginx.yaml
```

# Application Deployment Process

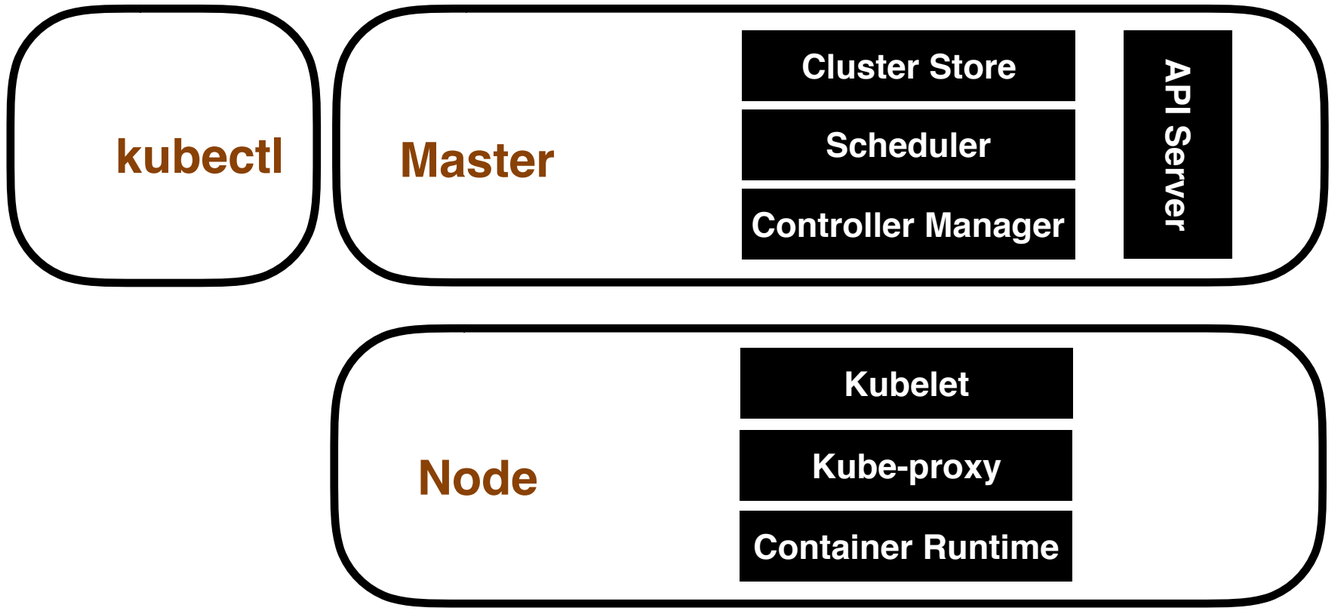


# Decoupling Data and Computation



# Demo!

- Deploying SQL Server in a **`Deployment`** with Persistent Storage
  - Recovery Scenario
  - Upgrading SQL Server



# Building Production Ready Clusters

- Scalability - number of Nodes
- Intra-cluster communication patterns (Network connectivity)
- High Availability
  - API Server - Load Balanced
  - etcd - Multiple Replicas
- Disaster Recovery
  - etcd Backups
- Persistent Volumes

# Review

- What is Kubernetes
- Kubernetes API Objects
- Exploring Kubernetes Architecture
- Deploying Applications
- Production Ready Clusters

# More Resources

- **Docker for Windows/Mac**
- **Minikube**
- **Managed Service Providers**
  - Azure Kubernetes Service **(AKS)**
    - https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough
  - **Elastic Container Service for Kubernetes (EKS)**
    - https://aws.amazon.com/getting-started/projects/deploy-kubernetes-app-amazon-eks/
  - **Google Kubernetes Engine (GKE)**
    - https://cloud.google.com/kubernetes-engine/docs/how-to/
- **Pluralsight! - Kubernetes Installation and Configuration Fundamentals**
  - https://app.pluralsight.com/profile/author/anthony-nocentino

# Need more data or help?

**http://www.centinosystems.com/blog/talks/**

Links to resources
Demos
Presentation
Pluralsight

**aen@centinosystems.com**
**@nocentino**
**www.centinosystems.com**

**Solving tough business challenges with technical innovation**

Centino systems

# Thank You!