

Inside Kubernetes Architecture Fundamentals

Anthony E. Nocentino
aen@centinosystems.com





Event Feedback
(not optional!)

<http://bit.ly/DataGrillen2019Event>



Session Feedback Day 2 (not optional!)

<http://bit.ly/DataGrillen2019Day2>



Anthony E. Nocentino

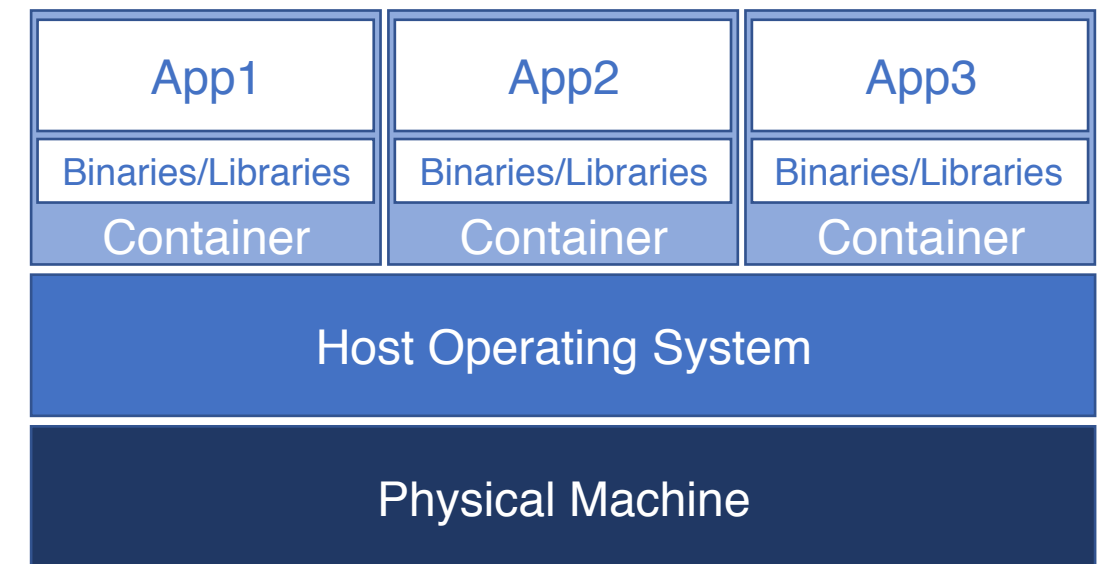
- **Consultant and Trainer**
- **Founder and President of Centino Systems**
 - Specialize in system architecture and performance
 - Masters Computer Science
 - Microsoft MVP - Data Platform - 2017 - 2018
 - Linux Foundation Certified Engineer
 - Friend of Redgate - 2015-2019
- **email:** aen@centinosystems.com
- **Twitter:** @nocentino
- **Blog:** www.centinosystems.com/blog
- **Pluralsight Author:** www.pluralsight.com



Agenda

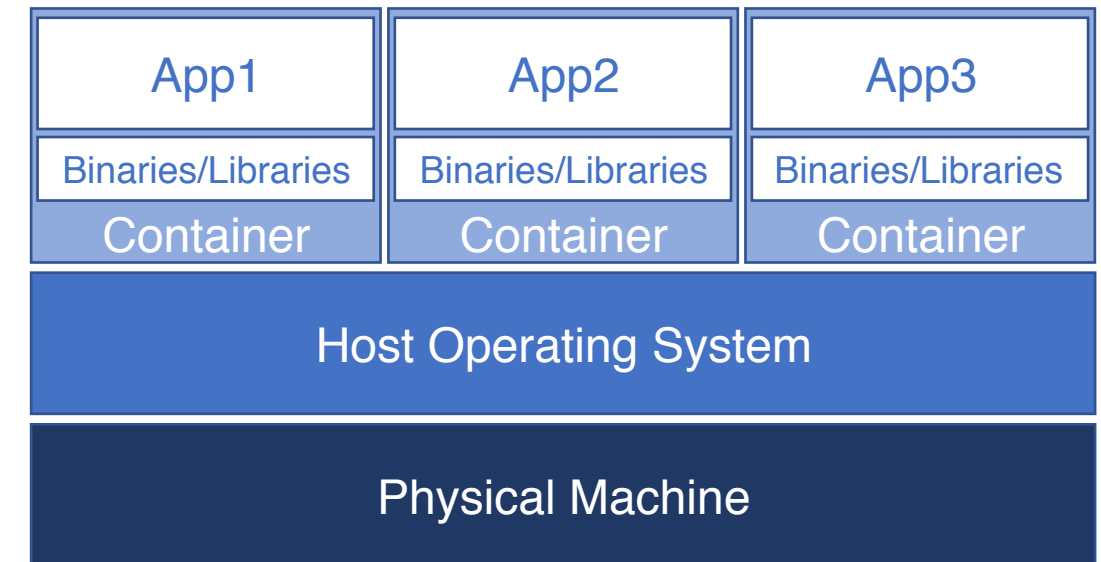
- What is Kubernetes
- Kubernetes API Objects
- Exploring Kubernetes Architecture
- Deploying Applications
- Production Ready Clusters

Container Based Application Deployment



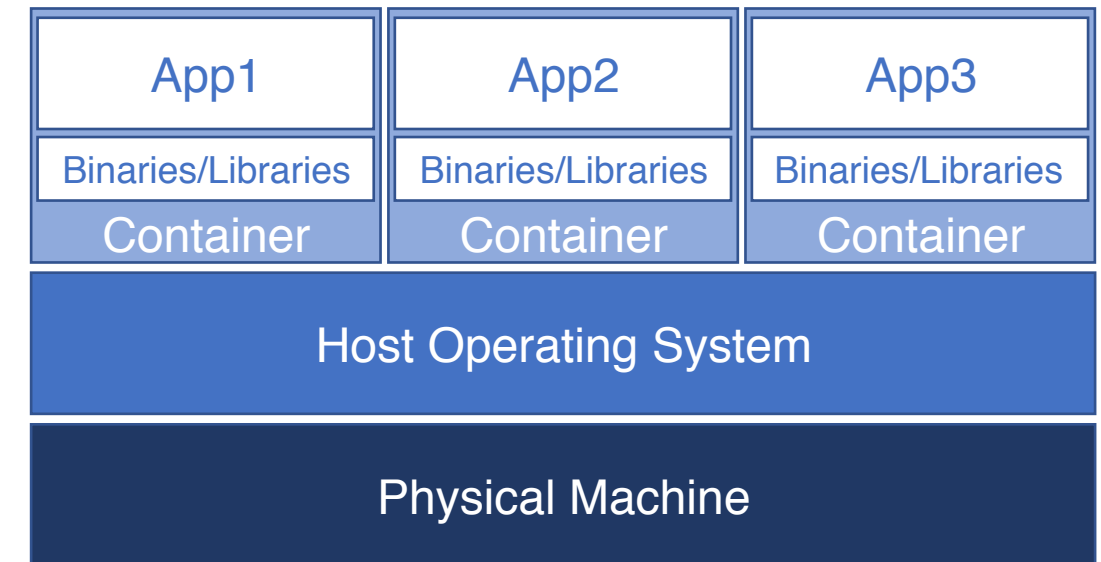
Container Based Application Deployment

- Single-tier applications - anything written by IBM



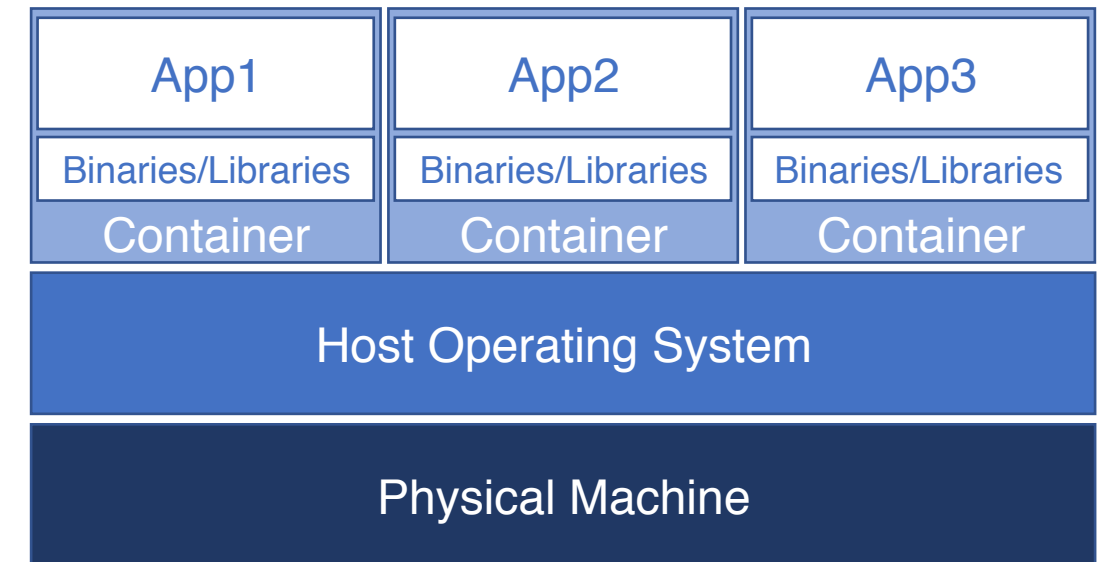
Container Based Application Deployment

- Single-tier applications - anything written by IBM
- Multi-tier applications - Service oriented, Client/Server...



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

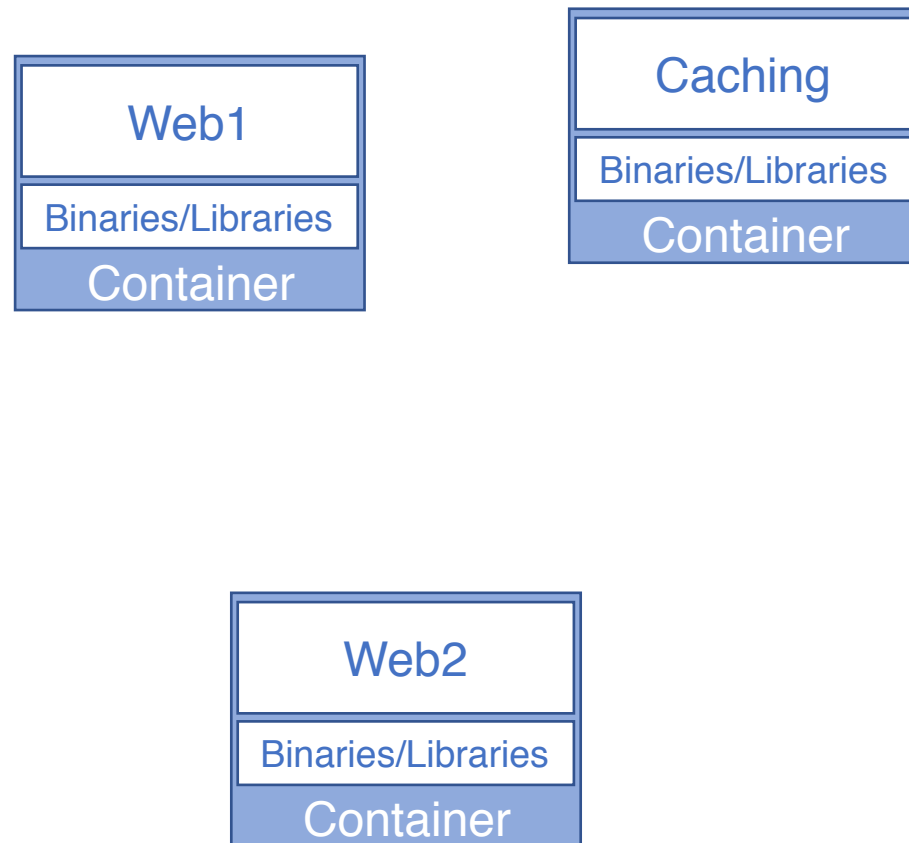
Modern Application Deployment



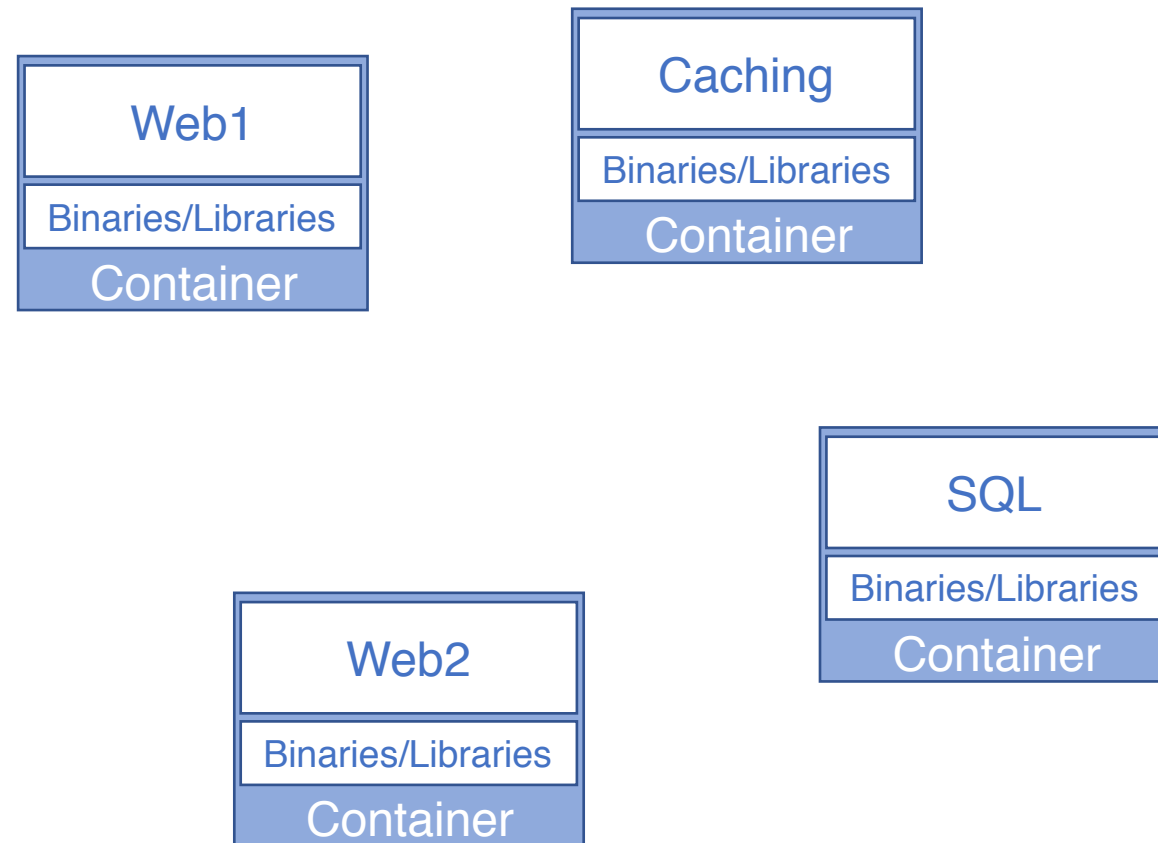
Modern Application Deployment



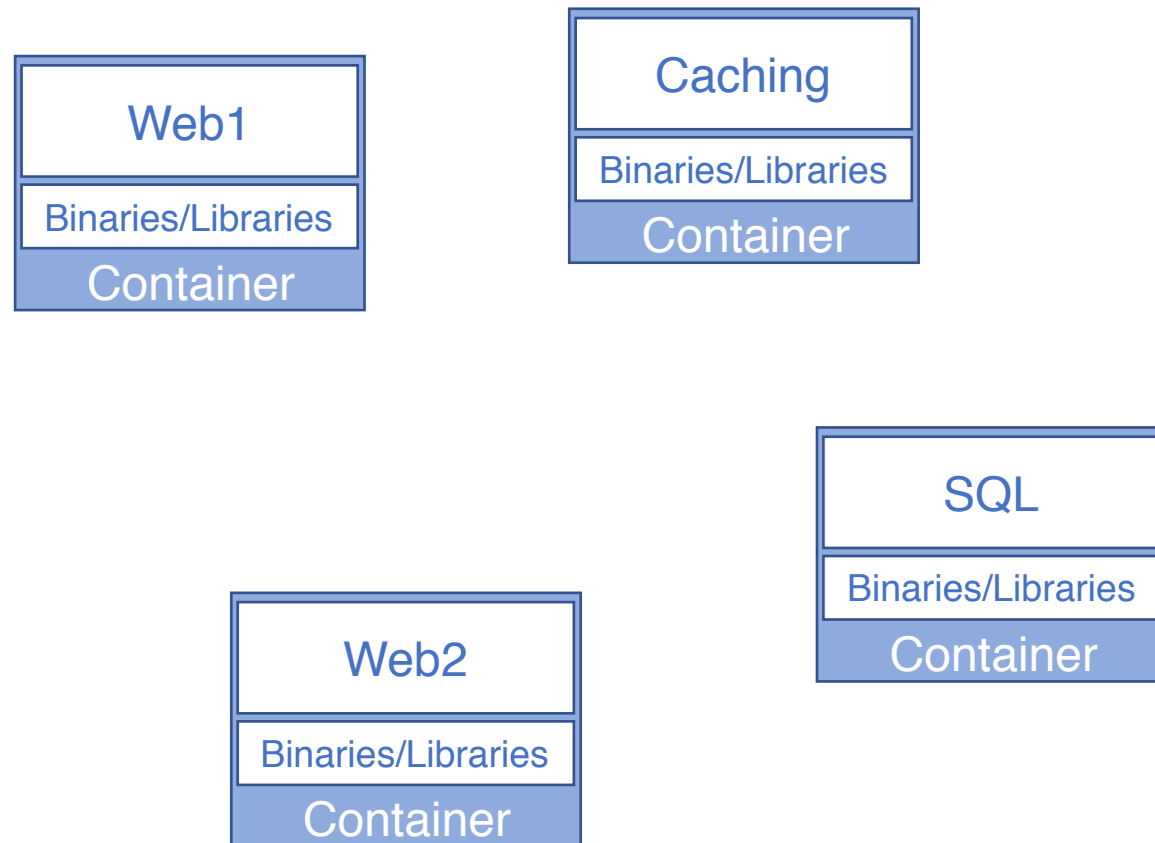
Modern Application Deployment



Modern Application Deployment

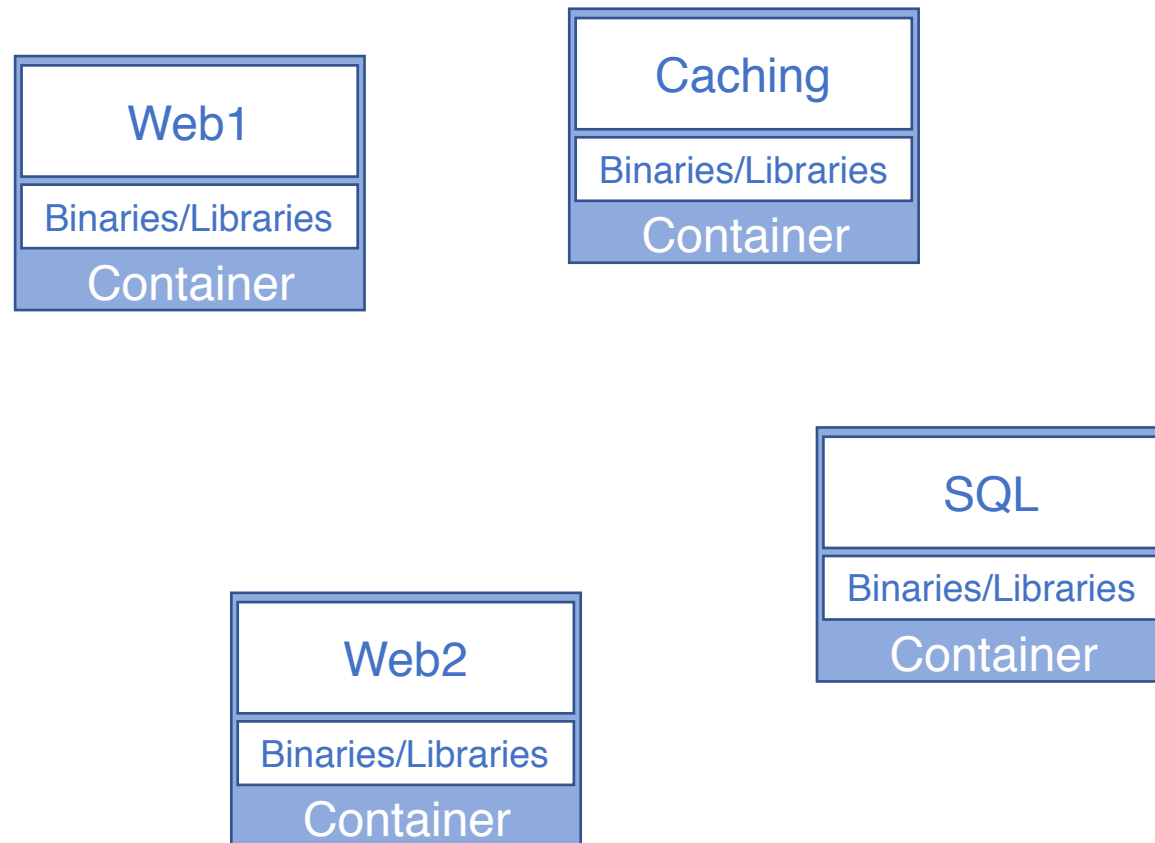


Modern Application Deployment



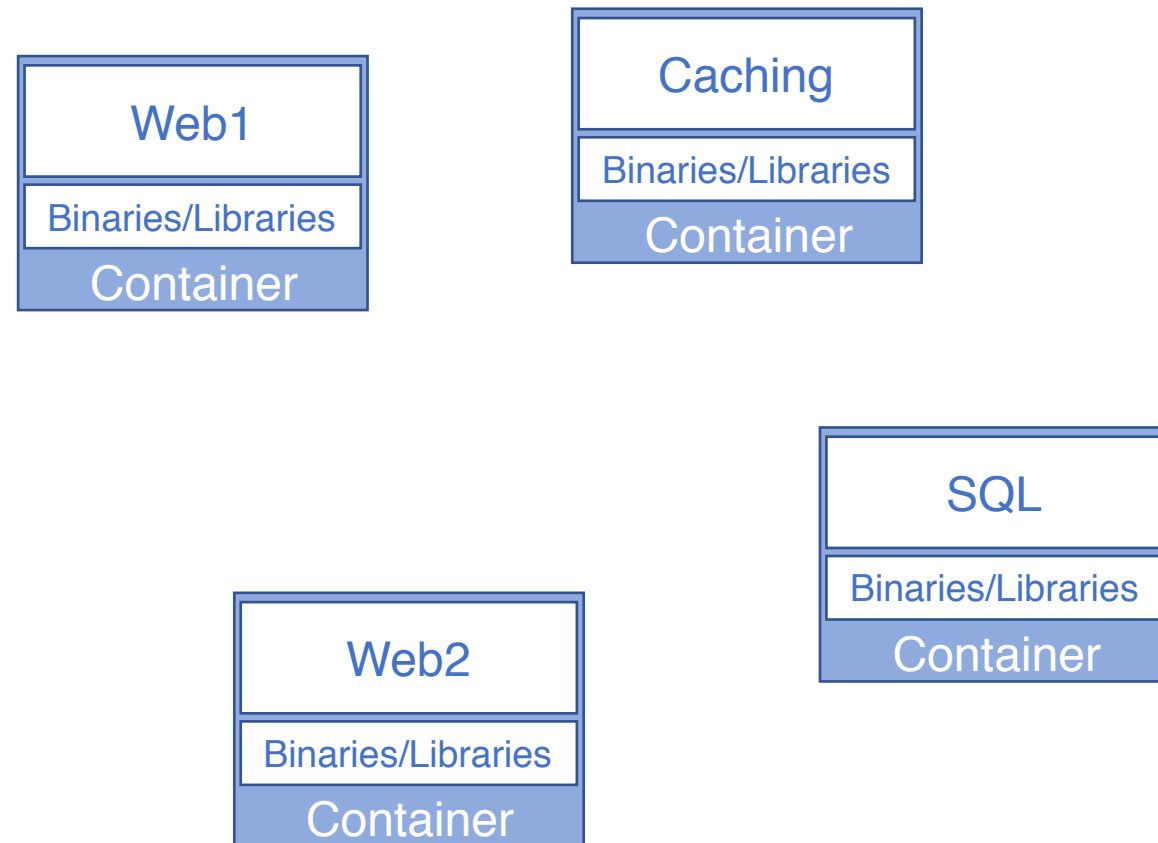
- Where do I run the application?

Modern Application Deployment



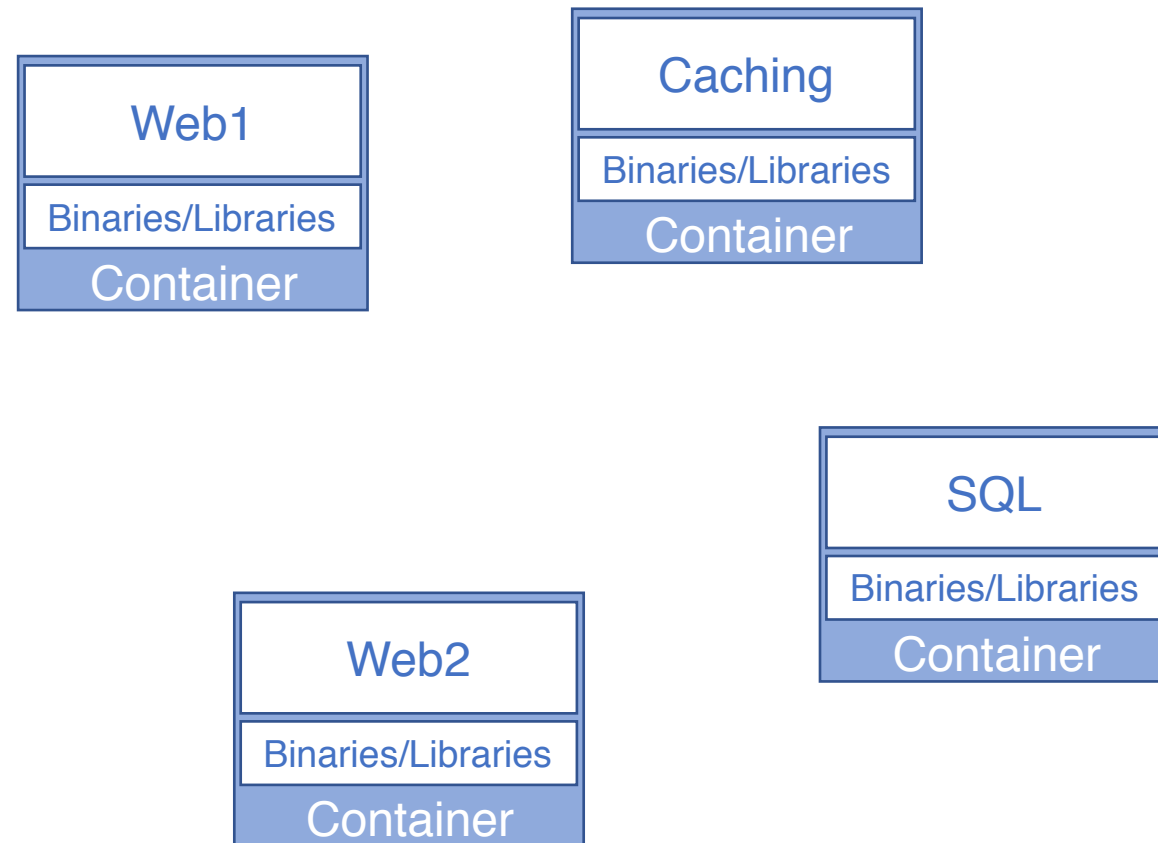
- Where do I run the application?
- How do I scale the application?

Modern Application Deployment



- Where do I run the application?
- How do I scale the application?
- How do I consistently deploy?

Modern Application Deployment



- Where do I run the application?
- How do I scale the application?
- How do I consistently deploy?
- How do I provide access to services?

What is Kubernetes?



What is Kubernetes?

- Container Orchestrator



What is Kubernetes?

- Container Orchestrator
- Infrastructure Abstraction



What is Kubernetes?

- Container Orchestrator
- Infrastructure Abstraction
- Desired State



Kubernetes Benefits

Kubernetes Benefits

- Managing state, starting things and keeping them up

Kubernetes Benefits

- Managing state, starting things and keeping them up
- Speed and consistency of deployment

Kubernetes Benefits

- Managing state, starting things and keeping them up
- Speed and consistency of deployment
- Ability to absorb change quickly

Kubernetes Benefits

- Managing state, starting things and keeping them up
- Speed and consistency of deployment
- Ability to absorb change quickly
- Ability to recovery quickly

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

Kubernetes Benefits

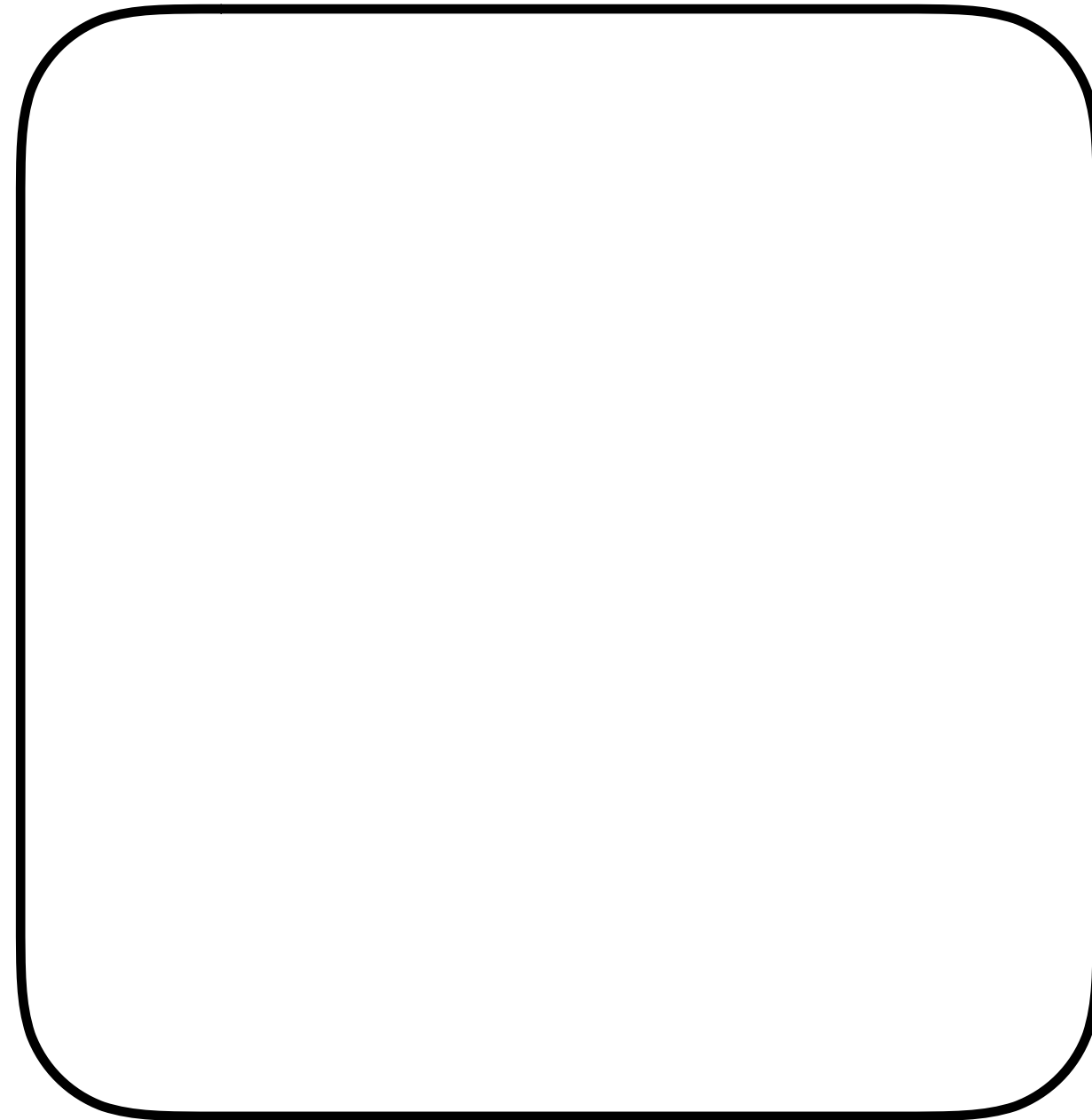
- Managing state, starting things and keeping them up
- Speed and consistency of deployment
- Ability to absorb change quickly
- Ability to recovery quickly
- Workload placement in cluster
- Hide complexity in Cluster

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

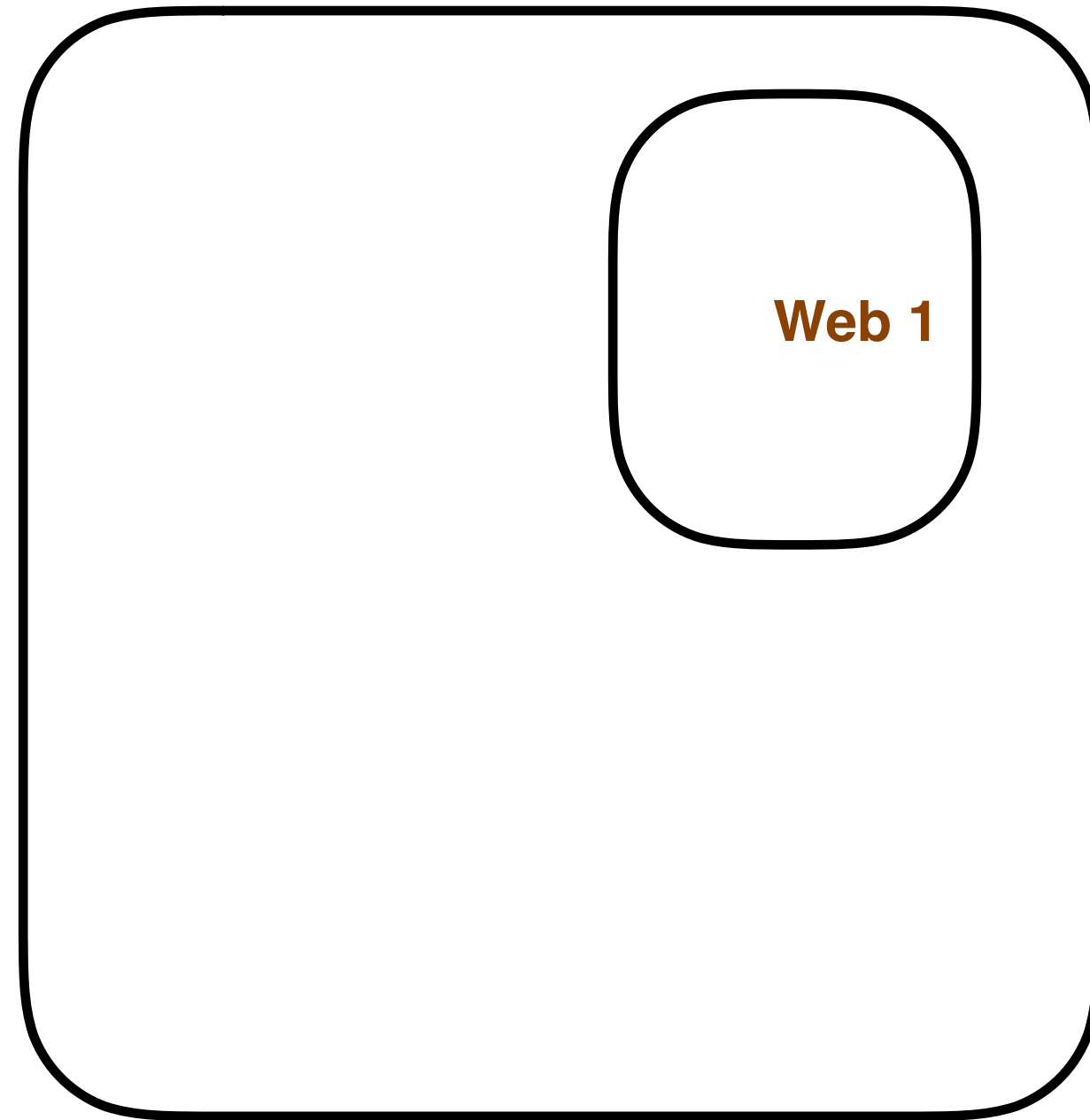
Kubernetes Cluster

Kubernetes Cluster



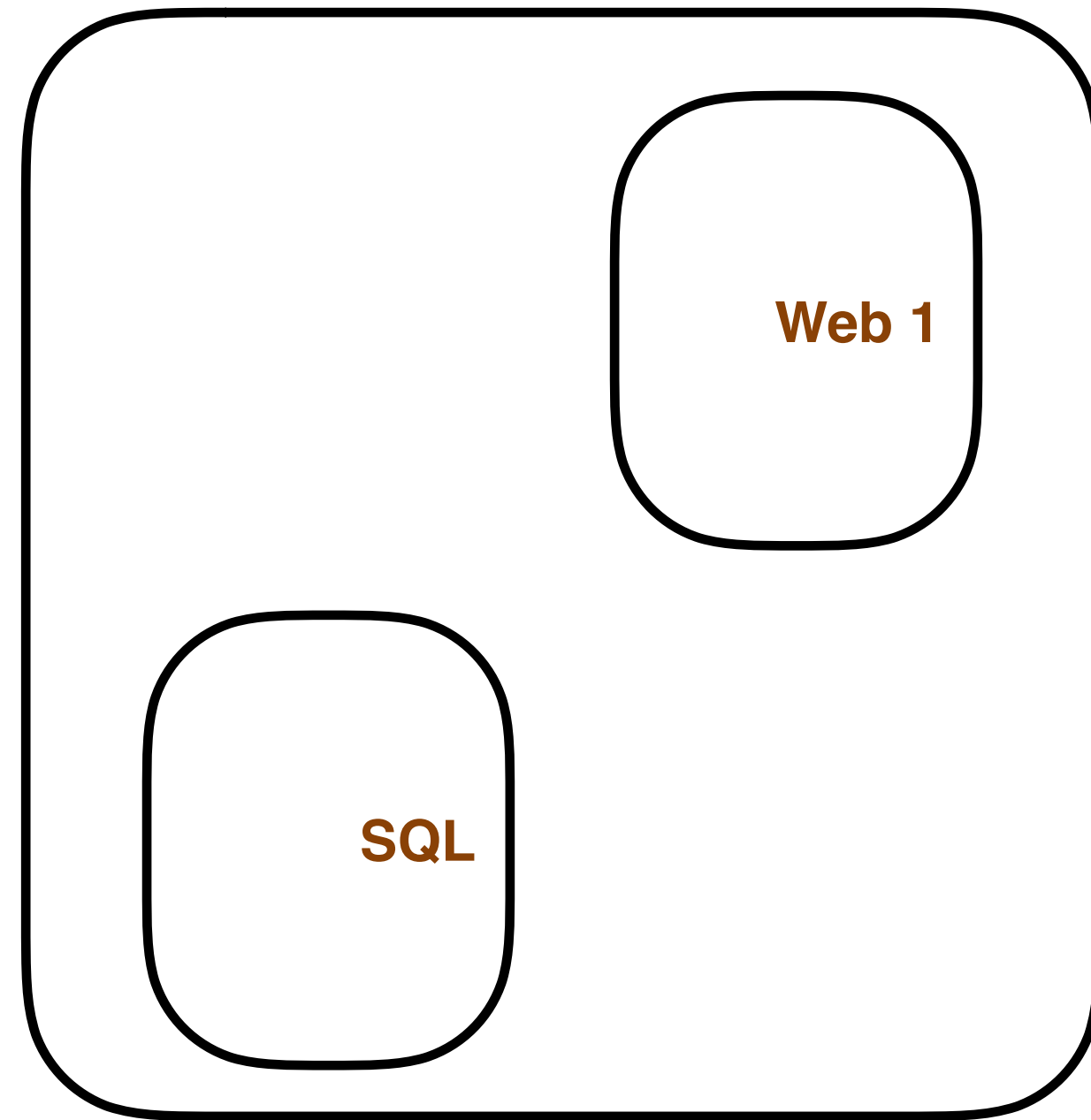
Cluster

Kubernetes Cluster



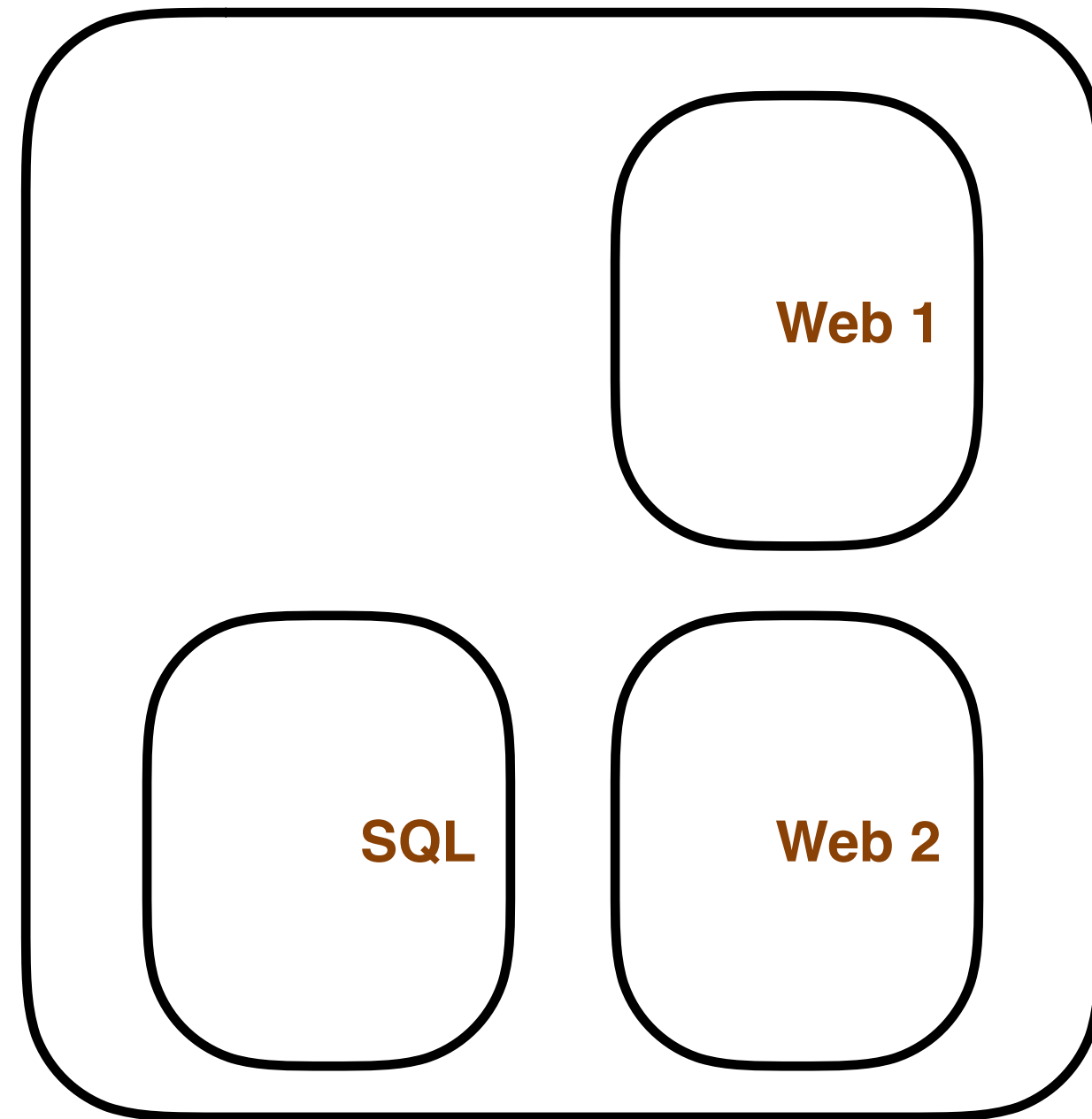
Cluster

Kubernetes Cluster



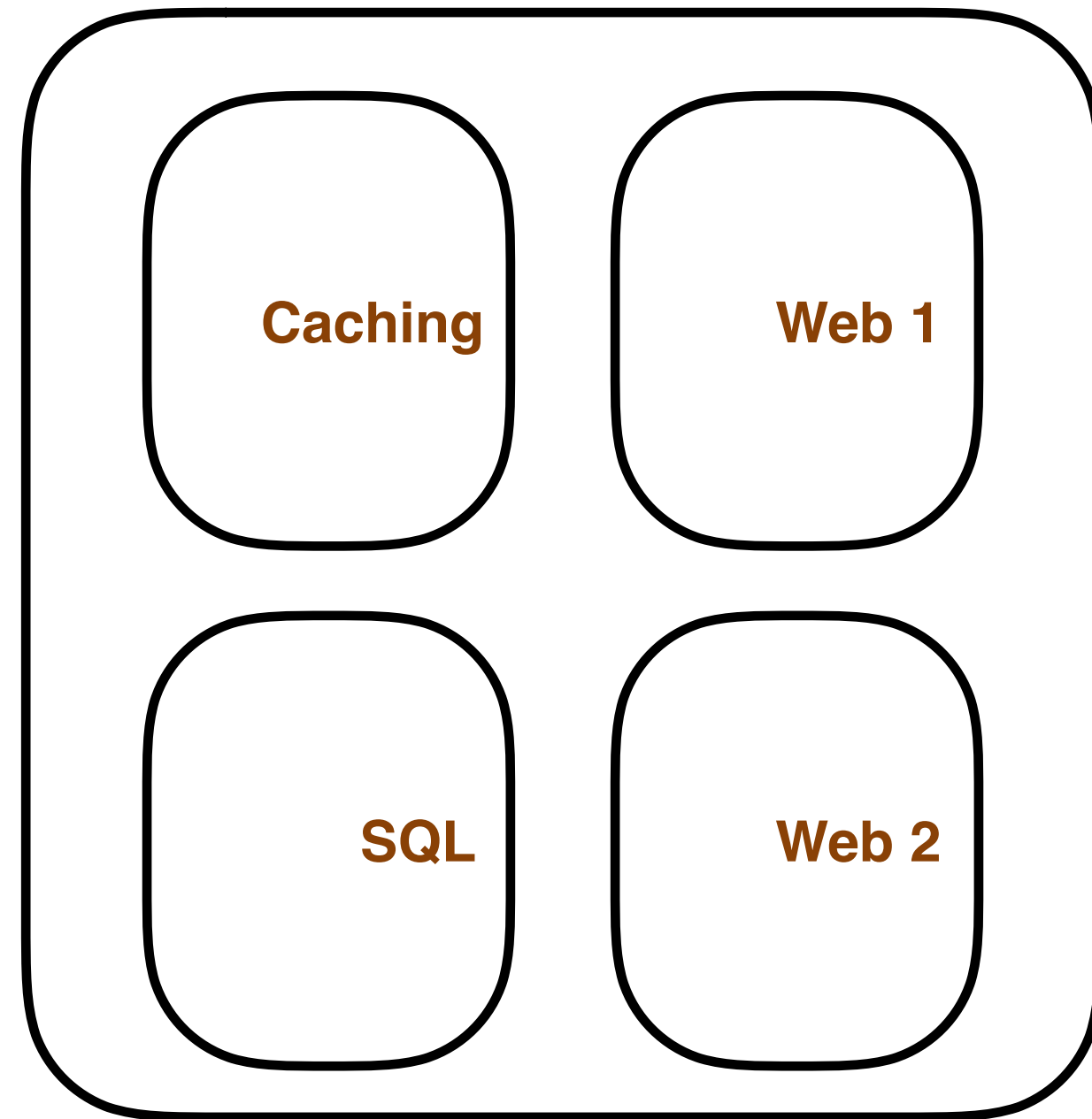
Cluster

Kubernetes Cluster



Cluster

Kubernetes Cluster



Cluster

Container Orchestrators

Container Orchestrators

- Docker Swarm

Container Orchestrators

- Docker Swarm
- Red Hat OpenShift

Container Orchestrators

- Docker Swarm
- Red Hat OpenShift
- Managed Services

Container Orchestrators

- Docker Swarm
- Red Hat OpenShift
- Managed Services
 - Azure Kubernetes Services (AKS)

Container Orchestrators

- Docker Swarm
- Red Hat OpenShift
- Managed Services
 - Azure Kubernetes Services (AKS)
 - Google Kubernetes Engine (GKE)

Container Orchestrators

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

Getting Kubernetes

Getting Kubernetes

Desktop

Getting Kubernetes

Desktop

kubeadm

Getting Kubernetes

Desktop

kubeadm

From Scratch

Getting Kubernetes

Desktop

kubeadm

From Scratch

<https://kubernetes.io/docs/setup/scratch/>
<https://github.com/kelseyhightower/kubernetes-the-hard-way/>

Getting Kubernetes

Desktop

kubeadm

From Scratch

Cloud Scenarios

<https://kubernetes.io/docs/setup/scratch/>
<https://github.com/kelseyhightower/kubernetes-the-hard-way/>

Getting Kubernetes

Desktop

kubeadm

From Scratch

Cloud Scenarios

<https://kubernetes.io/docs/setup/scratch/>

<https://github.com/kelseyhightower/kubernetes-the-hard-way/>

Kubernetes Installation and Configuration Fundamentals

Kubernetes API

Kubernetes API

- **API Objects** - Represent resources in your system

Kubernetes API

- **API Objects** - Represent resources in your system
 - Programmatically expose the resource in our data center

Kubernetes API

- **API Objects** - Represent resources in your system
 - Programmatically expose the resource in our data center
 - **Pods** - your container based applications

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

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

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

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

Pods

- One or more containers

Pods

- One or more containers
- It's your application or service

Pods

- One or more containers
- It's your application or service
- The most basic unit of work

Pods

- One or more containers
- It's your application or service
- The most basic unit of work
- Unit of scheduling

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

Controllers

- Create and manage Pods for you

Controllers

- Create and manage Pods for you
- Define your desired state

Controllers

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

Controllers

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

Controllers

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

Services

Services

- Adds persistency to our ephemeral world

Services

- Adds persistency to our ephemeral world
- Pods can come and go based on health and Controller operations

Services

- Adds persistency to our ephemeral world
- Pods can come and go based on health and Controller operations
- Networking abstraction for Pod access

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

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

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 endpoints are automatically updated

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 endpoints are automatically updated
- Scaled by adding/removing Pods

Storage

Storage

- Persistent Volumes

Storage

- Persistent Volumes
 - Pod independent storage

Storage

- Persistent Volumes
 - Pod independent storage
 - Administrator defined storage

Storage

- Persistent Volumes
 - Pod independent storage
 - Administrator defined storage
- Persistent Volume Claims

Storage

- Persistent Volumes
 - Pod independent storage
 - Administrator defined storage
- Persistent Volume Claims
 - The Pod “claims” the PV

Storage

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

Storage

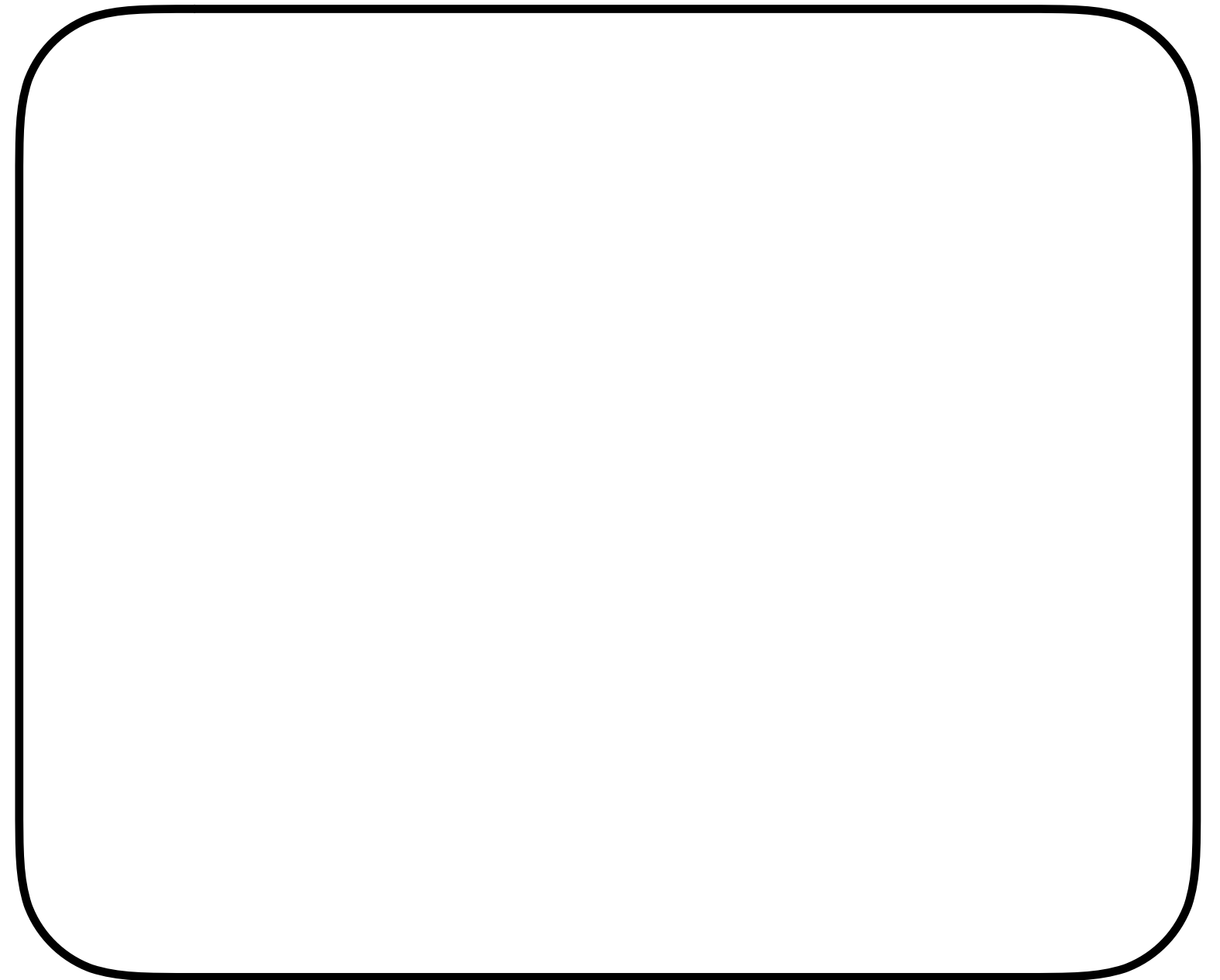
- Persistent Volumes
 - Pod independent storage
 - Administrator defined storage
- Persistent Volume Claims
 - The Pod “claims” the PV
 - Decouples the Pod and the storage
- StorageClass

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

Storage

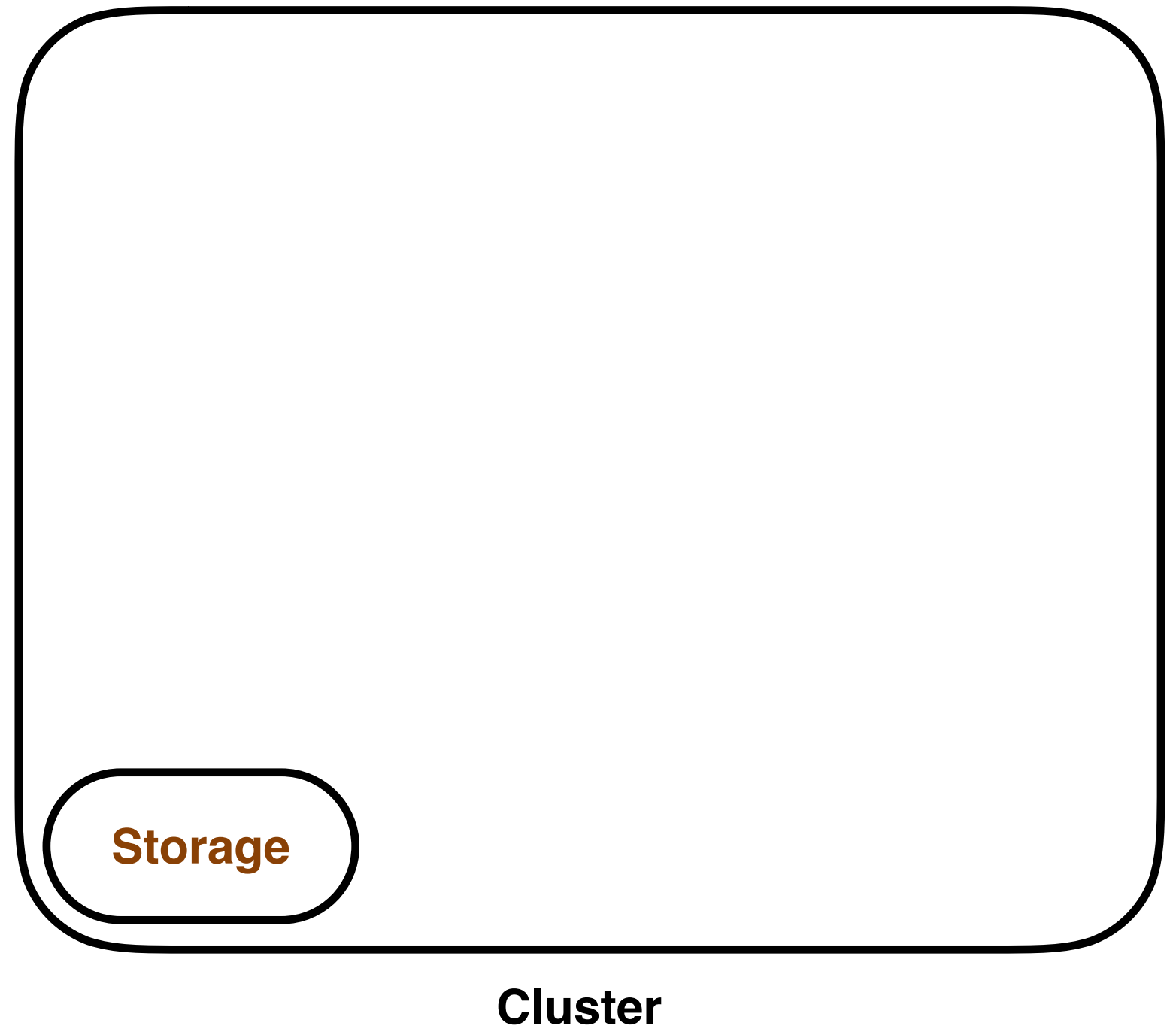
- Persistent Volumes
 - Pod independent storage
 - Administrator defined storage
- Persistent Volume Claims
 - The Pod “claims” the PV
 - Decouples the Pod and the storage
- StorageClass
 - Dynamic Provisioning



Cluster

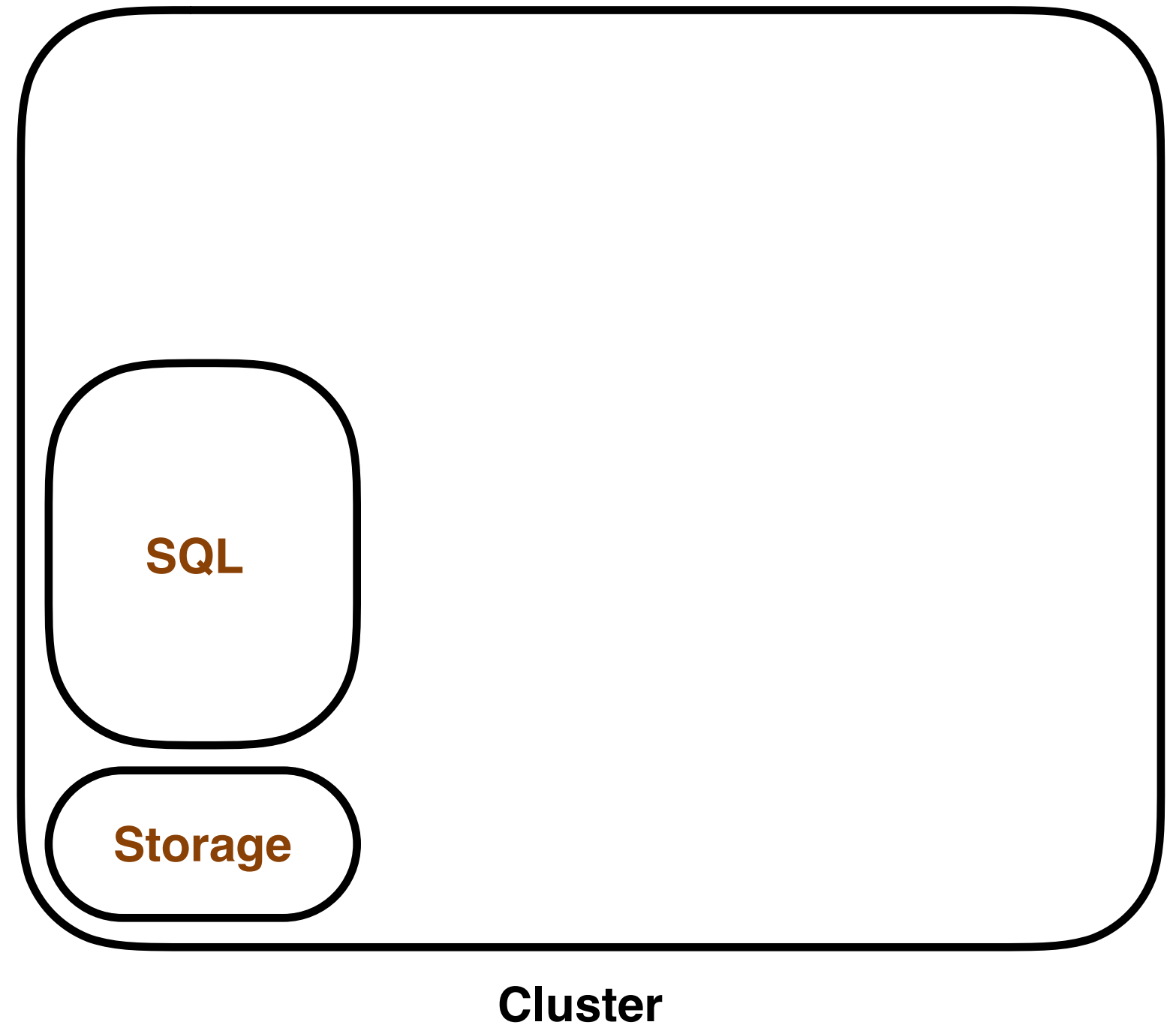
Storage

- 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



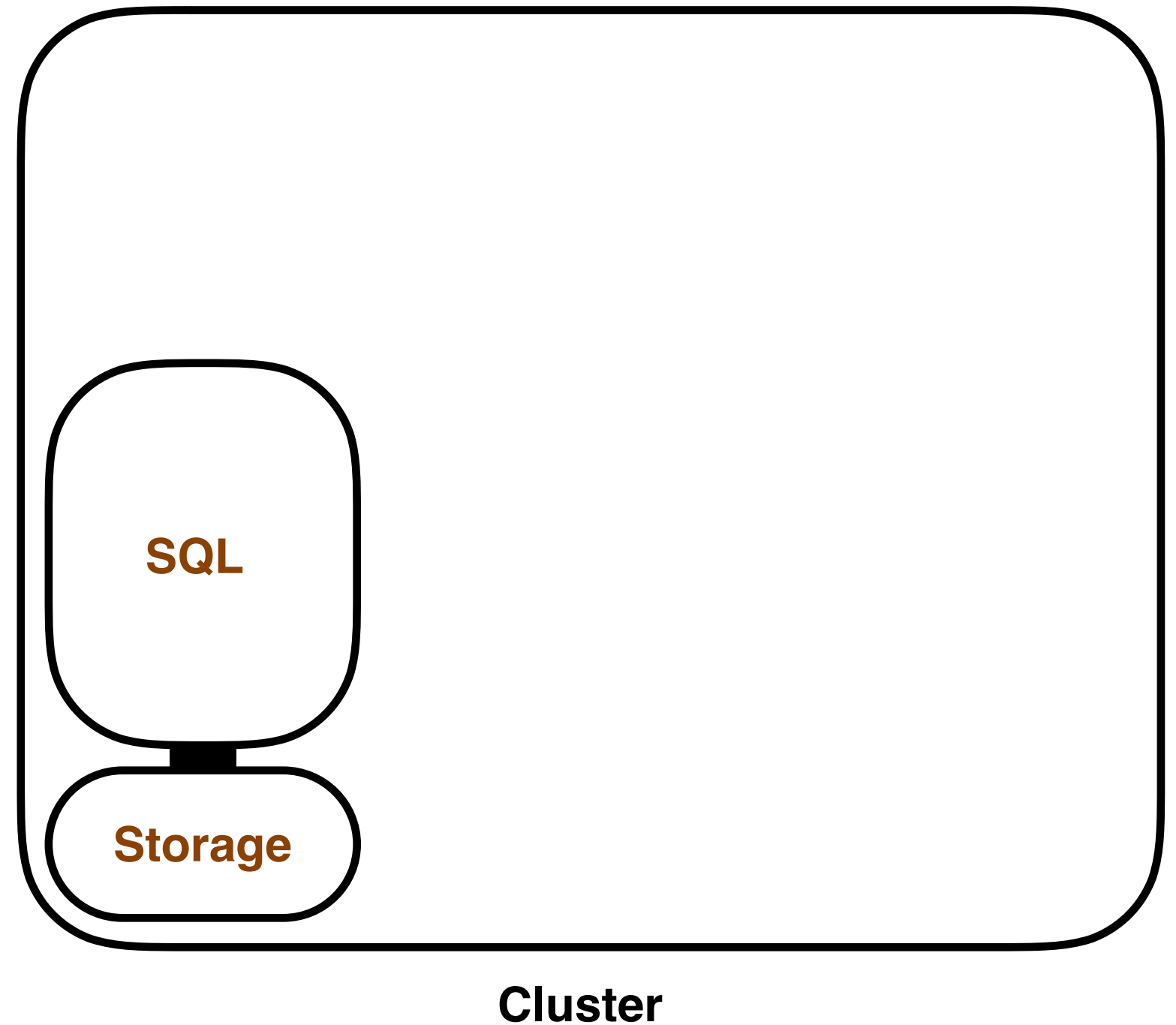
Storage

- 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



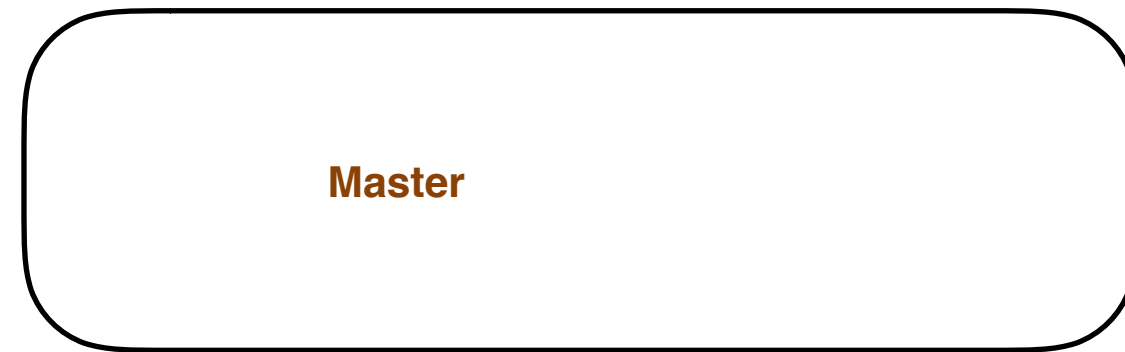
Storage

- 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

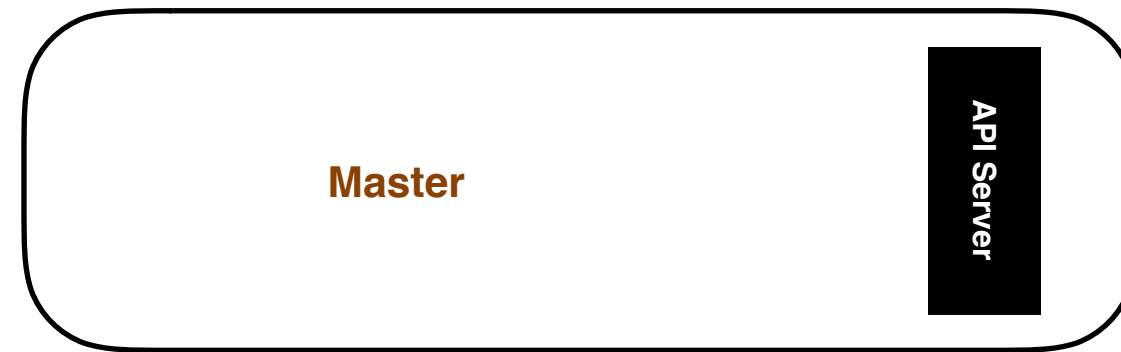


Exploring Kubernetes Architecture

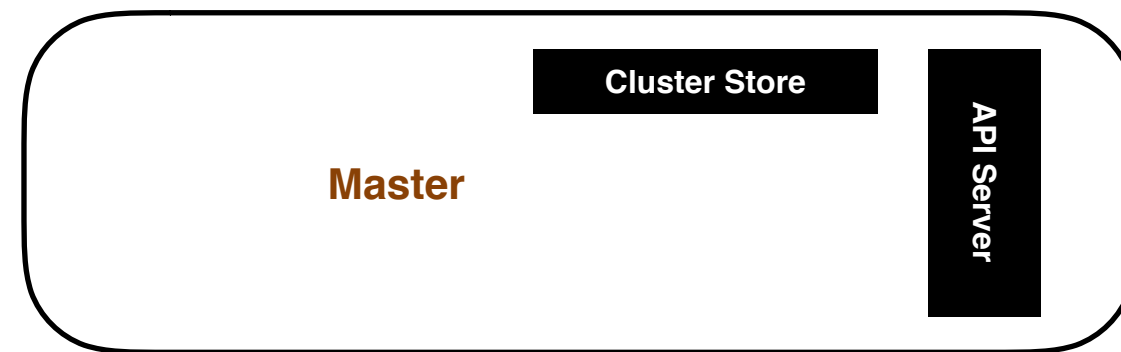
Exploring Kubernetes Architecture



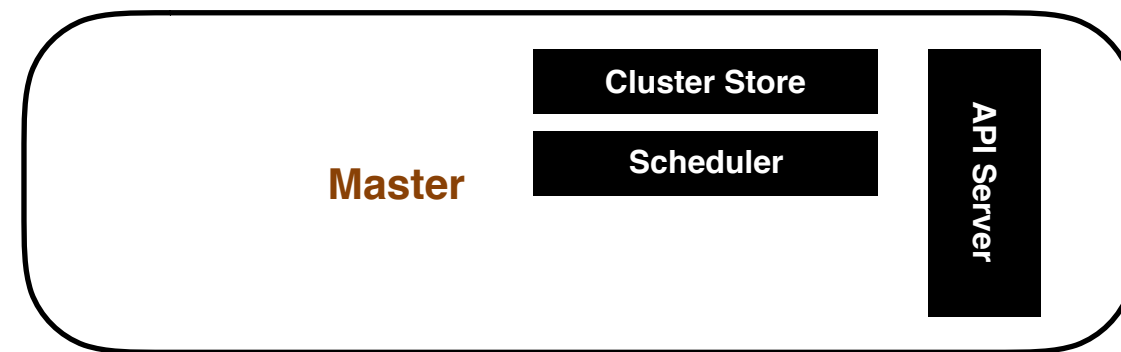
Exploring Kubernetes Architecture



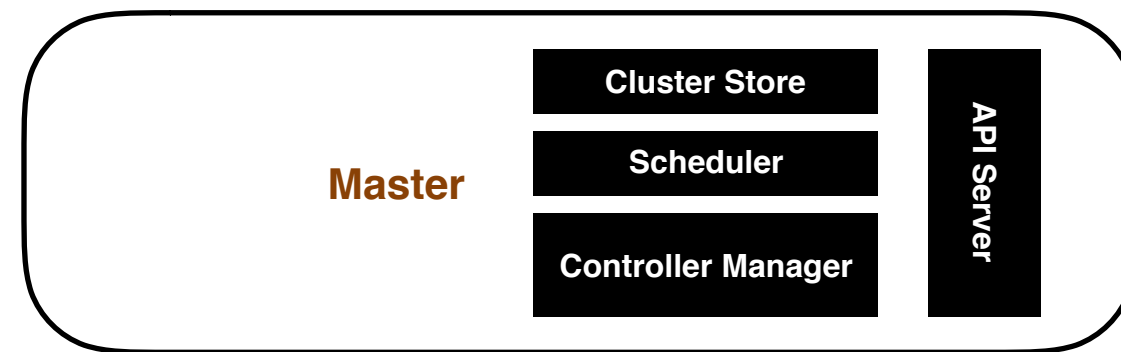
Exploring Kubernetes Architecture



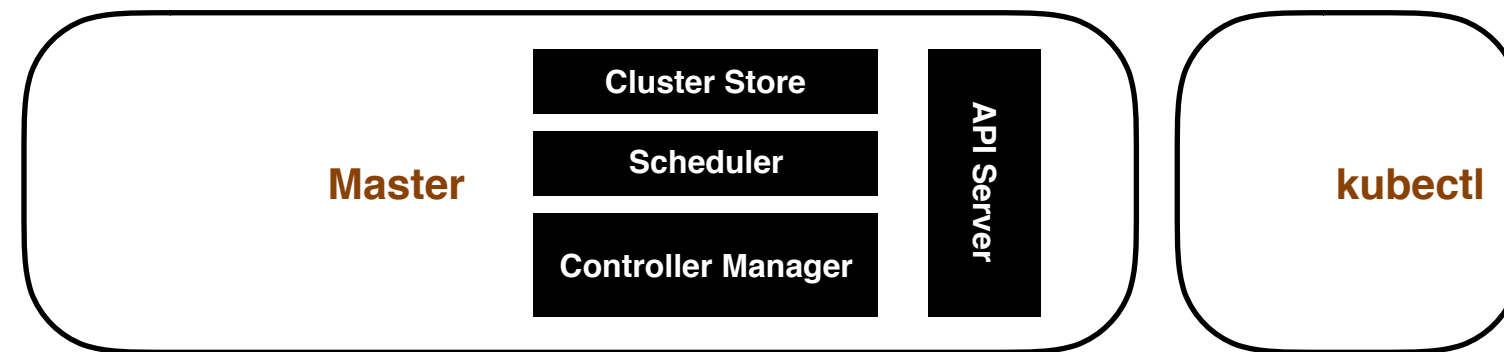
Exploring Kubernetes Architecture



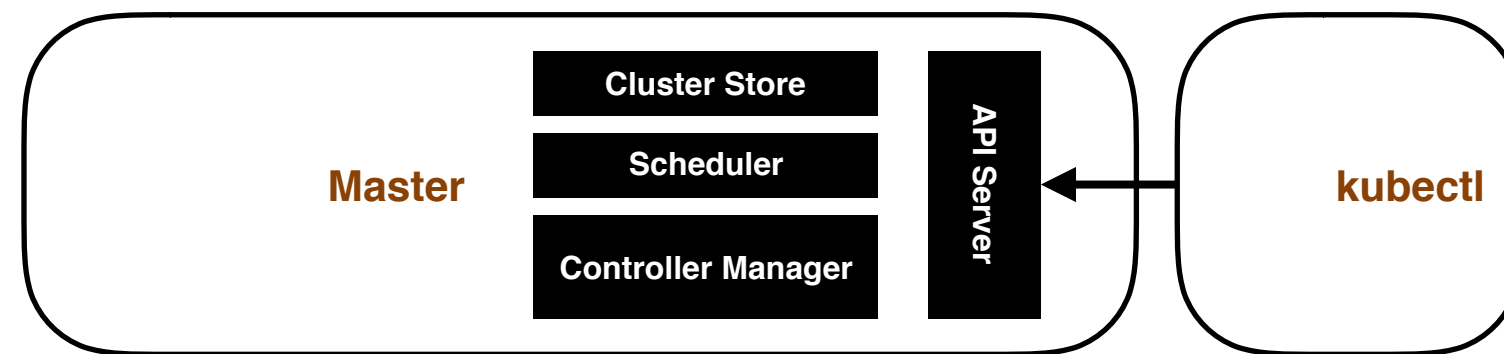
Exploring Kubernetes Architecture



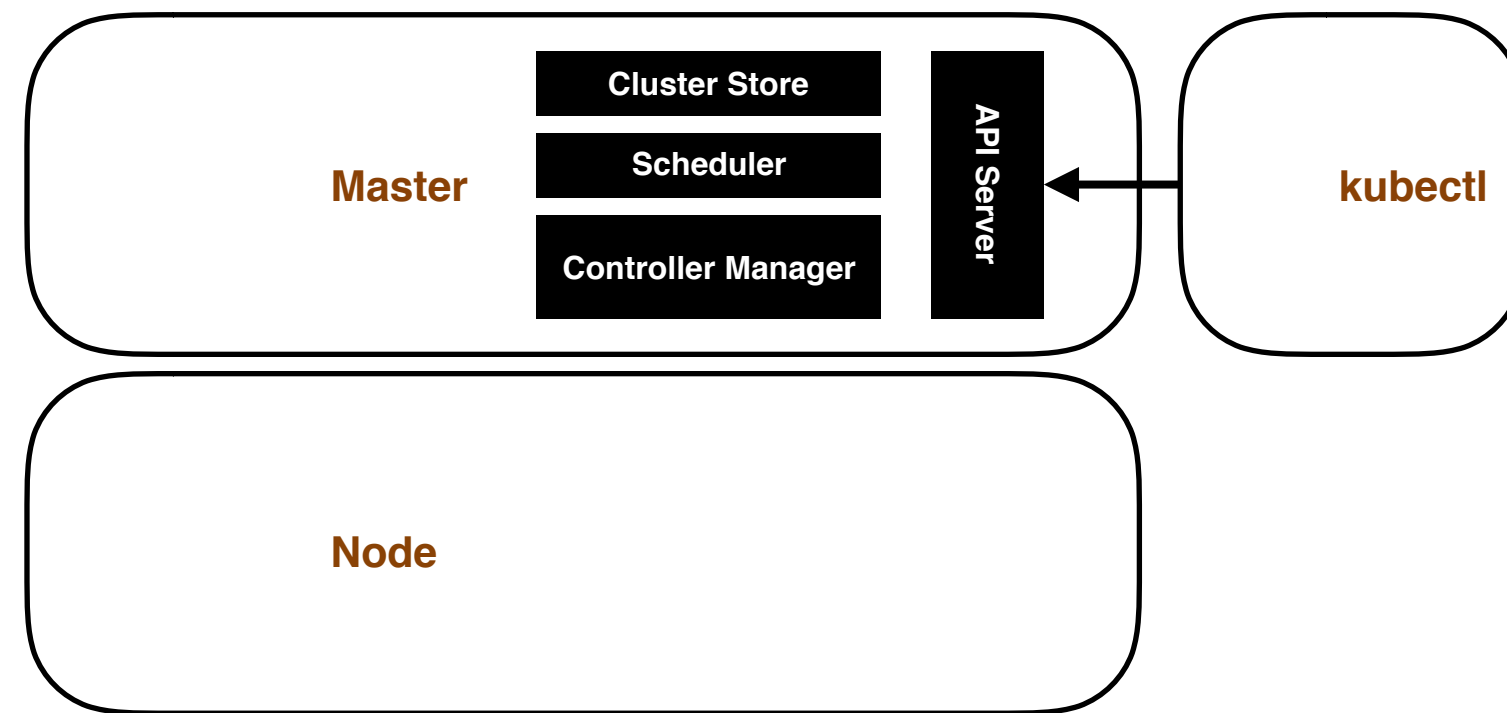
Exploring Kubernetes Architecture



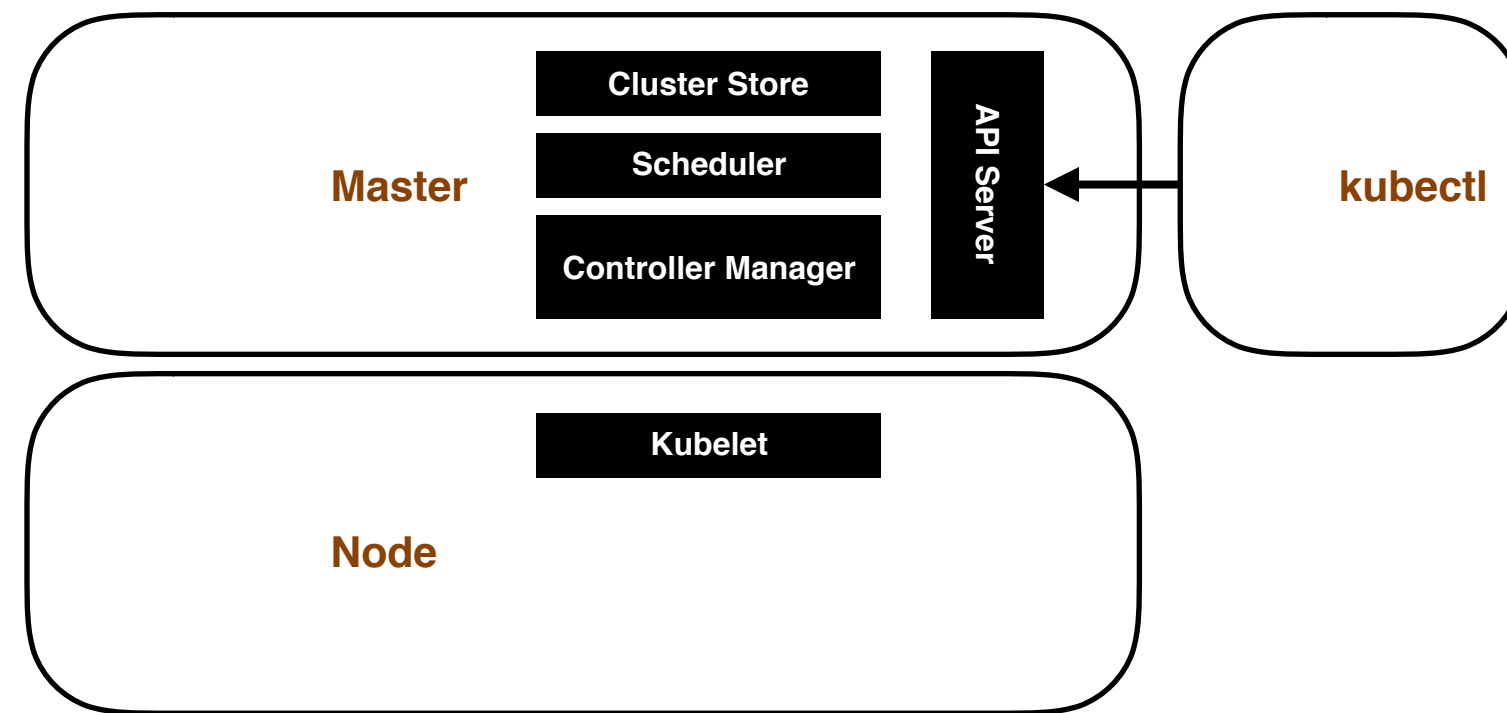
Exploring Kubernetes Architecture



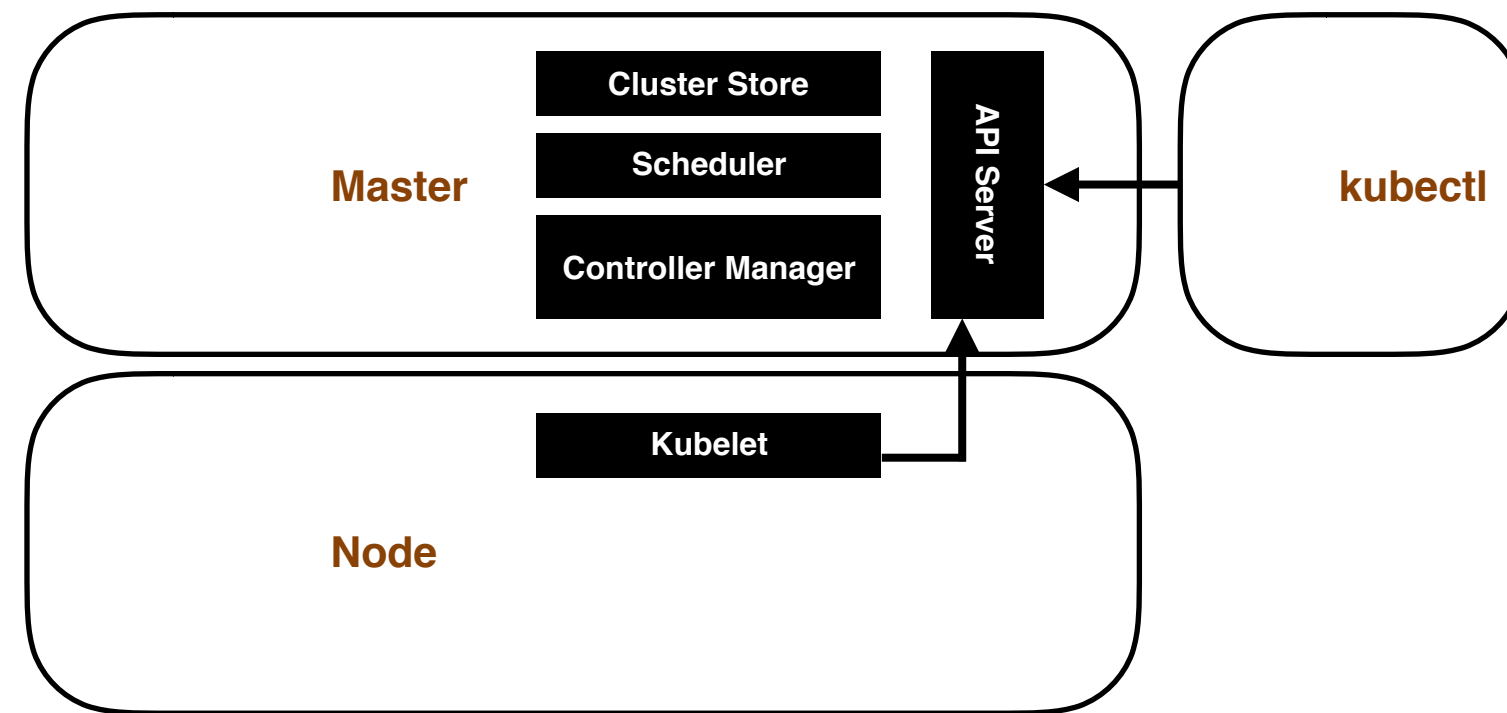
Exploring Kubernetes Architecture



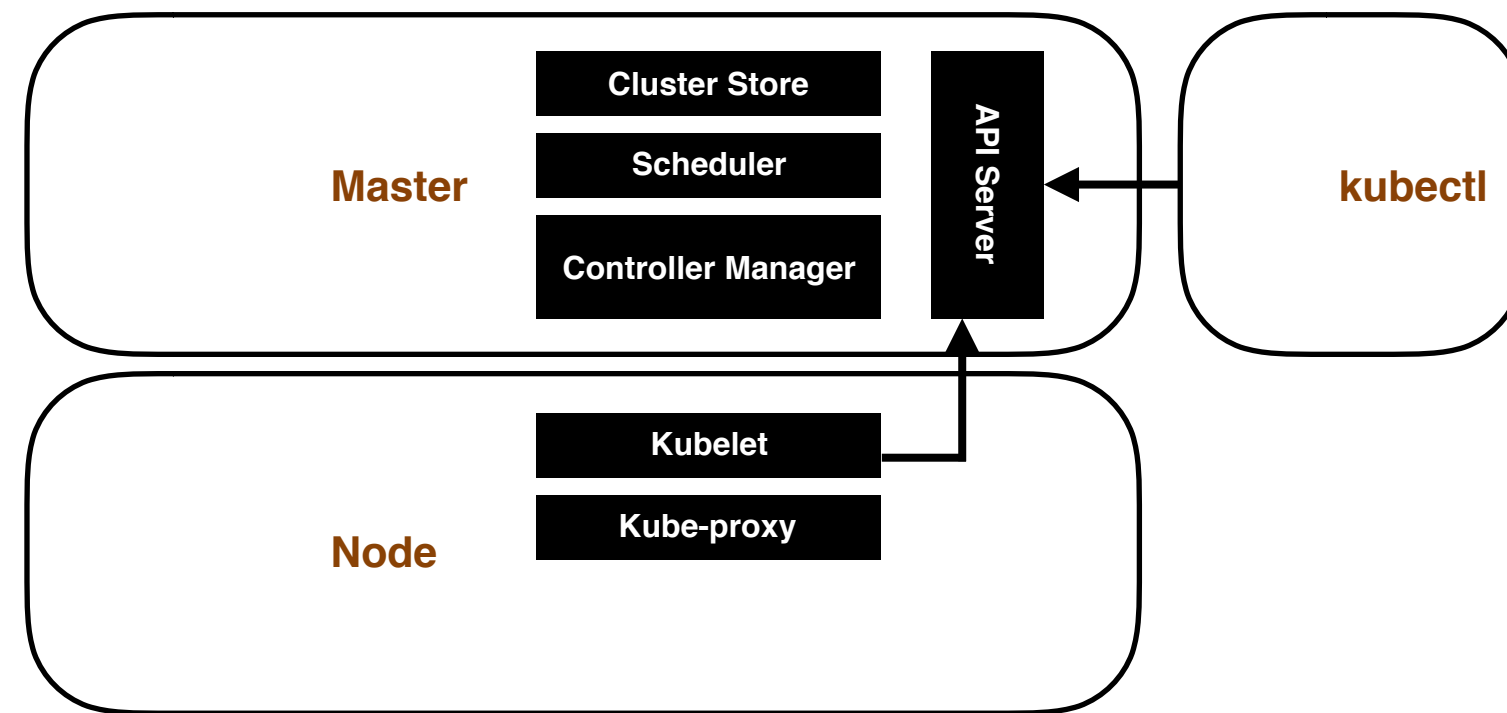
Exploring Kubernetes Architecture



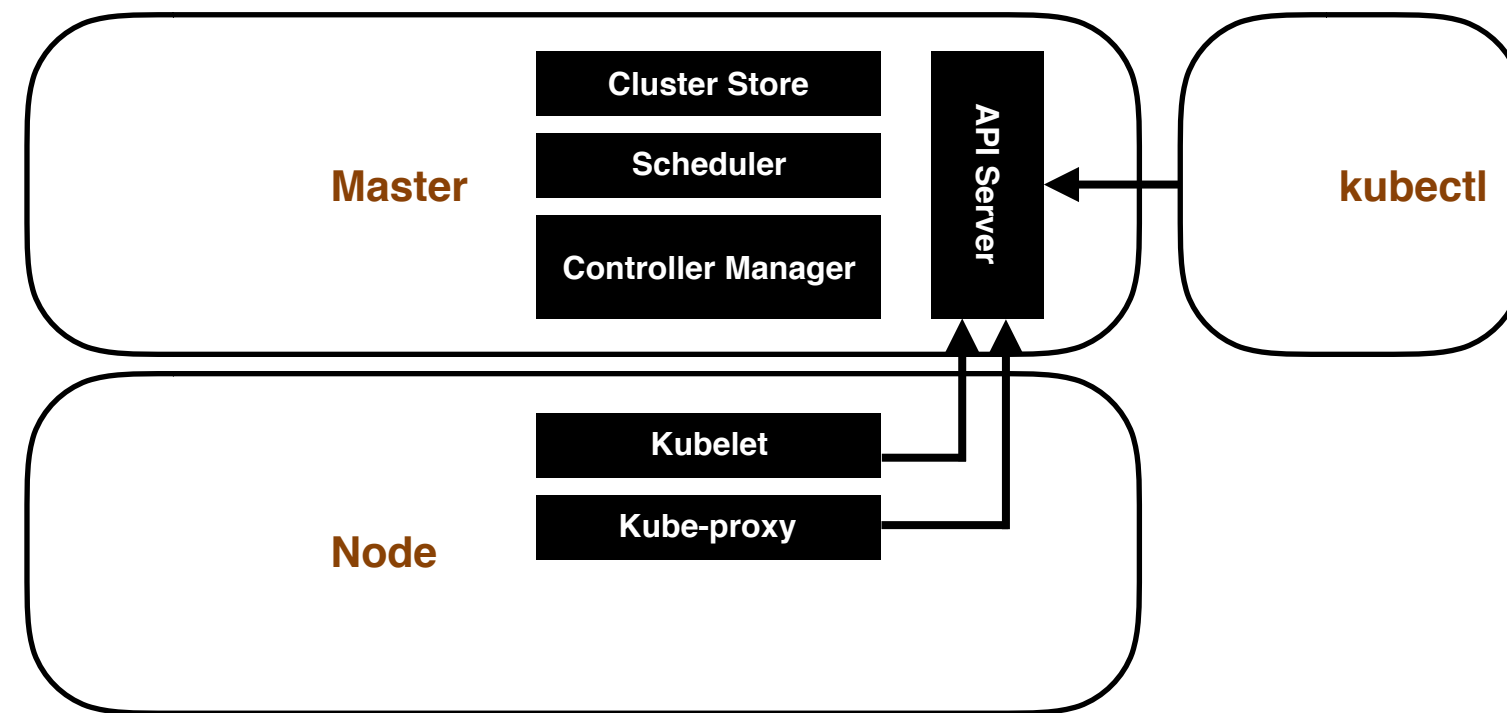
Exploring Kubernetes Architecture



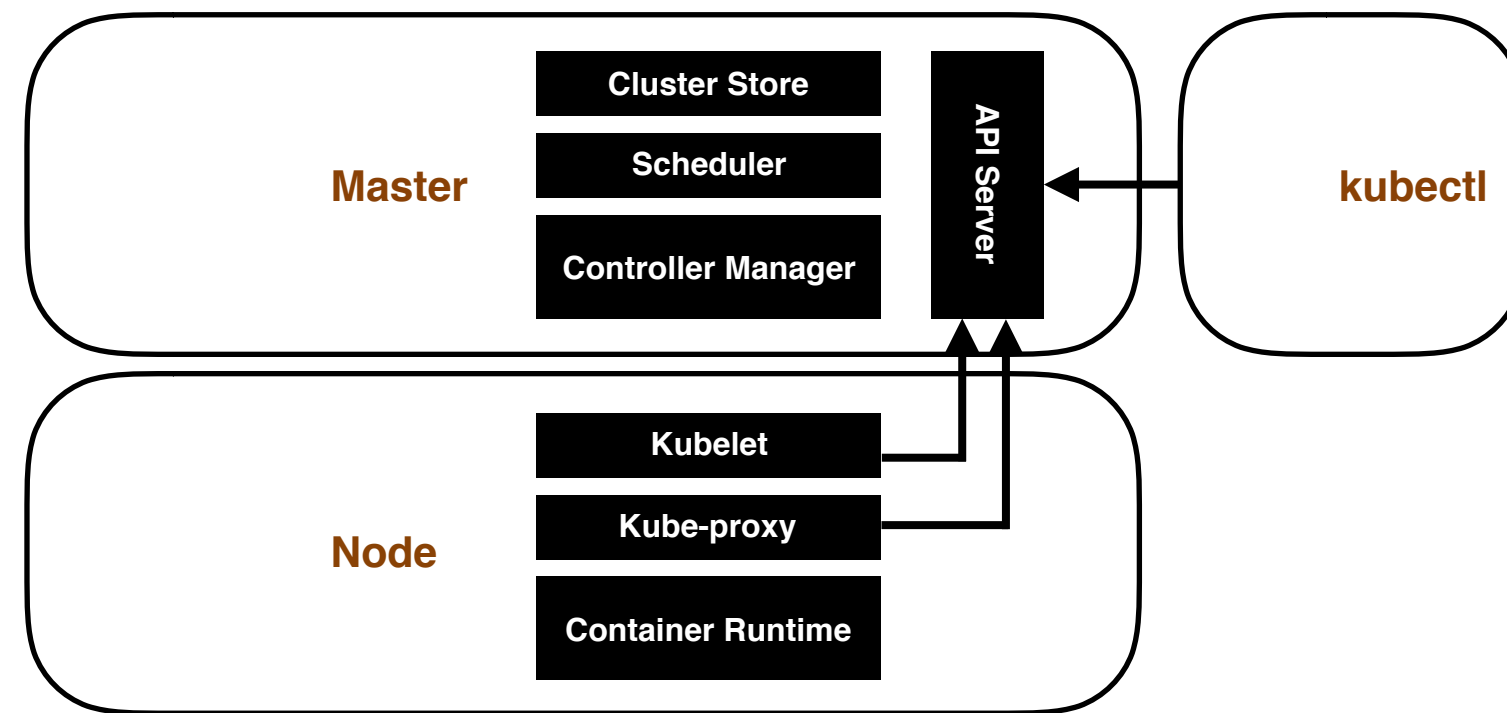
Exploring Kubernetes Architecture



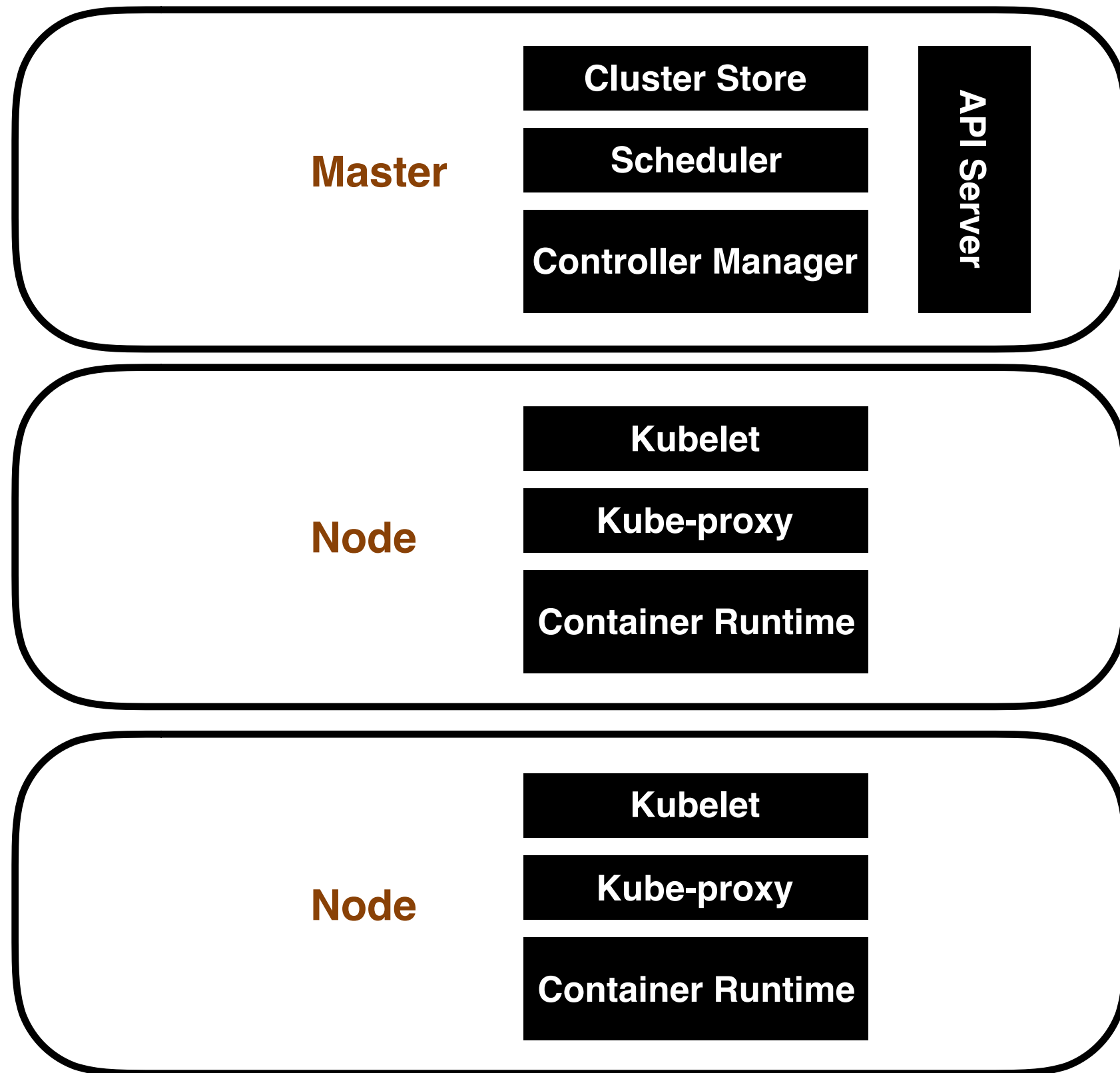
Exploring Kubernetes Architecture



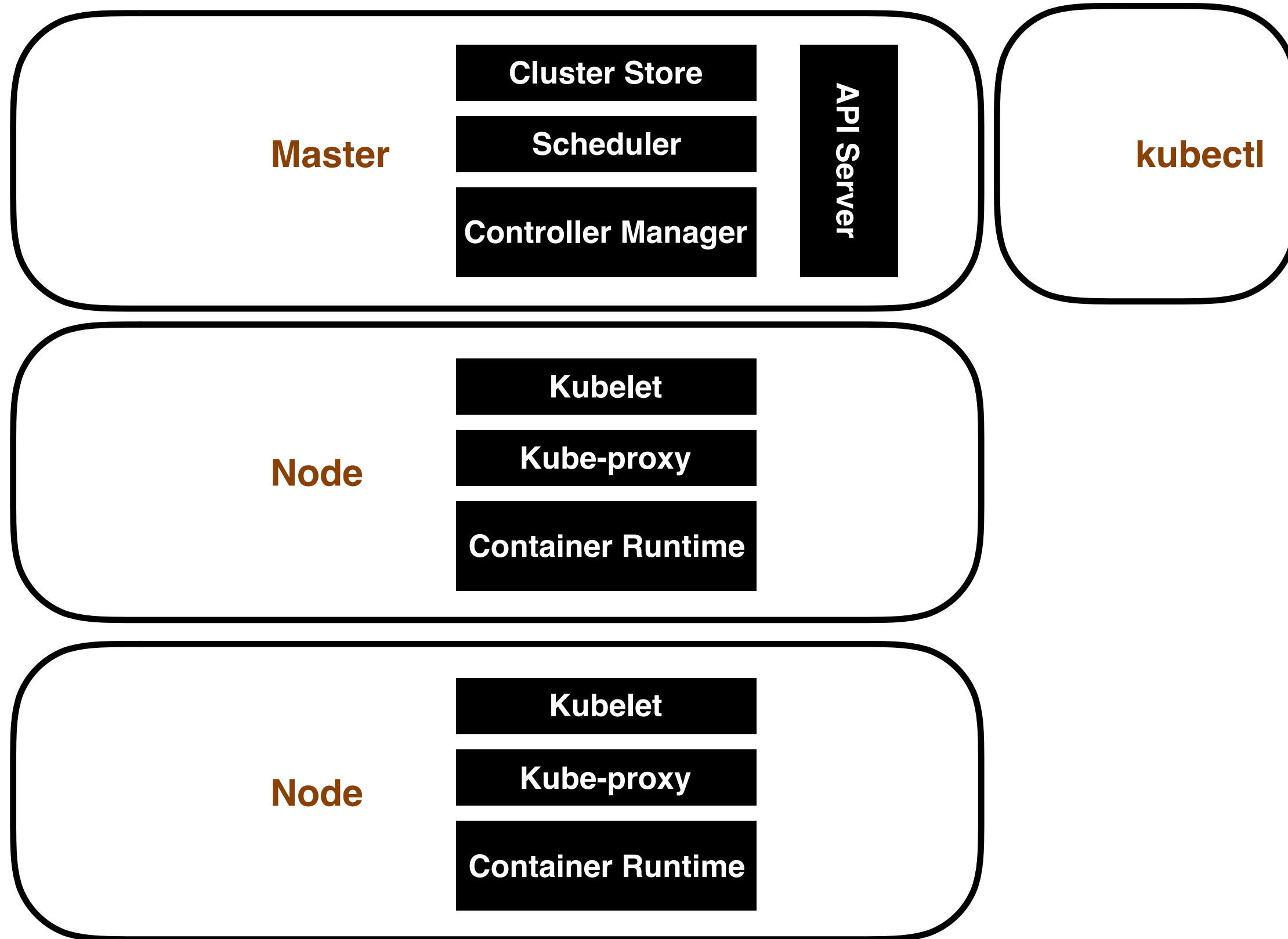
Exploring Kubernetes Architecture



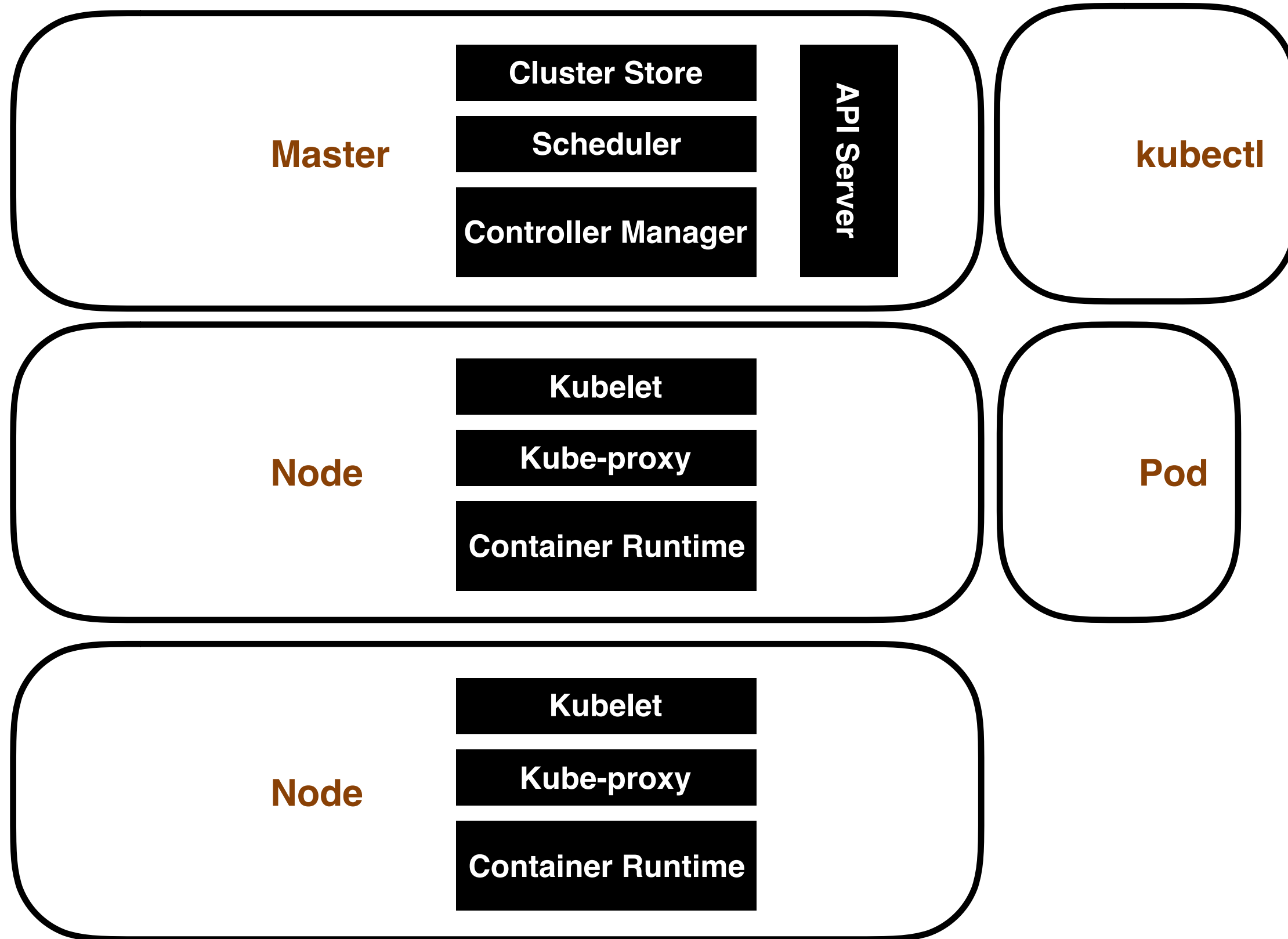
Controller Operations - ReplicaSet



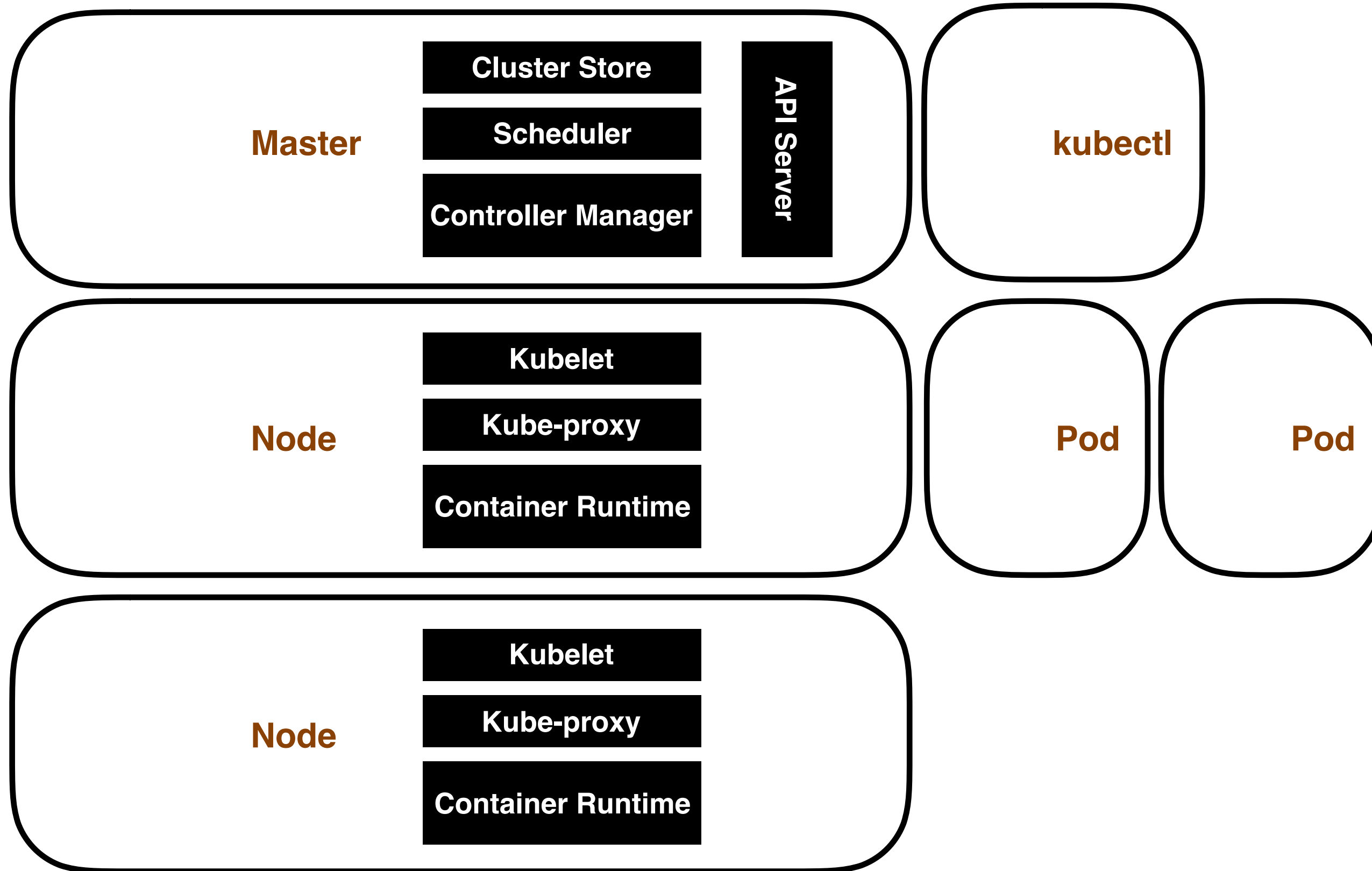
Controller Operations - ReplicaSet



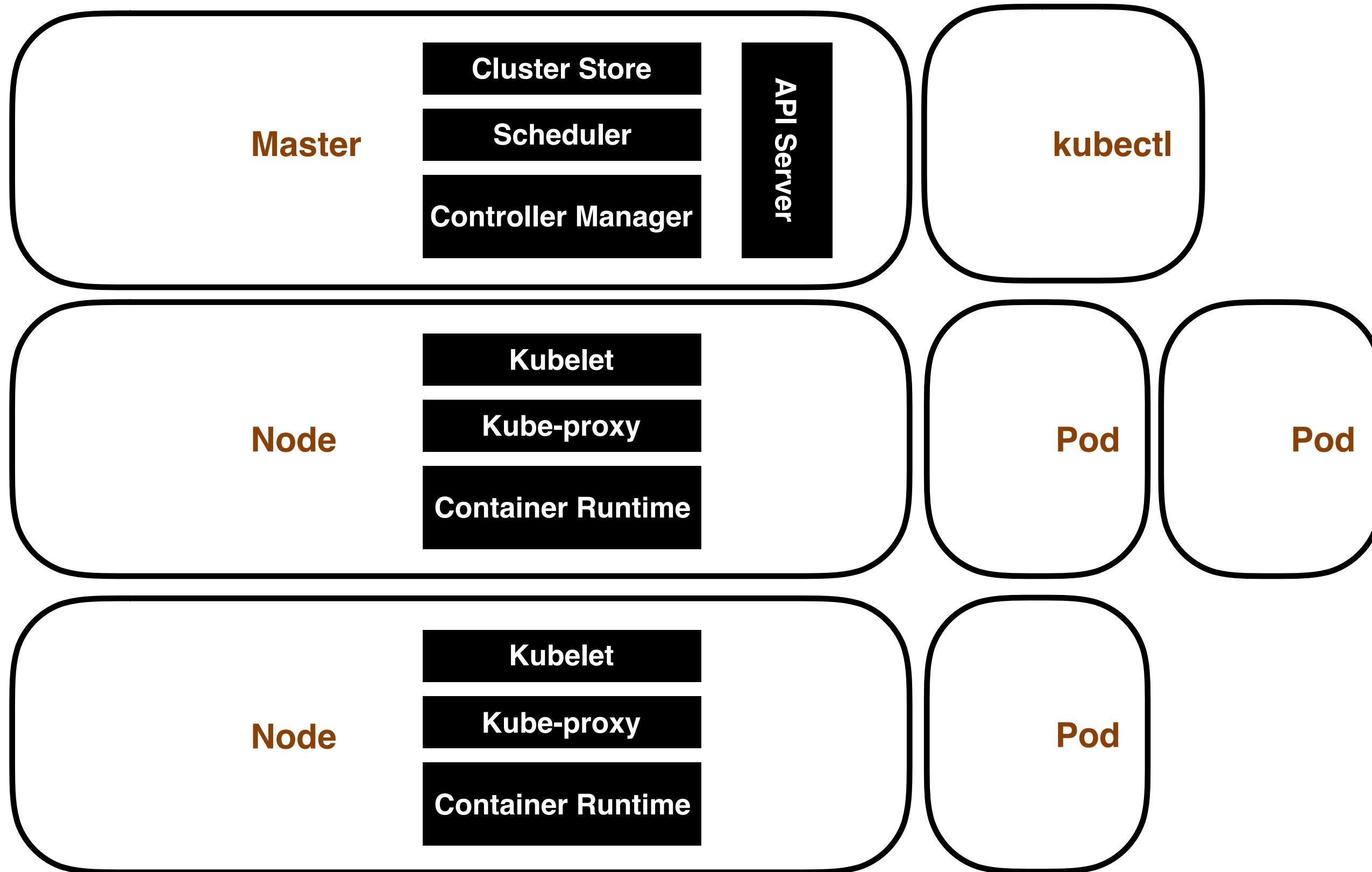
Controller Operations - ReplicaSet



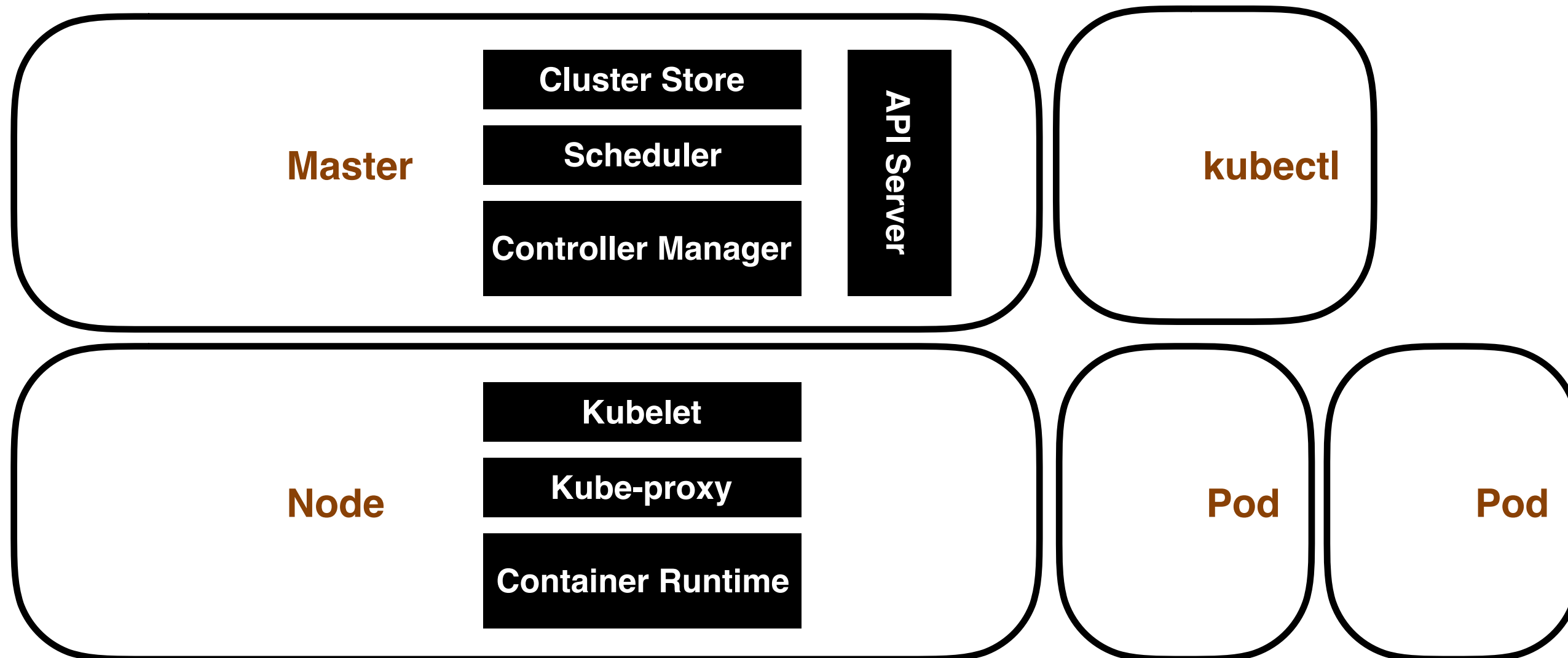
Controller Operations - ReplicaSet



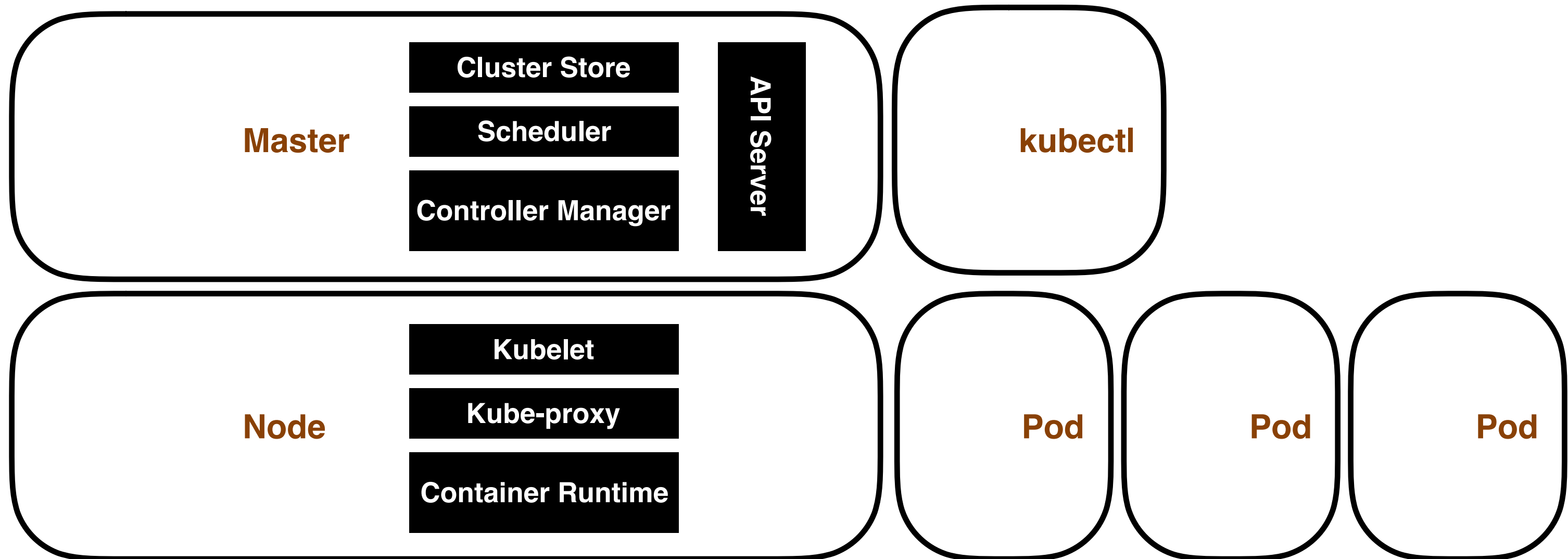
Controller Operations - ReplicaSet



Controller Operations - ReplicaSet

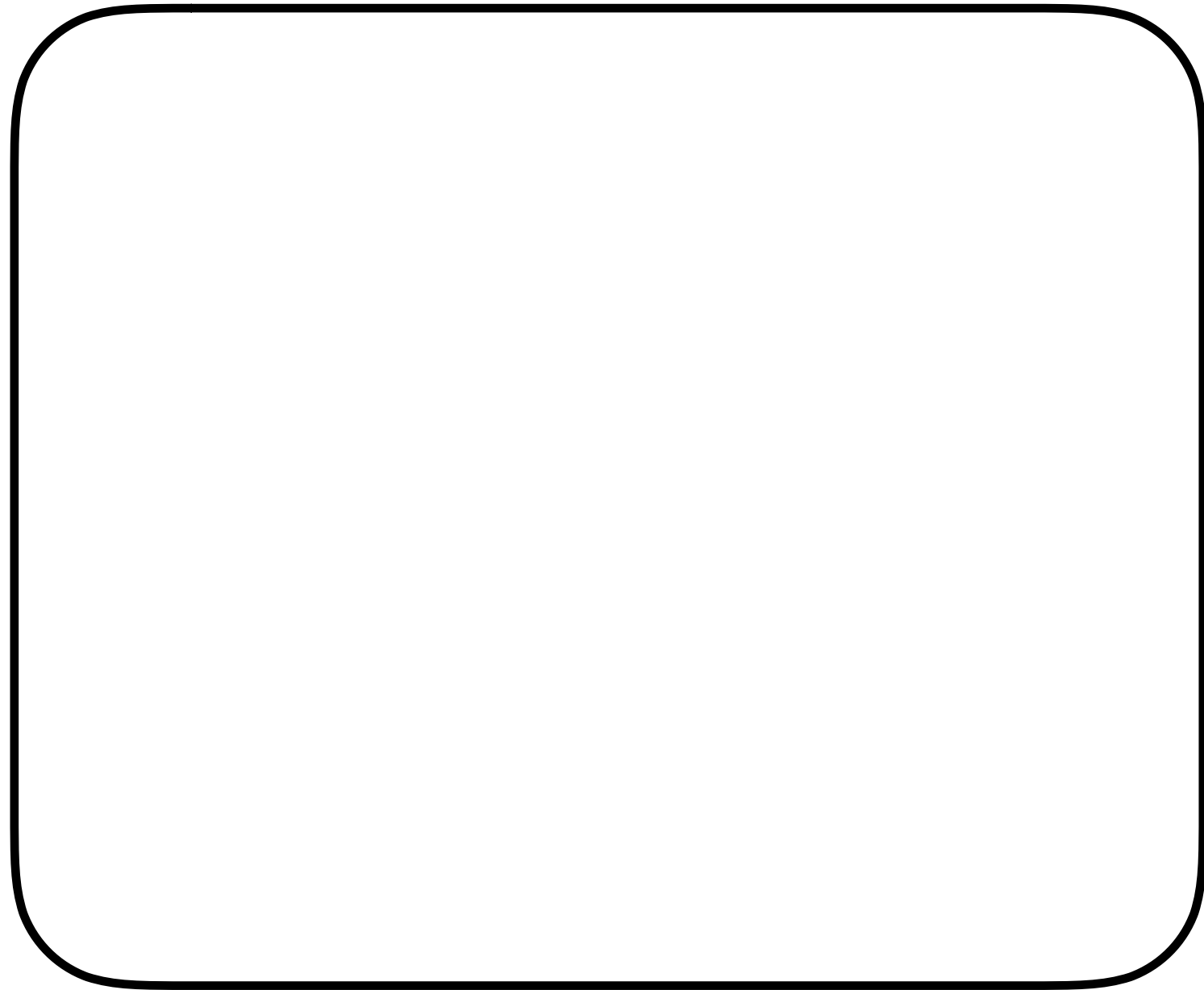


Controller Operations - ReplicaSet



Services

Services



Cluster

Services

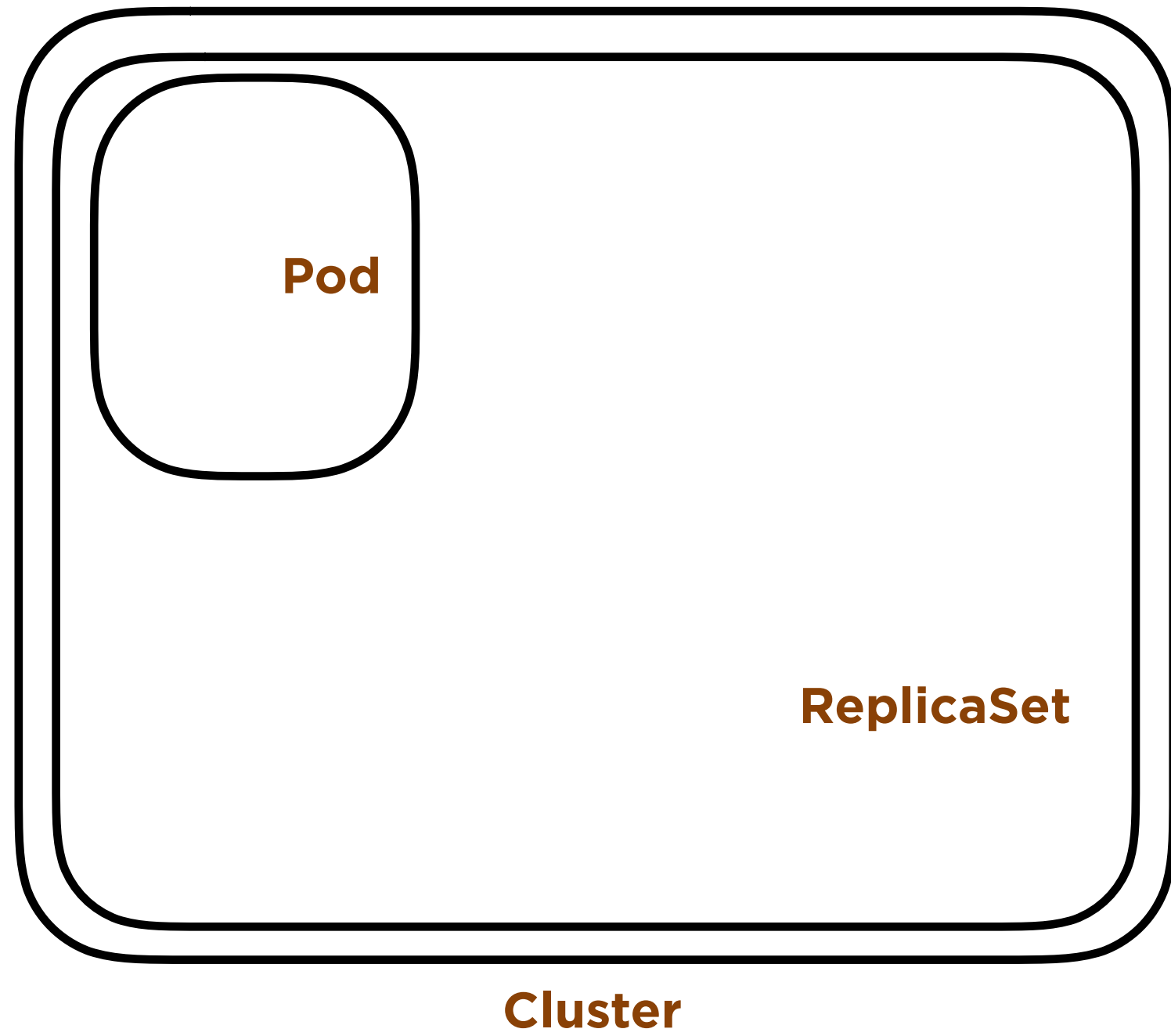


The diagram illustrates a database architecture. It features a large, double-bordered rounded rectangle on the left side of the slide. Inside this rectangle, the text 'ReplicaSet' is written in a brown font. Below the rectangle, the word 'Cluster' is written in the same brown font. The word 'Services' is positioned at the top center of the slide in a large, black, sans-serif font.

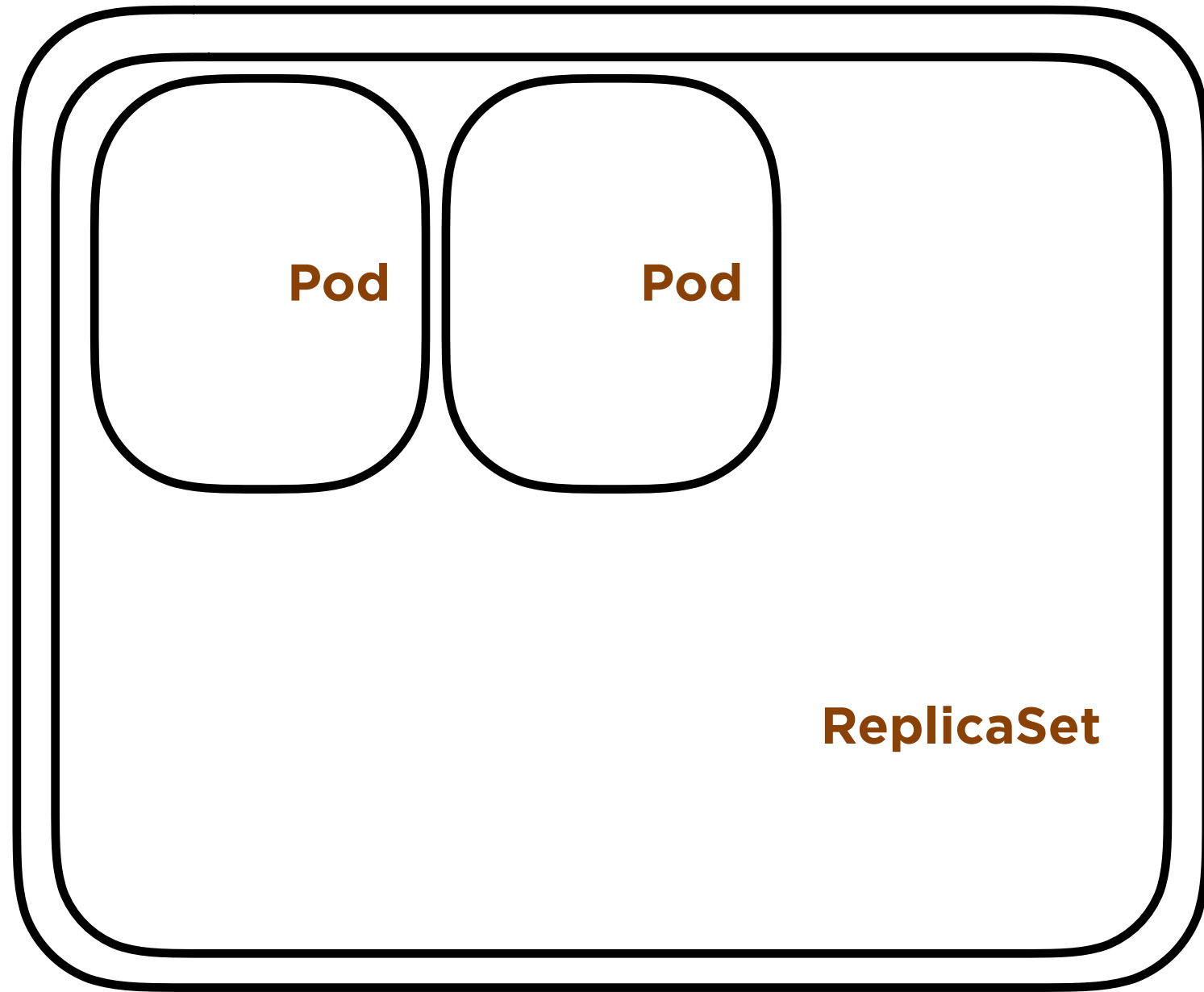
ReplicaSet

Cluster

Services

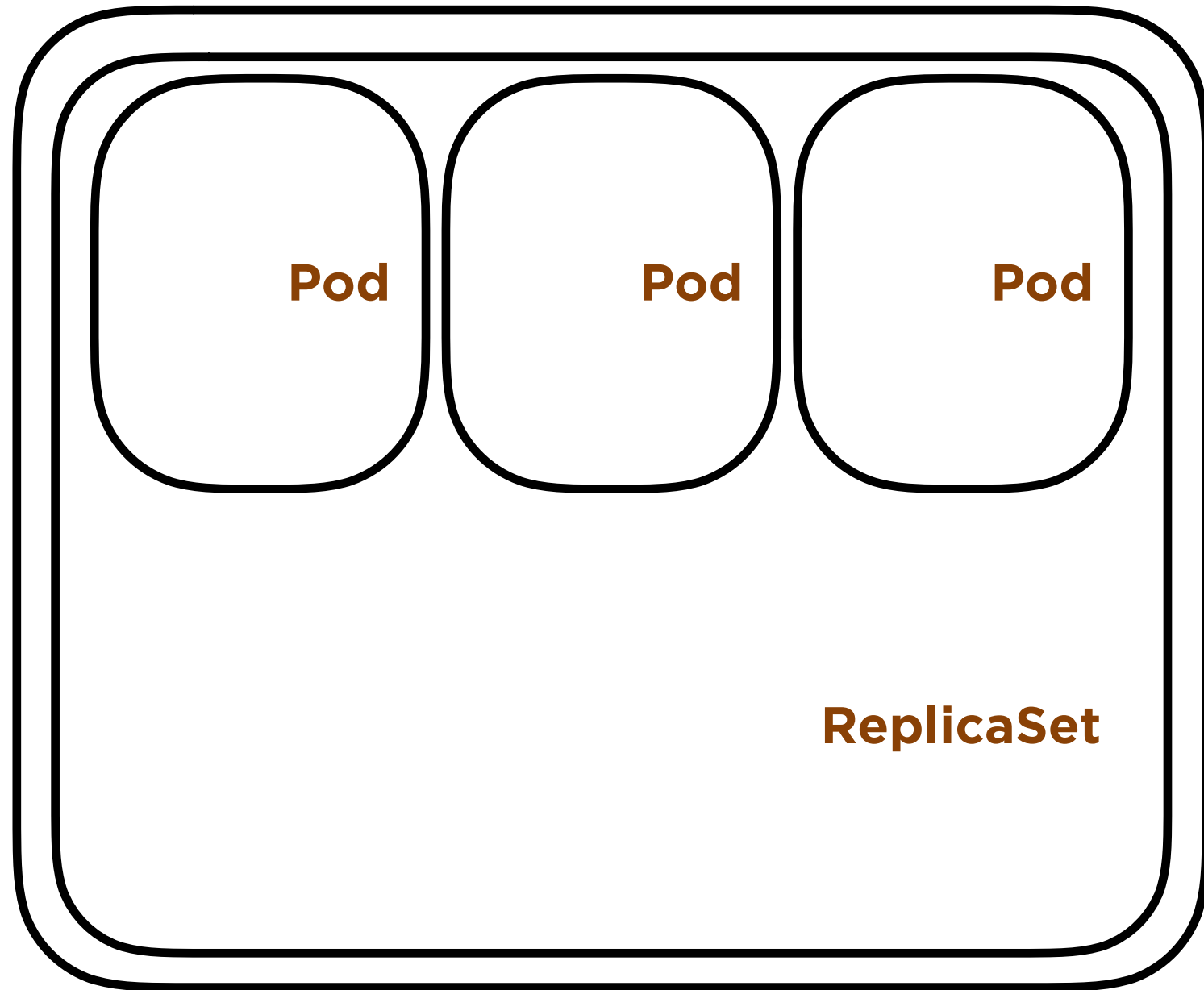


Services



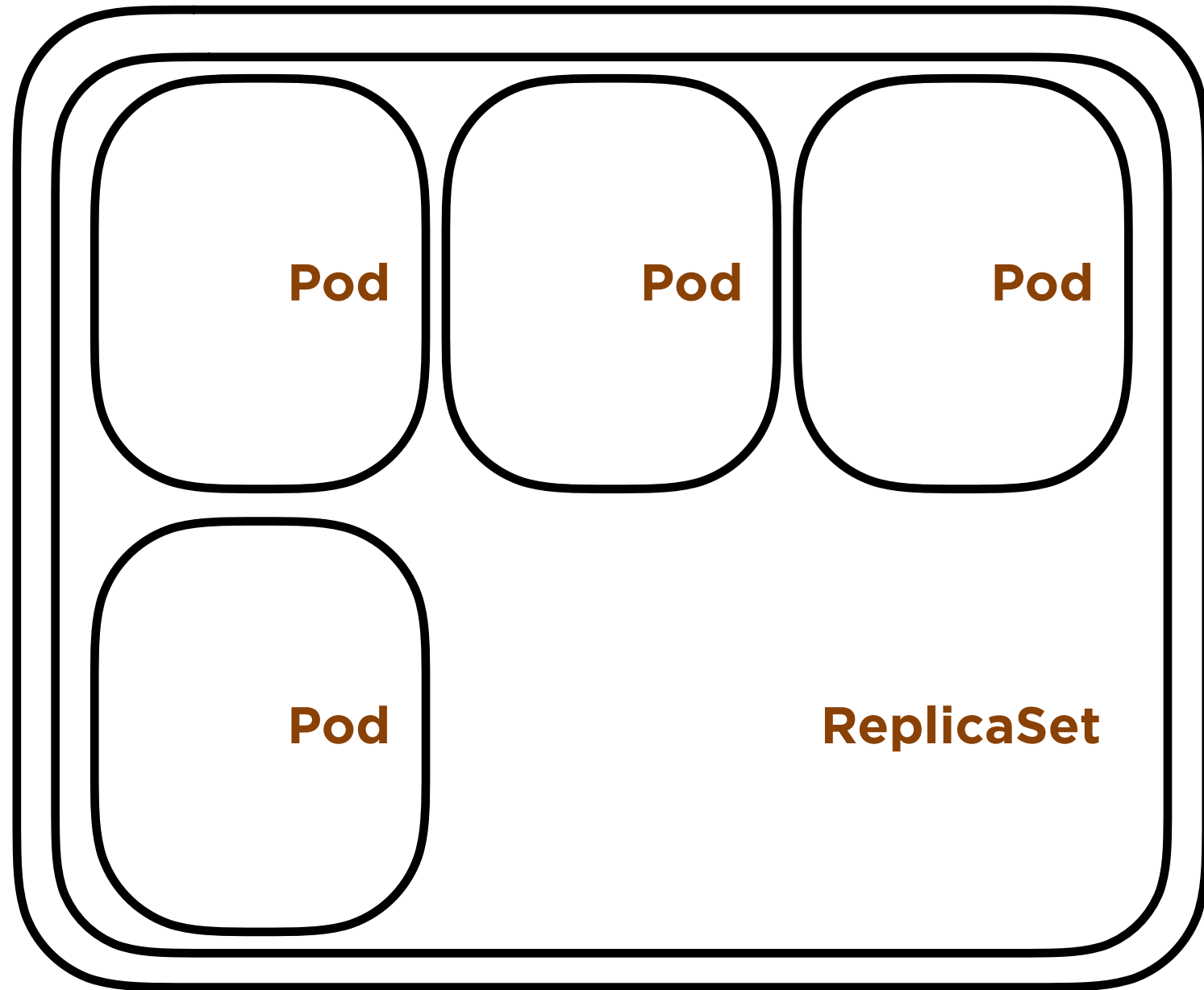
Cluster

Services



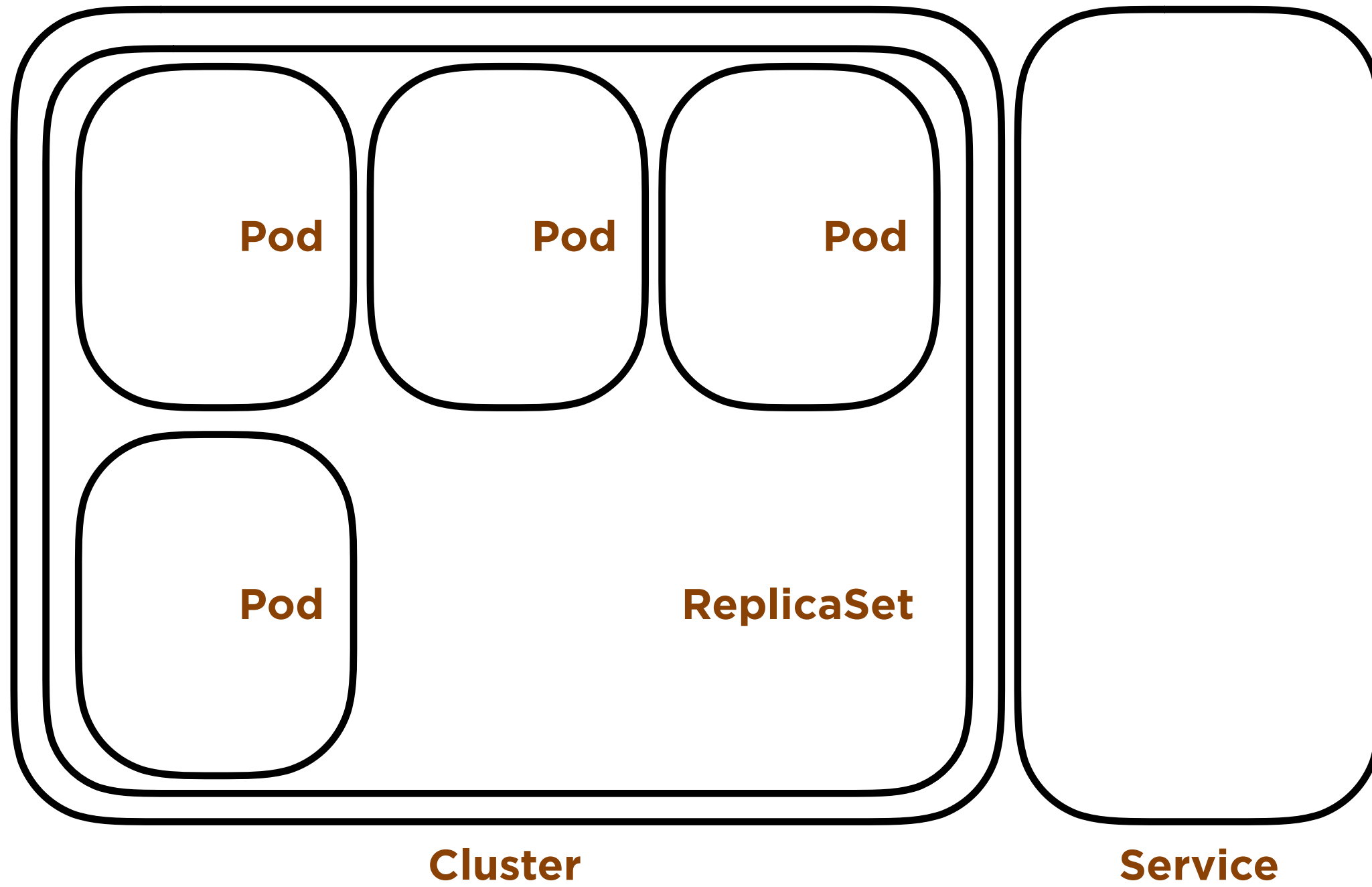
Cluster

Services

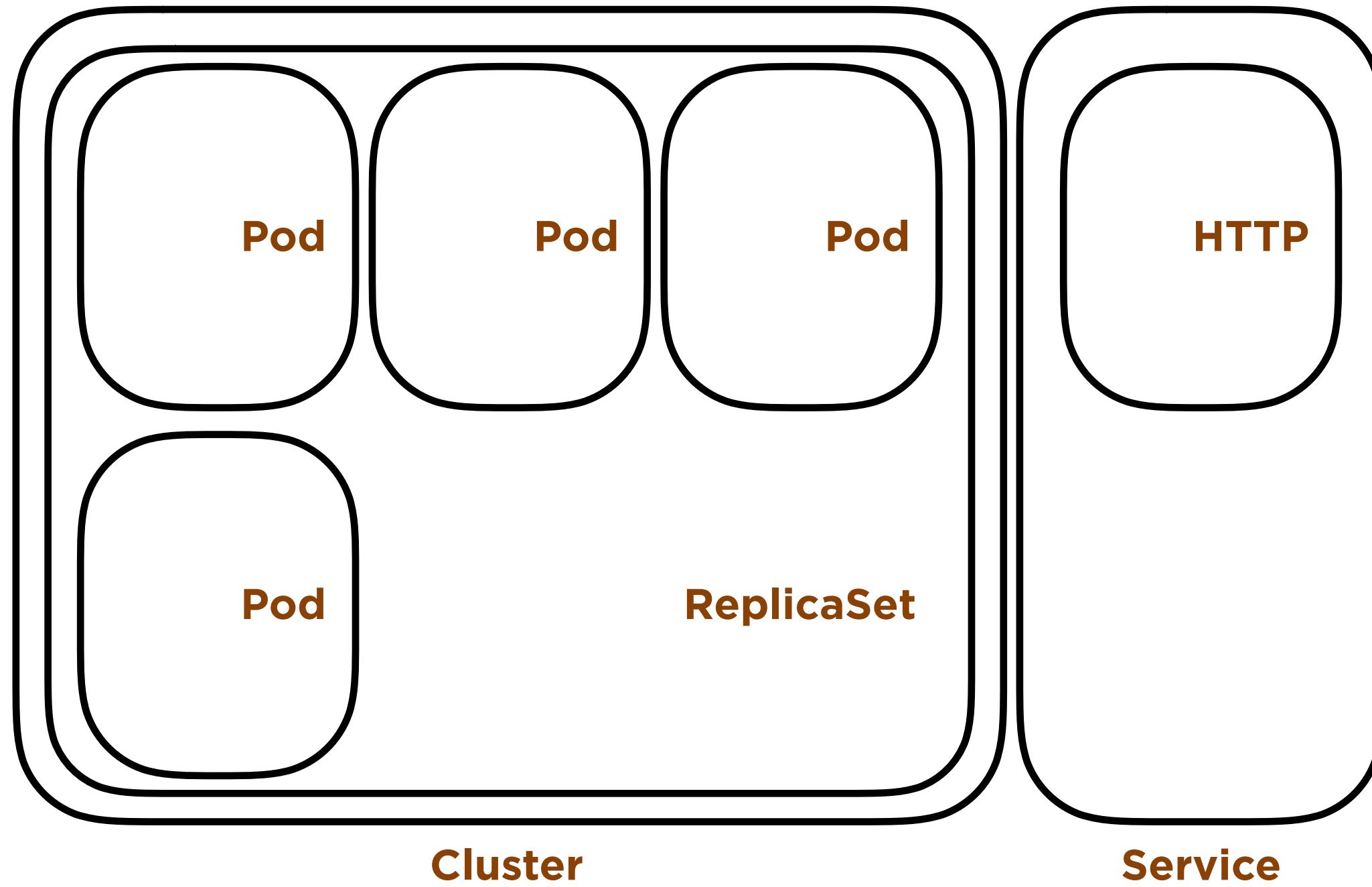


Cluster

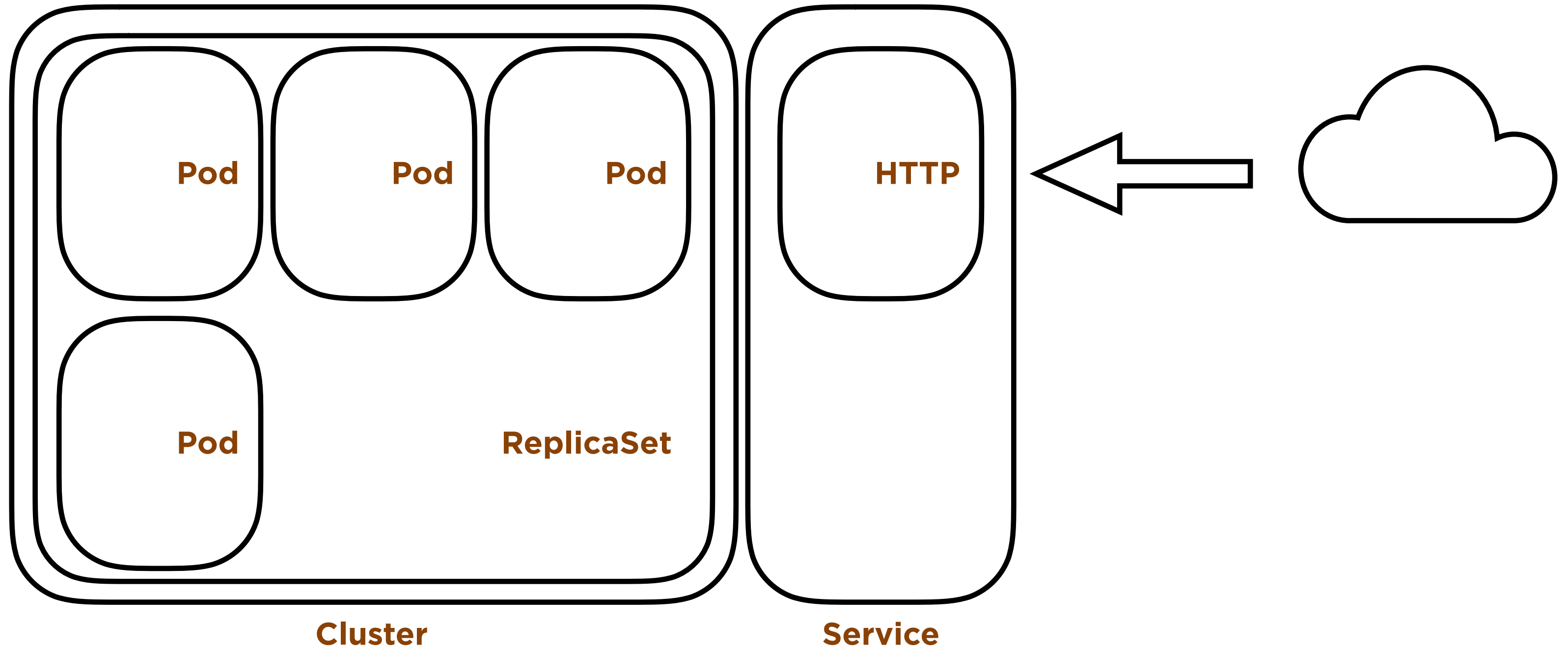
Services



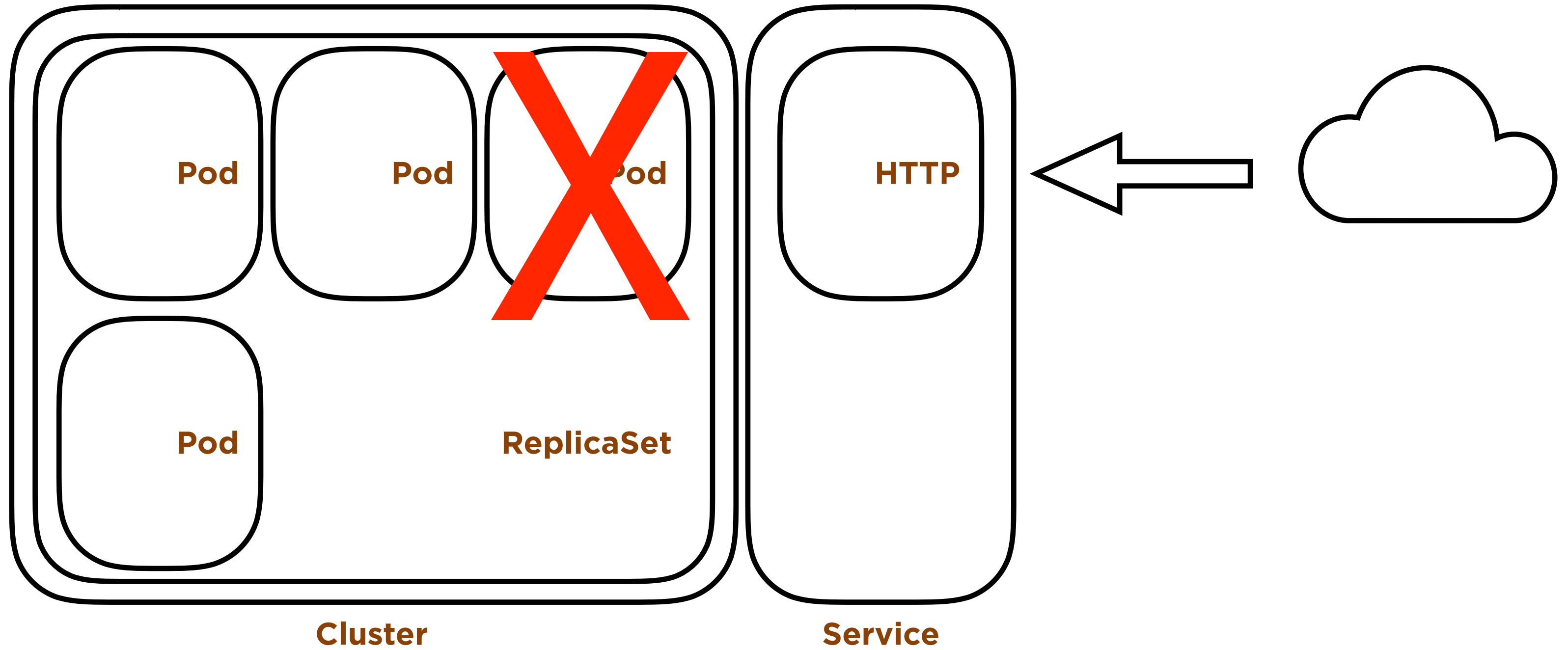
Services



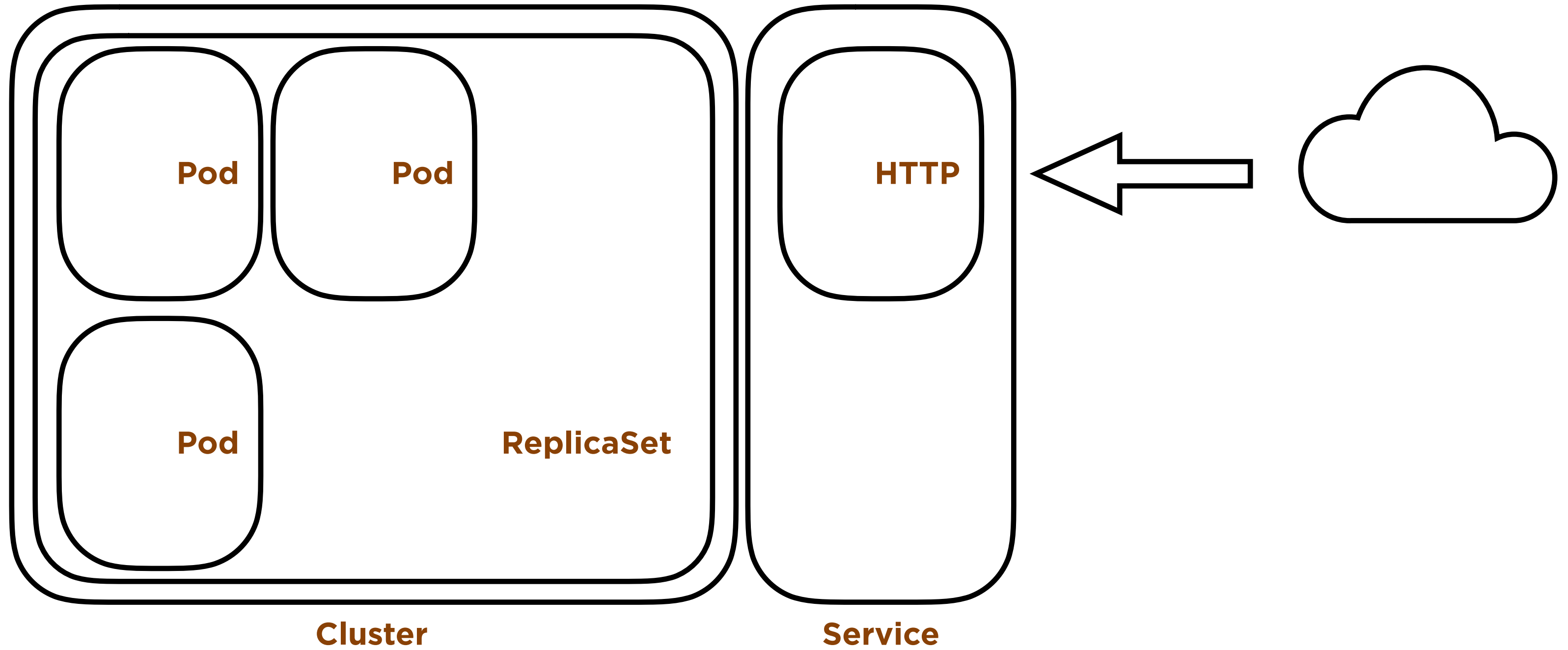
Services



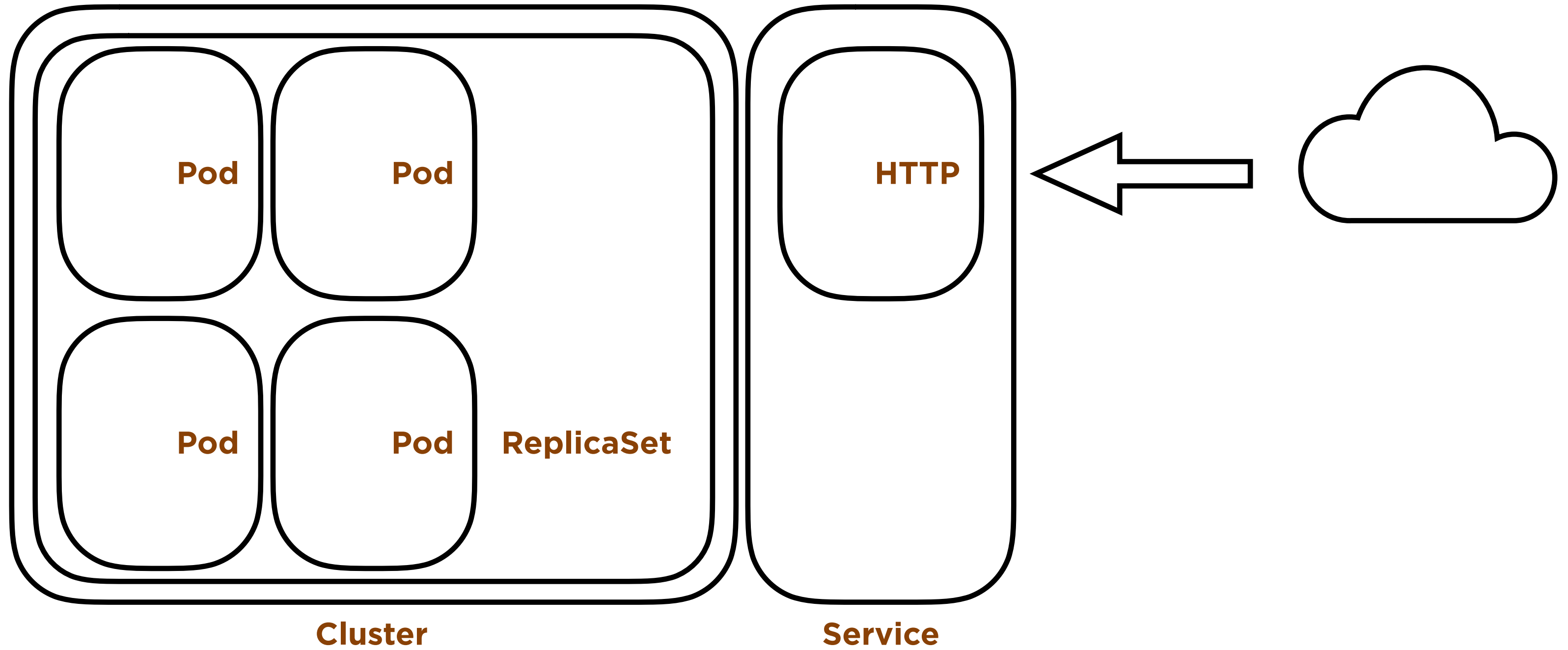
Services



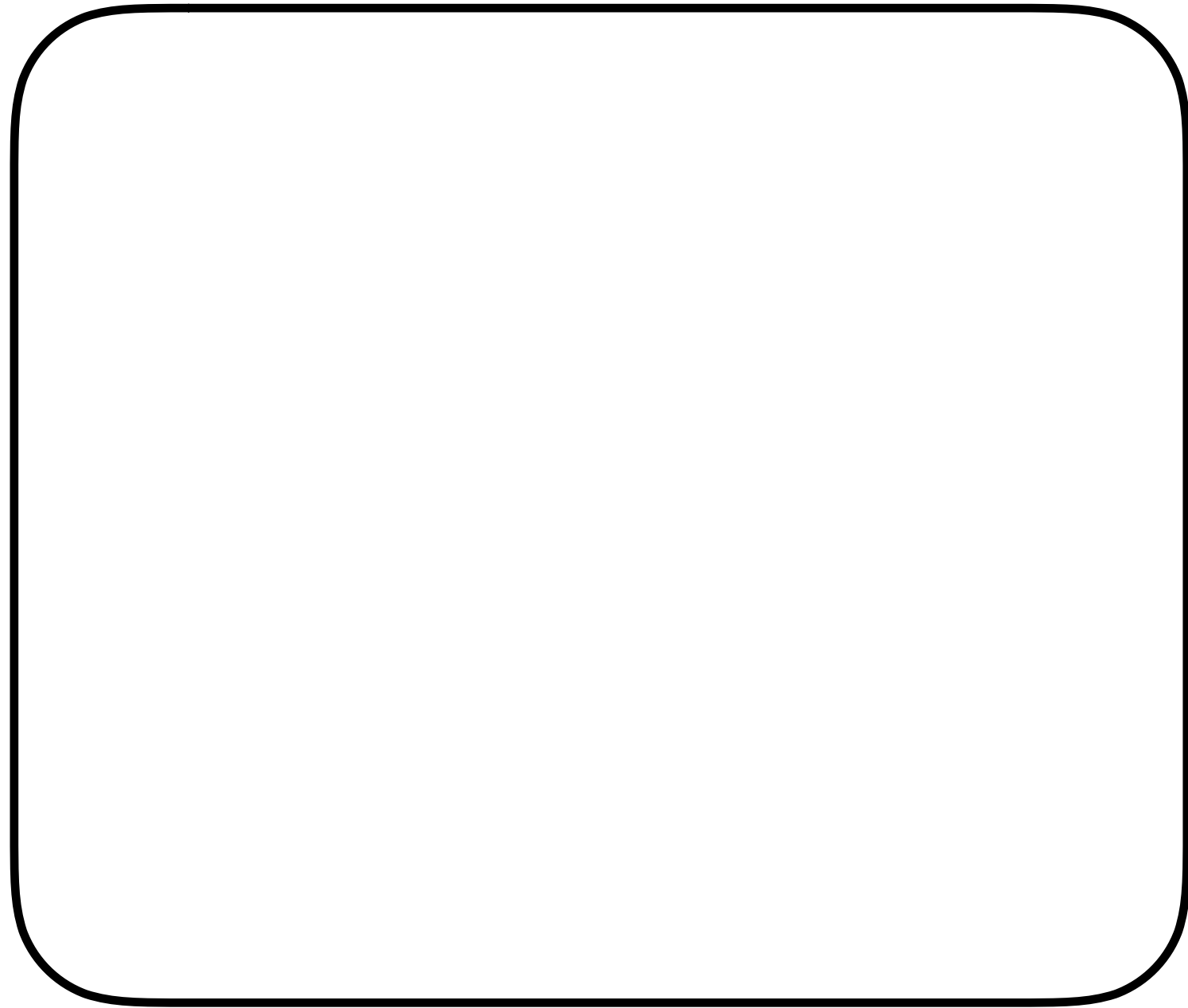
Services



Services

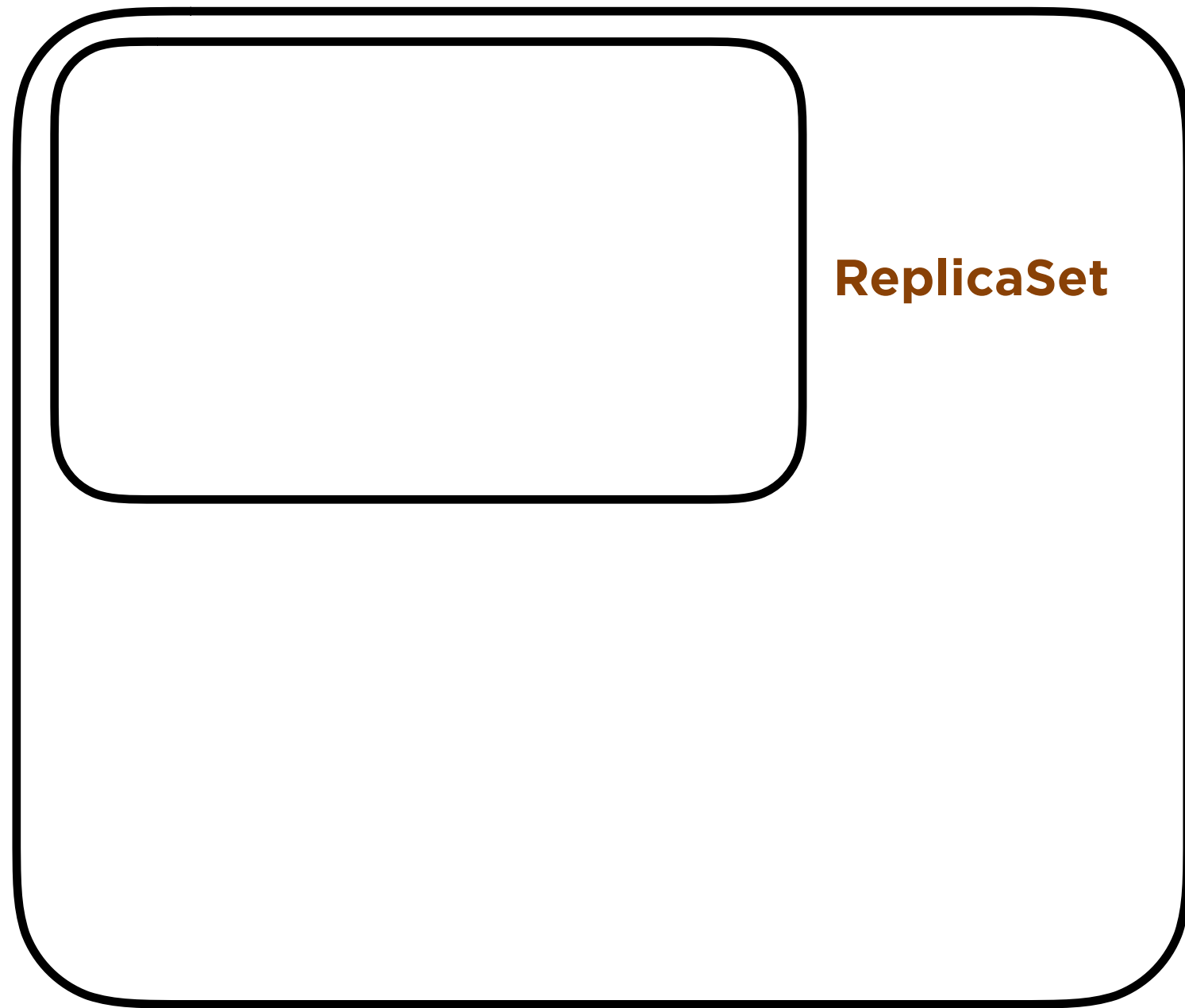


Controller Operations - Deployment



Cluster

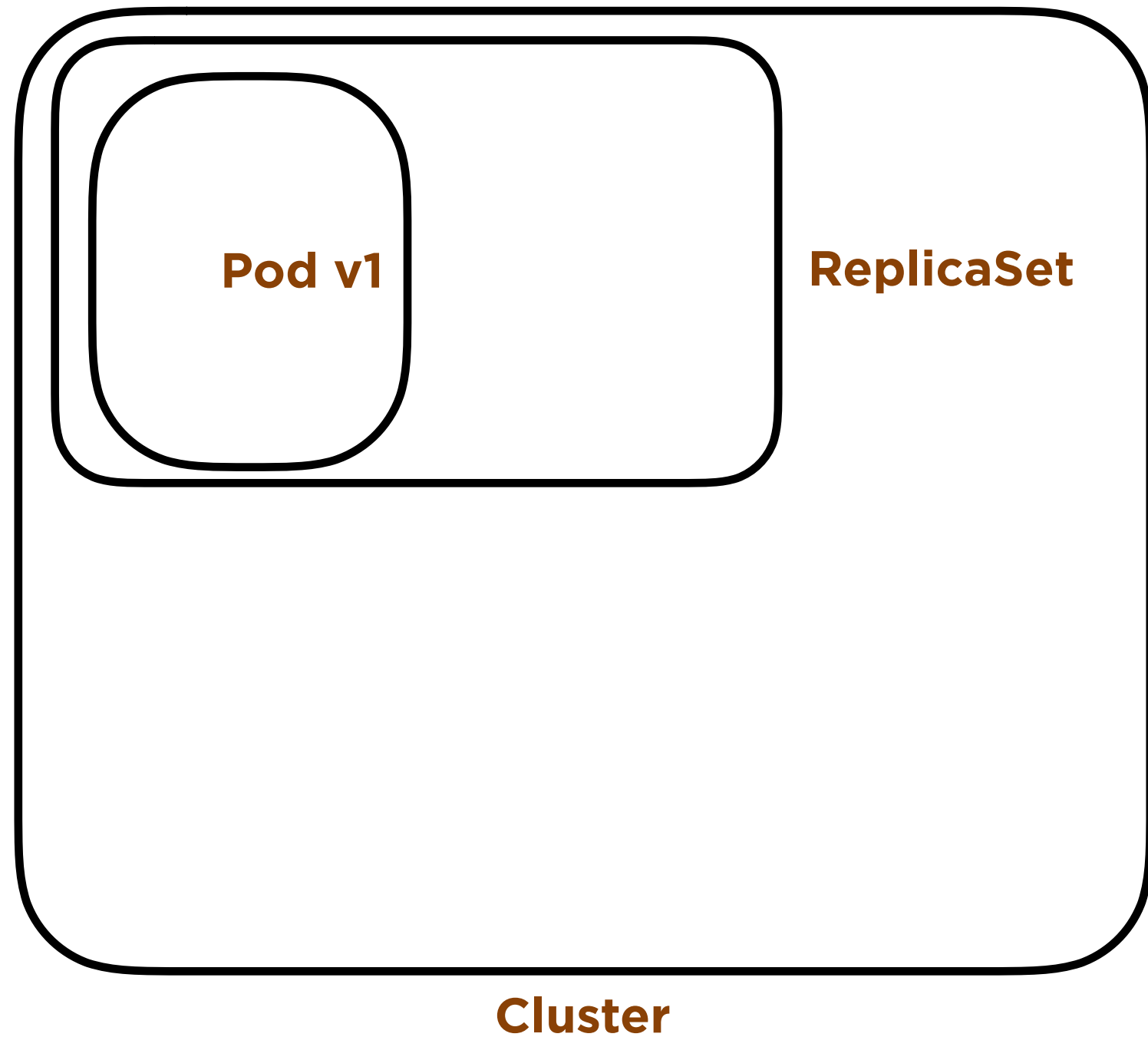
Controller Operations - Deployment



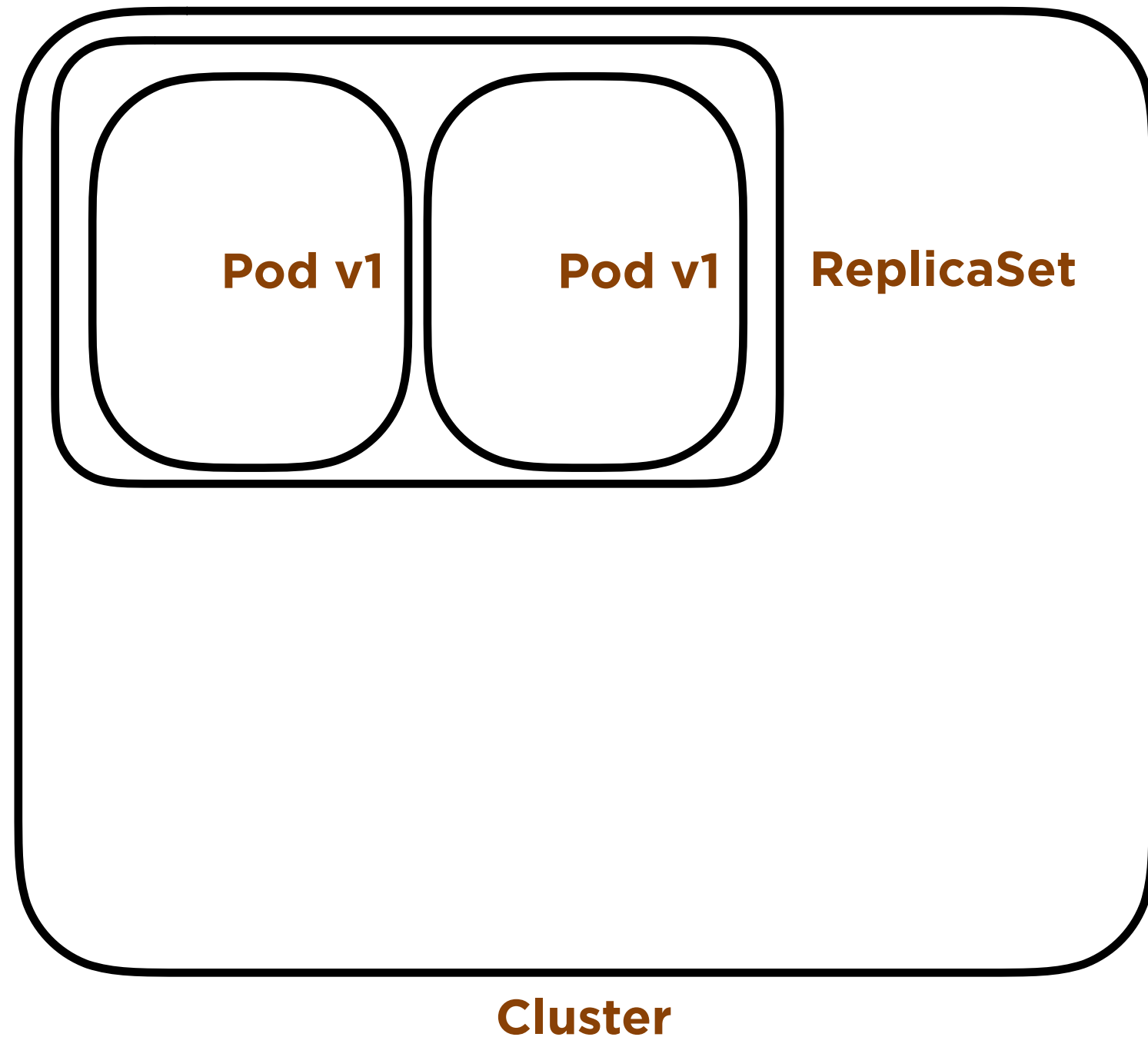
ReplicaSet

Cluster

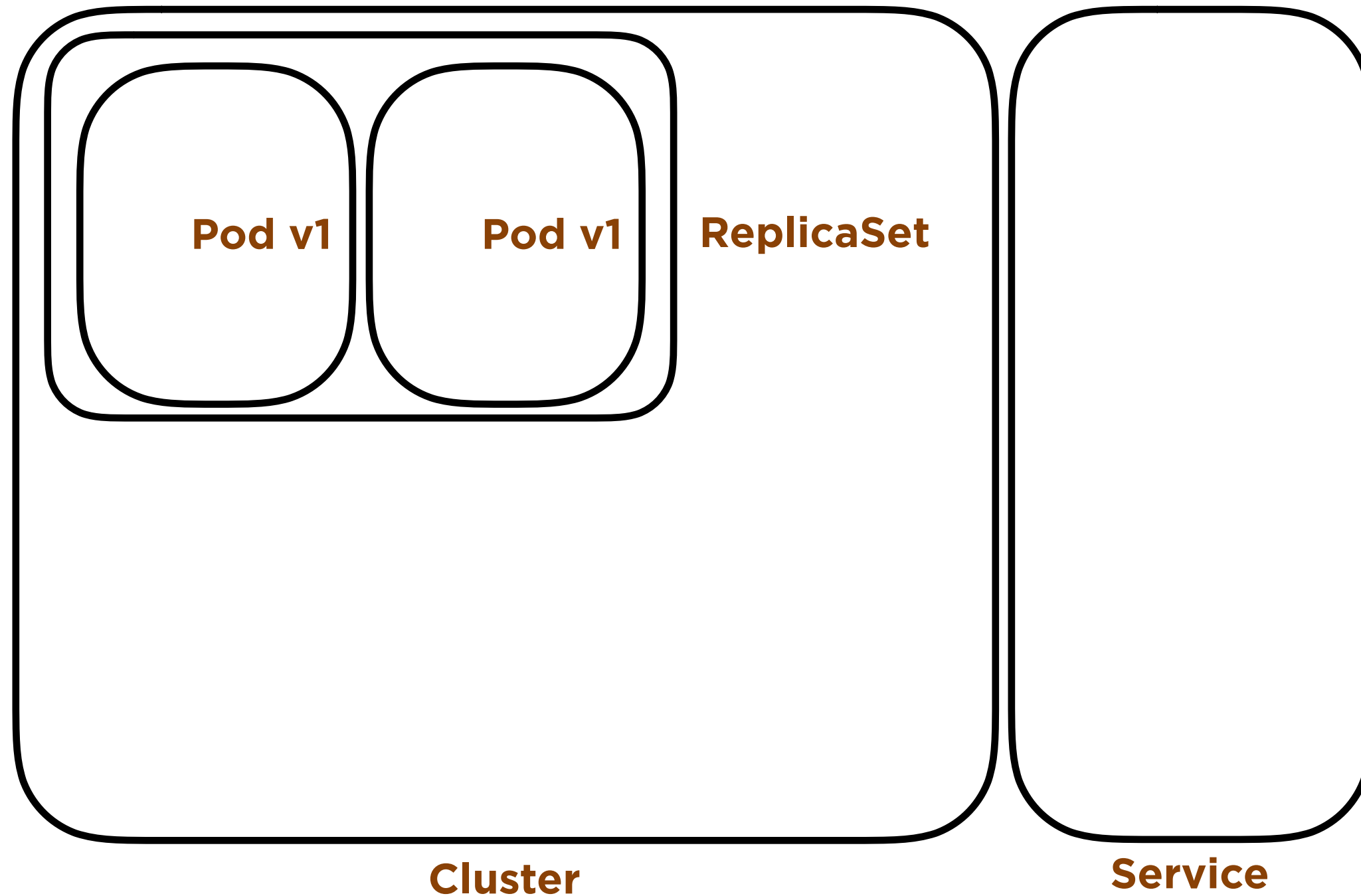
Controller Operations - Deployment



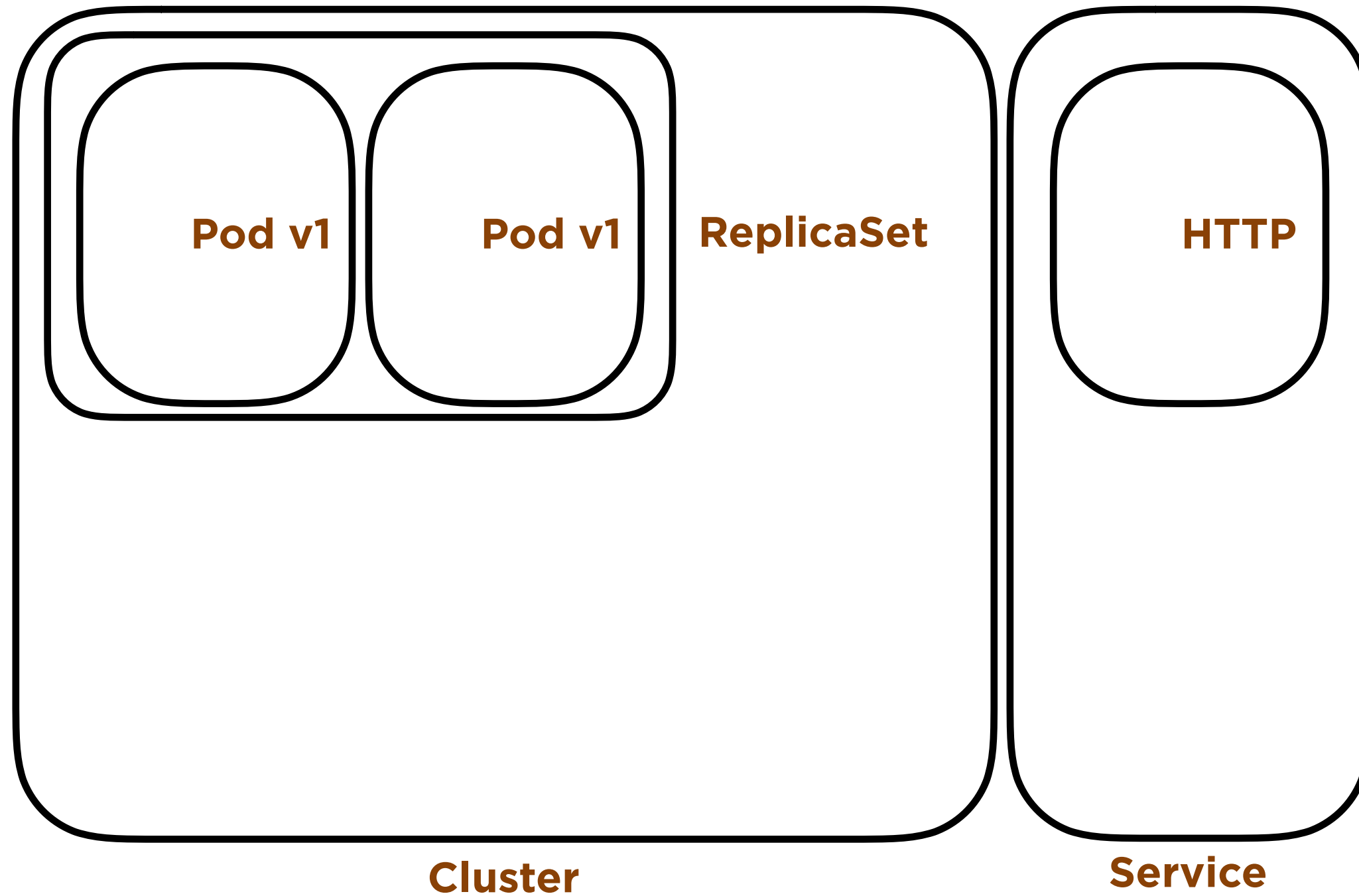
Controller Operations - Deployment



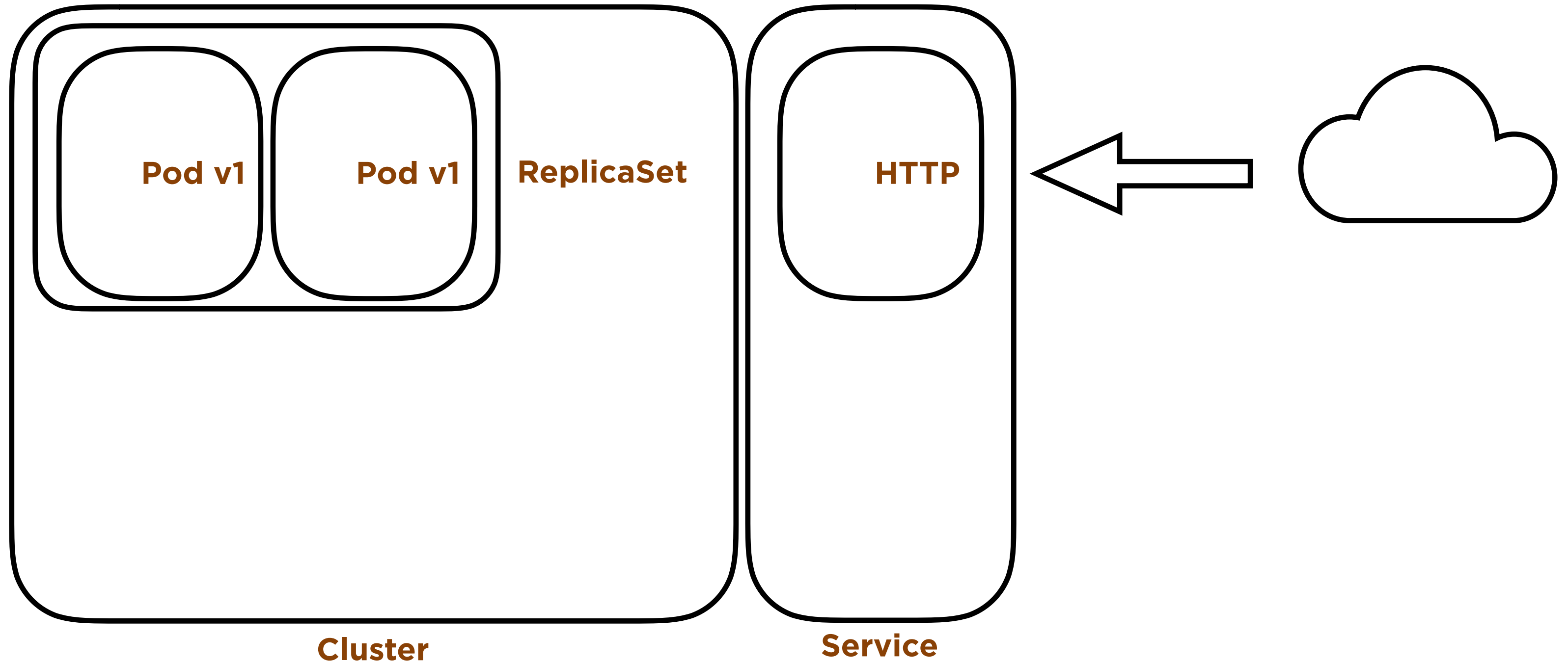
Controller Operations - Deployment



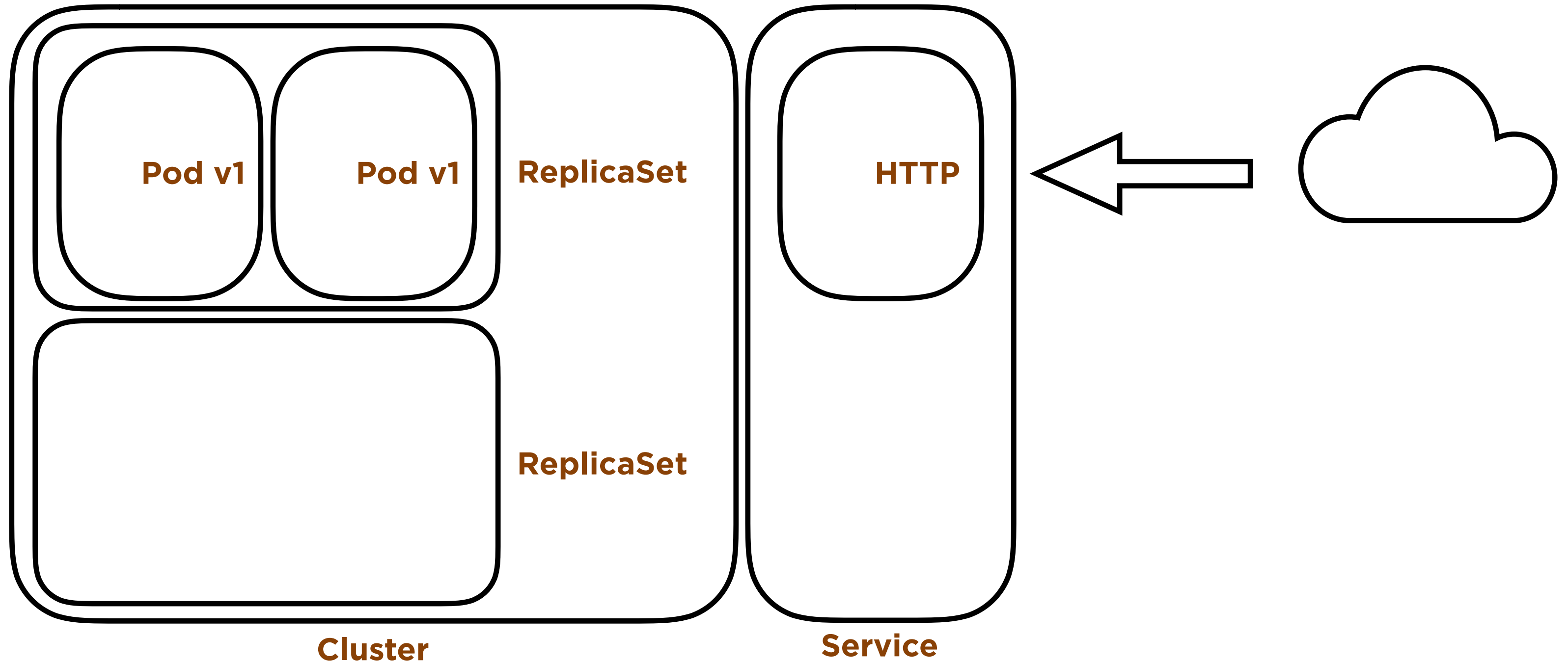
Controller Operations - Deployment



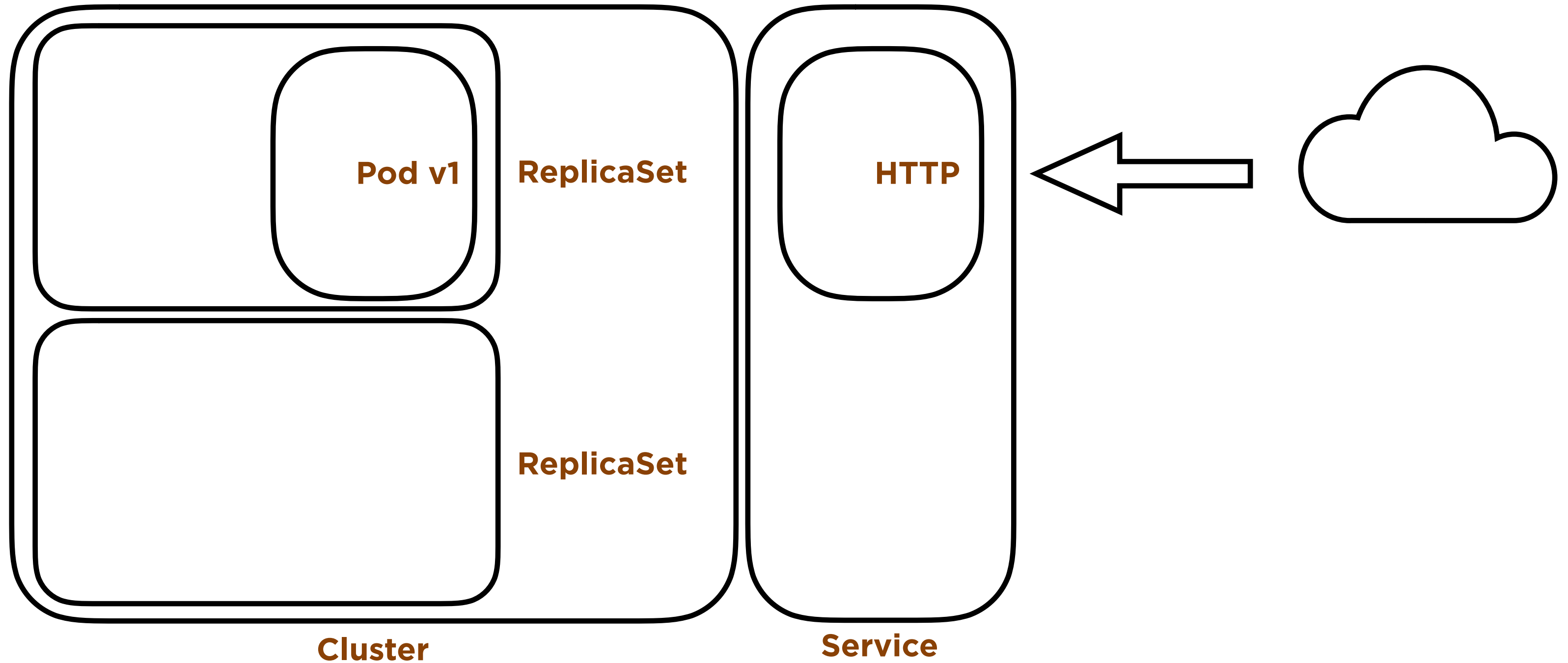
Controller Operations - Deployment



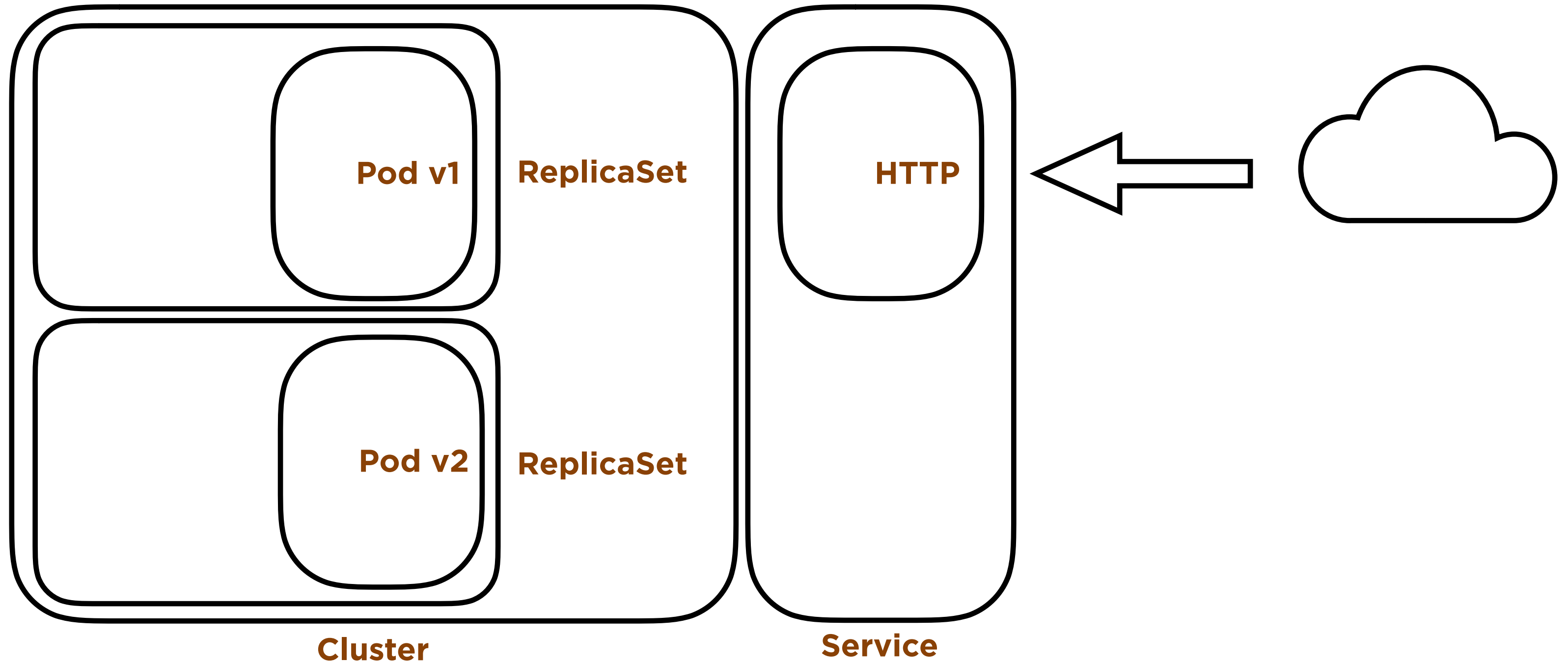
Controller Operations - Deployment



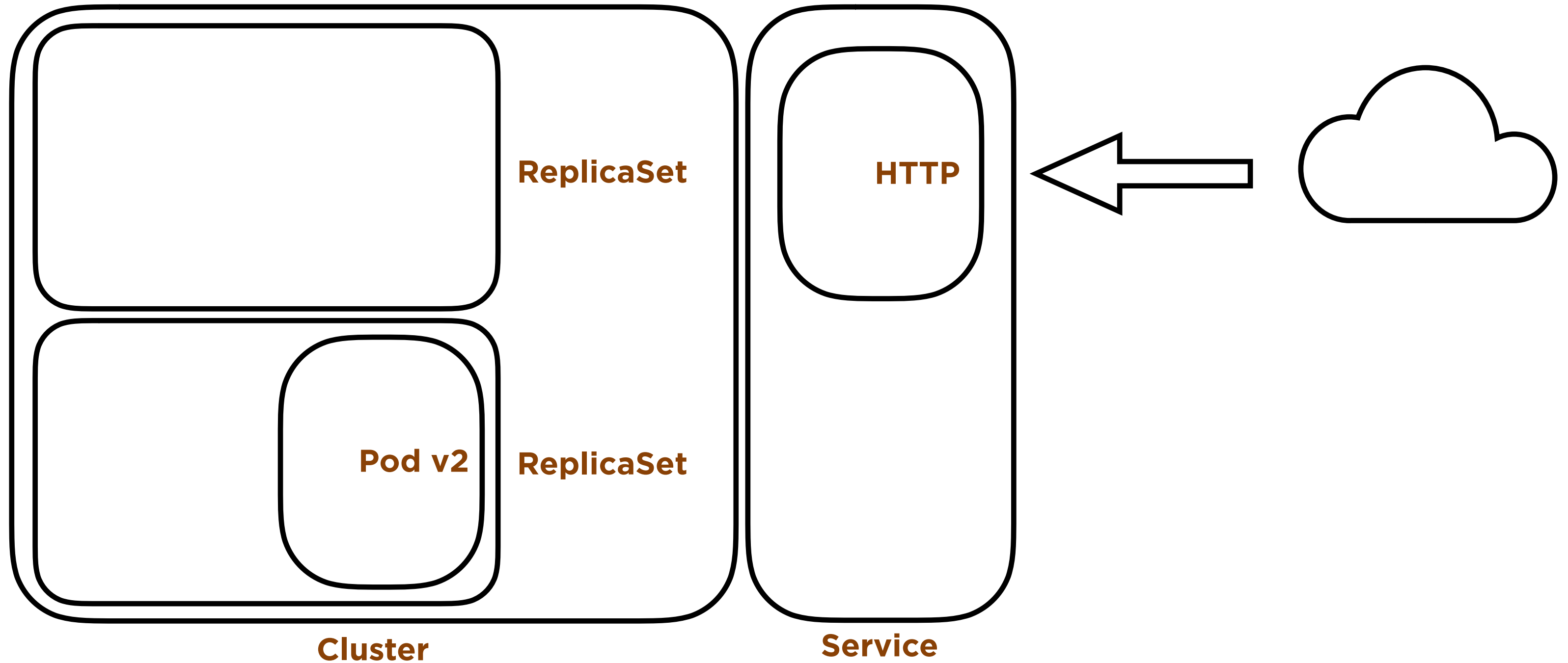
Controller Operations - Deployment



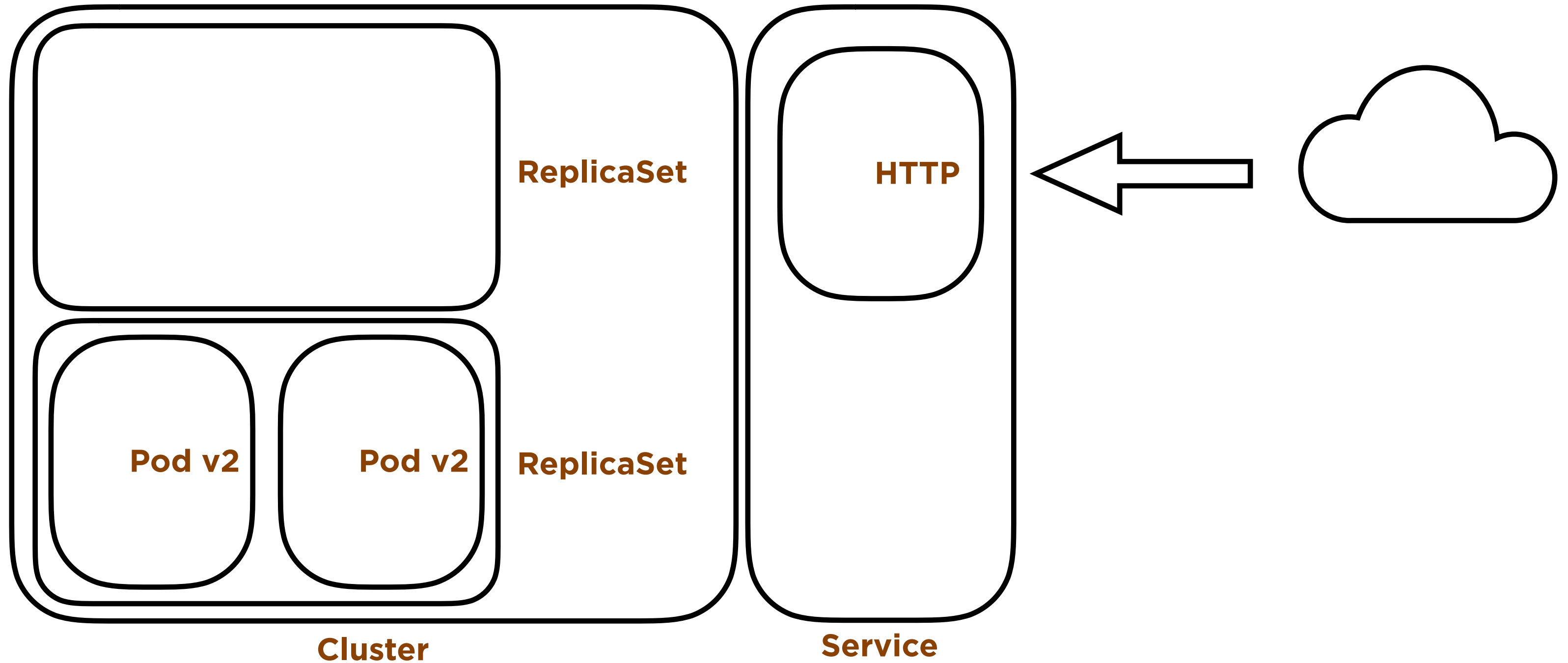
Controller Operations - Deployment



Controller Operations - Deployment



Controller Operations - Deployment



Deploying Applications

- Imperative
- Declarative
- YAML and JSON

Declarative Deployment - Manifests

Declarative Deployment - Manifests

```
apiVersion: v1
```

Declarative Deployment - Manifests

```
apiVersion: v1  
kind: Pod
```

Declarative Deployment - Manifests

```
apiVersion: v1  
kind: Pod  
metadata:
```

Declarative Deployment - Manifests

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: nginx-pod
```


Declarative Deployment - Manifests

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: nginx-pod  
spec:
```

Declarative Deployment - Manifests

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
```

Declarative Deployment - Manifests

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx
```

Declarative Deployment - Manifests

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

Declarative Deployment - Manifests

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

Declarative Deployment - Manifests

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

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

Application Deployment Process

Master

Cluster Store

Scheduler

Controller
Manager

API Server

Node

Kubelet

Kube-proxy

Container
Runtime

Application Deployment Process

Master

Cluster Store

Scheduler

Controller
Manager

API Server

Node

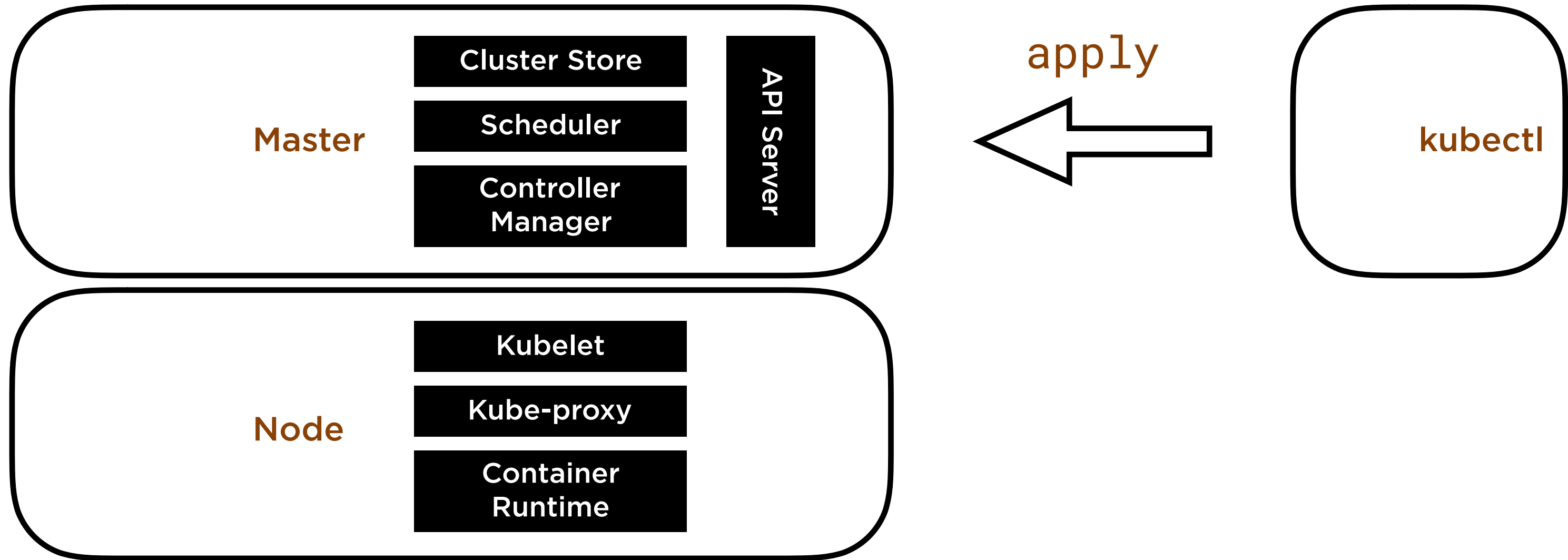
Kubelet

Kube-proxy

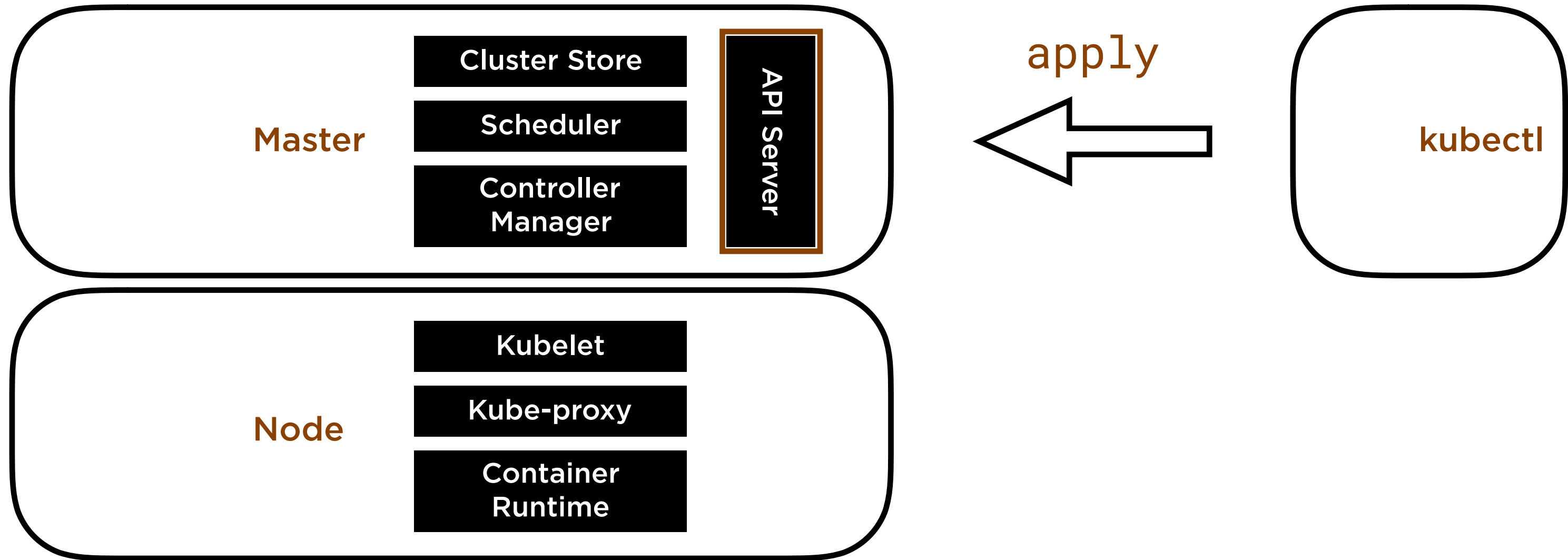
Container
Runtime

kubectl

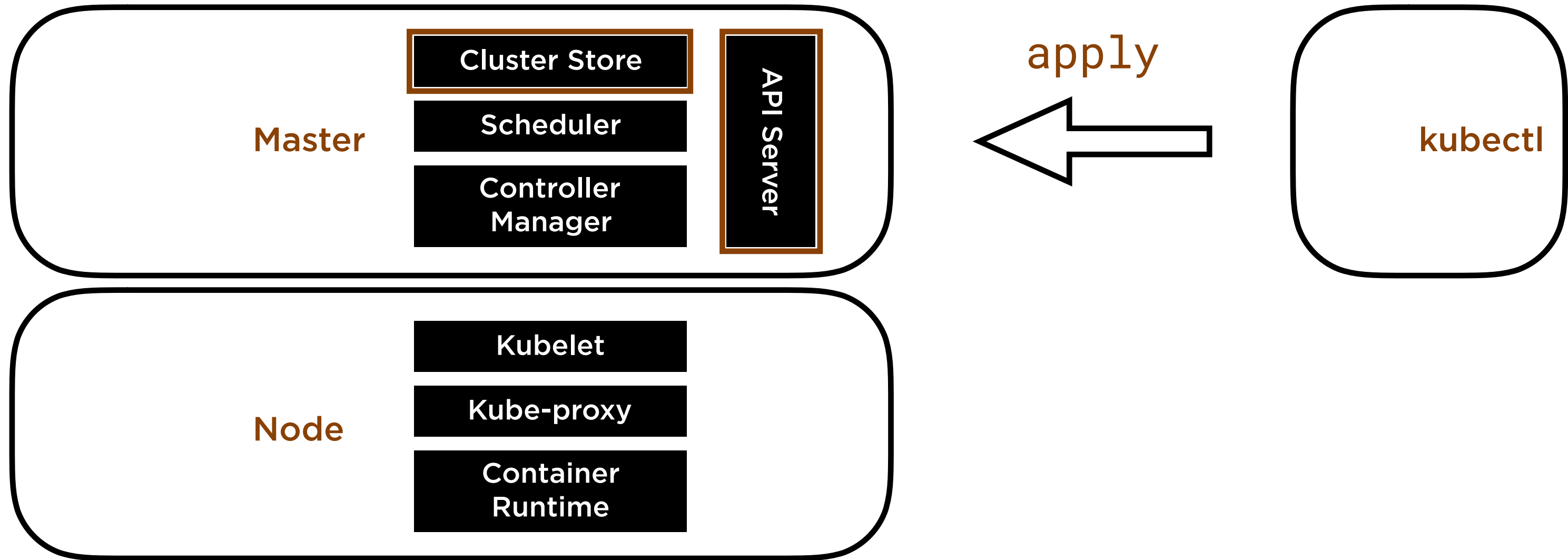
Application Deployment Process



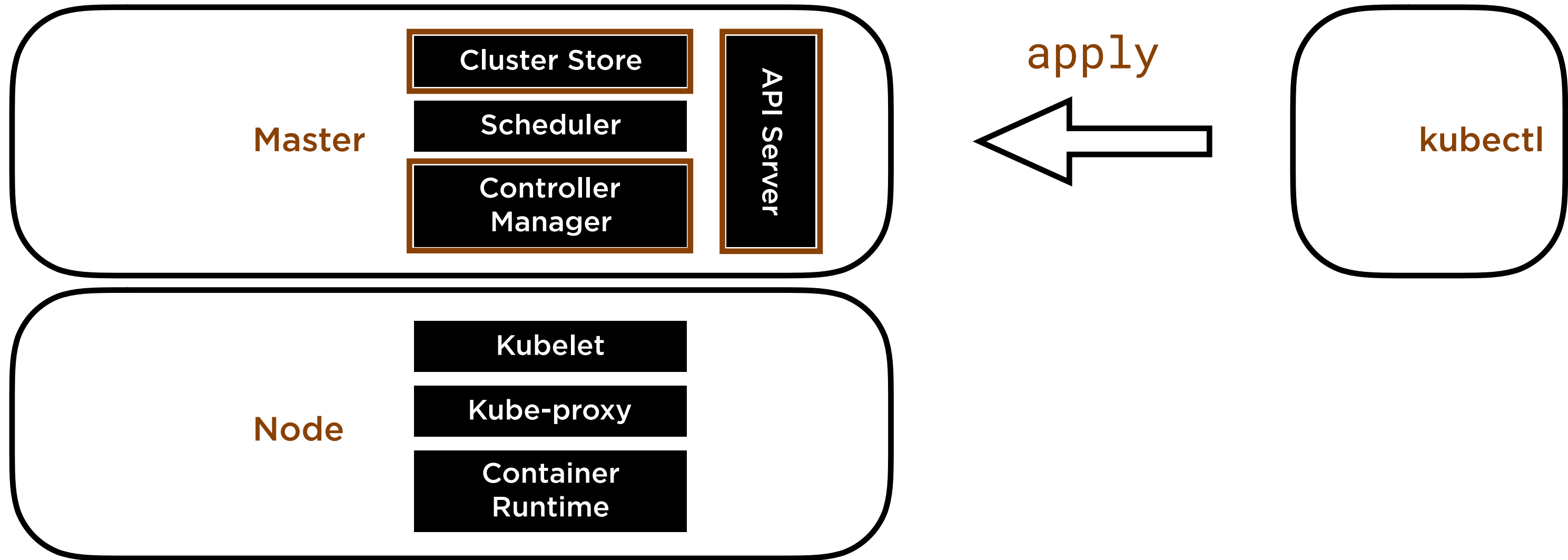
Application Deployment Process



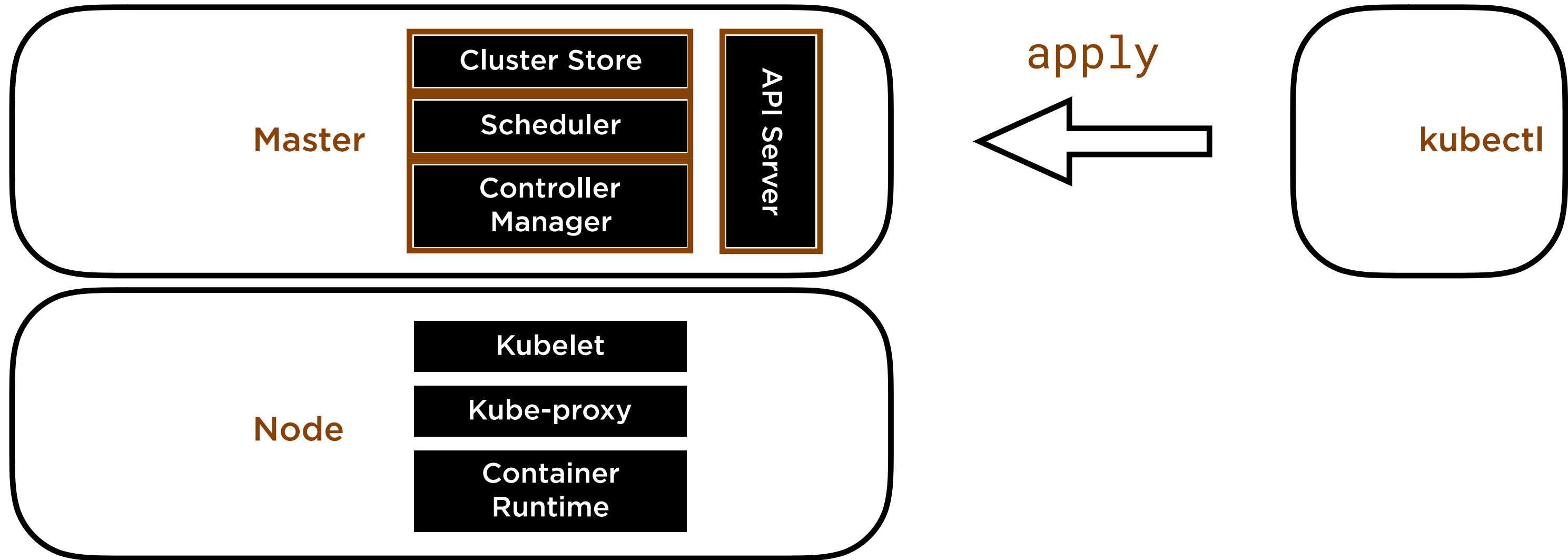
Application Deployment Process



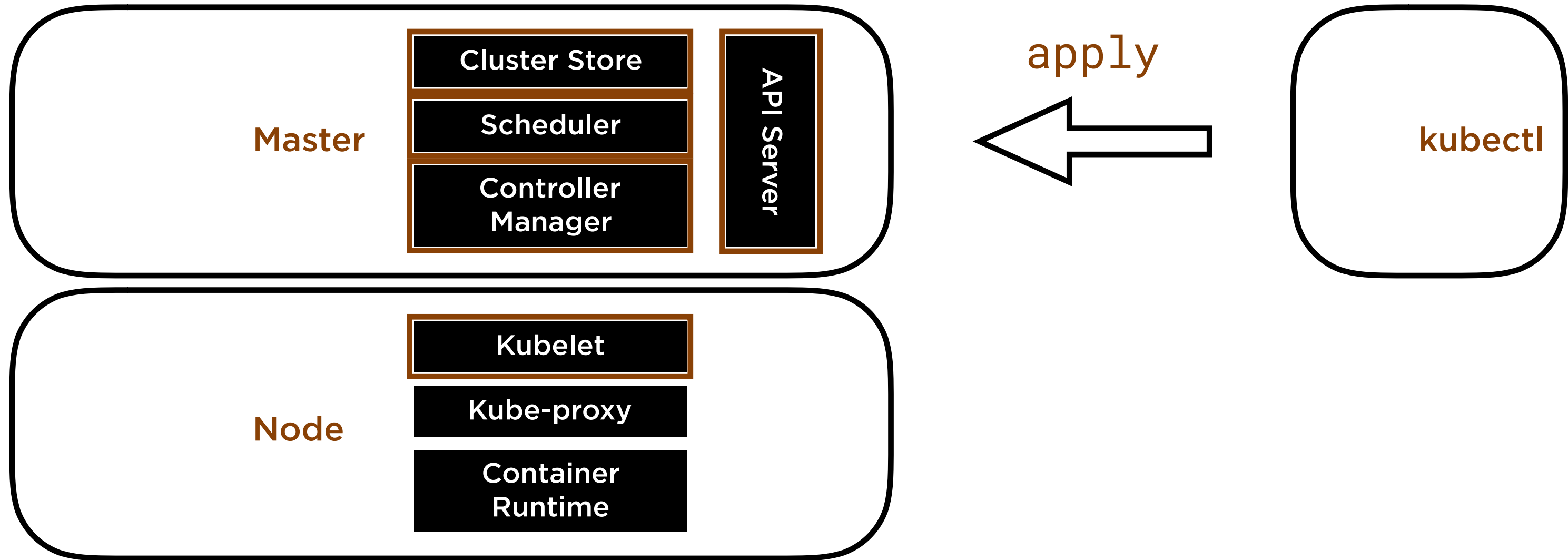
Application Deployment Process



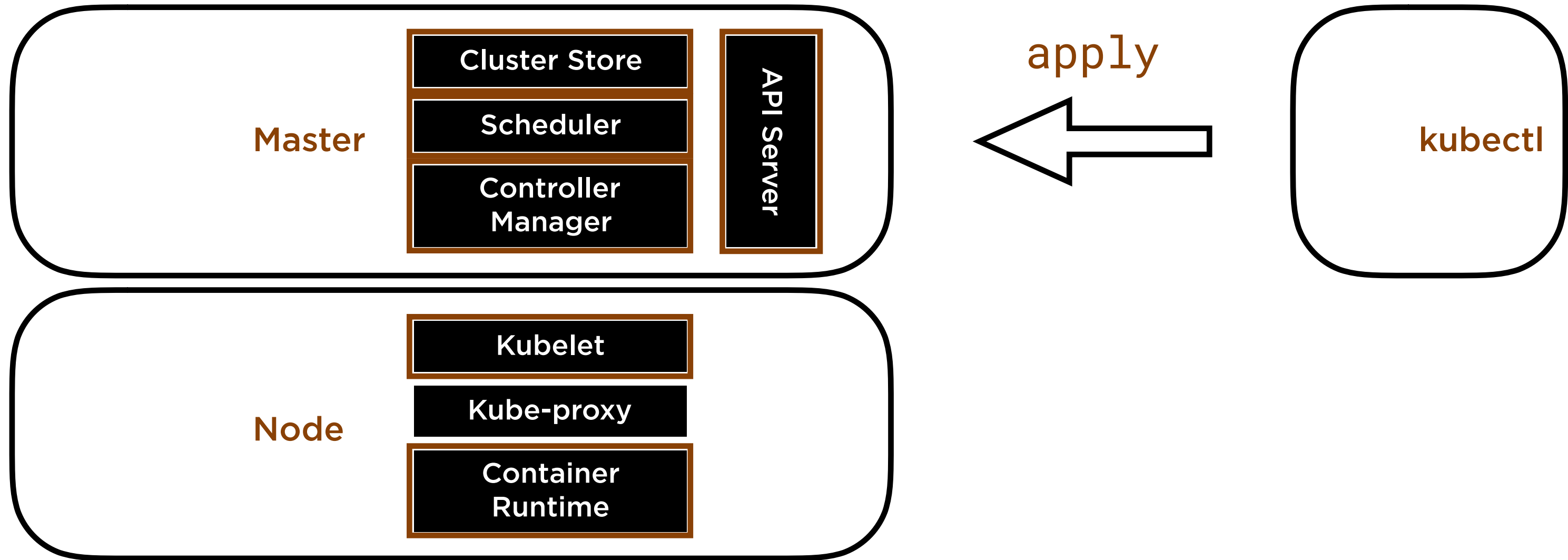
Application Deployment Process



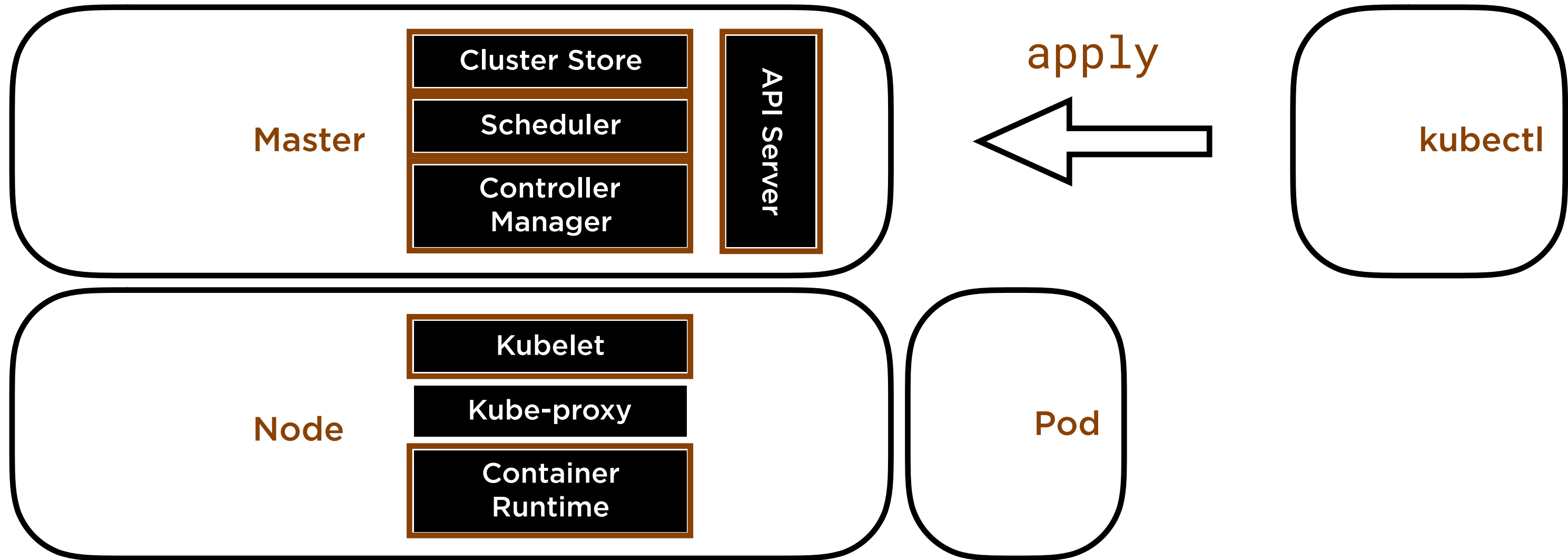
Application Deployment Process



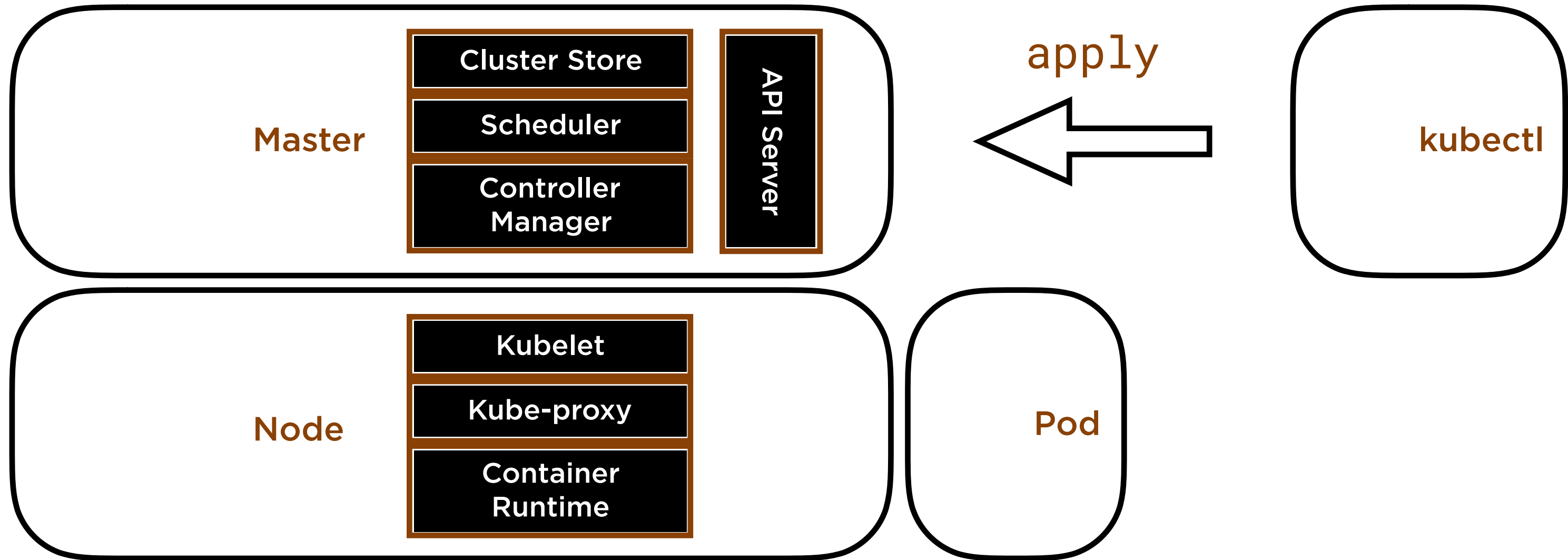
Application Deployment Process



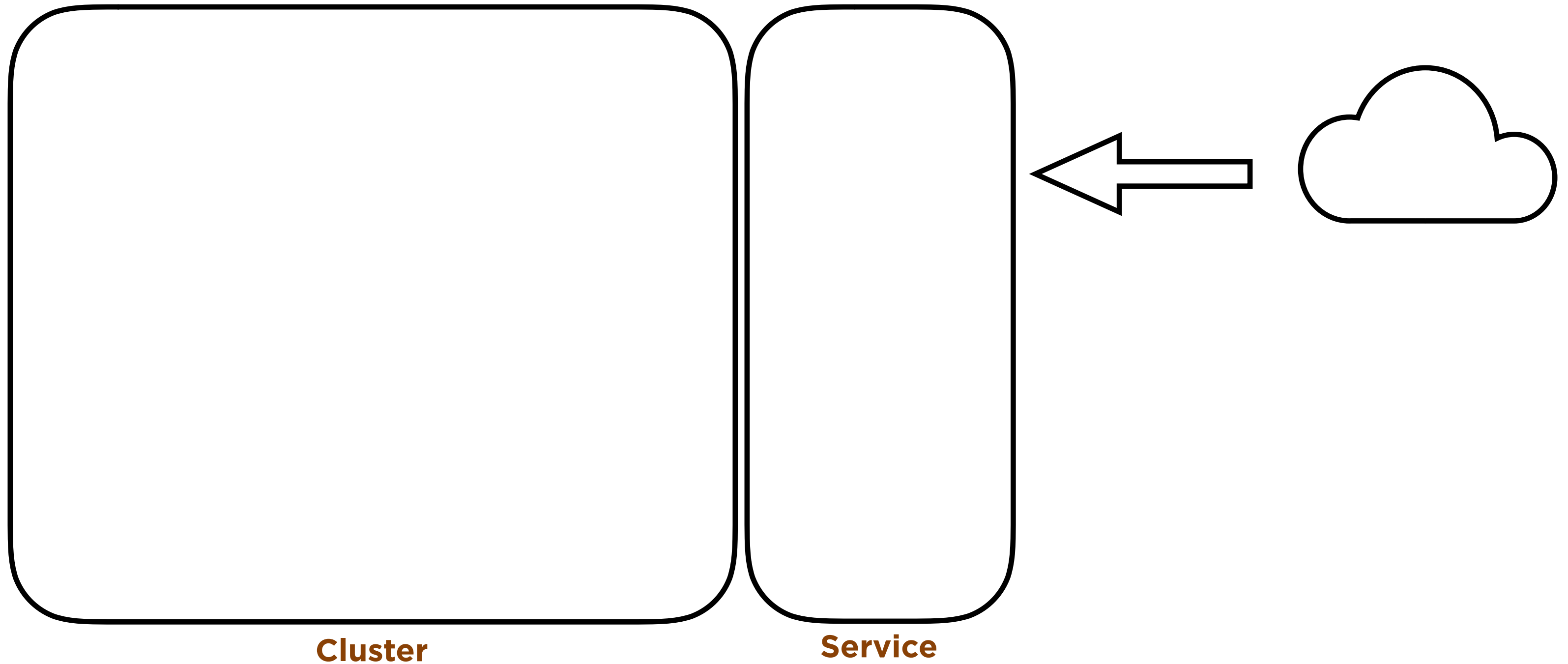
Application Deployment Process



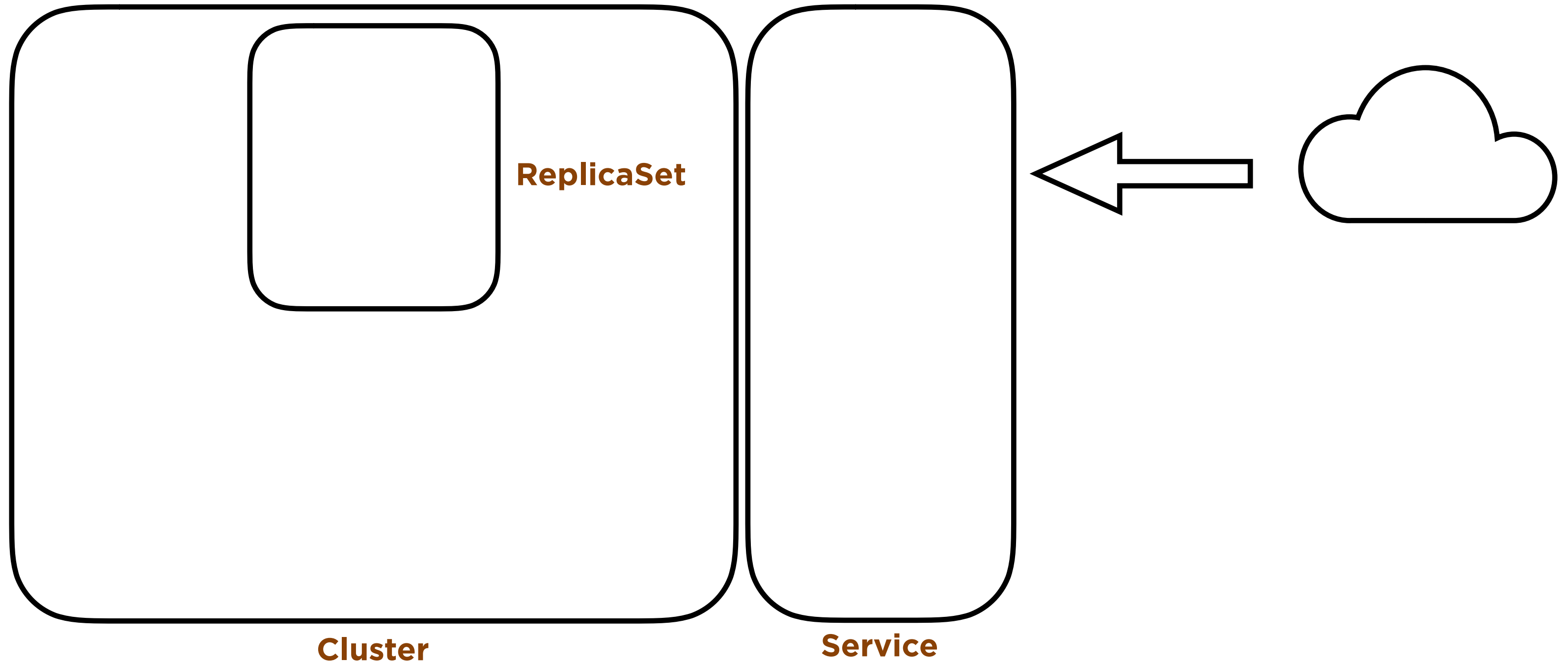
Application Deployment Process



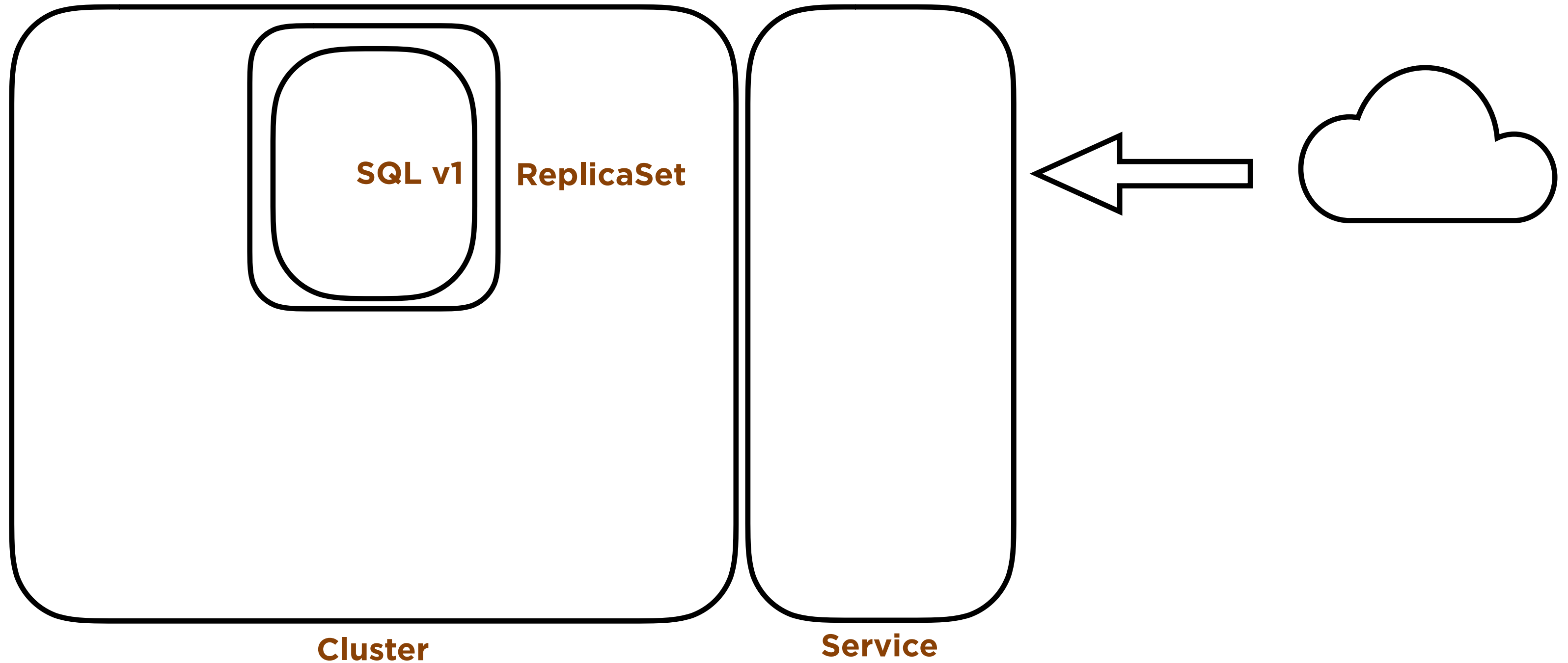
Decoupling Data and Computation



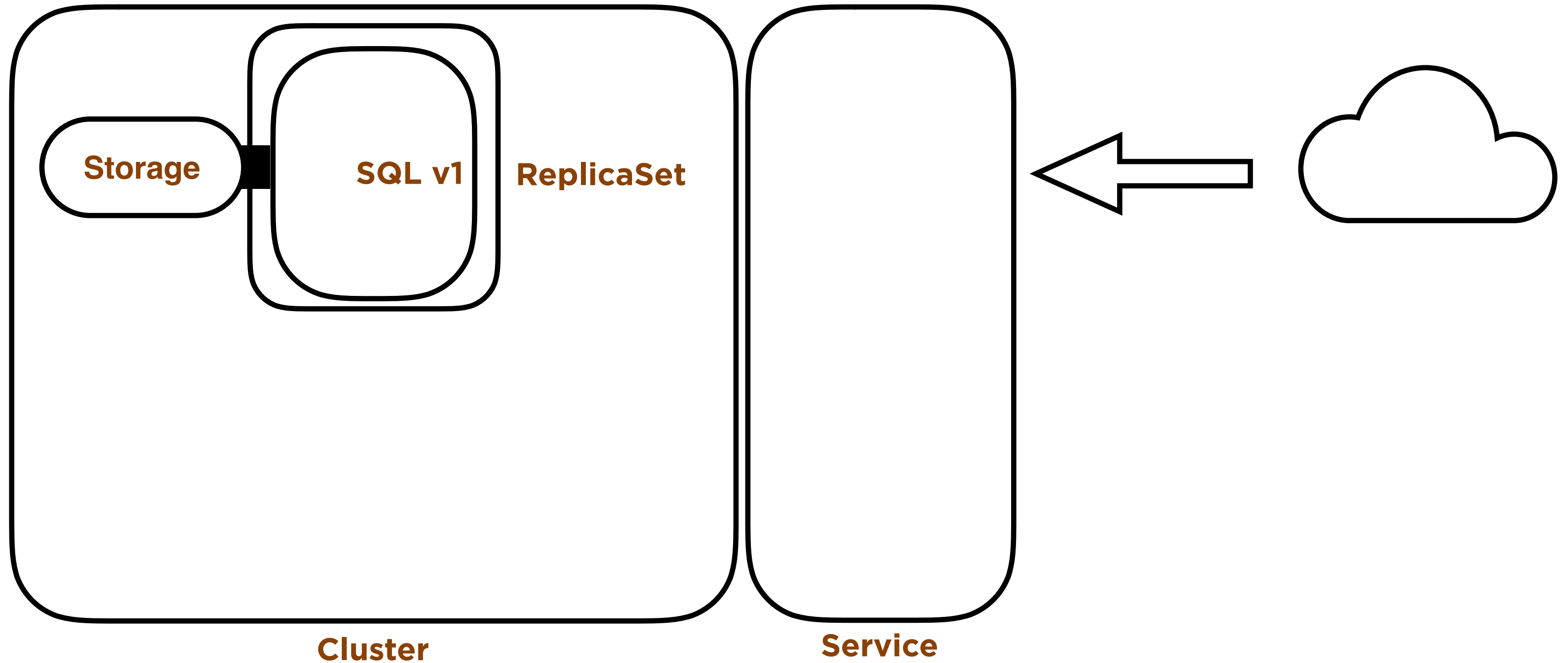
Decoupling Data and Computation



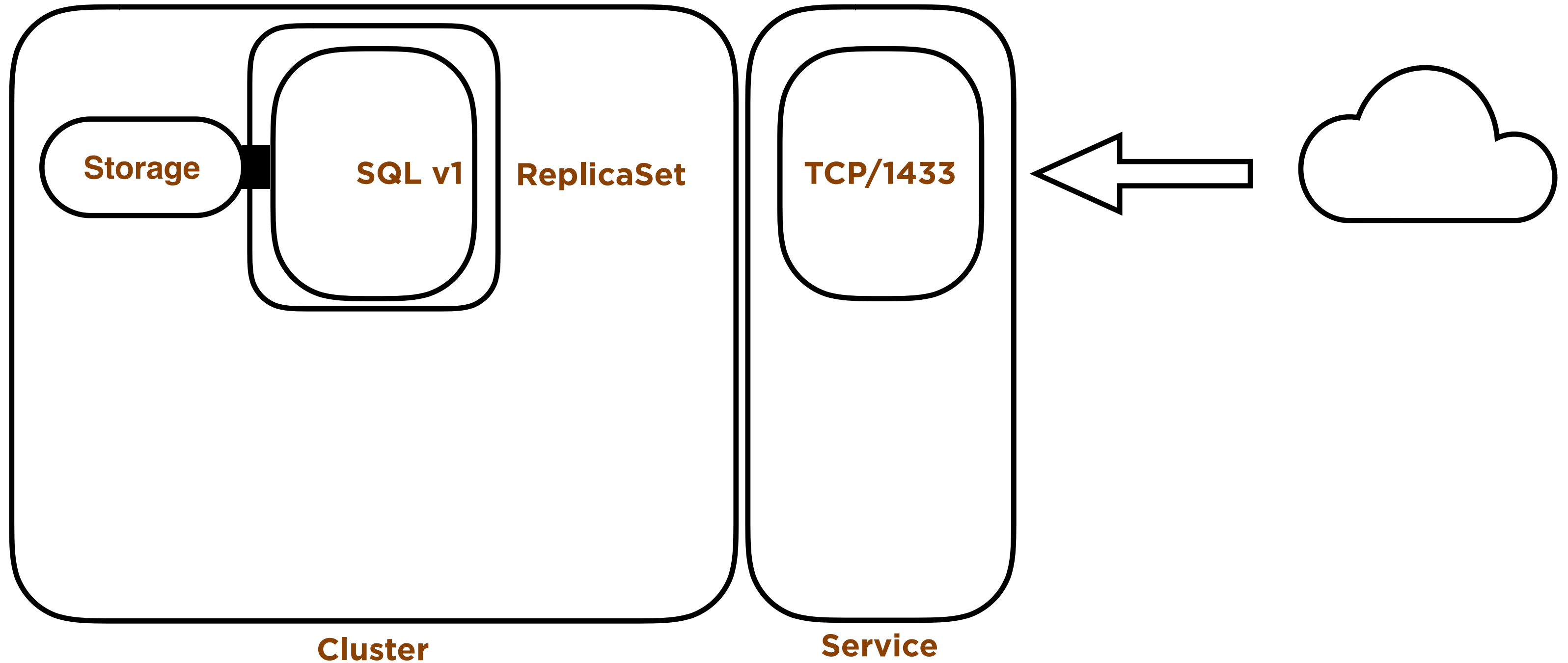
Decoupling Data and Computation



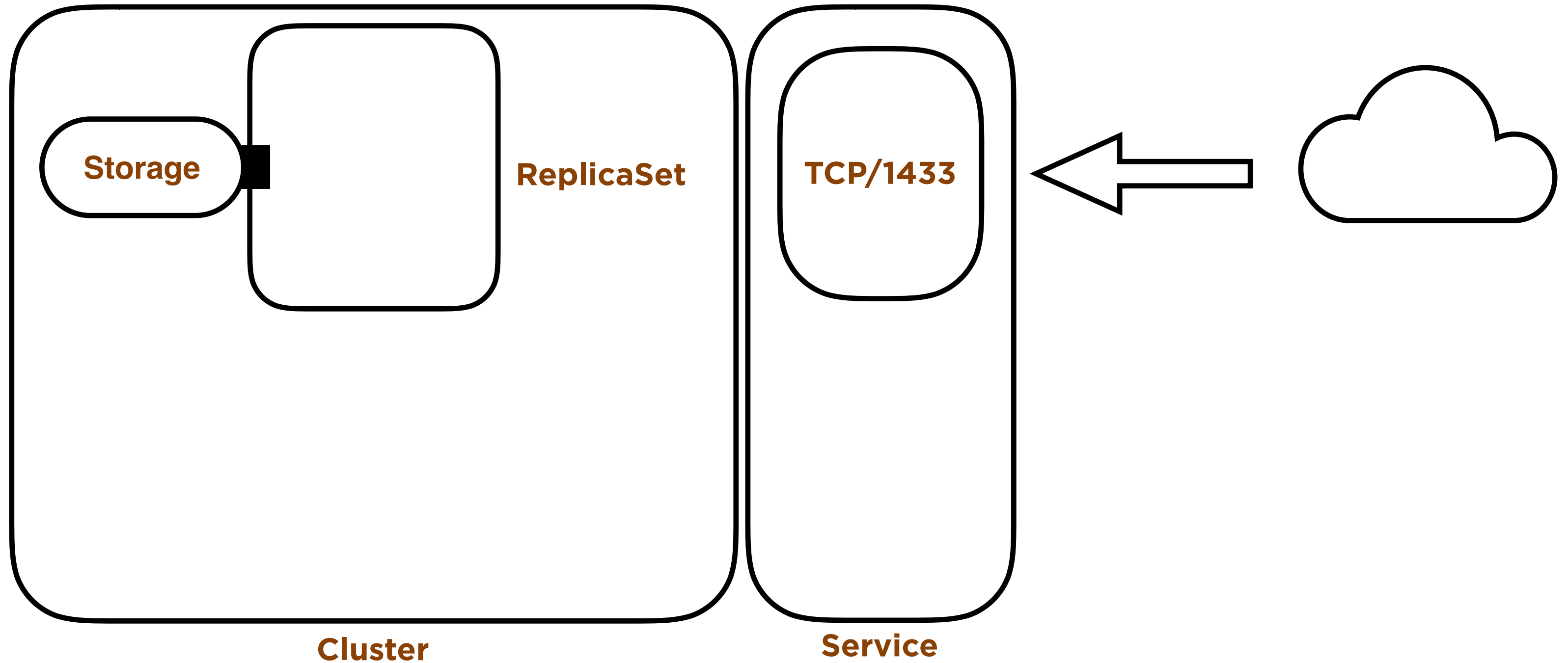
Decoupling Data and Computation



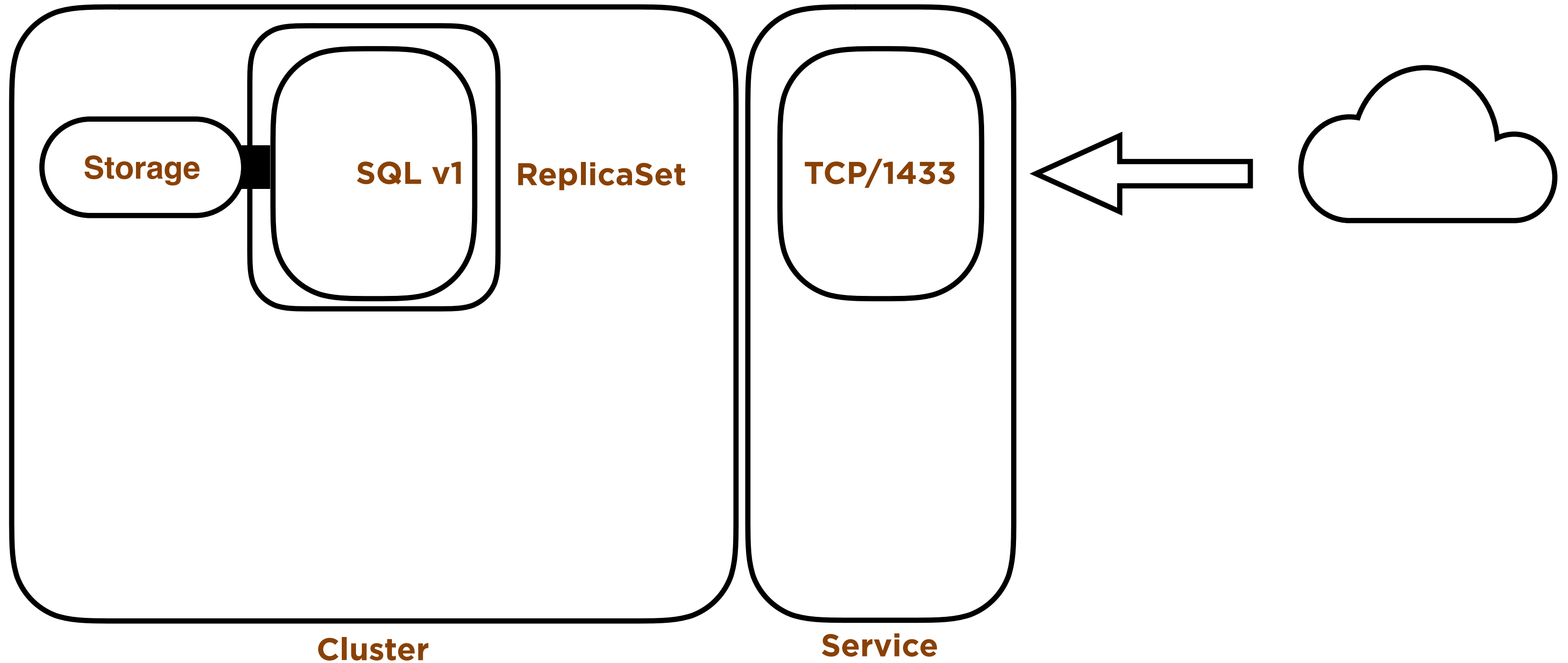
Decoupling Data and Computation



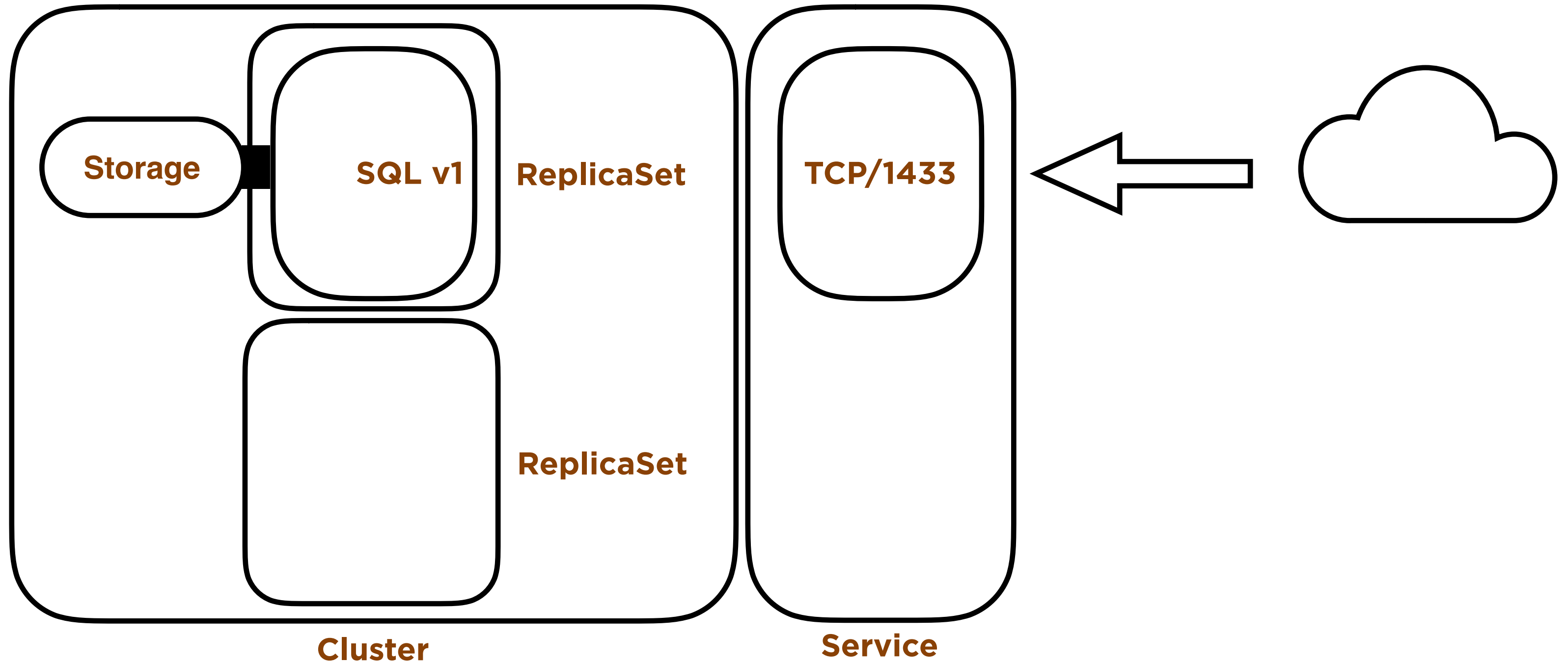
Decoupling Data and Computation



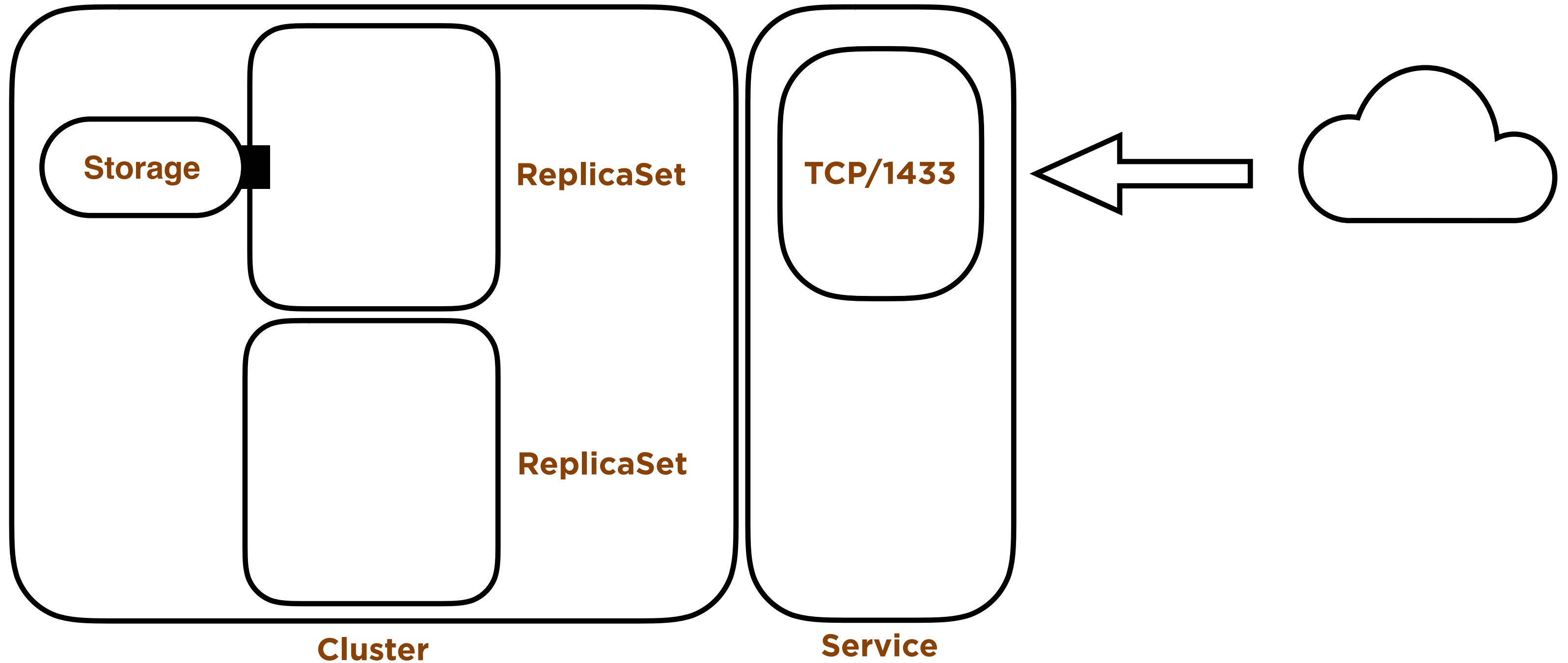
Decoupling Data and Computation



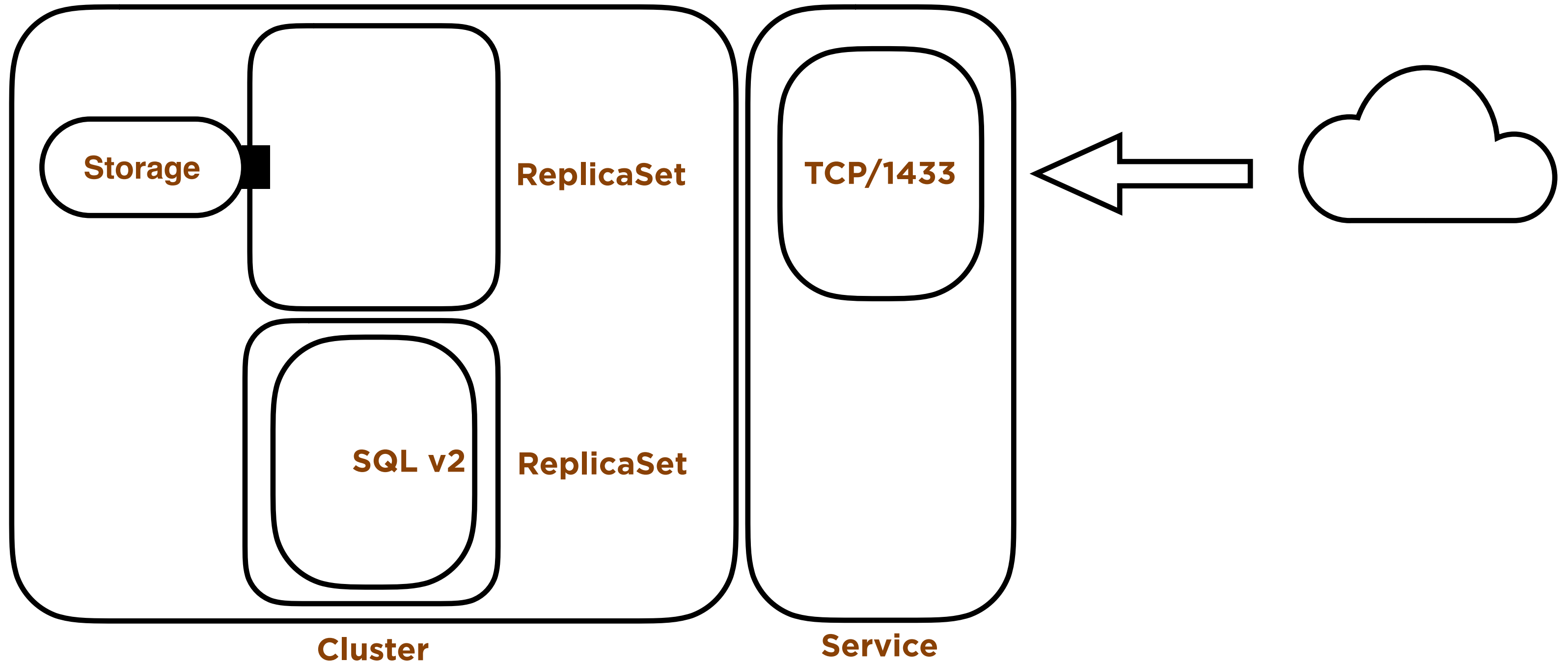
Decoupling Data and Computation



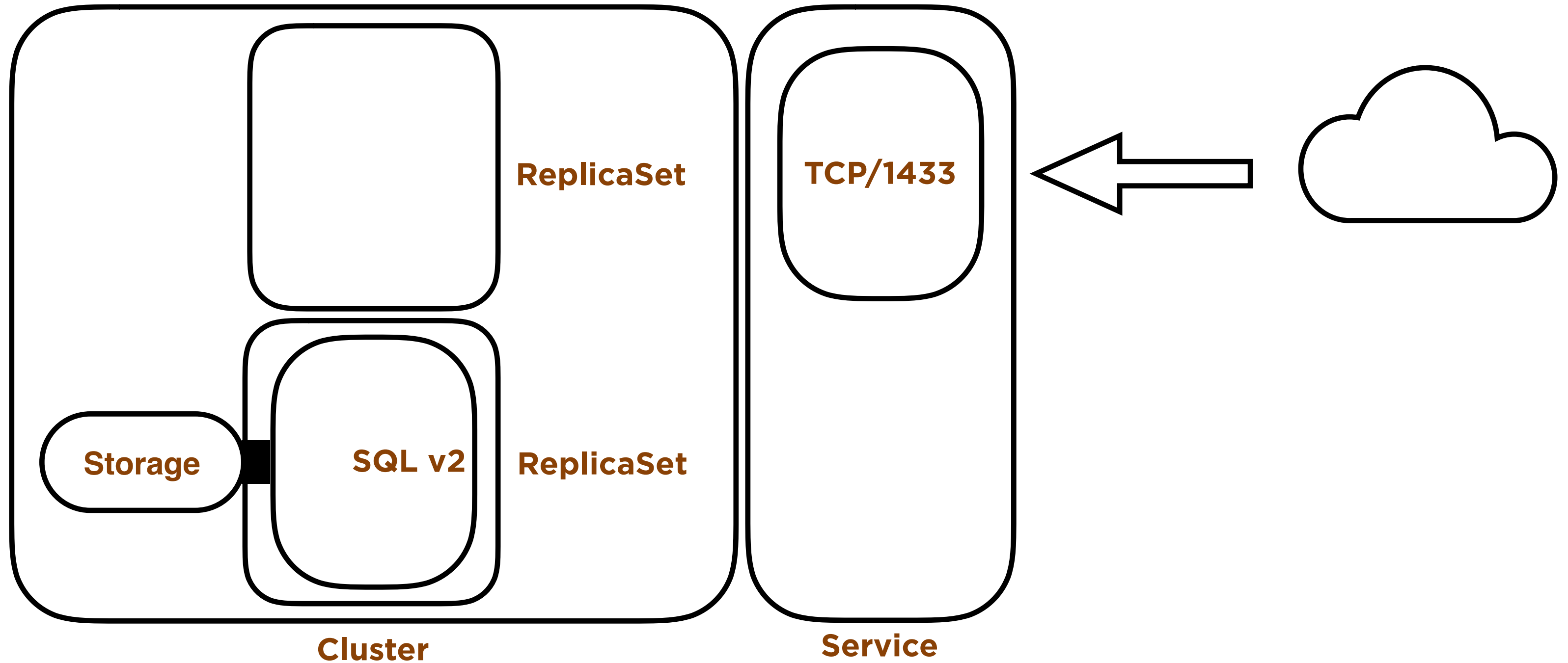
Decoupling Data and Computation



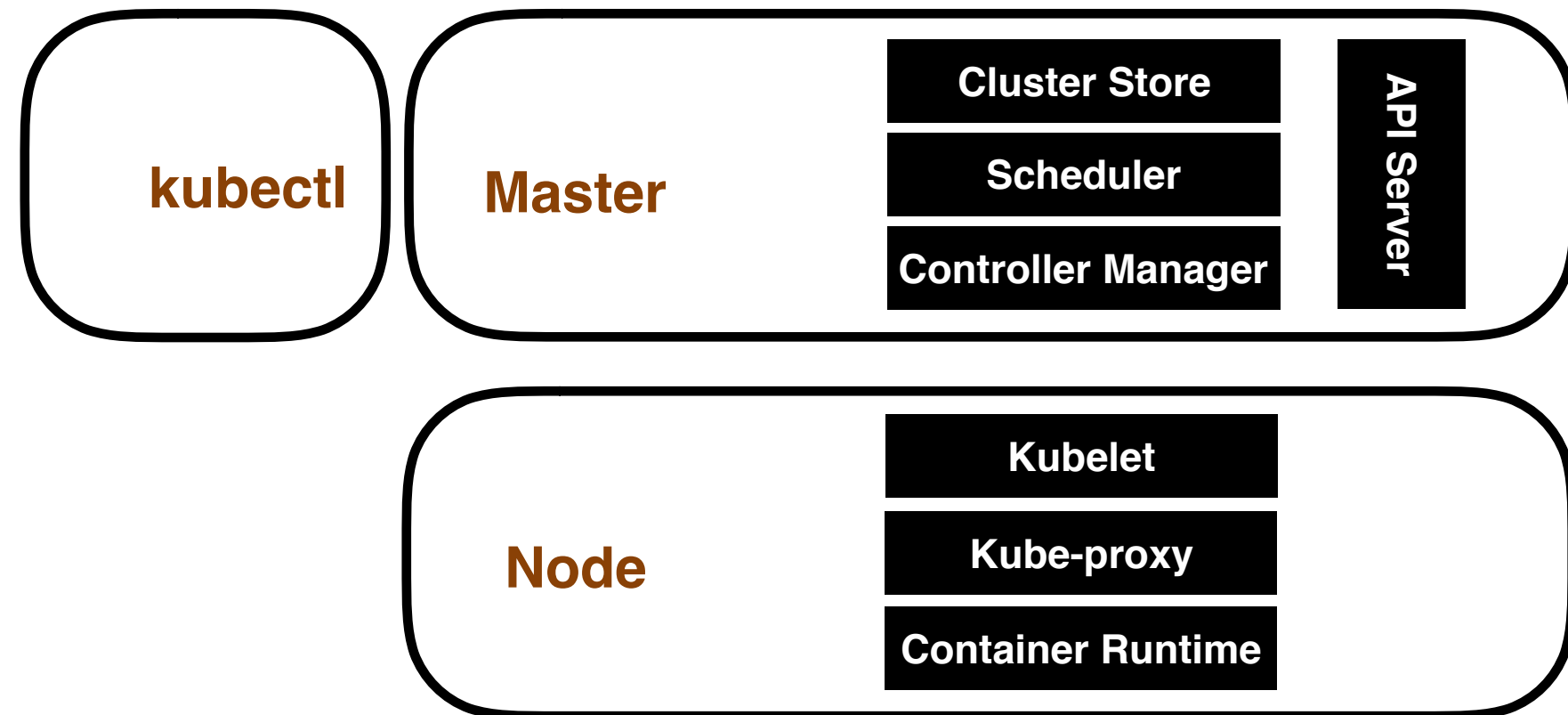
Decoupling Data and Computation



Decoupling Data and Computation

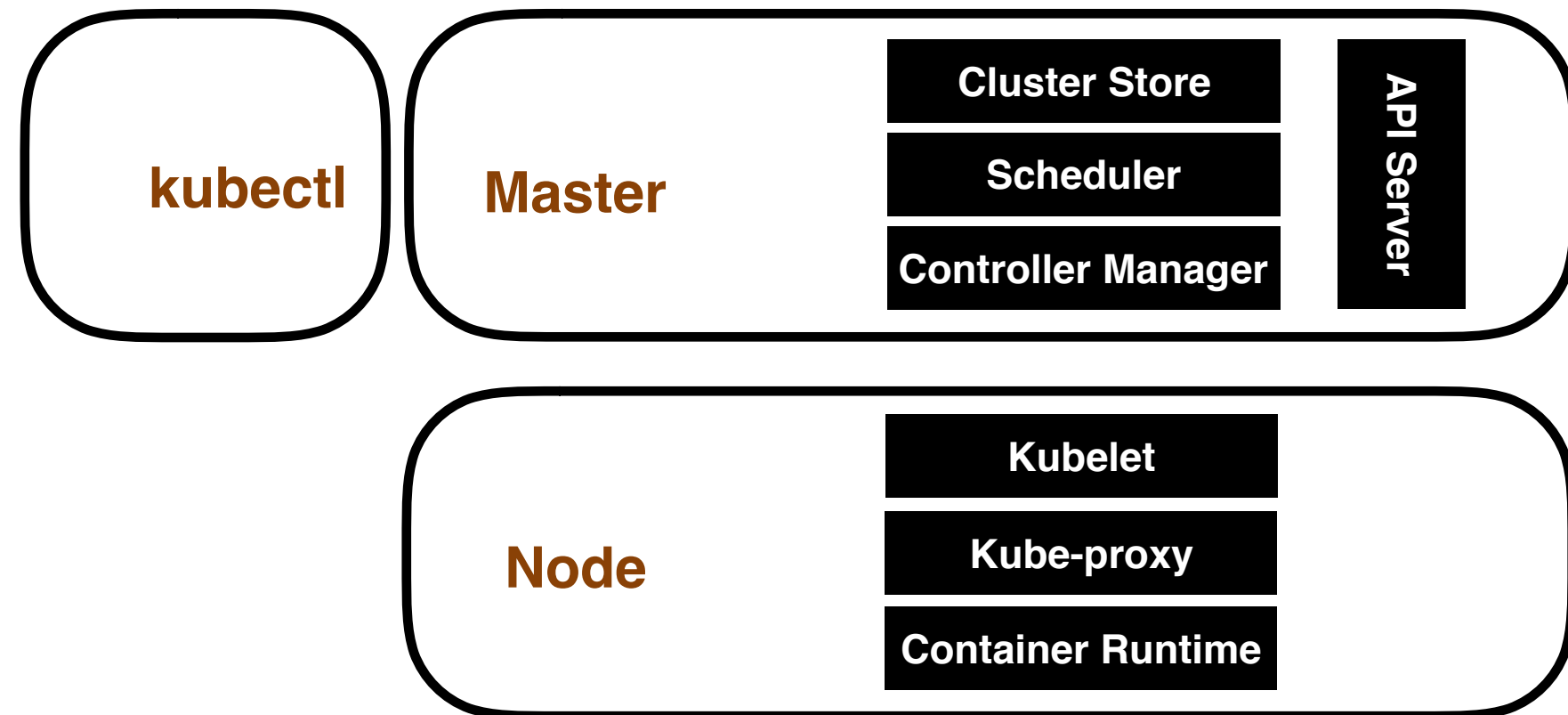


Demo!



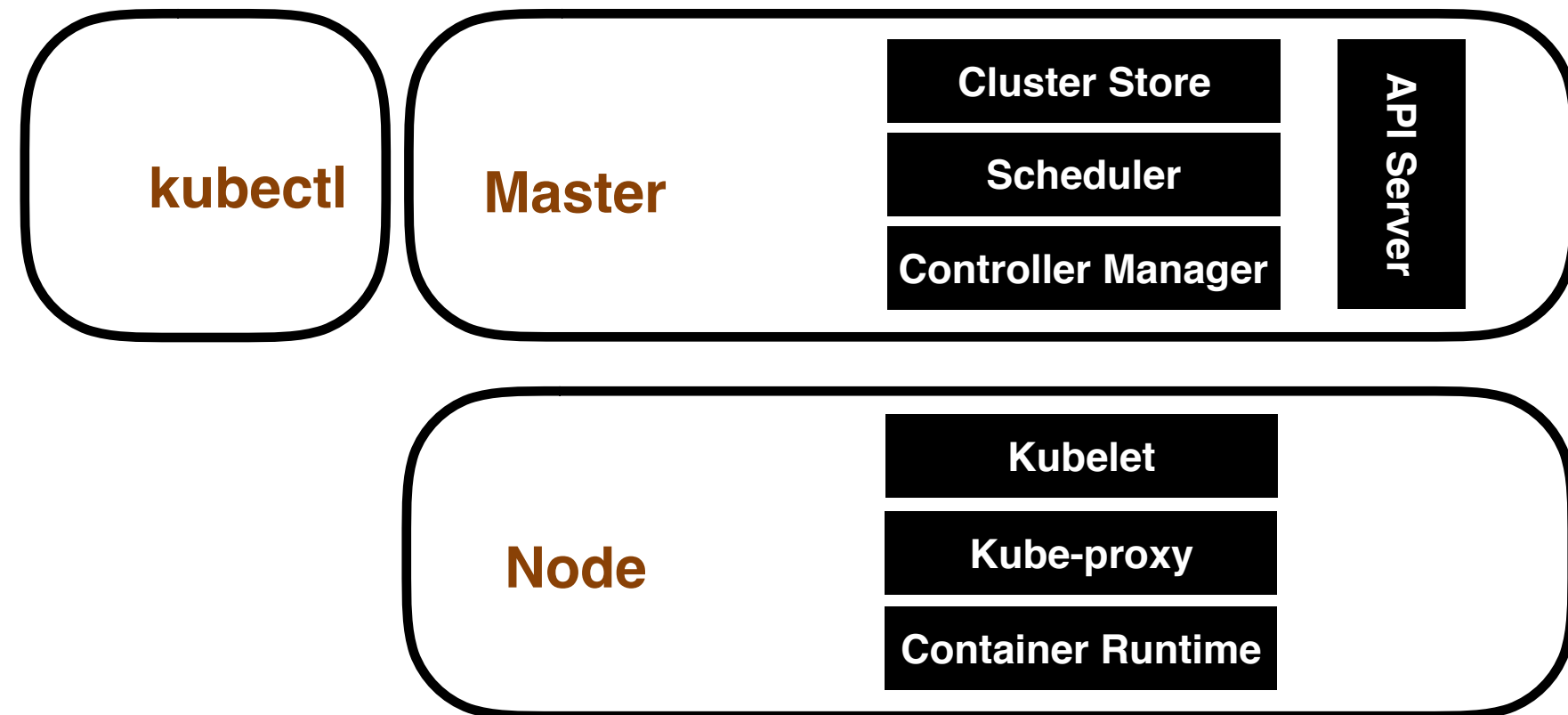
Demo!

- Deploy a stateless web app



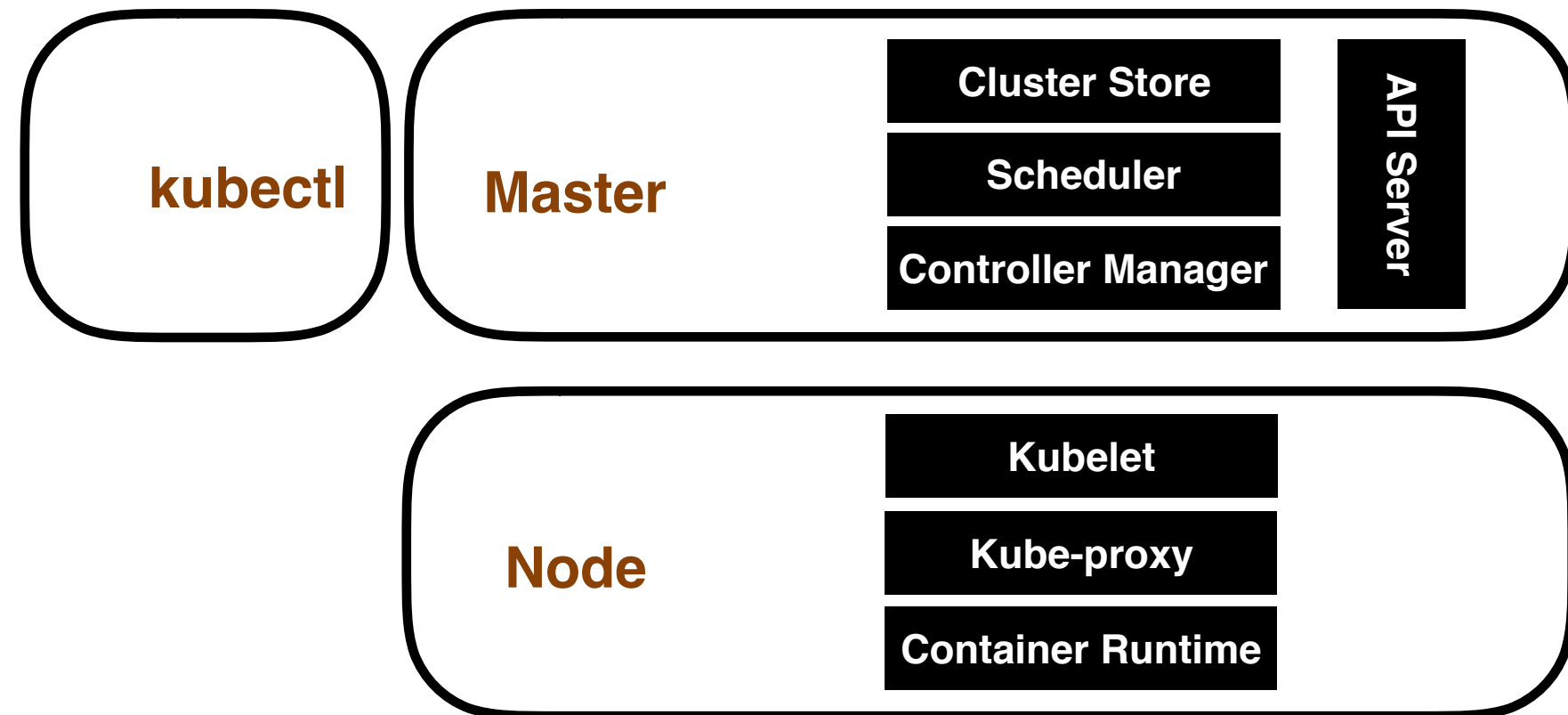
Demo!

- Deploy a stateless web app
- Deploying SQL Server in a **Deployment** with Persistent Storage



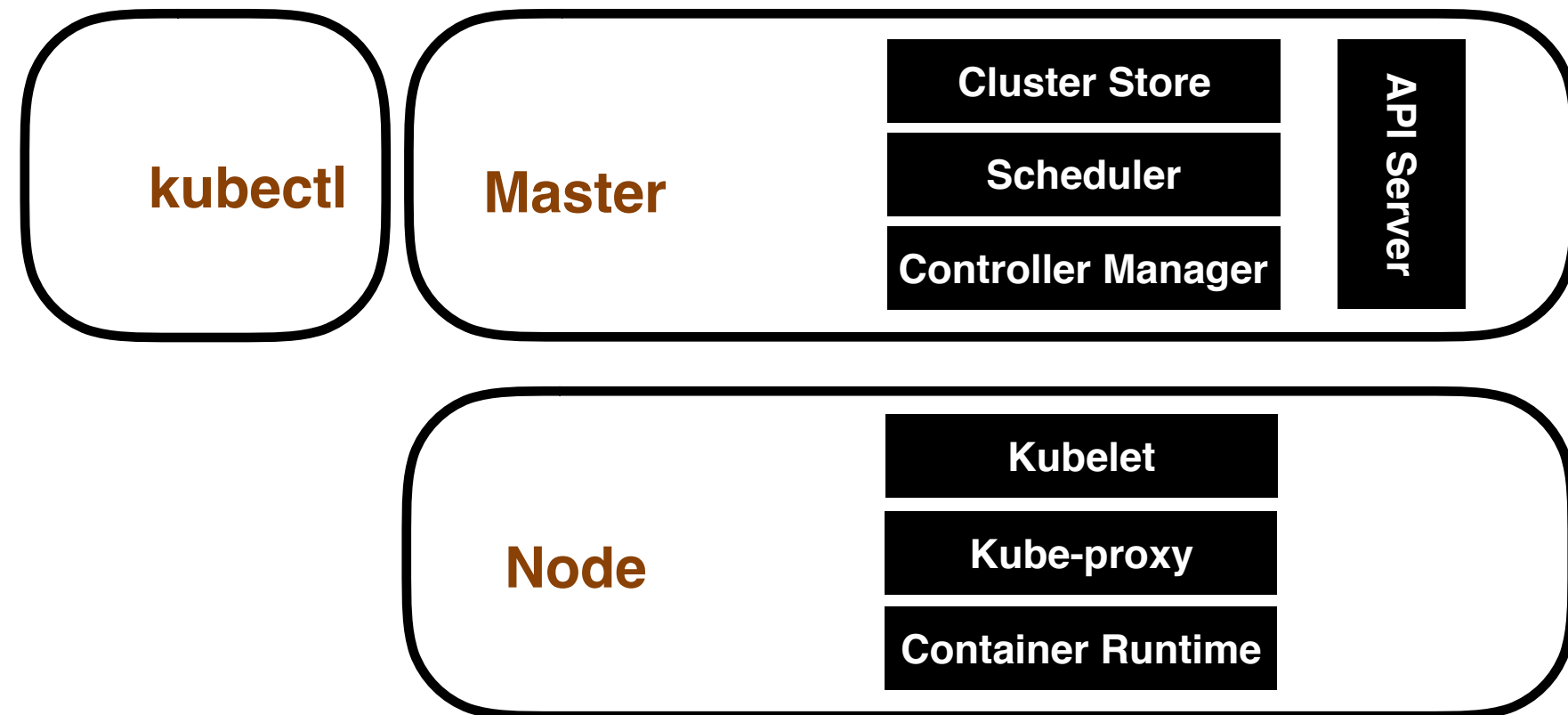
Demo!

- Deploy a stateless web app
- Deploying SQL Server in a **Deployment** with Persistent Storage
- Recovery Scenario



Demo!

- Deploy a stateless web app
- Deploying SQL Server in a **Deployment** with Persistent Storage
 - Recovery Scenario
 - Upgrading SQL Server



Building Production Ready Clusters

Building Production Ready Clusters

- Scalability - number of Nodes

Building Production Ready Clusters

- Scalability - number of Nodes
- Inter-cluster communication patterns (Network connectivity)

Building Production Ready Clusters

- Scalability - number of Nodes
- Inter-cluster communication patterns (Network connectivity)
- High Availability

Building Production Ready Clusters

- Scalability - number of Nodes
- Inter-cluster communication patterns (Network connectivity)
- High Availability
 - API Server - Load Balanced

Building Production Ready Clusters

- Scalability - number of Nodes
- Inter-cluster communication patterns (Network connectivity)
- High Availability
 - API Server - Load Balanced
 - etcd - Multiple Replicas

Building Production Ready Clusters

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

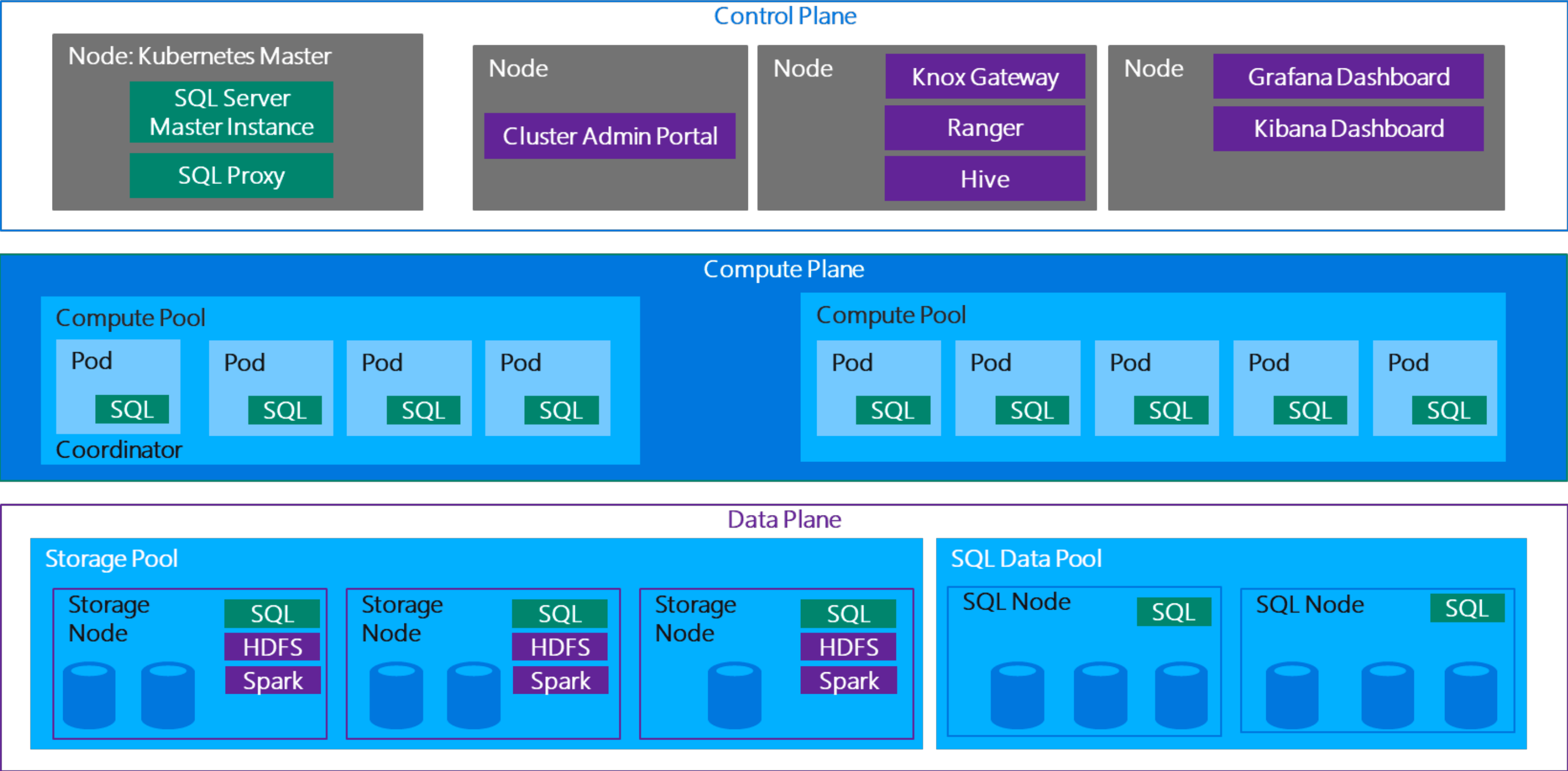
Building Production Ready Clusters

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

Building Production Ready Clusters

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

Kubernetes Cluster



From: <https://docs.microsoft.com/en-us/sql/big-data-cluster/big-data-cluster-overview?view=sqlallproducts-allversions>

Review

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

More Resources

More Resources

- **Docker for Windows/Mac**

More Resources

- **Docker for Windows/Mac**
- **Managed Service Providers**

More Resources

- **Docker for Windows/Mac**
- **Managed Service Providers**
 - Azure Kubernetes Service (**AKS**)

More Resources

- **Docker for Windows/Mac**
- **Managed Service Providers**
 - Azure Kubernetes Service (**AKS**)
 - <https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough>

More Resources

- **Docker for Windows/Mac**
- **Managed Service Providers**
 - Azure Kubernetes Service (**AKS**)
 - <https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough>
- **Elastic Container Service for Kubernetes (EKS)**

More Resources

- **Docker for Windows/Mac**
- **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/>

More Resources

- **Docker for Windows/Mac**
- **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)**

More Resources

- **Docker for Windows/Mac**
- **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/>

More Resources

- **Docker for Windows/Mac**
- **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! - <https://app.pluralsight.com/profile/author/anthony-nocentino>**

More Resources

- **Docker for Windows/Mac**
- **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! - <https://app.pluralsight.com/profile/author/anthony-nocentino>**
- **Kubernetes Installation and Configuration Fundamentals**

More Resources

- **Docker for Windows/Mac**
- **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! - <https://app.pluralsight.com/profile/author/anthony-nocentino>**
 - **Kubernetes Installation and Configuration Fundamentals**
 - **Managing the API Server and Pods**

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



Questions?

Event Feedback
(not optional!)

<http://bit.ly/DataGrillen2019Event>



Session Feedback Day 2 (not optional!)

<http://bit.ly/DataGrillen2019Day2>



Thank You!