# Inside Kubernetes Architecture Fundamentals

Anthony E. Nocentino

anocentino@purestorage.com

Anthony E. Nocentino

- **Principal Field Solution Architect @ Pure Storage**

  - Specialize in system architecture and performance

  - Masters Computer Science

- **email**: anocentino@purestorage.com

- **Twitter**: @nocentino

- **Blog**: www.nocentino.com

- **Pluralsight Author**: www.pluralsight.com
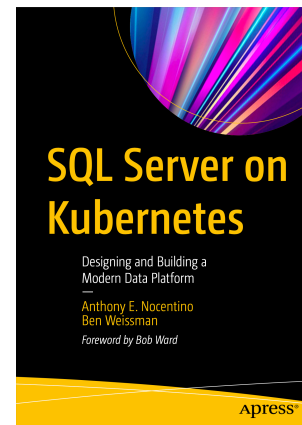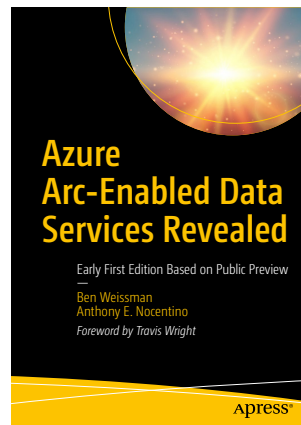
Agenda

- What is Kubernetes

- Benefits of Using Kubernetes

- Kubernetes API Objects

- Exploring Kubernetes Architecture

- Deploying Applications

- Deploying SQL Server

What is Kubernetes?

- Container Orchestrator

- Pods are Container Based Applications

- Infrastructure Abstraction

- Desired State

- Declarative Configuration in Code

Kubernetes Benefits

- Workload placement

- Managing state, starting things up and keeping things up

- Networking and Services

- Load balancing services

- Persistent storage

- Declarative model

Kubernetes Cluster

Caching

Web 1

SQL

Web 2

Cluster
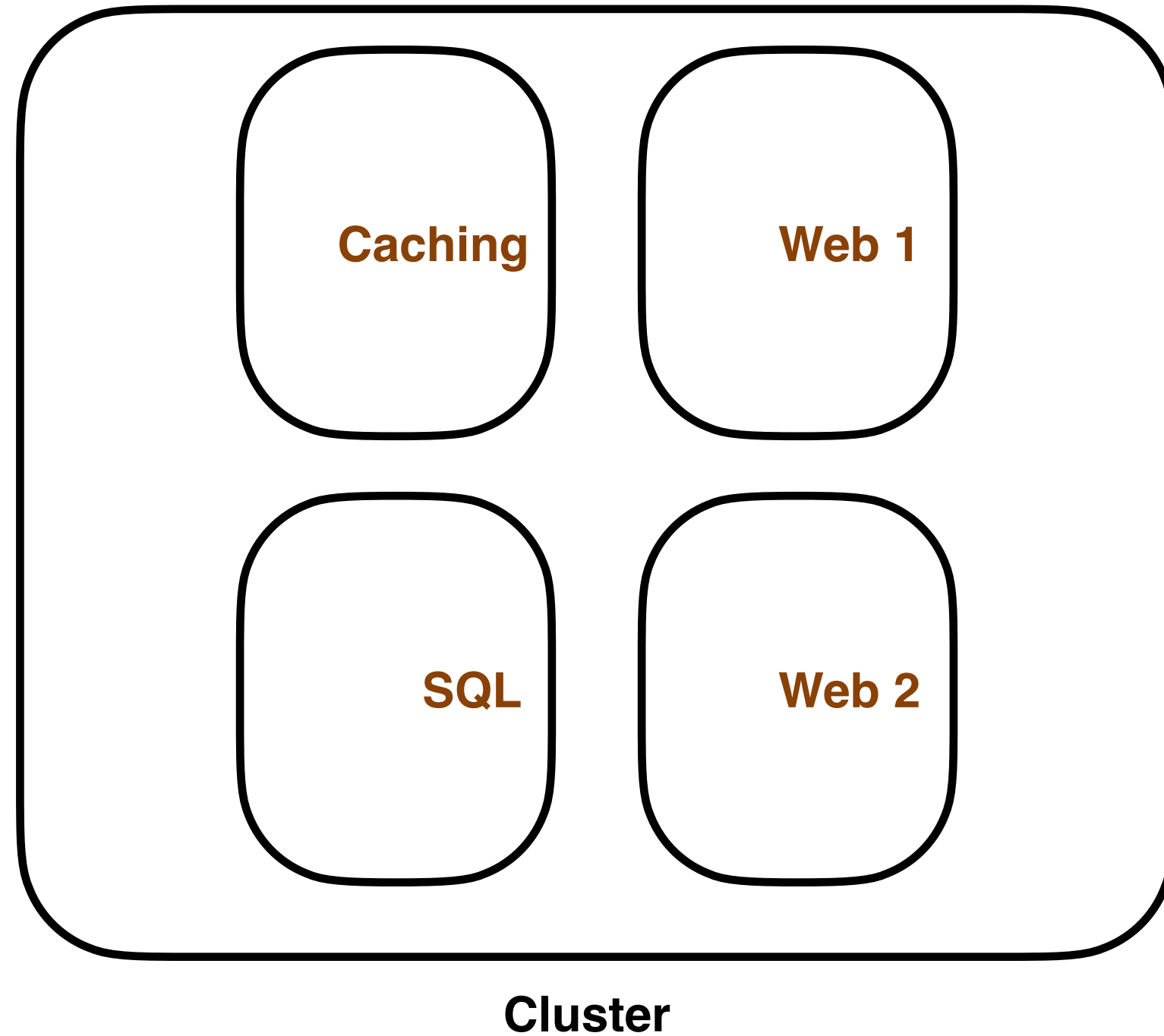
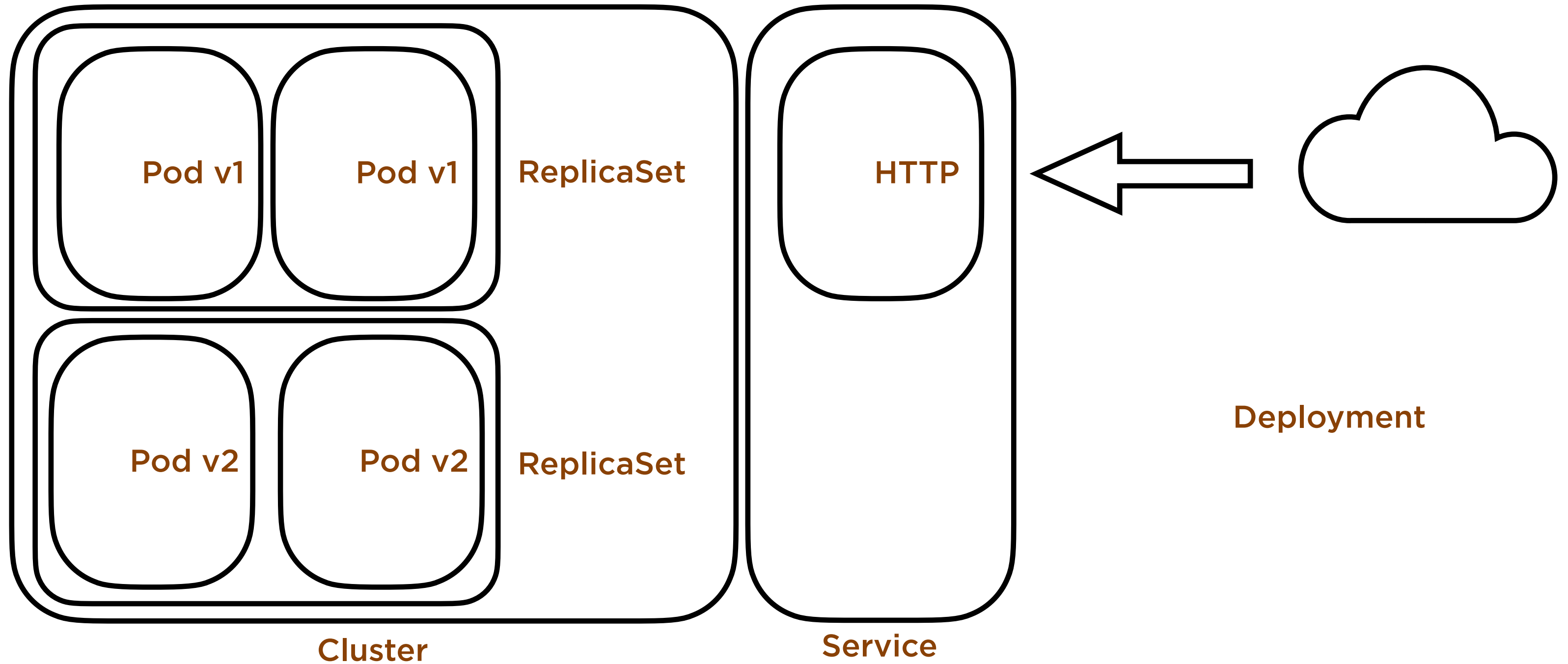# Kubernetes API

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

  - **Pods** - your container based applications

  - **Controllers** - maintain desired state

  - **Services** - persistent access to your apps

  - **Storage** - persistent storage for your data

  - …and more

Services and ReplicaSets

Pod

Pod

Pod

Pod

Pod

ReplicaSet

Cluster

HTTP

Service
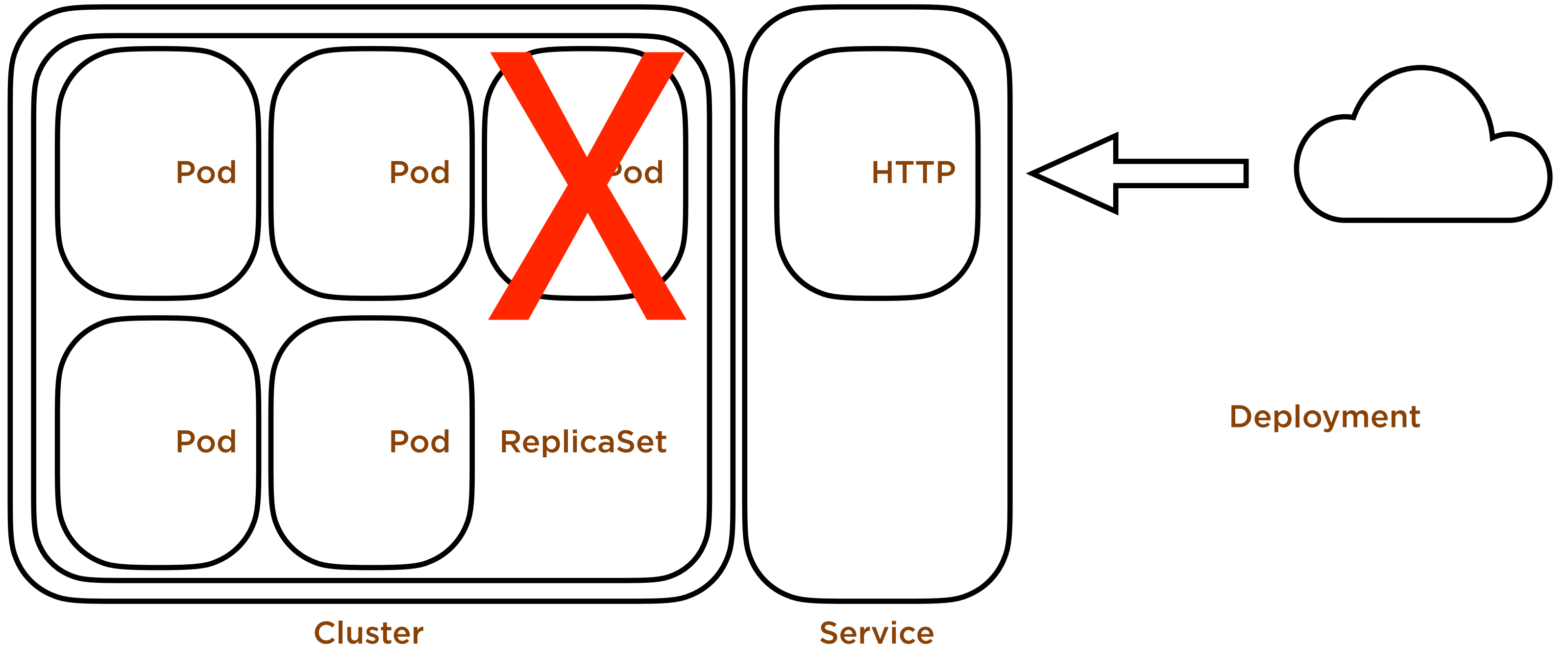
Deployment

Controller Operations - `Deployment`

Deploying Applications
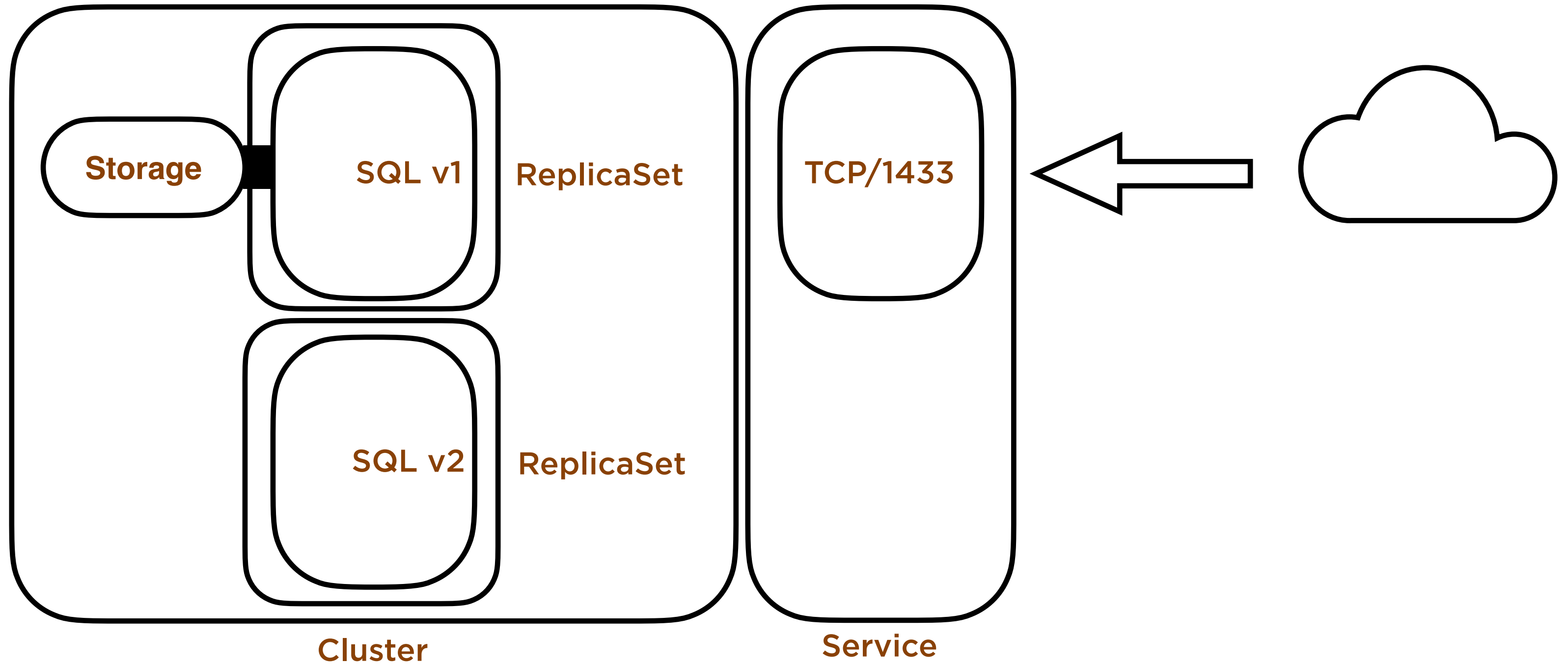
- Imperative

- Declarative

- YAML and JSON

Demo!

- Imperatively deploying a web application

- Accessing Services within a Cluster

Running SQL Server in Kubernetes

- A Pod goes back to its initial state each time it's deployed

- **State** - where do we store data?

- **Configuration** - how do we configure SQL Server?

Decoupling Data and Computation

Storage in Kubernetes

Cluster
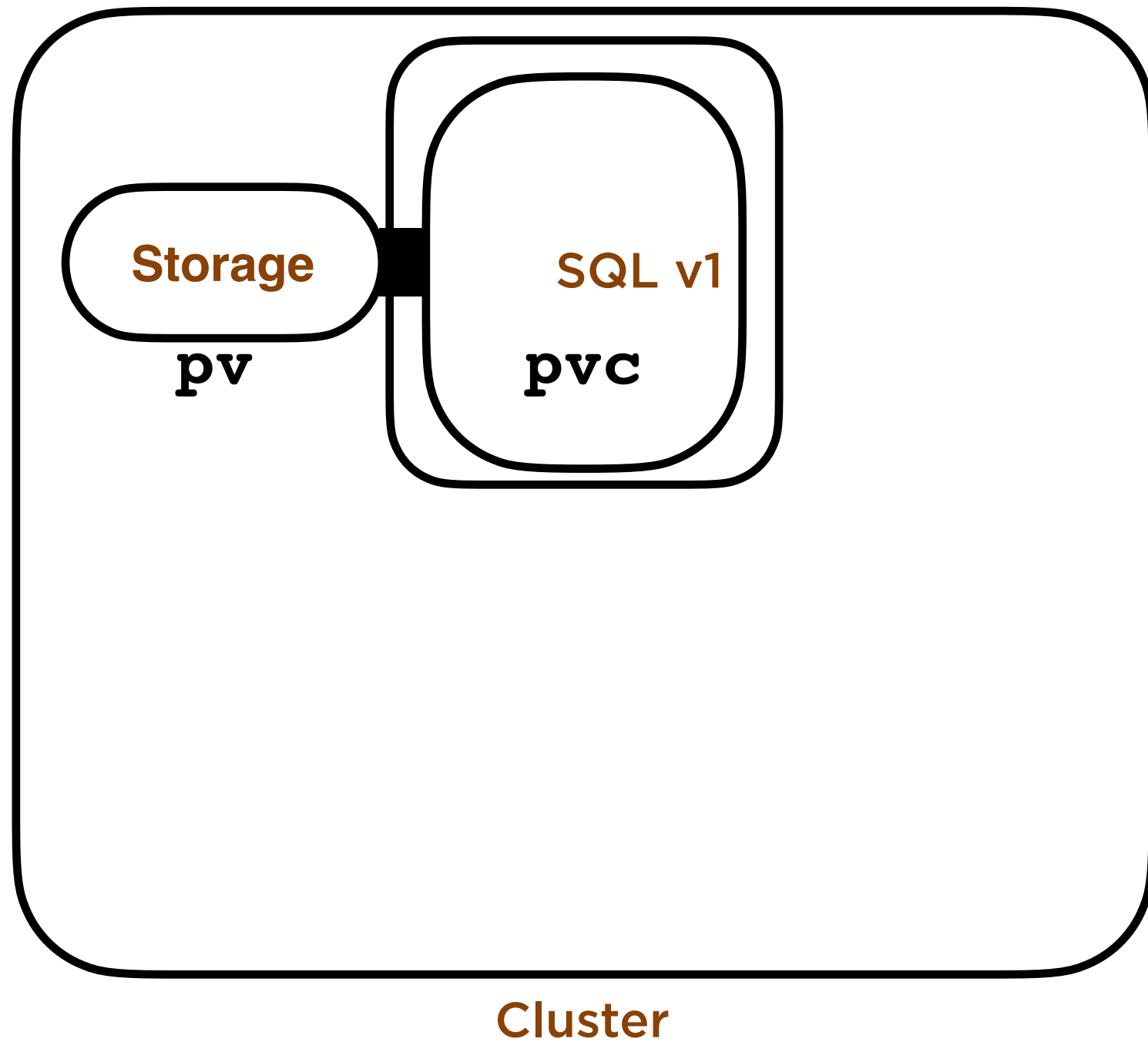
- **Persistent Volume (pv)**
  - Administrator defined storage
  - iSCSI, NFS, FC, AzureDisk…many more
- **Persistent Volume Claim (pvc)**
  - The Pod "claims" the **pvc**
  - The **pvc** is mapped to the **pv** by k8s
  - Decouples the Pod and the storage

Data Persistency in SQL Server in K8S

- Define **Persistent Volumes/Persistent Volume Claims**

  - Instance directory (error log, default trace, etc..)

    - **/var/opt/mssql/**

  - User Database default directory

    - **/var/opt/mssql/data**

SQL

/var/opt/mssql/

Storage
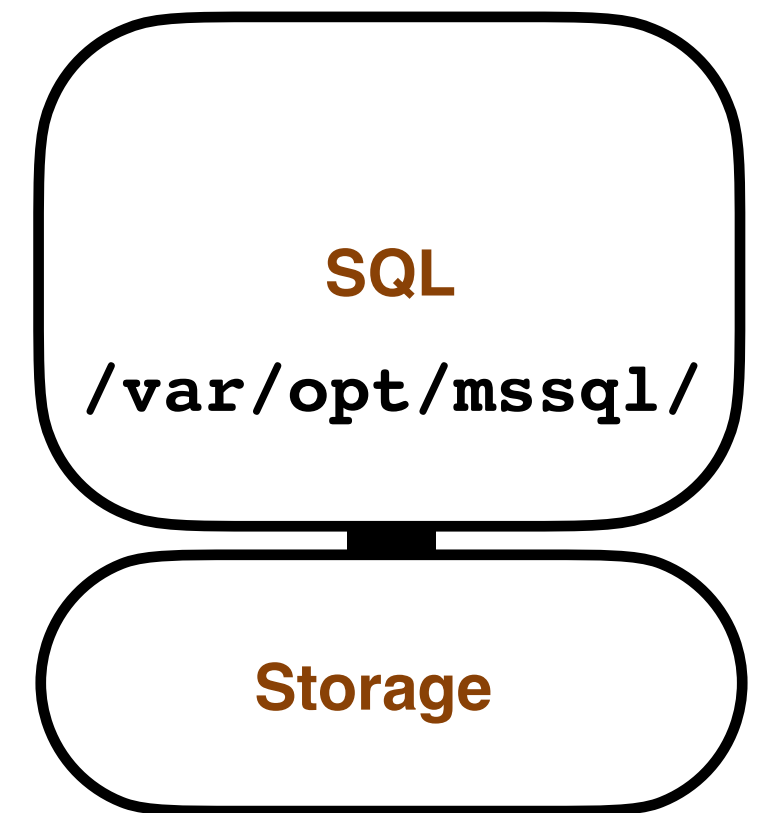
Configuring SQL Server in a Pod

- In our Pod configuration we define **Environment Variables**

  - Used at initial startup to configure the SQL Instance

    - **`ACCEPT_EULA`**

    - **`SA_PASSWORD`**

      - Stored in the cluster as a **`Secret`** (hashed, not encrypted)

  - Pods go back its initial state of the container image on creation

https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-configure-environment-variables

Define SQL Server in a Pod in YAML

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mssql-deploymen
spec:
  replicas: 1
  strategy:
    type: Recreate
  selector:
    matchLabels:
        app: mssql
  spec:
    hostname:
      sql01
    securityContext:
      fsGroup: 10001
    containers:
    - name: mssql
      image: '…/mssql/server:2019-CU9-ubuntu-18.04'
      ports:
      - containerPort: 1433
      env:
      - name: ACCEPT_EULA
        value: "Y"
      - name: SA_PASSWORD
        valueFrom:
          secretKeyRef:
            name: mssql
            key: SA_PASSWORD
      volumeMounts:
      - name: mssqldb
        mountPath: /var/opt/mssql
    volumes:
    - name: mssqldb
      persistentVolumeClaim:
        claimName: pvc-sql-data
```
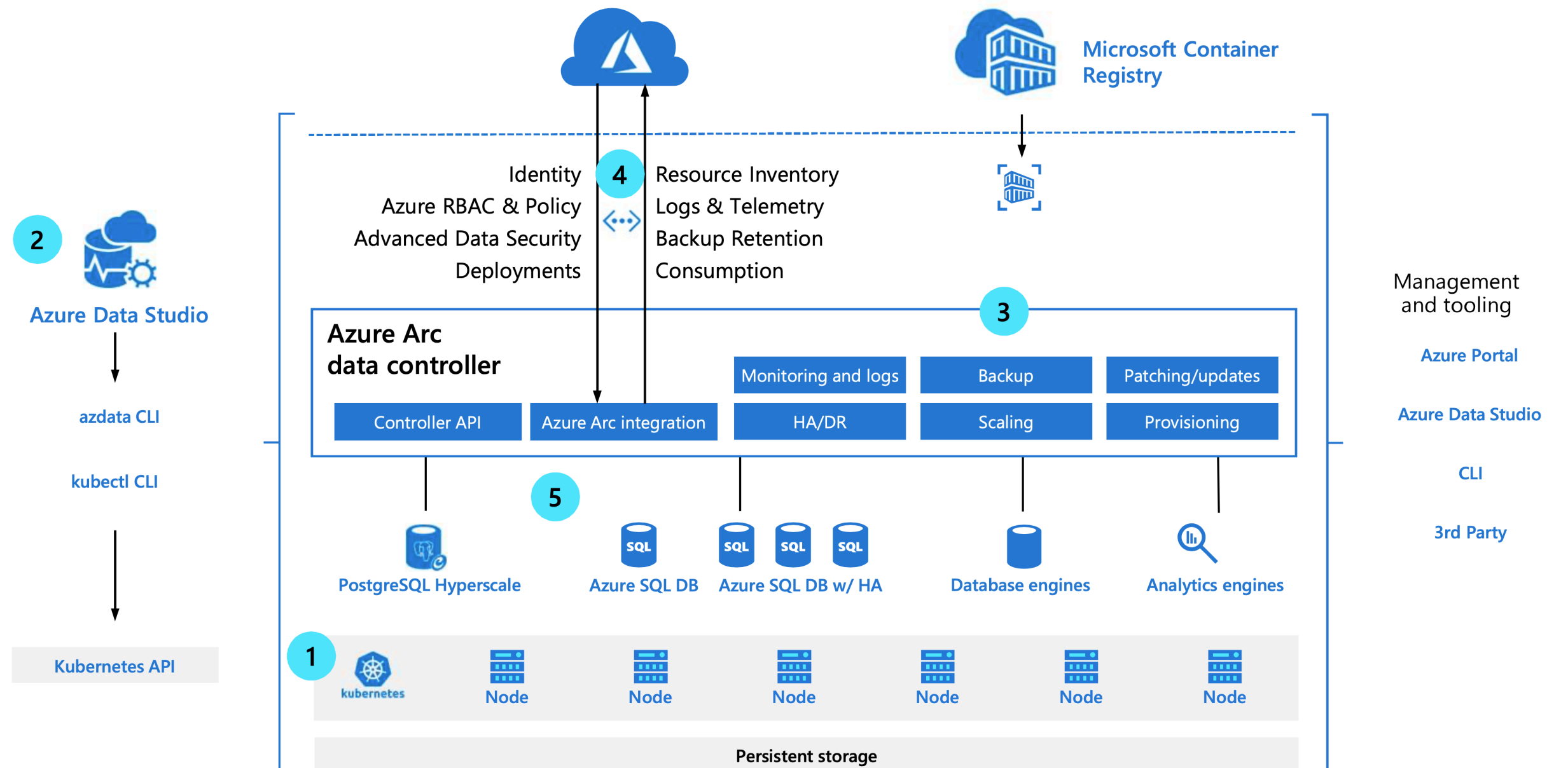
Demo

- Deploying SQL Server in a `Deployment` with Persistent Storage

  - Recovery Scenario

  - Upgrading SQL Server

Azure Arc Enabled Data Services

How it works: architecture of Azure data services on customer infrastructure

A few steps to get Azure data services in your environment:

1 Have Kubernetes on your infrastructure

2 Prepare environment with APIs and CLIs

3 Install Azure Arc data controller

4 Connect to Azure

5 Deploy and run Azure data services for your workloads

Azure Data Studio

↓

azdata CLI

kubectl CLI

↓

Kubernetes API

Microsoft Container Registry

Identity **4** Resource Inventory
Azure RBAC & Policy   Logs & Telemetry
Advanced Data Security   Backup Retention
Deployments   Consumption

Azure Arc data controller

3

| Monitoring and logs | Backup | Patching/updates |
| Controller API | Azure Arc integration | HA/DR | Scaling | Provisioning |

5

PostgreSQL Hyperscale   Azure SQL DB   Azure SQL DB w/ HA   Database engines   Analytics engines

1 kubernetes   Node   Node   Node   Node   Node   Node

Persistent storage

Management and tooling

Azure Portal
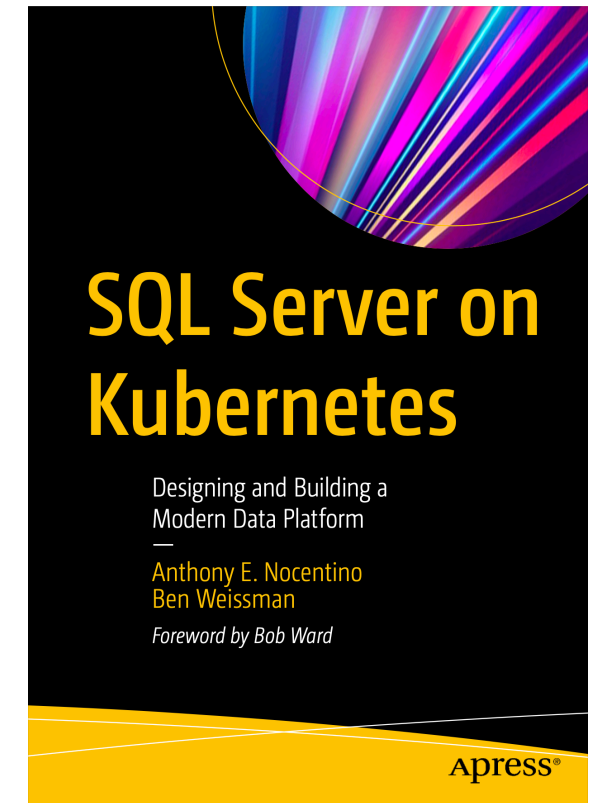
Azure Data Studio

CLI

3rd Party

Review

- What is Kubernetes

- Benefits of Using Kubernetes

- Kubernetes API Objects

- Exploring Kubernetes Architecture

- Deploying Applications

- Deploying SQL Server

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

SQL Server on Kubernetes

Designing and Building a
Modern Data Platform

Anthony E. Nocentino
Ben Weissman

Foreword by Bob Ward

Apress®

Need more data?

- **Contact me!**
  - **email:** anocentino@purestorage.com
  - **Twitter:** @nocentino

- **Blog**
  - www.nocentino.com   https://github.com/nocentino/presentations

- **Pluralsight**
  - Linux
  - Kubernetes
  - Azure
  - Hit me up to get free access to this content

Thank You!