Microsoft

Module 2 - Kubernetes

Kubernetes Tooling

Microsoft Services

Module Overview

- Walkthrough of Azure Bridge to Kubernetes
- Kubernetes Toolset Overview
  - Visual Studio
  - Visual Studio Code
  - Minikube
- Working with HELM
- Understanding the role of Init Containers

Kubernetes Dev Tools

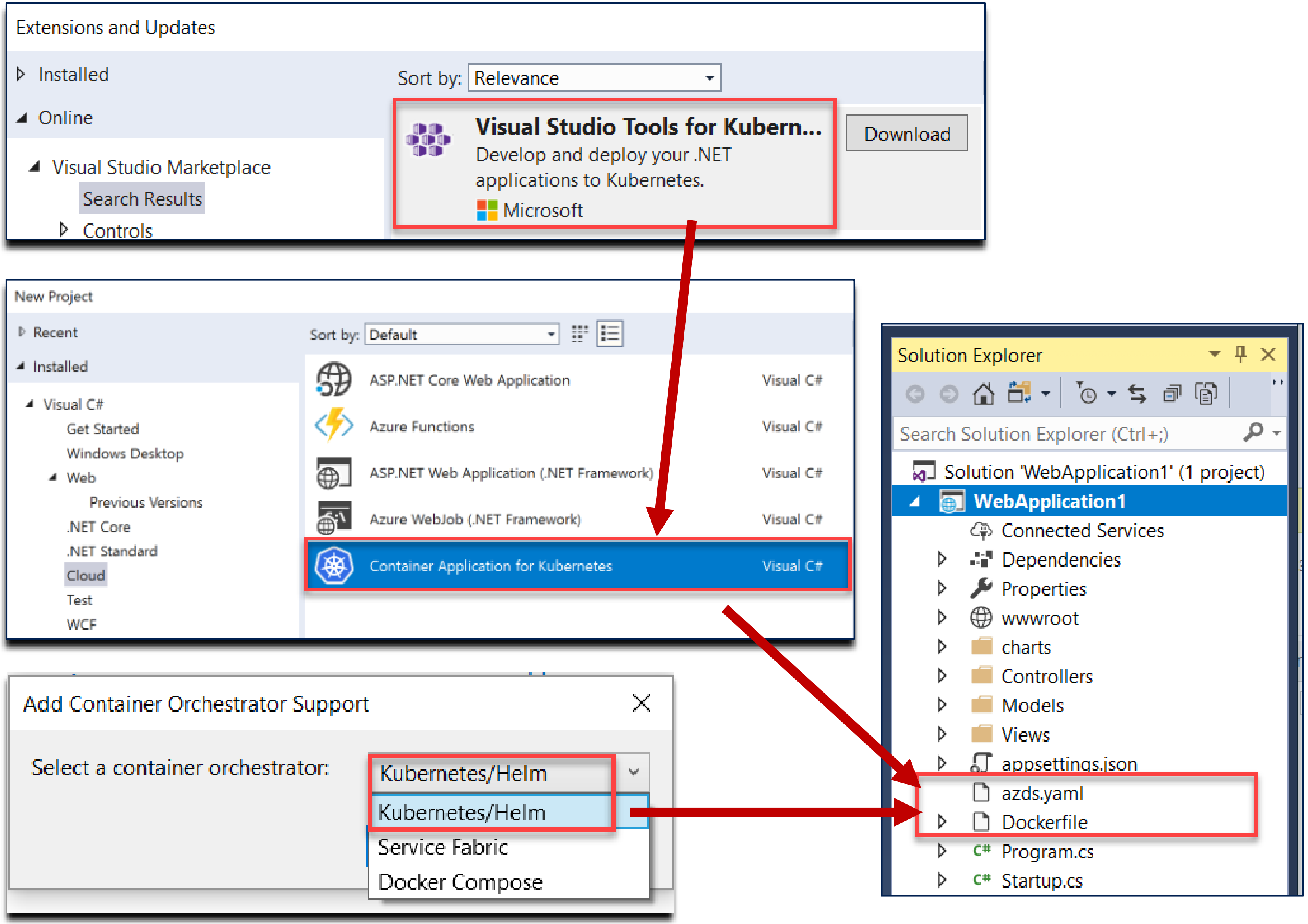
- Kubernetes has a large community supporting various toolset.
  - Visual Studio – Kubernetes templates
  - VS Code - Kubernetes extensions.
  - Helm - Package manager for Kubernetes.
  - Minikube - Single node cluster running Kubernetes.
  - Kubectl - CLI to run command against Kubernetes cluster.
  - Many more..

Visual Studio: Kubernetes Tools

- Install the Visual Studio Tools for Kubernetes extension

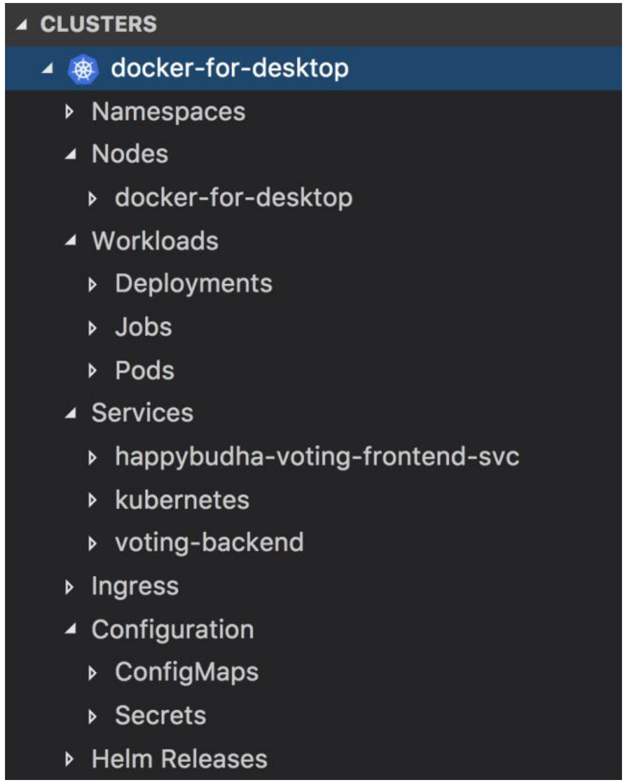
- Adds template that generates Dockerfile and Helm Chart

- Or, can directly add Kubernetes support to existing Core app


Extensions and Updates
Installed
Online
Visual Studio Marketplace
Search Results
Controls
Sort by: Relevance
Visual Studio Tools for Kubern...
Develop and deploy your .NET applications to Kubernetes.
Microsoft
Download
New Project
Recent
Installed
Visual C#
Get Started
Windows Desktop
Web
Previous Versions
.NET Core
.NET Standard
Cloud
Test
WCF
Sort by: Default
ASP.NET Core Web Application Visual C#
Azure Functions Visual C#
ASP.NET Web Application (.NET Framework) Visual C#
Azure WebJob (.NET Framework) Visual C#
Container Application for Kubernetes Visual C#
Add Container Orchestrator Support
Select a container orchestrator:
Kubernetes/Helm
Kubernetes/Helm
Service Fabric
Docker Compose
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'WebApplication1' (1 project)
WebApplication1
Connected Services
Dependencies
Properties
wwwroot
charts
Controllers
Models
Views
appsettings.json
azds.yaml
Dockerfile
Program.cs
Startup.cs

Visual Studio Code: Kubernetes Tools

- A Visual Studio Code extension - available in Preview Mode
- Helps you…
  - Install Helm, Draft and Minikube
  - Create a new or connect to an existing Kubernetes cluster
  - Deploy an app to Azure Kubernetes Service
  - Track the status of your deployment
  - Load a resource from the Kubernetes API
  - Show logs
  - Show events



```
◢ CLUSTERS
  ◢ docker-for-desktop
    ▹ Namespaces
    ◢ Nodes
      ▹ docker-for-desktop
    ◢ Workloads
      ▹ Deployments
      ▹ Jobs
      ▹ Pods
    ◢ Services
      ▹ happybudha-voting-frontend-svc
      ▹ kubernetes
      ▹ voting-backend
    ▹ Ingress
    ◢ Configuration
      ▹ ConfigMaps
      ▹ Secrets
    ▹ Helm Releases
```

Demo:

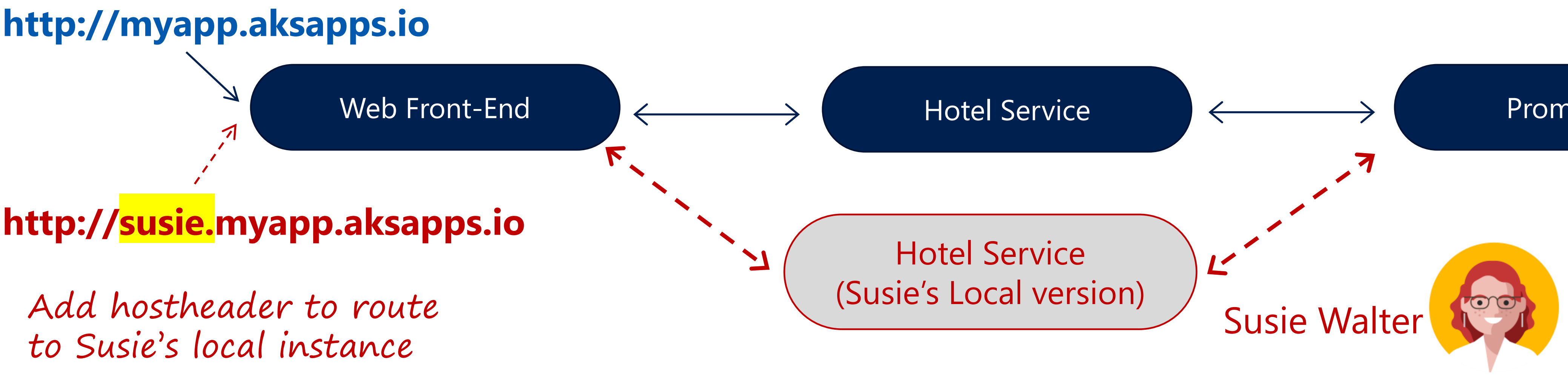
Visual Studio 2019

Show Kubernetes Tooling

Bridge to Kubernetes (in preview)

Tooling to enable team collaboration across microservice applications

- The team shares running (development) instance in an AKS cluster
- Developers use personal “spaces” on their machine to locally develop their services
- But, test from end-to-end in the AKS cluster - without replicating dependencies
- Enables teams to work with large microservice applications and rapidly iterate and debug code directly in Kubernetes using Visual Studio 2019 or Visual Studio Code



http://myapp.aksapps.io

Web Front-End

Hotel Service

Promotions

http://susie.myapp.aksapps.io

Hotel Service
(Susie’s Local version)

Susie Walter

*Add hostheader to route
to Susie’s local instance*

Demo: Bridge to Kubernetes

Bridge to Kubernetes Walkthrough

Minikube

- Runs a single-node Kubernetes cluster inside a VM on your local machine
- Develop locally with Kubernetes
- Minikube supports Kubernetes features such as:
  - Services, Deployment, Pods
  - NodePorts
  - ConfigMaps and Secrets
  - Dashboards
  - Container Runtime: Docker, rkt and CRI-O
  - Enabling CNI (Container Network Interface)
  - Ingress

Demo:

Minikube

Show Minikube

Helm

The package manager for Kubernetes

Why Helm?

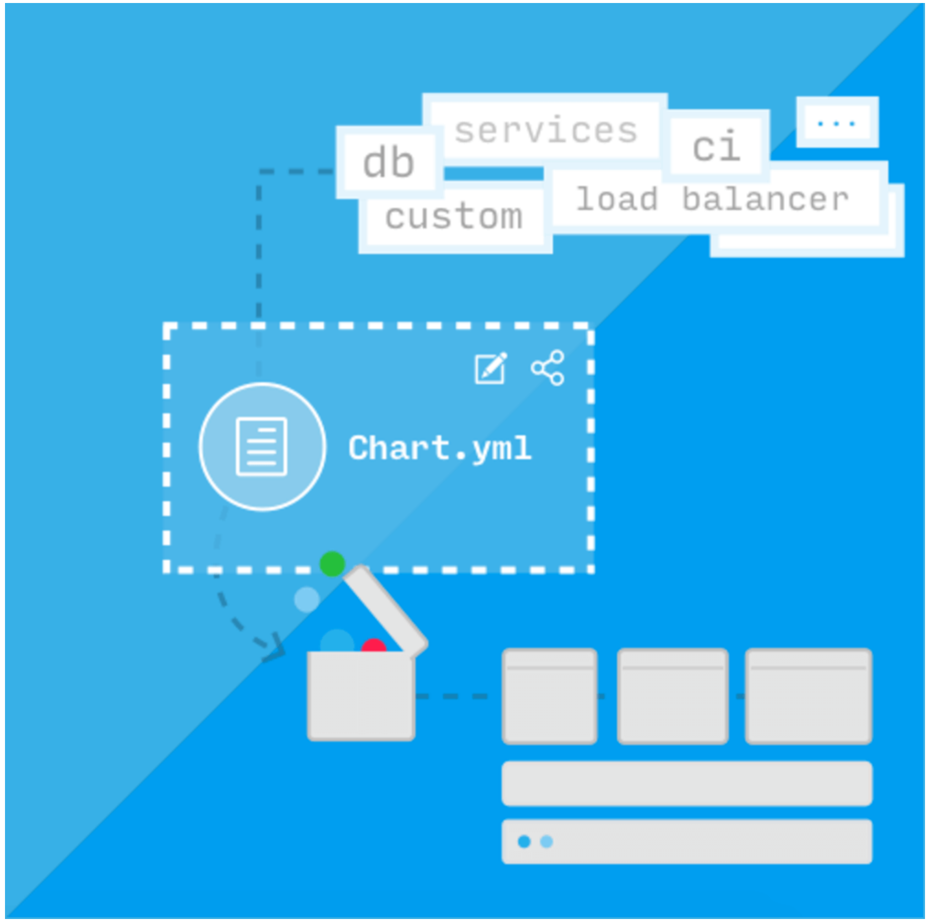
- Manage Complexity
- Easy Updates
- Simple Sharing
- Rollbacks

Helm

- Package manager for Kubernetes-based applications
- Streamlines installing/managing applications
  - Perform consistent releases without dealing with dozens of files
  - Update/rollback releases - as needed
- Package and preconfigure Kubernetes resources into *Charts*
  - Manage and share your own application chart packages
  - Leverage other popular software packaged as charts
  - Create reproducible builds of Kubernetes applications
  - Manage you Kubernetes manifest files

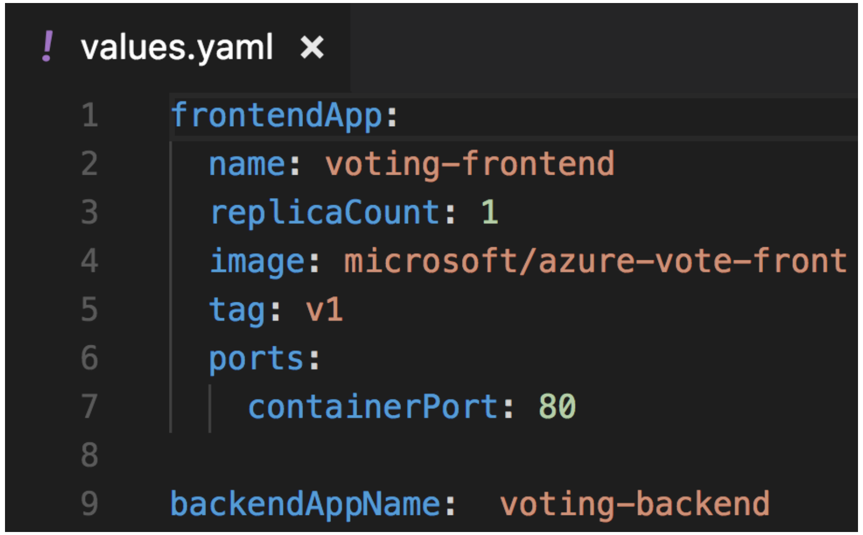
Helm | Charts

- Helm packages are called *Charts*
- *Chart.yaml* describes the chart
- Charts contains one or more templates
- Charts can be persisted on disk, or pulled from remote chart repositories (e.g. GitHub, private blob on Azure)
- Azure Container Registry can be used as a Helm chart repository
  - Private, secure, integrates with build pipelines
  - Provides geo-replication



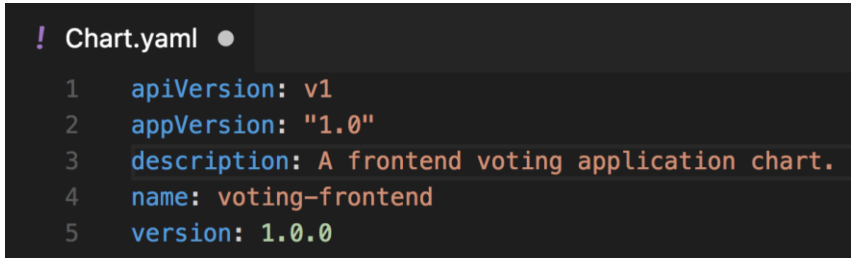
Helm | Voting Application Charts

- Voting application has frontend and backend components



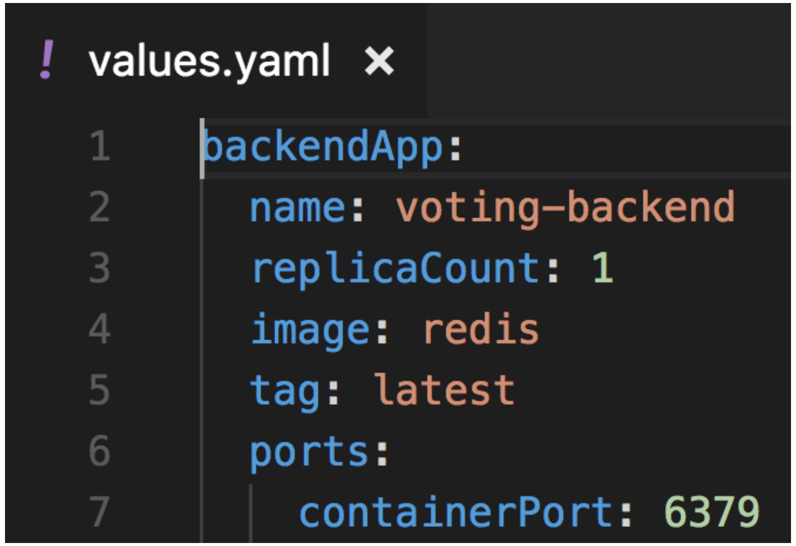
```yaml
frontendApp:
  name: voting-frontend
  replicaCount: 1
  image: microsoft/azure-vote-front
  tag: v1
  ports:
    containerPort: 80

backendAppName:  voting-backend
```

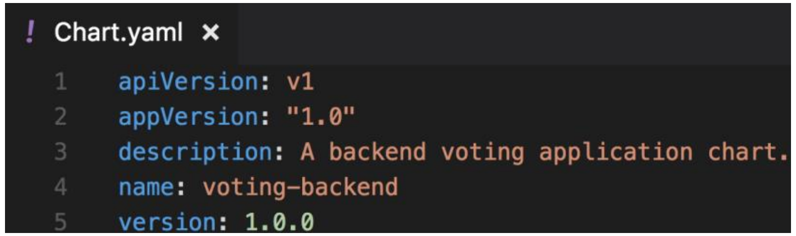


```yaml
apiVersion: v1
appVersion: "1.0"
description: A frontend voting application chart.
name: voting-frontend
version: 1.0.0
```

Frontend application chart and values YAML file



```yaml
backendApp:
  name: voting-backend
  replicaCount: 1
  image: redis
  tag: latest
  ports:
    containerPort: 6379
```

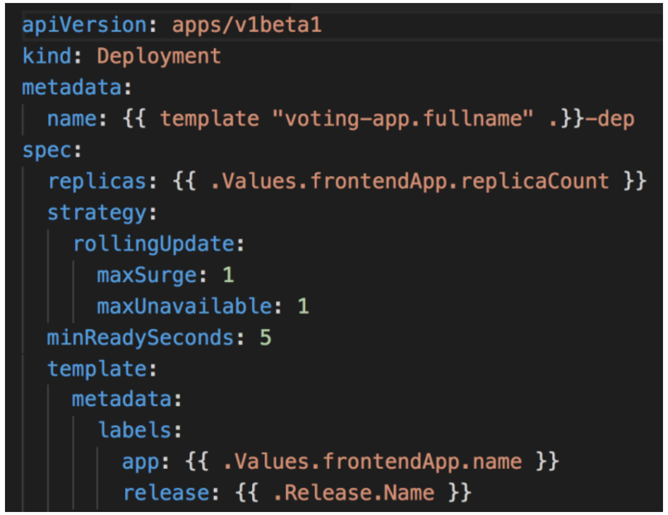


```yaml
apiVersion: v1
appVersion: "1.0"
description: A backend voting application chart.
name: voting-backend
version: 1.0.0
```

Backend application chart and values YAML file

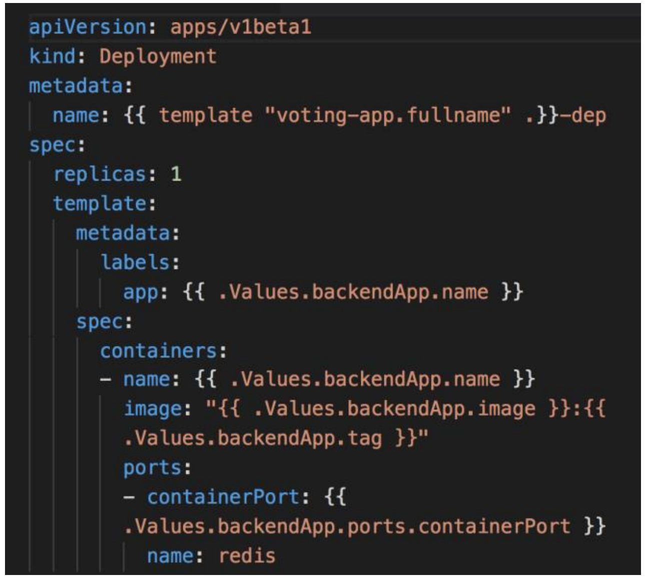
Helm | Voting Application Charts

- Kubernetes Deployments and Services references values from values.yaml file, which store default values
- { .Values.[Key] } resolves to a particular key-value entry in the relevant values.yaml file
- Templates – Functions that implements data-driven templates for generating textual output:
- *{ quote .Values.frontend.replicaCount}* will result in output *replicas: "2"* instead of *replicas: 2*



```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: {{ template "voting-app.fullname" .}}-dep
spec:
  replicas: {{ .Values.frontendApp.replicaCount }}
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  minReadySeconds: 5
  template:
    metadata:
      labels:
        app: {{ .Values.frontendApp.name }}
        release: {{ .Release.Name }}
```

Frontend application deployment definition in YAML



```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: {{ template "voting-app.fullname" .}}-dep
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: {{ .Values.backendApp.name }}
    spec:
      containers:
      - name: {{ .Values.backendApp.name }}
        image: "{{ .Values.backendApp.image }}:{{
        .Values.backendApp.tag }}"
        ports:
        - containerPort: {{
        .Values.backendApp.ports.containerPort }}
          name: redis
```

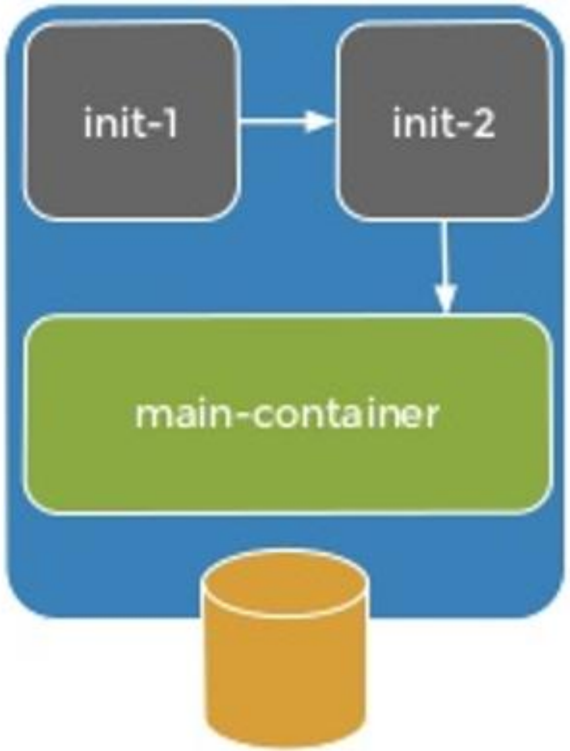
Backend application deployment definition in YAML

Demo: Helm

Installing multi-container voting application using Helm

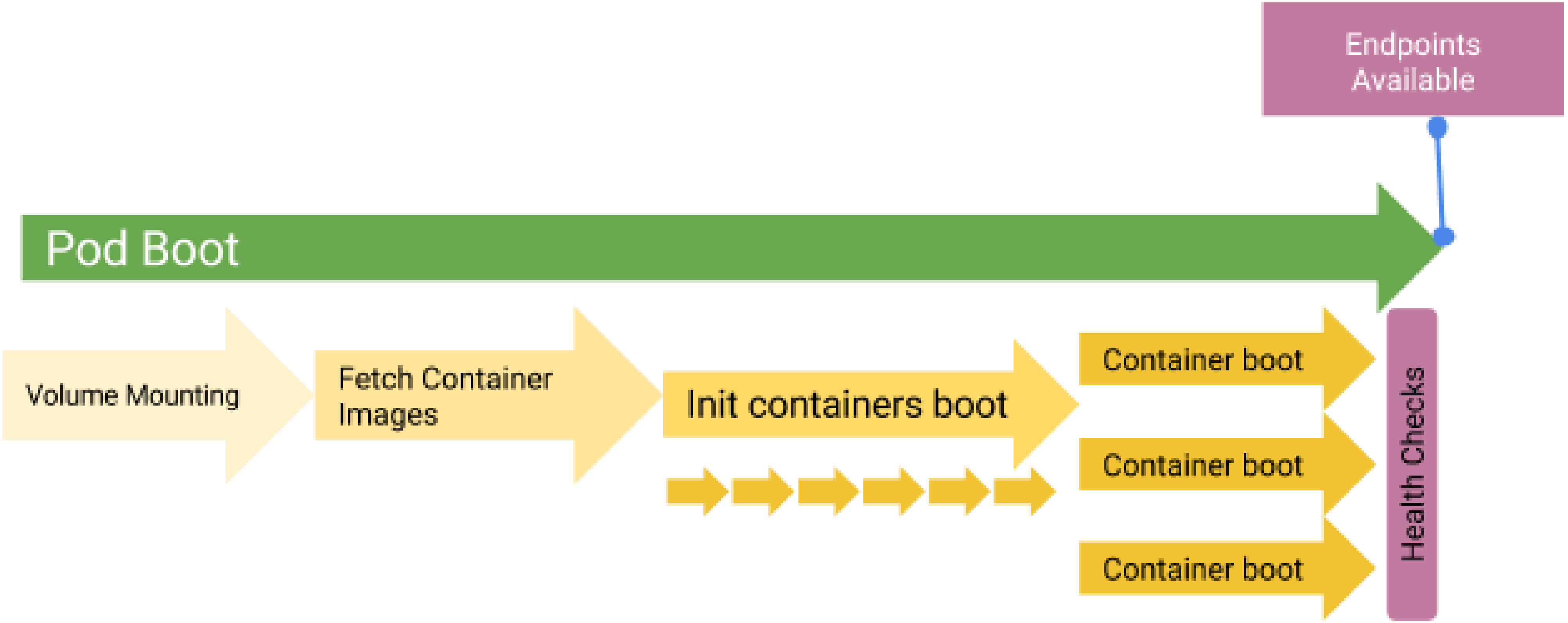
Init Containers

- Similar to app containers, but used for start-up related code…
  - Run utilities or custom code not included in app image
  - Block or delay app containers until some set of preconditions are met
- All Init Containers must to run to completion before the app container can start
  - A Pod can have multiple Init Containers
  - Each executes sequentially
  - Each must complete successfully before the next one can start



Container Start-Up Process

- Here is a more detailed view of the container start-up process…



Demo: Init Containers

Delay app container start using Init Container

Module Summary

- Many tools support Kubernetes development
  - Visual Studio
  - Visual Studio Code
  - Minikube
- HELM is a package manager for Kubernetes deployments
- Init Containers enable preprocessing before app containers start
- Bridge to Kubernetes supports microservice team development in Kubernetes

Lab: Module 5

Kubernetes Application Development & Toolset