AKS – Azure Kubernetes Services with Visual Studio Connected development environments

Jorge Arteiro

@jorgearteiro

jorge@Azuretar.com

https://www.slideshare.net/JorgeArteiro

@Azuretar

whoami

- Cloud developer/architect, DevOps, integration and tech pre-sales
- Working with Docker, Kubernetes, microservices and API management
- Speaker - Container Camp, Microsoft Ignite, API Days, NDC Sydney, ….
- Azure MVP
- Containers and DevOps community member
- Developed for Linux in early 2000s
- Current: Application DevOps Consultant for Kubernetes at NBN Co

Agenda

- AKS – Azure Kubernetes Services
  - AKS Architecture
  - Create your cluster
  - Current CI/CD process
- VSCE - Visual Studio Connected Environments
  - Developer Experience
  - Main Commands
 - End-to-end scenarios
  - Developer's Spaces
  - Demo

Install Client Tools

- Docker for Windows/Mac (Including local single node kubernetes)
- Helm https://helm.sh/
- Azure CLI version 2.31 or later
- VSCE – Visual Studio Connected Environments
  - For Windows: https://aka.ms/get-vsce-windows
  - For Mac: curl -L https://aka.ms/get-vsce-mac | bash
- Visual Studio Code https://code.visualstudio.com/Download

  Extension Download: https://aka.ms/get-vsce-code

  Install Extension: code --install-extension path-to-downloaded-extension/vsce-0.1.1.vsix
- Visual Studio latest (on Windows only)

  Extension Download: https://aka.ms/get-vsce-visualstudio
- WSL – Windows Subsystem for Linux (on Windows only)
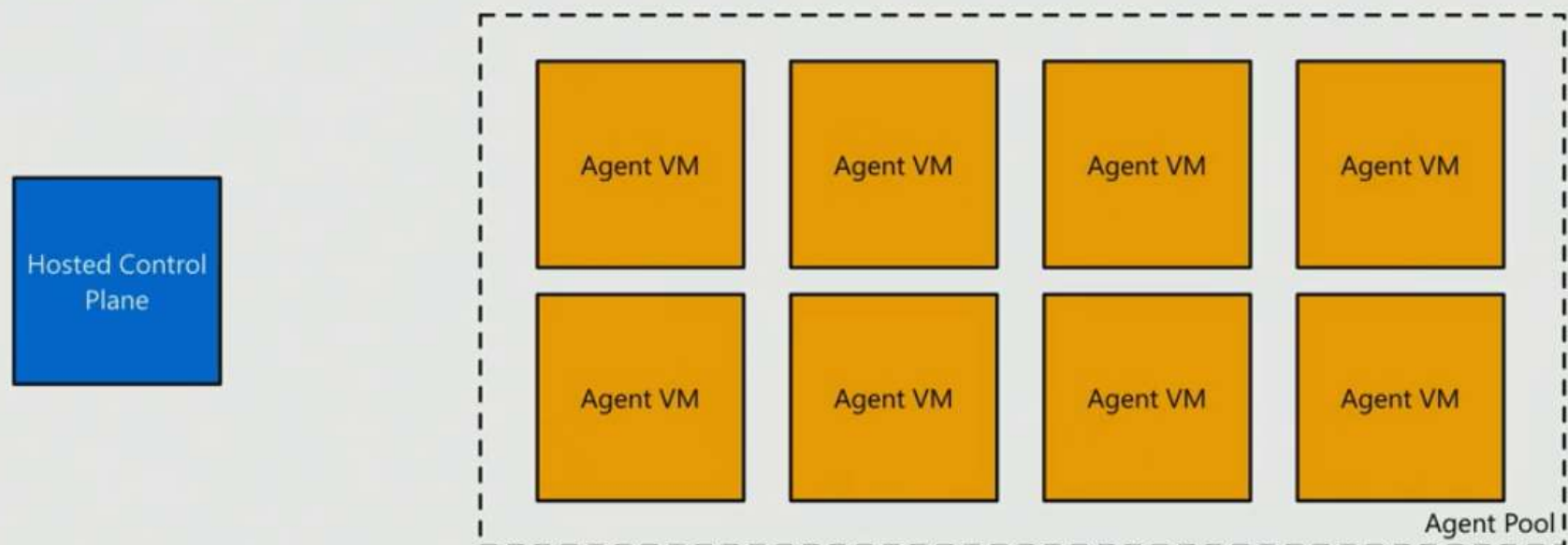
@Azuretar

Kubernetes without AKS
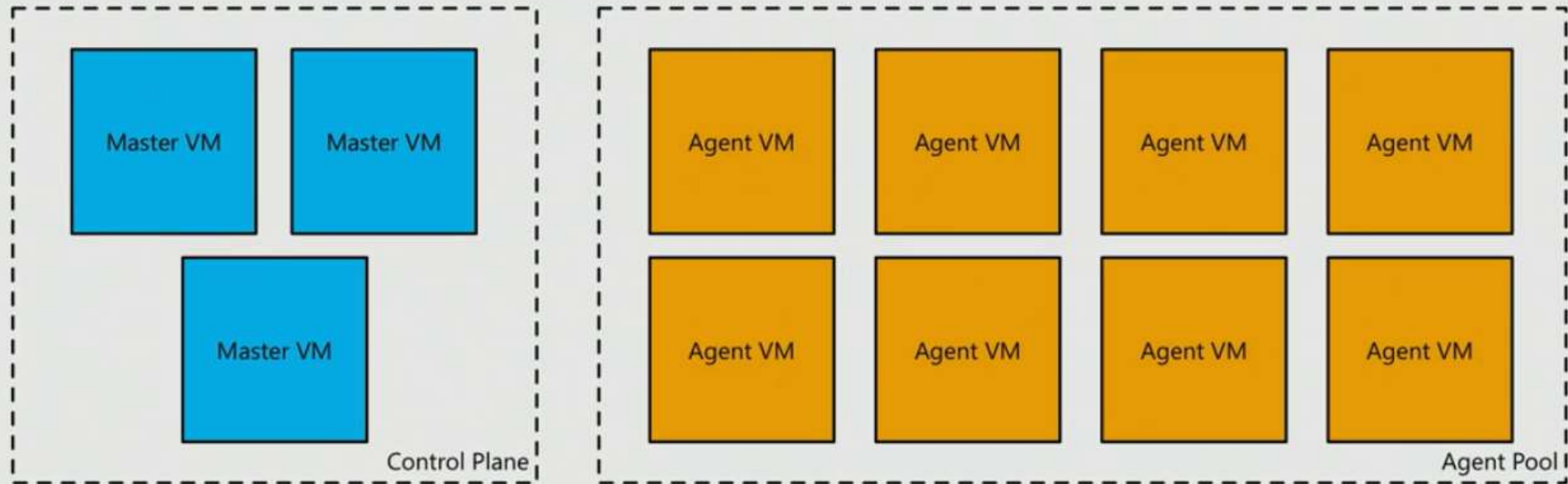
- ACS - Azure Container Services with ACS-engine open source ARM templates
- Integrated with Azure services (VNet, Load balancers, storage, Public IPs, etc)
- VMs managed by Customer
- Customer will migrate to AKS when public available

@Azuretar

Kubernetes with AKS

- AKS - Azure Kubernetes Services (Upstream Kubernetes)
- Integrated with Azure services (VNet, Load balancers, storage, Public IPs, etc)
- Fully managed cluster with free masters
- Cluster version updates managed by platform without outage

@Azuretar

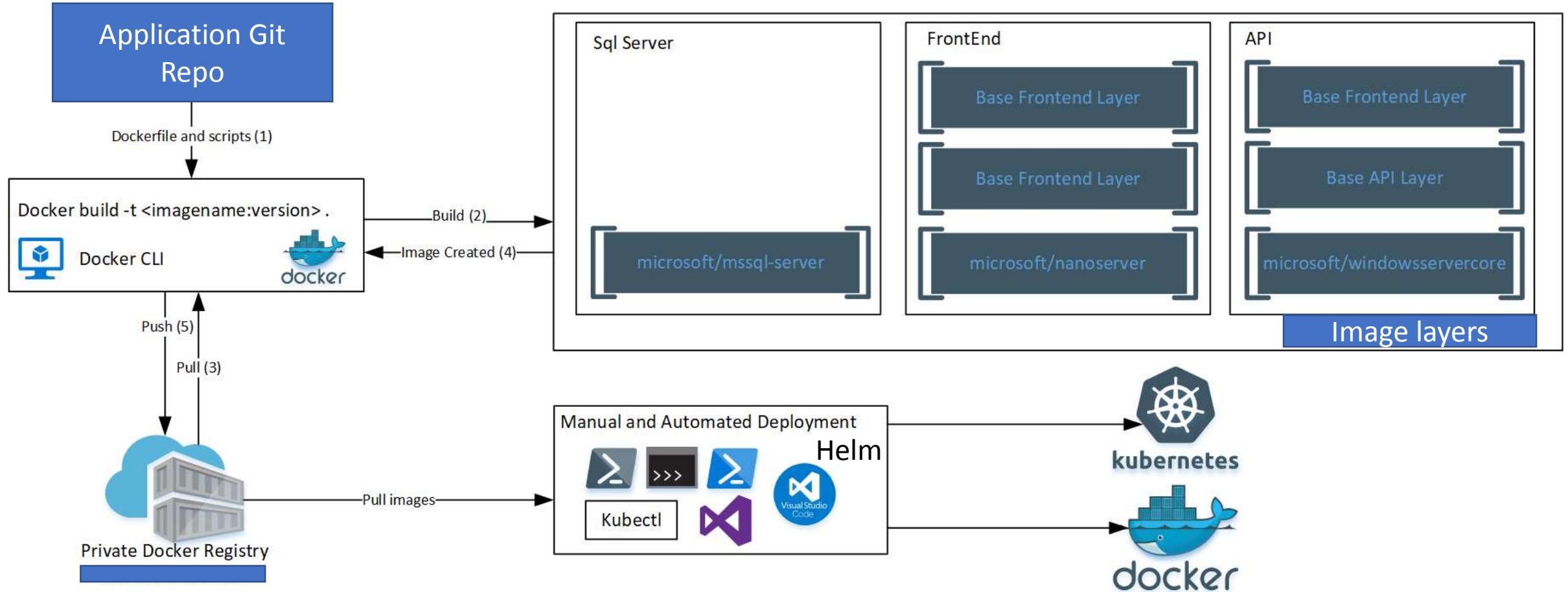# Virtual Kubelet – Serverless Kubernetes

ACI – Azure Container Instance
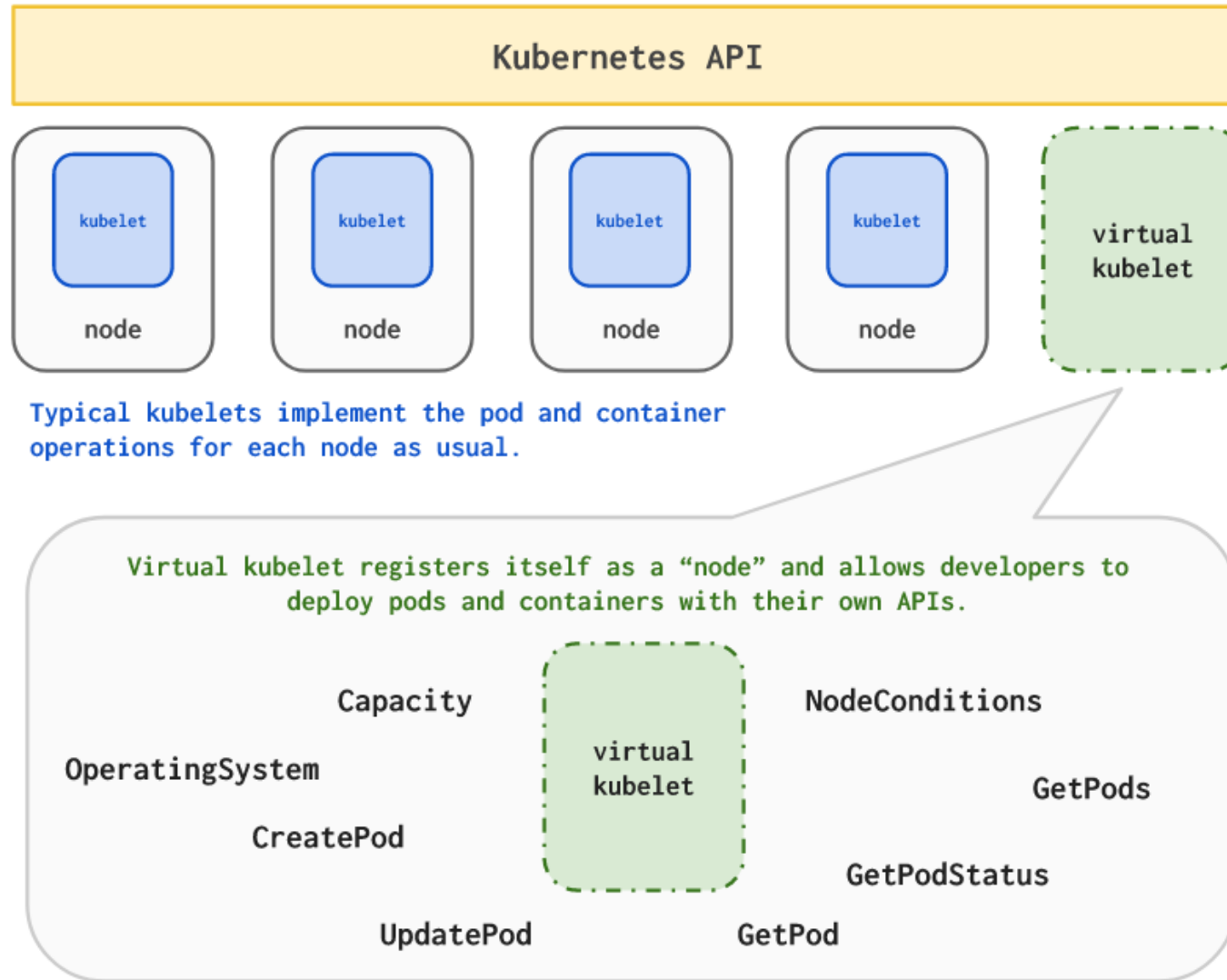
Create AKS Cluster

- az group create --name aznightk8s --location westeurope
- az aks create –g aznightk8s –n aznight1 –c 1 --generate-ssh-keys
- az aks get-credentials –g aznightk8s -n aznight1
- kubectl config use-context aznight1
- az aks browse -g aznightk8s -n aznight1
- az aks get-versions -n aznight1 -g aznightk8s --output table
- az aks upgrade -n aznight1 -g aznightk8s --kubernetes-version 1.9.2
- az aks scale -n aznight1 -g aznightk8s -c 2
- az aks install-connector -n aznight1 -g aznightk8s --connector-name aci-connector --os-type Both

https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough

@Azuretar

Current CI/CD Process

Helm as your Package Management

@Azuretar

Visual Studio Connected Environments
VSCE

Developer Experience

- Run or debug code prior to committing in a production-like development environment running in Azure

- Work with real upstream and downstream dependencies instead of mocks or simulators

- Use a shared team environment with isolated spaces to avoid complex setup and enable live, ongoing discovery of integration issues

VSCE Main Commands

- vsce **init** - prepares a directory of source code for connected development
  - VSCE.yaml
  - Helm Charts
  - Dockerfile

- vsce **up** - starts a connected development session against a target workload

- vsce **exec** - runs a one-off command inside an existing target workload
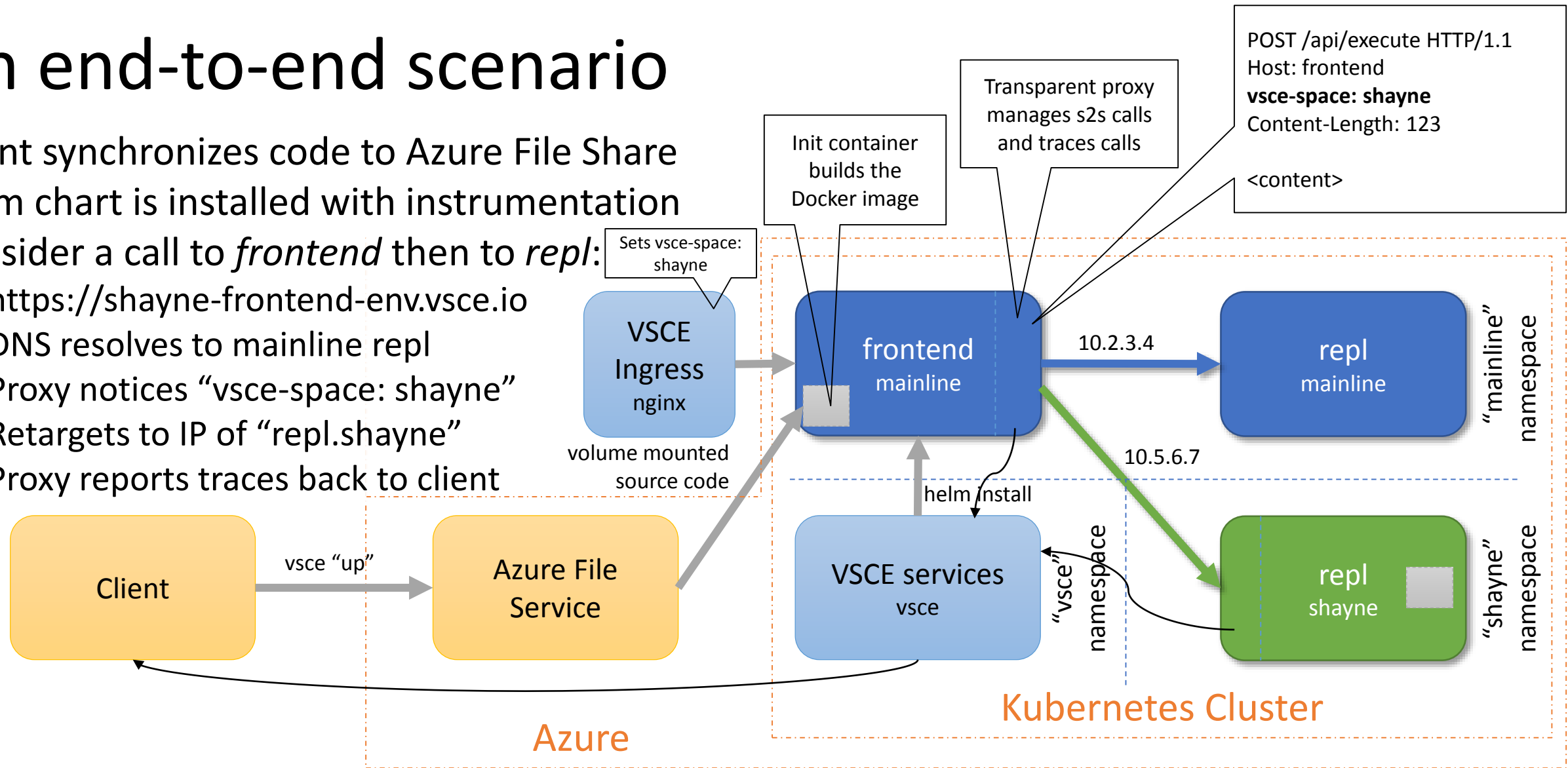
An end-to-end scenario
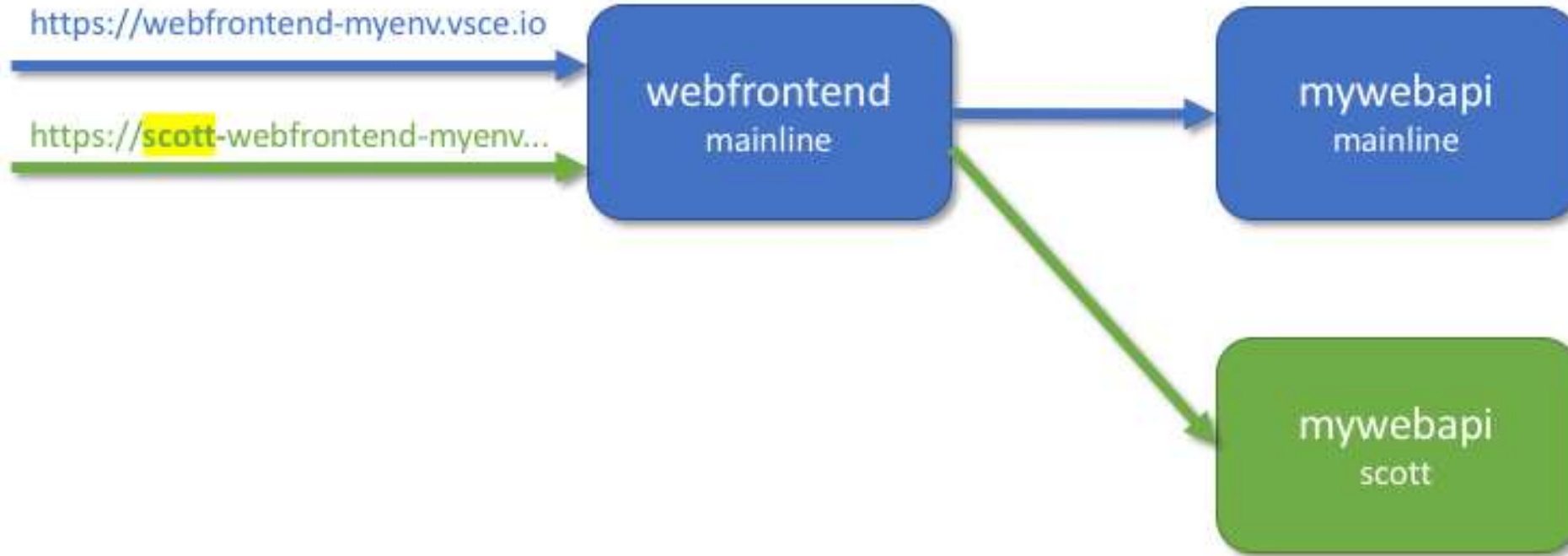
Client synchronizes code to Azure File Share

Helm chart is installed with instrumentation

Consider a call to *frontend* then to *repl*:

- https://shayne-frontend-env.vsce.io
- DNS resolves to mainline repl
- Proxy notices "vsce-space: shayne"
- Retargets to IP of "repl.shayne"
- Proxy reports traces back to client

Init container builds the Docker image

Transparent proxy manages s2s calls and traces calls

POST /api/execute HTTP/1.1
Host: frontend
vsce-space: shayne
Content-Length: 123

<content>

Sets vsce-space: shayne

VSCE Ingress nginx

volume mounted source code

frontend mainline

10.2.3.4

repl mainline

"mainline" namespace

helm install

10.5.6.7

Client

vsce "up"

Azure File Service

VSCE services vsce

"vsce" namespace

repl shayne

"shayne" namespace

Kubernetes Cluster

Azure

@Azuretar

One Namespace for each developer

Let's Demo VSCE

Thanks!

https://www.slideshare.net/JorgeArteiro
http://aka.ms/signup-vsce (VS connected Environment preview)
https://docs.microsoft.com/en-us/visualstudio/connected-environment/get-started
https://github.com/Azure/vsce
https://github.com/kubernetes/charts (helm Charts)
https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough
https://github.com/virtual-kubelet/virtual-kubelet (Virtual Kubelet)
https://shell.azure.com/
https://hub.docker.com/u/microsoft/
https://hub.docker.com/u/azuresdk/

@Azuretar