MICROSERVICES IN AKS

Mike Douglas
Director of Delivery Engineering

ABOUT ME

- Mike Douglas
- Solution Consultant and Director of Delivery Engineering at Lunavi
- Microsoft MVP – Developer Technologies – Azure DevOps
- Organizer of Omaha DevOps Meetup
- Competitive Robotics Club Coordinator for 4th – 6th Graders
- AIM Events Steering Committee
- UNO ISQA Curriculum Advisor
- Mentor at Do Space
- @mikedouglasdev on twitter

LATE TO THE PARTY

- **My Microservices Focus**
  - Azure Platform as a Service (PAAS)
  - Azure Functions (Serverless)
- **Dragging feet on Kubernetes**
  - Required high level of sophistication
  - AKS had limited enterprise features

LUNAVI

CHALLENGES WITH PAAS AND SERVERLESS AND MICROSERVICES

- Public by default / Access Restrictions
- Service to Service Communication
- ARM Templates
- Debugging and Local Development

LUNAVI

KUBERNETES AND AKS – HOW DOES AKS ADDRESS THE PROBLEMS

- **Public by default / Access Restrictions**
  - More granular control of each service
- **Service to Service Communication**
  - Easy management
  - Mutual TLS
- **ARM Templates**
- **Debugging and Local Development**

LUNAVI

KUBERNETES AND AKS

- **What additional challenges / problems does Kubernetes and AKS create?**
  - Complexity
    - Docker, Kubernetes, HELM, Az PS, many more,
  - “image” management
  - Security
  - Operations and Management

WHAT IS KUBERNETES AND AKS

KUBERNETES AND AKS

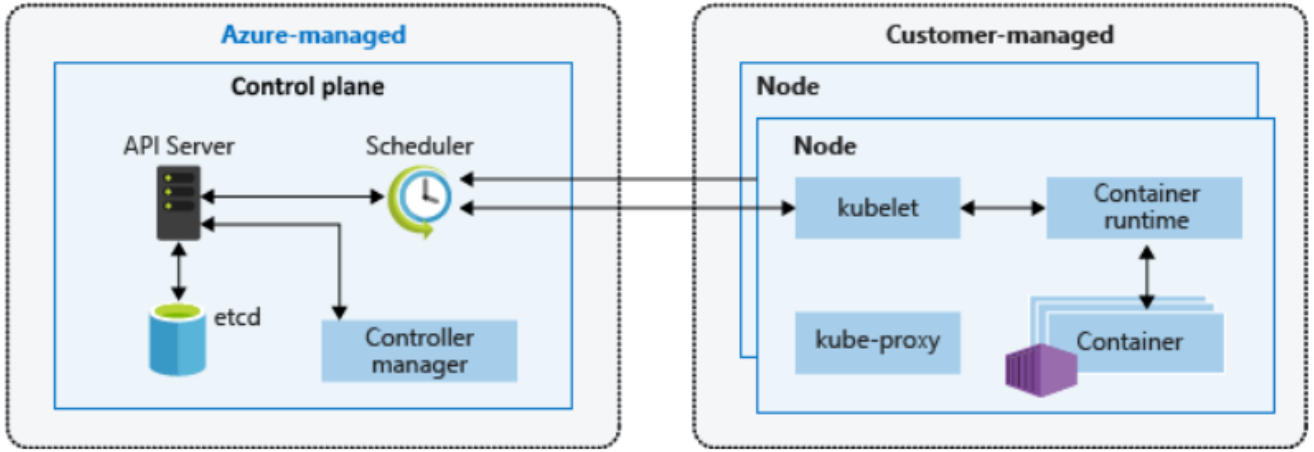
- **What is it?**
  - Kubernetes – open source orchestration tool for deploying, managing, and scaling container applications
  - AKS (Azure Kubernetes Service) – Managed Kubernetes Service
    - You don’t worry about underlying servers
    - You also don’t pay for the management servers

INFRASTRUCTURE RESOURCES

LUNAVI

KUBERNETES AND AKS – INFRASTRUCTURE RESOURCES

- **Control Plane - nodes that provide the core Kubernetes services and orchestration of application workloads**
- **Nodes - run your workloads**
- **Node Pools - nodes that are grouped together with the same configuration**

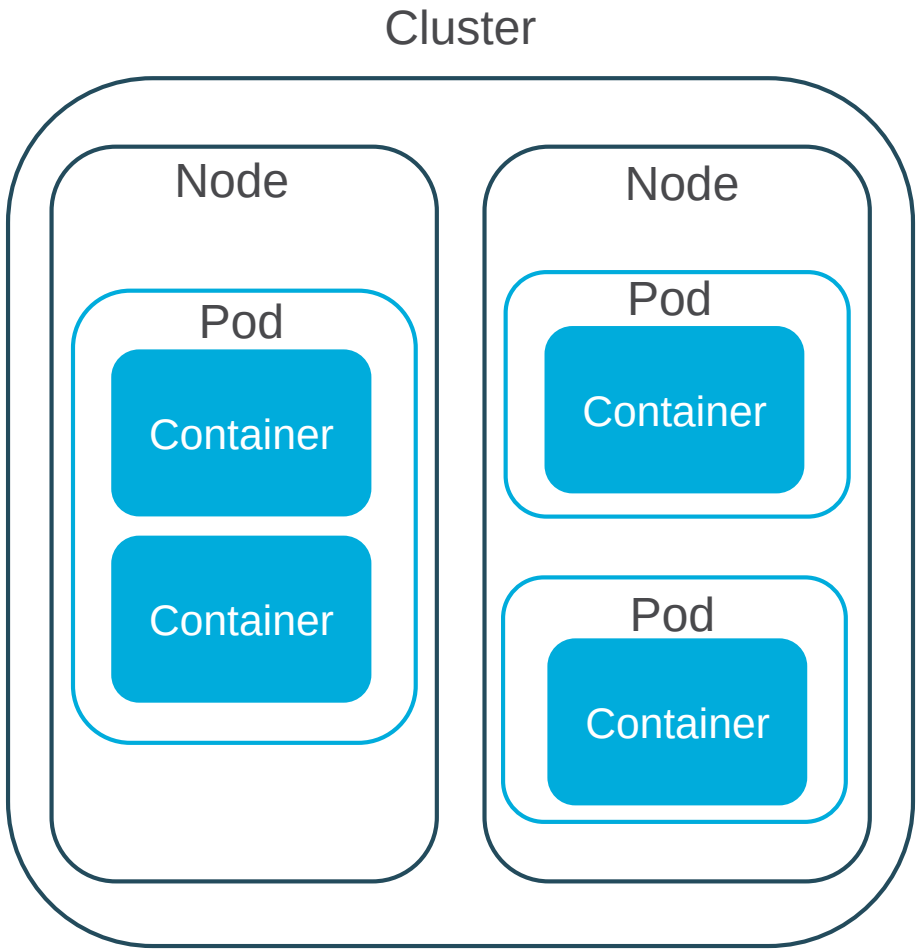


LUNAVI

WORKLOAD RESOURCES

KUBERNETES AND AKS - WORKLOAD RESOURCES

- **Pods**
  - unit of work in Kubernetes. A single pod hosts one or more containers
  - Share same network, process, and memory space
  - Pods can't be created directly. They are managed by other resources called controllers



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ratings-api
spec:
  selector:
    matchLabels:
      app: ratings-api
  template:
    metadata:
      labels:
        app: ratings-api # the label for the pods and the deployments
    spec:
      containers:
      - name: ratings-api
        image: acrMDDemo.azurecr.io/ratings-api:v1 # IMPORTANT: update with your own repository
        imagePullPolicy: Always
        ports:
        - containerPort: 3000 # the application listens to this port
        env:
        - name: MONGODB_URI # the application expects to find the MongoDB connection details in this environment variable
          valueFrom:
            secretKeyRef:
              name: mongosecret # the name of the Kubernetes secret containing the data
              key: MONGOCONNECTION # the key inside the Kubernetes secret containing the data
        resources:
          requests: # minimum resources required
            cpu: 250m
            memory: 64Mi
          limits: # maximum resources allocated
            cpu: 500m
            memory: 256Mi
        readinessProbe: # is the container ready to receive traffic?
```

environment variable

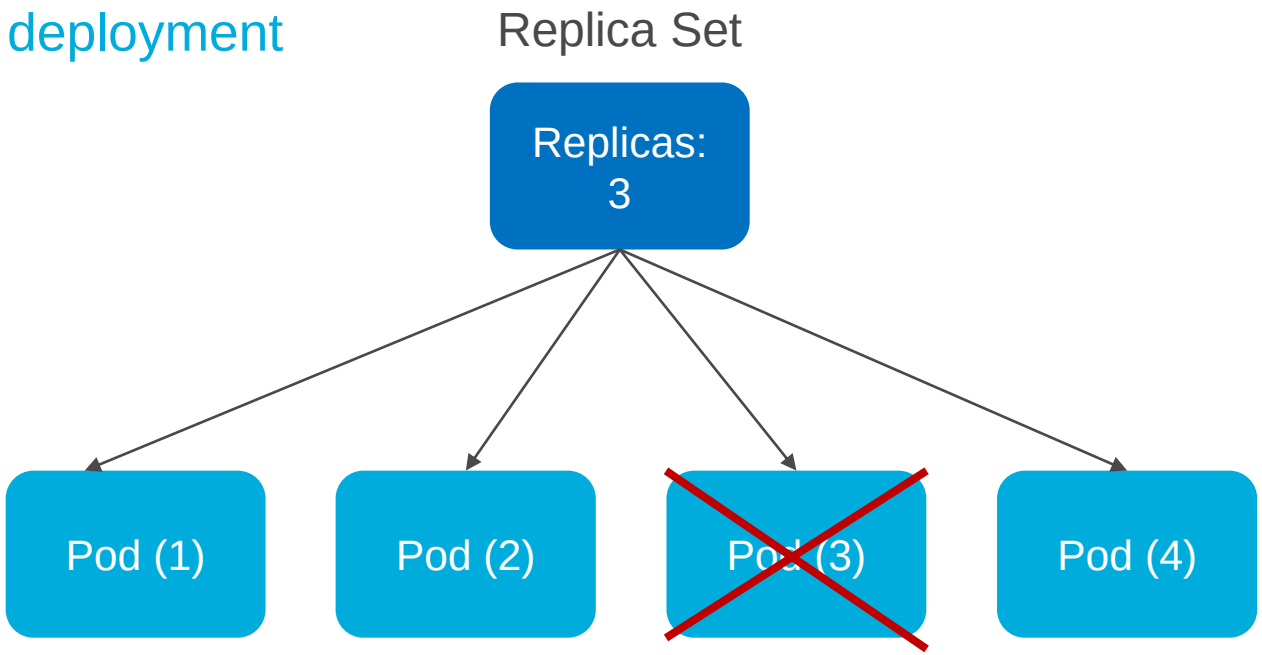
KUBERNETES AND AKS - WORKLOAD RESOURCES

- **Services**
  - Provides stable networking for pods
  - Enables communication between nodes, pods, and users of your application

```yaml
apiVersion: v1
kind: Service
metadata:
  name: ratings-api
spec:
  selector:
    app: ratings-api
  ports:
  - protocol: TCP
    port: 80
    targetPort: 3000
  type: ClusterIP
```

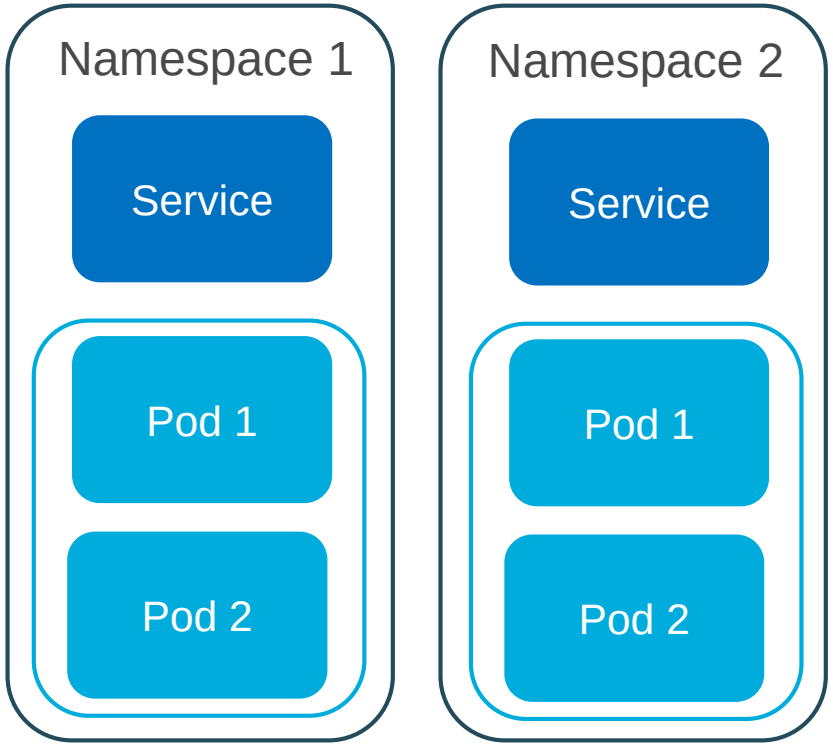

KUBERNETES AND AKS - WORKLOAD RESOURCES

- **Replica Set**
  - Manages a replicated set of pods.
  - Replica Set is managed by the deployment



KUBERNETES AND AKS - WORKLOAD RESOURCES

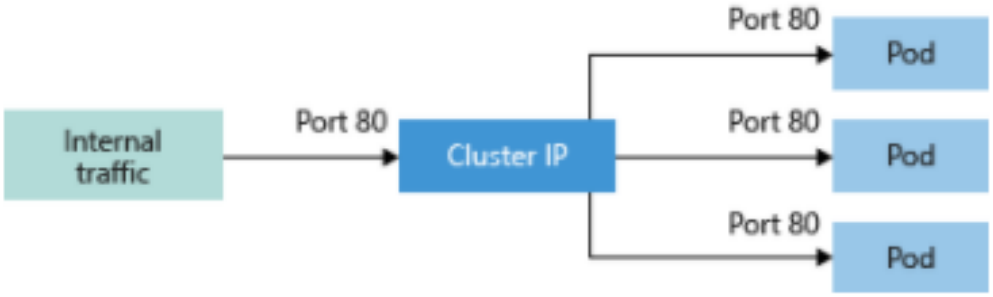
- **Namespaces**
  - Logical isolation boundary
  - Virtual group of pods/services
  - Examples: environments, products, type, etc.



SERVICE TYPES

KUBERNETES AND AKS - SERVICE TYPES

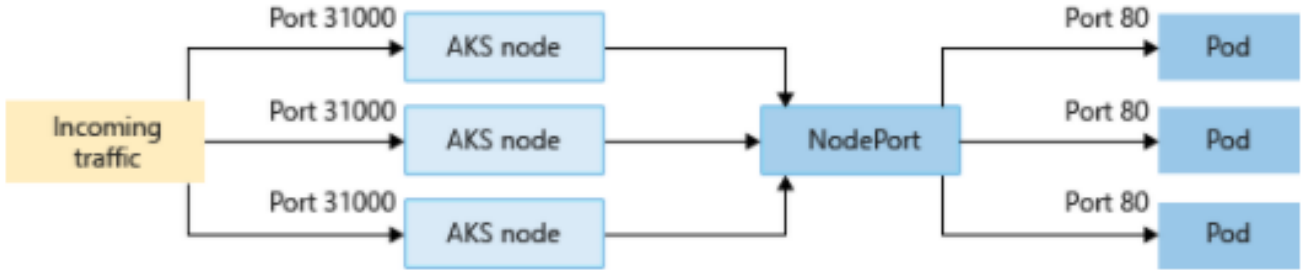
- **Cluster IP - Creates internal IP to group pods – internal only**



LUNAVI

KUBERNETES AND AKS - SERVICE TYPES

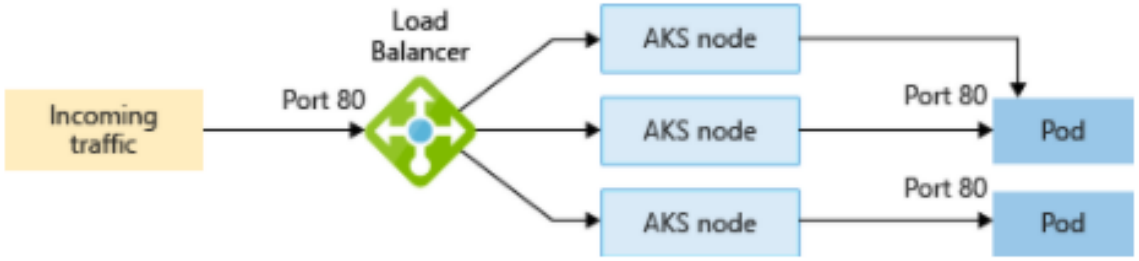
- **NodePort - Creates port mapping so NodePort can be directly accessible with node IP and port**



LUNAVI

KUBERNETES AND AKS - SERVICE TYPES

- **LoadBalancer - Creates load balancer resource, configures external IP address**



KUBERNETES AND AKS - SERVICE TYPES

- **ExternalName - Creates a specific DNS entry for easier application access**

LUNAVI

KUBERNETS AND AKS - NETWORKING

- **Kubenet – basic / default option – Azure manages the VNET resources**
  - Use for smaller and test workloads
  - Nodes get IP from the VNET
  - Pods get an IP from different address space than nodes
  - Source IP is NAT'd to node's primary IP
  - Can let AKS create VNET or specify
- **Azure CNI – advanced**
  - Use for larger, production workloads
  - Every pod gets an IP from subnet and can be accessed directly
  - Traffic to endpoints in the same VNET isn't NAT'd to node's primary IP

MICROSERVICES

MICROSERVICES

- **What problem are we trying to solve?**
  - Independently deploy and release
  - Smaller changes
- **What problems do they cause?**
  - Not just architecture
  - Forces changes to engineering practices, people, and culture
  - Can't just create a distributed "monolith"

LUNAVI

MICROSERVICES IN AKS

- **Benefits**
  - Containerized
  - Easy independent deployments
  - Easy to control access restrictions
  - Public, backend, cross API communication

LUNAVI

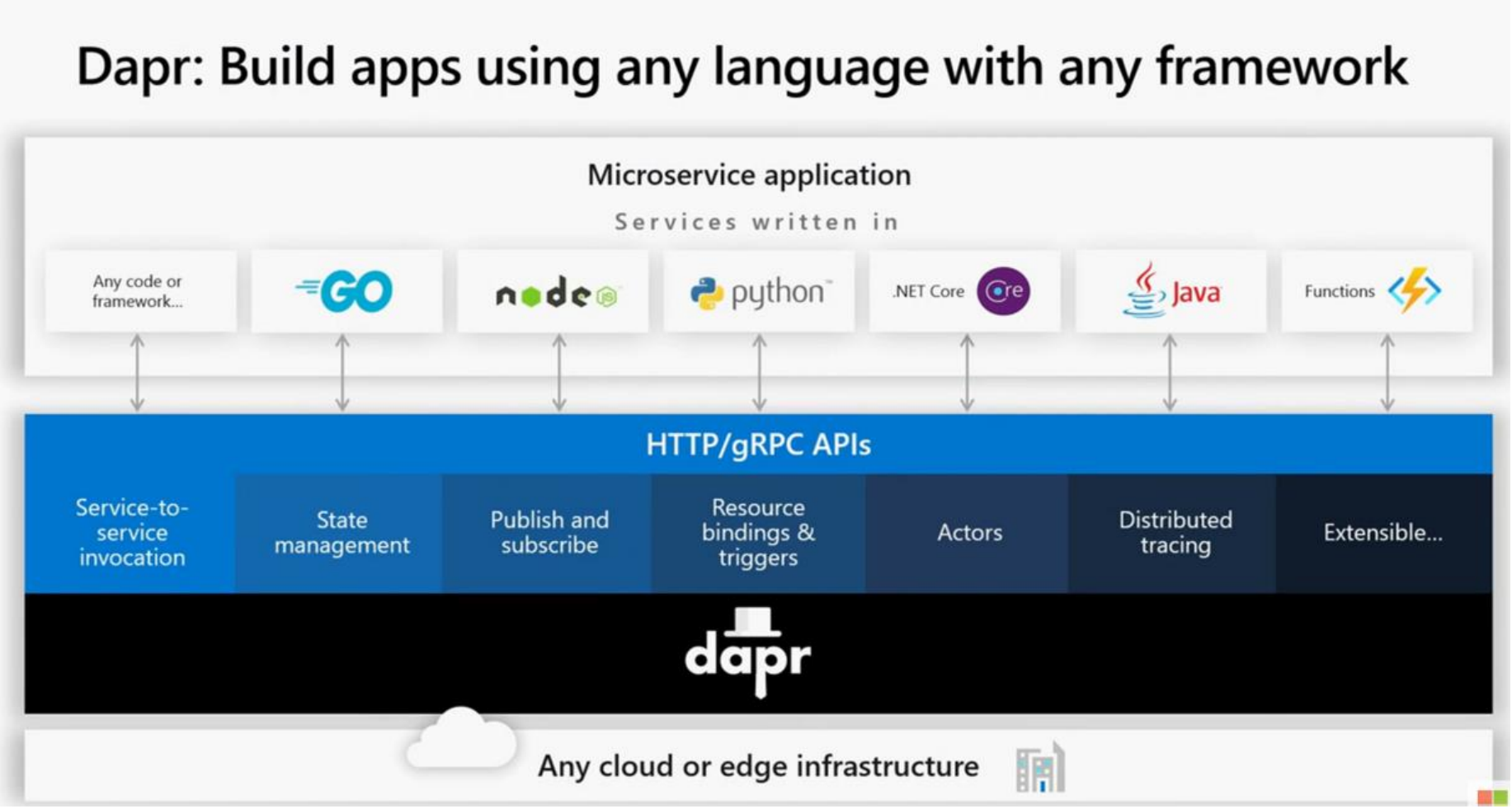
MICROSERVICES IN AKS

- **Tools for Cross Cutting Concerns**
  - Service Mesh
  - API Gateway
  - Dapr

LUNAVI

MICROSERVICES IN AKS

- **Dapr**



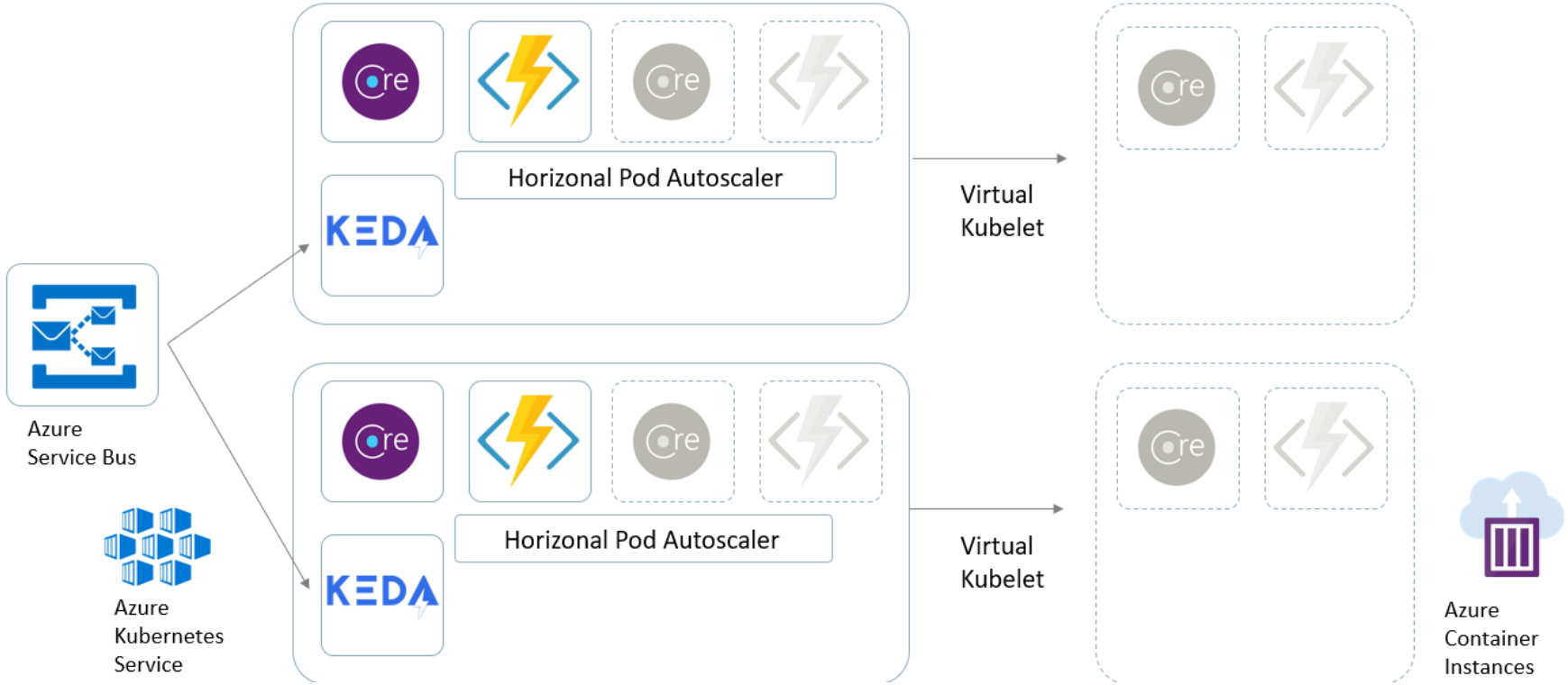
LUNAVI

MICROSERVICES IN AKS DEMO

- **Dapr Demo**

MICROSERVICES IN AKS

- **Scaling in AKS**



LUNAVI

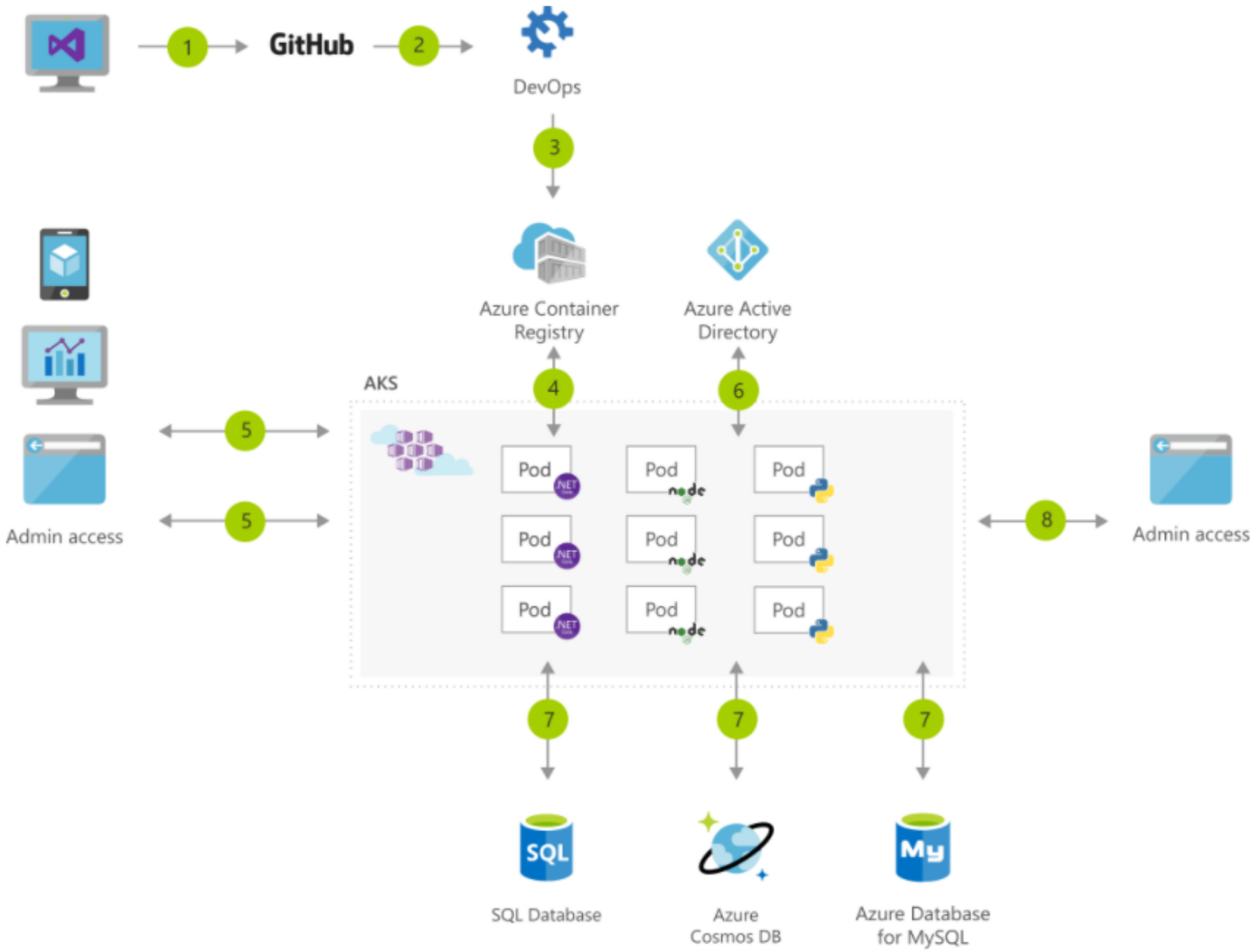
PIPELINES

AKS PIPELINES

- Building and Packaging
- Deployments
- Secure DevOps

LUNAVI

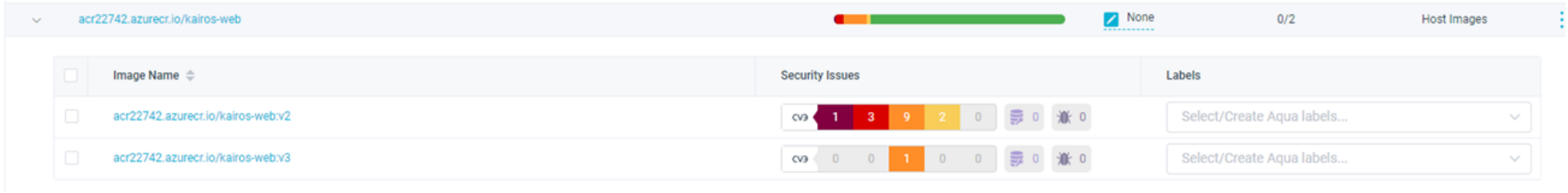
AKS PIPELINES

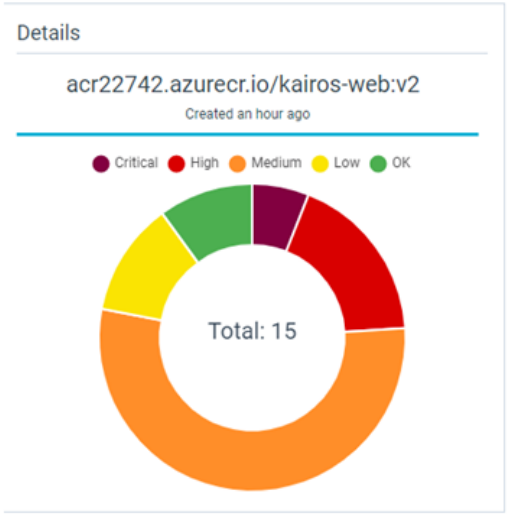


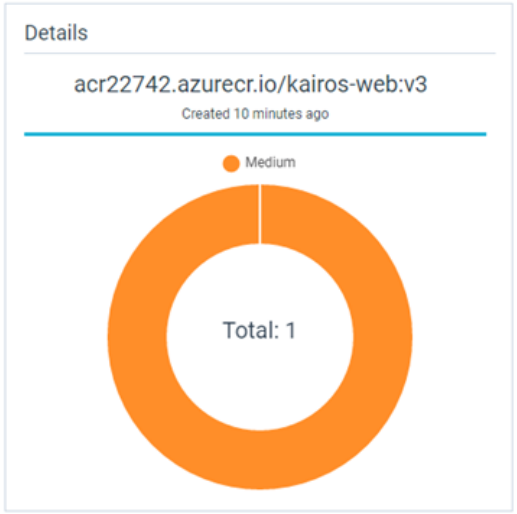
LUNAVI

SECURE DEVOPS

- **Aqua**







LUNAVI

LINKS

- **Kubernetes Cheat Sheet**
  - https://linuxacademy.com/site-content/uploads/2019/04/Kubernetes-Cheat-Sheet_07182019.pdf
- **Hands On Labs**
  - https://aksworkshop.io/
  - https://kubesec.aksworkshop.io/ (some items in lab are outdated)
- **Azure Pipelines HOL**
  - https://docs.microsoft.com/en-us/learn/modules/deploy-kubernetes/
- **Dapr**
  - https://dapr.io/

QUESTIONS?