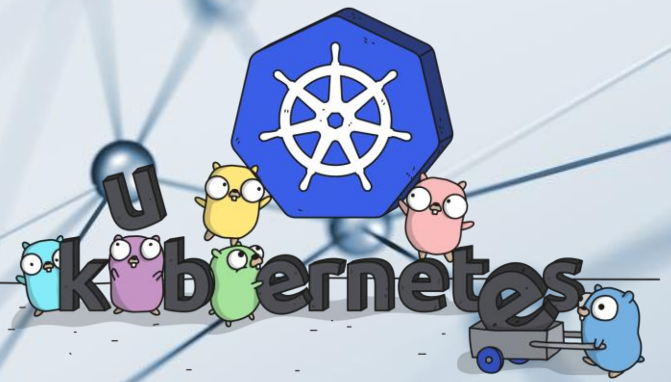
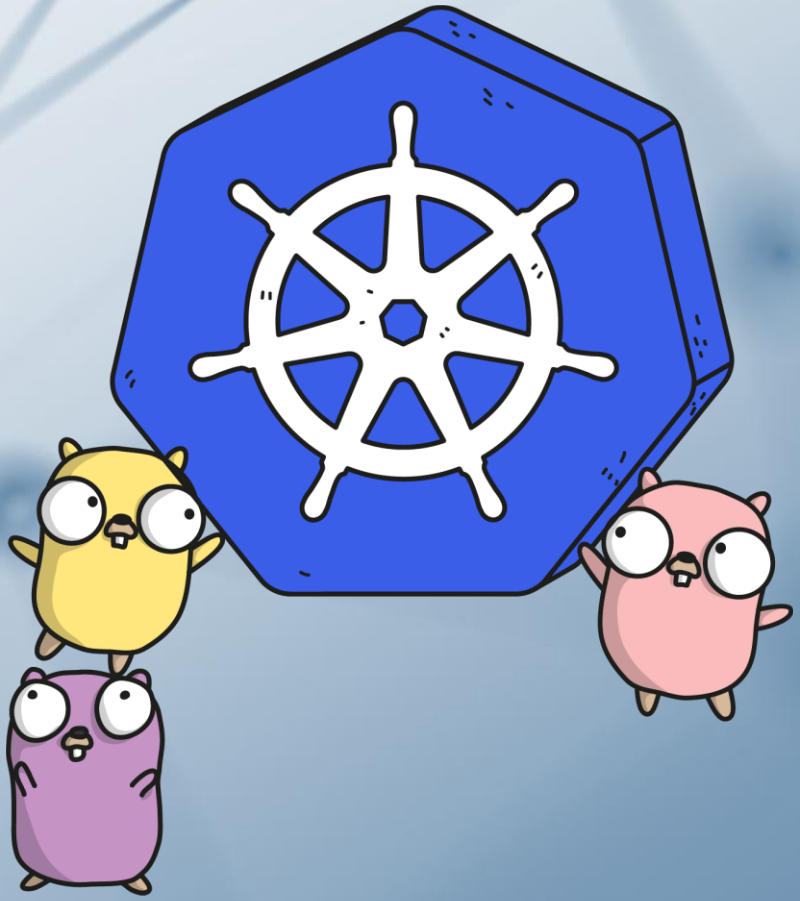


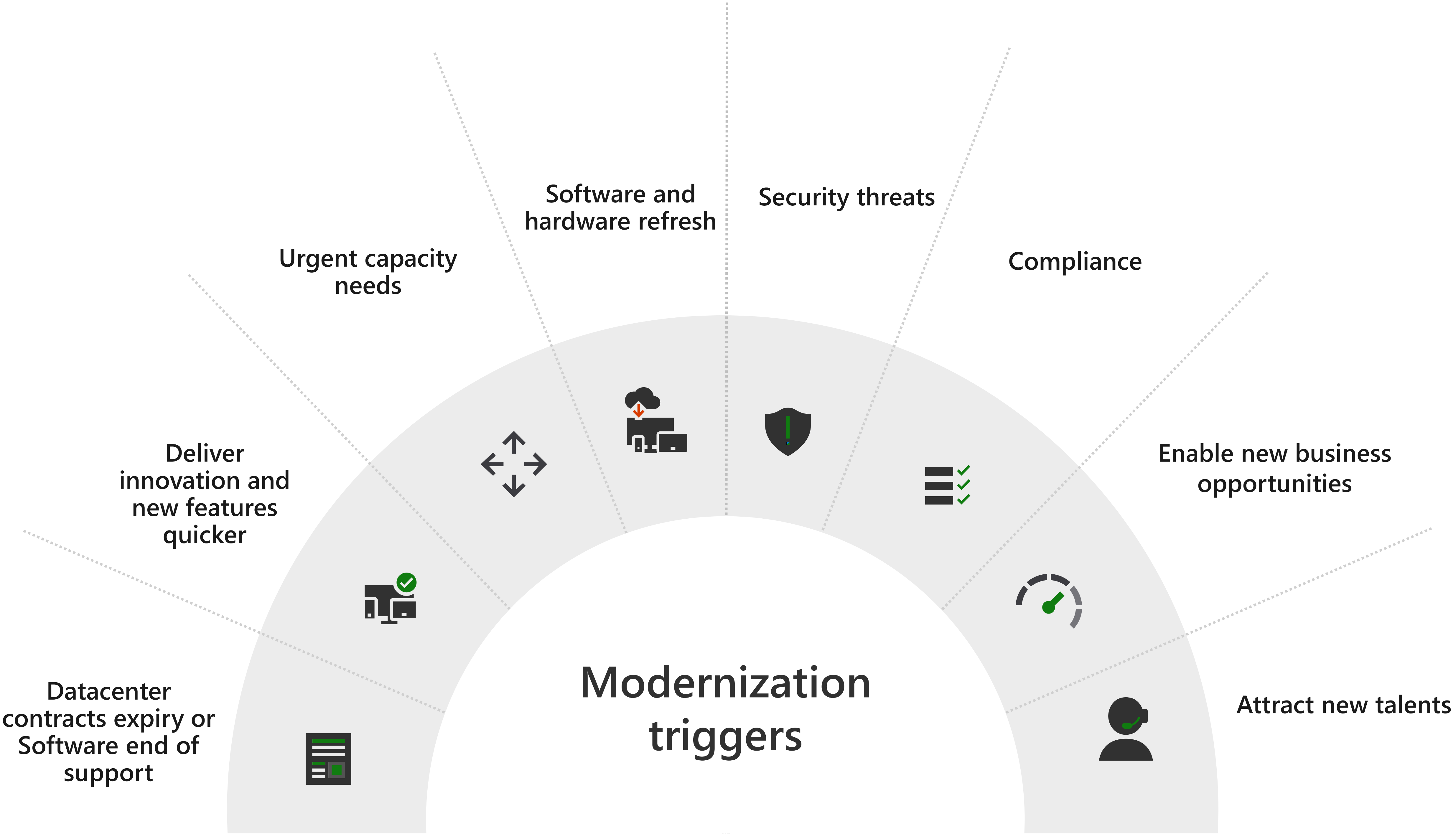


KUBERNETES 101

Gustav Kaleta

Global Black Belt
Tech Lead EMEA
Microsoft

Twitter: @kaletaii
email: gkaleta@



Modernization triggers

Datacenter contracts expiry or Software end of support

Deliver innovation and new features quicker

Urgent capacity needs

Software and hardware refresh

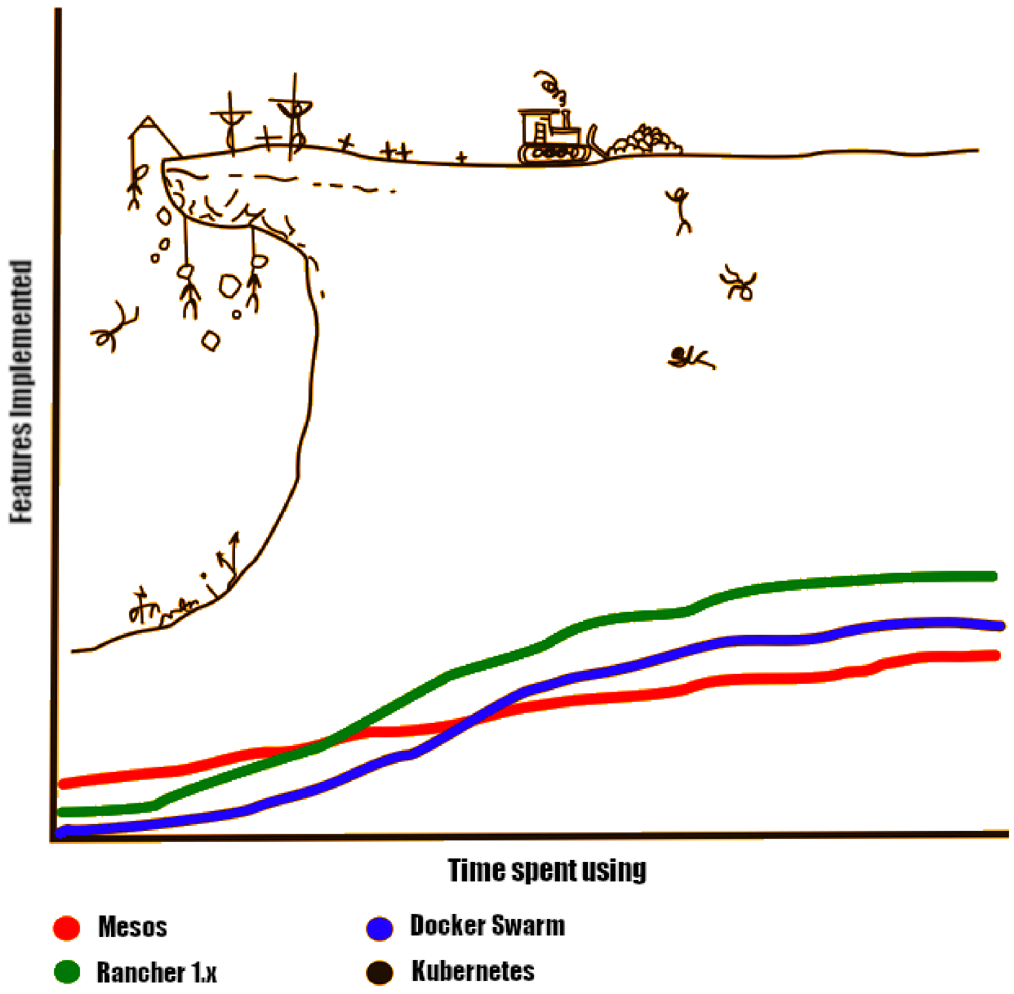
Security threats

Compliance

Enable new business opportunities

Attract new talents

Learning curves of some Container Orchestration Engines



Kubernetes momentum

By 2022

>75%

of global organizations will be running containerized applications in production[1]

[1] Gartner.

What’s behind the growth?

Kubernetes: the leading orchestrator shaping the future app development and management

It’s widely used

Kubernetes is in production for global companies across industries[1]

- Capital One
- eBay
- SAP
- New York Times
- Pokémon Go
- Spotify

It’s vendor-neutral

A variety of cloud providers offer robust Kubernetes support

- Azure
- AWS
- VMWare
- Red Hat

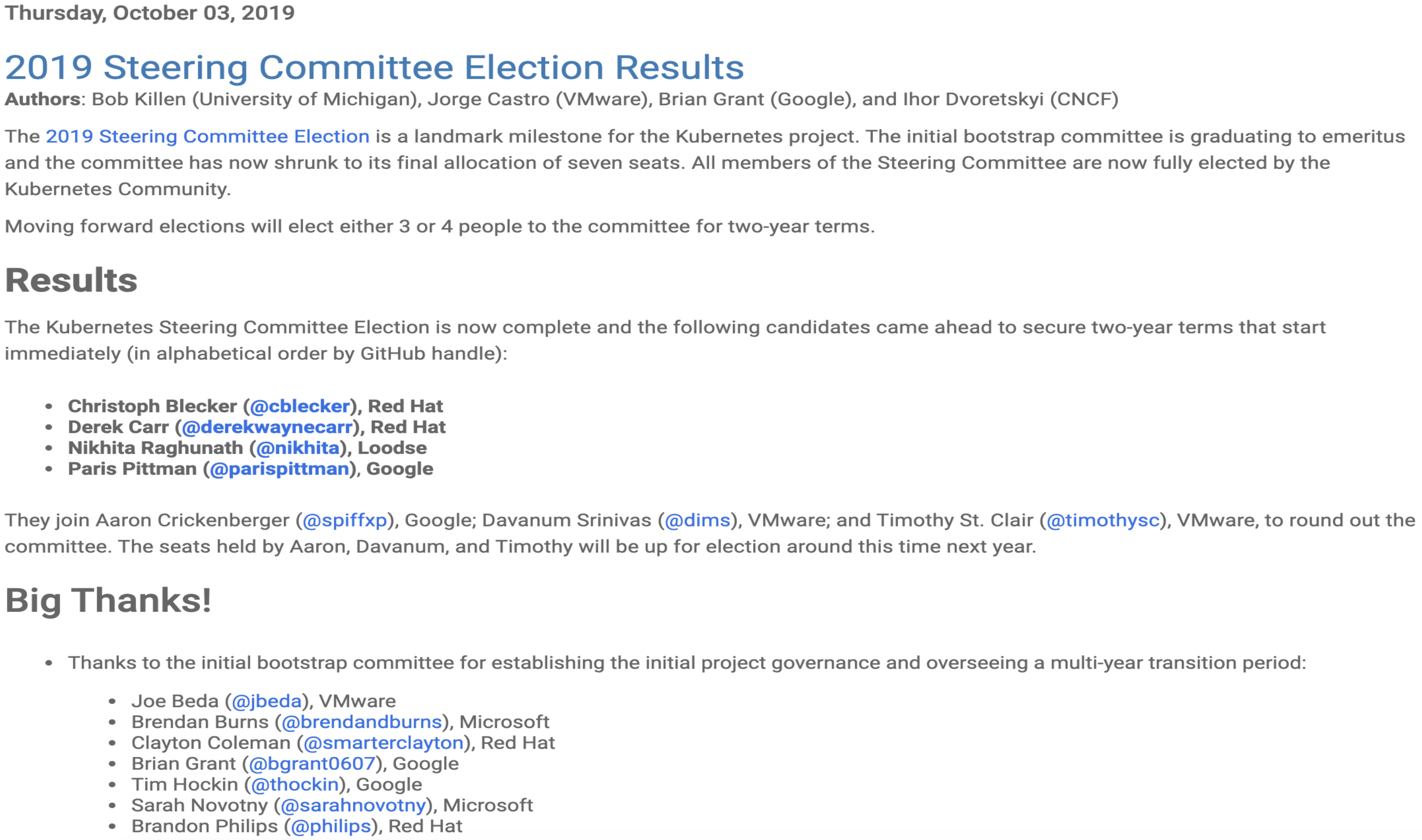
It’s community-supported

There’s a huge community of active contributors supporting Kubernetes[3]

24,000 contributors since 2016

1.1 million contributions since 2016

[1] Kubernetes.io. “Kubernetes User Case Studies.” [2] CNCF. “Kubernetes Is First…” [3] CNCF. Keynote address.



Thursday, October 03, 2019

2019 Steering Committee Election Results

Authors: Bob Killen (University of Michigan), Jorge Castro (VMware), Brian Grant (Google), and Ihor Dvoretskyi (CNCF)

The 2019 Steering Committee Election is a landmark milestone for the Kubernetes project. The initial bootstrap committee is graduating to emeritus and the committee has now shrunk to its final allocation of seven seats. All members of the Steering Committee are now fully elected by the Kubernetes Community.

Moving forward elections will elect either 3 or 4 people to the committee for two-year terms.

Results

The Kubernetes Steering Committee Election is now complete and the following candidates came ahead to secure two-year terms that start immediately (in alphabetical order by GitHub handle):

- **Christoph Blecker (@cblecker), Red Hat**
- **Derek Carr (@derekwaynecarr), Red Hat**
- **Nikhita Raghunath (@nikhita), Loodse**
- **Paris Pittman (@parispittman), Google**

They join Aaron Crickenberger (@spiffxp), Google; Davanum Srinivas (@dims), VMware; and Timothy St. Clair (@timothysc), VMware, to round out the committee. The seats held by Aaron, Davanum, and Timothy will be up for election around this time next year.

Big Thanks!

- Thanks to the initial bootstrap committee for establishing the initial project governance and overseeing a multi-year transition period:
    - Joe Beda (@jbeda), VMware
    - Brendan Burns (@brendandburns), Microsoft
    - Clayton Coleman (@smarterclayton), Red Hat
    - Brian Grant (@bgrant0607), Google
    - Tim Hockin (@thockin), Google
    - Sarah Novotny (@sarahnovotny), Microsoft
    - Brandon Philips (@philips), Red Hat

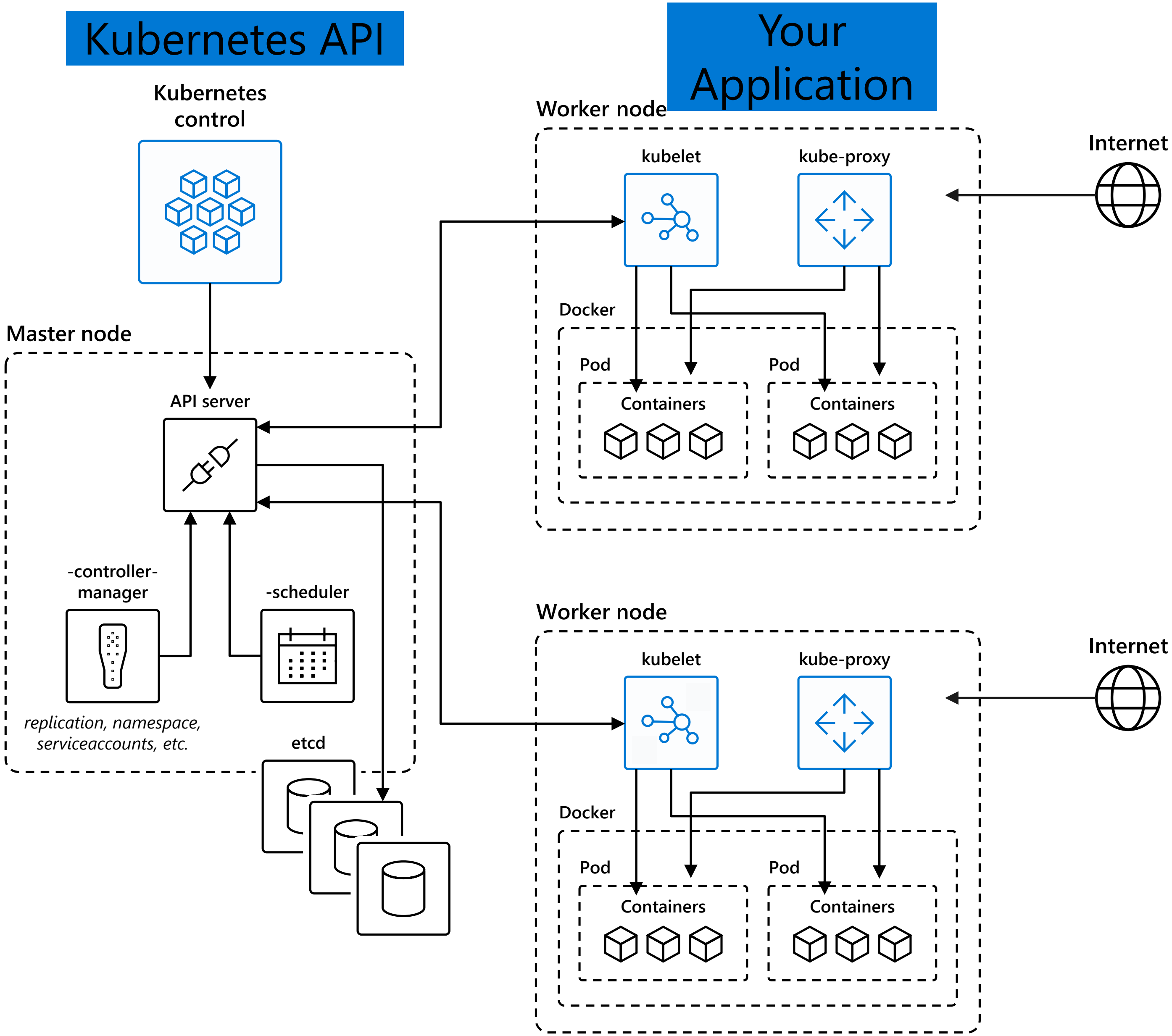
How Kubernetes works

1. Kubernetes users communicate with API server and apply desired state

2. Master nodes actively enforce desired state on worker nodes

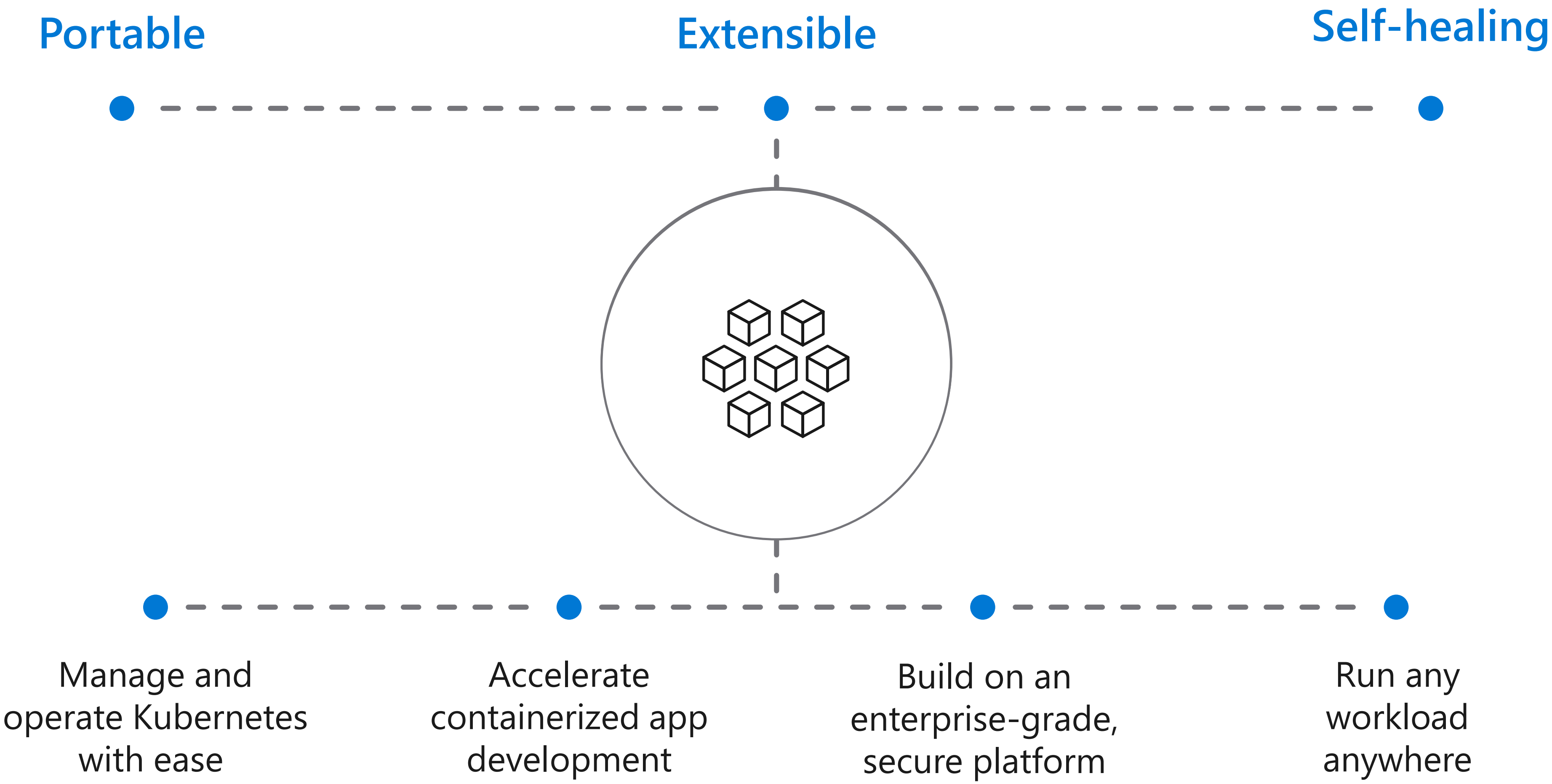
3. Worker nodes support communication between containers

4. Worker nodes support communication from the Internet



Kubernetes on Azure

Simplify the deployment, management, and operations of Kubernetes



Portable

Extensible

Self-healing

Manage and operate Kubernetes with ease

Accelerate containerized app development

Build on an enterprise-grade, secure platform

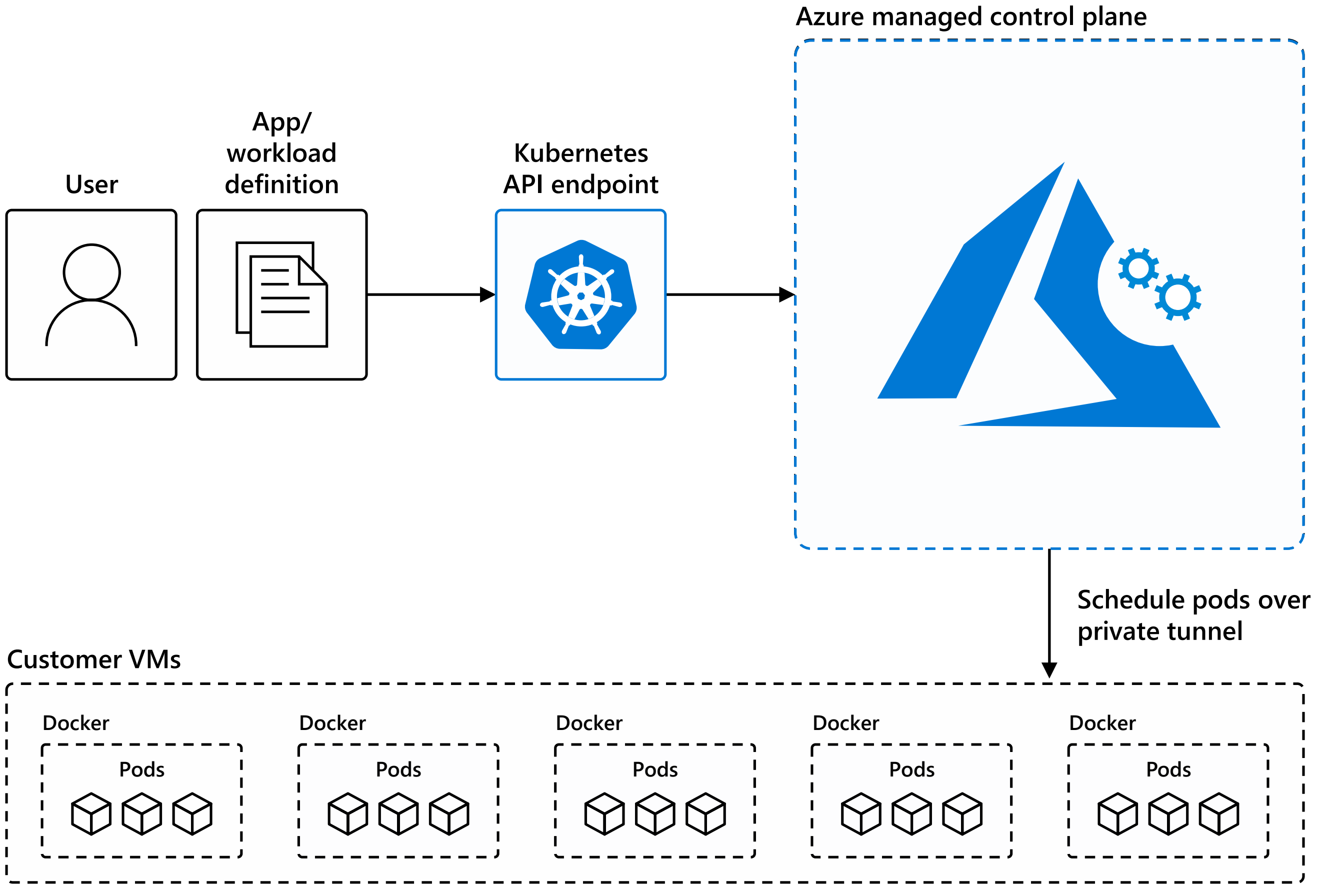
Run any workload anywhere

Manage Kubernetes with ease

Focus on your containers and code, not the plumbing of them

Responsibilities	DIY with Kubernetes	Managed Kubernetes on Azure
Containerization	🟦	🟦
Application iteration, debugging	🟦	🟦
CI/CD	🟦	🟦
Provisioning, upgrades, patches	🟦	🟦
Reliability availability	🟦	🟦
Scaling	🟦	🟦
Monitoring and logging	🟦	🟦

🟦 Customer 🟦 Microsoft



User

App/ workload definition

Kubernetes API endpoint

Azure managed control plane

Schedule pods over private tunnel

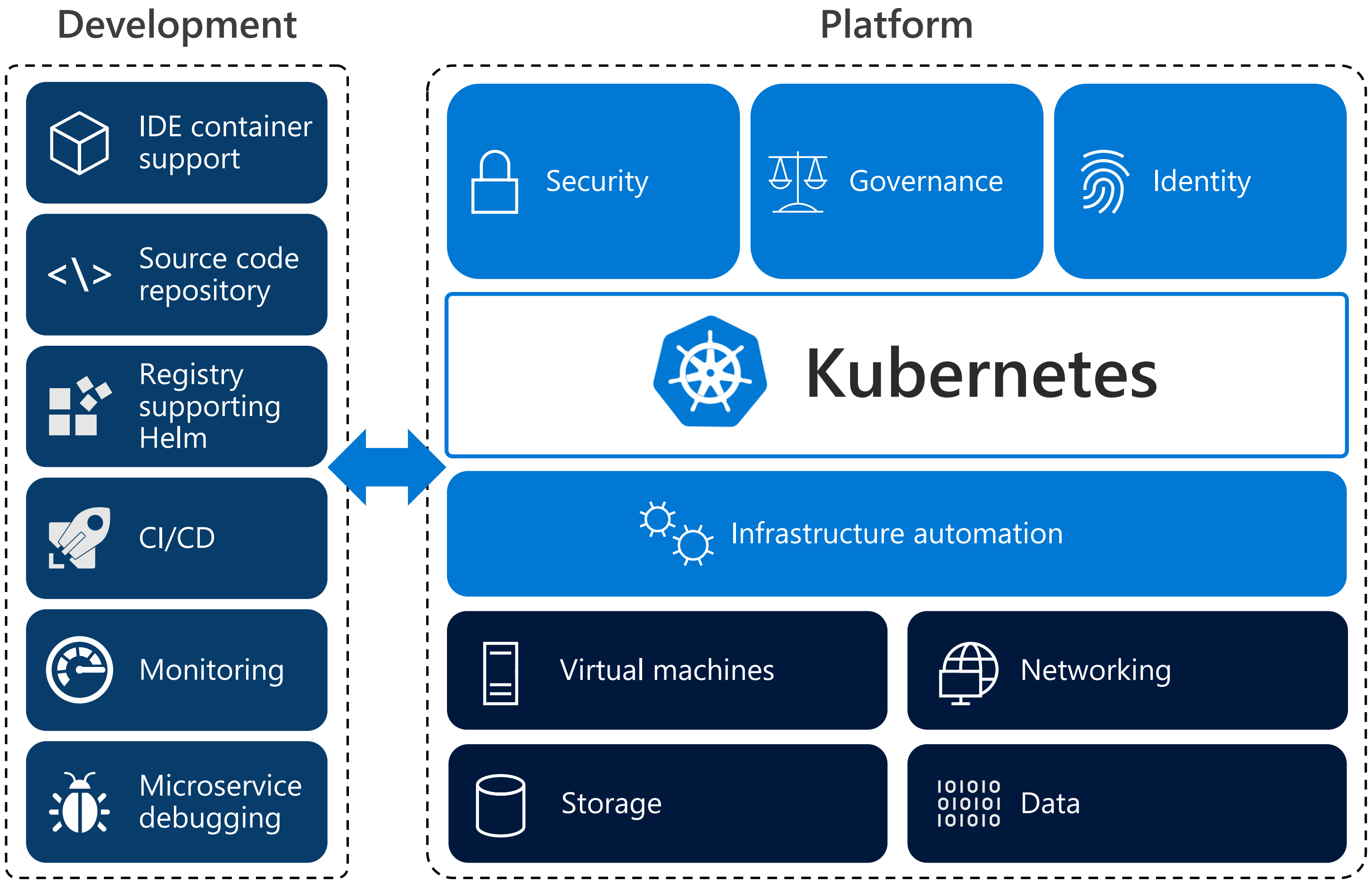
Customer VMs

Kubernetes on its own is not enough

Save time from infrastructure management and roll out updates faster without compromising security

Unlock the agility for containerized applications using:

- **Infrastructure automation** that simplifies provisioning, patching, and upgrading
- Tools for **containerized app development and CI/CD workflows**
- Services that support **security, governance, and identity and access management**



Accelerate containerized development

Kubernetes and DevOps better together

Develop

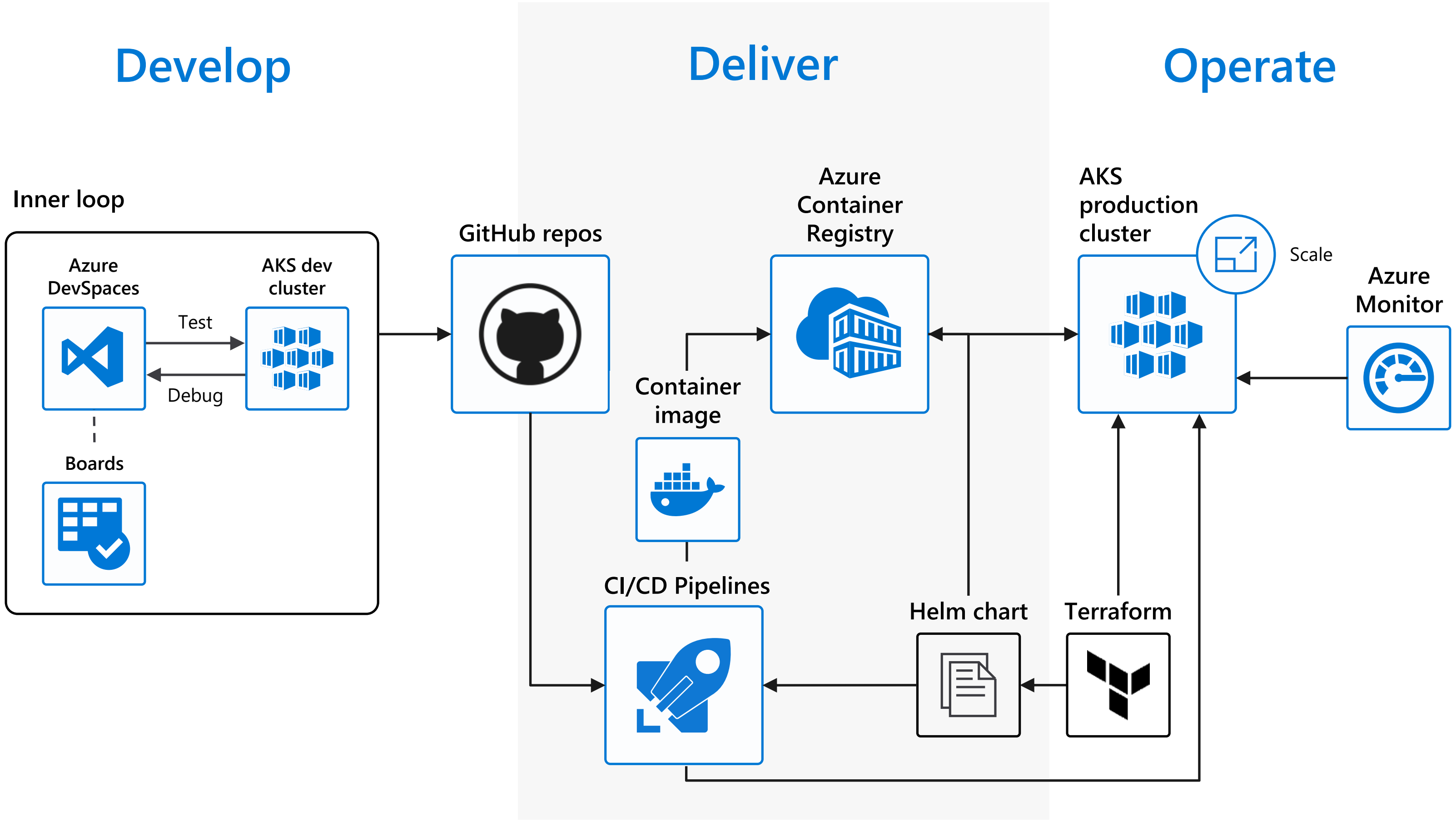
- Native containers and Kubernetes support in IDE
- Remote debugging and iteration for multi-containers
- Effective code merge
- Automatic containerization

Deliver

- CI/CD pipeline with automated tasks in a few clicks
- Pre-configured canary deployment strategy
- In depth build and delivery process review and integration testing
- Private registry with Helm support

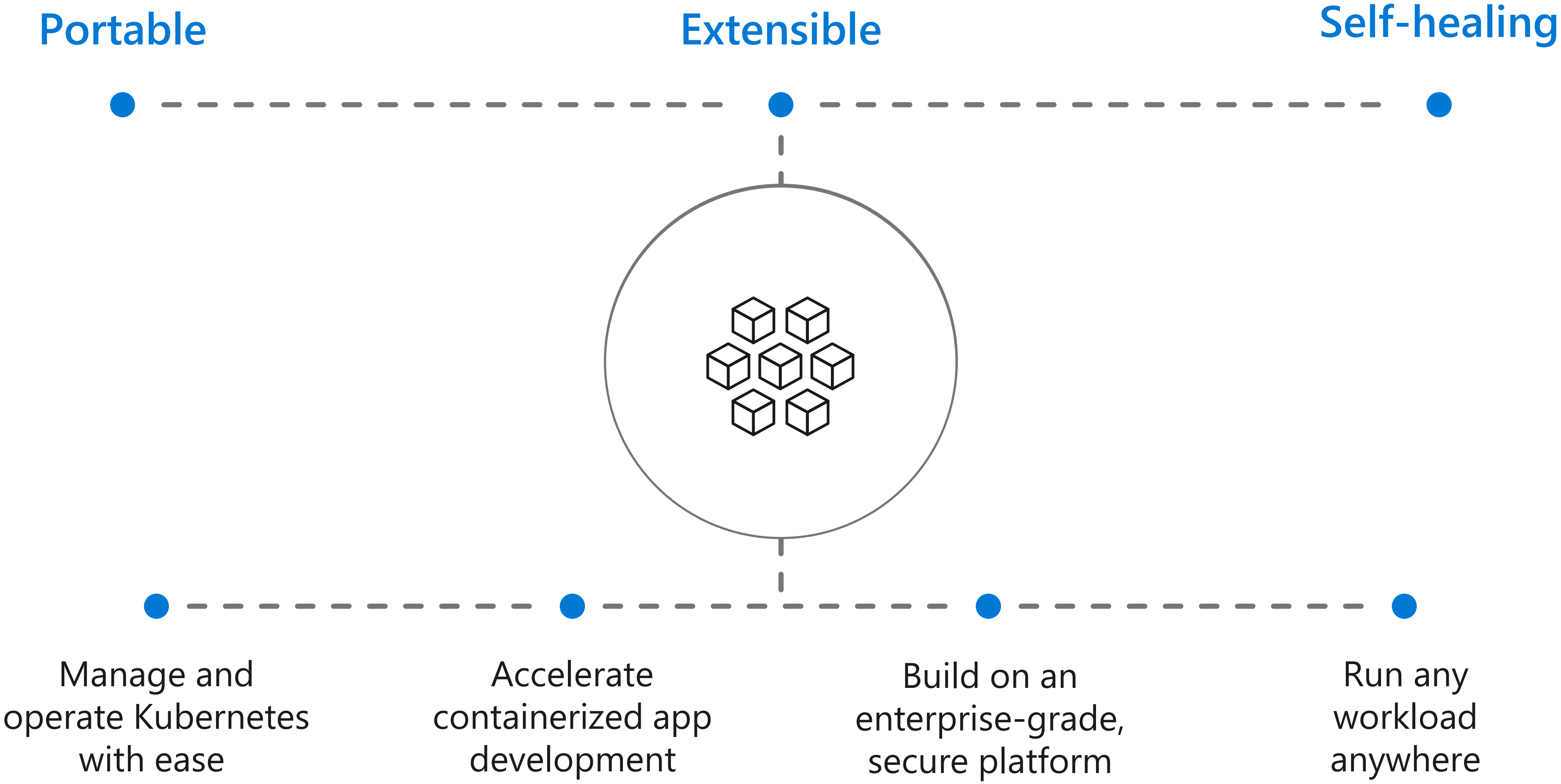
Operate

- Out-of-box control plane telemetry, log aggregation, and container health
- Declarative resource management
- Auto scaling



Kubernetes on Azure

Simplify the deployment, management, and operations of Kubernetes



Portable

Extensible

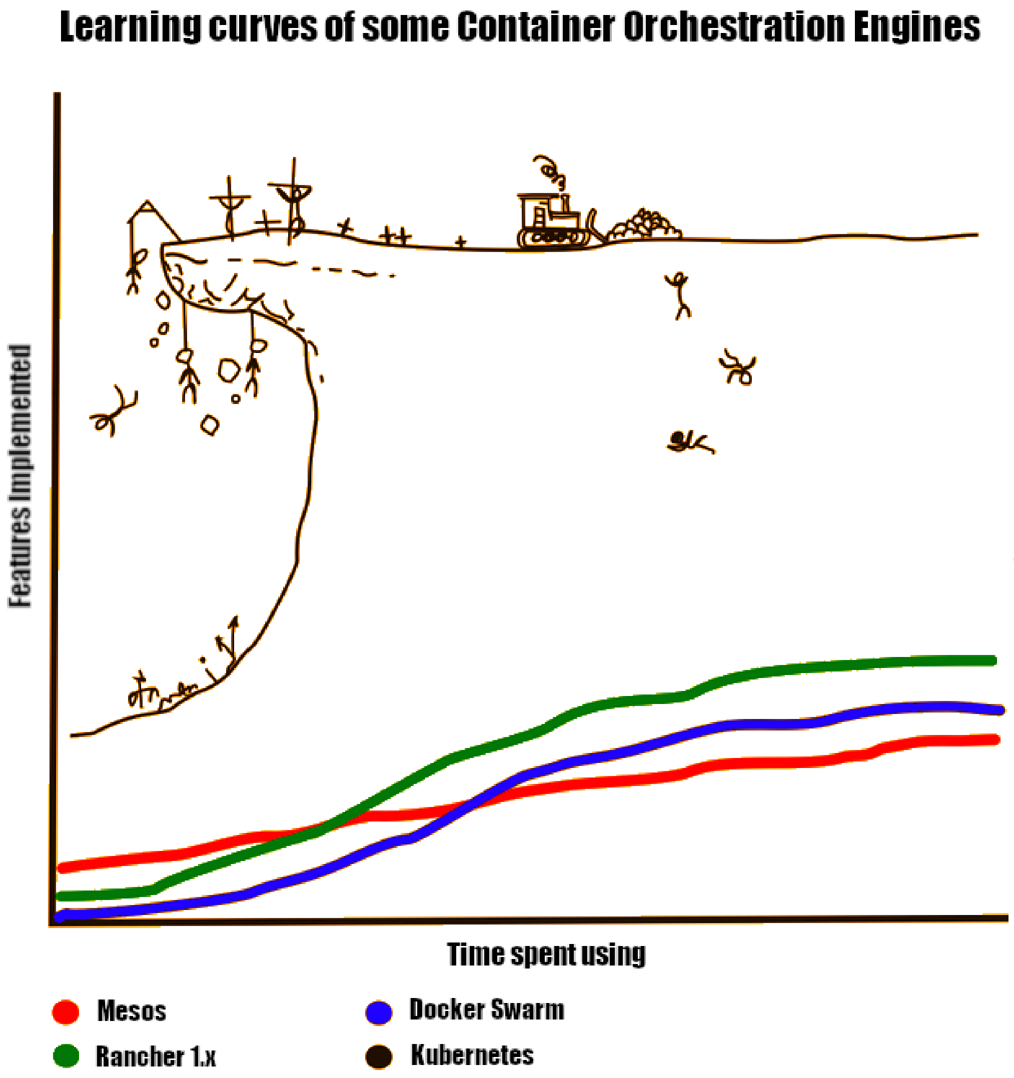
Self-healing

Manage and operate Kubernetes with ease

Accelerate containerized app development

Build on an enterprise-grade, secure platform

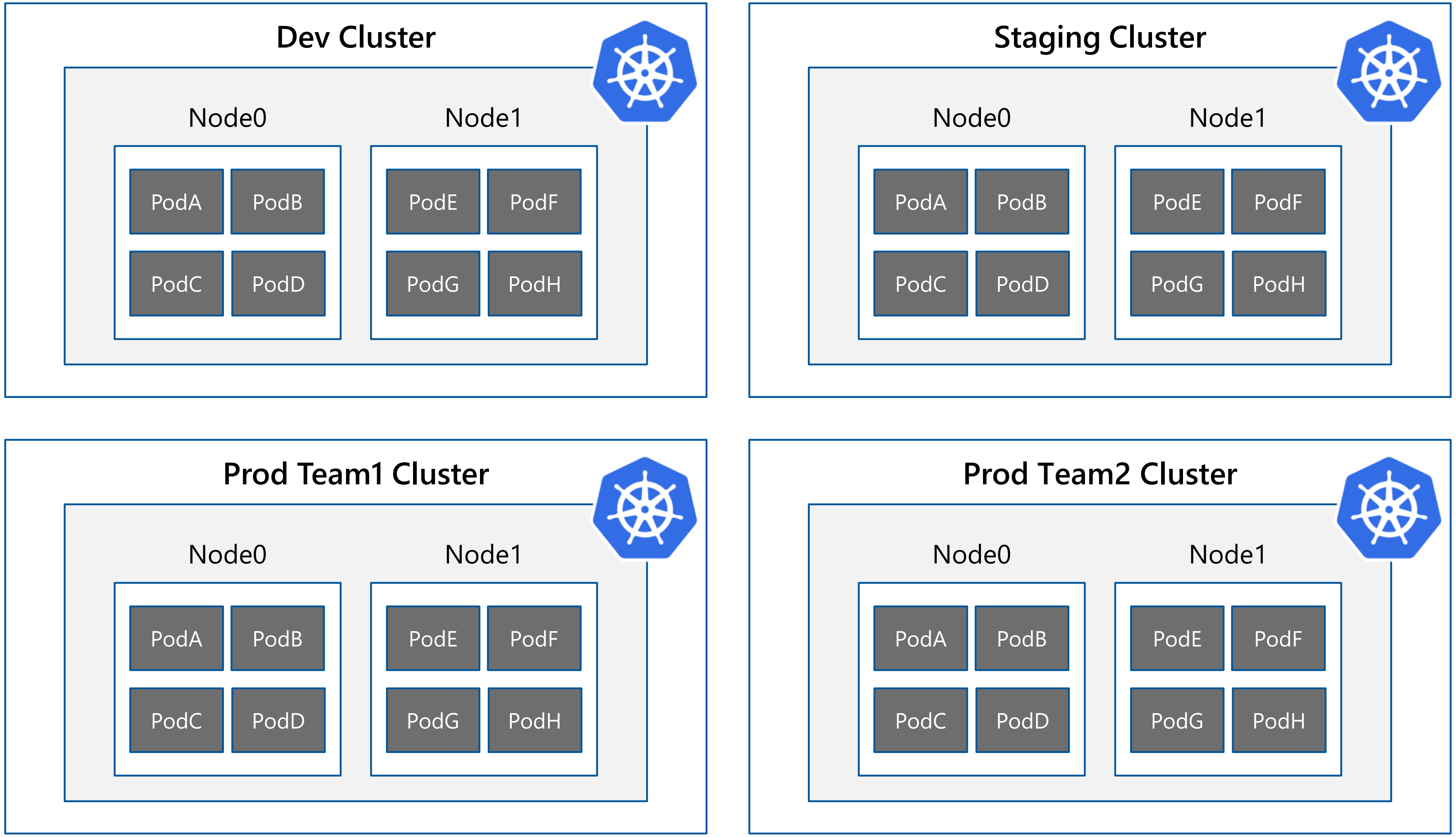
Run any workload anywhere



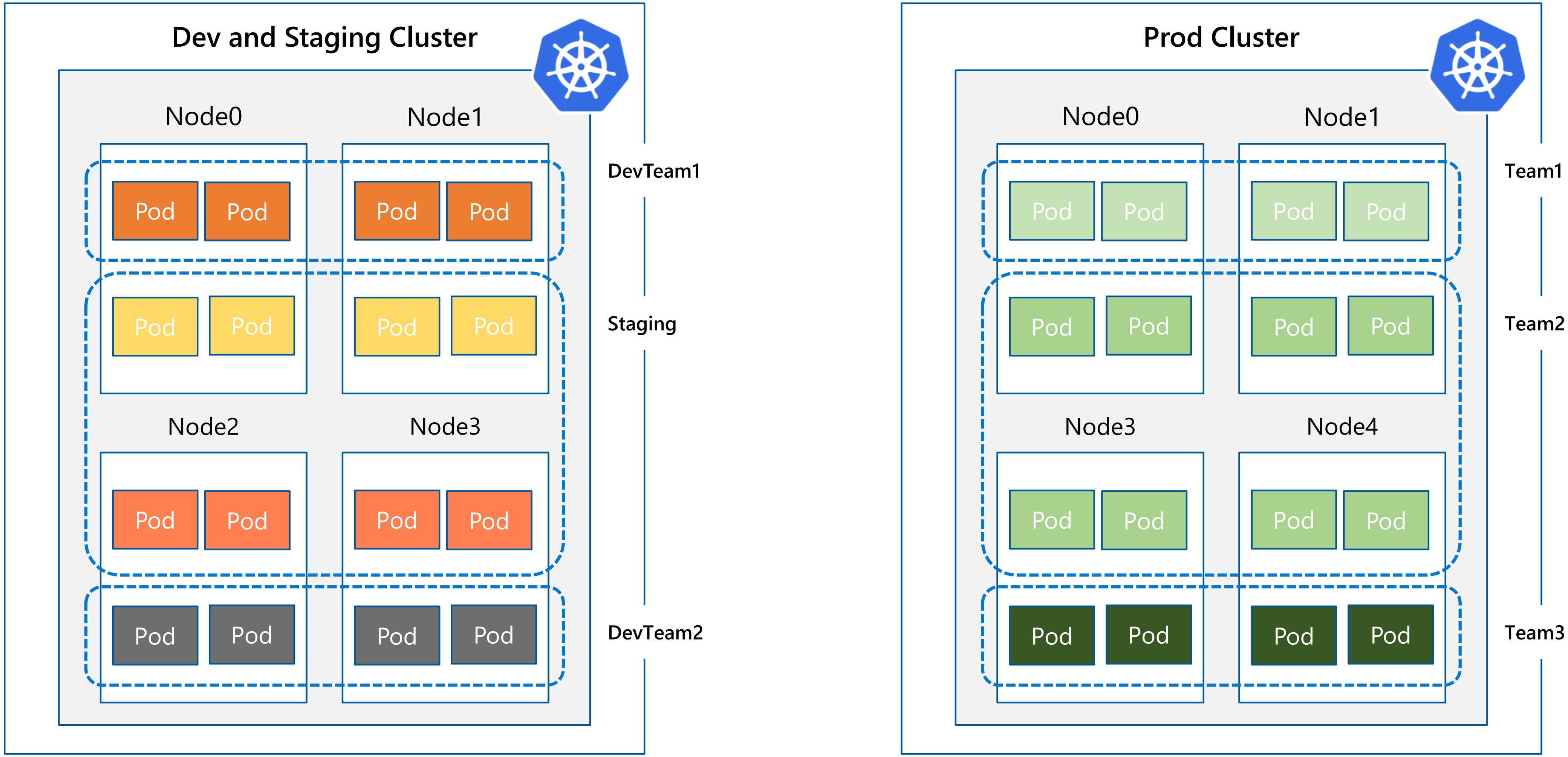
←Turn this into K8s ***Best Practices***

Join Skype Meeting

Cluster Isolation Patterns: Physical Isolation



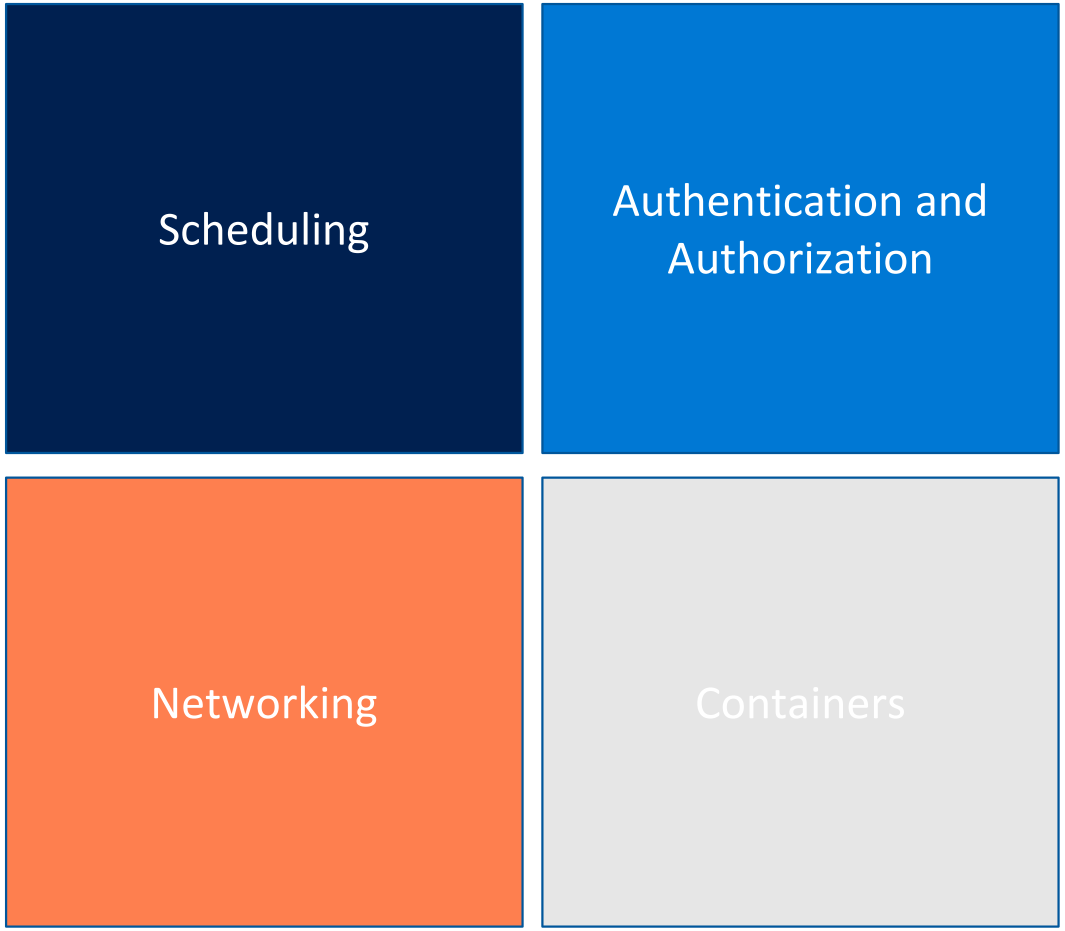
Cluster Isolation Patterns: Logical Isolation



Isolation Dimensions

Resource Quotas.
Node Selectors , Taints and Tolerations.
Node Affinity, Pod Affinity and Anti-Affinity
Pod Budget Policies
…

Network Policies



Scheduling

Authentication and Authorization

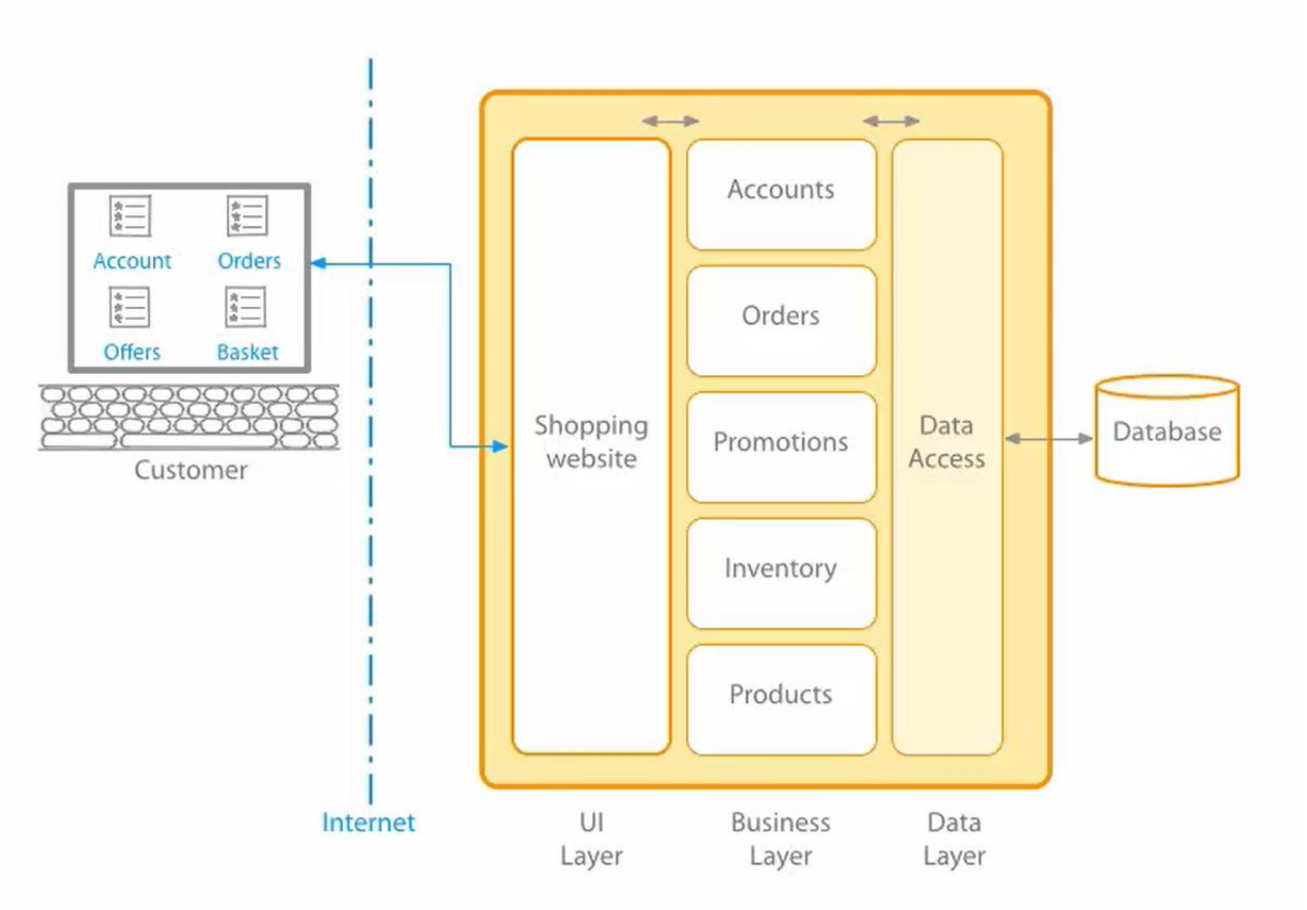
Networking

Containers

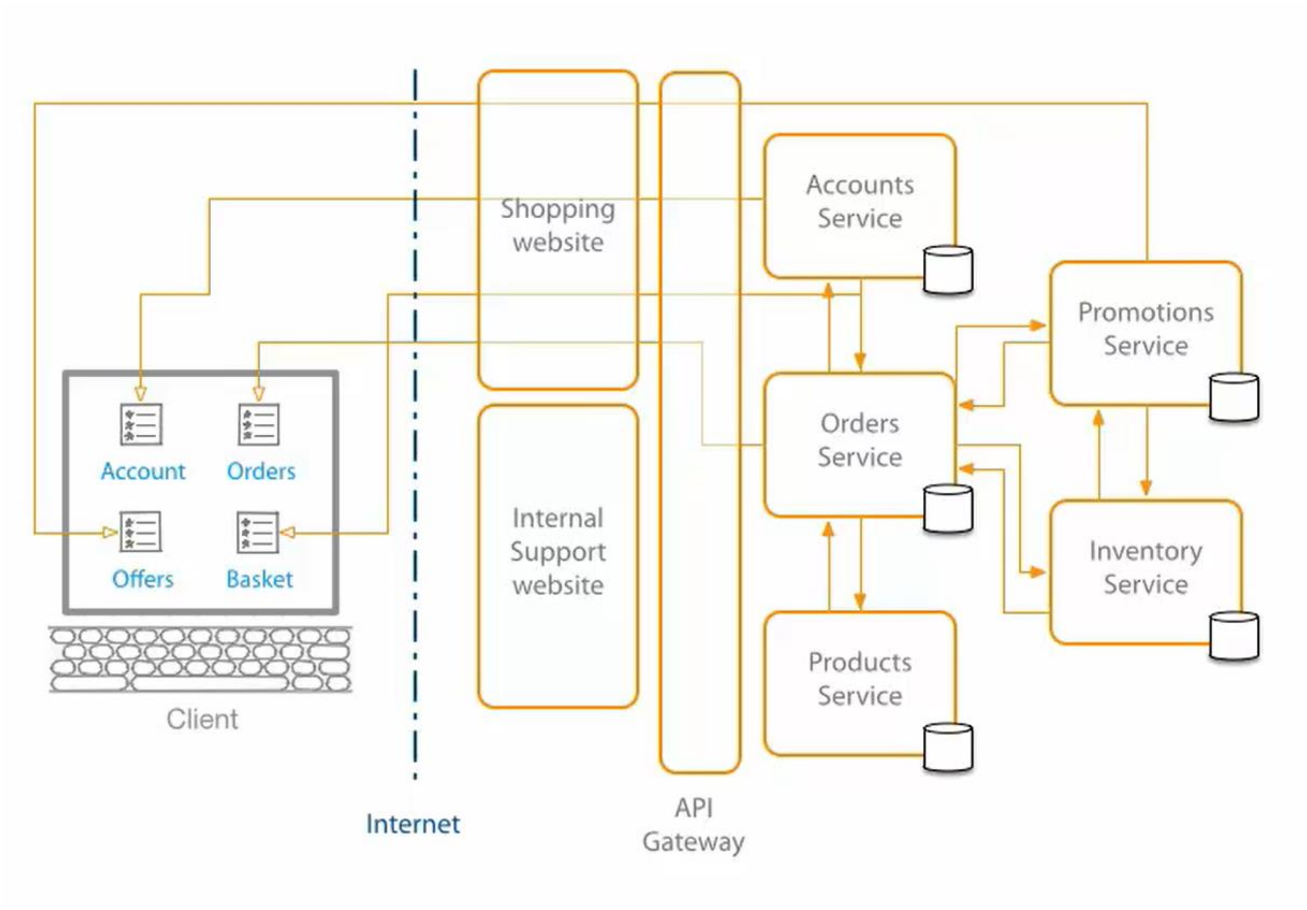
RBAC with AAD.
Pod Identity.
Secrets with Keyvault.
….

Scan images and runtime
Leverage Linux Capabilities
Pod security policy
Pod security context
….

A Monolith?


Account
Orders
Offers
Basket
Customer
Internet
Shopping website
Accounts
Orders
Promotions
Inventory
Products
Data Access
Database
UI Layer
Business Layer
Data Layer

Microservices…

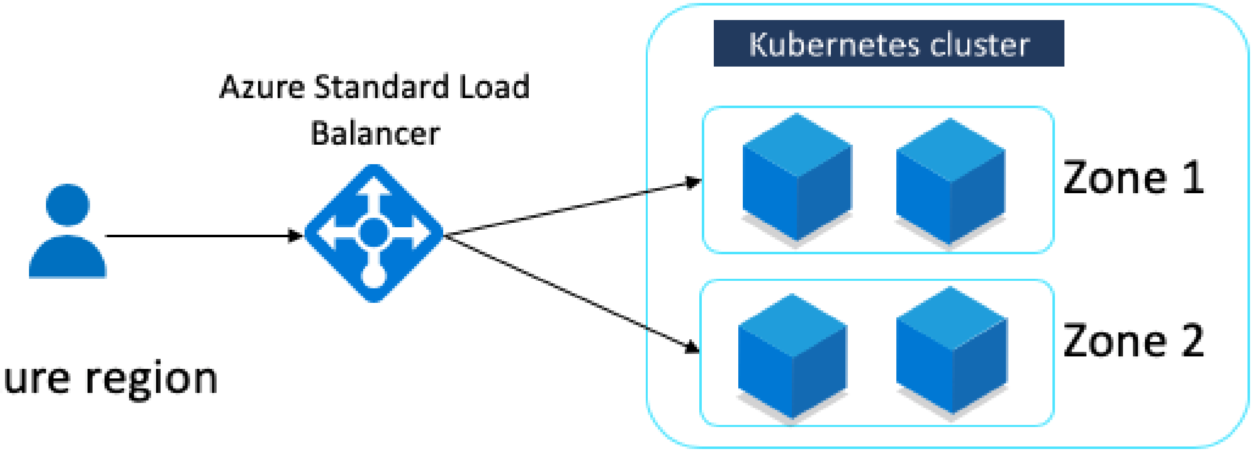


General Availability

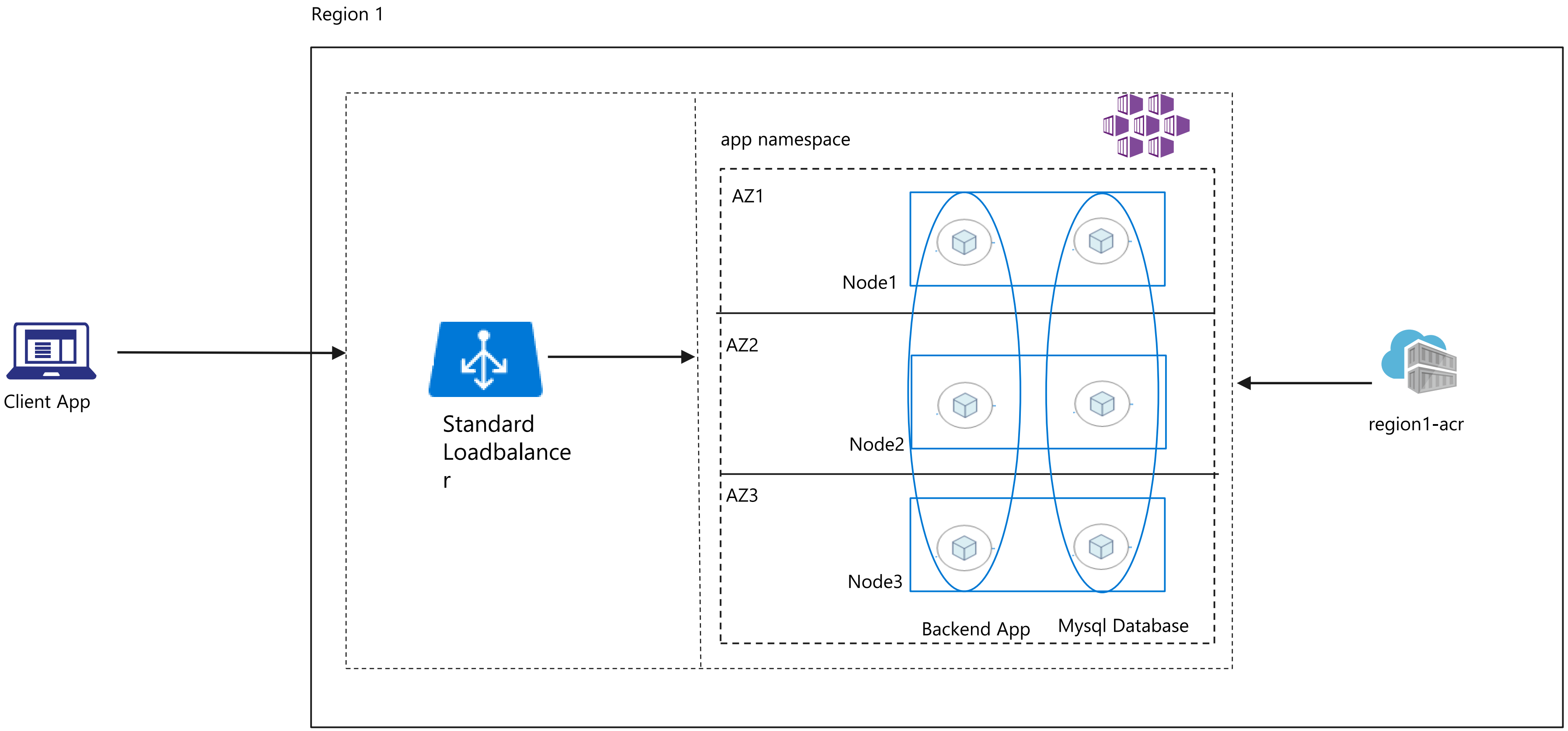
Availability Zones

Create an AKS cluster with nodes distributed across Availability Zones

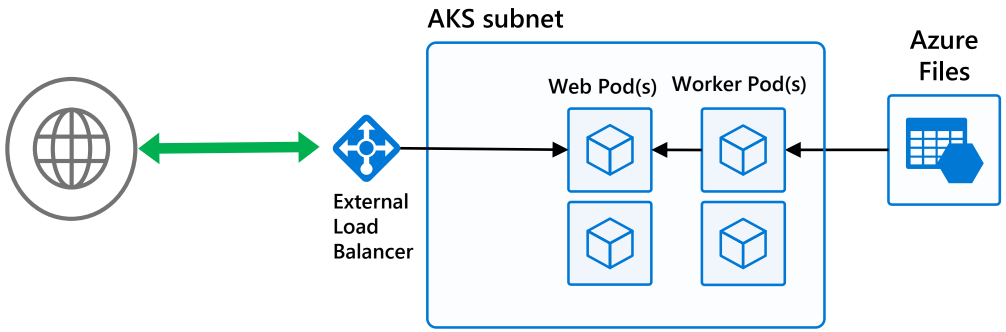
- An AZ is a unique physical location within an Azure region
- Provide a higher level of availability to your applications (99,99%)
- Note that regular Azure Disks are tied to a Zone
- Limited to regions that support Zones (10 regions now)
- Requires Standard Load Balancer SKU (Basic SKU does not support Zones)



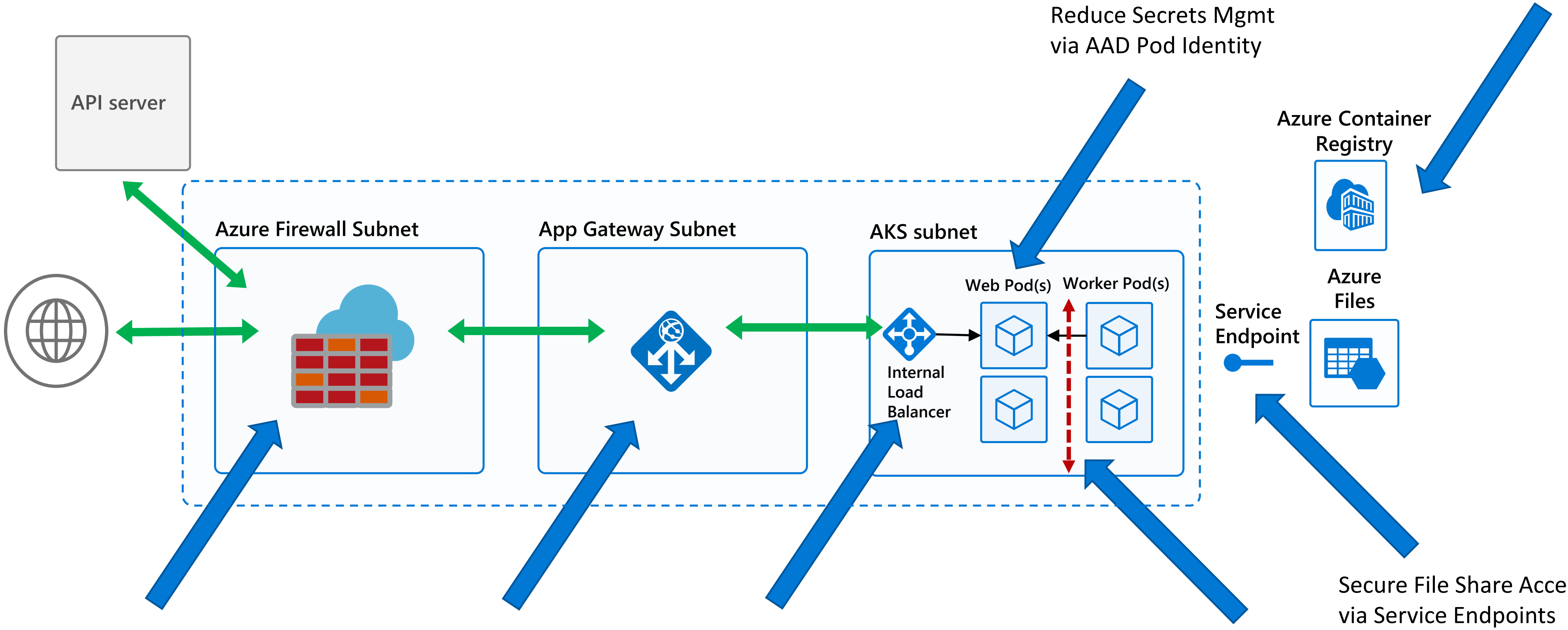
Our Application – with AZs



'AZ AKS CREATE' | App Before



App After



Secure Registry
via Service Endpoints
(Coming Soon)

Reduce Secrets Mgmt
via AAD Pod Identity

Control Outbound Traffic
via Azure Firewall

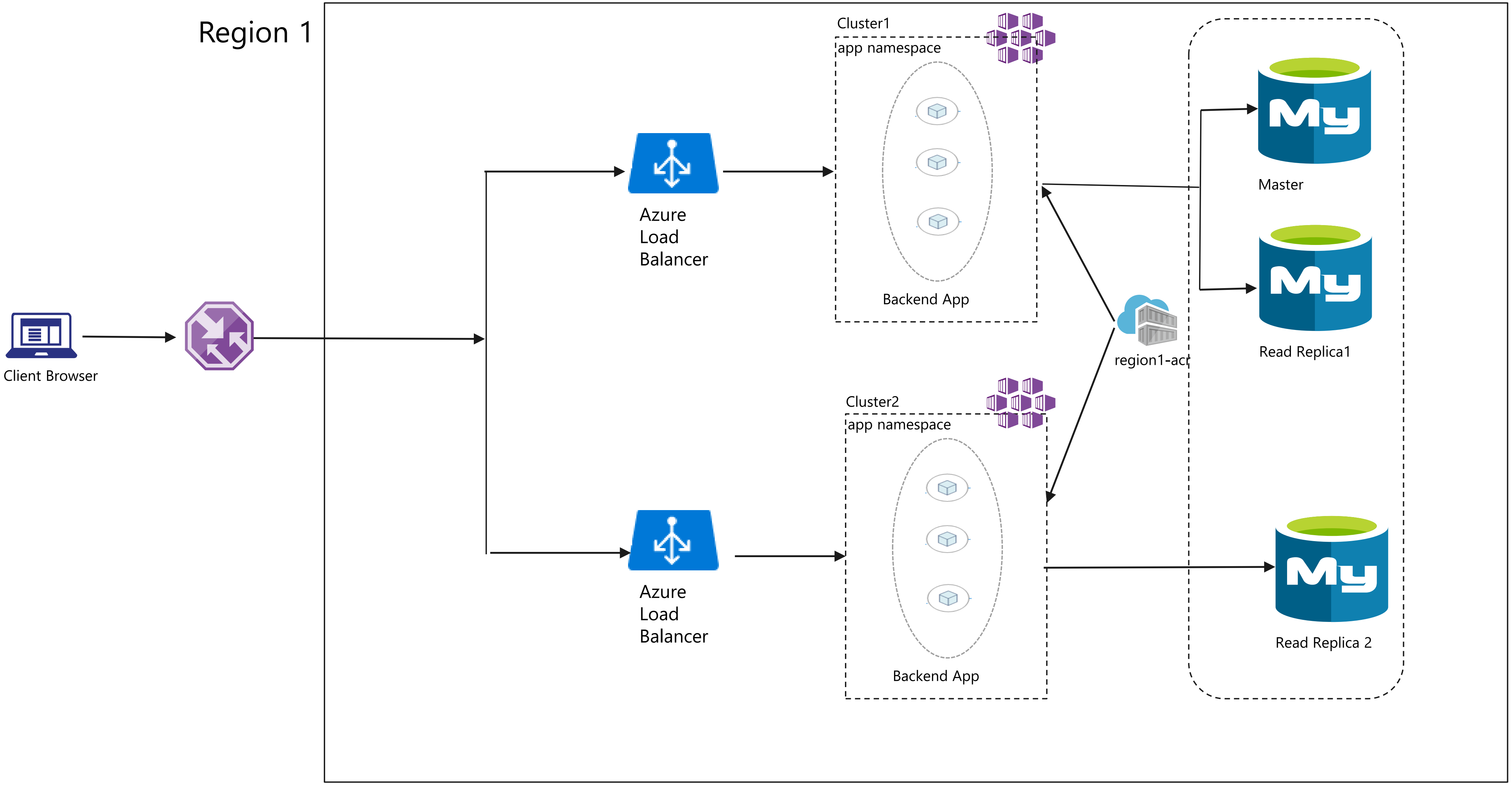
Adding WAF & Ingress
via App Gateway

No Public IP via
Internal LB

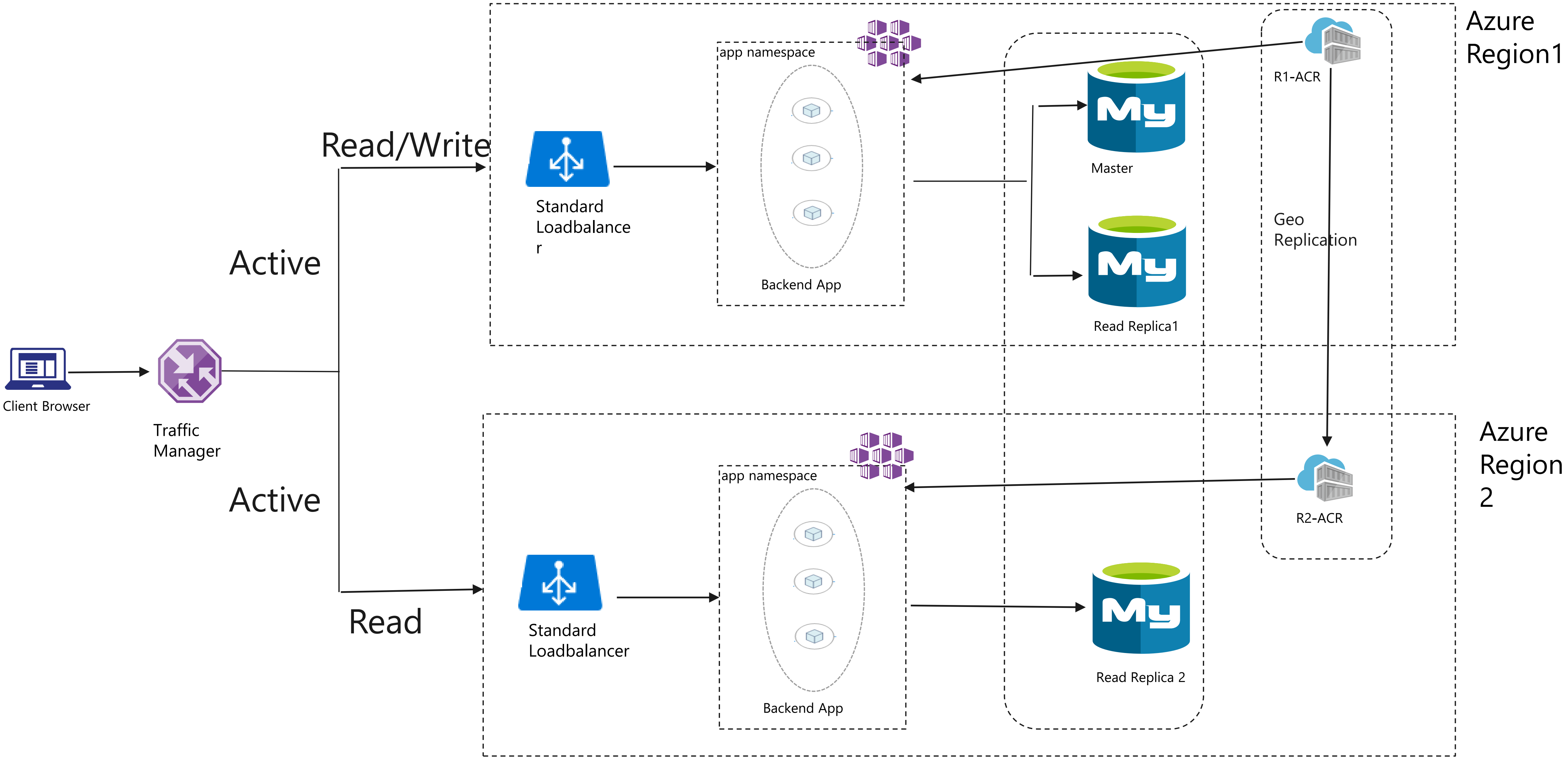
Network Segmentation
via Network Policy

Secure File Share Access
via Service Endpoints

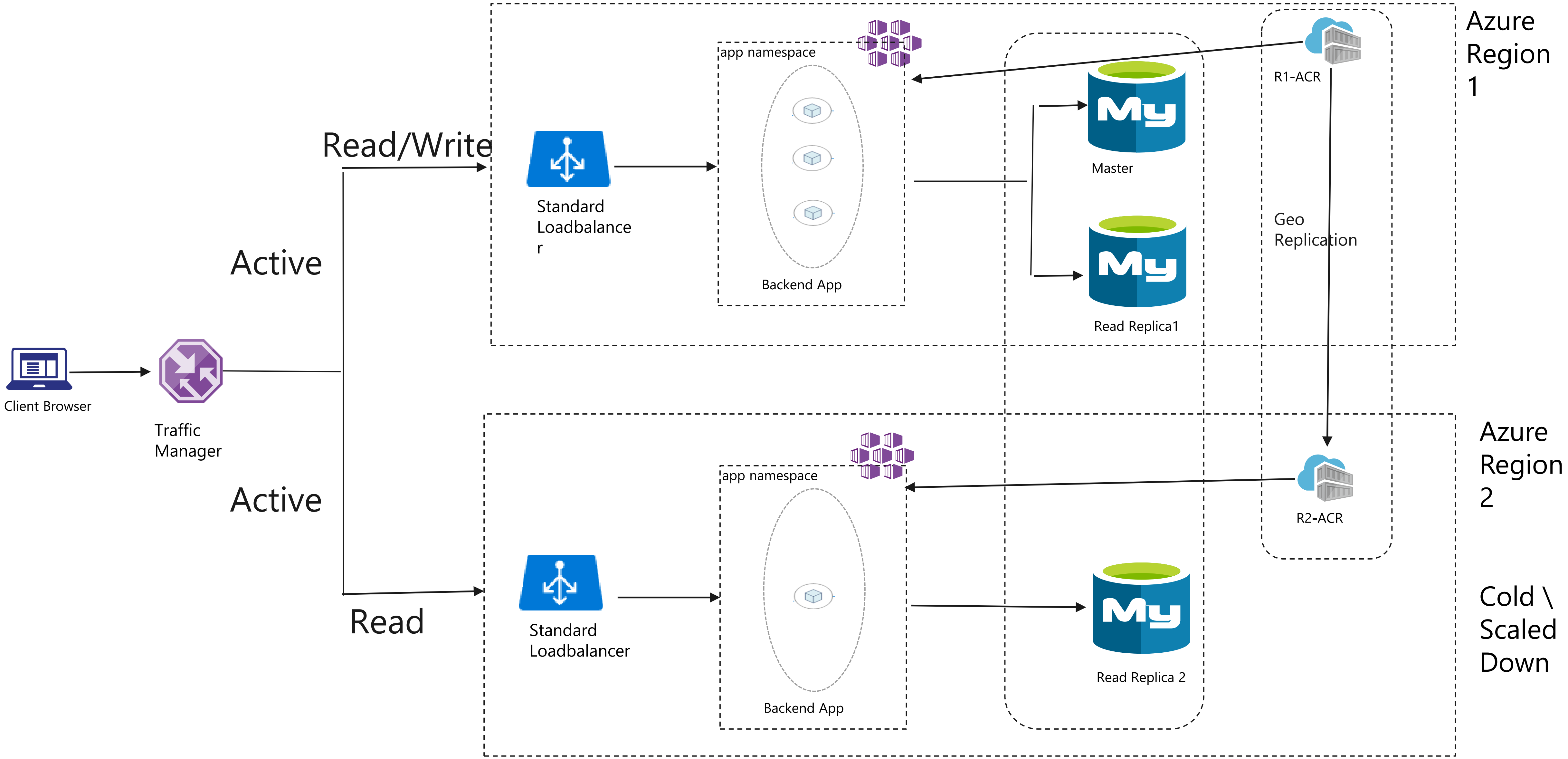
Multiple Clusters - Same Region



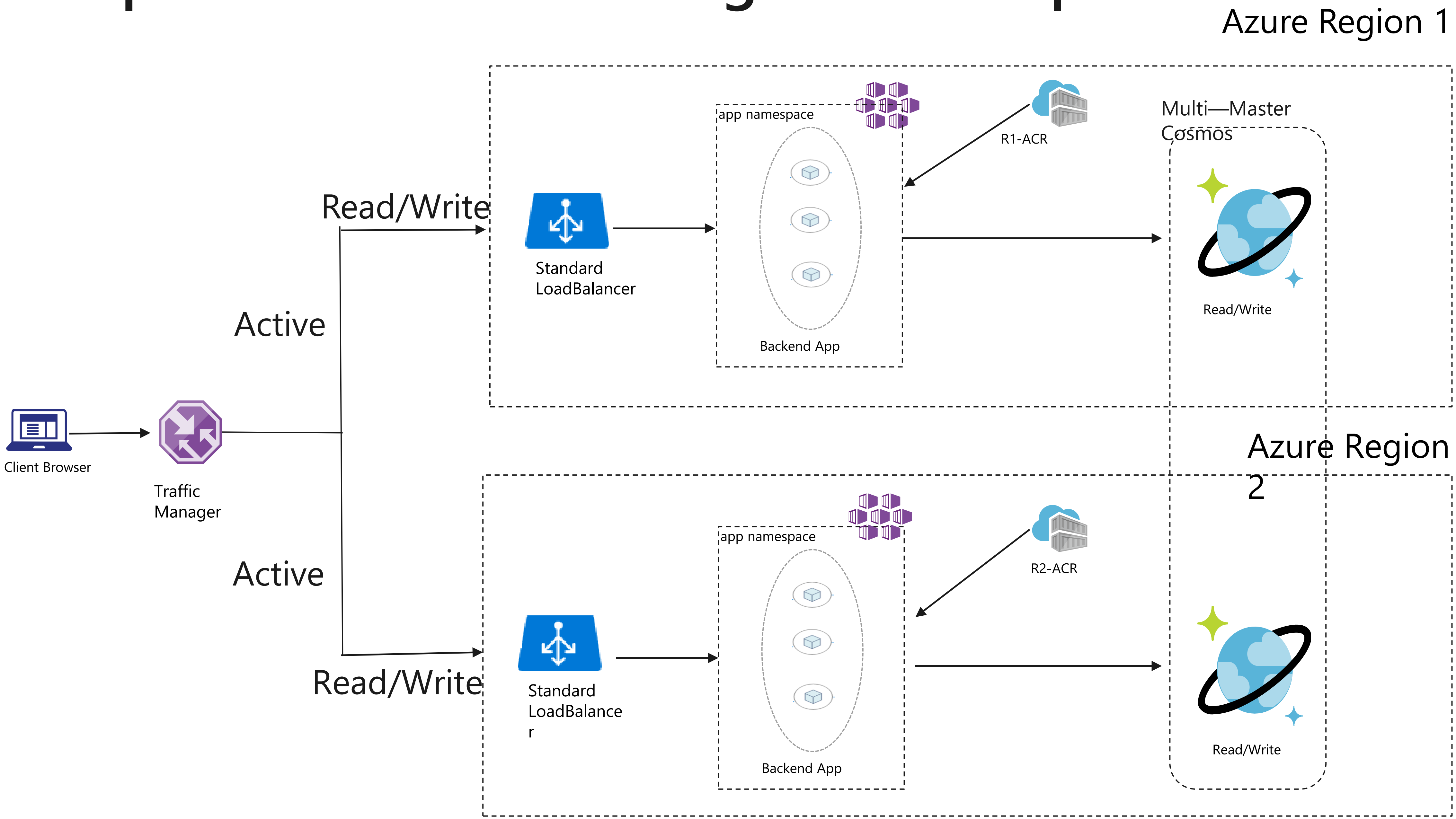
Multiple Clusters - Cross Region – One Master - Hot



Multiple Clusters - Cross Region – Cold/Scaled Down



Multiple Clusters - Cross Region – Multiple Masters



NodePool considerations

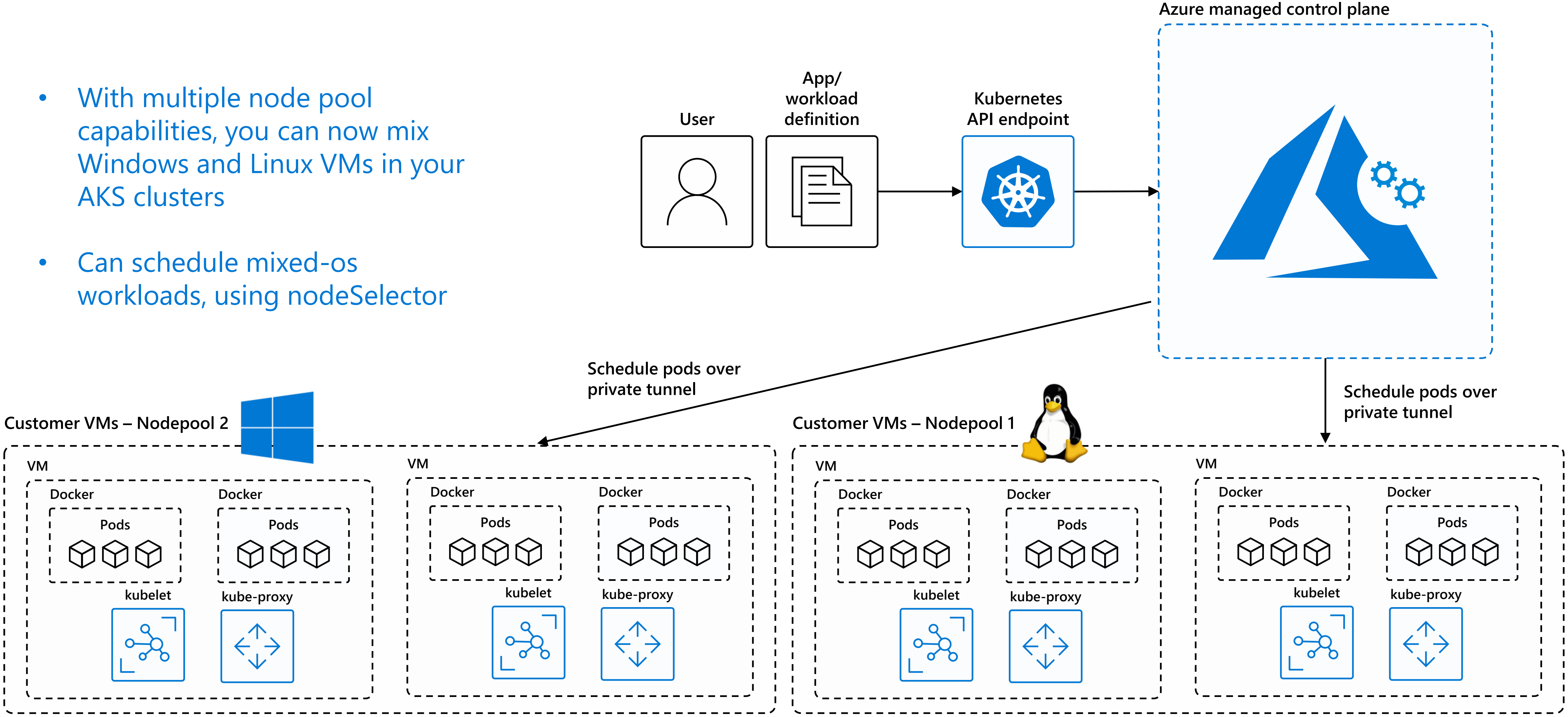
- AKS API operations will be decoupled for control plane and node pools.
- Most operations are now at node pool level (scale, upgrade,…)
- You can add taints to the node pool profile that will automatically add them to every new node
- Cluster AutoScaler works on a per node pool basis
- An AKS cluster can have a maximum of 8 node pools
- An AKS cluster can have a maximum of 400 nodes across those node pools
- You can leverage the Public IP per Node feature in selected node

Manage Kubernetes with ease

Windows Server Containers

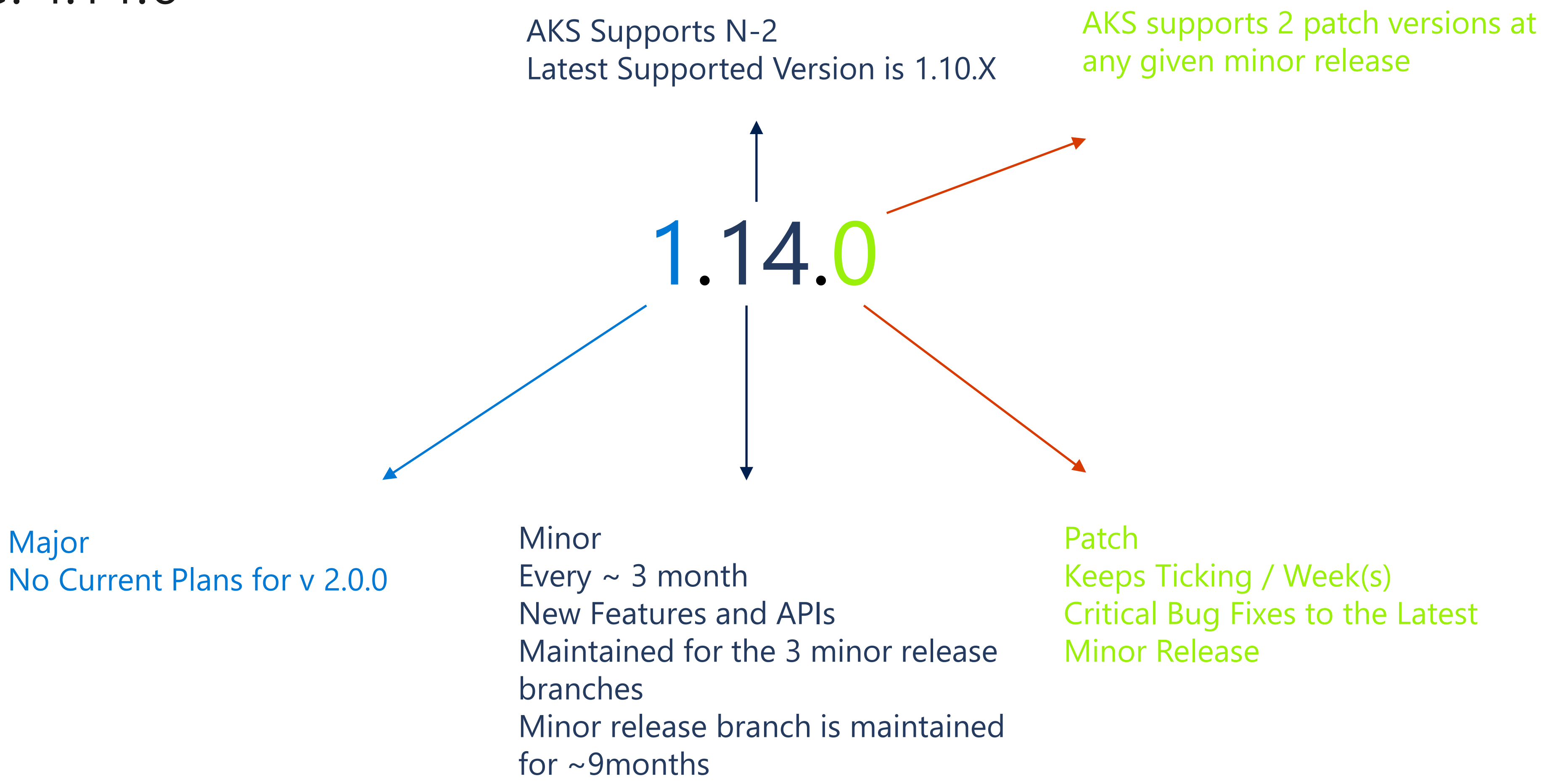
- With multiple node pool capabilities, you can now mix Windows and Linux VMs in your AKS clusters

- Can schedule mixed-os workloads, using nodeSelector


Azure managed control plane
User
App/ workload definition
Kubernetes API endpoint
Schedule pods over private tunnel
Schedule pods over private tunnel
Customer VMs – Nodepool 2
Customer VMs – Nodepool 1
VM
Docker
Pods
kubelet
kube-proxy

Kubernetes Versions

- X.Y.Z i.e. 1.14.0



AKS Supports N-2
Latest Supported Version is 1.10.X

AKS supports 2 patch versions at any given minor release

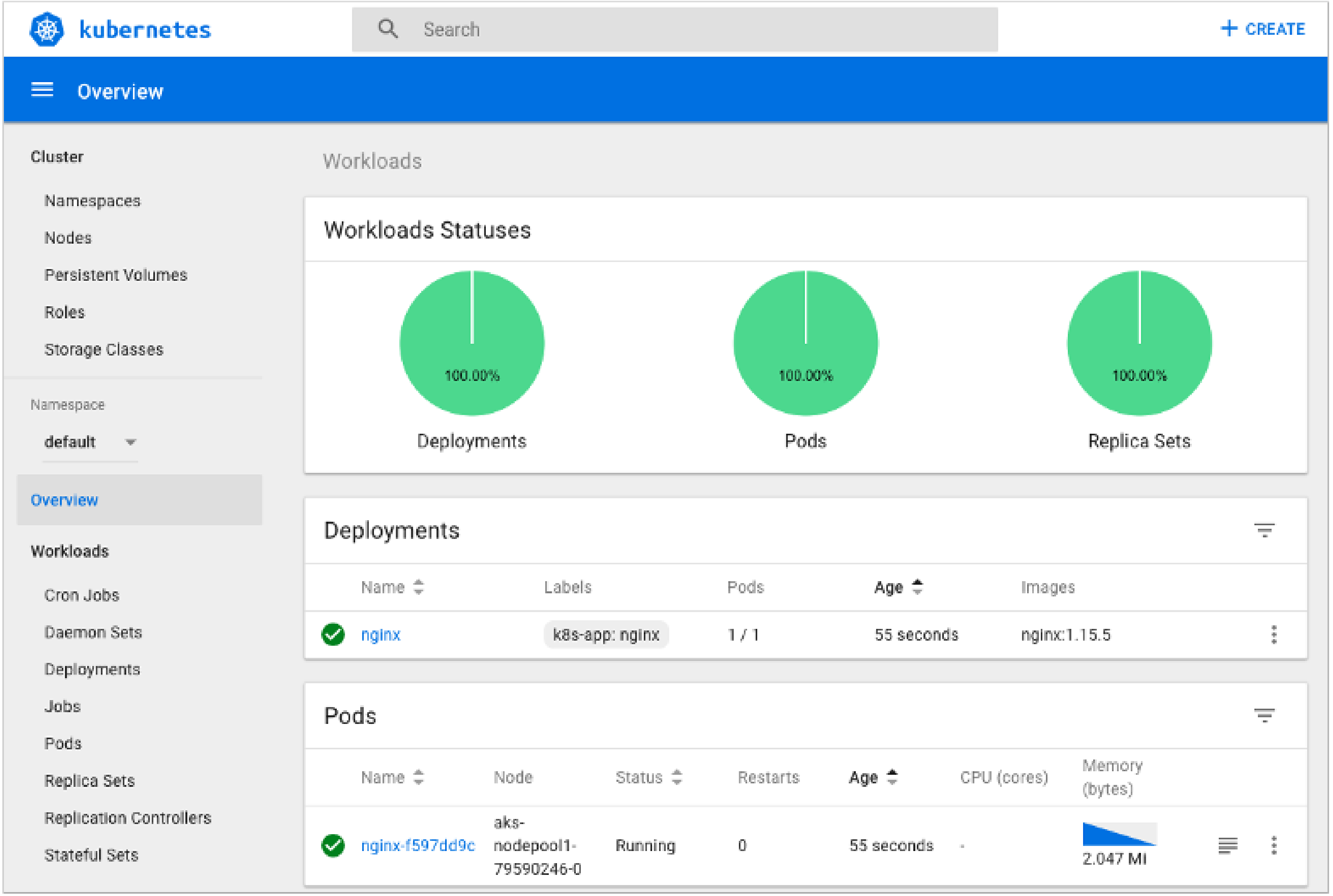
1.14.0

Major
No Current Plans for v 2.0.0

Minor
Every ~ 3 month
New Features and APIs
Maintained for the 3 minor release branches
Minor release branch is maintained for ~9months

Patch
Keeps Ticking / Week(s)
Critical Bug Fixes to the Latest Minor Release

Kubernetes Dashboard



Deployments

	Name	Labels	Pods	Age	Images	
✓	nginx	k8s-app: nginx	1 / 1	55 seconds	nginx:1.15.5	⋮

Pods

	Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)		
✓	nginx-f597dd9c	aks-nodepool1-79590246-0	Running	0	55 seconds	-	2.047 Mi	≡	⋮

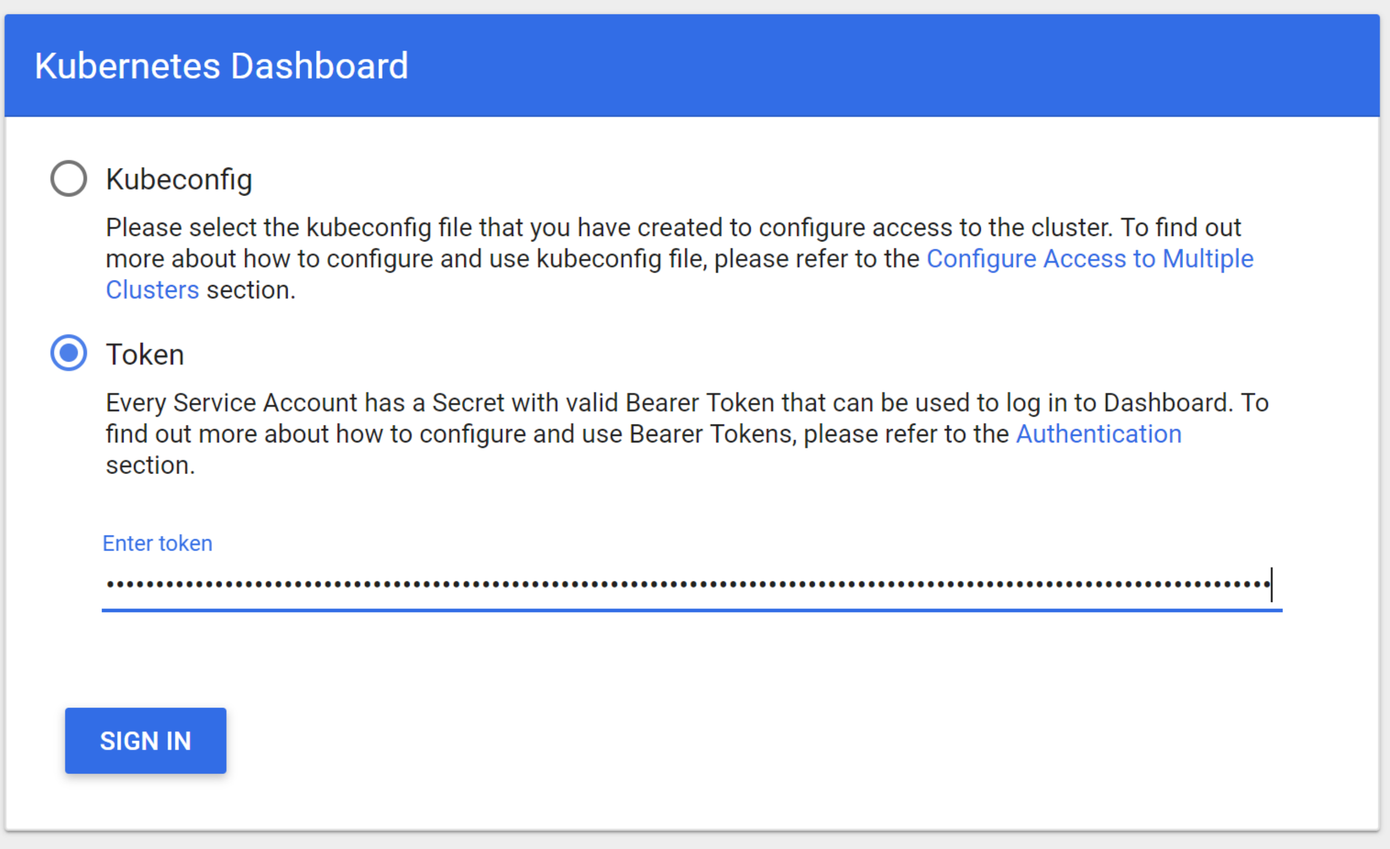
Disable the Dashboard (Recommended)

```
$ kubectl get pods -n kube-system | grep "dashboard"
kubernetes-dashboard-cc4cc9f58-whmhv   1/1     Running   0          30d

$ az aks disable-addons -a kube-dashboard -g k8s-demo -n k8s-demo-rbac

$ kubectl get pods -n kube-system | grep "dashboard"
```

Kubernetes Dashboard – less worse



Kubernetes Dashboard

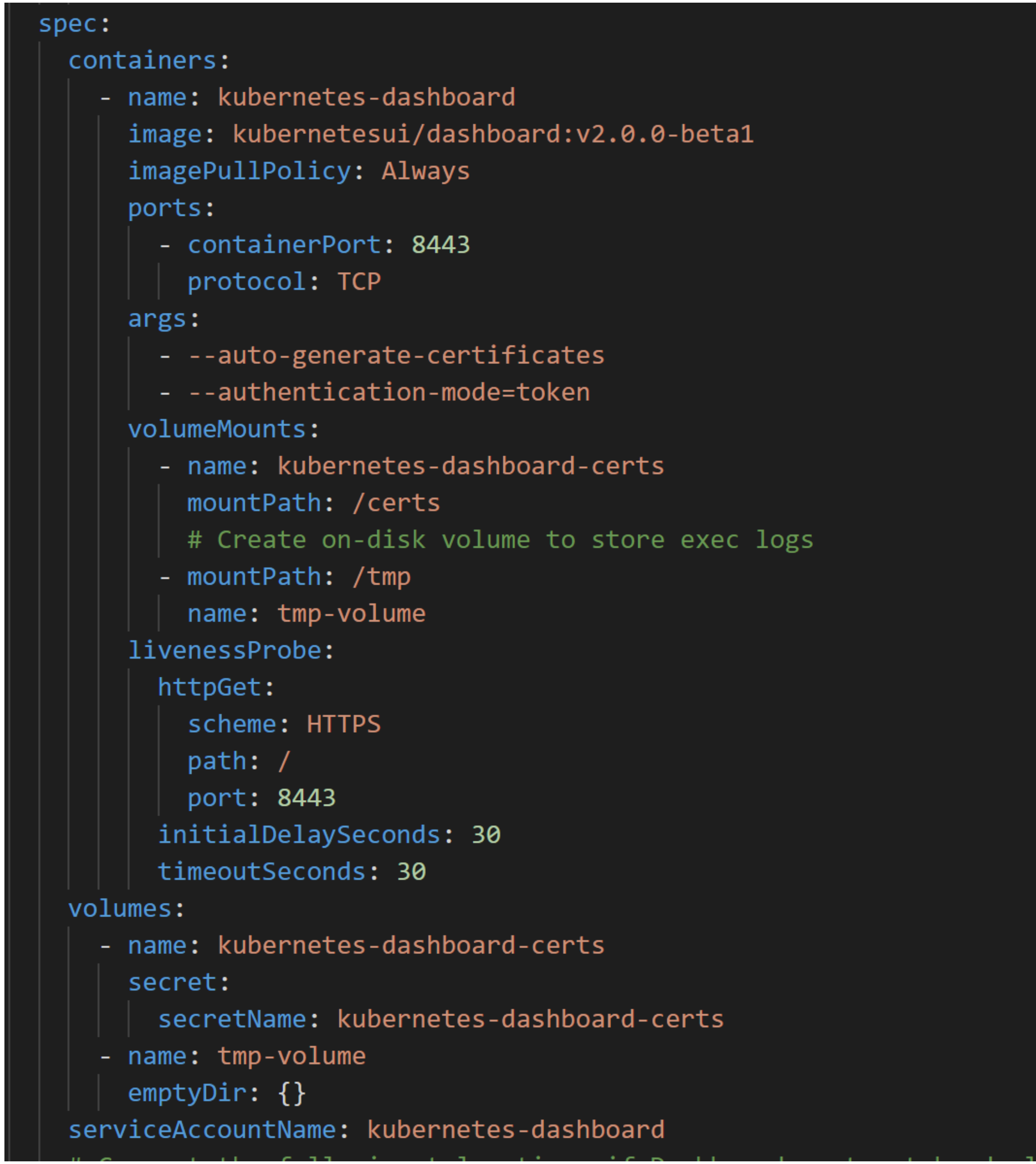
○ Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

● Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

Enter token



```yaml
spec:
  containers:
    - name: kubernetes-dashboard
      image: kubernetesui/dashboard:v2.0.0-beta1
      imagePullPolicy: Always
      ports:
        - containerPort: 8443
          protocol: TCP
      args:
        - --auto-generate-certificates
        - --authentication-mode=token
      volumeMounts:
        - name: kubernetes-dashboard-certs
          mountPath: /certs
          # Create on-disk volume to store exec logs
        - mountPath: /tmp
          name: tmp-volume
      livenessProbe:
        httpGet:
          scheme: HTTPS
          path: /
          port: 8443
        initialDelaySeconds: 30
        timeoutSeconds: 30
  volumes:
    - name: kubernetes-dashboard-certs
      secret:
        secretName: kubernetes-dashboard-certs
    - name: tmp-volume
      emptyDir: {}
  serviceAccountName: kubernetes-dashboard
```

Advanced networking

Uses the Azure CNI (Container Networking Interface)

CNI is a vendor-neutral protocol, used by container runtimes to make requests to Networking Providers

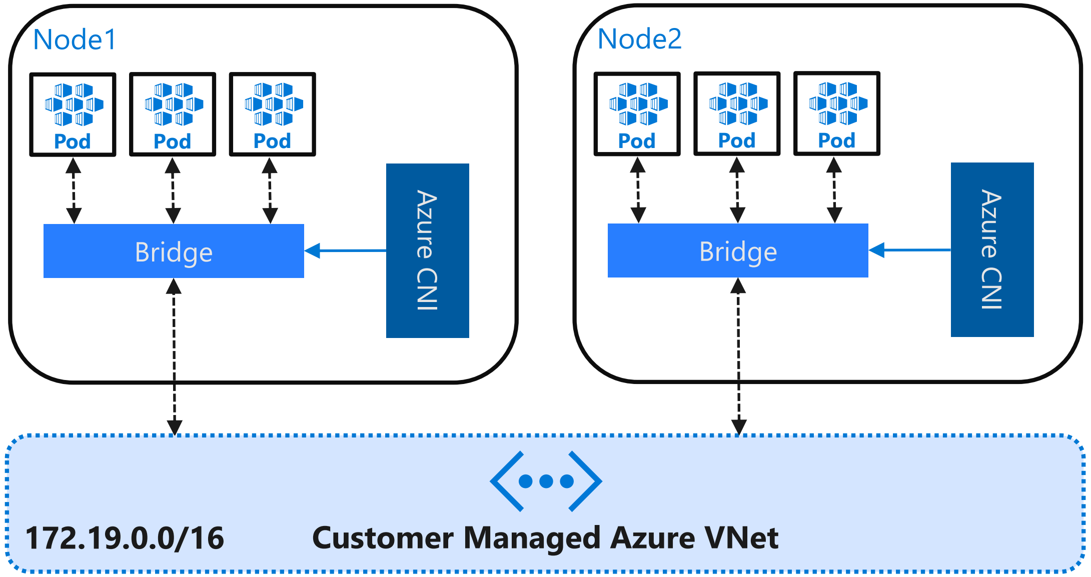
Azure CNI is an implementation which allows you to integrate Kubernetes with your VNET

Advantages

Single IP CIDR to manage

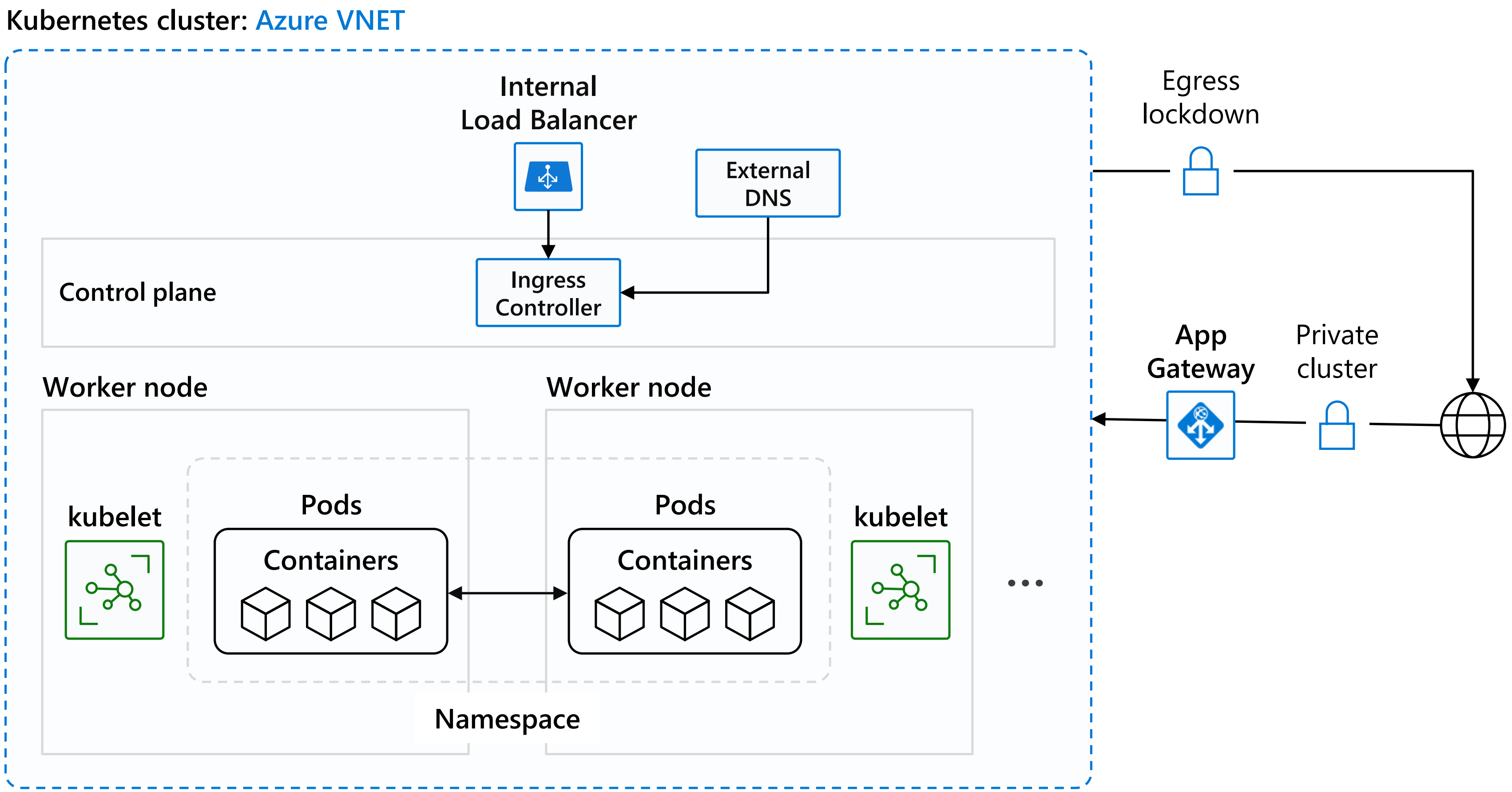
Better Performance

Peering and On-Premise connectivity is out of the box



Networking

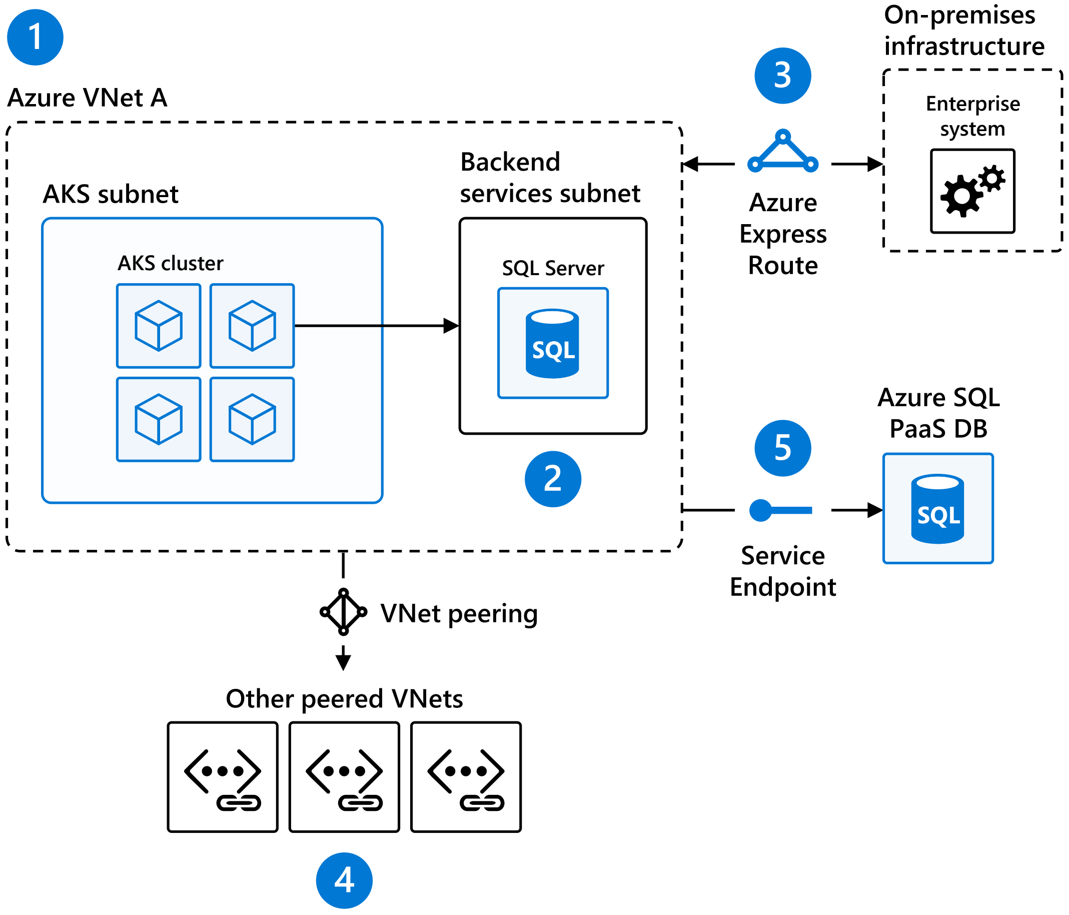
Secure your Kubernetes workloads with virtual network and policy-driven communication paths between resources



Scenarios enabled by Advanced Networking

1. Uses Azure subnet for both your containers and cluster VMs
2. Allows for connectivity to existing Azure services in the same VNet
3. Use Express Route to connect to on-premises infrastructure
4. Use VNet peering to connect to other VNets
5. Connect AKS cluster securely and privately to other Azure resources using VNet endpoints

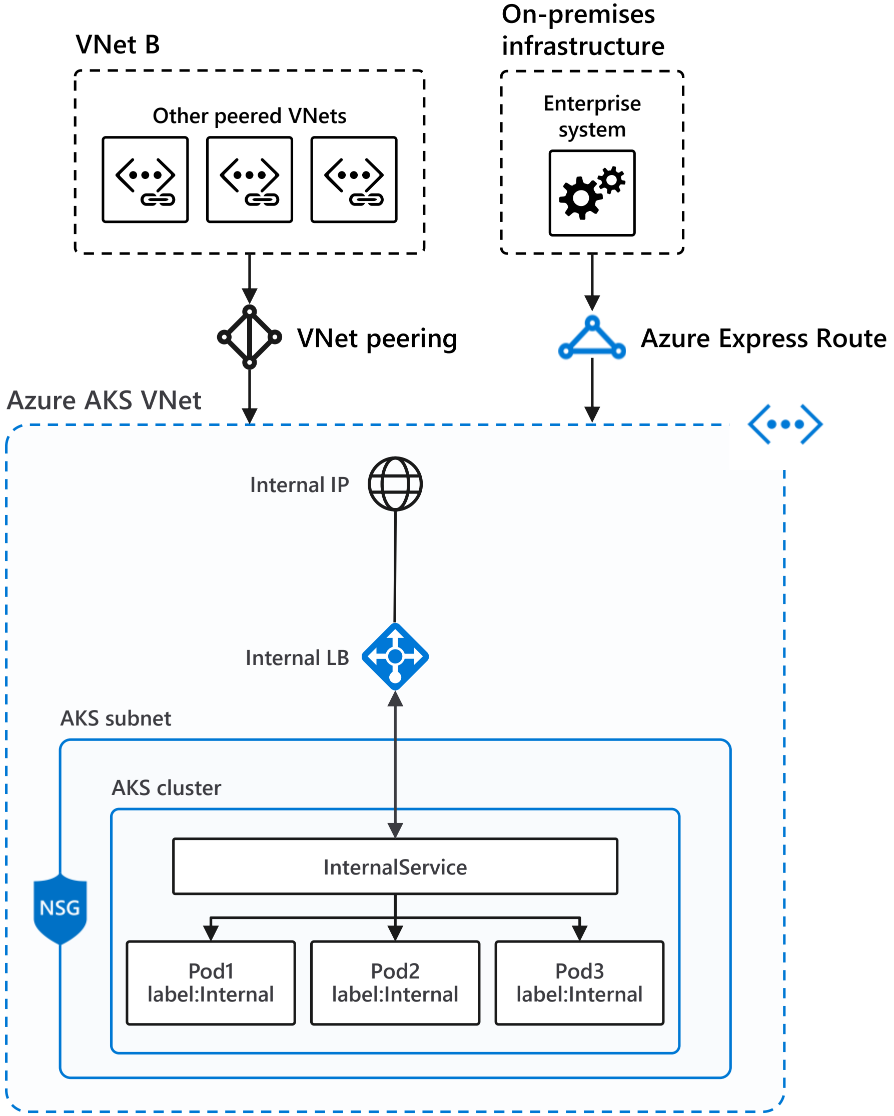
AKS VNet integration works seamlessly with your existing network infrastructure



Internal Service

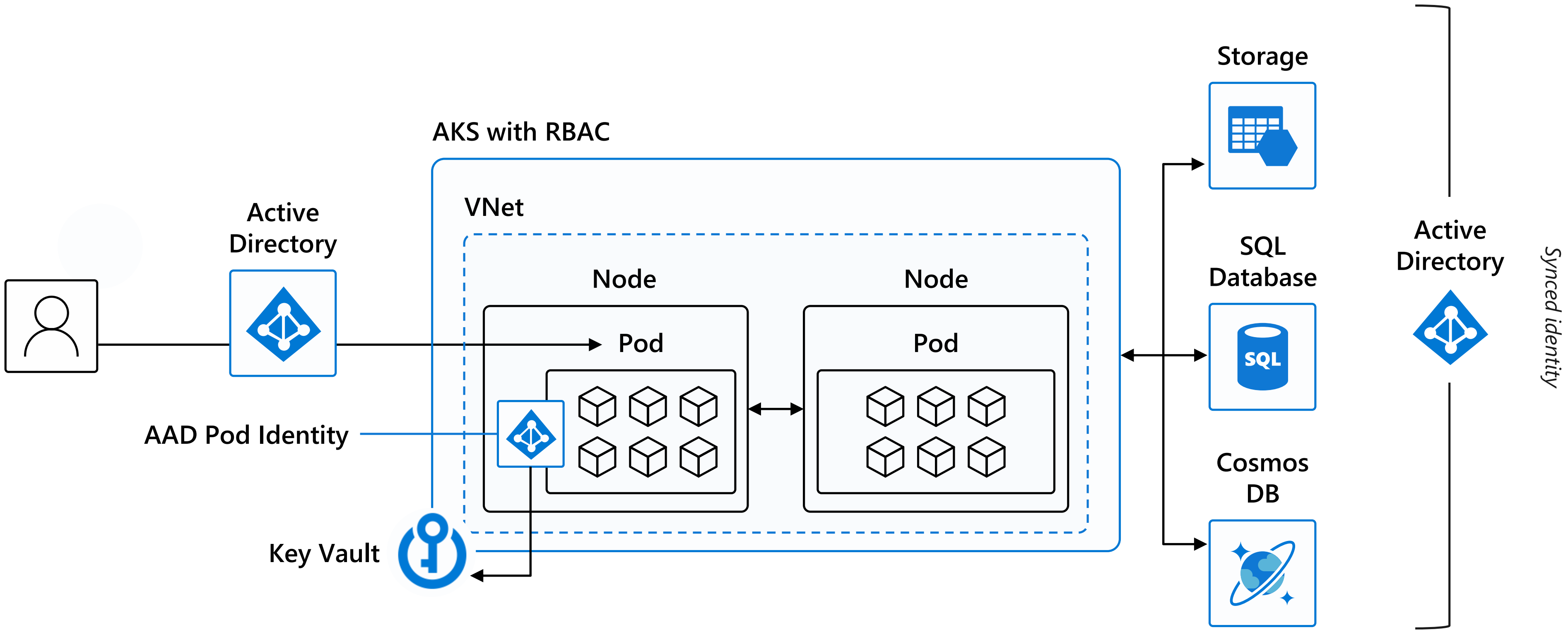
- Used for internal services that should be accessed by other VNETs or On-Premise only

```
apiVersion: v1
kind: Service
metadata:
  name: internalservice
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-internal:
"true"
spec:
   type: LoadBalancer
   loadBalancerIP: 10.240.0.25
   ports:
   - port: 80
   selector:
    app: internal
```



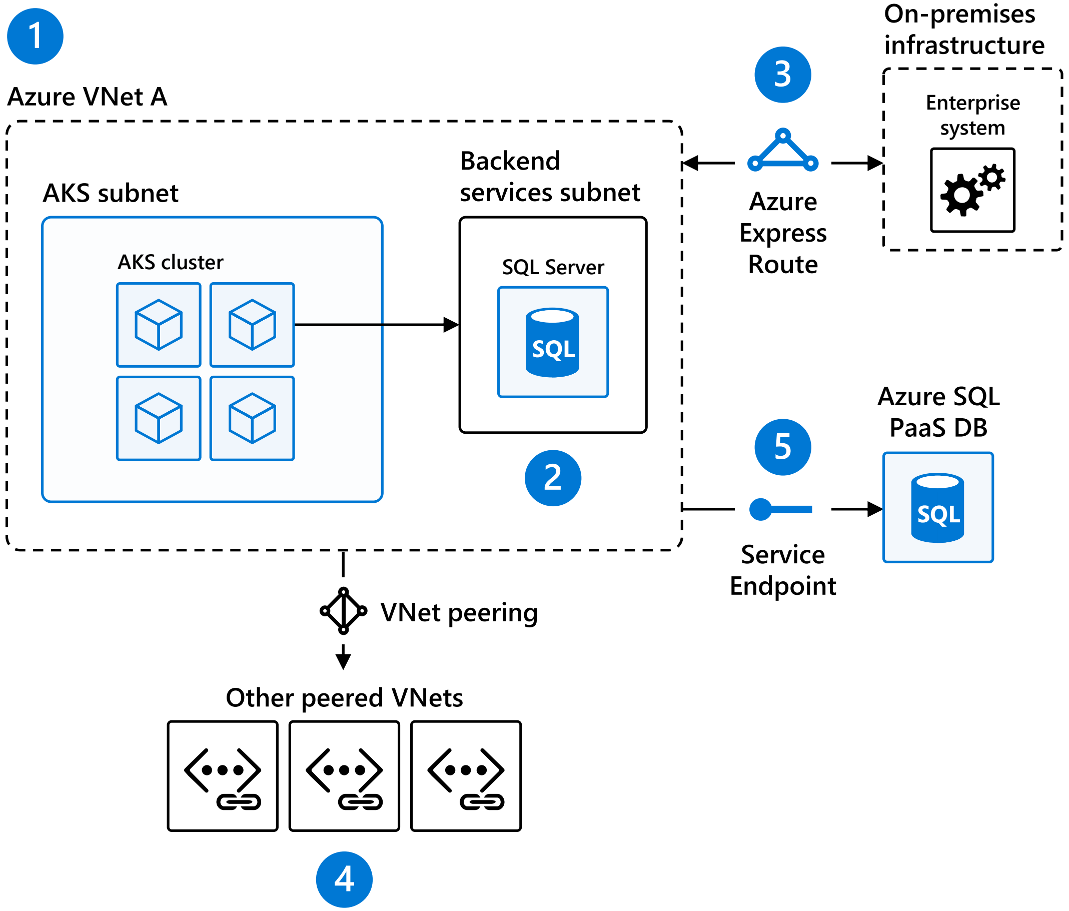
Identity

Use familiar tools like AAD for fine-grained identity and access control to Kubernetes resources from cluster to containers



Secure network communications with VNET and CNI

1. Uses Azure subnet for both your containers and cluster VMs
2. Allows for connectivity to existing Azure services in the same VNet
3. Use Express Route to connect to on-premises infrastructure
4. Use VNet peering to connect to other VNets
5. Connect AKS cluster securely and privately to other Azure resources using VNet endpoints



AKS VNet integration works seamlessly with your existing network infrastructure

Azure Pipelines build audit & enforcement using Azure Policy

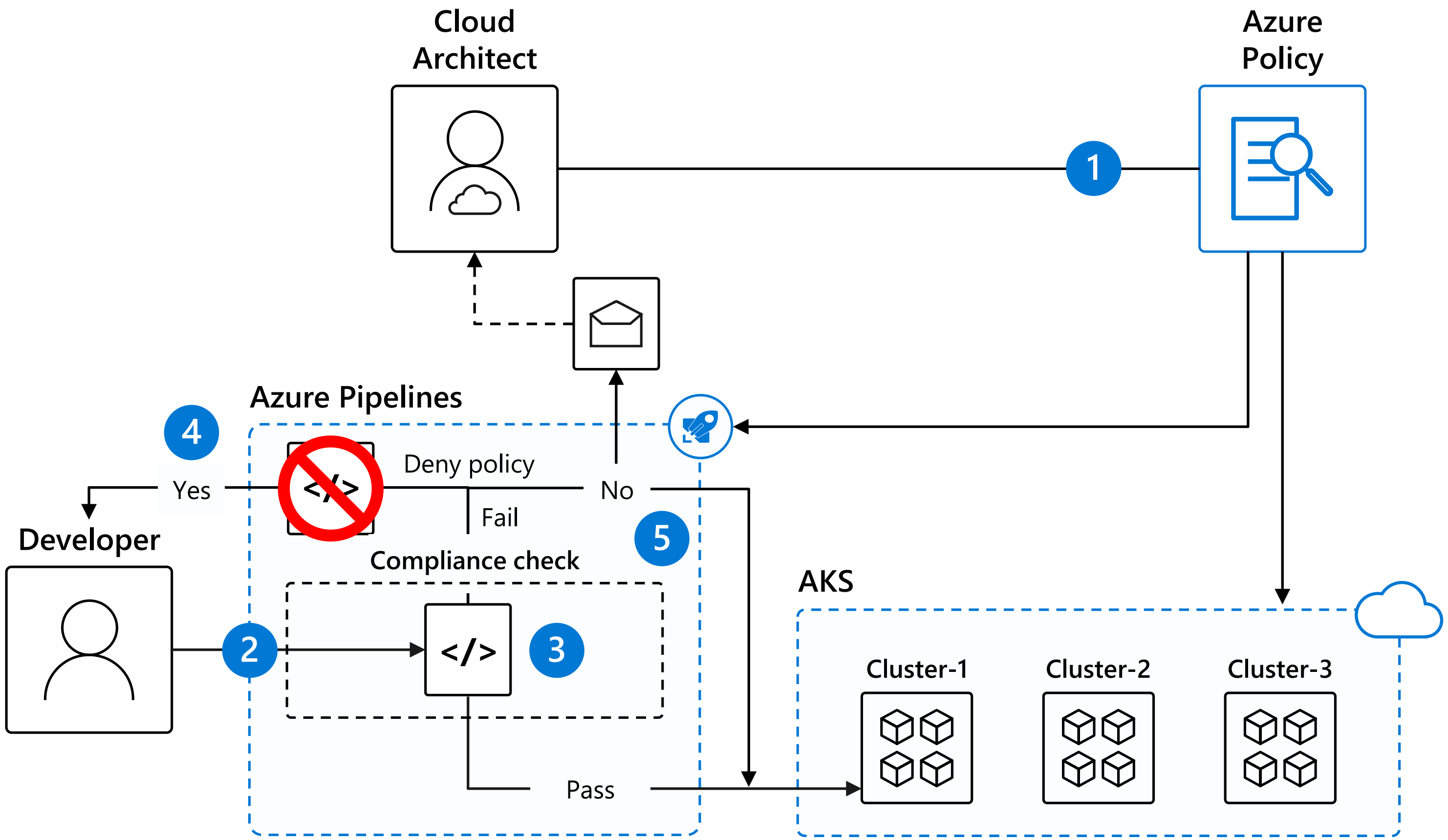
1. Cloud architect assigns a policy across clusters; policy can be set to block non-compliance (deny) or generate non-compliance warnings (audit)

2. Developer makes code change that kicks off an Azure Pipelines build

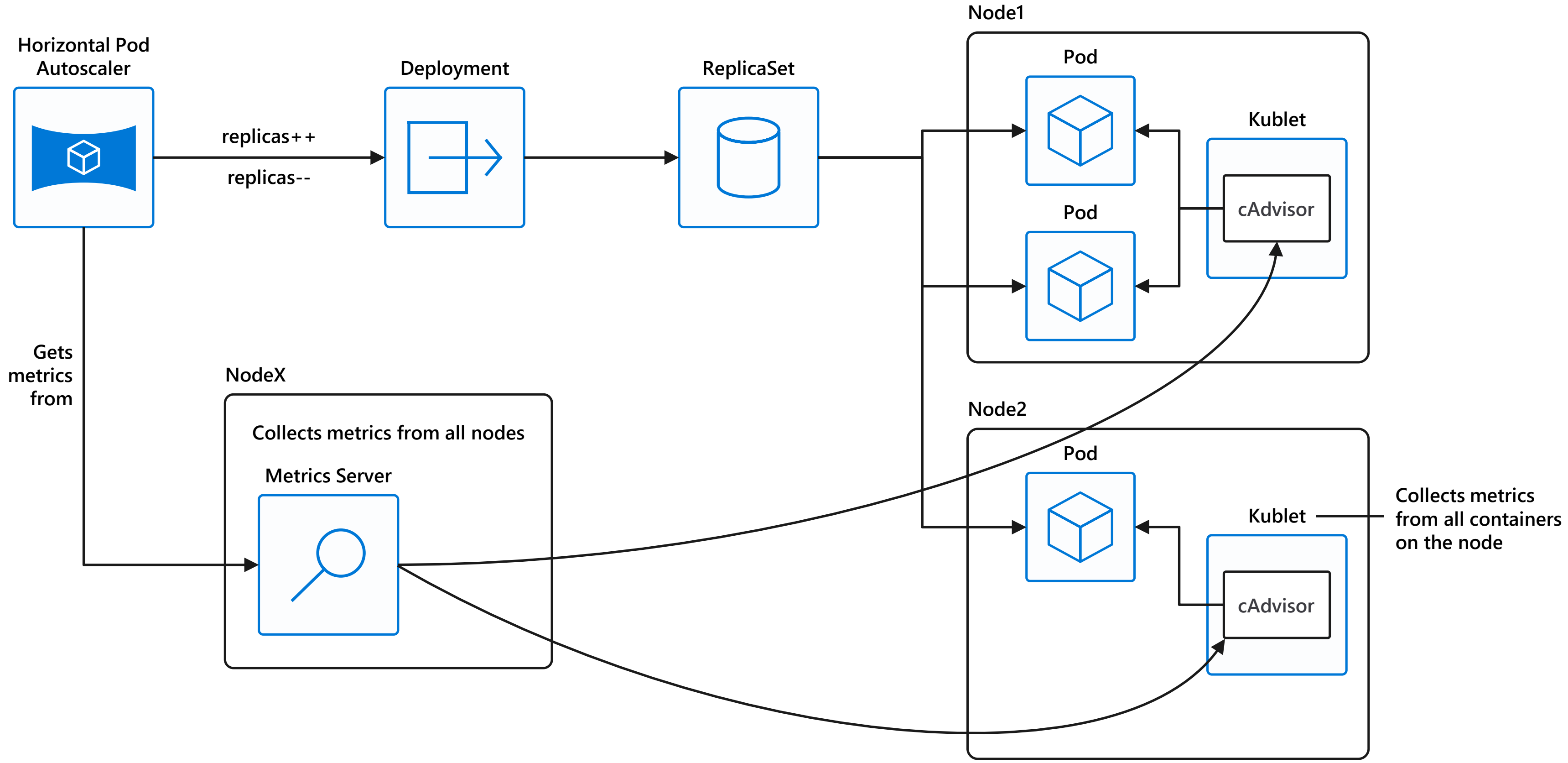
3. Azure Pipelines evaluates the request for policy compliance

4. If policy is set to deny, Azure Pipelines rejects the build attempt if any non-compliance is identified

5. If policy is set to audit, a non-compliance event is logged and the build is allowed to proceed

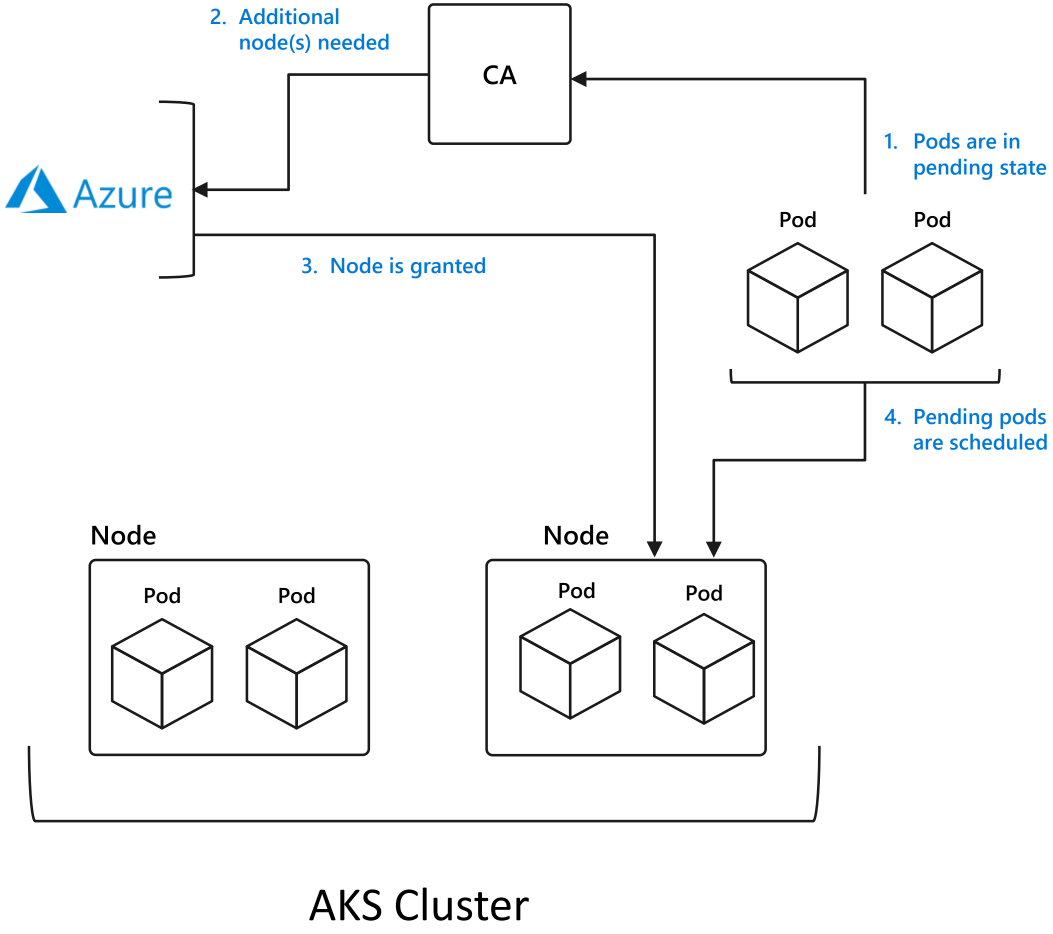


How HPA works?

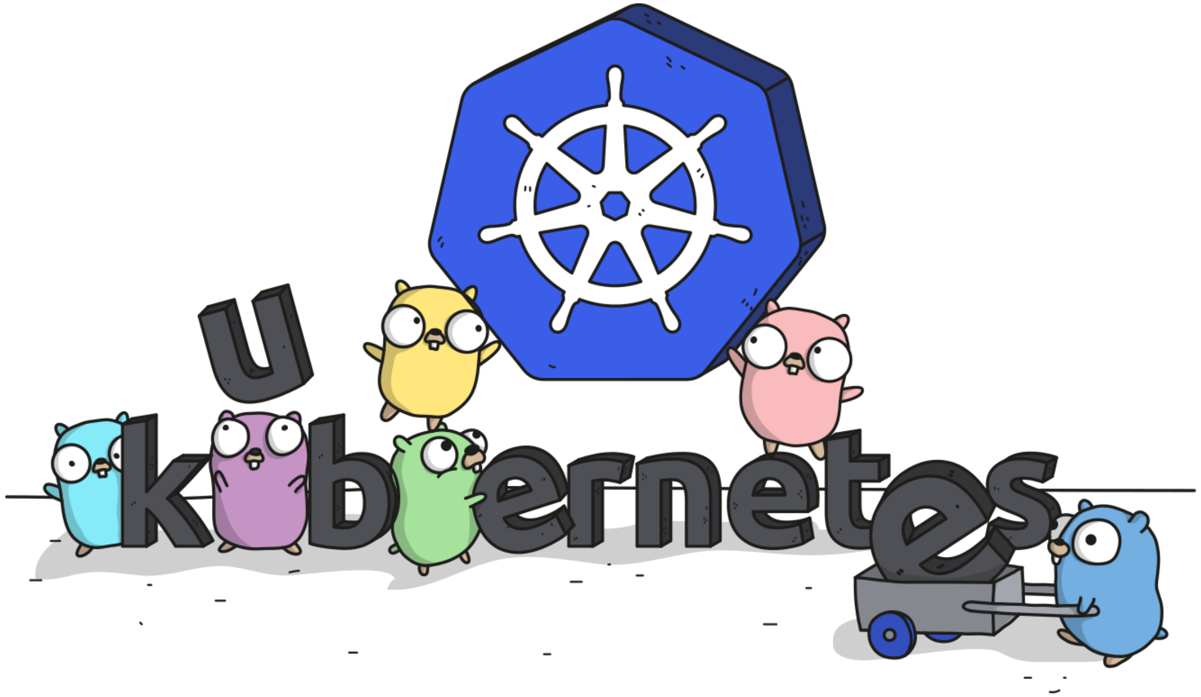


Cluster Autoscaler

- Scales nodes based on pending pods
- Scale up and scale down
- Reduces dependency on monitoring*
- Removes need for users to manage nodes and monitor service usage manually


2. Additional node(s) needed
CA
Azure
1. Pods are in pending state
Pod
Pod
3. Node is granted
4. Pending pods are scheduled
Node
Pod
Pod
Node
Pod
Pod
AKS Cluster

Q/A

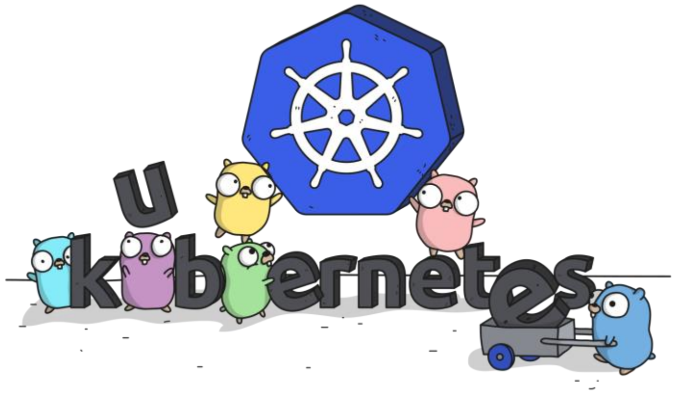


Gustav Kaleta

Global Black Belt
Tech Lead EMEA
Microsoft

Twitter: @kaletaii
email: gkaleta@

AKS + K8s + Containers Best Practices

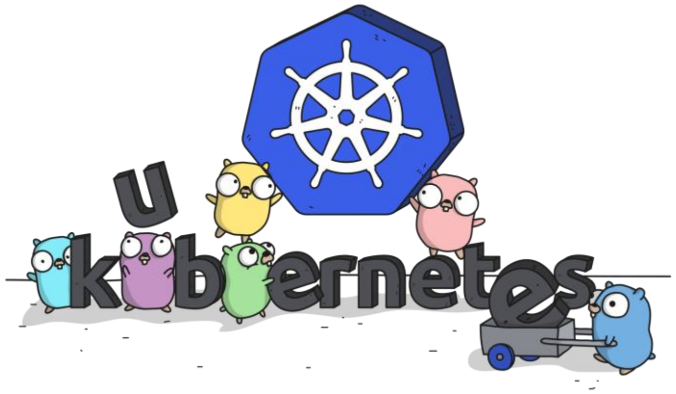


Goto these links below first, for the latest best practices.

This guide supplements and adds in additional best practice guidance.

https://aka.ms/aks/best-practices-sessions Operational best practices for Azure Kubernetes Service
https://github.com/Azure/aks-bestpractices-ignite19
https://www.youtube.com/watch?v=RJJ4CUyja6M
https://docs.microsoft.com/en-us/azure/aks/faq
https://docs.microsoft.com/en-us/azure/aks/best-practices
https://github.com/Azure/k8s-best-practices
https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-cluster-security
https://docs.microsoft.com/en-us/azure/aks/troubleshooting
https://docs.microsoft.com/en-us/azure/container-registry/container-registry-best-practices

AKS + K8s + Containers Best Practices



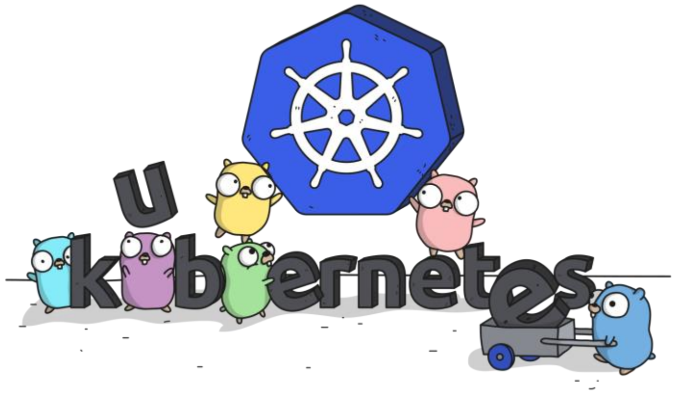
Best practices for cluster isolation in AKS
https://docs.microsoft.com/azure/aks/operator-best-practices-cluster-isolation

Best practices for business continuity and disaster recovery in Azure Kubernetes Service (AKS)
https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-multi-region

Best practices for authentication and authorization in AKS
https://docs.microsoft.com/azure/aks/operator-best-practices-identity

Best practices for pod security in AKS
https://docs.microsoft.com/azure/aks/developer-best-practices-pod-security

AKS + K8s + Containers Best Practices

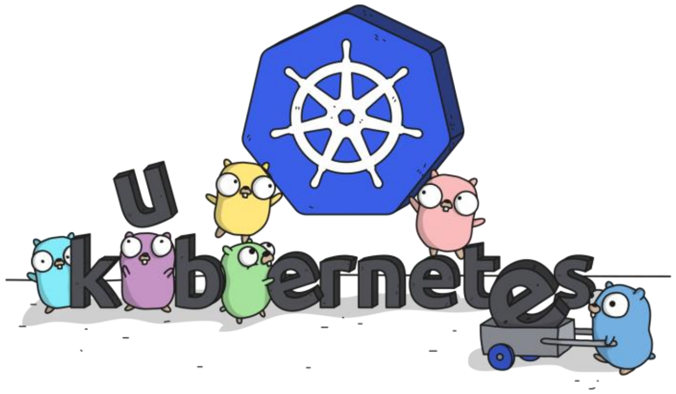


Best practices for business continuity and disaster recovery in AKS
https://docs.microsoft.com/azure/aks/operator-best-practices-multi-region

Best practices for container image management and security in AKS
https://docs.microsoft.com/azure/aks/operator-best-practices-container-image-management

Best practices for network connectivity and security in AKS
https://docs.microsoft.com/azure/aks/operator-best-practices-network

AKS + K8s + Containers Best Practices



Best practices for advanced scheduler features in AKS

https://docs.microsoft.com/azure/aks/operator-best-practices-advanced-scheduler

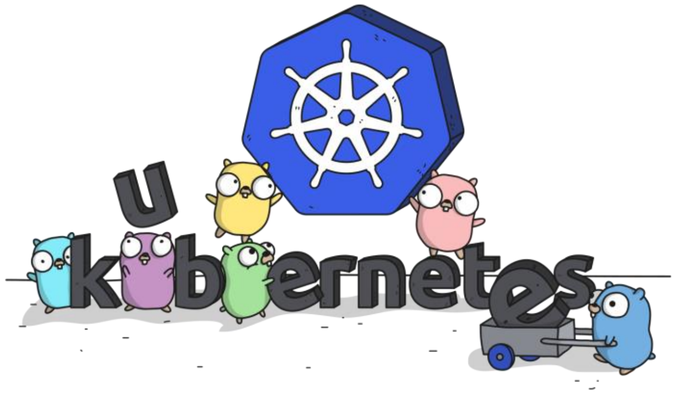
Best practices for storage and backups in AKS

https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-storage

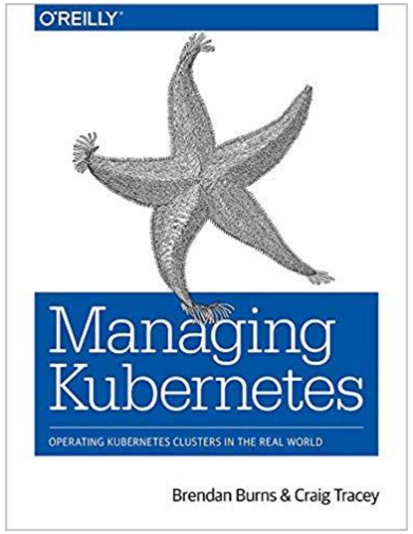
AKS Solution Booklet - Learn about Kubernetes benefits, challenges, and enhancements made possible by a managed platform. Get the most out of Azure Kubernetes Service (AKS) with top scenarios, Azure capabilities, and tools

https://azure.microsoft.com/en-us/resources/kubernetes-on-azure-solution-booklet/

AKS + K8s + Containers Best Practices



Books…

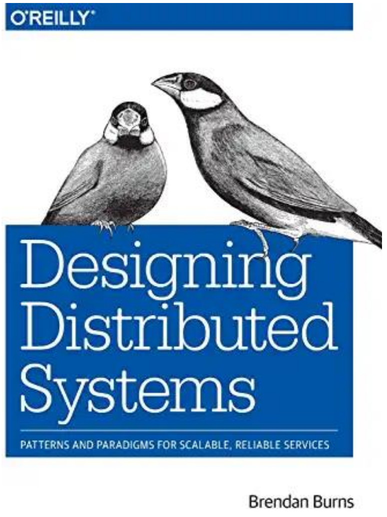


O'REILLY

Managing Kubernetes

OPERATING KUBERNETES CLUSTERS IN THE REAL WORLD

Brendan Burns & Craig Tracey

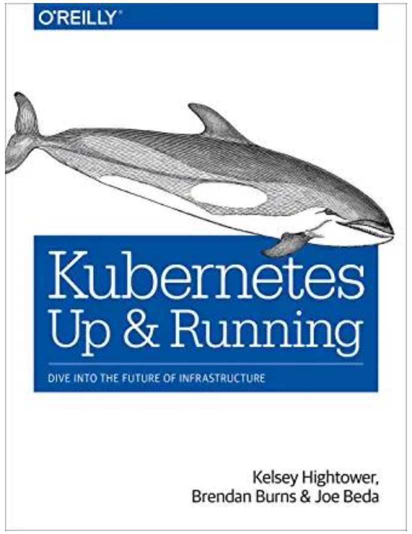


O'REILLY

Designing Distributed Systems

PATTERNS AND PARADIGMS FOR SCALABLE, RELIABLE SERVICES

Brendan Burns

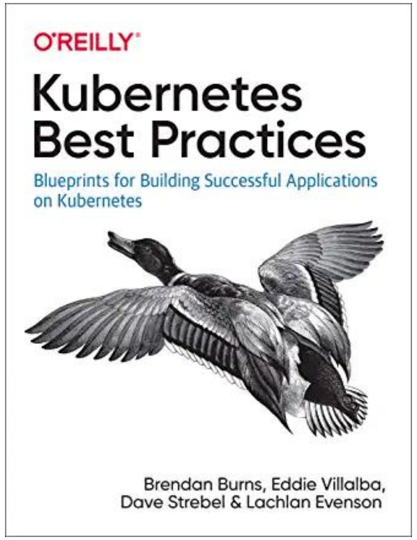


O'REILLY

Kubernetes Up & Running

DIVE INTO THE FUTURE OF INFRASTRUCTURE

Kelsey Hightower, Brendan Burns & Joe Beda



O'REILLY

Kubernetes Best Practices

Blueprints for Building Successful Applications on Kubernetes

Brendan Burns, Eddie Villalba, Dave Strebel & Lachlan Evenson

Container Best Practises

Remember that containers are designed to be ephemeral	Avoid including unnecessary packages within your container image
Use .dockerignore file • Reduce build context size • node_modules, npm-debug.log	Use multi-stage builds • Compile code and then package
Start with an appropriate image • Openjdk vs ubuntu image	Tag container images extensively • V1, v2, v3 etc..

Dockerfile Best Practises https://blog.docker.com/2019/07/intro-guide-to-dockerfile-best-practices

Azure Security Center

- Continuous discovery of managed AKS instances
- Actionable reccomendations on security best practices for AKS
- Host and Cluster based threat detection analysis

Azure ARC

- Access unique Azure security capabilities such as Azure Threat Detection
- Centrally manage access and security policies for resources with Role Based Access Control
- Enforce compliance and simplify audit reporting