Microsoft

Operational best practices for Azure Kubernetes Service

(邦題: Azure Kubernetes Service (AKS) 管理の ベスト プラクティス)

Saurya Das

Senior Program Manager
Azure Kubernetes Service
Microsoft Corporation.

CI32

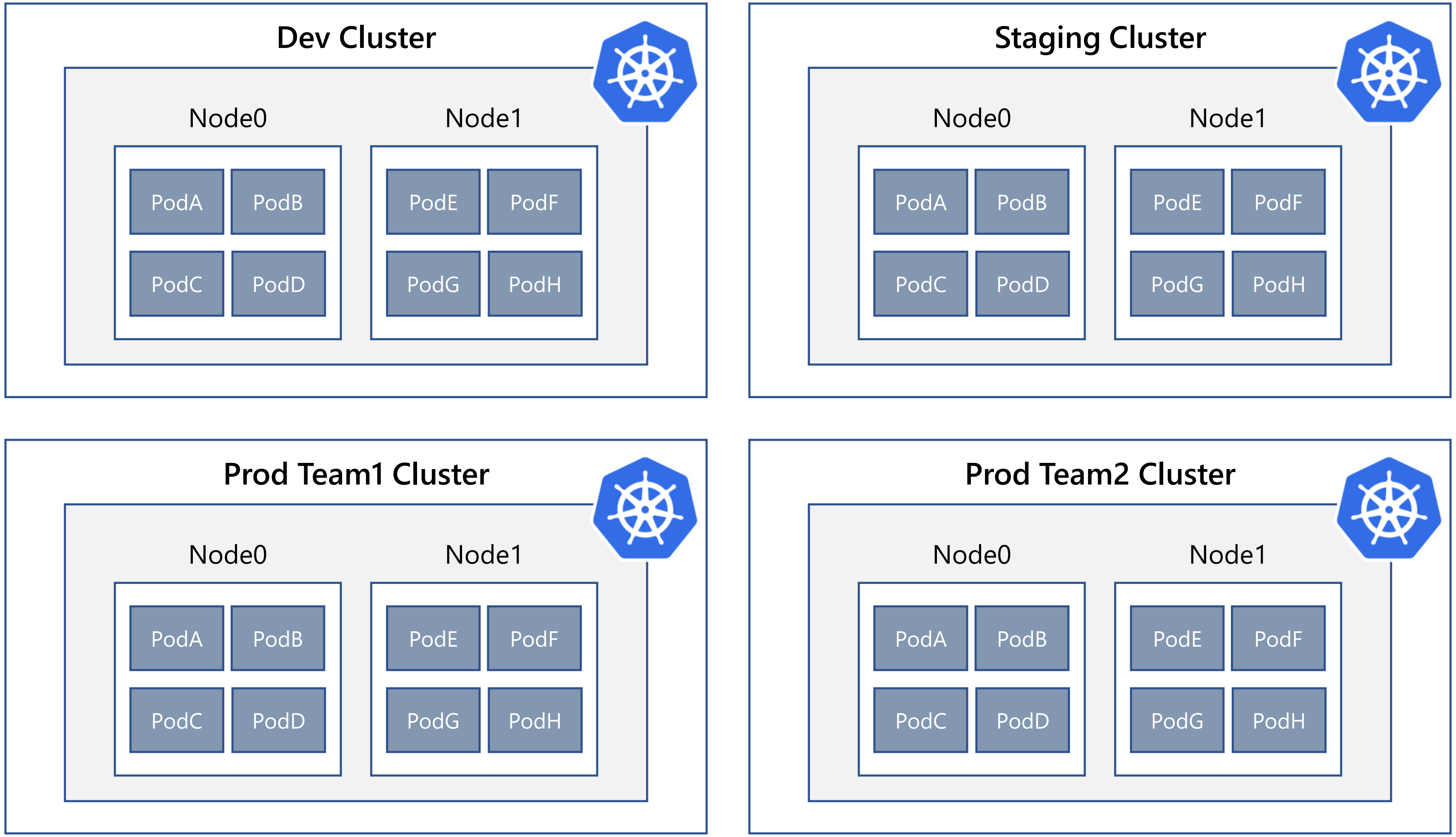
本情報の内容 (添付文書、リンク先などを含む) は、Microsoft Tech Summit 2018 開催日 (2018年11月5~7日) 時点のものであり、予告なく変更される場合があります

Agenda

- Cluster Isolation and Resource Management
- Networking
- Securing your Environment
- Scaling your Applications and Cluster
- Logging and Monitoring

Cluster Isolation and Resource Management

Cluster Isolation Patterns: Physical Isolation



Dev Cluster

Node0: PodA, PodB, PodC, PodD

Node1: PodE, PodF, PodG, PodH

Staging Cluster

Node0: PodA, PodB, PodC, PodD

Node1: PodE, PodF, PodG, PodH

Prod Team1 Cluster

Node0: PodA, PodB, PodC, PodD

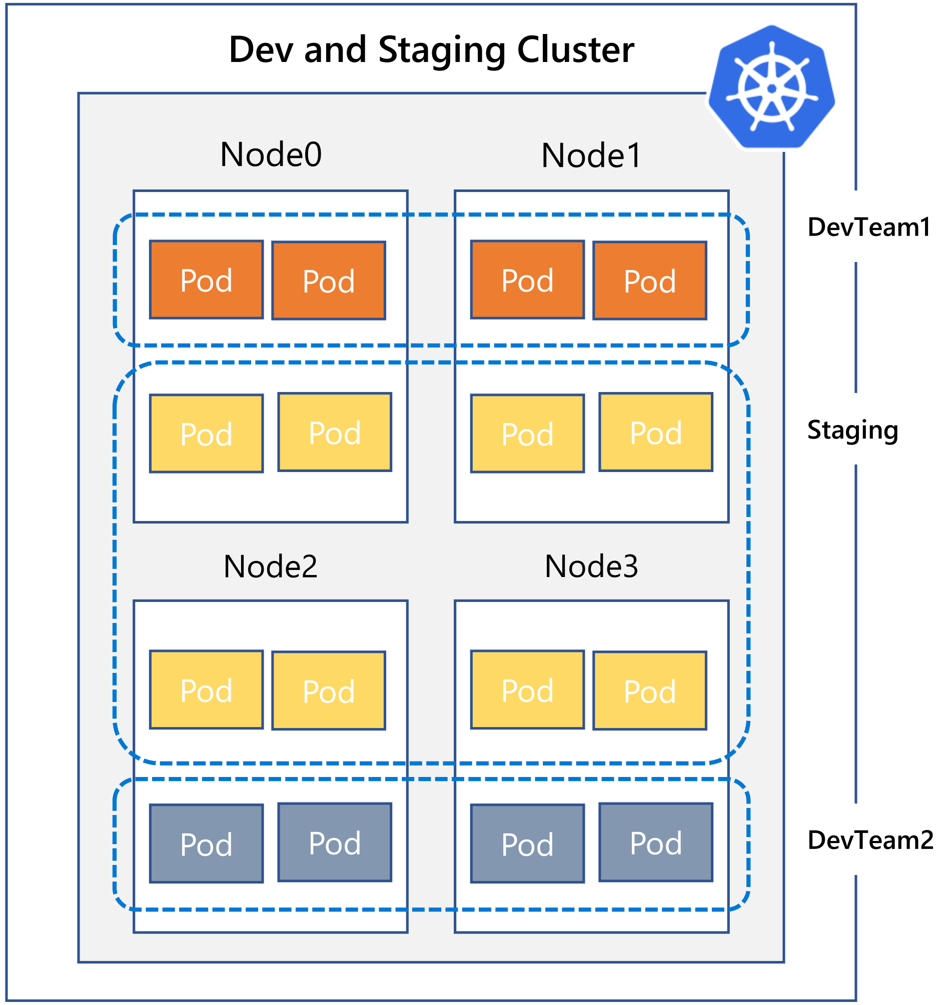
Node1: PodE, PodF, PodG, PodH

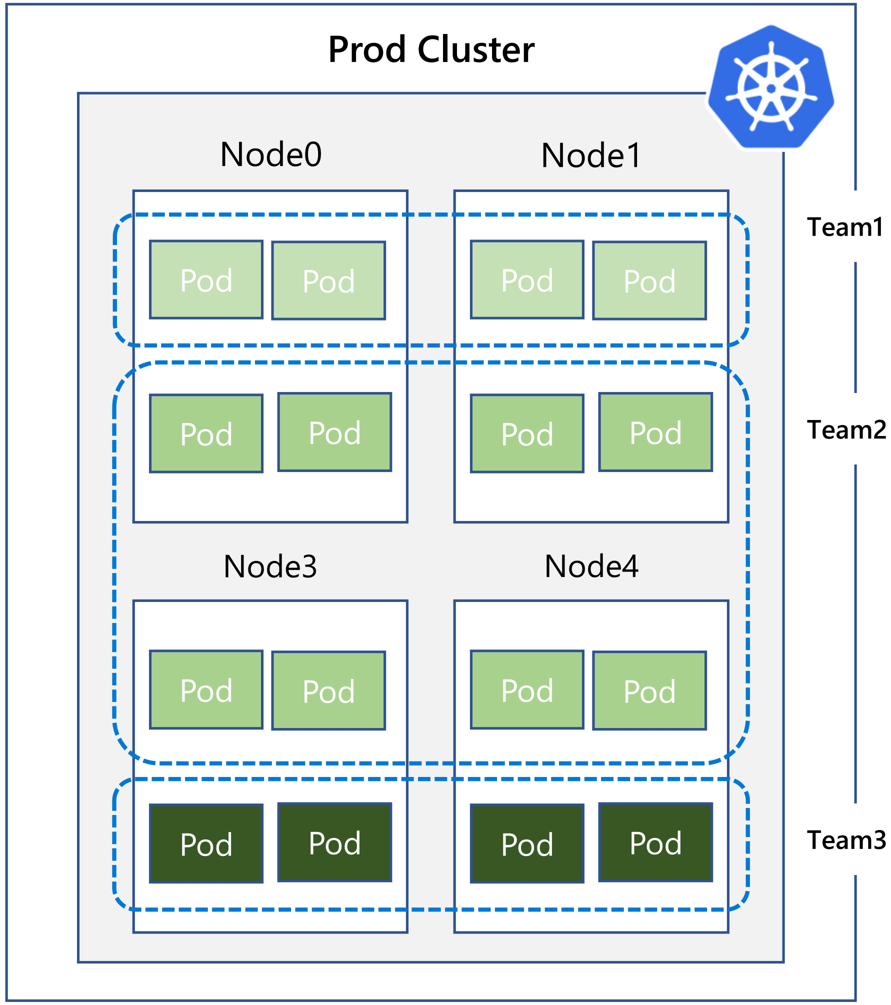
Prod Team2 Cluster

Node0: PodA, PodB, PodC, PodD

Node1: PodE, PodF, PodG, PodH

Cluster Isolation Patterns: Logical Isolation





Kubernetes Namespaces

- Namespaces Object is the logical Isolation boundary
- Kubernetes has features to help us safely isolate tenants
    - Scheduling: Resource Quota
    - Network Isolation using Network Policies
    - Authentication and Authorization: RBAC and Pod Security Policy
- Note: Container Level isolation still need to be done to achieve hard Isolation

Kubernetes Resource Quotas

- Constraints that limit aggregate resource consumption per namespace
- You can limit Compute Resources (CPU,Memory, Strage,…) and/or limit the number of Objects (Pods, Services, etc..) and
- When enabled, users must specify requests or limits, otherwise the quota system will fail the request.
- Kubernetes will not overcommit

Create a namespace:

```
$ kubectl create namespace ignite
```

Apply a resource quota to the namespace:

admin/resource/ignite.yaml

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: mem-cpu-demo
spec:
hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
```

Physical vs. Logical Isolation

	Physical	Logical
Pod Density	Low to Medium	Medium to High
Cost	$$	$
Kubernetes Experience	Low to Medium	Medium to High
Security	High (Surface is small)	High*
Blast Radius of Changes	Small	Big
Management and Operations	Owner Team	Single or Cross Functional Team

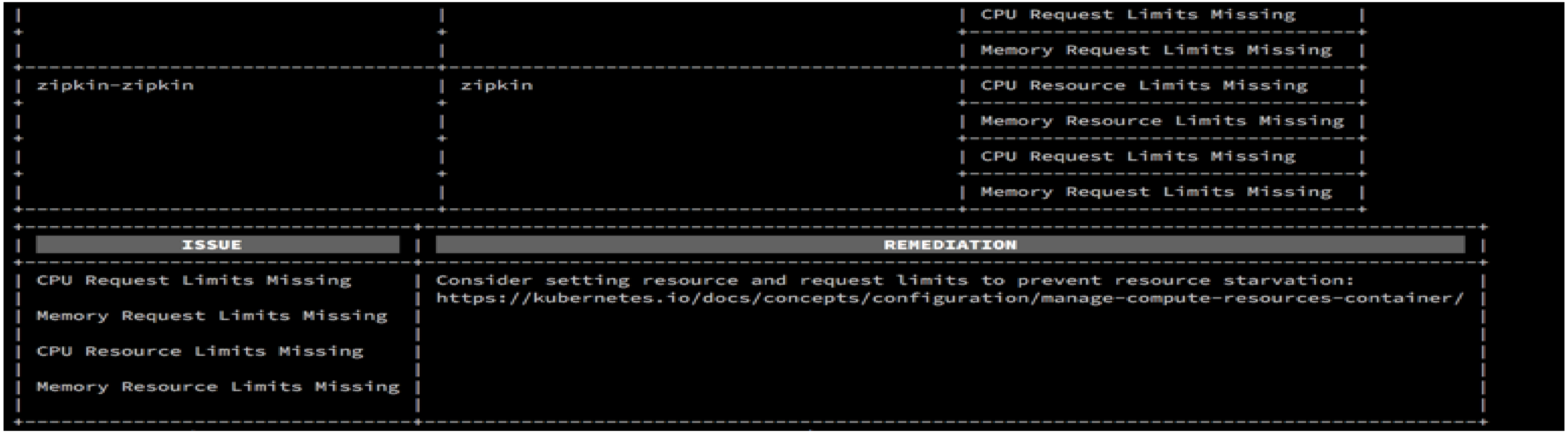
*Logical Isolation via Namespaces can achieve hard isolation assuming the cluster admin has applied all the required security controls

Demo

Create namespace
Apply a resource quota
Deploy a simple container within limits
Deploy another container beyond limits that fails

Kube-advisor

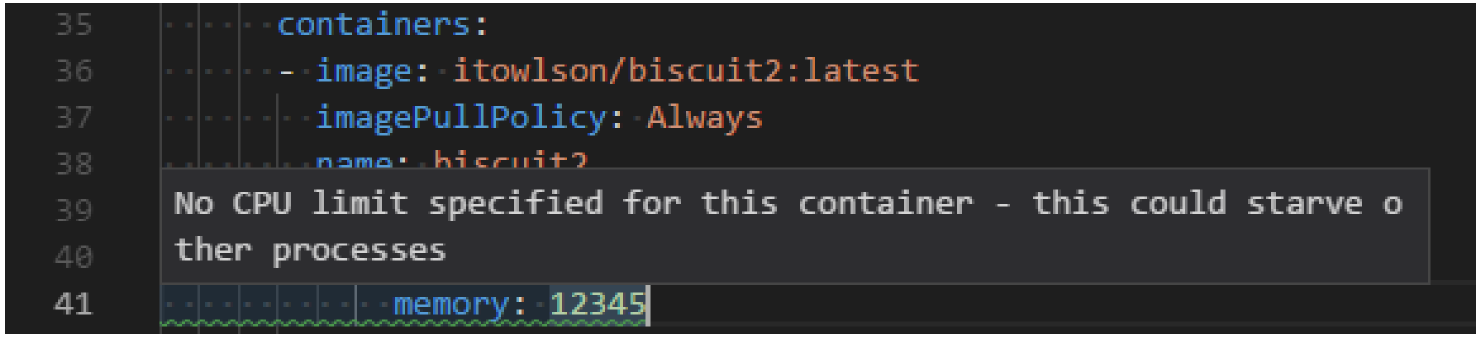
- Diagnostic tool for Kubernetes clusters. At the moment, it returns pods that are missing resource and request limits.
- More info can be found at https://github.com/Azure/kube-advisor



```
|                                  |                                         | CPU Request Limits Missing     |
+                                  +                                         +--------------------------------+
|                                  |                                         | Memory Request Limits Missing  |
+----------------------------------+-----------------------------------------+--------------------------------+
| zipkin-zipkin                    | zipkin                                  | CPU Resource Limits Missing    |
+                                  +                                         +--------------------------------+
|                                  |                                         | Memory Resource Limits Missing |
+                                  +                                         +--------------------------------+
|                                  |                                         | CPU Request Limits Missing     |
+                                  +                                         +--------------------------------+
|                                  |                                         | Memory Request Limits Missing  |
+----------------------------------+-----------------------------------------+--------------------------------+
+--------------------------------+--------------------------------------------------------------------------------------+
|             ISSUE              |                                     REMEDIATION                                      |
+--------------------------------+--------------------------------------------------------------------------------------+
| CPU Request Limits Missing     | Consider setting resource and request limits to prevent resource starvation:         |
|                                | https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/ |
| Memory Request Limits Missing  |                                                                                      |
|                                |                                                                                      |
| CPU Resource Limits Missing    |                                                                                      |
|                                |                                                                                      |
| Memory Resource Limits Missing |                                                                                      |
|                                |                                                                                      |
+--------------------------------+--------------------------------------------------------------------------------------+
```

VS Code extension for warnings

- Kubernetes VS Code extension adding warnings for resource request/limits



```
      containers:
      - image: itowlson/biscuit2:latest
        imagePullPolicy: Always
        name: biscuit2
```

```
No CPU limit specified for this container - this could starve o
ther processes
```

```
              memory: 12345
```

Cluster Isolation - Summary

- Think of the sensitivity of the workload, cost, organization culture, operations model, and blast radius, when trying to choose which isolation pattern to use, a mixture is fine too.
- Always use Namespaces even in physical isolation, never use the Default Namespace for production workloads
- Apply Resource Quotas

Networking

AKS Basic Networking

- Done using Kubenet network plugin and has the following features
  - Nodes and Pods are placed on different IP subnets
  - User Defined Routing and IP Forwarding is for connectivity between Pods across Nodes.

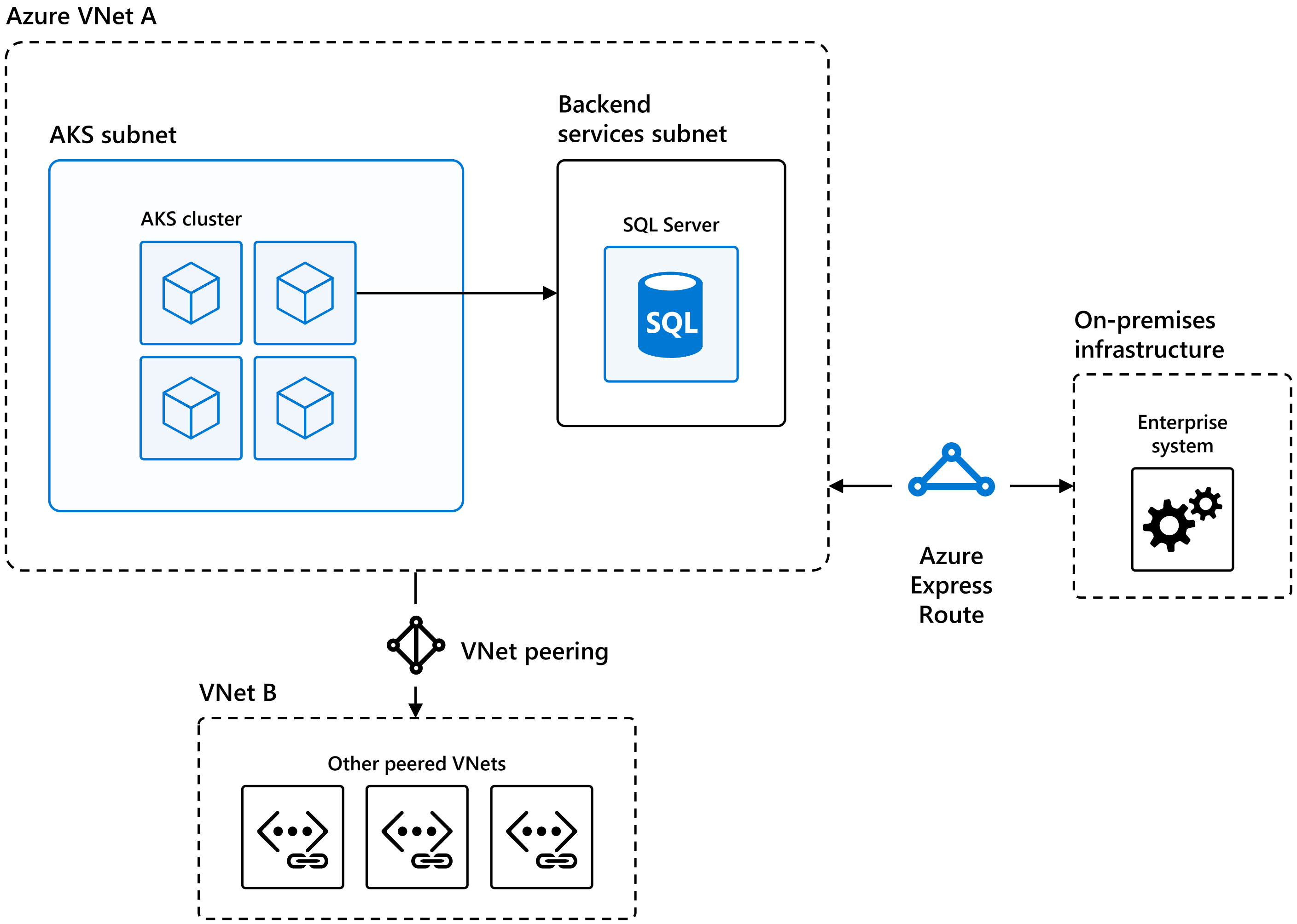
- Drawbacks
  - 2 different IP CIDRs to manage
  - Performance impact
  - Peering or On-Premise connectivity is hard to achieve

AKS Advanced Networking

- Done using 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

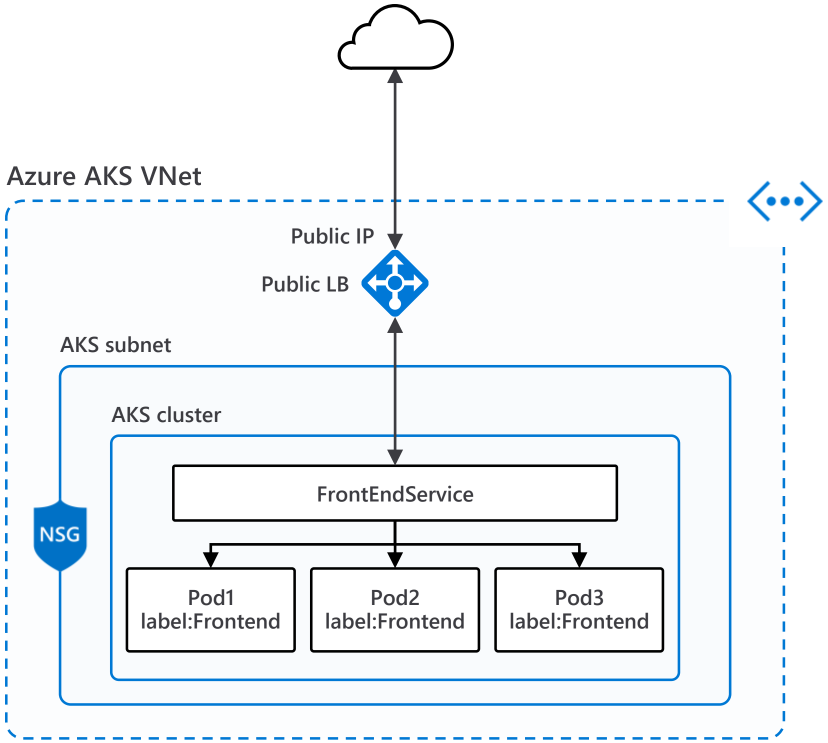
AKS with Advanced Networking



Public Service

- Service Type LoadBalancer
- Basic Layer4 Load Balancing (TCP/UDP)
- Each service as assigned an IP on the ALB

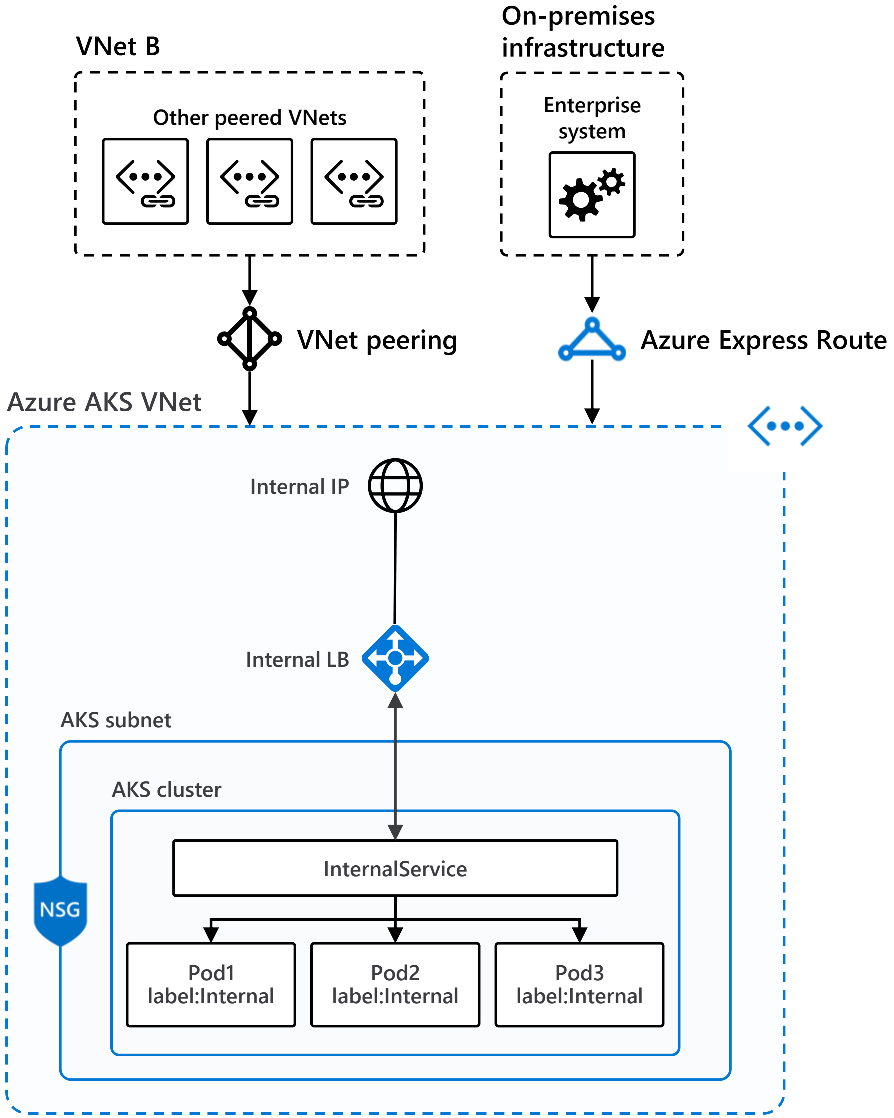
```yaml
apiVersion: v1
kind: Service
metadata:
  name: frontendservice
spec:
  loadBalancerIP: X.X.X.X
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    app: frontend
```



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
```



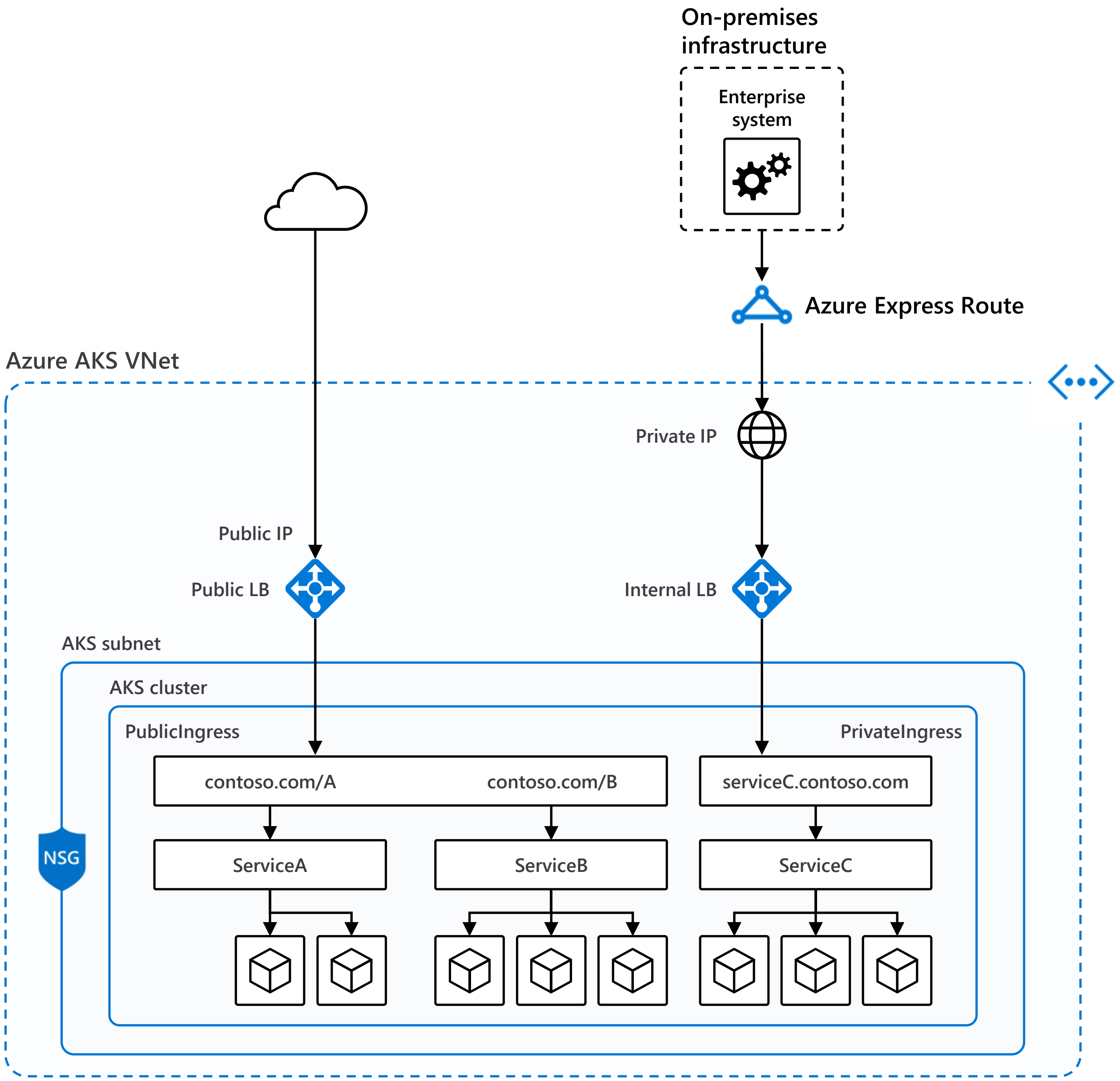
Ingress and Ingress Controllers

- Ingress is a Kubernetes API that manages external access to the services in the cluster
  - Supports HTTP and HTTPs
  - Path and Subdomain based routing
  - SSL Termination
  - Save on public Ips

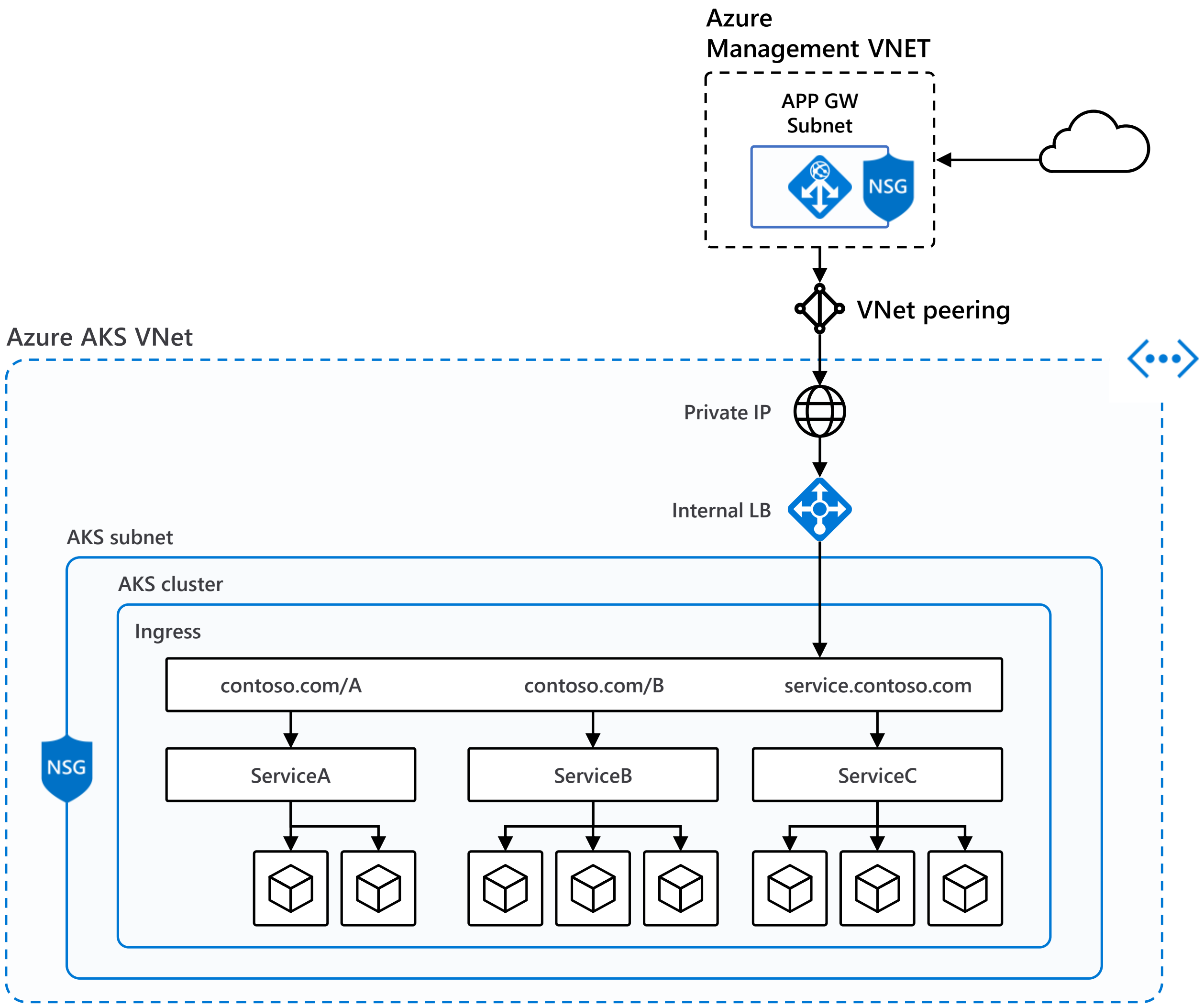
- Ingress controller is a daemon, deployed as a Kubernetes Pod, that watches the Ingress Endpoint for updates. Its job is to satisfy requests for ingresses. Most popular one being Nginx.

Ingress

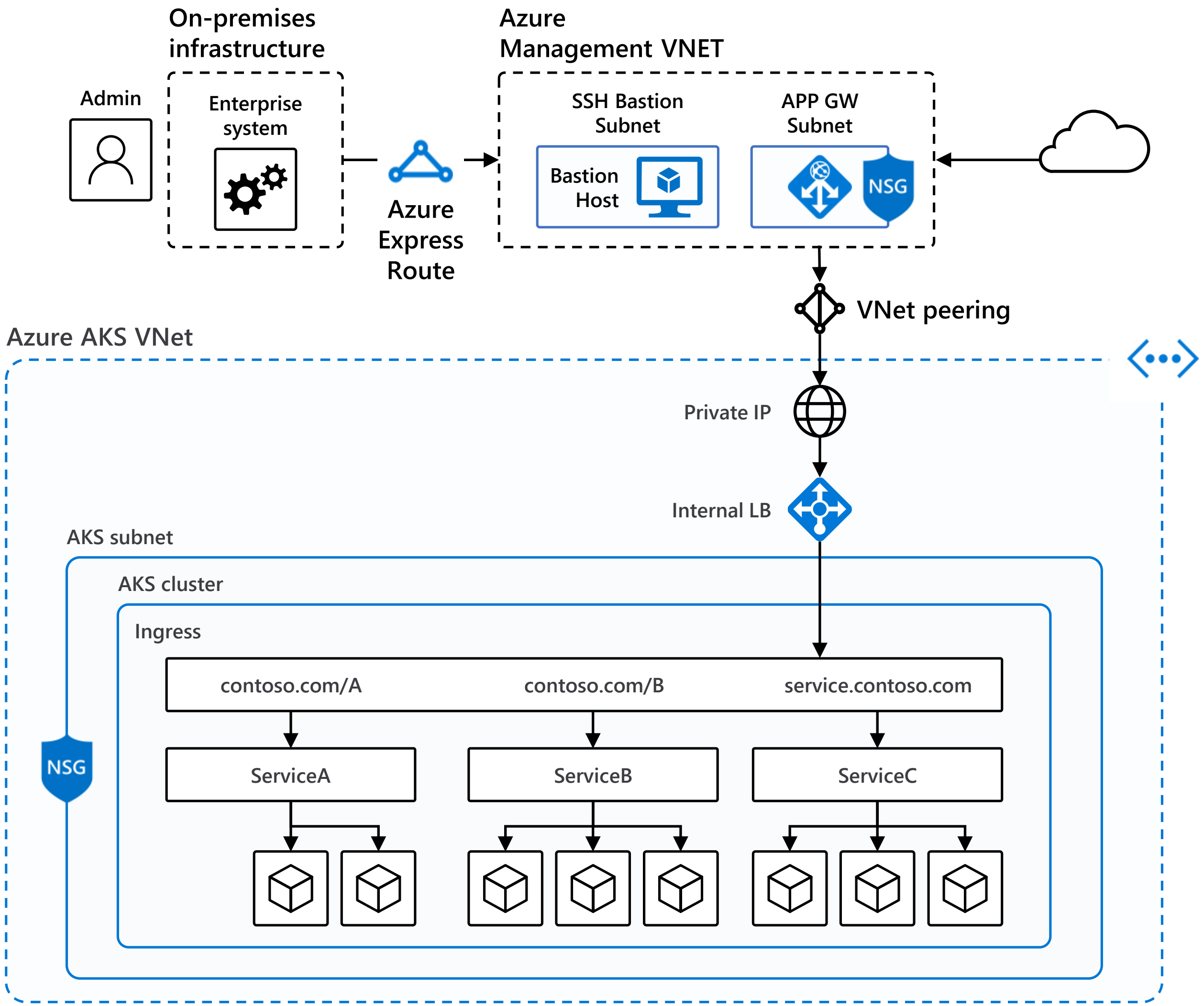
```
kind: Ingress
metadata:
 name: contoso-ingress
   annotations: kubernetes.io/ingress.class:
”PublicIngress"
spec:
 tls:
 - hosts:
   - contoso.com
   secretName: contoso-secret
 rules:
   - host: contoso.com
     http:
      paths:
      - path: /a
        backend:
         serviceName: servicea
         servicePort: 80
      - path: /b
        backend:
         serviceName: serviceb
         servicePort: 80
```



Securing Kubernetes Services with WAF



Cluster Management Through Bastion Host



Demo

Ingress
Café Application with 2 endpoints Coffee and Tea
SSL termination
Loadbalancing

Summary

- Use AKS Advanced networking for seamless integration with your VNET
- Use Ingress and Ingress controllers for HTTP and HTTPs services
- Use Azure Application Gateway or any other alternative from the Azure Market place to secure your services using a WAF
- Use Bastion Hosts to access your nodes when needed

Securing your environment

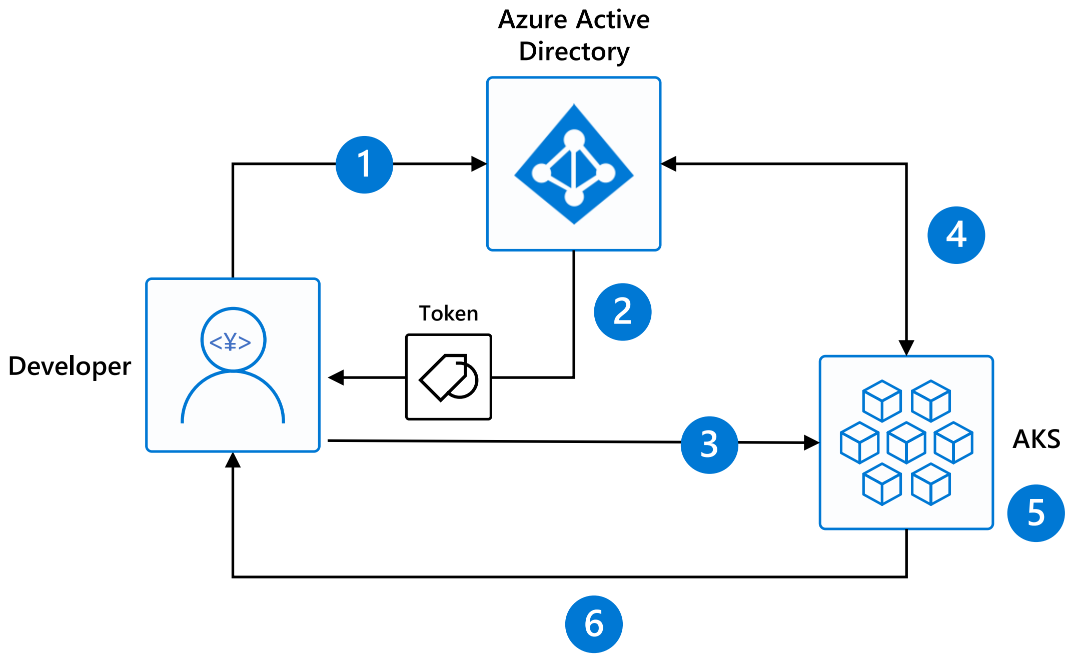
Cluster Level Security

Cluster Level Security

- Securing endpoints for API server and cluster nodes
    - Ensuring authentication and authorization (AAD + RBAC)
    - Setting up & keeping least privileged access for common tasks

Cluster Level - Identity and Access Management through AAD and RBAC

1. Kubernetes Developer authenticates with AAD
2. The AAD token issuance endpoint issues the access token
3. Developer performs action w/ AAD token.
   Eg. *kubectl create pod*
4. Kubernetes validates token with AAD and fetches the Developer's AAD Groups
   Eg. Dev Team A, App Group B
5. Kubernetes RBAC and cluster policies are applied
6. Request is successful or not based on the previous validation



Provisioning AD-enabled AKS

```
$ az aks create --resource-group myAKSCluster --name myAKSCluster --generate-ssh-keys ¥
  --aad-server-app-id <Azure AD Server App ID> ¥
  --aad-server-app-secret <Azure AD Server App Secret> ¥
  --aad-client-app-id <Azure AD Client App ID> ¥
  --aad-tenant-id <Azure AD Tenant>
```

```
$ az aks get-credentials --resource-group myAKSCluster –name  myAKSCluster --admin
Merged "myCluster" as current context ..
```

```
$ kubectl get nodes

NAME                       STATUS    ROLES     AGE       VERSION
aks-nodepool1-42032720-0   Ready     agent     1h        v1.9.6
aks-nodepool1-42032720-1   Ready     agent     1h        v1.9.6
aks-nodepool1-42032720-2   Ready     agent     1h        v1.9.6
```

Provisioning AD-enabled AKS

Setting up a Cluster Role

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  labels:
    kubernetes.io/cluster-service: "true"
  name: cluster-admin
rules:
- apiGroups:
  - extensions
  - apps
  resources:
  - deployments
  verbs:
  - get
  - list
  - watch
  - update
  - patch
- apiGroups:
  - ""
  resources:
  - events
  - namespaces
  - nodes
  - pods
  verbs:
  - get
  - list
  - watch
```

Bind the Cluster Role to a user

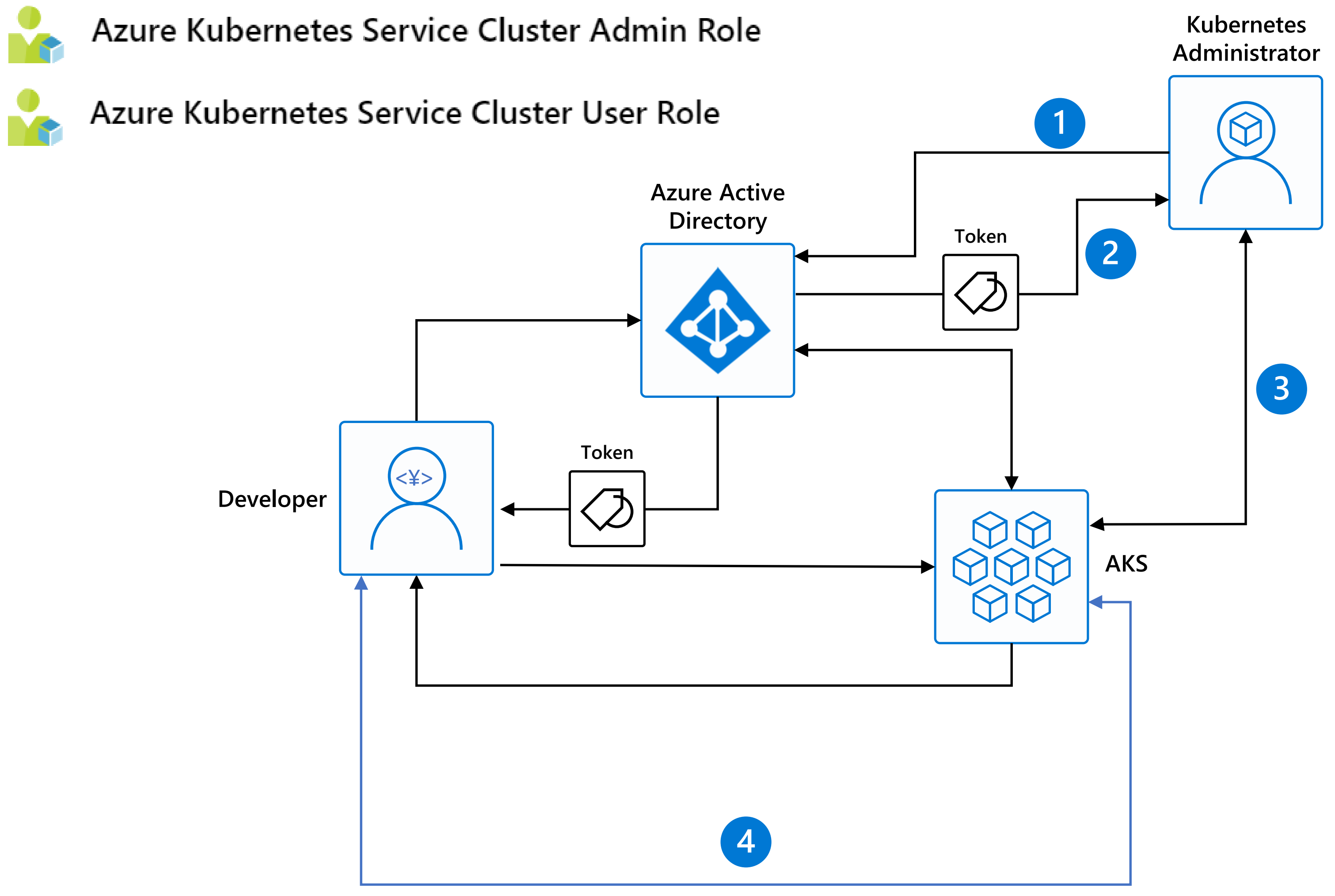
```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
name: contoso-cluster-admins
roleRef:
apiGroup: rbac.authorization.k8s.io
kind: ClusterRole
name: cluster-admin
subjects:
- apiGroup: rbac.authorization.k8s.io
kind: User
name: "user@contoso.com"
```

Bind the Cluster Role to a group

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
name: contoso-cluster-admins
roleRef:
apiGroup: rbac.authorization.k8s.io
kind: ClusterRole
name: cluster-admin
subjects:
- apiGroup: rbac.authorization.k8s.io
kind: Group
name: "894656e1-39f8-4bfe-b16a-510f61af6f41"
```

Azure Level - Identity and Access Management through AAD and RBAC

1. Kubernetes Administrator authenticates with AAD
2. The AAD token issuance endpoint issues the access token
3. Administrator fetches the admin kubeconfig and configures RBAC roles and bindings
4. Kubernetes Developer fetches the user kubeconfig



Azure Kubernetes Service Cluster Admin Role
Azure Kubernetes Service Cluster User Role
Kubernetes Administrator
1
Azure Active Directory
Token
2
3
Developer
Token
AKS
4

Provisioning AD-enabled AKS

```
$ az aks get-credentials --resource-group myAKSCluster --name myAKSCluster

$ kubectl get nodes

To sign in, use a web browser to open the page https://microsoft.com/devicelogin and
enter the code BUJHWDGNL to authenticate.

NAME                        STATUS    ROLES    AGE      VERSION
aks-nodepool1-42032720-0    Ready     agent    1h       v1.9.6
aks-nodepool1-42032720-1    Ready     agent    1h       v1.9.6
aks-nodepool1-42032720-2    Ready     agent    1h       v1.9.6
```

Or

```
Error from server (Forbidden): nodes is forbidden: User baduser@contoso.com cannot
list nodes at the cluster scope
```

Cluster Level Security

- Securing endpoints for API server and cluster nodes
  - Ensuring authentication and authorization (AAD + RBAC)
  - Setting up & keeping least privileged access for common tasks
  - Admission Controllers
    - DenyEscalatingExec
    - ValidatingAdmissionWebhooks
    - MutatingAdmissionWebhooks
    - ServiceAccount
    - Coming soon:
      - NodeRestriction
      - PodSecurityPolicy

Cluster Level – Nodes, Upgrade and Patches

- Regular maintenance, security and cleanup tasks
  - Maintain, update and upgrade hosts and kubernetes
  - Monthly ideal, 3 months minimum
  - Security patches
    - AKS automatically applies security patches to the nodes on a nightly schedule
    - You’re responsible to reboot as required
    - Kured DaemonSet https://github.com/weaveworks/kured

Upgrade to version 1.10.6

```
$ az aks upgrade --name myAKSCluster ¥
--resource-group myResourceGroup ¥
--kubernetes-version 1.10.6
```

- **SSH Access**
  - DenyEscalatingExec

- **Running benchmarks and tests to validate cluster setup**
  - Kube-bench
  - Aqua Hunter
  - Others

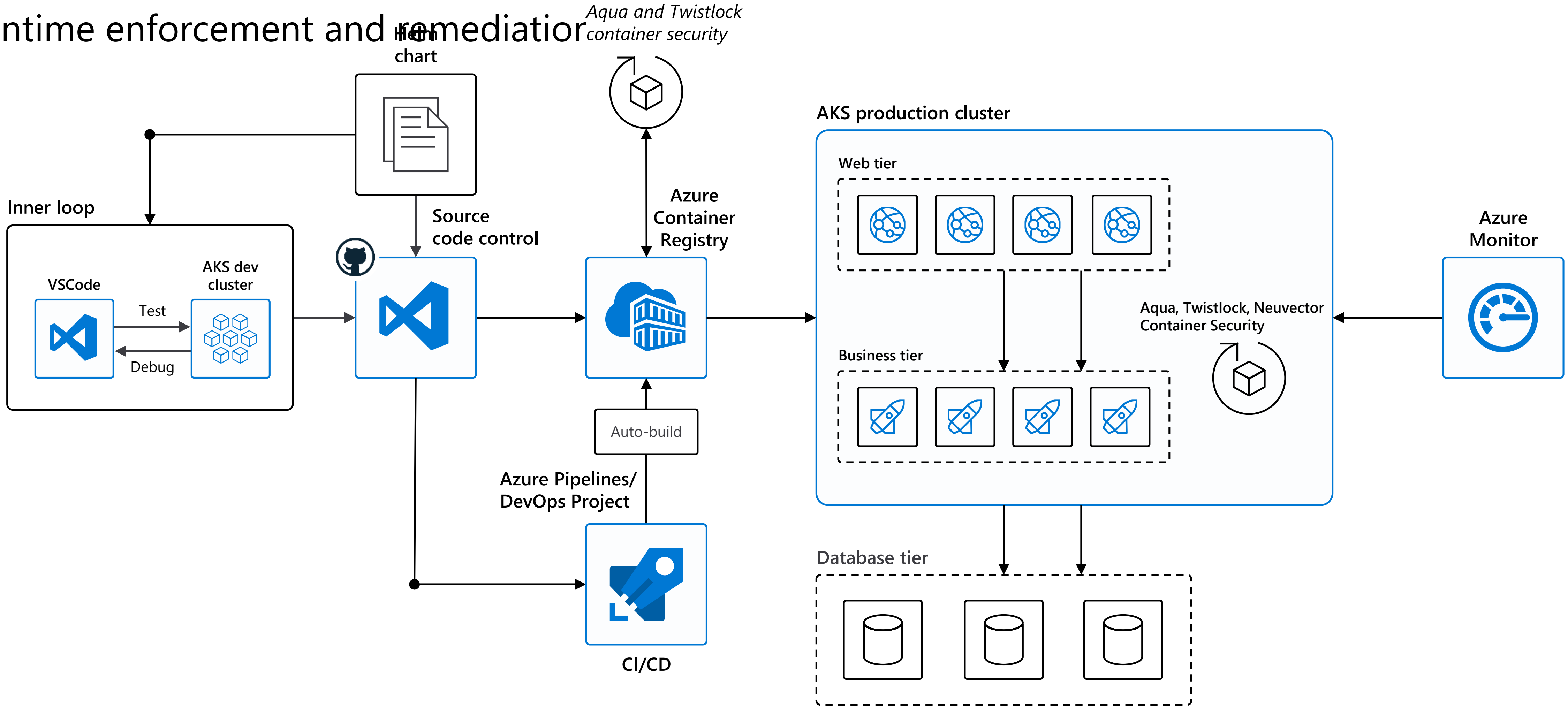
Container Level Security and Isolation

Container Level – The images

- Trusted Registry
- Regularly apply security updates to the container images

Container Level – Images and Runtime

- Scan your images, scan your containers
- Runtime enforcement and remediation



Container Level – The access

- Avoid access to HOST IPC namespace - only if absolutely necessary
- Avoid access to Host PID namespace - only if absolutely necessary
- Avoid root / privileged access
    - Consider Linux Capabilities

Container Level – App Armor

Securing a Pod with a deny-write.profile

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-apparmor
  annotations:
container.apparmor.security.beta.kubernetes.io/
hello: localhost/k8s-apparmor-example-deny-
write
spec:
  containers:
  - name: hello
    image: busybox
    command: [ "sh", "-c", "echo 'Hello
AppArmor!' && sleep 1h" ]
```

```
$ kubectl exec hello-apparmor touch /tmp/test
touch: /tmp/test: Permission denied
error: error executing remote command: command terminated with non-zero exit
code: Error executing in Docker Container: 1
```

deny-write.profile

```
#include <tunables/global>

profile k8s-apparmor-example-deny-
write flags=(attach_disconnected) {
  #include <abstractions/base>

  file,

  # Deny all file writes.
  deny /** w,
}
```

Container Level - Seccomp

Securing a Pod with a prevent-chmod profile

```
apiVersion: v1
kind: Pod
metadata:
  name: chmod-prevented
  annotations:
    seccomp.security.alpha.kubernetes.io/pod: 
localhost/prevent-chmod

spec:
  containers:
  - name: chmod
    image: busybox
    command:
      - "chmod“
    args:
      - “777“
      - /etc/hostname
  restartPolicy: Never
```

Seccomp Profile
/var/lib/kubelet/seccomp/prevent-chmod

```
{
  "defaultAction": "SCMP_ACT_ALLOW",
  "syscalls": [
    {
      "name": "chmod",
      "action": "SCMP_ACT_ERRNO"
    }
  ]
}
```

Container Level

```
$ kubectl create -f seccomp-pod.yaml
pod "chmod-prevented" created

$ kubectl get pods
NAME                      READY        STATUS       RESTARTS     AGE
chmod-prevented           0/1          Error        0            8s
```

Pod Level Security

Pod Level – Pod Security Context

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  securityContext:
    runAsUser: 1000
    fsGroup: 2000
  volumes:
  - name: sec-ctx-vol
    emptyDir: {}
  containers:
  - name: sec-ctx-demo
    image: ignite.azurecr.io/nginx-demo
    volumeMounts:
    - name: sec-ctx-vol
      mountPath: /data/demo
    securityContext:
      runAsUser: 2000
      allowPrivilegeEscalation: false
      capabilities:
        add: ["NET_ADMIN", "SYS_TIME"]
      seLinuxOptions:
        level: "s0:c123,c456"
```

Pod Level – Pod Security Context

```
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  securityContext:
    runAsUser: 1000
    fsGroup: 2000
  volumes:
  - name: sec-ctx-vol
    emptyDir: {}
  containers:
  - name: sec-ctx-demo
    image: ignite.azurecr.io/nginx-demo
    volumeMounts:
    - name: sec-ctx-vol
      mountPath: /data/demo
    securityContext:
      runAsUser: 2000
      allowPrivilegeEscalation: false
      capabilities:
        add: ["NET_ADMIN", "SYS_TIME"]
      seLinuxOptions:
        level: "s0:c123,c456"
```

Pod Level – Pod Security Policies

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: restricted
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: 'docker/default’
    apparmor.security.beta.kubernetes.io/allowedProfileNames: 'runtime/default’
    seccomp.security.alpha.kubernetes.io/defaultProfileName: 'docker/default’
    apparmor.security.beta.kubernetes.io/defaultProfileName: 'runtime/default’
spec:
  privileged: false
  allowPrivilegeEscalation: false # Required to prevent escalations to root.
  requiredDropCapabilities: # This is redundant with non-root + disallow privilege escalation, but we can provide it for defense in depth.
    - ALL
  volumes: # Allow core volume types.
    - 'configMap’
    - 'emptyDir’
    - 'projected’
    - 'secret’
    - 'downwardAPI’
    - 'persistentVolumeClaim’ # Assume that persistentVolumes set up by the cluster admin are safe to use.
  hostNetwork: false
  hostIPC: false
  hostPID: false
  runAsUser:
    rule: 'MustRunAsNonRoot’ # Require the container to run without root privileges.
  seLinux:
    rule: 'RunAsAny’ # This policy assumes the nodes are using AppArmor rather than SELinux.
  supplementalGroups:
    rule: 'MustRunAs’
    ranges:
      - min: 1      # Forbid adding the root group.
        max: 65535
  fsGroup:
    rule: 'MustRunAs’
    ranges:
      - min: 1     # Forbid adding the root group.
        max: 65535
  readOnlyRootFilesystem: false
```

Pod level

- Pod Security Context
- Pod Security Policies
- AlwaysPull Images

Securing Workloads

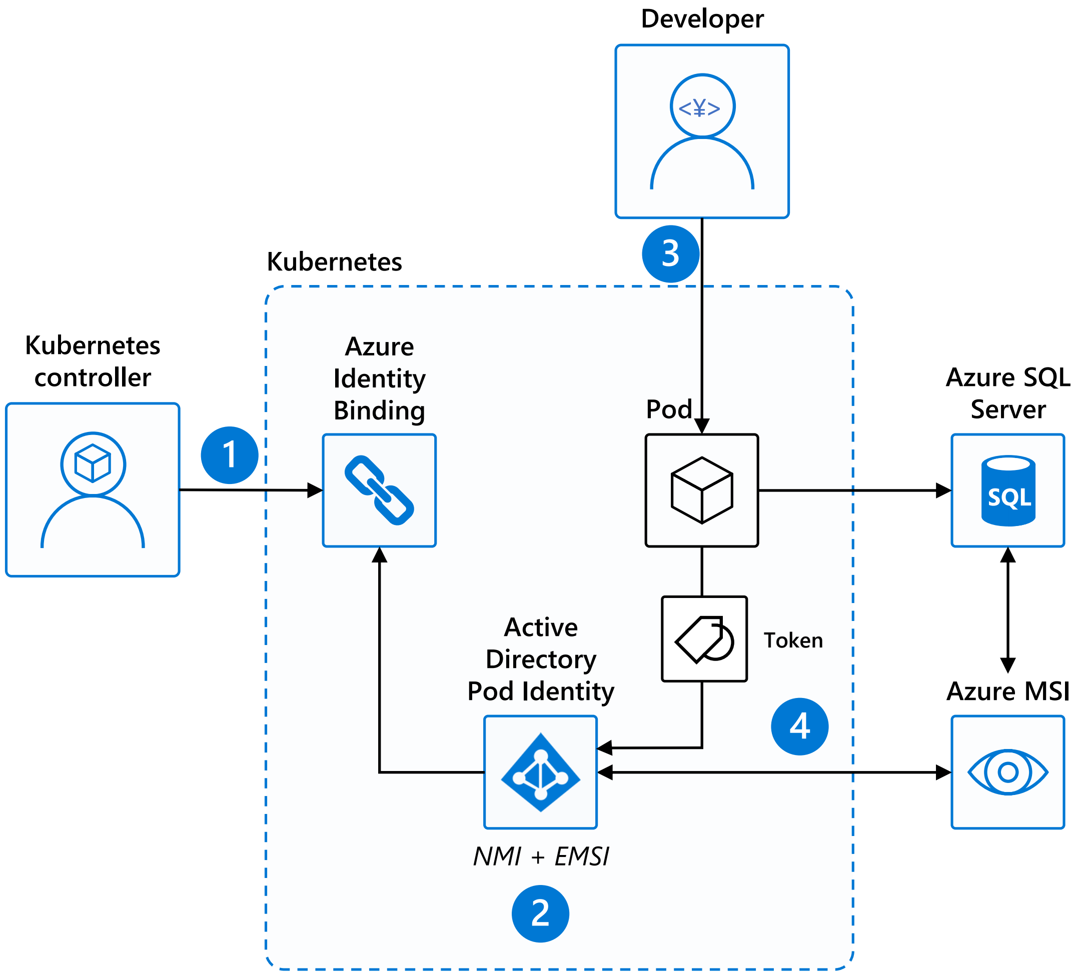
Pod Identity

1. Kubernetes operator defines an identity map for K8s service accounts

2. Node Managed Identity (NMI) watches for mapping reaction and syncs to Managed Service Identify (MSI)

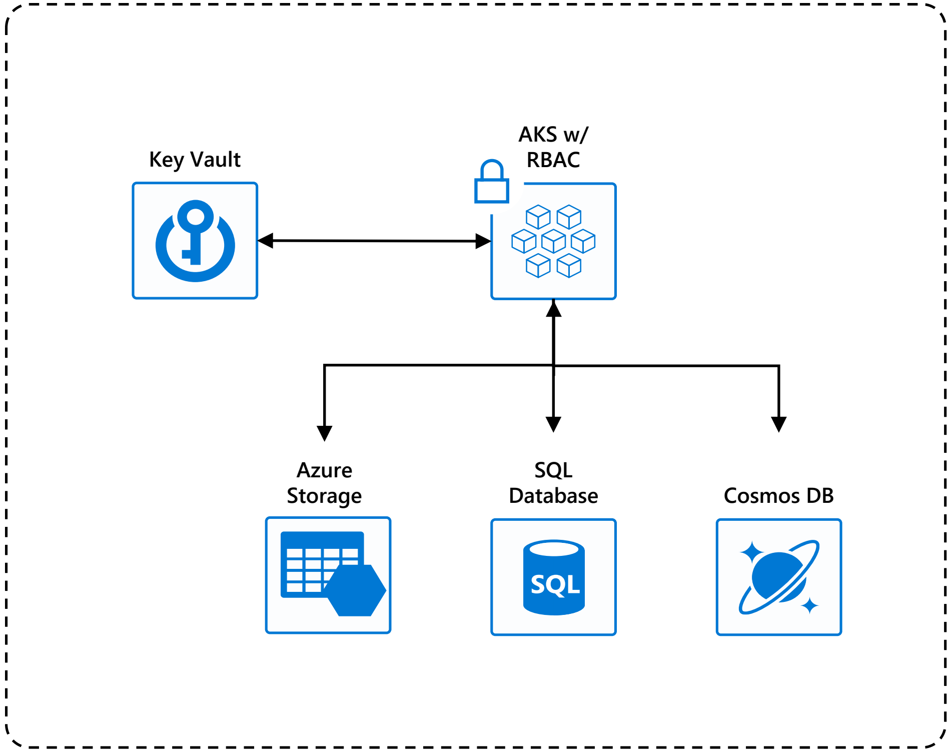
3. Developer creates a pod with a service account. Pod uses standard Azure SDK to fetch a token bound to MSI

4. Pod uses access token to consume other Azure services; services validate token



Securing workloads

- Managing secrets and privileged information
  - Azure Key Vault



Demo

Protect your secrets with Azure Key Vault

Securing workloads

- Service Endpoints
- Filter secrets from the logs
- Encrypted Service to Service Communication
  - mTLS between services
  - Service Meshes

Compliance

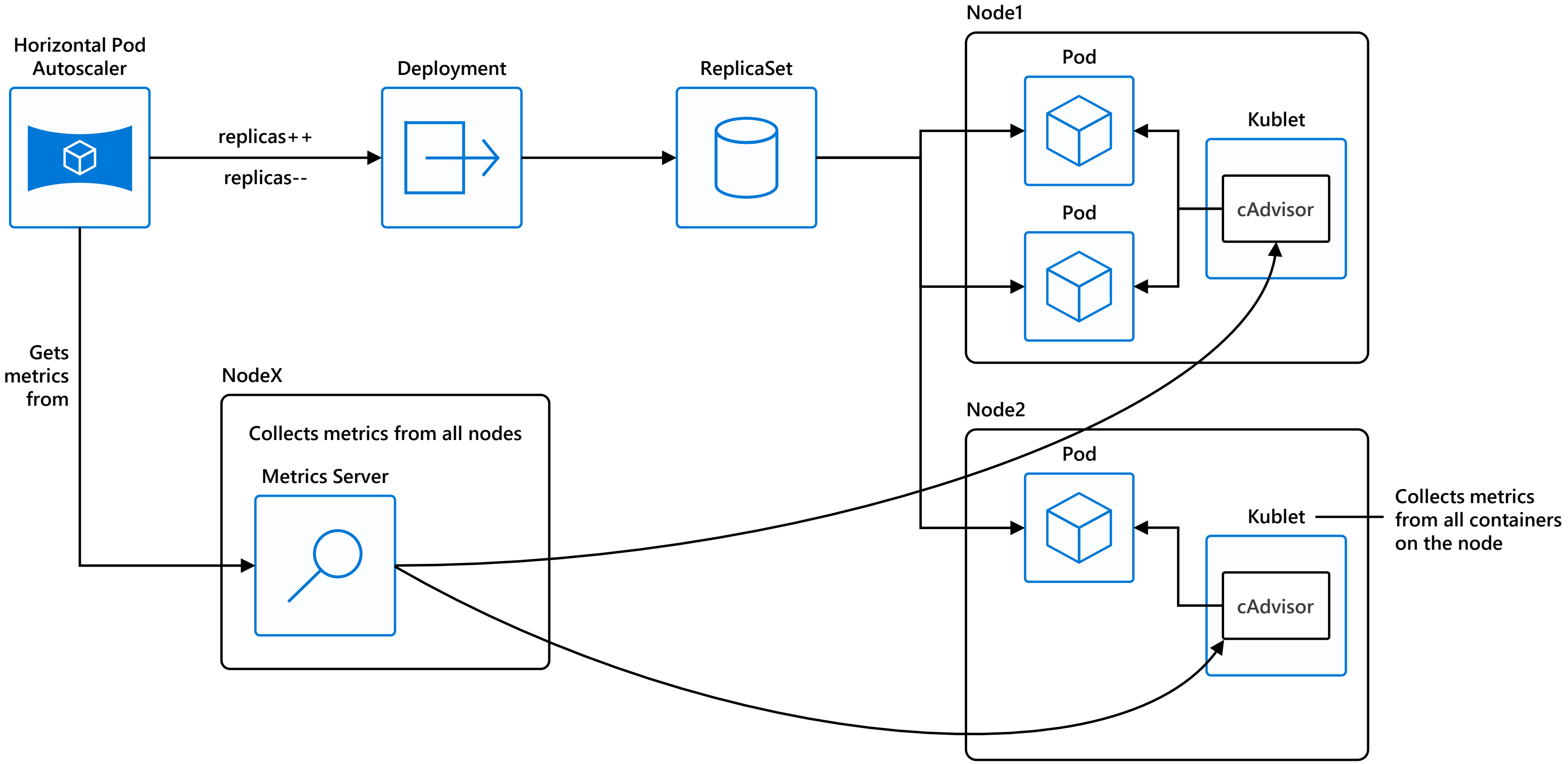
- AKS is SOC 1/2 , PCI , HIPPA and ISO certified
- All the details are listed in the Azure Trust Center

Autoscaling Applications and Clusters

Manual scaling is tedious and ineffective

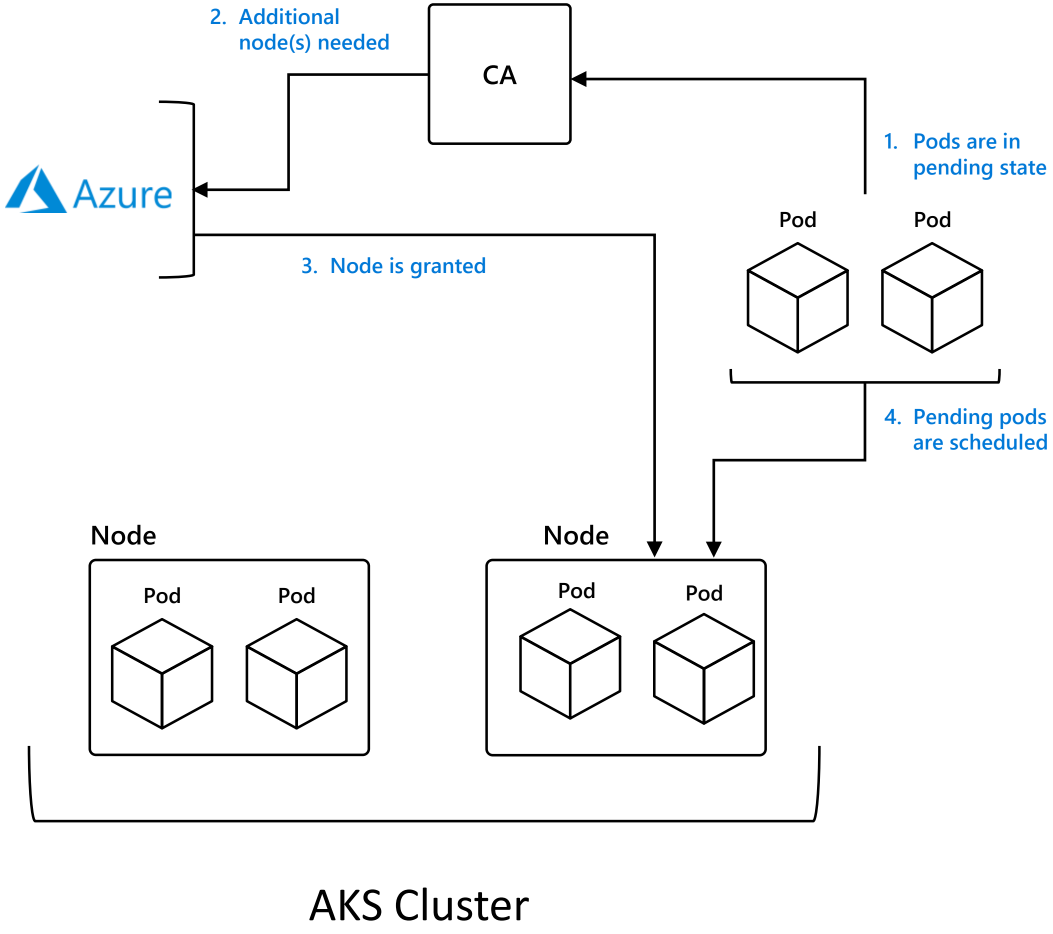
- Horizontal pod autoscaling(HPA) -> Scaling pods/containers
- Cluster Autoscaling -> Scaling infrastructure/VM’s
- AKS + ACI + VK for burst scenarios -> Scaling pods/containers

How HPA works?

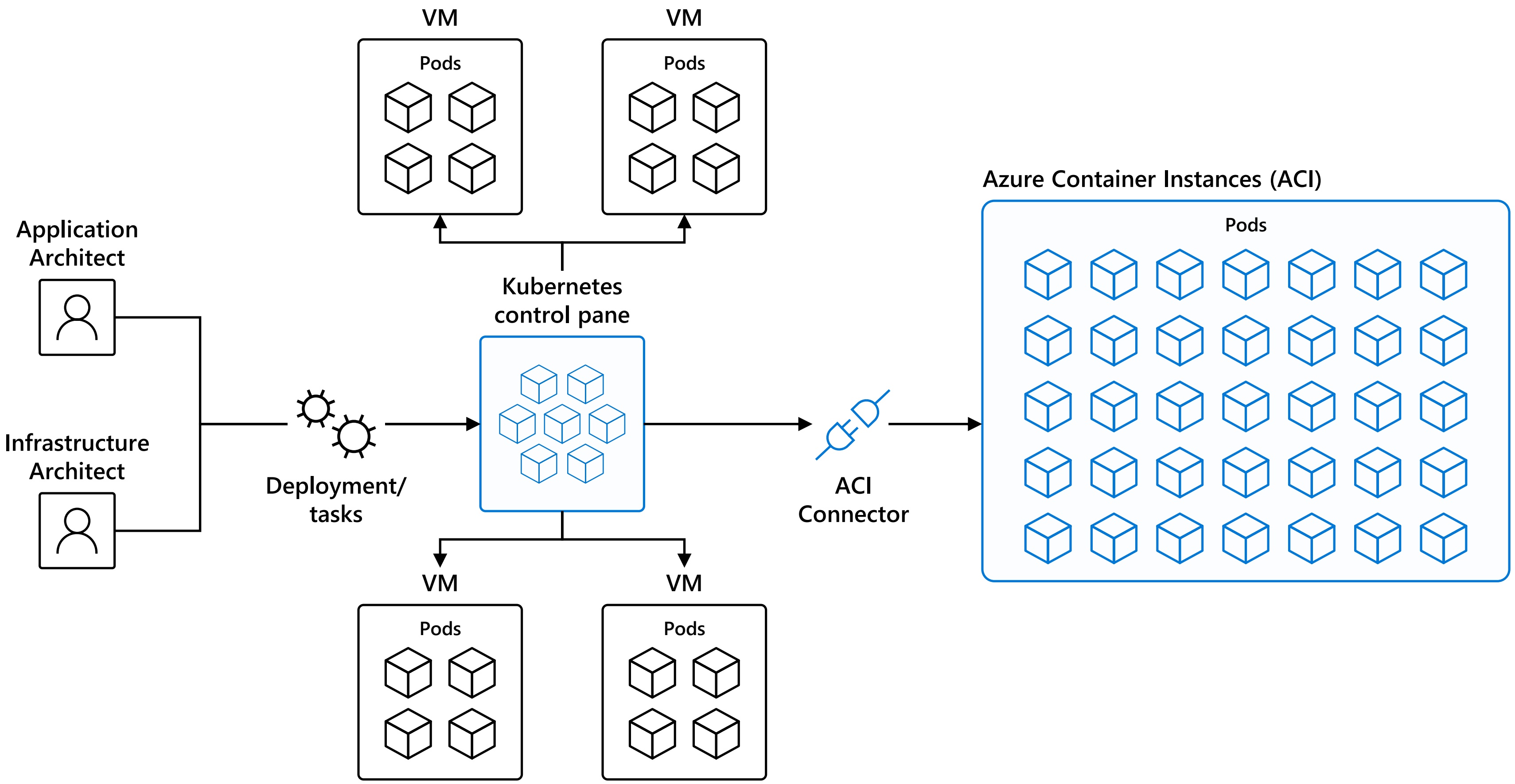


Cluster Autoscaler

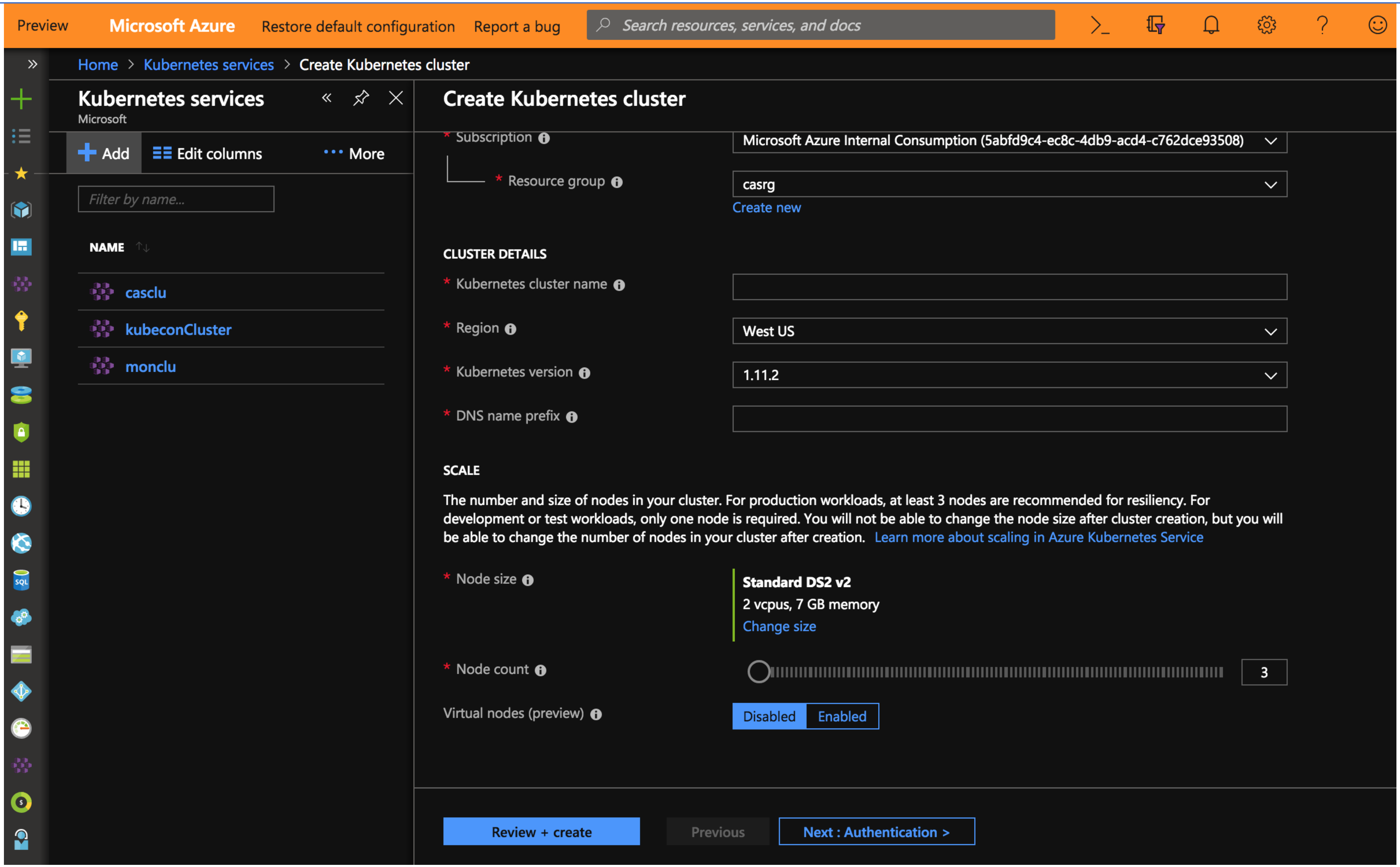
- Scales nodes based on pending pods
- Scale up and scale down
- Reduces dependency on monitoring
- Removes need for users to manage nodes and monitor service usage manually


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

Bursting with the ACI Connector/ Virtual Kubelet



Fast container autoscaling



Create Kubernetes cluster

* Subscription: Microsoft Azure Internal Consumption (5abfd9c4-ec8c-4db9-acd4-c762dce93508)

* Resource group: casrg

Create new

CLUSTER DETAILS

* Kubernetes cluster name:

* Region: West US

* Kubernetes version: 1.11.2

* DNS name prefix:

SCALE

The number and size of nodes in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. You will not be able to change the node size after cluster creation, but you will be able to change the number of nodes in your cluster after creation. Learn more about scaling in Azure Kubernetes Service

* Node size: **Standard DS2 v2**
2 vcpus, 7 GB memory
Change size

* Node count: 3

Virtual nodes (preview): [✓ Disabled] Enabled

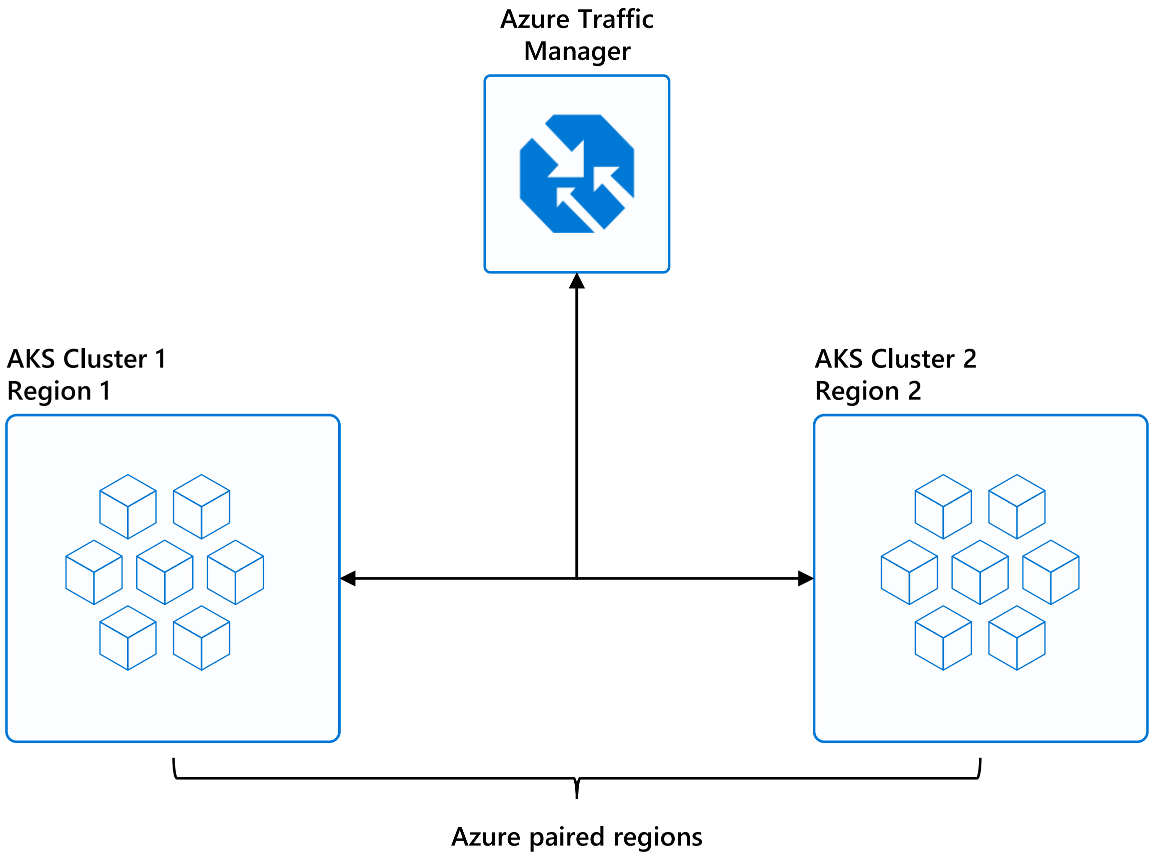
Demo

Cluster autoscaler
AKS + VK burst ACI

Multi-Region

Multi-Region Clusters

- Minimize downtime risk
- One live region
  - Another backup
  - Or weighted traffic
- A/B testing



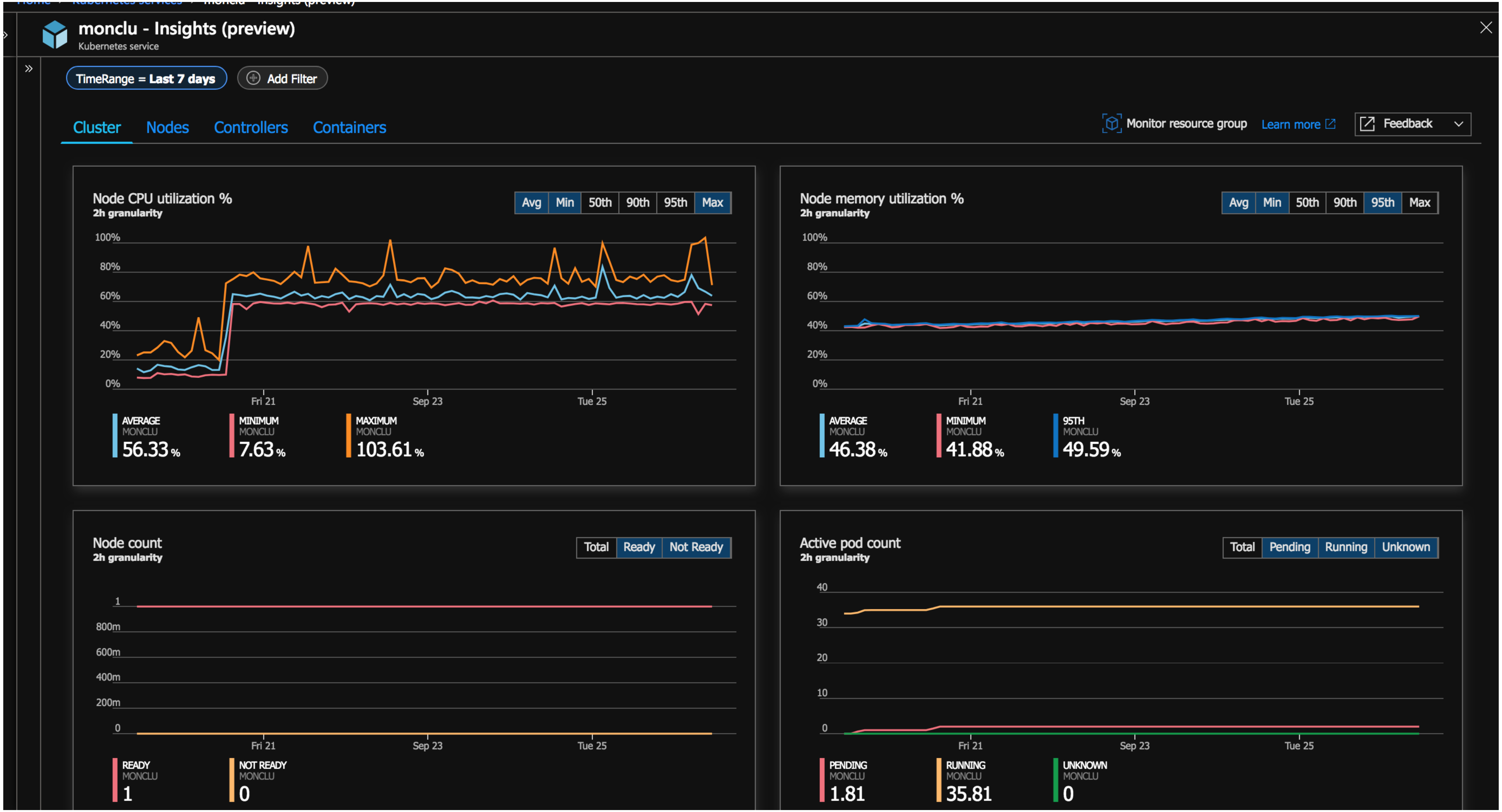
Logging and Monitoring

Monitoring/Logging your cluster

- Log Everything to stdout / stderr
- Key Metrics:
  - Node metrics (CPU Usage, Memory Usage, Disk Usage, Network Usage)
  - Kube_node_status_condition
  - Pod memory usage / limit; memory_failures_total
    - container_memory_working_set_bytes
  - Pod CPU usage average / limit
  - Filesystem Usage / limit
  - Network receive / transmit errors
- Azure Monitor for Containers

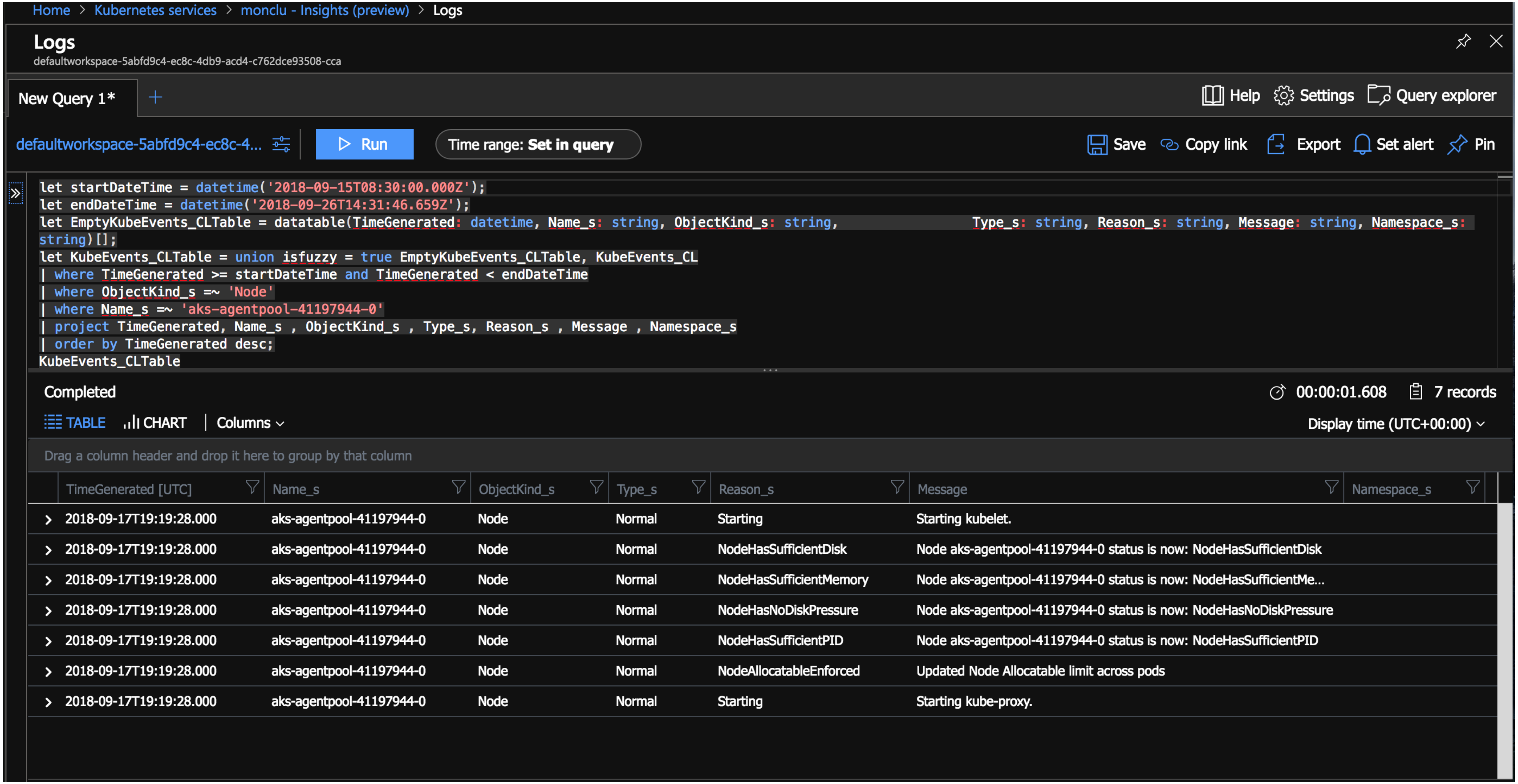
In the roadmap

Overview health of AKS cluster



TimeRange = **Last 7 days**

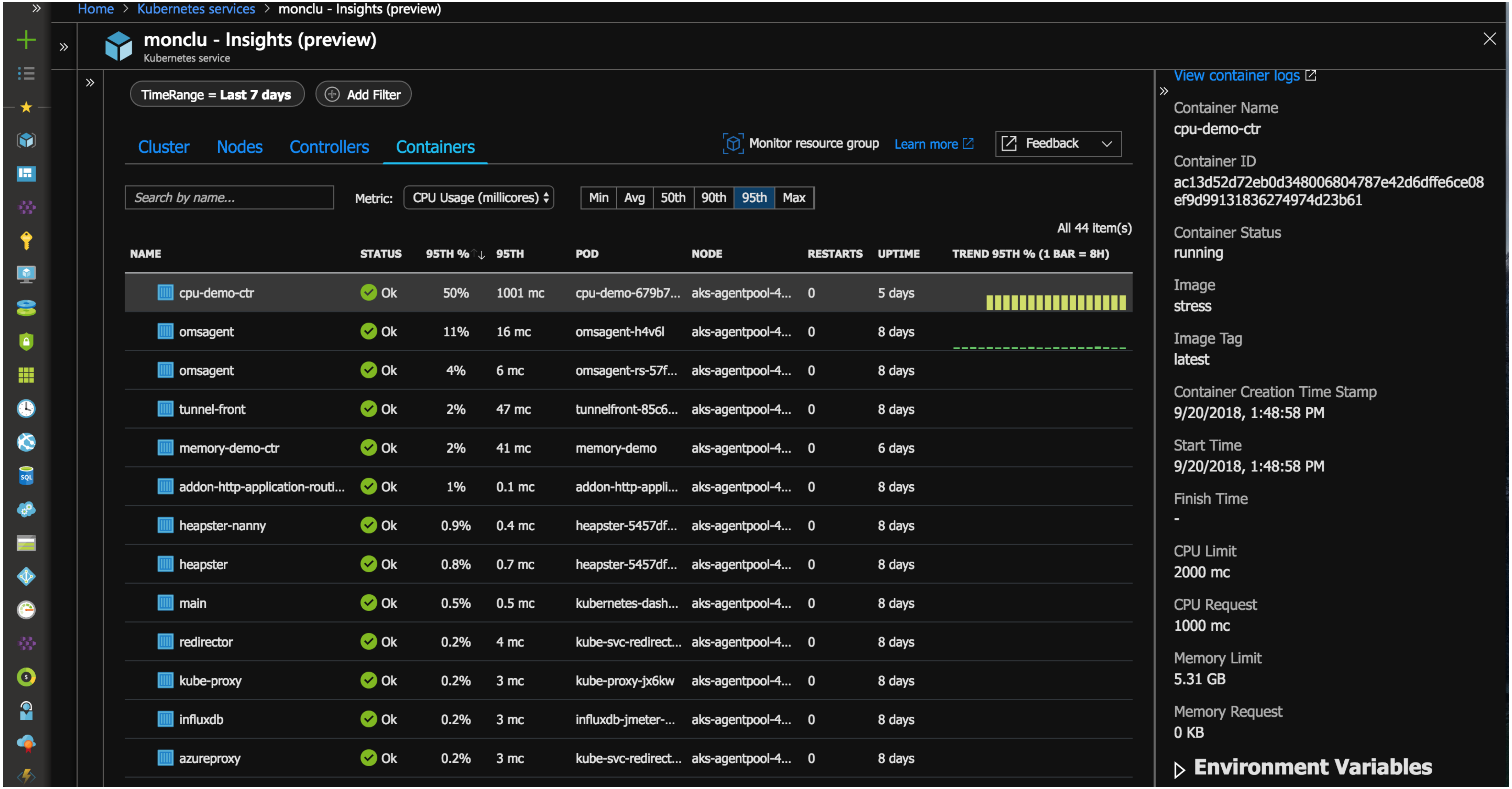
Node event Logs


Home > Kubernetes services > monclu - Insights (preview) > Logs
Logs
defaultworkspace-5abfd9c4-ec8c-4db9-acd4-c762dce93508-cca
New Query 1*
Help
Settings
Query explorer
defaultworkspace-5abfd9c4-ec8c-4...
Run
Time range: Set in query
Save
Copy link
Export
Set alert
Pin
Completed
00:00:01.608
7 records
TABLE
CHART
Columns
Display time (UTC+00:00)
Drag a column header and drop it here to group by that column

```
let startDateTime = datetime('2018-09-15T08:30:00.000Z');
let endDateTime = datetime('2018-09-26T14:31:46.659Z');
let EmptyKubeEvents_CLTable = datatable(TimeGenerated: datetime, Name_s: string, ObjectKind_s: string,            Type_s: string, Reason_s: string, Message: string, Namespace_s:
string)[];
let KubeEvents_CLTable = union isfuzzy = true EmptyKubeEvents_CLTable, KubeEvents_CL
| where TimeGenerated >= startDateTime and TimeGenerated < endDateTime
| where ObjectKind_s =~ 'Node'
| where Name_s =~ 'aks-agentpool-41197944-0'
| project TimeGenerated, Name_s , ObjectKind_s , Type_s, Reason_s , Message , Namespace_s
| order by TimeGenerated desc;
KubeEvents_CLTable
```

TimeGenerated [UTC]	Name_s	ObjectKind_s	Type_s	Reason_s	Message	Namespace_s
2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	Starting	Starting kubelet.	
2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasSufficientDisk	Node aks-agentpool-41197944-0 status is now: NodeHasSufficientDisk	
2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasSufficientMemory	Node aks-agentpool-41197944-0 status is now: NodeHasSufficientMe...	
2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasNoDiskPressure	Node aks-agentpool-41197944-0 status is now: NodeHasNoDiskPressure	
2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasSufficientPID	Node aks-agentpool-41197944-0 status is now: NodeHasSufficientPID	
2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeAllocatableEnforced	Updated Node Allocatable limit across pods	
2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	Starting	Starting kube-proxy.	

Pod usage and details



Metric: CPU Usage (millicores)

All 44 item(s)

NAME	STATUS	95TH %	95TH	POD	NODE	RESTARTS	UPTIME	TREND 95TH % (1 BAR = 8H)
cpu-demo-ctr	✓ Ok	50%	1001 mc	cpu-demo-679b7...	aks-agentpool-4...	0	5 days	
omsagent	✓ Ok	11%	16 mc	omsagent-h4v6l	aks-agentpool-4...	0	8 days	
omsagent	✓ Ok	4%	6 mc	omsagent-rs-57f...	aks-agentpool-4...	0	8 days	
tunnel-front	✓ Ok	2%	47 mc	tunnelfront-85c6...	aks-agentpool-4...	0	8 days	
memory-demo-ctr	✓ Ok	2%	41 mc	memory-demo	aks-agentpool-4...	0	6 days	
addon-http-application-routi...	✓ Ok	1%	0.1 mc	addon-http-appli...	aks-agentpool-4...	0	8 days	
heapster-nanny	✓ Ok	0.9%	0.4 mc	heapster-5457df...	aks-agentpool-4...	0	8 days	
heapster	✓ Ok	0.8%	0.7 mc	heapster-5457df...	aks-agentpool-4...	0	8 days	
main	✓ Ok	0.5%	0.5 mc	kubernetes-dash...	aks-agentpool-4...	0	8 days	
redirector	✓ Ok	0.2%	4 mc	kube-svc-redirect...	aks-agentpool-4...	0	8 days	
kube-proxy	✓ Ok	0.2%	3 mc	kube-proxy-jx6kw	aks-agentpool-4...	0	8 days	
influxdb	✓ Ok	0.2%	3 mc	influxdb-jmeter-...	aks-agentpool-4...	0	8 days	
azureproxy	✓ Ok	0.2%	3 mc	kube-svc-redirect...	aks-agentpool-4...	0	8 days	

Container Name
cpu-demo-ctr

Container ID
ac13d52d72eb0d348006804787e42d6dffe6ce08ef9d99131836274974d23b61

Container Status
running

Image
stress

Image Tag
latest

Container Creation Time Stamp
9/20/2018, 1:48:58 PM

Start Time
9/20/2018, 1:48:58 PM

Finish Time
-

CPU Limit
2000 mc

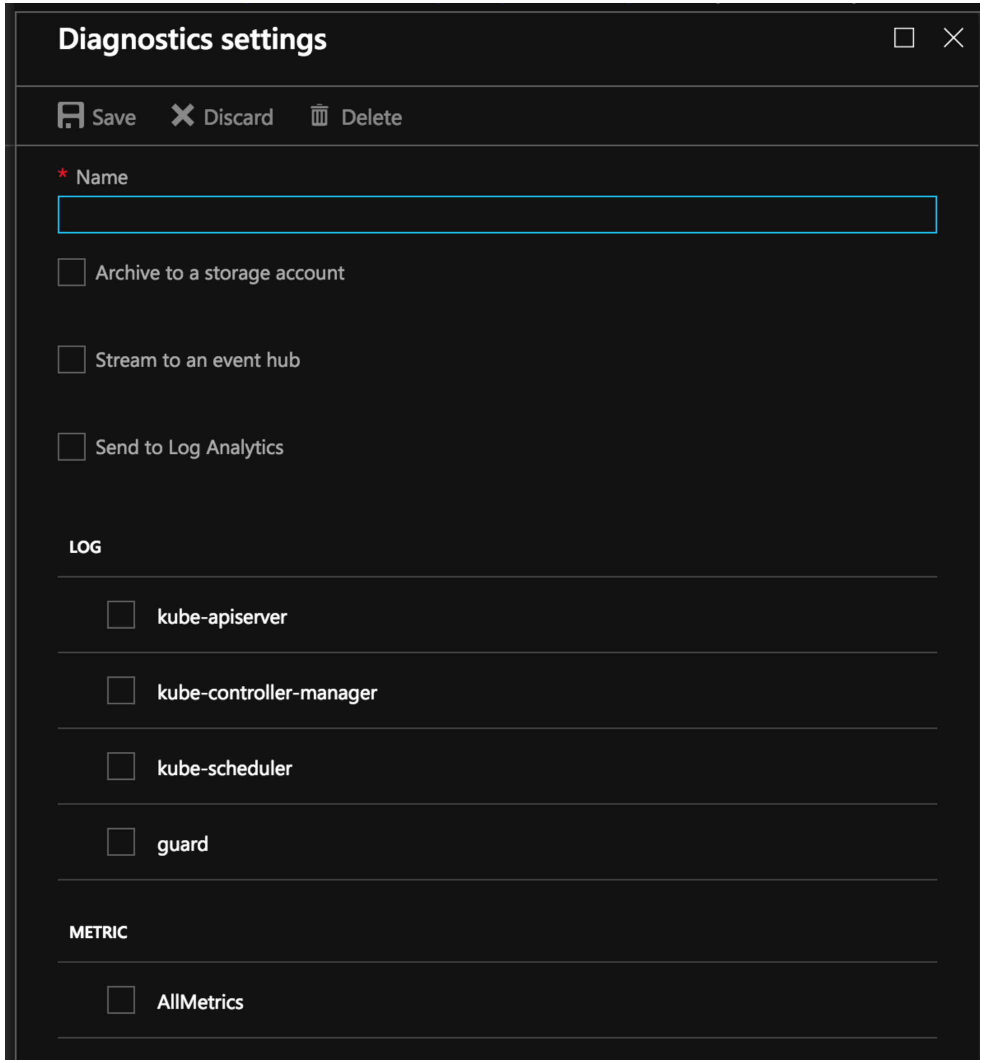
CPU Request
1000 mc

Memory Limit
5.31 GB

Memory Request
0 KB

Customer control plane logs

- Use the Azure portal to enable diagnostics logs
- Pipe logs to log analytics, event hub or a storage account
- Metrics available today
  - Kube-controller-manager
  - Kube-api-server
  - Kube-scheduler
  - Audit logs on the roadmap



* Name

□ Archive to a storage account

□ Stream to an event hub

□ Send to Log Analytics

LOG

□ kube-apiserver

□ kube-controller-manager

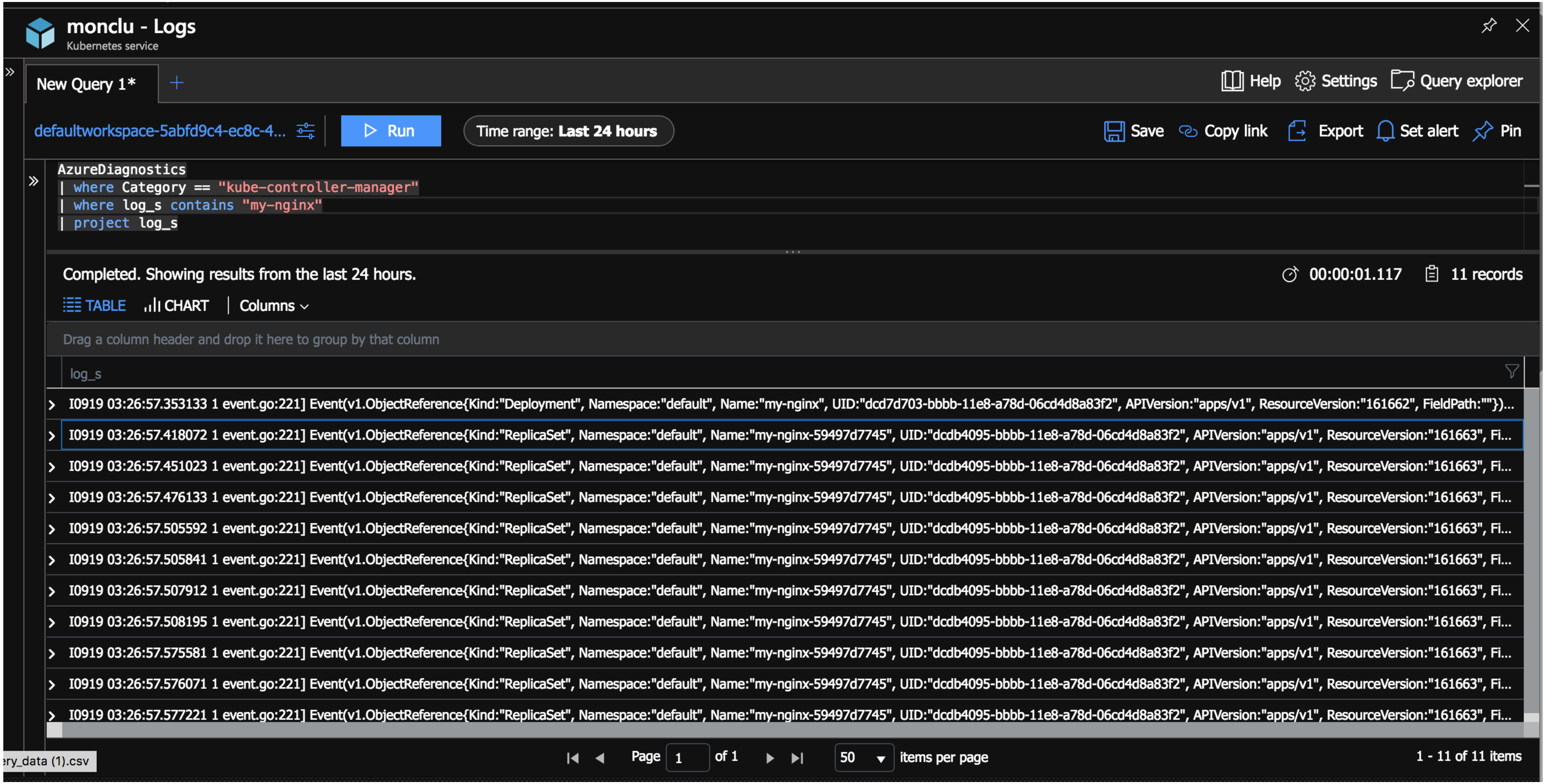
□ kube-scheduler

□ guard

METRIC

□ AllMetrics

Example control plane logs



```
AzureDiagnostics
| where Category == "kube-controller-manager"
| where log_s contains "my-nginx"
| project log_s
```

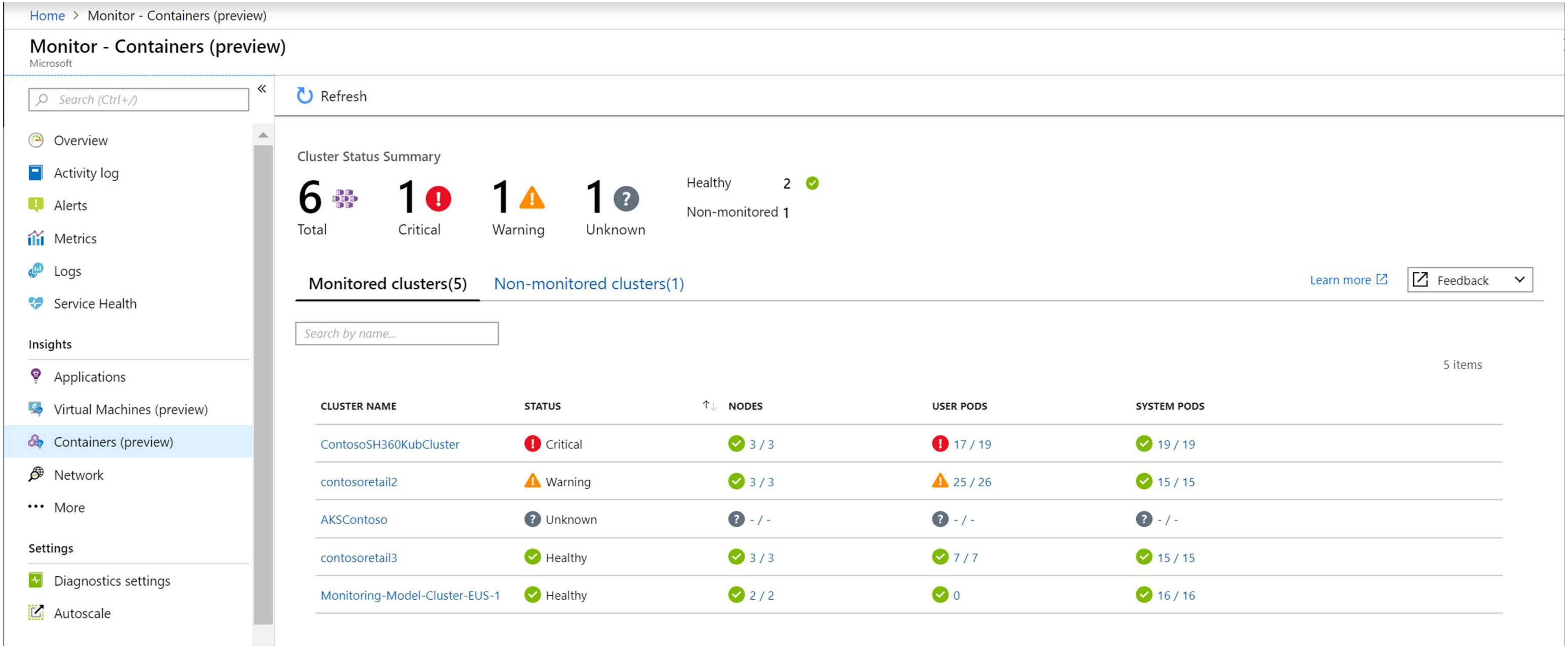
Completed. Showing results from the last 24 hours.

00:00:01.117 11 records

```
log_s
I0919 03:26:57.353133 1 event.go:221] Event(v1.ObjectReference{Kind:"Deployment", Namespace:"default", Name:"my-nginx", UID:"dcd7d703-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161662", FieldPath:""})...
I0919 03:26:57.418072 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.451023 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.476133 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.505592 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.505841 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.507912 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.508195 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.575581 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.576071 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
I0919 03:26:57.577221 1 event.go:221] Event(v1.ObjectReference{Kind:"ReplicaSet", Namespace:"default", Name:"my-nginx-59497d7745", UID:"dcdb4095-bbbb-11e8-a78d-06cd4d8a83f2", APIVersion:"apps/v1", ResourceVersion:"161663", Fi...
```

Page 1 of 1 — 50 items per page — 1 - 11 of 11 items

Multi cluster monitoring



Home > Monitor - Containers (preview)

Monitor - Containers (preview)
Microsoft

Cluster Status Summary

6	1 !	1 ⚠	1 ?
Total	Critical	Warning	Unknown

Healthy 2 ✓
Non-monitored 1

5 items

CLUSTER NAME	STATUS	NODES	USER PODS	SYSTEM PODS
ContosoSH360KubCluster	! Critical	✓ 3 / 3	! 17 / 19	✓ 19 / 19
contosoretail2	⚠ Warning	✓ 3 / 3	⚠ 25 / 26	✓ 15 / 15
AKSContoso	? Unknown	? - / -	? - / -	? - / -
contosoretail3	✓ Healthy	✓ 3 / 3	✓ 7 / 7	✓ 15 / 15
Monitoring-Model-Cluster-EUS-1	✓ Healthy	✓ 2 / 2	✓ 0	✓ 16 / 16

Demo

Monitoring and logging (Saurya)

1) Node/pod usage, kube events
2) Pod hogging resource, show resource request limits
3) Talk about percentile for capacity planning
4) Show filter –kube-system
5) https://aka.ms/multiaksinsights

Resources

Resources

- AKS Best Practices GitHub: https://github.com/Azure/k8s-best-practices
- AKS Hackfest: aka.ms/k8s-hackfest & https://github.com/Azure/kubernetes-hackfest
- Distributed systems Labs by Brendan Burns
- Kube Advisor: https://github.com/Azure/kube-advisor
- VSCode Kubernetes Extension
- Documentation resources
- Ebook for distributed systems
- AKS HoL
- Connect with us on twitter:
  - Jorge Palma - @jorgefpalma
  - Mohammad Nofal - @mohmd_nofal
  - Saurya Das - @sauryadas

  Andrew Randall - @andrew_randall

Thank You!

Microsoft