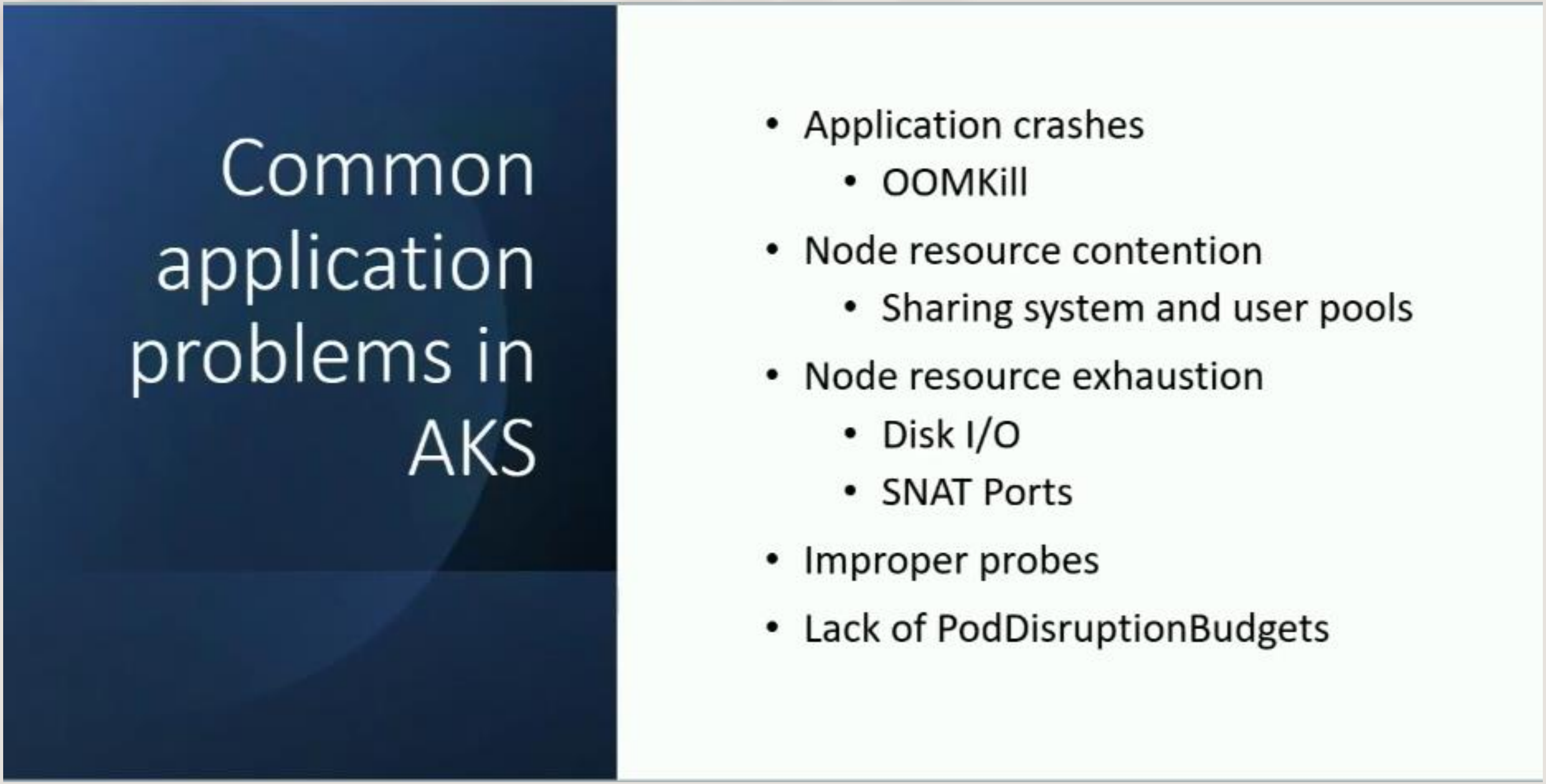
Maheshkumar R

Cloud Solution Architect, Microsoft Azure

maheshk@microsoft.com

https://github.com/Maheshk-MSFT/

Troubleshooting your Apps on AKS



Common application problems in AKS

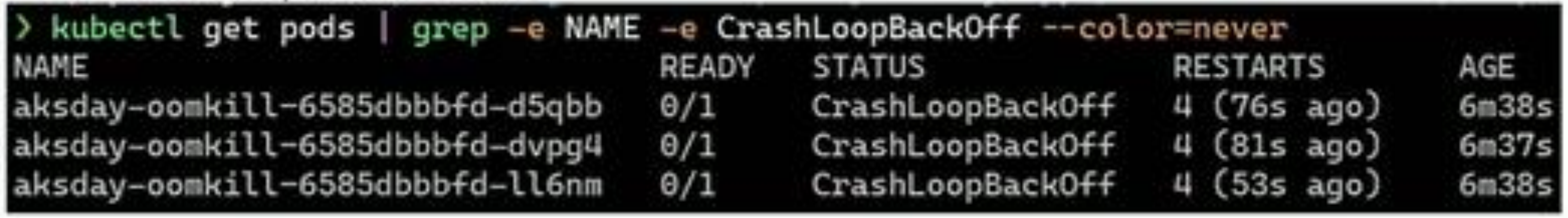
- Application crashes
  - OOMKill
- Node resource contention
  - Sharing system and user pools
- Node resource exhaustion
  - Disk I/O
  - SNAT Ports
- Improper probes
- Lack of PodDisruptionBudgets

My pods keep restarting or are nonresponsive

- Application crashes
  - Check pod logs via Container Insights or kubectl logs
  - `kubectl logs -p` will show the logs from the previous iteration of the pod – useful if your pod is restarting!
- Check your resource limits
  - Pods can’t exceed the CPU limits – they just get capped.
  - Pods that exceed the memory limits **get OOMKilled and restart**.
- CPU, memory, disk, or PID pressure
  - Make sure you have appropriate process limits set
  - Move to larger nodes if appropriate
  - Add anti-affinity rules to keep heavy talkers apart
  - Use dedicated node pools if necessary

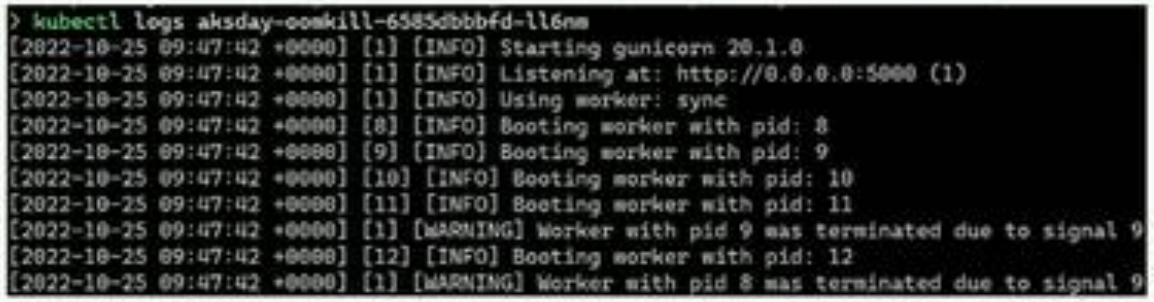
My pods keep restarting or are nonresponsive

- Pods with frequent restarts



```
> kubectl get pods | grep -e NAME -e CrashLoopBackOff --color=never
NAME                                READY   STATUS             RESTARTS       AGE
aksday-oomkill-6585dbbbfd-d5qbb     0/1     CrashLoopBackOff   4 (76s ago)    6m38s
aksday-oomkill-6585dbbbfd-dvpg4     0/1     CrashLoopBackOff   4 (81s ago)    6m37s
aksday-oomkill-6585dbbbfd-ll6nm     0/1     CrashLoopBackOff   4 (53s ago)    6m38s
```

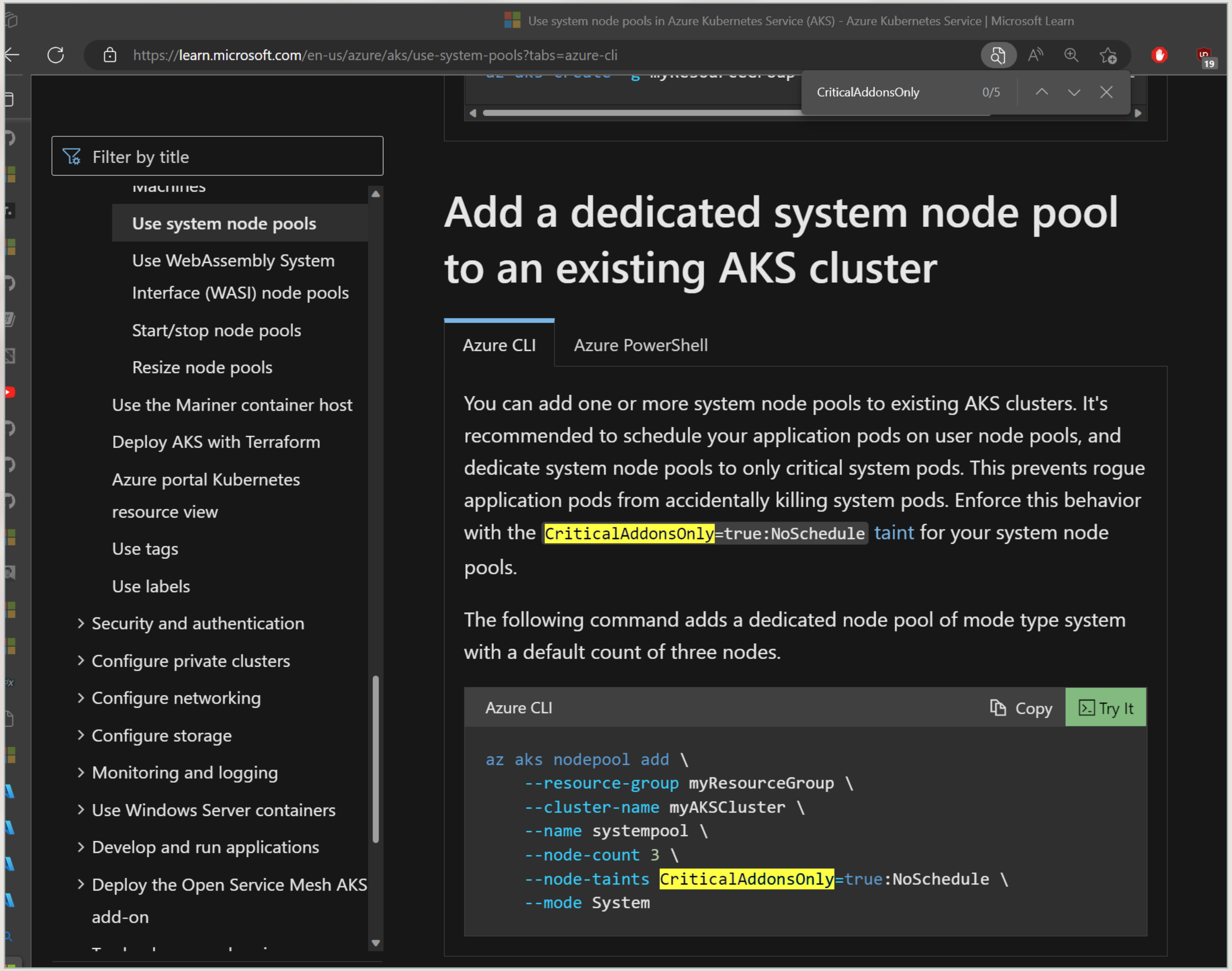
- OOMKill



```
> kubectl logs aksday-oomkill-6585dbbbfd-ll6nm
[2022-10-25 09:47:42 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2022-10-25 09:47:42 +0000] [1] [INFO] Listening at: http://0.0.0.0:5000 (1)
[2022-10-25 09:47:42 +0000] [1] [INFO] Using worker: sync
[2022-10-25 09:47:42 +0000] [8] [INFO] Booting worker with pid: 8
[2022-10-25 09:47:42 +0000] [9] [INFO] Booting worker with pid: 9
[2022-10-25 09:47:42 +0000] [10] [INFO] Booting worker with pid: 10
[2022-10-25 09:47:42 +0000] [11] [INFO] Booting worker with pid: 11
[2022-10-25 09:47:42 +0000] [1] [WARNING] Worker with pid 9 was terminated due to signal 9
[2022-10-25 09:47:42 +0000] [12] [INFO] Booting worker with pid: 12
[2022-10-25 09:47:42 +0000] [1] [WARNING] Worker with pid 8 was terminated due to signal 9
```

My pods keep restarting or are nonresponsive

- Use separate system and user pools
    - Shared pools can escalate a runaway app into a cluster outage if core cluster functionality like DNS is impacted
- Taint system pools with CriticalAddonsOnly
- Make sure your applications have appropriate resources and limits
    - Be very careful with memory
- Use pod anti-affinity where appropriate to separate chatty pods on CPU, memory, or disk I/O


Use system node pools in Azure Kubernetes Service (AKS) - Azure Kubernetes Service | Microsoft Learn
https://learn.microsoft.com/en-us/azure/aks/use-system-pools?tabs=azure-cli
CriticalAddonsOnly
0/5
Filter by title
Machines
Use system node pools
Use WebAssembly System Interface (WASI) node pools
Start/stop node pools
Resize node pools
Use the Mariner container host
Deploy AKS with Terraform
Azure portal Kubernetes resource view
Use tags
Use labels
Security and authentication
Configure private clusters
Configure networking
Configure storage
Monitoring and logging
Use Windows Server containers
Develop and run applications
Deploy the Open Service Mesh AKS add-on
Azure CLI
Azure PowerShell
Copy
Try It

Add a dedicated system node pool to an existing AKS cluster

You can add one or more system node pools to existing AKS clusters. It's recommended to schedule your application pods on user node pools, and dedicate system node pools to only critical system pods. This prevents rogue application pods from accidentally killing system pods. Enforce this behavior with the `CriticalAddonsOnly=true:NoSchedule` taint for your system node pools.

The following command adds a dedicated node pool of mode type system with a default count of three nodes.

Azure CLI

```
az aks nodepool add \
    --resource-group myResourceGroup \
    --cluster-name myAKSCluster \
    --name systempool \
    --node-count 3 \
    --node-taints CriticalAddonsOnly=true:NoSchedule \
    --mode System
```

https://learn.microsoft.com/en-us/azure/aks/use-system-pools?tabs=azure-cli

My app crashes when I upgrade my cluster

- Deployments have a rolling update strategy: maxUnavailable pods
- This **only applies to deployment rollouts**
- Applications must have a PodDisruptionBudget to deal with disruptions in the cluster, including
  - Hardware failures
  - VM scaling operations
  - Cluster upgrade operations
  - Network issues
  - Resource constraint evictions
- Without a PDB, nothing stops the scheduler from restarting all of your pods at the same time, even if you have maxUnavailable set to 1 in your Deployment.

My pods on a node are nonresponsive

- Container root filesystems and ephemeral volumes are part of the node/VM root filesystem
- Disk I/O throttling – two queues to consider
  - VM maximum rate – varies by VM SKU
  - Disk maximum rate – varies by OS disk size
- Much less common in newer clusters
  - AKS defaults to ephemeral disk if the SKU supports caching/temp disk and has enough space
  - The default OS disk size was increased from the early value of 30GB to 128GB+
- Solutions
  - Use ephemeral OS disks
  - Use SKUs with premium disk support
  - Use a larger OS disk

AKS VM SKU selection

VM SKU determines a lot when it comes to disk I/O (see the VM SKU naming conventions):

[Family] + *[Sub-family]** + **[# of vCPUs]** + *[Constrained vCPUs]** + **[Additive Features]** + *[Accelerator Type]** + **[Version]**

The additive features are key for AKS:

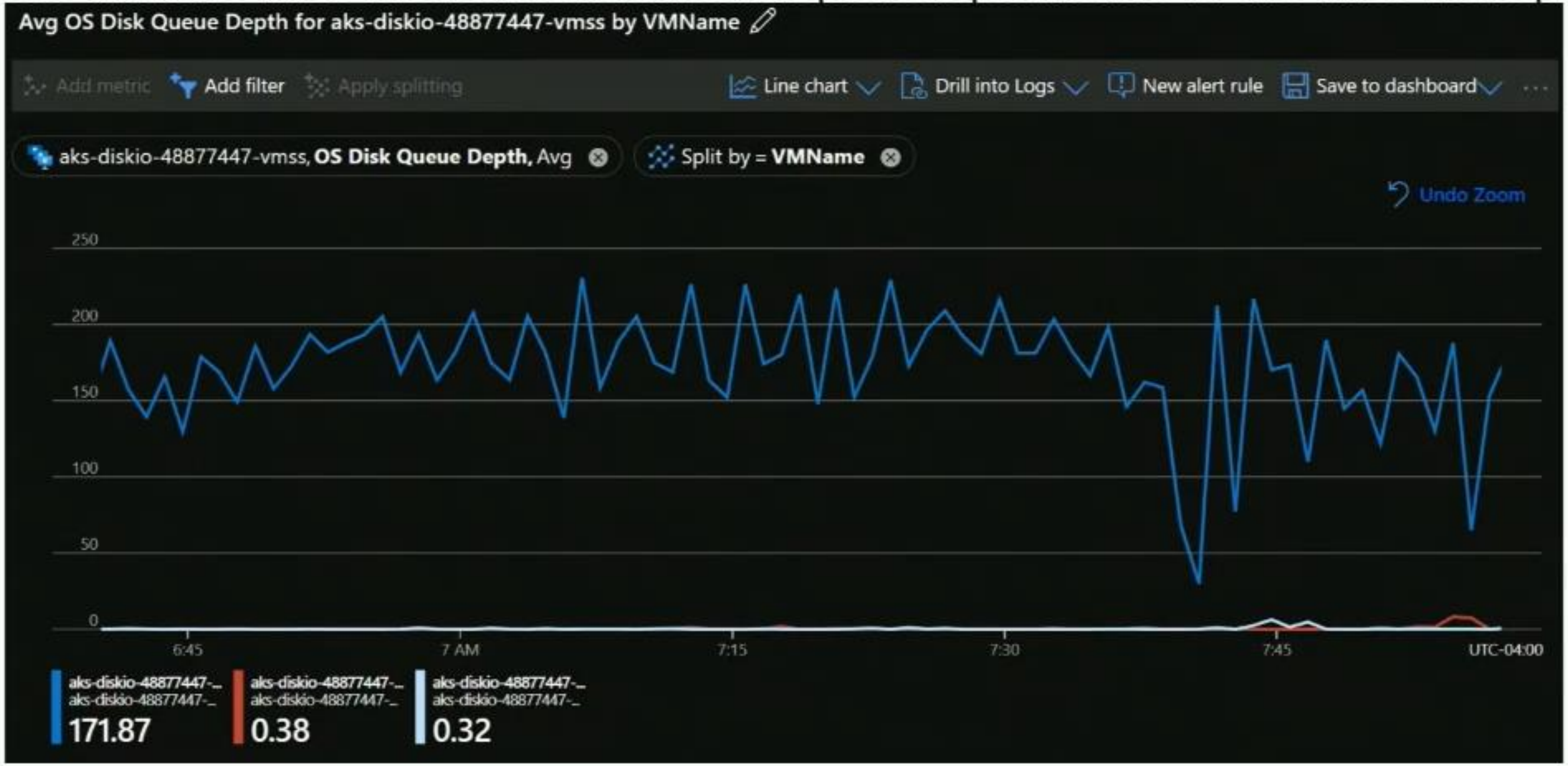
- SKUs with "s" in additive features support premium storage and will have much higher disk performance.
- SKUs with "d" in additive features (v4 and newer) can be used with Ephemeral OS disks, which provide much higher throughput by hosting the OS image in local temporary storage.
- V3 and older SKUs may support ephemeral OS disk; this can be determined on the VM SKU documentation page.

Example:

- Standard_D4_v4 – no premium storage, no ephemeral OS disk. Lowest I/O performance.
- Standard_D4s_v4 – premium storage but no ephemeral OS disk. Better I/O performance but can hit VM or disk limits.
- Standard_D4d_v4 – no premium storage, but supports ephemeral OS disk. Provides excellent performance for the OS disk and local storage, but doesn't support premium Azure Disks for external storage.
- Standard_D4ds_v4 – supports both premium storage and ephemeral OS disk. Provides the best I/O capabilities of the 4.

My pods on a node are nonresponsive

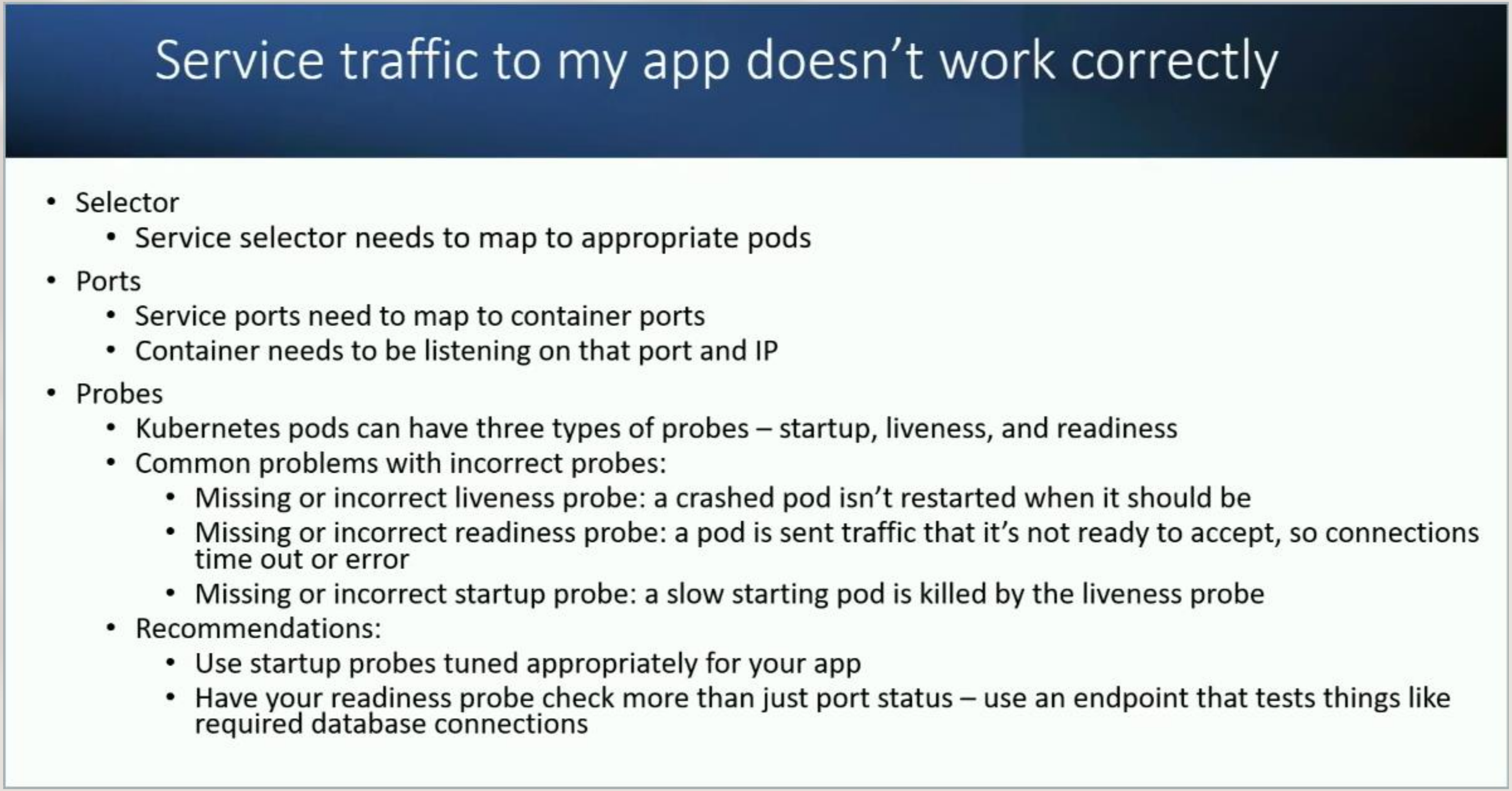
- Diagnose via Azure VMSS metrics: Look at the disk queue depth metrics for the VMSS in the portal



Avg OS Disk Queue Depth for aks-diskio-48877447-vmss by VMName

aks-diskio-48877447-vmss, **OS Disk Queue Depth,** Avg

Split by = **VMName**

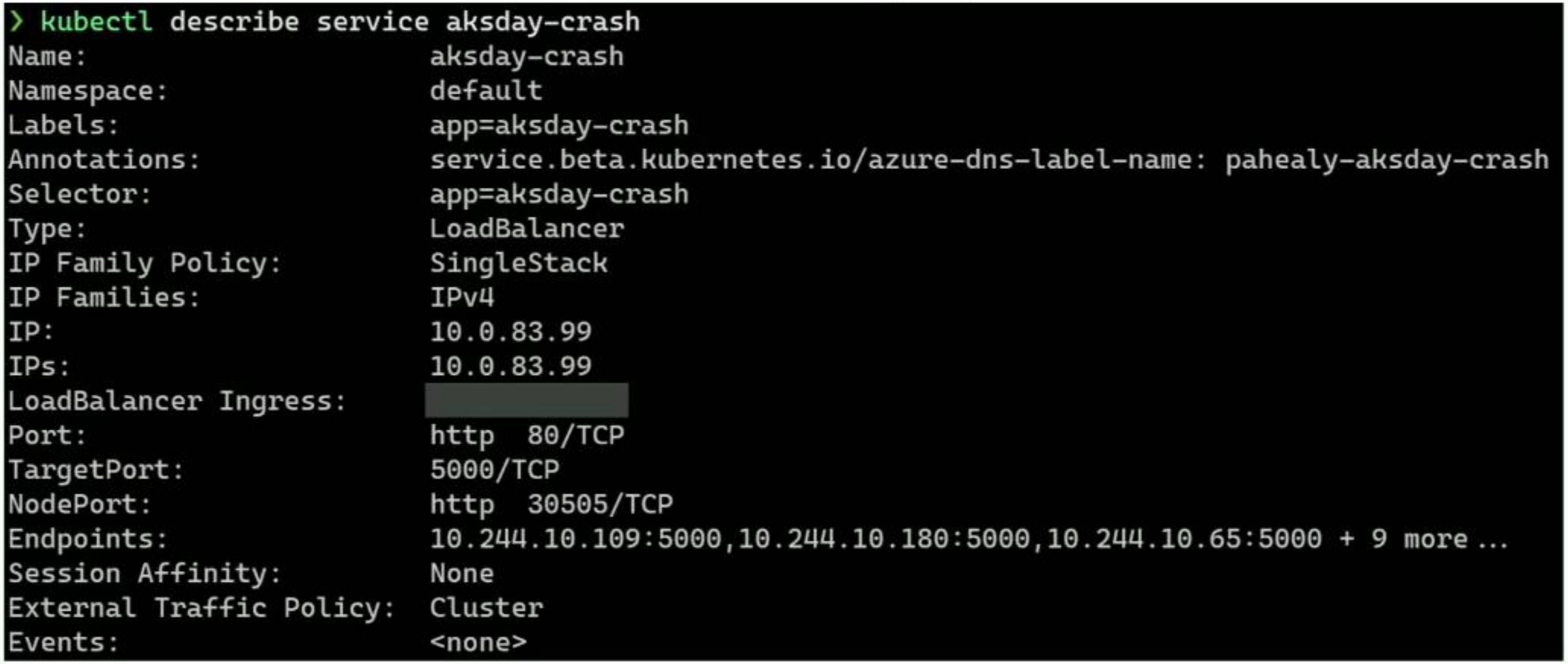


Service traffic to my app doesn’t work correctly

- Selector
  - Service selector needs to map to appropriate pods
- Ports
  - Service ports need to map to container ports
  - Container needs to be listening on that port and IP
- Probes
  - Kubernetes pods can have three types of probes – startup, liveness, and readiness
  - Common problems with incorrect probes:
    - Missing or incorrect liveness probe: a crashed pod isn’t restarted when it should be
    - Missing or incorrect readiness probe: a pod is sent traffic that it’s not ready to accept, so connections time out or error
    - Missing or incorrect startup probe: a slow starting pod is killed by the liveness probe
  - Recommendations:
    - Use startup probes tuned appropriately for your app
    - Have your readiness probe check more than just port status – use an endpoint that tests things like required database connections

Service traffic to my app doesn’t work correctly

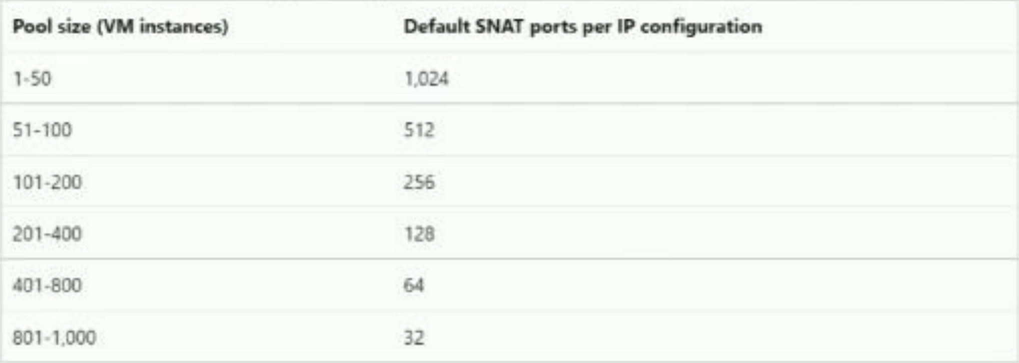
- Describe the service and make sure it has the appropriate information



```
> kubectl describe service aksday-crash
Name:                     aksday-crash
Namespace:                default
Labels:                   app=aksday-crash
Annotations:              service.beta.kubernetes.io/azure-dns-label-name: pahealy-aksday-crash
Selector:                 app=aksday-crash
Type:                     LoadBalancer
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.0.83.99
IPs:                      10.0.83.99
LoadBalancer Ingress:     [redacted]
Port:                     http  80/TCP
TargetPort:               5000/TCP
NodePort:                 http  30505/TCP
Endpoints:                10.244.10.109:5000,10.244.10.180:5000,10.244.10.65:5000 + 9 more...
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

My app has problems when I have lots of connections

- Occurs when there are many connections in a short period of time from the same node to the same IP:port destination
- Default configuration has some quirks due to needing to support many configurations
  - Allocated SNAT ports per node **decreases** as the cluster node count **increases**



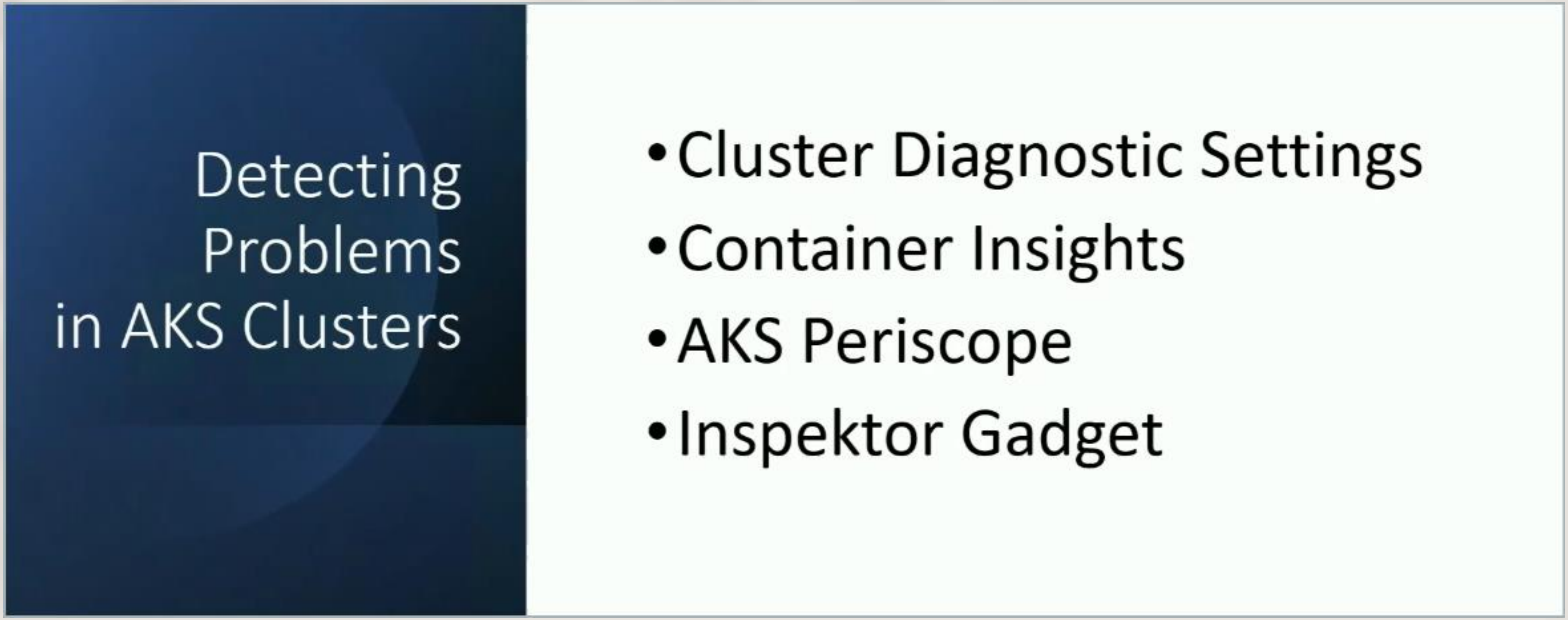
Pool size (VM instances)	Default SNAT ports per IP configuration
1-50	1,024
51-100	512
101-200	256
201-400	128
401-800	64
801-1,000	32

- Solutions
  - Spread out the connections across more pods and nodes
  - Use NAT Gateway for egress instead of SLB
  - Tune outbound SNAT ports and IP count per node

To calculate the number of required IPs for an AKS cluster:

$$\frac{portsPerNode\ * maxNodeCount}{64000}$$

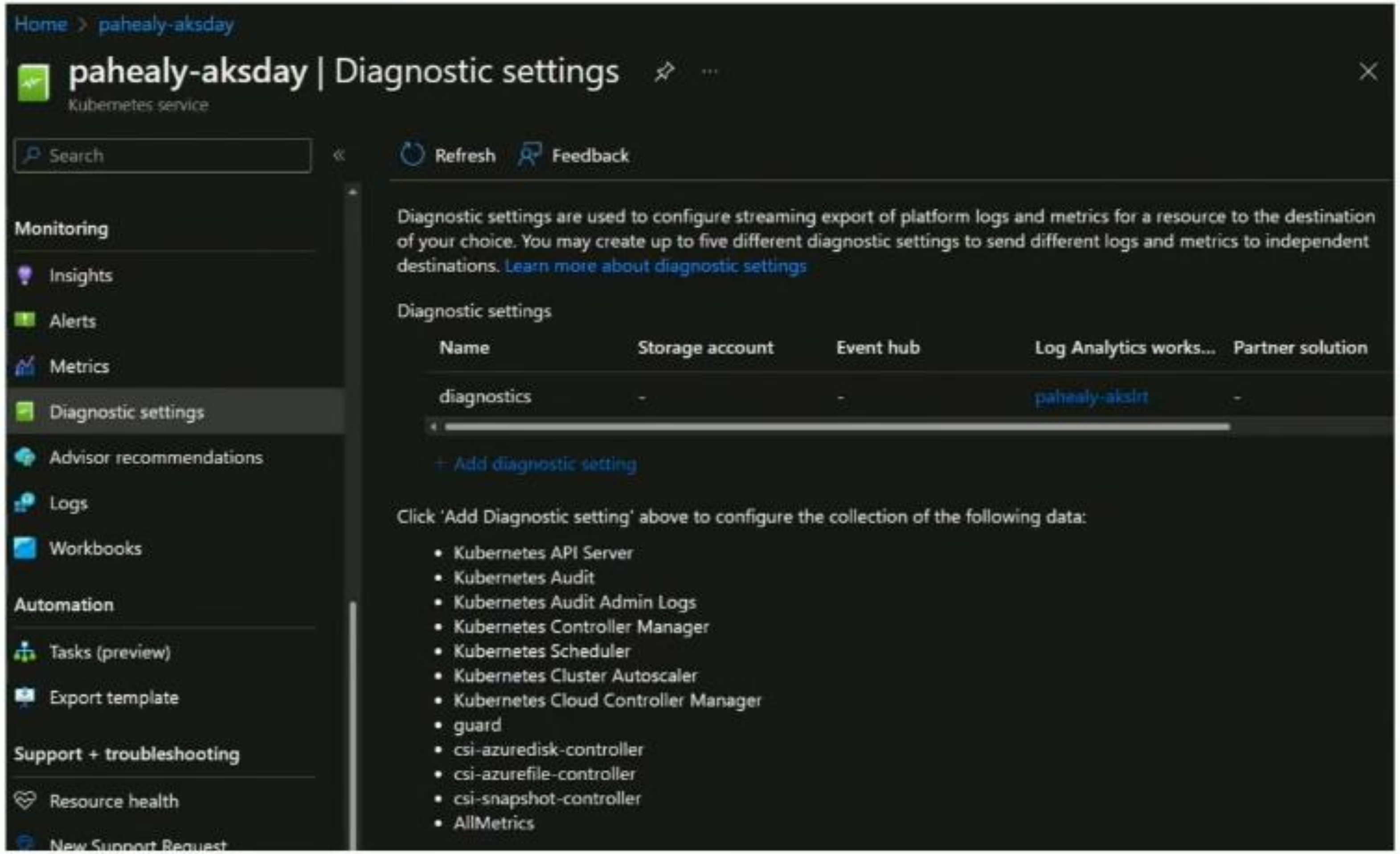
- portsPerNode must be a power of 2
- Round up to the next integer IP count



Detecting Problems in AKS Clusters

- Cluster Diagnostic Settings
- Container Insights
- AKS Periscope
- Inspektor Gadget

Cluster Diagnostic Settings



pahealy-aksday | Diagnostic settings

Kubernetes service

Diagnostic settings are used to configure streaming export of platform logs and metrics for a resource to the destination of your choice. You may create up to five different diagnostic settings to send different logs and metrics to independent destinations. Learn more about diagnostic settings

Diagnostic settings

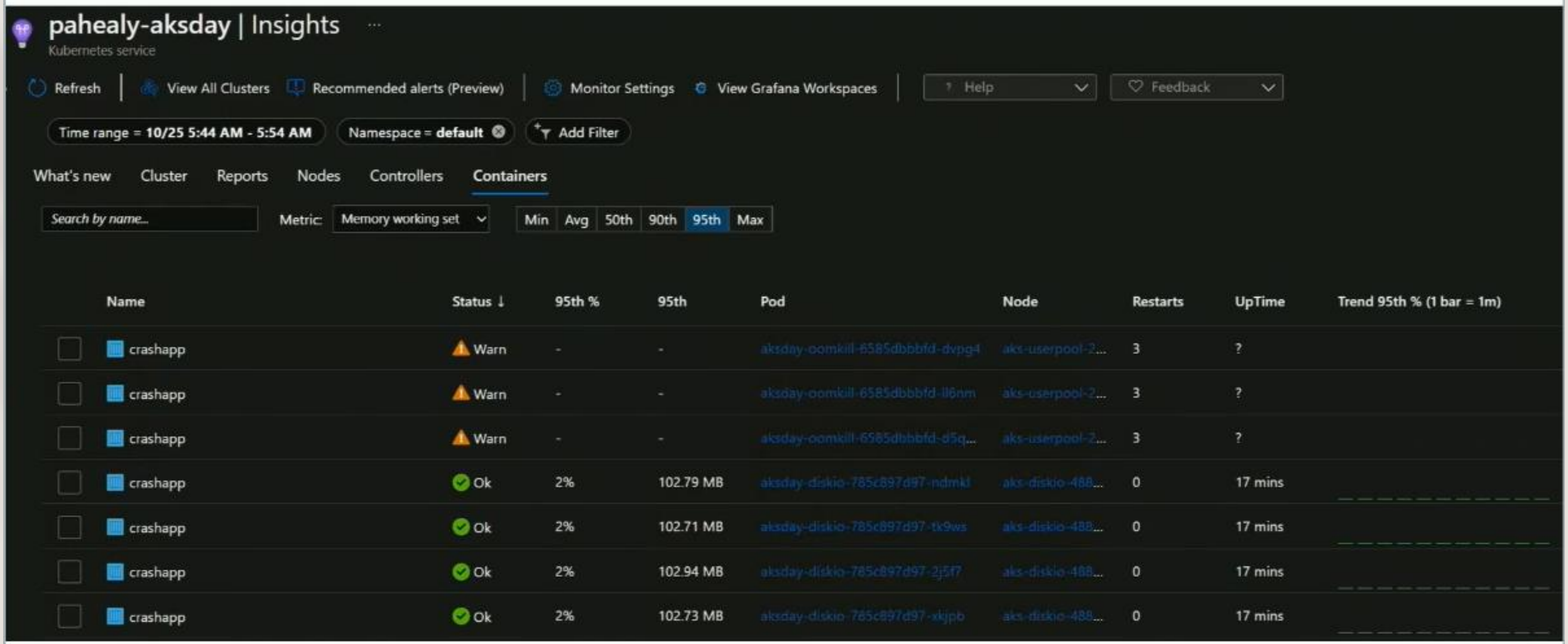
Name	Storage account	Event hub	Log Analytics works...	Partner solution
diagnostics	-	-	pahealy-akslrt	-

+ Add diagnostic setting

Click 'Add Diagnostic setting' above to configure the collection of the following data:

- Kubernetes API Server
- Kubernetes Audit
- Kubernetes Audit Admin Logs
- Kubernetes Controller Manager
- Kubernetes Scheduler
- Kubernetes Cluster Autoscaler
- Kubernetes Cloud Controller Manager
- guard
- csi-azuredisk-controller
- csi-azurefile-controller
- csi-snapshot-controller
- AllMetrics

Container Insights



pahealy-aksday | Insights

Kubernetes service

Time range = **10/25 5:44 AM - 5:54 AM**

Namespace = **default**

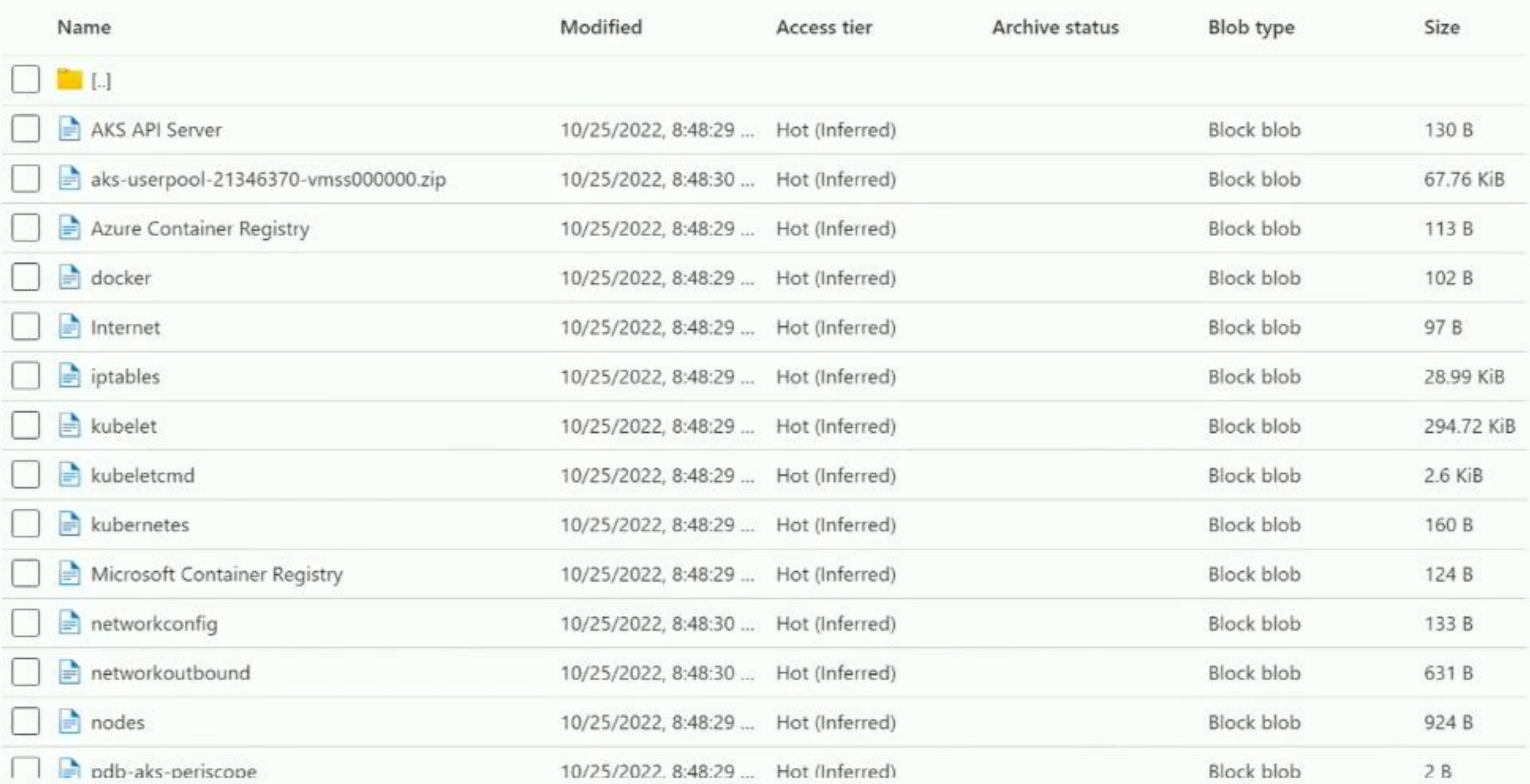
Metric: Memory working set

Name	Status ↓	95th %	95th	Pod	Node	Restarts	UpTime	Trend 95th % (1 bar = 1m)
□ crashapp	⚠ Warn	-	-	aksday-oomkill-6585dbbbfd-dvpg4	aks-userpool-2...	3	?	
□ crashapp	⚠ Warn	-	-	aksday-oomkill-6585dbbbfd-ll6nm	aks-userpool-2...	3	?	
□ crashapp	⚠ Warn	-	-	aksday-oomkill-6585dbbbfd-d5q...	aks-userpool-2...	3	?	
□ crashapp	✓ Ok	2%	102.79 MB	aksday-diskio-785c897d97-ndmkl	aks-diskio-488...	0	17 mins	
□ crashapp	✓ Ok	2%	102.71 MB	aksday-diskio-785c897d97-tk9ws	aks-diskio-488...	0	17 mins	
□ crashapp	✓ Ok	2%	102.94 MB	aksday-diskio-785c897d97-2j5f7	aks-diskio-488...	0	17 mins	
□ crashapp	✓ Ok	2%	102.73 MB	aksday-diskio-785c897d97-xkjpb	aks-diskio-488...	0	17 mins	

AKS Periscope

- Open source tool integrated with Azure CLI:
  ```
  az aks kollect
  ```
- Collects a point-in-time snapshot of the cluster state into a storage account
  - Container logs (by default all containers in the kube-system namespace).
  - Docker and Kubelet system service logs.
  - Network outbound connectivity, include checks for internet, API server, Tunnel, Azure Container Registry and Microsoft Container Registry.
  - Node IP Tables.
  - All node level logs (by default cluster provision log and cloud init log).
  - VM and Kubernetes cluster level DNS settings.
  - Describe Kubernetes objectsKubelet command arguments.
  - System performance (kubectl top nodes and kubectl top pods).
- Gets kube-system logs by default – add the namespace of the app being investigated with into the collection by specifying `--container-logs` and `--kube-objects`
- https://github.com/Azure/aks-periscope

AKS Periscope



	Name	Modified	Access tier	Archive status	Blob type	Size
□	[..]					
□	AKS API Server	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	130 B
□	aks-userpool-21346370-vmss000000.zip	10/25/2022, 8:48:30 ...	Hot (Inferred)		Block blob	67.76 KiB
□	Azure Container Registry	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	113 B
□	docker	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	102 B
□	Internet	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	97 B
□	iptables	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	28.99 KiB
□	kubelet	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	294.72 KiB
□	kubeletcmd	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	2.6 KiB
□	kubernetes	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	160 B
□	Microsoft Container Registry	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	124 B
□	networkconfig	10/25/2022, 8:48:30 ...	Hot (Inferred)		Block blob	133 B
□	networkoutbound	10/25/2022, 8:48:30 ...	Hot (Inferred)		Block blob	631 B
□	nodes	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	924 B
□	pdb-aks-periscope	10/25/2022, 8:48:29 ...	Hot (Inferred)		Block blob	2 B

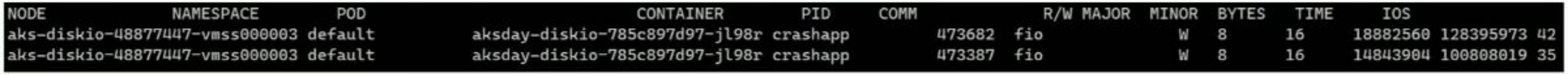
Inspektor Gadget

- Open source tool originally by Kinvolk, which is now a part of Microsoft
- Deploys eBPF plugins to inspect the live state of the cluster
  - CPU utilization
  - Disk I/O
    - Block I/O utilization
    - File accesses (all or only slower than a threshold)
  - Networking
    - Socket bindings
    - DNS queries
    - Connections made, including network streams, TCP connections, and TLS SNI data
  - Process executions

- https://www.inspektor-gadget.io/

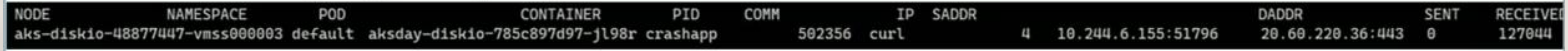
Inspektor Gadget Top

- kubectl gadget top block-io



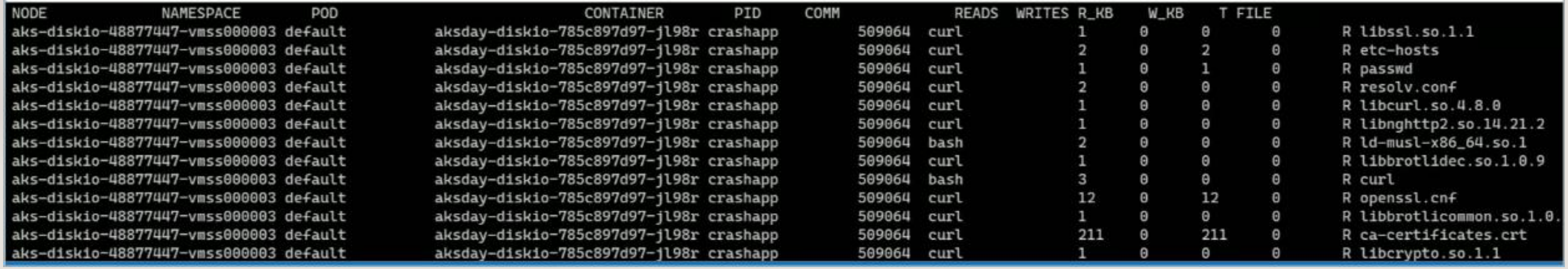
```
NODE              NAMESPACE        POD                        CONTAINER       PID       COMM             R/W MAJOR  MINOR  BYTES    TIME      IOS
aks-diskio-48877447-vmss000003 default         aksday-diskio-785c897d97-jl98r crashapp         473682  fio          W   8       16      18882560 128395973 42
aks-diskio-48877447-vmss000003 default         aksday-diskio-785c897d97-jl98r crashapp         473387  fio          W   8       16      14843904 100808019 35
```

- kubectl gadget top tcp



```
NODE              NAMESPACE       POD                      CONTAINER       PID      COMM              IP  SADDR                                      DADDR                SENT     RECEIVEI
aks-diskio-48877447-vmss000003 default  aksday-diskio-785c897d97-jl98r crashapp         502356   curl              4   10.244.6.155:51796         20.60.220.36:443     0        127044
```

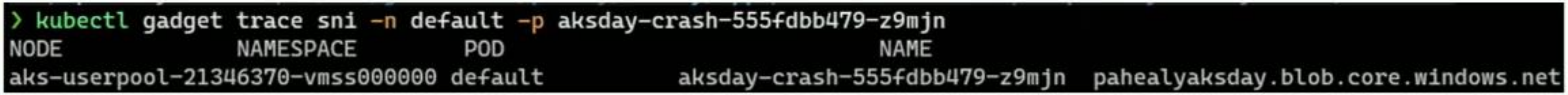
- kubectl gadget top file



```
NODE              NAMESPACE         POD                             CONTAINER        PID        COMM              READS  WRITES R_KB       W_KB       T FILE
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             1        0        0        0          R libssl.so.1.1
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             2        0        2        0          R etc-hosts
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             1        0        1        0          R passwd
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             2        0        0        0          R resolv.conf
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             1        0        0        0          R libcurl.so.4.8.0
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             1        0        0        0          R libnghttp2.so.14.21.2
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  bash             2        0        0        0          R ld-musl-x86_64.so.1
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             1        0        0        0          R libbrotlidec.so.1.0.9
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  bash             3        0        0        0          R curl
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             12       0        12       0          R openssl.cnf
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             1        0        0        0          R libbrotlicommon.so.1.0.
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             211      0        211      0          R ca-certificates.crt
aks-diskio-48877447-vmss000003 default          aksday-diskio-785c897d97-jl98r crashapp         509064  curl             1        0        0        0          R libcrypto.so.1.1
```

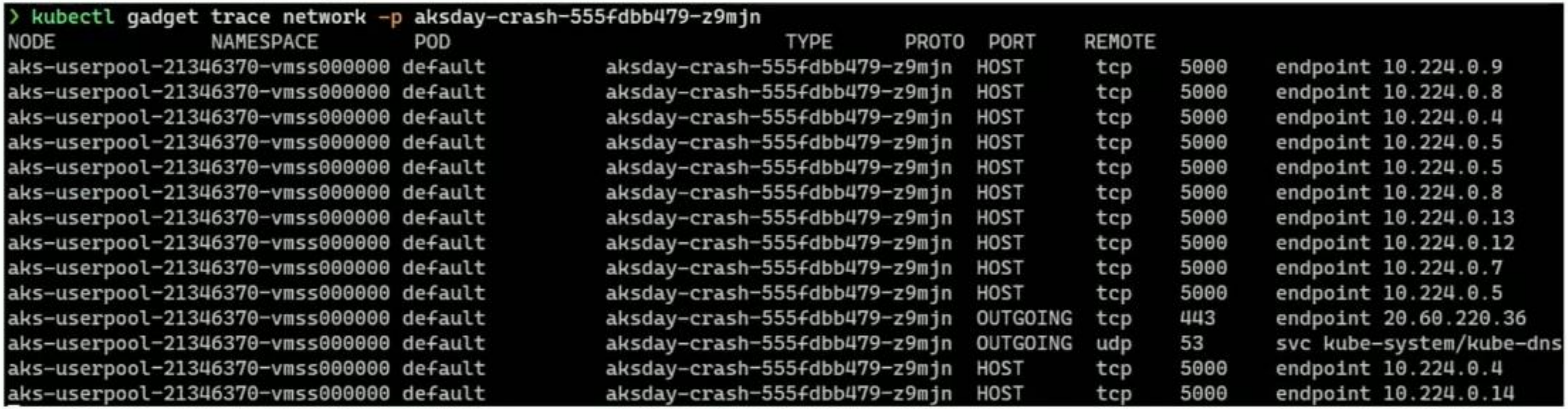
Inspektor Gadget Trace

- kubectl gadget trace sni



```
❯ kubectl gadget trace sni -n default -p aksday-crash-555fdbb479-z9mjn
NODE             NAMESPACE        POD                            NAME
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  pahealyaksday.blob.core.windows.net
```

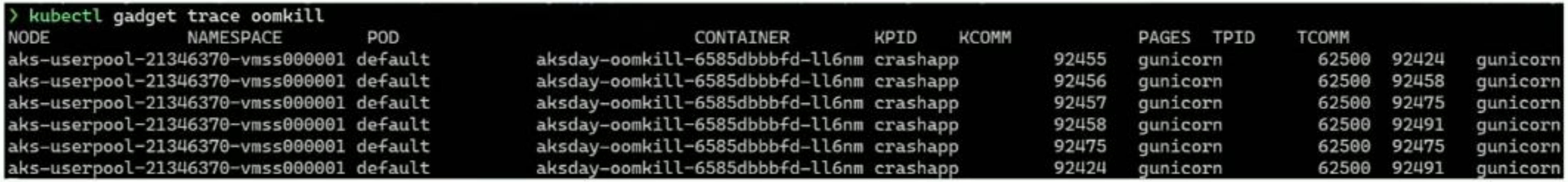
- kubectl gadget trace network



```
❯ kubectl gadget trace network -p aksday-crash-555fdbb479-z9mjn
NODE             NAMESPACE        POD                            TYPE      PROTO  PORT    REMOTE
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.9
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.8
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.4
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.5
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.5
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.8
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.13
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.12
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.7
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.5
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  OUTGOING  tcp    443     endpoint 20.60.220.36
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  OUTGOING  udp    53      svc kube-system/kube-dns
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.4
aks-userpool-21346370-vmss000000 default          aksday-crash-555fdbb479-z9mjn  HOST      tcp    5000    endpoint 10.224.0.14
```

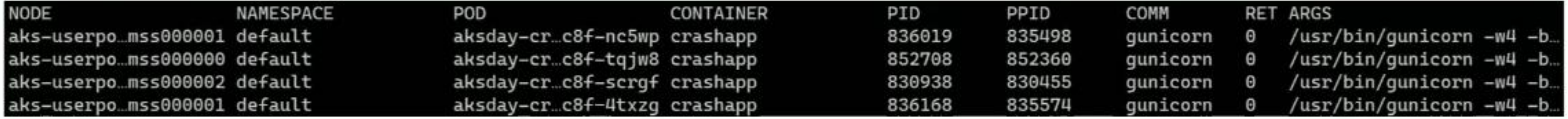
Inspektor Gadget Trace

- kubectl gadget trace oomkill



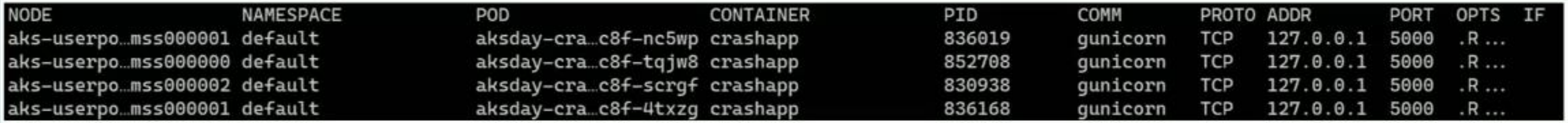
```
> kubectl gadget trace oomkill
NODE             NAMESPACE        POD                           CONTAINER        KPID    KCOMM            PAGES  TPID    TCOMM
aks-userpool-21346370-vmss000001 default           aksday-oomkill-6585dbbbfd-ll6nm crashapp         92455   gunicorn         62500  92424   gunicorn
aks-userpool-21346370-vmss000001 default           aksday-oomkill-6585dbbbfd-ll6nm crashapp         92456   gunicorn         62500  92458   gunicorn
aks-userpool-21346370-vmss000001 default           aksday-oomkill-6585dbbbfd-ll6nm crashapp         92457   gunicorn         62500  92475   gunicorn
aks-userpool-21346370-vmss000001 default           aksday-oomkill-6585dbbbfd-ll6nm crashapp         92458   gunicorn         62500  92491   gunicorn
aks-userpool-21346370-vmss000001 default           aksday-oomkill-6585dbbbfd-ll6nm crashapp         92475   gunicorn         62500  92475   gunicorn
aks-userpool-21346370-vmss000001 default           aksday-oomkill-6585dbbbfd-ll6nm crashapp         92424   gunicorn         62500  92491   gunicorn
```

- kubectl gadget trace exec



```
NODE                 NAMESPACE          POD                 CONTAINER         PID       PPID      COMM      RET ARGS
aks-userpo…mss000001 default            aksday-cr…c8f-nc5wp crashapp          836019    835498    gunicorn  0   /usr/bin/gunicorn -w4 -b…
aks-userpo…mss000000 default            aksday-cr…c8f-tqjw8 crashapp          852708    852360    gunicorn  0   /usr/bin/gunicorn -w4 -b…
aks-userpo…mss000002 default            aksday-cr…c8f-scrgf crashapp          830938    830455    gunicorn  0   /usr/bin/gunicorn -w4 -b…
aks-userpo…mss000001 default            aksday-cr…c8f-4txzg crashapp          836168    835574    gunicorn  0   /usr/bin/gunicorn -w4 -b…
```

- kubectl gadget trace bind



```
NODE                 NAMESPACE           POD                  CONTAINER           PID         COMM       PROTO ADDR       PORT  OPTS  IF
aks-userpo…mss000001 default             aksday-cra…c8f-nc5wp crashapp            836019      gunicorn   TCP   127.0.0.1  5000  .R...
aks-userpo…mss000000 default             aksday-cra…c8f-tqjw8 crashapp            852708      gunicorn   TCP   127.0.0.1  5000  .R...
aks-userpo…mss000002 default             aksday-cra…c8f-scrgf crashapp            830938      gunicorn   TCP   127.0.0.1  5000  .R...
aks-userpo…mss000001 default             aksday-cra…c8f-4txzg crashapp            836168      gunicorn   TCP   127.0.0.1  5000  .R...
```

https://github.com/Maheshk-MSFT/

maheshk@microsoft.com