



Streamlining AKS Debugging: Techniques to solve common & complex problems

Aritra Ghosh, Jose Blanquicet, Juan-Lee Pang





Streamlining AKS Debugging: Techniques to solve common & complex problems

Aritra Ghosh, Jose Blanquicet, Juan-Lee Pang

Let's jump right in
with a 2-question poll!

aka.ms/buildaksdebug



Agenda



Introductions



Create, Read,
Update, Delete



Opening



Best practices and
Announcements



Troubleshooting
Network Issues



Q&A



Node Health &
Resource Utilization

Introductions



Aritra Ghosh

Senior Product Manager

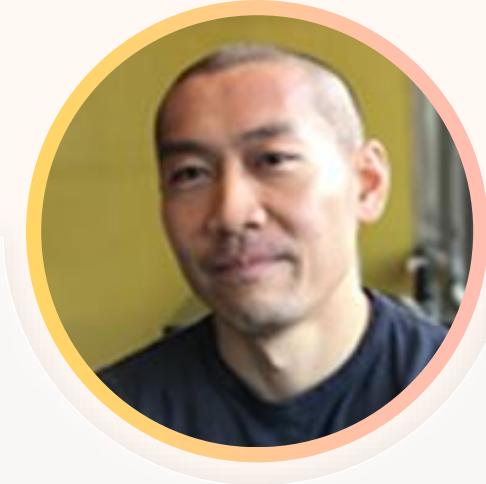
AKS Observability



Jose Blanquicet

Senior Software Engineer

AzCore Linux

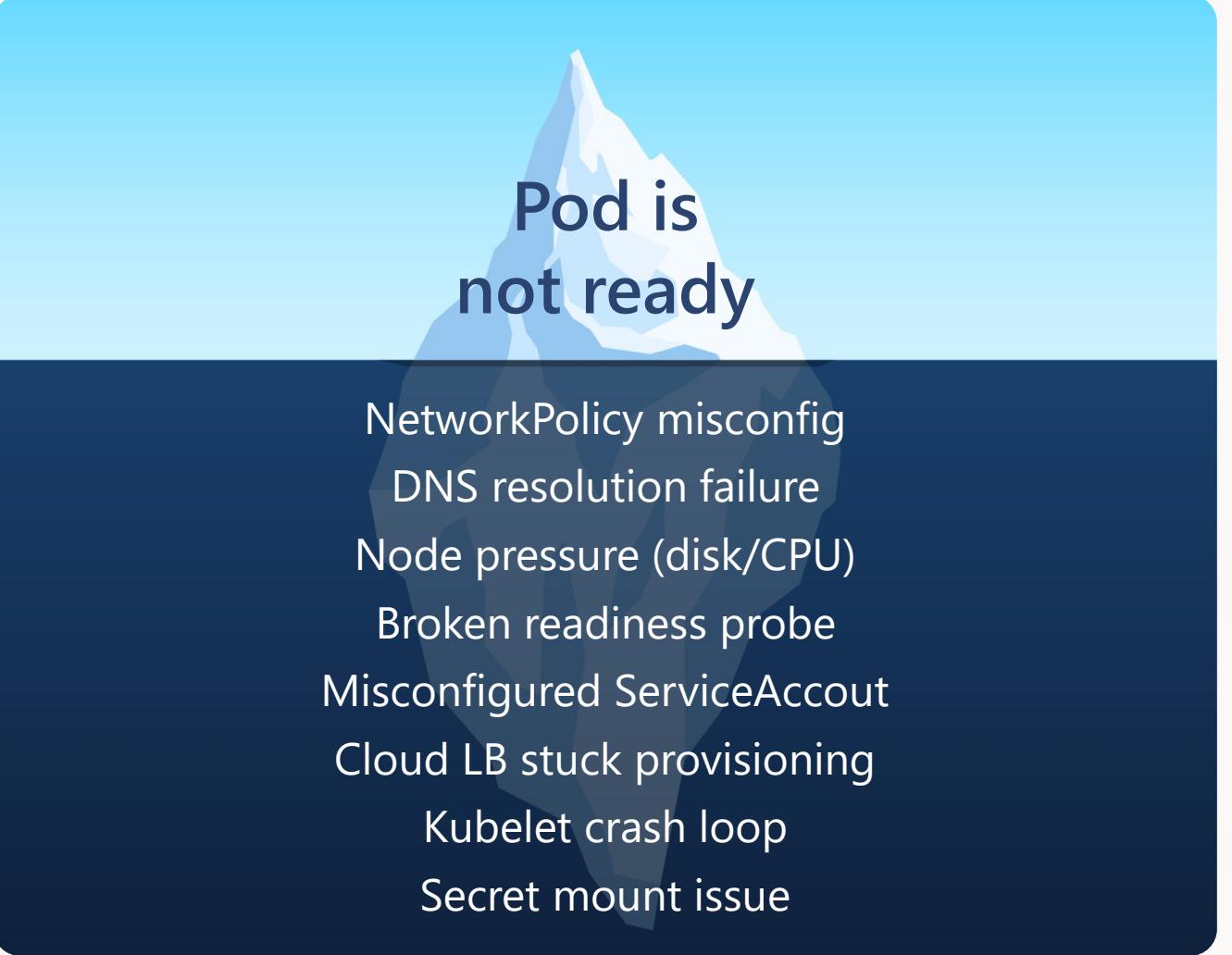


Juan-Lee Pang

Principal SWE Manager

AKS Observability

Kubernetes troubleshooting is easy!

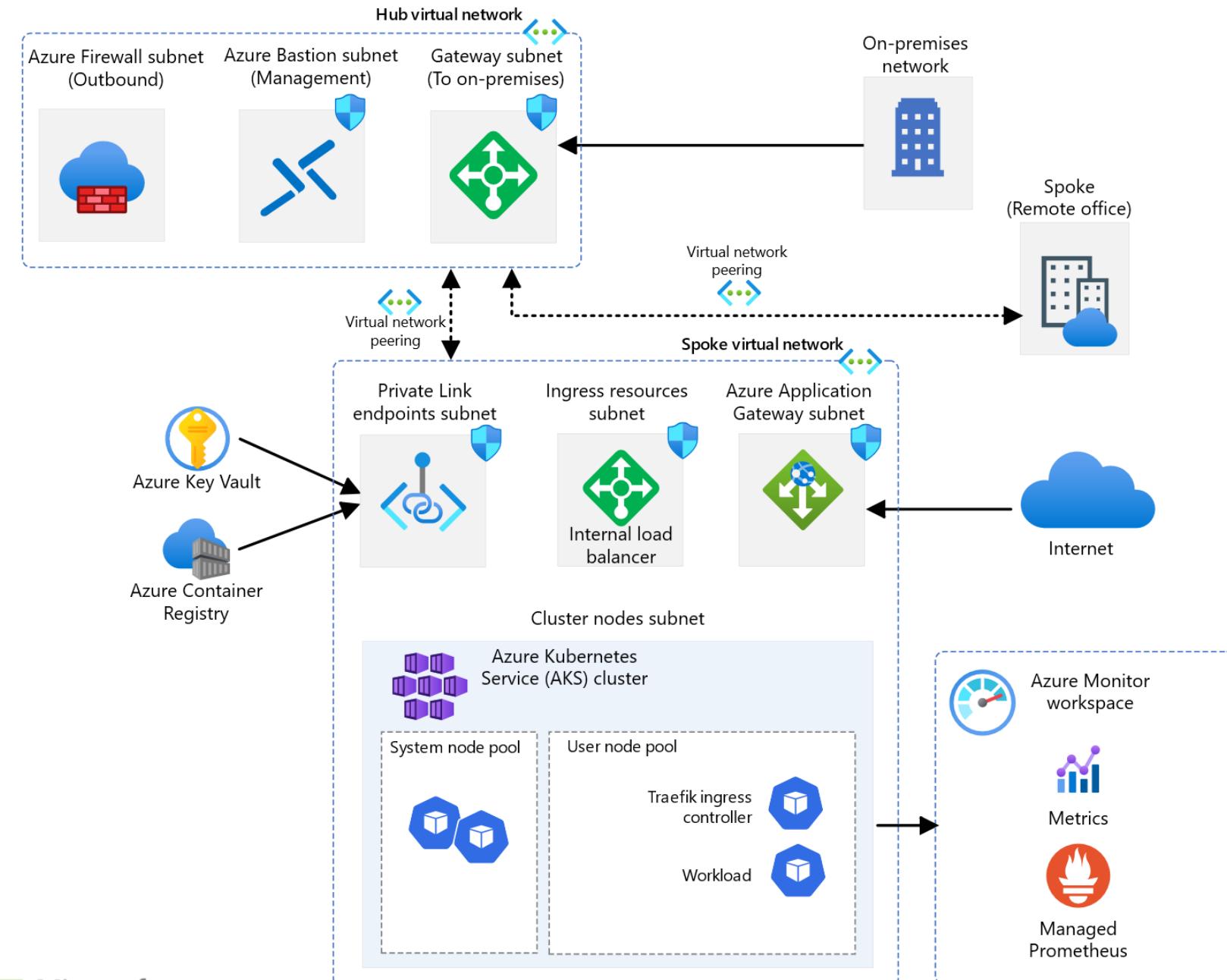


A large iceberg is visible above the waterline, with a much larger portion submerged below. This imagery serves as a metaphor for Kubernetes troubleshooting, where many issues are often hidden beneath the surface.

Pod is
not ready

- NetworkPolicy misconfig
- DNS resolution failure
- Node pressure (disk/CPU)
- Broken readiness probe
- Misconfigured ServiceAccount
- Cloud LB stuck provisioning
- Kubelet crash loop
- Secret mount issue

Navigating the journey of troubleshooting Kubernetes



Scenario 1: Troubleshooting Network Issues



[Meme Creator](#)

Focus on DNS and TCP

What happens during a typical network connection?

DNS resolution (translating domain name to IP addresses)



TCP/UDP transport setup



Application-layer communication (e.g., HTTPs, gRPC)

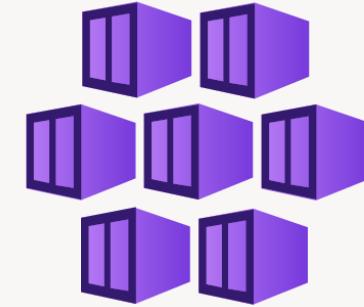


Debugging Tools



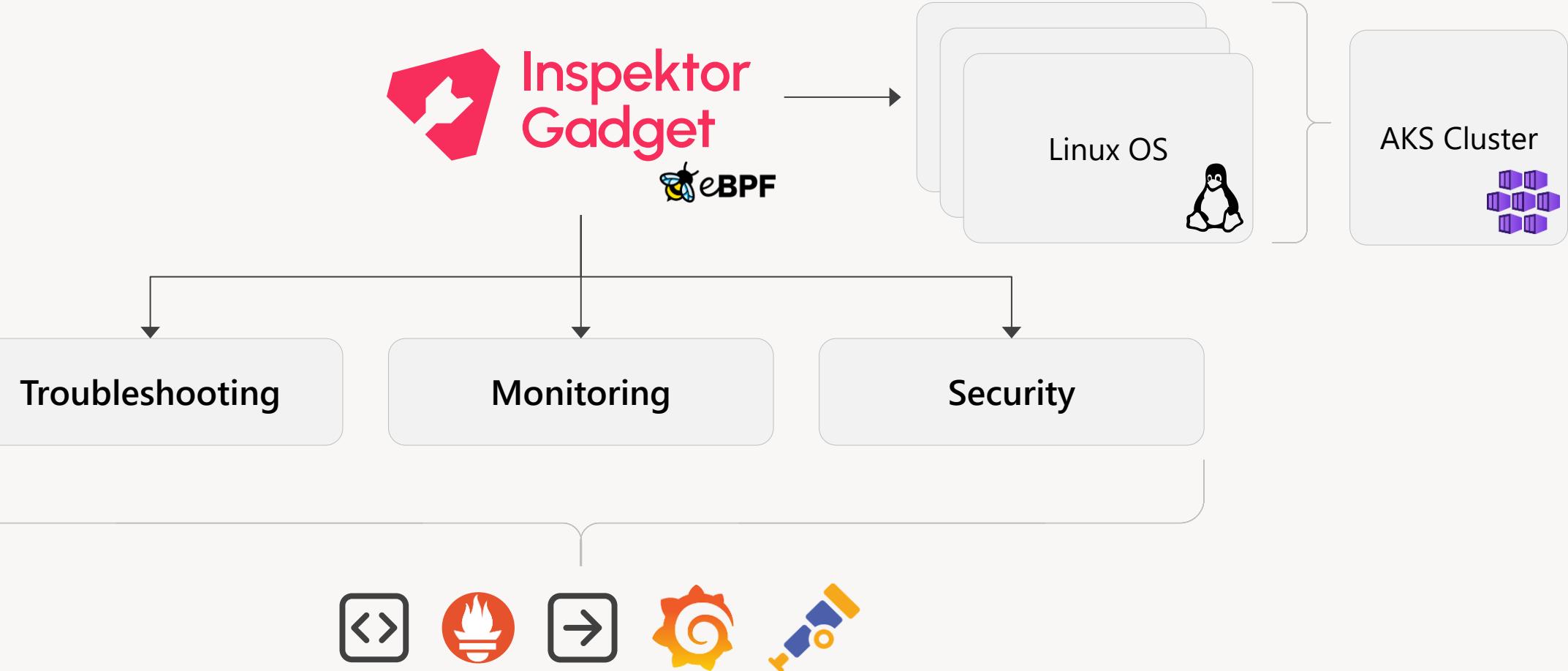
**Inspektor
Gadget**

*Real-Time debugging
with Inspektor Gadget*

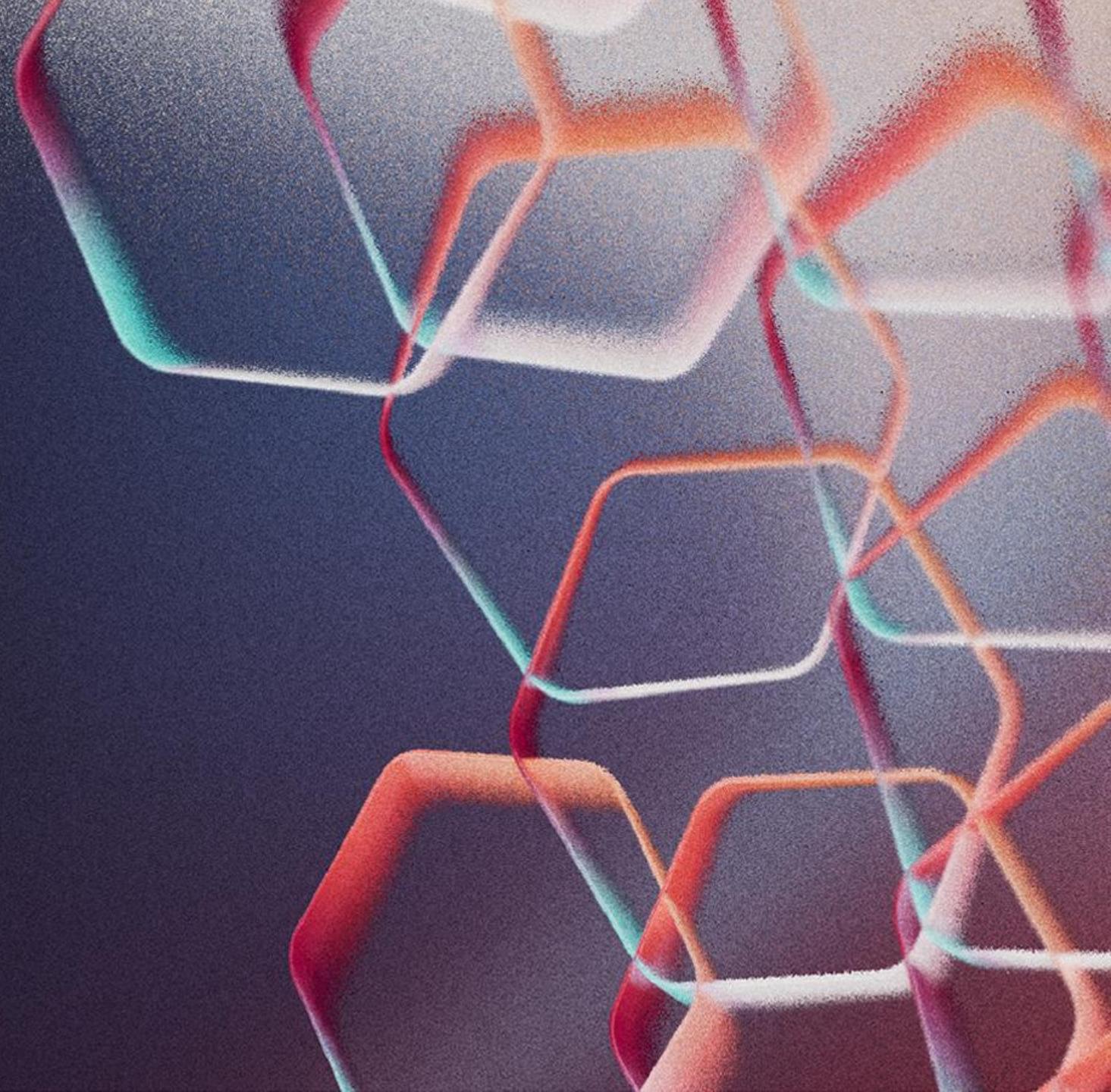


*Kubernetes
Status, Logs, etc.*

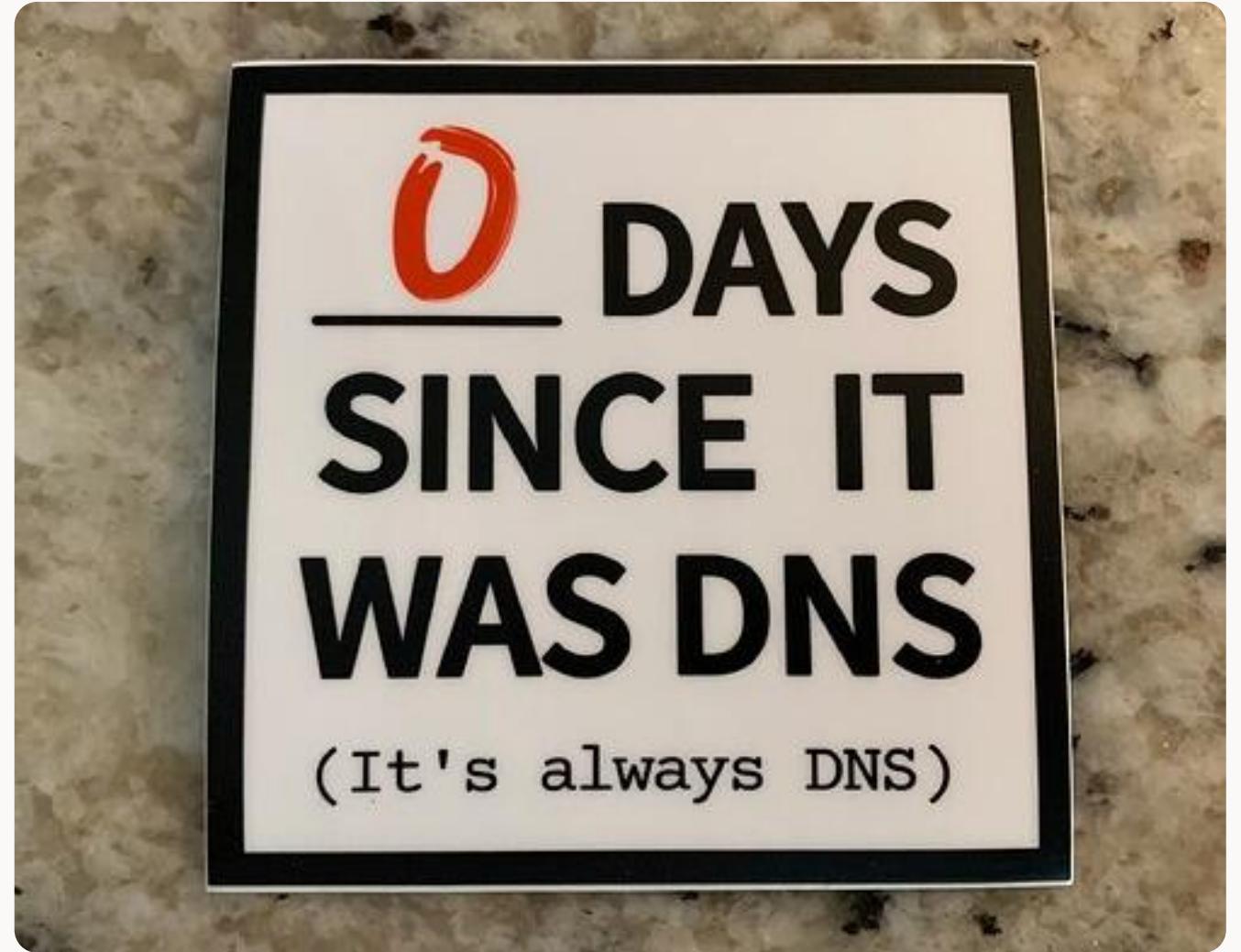
Inspektor Gadget



Demo

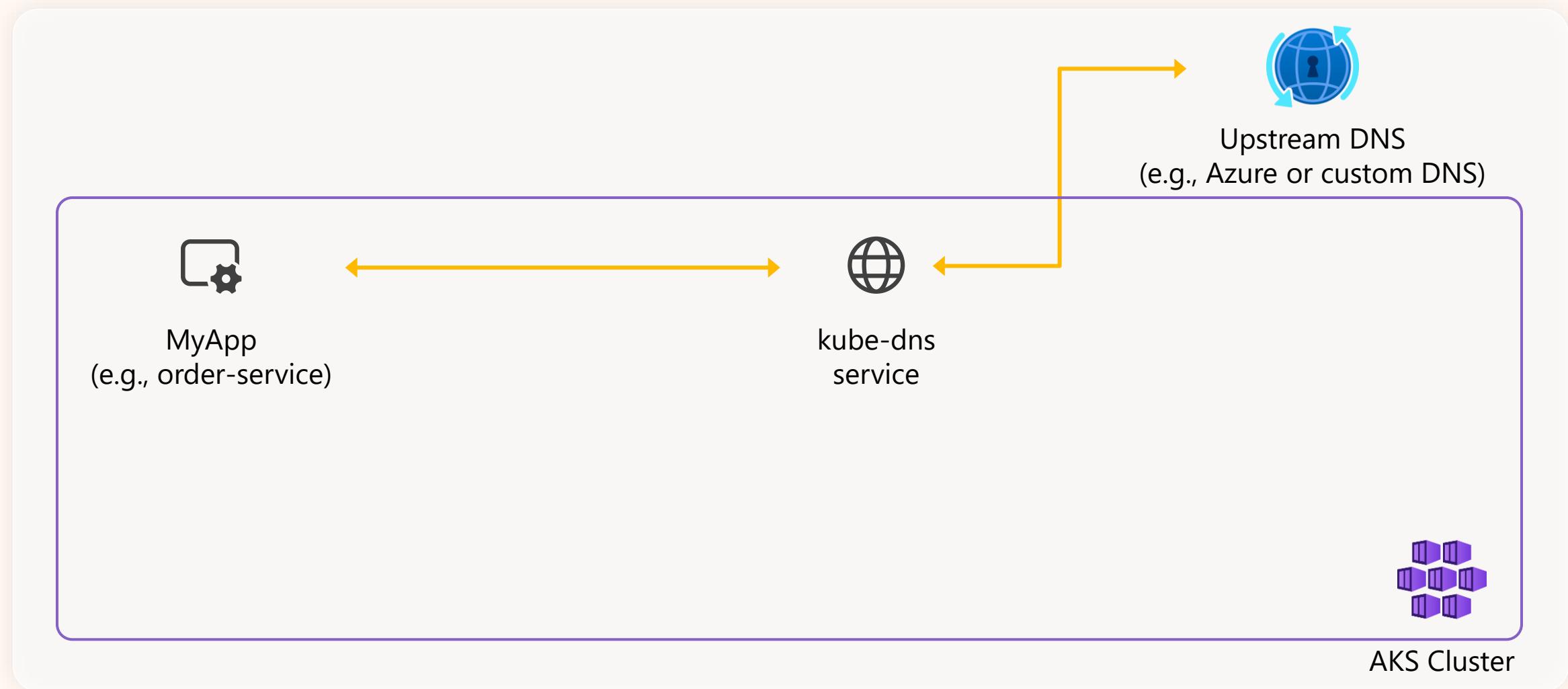


Troubleshooting DNS in AKS

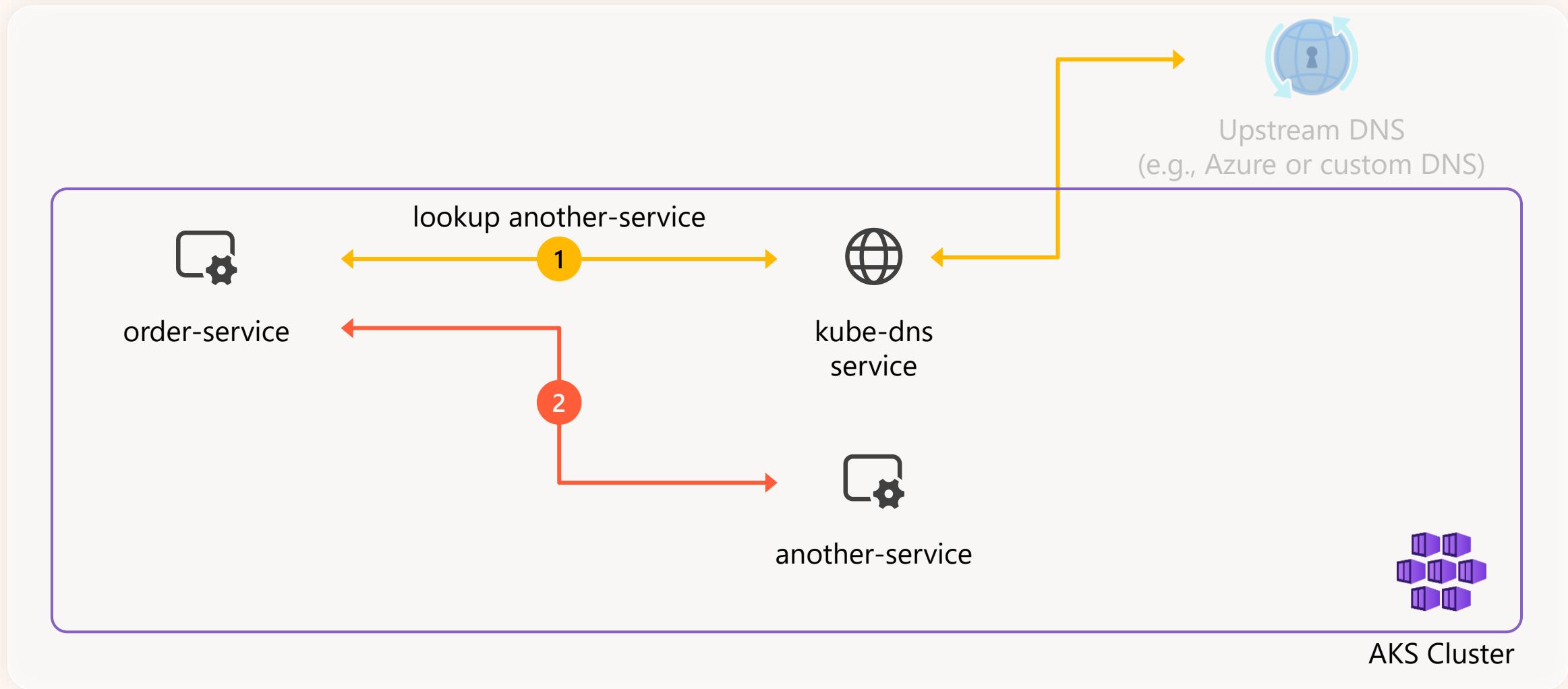


[Its always DNS lol – Humor – It's FOSS Community](#)

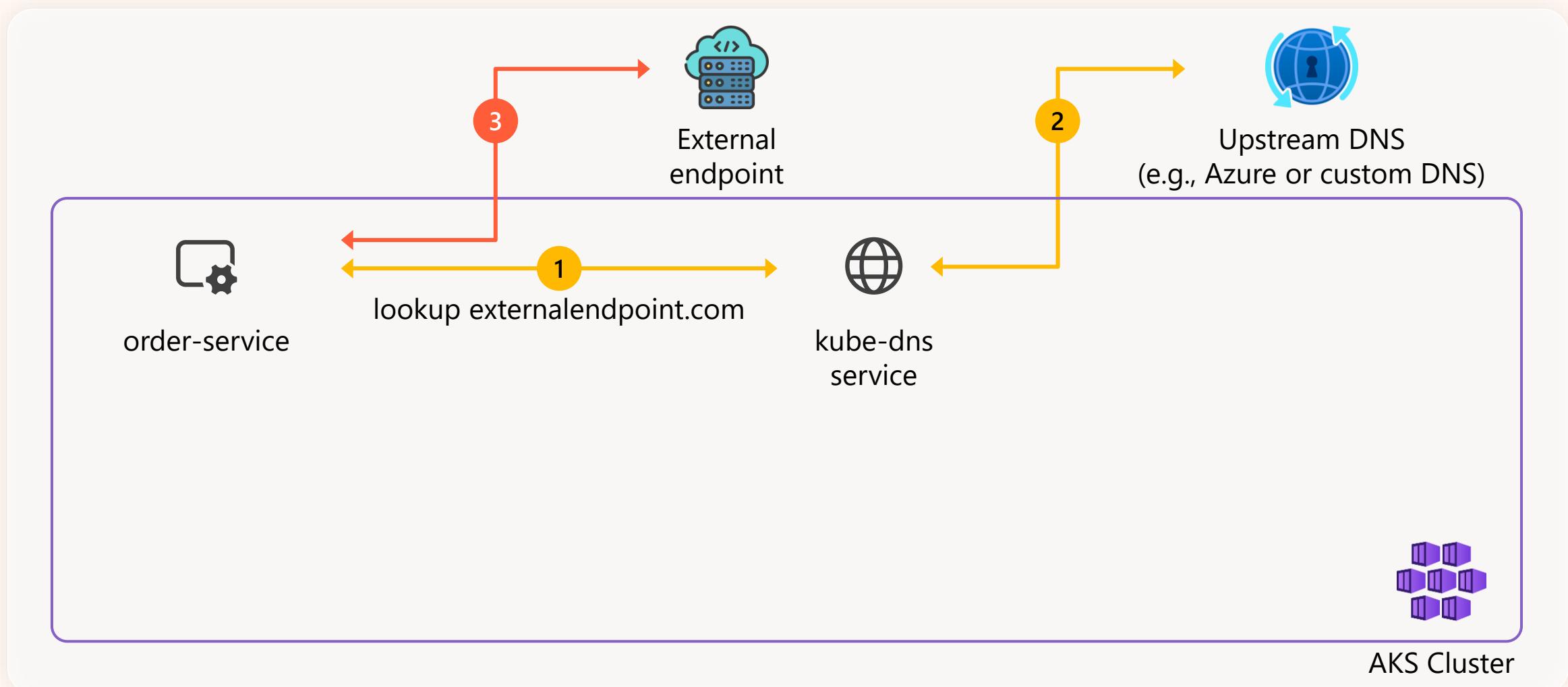
Context: Components involved in a DNS resolution



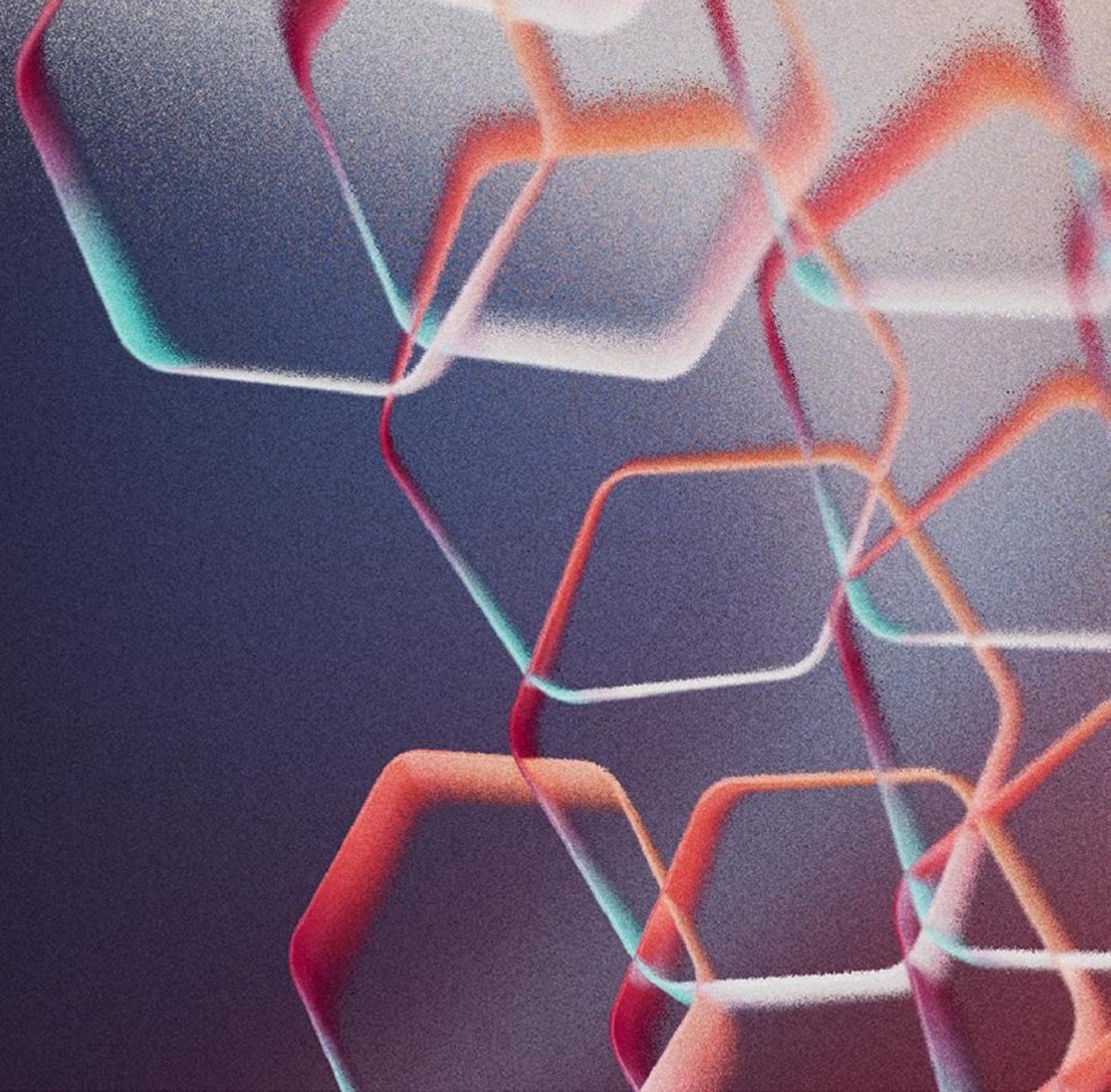
DNS: Local endpoints



DNS: External endpoints



Demo



Takeaways



Isolate

- Using an inside-out approach streamlines troubleshooting
- Start from one app and then see how it interacts with other etc.



Tools & data

- Logs – valuable source of information to start the analysis with
- Inspektor Gadget
 - Kubernetes aware
 - Keep manifest files for common troubleshooting scenarios
 - Interactive & multipurpose (security, networking, performance, etc.)



Best practices

- For those few static external names, use fully qualified domain names (FQDN) to avoid generating unnecessary DNS traffic



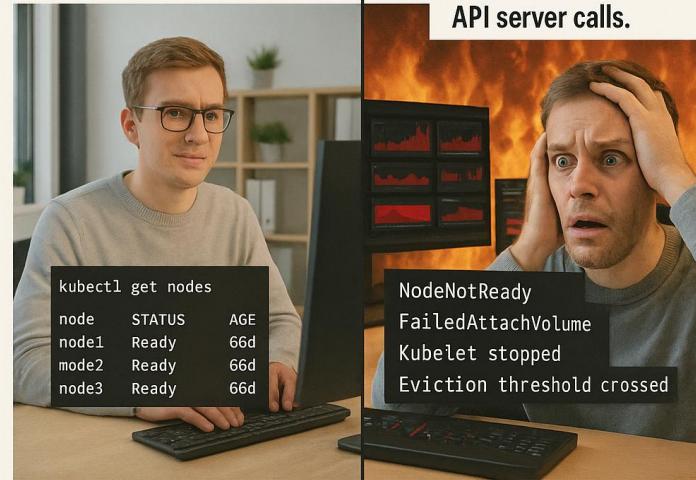
And that's not just what IG does... inspektor-gadget.io

Scenario 2: Node Health

TROUBLESHOOTING NODE HEALTH IN KUBERNETES

Should be a
quick check...

Turns out kubelet died,
disk pressure triggered eviction,
and network policy blocked
API server calls.



TURNS OUT KUBELET died,
disk pressure triggered eviction,
and network policy blocked
API server calls.

Scenario 2: Node Health Diagnosis



Node conditions



Kubernetes Events



Resource Metrics

Scenario 2: Node Conditions



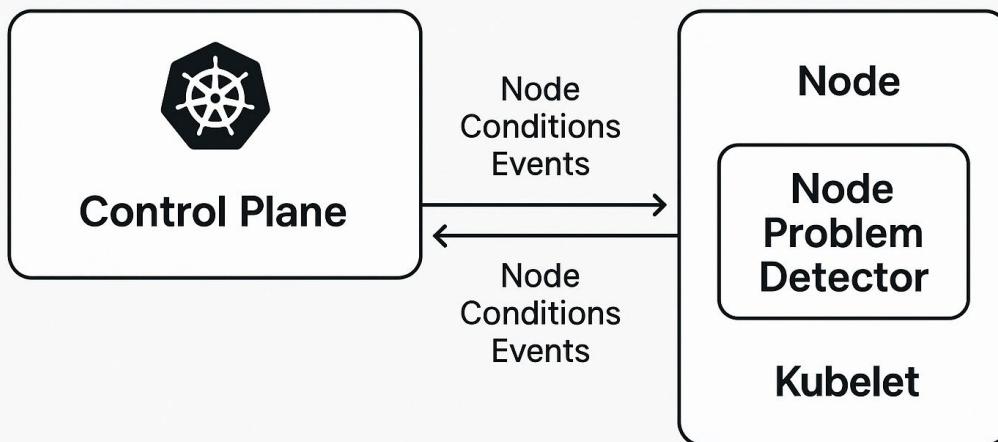
AKS installs Node Problem Detector (NPD) on nodes to monitor health



NodeNotReadyDiagnostics condition provides node readiness details

NPD in AKS

Node Problem Detector



Node Conditions in Azure Portal

Type	Status	Last probe time
ReadonlyFilesystem	False	2025-05-15T19:51:07Z
ContainerRuntimeProbl...	False	2025-05-15T19:51:07Z
KernelDeadlock	False	2025-05-15T19:51:07Z
VMEventScheduled	False	2025-05-15T19:51:07Z
KubeletProblem	False	2025-05-15T19:51:07Z
FrequentDockerRestart	False	2025-05-15T19:51:07Z
FrequentUnregisterNet...	False	2025-05-15T19:51:07Z
FrequentContainerdRest...	False	2025-05-15T19:51:07Z
FilesystemCorruptionPr...	False	2025-05-15T19:51:07Z
FrequentKubeletRestart	False	2025-05-15T19:51:07Z
PIDPressure	False	2025-05-15T19:49:25Z

Scenario 2: Node Conditions

Common Root Causes and How to Investigate

Kubectl describe node <>

Cause	How to Check	Possible Fix
 Kubelet Crash/Hang	Describe node, journalctl -u kubelet (if SSH access)	Restart kubelet (manually or via VM restart)
 Disk Pressure	Describe node > Conditions + df -h (SSH)	Clean up disk (logs, unused images)
 OOM/Memory Pressure	MemoryPressure condition	Evict pods/increase VM size, Check for memory leaks
 Container Runtime Down	Crictl info or systemctl statuscontainerd/docker	Restart container runtime

Scenario 2: Node Events

Events provide additional insight into the health of the node

kubectl get events – Field-selector involvedObject.kind = Node, involvedObject.name= <>

Message	Type	Reason	Source	Object	Namespace	Count
⚠ kubelet healthcheck failed: curl: (28) Operation timed out after 1000	Warning	KubeletIsDown	kubelet-custom-plugin-monitor, ...	Node/aks-testnodepool-13033019-vmss0...	default	2
✓ Node aks-testnodepool-13033019-vmss000000 status is now: Node	Normal	NodeNotReady	node-controller	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ Node condition KubeletProblem is now: True, reason: KubeletIsDow	Warning	KubeletIsDown	kubelet-custom-plugin-monitor, ...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ crictl -t 60s pods --latest failed!	Warning	ContainerRunti...	container-runtime-custom-plugi...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ kubelet healthcheck failed: curl: (28) Operation timed out after 1000	Warning	KubeletIsDown	kubelet-custom-plugin-monitor, ...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task kworker/1:3:786744 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815743 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815747 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ Node condition ContainerRuntimeProblem is now: True, reason: Cor	Warning	ContainerRunti...	container-runtime-custom-plugi...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815750 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815752 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815754 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815756 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
✓ Container image "ubuntu:20.04" already present on machine	Normal	Pulled	kubelet, aks-testnodepool-13033...	Pod/brutal-stress	default	1

Scenario 2: Node Metrics



Basic metrics: Memory, CPU, I/O, and Networking

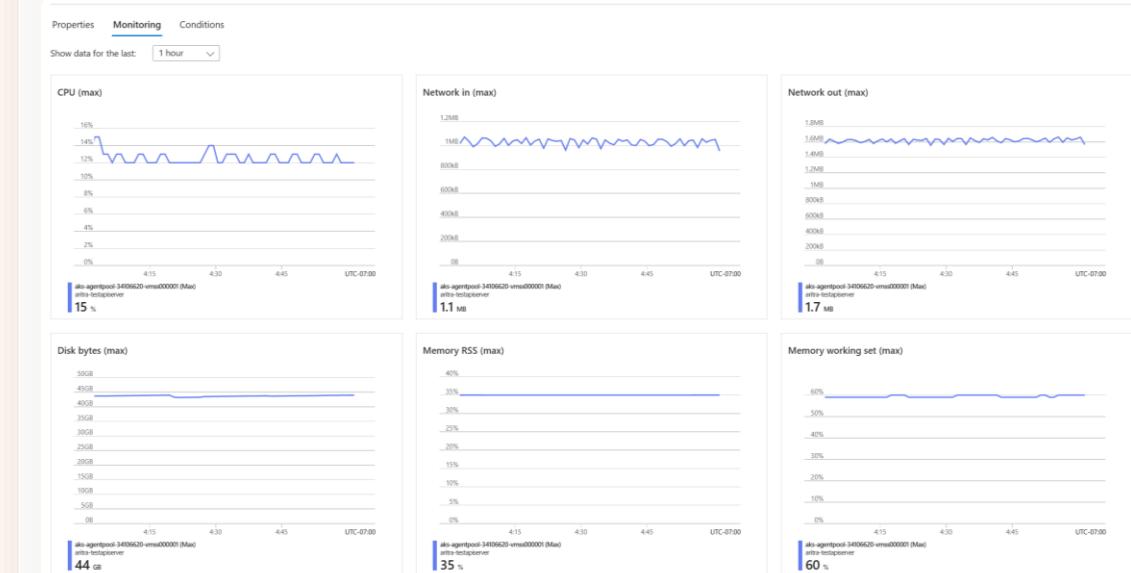


Node exporter and *Cadvisor* provide node metrics that can be collected using your preferred tools

Azure Grafana Node Dashboard



Node Views in Azure Portal



Scenario 2: Node Metrics

What if Azure Managed Grafana is not an option?



Azure Monitor dashboards with Grafana



No additional fees apply



You can utilize all standard dashboards, which can also be customized to meet your requirements



For more details, make sure to attend BRK188

Scenario 2: Azure Monitor dashboards with Grafana

Public Preview

Home > aritra-controlplanetrics-test

aritra-controlplanetrics-test | Dashboards with Grafana (preview) | Kubernetes | Compute Resources | Node (Pods)

Kubernetes service | Azure Monitor

Search Dashboards Edit Save As Share Export Help Feedback

Data source DefaultAzureMonitorWorkspace-centralus cluster aritra-controlplanetrics-test node aks-agentpool-62087549-vmss00000000 Last 1 hour Refresh 1m

CPU Usage

2025-05-07 14:52:00

Name	Last *
max capacity	8
ama-logs-rs-5795b89f4-t47qm	0.00680
ama-logs-z6qnf	0.0121
ama-metrics-854765d6d6-gvscv	0.00556
ama-metrics-ksn-b699d5fc-b75cc	0.00128
ama-metrics-node-x7qpn	0.00952

CPU Quota

Pod	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
csi-azurefile-node-6q84s	0.000167	0.0300	0.557%		
coredns-autoscaler-777f7c56d8-689hn	0.000167	0.0200	0.834%	0.200	0.0834%
retina-agent-rdr25	0.000120	0.100	0.120%	0.500	0.0241%
ama-metrics-854765d6d6-gvscv	0.00527	0.170	3.10%	7.50	0.0702%

Memory Usage (w/o cache)

Name	Value
azuredisknode-5f8f5c78c8-tcvrz	32.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	28.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	24.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	20.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	16.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	12.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	8.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	4.0 MiB
azuredisknode-5f8f5c78c8-tcvrz	0.8 MiB

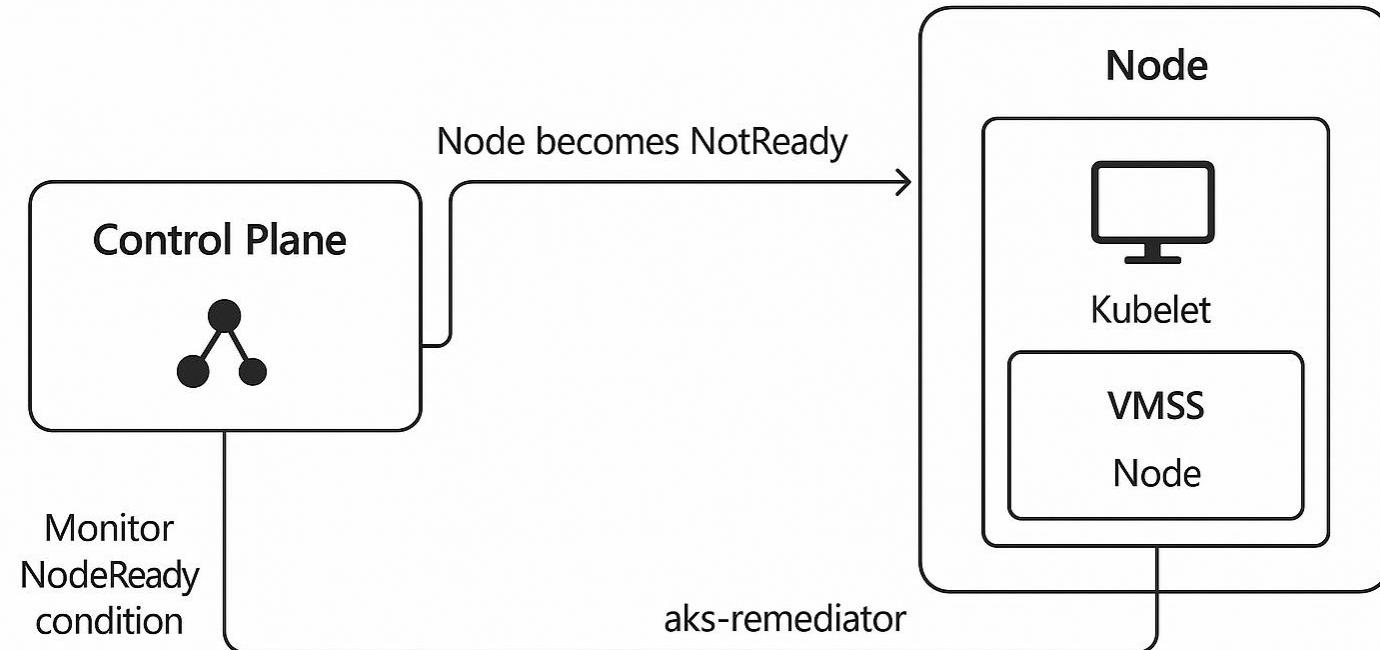
Memory Quota

Pod	Memory Usage	Memory Requests	Memory Requests %	Memory Limits	Memory Limits %	Memory Usage (RSS)	Memory Usage (Cache)	Memory Usage (Swap)
azure-ip-masq-agent-55ttx	17.5 MiB	50 MiB	34.9%	250 MiB	6.99%	12.2 MiB	24.2 MiB	0 B
ama-logs-z6qnf	295 MiB	600 MiB	49.2%	2.10 GiB	13.8%	271 MiB	7.76 MiB	0 B
metrics-server-5f8f5c78c8-tcvrz	59.7 MiB	154 MiB	38.8%	424 MiB	14.1%	55.6 MiB	45.1 MiB	0 B
csi-azuredisk-node-kf9tk	34.4 MiB	60 MiB	57.4%	2.73 GiB	1.23%	32.1 MiB	15.4 MiB	0 B

Dashboards with Grafana (preview)

Scenario 2: Node Auto Repair

- AKS automatically repairs nodes that remain unresponsive for over 10 minutes
- If the issue persists, we attempt a reboot, reimage, and redeploy in succession
- Should the node continue to show as not ready, a deeper investigation is necessary to resolve the issue



Auto-repair sequence

- Reboot the node
- Reimage the node
- Redeploy the node

Scenario 2: Node Health Deep Dive



It might be necessary to delve further to identify the root cause



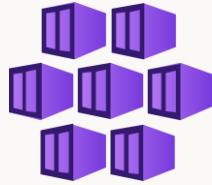
We are working on providing metrics at the process level within the Diagnose and Solve section



In uncommon situations, kubelet logs could offer insights into the failure

Scenario 2: Node Health Demo

Case: Developers report an elevated error rate on one of the apps in production.



Azure Prometheus + Grafana + AKS



Azure
Monitor



Application
Insights

Scenario 2: Node Health Demo

The screenshot shows the Microsoft Azure Application Insights Overview page for the resource group 'appinsightstest'. The page includes a top navigation bar with tabs for 'appinsightstest - Microsoft', 'aritra-controlplanemetrics', 'Kubernetes / Compute Re...', 'Kubernetes / Compute Re...', and '52.242.211.172'. Below the navigation is a search bar and a Copilot button. The main content area has a blue header with 'Microsoft Azure (Preview)' and a search bar. A sidebar on the left lists navigation options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Resource visualizer', 'Investigate', 'Application map', 'Smart detection', 'Live metrics', 'Transaction search', and 'Availability'. The 'Overview' section displays details such as Resource group (move) to 'aritra-test', Location 'East US', Subscription (move) to 'aritraghosh-subscription', Subscription ID '6107f264-6d8e-4c58-bb3f-c178d3cf23e8', Tags (edit), and Add tags. It also shows 'Show data for last:' with a 1 hour selection. Two cards at the bottom are 'Failed requests' and 'Server response time'.

appinsightstest - Microsoft | aritra-controlplanemetrics | Kubernetes / Compute Re... | Kubernetes / Compute Re... | 52.242.211.172

ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/6107f264-6d8e-4c58-bb3f-c178d3cf23e8/resourceGroups/appinsightstest

Microsoft Azure (Preview)

Search resources, services, and docs (G+)

Copilot

Home >

appinsightstest

Application Insights

Search

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Investigate

Application map

Smart detection

Live metrics

Transaction search

Availability

Overview

Application Dashboard

Getting started

Search

Logs

Monitor resource group

Feedback

Favorites

Rename

Delete

View Cost

JSON View

Resource group (move)
aritra-test

Location
East US

Subscription (move)
aritraghosh-subscription

Subscription ID
6107f264-6d8e-4c58-bb3f-c178d3cf23e8

Tags (edit)
Add tags

Show data for last:

30 minutes 1 hour 6 hours 12 hours 1 day 3 days 7 days 30 days

Failed requests

Server response time

Instrumentation Key
9dbf9a5c-05bf-440a-b678-9d1142c6aad9

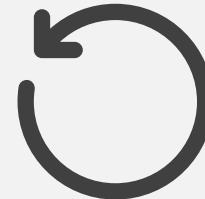
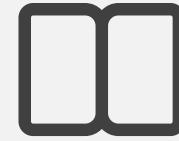
Connection String
InstrumentationKey=9dbf9a5c-05bf-440a-b678-9d1142c6aad9;IngestionE...

Workspace
6107f264-6d8e-4c58-bb3f-c178d3cf23e8-aritra-test-EUS

Add or remove favorites by pressing Ctrl + Shift + F

Scenario 3:

Create, Read, Update, Delete (CRUD)



Scenario 3: Failed Cluster/Nodepool

- Last Operation status is now available in Azure Portal and AKS API
- Has details of the exact root cause and Troubleshooting guide links

Last Operation Status

X



Reconcile



Abort operation



Refresh



Drain node aks-agentpool-21168581-vmss00000q failed when evicting pod nginx-deployment-86dcfdf4c6-csbbg. Evict blocked by conflicting disruption budgets. See <http://aka.ms/aks/debugdrainfailures>. Original error: This pod has more than one PodDisruptionBudget, which the eviction subresource does not support. PDB debug info: default/nginx-deployment-86dcfdf4c6-csbbg

Details

Operation status

Failed

Scenario 3: Failed Cluster/nodepool

- Activity logs typically include details about operations that failed
- In the case of upgrades, Azure Resource Graph (ARG) maintains a record of auto upgrades

Create or Update Agent Pool

Tue May 13 2025 14:06:55 GMT-0700 (Pacific Daylight Time)

[+ New alert rule](#) [+ New support request](#)

[Summary](#)

[JSON](#)

Change history (Change history is not applicable to this type of activity log)

Resource

Operation name

Create or Update Agent Pool

Time stamp

Tue May 13 2025 14:06:55 GMT-0700 (Pacific Daylight Time)

Event initiated by

Error code

ResourceOperationFailure

Message

The resource operation completed with terminal provisioning state 'Failed'.

Scenario 3: CRUD demo

II

I tried scaling one of my nodepools on the cluster and it failed. What do I do now and how can I fix it?

Tools used

1

Azure
Portal

2

Azure Resource
Graph

Scenario 3: CRUD demo

Setup: My cluster is in failed state.

How do I know what could have caused it and how do I fix it ?

The screenshot shows the Microsoft Azure (Preview) portal interface. The top navigation bar has tabs for 'nodenotreadytest - Microsoft A', 'Azure Resource Graph Explorer', 'nodenotreadytest - Microsoft A', and 'AKS Communication Manager'. Below the bar, the URL is ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/6107f264-6d8e-4c58-bb3f-c178d3cf23e8/resourceGrou... . The main header says 'Microsoft Azure (Preview)' and 'Search resources, services, and docs (G+/-)'. On the right, there's a 'Copilot' button and a user profile icon. The page title is 'Home > Kubernetes services > nodenotreadytest'. The left sidebar has links for 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Monitor', 'Diagnose and solve problems', 'Microsoft Defender for Cloud (preview)', 'Cost analysis', 'Resource visualizer', and 'Kubernetes resources'. The 'Overview' tab is selected. The 'Essentials' section displays the following information:

Resource group	Kubernetes version
agrg	1.31.7
Power state	API server address
Running	nodenotreadytest-dns-fy504gt.hcp.eastus2.azmk8s.io
Cluster operation status	Network configuration
Succeeded	Azure CNI Overlay
Subscription	Node pools
aritraghosh-subscription	2 node pools - 1 failed
Location	Container registries
East US 2	Attach a registry

At the bottom of the sidebar, it says 'Add or remove favorites by pressing Ctrl+Shift+F'.

Wouldn't it be great to avoid these issues...

Node Sizing

- Right size nodes
- Use taints/affinities to isolate system vs user workloads

Cluster Operations

- Review PDBs prior to upgrading
- Confirm that you have adequate quota and IP addresses

Network Resilience

- Avoid blocking control plane traffic via NSGs or misconfigured network policies
- Ensure DNS and Azure API endpoints are reachable

Monitoring

- Set alerts for things like failed auto upgrades, MemoryPressure, DiskPressure, and NodeReady state changes

Azure Monitor Prometheus Recommended alerts for AKS

General Availability

Learn More: [Aka.ms/prometheus-community-alerts](https://aka.ms/prometheus-community-alerts)

Thank you



Table Talk

- Chat with us further at the **Table Talk**
- May 21st, 2 pm PT
- [COMM452](#) - Troubleshooting AKS, Cost Optimization & AI in K8s



Upcoming Sessions

- [LAB345-R1](#) – Deploying and Inferencing AI Applications on K8s
 - May 21st, 2:45 pm PT
- [LAB342-R1](#) – Streamlining K8s for developers with AKS Automatic
 - May 22nd, 8:30 am PT



Collaborate on OSS Troubleshooting Tooling

- [node-problem-detector](#)
- [Inspektor-gadget.io](#)

Q&A

Appendix

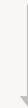
Kubernetes troubleshooting is easy!

What my PM thinks Kubernetes is...

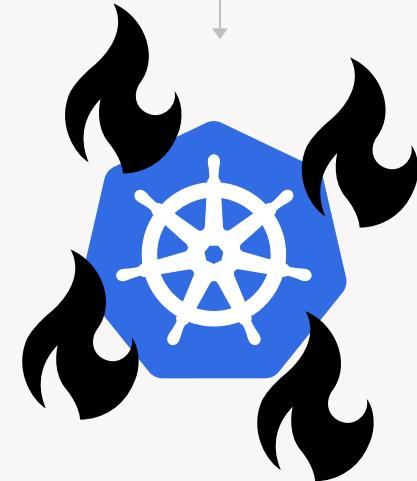


What I know Kubernetes is...

Misconfigured Resource Limits

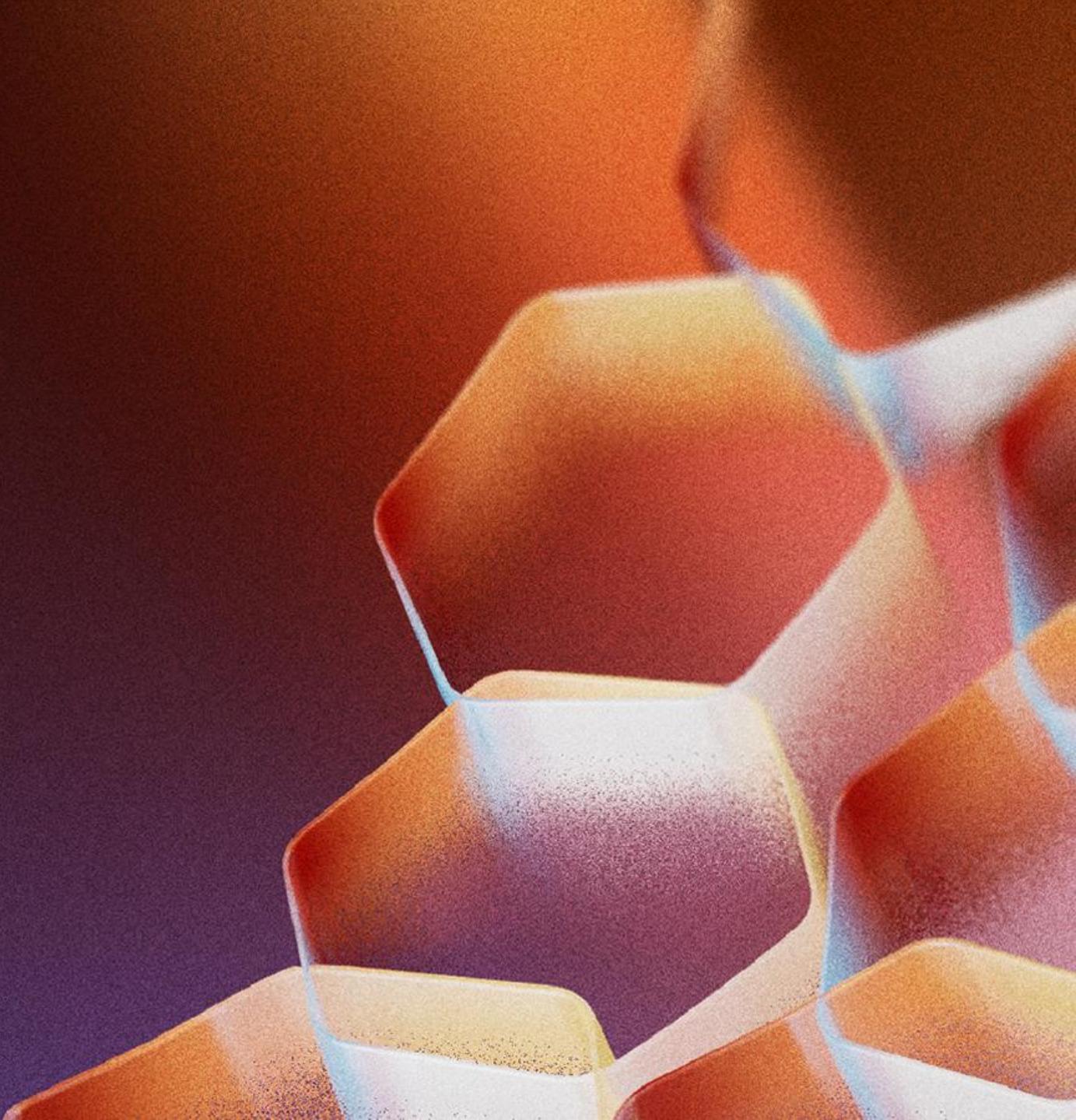


OOM Kill





Microsoft Build





Streamlining AKS Debugging: Techniques to solve common & complex problems

Aritra Ghosh, Jose Blanquicet, Juan-Lee Pang





Streamlining AKS Debugging: Techniques to solve common & complex problems

Aritra Ghosh, Jose Blanquicet, Juan-Lee Pang

Let's jump right in with a 2 question poll!

aka.ms/buildaksdebug



Agenda

- Introductions
- Opening
- Troubleshooting Network Issues
- Node Health & Resource Utilization
- Create, Read, Update, Delete
- Best practices and Announcements
- Q&A

Introductions



Aritra Ghosh
*Senior Product
Manager*

*AKS
Observability*



Jose Blanquicet
*Senior Software
Engineer*

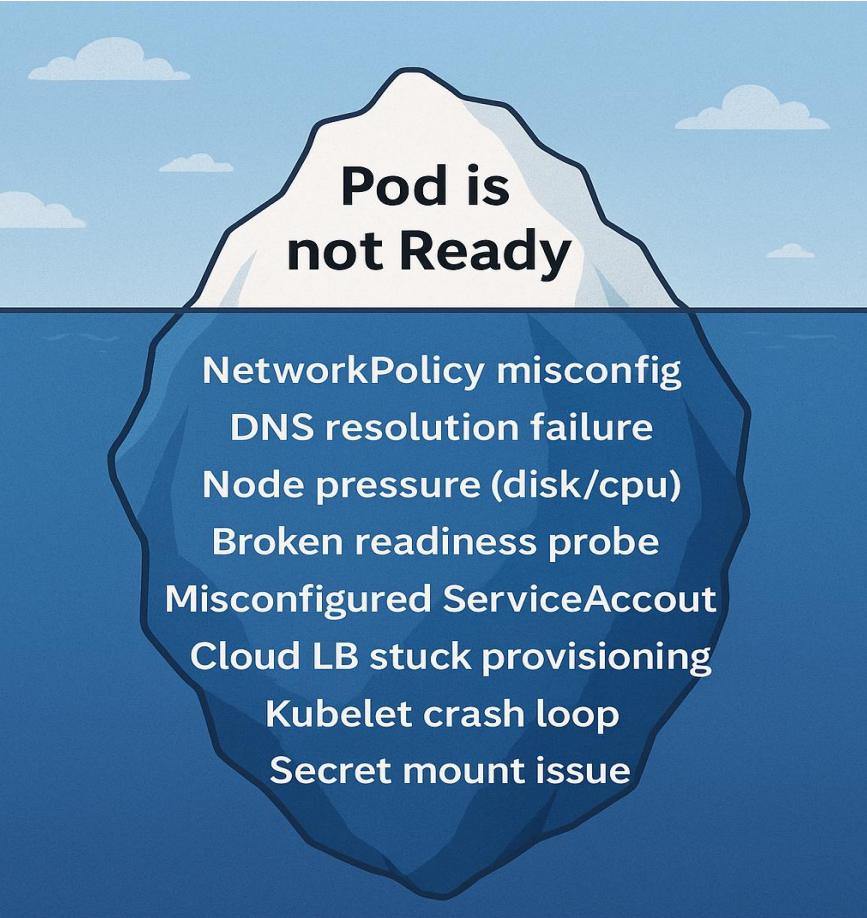
AzCore Linux



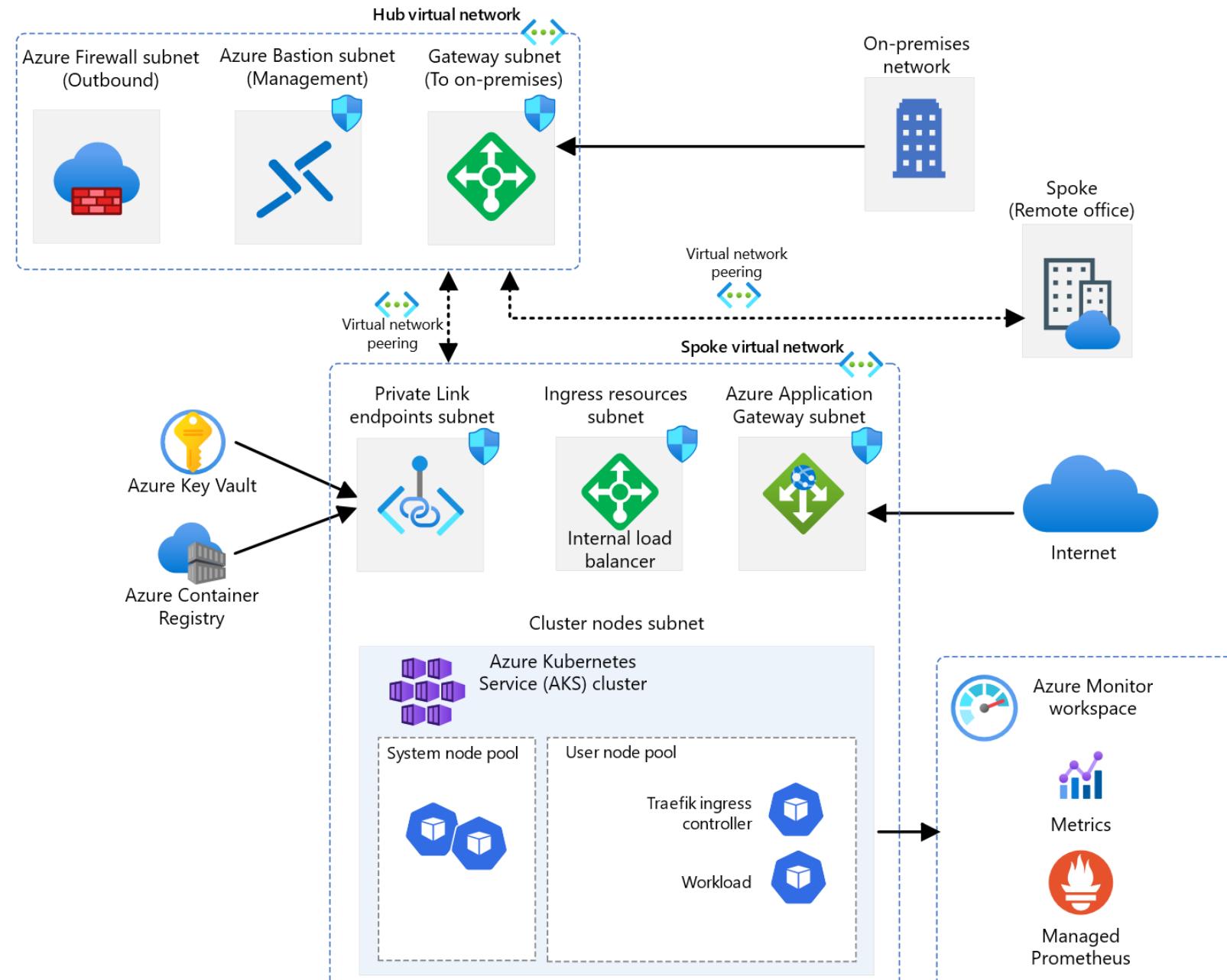
Juan-Lee Pang
*Principal SWE
Manager*

*AKS
Observability*

Kubernetes troubleshooting is easy!



Navigating the journey of troubleshooting Kubernetes



Scenario 1: Troubleshooting Network Issues



Focus on DNS and TCP

What happens during a typical network connection?

DNS resolution (translating domain name to IP addresses)



TCP/UDP transport setup



Application-layer communication (e.g., HTTPs, gRPC)

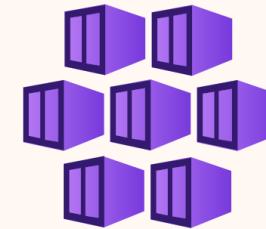


Debugging Tools



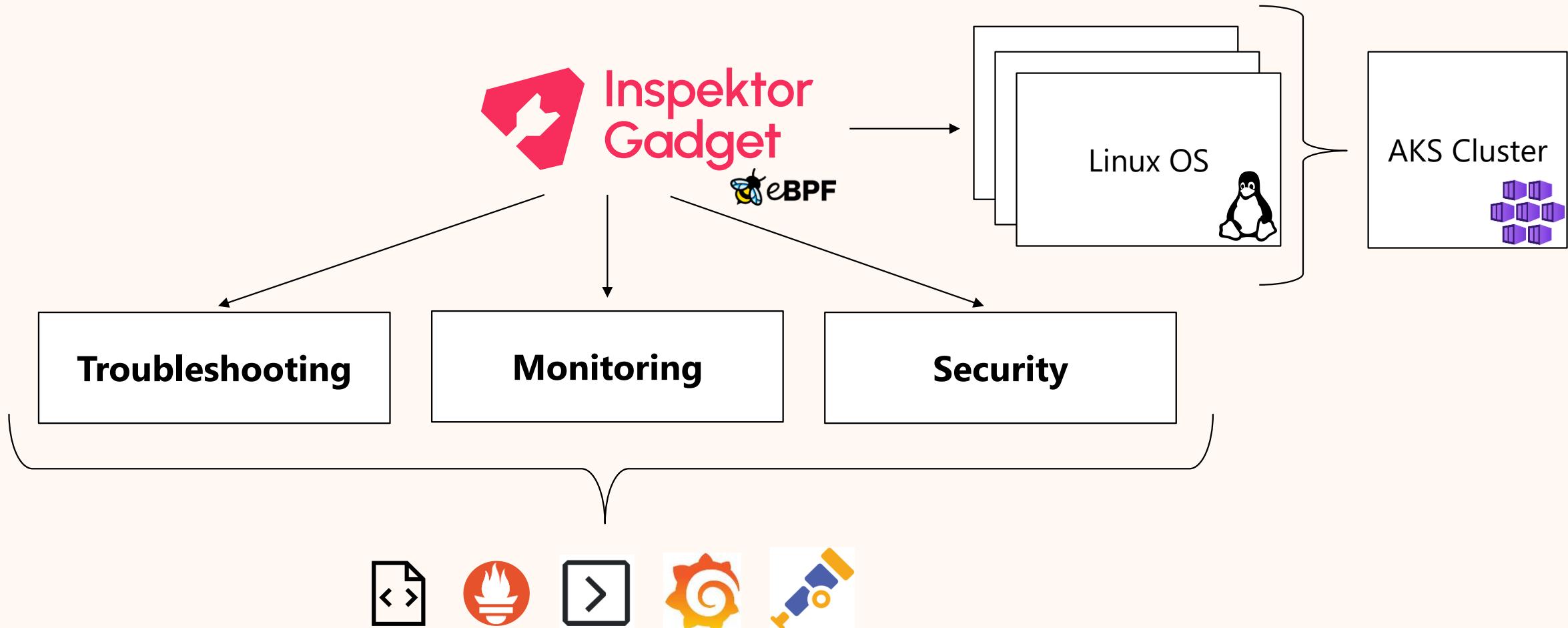
Inspektor
Gadget

*Real-Time debugging
with
Inspektor Gadget*

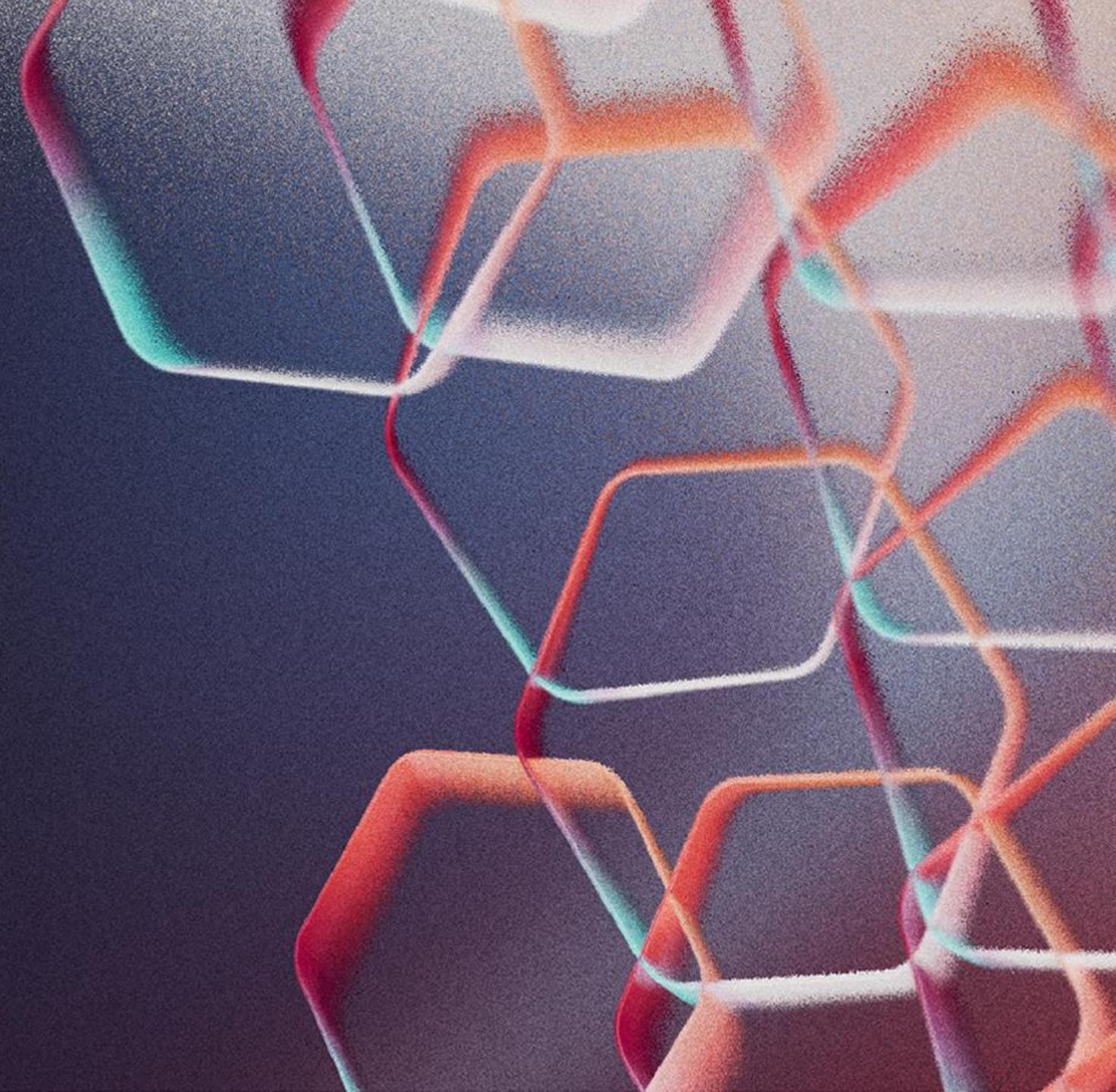


*Kubernetes
Status, Logs, etc.*

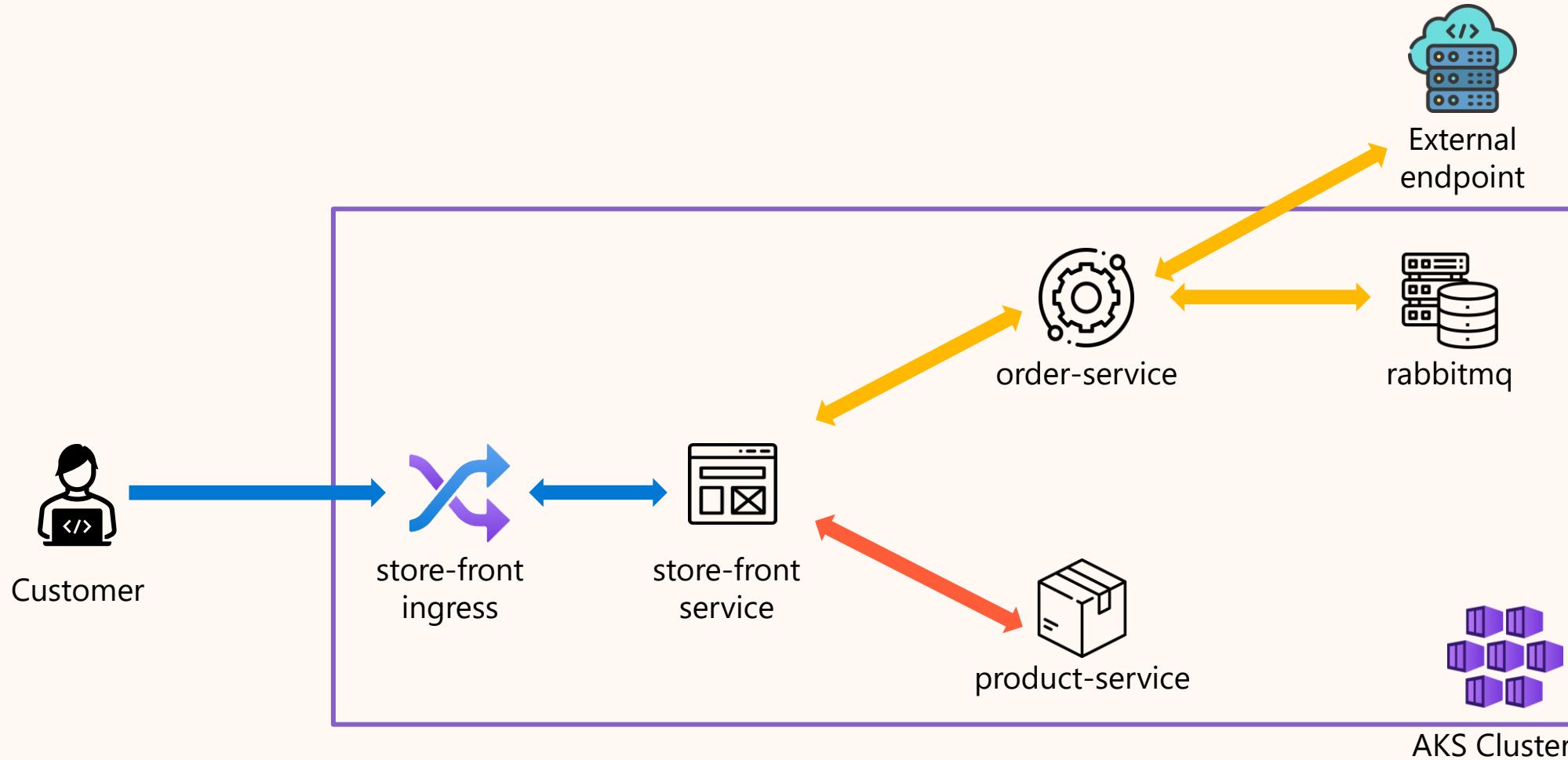
Inspektor Gadget



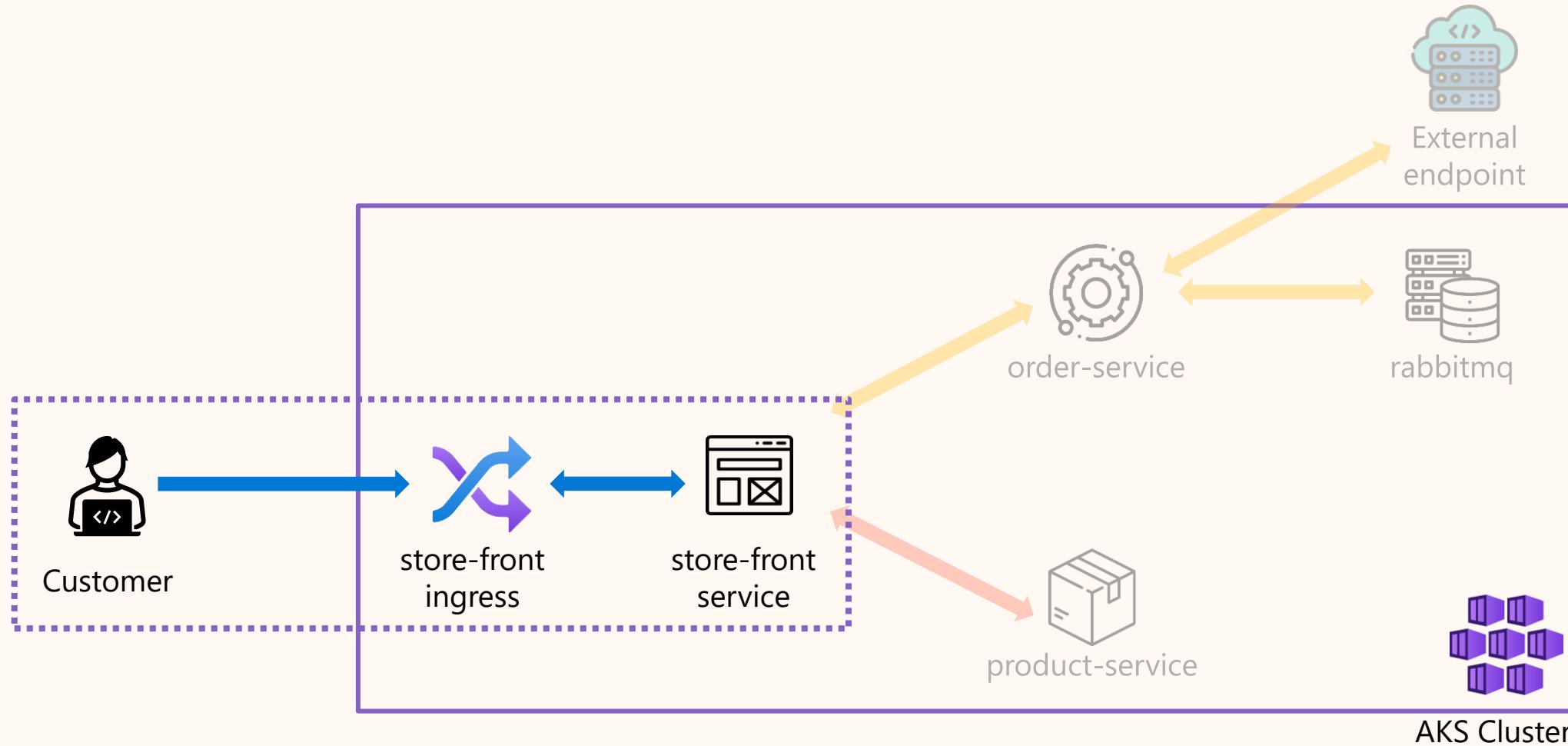
Demo



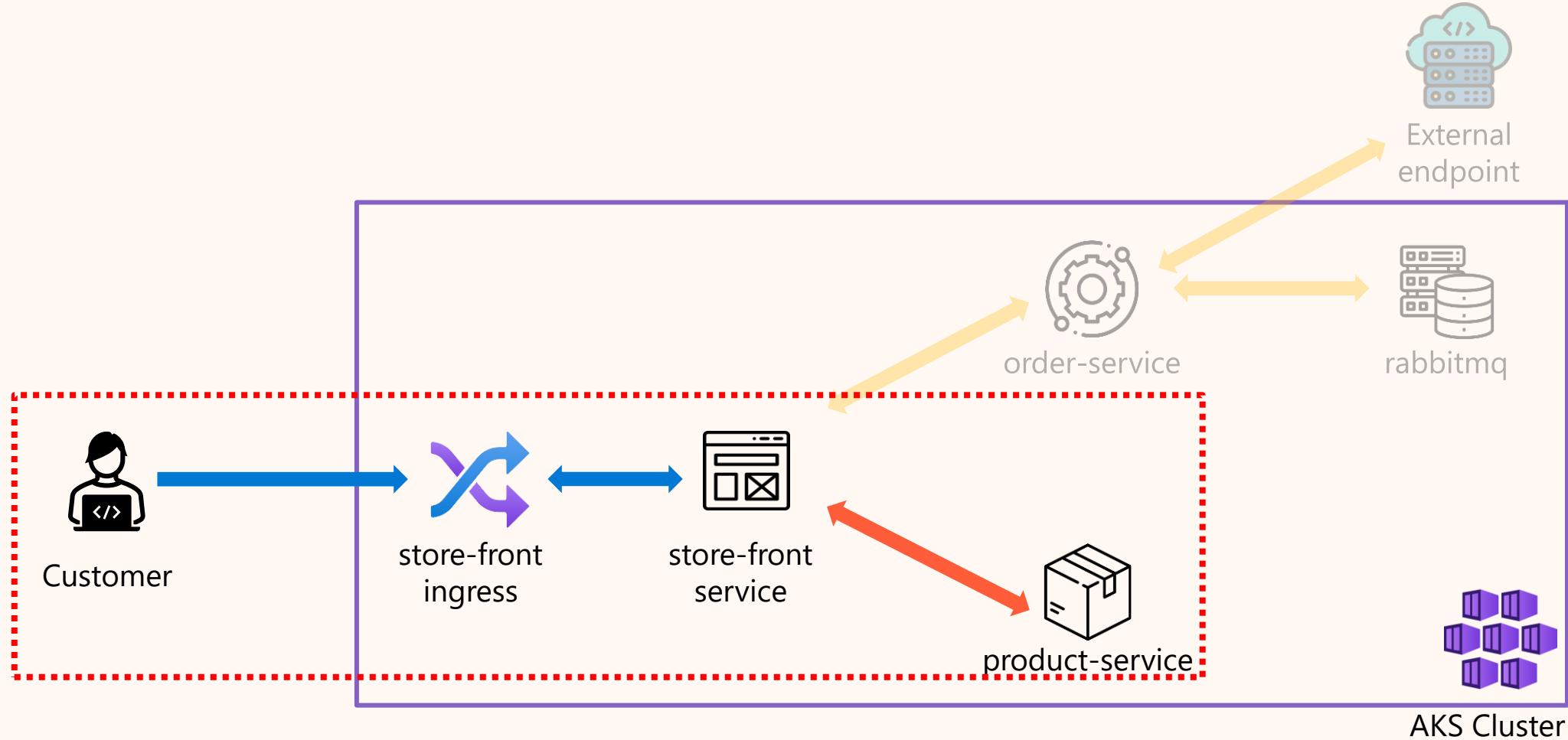
App Architecture



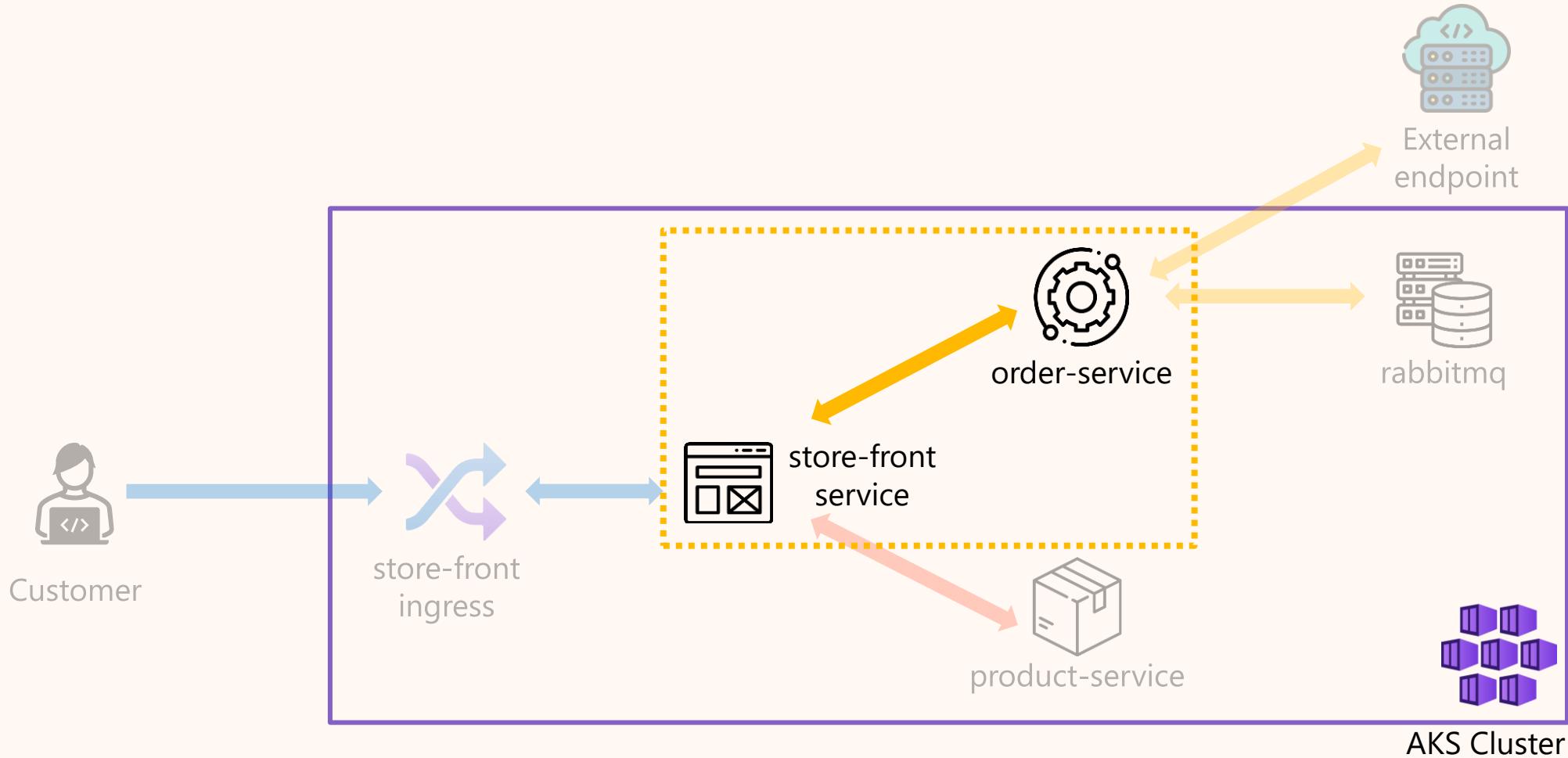
Flowchart: Entry-point



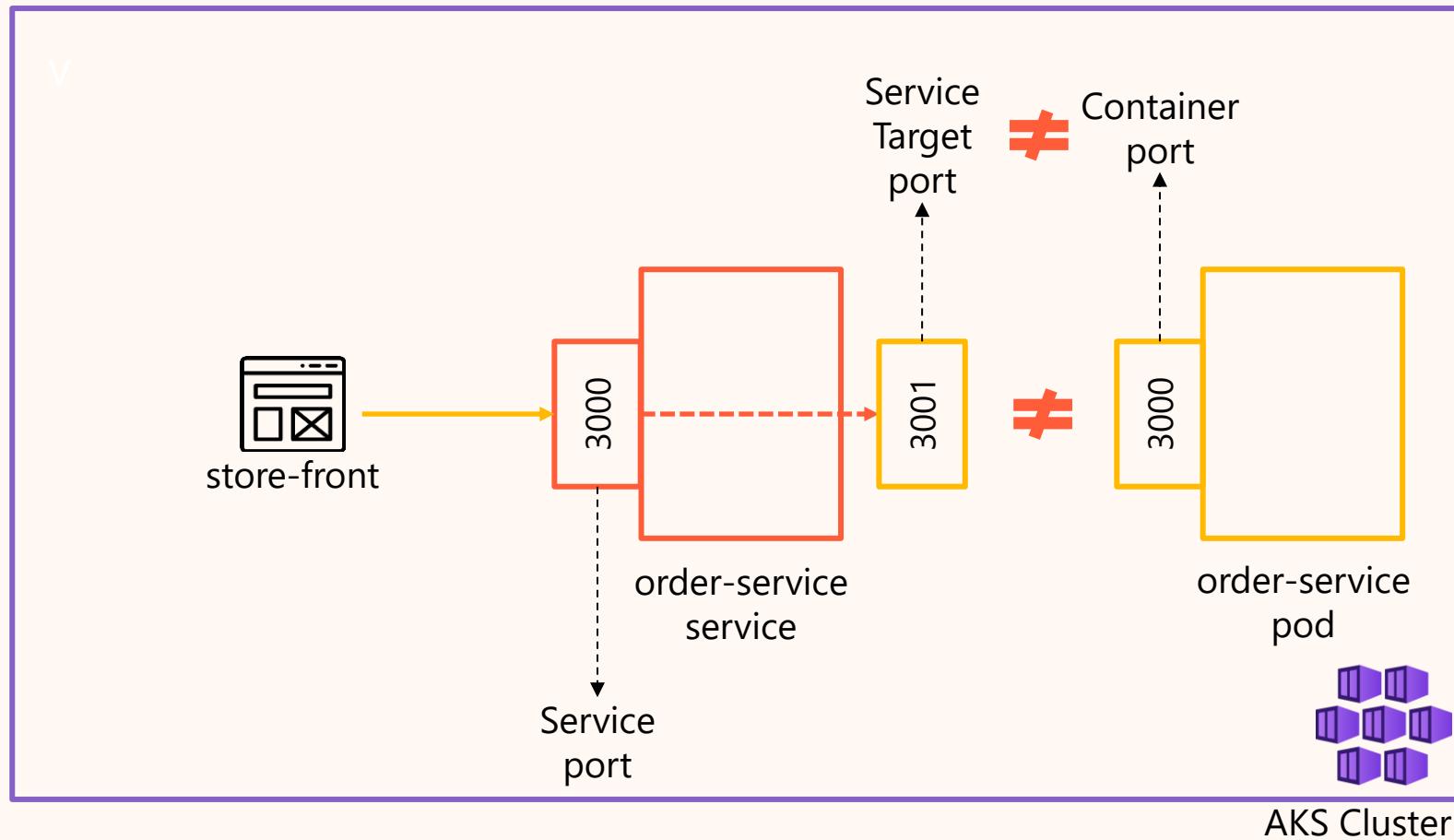
Flowchart: Building the front-end



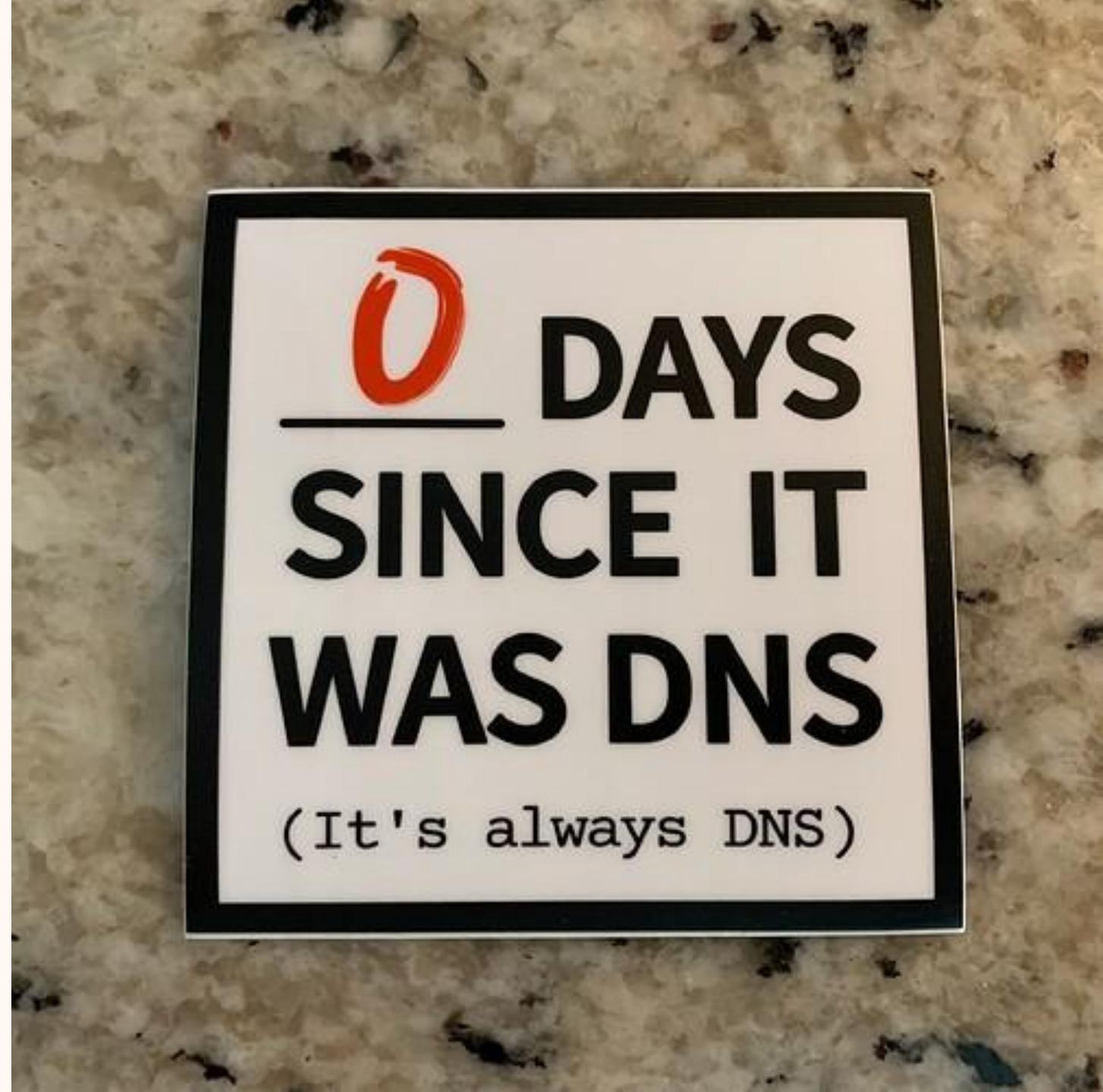
Flowchart: Checkout an order (Back-end)



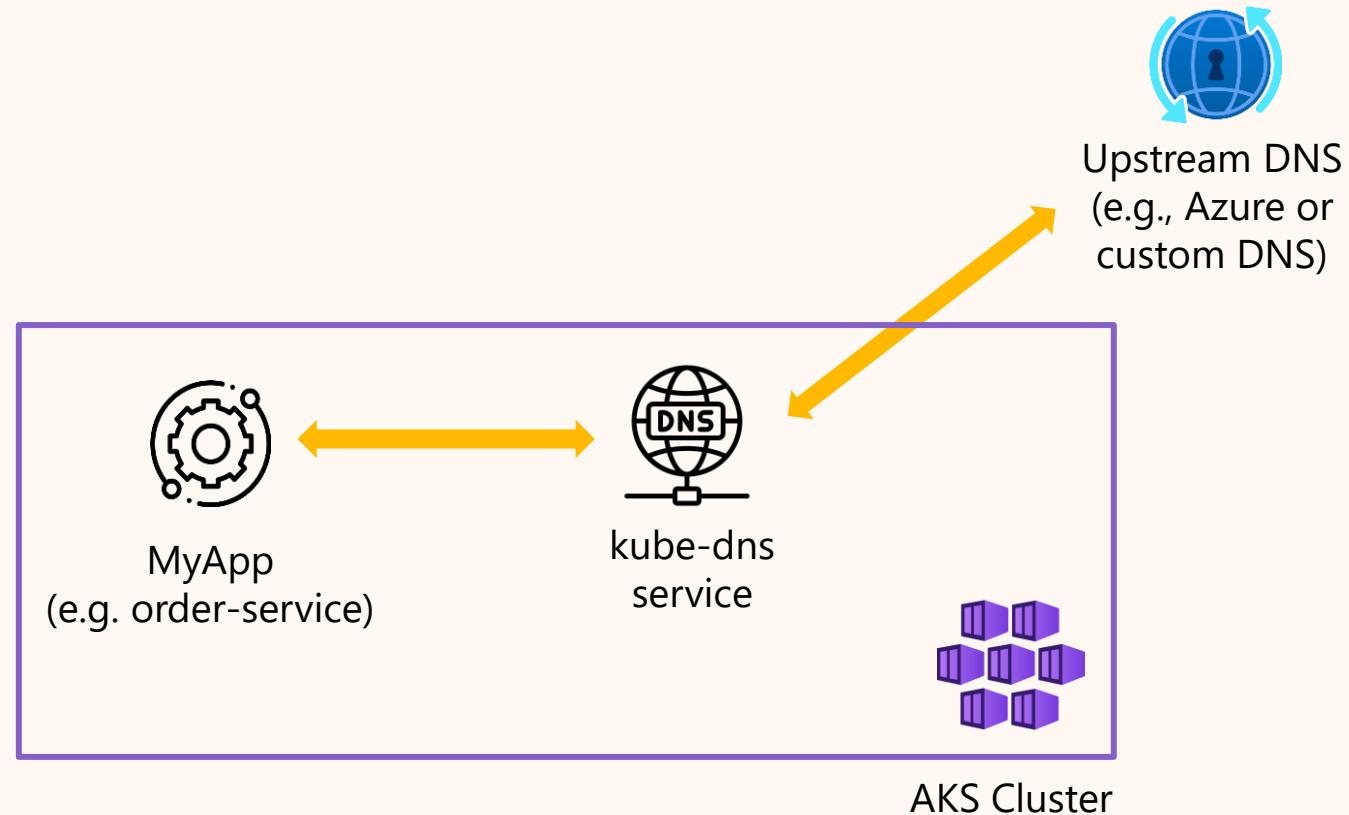
Issue: Wrong service configuration



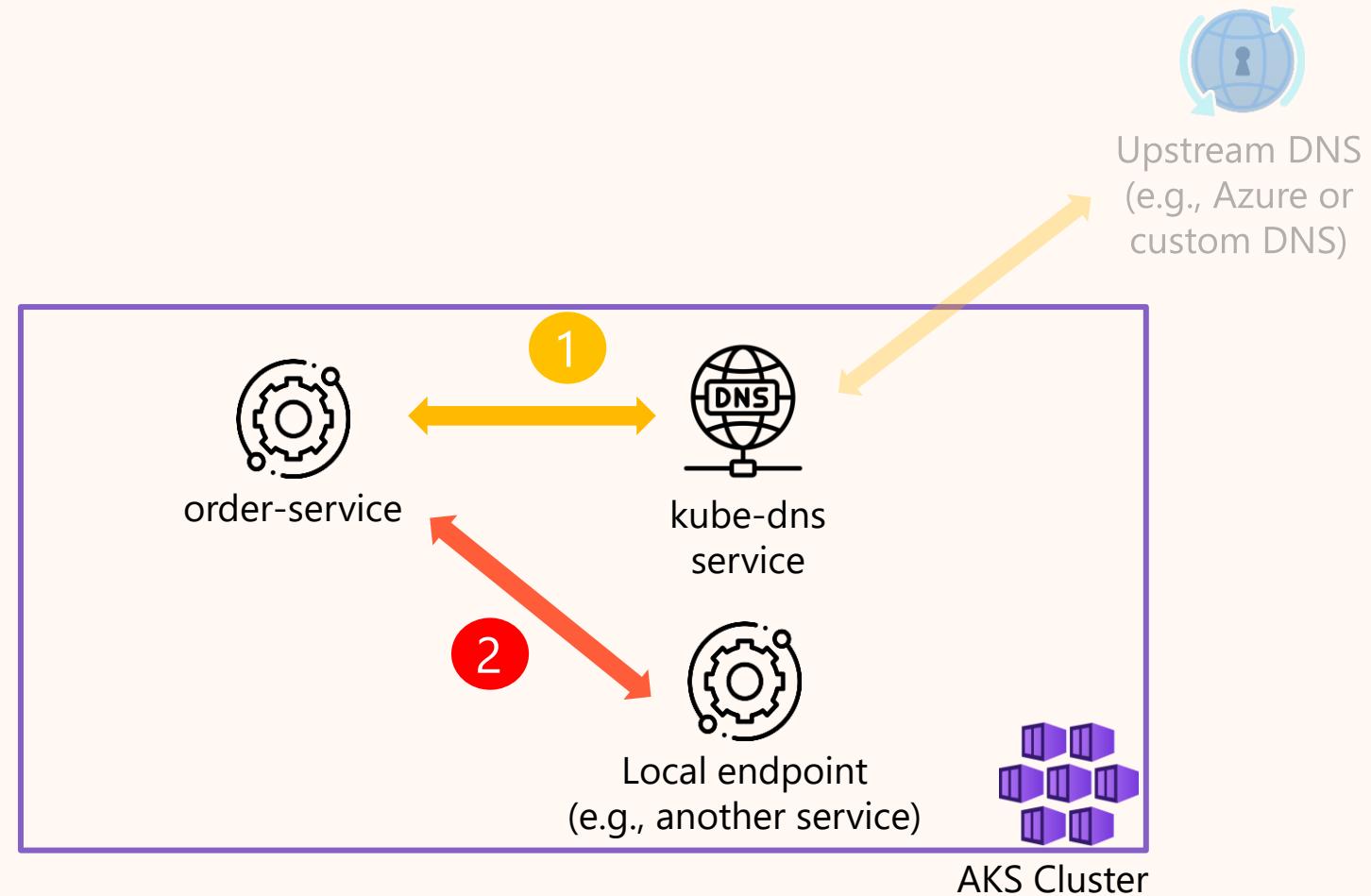
Troubleshooting DNS in AKS



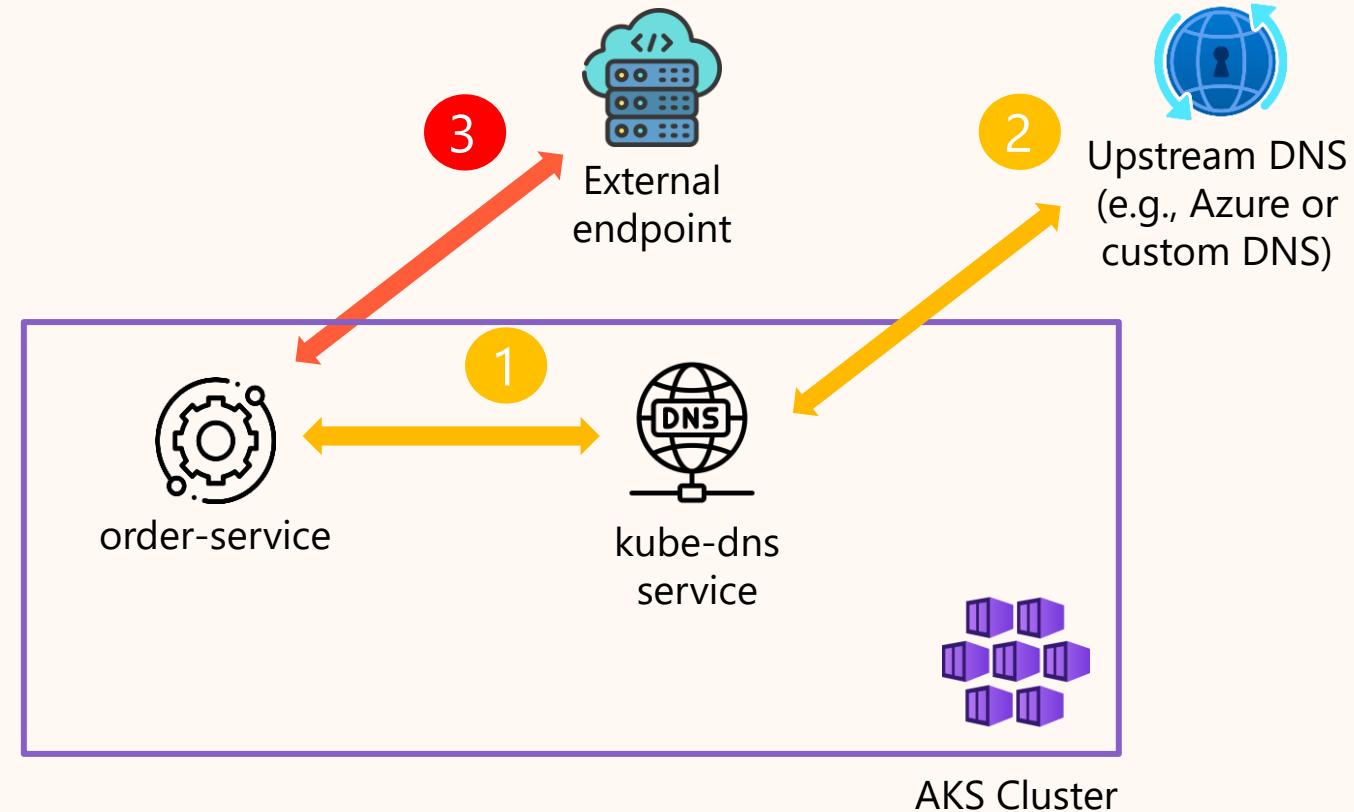
Context: Components involved in a DNS resolution



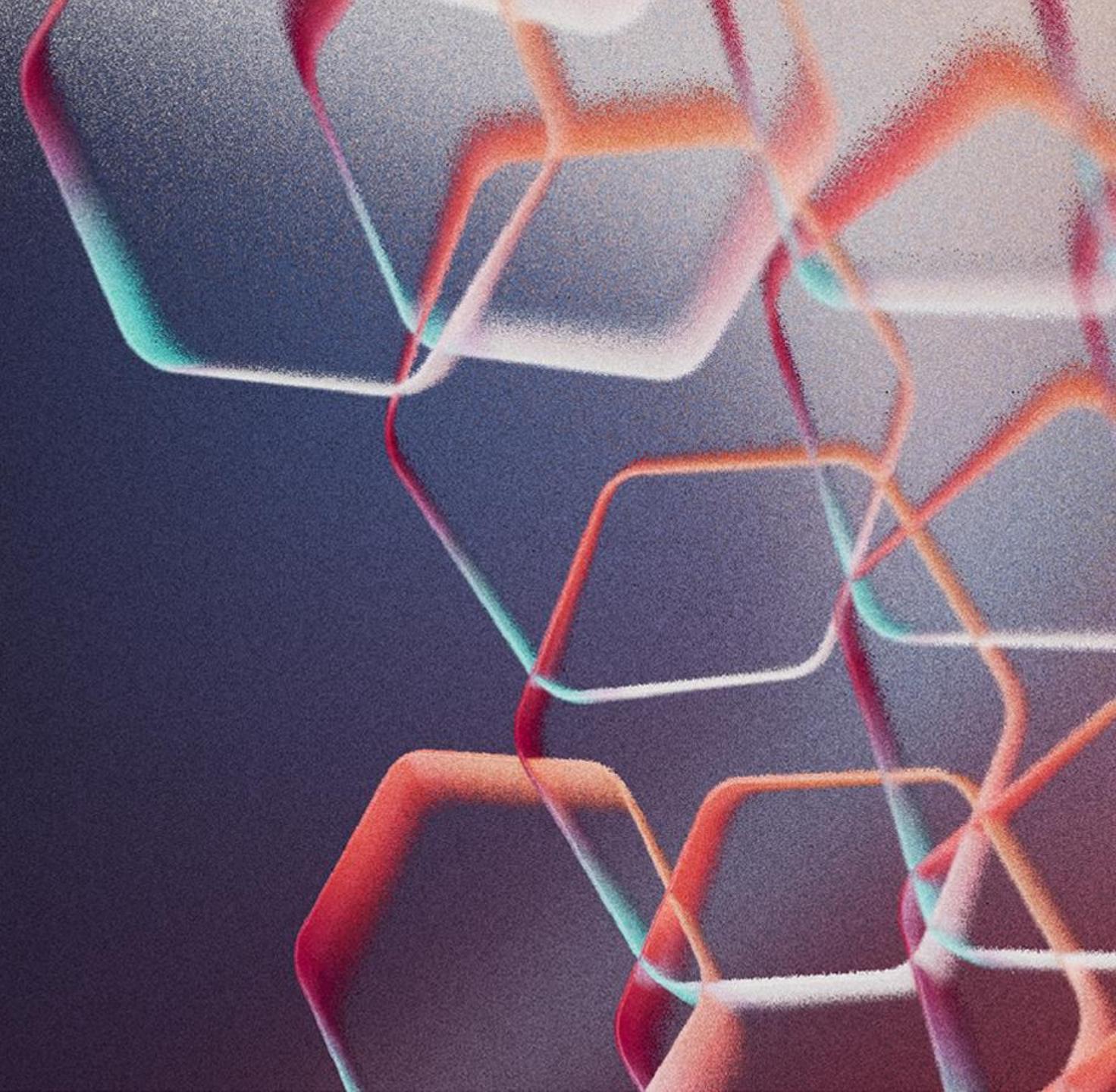
DNS: Local endpoints



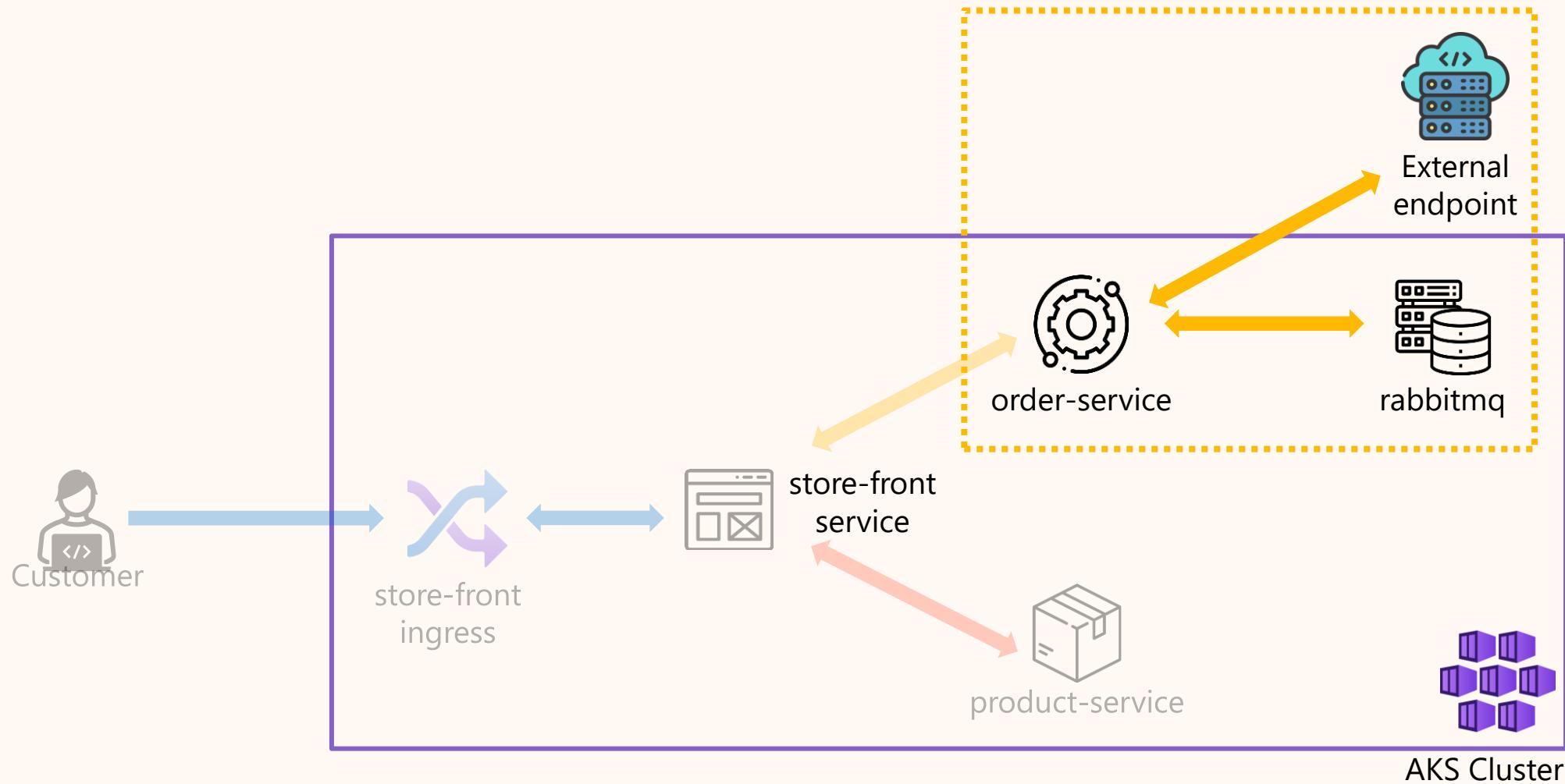
DNS: External endpoints



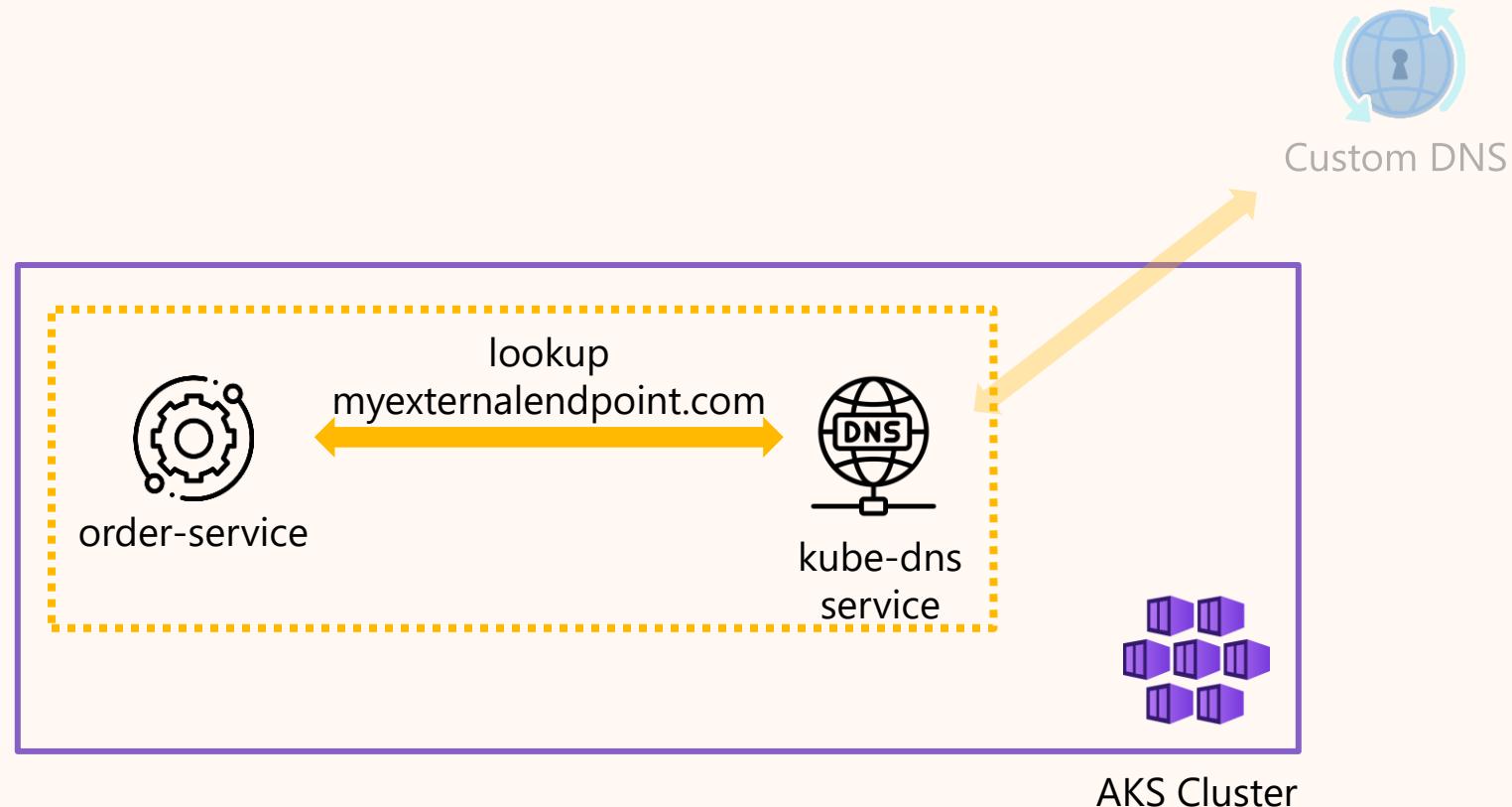
Demo



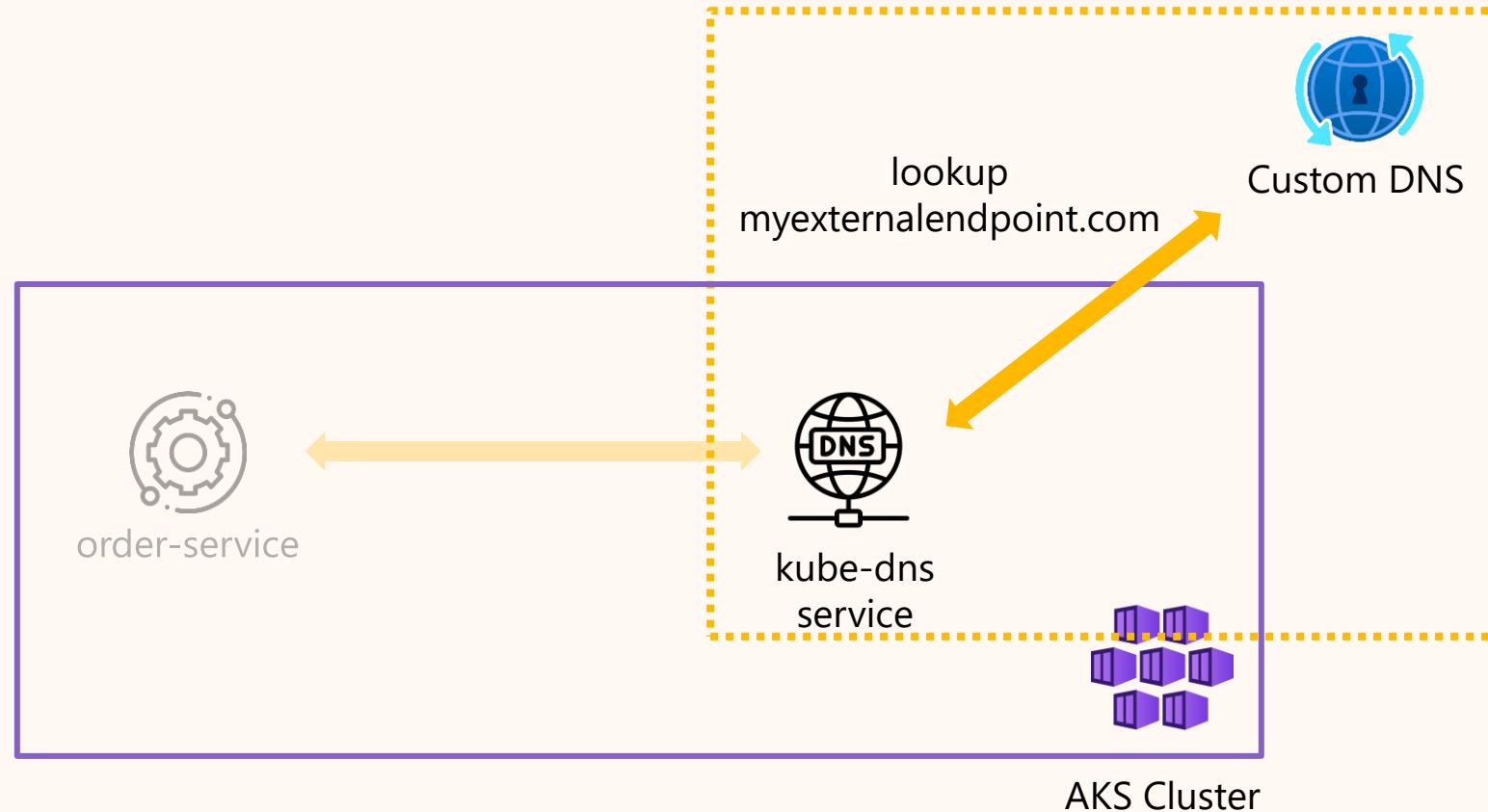
Demo: Simulate an issue with the external endpoint



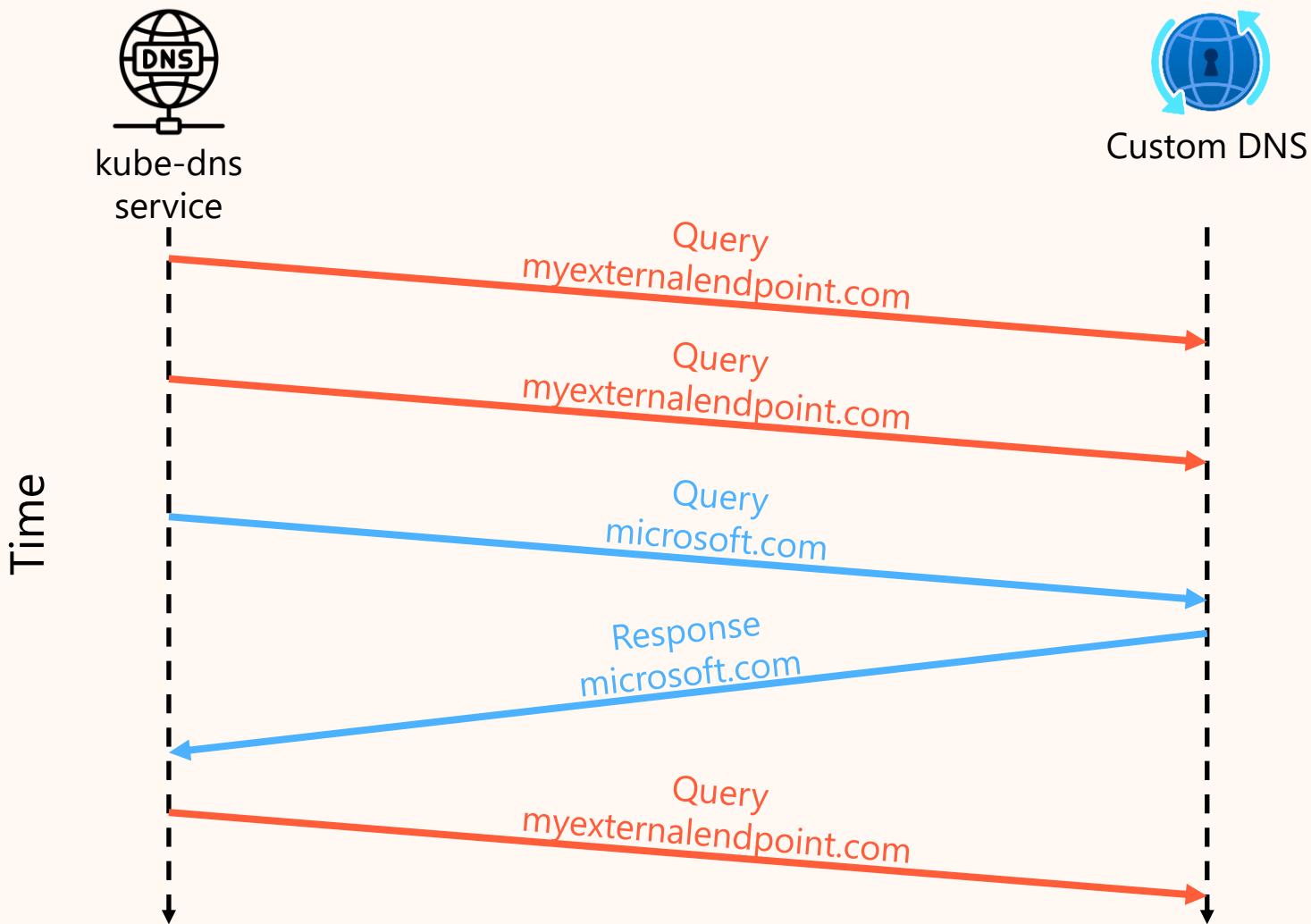
Demo: order-service <-> kube-dns



Demo: kube-dns <-> Custom DNS



Issue: Wrong custom DNS configuration



Takeaways

- Isolate
 - Using an inside-out approach streamlines troubleshooting
 - Start from one app and then see how it interacts with other etc...
- Tools & data
 - Logs - valuable source of information to start the analysis with
 - Inspektor Gadget
 - Kubernetes aware
 - Keep manifest files for common troubleshooting scenarios
 - Interactive & multipurpose (security, networking, performance, etc.)
- Best practices
 - For those few static external names, use fully qualified domain names (FQDN) to avoid generating unnecessary DNS traffic
- And that's not just what IG does... inspektor-gadget.io

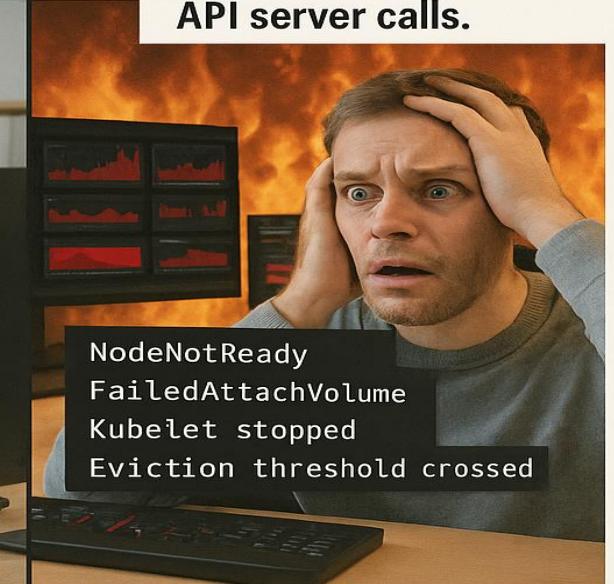
Scenario 2: Node Health

TROUBLESHOOTING NODE HEALTH IN KUBERNETES

Should be a
quick check...



Turns out kubelet died, disk
pressure triggered eviction,
and network policy blocked
API server calls.



**TURNS OUT KUBELET died,
disk pressure triggered eviction,
and network policy blocked
API server calls.**

Scenario 2: Node Health Diagnosis

Node
conditions

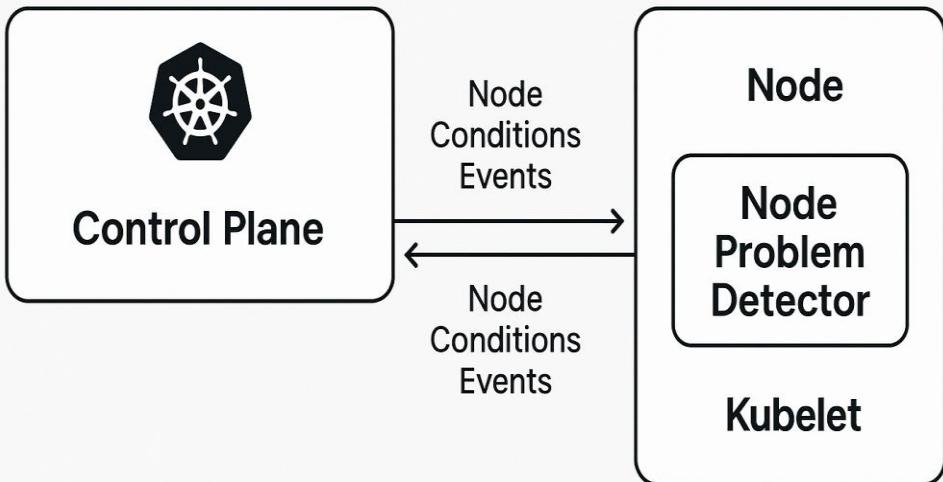
Kubernetes
Events

Resource
Metrics

Scenario 2: Node Conditions

- AKS installs *Node Problem Detector (NPD)* on nodes to monitor health
- *NodeNotReadyDiagnostics* condition provides node readiness details

Node Problem Detector



NPD in AKS

Type	Status	Last probe time
ReadonlyFilesystem	False	2025-05-15T19:51:07Z
ContainerRuntimeProbl...	False	2025-05-15T19:51:07Z
KernelDeadlock	False	2025-05-15T19:51:07Z
VMEventScheduled	False	2025-05-15T19:51:07Z
KubeletProblem	False	2025-05-15T19:51:07Z
FrequentDockerRestart	False	2025-05-15T19:51:07Z
FrequentUnregisterNet...	False	2025-05-15T19:51:07Z
FrequentContainerdRest...	False	2025-05-15T19:51:07Z
FilesystemCorruptionPr...	False	2025-05-15T19:51:07Z
FrequentKubeletRestart	False	2025-05-15T19:51:07Z
PIDPressure	False	2025-05-15T19:49:25Z

Node Conditions in Azure Portal

Scenario 2: Node Conditions

Common Root Causes and How to Investigate

kubectl describe node <>

Cause	How to Check	Possible Fix
 Kubelet Crash / Hang	describe node, journalctl -u kubelet (if SSH access)	Restart kubelet (manually or via VM restart)
 Network Unreachable / CNI Issues	Check NetworkUnavailable, check CNI pods in kube-system	Restart affected CNI pods, check route tables
 Disk Pressure	describe node > Conditions + df -h (SSH)	Clean up disk (logs, unused images)
 OOM / Memory Pressure	MemoryPressure condition	Evict pods / increase VM size, Check for memory leaks
 API Server Unreachable	Try kubectl get commands, see if control plane responds	Check NSGs, firewalls, kube-proxy logs
 Container Runtime Down	crtcl info or systemctl status containerd/docker	Restart container runtime

Scenario 2: Node Events

Events provide additional insight into the health of the node.

Message	Type	Reason	Source	Object	Namespace	Count
⚠ kubelet healthcheck failed: curl: (28) Operation timed out after 1000	Warning	KubeletIsDown	kubelet-custom-plugin-monitor, ...	Node/aks-testnodepool-13033019-vmss0...	default	2
✓ Node aks-testnodepool-13033019-vmss000000 status is now: Node	Normal	NodeNotReady	node-controller	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ Node condition KubeletProblem is now: True, reason: KubeletIsDow	Warning	KubeletIsDown	kubelet-custom-plugin-monitor, ...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ crictl -t 60s pods --latest failed!	Warning	ContainerRunti...	container-runtime-custom-plugi...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ kubelet healthcheck failed: curl: (28) Operation timed out after 1000	Warning	KubeletIsDown	kubelet-custom-plugin-monitor, ...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task kworker/1:3:786744 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815743 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815747 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ Node condition ContainerRuntimeProblem is now: True, reason: Cor	Warning	ContainerRunti...	container-runtime-custom-plugi...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815750 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815752 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815754 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
⚠ INFO: task stress-ng-io:815756 blocked for more than 120 seconds.	Warning	TaskHung	kernel-monitor, aks-testnodepoo...	Node/aks-testnodepool-13033019-vmss0...	default	1
✓ Container image "ubuntu:20.04" already present on machine	Normal	Pulled	kubelet, aks-testnodepool-13033...	Pod/brutal-stress	default	1

```
kubectl get events --field-selector involvedObject.kind=Node,involvedObject.name=<>
```

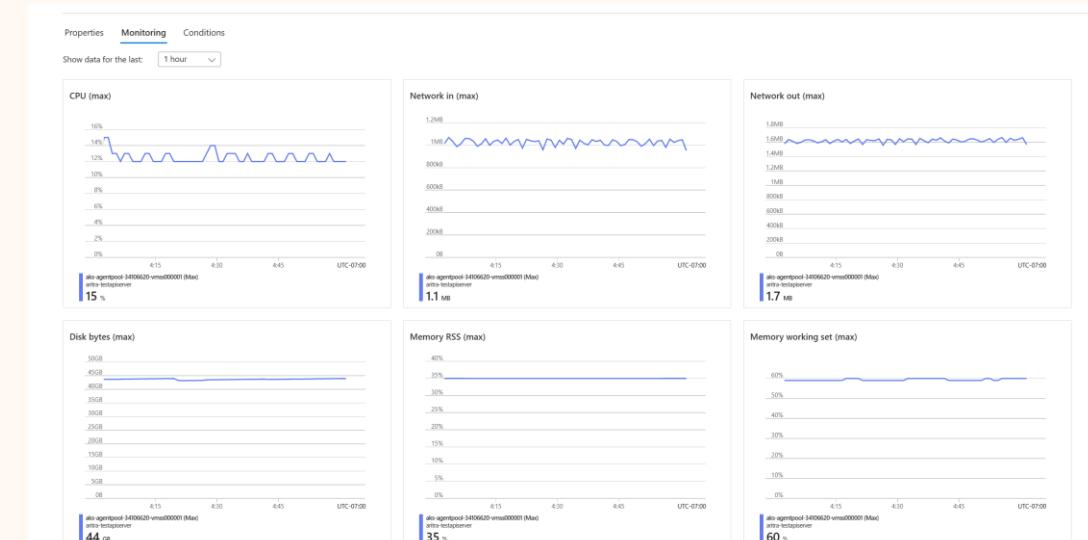
Scenario 2: Node Metrics

- Basic metrics : Memory, CPU, I/O, and Networking.
- *Node exporter* and *Cadvisor* provide node metrics that can be collected using your preferred tools.

Azure Grafana Node Dashboard



Node Views in Azure Portal



Scenario 2: Node Metrics

What if you Azure Managed Grafana is not an option?

- Azure Monitor dashboards with Grafana (*currently in Preview*)
- No additional fees apply
- You can utilize all standard dashboards, which can also be customized to meet your requirements.
- For more details, make sure to attend BRK 188

Scenario 2: Azure Monitor dashboards with Grafana(Preview)

Home > aritra-controlplanemetrics-test

aritra-controlplanemetrics-test | Dashboards with Grafana (preview) | Kubernetes | Compute Resources | Node (Pods) ☆ ...

Kubernetes service | Azure Monitor

Search ▾

Services and ingresses

Storage

Configuration

Custom resources

Events

Run command

Settings

Node pools

Upgrades

Security configuration

Application scaling

Networking

Extensions + applications

Backup

Service mesh - Istio

Open Service Mesh

GitOps

Automated deployments

Policies

Service Connector

Properties

Locks

Monitoring

Insights

Alerts

Metrics

Diagnostic settings

Advisor recommendations

Logs

Workbooks

Dashboards with Grafana (preview)

Data source DefaultAzureMonitorWorkspace-centralus ▾ cluster aritra-controlplanemetrics-test ▾ node aks-agentpool-62087549-vmss000000 ▾ Last 1 hour ▾ Refresh 1m ▾

CPU Usage

2025-05-07 14:52:00

Name

Name	Last
max capacity	8
ama-logs-rs-5795b89f4-t47qm	0.00680
ama-logs-z6qnf	0.0121
ama-metrics-854765d6d6-gvscv	0.00556
ama-metrics-ksm-b699d5fcf-b75cc	0.00128
ama-metrics-node-xr7qp	0.00952

CPU Quota

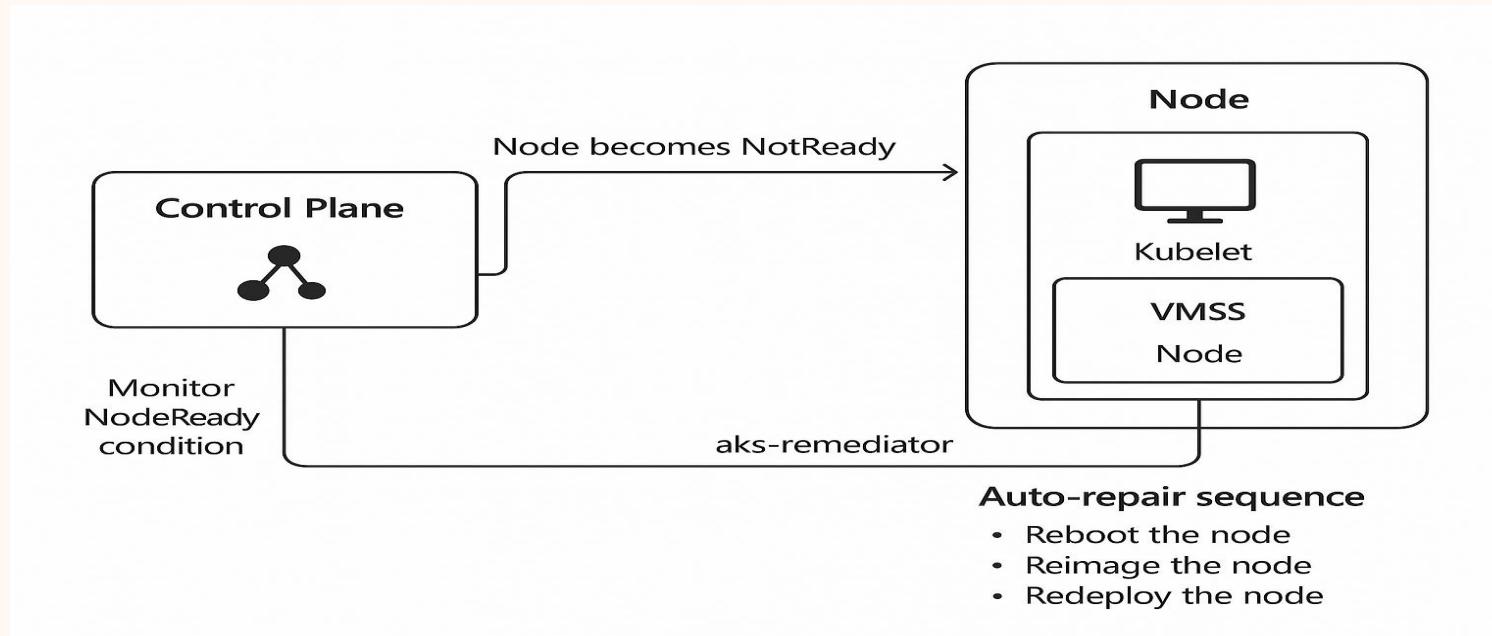
Pod	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
csi-azurefile-node-6q84s	0.000167	0.0300	0.557%		
coredns-autoscaler-777f7c56d8-689hn	0.000167	0.0200	0.834%	0.200	0.0834%
retina-agent-rdr25	0.000120	0.100	0.120%	0.500	0.0241%
ama-metrics-854765d6d6-gvscv	0.00527	0.170	3.10%	7.50	0.0702%

Memory Usage (w/o cache)

Pod	Memory Usage	Memory Requests	Memory Requests %	Memory Limits	Memory Limits %	Memory Usage (RSS)	Memory Usage (Cache)	Memory Usage (Swap)
azuredisknode-5f8f5c78c8-tcvrz	17.5 MiB	50 MiB	34.9%	250 MiB	6.99%	12.2 MiB	24.2 MiB	0 B
ama-logs-z6qnf	295 MiB	600 MiB	49.2%	2.10 GiB	13.8%	271 MiB	7.76 MiB	0 B
metrics-server-5f8f5c78c8-tcvrz	59.7 MiB	154 MiB	38.8%	424 MiB	14.1%	55.6 MiB	45.1 MiB	0 B
csi-azuredisk-node-kf9tk	34.4 MiB	60 MiB	57.4%	2.73 GiB	1.23%	32.1 MiB	15.4 MiB	0 B

Scenario 2: Node Auto Repair

- AKS automatically repairs nodes that remain unresponsive for over 10 minutes.
- If the issue persists, we attempt a reboot, reimagine, and redeploy in succession.
- Should the node continue to show as not ready, a deeper investigation is necessary to resolve the issue.

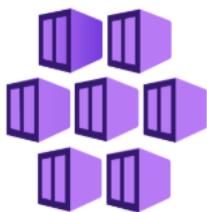


Scenario 2: Node Health Deepdive

- It might be necessary to delve further to identify the root cause.
- We are working on providing metrics at the process level within the Diagnose and Solve section.
- In uncommon situations, kubelet logs could offer insights into the failure.

Scenario 2: Node Health Demo

- Developers report an elevated error rate on one of the apps in production. As an SRE, you have correlated it to one of the nodes which went not ready. What could have caused it
- Tools used



Azure Monitor



Application Insights

Azure Prometheus + Grafana + AKS

ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/6107f264-6d8e-4c58-bb3f-c178d3cf23e8/resourceGroups/aritra-rg/providers/Microsoft.ContainerService/managedClusters/aritra-controlplanemetrics-test/overview

Microsoft Azure (Preview) Report a bug Search resources, services, and docs (G+) Copilot Home > aritra-controlplanemetrics-test Kubernetes service

Search Overview Activity log Access control (IAM) Tags Monitor Diagnose and solve problems Microsoft Defender for Cloud (preview) Cost analysis Resource visualizer Kubernetes resources Namespaces Workloads Services and ingresses Storage Configuration Custom resources Events Run command Settings Node pools Upgrades

Check deployment rollout in AKS List all pods in this AKS cluster Configure monitoring on aks clusters

Location : Central US Subscription ID : 6107f264-6d8e-4c58-bb3f-c178d3cf23e8 Fleet Manager : Click here to assign Tags (edit) : Add tags

Get started Properties Monitoring Recommendations

Kubernetes services

Encryption type	Encryption at-rest with a platform-managed key
Virtual node pools	Not enabled

Node pools

Node pools	4 node pools
Kubernetes versions	1.32.3
Node sizes	Standard_D8ds_v5, Standard_B2s, Standard_DS2_v2

Configuration

Kubernetes version	1.32.3
Auto Upgrade Type	Patch
Automatic upgrade scheduler	Every week on Sunday (recommended)
Node security channel type	Node Image
Security channel scheduler	Every week on Sunday (recommended)
Authentication and Authorization	Local accounts with Kubernetes RBAC
Local accounts	Enabled

Networking

API server address	aritra-controlplanemetrics-test-dns-zx9rt2xo.hcp.centralus.azmk8s.io
Network configuration	Azure CNI Overlay
Pod CIDR	10.244.0.0/16
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Cilium dataplane	Not enabled
Network Policy	None
Load balancer	Standard
Private cluster	Not enabled
Authorized IP ranges	Not enabled
Application Gateway ingress controller	Not enabled

Integrations

Container insights	Enabled
Workspace resource ID	DefaultWorkspace-6107f264-6d8e-4c58-bb3f-c178d3cf23e8-CUS
Service Mesh - Istio	Not enabled

Extensions + applications

Add or remove favorites by pressing Ctrl+Shift+F

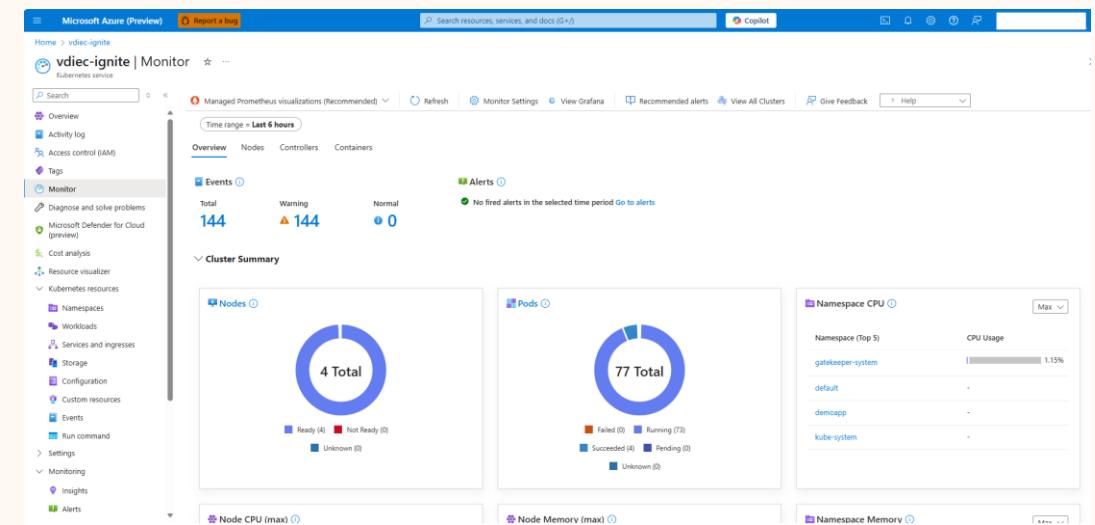
GA: Managed Prometheus visualizations in Azure Monitor for AKS and enhanced AKS Monitoring experience

This release brings comprehensive monitoring capabilities into a single, streamlined view—designed to address common challenges customers face when managing AKS clusters. Previously, Container insights visualizations were powered by metric data from Log Analytics. Customers now have the option to power these visualizations using managed Prometheus data, offering a more cost-efficient and performant solution.

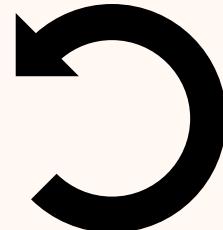
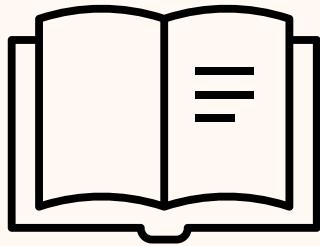
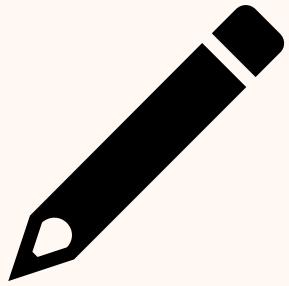
With this feature, customers can:

- Optimize costs by migrating from Log Analytics based metrics to managed Prometheus
- Improve performance with faster metric query response times
- Integrate with the new [Prometheus based recommended alerts](#)
- Gain visibility into control plane metrics for deeper troubleshooting
- Monitor at scale with the improved [multi-cluster view](#)

Learn More: <https://aka.ms/azmon-prometheus-visualizations>



Scenario 3: Create, Read, Update, Delete (CRUD)



Scenario 3: Failed Cluster/nodepool

- Performing any action on a cluster or node pool may lead it to enter a *failed* state.
- A failed state alone does not necessarily reflect the overall health of the cluster.
- The operation that most frequently encounters failure is the Upgrade (Node/K8s).

cluster | Node pools ⋮

« + Add node pool ⏪ Refresh ⏩ Start ⏻ Stop ⏪ Upgrade Kubernetes ⏪ Update image ⏪ Scale node pool ⏻ Delete

Node pools Nodes

Node pools provide space for applications to run. Node pools of different types can be added to the cluster to handle a variety of workloads, expanded and scaled, or node pools that are no longer needed can be deleted. Each node pool will contain nodes backed by virtual machines. [Learn more about node pools](#)

Diagnose Failed nodepool provisioning state

Autoscale events 0 Autoscale warnings 0 Scale-up not triggered 0

Node pool	Provisioning state	Power state	Scale method	Target nodes	Ready nodes	Auto-
agentpool	✖ Failed	Running	Manual	5	5	-
dfsdfds	Succeeded	Running	Autoscale	1	1	-

Last Operation Status

🔗 Reconcile ✖ Abort operation ⟳ Refresh

✖ Drain node aks-agentpool-21168581-vms00000p failed when evicting pod nginx-deployment-86dcfdf4c6-trntt. Evict blocked by conflicting disruption budgets. See <http://aka.ms/aks/debugdrainfailures>. Original error: This pod has more than one PodDisruptionBudget, which the eviction subresource does not support. PDB debug info: default/nginx-deployment-86dcfdf4c6-trntt

Details

Operation status	Failed
------------------	--------

Scenario 3: Failed Cluster/nodepool

Activity logs typically include details about operations that did not succeed.
In the case of upgrades, Azure Resource Graph maintains a record of automatic upgrades.

The screenshot shows two windows side-by-side. The left window is the 'Activity log' interface, and the right window is a detailed view of a specific activity entry.

Activity log (Left Window):

- Header:** Activity log, Search, Quick Insights.
- Filters:** Management Group: None, Subscription: aritraghosh-subscription, Event severity: All, Timespan: Last 2 weeks, Resource group: aria, Resource: tryrecommended.
- Table Headers:** Operation name, Status, Time, Time stamp, Subscription.
- Data:** A list of 71 items, mostly successful operations like 'List clusterAdmin credential' and 'List clusterUser credential'. One item, 'Create or Update Agent Pool', is highlighted in grey and shows a status of 'Failed'.

Create or Update Agent Pool (Right Window):

- Timestamp:** Sat Apr 26 2025 17:39:16 GMT-0700 (Pacific Daylight Time)
- Actions:** + New alert rule, + New support request.
- Summary:** Operation name: Create or Update Agent Pool, Time stamp: Sat Apr 26 2025 17:39:16 GMT-0700 (Pacific Daylight Time), Event initiated by: Microsoft.ContainerService.

Scenario 3: CRUD demo

I tried scaling one of my nodepools on the cluster and it failed. What do I do now and how can I fix it

Tools used

- Azure Portal
- Azure Resource Graph

ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/6107f264-6d8e-4c58-bb3f-c178d3cf23e8/resourceGroups/agrg/providers/Microsoft.ContainerService/managedClusters/nodenotreadytest/overview

Microsoft Azure (Preview) Report a bug Search resources, services, and docs (G+) Copilot Home > nodenotreadytest Kubernetes service

Downgrade this AKS cluster to free tier Update pricing tier for this AKS cluster Check deployment rollout in AKS

Create Connect Start Stop Delete Refresh Open in mobile Give feedback

View Cost JSON View

Overview

Resource group: agrg
Power state: Running
Cluster operation status: Succeeded
Subscription: aniraghosh-subscription
Location: East US 2
Subscription ID: 6107f264-6d8e-4c58-bb3f-c178d3cf23e8
Fleet Manager: Click here to assign
Tags (edit): Add tags

Kubernetes version: 1.31.7
API server address: nodenotreadytest-dns-fy5040gthcp.eastus2.azmk8s.io
Network configuration: Azure CNI Overlay
Node pools: 2 node pools - 1 failed
Container registries: Attach a registry

Get started Properties Monitoring Recommendations

Kubernetes services

Encryption type: Encryption at rest with a platform-managed key
Virtual node pools: Not enabled

Node pools

Node pools: 2 node pools - 1 failed
Kubernetes versions: 1.31.7
Node sizes: Standard_D2ds_v4, Standard_B2s

Configuration

Kubernetes version: 1.31.7
Auto Upgrade Type: Patch
Automatic upgrade scheduler: Every week on Sunday (recommended)
Node security channel type: Node Image
Security channel scheduler: Every week on Sunday (recommended)
Authentication and Authorization: Local accounts with Kubernetes RBAC
Local accounts: Enabled

Networking

API server address: nodenotreadytest-dns-fy5040gthcp.eastus2.azmk8s.io
Network configuration: Azure CNI Overlay
Pod CIDR: 10.244.0.0/16
Service CIDR: 10.0.0.0/16
DNS service IP: 10.0.0.10
Oflum dataplane: Not enabled
Network Policy: None
Load balancer: Standard
Private cluster: Not enabled
Authorized IP ranges: Not enabled
Application Gateway ingress controller: Not enabled

Integrations

Container insights: Not enabled
Workspace resource ID: -
Service Mesh - Istio: Not enabled

Extensions + applications

No extensions installed

Add or remove favorites by pressing Ctrl+Shift+F

Scenario 3: CRUD takeaways

- Set up alerts for failed auto upgrades
- Begin with Last Operation Status

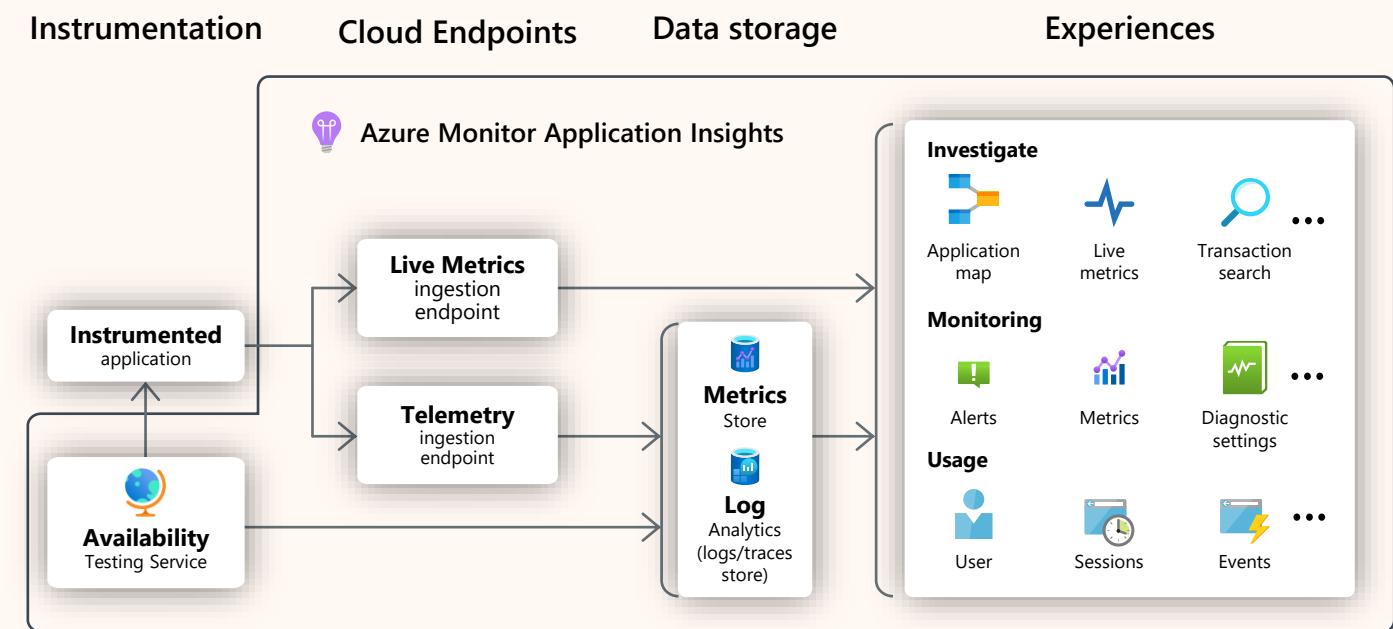
Auto-instrumentation with Application Insights

Benefits of application observability without the effort of instrumenting the application code

Automatically Instrument
applications with **no application**
source code modification
required.

Use the Azure Monitor Application
Insights OpenTelemetry Distro
in application pods **for Java**
or Node.js.

Per **namespace** or per **deployment**
onboarding.



<https://learn.microsoft.com/azure/azure-monitor/app/kubernetes-codeless>

Wouldn't it be great to avoid these issues...

Node Sizing	Disk Hygiene	Network Resilience	Monitoring
<ul style="list-style-type: none">• Right size nodes• Use taints/affinities to isolate system vs user workloads	<ul style="list-style-type: none">• Rotate logs, clean unused images & volumes regularly• Monitor inode and disk usage trends	<ul style="list-style-type: none">• Avoid blocking control plane traffic via NSGs or misconfigured network policies• Ensure DNS and Azure API endpoints are reachable• For static external names, use fully qualified names to avoid generating unnecessary DNS traffic	<ul style="list-style-type: none">• Set alerts for things like failed auto upgrades, MemoryPressure, DiskPressure, and NodeReady state changes• Track kubelet/container runtime health using Prometheus or Azure Monitor

GA: Azure Monitor Prometheus community recommended alerts for AKS

Azure Monitor now offers one-click enablement of Prometheus recommended alerts directly in the Azure Portal. These alerts, based on enhanced Prometheus community rules, provide comprehensive coverage across cluster, node, and pod levels. Previously, enabling these alerts required manual template downloads and CLI deployment.

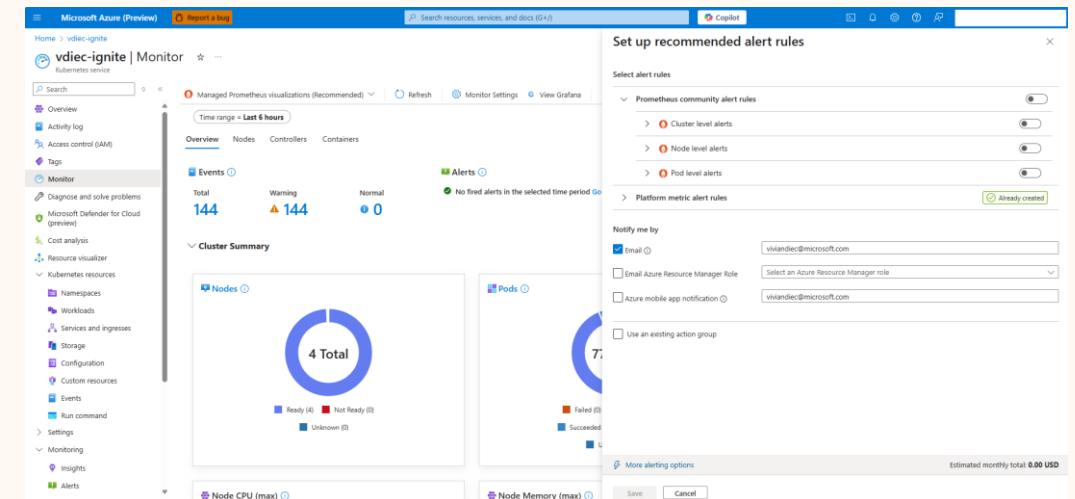
To use these alerts, your cluster must have [Azure Monitor managed service for Prometheus](#) enabled. They serve as the replacement for the legacy [Container insights recommended alerts \(custom metrics\) \(preview\)](#).

By enabling these alerts, customers will:

- Receive timely notifications on critical cluster issues
- Accelerate triage and troubleshooting with preconfigured signal coverage
- Improve cluster reliability and performance with minimal configuration

Soon, these alerts will also be enabled by default when provisioning AKS clusters through the Azure Portal.

Learn More: <https://aka.ms/prometheus-community-alerts>



Thank you

Table Talk

- Chat with us further at the **Table Talk**
- May 21st 2pm PT

Upcoming Sessions

- LAB345-R1 – Deploying and Inferencing AI Applications on K8s
 - May 21, 2:45pm PT
- LAB342-R1 – Streamlining K8s for developers with AKS Automatic
 - May 22, 8:30am PT

Collaborate on OSS troubleshooting tooling

- Kubernetes/node-problem-detector
- inspektor-gadget.io

Q&A

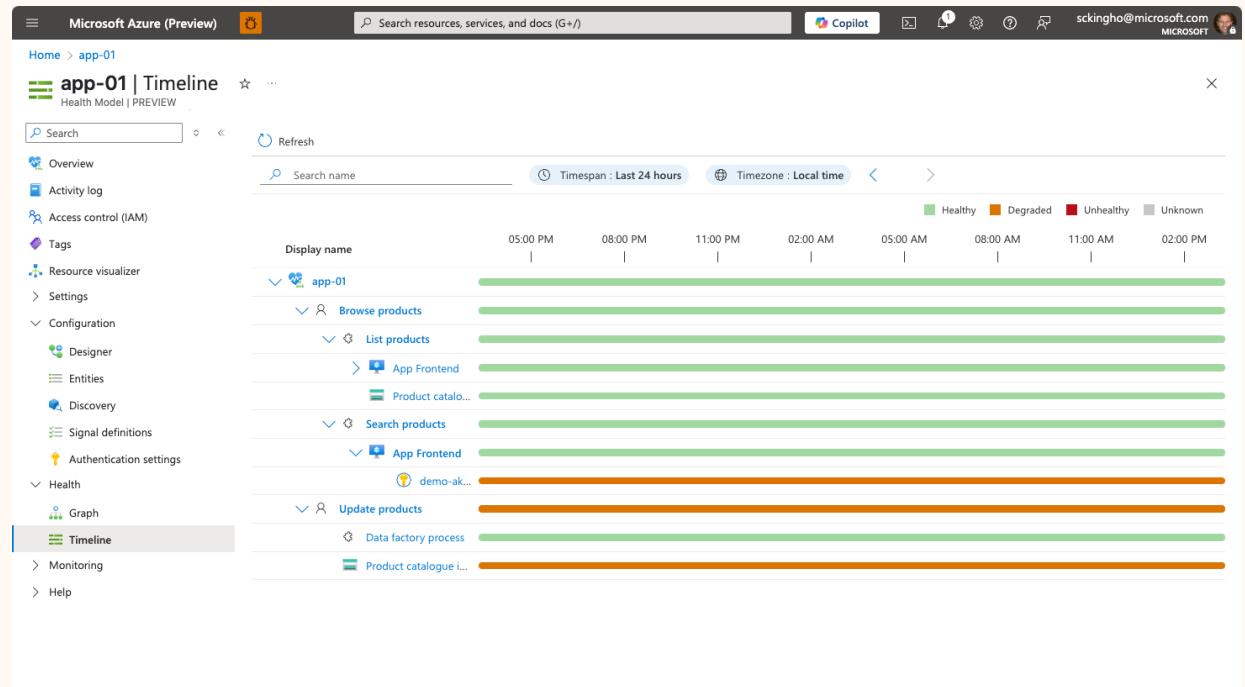
Appendix

Azure Monitor Health Model (Public Preview)

Simplifying and accelerating AKS workload troubleshooting with intelligent, customizable health insights

Key Benefits

- Unified, intelligent workload health enriched with application context
- Out-of-the-box health model tailored to your environment
- Consolidated signals to reduce alert noise
- Actionable insights to accelerate detection, diagnosis, and resolution
- aka.ms/Build25/AHMBlog



Kubernetes troubleshooting is easy!

What my PM thinks Kubernetes is ...



What I know Kubernetes is ...

Misconfigured Resource Limits



OOM Kill

