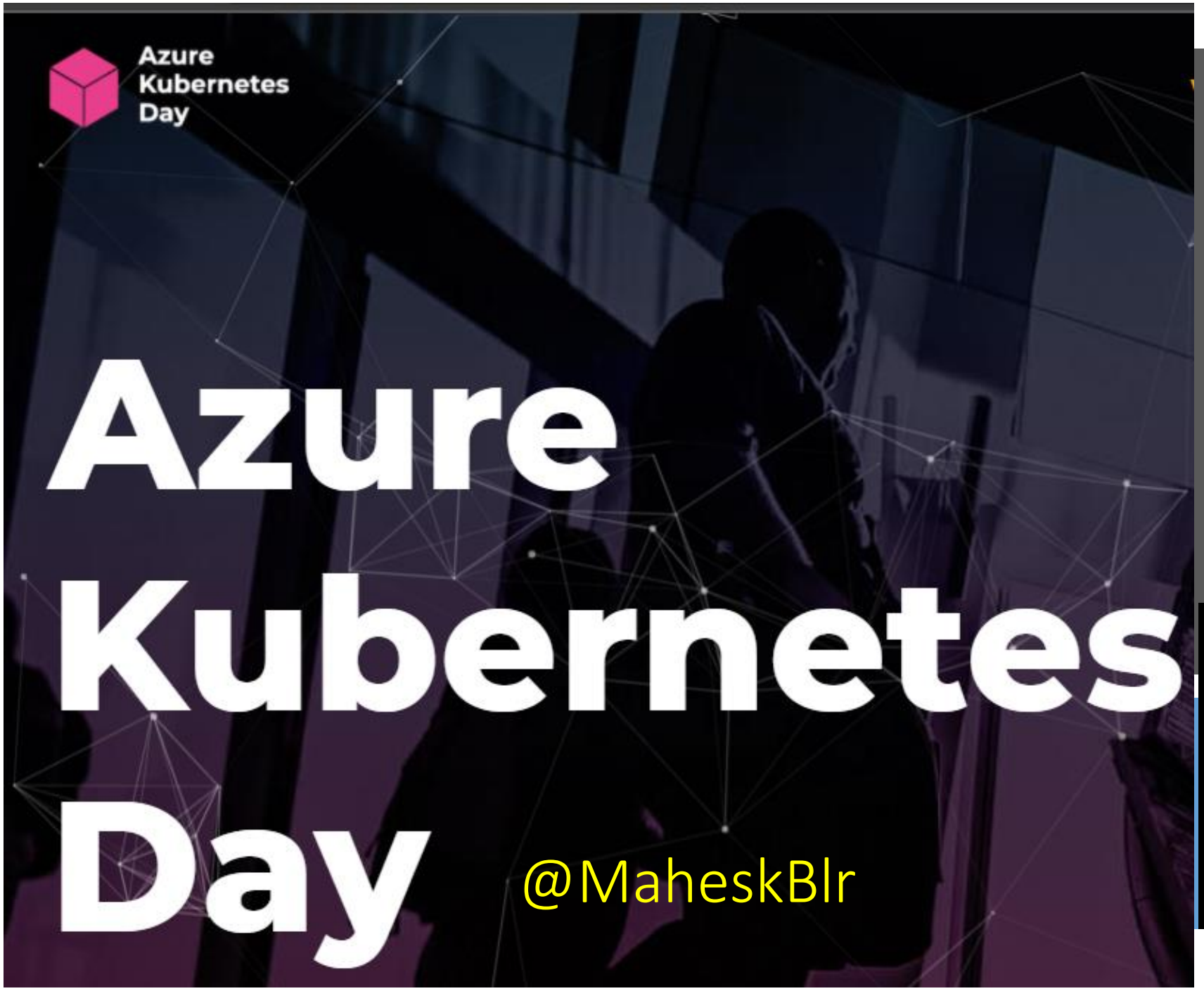


Must know Azure Kubernetes Best practices and features for better resiliency

Overview

AKS specific best practices after working with multiple customers

Mostly about Day-2 challenges and solve

Upcoming features, SLA's, Node pools, Availability Zones – for maximum resiliency

What's Your Kubernetes Maturity?

https://www.cncf.io/blog/2021/01/12/whats-your-kubernetes-maturity/

1. Multi-tenancy

- **Namespace** - logical isolation boundary
- **Scheduling** - use resource quotas, pdb's, advanced features like taints and tolerations, node selectors, node and pod affinity or anti-affinity
- **Networking** - use network policies to control the flow of traffic in and out of pods
- **Auth and Authorization** – use of RBAC and AAD, Pod Identies and Azure KeyValut
- **Containers** – Azure Policy Add-on to enforce pod security, security contexts, scanning images.

2. Enforce Resource Quota

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: dev-app-team
spec:
  hard:
    cpu: "10"
    memory: 20Gi
    pods: "10"
```

$ kubectl apply -f dev-app-team-quotas.yaml --namespace dev-apps

Best practice guidance - Plan and apply resource quotas at the namespace level. If pods don't define resource requests and limits, reject the deployment. Monitor resource usage and adjust quotas as needed.

3. Use Pod Distruption Budget (PDB's)

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: nginx-pdb
spec:
  minAvailable: 3
  selector:
   matchLabels:
     app: nginx-frontend
```

$ kubectl apply -f nginx-pdb.yaml

Best practice guidance - To maintain the availability of applications, define Pod Disruption Budgets (PDBs) to make sure that a minimum number of pods are available in the cluster.

4. Use Node Affinity, Inter-pod affinity and Anti-affinity

YAML

```yaml
kind: Pod
apiVersion: v1
metadata:
  name: tf-mnist
spec:
  containers:
  - name: tf-mnist
    image: mcr.microsoft.com/azuredocs/samples-tf-mnist-demo:gpu
    resources:
      requests:
        cpu: 0.5
        memory: 2Gi
      limits:
        cpu: 4.0
        memory: 16Gi
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: hardware
            operator: In
            values: highmem
```

Best practice guidance:

Control the scheduling of pods on nodes using node selectors, node affinity, or inter-pod affinity.

These settings allow the Kubernetes scheduler to logically isolate workloads, such as by hardware in the node.

Node 1	Node 2	Node 3
webapp-1	webapp-2	webapp-3
cache-1	cache-2	cache-3

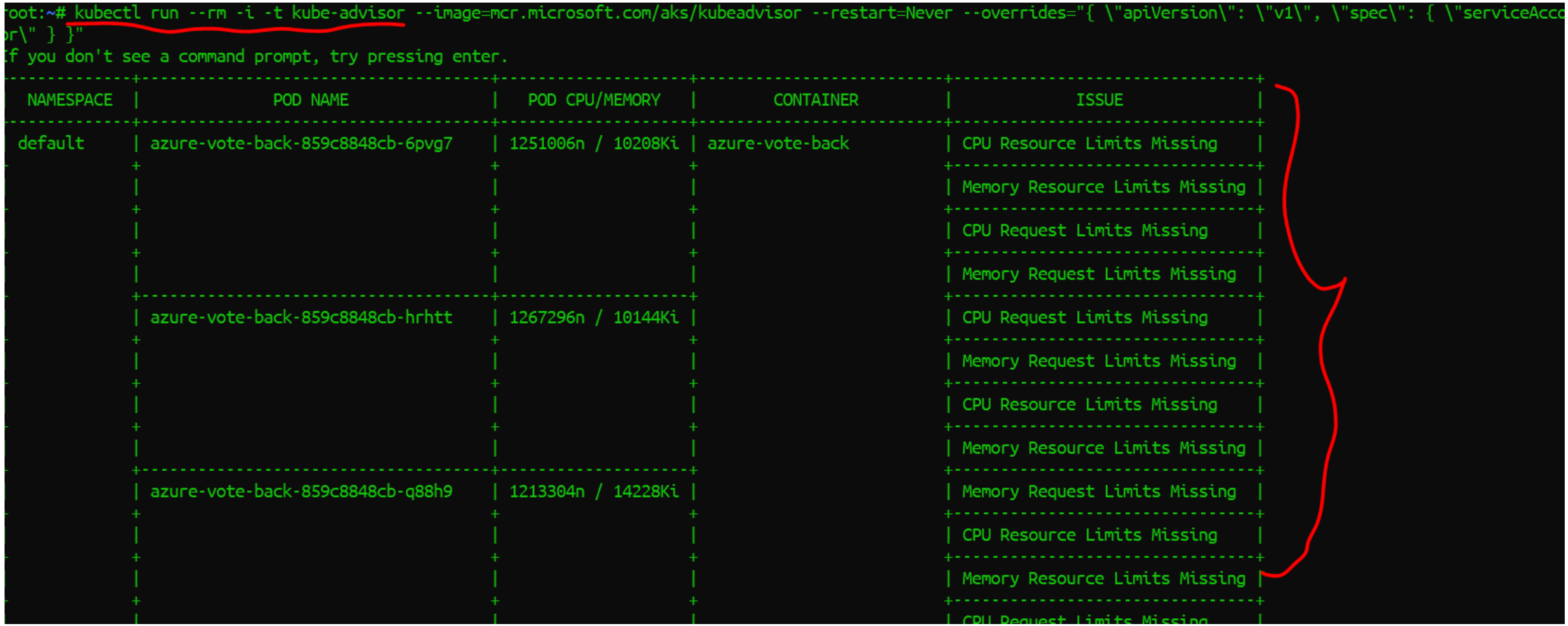
5. Use Kube-Advisor

Scans a cluster and reports on issues that it finds

Say, identify pods that don't have resource requests and limits in place

Best practice guidance

Regularly run the latest version of kube-advisor open source tool to detect issues in your cluster. If you apply resource quotas on an existing AKS cluster, run kube-advisor first to find pods that don't have resource requests and limits defined.



https://github.com/Azure/kube-advisor

6. AKS - Uptime SLA

Uptime SLA is an optional feature to enable a financially backed, higher SLA for a cluster.

99.95% of K8s API server endpoint for clusters that -> AZ

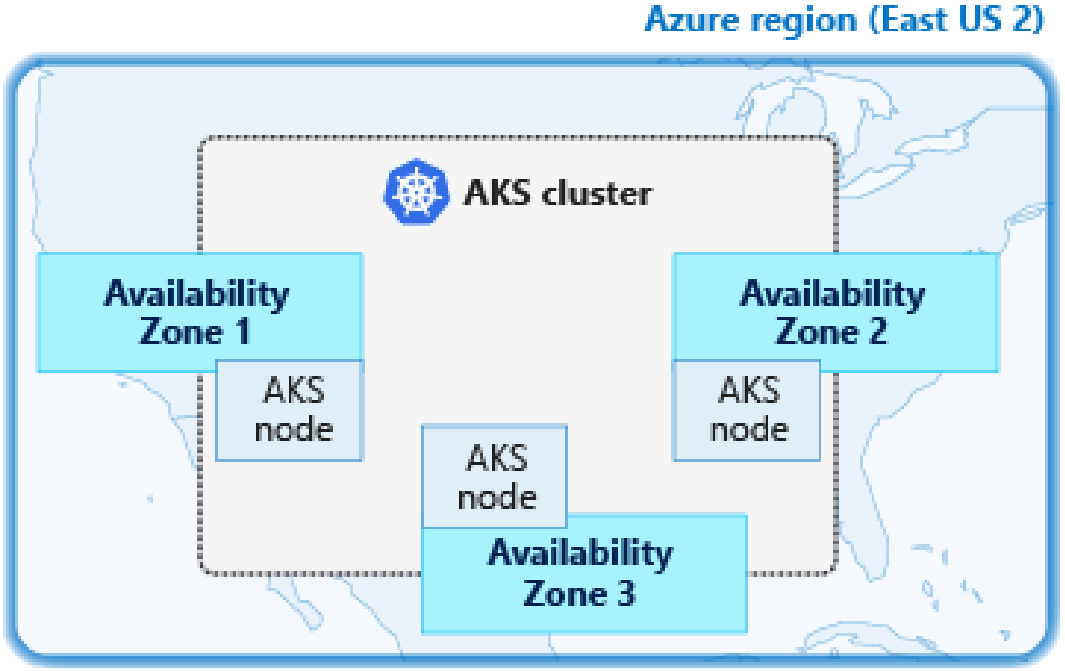
99.9% of availability for clusters that *don't use AZ.*

AKS uses master node replicas across update and fault domains to ensure SLA requirements are met.

7. Create an AKS cluster across availability zones

```
az group create --name myResourceGroup --location eastus2
```

```
az aks create \
    --resource-group myResourceGroup \
    --name myAKSCluster \
    --generate-ssh-keys \
    --vm-set-type VirtualMachineScaleSets \
    --load-balancer-sku standard \
    --node-count 3 \
    --zones 1 2 3
```

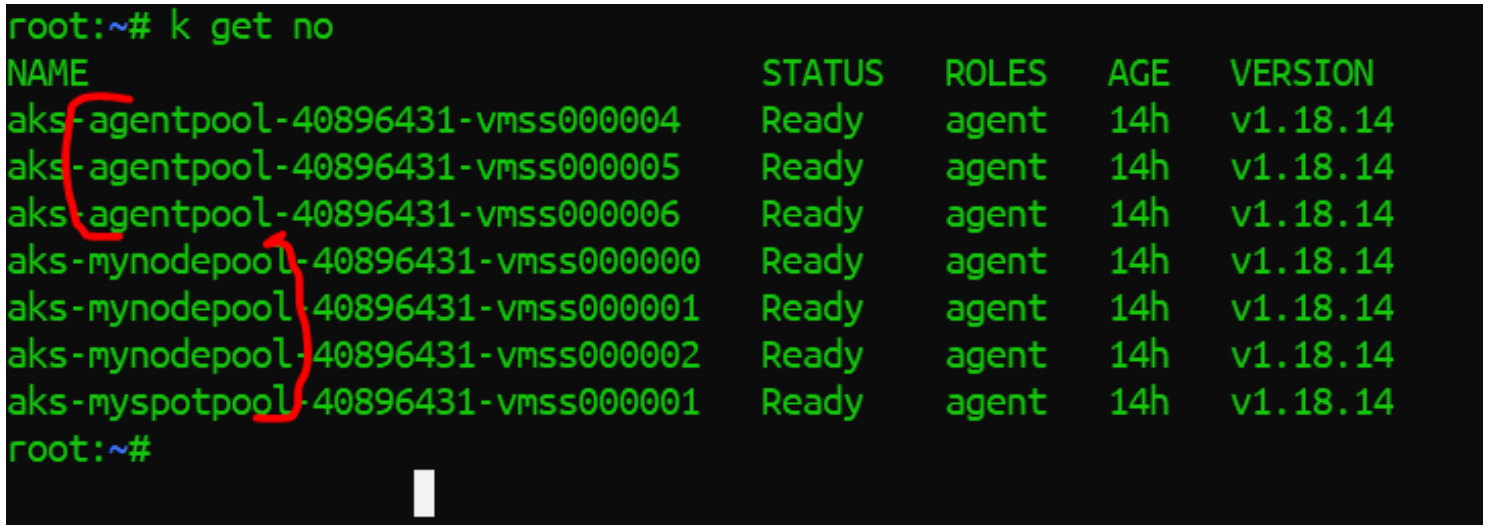



```
root:~# kubectl get nodes -o custom-columns=NAME:'{.metadata.name}',REGION:'{.metadata.labels.topology\.kubernetes\.io/region}',ZONE:'{metadata.labels.topology\.kubernetes\.io/zone}'
NAME                                REGION          ZONE
aks-agentpool-40896431-vmss000004   southeastasia   southeastasia-1
aks-agentpool-40896431-vmss000005   southeastasia   southeastasia-2
aks-agentpool-40896431-vmss000006   southeastasia   southeastasia-3
aks-mynodepool-40896431-vmss000000  southeastasia   0
aks-mynodepool-40896431-vmss000001  southeastasia   0
aks-mynodepool-40896431-vmss000002  southeastasia   0
```

8. Have more than 1 Node Pool

```
az aks nodepool add \
    --resource-group aksdayconf-rg \
    --cluster-name OpsTeamAKScluster \
    --name mynodepool \
    --node-count 3
```

```
az aks nodepool list --resource-group aksdayconf-rg --cluster-name OpsTeamAKScluster
```



9. Azure Policy

Continues compliance is must to maintain compliance in a proactive rather reactive approach.

Achieve real-time cloud compliance at scale with consistent resource governance. It has a quite an exhaustive list of policies here

https://github.com/azure/azure-policy

Best part is, we could roll out custom policies on the resources. The rules can be written in a declarative style.

10. Auto Scale Cluster nodes and pods

As demand for resources change, the number of cluster nodes or pods that run your services can automatically scale up or down.

Use both HPA & Cluster Autoscaler approach.

This approach to scaling lets the AKS cluster automatically adjust to demands and only run the resources needed.

```
az aks nodepool add \
    --resource-group aksdayconf-rg \
    --cluster-name OpsTeamAKScluster \
    --name mynodepool \
    --enable-cluster-autoscaler \
    --min-count 5 \
    --max-count 10 \
    --no-wait
```

11. Start and Stop AKS Cluster

1) az extension add --name aks-preview
2) az extension update --name aks-preview

3) az feature register --namespace "Microsoft.ContainerService" --name "StartStopPreview“

4) az feature list -o table --query "[?contains(name, 'Microsoft.ContainerService/StartStopPreview')].{Name:name,State:properties.state}“

5) az provider register --namespace Microsoft.ContainerService

6) az aks stop --name OpsTeamAKScluster --resource-group aksdayconf-rg

7) az aks start --name OpsTeamAKScluster --resource-group aksdayconf-rg

https://docs.microsoft.com/en-us/azure/aks/start-stop-cluster

** Preserve for 12 months, supports only VMSS*

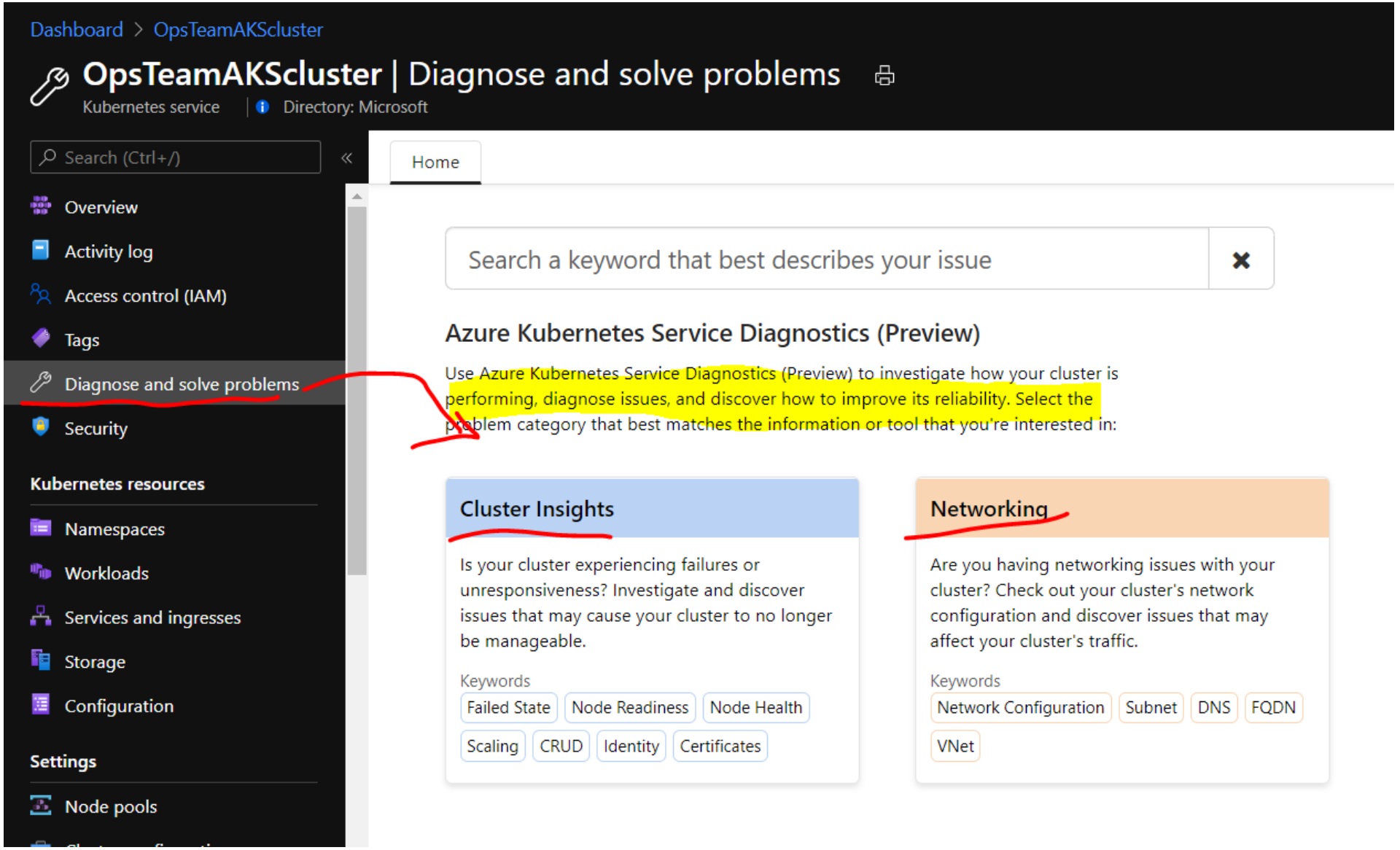
12. AKS Cluster Capacity Planning

1. How many nodes do I need in my AKS cluster?
2. Does the size of the subnet of my nodes matter?
3. How many pods could be run on the cluster?

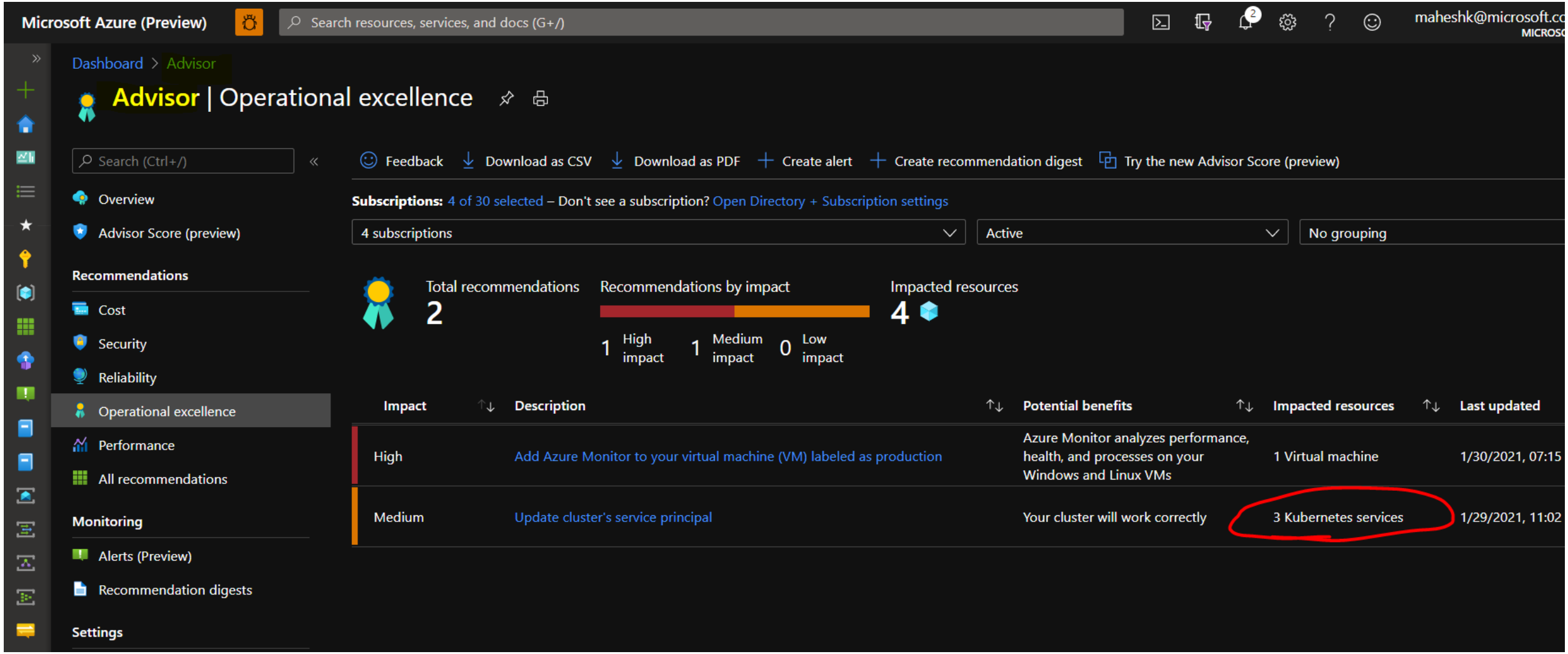
	A	B	C	D	E
1	**Max pods per node**	30			
2					
3			=B1		=A14+B14+D14+1+(C14*(D14+1))
4					
5					
6	**IP reserved per subnet in Azure**	**IP AKS private API endpoint**	**Max pods per node**	**Nodes in cluster**	**IP reserved by Azure CNI**
7	5	1	30	1	68
8	5	1	30	2	99
9	5	1	30	3	130
10	5	1	30	4	161
11	5	1	30	5	192
12	5	1	30	6	223
13					

https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/azure-kubernetes-service-cluster-capacity-planning/ba-p/1474990

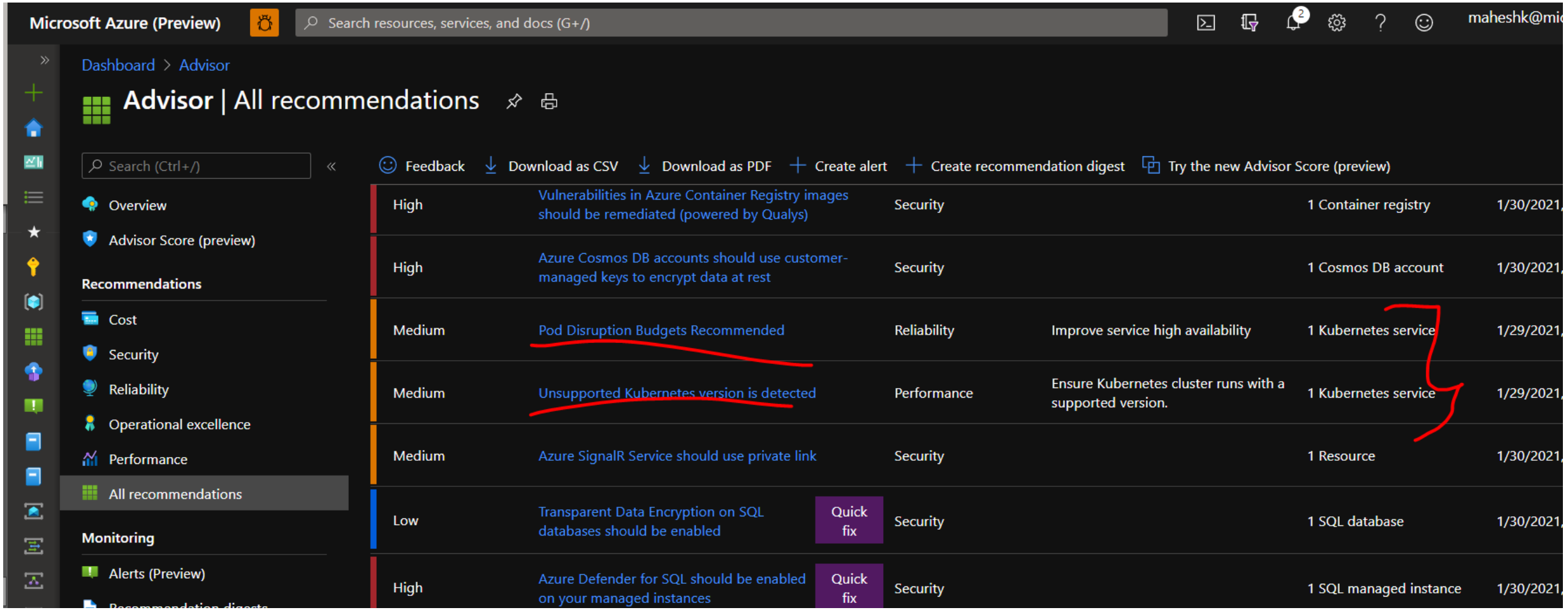
13. Use AKS Diagnostics



14. Use Azure Advisor

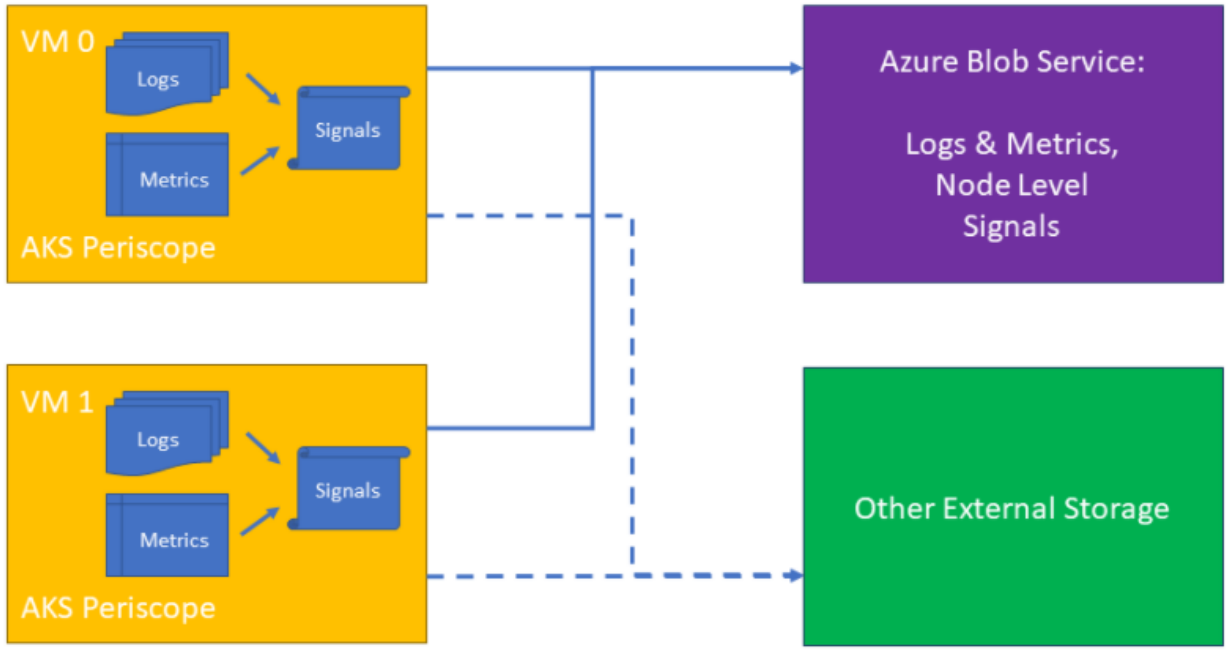


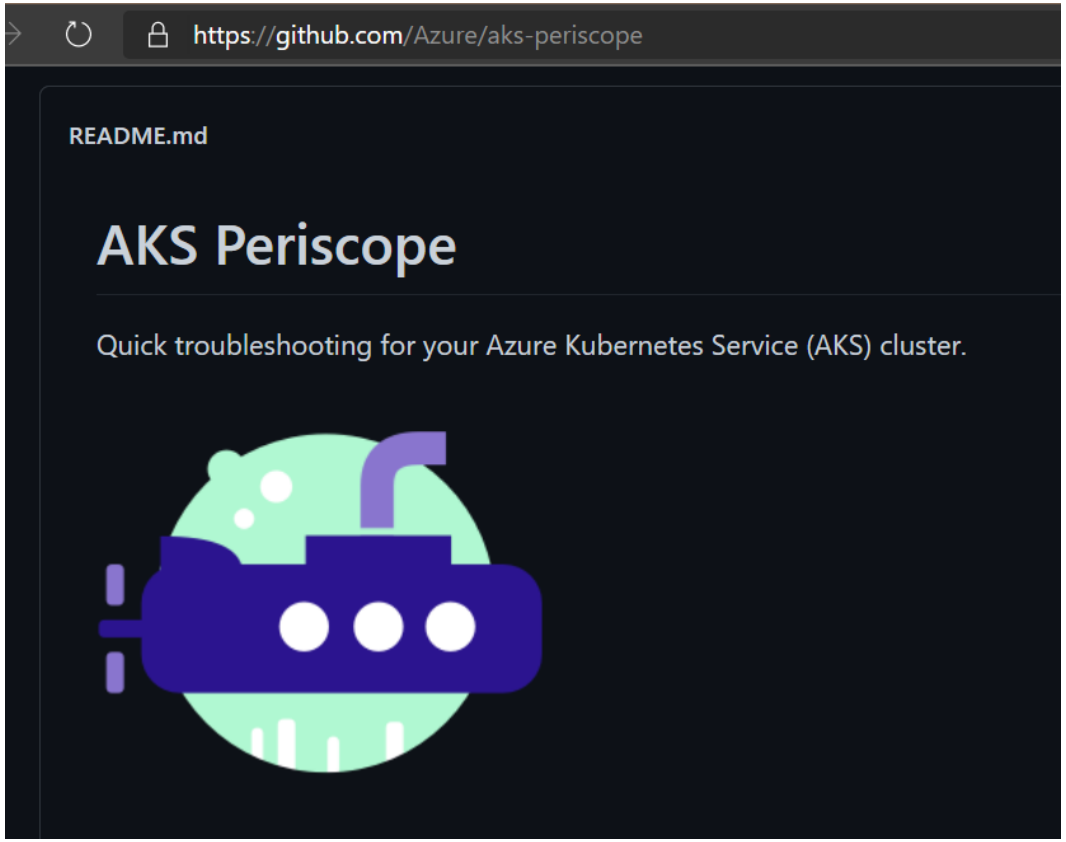
14.1 Use Azure Advisor



15. Use Azure Periscope

when things do go wrong, AKS customers need a tool to help them diagnose and collect the logs necessary to troubleshoot the issue.



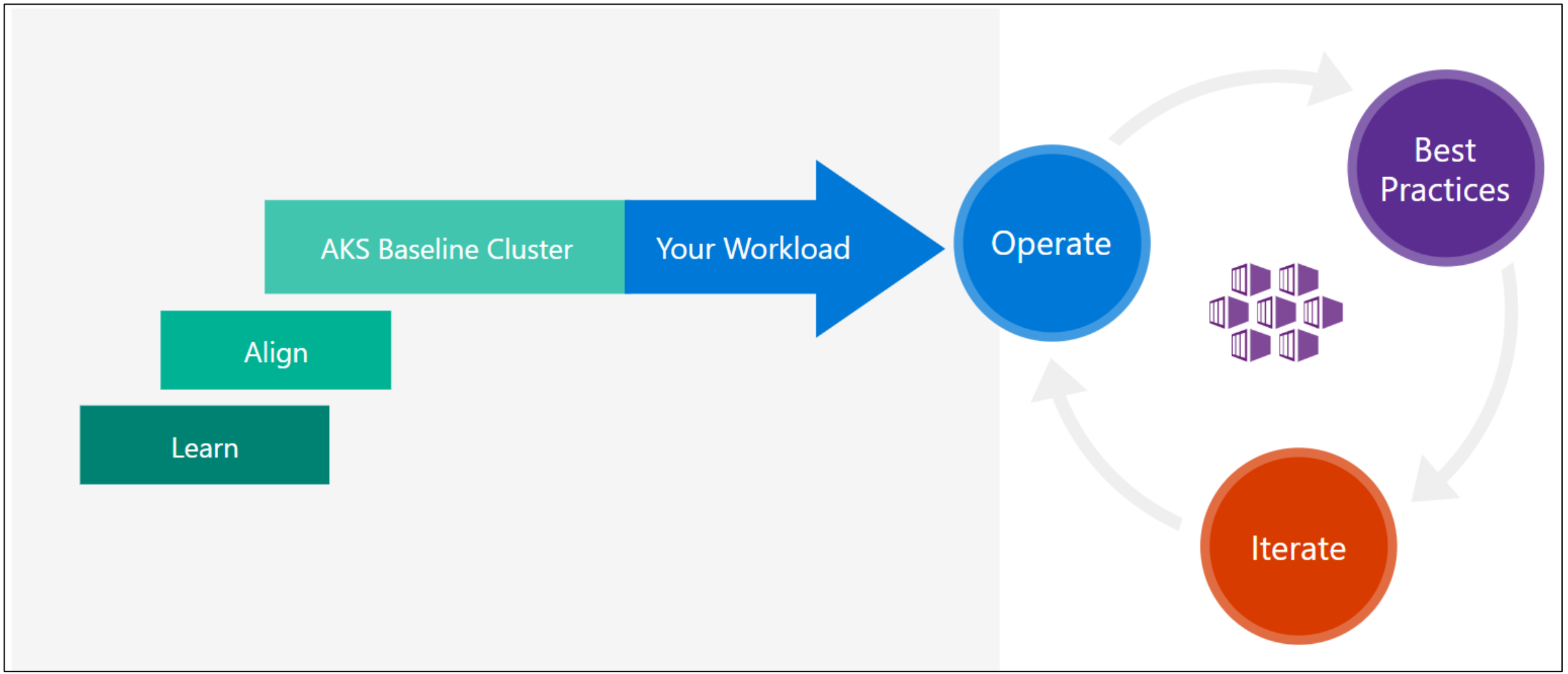


https://github.com/Azure/aks-periscope

16. Production Checklist

1. Regions - Select the region based on your compliance requirement – You cannot change later
2. Version – Select the most stable version for production
3. Use Node Pools and Az Zones – minimum of 2 pods and use AZ
4. Services - recommend using Ingress rather than exposing all of them as Load Balancer
5. **VM Type** – Select appropriate VM type – you can only add new node pools but cannot change types
6. **Max Pods in Cluster, Max Pods in Node, Pod request (CPU/Memory), Pod limits (CPU/Memory)**
7. Networking : Recommend Azure CNI instead Kubenet (Unless org has a restriction on IP Addr to be assigned to the subnet)
8. API Server Access – restrict via IP Whitelisting; Storage and Databases – use managed/PaaS as much as possible
9. Monitor – Use Prometheus, Filebeat or Azure Monitor (easy to implement)
10. Node restarts – recommend **Kured** for automating node reboots after OS Patching

Azure Kubernetes Service solution journey



AKS DevOps must links

- AKS Current preview features: https://aka.ms/aks/preview-features

- AKS Release notes: https://aka.ms/aks/releasenotes

- AKS Public roadmap: http://aka.ms/aks/roadmap

- AKS Known-issues: https://aka.ms/aks/knownissues

- AKS Feature Requests: https://aka.ms/aks/feature-requests

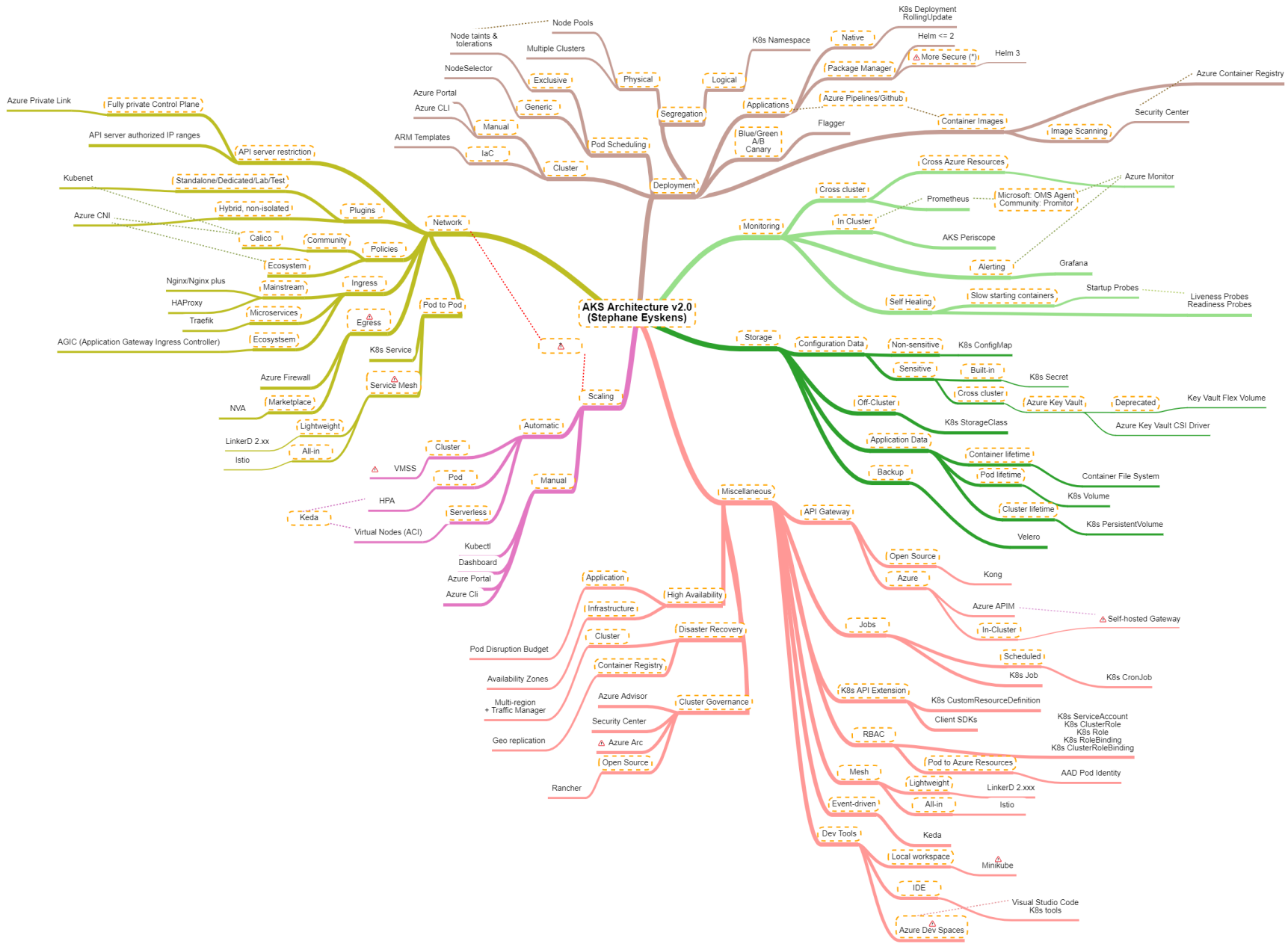
- AKS Public FAQ: https://aka.ms/aks/public-faq

https://www.the-aks-checklist.com/

Q&A - Thank you

MahesKBlr

https://www.linkedin.com/in/mfcmahesh/ Maheshk@microsoft.com


AKS Architecture v2.0
(Stephane Eyskens)
Network
Azure Private Link
Fully private Control Plane
API server authorized IP ranges
API server restriction
Kubenet
Standalone/Dedicated/Lab/Test
Azure CNI
Hybrid, non-isolated
Plugins
Calico
Community
Policies
Ecosystem
Nginx/Nginx plus
Mainstream
Ingress
HAProxy
Microservices
Traefik
AGIC (Application Gateway Ingress Controller)
Ecosystsem
Azure Firewall
Egress
NVA
Marketplace
Pod to Pod
K8s Service
Service Mesh
LinkerD 2.xx
Lightweight
Istio
All-in
Deployment
Segregation
Physical
Node Pools
Node taints & tolerations
Multiple Clusters
NodeSelector
Exclusive
Azure Portal
Azure CLI
Generic
Manual
ARM Templates
IaC
Pod Scheduling
Cluster
Logical
K8s Namespace
Applications
Native
K8s Deployment RollingUpdate
Package Manager
Helm <= 2
More Secure (*)
Helm 3
Azure Pipelines/Github
Container Images
Azure Container Registry
Image Scanning
Security Center
Blue/Green A/B Canary
Flagger
Monitoring
Cross cluster
Cross Azure Resources
Azure Monitor
Prometheus
Microsoft: OMS Agent
Community: Promitor
In Cluster
AKS Periscope
Alerting
Grafana
Self Healing
Slow starting containers
Startup Probes
Liveness Probes
Readiness Probes
Storage
Configuration Data
Non-sensitive
K8s ConfigMap
Sensitive
Built-in
K8s Secret
Cross cluster
Azure Key Vault
Deprecated
Key Vault Flex Volume
Azure Key Vault CSI Driver
Off-Cluster
K8s StorageClass
Application Data
Container lifetime
Pod lifetime
Container File System
K8s Volume
Backup
Cluster lifetime
K8s PersistentVolume
Velero
Scaling
Automatic
Cluster
VMSS
Pod
HPA
Keda
Serverless
Virtual Nodes (ACI)
Manual
Kubectl
Dashboard
Azure Portal
Azure Cli
Miscellaneous
High Availability
Application
Infrastructure
Pod Disruption Budget
Disaster Recovery
Cluster
Availability Zones
Container Registry
Multi-region + Traffic Manager
Cluster Governance
Azure Advisor
Security Center
Azure Arc
Geo replication
Open Source
Rancher
API Gateway
Open Source
Kong
Azure
Azure APIM
Self-hosted Gateway
In-Cluster
Jobs
Scheduled
K8s Job
K8s CronJob
K8s API Extension
K8s CustomResourceDefinition
Client SDKs
RBAC
K8s ServiceAccount
K8s ClusterRole
K8s Role
K8s RoleBinding
K8s ClusterRoleBinding
Pod to Azure Resources
AAD Pod Identity
Mesh
Lightweight
LinkerD 2.xxx
All-in
Istio
Event-driven
Keda
Dev Tools
Local workspace
Minikube
IDE
Visual Studio Code
K8s tools
Azure Dev Spaces

```
oot:~/.kube# k get no
AME                                STATUS   ROLES   AGE   VERSION
ks-agentpool-40896431-vmss000002   Ready    agent   11m   v1.18.14
ks-agentpool-40896431-vmss000003   Ready    agent   11m   v1.18.14
oot:~/.kube# k get po
AME                                READY   STATUS    RESTARTS   AGE
zure-vote-back-859c8848cb-7lk8t    1/1     Running   0          12m
zure-vote-back-859c8848cb-p7lmc    1/1     Running   0          12m
zure-vote-back-859c8848cb-zr2s5    1/1     Running   0          12m
zure-vote-front-5f55f4d7f8-95ksg   1/1     Running   0          12m
zure-vote-front-5f55f4d7f8-gwbfc   1/1     Running   0          12m
zure-vote-front-5f55f4d7f8-pg6wp   1/1     Running   0          12m
oot:~/.kube# az aks update --resource-group aksdayconf-rg --name OpsTeamAKScluster --enable-cluster-autoscal
r --min-count 1 --max-count 5
he behavior of this command has been altered by the following extension: aks-preview
AD role propagation done[##############################################]  100.0000%{
  "aadProfile": null,
  "addonProfiles": {
    "azurepolicy": {
      "config": {
        "version": "v2"
      },
      "enabled": true,
      "identity": {
        "clientId": "7594f2ba-bccd-4358-9fb6-ada706722018",
        "objectId": "b352a972-a1ce-4528-b62d-6299aac3e51c",
        "resourceId": "/subscriptions/38e1b8c4-c5bc-4dd5-a7e0-e909b45f4fad/resourcegroups/MC_aksdayconf-rg_Op
TeamAKScluster_southeastasia/providers/Microsoft.ManagedIdentity/userAssignedIdentities/azurepolicy-opsteama
scluster"
      }
```

Increase your application availability with pod anti-affinity settings in Azure Kubernetes Service

https://www.danielstechblog.io/increase-your-application-availability-with-pod-anti-affinity-settings-in-azure-kubernetes-service/

VERTICAL POD AUTOSCALING: THE DEFINITIVE GUIDE

https://povilasv.me/vertical-pod-autoscaling-the-definitive-guide/

Kubernetes Networking

https://dominik-tornow.medium.com/kubernetes-networking-22ea81af44d0

A Guide to the Kubernetes Networking Model

https://sookocheff.com/post/kubernetes/understanding-kubernetes-networking-model/

we'll build a baseline infrastructure that deploys an Azure Kubernetes Service (AKS) cluster. This article includes recommendations for networking, security, identity, management, and monitoring of the cluster based on an organization's business requirements.

- https://github.com/mspnp/aks-secure-baseline