Table of Contents

[Reference Architecture 3](#_Toc71710032)

[AKS Production checklist 3](#_Toc71710033)

[Secure Baseline Architecture 3](#_Toc71710034)

[Github Samples 3](#_Toc71710035)

[Links 3](#_Toc71710036)

[Windows Container and Private cluster Announcements 5](#_Toc71710037)

[Videos and YouTube Channels 5](#_Toc71710038)

[Hands-On 5](#_Toc71710039)

[Istio Support? 5](#_Toc71710040)

[How to deploy Nginx controller without HELM 5](#_Toc71710041)

[App Modernization on AKS – A practical approach 6](#_Toc71710042)

[How can I configure overprovisioning with Cluster Autoscaler? 6](#_Toc71710043)

[Containers extension in Windows Admin Center 6](#_Toc71710044)

[AKS Concepts 7](#_Toc71710045)

[What is a service?​ 7](#_Toc71710046)

[What is a Ingress?​ 7](#_Toc71710047)

[What is a Ingress Controller​ 7](#_Toc71710048)

[How does Ingress Controller Work? 7](#_Toc71710049)

[​What is Application Gateway Ingress Controller (AGIC)? 7](#_Toc71710050)

[AKS for Windows 8](#_Toc71710051)

[How do you select values for the horizontal pod autoscaler? 9](#_Toc71710052)

[How do you create and use Kubernetes custom resources? 10](#_Toc71710053)

[SQL Server container for Windows 10](#_Toc71710054)

[Persistent volume claims 11](#_Toc71710055)

[How to configure app gateway through helm? 13](#_Toc71710056)

[How does linux/system node pool not comingle? 13](#_Toc71710057)

[What is the resilience of the AKS cluster? At the region level? Across availability zones? Should we consider federating two clusters and route traffic to each? 14](#_Toc71710058)

[Can a cluster be for multiple regions? 14](#_Toc71710059)

[Can I join an on prem node to an AKS cluster? 15](#_Toc71710060)

[How does istio work? 15](#_Toc71710061)

[Istio provides the following set of capabilities 15](#_Toc71710062)

[Istio is well suited to and suggested for the following scenarios 16](#_Toc71710063)

[What is our process for draining nodes? Should it be a rundeck job? Should it be regular? How does the cluster remain balanced? 16](#_Toc71710064)

[Do we need to switch NUA01 to the managed identity cluster? How do we migrate? 17](#_Toc71710065)

[Horizontal pod autoscaler 17](#_Toc71710066)

[Cluster autoscaler 17](#_Toc71710067)

[How does patching occur on the node? 18](#_Toc71710068)

[Should we be running an audit tool as a part of our pipeline? (k8guard, copper, kube-bench)? 19](#_Toc71710069)

[Where do I find audit logging for the AKS cluster? 19](#_Toc71710070)

[Should we use kubectl diff on deployments to see what changed? As a log? 19](#_Toc71710071)

[Where does kubectl run from? Azure console? My desktop with VPN? (it's a context) 20](#_Toc71710072)

[What is the RBAC for kubectl 20](#_Toc71710073)

[How do we use tags? 20](#_Toc71710074)

[Container hardening 20](#_Toc71710075)

[How to allow TLS pass thru from AGIC to pod level? 20](#_Toc71710076)

[Singleton Pods 21](#_Toc71710077)

# Reference Architecture

<https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-microservices/aks-microservices>

# AKS Production checklist

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

# Secure Baseline Architecture

<https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks/secure-baseline-aks>

# Github Samples

<https://github.com/Azure/sg-aks-workshop>

<https://github.com/azure/aks-production-patterns>

<https://github.com/retaildevcrews/secure-aks-infra>

# Links

FAQ - <https://docs.microsoft.com/azure/aks/faq>

Kubernetes Best Practices – <https://aka.ms/aks/bestpractices>​

Reference Architecture - <https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/aks-api-first>

Taint and Toleration - <https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-advanced-scheduler>

Headless service - <https://itnext.io/how-to-access-kubernetes-applications-using-services-c437a4c940ad>

SQL Server in container - <https://docs.microsoft.com/en-us/sql/linux/tutorial-sql-server-containers-kubernetes?view=sql-server-ver15>

HPA Scale - <https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-scale>

Cluster Scale - <https://docs.microsoft.com/en-us/azure/aks/cluster-autoscaler>

Network policies - <https://docs.microsoft.com/en-us/azure/aks/use-network-policies>

Network concepts - <https://docs.microsoft.com/en-us/azure/aks/concepts-network>

API Managements and AKS - <https://docs.microsoft.com/en-us/azure/api-management/api-management-kubernetes>

Application gateway Ingress controller - <https://github.com/Azure/application-gateway-kubernetes-ingress>

AGIC - <https://roykim.ca/2020/02/09/understanding-ingress-controllers-and-azure-app-gateway-for-azure-kubernetes-part-1-intro/>

APIM - <https://pumpingco.de/blog/control-ingress-to-aks-with-azure-api-management/>

Continuous Deployment - <https://azure.github.io/application-gateway-kubernetes-ingress/how-tos/continuous-deployment/>

Image and container level security

<https://docs.microsoft.com/en-us/azure/security-center/azure-container-registry-integration>

Node and cluster level security

Use AAD integrated clusters with Kubernetes RBAC enabled

<https://docs.microsoft.com/en-us/azure/aks/managed-aad>

Use private clusters with egress through a NVA

<https://docs.microsoft.com/en-us/azure/aks/private-clusters>

Deploy kured to perform automatic reboots if needed as a result of daily automatic security patching

<https://docs.microsoft.com/en-us/azure/aks/node-updates-kured>

Disable the dashboard

<https://docs.microsoft.com/en-us/azure/aks/kubernetes-dashboard>

Use Azure Security Center with AKS

[https://docs.microsoft.com/en-us/azure/security-center/recommendations-reference#recs-containers](https://docs.microsoft.com/en-us/azure/security-center/recommendations-reference)

Use Azure Policy (public preview) with AKS

<https://docs.microsoft.com/en-us/azure/governance/policy/concepts/policy-for-kubernetes>

Pod level security

Use AAD Pod Identity

<https://github.com/Azure/aad-pod-identity>

Workload level security

Use Kubernetes namespaces

Use network policy to control east-west traffic in the cluster

<https://docs.microsoft.com/en-us/azure/aks/use-network-policies>

Use the Secrets Store CSI driver to mount secrets into Kubernetes pods

<https://docs.microsoft.com/en-us/azure/key-vault/general/key-vault-integrate-kubernetes>

# Windows Container and Private cluster Announcements

<https://azure.microsoft.com/en-us/blog/announcing-the-general-availability-of-windows-server-containers-and-private-clusters-for-azure-kubernetes-service/>

# Videos and YouTube Channels

Lachlan Evenson (aka Crocodile Dundee Of Kubernetes) [https://www.youtube.com/channel/UCC5NsnXM2lE6kKfJKdQgsRQ](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.youtube.com%2Fchannel%2FUCC5NsnXM2lE6kKfJKdQgsRQ&data=01%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C0dae892217c94734bb2108d6d7bb5af7%7C72f988bf86f141af91ab2d7cd011db47%7C1&sdata=fs6Dz3PLBk%2B9q%2BVXJIGLQ8267ouqatb5eZbEwpWt%2BJg%3D&reserved=0)

TGI Kubernetes by Heptio - [https://www.youtube.com/channel/UCjQU5ZI2mHswy7OOsii\_URg/videos](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.youtube.com%2Fchannel%2FUCjQU5ZI2mHswy7OOsii_URg%2Fvideos&data=01%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C0dae892217c94734bb2108d6d7bb5af7%7C72f988bf86f141af91ab2d7cd011db47%7C1&sdata=2yff%2BJhrO1nk%2B5hic2Pnf9MExSs0bNSPF2vGYw%2BO8I4%3D&reserved=0)

Linux Foundation CKA (Must Register With MSFT For Free Access) [https://lms.quickstart.com](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Flms.quickstart.com&data=01%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C0dae892217c94734bb2108d6d7bb5af7%7C72f988bf86f141af91ab2d7cd011db47%7C1&sdata=iEoMgP0CDIcFskYZosOayRt2mSxS2cbzT5kX%2BMmxGZc%3D&reserved=0)

# Hands-On

Kubernetes BootCamp - [https://kubernetesbootcamp.github.io/kubernetes-bootcamp/index.html](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fkubernetesbootcamp.github.io%2Fkubernetes-bootcamp%2Findex.html&data=01%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C0dae892217c94734bb2108d6d7bb5af7%7C72f988bf86f141af91ab2d7cd011db47%7C1&sdata=EaWm81bgJF3CYEpaoOLqVZLk40GxaXsEcB0PN4dBqmw%3D&reserved=0)

Katacoda Labs -[https://katacoda.com/courses/kubernetes](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fkatacoda.com%2Fcourses%2Fkubernetes&data=01%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C0dae892217c94734bb2108d6d7bb5af7%7C72f988bf86f141af91ab2d7cd011db47%7C1&sdata=0BzcVwL1dsSU1fHK2osK%2Fpcn3iLydx4c6YzbRJ3OgqU%3D&reserved=0)

Azure Citadel - [https://azurecitadel.github.io/](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fazurecitadel.github.io%2F&data=01%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C0dae892217c94734bb2108d6d7bb5af7%7C72f988bf86f141af91ab2d7cd011db47%7C1&sdata=CHfsw%2Fpc5WDqe11RstnG9VNeK0%2FL9KWS5zCheApeMX0%3D&reserved=0)

# Istio Support?

Helm templates for Istio deployment to the AKS clusters.

[https://istio.io/docs/setup/kubernetes/platform-setup/azure/](https://istio.io/docs/setup/kubernetes/platform-setup/azure/%20)

<https://docs.microsoft.com/en-us/azure/aks/servicemesh-istio-install?pivots=client-operating-system-windows>

# How to deploy Nginx controller without HELM

<https://dzone.com/articles/nginx-ingress-controller-configuration-in-aks>

<https://cloudblogs.microsoft.com/opensource/2018/11/27/tutorial-azure-devops-setup-cicd-pipeline-kubernetes-docker-helm/>

# App Modernization on AKS – A practical approach

<https://aka.ms/EVT504AL-OnDemand>

<https://azure.microsoft.com/en-us/resources/videos/azure-friday-intro-to-rudr-a-kubernetes-implementation-of-the-open-application-model/>

# How can I configure overprovisioning with Cluster Autoscaler?

<https://github.com/kubernetes/autoscaler/blob/master/cluster-autoscaler/FAQ.md#how-can-i-configure-overprovisioning-with-cluster-autoscaler>

# [Containers extension in Windows Admin Center](https://techcommunity.microsoft.com/t5/containers/announcement-new-updates-to-the-containers-extension-in-windows/ba-p/1449476)

A screenshot of a social media post

Description automatically generated

The above screenshot is from the Containers extension on Windows Admin Center. With this tool, customers can containerize their applications without having to write the docker file, and then are able to push the image to an external registry, such as ACR.

We have some details on the tool [here](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Ftechcommunity.microsoft.com%2Ft5%2Fcontainers%2Fannouncement-new-updates-to-the-containers-extension-in-windows%2Fba-p%2F1449476&data=02%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C61ce5c738b1e4d02e04108d8462eaef0%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637336513626533799&sdata=1oyKM3nuUuqA5xkZgR12slNQMuOW1VCqNzUqKqwXGi4%3D&reserved=0). We recently released an update to support [Web Deploy](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Ftechcommunity.microsoft.com%2Ft5%2Fcontainers%2Fannouncement-adding-webdeploy-support-to-container-image%2Fba-p%2F1579144&data=02%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C61ce5c738b1e4d02e04108d8462eaef0%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637336513626533799&sdata=S57o7JdmjjvqCjLoVUh8jRGZos8unuR5BP3DtY5I%2FvQ%3D&reserved=0). I also started a video series to show the end to end experience from WS2012R2 on-prem to AKS. Video 1 is available [here](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Ftechcommunity.microsoft.com%2Ft5%2Fitops-talk-blog%2Fjourney-of-an-app-how-to-extract-a-web-app-from-iis-to-create-a%2Fba-p%2F1590189&data=02%7C01%7CAkshay.Nimbalkar%40microsoft.com%7C61ce5c738b1e4d02e04108d8462eaef0%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637336513626543793&sdata=kAd5%2Fe7MoRgrVk3%2FxCZZBOcfMV7ua2GOgWUhUirAraI%3D&reserved=0).

AKS Concepts

## What is a service?​

Logical set of Pods and a policy by which to access them. The set of Pods targeted by a Service is (usually) determined by a Label Selector.​ Services provide important features that are standardized across the cluster: load-balancing (L4), service discovery between applications, and features to support zero-downtime application deployments.​

## What is a Ingress?​

A Kubernetes API that manages external access to the services in the cluster​

* Supports HTTP and HTTPs​
* Path and Subdomain based routing​
* SSL Termination​
* Save on Public IPs​

## What is a Ingress Controller​

A daemon, deployed as a Kubernetes Pod, that watches the Ingress Endpoint for updates. Its job is to satisfy requests for ingresses. ​Some popular Ingress Controllers include nginx, Traefik and HAProxy.​

## How does Ingress Controller Work?

A screenshot of a cell phone

Description automatically generated

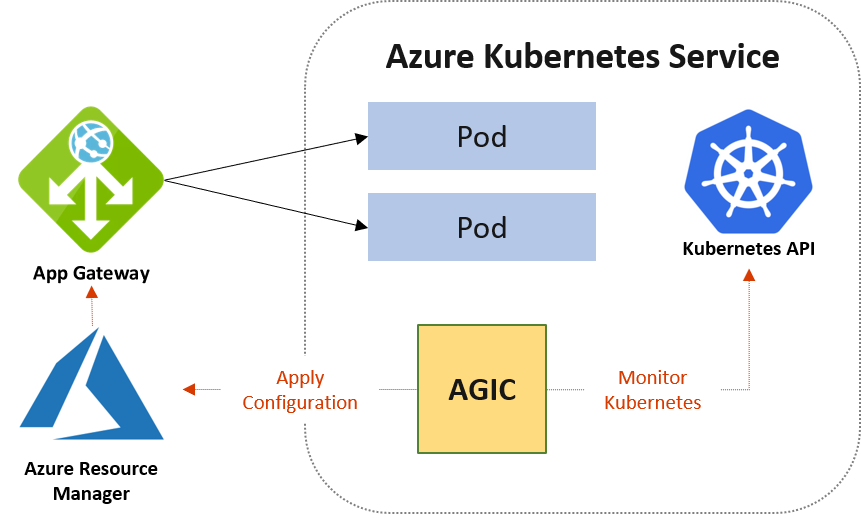
## ​What is Application Gateway Ingress Controller (AGIC)?

Application Gateway Ingress Controller (AGIC) is a Kubernetes application, which makes it possible for [Azure Kubernetes Service (AKS)](https://azure.microsoft.com/en-us/services/kubernetes-service/) customers to leverage Azure's native [Application Gateway](https://azure.microsoft.com/en-us/services/application-gateway/) L7 load-balancer to expose cloud software to the Internet. AGIC monitors the Kubernetes cluster it is hosted on and continuously updates an App Gateway, so that selected services are exposed to the Internet.

The Ingress Controller runs in its own pod on the customer’s AKS. AGIC monitors a subset of Kubernetes Resources for changes. The state of the AKS cluster is translated to App Gateway specific configuration and applied to the [Azure Resource Manager (ARM)](https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview).

AGIC is configured via the Kubernetes [Ingress resource](http://kubernetes.io/docs/user-guide/ingress/), along with Service and Deployments/Pods. It provides a number of features, leveraging Azure’s native App Gateway L7 load balancer. To name a few:

* URL routing
* Cookie-based affinity
* SSL termination
* End-to-end SSL
* Support for public, private, and hybrid web sites
* Integrated web application firewall



## AKS for Windows

Windows Server container support in the Azure Kubernetes Service is now GA. With this preview, you can:

* Lift and shift Windows applications to run on AKS
* Seamlessly manage Windows and Linux applications through a single unified API
* Mix Windows and Linux applications in the same Kubernetes cluster – with consistent monitoring experience and deployment pipelines

Now you can get the best of managed Kubernetes for all of your workloads whether they’re in Windows, Linux or both

References:

<https://azure.microsoft.com/en-us/updates/azure-kubernetes-service-aks-now-supports-windows-server-containers-3/>

<https://docs.microsoft.com/en-us/azure/aks/windows-container-cli>

<https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/>

<https://kubernetes.io/blog/2019/04/01/kubernetes-v1.14-delivers-production-level-support-for-windows-nodes-and-windows-containers/>

## How do you select values for the horizontal pod autoscaler?

When your application has lot of demand you will need to scale-up or down based on different factors like number of requests, or many other specific counters. Of the most interesting feature of a microservice architecture is scaling, that it is more granular than on a classical deployment, providing us the flexibility to scale only the components that are on stress. HPA allows us to optimize how we consume our computational resources.

**HPA Yaml File**

apiVersion: autoscaling/v1

kind: HorizontalPodAutoscaler

metadata:

name: azure-vote-back-hpa

spec:

maxReplicas: 10 # define max replica count

minReplicas: 3 # define min replica count

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: azure-vote-back

targetCPUUtilizationPercentage: 50 # target CPU utilization

apiVersion: autoscaling/v1

kind: HorizontalPodAutoscaler

metadata:

name: azure-vote-front-hpa

spec:

maxReplicas: 10 # define max replica count

minReplicas: 3 # define min replica count

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: azure-vote-front

targetCPUUtilizationPercentage: 50 # target CPU utilization

References:

<https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/microservices/aks#scalability-considerations>

<https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-scale>

<https://docs.microsoft.com/en-us/azure/aks/cluster-autoscaler>

## How do you create and use Kubernetes custom resources?

References:

<https://github.com/Azure/kubernetes-keyvault-flexvol>

<https://github.com/operator-framework/getting-started>

## SQL Server container for Windows

References:

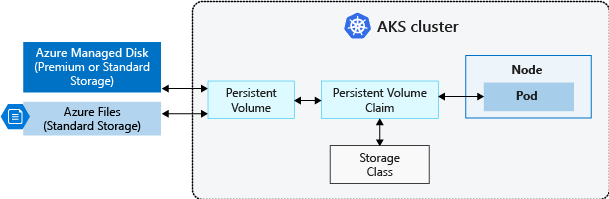
<https://github.com/microsoft/mssql-docker>

<https://hub.docker.com/_/microsoft-mssql-server?tab=description>

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-scale-introduction>

## Persistent volume claims

A PersistentVolumeClaim requests either Disk or File storage of a particular StorageClass, access mode, and size. The Kubernetes API server can dynamically provision the underlying storage resource in Azure if there is no existing resource to fulfill the claim based on the defined StorageClass. The pod definition includes the volume mount once the volume has been connected to the pod.



A PersistentVolume is bound to a PersistentVolumeClaim once an available storage resource has been assigned to the pod requesting it. There is a 1:1 mapping of persistent volumes to claims.

The following example YAML manifest shows a persistent volume claim that uses the managed-premium StorageClass and requests a Disk 5Gi in size:

YAMLCopy

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: azure-managed-disk

spec:

accessModes:

- ReadWriteOnce

storageClassName: managed-premium

resources:

requests:

storage: 5Gi

When you create a pod definition, the persistent volume claim is specified to request the desired storage. You also then specify the volumeMount for your applications to read and write data. The following example YAML manifest shows how the previous persistent volume claim can be used to mount a volume at /mnt/azure:

YAMLCopy

kind: Pod

apiVersion: v1

metadata:

name: nginx

spec:

containers:

- name: myfrontend

image: nginx

volumeMounts:

- mountPath: "/mnt/azure"

name: volume

volumes:

- name: volume

persistentVolumeClaim:

claimName: azure-managed-disk

References:

<https://docs.microsoft.com/en-us/azure/aks/concepts-storage#persistent-volume-claims>

<https://kubernetes.io/docs/concepts/storage/persistent-volumes/>

<https://docs.microsoft.com/en-us/sql/linux/tutorial-sql-server-containers-kubernetes?view=sql-server-ver15>

# How to configure app gateway through helm?

There are two ways to deploy AGIC for your AKS cluster. The first way is through Helm; the second is through AKS as an add-on. The primary benefit of deploying AGIC as an AKS add-on is that it's much simpler than deploying through Helm. For a new setup, you can deploy a new Application Gateway and a new AKS cluster with AGIC enabled as an add-on in one line in Azure CLI. The add-on is also a fully managed service, which provides added benefits such as automatic updates and increased support. Both ways of deploying AGIC (Helm and AKS add-on) are fully supported by Microsoft. Additionally, the add-on allows for better integration with AKS as a first class add-on.

[Helm](https://docs.microsoft.com/en-us/azure/aks/kubernetes-helm) is a package manager for Kubernetes. Leverage it to install the application-gateway-kubernetes-ingress package.

**References:**

1. [How to Install an Application Gateway Ingress Controller (AGIC) Using a New Application Gateway](https://docs.microsoft.com/en-us/azure/application-gateway/ingress-controller-install-new)
2. [Enable Application Gateway Ingress Controller add-on for an existing AKS cluster with an existing Application Gateway](https://docs.microsoft.com/en-us/azure/application-gateway/tutorial-ingress-controller-add-on-existing)
3. <https://azure.github.io/application-gateway-kubernetes-ingress/tutorials/tutorial.general/>

# How does linux/system node pool not comingle?

For a system node pool, AKS automatically assigns the label **kubernetes.azure.com/mode: system** to its nodes. This causes AKS to prefer scheduling system pods on node pools that contain this label. This label does not prevent you from scheduling application pods on system node pools. However, we recommend you isolate critical system pods from your application pods to prevent misconfigured or rogue application pods from accidentally killing system pods. You can enforce this behavior by creating a dedicated system node pool. Use the CriticalAddonsOnly=true:NoSchedule taint to prevent application pods from being scheduled on system node pools. You must have at least two system node pools on your AKS cluster before you can delete one of them.

**References:**

[System and User Node pools](https://docs.microsoft.com/en-us/azure/aks/use-system-pools)

# What is the resilience of the AKS cluster? At the region level? Across availability zones? Should we consider federating two clusters and route traffic to each?

AKS clusters that are deployed using availability zones can distribute nodes across multiple zones within a single region. For example, a cluster in the East US 2 region can create nodes in all three availability zones in East US 2. This distribution of AKS cluster resources improves cluster availability as they're resilient to failure of a specific zone.

**References:**

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

# Can a cluster be for multiple regions?

When you deploy multiple AKS clusters, choose regions where AKS is available, and use paired regions.

An AKS cluster is deployed into a single region. To protect your system from region failure, deploy your application into multiple AKS clusters across different regions. When you plan where to deploy your AKS cluster, consider:

* AKS region availability: Choose regions close to your users. AKS continually expands into new regions.
* Azure paired regions: For your geographic area, choose two regions that are paired with each other. Paired regions coordinate platform updates and prioritize recovery efforts where needed.
* Service availability: Decide whether your paired regions should be hot/hot, hot/warm, or hot/cold. Do you want to run both regions at the same time, with one region ready to start serving traffic? Or do you want one region to have time to get ready to serve traffic?

AKS workloads may not need to run continuously, for example a development cluster that is used only during business hours. This leads to times where your Azure Kubernetes Service (AKS) cluster might be idle, running no more than the system components. You can reduce the cluster footprint by [scaling all the User node pools to 0](https://docs.microsoft.com/en-us/azure/aks/scale-cluster#scale-user-node-pools-to-0), but your [System pool](https://docs.microsoft.com/en-us/azure/aks/use-system-pools) is still required to run the system components while the cluster is running. To optimize your costs further during these periods, you can completely turn off (stop) your cluster. This action will stop your control plane and agent nodes altogether, allowing you to save on all the compute costs, while maintaining all your objects and cluster state stored for when you start it again. You can then pick up right where you left of after a weekend or to have your cluster running only while you run your batch jobs.

**References:**

1. [Best practices for business continuity and disaster recovery in Azure Kubernetes Service (AKS)](https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-multi-region)
2. [Stop and Start an Azure Kubernetes Service (AKS) cluster](https://docs.microsoft.com/en-us/azure/aks/start-stop-cluster)

# Can I join an on prem node to an AKS cluster?

No, but can use Azure Arc which offers simplified management, faster app development, and consistent Azure services. Standardize visibility, operations, and compliance across a wide range of resources and locations by extending the Azure control plane. Build cloud-native apps anywhere, at scale. Code and ship applications from Azure to any Kubernetes distributions in any location. Get Azure innovation and cloud benefits by deploying Azure data services on any infrastructure, and securely build and train machine learning models anywhere.

1. Centrally manage a wide range of resources including [Windows](https://azure.microsoft.com/en-us/campaigns/windows-server/) and [Linux](https://azure.microsoft.com/en-us/overview/linux-on-azure/) servers, SQL server, [Kubernetes](https://azure.microsoft.com/en-us/services/kubernetes-service/) clusters, and [Azure services](https://azure.microsoft.com/en-us/services/azure-arc/hybrid-data-services/).
2. Establish central visibility in the [Azure portal](https://azure.microsoft.com/en-us/features/azure-portal/) and enable multi-environment search with Azure Resource Graph.
3. Meet [governance](https://azure.microsoft.com/en-us/solutions/governance/) and compliance standards for apps, infrastructure, and data with [Azure Policy](https://azure.microsoft.com/en-us/services/azure-policy/).
4. Delegate access and manage security policies for resources using role-based access control (RBAC) and [Azure Lighthouse](https://azure.microsoft.com/en-us/services/azure-lighthouse/).
5. Organize and inventory assets through a variety of Azure scopes, such as management groups, subscriptions, resource groups, and tags.

# How does istio work?

## Istio provides the following set of capabilities

1. Mesh – gateways (multi-cluster), virtual machines (mesh expansion)
2. Traffic Management – routing, splitting, timeouts, circuit breakers, retries, ingress, egress
3. Policy – access control, rate limit, quota, custom policy adapters
4. Security – authentication (jwt), authorisation, encryption (mTLS), external CA (HashiCorp Vault)
5. Observability – golden metrics, mirror, tracing, custom adapters, prometheus, grafana

## Istio is well suited to and suggested for the following scenarios

1. Require extensibility and rich set of capabilities
2. Mesh expansion to include VM based workloads
3. Multi-cluster service mesh

**Callout:** Microsoft supports but is not responsible for istio bugs. Resources work with them to pull them in. Better control over GRPC and http connections, traffic behavior For microservices.

Dapr helps developers build event-driven, resilient distributed applications. Whether on-premises, in the cloud, or on an edge device, Dapr helps you tackle the challenges that come with building microservices and keeps your code platform agnostic.

**References:**

1. <https://i:stio.io/latest/docs/concepts/what-is-istio/>
2. <https://docs.microsoft.com/en-us/azure/aks/servicemesh-about>
3. <https://dapr.io/>

# What is our process for draining nodes? Should it be a rundeck job? Should it be regular? How does the cluster remain balanced?

If the resource needs of your applications change, you can manually scale an AKS cluster to run a different number of nodes. When you scale down, nodes are carefully cordoned and drained to minimize disruption to running applications. When you scale up, AKS waits until nodes are marked Ready by the Kubernetes cluster before pods are scheduled on them.

To endure that your workloads remain available during maintenance, you can configure a [PodDisruptionBudget](https://kubernetes.io/docs/concepts/workloads/pods/disruptions/).

If availability is important for any applications that run or could run on the node(s) that you are draining, [configure a PodDisruptionBudgets](https://kubernetes.io/docs/tasks/run-application/configure-pdb/) first.

**References:**

1. <https://docs.microsoft.com/en-us/azure/aks/scale-cluster>
2. <https://kubernetes.io/docs/tasks/administer-cluster/safely-drain-node/>

# Do we need to switch NUA01 to the managed identity cluster? How do we migrate?

Azure Kubernetes Service (AKS) cluster (specifically, the Kubernetes cloud provider) requires an identity to create additional resources like load balancers and managed disks in Azure. This identity can be either a managed identity or a service principal. If you use a [service principal](https://docs.microsoft.com/en-us/azure/aks/kubernetes-service-principal), you must either provide one or AKS creates one on your behalf. If you use managed identity, this will be created for you by AKS automatically. Clusters using service principals eventually reach a state in which the service principal must be renewed to keep the cluster working. Managing service principals adds complexity, which is why it's easier to use managed identities instead. The same permission requirements apply for both service principals and managed identities.

**References:**

1. <https://docs.microsoft.com/en-us/azure/aks/use-managed-identity>
2. <https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-identity>

For autoscaling ([see code](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fncr-swt-hospitality%2Fncrpubliccloud-arm-hsp-aks%2Ftree%2Fmaster%2Fdeploy_AKS_mi&data=04%7C01%7Cakshay.nimbalkar%40microsoft.com%7C56e5dd3523b0401b3b4e08d8e3139e01%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637509020696559299%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=KbBlOHx7TxIXecxxcLEk0SRCp80JFfon972zNG7Gktg%3D&reserved=0)) we have a max of 20 machines, what happens if that goes above the ceiling?

## Horizontal pod autoscaler

Kubernetes uses the horizontal pod autoscaler (HPA) to monitor the resource demand and automatically scale the number of replicas. By default, the horizontal pod autoscaler checks the Metrics API every 30 seconds for any required changes in replica count. When changes are required, the number of replicas is increased or decreased accordingly. Horizontal pod autoscaler works with AKS clusters that have deployed the Metrics Server for Kubernetes 1.8+.

## Cluster autoscaler

To respond to changing pod demands, Kubernetes has a cluster autoscaler, that adjusts the number of nodes based on the requested compute resources in the node pool. By default, the cluster autoscaler checks the Metrics API server every 10 seconds for any required changes in node count. If the cluster autoscale determines that a change is required, the number of nodes in your AKS cluster is increased or decreased accordingly. The cluster autoscaler works with Kubernetes RBAC-enabled AKS clusters that run Kubernetes 1.10.x or higher.

Azure Container Instances can schedule both Windows and Linux containers with the same API. Simply specify the OS type when you create your [container groups](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-container-groups).

Some features are currently restricted to Linux containers:

* Multiple containers per container group
* Volume mounting ([Azure Files](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-volume-azure-files), [emptyDir](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-volume-emptydir), [GitRepo](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-volume-gitrepo), [secret](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-volume-secret))
* [Resource usage metrics](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-monitor) with Azure Monitor
* [Virtual network deployment](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-vnet)
* [GPU resources](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-gpu) (preview)

For Windows container deployments, use images based on common [Windows base images](https://docs.microsoft.com/en-us/azure/container-instances/container-instances-faq#what-windows-base-os-images-are-supported).

Rapidly scale application workloads in an AKS cluster, you can use virtual nodes. With virtual nodes, you have quick provisioning of pods, and only pay per second for their execution time. You don't need to wait for Kubernetes cluster autoscaler to deploy VM compute nodes to run the additional pods. Virtual nodes are only supported with Linux pods and nodes.

**References:**

1. <https://docs.microsoft.com/en-us/azure/aks/concepts-scale>
2. <https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/scale-using-aks-with-aci>
3. <https://docs.microsoft.com/en-us/azure/aks/virtual-nodes>

# How does patching occur on the node?

Microsoft provides patches and new images for image nodes weekly, but doesn't automatically patch them by default. Because AKS isn't a platform-as-a-service (PaaS), components like agent nodes have shared responsibility, and users must help maintain the AKS cluster. For example, applying an agent node operating system (OS) security patch requires user input.

AKS supports upgrading node images by using [az aks nodepool upgrade](https://docs.microsoft.com/en-us/cli/azure/ext/aks-preview/aks/nodepool" \l "ext_aks_preview_az_aks_nodepool_upgrade), so you can keep up with the newest OS and runtime updates. To keep your agent node OS and runtime components patched, consider checking and applying node image upgrades bi-weekly, or automating the node image upgrade process. For more information about automating node image upgrades, see [Node upgrade GitHub Actions](https://docs.microsoft.com/en-us/azure/aks/node-upgrade-github-actions).

An updated node image contains up-to-date OS security patches, kernel updates, Kubernetes security updates, newer versions of binaries like kubelet, and component version updates listed in the [release notes](https://github.com/Azure/AKS/releases). Node image updates have all relevant and validated security updates and feature updates. Using the node image upgrade method ensures you get only tested kernels and components that are compatible with those kernels.

You can use node image upgrades to streamline upgrades for both Windows and Linux node pools, but the processes differ slightly. Linux might receive daily security updates, but Windows Server nodes update by performing an AKS upgrade that deploys new nodes with the latest base Window Server image and patches.

Use [kured (KUbernetes REboot Daemon)](https://github.com/weaveworks/kured) to watch for Linux nodes that require a reboot, then automatically handle the rescheduling of running pods and node reboot process.

**References:**

1. <https://docs.microsoft.com/en-us/azure/architecture/operator-guides/aks/aks-upgrade-practices>
2. <https://docs.microsoft.com/en-us/azure/aks/node-updates-kured>

# Should we be running an audit tool as a part of our pipeline? (k8guard, [copper](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcopper.sh%2F&data=04%7C01%7Cakshay.nimbalkar%40microsoft.com%7C56e5dd3523b0401b3b4e08d8e3139e01%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637509020696569300%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=IvPPAJ5svf43tfz89eT4yJWRBx9HHE88mnEpO77gDXI%3D&reserved=0), kube-bench)?

**References:**

1. <https://github.com/fluxcd/flux>
2. <https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/secure-devops-for-kubernetes>
3. <https://azure.microsoft.com/en-us/solutions/devsecops/>

# Where do I find audit logging for the AKS cluster?

With Azure Kubernetes Service (AKS), the control plane components such as the kube-apiserver and kube-controller-manager are provided as a managed service. You create and manage the nodes that run the kubelet and container runtime, and deploy your applications through the managed Kubernetes API server. To help troubleshoot your application and services, you may need to view the logs generated by these control plane components. This article below shows you how to use Azure Monitor logs to enable and query the logs from the Kubernetes control plane components.

**References:**

1. <https://docs.microsoft.com/en-us/azure/aks/view-control-plane-logs>

# Should we use kubectl diff on deployments to see what changed? As a log?

**References:**

1. <https://kubernetes.io/blog/2019/01/14/apiserver-dry-run-and-kubectl-diff/#kubectl-diff>

# Where does kubectl run from? Azure console? My desktop with VPN? (it's a context)

kubectl uses a configuration file. The default kubectl configuration file is located at ~/. kube/config and is referred to as the kubeconfig file. It can run from your local machine, Azure CLI, Powershell, Terraform, etc.

**References:**

1. <https://kubernetes.io/docs/tasks/tools/>

# What is the RBAC for kubectl

**References:**

1. <https://docs.microsoft.com/en-us/azure/aks/manage-azure-rbac#create-a-new-cluster-using-azure-rbac-and-managed-azure-ad-integration>

# How do we use tags?

 AKS allows you to create and modify custom tags created by end users, and you can add those tags when [creating a node pool](https://docs.microsoft.com/en-us/azure/aks/use-multiple-node-pools#specify-a-taint-label-or-tag-for-a-node-pool). You might want to create or modify custom tags, for example, to assign a business unit or cost center. This can also be achieved by creating Azure Policies with a scope on the managed resource group.

However, modifying any **Azure-created tags** on resources under the node resource group in the AKS cluster is an unsupported action, which breaks the service-level objective (SLO). For more information, see [Does AKS offer a service-level agreement?](https://docs.microsoft.com/en-us/azure/aks/faq#does-aks-offer-a-service-level-agreement)

# Container hardening

<https://docs.microsoft.com/en-us/azure/aks/operator-best-practices-cluster-security>

# How to allow TLS pass thru from AGIC to pod level?

**References:**

1. <https://docs.microsoft.com/en-us/azure/application-gateway/ssl-overview#end-to-end-tls-with-the-v2-sku>
2. <https://github.com/fbeltrao/aks-letsencrypt/blob/master/setup-wildcard-certificates-with-azure-dns.md>

# Singleton Pods

If you configure a replicaSet to be a singleton with one replicas, the controller makes sure at least one instance is always running but occasionally it can be more instances.

Option 1: SatefulSets - maintains a sticky identity for each of their Pod; it will prevent the interchangeability of the pod and ensures its existence.

Option 2: In-application Locking - Whenever a service instance or a component inside the instance is activated, it can try to acquire a lock, and if it succeeds, the service becomes active. Any subsequent service instance that fails to acquire the lock waits and continuously tries to get the lock in case the currently active service release it. When a resource try to edit a configMap or secret, Kubernetes API provides leader election and singleton capabilities to ensure only one pod can update a configMap at a time.

**References:**

1. <https://medium.com/@fahed.dorgaa/kubernetes-singleton-pattern-84a88a06727a>