FSCP Azure Cookbook - Kubernetes Service (AKS)

Software Development

Exported on 06/06/2025

Table of Contents

Page Owner

Siebrand Feenstra

AKS Community and Product Announcements

AKS Container Community Channel: Container Community[1]
AKS Product Announcements: Announcements[2]

[1] https://teams.microsoft.com/l/channel/19%3aa805147e0dfb414ba6a2466b85c72645%40thread.skype/General?groupId=4f38b48b-b787-403c-b1c0-2b16f7daf8b9&tenantId=3a15904d-3fd9-4256-a753-beb05cdf0c6d

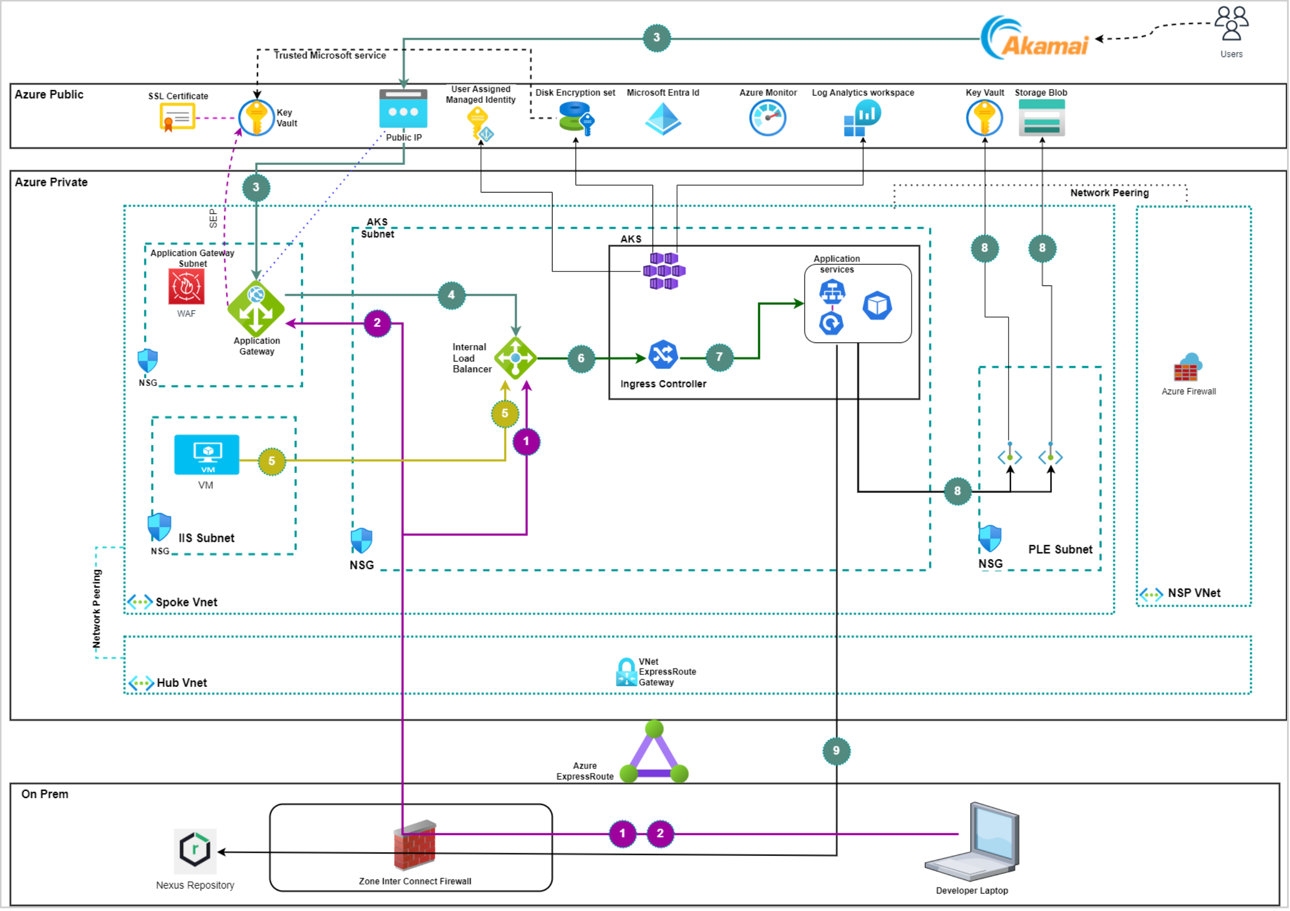
[2] https://teams.microsoft.com/l/channel/19%3aefeea77ee670404d9b19d834ad14a41b%40thread.skype/Announcements?groupId=4f38b48b-b787-403c-b1c0-2b16f7daf8b9&tenantId=3a15904d-3fd9-4256-a753-beb05cdf0c6d

1 Aim of the Service

In FSCP 3 .0, Azure Kubernetes Service is a Container orchestration platform with in a private cluster and there by ensuring the worker nodes and control plane interaction to be private and secure that connects to the Api Server securely and connects to other Azure PaaS components via Azure Private Link.

Private Endpoint uses a private IP address from customer VNet, resulting in bringing the service into customer managed network.

2 Design and network aspect of the Service



3 Connectivity and Integration

- **On-Premise to Azure**
  - From On-Premise to Azure Private routing is controlled by Express Route or directly connected via Domain Services.
- **Azure to Azure**
  - Azure Private to Azure Public services are connected via **PLE subnet and private endpoints.**
- **Azure to On-premise**
  - NSG rules are configured for “AKS Subnet” that allow traffic to only desired On-Prem
  - On-prem connectivity will be enabled via the normal request processes to open firewall ports (Inter-zone connect).
  - Connectivity for services hosted on CMS will be enabled through backend services (IAG/ESB APIs) over HTTPS which will be available directly.

Conn #	Source	Destination	Path	Port	Protocol	Authentication	Description
1	ABN AMRO user: • Zscaler • Worksquare	Internal Load Balancer	ZIF	443	HTTPS	Depended on the application.	ABN AMRO users from On Prem can access AKS application services privately via the Internal Load balance.
2	ABN AMRO user: • Zscaler • Worksquare	Application gateway (Private IP)	ZIF	443	HTTPS	Depended on the application.	ABN AMRO users from On Prem can access AKS application services privately via Application Gateway Internal Private IP, which is linked to the Internal Load balance.

Conn #	Source	Destination	Path	Port	Protocol	Authentication	Description
3	Users from the Internet	Application gateway (Public IP)	Akamai / WAF	443	HTTPS	Dependent on the application.	Public access to the AKS application services via Akamai and Application Gateway Web Application Firewall, which is linked to the Internal Load balance.
4	Application Gateway	Internal Load Balancer	Subnet to Subnet	443	HTTPS	N/A	
5	VM/ Application	Internal Load Balancer	Subnet to Subnet	443	HTTPS	Dependent on the application.	
6	Internal Load Balancer	Ingress Controller	Network Policy	443	HTTPS	N/A	Internal Load Balancer of type Basic gets created along with AKS Cluster Creation. Internal Load Balancer Configuration is managed by Ingress Controller. Hence Ingress Controller deployment assigns Private IP to Internal Load balancer And ILB act as a frontend for Ingress Controller. ILB allows external client within VNet to connect to Ingress Ingress controller , by using Stable IP address. • Straight forward ingress option without Web Application Firewall.

Conn #	Source	Destination	Path	Port	Protocol	Authentication	Description
7	Ingress Controller	AKS Application Services	Routing Rules defined In Ingress Controller resource.	443	HTTPS	N/A	Ingress Controller Is responsible for managing external access to services within Cluster. redirects traffic to respective Service in the cluster. For e.g Accessing AKS Application from Virtual Machine within same VNet.
8	AKS Application	Azure PaaS Services	PLE Subnet	Dependent on the PaaS service	Dependent on the PaaS service	Work Load Identity	Application deployed in AKS containers can access PaaS services privately via Private endpoints.
9	AKS Application	Nexus Repository	ZIF	443	HTTPS	Solo Credentials	

3.1 Calico Network Policy

AKS clusters in FSCP 3.0 have by default a 'deny-all' policy applied provided by the Calico project which disabled all network traffic within the cluster. Generic information and guidance can be found here; FSCP AKS Calico Network Policies[3]

3.2 AKS Roles

Azure Kubernetes Service (AKS) offers two types of role-based access control (RBAC) roles: RBAC-enabled and non-RBAC.

- RBAC-enabled roles allow you to grant granular permissions to users and service principals to perform specific tasks within AKS but not in Azure. This includes access to resources such as pods, services, and deployments. With RBAC, you can create custom roles with specific permissions, and assign those roles to users and service principals.
- **non**-RBAC AKS roles let you manage AKS-related resources in Azure, but not inside Kubernetes.

resources:

Azure RBAC Roles - Overview[4]
Azure Kubernetes Service Cluster Admin Roles[5]

3.3 Connect resources using AAD

Applications that we deploy in AKS clusters require AAD application credentials or managed identity to access AAD-protected resources. Azure AD Workload Identity for Kubernetes integrates with the capabilities native to Kubernetes to federate with Azure AD.

In AKS in order to access other PaaS services we can leverage the AAD Pod Identity project as a way to manage this however since 24 Oct 2022 this feature is deprecated[6] and is now replaced by Workload Identity[7].

Other methods to securely retrieve secrets from an Azure Key Vault or interact with a storage account can be done using AKS CSI driver feature.

[3] https://confluence.int.abnamro.com/x/0PkXHQ
[4] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81997/Azure-RBAC-Roles
[5] https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#azure-kubernetes-service-cluster-admin-role
[6] https://github.com/Azure/aad-pod-identity
[7] https://azure.github.io/azure-workload-identity/docs/

Method	Description	link
Azure AD Workload identity	Azure AD Workload identity is used to authenticate *applications* running on Kubernetes clusters with Azure AD. It allows you to use an Azure AD identity to authenticate with services that support Azure AD authentication.	FSCP AKS Workload Identity[8]
Azure Key Vault Provider for Secrets Store CSI driver	The Azure Key Vault Provider for Secrets Store CSI driver is used to provide an access identity to the Azure Key Vault Provider for Secrets Store CSI driver. It provides a way to store and retrieve secrets in Kubernetes applications by using a volume plugin.	Azure Key Vault Provider for Secrets Store CSI (see page 62)
Azure Files Container Storage Interface (CSI) driver	The Azure Files Container Storage Interface (CSI) driver is a CSI specification-compliant driver used by Azure Kubernetes Service (AKS) to manage the lifecycle of Azure file shares. The CSI is a standard for exposing arbitrary block and file storage systems to containerized workloads on Kubernetes.	Connect AKS to Azure Files (see page 58)
Azure Disk Container Storage Interface (CSI) driver	The Azure Disks Container Storage Interface (CSI) driver is a CSI specification[9]-compliant driver used by Azure Kubernetes Service (AKS) to manage the lifecycle of Azure Disk.	Use Container Storage Interface (CSI) driver for Azure Disk on Azure Kubernetes Service (AKS) - Azure Kubernetes Service | Microsoft Learn[10]
Azure Blob storage Container Storage Interface (CSI) driver	The Azure Blob storage Container Storage Interface (CSI) driver is a CSI specification[11]-compliant driver used by Azure Kubernetes Service (AKS) to manage the lifecycle of Azure Blob storage. The CSI is a standard for exposing arbitrary block and file storage systems to containerized workloads on Kubernetes.	Use Container Storage Interface (CSI) driver for Azure Blob storage on Azure Kubernetes Service (AKS) - Azure Kubernetes Service | Microsoft Learn[12]

[8] https://confluence.int.abnamro.com/x/sksQHw
[9] https://github.com/container-storage-interface/spec/blob/master/spec.md
[10] https://learn.microsoft.com/en-us/azure/aks/azure-disk-csi
[11] https://github.com/container-storage-interface/spec/blob/master/spec.md
[12] https://learn.microsoft.com/en-us/azure/aks/azure-blob-csi?tabs=NFS

In terms of use cases, if you want to store and retrieve secrets in Kubernetes applications using key vault or files, you can use AKS CSI driver. If you want to authenticate applications running on Kubernetes clusters with Azure AD, you can use Workload Identity.

4 FinOps Considerations

4.1 Generic guidelines

It's good practice to periodically review your architecture and reduce costs by adopting newer cloud offerings and adhere to some design principles when it comes to cost optimization. Especially during transitioning from FSCP 2.0 and FSCP 3.0 is a good time to reconsider your product choice. Examples of design principles specific to cost management are;

- Setting limits to stay within cost constraints and mitigate the risk of excessive cloud spend.
- Aim for scalable costs as the cloud eliminates the need to over provision
- Use (only) what you need and scale out to meet demand.
- Select the right resources and right size your infrastructure.
- Take advantage of Platform as a Service (PaaS) and
- Choose Virtual Machine (VM) size based actual usage.

4.2 Finance Policies

FSCP 3.0 introduces Finance Policies which are a result of the Standard of Cloud Cost Control (SCCC)[13] and registered per service description under '(Pre-)Production COST Controls'. The abbreviations here mean the following, taken from the SCCC reference.

abbreviations

[13] https://eur03.safelinks.protection.outlook.com/?url=https%3A%2F%2Fabnamro.sharepoint.com%2Fsites%2Fintranet-informatie_it-design-engineering%2FLists%2FDA%2520Designs%2520List%2FDispForm.aspx%3FID%3D1464%26e%3DFyktxj&data=05%7C01%7Csiebrand.feenstra%40nl.abnamro.com%7C4ffd45336a3d404f051408db76f19898%7C3a15904d3fd94256a753beb05cdf0c6d%7C0%7C0%7C638234552252392142%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=Y4pmN70m%2BvQ%2BGYNpVFiReu2HryAYmo3G9z01EVufRfI%3D&reserved=0

Abbreviations	Meaning
FC	**F**oundation **C**ost. Generic controls.
OP	**O**ver **P**rovisioned. When resources are deployed in a configuration that exceeds the need of an application, in either size or service tier, they create waste.
ST	**S**pend **T**arget. Exceeding the spend target can lead to excessive spending and might be a result of incorrect calculation or even an indication of a design violation. The reason for exceeding the spend target should be monitored and rectified.
SP	**SP**ending anomalies. Unexpected spikes in cloud spend can be an indicator for improper usage.
CO	**C**ost **O**ptimization. Reservations and Savings Plans contribute to lower cloud spend when efficiently allocated and used.

4.3 Recommendations

Service specific	Generic	Tooling
service name • Upgrade to Newer SKU's - Azure Pricing | Microsoft Azure[14] / Azure VM Comparison (azureprice.net)[15] • Finance Policies: (Pre-)Production COST Controls[16] • Save resources (and money) on clusters - Overview (azure.com)[17]	*Migrate to newer SKU's* • Latest Azure SKU's show energy efficient technology and offer better performance at lower price point. • Leverage technology evolution benefits passed on through modern SKU's. • Downscaling, Relocation and Reallocation wherever possible • Prevent waste by right size SKU family and service tier. • Leverage reservations and saving plans where applicable.	• Tower - Capacity Management - Power BI[18] • Use the filter (top right) to browse your application • Use the slider to adjust (default cost optimizations are shown > 250 euro's) • Guidance: Capacity Management[19] • Tower - Costs - Power BI[20] • See Actual Spend vs Spend Target • Adjust your spend target accordingly • Guidance: Costs[21]

4.4 Optimization example

provide example if any

[14] https://azure.microsoft.com/en-us/pricing/details/kubernetes-service/

[15] https://azureprice.net/

[16] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/67884/AAB-Azure-Kubernetes-Service-v1?anchor=pre-production-cost-controls

[17] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/GRD0001007.wiki/65365/Save-resources-(and-money)-on-clusters?anchor=reduce-the-number-of-pods%27-replicas

[18] https://app.powerbi.com/groups/me/apps/bcaf12de-c9c9-421a-9a81-695f18eee911/reports/7d915dbb-2165-4da9-930f-5bf6caaf9706/ReportSection3cdf9ac3274daa2422b6?ctid=3a15904d-3fd9-4256-a753-beb05cdf0c6d&experience=power-bi&clientSideAuth=0

[19] https://confluence.int.abnamro.com/x/nTLPFQ

[20] https://app.powerbi.com/groups/me/apps/bcaf12de-c9c9-421a-9a81-695f18eee911/reports/7d915dbb-2165-4da9-930f-5bf6caaf9706/ReportSection2a4fc9372d6c1731ce09?ctid=3a15904d-3fd9-4256-a753-beb05cdf0c6d&experience=power-bi&clientSideAuth=0

[21] https://confluence.int.abnamro.com/x/ympmDQ

5 Right Size and scale your Cluster

We do not provide specific recommendations as they are solution specific. You as a customer are responsible to right size your environment and knows best what is needed based on your solution needs. However some general suggestions and best practices can be given;

- As per policy FSCP deploys some mandatory extensions like Azure Policy Integration (gatekeeper) and flux. Microsoft set the limits for the pods deployed by those extensions quite high which can lead to overcommitting with a change that evicting customer's pods starts to happen. Take into consideration that the system node pool must run **all** the mandated components[22]. As a best practice use one system node pool for system workload and mandated components, plus one or more user node pools for user workloads.
- We suggest[23] a system node pool of `Standard_D4s_v5` for a *newly deployed, empty cluster*, so anything higher is even better.
- Customers **can** use arm64 nodes[24] in one or more node pools.
- Calculate-Subnet-Requirements[25]
- Follow the best practices[26], specifically the ones below on sizing and scaling.
- In D and T, consider using the VerticalPodAutoscaler[27] in your application to leverage trials and errors to find and set a reasonable amount of resources for your workload.
- *Avoid like the Plague enabling **both** the* HorizontalPodAutoscaler[28] *and the* VerticalPodAutoscaler[29] *in your application.* Update: K8S 1.27 introduced the ability (in alpha) to resize Pods' resources on the fly[30] , and this *might™* solve this particular issue.
- Force your app's Pods to run on tailored or dedicated Nodes when possible. Leverage Affinity and Anti-Affinity[31] and/or Topology Spread Constraints[32] in your application's definition.

[22] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81794/Mandatory-Components

[23] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81995/IaC-reference-examples?anchor=suggestions-about-values

[24] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/90727/ARM64-Nodes-and-multi-architecture-images

[25] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81994/Calculate-Subnet-Requirements

[26] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/87682/Best-practices

[27] https://learn.microsoft.com/en-us/azure/aks/vertical-pod-autoscaler

[28] https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/

[29] https://learn.microsoft.com/en-us/azure/aks/vertical-pod-autoscaler

[30] https://kubernetes.io/docs/tasks/configure-pod-container/resize-container-resources/

[31] https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#affinity-and-anti-affinity

[32] https://kubernetes.io/docs/concepts/scheduling-eviction/topology-spread-constraints/

5.1 Guidance on CNI Networking options

AKS clusters in FSCP use Azure CNI to reserve IPs and route traffic between nodes and pods.

A cluster requires one Subnet IP for each node VM, plus one Subnet IP per pod each node can run.
The total amounts to 31 IPs per node considering the default value of 30 pods per node[33].

Considered options

Other options have been considered to solve the issue above.
For a detailed list of pros and cons, check the pros and cons of the considered options[34] section below.

- Azure CNI with Overlay network[35].
  The feature illustrated in this very document.
- Azure CNI with a single Subnet[36].
  The default setting for AKS in 3.0 until December 2023.
- Azure CNI with an extra Subnet delegated and dedicated to the pods[37].
  A.K.A. *dynamic IP allocation*.

5.1.1 Configure CNI Overlay (*Preferred*)

The private IPv4 address space is scarce. With CNI Overlay there's no more need to reserve one IPv4 address per pod.

- It reduces the IP consumption from the internal private address space by using a different, private Subnet for the pods addressing.
- One must create a cluster with the Overlay settings in place from the start.
- Any AKS cluster can have **only one** Overlay network assigned which needs to be a `/16` IP address block.
- Within FSCP a `172.16.0.0/12` address block is reserved as the **base** IP address space for all overlay networks where you can choose any `/16` CIDR contained in that.

See this how to section for setting up a cluster with CNI overlay enabled: How To - Configure CNI Overlay

[33] https://learn.microsoft.com/en-us/azure/aks/azure-cni-overview#maximum-pods-per-node
[34] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/95392/CNI-Overlay?anchor=pros-and-cons-of-the-considered-networking-options#pros-and-cons-of-the-considered-options
[35] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/95392/CNI-Overlay?anchor=pros-and-cons-of-the-considered-networking-options#azure-cni-with-overlay-network
[36] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/95392/CNI-Overlay?anchor=pros-and-cons-of-the-considered-networking-options#azure-cni-with-a-single-subnet
[37] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/95392/CNI-Overlay?anchor=pros-and-cons-of-the-considered-networking-options#azure-cni-with-an-extra-subnet-delegated-and-dedicated-to-the-pods

5.1.2 Configure Azure CNI networking for dynamic allocation of IPs

The dynamic IP allocation capability in Azure CNI allocates pod IPs from a subnet separate from the subnet hosting the AKS cluster. Planning your IP addressing is much simpler with this feature. Since the nodes and pods scale independently, their address spaces can also be planned separately. Since pod subnets can be configured to the granularity of a node pool, you can always add a new subnet when you add a node pool. The system pods in a cluster/node pool also receive IPs from the pod subnet. Only one pod subnet can be assigned to a cluster or node pool. However, multiple clusters or node pools can share a single pod subnet.

- Requires an extra subnet per cluster, which complicates operations.
- Requires custom security group rules to allow traffic between nodes and pods.
- Consumes one routable IP per node + one routable IP per pod.
- Traffic isn't translated.

See for more info: Configure Azure CNI networking for dynamic allocation of IPs and enhanced subnet support - Azure Kubernetes Service | Microsoft Learn[38]

> *The 'AKS Quickstart Pipeline' (see Reference Pipeline Link or Onboard to FSCP 3.0 AKS[39]) section accounts for this feature by providing a 'podDynamicIpAllocation' parameter.*

[38] https://learn.microsoft.com/en-us/azure/aks/configure-azure-cni-dynamic-ip-allocation
[39] https://confluence.int.abnamro.com/display/GRIDAD/Onboard+to+FSCP+3.0+AKS

6 Prerequisites

- Enterprise Landing Zone and a customer resource group.
- Virtual Network with AKS subnet design pattern Pattern.
- User Assigned Managed Identity to deal with cluster connectivity with Azure PaaS.
- Log Analytics Workspace
- DiskEncryptionSet as a part of encrypt VM data (worker node) for recovery and security.
- Azure Key Vault can be used to store secrets securely (see Secrets Management (see page 62)).

> *AKS was integrated with Hashicorp Vault for this purpose. Vault is no longer desirable and has been effectively deprecated. In its place, we will use Azure Key Vault and the supplied integration with Kubernetes.*

7 General

7.1 Introduction to AKS on FSCP 3.0

AKS will be enabled in a different way compared to FSCP 2.0; there will no longer be a Secure Product i.e., Managed Container Platform (MCP). Instead, all the necessary governance controls will be implemented as part of the native Azure platform, using e.g., Azure Policy, Azure Policy for Kubernetes, and AKS GitOps Flux extension. See Intro to FSCP 3.0 AKS[40] for more information.

On this page you can find Frequently Asked Questions and recommendations with useful links to documentation on best practices for migrations. Transition from FSCP 2.0 MCP AKS to FSCP 3.0 AKS [41]

7.2 Best Practices

This is a collection of suggestions and best practices to squeeze all you can from your resources. All suggestions are given in no particular order.
Stratus - Best practices - Overview (azure.com) - Stratus[42] & Save resources (and money) on clusters - Overview (azure.com)[43]

7.3 Concepts

These pages capture some sort of abstract idea or artefact, like a design. They only serve to explain the concept, they do not instruct in how something should be done.
Stratus - Concepts - Overview (azure.com)[44]

7.4 Responsibility and differences from FSCP 2.0

Differences from FSCP (MCP) 2.0 - Stratus[45]

You will be responsible for almost everything

You'll need to create your own:

- Infrastructure as Code (Bicep, Terraform, ARM Template, etc.)
- Azure DevOps artefacts (pipelines, service connections, etc)

[40] https://confluence.int.abnamro.com/x/dSBSG
[41] https://confluence.int.abnamro.com/x/eCBSG
[42] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/87682/7-Best-practices
[43] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/GRD0001007.wiki/65365/Save-resources-(and-money)-on-clusters?anchor=reduce-the-number-of-pods%27-replicas
[44] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81771/Concepts
[45] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/GRD0001007.wiki/69682/FSCP-3.0-AKS?anchor=you-will-not-be-able-to-expect-the-same-support-from-stratus

- Setup Workload identity or CSI driver where applicable
- Deployment of your ingress solution

Stratus, as part of FSCP, will continue to be responsible for making AKS available at the platform-level with very narrow, specific responsibilities.

7.4.1 Responsibility matrix

See this page for the RACI (responsibility matrix) model around AKS; FSCP 3.0 responsibility overview - Overview (azure.com)[46]

7.5 Monitoring and logging

In FSCP 3.0, it is mandatory for DevOps teams to forward their AKS Container Insights logs to Sentinel Log Analytics Workspace (sen-p-la), to allow for ASM and ISM-related alerting and monitoring. FSCP has implemented policies to send Diagnostics logging to Sentinel. Container insights logs are sent to your custom Log Analytics Workspace which is deployed as part of your cluster deployment and than exported to Sentinel.

More information can be found here; Central Logging Design and Responsibilities - Overview (azure.com)[47]

Currently 03 Jan 2024 no logs are being sent to infra-p-oms.

If you have a requirement to gather diagnostics logs or metrics from AKS to your workspace you can do so. Please take into account that data injestion and especially retention can increase costs. See FSCP Azure Monitor - FinOps Recommendations[48] for more information.

7.6 Mandatory Components

The FSCP Secure Context solution for AKS will enforce compliance by denying in-compliant configurations and by automatically installing mandatory components.

see; Mandatory Components - Stratus[49]

7.6.1 Mandatory Tags

Current the two following mandatory tags should be applied as part of your AKS cluster;

[46] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/GRD0001007.wiki/74354/FSCP-3.0-responsibility-overview?anchor=tl%3Bdr

[47] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/GRD0001007.wiki/84122/Central-Logging-Design-and-Responsibilities?anchor=azure-policies

[48] https://confluence.int.abnamro.com/x/2-ZkGQ#FSCPAzureCookbookAzureMonitor(Loganalytics)-FinOpsConsiderations

[49] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81794/Mandatory-Components

- NSF-Function[50] - Add an NSF Function tag based on the relevant NSF Function. To determine what value to provide you should request for NSF Intake[51]. Pods can be assigned the following NSF-Function tags:
    - Compute components responsible for a presentation interface will have an NSF-Function of **Presentation**.
    - Compute components responsible for data processing, data transformation and streaming will have an NSF-Function of **Application**.
    - 'NSF-Function' tag should be applied on managedClusters and for the VMSS pools under 'agentPoolProfiles':

[50] https://abnamro.sharepoint.com/sites/NSS/SitePages/NSF-Function-Tag---Kubernetes-Service-(AKS).aspx?csf=1&web=1&e=TH9DQM&ovuser=3a15904d-3fd9-4256-a753-beb05cdf0c6d%2csiebrand.feenstra%40nl.abnamro.com&OR=Teams-HL&CT=1715421003705&clickparams=eyJBcHBOYW1lIjoiVGVhbXMtRGVza3RvcCIsIkFwcFZlcnNpb24iOiI0OS8yNDAzMzEwMTgxNyIsIkhhc0ZlZGVyYXRlZFVzZXIiOmZhbHNlfQ%3d%3d&cid=88bc24f7-795e-44f5-a5fc-263e2d9830a0

[51] https://abnamro.sharepoint.com/:u:/r/sites/NSS/SitePages/NSF-Intake---How-To-Page.aspx?csf=1&web=1&e=INuJqm

b01bc60f2169%22%2C%22%2Fsubscriptions%2F62a480f8-d43d-4514-a6ab-e40c52fc7d59%22%2C%22%2Fsubscriptions%2Fe04bc133-bdbc-4a0e-bb67-0f61f4e9a59a%22%2C%22%2Fsubscriptions%2Fe2cf867f-817d-4035-ae21-2558adf93879%22%2C%22%2Fsubscriptions%2F72ca9d79-e6cd-479d-993e-84d0963c78cb%22%2C%22%2Fsubscriptions%2Fb0786e83-469f-4de9-8c0b-136def86a7c4%22%2C%22%2Fsubscriptions%2Fb0191934-3139-4d8b-a185-26a983866005%22%2C%22%2Fsubscriptions%2Ff51b7919-671e-44a9-b2ac-67ea2deb4f9d%22%2C%22%2Fsubscriptions%2F82f9165e-9683-467c-b86c-864e0f0a24c4%22%2C%22%2Fsubscriptions%2F9b1d3f0b-6a34-4084-a4df-157f59acd4ec%22%2C%22%2Fsubscriptions%2Fdf24fea3-c2ef-4356-b025-fae715565030%22%2C%22%2Fsubscriptions%2Fc4b35ecd-a7b5-4f93-8cd3-6825d5248414%22%2C%22%2Fsubscriptions%2Fb688a276-829f-4350-a61f-a926d226d1dd%22%2C%22%2Fsubscriptions%2F469cd825-8d77-4f0b-95f5-50e0ae6e8ff2%22%2C%22%2Fsubscriptions%2F2d73ecf3-b0ff-4eee-9031-fe94ed4efbf2%22%2C%22%2Fsubscriptions%2F61cd4fa4-35c3-44e9-9261-7a76628f9481%22%2C%22%2Fsubscriptions%2F31de0114-83d1-4fc9-a616-9a6c6cc7661c%22%2C%22%2Fsubscriptions%2F6867335c-62f2-4928-a470-a2b7cf4cdb7d%22%2C%22%2Fsubscriptions%2Fd3bf3483-803c-4225-b48c-ac72cb6ab6bd%22%2C%22%2Fsubscriptions%2F5a7fd423-387a-44f9-b727-21cec32332bf%22%2C%22%2Fsubscriptions%2F621690f7-5d68-44c7-96d4-50e26753853a%22%2C%22%2Fsubscriptions%2Ffec6bb2a-bc94-4b06-ba58-c7a2b54d01bc%22%2C%22%2Fsubscriptions%2Feb2fb090-1eff-4181-9dfa-1b49eda70538%22%2C%22%2Fsubscriptions%2F8f59c2d1-7abf-476c-8272-558037c43230%22%2C%22%2Fsubscriptions%2Fa6b24989-88bc-4321-8326-6f577b820b87%22%2C%22%2Fsubscriptions%2Fc1c0e100-aa8e-48bc-9367-ed0d5c9b64e0%22%2C%22%2Fsubscriptions%2F1a6426e6-8866-49b8-b6c3-aca958621472%22%2C%22%2Fsubscriptions%2F13f01c82-cec6-48e3-bb5b-319e5324337c%22%2C%22%2Fsubscriptions%2F6b3b15cc-a33a-4ce3-a600-257c6186f332%22%2C%22%2Fsubscriptions%2Fb1dae516-77d3-41b9-9955-d0e72f672e03%22%2C%22%2Fsubscriptions%2F5f413eb0-9002-49de-b325-c5ccfef13f76%22%2C%22%2Fsubscriptions%2F3d13e72d-0b3c-44fc-96f5-1d0ae037424b%22%2C%22%2Fsubscriptions%2F051cfa0a-6e5d-4487-9244-794894ed8b32%22%2C%22%2Fsubscriptions%2F554bee05-a55f-493f-921e-2248245dc1fc%22%2C%22%2Fsubscriptions%2F144c9efe-c552-40c1-af7e-82850149fa6d%22%2C%22%2Fsubscriptions%2F379bef1d-0acb-4376-b2fb-c81519a07c54%22%2C%22%2Fsubscriptions%2Fe2896fc1-88b2-4309-ba22-fb702817a8fa%22%2C%22%2Fsubscriptions%2F5259af2b-cd18-47d7-b55d-e812aadf1e89%22%2C%22%2Fsubscriptions%2F576e62de-ff04-4f95-8645-91bdb81e4b84%22%2C%22%2Fsubscriptions%2Feb138aa7-f2b5-4c18-88d8-1f6dffea22e2%22%2C%22%2Fsubscriptions%2Ff63f8f2d-ba2e-4094-964c-757abf30fb33%22%2C%22%2Fsubscriptions%2Fbade0f31-4e36-418c-952e-413b53e891ec%22%2C%22%2Fsubscriptions%2Fc6f7960c-9210-43a8-9034-00e2693d114b%22%2C%22%2Fsubscriptions%2F8e1874dd-2cf0-48c8-9912-fb3bfdbc6a1a%22%2C%22%2Fsubscriptions%2F89e78db6-3618-456e-9bf4-f8f7d15ba689%22%2C%22%2Fsubscriptions%2Fb4f27b8b-40b0-46a8-9df0-a0a7b4c699fc%22%2C%22%2Fsubscriptions%2F6607d23f-4aeb-4d64-8c20-3a3533d26058%22%2C%22%2Fsubscriptions%2Fbff8b5ba-302e-4720-a0ec-d7174f62a6ad%22%2C%22%2Fsubscriptions%2F28eddcdd-81c0-4086-95d0-6a230f8585bd%22%2C%22%2Fsubscriptions%2Fac9b4eff-9b0a-42bc-8967-726db46f1092%22%2C%22%2Fsubscriptions%2F747b6a74-f835-4049-a0ed-fda61ca31c96%22%2C%22%2Fsubscriptions%2F4da5e866-8c7c-4ff5-abc6-148d1fe13460%22%2C%22%2Fsubscriptions%2F315b1369-14f0-4e49-9e76-30121c654ebc%22%2C%22%2Fsubscriptions%2F14680693-c5c5-47a4-af52-9dd81d1b68c2%22%2C%22%2Fsubscriptions%2F2293ee8d-7196-401e-8535-d582f72d99ce%22%2C%22%2Fsubscriptions%2F40d73396-2497-4718-af9c-cbfdaf7bdb21%22%2C%22%2Fsubscriptions%2F657ff1ea-291f-4f60-b7c1-bab383c00879%22%2C%22%2Fsubscriptions%2Ff3f63752-980a-40b4-a70a-92ead0ee9a6c%22%2C%22%2Fsubscriptions%2Fc7b0c2f4-63ad-46c0-834a-678dc9765c7f%22%2C%22%2Fsubscriptions%2F5d5bb64b-20f0-42e6-9036-1787365e41b2%22%2C%22%2Fsubscriptions%2F61c04520-d03b-4e4c-9ac7-4c24705873e3%22%2C%22%2Fsubscriptions%2Fffc0f46b-60cc-4727-ba05-a32d04f29f26%22%2C%22%2Fsubscriptions%2F889de3b2-01c8-4b31-a1b6-55224dd69512%22%2C%22%2Fsubscriptions%2F67cfeb3e-e395-4745-bbeb-5b1468e8ba6a%22%2C%22%2Fsubscriptions%2F1b63d2a8-5993-45af-bc67-18cdcc453a66%22%2C%22%2Fsubscriptions%2Fbc959a5b-4d3f-49ef-97fc-47bbfc6898d6%22%2C%22%2Fsubscriptions%2F97f0f04a-2b7e-4a73-b219-ed3a208353d2%22%5D

- ResourceGroupServicePrincipalObjectId[52] – The Object ID of the Resource Group Service Principal need to be tagged to comply to Azure Policy 'AAB Platform - Managed Resource Group Tag DENY v1'

Make sure to include the tags in your (bicep) deployment to stay compliant

Mandatory tags

```
resource managedCluster 'Microsoft.ContainerService/managedClusters@2023-03-01' = {
  name: clusterName
  location: location
  tags: {
    ResourceGroupServicePrincipalObjectId: servicePrincipleObjectId
    'NSF-Function': 'Application'
  }
...
  agentPoolProfiles: [
  {
    name: system
    tags: {
      'NSF-Function': 'Application'
    }
   }
  ]
}
```

7.7 Container Images and Nexus

Nexus should be used to consume Artifacts from to be used within AKS and Azure Container Registries should not be used. Nexus Repository Manager[53] (NXRM) manages software components required for development, deployment, and provisioning. Upstream Images can not be consumed in AKS directly but can be consumed from the Nexus hosted repository directly or by leveraging proxy repositories and wrapping them using a dockerfile, tag an push them to Nexus hosted repository. Nexus is the single point of truth and ACR's are no longer allowed to consume images from. ACR was used primarily to improve performance of image pulling and for its feature to store helm charts. Nexus can act as an private docker registry as well. More information on this matter and how to handle COTS images can be read here; Nexus as Private Docker Registry - Software Development - Confluence (abnamro.com)[54]. Some base images will be maintained and updated by the CGT (Container Governance Team) in near future.

7.7.1 Proxy repository and AKS 3.0

Nexus supports multiple repository types[55]. A proxy repository is an internal representation of an *external* hosted repository. Proxy repositories are available on Nexus which can be used to retrieve images from 3rd party registries like registry.k8s.io[56] proxied via Nexus via 'docker-k8s-gcr-proxy' or Docker

[53] https://confluence.int.abnamro.com/x/zcrWCg
[54] https://confluence.int.abnamro.com/display/GRIDAD/Nexus+as+Private+Docker+Registry
[55] https://confluence.int.abnamro.com/x/zcrWCg#NexusRepositoryManager-NXRMrepositorytypes
[56] http://registry.k8s.io/

Hub[57] as 'docker-hub-proxy'. Remember that upstream images must be pulled through the Nexus hosted repository. Several proxy repositories are available and new ones can be requested (see 'When my artifact is not available through Nexus').

Proxy repositories can be seen in the Nexus browser Browse - Sonatype Nexus (abnamro.com)[58]. Due to Azure policy*: 'Kubernetes cluster containers should only use allowed images'* AKS only allows images coming from our hosted Nexus repository 'DOCKER_HOSTED_REPO' which is available over port 18443. (https://p-nexus-3.development.nl.eu.abnamro.com:**18443**[59]) which is used to push and pull private (team or grid specific) docker images. Alternative registries which are currently allowed by policy are .azurecr.io (ACR) and mcr.microsoft.com[60].

To consume Images in *AKS* they can be pulled using the proxy repositories via the Nexus group repository 'DOCKER_GROUP_REPO' which is available over port 18445. (https://p-nexus-3.development.nl.eu.abnamro.com:**18445**[61]). You can pull Images when they are not available yet on the Nexus hosted repository 'DOCKER_HOSTED_REPO' or as part of Life Cycle Management using the proxy repository by simply filling e.g "coredns/coredns:*tag*" as <DOCKER_IMAGE_NAME> like this;

```
docker pull p-nexus-3.development.nl.eu.abnamro.com:18445/coredns/coredns
```

To consume images from proxy repositories or COTS images from 3rd party vendor you need to use a Pipeline template Docker[62] that builds a local docker image which you than tag and push to your GRID/ BLOCK related repository on our hosted Nexus repository which acts as a private docker registry. This way the image will be scanned for vulnerabilities using Prisma Cloud[63]. From there you can consume it in AKS. An how to is available here; AKS Ingress Container Images and Nexus[64].

Be aware that the docker flow[65] uses a self hosted Azure Devops Agent[66]. That way it can connect to on-prem resources like Nexus as well as some whitelisted Internet Endpoints[67].

7.7.2 Helm Support

Support for Nexus as a Helm chart repository support is available within AAB see; Nexus for Helm[68]. See this information regarding the helm flow: Helm Pipeline Template[69]

[57] https://hub.docker.com/_/registry/
[58] https://p-nexus-3.development.nl.eu.abnamro.com:8443/#browse/browse
[59] https://p-nexus-3.development.nl.eu.abnamro.com:18445/
[60] http://mcr.microsoft.com
[61] https://p-nexus-3.development.nl.eu.abnamro.com:18445/
[62] https://dev.azure.com/cbsp-abnamro/GRD0001045/_wiki/wikis/PITA%20templates/11782/docker
[63] https://confluence.int.abnamro.com/display/GRIDAD/Prisma+Cloud+Compute+%28PCC%29+FAQ
[64] https://confluence.int.abnamro.com/x/V7HoG#FSCP2to3AKSIngress-ContainerImagesandNexus
[65] https://dev.azure.com/cbsp-abnamro/GRD0001045/_wiki/wikis/PITA%20templates/11782/docker?anchor=yaml-schema
[66] https://confluence.int.abnamro.com/x/LZS5G#FSCP2to3AzureDevOpsAgents-ConnectivityandIntegration
[67] https://confluence.int.abnamro.com/display/GRIDAD/Internet+Access+from+Private+Agents#InternetAccessfromPrivateAgents-Allowedtraffictopublicdomainsoninternet
[68] https://confluence.int.abnamro.com/display/GRIDAD/Nexus+for+Helm
[69] https://dev.azure.com/cbsp-abnamro/GRD0001045/_git/pita-pipeline-templates?path=/docs/flows/helm.md&_a=preview

7.7.3 When my artifact is not available through Nexus

When an image or helm chart dependency is not available through our hosted or proxy repositories you cannot consume the image in AKS. To make this available the first step is to raise a SNG ticket[70] with the SOLO team to request a new proxy repository. For that use the Business service offering: Repository Manager (Nexus) (Prod) with the query 'I want a Nexus Commercial Off-The-Shelf (COTS) Repository to store my artifacts from a 3rd party vendor.' SOLO will check with legal and for the technical feasibility of the request.

If the request for a new proxy repository is denied by SOLO you may need to request to whitelist the internet location for our self hosted DevOps Agents. First check; Allowed traffic to public domains on internet[71] if the url is not yet listed and check the prerequisites for the request Whitelist requests to Internet endpoints[72]. When met raise a SNG ticket with the ADOPT team under Business service offering: Agents (Azure Devops).

7.7.4 Vulnerabilities

Docker images added to Nexus are scanned for vulnerabilities using Prisma Cloud (Twistlock) as part of the docker flow[73]. When you face a Security-Build Breaker because of found critical vulnerabilities your pipeline fails and the build image will not be pushed to Nexus. This might even happen when an image using the same binaries is build and already available in Nexus because the scan can identify new vulnerabilities against the same code base based on new CVE's.

twistcli

```
Scan results for: image p-nexus-3.development.nl.eu.abnamro.com:18445/***

Vulnerabilities found for image p-nexus-3.development.nl.eu.abnamro.com:18445/**** :
total - 8, critical - 1, high - 7, medium - 0, low - 0

Vulnerability threshold check results: FAIL
Scan failed due to vulnerability policy violations: Security-Build Breaker, 1
vulnerabilities, [critical:1]
```

When this happens there are two main routes;

1. Verify if there's a newer version of the upstream image or building blocks which are used to build your docker image and try again.
2. Request for a temporary risk acceptance by; How to start Temporary Risk Acceptance Process

[70] https://aabsiampr.service-now.com/myit?id=myit_support_msg

[71] https://confluence.int.abnamro.com/display/GRIDAD/Internet+Access+from+Private+Agents#InternetAccessfromPrivateAgents-Allowedtraffictopublicdomainsoninternet

[72] https://confluence.int.abnamro.com/display/GRIDAD/Internet+Access+from+Private+Agents#InternetAccessfromPrivateAgents-WhitelistrequeststoInternetendpoints

[73] https://dev.azure.com/cbsp-abnamro/GRD0001045/_wiki/wikis/PITA%20templates/11782/docker?anchor=yaml-schema

7.7.5 resources

Stratus: Container image and Helm chart sources[74]
Nexus Repository Manager[75]
Nexus as Private Docker Registry - Software Development[76]
Pipeline template Docker[77]
Nexus for Helm[78]
Helm Pipeline Template[79]
Prisma Cloud Compute (PCC) FAQ - Software Development - Confluence (abnamro.com)[80]

[74] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81831/Container-image-and-Helm-chart-sources

[75] https://confluence.int.abnamro.com/x/zcrWCg

[76] https://confluence.int.abnamro.com/display/GRIDAD/Nexus+as+Private+Docker+Registry

[77] https://dev.azure.com/cbsp-abnamro/GRD0001045/_wiki/wikis/PITA%20templates/11782/docker

[78] https://confluence.int.abnamro.com/pages/viewpage.action?spaceKey=GRIDAD&title=Nexus+for+Helm

[79] https://dev.azure.com/cbsp-abnamro/GRD0001045/_wiki/wikis/PITA%20templates/11790/helm

[80] https://confluence.int.abnamro.com/display/GRIDAD/Prisma+Cloud+Compute+%28PCC%29+FAQ

8 Cluster Life Cycle Management

For newly created AKS cluster the kubernetesVersion (control plane) and the nodepool version will be the same. During the Life Cycle of the cluster the owning DevOps team is responsible to manage the versioning of the cluster and it's components up to date to prevent vulnerability exploits.

AKSVersion

As a best practice, you should upgrade all node pools in an AKS cluster to the same Kubernetes version. The node pool version must have the same major version as the control plane. The node pool minor version must be within two minor versions of the control plane version. The node pool version cannot be greater than the control plane version. A way could be to keep the control plane and nodepool versions aligned using the approach in the below code block.

for 'aksVersion' both patch version {major.minor.patch} and {major.minor} are supported. When {major.minor} is specified, the latest supported patch version is chosen automatically. Updating the agent pool with the same {major.minor} once it has been created will not trigger an upgrade, even if a newer patch version is available. When you upgrade a supported AKS cluster, Kubernetes minor versions cannot be skipped. All upgrades must be performed sequentially by major version number. For example, upgrades between 1.14.x -> 1.15.x or 1.15.x -> 1.16.x are allowed, however 1.14.x -> 1.16.x is not allowed.

aksVersion

```
  properties: {
    ...
    agentPoolProfiles: [
      {
        ...
        orchestratorVersion: aksVersion # following kubernetesVersion
      }
    ]
    ...
    kubernetesVersion: aksVersion // check (az aks get-versions -l 'westeurope' -o table) for available versions
    ...
  }
```

If you don't specify the orchestratorVersion under agentPoolProfiles you need to maintain the nodepool upgrades manually. For more information to maintain the nodepool aksVersion see upgrading a node pool. az aks nodepool | Microsoft Learn[81]

[81] https://learn.microsoft.com/en-us/cli/azure/aks/nodepool?view=azure-cli-latest#az-aks-nodepool-upgrade

Nodepool Images

Like the AKS version also the nodepool images need to stay up to date. Azure Kubernetes Service (AKS) regularly provides new node images, so it's beneficial to upgrade your node images frequently to use the latest AKS features. Linux node images are updated weekly, and Windows node images are updated monthly. Image upgrade announcements are included in the AKS release notes[82], and it can take up to a week for these updates to be rolled out across all regions. Node image upgrades can also be performed automatically (using specific upgrade channels) and scheduled using planned maintenance. For more details, see Automatically upgrade node images[83].

8.1 resources

Upgrade options for Azure Kubernetes Service (AKS) clusters - Azure Kubernetes Service | Microsoft Learn[84]
Upgrade Azure Kubernetes Service (AKS) node images - Azure Kubernetes Service | Microsoft Learn[85]

[82] https://github.com/Azure/AKS/releases
[83] https://learn.microsoft.com/en-us/azure/aks/auto-upgrade-node-image
[84] https://learn.microsoft.com/en-us/azure/aks/upgrade-cluster
[85] https://learn.microsoft.com/en-us/azure/aks/node-image-upgrade

9 Onboard to FSCP 3.0 AKS[86]

Before you get your hands dirty examine the following docs on the high level changes in FSCP 3 in regards to the MCP product in FSCP 2.0 and the steps involved to setup your AKS cluster.

- Stratus: FSCP 3.0 AKS[87]
- Stratus: Getting Started - Overview (azure.com)[88]
- Stratus: Stratus : AKS 3.0 launch and introduction meeting - ABN AMRO Video[89]

Leverage the section below to setup your cluster step by step which includes the following;

AKS FSCP 3.0 Step by Step

1. Deploy your cluster using the pipeline
2. Connect to your cluster
3. Network connectivity
4. Deploy Applications
5. Deploy Ingress controller
6. Connect resources using AAD

Naming

AKS Cluster name

- Character limit: 1-63, Alphanumerics, underscores, and hyphens. Start and end with alphanumeric.

Application Name

- Added as NodePool Label: The App ID conforms to the agreed format (3 to 5 characters starting with a letter and no additional labels).

[86] https://confluence.int.abnamro.com/display/GRIDAD/Onboard+to+FSCP+3.0+AKS

[87] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/GRD0001007.wiki/69682/FSCP-3.0-a.k.a.-Secure-Context

[88] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81769/Getting-Started

[89] https://www.abnamro.video/media/t/0_rmqdc1qg/16567

9.1 1. Deploy your cluster using the 'QuickStart' pipeline

The FSCP Transition Team created a quickStartAksPipeline[90] which can be leveraged to quick start deploy your cluster. This pipeline can help transition customers with the deployment of an Azure Kubernetes (AKS) cluster in a FSCP 3.0 Landing Zone. This pipeline deploys a compliant AKS cluster with the following components;

1. Main bicep template
   - Key vault with Private Link Endpoint
   - Key Vault Access policies for the Service Principal, DevOps Group and the UMI.
   - Log analytics workspace
   - User assigned managed identity
   - Disk encryption set with Customer Managed Key
2. Self-Service Network Solution custom pipeline task
   - Subnet(s) based on SSNS AKS pattern
   - Inbound NSG rule for private agents
3. AKS bicep template
   - AKS cluster

Checkout the readme[91]for a step by step guideline how to use the pipeline.

9.2 2. Connect to your cluster

For connectivity to you cluster from you workplace see this section; Connect to the AKS control plane (see page 43)

9.3 3. Network connectivity

AKS clusters in FSCP 3.0 have by default a 'deny-all' policy applied provided by the Calico project which disabled all network traffic within the cluster. Generic information and guidance can be found here; FSCP AKS Calico Network Policies[92]

9.4 4. Deploy Applications

Deploying applications should be done from your pipeline which needs either a service connection being setup or by invoking kubectl from the command line. Both options required some steps to be able to use them.

- Service Connection - This requires a service account to setup with the appropriate permissions which than can be used to create a Kubernetes Service Connection which makes deploying stuff to you cluster straight forward. See; Using Kubernetes Service Connection (see page 50)

[90] https://dev.azure.com/cbsp-abnamro/FSCP%20Azure%20Community/_git/FSCPAzureCommunity?path=/FTT-Accelerator/Pipelines/quickStartAksPipeline

[91] https://dev.azure.com/cbsp-abnamro/FSCP%20Azure%20Community/_git/FSCPAzureCommunity?path=/FTT-Accelerator/Pipelines/quickStartAksPipeline&version=GBmain&_a=contents

[92] https://confluence.int.abnamro.com/x/0PkXHQ

- Kubectl - This requires kubectl to install in your pipeline and kubelogin to interact with your cluster each pipeline run. This approach does not require an additional service connection to be set up. But a strict set of commands will be needed for each interaction with your cluster. See; Using Non-interactive login with Kubelogin (see page 50)

9.5 5. Deploy Ingress controller

Ingress exposes HTTPS routes from outside the cluster to services[93] within the cluster. More on ingress options for an AKS cluster can be found here; FSCP AKS Ingress[94].

9.6 6. Connect resources using AAD

Containerized applications can leverage any cloud resource that depends on AAD as an identity provider. More information can be found here; Connect resources using AAD (see page 13)

[93] https://kubernetes.io/docs/concepts/services-networking/service/

[94] https://confluence.int.abnamro.com/x/V7HoG

10 **IaC**

10.1 Bicep

The latest reference (Stratus) template can be found here[95].

Key Bicep template properties in AKS resource,

- enablePrivateCluster: This signifies the cluster created is a private one.
- diskEncryptionSet: diskencryptionSet to be created prior to creating cluster
- userAssignedManagedIdentityId: The User assigned MI which the cluster uses for authentication.
- keyVaultName: Specify the key vault attached to the diskencryotionSet.

[95] https://dev.azure.com/cbsp-abnamro/GRD0001007/_git/mcpk-reference?path=/examples/aks/hello-aks/bicep/resources.bicep

11 Reference Pipeline Link

11.1 FTT Quickstart Pipeline

quickStartAksPipeline - Repos (azure.com)[96] - This is an example YAML pipeline that is provided by the FSCP Transition Team. This pipeline can help transition customers with the deployment of an Azure Kubernetes (AKS) cluster in a FSCP 3.0 Landing Zone.

This pipeline consists of three deployment stage. First the required resource are deployed. Next the Self-Service Network Solution tasks are deployed. And finally the AKS cluster is deployed. It supports the deployment of the same pipeline and templates to different environments. It assumes that the virtual network should contain only one AKS subnet. If none exists it will create one. If there are more than one it will fail. The templates in this repo deploy the bare-minimum required for a successful, compliant deployment and can be used by DevOps teams to kick-start their own Infrastructure-as-Code (IaC) code if using AKS.

Reference AKS Bicep template from Stratus[97]

Terraform Repo Link:

Terraform repo: https://dev.azure.com/cbsp-abnamro/GRD0001007/_git/mcpk-reference?path=/reference_templates/terraform/main.tf

[96] https://dev.azure.com/cbsp-abnamro/FSCP%20Azure%20Community/_git/FSCPAzureCommunity?path=/FTT-Accelerator/Pipelines/quickStartAksPipeline

[97] https://dev.azure.com/cbsp-abnamro/GRD0001007/_git/mcpk-reference?path=/examples/aks/hello-aks/bicep/resources.bicep

12 Policy Link

AAB Azure Kubernetes Service v1 - Overview[98]
Index of policy-driven values for parameters (Stratus)[99]

[98] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/67884/AAB-Azure-Kubernetes-Service-v1

[99] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/83075/Index-of-policy-driven-values-for-parameters

13 Limitations

Disclaimer

As of 📅 31 Mar 2023 AKS is available in pre-production as in production. Details on AKS roadmap can be found here; AKS 3.0 Roadmap[100]

Connecting to AKS cluster is so far not yet standardized. Enabling private link will create different option to connect to the cluster.

- Using VDI/MacOS
- Using Windows Laptops
  - Windows Laptops using Cisco Any Connect provide issues connecting the cluster over PoSh/Bash. Command prompt is possible.
- Using A VM with in the same Azure Network
  - Deploy a VM in the same network as AKS and use this as a remote management server
  - Use virtual network peering to connect your management network to the AKS network
  - Use Express Route or VPN to connect your on-premises network to the AKS network
  - Use the AKS command invoke feature[101] to run commands remotely on your AKS cluster

[100] https://confluence.int.abnamro.com/x/akAvGg

[101] https://docs.microsoft.com/en-us/azure/aks/command-invoke

14 Exemptions/Special cases

none

15 How To

15.1 Perform specific tasks or actions (Stratus Wiki)

How-Tos - Overview (azure.com)[102]

15.2 Availability Zones in AKS

Availability Zones[103] are a high-availability offering that protects your applications and data from datacenter failures and **increases resiliency**. Each zone includes one or more datacenters equipped with independent power, cooling, and networking. The physical separation of availability zones within a region protects applications and data from datacenter failures.

See guidance from Stratus; Availability Zones in AKS - Overview (azure.com)[104]

15.3 Enable Zone redundancy on **existing cluster**.

If you want to enable zone redundancy on **existing cluster** please follow below steps (this Is reference taken form One of the team. and they have created azure CLI task using below scripts)

1) delete user Node Pool in the cluster.

```
az aks nodepool delete --cluster-name <<AKS Cluster Name>> --name system --resource-
group <<RG Name>> --no-wait
```

2) Add a new System node Pool (zone option is specified while adding node pool)

```
export MSYS_NO_PATHCONV=1
az aks nodepool add \
-g <<rg Name>> \
-n systems \
--cluster-name <<cluster name>> \
--mode System \
```

[102] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81770/How-Tos
[103] https://learn.microsoft.com/en-us/azure/aks/availability-zones
[104] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/95491/Availability-Zones-in-AKS

```
--os-type Linux \
--kubernetes-version 1.29.2 \
--node-count 3 \
--labels abnamro.bank/application-name=XYZ \
--max-pods 30 \
--node-vm-size Standard_D4s_v5 \
--enable-encryption-at-host \
--zones 1 2 3 \
--vnet-subnet-id '/subscriptions/<<subscription Id>>/resourceGroups/<<Virtual network
RG Name>>/providers/Microsoft.Network/virtualNetworks/<<Virtual network Name>>/
subnets/<<aks-subnet-name>>'
```

3) Delete Old system node Pool

4) Add a new user node Pool

```
export MSYS_NO_PATHCONV=1
az aks nodepool add \
-g <<rg-name>> \
-n batches \
--cluster-name <<clsuter-name>> \
--mode User \
--os-type Linux \
--kubernetes-version 1.29.2 \
--node-count 3 \
--labels abnamro.bank/application-name=<<application-name>> \
--min-count 3 \
--max-count 5 \
--max-pods 30 \
--node-vm-size Standard_D16s_v3 \
--enable-cluster-autoscaler \
--enable-encryption-at-host \
--zones 1 2 3 \
--node-osdisk-type Managed \
--vnet-subnet-id '/subscriptions/<<subscription Id>>/resourceGroups/<<VNet Resource
group>>/providers/Microsoft.Network/virtualNetworks/<<Virtual Network Name>>/subnets/
<<aks-subnet>>'
```

5) get VMSS permission for virtual machine scale sets. (for Pod Identity)

6) restart deployment(Service) so that all deployment will run on user node pool using node selector

15.4 Kubeconfig needed for AKS 1.24 and onwards

Starting with Kuberneets 1.24 kubelogin is required to access the cluster from command line or DevOps task. There are several ways which can be used to accomplish this;

- Kubernetes Task by creating a ServiceAccount on your cluster and then assign the correct role and a Service Connection in ADO. See further down in this article under "Create Service Account and binding it to a cluster role[105]".
- Non-interactive login with a bash example: Non-interactive login with Kubelogin[106]

Addition Resources:

Kubectl task - Azure Pipelines & TFS | Microsoft Docs[107]
https://github.com/Azure/kubelogin#user-principal-login-flow-non-interactive

15.5 Connect to the AKS control plane

15.5.1 Proxy Settings

VPN Client

23 Sep 2022 The Proxy Settings below are only needed when a VPN connected device is used in conjunction with the Cisco Anywhere Connected client or a VDI. Windows workstations using the Zscaler Client Connector do not need additional proxy configuration.

To control the private AKS using kubectl the following settings need to be in place as a minimum. For more info also see Connecting to your cluster[108] and Proxy Setup, Principles and Guidelines (October 2022)[109]

Environment variables (CMD)

Environment Variables - Command prompt

```
# (For Azure cli to work)
set HTTPS_PROXY=http://nl-userproxy-access.net.abnamro.com:8080
# (Connect directly to private API servers)
set NO_PROXY=*.abnamro.com,kubernetes.docker.internal,*.hcp.westeurope.azmk8s.io
```

Environment variables (Powershell and Bash)

[105] https://confluence.int.abnamro.com/x/pVJhFw

[106] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81833/Non-interactive-login-with-Kubelogin

[107] https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/deploy/kubernetes?view=azure-devops

[108] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/GRD0001007.wiki/64199/Step-3-Connecting-to-your-cluster

[109] https://confluence.int.abnamro.com/pages/viewpage.action?pageId=215899283

Known Issue

As of 17 August 2022 connecting to the control plane from a VPN connected Windows AAB Laptop leveraging Cisco Anywhere Client connecting to a cluster using Powershell or git bash is not working. A workaround is to use a Virtual Desktop which can be request through AGF.

Powershell

Environment Variables - Powershell

```
# (For Azure cli to work)
[Environment]::SetEnvironmentVariable("HTTPS_PROXY", "http://nl-userproxy-
access.net.abnamro.com:8080", "User")
# (Connect directly to private API servers)
[Environment]::SetEnvironmentVariable("NO_PROXY",
"*.abnamro.com,kubernetes.docker.internal,*.hcp.westeurope.azmk8s.io", "User")
```

bash

Environment Variables - git Bash

```
# (For Azure cli to work)
export HTTPS_PROXY=http://nl-userproxy-access.net.abnamro.com:8080
# (Connect directly to private API servers)
export NO_PROXY=*.abnamro.com,kubernetes.docker.internal,*.hcp.westeurope.azmk8s.io
```

15.5.1.1 Resources

Bye Bye Cisco AnyConnect, hello ZScaler Private Access! (sharepoint.com)[110]

15.5.2 Install Kubectl

1. To be done on your workstation. First install 'Azure cli' package from Software Center. *If you have installed the package manually before re-install the package from Software Center to get the settings in place correctly.*

2. Install cli tooling using the following command

[110] https://abnamro.sharepoint.com/sites/intranet-informatie_it/SitePages/en/Bye-Bye-Cisco-AnyConnect%2C-hello-ZScaler-Private-Access!.aspx

Install kubectl

```
az aks install-cli
```

3. The following path locations should be added to the %PATH% variable

Path variable

```
$oldpath = (Get-ItemProperty -Path 'Registry::HKEY_CURRENT_USER\Environment').Path
$newpath = "$oldpath;$env:userprofile\.azure-kubectl;$env:userprofile\.azure-kubelogin"
Set-ItemProperty -Path 'Registry::HKEY_CURRENT_USER\Environment' -Name PATH -Value $newpath
```

15.5.3 Install kubelogin manually

> *When you have installed kubectl using the section above (az aks install-cli), kubelogin is installed as well and these steps are not needed.*

Check the following link for the most recent releases: Releases · Azure/kubelogin (github.com)[111].

Install kubelogin on Windows

```
md $env:userprofile'\.kube'
Invoke-WebRequest -Uri "https://github.com/Azure/kubelogin/releases/download/v0.0.25/kubelogin-win-amd64.zip" -OutFile "$env:userprofile\.azure-kubelogin\kubelogin-win-amd64.zip"
Expand-Archive -LiteralPath "$env:userprofile\.azure-kubelogin\kubelogin-win-amd64.zip" -DestinationPath "$env:userprofile\.azure-kubelogin\kubelogin.exe"
[Environment]::SetEnvironmentVariable("KUBECONFIG", "$env:userprofile\.kube\config", "User")
$oldpath = (Get-ItemProperty -Path 'Registry::HKEY_CURRENT_USER\Environment').Path
$newpath = "$oldpath;$env:userprofile\.azure-kubelogin"
Set-ItemProperty -Path 'Registry::HKEY_CURRENT_USER\Environment' -Name PATH -Value $newpath
```

15.5.4 Steps to connect from your workstation

When you try to connect from your workstation you can use the following method.

The minimum required permissions are:

[111] https://github.com/Azure/kubelogin/releases

- Azure Kubernetes Service RBAC Reader

Connect to your cluster

```
az login --use-device-code
az account set --subscription <id-or-name>
az aks get-credentials --subscription <id-or-name> -g <rg-name> -n <cluster-name>
kubelogin convert-kubeconfig -l azurecli

# example command
Kubectl get pod
```

15.6 Ingress

Ingress[112] exposes HTTPS routes from outside the cluster to services[113] within the cluster. More on ingress options for an AKS cluster can be found here[114].

15.7 Kustomize

'Kustomize' is a configuration management tool (part of kubectl) for kubernetes manifest files. Kustomize provides the ability to create distinctive 'otap' manifest files based on a generic base yaml file and 'patches' in so-called 'overlay' directory structure. This allows easy differentiation between development and production manifest files.

kustomize

The trigger and configuration file for kustomize is the 'kustomize.yml'. This shows where the 'base' is located in relation to the 'overlay' folder. In addition, it describes, among other things, which patch strategy is applied, or which namespace or image is to be applied.

An example on how kustomize can be used to deploy an ingress controller can be found here; AKS Ingress[115]

15.7.1 Structure

Sub-directories of `overlays/` are Kustomizations, and as such:

- must have a `kustomization.yaml` file
- use the `Kustomization` kind

[112] https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.24/#ingress-v1-networking-k8s-io
[113] https://kubernetes.io/docs/concepts/services-networking/service/
[114] https://confluence.int.abnamro.com/x/V7HoG
[115] https://confluence.int.abnamro.com/x/V7HoG#AKSIngress-Kustomize

The folder structure can be like this:

folder structure

```
<root>/
+- overlays/          # Kustomize settings per environment (DTAP)
¦  +- base/          # common settings inherited by all environments
¦  +- development/
¦  +- test/
¦  +- acceptance/
¦  +- production/
|
```

The `overlays/common` is used to reference components that are used across all environments. Any common settings that are not environment-dependant is also done here.
Each DTAP environment has an overlay. These overlays inherit directly from the `overlays/common`. Any environment-specific resource/configuration must be configured on the respective directory.

15.7.2 Patch strategy with Kustomize

15.7.2.1 Patches[116]

'Patches' provides the ability to apply inline patches *(formerly known as patchesJson6902 and patchesStrategicMerge*), among others, which can make defining complex though. Providing a patch manifest as 'target', the target resource is matched with the 'apiVersion', 'kind' and 'name' from the patch. Using 'patches' is the easiest to use. If 'apiVersion', 'child' and 'name' cannot be matched, inline 'patches' by using a JSON match pattern can be chosen and a 'target' manifest file can be omitted.

Each entry in this list must be either a relative file path or an inline definition. The entries 'apiVersion', 'name' and 'child' are used to patch (modify) resources.

Example:

Kustomization.yml overlay:

```
patches:
- templatejsonpatch.yml
```

[116] https://kubectl.docs.kubernetes.io/references/kustomize/kustomization/patches/

templatejsonpatch.yml

```
kind: Service
apiVersion: v1
metadata:
  name: ingress-nginx
  namespace: ns-at-n-cf-0021
spec:
  clusterIP: 10.0.255.254
  loadBalancerIP: 10.234.246.254
```

A inline 'patch' allows different 'operations' to be applied to json/yaml files such as adding, replacing or deleting entries.

Example:

templatejsonpatch.yml

```
patches:
- path: templatepatchesJson6902.yml
  target:
  group: ''
  version: v1
  kind: Deployment
  name: currentname
```

templatepatchesJson6902.yml

```
[
  {"op": "replace",
   "path": "/metadata/name",
   "value": "newname"}
]
```

15.7.2.2 Namespaces[117]

Will overwrite the existing namespace if it is set to a resource, or add it if it is not set to a resource.

[117] https://kubectl.docs.kubernetes.io/references/kustomize/kustomization/namespace/

kustomization.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

namespace: kustomize-namespace
```

15.7.2.3 Images[118]

Images changes the name, tags and/or digest for images without creating patches.

kustomization.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

namespace: kustomize-namespace
```

15.7.3 Resources

- Kustomize Glossary[119]
- Kustomization file[120]
- Releases[121]

15.7.4 Usage

Overlays can be applied using:

- `kubectl apply -k overlays/<name>`
- `kustomize build overlays/<name> | kubectl apply -f -`

15.8 Deploying Application to the AKS Cluster

Application and deploying AKS resources can be deployed to the cluster in various ways.

[118] https://kubectl.docs.kubernetes.io/references/kustomize/kustomization/images/
[119] https://kubectl.docs.kubernetes.io/references/kustomize/glossary/
[120] https://kubectl.docs.kubernetes.io/references/kustomize/kustomization/
[121] https://github.com/kubernetes/ingress-nginx/releases

15.8.1 Using Portal way of Deployment (DEV only)

YAML manifests can be deployed manually to AKS cluster using the Azure Portal. This can be done for quick verification only as part of a POC.

Steps-

- Login to Azure Portal
- Navigate to the created AKS cluster.
- Go to Workloads section and Choose Create with YAML option for creating pod/deployment specs.
- Follow similar route for creating other resources

15.8.2 Using Kubernetes Service Connection

A kubernetes service connection is used as a straight forward method to deploy resources to your cluster. It is a pre-requisite for helm deployments using pita templates. Once the service connection is created you can use kubernetes task in Azure Devops (ADO) to deploy your application. Kubernetes tasks provided by ADO has Kubectl bundled with it. So it will automatically takes care of running Kubectl commands to you. No need to install Kubectl and Kubelogin unless you use Kubectl commands explicitly via shell command.

See: Create a DevOps Kubernetes Service Connection (see page 54)

Azure Devops AzureCLI Task

```
- task: Kubernetes@1
  displayName: kubectl apply kubernetes manifest
  condition: true
  inputs:
    connectionType: Kubernetes Service Connection
    kubernetesServiceEndpoint: 'aec-aks-ftt-poc-k8s'
    command: apply
    useConfigurationFile: true
    configuration: '$(System.DefaultWorkingDirectory)/Services/AzureKubernetesService/ingress-nginx.yml'
    namespace: 'ingress-nginx'
```

15.8.3 Using Non-interactive login with Kubelogin

The below task can be used to connect to your cluster using kubectl for cluster v1.24 and onwards where kubelogin is mandatory. The example requires the *'Private Pool Deployment Docker'* pool. This task is using the approach provided by Stratus where additional information and examples like how to interact using MSI can be found as well; Non-interactive login to AKS cluster -[122]Overview (azure.com)[123]. This approach leverages Nexus to prevent rate limit issues while downloading the software from Internet directly.

Be aware that for every interaction with the cluster this installation will be needed. For recurring deployments use the kubernetes service connection instead.

prerequisite:

- Your resource group SPN (Service Connection) needs to have RBAC permissions on the cluster e.g. 'Azure Kubernetes Service RBAC Cluster Admin'.

Azure Devops AzureCLI Task

```
#####################################################################################
#######
# Stage: Connect CLuster
#####################################################################################
#######

- stage: connect_cluster
  displayName: Connect Cluster using Kubelogin
  variables:
    - template: ./parameter/${{parameters.targetEnvironment}}.variables.yml

  jobs:
  - job: connect_cluster_example
    displayName: Install Tools and Connect
    steps:

    - task: Bash@3
      displayName: Install tools
      inputs:
        targetType: 'inline'
        script: |
          TARGETOS=$(echo "$(Agent.OS)" | tr '[:upper:]' '[:lower:]')
          TARGETARCH=$(case "$(Agent.OSArchitecture)" in "X86") echo "x86";; "X64")
echo "amd64";; "ARM") echo "arm64";; esac)
          mkdir -p "$(Pipeline.Workspace)/.local/bin"
          if [ $(Kubectl.Version) == 'latest' ];then
            latestVersion=$(curl -L -s $(Solo.Nexus3.Repositories.Uri)repository/
generic-group/stable.txt)
            echo "kubectl download latest"
            curl -LO $(Solo.Nexus3.Repositories.Uri)repository/generic-group/
$latestVersion/bin/$TARGETOS/$TARGETARCH/kubectl
            mv ./kubectl $(Pipeline.Workspace)/.local/bin/
```

[122] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81833/Non-interactive-login-to-AKS-cluster?anchor=using-active-directory-cluster-user-credentials-tagged-to-service-principal

[123] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81833/Non-interactive-login-to-AKS-cluster?anchor=using-active-directory-cluster-user-credentials-tagged-to-service-principal

```
     else
      echo "kubectl download $(Kubectl.Version)"
      curl -LO $(Solo.Nexus3.Repositories.Uri)repository/generic-group/v$(Kubectl.Version)/bin/$TARGETOS/$TARGETARCH/kubectl
      mv ./kubectl $(Pipeline.Workspace)/.local/bin/
     fi

     if [ $(Kubelogin.Version) == 'latest' ];then
      echo "kubelogin download latest"
      curl -LO $(Solo.Nexus3.Repositories.Uri)repository/generic-group/latest/download/kubelogin-$TARGETOS-$TARGETARCH.zip
     else
      echo "kubelogin download v$(Kubelogin.Version)"
      curl -LO $(Solo.Nexus3.Repositories.Uri)repository/generic-group/download/v$(Kubelogin.Version)/kubelogin-$TARGETOS-$TARGETARCH.zip
     fi

     unzip kubelogin-$TARGETOS-$TARGETARCH.zip
     mv ./bin/"$TARGETOS"_"$TARGETARCH"/kubelogin $(Pipeline.Workspace)/.local/bin/
     rm kubelogin-$TARGETOS-$TARGETARCH.zip
     ls -la $(Pipeline.Workspace)/.local/bin/
     chmod +x $(Pipeline.Workspace)/.local/bin/*
     echo "##vso[task.setvariable variable=PATH]${PATH}:$(Pipeline.Workspace)/.local/bin"

  - task: AzureCLI@2
   displayName: 'Kubelogin Example'
   enabled: true
   inputs:
    azureSubscription: ${{ variables.resourceGroup }}
    addSpnToEnvironment: true
    scriptType: 'bash'
    scriptLocation: 'inlineScript'
    inlineScript: |
     # set -euox pipefail
     # Get AKS credentials
     az aks get-credentials -g ${{ variables.resourceGroup }} \
      -n ${{ variables.clusterName }} \
      --subscription ${{ variables.subscriptionName }} \
      --overwrite-existing
     # Convert AKS credentials to kubelogin-style
     kubelogin convert-kubeconfig -l spn \
      --client-id $servicePrincipalId \
      --client-secret $servicePrincipalKey \
      --tenant-id $tenantId

     # Test use of kubectl
     kubectl get pod
     kubectl cluster-info
     kubectl get nodes
     # List all services in the namespace
```

```
          kubectl get services
          # List all pods in all namespaces
          kubectl get pods --all-namespaces
          # Get all running pods in the namespace
          kubectl get pods --field-selector=status.phase=Running
          # Show labels for all pods (or any other Kubernetes object that supports
labelling)
          kubectl get pods --show-labels
          # Show API Server IP Address
          kubectl get endpoints --namespace default kubernetes
```

15.8.4 Using Pita Templates along with Helm

Pita templates (flows/blocks) should be used as much as possible to deploy like Helm templates. Nexus supports Helm and can be used as Helm chart repositories.

Wiki - Pipeline templates link[124]
Helm Templates: helm - Repos (azure.com)[125]

Azure Devops Helm Deployment

```
- stage: HelmPackage
    displayName: Build and Save Helm Chart
    variables:
      - template: 'vars-d.yml'
    jobs:
      - template: flows/helm-package.yml@templates
        parameters:
          versioning: 'use-published'
          chart_path:  'helm'
          acr_resource_group_sc: ${{ variables.resource_group_sc }}
          #  Other parameters are taken through variables, see chapter on Parameters
and variables
  - stage: HelmDeployDev
    displayName: Download and deploy Helm Chart
    variables:
      - template: 'vars-d.yml'
    jobs:
      - deployment: Deploy
        displayName: Deploy to Helm Chart
        #   environment: cbe-aks-d-sc-dev
        #environment: ${{ variables.environment }}    #cbe-d-cluster    cbe01-d-aks
        environment: 'ftt-dev'
        pool:
          #vmImage: 'ubuntu-20.04'
          name: Private Pool Deployment
```

[124] https://dev.azure.com/cbsp-abnamro/GRD0001045/_wiki/wikis/PITA%20templates/11780/pipeline-templates
[125] https://dev.azure.com/cbsp-abnamro/GRD0001045/_git/pita-pipeline-templates?path=/blocks/helm

```
        strategy:
          runOnce:
            deploy:
              steps:
                - template: /blocks/helm/deploy.yml@templates
                  parameters:
                    acr_resource_group_sc: ${{ variables.resource_group_sc }} #
Mandatory; Serviceconnection to resource group which contains ACR
                    kubernetes_sc: ${{ variables.kubernetes_sc }}  # Mandatory;
Kubernetes Serviceconnection to specific deploy rights to AKS cluster and the
namespace
                    chart_version: ${{ variables.imageTag }} # Mandatory; Chart
version to download from ACR
                    #-- Optional: (if values are available as variable)
                    acr_name: ${{ variables.acr_name }} # Optional; Name of the ACR,
also respects the variable $(AcrName)
                    chart_name: '$(ChartName)' # Optional; Name of the Chart, also
respects the variable $(ChartName)
                    namespace: ${{ variables.namespace }}  # Optional; Namespace to
deploy chart into AKS, also respects the variable $(Namespace)
                    release_name: '$(ReleaseName)' # Optional: Release name for the
helm deployment, also respects the variable $(ReleaseName)
                    override_values: 'image.tag=${{ variables.imageTag }}' #
Optional: Used to override values or secrets.
```

Apart from above ways now Devops teams are free to use other tools like **Kustomize**[126]also for deploying resources to AKS.

15.9 Create an DevOps Kubernetes Service Connection

In order to access the AKS cluster from Azure devops pipeline we should create a service connection. By default, the resource group SPN you used in the pipeline to deploy the AKS cluster is the owner of that resource. That means that you can use this SPN to interact with the cluster using kubelogin. Alternatively you can look at creating your own `ServiceAccount`, `(Cluster)Role` and `(Cluster)RoleBinding` reso urces in order to get the details needed to create an Azure DevOps (Kubernetes) Service Connection to deploy resources to the cluster directly.

Creation of the Azure DevOps Service Connection can be either manually created or can be automated via a Pipeline.

15.9.1 ***Automated***:

- Look at the cookbook subpage; FSCP AKS Service Connection Automation[127]

[126] https://github.com/kubernetes-sigs/kustomize
[127] https://confluence.int.abnamro.com/x/NZcdHg

15.9.2 ***Manual***:

- The steps below;

Manually creating a Kubernetes Service Connection

In order to create it the following steps need to be passed through.

- Setting up Custom Cluster Role for least privilege.
- Creating and binding a new service account to the custom cluster role.
- Testing the permissions of the service account.
- Adding the service account into Kubernetes Service Connection.
- Referencing the Service connection in a build pipeline.

15.9.2.1 **Creating Custom cluster role in Kubernetes**

In majority of the cases the built in role **cluster-admin** the most commonly used role and what people tend to use by default as it provides full admin access to the cluster. We want to reduce this and insure we provide least privilege access to the cluster. Kubernetes RBAC model does allow very granular role based access model, with the ability to define the level of access (known as verb) for every resource type available. The clusterrole represents a set of permissions, which is non-namespaced, cluster-wide. See for more info AKS RBAC Roles

> *If you want to define a role within a namespace, use a Role; if you want to define a role cluster-wide, use a ClusterRole.*

There are mainly 7 main verb types available for each resource `[`**`'get', 'list', 'watch', 'create', 'update', 'patch', 'delete'`**`]`.
More information about the main verb types and some specials can be found here[128].

Following is a sample manifest to create a custom role.

Create cluster role

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
 # "namespace" omitted since ClusterRoles are not namespaced
 name: azure-devops-clr
rules:
- apiGroups: ["*"]
  resources: ["deployments","pods","namespaces","services","secrets","replicasets","networkpolicies","serviceaccounts","roles","clusterroles"]
```

[128] https://kubernetes.io/docs/reference/access-authn-authz/authorization/

```
    verbs: ["get","list","watch","create","update","patch","apply","delete"]
```

15.9.2.2 Create Service Account and binding it to a cluster role

Post the cluster role creation we need to have a binding created with the subject i.e service account. A role binding grants the permissions defined in a role to a user, service account or set of users.

Following is a sample for creating the service account -

Create service account

```
apiVersion: v1
kind: ServiceAccount
metadata:
 name: azure-devops-svc
```

A RoleBinding grants permissions within a specific namespace whereas a ClusterRoleBinding grants that access cluster-wide. Below an example on how a ClusterRoleBinding is set using YAML.

clusterRoleBinding Yaml

```
apiVersion: rbac.authorization.k8s.io/v1
# This cluster role binding allows service account 'azure-devops-svc' cluster-wide
permission defined in role 'azure-devops-conn'.
kind: ClusterRoleBinding
metadata:
  name: azure-devops-role-binding-svc
subjects:
- kind: ServiceAccount
  name: azure-devops-svc
  namespace: default
roleRef:
  kind: ClusterRole
  name: azure-devops-clr
  apiGroup: rbac.authorization.k8s.io
```

And an example to provide the same outcome from command line using kubectl.

clusterRoleBinding kubectl

```
 kubectl create clusterrolebinding azure-devops-role-binding-svc --clusterrole=azure-
devops-clr --serviceaccount=default:azure-devops-svc
clusterrolebinding.rbac.authorization.k8s.io/azure-devops-role-binding-svc
```

15.9.2.2.1 New in Kubernetes 1.24 and up

From Kubernetes 1.24 onwards when you create service accounts, non-expiring service account tokens are no longer implicitly generated for every service account. To create a long-lived token to be used in the service connector you have to create a secret with a specific annotation and bound to the service account. more details can be found here Configure Service Accounts for Pods | Kubernetes[129].

secret

```
apiVersion: v1
kind: Secret
type: kubernetes.io/service-account-token
metadata:
  name: azure-devops-token
  annotations:
    kubernetes.io/service-account.name: azure-devops-svc
```

15.9.2.3 Resources

Using RBAC Authorization | Kubernetes[130]

Service Account Tokens in Kubernetes v1.24 | D2iQ Engineering[131]

15.9.2.4 **Get the exact secret contents**

Now we need to output the contents of the secret which is used by the Azure Devops service connector to authenticate to the cluster.

Get the secret contents

```
kubectl get secret azure-devops-token -n default -o json
```

15.9.2.5 **Get the API server URL**

The last item will be the fqdn for the (private) API server address needed for the service connection.

[129] https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/#manually-create-a-long-lived-api-token-for-a-serviceaccount

[130] https://kubernetes.io/docs/reference/access-authn-authz/rbac/#role-and-clusterrole

[131] https://eng.d2iq.com/blog/service-account-tokens-in-kubernetes-v1.24/#whats-changed-in-kubernetes-v124

Get the secret contents

```
kubectl config view --minify -o jsonpath="{.clusters[0].cluster.server}"
```

15.9.2.6 Requirements to apply manifests from the pipeline

Make sure the Azure Devops agent pool has inbound access to the AKS Subnet NSG. You will need to use a self-hosted private pool and the respective subnet needs to be allowed on the NSG.
See this link Usage Guidance - Overview (azure.com)[132] for an example. And this link to check the needed subnets Internet Access from Private Agents[133]

15.9.2.7 Apply the API Address & secret in the Azure Devops Kubernetes service Connection

1. Create or edit a Kubernetes service connection.
2. You need to choose the 'Service Account' option.
3. Paste the output of the API address under server Url.
4. Paste the output of the secret contents in the textbox under 'secret'.

15.10 Allow pod communication using calico network policies.

See the How To section around Calico network policies here; FSCP AKS Calico Network Policies How To[134]

15.11 Connect AKS to Azure Files

In AKS the Azure Files Container Storage Interface (CSI) driver is available to manage the lifecycle of Azure file shares. The driver is enabled as 'fileCsiDriver' under the AKS storageProfile. The CSI driver uses the configured identities to access the storage account. manually managing secrets is not needed.

> ***Best practice guidance***
>
> *To reduce management overhead and enable scaling, avoid statically create and assign persistent volumes. Use dynamic provisioning. In your storage classes, define the appropriate reclaim policy to minimize unneeded storage costs once pods are deleted.*

[132] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/59377/Usage-Guidance?anchor=nsg-inbound-to-ple-subnet-from-private-host-agent-(example-given-on-443)

[133] https://confluence.int.abnamro.com/pages/viewpage.action?spaceKey=GRIDAD&title=Internet+Access+from+Private+Agents

[134] https://confluence.int.abnamro.com/x/0PkXHQ#FSCP2to3AKSCalicoNetworkPolicies-HowTo

Azure Files Container Storage Interface (CSI)

To be able to mount a volume on a pod which is hosted on a storage account using a private link endpoint we need the following list of prerequisites.

- NSG outbound rule port 445 with source AKS Subnet and destination PLE Subnet.
- NSG inbound rule port 445 with source AKS Subnet to destination PLE Subnet
- Storage Account Contributor for AKS User assigned Managed Identity (--query "[id, identity]")
- Storage File Data SMB Share Reader for AKS User assigned Managed Identity (--query "[id, identity]")
- Storage Account Key Operator Service Role for (--query "[id, identityProfile]")
- Enabled Access Keys on Storage Account.
    - The CSI driver needs read access to the access keys using the kubelet Identity. Therefore the 'allowSharedKeyAccess' needs to be enabled on the storage account. This will result in a non-compliant **Informational**. See the statement in the storage account cookbook.[135]
- Custom 'storageClass'
- 'PersistentVolumeClaim' using the custom 'storageClass'
- volume Mount

az aks show

```
az aks show -g <resource-group> -n <cluster-name> --query "[id, identity]"
```

15.11.1 Storage Class

If your Azure Files resources are protected with a private endpoint, you must create your own storage class that's customized with the following parameters:

- `resourceGroup` : The resource group where the storage account is deployed.
- `storageAccount` : The storage account name.
- `server` : The FQDN of the storage account's private endpoint (for example, `<storage account name>.privatelink.file.core.windows.net`[136]).

[135] https://confluence.int.abnamro.com/x/1ycBFg#FSCP2to3StorageAccount-Usageofaccesspolicies

[136] http://privatelink.file.core.windows.net

storageClaim

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: private-azurefile-csi
provisioner: file.csi.azure.com
allowVolumeExpansion: true
parameters:
  shareName: aksshare
  resourceGroup: lzftt-d-rg
  storageAccount: sfeakssa01
  server: sfeakssa01.privatelink.file.core.windows.net
  storeAccountKey: 'false'
reclaimPolicy: Retain # Default is Delete
volumeBindingMode: Immediate
mountOptions:
  - dir_mode=0777
  - file_mode=0777
  - uid=0
  - gid=0
  - mfsymlinks
  - cache=strict  # https://linux.die.net/man/8/mount.cifs
  - nosharesock  # reduce probability of reconnect race
  - actimeo=30  # reduce latency for metadata-heavy workload
```

15.11.2 Persistent volume claim

The claim using the custom storageClass

persistentVolumeClaim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: private-azurefile-pvc
  namespace: private-azfiles
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: private-azurefile-csi
  resources:
    requests:
      storage: 1Gi
```

15.11.3 Volume mount

piece of the pod spec using the persistent volume claim.

volumeMount

```
spec:
  containers:
    volumeMounts:
      - name: azure
        mountPath: /mnt/azurefile
  volumes:
  - name: azure
    persistentVolumeClaim:
      claimName: private-azurefile-pvc
```

15.11.4 Example validation

Use the following example pod manifest with the volume mount using the persistentVolumeClaim.

mypod

```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    resources:
      requests:
        cpu: 100m
        memory: 128Mi
      limits:
        cpu: 250m
        memory: 256Mi
    volumeMounts:
    - mountPath: "/mnt/azure"
      name: azure
  volumes:
    - name: azure
      persistentVolumeClaim:
        claimName: private-azurefile-pvc
```

To see if you can write to the volume you could use the following commands

Validation

```
kubectl exec -it mypod -- touch /mnt/azurefile/test.txt
kubectl exec -it mypod -- ls -l /mnt/azurefile
```

15.11.5 resources

https://learn.microsoft.com/en-us/azure/aks/csi-storage-drivers#enable-csi-storage-drivers-on-an-existing-cluster
https://learn.microsoft.com/en-us/azure/aks/azure-files-csi
https://learn.microsoft.com/en-us/azure/aks/concepts-storage#persistent-volume-claims
https://learn.microsoft.com/en-us/troubleshoot/azure/azure-kubernetes/fail-to-mount-azure-file-share
https://learn.microsoft.com/en-us/azure/aks/azure-files-csi#use-a-persistent-volume-with-private-azure-files-storage-private-endpoint

15.12 Secrets Management

Consuming secrets, keys and certificates in FSCP 3.0 has changed. *In FSCP 2.0, AKS was integrated with Hashicorp Vault for this purpose. Hashicorp Vault is no longer desirable and has been effectively deprecated. In its place, we will use Azure Key Vault and the supplied integration with Kubernetes*.

15.12.1 Azure Key Vault Provider for Secrets Store CSI

Although **not mandatory** you may want to leverage the Azure Key Vault Provider for Secrets Store CSI (Container Storage Interface) Driver to allow for the integration of an Azure key vault as a secret store with an Azure Kubernetes Service (AKS) cluster via a CSI volume. Check Secrets management on AKS - Stratus[137] for addition info.

15.12.1.1 Enable Azure Key Vault Provider on your AKS Cluster

To check if the add-on is enabled on your cluster

[137] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81834/Secrets-management-on-AKS

query azureKeyvaultSecretsProvider

```
az aks show -g <resource-group> -n <cluster-name> --query
addonProfiles.azureKeyvaultSecretsProvider.enabled
```

If not 'true' you can enable the add-on by adding it to the addonProfiles section in the AKS deployment template.

azureKeyvaultSecretsProvider

```
resource example 'Microsoft.ContainerService/managedClusters@2022-09-02-preview' = {
  name: 'exampleCluster'
  location: 'west-europa'
  tags: {
    ...
  }

  ...

  properties: {
    addonProfiles: {
      azureKeyvaultSecretsProvider: {
        enabled: true
      }
    }

    ...

  }
}
```

The Secrets Store CSI Driver allows for the following methods to access an Azure key vault:

- An Azure Active Directory workload identity[138] - FSCP AKS Workload Identity[139]
- A user-assigned or system-assigned managed identity[140]

15.12.2 Setting up Key Vault integration using the Secrets Store CSI Driver

Leveraging Entra ID Workload Identity

[138] https://learn.microsoft.com/en-us/azure/aks/workload-identity-overview

[139] https://confluence.int.abnamro.com/x/sksQHw

[140] https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-identity-access#access-with-a-user-assigned-managed-identity

The following steps provide an example on how to integrate a Key Vault using Entra ID workload Identity

15.12.2.1 Prerequisites

To be able to mount the secrets store volume on a pod using a private link endpoint connected Key Vault integration we need the following list of prerequisites.

- NSG outbound rule port 443 with source AKS Subnet and destination PLE Subnet.
- NSG inbound rule port 443 with source AKS Subnet to destination PLE Subnet
- Setup Entra ID Workload Identity: FSCP AKS Workload Identity[141]

15.12.2.2 SecretProviderClass

First we create a 'SecretProviderClass'. We're leveraging the workload identity.

- Namespace
- The name of the secret object in the key vault
- The object type (secret, key, or certificate)
- The name of your Azure key vault resource
- The Azure tenant ID that the subscription belongs to

secretProviderClass

```
# This is a SecretProviderClass example using user-assigned identity to access your key vault
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: <SecretProviderClassName> # Name of the SecretProviderClass
  namespace: <namespacename> # Name of the namespace for the SecretProviderClass
 spec:
  provider: azure
  parameters:
    usePodIdentity: "false"
    useVMManagedIdentity: "true" # Set to true for using managed identity
    clientID: "<user_assigned_client_id>" # Setting this to use the workload identity client id
    keyvaultName: <keyVaultName> # Set to the name of your key vault
    cloudName: "" # [OPTIONAL for Azure] if not provided, the Azure environment defaults to AzurePublicCloud
    objects:  |
      array:
        - |
```

[141] https://confluence.int.abnamro.com/x/sksQHw

```
          objectName: <secretName> # name of the secret
          objectType: secret
          objectVersion: "" # (optional) version of the secret         - |
          objectName: <secretName> # name of the secret
          objectType: key
          objectVersion: <objectVersion> # (optional) version of the secret
    tenantId: <tenantId> # The tenant ID of the key vault
```

15.12.2.3 Volume mount

Piece of the pod spec to mount the secret store based on the SecretProviderClass. By default the SecretProviderClass is to be found in the same namespace.

volumeMount

```
spec:
  containers:
    volumeMounts:
      - name: secrets-store01-inline
        mountPath: "/mnt/secrets-store"
        readOnly: true
  volumes:
    - name: secrets-store01-inline
      csi:
        driver: secrets-store.csi.k8s.io
        readOnly: true
        volumeAttributes:
            secretProviderClass: <SecretProviderClassName> # Name of the
SecretProviderClass
```

15.12.2.4 Example validation

To see if the secret and key are available which have been defined in the secretProviderClass use the following example pod manifest with the volume mount using the secretProviderClass.

mypod

```
# This is a sample pod definition for using SecretProviderClass and workload identity
to access your key vault
kind: Pod
apiVersion: v1
metadata:
  name: mypod
  namespace: <namespacename> # Name of the namespace for the SecretProviderClass
  labels:
    azure.workload.identity/use: "true"
spec:
  serviceAccountName: <workloadIdentityServiceAccountName> @ Name of the service
account created for Workload Identity
  containers:
  - name: mypod
    image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    resources:
      requests:
        cpu: 100m
        memory: 128Mi
      limits:
        cpu: 250m
        memory: 256Mi
    volumeMounts:
      - name: secrets-store01-inline
        mountPath: "/mnt/secrets-store"
        readOnly: true
  volumes:
    - name: secrets-store01-inline
      csi:
        driver: secrets-store.csi.k8s.io
        readOnly: true
        volumeAttributes:
          secretProviderClass: <SecretProviderClassName> # Name of the
SecretProviderClass
```

run the following commands;

Validation

```
## show secrets held in secrets-store
kubectl exec mypod -n <namespacename -- ls /mnt/secrets-store/

## print a test secret '<secretName>' held in secrets-store
kubectl exec mypod -n <namespacename -- cat /mnt/secrets-store/<secretName>
```

If you face issues you can follow these troubleshooting steps[142] here to retrieve the secret store provider logs.

Leveraging Managed Identity

The following steps provide an example on how to integrate a Key Vault

[142] https://learn.microsoft.com/en-us/troubleshoot/azure/azure-kubernetes/troubleshoot-key-vault-csi-secrets-store-csi-driver

15.12.2.5 Prerequisites

To be able to mount the secrets store volume on a pod using a private link endpoint connected Key Vault integration we need the following list of prerequisites.

- NSG outbound rule port 443 with source AKS Subnet and destination PLE Subnet.
- NSG inbound rule port 443 with source AKS Subnet to destination PLE Subnet
- When using Access Policies for Azure Key Vault provide 'get' permissions on secrets or 'Key Vault Secrets User' when using RBAC for ***one*** of the following Identity options.

  > *Make sure you add an Access Policy or RBAC role assignment for the identity option of your choice to your IaC deployment.*

  a. ***(Preferred)*** The 'CSI Secret Store addon user-assigned managed identity' - 'azurekeyvaultsecretsprovider-*clustername*'. Which is created during the add-on deployment. This auto created UMI is being added as an identity to the AKS VMSS nodepools by default.

     - If you want to leverage the 'CSI Secret Store addon user-assigned managed identity' you can get the client id by using the following command;

       query azureKeyvaultSecretsProvider

       ```
       az aks show -g <resource-group> -n <cluster-name> --query
       addonProfiles.azureKeyvaultSecretsProvider.identity.clientId -o
       tsv
       ```

  b. Using your own user assigned managed identity provided during cluster deployment.

     - The properties of this UMI can be retrieved using the following command;

       query azureKeyvaultSecretsProvider

       ```
       az aks show -g <resource-group> -n <cluster-name> --query "[id,
       identity]"
       ```

     - This identity needs to be assigned to the AKS VMSS in the Infrastructure resource group; Provide an access identity to the Azure Key Vault Provider for Secrets Store CSI Driver for Azure Kubernetes Service (AKS) secrets - Azure Kubernetes Service | Microsoft Learn[143]

[143] https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-identity-access#access-with-a-user-assigned-managed-identity

query azureKeyvaultSecretsProvider

```
az vmss identity assign -g <MC_resource-group> -n <agent-pool-
vmss> --identities <identity-resource-id
```

c. Using your own user assigned managed identity.

- This identity needs to be assigned to the AKS VMSS in the Infrastructure resource group; Provide an access identity to the Azure Key Vault Provider for Secrets Store CSI Driver for Azure Kubernetes Service (AKS) secrets - Azure Kubernetes Service | Microsoft Learn[144]

query azureKeyvaultSecretsProvider

```
az identity create -g <resource-group> -n <identity-name>
az vmss identity assign -g <MC_resource-group> -n <agent-pool-
vmss> --identities <identity-resource-id>
```

d. Using a System Assigned Managed Identity.

- Several System Assigned Managed Identities are created with your AKS cluster which reside in the Nodepool RG. You can for example leverage the Kubelet Identity which is by default called like 'AKS Cluster Name-agentpool' for which the properties of this UMI can be retrieved using the following command;

query azureKeyvaultSecretsProvider

```
az aks show -g <resource-group> -n <cluster-name> --query "[id,
identityProfile]"
```

- You could also create your own kubelet identity; Use a managed identity in Azure Kubernetes Service - Azure Kubernetes Service | Microsoft Learn[145]

15.12.2.6 SecretProviderClass

First we create a 'SecretProviderClass'. The most straightforward method is using the 'azureKeyvaultSecretsProvider' add-on managed identity which is created by the addon itself during deployment of the cluster. Other methods like your own user assigned managed identity are available as well

[144] https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-identity-access#access-with-a-user-assigned-managed-identity

[145] https://learn.microsoft.com/en-us/azure/aks/use-managed-identity#create-a-cluster-using-user-assigned-kubelet-identity

which can be seen here Provide an identity to access the Azure Key Vault Provider for Secrets Store CSI Driver[146].

- Namespace
- The name of the secret object in the key vault
- The object type (secret, key, or certificate)
- The name of your Azure key vault resource
- The Azure tenant ID that the subscription belongs to

secretProviderClass

```
# This is a SecretProviderClass example using user-assigned identity to access your
key vault
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: <SecretProviderClassName> # Name of the SecretProviderClass
  namespace: keyvaultsecret
spec:
  provider: azure
  parameters:
    usePodIdentity: "false"
    useVMManagedIdentity: "true" # Set to true for using managed identity
    userAssignedIdentityID: <clientID> # Set the clientID of the user-assigned
managed identity to use
    keyvaultName: <keyVaultName> # Set to the name of your key vault
    cloudName: "" # [OPTIONAL for Azure] if not provided, the Azure environment
defaults to AzurePublicCloud
    objects:  |
      array:
        - |
          objectName: <secretName> # name of the secret
          objectType: secret
          objectVersion: ""
        - |
          objectName: <secretName> # name of the secret
          objectType: key
          objectVersion: <objectVersion> # (optional) version of the secret
    tenantId: <tenantId> # The tenant ID of the key vault
```

15.12.2.7 Volume mount

Piece of the pod spec to mount the secret store based on the SecretProviderClass. By default the SecretProviderClass is to be found in the same namespace.

[146] https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-identity-access

volumeMount

```
spec:
  containers:
    volumeMounts:
      - name: secrets-store01-inline
        mountPath: "/mnt/secrets-store"
        readOnly: true
  volumes:
    - name: secrets-store01-inline
      csi:
        driver: secrets-store.csi.k8s.io
        readOnly: true
        volumeAttributes:
            secretProviderClass: <SecretProviderClassName> # Name of the
SecretProviderClass
```

15.12.2.8 Example validation

To see if the secret and key are available which have been defined in the secretProviderClass use the following example pod manifest with the volume mount using the secretProviderClass.

mypod

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  nodeSelector:
    kubernetes.io/os: linux
  containers:
  - image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    name: mypod
    resources:
      requests:
        cpu: 100m
        memory: 128Mi
      limits:
        cpu: 250m
        memory: 256Mi
    volumeMounts:
      - name: secrets-store01-inline
        mountPath: /mnt/secrets-store
        readOnly: true
  volumes:
```

```
  - name: secrets-store01-inline
    csi:
      driver: secrets-store.csi.k8s.io
      readOnly: false
      volumeAttributes:
        secretProviderClass: <SecretProviderClassName> # Name of the
SecretProviderClass
```

run the following commands;

Validation

```
## show secrets held in secrets-store
kubectl exec mypod -n keyvaultsecret -- ls /mnt/secrets-store/

## print a test secret 'ExampleSecret' held in secrets-store
kubectl exec mypod -n keyvaultsecret -- cat /mnt/secrets-store/secret01
```

If you face issues you can follow these troubleshooting steps[147] here to retrieve the secret store provider logs.

15.12.3 resources

Provide an access identity to the Azure Key Vault Provider for Secrets Store CSI Driver for Azure Kubernetes Service (AKS) secrets - Azure Kubernetes Service | Microsoft Learn[148]
Use the Azure Key Vault Provider for Secrets Store CSI Driver for Azure Kubernetes Service secrets - Azure Kubernetes Service | Microsoft Learn[149]
Troubleshoot Azure Key Vault Provider for Secrets Store CSI Driver on Azure Kubernetes Service (AKS) - Azure | Microsoft Learn[150]

15.13 Add an Image to Nexus Private registry

> *For some general information and prerequisites regarding consuming images in AKS 3,0 and AKS check* Container Images and Nexus (see page 27) *previously in this cookbook.*

[147] https://learn.microsoft.com/en-us/troubleshoot/azure/azure-kubernetes/troubleshoot-key-vault-csi-secrets-store-csi-driver

[148] https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-identity-access#use-the-csi-secret-store-addon-user-assigned-managed-identity

[149] https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-driver

[150] https://learn.microsoft.com/en-us/troubleshoot/azure/azure-kubernetes/troubleshoot-key-vault-csi-secrets-store-csi-driver

To consume an upstream image in AKS it needs to be available through Nexus hosted repository acting as a private docker registry https://p-nexus-3.development.nl.eu.abnamro.com:**18443**[151]. This can be done with a one-liner Dockerfile and the `FROM` instruction in relation with a Pipeline template Docker[152]. An how to is available here; AKS Ingress Container Images and Nexus[153]

15.13.1 resources:

Stratus: Container image and Helm chart sources - FAQ[154]

15.14 Deploy you cluster with CNI Overlay

If you want to setup your cluster using CNI Overlay you need to follow next steps;

- Read and understand the CNI Overlay concepts[155].
- With the pods taking their IP address from the Overlay network, the nodes are the only ones consuming IP addresses from the SSNS Subnet. This calls for new math. Check how to (re-)calculate your Subnet requirements[156]
- The rules you need are included in the NSG rules of the AKS SSNS pattern[157] review your current NSG setup to verify if the new rules related to 172.16.0.0/12 exist or redeploy your subnet.
- Recreate your cluster and enable overlay out of the gate. Enabling overlay on an existing cluster is not possible.

See this how to see more context on setting up CNI overlay for your setup and ARM/Terraform examples; CNI Overlay - Stratus[158]

Enable CNI Overlay Bicep

```
...
    networkProfile: {
      networkPluginMode: 'overlay' // Enabled CNI Overlay, podCidr needs to be set to '172.16.0.0/16'
      podCidr: '172.16.0.0/16' // Fixed CIDR range when networkPluginMode: 'overlay' is provided.
      serviceCidr: '10.236.0.0/16'
```

[151] https://p-nexus-3.development.nl.eu.abnamro.com:18443/

[152] https://dev.azure.com/cbsp-abnamro/GRD0001045/_wiki/wikis/PITA%20templates/11782/docker

[153] https://confluence.int.abnamro.com/x/V7HoG#FSCP2to3AKSIngress-ContainerImagesandNexus

[154] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/81831/Container-image-and-Helm-chart-sources?anchor=faq

[155] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation?wikiVersion=GBtrunk&pagePath=/Concepts/CNI%20Overlay

[156] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation?wikiVersion=GBtrunk&pagePath=/How%252DTos/Calculate%20Subnet%20Requirements&anchor=using-cni-overlay

[157] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/59117/Product-Description?anchor=network-security-group

[158] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/95397/CNI-Overlay

```
      dnsServiceIP: '10.236.0.10'

      // Locked by FSCP's Azure Policies
      outboundType: 'userDefinedRouting'
      networkPlugin: 'azure'
      networkPolicy: 'calico'
    }
...
```

Pods (but **not** other resources like DaemonSets) will be assigned IP addresses from the pod CIDR you specified during the cluster's creation:

Check assigned ip addresses

```
kubectl get pods -A -o custom-columns='NAME:.metadata.name,IP:.status.podIP'
```

15.15 Multi-Subnet AKS Cluster

15.15.1 Kudo's to Raji Rajamuthu & Nikolay Nikolov

A multi-subnet AKS cluster doesn't work out of the box. Which means that deploying multiple AKS patterned subnets is not enough to get this setup working correctly. After 2 AKS subnets have been provisioned additional NSG rules need to be configured to allow communication between those two subnets, both on the *inbound* and *outbound* side on each subnet. The assumption was made that the master/system nodepool runs in aks01 and an extra worker nodepool added in aks02.

15.15.2 Allow bi-directional k8s traffic aks01 <=> aks02

The bulk of the ports are described here: https://kubernetes.io/docs/reference/networking/ports-and-protocols/.
These ports need to be added both to 'aks01' outbound and 'aks02' inbound.

Port	Protocol	Description
443, 80, 8080	TCP	General Ports for communication
9099	TCP	Needed for Calico
5473	TCP	Needed for Felix and Calico
10250	TCP	Needed for K8 Api Server

Port	Protocol	Description
1433	TCP	For like SQL DBs, if applicable or others based on your requirements
30000-32767	TCP	NodePort Services - https://kubernetes.io/docs/concepts/services-networking/service/

15.15.3 Allow k8s unidirectional traffic aks01 => aks02 - system ports

Port	Protocol	Description
10249, 10256	TCP	kube-proxy: https://kubernetes.io/docs/reference/command-line-tools-reference/kube-proxy/
10257	TCP	kube-controller-manager: https://kubernetes.io/docs/reference/command-line-tools-reference/kube-controller-manager/
19100	TCP	node-exporter metrics endpoint: https://github.com/Azure/AKS/issues/2397
20257	TCP	Node problem detector metrics endpoint: https://learn.microsoft.com/en-us/azure/aks/node-problem-detector
6443	TCP	Kubernetes API server
2379-2380	TCP	etcd server client API
10259	TCP	kube-scheduler

15.15.4 DNS broadcast requests aks02 => aks01

Port	Protocol	Description
53	UDP	DNS

15.15.5 Debug network issue between subnets

Execute the below query in infra-p-oms to find out the denied rules if you face any issues while setting up additional subnets in AKS

Debug

```
let aks01aIP = "10.xxx.xxx";
let aks02aIP = "10.xxx.xxx";
AzureNetworkAnalytics_CL
| where SubType_s == "FlowLog"
| where (SrcIP_s startswith aks01aIP and DestIP_s startswith aks02aIP) or (SrcIP_s
startswith aks02aIP and DestIP_s startswith aks01aIP)
| project TimeGenerated, SrcIP_s, DestIP_s, DestPort_d, FlowDirection_s, NSGRules_s,
FlowStatus_s, Subnet_s
```

15.15.5.1 resources:

2 node pools 1 cluster - Stratus[159]

15.16 Backup & Restore

Although the Azure Backup Vault facilitates backing up AKS Clusters (including cluster resources and persistent volumes attached to the cluster), there are prerequisites and limitations to consider. It supports only operational backups and is compatible with persistent volumes that utilize the Container Storage Interface (CSI) driver-based Azure Disk Storage.

To activate the backup and restore functionality via the Backup Vault, the Backup Extension must be set up for the AKS Cluster. Furthermore, establishing trusted access between the AKS Cluster and the Backup Vault is necessary. The Backup Extension is available as container images in MCR.

DevOps teams are advised to use Azure File Share and Azure Blob to store application data in the AKS Cluster instead of relying on Azure Disk storage. The data stored in Azure File Share and Azure Blob should be backed up using the *Azure Recovery Services Vault or Azure Backup Vault.*

Consider the limitation in place, we do not **support Azure Backup Vault to take backup** for Azure Kubernetes Service.

The current proposed strategy involves **recreating the AKS Cluster** from the pipeline, deploying the application, and subsequently **restoring the application data** (persistent volumes) using the Azure Backup Vault.

[159] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/89919/WIP-2-node-pools-1-cluster

16 Troubleshooting

Challenge	Solution
Network Connection Troubleshooting	Basic network connection troubleshooting steps specifically for AKS can be found in the Azure DevOps Wiki[160].
Troubleshooting section Stratus	Common issues and their troubleshootin; Troubleshooting - Overview (azure.com)[161]

[160] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/55883/Network-Connection-Troubleshooting?anchor=aks-troubleshooting

[161] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/89561/Troubleshooting

Challenge	Solution
Issue: InvalidTemplateDeployment - Service clusters should be encrypted at host **Problem Statement and context:** While trying to create AKS cluster using the base template it throws this error – **ERROR: InvalidTemplateDeployment - The template deployment failed because of policy violation. Please see details for more information.** RequestDisallowedByPolicy - Resource 'ftt-baseline-template' was disallowed by policy. Policy identifiers: '[{"policyAssignment":{"name":"AAB Azure Kubernetes Service Non-Critical Pre-production v1","id":"/providers/Microsoft.Management/managementgroups/80deeff8-b249-48f2-97cf-e6c6432b4c89/providers/Microsoft.Authorization/policyAssignments/aab-aks-non-pre-v1"},"policyDefinition":{"name":"Temp disks and cache for agent node pools in Azure Kubernetes Service clusters should be encrypted at host","id":"/providers/Microsoft.Authorization/policyDefinitions/41425d9f-d1a5-499a-9932-f8ed8453932c"},"policySetDefinition":{"name":"AAB Azure Kubernetes Service Non-Critical Pre-production v1","id":"/providers/Microsoft.Management/managementGroups/80deeff8-b249-48f2-97cf-e6c6432b4c89/providers/Microsoft.Authorization/policySetDefinitions/aab-aks-noncritical-preproduction-v1"},"policyDefinitionReferenceId":"7220681900714355738"}]'. ##[error]Script failed with exit code: 1	**enableEncryptionAtHost: true** needs to be enabled in the agent profile configuration.

Challenge	Solution
Issue: InvalidTemplateDeployment - Missing input parameters **Problem Statement and context:** While using the base template certain parameters were missing hence got an error – /usr/bin/bash /home/vsts/work/_temp/azureclitaskscript1654236983731.sh **ERROR: Missing input parameters: clusterName, dnsPrefix, keyVaultName, logAnalyticsWorkspaceId, nodePoolVnetSubnetID, userAssignedManagedIdentityId** ##[error]Script failed with exit code: 1	Provide these mandatory inputs. See the reference pipeline section[162] for an example and a quickstarter pipeline/
Issue: InvalidTemplateDeployment - Pod subnet must be different than agentpools vnet subnet **Problem Statement and context:** While deploying after giving all the mandatory inputs got an error like below – ERROR: {"status":"Failed","error":{"code":"DeploymentFailed","message":"At least one resource deployment operation failed. Please list deployment operations for details. Please see https://aka.ms/DeployOperations for usage details.","details":[{"code":"BadRequest","message":"{\r\n \"code\": \"InvalidParameter\",\r\n \"message\": \"Pod subet must be different than agentpools vnet subnet\",\r\n \"subcode\": \"\"\r\n}"}]}} ##[error]Script failed with exit code: 1	Need to comment the podSubnetID: systemNodePoolPodSubnetID as its not required for base cluster deployment. In order to tackle IP exhaustion issues and realizing Dynamic IP allocation from a different subnet using calico network policy ,these details will be required.

[162] http://confluence.int.abnamro.com#FSCP2to3KubernetesService(AKS)-ReferencePipelineLink

<table>
<tr><th>Challenge</th><th>Solution</th></tr>
<tr>
<td>

Issue: InvalidTemplateDeployment - **QuotaExceeded**

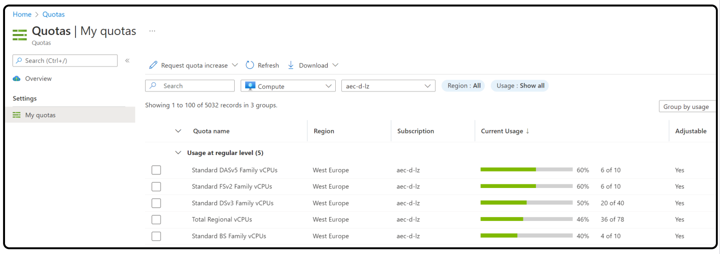
Problem Statement and context:

ERROR: {"code": "InvalidTemplateDeployment", "message": "The template deployment 'aks' is not valid according to the validation procedure. The tracking id is ''. See inner errors for details."} Inner Errors: {"code": "***QuotaExceeded***", "message": "Provisioning of resource(s) for container service ***** in resource group **** failed. Message: **Operation could not be completed as it results in exceeding approved Total Regional Cores quota.** Additional details - *Current Limit: 10, Current Usage: 0, Additional Required: 12, (Minimum) New Limit Required: 12.*

</td>
<td>

Verify if you can align the number of cores used in your solution with the default 'regional cores quota' of 10. From a cost perspective the number of cores/vCPU's should be set to a functional minimum in line with your requirements. There may be a differentiation in required VM sizes per environment for your solution based on SLA and resource demand. To prevend exceeding the cores quota you can change the VM size (nodePoolVmSize), or the number of VM's (nodePoolCount).

The Quotas are defined on the subscription level and can be updated there as well when needed.



Home > Quotas
Quotas | My quotas
Quotas
Search (Ctrl+/)
Overview
Settings
My quotas
Request quota increase
Refresh
Download
Search
Compute
aec-d-lz
Region : All
Usage : Show all
Group by usage

Showing 1 to 100 of 5032 records in 3 groups.

<table>
<tr><th>Quota name</th><th>Region</th><th>Subscription</th><th>Current Usage ↓</th><th></th><th>Adjustable</th></tr>
<tr><td>Usage at regular level (5)</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>□ Standard DASv5 Family vCPUs</td><td>West Europe</td><td>aec-d-lz</td><td>60%</td><td>6 of 10</td><td>Yes</td></tr>
<tr><td>□ Standard FSv2 Family vCPUs</td><td>West Europe</td><td>aec-d-lz</td><td>60%</td><td>6 of 10</td><td>Yes</td></tr>
<tr><td>□ Standard DSv3 Family vCPUs</td><td>West Europe</td><td>aec-d-lz</td><td>50%</td><td>20 of 40</td><td>Yes</td></tr>
<tr><td>□ Total Regional vCPUs</td><td>West Europe</td><td>aec-d-lz</td><td>46%</td><td>36 of 78</td><td>Yes</td></tr>
<tr><td>□ Standard BS Family vCPUs</td><td>West Europe</td><td>aec-d-lz</td><td>40%</td><td>4 of 10</td><td>Yes</td></tr>
</table>

At this stage this is a manual step. Go to <subscription> / Usage + quotas / select <regional cores vCPU's' and on the right select 'request increase'. Set it to the required minimum.
How to: Manage Azure Subscription Compute (CPU) and other quotas - Overview[163]
Increase VM-family vCPU quotas - Azure Quotas | Microsoft Learn[164]

</td>
</tr>
</table>

[163] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/74301/How-to-Manage-Azure-Subscription-Compute-(CPU)-and-other-quotas

[164] https://learn.microsoft.com/en-us/azure/quotas/per-vm-quota-requests

Challenge	Solution
Issue: InvalidTemplateDeployment - nodeLabel is not set or do not have the proper format **Problem Statement and context:** While trying to create AKS cluster using the base template it throws this error: ##[error]The template deployment failed because of policy violation. Please see details for more information. ##[error]Details: ##[error]Resource 'aks01' was disallowed by policy. Error Type: PolicyViolation, Policy Definition Name : AAB Azure Kubernetes Service - Application Name node label DENY v1, Policy Assignment Name : aab-aks-cri-pre-v1. Error Type: PolicyViolation, Policy Definition Name : AAB Platform - Managed Resource Group Tag DENY v1, Policy Assignment Name : aab-pla-aks-m-rg-role-v1.	Azure policy "**AAB Azure Kubernetes Service - Application Name node label DENY v1**" checks for 3 to 5 characters-long value as that's the definition of an application ID in AAB. That label should receive the registered App ID, not a vanity value. And no additional labels are supported. App ID Registration - Software Development - Confluence (abnamro.com)[165]
Issue: Connect to the cluster using '-admin' is restricted. **Problem Statement and context:** While trying to connect the cluster from az cli to get the aks credentials with –admin is restricted which was possible in FSCP 2.0.	Local accounts are no longer allowed. Make sure az aks get-credentials is fired without the --admin tag. see Connect to the AKS control plane

[165] https://confluence.int.abnamro.com/display/GRIDAD/App+ID+Registration

Challenge	Solution
Issue: **Connect to the cluster keep asking for MFA code** **Problem Statement and context:** While connecting to the AKS cluster from devops agent it keeps asking for MFA Code if not provided it errors out.	Use a Kubernetes Service Connector or Install Kubelogin on the agents. see Connect to the AKS control plane and Create a DevOps Kubernetes Service Connection (see page 54)
Issue: **SSNS error when deploying AKS pattern** **Problem Statement and context:** While deploying a subnet for AKS with the SSNS pipeline task an error message can pop-up: ##[error]Cannot bind argument to parameter 'GroupName' because it is an empty string.	The following pipeline variables should be defined in the pipeline or pipeline variables group: • GroupId • GroupName
Issue: Pre-allocated IPs ** exceeds IPs available ** in Subnet CIDR **Problem Statement and context:** The deployment of the AKS Cluster resource fails with the following error: "Pre-allocated IPs 62 exceeds IPs available 59 in Subnet CIDR 10.146.15.0/26. http://aka.ms/aks/insufficientsubnetsize"	Make sure the AKS subnet size is large enough to accommodate the required number of ip addresses. More info can be found here: https://docs.microsoft.com/en-us/azure/aks/configure-azure-cni#plan-ip-addressing-for-your-cluster

Challenge	Solution
Issue: Application deployment - Dial tcp 10.*.*.*:443: i/o timeout **Problem Statement and context:** Dial tcp 10.145.44.4:443: i/o timeout during an application deployment using a Kubernetes Service Connection.	Make sure the Azure Devops agent pool has inbound access to the AKS Subnet NSG. You will need to use a self-hosted private pool and the respective subnet needs to be allowed on the NSG. See this link Usage Guidance - Overview (azure.com)[166] for an example. And this link to check the needed subnets Internet Access from Private Agents[167]
Issue: Kubectl - create namespace forbidden **Problem Statement and context:** $ kubectl create namespace ***** Error from server (Forbidden): namespaces is forbidden: User "****@nl.abnamro.com[168]" cannot create resource "namespaces" in API group "" at the cluster scope: User does not have access to the resource in Azure. Update role assignment to allow access.	You need a service account plus role created in the cluster as described in the cookbook. That service account should be given the permission to be able to create namespaces. To be able to create the service account / role and rolebinding you need RBAC permissions on the Azure Resource level. This is done by assigning a user/group the role permission 'Azure Kubernetes Service RBAC Admin'. That should provide you permissions to create the service account / role / role binding. You should create namespace through the pipeline as a best practice. You could assign a user/group the 'Azure Kubernetes Service RBAC **Cluster** Admin' rights in lower environments to be able to create namespace from the portal. See; Create a DevOps Kubernetes Service Connection (see page 54)
Issue: Kubectl - Dial tcp 10.*.*.*:443: i/o timeout **Problem Statement and context:** While retrieving logs from a AKS resource using kubectl like 'kubectl logs pod mypod -n namespace' the following error message is shown; 'dial tcp 10.145.176.132:443: i/o timeout'	De default network policies disallows traffic towards a pod or Kuberenetes apiserver. Create a calico networkpolicy to allow the traffic towards the ip or service. Scoping the traffic to a namespace is straightforward and easy to maintain and therefore recommended.

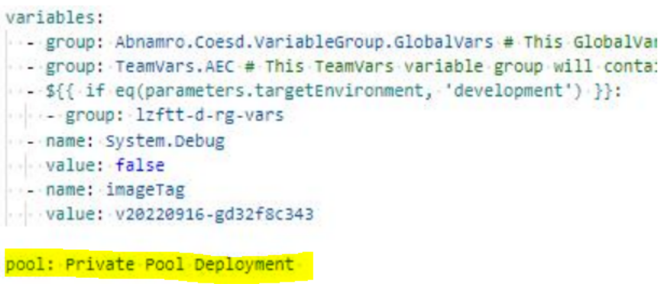
[166] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/59377/Usage-Guidance?anchor=nsg-inbound-to-ple-subnet-from-private-host-agent-(example-given-on-443)

[167] https://confluence.int.abnamro.com/pages/viewpage.action?spaceKey=GRIDAD&title=Internet+Access+from+Private+Agents

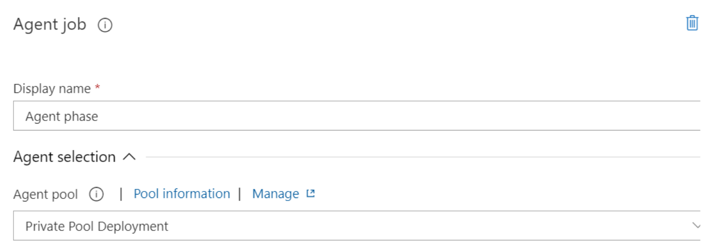
[168] http://nl.abnamro.com

Challenge	Solution
Issue: Kubectl - The azure auth plugin is deprecated **Problem Statement and context:** The azure auth plugin is deprecated in v1.22+, unavailable in v1.25+; use https://github.com/Azure/kubelogin instead	Kubelogin is needed to convert and store AKS credentials. Follow the steps outlined here; InstallKubectl (see page 44)
Issue: Kubectl - Please enter Username / Password **Problem Statement and context:** When you use kubectl to connect to your cluster your are prompted to provide your username and password	Kubectl.exe is outdated and needs to be updated. This can be done by running 'az aks cli-install' which will download the latest versions of kubectl to 'C:\Users\userid\\.azure-kubectl\kubectl' This location should be in your path variable as mentioned earlier on this page. If the issue persists, you have kubectl.exe instances in other folders which take precedence in the path variable over the one installed in your profile folder location. This can be solved by removing these instances or change the order in the path variable.

Challenge	Solution

Issue: Kubectl - namespace creation is forbidden
Problem Statement and context:
Team is trying to deploy AKS in FSCP 3.0. AKS is deployed , but now while deploying namespace pipeline is breaking and getting below Error;
namespaces is forbidden: User "XYZ@nl.abnamro.com[169]" cannot create resource "namespaces" in API group "" at the cluster scope: User does not have access to the resource in Azure. Update role assignment to allow access.
Namespaces are a way to organize clusters into virtual sub-clusters — they can be helpful when different teams or projects share a Kubernetes cluster. Any number of namespaces are supported within a cluster, each logically separated from others but with the ability to communicate with each other.
Kubernetes resources, such as pods and deployments, are logically grouped into a namespace to divide an AKS cluster and restrict create, view, or manage access to resources. For example, you can create namespaces to separate business groups.
Note: while deploying namespace using kubectl, if you are getting similar error of forbidden. Please refer below challenge in the current Troubleshoot section above.
	This Error is thrown because the pipeline task that was running on VM(having Azure Pipeline Agent) to create namespace didn't have Access to private AKS Cluster.
	An agent pool is a collection of agents. Instead of managing each agent[170] individually, you organize agents into agent pools. It can be private pool or Microsoft hosted Pool. It contains properties like VM Image which indicates the operating System of the Pool Agent on which pipeline jobs need to run. When you configure an agent, it is registered with a single pool, and when you create a pipeline, you specify the pool in which the pipeline runs. When you run the pipeline, it runs on an agent from that pool that meets the demands[171] of the pipeline.
	Solution: use correct Agent Pool name: "**Private Pool Deployment**" as shown below, So that respective Job runs on agent(VM) having all necessary Configurations to run that job. Since our AKS Cluster is private we must use private pool Deployment agent .
	• **Using YAML Pipeline (Recommended Approach in FSCP 3.0)**
	
	`variables:`
	`  - group: Abnamro.Coesd.VariableGroup.GlobalVars # This GlobalVa`
	`  - group: TeamVars.AEC # This TeamVars variable group will conta`
	`  - ${{ if eq(parameters.targetEnvironment, 'development') }}:`
	`    - group: lzftt-d-rg-vars`
	`  - name: System.Debug`
	`    value: false`
	`  - name: imageTag`
	`    value: v20220916-gd32f8c343`
	
	`pool: Private Pool Deployment`
	• **Using Classic Pipeline**
	********Using Classic Pipeline is Not a recommended in FSCP 3.0********

[169] http://nl.abnamro.com
[170] https://learn.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=azure-devops
[171] https://learn.microsoft.com/en-us/azure/devops/pipelines/process/demands?view=azure-devops

Challenge	Solution
	For Classic Pipeline change Agent pool as shown below .  Agent job Display name * Agent phase Agent selection Agent pool | Pool information | Manage Private Pool Deployment
Issue: Docker - Pull image fails with 'EOF' **Problem Statement and context:** While trying to pull an upstream image using a docker file using the docker flow pita template using a FROM statement like 'FROM registry.k8s.io/ingress-nginx/controller:${TAG[172]}' is failing with 'Get "https://registry.k8s.io/v2/"[173]: EOF'	Pulling upstream images using a docker file directly is not allowed. Upstream Images need to be consumed from the Nexus proxy repository. That way they will be scanned for vulnerabilities. See also Container Images and Nexus (see page 27)
Issue: Nexus - failed to resolve reference **Problem Statement and context:** failed to resolve reference "p-nexus-3.development.nl.eu.abnamro.com[174]:18443/****": failed to do request: Head "https://p-nexus-3.development.nl.eu.abnamro.com:18443/v2/****":[175] dial tcp 10.240.53.177:18443: i/o timeout	Add an outbound NSG rule on the AKS subnet for ports 8443,18443,18445 to 10.240.5.119,10.240.53.177

[172] http://registry.k8s.io/ingress-nginx/controller:${TAG

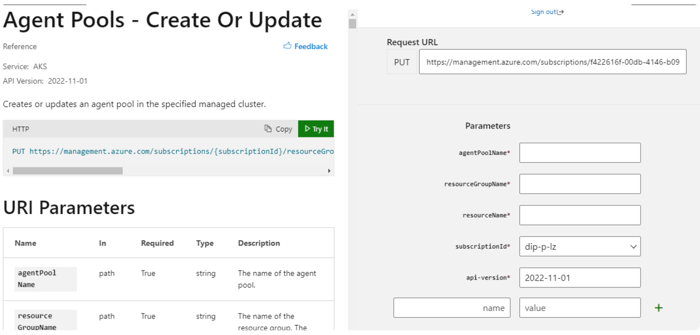
[173] https://registry.k8s.io/v2/%22

[174] http://p-nexus-3.development.nl.eu.abnamro.com

[175] https://p-nexus-3.development.nl.eu.abnamro.com:18443/v2/mcpk/ingress-nginx/kube-webhook-certgen/manifests/sha256:78351fc9d9b5f835e0809921c029208faeb7fbb6dc2d3b0d1db0a6584195cfed%22:

Challenge	Solution
Issue: Nexus - failed to pull and unpack image **Problem Statement and context:** After deploying a pod to AKS its status is stuck on ContainerCreation or Failed. The following example message is shown in the portal for the pod events or trough describing the pod using kubectl. message: >- Failed to pull image "p-nexus-3.development.nl.eu.abnamro.com[176]:18443/*****": rpc error: code = NotFound desc = failed to pull and unpack image "p-nexus-3.development.nl.eu.abnamro.com[177]:18443/***** failed to resolve reference "p-nexus-3.development.nl.eu.abnamro.com[178]:18443/***** not found	The image does not yet exist in the DOCKER_HOSTED_REPO. Images can not be retrieved using the DOCKER_GROUP_REPO but need to be consumed from DOCKER_HOSTED_REPO by building the image locally using a docker pipeline template and push them to the private Nexus DOCKER_HOSTED_REPO as shown in the how to section Add an Image to Nexus Private registry (see page 71)
Issue: Nexus - Container image has not been allowed **Problem Statement and context:** After deploying a pod to AKS its status is stuck on ContainerCreation or Failed. The following example message is shown in the portal for the pod events or trough describing the pod using kubectl. message: >- Error creating: admission webhook "validation.gatekeeper.sh" denied the request: [azurepolicy-k8sazurev2containerallowedimag-5ede031890fdf5eb6394]	The image might be available in a proxy repository but images can not be retrieved using the DOCKER_GROUP_REPO which is blocked by policy. Images need to be consumed from DOCKER_HOSTED_REPO by building the image locally using a docker pipeline template and push them to the private Nexus DOCKER_HOSTED_REPO as shown in the how to section Add an Image to Nexus Private registry (see page 71)

[176] http://p-nexus-3.development.nl.eu.abnamro.com
[177] http://p-nexus-3.development.nl.eu.abnamro.com
[178] http://p-nexus-3.development.nl.eu.abnamro.com

Challenge	Solution
Container image p-nexus-3.development.nl.eu.abnamro.com[179]:18445/*****1 for container controller has not been allowed.	
Issue: Calico - issue regarding network policies **Problem Statement and context:** You face issue regarding calico network policies	Review the dedicated sub page; FSCP AKS Calico Network Policies Troubleshooting[180]
Issue: Deployment - Provisioning of resource(s) for container service failed **Problem Statement and context:** Deploying an AKS cluster referenced in the FTT Quickstart Pipeline provisions a single system nodepool, If the customer updates the bicep template to add another user nodepool. The following error is encounterd. This seems to be an issue with MS which is being worked upon. *##[error]BadRequest: Provisioning of resource(s) for container service ***** in resource group ****-rg failed. Message: A new agent pool was introduced. Adding agent pools to an existing cluster is not allowed through managed cluster operations. For agent pool specific change, please use per agent pool operations:* https://aka.ms/agent-pool-rest-api.	Either delete the existing cluster and add a usernodepool block to the existing template and then rerun the pipeline or if customer wants to avoid deleting the existing cluster, this can be achieved using the Agent Pool REST API mentioned below- https://learn.microsoft.com/en-us/rest/api/aks/agent-pools/create-or-update?tabs=HTTP#code-try-0 This API can also be embedded to the pipeline using Powershell for automation purposes. PFB the snapshot -  Agent Pools - Create Or Update Reference Service: AKS API Version: 2022-11-01 Creates or updates an agent pool in the specified managed cluster. HTTP `PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGro` URI Parameters Name / In / Required / Type / Description agentPoolName / path / True / string / The name of the agent pool. resourceGroupName / path / True / string / The name of the resource group. The Request URL PUT https://management.azure.com/subscriptions/f422616f-00db-4146-b09 Parameters agentPoolName*: resourceGroupName*: resourceName*: subscriptionId*: dip-p-lz api-version*: 2022-11-01 Feedback Copy Try It Sign out name value +

[179] http://p-nexus-3.development.nl.eu.abnamro.com
[180] https://confluence.int.abnamro.com/x/0PkXHQ#FSCP2to3AKSCalicoNetworkPolicies-Troubleshooting

Challenge	Solution
Issue: Deployment - flux[181] is not working correctly **Problem Statement and context:** The flux extension is not successfully applied to your cluster or twistlock defender is not applied or configuration settings coming trough flux are not applied correctly.	Flux is being used to apply default configuration settings to your cluster like twistlock-defender, calico network policy. The flux extension is taking care of this by applying configuration files using kustomize from a git repo. Sometimes the flux extension might need to be reinstalled. 1. Uninstall the current flux extension in your AKS cluster (cluster | Extensions + applications) 2. Run a policy state trigger 'az policy state trigger-scan --resource-group >resource group>' or wait on the next cycle 3. The cluster should provide an incompliance on the extension not being present in under 30 minutes. 4. Run a remediation task for the 'AAB Azure Kubernetes Service - GitOps Flux Extension DINE v1' policy scoped to the cluster. 5. Wait for the extension to become 'stable' and everything should be in place at that stage.
Issue: Redeployment of the AKS cluster fails because of a policy violation **Problem Statement and context:** Redeployment of the AKS cluster is failing because of a policy violation that was not there before.	This could be an indication that since you have last deployed the AKS Cluster there has been a policy change, i.e. a new policy has been introduced. To resolve this: 1. Examine and understand the exact nature of the policy violation. 2. Check the definition of the applicable risk controls[182] for AKS and also check the Index of policy-driven values for parameters[183]. 3. Adapt your deployment template to make it compliant to all of the latest policies. 4. Redeploy the AKS cluster succesfully. If after this redeployment still fail please raise a ticket[184] in MyIT for the Azure Compliance team.

[181] https://portal.azure.com/#blade/Microsoft_Azure_ContainerService/ExtensionPropertiesConfiguration.ReactView/extensionName/fluxextension/clusterId/%2Fsubscriptions%2Feadb9dfd-2100-4b4d-981a-3225eb77daa7%2FresourceGroups%2Flzftt-d-rg%2Fproviders%2FMicrosoft.ContainerService%2FmanagedClusters%2Fsfe-aks01

[182] https://dev.azure.com/cbsp-abnamro/Azure/_wiki/wikis/Azure.wiki/67884/AAB-Azure-Kubernetes-Service-v1

[183] https://dev.azure.com/cbsp-abnamro/GRD0001007/_wiki/wikis/AKS%20Documentation/83075/Index-of-policy-driven-values-for-parameters

[184] https://servicenow.abnamro.org/esc?id=myit_ticket&table=u_it_product_knowledge&sysparm_business_service=e3929b331b58c5104903a9b1604bcbe4&sysparm_short_description=9707807e872e61143820cb76cebb35ce

17 Reference link

Introduction to Azure Kubernetes Service - Azure Kubernetes Service | Microsoft Docs[185]

[185] https://docs.microsoft.com/en-us/azure/aks/intro-kubernetes

18 Related articles

Certificate Management Cookbook[186]
Ingress options for AKS Cookbook[187]
FSCP 3.0 AKS FAQ[188]
Best practices for managing identity - Azure Kubernetes Service | Microsoft Learn[189]

[186] https://confluence.int.abnamro.com/x/sf3oG
[187] https://confluence.int.abnamro.com/x/V7HoG
[188] https://confluence.int.abnamro.com/x/blYWG
[189] https://learn.microsoft.com/en-us/azure/aks/operator-best-practices-identity