

Where to host Docker Containers on Azure (AKS, ASE, or ASF)?

buchatech.com

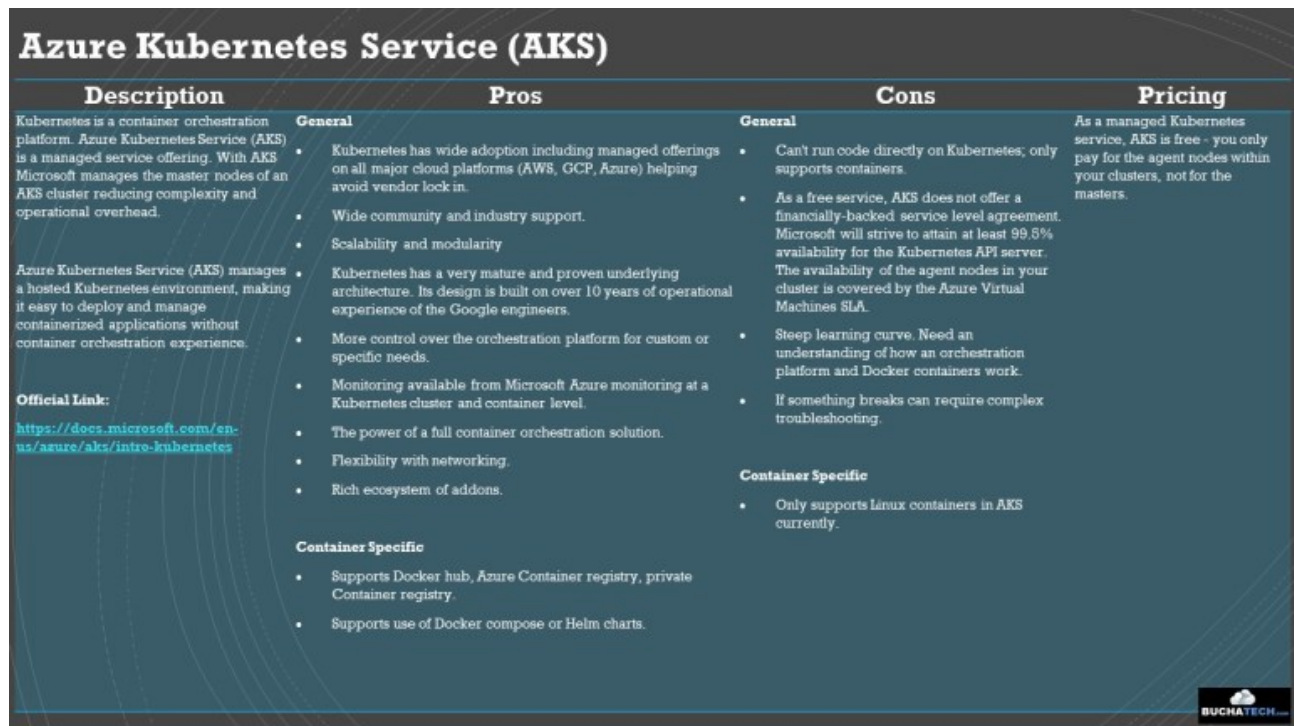
sbuchanan

January 27, 2019

Azure Kubernetes Service (AKS) service Azure App Service Environment (ASE) Azure Service Fabric (ASF) Comparison

Scenario:

So, your team recently has been tasked with developing a new application and running it. The team made the decision to take a microservices based approach to the application. Your team also has decided to utilize Docker containers and Azure as a cloud platform. Great, now it's time to move forward right? Not so fast. There is no question that Docker containers will be used, but what is in question is where you will run the containers. In Azure containers can run on Azure's managed **Kubernetes** (AKS) service, an App Service Plan on **Azure App Service Environment** (ASE), or Azure **Service Fabric** (ASF). Let's look at each one of these Azure services including an overview, pro's, cons, and pricing.



Description	Pros	Cons	Pricing
<p>Kubernetes is a container orchestration platform. Azure Kubernetes Service (AKS) is a managed service offering. With AKS Microsoft manages the master nodes of an AKS cluster reducing complexity and operational overhead.</p> <p>Azure Kubernetes Service (AKS) manages a hosted Kubernetes environment, making it easy to deploy and manage containerized applications without container orchestration experience.</p> <p>Official Link: https://docs.microsoft.com/en-us/azure/aks/intro-kubernetes</p>	<p>General</p> <ul style="list-style-type: none">• Kubernetes has wide adoption including managed offerings on all major cloud platforms (AWS, GCP, Azure) helping avoid vendor lock in.• Wide community and industry support.• Scalability and modularity• Kubernetes has a very mature and proven underlying architecture. Its design is built on over 10 years of operational experience of the Google engineers.• More control over the orchestration platform for custom or specific needs.• Monitoring available from Microsoft Azure monitoring at a Kubernetes cluster and container level.• The power of a full container orchestration solution.• Flexibility with networking.• Rich ecosystem of addons. <p>Container Specific</p> <ul style="list-style-type: none">• Supports Docker hub, Azure Container registry, private Container registry.• Supports use of Docker compose or Helm charts.	<p>General</p> <ul style="list-style-type: none">• Can't run code directly on Kubernetes; only supports containers.• As a free service, AKS does not offer a financially-backed service level agreement. Microsoft will strive to attain at least 99.5% availability for the Kubernetes API server. The availability of the agent nodes in your cluster is covered by the Azure Virtual Machines SLA.• Steep learning curve. Need an understanding of how an orchestration platform and Docker containers work.• If something breaks can require complex troubleshooting. <p>Container Specific</p> <ul style="list-style-type: none">• Only supports Linux containers in AKS currently.	<p>As a managed Kubernetes service, AKS is free - you only pay for the agent nodes within your clusters, not for the masters.</p>

This Azure Kubernetes Service (AKS) image is clickable.

Azure App Service Environment (ASE)

Description	Pros	Cons	Pricing
<p>Azure App Service Environment (ASE) is an isolated and dedicated environment for running App Service apps. ASE is a Premium service plan option of the Azure App Service. ASE runs in your own Azure virtual network (VNet), providing network isolation and improved scaling capabilities. Azure app services include:</p> <ul style="list-style-type: none"> Windows web apps Linux web apps Docker containers Mobile apps Functions <p>Official Link: https://docs.microsoft.com/en-us/azure/app-service/environment/intro</p> <p>NOTE: I specifically chose to evaluate App Service Plan on Azure App Service Environment due to its isolated and dedicated nature bringing it to a more level playing field with AKS and ASF.</p>	<p>General</p> <ul style="list-style-type: none"> Isolation and secure network access. ideal for horizontally scaling stateless application tiers Apps running on ASEs can have their access gated by upstream devices such as App Gateway WAF An ASE is dedicated exclusively to a single subscription and can host 100 App Service Plan instances. Has dedicated front ends that are responsible for HTTP/HTTPS termination and automatic load balancing of app requests within the ASE. ASE can be either Internet-facing with a public IP address or internal-facing with only an Azure internal load balancer (ILB) address. ASE supports Network Security Groups (NSGs). Can run code or a container. Supports UDR's for forced tunneling of traffic to specific routes. Has a dedicated SLA from Microsoft. Can leverage deployment slots for blue/green or canary deployments. Easy to scale and has automatic scaling. <p>Container Specific</p> <ul style="list-style-type: none"> Supports Docker hub, Azure Container registry, private Container registry. Windows and Linux containerized web applications can be deployed into the same ASE. They share the same VNet but use separate app service plans. 	<p>General</p> <ul style="list-style-type: none"> When deploying a web app into an ASE or performing a scaling operation, the operation can take a couple of hours or more. This is not a promised SLA. No access to underlying servers for deep troubleshooting if needed. <p>Container Specific</p> <ul style="list-style-type: none"> Even though able to run containers on ASE app service plans it is still not a container orchestration platform, thus lacking functionality you get with a container orchestration platform. Currently, Linux on ASE is available in these 6 regions: West US, East US, West Europe, North Europe, Australia East, Southeast Asia. Supports Multi-Container app but is in preview. Only supports Docker Compose does not support Helm Charts for multi-container deployment. 	<p>\$0.30 per hour. 1TB disk space, Up to 100 instances.</p> <p>The App Service Environment Base Fee covers the cost of all the infrastructure required to run your single-tenant and isolated Azure App Services; including load balancing, high-availability, publishing, continuous delivery, app settings shared across all instances, deployment slots, management APIs, etc.</p>

This Azure App Service Environment (ASE) image is clickable.

Azure Service Fabric (ASF)

Description	Pros	Cons	Pricing
<p>Azure Service Fabric is a distributed platform for running applications based on microservices and containers. Service Fabric also serves as the orchestration platform for microservices and containers.</p> <p>Official Link: https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview</p> <p>NOTE: On Azure there are two Service Fabric offerings. #1 is a Service Fabric Cluster and #2 is Service Fabric Mesh. The cluster offering is a Service Fabric instance you manage. Mesh is a managed Service Fabric but is currently in preview.</p>	<p>General</p> <ul style="list-style-type: none"> Service Fabric can run in Azure but also can run on-premises or other clouds helping to avoid vendor lock-in. Service Fabric is now open source. Ability to run code or a container. Ability to have an application that combines containers and Service Fabric microservices. For example, NGINX for web front end running as a container and the rest of the application components running as services all on Service Fabric. Automatic load balancing. Automatic high availability. Automated load distribution and scheduling. Proven stability and enterprise use as it is used to power majority of Microsoft Azure cloud services. <p>Container Specific</p> <ul style="list-style-type: none"> Service Fabric supports both Linux and Windows containers. 	<p>General</p> <ul style="list-style-type: none"> The Service Fabric service itself is a free service and therefore does not carry an SLA. The Service Fabric availability is based on the SLA's of the underlying Azure services including virtual machines and storage. Semi-steep learning curve. Need an understanding of how an orchestration/micro services platform and Docker containers work. Service Fabric is now open source, but the community support and expertise are not as wide spread as a platform such as Kubernetes. If something breaks can require complex troubleshooting. <p>Container Specific</p> <ul style="list-style-type: none"> Service Fabric's focus is on programming frameworks for .NET and Java libraries. Containers are also supported but they are second-class citizens on the platform as they run as guest workloads and therefore do not benefit from the full feature sets as the programming frameworks. 	<p>Charged for the compute instances, storage, networking resources, and IP addresses used in a Service Fabric cluster on Azure. No charge for the Service Fabric itself.</p>

This Azure Service Fabric (ASF) image is clickable.

Conclusion:

Choose Azure Kubernetes Service if you need more control, want to avoid vendor lock-in (can run on Azure, AWS, GCP, on-prem), need features of a full orchestration system, flexibility of auto scale configurations, need deeper monitoring, flexibility with networking, public IP's, DNS, SSL, need a rich ecosystem of addons, will have many multi-container deployments, and plan to run a large number of containers. Also, this is a low cost.

Choose Azure App Service Environment if don't need as much control, want a dedicated SLA, don't need deep monitoring or control of the underlying server infrastructure, want to leverage features such as deployment slots, green/blue deployments, will have simple and a low number of multi-container deployments via Docker compose, and plan to run a smaller number of containers. Regarding cost, running a containerized application in an App Service Plan in ASE tends to be more expensive compared to running in AKS or Service Fabric. The higher cost of running containers on ASE is because with an App Service Plan on ASE, you are paying costs for a combination of resources and the managed service. With AKS and ASF you are only paying for the resources used.

Choose Service Fabric if you want a full microservices platform, need flexibility now or in the future to run in cloud and or on-premises, will run native code in addition to containers, want automatic load balancing, low cost.

A huge thanks to my colleague Sunny Singh ([@sunnys101](#)) for giving his input and reviewing this post. Thanks for reading and check back for more Azure and container contents soon.