Fairwinds

WHITE PAPER

How to Pick Your Kubernetes Cloud Provider

Pros & Cons of AKS, EKS, & GKE

TABLE OF CONTENTS

INTRODUCTION

In the evolving cloud-native world, Kubernetes is the gold standard for container orchestration. To cater to this market, the top three cloud providers offer managed Kubernetes services: Microsoft's Azure Kubernetes Service (AKS), Amazon Elastic Kubernetes Service (EKS), and Google Kubernetes Engine (GKE). While all provide compelling capabilities, like every technology, these robust solutions are great at some things and not as great at others. It's important to understand what those pros and cons for each of them are so you can choose the cloud provider that best meets the needs of your organization or your team. This paper shares Kubernetes expertise based on Fairwinds' experience deploying for diverse clients across different infrastructures.

Choosing AKS, just like choosing Amazon Elastic Kubernetes Service (EKS) or Google Kubernetes Engine (GKE), will make it easier for you to deploy Kubernetes. While each of these providers simplifies Kubernetes deployment, they are all different in terms of cluster configuration and how to best use the service.

AZURE KUBERNETES SERVICE

Pros of AKS

Let's start with the pros of using AKS.

1. Integration with the Microsoft Ecosystem

For companies that are already .NET or Microsoft shops, AKS makes sense because it is well supported in the Microsoft ecosystem. AKS also makes it easy to develop and deploy cloud-native apps in Azure, data centers, or at the edge with built-in code-to-cloud pipelines and guardrails. AKS works well for on-premises, edge, and multi cloud Kubernetes clusters. It also interoperates well with Azure security, identity, cost management, and migration services.
The Azure Active Directory integration makes cluster authentication easy, leveraging its advanced identity and access management (IAM) to monitor and maintain container security.

2. Quick Start for Cloud-Native Development

AKS allows organizations to spin up Kubernetes clusters easily, enabling rapid development and deployment of cloud-native applications. It offers automated management and scaling as well as support for Linux, Windows Server, and IoT resources via Azure Arc.

3. Simplified Kubernetes Management

Managing Kubernetes infrastructure can be complicated and time-consuming. AKS simplifies some Kubernetes management by including debugging, continuous integration/continuous delivery (CI/CD), logging, monitoring, and automated node maintenance.

4. Cost-Effective

AKS ensures that you pay only for compute services. Essentially, you pay for the virtual machines and the associated storage and networking resources associated with them. AKS offers a free tier for cluster management that you can use to get started or to experiment with. It's easy to upgrade to the standard tier, which is ideal for large-scale applications and mission-critical workloads. The standard tier offers high performance, scalability, and their service level agreement (SLA). AKS only charges for visible infrastructure: regular on-demand, reserved, or spot instance prices. Many services, including master node, cluster management, and others, are available free of charge.

5. Configuration and CLI Support

It's easy to configure the virtual network and subnet in AKS. In addition, it's simple to use the Azure command line interface (CLI), which is a robust cross-platform command-line tool that allows you to connect to Azure and execute administrative commands on Azure resources. You can install it locally on Windows, macOS, and Linux computers, from a browser through the Azure Cloud Shell or from inside a Docker container. AKS offers an automatic repair feature for nodes, which can be paired with auto-scaling node pools.

Cons of AKS

1. Market Share

AKS has been available since 2018. It has 20% market share, serves 1.5 million users, and is available in 145+ regions and 300+ availability zones. This is less than GKE and EKS, but certainly robust enough for most needs.

2. Infrastructure-as-Code Support

At Fairwinds, we believe strongly in using infrastructure as code, relying primarily on HashiCorp's Terraform. Azure can be challenging to use with Terraform as its application programming interface (API) endpoints seem to be much slower than equivalent Google Cloud Platform (GCP) or Amazon Web Services (AWS) APIs, throwing you out of flow on projects of all sizes. Other issues with Terraform on AKS include a more cumbersome cluster upgrade process.

3. Limited Operating Systems

You can only run on a few underlying operating systems in AKS: Ubuntu Linux, Azure Linux, and Windows Server 2022. The versions of Ubuntu, like 20.04, 22.04, and so on, are also pinned to specific versions of Kubernetes. In addition, the virtual machines do not support customization directly and it is not possible to provide a cloud init or user data script.

4. Rigid Deployment

When deploying a cluster, you have to run a default node pool. The default node pool ***always has to be there***. In addition, you can't change server types once deployed. AKS also does not give you the option to bring your own AMI or user data, relying instead on the daemon set container to schedule pods on nodes early in the cluster boot process.

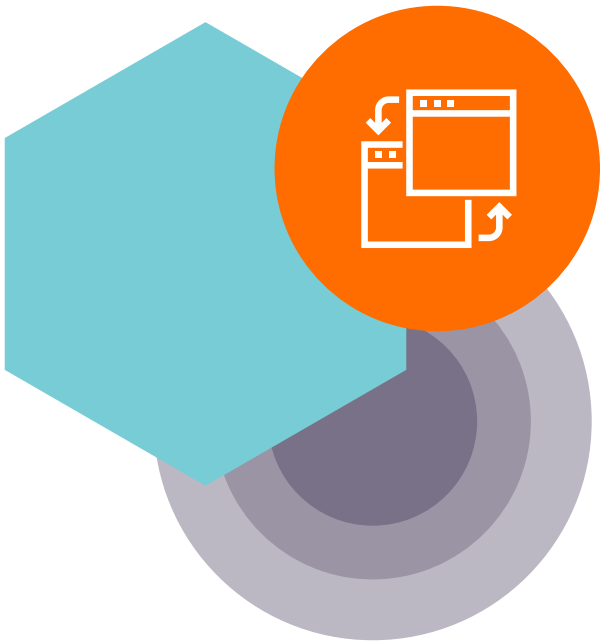
5. Uptime Requirements

AKS requires additional costs to achieve an uptime of 99.95%, while free clusters have guaranteed uptime of 99.5%. For applications and services that require more uptime, expect your spend to increase beyond purely compute costs, and don't rely on free clusters long term.

Is AKS Right for You?

AKS offers quite a few managed add-ons and extensions, such as Prometheus-based monitoring, Application Gateway Ingress Controller, keda/event-based autoscaling, policy, GitOps, and so on. Learn more about the cluster extensions available for AKS, some of which are only currently offered by AKS. These additional capabilities are worth exploring further if you're planning to use AKS.

Overall, Azure Kubernetes Service offers a robust, scalable, and integrated solution for managing containerized applications, particularly for organizations that are already working in the Microsoft ecosystem. AKS makes it easy to get started quickly, offering simplified management and cost-effective options that are appealing for many organizations.



ELASTIC KUBERNETES SERVICE

Pros of EKS

Choosing EKS helps to streamline the deployment of Kubernetes. Keep in mind, deploying Kubernetes alone isn't all you need — you'll also want to consider security, additional cloud services, additional deployment support, and bespoke node customization (among other things). Let's explore the pros of using EKS.

1. Integration with the Amazon Ecosystem

EKS was introduced in 2018 and is fully supported by and built upon Amazon's cloud computing platform, AWS. It is a mature solution that has solid integration with other industry standard tools, such as Terraform and CloudFormation, which can provide an excellent base for codifying cloud resources with Infrastructure as Code (IaC). Amazon publishes and maintains an excellent collection of Terraform modules for provisioning EKS clusters that offers a well-defined framework for integration with many other AWS services. This makes it a great choice for organizations already heavily invested in AWS services and infrastructure.

2. Strong Choice for Cloud-Native Development

EKS commands 30% of the managed Kubernetes service market share and caters to two million customers. It is available in over 200 regions across 400 availability zones, and supports a range of Kubernetes versions. EKS has first party integration with AWS for services critical to running your Kubernetes cluster, including DNS management, certificate management, network load balancing, and more. Kubernetes on AWS via EKS is well documented and there is a large community of users and useful how-to and getting started guides that make finding information easy. The AWS free tier provides a free, hands-on experience for a year to help you get started.

3. Robust Integrations

EKS integrates with many AWS services, including IAM for authentication, CloudWatch for observability and monitoring, and Elastic Load Balancing (ELB) for traffic distribution. AWS also develops Kubernetes-specific tooling, such as Karpenter, an open source cluster autoscaler that can improve application availability and cluster cost efficiency.

In addition to native AWS services, many popular open source projects such as cert-manager, Istio, and Falco have official support for EKS and include many guides on implementing and operating within an EKS environment.

3. Customizable

You can deploy your nodes using your own Amazon machine images (AMIs) or pre-built Amazon EKS AMIs. This allows you to pre-install any required packages on an Amazon Elastic Compute Cloud (Amazon EC2) instance, launch production-ready instances quickly, and implement security controls for all instances at the same time. You can search for an AMI from AWS, the user community, or through the Amazon command line interface (CLI).

Unlike many other managed Kubernetes environments, you aren't locked in to using default Amazon tooling included in new EKS clusters. CoreDNS, the cluster container network interface (CNI), and kube-proxy can be managed by EKS, extensively configured, or self-managed completely, potentially allowing you to reduce the initial overhead imposed by EKS required Daemonsets – an option that is limited, if available at all (such is the case with GKE) in other Managed Kubernetes Services.

For example, Fairwinds replaces coredns and aws-vpc-cni with our own versions, managed via IaC. This allows for tighter security configuration and the ability to decide between new versions ahead of the EKS general availability release cadence (to experiment with new features) or delaying upgrading to new versions until changes can be tested in non-production environments.

5. Configuration and CLI Support

If you prefer to interact with EKS more directly, you can use the official CLI tool (eksctl) to easily create and manage clusters on EKS without ever visiting the Amazon Console. Amazon provides many other developer tools, including various software development kits (SDKs), available in a plethora of languages.

Cons of EKS

1. Complex Setup

While EKS oversees the complex task of managing the Control Plane configuration and operation for you (something you'd have to do yourself with a self-managed or bare-metal cluster) setting up an EKS cluster can still be a daunting prospect. The vast configurability of EKS comes at the cost of having what many consider to be a complex, involved setup process. Many onerous choices surrounding adding and customizing node pools remain that may have drastic effects on the cost, reliability, and performance of your cluster.

It's important to consider your needs when selecting an EC2 Instance type as the basis for your node groups. EKS can be more complex to set up than GKE or AKS; it may be the preferred platform for more advanced users who need specific configurations or want to implement non-standard setups. These users may find EKS restrictive compared to self-managed Kubernetes solutions.

2. Multiple Node Groups

An EKS cluster contains one or more EC2 nodes that Pods are scheduled on. EKS nodes run in your AWS account and connect to the control plane of your cluster through the cluster API server endpoint. You deploy one or more nodes into a node group. A node group is one or more EC2 instances that are deployed in an EC2 Auto Scaling group. For those unfamiliar with the various AWS resources and numerous options available with respect to cluster node groups, configuration can be confusing, and adds to the overall complexity of deploying EKS. There are many ways to accept worker nodes into your cluster, including:

- **EKS Managed Node Groups:** AWS provides the instances and manages patching the OS on the nodes. A majority of the time you will use this configuration. Setup can still be quite complex, especially if you require custom AMIs with prebuilt non-standard packages, non-standard OS configurations, or configuration that must be done on startup.
- **Self-managed Nodes:** Self-managed nodes may be a good option if you need to include a bespoke node in your cluster or nodes that are configured on the fly to meet certain criteria that can't be identified beforehand (for example, an instance template that has been generated via a process outside the normal IaC). In this case, you need not only your own custom settings, but you also bear the burden of the full lifecycle of the node (including upgrades and patches that are automatically provided via Managed Node Groups).
- **Self-managed Node Groups:** You may also choose to make your configuration even more granular by introducing additional node controllers to your cluster. An example of this is AWS Karpenter, which can be deployed to try and help select the node types best suited for the current workloads on your cluster.

3. Cost

EKS delivers great services, but you do pay for compute resources, the control plane, and any other AWS services that you have integrated with your clusters. Over-provisioning node sizes (or selecting Node types that are ill-suited to your specific workloads) can exponentially increase overall cost. At Fairwinds, we developed Goldilocks, an open source utility that can help identify a starting point for setting appropriate resource requests and limits. Properly setting requests and limits helps to combat underutilization by increasing the efficiency of pod scheduling. Getting proper guidance can help you avoid forming the habit of over-provisioning.

While EKS does support multiple versions of Kubernetes, costs for EKS will increase for organizations running on versions that are no longer supported. EKS extended support currently increases the cost by 600% if you fail to upgrade the version of Kubernetes on your cluster within the support window.

4. Kubernetes Version upgrades

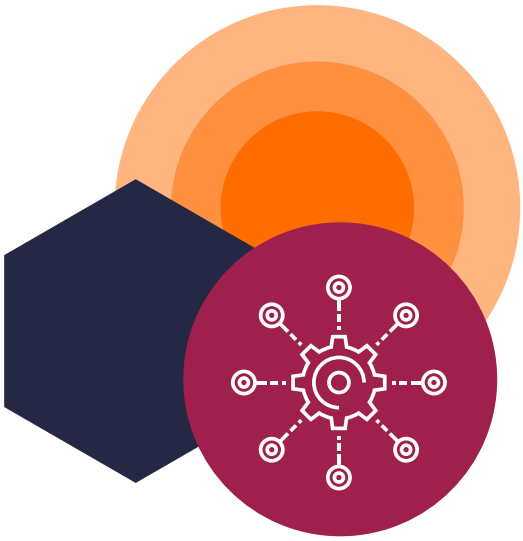
While EKS does assume responsibility for many of the duties that comes with managing a Kubernetes cluster, it **does not** proactively upgrade the version of Kubernetes your control plane is running; it is only upgraded automatically to keep pace with the oldest supported version – at additional cost, as described above. Worker nodes (whether part of Managed Node Groups or self-managed Node Groups) create Amazon EC2 instances in your account that will also need to be separately upgraded to keep pace with your control plane Kubernetes version.

5. Limited Customization

Even with the large amount of customization options that EKS provides, having the control plane managed means you are beholden to Amazon for how quickly new versions are made available. This means you may not be able to test out newer alpha Kubernetes features that are controlled by feature flags on the kubelet. In addition, you can't do much (if any) customization to the control-plane virtual machines (VMs).

Is EKS Right for You?

Overall, EKS offers a robust, scalable, and integrated solution for managing containerized applications, particularly for organizations that are already working in the Amazon ecosystem. You can quickly get started with EKS by taking advantage of its simplified management, robust documentation, and wide integration of industry standard tooling and best practices to develop a cost-effective solution.



GOOGLE KUBERNETES ENGINE

Kubernetes was released just over ten years ago by Google as an open source project to improve container orchestration. GKE was, unsurprisingly, the first to market, released in 2015.

Pros of GKE

The main benefits of GKE are that it takes responsibility for many of the underlying infrastructure items. It offers a lot of features out of the box and delivers many automated capabilities. Like the other Kubernetes managed services, GKE takes care of the heavy lifting of managing Kubernetes clusters. This enables developers to not worry about tasks outside of their main focus (building and deploying applications), such as infrastructure provisioning, configuration, and maintenance.

1. Integration with the Google Cloud Ecosystem

GKE integrates seamlessly with GCP services, including services such as Anthos, BigQuery, Pub/Sub, Google Cloud Operations, and Cloud Storage. This makes it the logical choice for organizations already invested in GCP. GKE also provides easy access to Google Cloud's networking, storage, analytics, and security services. In addition, the native integration of GKE with other Google Cloud services ensures that developers enjoy a smooth experience when building and deploying applications on GCP. GKE also offers the Cloud Code console extension for Visual Studio Code and IntelliJ for developer environments.

2. Simplified Deployment

GKE offers a lot of tools and features that enable streamlined container deployment, such as rollbacks, scaling, and automated rollouts. This makes it easy for devs to push code updates and get applications running quickly in production environments. GKE also integrates well with popular DevOps tools and CI/CD pipelines, enabling developers to deploy containerized apps easily as part of their already existing workflows. It's quite easy to deploy a Kubernetes cluster in GKE, and the command line tool and web console are both very user friendly. GKE also simplifies ongoing management with features such as:

- **Standard mode:** This mode allows you to manage the underlying infrastructure, including configuring the individual nodes.
- **Autopilot Mode:** A "hands-off" mode where GKE manages most aspects of the cluster for you.
- **Node Auto-Provisioning:** Automatically scales and provisions node pools based on your application's resource needs.

3. Straightforward Pricing

GKE offers a straightforward pay-as-you-go pricing model based on the number and types of nodes deployed. It also offers different pricing plans based on the level of support your organization needs. And while not directly related to pricing (but relevant to costs), GKE offers features for resource quotas and monitoring, which enables developers to optimize resource utilization and improve cost efficiency.

4. Security and Stability

GKE provides Container-Optimized OS, which is an operating system engine for Compute Engine VMs optimized for running containers. It's maintained by Google and helps you bring containers up on GCP quickly and securely. It comes with the Docker and containerd runtimes and cloud-init. Because Container-Optimized OS has a small footprint, it reduces the potential attack surface for your instance. It also includes a locked-down firewall and other security settings by default, a perk as Kubernetes is known not to be secure by default. It also includes Identity and Access Management (IAM) features that help GKE customers better protect sensitive workloads as well as Virtual Private Clouds (VPCs) for network isolation.

The Cons of GKE

1. Steeper Learning Curve

While GKE simplifies many tasks, it can be more complex for those new to Kubernetes or Google Cloud. Concepts like node pools, load balancers, and Ingress can require a deeper understanding compared to simpler managed services, such as AKS.

2. Costs

While straightforward (cluster management is free), GKE can be expensive, particularly if you use various Google Cloud services with your clusters. In addition, only one zonal cluster or Autopilot cluster is free. Basically, the GKE free tier provides $74.40 in monthly credits for each billing account, and these are applied to zonal and Autopilot clusters. (The fee doesn't apply to GKE Enterprise clusters.) Also, GKE meets the 99.95% SLA ***if*** you use Regional Clusters, which costs $0.10 per cluster per hour.

3. Vendor Lock-In

Similar to Amazon EKS and Microsoft AKS in their respective ecosystems, GKE is deeply integrated with Google Cloud, which can lead to vendor lock-in. This can make GKE less portable if you adopt a multi-cloud or hybrid cloud approach. In addition, if you are now supporting clients in the public sector (or you're thinking about doing so in the future), GKE does not have a government cloud, so if you need that for your Kubernetes engine, you may wish to pick another provider.

4. Cluster Type Considerations

GKE offers Zonal and Regional clusters, each with trade-offs:

- **Zonal Clusters:** Simpler and cost-effective, but less resilient to zonal failures.
- **Regional Clusters:** More fault-tolerant and meet higher SLA requirements but can be more costly.

Choosing the wrong cluster type can lead to unexpected issues later on.

Is GKE Right for You?

Overall, Google Kubernetes Engine offers a powerful, scalable, and integrated solution for managing containerized applications, particularly for organizations already working in the Google ecosystem. GKE makes it easy to get started quickly, offering cutting-edge Kubernetes features and improvements in a system that is highly reliable and well optimized, which is appealing for many organizations.

COMPARING AKS, EKS, & GKE FEATURES

FEATURE	AZURE AKS	AMAZON EKS	GKE
Cloud Integration	Deeply integrated with Azure	Deeply integrated with AWS	Deeply integrated with Google Cloud
Ease of Use	Considered easier to use than GKE	Offers a good balance	Can be complex for beginners
Advanced Features	Good for Windows-based workloads	Strong on networking and security features	Wide range of features, including Autopilot
Pricing	Competitive pricing pen_spark	Pay-as-you-go	Can get expensive with heavy usage

CHOOSING YOUR KUBERNETES CLOUD PROVIDER

Understanding the pros and cons of AKS, EKS, and GKE will enable you to make informed decisions when selecting a managed K8s provider, which will help you harness the full power of Kubernetes for your organization. Fairwinds can also accelerate your Kubernetes journey by helping you to make the right decision on AKS, EKS, or GKE and providing white-glove services so you don't need to manage your infrastructure, but can instead focus on your core business.

Learn how you can get started with AKS, EKS, or GKE fast, without requiring in-house Kubernetes expertise or specialized training with production-grade clusters, best-in-class tooling and add-ons, and a seamless way to move your workloads to production.

HOW FAIRWINDS ACCELERATES YOUR KUBERNETES JOURNEY

Fairwinds provides white-glove Managed Kubernetes-as-a-Service specifically tailored for AKS, EKS, and GKE. We help you:

Get Started Quickly: Launch production-ready clusters without deep in-house Kubernetes expertise.

Optimize for Production: Configure AKS, EKS, and GKE clusters with best practices for security, reliability, and cost efficiency.

Ongoing Support: Manage and maintain your clusters with expert support and troubleshooting.

Focus on your core business; let the management of Kubernetes infrastructure be handled by an expert team.

Fairwinds

WHY FAIRWINDS

Fairwinds provides Managed Kubernetes-as-a-Service, powered by expert people and best-in-class software to make your Kubernetes platform fast, secure, and stable. Companies of all sizes turn to Fairwinds experts to make Kubernetes an ideal platform to run services and applications. Fairwinds makes tool selection, lifecycle management for Kubernetes and its add-ons, and Kubernetes management easy.

For more information, visit
WWW.FAIRWINDS.COM

Read our blog or follow us on Twitter
@FAIRWINDSOPS