Fairwinds

WHITE PAPER

Closing the Gap of Kubernetes Services

Steps to Achieving a Production-Grade
Kubernetes Cluster in Amazon EKS, GKE or AKS

TABLE OF CONTENTS

INTRODUCTION

Amazon Elastic Kubernetes Service (Amazon EKS), Google Kubernetes Engine (GKE) and Azure Kubernetes Service (AKS) have solved how to easily set up Kubernetes Clusters. The next step is to get your clusters configured correctly and effectively leveraged with the appropriate add-ons necessary to run your application in production. This paper first discusses when to use a Kubernetes service, the strengths and weaknesses of a managed Kubernetes service and reasons why leveraging a Kubernetes service is not the same as using Kubernetes effectively.

WHEN TO USE KUBERNETES SERVICES

If software is music, the cloud is a recording studio. Individual bands spend all their time creating something new and unique, then leverage shared resources to turn that into a final, marketable product. The recording equipment, studio space—and mixing and mastering tools are absolutely essential for the band to record that music. But getting really good at designing studio space—or mixing/ mastering the band's music would be a distraction. Bands know what makes them money—and what they're good at—and they want to spend their time becoming the best musicians in the world.

With this in mind, musicians spend their lives practicing and making something new and unique. Then they leverage the same studio space as many other bands before them and trust those specialists in mixing/ mastering to do what they do best. Likewise, cloud infrastructure is not the thing that makes most companies money. Just like the band with the studio, companies need great cloud infrastructure or they can't make money.

But becoming great at building the world's most effective and reliable cloud-infrastructure is a distraction from building innovative and differentiated products and services—the things that actually make them money. Working with a Kubernetes service provides exactly the same benefit. Amazon, Google or Microsoft spend the time managing the cloud, and create the cluster you need for your application rather than attempting to build the world's best Kubernetes install.

Just like the band with the studio, companies need **great cloud infrastructure** or they can't make money.

STRENGTHS AND WEAKNESSES OF KUBERNETES SERVICES

If you think managed Kubernetes will help as you deploy an application or service, the next step is selecting which service. You'll need to look at strengths and weaknesses of all. Here is a summary to get you started.

As a disclaimer, if asked what to use, we counter by asking about the application and services it depends on. Kubernetes isn't the only thing your application needs so you'll need to factor in requirements—security, additional cloud services and node customization for example.

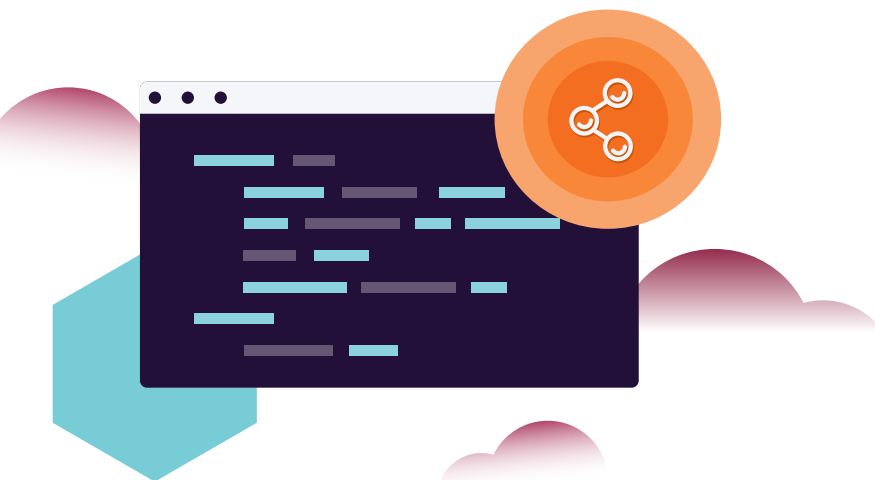### Strengths of all Kubernetes services:

Selecting AKS, EKS or GKE removes the burden of installing Kubernetes and making sure the setup works with the underlying cloud infrastructure and networking. Each managed service abstracts the control plane from view. Operators only need to worry about selecting the right server types for running their workloads.

Each managed Kubernetes service fully supports the unique piece of its underlying cloud provider. This empowers Kubernetes to create cloud resources on your behalf—such as load balancers or persistent storage. While you can run Kubernetes from scratch, it can be a challenge to run on AKS/AWS/GCP while seamlessly interacting with cloud services.

All the providers support private nodes and private APIs. From a Kubernetes security perspective, the offerings are pretty similar. However, there are times when a container workload will need permission to interact with the cloud provider API. Each provider has its own service account and IAM hierarchy. Whatever provider you select, you'll need to get up to speed with its security terminology and best practices.

Weaknesses of all Kubernetes services

All providers are dependent on upstream Kubernetes development. Kubernetes is being developed faster and faster. Each service provider must keep up with the pace of development, especially as Kubernetes is only patching the latest three cycles. If you are still on version 1.15, don't expect any patch support.

With each service you have very limited options for configuring the control plane, and it very likely cannot be enabled. This limits an operator's ability to turn on/ off Kubernetes API features. For example, if there were an alpha feature or configuration flag your version of Kubernetes supports—it cannot be enabled on a managed service provider.

Autoscaling servers into and out of the cluster can be challenging on all providers. Each service leverages a tool called the Cluster Autoscaler. While the cluster autoscaler does a great job, it does have limits on what it can do. It is important to know these limitations and be able to account for those on your selected cloud provider. For example, the cluster autoscaler isn't availability-zone aware. This can lead to node imbalance if a node pool spans multiple availability zones. Because of this limitation, you may want to run a single node pool or autoscaling group per availability zone.

Finally, while all three providers have been certified as CNCF Kubernetes conformant, none are running directly from source - they are pre-packaged and introduce additional features. This isn't really a weakness, but you may run into some inconsistencies when comparing a managed service with the Kubernetes code base.

CLOSING THE GAP TO ACHIEVE PRODUCTION-GRADE KUBERNETES

You need to select the right provider for the success of your Kubernetes cluster, but also it's your responsibility to get your clusters configured correctly and effectively leveraged. This is why using a Kubernetes service is not the same as running Kubernetes effectively. Here are four examples of why to close the gap between what a Kubernetes service provides and what you must obtain a production-grade Kubernetes deployment.

The right setup enables a new world where engineers own what they build.

Avoid unnecessary rebuilds

When deploying your first or tenth cluster on a Kubernetes service, there are decisions to make at the beginning. The wrong choice could require a time consuming cluster rebuild or have network or cost consequences. Some design considerations include:

- How many clusters should you have, in what regions and with how many AZs?
- How many separate environments, clusters and namespaces are needed?
- What workloads should you run and not run inside Kubernetes?
- How should services communicate with or discover one another?
- Should you set up security at the VPC, cluster or pod level?
- Are there any networking concerns that the default CNI won't support?
- How will you handle upgrades? Security patches? And ensure uptime throughout?

Select the right regions

Planning for HA (High Availability)

Which region and the number of availability-zones you should put your Kubernetes cluster in varies by your application. You need to consider what service tends to have better availability region by region, and location of your customers, latency needs, etc. You also need to consider your application fault tolerance to help inform your HA strategy to best meet your business needs.

Services by Region

Services available by cloud providers vary region by region. You need to avoid selecting a region that doesn't have the right container registry, cloud scheduler or secrets manager you need. By choosing correctly at the beginning, you avoid tearing down clusters and rebuilding.

Enable service ownership

When engineers have an API with Kubernetes that makes sense, and workflows for deployments that they can understand, they can own the services they build all the way through to production. The old way of shipping software required an operations team to take the service and ship it through to production, making sure it was up and running. The wrong setup with Kubernetes requires you to maintain that old paradigm, but the right setup enables a new world where engineers own what they build.

Our advice is hard-won, **helping you avoid broken functionality** and unreliable or insecure configurations.

Kubernetes and beyond

Kubernetes is your starting point. You still need tools that layer on top so you can run secure, reliable and scalable Kubernetes infrastructure.

Examples include:

- Automatic management of DNS
- Automatic TLS certificate management
- Monitoring for applications
- Shipping logs

GETTING HELP

As mentioned earlier, being an expert in Kubernetes infrastructure is not the thing that makes most companies money. Employing a team that knows all the answers to every question above is also not always possible. That's where Kubernetes experts can help.

Fairwinds is the Kubernetes enablement company. Our SRE team has already done the hard work for you. You benefit from more time working on your application or services and less time spent on Kubernetes infrastructure deployment and management.

Add experienced senior DevOps to your team

Our SREs help you avoid pain and gain confidence that you've deployed Kubernetes effectively. Your engineers new to Kubernetes get deployments done right the first time vs. facing problems requiring painful rebuilds in months or years.

Tried and tested

Our team has already tried and tested GKE, EKS and AKS. Our methodical nature and extensive experience means we have already discovered all the places to change default settings, patch and upgrade. Our advice is hard-won, helping you avoid broken functionality and unreliable or insecure configurations.

Managed add-ons

Add-ons can break your Kubernetes deployment. Fairwinds takes a standardized approach. We've built a library of approved add-ons that are solid, secure and work well with others. Did you know that there are configuration options for EKS/GKE/AKS managed services too? Fairwinds also helps manage non-default best practices for those add-ons.

A SUITE OF SERVICES

With Fairwinds, you can choose what is right for you.

Managed Kubernetes - ClusterOps is a fully-managed Kubernetes cluster management solution that integrates infrastructure as code, open source software and SRE expertise as a subscription service. The ClusterOps platform sits between the cloud infrastructure layer and application workloads so Platform Engineering teams can quickly close the gap between what you get from a cloud provider, and what you need to be production-grade.

Kubernetes Audit and Improve - A combination of services and software, Kubernetes Audit and Improve is an advisory service that helps teams avoid unnecessary trial and error, prevents technical debt and ensures teams realize the full benefits of Kubernetes infrastructure.

Fairwinds Elements - Supported suite of open source software to help manage Kubernetes infrastructure at enterprise scale.

We are your Kubernetes enablement partner adding expertise, experience and capacity to your team. Nothing proprietary, just best practices, documented as code.

Take the next step

Fairwinds can help close the gap between a Kubernetes service and getting to a production-grade deployment. If you want to talk to a Kubernetes expert, **get in touch.**

ADDITIONAL RESOURCES:

- **Kubernetes Best Practices**
- **Fairwinds ClusterOps**
- **The Fairwinds Difference**

⫶F Fairwinds

─────

WHY FAIRWINDS

Fairwinds is your trusted partner for Kubernetes security, policy and governance. With Fairwinds, customers ship cloud-native applications faster, more cost effectively, and with less risk. We provide a unified view between dev, sec, and ops, removing friction between those teams with software that simplifies complexity. Fairwinds Insights is built on Kubernetes expertise and integrates our leading open source tools to help you save time, reduce risk, and deploy with confidence.

WWW.FAIRWINDS.COM