



Are You Ready For Kubernetes?

INTRODUCTION

A current trend in software architecture is to decompose large systems into many smaller services (called microservices), each with a small and focused scope. Previously monolithic systems were hard to iterate on because a small change to one part of the code could impact any part of the system, requiring a full test and re-deployment of the entire system. This slows down the pace of innovation and feature delivery. A microservice is a smaller and self-contained set of functionality that is easier to rapidly develop, enhance, test, and deploy. A well-architected microservice can be updated, fully tested, and deployed to production in minutes. When pushing new features to production, a smaller portion of the system is changing which limits the scope of problems that could arise when rolling out changes. This architectural paradigm greatly reduces the time it takes to bring new features to market, while also reducing risk.

A common method of delivering microservices is to build them as containers that insulate them from the details of the underlying hardware and operating system. Containers make your services more portable across environments. Delivering your services as containers increases confidence that the system will work properly as it is promoted from development and test environments to production. However, the more microservices and containers you have the more complex it is to manage them. As teams migrate to a containerized, microservice architecture they invariably hit a point where managing larger numbers of containers becomes technically infeasible or too costly to do manually.

Kubernetes is the de facto standard container orchestration platform for modern software applications. Kubernetes and its ecosystem automate the critical tasks required to deploy and maintain a highly available, scalable, and secure microservice architecture. Moving to Kubernetes can allow your product, team, and business to scale to the next level.

When Should You Make The Move?

Kubernetes is both powerful and popular, but that does not mean it's the right choice for every team and every app. Some indications that it's the right time to consider moving to Kubernetes include:

- Your app is composed of multiple containers that need to interact for your app to function.
- Your application requires multiple servers to run.
- High availability is important to the continuity of business operations.
- Your application must scale up and down to handle spikes in usage.
- Your team is spending a large amount of time manually deploying software.
- Orchestrating upgrades and updates with minimal downtime is becoming complex.
- You have a Continuous Integration (CI) pipeline and are looking to enable continuous deployment (CI/CD).
- Efficient use of resources is important to keep infrastructure costs down.
- Speed and agility are critical to timing the market and outpacing competitors.
- The rapid evolution of user experience drives customer acquisition and retention.



What's The Catch?

If you've decided that moving to Kubernetes would improve speed to market, scalability, availability, or agility in the face of competition, you need to be aware of some of the complexity, and related costs, that come with managing Kubernetes. Some of these considerations include:

Technical Complexity:

Kubernetes uses containers, distributed databases, software-defined networking, software-defined storage, certificates, etc. What's more Kubernetes is highly modular with a rich ecosystem of components. It's up to the platform engineers to select which components to use and how to configure them.

Conceptual Complexity:

Kubernetes use containers, but adds on concepts like pods, deployments, replica sets, stateful sets, services, ingresses, secrets, cluster managers, etc. The team must learn and understand these concepts. This can be a steep learning curve.

Getting To Production:

These days it's fairly easy to get a small Kubernetes instance up and running. However, a proof of concept is not indicative of success in getting to production. Managing a production-ready Kubernetes instance that is secure, highly available, scalable, upgradable, and maintainable is much, much harder.

Rounding Out The Platform:

Simply deploying Kubernetes is not the end of the story. Service meshes, logging, monitoring, policy enforcement, cloud integration, ingresses, etc. are all part of a full-fledged Kubernetes environment, and each of these brings additional complexity.

If these seem daunting, don't worry; we'll cover some options that can potentially reduce the complexity of using Kubernetes in your business a little further below!

Are You Ready?

Before committing to the move, teams should ensure they are ready to make the transition. If your team doesn't have some of the prerequisites out of the way, adopting Kubernetes may be significantly more difficult, and/or the organization may not realize the full benefits Kubernetes can provide. The table below presents a self-assessment worksheet to help teams determine if they are ready.

SCORING: For each question/topic rate yourself using the following scale:

SCORE	READINESS	GUIDANCE
0	Not Ready	Your team, application, or organization does not have this capability, architecture, or skillset. You might not even be sure what this even means.
1	Somewhat Ready	Your team, application, or organization has this partial capability, architecture, or skill set. You've made some progress but aren't quite there yet.
2	Ready	Your team, application, or organization has this capability, architecture, or skill set. You feel completely comfortable.

Once you've scored each item, add the individual scores up and calculate the total.



Kubernetes Self-Assessment Worksheet

ITEM		SCORE (1-3)
Organization and Process		
1	Have you defined the specific business and/or technical problems that the transition to Kubernetes is intended to address?	
2	Do you have a budget and timeline set aside for the migration?	
3	Have you estimated the costs around continued administration and maintenance of Kubernetes?	
4	Do you have a current handle on your cloud cost and can understand the variable costs as your system scales?	
5	Do you currently have a defined SLA or availability targets for your product/service?	
Architecture and Infrastructure		
6	Is your application containerized?	
7	Do your containers follow container security best practices?	
8	Do you have a microservice design where parts of your product or service can scale independently?	
9	Do your containers expose health checks and metrics that drive failover and scaling?	
10	Do you understand the resource requirements of each container/service?	
11	Do you have a current CI/CD pipeline that automates building and testing your containers?	
Workforce		
12	Is your team comfortable with Infrastructure as Code and immutable infrastructure concepts?	
13	Does your team understand DevOps principles and the common tools and technologies that support modern DevSecOps?	
14	Is your team familiar with YAML and Kubernetes manifests?	
15	Do you have a workforce that can deploy and sustain a highly available and secure Kubernetes environment?	
Total		

SCORING RESULTS

Total Score	Guidance
0 - 20	Your organization is probably not ready to move to Kubernetes on your own. Your team should attempt to improve the areas that need the most work before considering a move (see below for some alternatives).
21 - 35	Your organization is almost ready to move. It's a good time to start a proof of concept. In parallel, consider leveling up in your weakest areas or getting some external help to improve the probability of success and maximize value you will get from the transition.
35 - 45	Your organization is pretty much ready to make the move. Addressing the few areas that could be improved would maximize the return on investment.

Alternatives

Whether your team is ready to make the move or not, the following options are some alternatives that could help decrease risk and lower the barriers to entry when adopting Kubernetes.

WEIGH YOUR OPTIONS

1

Use Your Cloud Providers Autoscaling

Some teams opt to leverage their Cloud Service Providers' (CSPs) autoscaling capabilities.

TRAINING

Lots of Training Required

PRICING

Higher Cloud Costs \$\$

DIFFICULTY


DIFFICULT EASY

2

Use A Managed Kubernetes Instance

Leverage a managed Kubernetes deployment such as Amazon EKS, Azure AKS, or Google's GKE.

TRAINING

Some Training Required

PRICING

Very Expensive \$\$\$

DIFFICULTY


DIFFICULT EASY

3

harpoon

harpoon was specifically designed to lower the barrier of entry for organizations to transition to Kubernetes

TRAINING

Minimal Training Required

PRICING

Affordable Pricing \$

DIFFICULTY


DIFFICULT EASY

1

PROS

- Removes Kubernetes.
- Don't have to containerize your application.
- A smaller step if your team is already leveraging a cloud service provider.

CONS

- Requires a large amount of knowledge.
- You lose the resource efficiency of Kubernetes.
- Heavily ties you to the specific CSP and does not support on-premise deployments.

2

PROS

- Removes the need for your team to install/maintain basic Kubernetes implementation.
- Often provides raw access to the Kubernetes API allowing for full customization.

CONS

- Requires your workforce to be intimately familiar with Kubernetes concepts.
- These Kubernetes environments may still require additional services.
- Heavily ties you to the specific CSP and does not support on-premise deployments.

3

PROS

- harpoon requires little to no training to start using.
- Teams do not need to know how to install and maintain Kubernetes.
- harpoon shields teams from the specifics of a cloud environment and also works on premise.

CONS

- Users do not have a "blank slate" the way you might with a command line prompt. However, we've gone to great lengths to identify common areas of customization and have given developers an easier way to make those changes, including custom YAML modifications for all Kubernetes resources.

Conclusion

Kubernetes provides tremendous value, but it comes with cost and complexity that make it hard to make the transition. Kubernetes can be deceptively easy to use in a proof of concept, lulling teams into a false sense of security, only to discover a steep learning curve required to get to production. Alternatives like harpoon allow users to get the benefits without the large upfront learning curve and cost of investment.

If you're looking to adopt Kubernetes without all of the complexity, try out harpoon for free or book a demo. If you need help evaluating if your team is ready to make the move and how to reduce risk, please reach out and contact us. Our team has worked with a multitude of clients in different stages of evolution to help them navigate their transition to Kubernetes. We have deep expertise in all Kubernetes-based deployment solutions and can assist in your Kubernetes transition whether or not harpoon is right for your team.

