SUMMER 2023

Steering the ship: keep on top of containerisation with Kubernetes

Adaptavist

Contents

Introduction

For organisations to survive and thrive, they have to release new features much more frequently than ever before. The shift to cloud-based infrastructure provisioning has enabled most businesses to support this speed to market. But in order to effectively build and manage applications in these environments, the industry has turned to cloud-native technologies too.

And Kubernetes is the industry standard for orchestration. In 2021, VMWare estimated that 65 percent of large companies are using it in production. While towards the end of 2022, Gartner identified 'industry cloud platforms' and 'platform engineering' as two of its top 10 strategic technology trends, predicting that 80 percent of software engineering organisations would have established platform teams by 2026.

Through container configuration, automation, and scaling, Kubernetes has revolutionised the way software is developed and deployed, particularly when it comes to scaling and managing microservice-based applications.

Microservices architecture (or microservices), is a style of development that breaks an app up into multiple independent parts (or services) to accommodate a specific application feature or task, rather than one monolithic application. While each microservice can communicate with the others, you don't have to worry about dependencies. The majority of cloud-native applications are now built using containers – one example of a microservices architecture

Apps with a microservices architecture are easier to maintain, test, and deploy than traditional architectures, and they've transformed rules around scalability, enabling organisations to build more resilient, reliable, and responsive software, much more quickly. However, it's been Kubernetes that has come to the rescue in terms of figuring out how to manage this sharp up-tick in containers.

If you’re not already using Kubernetes, there’s a good chance you will be soon. In this eBook, we’re going to provide a quick recap to put Kubernetes in context; explore the key challenges it helps your business solve, including how it can support you as you scale; look at the benefits containerisation can bring; and also ask whether it’s the right solution for your organisation.

Finally, we’ll look at what role Adaptavist can play in helping you get started with Kubernetes or improve your current approach. Now you know what it contains (pun intended), let’s get started.

Part one:

Putting Kubernetes in context

Organisations are on the hunt for new technologies to help them launch features as quickly as possible, for a number of reasons. First, their legacy applications probably incorporate old technologies or languages, and training people up on them is costly. Second, new technologies, using newer languages, are more efficient and come complete with rich features. Third, there are certain requirements they need, which can only be fulfilled by certain technologies. And fourth, but by no means least, they are moving to a microservices-based architecture.

But when organisations use different technologies and programming languages, it makes their applications complex to build, test, and run – with differences depending on a particular set of dependencies or versions, and different deployment methods. They may even need their own platform to run. Organisations don't want to waste money running each microservice on a separate compute node or instance, which is where containers come in. These applications can be run much more easily on self-contained platforms or containers.

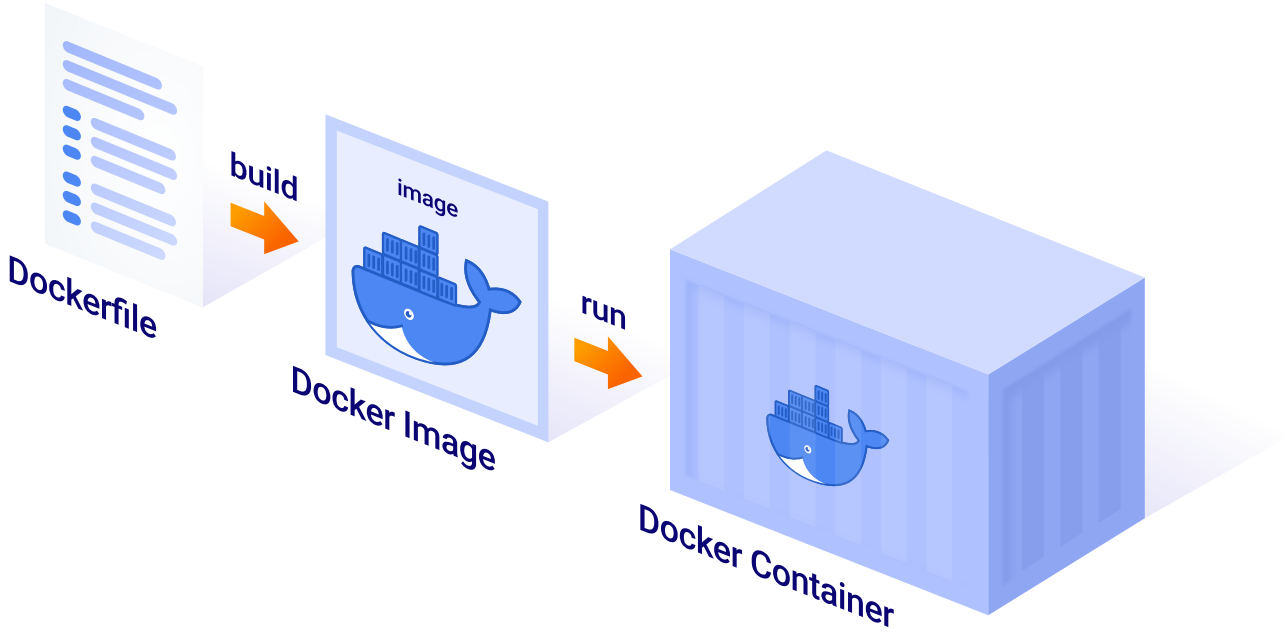
Containers share elements of the host's operating system, rather than needing their own, which makes them more lightweight and quick to spin up. Since they're not as resource intensive, they're easy to destroy or recreate. They ensure the app will run, irrespective of its underlying environment. They're also immutable at runtime. But because of that, states do not persist inside the container, and need to be sent out through API or via externally mounted volumes.

This makes apps portable, so they can be run from almost anywhere with only configuration changes, and allows the creation of environments for development, testing and production, either on-premise, or in private or public cloud.

How does Kubernetes help?

Kubernetes, which was created by Google and is now maintained by the Cloud Native Computing Foundation, was made to orchestrate containers.

The more containers you have, the more chaotic things can become. Kubernetes provides an open-source API that orchestrates containers, and controls how and where each container will run, over a fleet of worker nodes, based on what compute resources are required and what's available.



Kubernetes groups containers into pods, which usually consist of a single-purpose primary container with possible helper containers. Various controllers allow you to automatically set up multiple pods, manage them, and scale them. Kubernetes summons new pods when they're needed and kills off redundant ones. So by leveraging declarative 'manifests', the software becomes self-healing. If there's a problem and something within a pod malfunctions, Kubernetes will step in to restart or replicate the pod as necessary.

Can you use Kubernetes out of the box?

Kubernetes is open source, but it doesn't specify a particular installation tool (or version of that tool) you need to use or which components are compatible with others, and you'll have to make a number of configuration choices to get up and running, such as selecting a container network interface (CNI), storage classes, log messaging tool, and so on. You'll need to decide on the appropriate software to deploy to your clusters and appropriate settings to keep everything secure and efficient.

Hosted Kubernetes, whereby infrastructure providers like AWS, Azure, and Google, let you spin up Kubernetes clusters on-demand, take responsibility for some of that management, so you can get started developing your apps and deploying them on hosted Kubernetes services much more quickly. This convenience comes with a clear drawback: you're bound to the cloud provider and will have less flexibility, with stricter limits on cluster configuration.

What technical challenges does Kubernetes help solve?

There are a number of common issues using containers and managing them which Kubernetes can address, including the need for high availability, reliability, scalability, and fault tolerance, as well as spiralling costs. Let's take a look at some of these in more detail.

Brittle and inconsistent application environments

Since container applications aren't tied to their infrastructure, Kubernetes makes containerisation a more flexible way to run applications. You can now move applications from local machines to data centres or cloud-based development, testing, and production environments, gaining consistency across environments and encouraging the idea of ephemeral environments. You can even switch between different cloud providers more readily than with other infrastructure techniques.

Not enough horizontal scaling space for application

Undoubtedly, one of the biggest draws of Kubernetes is its scalable capabilities. It enables you to spin up the extra capacity you need to serve your users – for example, during peak shopping times – and then remove it when you don't. And as Kubernetes scales your applications according to your desired state, it monitors and maintains the health of your cluster nodes and containers as well.

High complexity of running a highly availability and reliable distributed environment

With more routine infrastructure approaches connecting together multiple regional or zonal homes for your apps to run in, so they can be more highly available and fault tolerant in case of outages, can be a very sophisticated and tricky challenge. Kubernetes simplifies this with the ability to connect together multiple clusters, regardless of where or how the backing infrastructure is hosted. If something major goes down Kubernetes will adjust accordingly.

Part two:

What are the benefits?

Whether you're getting started or you're already well on your way with containers and Kubernetes, it's important to understand how it supports your digital transformation.

Beyond basic container orchestration

Dealing with thousands of individual containers, each performing their own tasks, on their own sets of servers, can be overwhelming, so orchestration is essential. Kubernetes lightens the load using automated declarative manifests to configure, deploy, manage, and monitor all your containers, no matter how complex your environment.

Let's take a look at some of the key benefits you should take note of:

Automation to alleviate effort

You can use Kubernetes to schedule and coordinate containers across different nodes and/or clusters, scale them up and down to meet your needs automatically, and manage workloads to maintain a stable environment. And it includes built-in features for load balancing so high-traffic volumes are distributed efficiently across numerous containers.

It also allows for failed pods to be restarted automatically, can dispose of old ones, and can release updates without you having to worry about downtime. And if you want to change your desired state, the interface makes it easy. You can automatically create new preferred state pods and move existing pods to those new instances, getting rid of any you no longer need.

Scale with confidence and without complexity

Because microservices are small and stateless, they're independently scalable. You don't need to let the rest of your software know if you want to spin up a hundred more. By using cloud-native architecture, it's much simpler to create and configure a platform or application cluster. APIs and configuration files lead the charge, meaning once your app is up and running, you can make it as big or small as you need it to be.

Many companies have already benefited from scaling with Kubernetes. Booking.com, **embraced a steep learning curve** to get developers up to speed on Kubernetes. Before they were using containers, it took a couple of days to create a new service. Now, it can take as little as 10 minutes. In the first eight months of using the platform, they had built 500 new services. Open AI, meanwhile, has **written extensively** about how it scaled its Kubernetes clusters to 7,500 nodes, to support researchers working across GPT-3, CLIP, and DALL·E, as well as smaller projects.

Use it where you want

Without the need for an operating system and full software stack, containers hold just enough information to run your application. As a result, containers are lightweight and faster to work with. This makes Kubernetes highly flexible, so you can use it on a wide variety of different on-premises or cloud infrastructure and environment configurations.

You can migrate workloads into the cloud or across multiple cloud platforms, without making significant changes, and you can scale across disparate environments too. This capability means you can be agnostic when it comes to who or what is providing the compute capacity (on-premise data centre or various cloud platforms) and through that have a much more resilient and capable infrastructure.

Boost productivity

Developers will thank you for incorporating Kubernetes into their workflow. But remember, it's a steep learning curve, so you need to give them the time to get up to speed. Alternatively, set up a platform team to support your engineers as they learn. The platform team can develop tools and templates to accelerate adoption or even work right alongside your developers.

Make that commitment, and you'll soon notice an increase in your teams' productivity in return. Here are a few reasons why:

- Kubernetes allows full encapsulation of the application and dependencies, it can be very freeing for developers to pick whatever tooling they're comfortable with (within reason!).
- The declarative and ops-friendly approach means quicker deployment and feedback loops, identifying potential issues.
- Teams can specify their desired state of resources and automatically roll back if necessary.
- With less time spent taking care of infrastructure, your developers can get on with what they're there for – coding! – and operations staff can start working on more interesting things than fighting fires.
- There are useful techniques like mapping your local code directly into a running pod in a remote cluster that helps speed up the feedback loop of seeing how your code works right in a target environment.
- There's a large ecosystem of extensions, plugins, and tools you can access to support you in managing and deploying your apps. These include tools for logging, monitoring, and security, and issue resolution.
- Kubernetes will automatically schedule containers to maximise available resources, to keep your app running smoothly.

Optimise your compute budget

If you're dealing with large-scale complex applications, Kubernetes will help keep costs down. Without it, you might over-provision hardware or virtual infrastructure with unplanned spikes in mind – and manually scaling containerised applications can be complex. Kubernetes has auto-scaling built in, so you can respond automatically based on the traffic you experience in real-time. It is much more efficient and means you won't end up paying for resources you don't need.

Achieve high availability

If you want to run your containerised applications reliably, Kubernetes guarantees high availability, even when a node fails. By automatically replacing and balancing your containerised workloads to reflect demand, you can be sure your applications will run continuously and maintain an impressive uptime. Self-healing capabilities mean that containers are restarted, rescheduled, and replaced when they fail or a node in a multi-node cluster fails.

Speed up your software development cycles

To meet customer demands, you have to stay on your toes. Predicting and reacting to market needs means more frequent updates, more agile operations, and a faster life cycle than ever before. With Kubernetes embedded in your organisation, it's much easier and faster to get software features from concept stage to release, or course correct if something isn't working. That way customers can start benefiting from your ideas and improvements right away, giving you a competitive advantage.

A diverse and supportive community

Because of its popularity, the Kubernetes community is big. So when you're having issues, you can bet there's someone out there who can help (like Adaptavist). This varied community of users, contributors, and maintainers means that there are always new practices and processes developed to

extend the functionality of the platform. When you have a specific need, rather than figuring it out for yourself, you can lean on the expertise and experience of the community to find a solution.

It's a standard practice

In the early days of orchestration, there were a number of different open-source projects in the mix, but Kubernetes has risen to become the industry standard for deploying containers into production. The Open Container Initiative was also established to guide containerisation standards across all organisations. Since thousands of teams around the world are using it and all major cloud computing platforms support it, it is tried and tested. That means more reliability and stability for you.

Set yourself apart

Speaking of competition, when it comes to survival of the fittest, you're going to struggle to stay alive if you're wasting all your time focusing on infrastructure. Infrastructure is not what differentiates what you do from what your competitors do. It's the microservices – those individual value adds and ideas that your products offer customers – that really matter. With Kubernetes you can spend more time thinking about what customers are willing to pay for, building for their needs, and less time worrying about the underlying support. This needs to be framed around product teams, worrying less about infrastructure. The business's platform/infrastructure teams still care, they are providing a service to the product teams that enables this.

Part three:

Is Kubernetes right for my organisation?

In a volatile marketplace it can be hard to forecast demand, prepare for it far in advance, and have the resources available to support such an unpredictable approach. But as a business, you can't afford not to scale to meet customer needs.

Adopting Kubernetes means you can spin up capacity when you need it, such as during peak shopping times. It's not only good for your customers but cost-effective for your organisation, especially in cloud environments. Because when demand decreases, you can remove what you don't need. Meaning you pay for what you want and nothing more.

Auto-scaling explained

Auto-scaling is one of Kubernetes' most impressive and important features. Without it, every time conditions change, you have to manually provision and then scale down resources. Either you'll be running at peak capacity all the time to ensure you have the availability you need or you won't have enough resources to handle spikes. Neither scenario is ideal.

Kubernetes optimises resource usage and costs by automatically scaling clusters up and down to meet demand. For example, if a specific service experiences greater demand at a certain time of day, Kubernetes will scale it and automatically deploy more pods to account for the spike in demand. Conversely, and arguably more importantly, once the spike subsides, Kubernetes scales the extra capacity back down, saving you money.

What are the concerns with containers?

Distributed applications, microservices in particular, can benefit considerably from containers, where services can be individually contained and scaled

independently thanks to tools like Kubernetes. Container technology is always evolving, even with Kubernetes taking the pole position as the go-to approach to orchestration, hoping to address the two major issues of security, in multi-cluster and multi-cloud deployments, and infrastructure integration. There has been a heightened focus on security recently, with a number of high-profile acquisitions by service providers looking to address these issues.

Choosing a cloud provider

Even with the challenges Kubernetes and containers pose, the benefits are undeniable – faster, more efficient, and cost-savings too. Gartner predicts that 90 percent of global organisations will be running containerised applications in production in three years' time. So cloud providers are going to be busy – from managed public cloud container-as-a-service (CaaS) options from Amazon Web Services (AWS), Microsoft's Azure Kubernetes Service (AKS), and Google Kubernetes Engine (GKE) to platform-as-a service providers Google Cloud Anthos, Red Hat OpenShift, VMWare Tanzu Application Service, and SUSE Rancher.

Most public and private cloud providers offer high-level services to allow you to run software either in containers or as functions. While there are different cloud options to consider, some which require more vendor lock-in and less visibility than others, typically you only pay for what you use when you use it. Most Kubernetes-based infrastructure offers a great balance of operability, usability, and low administrative costs.

It used to be very difficult to scale infrastructure when you were tied to physical hardware, but software-defined infrastructure allows for a far more flexible and efficient way to scale up and down on demand. Kubernetes takes that one step further. It helps you advance your transformational goals, be prepared for (and protected from) volatility in workloads, and ensure you're always ready to respond to demand.

It’s clear containers and orchestration tools like Kubernetes will remain intrinsic to software development in years to come. Now is the time for companies, if they haven’t already, to invest and maximise containerisation as a key part of their digital transformation.

A platform for building platforms

The native components that make up Kubernetes – essentially an infrastructure platform – enable you to build cloud-native software systems on top of it. These include its wide-ranging infrastructure APIs, cloud-neutral flexibility, production-ready containers, ability to utilise resources, rolling updates and autonomy, extensibility, and security features. Take Spotify: it used Kubernetes to build Slingshot – a tool that creates a self-destructing staging environment.

Whether you’re going out of your way to build and deploy using Kubernetes or not, chances are a lot of the software you’re using is packaged up in containers – so if you’re not familiar with Kubernetes, you’ll need to manage those manually. And big players like Microsoft, Pivotal, and Google are getting in on the action, recognising that Kubernetes is a platform for building and running other platforms too – from container build systems and marketplaces, to app deployment and management.

Part four:

Scaling with Kubernetes and containers

So far, we've focused on all the positives. But it's worth asking the question: should all applications be built using containers and Kubernetes? Is it right for your organisation?

Ultimately, all applications can be run in this way, but if your application is monolithic, or legacy software, there's a bit of risk involved in moving it over to a containerised architecture, such as growing pains and the potential to introduce new bugs. Ideally, looking at your current architecture, ask yourself, if you're moving to an API-first approach, using microservices, are you using multiple cloud providers and/or local data centres? These factors will all play into whether Kubernetes is the right way to go.

However, that shouldn't put you off entirely – you can still make the transition, one chunk of your application at a time. And we can't deny the importance of teams learning about Kubernetes, because it's the present and future of software development.

With that in mind, here are some key questions you need to answer when making the move to a Kubernetes infrastructure:

Are you currently using containers?

It's worth noting, throwing apps into containers for the sake of it isn't like waving a magic wand – it won't automatically improve everything. You might be working with specific constraints that make containerisation untenable, for example, licensing that means you can't break apart a monolithic app or a dependency on third-party software.

Containerisation enables you to distribute your application into different components or microservices. But this requires an application architecture

that is more complex to set-up and build. The benefits, on the other hand, are going to be felt much more strongly in the longer term by adopting an application architecture that embraces containerisation.

Is it safe enough?

You're right to be concerned about the security of containerised applications and Kubernetes infrastructure. Security breaches can happen because of configuration errors. All those uncontrolled instances are going to make it next to impossible to ensure your security is spot-on everywhere. But with all the potential benefits, is this a risk you can afford to take?

Can you handle the complexity?

The complexity involved shouldn't be overlooked or underestimated. There are a lot of moving parts to Kubernetes, and the learning curve can be pretty steep. Consider where your teams are at – what experience do they have with containers and Kubernetes? How much time and resources can you dedicate to help them learn, set up, and utilise the platform? What training will you provide?

And if you don't have the right people in-house to manage and operate this kind of infrastructure, how will you attract and retain the top talent you need? There's currently a giant skills gap across the IT industry, so it won't be easy to find IT experts with the knowledge and experience you're looking for.

These are all important considerations, especially if your developers have little to no experience with containers or your infrastructure support teams aren't familiar with infrastructure automation. Without the right skills, they won't be able to troubleshoot, and you won't see the benefits. A good first step is to begin using something like Docker (for local development, for instance) without a full-fledged Kubernetes implementation right away.

Is it affordable?

Or maybe a better question here is, can you afford it? Although Kubernetes software is free, the resources required to support it are not. You need to run it on something, AWS for example, and that will cost you. If you choose to run it on-prem, you're going to need to shell out for the hardware. You'll need to weigh up your current software development costs to assess if the ROI works for your organisation.

Is it too flexible?

Inconsistency could become a problem when everyone uses different tools to run Kubernetes. While Kubernetes itself will be the same from any vendor, the configuration and installed tools your people choose to use could differ. This might mean dependencies on a specific networking or storage tool, for example, where applications will run differently if those tools are then changed. Managing multiple versions of Kubernetes across private and public installations is a reality you will have to be comfortable with and capable of.

Will it integrate with your system?

Don't lose sight of your existing infrastructure and any legacy systems. You need to consider how Kubernetes will integrate with these (or not), what you are willing to leave behind, and what is non-negotiate. This can be a challenging task for many organisations, and shouldn't be overlooked.

Are you prepared?

Even if you use a more distributed architecture and have the right skills in place, the transition to Kubernetes might not be smooth sailing. You'll have to figure out how to run your existing software alongside Kubernetes as you adapt it or try a 'big-bang' approach. The effort and resources involved will depend on how complex, or coupled, your current system and application

architectures are and whether your teams are already leveraging containers. And you'll need to adapt your existing business and technical processes to leverage Kubernetes, which will require significant buy-in from across the organisation.

Conclusion:

Implementing Kubernetes with Adaptavist

At Adaptavist our end-to-end DevOps solutions propel your digital transformation initiatives forward. Whether it's harnessing the power of the cloud for your DevOps teams, helping you implement a robust CI/CD strategy, or enabling automation of your tools and processes, we have the skills and experience to get you to the finish line.

When it comes to containerisation and infrastructure automation, we're here to help. We can show you how tools like Kubernetes and Docker – the industry-leading container platform for rapid application development and delivery – can transform the way your teams work.

We're a value-added reseller of Docker and partnered with GitLab, AWS, and Atlassian to get you the best solution for your challenges.

We can help you modernise your applications using containerisation and use Kubernetes to leverage all the benefits we covered, and avoid some of the pitfalls. On top of that, we offer DevOps maturity assessments, consultancy, implementation strategy, training, and integration solutions. So whatever stage you're at, we have a service to suit.

Can't contain your excitement for Kubernetes?

Get in touch today

Adaptavist

We help organisations transform to continuous change being their business as usual. We do this by supplying technology, providing advice, and delivering change through modern, iterative approaches to development, deployment, and application lifecycle management.

Adaptavist is Atlassian's largest platinum partner, supporting more than half of the Fortune 500. We are uniquely placed to provide our experience, expertise, and insight to help your business.

Whether you want training for your team, to build a software platform for your company, or to automate your existing tooling, we can help you. If you want to unlock the full power of Atlassian and transform your business at scale, get in touch with our team today.

Learn more or get in touch:

adaptavist.com

ATLASSIAN Platinum Solution Partner ENTERPRISE

GitLab Professional Services Partner

aws PARTNER Advanced Tier Services