ComputerWeekly.com TechTarget

E-guide

CW+ Content

Understanding Kubernetes to build a cloud-native enterprise

In this e-guide

For any enterprise wanting to revamp their infrastructure and application estates to accommodate cloud-native computing, an understanding of container technologies will be required.

Kubernetes, specifically, has rapidly become the go-to container technology for enterprise CIOs who are looking create microservices-based applications to run in private and public clouds, as well as more traditional on-premise datacentre environments.

One reason why Kubernetes has become the dominant form of container technology within enterprises can be traced back to its open source roots and the fact that it is, at its core, a Google-backed technology that is now maintained by the Cloud Native Computing Foundation.

Interest in containers has also risen as enterprises have moved to build hybrid and multi-cloud environments to run workloads in, as the technology allows developers to package up an

application (and all the dependencies that go with it), so they can run in any environment.

This not only paves the way for organisations to move applications from the cloud to on-premise environments (and back again), but also frees enterprises up from having to use specific mobile devices to run their applications on.

This e-guide looks at how enterprises are using Kubernetes, while shining a light on the steps CIOs must take to make their application and infrastructure estates container-ready.

Kubernetes vs. Docker: What's the difference?

Bob Reselman,

Docker is a technology for creating and running containers, while Kubernetes is a container orchestration technology. Let's explore how Docker and Kubernetes align and how they support cloud-native computing.

What is Docker?

Docker is a technology that is used to create and run software containers. A container is a collection of one or more processes, organized under a single name and identifier. A container is isolated from the other processes running within a computing environment, be it a physical computer or a virtual machine (VM).

Docker technology has two main components: the client command-line interface (CLI) tool and the container runtime. The CLI tool is used to execute instructions to the Docker runtime at the command line. The job of the Docker runtime is to create containers and run them on the operating system.

Docker uses two main artifacts that are essential to container technology. One is the actual container. The other is the container image, which is a template upon which a container is realized at runtime.

A container has no life of its own outside of the operating system. Thus, in terms of an automated continuous integration and continuous deployment (CI/CD) process, a real or virtual machine with an operating system must exist for Docker to work. Also, that machine must have the Docker runtime and daemon installed. Typically, in an automated CI/CD environment, a VM can be provisioned with a DevOps tool like Vagrant or Ansible.

In this e-guide

What is Kubernetes?

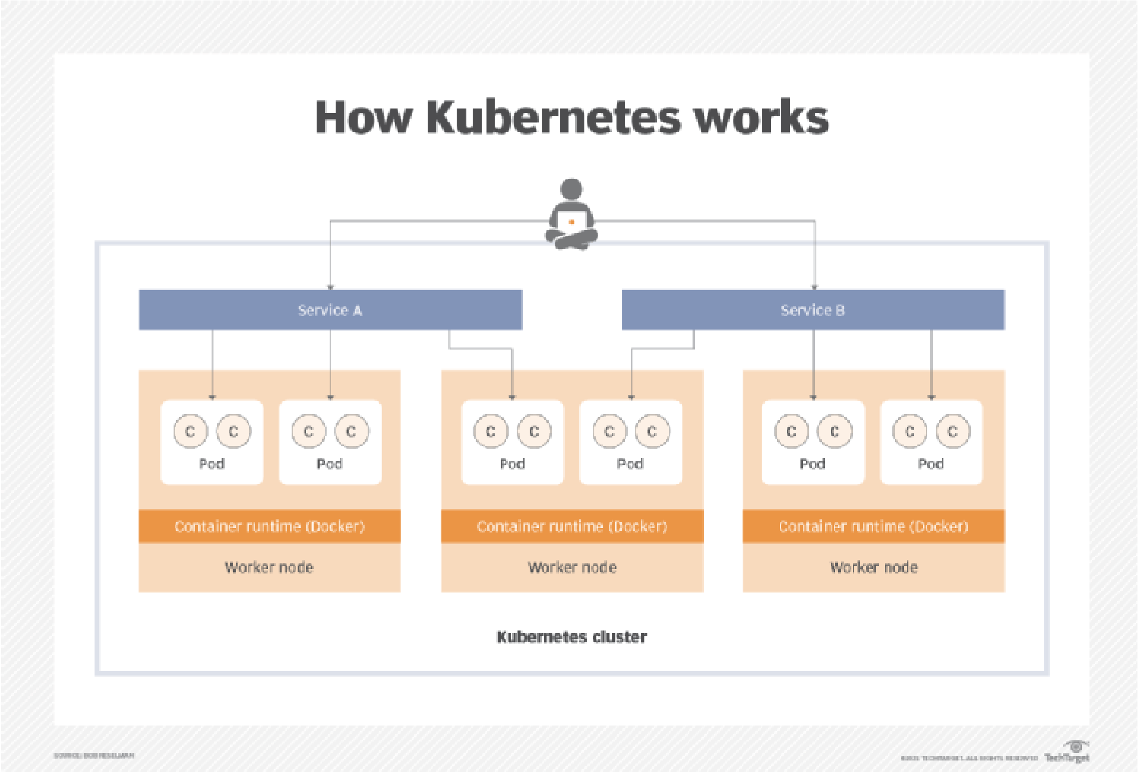
On the other hand, Kubernetes is a container orchestration technology.

Kubernetes groups the containers that support a single application or microservice into a pod. A pod is exposed to the network by way of another Kubernetes abstraction called a service. In short, the network knows about Kubernetes services and a service knows about the pod(s) that has its logic. Within each pod is one or many containers that realize the logic in the given pod.

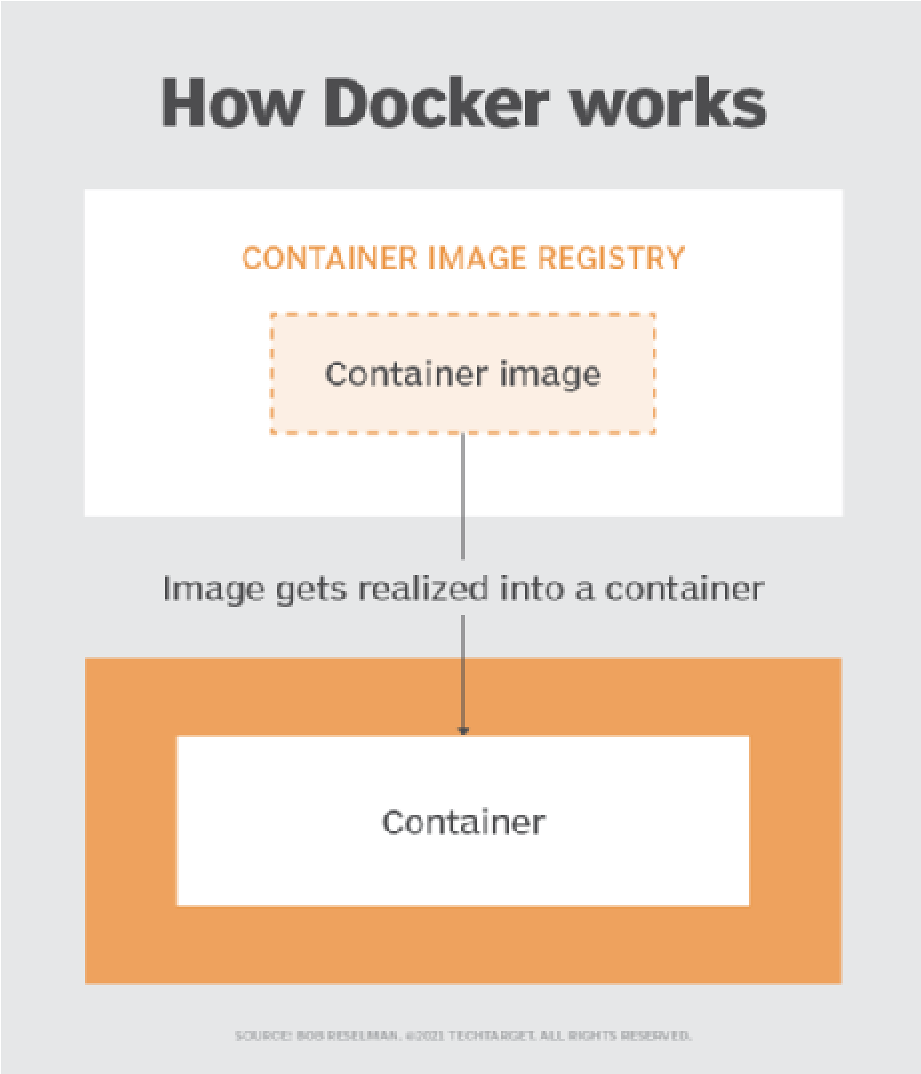
Containers, pods and services are hosted within a collection of one or many computers, real or virtual. In Kubernetes parlance, a computer is known as a node. Kubernetes runs over a number of nodes. The collection of nodes is called a Kubernetes cluster.

Kubernetes separates the node that controls activity in the cluster from the other nodes. This boss node is called the control plane node. The other nodes are called worker nodes. The containers that make up a pod run on one or many worker nodes. Each worker node in the Kubernetes cluster must have a container runtime installed.

In this e-guide



For a long time, Docker was the default container runtime used by Kubernetes. Today other alternative container runtimes such as containerd and CRI-O have become popular.


How Docker works
CONTAINER IMAGE REGISTRY
Container image
Image gets realized into a container
Container
SOURCE: BOB RESELMAN. ©2021 TECHTARGET. ALL RIGHTS RESERVED.

In this e-guide

Kubernetes and Docker deployments

Kubernetes deployments are versatile, scalable and fault-tolerant.In terms of versatility, Kubernetes supports modifying or upgrading pods at runtime with no interruption of service. You can set Kubernetes to add more pods at runtime as the demand increases, thus making applications running under Kubernetes scalable. And, if a VM goes down, Kubernetes can replenish the pods and containers automatically on another machine running within the given Kubernetes cluster of machines. Hence, Kubernetes is fault-tolerant.

Kubernetes is a complex technology, made up of components -- also called resources -- beyond pods and services. Kubernetes ships with default resources that facilitate security, data storage and network management. Also, developers can make custom resources in order to extend the capabilities of a Kubernetes cluster to meet a special need.

Kubernetes vs. Docker

The important thing to understand about Docker and Kubernetes is that one is a technology for defining and running containers, and the other is a container orchestration framework that represents and manages containers within a web application. Kubernetes does not make containers. Rather, it relies upon a container realization technology such as Docker to make them.

In this e-guide

Preparing for enterprise-class containerisation

Adrian Bridgwater,

Despite the option to move essentially ephemeral computing resources and data between public, private and hybrid clouds, there is still an all-encompassing push to deploy unmodified monolithic applications in virtual machines (VMs) running on public cloud infrastructure.

However, it is more efficient to break down an application into functional blocks, each of which runs in its own container. The Computer Weekly Developer's Network (CWDN) asked industry experts about the modern trends, dynamics and challenges facing organisations as they migrate to the micro-engineering software world of containerisation.

Unlike VMs, containers share the underlying operating system (OS) and kernel, which means a single OS environment can support multiple containers. Put simply, containers can be seen as virtualisation at the process (or application) level, rather than at the OS level.

Those essential computing resources include core processing power, memory, data storage and input/output (I/O) provisioning, plus all the modern incremental "new age" functions and services, such as big data analytics engine calls, artificial intelligence (AI) brainpower and various forms of automation.

Although the move to containers provides more modular composability, the trade-off is a more complex interconnected set of computing resources that need to be managed, maintained and orchestrated. Despite the popularisation of Kubernetes and the entire ecosystem of so-called "observability" technologies, knowing the health, function and wider state of every deployed container concurrently is not always straightforward.

Migrating to containers

"The question I am often asked is how best to migrate applications from a VM environment to containers," says Lei Zhang, tech lead and engineering manager of Alibaba's cloud-native application management system, Alibaba Cloud Intelligence. "Every customer is trying to build a Kubernetes environment, and the ways to do it can seem complex. However, there is a range of methods, tools and best practice available for them to use."

Zhang recommends that the first thing organisations looking to containerise their VM stack should do is create a clear migration plan. This involves breaking the migration into steps, beginning with the most stable applications, for example their website, and leaving the more complex applications until the container stack is more mature.

According to Lewis Marshall, technology evangelist at Appvia, the mitigation of risk alone is a huge benefit that seemingly makes the decision to containerise legacy systems easier to make. "Using inherently immutable containers with your legacy systems is an opportunity to remove the bad habits, processes and operational practices that exist with systems that have to be upgraded in place, and are therefore non-immutable," he says.

Three quick wins of rehosting

By breaking the containerisation process into manageable pieces sorted out by complexity, you can begin to prioritise quick wins and create a longer-term strategy, says Jiani Zhang, president of the alliance and industrial business unit at Persistent Systems. Here are three steps she suggests IT decision-makers should consider when looking at containerisation:

- **Rehosting:** Look to apply the simplest containerisation technique possible to get quick wins early. Rehosting, otherwise known as the lift-and-shift method, is the easiest way to containerise your legacy application and move it to the cloud. Rehosting can dramatically increase return on investment in a short time. Not all applications can be rehosted, but the earlier you start, the longer you can enjoy the benefits while you spend time on the more difficult tasks.
- **Refactoring:** Refactoring is certainly more time-consuming than rehosting, but by isolating individual pieces of legacy applications into containerised microservices, you can get the benefits of moving the most important aspects of the application without having to refactor the entire

codebase. From a time and effort standpoint, it often makes sense to only move the most important components, rather than the entire application. One practical example of this is by refactoring a legacy application's storage mechanism, such as the logs or user files. This will allow you to run the application in the container without losing any data, but also without moving everything into the container.

- **Rebuild:** Sometimes you have to cut your losses and rebuild an application that has passed its shelf life. Although this is time-consuming, these are often the most expensive and least productive applications running on your system, and the work can pay off in the long run.

In Marshall's experience, containers have the capacity to increase security while decreasing operating and maintenance costs. For instance, some legacy systems have a lot of manual operational activities, which makes any sort of update incredibly labour-intensive and fraught with risk.
Marshall recommends that IT administrators try to ensure that the cost of operating legacy systems trends downwards, towards zero. "If your system is a cost sink while adding limited business value, then updating or upgrading it should become your priority," he says. "If your system is dependent on a few individuals who regularly put in lots of overtime to 'keep the lights on', that should be a huge red flag.

"It is also worth remembering that as a system ages, it generally becomes more expensive to maintain and the security and instability risks rise."

In this e-guide

Challenges of containerisation

The immutable nature of container-based services, which can be deleted and redeployed when a new update is available, highlights the flexibility and scale they present. But, as Bola Rotibi, research director CCS Insight, pointed out in a recent Computer Weekly article, while containers may come and go, there will be critical data that must remain accessible and with relevant controls applied.

She says: “For the growing number of developers embracing the container model, physical computer storage facilities can no longer be someone else’s concern. Developers will need to become involved in provisioning storage assets with containers. Being adept with modern data storage as well as the physical storage layer is vital to data-driven organisations.”

Douglas Fallstrom, vice-president of product and operations at Hammerspace, says applications need to be aware of the infrastructure and where data is located. This, he warns, adds to the overall complexity of containerisation and contributes to the need to reconfigure applications if something changes. Also, the idea of data storage is not strictly compatible with the philosophy of cloud-native workloads.

“Just as compute has gone serverless to simplify orchestration, we need data to go storageless so that applications can access their data without knowing anything about the infrastructure running underneath,” he says.

"When we talk about storageless data, what we are really saying is that data management should be self-served from any site or any cloud and let automation optimise the serving and protection of data without putting a call into IT."

From a data management perspective, databases are generally not built to run in a cloud-native architecture. According to Jim Walker, vice-president of product marketing at Cockroach Labs, management of a legacy database on modern infrastructure such as Kubernetes is very difficult. He says many organisations choose to run their databases alongside the scale-out environment provided by Kubernetes.

"This often creates a bottleneck, or worse, a single point of failure for the application," he adds. "Running a NoSQL database on Kubernetes is better aligned, but you will still experience transactional consistency issues."

Without addressing this issue with the database, Walker believes that software developers building cloud-native applications only get a fraction of the value offered by containers and orchestration. "We've seen great momentum in Kubernetes adoption, but it was originally designed for stateless workloads," he says. "Adoption has been held back as a result. The real push to adoption will occur as we build out data-intensive workloads to Kubernetes."

Management considerations

Beyond the challenges of taking a cloud-native approach to legacy IT modernisation, containers also offer IT departments a way to rethink their software development pipeline. More and more companies are adopting containers, as well as Kubernetes, to manage their implementations, says Sergey Pronin, product owner at open source database company Percona.

“Containers work well in the software development pipeline and make delivery easier,” he says. “After a while, containerised applications move into production, Kubernetes takes care of the management side and everyone is happy.”

Thanks to Kubernetes, applications can be programmatically scale up and down to handle peaks in usage by dynamically handling processor, memory, network and storage requirements, he adds.

However, while the software engineering teams have done their bit by setting up auto-scalers in Kubernetes to make applications more available and resilient, Pronin warns that IT departments may find their cloud bills starting to snowball.

For example, an AWS Elastic Block Storage user will pay for 10TB of provisioned EBS volumes even if only 1TB is really used. This can lead to sky-high cloud costs. “Each container will have its starting resource requirements reserved, so overestimating how much you are likely to need can add a substantial amount to your bill over time," says Pronin.

As IT departments migrate more workloads into containers and put them into production, they will eventually need to manage multiple clusters of containers. This makes it important for IT departments to track container usage and spend levels in order to get a better picture of where the money is going.

Anti-food-waste app Karma taps up Google Cloud to power global expansion plans

Caroline Donnelly, Senior Editor, UK

Food waste is a known contributor to climate change, and it is a problem Swedish startup Karma is helping consumers and businesses tackle through its food rescue apps.

The first of these apps is a consumer-facing offering that connects users to food retailers in their area that have surplus stock that they can buy at a reduced price so it does not go to waste.

The second app is for Karma's retail partners, and sees the firm providing them with granular feedback on the stock level changes they can make to reduce the amount of surplus food they have each day, as well as providing a means of selling on any excess that does accrue.

The fact that it can advise its retailers in this way is an important point of competitive difference for Karma, the firm’s brand manager Charlotte Humphries tells Computer Weekly, and it is down to the way it processes its sales data.

“It’s really important to us as a way to stand out and be distinct versus the other potential food waste companies that are out there as well,” she says. “We have a direct competitor that is not able to do such a thing because they don’t sell item by item on the app. They sell a ‘mystery bag of food items’ for less at the end of the day.

“By selling item by item and by using machine learning, we’re able to offer a solution that actually stops the symptoms of food waste through redistribution, and the cause of food waste, which is overproduction,” she adds.

The company is in the process of building out its presence in the UK and France, having already established itself in its native Sweden. It claims to have saved more than four million meals from going to waste since its launch in 2016.

Enabling growth

Karma did not initially start out as an app solely dedicated to addressing the issue of food waste, Elsa Bernadotte, the firm's co-founder and chief product officer tells Computer Weekly.

"Like most good ideas, it started out in 2015 as a not-so-good idea, and for the first eight to nine months we used to say it was a failure. At that time, it was a deals platform, and a bit like a crowdsourced Groupon [service]," she says.

"By selling item by item and by using machine learning, we're able to offer a solution that actually stops the symptoms of food waste through redistribution, and the cause of food waste, which is overproduction"

Charlotte Humphries, Karma

The company soon reached a "sink or swim crunch point" with the app, which prompted a decision by the team to make more of one of its best-performing features – its meal deals – and seize on that to tackle food waste. Three weeks after that decision was made, Karma in its current form was born.

"Once we started looking deeper into the environmental implications of food waste, and we understood just how vast and monumental the problem of food waste really was and still is, it then became our ambition to solve one of the world's largest climate issues using technology," says Bernadotte.

After several years of steady growth in Sweden, the company secured additional investment through a funding round that would pave the way for the company to expand its operations to the UK and France in late 2018.

But before the company could do that, it needed to address some shortcomings in the app's underlying infrastructure that had emerged, which had the potential to stunt Karma's international growth and its ability to innovate. This, in turn, prompted the firm to embark on a shake-up of its infrastructure.

The source of its technology issues lay in its reliance on a simple-to-use bare metal infrastructure that required a lot of expensive manual handling and maintenance. So much so that keeping it up and running required three full-time DevOps engineers, which is a sizeable overhead for a startup to bear, says Karma product manager Koen Brörmann.

"When we started expanding internationally, our infrastructure became a bottleneck – from a scalability, innovation and delivery perspective. We had a large DevOps team of three people dedicated to maintaining that," he says. "What we had was working, but there was so much manual work involved that it felt prohibitively slow."

For example, if the company wanted to ship an update to the application, that would require logging into a remote server, pulling in new code, followed by a manual restart of the program using a node process manager.

"We had a really bifurcated setup where you had a DevOps team and an engineering team, and we wanted to move to a situation where our engineers are also responsible for the delivery and maintenance of the infrastructure," adds Brörmann.

"We wanted a setup that would not only allow us to move really fast [from an innovation perspective], but also give our engineers ownership."

Up in the clouds

With the help of its newly acquired investors, the company set about scouring the market for a public cloud provider that could help it simplify the management of its infrastructure operations, before deciding on the Google Cloud Platform (GCP).

The migration took around 12 months, with Brörmann crediting a company decision made prior to the move to transition its application over to a microservices-based architecture with helping make the shift to GCP a very smooth process.

"In about six months, we were 80% done [with the migration] and the last 20% took another three to six months, but we were able to move over fairly quickly due to our microservices setup," he says.

As part of the migratory process, the firm set about replicating the app using the Google-developed open source container-based technology Kubernetes, which in turn led to Google Kubernetes Engine (GKE) forming the core of its revamped infrastructure.

GKE is billed by Google Cloud as a fully managed Kubernetes service that provides enterprises with an autopilot-like mode of operation, which the Karma team said helped it achieve its goal of simplifying its operations management processes.

It has since sought to automate the management of its infrastructure further by leaning on Google Cloud Functions and Google Cloud Run, while the migration also saw it start tapping into Google BigQuery to aid the management of its app databases.

"I was really excited about BigQuery because I come from a traditional background where there was a whole data management team, and if I had a request, I had to send it to them, they would write a query, get the data and give that to me," says Brörmann.

"So I was excited to be able to manage all that data live and get access to it directly myself – and 80% of the team can get the data they want too. That has allowed us to move even faster than we did before, while remaining data-driven," he adds.

As an example of this, Brörmann cites how quickly the company was able to expand the takeaway functionality of its apps in response to the onset of the Covid-19 coronavirus pandemic in spring 2020 to include delivery options as well.

“We were able to pivot to delivery super fast. I was blown away by how fast we had something not just built, but rolled out and [live] the next day,” he says.

A proactive approach

The move to GCP also brought uptime improvements, and the Google team is proactive in helping Karma’s engineers find new ways to expand the functionality of its apps so the company can do more to help its retail partners address the causes of food waste, adds Brörmann.

To this point, he shares an example whereby Google’s engineers provided the Karma team with a walkthrough of how using its BigQuery ML tool would enable the startup to create and deploy machine learning (ML) models using standard SQL database queries that would help retailers tweak their stock levels to prevent food waste.

“We knew that BigQuery ML was available, but we hadn’t found a way to use it at that point, so Google approached us and said, ‘We can do a small slide deck presentation and walk you through what the opportunities are’,” says Brörmann.

"We had three of our engineers talk to one of the Google people about that, and based on that talk we started using BigQuery ML, which has opened up a lot of avenues for us with prediction and prevention of food waste, which are areas we want to do more with in the future."

Specifically, BigQuery ML is currently being used by Karma to provide its retail partners with an indication of how high or low their foot traffic is likely to be in the coming days so they can prepare more or less food depending on what the data tells them.

Looking ahead, the company is hoping to do more with the GCP portfolio of artificial intelligence (AI) and machine learning tools to refine its operations further, which includes tapping into its Vision AI image recognition tool to enable restaurant partners to upload their menu data to the app far more quickly.

"We want to scale faster and add more businesses to the platform, so we're experimenting with using Vision AI so our partners can take pictures of the menu and all the data from that will be in the app within 10 seconds, whereas at the moment that takes 30 minutes or so," says Brörmann.

The Karma team also has aspirations to take the brand worldwide, which is something that will be made far easier by the fact Google has datacentre regions in the US, Europe and Asia too.

"With Google, you don't really have to worry about it not having availability in other countries, so that's definitely a load off our mind for the future," adds Brörmann.

Getting more CW+ exclusive content

As a CW+ member, you have access to TechTarget's entire portfolio of 140+ websites. CW+ access directs you to previously unavailable "platinum members-only resources" that are guaranteed to save you the time and effort of having to track such premium content down on your own, ultimately helping you to solve your toughest IT challenges more effectively—and faster—than ever before.

Take full advantage of your membership by visiting www.computerweekly.com/eproducts

Images; stock.adobe.com