Next stage of virtualisation: Containers

In this e-guide

In this e-guide:

The buzz and hype surrounding container technologies has reached fever pitch in recent years, prompting CIOs and IT decision makers to mull over what role, if any, they should and could play in their digital transformation plans.

In simple terms, containers are a form of operating system virtualisation that allows developers to isolate and package up all or part of an application, effectively into a portable building block.

The technology's appeal can be traced back to the portability it gives IT departments, in that containers allow developers and IT operations teams to create, deploy and run applications in the environment of their choosing. Whether that be in the cloud, on-premise or across multiple virtual machines.

For this reason, the adoption of container technologies has closely followed the take-up of cloud computing in

In this e-guide

enterprises, as the former makes it easier for organisations to move applications between different environments and even providers.

Their use has also accelerated as enterprises have moved away from their traditional, monolithic application stacks and adopted a microservices-style approach to app development.

In this e-guide, we take a closer look at what containers are, the technologies that complement and enhance their use, and get a first-hand insight into the impact their use can have on an organisation's IT strategy and setup.

Caroline Donnelly, datacenter editor

In this e-guide

What are containers and microservices?

Bob Tarzey, guest contributor

Containers encapsulate discrete components of application logic provisioned only with the minimal resources needed to do their job.

Unlike virtual machines (VM), containers have no need for embedded operating systems (OS); calls are made for OS resources via an application programming interface (API).

Containerisation is, in effect, OS-level virtualisation (as opposed to VMs, which run on hypervisors, each with a full embedded OS).

Containers are easily packaged, lightweight and designed to run anywhere. Multiple containers can be deployed in a single VM.

A microservice is an application with a single function, such as routing network traffic, making an online payment or analysing a medical result.

The concept is not new; it has evolved from web services, and stringing microservices together into functional applications is an evolution of the service-oriented architecture (SOA), which was all the rage a few years ago.

In this e-guide

Not the same thing

Containers and microservices are not the same thing. A microservice may run in a container, but it could also run as a fully provisioned VM. A container need not be used for a microservice. However, containers are a good way to develop and deploy microservices, and the tools and platforms for running containers are a good way to manage microservice-based applications. In many cases, the terms can be interchanged.

Containers have been integral to Unix and Linux for years. A recent change has been the ease with which they can be used by all developers, and an entire supporting ecosystem has grown up around them. Containerisation is not something happening on the fringes of IT, it is core to the way many web-scale services operate and is increasingly being adopted by more conservative organisations. The suppliers mentioned in this article cite customers ranging from the NHS to large banks.

There are many suppliers involved, but no one disputes that Docker has led the charge and sits at the heart of the market. Docker says millions of developers and tens of thousands of organisations are now using its technology. However, another statistic indicates the novelty of containerisation for many, with only 40% of Docker's customers running containers in production.

In this e-guide

Docker's dominance does not mean it holds a monopoly; far from it. Across the whole container ecosystem, there is plenty of choice. There are many startups and the great and good of the IT industry are all on board, as a glance at the sponsors of the February 2016 Container World event shows. The top sponsor is IBM, which is one of Docker's three main global go-to market partners, along with Microsoft and Amazon Web Services (AWS).

Open source componentisation

Multiple containers are deployed in clusters and managed using a range of tools. Many of these containers will be pre-built components that can be layered together to build up application images. A prime benefit is that it is easy to "overwrite" an individual container while the application is still running – less scheduled downtime means better business continuity.

This has led to the rise of the DevOps concept, which allows faster deployment of new software capabilities directly into an operating environment.

Much of the core containerisation technology is open source, and suppliers that have previously eschewed it, such as VMware, are being drawn in. At its heart is the Open Container Initiative (OCI) launched in 2015. This operates under the auspices of the Linux Foundation to create open industry standards around container formats and their runtime environment. Docker

In this e-guide

has donated its own format and runtime to the OCI to serve as the cornerstone.

Many containerised components are downloadable from open collaboration projects such as GitHub and Docker Hub. As with all open source technologies, the suppliers that operate in the market must earn their money by providing stable versions with associated support services.

The container stack

There are four technology layers that need consideration:

1. Container operating systems

Even though containers do not have an embedded OS, one is still needed. Any standard OS will do, including Linux or Windows. However, the actual OS resources required are usually limited, so the OS can be too.

This has led to the development of specialist container operating systems such as Rancher OS, CoreOS, VMware Photon, Ubuntu Snappy, the Red Hat-backed Project Atomic and Microsoft Nano Server. The benefit here is that the VMs provisioned to run containers are lightweight (some run in about 25MB) and when it comes to security, the attack surface is minimised. Cloud platform providers are embedding their own support.

In this e-guide

2. The container engine

This is where Docker dominates, but there are competitors, such as CoreOS Rocket (Rkt). AWS says Docker is by far the most popular engine with its customers, and therefore the focus of its support plans. Engines come with supporting tools, for example the Docker Toolbox, which simplifies the setup of Docker for developers, and the Docker Trusted Registry for image management. There are also third-party tools, such as Cloud66.

3. Container orchestration

Containers need to be intelligently clustered to form functioning applications, and this requires orchestration. Orchestration is where much of the differentiation lies in the containerisation ecosystem and it is where the competition is hotting up most.

The engines provide basic support for defining simple multi-container applications, for example Docker Compose. However, full orchestration involves scheduling of how and when containers should run, cluster management and the provision of extra resources, often across multiple hosts. Tools include Docker Swarm, the Google-backed Kubernetes and Apache Mesos. You could use general-purpose configuration tools, such as Chef and Puppet (both open source) or commercial offerings such as HashiCorp Atlas or Electric Cloud ElectricFlow. None of these is container-specific, however.

4. Application support services

Many additional tools are emerging to support containerised applications – some examples follow.

There is a danger of containerisation ending up like herding cats, which is a problem for application portability. An organisation may want to move an app from one cloud platform to another. Software suppliers will need to consistently recreate their applications for user deployments. How do you ensure all the dependencies and necessary containers are copied and recreated? Rancher Labs' core product (it also has an OS) enables applications to be built up from containers so that the full operating environment can be recreated, including the containers themselves, load balancers, networking, and so on.

Networking is an issue, especially across platforms. In 2015, Docker released Docker Networking to enable virtual connections between containers. UK-based Weaveworks also focuses on networking with WeaveNet, a micro-software-defined network (SDN). Metaswitch's Project Calico is all about making container networking more secure through the dynamic construction of firewalls, taking policy from orchestrators.

Docker, too, is developing new tools to support the lifecycle of containerised applications. Last year, it acquired a company called Tatum, an on-demand service for building, deploying and managing applications. The Docker Universal Control Plane (UCP) provides similar on-premise capability. Both

products are yet to go into production. Weaveworks also has a tool called WeaveScope for production monitoring of containerised applications.

Containerisation platforms

All the big industry players are joining the containerisation bandwagon, with a range of initiatives under way.

Google is an old hand with containers – it has been developing and deploying billions internally for years. The company has been a major contributor to various container-related open source projects, included the Kubernetes orchestrator, which it donated.

Google has now opened up this expertise to its customers and added the Google Container Engine to the Google Cloud Platform.

Microsoft has added container support with Windows Server Containers enabling the sharing of the OS kernel between a host and the containers it runs. Hyper-V Containers expands on this by running each container in an optimised virtual machine. For the cloud, there is the Azure Container Service (ACS), developed in conjunction with Docker, which can manage clusters of containers with "master machines" for orchestration. ACS also supports other orchestration tools, such as Kubernetes.

AWS customers were quick off the mark to deploy containers on its EC2 platform, so AWS has followed up by providing a cluster management and

scheduling engine for Docker called the EC2 Container Service (ECS). This is supported by the EC2 Container Registry (ECR) to support the storing, management and deployment of container images. ECS is widely available, whereas ECR is currently available only in the eastern US.

VMware has not taken the move to containers lying down. Later this year, it will release vSphere Integrated Containers, using VMware's Photon OS to turn VMs into Docker-like containers based on OCI. This will allow users to take advantage of existing vSphere support tools. In a first for VMware, it has open-sourced both the PhotonOS and the associated Photon Controller.

Other examples include IBM Containers for Bluemix, Rackspace Carina (based on OpenStack Magnum, embedded support for containers and orchestration). Another open source initiative is Deis, a platform-as-a-service (PaaS) based on CoreOS.

Big decisions

For developers, the open source nature of the container marketplace makes it easy to access the technology and supporting tools and crack on with building agile applications through a DevOps-style process.

This offers many benefits to businesses, but they must consider the supporting platforms and technologies that are endorsed to ensure longer-

term stability and support. Making such decisions will not be easy as the containerisation market changes.

Next article

In this e-guide

Microservices: Small parts with big advantages

Danny Bradbury, guest contributor

The microservices concept is not really anything new – it is the implementation that has evolved.

The idea is to break down traditional, monolithic applications into many small pieces of software that talk to each other to deliver the same functionality.

This will give those who lived through component-based software, web services and service-oriented architecture (SOA) in the early 2000s a sense of déjà vu.

They were meant to do something similar. So what's the difference? "Microservices are much lighter-weight than SOA, with all of the standards which that entailed," says Ben Wootton, co-founder of Seattle-based DevOps consultancy Sendachi.

SOA was a supplier-driven phenomenon, with an emphasis on complex enterprise service buses – a form of middleware needed to communicate between all the services.

In this e-guide

"Message standards are looser and are exchanged over lightweight message brokers," says Wootton. "The tooling has evolved from the open source community rather than big enterprise."

Speed and agility Companies are interested in microservices because they can bring speed and agility and encapsulate relatively small business functions, says Wootton. A currency conversion service is a good example, or an e-commerce shopping cart.

Companies can develop services like these more quickly and can change them more readily, because they are dealing with smaller code bases. This is not something that traditional, monolithic applications with code millions of lines long were designed for.

The testing overhead is immense when changing such vast code, because of all the interdependencies involved. The other advantage is scalability. Microservices are designed to work in cloud environments, which can increase and decrease the computing resources needed for particular applications at will. If you need more computing power, simply start up another microservice on another cluster of computers. You can't do that easily, if at all, with monolithic software that is designed to scale up on one piece of hardware.

Distributed computing like this also makes it easier to recover from infrastructure failures. Microservices are designed to be easily replicable, so

many of them can be run to pick up the slack should a particular service stop working.

Cloud-native software model

All of this makes microservices useful for cloudnative applications, which are designed to run in cloud environments with lots of commodity hardware resources that can dynamically respond to fluctuations in demand for certain applications.

These infrastructures are designed to fail over quickly. If a server dies, there is another one in the infrastructure to take its place. For microservices to operate that way, they need to interact differently with the IT infrastructure, says Wootton. "You need to lean on automation a lot more," he adds. "You might find your previous application turns into 50-100 independent services. Maybe they have to be duplicated for resilience. You are quickly left with hundreds of processes to be managed."

To automate the management of the microservices and the provisioning of the infrastructure supporting them, the whole computing stack needs to change. The microservices software itself, or the software layer that manages it, must talk to the IT infrastructure to provision CPU cycles, networking and storage.

In this e-guide

This calls for software-defined infrastructure, which has underpinned the management of cloud-based resources for a while. Companies ranging from IBM to OpenStack are proposing IT environments in which computing, networking and storage resources are accessed and controlled via software application programming interfaces (APIs), rather than from a systems management tool or command line.

How can you prepare your infrastructure for this? Suppliers of converged and hyper-converged systems would like you to throw away your expensive storage array and replace it with dumb drives that their software will control for you. But that isn't necessary, says Donnie Berkholz, director of development, DevOps and IT operations at analyst firm 451 Research. You can typically integrate existing supplierspecific infrastructure with these new cloud management systems, he says: "A lot of cloud environments already have plug-ins.

OpenStack networking has all different kinds of back ends." Automating the creation and deployment of applications like these requires some form of software container that shields it from any idiosyncrasies in the platform.

A microservice may be deployed on a server running a different network driver, Linux distribution or version of Python than the one on which it was developed, but the container shields it from that. "Microservices can be built in any language and any stack, as long as the boundaries can be defined," says Kamesh Pemmaraju, vice-president of product marketing at Mirantis,

which creates software and services to help IT departments using the OpenStack private cloud management system. "They can be thrown into a container, and it is portable."

Talk about Docker

Most people describe Docker when talking about containers. This open source project shares elements of the operating system between different containers, but bundles all the application's dependencies and libraries in the container itself. But this isn't the only show in town. Containers have been around for a while, from Solaris Containers (Zones) to the Linuxbased LXC containers on which Docker was originally based. RunC is a container runtime designed to implement a container specification standard created by the Open Container Initiative, while Virtuozzo has its own Linuxbased container technology called OpenVZ. And then there's CoreOS, which has its own rkt container runtime.

Suppliers have been quick to jump on this. VMware launched its own technology, called vSphere Integrated Containers (VIC), last August, which is designed to let developers connect to virtual container hosts using a Docker command line interface. VMware containers run alongside standard virtual machines.

Microsoft announced support for Docker containers on Linux virtual machines (VMs) from within Azure in June 2014. Since then, it has worked

on supporting Docker containers on Windows Server, and also announced its own container technology for the Hyper-V hypervisor, along with a Nano Server minimal footprint installation of Windows Server designed for container use.

Where to start?

It may all sound exciting, but ripping out and replacing existing applications with microservices is not a realistic proposition for anyone, so where should a firm start?

Gently does it, says Sendachi's Wootton, who recommends an iterative approach. "You just dip your toe in and get used to that lifecycle," he says. "I would pick off new functions and slowly bring them into microservices. I would never re-architect a whole application for the first time." A company with an established inventory management system might steer clear of replacing it straight away, but it might consider implementing other new functions on its website as microservices, such as a customer chatroom or a product recommendation service, perhaps.

Cut code quickly

Areas where this makes sense are where you need to cut code quickly and innovate rapidly. Mobile apps are a good example, as are customer-facing services that you expect to be used at scale. The legacy batch order

processing software that has been doing its job reliably for years may not need the microservices treatment, however.

This practice phase is important, because microservices involve a deep change to the software development and deployment process. The biggest mistake, says Wootton, is companies trying to implement microservices without changing their old ways of working. "You want to move to a DevOps model where people are working more collaboratively," he adds. DevOps involves a meeting of minds between developers and operations staff, says 451 Research's Berkholz. "It involves operations staff learning what it looks like to be software developers, but also developers learning what it looks like to do production. You can't do microservices until you've done both of those."

Work together

This approach enables the two parties to work together in a world where infrastructure is provisioned using software interfaces at a moment's notice, tailored for the development, testing and production deployment of many tiny applications. In practice, that means operations staff might be checking configuration instructions out of GitHub instead of just writing their own batch files. And while operations staff may be responsible for the platform that code is running on, the developers become responsible for their own code's operation. "They don't get to hand it off any more and be done," says Berkholz. "They have to say 'I'm on pager duty, and if my code breaks at

2am, I get woken up'." This might give developers and operations staff alike pause for thought. Microservices is not a free lunch. It needs sophistication in technical infrastructure, along with a highly mature IT team. Many firms will have their work cut out before they are ready.

Next article

In this e-guide

In this e-guide

How to start building enterprise momentum for microservices

Cliff Saran, managing editor

The Cloud Native Computing Foundation (CNCF) is just a year old, but at the start of its second conference in Berlin, it unveiled a number of initiatives that aim to improve support for some of the major container technologies.

Docker's core container runtime, together with Kubernetes runtime and gRPC, a high-performance remote procedure call technology, have been accepted by the Technical Oversight Committee as incubating projects within CNCF.

CNCF's vision is to promote interoperability and portability across different container execution environments. In other words, it should be possible for an application running in a Docker container to talk to a containerised Kubernetes application, and vice versa.

A cloud-native application runs in such containers and can make use of additional code, or microservices, running in different containers.

In this e-guide

While Docker and Kubernetes are well established, many of the technologies presented during the opening keynote are very much unknown, particularly in mainstream enterprises.

What is even more remarkable is that some of the key open source technologies began just a few years ago as tiny operations, and are now supported by leading open source companies with hundreds of code contributors.

One of these is Prometheus. Brian Brazil, founder of Robust Perception, worked on the original project. "It started in 2012 as one company with two people," he said. "Prometheus now has more than 300 contributors, and 500 companies are using it in production."

Prometheus is a metrics monitoring system which uses a time series database and has its own query language. A few enterprises are now engineering IT systems to use containers to improve the way software is developed, and Prometheus is among the tools of choice for monitoring such systems.

Justin Dean, senior vice-president for technical operations at Ticketmaster, said: "We use Docker for containers and the Kubernetes ecosystems, plus we make heavy use of Prometheus. People love how much easier it is to handle time series with Prometheus."

In this e-guide

Ticketmaster was one of the major organisations on a CNCF panel discussion to showcase the use of containers in big business. "It is really hard to be nimble and roll out new products and features compared to startups," he said. "A few years ago, we started to realise we needed to get faster and deliver software faster, otherwise we would start losing to competitors."

Using a DevOps approach

Industry consensus is to use a DevOps approach to software development, delivery and deployment, giving coding teams responsibility for all aspects of the software they build.

"Anyone who has been through the process understands the work required," said Dean. "We wanted autonomous software delivery, where every team had everything they needed to deliver their product to market, and be responsible even for profit and loss. We were trying to create mini micro businesses all across the company that could move as fast as the competition instead of large, monolithic teams."

The challenge in achieving this is both cultural and technical. There were too many tools required for the engineers at Ticketmaster to use. "We quickly got to a situation where we needed to revamp the tech, and ended up in the container space and the Kubernetes ecosystem," he said.

Global ticket distribution system Amadeus began its journey in 2014. Eric Mountain, senior expert of distributed systems at the company, said research and development teams also need to buy into the idea that DevOps is the right approach to develop software.

"R&D already has a system that works, so why move? You need an awful lot of communication to convince them it will given them something easier," he said. "Containers makes that conversation easier."

A shift in culture

Dean agreed the shift in culture was Ticketmaster's biggest challenge. "We have 350 products and tonnes of teams with different software stacks, and they deliver software differently. There's a lot of muscle memory built up. When you force a change, it has to be welcome."

But, like Mountain at Amadeus, Dean found containers made the approach easier. The container effectively ring fences the code being developed. The coders cannot simply tweak some other piece of code or even alter the hardware, to make their software work. Essentially, the new code being developed plugs into a pipeline.

"When you deliver software in a container through a pipeline, it forces a massive change," said Dean. "This single-handedly had more impact than anything else and dovetailed into DevOps."

In this e-guide

The combination of containers, software-based infrastructure and microservices represents a major evolution in the way applications are deployed and managed. It is a fast-paced, changing world, where standards and preferred tools are still emerging. When Amadeus began its journey, Prometheus did not exist, so it needed to engineer its own monitoring, but this was done in a way that allowed it to plug in and replace functionality as the tools ecosystem evolved.

Arguably, for the traditional enterprise, this is perhaps a major roadblock. Unlike old-school enterprise applications that were vertically integrated and monolithic, the new world order is cloud-native, built around loosely coupled containerised applications where developers need to define the bits of code their applications require (known as dependencies).

"There is a pretty big hurdle to get an application into a container," said Dean. "The barrier to entry is steep and there is a whole new language to learn."

Next article

In this e-guide

Case study: SEO startup uses Elastichosts' containers to solve cloud app scalability issues

Caroline Donnelly, datacentre editor

Enterprise website auditor Seopler.com has highlighted the challenges startups face around billing and support when building a business in the cloud.

The Dublin-based startup's website-crawling software is used by enterprises such as Nissan Ireland to sniff out broken links, site redirects, invalid HTML markups and other elements that could adversely affect how highly their pages are ranked by search engines.

Once a site crawl is completed, a PDF is generated and sent to the user advising them of the corrective action they must take to improve the search engine optimisation (SEO) credentials of their website.

The compute resources that the Linux-based software carrying out the crawl requires to do its job varies depending on the size of the website and how many pages it contains, which is one of the reasons why Seopler decided to run the software in the cloud.

Also, the company rarely gets much prior warning about the size of the website it will be called upon to audit before a customer enlists its help, which makes it difficult to know how much compute power will be needed to perform its tasks.

"As Seopler crawls larger websites, the more data it needs to collect, and we used SQLite databases and kept those in-memory, so it's very fast to collect that data when it crawls the site," Graham O'Shea, the company's CEO, told Computer Weekly.

Time is of the essence during the process, because the longer a crawl takes, the more likely it is to cause a level of disruption to the website undergoing the audit.

In the lead-up to the company's launch last year, O'Shea set about finding a suitable infrastructure-as-a-service (IaaS) platform to run the software on, and initially opted for Amazon Web Services' EC2 service.

Unforeseen problems

However, the unpredictability of the size of each job undertaken by Seopler's software created some unforeseen problems when it came to costs and resource provisioning.

"The problem with Amazon EC2 was we had to specify them before we did a crawl and, based on a million- page website, we would have had to

commission an EC2 virtual machine with 8GB of RAM and 50-60GB of hard disk space before we ran the job," he said.

"It started off and it looked quite cheap, but when the bills started coming in, we realised this isn't going to work for us in the long term because you have to pay extra for support in the long term.

"Also, we don't know the client's website. They sign up for it at a switching level, and we don't know what size their site is. So to do that for each client is really expensive. We tried it with Azure, and it was the same thing."

While the company was working through these challenges, it had the added pressure of needing to have a workable demo in place to present to the crowd at the Web Summit Conference in Dublin on 3 November 2015.

Seopler was already using Elastichosts and its cloud servers to stand up some of its websites, so O'Shea and his team decided to give its container service a go.
"To be honest, it was there staring us in the face all the time, in that they had this new container service they had launched, and we didn't really understand it," he said. "It was only after going down the EC2 and Azure route that we worked out that it did exactly what we were looking for.

"We were able to set up the scripts to build everything in a day, whereas with Azure or EC2 it took a week or more, and it still didn't do exactly what we needed it to do."

The Web Summit deadline was met comfortably, and now Seopler, having worked through its hosting challenges, is looking to build out the software's functionality and expand its operations overseas with the help of Elastichosts' international datacentres.

"When someone performs a crawl, we want that done as quickly as possible, no matter where they are in the world," said O'Shea. "So, if we get a subscriber in the US, we will automatically put them on a server being run out of the US. That will just accelerate the crawl and speeds things up because they're performing it local to them."

Next article

In this e-guide

Interview: Cloud Foundry on its 2017 awareness-raising plans for open source PaaS

Caroline Donnelly, datacentre editor

Two years have passed since the Cloud Foundry Foundation was created, and in that time, use of the open source platform-as-a-service (PaaS) that shares its name has increased markedly across the US and Europe.

The Cloud Foundry was originally developed in-house at VMware before being handed over to EMC/VMware spin-off Pivotal Software, which, in February 2014, put in motion a plan to establish an open governance model for the PaaS. This, in turn, paved the way for the foundation to be established in January 2015.

The number of organisations and community members committed to supporting the software's ongoing development has risen accordingly in that time, with the likes of Atos, IBM, Pivotal and SAP all offering enterprises their own distribution and take on Cloud Foundry.

Among Cloud Foundry's user base are a number of household names, including banking giant HSBC, car manufacturer Volkswagen and the UK's Government Digital Service (GDS).

According to the foundation, central to the platform's success are its openness and container-based architecture, which allow enterprises to run applications at scale in an on-premise datacentre or in the public cloud without having to modify any code.

Although the pace at which the enterprise market has moved to adopt Cloud Foundry is encouraging, Abby Kearns, the foundation's executive director, tells Computer Weekly that the platform is still considered one of the cloud market's "best-kept secrets".

"Cloud Foundry is an interesting product in that many products suffer from huge amounts of awareness and little adoption, but Cloud Foundry has tremendous adoption and very little awareness," she says.

Best-kept secret

The "best-kept secret" label might work as a marketing ploy for businesses in the retail or hospitality sector, but it is not so helpful for an organisation that is looking to build awareness of its brand across the globe, says Kearns. "The best-kept secret descriptor – while that's cute for restaurants, it's not so cute for our product."

In this e-guide

Part of the problem is the product itself and what it does, says Kearns, who goes on to describe it as a technology that essentially fulfils the roles of the datacentre's operating system and drives business innovation from the ground up, starting with the "foundations" of the infrastructure.

"It's not sexy and fun, but it's an enabler for you to build a business on, and an enabler for you to build and change the way you think about applications," she says. "And that's where it gets exciting.

"When we think about HSBC, its business is centred on building mobile apps that customers can use to transfer money, view their bank balance and deposit cheques, and Cloud Foundry is the invisible technology behind the scenes that allows it to do that."

The best way to explain the difference that using Cloud Foundry makes to any organisation's business transformation efforts is to imagine what life would be like if it did not exist, she says.
"How often do you talk about how great your Windows or Mac operating system is? But if you can imagine life without it, it would be a lot more complicated. For instance, you can still run your applications without an operating system – but it's a big pain," she says.

To remedy this, Kearns – who took over as the foundation's executive director in November 2016 – says raising Cloud Foundry's profile is a top priority for the organisation's leadership team this coming year.

In this e-guide

Essential to this process will be drawing on the support of the organisation's community of existing certified providers, all of whom have played an important role in helping Cloud Foundry make inroads into the enterprise.

"Some of it has been down to word of mouth and leveraging the expertise of the providers that have capabilities in that space," says Kearns. "For instance, Pivotal [a certified partner] has a large digital transformation practice, and so does IBM, and they have been really helping their customers take advantage of what Cloud Foundry can provide."

Depending on what enterprises want to do, as well as the capabilities they might have in-house to achieve their digital aims, organisations have a choice between using Cloud Foundry as is, or one of the distributions created by its certified partners.

"We have 11 certified providers of Cloud Foundry right now," says Kearns. "Pivotal is one of the better known, IBM Bluemix, SAP Hana Cloud Platform runs on Cloud Foundry, SUSE has recently acquired HPE's OpenStack and Cloud Foundry assets, so they will now have a distribution too.

"The majority of users don't use the open source – they are using a distribution to allow a variety of things. Some have a great user interface and some do a fully-managed version, so they can offer that turnkey experience."

In this e-guide

Developer focus

Central to the success of any digital transformation project is the input and output of the developer community. For this reason, the foundation is also ramping up its efforts to court the developer community this year, as part of its awareness-building activities.

"Developers are front and centre for companies that want to transform, and the developer capabilities mean a lot to companies that are trying to become software companies through digital transformation," says Kearns.

"So we are really going to talk more about developers this year and the value Cloud Foundry can bring to them, and enable them to develop and deploy applications into production quickly and easily."

In this context, the need for speed is essential, says Kearns, as enterprises look to their developers to help them keep one step ahead of the competition, and as they battle to manage the demands of their legacy systems with newer apps and service deployments.

"As you think about legacy versus greenfield applications, working out what is the best path forward will require the business to take a step back and ask themselves what they are trying to do, what their vision for the future is and work out how technology can help," she says.

In this e-guide

"That might sound simplistic, but it's hard because it's not something businesses are really accustomed to doing."

Either way, for most enterprises, the likelihood is that they will end up managing a hybrid estate featuring a mix of legacy and on-premise systems, as well as newer cloud-based applications and services, including systems that have not even been created yet.

"Realistically, for most organisations, in five years from now, 90% of the applications they are going to be managing are not written yet," says Kearns. "Having a vision to the future and figuring out how to reduce the investment in the legacy they have is also going to be key."

This is where developers and the open and extensible nature of Cloud Foundry comes into its own, and will continue to do so as enterprises move to source cloud services from multiple providers in years to come, she says.

"The proliferation of public cloud has grown so quickly over the last 18 months, and we're starting to see those capabilities really accelerate," says Kearns.

"Amazon Web Services [AWS] has dominated for so long, but Azure and Google Cloud Platform have made such strides in the last few years, and when you factor in predictive analytics, native analytics and artificial

intelligence, it starts to become a really interesting promise that public cloud can offer enterprises."

In this e-guide

Next article

In this e-guide

Getting more CW+ exclusive content

As a CW+ member, you have access to TechTarget's entire portfolio of 120+ 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; Fotalia