ecosystm

APPLICATION MODERNISATION
with Containers and Kubernetes

November 2020

AUTHORED BY

DARIAN BIRD
Principal Advisor

Hewlett Packard Enterprise

Executive Summary

Application modernisation has become an essential mechanism to ensure organisations can rapidly launch new, innovative offerings and iteratively improve on established ones.

The arrival of disruptive born-in-the-cloud new entrants in most industries has forced incumbents to embark on their own transformation journeys. This phenomenon means even traditional products and services are now delivered to users with a digital experience. Moreover, the pace of change in all industries has accelerated. Application modernisation has become an essential mechanism to ensure organisations can rapidly launch new, innovative offerings and iteratively improve on established ones. A key aspect of this includes shifting to a microservices architecture underpinned by the adoption of containers and orchestration tools, such as Kubernetes. These technologies provide a range of benefits, including efficiency, agility, scalability, and availability.

This whitepaper provides an overview of the agility and cost benefits of containers and orchestration from the perspective of application modernisation with a microservices architecture. It also includes essential guidance on how to deliver value faster during the adoption process.

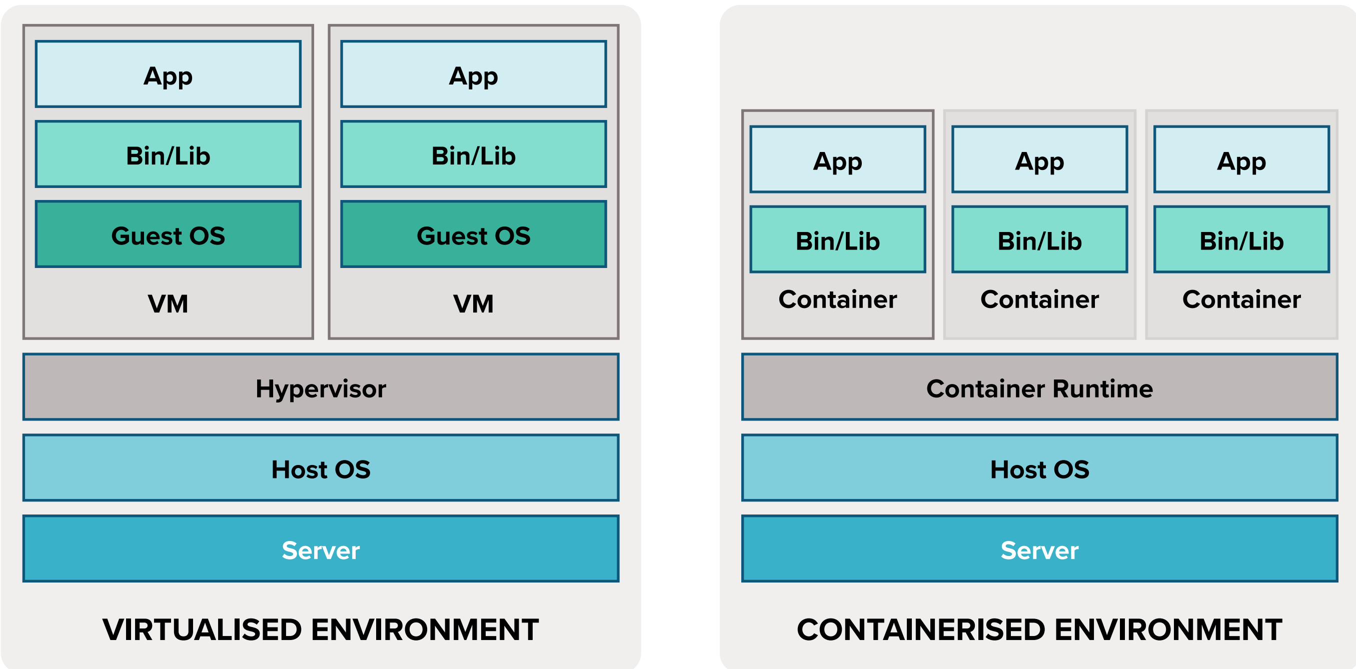
Application Modernisation with Containers and Kubernetes - Key Benefits

COST SAVINGS: EFFICIENCY WITH CONTAINERS

One of the defining characteristics of containers is that they are lightweight compared to virtual machines (VMs) resulting in significantly improved density.

Unlike VMs, containers share an operating system rather than requiring their own, reducing their footprint. The size of container images can therefore be measured in tens of MBs instead of the GBs commonly associated with VM images. This means more of the server's resources can be directed to running the application in the container and not the OS. Enhancing efficiency further, are the management capabilities of Kubernetes. Even when they are sitting idle, VMs may have resources dedicated to them in anticipation of periods of higher load. Kubernetes reduces the need for over-provisioning by scaling up the number of pods according to predefined policies. Its native load-balancing features also increases efficiency. These factors combined, mean a greater number of containers can be packed onto the same sized server than would be possible with virtualisation.

FIGURE 1:
Structure of Containers Compared to Virtual Machines



SOURCE:
Ecosystm, 2020

Another factor to consider when deciding between containerisation and virtualisation is a third option - whether to deploy containers on VMs or on bare metal. With VMs being ubiquitous - and until recently - the preferred means of deploying applications, many organisations building their first proof of concept (PoC) with containers, have opted to do so in a virtualised environment. This method allows organisations to leverage the experience they have with VMs while they familiarise themselves with containers. VMs also offer a greater level of isolation than containers because they do not share the host OS kernel.

AS ORGANISATIONS MOVE THROUGH THE PROCESS FROM POC TO APPLICATION IN PRODUCTION; SOME MAY PREFER TO ADOPT A BARE-METAL CONTAINER ENVIRONMENT FOR SEVERAL REASONS:

01

VM licences
Containers on bare metal eliminate the cost of VM licences, which can rapidly consume infrastructure budgets.

02

Density
The higher density associated with containers on bare metal reduces the cost of physical infrastructure required.

03

Performance
Some evidence points to higher performance of containers on bare metal, again reducing the resources required for the same number of containers.

ADOPTION OF THIRD-PARTY CONTAINERISED SOFTWARE

With a wide landscape of infrastructure and cloud providers developing their own flavour of Kubernetes, the availability of third-party software for containerised environments has blossomed.

Several marketplaces now host official distributions of software such as databases, monitoring, load balancing, development and testing tools, security and identity, machine learning, and collaboration. These marketplaces provide centralised billing and ensure DevOps teams have access to the latest versions of the most popular third-party tools. Some can be used to create customised internal portals with only authorised software while others can also act as a repository for developer teams to host their own containerised applications.

Many of the most popular software tools used in developing containerised applications are open source, reducing costs and increasing flexibility. Examples include Postgres (RDBMS), MySQL (RDBMS), MongoDB (NoSQL database), CouchDB (NoSQL database), Redis (in-memory database), Nginx (web server), Traefik (reverse proxy and load balancer), Jenkins (continuous integration), Prometheus (monitoring), Kong (API management), Sysdig (security), and Spinnaker (continuous delivery). Proprietary monitoring tools by Dynatrace and Datadog are also widely used.

MAKING SOFTWARE ENGINEERING MORE AGILE

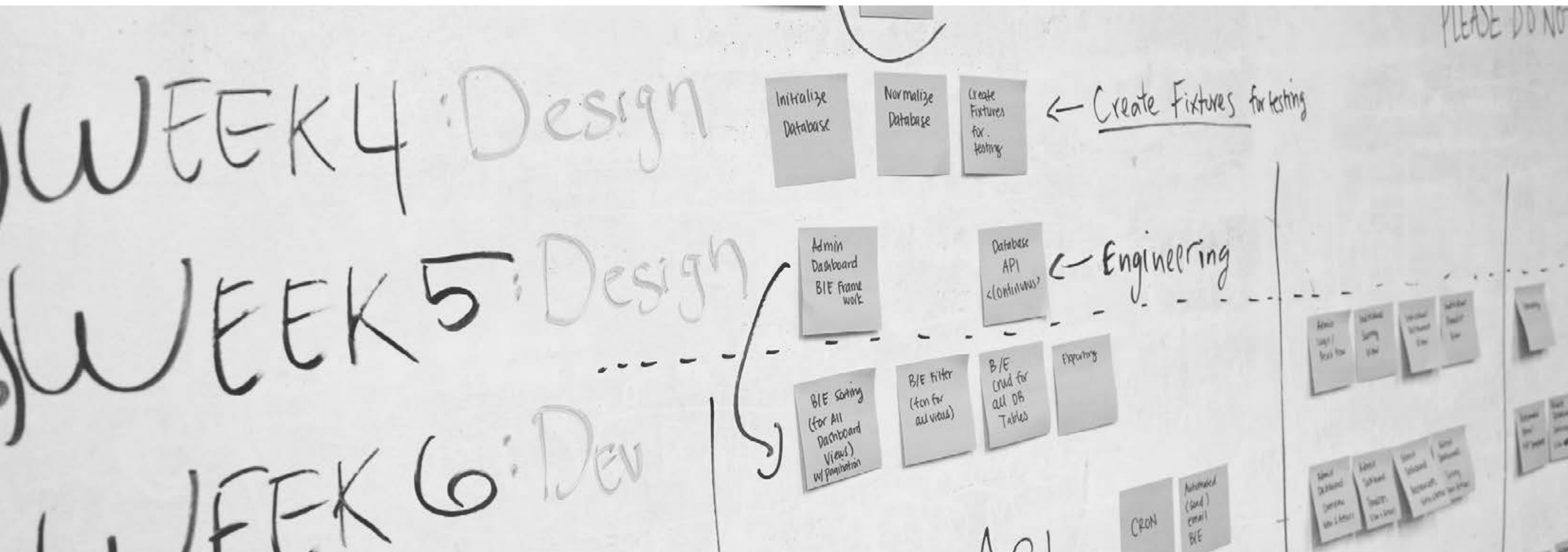
Architecting an application as a collection of microservices inside pods of containers enables smaller teams to concurrently develop, test, and deploy as quickly as needed without excessive communication or waiting.

A primary motivator for adopting a containerised environment is breaking down monolithic or virtualised applications into slimmer components that can be developed and deployed with greater agility. Architecting an application as a collection of microservices inside pods of containers enables smaller teams to concurrently develop, test, and deploy as quickly as needed without excessive communication or waiting. This shift to a microservices architecture benefits greatly from the lightweight nature of containers, which provides a consistent runtime environment and a high level of isolation.

Another aspect of microservices that is enabled by the smaller size of containers is the move towards DevOps with the continuous integration/continuous deployment (CI/CD) model. Developers are now able to focus on a single independent service running inside a container that can automatically be tested as part of a continuous integration process to ensure that it does not break the production environment. Following continuous delivery to a repository, continuous deployment then automates release into production according to the schedule set out by the business - whether this is weekly, daily, or instantly.

FIGURE 2:
CI/CD Pipeline

SOURCE:
Ecosystm, 2020



AGILITY: THE INFLUENCE OF CONTAINERS ON INNOVATION

Application modernisation and the underlying technologies of microservices, containers, and container orchestration not only have consequences for IT but also enable businesses to deliver innovative offerings to their customers, employees, and partners. Whether a business is an incumbent or a start-up, digital transformation is essential to avoid disruption. This has become particularly evident during recent turbulent times when business models changed in days rather than years.

KEY BUSINESS BENEFITS THAT STEM FROM ADOPTING CONTAINERS AND ORCHESTRATION INCLUDE:

01

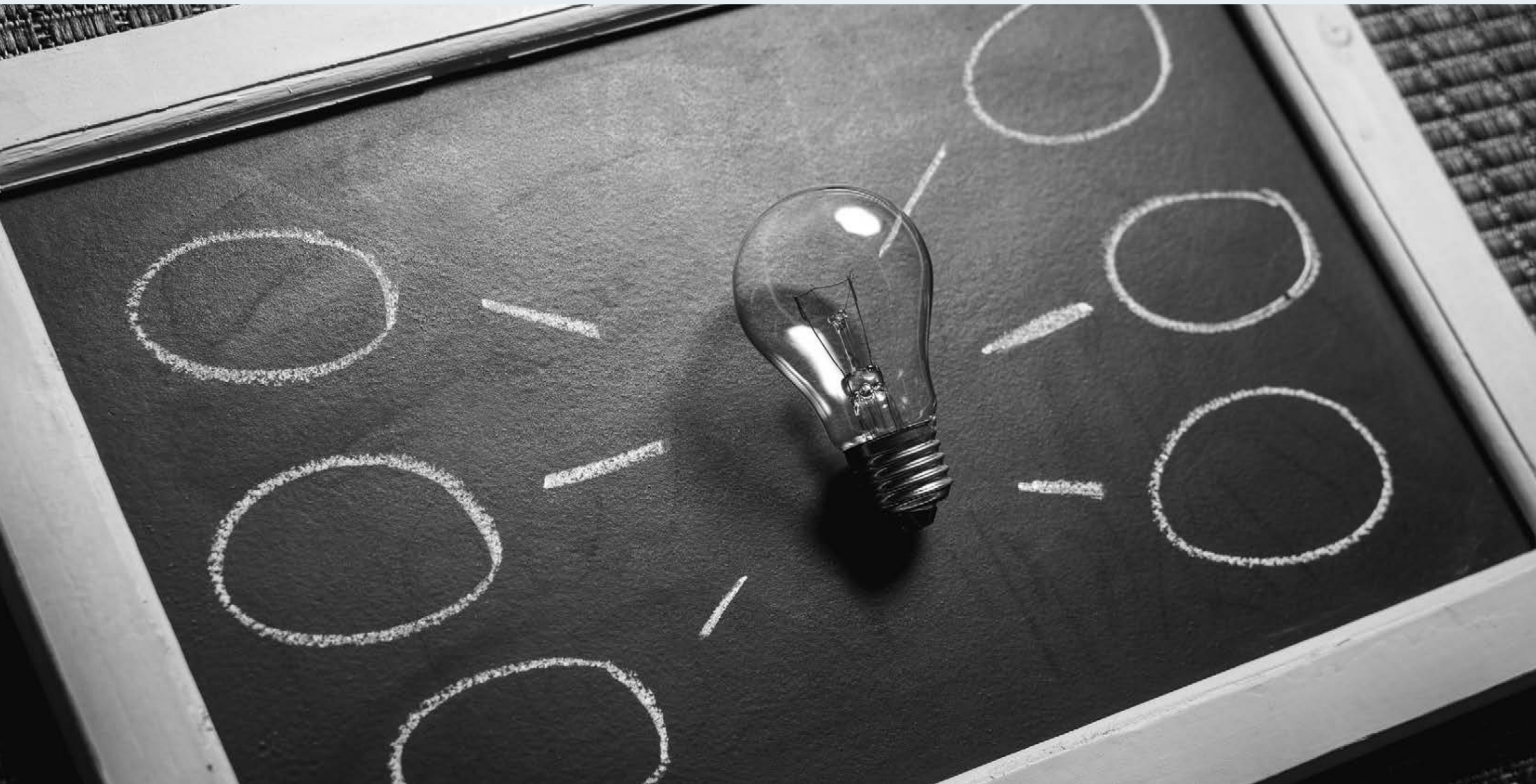
Speed and iteration
New services can be launched and iteratively improved upon at a quicker pace with less concern for the stability of already established applications.

02

Automation
Once Kubernetes best practices are adopted, Day 2 Operations can become automated, leaving the operations team to focus on building or improving other services.

03

Scalability and Availability
The scalability and self-healing features of Kubernetes allow businesses to develop offerings without the anxiety of degraded performance or outages during periods of peak demand.



How to Deliver Value Faster

CONTAINER ADOPTION CYCLE

Early container adoption was initiated by developers looking to take advantage of their compact size and suitability to a microservices architecture.

Operations teams are now driving the next phase of uptake, convinced of the benefits of containers and Kubernetes, such as scalability, density, portability, and availability. Most large organisations have already experimented with containers and have at least some containerised applications running in production. Making the transition to a containerised environment with an orchestration system is not an overnight process and will depend heavily on how advanced an organisation is with complementary technologies and techniques.

FACTORS THAT WILL HELP ORGANISATIONS DELIVER VALUE FASTER WITH CONTAINERS AND KUBERNETES INCLUDE:

01

Microservices roadmap

Develop a plan for containerising appropriate workloads first as part of microservices architecture. This should begin with business-led goals.

02

DevOps

Create a DevOps organisation to foster better, faster collaboration between developers and operations.

03

Automation

Intend to automate as much as possible to gain the full speed benefits of containers. This includes automation of CI/CD, security, and operations.

04

Skills

Upskill the operations team and form a centralised unit with deep Kubernetes knowledge to act as trouble shooters when ramping up the roll out.

05

Container management platform

While early adopters went the DIY route, opting for a commercially available platform will simplify the adoption of Kubernetes.

MICROSERVICES ROADMAP

Containerisation should be viewed as one method to be adopted as part of a larger process of application modernisation.

This process in turn should be initiated with business goals in mind. Goals might include faster time to market with new services, flexibility to adapt current services, business resiliency, reduced cost structure, or improved customer experience. When beginning with these goals, buy-in from stakeholders from beyond IT is necessary, from stakeholders including Security, Finance, Marketing, Operations, and Product Development.

When developing a roadmap to identify suitable workloads for containerisation, an early consideration is whether to lift and shift an existing application, refactor one, or build a new cloud-native one. Another determination is whether to begin with a project that will realise its value in the shortest timeframe possible or concentrate on a simple build to gain some early experience.

WHEN MAKING DECISIONS, THE ANSWERS TO THE FOLLOWING QUESTIONS SHOULD BE TAKEN INTO ACCOUNT:

- [ ] Is there already a containerised third-party option for the application under consideration?
- [ ] Is the application stateless or are there stateless components that could be containerised?
- [ ] Does the application require frequent updates?
- [ ] Does the application require scalability?
- [ ] Is there an application that is already developed using DevOps and CI/CD processes?
- [ ] Is the application large and able to be broken down into microservices?
- [ ] Are there components in the application that are reusable?

AUTOMATION

While shifting to a containerised environment including a microservices architecture, brings many benefits, it also presents new challenges.

Splitting a monolithic application into many individual services increases complexity, can result in cluster sprawl, and by design, quickens the pace at which developer and operations teams must work. These are not necessarily drawbacks, but the importance of automating application development, deployment, and Day 2 operations in order to succeed, must be emphasised.

KEY AREAS OF AUTOMATION SHOULD INCLUDE:

01

CI/CD Pipeline

The goal of automation in the CI/CD pipeline is to allow developers to have their code integrated into the main branch and deployed into production as quickly as possible without waiting for manual processes or creating conflicts with changes made by other developers. Examples include static code analysis, unit testing, end-to-end testing, performance tests, integration tests, build triggering, deployment triggering, and canary deployment.

02

Security

Many automated security functions should be built into each step of the application development and deployment process, e.g. container hardening during development, vulnerability scanning of images during integration, admission controls during deployment, and automated alerting and response during Day 2 operations.

03

Operations

Much of the automation in operations will come from Kubernetes and complementary tools, e.g., readiness and liveness probes, horizontal scaling, load balancing, logging, and storage orchestration. Additional tools, such as AIOps can provide noise reduction, root cause analysis, and proactive response.

UPSKILLING STAFF AND HR CHALLENGES

Adopting containers and Kubernetes requires important adjustments within the workforce in terms of skills and culture.

Developers have embraced agile methodology for some time now and the use of containers has matured significantly since Docker's original release in 2013. IT operations, however, has had less pressure to adopt Kubernetes on a large scale while containers were mostly used in limited trials. Now that containers are being widely deployed in production, there is a Kubernetes skills gap that needs to be addressed. A challenge to upskilling is that Kubernetes platform engineers require deep knowledge of container orchestration and at least a basic understanding of compute, storage, networking, and security. The complexity that this brings and the need for 24/7 support exacerbates this issue for any but the largest organisations. Upskilling will be a gradual process that begins by identifying appropriate workloads to deploy in containers with orchestration. Those that learn from this initial process will form the core team that helps to disseminate knowledge throughout the organisation.

To get the most out of cloud-native technologies, like microservices, containers, and Kubernetes, organisations must change to keep up with the increased pace of development.

DevOps principles to bring developer teams and operations closer together is the most effective approach to prevent developers from simply shipping containerised services without concern for how they will be deployed, managed, or scaled. Beyond organisational realignment of development and operations teams, cultural changes are also needed. A more dynamic culture that encourages faster innovation, flexibility, and empowerment of smaller teams should permeate across the organisation.

Parameters to Consider When Adopting Kubernetes for a Production-Ready Environment

CONTAINER ORCHESTRATION IN MULTI-CLOUD ENVIRONMENTS

A multi-cloud approach allows organisations to select the right cloud for the right workload at the right time.

Portability is a significant benefit of containerised applications, giving organisations the ability to build once and deploy across a range of environments without fear of lock in. A multi-cloud approach allows organisations to select the right cloud for the right workload at the right time. Key reasons for differentiating between on-prem, public cloud, or edge computing include data residency requirements and security, latency sensitivity, local processing needs, scalability, availability, and cost optimisation.

When selecting a container management platform, it is important to ensure the ability to manage clusters on each of the largest public cloud providers, namely Amazon Elastic Kubernetes Service (EKS), Azure Kubernetes Service (AKS), and Google Kubernetes Engine (GKE). Additionally, support for the organisation's preferred hypervisor should be included, with the major ones being VMware vSphere, Microsoft Hyper-V, and Red Hat Virtualization. The option for bare metal is also becoming increasingly important as organisations become more comfortable shifting away from VMs. Finally, container platform and third-party tool providers are developing novel approaches to tackle the challenge of orchestrating containers in edge computing environments.

PERSISTENT STORAGE

Containers were originally designed to be stateless, meaning that they would carry out a process but once that was ended or the container replaced, no data would be stored.

This ephemeral storage is suited to simple web services, which are often temporary in nature, but not to enterprise applications that require persistent storage. To enable persistent storage, Kubernetes provides mapping to volumes that reside outside of the container. Other situations arise however, where third-party tools are required, such as a service gateway or data fabric. One example is the need for sticky sessions where user authentication would be lost in a multi-page web application. Another is processing large volumes of video or sensor data that needs to be attached to a specific container or pod.

SECURITY

The increased complexity inherent in Kubernetes, the distributed nature of containers, and lateral (east-west) communication between pods all come with risks.

As with any new system, containerised applications managed by Kubernetes create the potential for additional vulnerabilities without appropriate security adjustments. The increased complexity inherent in Kubernetes, the distributed nature of containers, and lateral (east-west) communication between pods all come with risks. These factors can be magnified by the desire to develop and deploy as quickly as possible. As organisations get to grips with containers, misconfigurations represent the greatest security challenge. Furthermore, runtime security incidents and vulnerabilities need to be addressed.

METHODS TO REDUCE SECURITY RISKS IN CONTAINERS AND KUBERNETES INCLUDE:

01

Shift left

DevSecOps approach including vulnerability scanning earlier in the CI/CD pipeline.

02

Minimise privilege

Use Role-based Access Control to minimise the number of privileged containers.

03

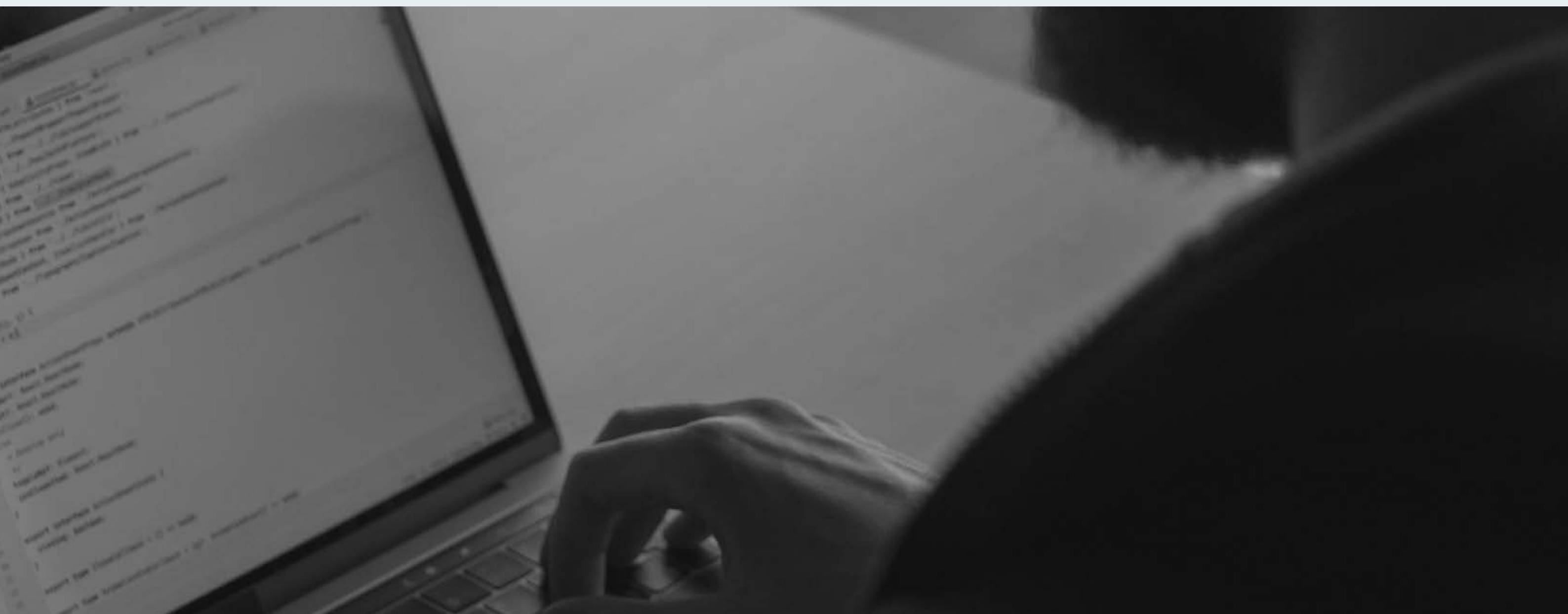
Micro segmentation

Reduce the scale of potential attacks with workload and network segmentation.

04

Upgrade often

As Docker and Kubernetes grow in popularity so do the number of known vulnerabilities.



Container Management Platforms: Toolset Features and Capabilities

NO ONE SIZE FITS ALL

Container management platforms are now available from a range of vendors, including cloud service providers [AWS, Google, Microsoft], niche Kubernetes specialists [Rancher (SUSE), Platform9, Mirantis, D2iQ, Diamanti], virtualisation vendors [Red Hat (IBM), VMware], and cloud infrastructure providers [HPE].

While most of these offerings are based on Kubernetes, each has additional functionality to cater to specific market segments. When evaluating these offerings, assess your organisation's IT systems - whether they are primarily on-prem, single cloud, or multi-cloud; how geographically dispersed they are; the level of complexity; and the state of automation. Factor in current skills and culture to decide if your organisation is ready to adopt containers alone or with a service provider. With these in mind, consider the suitability of the deployment model, e.g., DIY, public cloud, hybrid cloud, SaaS, or managed service. Also pay close attention to whether the platform satisfies the requirements of developers, operations, or both.

TOOLSET FEATURES AND CAPABILITIES

Besides the toolset features mentioned (software marketplace, automation, multi-cloud support, persistent storage, and security), here are some important capabilities to consider when selecting a vendor.

FEATURE	DETAILS
Ease of use	Kubernetes involves a steep learning curve, which should be at least somewhat mitigated by a simplified user experience. Dashboards to improve container visibility are essential.
True open source	Some proprietary Kubernetes distributions have modifications not supported by the community. True open-source platforms ensure users can troubleshoot without vendor support.
Service support	The openness of Kubernetes ensures self-service is possible for many small issues; however, 24/7 support should also be a consideration.
Multi-version support	Kubernetes has four major releases each year in addition to patch and security updates. Organisations quickly find they deploy multiple versions across departments and applications. Platforms should support multiple versions of Kubernetes.
High availability and self-healing	Kubernetes has self-healing at the application layer built in by monitoring pod health and redeploying in the case of faults. Some platforms go further to include high availability for the Kubernetes component layer.
Service mesh	One of the capabilities that is included in some platforms is a service mesh, such as Istio. The service mesh provides load balancing, traffic routing, cluster ingress and egress, and authentication to ensure secure service-to-service communication.

SOURCE:
Ecosystm, 2020

About the Author

DARIAN BIRD

Principal Advisor, Ecosystm

Darian helps businesses navigate the path towards digital transformation, providing insight into cloud, automation, cybersecurity, and outsourcing. He has spent two decades advising business leaders on using technology to enter new markets, improve client experience, and enhance service delivery.

Previously, Darian spent ten years at IBM, where he was a principal advisor for infrastructure services and hybrid cloud in Europe. Prior to this, he was a research manager at IDC, gaining emerging markets experience in Asia Pacific, Central Eastern Europe, Middle East, and Africa. In his final position, Darian headed up IDC's ANZ offshore research team based in Kuala Lumpur.

Originally from New Zealand, Darian is based in Prague, the Czech Republic. He holds a Bachelor of Business, majoring in marketing, from the University of Auckland. Outside of the office, Darian enjoys running up mountains, biking with his young daughters, and researching his family tree.

ABOUT ECOSYSTM

Ecosystm is a private equity backed Digital Research and Advisory Platform with global headquarters in Singapore.

As a global first, Ecosystm brings together tech buyers, tech vendors and analysts into one integrated platform to enable the best decision making in the evolving digital economy. The firm moves away from the highly inefficient business models of traditional research firms and instead focuses on research democratisation, with an emphasis on accessibility, transparency and autonomy.

Ecosystm's research originates from its custom designed "Peer2-Peer" platform, which allows Tech Buyers to benchmark their organisation in "real-time" against their industry or market. This bold new research paradigm enables Ecosystm to provide Tech Vendors access to ongoing and real-time Market Insights in an affordable "as-a-Service" subscription model.

ABOUT HEWLETT PACKARD ENTERPRISE

Hewlett Packard Enterprise are in the acceleration business.

We help local customers use technology to slash the time to turn ideas into value. In turn, they transform industries, markets and lives across the country.

Some of our customers run traditional IT environments. Most are transitioning to a secure, cloud-enabled, mobile-friendly infrastructure. Many rely on a combination of both. Wherever they are in that journey, we provide the technology and solutions to help them succeed.

We make IT environments more efficient, productive and secure, enabling fast, flexible responses to a rapidly changing competitive landscape. We enable some of New Zealand's largest organisations to act quickly on ideas by delivering infrastructure that can be easily composed and recomposed to meet shifting demands, so they can lead in today's marketplace of disruptive innovation.

This white paper is sponsored by Hewlett Packard Enterprise. It is based on the analyst's subject matter expertise on the area of coverage in addition to specific research based on interactions with technology buyers from multiple industries and technology vendors, industry events, and secondary research.