

# Emerging and future trends

*by* Gareth Michael Cox

---

|                |                                                                                         |                 |       |
|----------------|-----------------------------------------------------------------------------------------|-----------------|-------|
| FILE           | 9121_GARETH_MICHAEL_COX_EMERGING_AND_FUTURE_TRENDS_226<br>879_1313720420.DOCX (366.16K) |                 |       |
| TIME SUBMITTED | 20-FEB-2020 12:52PM (UTC+0000)                                                          | WORD COUNT      | 2770  |
| SUBMISSION ID  | 120373036                                                                               | CHARACTER COUNT | 15252 |



PRIFYSGOL CYMRU  
**Y Drindod Dewi Sant**  
UNIVERSITY OF WALES  
**Trinity Saint David**  
SWANSEA - ABERTAWE

***MSc Computer Networks & Cyber Security***  
***School of Applied Computing***

**Emerging & Future Trends ACCA7016**

---

# Cloud Services and Containerization

By

**Gareth Cox**

**P014666**

February 2020

---

For

**Dr Mike Dacey & Dr Tim Bashford**

## **Contents**

|                                               |    |
|-----------------------------------------------|----|
| Contents.....                                 | 1  |
| Figures.....                                  | 2  |
| Tables .....                                  | 2  |
| Question 1.....                               | 3  |
| Cloud Computing Services .....                | 3  |
| What is the Cloud.....                        | 3  |
| What is a Service .....                       | 3  |
| Cloud Services .....                          | 4  |
| What is a Container.....                      | 6  |
| Hypervisor vs Container .....                 | 6  |
| Docker.....                                   | 8  |
| Pros .....                                    | 9  |
| Cons .....                                    | 10 |
| Question 2.....                               | 11 |
| Infrastructure as Code (IaC) .....            | 11 |
| Standard Operating Environments (SOE's) ..... | 11 |
| Question 3.....                               | 12 |
| Orchestration management.....                 | 12 |
| Kubernetes .....                              | 12 |
| Scenario.....                                 | 12 |
| References .....                              | 13 |

## **Figures**

|                                                                                 |   |
|---------------------------------------------------------------------------------|---|
| Figure 1- Infrastructure - Platform - Software (as a Service) (IBM, 2018) ..... | 5 |
| Figure 2 - VM vs Container Virtualization (Hewlett Packard, 2018) .....         | 6 |
| Figure 3 - Simplified Comparison of Hypervisor Vs Container.....                | 7 |
| Figure 6- DevOps Build-Ship-Run Cycle (Microsoft, 2019).....                    | 9 |

## **Tables**

|                                  |   |
|----------------------------------|---|
| Table 1- VM Vs Container .....   | 7 |
| Table 2 - Virtual Machines ..... | 7 |
| Table 3 – Containers .....       | 7 |

## **Question 1**

### Cloud Computing Services

#### What is the Cloud

At a very basic level the cloud is the internet on some one else computer, however an official definition by National institute of standards and technology define the cloud as,

*'cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction'* (NIST 800 - 145, 2011)

So, in essence it is a means, for a business or individual to use the computing power of a 3<sup>rd</sup> parties' equipment to carry out the tasks that traditionally required your own equipment to do, this has led to the Model 'As a Service' being created and deployed.

These are typically deployed across three variants of the Cloud (IBM, 2018) (IEEE, 2016) (NIST 800 - 145, 2011) (Mohan, 2017)

- Public
- Private
- Hybrid

#### What is a Service

Companies who own the equipment to market and sell (rent) its equipment's use to a business or individual, as and when it is required, as a service to that customer allowing the agility of availability for scalable use on demand to the requirements that is secure while being accessible anywhere and anytime without the overhead costs of purchasing and maintaining the equipment (Castro-Leon, 2016).

## Cloud Services

There are three main types of cloud services (known as Cloud service models) (IBM, 2018) (IEEE, 2016) (NIST 800 - 145, 2011) (Deloitte, 2017) these are: -

- *Infrastructure as a service (IaaS)*
  - *The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls). (NIST 800 - 145, 2011)*

Infrastructure as a Service is basically the rental of computing resources over the cloud. The 3<sup>rd</sup> party can rent the entire range of computing infrastructures such as storage, servers, networking hardware where the 3<sup>rd</sup> party is then responsible for the maintenance and support, of this equipment, Microsoft Azure is one of the leading providers for this service.

- *Platform as a service (PaaS)*
  - *The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. (NIST 800 - 145, 2011)*

Platform as a Service is essentially a cloud base which you can use as a platform to develop, and test different applications for your business. It can and does simplify the process of software development. The resources rented in the form of servers, storage and networking are normally then manageable by the company.

- *Software as a Service (SaaS)*
  - *The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings (NIST 800 - 145, 2011)*

Software as a Service allows a company or individual to access cloud-based applications. A 3<sup>rd</sup> party controls the equipment, that runs the applications, this mean there is no need for installing applications on each of your computers with the company and the equipment's and the maintenance and support is the responsibility of the 3<sup>rd</sup> party. An example of this software is Microsoft Office 365

The diagram below shows a visual representation for this.

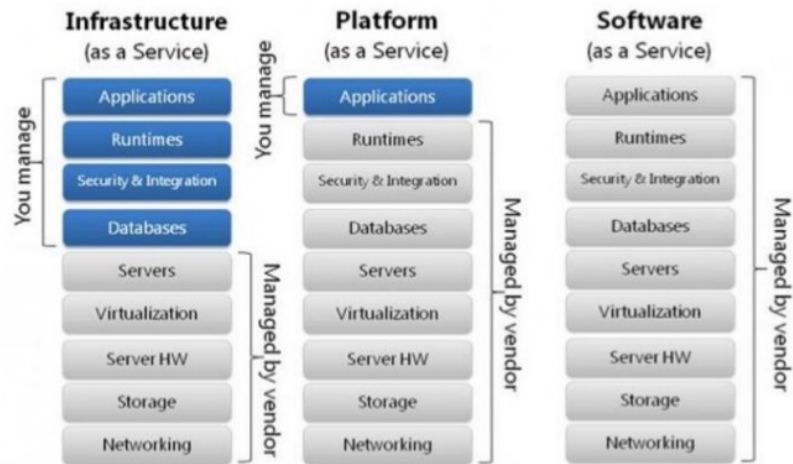


Figure 1- Infrastructure - Platform - Software (as a Service) (IBM, 2018)

## What is a Container

Containers are used to as environments to run and develop applications. They contain only the parts of the system called Images that are needed to run that application. Containers function similarly to Virtual Machines. However, because they only include the necessary libraries and settings to run the applications, they are used to isolate applications into extremely light weight chunks which means they're a fraction of the size of a VM, using fewer resources and spin up in a fraction of the time.

Because of this a computer would treat a contain more like another program to run rather than the resource heavy virtual machines or hyper visors.

## Hypervisor vs Container

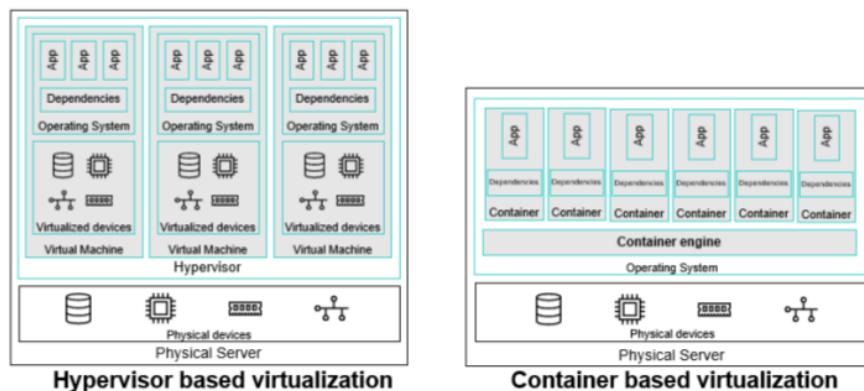


Figure 2 - VM vs Container Virtualization (Hewlett Packard, 2018)

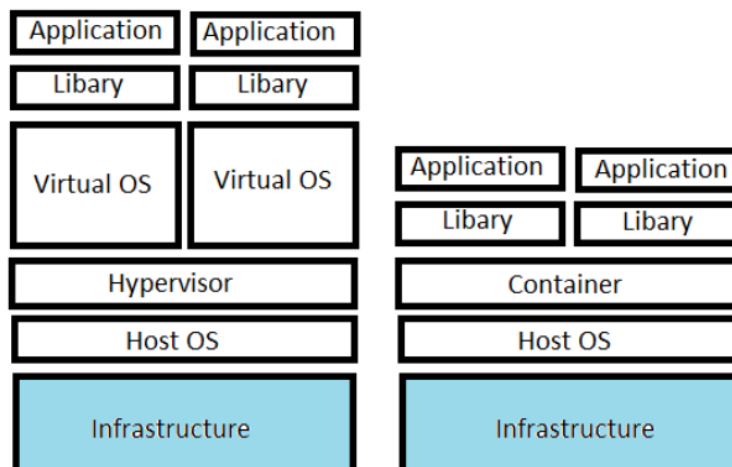


Figure 3 - Simplified Comparison of Hypervisor Vs Container

| Hypervisor                                                              | Container                                                              |
|-------------------------------------------------------------------------|------------------------------------------------------------------------|
| Host OS installed on infrastructure                                     | Host OS installed on Infrastructure                                    |
| Hypervisor runs on OS                                                   | Containers run on Host OS shared kernel                                |
| Hypervisor creates and manages Virtual OS's                             | Containers contains libraries and dependencies for applications to run |
| Each Virtual OS contains libraries and dependencies to run applications |                                                                        |
|                                                                         |                                                                        |

Table 1- VM Vs Container

| Virtual Machine ( Hypervisor)                              |                                                      |
|------------------------------------------------------------|------------------------------------------------------|
| Pros                                                       | Cons                                                 |
| Multiple Virtual Operating systems can be installed        | Host and virtual machine heavily entangled           |
| Virtual machines are isolated from each other and the Host | Large files size                                     |
|                                                            | Requires large resources                             |
|                                                            | Applications and operating system share file systems |

Table 2 - Virtual Machines

| Container                                                                          |                                                                               |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Pros                                                                               | Cons                                                                          |
| Runs as if program on host                                                         | Runs on Hosts Kernel                                                          |
| Small file size                                                                    | Tied to host kernel                                                           |
| Containers are not isolated from each other or Host providing interconnectivity    | Containers are not isolated from each other or Host creating reduced Security |
| Requires minimum resources                                                         |                                                                               |
| Each container can be its own application and have its own library and dependences |                                                                               |

Table 3 – Containers

So from this it's able to conclude that hypervisor provide more the Infrastructure as a Service with allocation and hardware management, and Containers more towards providing a Platform as a Service by providing tools to support the required applications.

## Docker

Docker was launched in 2011 by Solomon Hykes and Sebastien Pahl it is currently the market leader for creating and deploying containers.

Docker packages the specific applications dependencies into a container known as an image, often found on an online repository called dockerhub (Zhanibek, 2017).

Dockers function is to make it easier to create and connect containers as we have seen make it easier to develop and run applications, it is an open source and allows for the creation of containers in the following environments: -

- Docker Datacenter
- Docker Linux
- Docker on Windows

## Pros

### Portability

The container image can run on any host, this removes the chance of variance in the stages of production as not only is the environment can be version controlled and can be created to mirror the exact production environment, as shown below in the diagram

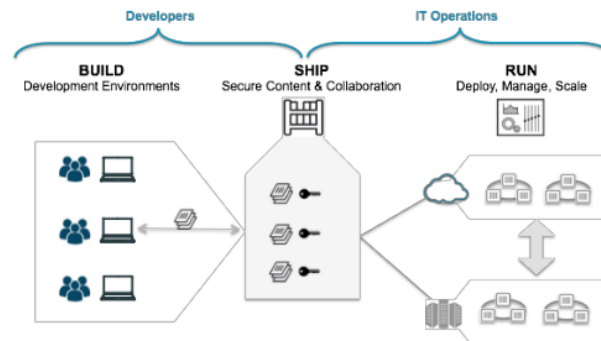


Figure 4 - DevOps Build-Ship-Run Cycle (Microsoft, 2019)

### Scalability

Containers are considered very lightweight and require low resources, when images are created they are normally very small, this allows them to have very short life cycles.

They can be started and stopped in seconds, compared to the more traditional approach of using VMs which can take quite a while to spin up.

This function has allowed for the possibility to treat a container as a disposable product that can be started and stopped or even discarded and created at will. Allowing a systems resources to be used only when needed. This means those resources could then be used to fulfill another role allowing greater flexibility in the system.

### Equipment utilization

Utilizing the shared operating system (Kernel) means the host's resources are dedicated to a single host and not shared between the host and any number of virtual machines, this means Individual server utilization and consequently scaled utilization is much efficient.

- Traditional datacenters use is on average only 10% utilization
- Virtual Machines improve datacenter use to 50%
- Containers can improve datacenter use to 90%

(IBM Thomas J. Watson Research,, 2018)

## Cons

Containers although still consider a new relatively new tool in the cloud services world bring many benefits but must be balanced against the increased complexity of a running and scaling applications. (Ruan, 2018) (Higgins, 2019) These create new problems specifically

- Complexity and scale
  - How to manage the container scaling
  - How to manage the workload of containers once scaled
  - How do my containers find each other
- Managing the Data
  - Containers are short lived so what happens to data when container is shut down
  - How is the data accessible once containers are scaled
- Managing the Micro services
  - Requires detailed pre planning for conditions and set up
  - Automation in place
- Security
  - containers are interconnected
  - containers are tied to host operating system

There are uses for both hypervisors and containers, both independently and in combination – The reason why containers have a performance advantage over virtual machines is as stated earlier in this assignment from the computers or hosts perspective a container is just another program running on the system, This is also the reasons for some of the concerns most notably the security is limited, The container is also completely reliant on the to the host operating system limiting its adaptability. The networking containers is also becomes more complex the greater the scale limiting the speed of scalability. (Roberto Morabito, 2015)

The cloud is fundamentally about supplying the customer the ability to always have the availability of data when and where it is needed, although traditionally the services have been supplied in a monolithic fashion, containerization on the other hand fit perfectly into supporting the newer models of the micro- service as it allows for the rapid deployment of application specific platforms, and reducing the cost of resources required to support the service offered, the model of now offering this in a package known as “Container as a service” or CaaS allows a customer to purchase only what is required and when it is required keeping overheads lower and allowing more flexibility in their allocation of resources as the application environments, scaling / descaling management and monitoring the health (Maintenance) of containers is all taken care of by the provider.

## Question 2

### Infrastructure as Code (IaC)

Infrastructure as code is not the replacement of physical hardware by some clever use of coding as the name implies (Rahman, 2019), but rather the allocation and use of the hardware which is often configured as a virtual machine or as we have seen in this assignment containerized to create an environment that is then through the use of a file, this allows automation for the replication of the allocation of the physical hardware's resource (Guckenheimer, 2017) often housed in large data centres without the need for constant configuration or interaction with manual process (Morris, 2016).

### Standard Operating Environments (SOE's)

Standardization is a mindset, used throughout manufacturing and engineering it has been adopted by DevOps and has become the base of what is known as 'Standard operating environments' in an ever-expanding cloud world, a service provider will potentially need to manage thousands of servers. Deploying all these manually or even semi-manually is expensive and prone to human errors.

Without standardization, IT teams have historically had to create custom scripts to deploy and maintain these platforms.

As seen in the manufacturing industry variation due to Inconsistency can and does cost time and money and the continued maintenance of these systems is complex. Requiring specialized personal to support them.

SOEs are its form of a standardization model to create predictable, repeatable, reliable systems. By employing automation, systems can be deployed faster standardization lets the organization:-

- Remove the variance in systems by automating manual tasks
- Manage the system from a central location.
- Reduce the time it takes to install, upgrade, and patch software.
- Improve security



## Question 3

### Orchestration management

Because containers are so rapidly deployable and disposable often set up so there is one container image for every application (micro services) then the amount of containers in use when operations need to scale can grow exponentially, therefore managing these containers is just as vital as managing deployment of the application. (Higgins, 2019) This process is known as Orchestration, it is “The automated configuration, management, and coordination of computer systems, applications, and services. Orchestration helps IT to more easily manage complex tasks and workflows. IT teams must manage many servers and applications, but doing so manually isn't a scalable strategy”. (RedHat, 2019)

### Kubernetes

**Kubernetes** is an open source container orchestration management software originally created by Google, it uses a master / slave architecture, it has the ability to automate the deployment of containers with load balancing and service discovery, **unlike** docker it also manages the networking of the containers created known as nodes allowing for the simplification of scaling, but it is notoriously difficult to master,  7

### Scenario

Similar to when a spike in power consumption is required famously the break of a major television soap when customers put the kettle on, the grid must pre-empt this spike however a power station cannot be just turned on and off at will they require a start-up and shut down similarly a virtual machine also requires time to spin up this is where containers really shine as they can be scaled up and down on demand although it is services they provide, the principle is the same.  8

One such occasion for this could be a major shopping event such as black Friday where retailers predict a massive increase in web traffic but are unable to predict exactly the services required they are able to forecast it a larger retailer would absorb the cost of excess capacity but these costs could cripple a smaller retailer, this is where the ability to turn on or off services at will ensures the retailer does not have to purchase large amounts of potential over capacity to ensure they are able to meet demand but instead able to purchase the services as and when they are required to ensure uninterrupted operations.

## References

- Castro-Leon, E., 2016. *Cloud as a Service, Understanding the Service innovation Ecosystem*. s.l.:Apress.
- Deloitte, 2017. *Everything-as-a-service: Modernizing the core through a services lens*, s.l.: Deloitte.
- Guckenheimer, S., 2017. *What is Infrastructure as Code?*. [Online]  
Available at: <https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code>  
[Accessed 2020].
- Hewlett Packard, 2018. *Understanding Containers*, s.l.: Hewlett Packard Enterprise.
- Higgins, J., 2019. *Orchestrating Docker Containers in the HPC enviromnt*. s.l.:The University of Huddersfield.
- IBM Thomas J. Watson Research,, 2018. *A Comparative Study of Containers and Virtual Machines in Big Data Environment*, s.l.: s.n.
- IBM, 2018. *IaaS, PaaS and SaaS – IBM Cloud service models*, s.l.: IBM.
- IEEE, 2016. *Cloud computing service models: A comparative study*, s.l.: IEEE.
- Microsoft, 2019. *Containers as the foundation for DevOps collaboration*. [Online]  
Available at: <https://docs.microsoft.com/en-us/dotnet/architecture/containerized-lifecycle/docker-application-lifecycle/containers-foundation-for-devops-collaboration>  
[Accessed 2020].
- Mohan, L., 2017. A Comparative Study on SaaS, PaaS and IaaS Cloud Delivery Models in cloud computing. *International Journal on Emerging Technologies*, Issue 0975-8364 , pp. 158 -160.
- Morris, K., 2016. *Infrastructure as Code Managing Servers in the Cloud*. s.l.:O'Reilly.
- NIST 800 - 145, 2011. *The NIST definition of Cloud Computing*, s.l.: NIST (SP).
- Rahman, A., 2019. A systematic mapping study of infrastructure as code research. *Information and Software technology*, Volume 108, pp. 65 - 77.
- RedHat, 2019. *The Automated Enterprise, Build an automation foundation for digital business operations*. s.l.:RedHat.
- Roberto Morabito, J. K. ., K., 2015. *Hypervisors vs. Lightweight Virtualization: A Performance Comparison* , s.l.: IEEE.
- Ruan, B., 2018. A Performance Study of Containers in Cloud Environment. *Services Computing Technology and Systems*.
- Zhanibek, K., 2017. A performance comparison of container-based technologies for the Cloud. *Future generation computing systems*, Volume 68, pp. 175-182.



# Emerging and future trends

---

## GRADEMARK REPORT

---

### FINAL GRADE

52/100

### GENERAL COMMENTS

#### Instructor

PaaS (50) 35

IaC (25) 13

K8s (25) 10

#### Comments

=====

More references required.

PaaS and CaaS discussion lacked a bit of analysis. More analysis should have been given of the

benefits and drawbacks. e.g. "How to manage the container scaling", what does this mean?

IaC discussion is very general and neither effectively defines IaC nor evaluates the benefits or drawbacks of the approach.

K8s is very poorly described and the scenario provided needed to explain how this technology specifically solved specific related to it.

PAGE 3

---

PAGE 4

---

PAGE 5

---

PAGE 6

---

PAGE 7

---

PAGE 8

---

PAGE 9

---

PAGE 10

---



### **Comment 1**

reference



### **Comment 2**

reference

PAGE 11

---



### **Comment 3**

references



### **Comment 4**

reference

PAGE 12

---



### **Comment 5**

referencing

PAGE 13

---



### **Comment 6**

What is Kubernetes?



## Comment 7

reference



## Comment 8

reference

PAGE 14

---

PAGE 15

---