

CLOUD-BASED WEB SERVICES

Adam Vimpany
ACCA6024 1810613

TABLE OF CONTENTS

Introduction.....	2
Cloud computing	2
Benefits and drawbacks.....	3
Node.js and Docker	4
Evaluation of node.js application development.....	5
Conclusion.....	5
Bibliography.....	6
Appendix 1: Video link.....	8

INTRODUCTION

This report investigates the benefits and drawbacks of cloud computing and the technologies used to develop and support cloud services.

CLOUD COMPUTING

Cloud computing is where a shared pool of computing resources, such as data storage, servers, software and other services, are accessed remotely rather than being controlled by a local server or computer [1]. Cloud computing offers users the ability to access the infrastructure or resources they require rapidly and flexibly, over the internet. Despite being a relatively new technology, cloud computing has been widely adopted by both organisations and individuals [2]. There are many different types and models of cloud computing architecture, and many different services offered, and the most appropriate solution for a consumer will vary greatly based on their requirements.

There are four different deployment models of cloud services. The first is the public cloud, where a third-party owns and operates a cloud service, and opens it to the public for use [3]. The third-party owns and maintains all hardware and software associated with the service, while any user who wishes to use their service can pay to access them over the internet [2]. The second is a private cloud, where the resources are owned and controlled by a single organisation, and the resources are secured to ensure only users authorised by that organisation can access them [3]. The organisation maintains a high level of control over their resources without needing these resources to be on-premises [4]. The cloud services may be hosted by the organisation itself, or a third party [2]. The third deployment is the community cloud, which falls somewhere between the private and public clouds. Multiple consumers may use the cloud service, but the service is specific to a community of organisations or users that share common requirements [2]. The final type is the hybrid cloud, where multiple clouds of the above types co-exist. For example, a large organisation may have their own private cloud and datacentre to handle their regulated or sensitive data [3], but also use a public cloud for other tasks and thereby flexibly outsource their computing capacity.

Cloud computing can also be broken down into the different types of cloud service models that have evolved over the years, including infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), serverless and software-as-a-service (SaaS) [1]. These each have different applications, benefits and drawbacks.

IaaS cloud providers supply the consumer with networks, storage, virtual machines and other computing resources on-demand [3]. The consumer can run applications using the infrastructure, including operating systems and databases, often including the ability to host firewalls and configure some networking components or servers, but the cloud vendor owns and maintains the underlying hardware and software [2]. The vendor will own and maintain the necessary servers, power, storage systems and recovery data, networking components and security systems [2]. It is the most basic form of cloud computing, and other services have evolved from it. An example of IaaS is Amazon WebService's (AWS) Amazon Elastic Compute Cloud (Amazon EC2), where users can select from different processor, storage and networking models to meet their needs and budget [5].

PaaS provides the consumer with an environment with programming languages, libraries and tools that are supported and maintained by the cloud service provider using the vendors infrastructure, which the consumer can use to develop, modify and host their own software or applications [3]. The vendor will manage all ongoing maintenance of the environment and the resources required to use it. An example is the Google App Engine which allows a user to create applications using a variety of different programming languages, such as Python, Java or Node.js, and built-in APIs, such as Blobstore and GAE Cloud Storage. The engine also offers debugging, error reporting and testing functions [6].

Serverless architecture overlaps with PaaS. It too allows a user to develop their own applications using the vendor's development environment, with no need to manage servers or infrastructure. It launches consumer applications as needed using the cloud servers, without the need for the consumer to virtualise the servers themselves as they would using IaaS [1]. Where PaaS that are not serverless differ from serverless architectures is that the applications created using serverless computing can scale automatically on-demand [7]. Functions that are not currently in use by applications hosted on Serverless cloud environments will be suspended completely and automatically, and new instances can be started and scaled up rapidly as required [7]. This is in contrast with those hosted on PaaS,

which are scalable, but cannot scale up and down as rapidly [8]. Pricing usually differs between PaaS and serverless: with serverless the user often only pays for the exact resources used on a pay-as-you-go basis (i.e. if no resources are used, then the user does not pay), while PaaS often has a base price in addition to usage costs because there are background resources constantly in use [8]. AWS Aurora Serverless v2 is an example of a PaaS that is not serverless, while AWS Lambda is an example of serverless architecture [8]. Google App Engine is an example of both PaaS and serverless cloud computing services [6].

SaaS allows a consumer to run the service provider's software applications over the internet. The applications are managed and hosted by the service provider [3]. Common examples include web-based email services such as Outlook or Gmail, where both the email software and messages are hosted and stored on the vendor infrastructure, which the user accesses through a webpage [9]. Some SaaS are free to use for personal use (such as the previously mentioned web-based email services) but there are many SaaS applications that are accessed through a subscription model, particularly those that are more complex or specialised, or those used by organisations [9].

There are many other computing tasks that are now offered through the cloud "as a service", for example, Database-as-a-Service (DBaaS) [10] or Testing-as-a-Service (TaaS) [3], and it is not practical to list them all here.

BENEFITS AND DRAWBACKS

One of the prime benefits of cloud computing is that it is quick and easy to set-up [2]. It can remove the need for an organisation to set up their own dedicated servers and networks, with all the associated time, knowledge and resources required to do so. This can be a significant barrier in the time-to-market for a product or service [2], especially for small to medium sized organisations [11].

Using the cloud removes the need for a company to have their own infrastructure that is future-proofed for the expansion of the company, which can be costly to purchase up-front, deploy and then maintain over time [12]. It may also impede future growth if insufficient resources were used initially, but this must be balanced against the risk of increased overheads and wasting resources if they are under-utilised [2]. Instead, cloud computing can allow the organisation to use and pay for the exact resources they need, when they need them, with the option to increase or decrease flexibly over time [1]. This is beneficial, as even if demand increases unexpectedly, the organisation will be able to meet this demand automatically and not be hindered by their own infrastructure. The third-party service will handle all scaling automatically [9]. In fact, the UK Government Central Digital and Data Office guidance states that all central government and wider public sector organisations should consider cloud solutions first before other options due to the quality, effectiveness and value for money these solutions offer [13].

The need for manpower and expertise to maintain the resources is reduced, as it is outsourced to the third-party [12]. For small organisations, reduced operational burden is an important benefit, as they will not need staff with specialist enterprise networking, security and development knowledge, as well as physical hardware which can be expensive and a barrier to entering the market. Most public cloud services also offer 24/7 technical support [3]. In addition, some organisations may have physical space limitations to dedicate to infrastructure, and cloud services are not limited by this [3].

Cloud service providers can provide a very cost-effective option to both large and small organisations. By offering services to many users, they have the resources to improve the quality of the service over that which the users would have themselves. The providers can focus on improving, maintaining and securing their infrastructure in a way that would be unfeasible for smaller users, and can benefit from economies of scale [14]. This means that their prices are often very competitive. For example, Google App Engine is free to use if usage remains under a certain level for resources such as CPU, storage and API calls [6]. The third-party cloud service provider will have a transparent model of metering use (for example, storage or bandwidth use) to charge for use [3].

There are many specialist services available, from IaaS to SaaS which allows users a lot of choice and the ability to find the exact service that will meet their needs. For SaaS in particular, sophisticated applications that would be cost and resource-prohibitive for small organisations to develop themselves, have become accessible [9]. The ability for SaaS to be accessed through the web browser means that there is no need to roll out software across an enterprise, managing installations and updates: staff simply access the software as and when they need to, even remotely, with no additional resources needed.

Cloud service providers are usually responsible for backing up client data, which removes another operational burden from the user. Cloud services also provide security, and some specialist services may meet regulatory requirements for certain industries where sensitive data must be protected [2].

One downside of using the public cloud, however, is that a third-party is responsible for the user's data, and the implications of this must be considered by the user. Different cloud service providers will have different privacy policies, which may impact a service user: data may be collected and accessed by the third-party, which may not be acceptable to the user [15]. Even if the company has acceptable privacy agreements, there is a risk that they will not abide by them, or that the data stored by them is breached by a malicious actor. A well-known example of this risk would be the Cambridge Analytica scandal, where the data of 87 million Facebook users was used without permission [15]. Cloud service providers who provide services to EU-based users will need to comply with the General Data Protection Regulation [15], but some industries will have to abide by more stringent regulations, for example if they handle or store sensitive data, and this may limit choice of provider. Where the cloud service provider data is stored physically may also have an impact: some countries require that data about their citizens must be stored only in that country, and there is additional legal complexity in keeping data overseas [15].

Although using the cloud for computing needs can be convenient and economical, there is a risk that it can lead to having a single point of failure with serious real-world knock-on effects when something goes wrong. In February 2022, for example, British Airways online services became inaccessible, which meant that customers could not make bookings or check-in and staff could not access crucial systems, resulting in hundreds of flights being cancelled [16]. Another concern is that users may find themselves "locked in" to a service, as there is little standardisation between services, and therefore little interoperability between systems, resulting in prohibitive costs and resources to change provider [11]. There is added difficulty because many applications interact using messages, which also require contextual interoperability which is difficult to transfer [17].

NODE.JS AND DOCKER

Virtualisation and containerisation are both methods of getting multiple applications to run on a single host machine and are both commonly used in cloud computing.

Virtualisation uses a hypervisor to create virtual machines (VM) to set up a software infrastructure. VMs are a sandbox that simulates a physical computer environment. Multiple VMs can run on a single host, sharing resources such as CPU and RAM, as allocated by the hypervisor [18]. They are commonly used because they allow the availability of all operating system (OS) resources for applications and have management tools to allow all virtual machines to be managed from one location, which therefore centralises the workload [18]. They also reduce overheads and increase efficiency [18]. Common VM providers include Microsoft Hyper-V and VirtualBox. However, virtualisation is resource-intensive, as it requires significant RAM and CPU resources to run multiple OS [19]. There can also be compatibility problems when running older or specialised software applications on a VM, and configuration can be difficult. However, the opposite also holds true: a VM can be used to run obsolete or legacy software or hardware which is no longer supported by modern systems [20]. Virtualisation also allows for some resource scaling, which is useful in a cloud infrastructure, either by deploying more virtual machines (horizontal scaling) or by allocating more resources (vertical scaling) [21]. However, both scaling techniques have a detrimental impact on latency and responsiveness [21]. In addition, there are challenges associated with using VMs in a cloud environment because of the added complexity inherent to it [22].

The leading alternative to using VMs for virtualisation to deploy applications over the cloud is the use of containers [21]. Containers, like VMs, share physical resources, but unlike VMs also share the OS and supporting libraries [21]. The benefit of this is reduced resource costs, as there is only one non-isolated instance of the OS and not multiple [22]. Code, dependencies and the OS are all packaged into a self-contained package, so that the container can run in the same way on many different platforms [22]. The most popular method of containerising applications is through the use of Docker [21]. Containerisation is often used in conjunction with microservices, where a larger, more complex application is broken down into smaller, individual, linked components which can be updated independently. Node.js is a development environment that has increased in popularity due to its microservices architecture and ability to run multiple tasks concurrently without multithreading. It is also able to push resource intensive tasks to run concurrently on worker threads, and so it has enabled the development of applications that take advantage of this efficiency by using real-time functions

[23]. It is commonly used in conjunction with Docker to containerise the completed microservice-based application.

Cloud containers offer greater flexibility and scalability over traditional hypervisor VMs [24], as well as improving ease of management [22]. The ease of management is increased when used in conjunction with microservice-based applications, because they do not operate as a monolith as they do on VMware, but instead have lightweight, interconnected services which can be modified independently, and therefore failure of one component does not prevent the functionality of other components [21] [22]. This also means that microservice-based applications are very reliable, as failure of a single component will not usually result in an outage [23].

This lighter-weight approach is hugely beneficial when deploying applications to the cloud and is the main contributing factor in the adoption of Docker. The reduced resource costs of containerised applications over VM-based deployments allows for a better user experience on the same hardware [24]. Use of Docker in particular has been shown to minimise costs associated with deploying microservice-based applications into cloud environments [21]. By using Docker to containerise applications, it is simple to pull an image and deploy it to a platform, but with VMs, the VM must be set up and configured, the application installed on the OS and then the application published to the chosen platform, which is time-consuming [21]. By being less time-consuming and less resource-intensive, Docker reduces costs by allowing smaller development teams to be more productive. The fact that Docker is open source, free to use (although it has now moved to a paid model, there is a free to use version for personal use) and cross-platform makes it accessible to both enterprise and individuals.

EVALUATION OF NODE.JS APPLICATION DEVELOPMENT

To run docker with Node.js, it is required to have the docker engine installed on the machine by installing docker desktop app. This includes both the engine and importantly, the docker Compose file. A benefit of the docker desktop app is that it has vulnerability scanning included and can give recommendations to resolve them in the images [25].

The first prerequisite is to create a “dockerfile”. This file is used to assemble the image. The file is used to set the working directory, install other dependencies and add meta data amongst other commands [26]. A benefit of the way a dockerfile works is that it is possible to make multiple dockerfiles for variations of an image and deploy them multiple times quickly, without needing to do it manually [27].

The docker compose file (docker-compose.yml) is key to containerising an application, as this is where the services are defined. For the example in class, the web server and port would need to be defined in this file. Additional dependencies such as ‘nodemon’ can be included to run via this file. Docker compose applies the rules defined within the configuration file, which is especially useful to improve management of multiple containers.

Only after the docker-compose file is complete, should the command to create the container be run. This is the “docker-compose up” command. This command allows multiple micro-services to be coordinated to run together. In the case of the class exercise, this would be run in the WSL command-line functionality, as we are using the ubuntu sub-system for managing the NPM packages. Docker can be used on ubuntu by pulling it down from the correct repositories. One way to check that it is set up successfully is to check if it is created a local Docker registry running on 127.0.0.1:5000.

CONCLUSION

The growth in cloud computing has been driven by its wide range of applications and diversity of services and the huge advantages over traditional networks and development [11]. Cloud computing has allowed both individuals and corporations to access resources and software flexibly, cheaply and reliably [17]. The downside is that the pervasiveness of cloud computing and the relatively small number of major service providers means that there can be serious real-life consequences when a service outage or other problem occurs [16]. The growth of cloud computing has also been aided by the innovations in technology, such as Docker and node.js, that have made resource management more efficient, and made real-time applications in the cloud feasible.

BIBLIOGRAPHY

- [1] Microsoft Azure, "What is Cloud Computing? A Beginner's Guide," Microsoft Azure, [Online]. Available: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/>. [Accessed 04 01 2023].
- [2] S. Goyak, "Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review," *International Journal of Computer Network and Information Security*, vol. 3, pp. 20-29, 2014.
- [3] S. M. a. G. K. Shyam, *Cloud Computing: Concepts and Technologies*, Oxon: CRC Press, 2021.
- [4] Microsoft Azure, "What is a private cloud?," Microsoft Azure, [Online]. Available: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-private-cloud/>. [Accessed 2023 01 05].
- [5] Amazon Webservices Inc, "Amazon EC2," Amazon Webservices Inc, 2023. [Online]. Available: <https://aws.amazon.com/ec2/>. [Accessed 05 01 2023].
- [6] Google Cloud, "App Engine," Google Cloud, [Online]. Available: <https://cloud.google.com/appengine>. [Accessed 05 01 2023].
- [7] Cloudflare Inc., "How are serverless computing and Platform-as-a-Service different?," Cloudflare Inc., [Online]. Available: <https://www.cloudflare.com/en-gb/learning/serverless/glossary/serverless-vs-paas/>. [Accessed 06 01 2023].
- [8] M. Radzikowski, "No, AWS, Aurora Serverless v2 Is Not Serverless," *Last Week in AWS*, 06 10 2022. [Online]. Available: <https://www.lastweekinaws.com/blog/no-aws-aurora-serverless-v2-is-not-serverless/>. [Accessed 06 01 2023].
- [9] Microsoft Azure, "What is SaaS?," Microsoft Azure, 2023. [Online]. Available: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-saas/>. [Accessed 12 01 2023].
- [10] IBM, "What is Database-as-a-Service (DBaaS)?," IBM, [Online]. Available: <https://www.ibm.com/topics/dbaas>. [Accessed 13 01 2023].
- [11] A. L. a. N.-C. Chen, "Cloud computing as an innovation: Perception, attitude, and adoption," *International Journal of Information Management*, vol. 32, no. 6, pp. 533-540, 2012.
- [12] M. Z. a. C. T. H. M. H. Ghahramani, "Toward Cloud Computing QoS Architecture: Analysis of Cloud Systems and Cloud Services," *Journal of Automatica Sinica*, vol. 4, no. 1, pp. 6-18, 2017.
- [13] Central Digital and Data Office, "Government Cloud First Policy," 21 07 2022. [Online]. Available: <https://www.gov.uk/guidance/government-cloud-first-policy>. [Accessed 15 01 2023].
- [14] NHS Digital, "NHS and social care data: off-shoring and the use of public cloud services," NHS Digital, 2022.

- [15] C. Miyachi, "Community of Practice: Who has your Cloud-Back?," *IEEE Cloud Computing*, vol. 5, no. 2, pp. 7-10, March/April 2018.
- [16] A. Bednarz, "Top 10 outages of 2022," Network World, 13 01 2023. [Online]. Available: <https://www.networkworld.com/article/3685490/top-10-outages-of-2022.html>. [Accessed 15 01 2023].
- [17] J. C. Delgado, "An Interoperability Framework for Enterprise Applications in Cloud Environments," in *Enterprise Management Strategies in the Era of Cloud Computing*, Hershey, PA, IGI Global, 2015, pp. 26-59.
- [18] Oracle, "Oracle VM VirtualBox User Manual," Oracle, 2022. [Online]. Available: <https://www.virtualbox.org/manual/UserManual.html>. [Accessed 15 01 2023].
- [19] VMware vSphere, "Insufficient CPU or Memory Resources," VMware, 18 07 2022. [Online]. Available: <https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.resmgmt.doc/GUID-B5117351-985E-4D2D-8C20-5A085C4D9253.html>. [Accessed 14 01 2023].
- [20] D. Ward, "The pros and cons of using a virtualized machine," Control Engineering, 10 03 2015. [Online]. Available: <https://www.controleng.com/articles/the-pros-and-cons-of-using-a-virtualized-machine/>. [Accessed 13 01 2023].
- [21] X. G. T. W. G. B. & B.-Y. C. Xili Wan, "Application deployment using Microservice and Docker containers: Framework and optimization," *Journal of Network and Computer Applications*, vol. 119, pp. 97-109, 2018.
- [22] middleware, "Containerization vs. Virtualization: Top Differences DevOps Should Know," middleware, 12 01 2023. [Online]. Available: <https://middleware.io/blog/containerization-vs-virtualization/>. [Accessed 13 01 2023].
- [23] B. Ciszewski, "Node.js Advantages – How Your App Will Benefit From This Javascript Framework," netguru, 18 07 2018. [Online]. Available: <https://www.netguru.com/blog/node-js-advantages>. [Accessed 14 01 2023].
- [24] Z. L. C. W. Ruiting Zhou, "Scheduling Frameworks for Cloud Container Services," *Scheduling Frameworks for Cloud Container Services*, vol. 26, no. 1, pp. 436-450, 2018.
- [25] Docker Inc, "Docker Desktop," Docker Inc, 2023. [Online]. Available: <https://www.docker.com/products/docker-desktop/>. [Accessed 12 01 2023].
- [26] docker docs, "Dockerfile reference," Docker Inc, 2023. [Online]. Available: <https://docs.docker.com/engine/reference/builder/>. [Accessed 12 01 2023].
- [27] J. Wallen, "Docker Basics: How to Use Dockerfiles," The New Stack, 19 06 2019. [Online]. Available: <https://thenewstack.io/docker-basics-how-to-use-dockerfiles/>. [Accessed 12 01 2023].
- [28] Web Dev Simplified, "JavaScript Shopping Cart Tutorial for Beginners," Web Dev Simplified, 4 10 2018. [Online]. Available: <https://www.youtube.com/watch?v=YeFzKC2awTM>. [Accessed 05 01 2023].

APPENDIX 1: VIDEO LINK

For the practical section of the first assignment for this module, a dynamic webpage using AJAX and a REST API was created to exchange data with webservices. For the second assignment, the completed webservice was updated to include eCommerce functionality [28]. A demonstration of the application and code explanation can be found in the following link.

<https://www.youtube.com/watch?v=kggi1ZjbQ00>