

CLOUD-BASED WEB SERVICES

ASSIGNMENT 2 - ACCA6024

Siobhan Stephens - 1800709

UNIVERSITY OF WALES TRINITY SAINT DAVID

Table of Contents

1.	Introduction	1
2.	Cloud Computing	1
2.1.	<i>Benefits and Drawbacks of Deploying Applications to the Cloud.....</i>	<i>1</i>
3.	Docker - Virtualisation and Containerisation.....	5
4.	Development Process of Node.js Applications for Docker.....	6
5.	Evaluation.....	8
6.	References.....	9
7.	Appendix A.....	11
7.1.	<i>Cloud Service Types.....</i>	<i>11</i>
7.2.	<i>Netflix Case Study</i>	<i>11</i>

1. Introduction

Cloud Computing has changed the way in which IT services are developed, deployed, scaled, maintained, and funded since its widespread rise around 2006 [1]. The key aspect of cloud computing is that it lowers the entry point to high performance computing, allowing organisations of any size to “leverage computing power for which they have neither the capital budget, nor the operational expertise to acquire” [1].

Daily lives rely heavily on cloud-based services, for data storage, writing documents, accessing software, managing business operations, and playing games online. Cloud computing has provided the infrastructure for digital transformation and allowed for key trends including the Internet of Things, big data, artificial intelligence, and mobile computing [2].

This report will explore the benefits of cloud computing over traditional IT service environment-including scalability, flexibility, reduced capital, and higher resource utilization [3].

2. Cloud Computing

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centres and servers, you can access technology services such as compute, storage, and databases on an as-needed basis from a cloud provider [4]. In 2021, the cloud is used by organisations of every time, size, and industry for a wide variety of use cases, including software development and testing, data backup and disaster recovery, virtual desktops, big data analytics and web applications [4].

Cloud computing is often split into three services, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [2]. Appendix A.1 outlines these service types.

2.1. Benefits and Drawbacks of Deploying Applications to the Cloud

It can be difficult to compare on-premises costs with cloud costs as an exact like-for-like, therefore for the purpose of this report the costs involved can be broken down into cost of the server hardware, power consumption, storage and space. In addition to this, there is an assumed a 3-year lifecycle for all hardware.

A single Dell R320 server costs \$2,000 to purchase, or \$55 per month over the lifecycle [5]. Assuming a cost of \$10 per square foot per month for on-premises space, approximating the required space of 8ft square or \$80 a month. Energy consumption can be approximated to \$29. Therefore, a single on-premises server is said to cost approximately \$164 per month [5], without considering security, staffing or maintenance costs. An equivalent server in Microsoft Azure is a D2 2 core with 7GB RAM and 100GB storage, costing approximately \$200 per month [5], which includes power consumption, server and space. Consequently, in this scenario cloud infrastructure is 20% more expensive than on-premises. However, the benefits of deploying to the cloud must be considered in addition to cost.

For example, the time to provision a single working cloud server is said to take approximately 15 minutes [5], comparatively it would take days or weeks to order, receive, install, and set up a single on-premises server, and would require specialist personnel to install, incurring potential further costs.

One of the primary benefits of cloud computing is the low initial cost when compared to traditional on-premises compute [6], which requires a significant initial outlay to purchase hardware resources, cover installation fees and purchase software licences, however, also requires weighty maintenance and ongoing costs for electricity, cooling, hardware and software updates, and personnel to undertake the

maintenance [6]. The savings from initial capital expenditure could be more efficiently invested into innovation or growth of an organisation. In addition to this the low entry-cost provided by the on-demand pricing models remove the barrier of large investments on hardware, allowing smaller businesses and start-ups to achieve their desired objectives much sooner and produce solutions with less risk and hassle. One of the main reasons for cost savings whilst using a large cloud provider is the cost-benefit they gain using shared data centres and economies of scale, which is then passed down to the customer.

It is widely known that most enterprise servers are under-utilised, increasing costs for both hardware and maintenance for surplus servers. Unlike on-premises servers, if a cloud server is no longer required it can be ‘dropped’ quickly and easily to reduce costs. To maximise the potential cost savings provided by the cloud, it is advisable to continually monitor usage to check the utilisation of resources, and if under-utilised, use cheaper hardware with less RAM or virtual CPU cores. For example, an AWS EC2 instance in eu-west-2 (London) of type t4g.small with 2 vCPUs and 2GiB of memory is \$0.0188 an hour or \$8.69 a month, compared to \$0.1504 an hour for a t4g.xlarge with 4 vCPUs and 6GiB, which is approximately \$69.28 a month, a difference in cost of \$60.59 a month on only one instance [7]. If this were to be multiplied across many instances, the cost savings could be a great amount.

In addition to this, auto-scaling possibilities throughout cloud solutions allow resources to auto-scale when they are not being used, for example many websites are very active during the day, and not at night, and so scaling back resources overnight and back up as traffic ensures optimal capacity and performance at any given moment to ensure the most cost-effective solution [6]. Auto scaling offers better fault tolerance as it can detect when an instance is ‘unhealthy’ and if necessary, terminate the instance and launch a new one to replace it, potentially across availability zones if required, which reduces the risk of application outages [8].

Server maintenance is a time consuming and costly task. A major benefit of using a cloud provider is that they manage the maintenance, improvements, and updates, ensuring that customers have access to the latest technology and software versions without having to invest effort into update and upgrade planning, development, and execution. For example, if a new version of Linux was released, the cloud provider can automatically apply the updates for you.

The cloud offers better reliability than on-premises servers, with service-level agreements (SLA) and service commitments to ensure their services remain available. AWS state that “[we] will use commercially reasonable efforts to make Amazon EC2 and Amazon EBS each available with a Monthly Uptime Percentage of at least 99.99%” [9]. Azure state that “we guarantee you will have Virtual Machine Connectivity to at least one instance at least 99.99% of the time” [10]. There is no guarantee with on-premises servers. Appendix A.2 outlines a case study of an outage that affected Netflix and AWS and how AWS responded to the event.

Cloud providers group data centres into availability zones, connected to each-other through low-latency links. If a data centre fails in one of these zones, due to technical or environment damage, then other instances of the application can continue to run in data centres in other zones. This redundancy reduces outages and the potential for data loss. To ensure the overall resiliency of a system, it is critical to always have good visibility into the state of the system in order to detect a failure quickly, and ways to mitigate the failure once it occurs [8]. Cloud providers offer tools such as AWS CloudWatch [11] and Azure Monitor [12] to enable engineers to implement operational metrics in a scalable way, and trigger alerts based on any number of possible failure scenarios [8].

One of the other main benefits of deploying applications to the cloud is that applications can be run within geographical regions anywhere in the world, ensuring compliance with data protection regulations and achieving optimal performance with low latency where the data is closest to the consumer [8]. AWS for example offers cloud services across 25 geographic regions, with plans to launch five more AWS regions [13]. Of the cloud providers, Azure offers the highest number of regions at 60+ regions [14]. This is especially helpful in worldwide organisations, or organisations looking to

branch out to new regions. There is no need to replicate the hassle of multiple data centres and maintenance throughout many countries, instead technical personnel can all work from a single base and remotely deploy applications all over the world.

One of the most critical benefits provided by a Cloud provider is the upgraded security. Cloud providers such as AWS and Azure are continually testing for security breaches and improving their services, software, and hardware to protect against them. However, many of the security issues faced by traditional on-premises infrastructure still exist within the cloud. Cloud providers work with a shared-responsibility model, for example AWS states that “While AWS manages security of the cloud, you are responsible for security in the cloud. This means that you retain control of the security you choose to implement to protect your own content, platform, applications, systems, and networks no differently than you would in an on-site data centre.” [15].

The security risk also differs between the varying cloud delivery types, where IaaS shares the greatest risk to the consumer, PaaS bears a higher risk than the consumer as the provider is in control of more of the layers, and SaaS where the security responsibility lies mainly with the service provider – the consumer is mainly responsible for security client-side vulnerabilities [16]. This may be of benefit to the consumer as a significant amount of responsibility is taken from them, especially in PaaS services.

One of these risks includes data breaches, with data protection moving from cloud consumer to cloud service provider, the risk of accidental, malicious, and intentional data breach is high [16]. Denial of Service (DoS) attacks may also cause outages as any denial-of-service attack on the cloud provider can affect all tenants [16], however providers may provide services such as AWS’ CloudFront (Content Delivery Network CDN) which helps to prevent DDoS attacks, hacking attacks and act as caches for data to increase performance [17]. Governance and compliance are also potentially complicated using the cloud, due to the lack of visibility and control of some services, however most cloud providers have certification, such as ISO 9001, ISO 27001, ISO 27017, ISO 27018 [15] ensuring that best security practices are in place. To ensure compliance within the cloud, employees may require new training and new best-practices implemented to meet the new requirements and platform.

Another drawback is that cloud providers may make changes to their technologies or business models, in which case consumers must make changes to ensure compatibility and eliminate exposure to risk. Additionally, shared technology vulnerabilities such as of a hypervisor or cloud orchestration, where an increased leverage of resources, gives the attackers a single point of attack which can cause significant damage [16]. As stated by Ramachandra et al “Cloud computing due to its disruptive nature, complex architecture, and leveraged-resources pose a unique and severe risk to all actors. It is critical to all stakeholders and actors to understand the risk and mitigate it appropriately. Security needs to be built at every layer in a cloud-computing platform by incorporating best practices and emerging technologies to effectively mitigate the risk” [16].

In contrast to the like-for-like scenario outlined above, other options such as serverless functions can be used, including AWS Lambda functions or Azure functions, which allow developers to focus on writing code and not infrastructure. They allow simple and quick access to compute in multiple languages without the need to specifically provision a server, and when combined with API Gateways (AWS) or API Management (Azure), complex APIs can be very quickly configured. This also allows organisations to spend more time innovating than focusing on the hardware aspect, which speeds up new trends and technologies [18].

To compare multiple on-premises servers to multiple cloud-hosted servers, we will assume 42 servers are required. The cost for 42 on-premises servers can be simplified to \$54 (racks) + \$55 (software) + \$275 (load balancing) + \$2,310 (servers) + \$1,218 (power) + \$160 (space) = \$4,072 per month [5] without storage, or \$8,400 per month for the cloud.

Assuming 1TB of storage is required, 1TB of cloud storage can be simplified to \$0.10 per GB, a total of \$100. The storage offered by cloud providers is spread across availability zones, to ensure

redundancy. Assuming the storage of data is spread across 6 nodes in total: 3 local nodes and 3 external geographically separate nodes, this offers 6 points of redundancy [5]. When attempting to replicate this with on-premises servers, additional data-centre space must be considered, along with the staffing for this additional centre. Costs for on-premises storage for an equivalent 6TB would be approximately \$8,000 per month, raising the total on-premises cost to approximately \$12,000, compared to \$8,500 in the cloud, a potential saving of 30% [5].

When the cost-savings and benefits of the cloud are considered, it is no surprise that the adoption of cloud services has increased significantly in recent years. However, whilst uptake of the Cloud is on the rise, there is still a lack of employees with the relevant knowledge on cloud computing, and it may not be feasible to up-skill or hire a group of individuals to undertake cloud migration and maintenance. Although, there are now many organisations which offer cloud management services such as migration, and even aid in upskilling existing developers to allow them to be mostly self-sufficient in deploying applications to the cloud. In addition to this, many cloud providers also offer training and certifications such as Cloud Developer to allow employees to improve their knowledge.

Finally, whilst developing new applications may be a simple process, migrating existing applications to the cloud has many complications, requiring thorough pre-testing and planning. Difficulties occur with legacy systems and compatibility, and it may be a good time for an organisation to re-develop their applications to modernise during this process. Integration may also be more complex between the public cloud and on-premises systems, requiring careful planning [19].

3. Docker - Virtualisation and Containerisation

Virtualisation is the process of creating a software-based or “virtual” version of a computer, with dedicated amounts of CPU, memory and storage that are borrowed from a physical host computer or server [20]. Virtual Machines or VMs are called images. An image is a computer file that behaves like a physical computer [21]. VMs can run in a new window or in a separate environment and are often used to run different operating systems. The VM is partitioned from the host computer’s primary operating system and therefore cannot make changes outside of its own set partition.

Virtual Machines can be used to build and deploy applications to the cloud, for example to quickly spin up a new test environment to run dev-scenarios on varying operating systems, to run legacy or old operating systems, or to run software that is no longer supported by new operating systems [22]. Virtualisation and virtual machines provide other benefits including cost savings, as multiple environments can be run from a single infrastructure. VM images are also highly portable allowing them to be moved between hypervisors on different machines, enabling them to be a good backup solution. They also allow for increased scalability as VMs can be replicated across hardware to increase the availability and performance of applications. In addition, VMs provide increased security as they are partitioned from the host OS.

Paenevirtualization, or containerisation is a type of Operating-System level virtualisation, where there is no hypervisor. Like virtual machines, containers allow for applications to be packaged with libraries and other dependencies providing isolated environments for running software, however, the fundamental difference between containers and virtual machines, is that instead of virtualising the hardware stack, containers virtualise at the OS level, with multiple ‘containers’ running above the OS kernel directly [23]. This means that containers are more lightweight and use a fraction of the memory compared to VMs due to them not needing to boot an entire OS [24].

Containers offer a way for applications to be abstracted from the environment in which they run, allowing them to be deployed easily and consistently across different hardware and infrastructure – from development on a personal laptop to deployment onto cloud servers [23]. This separation allows developers to focus purely on application logic and dependencies without concern of matching software versions and configurations across multiple platforms [24]. The diagram below outlines the structural differences described between containers and virtual machines.

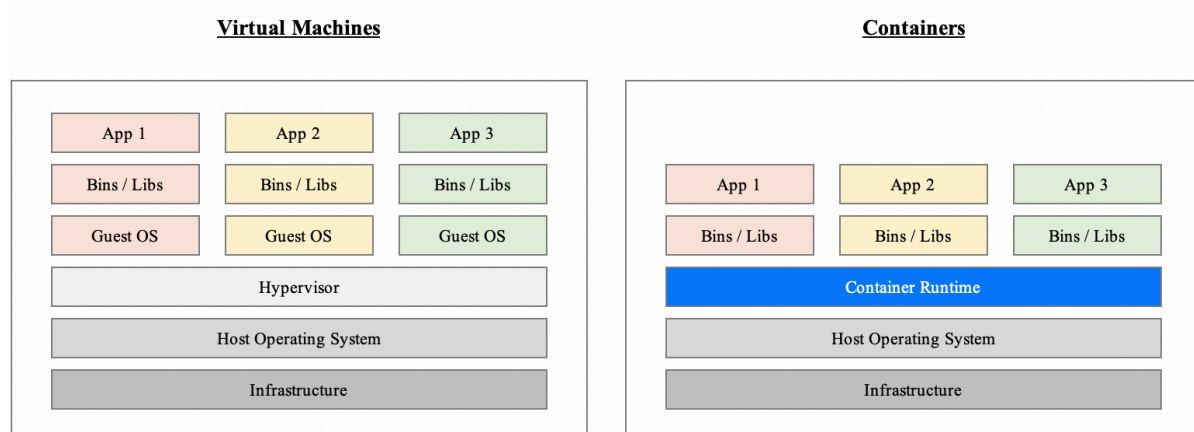


Figure 1. Containers vs Virtual Machines [24]

Despite the many benefits of VMs, the lightweight, portability of containers have enabled their popularity to increase.

Docker is an open-source platform that runs applications and aids in the development and distribution process [25]. In recent years, the popularity of docker has increased due to its many benefits [26]. Docker allows facilities to create and control containers on a local machine, and co-ordinate with third party resources which help to easily deploy and manage docker containers which can then be pushed into the cloud simply and easily [26]. Docker is designed to provide a quick and lightweight environment where code can be run efficiently and consistently, with applications built in Docker packaged with all required dependencies into a container. For this reason, Docker is heavily used in the development of applications that require networked services such as web servers [25]. The main advantages of docker are speed, portability, scalability, rapid delivery, and density [26].

Docker containers are easily scalable as they can be deployed across multiple platforms quickly and easily, such as between local and cloud environments. The format of a Docker container is standardised and so development teams can ensure compatibility across environments, providing consistent and predictable results between local, test, and production environments [25]. Containers provide the added benefit of being more lightweight than their VM counterpart, however Docker improves upon this due to the controlled management of containers and no overhead wastage of resources [26].

It is for these reasons that Docker is the leading PaaS for OS-level virtualization [26].

4. Development Process of Node.js Applications for Docker

Node.js is an asynchronous event-driven JavaScript runtime, designed to build scalable network applications [27]. Throughout in-class exercises, the Express.js framework was used to build applications. Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications [28]. It provides HTTP utility methods and middleware, allowing developers to create both web servers and web services. Express provides a thin layer of fundamental web application features, without obscuring Node.js features [28].

The first step of building web applications in Node.js and Docker is to set up a working environment. This is where docker proves an extremely useful development tool – as the required environment elements can be defined in the docker-compose.yml file, which will be packaged along with dependencies to create the required services. For example, the docker.yml file used throughout in-class exercises provided us with a web server, a web service, a PostgreSQL database, adminer to view the database in a browser, a MySQL database, a mongo dB database and a network to link the services. Required packages and dependencies are outlined in the docker file, ensuring that if the project is shared with another developer on another machine, they can run the application within the same environment. In addition to this in later stages of the development process, the docker container can easily be deployed to a test environment and later to a production environment.

Next, is the process of building the web service. To begin building the web service the back-end database is first considered and implemented. As mentioned, adminer was created with docker and therefore can be used as part of this process. Using adminer as a browser-based graphical user interface allows for the database schema to be added quickly and easily without the need to write SQL queries.

Next, the web service's service.ts file is set up to include all of the required routes of the API. An example route is '/students' with the GET action. Within this route a call is made to the database to run an SQL SELECT statement, gathering all student data. Query parameters are easily added to routes using the express.js methods, for example '/students?id=1' can be split using res.query.id to gather the id and a query run to the database to return the selected student's details. The express.js framework's built-in methods allow routes to be written with calls to the database in only a few lines of code, allowing developers to focus on the functionality of the service and to build robust services in a short amount of time. Database connectivity is also added as part of the framework and requires only a few lines of code.

The web service running in docker can be easily tested by using the exposed port of 1339, using Postman [29]. Postman allows the developer to test the varying APIs in the web service before the web server has been set up and enables for much quicker testing than through the front end during the initial phases. Postman allows all requests to be sent, including GET, PUT, POST, DELETE and PATCH.

Once the web service is implemented, the web server and front-end code is to be added. The web server was exposed on port 80 during in-class exercises, allowing local testing on localhost. Firstly, routes were added to the server.ts file which link to the previously implemented web service APIs. An example of this is the '/' route with GET action. This route was set up in one exercise to return all student data on opening of the website, which calls a web service route '/students' with a GET action that returns json student data. The result of which was sent to be rendered by the EJS template 'home'. EJS templating, or Embedded JavaScript Templating allows for data to be rendered into views quickly and easily, with simple syntax [30].

Inside the ejs templates loops can be added to iterate through student details, simplifying the HTML data display and reducing the time required to build complex pages. For example, to display a list of student details, a loop can be used `'% students.forEach( (student) =>{ %>'` to print each list item, with individual data items to be printed using the loop variable such as `'<%=student.name%>'`. If this were to be done using jQuery it would be a more time-consuming process and would require more lines of code.

The overall method of allowing the developer to work on a web service and web server in parallel speeds up the development process significantly. In addition, by using the Node.js environment, the language and syntax is consistent throughout the project and so there is less of a steep learning curve as much of the code is similar for both the web service and web server. This also improves readability for other developers who may be new to the project.

5. Evaluation

Cloud computing has revolutionised the way IT services are developed, deployed, scaled, maintained, and has allowed organisations access to compute that would not have been previously possible due to the lower entry-cost and worldwide reach. It offers lower costs, greater scalability, reliability, security, and flexibility than on-premises hosting, and thanks to this, technology and innovation is growing at a greater rate. Cloud computing has provided the infrastructure for digital transformation and enabled key computing trends such as the Internet of Things, big data, artificial intelligence, and mobile computing to thrive. Additionally, it has offered a new career path and skill set for employees within IT.

6. References

- [1] M. G. Avram, "Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective," *Procedia Technology*, vol. 12, pp. 529–534, 2014: 10.1016/j.protcy.2013.12.525.
- [2] P. Srivastava and R. Khan, "A Review Paper on Cloud Computing," *International Journals of Advanced Research in Computer Science and Software Engineering*, no. 6, pp. 2277–128, 2018, doi: 10.23956/ijarcsse.v8i6.711.
- [3] S. VNandgaonkar and A. B. Raut, "International Journal of Computer Science and Mobile Computing A Comprehensive Study on Cloud Computing," 2014. Accessed: Apr. 27, 2021. [Online]. Available: www.ijcsmc.com.
- [4] "What is Cloud Computing." <https://aws.amazon.com/what-is-cloud-computing/> (accessed Apr. 15, 2021).
- [5] G. Deckler, "Cloud vs. On-Premises – Hard Dollar Costs," Mar. 02, 2017.
- [6] S. Ayob, "Cloud Computing Benefits CLOUD COMPUTING BENEFITS ii," 2016. Accessed: May 02, 2021. [Online]. Available: <http://ssrn.com/abstract=2781593><http://ssrn.com/abstract=2781593>.
- [7] "AWS Pricing Calculator." <https://calculator.aws/#/> (accessed Apr. 18, 2021).
- [8] Y. Izrailevsky and C. Bell, "Cloud Reliability," *IEEE Cloud Computing*, vol. 5, no. 3. Institute of Electrical and Electronics Engineers Inc., pp. 39–44, May 01, 2018, doi: 10.1109/MCC.2018.032591615.
- [9] "Amazon EC2 Service Level Agreement – Amazon Web Services." <https://aws.amazon.com/ec2/sla/historical/> (accessed Apr. 19, 2021).
- [10] "SLA for Virtual Machines | Microsoft Azure." https://azure.microsoft.com/en-gb/support/legal/sla/virtual-machines/v1_9/ (accessed Apr. 25, 2021).
- [11] "Amazon CloudWatch - Application and Infrastructure Monitoring." <https://aws.amazon.com/cloudwatch/> (accessed Apr. 22, 2021).
- [12] "Azure Monitor overview - Azure Monitor | Microsoft Docs." <https://docs.microsoft.com/en-us/azure/azure-monitor/overview> (accessed Apr. 22, 2021).
- [13] "Global Infrastructure." <https://aws.amazon.com/about-aws/global-infrastructure/> (accessed Apr. 15, 2021).
- [14] "Azure Products by Region | Microsoft Azure." <https://azure.microsoft.com/en-in/global-infrastructure/services/> (accessed Apr. 18, 2021).
- [15] "Security and Compliance - Overview of Amazon Web Services." <https://docs.aws.amazon.com/whitepapers/latest/aws-overview/security-and-compliance.html> (accessed Apr. 25, 2021).
- [16] G. Ramachandra, M. Iftikhar, and F. A. Khan, "A Comprehensive Survey on Security in Cloud Computing," in *Procedia Computer Science*, Jan. 2017, vol. 110, pp. 465–472, doi: 10.1016/j.procs.2017.06.124.
- [17] "Content Delivery Network (CDN) | Low Latency, High Transfer Speeds, Video Streaming | Amazon CloudFront." <https://aws.amazon.com/cloudfront/> (accessed Apr. 20, 2021).
- [18] H. Saini, A. Upadhyaya, and M. K. Khandelwal, "Benefits of Cloud Computing for Business Enterprises: A Review," *SSRN Electronic Journal*, Oct. 2019, doi: 10.2139/ssrn.3463631.
- [19] M. Haris and R. Zaman Khan, "A Systematic Review on Cloud Computing," *Article in INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*, 2018, doi: 10.26438/ijcse/v6i11.632639.
- [20] "What is virtualization?" <https://www.redhat.com/en/topics/virtualization/what-is-virtualization> (accessed Apr. 28, 2021).
- [21] "What is virtualization?" <https://www.redhat.com/en/topics/virtualization/what-is-virtualization> (accessed May 06, 2021).
- [22] "What is a virtual machine and how does it work | Microsoft Azure." <https://azure.microsoft.com/en-gb/overview/what-is-a-virtual-machine/> (accessed Apr. 29, 2021).

- [23] “What is a Container? | App Containerization | Docker.” <https://www.docker.com/resources/what-container> (accessed Apr. 29, 2021).
- [24] “What are Containers and their benefits | Google Cloud.” <https://cloud.google.com/containers> (accessed May 06, 2021).
- [25] C. Anderson, “Docker,” *IEEE Software*, vol. 32, no. 3. IEEE Computer Society, pp. 102–105, May 01, 2015, doi: 10.1109/MS.2015.62.
- [26] Assoc Prof Dr Babak Bashari Rad, “An Introduction to Docker and Analysis of its Performance,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 17, no. 3, Mar. 2017, Accessed: May 06, 2021. [Online]. Available: https://dlwqtxts1xzle7.cloudfront.net/52736106/IJCSNS-20170327-with-cover-page.pdf?Expires=1620301601&Signature=YX4SvGD3ayDscIZN86MVGr~HBNF5sUfVUqPI7UekV~eo0KTWXdlJrak3W0hjgoFn-QACZzzJgmy2f4jheQ7wVIusFs1rFwNF~o94GjyYt~nYl67ssFeNSShpizsjKOYeePY~2zhZP~oXez873tWwBlfsaoITzaXD8U2pJrFOFfv2TpukNJEPIKxr~UX~zu3GEf7y2e~eIhTQPXLztPsZEOYrdJepACmnd-LIj2gxRMv0WYBd0G~yIY34vPY9F~GtUXWsKtpwhG0w5zUQojM9c9QDsSwVkSP9Uk5qmWcXUORa4H1OOPow5ISC~8E6A79z5wHljumNUPGUMS2Mg4gvNg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA.
- [27] “About | Node.js.” <https://nodejs.org/en/about/> (accessed May 02, 2021).
- [28] “Express - Node.js web application framework.” <https://expressjs.com/> (accessed May 02, 2021).
- [29] “Postman | The Collaboration Platform for API Development.” <https://www.postman.com/> (accessed May 02, 2021).
- [30] “EJS -- Embedded JavaScript templates.” <https://ejs.co/> (accessed May 02, 2021).
- [31] “What is PaaS (Platform-as-a-Service) | IBM.” <https://www.ibm.com/cloud/learn/paas> (accessed Apr. 15, 2021).
- [32] “What is IaaS (Infrastructure-as-a-Service) | IBM.” <https://www.ibm.com/cloud/learn/iaas> (accessed Apr. 15, 2021).
- [33] “A Closer Look at the Christmas Eve Outage | by Netflix Technology Blog | Netflix TechBlog.” <https://netflixtechblog.com/a-closer-look-at-the-christmas-eve-outage-d7b409a529ee> (accessed Apr. 20, 2021).
- [34] “Summary of the December 24, 2012 Amazon ELB Service Event in the US-East Region.” <https://aws.amazon.com/message/680587/> (accessed Apr. 23, 2021).

7. Appendix A

7.1. Cloud Service Types

Software as a Service allows users to access software via the internet, in place of installing it directly to a computer. It removes the requirement for storage space, or maintenance of software updates, the user must simply have an internet connection and access to the application. Examples of SaaS are Microsoft Office 365 and Google Apps [6].

Platform as a Service is a platform or development environment upon which users can develop, deploy, and manage their own code and software to construct applications that run on the cloud provider's infrastructure without the cost and complexity of building and maintaining a platform on-premises [31].

Infrastructure as a Service provides customers with computing resources such as compute, storage, network and operating system hardware on demand, over the internet, on a pay-as-you-go basis [32]. A key aspect of IaaS is that it enables end users to scale up and shrink resources on an as-required basis, eliminating the need for high up-front expenditure or unnecessary surplus owned infrastructure, especially important for businesses with fluctuating workloads. IaaS also provides the lowest-level control of resources in the cloud [32].

7.2. Netflix Case Study

On December 24, 2012, Netflix experienced an 18-hour outage affecting most of its customers [33]. It resulted from a regional failure of the AWS Elastic Load Balancer (ELB) service in the US-East-1 cloud region, where Netflix had all of its cloud operations at that time. Subsequently, AWS made significant improvements to its ELB service architecture and configuration management process [34]. Also learning from this outage, Netflix moved to an Active-Active cross regional architecture, where customer traffic is load balanced across three cloud regions, and data is kept in an eventually consistent sync using Cassandra (a distributed, cross-regional NoSQL database) replication [8]. In case of an outage, traffic from the affected region can be redirected to the remaining healthy regions [8].

Izrailevsky et al stated that “Cloud application developers can employ well established design patterns, such as circuit breakers and bulkhead, to build systems that are resilient to failures of one or several underlying components” [8]. Occasional failures are inevitable in such large scale, complex distributed systems, however it is possible to reduce the criticality of these failures by embracing fault tolerant architectural principles such as redundancy [8].