

ACCA6024 Cloud Computing

1903723 HYWEL JONES

07/01/2023

Contents

Cloud Computing	1
Contents.....	2
BENEFITS AND DRAWBACKS OF APPLICATION CLOUD DEPLOYMENT	3
COSTS	4
SCENARIOS, EXAMPLES, RELIABILITY AND SCALABILITY	5
NODE.JS DEVELOPMENT IN DOCKER.....	7
Bibliography	8

BENEFITS AND DRAWBACKS OF APPLICATION CLOUD DEPLOYMENT

There are several different ways that an application can be deployed and Cloud Computing is one of those deployment mechanisms available for online applications. There are three basic models of deploying internet based applications, that is a local deployment, a cloud deployment or a hybrid of the two. A typical local deployment could involve a singular server running the application directly on hardware or it could be a more sophisticated local cloud which would typically be several virtual machines running different components of the application (such as webserver, database, load balancer etc). A cloud environment fully embraces virtualization in that the deployment would be on a virtual machine or even a virtual container based approach. [1]

As applications got more and more complicated and face more and more user traffic the problem of how to scale applications became a more widespread issue. Traditionally the solution was to upgrade the server running the application with a better CPU, more RAM or faster drives but this sort of scaling, often called "vertical scaling", quickly ran into issues. If traffic was sufficiently high or the application was using a lot of resources a couple thousand concurrent connections could easily cripple the capacity a single local server and this would lead to the entire site being brought down. As more and more applications were designed to leverage reusable web services and external resources the ability for a single application to be hosted on multiple servers started to become more feasible and the idea of cloud computing was born. Individual components of an application were split up, so for example the load balancer, application and database would be on their own virtual machine which could allow "horizontal scaling". Horizontal scaling is when the number of servers serving the application is increased instead of the quality of the hardware, for example if a website is receiving lots of incoming connections a new application server could be spun up and the load balancer directed to share the increased load between two fully functioning application servers. [3]

Spooling up new virtual machines was the first generation approach to cloud computing but that has now changed with improvements to virtualization, especially improvements such as Hyper-V, VMware and subsequently the adoption of containerisation technologies like Docker. [5] Docker allows an additional level of abstraction for virtualization, instead of using full operating systems in each virtual machine the containers now share an abstraction of the operating system between each other but still run their code in their own execution environment. [6]

COSTS

The price of deployment is also often specified as a reason to use cloud computing. Business are able to convert large upfront capital costs of servers into monthly payments that only cover the server infrastructure that they need. This can be especially useful when a company becomes suddenly successful and faces a substantial increase in web traffic, it is easier to increase the monthly payments rather than having to buy several high spec servers. This approach is also quicker as there is no need for the new server to be delivered, the cloud instance can spin up a new virtual machine in minutes. For the price comparison we will use the two cloud service providers that currently have the biggest market share, that is Amazon AWS and Microsoft Azure. [8]

Virtual Machine Comparison:

Azure Virtual Machine (B-tier instance) [2 core 8GB RAM £57.2/month] [6]

Amazon (AWS EC2 instance) [2 core 8GB RAM £54.9/month] [7]

Container Group Comparison:

Azure Container [£0.00443 per GB per hour, £0.04038 per CPU per hour] [calculated per second] [9]

Amazon (AWS EC2 pay-as-you-go instance 1CPU, 0.5GB) [£0.116 per hour] [calculated per second] [10]

As we can see from the above the full virtual machine instances are more expensive than the containerized instances. Here the typical VM environment is about £56 (average between both) for a month whilst the container is about £57 a month (average of both at 100% util for a month), the VM environment being a flat fee whilst the container fee is a maximum for 100% utilization of those resources for the entire month. Most web applications are very light on resources and at most have occasional spikes in resource utilization that can let the containerized approach be cheaper.

SCENARIOS, EXAMPLES, RELIABILITY AND SCALABILITY

A relatively easy scenario to envisage is a basic container website, that is a single container web server, single container web application container with a single database container (a similar configuration to the app developed for this assignment) that suddenly becomes very popular due to some external viral advertising. To allow the site to suddenly expand and cope with this increasing load the containerized approach gives us powerful options to scale.

Here we would consider adding an Nginx reverse proxy server to load balance across our web application containers as we spin up several more. In this configuration the Nginx instance is able to distribute requests from our simple and lightweight web server instance across multiple docker containers that are running our resource intensive application containers yet the response is routed back to the client as they were talking to one server. Once set up the difficulty of adding more application servers is reduced to a very simple task of adding another docker container and adding its IP address to the Nginx config file. [11]

It also adds other interesting avenues for scalability through additional development, for example our current example uses one single Postgres database but if we wanted to add substantial full text search capability over our product catalogue a relational database might not be the best scalable solution. We could use a NoSQL database, such as Elasticsearch, and in this scenario our already scaled solution consisting of multiple web application containers would now point to a single Elasticsearch instance (which can be horizontally scaled itself across multiple nodes) for the specific full text search functionality. This does increase complexity in synchronizing the two data stores but can provide significant performance increases in certain use cases, in this scenario the NoSQL allows us to improve fetch performance whilst sacrificing insert speed which isn't needed for the search functionality.

Another important additional aspect that can be included to increase scalability even further in the cloud is the geo location of cloud servers. To have a highly responsive app that has low latency it is often required to have an app instance in a geographically close location to where the majority of users will access it from, for example having one server for the USA and one for Eastern Europe is a common setup to serve US and European customers. Since most cloud providers have data centres that can serve these different geolocations it is possible to have multiple application instances deployed for each geolocation that needs to be covered. These setups can share a database instance if data is not region specific or can use their own database instance, for example in our shop example we might have different products available for US customers and at different price points to the European customers. This type of solution can be achieved by deploying a second instance of our app in a different geolocation and then setting up an authoritative DNS server that can respond to a name resolution request of our site with a response based on the users geolocation data, similar to how European users are directed to "Google.co.uk" whilst US users are directed to the "Google.com" site based on their location. [13]

Reliability is also an important issue that the cloud can address in several ways. The actual cloud infrastructure is managed by the company, such as Microsoft or Amazon and as such has several data centres and many staff working 24/7 to maintain and upgrade the infrastructure so to keep uptime as high as possible. For example Azure has a "uptime SLA" regarding its container app services that gives credit back to the customer if service ever degrades past a certain level. The current ones for Azure container services being: 10% reduced cost at less than 99.95% uptime, 25% reduced cost at less than 99% uptime and 100% refund if less than 95% uptime. Compared to some on premise solutions for small businesses where entire servers need to be taken down regularly for maintenance and there is limited to no resiliency, this can be a substantial boost to reliability. [12]

NODE.JS DEVELOPMENT FOR DOCKER

Due to the popularity of Nodejs and of Docker there has been significant improvements in the tooling for developers. Visual Studio Code for example now has excellent javascript intellisense, allowing developers to find fields and methods much easier than it previously. The integration with the linux command line is also of great benefit, with NPM, docker and other commands to do all variety of different operations all in the same terminal, usually this sort of consolidation would be from a full IDE (Integrated Development Environment) such as Visual Studio.

Using docker as a container for the various components of the application has also dramatically sped up development time. The installation process for MySQL (not used), Postgres and NodeJs was all condensed and took substantially less time than installing and configuring each of these individually as one would normally on windows machines. Due to the docker image this setup is now also highly portable and was utilised for this project as it was migrated from the virtual machine set up in class to a personal Linux machine.

As we have covered different cloud providers above this container approach obviously allows us to switch from, say, an Azure deployment to an AWS deployment with minimal configuration changes. This can also be used to deploy additional regional deployments, for example if a new local application would need to be deployed to a specific geographical location.

Adding additional scalability via Docker containers is also easier and cheaper, adding a configuration line is all that is required for spawning up another container and the business only has to pay for the CPU/Memory utilization of that container. Non cloud solutions would need additional hardware and/or would need to install a new server OS after creating a new VM on their local infrastructure. Depending on updates and server versions, such as differing windows updates from the fresh install to the currently running server, simple deployment tasks can quickly become problematic.

Bibliography

- [1] Wikipedia Contributors, "Cloud computing," *Wikipedia*, Apr. 14, 2019.
https://en.wikipedia.org/wiki/Cloud_computing
- [2] S. S. Manvi and G. K. Shyam, *Cloud computing: concepts and technologies*, First edition. Boca Raton, FL: CRC Press, 2021.
- [3] T. Erl, R. Puttini, and Z. Mahmood, *Cloud computing: concepts, technology, & architecture*. Upper Saddle River, NJ: Prentice Hall, 2013.
- [4] "What Is Hardware Virtualization? Definition from SearchServerVirtualization," *SearchITOperations*. <https://www.techtarget.com/searchitoperations/definition/hardware-virtualization>
- [5] "There's Application Virtualization and There's Docker | Docker," *www.docker.com*, Apr. 15, 2016. <https://www.docker.com/blog/app-virtualization-docker/> (accessed Jan. 09, 2023).
- [6] "Pricing - Windows Virtual Machines | Microsoft Azure," *azure.microsoft.com*.
<https://azure.microsoft.com/en-gb/pricing/details/virtual-machines/windows/#pricing>
- [7] "AWS Pricing Calculator," *calculator.aws*. <https://calculator.aws/#/addService/ec2-enhancement> (accessed Jan. 09, 2023).
- [8] "Cloud market share 2021: An Overview Of This Growing Ecosystem.," Oct. 14, 2021.
<https://www.wpoven.com/blog/cloud-market-share/>
- [9] "Pricing - Container Instances | Microsoft Azure," *azure.microsoft.com*.
<https://azure.microsoft.com/en-us/pricing/details/container-instances/> (accessed Jan. 09, 2023).
- [10] "EC2 Instance Pricing – Amazon Web Services (AWS)," *Amazon Web Services, Inc.*, 2003.
<https://aws.amazon.com/ec2/pricing/on-demand/>
- [11] "How to run a load-balanced service in Docker containers," *Superuser*, Sep. 16, 2016.
<https://superuser.openstack.org/articles/run-load-balanced-service-docker-containers-openstack/> (accessed Jan. 09, 2023).
- [12] "SLA for Azure Container Apps | Microsoft Azure," *azure.microsoft.com*.
https://azure.microsoft.com/en-us/support/legal/sla/container-apps/v1_0/ (accessed Jan. 09, 2023).
- [13] JasonGerend, "Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers," *learn.microsoft.com*. <https://learn.microsoft.com/en-us/windows-server/networking/dns/deploy/primary-geo-location> (accessed Jan. 09, 2023).