

Cloud computing report

What is cloud computing

IBM defines cloud computing as:

“Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP).”[1]

But what does this mean in practical terms?

Consider that most businesses need access to data storage infrastructure. They have two options:

- 1) Purchase, set up and maintain all of the required hardware and software themselves
- 2) Outsource this to a CSP.

This, in its simplest form, is cloud computing – removing the requirement to have all of the physical hardware owned and located at/within a business, reducing the maintenance costs and staff requirements to keep them running, providing for instantaneous scalability etc.

As the opening definition states, cloud computing doesn't just stop at data storage and below, the concepts behind modern day cloud computing will be discussed.

SPI model

Cloud computing is broken down into a number of areas, with the three key being:

- 1) Software as a Service (SaaS)
- 2) Platform as a Service (PaaS)
- 3) Infrastructure as a Service (IaaS)

SaaS

Starting with software as a service, Microsoft defines it as:

“Software as a service (SaaS) allows users to connect to and use cloud-based apps over the Internet. Common examples are email, calendaring, and office tools (such as Microsoft Office 365).”[2]

There has been a real time shift to companies offering SaaS, and it can be observed in the changes to the ownership model of software packages. Take into account something like adobe photoshop, or Microsoft office suite – previously, these would have been sold as a product in which the user owns and installs on their machine and runs. In recent years, software developers have moved towards subscription models, instead charging a monthly/annual sum for access to the software.

The benefits to the user are that the software packages should always be available at the latest version, without any need to pay for updates. Not only that, but access should be universal, without the requirement of an install disc and key. For in browser software, this also means that access across platform, such as phone and tablet, should also be a lot easier.

This does come with some downsides, however, as it can be to the detriment of infrequent or casual users of software packages, to require a subscription. When software was previously owned, old versions could still be used on a machine from time to time without requiring additional payment.

PaaS

Platform as a service is defined by Microsoft as follows:

“Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.”[3]

Like IaaS, PaaS incorporates infrastructure, but can also incorporate more of the stack, such as middleware & OS[4].

To understand PaaS, think of a rental car. You don’t own it, but can use it for your requirements, however, it still needs a driver and you still need to put fuel into it. The difference between PaaS and SaaS in this example, would be something like a taxi – SaaS would fully deliver everything you need for that one trip – the road, the car, the driver and the fuel.

IaaS

Google defines IaaS as:

“IaaS is the on-demand availability of almost infinitely scalable computing resources as services over the internet. It eliminates the need for enterprises to procure, configure, or manage infrastructure themselves, and they only pay for what they use.”[5]

Think of this like the data storage hardware example in the introduction.

The benefits of outsourcing computing infrastructure have been mentioned above, but to reiterate: businesses can save time and money on purchasing, setting up and maintaining infrastructure. Not only that, but IaaS offers scalability, reliability and can even offer a faster time to market for start ups.

To compare to the example of PaaS above and the rental car, IaaS would be more like the road the car drives on. If you want to get from point A to B and you own your own car, have a driver and have fuel, IaaS has already built the road you will travel on. Without IaaS you would have to build those roads yourself.

Pros and cons of cloud computing

Some have been discussed already, such as time and cost saving. But its worth pointing out that there is a big negative with cloud computing that is often overlooked – vendor lock-in.

As its often difficult to move from one provider to another, companies can often be locked into the vendor they initially chose, despite others making improvements at a later date and offering services to better match the clients needs.

Although, that’s not to say that companies are entirely beholden to their providers – as its perfectly feasible that large companies with multiple different divisions, may utilise multiple different cloud providers.

Costing exercise

Imagine the scenario where a business purchased all their own hardware and deployed it, how much would it cost? How much would it cost to move it to the cloud? Below, this will be investigated, along with some discussion on which would be the best option.

Hardware specification and cost

The stated hardware specification is as follows:

- Processor: Intel Xeon Gold 6248 2.5 GHz
- RAM: 64 Gb Server RAM
- Storage: SAMSUNG PM1725b Enterprise SSD 12.8 TB
- Server Rack: APC Net Shelter CX 38U

The assumption made here is that the hardware will be purchased new (i.e. not refurbished) and will come with no installation, i.e. the company sets it up themselves and the price will include VAT.

Processor cost:

Its difficult to find a new price for this component (none-refurbished) as it appears to be quite dated, however, Dell [6] has it priced at £4,762.79.

Other prices are available, but from vendors such as amazon market place, or refurbished sellers.

RAM cost:

Description leaves a LOT to be desired as theres no tech spec at all, so how long is a piece of string? Going to assume here as Samsung storage is picked later, that a more premium quality ram is desired, so will also choose Samsung RAM. Without knowing too much about Ram specifications, the price comes to £308.12 including VAT, based on [7]

Storage cost:

£3970.16 – based on [8]

Rack cost:

£6,669.78 – based on [9]

Total cost:

£15,710.85.

Cloud cost

Monthly estimate	
\$3,291.36	
That's about \$4.51 hourly	
Pay for what you use: No upfront costs and per second billing	
Item	Monthly estimate
20 vCPU + 64 GB memory	\$848.70
12800 GB SSD persistent disk	\$2,611.20
10 GB balanced persistent disk	\$1.20
Sustained use discount	-\$169.74
Total	\$3,291.36
Compute Engine pricing	
^ LESS	

Some notes: N2 type chosen with minimum of cascade lake and two vCPU per core to match the processor. SSD was chosen for the persistent disk to match the storage. This is inc VAT.

Cost Discussion:

Over a year, the cloud storage cost would be almost \$39,500 – if we adjust to £ using xe.com, that would be £31,316.76.

While this cost, on the face of it, looks much steeper than the set up cost for the hardware, around twice the amount and only for a single year – it really doesn't paint the full picture. Firstly as we mentioned before, there are staffing costs to maintaining this hardware. Either internally, or using an external consultant, these can rack up to thousands of pounds. Next, if anything breaks or needs replacing, that's another cost – and if it's the storage, which SSDs have limited read/write capacity, it could feasibly need replacing in a few years.

Lastly – there is one huge thing missing from this, and that's cooling. Servers generate a LOT of heat and need round the clock cooling, which just isn't cheap at all. The cost of the required space for the server, setting up required air conditioning, and maintenance of the air conditioning all need to be factored in too.

Next, if we're talking about a start up that is doubling their growth annually, this server set up does not allow for scalability in the same way the cloud services do, especially on such a fast scale where air conditioning set ups and physical locations of servers may need to change in a few years.

For these reasons alone, and without costing further, I believe the cloud services to present better value for money and would choose to use those over setting up a server.

Cloud practical assignment walkthrough

Deployment decisions and comments:

I decided to use google cloud platform for hosting both the VM and database for this assignment.

In all honesty the decision wasn't driven by anything to do with costs that may incur in the assignment, as they were always going to be minimal, but it was mainly driven by practicality.

I had issues with amazon and getting my card to work as it was already linked to an amazon business account I had set up because of I do twitch streaming, and it wouldn't offer me the free tier.

I also looked into Azure and DigitalOcean and both platforms offered \$100 of free credit to students, but google cloud platform offered an impressive \$300 of credit. I'm aware that any of these would have been enough to complete the assignment, with neither seeming wildly different on pricing for what the assignment required, but, for peace of minds sake, I chose to utilise google cloud platform to avoid any unexpected costs that may arise if I configured something wrong.

Also, its worth noting that we had utilised google cloud platform in the labs before, and I knew that my account had been set up and worked fine, so it helped minimise any friction.

This definitely wouldn't be the approach I would take had this been a real project deployed in a business, where consideration of current and future use, along with patterns of use, would be the most sensible approach to take. E.g., if I'm working for a small online business that has seasonal web pages that only look to access certain databases for a small portion of the year, for example, they are a media company that does an awards programme once a year, I would consider looking

into pay-as-you go packages in that instance, as it makes little to no sense to pay a flat rate on a site that's not going to be visited consistently throughout the year. However, if I work at a business that drives consistent traffic through a web application, such as something like quickbooks, it makes more sense to look at pricing packages that include a subscription model, that can then end up in a resultant cost reduction over simple pay as you go packages. Obviously, the price packages over other vendors would be a driving force in this, but also the services and downtime they offer would need to be considered as a bigger picture.

Creating VM instance

The VM instance was created with the following configuration:

Name *
dsmith21435099 ?

Labels ?
[+ ADD LABELS](#)

Region *
europe-west2 (London) ?
Region is permanent

Zone *
europe-west2-c ?
Zone is permanent

Machine configuration


Machine family

[GENERAL-PURPOSE](#) [COMPUTE-OPTIMIZED](#) [MEMORY-OPTIMIZED](#) [GPU](#)

Machine types for common workloads, optimized for cost and flexibility


Series
E2 ▼
CPU platform selection based on availability

Machine type
e2-medium (2 vCPU, 4 GB memory) ▼



vCPU	Memory
1 shared core	4 GB

Boot disk ?

Name	dsmith21435099
Type	New balanced persistent disk
Size	10 GB
Image	 Ubuntu 18.04 LTS

[CHANGE](#)

Identity and API access ?

Service accounts ?

Service account

Compute Engine default service account

Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. [Learn more](#)

Access scopes ?

☐ Allow default access

☒ Allow full access to all Cloud APIs

☐ Set access for each API

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

☒ Allow HTTP traffic

☒ Allow HTTPS traffic

Some points to note – Europe-west was selected for latency reasons, with London being the closest available servers.

The Machine type could have been set to micro, but the pricing difference was so negligible for this assessment that it was left on medium.

The boot disk was set to Ubuntu 18.04 as per the assignment instructions.

Access was granted to all Cloud APIs and all HTTP traffic and HTTPS traffic. It's worth pointing out that after installation of apache, I had issues granting access to the manager app, so additional configuration of the firewall to allow default route was granted as below.

Firewall settings

Network
default
Priority
1000
Direction
Ingress
Action on match
Allow
Source filters
IP ranges 0.0.0.0/0
Protocols and ports
all
Enforcement
Enabled
Insights
None

Remainder of LAMP stack installation:

Obviously, Linux is already installed, so we just need Apache, MariaDB and PHP installation from this point.

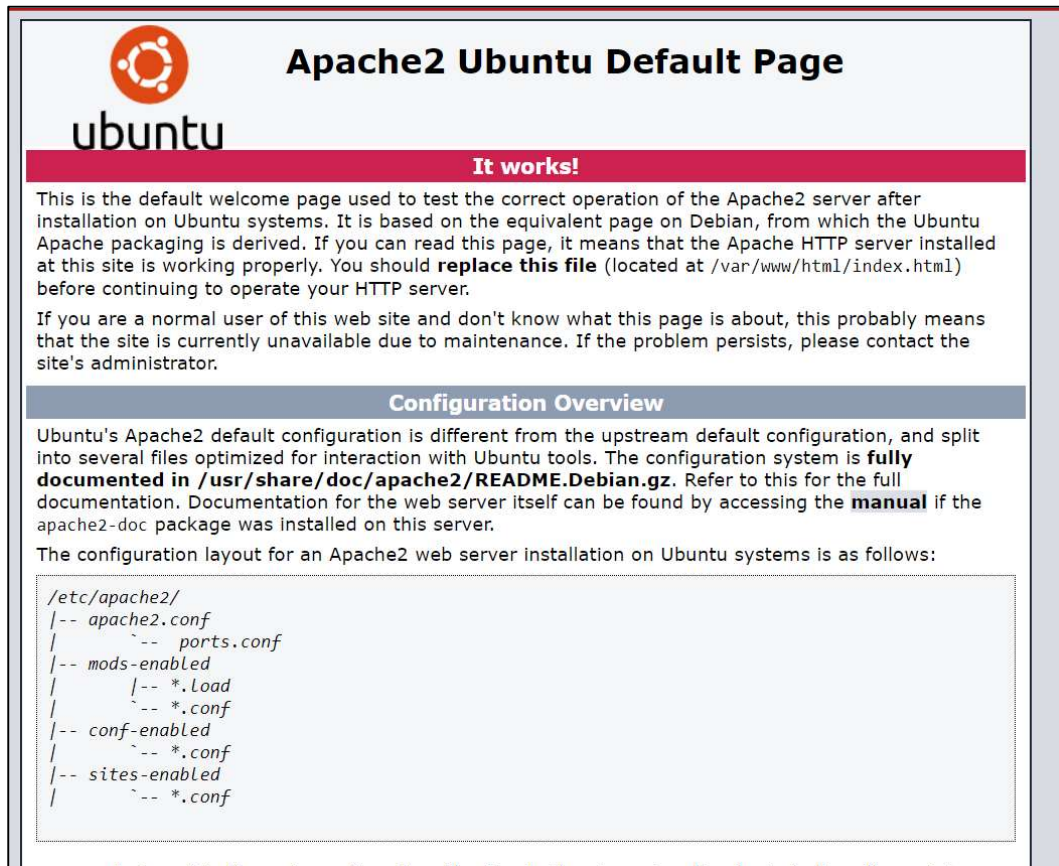
For this, I followed googles own guide on installation of a LAMP stack, finishing at the configuration of Mariadb, as nothing else in the document was relevant for the assignment.

The commands utilised were as follows:

```
sudo apt-get update
sudo apt-get install apache2 php libapache2-mod-php
sudo sh -c 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php'
sudo apt-get update
sudo apt-get install mariadb-server php php-mysql
sudo mysql_secure_installation – where the password for root was set as root
```


To test the installation, the IP of the VM instance is utilised as follows:

<http://34.142.117.246/> - this tests the apache installation, which provides the following confirmation page:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.


Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

<http://34.142.117.246/phpinfo.php> - this tests the php installation and provides the following confirmation page:

PHP Version 7.2.24-0ubuntu0.18.04.11	
	
System	Linux dsmith21435099 5.4.0-1072-gcp #77~18.04.1-Ubuntu SMP Wed Apr 13 07:49:40 UTC 2022 x86_64
Build Date	Mar 2 2022 17:52:35
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

To test the status of Mariadb, the command “sudo systemctl status mariadb” is entered into command line in the VM, producing the following result:

```
cm10d@dsmith21435099:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.1.48 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-05-31 10:51:41 UTC; 6s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 23396 (mysqld)
   Status: "Taking your SQL requests now..."
     Tasks: 27 (limit: 4662)
    CGroup: /system.slice/mariadb.service
            └─23396 /usr/sbin/mysqld

May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: information_schema
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: mysql
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: performance_schema
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: Phase 6/7: Checking and upgrading tables
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: Processing databases
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: information_schema
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: performance_schema
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: Phase 7/7: Running 'FLUSH PRIVILEGES'
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23432]: OK
May 31 10:51:41 dsmith21435099 /etc/mysql/debian-start[23508]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
```

Installing Java JDK

After the installation of the LAMP stack, the next step to deploying the WAR file, is to ensure that the Java is installed. The following command was used to install the JDK:

```
sudo apt install default-jdk -y
```

Where a check of the version shows the following installation:

```
cm10d@dsmith-21435099:~$ java --version
openjdk 11.0.15 2022-04-19
OpenJDK Runtime Environment (build 11.0.15+10-Ubuntu-0ubuntu0.18.04.1)
OpenJDK 64-Bit Server VM (build 11.0.15+10-Ubuntu-0ubuntu0.18.04.1, mixed mode, sharing)
```

Installing and configuring Tomcat

Finally, I installed a Java server platform, and for this assignment we were required to install Tomcat 9. After installation, tomcat needs to configuration, so I will cover that here too.

To start with, I created a group called tomcat and added a user called tomcat to that group. To do this, I used the following commands:

```
cm10d@dsmith-21435099:~$ sudo groupadd tomcat
```

```
cm10d@dsmith-21435099:/opt/tomcat$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

Next, I needed to download tomcat. To do this, I move to the tmp directory and use the curl command to download tomcat from the website. I used the following link:

<https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.63/bin/apache-tomcat-9.0.63.tar.gz>

Using the following commands:

```
cm10d@dsmith-21435099:~$ cd /tmp
cm10d@dsmith-21435099:/tmp$ sudo curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.63/bin/apache-tomcat-9.0.63.tar.gz
```

Next, I install tomcat in the /opt/tomcat directory using the following commands:

```
cm10d@dsmith-21435099:/tmp$ sudo mkdir /opt/tomcat
cm10d@dsmith-21435099:/tmp$ sudo tar xzvf apache-tomcat-9.0.63.tar.gz -C /opt/tomcat --strip-components=1
```

After that, I granted permissions to the tomcat user in the entire /opt/tomcat directory, as follows:

```
cm10d@dsmith-21435099:/opt/tomcat$ sudo chgrp -R tomcat /opt/tomcat
cm10d@dsmith-21435099:/opt/tomcat$ sudo chmod -R g+r conf
cm10d@dsmith-21435099:/opt/tomcat$ sudo chmod -R g+x conf
```

Then I grant ownership of the webapps, work, temp and logs directory to the tomcat user using:

```
cm10d@dsmith-21435099:/opt/tomcat$ sudo chown -R tomcat webapps/ work/ temp/ logs
```

Access in the directory now looks as follows:

```
drwxr-xr-x 9 root    tomcat  4096 May 31 12:16 .
drwxr-xr-x 3 root    root    4096 May 31 12:15 ..
-rw-r----- 1 root    tomcat 18986 May 11 07:52 BUILDING.txt
-rw-r----- 1 root    tomcat  6210 May 11 07:52 CONTRIBUTING.md
-rw-r----- 1 root    tomcat 57092 May 11 07:52 LICENSE
-rw-r----- 1 root    tomcat  2333 May 11 07:52 NOTICE
-rw-r----- 1 root    tomcat  3398 May 11 07:52 README.md
-rw-r----- 1 root    tomcat  6901 May 11 07:52 RELEASE-NOTES
-rw-r----- 1 root    tomcat 16505 May 11 07:52 RUNNING.txt
drwxr-x--- 2 root    tomcat  4096 May 31 12:16 bin
drwxr-x--- 2 root    tomcat  4096 May 11 07:52 conf
drwxr-x--- 2 root    tomcat  4096 May 31 12:16 lib
drwxr-x--- 2 tomcat tomcat  4096 May 11 07:52 logs
drwxr-x--- 2 tomcat tomcat  4096 May 31 12:16 temp
drwxr-x--- 7 tomcat tomcat  4096 May 11 07:52 webapps
drwxr-x--- 2 tomcat tomcat  4096 May 11 07:52 work
```

Next tomcat needs to know where Java is installed. To do this, I created the systemd service file. First, I opened a file called tomcat.services using the following command:

```
sudo nano /etc/systemd/system/tomcat.service
```

Next, the following information was written into the file:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking
Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

After that, the systemd daemon was reloaded and the tomcat server was started using the following commands:

```
cm10d@dsmith-21435099:/opt/tomcat$ sudo systemctl daemon-reload
cm10d@dsmith-21435099:/opt/tomcat$ sudo systemctl start tomcat
cm10d@dsmith-21435099:/opt/tomcat$ sudo systemctl status tomcat
```

Then the status of tomcat was checked to ensure it was running:

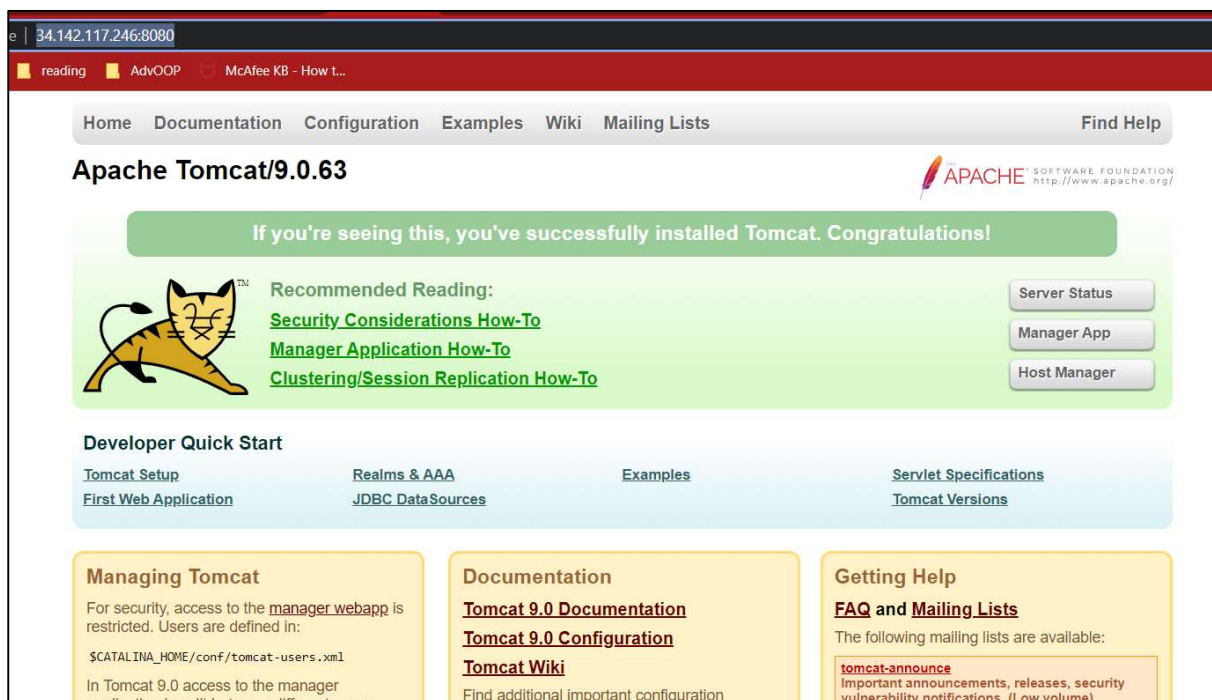
```
cm10d@dsmith-21435099:/opt/tomcat$ sudo systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-05-31 12:50:46 UTC; 13s ago
     Process: 18079 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
    Main PID: 18092 (java)
      Tasks: 29 (limit: 4662)
     CGroup: /system.slice/tomcat.service
            └─18092 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config

May 31 12:50:46 dsmith-21435099 systemd[1]: Starting Apache Tomcat Web Application Container.
May 31 12:50:46 dsmith-21435099 startup.sh[18079]: Tomcat started.
May 31 12:50:46 dsmith-21435099 systemd[1]: Started Apache Tomcat Web Application Container.
```

Next – as tomcat uses port 8080 for all conventional requests, it was opened using the following command:

```
cm10d@dsmith-21435099:/opt/tomcat$ sudo ufw allow 8080
```

The active status of the tomcat installation was then checked using <http://34.142.117.246:8080/> - giving a response page as follows:



Tomcat Web Management configuration:

After I had tomcat installed, I then had to configure the web management application. I.E when I pressed things like the Manager App button on the screen to go to upload the WAR, it actually worked.

To do this, I edited the users.xml file in tomcat using the command:

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

What I did when this file was opened, was to uncomment the start of the admin user and reopen the comment at the start of the robot user instead. I then added a password of root to the admin user, and the admin-gui role alongside the manager-gui one – as follows:

```
<tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  version="1.0">
<!--
  By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application.  If you wish to use this app,
  you must define such a user - the username and password are arbitrary.

  Built-in Tomcat manager roles:
    - manager-gui      - allows access to the HTML GUI and the status pages
    - manager-script   - allows access to the HTTP API and the status pages
    - manager-jmx      - allows access to the JMX proxy and the status pages
    - manager-status   - allows access to the status pages only

  The users below are wrapped in a comment and are therefore ignored.  If you
  wish to configure one or more of these users for use with the manager web
  application, do not forget to remove the <!-- .. --> that surrounds them.  You
  will also need to set the passwords to something appropriate.
-->

  <user username="admin" password="root" roles="manager-gui,admin-gui"/>
<!-- <user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
  The sample user and role entries below are intended for use with the
```

Next, I had problems because it still didn't appear to want to open the manager. I discovered that tomcat restricts access to manager and host-manager apps to connections coming from the server itself, so I had to configure the relevant XML files to stop that happening. To do that I used:

```
sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

and

```
sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
```

And then commented out the IP restrictions in both as follows:

```
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
  <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|S
</Context>
```

I understand this isn't the best approach for security purposes – but for this assignment, it would be sufficient.

Then tomcat was updated to reflect these changes, using;

`sudo systemctl restart tomcat`

Finally, this was checked using the following url <http://34.142.117.246:8080/manager/html> and entering the username admin and password root – showing the manager page and the WAR file was deployed.

Creating the MySQL DB:

In Google Cloud platform, the database was created with the following configuration:

Instance info

Instance ID *

dsmith-21435099

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password *

root

GENERATE

Set a password for the root user. [Learn more](#)

Hide password

☐ No password

Database version *

MySQL 8.0

Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region

europa-west2 (London)

The IP of the VM was added into the networking section of the database as follows:

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. [Learn more](#)

You can use the Cloud SQL Proxy for extra security with either option. [Learn more](#)

Instance IP assignment

☐ Private IP

Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. [Learn more](#)

☒ Public IP

Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

Authorized networks

You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)

cloudassignment (34.142.117.246)

ADD NETWORK

Next, the connection was established using DBeaver, and the SQL script was run using the assignment starter files, to produce the following data tables in the db:

Collections –

Table Name: collection

Engine: InnoDB

Auto Increment: 5

Charset: latin1

Collation: latin1_swedish_ci

Description:

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
ID	1	int	[v]	[v]	PRI	
Title	2	varchar(255)	[v]	[]		'0'
Genre	3	varchar(50)	[v]	[]		'0'
Year	4	int	[v]	[]		0

ID	Title	Genre	Year
1	Bad Boys 3	Action	2,019
3	Batman - Dark Knight Rises	Action	2,011
4	Mission Impossible 4	Action	2,018

Users –

Table Name: users

Engine: InnoDB

Auto Increment: 3

Charset: latin1

Collation: latin1_swedish_ci

Description:

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
id	1	int	[v]	[v]	PRI	
username	2	varchar(50)	[v]	[]		
password	3	varchar(50)	[v]	[]		
apikey	4	varchar(10)	[v]	[]		

The screenshot shows the DBeaver interface with a database connection to 35.242.140.18. The left sidebar shows the database structure: Databases > dvddb > Tables > users. The main pane displays a table with the following data:

id	username	password	apikey
1	kaleem	1234	abc123
2	admin	1234	FbMTh0pDq

Configuring the WAR and the database:

Starting with the url <http://34.142.117.246:8080/dvdWebApp/dbconfig> - the following information was input:

The screenshot shows a web form titled "Select database type:" with the following fields and values:

- Select database type:** ☒ localhost ☐ AWS RDS ☒ Google Cloud SQL ☐ Azure
- Database IP/End point:** 35.242.140.18
- Instance connection name (Google Cloud SQL Only):** silent-vim-351410:europa-west2:dsmith-21435099
- Database Name:** dvddb
- Database Username:** root
- Database Password:** root

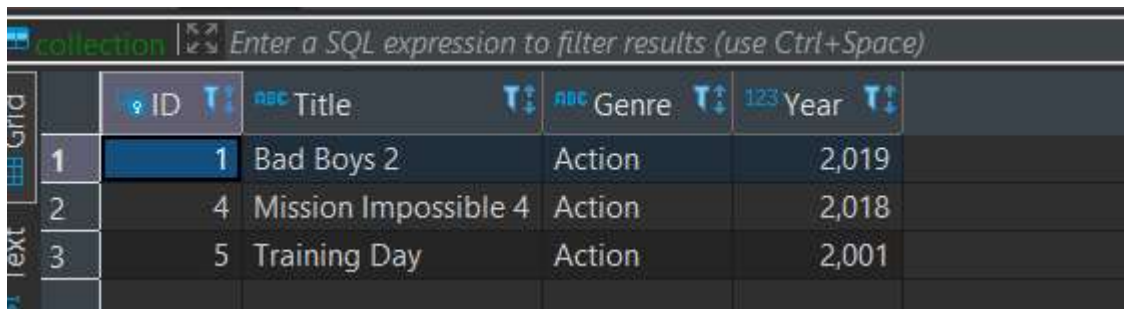
A "Save" button is located at the bottom right of the form.

The webapp loaded on the following URL: <http://34.142.117.246:8080/dvdWebApp/home>

I will leave this live for a number of weeks after submission so you can test/see it yourself.

All function was tested, such as logging in using usernames in the user table which was successful.

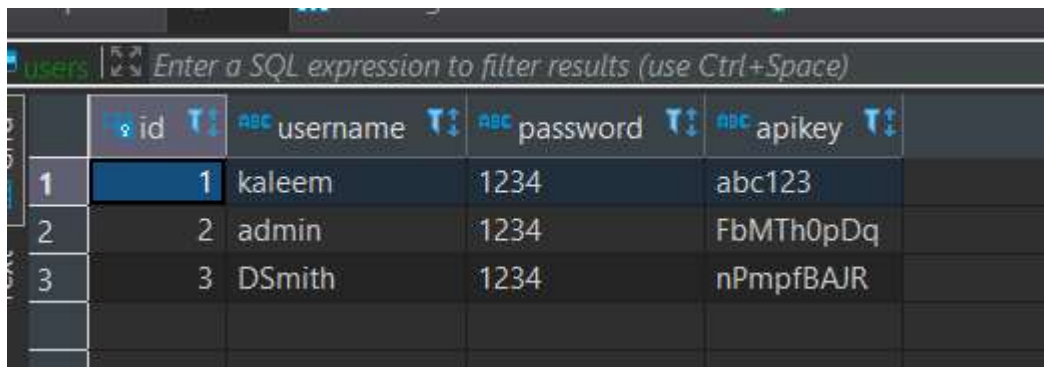
Editing Bad Boys 3 to Bad Boys 2 was successful, deleting The Dark knight rises and adding training day all worked as can be seen in DBeaver collections table:



collection Enter a SQL expression to filter results (use Ctrl+Space)

	ID	Title	Genre	Year
1	1	Bad Boys 2	Action	2,019
2	4	Mission Impossible 4	Action	2,018
3	5	Training Day	Action	2,001

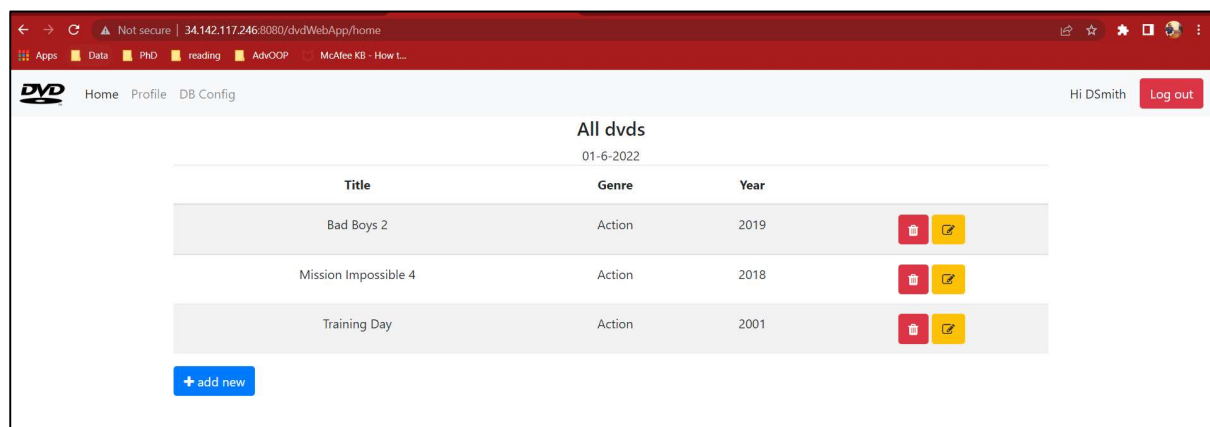
Finally, adding a new user worked as can be seen in the user table:



users Enter a SQL expression to filter results (use Ctrl+Space)

	id	username	password	apikey
1	1	kaleem	1234	abc123
2	2	admin	1234	FbMTh0pDq
3	3	DSmith	1234	nPmpfBAJR

Lastly, an image in the webapp – showing me logged in as a newly created user, with the changes implemented as above:



Not secure | 34.142.117.246:8080/dvdWebApp/home

Apps Data PhD reading AdvOOP McAfee KB - How t...

DVD Home Profile DB Config Hi DSmith Log out

All dvds
01-6-2022

Title	Genre	Year
Bad Boys 2	Action	2019
Mission Impossible 4	Action	2018
Training Day	Action	2001

+ add new

References:

- [1]: <https://www.ibm.com/cloud/learn/cloud-computing>
- [2]: <https://azure.microsoft.com/en-gb/overview/what-is-saas/>
- [3]: <https://azure.microsoft.com/en-gb/overview/what-is-paas/>
- [4]: Week 1 lecture notes
- [5]: <https://cloud.google.com/learn/what-is-iaas#section-1>
- [6]: <https://www.dell.com/en-uk/shop/intel-xeon-gold-6248-25ghz-twenty-core-processor-20c-40t-104gt-s-275m-cache-turbo-ht-150w-ddr4-2933/apd/338-brvk/processors>

[7]: <https://www.techbuyer.com/uk/840758-191-samsung-32gb-1x32gb-pc4-20800v-2rx4-server-memory-106418>

[8]: https://www.buykingston.co.uk/samsung-pm1725b-25-12800-gb-pci-express-30-v-nand-nvme?gclid=Cj0KCQjw-daUBhCIARIsALbkjSamRfW3jchi6wnPUfsIbPnMolHZaKuFJ7XNhbCnuFTkBNm54iS7lkoaAmT0EALw_wcB

[9]: <https://www.cablemonkey.co.uk/apc-netshelter-cx-sound-proof-cabinets/6744-apc-netshelter-cx-38u-0731304300298.html>