

Matthew Cooze 1911017 ACCA7027 Cloud Engineering Principles and Practice coursework

by Matthew Cooze

FILE	18625_MATTHEW_COOZE_MATTHEW_COOZE_1911017_ACCA7027_CLOUD_ENGINEERING_PRINCIPLES_AND_PRACTICE_COURSEWORK_343871_1506701715.PDF (373.69K)		
TIME SUBMITTED	09-MAY-2021 11:49PM (UTC+0100)	WORD COUNT	1876
SUBMISSION ID	152394771	CHARACTER COUNT	10272

ACCA7027 Cloud Engineering Principles and Practice

Matthew Cooze 1911017

May 9, 2021

Contents

1	introduction	2
2	Importance of containerisation for running applications efficiently in the cloud	3
2.1	The benefits of containerisation	3
2.1.1	Portability	3
3	code bellow sorry for being in middle Latex played up with code	5
4	Kubernetes	13
4.1	Kubernetes is an example of orchestration software that is used to execute containers	13
4.2	build a networked service application of your choice using an appropriate controller and service with YAML	14
4.3	references	14

Chapter 1

introduction

The goal of this assignment is to write a report showing the importance of containerisation within the cloud.

Chapter 2

Importance of containerisation for running applications efficiently in the cloud .

Docker is a containerisation environment for running applications efficiently in the cloud.

containerisation is the act of virtualising the Operating System(OS) and then the application is run inside this virtual System, this allows for much greater flexibility and compatibility allowing for the application to be run on any software or server even if normally it would never run that software.

due to how containerisation is designed to work on any computer, there are some very strict limitations in place for programming the applications. for example YAML forces people to use 2 spaces, tabs are banned and if the wrong number of spaces enter the entire application will not work.

2.1 The benefits of containerisation

2.1.1 Portability

containerisation works by having everything needed to run the application bundled together, this means all code, libraries, dependencies and even the operating system,[1] this makes containers faster and quicker to add new resources when needed on a micro-service meaning there is no need to copy the entire system

to improve performance just add more of the struggling components.

as previously mentioned containerisation allows for the running of applications designed for it e.g YAML, docker etc. to be run on almost any systems.e.g computers,servers, the cloud.

this makes it great for portability as the file(application) can be kept on a hard-drive or on a server and loaded and run on any computers with docker or other software for running containerisation applications e.g kubernetes.

there are also some issues with this method if the server is down at time of access and and you are not running a kubernetes cluster you will loose access to the information for the duration the server is down the same would apply if the hard-drive was lost.

due to micro-service offered containerisation is becoming popular with companies as they can rent the needed server space and scale it up and down as needed, this means they are not paying for servers or services they do not need. this means there are no servers costing money but not being used.

due to the portability and flexibility provided by containers, they are perfect for the cloud where applications might need to be run anywhere on the world on any hardware, without containerisation the code might have to be changed slightly or paths updated for all new instances where with containers it can be run Hassell free once the application is made and working as intended,.

Chapter 3

code bellow sorry for being
in middle Latex played up
with code

\paragraph{ please see nginc being constructed into an image}

PS

E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker build -t godofcrafting/my-nginx .

[+] Building 0.1s (6/6) FINISHED

=> [internal] load build definition from Dockerfile

0.0s

=> => transferring dockerfile: 157B

0.0s

=> [internal] load .dockerignore

0.0s

=> => transferring context: 2B

0.0s

=> [internal] load metadata for docker.io/library/ubuntu:latest

0.0s

=> [1/2] FROM docker.io/library/ubuntu

0.0s

=> CACHED [2/2] RUN apt-get update && apt-get install -y nginx

0.0s

=> exporting to image

0.0s

=> => exporting layers

0.0s

=> => writing image

sha256:05b21ad8344c7255f12ab2cf1ba92d4fc2ca9826227b6202aba310ab3efe771d

0.0s

=> => naming to docker.io/godofcrafting/my-nginx

0.0s

PS

E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx>

\subsection{Publishing images}

publishing images is very simple once you have had enough practice, there are two main ways to make a dockerfile image one is using a plain-text file e.g notepad and then saving without the .txt at the end instead putting .yaml these can then be run by docker. the other way is to use a ready made container change the application as it runs and then saving the results as a new docker container. {2}

\paragraph{please see pushing an image to dockerhub below }

all code to push to my account on dockerhub

PS

```
E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker build -t godofcrafting/my-nginx .
```

```
[+] Building 0.1s (6/6) FINISHED
```

```
=> [internal] load build definition from Dockerfile
```

```
0.0s
```

```
=> => transferring dockerfile: 157B
```

```
0.0s
```

```
=> [internal] load .dockerignore
```

```
0.0s
```

```
=> => transferring context: 2B
```

```
0.0s
```

```
=> [internal] load metadata for docker.io/library/ubuntu:latest
```

```
0.0s
```

```
=> [1/2] FROM docker.io/library/ubuntu
```

```
0.0s
```

```
=> CACHED [2/2] RUN apt-get update && apt-get install -y nginx
```

```
0.0s
```

```
=> exporting to image
```

```
0.0s
```

```
=> => exporting layers
```

```
0.0s
```

```
=> => writing image
```

```
sha256:05b21ad8344c7255f12ab2cf1ba92d4fc2ca9826227b6202aba310ab3efe771d
```

```
0.0s
```

```
=> => naming to docker.io/godofcrafting/my-nginx
```

```
0.0s
```

PS

```
E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker push godofcrafting/my-nginx
```

Using default tag: latest

The push refers to repository [docker.io/godofcrafting/my-nginx]

a1085005f925: Layer already exists

02473afd360b: Layer already exists

dbf2c0f42a39: Layer already exists

9f32931c9d28: Layer already exists

latest: digest:

sha256:a7cc9fab6fc2a0e250781aaeb8d57a4c13e2cfdcbb97e929f894ace2b0fd9 size: 1155

PS

```
E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx>
```

\includegraphics[scale=0.5]{ACCA7027_Cloud_Engineering_Principles_and_Practice/coursework/powershell commands pushing}

\subsection{Container volumes}

a docker volume is how docker likes to save its data, this is due to it being easier to backup and move compared to binding a save location in place, another advantage of this is data can be moved from one application to another as long as they use the same data due to it not being buried in the OS, you can bind the data to a place easy to locate e.g C or D drive and have the container working in the E drive. [4]

the -v is telling where to mount the docker web html onto the original source location of the nginx html file

PS

```
E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker run -d -P -v ~/docker/web:/usr/share/nginx/html --hostname nginx-container nginx
```

\subsection{Container networking}

container networking allows you to remote into running containers, and see how it would look like when live on the internet for checking the website/application runs as intended and for testing for any bugs. it is allows for connecting many different containers running different applications and connect them securely.

\subsection{Creating your own application}

please see the image attached below of a simple docker image (website), talking about 5 a side football venue coming soon and a cute cat. \vspace{0.5em}

the following is what the website should of looked like

\includegraphics[scale=0.5]{ACCA7027_Cloud_Engineering_Principles_and_Practice/coursework/what website should of looked like} \vspace{0.5em}

everytime i executed the command to save the HTML as the nginx.html is broke \vspace{0.5em}

PS

E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker run -d -P -v ~/docker/web:/usr/share/nginx/html --hostname nginx-container nginx

please see the code used to produce the website

\paragraph{please see powershell commands to build the my-nginx}

PS

E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker image build -t my-ngminx .

[+] Building 0.1s (6/6) FINISHED

=> [internal] load build definition from Dockerfile
0.0s

=> => transferring dockerfile: 31B
0.0s

=> [internal] load .dockerignore
0.0s

=> => transferring context: 2B
0.0s

=> [internal] load metadata for docker.io/library/ubuntu:latest
0.0s

=> [1/2] FROM docker.io/library/ubuntu
0.0s

=> CACHED [2/2] RUN apt-get update && apt-get install -y nginx
0.0s

=> exporting to image
0.0s

=> => exporting layers
0.0s

=> => writing image
sha256:7274ee05724079f1829667605a2972e6d064b90e009fec941d578c5f8704baab
0.0s

=> => naming to docker.io/library/my-ngminx
0.0s

PS

E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx>

\paragraph{please see index.html code below }

```

<!DOCTYPE html>

<html>

<head>

<title>Hello page</title>

<style>
img {
    width: 100%;
}
</style>
</head>

<body>

<h2>
5 a side football club soon opening
</H2>



<br>
credits in referances for photos

</body>

</html>

```

```
\paragraph{please see dockerfile code below }
```

```
FROM ubuntu
```

```
RUN apt-get update && apt-get install -y nginx
```

```
EXPOSE 80
```

```
ENTRYPOINT ["nginx","-g","daemon off;"]
```

```
Administrator: Windows PowerShell
PS E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker build -t godofcrafting/my-nginx .
[+] Building 0-1s (6/6) FINISHED
    docker.io/godofcrafting/my-nginx:latest
PS E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx> docker push godofcrafting/my-nginx
Using default tag: latest
The push refers to repository [docker.io/godofcrafting/my-nginx]
a10b5005f925: Layer already exists
02473a4d3600: Layer already exists
0bf2c0f42a39: Layer already exists
9f32931c9d28: Layer already exists
latest: digest: sha256:a7cc9fabcf2a0e250781aaeb8d57a4c13e2cfdcbbeb97e929f894ace2b0fd9 size: 1155
PS E:\university\postgraduate\ACCA7027_Cloud_Engineering_Principles_and_Practice\coursework\my-nginx>
```

5 a side football club soon opening



website did not work this is what it was supposed to look like credits in references for photos

Chapter 4

Kubernetes

4.1 Kubernetes is an example of orchestration software that is used to execute containers.

Critically evaluate the use of Kubernetes controllers and services in managing containers to produce networked applications. (Marks 25)

Kubernetes controllers are a brilliant method of being able to scale up an application massively at the drop of a hat, due to the main controller(job controller) managing the cluster (API's) application programming interfaces, if a pod goes down more can be up and running immediately. [5]

the job controller can manage the API controller to make more pods to run when needed and shut down when no longer needed, this means when a task comes in to the job controller it distributes it to the API that creates as many pods are needed to run the task and no more.

the security on the Kubernetes pods works in a few different ways: one is no one really needs to know dedicated IP addresses adding anonymity to each pod controller and making it more difficult for hackers to get access to. another great security measures is each open IP address must be made either in the YAML code or programmed in to the controller after it has started running.

another great service is the pods being recreated as internet access is lost or they go down meaning the company does not need to know about the one pod playing up as a new one will take its place and keep all systems running.

another brilliant security feature is allowing one container to bridge into another, allowing them to talk securely whilst still being closed to other IP addresses.

4.2 build a networked service application of your choice using an appropriate controller and service with YAML

4.3 references

using Textstudio please download pdf as links are inbuilt into references

- [1] IBM Cloud Team, IBM Cloud, Everything you need to know to understand the containers vs. VMs debate and why containers are growing in popularity. 9 April 2021 IBM Cloud Team, IBM Cloud
 - [2] docker Docker overview 2013-2021 docker inc
 - [3] w3school HTML Images 1999-2021 w3school
 - [4] docker Use volumes 2013-2021 docker inc
 - [5] kubernetes Controllers 2021 kubernetes
- cat 2wikipedia File:Cute-kittens-12929201-1600-1200.jpg

Matthew Cooze 1911017 ACCA7027 Cloud Engineering Principles and Practice coursework

GRADEMARK REPORT

FINAL GRADE

21 / 100

GENERAL COMMENTS

Instructor

Unfortunately this is a long way from complete. A partial implementation (including the log rather than the yaml) and a report which is far too light on the detail and a significant amount under the expected word count.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15