Project Terms of Reference — To compare Virtualisation and Containerisation for the context of hosting headless servers.

James Poxon

31st October 2020

1 Background

Virtualisation has been utilised by a number of industries for a long time now, with first iterations of virtual machines dating back to IBM in the 1960s [Pugh, 1995]. System Virtual machines allow an operating system to emulate the function of a full operating system layered on top of a base operating system. Functionally, this allows multiple different logical 'computers' with varying operating systems to run on one physical computer. In most modern implementations, virtualisation requires software (known as a hypervisor) to manage and create the virtual machines.

Whilst I was on a year-long placement provided as part of a sandwich course at my university, I had the chance to work in an IT risk department at a reputable enterprise in Newcastle Upon Tyne. whilst working there I witnessed first hand, infrastructure and operations departments using virtualisation for much of the internally and externally facing server infrastructure, in what was an unwieldy and cumbrous use of scripts to update and install patches and dependencies across a multitude of systems. This resulted in a number of incidents where systems had to be pulled down in order to update the Operating Systems of individual virtual machines. These methods often caused unnecessary down time for systems, resulting in substantial risk for the business. Furthermore, this method often put a substantial stress on the hardware of these systems, with hardware boxes often running at full resource potential under heavy load, and whilst these boxes were designed to withstand these kinds of loads, it still affected the longevity of the hardware.

In recent years there has been research into the use of containers instead of virtual machines for various operations across computing industries [Watada et al., 2019]. Containers are different from virtual machines in that they run on the base OS, and don't require a secondary emulated operating system to be managed by a hypervisor. This is a benefit as it reduces the resources [Joy, 2015] used by each instance. Overall, containers are a more lightweight and efficient system, that are easier to keep up to date. Whilst research has claim in displaying the benefits of containerisation, much of the server infrastructure of enterprise remains reliant on Virtualisation, not Containerisation.

I have also had the opportunity to partake in gatherings held by CoTech, a network of tech co-operatives that meet semi-regularly to discuss various tech related topics. Whilst I was there, it was discussed that a large portion of these small enterprises used Docker, a system for packaging and deploying containers, to develop applications for their clients.

There is historical research that compares virtualisation and containerisation for other technical applications, for example [Dua et al., 2014] compares the two methods inside the context of PaaS (Platform as a service), which is similar to co-techs use of Docker. It seems that the application for containerisation has been realised as early as 2014 when it is being applied to the development and hosting of applications, but not as much when talking about the running of wider server infrastructure. This is further supported by research from '451 Research' who in 2017 estimated a compound annual growth rate of 40% for containers in 2020 [Research, 2017], suggesting a coming paradigm shift from virtualisation to containerisation.

2 Proposed Work

I plan to do similar work to the studies I have already mentioned [Joy, 2015], [Dua et al., 2014], but instead focus this work on my own context and my own interest in Operating Systems and Server Infrastructure. I have experience with running headless servers on virtual machines already, through the Advanced Operating Systems module I did in my second year of study at Northumbria University.

It is important to set a baseline system here, and to best reflect a realistic topology, I will use LAMP (Linux + Apache + MySQL + PHP). I have experience with Ubuntu server, so I will use this version of Linux as the base operating system for my virtual machines, and plan to host a full working topology of servers, including a DNS architecture, a web-server architecture that includes HTTP (Apache) servers, NFS servers and MySQL servers. The reasoning for doing this is to create as realistic a topology as possible, and to ensure a somewhat realistic level of network traffic so that meaningful data can be captured.

I will also need to set a standard for measurement to ensure that my data is meaningful and uses the scientific method. I plant to use tools such as iPerf3 (for network performance) and Sysbench (for hardware performance) across a number of servers. It is important to measure both network and hardware performance (CPU, Memory, Disk, etc) to ensure that described benefits are achieved for the system as a whole. iPerf is an industry standard tool for measuring network performance on Linux systems, whilst Sysbench is a full benchmarking suite for linux that will let me benchmark the CPU, file IO, and importantly, MySQL performance (allowing direct measurement of database performance on the machine that will be hosting the previously mentioned MySQL database. Whilst I have mentioned these benchmarking tools, it is possible that they could have problems with integrating with certain applications or environments, in this case, there is a number of other standard benchmark utilities (Phoronix Test Suite, KDiskMark, UnixBench, etc) available that should give me the flexibility to create measurements. Whilst these utilities all have differing overheads within them (meaning they will use up varying resources to run the benchmark), as long as I use the same benchmark across the same machines I am comparing, this overhead shouldn't effect the measurable

performance difference between these machines.

3 Aims and Objectives

3.1 Aims

To compare and contrast the performance difference between using virtualisation or containerisation for running headless servers.

3.2 Objectives

Your objective list is a series of measurable objectives, can you tick each one off as *done*? I usually expect between 8 and 12 objectives

The **enumerate** environment is useful here for generating a numbered list. You can put commands in with a keyword and then refer to the label with a command, it puts the number of the objective in the text

See objective 1

1. **Classify the problem domain** this is where you develop an understanding of the nature of the problem/project
2. **Identify Techniques to solve** What Algorithms are you to use, how is a database structured,
3. **Select tools to use** What languages, software, hardware; are you using?
4. **Design the system to be build** Its requirements, the **test plan**, the architecture (Layer model/Model-View-Controller)
5. **Build the system** I'd include testing here, as the result is a *working wywtem*

4 Skills

This is where you can cover the skills you have relevant to the project and the new skills you are going to acquire during the project.

1. Programming in C, see module KFxxx
2. Hardware Design

5 Resources

This is an important section, it lists the hardware and software you are going to need for the project.

5.1 Hardware

For Hardware this is more critical, as we need to identify any hardware we have, or that you are going to buy. We do have an ordering mechanism in the Department, but time and budget are critical constraints here.

5.2 Software

In the case of software, there isn't usually an issue, unless you're needing huge amounts of run-time (we don't have a super-computer handy).

6 Structure and Contents of the Report

Here you set out the likely chapters you will have in your report. Usually each objective lends itself to one or more chapters. You can refer back to the objectives set.

6.1 Report Structure

Introduction Sets out the background and motivation for the project. Summarises the work done, the results, the conclusions, and the recommendations for future work. It is a one chapter summary of the *entire* project.

Defining the problem Objective 1 requires a precise definition of the problem you are solving. Don't forget to reference good source material See section 2.

Possible Solutions Discuss the possible solutions, compare the alternatives, and select the one to use for the implementation.

6.2 List of Appendices

What Appendices you will include. A copy of the TOR should be the first, followed by the Ethics form and the Risk Assessment.

Others might include design documentation, code listings, tables of results (if too large to include in the main text).

7 Marking Scheme

The marking scheme sets out what criteria we are going to use for the project.

Project Type General Computing or Software Engineering projects

Project Report State which chapters constitute the *Analysis*, the *Synthesis*, and the *Evaluation*. This help me when marking to know when to stop reading one section and put a mark down for it.

Product List the deliverables that make up the *Product*. Code, design, requirements specifications, test plans, etc.

For the *Fitness for Purpose* and *Build Quality* list the critera used to asses the product by

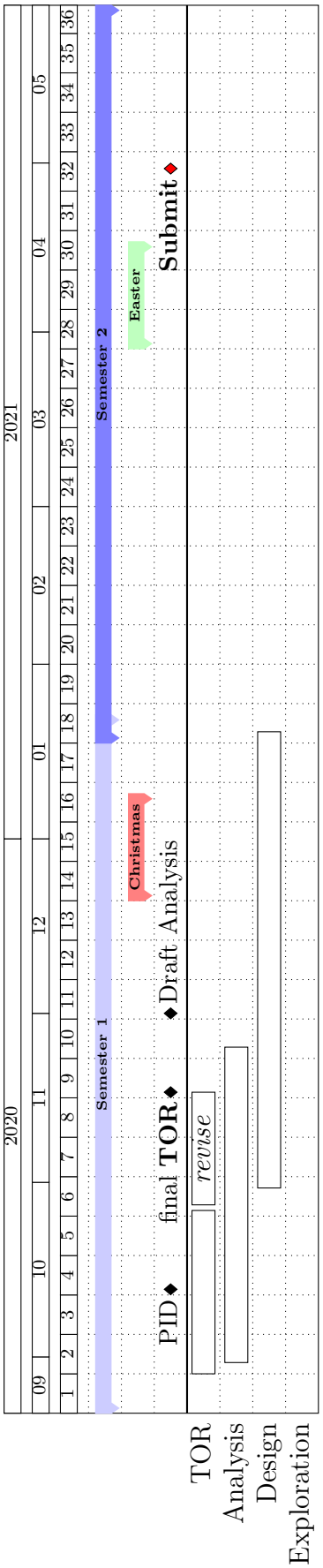
Fitness for Purpose

- meet requirements identified
- other appropriate measures

Build Quality

- Requirements specification and analysis
- Design Specification
- Code quality
- Test plan and Results

8 Project Plan



9 Bibliography

R. Dua, A. R. Raja, and D. Kakadia. Virtualization vs containerization to support paas. In *2014 IEEE International Conference on Cloud Engineering*, pages 610–614, 2014. doi: 10.1109/IC2E.2014.41.

A. M. Joy. Performance comparison between linux containers and virtual machines. In *2015 International Conference on Advances in Computer Engineering and Applications*, pages 342–346, 2015. doi: 10.1109/ICACEA.2015.7164727.

Emerson Pugh. *Building IBM : shaping an industry and its technology*. MIT Press, Cambridge, Mass, 1995. ISBN 9780262161473.

451 Research. Application containers will be a $2.7bn market by 2020, 2017. URL `https://451research.com/images/Marketing/press_releases/Application-container-market-will-reach-2-7bn-in-2020_final_graphic.pdf`.

J. Watada, A. Roy, R. Kadikar, H. Pham, and B. Xu. Emerging trends, techniques and open issues of containerization: A review. *IEEE Access*, 7:152443–152472, 2019. doi: 10.1109/ACCESS.2019.2945930.

A Ethics Form

If you scan the Ethics form on one of the multifunction printers, you can get a pdf copy. This can then be included with the LaTeX command

Assuming of course you have saved the form as `ethics.pdf`

Ethical category of project *[Complete after approval]*	
Red	☐
Amber	☐
Green	☐

Department of Computer Science and Digital Technologies

UNDERGRADUATE COMPUTING PROJECTS: ETHICS REGISTRATION AND APPROVAL FORM

Section One: Registration *[To be completed by student]*

Title of research project/dissertation	
Researcher's name	
Programme of study	
Academic Year	
Module code	
Supervisor's name	
Second marker's name	
Start Date of Project	

Brief outline of research topic:

Specimen

Short description of proposed research methods including identification of participants:

Ethical considerations in the research project	YES	NO
1. Does your research involve an external organisation or partner?	☐	☐
2. Does your research involve human participants?	☐	☐
3. If yes to Q.2, will you inform the participants about the research?	☐	☐
4. Will you obtain their consent using the standard consent form?	☐	☐
5. Is any deception involved?	☐	☐
6. Do any participants constitute a 'vulnerable group'? (refer to definition of Vulnerable People)	☐	☐
7. Will the research involve the following information?		
Commercially sensitive	☐	☐
Personally sensitive	☐	☐
Politically sensitive	☐	☐
Legally sensitive	☐	☐
8. Is the research likely to have any significant environmental impacts?	☐	☐
9. Are there likely to be any risks for the participants in your research?	☐	☐
10. Are there likely to be any risks for you in conducting the research?	☐	☐
11. If yes [to 5, 6, 7, 8, 9 or 10 above] have you identified steps to address the issues and mitigate any risks to participants, yourself or the environment?	☐	☐

Specimen

Statement to explain how any issues identified above will be addressed and what steps will be taken to mitigate such risks or adverse impacts

Ethical category of research project
Based on the above Ethical Considerations and with reference to the University's Ethical Scrutiny Risk Assessment tool identify the Ethical category of your research project (refer to http://www.northumbria.ac.uk/static/5007/respdf/riskassesmenttool for further guidance):

[Please tick as appropriate]

Red	☐	vulnerable participants; human tissue; sensitive data; risks to participants & researchers etc.
Amber	☐	human participants requiring informed consent; commercially sensitive information etc.
Green	☐	no participants involved; secondary data only; no sensitive data

I have read the University and the Faculty Ethics Policy and Procedures and confirm that the answers I have given above are correct. Where issues arise under items 5, 6, 7, 8, 9 or 10 [above] I have described in writing how I intend to approach these issues in the research.

Researcher's signature .....

Date .....

Section 1 Ethics Registration to be submitted to Principal Supervisor or Module Tutor and allocated to a reviewer as follows:

Green risk - may be approved by Supervisor
Amber risk - to be submitted for approval by one independent reviewer (second marker)
Red risk - to be submitted for approval by two independent members of Faculty Research Ethics Committee

Specimen

B Risk Assessment Form

Likewise you can scan and include the Risk Assessment Form