**Group Research Report**

**Running Apache Web Server as container (with persistent mounting)**

Declaration of Non-Plagiarism (Appendix A)

|  |
| --- |
| **Server Administration and Security (CIML010)**  **Group Research Report**  Submitted by: <Matric Number> Norman Dzulkarnaen Bin Omar  Submitted by: <Matric Number> Syahrul Afiq Bin Suran  Submitted by: <Matric Number> Desmond Koh Ti Yong  Submitted by: <Matric Number> Chong Nyuk Thong  Submitted by: 5387244Q Chan Yong Hoow  Submitted by: <Matric Number> Chua Boon Kheng  Date: 21/02/2025  “By submitting this work, we are declaring that we are the originators of this work and that all other original sources used in this work has been appropriately acknowledged.  We understand that plagiarism is the act of taking and using the whole or any part of another person’s work and/or the work generated by Artificial Intelligence presenting it as our own without proper acknowledgement.  We also understand that plagiarism is an academic offence and that disciplinary action will be taken if I am found to have committed or abetted the offence of plagiarism.”  Name and signature of Student: Norman Dzulkarnaen Bin Omar  Name and signature of Student: Syahrul Afiq Bin Suran  Name and signature of Student: Desmond Koh Ti Yong  Name and signature of Student: Chong Nyuk Thong  Name and signature of Student: Chan Yong Hoow  Name and signature of Student: Chua Boon Kheng |

|  |
| --- |
| **Declaration on the use of Generative AI tools for assignments**  Describe how you have used Generative AI tools such as ChatGPT or Dell.E-2 in your assignment.  Show snapshots of the conversations with the AI tool (i.e., the prompts you used and the response you get from the AI tool).  How to indicate the reference?  The content generated by AI tools is not retrievable except by the user who generated them, so they are considered non-recoverable sources. For non-recoverable sources:   * Do not include in the reference list * Cite within the text as personal communications or correspondences   Based on APA 7th edition referencing format  (Communicator, personal communication, Month Day, Year)  E.g. (Paraphrase from OpenAI’s ChatGPT AI language model, personal communication, March 9, 2023).  Important Note:   * Do not copy answers produced by the AI tool in totality as it is considered plagiarism. * Do not rely on any information produced by the AI tool blindly. You should always verify the answer with other sources. Do not assume that these answers provided by the AI tool are correct. * To achieve quality outputs from the AI tool, you should provide good prompts that are clear and specific. Be precise and provide context. Avoid making open-ended questions. |

**Introduction**

The age of virtualization has done a lot for allowing system admin and developers to build and experiment from their computers as if they were different systems. We can run a Linux virtual machine from our Windows PC to test the functionality of a service like Apache Web Server running on a Linux OS. It is far cheaper and easier to run a virtual machine than it is to buy a whole new computer.

**Problems existing with Virtual Machines**

However, there are still problems which Virtual machines may not solve. For example:

* Compatibility of each service with the libraries and dependencies of OS (One service requires version X of OS library. Another service – version Y of same library)
* Every time version of any service updates, you might need to recheck compatibilities with underlying OS infrastructure
* Virtual machines OS may take up large drive memory space and CPU computing resources.

**What are containers?**

Containers are efficient with their consumption of resources, such as CPU computing power and memory, compared to deploying separate virtual machines. They also offer design advantages both in how they isolate applications and how they abstract the application layer away from the infrastructure layer.

**What is Docker?**

Docker is an open-source tool designed to simplify the process of creating, managing, and deploying containers. Launched in 2013, Docker has rapidly become the go-to solution for containerization due to its ease of use, community support, and powerful ecosystem of tools.

**Benefits / Problem Solved by Container Technology**

Containers’ popularity stems from their ability to solve a variety of challenges developers face today:

Consistency Across Environments: Developers can "build once, run anywhere," ensuring the same application works the same way in different environments, from local development to production.

Speed: Containers are fast to start and stop, making them ideal for testing and deploying pipelines.

Efficient Use of Resources: Since containers share the host system's resources more effectively than virtual machines, they reduce overhead and allow for greater density in deployments.

Version Control for Your Applications: Containers allow you to version control not only your code but also the environment in which your code runs. This is particularly useful for rolling back to previous versions or debugging issues in production.

**Installation, Configuration and Test Scenarios**

**Installing Docker on our VM**

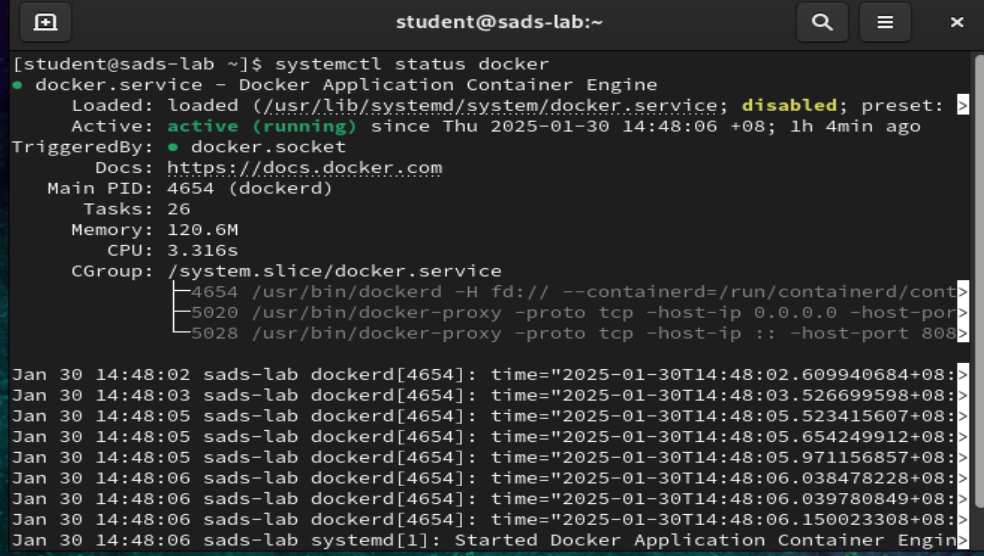
First, start our VM. Install Docker using the following curl command, which will download and run a shell script that will add the Docker repository to our system and install the package.

|  |
| --- |
| curl -fsSL https://get.docker.com | sh |

Next, use the systemctl command to start the Docker service, enable it to start during reboots, and check its status.

|  |
| --- |
| systemctl start docker  systemctl enable docker  systemctl status docker |

You should see the Docker service running in your VM.

****

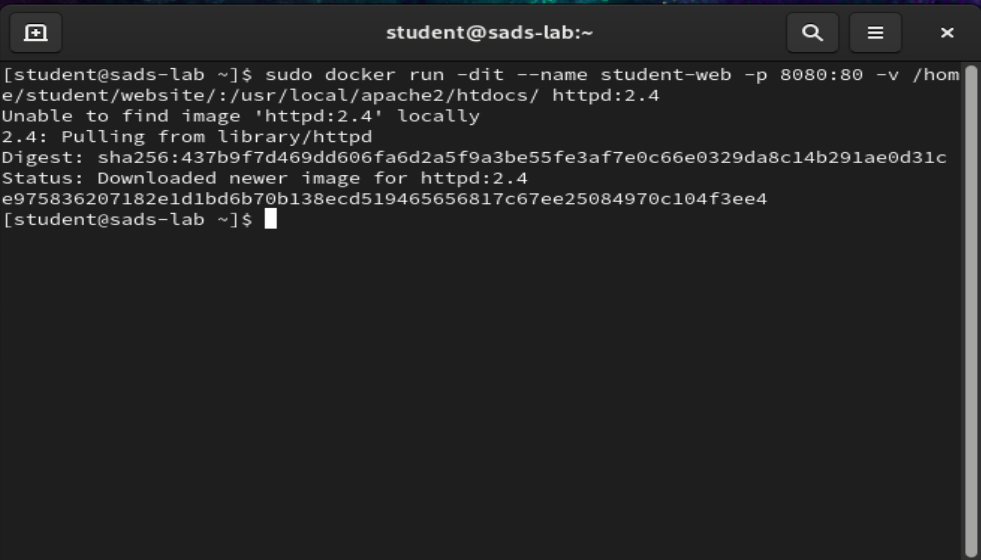
**Setting Up an Apache Container**

Next, we will run an **Apache 2.4** container and name it “**student-web**”, from the terminal. We will get an image called **httpd:2.4** from Docker Hub.

Our plan is to have requests made to our IP address on port 8080 be redirected to port 80 on the container. Also, instead of serving content from the container itself, we will serve a simple web page from **/home/student/website**.

We do this by mapping **/home/student/website/** to the **/usr/local/apache2/htdocs/** on the container. The command is as follows. Note it is one continuous line of command below.

|  |
| --- |
| sudo docker run -dit --name student-web -p 8080:80 -v /home/student/website/:/usr/local/apache2/htdocs/ httpd:2.4 |



If the command runs successfully, our Apache container **httpd:2.4** should be up and running. We can check the docker process using **ps** command.

|  |
| --- |
| sudo docker ps |

A screenshot of a computer

Description automatically generated

Now we create a simple web page html file named **container.html** inside the /home/user/website directory using vi editor.

|  |
| --- |
| sudo vi /home/student/website/container.html |

Our simple container.html file is as follows:

A screen shot of a computer

Description automatically generated

Next, point our VM’s browser to **127.0.0.1:8080/container.html**. You should be presented with the page we created previously.

**A screenshot of a computer

Description automatically generated**

If we see this page, it shows our VM is running the Apache Web Server container and displaying the simple web page we created at the VM’s IP address 127.0.0.1.

**Connecting to the VM’s Apache’s Web server**

If we need to test whether we can connect to the VM’s Apache Server from the host PC or from other VMs, we can modify the VM’s network adapter settings in Virtual Box.

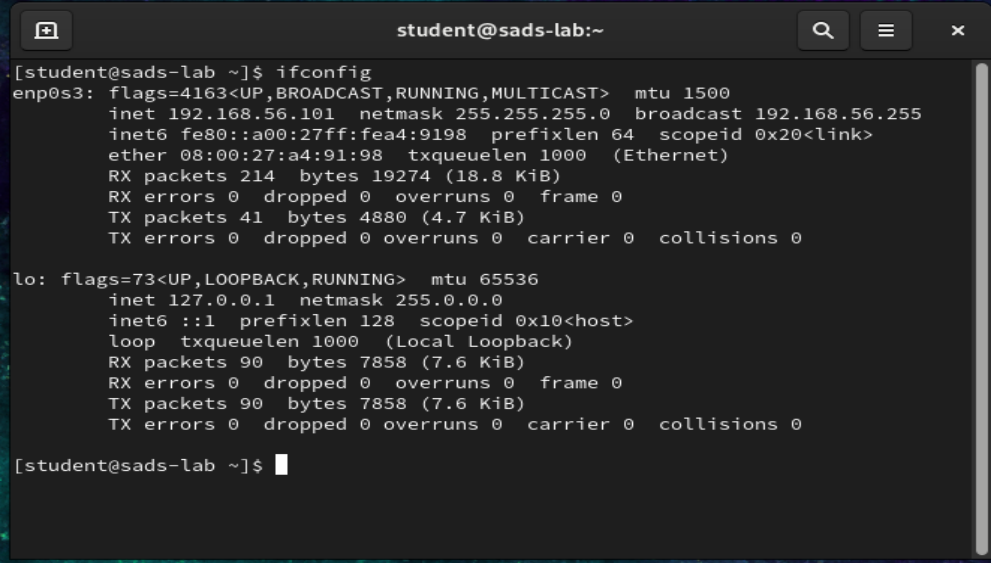
In Virtual Box, select Machine > Settings > Network.

Change adapter 1 from using the drop down menu to change NAT to **Host only adapter**.

A computer screen shot of a network

Description automatically generated

Using **ifconfig** command**,** we can see the VM’s IP address, in this case it is 192.168.56.101.



We can then use the host PC’s browser or other VM in the same network to connect to this VM’s Apache server, using this IP address.

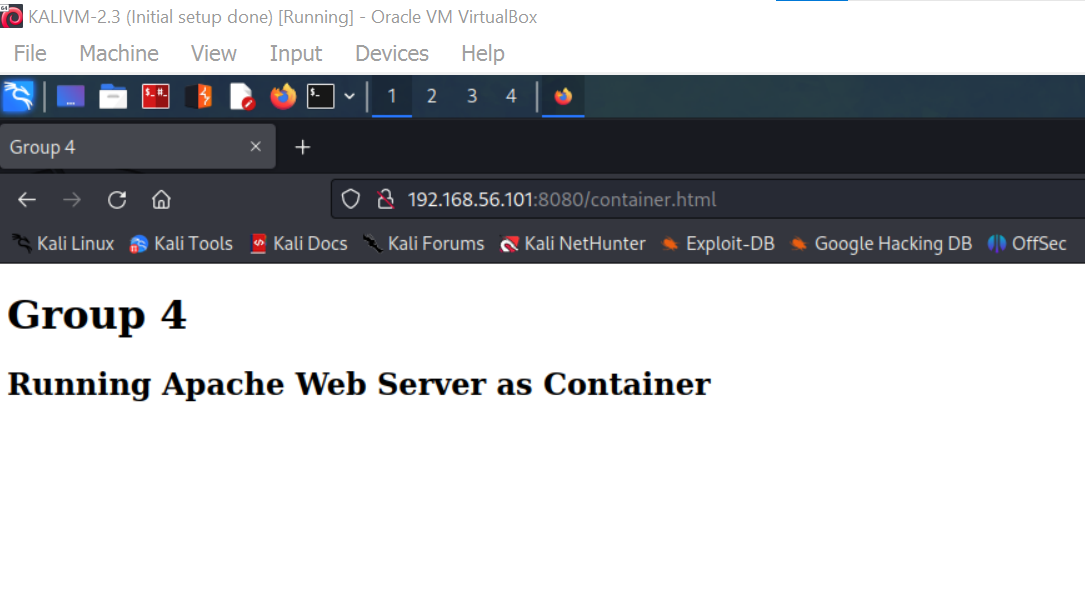
For example. we can start the host PC’s browser and point it to **http://192.168.56.101:8080/container.html**

A screenshot of a computer

Description automatically generated

If we can see this webpage displayed, it shows we have successfully connected to our VM’s Apache’s web server displaying our simple web page using our host PC’s Edge browser.

Connecting using another VM in the same network is also possible. For example, we can start another VM’s browser and point it to our VM’s IP address **192.168.56.101:8080/container.html.**



If we can see this simple web page, it shows we are able to access our VM’s Apache Web server from another VM.

**Cleaning Up**

To stop the container and remove it, use the following commands:

|  |
| --- |
| sudo docker stop student-web  sudo docker rm student-web |

To delete the Apache Web Server **httpd:2.4** image used in the container, use the following commands:

|  |
| --- |
| sudo docker image remove httpd:2.4 |

**Conclusion**

After setting up a VM, the next question is to ask the most efficient method to add services and software so that the VM can be useful to the users.

Containers isolate the software and manage its fundamental dependencies, enabling it to run consistently across various environments. This capability grants users the flexibility of portability and the ability to create multiple instances of the software as needed, all while efficiently utilizing resources.

**Reference**

How to set up Apache Web Server in a Docker Container. Link: <https://www.tecmint.com/install-apache-web-server-in-a-docker-container/>