

LOW COST COMPUTER LAB SETUP FOR SCHOOLS AND PANCHYAT USING DOCKER

PROJECT REFERENCE NO: 46S__BE_3315

Shridevi Institute of Engineering and Technology

Branch : Information Science and Engineering

Student(s) name:

Guide Name: Prof.Girish L

Raviteja S [1SV19IS013]

Priya Agadi [1SV19IS012]

Chaithra B S [1SV19IS002]

Thanmayi P [1SV19IS022]

INTRODUCTION

Nowadays traditional computer labs consist of several computers along with a CPU but it cost a high price and consumes more space to overcome this virtualization is introduced. Developers realized that the process of developing a particular application may differ developing on the environment in which the application will be executed. Therefore, they must provide all requirements of the environment in which the application running before starting to develop it. the most common of them are virtualization and containerization. In this project, we choose to design the proposed project in the shape of a portable program depending on containerization with the Docker container platform. The Docker container platform can host the program and attach all of its dependencies in one unit (container) to run the application in a number of environments. The client-server model is used in most network applications. Both client and server containers are based on ubuntu and developed using Docker network. The Docker will build the containers depending on a writing Dockerfile for each container and push them to the

registry (docker hub). When pulled the image of the containers from the registry account, then it could be run the container on a host.

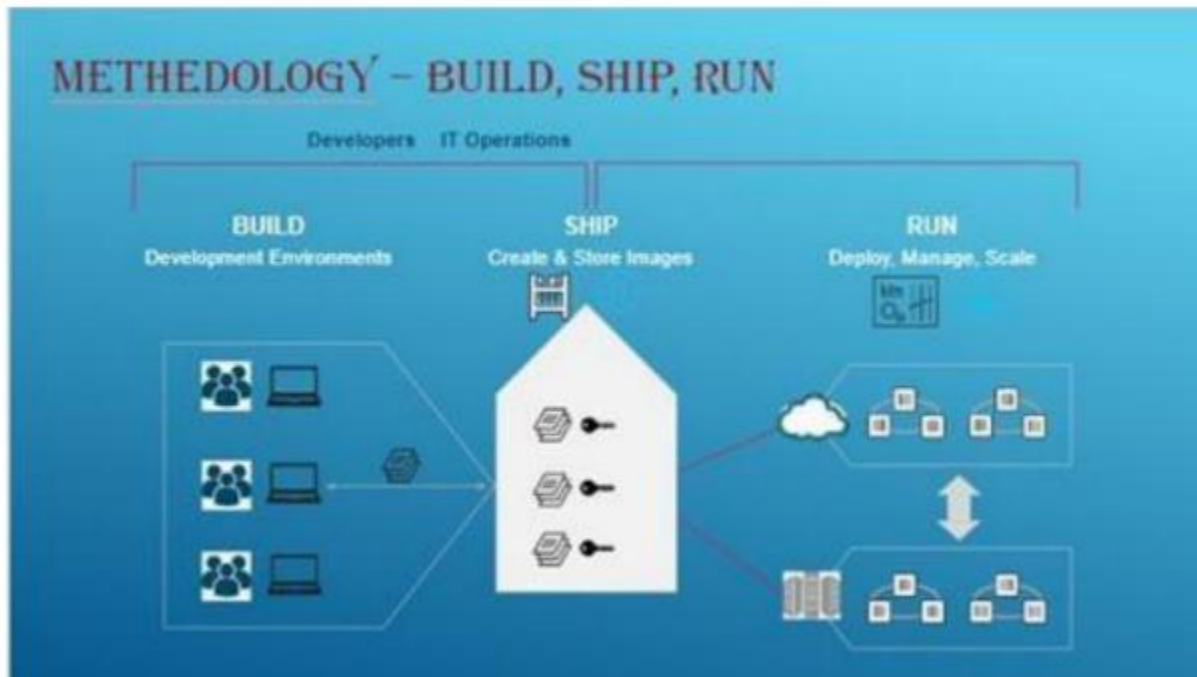
Keywords: Docker, Thin-client, Container, DevOps

OBJECTIVES

This project aims to provide computer education to schools at a low cost and easily used by students.

- The main aim is to outline the different goals and objectives of included studies, we have further plans to improve the standards of schooling by introducing some basic computer education.
- The use of new technologies like DevOps leads to the proper utilization of resources and time.
- A client-server model is used in many networks and an effort of many programmers going to build and implement this model as well as a configuration process to run it in different environments. The process of providing these applications to developers or companies decreases the effort and the cost. Therefore, striving in this research provides a client-server model in the form of containers that can be run in any execution environment.

METHODOLOGY



As discussed earlier, each Docker container has its own filesystems, namespaces, and cgroups. These three features help us to understand working on Docker containers.

- **Filesystems:** – Docker containers use a stackable filesystem that means files and directories in different branches can be overlaid to make a single file system. It helps us to avoid duplicating data each time we deploy the container.

- **Namespaces:** – Each container has its own multiple namespaces, and each namespace has a different type of information about the OS. For example, the MNT namespace has information about mounted filesystems that containers can use; similarly, a USER namespace has information about container users and group IDs.

- **Control groups:** – Control groups are responsible for managing resource usage of containers such as CPU, memory, disk I/O and network. We can also restrict usage limits using Control groups.

PROJECT INNOVATION



Fig: Client Server Automation Setup

- Instead of using high cost CPU we use low cost thin clients and a server
- We written own Dockerfile for installation running of applications
- This setup consist of server and client where we write a Dockerfile to connect the server and client.
- With shared terminal services, all users at thin client stations share a server-based operating system and applications.
- The each desktop acts like a whole computer system, which is partitioned off from other computer systems in the server
- . A browser-based approach to using thin clients means that an ordinary device connected to the internet carries out its application functions within a web browser instead of on a remote server

RESULTS AND CONCLUSION

Client setup

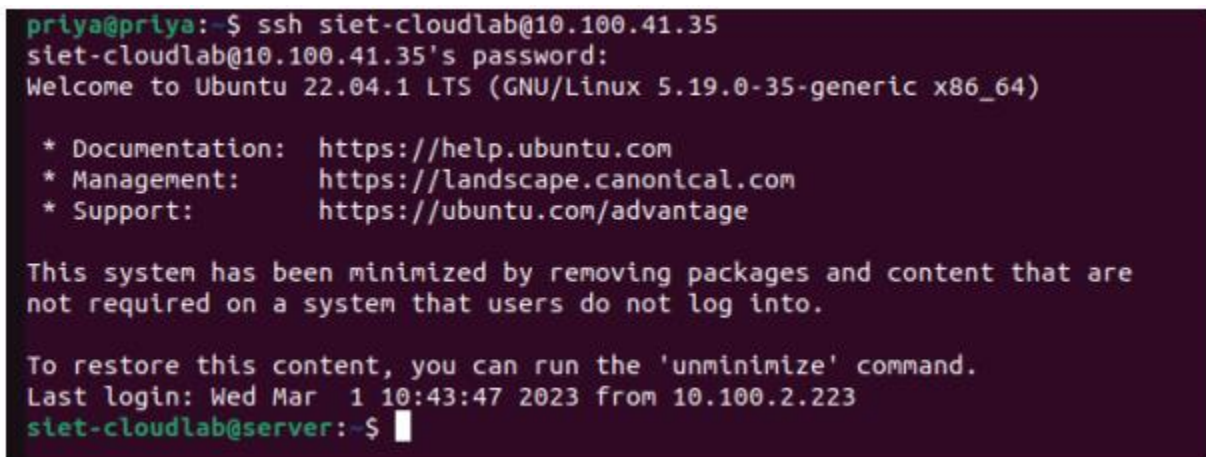
Use the below command to install openssh-client

```
sudo apt install openssh-client
```

You can verify it by using the below command.

ssh -V Login to the already created Container

```
ssh siet-cloudlab@10.100.41.35
```

A terminal window showing an SSH session. The user 'priya' at 'priya' runs the command 'ssh siet-cloudlab@10.100.41.35'. The terminal displays the password prompt, a welcome message for Ubuntu 22.04.1 LTS, and links for documentation, management, and support. It also shows a message about system minimization and the last login time.

```
priya@priya:~$ ssh siet-cloudlab@10.100.41.35
siet-cloudlab@10.100.41.35's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.19.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Wed Mar  1 10:43:47 2023 from 10.100.2.223
siet-cloudlab@server:~$
```

The command `ssh siet-cloudlab@10.100.41.35` initiates an SSH connection to a remote host with the IP address 10.100.41.35 and the username siet-cloudlab.

When you execute this command in a terminal or command prompt, the OpenSSH client (assuming it's installed on your system) will establish an encrypted SSH connection to the specified remote host.

Once the connection is established, you will be prompted to enter the password for the specified username on the remote host. If the password is correct, you will be logged in to the remote host's shell environment, and you can execute commands or transfer files securely between your local machine and the remote machine.

Once the connection is established, you will be prompted to enter the password for the specified username on the remote host. If the password is correct, you will be logged in to the remote host's shell environment, and you can execute commands or transfer files securely between your local machine and the remote machine.

SCOPE FOR FUTURE WORK

- ✓ **Improved Hardware Integration:** Explore the integration of energy-efficient and low-power hardware components to further reduce energy consumption and operating costs.
- ✓ **Expanded Software Capabilities:** Continuously update and expand the range of educational software and tools available in the Docker containers to cater to evolving curriculum needs and technological advancements.
- ✓ **Enhanced Remote Management:** Implement advanced monitoring and management tools to remotely track system performance, troubleshoot issues, and provide proactive maintenance, minimizing downtime and improving overall system efficiency.
- ✓ **Internet Connectivity Options:** Explore alternative connectivity options such as satellite internet or wireless mesh networks to ensure reliable and high-speed internet access in remote areas where traditional wired connections are limited.

A handwritten signature in black ink, appearing to be 'Jen', is written over a solid horizontal line.