IT Platform Project Report

Virtual Machine and Docker

Winter 2024/25

Prepared for:

Prof. Dr. Rand Kouatly

University of Europe for Applied Sciences

Prepared by:

Alvaro , Batool , Bibhushan, Omar, Roza

Submission Date: 15 January 2025

TASK 1: Firewall

1. Check and correct all network configurations
    a. Check the configurations of the firewall by running the command **show running-config**
        i. Verify if all the ip addresses are accurate
        ii. Check if there are any ACL rules that are too broad or too strict in that they don't allow access to any commands
    b. Check the port Ipv4 addresses and if they are correct
    c. Test connections by random computers and opening the website
        i. Used command prompt to ping the computers and internet explorer to launch the websites
        ii. Launched the website through both the website name and IP address
    d. Simplified the firewall settings
2. Change the serverpool network into DMZ and add firewalls
3. Remove the DHCP server and configure the firewall as a DHCP server
    a. DHCP server was removed
    b. Replace the server with a firewall between every router and its own network.
4. Configure the firewall to protect from intrusion from outside networks
    a. Added some ACL rules to restrict access of other networks to each network.

TASK 2 : Install and Use Linux OS

1. **Install Virtual Machine(VM)**
2. **Install Ubuntu and Derbian in VM**
    a. Configure network for both. To configure both linux and get ip address from connected internet . We have to open setting in VM for which OS you want to configure.
    b. Then Click Network -> Adapter 2 and Change attached to Host-only Adapter.
    c. Run both OS at the same time and ping from Ubuntu to Derbian and viceversa.

3. **In Ubuntu**
    a. Create 2 seperate users . To create seperate users you can do while setting up the OS in the beginning or you can create from the setting of the device.
    b. To change the view to Full screen in OS , we can see View option in title bar at the top click on that and you can see option to make Full screen or use Shortcut key Host + F (Host = Right Ctrl)
    c. To Install Java in VM use cmd in terminal
        - Sudo apt install openjdk-17-jdk
    d. Create and run small java program
        - In terminal change the path where you want to create the file and use *touch helloworld.java* to create java file
        - Use *nano helloworld.java* to open code editor for this java file.
        - Write you code in nano , to save you have to press CTRL + O and to exit ou have to press CTRL + X
        - After that you have to compile the javafile. To compile use *javac helloworld.java*
        - And *java helloworld* to run the javafile.
4. **In derbian**
    a. To configure sudo user you have to use –*su - username* cmd in terminal or if you alredy have sudo user then use only –*su-* (where in username use our own username).
    b. Install synaptic Package.
        - Use *sudo apt install synaptic –y* to install
        - *sudo synaptic* to open/launch
    c. Install PhotoEditor from Synaptic
        - To install any Photo editing software , we have to type name of any software in the search bar .
        - When it show the result Right Click and Mark for Installation .
        - Then , Click apply .
        - When Download is complete . For example i downloaded GIMP .
        - To open that type *gimp* in terminal or open directly from download folder.

TASK 3 : Install and Uses Docker in Ubuntu OS

1. **Install Docker**
    - To install Docker in your Linux OS use following cmd:

        - *sudo apt install docker-io*
        - *sudo systemctl enable docker*
        - *sudo systemctl status docker* (This is to check wheather its running or not).
        - *sudo systemctl start docker* (To start Docker).
2. **Create Docker Account**
    - Create your docker account using your email from the Docker official website .
    - Using that Login Details you can now login with Docker
        - *sudo docker login*
    - You have to type you login details and then you are logged in with Docker .
3. **Pull hello world image and run it.**
    - Use *sudo docker pull hello-world* to pull image.
    - Use *sudo docker run hello-world* to run .
4. **Pull Ubuntu Docker image from Docker hub, and run the image.**
    - We have to pull Ubuntu Docker Image and use for that we have to type *sudo docker pull ubuntu*
    - *sudo docker run –it ubuntu bash* after this you are using Ubuntu root from Docker but it will have only limited command .
5. **Check your running containers.**
    - To check running container you have to use *sudo docker ps* , This will show you all the running container in Docker.
6. **Run the batch in the Ubuntu container, and see the bin folder, and create new ls command.**
    - Run *sudo docker run –it ubuntu bash* to run Ubuntu Container .
    - After that to see whats in bin type *ls bin* , it will list all the files inside bin folder.
    - Now to create new ls cmd type
        - *cp /bin/ls /bin/my-ls*
        - */bin /my-ls*

TASK 4 : Create Java Docker or Python Container.

For this we have to use the Java sample file we have created in Task 2 or Create new java File .

a. To create new directory using command line in Ubuntu .
    - *Mkdir JavaCode* (Where JavaCode is the name of the Directory)
    - If we want to create javafile again we follow the the steps.
    - Only thing you have to do first is change path to JavaCode first
        - *cd Documents/JavaCode /*
        - *touch Hello-UE.java*

        - *nano Hello-UE.java*
        - *javac Hello-UE.java*
        - *java Hello-UE*
b. Create Docker File.
    - To Create Docker file and type instuction or command inside the Docker File , we have to use following instruction
        - *cd Documents/JavaCode/*
        - *touch Dockerfile*
        - *nano Dockerfile*
    - *Now you have open nano for Docker file.You have to write following command inside the files.*
        - *FROM openJDK*
        - *WORKDIR /app*
        - *COPY . /app*
        - *RUN javac Hello-UE.java*
        - *CMD ["java" , "Hello-UE""]*
c. Create JavaContainer
    - To create JavaContainer use *sudo docker build –t javaprogram .*(there is fullstop at the end of the cmd we should not forget that).
    - *Sudo docker image ls* (When you use this cmd you can check wheather the container is created or not with name **javaprogram**)
d. Run Container to see result.
    - *sudo docker run –name jave1 javaprogram*
    - When we use this cmd you can see the output of the contanier which we have created.

TASK 5: Change System Configuration using Python Container

a. Create a new directory using cmd line called myproject.
    - mkdir myproject
    - cd myproject/
b. *Myproject folder must have python file and create another directory called templates which contain HTML file .*
    - *To create .py file use*
        - *touch app.py*
        - *nano app.py*
    - *Inside app.py paste the code given by the professor in Pdf for the assignement or you can use your own*
    - *Next we have to create templates directory*
        - *mkdir templates*

        - *cd templates*
    - *Now create index.html in templates*
        - *touch index.html*
        - *nano index.html* (Add code from PDF or use our own HTML code)

c. *Create Docker file*
    - *Create Docker file in myproject folder which we have just created.*
        - touch dockerfile
        - nano dockerfile
    - *When you open nano code editor paste the following command inside docker file*
        - FROM python: 3.9
        - WORKDIR /app
        - COPY app.py /app
        - COPY templates /app/templates
        - RUN pip install Flask
        - ENV MY_COLOR = lightblue
        - EXPOSE 8080
        - CMD [“python” , “app.py”]

d. *Run Docker file to create python Container*
    - sudo docker build –t my-flask-app .

e. *Run Python docker container*
    - sudo docker run –d –p 8080:8080 –name my-python-app my-flask-app
    - *Now open http://localhost:8080*

Conclusion

This project successfully integrated modern IT practices, showing a good command of network security and system configuration, application deployment. Implementation of the DMZ and firewall, configuration of virtual machines with Ubuntu and Debian operating systems to communicate with each other, deployment of Java and Python programs using Docker containers, are highlighting scalability and security concerns. The development of a lightweight web app using Python and Flask further demonstrates practical programming skills. The results of the project in general testify to deeply advanced IT solutions and the ability to deliver secure, efficient, and scalable systems.