LOMBA KOMPETENSI SISWA
SMK TINGKAT NASIONAL KE-29
TAHUN TAHUN 2021

TEST PROJECT
MODUL C - NETDEVOPS

BIDANG LOMBA

TEKNOLOGI INFORMASI SISTEM ADMINISTRASI JARINGAN
IT NETWORK SYSTEMS ADMINISTRATION

DESCRIPTION OF PROJECT

A small group of programmers is starting to adopt Docker to containerize their services. Their network infrastructure is usually configured by calling the APIs from a programming language. Recently their netdevops engineer has gotten a long paid leave, you need to take over his responsibility.

NETWORK AND SYSTEM PROGRAMMABILITY

API Configuration

- Access development server to configure management router.
- Enable restconf API and/or other tools to manage this router’s interface via script.
  - Configure IP Address of network device according to the addressing table.
- Write a python3 script to configure IP addresses via the API.
  - Save the script in /home/ubuntu/network-change.py
  - When the script is executed, IP of all routers will be changed to the correct addresses according to the Appendix.
  - Use restconf API for highest score, otherwise you can use any other tools.

Python Web Application

- Programmers asked us to make sure they can run sample code below in the python3 virtual environment in the development server.
- Install python3 and other required packages to run the code main.py
- Enable python3 virtual environment and create a new virtual environment on “/opt/env/myenv”
  - Install the required packages to run the code in this virtual environment
- Save the following code in directory “/opt/sample-web”
  - main.py

```
from flask import Flask
from flask import request
from flask import render_template

sample = Flask(__name__)

@sample.route("/")
def main():
      return render_template("index.html")

      if __name__ == "__main__":
      sample.run(host="0.0.0.0", port=8080)
```

  - index.html

```
<html>
<head>
      <title>Sample app</title>
</head>
<body>
      <h1>You are calling me from {{request.remote_addr}}</h1>
</body>
</html>
```

- Make sure the code can be executed in the server without any trouble

CONTAINERIZED SERVICE

All docker-related tasks will be done on this server.

Docker Installation/Configuration

- Docker is already Installed.
- Configure Docker in the development server, make sure user ‘patah’ can use docker command without sudo.
- To reach the internet, use the 2nd interface with IP Address from DHCP.

Local Registry

- Run registry server container
  - Container Name : registry
  - Use image registry from Docker Hub
  - Volume Mounts : /mnt/registry to /var/lib/registry
  - Configure to Listen in port 5000
- Pull image **python:3.8-slim** from Docker Hub.
  - Push this image to the local registry container with the same name and tag.
- Pull image **nginx:latest** from Docker Hub.
  - Push this image to the local registry container with name **webserver:base**

Running Python Web Application

- Use the code Python Web Application in /opt/sample-web to create a Dockerfile
  - Use base image python:3.8-slim
  - Expose port 80
  - Copy all code into workdir.
  - Use /opt/sample-web as workdir inside the container.
  - Save the Dockerfile in /opt/sample-web/Dockerfile
- Build the Dockerfile into image with name **sampleweb:latest**
  - Run the application from this image
  - Container Name : sampleweb
  - Configure to Listen in Port 80
- Push the image to the local registry container with the same name and tag.

Running Normal Services

- Use the image **nginx:latest** and create a container based web server serving the file public.html as specified in the appendix.
  - Container name : farm-1
  - Make it accessible via http://{ip_address}/public.html
  - Do not expose any port on the host.
- Use the image **nginx:latest** and create a container based web server serving the file external.html as specified in the appendix.
  - Container name : farm-2
  - Make it accessible via http://{ip_address}/external.html
  - Do not expose any port on the host.

APPENDIX

Development Server Console Access

Username	**user**
Password	Skills39

Addressing Table

Device Name	**IP Address CIDR**	**Remark**
Management Router	10.110.101.1/24	Management
	10.200.200.1/24	Testing
Development Server	10.110.101.10/24	Management
	DHCP	ISP

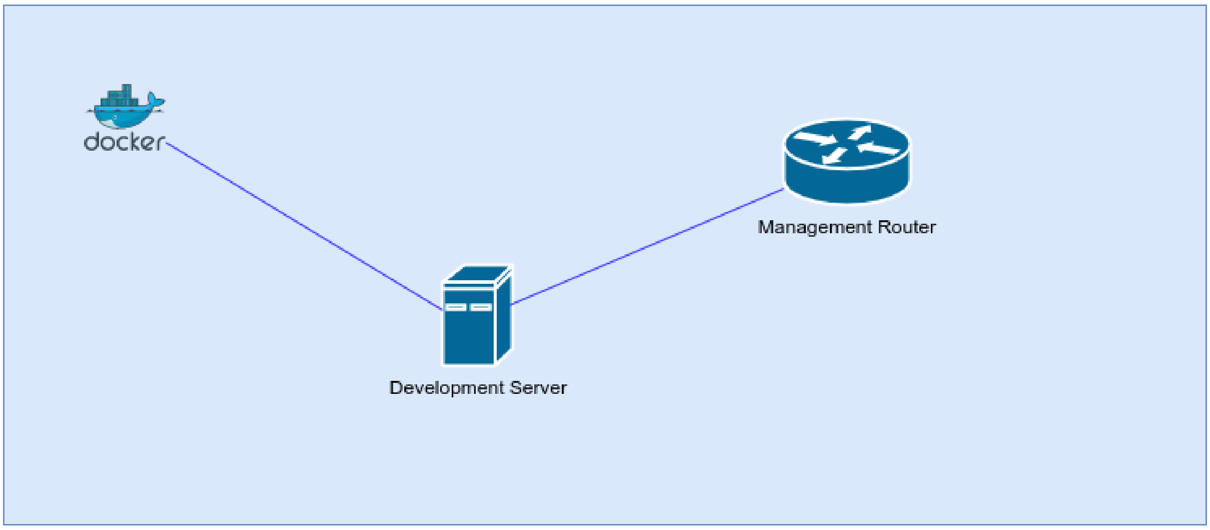
public.html

```
<h1> Public Web Page </h1><br>
<br>
Welcome to ITNSA LKSN 2021
```

external.html

```
<h1> External Web Page </h1><br>
<br>
Welcome to ITNSA LKSN 2021
```

TOPOLOGY


docker
Management Router
Development Server