TIE-23546 Cloud Platforms
Exercise 3: Operating System Level Containers
v1.3

Exercise 3: Operating System Level Containers

In the exercises one and two, we used VMware Workstation virtualization software. We defined computing, memory, storage capacity and network adapters for the virtual machine. This is a traditional approach to run virtual machines. In this exercise, we focus on operating system level containers. Containers are light compared to traditional virtual machines. Using same resources, we can run more containers than virtual machines. Modern cloud platform operators support traditional virtual machines and operating system level containers. If you want to save money and resources, operating system level containers might be the right selection. At the same time, it should be noted that Microsoft Windows does not support operating system level containers.

Linux LXC and LXD

Linux kernel has native support for operating system level containers. In this exercise, we use Linux LXC and LXD container technologies. Linux kernel has native LXC support. LXD adds additional management layer to LXC. In a real environment, running operating system level containers inside a virtual machine is not a good idea, but we can accept that in teaching. The following instructions are made for Microsoft Windows. If you are using MacOS, then you have to adapt instructions.

1. You must use VMware. If you are using Oracle VirtualBox, move on to the next exercise. Do exercises 3, 7 and 8 in a row.

2. Take a snapshot [VMware Workstation\ Ubuntu Server 01 -> Snapshot\ Take Snapshot\ Snapshot 3].

3. Select NAT adapter [VMware Workstation\ Ubuntu Server 01 -> Settings\ Network adapter\ NAT].

4. Turn on the Ubuntu Server 01 and open the SSH connection.

5. Install and configure LXD. Read installation instructions from the link.

6. Answer following questions:

Question	Command
List all images	
List all Debian images	
Launch latest version of Debian	
List all installed containers	
Display container’s information	
Move inside your container	
Start a container	
Stop the container	
Create an LXC snapshot	

Table 1. LXC and LXD commands.

7. Launch a latest version of Debian.

8. As we can see, a new virtual instance is launched in the blink of an eye.

9. Move inside Debian container and install Apache web server. Read installation instructions from the link. Do not configure the firewall.

10. Debian container works like a regular operating system. Log out of Debian container and execute the following commands in Ubuntu host. Paste output to the table 2.

```
# curl -I --http2 <IP address to the Debian container>
```

Table 2. LXC and LXD commands.

11. Debian container have private IP addresses. In the following chapter we connect Ubuntu 18.04 container to the LAN (Local Area Network).

Connecting LXD container to the LAN

By default, LXC containers are connected to a private subnet. The containers cannot be reached from the public network. We will create a new LXD profile and connect containers to the public subnet. In reality, we still use private IP addresses, but the same idea is used to connect containers to the public network.

1. Create a new LXD profile and use macvlan to connect container to the LAN. Then, launch one Ubuntu 18.04 container under the new profile. Name the new container to Minetest. Read installation instructions from the link.

2. You cannot connect from Ubuntu host to Minetest container because netplan does not support macvlan interface. This is confusing, but you can connect to Minetest container from the desktop PC and other virtual instances in same the LAN.

3. Move in to Ubuntu 18.04 container and install Minetest server. Minetest is an open source version of Minecraft. Read installation instructions from the link. Note the following refinements:

    - Install packets wget, nano and software-properties-common.
    - Do not configure the firewall.
    - Replace command `su minetest` with command `su minetest -s /bin/bash`.
    - Add the Minetest container's IP address to bind_address variable.
    - Skip SQLite3 definitions.
    - Do not define minetest.service file. Start Minetest server directly from the command line.

4. Download Minetest client from Minetest home page (Link). Save ZIP file to directory C:\Temp and extract the file.

5. Rename minetest.conf.example to minetest.conf. Open minetest.conf with Notepad program. Make the following changes and save the file.

```
# enable_sound = true
--->
enable_sound = false
```

6. Launch Minetest client and open the Join game tab. Connect to local Minetest server, and enter your usename / password. You can freely choose your username and password. Finally, press the connect button and re-enter the password.

7. The journey was a long, but now we can relax and play a couple of minutes at Minetest.

8. Wait for daylight and climb the mountain. Take a postcard photo and add the screenshot to appendix one.

9. Take a screenshot of the Minetest server console window and add the screenshot to appendix two. Trim the screenshot if necessary.

Finally

1. Close Minetest client. Remove the ZIP file and the Minetest subdirectory under C:\Temp.

2. Kill Minetest server process and shutdown Minetest container (poweroff)

3. First, shutdown Ubuntu Server 01 first (sudo poweroff). Then, close WMware software.

4. Do not destroy the virtual machine. We will use it in future exercises.

5. Save the final report to your home directory in Word Doc and Adobe PDF format (Word / File / Save as / PDF).

6. Return the final report in PDF format.

7. Good work! It is time to go for lunch or coffee.

Appendix 1: A screen shot from Minetest game.

Appendix 2. A screen shot from Minetest server console.